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.

To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

RENESAS
Renesas Technology Corp.

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation 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 Corporation 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 Corporation or a third party.
2. Renesas Technology Corporation 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 Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation by various means, including the Renesas Technology Corporation 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 Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation 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 Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation for further details on these materials or the products contained therein.

APPLICATION NOTE

Control of a Brushless DC Motor (H8/3687)

Introduction

The H8/3687 is used to control a brushless DC motor in the way shown in Figure 1.1.

Target Device

H8/300H Tiny Series H8/3687

Contents

Feb. 2003

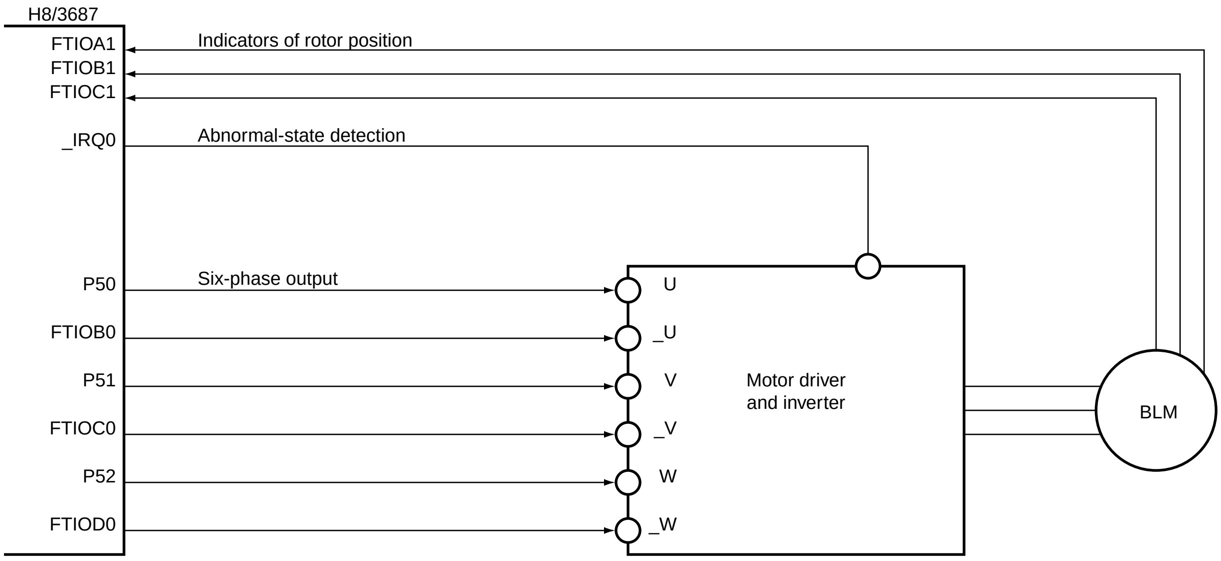
ADE-502-119

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

1. Specifications

1. The H8/3687 is used to control a brushless DC motor in the way shown in Figure 1.1.
2. The H8/3687 detects signals that indicate the positions of the rotor's magnetic poles and operates the motor by producing six PWM waveforms that provide control of the rotating magnetic field according to the positional signals from the motor. The PWM signals are output through an I/O port and timer Z.
3. The H8/3687's built-in timer generates a PWM waveform that handles chopping control for the motor.



Note:

BLM	: Brushless DC motor (120-degree type)
U	: U phase
_U	: $\overline{U}$ phase
V	: V phase
_V	: $\overline{V}$ phase
W	: W phase
_W	: $\overline{W}$ phase
FTIOCA	: Input-capture pin A
FTIOCB	: Input-capture pin B
FTIOCC	: Input-capture pin C
IRQ0	: $\overline{IRQ0}$ pin (input)
TMOV	: Output pin for timer V waveform
P50 to P52	: Port pins P50 to P52 (outputs)

Figure 1.1 Set-up for Controlling a Brushless DC Motor

2. Design

1. The PWM waveforms for motor control are generated by timer Z and output through I/O-port and timer Z pins.
2. In the initial stage of the motor's operation, the motor is started by sequential switching of the excited phase on a constant cycle and outputs the control signals from the port and timer Z.
3. After switching the excited phase 12 times, control by the CPU shifts to the procedure where control of the motor is based on the rotor-positional signals from the motor.
4. The positional signals from the motor are taken in through the input-capture terminals of timer Z (CH0) and drive the generation of interrupts.
5. These interrupts drive switching to produce a rotating magnetic field and control phase excitation through chopping.

3. Description of Functions Used

1. As shown in Figure 3.1, timer Z (input capture/output comparison), an I/O port (Port 5) and the IRQ (external interrupt) perform the functions required to control the motor.

