

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 (Software Activation)

Introduction

Transfers data on an SRAM chip to other addresses on the chip by DTC that is activated by software.

Target Device

H8S/2215

Contents

1. Overview	2
2. Configuration	2
3. Description of Functions.....	3
4. Principles of Operation	4
5. Description of Sample Program	15
6. Flowchart.....	19

1. Overview

The H8S/2215 transfers data on an SRAM chip to other addresses on the chip using DTC that is activated by software.

2. Configuration

The configuration of the confirmed operation of this application note is shown in figure 1.

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)

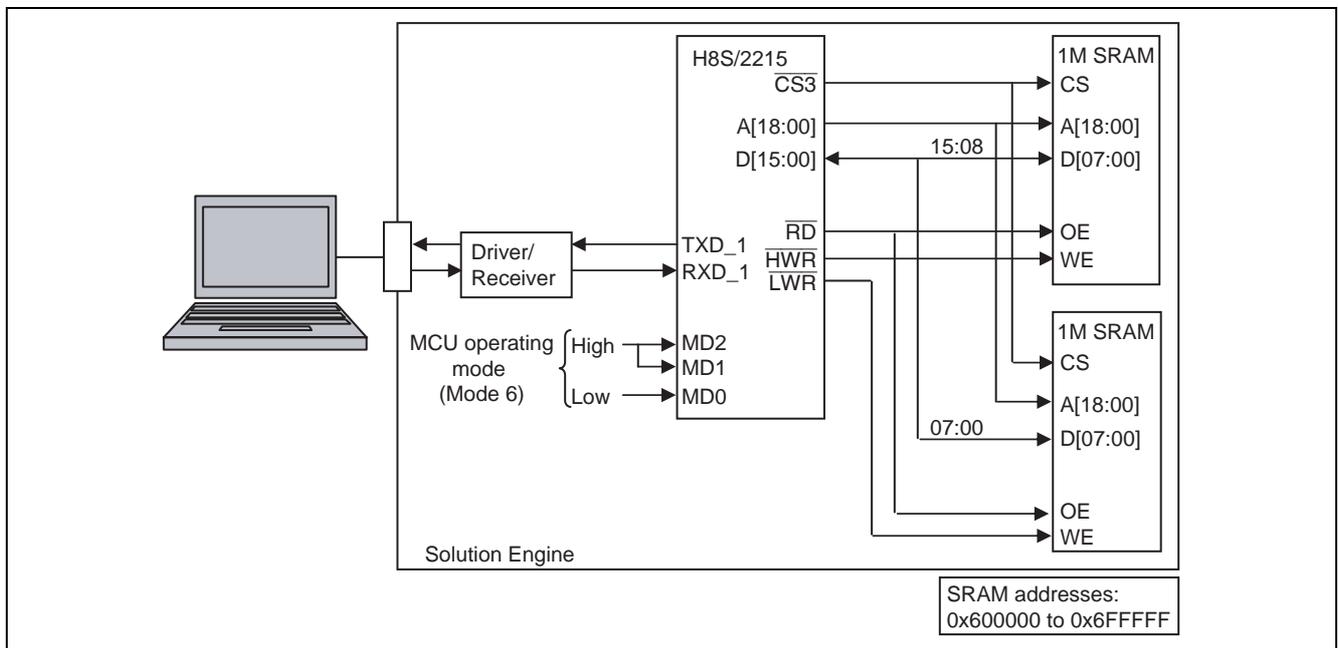


Figure 1 Confirmed Configuration

3. Description of Functions

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 DTC activated by software (Normal mode) Transfers data at 0x600000 to 0x60000F to 0x610000 to 0x61000F on an SRAM chip.	Format: dtΔ0
2	Data transfer with DTC activated by software (Repeat mode) Transfers data at 0x600000 (1 byte) repeatedly to 0x610000 to 0x61000F on an SRAM chip.	Format: dtΔ1
3	Data transfer with DTC activated by software (Block transfer mode) Transfers one block of data at 0x600000 to 0x60000F 4 times to 0x610000 and after on an SRAM chip.	Format: dtΔ2
4	Data of 64 bytes is read from a specified memory address for display.	Format: mrΔaddress (Specify a 4-byte value in address.)
5	Data of 4 bytes is written starting at a specified memory address.	Format: mwΔaddressΔdata (Specify a 4-byte value in address and data.)

4. Principles of Operation

4.1 Initialization Processing

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

1. Control for low power consumption, clock oscillator initialization

Register name	Bit	Name	Value	Contents
←Set value				
LPWCR	7-4	----	0000	
←0x03	3	RFCUT	0	Uses 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 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 A/D module.
	0	MSTPA0	1	(Reserved)
MSTPCRB	7	MSTPB7	0	Operates the SCI0 module.
←0x1F	6	MSTPB6	0	Operates the SCI1 module.
	5	MSTPB5	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 D/A module.
	4:0	MSTPC[4:0]	11111	(Reserved)

2. I/O 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 ←0x77	7	ABW7	0	Area 7 Bus width 16 bits (Not used)
	6:4	ABW[6:4]	111	Areas 6 to 4 Bus width 8 bits (Not used)
	3	ABW3	0	Area 3 Bus width 16 bits (SRAM)
	2:0	ABW[2:0]	111	Areas 2 to 0 Bus width 8 bits (Not used)
PFCR ←0x0F	7:4	----	0	
	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	Bit	Name	Value	Contents
←Set value				
TCR_0 ←0x23	7:5	CCLR[2:0]	001	Counter clear by TGRA compare match
	4:3	CKEG[1:0]	00	Count on rising edge
	2:0	TPSC[2:0]	011	Count with $\phi/64$
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 A/D 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 TGRD interrupt. (Not used)
	2	TGIEC	0	Disables TGRC interrupt. (Not used)
	1	TGIEB	0	Disables TGRB interrupt. (Not used)
	0	TGIEA	0	Enables TGRA interrupt.
TGRA_0 ←25000	15:0	TGRA_0	25000	TBRA output compare value (To be set to generate an interrupt at intervals of 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 DTC.

Register name	Bit	Name	Value	Contents
←Set value				
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 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 time for at least one stop bit is awaited. (38400 bps: About 30 μ s)

— Receive processing is enabled.

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 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

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

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

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

4.2.2 Data Transfer by DTC Activated by Software

1. Overview

Activate DTC by entering the following command from the terminal software connected to SCI_1:

Format: dtΔ(mode)

When mode is 0, DTC is activated in the normal mode.

When mode is 1, DTC is activated in the repeat mode.

When mode is 2, DTC is activated in the block transfer mode.

In the normal mode, the H8S/2215 executes the specified DTC transfer only once. DTC transfers data at 0x600000 to 0x60000F to 0x610000 to 0x61000F on an SRAM chip.

In the repeat mode, the H8S/2215 executes DTC transfer repeatedly. DTC transfers data at 0x600000 (1 byte) repeatedly to 0x610000 to 0x61000F on an SRAM chip.

In the block transfer mode, the H8S/2215 executes DTC transfer the specified number of times. DTC transfers one block of data at 0x600000 to 0x60000F 4 times to 0x610000 and after on an SRAM chip.

Use software to activate DTC. In the normal mode and block transfer mode, when data transfer ends, a software activation data transfer end interrupt (SWDTEND) occurs.

In the repeat mode, the SWDTEND does not occur because data is transferred repeatedly.

The following explains how to activate DTC.

2. Setting DTC transfer parameters

First, set parameters to specify how DTC transfer is carried out.

DTC transfer parameters must be set in 0xFFEBC0 to 0xFFEFBF of the internal RAM in the following format: ch_no specifies the locations of DTC transfer parameters to be set, so users do not need to be directly concerned with 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 in ch_no = 0 (internal RAM 0xFFEBC0 and after).

You can use one subroutine (dct_set_parm) to set DTC transfer parameters collectively.

For more information about subroutines, see section 5, "Description of Sample Program."

