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.

H8S Family

Using the HCAN (5): Remote Frame Transmission and Reception

Introduction

The Controller Area Network (HCAN) module is used to control the Controller Area Network (CAN), which provides a means for real-time communications in automobiles and industrial equipment systems.

This application note presents an example of communications operation using the H8S/2636's on-chip HCAN module and is offered to users for reference in the software and hardware design processes.

Although the operation of the sample application and programs provided in this application note has been confirmed, please verify operation in your environment before actually using them.

Target Device

H8S/2636

Contents

1. Specifications

In response to a remote frame, data transmission and reception take place between two H8S/2636 devices. The messages are in the standard format.

(1) Specifications common to the transmitter and receiver

- Channel 0 (HCAN0) is used
- Baud rate: 250 Kbps (in 20-MHz operation)
- Message identifier: H'555
- Messages are received if the masked identifier matches

(2) Specifications of the remote frame transmitter

- Uses mailbox 1 to transmit a remote frame
- Uses mailbox 0 to receive a data frame
- Requests eight bytes of data
- Uses the receive message interrupt (IRRI)
  (a) After transmitting a remote frame, waits for a receive message interrupt to occur.
  (b) The receive message interrupt routine clears the reception-complete flag and disables the receive message interrupt.
  (c) The receive message interrupt routine activates the DTC by the software-activation method to store the received data in on-chip RAM.
  (d) The DTC is used in block-transfer mode, activated by software, and transfers one block consisting of eight bytes.
- Uses the DTC transfer-end interrupt
- The DTC transfer-end interrupt routine disables DTC activation by software and places the HCAN in sleep mode

(3) Specifications of the remote frame receiver

- Uses mailbox 0 to receive the remote frame
- Uses mailbox 1 to transmit a data frame
- Polls the reception-complete flag and remote request flag
- Data for transmission are H'55, H'66, H'77, H'88, H'99, H'AA, H'BB and H'FF
- Polls the transmission-complete flag during transmission
- After confirming that the transmission-complete flag has been set, clears the flag as the final operation

2. Functional Descriptions of the Transmitter and Receiver

Tables 1 and 2 list the function assignments of the relevant pins and registers.

Table 1 Function Assignment for the HCAN Module

Pin Usage		Function
Pin	HTxD0	Used for message transmission by the HCAN module (pin 97)
	HRxD0	Used for message reception by the HCAN module (pin 98)

Relevant Registers		Function
Registers common to transmission and reception	MSTPCRC	Module stop control register C Takes HCAN0 out of the module stop mode.
	IRR	Interrupt register Indicates the states of individual interrupt sources.
	BCR	Bit configuration register Configures the baud-rate prescaler for CAN and sets up the bit-timing parameters.
	MBCR	Mailbox configuration register Configures mailboxes for transmission or reception.
	MCR	Master control register Controls the CAN interface.
	GSR	General status register Indicates the CAN bus states.
	MCx[n]	Message control registers (x = mailbox number)
	n = 1	Sets the data length for data frames and remote frames.
	n = 2 to 4	Reserved
	n = 5	Holds standard ID bits (STD_ID2 to STD_ID0), extended ID bits (EXD_ID17 and EXD_ID16), RTR (indicates data frame or remote frame), and IDE (indicates standard format or extended format).
	n = 6	Holds standard ID bits (STD_ID10 to STD_ID3)
	n = 7	Holds extended ID bits (EXD_ID7 to EXD_ID0)
	n = 8	Holds extended ID bits (EXD_ID15 to EXD_ID8)
	MDx[n]	Message data registers (x = mailbox number)
	n = 1 to 8	Hold CAN message data for transmission or received CAN message data.
Transmission-related registers	TXPR	Transmit wait register After a message for transmission has been stored in the mailbox, the corresponding bit in this register is set, indicating a transmission-wait state.
	TXACK	Transmit acknowledge register Each bit in this register indicates whether or not the message in the corresponding mailbox has been transmitted normally.
Reception-related registers	RXPR	Receive complete register Each bit in this register indicates that a message has been received normally in the corresponding mailbox.
	LAFMH, LAFML	Local acceptance filter mask H, L Identifier filter mask settings for the mailboxes configured for reception.

Relevant Registers		Function
Interrupt-related registers	MBIMR	Mailbox interrupt mask register Enables or disables interrupt requests for the individual mailboxes.
	IMR	Interrupt mask register Enables or disables interrupt requests by the IRR interrupt flag.
	IPRM	Interrupt priority register Sets the priority level for HCAN interrupts.
	SYSCR	System control register Sets the interrupt control mode.

