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.

SH7046 Group

DC Brushless Motor Control

Contents

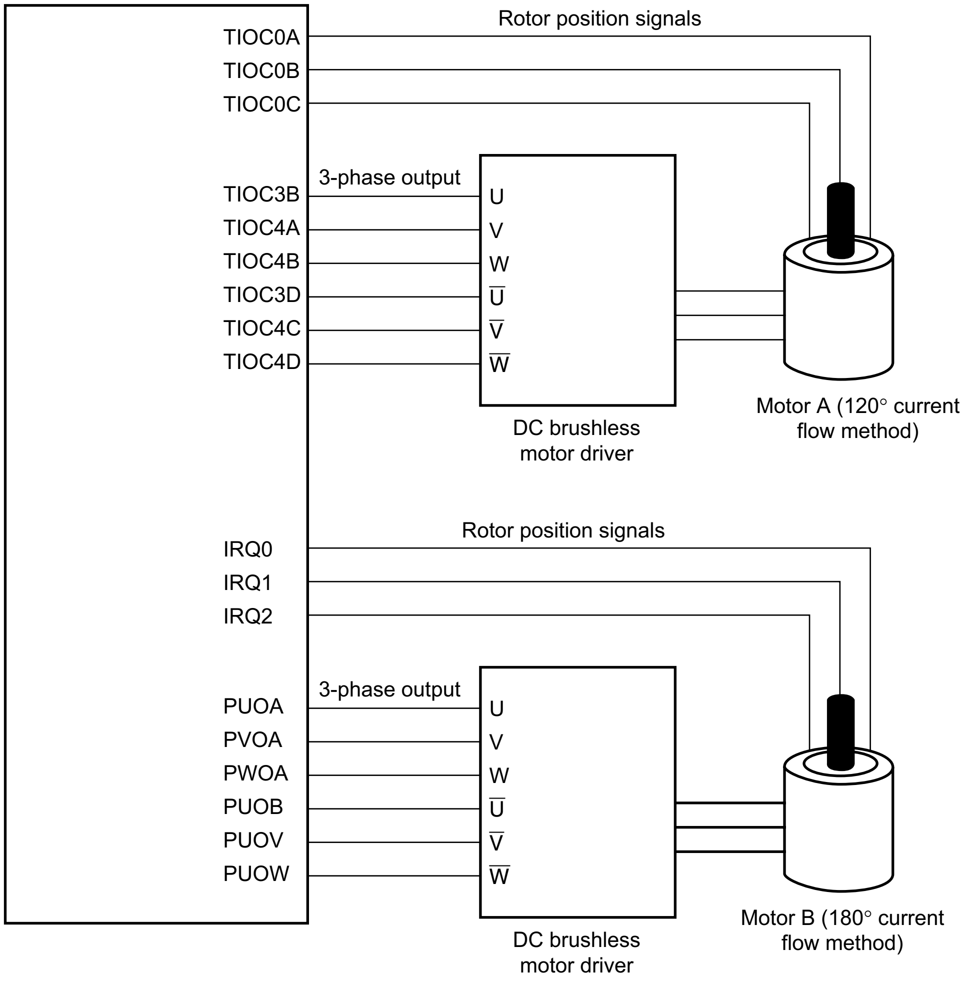
1. Specifications

An SH7046 is used to control DC brushless motors A and B by means of a 120° current flow method and 180° current flow method, as shown in figure 1.

In DC brushless motor control, rotor pole position signals are detected, and a drive waveform for each position signal is output from a timer output pin.

On the 120° current flow method side (motor A), the MTU's reset-synchronized PWM mode is used, and control is performed by means of positive-phase-side level output and negative-phase-side chopping waveform output.

On the 180° current flow method side (motor B), the MMT is used, and control is performed by means of chopping waveform output for both the positive phase and negative phase.



Legend:

U:	U phase	TIOC3D:	MTU/PWM U-phase output (negative phase)
V:	V phase	TIOC4C:	MTU/PWM V-phase output (negative phase)
W:	W phase	TIOC4D:	MTU/PWM W-phase output (negative phase)
$\overline{U}$:	U phase negative phase	IRQ0:	IRQ external interrupt request input pin 0
$\overline{V}$:	V phase negative phase	IRQ1:	IRQ external interrupt request input pin 1
$\overline{W}$:	W phase negative phase	IRQ2:	IRQ external interrupt request input pin 2
TIOC0A:	MTU/ch0 input capture A pin	PUOA:	MTU/PWM U-phase output (positive phase)
TIOC0B:	MTU/ch0 input capture B pin	PVOA:	MTU/PWM V-phase output (positive phase)
TIOC0C:	MTU/ch0 input capture C pin	PWOA:	MTU/PWM W-phase output (positive phase)
TIOC3B:	MTU/PWM U-phase output (positive phase)	PUOB:	MTU/PWM U-phase output (negative phase)
TIOC4A:	MTU/PWM V-phase output (positive phase)	PVOB:	MTU/PWM V-phase output (negative phase)
TIOC4B:	MTU/PWM W-phase output (positive phase)	PWOB:	MTU/PWM W-phase output (negative phase)

Figure 1 DC Brushless Motor Control

2. Design Concept

(1) Motor A (120° current flow method)

The MTU's reset-synchronized PWM mode is used, and positive-phase-side level output and negative-phase-side chopping output 3-phase PWM waveforms are generated, and output from MTU output pins.

In initial control, MTU/ch0 compare match interrupts are used, and the excitation phase is switched by software at fixed intervals.

After the end of initial control, a transition is made to timer output switching control by means of external input, and excitation phase switching is performed automatically by capturing rotor pole position signals output from the motor from MTU/ch0 input capture pins, and generating input capture interrupts at input signal edges.

(2) Motor B (180° current flow method)

The MMT is used, and chopping output 3-phase PWM waveforms are generated for both the positive-phase-side and negative-phase-side, and output from MMT output pins.

The dead time in positive-phase/negative-phase output on/off operations is set to 50 µs using a software counter.

In initial control, MTU/ch1 compare match interrupts are used, and the excitation phase is switched by software at fixed intervals.

After the end of initial control, a transition is made to timer output switching control by means of external input, and excitation phase switching is performed automatically by capturing rotor pole position signals output from the motor from IRQ pins, and generating IRQ interrupts at input signal edges.

3. Functions Used

Control of two DC brushless motors is performed by assigning the MTU, MMT, and IRQ functions of the SH7046, as shown in figure 2.

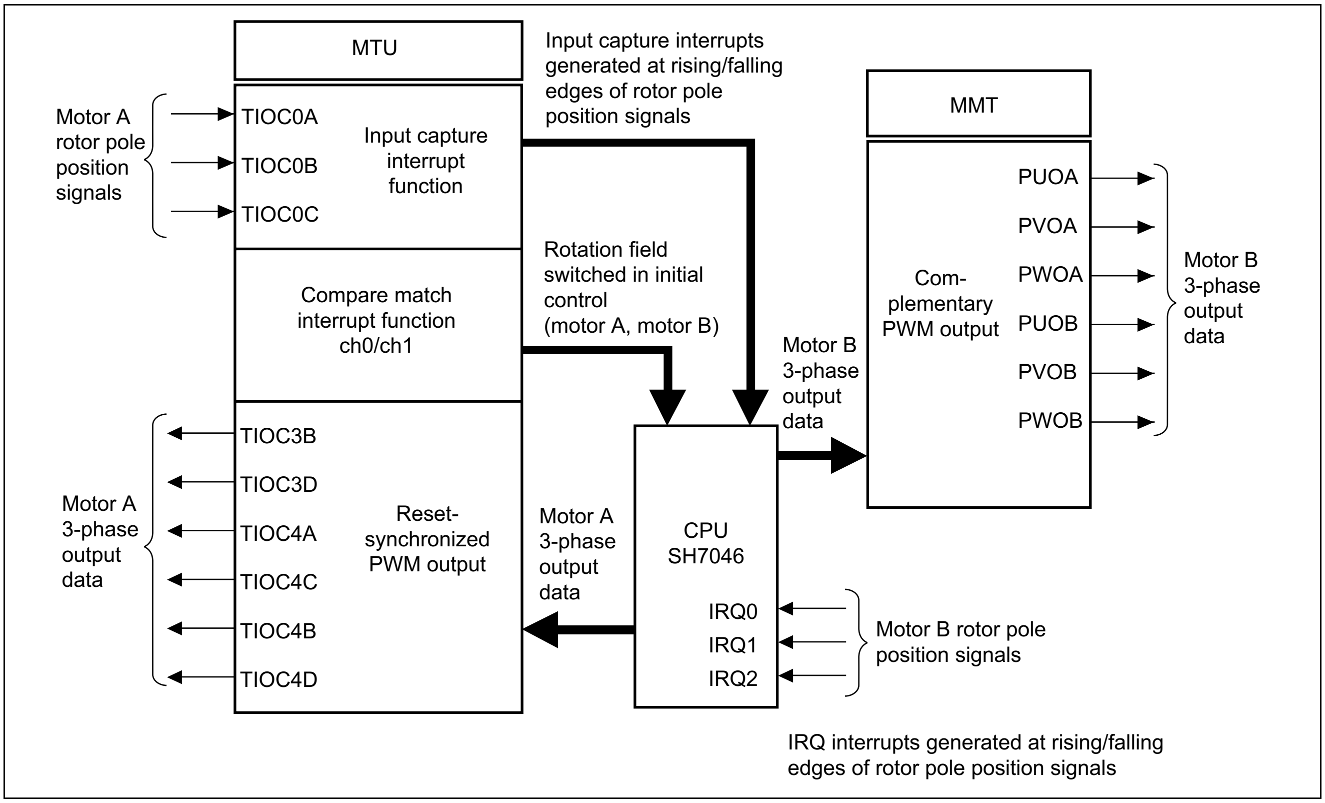

MTU
Motor A rotor pole position signals
TIOC0A
TIOC0B
TIOC0C
Input capture interrupt function
Input capture interrupts generated at rising/falling edges of rotor pole position signals
MMT
Compare match interrupt function ch0/ch1
Rotation field switched in initial control (motor A, motor B)
Com-plementary PWM output
PUOA
PVOA
PWOA
PUOB
PVOB
PWOB
Motor B 3-phase output data
Motor B 3-phase output data
Motor A 3-phase output data
TIOC3B
TIOC3D
TIOC4A
TIOC4C
TIOC4B
TIOC4D
Reset-synchronized PWM output
Motor A 3-phase output data
CPU SH7046
IRQ0
IRQ1
IRQ2
Motor B rotor pole position signals
IRQ interrupts generated at rising/falling edges of rotor pole position signals

Figure 2 Block Diagram of DC Brushless Motor Control

SH7046 function assignments are described below.

- MTU compare match interrupt function: In motor A and motor B initial control, this function is used to request a CPU interrupt every fixed rotation field switching period (time corresponding to a 60° portion of the motor frequency) until the motor rotates once or more and a rotor pole position signal is detected.
- MTU input capture interrupt function: After the end of motor A initial control, this function is used to detect rising/falling edges of rotor pole position signals from motor A, and issue interrupt requests to the CPU.
- MTU reset-synchronized PWM waveform output function: This function is used to generate motor A positive-phase-side level output and negative-phase-side chopping output waveforms.
- MMT complementary PWM waveform output function: This function is used to generate chopping output waveforms for motor B positive-phase-side and negative-phase-side control.
- IRQ external interrupt function: After the end of motor B initial control, this function is used to detect rising/falling edges of rotor pole position signals, and issue interrupt requests to the CPU.

Each function is described below.

1. The MTU is used in reset-synchronized PWM mode to perform a compare match interrupt function for measuring the rotation field switching period in motor A and motor B initial control, and an input capture interrupt function that, after the end of motor A initial control, generates an interrupt on detection of a rotor pole position signal from the motor and performs excitation phase switching, and to execute motor A control by performing positive-phase-side level output and negative-phase-side chopping output.
   Functions used in common by the MTU's compare match interrupt function and input capture interrupt function are described below.
   - The system clock (Pϕ) is the reference clock for operating the CPU and peripheral functions. The system clock is scaled to a frequency of ϕ/2 to ϕ/8192 by a prescaler, and supplied to the respective peripheral modules.
   - A timer counter (TCNT) is a 16-bit readable/writable counter. Its input clock is set by means of TCR.
   - A timer control register (TCR) is an 8-bit readable/writable register that selects the TCNT input clock and clearing source.
   - A timer status register (TSR) is an 8-bit readable/writable register that performs control of interrupt request signals.
   - A timer interrupt enable register (TIER) is an 8-bit readable/writable register that controls enabling/disabling of interrupt requests. In this sample task, the ch0 TGFD interrupt request (TGID) is enabled in motor A initial control, and the ch1 TGFA interrupt request (TGIA) is enabled in motor B initial control.
   - A timer I/O control register (TIOR) is an 8-bit readable/writable register that is used for TGRA, TGRB, TGRC, and TGRD function selection, and input capture input edge selection.
   - Functions used in reset-synchronized PWM mode are described below.
   - By combining MTU/ch3 and ch4, a 3-phase PWM waveform output is performed in which one waveform transition point is common.
   - The ch3 counter clock is set with timer control register_3 (TCR_3), and a TGRA compare match is set as the counter clearing source.
   - The timer gate control register is an 8-bit readable/writable register that is used for positive-phase and negative-phase output waveform selection and to set the feedback signal input source.
   - Timer general register_3 (TGRA_3) is a 16-bit readable/writable register that sets the PWM output period. Duty cycles are set in TGRB_3, TGRA_4, and TGRB_4.
   - Timer mode register_3 (TMDR_3) is an 8-bit readable/writable register that sets the operating mode of each channel. In this sample task, reset-synchronized PWM mode is set.
   - The timer output master enable register (TOER) is an 8-bit readable/writable register that enables PWM waveform output.

2. The MTU input capture function is used to detect rising/falling edges of motor A rotor pole position signals, and generate interrupt requests. Figure 3 shows a block diagram of rotor pole position signal input edge triggered interrupt requests generated by means of the input capture function.
   The block diagram is described below.
   - Input capture input pins (TIOC0A, TIOC0B, TIOC0C) function as motor A rotor pole position signal detection signal input pins.
   - The input capture registers (TGRA_0/B_0/C_0) are 16-bit readable/writable registers. When an input edge of an input capture input signal is detected, the TCNT value at that time is transferred, and an interrupt request is issued to the CPU.

