SH7262/SH7264 Group

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

REJ06B0982-0100
Rev.1.00
Apr. 28, 2010

Summary

This application note describes the configuration example of the SH7264 microcomputers (MCUs) to receive data frames using the Controller Area Network, and to store data frames in internal RAM using the Direct Memory Access Controller.

Target Device

SH7262/7264 MCU (In this document, SH7262/SH7264 are described as SH7264.)

Contents

1. Introduction 2

2. Applications 3

3. Sample Program Listing 17

4. References 28

1. Introduction

1.1 Specifications

This application activates the SH7264 Direct Memory Access Controller and stores the data in large-capacity internal RAM every time the SH7264 Controller Area Network receives a data frame.

1.2 Modules Used

- Controller Area Network module
- Direct Memory Access Controller (DMAC)

1.3 Applicable Conditions

MCU	SH7262/SH7264
Operating Frequencies	Internal clock: 144 MHz Bus clock: 72 MHz Peripheral clock: 36 MHz
Integrated Development Environment	Renesas Technology Corp. High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Technology SuperH RISC engine Family C/C++ Compiler Package Ver.9.03 Release 00
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu -fpu=single -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

- SH7262/SH7264 Group Example of Initialization
- SH7262/SH7264 Group Controller Area Network, Configuration to Transmit Data Frames
- SH7262/SH7264 Group Controller Area Network, Configuration to Receive Data Frames
- SH7262/SH7264 Group Controller Area Network, Configuration to Transmit Remote Frames
- SH7262/SH7264 Group Controller Area Network, Configuration to Receive Remote Frames

1.5 About Active-low Pins (Signals)

The symbol "#" suffixed to the pin (or signal) names indicates that the pins (or signals) are active-low.

2. Applications

This application receives the data frame using the Controller Area Network, and stores the data frame in large-capacity internal RAM using the Direct Memory Access Controller.

2.1 Overview of Modules

(1) Controller Area Network
The SH7264 includes two Controller Area Network channels which are compliant with the CAN protocol, version 2.0B active, and ISO 11898.

The Controller Area Network module has 31 programmable mailboxes for transmission/reception, one mailbox for reception, and one programmable receive filtering mask to provide flexible communication procedure. Table 1 lists 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 interrupt only. For more details, refer to the Controller Area Network chapter in the SH7262 Group, SH7264 Group Hardware Manual.

Table 1 Controller Area Network Features

Items	Description
Protocol	CAN protocol, version 2.0B. Bit timing is compliant to ISO 11898
Number of channels	2 (1)
Number of mailboxes	32 (1) (31 programmable transmit/receive mailbox, and one receive mailbox)
Transfer speed	Up to 1 Mbps
Number of interrupt sources	16
Trigger mode	Event-trigger mode, and time-trigger mode
Test mode	Includes listen-only mode, and error passive mode
DMA transfer	DMAC can be activated by mailbox 0 data frame received interrupt, or remote frame received interrupt

Note: Two CAN channels can be used together as 64 mailboxes × one CAN channel.

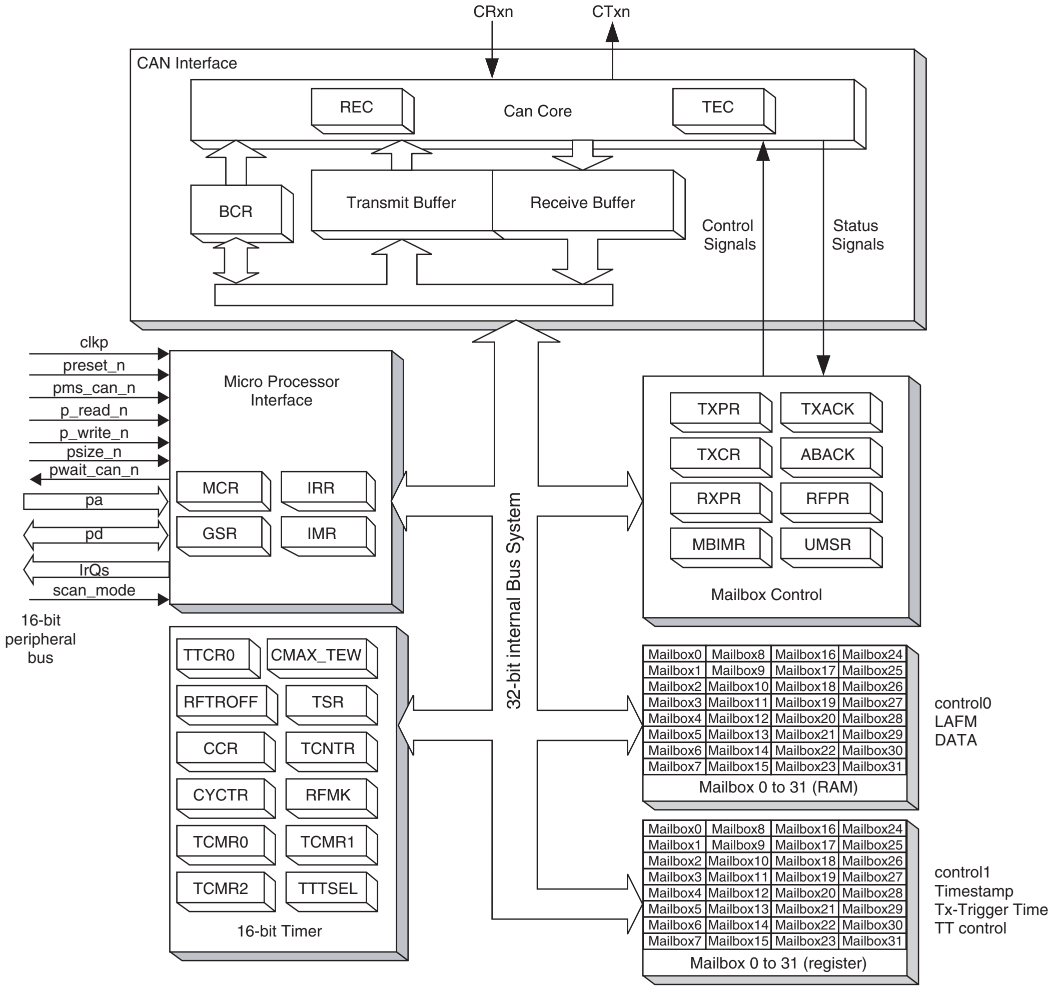

CRxn
CTxn
CAN Interface
REC
Can Core
TEC
BCR
Transmit Buffer
Receive Buffer
Control Signals
Status Signals
clkp
preset_n
pms_can_n
p_read_n
p_write_n
psize_n
pwait_can_n
pa
pd
IrQs
scan_mode
16-bit peripheral bus
Micro Processor Interface
MCR
IRR
GSR
IMR
32-bit internal Bus System
TXPR
TXACK
TXCR
ABACK
RXPR
RFPR
MBIMR
UMSR
Mailbox Control
TTCR0
CMAX_TEW
RFTROFF
TSR
CCR
TCNTR
CYCTR
RFMK
TCMR0
TCMR1
TCMR2
TTTSEL
16-bit Timer
Mailbox0 Mailbox8 Mailbox16 Mailbox24
Mailbox1 Mailbox9 Mailbox17 Mailbox25
Mailbox2 Mailbox10 Mailbox18 Mailbox26
Mailbox3 Mailbox11 Mailbox19 Mailbox27
Mailbox4 Mailbox12 Mailbox20 Mailbox28
Mailbox5 Mailbox13 Mailbox21 Mailbox29
Mailbox6 Mailbox14 Mailbox22 Mailbox30
Mailbox7 Mailbox15 Mailbox23 Mailbox31
Mailbox 0 to 31 (RAM)
control0
LAFM
DATA
Mailbox0 Mailbox8 Mailbox16 Mailbox24
Mailbox1 Mailbox9 Mailbox17 Mailbox25
Mailbox2 Mailbox10 Mailbox18 Mailbox26
Mailbox3 Mailbox11 Mailbox19 Mailbox27
Mailbox4 Mailbox12 Mailbox20 Mailbox28
Mailbox5 Mailbox13 Mailbox21 Mailbox29
Mailbox6 Mailbox14 Mailbox22 Mailbox30
Mailbox7 Mailbox15 Mailbox23 Mailbox31
Mailbox 0 to 31 (register)
control1
Timestamp
Tx-Trigger Time
TT control