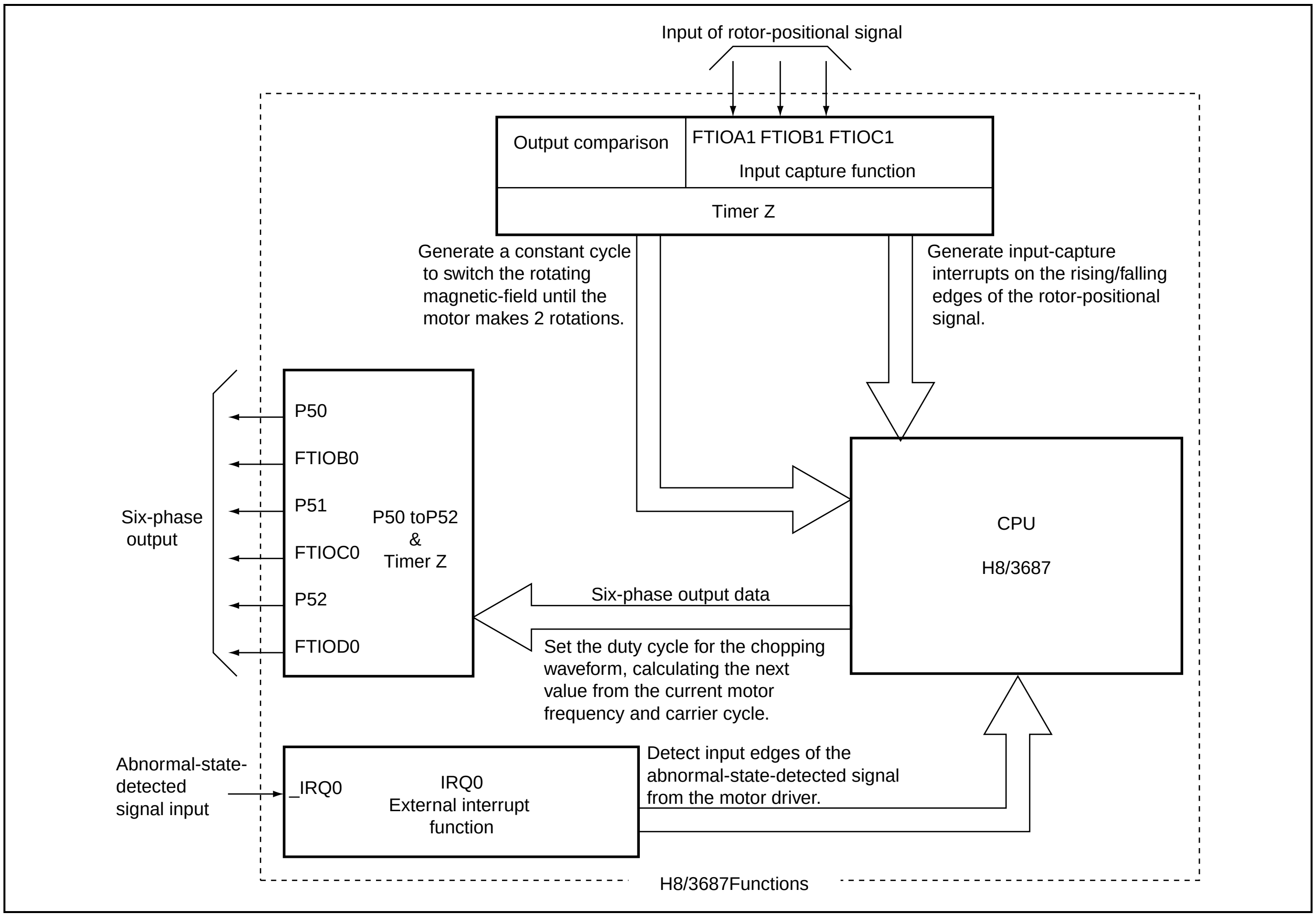

Input of rotor-positional signal
Output comparison
FTIOA1 FTIOB1 FTIOC1
Input capture function
Timer Z
Generate a constant cycle to switch the rotating magnetic-field until the motor makes 2 rotations.
Generate input-capture interrupts on the rising/falling edges of the rotor-positional signal.
Six-phase output
P50
FTIOB0
P51
FTIOC0
P52
FTIOD0
P50 toP52 & Timer Z
CPU
H8/3687
Six-phase output data
Set the duty cycle for the chopping waveform, calculating the next value from the current motor frequency and carrier cycle.
Abnormal-state-detected signal input
_IRQ0
IRQ0 External interrupt function
Detect input edges of the abnormal-state-detected signal from the motor driver.
H8/3687Functions

Figure 3.1 Block Diagram of the Configuration for Controlling a Brushless DC Motor

The tasks performed by the H8/3687 functional blocks are outlined below.

- Timer Z (CH1) input capture: generates an interrupt request for the CPU on the rising and falling edges of the rotor-positional signal.
- Timer Z (CH1) output-comparison: generates an interrupt request for the CPU at the end of each magnetic-field rotation switching cycle (period that corresponds to 60 degrees at a given motor rotation frequency) until the motor has made two rotations.
- Timer Z (CH0) PWM-mode function: Chopping waveform is generated to perform chopping-control when the driver transistor turns on, and the data for the three negative-phase signals is output to the driver from pins FTIOCB0-FTIOCD0.
- Port 5 function: Outputs the data for the three positive phases to the driver.
- IRQ external interrupt function: Stops the motor on detection of the abnormal-state-detected signal from the driver.

2. Detailed descriptions of the individual functions used are given in the following pages.

1) Two functions of Timer Z (CH1) are used: input capture, to generate interrupts in response to the rising and falling edges of the rotor-positional signal, and output comparison, to measure the magnetic-field rotation switching cycle in the initial stage of motor control.

The following functions are used in both the input capture and the output–comparison function of timer Z.

- The system clock (ϕ) is the 16-MHz standard clock that drives the CPU and peripheral functions.
- Timer Counter 0/1 (TCNT0/1) of timer Z is a 16-bit readable/writable up-counter, the clock source can be selected from among a total of five signals: the four obtained by dividing the system clock signal by 1, 2, 4, or 8, and an external clock. The input clock of TCNT1 is set by TCRT1. In this sample task, the system clock ϕ is selected.
- Timer Control Register 0/1 (TCR0/1) is an 8-bit readable/writable register used to select the factor that drives clearing of timer counter 0/1, the sense of the driving edge when an external clock is selected, and the clock that drives counting by timer counter 1.
- Timer I/O Control Register 0/1 (TIOR0/1) is an 8-bit readable/writable register that controls the general register (GR) of general timer Z. When operation as an output-comparison register is selected, the value that is output on a compare match is to be specified. When operation as an input-capture register is selected, the input edge is to be specified.
- Timer Status Register 0/1 (TSR0/1) is an 8-bit readable/writable register that indicates the states of the input-capture and output-comparison register and overflow of the timer.
- PWM Mode Output Level Control Register (POCR) is an 8-bit readable/writable register that is used to select the active level for the PWM mode. In this sample task, active-high is selected as the active output level for FTIOB1-FTIOD1.
- Timer PWM Mode Register (TPMR) is an 8-bit readable/writable register that selects the operating mode (normal or PWM), in the output of a waveform generated by compare-matching on FTIOB – FTIOD. In this sample task, FTIOB0 – FTIOD0 are placed in the PWM mode.
- Timer Output Enable Register (TOER) is an 8-bit readable/writable register that is used to enable and disable output through channel 0 and 1. In this sample task, FTIOB0 - FTIOD0 are set as output pins.
- Timer Output Control Register (TOCR) is an 8-bit readable/writable register that is used to set up initial outputs that are valid until the first compare-match occurs. In this sample task, the FTIOB0 – FTIOD0 pins are set to output "0".

- Timer Interrupt Enable Register (TIER), an 8-bit readable/writable register, controls enabling/disabling of each of the interrupt-request signals IMFA to IMFD and OVF. In this sample task, an interrupt request (IMIA1) by IMFA1 is enabled.
- Timer Start Register (TSTR) of timer Z is an 8-bit readable/writable register that starts/stops the counter operation for channels 0 and 1. In this sample task, this register starts counting by TCNT0/1.

2) Timer Z (CH1) input capture function provides a way to generate interrupts on detecting the rising and falling edges of the rotor-positional signal. Figure 3.2 is a block diagram of how the input-capture function is used to generate interrupt requests in response to edges of the rotor-positional signal.

- The rotor-positional signal is input through input-capture pins A, B, and C (FTIOA1, FTIOB1, FTIOC1).
- Input capture registers A, B, and C (GRA1, GRB1, GRC1) are 16-bit readable/writable registers. When an edge is detected in input signal A, B or C, respectively, the value of TCNT1 is transferred to GRA1, GRB1 or GRC1, and IMFA1, IMFB1 or IMFC1 of TSR1 is set to "1". In this case, when the value of the corresponding IMIAE, IMIBE or IMICE bit in TIER is "1", an interrupt request is sent to the CPU.