A. Normal mode: DTC transfers data at 0x600000 to 0x60000F to 0x610000 to 0x61000F on an SRAM chip.

Register Name	Settings	Description
MRA	0xA0	<ul style="list-style-type: none"> • Increments SAR by 1 after data transfer. • Increments DAR by 1 after data transfer. • DTC mode = normal mode • Size of the data to be transferred once for each (1) byte
MRB	0x00	<ul style="list-style-type: none"> • DTC chain unavailable ^{*1} • When all the specified data transfers have ended, generates the software activation data transfer end interrupt (SWDTEND).
SAR	0x600000	Specifies a source address.
DAR	0x610000	Specifies a destination address.
CRA	0x0010	Specifies the number of times DTC transfers data (1 to 65536).
CRB	0x0000	(Not used)

^{*1} This application note does not describe DTC chain function. See the related hardware manuals.

B. Repeat mode: DTC transfers data at 0x600000 (1 byte) repeatedly to 0x610000 to 0x61000F on an SRAM chip.

Register Name	Settings	Description
MRA	0x24	<ul style="list-style-type: none"> SAR (source address) is fixed. Increments DAR (destination address) by 1 after data transfer. DTC mode = repeat mode (The DAR side is the repeat area.) Size of the data to be transferred for each (1) byte
MRB	0x00	<ul style="list-style-type: none"> DTC chain unavailable ^{*1} When all the data transfers have ended, generates the interrupt for the software activation data transfer end (SWDTEND).
SAR	0x600000	Specifies a source address.
DAR	0x610000	Specifies a destination address.
CRA	0x1010	<p>Specifies the number of times DTC transfers data (1 to 256). In the repeat mode, the same values are set in both the upper 8 bits and lower 8 bits.</p> <p>The lower 8 bits are decremented by 1 each time data is transferred. When the count reaches 0x00, the contents of the upper 8 bits are transferred. This data transfer is repeated.</p>
CRB	0x0000	(Not used)

^{*1} This application note does not describe DTC chain function. See the related hardware manuals.

C. Block transfer mode: Transfers one block of data at 0x600000 to 0x60000F 4 times to 0x610000 and after on an SRAM chip.

Register Name	Settings	Description
MRA	0xAA	<ul style="list-style-type: none"> Increments SAR (source address) by 1 after data transfer. Increments DAR (destination address) by 1 after data transfer. DTC mode = block transfer mode (The SAR side is the block area.) Size of the data to be transferred for each (1) byte
MRB	0x00	<ul style="list-style-type: none"> DTC chain unavailable ^{*1} When all the specified data transfers have ended, generates the software activation data transfer end interrupt (SWDTEND).
SAR	0x600000	Specifies a source address.
DAR	0x610000	Specifies a destination address.
CRA	0x1010	<p>Specifies the number of times DTC transfers data (1 to 256). In the block transfer mode, the same values are set in both the upper 8 bits and lower 8 bits.</p> <p>The lower 8 bits are decremented by 1 each time data is transferred. When the count reaches 0x00, the contents of the upper 8 bits are transferred. This data transfer is repeated.</p>
CRB	0x0004	Number of times DTC transfers data in the block transfer mode

^{*1} This application note does not describe DTC chain function. See the related hardware manuals.

3. Setting DTC vector addresses

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 DTC transfer parameters set in item 2 are set. When DTC is activated, DTC transfer parameters are read from the corresponding vector address and 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 DTC transfer parameters must be initially set in the vector address area for compilation.

128 vector addresses can be specified for DTC activated by software using the vector number (bit 6 (DTVEC6) to bit 0 (DTVEC0) of the DTVECR register). The vector address is expressed as $0x400 + \text{vector number} \times 2$.

The vector address is 0x420 because 0x10 is used as the vector number.

The lower 2 bytes (0xEBC0) of the starting address of DTC transfer parameters must be initially set in advance at this vector address for compilation.

4. DTC activation

When activating DTC by software, specify a vector number (DTVEC6 to DTVEC0) in the DTVECR register and set the SWDTE bit for software activation to "1." 0x90 is set in the DTVECR register because 0x10 is used as the vector address.

Specify an activation source in the `dtc_start` subroutine. For more information about this subroutine, see section 5, Description of Sample Program.

5. DTC operation

DTC transfer is performed by 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)

Each time data is transferred, the SWDTE bit (software activation) in the DTVECR register is reset to "0" and DTC transfer is halted. To continue data transfer, the SWDTE bit must be set to "1" again.

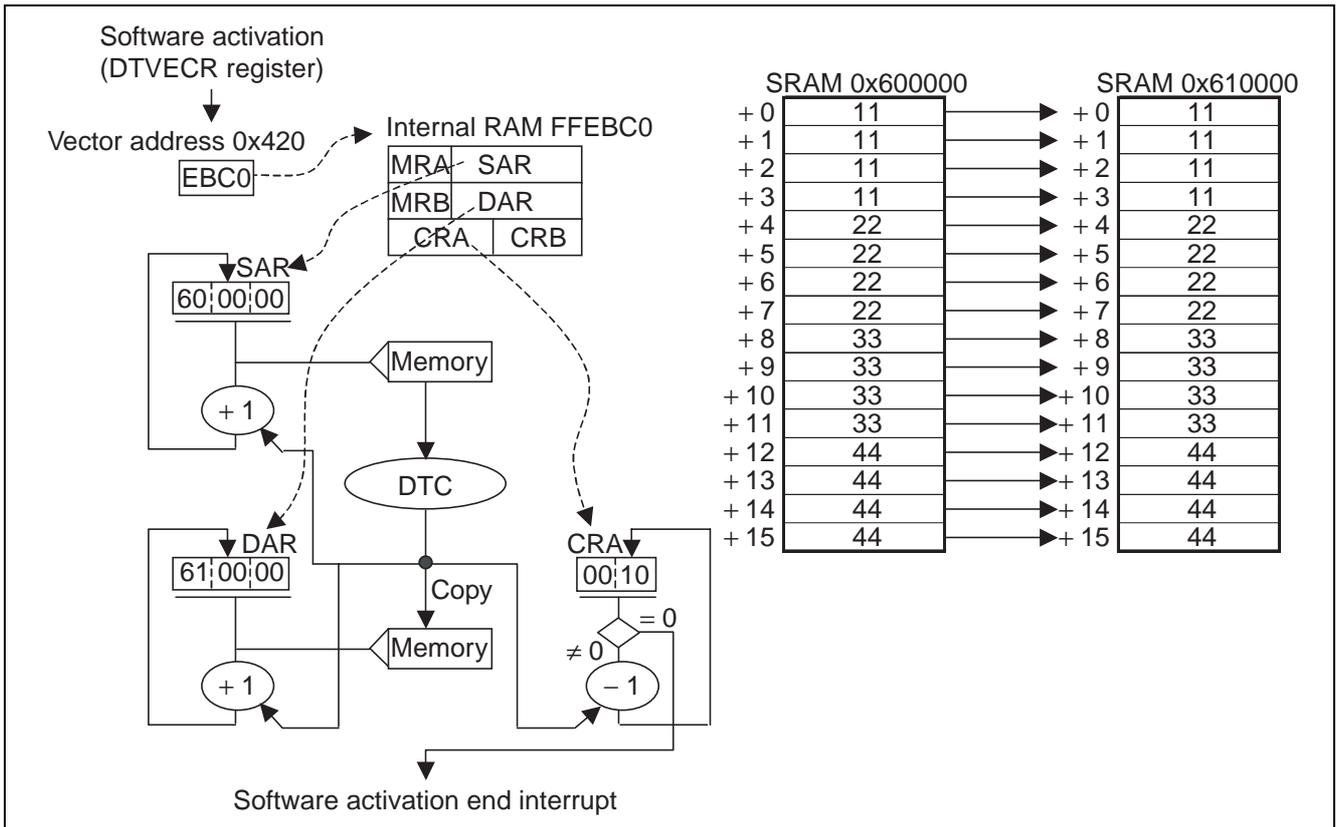
In the normal mode and the block transfer mode, when all data transfers have ended, the software activation data transfer end interrupt (SWDTEND) occurs. At this time, the SWDTE bit is still held at "1", so clear it to "0" to end data transfer.

