To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

SH7147

Resolver Built-in DC Brushless Motor Control

1. Abstract

This application note is organized based on sample of resolver built-in DC brushless motor control, which uses SH7147 embedded ADC_0, MTU2S_3,4, and CMT_0,1, and is aimed to provide information users may need during software and hardware design.

Although the operation of each program in this application note has been checked, make sure that you conduct your own operation checks before actually using.

2. Introduction

2.1 Specification

(1) System Architecture Specification

Figure1 describes system architecture of this control.

- SH7147 is used as motor control MCU.
- M59314 is used as step-up predriver for inverter driver
- Six H7N0602LS (power MOS FET) is used as 3-phase inverter

(2) Motor Control Specification

Resolver built-in DC brushless motor is controlled in 120-degree energization drive system.

- Resolver output signal is converted to digital at ADC_0 and motor position is detected at every 30 degree.
- Motor drive wave most suitable to motor position will be outputted from MTU2S (level output on positive phase side, PWM output on antiphase side)
- Motor revolving direction reverses at every 4[sec].

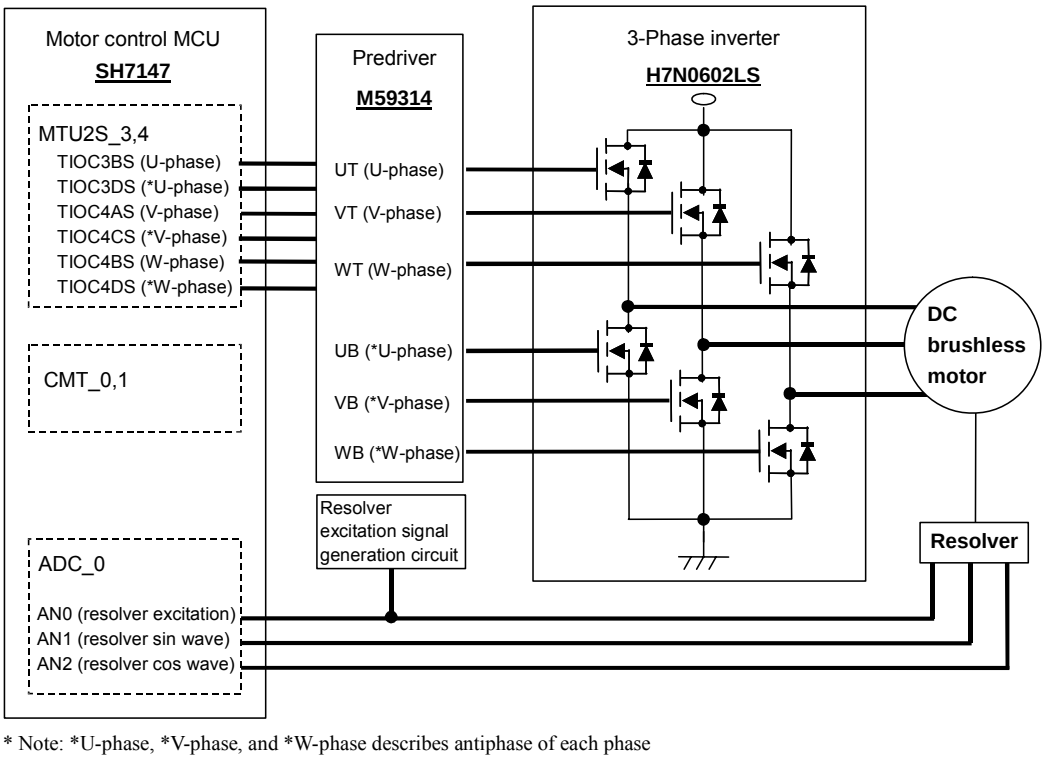

Figure1 Overview of Resolver Built-in DC Brushless Motor Control System

2.2 Functions Used

- Multi-function Timer pulse Unit 2S channel 3,4 (MTU2S_3,4)
- A/D converter 0 (ADC_0)
- Compare Match Timer channel 0,1 (CMT_0,1)

2.3 Applicable Conditions

- MCU : SH7147(R5F71474AK64FPV)
- Operation Frequency : Internal Clock 64MHz
  : Bus Clock 32MHz
  : Peripheral Clock 32MHz
  : MTU2S Clock 64MHz
  : MTU2 Clock 32MHz
- C Compiler : Renesas Technology product
  SuperH RISC engine Family C/C++ compiler package
  Ver.9.00
- Compile Option : Default setting by HEW

2.4 Related Application Note

- Power MOS FET Application Note

3. Description of Control Sample

This control sample determines motor position by A/D converting output signal of resolver, which is built in DC brushless motor. A motor drive wave, most suitable to the determined mortar position, will be output from MTU2S and controls DC brushless motor.

3.1 Overview of Entire Operation

Following describes functions used by SH7147 for controlling resolver built-in DC brushless motor (Figure 2.)

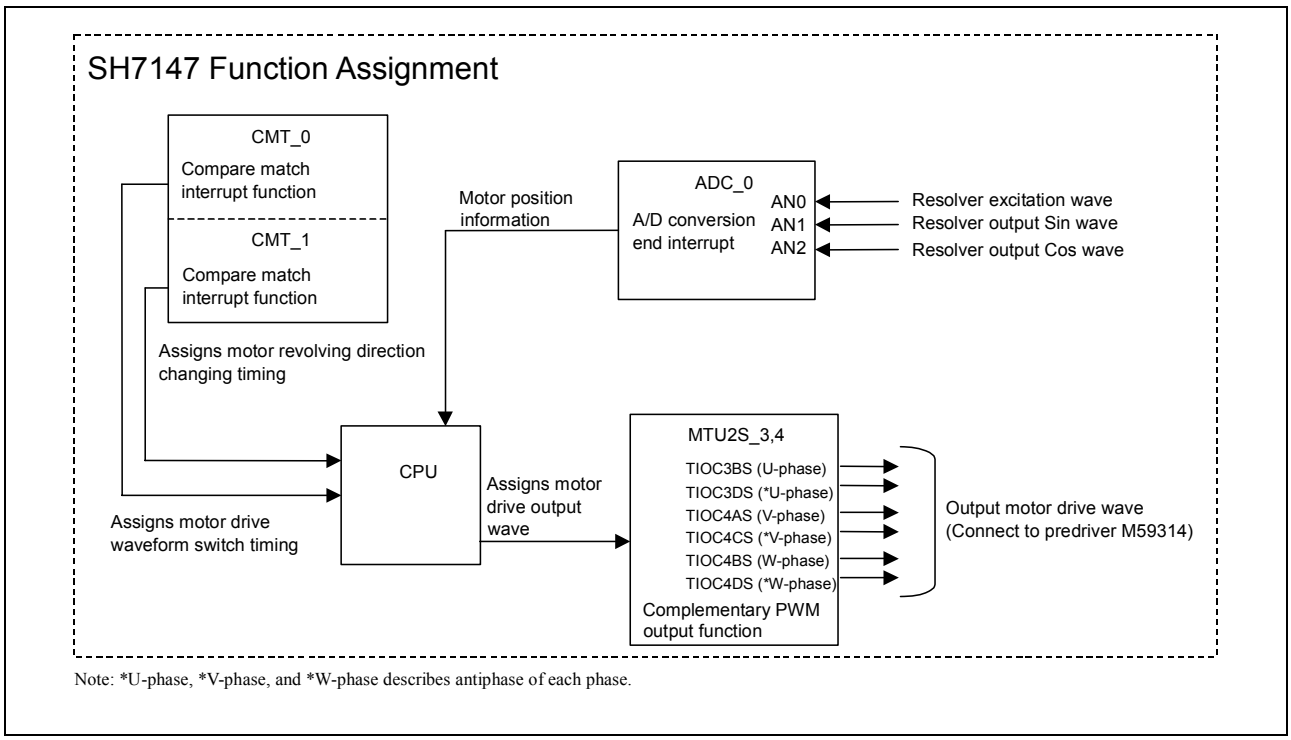

Figure 2 SH7147 Control Block Diagram

Following describes each function.

- A/D Conversion End Interrupt: After conversing resolver input excitation signal, resolver output Sin wave signal, and resolver output Cos wave signal into digital, it requests CPU for interrupt.
  A/D conversion result is stored to RAM by interrupt procedure.
- Compare Match Interrupt Function (CMT_0): Request CPU for interrupt at every 50µ[sec].
  Interrupt procedure determines motor position and outputs motor drive wave.
- Compare Match Interrupt Function (CMT_1): Request CPU for interrupts at every 0.8[sec]
  Interrupt procedure counts number of interrupting times and reverses motor revolving direction at every 5 interrupts process. (reverse motor revolving direction at every 0.8[sec] x 5 = 4[sec])
- Complementary PWM Output Function: Level output on positive phase side and chopping wave output on antiphase side.

3.2 Description of Functions Used

[1] A/D Converter

(a) Overview of A/D converter (ADC) operation

Resolver input excitation signal, resolver output Sin wave signal, and resolver output Cos wave signal are input from analog input pin 0,1,2 (AN0, AN1, and AN2) respectively. Input voltage of AN0 to 2 are sampled at a same timing at sample & hold circuit (the step before analog multiplexer) and A/D converted sequentially. The operation mode is continuous scan mode.

Figure 3 describes ADC_0 block diagram.

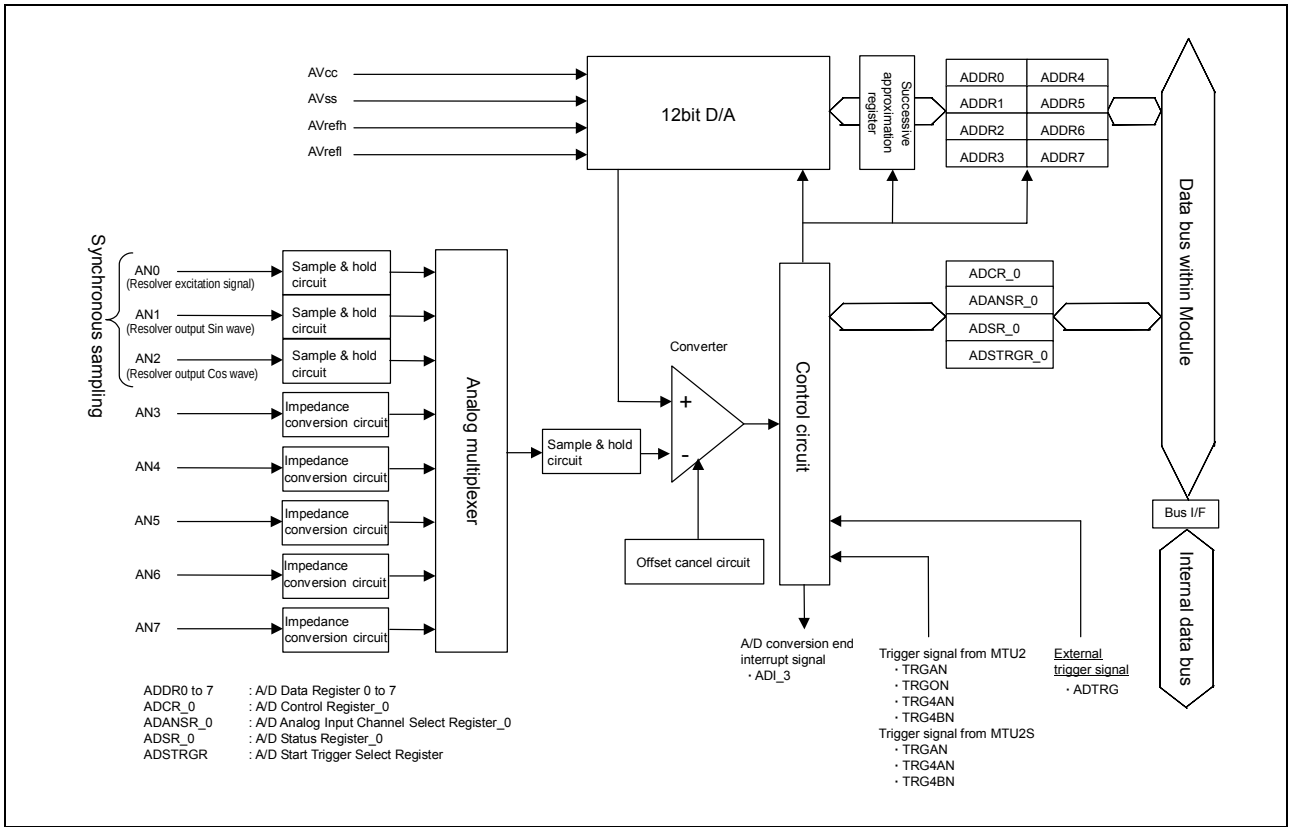

Figure 3 ADC_0 Block Diagram

(b) Overview of Resolver

Following describes resolver in motor position detection sensor, implemented in this system (Figure 4.) A coil is fixed on the rotor, and two coils are fixed 90 degree of each other on the stator. When resolver input excitation signal (Vr) is inputted, resolver output Sin wave (V_{sin}) and resolver output Cos wave (V_{cos}), which has 90 degree phase difference, are outputted from two coils on the stator.

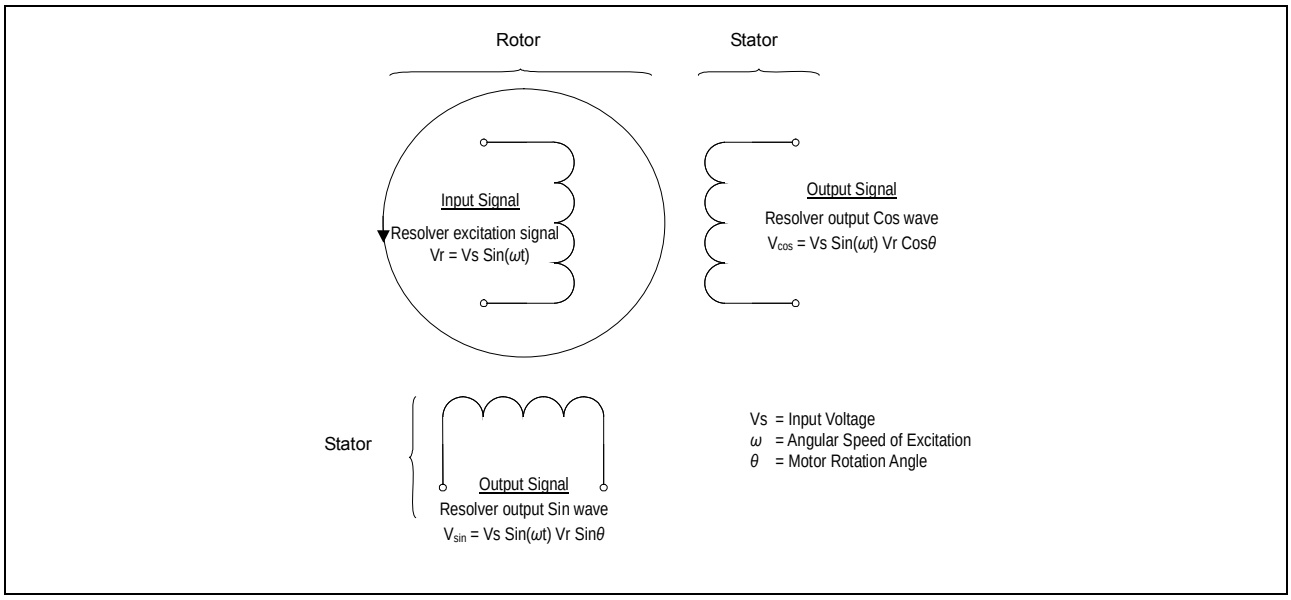

Figure 4 Diagram of Resolver Principle

(c) Resolver Input/Output Signal and Motor Rotation Position

Figure 5 describes resolver input excitation signal, resolver output Sin wave, and resolver output Cos wave per revolution of the motor. One motor revolution is equivalent to one cycle of Sin waveform and Cos waveform obtained from resolver output Sin wave and resolver output Cos wave.

