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.

32176 Group

Application of the CAN Module (Data Frame Reception)

1. Overview

The sample task described in this document uses the 32176 Group microcomputer's on-chip CAN (Controller Area Network) module.

2. Introduction

The sample task described in this document uses the following microcomputers, under the respective conditions.

- Microcomputer: 32176 Group (M32176FnVFP, M32176FnTFP)
- Operating Frequency: 20 to 40 MHz (The sample program is compiled assuming a frequency of 40 MHz.)
- Operating Board: Starter kit for 32176 Group

3. Explanation of the Technology Applied

3.1 Outline of the CAN Module

The 32176 includes a 2-channel Full CAN module which conforms to the CAN Specification V2.0B active. By using 16 message slots and three mask registers effectively, the load on the CPU during data processing can be reduced.

For details on CAN functions, refer to the 32176 Group User's Manual and the 32176 Group Outline of the CAN Module Application Note.

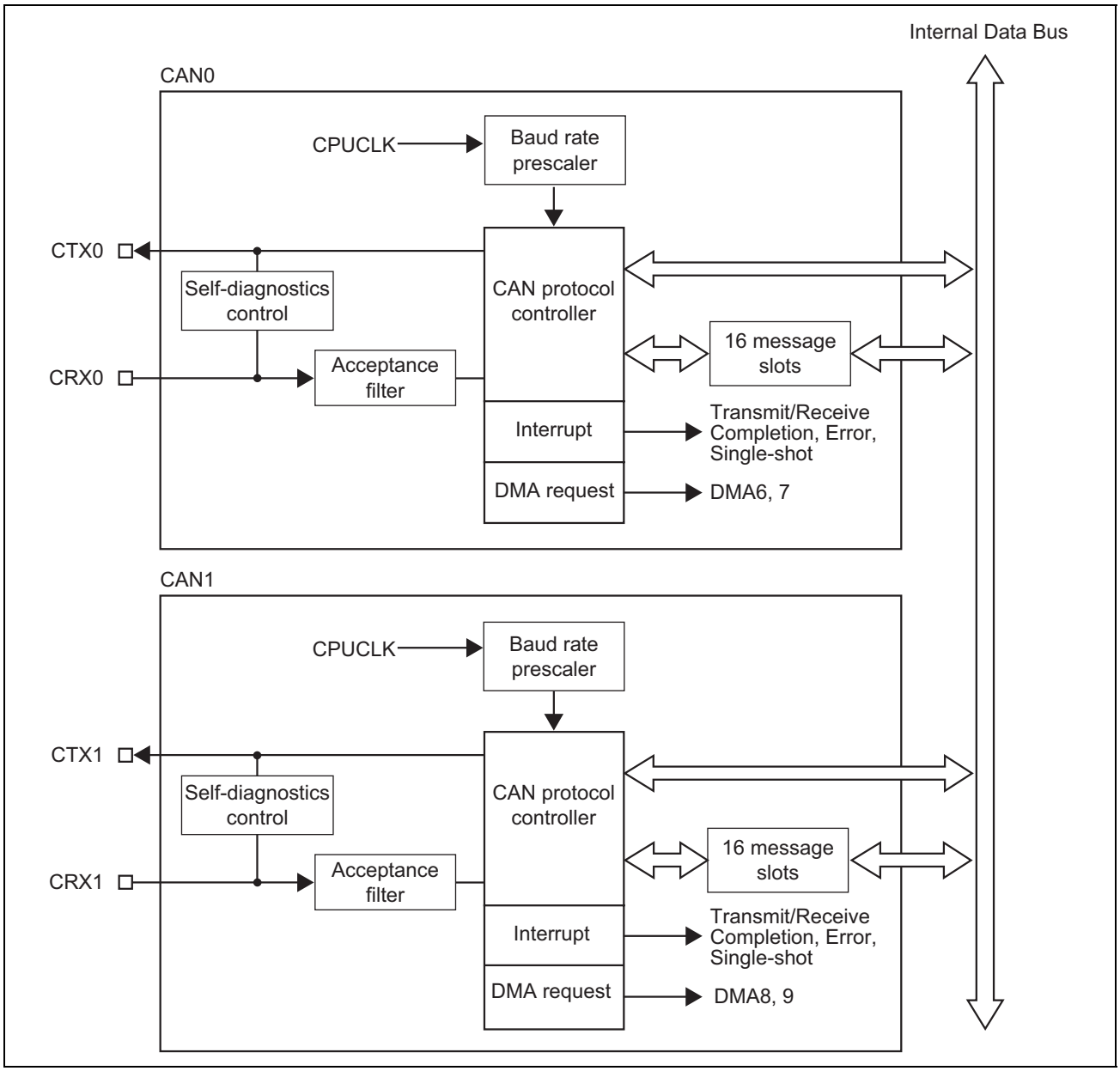

Figure 3.1.1 CAN Module Block Diagram

4. CAN Data Frame Reception Sample Program

4.1 Outline of the Sample Program

In this sample program, the CAN bus speed is at 125 kbps, and the standard format data frame of ID: 0, DLC: 2, DATA: H'C1C2 are received from the CAN0 slot 0.

After receiving Data Frame, data is read out on the RAM.

Interrupts and DMA Reception are not used.

4.2 Initial Setting Processing

Figure 4.2.1 shows the flowchart for the CAN module initial settings.