Table 2 Function Assignment for the DTC

DTC-Related Registers	Function
MSTPCRA	Module stop control register A Takes the DTC out of module stop mode.
MRA, MRB	DTC mode register A, B Control the operating mode of the DTC.
SAR	DTC source address register Specifies the address of the source data area for DTC transfer.
DAR	DTC destination address register Specifies the address of the destination data area for DTC transfer.
CRA	DTC transfer count register A Specifies the number of data transfers by the DTC.
CRB	DTC transfer count register B In block transfer mode, specifies the block length.
DTCERA to DTCERG	DTC enable register Selects the interrupt source that activates the DTC.
DTVECR	DTC vector register Enables or disables activation of the DTC by software and sets the vector number for the software activation interrupt.

3. Flowchart for the Remote Frame Transmitter

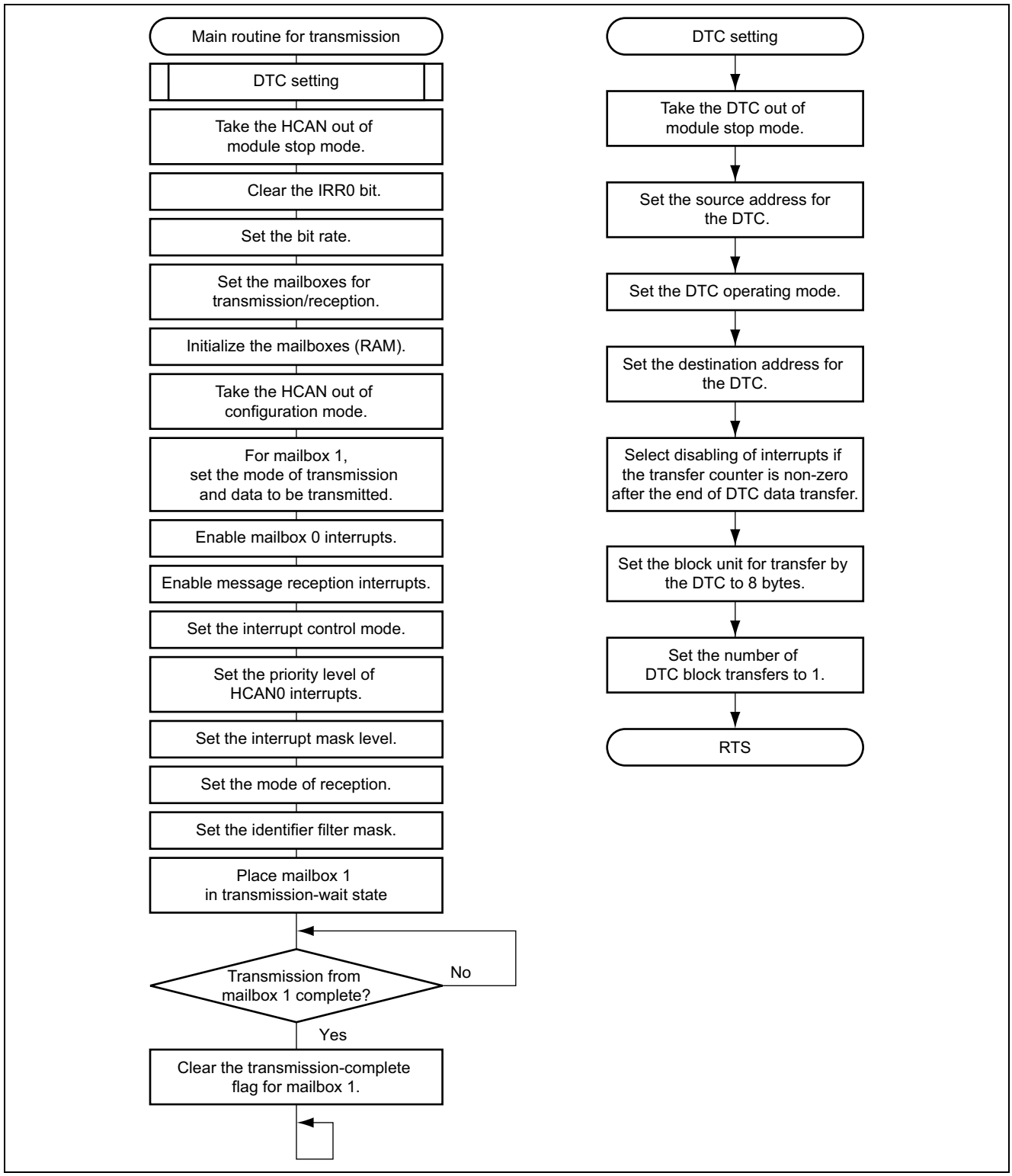

Figure 1 Flowchart for the Transmitter

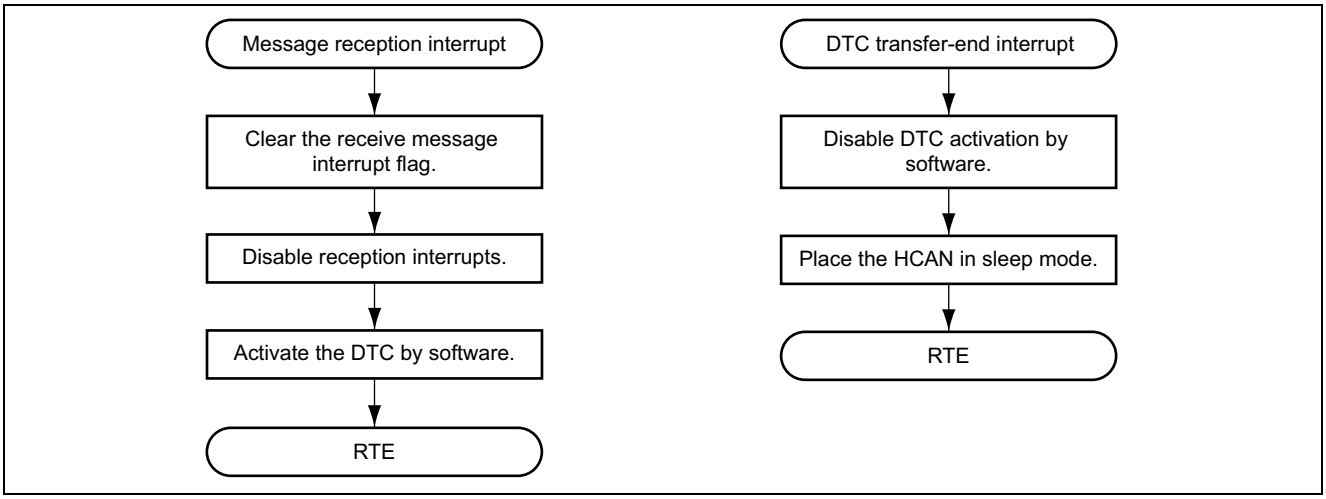

Figure 2 Flowchart of Interrupt Routines for the Transmitter

4. Description of Software (Transmitter)

4.1 Modules

Table 3 Description of Modules

Module	Label	Function
Main Routine	t_main	Initialize the HCAN and DTC and makes settings for remote frame transmission and data frame reception.
Reception interrupt routine	RM0	Clears the reception-complete, disables reception interrupts, and stores the received data by activating the DTC.