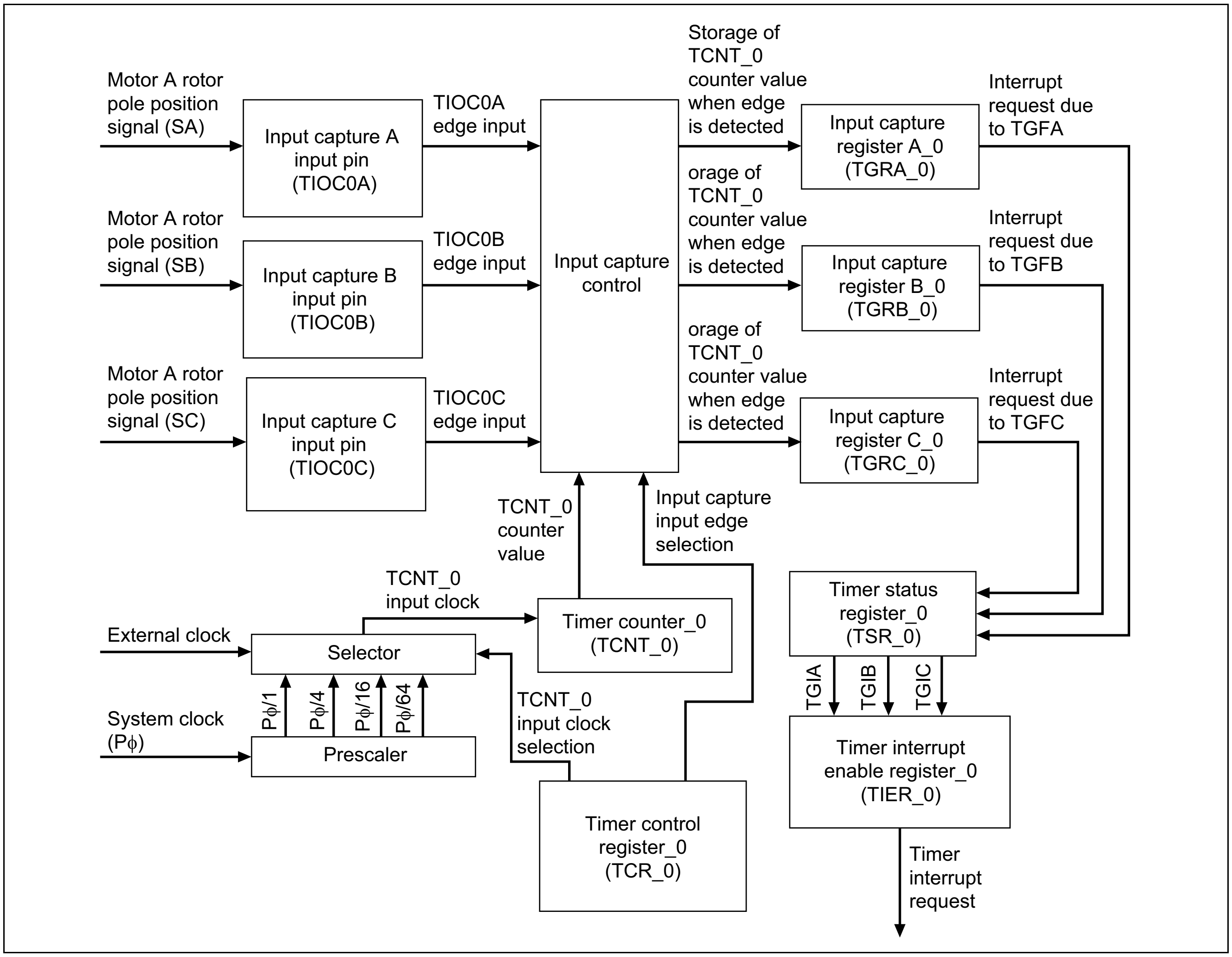

Motor A rotor pole position signal (SA)
Input capture A input pin (TIOC0A)
TIOC0A edge input
Motor A rotor pole position signal (SB)
Input capture B input pin (TIOC0B)
TIOC0B edge input
Motor A rotor pole position signal (SC)
Input capture C input pin (TIOC0C)
TIOC0C edge input
Input capture control
Storage of TCNT_0 counter value when edge is detected
orage of TCNT_0 counter value when edge is detected
orage of TCNT_0 counter value when edge is detected
Input capture register A_0 (TGRA_0)
Input capture register B_0 (TGRB_0)
Input capture register C_0 (TGRC_0)
Interrupt request due to TGFA
Interrupt request due to TGFB
Interrupt request due to TGFC
TCNT_0 counter value
Input capture input edge selection
TCNT_0 input clock
External clock
Selector
Timer counter_0 (TCNT_0)
Timer status register_0 (TSR_0)
TGIA
TGIB
TGIC
System clock (Pϕ)
Pϕ/1
Pϕ/4
Pϕ/16
Pϕ/64
Prescaler
TCNT_0 input clock selection
Timer interrupt enable register_0 (TIER_0)
Timer control register_0 (TCR_0)
Timer interrupt request

Figure 3 Block Diagram of Interrupt Generation on Detection of Motor A Rotor Pole Position Signal Input Edge Detection using Input Capture Function

3. The MTU compare match interrupt function is used to issue an interrupt request to the CPU every fixed rotation field switching period (time corresponding to a 60° portion of the motor rotation frequency) until a motor rotates once or more and a rotor pole position signal is detected. Figure 4 shows a block diagram of interrupt request generation every rotation field switching period using the MTU compare match interrupt function. The block diagram is described below.
   - The output compare registers (TGRD_0/TGRA_1) are 16-bit readable/writable registers. The contents of TGRD_0/TGRA_1 are constantly compared with TCNT_0/TCNT_1, and when both values match the TGFD bit of TSR_0 or TGFA bit of TSR_1 is set to 1. If TGIED of TIER_0 or TGIEA of TIER_1 is 1 at this time, an interrupt request is issued to the CPU.

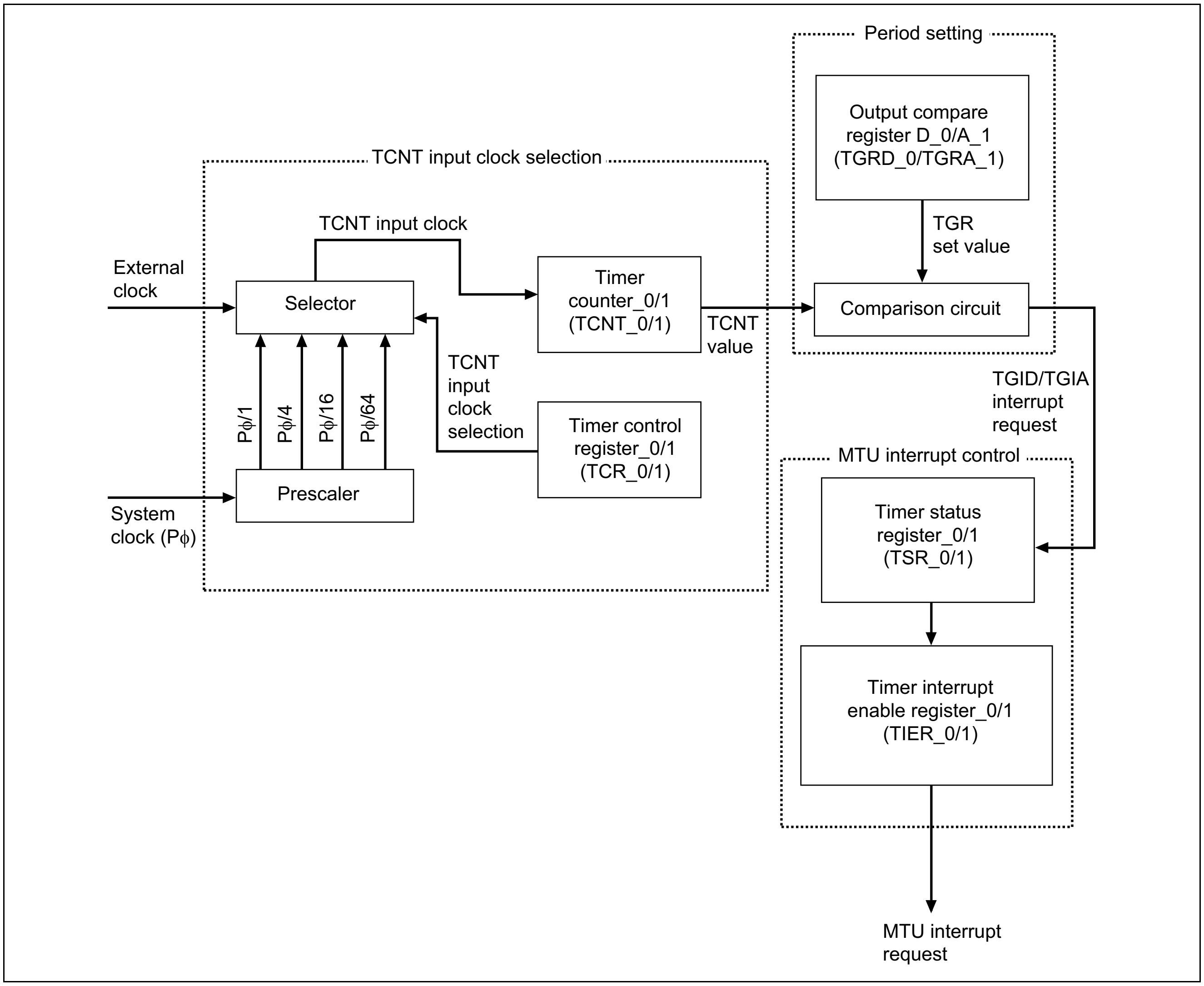

Period setting
Output compare register D_0/A_1 (TGRD_0/TGRA_1)
TGR set value
Comparison circuit
TCNT input clock selection
TCNT input clock
External clock
Selector
Timer counter_0/1 (TCNT_0/1)
TCNT value
TCNT input clock selection
Timer control register_0/1 (TCR_0/1)
Pϕ/1
Pϕ/4
Pϕ/16
Pϕ/64
Prescaler
System clock (Pϕ)
TGID/TGIA interrupt request
MTU interrupt control
Timer status register_0/1 (TSR_0/1)
Timer interrupt enable register_0/1 (TIER_0/1)
MTU interrupt request

Figure 4 Block Diagram of Interrupt Request Generation Every Rotation Field Switching Period Using MTU ch0/1 Compare Match Interrupt Function

4. By combining MTU/ch3 and ch4 (reset-synchronized PWM function), 3-phase PWM (positive-phase, negative-phase) waveform output is performed in which one waveform transition point is common. Timer counter_3 (TCNT_3) functions as an up-counter. After the end of initial control, the on/off status of the output of each phase can be switched automatically by inputting motor A rotor pole position signals to ch0 timer input pins TIOC0A, TIOC0B, and TIOC0C.
    - The TIOC3B pin functions as the base driver U-phase output pin.
    - The TIOC3D pin functions as the base driver $\overline{U}$-phase output pin.
    - The TIOC4A pin functions as the base driver V-phase output pin.
    - The TIOC4C pin functions as the base driver $\overline{V}$-phase output pin.
    - The TIOC4B pin functions as the base driver W-phase output pin.
    - The TIOC4D pin functions as the base driver $\overline{W}$-phase output pin.
    - The TIOC0A pin functions as the motor A rotor pole position signal (SA) input pin.
    - The TIOC0B pin functions as the motor A rotor pole position signal (SB) input pin.
    - The TIOC0C pin functions as the motor A rotor pole position signal (SC) input pin.

5. By means of the MMT, a 3-phase chopping waveform is generated, and is output as the on-time motor B positive-phase/negative-phase control waveform.
   (When off, port output is set, and 0 is output.)
    - The PU0A pin functions as the base driver U-phase output pin.
    - The PU0B pin functions as the base driver $\overline{U}$-phase output pin.
    - The PV0A pin functions as the base driver V-phase output pin.
    - The PV0B pin functions as the base driver $\overline{V}$-phase output pin.
    - The PW0A pin functions as the base driver W-phase output pin.
    - The PW0B pin functions as the base driver $\overline{W}$-phase output pin.

6. Using the IRQ external interrupt function, control is performed by means of rotor pole position signal input from the motor after the end of motor B initial control.
   Figure 5 is a block diagram of interrupt request generation due to rotor pole position signal input edge detection, using the IRQ external interrupt function.
   The block diagram is described below.
    - The IRQ0/1/2 pins are used as motor B rotor pole position signal (HU, HV, HW) input pins.
    - An IRQ interrupt is generated by detection of an IRQ0/1/2 pin input edge. Rising or falling input edge sensing can be selected for the IRQ pins.
    - Interrupt control register 1 (ICR1) is a 16-bit register that performs IRQ pin input signal detection mode setting. In this sample task, external signal input edge interrupt request detection is set.
    - Interrupt control register 2 (ICR2) is a 16-bit register that sets the IRQ pin edge detection mode.
    - The IRQ status register (ISR) is a 16-bit register in which a corresponding bit is set to 1 when an IRQ interrupt request is generated.
    - The IRQ0 pin functions as the motor B rotor pole position signal (HU) input pin.
    - The IRQ1 pin functions as the motor B rotor pole position signal (HV) input pin.
    - The IRQ2 pin functions as the motor B rotor pole position signal (HW) input pin.

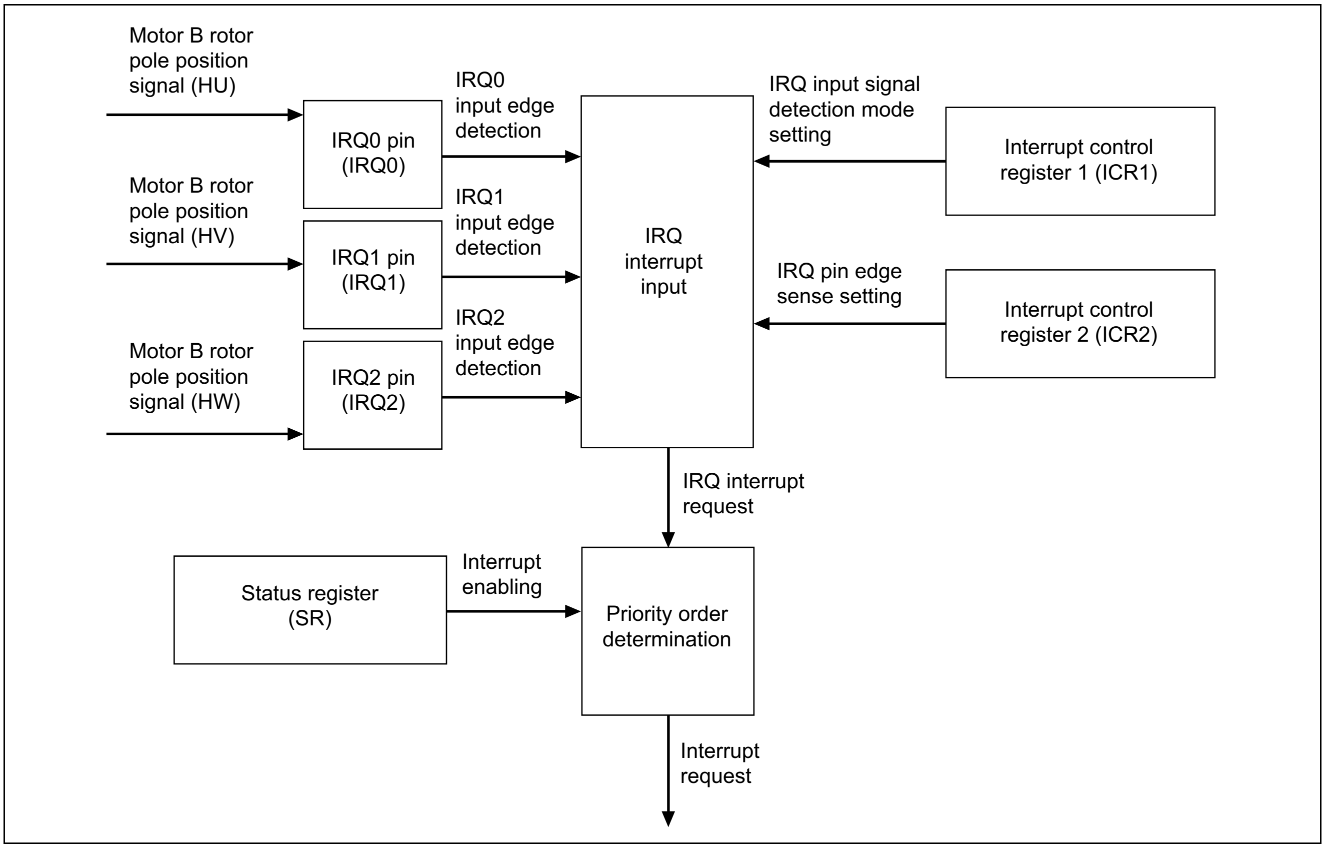

Motor B rotor pole position signal (HU)
Motor B rotor pole position signal (HV)
Motor B rotor pole position signal (HW)
IRQ0 pin (IRQ0)
IRQ1 pin (IRQ1)
IRQ2 pin (IRQ2)
IRQ0 input edge detection
IRQ1 input edge detection
IRQ2 input edge detection
IRQ interrupt input
IRQ input signal detection mode setting
Interrupt control register 1 (ICR1)
IRQ pin edge sense setting
Interrupt control register 2 (ICR2)
IRQ interrupt request
Status register (SR)
Interrupt enabling
Priority order determination
Interrupt request

Figure 5 Block Diagram of Interrupt Generation Due to Motor B Rotor Pole Position Signal Input Edge Detection Using IRQ External Interrupt Function

Table 1 shows the function assignments used in this sample task. 120° current flow control and 180° current flow control is performed for two DC brushless motors by assigning SH7046 functions as shown in table 1.