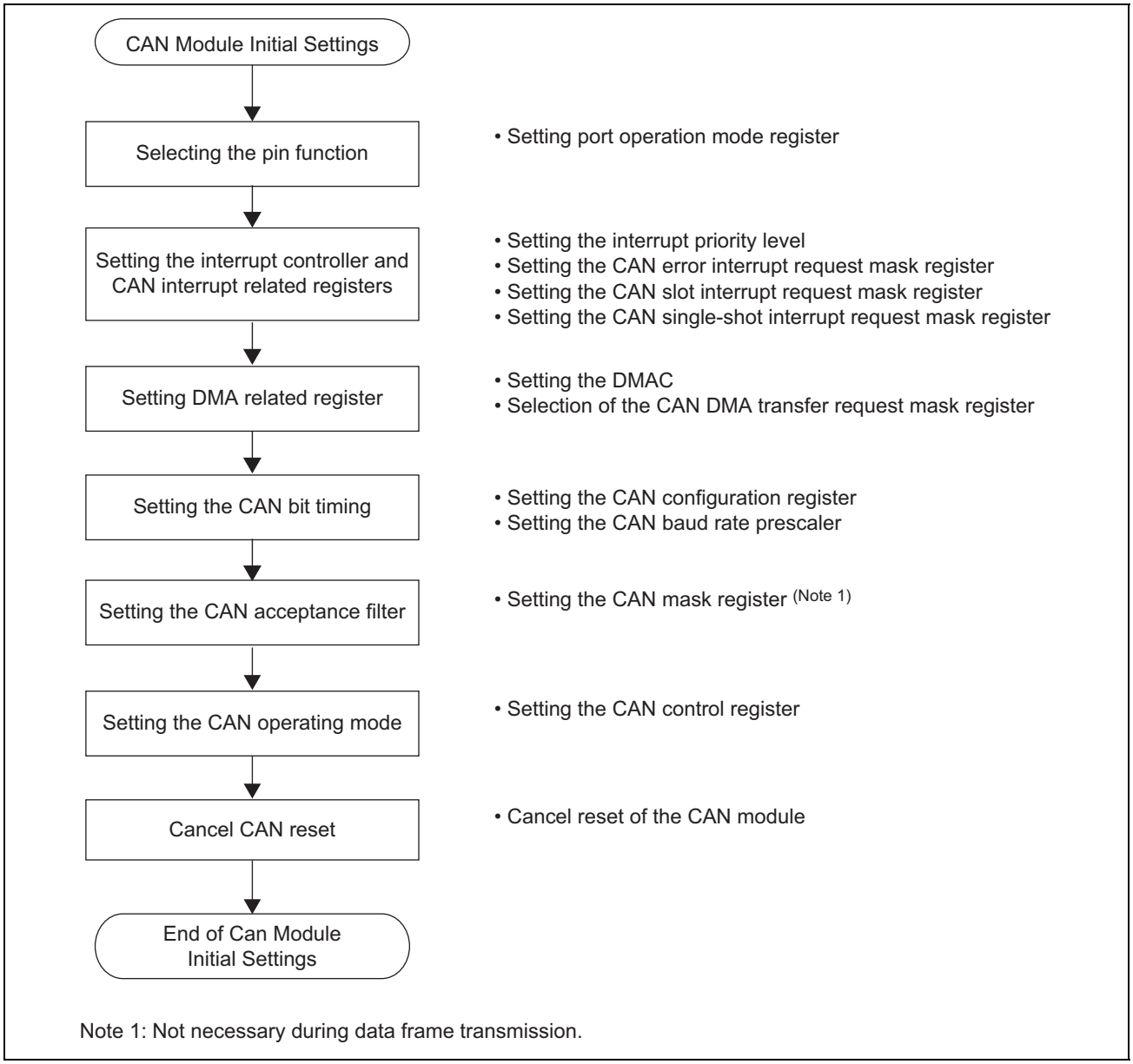

Figure 4.2.1 CAN Module Initial Setting

4.3 Reception Processing

Figure 4.3.1 shows the flowchart for data frame Reception processing.