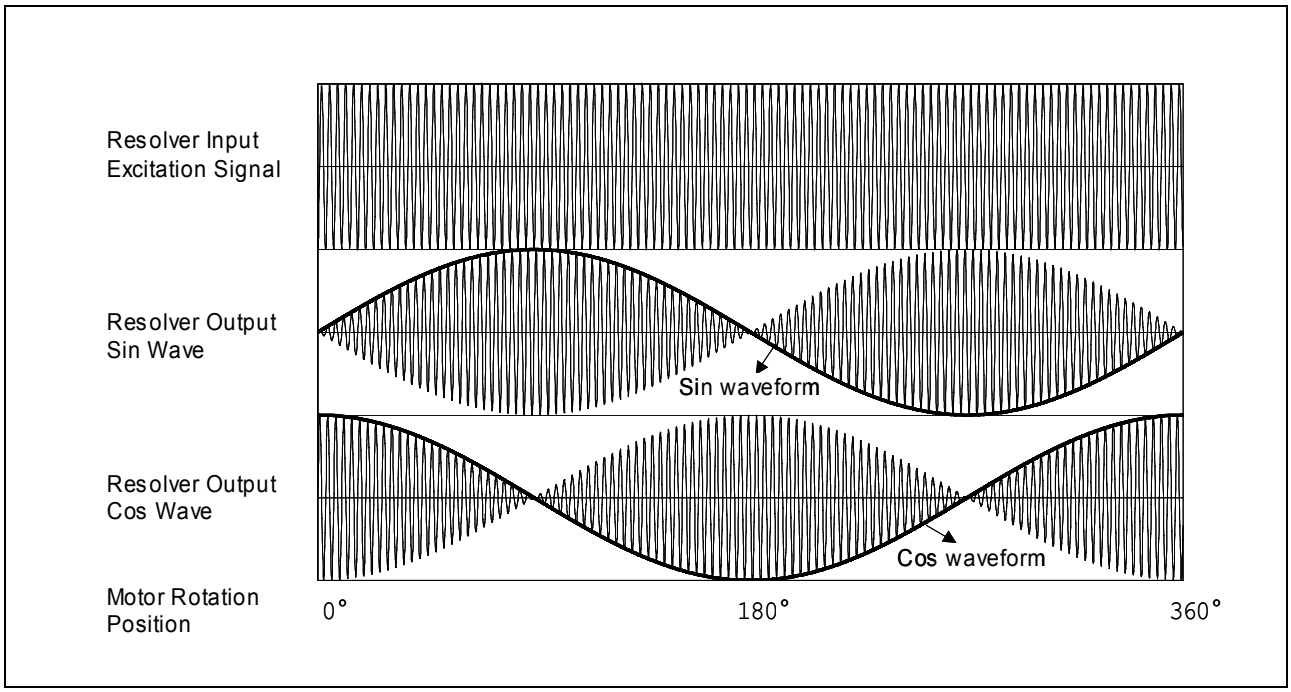

Figure 5 Resolver Input/Output Signal and Sin wave/ Cos wave (per motor revolution)

(d) Obtaining Sin waveform/ Cos waveform from each Resolver Output wave

Resolver output Sin wave/ Cos wave can be described in following formula, as been shown in Figure 4 Diagram of Resolver Principle.

Resolver Output Sin Wave Signal: $V_{sin} = VsSin(\omega t)VrSin\theta$

Resolver Output Cos Wave Signal: $V_{cos} = VsSin(\omega t)VrCos\theta$

If resolver input excitation signal ($VsSin(\omega t)$) is regarded as constant,

Resolver Output Sin Wave Signal: $V_{sin} = ASin\theta$

Resolver Output Cos Wave Signal: $V_{cos} = ACos\theta$

A : $VsVrSin(\omega t)$

Sin/Cos waveform can be obtained from above.

Following describes how to obtain Sinθ and Cosθ from resolver output Sin wave signal and resolver output Cos wave signal with regarding resolver input excitation signal as constant.

(i) In this control sample, voltage of resolver input excitation signal (VsSin(ωt)) is kept constant to have above "A" to be constant. "A" can be regarded as amplitude of resolver output signal. For this amplitude to have the maximum value, the voltage of resolver input signal for achieving the maximum amplitude of resolver output signal shall be obtained in beforehand by examining resolver input/output signal waveform and shall be defined as the reference value.

(ii) Voltage of resolver input excitation signal, resolver output Sin wave signal, and resolver output Cos wave signal are sampled at a same time and are A/D converted in continuous conversion mode.

(iii)When A/D conversion result of resolver input excitation signal and the reference voltage defined at (i) matches, in other words, when amplitude of resolver output signal becomes maximum, A/D conversion result of two output signals, ASinθ and ACosθ, are stored to RAM.

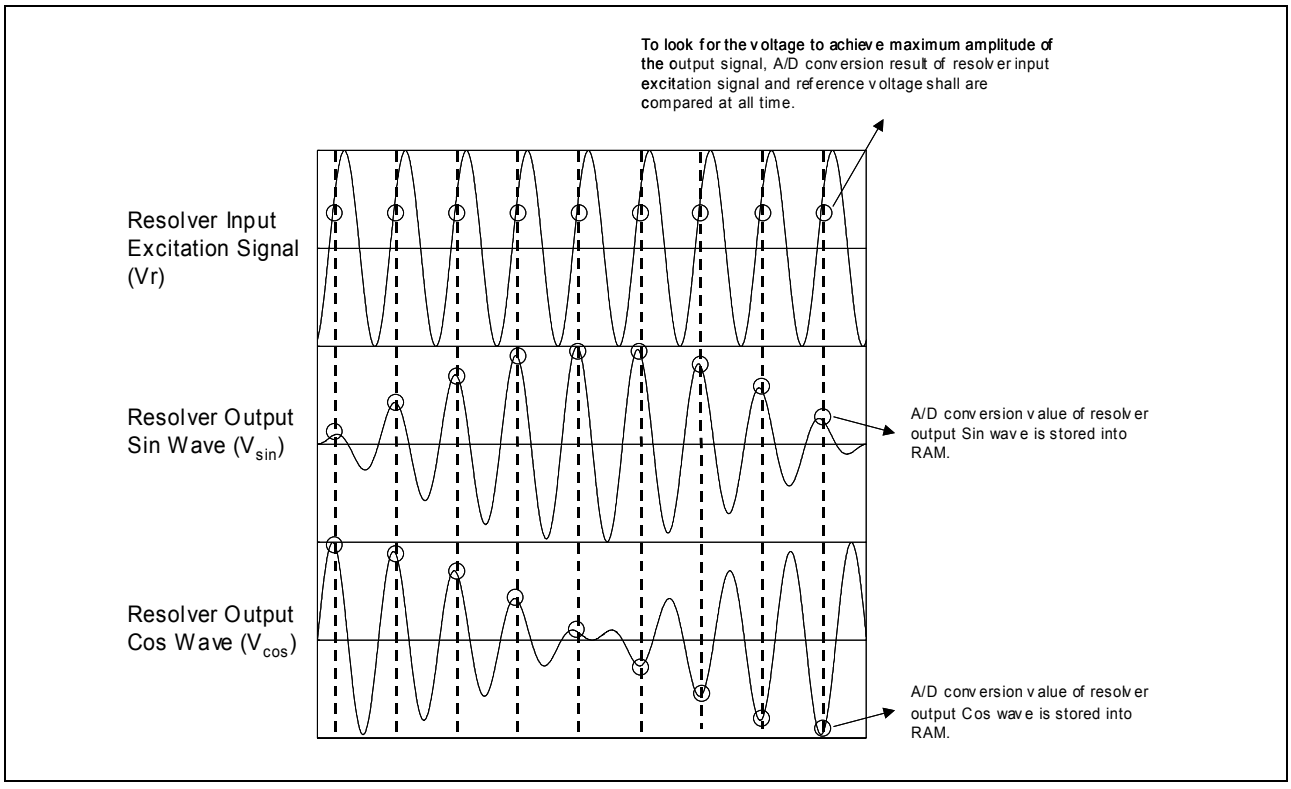

Figure 6 RAM Storage Timing of Resolver Output Signal

In this control sample, resolver input excitation signal cycle is set 900µ[sec] and motor rotation speed is set 300[rpm] (200m[sec]/revolution).

There are about 222 resolver input excitation signal cycles per motor revolution. As shown in Figure 6, Sinθ and Cosθ can be achieved once in one resolver input excitation signal cycle. Therefore, Sinθ and Cosθ can be achieved 222 times per one motor revolution and motor position can be detected per approximately 1.6 degree (360 degree/222) (A/D conversion error not regarded). As for this control sample requires motor position information per 30 degree, 1.6 degree motor positioning meets the requirement.

(e) Motor Position Determination

Following describes how to calculate an angle from Sin waveform/ Cos waveform achieved from resolver input/output signal. As for this sample controls motor in 120-degree energization method, it captures motor position information for each 30 degree. Figure 7 describes relation between motor position and Sin waveform/ Cos waveform.

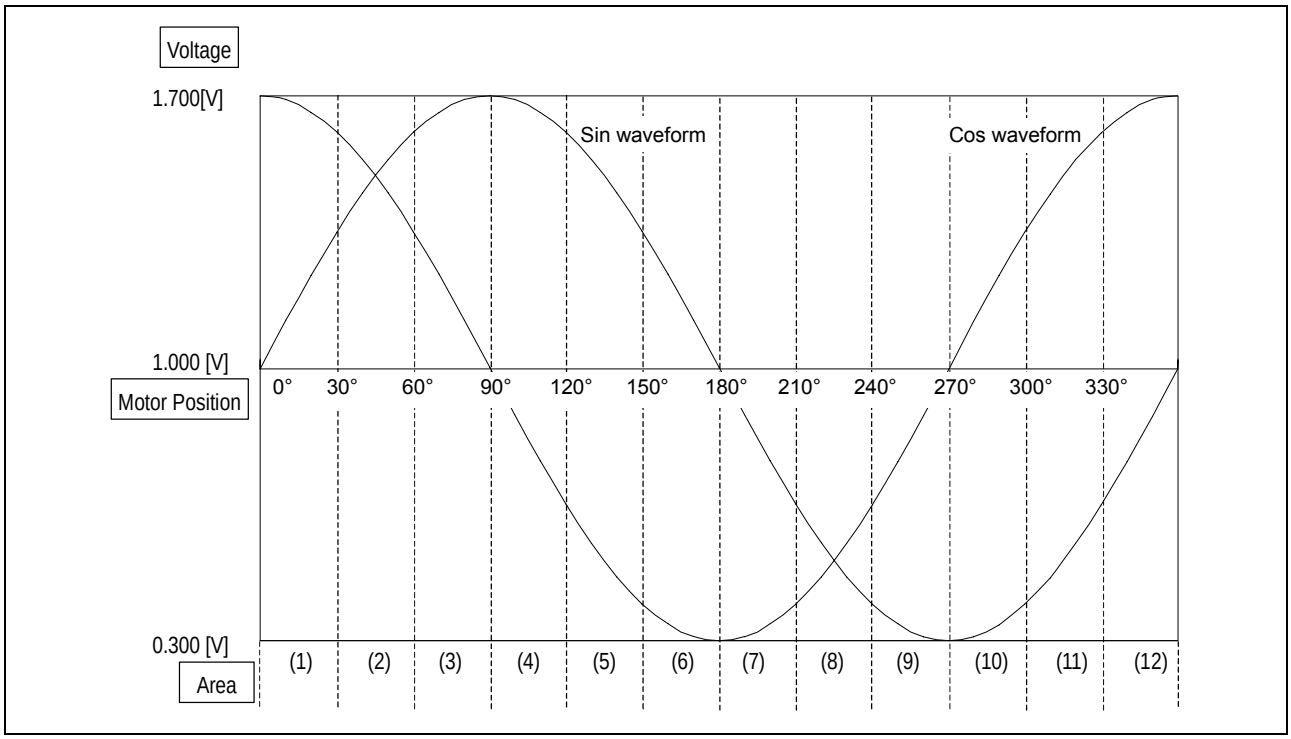

Figure 7 Motor Position and Sin Waveform/ Cos Waveform

Motor position can be determined from Sin waveform/Cos waveform with defining following provision.

Area (1): Cosθ > Cos30°	&	Sinθ > 0		Area (7): Cosθ > Cos150°	&	Sinθ < 0	
Area (2): Sin60° > Sinθ > Sin30°	&	Cosθ > 0		Area (8): Sin210° > Sinθ > Sin240°	&	Cosθ < 0	
Area (3): Sinθ > Sin60°	&	Cosθ > 0		Area (9): Sinθ > Sin240°	&	Cosθ < 0	
Area (4): Sinθ > Sin60°	&	Cosθ < 0		Area (10):Sinθ > Sin240°	&	Cosθ > 0	
Area (5): Sin60° > Sinθ > Sin30°	&	Cosθ < 0		Area (11):Sin210° > Sinθ > Sin240°	&	Cosθ > 0	
Area (6): Cosθ > Cos150°	&	Sinθ > 0		Area (12):Cosθ > Cos330°	&	Sinθ < 0	

Following describes data comparing Sin waveform/Cos waveform.

Analog input voltage of this control sample is maximum of 1.7[V], minimum of 0.3[V], and intermediate of 1.0[V]. Avref is set to 5.0[V].

As for motor is positioned by 30 degrees, voltage for comparing Sin/Cos waveform and constant to be compared is defined as Table 1.

Table 1 Voltage of Sin/Cos waveform and Comparison Value

	Sin270°, Cos180°	**Sin240°, Sin300°, Cos150°, Cos210°**	**Sin210°, Sin330°, Cos120°, Cos240°**
Voltage [V]	0.3	0.395	0.65
Compared constant	H'0F5	H'142	H'213
	Sin60°, Sin120°, Cos30°, Cos330°	Sin30°, Sin150°, Cos60°, Cos300°	Sin0°, Cos90°, Sin180°, Cos270°
Voltage [V]	1.0	1.35	1.605
Compared constant	H'332	H'451	H'522
	Sin90°, Cos0°		
Voltage [V]	1.7		
Compared constant	H'570		

Flowchart for actually determining motor position by this controller is described in Figure 8.

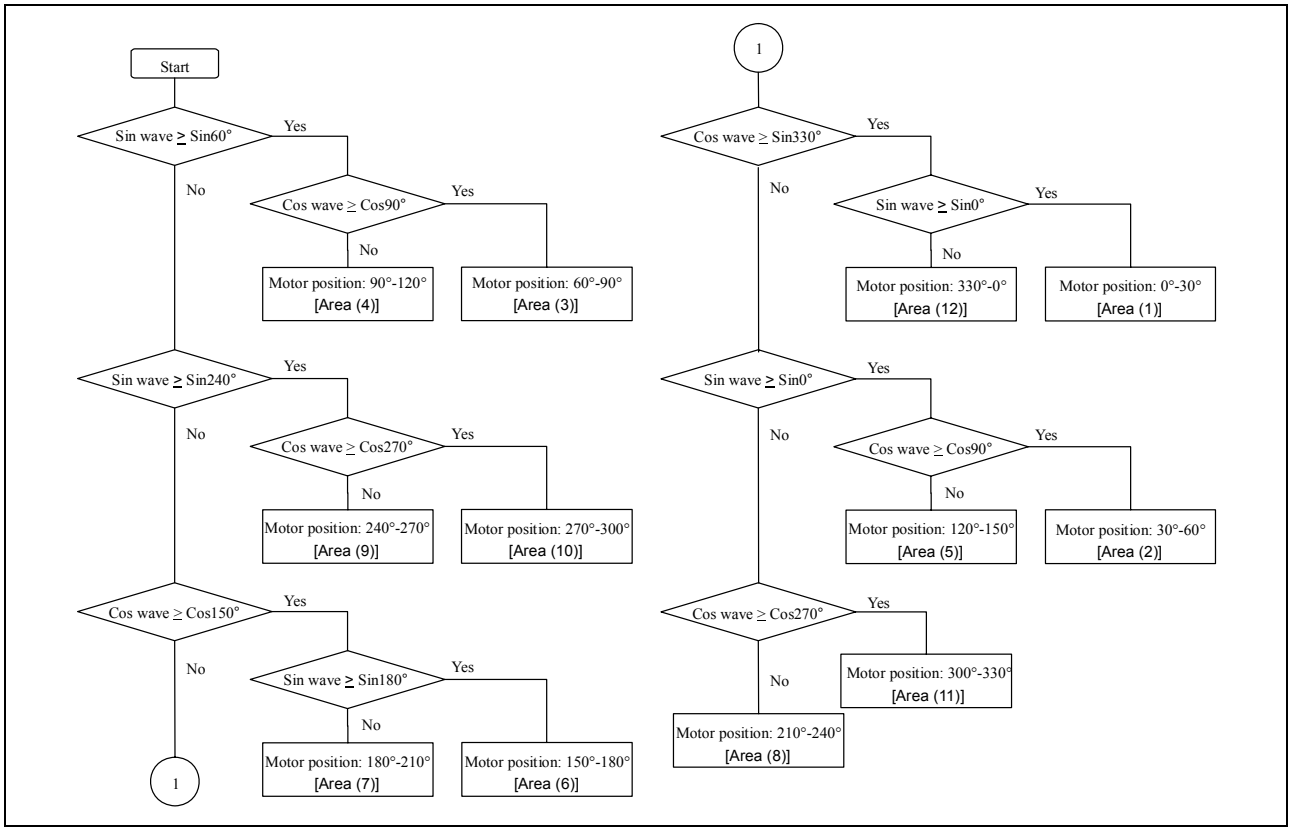

Figure 8 Motor Position Determining Flow

(f) Principle of A/D Converting Operation

Figure 9 describes principle of A/D converting operation. Table 2 shows details of software and hardware processing.

ADST bit is set to "1" by software and executes A/D conversion in continuous scan mode.

Following are processed by A/D conversion end interrupt.

- When A/D conversion result of resolver input excitation signal meets the condition described in Figure 6, A/D conversion result of resolver output Sin/Cos wave signal will be stored to RAM.
- When Figure 6 condition meets (when Sinθ and Cosθ are defined) for the first time, compare match interrupt by compare match timer_0 is enabled.

  (Compare match timer_0 (CMT_0) interrupt processing (output motor drive wave) is executed after motor position is defined.)