Table 1 Function Assignments in this Sample Task (1)

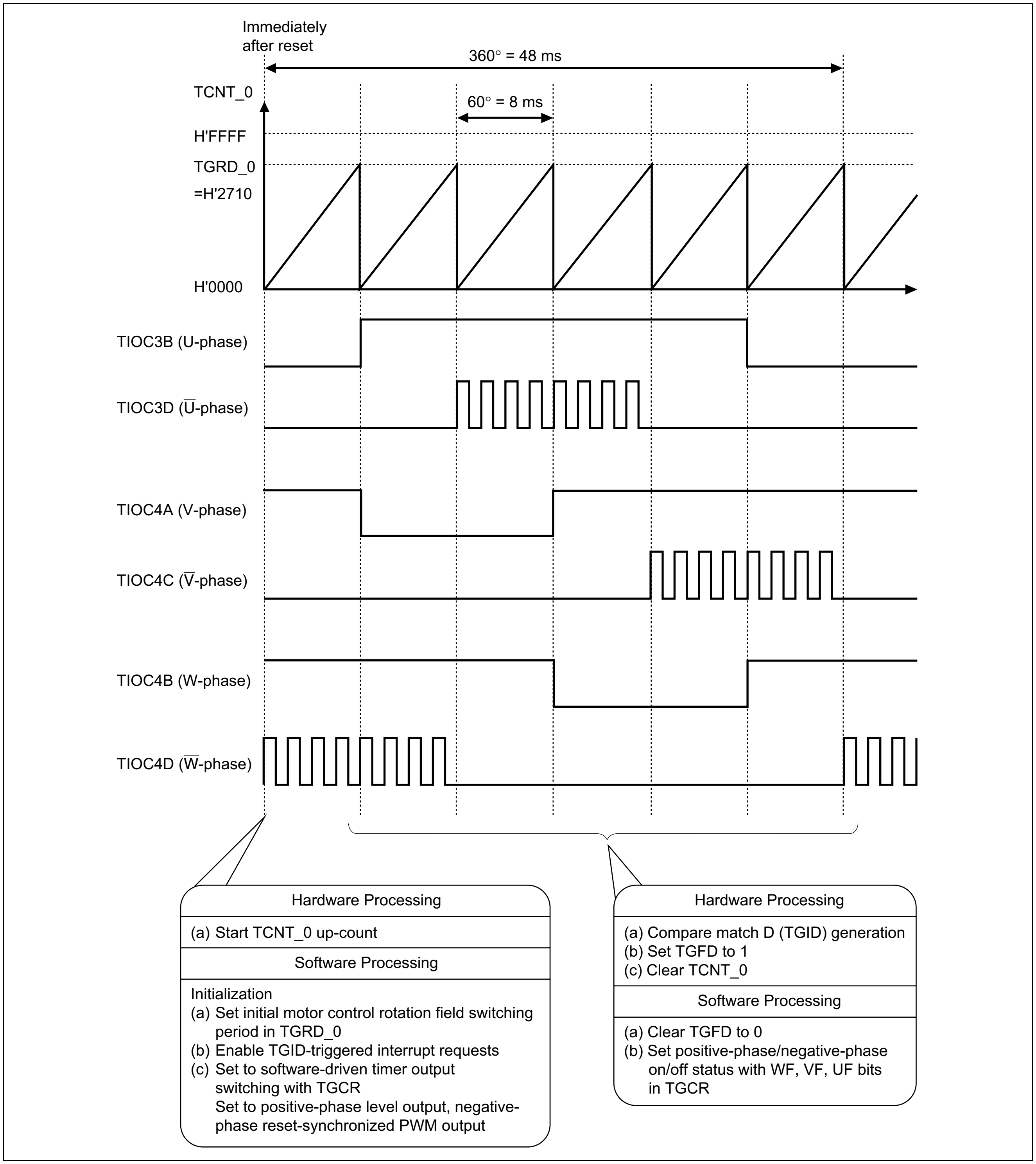
SH7046 Function	
Motor A	**Function Assignment**
TIOC0A/B/C	Motor A rotor pole position signal input pins (SA, SB, SC)
TCNT_0	16-bit up-counter. Input clock is set with TCR_0
TCR_0	TCNT_0 input clock setting. TCNT_0 clearing source selection
TIER_0	Sets MTU/ch0 interrupt request enabling or disabling
TGRD_0	Motor rotation field switching period setting in initial control
TIORH_0,TIORL_0	Rotor pole position signal detection edge selection
TIOC3B	U-phase output pin
TIOC4A	V-phase output pin
TIOC4B	W-phase output pin
TIOC3D	$\overline{U}$-phase output pin
TIOC4B	$\overline{V}$-phase output pin
TIOC4D	$\overline{W}$-phase output pin
TCNT_3	16-bit up-counter. Input clock is set with TCR_3
TCR_3	TCNT_3 input clock setting. TCNT_3 clearing source selection
TGRA_3	Sets PWM output period
TGRB_3/4,TGRA_4	Setting of U-phase, V-phase, W-phase PWM waveform duty cycles
TOCR	Compare match positive-phase/negative-phase output level setting
TMDR_3	Timer operating mode setting

Table 1 Function Assignments in this Sample Task (2)

SH7046 Function	
Motor B	**Function Assignment**
TCNT_1	16-bit up-counter. Input clock is set with TCR_1
TCR_1	TCNT_1 input clock setting. TCNT_1 clearing source selection
TIER_1	Sets MTU/ch1 interrupt request enabling or disabling
TGRA_1	Motor rotation field switching period setting in initial control
PUOA	U-phase output pin
PVOA	V-phase output pin
PWOA	W-phase output pin
PUOB	$\overline{U}$-phase output pin
PVOB	$\overline{V}$-phase output pin
PWOB	$\overline{W}$-phase output pin
MMT_TCNT	16-bit up-counter. Input clock set with MMT_TMDR
MMT_TMDR	MMT_TCNT input clock setting. Sets operating mode output level
TPBR	Sets 1/2 PWM output period
TBRU/V/W	Setting of U-phase, V-phase, W-phase PWM waveform duty cycles
IRQ0/1/2	Motor B rotor pole position signal input pins (HU, HV, HW)
ICR1	Sets IRQ pin input signal detection mode
ICR2	Sets IRQ pin edge detection mode
ISR	Reflects presence/absence of IRQ external interrupt request
TOCR	Positive-phase/negative-phase PWM output compare match output level setting

4. Principles of Operation

Figure 6 illustrates the principles of operation in motor A initial control (in which the rotation field is switched every fixed period until the motor rotates once and a rotor pole position signal is detected). Motor A initial control is performed by SH7046 hardware and software processing as shown in figure 6.



Hardware Processing

(a) Start TCNT_0 up-count

Software Processing

Initialization
(a) Set initial motor control rotation field switching period in TGRD_0
(b) Enable TGID-triggered interrupt requests
(c) Set to software-driven timer output switching with TGCR
Set to positive-phase level output, negative-phase reset-synchronized PWM output

Hardware Processing

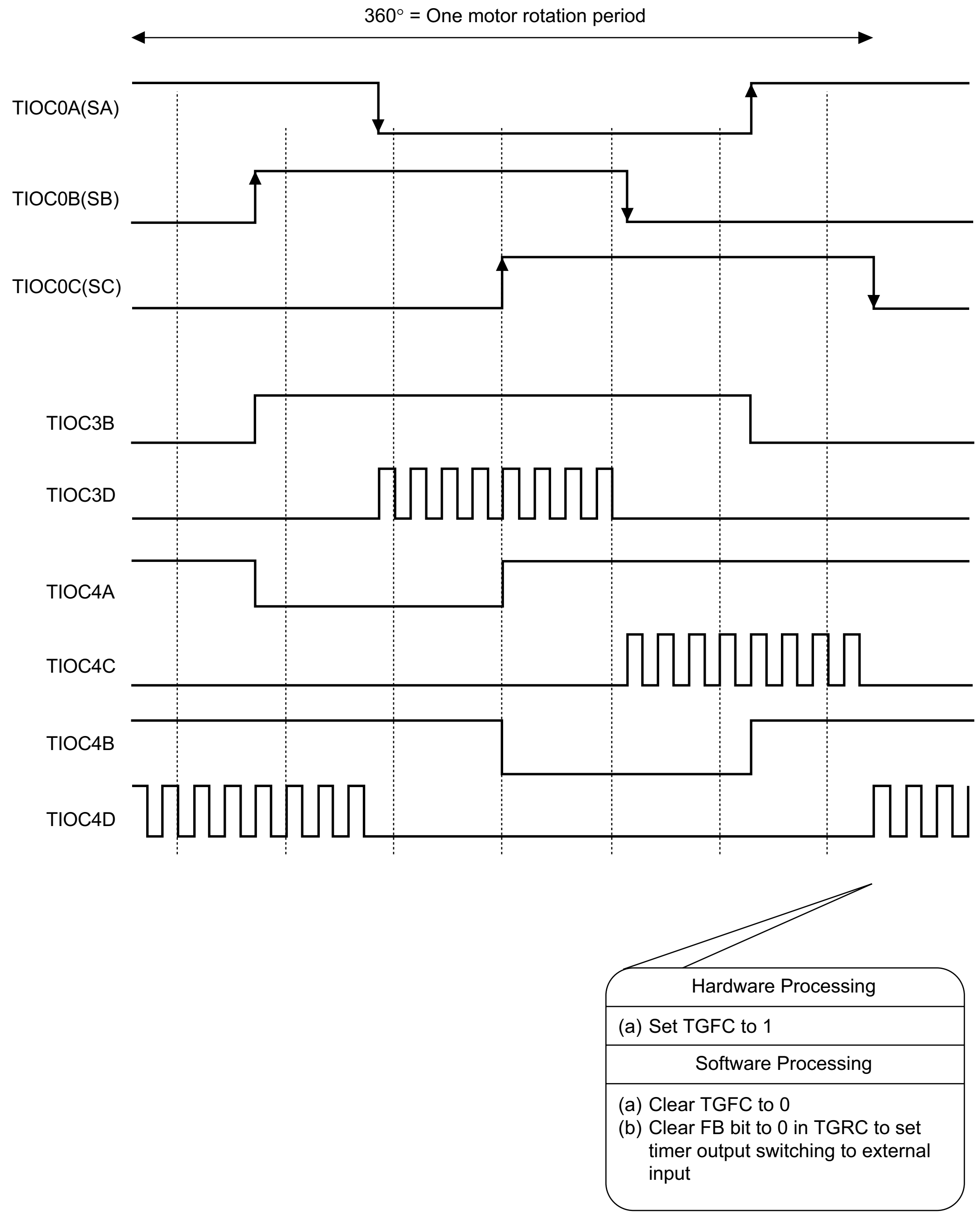
(a) Compare match D (TGID) generation
(b) Set TGFD to 1
(c) Clear TCNT_0

Software Processing

(a) Clear TGFD to 0
(b) Set positive-phase/negative-phase on/off status with WF, VF, UF bits in TGCR

Figure 6 Principles of Operation in Motor A Initial Control

Figure 7 illustrates the principles of operation in automatic rotation field switching control by means of motor A rotor pole position signal detection. Motor control by means of rotor pole position signal detection is performed by SH7046 hardware and software processing as shown in figure 7.



Hardware Processing

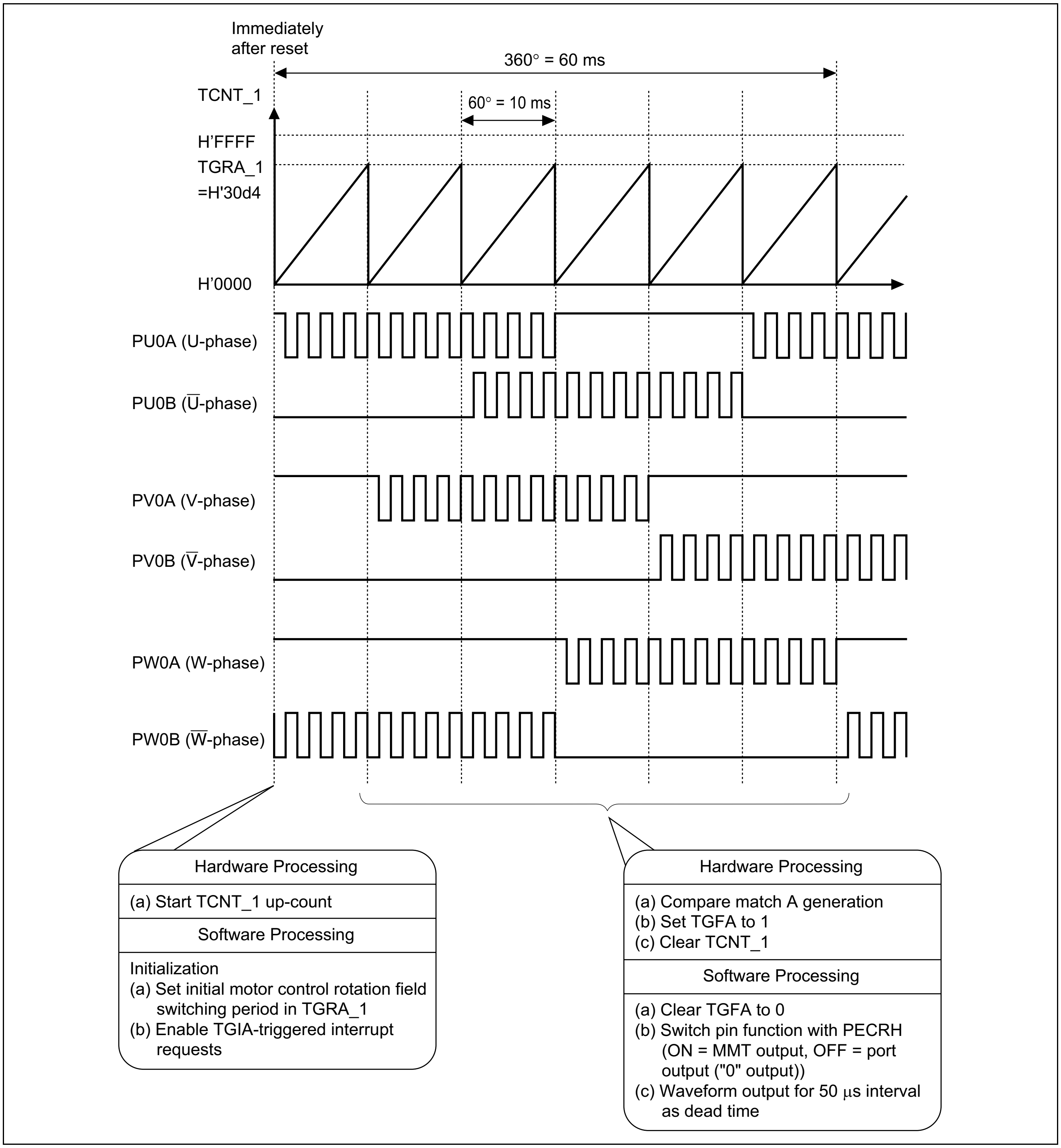
(a) Set TGFC to 1

Software Processing

(a) Clear TGFC to 0
(b) Clear FB bit to 0 in TGRC to set timer output switching to external input

Figure 7 Principles of Operation in Excitation Phase Switching Control by Motor A Rotor Pole Position Detection

Figure 8 illustrates the principles of operation in motor B initial control (in which the rotation field is switched every fixed period until the motor rotates once and a rotor pole position signal is detected). Motor initial control is performed by SH7046 hardware and software processing as shown in figure 8.



Hardware Processing

(a) Start TCNT_1 up-count

Software Processing

Initialization
(a) Set initial motor control rotation field switching period in TGRA_1
(b) Enable TGIA-triggered interrupt requests

Hardware Processing

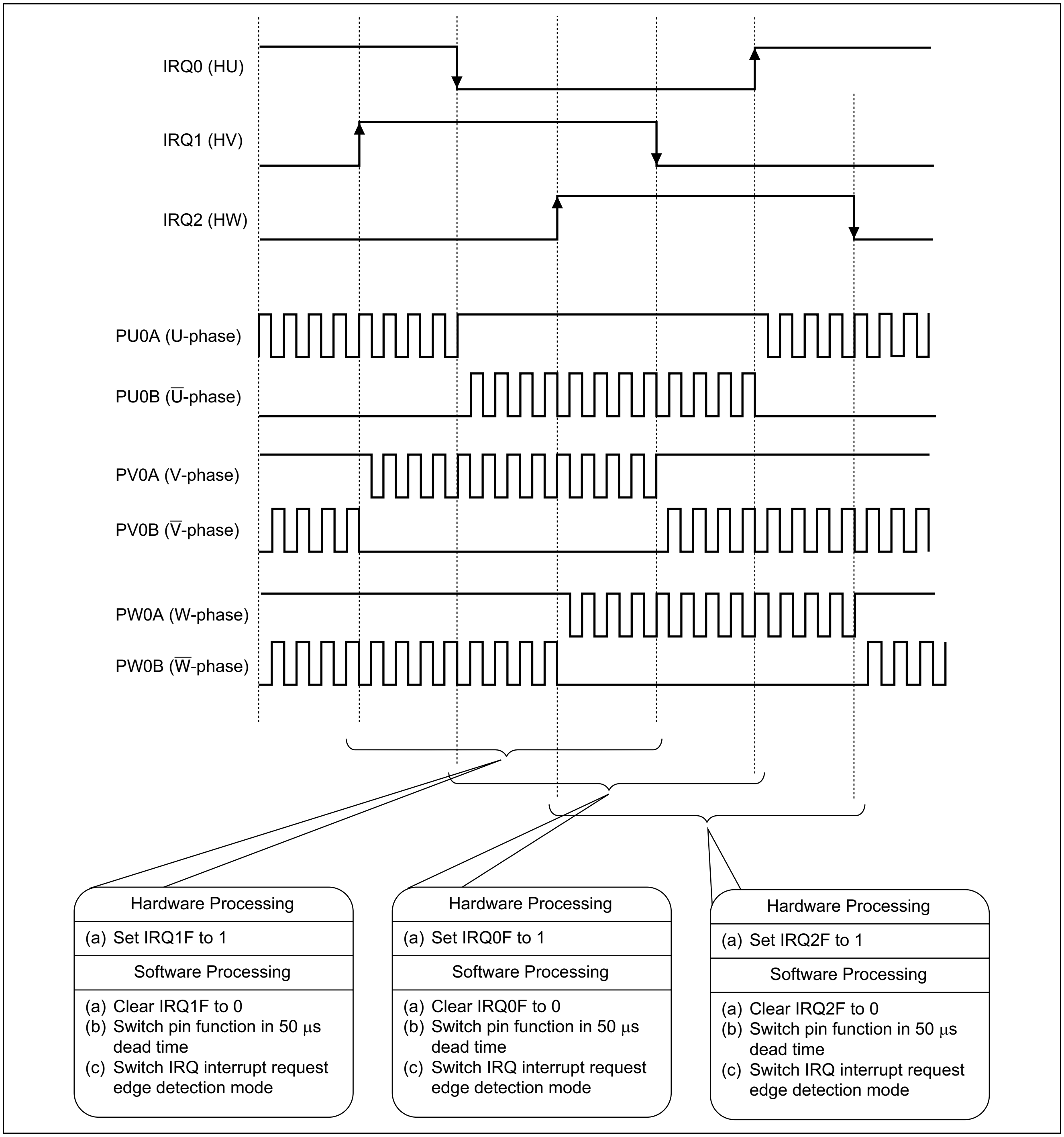
(a) Compare match A generation
(b) Set TGFA to 1
(c) Clear TCNT_1

Software Processing

(a) Clear TGFA to 0
(b) Switch pin function with PECRH (ON = MMT output, OFF = port output ("0" output))
(c) Waveform output for 50 μs interval as dead time

Figure 8 Principles of Operation in Motor B Initial Control

Figure 9 illustrates the principles of operation in rotation field switching control by means of motor B rotor pole position signal detection. Control by means of rotor pole position signal detection is performed by SH7046 hardware and software processing as shown in figure 9.