When the DISEL bit is 1:

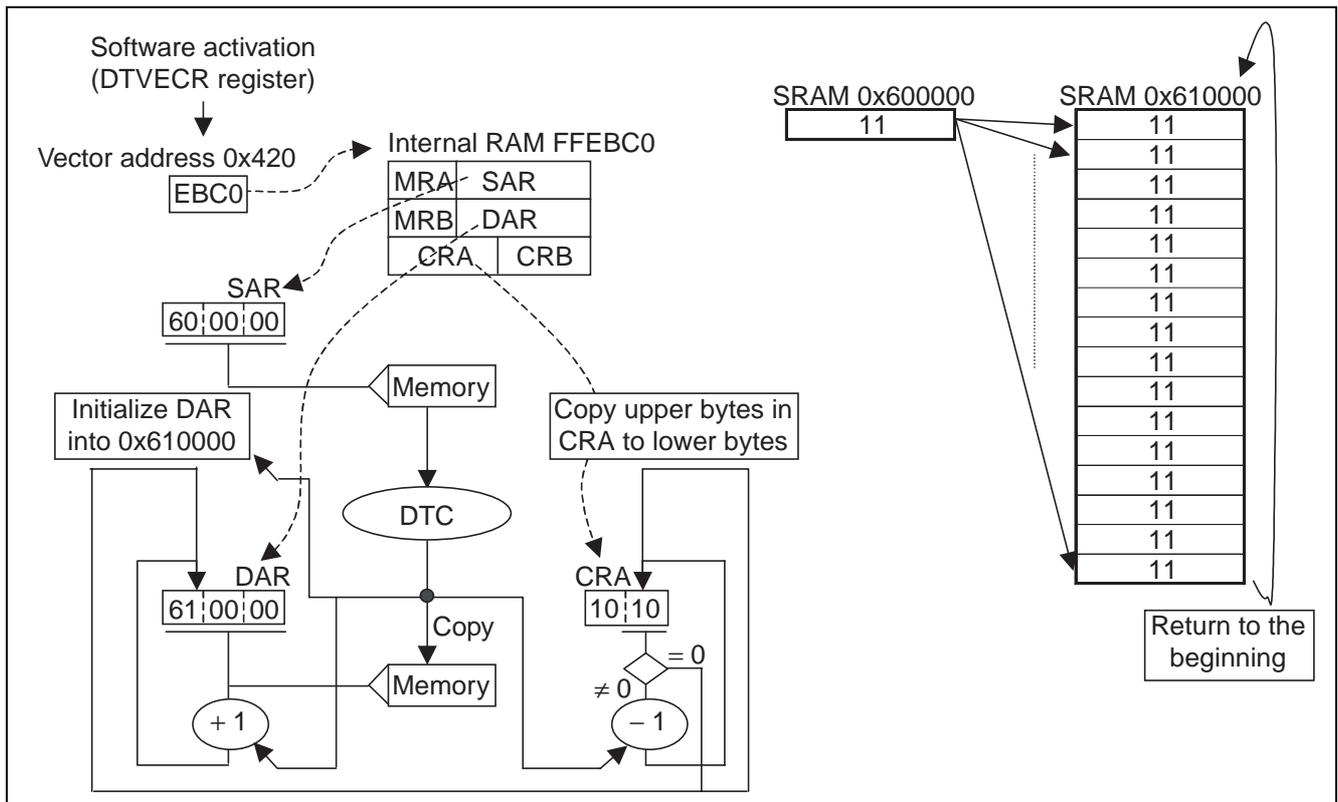
Each time data is transferred, the software activation data transfer end interrupt (SWDTEND) occurs. At this time, the SWDTE bit (software activation) in the DTVECR register is still held at "1." Also when all data transfers end, the SWDTEND interrupt occurs. Since there is only one interrupt type, you cannot determine whether all data transfers have ended even if the SWDTEND interrupt occurred. For this reason, when using this mode, you must prepare a software counter, perform addition/subtraction during interrupt processing, and monitor how far data transfer has advanced.

The following shows DTC transfer flows in the normal mode, repeat mode, and block transfer mode in this application note (DISEL = 0).

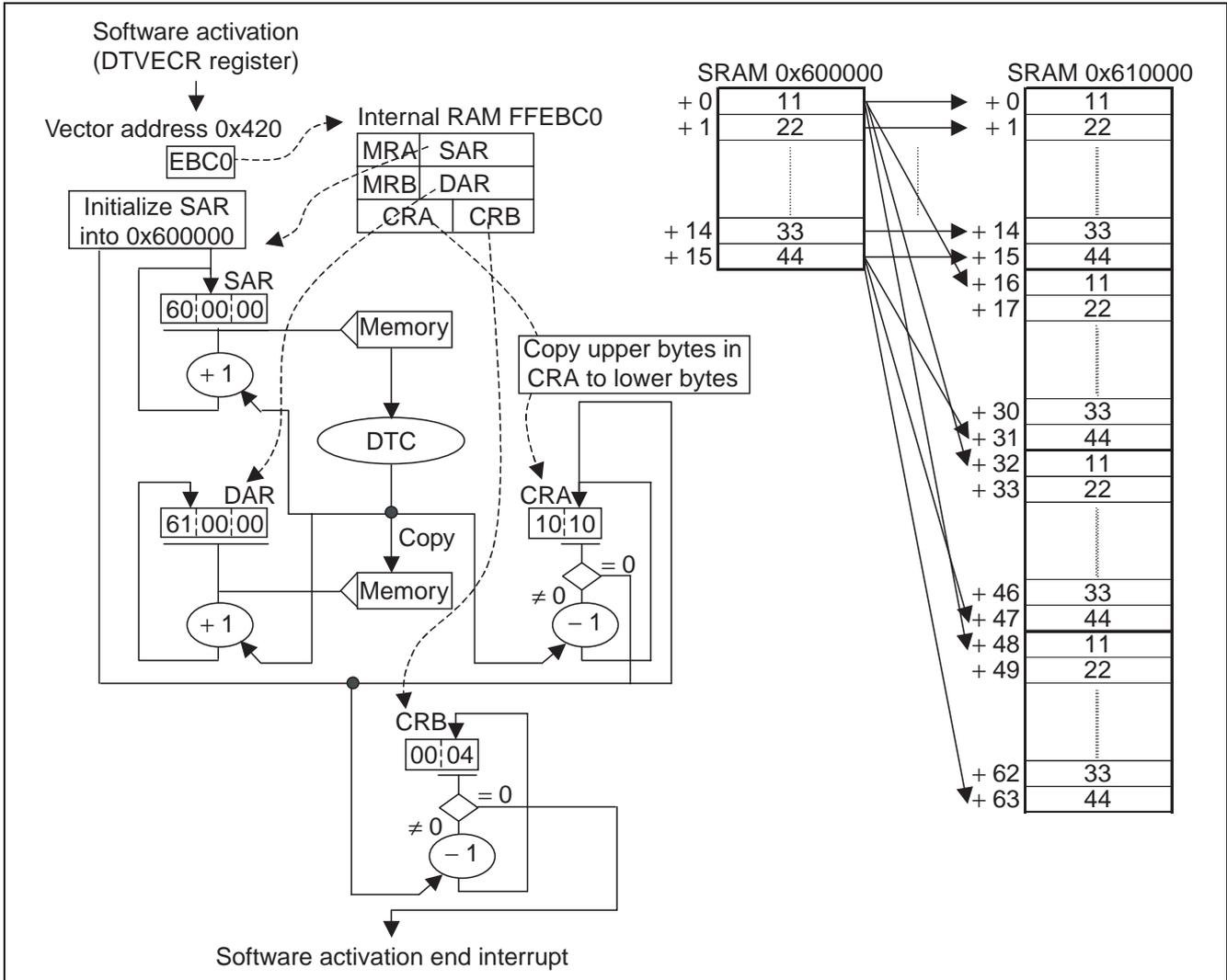
6. DTC transfer flow in the normal mode
 - A. DTC reads DTC transfer parameters (12 bytes from 0xFFEBC0) from vector address 0x420.
 - B. DTC copies the contents of the source address indicated by SAR to the destination address indicated by DAR.
 - C. DTC increments the contents of SAR and DAR by 1 according to the transfer parameters. Then DTC decrements the transfer count indicated by CRA by 1, and clears the SWDTE bit in the DTVECR register to "0." When it detects that the SWDTE bit is "0", software sets this bit to "1" and resumes data transfer.
 - D. B. and C. are repeated until CRA becomes 0.
 - E. When CRA becomes 0, the software activation data transfer end interrupt (SWDTEND) occurs.
 - F. Software clears the SWDTE bit (software activation) in the DTVECR register to "0" in the SWDTEND interrupt and ends DTC transfer.