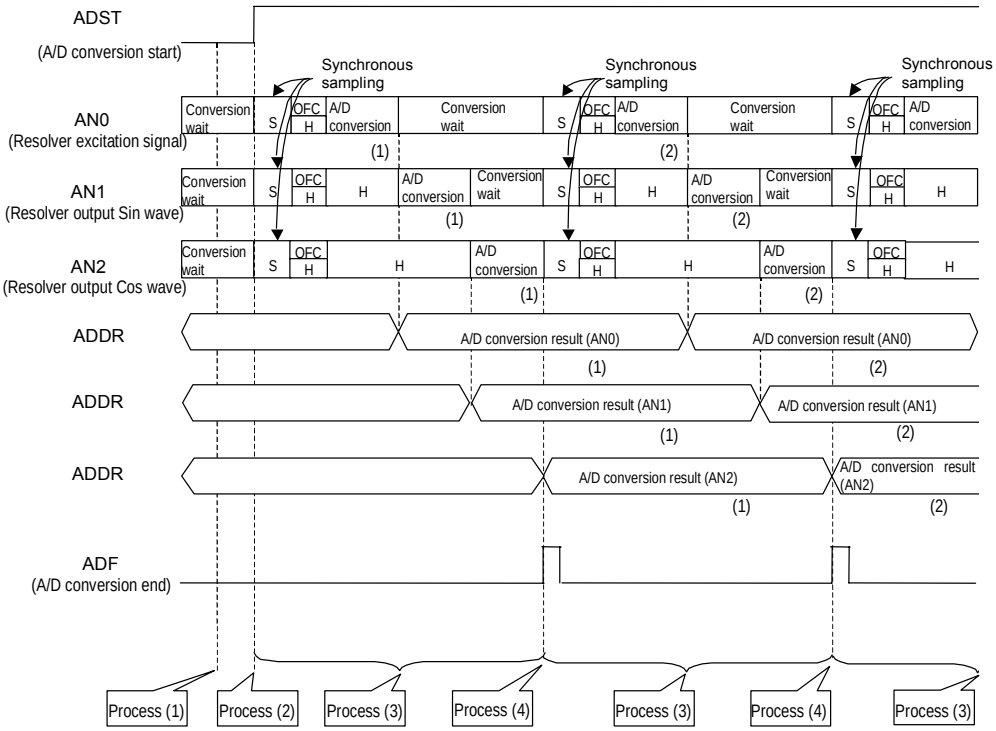

Figure 9 A/D Conversion Operation in Continuous Scan Mode

Table 2 Details of Software/Hardware process for A/D Conversion

	Software Process	Hardware Process
Process (1)	After releasing AD_0 module stop, followings are set • Set continuous scan mode by setting ADCS bit to “1” • Select AN0 to 2 as input channel by setting ANS0-2 to “1” • Enable A/D conversion end interrupt (ADI_3) by setting ADIE bit to “1”	• None
Process (2)	• Set A/D conversion start by setting ADST bit to “1”	• Start A/D conversion
Process (3)	• None	• Sample AN0-2 synchronously and execute A/D conversion • Store A/D conversion result sequentially to ADDR0-2
Process (4)	• Transfer A/D conversion result of AN0 (resolver input excitation signal) to RAM • If A/D conversion result of AN0 meets the condition (please refer figure 6), following are executed. (1) Store A/D conversion result of AN1 (resolver output Sin wave) and AN2 (resolver output Cos wave) to RAM. (2) When condition meets for the first time, CMT_0 compare match interrupt is enabled. • Clear ADF to “0”.	• Set ADF to “1”. • A/D conversion end interrupt been generated (ADI_3).

[2] MTU2S

(g) Overview of MTU2S Operation

Complementary PWM wave of three positive phase and three antiphase are outputted by using channel 3 and channel 4 of MTU2S. Operation mode is complementary PWM mode.

Correlation between UF bit, VF bit, and WF bit of timer gate control register S (TGCRS) and output level is shown in Table 3. (active level is set Low in this sample)

UF bit, VF bit, and WF bit are switched by software depending on motor revolving direction and motor position, and then motor drive wave is outputted (Figure 10).

Table 3 TGCRS (UF, VF, WF bit) and Output Level

Bit 2	Bit 1	Bit 0	Function					
WF	VF	UF	TIOC3BS	TIOC4AS	TIOC4BS	TIOC3DS	TIOC4CS	TIOC4D
			U-phase	V-phase	W-phase	*U-phase	*V-phase	*W-phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

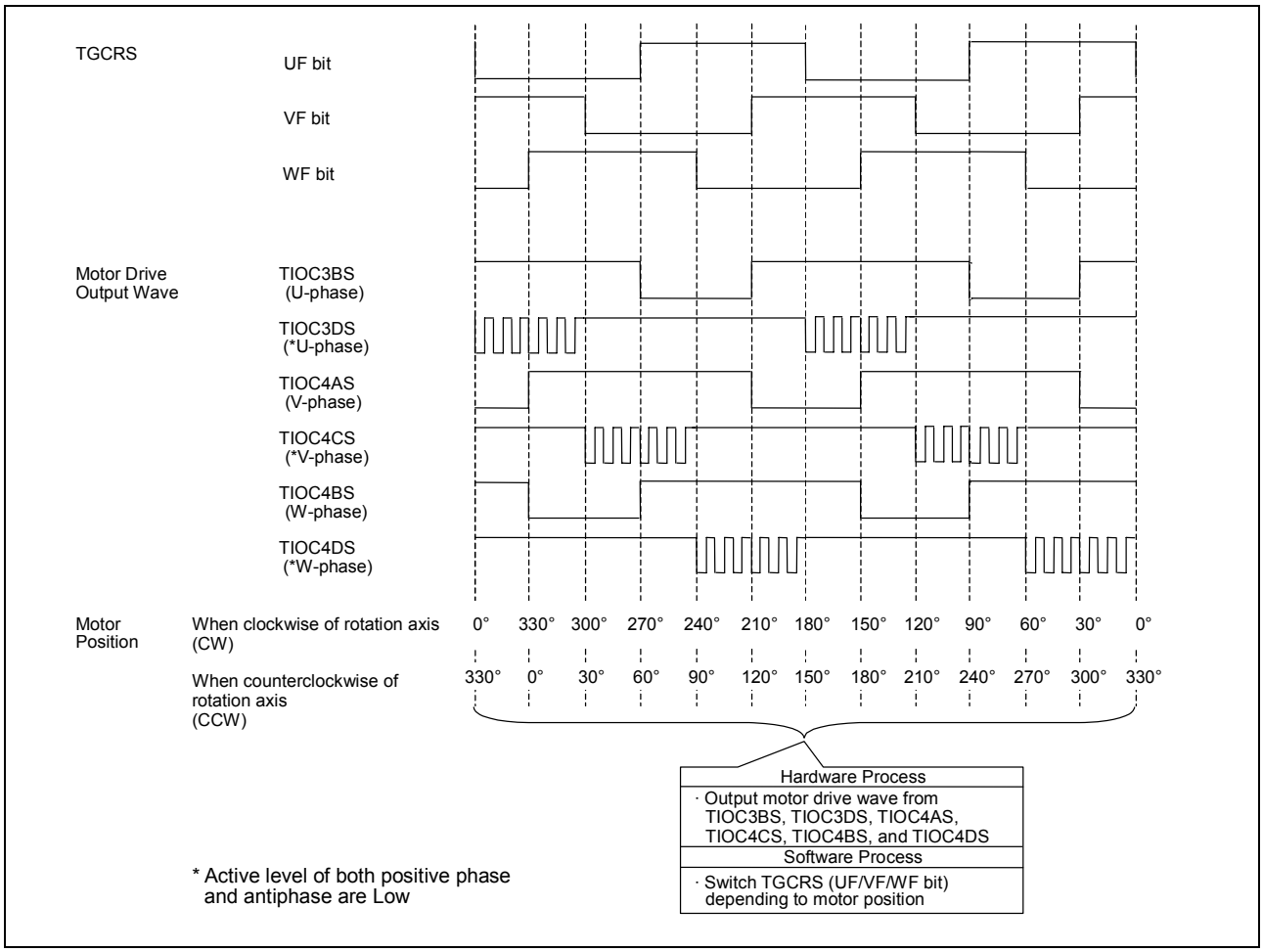

Figure 10 Motor Drive Wave by TGCRS (UF, VF, WF bit) Switch

(h) Duty Setting

Figure 11 describes MTU2S (ch3, ch4) in complementary PWM mode and duty setting is explained after it.

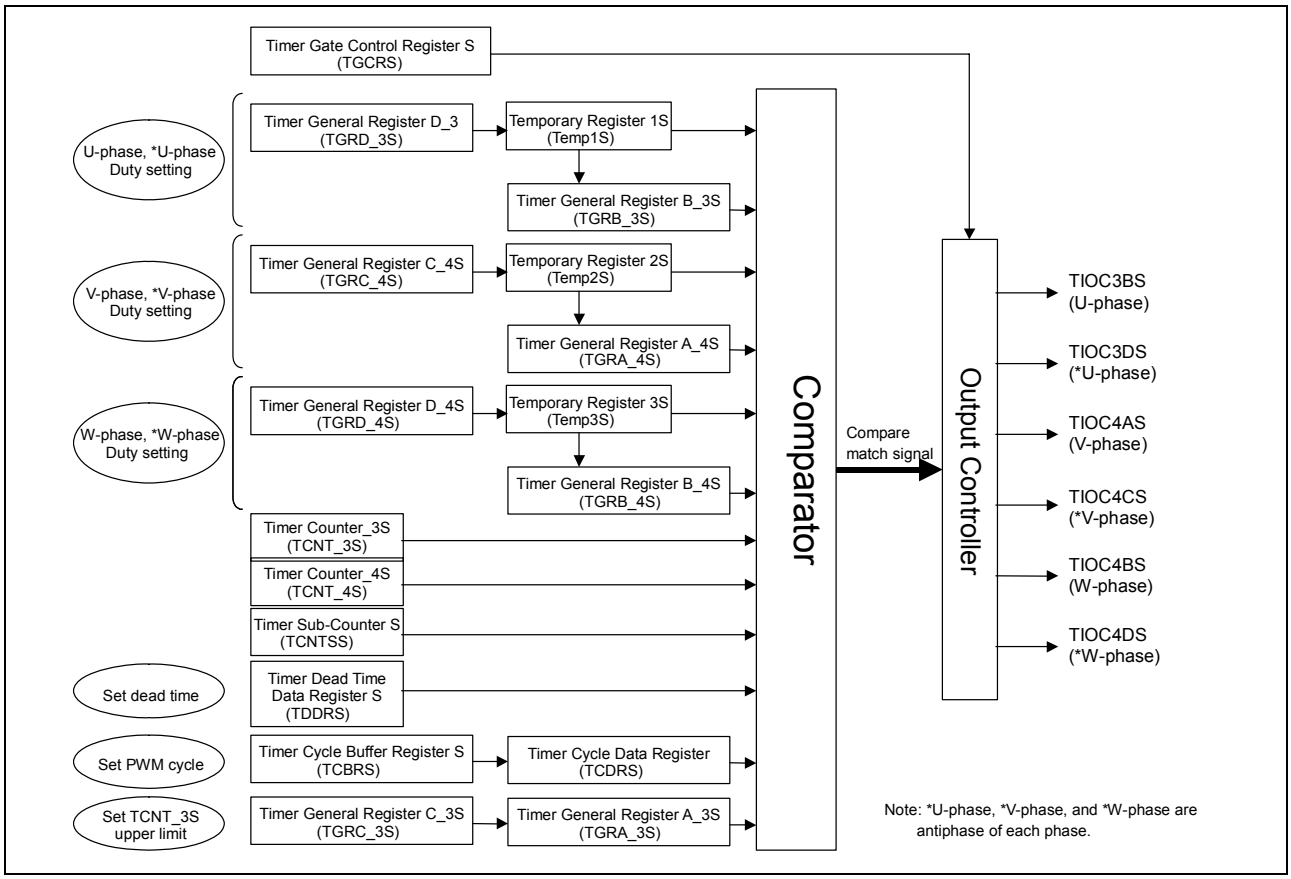

Figure 11 Block Diagram when in Complementary PWM Mode (Channel 3/4)

[Notes]
- CPU cannot directly access Temporary Register (Temp1S, 2S, 3S)
- Write into TGRB_3S, TGRA_4S, TGRB_4S, TGRA_3S, and TCDRS are prohibited during timer operation.
- When write into above listed register during timer operation is necessary, please write into their buffer register TGRD_3S, TGRC_4S, TGRD_4S, TGRC_3S, and TCBRS.

Following describes duty setting. This control sample outputs PWM only for antiphase. Level output is made for positive phase. Duty is determined by setting values of TGRB_3S, TGRA_4S, and TGRB_4S. Motor drive current changes by changing duty setting, and it allows changing motor torque and speed. Figure 12 describes operation principle and Table 4 shows details of software/hardware process.

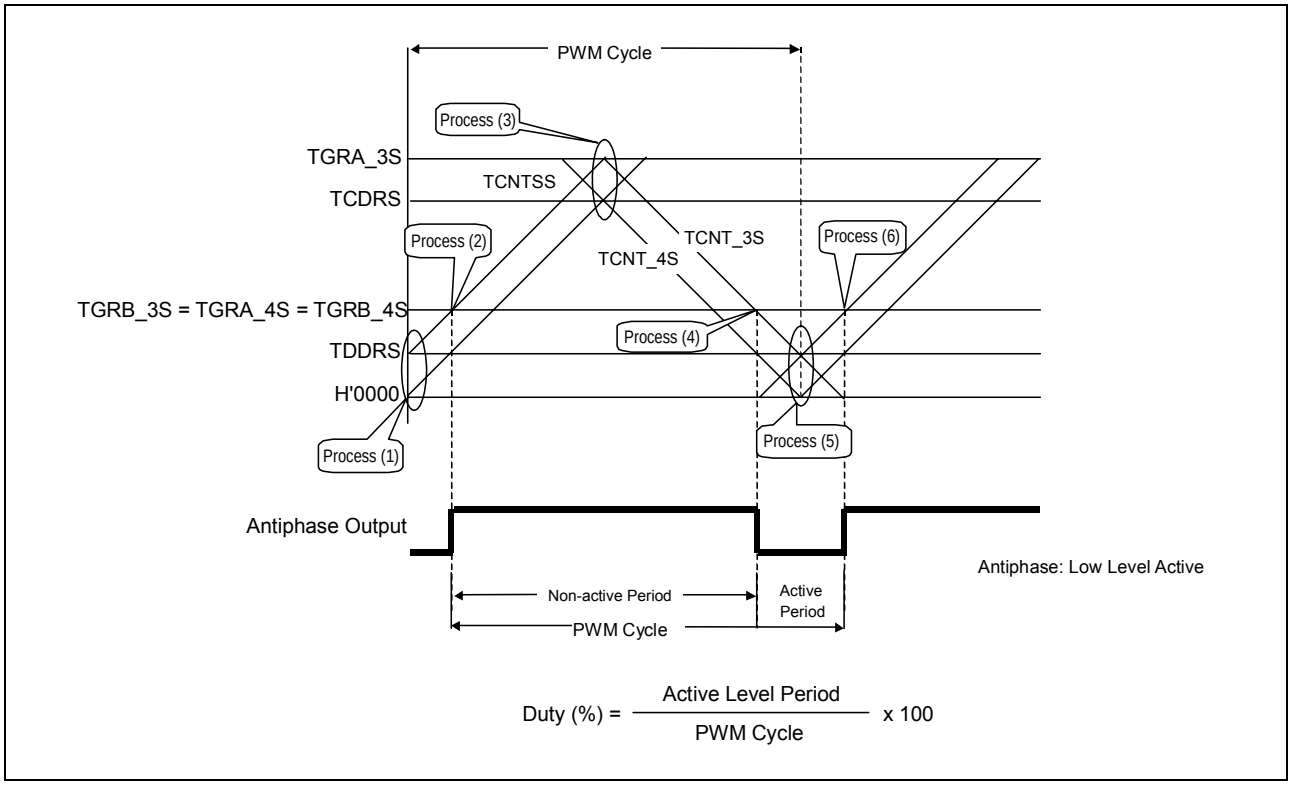

Figure 12 Counter Operation and Duty/PWM Cycle

Table 4 Details of Software/Hardware Processing for Duty Setting

	Software Process	Hardware Process
Process (1)	Configure followings after releasing AD_0 module stop and setting PFC • Timer count start • Output L level from pin	• TCNT_3S and TCNT_4S start up-count
Process (2)	• None	• Output H level from pin by compare match between TCNT_3S and TGRB_3S, TGRA_4S, and TGRB_4S
Process (3)	• None	• TCNT_3S starts down-count by compare match between TCNT_3S and TGRA_3S • TCNT_4S starts down-count by compare match between TCNT_4S and TCDRS
Process (4)	• None	• Output L level from pin by compare match between TCNT_3S and TGRB_3S, TGRA_4S, and TGRB_4S
Process (5)	• None	• TCNT_3S starts up-count by compare match between TCNT_3S and TGRA_3S • TCNT_4S starts up-count by compare match between TCNT_4S and TCDRS
Process (6)	• None	• Output L level from pin by compare match between TCNT_3S and TGRB_3S, TGRA_4S, and TGRB_4S

[3]CMT

(i) Overview of CMT Operation

Interrupt process is executed at every 50µ [sec] by using compare match timer channel 0 (CMT_0). Interrupt process switches timer gate control register S (TGCRS) of MTU2S for outputting a PWM wave, which is most suitable to motor position.

Also, interrupt process is executed at every 0.8 [sec] by using compare match timer channel 1 (CMT_1). This interrupt process counts number of interrupt process and reverses motor revolving direction (clockwise ↔ counterclockwise) at every 5 interrupt process (every 4 [sec]). Figure 13 describes CMT block diagram.