Figure 1 Controller Area Network Block Diagram (Per One Channel)

Table 2 Controller Area Network Interrupt Sources (Per One Channel)

Interrupt	Source	Interrupt flag	Activating the Direct Memory Access Controller
ERS	Error passive (TEC ≥ 128, or REC ≥ 128)	IRR5	Not allowed
	Bus off (TEC ≥ 256)/Bus off recovery	IRR6	
	Error warning (TEC ≥ 96)	IRR3	
	Error warning (REC ≥ 96)	IRR4	
OVR	Transition to Reset/halt/CAN sleep	IRR0	
	Overload frame transmission	IRR7	
	Unread message overwrite (overrun)	IRR9	
	Start of new system matrix	IRR10	
	TCMR2 compare match	IRR11	
	Detection of CAN bus operation in CAN sleep mode	IRR12	
	Timer overrun/Next_is_Gap/Message error	IRR13	
	TCMR0 compare match	IRR14	
	TCMR1 compare match	IRR15	
RM0 [1] RM1 [1]	Data frame reception	IRR1 [2]	Allowed [3]
	Remote frame reception	IRR2 [2]	
SLE	Message transmission/transmission disabled (slot empty)	IRR8	Not allowed

Notes: 1 RM0 is an interrupt generated by the remote frame pending flag for mailbox 0 (RFPR0 [0]) or the data frame pending flag for mailbox 0 (RXPR0 [0]). RM1 is an interrupt generated by the Remote frame pending flag for mailbox n (RFPR1 [n]) or the data frame pending flag for mailbox n (RXPR1 [n]) (n = 1 to 31).

2. IRR1 is a data frame received interrupt flag for mailboxes 0 to 31, and IRR2 is a remote frame receive interrupt flag for mailboxes 0 to 31.

3. The Direct Memory Access Controller can be activated only by the RM0 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, internal 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.

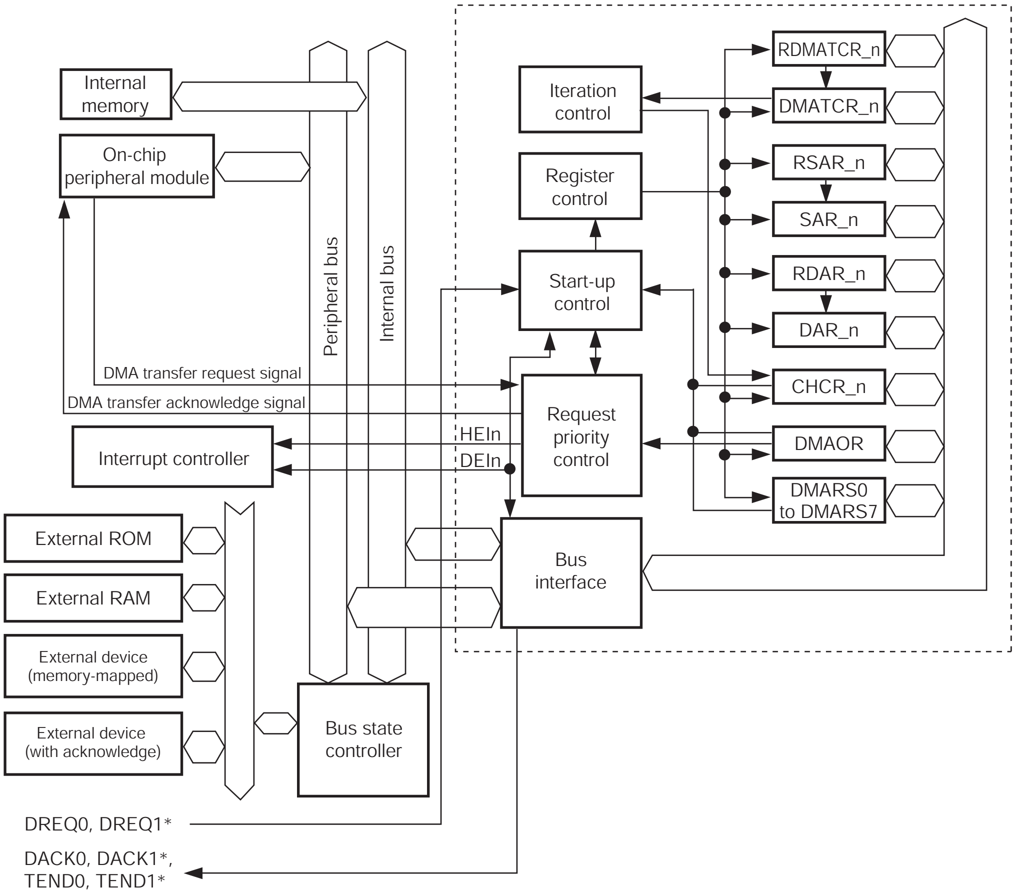
In cycle steal mode, the Direct Memory Access Controller leaves the bus to other masters when it finishes "a transmit" (in bytes, words, long words, or 16 bytes). When the Direct Memory Access Controller receives another transfer request, it retrieves the bus again. Then, it transfers data in unit of 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 large-capacity internal RAM using cycle steal mode.

Table 3 lists the features of the Direct Memory Access Controller. Figure 2 shows its block diagram. For more information, refer to the Direct Memory Access Controller chapter in the "SH7262 Group, SH7264 Group Hardware manual".

Table 3 Direct Memory Access Controller Features

Items	Description
Number of channels	16 channels
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, and reload register.



[Legend]

RDMATCR:	DMA reload transfer count register	CHCR:	DMA channel control register
DMATCR:	DMA transfer count register	DMAOR:	DMA operation register
RSAR:	DMA reload source address register	DMARS0 to DMARS7:	DMA extension resource selectors 0 to 7
SAR:	DMA source address register	HEIn:	DMA transfer half-end interrupt request to the CPU
RDAR:	DMA reload destination address register	DEIn:	DMA transfer end interrupt request to the CPU
DAR:	DMA destination address register	n = 0 to 15	

Note: * Pins in channel 1 can be used only in the SH7264 Group.

