SH7262/SH7264 Group

Using the Multi-function Timer Pulse Unit 2, A/D Converter and Direct Memory Access Controller

REJ06B0989-0100
Rev.1.00
May 14, 2010

Summary

This application note provides an example to use the SH7264 Multi-function Timer Pulse Unit 2, A/D Converter, and Direct Memory Access Controller.

Target Device

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

Contents

1. Introduction

1.1 Specifications

The SH7264 Multi-function Timer Pulse Unit 2 activates the A/D Converter, and SH7264 Direct Memory Access Controller transfers the A/D conversion value to the SH7264 high-speed internal RAM.

1.2 Modules Used

- Multi-function Timer Pulse Unit 2 (MTU2)
- A/D Converter (ADC)
- 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 Electronics High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Electronics 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

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 uses the Multi-function Timer Pulse Unit 2 (MTU2) to activate the A/D Converter (ADC), and transfers the A/D conversion value to high-speed internal RAM by the Direct Memory Access Controller (DMAC).

2.1 Overview of Modules

2.1.1 Multi-function Timer Pulse Unit 2 (MTU2)

The MTU2 is an advanced timer unit, consisting of five 16-bit timer channels. The compare match function, and input capture function can be specified on each channel of the MTU2. Set channels 3 and 4 of the MTU2 in reset-synchronized PWM mode or complementary PWM mode to control 6-phase PWM output. Also, use the compare match or input capture as triggers to activate the DMAC, DTC, and ADC directly, not via the CPU.

Table 1 lists the features of the MTU2. Figure 1 shows its block diagram. For more information, refer to the Multi-function Timer Pulse Unit 2 (MTU2) chapter in the SH7262 Group, SH7264 Group Hardware Manual.

Table 1 MTU2 Features

Items	Description
Number of channels	16-bit timer × 5 channels (channels 0 to 4)
Number of pulse I/O pins	Up to 16 (Channels 0, 3, 4: 4 for each, channels 1 and 2: 2 for each)
Counter clock	8 counter input clocks are available per channel
Channels operation	• Clears the counter, outputs waveforms by compare match, input capture function, synchronizing operation, PWM mode 1 (channels 0 to 4) • PWM mode 2 (channels 0 to 2) • Complementary PWM mode, reset-synchronized PWM mode (channels 3 and 4 interlocked) • AC synchronous motor drive mode (channels 0, 3, and 4 interlocked) • Phase counting mode (channels 1 and 2)
Buffer operation	Register buffer operation can be specified on channels 0, 3, and 4
Cascade connection	Connects channels 1 and 2 (16-bit counters) to use it as 32-bit counter
Activating DMAC	DMAC can be activated by channels 0 to 4 compare match or input capture
A/D converter start triggers	• Activates the ADC by the TGRA input capture and compare match, and at the trough of the TCNT_4 in complementary PWM mode • Activates the ADC by the compare match between TCNT_0, and TGRE_0 • Activates the ADC by the A/D converter start request delaying function
Operating modes	• Channels 0 to 4: PWM mode can be specified • Channels 1 and 2: Phase counting mode can be specified, respectively • Channels 3 and 4 (interlocked): Outputting six-phase PWM waveforms (three-phase positive and three-phase negative) in complementary PWM and reset-synchronized PWM modes can be specified
Interrupt requests	25 interrupt sources (e.g. Compare match interrupt, input capture interrupt, overflow interrupt, underflow interrupt)
Skipping interrupts	Channel 3 compare match interrupt, and channel 4 overflow/underflow interrupts can be skipped
Other	• High-speed access by the internal 16-bit bus • Module standby mode can be specified