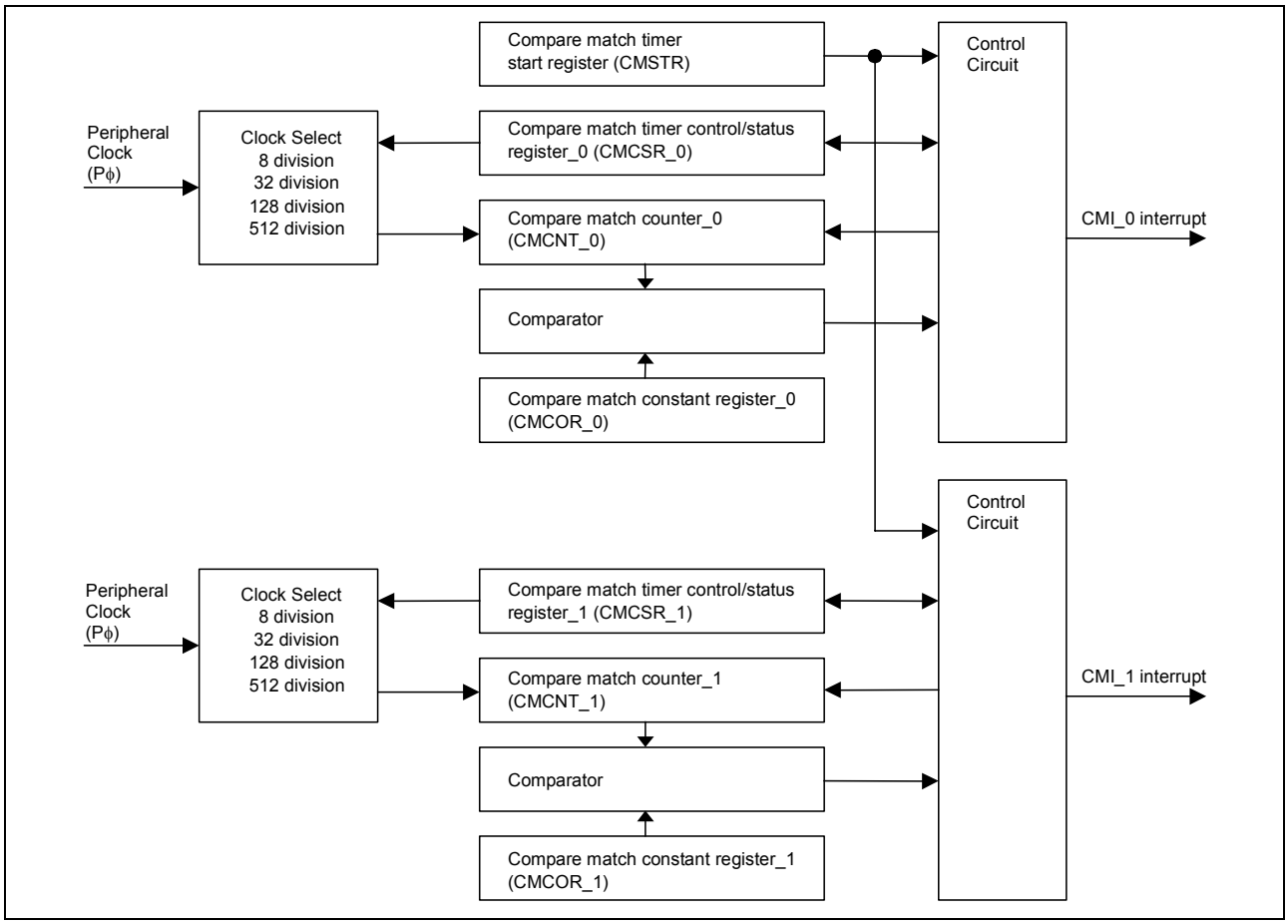

Figure 13 CMT Block Diagram

(j) Principle of CMT Operation

Figure 14 and Figure 15 describes principles of CMT_0 and CMT_1 operation.

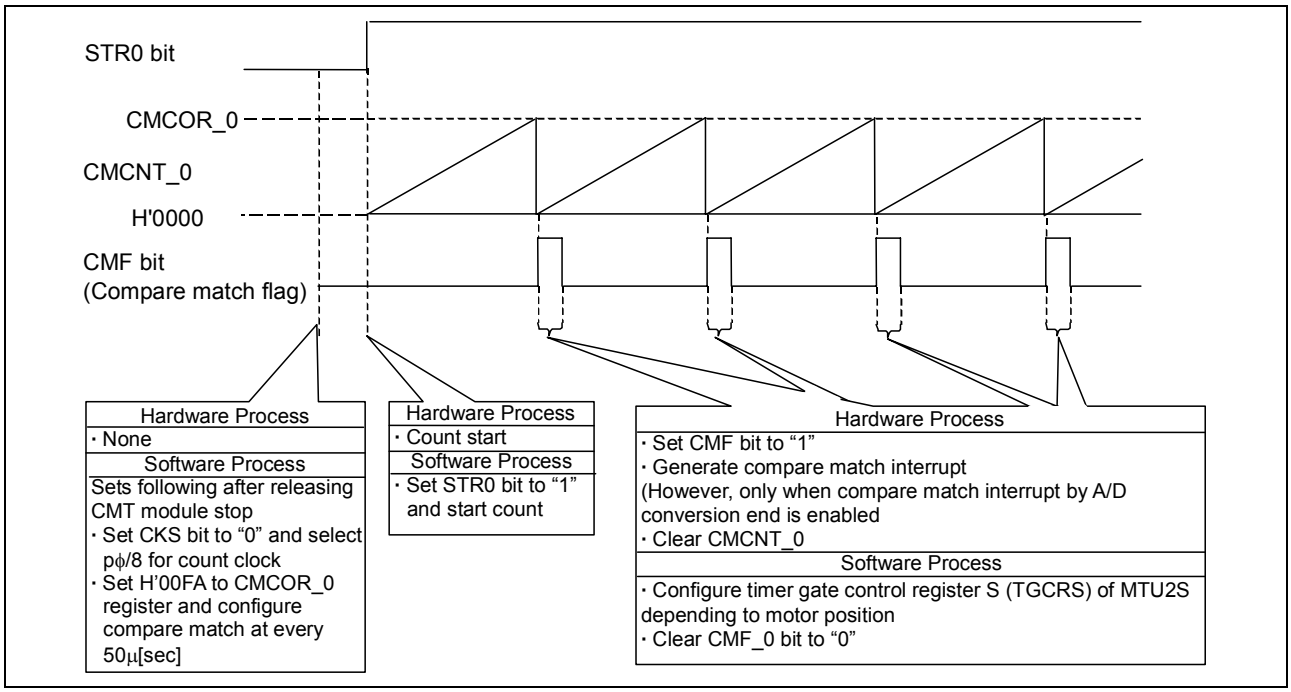

Figure 14 Principle of CMT_0 Operation (Assign TGCRS setting timing)

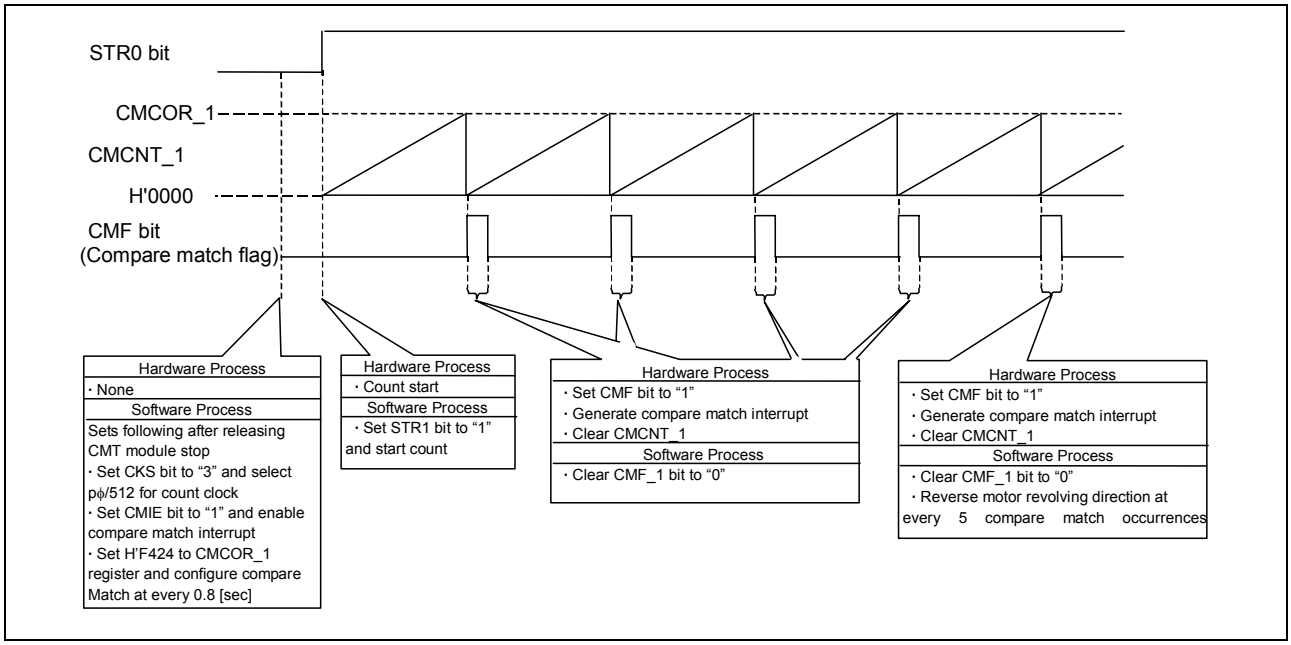

Figure 15 Principle of CMT_1 Operation (Assign motor revolving direction reversing timing)

3.3 Sample program Procedure

Following describes each register setting.

Table 5 ADC_0 Setting

Register Name	Bit Name	Bit	Function	Set Value
A/D Control Register (ADCR_0)	ADST	7	0: A/D conversion start	0
			1: A/D conversion stop	1
	ADSC	6	0: 1 cycle scan 1: Continuous scan	1
	ACE	5	0: Prohibit automatic clear, after ADDR read 1: Enable automatic clear, after ADDR read	0
	ADIE	4	0: Prohibit A/D conversion end interrupt 1: Enable A/D conversion end interrupt	1
	Reserve	3-2	—	0
	TRGE	1	0: Prohibit A/D conversion triggered by exterior or MTU2/2S 1: Enable A/D conversion triggered by exterior or MTU2/2S	0
	EXTRG	0	0: Select MTU2/MTU2S as A/D conversion start trigger 1: Select external pin (ADTRG) as A/D conversion start trigger	0
A/D Status Register_0 (ADSR_0)	Reserve	7-1	—	0
	ADF	0	A status flag indicating A/D conversion end 1 Set: A/D conversion of all the selected channels have ended. (Settable only by hardware process) 0 Clear: ・After read “1”, write “0” ・Activate DTC by A/D conversion end interrupt and read ADDR	0 Note1
A/D Analog Input Channel Select Register_0 (ADANSR_0)	ANS7	7	0: Unselect AN7 as input channel 1: Select AN7 as input channel	0
	ANS6	6	0: Unselect AN6 as input channel 1: Select AN6 as input channel	0
	ANS5	5	0: Unselect AN5 as input channel 1: Select AN5 as input channel	0
	ANS4	4	0: Unselect AN4 as input channel 1: Select AN4 as input channel	0
	ANS3	3	0: Unselect AN3 as input channel 1: Select AN3 as input channel	0
	ANS2	2	0: Unselect AN2 as input channel 1: Select AN2 as input channel	1
	ANS1	1	0: Unselect AN1 as input channel 1: Select AN1 as input channel	1
	ANS0	0	0: Unselect AN0 as input channel 1: Select AN0 as input channel	1
A/D Data Register 0 (ADDR0)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 0(AN0) (12bit)	
A/D Data Register 1 (ADDR1)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 1(AN1) (12bit)	

Register Name	Bit Name	Bit	Function	Set Value
A/D Data Register 2 (ADDR2)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 2(AN2) (12bit)	
A/D Data Register 3 (ADDR3)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 3(AN3) (12bit)	
A/D Data Register 4 (ADDR4)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 4(AN4) (12bit)	
A/D Data Register 5 (ADDR5)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 5(AN5) (12bit)	
A/D Data Register 6 (ADDR6)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 6(AN6) (12bit)	
A/D Data Register 7 (ADDR7)	Reserve	15-12	–	R
	ADD[11:0]	11-0	A/D conversion result of analog input pin 7(AN7) (12bit)	

Note: 1: ADF is automatically set to “1”

R: Read-only bit

Table 6 MTU2S Setting

Register Name	Bit Name	Bit	Function	Set Value
Timer Control Register_3S (TCR_3S)	CCLR[2:0]	7-5	Select counter clear factor of timer counter_3S (TCNT_3S) 000: Prohibit clear 001: Clear by compare match/input capture of TGRA_3S 010: Clear by compare match/input capture of TGRB_3S 011: Clear by counter clear of other channel, which is cleared/operated synchronically 100: Prohibit clear 101: Clear by compare match/input capture of TGRC_3S 110: Clear by compare match/input capture of TGRD_3S 111: Clear by counter clear of other channel, which is cleared/operated synchronically	H'000
	CKEG[1:0]	4-3	Select input clock edge Effective only when input clock is slower than MIΦ/4 When selected input clock is faster than MIΦ/4, it is counted at only rising edge. 00: Count at rising edge 01: Count at falling edge 10: Count at both rising/falling edge 11: Count at both rising/falling edge	H'00
	TPSC[2:0]	2-0	Select counter clock of timer counter_3S (TCNT_3S) 000: MIΦ/1 001: MIΦ/4 010: MIΦ/16 011: MIΦ/64 100: MIΦ/256 101: MIΦ/1024 110: External clock (TCLKA) 111: External clock (TCKLB)	H'000
Timer Control Register_4S (TCR_4S)	CCLR[2:0]	7-5	Select counter clear factor of timer counter_4S (TCNT_4S) 000: Prohibit clear 001: Clear by compare match/input capture of TGRA_4S 010: Clear by compare match/input capture of TGRB_4S 011: Clear by counter clear of other channel, which is cleared/operated synchronically 100: Prohibit clear 101: Clear by compare match/input capture of TGRC_4S 110: Clear by compare match/input capture of TGRD_4S 111: Clear by counter clear of other channel, which is cleared/operated synchronically	H'000
	CKEG[1:0]	4-3	Select input clock edge Effective only when input clock is slower than MIΦ/4 When selected input clock is faster than MIΦ/4, it is counted at only rising edge. 00: Count at rising edge 01: Count at falling edge 10: Count at both rising/falling edge 11: Count at both rising/falling edge	H'00

Register Name	Bit Name	Bit	Function	Set Value
Timer Control Register_4S (TCR_4S)	TPSC[2:0]	2-0	Select counter clock of timer counter_4S (TCNT_4S) 000: MIΦ/1 001: MIΦ/4 010: MIΦ/16 011: MIΦ/64 100: MIΦ/256 101: MIΦ/1024 110: External clock (TCLKA) 111: External clock (TCKLB)	H'000
Timer Mode Register_3S (TMDR_3S)	Reserve	7	—	0
	Reserve	6	—	0
	BFB	5	0: TGRB_3S and TGRD_3S are in normal operation 1: TGRB_3S and TGRD_3S are in buffer operation	1
	BFA	4	0: TGRA_3S and TGRC_3S are in normal operation 1: TGRA_3S and TGRC_3S are in buffer operation	1
	MD[3:0]		0000: Normal operation 0001: Setting prohibited 0010: PWM mode 1 0011: Setting prohibited 0100: Setting prohibited 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: Reset-synchronized PWM mode 1001: Setting prohibited 1010: Setting prohibited 1011: Setting prohibited 1100: Setting prohibited 1101: Complementary PWM mode 1 (transfer at peaks) 1110: Complementary PWM mode 2 (transfer at valleys) 1111: Complementary PWM mode 3 (transfer at peaks and valleys)	H'1101

Register Name	Bit Name	Bit	Function	Set Value
Timer Mode Register_4S (TMDR_4S)	Reserve	7	—	0
	Reserve	6	—	0
	BFB	5	0: TGRB_4S and TGRD_4S are in normal operation 1: TGRB_4S and TGRD_4S are in buffer operation	1
	BFA	4	0: TGRA_4S and TGRC_4S are in normal operation 1: TGRA_4S and TGRC_4S are in buffer operation	1
	MD[3:0]	3-0	0000: Normal operation 0001: Setting prohibited 0010: PWM mode 1 0011: Setting prohibited 0100: Setting prohibited 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: Setting prohibited 1001: Setting prohibited 1010: Setting prohibited 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited	H'0000 Note2
Timer Output Master Enable Register S (TOERS)	Reserve	7-6	—	3
	OE4D	5	0: Prohibit MTU2S output by TIOC4DS pin (Non-active level) Note2 1: Enable MTU2S output by TIOC4DS pin (*W-phase)	1
	OE4C	4	0: Prohibit MTU2S output by TIOC4CS pin (Non-active level) Note2 1: Enable MTU2S output by TIOC4CS pin	1
	OE3D	3	0: Prohibit MTU2S output by TIOC3DS pin (Non-active level) Note2 1: Enable MTU2S output by TIOC3DS pin	1
	OE4B	2	0: Prohibit MTU2S output by TIOC4BS pin (Non-active level) Note2 1: Enable MTU2S output by TIOC4BS pin	1
	OE4A	1	0: Prohibit MTU2S output by TIOC4AS pin (Non-active level) Note2 1: Enable MTU2S output by TIOC4AS pin	1
	OE3B	0	0: Prohibit MTU2S output by TIOC3BS pin (Non-active level) Note2 1: Enable MTU2S output by TIOC3BS pin	1