7. DTC transfer flow in the repeat mode
 - A. DTC reads DTC transfer parameters (12 bytes from 0xFFEBC0) from vector address 0x420.
 - B. DTC copies the contents of the source address indicated by SAR to the destination address indicated by DAR.
 - C. DTC holds the contents of SAR as they are and increments the contents of DAR by 1 according to the transfer parameters. Then DTC decrements the transfer count indicated by CRA by 1, and clears the SWDTE bit in the DTVECR register to "0." When software detects that the SWDTE bit is "0", it sets this bit to "1" and resumes data transfer.
 - D. B. and C. are repeated until CRA becomes 0.
 - E. When CRA becomes 0, DTC initializes the contents of DAR to 0x610000, copies the contents of the upper bytes in CRA to the lower bytes, and initializes the contents of CRA to 0x1010.
 - F. DTC returns to B. In the repeat mode, data transfer is repeated and the software activation data transfer end interrupt never occurs.



8. DTC transfer flow in the block transfer mode
 - A. DTC reads DTC transfer parameters (12 bytes from 0xFFEBC0) from vector address 0x420.
 - B. DTC copies the contents of the source address indicated by SAR to the destination address indicated by DAR.
 - C. DTC increments the contents of SAR and DAR by 1 according to the transfer parameters. Then DTC decrements the transfer count indicated by CRA by 1, and clears the SWDTE bit in the DTVECR register to "0." When software detects that the SWDTE bit is "0", it sets this bit to "1" and resumes data transfer.
 - D. B. and C. are repeated until CRA becomes 0.
 - E. When CRA becomes 0, DTC initializes the contents of SAR to 0x600000. Then DTC copies the contents of the upper bytes in CRA to the lower bytes, and initializes the contents of CRA to 0x1010. DTC also decrements the block transfer count indicated by CRB by 1.
 - F. B. to E. are repeated till CRB becomes 0.
 - E. When CRB becomes 0, the software activation data transfer end interrupt (SWDTEND) occurs.
 - F. Software clears the SWDTE bit (software activation) in the DTVECR register to "0" in the SWDTEND interrupt and ends DTC transfer.



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 vector address 0 when a reset is input to the microcomputer
2	intprg.c	Executed when an interrupt source other than a reset is generated. DTC vector address is also set in this file.
3	dbstc.c	Processing for setting the start and end addresses of a section used by the _INITSCT function of resetprg.c in the section initialization table. For details of the processing, 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 the modifications, check the source code.
7	cwtbl.h	Variable and constant definitions are defined for this application note.
8	prototype.h	A prototype declaration is made for this application note.
9	stackst.h	A stack size definition is made.

^{*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_SOFT	0x000420
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) Where in the RAM address DTC parameters are set 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 section 4, 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) DAR 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 size of the data to be transferred is 1 byte (byte size). DTC_TRANS_2BYTE (0x01) The size of the data 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 enable DTC_NO_CHAIN (0x00) Chain transfer is disabled. DTC_CHAIN (0x80) Chain to the next channel is enabled.

Parameter	Setting
	Bit 6: Interrupt method DTC_END_INT_MODE (0x00) An interrupt occurs when all data transfers end. DTC_ALL_INT_MODE (0x40) An interrupt occurs per data transfer. Bits 5 to 0: Not used Note: The subroutine specification source specifies DTC parameters in combination such as DTC_NO_CHAIN DTC_END_INT_MODE.
DAR	Specifies the starting address of the transfer destination.
CRA	Specifies the number of times DTC transfers data (data transfer count). In the 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 DTC transfers data in the block transfer mode. You can specify a value from 1 to 65536.