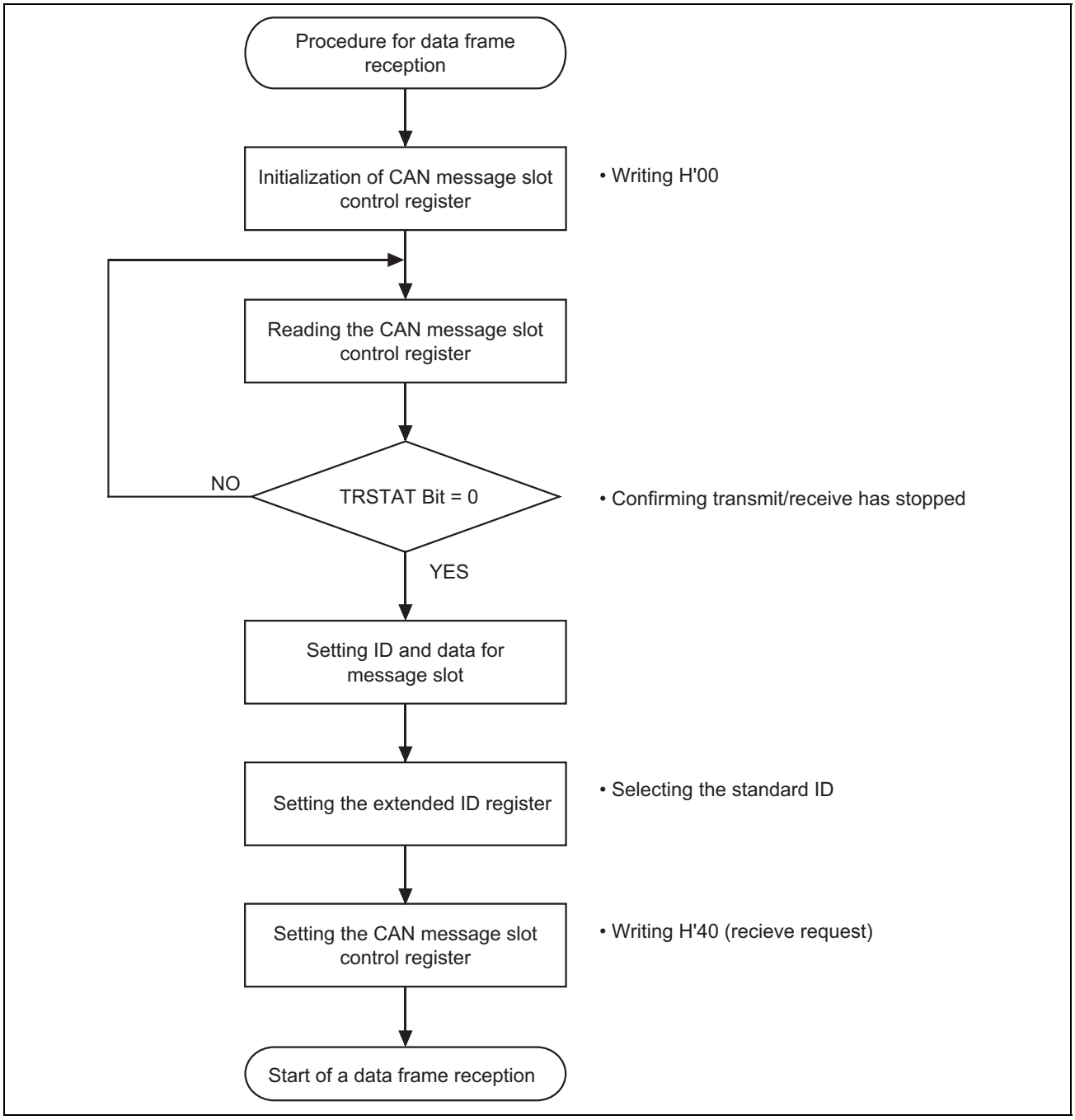

Figure 4.3.1 Data Frame Reception Processing

4.4 The State of CAN Message Slot Control Register

Figure 4.4.1 shows the state reception diagram of CAN message slot control registers during data frame Reception.