DTC transfer-end interrupt routine	SWDTEND	Disables activation of the DTC by software and places the HCAN in sleep mode.

4.2 Registers

Table 4 Description of Registers*

Register	Function	Setting	Used in
MAILBOX.MDATA[15][8]	Storage for the received data Address range: H'FFE000 to H'FFE0F7	—	Main routine
MSTP.CRA.BYTE	Takes the DTC out of module stop mode.	H'3F	
DTC_SAR	Sets the first address of the message data area for mailbox 0 as the source address for transfer.	H'FFF8B0	
DTC_MRA	Sets the incrementation of both DTC_SAR and DTC_DAR after each transfer, and selects block transfer mode and byte-sized transfer.	H'AA	
DTC_DAR	Sets the first address of the received-data storage area as the destination address for transfer.	H'FFE000	
DTC_MRB	Disables the interrupt to the CPU if the transfer counter value is non-zero after the end of DTC transfer.	H'00	
DTC_CRA	Sets the size of a block (8 bytes).	H'0808	
DTC_CRB	Sets the number of block transfers (one).	H'0001	
MSTP.CRC.BYTE	Takes HCAN0 out of module stop mode.	H'F7	
HCAN0.IRR.WORD	The reset interrupt flag in this register is cleared. (Clearing condition: writing a 1 to the bit)	H'0100	
HCAN0.BCR.WORD	Sets the bit rate to 250 Kbps when ϕ = 20 MHz	H'0334	
HCAN0.MBCR.WORD	Sets mailbox 1 for transmission and mailbox 0 for reception.	H'FDFF	