2. Activate 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
    
```

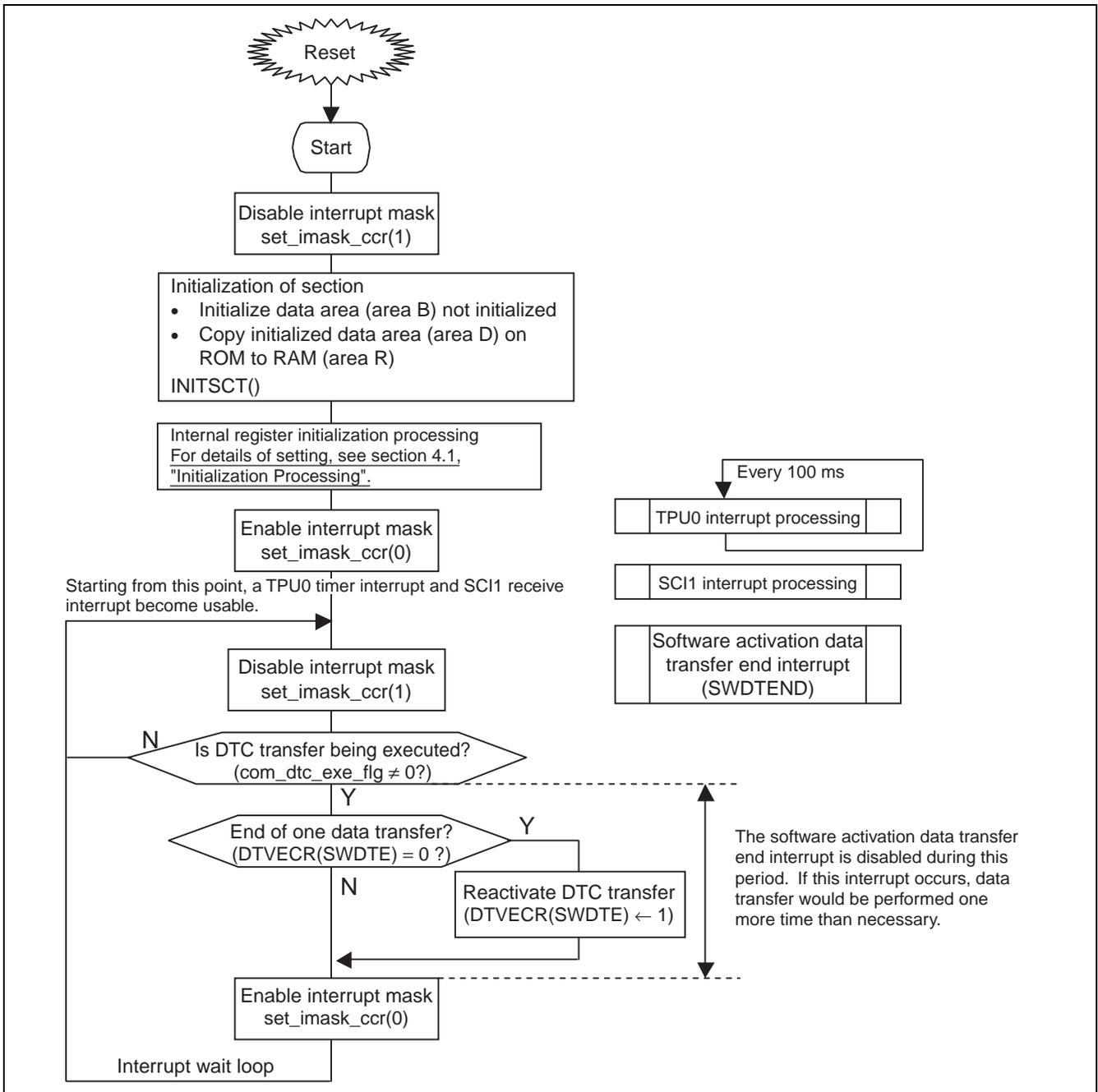
Argument	Setting
action_item	Specifies an activation source. DTC_ACT_MANUAL (0x0000): Software activation DTC_ACT_IRQ0 (0x0A80): External pin interrupt IRQ0 DTC_ACT_IRQ1 (0x0A40): External pin interrupt IRQ1 DTC_ACT_IRQ2 (0x0A20): External pin interrupt IRQ2 DTC_ACT_IRQ3 (0x0A10): External pin interrupt 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

Argument	Setting
	<p>DTC_ACT_RXI2 (0x0F80): SCI channel 2 reception interrupt</p> <p>DTC_ACT_TXI2 (0x0F40): SCI channel 2 transmission interrupt</p> <p>This application note uses only DTC_ACT_MANUAL (software activation) but supports DTC activation by an interrupt as the subroutine function.</p>