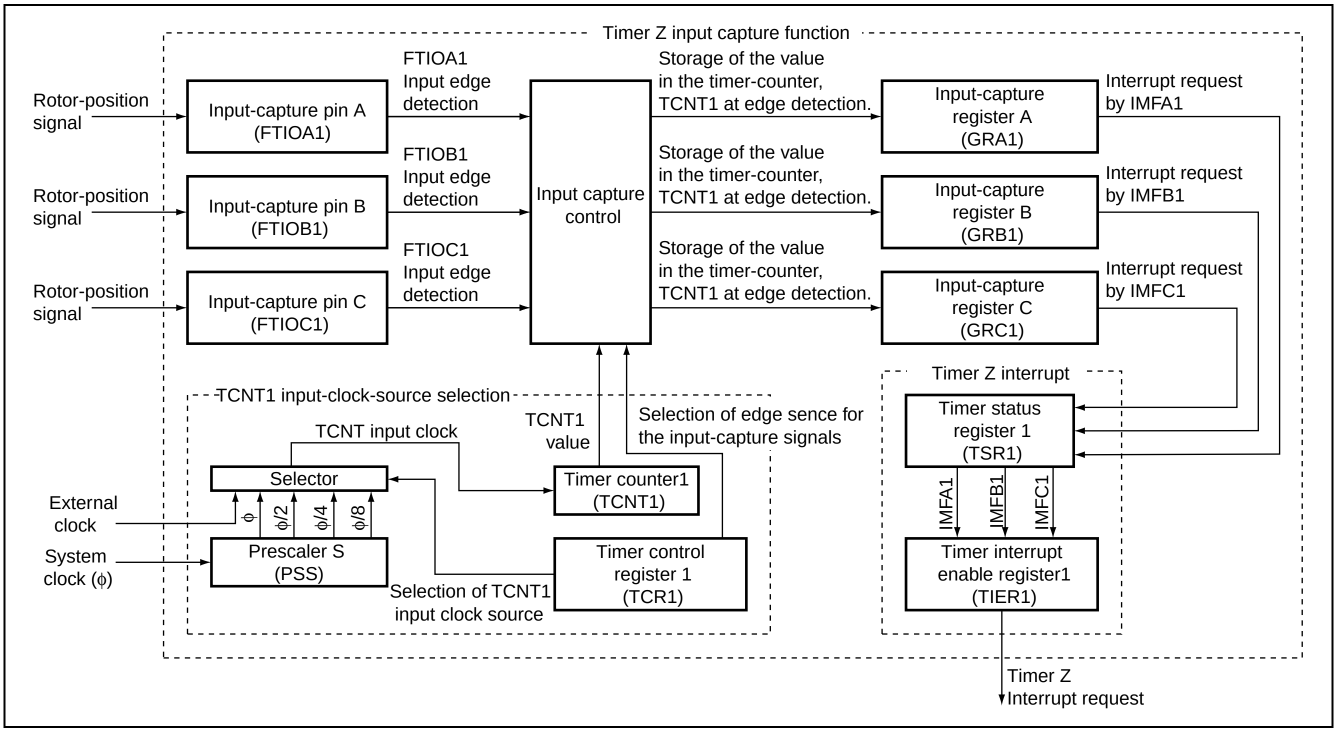

Timer Z input capture function
Rotor-position signal
Input-capture pin A (FTIOA1)
FTIOA1 Input edge detection
Rotor-position signal
Input-capture pin B (FTIOB1)
FTIOB1 Input edge detection
Rotor-position signal
Input-capture pin C (FTIOC1)
FTIOC1 Input edge detection
Input capture control
Storage of the value in the timer-counter, TCNT1 at edge detection.
Storage of the value in the timer-counter, TCNT1 at edge detection.
Storage of the value in the timer-counter, TCNT1 at edge detection.
Input-capture register A (GRA1)
Input-capture register B (GRB1)
Input-capture register C (GRC1)
Interrupt request by IMFA1
Interrupt request by IMFB1
Interrupt request by IMFC1
TCNT1 input-clock-source selection
TCNT input clock
TCNT1 value
Selection of edge sence for the input-capture signals
Selector
External clock
System clock (ϕ)
ϕ
ϕ/2
ϕ/4
ϕ/8
Prescaler S (PSS)
Timer counter1 (TCNT1)
Timer control register 1 (TCR1)
Selection of TCNT1 input clock source
Timer Z interrupt
Timer status register 1 (TSR1)
IMFA1
IMFB1
IMFC1
Timer interrupt enable register1 (TIER1)
Timer Z Interrupt request

Figure 3.2 Block Diagram of Interrupt Generation by the Input-Capture Function in Response to Edges of the Rotor-Positional Signal

3) The timer Z (CH1) output-comparison function is used to issue an interrupt request for the CPU at the end of every magnetic-field rotation switching cycle (i.e., at the intervals that correspond to 60 degrees at the motor's frequency of rotation) until the motor has gone through ten rotation. Figure 3.3 is a block diagram of the control of these interrupt requests by timer Z's output-comparison function. The operation of the blocks is described below.

— Output–compare register A1 (GRA1) is a 16-bit readable/writable register. The content of GRA1 is constantly compared with TCNT1. When the two values match, IMFA1 in TSR1 is set to "1". In this case, if the OCIAE bit of TIER is set to "1", an interrupt request is sent to the CPU.

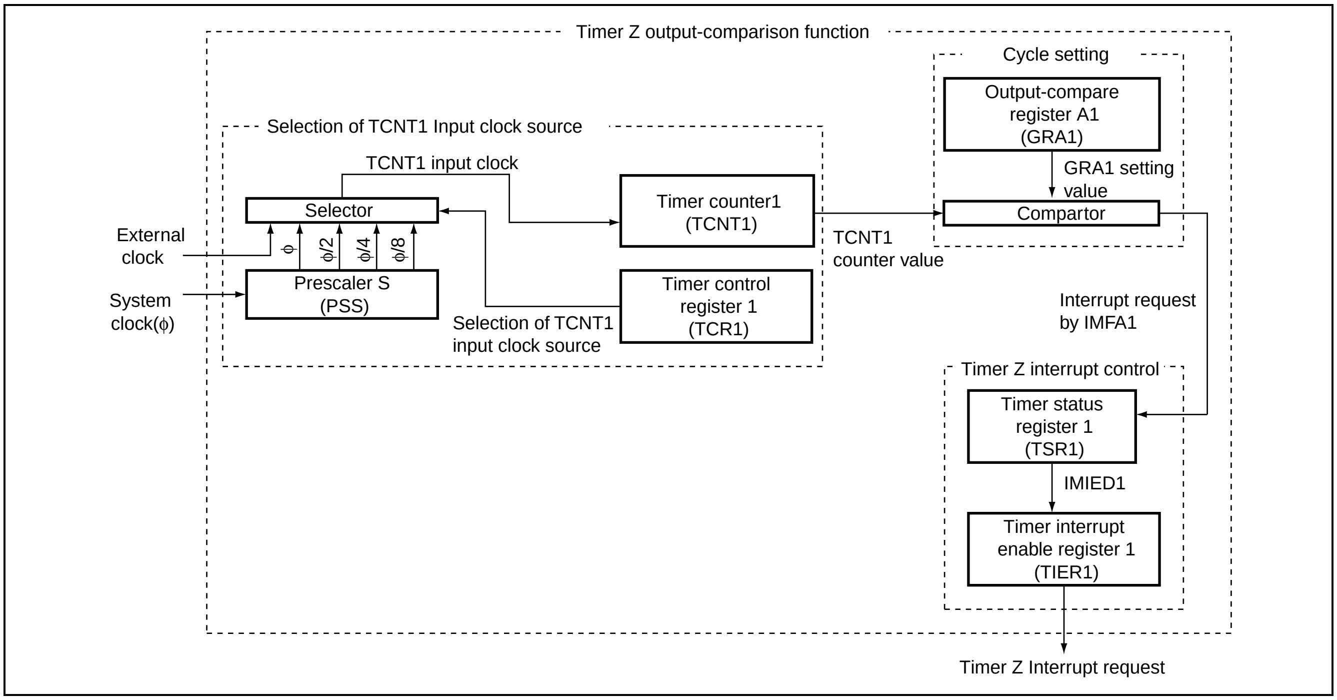

Timer Z output-comparison function
Cycle setting
Output-compare register A1 (GRA1)
GRA1 setting value
Compartor
Selection of TCNT1 Input clock source
TCNT1 input clock
Selector
Timer counter1 (TCNT1)
External clock
φ
φ/2
φ/4
φ/8
Prescaler S (PSS)
System clock(φ)
Timer control register 1 (TCR1)
Selection of TCNT1 input clock source
TCNT1 counter value
Interrupt request by IMFA1
Timer Z interrupt control
Timer status register 1 (TSR1)
IMIED1
Timer interrupt enable register 1 (TIER1)
Timer Z Interrupt request

