SH7216 Group

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

REJ06B1002-0100
Rev. 1.00
Jun. 03, 2010

Summary

This application note provides an example to use the SH7216 Multi-function Timer Pulse Unit 2, A/D Converter, and Direct Memory Access Controller.

Target Device

SH7216 MCU

Contents

1. Introduction 2

2. Applications 3

3. Sample Program Listing 18

4. References 29

1. Introduction

1.1 Specifications

The SH7216 Multi-function Timer Pulse Unit 2 activates the A/D Converter, and SH7216 Direct Memory Access Controller transfers the A/D conversion value to the SH7216 on-chip RAM.

1.2 Modules Used

- Multi-function Timer Pulse Unit 2
- A/D Converter
- Direct Memory Access Controller

1.3 Applicable Conditions

MCU	SH7216
Operating Frequencies	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz AD clock: 50 MHz
Integrated Development Environment	Renesas Electronics Corporation 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 -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 Related Application Note

For more information, refer to the following application note:

- SH7216 Group Example of Initialization

2. Applications

2.1 Overview of Modules

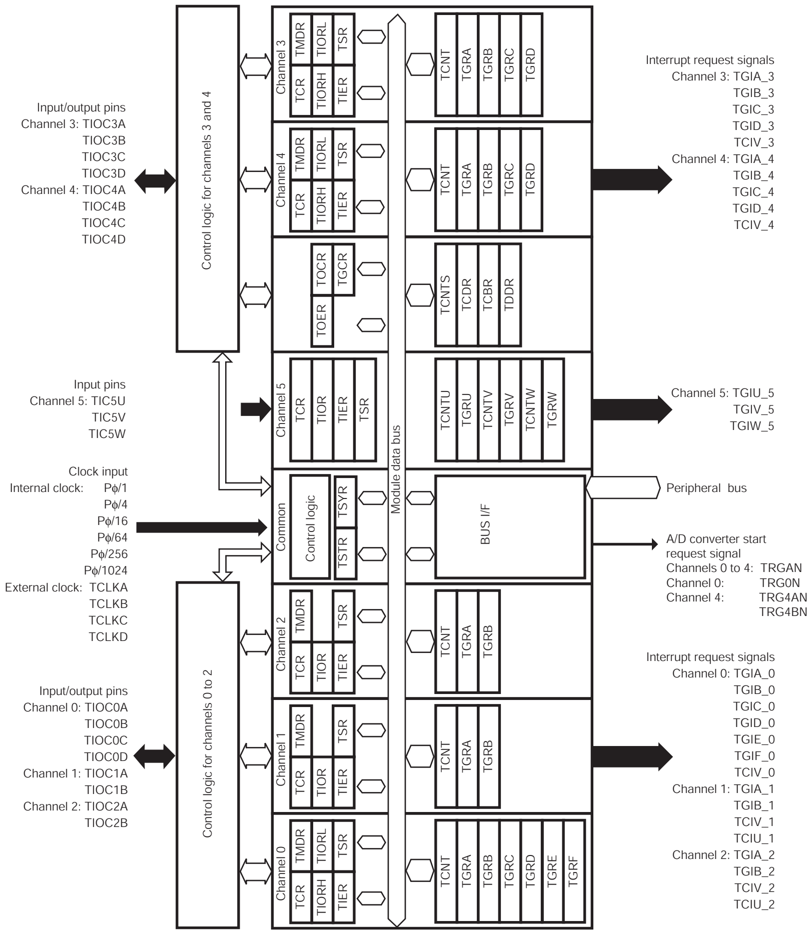
2.1.1 Multi-function Timer Pulse Unit 2

The Multi-function Timer Pulse Unit 2 is an advanced timer unit, consisting of six 16-bit timer channels. The compare match function, and input capture function can be specified on each channel of the Multi-function Timer Pulse Unit 2. Set channels 3 and 4 of the Multi-function Timer Pulse Unit 2 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 Direct Memory Access Controller, Data Transfer Controller, and A/D Converter directly, not via the CPU.

Table 1 lists the specifications of the Multi-function Timer Pulse Unit 2. Figure 1 shows its block diagram. For more information, refer to the Multi-function Timer Pulse Unit 2 (MTU2) chapter in the SH7216 Group Hardware Manual.

Table 1 Multi-function Timer Pulse Unit 2 Specifications

Item	Description
Number of channels	16-bit timer × 6 channels (channels 0 to 5)
Counter clock	Internal clock or external clock can be used on each channel Note: Only the internal clock can be specified on channel 5.
Channels 0 to 5 operation	• Outputs waveforms by compare match, and input capture • Clears counter, writes to multiple timer counters (TCNT) simultaneously, clears counter simultaneously by compare match and input capture • Synchronized inputs to or from registers by counter synchronization, and 12-phase PWM waveform outputs (max.) by using with synchronization
A/D converter start trigger	• A/D converter start trigger can be generated • In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped
Buffer operation	Register buffer operation can be specified on channels 0, 3, and 4
Operating mode	• 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 request	28 interrupt sources (e.g. Compare match interrupt, input capture interrupt)
Other	• Cascade connection • Module standby mode can be specified • Dead time compensation counter available by channel 5



[Legend]

TSTR:	Timer start register	TCDR:	Timer cycle data register
TSYR:	Timer synchronous register	TCBR:	Timer cycle buffer register
TCR:	Timer control register	TDDR:	Timer dead time data register
TMDR:	Timer mode register	TGRA:	Timer general register A
TIOR:	Timer I/O control register	TGRB:	Timer general register B
TIORH:	Timer I/O control register H	TGRC:	Timer general register C
TIORL:	Timer I/O control register L	TGRD:	Timer general register D
TIER:	Timer interrupt enable register	TGRE:	Timer general register E
TGCR:	Timer gate control register	TGRF:	Timer general register F
TOER:	Timer output master enable register	TGRU:	Timer general register U
TOCR:	Timer output control register	TGRV:	Timer general register V
TSR:	Timer status register	TGRW:	Timer general register W
TCNT:	Timer counter		
TCNTS:	Timer subcounter		

Figure 1 Multi-function Timer Pulse Unit 2 Block Diagram

2.1.2 A/D Converter

The A/D Converter includes two A/D modules (A/D_0 and A/D_1) to input four channels with 12-bit resolution. Data converted by the A/D Converter is stored in the A/D data register (ADDR).

A/D Converter operates in single-cycle scan mode, and continuous scan mode. In single-cycle scan mode, the A/D Converter converts the input analog voltage to digital on one or more channels specified, and enters the A/D conversion wait state. In continuous scan mode, the A/D Converter converts the input analog voltage to digital repeatedly on one or more channels specified. When converting analog to digital is completed, the A/D conversion end interrupt can be generated to the CPU. The Direct Memory Access Controller and Data Transfer Controller can be activated when the A/D conversion end interrupt occurs (note).

Table 2 lists the A/D Converter specifications. Figure 2 shows its block diagram. For more information, refer to the A/D Converter (ADC) chapter in the SH7216 Group Hardware Manual.

Note: When the Direct Memory Access Controller is activated, the CPU interrupt is not generated. The Direct Memory Access Controller can only be activated by A/D module_0 (A/D_0).