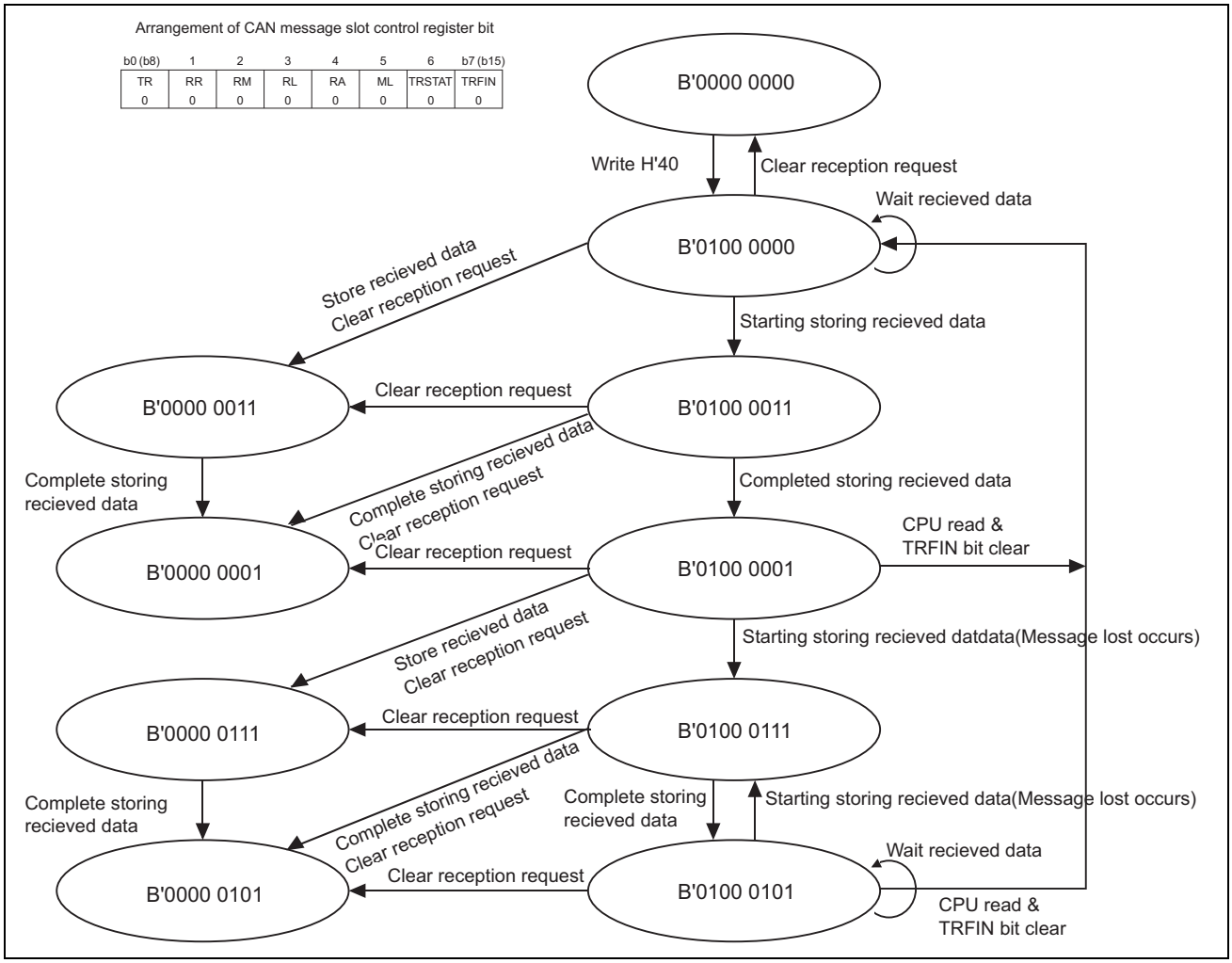

Figure 4.4.1 The State of CAN Message Slot Control Registers during Data Frame Reception

4.5 Interpretation of the Sample Program

Note: The registers used are indicated as (register name : bit name).

4.5.1 CAN Module Initialization Function (can_init ())

(1) Setting the pin function.

- Set the port P220 operation mode bit in the P22 operation mode register to "1" (CTX0). (P22MOD: P220MOD)

Note: When using CAN1, add the processing for setting port input enable bit of port input special function control register (PICNT: PIEN0) to "1" (input enabled).

(2) Setting the interrupt.

- Set the CAN0 transmit/receive & error interrupt control register to interrupts disabled.(ICAN0CR: ILEVEL)

(3) Setting the CAN0 interrupt-related registers.

- Clear the CAN0 slot interrupt request status register. (CAN0SLIST)
- Clear the CAN0 error interrupt request status register. (CAN0ERIST)
- Set the CAN0 slot interrupt request mask register to interrupt request disabled. (CAN0SLIMK)
