SH7216 Group

Configuration to Receive Data Frames Using the Controller Area Network and Direct Memory Access Controller

REJ06B0975-0100
Rev. 1.00
Jun 04, 2010

Summary

This application note describes the configuration example of the SH7216 microcomputers (MCUs) to receive data frames using the Controller Area Network, and to store data frames in on-chip RAM using the Direct Memory Access Controller.

Target Device

SH7216 MCU

Contents

1. Introduction

1.1 Specifications

This application activates the SH7216 Direct Memory Access Controller and stores the data in on-chip RAM every time the SH7216 Controller Area Network receives a data frame.

1.2 Modules Used

- Controller Area Network module
  - Transmission speed: 1 Mbps
  - Receive mailbox: Mailbox 0
  - Mailbox 0 setting: Identifier: H'00A, standard format

- Direct Memory Access Controller
  - Channel number to use: Channel 4
  - Bus mode: Cycle steal mode
  - Transfer data length: In words
  - Source to activate: Controller Area Network mailbox 0 data frame received interrupt
  - Reload function: Use
  - Transfer destination: Controller Area Network mailbox 0
  - Transfer source: On-chip RAM (address: H'FFF8 4000 to H'FFF8 4011)

1.3 Applicable Conditions

MCU	SH7216
Operating Frequencies	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz
Integrated Development Environment	Renesas Electronics Corporation High-performance Embedded Workshop Ver.4.05.01
C Compiler	Renesas Electronics SuperH RISC engine Family C/C++ Compiler Package Ver.9.03 Release 00
Compiler Options	-cpu=sh2afpu -fpu=single -include="$(WORKSPDIR)\inc" -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo

1.4 Related Application Notes

For more information, refer to the following application note:

- SH7216 Group Example of Initialization

2. Applications

This application receives a data frame by the Controller Area Network, and stores the data in on-chip RAM by the Direct Memory Access Controller.

2.1 Overview of Modules

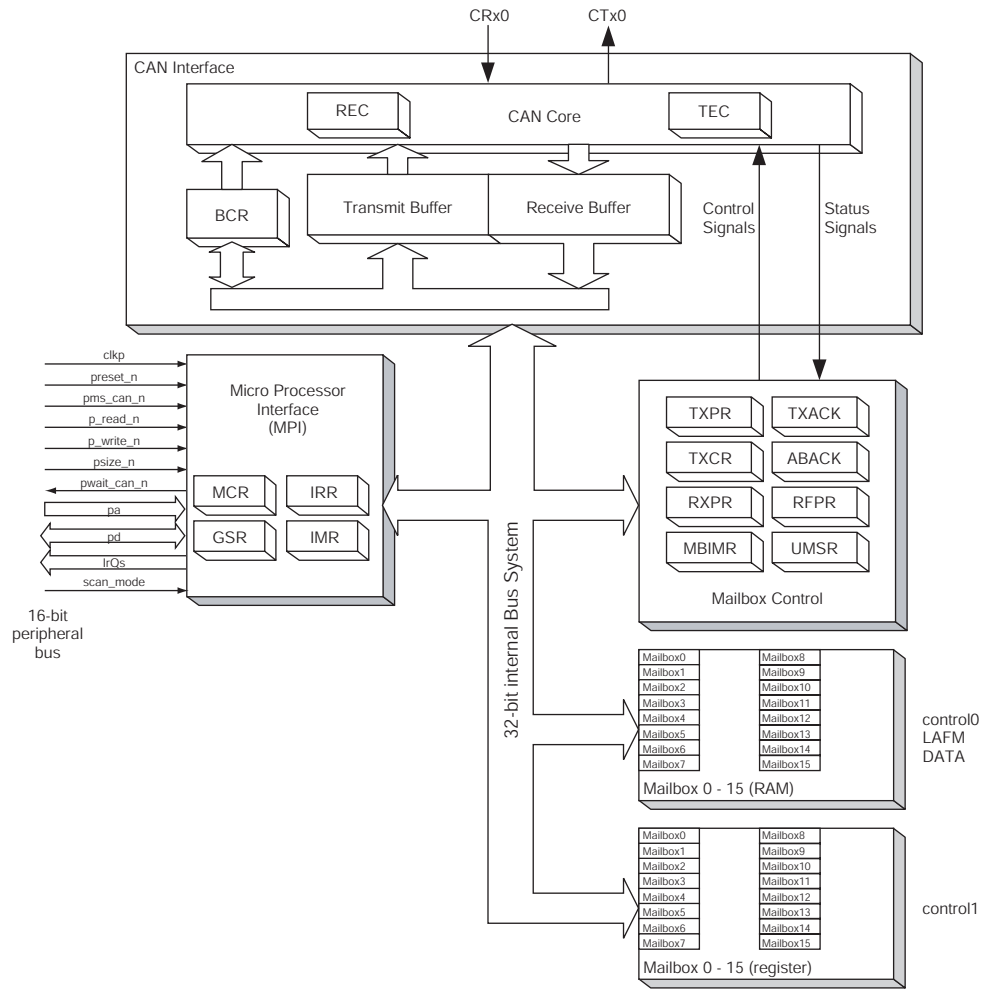
(1) Controller Area Network

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

The Controller Area Network module has 15 programmable mailboxes for transmission/reception, one mailbox for reception, and one programmable receive filtering mask to provide flexible communication procedure. Table 1 lists the features of the Controller Area Network. Figure 1 shows its block diagram. Table 2 lists interrupt sources. Sources to activate the Direct Memory Access Controller are mailbox 0 data frame received interrupt, and remote frame received frame only. For more information, refer to the Controller Area Network chapter in the SH7216 Group Hardware Manual.

Table 1 Controller Area Network Features

Item	Description
Protocol	CAN protocol, version 2.0B. Bit timing is compliant to ISO 11898
Number of mailboxes	16 (15 programmable transmit/receive mailbox, and one receive mailbox)
Transfer speed	Up to 1 Mbps
Number of interrupt sources	12
Test mode	Includes listen-only mode, and error passive mode



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

Figure 1 Controller Area Network Block Diagram

Table 2 Controller Area Network Interrupt Sources

Interrupt	Source	Interrupt flag	Activating the Data Transfer Controller/Direct Memory Access Controller
ERS_0	Error passive	IRR5	Not allowed
	Bus off/Bus off recovery	IRR6	
	Error warning (TEC ≥ 96)	IRR3	
	Error warning (REC ≥ 96)	IRR4	
OVR_0	Message error detection	IRR13 (1)	
	Transition to Reset/halt/CAN sleep	IRR0	
	Overload frame transmission	IRR7	
	Unread message overwrite (overrun)	IRR9	
	Detection of CAN bus operation in CAN sleep mode	IRR1	
RM0_0 (2) RM1_0 (2)	Data frame reception	IRR1 (3)	Allowed (4)
	Remote frame reception	IRR2 (3)	
SLE_0	Message transmission/transmission disabled	IRR8	Not allowed

Notes: 1. Available only in test mode.
2. RM0_0 is an interrupt generated by the Remote frame pending flag for mailbox 0 (RFPR0 [0]) on the Data frame pending flag for mailbox 0 (RXPR0 [0]). RM1_0 is an interrupt generated by the Remote frame pending flag for mailbox n (RFPR0 [n]) or the Data frame pending flag for mailbox n (RXPR0 [n]) (n = 1 to 15).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 15, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 15.
4. The Data Transfer Controller and the Direct Memory Access Controller can be activated only by the RM0_0 interrupt.