Register	Function	Setting	Used in
HCAN0.MCR.BYTE	Selects transmission in mailbox-number order and takes the HCAN module out of configuration mode.	H'04	Main routine
HCAN0.GSR.BYTE	Checked to confirm that HCAN0 is out of configuration mode.	—	
HCAN0.MC[1][4]	For mailbox 1, sets the frame type to remote frame and the frame format to standard format. Also holds the message identifier bits, STD_ID2 to STD_ID0 (for message ID = H'555).	H'B0	
HCAN0.MC[1][5]	Holds the message identifier bits, STD_ID10 to STD_ID3 (for message ID = H'555).	H'AA	
HCAN0.MC[1][0]	Sets the data length for transmission from mailbox 1 to eight bytes.	H'08	
HCAN0.MBIMR.WORD	Enables interrupt requests of mailbox 0.	H'FEFF	
HCAN0.IMR.WORD	Enables message reception and bus operation interrupts.	H'FCFF	
SYSCR.BYTE	Sets interrupt control mode 2.	H'20	
INTC.IPRM.BYTE	Sets the priority level of HCAN interrupts.	H'07	
HCAN0.MC[0][4]	For mailbox 0, sets the frame type to data frame and the frame format to standard format. Also holds the message identifier bits, STD_ID2 to STD_ID0 (for message ID = H'555).	H'A0	
HCAN0.MC[0][5]	Holds the message identifier bits, STD_ID10 to STD_ID3 (for message ID = H'555).	H'AA	
HCAN0.LAFMH.WORD	Mailbox 0 receives data if identifier bits match.	H'0000	
HCAN0.TXPR.WORD	Places mailbox 1 in the transmission-wait state.	H'0200	
HCAN0.TXACK.WORD	Checked to see if the transmission-complete flag for mailbox 1 is set; when set, the flag is cleared. (Clearing condition: writing a 1 to the bit)	H'0200	
HCAN0.RXPR.WORD	The reception-complete flag for mailbox 0 in this register is cleared. (Clearing condition: writing a 1 to the bit)	H'FFFF	Reception interrupt routine
HCAN0.IMR.WORD	Disables receive message interrupts.	H'FEFF	
DTC.DTVECR.BYTE	Sets the vector address for the DTC (H'0000045A) and enables DTC activation by software.	H'80	
DTC.DTVECR.BYTE	Disables DTC activation by software.	H'7F	DTC transfer-end interrupt routine
HCAN0.MCR.BYTE	Places the HCAN in sleep mode.	H'20	

Note: * The register names shown above are defined in a header file which is available for downloading from the following web page.
http://download.renesas.com/eng/mpumcu/sample_codes/h8sx_h8s_h8_family/io_register/index.html

