To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

H8S/2200 Series

DTC Transfer (Activation by 8-Bit Timer Channel 0A Interrupt, Examples of Using Chain Function)

Introduction

Transfers data on an SRAM chip to other addresses on the chip with the DTC that is activated by the 8-bit timer channel 0A interrupt. It also transfers data continuously to other addresses on the SRAM chip with the DTC chain transfer function.

Target Device

H8S/2215

Contents

1. Overview

The H8S/2215 transfers data on an SRAM to other addresses with the DTC that is activated by the 8-bit timer channel 0A interrupt. It also transfers data continuously to other addresses on the SRAM with the DTC chain transfer function.

2. Configuration

Figure 1 shows the configuration of the confirmed operation of this application note.

List of Components Used

No.	Component	Specifications
1	Solution Engine H8S/2215 CPU board (Manufactured by Hitachi ULSI Systems)	Board power supply input: 5 VDC Operating frequency: 16 MHz MCU operating mode: 6 SRAM (128k × 16 bits)

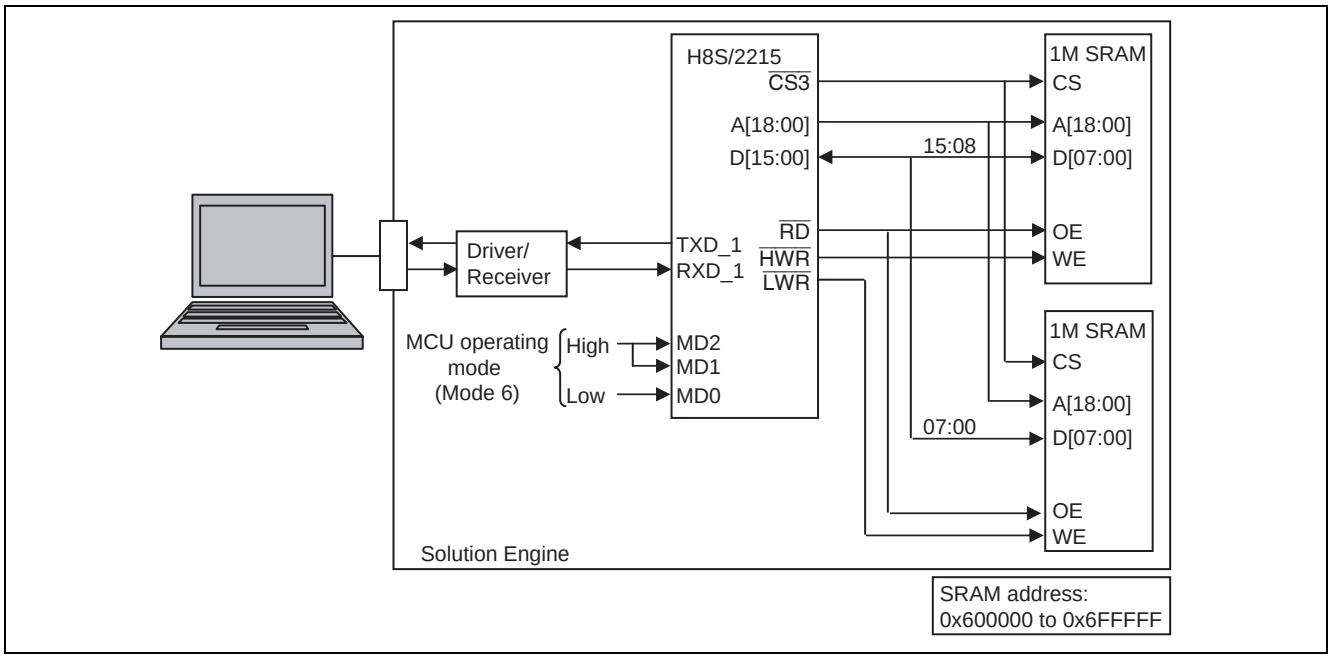


Figure 1 Confirmed Configuration

3. Description of Functions

The DTC is started by inputting a command from the terminal software connected to the RS-232C interface. Moreover, a function for reading from and writing to memory is provided for debugging.

1. Communication specifications
   The terminal software is connected according to the specifications below.

Communication method	Asynchronous
Bit rate	38400 bps
Data size	8 bits
Parity	None
Stop bit	1 bit
Terminating code	Line feed

2. Supported functions

No.	Function	Specifications
1	Data transfer with the DTC activated by the 8-bit timer channel 0A interrupt (Chain transfer mode) Transfers data at 0x600000 - 0x60000F to 0x610000 - 0x61000F and 620000 to 0x62000F on an SRAM chip.	Format: dt
2	Read the 64 byte data from a specific memory address for display.	Format: mrΔaddress (Specify a 4-byte value in address.)
3	Write the 4 byte data from a specific memory address.	Format: mwΔaddressΔdata (Specify address in the length of four bytes.)

4. Principles of Operation

4.1 Initialization Processing

Before exercising DTC control, start up the microcomputer and perform operations such as internal register initial setting.

1. Control for low power consumption, clock oscillator initialization

Register name ←Set value	Bit	Name	Value	Contents
LPWCR	7-4	----	0000	
←0x03	3	RFCUT	0	Uses the internal feedback resistance control.
	2	----	0	
	1:0	STC[1:0]	11	Bypasses PLL.
MSTPCRA	7	MSTPA7	0	Operates the DMAC module.
←0x0D	6	MSTPA6	0	Operates the DTC module.
	5	MSTPA5	0	Operates the TPU module.
	4	MSTPA4	0	Operates the TMR module.
	3:2	MSTPA[3:2]	11	(Reserved)
	1	MSTPA1	0	Operates the AD module.
	0	MSTPA0	1	(Reserved)
MSTPCRB	7	MSTPB7	0	Operates the SCI0 module.
←0x1F	6	MSTPB7	0	Operates the SCI1 module.
	5	MSTPB7	0	Operates the SCI2 module.
	4:1	MSTPB[4:1]	1111	(Reserved)
	0	MSTPB0	1	Stops the USB module.
MSTPCRC	7:6	MSTPC[7:6]	11	(Reserved)
←0xDF	5	MSTPC5	0	Operates the DA module.
	4:0	MSTPC[4:0]	11111	(Reserved)

2. IO port initialization
   Set the input/output pins of port G as indicated below. Set all other pins to the output mode.

Port	Register name ←Set value	Bit	Name	Value	Contents
G	PGDDR	7:5	---	11	
	←0xFF	4:2	PG[4:2]DDR	111	Output (Not used)
		1	PG1DDR	1	Output (CS3 enabled)
		0	PG0DDR	1	Output (Not used)

3. Bus controller initialization
   Set the bus so that the externally connected SRAM (128k × 16 bits) can be accessed.

Register name ←Set value	Bit	Name	Value	Contents
ABWCR	7	ABW7	0	Area 7: 16 bits of bus width (not used)
←0x77	6:4	ABW[6:4]	111	Areas 6 to 4: 8 bits of bus width (not used)
	3	ABW3	0	Area 3: 16 bits of bus width (SRAM)
	2:0	ABW[2:0]	111	Areas 2 to 0: 8 bits of bus width (not used)
PFCR	7:4	----	0	
←0x0F	3:0	AE[3:0]	1111	Enables A23:00 output.

4. TPU0 timer initialization
   Make settings so that a timer interrupt is generated at intervals of 100 ms for timer monitoring.

Register name ←Set value	Bit	Name	Value	Contents
TCR_0 ←0x23	7:5	CCLR[2:0]	001	Counter clear on the TGRA compare match
	4:3	CKEG[1:0]	00	Count on rising edge
	2:0	TPSC[2:0]	011	Count with ϕ/64
TMDR_0 ←0x00	7:6	----	00	
	5	BFB	0	Normal TGRB operation
	4	BFA	0	Normal TGRA operation
	3	----	00	
	2:0	MD[2:0]	000	Normal operation
TIORH_0 ←0x00	7:4	IOB[3:0]	0000	TBRB output compare (Not used)
	3:0	IOA[3:0]	0000	TBRA output compare
TIORL_0 ←0x00	7:4	IOD[3:0]	0000	TBRD output compare (Not used)
	3:0	IOC[3:0]	0000	TBRC output compare (Not used)
TIER_0 ←0x00	7	TTGE	0	Disables the AD conversion start request. (Not used)
	6	----	0	
	5	TCIEU	0	Disables underflow interrupt. (Not used)
	4	TCIEV	0	Disables overflow interrupt. (Not used)
	3	TGIED	0	Disables the TGRD interrupt. (Not used)
	2	TGIEC	0	Disables the TGRC interrupt. (Not used)
	1	TGIEB	0	Disables the TGRB interrupt. (Not used)
	0	TGIEA	0	Enables the TGRA interrupt.
TGRA_0 ←25000	15:0	TGRA_0	25000	TBRA output compare value (Set the timer to generate at every 100 ms.)
TCNT_0 ←0x0000	15:0	TCNT_0	0x0000	Counter clear
TSTR ←0x01	7:0	TSTR	0x01	TCNT_0 counter start

5. Serial interface (SCI_1) initialization
   Make settings to connect the terminal software for starting the DTC.

Register name ←Set value	Bit	Name	Value	Contents
SCR_1 ←0x00	7	TIE	0	Disables transmit interrupt.
	6	RIE	0	Disables receive interrupt.
	5	TE	0	Disables transmit operation.
	4	RE	0	Disables receive operation.
	3	MPIE	0	Disables the multi-processor interrupt.
	2	TEIE	0	Disables transmit end interrupt.
	1:0	CKE[1:0]	00	Asynchronous, internal clock used
	7	TIE	0	Disables transmit interrupt.
SMR_1 ←0x00	7	C/A	0	Asynchronous mode
	6	CHR	0	8-bit length
	5	PE	0	No parity check.
	4	O/E	0	Even parity (Not used)
	3	STOP	0	One stop bit
	2	MP	0	Disables the multi-processor communication function.
	1:0	CKS[1:0]	00	Clock source = ϕ
SCMR_1 ←0x00	7:4	----	0000	
	3	DIR	0	LSB first
	2	INV	0	No data inversion
	1-0	----	00	
BRR ←12	7:0	BRR	12	Sets the transmission speed to 38400 bps.