(2) Direct Memory Access Controller

The Direct Memory Access Controller transfers data among an external device with DACK (transfer request acknowledge signal), an external memory, on-chip memory, memory-mapped external device, and on-chip peripheral modules, instead of the CPU. It has two bus modes; cycle steal mode and burst mode.

In cycle steal mode, the Direct Memory Access Controller leaves the bus to other masters when it finishes "a transmit" (in bytes, words, long words, or 16 bytes). When the Direct Memory Access Controller receives another transfer request, it retrieves the bus again. Then, it transfers data in unit of a transfer, and leaves the bus again to the other bus. The Direct Memory Access Controller repeats this operation until the transfer end conditions are satisfied.

This application transfers the data received in the Controller Area Network mailbox 0 to the on-chip RAM using cycle steal mode.

Table 3 lists the features of the Direct Memory Access Controller. Figure 2 shows the block diagram of the Direct Memory Access Controller. For more information, refer to the Direct Memory Access Controller chapter in the SH7216 Group Hardware Manual.

Table 3 Direct Memory Access Controller Features

Item	Description
Number of channels	8
Address space	4 GB physically
Transfer data length	Byte, word, long word, and 16 bytes
Number of transfers	16,777,216 (24-bit) times
Address mode	Single address mode, dual address mode
Transfer request	External request, on-chip peripheral module request, auto-request
Bus mode	Cycle steal mode (normal mode and intermittent mode) Burst mode
Interrupt source	One-half of the data transfer completed, data transfer completed
Reload function	DMA transfer using the same setting as the current DMA transfer can be repeated automatically without specifying the setting again. Specify the reload register during the DMA transfer to execute the next DMA transfer with another setting. The reload function can be enabled or disabled per channel.

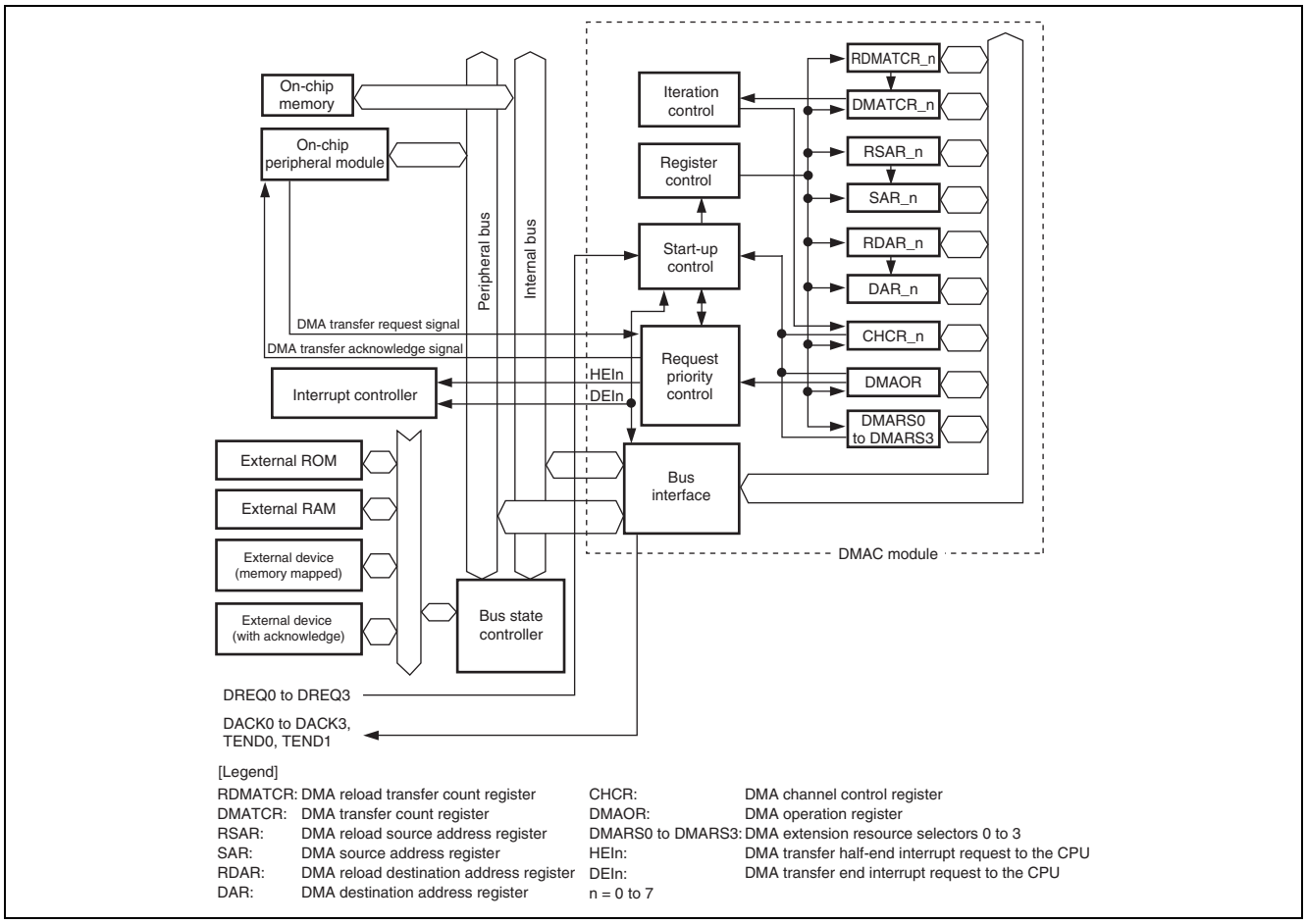


Figure 2 Direct Memory Access Controller Block Diagram

2.2 Configuration Procedure

(1) Steps to configure the Controller Area Network.
Configure the Controller Area Network in reset mode (configuration mode). After configuration is completed, clear the reset mode to join the CAN bus activity. To activate the Direct Memory Access Controller by the Controller Area Network data frame received interrupt (RM0_0), set bit 1 in the Interrupt mask register, and bit 0 in the Mailbox interrupt mask register 0 to enable interrupts.

