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.

SH7137 Group

Sample Application for the CAN Module (Remote Frame Transmission)

Introduction

This application note describes the controller area network module (RCAN-ET) and provides an example of its application to remote frame transmission.

Target Devices

SH7137

Contents

1. Preface

1.1 Specifications

- Transfer rate: 500 kbps
- Mailbox for transmission: Mailbox 0
- Received data frame is as follows.
  ID: 0 (standard format), data length code (DLC): 2, and data: H'C1C2

1.2 Module Used

- Controller area network (RCAN-ET)

1.3 Applicable Conditions

• MCU	SH7137	
• Operating frequency	Internal clock:	80 MHz
	Bus clock:	40 MHz
	Peripheral clock:	40 MHz
• C compiler:	SuperH RISC engine family C/C++ compiler package Ver.9.01 Release01 from Renesas Technology	
• Compiler options:	Default settings of the High-performance Embedded Workshop (-cpu = sh2 -debug -gbr = auto -global_volatile = 0 -opt_range = all -infinite_loop = 0 -del_vacant_loop = 0 -struct_alloc = 1)	

1.4 Related Application Note

None

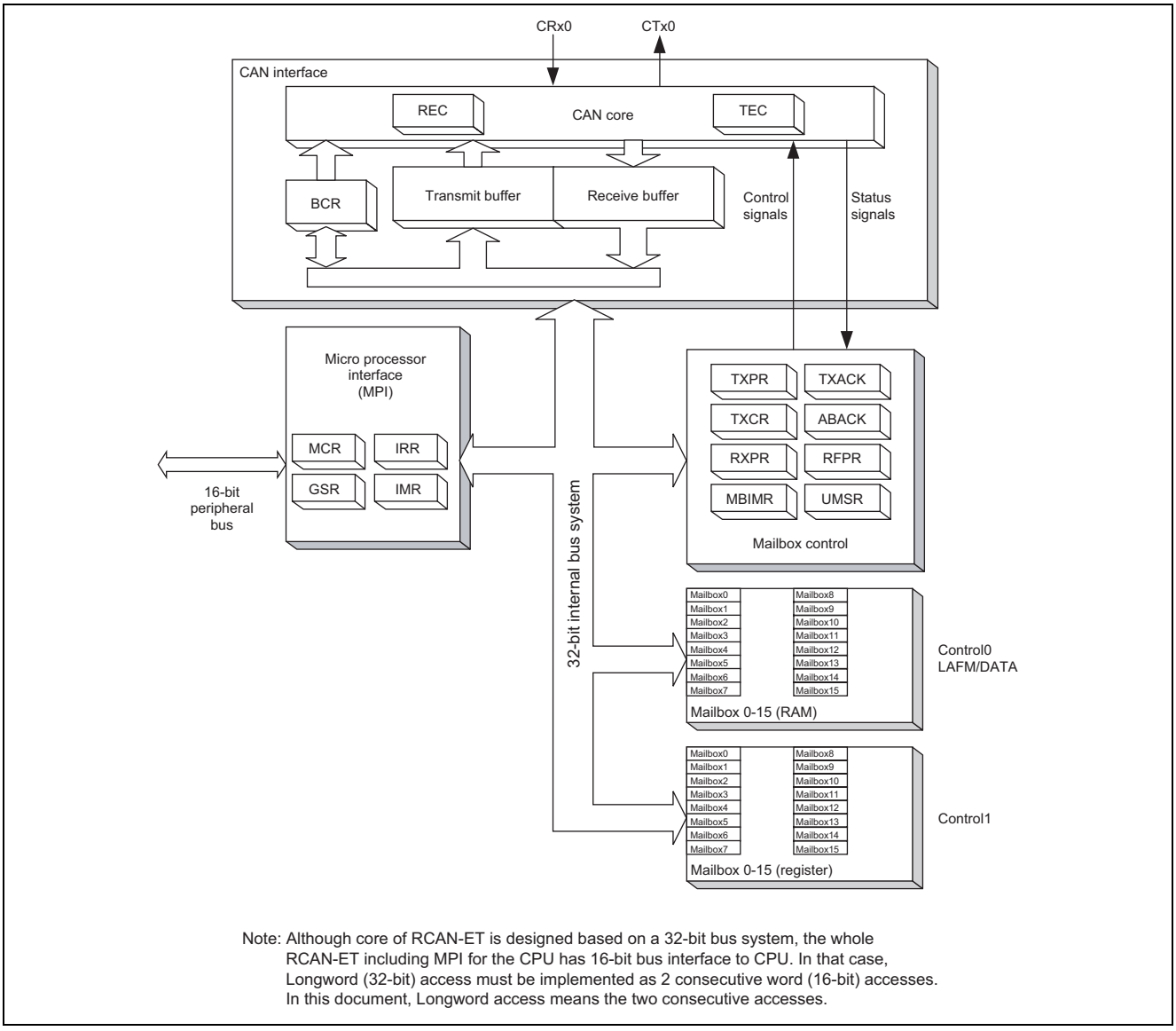
2. Description of the Sample Application

This sample program employs the RCAN-ET module to transmit a remote frame (DLC: 2) in standard format (ID: 0) and receive a data frame in standard format (ID: 0).