Figure 3.3 Block Diagram of Interrupt Generation Per Magnetic-Field Switching Cycle by Timer Z's Output-Comparison Function

4) Timer Z (CH0) is placed in PWM mode and its compare–match function is used to generate the chopping waveform for output on the FTIOB0 to FTIOD0 pins. Figure 3.4 is a block diagram of the control of chopping-waveform output by timer Z's compare–match function in PWM mode. The operation of the blocks is described below.

— Timer Counter 0 (TCNT0), a 16-bit readable/writable up-counter, is incremented by either an internal or external clock input. The clock-source selection is made in bits TPSC2 to TPSC0 of the TCR0 register. The value in TCNT0 is always readable/writable from the CPU, and is cleared by the input of an external reset signal or by compare–match signal (GRA0, GRB0, GRC0, or GRD0). The clearing signal is selected by bits CCLR2 to CCLR0 of TSR0.

— General Register (GRA0) is an 8-bit readable/writable register. The value in GRA0 is constantly compared with TCNT0, and when the values match, IMIFA of TSR is set to "1". When IMIEA of TIER0 is "1", an interrupt request is sent to the CPU.

— Timer Control Register 0 (TCR0), an 8-bit readable/writable register, is used to select the input clock, designate the signal that clears the TCNT0 and enable interrupt requests. In this sample task, interrupt requests by IMFA are enabled and other interrupt requests are disabled. Compare–match IMFA is selected to drive the clearing of TCNT0. The input clock source for TCNT1 is selected by the TPSC2 to TPSC0 bits of TCR1.

— Timer Status Register 0 (TSR0), an 8-bit register, is used to set the compare–match flag and control output in response to a compare–match. In this sample task, output of "0" is selected for compare–match B and output of "1" is selected for compare–match A.

— Timer PWM Mode Register (TPMR) is an 8-bit readable/writable register that is used to select compare–match output as the operating mode of the … pins (FTIOB0 to FTIOD0 and FTIOB1 to FTIOD1).

— Timer Output Master Enable Register (TOER) is an 8-bit readable/writable register used to enable/disable the output setting of channels 0 and 1.

— Timer Output Control Register (TOCR) is an 8-bit readable/writable register used to set the initial output until the first compare–match occurs.

— The chopping waveform is output by timer Z (CH0) output pins (FTIOB0, FTIOC0 and FTIOD0).

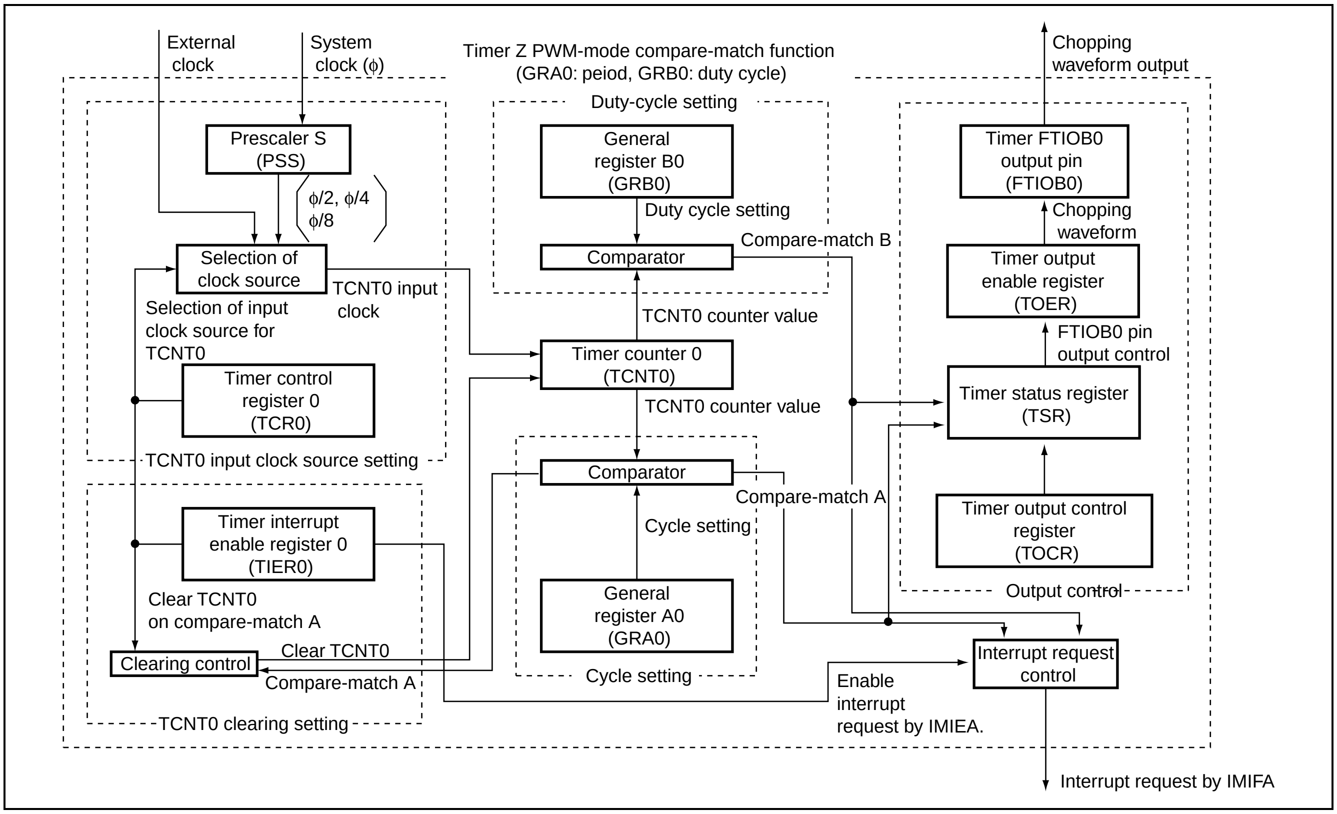

External clock
System clock (ϕ)
Timer Z PWM-mode compare-match function
(GRA0: peiod, GRB0: duty cycle)
Chopping waveform output
Prescaler S (PSS)
ϕ/2, ϕ/4 ϕ/8
Selection of clock source
TCNT0 input clock
Selection of input clock source for TCNT0
Timer control register 0 (TCR0)
TCNT0 input clock source setting
Duty-cycle setting
General register B0 (GRB0)
Duty cycle setting
Comparator
Compare-match B
TCNT0 counter value
Timer counter 0 (TCNT0)
TCNT0 counter value
Comparator
Compare-match A
Cycle setting
General register A0 (GRA0)
Cycle setting
Timer FTIOB0 output pin (FTIOB0)
Chopping waveform
Timer output enable register (TOER)
FTIOB0 pin output control
Timer status register (TSR)
Timer output control register (TOCR)
Output control
Timer interrupt enable register 0 (TIER0)
Clear TCNT0 on compare-match A
Clearing control
Clear TCNT0
Compare-match A
TCNT0 clearing setting
Enable interrupt request by IMIEA.
Interrupt request control
Interrupt request by IMIFA

Figure 3.4 Block Diagram of Chopping Waveform Output Using the PWM-Mode Compare-Match Function of Timer Z (CH0)

5) Port 5 function is used as the output for the positive-phase components of the six-phase output. It is also used as an input to detect the state of the motor-stop switch. Figure 3.5 is a block diagram of six-phase output and detection of the motor stop signal by port 5. The functions of the blocks are described below.