Figure 3 and Figure 4 show the flow charts for configuring the Controller Area Network. For details on register settings, refer to the SH7216 Group Hardware Manual.

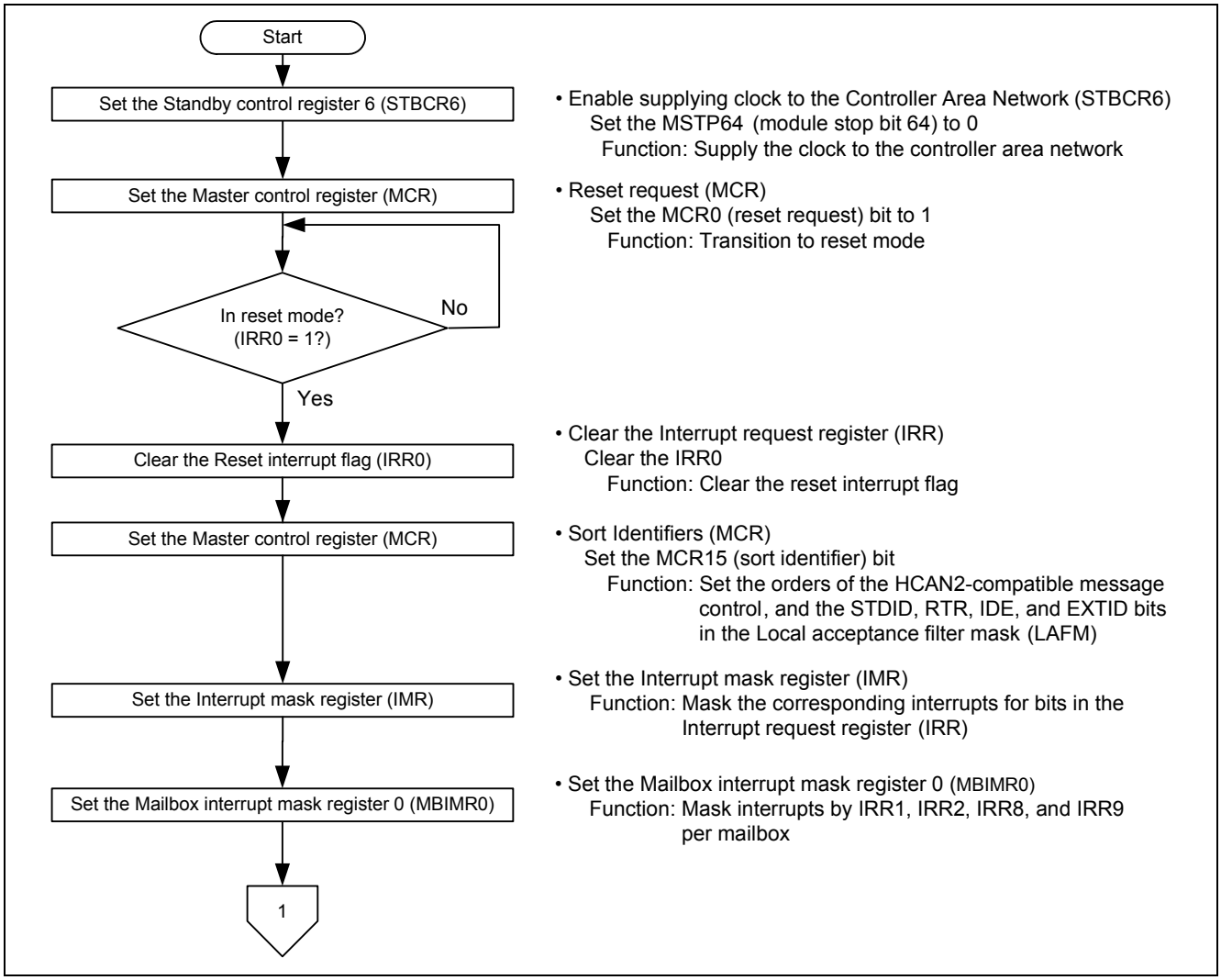


Figure 3 Flow Chart for Configuring the Controller Area Network (1/2)