2.1 Overview of Operations by the Module Used

The SH7137 CPU has an internal RCAN-ET module that support CAN2.0B and comply with ISO-11898.

The RCAN-ET module has 15 programmable transmit/receive mailboxes and one receive-only mailbox, each supporting a programmable reception filter mask, providing for highly flexible communications. Figure 1 shows the structure of the RCAN-ET module. For details on the module, refer to the section on the controller area network in the *SH7137 Group Hardware Manual*.



Note: Although core of RCAN-ET is designed based on a 32-bit bus system, the whole RCAN-ET including MPI for the CPU has 16-bit bus interface to CPU. In that case, Longword (32-bit) access must be implemented as 2 consecutive word (16-bit) accesses. In this document, Longword access means the two consecutive accesses.

Figure 1 Structure of the RCAN-ET Module

2.2 Procedure for Setting the Module Used

This section describes initial settings for the transmission of remote frames by the RCAN-ET module.

Initial settings of the module are made in reset mode (configuration mode). On subsequent release from reset mode, the RCAN-ET module participates in CAN-bus activity. Figures 2 and 3 show examples of the flow of initialization for the RCAN-ET module. For details on the settings made to individual registers, refer to the *SH7137 Group Hardware Manual*.

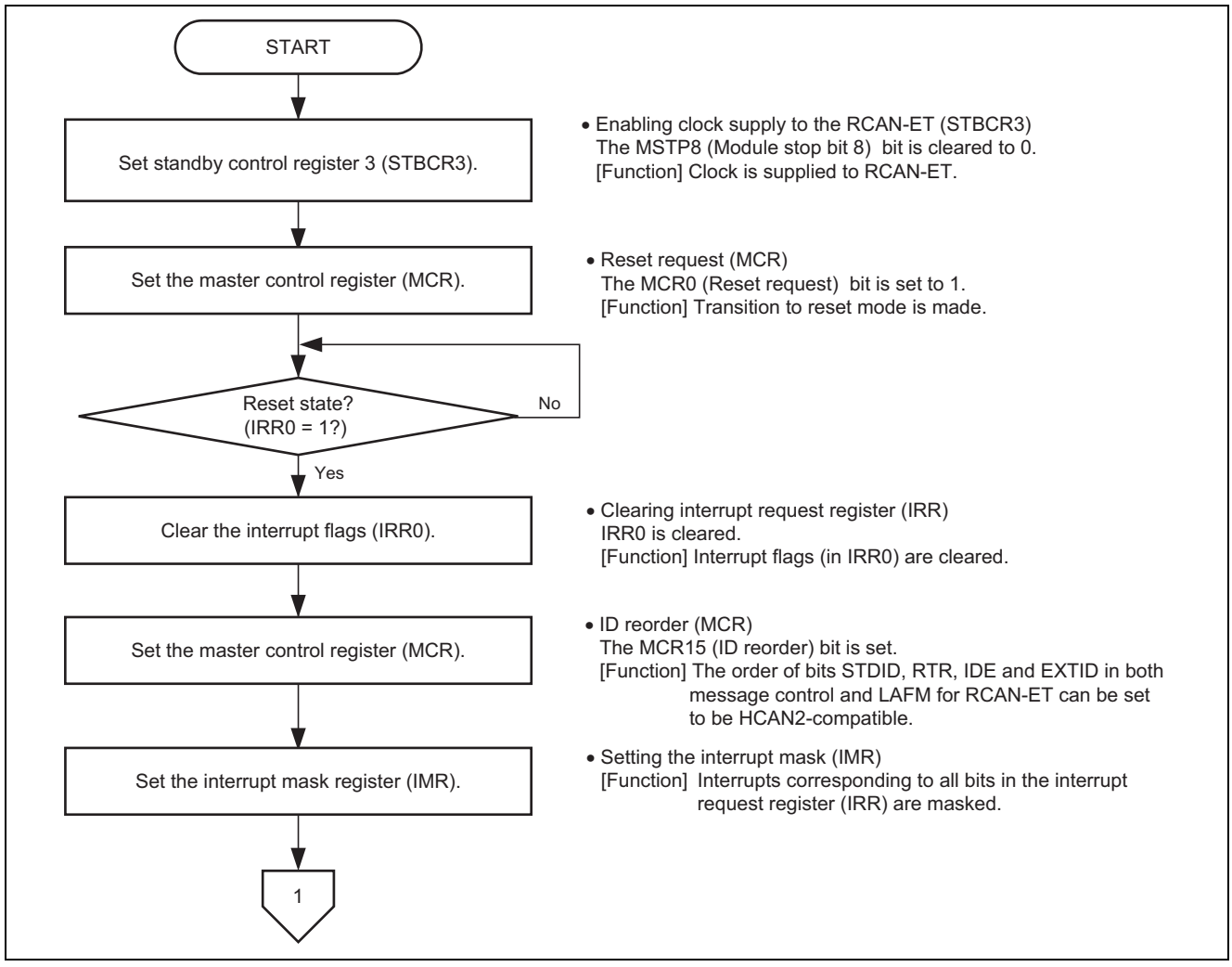


Figure 2 Example of Initialization Flow for the RCAN-ET Module (1)