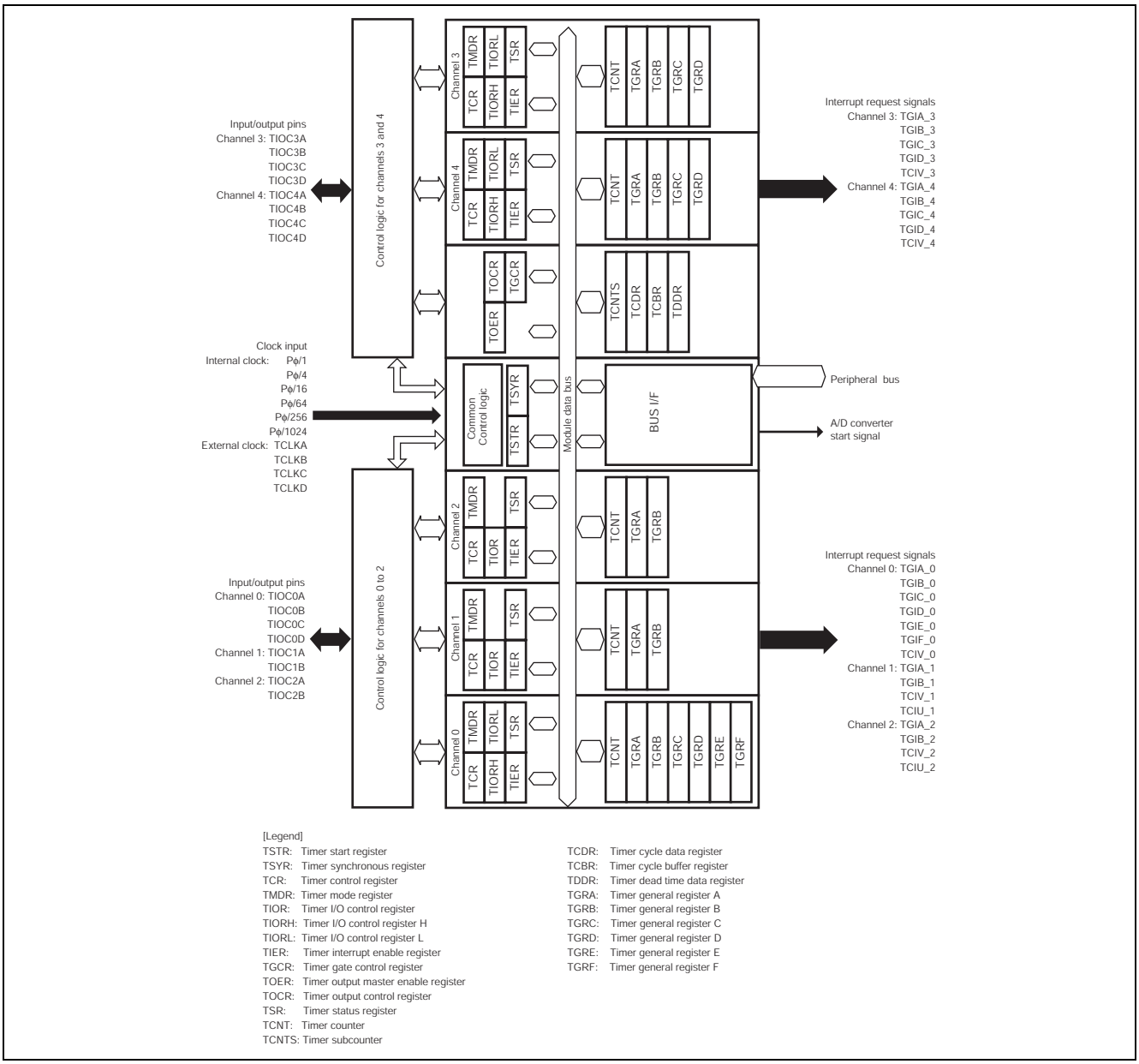


Figure 1 MTU2 Block Diagram

2.1.2 A/D Converter (ADC)

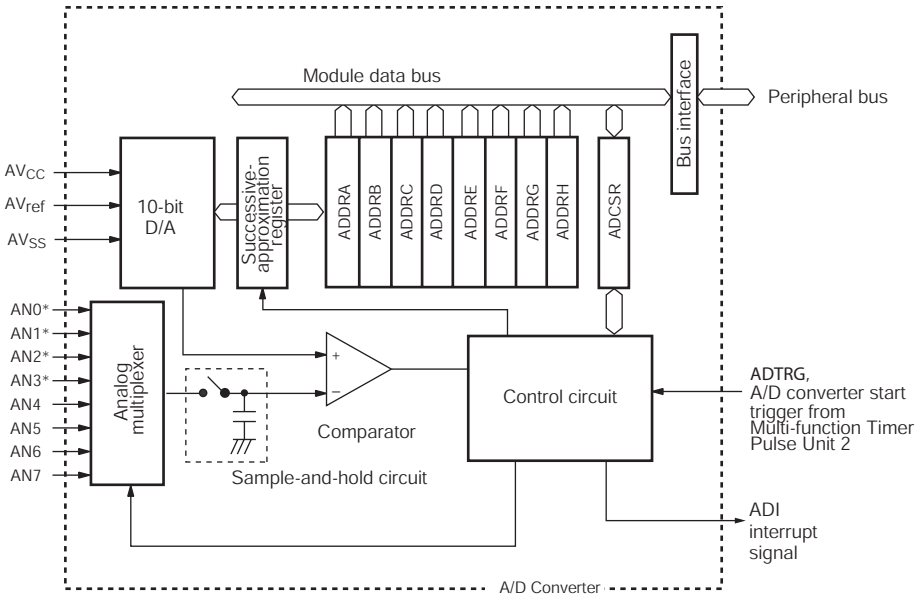
The SH7264 includes a 10-bit successive-approximation ADC. Up to eight analog input channels can be specified (up to four channels for SH7262).

ADC operates in single mode, multi mode, and scan mode. In single mode and multi mode, the ADC converts input analog voltage to digital on channels more than one specified, and enters the A/D conversion wait state. In scan mode, the ADC converts the input analog voltage to digital repeatedly on channels more than one specified. When converting analog to digital is completed, the A/D conversion end interrupt can be generated to the CPU. The DMAC can be activated when the A/D conversion end interrupt occurs. The CPU interrupt is not generated when the DMAC is activated.

Table 2 lists the features of the ADC. Figure 2 shows its block diagram. For more information, refer to the A/D Converter chapter in the SH7262 Group, SH7264 Group Hardware Manual.

Table 2 ADC Features

Item	Description
Resolution	10-bit
Number of input channels	SH7262: 4, SH7264: 8
Conversion speed	4.0 µs per channel
Absolute accuracy	± 4 LSB
Operating mode	• Single mode: A/D conversion on one channel • Multi mode: A/D conversion on channels 1 to 4, or A/D conversion on channels 1 to 8 (SH7262: channels 1 to 4) • Scan mode: Continuous A/D conversion on channels 1 to 4, or on channels 1 to 8 (SH7262: channels 1 to 4)
Number of data registers	8 (16-bit data registers for each channel store the A/D conversion value)
Sample-and-hold function	Supported
A/D conversion trigger	• Software: ADST bit setting • Conversion triggers: TRGAN, TRG0N, TRG4AN, and TRG4BN from the MTU2 • External trigger: ADTRG# pin
Interrupt source	A/D conversion end interrupt (ADI) can be generated when the A/D conversion is completed
Other	• Module standby mode can be specified • DMAC can be activated by the A/D conversion end interrupt (ADI)



[Legend]
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D
ADDRE: A/D data register E
ADDRF: A/D data register F
ADDRG: A/D data register G
ADDRH: A/D data register H

Note: * Only AN0 to AN3 can be used in the SH7262 Group.

Figure 2 ADC Block Diagram

2.1.3 Direct Memory Access Controller (DMAC)

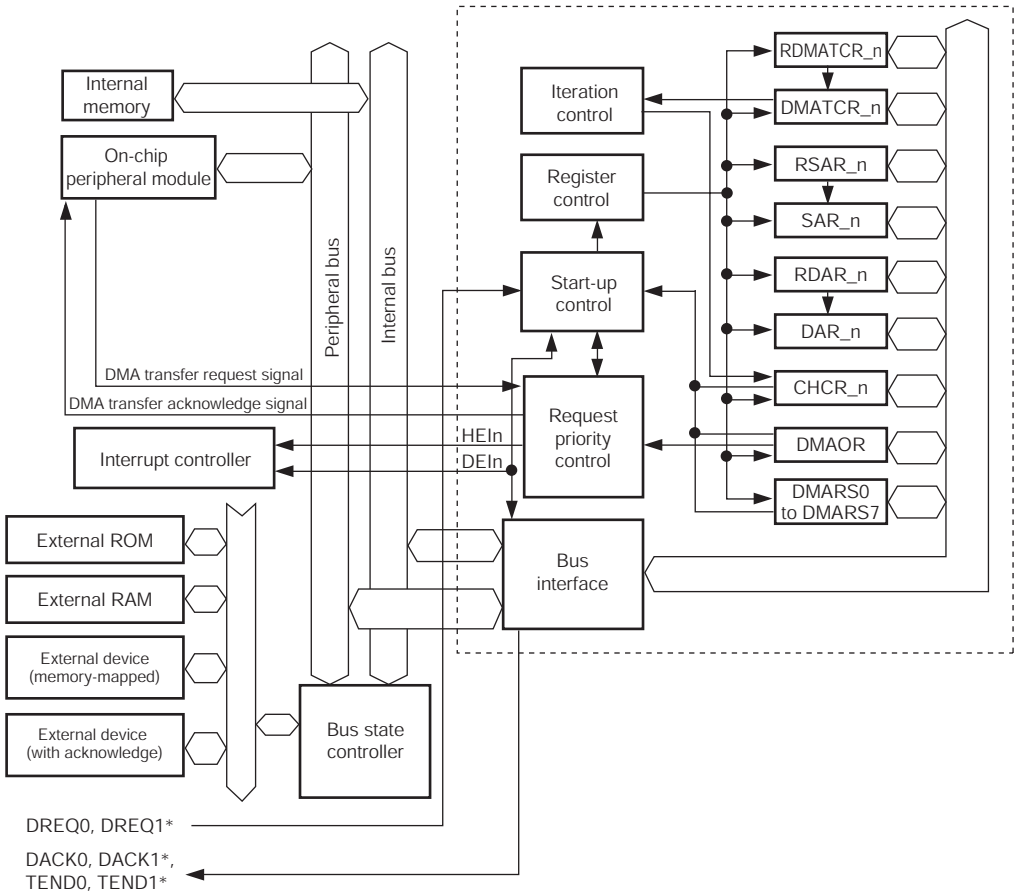
The DMAC transfers data among an external device with DACK (transfer request acknowledge signal), an external memory, internal memory, memory-mapped external device, and on-chip peripheral modules, instead of CPU. It has two bus modes; cycle steal mode and burst mode.