5. Program Listing (Transmission)

```
/*****************************************************************************************/
/*  HCAN Transmission Program (No.5)                                                     */
/*****************************************************************************************/
#include <stdio.h>                          /* Header file for library functions         */
#include <machine.h>                        /* Header file for library functions         */
#include "2636S.h"                          /* Header file of peripheral register definitions   */

/*****************************************************************************************/
/* Definitions of Constants                                                              */
/*****************************************************************************************/
volatile struct MB{                         /* struct MAILBOX0-15                        */
    unsigned char MDATA[1][8];              /* Storage of received data                  */
};
#define MAILBOX     (*(volatile struct MB      *)0xFFE000) /* First address of           */
                                                        /*            received data storage */
#define DTC_SAR     (*(volatile unsigned long  *)0xFFEBC0) /* DTC register info setting  */
#define DTC_MRA     (*(volatile unsigned char  *)0xFFEBC0) /* DTC register info setting  */
#define DTC_DAR     (*(volatile unsigned long  *)0xFFEBC4) /* DTC register info setting  */
#define DTC_MRB     (*(volatile unsigned char  *)0xFFEBC4) /* DTC register info setting  */
#define DTC_CRA     (*(volatile unsigned short *)0xFFEBC8) /* DTC register info setting  */
#define DTC_CRB     (*(volatile unsigned short *)0xFFEBCA) /* DTC register info setting  */

void t_main(void){
    unsigned char i,j;
/* DTC initialization */
    MSTP.CRA.BYTE = 0x3F;                   /* Cancel module stop mode of DTC            */
    DTC_SAR = (long)(&HCAN0.MD[0][0]);      /* Set transfer source address               */
    DTC_MRA = 0xAA;                         /* SAR and DAR incremented after transfer;   */
                                            /*                          set block transfer mode  */
    DTC_DAR = (long)(&MAILBOX.MDATA[0][0]);/* Set transfer destination address (on-chip RAM)   */
    DTC_MRB = 0x00;                         /* Disable interrupt if transfer counter is non-zero*/
                                            /*                       after end of DTC transfer  */
    DTC_CRA = 0x0808;                       /* Block transfer size: 8 bytes              */
    DTC_CRB = 0x0001;                       /* Number of block transfers: 1              */
/* HCAN initialization */
    MSTP.CRC.BYTE = 0xF7;                   /* Cancel module stop mode of HCAN           */
    HCAN0.IRR.WORD = 0x0100;                /* Initialize reset flag for HCAN module     */
    HCAN0.BCR.WORD = 0x0334;                /* Bit rate: 250 kbps                        */
    HCAN0.MBCR.WORD = 0xFDFF;               /* Set mailbox 1 for transmission and        */
                                            /*                           mailbox 0 for reception */
    for(i=0; i<=15; i++){                   /* Initialize mailboxes (RAM)                */
        for(j=0; j<=7; j++){
            HCAN0.MC[i][j] = 0x00;
        }
    }
    for(i=0; i<=15; i++){                   /* Initialize mailboxes (RAM)                */
        for(j=0; j<=7; j++){
            HCAN0.MD[i][j] = 0x00;
        }
    }
    HCAN0.MCR.BYTE = 0x04;                  /* Transmission in mailbox No. order;        */
                                            /*                              cancel config. mode */
```

```
    while(HCAN0.GSR.BYTE & 0x08);          /* Configuration mode cancellation check            */
/* Transmit data settings */
    HCAN0.MC[1][4] = 0xB0;                 /* Standard format, remote frame, and               */
                                                                          identifier setting */
    HCAN0.MC[1][5] = 0xAA;                 /* Identifier setting                               */
    HCAN0.MC[1][0] = 0x08;                 /* Data length: 8 bytes                             */
/* Interrupt settings */
    HCAN0.MBIMR.WORD = 0xFEFF;             /* Enable mailbox 0 interrupt requests              */
    HCAN0.IMR.WORD = 0xFCFF;               /* Enable message reception interrupts              */
    SYSCR.BYTE |= 0x20;                    /* Set interrupt control mode 2                     */
    INTC.IPRM.BYTE = 0x07;                 /* Set the priority level of HCAN0 interrupts to 7  */
    set_imask_exr(0);                      /* Set interrupt request mask level                 */
/* Reception data settings */
    HCAN0.MC[0][4] = 0xA0;                 /* Standard format, data frame, and                 */
                                           /*                                  identifier setting */
    HCAN0.MC[0][5] = 0xAA;                 /* Identifier setting                               */
    HCAN0.LAFMH.WORD = 0x0000;             /* Mailbox 0 receives data if identifier bits match */
/* Message transmission */
    HCAN0.TXPR.WORD = 0x0200;              /* Place mailbox 1 in a transmission wait state     */
    while((HCAN0.TXACK.WORD & 0x0200) != 0x0200); /* Wait until transmission is complete       */
/* Transmission-complete flag clearing */
    HCAN0.TXACK.WORD &= 0x0200;            /* Clear transmission-complete flag                 */
    while(1);
}

/*****************************************************************************************/
/*  Message Reception Interrupt Routine                                                  */
/*****************************************************************************************/
#pragma interrupt(RM0)
void RM0(void){
    HCAN0.RXPR.WORD &= 0xFFFF;             /* Clear IRR1 (reception message interrupt flag)    */
    HCAN0.IMR.WORD = 0xFEFF;               /* Disable message reception interrupts             */
/* Settings for DTC activation by software */
    while(DTC.DTVECR.BYTE & 0x80);         /* Check if software-activated DTC transfer is      */
                                           /*                                        in progress */
    DTC.DTVECR.BYTE |= 0x08;               /* Set vector number (H'410)                        */
    while((DTC.DTVECR.BYTE & 0x7F) != 0x08){ /* Check the vector No. written to DTVECR         */
        DTC.DTVECR.BYTE |= 0x08;           /* Set vector number (H'410) again                  */
    }
    DTC.DTVECR.BYTE |= 0x80;               /* Enable activation of DTC by software             */
}

/*****************************************************************************************/
/*  DTC Transfer-End Interrupt Routine                                                   */
/*****************************************************************************************/
#pragma interrupt(SWDTEND)
void SWDTEND(void){
    DTC.DTVECR.BYTE &= 0x7F;               /* Disable activation of DTC by software            */
/* HCAN sleep mode setting */
    HCAN0.MCR.BYTE |= 0x20;                /* Put HCAN in sleep mode                           */
}
```