Table 2 A/D Converter Specifications

Item	Description
Resolution	12-bit
Conversion speed	Minimum conversion timer per channel: 1.0 μs (Aφ is operating at 50 MHz)
Number of modules	2 (A/D_0, A/D_1)
Number of input channels	8 (AN0 to AN7)
Operating mode	• Single-cycle scan mode • Continuous scan mode
Sample-and-hold function	• Channels 0 to 3 share one circuit, channels 4 to 7 share one circuit • Channels 0 to 2 have dedicated circuits for each channel (3 circuits in total)
A/D conversion trigger	• Software: ADST bit setting • Timer: — TRGAN, TRG0N, TRG4AN, and TRG4BN from the Multi-function Timer Pulse Unit 2 — TRGAN, TRG4AN, and TRG4BN from the Multi-function Timer Pulse Unit 2 • External trigger: $\overline{\text{ADTRG}}$

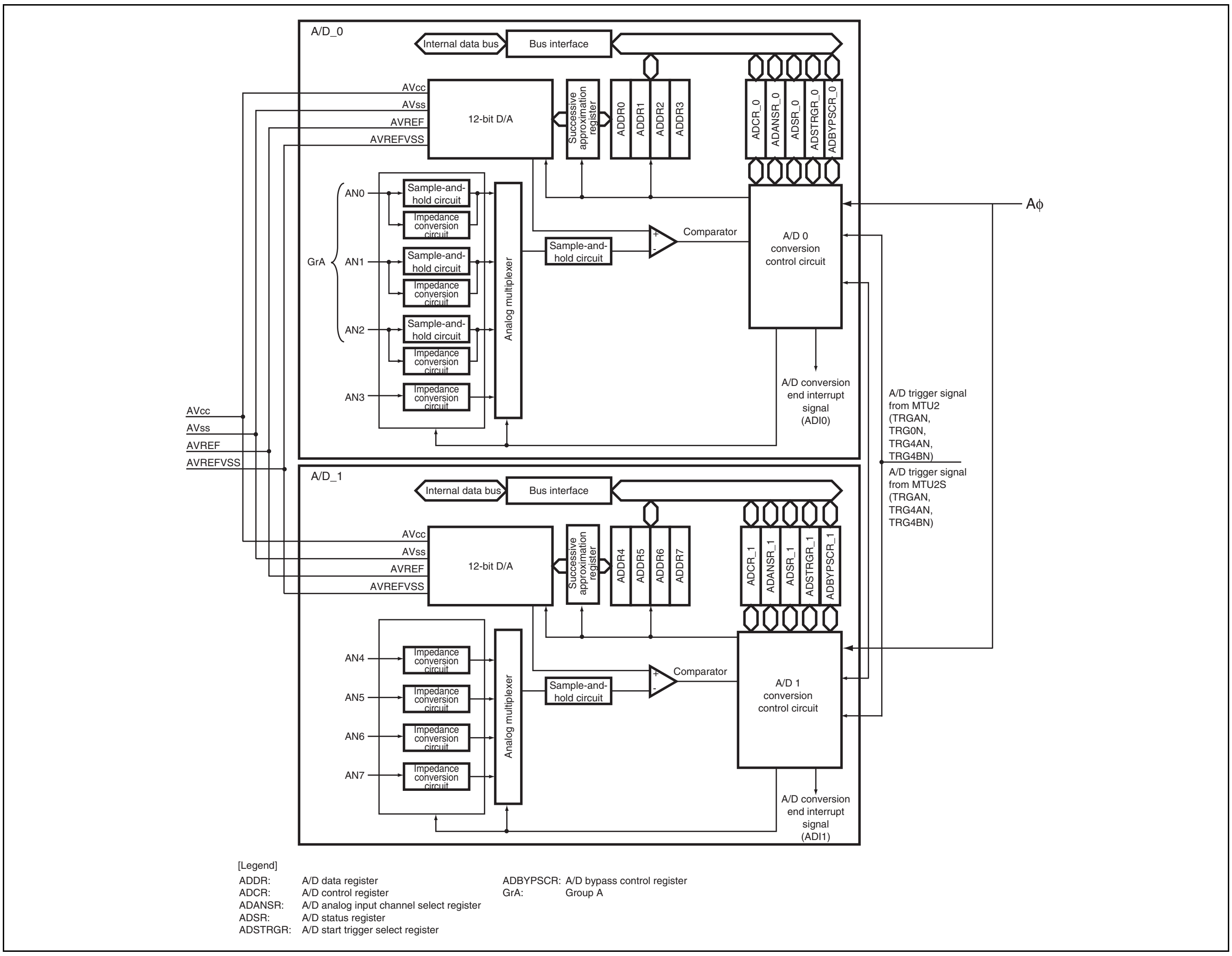

A/D_0
Internal data bus
Bus interface
AVcc
AVss
AVREF
AVREFVSS
12-bit D/A
Successive approximation register
ADDR0
ADDR1
ADDR2
ADDR3
ADCR_0
ADANSR_0
ADSR_0
ADSTRGR_0
ADBYPSCR_0
AN0
AN1
AN2
AN3
GrA
Sample-and-hold circuit
Impedance conversion circuit
Analog multiplexer
Sample-and-hold circuit
Comparator
A/D 0 conversion control circuit
Aϕ
A/D conversion end interrupt signal (ADI0)
A/D trigger signal from MTU2 (TRGAN, TRG0N, TRG4AN, TRG4BN)
A/D trigger signal from MTU2S (TRGAN, TRG4AN, TRG4BN)
AVcc
AVss
AVREF
AVREFVSS
A/D_1
Internal data bus
Bus interface
AVcc
AVss
AVREF
AVREFVSS
12-bit D/A
Successive approximation register
ADDR4
ADDR5
ADDR6
ADDR7
ADCR_1
ADANSR_1
ADSR_1
ADSTRGR_1
ADBYPSCR_1
AN4
AN5
AN6
AN7
Impedance conversion circuit
Analog multiplexer
Sample-and-hold circuit
Comparator
A/D 1 conversion control circuit
A/D conversion end interrupt signal (ADI1)
[Legend]
ADDR: A/D data register
ADCR: A/D control register
ADANSR: A/D analog input channel select register
ADSR: A/D status register
ADSTRGR: A/D start trigger select register
ADBYPSCR: A/D bypass control register
GrA: Group A

Figure 2 A/D Converter Block Diagram

2.1.3 Direct Memory Access Controller

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

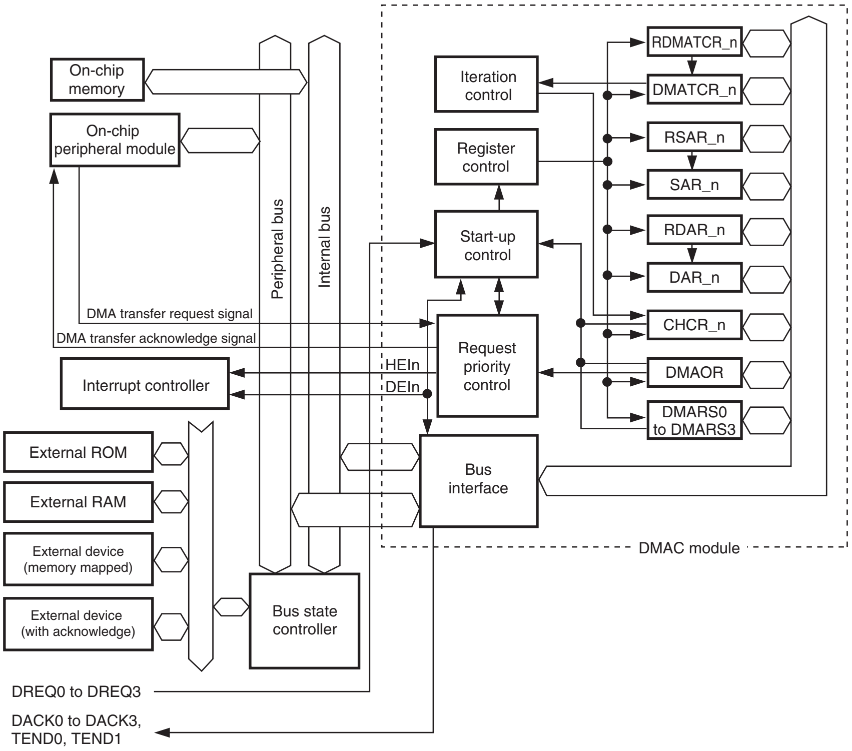
In cycle steal mode, the Direct Memory Access Controller leaves the bus to the other masters when it finishes "a transmit" (in bytes, words, long words, or 16 bytes). When the Direct Memory Access Controller receives another transfer request, it retrieves the bus again. Then, it transfers data in unit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

The Direct Memory Access Controller can generate the DMA transfer end interrupt to the CPU when the DMA transfer is completed.

Table 3 lists the Direct Memory Access Controller specifications. Figure 3 shows its block diagram. For more information, refer to the Direct Memory Access Controller (DMAC) chapter in the SH7216 Group Hardware Manual.