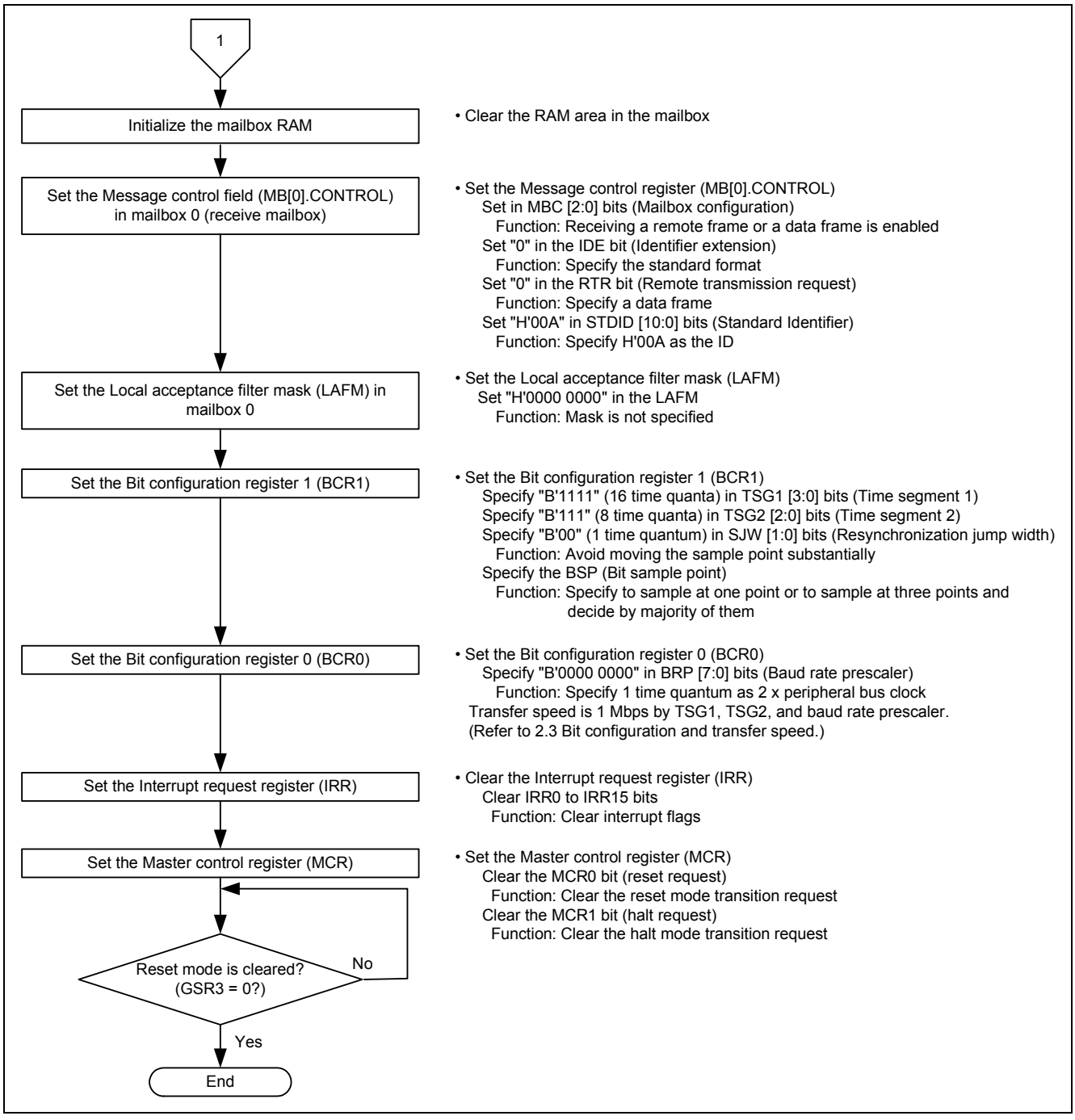


Figure 4 Flow Chart for Configuring the Controller Area Network (2/2)

(2) Steps to configure the Direct Memory Access Controller

When using the Controller Area Network data frame received interrupt (RM0_0) as an interrupt source, only cycle steal mode can be specified. This application uses the reload function. Figure 5 shows the flow chart for configuring the Direct Memory Access Controller. For more information on register settings, refer to the SH7216 Group Hardware Manual.

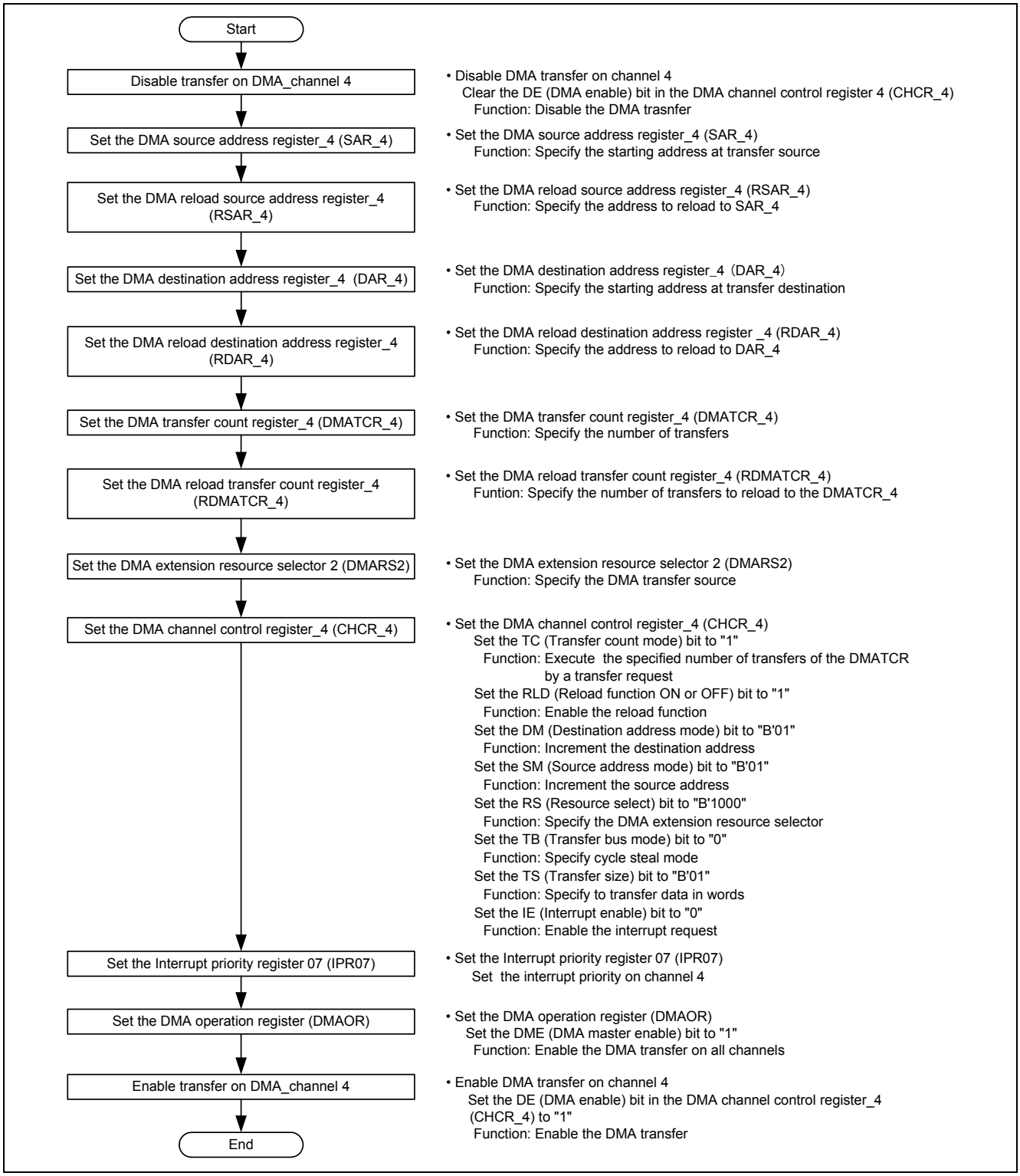


Figure 5 Flow Chart for Configuring the Direct Memory Access Controller

2.3 Bit Configuration and Transmission Speed

One bit in the Controller Area Network consists of the following four segments:

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

Each segment is composed of the reference time Tq (time quanta). Figure 6 shows the bit configuration example when SS = 1 Tq, PRSEG = 8 Tq, PHSEG1 = 8 Tq, and PHSEG2 = 8 Tq.