soft_vect_no	<p>Specifies a vector number when the activation source is "software activation."</p> <p>You can specify a value from 0x00 to 0x7F. The vector address used in software activation depends on the specified vector number.</p> <p style="padding-left: 20px;">Vector address = 0x400 + soft_vect_no * 2</p> <p>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 section 4, Principles of Operation.</p>

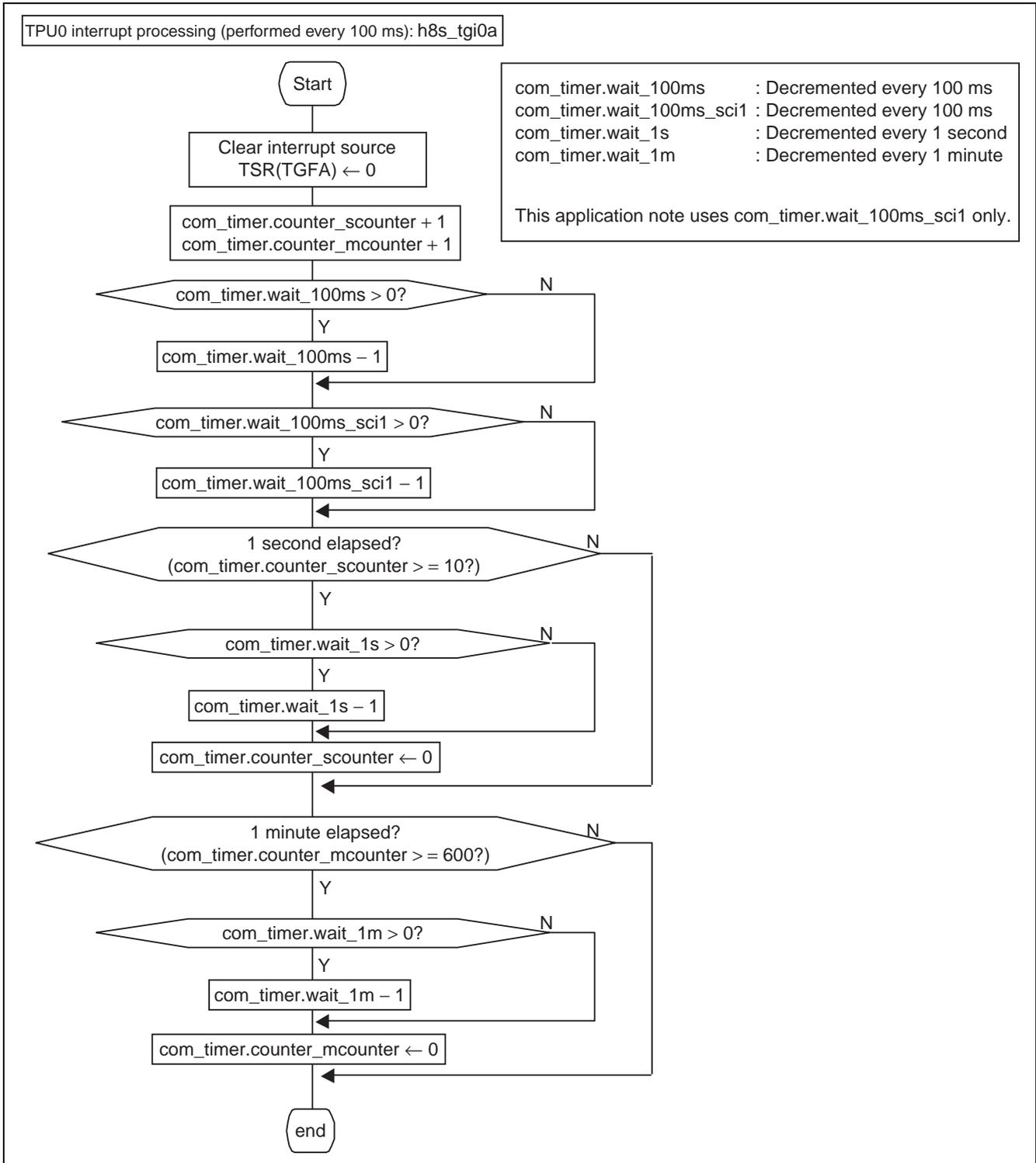
Return value	Description
0	Normal end of DTC activation
-1	DTC cannot be activated because DTC software is being activated.

6. Flowchart

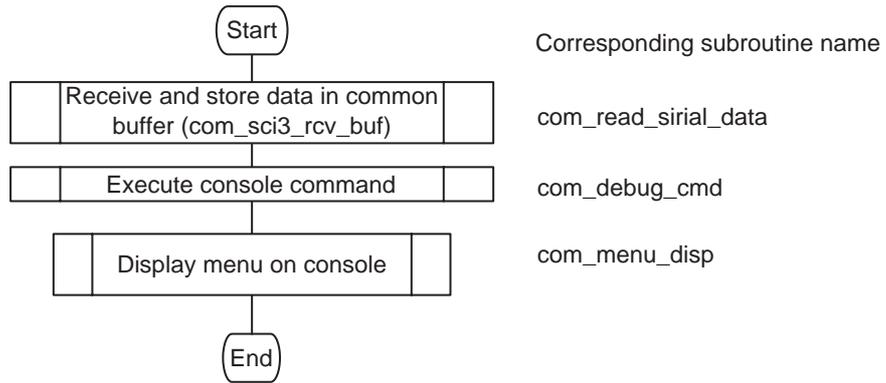
6.1 Overall Flow



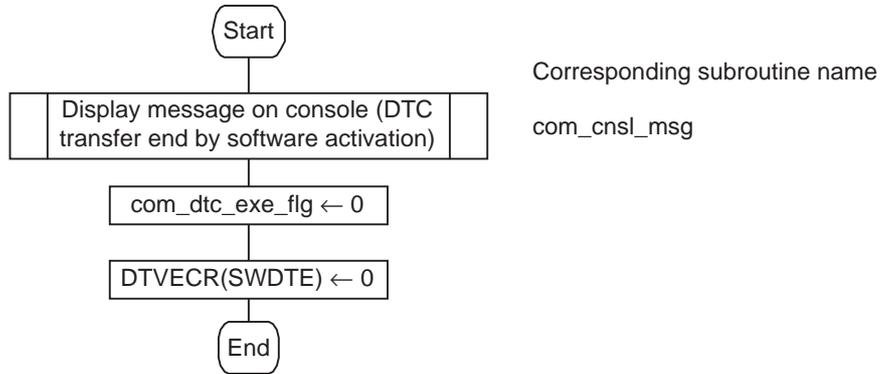
6.2 Interrupt Processing



SCI1 receive interrupt processing : h8s_sci1_rxi

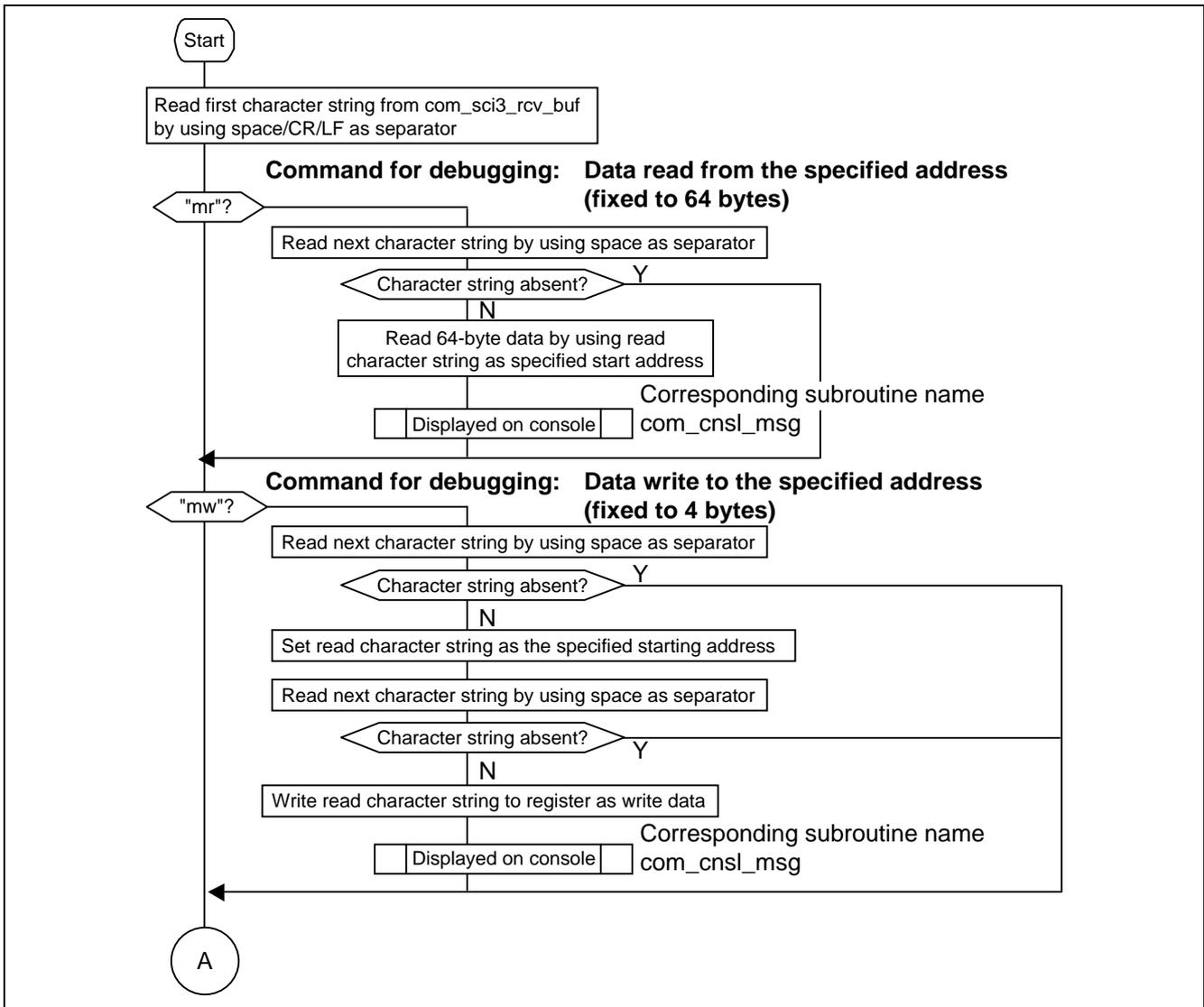


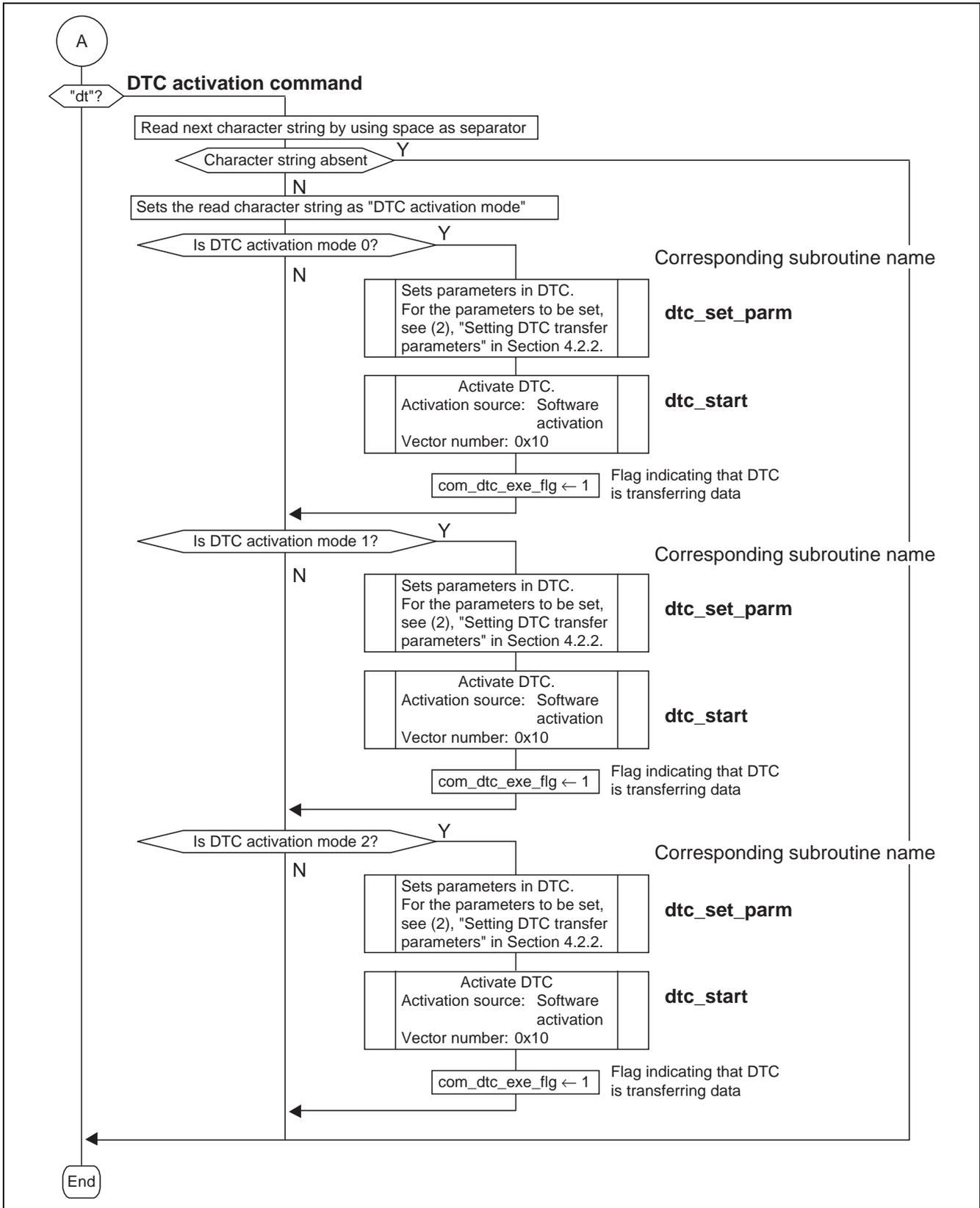