- Set the CAN0 error interrupt request mask register to CAN bus error interrupt disabled, error passive interrupt disabled and bus off interrupt disabled. (CAN0ERIMK: EIM, PIM, OIM)

(4) Setting the CAN0 configuration register. (CAN0CONF: SJW, PH2, PH1, PRB, SAM)

- Set the propagation segment (PRB) to 5 Tq.
- Set phase segment 1 (PH1) to 7 Tq.
- Set phase segment 2 (PH2) to 7 Tq.
- Set the reSynchronization Jump Width (SJW: resynchronization width) to 1Tq.
- Set the number of samplings to once.

In the above settings the number of Tq within 1 bit is 20 and the sampling point is 65%.
Figure 4.5.1 shows the bit timing.

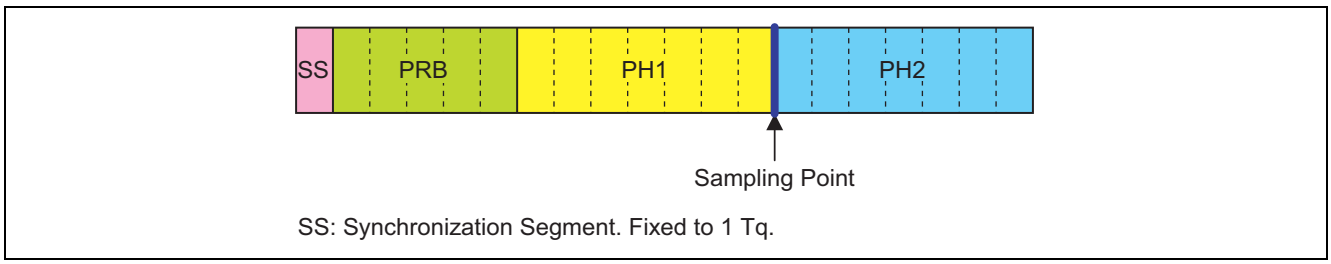

Figure 4.5.1 Bit Timing

(5) Setting the CAN0 baud rate prescaler (CAN0BRP)

- Set the baud rate prescaler to "15" (40 MHz / (125 kbps × 20 Tq) – 1 = 15)
  The formula for calculating the setup value in the baud rate prescaler (BRP) is given below

$$\text{BRP setup value} = \frac{\text{CPUCLK}}{\text{Baud rate (bps)} \times \text{the number of Tq within 1 bit}} - 1$$

(6) Setting the CAN0 acceptance filter.