Hardware Processing

(a) Set IRQ1F to 1

Software Processing

(a) Clear IRQ1F to 0
(b) Switch pin function in 50 µs dead time
(c) Switch IRQ interrupt request edge detection mode

Hardware Processing

(a) Set IRQ0F to 1

Software Processing

(a) Clear IRQ0F to 0
(b) Switch pin function in 50 µs dead time
(c) Switch IRQ interrupt request edge detection mode

Hardware Processing

(a) Set IRQ2F to 1

Software Processing

(a) Clear IRQ2F to 0
(b) Switch pin function in 50 µs dead time
(c) Switch IRQ interrupt request edge detection mode

Figure 9 Principles of Operation in Excitation Phase Switching Control by Motor B Rotor Pole Position Detection

5. Software

(1) Modules

Table 2 shows the modules used in this sample task.

Table 2 Modules

Module Name	Label	Functions
Main routine	main	Performs module standby mode clearing, SR setting, and timer counter starting
Initialization	IO_INIT	Performs register initialization and setting of RAM used
Motor A initial control routine	rotateA	Using MTU/ch0 compare match interrupts, switches rotation field every fixed period until motor A rotates once and rotor pole position signal is detected
Motor B initial control routine	rotateB	Using MTU/ch0 compare match interrupts, switches rotation field every fixed period until motor B rotates once and rotor pole position signal is detected
Motor B rotor pole position signal (HU) driven rotation field switching routine	HU	Performs excitation phase switching by means of rotor pole position signal (HU) output from motor B to IRQ0 pin
Motor B rotor pole position signal (HV) driven rotation field switching routine	HV	Performs excitation phase switching by means of rotor pole position signal (HV) output from motor B to IRQ1 pin
Motor B rotor pole position signal (HW) driven rotation field switching routine	HW	Performs excitation phase switching by means of rotor pole position signal (HW) output from motor B to IRQ2 pin

(2) Arguments

This sample task does not use any arguments.

(3) Internal Registers Used

Table 3 shows the internal registers used by this sample task.

Table 3 Internal Registers Used (1)

MTU/ch0

Register Name		Function	Address	Set Value
TCR_0		Timer control register_0 • Sets TGRD_0 compare match as counter clearing source • Sets Pϕ/16 as TCNT_0 counter clock	H'FFFF8260	H'C2
TMDR_0		Timer mode register_0 • Sets normal operation as timer operating mode	H'FFFF8261	H'C0
TIER_0	TGIEA	Timer interrupt enable register_0 (TGR interrupt enable A) • When TGIEA = 0, interrupts by TGFA bit are disabled • When TGIEA = 1, interrupts by TGFA bit are enabled	H'FFFF8264 Bit 0	0
	TGIEB	Timer interrupt enable register_0 (TGR interrupt enable B) • When TGIEB = 0, interrupts by TGFB bit are disabled • When TGIEB = 1, interrupts by TGFB bit are enabled	H'FFFF8264 Bit 1	0
	TGIEC	Timer interrupt enable register_0 (TGR interrupt enable C) • When TGIEC = 0, interrupts by TGFC bit are disabled • When TGIEC = 1, interrupts by TGFC bit are enabled	H'FFFF8264 Bit 2	0
	TGIED	Timer interrupt enable register_0 (TGR interrupt enable D) • When TGIED = 0, interrupts by TGFD bit are disabled • When TGIED = 1, interrupts by TGFD bit are enabled	H'FFFF8264 Bit 3	1
TGRD_0		Timer general register D_0 • Used as output compare register. Constantly compared with TCNT_0; when compare match occurs, TGFD bit is set to 1 in TSR_0	H'FFFF826E	H'2710
TIORH_0	IOA0 IOA1 IOA2 IOA3	Timer I/O control register H_0 • When 1000, input capture at rising edge of TIOC0A • When 1001, input capture at falling edge of TIOC0A	H'FFFF8262 Bit 0 Bit 1 Bit 2 Bit 3	H'88
	IOB0 IOB1 IOB2 IOB3	Timer I/O control register H_0 • When 1000, input capture at rising edge of TIOC0B • When 1001, input capture at falling edge of TIOC0B	H'FFFF8262 Bit 4 Bit 5 Bit 6 Bit 7	
TIORL_0	IOC0 IOC1 IOC2 IOC3	Timer I/O control register L_0 • When 1000, input capture at rising edge of TIOC0C • When 1001, input capture at falling edge of TIOC0C	H'FFFF8263 Bit 0 Bit 1 Bit 2 Bit 3	H'09
	IOD0 IOD1 IOD2 IOD3	Timer I/O control register L_0 • When 0000, TGRD_0 functions as output compare register	H'FFFF8263 Bit 4 Bit 5 Bit 6 Bit 7	

Table 3 Internal Registers Used (2)

MTU/ch1

Register Name		Function	Address	Set Value
TCR_1		Timer control register_1 • Sets TGRA_1 compare match as counter clearing source • Sets Pϕ/16 as TCNT_1 counter clock	H'FFFF8280	H'22
TMDR_1		Timer mode register_1 • Sets normal operation as timer operating mode	H'FFFF8281	H'C0
TIER_0	TGIEA	Timer interrupt enable register_1 (TGR interrupt enable A) • When TGIEA = 0, interrupts by TGFA bit are disabled • When TGIEA = 1, interrupts by TGFA bit are enabled	H'FFFF8284 Bit 0	1
	TGIEB	Timer interrupt enable register_1 (TGR interrupt enable B) • When TGIEB = 0, interrupts by TGFB bit are disabled • When TGIEB = 1, interrupts by TGFB bit are enabled	H'FFFF8284 Bit 1	0
	TGIEC	Timer interrupt enable register_1 (TGR interrupt enable C) • When TGIEC = 0, interrupts by TGFC bit are disabled • When TGIEC = 1, interrupts by TGFC bit are enabled	H'FFFF8284 Bit 2	0
	TGIED	Timer interrupt enable register_1 (TGR interrupt enable D) • When TGIED = 0, interrupts by TGFD bit are disabled • When TGIED = 1, interrupts by TGFD bit are enabled	H'FFFF8284 Bit 3	0
TGRA_1		Timer general register A_1 • Used as output compare register. Constantly compared with TCNT_1; when compare match occurs, TGFA bit is set to 1 in TSR_1	H'FFFF8288	H'30D4

Table 3 Internal Registers Used (3)

MMT

Register Name		Function	Address	Set Value
MMT_TMDR	OLSP	Timer mode register (Output level select P) Selects positive-phase output level in operating mode • When 0, low level • When 1, high level	H'FFFF8A00 Bit 2	H'C0
	OLSN	Timer mode register (Output level select N) Selects negative-phase output level in operating mode • When 0, low level • When 1, high level	H'FFFF8A00 Bit 3	
TPBR		Timer period buffer register • Sets 1/2 PWM carrier period	H'FFFF8A0A	H'01F4
TBRU		Timer buffer register U • Sets U-phase output waveform PWM duty cycle	H'FFFF8A1C	H'007D
TBRV		Timer buffer register V • Sets V-phase output waveform PWM duty cycle	H'FFFF8A2C	H'007D
TBRW		Timer buffer register W • Sets W-phase output waveform PWM duty cycle	H'FFFF8A3C	H'007D

Table 3 Internal Registers Used (4)

INTC

Register Name		Function	Address	Set Value
ICR1	IRQ2S	Interrupt control register 1 (IRQ2 sense select) • When 0, interrupt request detected at IRQ2 input low level • When 1, interrupt request detected at IRQ2 input edge (Edge direction selected with ICR2)	H'FFFF8358 Bit 5	1
	IRQ1S	Interrupt control register 1 (IRQ1 sense select) • When 0, interrupt request detected at IRQ1 input low level • When 1, interrupt request detected at IRQ1 input edge (Edge direction selected with ICR2)	H'FFFF8358 Bit 6	1
	IRQ0S	Interrupt control register 1 (IRQ0 sense select) • When 0, interrupt request detected at IRQ0 input low level • When 1, interrupt request detected at IRQ0 input edge (Edge direction selected with ICR2)	H'FFFF8358 Bit 7	1
ICR2	IRQ2ES0 IRQ2ES1	Interrupt control register 2 • When 00, interrupt request detected at $\overline{\text{IRQ2}}$ input falling edge • When 01, interrupt request detected at $\overline{\text{IRQ2}}$ input rising edge	H'FFFF8366 Bit 10 Bit 11	00
	IRQ1ES0 IRQ1ES1	Interrupt control register 2 • When 00, interrupt request detected at $\overline{\text{IRQ2}}$ input falling edge • When 01, interrupt request detected at $\overline{\text{IRQ2}}$ input rising edge	H'FFFF8366 Bit 12 Bit 13	00
ISR	IRQ2F	IRQ status register • Set to 1 when IRQ2 input pin edge is detected	H'FFFF835A Bit 5	0
	IRQ1F	IRQ status register • Set to 1 when IRQ1 input pin edge is detected	H'FFFF835A Bit 6	0
	IRQ0F	IRQ status register • Set to 1 when IRQ0 input pin edge is detected	H'FFFF835A Bit 7	0

Table 3 Internal Registers Used (5)

PFC

Register Name		Function	Address	Set Value
PECRH		Port E control register H • When 01, MMT output • When 00, port output	H'FFFF83BC	H'0000
PECRL1		Port E control register L1 • Set to MTU output	H'FFFF83B8	H'5544
PECRL2	PE0MD0 PE0MD1	Port E control register L2 (PE0 mode bit) • Motor A rotor pole position signal (SA) input pin, set as input capture input pin	H'FFFF83BA Bit 0 Bit 1	01
	PE1MD0 PE1MD1	Port E control register L2 (PE1 mode bit) • Motor A rotor pole position signal (SB) input pin, set as input capture input pin	H'FFFF83BA Bit 2 Bit 3	01
	PE2MD0 PE2MD1	Port E control register L2 (PE2 mode bit) • Motor A rotor pole position signal (SC) input pin, set as input capture input pin	H'FFFF83BA Bit 4 Bit 5	01
PBCR1 PBCR2 PBCR2	PB2MD2 PB2MD0 PB2MD1	Port B control register 1 Port B control register 2 (PB2 mode) • Motor B rotor pole position signal (HU) input pin, set as IRQ0 input pin	H'FFFF8398 Bit 10 H'FFFF839A Bit 4 Bit 5	001
PBCR1 PBCR2 PBCR2	PB3MD2 PB3MD1 PB3MD0	Port B control register 1 Port B control register 2 (PB3 mode) • Motor B SB rotor pole position signal input pin, set as IRQ1 input pin	H'FFFF8398 Bit 11 H'FFFF839A Bit 6 Bit 7	001
PBCR1 PBCR2 PBCR2	PB4MD2 PB4MD1 PB4MD0	Port B control register 1 Port B control register 2 (PB4 mode) • Motor B SC rotor pole position signal input pin, set as IRQ2 input pin	H'FFFF8398 Bit 12 H'FFFF839A Bit 8 Bit 9	001