Software activation transfer end interrupt (SWDTEND) : h8s_swdtend



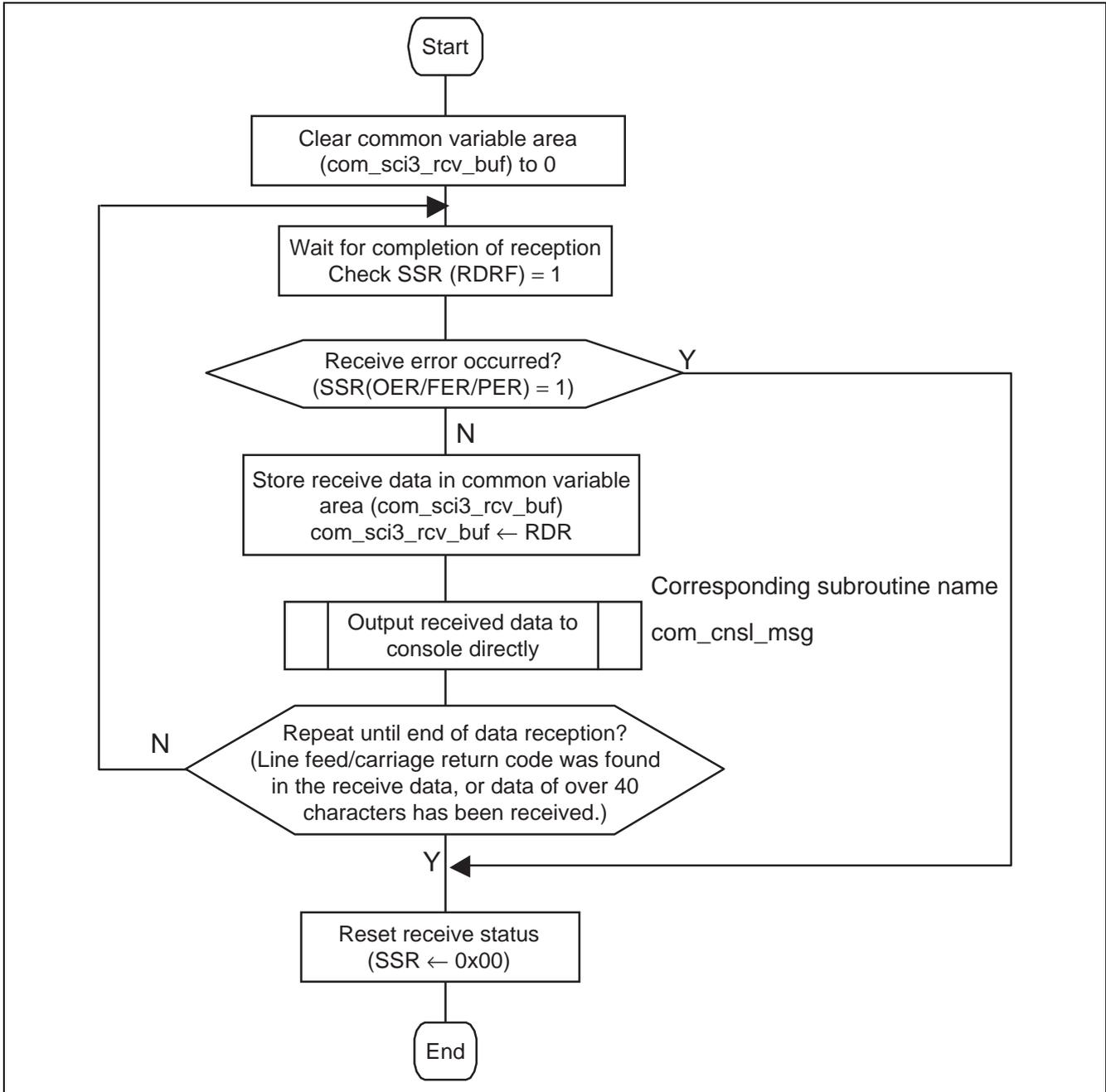
6.3 Detailed Processing

com_debug_cmd
: Console command analysis and execution

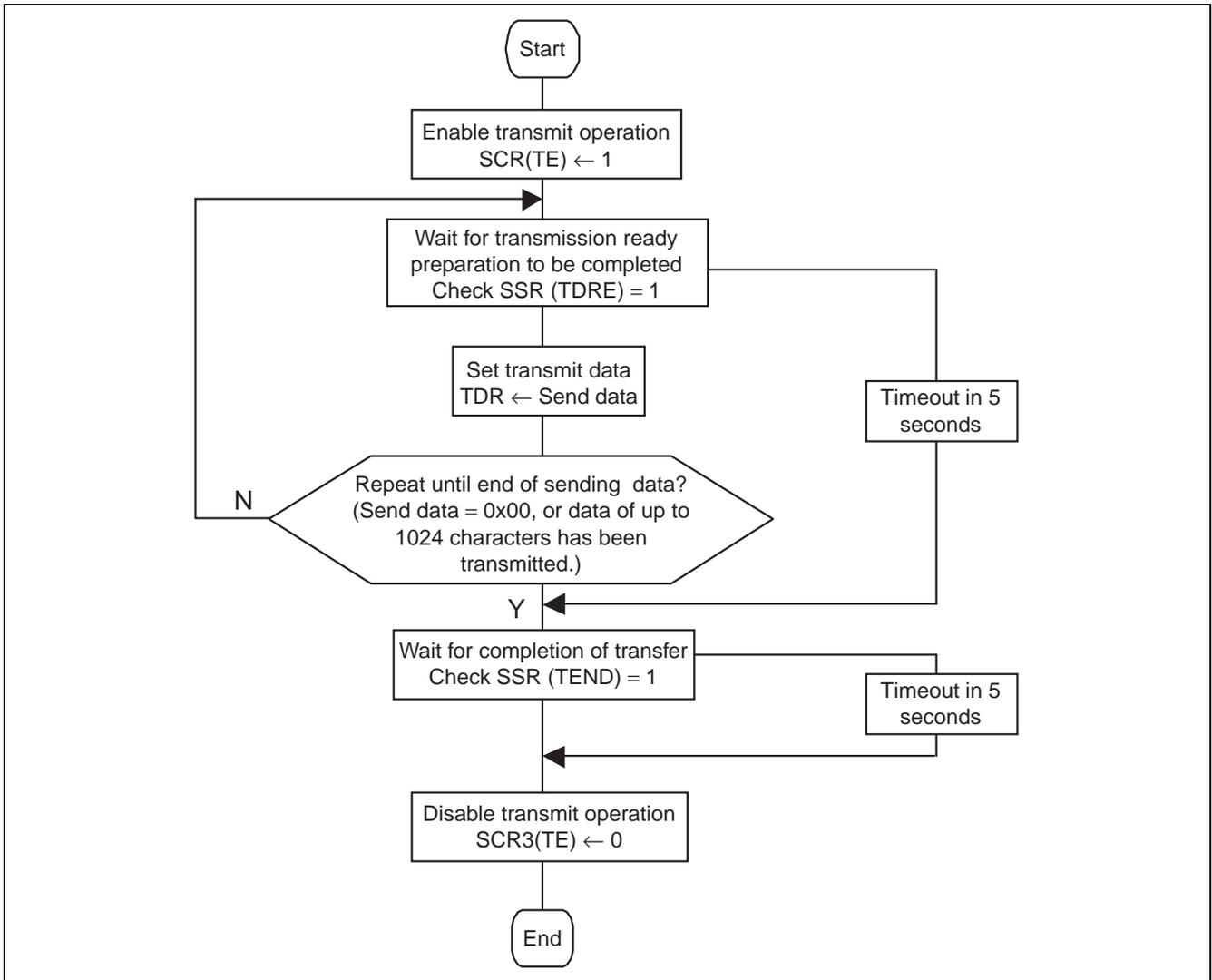




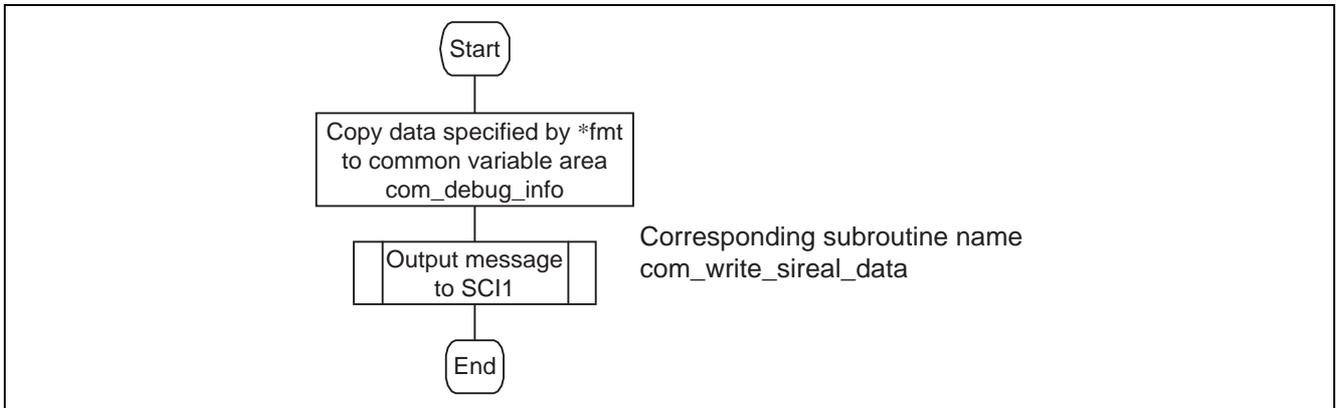
com_read_sirial_data
: Reception of a message from the SCI1 interface
Receive data not longer than 40 characters is received
in the common variable area (com_sci3_rcv_buf).



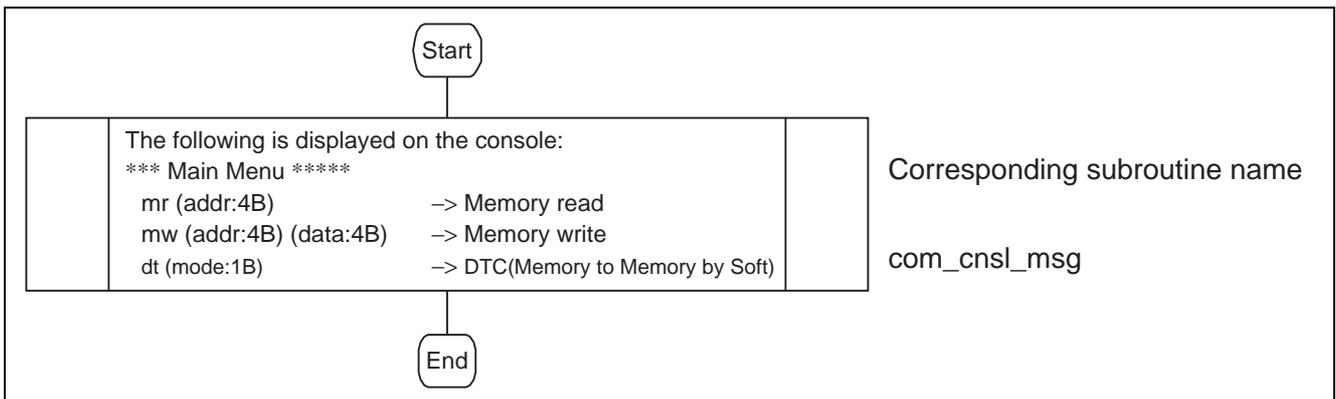
```
com_write_sireal_data ( char *p )
: Transmission of a message to the SCI3 interface
*p : Address where message data is stored
```



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



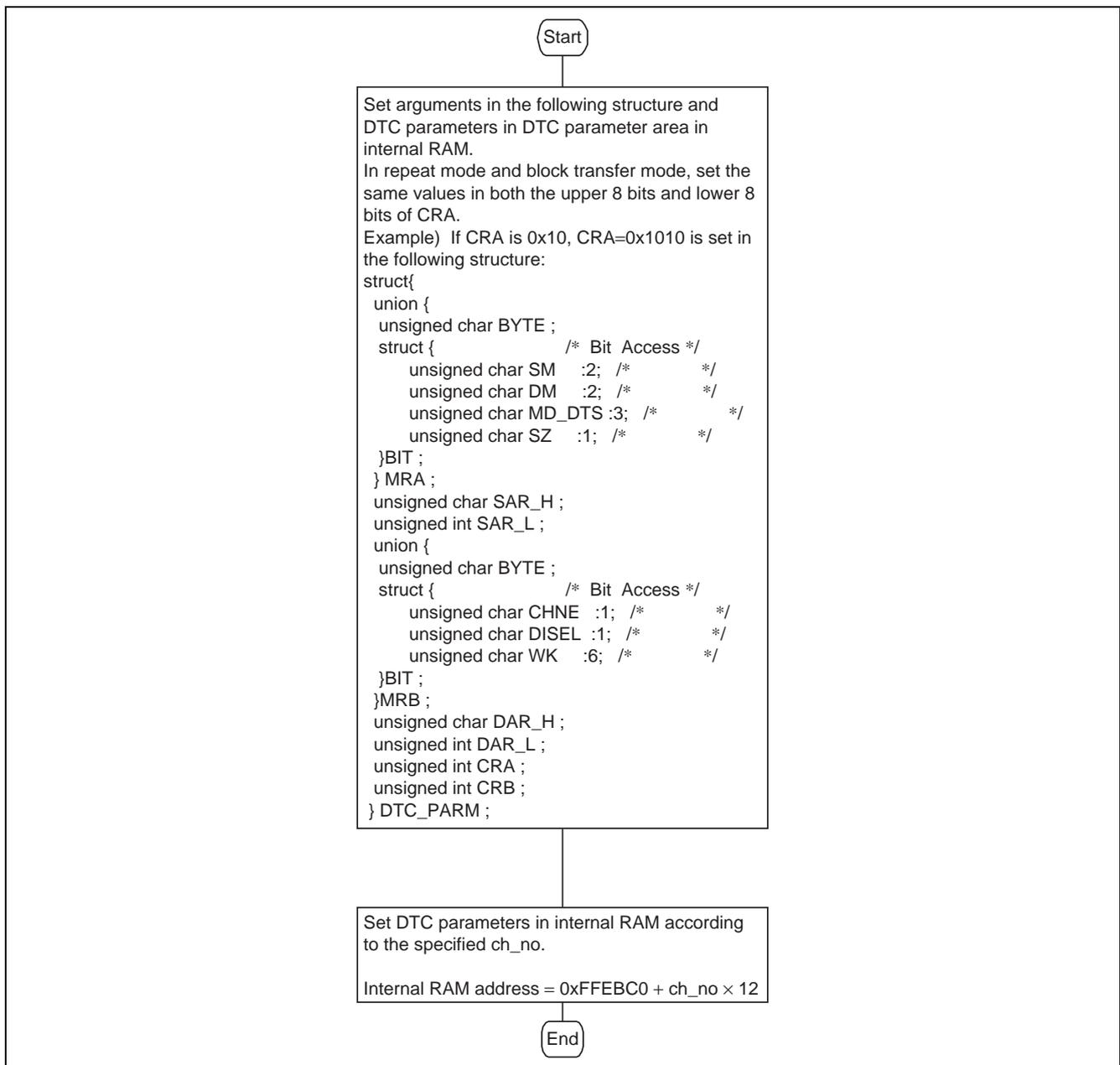
```
com_menu_disp
: Display of the operation menu on the console
```