In cycle steal mode, the DMAC leaves the bus to other masters when it finishes "a transmit" (in bytes, words, long words, or 16 bytes). When the DMAC 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 DMAC repeats this operation until the transfer end conditions are satisfied.

Table 3 lists the features of the DMAC. Figure 3 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 DMAC Features

Items	Description
Number of channels	16
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 3 Direct Memory Access Controller Block Diagram

2.2 Configuration Procedure

2.2.1 Configuring the Multi-function Timer Pulse Unit 2 (MTU2)

This section describes how to configure the MTU2 channel 0 to activate the ADC. Specify the operating mode as PWM mode 1.

Figure 4 shows the flow chart for configuring the MTU2. For more information on register settings, refer to the SH7262 Group, SH7264 Group Hardware Manual.

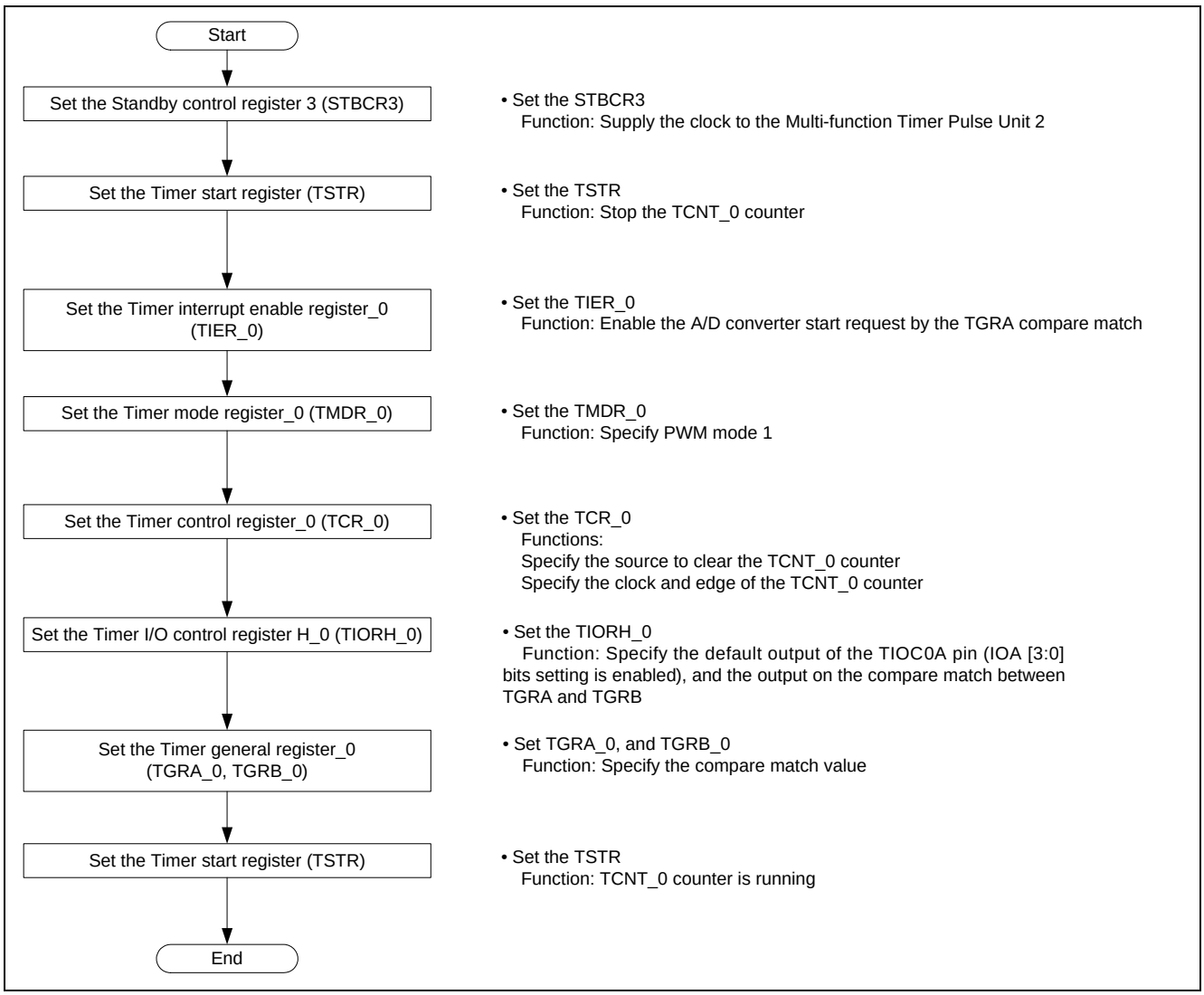


Figure 4 Flow Chart for Configuring the MTU2

2.2.2 Configuring the A/D Converter (ADC)

This section describes how to configure the ADC by the A/D converter start request from MTU2. Specify the operating mode as single mode to convert the analog input voltage AN0 to digital.

Figure 5 shows the flow chart for configuring the ADC used in this application. For more information on register settings, refer to the SH7262 Group, SH7264 Group Hardware Manual.

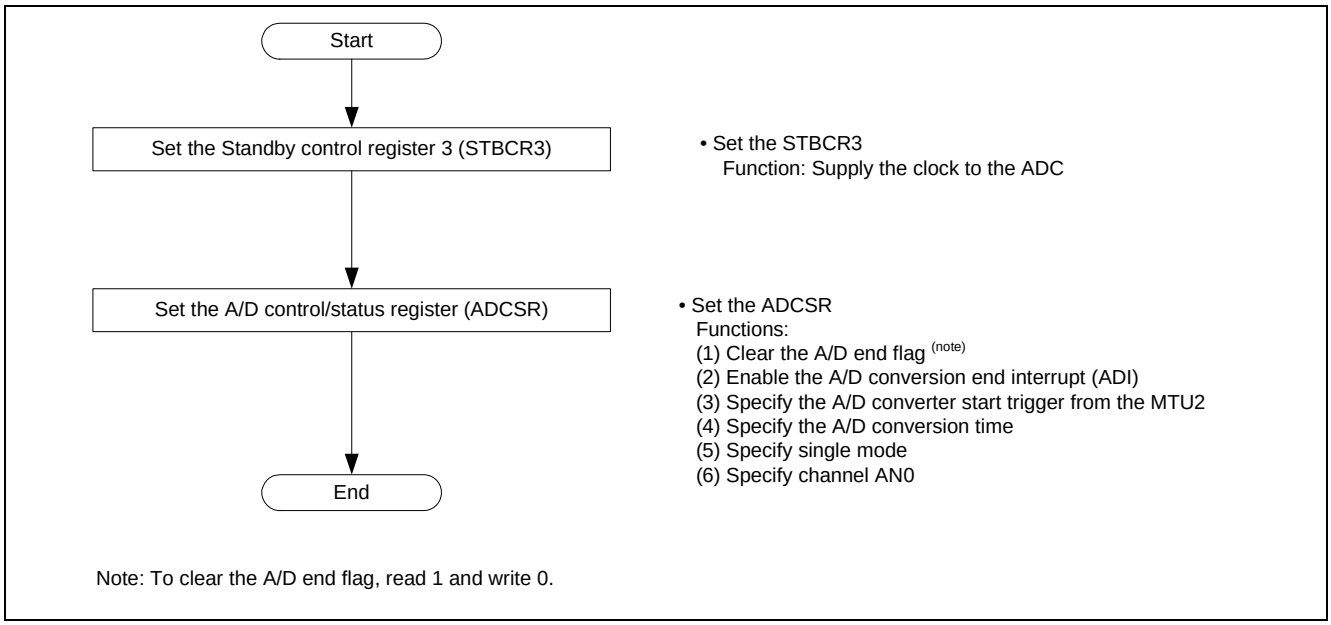


Figure 5 Flow Chart for Configuring the ADC

2.2.3 Configuring the Direct Memory Access Controller (DMAC)

This section describes how to configure the DMAC by the ADC A/D conversion end interrupt (ADI). When using the ADI as the source to activate the DMAC, only cycle steal mode can be specified.