— The system waits at least one stop bit period. (38400 bps: About 30 μs)
— Set the receive processing to enable.

Register name (Address←Set value)	Bit	Name	Value	Contents
SCR_1 ←0x50	7	TIE	0	Disables transmit interrupt.
	6	RIE	1	Enables receive interrupt.
	5	TE	0	Disables transmit operation.
	4	RE	1	Enables receive operation.
	3	MPIE	0	Disables the multi-processor interrupt.
	2	TEIE	0	Disables transmit end interrupt
	1:0	CKE[1:0]	00	Asynchronous, internal clock used

4.2 DTC Control

4.2.1 Overview of DTC Transfer

The DTC implements data transfer with hardware according to the transfer parameters specified in advance. Efficient data transfer is possible because the DTC can be operated in the background of the program processing.

A DMAC also has similar functions as the DTC. The transfer performance of the DTC is inferior to that of the DMAC but the DTC can simultaneously specify more channels than the DMAC and can also chain channels (chain function). This means that the DTC has a greater capacity than the DMAC. The DTC is activated by an interrupt or directly by software.

This application note explains how the DTC activated by software transfers data.

4.2.2 Data Transfer with DTC Activated by 8-Bit Timer Channel 0A Interrupt

1. Overview
   Activate the DTC by entering the following command from the terminal software connected to SCI_1:
   Format: dt

   The H8S/2215 performs DTC transfer continuously twice with the chain transfer function when the 8-bit timer channel 0A interrupt is triggered. The H8S/2215 uses the normal mode as DTC operating mode. In the normal mode, the DTC transfers data at 0x600000 to 0x60000F to 0x610000 to 0x61000F and 620000 to 0x62000F on an SRAM chip. When data transfer ends in this mode, the 8-bit timer channel 0A interrupt occurs. The following explains how to activate the DTC.

2. Setting the 8-bit timer channel 0A interrupt
   Set the 8-bit timer channel 0A interrupt to occur every 10 milliseconds.
   This interrupt serves as the DTC transfer activation source and data transfer is performed every 10 milliseconds.
   Start the timer after the DTC has been activated.

3. Setting DTC transfer parameters
   First, set parameters to specify how DTC transfer is carried out.
   The DTC transfer parameters must be set in 0xFFEBC0 to 0xFFEFBF of the internal RAM in the following format:
   ch_no specifies the locations of the DTC transfer parameters to be set, so users do not need to directly worry about the internal RAM addresses. The relationship between ch_no and internal RAM addresses is as follows:

   Starting address of DTC transfer parameters = 0xFFEBC0 + ch_no × 12

0	1	2	3
MRA	SAR		
MRB	DAR		
CRA		CRB	

   MRA: DTC mode register A
   MRB: DTC mode register B
   SAR: DTC source address register
   DAR: DTC destination address register
   CRA: Transfer counter register A
   CRB: Transfer counter register B
   For more information about these registers, see the related hardware manuals.

Set the following for ch_no = 0 (internal RAM 0xFFEBC0 and after) and ch_ no = 1 (internal RAM 0xFFEBCC and after).
You can use one subroutine (dtc_set_parm) to set DTC transfer parameters collectively.
For more information about subroutines, see Chapter 5, "Description of Sample Program."

A. Setting transfer parameters for ch_no 0
Normal mode: The DTC transfers data at 0x600000 - 0x60000F to 0x610000 - 0x61000F on an SRAM.

Register Name	Settings	Description
MRA	0xA0	• Increments SAR by one after data transfer. • Increments DAR by one after data transfer. • DTC mode = normal mode • Size of the data to be transferred once = 1 byte
MRB	0x80	• DTC chain available • When transferring all the specified data has ended, generate the 8-bit timer channel 0A interrupt. *1
SAR	0x600000	Specifies a source address.
DAR	0x610000	Specifies a destination address.
CRA	0x0010	Specifies the number of times the DTC transfers data (1 to 65536).
CRB	0x0000	(Not used)

*1 When DTC chain transfer is enabled, no interrupt occurs even if DTC transfer ch_no 0 ends. In this case, DTC transfer where ch_no is 1 is subsequently performed.

B. Setting transfer parameters in the area whose ch_no is 1
Normal mode: The DTC transfers data at 0x600000 to 0x60000F to 0x620000 to 0x62000F on an SRAM chip.

Register Name	Settings	Description
MRA	0xA0	• Increments SAR by 1 after data transfer. • Increments DAR by 1 after data transfer. • DTC mode = normal mode • Data size per transfer = one byte
MRB	0x00	• DTC chain available • When transferring all the specified data has ended, generate the 8-bit timer channel 0A interrupt.
SAR	0x600000	Specifies a source address.
DAR	0x610000	Specifies a destination address.
CRA	0x0010	Specifies the number of times the DTC transfers data (1 to 65536).
CRB	0x0000	(Not used)

4. Setting DTC vector addresses
The DTC has a vector address corresponding to an activation source. Each DTC vector address has a 2-byte area where the lower two bytes of the starting address of the DTC transfer parameters set in 2. are set. When the DTC is activated, the DTC transfer parameters are read from the corresponding vector address and the DTC transfer is activated.
In an H8 microcomputer, the vector address area is a ROM area, so the lower 2 bytes of the starting address of the DTC transfer parameters must be initially set in the vector address area for compilation.
The vector address for interrupt activation differs according to the interrupt source. The vector address is 0x480 because the 8-bit timer channel 0A interrupt is used as the interrupt source.
The lower 2 bytes (0xEBC0) of the starting address of the DTC transfer parameters must be initially set in advance at this vector address for compilation.
The table below lists a vector address for interrupt sources.

Correspondence between DTC Activation Sources, DTC Vector Addresses, and DTCEs

DTC Activation Source Occurrence Source	DTC Activation Source (Interrupt Source)	Vector No.	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + DTVECR[6:0] × 2	–	High
External pin	IRQ0	16	H'0420	DTCEA7	
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
	IRQ4	20	H'0428	DTCEA3	
	IRQ5	21	H'042A	DTCEA2	
	IRQ7	23	H'042E	DTCEA0	
A/D	ADI	28	H'0438	DTCEB6	
TPU channel 0	TGI0A	32	H'0440	DTCEB5	
	TGI0B	33	H'0442	DTCEB4	
	TGI0C	34	H'0444	DTCEB3	
	TGI0D	35	H'0446	DTCEB2	
TPU channel 1	TGI1A	40	H'0450	DTCEB1	
	TGI1B	41	H'0452	DTCEB0	
TPU channel 2	TGI2A	44	H'0458	DTCEC7	
	TGI2B	45	H'045A	DTCEC6	
8-bit timer channel 0	CMIA0	64	H'0480	DTCED3	
	CMIB0	65	H'0482	DTCED2	
8-bit timer channel 1	CMIA1	68	H'0488	DTCED1	
	CMIB1	69	H'048A	DTCED0	
DMAC	DEND0A	72	H'0490	DTCEE7	
	DEND0B	73	H'0492	DTCEE6	
	DEND1A	74	H'0494	DTCEE5	
	DEND1B	75	H'0496	DTCEE4	
SCI channel 0	RXI0	81	H'04A2	DTCEE3	
	TXI0	82	H'04A4	DTCEE2	
SCI channel 1	RXI1	85	H'04AA	DTCEE1	
	TXI1	86	H'04AC	DTCEE0	
SCI channel 2	RXI2	89	H'04B2	DTCEF7	
	TXI2	90	H'04B4	DTCEF6	Low