— Port 50 pin (P50) functions as the output pin for the U phase of the motor driver.

— Port 51 pin (P51) functions as the output pin for the V phase of the motor driver.

— Port 52 pin (P52) functions as the output pin for the W phase of the motor driver.

— Port data register 5 (PDR5) is an 8-bit register that stores the data for each pin, P50 to P57, of port 5. When the value in a bit of the port 5 control register (PCR5) is "1", the value read from the data bit is that of the corresponding bit in PDR5. Therefore, PDR5 is not affected by the pin states. When the value in PCR5 is "0", the pin is an input and the value read is its state.

— Port Control Register (PCR5) provides bit-by-bit control of the input/output state of each of pins P50 to P57 on port 5. In this sample task, pins P50 to P52 are set as output pins and P53 to P57 are set as input pins.

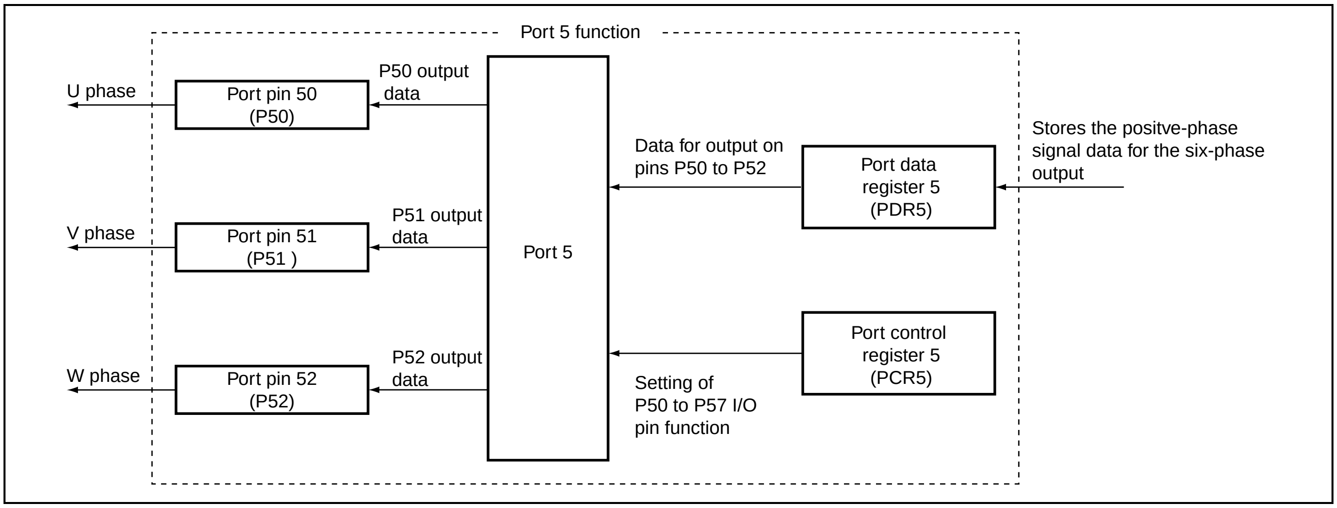

Port 5 function
U phase
Port pin 50 (P50)
P50 output data
V phase
Port pin 51 (P51)
P51 output data
W phase
Port pin 52 (P52)
P52 output data
Port 5
Data for output on pins P50 to P52
Port data register 5 (PDR5)
Stores the positve-phase signal data for the six-phase output
Port control register 5 (PCR5)
Setting of P50 to P57 I/O pin function

Figure 3.5 Block Diagram of Six-Phase Output on Port 5

6) The IRQ external interrupt line is used to detect the abnormal-state-detected signal. Figure 3.6 is a block diagram of how the IRQ external interrupt function is used to handle this signal. The operation of the blocks is described below.

— The _IRQ0 pin is used as the input pin for the externally generated abnormal-state-detected signal.

— An IRQ0 interrupt is requested in response to the input of an edge on the IRQ0 pin. Rising or falling may be selected as the sense of the _IRQ0 pin.

— Interrupt Edge Select Register 1 (IEGR1) is an 8-bit readable/writable register used to designate rising- or falling-edge detection by the _IRQ0 pin.

— Interrupt Enable Register 1 (IENR1) is an 8-bit readable/writable register for the enabling and disabling of interrupts. In this sample task, IRQ0 interrupt requests are enabled and the direct-transition, timer A, wake-up, IRQ1, IRQ2, and IRQ3 interrupts are disabled.

— Interrupt Request Register 1 (IRR1) is an 8-bit readable/writable register of interrupt flags. When a direct-transition, timer A, or IRQ0, …, IRQ3 interrupt is generated, the corresponding flag is set to "1". Interrupt-request flags are not automatically cleared, even by acceptance of the interrupt. Write a "0" to clear any of the interrupt-request flags.