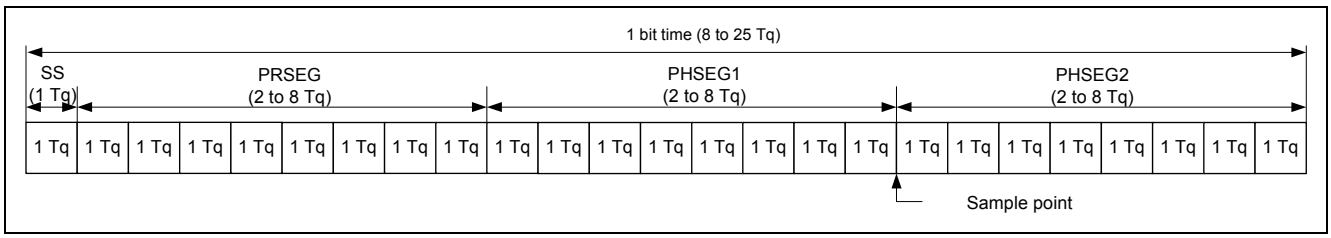


Figure 6 Bit Configuration

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

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

$$= \frac{\text{Peripheral Bus Clock}}{2 \times (BRP\ [7:0] + 1) \times \{(TSG1[3:0]+1) + (TSG2\ [2:0] + 1) + 1\}}$$

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

In the following description, bits BRP [7:0], TSG1 [3:0], and TSG2 [2:0] are register values, and bits BRP, TSEG1. TSEG2, and SJW are the corresponding values for the register values. For the corresponding values for register values, refer to the Controller Area Network chapter in the SH7216 Group Hardware Manual.

Following is the restriction on setting the bit configuration register.

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

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

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

As this sample program specifies the peripheral bus clock as 50 MHz, BRP [7:0] = 0, TSG1 [3:0] = 15, and TSEG2 [2:0] = 7, the transmission speed is calculated as follows:

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

2.4 Sample Program Operation

This sample program receives a standard CAN data frame with identifier H'00A in mailbox 0. Activate the Direct Memory Access Controller by the data frame received interrupt (RM0_0) to transfer data in mailbox 0 to the on-chip RAM. Note that the data frame received interrupt does not occur to the CPU. After the transfer is completed, the Direct Memory Access Controller resets the DMA source address register, DMA destination address register, DMA transfer count register to default values by its reload function. Clear the TE flag using the transfer end interrupt by the Direct Memory Access Controller and prepare for the next data frame received interrupt.

Note: When the Direct Memory Access Controller transfers data in mailbox 0 using the data frame received interrupt (RM0_0) as an interrupt source, data from message control field 0 (CONTROL0) to message control field 1 (CONTROL1) in mailbox 0 must be included.

Figure 7 shows the sample program operation (overview).

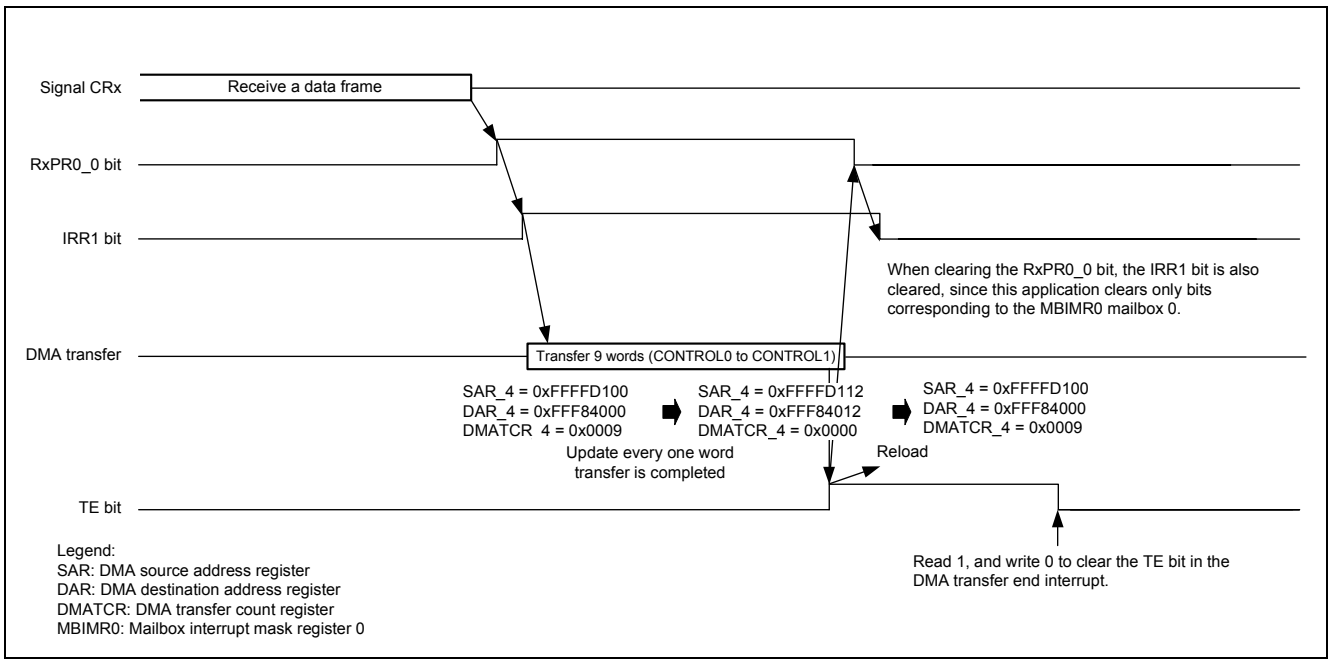


Figure 7 Sample Program Operation (Overview)

2.5 Sample Program Procedure

The table below lists setting examples of the Controller Area Network. Table 5 lists setting examples of the Direct Memory Access Controller. Figure 8 shows the configuration flow chart of this sample program.

Table 4 Controller Area Network Settings

Register Name	Address	Setting	Description
Standby control register (STBCR6)	H'FFFE 041C	H'8F	MSTP64 = "0": Controller Area Network is operating
Master control register (MCR)	H'FFFF D000	H'0001	MCR0 = "1": Reset mode transition request
		H'8001	MCR15 = "1": The order of the RCAN-ET message and of the HCAN2 message are different
		H'8000	MCR0 = "0": Reset mode is cleared
Mailbox interrupt mask register 0 (MBIMR0)	H'FFFF D052	H'FFFE	MBIMR00 = "0": Enable the receive interrupt in mailbox 0
Interrupt mask register (IMR)	H'FFFF D00A	H'FFFD	Enable the data frame received interrupt
Bit configuration register 1 (BCR1)	H'FFFF D004	H'F700	TSG1 [3:0] = "B'1111": PRSEG + PHSEG1 = 16 Tq TSG2 [2:0] = "B'111": PHSEG2 = 8 Tq SJW="0": SJW = 1 Tq BSP = "0": Bit sampling at one point
Bit configuration register 0 (BCR0)	H'FFFF D006	H'0000	BRP [7:0] = "0": 1 Tq = 2 x Pφ
Message control field in mailbox 0 (MB[0].CONTROL1)	H'FFFF D110	H'0200	MBC [2:0] = "B'010": Receiving the data frame and remote frame is enabled
Message control field in mailbox 0 (MB[0].CONTROL0)	H'FFFF D100	H'0028 0000	IDE = "0": Standard format RTR = "0": Data frame STDID [10:0] = "H'00A": Standard identifier is H'00A
Local acceptance filter mask in mailbox 0 (MB[0].LAFM)	H'FFFF D104	H'0000 0000	Clear: Mask is not specified