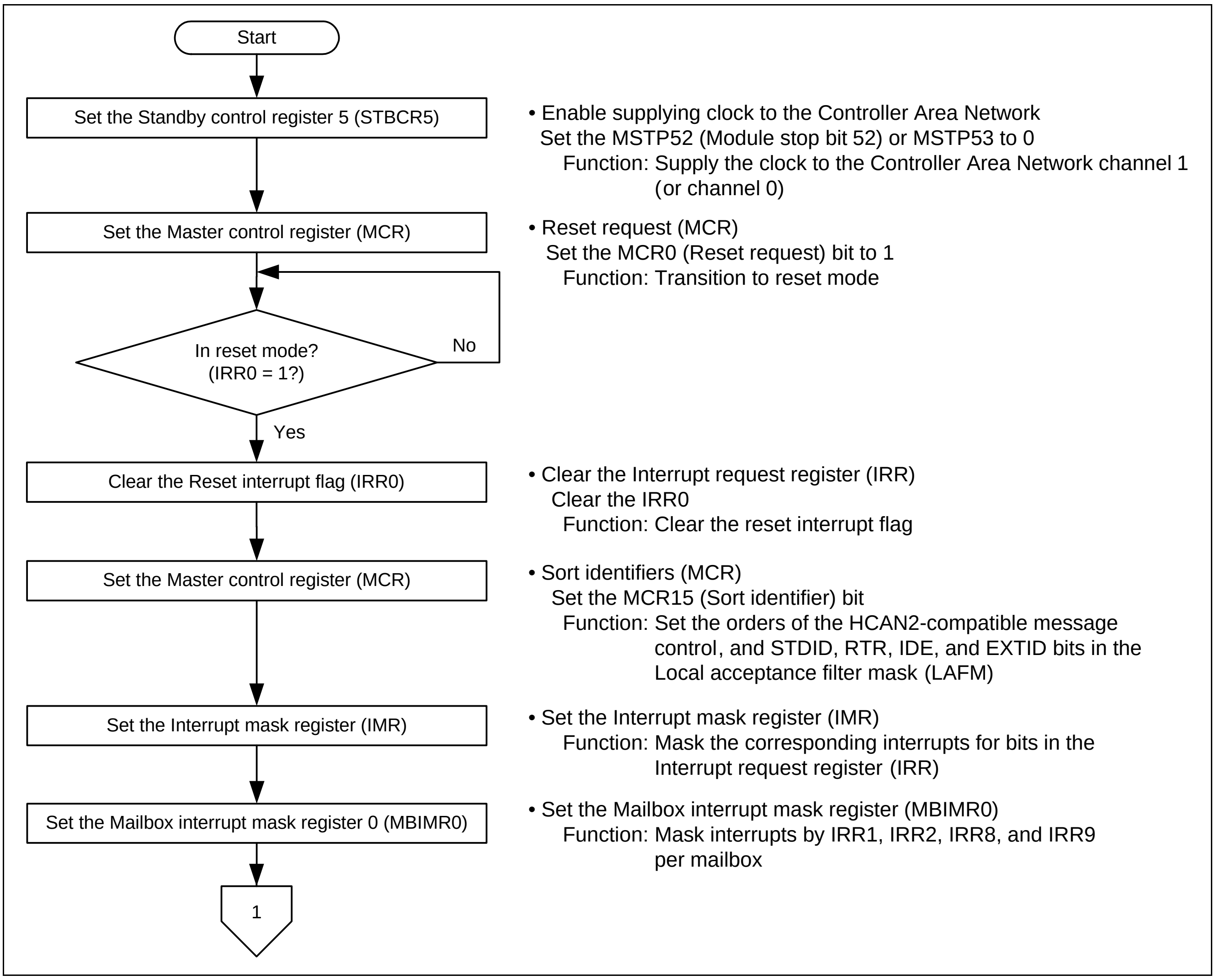
Figure 2 Direct Memory Access Controller Block Diagram

2.2 Configuration Procedure

(1) Steps to configure the Controller Area Network

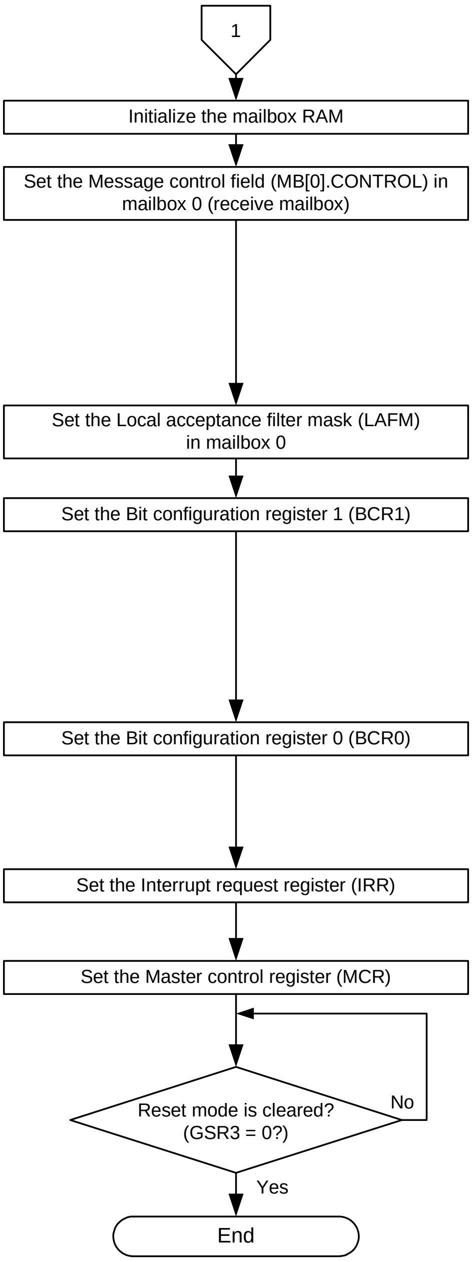
Configure the Controller Area Network in reset mode (configuration mode). After configuration is completed, clear the reset mode to join the CAN bus activity. To activate the Direct Memory Access Controller by the Controller Area Network data frame received interrupt (IRR1), set bit 1 (IMR1) in the Interrupt mask register (IMR), and bit 0 (MBIMR0[0]) in the Mailbox interrupt mask register 0 (MBIMR0) to enable interrupts.

Figure 3 and Figure 4 show flow charts for configuring the Controller Area Network. For details on register settings, refer to the SH7262 Group, SH7264 Group Hardware Manual.


Start
Set the Standby control register 5 (STBCR5)
Set the Master control register (MCR)
In reset mode? (IRR0 = 1?)
No
Yes
Clear the Reset interrupt flag (IRR0)
Set the Master control register (MCR)
Set the Interrupt mask register (IMR)
Set the Mailbox interrupt mask register 0 (MBIMR0)
1

- Enable supplying clock to the Controller Area Network
  Set the MSTP52 (Module stop bit 52) or MSTP53 to 0
  Function: Supply the clock to the Controller Area Network channel 1 (or channel 0)
- Reset request (MCR)
  Set the MCR0 (Reset request) bit to 1
  Function: Transition to reset mode
- Clear the Interrupt request register (IRR)
  Clear the IRR0
  Function: Clear the reset interrupt flag
- Sort identifiers (MCR)
  Set the MCR15 (Sort identifier) bit
  Function: Set the orders of the HCAN2-compatible message control, and STDID, RTR, IDE, and EXTID bits in the Local acceptance filter mask (LAFM)
- Set the Interrupt mask register (IMR)
  Function: Mask the corresponding interrupts for bits in the Interrupt request register (IRR)
- Set the Mailbox interrupt mask register (MBIMR0)
  Function: Mask interrupts by IRR1, IRR2, IRR8, and IRR9 per mailbox