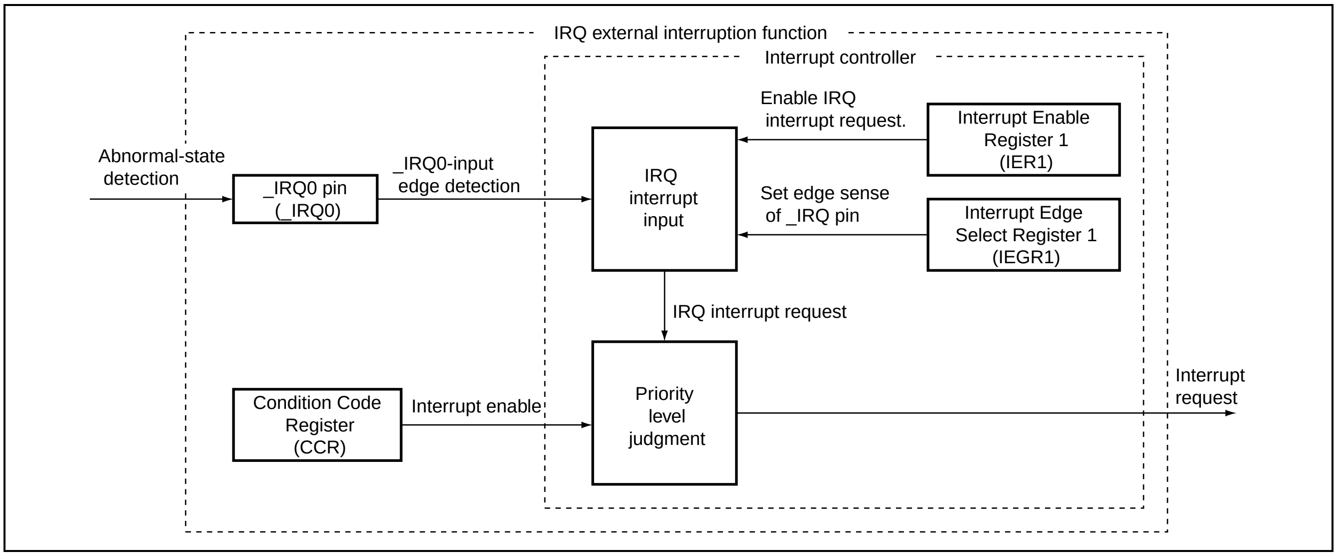

IRQ external interruption function
Interrupt controller
Enable IRQ interrupt request.
Interrupt Enable Register 1 (IER1)
Abnormal-state detection
_IRQ0 pin (_IRQ0)
_IRQ0-input edge detection
IRQ interrupt input
Set edge sense of _IRQ pin
Interrupt Edge Select Register 1 (IEGR1)
IRQ interrupt request
Condition Code Register (CCR)
Interrupt enable
Priority level judgment
Interrupt request

Figure 3.6 Block Diagram of the Use of the IRQ External-Interrupt Function to Detect Abnormal States

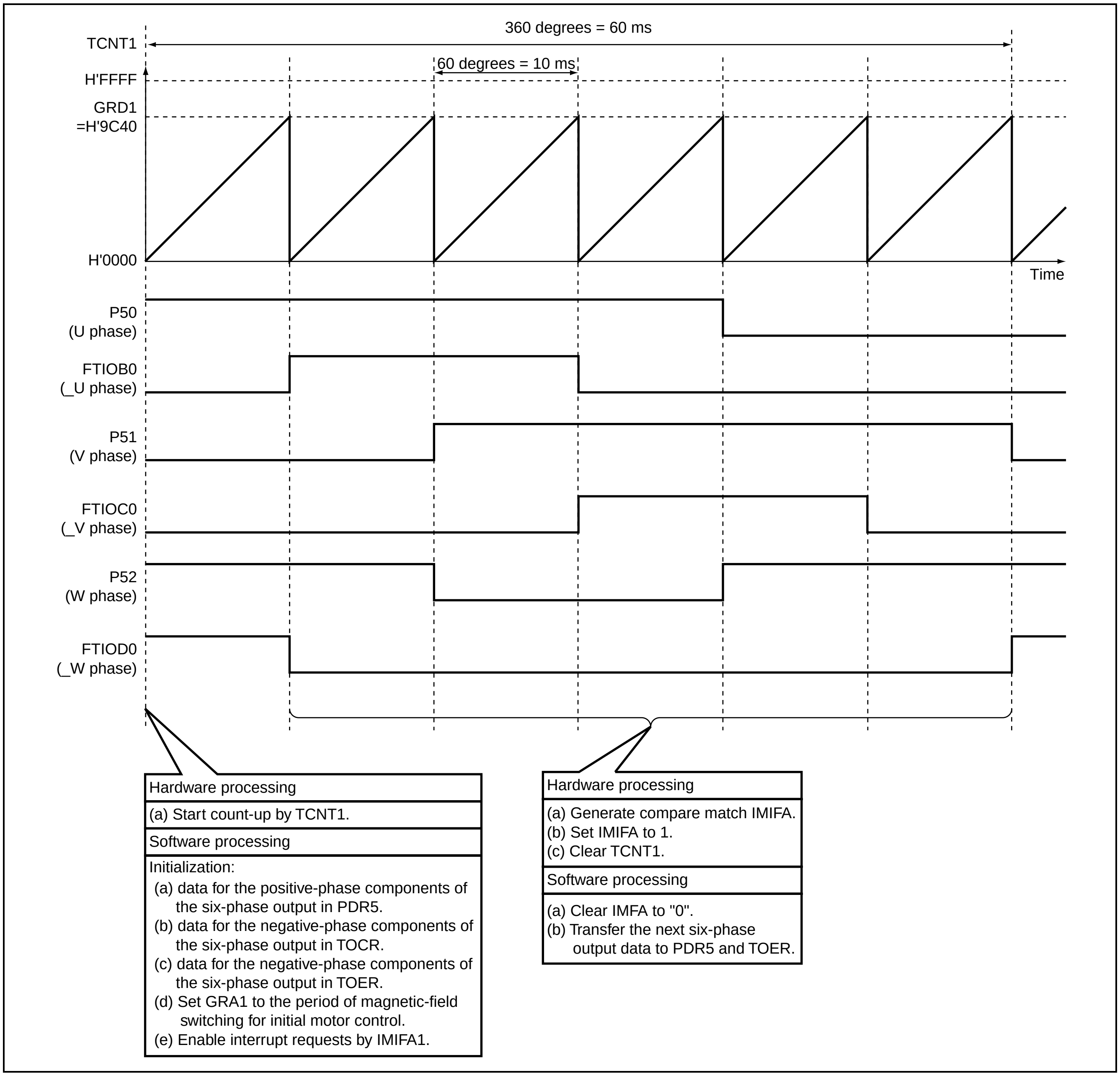
3. Table 3.1 lists the assignment of functions to H8/3687 elements for this sample task, i.e., control of the brushless DC motor.

Table 3.1 H8/3687 Function Assignments

H8/3687 Element	Assigned Function
FTIOA1 FTIOB1 FTIOC1	• Rotor-positional signal input
TMPR	• Select operating mode of compare–match waveform output.
GRD1	• Selection of the initial switching period for the motor-rotating field
TSR1	• Reflects the presence of timer Z (CH1) interrupt requests (IMIFA to IMIFD).
TIER1	• Enable/disable interrupt requests (IMIFA to IMIFD) from timer Z (CH1).
FTIOB0 FTIOC0 FTIOD0	• Six-phase output: negative-side signals
TCNT0	• 16-bit up-counter (timer Z: CH0).
TCNT1	• 16-bit up-counter (timer Z: CH1).
GRA0	• Set the chopping waveform's period
GRB0 GRC0 GRD0	• Setting for the chopping output waveform's duty cycle
TCR0	• Selection of the counter-clearing signal, driving clock edge, and clock source for counter TCNT0
TCR1	• Selection of the counter-clearing signal, driving clock edge, and clock source for counter TCNT1
TIOR	• Set the general-register (GR) functions
TOER	• Enable/disable CH0 and CH1 output of timer Z
TOCR	• Set initial output until the first compare–match
P50 to P52	• Six-phase output: positive-side signals
PCR5	• Set P50 to P52 input/output pin function
PDR5	• Storage of the data for output on the P50 to P52 input/output pins and reading of the pin levels
_IRQ0	• Abnormal-state-detected signal input
IEGR1	• Selection of the input edge for detection by _IRQ0
IER1	• Reflects the sources of IRQ0 interrupt requests

4. Description of Operation

1. Figure 4.1 shows the principle of operation in the initial stage of motor control (ie, switching of the rotating magnetic-field takes place at a constant period until the motor has gone through its first ten rotations). Initial control of the brushless DC motor is through hardware and software processing by the H8/3687 as shown in figure 4.1.



Hardware processing

(a) Start count-up by TCNT1.