Figure 6 shows the flow chart for configuring the DMAC. For more information on register settings, refer to the SH7262 Group, SH7264 Group Hardware Manual.

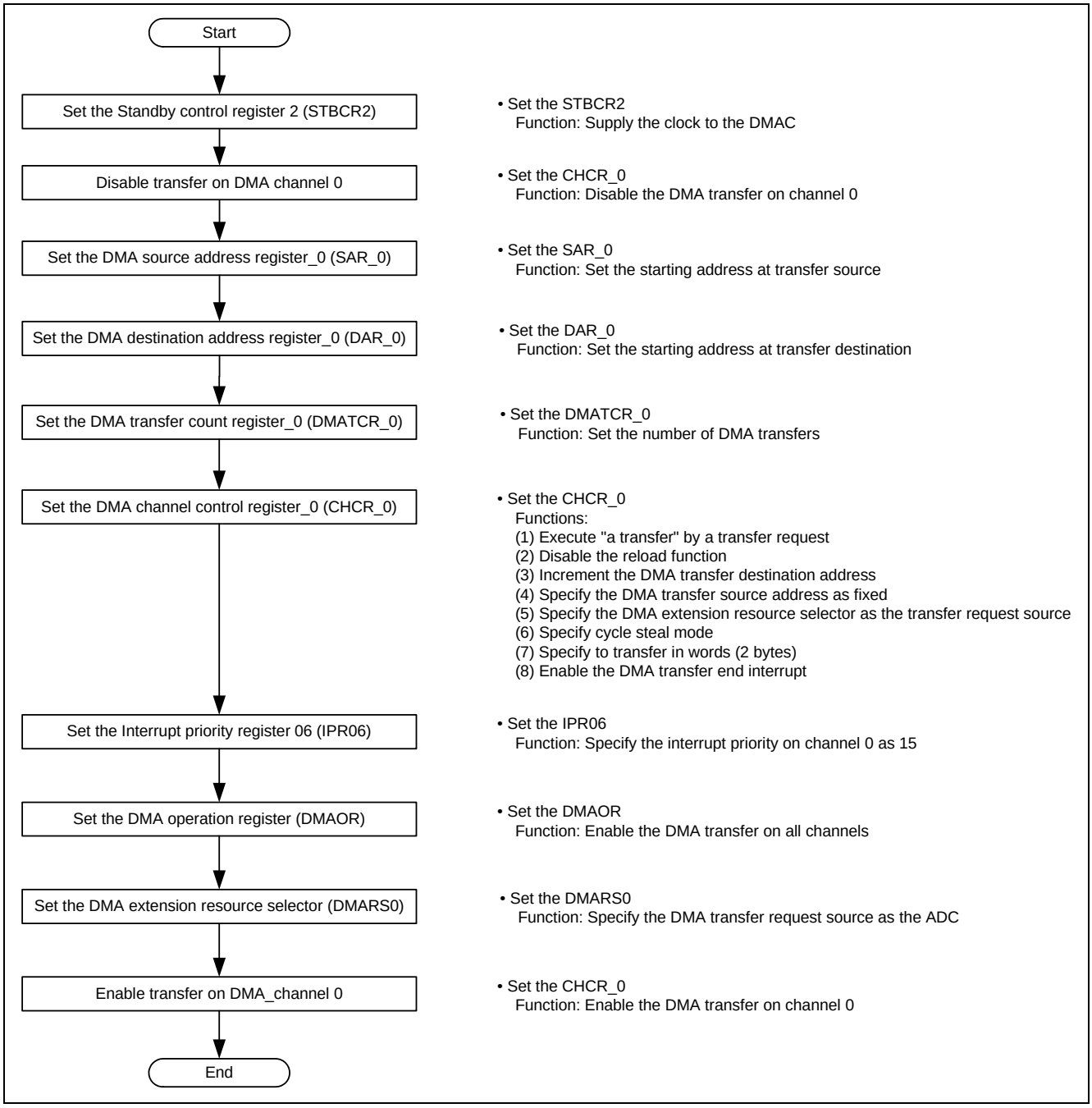


Figure 6 Flow Chart for Configuring the DMAC

2.3 Sample Program Procedure

2.3.1 Sample Program Operation

The sample program uses the MTU2 to output PWM waveform, and activates the ADC in every cycle. Then, it transfers the A/D conversion value to high-speed internal RAM by the DMAC every time the A/D conversion is completed. The DMAC stops when transferring 1-kB data is completed.