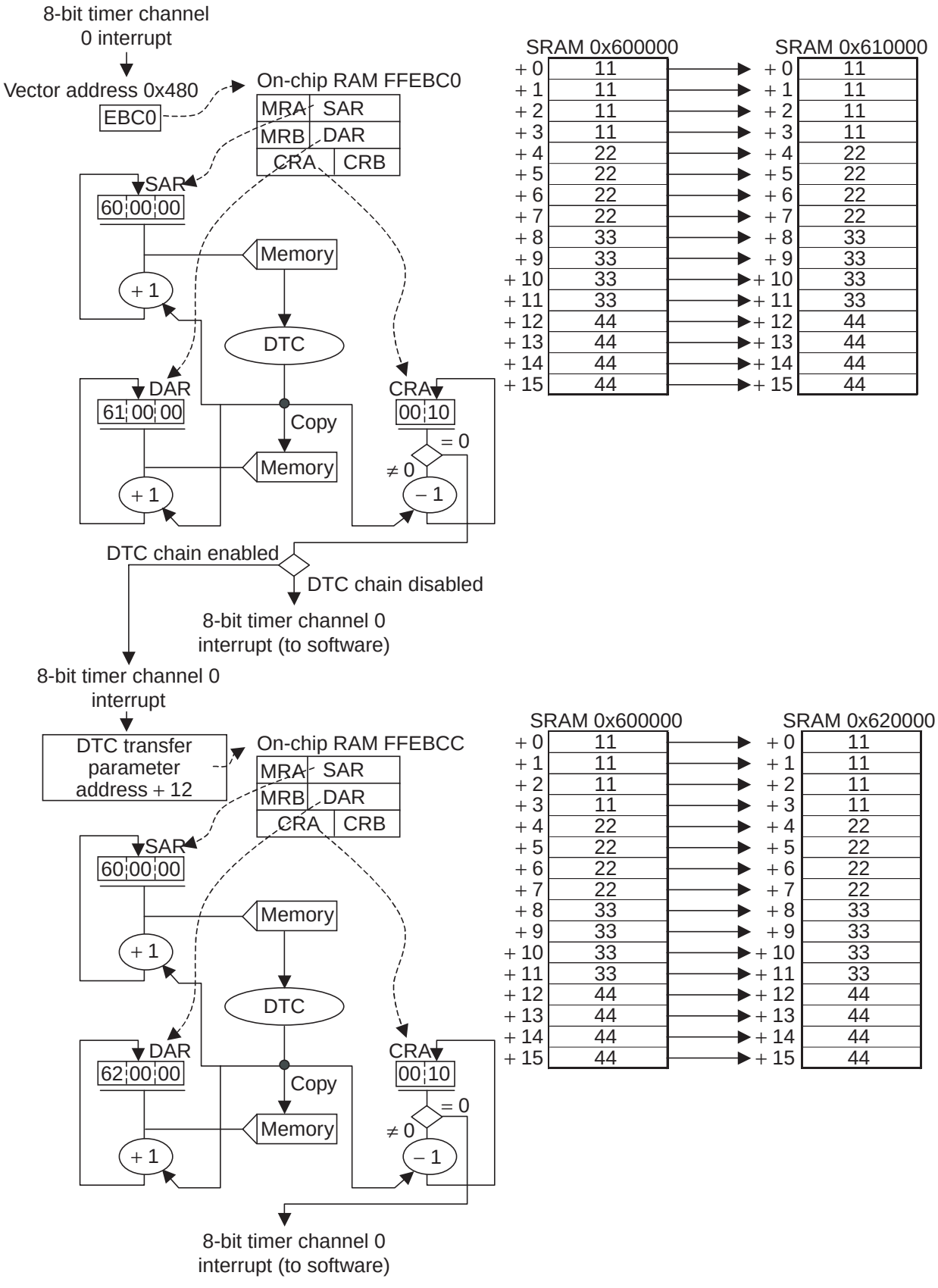
* The DTCE bits without the corresponding interrupts serve as the reserved bit. Write 0 to these bits.

5. DTC activation
   For interrupt activation, set the bits corresponding to the interrupt sources in the DTCERA to DTCERF registers to "1."
   Set bit 3 (DTCED3) in the DTCERD register to "1" because the 8-bit timer channel 0A interrupt is used as the DTC activation source. For information about interrupt sources and registers, see the previous page.
   An activation source is specified in the dtc_start subroutine, so registers are transparent to users. For more information about this subroutine, see Chapter 5, "Description of Sample Program."

6. DTC operation
   DTC transfer is performed based on the above settings. The behavior of DTC transfer depends on the DISEL bit setting in the MRB register.
   When the DISEL bit is 0 (0 is set in this application note)
   Data transfer is performed each time an interrupt source occurs. The interrupt routine is not executed because the interrupt source is reset by the DTC. When data transfer for ch_no 0 ends, data transfer for ch_no 1 is consecutively started. When all data transfers end, the bit corresponding to the interrupt source in the DTCERA to DTCERF registers is reset to "0" and the interrupt routine is executed.
   When the DISEL bit is 1
   The interrupt routine is executed each time data transfer is performed. At this time, the bit corresponding to the interrupt source in the DTCERA to DTCERF registers is reset to "0." To restart data transfer, you must set this bit "1" again.
   Also when all data transfers end, the interrupt routine is executed and the bit corresponding to the interrupt source in the DTCERA to DTCERF registers is reset to "0." Since there is only one interrupt type, you cannot determine by the interrupt itself whether all data transfers have ended. For this reason, when using this mode, you must prepare a software counter, perform addition/subtraction during interrupt processing, and monitor how many times data transfer has occurred.

   The following shows the DTC transfer flow in the chain transfer mode in this application note (DISEL = 0).

7. DTC transfer flow
   A. When an 8-bit timer channel 0A interrupt occurs, the DTC reads the DTC transfer parameters (12 bytes from 0xFFEBC0) from the vector address 0x480.
   B. The DTC copies the contents of the source address indicated by SAR to the destination address indicated by DAR.
   C. The DTC increments the contents of SAR and DAR by one according to the transfer parameters, decrements the transfer count indicated by CRA by 1, and resets the 8-bit timer channel 0A interrupt source.
   D. B. and C. are repeated till CRA becomes 0.
   E. When CRA becomes 0, DTC transfer parameters are read from the area whose ch_no is 1 (12 bytes from 0xFFEBC0) and DTC transfer is continued.
   F. The DTC copies the contents of the source address indicated by SAR to the destination address indicated by DAR.
   G. The DTC increments the contents of SAR and DAR by one according to the transfer parameters, decrements the number of transfers indicated by CRA by one, and resets the 8-bit timer channel 0A interrupt source.
   H. F. and G. are repeated till CRA becomes 0.
   I. When CRA becomes 0, the bit corresponding to the interrupt source in the DTCERA to DTCERF registers is reset to "0" and the 8-bit timer channel 0A interrupt routine is executed to end DTC transfer.


8-bit timer channel 0 interrupt
Vector address 0x480
EBC0
On-chip RAM FFEBC0
MRA
SAR
MRB
DAR
CRA
CRB
SAR
60 00 00
Memory
+ 1
DTC
DAR
61 00 00
Copy
CRA
00 10
= 0
≠ 0
Memory
+ 1
– 1
DTC chain enabled
DTC chain disabled
8-bit timer channel 0 interrupt (to software)
SRAM 0x600000
SRAM 0x610000
+ 0 11
+ 1 11
+ 2 11
+ 3 11
+ 4 22
+ 5 22
+ 6 22
+ 7 22
+ 8 33
+ 9 33
+ 10 33
+ 11 33
+ 12 44
+ 13 44
+ 14 44
+ 15 44
8-bit timer channel 0 interrupt
DTC transfer parameter address + 12
On-chip RAM FFEBCC
MRA
SAR
MRB
DAR
CRA
CRB
SAR
60 00 00
Memory
+ 1
DTC
DAR
62 00 00
Copy
CRA
00 10
= 0
≠ 0
Memory
+ 1
– 1
8-bit timer channel 0 interrupt (to software)
SRAM 0x600000
SRAM 0x620000
+ 0 11
+ 1 11
+ 2 11
+ 3 11
+ 4 22
+ 5 22
+ 6 22
+ 7 22
+ 8 33
+ 9 33
+ 10 33
+ 11 33
+ 12 44
+ 13 44
+ 14 44
+ 15 44

5. Description of Sample Program

5.1 File Configuration

A sample program is provided as a project of HEW (High-performance Embedded Workshop). When h8s.hws is executed, HEW starts up to enable source program referencing and updating. If you do not have HEW, directly reference the following source files with an editor:

No.	File name	Application
1	resetprg.c	Executed starting at reset the vector address zero when the microcomputer is reset
2	intprg.c	Executed when an interrupt source other than a reset is generated. The DTC vector address is also set in this file.
3	dbsct.c	Setting process of the start and end addresses of the section used by the _INITSCT function of resetprg.c in the section initialization table. For details, refer to sections 9 and 10 of the "H8S and H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual."
4	h8s.c	Main routine of this application note
5	com.c	Main common and interrupt processing routine of this application note
6	2215S.H	Structure definition file of the internal registers of the H8S/2215 This file can be obtained from *2. However, modifications are made to the DMAC-related definition. For information on the modifications, refer to the source code.
7	cwtbl.h	Variable and constant definitions are made for this application note.
8	prototypeh	A prototype declaration is made for this application note.
9	stacksct.h	A stack size is defined.

*1 http://www.renesas.com
*2 http://www.renesas.com

5.2 Linkage

The linkage address of each section is indicated below.

In a HEW project file, a section can be referenced or set with Category : section in the Link/Librarq tab of the -Standard Toolchain option.

Section	Start address
CDtc_vect_CMIA0	0x000480
PResetPRG	0x000800
PIntPRG	
P	0x001000
C	
C$DSEC	
C$BSEC	
D	
B	0xFFB000
R	
S	0xFFEDB0

5.3 Subroutine Specifications

With this application note, DTC parameters can be set and started using one subroutine. This function eases use of DTC.