Table 5 Direct Memory Access Controller Setting

Register Name	Address	Setting	Description
DMA source address register_4 (SAR_4)	H'FFFE 1040	H'FFFF D100	Transfer source starting address: Specify the starting address in mailbox 0
DMA reload source address register_4 (RSAR_4)	H'FFFE 1140	H'FFFF D100	Address to reload to the SAR_4: Specify the starting address in mailbox 0
DMA destination address register_4 (DAR_4)	H'FFFE 1044	H'FFF8 4000	Transfer destination starting address: Specify the starting address in the on-chip RAM page 1
DMA reload destination address register_4 (RDAR_4)	H'FFFE 1144	H'FFF8 4000	Address to reload to the DAR_4: Specify the starting address in the on-chip RAM page 1
DMA transfer count register_4 (DMATCR_4)	H'FFFE 1048	H'0000 0009	Number of transfers: 9
DMA reload transfer count register_4 (RDMATCR_4)	H'FFFE 1148	H'0000 0009	Number of transfers to reload to the DMATCR_4: 9
DMA channel control register_4 (CHCR_4)	H'FFFE 104C	H'0000 0000	DE = "0": Disable the DMA transfer
		H'9000 580C	TC = "1" Execute the specified number of transfers of the DMATCR0 by a transfer request RLD = "1": Enable the reload function DM = "B'01": Increment the destination address SM = "B'01": Increment the source address RS = "B'1000": Specify the DMA extension resource selector TB = "0": Specify cycle steal mode TS = "B'01": Specify to transfer data in words IE = "0": Enable the interrupt request
		H'9000 580D	DE = "1": Enable the DMA transfer
DMA operation register (DMAOR)	H'FFFE 1200	H'0001	
DMA extension resource selector 2 (DMARS2)	H'FFFE 1308	H'0086	Specify the Controller Area Network as the DMA transfer source
Interrupt priority register 07 (IPR07)	H'FFFE 0C02	H'A000	Transfer end interrupt priority: Specify level 10