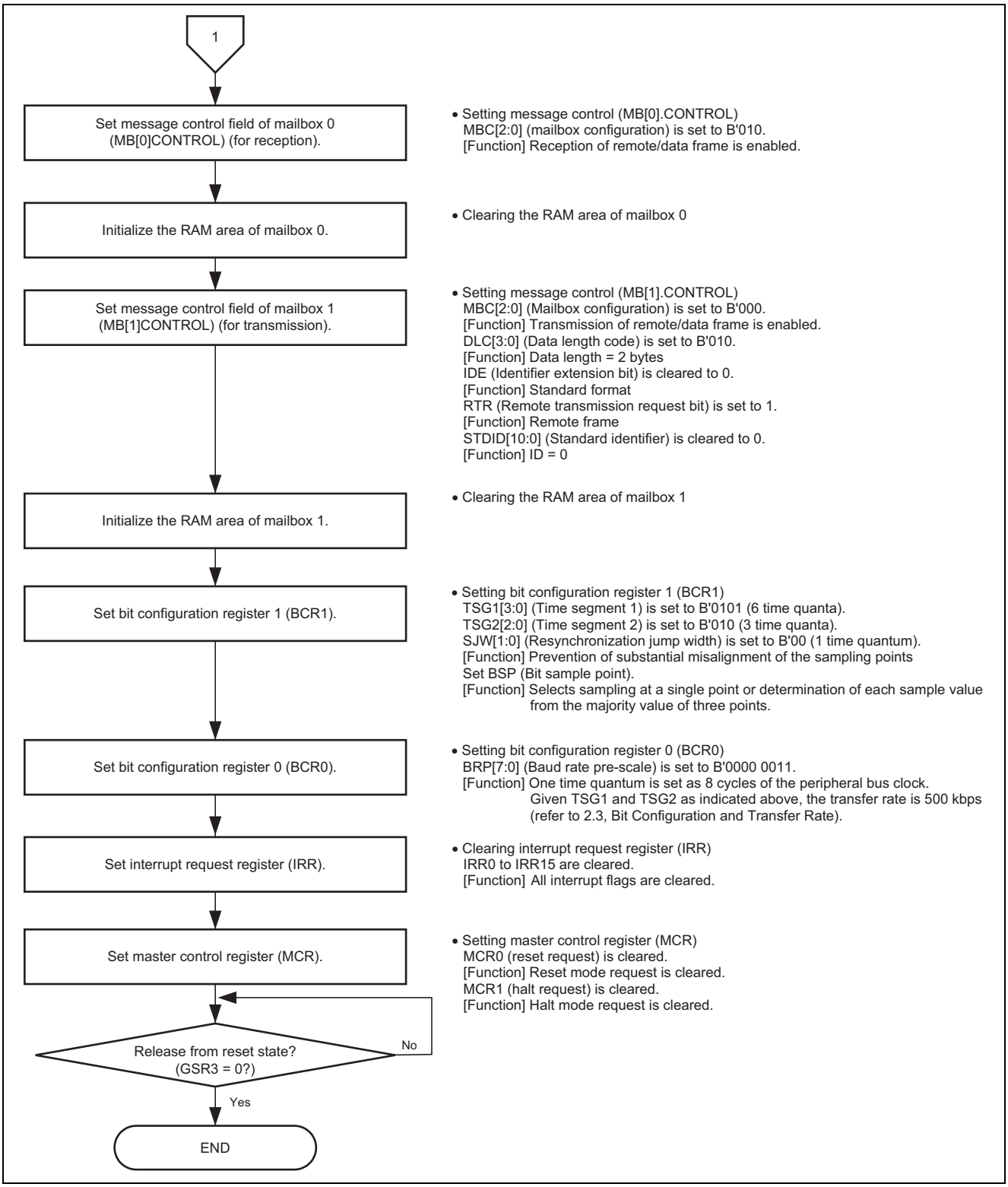


Figure 3 Example of Initialization Flow for the RCAN-ET Module (2)

2.3 Bit Configuration and Transfer Rate

One-bit time for the CAN module has the four segments indicated below.

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

Furthermore, the individual segments are structured in units of a base time called the time quantum (Tq). Figure 4 shows an example of the configuration of a bit in the case where SS = 1Tq, PRSEG = 3Tq, PHSEG1 = 3Tq, and PHSEG2 = 3Tq.