1. Set DTC parameters.

```
Subroutine name: void dtc_set_parm (int ch_no ,  // Setting of channel number
                    unsigned char MRA ,   // Setting of MRA register
                    unsigned long SAR ,   // Setting of SAR (source) address
                    unsigned char MRB ,   // Setting of MRB register
                    unsigned long DAR ,   // Setting of DAR (destination)
                                                address
                    unsigned int  CRA ,   // Setting of CRA (transfer count)
                    unsigned int  CRB )   // Setting of CRB (number of
                                                blocks)
```

Parameter	Setting
ch_no	Specifies a DTC parameter setting area. (0 to 84) RAM addresses which set DTC parameters depends on the channel numbers. RAM address = 0xFFEBC0 + ch_no*12 The lower 2 bytes of the internal RAM address calculated from the above expression must be initially set in the vector address corresponding to the activation source in advance. For details, see (3), "Setting DTC vector addresses," in the Principles of Operation.
MRA	Sets an operating mode. Bits 7 and 6: Source address mode **DTC_ADDR_SAR_FIX** (0x00) SAR is fixed. **DTC_ADDR_SAR_INC** (0x80) SAR is incremented by 1 or 2 according to the setting of bit 0 after data transfer. **DTC_ADDR_SAR_DEC** (0xC0) SAR is decremented by 1 or 2 according to the setting of bit 0 after data transfer. Bits 5 and 4: Destination address mode **DTC_ADDR_DAR_FIX** (0x00) SAR is fixed. **DTC_ADDR_DAR_INC** (0x20) DAR is incremented by 1 or 2 according to the setting of bit 0 after data transfer. **DTC_ADDR_DAR_DEC** (0x30) DAR is decremented by 1 or 2 according to the setting of bit 0 after data transfer. Bits 3 to 1: DTC transfer mode **DTC_NORMAL** (0x00) Normal mode **DTC_REPEAT_DAR** (0x04) The DAR side is in the repeat mode. **DTC_REPEAT_SAR** (0x06) The SAR side is in the repeat mode. **DTC_BLOCK_DAR** (0x08) The DAR side is in the block transfer mode. **DTC_BLOCK_SAR** (0x0A) The SAR side is in the block transfer mode. Bit 0: Data transfer size **DTC_TRANS_1BYTE** (0x00) The data size to be transferred is 1 byte (byte size). **DTC_TRANS_2BYTE** (0x01) The data size to be transferred is 2 bytes (word size). Note: The subroutine specification source specifies DTC parameters in combination like DTC_ADDR_SAR_INC|DTC_ADDR_DAR_INC|DTC_NORMAL|DTC_TRANS_1BYTE.
SAR	Specifies the starting address of the transfer source.
MRB	Specifies a DTC operating mode. Bit 7: Chain transfer enabled DTC_NO_CHAIN (0x00) Chain transfer disabled. DTC_CHAIN (0x80) Chain transfer to the next channel enabled. Bit 6: Interrupt method

Parameter	Setting
	DTC_END_INT_MODE (0x00) An interrupt occurs when all data transfers end. DTC_ALL_INT_MODE (0x40) An interrupt occurs at every data transfer. Bits 5 to 0: Not used Note: The subroutine specification source specifies DTC parameters in combination like DTC_NO_CHAIN|DTC_END_INT_MODE.
DAR	Specifies the starting address of the transfer destination.
CRA	Specifies the number of times the DTC transfers data (data transfer count). In normal mode, you can specify a value from 1 to 65536. In the repeat mode and the block transfer mode, you can specify a value from 1 to 256.
CRB	Specifies the number of times the DTC transfers data in the block transfer mode. You can specify a value from 1 to 65536.

2. Activate the DTC.

```
Subroutine name:  int dtc_start
                  (int action_item ,      // Setting of activation source
                   unsigned char  soft_vect_no)  // The Vector number when the
                                                 activation source is
                                                 software
```

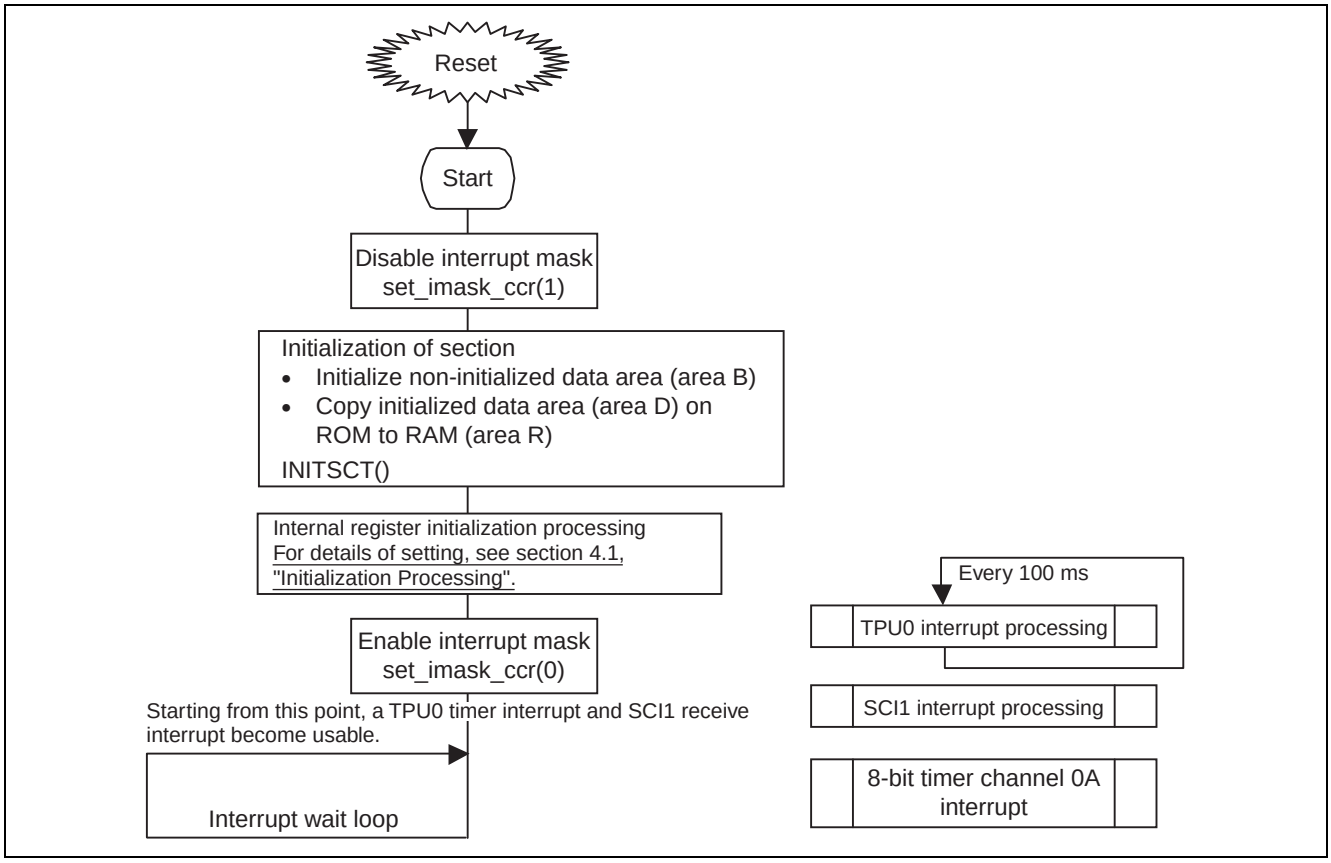
Argument	Setting
action_item	Specifies an activation source. **DTC_ACT_MANUAL** (0x0000): Software activation **DTC_ACT_IRQ0** (0x0A80): External pin interrupt IRQ0 **DTC_ACT_IRQ1** (0x0A40): External pin interrupt IRQ1 **DTC_ACT_IRQ2** (0x0A20): External pin interrupt IRQ2 **DTC_ACT_IRQ3** (0x0A10): External pin interrupt IRQ3 **DTC_ACT_IRQ4** (0x0A08): External pin interrupt IRQ4 **DTC_ACT_IRQ5** (0x0A04): External pin interrupt IRQ5 **DTC_ACT_IRQ7** (0x0A01): External pin interrupt IRQ6 **DTC_ACT_ADC** (0x0B40): AD conversion end interrupt **DTC_ACT_TGI0A** (0x0B20): TPU channel 0A interrupt **DTC_ACT_TGI0B** (0x0B10): TPU channel 0B interrupt **DTC_ACT_TGI0C** (0x0B08): TPU channel 0C interrupt **DTC_ACT_TGI0D** (0x0B04): TPU channel 0D interrupt **DTC_ACT_TGI1A** (0x0B02): TPU channel 1A interrupt **DTC_ACT_TGI1B** (0x0B01): TPU channel 1B interrupt **DTC_ACT_TGI2A** (0x0C80): TPU channel 2A interrupt **DTC_ACT_TGI2B** (0x0C40): TPU channel 2B interrupt **DTC_ACT_CMIA0** (0x0D08): 8-bit timer channel 0A interrupt **DTC_ACT_CMIB0** (0x0D04): 8-bit timer channel 0B interrupt **DTC_ACT_CMIA1** (0x0D02): 8-bit timer channel 1A interrupt **DTC_ACT_CMIB1** (0x0D01): 8-bit timer channel 1B interrupt **DTC_ACT_DEND0A** (0x0E80): DMAC channel 0A end interrupt **DTC_ACT_DEND0B** (0x0E40): DMAC channel 0B end interrupt **DTC_ACT_DEND1A** (0x0E20): DMAC channel 1A end interrupt **DTC_ACT_DEND1B** (0x0E10): DMAC channel 1B end interrupt **DTC_ACT_RXI0** (0x0E08): SCI channel 0 reception interrupt **DTC_ACT_TXI0** (0x0E04): SCI channel 0 transmission interrupt **DTC_ACT_RXI1** (0x0E02): SCI channel 1 reception interrupt **DTC_ACT_TXI1** (0x0E01): SCI channel 1 transmission interrupt **DTC_ACT_RXI2** (0x0F80): SCI channel 2 reception interrupt

Argument	Setting
	DTC_ACT_TXI2 (0x0F40): SCI channel 2 transmission interrupt This application note uses only **DTC_ACT_CMIA0** (8-bit timer channel 0A interrupt) but supports all the functions as the subroutine function.
soft_vect_no	Specifies a vector number when the activation source is "software activation." You can specify 0x00 to 0x7F. The vector address used in software activation depends on the vector number to be specified. Vector address = 0x400 + soft_vect_no *2 The lower 2 bytes of the RAM address where DTC parameters are to be set must be initially set in the vector address calculated from the above expression in advance. For details, see (3), "Setting DTC vector addresses," in the Principles of Operation.