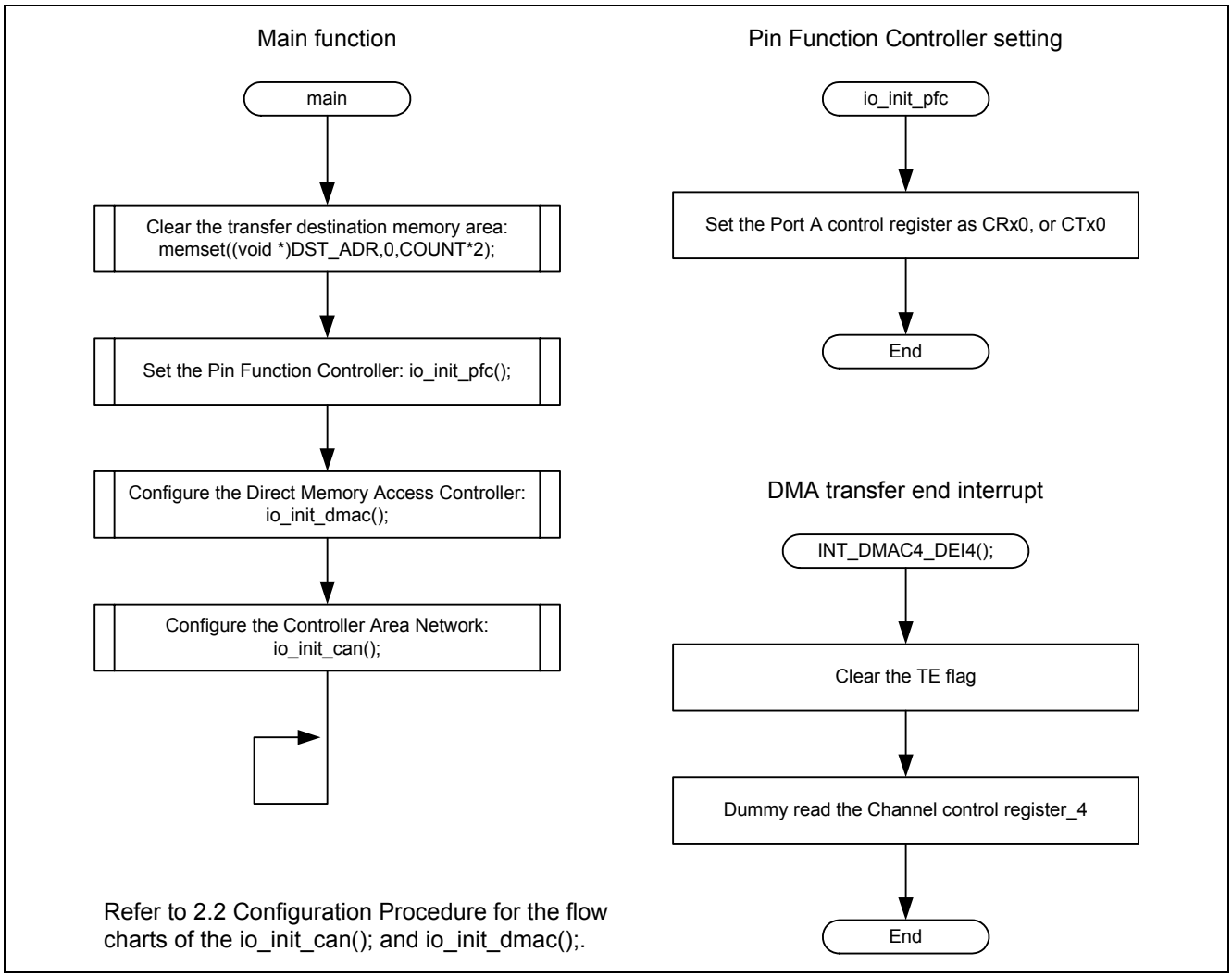


Figure 8 Sample Program Flow Chart

3. Sample Program Listing

3.1 Sample Program List "main.c" (1/7)

```
1   /***********************************************************************
2   *   DISCLAIMER
3   *
4   *   This software is supplied by Renesas Technology Corp. and is only
5   *   intended for use with Renesas products. No other uses are authorized.
6   *
7   *   This software is owned by Renesas Technology Corp. and is protected under
8   *   all applicable laws, including copyright laws.
9   *
10  *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11  *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12  *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13  *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14  *   DISCLAIMED.
15  *
16  *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17  *   TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18  *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19  *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20  *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21  *
22  *   Renesas reserves the right, without notice, to make changes to this
23  *   software and to discontinue the availability of this software.
24  *   By using this software, you agree to the additional terms and
25  *   conditions found by accessing the following link:
26  *   http://www.renesas.com/disclaimer
27  ***********************************************************************
28  *   Copyright (C) 2010. Renesas Technology Corp., All Rights Reserved.
29  *""FILE COMMENT""*********** Technical reference data *************************
30  *   System Name : SH7216 Sample Program
31  *   File Name   : main.c
32  *   Abstract    : DMAC+CAN Module Application (Data Frame Receive)
33  *   Version     : 1.00.00
34  *   Device      : SH7216
35  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36  *               : C/C++ compiler package for the SuperH RISC engine family
37  *               :                               (Ver.9.03 Release00).
38  *   OS          : None
39  *   H/W Platform: R0K572167 (CPU board)
40  *   Description :
41  ***********************************************************************
42  *   History     : Jan.26,2010 ver.1.00.00
43  *""FILE COMMENT END""*****************************************************/
```

3.2 Sample Program List "main.c" (2/7)

```
44  #include "iodefine.h"
45  #include <stdio.h>
46
47  /* ---- prototype declaration ---- */
48  void main(void);
49  void io_init_pfc(void);
50  void io_init_can(void);
51  void io_init_dmac(void *src, void *dst, int count);
52
53  /* ---- symbol definition ---- */
54  #define CAN_IRR0 0x0001
55
56  #define DST_ADR       0xfff84000  /* Transfer destination address:Starting address in
57                                       the on-chip SRAM page 1 */
58  #define SRC_ADR       0xffffd100  /* Transfer source address: Starting address in
59                                       mailbox 0 */
60  #define COUNT         9           /* Transfers 9 words */
61
62  /*""FUNC COMMENT""*****************************************************************
63   * ID            :
64   * Outline       : Sample program main
65   *-----------------------------------------------------------------------------
66   * Include       : "iodefine.h"
67   *-----------------------------------------------------------------------------
68   * Declaration   : void main(void);
69   *-----------------------------------------------------------------------------
70   * Description   : Sets the Pin Function Controller, configures the Direct Memory
71   *               : Access Controller, and Controller Area Network.
72   *               : Receiving a data frame and transferring the data frame is
73   *               : executed automatically.
74   *-----------------------------------------------------------------------------
75   * Argument      : void
76   *-----------------------------------------------------------------------------
77   * Return Value  : void
78   *-----------------------------------------------------------------------------
79   * Note          :
80   *""FUNC COMMENT END""*************************************************************/
81  void main(void)
82  {
83    /* ==== Clears the transfer destination memory area ==== */
84    memset((void *)DST_ADR,0,COUNT*2);
85
86    /* ==== Sets the Pin Function Controller ==== */
87    io_init_pfc();
88
89    /* ==== Configures the Direct Memory Access Controller ==== */
90    io_init_dmac((void *)SRC_ADR, (void *)DST_ADR, COUNT);
91
```

3.3 Sample Program List "main.c" (3/7)

```
92        /* ==== Configures the Controller Area Network ==== */
93        io_init_can();
94
95        while(1){
96          /* loop */
97        }
98      }
99
100     /*""FUNC COMMENT""*****************************************************************
101      * ID            :
102      * Outline       : PFC setting
103      *-------------------------------------------------------------------------------
104      * Include       : "iodefine.h"
105      *-------------------------------------------------------------------------------
106      * Declaration   : void io_init_pfc(void);
107      *-------------------------------------------------------------------------------
108      * Description   : Set pin functions (CRx0 input and CTx0 output).
109      *-------------------------------------------------------------------------------
110      * Argument      : void
111      *-------------------------------------------------------------------------------
112      * Return Value  : void
113      *-------------------------------------------------------------------------------
114      * Note          :
115      *""FUNC COMMENT END""*************************************************************/
116     void io_init_pfc(void)
117     {
118       /* ==== Sets the Pin Function Controller ==== */
119       PFC.PACRL1.BIT.PA0MD = 0x5;             /* Set CRx0 */
120       PFC.PACRL1.BIT.PA1MD = 0x5;             /* Set CTx0 */
121     }
122
```

3.4 Sample Program List "main.c" (4/7)

```
123  /*""FUNC COMMENT""*************************************************************
124   * ID            :
125   * Outline       : RCAN setting
126   *-----------------------------------------------------------------------------
127   * Include       : "iodefine.h"
128   *-----------------------------------------------------------------------------
129   * Declaration   : void io_init_can(void);
130   *-----------------------------------------------------------------------------
131   * Description   : Configure the Controller Area Network (RCAN).
132   *               : Transfer speed is set as 1 Mbps.
133   *-----------------------------------------------------------------------------
134   * Argument      : void
135   *-----------------------------------------------------------------------------
136   * Return Value  : void
137   *-----------------------------------------------------------------------------
138   * Note          :
139   *""FUNC COMMENT END""*********************************************************/
140  void io_init_can(void)
141  {
142    int i,j;
143
144    /* ==== Sets the Standby control register 6 ==== */
145    STB.CR6.BYTE = 0x8f;               /* Clear the RCAN module standby */
146
147    /* ==== Sets the Master control register ==== */
148    RCANET.MCR.WORD = 0x0001;              /* Sets reset mode */
149    while((RCANET.IRR.WORD & CAN_IRR0) != CAN_IRR0){
150      /* Waits for transition to reset mode end */
151    }
152    /* ==== IRR = 1, GSR = 1 (Automatically set) ==== */
153
154    /* ---- Clear the Reset interrupt flag ---- */
155    RCANET.IRR.WORD = 0x0001;
156
157    /* ---- Sets the Master control register ---- */
158    RCANET.MCR.WORD |= 0x8000;             /* The order of the RCAN-ET message and
159                                           of the HCAN2 message are different */
160
161    /* ---- Sets the Interrupt mask register ---- */
162    RCANET.IMR.WORD  = 0xfffd;
163
164    /* ---- Sets the Mailbox interrupt mask register 0 ---- */
165    RCANET.MBIMR0.WORD  = 0xfffe;
166
```