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

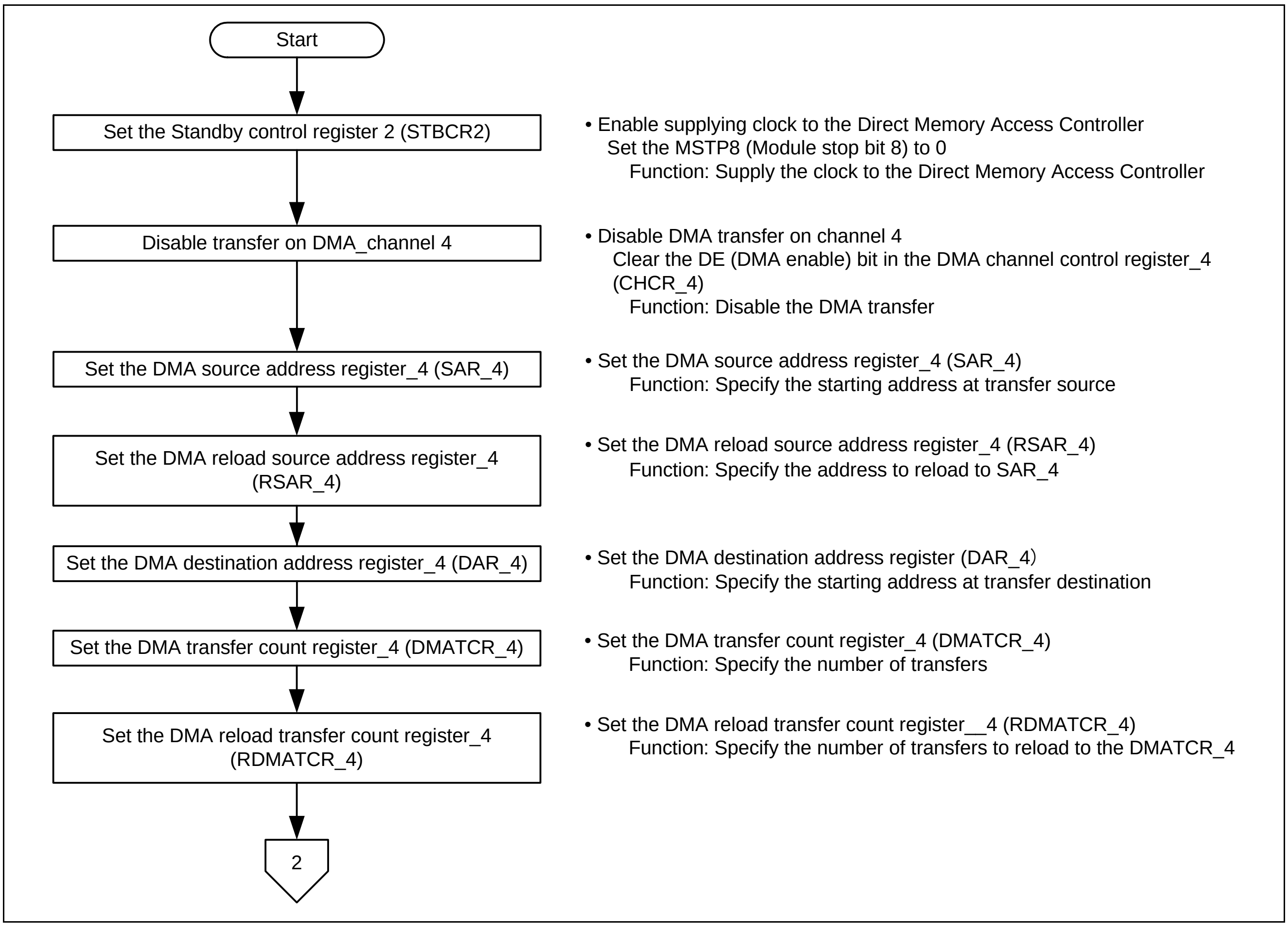

1
Initialize the mailbox RAM
Set the Message control field (MB[0].CONTROL) in mailbox 0 (receive mailbox)
Set the Local acceptance filter mask (LAFM) in mailbox 0
Set the Bit configuration register 1 (BCR1)
Set the Bit configuration register 0 (BCR0)
Set the Interrupt request register (IRR)
Set the Master control register (MCR)
Reset mode is cleared? (GSR3 = 0?)
No
Yes
End

- Clear the RAM area in the mailbox
- Set the Message control register (MB[0].CONTROL)
  Set MBC [2:0] (Mailbox configuration) bits to "B'010"
  Function: Receiving a remote frame or a data frame is enabled
  Set the IDE (Identifier extension) bit to 0
  Function: Specify the standard format
  Set the RTR (Remote transmission request) to 0
  Function: Specify a data frame
  Set STDID [10:0] (Standard identifier) bits to "H'00A"
  Function: Specify the identifier as H'00A
- Set the Local acceptance filter mask (LAFM)
  Set the LAFM to H'0000 0000
  Function: Mask is not specified
- Set the Bit configuration register 1 (BCR1)
  Set TSG1 [3:0] (Time segment 1) bits to "B'1001" (10 time quanta)
  Set TSG2 [2:0] (Time segment 2) bits to "B'110" (7 time quanta)
  Set SJW [1:0] (Resynchronization jump width) bits to "B'00" (1 time quantum)
  Function: Avoid moving the sample point substantially
  Set the BSP (Bit sample point) bit
  Function: Specify to sample at one point or to sample at three points and decide by majority of them
- Set the Bit configuration register 0 (BCR0)
  Set BRP [7:0] (Baud rate prescaler) bits to "B'0000 0000"
  Function: Specify 1 time quantum as 2 × peripheral bus clock
  Transfer speed is 1 Mbps by TSG1, TSG2, and baud rate prescaler.
  (Refer to 2.3 Bit configuration and transfer speed)
- Clear the Interrupt request register (IRR)
  Clear IRR0 to IRR15 bits
  Function: Clear interrupt flags
- Set the Master control register (MCR)
  Clear the MCR0 (Reset request) bit
  Function: Clear the reset mode transition request
  Clear the MCR1 (Halt request) bit
  Function: Clear the halt mode transition request

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

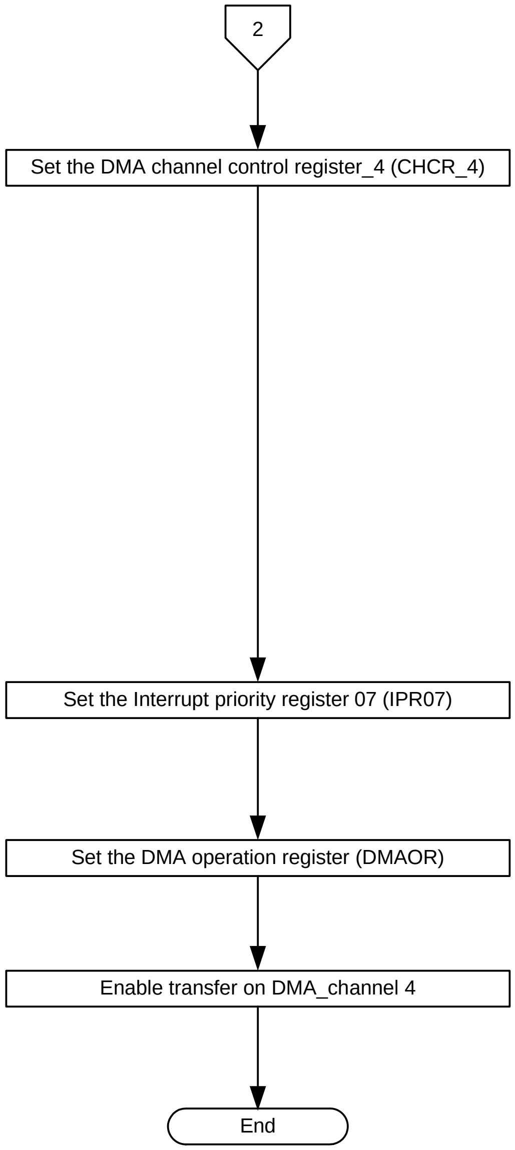
(2) Steps to configure the Direct Memory Access Controller

When using the Controller Area Network data frame received interrupt (IRR1) as an interrupt source, only cycle steal mode can be specified. This application uses the reload function in the DMA source address register, and DMA transfer count register. Figure 5 and Figure 6 show flow charts for configuring the Direct Memory Access Controller. For details on register settings, refer to the SH7262 Group, SH7264 Group Hardware Manual.