Register Name	Bit Name	Bit	Function	Set Value
Timer Gate Control Register S (TGCRS)	Reserve	7	—	1
	BDC	6	0: Disable TGCRS register (this register) setting 1: Enable TGCRS register (this register) setting	1
	N	5	Select output state of antiphase output pin (TIOC3DS (*U-phase), TIOC4CS (*V-phase), TIOC4DS (*W-phase)) Note3 0: Level output 1: Reset-synchronized PWM/ Complementary PWM	1
	P	4	Select output state of positive phase output pin (TIOC3BS (U-phase), TIOC4AS (V-phase), TIOC4BS (W-phase)) Note3 0: Level output 1: Reset-synchronized PWM/ Complementary PWM	0
	FB	3	0: Switch output phase by external input 1: Switch output phase by software (setting of UF, VF, and WF bit of TGCRS)	1
	WF	2	Set output phase by switching WF, VF, and UF bit Please refer Table 7	H'000 Note4
Timer Output Control Register1S (TOCR1S)	Reserve	7	—	0
	PSYE	6	0: Prohibit toggle output synchronized to PWM cycle 1: Enable toggle output synchronized to PWM cycle	0
	Reserve	5-4	—	0
	TOCL	3	0: Enable write to TOCS, OLSN, and OLSP bit 1: Prohibit write to TOCS, OLSN, and OLSP bit	0
	TOCS	2	Set output level in complementary PWM mode/reset-synchronized PWM mode 0: TOCR1 setting effective 1: TOCR2 setting effective	0
	OLSN	1	Select antiphase output level in complementary/ reset-synchronized PWM mode 0: Initial output → H, Active level → L, Compare match while up-count → H, Compare match while down-count → L 1: Initial output → L, Active level → H, Compare match while up-count → L, Compare match while down-count → H	0
	OLSP	0	Select positive phase output level in complementary/ reset-synchronized PWM mode 0: Initial output → H, Active level → L, Compare match while up-count → L, Compare match while down-count → H 1: Initial output → L, Active level → H, Compare match while up-count → H, Compare match while down-count → L	0

Register Name	Bit Name	Bit	Function	Set Value
Timer Counter_3S (TCNT_3S)		15-0	Counter of channel 3	H'0000 Note5
Timer Counter _4S (TCNT_4S)		15-0	Counter of channel 4	H'0000 Note5
Timer Cycle Data Register S (TCDRS)		15-0	A register been used only in complementary PWM mode Set PWM carrier period to 1/2	H'0640
Timer Dead Time Data Register S (TDDRS)		15-0	A register been used only in complementary PWM mode Set dead time	H'0001
Timer General Register A_3S (TGRA_3S)		15-0	When complementary PWM mode is set; Set (1/2 of PWM carrier period + dead time)	H'0641
Timer General Register B_3S (TGRB_3S)		15-0	When complementary PWM mode is set; Compare register of TIOC3BS(U-phase)/TIOC3DS(*U-phase) pin output	H'00A0
Timer General Register A_4S (TGRA_4S)		15-0	When complementary PWM mode is set; Compare register of TIOC3AS(V-phase)/TIOC4CS(*V-phase) pin output	H'00A0
Timer General Register B_4S (TGRB_4S)		15-0	When complementary PWM mode is set; Compare register of TIOC4BS(W-phase)/ TIOC4DS(*W-phase) pin output	H'00A0
Timer Cycle Buffer Register S (TCBRS)		15-0	A register been used only in complementary PWM mode Buffer register of TCDRS	H'0640
Timer General Register C_3S (TGRC_3S)		15-0	When complementary PWM mode is set; Buffer register of TGRA_3S TGRC_3S is set when changing TGRA_3S during timer operation	H'0641
Timer General Register D_3S (TGRD_3S)		15-0	When complementary PWM mode is set; Buffer register of TGRB_3S TGRD_3S is set when changing TGRB_3S during timer operation	H'00A0
Timer General Register C_4S (TGRC_4S)		15-0	When complementary PWM mode is set; Buffer register of TGRA_4S TGRC_4S is set when changing TGRA_4S during timer operation	H'00A0
Timer General Register D_4S (TGRD_4S)		15-0	When complementary PWM mode is set; Buffer register of TGRB_4S TGRD_4S is set when changing TGRB_4S during timer operation	H'00A0
Timer Dead Time Enable Register S (TDERS)	Reserve	7-1	—	
	TDER	0	0: Do not generate dead time 1: Generate dead time	1

Register Name	Bit Name	Bit	Function	Set Value
Timer Start Register S (TSTRS)	CST4	7	0: Count stop of timer counter_4S (TCNT_4S)	0
			1: Count start of timer counter_4S (TCNT_4S)	1
	CST3	6	0: Count stop of timer counter_3S (TCNT_3S)	0
			1: Count start of timer counter _3S (TCNT_3S)	1
	Reserve	5-3	—	0
	Reserve	2	—	0
	Reserve	1	—	0
	Reserve	0	—	0

Note2 Non-active level complies TOCR1S/2S setting

Note3 U-phase, V-phase, and W-phase are positive phase.

*U-phase, *V-phase, and *W-phase are antiphase.

Note4 Initial value. This control sample switches WF, VF, and UF bit depending on motor position.

Note5 Initial value.

Changes during timer operation.

Table 7 Output Level Select Function

Bit 2	Bit 1	Bit 0	Function					
WF	VF	UF	TIOC3BS	TIOC4AS	TIOC4BS	TIOC3DS	TIOC4CS	TIOC4D
			U-phase	V-phase	W-phase	*U-phase	*V-phase	*W-phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

Table 8 CMT_0,1 Setting

Register Name	Bit Name	Bit	Function	Set Value
Compare Match Timer Start Register (CMSTR)	Reserve	15-2	—	0
	STR1	1	0: Count stop of compare match counter_1	0
			1: Count start of compare match counter_1	1
	STR0	0	0: Count stop of compare match counter_0	0
			1: Count start of compare match counter_0	1
Compare Match Timer Control Register_0 (CMCSR_0)	Reserve	15-8	—	0
	CMF	7	A flag which indicates match of compare match constant register_0 (CMCOR_0) value and compare match timer counter_0 (CMCNT_0) value. 1 Set: CMCOR_0 and CMCNT_0 matches (automatically set by hardware) 0 Clear: ・Write “0” ・When CMT register is accessed while DTC module is activated by CMI_0 interrupt and MRB/DISEL bit of DTC is “0”	0 *Note6
	CMIE	6	0: Prohibit compare match interrupt (CMI_0) 1: Enable compare match interrupt (CMI_0)	1
	Reserve	5-2	—	0
	CKS[1:0]	1-0	Select frequency for counting CMCNT_0 00: PΦ/8 01: PΦ/32 10: PΦ/128 11: PΦ/512	H'00
Compare Match Counter_0 (CMCNT_0)		15-0	Counter for up-count	H'0000
Compare Match Constant Register_0 (CMCOR_0)		15-0	Set the duration until matching with CMCNT_0	H'00C8
Compare Match Timer Control Register_1 (CMCSR_1)	Reserve	15-8	—	0
	CMF	7	A flag which indicates match of compare match constant register_1 (CMCOR_1) value and compare match timer counter_1 (CMCNT_1) value. 1 Set: CMCOR_1 and CMCNT_1 matches (automatically set by hardware) 0 Clear: ・Write “0” ・When CMT register is accessed while DTC module is activated by CMI_1 interrupt and MRB/DISEL bit of DTC is “0”	0 *Note6
	CMIE	6	0: Prohibit compare match interrupt (CMI_1) 1: Enable compare match interrupt (CMI_1)	1
	Reserve	5-2	—	0
	CKS[1:0]	1-0	Select frequency for counting CMCNT_1 00: PΦ/8 01: PΦ/32 10: PΦ/128 11: PΦ/512	H'11
Compare MatchCounter_1 (CMCNT_1)		15-0	Counter for up-count	H'0000
Compare Match Constant Register_1 (CMCOR_1)		15-0	Set the duration until matching with CMCNT_1	H'C350

Note6: CMF is automatically set to “1”

Table 9 PFC Setting

Register Name	Bit Name	Bit	Function	Set Value
Port E IO Register H (PEIORH)	Reserve	15-6	—	0
	PE21	5	0: PE21 pin (2pin) is input 1: PE21 pin (2pin) is output	1
	PE20	4	0: PE20 pin (4pin) is input 1: PE20 pin (4pin) is output	1
	PE19	3	0: PE19 pin (5pin) is input 1: PE19 pin (5pin) is output	1
	PE18	2	0: PE18 pin (6pin) is input 1: PE18 pin (6pin) is output	1
	PE17	1	0: PE17 pin (7pin) is input 1: PE17 pin (7pin) is output	1
	PE16	0	0: PE16 pin (8pin) is input 1: PE16 pin (8pin) is output	1
Port E Control Register H2 (PECRH2)	Reserve	15-6	—	0
	PE21MD1 PE21MD0	5 4	00: PE21 input/output (port) 01: TIOC4DS input/output (MTU2S) 10: WRL output (BSC) 11: Setting prohibited	H'01
	Reserve	3-2	—	0
	PE20MD1 PE20MD0	1 0	00: PE20 input/output (port) 01: TIOC4CS input/output (MTU2S) 10: Setting prohibited 11: Setting prohibited	H'01
Port E Control Register H1 (PECRH1)	Reserve	15-14	—	0
	PE19MD1 PE19MD0	13 12	00: PE19 input/output (port) 01: TIOC4BS input/output (MTU2S) 10: RD output (BSC) 11: Setting prohibited	H'01
	Reserve	11-10	—	0
	PE18MD1 PE18MD0	9 8	00: PE18 input/output (port) 01: TIOC4AS input/output (MTU2S) 10: CS1 output (BSC) 11: Setting prohibited	H'01
	Reserve	7-6	—	0
	PE17MD1 PE17MD0	5 4	00: PE17 input/output (port) 01: TIOC3DS input/output (MTU2S) 10: CS0 output (BSC) 11: Setting prohibited	H'01
	Reserve	3	—	0
	PE16MD2 PE16MD1 PE16MD0	2 1 0	000: PE16 input/output (port) 001: TIOC3BS input/output (MTU2S) 010: WAIT output (BSC) Other than above: Setting prohibited	H'01

Table 10 STBYCR Setting

Register Name	Bit Name	Bit	Function	Set Value
Standby Control Register (STBCR4)	MSTP23	7	0: MTU2S is operating 1: Clock supply to MTU2S is halted (MTU2S is not operating)	0
	MSTP22	6	0: MTU2 is operating 1: Clock supply to MTU2 is halted (MTU2 is not operating)	1
	MSTP21	5	0: CMT is operating 1: Clock supply to CMT is halted (CMT is not operating)	0
	MSTP20	4	0: A/D_1 is operating 1: Clock supply to A/D_1 is halted (A/D_1 is not operating)	1
	MSTP19	3	0: A/D_0 is operating 1: Clock supply to A/D_0 is halted (A/D_0 is not operating)	0
	Reserve	2-0	—	1

Table 11 INTC Setting

Register Name	Bit Name	Bit	Function	Set Value
Interrupt Priority Register J (IPRJ)	IPR[15:12]	15-12	Set the priority order of CMT_0 interrupt factor. Priority level is set between 0 to 15 0000: Priority level 0 (lowest) → 11111: Priority level 15 (highest)	H'0010
	IPR[11:8]	11-8	Set the priority order of CMT_1 interrupt factor. Priority level is set between 0 to 15 0000: Priority level 0 (lowest) → 11111: Priority level 15 (highest)	H'0011
Interrupt Priority Register K (IPRK)	IPR[7:4]	7-4	Set the priority order of AD_0 interrupt factor. Priority level is set between 0 to 15 0000: Priority level 0 (lowest) → 11111: Priority level 15 (highest)	H'0001

Table 12 CPG Setting

Register Name	Bit Name	Bit	Function	Set Value
Frequency Control Register (FRQCR)	Reserve	15	—	0
	IFC[2:0]	14-12	Division ratio of internal clock (Iϕ) frequency 0: ×1, When input clock 8MHz Iϕ: 64MHz	0
	BFC[2:0]	11-9	Division ratio of bus clock (Bϕ) frequency 1: ×1/2, When input clock 8MHz Bφ: 32MHz	1
	PFC[2:0]	8-6	Division ratio of peripheral clock (Pϕ) frequency 1: ×1/2, When input clock 8MHz Pϕ: 32MHz	1
	MIFC[2:0]	5-3	Division ratio of MTU2S clock (MIϕ) frequency 0: ×1, When input clock 8MHz MIϕ: 64MHz	0
	MPFC[2:0]	2-0	Division ratio of MTU2S clock (Iϕ) frequency 1: ×1/2, When input clock 8MHz MPϕ: 32MHz	1

3.4 Used Function Setting Procedure

Following describes procedure flow of sample programs. Please refer Table 1 to 7 or “SH7147 Hardware Manual” for each register settings.