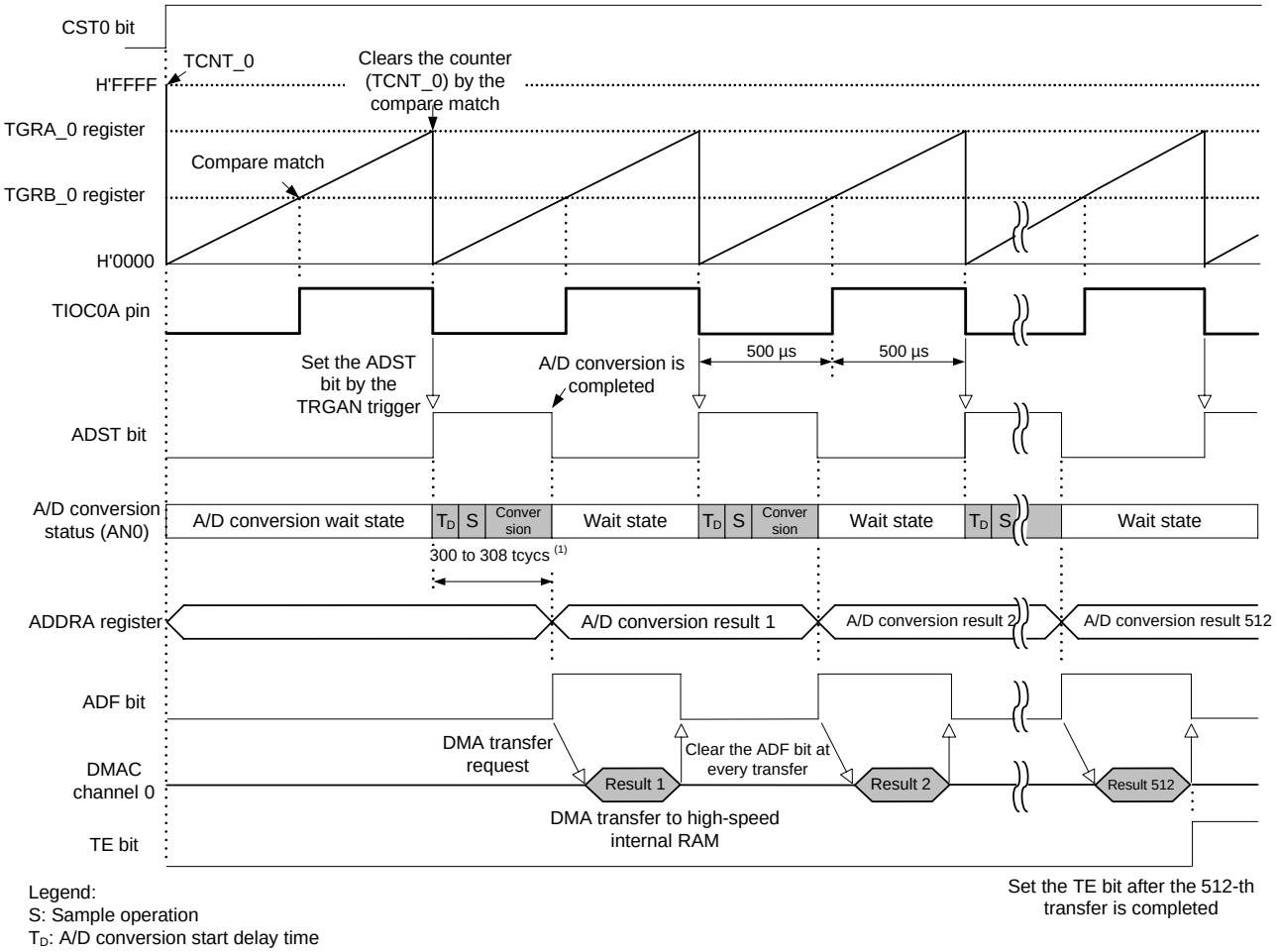
The MTU2 operates in PWM mode 1, and outputs PWM waveform (duty cycle = 50%, one cycle = 1 ms) from the TIOC0A pin. It specifies the cycle in the Timer general register A_0 (TGRA_0), and the duty cycle in the Timer general register B_0 (TGRB_0). The A/D converter start request signal (TRGAN) occurs on the compare match between the Timer counter_0 (TCNT_0) and TGRA_0 (note).

The ADC operates in single mode to start converting the analog input channel AN0 to digital by the TRGAN. The A/D conversion end interrupt signal (ADI) occurs when the A/D conversion is completed.

The DMAC operates in cycle steal mode to transfer the A/D conversion data from the A/D data register A (ADDRA) to high-speed internal RAM by the ADI. The DMA transfer end interrupt (DEI) occurs when transferring 1-KB data is completed.

Note: The A/D converter start request can be generated no matter whether the MTU2 outputs PWM waveforms or not.

Figure 7 shows sample program operation (overview).



Note 1: tcyc = 1/Bφ. Set CKS [2:0] bits in the ADCSR register to 4 µs, the minimum conversion time.

Figure 7 Sample Program Operation (Overview)

2.3.2 Multi-function Timer Pulse Unit 2 (MTU2) Register Setting

Table 4 lists the register settings for the MTU2.

Table 4 MTU2 Register Settings

Register Name	Address	Setting	Description
Standby control register 3 (STBCR3)	H'FFFE 0408	H'5A	• MSTP35 = "0": Supply the clock to the MTU2
Port G control register 3 (PGCR3)	H'FFFE 38C8	H'0002	• PG12MD = "2": Specify the PG12 pin as the TIOC0A
Port G I/O register 0 (PGIOR0)	H'FFFE 38D2	H'1000	• PG12IOR = "1": Specify the TIOC0A to output
Timer start register (TSTR)	H'FFFE 4280	H'00	• CST0 = "0": TCNT_0 counter is stopped
		H'01	• CST0 = "1": TCNT_0 counter is running
Timer mode register_0 (TMDR_0)	H'FFFE 4301	H'02	• MD [3:0] = "B'0010": Specify PWM mode 1
Timer interrupt enable register_0 (TIER_0)	H'FFFE 4304	H'80	• TTGE = "1": Enable to generate the A/D converter start request by the TGRA compare match
Timer control register_0 (TCR_0)	H'FFFE 4300	H'23	• CCLR [2:0] = "B'001": Clear the TCNT_0 by the TGRA_0 compare match or input capture • CKEG [1:0] = "B'00": Specify the TCNT_0 to count at rising edge of the input clock • TPSC [2:0] = "B'011": Specify the TCNT_0 to count when internal clock is Pφ/64
Timer I/O control register H_0 (TIORH_0)	H'FFFE 4302	H'21	• IOB [3:0] = "B'0010": Specify the TGRB to operate as the output compare register to output 0 by default (invalid when specifying PWM mode1), to output 1 by the compare match • IOA [3:0] = "B'0001": Specify the TGRA to operate as the output compare register to output 0 by default, to output 0 by the compare match
Timer general register A_0 (TGRA_0)	H'FFFE 4308	H'0232	Specify the compare value with the TCNT_0
Timer general register B_0 (TGRB_0)	H'FFFE 430A	H'0118	Specify the compare value with the TCNT_0

2.3.3 A/D Converter (ADC) Register Setting

Table 5 lists the register settings for the ADC.

The following settings enables the A/D conversion end interrupt, however, the CPU interrupt is not generated because the DMAC is activated in this application.

Table 5 ADC Register Settings

Register Name	Address	Setting	Description
Standby control register 3 (STBCR3)	H'FFFE 0408	H'5A	• MSTP32 = "0": Supply the clock to the ADC
Port H control register 0 (PHCR0)	H'FFFE 38EE	H'0001	• PH0MD = "1": Specify the PH0 pin as AN0
A/D control/status register (ADCSR)		H'4280	• ADF = "0": Clear the A/D end flag • ADIE = "1": Enable the A/D conversion end interrupt request • TRGS [3:0] = "B'0001": Activate the ADC by the TRGAN, the A/D converter start trigger from the MTU2 • CKS [2:0] = "B'010": Conversion time = 308 tcyc (note) (max.) • MDS [2:0] = "B'000": Single mode • CH [2:0] = "B'000": AN0

Note: tcyc = 1/Bφ

2.3.4 Direct Memory Access Controller (DMAC) Register Setting

Table 6 lists the register settings for the DMAC.