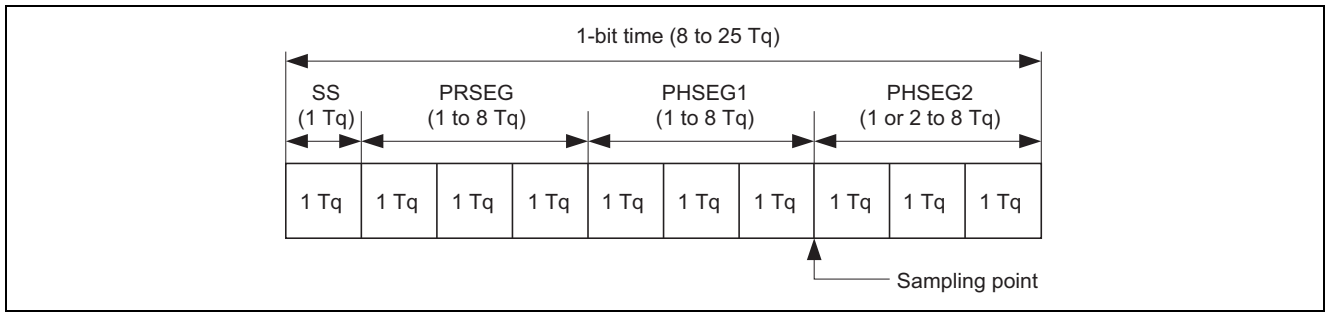


Figure 4 Configuration of One-Bit Time

In the RCAN-ET, the Tq of PRSEG + PHSEG1 is set to TSG1[3:0] in bit configuration register 1 (BCR1) and the Tq of PHSEG2 is set to TSG2[2:0] (Tq = set value + 1). Additionally, the number of cycles of the peripheral-bus clock corresponding to 1Tq is set in BRP[7:0] of bit configuration register 0 (BCR0).

In the following description, BRP[7:0], TSG1[3:0] and TSG2[2:0] indicate the register settings, and BRP, TSEG1, TSEG2, and SJW indicate the values that correspond to these register settings. For the values corresponding to the values set in registers, refer to the section on the controller area network in the *SH7137 Group Hardware Manual*.

By definition, Tq for the RCAN-ET module is 1Tq = 2 × (BRP[7:0] + 1)/peripheral bus clock, and the transfer rate is calculated as follows.

Transfer rate = peripheral bus clock/(2 × (BRP[7:0] + 1) × the number of Tq in 1-bit time) = peripheral bus clock/(2 × (BRP[7:0] + 1) × ((TSG1[3:0] + 1) + (TSG2[2:0] + 1) + 1)

The following restrictions apply to settings of the bit-configuration registers.

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

SJW: Jump width for resynchronization. This segment is used to correct phase errors by extending phase buffer segment 1 or shortening phase buffer segment 2.

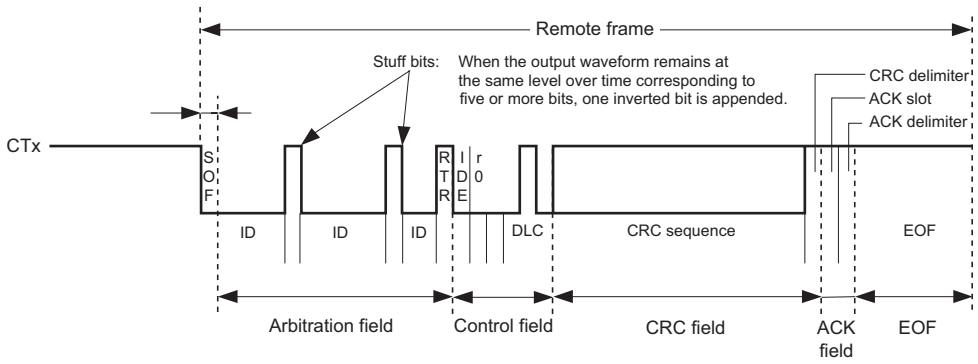
8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta
TSEG2 ≥ 2

Since the settings in this sample program are as follows: peripheral bus clock = 40 MHz, BRP = 3, TSG1 = 5, TSG2 = 2, the transfer rate is calculated with the following formula.

Transfer rate (bps) = 40 M (2 × (3 + 1) × ((5 + 1) + (2 + 1) + 1) = 500 k

2.4 Operation of the Sample Program

In this sample program, a remote frame (DLC: 2) in standard format (ID: 0) is transmitted from mailbox 1 once and then a data frame in standard format (ID: 0) is received in mailbox 0 at a transfer rate of 500 kbps. Figure 5 shows the waveform for remote frame transmission.



SOF (Start of frame)	: Indicates start of data frame and remote frame.
ID (Identifier)	: Indicates priority of data.
RTR (Request for remote transmission)	: Indicates distinction between data frame and remote frame.
IDE (Identifier extension)	: Indicates distinction between standard format and extended format.
r0	: Reserved bit
DLC (Data length code)	: Indicates the number of bytes in the data field. Setting range: 0 to 8 bytes
CRC sequence	: This value is used to check frames.
CRC delimiter	: This bit is used to delimit the CRC field and ACK field.
ACK slot	: The receiving unit is dominant, and overwrites this when no errors are detected in a transmitted message.
ACK delimiter	: This bit is used to delimit the ACK field and EOF.
EOF (End of frame):	: Indicates the end of the frame.

Figure 5 Waveform for Remote Frame Transmission by the RCAN-ET

2.5 Processing Procedure by the Sample Program

Table 1 gives an example of the settings for the controller area network (RCAN-ET). Figures 6 and 7 show an example of the flow of processing by this sample program.