Start
Set the Standby control register 2 (STBCR2)
Disable transfer on DMA_channel 4
Set the DMA source address register_4 (SAR_4)
Set the DMA reload source address register_4 (RSAR_4)
Set the DMA destination address register_4 (DAR_4)
Set the DMA transfer count register_4 (DMATCR_4)
Set the DMA reload transfer count register_4 (RDMATCR_4)
2

- Enable supplying clock to the Direct Memory Access Controller
  Set the MSTP8 (Module stop bit 8) to 0
  Function: Supply the clock to the Direct Memory Access Controller
- Disable DMA transfer on channel 4
  Clear the DE (DMA enable) bit in the DMA channel control register_4 (CHCR_4)
  Function: Disable the DMA transfer
- Set the DMA source address register_4 (SAR_4)
  Function: Specify the starting address at transfer source
- Set the DMA reload source address register_4 (RSAR_4)
  Function: Specify the address to reload to SAR_4
- Set the DMA destination address register (DAR_4)
  Function: Specify the starting address at transfer destination
- Set the DMA transfer count register_4 (DMATCR_4)
  Function: Specify the number of transfers
- Set the DMA reload transfer count register__4 (RDMATCR_4)
  Function: Specify the number of transfers to reload to the DMATCR_4

Figure 5 Flow Chart for Configuring the Direct Memory Access Controller (1/2)


2
Set the DMA channel control register_4 (CHCR_4)
Set the Interrupt priority register 07 (IPR07)
Set the DMA operation register (DMAOR)
Enable transfer on DMA_channel 4
End

- Set the DMA channel control register_4 (CHCR_4)
  Set the TC (Transfer count mode) bit to 1
  Function: Execute the number of transfers specified in the DMATCR by a transfer request
  Set the RLDSAR (SAR reload function ON or OFF) bit to 1
  Function: Enable the reload function to the SAR or DMATCR
  Set the RLDDAR (DAR reload function ON or OFF) bit to 0
  Function: Disable the reload function to the DAR or DMATCR
  Set the DM (Destination address mode) bit to "B'01"
  Function: Increment the destination address
  Set the SM (Source address mode) bit to "B'01"
  Function: Increment the source address
  Set the RS (Resource select) bit to "B'1010"
  Function: Specify the Controller Area Network channel 1
  Set the TB (Transfer bus mode) bit to 0
  Function: Specify cycle steal mode
  Set the TS (Transfer size) bit to "B'01"
  Function: Specify to transfer data in words
  Set the IE (Interrupt enable) bit to 0
  Function: Enable the interrupt request

- Set the Interrupt priority register 07 (IPR07)
  Set the interrupt priority on channel 4

- Set the DMA operation register (DMAOR)
  Set the DME (DMA master enable) bit to 1
  Function: Enable the DMA transfer on all channels

- Enable DMA transfer on channel 4
  Set the DE (DMA enable) bit in the DMA channel control register_4 (CHCR_4) to 1
  Function: Enable the DMA transfer

Figure 6 Flow Chart for Configuring the Direct Memory Access Controller (2/2)

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 7 shows the bit configuration example when SS = 1 Tq, PRSEG = 8 Tq, PHSEG1 = 8 Tq, and PHSEG2 = 8 Tq.

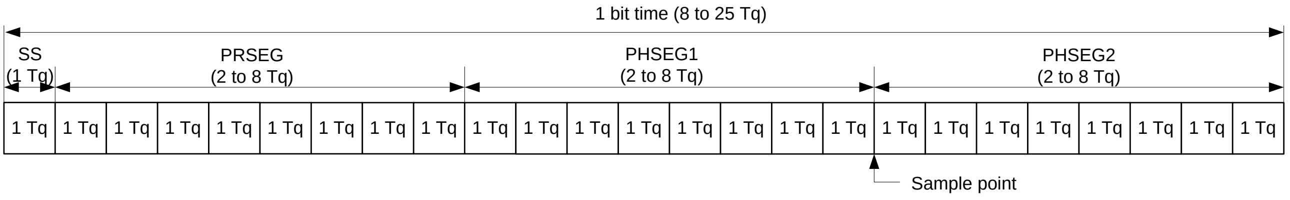


Figure 7 Bit Configuration

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 the 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 SH7262 Group, SH7264 Group Hardware Manual.

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

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

$$= \frac{\text{Peripheral bus clock}}{\langle 2\times(\mathrm{BRP}[7:0]+1)\times\{(\mathrm{TSG}[3:0]+1)+(\mathrm{TSG2}[2:0]+1)+1\}\rangle}$$

Following is the restriction on setting the Bit configuration register.

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

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

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

As this sample program specifies the peripheral bus clock as 36 MHz, BRP [7:0] = 0, TSG 1 [3:0] = 9, and TSG2 [2:0] = 6, the transmission speed is calculated as follows:

$$\text{Transmission speed} = \frac{36\mathrm{M}}{2\times(0+1)\times((9+1)+(6+1)+1)} = 1\mathrm{M....1\ Mbps}$$

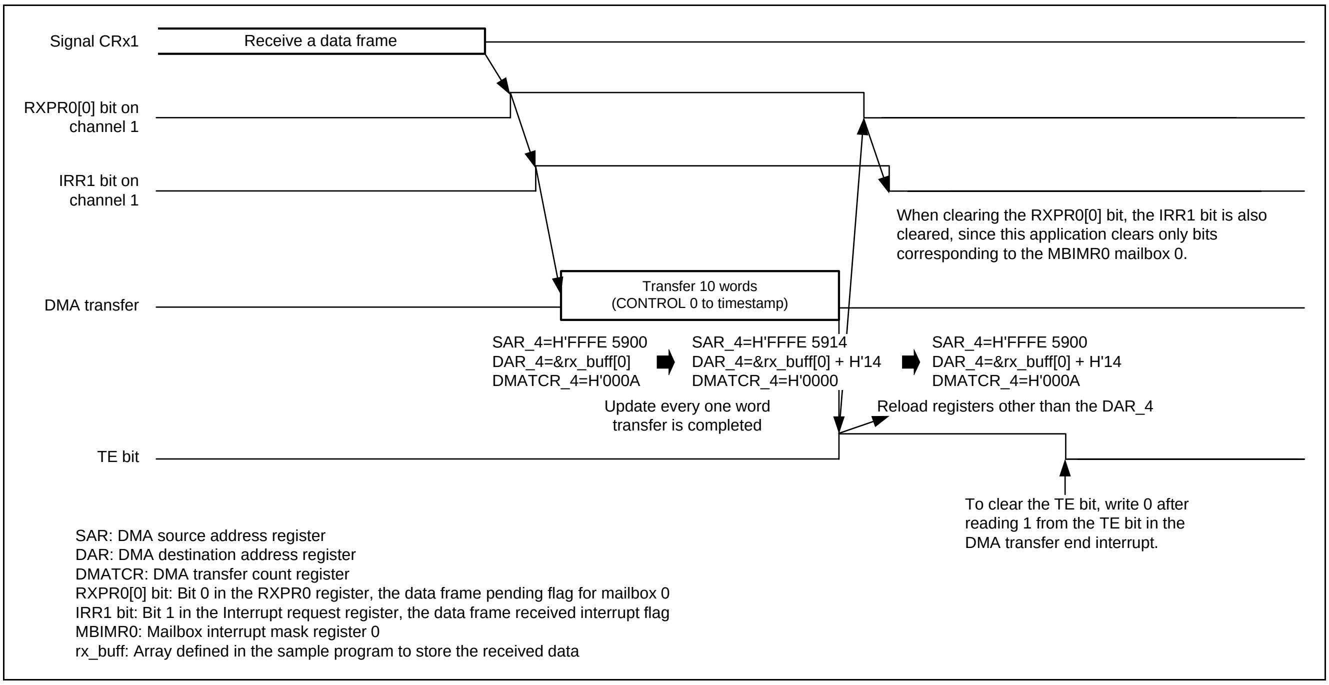
2.4 Sample Program Operation

This sample program uses the Controller Area Network channel 1 to receive a standard CAN data frame with identifier H'00A in mailbox 0. Activate the Direct Memory Access Controller by the data frame received interrupt (IRR1) to transfer data in mailbox 0 to the large-capacity internal 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, and DMA transfer count register to default values by its reload function. As the Direct Memory Access Controller stores the received data in contiguous area in sequence, it does not reload the DMA destination address register. Clear the TE flag using the transfer end interrupt by the Direct Memory Access Controller and prepare for the next data frame received interrupt. When there is no space in the receive buffer variable, the sample program disables the Direct Memory Access Controller.

When specifying the cache-enabled area as the transfer destination, disable entries in cache as appropriate to maintain the coherency between the cache and memory.

Figure 8 shows the sample program operation (overview).

Note: The sample program transmits and receives data frames; however, this application note describes only the reception.



When clearing the RXPR0[0] bit, the IRR1 bit is also cleared, since this application clears only bits corresponding to the MBIMR0 mailbox 0.