Table 6 DMAC Register Settings

Register Name	Address	Setting	Description
Standby control register 2 (STBCR2)	H'FFFE 0018	H'00	• MSTP8 = "0": Supply the clock to the DMAC
DMA source address register_0 (SAR_0)	H'FFFE 1000	H'FFFF 9800	Transfer source address: ADDRA register address
DMA destination address register_0 (DAR_0)	H'FFFE 1004	H'FFF8 4000	Transfer destination address: Starting address in high-speed internal RAM (Transfer destination)
DMA transfer count register_0 (DMATCR_0)	H'FFFE 1008	H'0000 0200	Number of transfers: 512
DMA channel control register_0 (CHCR_0)	H'FFFE 100C	H'0000 0000	• DE = "0": Disable the DMA transfer
		H'0000 480C	• TC = "0": Execute "a transfer" by a DMA transfer request • RLDSAR = "0", RLDDAR = "0": Disable the reload function • DM [1:0] = "B'01": Increment the destination address • SM [1:0] = "B'00": Specify the source address as fixed • RS [3:0] = "B'1000": Specify the DMA extension resource selector • TB = "0": Specify cycle steal mode • TS [1:0] = "B'01": Specify to transfer data in words • IE = "1": Enable the interrupt request
		H'0000 480D	• DE = "1": Enable the DMA transfer on channel 0
DMA operation register (DMAOR)	H'FFFE 1200	H'0001	• DME = "1": Enable the DMA transfer on all channels
DMA extension resource selector 0 (DMARS0)	H'FFFE 1300	H'00B3	• CH0 MID [5:0] = "B'1011 00": • CH0 RID [1:0] = "B'11": Specify A/D module_0 as the DMA transfer request source
Interrupt priority register 06 (IPR06)	H'FFFE 0C00	H'F000	Transfer end interrupt level: 15

2.3.5 Sample Program Flow Chart

Figure 8 shows the flow chart of the sample program.

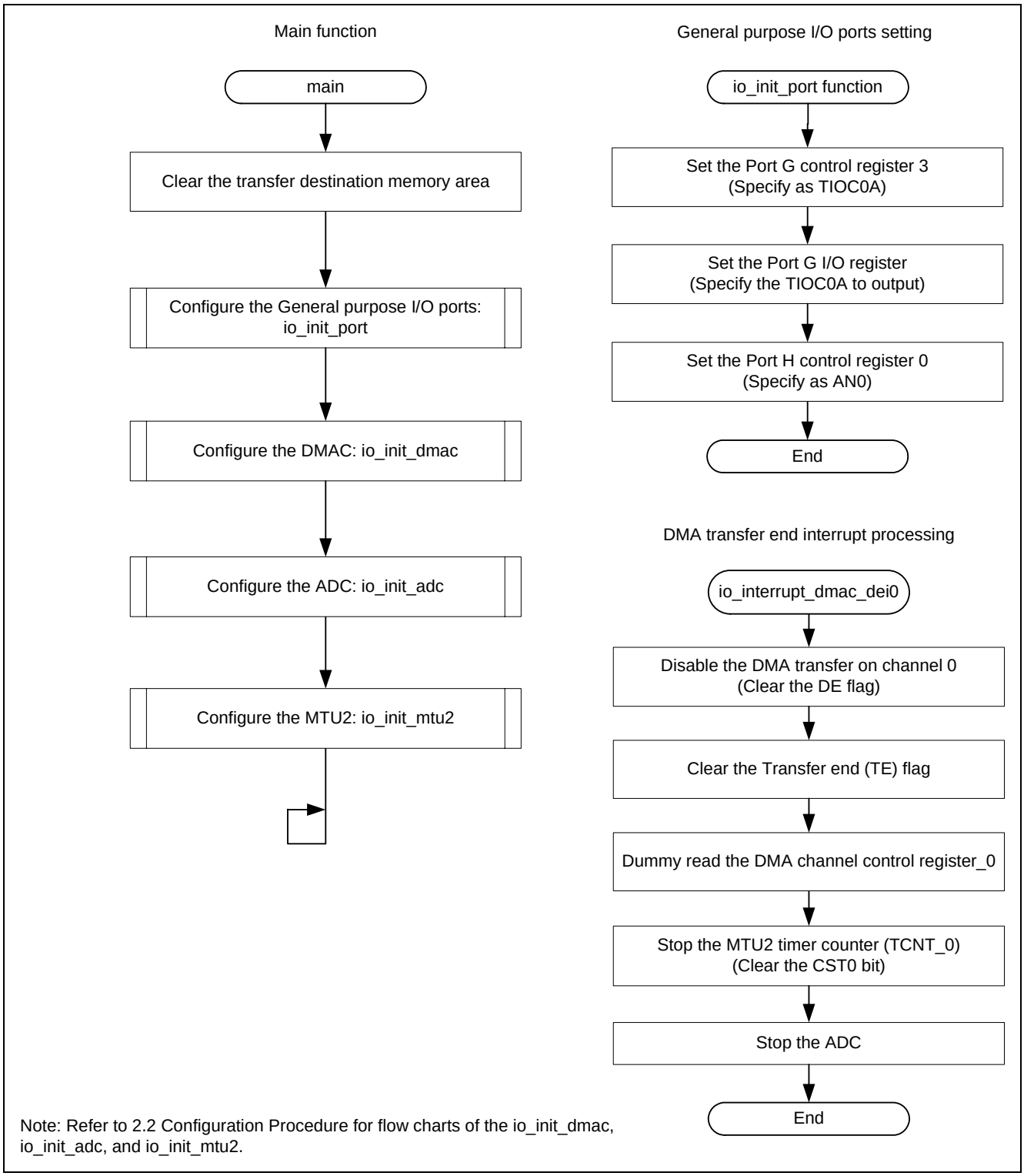


Figure 8 Sample Program Flow Chart

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" (1/10)

```
1       /***********************************************************************
2       *   DISCLAIMER
3       *
4       *   This software is supplied by Renesas Electronics Corp. and is only
5       *   intended for use with Renesas products. No other uses are authorized.
6       *
7       *   This software is owned by Renesas Electronics Corp. and is protected under
8       *   all applicable laws, including copyright laws.
9       *
10      *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11      *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12      *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13      *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14      *   DISCLAIMED.
15      *
16      *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17      *   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18      *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19      *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20      *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21      *
22      *   Renesas reserves the right, without notice, to make changes to this
23      *   software and to discontinue the availability of this software.
24      *   By using this software, you agree to the additional terms and
25      *   conditions found by accessing the following link:
26      *   http://www.renesas.com/disclaimer
27      ***********************************************************************
28      * (C) 2010 Renesas Electronics Corporation. All rights reserved.
29      *""FILE COMMENT""*********** Technical reference data *************************
30      *   System Name : SH7264 Sample Program
31      *   File Name   : main.c
32      *   Abstract    : MTU2+ADC+DMAC Module Application
33      *   Version     : 1.00.01
34      *   Device      : SH7262/SH7264
35      *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36      *               : C/C++ compiler package for the SuperH RISC engine family
37      *               :                              (Ver.9.03 Release00).
38      *   OS          : None
39      *   H/W Platform: M3A-HS64G50(CPU board)+M3A-HS64G02(IO board)
40      *   Description :
41      ***********************************************************************
42      *   History     : Feb.16,2010 ver.1.00.00
43      *               : Apr.15,2010 Ver.1.00.01 Changed the company name
44      *""FILE COMMENT END""*****************************************************/
45      #include <string.h>
46      #include "iodefine.h"       /* SH7264 iodefine  */
```