6. Flowchart for the Remote Frame Receiver

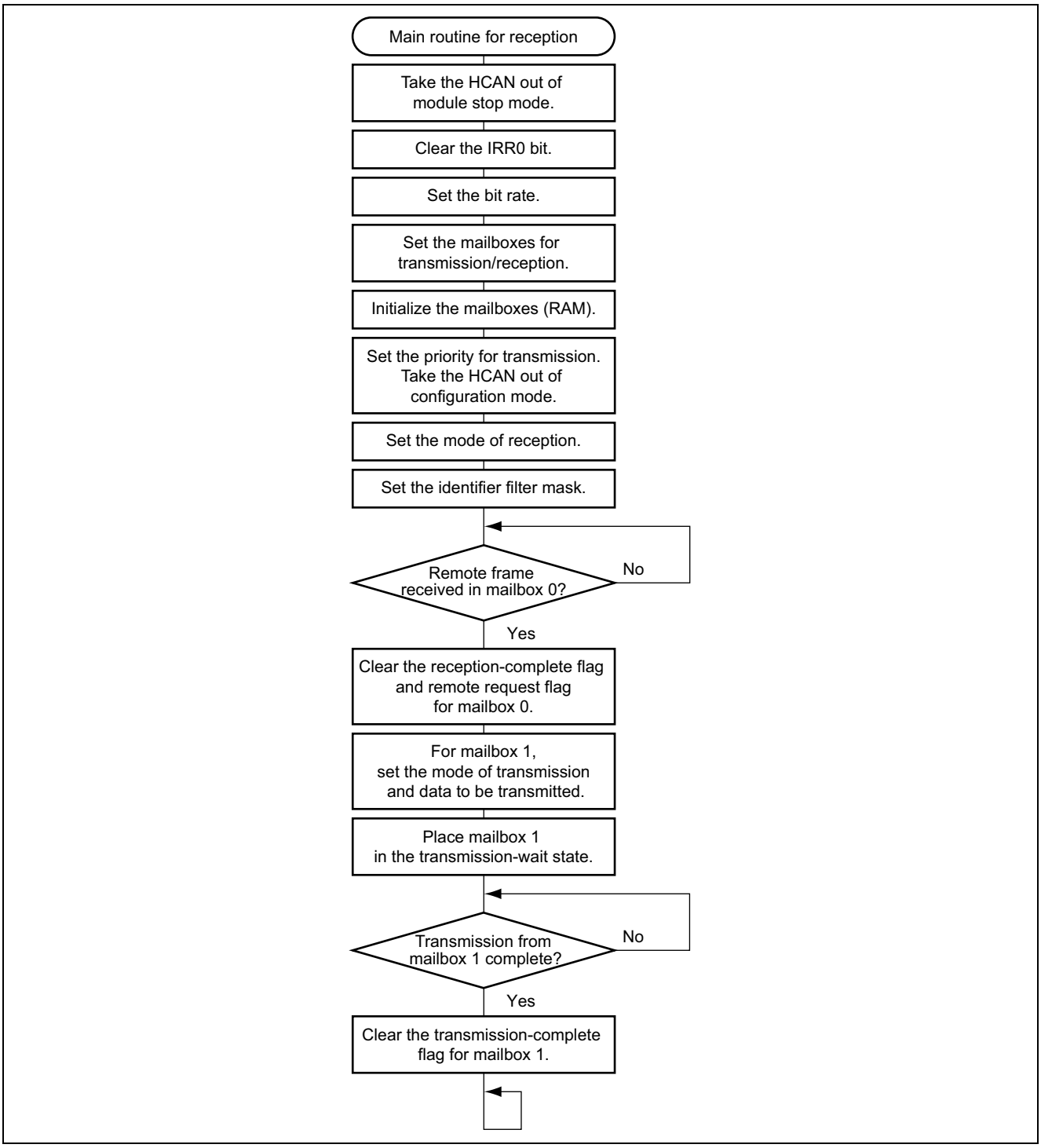

Figure 3 Flowchart for the Receiver

7. Description of Software (Receiver)

7.1 Module

Table 5 Description of Module

Module	Label	Function
Main Routine	r_main	Initializes the HCAN, receives a remote frame, and transmits a data frame.

7.2 Registers

Table 6 Description of Registers*

Register	Function	Setting	Used in
MSTP.CRC.BYTE	Takes HCAN0 out of module stop mode.	H'F7	Main routine
HCAN0.IRR.WORD	The reset interrupt flag in this register is cleared. (Clearing condition: writing a 1 to the bit)	H'0100	
HCAN0.BCR.WORD	Sets the bit rate to 250 Kbps when ϕ = 20 MHz	H'0334	
HCAN0.MBCR.WORD	Sets mailbox 0 for reception and mailbox 1 for transmission.	H'0100	
HCAN0.MCR.BYTE	Selects transmission in mailbox-number order and takes the HCAN module out of configuration mode.	H'04	
HCAN0.GSR.BYTE	Checked to confirm that HCAN0 is out of configuration mode.	—	
HCAN0.MC[0][4]	For mailbox 0, sets the frame type to remote frame and the frame format to standard format. Also holds the message identifier bits, STD_ID2 to STD_ID0 (for message ID = H'555).	H'B0	
HCAN0.MC[0][5]	Holds the message identifier bits, STD_ID10 to STD_ID3 (for message ID = H'555).	H'AA	
HCAN0.LAFMH.WORD	Mailbox 0 receives the message if the identifier bits match.	H'0000	
HCAN0.RXPR.WORD	The remote request flag for mailbox 0 in this register is cleared. (Clearing condition: writing a 1 to the bit)	H'0100	
HCAN0.RFPR.WORD	The reception-complete flag for mailbox 0 in this register is cleared. (Clearing condition: writing a 1 to the bit)	H'0100	