Table 3 Direct Memory Access Controller Specifications

Item	Description
Number of channels	8 (Channels 0 to 7)
Address space	4 GB physically
Transfer data length	Byte, word (2 bytes), long word (4 bytes), and 16 bytes (4 long words)
Number of transfers	16,777,216 (24-bit) times
Address mode	• Single address mode • Dual address mode
Transfer request	• Auto-request • External request (Only 4 channels, channels 0 to 3) • On-chip peripheral module request (number of requests: 19)
Bus mode	• Cycle steal mode • Burst mode
Interrupt source	CPU interrupt is generated when one-half of the data transfer ("a transfer") is completed or "a transfer" is completed
External request detection	DREQ (Transfer request signal from an external device) input can be detected in low level or high level, and at rising edge or falling edge
DMA transfer request acknowledge signal/DMA transfer end	Active levels for DACK and TEND signals can be specified
Reload function	The reload function can be enabled or disabled per channel



[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 DMARS3:	DMA extension resource selectors 0 to 3
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 7	

Figure 3 Direct Memory Access Controller Block Diagram

2.2 Configuration Procedure

2.2.1 Configuring the Multi-function Timer Pulse Unit 2

Figure 4 shows the flow chart for configuring the Multi-function Timer Pulse Unit 2 used in this application. For more information on register settings, refer to the SH7216 Group Hardware Manual.

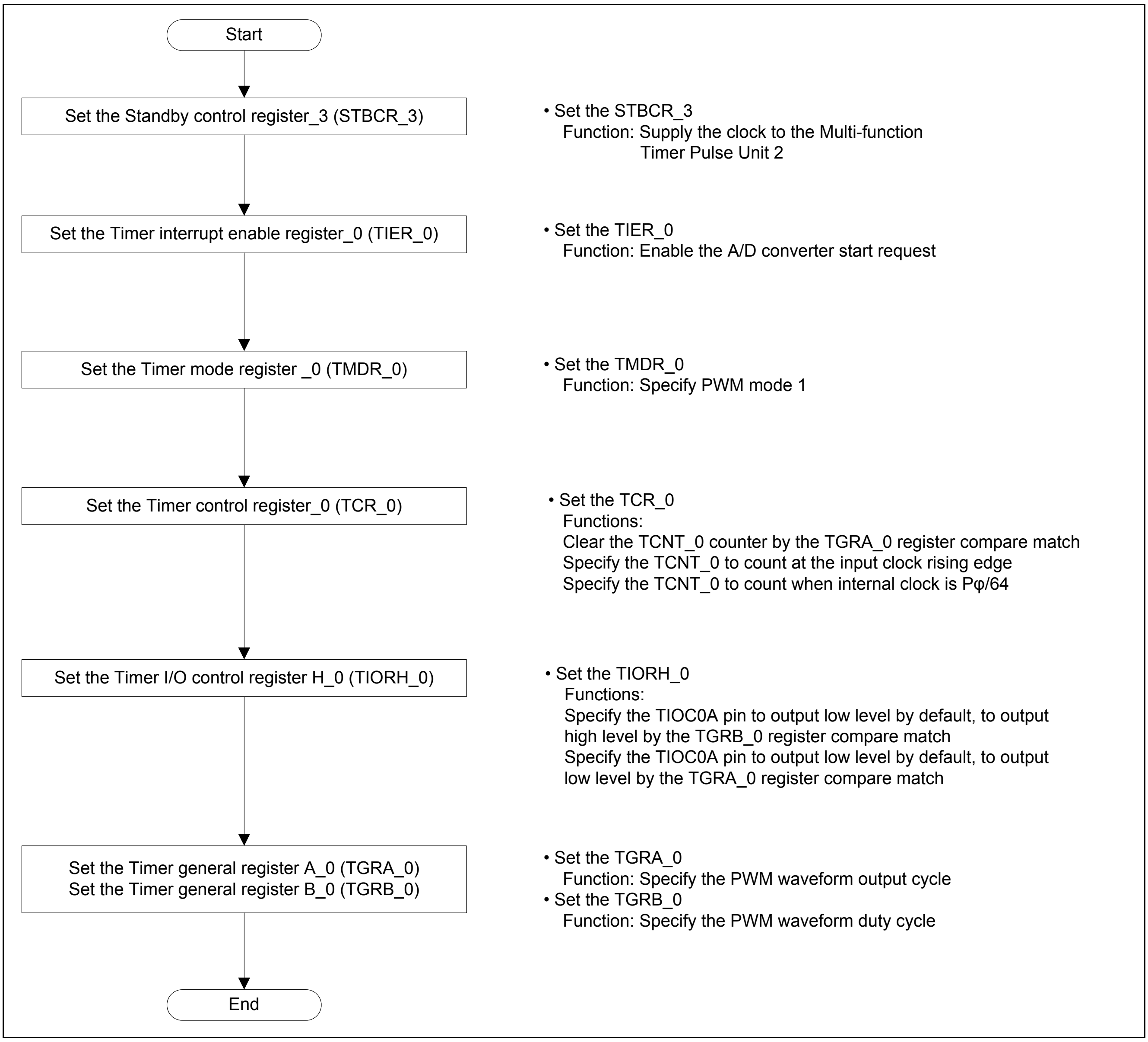
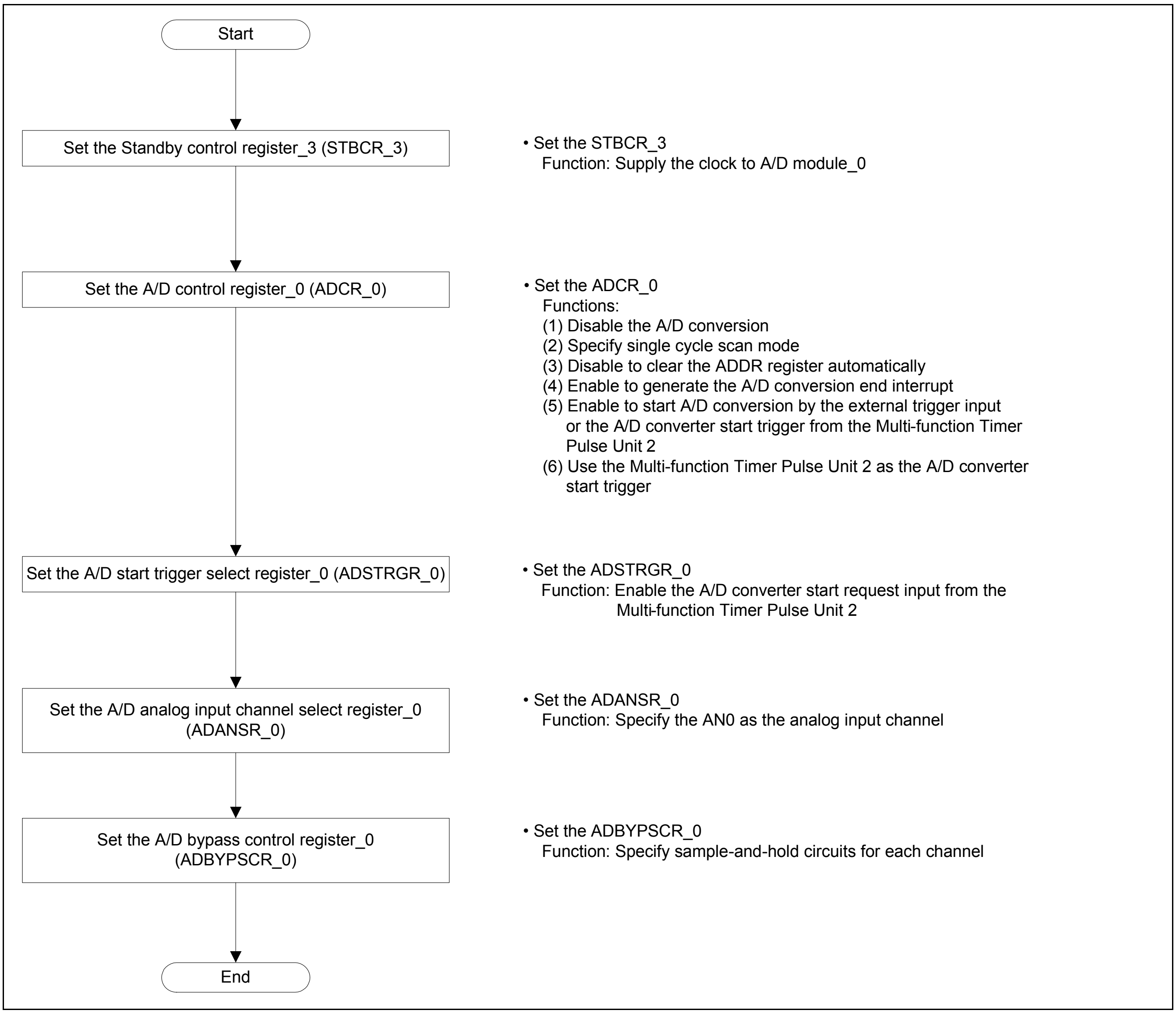