ID check by acceptance filter is not performed when the CAN module acts as a Reception node. However, the following registers are all set to perform ID checks because this sample task conducts CAN transmit/receive processing.

- CAN0 Global Mask Register Standard ID. (C0GMSKS0, C0GMSKS1)
- CAN0 Global Mask Register Extended ID. (C0GMSKE0, C0GMSKE1, C0GMSKE2)
- CAN0 Local Mask Register A Standard ID. (C0LMSKAS0, C0LMSKAS1)
- CAN0 Local Mask Register A Extended ID. (C0LMSKAE0, C0LMSKAE1, C0LMSKAE2)
- CAN0 Local Mask Register B Standard ID. (C0LMSKBS0, C0LMSKBS1)
- CAN0 Local Mask Register B Extended ID. (C0LMSKBE0, C0LMSKBE1, C0LMSKBE2)

(7) Setting the CAN0 extended ID register.

- Set all slots to the standard ID format. (CAN0EXTID)

(8) Setting the CAN0 control register. (CAN0CNT: TSP, FRST, BCM, LBM, RST)

- Select the CAN bus bit clock in the timestamp prescaler.
- Cancel forcible reset.
- Set the BasicCAN function to disabled.
- Set the loopback function to disabled.
- Cancel CAN reset.

4.5.2 Main Function (main ())

(1) Calling the CAN module initialization function.
(2) Calling the data frame receive processing function.
(3) Confirm completing data frame reception

- Confirm CAN0 slot 0 interrupt request bit (CAN0SLIST: SSB0)

(4) Call data read processing function

4.5.3 The Data Frame Receive Processing Function (data_receive())

(1) Initialization of the CAN0 message slot 0 control register.

- Clear all flags and stop transmitting/receiving. (C0MSL0CNT)

(2) Confirm the transmit/receive operation stopped.

- Confirm transmit/receive status bit is "0". (C0MSL0CNT: TRSTAT)

(3) Creating data to be transmitted from slot 0.

- Set the ID to “0”. (C0MSL0SID0, C0MSL0SID1)

(4) Setting the CAN0 extended ID register.

- Set all the slots to standard ID format. (CAN0EXTID)

(5) Setting the CAN0 message slot 0 control register.

- Set the data frame recieve request. (C0MSL0CNT: RR)

4.5.4 The Data Read Processing Function (read_data ())

(1) Clear complete transmit/receive (C0MSL0CNT: TRFIN)
(2) Read out received data. (C0MSL0DT0 to C0MSL0DT7)
(3) Confirm complete transmit/receive (C0MSL0CNT: TRFIN)

When complete transmit/receive bit is set, it means new received data is stored during reading out received data. In this case read out data includes infinite value, so that start again from clearing complete transmit/receive bit.

4.6 Sample Program

The sample program for the CAN0 data frame Reception is shown below.

Note that the sample program below requires the SFR definition file. The latest SFR definition file can be downloaded from Renesas Technology website. When using the SFR definitions file, adjust the path setting to match the operating computer environment.