Register	Function	Setting	Used in
HCAN0.MC[1][4]	For mailbox 1, sets the frame type to data frame and the frame format to standard format. Also holds the message identifier bits, STD_ID2 to STD_ID0 (for message ID = H'555).	H'A0	Main routine
HCAN0.MC[1][5]	Holds the message identifier bits, STD_ID10 to STD_ID3 (for message ID = H'555).	H'AA	
HCAN0.MC[1][0]	Sets the data length for transmission from mailbox 1 to eight bytes.	H'08	
HCAN0.MD[1][0]	Holds the 1st byte for transmission from mailbox 1.	H'55	
HCAN0.MD[1][1]	Holds the 2nd byte for transmission from mailbox 1.	H'66	
HCAN0.MD[1][2]	Holds the 3rd byte for transmission from mailbox 1.	H'77	
HCAN0.MD[1][3]	Holds the 4th byte for transmission from mailbox 1.	H'88	
HCAN0.MD[1][4]	Holds the 5th byte for transmission from mailbox 1.	H'99	
HCAN0.MD[1][5]	Holds the 6th byte for transmission from mailbox 1.	H'AA	
HCAN0.MD[1][6]	Holds the 7th byte for transmission from mailbox 1.	H'BB	
HCAN0.MD[1][7]	Holds the 8th byte for transmission from mailbox 1.	H'FF	
HCAN0.TXPR.WORD	Places mailbox 1 in the transmission-wait state.	H'0200	
HCAN0.TXACK.WORD	Checked to see if the transmission-complete flag for mailbox 1 is set; when set, the flag is cleared. (Clearing condition: writing a 1 to the bit)	H'0200	

Note: * The register names shown above are defined in a header file which is available for downloading from the following web page.
http://download.renesas.com/eng/mpumcu/sample_codes/h8sx_h8s_h8_family/io_register/index.html

8. Program Listing (Reception)

```
/*****************************************************************************************/
/*  HCAN Reception Program (No.5)                                                        */
/*****************************************************************************************/
#include <stdio.h>                          /* Header file for library functions          */
#include <machine.h>                        /* Header file for library functions          */
#include "2636S.h"                          /* Header file of peripheral register definitions */

void r_main(void)
{
    unsigned char i,j;
    unsigned long lp;
/* Initialization */
    MSTP.CRC.BYTE = 0xF7;                   /* Cancel module stop mode of HCAN            */
    HCAN0.IRR.WORD = 0x0100;                /* Initialize reset flag for HCAN module      */
    HCAN0.BCR.WORD = 0x0334;                /* Bit rate: 250 kbps                         */
    HCAN0.MBCR.WORD = 0x0100;               /* Set mailbox 0 for reception and            */
                                            /*                       mailbox 1 for transmission */
    for(i=0; i<=15; i++){                   /* Initialize mailboxes (RAM)                 */
        for(j=0; j<=7; j++){
            HCAN0.MC[i][j] = 0x00;
        }
    }
    for(i=0; i<=15; i++){                   /* Initialize mailboxes (RAM)                 */
        for(j=0; j<=7; j++){
            HCAN0.MD[i][j] = 0x00;
        }
    }
    HCAN0.MCR.BYTE &= 0xFE;                 /* Cancel configuration mode                  */
    while(HCAN0.GSR.BYTE & 0x08);           /* Configuration mode cancellation check      */
/* Reception data setting */
    HCAN0.MC[0][4] = 0xB0;                  /* Standard format, remote frame, and         */
                                            /*                                 identifier setting */
    HCAN0.MC[0][5] = 0xAA;                  /* Identifier setting                         */
    HCAN0.LAFMH.WORD = 0x0000;              /* Mailbox 0 receives data if identifier bits match */
/* Wait for reception */
    while(((HCAN0.RXPR.WORD & 0x0100) != 0x0100) /* Wait until remote frame is received   */
        &&((HCAN0.RFPR.WORD & 0x0100) != 0x0100));
    HCAN0.RXPR.WORD &= 0x0100;              /* Clear reception-complete flag              */
    HCAN0.RFPR.WORD &= 0x0100;              /* Clear remote request flag                  */
/* Transmit data setting */
    HCAN0.MC[1][4] = 0xA0;                  /* Standard format, data frame, and           */
                                            /*                                 identifier setting */
    HCAN0.MC[1][5] = 0xAA;                  /* Identifier setting                         */
    HCAN0.MC[1][0] = 0x08;                  /* Data length: 8 bytes                       */
    HCAN0.MD[1][0] = 0x55;                  /* Message data: 01010101                     */
    HCAN0.MD[1][1] = 0x66;                  /* Message data: 01100110                     */
    HCAN0.MD[1][2] = 0x77;                  /* Message data: 01110111                     */
    HCAN0.MD[1][3] = 0x88;                  /* Message data: 10001000                     */
    HCAN0.MD[1][4] = 0x99;                  /* Message data: 10011001                     */
    HCAN0.MD[1][5] = 0xAA;                  /* Message data: 10101010                     */
    HCAN0.MD[1][6] = 0xBB;                  /* Message data: 10111011                     */
    HCAN0.MD[1][7] = 0xFF;                  /* Message data: 11111111                     */
```

```
/* Message transmission */
    HCAN0.TXPR.WORD = 0x0200;              /* Place mailbox 1 in a transmission wait state     */
    while((HCAN0.TXACK.WORD & 0x0200) != 0x0200);
/* Transmission-complete flag clearing */
    HCAN0.TXACK.WORD &= 0x0200;            /* Clear transmission-complete flag                 */
    while(1);
}
```