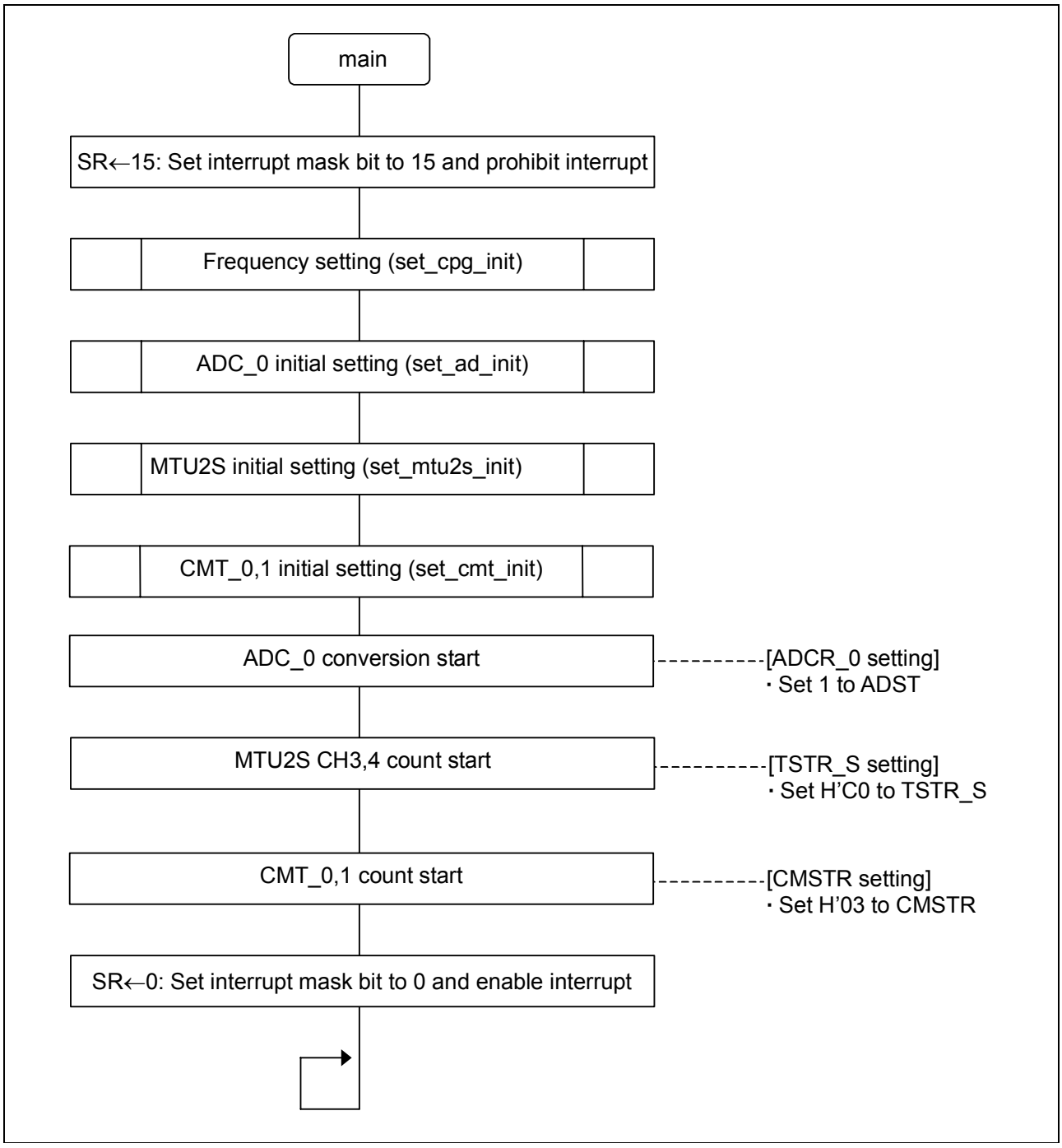

Figure 16 Main Function Processing Flow

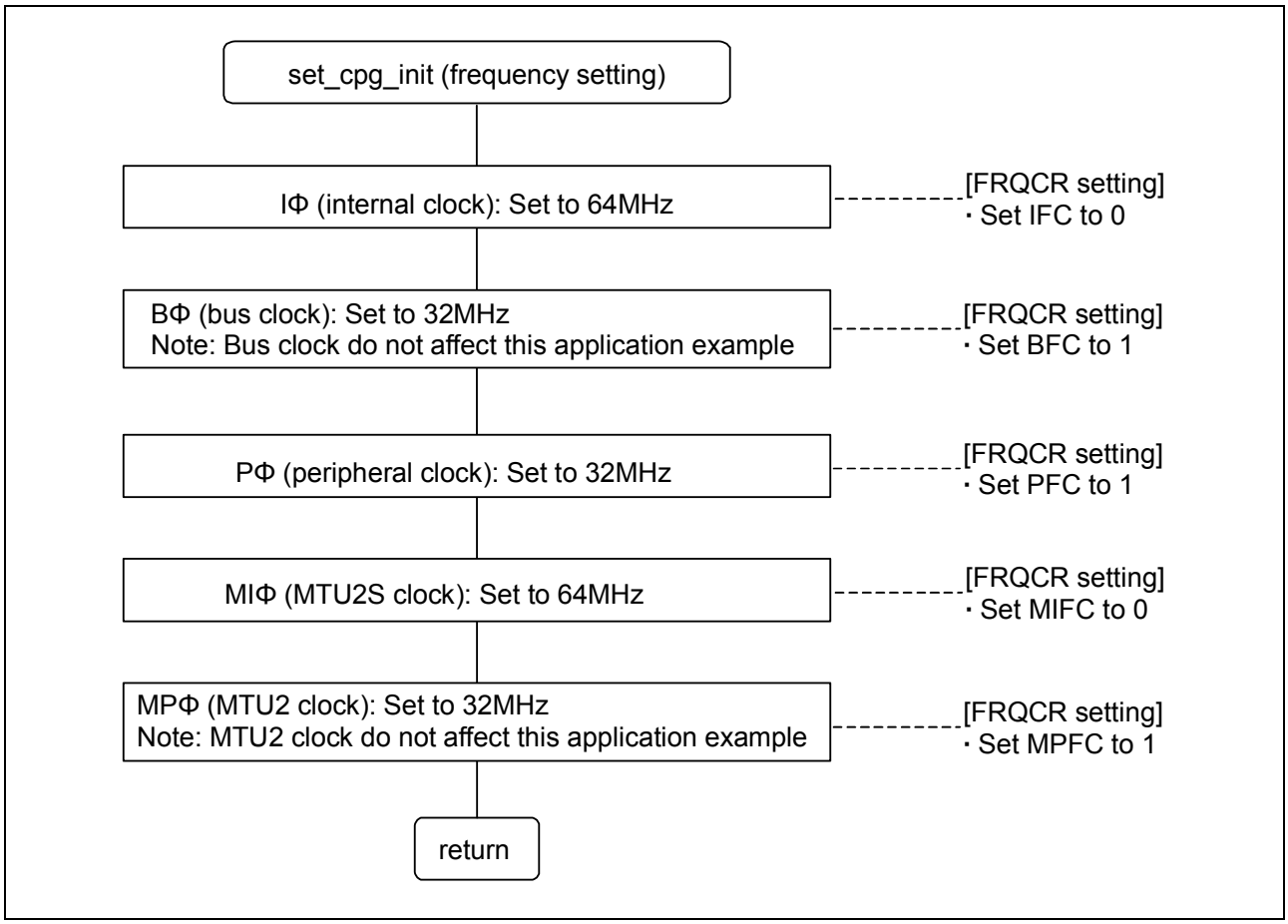

Figure 17 Frequency Setting Process Flow

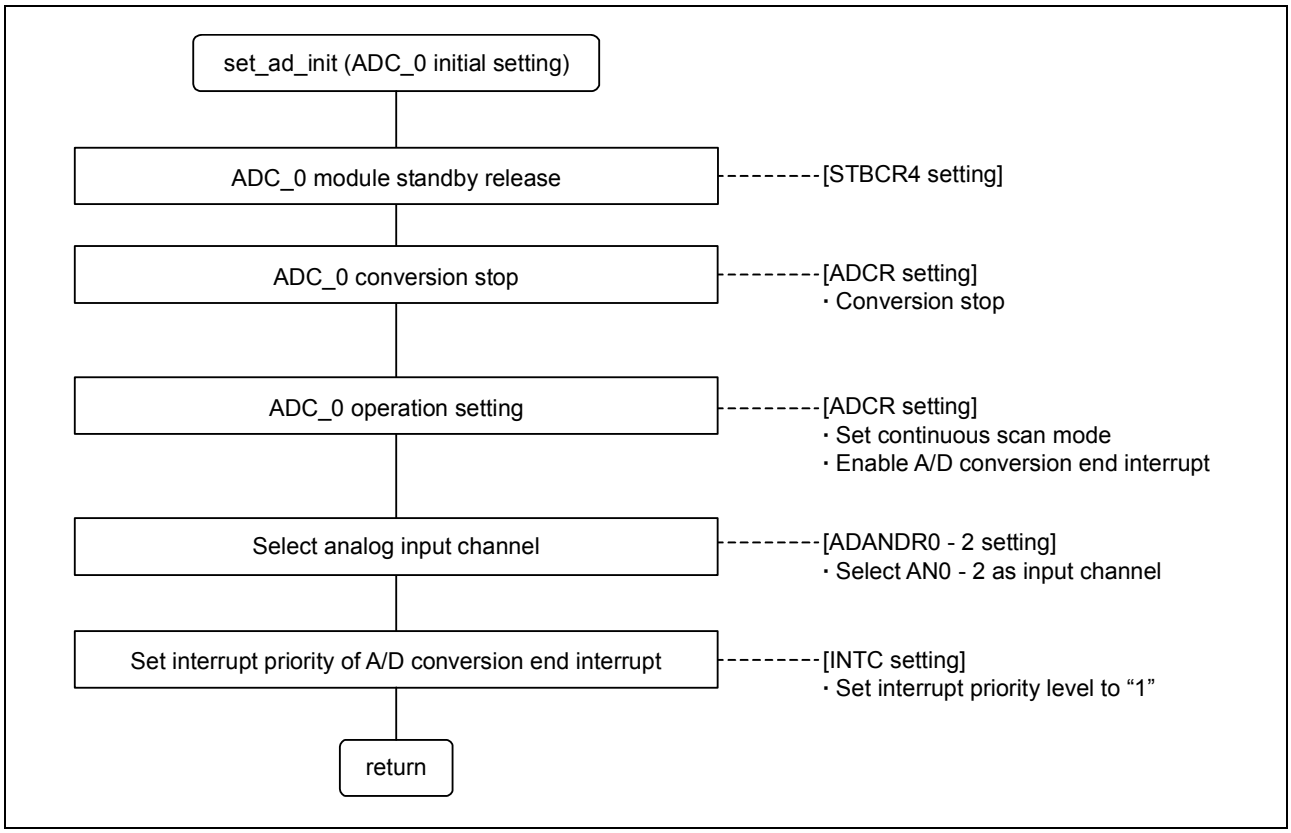

Figure 18 ADC_0 Initial Setting Flow

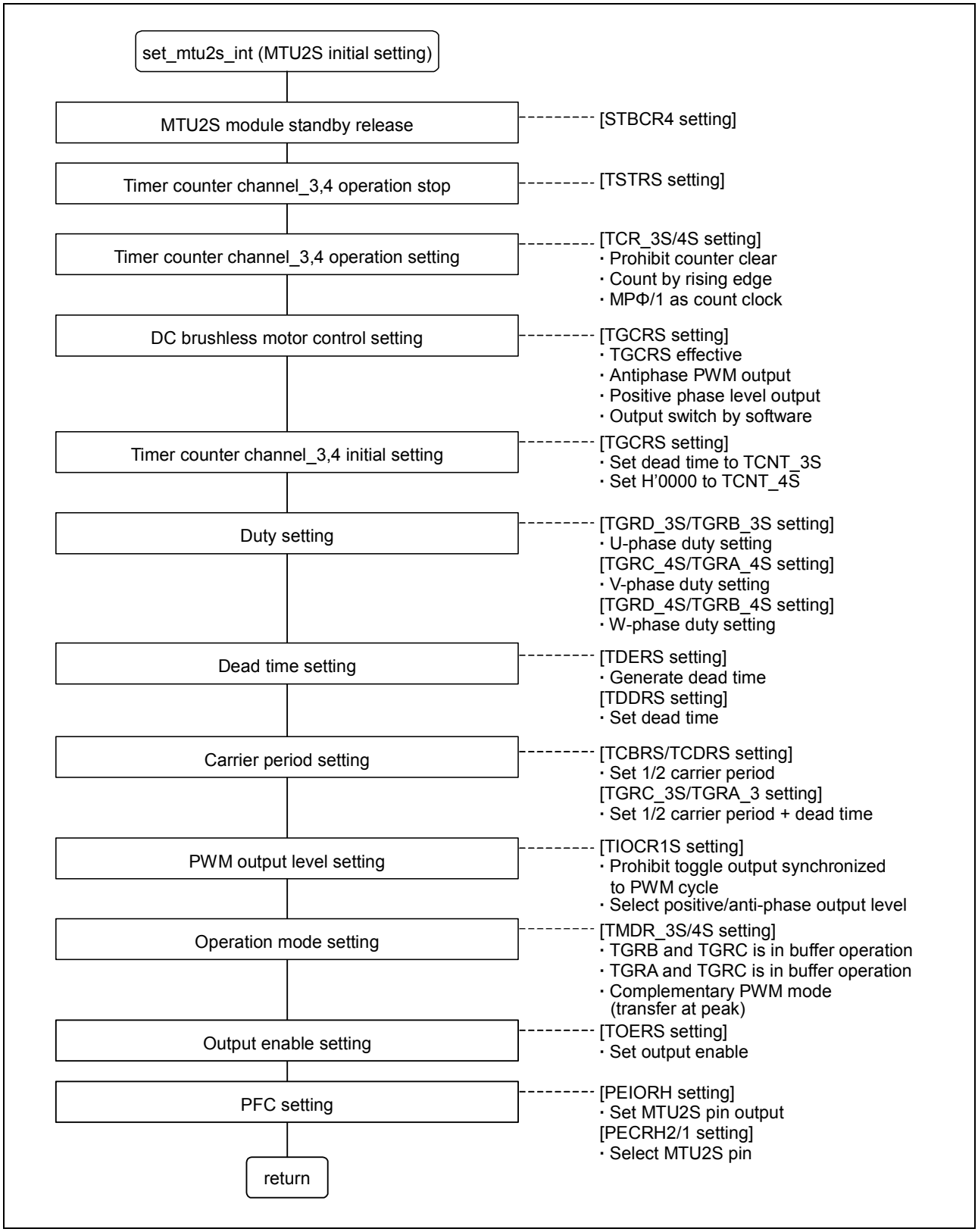

Figure 19 MTU2S Initial Setting Flow

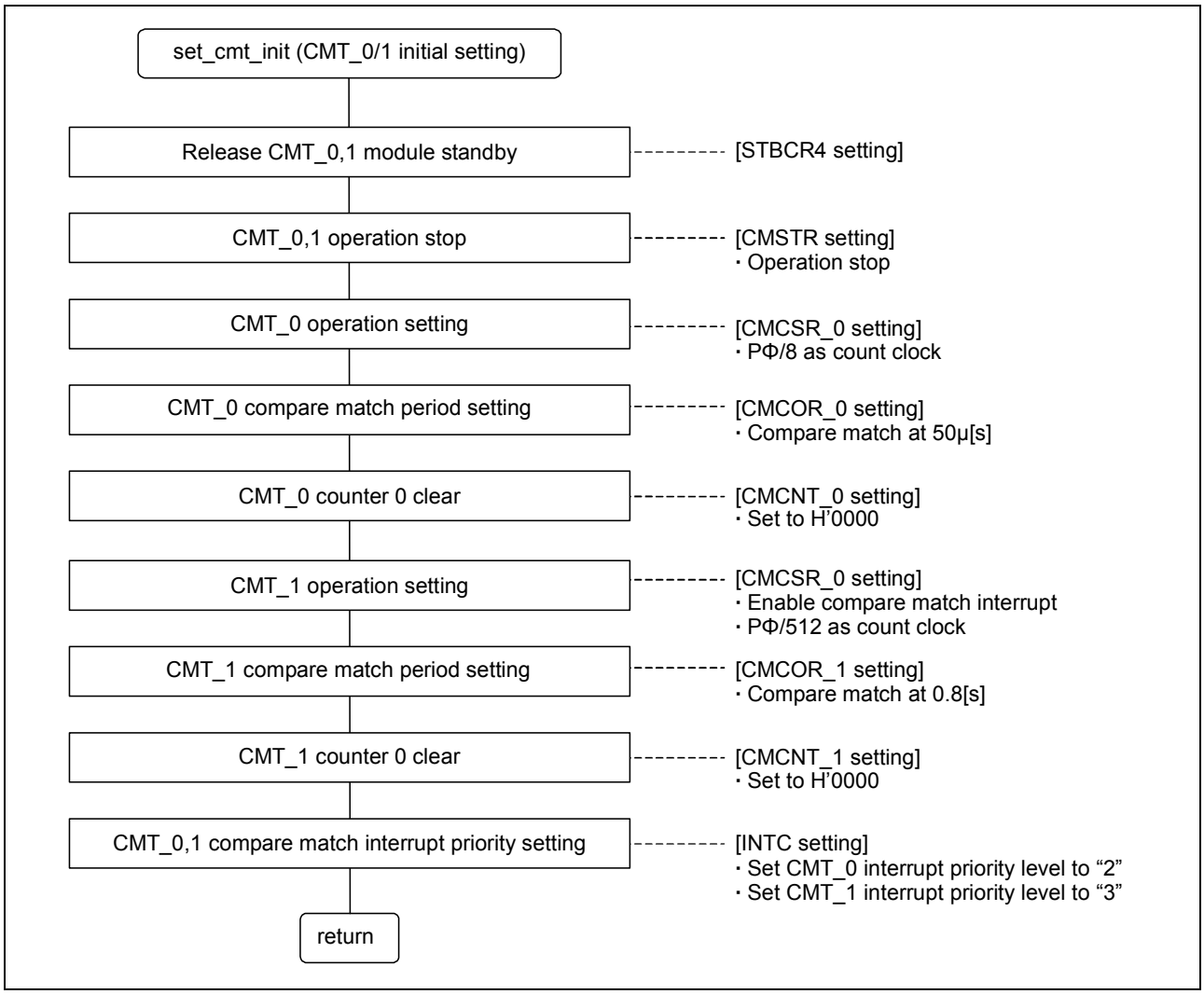

Figure 20 CMT Initial Setting Flow

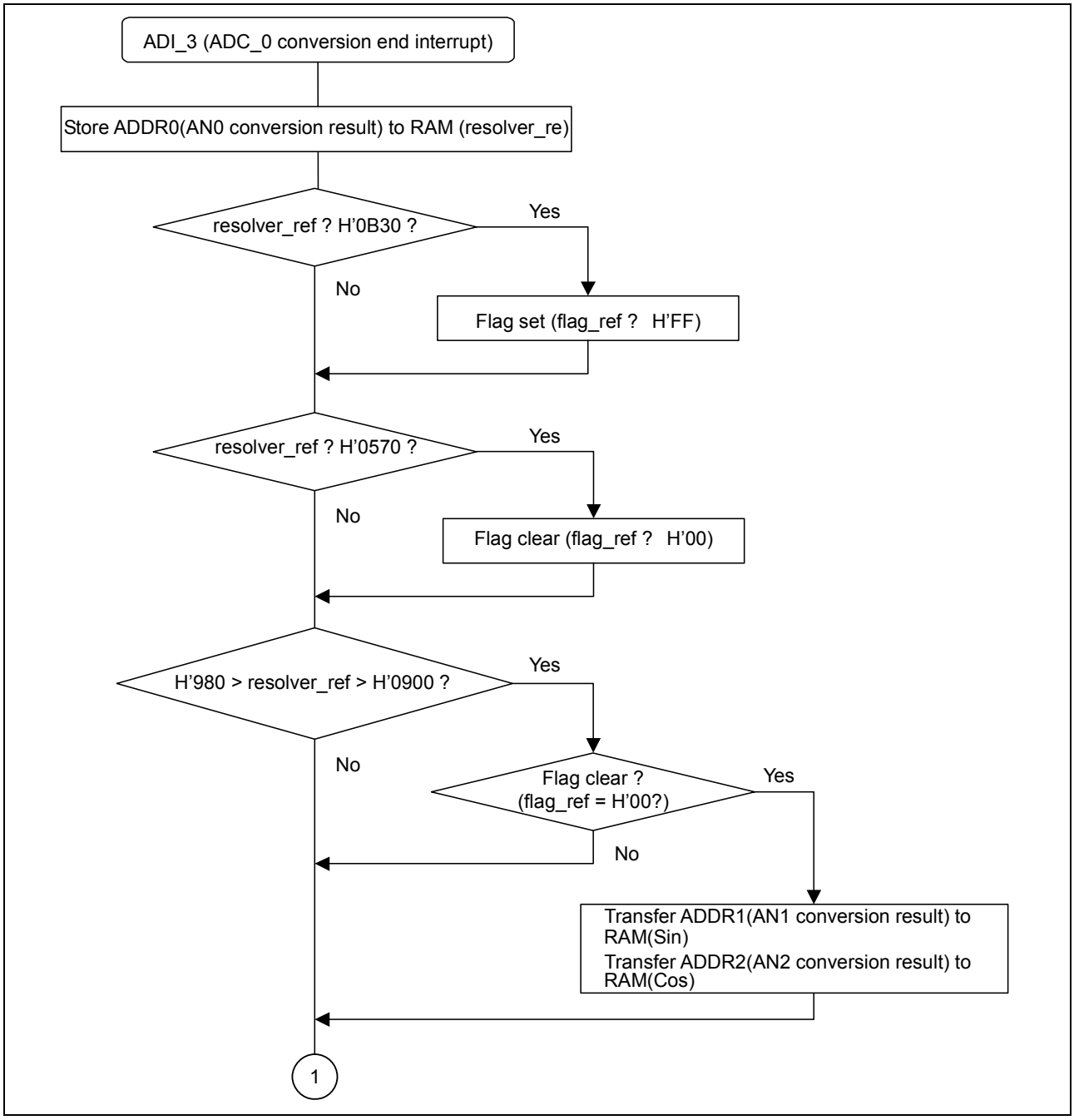

Figure 21 ADC_0 Interrupt Process Flow (1)