Start
Set the Standby control register_3 (STBCR_3)
• Set the STBCR_3
Function: Supply the clock to the Multi-function Timer Pulse Unit 2
Set the Timer interrupt enable register_0 (TIER_0)
• Set the TIER_0
Function: Enable the A/D converter start request
Set the Timer mode register _0 (TMDR_0)
• Set the TMDR_0
Function: Specify PWM mode 1
Set the Timer control register_0 (TCR_0)
• Set the TCR_0
Functions:
Clear the TCNT_0 counter by the TGRA_0 register compare match
Specify the TCNT_0 to count at the input clock rising edge
Specify the TCNT_0 to count when internal clock is Pφ/64
Set the Timer I/O control register H_0 (TIORH_0)
• Set the TIORH_0
Functions:
Specify the TIOC0A pin to output low level by default, to output high level by the TGRB_0 register compare match
Specify the TIOC0A pin to output low level by default, to output low level by the TGRA_0 register compare match
Set the Timer general register A_0 (TGRA_0)
Set the Timer general register B_0 (TGRB_0)
• Set the TGRA_0
Function: Specify the PWM waveform output cycle
• Set the TGRB_0
Function: Specify the PWM waveform duty cycle
End

Figure 4 Flow Chart for Configuring the Multi-function Timer Pulse Unit 2

2.2.2 Configuring the A/D Converter

Figure 5 shows the flow chart for configuring the A/D Converter used in this application. For more information on register settings, refer to SH7216 Group Hardware Manual.


Start
Set the Standby control register_3 (STBCR_3)
Set the A/D control register_0 (ADCR_0)
Set the A/D start trigger select register_0 (ADSTRGR_0)
Set the A/D analog input channel select register_0 (ADANSR_0)
Set the A/D bypass control register_0 (ADBYPSCR_0)
End

- Set the STBCR_3
  Function: Supply the clock to A/D module_0

- Set the ADCR_0
  Functions:
  (1) Disable the A/D conversion
  (2) Specify single cycle scan mode
  (3) Disable to clear the ADDR register automatically
  (4) Enable to generate the A/D conversion end interrupt
  (5) Enable to start A/D conversion by the external trigger input or the A/D converter start trigger from the Multi-function Timer Pulse Unit 2
  (6) Use the Multi-function Timer Pulse Unit 2 as the A/D converter start trigger

- Set the ADSTRGR_0
  Function: Enable the A/D converter start request input from the Multi-function Timer Pulse Unit 2

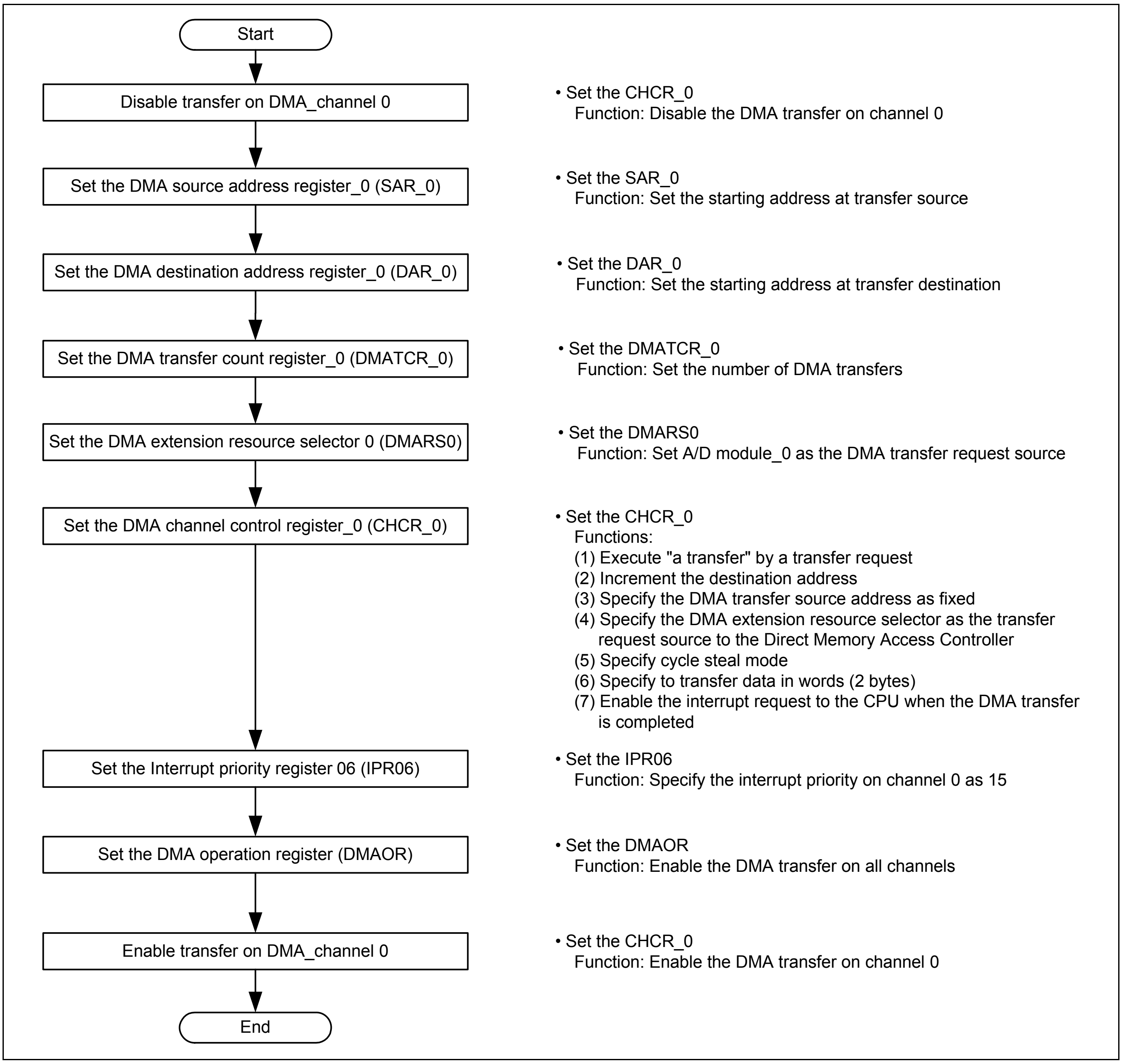
- Set the ADANSR_0
  Function: Specify the AN0 as the analog input channel

- Set the ADBYPSCR_0
  Function: Specify sample-and-hold circuits for each channel

Figure 5 Flow Chart for Configuring the A/D Converter

2.2.3 Configuring the Direct Memory Access Controller

Figure 6 shows the flow chart for configuring the Direct Memory Access Controller used in this application. For more information on register settings, refer to the SH7216 Group Hardware Manual.


Start
Disable transfer on DMA_channel 0
Set the DMA source address register_0 (SAR_0)
Set the DMA destination address register_0 (DAR_0)
Set the DMA transfer count register_0 (DMATCR_0)
Set the DMA extension resource selector 0 (DMARS0)
Set the DMA channel control register_0 (CHCR_0)
Set the Interrupt priority register 06 (IPR06)
Set the DMA operation register (DMAOR)
Enable transfer on DMA_channel 0
End

- Set the CHCR_0
  Function: Disable the DMA transfer on channel 0
- Set the SAR_0
  Function: Set the starting address at transfer source
- Set the DAR_0
  Function: Set the starting address at transfer destination
- Set the DMATCR_0
  Function: Set the number of DMA transfers
- Set the DMARS0
  Function: Set A/D module_0 as the DMA transfer request source
- Set the CHCR_0
  Functions:
  (1) Execute "a transfer" by a transfer request
  (2) Increment the destination address
  (3) Specify the DMA transfer source address as fixed
  (4) Specify the DMA extension resource selector as the transfer request source to the Direct Memory Access Controller
  (5) Specify cycle steal mode
  (6) Specify to transfer data in words (2 bytes)
  (7) Enable the interrupt request to the CPU when the DMA transfer is completed
- Set the IPR06
  Function: Specify the interrupt priority on channel 0 as 15
- Set the DMAOR
  Function: Enable the DMA transfer on all channels
- Set the CHCR_0
  Function: Enable the DMA transfer on channel 0

Figure 6 Flow Chart for Configuring the Direct Memory Access Controller

2.3 Sample Program Procedure

2.3.1 Sample Program Operation

The sample program uses the Multi-function Timer Pulse Unit 2 to output PWM waveform (one cycle is 1 ms), and activates the A/D Converter in every cycle. Then, it transfers the A/D conversion value to on-chip RAM by the Direct Memory Access Controller every time the A/D conversion is completed.