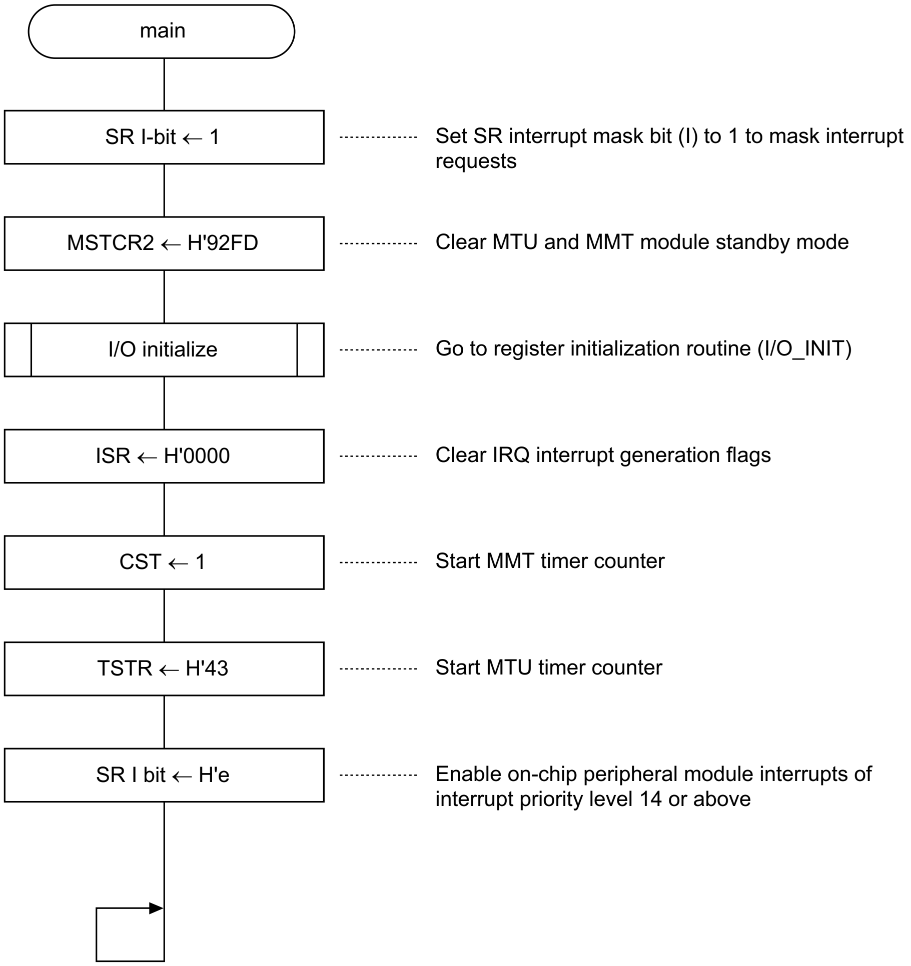
(4) RAM Used

Table 4 RAM Used

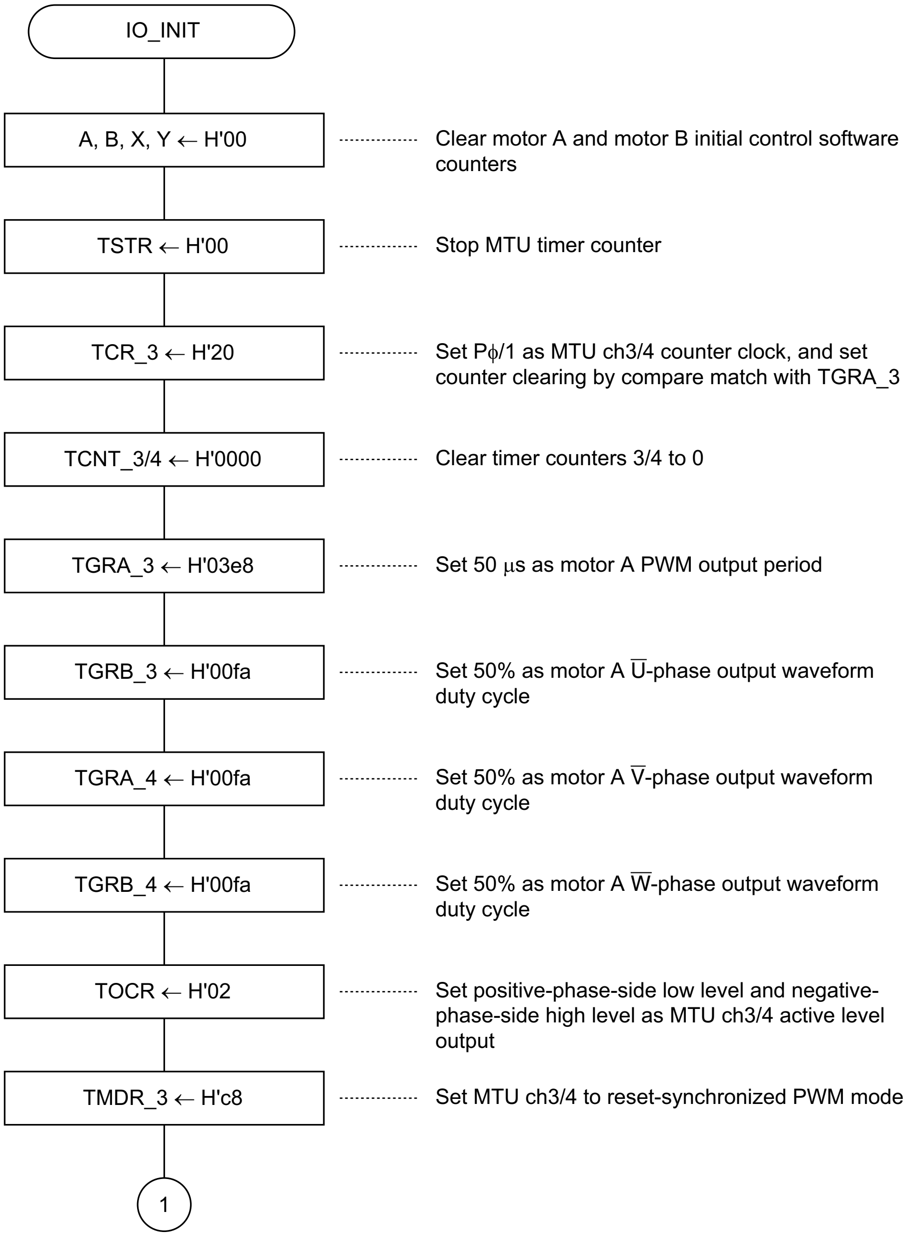
Label	Function	Address	Module
A	Motor A initial control excitation phase switching counter	H'FFFFD000	rotateA
B	Motor A rotation speed control counter	H'FFFFD001	rotateA
X	Motor B initial control excitation phase switching counter	H'FFFFD002	rotateB
Y	Motor B rotation speed control counter	H'FFFFD003	rotateB
i	Dead time generation counter	H'FFFFD004	HU, HV, HW

6. Flowcharts

(1) Main routine


main
SR I-bit ← 1
Set SR interrupt mask bit (I) to 1 to mask interrupt requests
MSTCR2 ← H'92FD
Clear MTU and MMT module standby mode
I/O initialize
Go to register initialization routine (I/O_INIT)
ISR ← H'0000
Clear IRQ interrupt generation flags
CST ← 1
Start MMT timer counter
TSTR ← H'43
Start MTU timer counter
SR I bit ← H'e
Enable on-chip peripheral module interrupts of interrupt priority level 14 or above

(2) Register initialization routine



Clear motor A and motor B initial control software counters

Stop MTU timer counter

Set Pϕ/1 as MTU ch3/4 counter clock, and set counter clearing by compare match with TGRA_3

Clear timer counters 3/4 to 0

Set 50 µs as motor A PWM output period

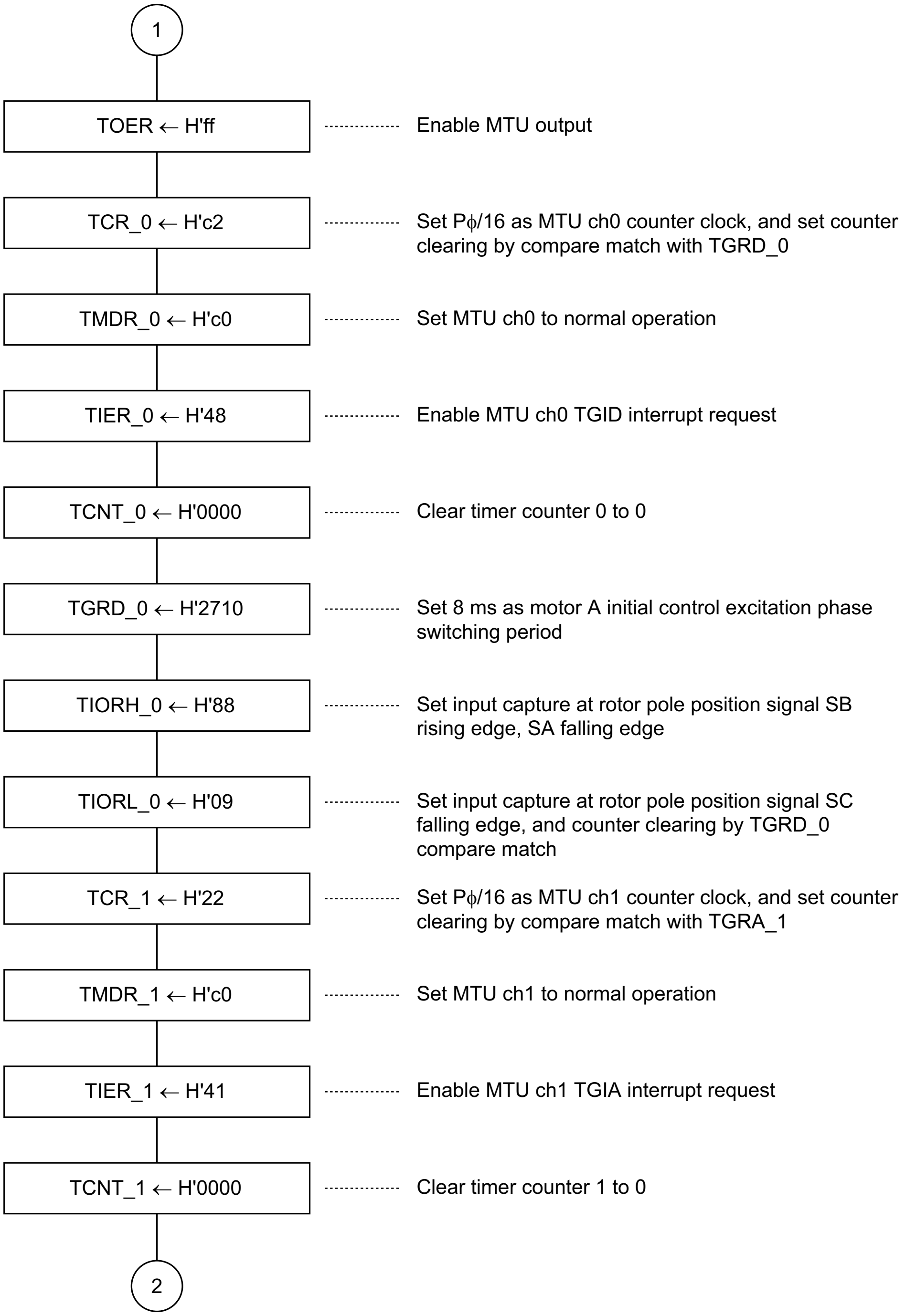
Set 50% as motor A $\overline{U}$-phase output waveform duty cycle

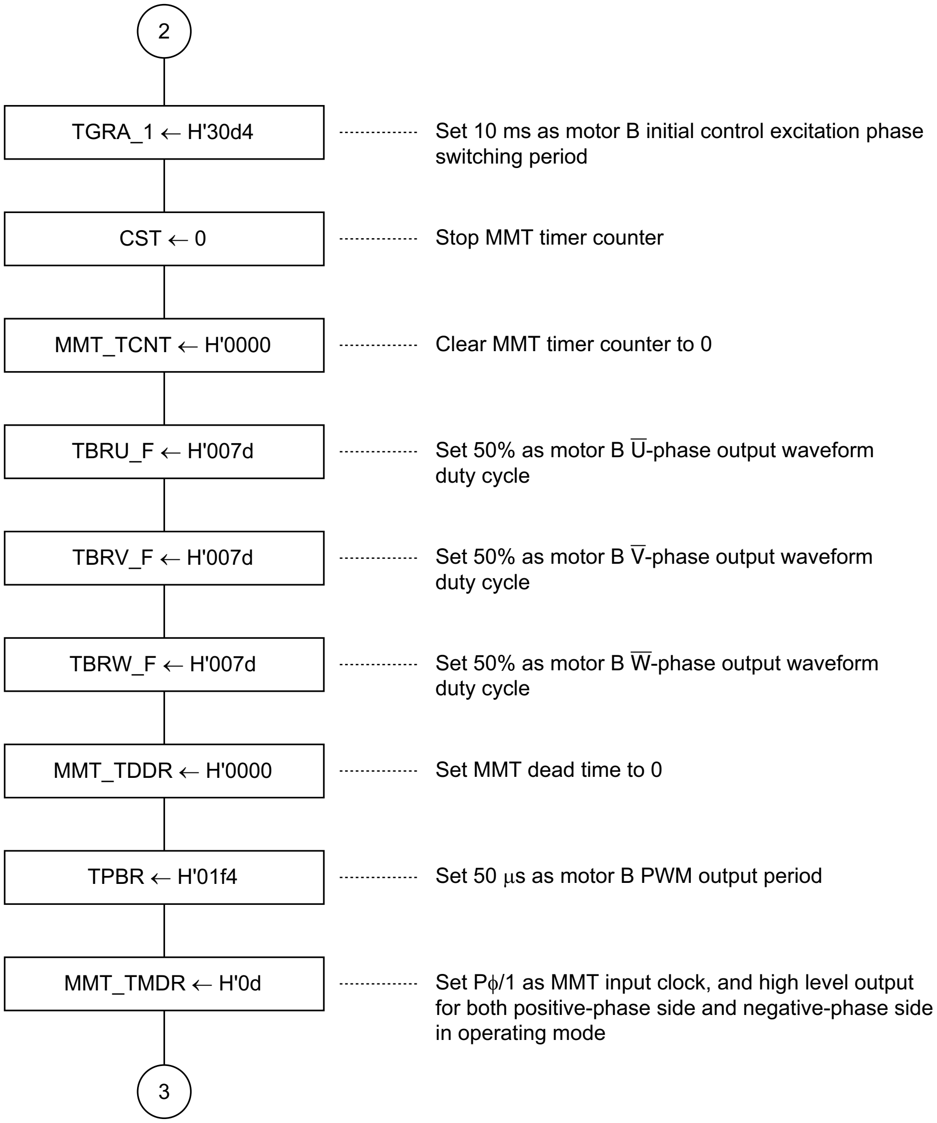
Set 50% as motor A $\overline{V}$-phase output waveform duty cycle

Set 50% as motor A $\overline{W}$-phase output waveform duty cycle

Set positive-phase-side low level and negative-phase-side high level as MTU ch3/4 active level output

Set MTU ch3/4 to reset-synchronized PWM mode


1
TOER ← H'ff
Enable MTU output
TCR_0 ← H'c2
Set Pϕ/16 as MTU ch0 counter clock, and set counter clearing by compare match with TGRD_0
TMDR_0 ← H'c0
Set MTU ch0 to normal operation
TIER_0 ← H'48
Enable MTU ch0 TGID interrupt request
TCNT_0 ← H'0000
Clear timer counter 0 to 0
TGRD_0 ← H'2710
Set 8 ms as motor A initial control excitation phase switching period
TIORH_0 ← H'88
Set input capture at rotor pole position signal SB rising edge, SA falling edge
TIORL_0 ← H'09
Set input capture at rotor pole position signal SC falling edge, and counter clearing by TGRD_0 compare match
TCR_1 ← H'22
Set Pϕ/16 as MTU ch1 counter clock, and set counter clearing by compare match with TGRA_1
TMDR_1 ← H'c0
Set MTU ch1 to normal operation
TIER_1 ← H'41
Enable MTU ch1 TGIA interrupt request
TCNT_1 ← H'0000
Clear timer counter 1 to 0
2



Set 10 ms as motor B initial control excitation phase switching period

Stop MMT timer counter

Clear MMT timer counter to 0

Set 50% as motor B $\overline{U}$-phase output waveform duty cycle

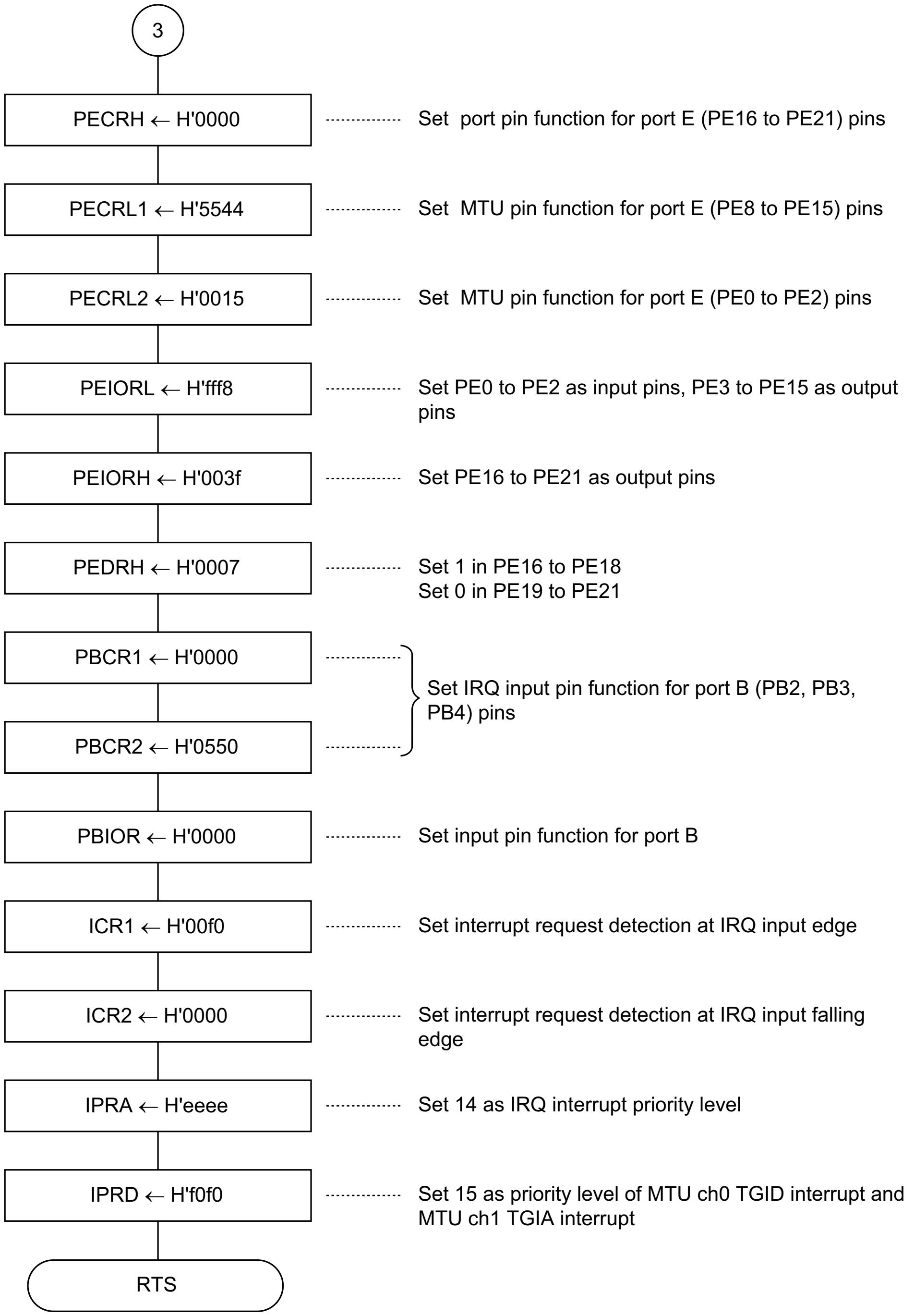
Set 50% as motor B $\overline{V}$-phase output waveform duty cycle

Set 50% as motor B $\overline{W}$-phase output waveform duty cycle

Set MMT dead time to 0

Set 50 µs as motor B PWM output period

Set Pϕ/1 as MMT input clock, and high level output for both positive-phase side and negative-phase side in operating mode