Software processing

Initialization:
(a) data for the positive-phase components of the six-phase output in PDR5.
(b) data for the negative-phase components of the six-phase output in TOCR.
(c) data for the negative-phase components of the six-phase output in TOER.
(d) Set GRA1 to the period of magnetic-field switching for initial motor control.
(e) Enable interrupt requests by IMIFA1.

Hardware processing

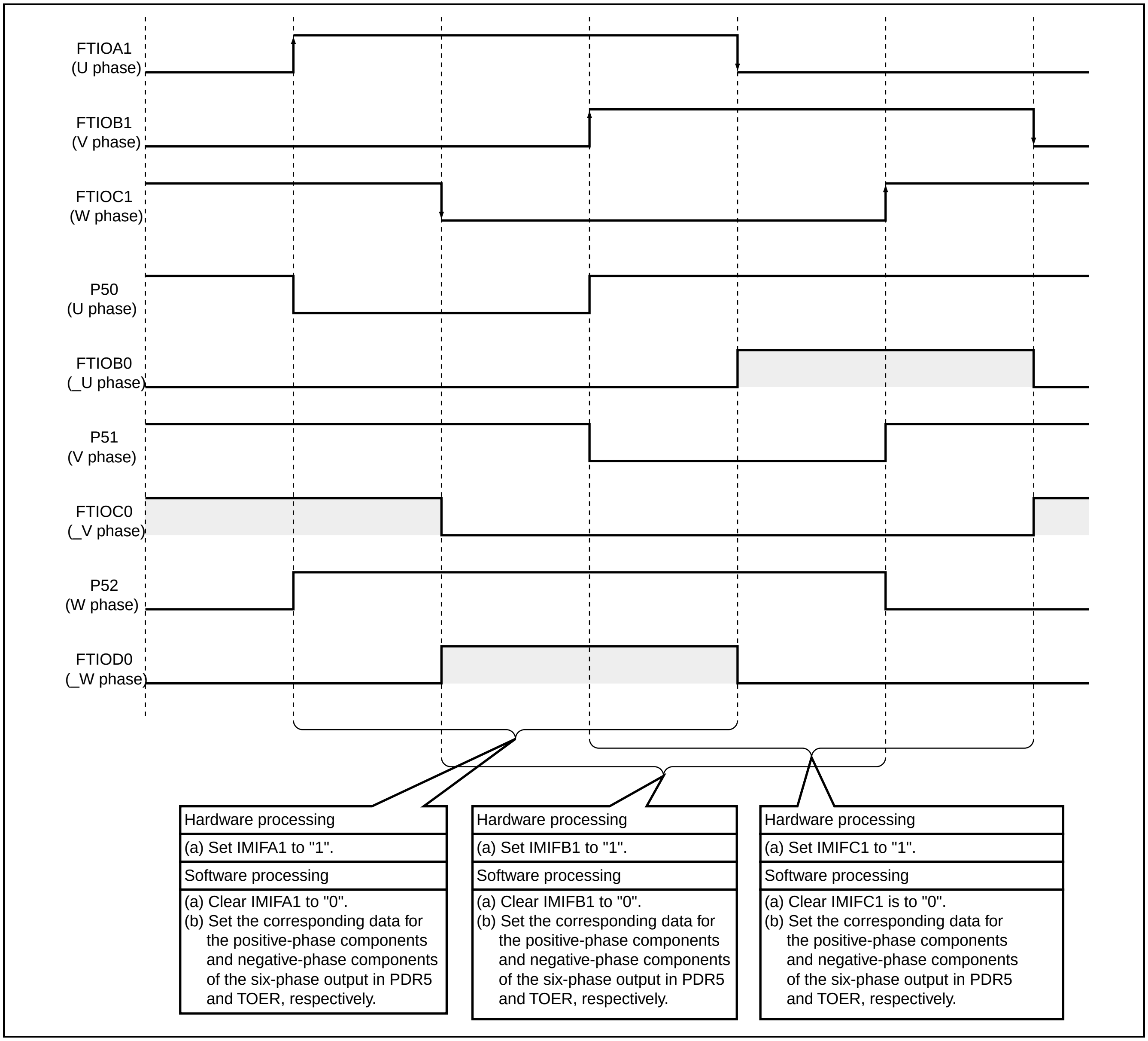
(a) Generate compare match IMIFA.
(b) Set IMIFA to 1.
(c) Clear TCNT1.

Software processing

(a) Clear IMFA to "0".
(b) Transfer the next six-phase output data to PDR5 and TOER.

Figure 4.1 Initial Control of the Brushless DC Motor: Principle of Operation

2. Figure 4.2 shows the principle of control to make the magnetic field rotate in response to the rotor-positional signal. Control of the brushless DC motor is through hardware and software processing by the H8/3687, based on the detected rotor-positional signal, as is shown in figure 4.2.



Hardware processing
(a) Set IMIFA1 to "1".
Software processing
(a) Clear IMIFA1 to "0".
(b) Set the corresponding data for the positive-phase components and negative-phase components of the six-phase output in PDR5 and TOER, respectively.

Hardware processing
(a) Set IMIFB1 to "1".
Software processing
(a) Clear IMIFB1 to "0".
(b) Set the corresponding data for the positive-phase components and negative-phase components of the six-phase output in PDR5 and TOER, respectively.

Hardware processing
(a) Set IMIFC1 to "1".
Software processing
(a) Clear IMIFC1 is to "0".
(b) Set the corresponding data for the positive-phase components and negative-phase components of the six-phase output in PDR5 and TOER, respectively.

Figure 4.2 Principle of Motor Control Based on the Rotor-Positional Signal

3. In this sample task, chopping control is applied when the driver transistors on the negative-phase side are turned on. Chopping waveforms are generated by the timer Z compare–match function and output on the FTIOB0 to FTIOD0 pins. Figure 4.3 shows the principle of operation for output of the chopping waveform.