Table 1 Register Settings for Controller Area Network (RCAN-ET)

Register Name	Address	Setting Value	Description
Standby control register 3 (STBCR3)	H'FFFF E806	H'F6	• MSTP8 = 0: RCAN-ET runs
Master control register (MCR)	H'FFFF D800	H'0001	• MCR0 = 1: Reset mode transition request
		H'1001	• MCR15 = 1: RCAN-ET is not the same as HCAN2
		H'1000	• MCR0 = 0: Release from reset mode
Interrupt mask register (IMR)	H'FFFF D80A	H'FFFF	• Disables all interrupts of RCAN
Bit configuration register 1 (BCR1)	H'FFFF D804	H'5200	• TSG1[3:0] = 0101: PRSEG + PHSEG1 = 6 Tq • TSG2[2:0] = 010: PHSEG2 = 3 Tq • SJW[1:0] = 00: SJW = 1 Tq • BSP = 0: Bit sampling at one point
Bit configuration register 0 (BCR0)	H'FFFF D806	H'0003	• BRP[7:0] = 3: 1 Tq = 8 × Pϕ
Message control field (MB[0].CONTROL1)	H'FFFF D910	H'0200	• MBC[2:0] = 010: Enables reception of data frames and remote frames
Message control field (MB[1].CONTROL1H)	H'FFFF D930	H'0002	• MBC[2:0] = 000: Enables transmission of data frames and remote frames. • DLC[3:0] = 0010: 2-byte data length
Message control field (MB[1].CONTROL0H)	H'FFFF D920	H'4000 0000	• IDE = 0: Standard format • RTR = 0: Remote frame • STDID[10:0] = 0: Standard ID = 0
Local acceptance filter mask (MB[0].LAFM)	H'FFFF D904	H'0000 0000	• Clear: MASK is not set
Transmit pending register (TXPR)	H'FFFF D820	H'0000 0002	• TXPR[31:0] = H'0000 0002: Generates a transmission request in mailbox 1
Transmit acknowledge register 0 (TXACK0)	H'FFFF D832	H'0002	• Clears the transmit acknowledge flag
Data frame receive pending register 0 (RXPR0)	H'FFFF D842	H'0001	• Clears the data frame receive pending flag

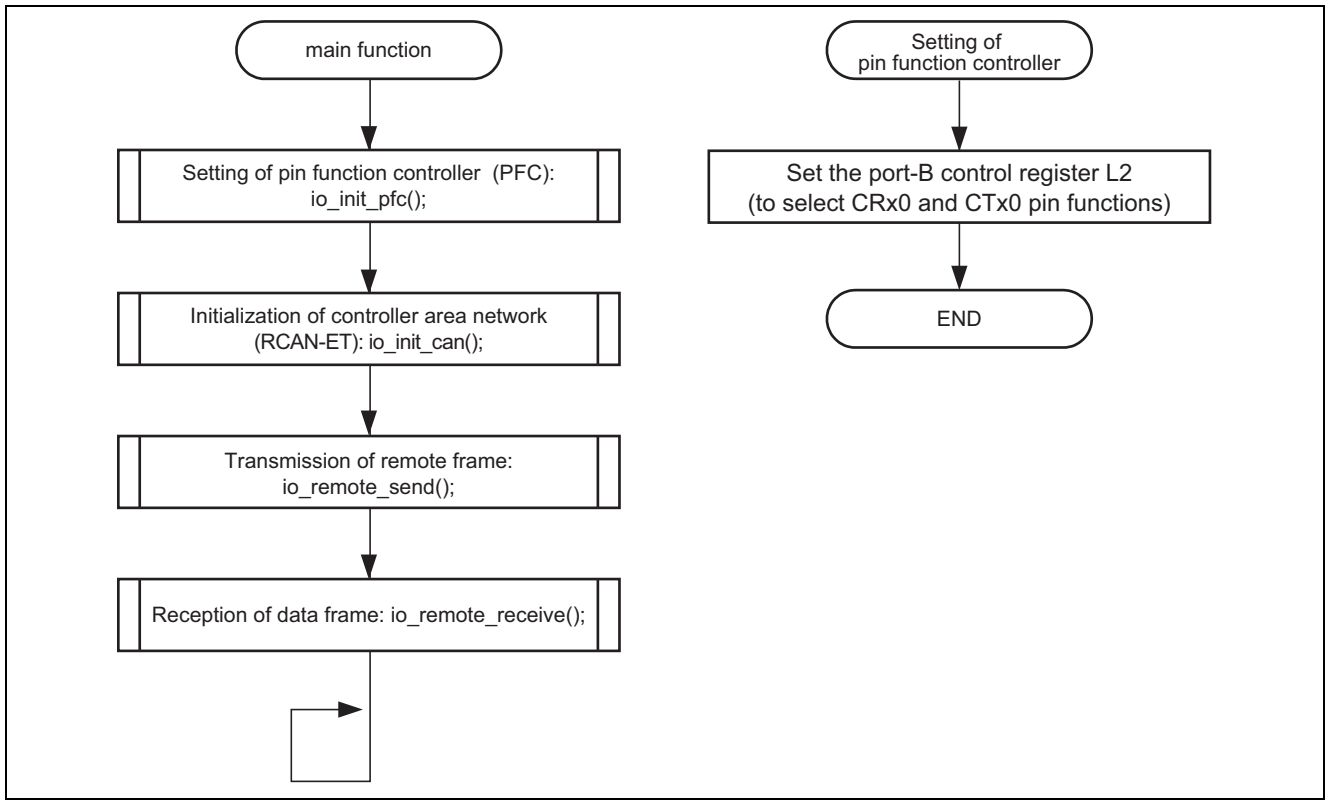


Figure 6 Example of Flow of Processing by the Sample Program (1)