```
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 )
```

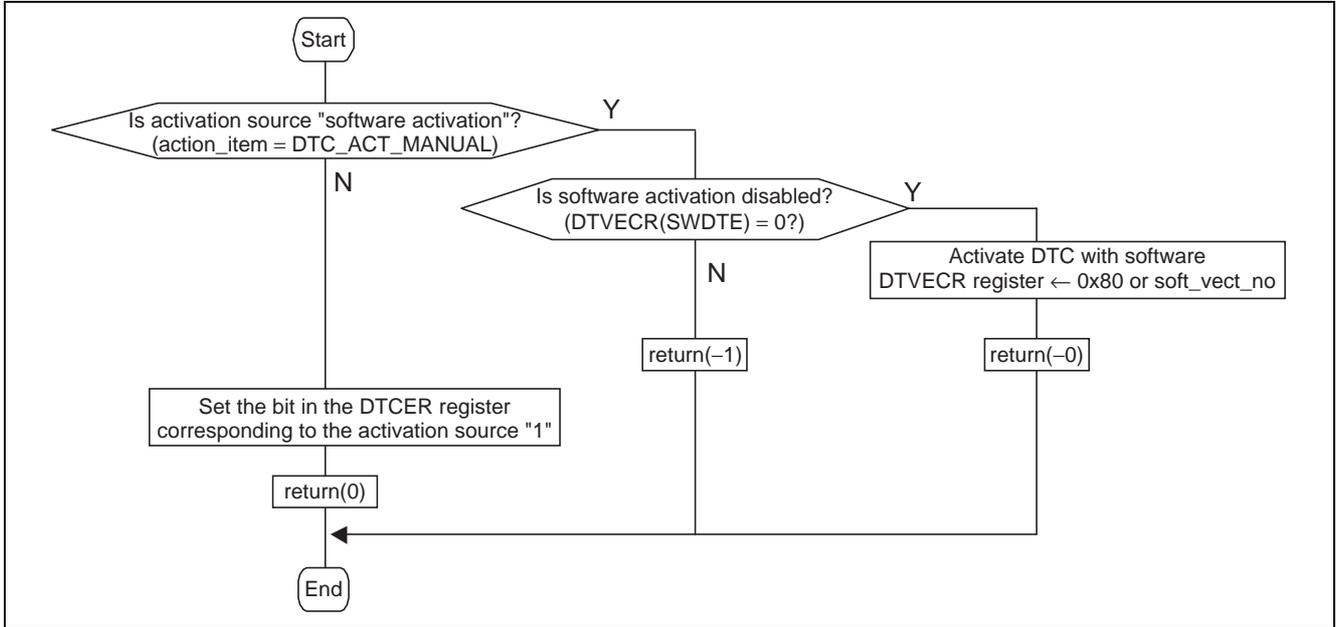
: Set the following DTC parameters.

```
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 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

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

All trademarks and registered trademarks are the property of their respective owners.

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.