The Multi-function Timer Pulse Unit 2 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 Timer general register A_0 (TGRA_0), and the duty cycle in Timer general register B_0. The A/D converter start request signal (TRGAN) occurs on the compare match between Timer counter_0 (TCNT_0) and TGRA_0 (note).

The A/D Converter operates in single scan mode to start the A/D conversion on the analog input channel AN0 by the TRGAN. The A/D conversion end interrupt signal (ADI0) occurs when the A/D conversion is completed.

The Direct Memory Access Controller operates in cycle steal mode to transfer the A/D conversion data from A/D data register 0 (ADDR0) to on-chip RAM by ADI0. The DMA transfer end interrupt (DEI) occurs when transferring 1-KB data is completed.

After the CPU is reset, the sample program configures the A/D Converter, Direct Memory Access Controller, and Multi-function Timer Pulse Unit 2, and starts TCNT_0 counter. Then, it executes the A/D conversion on every TGRA_0 compare match, and transfers the A/D conversion value to on-chip RAM by the Direct Memory Access Controller. In the DMA transfer end interrupt processing, the sample program disables the DMA transfer, clears the transfer end flag (TE bit), and stops TCNT_0 counter.

Figure 7 shows the sample program operation (overview).

Note: The A/D converter start request can be generated no matter whether the Multi-function Timer Pulse Unit 2 outputs PWM waveforms or not.

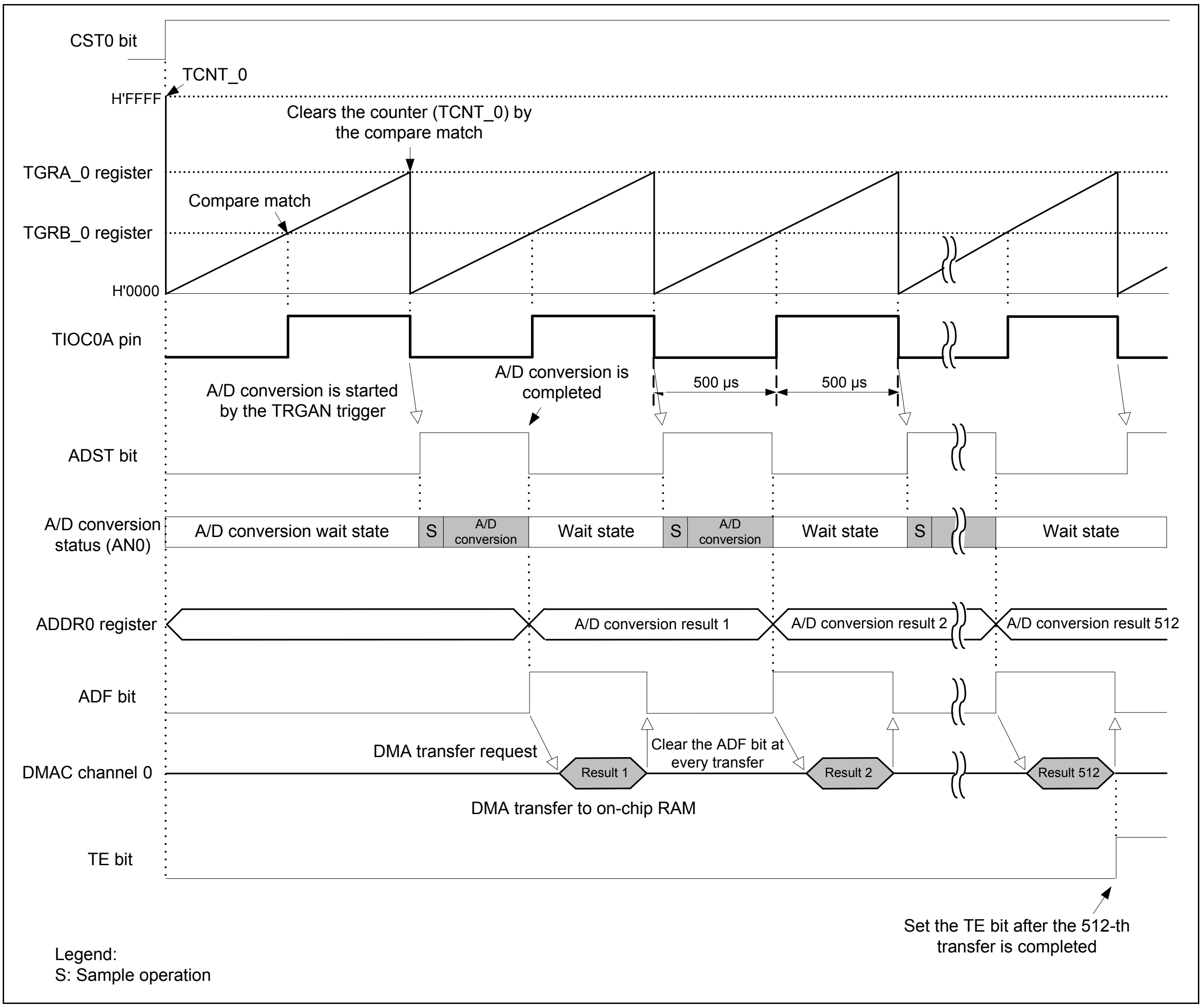

CST0 bit
TCNT_0
H'FFFF
Clears the counter (TCNT_0) by the compare match
TGRA_0 register
Compare match
TGRB_0 register
H'0000
TIOC0A pin
A/D conversion is started by the TRGAN trigger
A/D conversion is completed
500 µs
500 µs
ADST bit
A/D conversion status (AN0)
A/D conversion wait state
S
A/D conversion
Wait state
ADDR0 register
A/D conversion result 1
A/D conversion result 2
A/D conversion result 512
ADF bit
DMA transfer request
Clear the ADF bit at every transfer
DMAC channel 0
Result 1
Result 2
Result 512
DMA transfer to on-chip RAM
TE bit
Set the TE bit after the 512-th transfer is completed
Legend:
S: Sample operation

Figure 7 Sample Program Operation (Overview)

2.3.2 Multi-function Timer Pulse Unit 2 Register Setting

Table 4 lists the register settings for the Multi-function Timer Pulse Unit 2.

Table 4 Multi-function Timer Pulse Unit 2 Register Settings

Register Name	Address	Setting	Description
Standby control register 3 (STBCR3)	H'FFFE 0408	H'5A	• MSTP35 = "0": Supply the clock to the Multi-function Timer Pulse Unit 2
Port E control register L1 (PECRL1)	H'FFFE 3A16	H'0004	• PE0MD = "4": Specify PE0 pin as the TIOC0A
Port E I/O register L (PEIORL)	H'FFFE 3A06	H'0001	• PE0IOR = "1": Specify the TIOC0A to output
Timer control register_0 (TCR_0)	H'FFFE 4300	H'23	• CCLR[2:0] = "B'001": Clear TCNT_0 by TGRA_0 compare match or input capture • CKEG[1:0] = "B'00": Specify TCNT_0 to count at rising edge of the input clock • TPSC[2:0] = "B'011": Specify 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 to output low level signal by default, to output high level signal by the compare match • IOA[3:0] = "B'0001": Specify to output low level signal by default, to output low level signal by the compare match
Timer interrupt enable register_0 (TIER_0)	H'FFFE 4304	H'80	• TTGE = "1": Enable to generate the A/D converter start request
Timer general register A_0 (TGRA_0)	H'FFFE 4308	D'781	Specify the PWM waveform output cycle
Timer general register B_0 (TGRB_0)	H'FFFE 430A	D'390	Specify the PWM waveform duty cycle
Timer mode register_0 (TMDR_0)	H'FFFE 4301	H'02	• MD[3:0] = "B'0010": Specify PWM mode 1
Timer start register (TSTR)	H'FFFE 4280	H'01	• CST0 = "1": TCNT_2 to TCNT_0 counters are running

2.3.3 A/D Converter Register Setting

Table 5 lists the register settings for the A/D Converter.