4.6.1 init.c

```
/*""FILE COMMENT""*****************************************************
 *      M32R C Programming              Rev. 1.01
 *              < Sample Program for 32176 >
 *              < CAN init >
 *
 *      Copyright (c) 2004 Renesas Technology Corporation
 *                         All Rights Reserved
 **********************************************************************/

/**********************************************************************/
/*              Include file                                          */
/**********************************************************************/

#include                "..\inc\sfr32176_pragma.h"

/**********************************************************************/
/*              Function prototype declaration                        */
/**********************************************************************/

        void            can_init(void);

/*""FUNC COMMENT""*****************************************************
 * Function name: void can_init(void)
 *---------------------------------------------------------------------
 * Description  : Initializes CAN module
 *---------------------------------------------------------------------
 * Argument     : -
 *---------------------------------------------------------------------
 * Returns      : -
 *---------------------------------------------------------------------
 * Notes        : -
 *""FUNC COMMENT END""************************************************/
void can_init(void)
{
        /* Setting input/output port operation mode register (CRX pin does not need to be set) */
        P22MOD |= 0x80u;                        /* P220 used as CTX */

        /* To use CAN1, set it up here */
        /*
         * - P7MOD  &= ~0x03u
         * - P7SMOD |=  0x03u
         * - P7MOD  |=  0x03u
         * - PICNT  |=  0x01u
         */

        /* Setting interrupt controller */
        ICAN0CR = 0x07;                         /* CAN0 interrupt priority level 7 (interrupt
disabled) */


        /* Setting CAN0 related interrupt mask register */
        CAN0SLIST = 0x0000;                     /* Clear CAN0 slot transmit/receive-finished interrupt
request */
        CAN0ERIST = 0x00;                       /* Clear CAN0 error interrupt request */
        CAN0SLIMK = 0x0000;                     /* Disable CAN0 slot transmit/receive-finished
interrupt */
        CAN0ERIMK = 0x00;                       /* Disable CAN0 error interrupt */


        /* Setting CAN configuration register */
        CAN0CONF = 0x3680;                      /* SJW=1, Sync(1)+Prop(5)+PH1(7)+PH2(7), sampling
count = 1 */
        CAN0BRP = (16 - 1);                     /* Baud rate: 40 MHz / divided by 16 / 20 Tq -> 125
Kbps */

```

```
   61          /* Setting ID mask register */
   62          C0GMSKS0 = 0xff;                        /* Global mask register */
   63          C0GMSKS1 = 0xff;
   64          C0GMSKE0 = 0xff;
   65          C0GMSKE1 = 0xff;
   66          C0GMSKE2 = 0xff;
   67          C0LMSKAS0 = 0xff;                       /* Local mask register A */
   68          C0LMSKAS1 = 0xff;
   69          C0LMSKAE0 = 0xff;
   70          C0LMSKAE1 = 0xff;
   71          C0LMSKAE2 = 0xff;
   72          C0LMSKBS0 = 0xff;                       /* Local mask register B */
   73          C0LMSKBS1 = 0xff;
   74          C0LMSKBE0 = 0xff;
   75          C0LMSKBE1 = 0xff;
   76          C0LMSKBE2 = 0xff;
   77
   78          /* To use in BasicCAN mode, set it up here. */
   79          /*
   80           *      - Set IDE14/15 of CAN0EXTID
   81           *      - Set ID of slots 14/15
   82           *      - Set local mask registers A/B
   83           *      - Set slots 14/15 for data frame reception
   84           */
   85
   86          /* Setting CAN operation mode */
   87          CAN0EXTID = 0x0000;                     /* Select standard format frame */
   88
   89          /* Negating CAN reset */
   90          CAN0CNT = 0x0000;                       /* Clear FRST and RST bits and disable BasicCAN
function */
   91                                                  /* Disable loopback function and select timestamp
divide-by-1 */
   92  }
```