Set port pin function for port E (PE16 to PE21) pins

Set MTU pin function for port E (PE8 to PE15) pins

Set MTU pin function for port E (PE0 to PE2) pins

Set PE0 to PE2 as input pins, PE3 to PE15 as output pins

Set PE16 to PE21 as output pins

Set 1 in PE16 to PE18
Set 0 in PE19 to PE21

Set IRQ input pin function for port B (PB2, PB3, PB4) pins

Set input pin function for port B

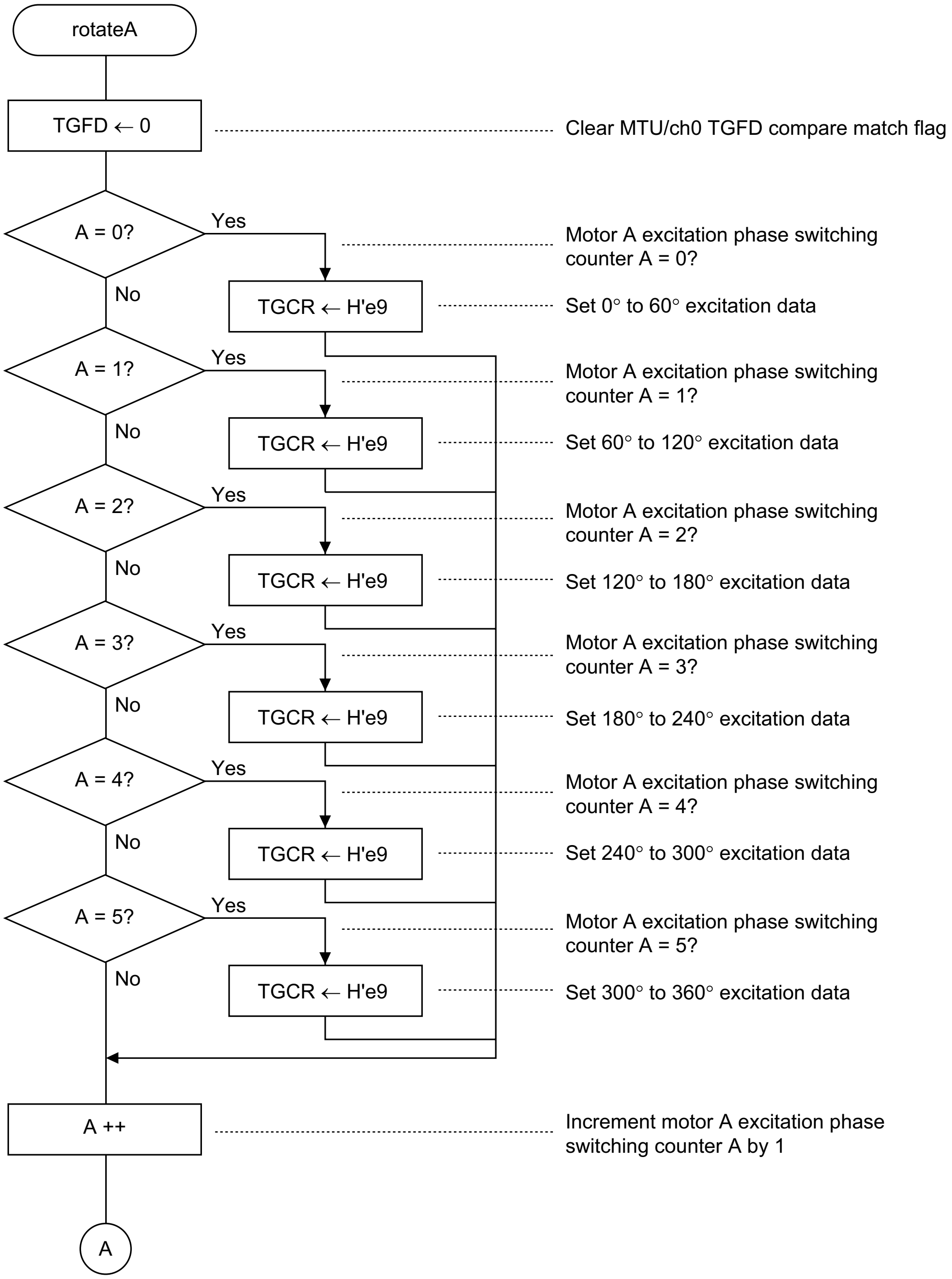
Set interrupt request detection at IRQ input edge

Set interrupt request detection at IRQ input falling edge

Set 14 as IRQ interrupt priority level

Set 15 as priority level of MTU ch0 TGID interrupt and MTU ch1 TGIA interrupt

(3) MTU ch0 interrupt handling



Clear MTU/ch0 TGFD compare match flag

Motor A excitation phase switching counter A = 0?

Set 0° to 60° excitation data

Motor A excitation phase switching counter A = 1?

Set 60° to 120° excitation data

Motor A excitation phase switching counter A = 2?

Set 120° to 180° excitation data

Motor A excitation phase switching counter A = 3?

Set 180° to 240° excitation data

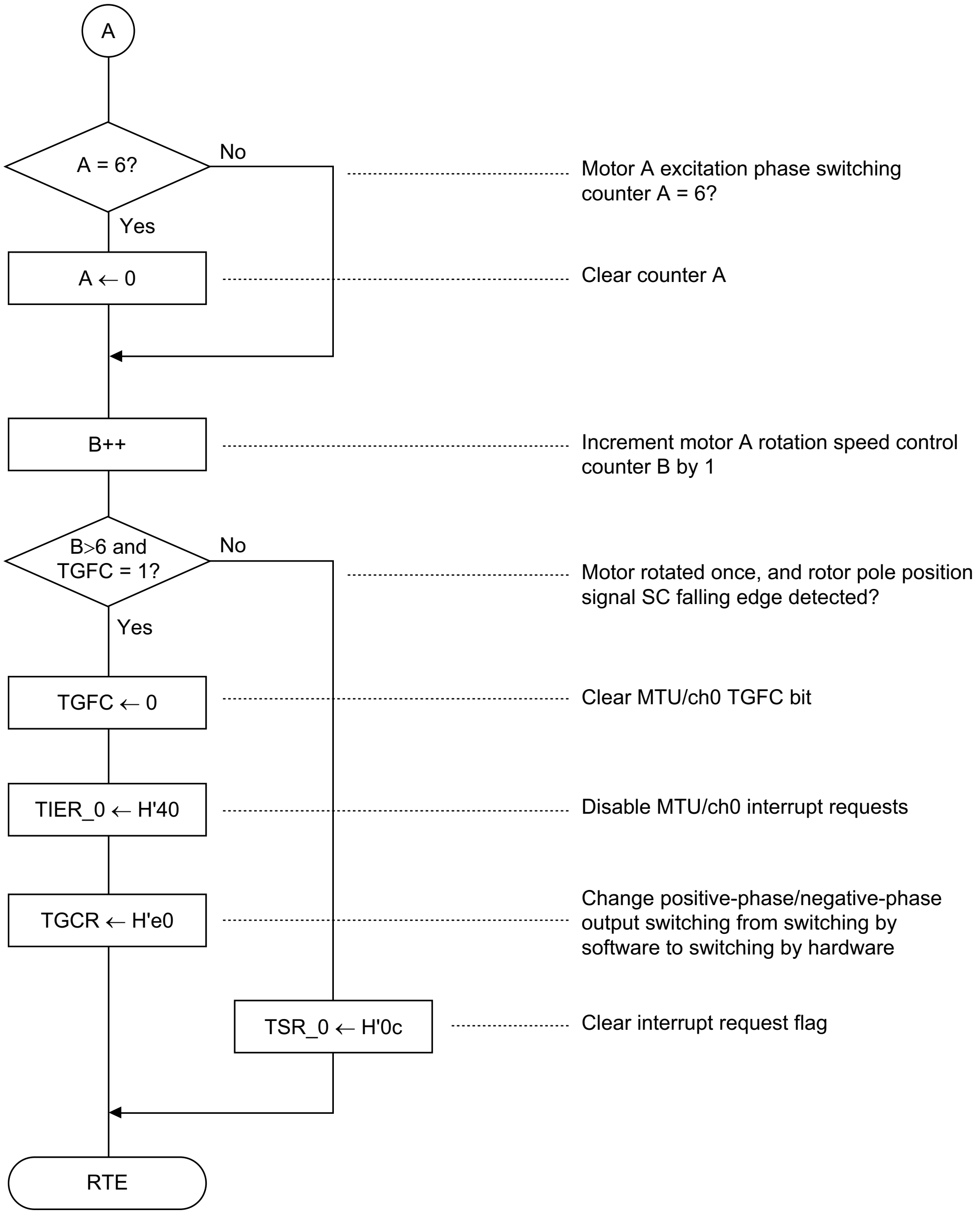
Motor A excitation phase switching counter A = 4?

Set 240° to 300° excitation data

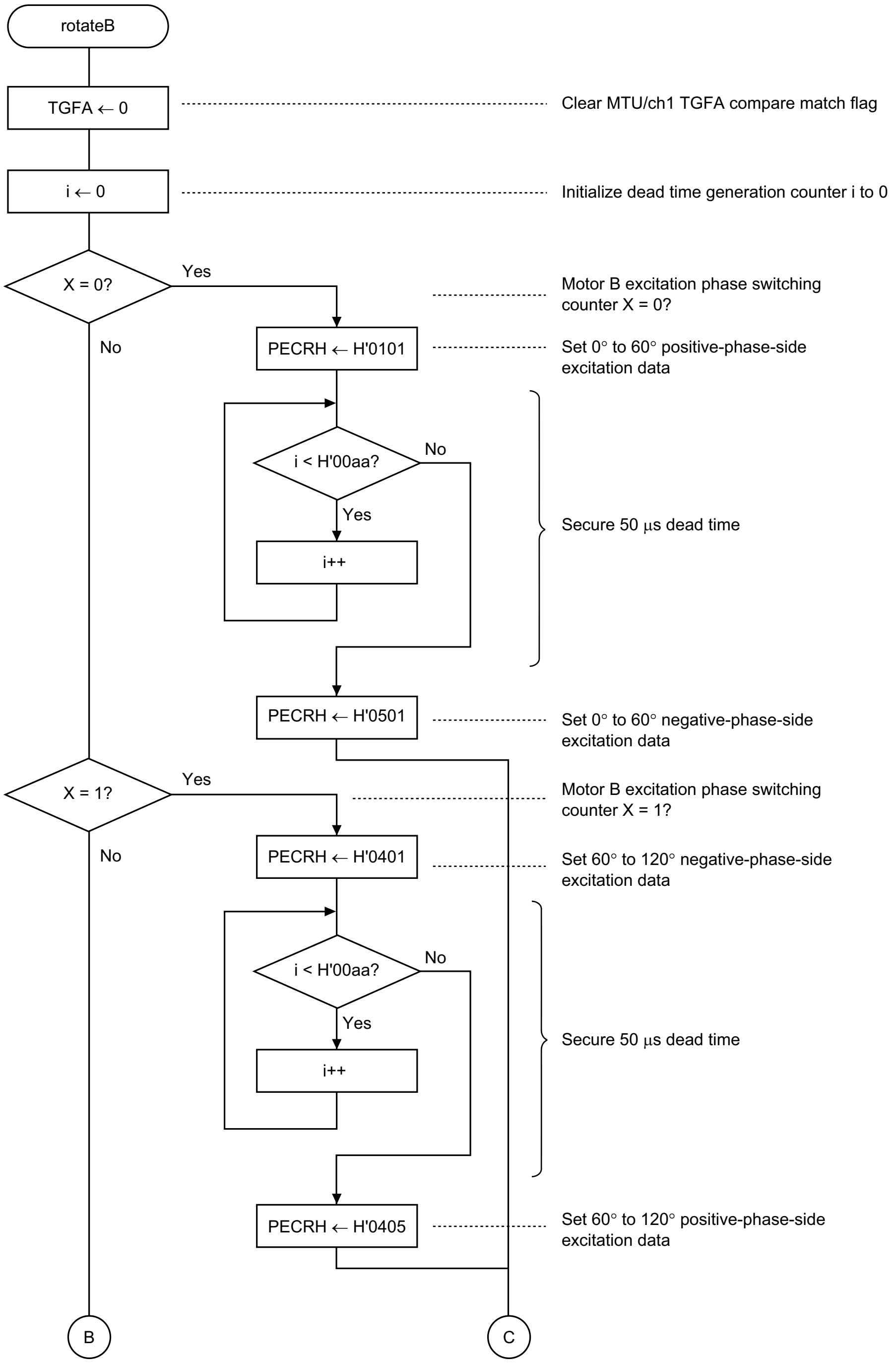
Motor A excitation phase switching counter A = 5?