To clear the TE bit, write 0 after reading 1 from the TE bit in the DMA transfer end interrupt.

SAR: DMA source address register
DAR: DMA destination address register
DMATCR: DMA transfer count register
RXPR0[0] bit: Bit 0 in the RXPR0 register, the data frame pending flag for mailbox 0
IRR1 bit: Bit 1 in the Interrupt request register, the data frame received interrupt flag
MBIMR0: Mailbox interrupt mask register 0
rx_buff: Array defined in the sample program to store the received data

Figure 8 Sample Program Operation (Overview)

2.5 Sample Program Procedure

Table 4 lists setting examples of the Controller Area Network. Table 5 lists setting examples of the Direct Memory Access Controller. Figure 9 shows the configuration flow chart of this sample program.

Note: The sample program transmits and receives data frames, however, this application note describes only the reception.

Table 4 Register Settings

Register Name	Address	Setting	Description
Standby control register 5 (STBCR5)	H'FFFE 0410	H'FB	MSTP52 = "0": Controller Area Network channel 1 is operating
Master control register (MCR)	H'FFFF 5800	H'0001	MCR0 = "1": Reset mode transition request
		H'8001	MCR15 = "1": The order of the RCAN-ET message and of the HCAN2 message are different
		H'8000	MCR0 = "0": Reset mode is cleared
Interrupt mask register (IMR)	H'FFFF 580A	H'FFFD	Enable the data frame received interrupt (IRR1)
Mailbox interrupt mask register 0 (MBIMR0)	H'FFFE 5852	H'FFFE	MBIMR0[0] = "0": Enable the receive interrupt in mailbox 0
Bit configuration register 1 (BCR1)	H'FFFF 5804	H'9600	TSG1 [3:0] = "B'1001": PRSEG + PHSEG1 = 10 Tq TSG2 [2:0] = "B'110": PHSEG2 = 7 Tq SJW="0": SJW = 1 Tq BSP = "0": Bit sampling at one point
Bit configuration register 0 (BCR0)	H'FFFF 5806	H'0000	BRP [7:0] = "0": 1 Tq = 2 x Pφ
Message control field 1 in mailbox 0 (MB[0].CONTROL1)	H'FFFF 5910	H'0200	MBC [2:0] = "B'010": Receiving the data frame and remote frame are enabled

Message control field 0 in mailbox 0 (MB[0].CONTROL0)	H'FFFF 5900	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 5904	H'0000 0000	Clear: Mask is not specified

Table 5 Direct Memory Access Controller Setting

Register Name	Address	Setting	Description
Standby control register_2 (STBCR_2)	H'FFFE 0018	H'00	MSTP8 = "0": DMAC is operating
DMA source address register_4 (SAR_4)	H'FFFE 1040	H'FFFF 5900	Transfer source starting address: Specify the starting address in mailbox 0
DMA reload source address register_4 (RSAR_4)	H'FFFE 1140	H'FFFF 5900	Address to reload to the SAR_4: Specify the starting address in mailbox 0
DMA destination address register_4 (DAR_4)	H'FFFE 1044	&rx_buff[0]	Transfer destination starting address: Specify the starting address in the receive buffer
DMA transfer count register_4 (DMATCR_4)	H'FFFE 1048	H'0000 000A	Number of transfers: 10
DMA reload transfer count register_4 (RDMATCR_4)	H'FFFE 1148	H'0000 000A	Number of transfers to reload to the DMATCR_4: 10
DMA channel control register_4 (CHCR_4)	H'FFFE 104C	H'0000 0000	DE = "0": Disable the DMA transfer
		H'A000 5A0C	TC = "1" Execute the specified number of transfers of the DMATCR0 by a transfer request RLDSAR = "1": Enable the source address reload function RLDDAR = "0": Disable the destination address reload function DM = "B'01": Increment the destination address SM = "B'01": Increment the source address RS = "B'1010": Specify the Controller Area Network channel 1 as the transfer request source TB = "0": Specify cycle steal mode TS = "B'01": Specify to transfer data in words IE = "0": Enable the interrupt request
		H'A000 5A0D	DE = "1": Enable the DMA transfer
DMA operation register (DMAOR)	H'FFFE 1200	H'0001	DME = Enable the DMA transfer on all channels
Interrupt priority register 07 (IPR07)	H'FFFE 0C02	H'A000	Transfer end interrupt priority: Specify level 10

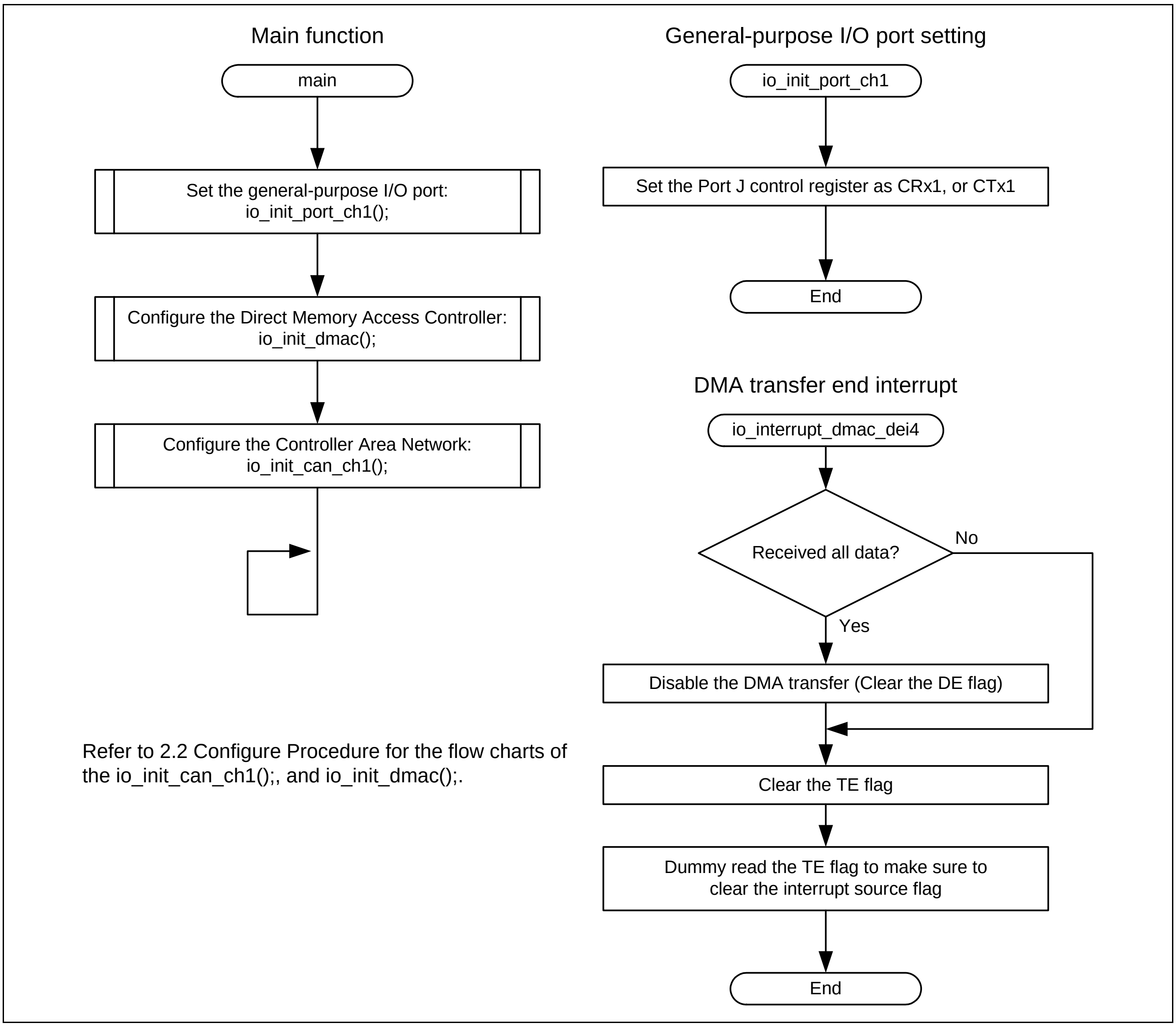

Main function
main
Set the general-purpose I/O port:
io_init_port_ch1();
Configure the Direct Memory Access Controller:
io_init_dmac();
Configure the Controller Area Network:
io_init_can_ch1();
Refer to 2.2 Configure Procedure for the flow charts of the io_init_can_ch1();, and io_init_dmac();.
General-purpose I/O port setting
io_init_port_ch1
Set the Port J control register as CRx1, or CTx1
End
DMA transfer end interrupt
io_interrupt_dmac_dei4
Received all data?
No
Yes
Disable the DMA transfer (Clear the DE flag)
Clear the TE flag
Dummy read the TE flag to make sure to
clear the interrupt source flag
End