The following setting enables the A/D conversion end interrupt, however, the CPU interrupt is not generated because the Direct Memory Access Controller is activated in this application.

Table 5 A/D Converter Register Settings

Register Name	Address	Setting	Description
Standby control register 3 (STBCR3)	H'FFFE 0408	H'5A	• MSTP32 = "0": Supply the clock to A/D_0
A/D control register_0 (ADCR_0)	H'FFFF E800	H'12	• ADST = "0": Disable the A/D conversion • ADCS = "0": Specify single-cycle scan mode • ACE = "0": Disable to clear the ADDR register automatically by reading the ADDR register • ADIE = "1": Enable to generate the A/D conversion end interrupt • TRGE = "1": Enable the A/D conversion by the external trigger input or the A/D converter start trigger from the Multi-function Timer Pulse Unit 2 • EXTRG = "0": Activate the A/D Converter by the A/D converter start trigger from the Multi-function Timer Pulse Unit 2
A/D start trigger select register_0 (ADSTRGR_0)	H'FFFF E81C	H'04	• STR2 = "1": Enable to start the A/D conversion by the TRGAN trigger (Multi-function Timer Pulse Unit 2)
A/D analog input channel select register_0 (ADANSR_0)	H'FFFF E820	H'01	• ANS0 = "1": Specify the analog input channel AN0
A/D bypass control register_0 (ADBYPSCR_0)	H'FFFF E830	H'01	• SH = "1": Specify the exclusive sample-and-hold circuit for each channel

2.3.4 Direct Memory Access Controller Register Setting

Table 6 lists the register settings for the Direct Memory Access Controller.

Table 6 Direct Memory Access Controller Register Settings

Register Name	Address	Setting	Description
DMA source address register_0 (SAR_0)	H'FFFE 1000	H'FFFF E840	Transfer source address: ADDR0 register address
DMA destination address register_0 (DAR_0)	H'FFFE 1004	H'FFF8 0000	Transfer destination address: Starting address in on-chip RAM (Transfer destination)
DMA transfer count register_0 (DMATCR_0)	H'FFFE 1008	D'512	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 • RLD = "0": Disable the reload function • DM[1:0] = "B'01": Increment the destination address by two • 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 580D	• DE = "1": Enable the DMA transfer on channel 0
DMA operation register (DMAOR)	H'FFFE 1200	H'0007 (note)	• 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

Note: To avoid clearing the address error flag and NMI flag, the above setting writes 1 in bits AE and NMIF.

2.3.5 Sample Program Flow Chart

Figure 8 shows the flow chart of the sample program.

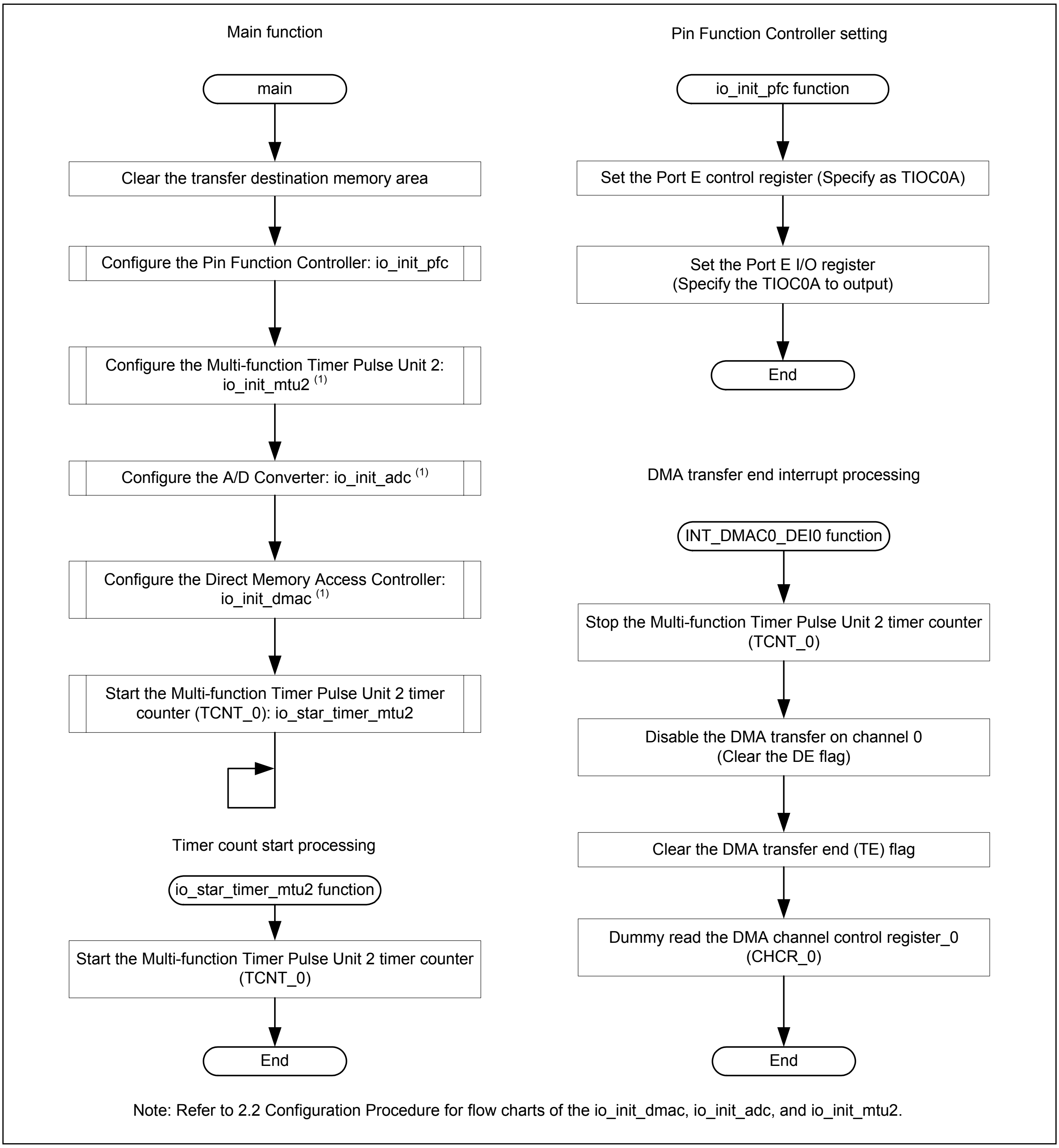

Main function
main
Clear the transfer destination memory area
Configure the Pin Function Controller: io_init_pfc
Configure the Multi-function Timer Pulse Unit 2: io_init_mtu2 (1)
Configure the A/D Converter: io_init_adc (1)
Configure the Direct Memory Access Controller: io_init_dmac (1)
Start the Multi-function Timer Pulse Unit 2 timer counter (TCNT_0): io_star_timer_mtu2
Timer count start processing
io_star_timer_mtu2 function
Start the Multi-function Timer Pulse Unit 2 timer counter (TCNT_0)
End
Pin Function Controller setting
io_init_pfc function
Set the Port E control register (Specify as TIOC0A)
Set the Port E I/O register (Specify the TIOC0A to output)
End
DMA transfer end interrupt processing
INT_DMAC0_DEI0 function
Stop the Multi-function Timer Pulse Unit 2 timer counter (TCNT_0)
Disable the DMA transfer on channel 0 (Clear the DE flag)
Clear the DMA transfer end (TE) flag
Dummy read the DMA channel control register_0 (CHCR_0)
End
Note: Refer to 2.2 Configuration Procedure for flow charts of the io_init_dmac, io_init_adc, and io_init_mtu2.