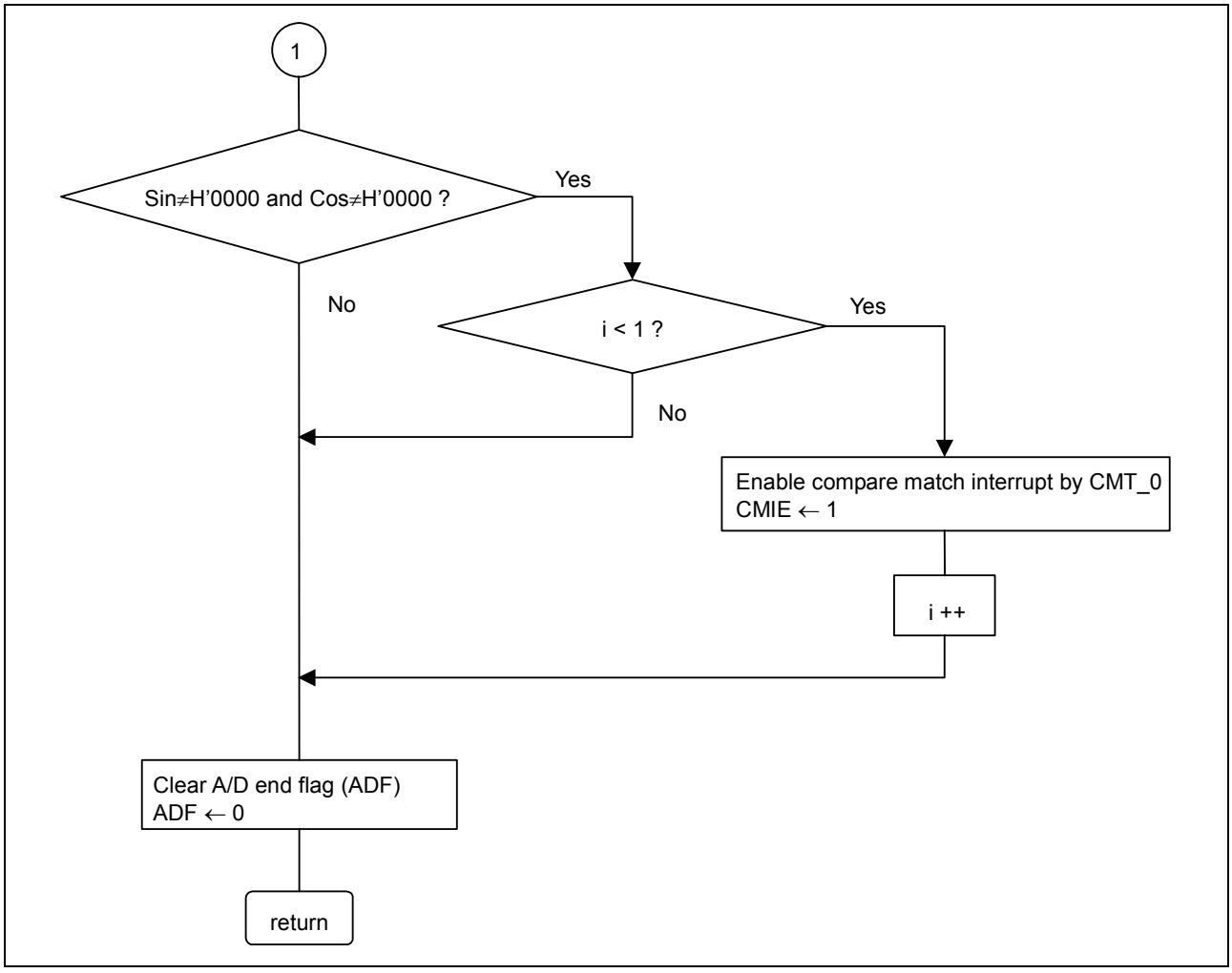

Figure 22 ADC_0 Interrupt Process Flow (2)