4.6.2 data_recieve.c

```
/*""FILE COMMENT""*******************************************************
 *      M32R C Programming              Rev. 1.01
 *              < Sample Program for 32176 >
 *              < CAN data_receive >
 *
 *      Copyright (c) 2004 Renesas Technology Corporation
 *                         All Rights Reserved
 ***********************************************************************/

/************************************************************************/
/*              Include file                                            */
/************************************************************************/

#include                "..\inc\sfr32176_pragma.h"

/************************************************************************/
/*              Function prototype declaration                          */
/************************************************************************/

        void            main(void);
        void            data_receive(void);
        void            read_data(void);
extern  void            can_init(void);

/************************************************************************/
/*              Externally referenced variable                          */
/************************************************************************/

        UCHAR           CAN_DATA[8];

/*""FUNC COMMENT""*******************************************************
 * Function name: void main(void)
 *-----------------------------------------------------------------------
 * Description  : Data frame transmission sample program
 *-----------------------------------------------------------------------
 * Argument     : -
 *-----------------------------------------------------------------------
 * Returns      : -
 *-----------------------------------------------------------------------
 * Notes        : -
 *""FUNC COMMENT END""***************************************************/
void main(void)
{
        /* Initializing CAN module */
        can_init();

        /* CAN module operation */
        data_receive();

        /* Wait until data is received in slot 0 */
        while( ( C0NSLIST & SSB0) == 0x0000u ){
                ;
        }

        /* Read out received data */
        read_data();

        /* End of program */
        while( 1 ){
                ;
        }
}

/*""FUNC COMMENT""*******************************************************
 * Function name: void data_receive(void)
 *-----------------------------------------------------------------------
 * Description  : Receives data frame in slot 0
 *-----------------------------------------------------------------------
 * Argument     : -
 *-----------------------------------------------------------------------
 * Returns      : -
 *-----------------------------------------------------------------------
 * Notes        : -
 *""FUNC COMMENT END""***************************************************/
```

```
 75  void data_receive(void)
 76  {
 77          C0MSL0CNT = 0x00;                               /* Initialize CAN message slot control
register */
 78          while ( ( C0MSL0CNT & TRSTAT) != 0u){           /* Verify that transmit operation is idle */
 79                  ;
 80          }
 81
 82          /* Set ID in message slot 0 */
 83          C0MSL0SID0 = 0x00;                              /* ID : 0 */
 84          C0MSL0SID1 = 0x00;
 85
 86          /* Set extended ID register */
 87          CAN0EXTID = 0x0000;                             /* Select standard format */
 88
 89          /* Set CAN message slot control register */
 90          C0MSL0CNT = 0x40;                               /* Request reception of data frame */
 91
 92          /*
 93           * Because ID mask register is set within can_init() for "all bits to be checked,"
 94           * slot 0 receives only data frames in standard format with ID:0.
 95           */
 96  }
 97
 98  /*""FUNC COMMENT""*******************************************************
 99   * Function name: void data_send(void)
100   *-----------------------------------------------------------------------
101   * Description  : Transmits data frame from slot 0
102   *-----------------------------------------------------------------------
103   * Argument     : -
104   *-----------------------------------------------------------------------
105   * Returns      : -
106   *-----------------------------------------------------------------------
107   * Notes        : -
108   *""FUNC COMMENT END""*************************************************/
109  void read_data(void)
110  {
111          do {
112                  C0MSL0CNT = 0x4E;                       /* Clear TRFIN bit */
113
114                  /* Read out message slot 0 */
115                  CAN_DATA[0] = C0MSL0DT0;
116                  CAN_DATA[1] = C0MSL0DT1;
117                  CAN_DATA[2] = C0MSL0DT2;
118                  CAN_DATA[3] = C0MSL0DT3;
119                  CAN_DATA[4] = C0MSL0DT4;
120                  CAN_DATA[5] = C0MSL0DT5;
121                  CAN_DATA[6] = C0MSL0DT6;
122                  CAN_DATA[7] = C0MSL0DT7;
123
124          } while( ( C0MSL0CNT & TRFIN) != 0u );          /* Redo if TRFIN bit is set */
125
126          /* If necessary, check ML bit to see if message is lost
127           *
128           * if( (*C0MSL0CNT & ML) != 0)
129           *      (Processing for messages lost);
130           */
131  }
```

5. Reference Documents

- 32176 Group User's Manual (Rev.1.01)
- 32176 Group Outline of CAN Module (Rev.1.00)
- M32R Family Software Manual (Rev.1.20)
- M3T-CC32R V.4.30 User's Manual (Compiler)
- M3T-CC32R V.4.30 User's Manual (Assembler)

(Please get the latest one from Renesas Technology Corp. website.)

6. Website and Support Center

- Renesas Technology Corp. website
  http://www.renesas.com/

- Inquires for all Renesas products and technical inquiries for the M32R Family products:
  Customer Support Center: csc@renesas.com

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Dec.09.05	—	First edition issued

Keep safety first in your circuit designs!

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

Notes regarding these materials

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