Figure 8 Sample Program Flow Chart

3. Sample Program Listing

3.1 Sample Program List "main.c" (1/9)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
********************************************************************************
*   (C) 2010 Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data **************************
*   System Name : SH7216 Sample Program
*   File Name   : main.c
*   Abstract    : MTU2+ADC+DMAC Module Application
*   Version     : 1.00.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description :
********************************************************************************
*   History     : Jun.03,2010 Ver.1.00.00
*""FILE COMMENT END""**********************************************************/
#include "iodefine.h"

/* ==== Macro definition ==== */
#define DMA_SRC_ADR   0xffffe840     /* DMA transfer source (ADDR0) address */
#define DMA_COUNT  512           /* Number of transfers: 512 (1 word/1 transfer) */

```

3.2 Sample Program List "main.c" (2/9)

```
/* ==== Prototype declaration ==== */
void main(void);
void io_init_pfc(void);
void io_init_mtu2(void);
void io_init_adc(void);
void io_init_dmac(void *src, void *dst, int count);
void io_start_timer_mtu2(void);

/* ==== Global variable ==== */
unsigned short ad_data[DMA_COUNT];    /* Transfer destination of the A/D conversion data */
volatile unsigned char f_dma_end; /* Variable to check if the DMA transfer is completed */

/*""FUNC COMMENT""**************************************************************
 * ID              :
 * Outline         : Sample program main
 *-------------------------------------------------------------------------------
 * Include         :
 *-------------------------------------------------------------------------------
 * Declaration     : void main(void);
 *-------------------------------------------------------------------------------
 * Description     : Clears the DMA transfer destination, configures the PFC,
 *                 : MTU2, ADC and DMAC.
 *                 : Then, this function starts the MTU2 timer to count.
 *                 : After counting is started, it outputs PWM waveform in 1-ms
 *                 : cycle, and activates the ADC in every cycle.
 *                 : It transfers the A/D conversion value to on-chip RAM every
 *                 : time the A/D conversion is completed.
 *                 : After the timer starts counting, this function waits until
 *                 : transferring 1-KB data is completed.
 *-------------------------------------------------------------------------------
 * Argument        : void
 *-------------------------------------------------------------------------------
 * Return Value    : void
 *-------------------------------------------------------------------------------
 * Note            : None
 *""FUNC COMMENT END""**********************************************************/
void main(void)
{
  int i;                        /* Loop counter variable */

  /* ==== Clears the transfer destination memory area ==== */
  for(i = 0; i < DMA_COUNT; i++){
    ad_data[i] = 0x0000;        /* Clears the area storing A/D conversion data */
                                /* in on-chip RAM to 0 */
  }

  /* ==== Configures the PFC ==== */
  io_init_pfc();

```

3.3 Sample Program List "main.c" (3/9)

```
  /* ==== Configures the MTU2 ==== */
  io_init_mtu2();

  /* ==== Configures the ADC ==== */
  io_init_adc();

  /* ==== Configures the DMAC ==== */
  io_init_dmac((void *)DMA_SRC_ADR, (void *)ad_data, DMA_COUNT);

  /* ==== Specifies the MTU2 timer (TCNT_0) to start counting ==== */
  io_start_timer_mtu2();

  while(f_dma_end == 0){
    /* Waits until the DMA transfer is completed */
  }

  while(1){
    /* loop */
  }
}

/*""FUNC COMMENT""**************************************************************
 * ID            :
 * Outline       : PFC configuration
 *------------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration   : void io_init_pfc(void);
 *------------------------------------------------------------------------------
 * Description   : Configures the Pin Function Controller (PFC).
 *               : Specifies the PE0 pin function to TIOC0A output.
 *------------------------------------------------------------------------------
 * Argument      : void
 *------------------------------------------------------------------------------
 * Return Value  : void
 *------------------------------------------------------------------------------
 * Note          : None
 *""FUNC COMMENT END""**********************************************************/
void io_init_pfc(void)
{
  /* ==== Sets the Port E control register L1 (PECRL1) ==== */
  PFC.PECRL1.BIT.PE0MD = 4;   /* Sets the PE0 pin function to TIOC0A */

  /* ==== Sets the Port E IO register L (PEIORL) ==== */
  PFC.PEIORL.BIT.B0 = 1;      /* Sets the TIOC0A pin to output */
}

```

3.4 Sample Program List "main.c" (4/9)

```
/*""FUNC COMMENT""***************************************************************
 * ID               :
 * Outline          : MTU2 configuration
 *-------------------------------------------------------------------------------
 * Include          : "iodefine.h"
 *-------------------------------------------------------------------------------
 * Declaration      : void io_init_mtu2(void);
 *-------------------------------------------------------------------------------
 * Description      : Configures the Multi-function Timer Pulse Unit 2 (MTU2).
 *                  :  - Operating mode: PWM mode 1
 *                  :  - TCNT_0: Count when the internal clock is P-clock/64
 *                  :  - PWM waveform cycle: Set to TGRA_0
 *                  :  - PWM waveform duty cycle: Set to TGRB_0
 *-------------------------------------------------------------------------------
 * Argument         : void
 *-------------------------------------------------------------------------------
 * Return Value     : void
 *-------------------------------------------------------------------------------
 * Note             : None
 *""FUNC COMMENT END""***********************************************************/
void io_init_mtu2(void)
{
  /* ==== Sets the Standby control register (STBCR3) ==== */
  STB.CR3.BIT._MTU2 = 0;      /* Supplies the clock to the MTU2 */

  /* ==== Sets the Timer interrupt enable register_0 (TIER_0) ==== */
  MTU20.TIER.BIT.TTGE = 1;    /* Enables the A/D converter start request */

  /* ==== Sets the Timer control register_0 (TCR_0) ==== */
  MTU20.TCR.BYTE = 0x23;
    /*
        bit 7 to 5: CCLR[2:0] = B'001 --- Clears the Timer counter (TCNT) by the
                  :                       Timer general register (TGRA) compare match
        bit 4, 3: CKEG[1:0] = B'00 ------ Specifies the TCNT to count at the
                :                         rising edge of the input clock
        bit 2 to 0: TPSC[2:0] = B'011 --- Specifies the TCNT to count when
                  :                       the internal clock is P-clock/64
    */

```

3.5 Sample Program List "main.c" (5/9)

```
  /* ==== Sets the Timer I/O control register H_0 (TIORH_0) ==== */
  MTU20.TIOR.BIT.IOB = 2;     /* TIOC0A pin outputs low level signal by default, */
                              /* outputs high level signal by the TGRB_0 register */
                              /* compare match */
  MTU20.TIOR.BIT.IOA = 1;     /* TIOC0A pin outputs low level signal by default, */
                              /* outputs low level signal by the TGRA_0 register */
                              /* compare match */

  /* ==== Sets the Timer general register A_0 (TGRA_0) ==== */
  MTU20.TGRA = 782 - 1;       /* Specifies the PWM waveform 1-ms cycle by P-clock/64 */

  /* ==== Sets the Timer general register B_0 (TGRB_0) ==== */
  MTU20.TGRB = 391 - 1;       /* Specifies the PWM duty cycle 50% by P-clock/64 */

  /* ==== Sets the Timer mode register_0 (TMDR_0) ==== */
  MTU20.TMDR.BIT.MD = 2;      /* Specifies PWM mode 1 */
}

/*""FUNC COMMENT""**************************************************************
 * ID              :
 * Outline         : ADC configuration
 *-------------------------------------------------------------------------------
 * Include         : "iodefine.h"
 *-------------------------------------------------------------------------------
 * Declaration     : void io_init_adc(void);
 *-------------------------------------------------------------------------------
 * Description     : Configures the A/D Converter (ADC).
 *                 :  - Operating mode: Single-cycle scan mode
 *                 :  - A/D converter start trigger: TRGAN (MTU2)
 *                 :  - Conversion circuit: Sample-and-hold circuit
 *-------------------------------------------------------------------------------
 * Argument        : void
 *-------------------------------------------------------------------------------
 * Return Value    : void
 *-------------------------------------------------------------------------------
 * Note            : None
 *""FUNC COMMENT END""**********************************************************/
 void io_init_adc(void)
{
  /* ==== Sets the Standby control register 3 (STBCR3) ==== */
  STB.CR3.BIT._ADC0 = 0;      /* Supplies the clock to A/D module_0 */

```

3.6 Sample Program List "main.c" (6/9)

```
  /* ==== Sets the A/D control register_0 (ADCR_0) ==== */
  ADC0.ADCR.BYTE = 0x12;
    /*
        bit 7: ADST = 0 ---- Not used
        bit 6: ADCS = 0 ---- Single-cycle scan mode
        bit 5: ACE = 0 ----- Disables to auto-clear the ADDR by reading the ADDR
        bit 4: ADIE = 1 ---- Enables the A/D conversion end interrupt
        bit 3, 2: Reserved(0)
        bit 1: TRGE = 1 ---- Enables the A/D conversion by the external trigger or
             :               the A/D converter start trigger from the MTU2/MTU2S
        bit 0: EXTRG = 0 --- Activates the A/D Converter by the A/D converter
             :               start trigger from the MTU2/MTU2S
    */