9. Waveforms during Operation (Transmission and Reception)

Figure 4 shows the waveforms seen during the execution of this application.

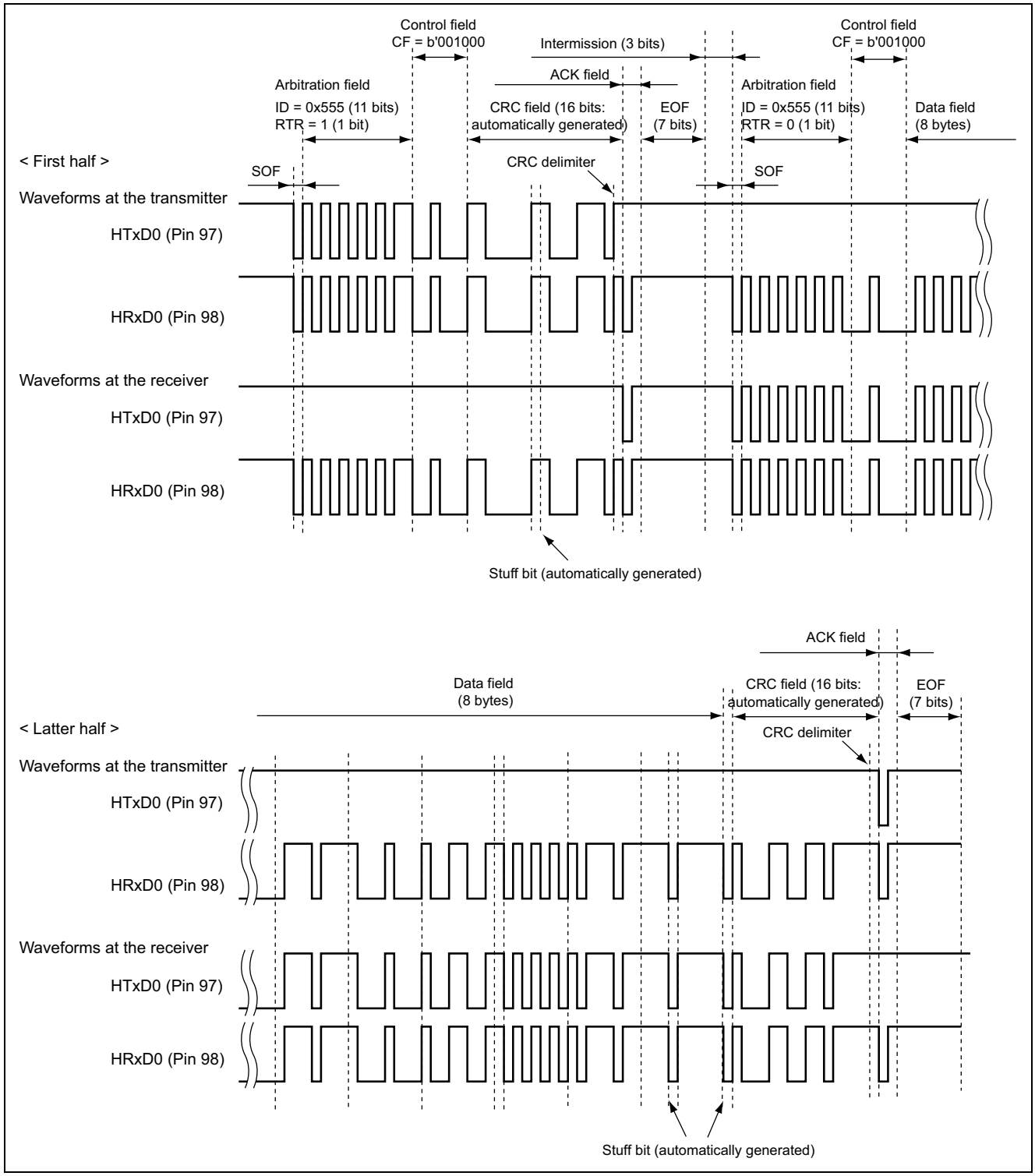

Figure 4 Waveforms during Operation

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Jul.22.05	—	First edition issued

Keep safety first in your circuit designs!

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

Notes regarding these materials

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