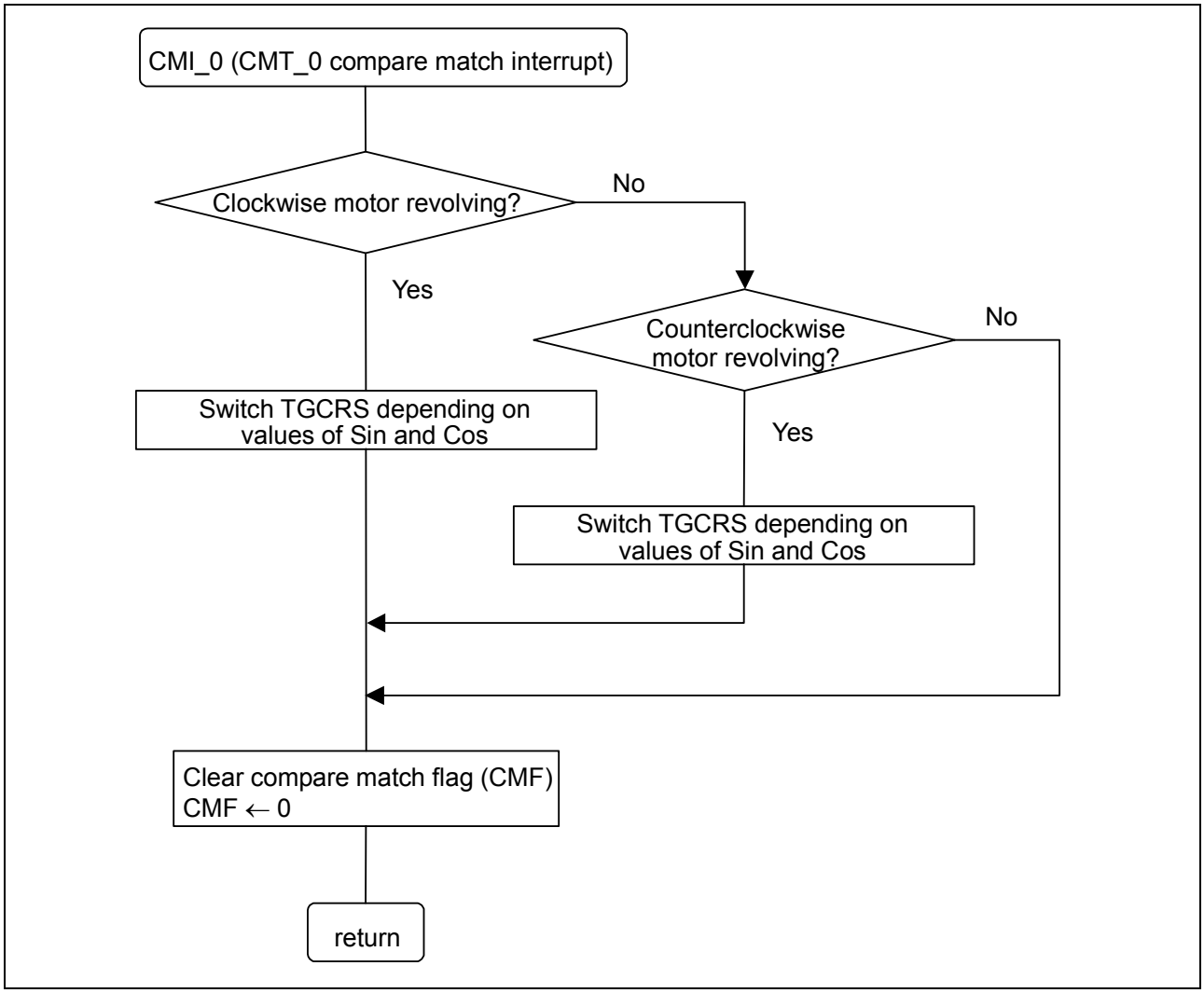

Figure 23 CMT_0 Interrupt Process Flow

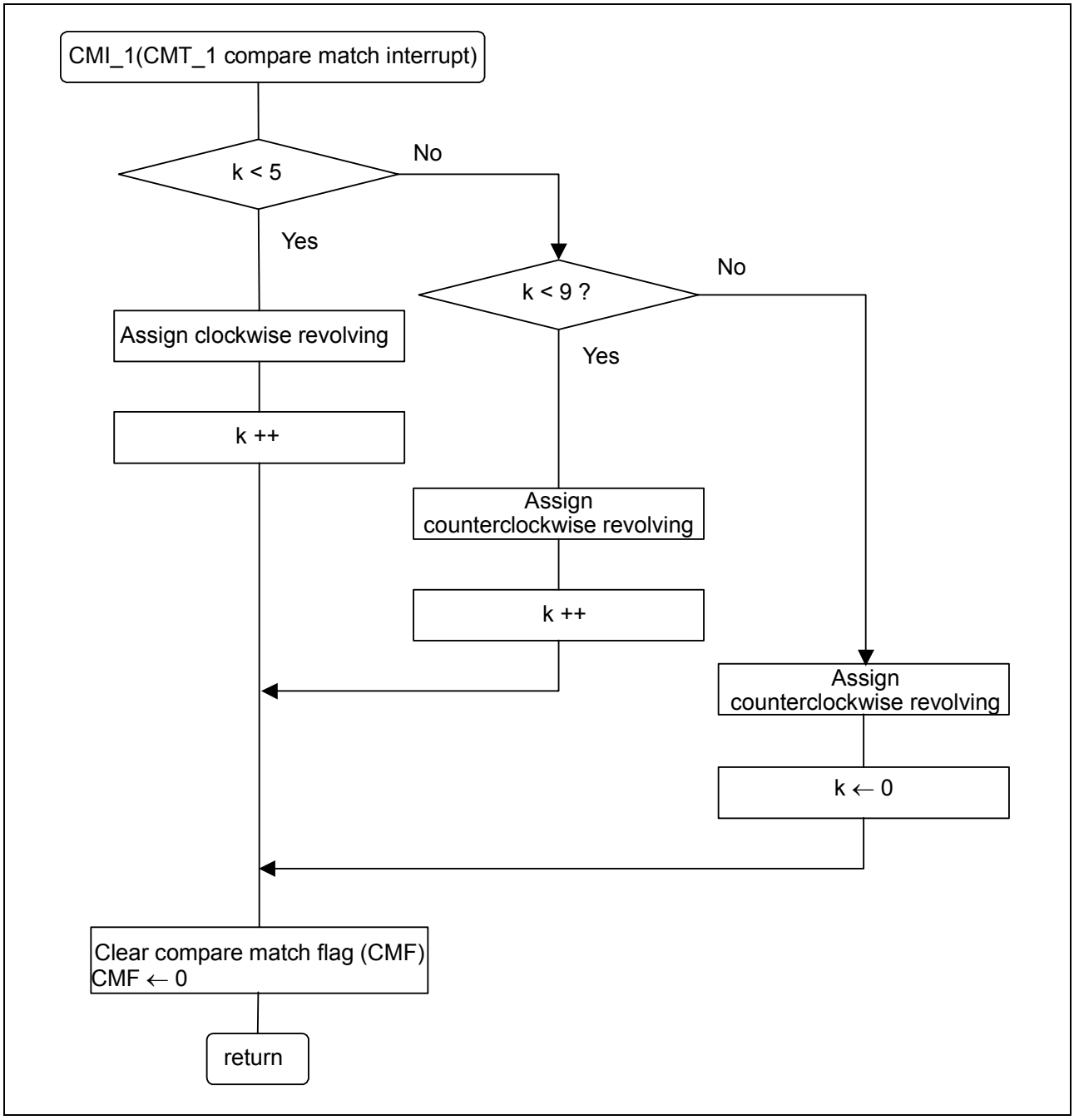

Figure 24 CMT_1 Interrupt Process Flow

4. Sample program

```
/*****************************************************************************/
/* SH7147 Motor Control                                                      */
/* - Resolver Built-in DC Brushless Motor -                                  */
/* Used Module:                                                              */
/*    ADC/MTU2S/CMT:                                                         */
/*****************************************************************************/

/*------------ Include File-----------------------------------------------*/
#include <machine.h>
#include"iodefine.h"

/*****************************************************************************/
/*    Function Declaration                                    */
/*****************************************************************************/
void main(void);               /* Main Function */
void set_cpg_init(void);       /* Each Clock Setting */
void initial_value(void);      /* Initializing Variable */
void set_ad_init(void);        /* ADC Initial Setting */
void set_cmt_init(void);       /* CMT  Initial Setting */
void set_mtu2s_init(void);     /* MTU2S  Initial Setting */

/*****************************************************************************/
/*    Variable Declaration                                    */
/*****************************************************************************/
/* Used in ADC Interrupt Routine */
unsigned char flag_ref;       /* Resolver Excitation Signal Slope Determination Flag */
unsigned short resolver_ref;  /* Result of A/D Converting Resolver Excitation Signal  */
unsigned short Sin;           /* Result of A/D Converting Resolver Output Sin Wave  */
unsigned short Cos;           /* Result of A/D Converting Resolver Output Cos Wave  */
unsigned char i;              /* CMT_0 Interrupt Enable Flag */

/* Used in CMT_0 Interrupt Routine */
unsigned char flag_rot;       /*  Motor Revolving Direction Assigning  Flag */
```

```
/* Used in CMT_1 Interrupt Routine */
unsigned char k;            /* CMT_1 Interrupting Times Counting Flag */

/************************************************************************/
/*    Symbol Declaration                                                */
/************************************************************************/
/* Angle and A/D Conversion Result of Resolver Output Sin Wave */
#define Sin_0  0x0332         /* Result of A/D Converting Sin0    */
#define Sin_60  0x0522         /* Result of A/D Converting Sin60°  */
#define Sin_180 0x0332         /* Result of A/D Converting Sin180°  */
#define Sin_240 0x0142         /* Result of A/D Converting Sin240°  */
/* Angle and A/D Conversion Result of Resolver Output Cos Wave */
#define Cos_90  0x0332         /* Result of A/D Converting Cos90°  */
#define Cos_150 0x0142         /* Result of A/D Converting Cos150°  */
#define Cos_270 0x0332         /* Result of A/D Converting Cos270°  */
#define Cos_330 0x0522         /* Result of A/D Converting Cos330°  */
/* Revolving Direction */
#define CW  1                  /* Clockwise of motor */
#define CCW 2                 /* Counterclockwise of motor */

/* PWM Carrier Period/Dead Time */
#define CARRIER    0x0c80         /* PWM Carrier Period: 50[µs] */
#define HALF_CARRIER   0x0640    /* 1/2 of PWM Carrier Period: 25[µs] */
#define DEAD_TIME  0x0001       /* Dead Time: 15.625[ns] */
#define TCNT3_COMP  (DEAD_TIME + HALF_CARRIER)  /* Dead Time + 1/2 of PWM Carrier Period
*/


/************************************************************************/
/*    Main Function                                                     */
/************************************************************************/
void main(void){


   set_imask(15);          /* Interrupt Prohibited */
```

```
    initial_value();          /* Initializing Variable */
    set_cpg_init();           /* Each Clock Setting */
    set_ad_init();           /* ADC  Initial Setting */
    set_mtu2s_init();         /* MTU2S  Initial Setting */
    set_cmt_init();           /* CMT  Initial Setting */
    AD0.ADCR.BIT.ADST = 1;    /* ADC_0 Conversion Start */
    MTU2S.TSTR.BYTE |= 0xC0;  /* MTU2S ch3,4 Count Start */
    CMT.CMSTR.WORD |= 0x03;   /* CMT ch0,1 Count Start */

    set_imask(0);            /* Interrupt Enable */

    while(1);
}

/************************************************************************/
/*    Initializing Variable                                             */
/************************************************************************/
void initial_value(void){

    /* Used in ADC Interrupt Routine */
    flag_ref = 0;           /* Resolver Excitation Signal Slope Flag */
    i = 0;                  /* CMT_0 Interrupt Enable Flag */
    resolver_ref = 0x00;    /* Result of A/D Converting Resolver Excitation Signal  */
    Sin = 0x0000;          /* Result of A/D Converting Resolver Output Sin Wave  */
    Cos = 0x0000;          /* Result of A/D Converting Resolver Output Cos Wave  */

    /* Used in CMT_0 Interrupt Routine */
    flag_rot = 0;         /*  Motor Revolving Direction Assigning  Flag */

    /* Used in CMT_1 Interrupt Routine */
    k = 0;              /* CMT_1 Interrupting Times Counting Flag */
}

/************************************************************************/
/*    Each Clock Setting                                                */
/*    IΦ:BΦ:PΦ:MIΦ:MPΦ = 64MHz:32MHz:32MHz:64MHz:32MHz                  */
```

```
/************************************************************************/
void set_cpg_init(void){

   CPG.FRQCR.BIT.IFC = 0;     /* IΦ=64MHz */
   CPG.FRQCR.BIT.BFC = 1;     /* BΦ=32MHz */
   CPG.FRQCR.BIT._PFC = 1;    /* PΦ=32MHz */
   CPG.FRQCR.BIT.MIFC = 0;    /* MIΦ=64MHz */
   CPG.FRQCR.BIT.MPFC = 1;    /* MPΦ=32MHz */
}

/************************************************************************/
/*    ADC  Initial Setting                                              */
/************************************************************************/
void set_ad_init(void){

   STB.CR4.BIT._AD0 = 0;     /* 0: AD_0 Operating   1: AD_0 Halting */

   AD0.ADCR.BIT.ADST = 0;     /* 0: Conversion Stop   1: Conversion Start */
   AD0.ADCR.BIT.ADCS = 1;     /* 0: 1 Cycle Conversion   1: Continuous Conversion */
   AD0.ADCR.BIT.ADIE = 1;     /* 0: Interrupt Prohibited 1: Interrupt Prohibited */



   /* Select AN0 - 2 */
   AD0.ADANSR.BIT.ANS2 = 1;   /* 0: Not Selected     1: Selected */
   AD0.ADANSR.BIT.ANS1 = 1;   /* 0: Not Selected     1: Selected */
   AD0.ADANSR.BIT.ANS0 = 1;   /* 0: Not Selected     1: Selected */

   INTC.IPRK.BIT._AD0 = 1;    /* Interrupt Priority Level Setting */

}

/************************************************************************/
/*    MTU2S  Initial Setting                                            */
/************************************************************************/
void set_mtu2s_init(void){
```

```
    STB.CR4.BIT._MTU2S = 0;    /* 0:MTU2S Operating    1:MTU2S Halting */

    MTU2S.TSTR.BIT.CST3 = 0;   /* 0:CH3 Count Stop     1:CH3 Count Start */
    MTU2S.TSTR.BIT.CST4 = 0;   /* 0:CH4 Count Stop     1:CH4 Count Start */

    MTU2S3.TCR.BIT.CCLR = 0;   /* 0,4: Clear Prohibited */
                               /*  1 : Cleared by TGRA */
                               /*  2 : Cleared by TGRB */
                               /*  5 : Cleared by TGRC */
                               /*  6 : Cleared by TGRD */
                               /* 3,7: Cleared by Other Synchronizing Channel */
    MTU2S3.TCR.BIT.CKEG = 0;   /* 0:↑, 1:↓, 2,3:↓↑ */
    MTU2S3.TCR.BIT.TPSC = 0;   /* 0:MP Φ/1, 1:MP Φ/4, 2:MP Φ/16, 3:MP Φ/64,*/
                               /* 4:MP Φ/256, 5:MP Φ/1024,6:TCLKA, 7:TCLKB */

    MTU2S4.TCR.BIT.CCLR = 0;   /* 0,4: Clear Prohibited */
                               /*  1 : Cleared by TGRA */
                               /*  2 : Cleared by TGRB */
                               /*  5 : Cleared by TGRC */
                               /*  6 : Cleared by TGRD */
                               /* 3,7: Cleared by Other Synchronizing Channel */
    MTU2S4.TCR.BIT.CKEG = 0;   /* 0: ↑, 1:↓, 2,3:↓↑ */
    MTU2S4.TCR.BIT.TPSC = 0;   /* 0:MPΦ/1, 1:MPΦ/4, 2:MPΦ/16, 3:MPΦ/64, */
                               /* 4:MPΦ/256, 5:MPΦ/1024,/* 6:TCLKA, 7:TCLKB */

    MTU2S.TGCR.BIT.BDC = 1;    /* Register Setting 0: Not Effective (normal output) 1:
Effective */
    MTU2S.TGCR.BIT.N = 1;      /* Antiphase Output 0:level        1:PWM */
    MTU2S.TGCR.BIT.P = 0;      /* Positive Output  0:level        1:PWM */
    MTU2S.TGCR.BIT.FB = 1;     /* Output Switch Setting */
                       /* 0: Input Capture TGRA0, TGRB0, and TGRC0 */
                       /* 1: Software Control UF, VF, and WF bit */
    MTU2S.TGCR.BIT.WF = 0;     /* 0   0   0   0   1   1   1   1 */
    MTU2S.TGCR.BIT.VF = 0;     /* 0   0   1   1   0   0   1   1 */
    MTU2S.TGCR.BIT.UF = 0;     /* 0   1   0   1   0   1   0   1 */
```

```
            /*TIOC3B(U-phase)  OFF   ON  OFF  OFF  OFF   ON  OFF  OFF */
            /*TIOC4A(V-phase)  OFF  OFF   ON   ON  OFF  OFF  OFF  OFF */
            /*TIOC4B(W -phase) OFF  OFF  OFF  OFF   ON  OFF   ON  OFF */
            /*TIOC3D(U -phase) OFF  OFF   ON  OFF  OFF  OFF   ON  OFF */
            /*TIOC4C(V -phase) OFF  OFF  OFF  OFF   ON   ON  OFF  OFF */
            /*TIOC4D(W -phase) OFF   ON  OFF   ON  OFF  OFF  OFF  OFF */

   MTU2S3.TCNT = DEAD_TIME;  /* Dead Time Setting */
   MTU2S4.TCNT = 0x0000;     /* 0 Clear */

   MTU2S3.TGRB = 0x00A0;     /* Set PWM Duty to 10% */
   MTU2S4.TGRA = 0x00A0;     /* Set PWM Duty to 10% */
   MTU2S4.TGRB = 0x00A0;     /* Set PWM Duty to 10% */

   MTU2S3.TGRD = 0x00A0;     /* Set PWM Duty to 10% (Buffer register) */
   MTU2S4.TGRC = 0x00A0;     /* Set PWM Duty to 10% (Buffer register) */
   MTU2S4.TGRD = 0x00A0;     /* Set PWM Duty to 10% (Buffer register) */

   MTU2S.TDER.BIT.TDER = 1;   /* 0: Dead Time Generation Prohibited  1: Dead Time
Generation */
   MTU2S.TDDR = DEAD_TIME;    /* Dead Time Setting */

   MTU2S.TCDR = HALF_CARRIER; /* 1/2 of Carrier Period */
   MTU2S.TCBR = HALF_CARRIER; /* 1/2 of Carrier Period */

   MTU2S3.TGRA = TCNT3_COMP;  /* 1/2 of Carrier Period + Dead Time */
   MTU2S3.TGRC = TCNT3_COMP;  /* 1/2 of Carrier Period + Dead Time */

   MTU2S.TOCR1.BIT.PSYE = 0;  /* 0: Toggle Output Prohibited  1: Toggle Output Enable
*/
   MTU2S.TOCR1.BIT.TOCL = 0;  /* 0: Enable Write to TOCS, OLSN, and OLSP bit */
                         /* 1: Prohibit Write to TOCS, OLSN, and OLSP bit */
   MTU2S.TOCR1.BIT.TOCS = 0;  /* 0: TOCR1 Effective   1: TOCR2 Effective */
   MTU2S.TOCR1.BIT.OLSN = 0;  /*            0     1 */
      /* Initial Output                   H     L */
      /* Active Level                     L     H */
```

```
      /* Compare Match Output (when up-count)   H     L */
      /* Compare Match Output (when down-count) L     H */

   MTU2S.TOCR1.BIT.OLSP = 0;  /*             0     1 */
      /* Initial Output                      H     L */
      /* Active Level                        L     H */
      /* Compare Match Output (when up-count)   L     H */
      /* Compare Match Output (when down-count) H     L */

   MTU2S3.TMDR.BIT.BFB = 1;   /* 0:TGRB_3S is in normal operation, 1:TGRB_3S is in
buffer operation */
   MTU2S3.TMDR.BIT.BFA = 1;   /* 0:TGRA_3S is in normal operation, 1:TGRA_3S is in
buffer operation */
   MTU2S3.TMDR.BIT.MD = 13;   /* 0: Normal Operation,        1: Setting Prohibited, */
                      /* 2: PWM Mode 1        3: Setting Prohibited, */
                      /* 4: Setting Prohibited,        5: Setting Prohibited, */
                      /* 6: Setting Prohibited,        7: Setting Prohibited, */
                      /* 8: Reset Synchronized PWM,     9: Setting Prohibited, */
                      /* 10: Setting Prohibited,       11: Setting Prohibited, */
                      /* 12: Setting Prohibited,       13:Complementary PWM(transfer at
peak) */
                      /* 14:Complementary PWM(transfer at valley)
                         15:Complementary PWM(transfer at peak/valley) */

   MTU2S4.TMDR.BIT.BFB = 1;   /* 0:TGRB_4S is in normal operation, 1:TGRB_4S is in
buffer operation */
   MTU2S4.TMDR.BIT.BFA = 1;   /* 0:TGRA_4S is in normal operation, 1:TGRA_4S is in
buffer operation */
   MTU2S4.TMDR.BIT.MD = 0;    /* 0: Normal Operation,        1: Setting Prohibited, */
                      /* 2: PWM Mode 1,       3: Setting Prohibited, */
                      /* 4: Setting Prohibited,        5: Setting Prohibited, */
                      /* 6: Setting Prohibited,        7: Setting Prohibited, */
                      /* 8: Reset Synchronized PWM      9: Setting Prohibited, */
                      /* 10: Setting Prohibited,       11: Setting Prohibited, */
                      /* 12: Setting Prohibited,       13:Complementary PWM(transfer at
peak) */
```

```
                    /* 14:Complementary PWM(transfer at valley)
                       15:Complementary PWM(transfer at peak/valley) */

  MTU2S.TOER.BIT.OE4D = 1;  /* 0:TIOC4DS  Output Prohibited   1:TIOC4DS  Output Enable */
  MTU2S.TOER.BIT.OE4C = 1;  /* 0:TIOC4CS  Output Prohibited   1:TIOC4CS  Output Enable */
  MTU2S.TOER.BIT.OE3D = 1;  /* 0:TIOC3DS  Output Prohibited   1:TIOC3DS  Output Enable */
  MTU2S.TOER.BIT.OE4B = 1;  /* 0:TIOC4BS  Output Prohibited   1:TIOC4BS  Output Enable */
  MTU2S.TOER.BIT.OE4A = 1;  /* 0:TIOC4AS  Output Prohibited   1:TIOC4AS  Output Enable */
  MTU2S.TOER.BIT.OE3B = 1;  /* 0:TIOC3BS  Output Prohibited   1:TIOC3BS  Output Enable */


  PFC.PEIORH.WORD = 0x003f;  /* PE21 - PE16 => Output Setting */
  PFC.PECRH2.WORD = 0x0011;  /* P21 => TIOC4DS,  P20 => TIOC4CS */
  PFC.PECRH1.WORD = 0x1111;  /* P19 => TIOC4BS,  P18 => TIOC4AS */
                             /* P17 => TIOC3DS,  P16 => TIOC3BS */

}

/************************************************************************/
/*    CMT Initial Setting                                               */
/************************************************************************/
void set_cmt_init(void){

   STB.CR4.BIT._CMT = 0;     /* 0: CMT Operating    1: CMT Halting */

   CMT.CMSTR.BIT.STR0 = 0;    /* 0: CH0 Count Stop   1: CH0 Count Start */
   CMT.CMSTR.BIT.STR1 = 0;    /* 0: CH1 Count Stop   1: CH1 Count Start */

   CMT0.CMCSR.BIT.CKS = 0;    /* 0:P Φ/8 1:P Φ/32 2:P Φ/128 3:P Φ/512 */
   CMT0.CMCOR = 0x00c8;       /* Compare Match at 50[µs] Cycle */
   CMT0.CMCNT = 0x0000;       /* Count 0 Clear */

   CMT1.CMCSR.BIT.CKS = 3;    /* 0:P Φ/8 1:P Φ/32 2:P Φ/128 3:P Φ/512 */
   CMT1.CMCSR.BIT.CMIE = 1;   /* 0: CMT_1 Interrupt Prohibit   1: CMT_1 Interrupt Enable
*/
   CMT1.CMCOR = 0xc350;       /* Compare Match at 0.8[s] Cycle */
```

```
   CMT1.CMCNT = 0x0000;      /* Counter 0 Clear */

   INTC.IPRJ.BIT._CMT0 = 2;   /* Interrupt Priority Level Setting */
   INTC.IPRJ.BIT._CMT1 = 3;   /* Interrupt Priority Level Setting */

}

/************************************************************************/
/*    AD_0 Interrupt Routine                                            */
/************************************************************************/
#pragma interrupt(ADI_3)
void ADI_3(void){

   /* Store AD Conversion Value to RAM */
   resolver_ref = AD0.ADDR0;     /* Insult AD Conversion Value of AN0( Resolver
Excitation Signal) */

   if(resolver_ref >= 0x0b30){
      flag_ref = 0xff;
   }else if(resolver_ref <= 0x0570){
      flag_ref = 0x00;
   }else if((resolver_ref > 0x0900) && (resolver_ref < 0x0980)){
      if(flag_ref == 0x00){
         Sin = AD0.ADDR1;        /* Insult AD Conversion Value of AN1( Resolver Output
Sin wave )*/
         Cos = AD0.ADDR2;        /* Insult AD Conversion Value of AN2( Resolver Output
Cos wave )*/
      }
   }

   /* CMT_0 Interrupt Enable */
   if(Sin != 0x000 && Cos !=0x000){
      if(i <1){
      CMT0.CMCSR.BIT.CMIE = 1;     /* 0: CMT_0 Interrupt Prohibited 1: CMT_0 Interrupt
Enable */
      i++;
```

```
      }
   }

   AD0.ADSR.BIT.ADF = 0;          /* Clear AD Conversion End Flag */
}


/************************************************************************/
/*    CMT_0  Interrupt Process Routine                                  */
/************************************************************************/
#pragma interrupt(CMI_0,CMI_1)
void CMI_0(void){

   /* Check Motor Position from Resolver Signal for each 30°
     and Switch TGCR Depending to Motor Position */

   if(flag_rot == CW){      /* Clockwise of Motor */
      if(Sin >= Sin_60){
         if(Cos >= Cos_90){
            MTU2S.TGCR.BYTE = 0xeb;
         }else{
            MTU2S.TGCR.BYTE = 0xea;
            }

      }else if(Sin <= Sin_240){
         if(Cos >= Cos_270){
            MTU2S.TGCR.BYTE = 0xea;
         }else{
            MTU2S.TGCR.BYTE = 0xeb;
         }

      }else if(Cos <= Cos_150){
         if(Sin >= Sin_180){
            MTU2S.TGCR.BYTE = 0xec;
         }else{
            MTU2S.TGCR.BYTE = 0xed;
```

```
        }

    }else if(Cos >= Cos_330){
        if(Sin >= Sin_0){
            MTU2S.TGCR.BYTE = 0xed;
        }else{
            MTU2S.TGCR.BYTE = 0xec;
        }

    }else if(Sin >= Sin_0){
        if(Cos >= Cos_90){
            MTU2S.TGCR.BYTE = 0xe9;
        }else{
            MTU2S.TGCR.BYTE = 0xee;
        }

    }else{
        if(Cos >= Cos_270){
            MTU2S.TGCR.BYTE = 0xee;
        }else{
            MTU2S.TGCR.BYTE = 0xe9;
        }
    }
}
else if(flag_rot== CCW){   /* Counterclockwise of Motor */
    if(Sin >= Sin_60){
        if(Cos >= Cos_90){
            MTU2S.TGCR.BYTE = 0xee;
        }else{
            MTU2S.TGCR.BYTE = 0xec;
        }

    }else if(Sin <= Sin_240){
        if(Cos >= Cos_270){
            MTU2S.TGCR.BYTE = 0xec;
        }else{
```

```
            MTU2S.TGCR.BYTE = 0xee;
         }

      }else if(Cos <= Cos_150){
         if(Sin >= Sin_180){
            MTU2S.TGCR.BYTE = 0xe9;
         }else{
            MTU2S.TGCR.BYTE = 0xeb;
         }

      }else if(Cos >= Cos_330){
         if(Sin >= Sin_0){
            MTU2S.TGCR.BYTE = 0xeb;
         }else{
            MTU2S.TGCR.BYTE = 0xe9;
         }

      }else if(Sin >= Sin_0){
         if(Cos >= Cos_90){
            MTU2S.TGCR.BYTE = 0xea;
         }else{
            MTU2S.TGCR.BYTE = 0xed;
         }

      }else{
         if(Cos >= Cos_90){
            MTU2S.TGCR.BYTE = 0xed;
         }else{
            MTU2S.TGCR.BYTE = 0xea;
         }
      }
   }

   CMT0.CMCSR.BIT.CMF = 0;    /* Clear Compare Match Flag */
}
```

```
/************************************************************************/
/*    CMT_1  Interrupt Process Routine                                  */
/************************************************************************/
void CMI_1(void){

   /* Reverse Motor Revolving Direction at each 4[s] */
   if(k < 5){
      flag_rot = CW;       /* Clockwise of Motor */
      k++;
   }
   else if(k < 9){
      flag_rot = CCW;      /* Counterclockwise of Motor */
      k++;
   }
   else{
      flag_rot = CCW;      /* Counterclockwise of Motor */
      k=0;
   }

   CMT1.CMCSR.BIT.CMF = 0;    /* Clear Compare Match Flag */

}
```

5. Reference

Hardware Manual

SH7147 Group Hardware Manual

6. Web-site and contact for support

Renesas Web-site

http://www.renesas.com/

Revision History	Resolver Built-in DC Brushless Motor Control

Rev.	Date	Description	
		Page	Summary
Rev.1.00	2006.10.12	-	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.