3.3 Sample Program Listing "main.c" (2/10)

```
47
48    /* ---- Prototype declaration ---- */
49    void main(void);
50    void io_init_port(void);
51    void io_init_dmac(void *src, void *dst, int count);
52    void io_init_adc(void);
53    void io_init_mtu2(void);
54    void io_interrupt_dmac_dei0( void );
55
56    /* ---- Macro definition ---- */
57    #define NUM_OF_BUFF        10          /* Number of the receive buffers */
58    #define DST_ADR        0xfff84000 /* Starting address at DMA transfer destination
59                                  (high-speed internal RAM) */
60    #define SRC_ADR        0xffff9800 /* DMA transfer source (A/D data register A) address */
61    #define COUNT          0x200      /* Number of transfers: 512 (1 word/1 transfer) */
62
63
64
```

3.4 Sample Program Listing "main.c" (3/10)

```
65    /*""FUNC COMMENT""*****************************************************************
66     * ID            :
67     * Outline       : Sample program main
68     *-------------------------------------------------------------------------------
69     * Include       : <string.h>
70     *-------------------------------------------------------------------------------
71     * Declaration   : void main(void);
72     *-------------------------------------------------------------------------------
73     * Description   : After clearing internal memory storing data, configure PORT,
74     *               : DMAC, ADC, and MTU2. Then, the MTU2 TCNT starts counting,
75     *               : starts the A/D conversion by the TCNT compare match to transfer
76     *               : the conversion data to internal RAM by DMAC. After 1-KB data is
77     *               : transferred by the DMA, the DMA transfer end interrupt occurs
78     *               : to the CPU. A series of processing is executed by hardware.
79     *-------------------------------------------------------------------------------
80     * Argument      : void
81     *-------------------------------------------------------------------------------
82     * Return Value  : void
83     *-------------------------------------------------------------------------------
84     * Note          :
85     *""FUNC COMMENT END""*************************************************************/
86    void main(void)
87    {
88      /* ==== Clears the transfer destination memory area ==== */
89      memset((void *)DST_ADR, 0, COUNT*2);
90
91      /* ==== Sets general purpose I/O ports ==== */
92      io_init_port();
93
94      /* ==== Configures the DMAC ==== */
95      io_init_dmac((void *)SRC_ADR, (void *)DST_ADR, COUNT);
96
97      /* ==== Configures the ADC ==== */
98      io_init_adc();
99
100     /* ==== Configures the MTU2 ==== */
101     io_init_mtu2();
102
103     while(1){
104       /* loop */
105     }
106   }
```

3.5 Sample Program Listing "main.c" (4/10)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : General purpose I/O ports setting
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void io_init_port(void);
 *-----------------------------------------------------------------------------
 * Description   : Sets the pin function, and the direction of data (I/O)
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""*************************************************************/
void io_init_port(void)
{
  /* ==== Sets the MTU2 (TIOC0A pin to output) ==== */
  PORT.PGCR3.BIT.PG12MD = 2;      /* Specifies the PG12 pin as the TIOC0A */
  PORT.PGIOR0.BIT.PG12IOR = 1;/* Specifies the PG12 pin to output */
                              /* Specifies the direction of data when
                                  specifying the MTU2 */
                              /* After this setting, the pin outputs
                                  low level signal */

  /* ==== Sets the ADC (AN0 input) ==== */
  PORT.PHCR0.BIT.PH0MD = 1;       /* Specifies the PH0 pin as AN0 */
}
```

3.6 Sample Program Listing "main.c" (5/10)

```
137  /*""FUNC COMMENT""*****************************************************************
138   * ID            :
139   * Outline       : DMAC configuration
140   *-------------------------------------------------------------------------------
141   * Include       : "iodefine.h"
142   *-------------------------------------------------------------------------------
143   * Declaration   : void io_init_dmac(void *src, void *dst, int count);
144   *-------------------------------------------------------------------------------
145   * Description   : Configures the Direct Memory Access Controller (DMAC).
146   *               : Operating mode: Cycle steal mode
147   *               : On-chip peripheral module request: ADC
148   *               : Transfer source: A/D data register A, destination: High-speed
149   *               : internal RAM, transfer data length: In words.
150   *               : Reload function is not used.
151   *-------------------------------------------------------------------------------
152   * Argument      : void *src ; I : Transfer source address
153   *               : void *dst ; O : Transfer destination address
154   *               : int count ; I : Number of transfers
155   *-------------------------------------------------------------------------------
156   * Return Value  : void
157   *-------------------------------------------------------------------------------
158   * Note          :
159   *""FUNC COMMENT END""*************************************************************/
160  void io_init_dmac(void *src, void *dst, int count)
161  {
162    volatile unsigned long dummy;
163
164    /* ==== Sets the Standby control register 2 (STBCR2) ==== */
165    CPG.STBCR2.BIT.MSTP8 = 0;              /* Supplies the clock to the DMAC */
166
167    /* ==== Disables the DMA transfer on channel 0 ==== */
168    DMAC.CHCR0.BIT.DE = 0x0;            /* DMA transfer disabled */
169
170    /* ==== Sets the DMA source address register_0 (SAR_0) ==== */
171    DMAC.SAR0.LONG = (unsigned long)src;
172
173    /* ==== Sets the DMA destination address register_0 (DAR_0) ==== */
174    DMAC.DAR0.LONG = (unsigned long)dst;
175
176    /* ==== Sets the DMA transfer count register_0 (DMATCR_0) ==== */
177    DMAC.DMATCR0.LONG = count;
178
```

3.7 Sample Program Listing "main.c" (6/10)