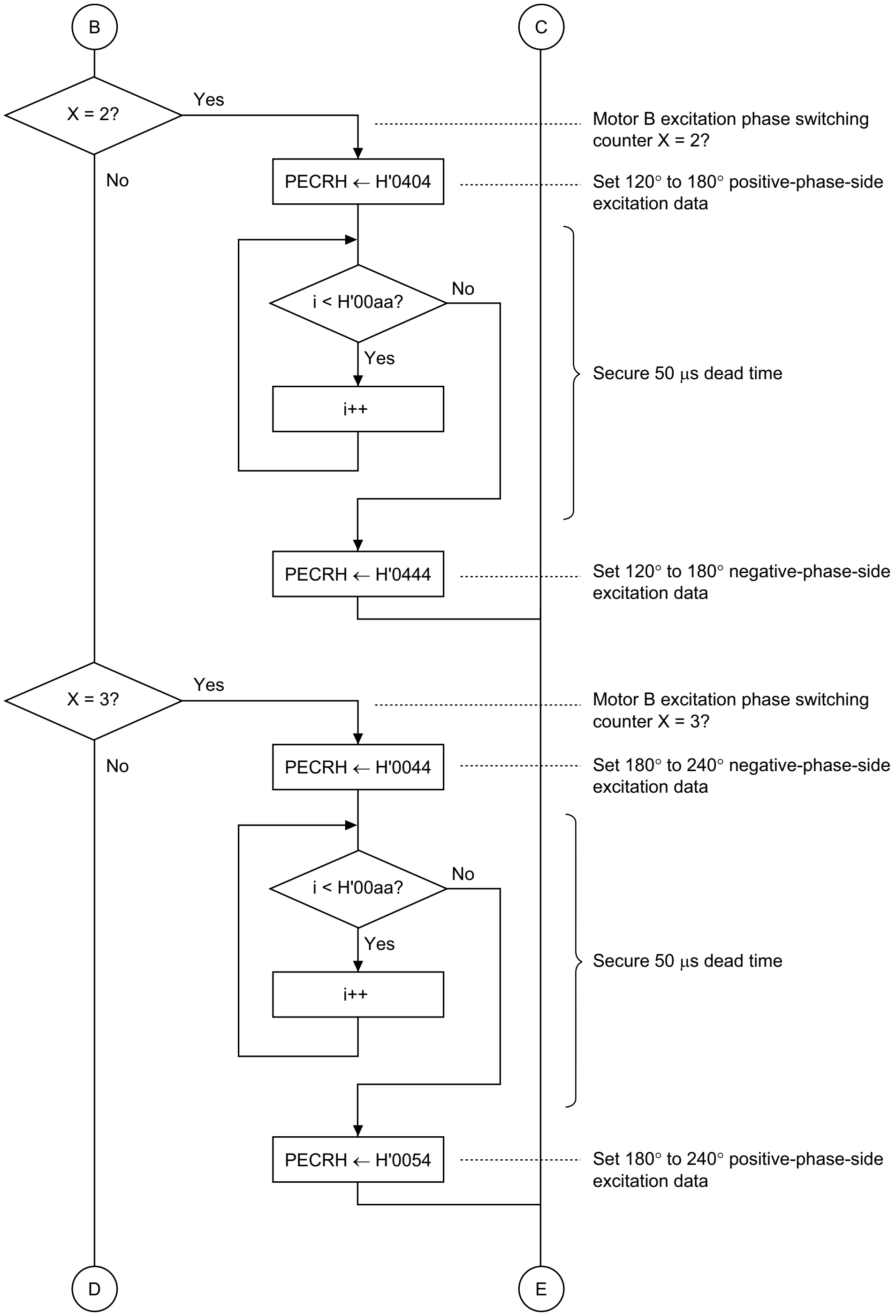
Set 300° to 360° excitation data

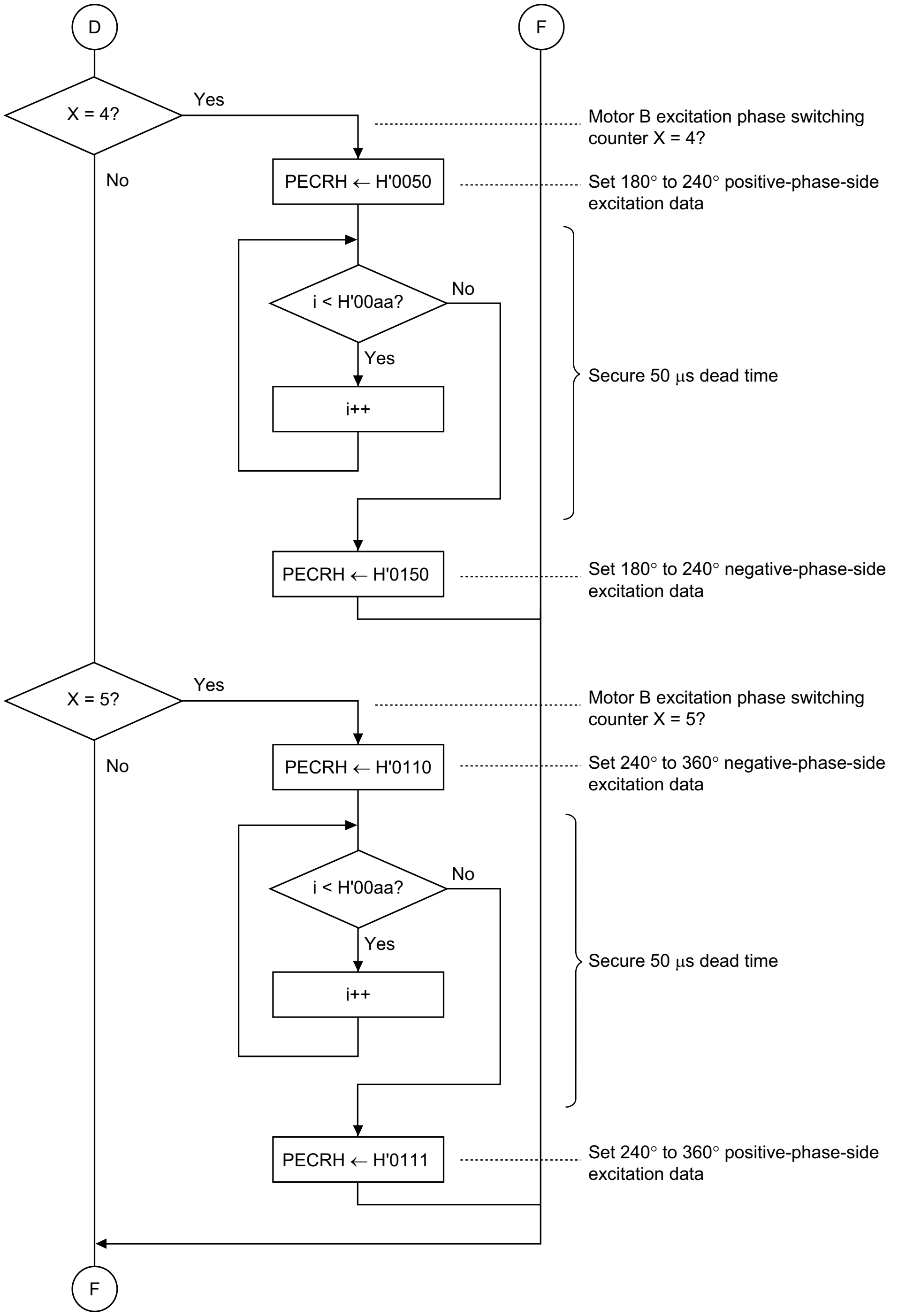
Increment motor A excitation phase switching counter A by 1

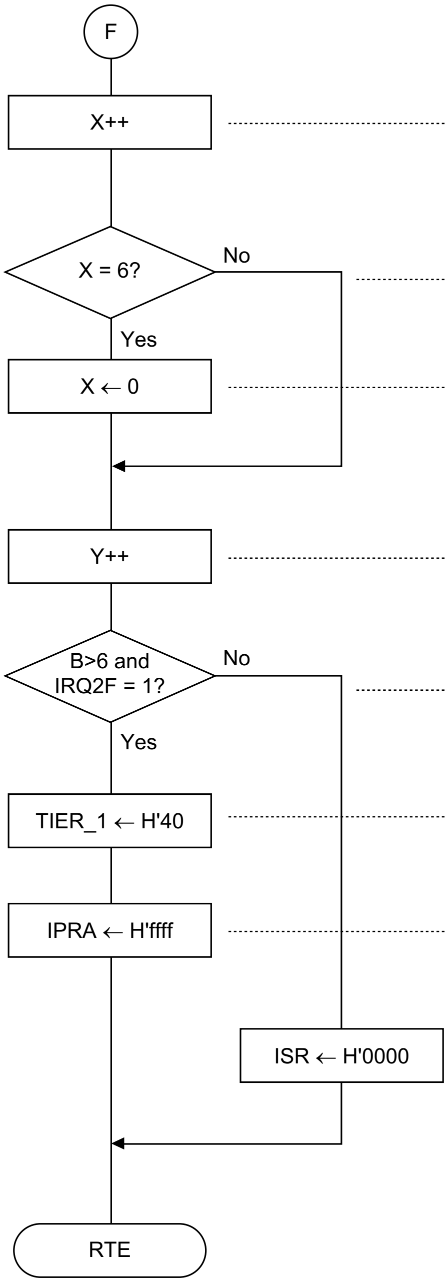

A
A = 6?
No
Yes
A ← 0
B++
B>6 and TGFC = 1?
No
Yes
TGFC ← 0
TIER_0 ← H'40
TGCR ← H'e0
TSR_0 ← H'0c
RTE
Motor A excitation phase switching counter A = 6?
Clear counter A
Increment motor A rotation speed control counter B by 1
Motor rotated once, and rotor pole position signal SC falling edge detected?
Clear MTU/ch0 TGFC bit
Disable MTU/ch0 interrupt requests
Change positive-phase/negative-phase output switching from switching by software to switching by hardware
Clear interrupt request flag

(4) MTU ch1 interrupt handling


rotateB
TGFA ← 0
Clear MTU/ch1 TGFA compare match flag
i ← 0
Initialize dead time generation counter i to 0
X = 0?
Yes
No
Motor B excitation phase switching counter X = 0?
PECRH ← H'0101
Set 0° to 60° positive-phase-side excitation data
i < H'00aa?
No
Yes
i++
Secure 50 µs dead time
PECRH ← H'0501
Set 0° to 60° negative-phase-side excitation data
X = 1?
Yes
No
Motor B excitation phase switching counter X = 1?
PECRH ← H'0401
Set 60° to 120° negative-phase-side excitation data
i < H'00aa?
No
Yes
i++
Secure 50 µs dead time
PECRH ← H'0405
Set 60° to 120° positive-phase-side excitation data
B
C


B
C
X = 2?
Yes
No
PECRH ← H'0404
i < H'00aa?
No
Yes
i++
PECRH ← H'0444
X = 3?
Yes
No
PECRH ← H'0044
i < H'00aa?
No
Yes
i++
PECRH ← H'0054
D
E
Motor B excitation phase switching counter X = 2?
Set 120° to 180° positive-phase-side excitation data
Secure 50 µs dead time
Set 120° to 180° negative-phase-side excitation data
Motor B excitation phase switching counter X = 3?
Set 180° to 240° negative-phase-side excitation data
Secure 50 µs dead time
Set 180° to 240° positive-phase-side excitation data


D
F
X = 4?
Yes
No
PECRH ← H'0050
i < H'00aa?
No
Yes
i++
PECRH ← H'0150
X = 5?
Yes
No
PECRH ← H'0110
i < H'00aa?
No
Yes
i++
PECRH ← H'0111
F
Motor B excitation phase switching counter X = 4?
Set 180° to 240° positive-phase-side excitation data
Secure 50 µs dead time
Set 180° to 240° negative-phase-side excitation data
Motor B excitation phase switching counter X = 5?
Set 240° to 360° negative-phase-side excitation data
Secure 50 µs dead time
Set 240° to 360° positive-phase-side excitation data



Increment motor B excitation phase switching counter by 1

Motor B excitation phase switching counter X = 6?

Clear counter X

Increment motor A rotation speed control counter Y by 1

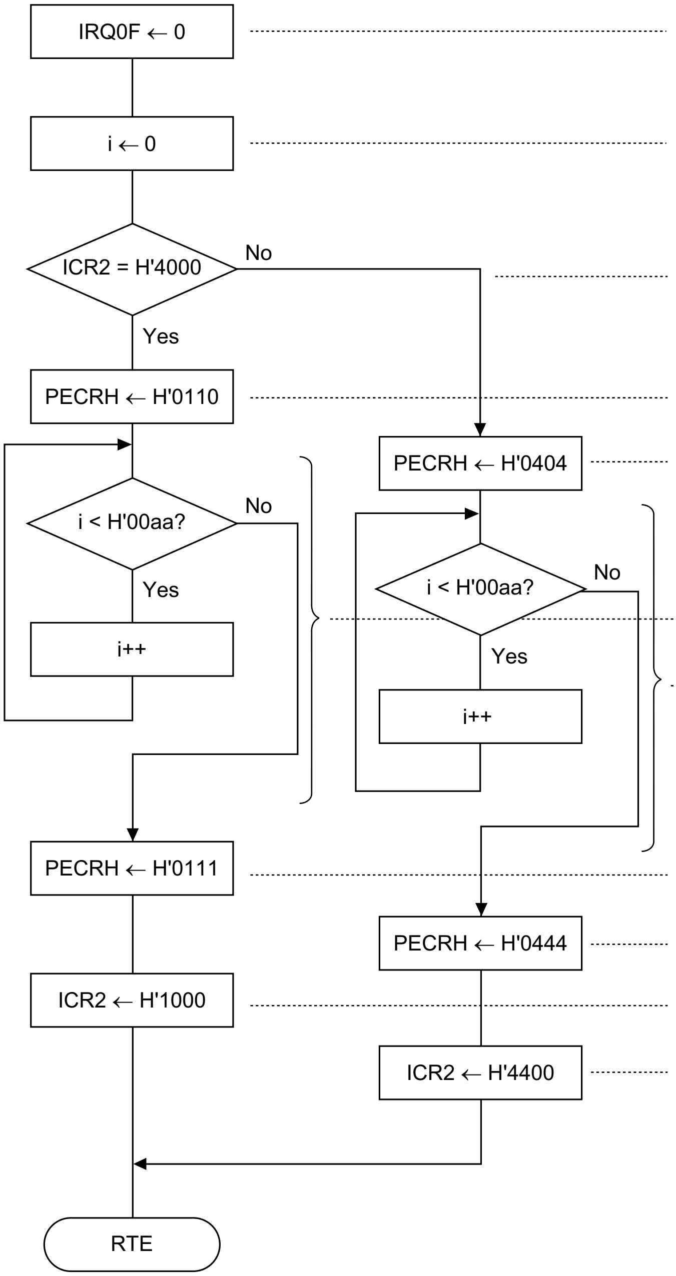
Motor rotated once, and rotor position signal HW falling edge detected?

Disable MTU/ch1 interrupt requests

Set 15 as IRQ external interrupt priority level, and change motor B excitation phase switching to switching control by rotor position signal

Clear interrupt request flags

(5) Motor B rotor pole position signal (HU) driven rotation field switching routine



Clear IRQ0F bit indicating external interrupt generation by HU edge detection

Initialize dead time generation counter i to 0

Input capture on HU rising edge?

Set 300° to 360° negative-phase-side excitation data

Set 120° to 180° negative-phase-side excitation data

Secure 50 µs dead time

Secure 50 µs dead time

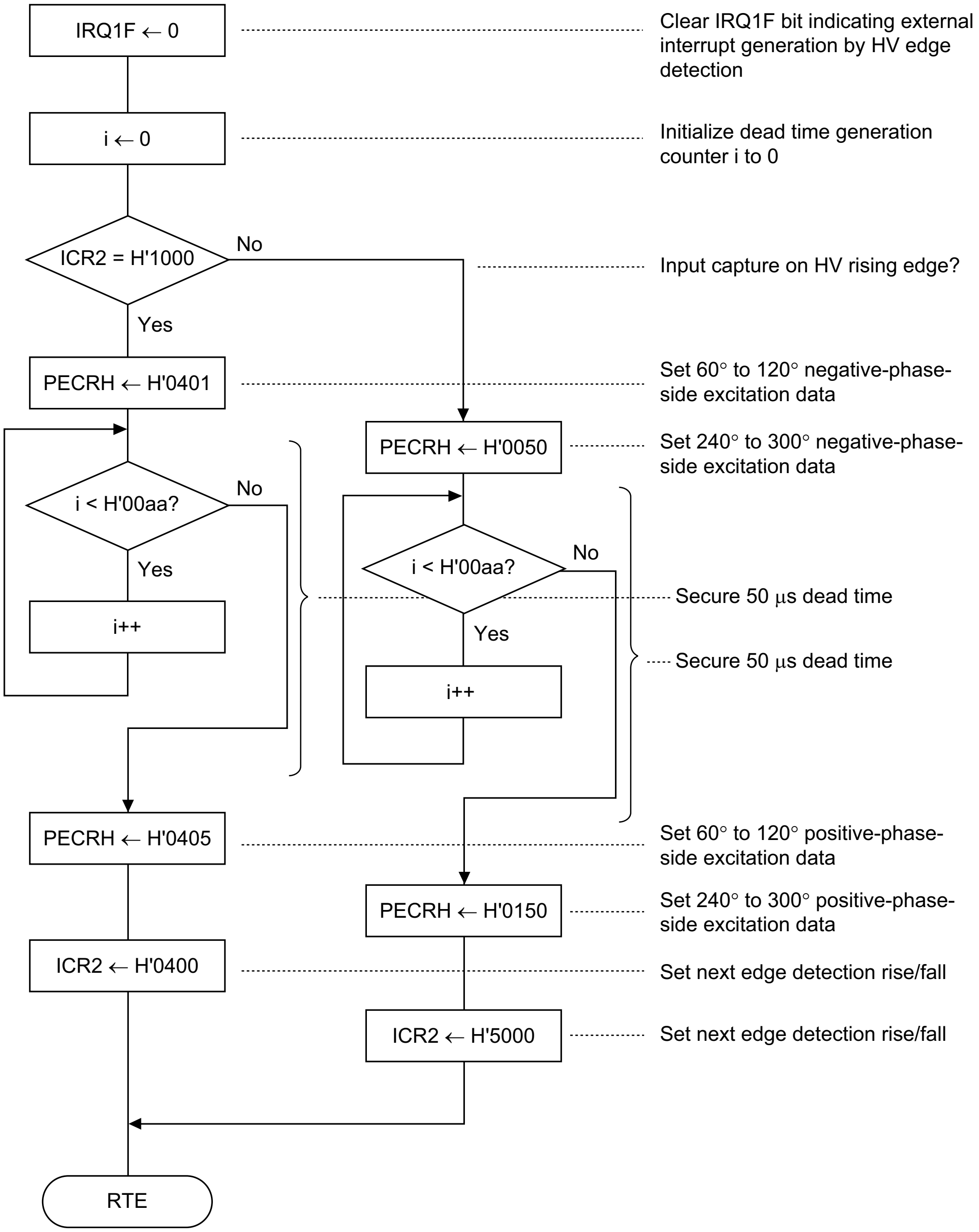
Set 300° to 360° positive-phase-side excitation data

Set 120° to 180° positive-phase-side excitation data

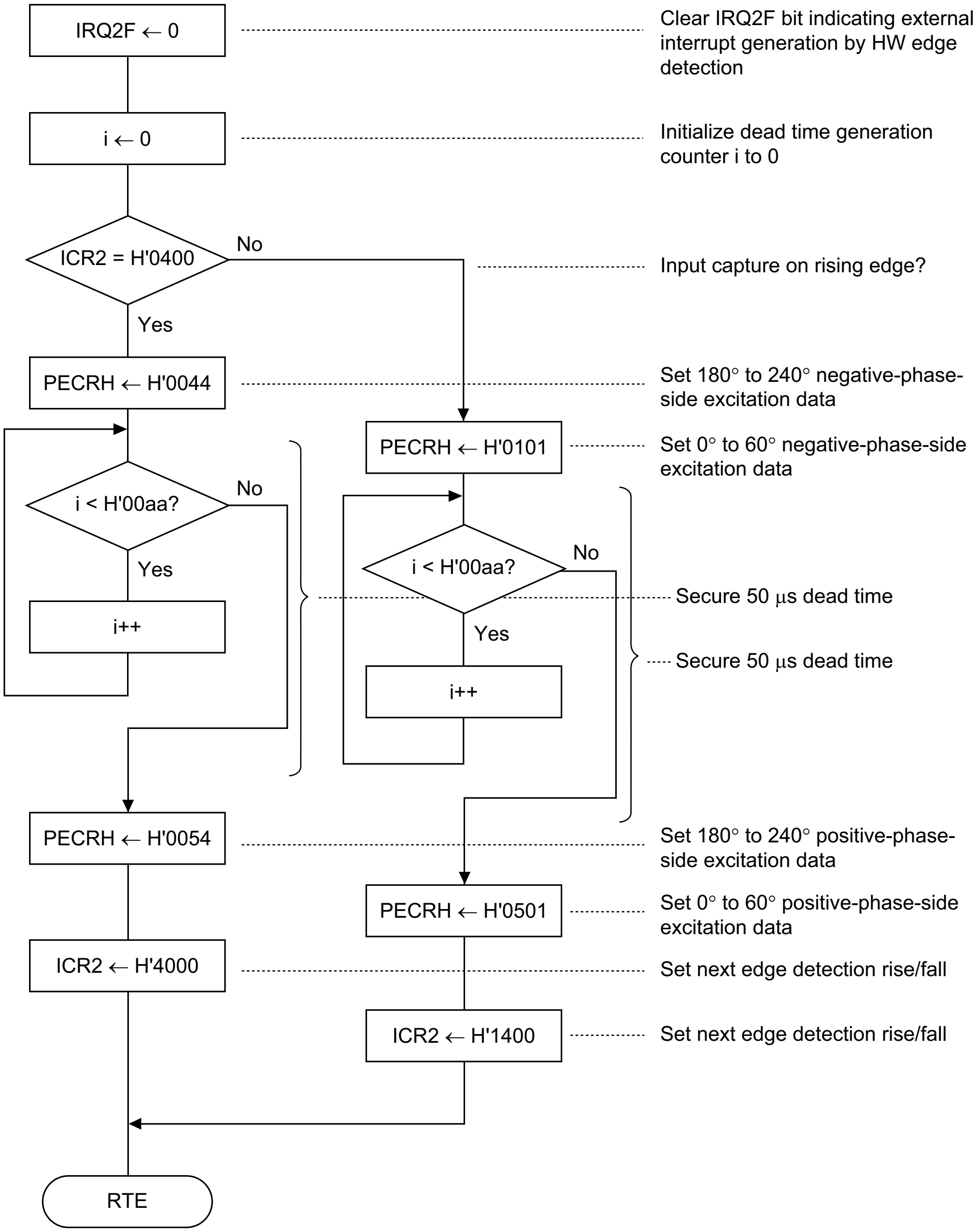
Set next edge detection rise/fall

Set next edge detection rise/fall

(6) Motor B rotor pole position signal (HV) driven rotation field switching routine