Figure 9 Sample Program Flow Chart

3. Sample Program Listing

3.1 Supplement to the Sample Program

As the capacity of the SH7264 large-capacity internal RAM varies as 1 MB or 640 KB, depending on the MCU type, the section alignment and register setting must be partly altered. To support both MCU types, this application note provides two types of sample programs (workspaces) for 1-MB RAM and 640-KB RAM.

As the MCU with 640-KB RAM must be write-enabled before writing data in the data-retention RAM, the System control register 5 (SYSCR5) is set to write-enable the RAM in the sample program for 640-KB RAM.

Review your product and use the appropriate workspace.

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

```
/******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
********************************************************************************
* (C) 2010 Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data ***************************
*   System Name : SH7264 Sample Program
*   File Name   : main.c
*   Abstract    : DMAC+CAN Module Application (Data Frame Transmit and Receive)
*   Version     : 1.00.01
*   Device      : SH7262/SH7264
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: M3A-HS64G50(CPU board)+M3A-HS64G01(IO board)
*   Description :
********************************************************************************
*   History     : Feb.09,2010 ver.1.00.00
*               : Apr.15,2010 Ver.1.00.01 Changed the company name
*""FILE COMMENT END""**********************************************************/
#include "iodefine.h"       /* SH7264 iodefine  */

```

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

```
/* ---- Prototype declaration ---- */
void main(void);
void io_interrupt_dmac_dei4(void);
extern void io_init_port_ch0(void);
extern void io_init_port_ch1(void);
extern void io_init_can_ch0(void);
extern void io_init_can_ch1(void);
extern void io_init_dmac(void *src, void *dst, int count);
extern void io_data_send_ch0(unsigned char *addr, int size);

/* ---- Macro definition ---- */
#define NUM_OF_BUFF       10          /* Number of faces in the receive buffer */

/* ---- Type definition ---- */
typedef struct{
  unsigned long    control0;      /* Message control field 0 */
  unsigned long    lafm;          /* Local acceptance filter mask */
  unsigned char    data[8];       /* Data */
  unsigned short   control1;      /* Message control field 1 */
  unsigned short   timestamp;     /* Timestamp */
}MB_TYPE;

/* ---- Variable definition ---- */
MB_TYPE        rx_buff[ NUM_OF_BUFF ];    /* Receive buffer */
unsigned char  tx_data[8] = {0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7};
                                  /* Transmit data */

```

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

```
/*""FUNC COMMENT""**************************************************************
 * ID            :
 * Outline       : Sample program main
 *------------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration   : void main(void);
 *------------------------------------------------------------------------------
 * Description   : After configuring the Controller Area Network (RCAN), channel 0
 *               : transmits a data frame, and channel 1 receives the data frame.
 *               : It uses the Direct Memory Access Controller (DMAC) to transfer
 *               : the received data.
 *------------------------------------------------------------------------------
 * Argument      : void
 *------------------------------------------------------------------------------
 * Return Value  : void
 *------------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""**********************************************************/
void main(void)
{
  int i;
  unsigned char *p;
  int idx, sz;

    /* ==== Configures the general-purpose I/O port ==== */
    io_init_port_ch1();
    io_init_port_ch0();

  /* ==== Configures the DMAC (channel 1) ==== */
  io_init_dmac( &RCAN1.MB[0], rx_buff, sizeof(MB_TYPE) / sizeof(short) );
                          /* Transfers the received data in channel 1
                           mailbox 0 to the internal RAM */

  /* ==== Configures the CAN ==== */
    io_init_can_ch1();
    io_init_can_ch0();

  /* ==== Transmits data (channel 0) ==== */
  for( i=0; i<NUM_OF_BUFF; i++ ){
    io_data_send_ch0( tx_data, 8 );
  }

    while(1){
    /* wait */
    }
}
```

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

```
/*""FUNC COMMENT""**************************************************************
 * ID            :
 * Outline       : DMAC setting
 *------------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration   : void io_init_dmac(void *src, void *dst, int count);
 *------------------------------------------------------------------------------
 * Description   : Configure the Direct Memory Access Controller (DMAC).
 *               : Specify cycle steal mode, dual address mode, an on-chip peripheral
 *               : module request (RCAN). Specify the transfer data length in words,
 *               : RCAN mailbox 0 as the source, and internal RAM as the destination.
 *               : The reload function is used to the source address.
 *------------------------------------------------------------------------------
 * Argument      : void *src ; I : Transfer source address
 *               : void *dst ; O : Transfer destination address
 *               : int count ; I : Number of transfers
 *------------------------------------------------------------------------------
 * Return Value  : void
 *------------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""***********************************************************/
void io_init_dmac(void *src, void *dst, int count)
{
  /* ==== Sets the Standby control register 2 ==== */
  CPG.STBCR2.BIT.MSTP8 = 0;               /* Clears the DMAC module standby */

  /* ==== Disables transfer on DMA_channel 4 ==== */
  DMAC.CHCR4.BIT.DE = 0x0;            /* Disables the DMA transfer */

  /* ==== Sets the DMA source address register_4 (SAR_4) ==== */
  DMAC.SAR4.LONG = (unsigned long)src;

  /* ==== Sets the DMA reload source address register_4 (RSAR_4) ==== */
  DMAC.RSAR4.LONG = (unsigned long)src;

  /* ==== Sets the DMA destination address register_4 (DAR_4) ==== */
  DMAC.DAR4.LONG = (unsigned long)dst;

  /* ==== Sets the DMA transfer count register_4 (DMATCR_4) ==== */
  DMAC.DMATCR4.LONG = count;

  /* ==== Sets the DMA reload transfer count register_4 (RDMATCR_4) ==== */
  DMAC.RDMATCR4.LONG = count;

```

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

```
  /* ==== Sets the DMA channel control register_4 (CHCR_4) ==== */
  DMAC.CHCR4.LONG = 0xa0005a0c;
               /*
                       bit 31    : TC DMATCR transfer: 1---- Executes the specified number
                                                                 of transfers of the DMATCR
                       bit 30    : reserve 0
                       bit 29    : RLDSAR ON : 1------------ Enables the reload function (RSAR)
                       bit 28    : RLDDAR ON : 0------------ Disables the reload function
                                                             (RDAR)
                       bit 27    : reserve 0
                       bit 26    : DAF: 0------------------- Not used
                       bit 25    : SAF: 0------------------- Not used
                       bit 24    : reserve 0
                       bit 23    : DO over run0 : 0--------- Not used
                       bit 22    : TL TEND low active : 0--- Not used
                       bit 21    : reserve 0
                       bit 20    : TEMASK :0---------------- Not used
                       bit 19    : HE :0-------------------- Not used
                       bit 18    : HIE :0------------------- Not used
                       bit 17    : AM :0-------------------- Not used
                       bit 16    : AL :0-------------------- Not used
                       bit 15 to 14: DM1:0 DM0:1------------ Increments the destination
                                                             address
                       bit 13 to 12: SM1:0 SM0:1------------ Increments the source address
                       bit 11 to 8 : RS : B'1010------------ Transfer request is from
                                                             RCAN channel 1
                       bit 7     : DL : DREQ level : 0 ----- Not used
                       bit 6     : DS : DREQ select :0 Low level Not used
                       bit 5     : TB :cycle :0------------- Cycle steal mode
                       bit 4 to 3: TS : transfer size: B'01- Transfers data in words
                       bit 2     : IE : interrupt enable: 1- Enables interrupt
                       bit 1     : TE : transfer end: 0----- Clears the TE flag
                                                               (Clear the flag to 0 after
                                                               reading 1)
                       bit 0     : DE : DMA enable bit: 0--- Disables the DMA transfer
               */
  /* ======= Sets the Interrupt priority register 07 (IPR07) ======= */
  INTC.IPR07.BIT._DMAC4 = 0xa;

  /* ---- Sets the DMA operation register (DMAOR)---- */
  DMAC.DMAOR.WORD |= 0x0007;              /* Sets the DME bit. To avoid clearing */
                                      /* address error and NMI flags, write 1 */
                                      /* to the AE bit and NMIF flag */
  /* ======= Enable transfer on DMA_channel 4 ======= */
  DMAC.CHCR4.BIT.DE = 0x1;

}
```