```
179     /* ==== Sets the DMA channel control register_0 (CHCR_0) ==== */
180     dummy = DMAC.CHCR0.LONG;    /* Execute dummy read to make sure to clear the TE flag */
181     DMAC.CHCR0.LONG = 0x0000480c;
182                 /*
183                         bit 31    : TC DMATCR transfer: 0-Executes "a transfer"
184                                                           by a transfer request
185                         bit 30    : reserve 0
186                         bit 29    : RLDSAR ON: 0--------  Disables the
187                                                           reload function (RSAR).
188                         bit 28    : RLDDAR ON: 0--------  Disables the reload function (RDAR)
189                         bit 27    : reserve 0
190                         bit 26    : DAF: 0--------------  Not used
191                         bit 25    : SAF: 0--------------  Not used
192                         bit 24    : reserve 0
193                         bit 23    : DO over run0: 0-----  Not used
194                         bit 22    : TL TEND low active: 0-Not used
195                         bit 21    : reserve 0
196                         bit 20    : TEMASK: 0-----------  Not used
197                         bit 19    : HE: 0---------------  Not used
198                         bit 18    : HIE: 0--------------  Not used
199                         bit 17    : AM: 0---------------  Not used
200                         bit 16    : AL: 0---------------  Not used
201                         bits 15 to 14: DM1:0 DM0:1------  Increments the destination address
202                         bits 13 to 12: SM1:0 SM0:0------  Specifies the
203                                                           source address as fixed
204                         bits 11 to 8: RS : B'1000-------  Specifies the DMA extension
205                                                           resource selector
206                         bit 7     : DL : DREQ level: 0 -  Not used
207                         bit 6     : DS : DREQ select: 0 Low level Not used
208                         bit 5     : TB :cycle :0--------  Cycle steal mode
209                         bits 4 to 3: TS : transfer size: B'01-    Specifies to transfer
210                                                              data in words
211                         bit 2     : IE : interrupt enable: 1---   Enables the interrupt
212                         bit 1     : TE : transfer end: 0-------   Clears the TE flag
213                                                                 (Clear the flag to 0 after
214                                                                 reading 1)
215                         bit 0     : DE : DMA enable bit: 0-----   Disables the DMA transfer
216                 */
217     /* ======= Sets the Interrupt priority register 06 (IPR06) ======= */
218     INTC.IPR06.BIT._DMAC0 = 0xf;/* Specifies the interrupt priority as 15 */
219
220     /* ---- Sets the DMA operation register (DMAOR) ---- */
221     DMAC.DMAOR.WORD |= 0x0007;      /* Sets the DME bit. To avoid clearing */
222                                 /* address error and NMI flags, writes 1 */
223                                 /* to bits AE and NMIF */
224     /* ---- Sets the DMA extension resource selector (DMARS0) ---- */
225     DMAC.DMARS0.BYTE.CH0 = 0xB3;    /* Specifies the transfer request source as the ADC */
226
227     /* ======= Enables the DMA transfer on channel 0 ======= */
228     DMAC.CHCR0.BIT.DE = 0x1;
229   }
```

3.8 Sample Program Listing "main.c" (7/10)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : ADC configuration
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void init_adc(void);
 *-----------------------------------------------------------------------------
 * Description   : Configures the A/D Converter (ADC).
 *               : Operating mode: Single more, A/D activation source: TRGAN (MTU2)
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""*************************************************************/
void io_init_adc(void)
{
  volatile unsigned short dummy;

  /* ==== Sets the Standby control register 3 (STBCR3) ==== */
  CPG.STBCR3.BIT.MSTP32 = 0;      /* Supplies the clock to the ADC */

  /* ==== Sets the A/D control/status register (ADCSR) ==== */
  dummy = ADC.ADCSR.WORD;         /* Dummy read to clear the ADF flag */
  ADC.ADCSR.WORD = 0x4280;
              /*
                  bit 15   : ADF:0------------- Clears the A/D end flag
                  bit 14   : ADIE:1------------ Enables the A/D conversion
                                                end interrupt (ADI).
                  bit 13   : ADST:0------------ Stops the A/D converter
                  bits 12 to 9: TRGS[3:0]:B'0001 - TRGAN, A/D converter start
                                                   trigger from the MTU2
                  bits 8 to 6: CKS[2:0]:B'010 --- Conversion time = 308*tcyc > 4 us
                                                  (B-clock is at 72 MHz)
                  bits 5 to 3: MDS[2:0]:B'000 --- Single mode
                  bits 2 to 0: CH[2:0]:B'000 ---- Channel: AN0
              */
}
```

3.9 Sample Program Listing "main.c" (8/10)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : MTU2 configuration
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void init_mtu2(void);
 *-----------------------------------------------------------------------------
 * Description   : Configures the Multi-function Timer Pulse Unit 2 (MTU2).
 *               : Operating mode: PWM mode 1
 *               : TCNT: Count when the internal clock is P-clock/64
 *               : Requests the A/D converter start.
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""*************************************************************/
void io_init_mtu2(void)
{
  /* ==== Sets the Standby control register (STBCR3) ==== */
  CPG.STBCR3.BIT.MSTP35 = 0;      /* Supplies the clock to the MTU2 */

  /* ==== Sets the Timer control register (TSTR) ==== */
  MTU2.TSTR.BIT.CST0 = 0;         /* Specifies the timer counter (TCNT_0) to stop */

  /* ==== Sets the Timer mode register_0 (TMDR_0) ==== */
  MTU2.TMDR_0.BYTE = 0x02;    /* Specifies PWM mode 1 */

  /* ==== Sets the Timer interrupt enable register (TIER_0) ==== */
  MTU2.TIER_0.BYTE = 0x80;    /* Enables the A/D converter start request
                                 by the TGRA input capture/compare match */

  /* ==== Sets the Timer control register (TCR_0) ==== */
  MTU2.TCR_0.BYTE = 0x23;
          /*
              bits 7 to 5: CCLR:B'001---- Clears the TCNT by the TRGA compare match
              bits 4 to 3: CKEG:B'00----- Specifies the TCNT to count at rising edge
                                          of the input clock
              bits 2 to 0: TPSC:B'011---- Specifies the TCNT to count when internal
                                          clock is P-clock/64
          */

```

3.10 Sample Program Listing "main.c" (9/10)

```
314     /* ==== Sets the Timer I/O control register H_0 (TIORH_0) ==== */
315     MTU2.TIORH_0.BYTE = 0x21;
316               /*
317                   bits 7 to 4: IOB:B'0010---Specifies the TGRB to operate as the output
318                                       compare register to output 0 by default (invalid
319                                       when specifying PWM mode 1), to output 1 by the
320                                       compare match
321                   bits 3 to 0: IOA:B'0001---Specifies the TGRA to operate as the output
322                                       compare register to output 0 by default, to
323                                       output 0 by compare match
324               */
325     /* ==== Sets the Timer general register (TGRA_0) ==== */
326     MTU2.TGRA_0.WORD = 562;         /* f (1 kHz) = TCNT counter clock (36 MHz/64)/(N+1) */
327                                 /* N = 562 (0x232) */
328     /* ==== Sets the Timer general register (TGRB_0) ==== */
329     MTU2.TGRB_0.WORD = 280;         /* f (2 kHz) = TCNT counter clock (36 MHz/64)/(N+1) */
330                                 /* N = 280 (0x118) */
331     /* ==== Sets the Timer start register (TSTR) ==== */
332     MTU2.TSTR.BIT.CST0 = 1;         /* Specifies the timer counter (TCNT_0) to start */
333   }
334
```

3.11 Sample Program Listing "main.c" (10/10)

```
335  /*""FUNC COMMENT""*****************************************************************
336   * ID            :
337   * Outline       : DMA transfer end interrupt (channel 0).
338   *-----------------------------------------------------------------------------
339   * Include       : "iodefine.h"
340   *-----------------------------------------------------------------------------
341   * Declaration   : void io_interrupt_dmac_dei0(void);
342   *-----------------------------------------------------------------------------
343   * Description   : Handles the transfer end interrupt on DMAC channel 0.
344   *               : Stops the DMA, and clears the TEND flag.
345   *               : Then, it stops the MTU2 timer counter.
346   *-----------------------------------------------------------------------------
347   * Argument      : void
348   *-----------------------------------------------------------------------------
349   * Return Value  : void
350   *-----------------------------------------------------------------------------
351   * Note          :
352   *""FUNC COMMENT END""*************************************************************/
353  void io_interrupt_dmac_dei0( void )
354  {
355    volatile unsigned long dummy;
356
357    /* ==== Disables the DMA transfer ==== */
358    DMAC.CHCR0.BIT.DE = 0;
359
360    /* ==== Clears the TEND flag ==== */
361    DMAC.CHCR0.BIT.TE = 0;
362    dummy = DMAC.CHCR0.LONG;/* Dummy read to make sure to clear the flag */
363
364    /* ==== Stops the MTU2 timer counter (TCNT_0) ==== */
365    MTU2.TSTR.BIT.CST0 = 0;
366
367    /* ==== Stops the ADC ==== */
368    dummy = ADC.ADCSR.WORD;         /* Dummy read to clear the ADF flag */
369    ADC.ADCSR.WORD = 0x0000;
370  }
371  /* 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	May 14.10	—	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