IRQ1F ← 0
Clear IRQ1F bit indicating external interrupt generation by HV edge detection
i ← 0
Initialize dead time generation counter i to 0
ICR2 = H'1000
No
Yes
Input capture on HV rising edge?
PECRH ← H'0401
Set 60° to 120° negative-phase-side excitation data
PECRH ← H'0050
Set 240° to 300° negative-phase-side excitation data
i < H'00aa?
No
Yes
i++
i < H'00aa?
No
Yes
i++
Secure 50 μs dead time
Secure 50 μs dead time
PECRH ← H'0405
Set 60° to 120° positive-phase-side excitation data
PECRH ← H'0150
Set 240° to 300° positive-phase-side excitation data
ICR2 ← H'0400
Set next edge detection rise/fall
ICR2 ← H'5000
Set next edge detection rise/fall
RTE

(7) Motor B rotor pole position signal (HW) driven rotation field switching routine


IRQ2F ← 0
Clear IRQ2F bit indicating external interrupt generation by HW edge detection
i ← 0
Initialize dead time generation counter i to 0
ICR2 = H'0400
No
Yes
Input capture on rising edge?
PECRH ← H'0044
Set 180° to 240° negative-phase-side excitation data
PECRH ← H'0101
Set 0° to 60° negative-phase-side excitation data
i < H'00aa?
No
Yes
i++
Secure 50 μs dead time
i < H'00aa?
No
Yes
i++
Secure 50 μs dead time
PECRH ← H'0054
Set 180° to 240° positive-phase-side excitation data
PECRH ← H'0501
Set 0° to 60° positive-phase-side excitation data
ICR2 ← H'4000
Set next edge detection rise/fall
ICR2 ← H'1400
Set next edge detection rise/fall
RTE

7. Program Listing

```
/*****************************************************************************/
/*                               INCLUDE FILE                                */
/*****************************************************************************/
#include<machine.h>
#include"iodefine_7046.h"
/*****************************************************************************/
/*                                 PROTOTYPE                                 */
/*****************************************************************************/
#pragma interrupt(rotateA,rotateB,HU,HV,HW)
void main(void);
void IO_INIT(void);
/*****************************************************************************/
/*                                 RAM DEFINE                                */
/*****************************************************************************/
unsigned char A;
unsigned char B;
unsigned char X;
unsigned char Y;
unsigned short i;
/*****************************************************************************/
/*                                 MAIN PROGRAM                              */
/*****************************************************************************/
void main(void)
{
 set_imask(0xf);
P_STBY.MSTCR2.WORD = 0x92fd;    /* MTU,MMT standby mode clear */
IO_INIT();
P_INTC.ISR.WORD &= 0x0000;
P_MMT.TCNR.BIT.CST = 1;/* MMT  counter start */
P_MTU34.TSTR.BYTE = 0x43;       /* MTU ch0/1/3 counter start */
set_imask(0xe);
while(1){
         ;
        }
}


/*****************************************************************************/
/*                            I/O initialize routine                         */
/*****************************************************************************/
void IO_INIT(void)
{
A = 0x00; /* Clear counter */
B = 0x00;
X = 0x00;
Y = 0x00;
   /* MTU ch3/4 initialize */
P_MTU34.TSTR.BYTE = 0x00;       /* MTU ch0/1/3 counter stop */
P_MTU34.TCR_3.BYTE = 0x20;            /* MTU counter clock ϕ=1 */
P_MTU34.TGCR.BYTE = 0xe8;             /* Change phase by software */
P_MTU34.TCNT_3 = 0x0000;
P_MTU34.TCNT_4 = 0x0000;
```

```
P_MTU34.TGRA_3 = 0x03e8;              /* PWM period = 50µs */
P_MTU34.TGRB_3 = 0x00fa;              /* U̅ duty = 25% */
P_MTU34.TGRA_4 = 0x00fa;              /* V̅ duty = 25% */
P_MTU34.TGRB_4 = 0x00fa;              /* W̅ duty = 25% */
P_MTU34.TOCR.BYTE = 0x02;
P_MTU34.TMDR_3.BYTE = 0xc8;           /* Set reset PWM mode */
P_MTU34.TOER.BYTE = 0xff;             /* Enable MTU output */

                                      /* MTU ch0 initialize */
P_MTU0.TCR_0.BYTE = 0xc2;             /* Clear counter by compare match
                                   with TGRD_0 */

P_MTU0.TMDR_0.BYTE = 0xc0;
P_MTU0.TIER_0.BYTE = 0x48;            /* Enable TGFD interrupt */
P_MTU0.TCNT_0 = 0x0000;
P_MTU0.TGRD_0 = 0x2710;               /* Output compare period = 8ms */
P_MTU0.TIORH_0.BYTE = 0x88;           /* Input capture HV rising edge
                                         and HU falling edge */
P_MTU0.TIORL_0.BYTE = 0x09;           /* Output compare TGRD_0 input
                                   capture HW falling edge */

       /* MTU ch1 initialize */
P_MTU1.TCR_1.BYTE = 0x22;             /* Clear counter by compare match
                                         with TGRA_1 */
P_MTU1.TMDR_1.BYTE = 0xc0;
P_MTU1.TIER_1.BYTE = 0x41;            /* Enable TGIA interrupt */
P_MTU1.TCNT_1 = 0x0000;
P_MTU1.TGRA_1 = 0x30d4;               /* Output compare period = 10ms */

                                /* MMT initialize */
P_MMT.TCNR.BIT.CST = 0;               /* MMT timer counter stop */
P_MMT.MMT_TCNT = 0x0000;
P_MMT.TBRU_F = 0x007d;                /* U̅ PWM duty = 25% */
P_MMT.TBRV_F = 0x007d;                /* V̅ PWM duty = 25% */
P_MMT.TBRW_F = 0x007d;                /* W̅ PWM duty = 25% */
P_MMT.MMT_TDDR = 0x0000;
P_MMT.TPBR = 0x01f4;                  /* 1/2 PWM period = 25µs */
P_MMT.MMT_TMDR.BYTE = 0x0d;           /* MMT counter clock = φ/1 */

                                /* PFC initialize */
P_PORTE.PECRH.WORD = 0x0000;
P_PORTE.PECRL1.WORD = 0x5544;
P_PORTE.PECRL2.WORD = 0x0015;
P_PORTE.PEIORL.WORD = 0xfff8;
P_PORTE.PEIORH.WORD = 0x003f;
P_PORTE.PEDRH.WORD = 0x0007;
P_PORTB.PBCR1.WORD = 0x0000;                 /* PB function = IRQ0/1/2/3 */
P_PORTB.PBCR2.WORD = 0x0550;
P_PORTB.PBIOR.WORD = 0x0000;                 /* PB2/3/4 = input pin */

                                      /* INTC initialize */
P_INTC.ICR1.WORD = 0x00f0;            /* IRQ edge select */
P_INTC.ICR2.WORD = 0x0000;
P_INTC.IPRA.WORD = 0xeeee;
P_INTC.IPRD.WORD = 0xf0f0;
P_INTC.IPRE.WORD = 0x0000;
P_INTC.IPRF.WORD = 0x0000;
P_INTC.IPRG.WORD = 0x0000;
P_INTC.IPRH.WORD = 0x0000;
P_INTC.IPRI.WORD = 0x0000;
P_INTC.IPRJ.WORD = 0x0000;
P_INTC.IPRK.WORD = 0x0000;

}

/*****************************************************************************/
/*                  MTU ch0 interrupt routine (motorA(120°))                 */
/*****************************************************************************/
void rotateA(void)
{
P_MTU0.TSR_0.BIT.TGFD = 0;
switch(A)
{
   case 0x00:  P_MTU34.TGCR.BYTE = 0xe9;     /* U=L,V=H,W=H  U̅=L, V̅=L, W̅=H */
                                  break;

   case 0x01:  P_MTU34.TGCR.BYTE = 0xeb;     /* U=H,V=L,W=H  U̅=L, V̅=L, W̅=H */
                                  break;

   case 0x02:  P_MTU34.TGCR.BYTE = 0xea;     /* U=H,V=L,W=H  U̅=H, V̅=L, W̅=L */
                                  break;

   case 0x03:  P_MTU34.TGCR.BYTE = 0xee;     /* U=H,V=H,W=L  U̅=H, V̅=L, W̅=L */
                                  break;

   case 0x04:  P_MTU34.TGCR.BYTE = 0xec;     /* U=H,V=H,W=L  U̅=L, V̅=H, W̅=L */
                                  break;

   case 0x05:  P_MTU34.TGCR.BYTE = 0xed;     /* U=L,V=H,W=H  U̅=L, V̅=H, W̅=L */
                                  break;
}

A++;
if(A == 0x06)
 {
   A = 0x00;
 }
B++;
if((B>0x0c)&&(P_MTU0.TSR_0.BIT.TGFC == 1))
 {
  P_MTU0.TSR_0.BIT.TGFC = 0;          /* Clear TGFC flag */
  P_MTU0.TIER_0.BYTE = 0x40;          /* Disable TGIC,TGFB,TGFA,TGFD */
  P_MTU34.TGCR.BYTE = 0xe0;           /* change phase by hardware */
}
else
{
  P_MTU0.TSR_0.BYTE = 0x0c;
 }
```

```
}

/******************************************************************************/
/*                     MTU ch1 interrupt routine (motorA(180°))                 */
/******************************************************************************/
void rotateB(void)
{
  P_MTU1.TSR_1.BIT.TGFA = 0;
  i = 0x0000;
  switch(X)
{
    case 0x00:  P_PORTE.PECRH.WORD = 0x0101;
    while(i < 0x00aa)                 /* dead time = 50µs */
       {
         i++;
       }
                P_PORTE.PECRH.WORD = 0x0501; /* U=L,V=H,W=H  U̅=L, V̅=H, W̅=H */
                                      break;

    case 0x01:  P_PORTE.PECRH.WORD = 0x0401;
    while(i < 0x00aa)                 /* dead time = 50µs */
       {
         i++;
       }
                P_PORTE.PECRH.WORD = 0x0405; /* U=L,V=L,W=H  U̅=L, V̅=L, W̅=H */
                                      break;

    case 0x02:  P_PORTE.PECRH.WORD = 0x0404;
    while(i < 0x00aa)  /* dead time = 50µs */
       {
         i++;
       }
                P_PORTE.PECRH.WORD = 0x0444; /* U=H,V=L,W=H  U̅=H, V̅=L, W̅=H */
                                      break;

    case 0x03:  P_PORTE.PECRH.WORD = 0x0044;
    while(i < 0x00aa)           /* dead time = 50µs */
       {
         i++;
       }
                P_PORTE.PECRH.WORD = 0x0054; /* U=H,V=L,W=L  U̅=H, V̅=L, W̅=L */
                                      break;

    case 0x04:  P_PORTE.PECRH.WORD = 0x0050;
    while(i < 0x00aa)                        /* dead time = 50µs */
       {
         i++;
       }
                P_PORTE.PECRH.WORD = 0x0150; /* U=H,V=H,W=L  U̅=H, V̅=H, W̅=L */
                                      break;

    case 0x05:  P_PORTE.PECRH.WORD = 0x0110;
    while(i < 0x00aa)                 /* dead time = 50µs */
       {
         i++;
       }
                P_PORTE.PECRH.WORD = 0x0111; /* U=L,V=H,W=L  U̅=L, V̅=H, W̅=L */
                                      break;
}

X++;
if(X == 0x06)
 {
   X = 0x00;
}

Y++;
if((Y>0x0c)&&(P_INTC.ISR.BIT.IRQ2F==1))
 {
  P_MTU1.TIER_1.BYTE = 0x40;                 /* Disable TGIA interrupt */
  P_INTC.IPRA.WORD = 0xffff;
 }
 else
 {
  P_INTC.ISR.WORD &= 0x0000;
 }
}

/******************************************************************************/
/*                           IRQ0(HU) interrupt routine                        */
/******************************************************************************/
void HU(void)
{
P_INTC.ISR.BIT.IRQ0F = 0;             /* Clear IRQ0F interrupt flag */
i= 0x0000;                      /* Clear counter */
if(P_INTC.ICR2.WORD & 0x4000)                /* If HU = rising edge */
 {
  P_PORTE.PECRH.WORD = 0x0110;
  while(i<0x00aa)                     /* dead time = 50µs */
    {
       i++;
    }
  P_PORTE.PECRH.WORD = 0x0111;  /* U=L,V=H,W=L   U̅=L, V̅=H, W̅=L */
  P_INTC.ICR2.WORD = 0x1000;
}
else
 {
  P_PORTE.PECRH.WORD = 0x0404;
  while(i<0x00aa)                     /* dead time = 50µs */
    {
       i++;
    }
  P_PORTE.PECRH.WORD = 0x0444;        /* U=H,V=L,W=H   U̅=H, V̅=L, W̅=H */
  P_INTC.ICR2.WORD = 0x4400;
   }
  }
/******************************************************************************/
/*                          IRQ0(HV) interrupt routine                         */
```

```
/******************************************************************************/
void HV(void)
{
  P_INTC.ISR.BIT.IRQ1F = 0;                  /* Clear IRQ1F interrupt flag */
  i= 0x0000;                    /* Clear counter */
  if(P_INTC.ICR2.WORD & 0x1000)              /* If HV = rising edge */
  {
    P_PORTE.PECRH.WORD = 0x0401;
    while(i<0x00aa)                    /* dead time = 50µs */
      {
       i++;
      }
    P_PORTE.PECRH.WORD = 0x0405;       /* U=L,V=L,W=H   U̅=L, V̅=L, W̅=H */
    P_INTC.ICR2.WORD = 0x0400;
 }
  else
  {
    P_PORTE.PECRH.WORD = 0x0050;
    while(i<0x00aa)                    /* dead time = 50µs */
      {
       i++;
      }
    P_PORTE.PECRH.WORD = 0x0150;       /* U=H,V=H,W=L   U̅=H, V̅=H, W̅=L */
    P_INTC.ICR2.WORD = 0x5000;
     }
  }
/******************************************************************************/
/*                           IRQ0(HW) interrupt routine                       */
/******************************************************************************/
void HW(void)
{
  P_INTC.ISR.BIT.IRQ2F = 0;                  /* Clear IRQ2F interrupt flag */
  i= 0x0000;                    /* Clear counter */
  if(P_INTC.ICR2.WORD & 0x0400)              /* If HW = rising edge */
    {
      P_PORTE.PECRH.WORD = 0x0044;
      while(i<0x00aa)                  /* dead time = 50µs */
       {
         i++;
       }
      P_PORTE.PECRH.WORD = 0x0054;     /* U=H,V=L,W=L   U̅=H, V̅=L, W̅=L */
      P_INTC.ICR2.WORD = 0x4000;
    }
else
    {
      P_PORTE.PECRH.WORD = 0x0101;
      while(i<0x00aa)                  /* dead time = 50µs */
       {
         i++;
       }
      P_PORTE.PECRH.WORD = 0x0501;     /* U=LV=H,W=H  U̅=L, V̅=H, W̅=H */
      P_INTC.ICR2.WORD = 0x1400;
  }
}
```

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.