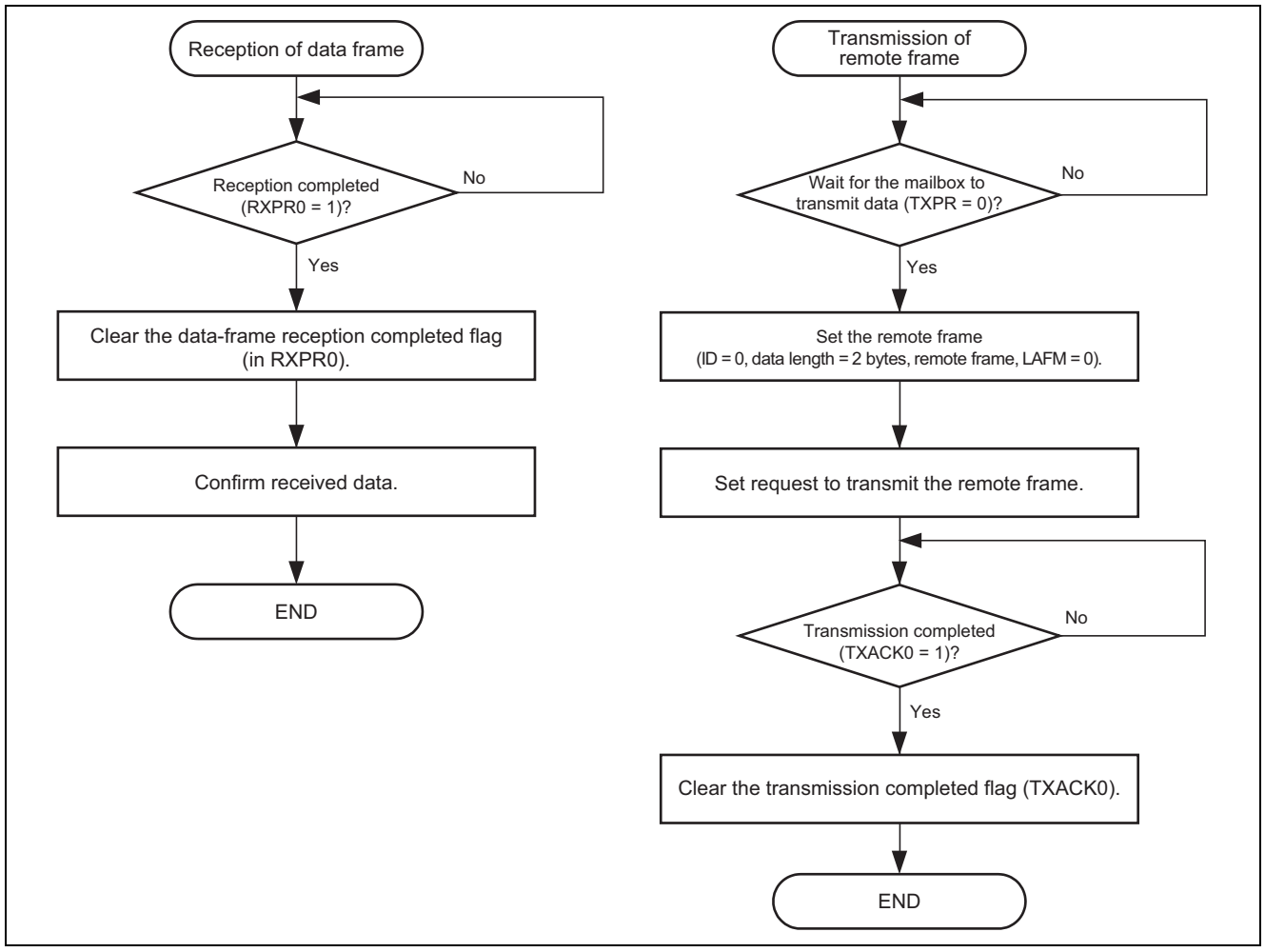


Figure 7 Example of Flow of Processing by the Sample Program (2)

3. Listing of the Sample Program

1. Sample Program Listing: main.c (1)

```
/*""FILE COMMENT""*****************************************************************
*
*       System Name : SH7137 Sample Program
*       File Name   : main.c
*       Contents    : CAN Module Application (Remote Frame Transmit)
*       Version     : 1.00.00
*       Model       : M3A-HS37
*       CPU         : SH7137
*       Compiler    : SHC9.1.1.0
*       note        : The module transmits a remote frame (DLC: 2) in standard format
*                     (ID: 0) from mailbox 1 of CAN at a 500-kbps transfer rate over the
*                     CAN bus once. After transmission, it receives a data frame from
*                     mailbox 0 of CAN1 and writes the received frame to RAM.
*
*                     <Caution>
*                     This sample program is for reference
*                     and its operation is not guaranteed.
*                     Customers should use this sample program for technical reference
*                     in software development.
*
*       The information described here may contain technical inaccuracies or
*       typographical errors. Renesas Technology Corporation and Renesas Solutions
*       assume no responsibility for any damage, liability, or other loss rising
*       from these inaccuracies or errors.
*
*       Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
*       AND Renesas Solutions Corp. All Rights Reserved
*
*       history   : 2008.03.24 ver.1.00.00
*""FILE COMMENT END""*************************************************************/
#include "iodefine.h"       /* SH7137 iodefine  */

/* ---- prototype declaration ---- */
void main(void);
void io_init_pfc(void);
void io_init_can(void);
void io_remote_send(void);
void io_data_receive(void);

/* ---- symbol definition ---- */
#define CAN_GSR3 0x0008
#define CAN_IRR0 0x0001
#define CAN_MB0  0x0001
#define CAN_MB1  0x0002
#define CAN_MB01 0x00000002

/* ---- RAM allocation variable declaration ---- */
unsigned char   nIDE = 0;           /* ide */
unsigned char   nRTR = 0;           /* rtr */
unsigned char   nDLC = 0;           /* dlc */
unsigned int    nSID = 0;           /* sid */
unsigned int    nEID = 0;           /* eid */
unsigned char   gRcv_data[8];       /* data of message */
```