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

```
/*""FUNC COMMENT""**************************************************************
 * ID               :
 * Outline          : Sample program main
 *------------------------------------------------------------------------------
 * Include          : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration      : void io_interrupt_dmac_dei4(void);
 *------------------------------------------------------------------------------
 * Description      : Handles the transfer end interrupt on DMAC channel 4.
 *                  : Use the reload register to receive data repeatedly.
 *                  : When transferring buffer size of data is completed, the DMA is
 *                  : is stopped.
 *------------------------------------------------------------------------------
 * Argument         : void
 *------------------------------------------------------------------------------
 * Return Value     : void
 *------------------------------------------------------------------------------
 * Note             :
 *""FUNC COMMENT END""**********************************************************/
void io_interrupt_dmac_dei4( void )
{
  volatile int dummy;

  /* ==== When the buffer size of data is transferred, the DMA is stopped ==== */
  if( DMAC.DAR4.LONG >= (unsigned long)&rx_buff[ NUM_OF_BUFF ] ){
    DMAC.CHCR4.BIT.DE = 0;     /* Disables the transfer */
  }
  /* ==== Clears the TE flag ==== */
  DMAC.CHCR4.BIT.TE = 0;

  /* ==== Execute dummy read to make sure to clear the interrupt source flag ==== */
  dummy=DMAC.CHCR4.LONG;

}
/* End of File */
```

3.8 Sample Program Listing "can1.c" (1/4)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
********************************************************************************
* (C) 2010 Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7264 Sample Program
*   File Name   : can1.c
*   Abstract    : DMAC+CAN Module Application (Data Frame Receive)
*   Version     : 1.00.01
*   Device      : SH7262/SH7264
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: M3A-HS64G50(CPU board)+M3A-HS64G01(IO board)
*   Description :
********************************************************************************
*   History     : Feb.09,2010 ver.1.00.00
*               : Apr.15,2010 Ver.1.00.01 Changed the company name
*""FILE COMMENT END""**********************************************************/
#include "iodefine.h"       /* SH7264 iodefine  */

```

3.9 Sample Program Listing "can1.c" (2/4)

```
/* ---- prototype declaration ---- */
void io_init_port_ch1(void);
void io_init_can_ch1(void);

/* ---- symbol definition ---- */
#define CAN_IRR0 0x0001

/*""FUNC COMMENT""**************************************************************
 * ID            :
 * Outline       : General-purpose I/O port configuration
 *------------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration   : void io_init_port_ch1(void);
 *------------------------------------------------------------------------------
 * Description   : Set pin functions (CRx1 input, CTx1 output).
 *------------------------------------------------------------------------------
 * Argument      : void
 *------------------------------------------------------------------------------
 * Return Value  : void
 *------------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""**********************************************************/
void io_init_port_ch1(void)
{
    /* ==== Setting of PORT ==== */
  PORT.PJCR0.BIT.PJ2MD = 0x1;     /* Set CTx1 */
    PORT.PJCR0.BIT.PJ3MD = 0x1;     /* Set CRx1 */
}

```

3.10 Sample Program Listing "can1.c" (3/4)

```
/*""FUNC COMMENT""**************************************************************
 * ID            :
 * Outline       : RCAN configuration
 *------------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration   : void io_init_can_ch1(void);
 *------------------------------------------------------------------------------
 * Description   : Configure the Controller Area Network (RCAN) channel 1.
 *               : Transfer speed is set as 1 Mbps. Enable the interrupt mask to
 *               : allow the Direct Memory Access Controller to transfer received
 *               : data in mailbox 0.
 *------------------------------------------------------------------------------
 * Argument      : void
 *------------------------------------------------------------------------------
 * Return Value  : void
 *------------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""*********************************************************/
void io_init_can_ch1(void)
{
    int i,j;

  /* ==== Sets the Standby control register 5 ==== */
  CPG.STBCR5.BIT.MSTP52 = 0;              /* Clears RCAN channel 1 module standby */

  /* ==== Sets the Master control register ==== */
    RCAN1.MCR.WORD |= 0x0001;             /* Sets the reset mode */
    while((RCAN1.IRR.WORD & CAN_IRR0) != CAN_IRR0){
    /* Waits for completing transition to reset mode */
    }
  /* ==== IRR = 1, GSR = 1 (set automatically) ==== */

  /* ---- Clears the reset interrupt flag ---- */
    RCAN1.IRR.WORD = 0x0001;

  /* ---- Sets the Master control register ---- */
    RCAN1.MCR.WORD |= 0x8000;             /* RCAN is not same as HCAN2 */

  /* ---- Sets the Interrupt mask register ---- */
    RCAN1.IMR.WORD  = 0xfffd;

  /* ---- Sets Mailbox interrupt mask register 0 ---- */
  RCAN1.MBIMR0.BIT.MB0 = 0;

```

3.11 Sample Program Listing "can1.c" (4/4)

```
  /* ----Clears mailbox RAM ---- */
    for(i = 0; i < 32; i++){
        RCAN1.MB[i].CONTROL0.LONG = 0x00000000;
        RCAN1.MB[i].LAFM.LONG = 0x00000000;
        for(j = 0; j < 8; j++){
            RCAN1.MB[i].MSG_DATA[j] =  0x00;
        }
    }
  /* ---- Sets mailbox 0 ---- */
    RCAN1.MB[0].CONTROL1.WORD = 0x0200;      /* MBC = 2, dlc = 0  */
    RCAN1.MB[0].CONTROL0.LONG = 0x00280000;  /* Standard data frame, id=0x00a */
    RCAN1.MB[0].LAFM.LONG = 0x00000000;
    for(i = 0; i < 8; i++){                  /* Clears data */
        RCAN1.MB[0].MSG_DATA[i] =  0x00;
    }
  /* ---- Sets the Bit configuration register ---- */
    RCAN1.BCR1.WORD = 0x9600;         /* tsg1 = 9 (10-bit), tsg2 = 6 (7-bit), sjw = 0 (1-bit),
                                   bsp = 0 */
    RCAN1.BCR0.WORD = 0x0000;         /*   1 Mbps */

  /* ---- Sets the Interrupt request register ---- */
    RCAN1.IRR.WORD = 0xffff;

  /* ---- Sets the Master control register ---- */
    RCAN1.MCR.WORD &= 0xf8fc;      /* Clears MCR0, and MCR1 */
    while( (RCAN1.GSR.WORD & 0x0008) != 0x0000 ){
        /* Reset state is end */
    }
}

/* End of File */
```

4. References

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

- Hardware Manual
  SH7262 Group, SH7264 Group Hardware Manual Rev. 2.00
  The latest version of the hardware manual can be downloaded from the Renesas website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Apr.28.10	—	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