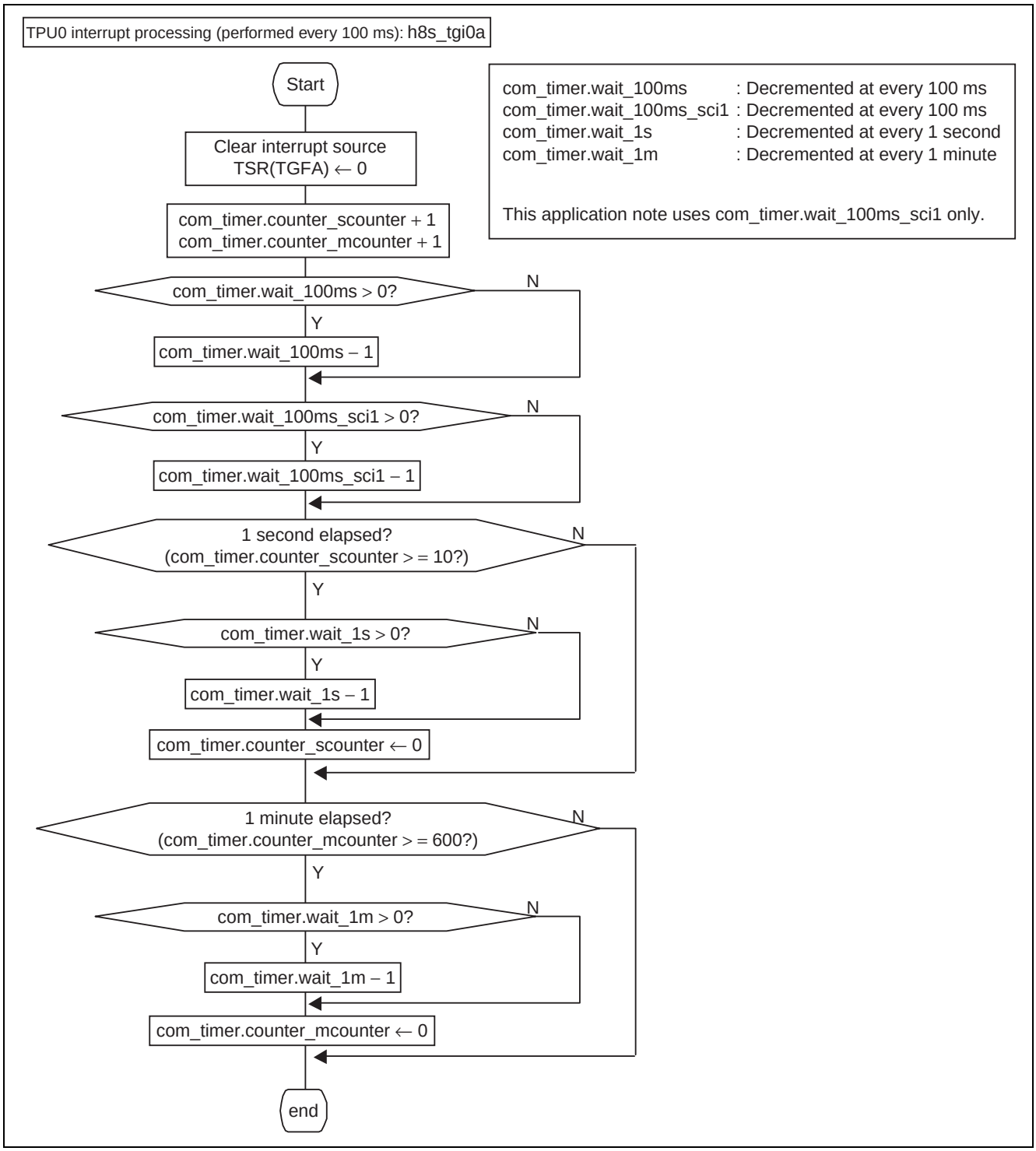
Return value	Description
0	Normal end of DTC activation
–1	The DTC cannot be activated because the DTC software is being activated.