2. Sample Program Listing: main.c (2)

```
 55  /*""FUNC COMMENT""*****************************************************
 56   * Outline     : Sample program main
 57   *-------------------------------------------------------------------------
 58   * Include     : non
 59   *-------------------------------------------------------------------------
 60   * Declaration : void main(void);
 61   *-------------------------------------------------------------------------
 62   * Function    : Sample program main
 63   *-------------------------------------------------------------------------
 64   * Argument    : void
 65   *-------------------------------------------------------------------------
 66   * Return Value: void
 67   *-------------------------------------------------------------------------
 68   * Notice      : non
 69   *""FUNC COMMENT END""*************************************************/
 70  void main(void)
 71  {
 72      /* ==== Setting of PFC ==== */
 73      io_init_pfc();
 74
 75      /* ==== Initializing CAN module ==== */
 76      io_init_can();
 77
 78      /* ==== CAN remote frame transmission ==== */
 79      io_remote_send();
 80
 81      /* ==== CAN data frame reception ==== */
 82      io_data_receive();
 83
 84      while(1){
 85              /* loop */
 86      }
 87  }
 88
 89  /*""FUNC COMMENT""*****************************************************
 90   * Outline     : PFC setting
 91   *-------------------------------------------------------------------------
 92   * Include     : #include "iodefine.h"
 93   *-------------------------------------------------------------------------
 94   * Declaration : void io_init_pfc(void);
 95   *-------------------------------------------------------------------------
 96   * Function    : Pin function controller (PFC) setting
 97   *-------------------------------------------------------------------------
 98   * Argument    : void
 99   *-------------------------------------------------------------------------
100   * Return Value: void
101   *-------------------------------------------------------------------------
102   * Notice      : non
103   *""FUNC COMMENT END""*************************************************/
104  void io_init_pfc(void)
105  {
106      /* ==== Setting of PFC ==== */
107      /* ---- Port B control register L2 ---- */
108      PFC.PBCRL2.BIT.PB7MD = 0x6;   /* Set CRx0 */
109      PFC.PBCRL2.BIT.PB6MD = 0x6;   /* Set CTx0 */
110      PFC.PBIORL.BIT.B7    = 0;     /* PB7(CRX0) input */
111      PFC.PBIORL.BIT.B6    = 1;     /* PB6(CTX0) output */
112  }
113
```

3. Sample Program Listing: main.c (3)

```
114    /*""FUNC COMMENT""*****************************************************
115     * Outline     : RCAN setting
116     *-------------------------------------------------------------------------
117     * Include     : #include "iodefine.h"
118     *-------------------------------------------------------------------------
119     * Declaration : void io_init_can(void);
120     *-------------------------------------------------------------------------
121     * Function    : Controller area network (RCAN) setting
122     *-------------------------------------------------------------------------
123     * Argument    : void
124     *-------------------------------------------------------------------------
125     * Return Value: void
126     *-------------------------------------------------------------------------
127     * Notice      : non
128     *""FUNC COMMENT END""*************************************************/
129    void io_init_can(void)
130    {
131        int i;
132        int j;
133
134        /* ==== Setting of power down mode(RCAN) ==== */
135        STB.CR3.BYTE = 0xf6;                /* Module Standby Clear */
136                                            /* RCAN */
137        /* ==== Initializing CAN module ==== */
138        RCANET.MCR.WORD |= 0x0001;          /* CAN Interface reset mode */
139        while((RCANET.IRR.WORD & CAN_IRR0) != CAN_IRR0){
140                /* Reset state waiting */
141        }
142        /* ==== IRR = 1, GSR = 1 (Auto SET) ==== */
143
144        /* ---- Clear IRR0 ---- */
145        RCANET.IRR.WORD = 0x0001;
146
147        /* ---- RCAN mode selection(MCR15) ---- */
148        RCANET.MCR.WORD |= 0x8000;          /* RCAN-ET is not same as HCAN2 */
149
150        /* ---- Disable all can interrupt ---- */
151        RCANET.IMR.WORD  = 0xffff;
152
153        /* ----All mailbox init ---- */
154        for(i = 0; i < 16; i++){
155                RCANET.MB[i].CTRL0.LONG = 0x00000000;
156                RCANET.MB[i].LAFM.LONG = 0x00000000;
157                for(j = 0; j < 8; j++){
158                        RCANET.MB[i].MSG_DATA[j] =  0x00;
159                }
160        }
161
162        /* ---- Config mailbox0 as reception slot ---- */
163        RCANET.MB[0].CTRL1.WORD = 0x0200;       /* can receive data and remote frame */
164        RCANET.MB[0].CTRL0.LONG = 0x00000000;   /* Initialize the Message CTRL Field */
165        RCANET.MB[0].LAFM.LONG = 0x00000000;
166        for(i = 0; i < 8; i++){                 /* data clear */
167                RCANET.MB[0].MSG_DATA[i] =  0x00;
168        }
```