  /* ==== Sets the A/D start trigger select register_0 (ADSTRGR_0) ==== */
  ADC0.ADSTRGR.BIT.STR2 = 1;  /* Enables to start the A/D conversion by the */
                              /* TRGAN trigger (MTU2) */

  /* ==== Sets the A/D analog input channel select register_0 (ADANSR_0) ==== */
  ADC0.ADANSR.BIT.ANS0 = 1;   /* Specifies the A/D analog input channel AN0 */

  /* ==== Sets the A/D bypass control register_0 (ADBYPSCR_0) ==== */
  ADC0.ADBYPSCR.BIT.SH = 1;   /* Specifies the sample-and-hold circuit */
}

```

3.7 Sample Program List "main.c" (7/9)

```
/*""FUNC COMMENT""**************************************************************
 * ID              :
 * Outline         : DMAC configuration
 *------------------------------------------------------------------------------
 * Include         : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration     : void io_init_dmac(void *src, void *dst, int count);
 *------------------------------------------------------------------------------
 * Description     : Configures the Direct Memory Access Controller (DMAC).
 *                 :  - Operating mode: Cycle steal mode
 *                 :  - On-chip peripheral module request: A/D module_0
 *                 :  - Transfer source: A/D data register_0 (ADDR0)
 *                 :  - Transfer destination: On-chip RAM
 *                 :  - Transfer data length: In words
 *                 :  - Reload function: Not used
 *------------------------------------------------------------------------------
 * Argument        : void *src ; Transfer source address
 *                 : void *dst ; Transfer destination address
 *                 : int count ; Number of transfers
 *------------------------------------------------------------------------------
 * Return Value    : void
 *------------------------------------------------------------------------------
 * Note            : None
 *""FUNC COMMENT END""**********************************************************/
void io_init_dmac(void *src, void *dst, int count)
{
  /* ==== Disables the DMA transfer on channel 0 ==== */
  DMAC0.CHCR.BIT.DE = 0;

  /* ==== Sets the DMA source address register_0 (SAR_0) ==== */
  DMAC0.SAR = src;            /* Sets the DMA transfer source address */

  /* ==== Sets the DMA destination address register_0 (DAR_0) ==== */
  DMAC0.DAR = dst;            /* Sets the DMA transfer destination address */

  /* ==== Sets the DMA transfer count register_0 (DMATCR_0) ==== */
  DMAC0.DMATCR = count;       /* Sets the number of DMA transfers */

  /* ==== Sets the DMA extension resource selector 0 (DMARS0) ==== */
  DMAC.DMARS0.WORD =0x00b3;   /* Sets A/D module_0 as the DMA transfer request source */

```

3.8 Sample Program List "main.c" (8/9)

```
  /* ==== Sets the DMA channel control register_0 (CHCR_0) ==== */
  DMAC0.CHCR.LONG = 0x0000480c;
    /*
        bit 31: TC = 0 ------------------ Executes "a transfer" by a transfer request
        bit 30, 29: Reserved(0)
        bit 28: RLD = 0 ----------------- Disables the reload function
        bit 27 to 24: Reserved(0).
        bit 23: DO = 0 ------------------ Not used
        bit 22: TL = 0 ------------------ Not used
        bit 21, 20: Reserved(0)
        bit 19: HE = 0 ------------------ Not used
        bit 18: HIE = 0 ----------------- Not used
        bit 17: AM = 0 ------------------ Not used
        bit 16: AL = 0 ------------------ Not used
        bit 15, 14: DM[1:0] = B'01 ------ Increments the destination address by 2
        bit 13, 12: SM[1:0] = B'00 ------ Specifies the source address as fixed
        bit 11 to 8: RS[3:0] = B'1000 --- Specifies the DMA extension
                                          resource selector
        bit 7: DL = 0 ------------------- Not used
        bit 6: DS = 0 ------------------- Not used
        bit 5: TB = 0 ------------------- Specifies cycle steal mode
        bit 4, 3: TS[1:0] = B'01 -------- Specifies to transfer data in words
        bit 2: IE = 1 ------------------- Enables the interrupt request
        bit 1: TE = 0 ------------------- Clears the TE flag
        bit 0: DE = 0 ------------------- Disables the DMA transfer
    */

  /* ==== Sets the Interrupt priority register 06 (IPR06) ==== */
  INTC.IPR06.BIT._DMAC0 = 15;     /* Specifies the DMAC0 interrupt level as 15 */

  /* ==== Sets the DMA operation register (DMAOR) ==== */
  DMAC.DMAOR.WORD |= 0x0007;
               /* bit 0: DME = 1 --- Enables the DMA transfer on all channels */
               /* To avoid clearing the address error flag, and  NMI flag, */
               /* writes 1 in bits AE and NMIF */

  /* ==== Enables the DMA transfer on channel 0 ==== */
  DMAC0.CHCR.BIT.DE = 1;
}

```

3.9 Sample Program List "main.c" (9/9)

```
/*""FUNC COMMENT""**************************************************************
 * ID            :
 * Outline       : MTU2 Timer Count (TCNT_0) Start Setting
 *------------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration   : void io_start_timer_mtu2(void);
 *------------------------------------------------------------------------------
 * Description   : Starts the MTU2 timer counter_0 (TCNT_0) operation.
 *------------------------------------------------------------------------------
 * Argument      : void
 *------------------------------------------------------------------------------
 * Return Value  : void
 *------------------------------------------------------------------------------
 * Note          : None
 *""FUNC COMMENT END""**********************************************************/
void io_start_timer_mtu2(void)
{
  MTU2.TSTR.BIT.CST0 = 1;
}

/* End of File */
```

3.10 Sample Program List "intprg.c" (1/2)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
********************************************************************************
*   (C) 2010 Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data **************************
*   System Name : SH7216 Sample Program
*   File Name   : intprg.c
*   Abstract    : Interrupt Functions
*   Version     : 1.00.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description :
********************************************************************************
*   History     : Jun.03,2010 Ver.1.00.00
*""FILE COMMENT END""**********************************************************/
#include <machine.h>
#include "vect.h"
#include "iodefine.h"

extern unsigned char f_dma_end;

```

3.11 Sample Program List "intprg.c" (2/2)

```
#pragma section IntPRG

// 4 Illegal code
void INT_Illegal_code(void){/* sleep(); */}
// 5 Reserved

// 6 Illegal slot
void INT_Illegal_slot(void){/* sleep(); */}

…

// 108 DMAC0 DEI0
void INT_DMAC0_DEI0(void)
{
  unsigned long dummy;        /* Variable for dummy read */

  f_dma_end = 1;

  /* ==== Stops the MTU2 timer counter (TCNT_0) ==== */
  MTU2.TSTR.BIT.CST0 = 0;

  /* ==== Disables the DMA transfer on channel 0 ==== */
  DMAC0.CHCR.BIT.DE = 0;

  /* ==== Clears the DMA transfer end flag ==== */
  DMAC0.CHCR.BIT.TE = 0;

  dummy = DMAC0.CHCR.LONG;    /* Dummy read */
}

…

// 254 SCIF SCIF3 RXI3
void INT_SCIF_SCIF3_RXI3(void){/* sleep(); */}
// 255 SCIF SCIF3 TXI3
void INT_SCIF_SCIF3_TXI3(void){/* sleep(); */}
// Dummy
void Dummy(void){/* sleep(); */}

/* 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 Electronics website.

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

- Technical Update
  Addition to and correction of errors in the SH7280 Group Hardware Manual and SH7216 Group Hardware Manual (TN-SH7-A727A/E)
  Correction of errors in the SH7216 Group Hardware Manual (TN-SH7-A754A/E)
  Correction of errors in the SH7216 Group Hardware Manual (TN-SH7-A761A/E)
  Amendment to Product Code Lineup in the SH7216 Group Hardware Manual (TN-SH7-A762A/E)
  Limitation on Changes to the Frequency Control Register and Correction of Error in the Hardware Manual (TN-SH7-A769A/E)
  Correction of Errors in the Hardware Manual (TN-SH7-A771A/E)
  The latest version can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Jun.03.10	—	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