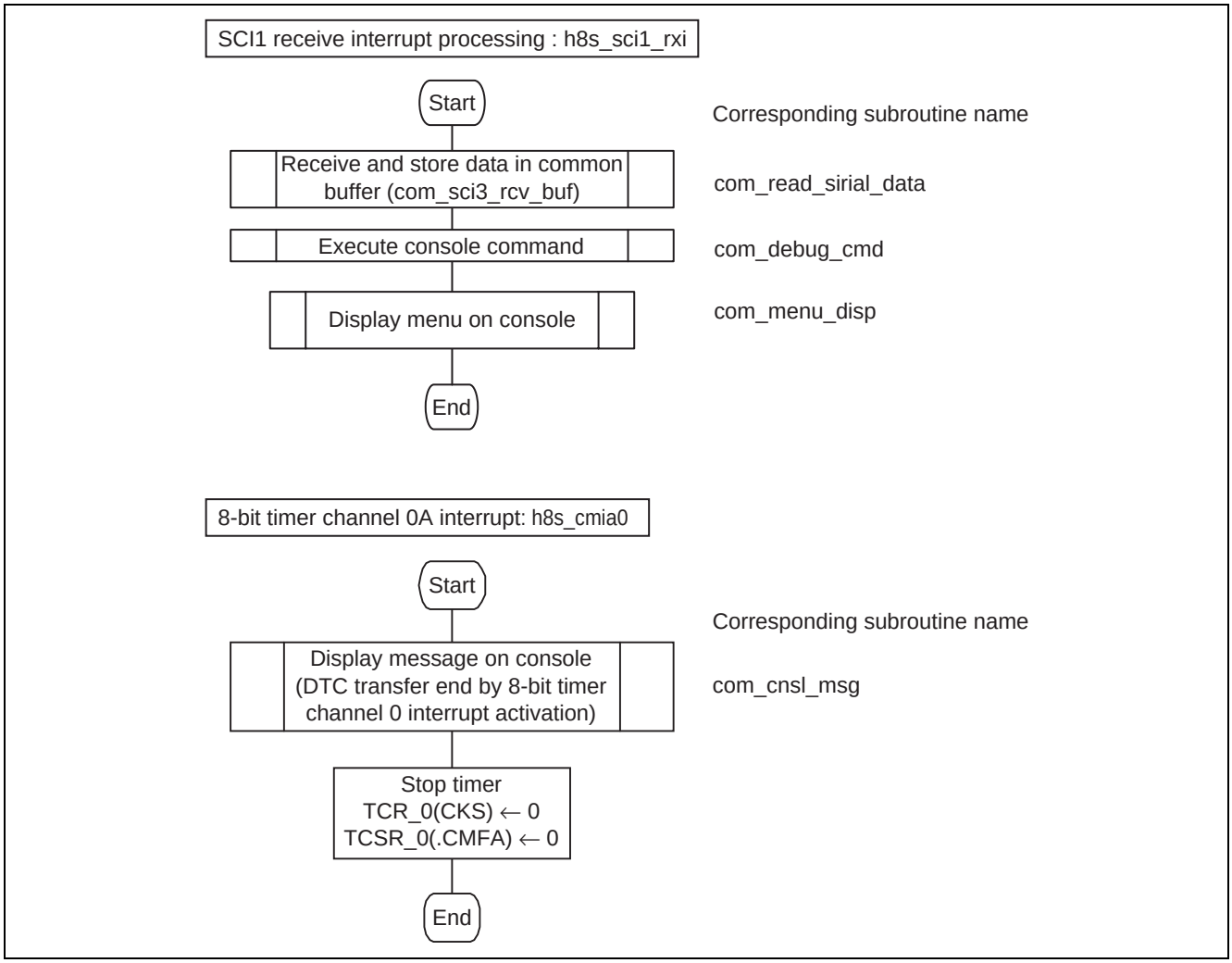
6. Flowchart

6.1 Overall Flow



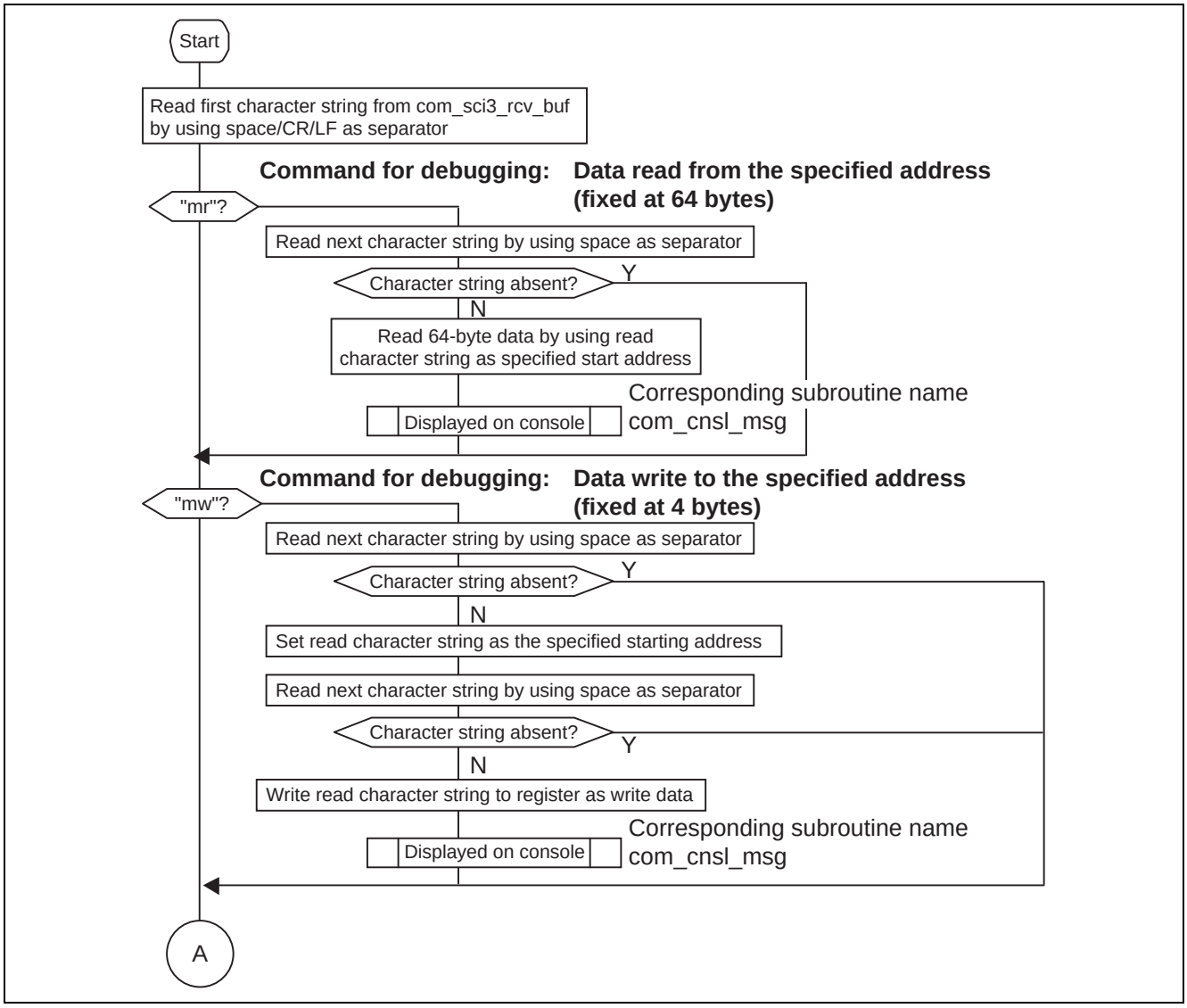
6.2 Interrupt Processing

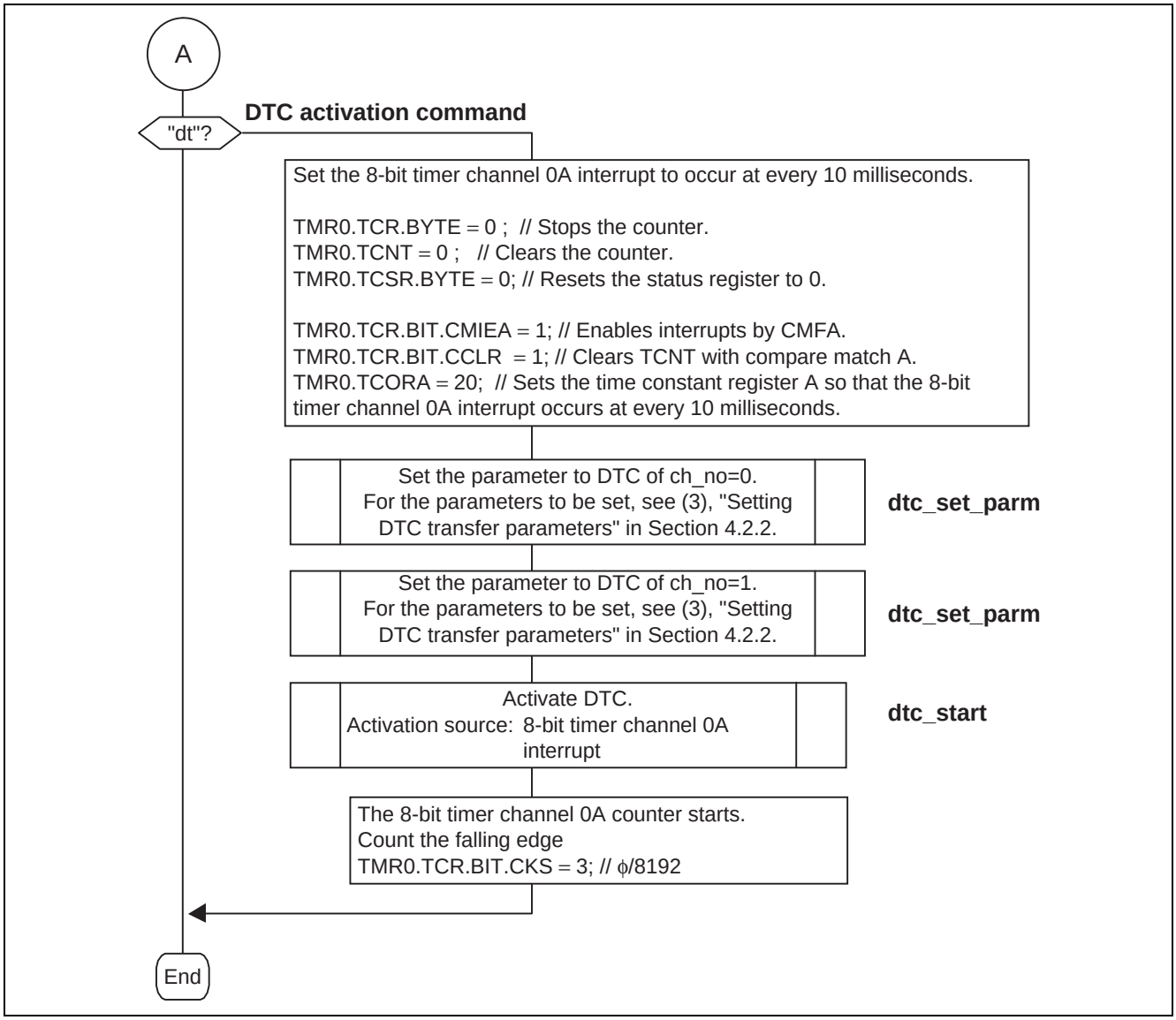



SCI1 receive interrupt processing : h8s_sci1_rxi
Start
Receive and store data in common buffer (com_sci3_rcv_buf)
Execute console command
Display menu on console
End
Corresponding subroutine name
com_read_sirial_data
com_debug_cmd
com_menu_disp
8-bit timer channel 0A interrupt: h8s_cmia0
Start
Display message on console (DTC transfer end by 8-bit timer channel 0 interrupt activation)
Stop timer
TCR_0(CKS) ← 0
TCSR_0(.CMFA) ← 0
End
Corresponding subroutine name
com_cnsl_msg