4. Sample Program Listing: main.c (4)

```
169         /* ---- Config mailbox1 as transmission slot ---- */
170         RCANET.MB[1].CTRL1.WORD = 0x0002;         /* Can send data or remote frame, dlc=2 */
171         RCANET.MB[1].CTRL0.LONG = 0x00000000;     /* standard data frame, id=0x000 */
172         RCANET.MB[1].LAFM.LONG = 0x00000000;
173         for(i = 0; i < 8; i++){                   /* data clear */
174                 RCANET.MB[1].MSG_DATA[i] =  0x00;
175         }
176
177         /* ---- Config baudrate ---- */
178         RCANET.BCR1.WORD = 0x5200;        /* tsg1=6(7 bits),tsg2=3(4 bits),sjw=0(1bit),bsp=0 */
179         RCANET.BCR0.WORD = 0x0003;        /* 500 kbps */
180     //  RCANET.BCR0.WORD = 0x0007;        /* 250 kbps */
181     //  RCANET.BCR0.WORD = 0x000f;        /* 125 kbps */
182
183         /* ---- Clear interrupt flags ---- */
184         RCANET.IRR.WORD = 0xffff;
185
186         /* ---- Clear reset and halt ---- */
187         RCANET.MCR.WORD &= 0xf8fc;
188
189         while( (RCANET.GSR.WORD & CAN_GSR3) != 0x0000 ){
190                 /* reset state is end */
191         }
192     }
193
194     /*""FUNC COMMENT""*************************************************
195      * Outline     : Remote frame transmit
196      *-----------------------------------------------------------------
197      * Include     : #include "iodefine.h"
198      *-----------------------------------------------------------------
199      * Declaration : void io_remote_send(void);
200      *-----------------------------------------------------------------
201      * Function    : Transmits the remote frame by using RCANET
202      *-----------------------------------------------------------------
203      * Argument    : void
204      *-----------------------------------------------------------------
205      * Return Value: void
206      *-----------------------------------------------------------------
207      * Notice      : non
208      *""FUNC COMMENT END""*************************************************/
209     void io_remote_send(void)
210     {
211         /* ---- Transmission waiting ---- */
212         while((RCANET.TXPR10.LONG & CAN_MB01) == CAN_MB01){
213         }
214
215         /* ---- Transmission data set ---- */
216         RCANET.MB[1].CTRL1.WORD = 0x0002;     /* Can send data or remote frame, dlc=2 */
217         RCANET.MB[1].CTRL0.LONG = 0x40000000; /* standard remote frame, id=0x000 */
218
219         /* ---- Transmit the data ---- */
220         RCANET.TXPR10.LONG = CAN_MB01;
221
222         /* ---- Transmission completion waiting ---- */
223         while((RCANET.TXACK0.WORD & CAN_MB1) != CAN_MB1){
224         }
225
226         /* ---- Transmission completion flag clear ---- */
227         RCANET.TXACK0.WORD = CAN_MB1;
228     }
229
```

5. Sample Program Listing: main.c (5)

```
/*""FUNC COMMENT""*****************************************************
 * Outline     : Data frame receive
 *---------------------------------------------------------------------
 * Include     : #include "iodefine.h"
 *---------------------------------------------------------------------
 * Declaration : void io_data_receive(void);
 *---------------------------------------------------------------------
 * Function    : Receives the data frame by using RCANET
 *---------------------------------------------------------------------
 * Argument    : void
 *---------------------------------------------------------------------
 * Return Value: void
 *---------------------------------------------------------------------
 * Notice      : non
 *""FUNC COMMENT END""*************************************************/
void io_data_receive(void)
{
    int i;

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

    /* ---- Receive data storage ---- */
    nIDE = RCANET.MB[0].CTRL0.BIT.IDE;
    nRTR = RCANET.MB[0].CTRL0.BIT.RTR;
    nDLC = RCANET.MB[0].CTRL1.BIT.DLC;
    nSID = RCANET.MB[0].CTRL0.BIT.STDID;
    nEID = RCANET.MB[0].CTRL0.BIT.EXDID;
    if(nDLC > 8){
            nDLC = 8;
    }
    for(i = 0; i < nDLC; i++){
            gRcv_data[i] = RCANET.MB[0].MSG_DATA[i];
    }

  /* ---- Reception completion flag clear ---- */
  RCANET.RXPR0.WORD = CAN_MB0;
}

/* End of File */
```

4. Documents for Reference

- Software Manual
  SH-1/SH2/SH-DSP Software Manual (REJ09B0171)
  The most up-to-date version of this document is available on the Renesas Technology Website.
- Hardware Manual
  SH7137 Group Hardware Manual (REJ09B0402)
  The most up-to-date versions of the documents are available on the Renesas Technology Website.

Website and Support

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

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

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Nov.19.08	—	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.