3.5 Sample Program List "main.c" (5/7)

```
167     /* ---- Clears the mailbox RAM area ---- */
168     for(i = 0; i < 16; i++){
169       RCANET.MB[i].CTRL0.LONG = 0x00000000;
170       RCANET.MB[i].LAFM.LONG = 0x00000000;
171       for(j = 0; j < 8; j++){
172           RCANET.MB[i].MSG_DATA[j] =  0x00;
173       }
174     }
175
176     /* ---- Sets mailbox 0 ---- */
177     RCANET.MB[0].CTRL1.WORD = 0x0200;       /* MBC = 2, dlc = 0  */
178     RCANET.MB[0].CTRL0.LONG = 0x00280000;   /* Standard data frame, id=0x00a */
179     RCANET.MB[0].LAFM.LONG = 0x00000000;
180     for(i = 0; i < 8; i++){                 /* Clear data */
181       RCANET.MB[0].MSG_DATA[i] =  0x00;
182     }
183
184     /* ---- Set the Bit configuration register ---- */
185     RCANET.BCR1.WORD = 0xf700;              /* tsg1 = 15 (16-bit), tsg2 = 7 (8-bit),
186                                             sjw = 0 (1-bit), bsp = 0 */
187     RCANET.BCR0.WORD = 0x0000;              /* 1 Mbps */
188
189     /* ---- Set the Interrupt request register ---- */
190     RCANET.IRR.WORD = 0xffff;
191
192     /* ---- Set the Master control register ---- */
193     RCANET.MCR.WORD &= 0xf8fc;              /* MCR0, MCR1 clear */
194     while( (RCANET.GSR.WORD & 0x0008) != 0x0000 ){
195       /* Waits for reset mode end */
196     }
197   }
198
```

3.6 Sample Program List "main.c" (6/7)

```
/*""FUNC COMMENT""*************************************************************
 * ID            :
 * Outline       : DMAC setting
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void io_init_dmac(void);
 *-----------------------------------------------------------------------------
 * Description   : Configures the Direct Memory Access Controller (DMAC).
 *               : Specifies cycle steal mode, dual address mode, on-chip
 *               : peripheral module request (RCAN). Specifies the transfer data
 *               : length in bytes, RCAN mailbox 0 as transfer source, on-chip RAM
 *               : as the transfer destination. It uses the reload function.
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""*********************************************************/
void io_init_dmac(void *src, void *dst, int count)
{
  /* ==== Disable the DMA transfer on channel 4 ==== */
  DMAC4.CHCR.BIT.DE = 0x0;            /* Disable the DMA transfer */

  /* ==== Sets the DMA source address register_4 (SAR_4) ==== */
  DMAC4.SAR = src;

  /* ==== Sets the DMA reload source address register_4 (RSAR_4) ==== */
  DMAC4.RSAR = src;

  /* ==== Sets the DMA destination address register_4 (DAR_4) ==== */
  DMAC4.DAR = dst;

  /* ==== Sets the DMA reload destination address register_4 (RDAR_4) ==== */
  DMAC4.RDAR = dst;

  /* ==== Sets the DMA transfer count register_4 (DMATCR_4) ==== */
  DMAC4.DMATCR =COUNT;

  /* ==== Sets the DMA reload transfer count register_4 (RDMATCR_4) ==== */
  DMAC4.RDMATCR =COUNT;

  /* ==== Sets the DMA extension resource selector 2 (DMARS2) ==== */
  DMAC.DMARS2.WORD =0x0086;

```

3.7 Sample Program List "main.c" (7/7)

```
245     /* ==== Sets the DMA channel control register_4 (CHCR_4) ==== */
246     DMAC4.CHCR.LONG = 0x9000580c;
247                 /*
248                     bit 31       : TC DMATCR transfer: 1----  Transfers DMATCR
249                                    for specified number of times
250                     bits 30 to 29: reserve 0
251                     bit 28       : RLD ON : 1--------------  Enables the reload function
252                     bits 27 to 24: reserve 0
253                     bit 23       : DO over run0 : 0----------Not used
254                     bit 22       : TL TEND low active : 0----Not used
255                     bits 21 to 20: reserve 0
256                     bit 19       : HE :0--------------------Not used
257                     bit 18       : HIE :0-------------------Not used
258                     bit 17       : AM :0--------------------Not used
259                     bit 16       : AL :0--------------------Not used
260                     bits 15 to 14: DM1:0 DM0:1--------------Increments the
261                                                             destination address
262                     bits 13 to 12: SM1:0 SM0:1--------------Increments the
263                                                             source address
264                     bits 11 to 8 : RS : auto request : B'1000-   Auto request
265                     bit 7        : DL : DREQ level : 0 -------   Not used
266                     bit 6        : DS : DREQ select :0 Low level Not used
267                     bit 5        : TB :cycle :0---------------   Cycle steal mode
268                     bits 4 to 3  : TS : transfer size : B'01---   Transfer in words
269                     bit 2        : IE : interrupt enable : 1---   Enables interrupt
270                     bit 1        : TE : transfer end : 0-------   Clears the TE flag
271                     bit 0        : DE : DMA enable bit : 0-----   Disables the DMA transfer
272                 */
273
274
275     /* ======= Sets the Interrupt priority register 07 (IPR07) ======= */
276     INTC.IPR07.BIT._DMAC4 = 0xa;
277
278
279     /* ---- Sets the DMA operation register ---- */
280     DMAC.DMAOR.WORD |= 0x0007;              /* Sets the DME bit, writes 1 to bits
281                                         /* AE and NMIF to prevent the address error */
282                                         /* and NMI flags from clearing */
283
284
285     /* ======= Enables the DMA transfer on _channel 4 ======= */
286     DMAC4.CHCR.BIT.DE = 0x1;
287
288
289   }
290
291   /* End of File */
```

3.8 Sample Program List "intprg.c"

```
    ==== Preceding information deleted ====



293   // 124 DMAC4 DEI4
294   void INT_DMAC4_DEI4(void)
295   {
296
297     int dummy;
298
299     /* ======= Clears the TE flag ======= */
300     DMAC4.CHCR.BIT.TE = 0;
301
302     /* ======= Dummy read the Channel control register_4 ======= */
303     dummy=DMAC4.CHCR.WORD.L;
304
305   }

      ==== additional information deleted ====
```

4. References

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

- Hardware Manual
  SH7216 Group Hardware Manual Rev. 1.01
  The latest version of the hardware manual can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Jun.04.10	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

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

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

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

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

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

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

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

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

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

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

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

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

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

Colophon 1.0