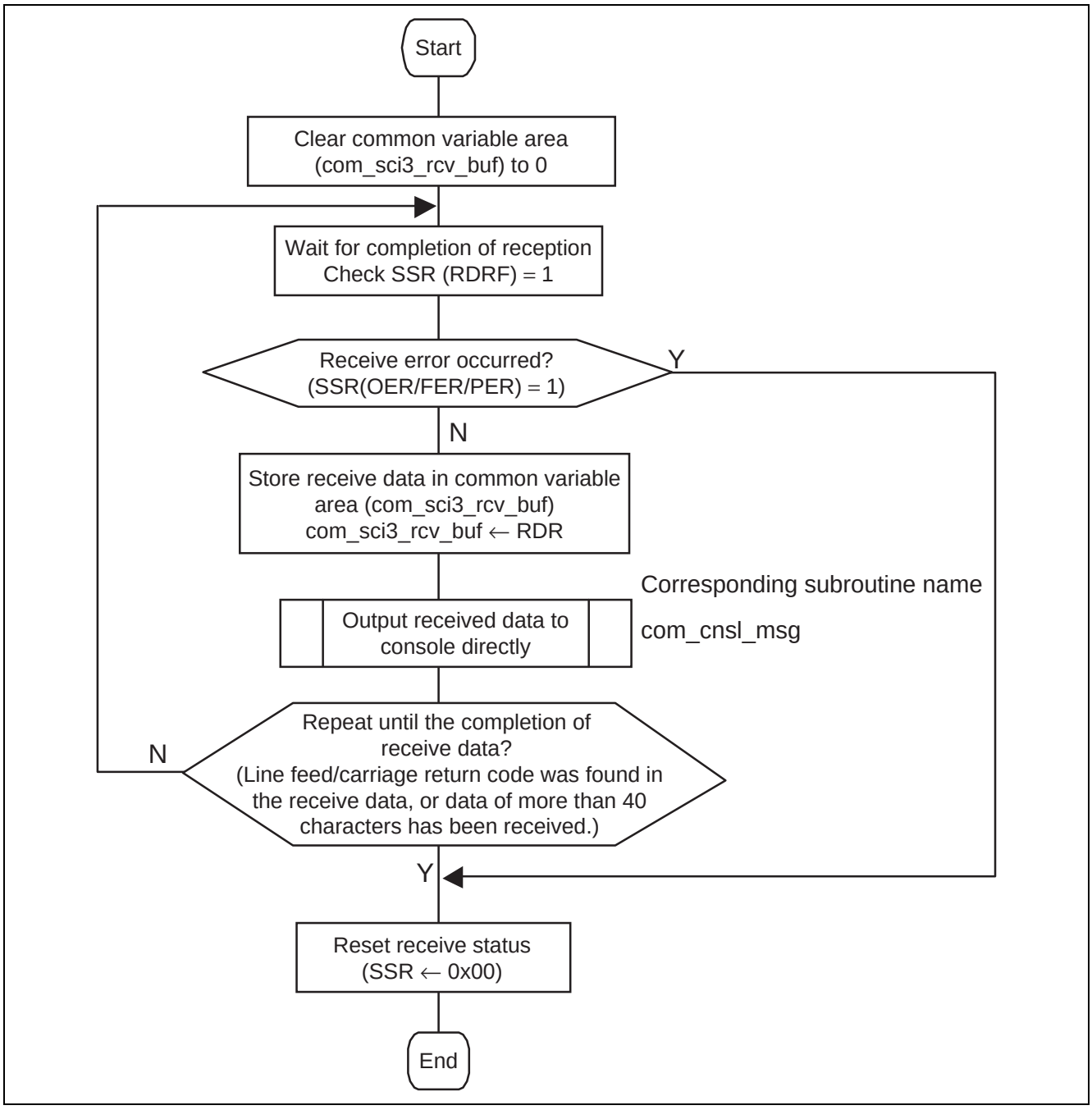
6.3 Detailed Processing

```
com_debug_cmd
: Console command analysis and execution
```

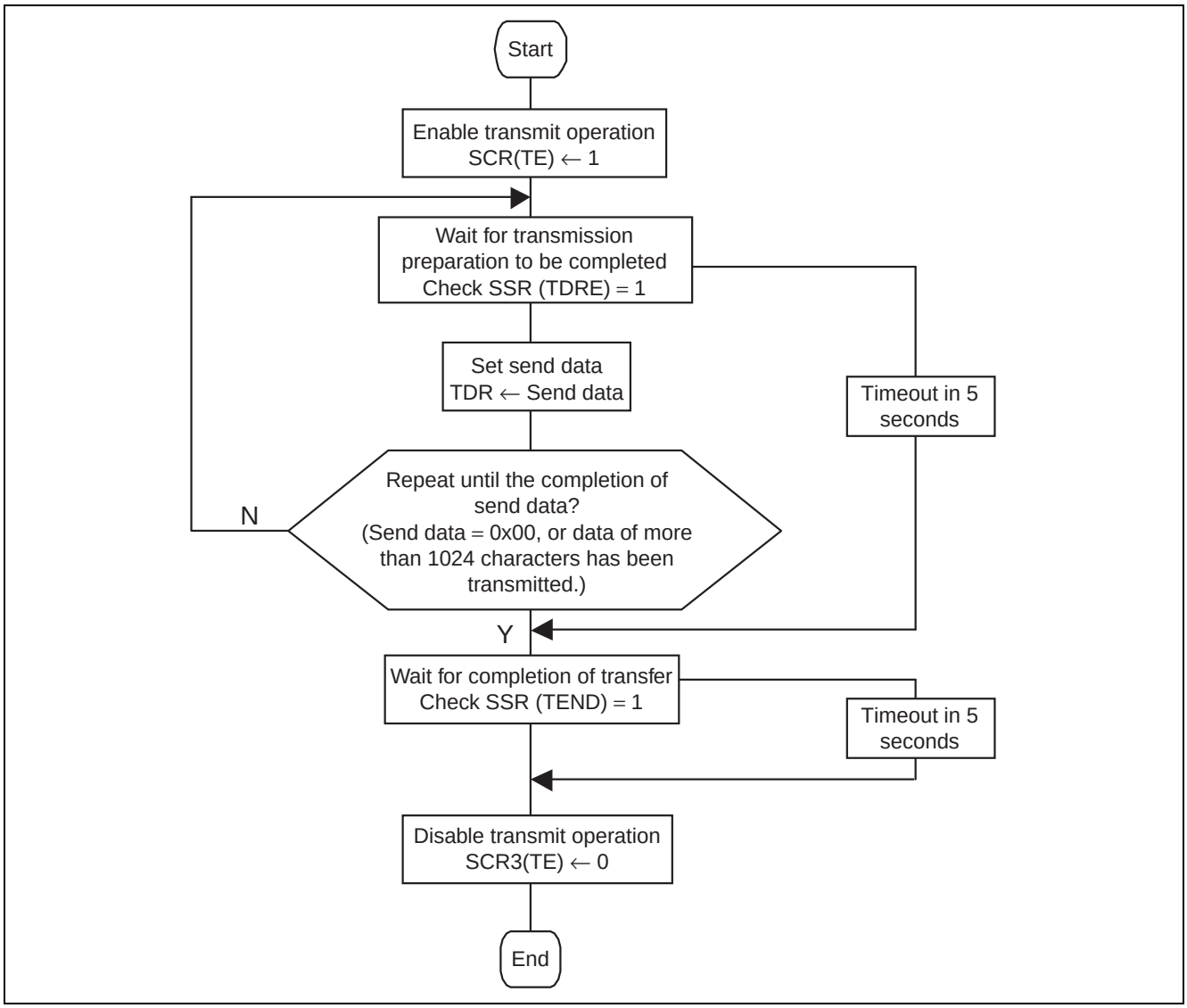



A
"dt"?
DTC activation command
Set the 8-bit timer channel 0A interrupt to occur at every 10 milliseconds.
TMR0.TCR.BYTE = 0 ; // Stops the counter.
TMR0.TCNT = 0 ; // Clears the counter.
TMR0.TCSR.BYTE = 0; // Resets the status register to 0.
TMR0.TCR.BIT.CMIEA = 1; // Enables interrupts by CMFA.
TMR0.TCR.BIT.CCLR = 1; // Clears TCNT with compare match A.
TMR0.TCORA = 20; // Sets the time constant register A so that the 8-bit timer channel 0A interrupt occurs at every 10 milliseconds.
Set the parameter to DTC of ch_no=0. For the parameters to be set, see (3), "Setting DTC transfer parameters" in Section 4.2.2.
dtc_set_parm
Set the parameter to DTC of ch_no=1. For the parameters to be set, see (3), "Setting DTC transfer parameters" in Section 4.2.2.
dtc_set_parm
Activate DTC. Activation source: 8-bit timer channel 0A interrupt
dtc_start
The 8-bit timer channel 0A counter starts. Count the falling edge
TMR0.TCR.BIT.CKS = 3; // ϕ/8192
End

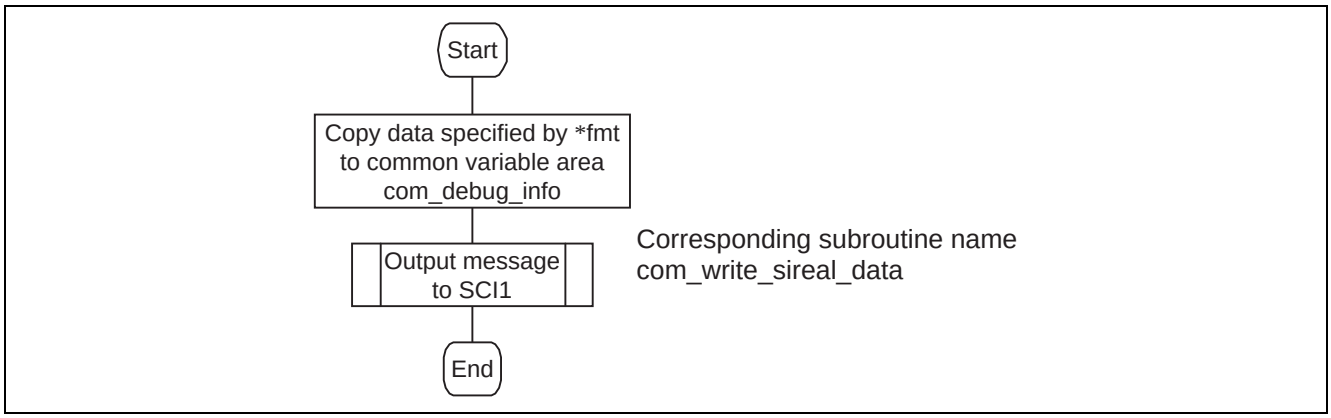
```
com_read_sirial_data
: Reception of a message from the SCI1 interface
   Receive data not longer than 40 characters is received
   in the common variable area (com_sci3_rcv_buf).
```



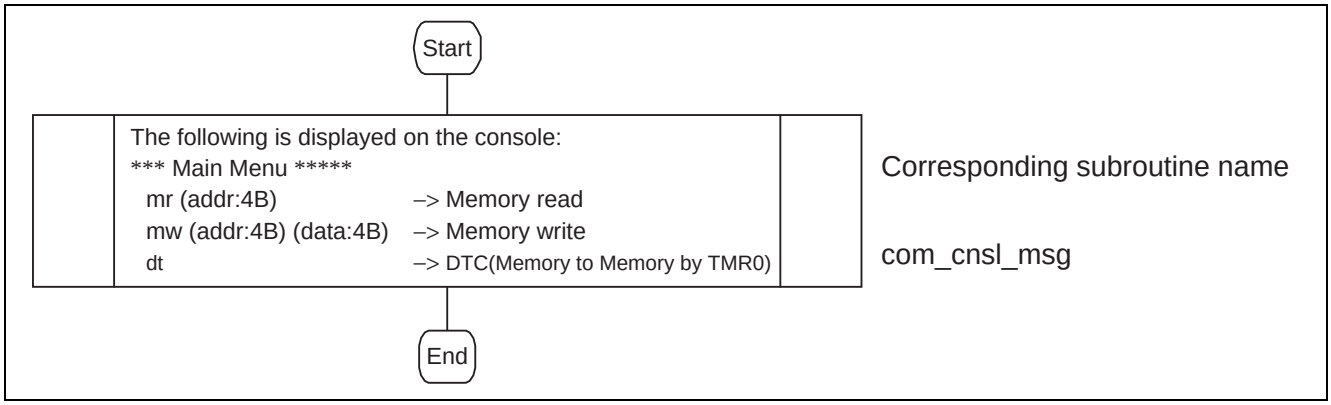
```
com_write_sireal_data ( char *p )
: Transmission of a message to the SCI3 interface
  *p : Address where message data is stored
```



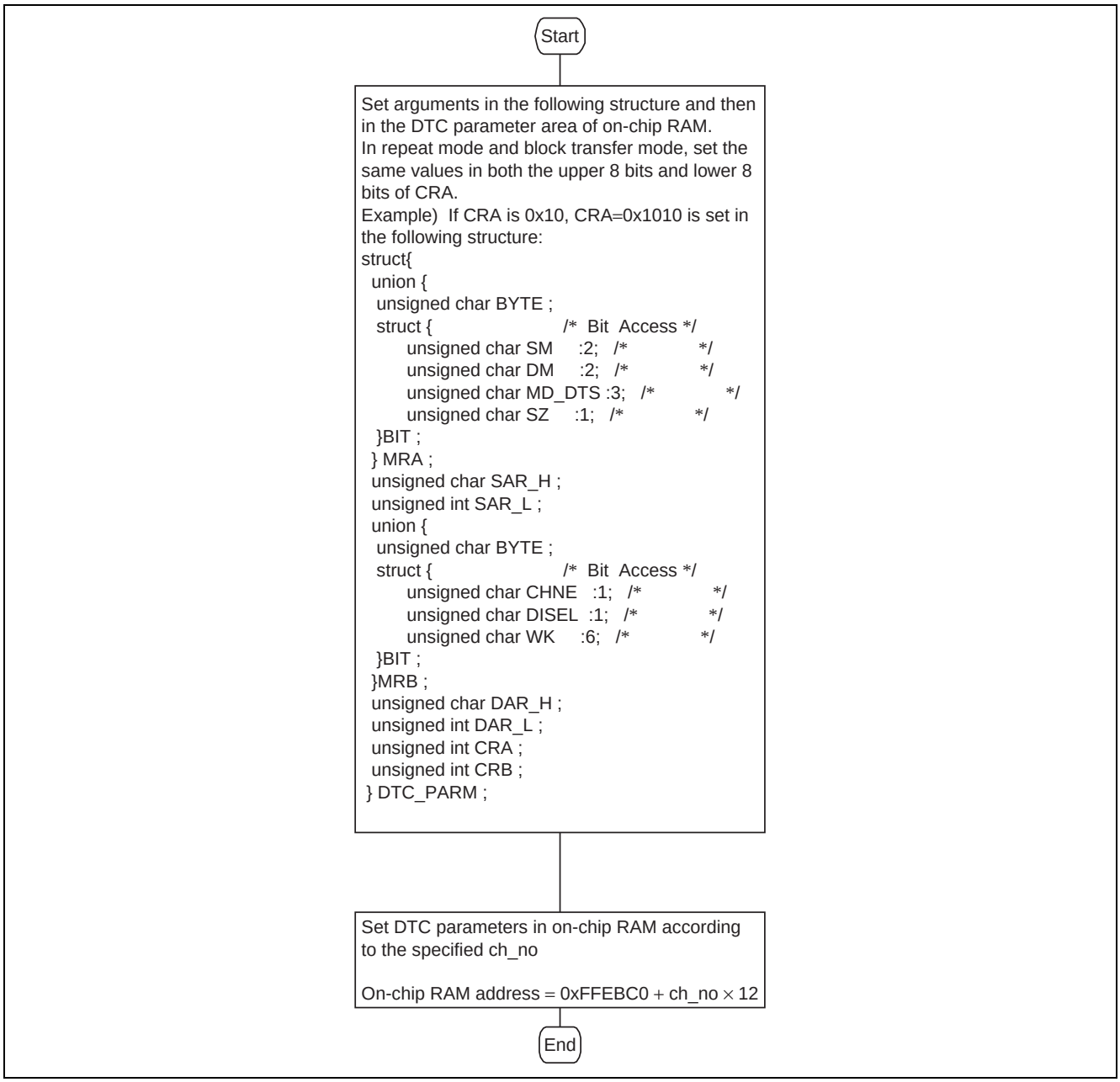
```
com_cnsl_msg(char *fmt, ...)
: Transmission of a message to the console
  *fmt : Address where variable-length message data is stored
```



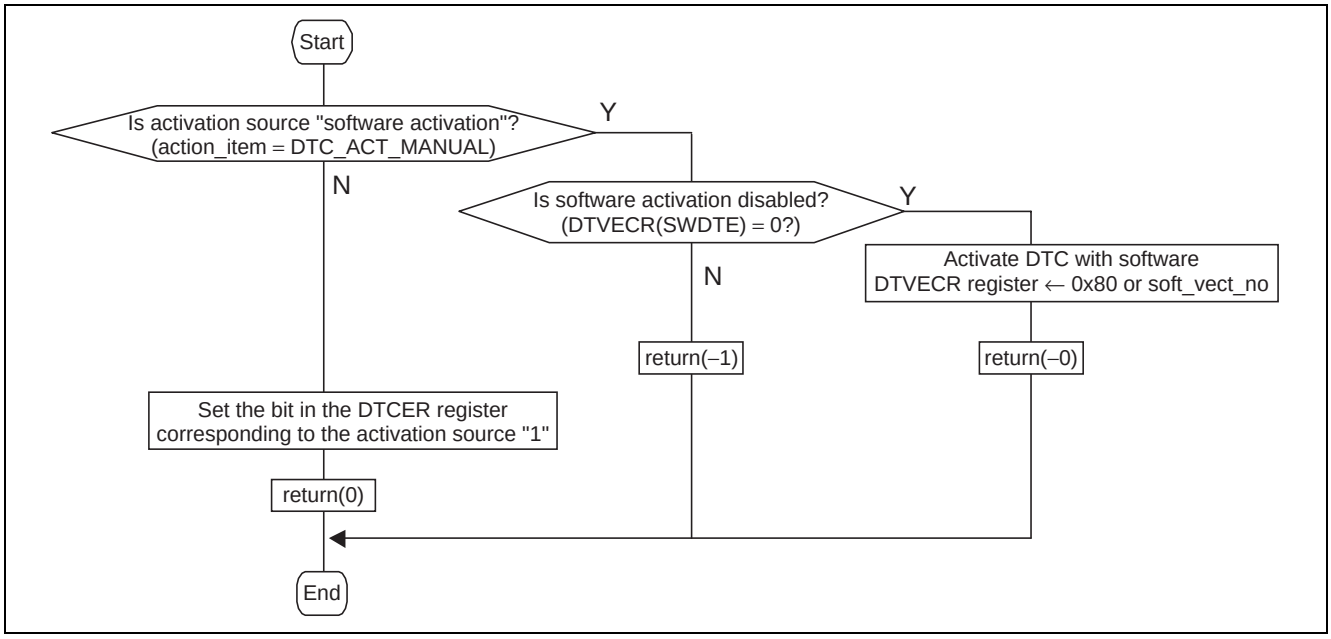

```
com_menu_disp
: Display of the operation menu on the console
```



```
Subroutine name : void dtc_set_parm
                 (int ch_no , unsigned char MRA ,  unsigned long SAR ,
                  unsigned char MRB , unsigned long DAR ,
                  unsigned int  CRA ,   unsigned int  CRB )
: Set the following DTC parameters.
  ch_no:  Sets a channel number.
  MRA   :  Sets the MRA register.
  SAR   :  Sets a SAR (source) address.
  MRB   :  Sets the MRB register.
  DAR   :  Sets a DAR (destination) address.
  CRA   :  Sets CRA (transfer count).
  CRB   :  Sets CRB (number of blocks).
```



```
Subroutine name : int dtc_start
                  (int action_item ,  unsigned long soft_vect_no)
: Activate the DTC.
  action_item:  Activation source
  soft_vect_no:  Vector number when the activation source is "software
         activation"
```



Website and Support

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

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

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Mar.16.04	—	First edition issued
1.01	Mar.06.09	—	Source file is updated

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.