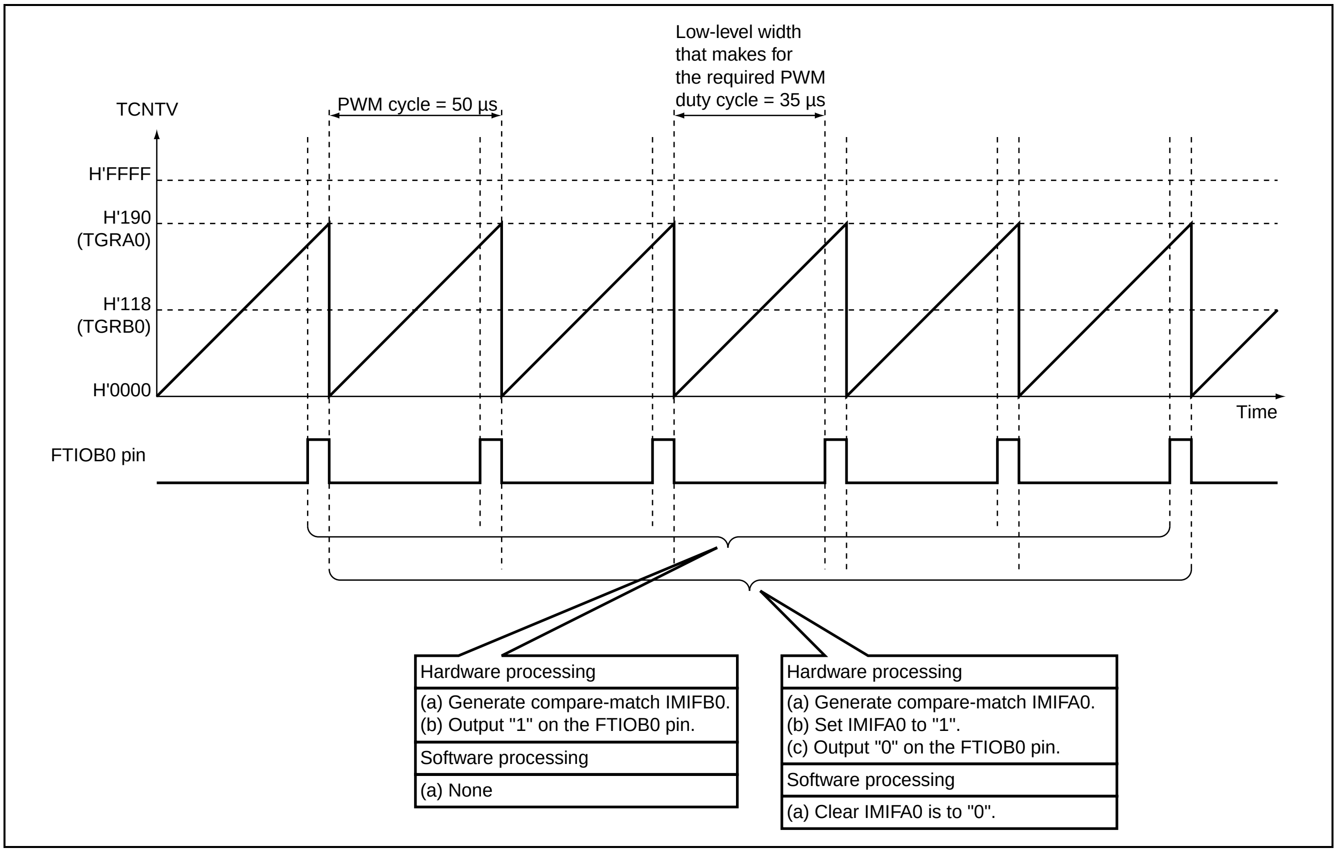


Figure 4.3 Output of the Chopping Waveform: Principle of Operation

- Method for calculating the phase and setting the PWM duty cycle:

  The value for establishing the next PWM duty cycle is calculated by the IMIFA interrupt routine and is set in GRB0. In calculating the value for setting in GRB0, the advancement of the phase by one carrier-period is calculated from the present frequency of the motor and carrier period.

  360 degrees / (Carrier period / Frequency of motor) = Advancement of the phase by one carrier period ---(1)

  Transform expression (1) to get

  (360 degrees × Motor-rotation period) / Carrier period = Advancement of the phase by one carrier period

  The value that corresponds to the calculated phase is obtained from a reference data table and set in GRB0.

Note: In this sample task, the PWM period (GRA0 setting) is fixed at 50 µs (H'320) and the low-level width that makes for the required PWM duty cycle (GRB0 setting) is fixed at 35µs (H'230).

5. Description of Software

5.1 Description of Modules

Table 5.1 describes the modules used in this sample task.

Table 5.1 Description of Modules

Module Name	Label Name	Function
Main routine	Reset	Initializes the stack pointer, sets up the IRQ0 interrupt, the port, places CH0 of timer Z in the PWM timer mode, sets up CH1 of timer Z for output comparison, and sets up the RAM to be used and the other interrupts.
Timer Z interrupt routine	Timer Z	This is the timer Z interrupt handler, which clears the interrupt-request flag and controls the six-phase output.
IRQ0 interrupt processing routine	IRQ0	The IRQ0 interrupt routine clears the interrupt request flag and sets the error flag to stop the motor.

5.2 Description of Arguments

Table 5.2 describes the arguments used in this sample task.

Table 5.2 Description of Arguments

Argument Name	Function	Used in	Size (Bytes)	I/O
GRA1	Initial motor control: holds the initial switching cycle for the motor's rotating field.	main routine	1	input
GRA0	Chopping output waveform: period setting	Timer Z	1	input
GRB0	Chopping output waveform: duty cycle	Timer Z	1	input
GRC0	Chopping output waveform: duty cycle	Timer Z	1	input
GRD0	Chopping output waveform: duty cycle	Timer Z	1	input

5.3 Description of Internal Registers

Table 5.3 describes the internal registers used in this sample task.

Table 5.3 Description of Internal Registers

Register Name	Function	Address	Setting
PDR5	Port Data Register 5: When P5n is "0", the level on pin P5n is "Low" (n=0 to 7). When P5n is "1", the level on pin P5n is "High" (n=0 to 7).	H'FFD8	H'04
PCR5	Port Control Register 5: When PCR5n is "0", pin P5n is an input pin.(n=0 to 7) When PCR5n is "1", pin P5n is an output pin.(n=0 to 7)	H'FFE8	H'07

Table 5.4 Description of Internal Registers (Ctd.)

Register Name		Function	Address	Setting

TCR0	CCLR2
	CCLR1
	CCLR0
		Timer Control Register 0
		(Counter clear 2 to 0):
		When CCLR2 = "0", CCLR1 = "0" and CCLR0 = "1", clearing of TCNT0 occurs on a GRA0 compare-match.
			H'F700
			Bit 7
			Bit 6
			Bit 5
				CCLR2=0
				CCLR1=0
				CCLR0=1

	CKEG1
	CKEG0
		Timer Control Register 0
		(Clock edge 1 to 0):
		When CKEG1 = "0" and CKEG0 = "0", counting is driven by rising edges.
			H'F700
			Bit 4
			Bit 3
				CKEG1=0
				CKEG0=0

	TPSC2
	TPSC1
	TPSC0
		Timer Control Register 0
		(Timer prescaler 2 to 0):
		When TPSC2 = "0", TPSC1 = "0" and TPSC0 = "0", counting is done by the internal clock cycle.
			H'F700
			Bit 2
			Bit 1
			Bit 0
				TPSC2=0
				TPSC1=0
				TPSC0=0

TIORA0	IOB2
	IOB1
	IOB0
		Timer I/O Control Register 0
		(I/O control B2 to B0):
		When IOB2 = "0", IOB 1 = "0" and IOB 0 = "0", pin output of the GRB0 compare-match state is disabled.
			H'F701
			Bit 6
			Bit 5
			Bit 4
				IOB2=0
				IOB1=0
				IOB0=0

TIORA0	IOA2
	IOA1
	IOA0
		Timer I/O Control Register 0
		(I/O control A2 to A0):
		When IOA2 = "0", IOA1 = "0" and IOA0 = "0", pin output by GRA0 compare-match is disabled.
			H'F701
			Bit 2
			Bit 1
			Bit 0
				IOA2=0
				IOA1=0
				IOA0=0

TIORC0	IOD2
	IOD1
	IOD0
		Timer I/O Control Register 0
		(I/O control D2 to D0):
		When IOD2 = "0", IOD1 = "0" and IOD0 = "0", pin output by GRD0 compare-match is disabled.
			H'F702
			Bit 6
			Bit 5
			Bit 4
				IOD2=0
				IOD1=0
				IOD0=0

TIORC0	IOC2
	IOC1
	IOC0
		Timer I/O Control Register 0
		(I/O control C2 to C0):
		When IOC2 = "0", IOC1 = "0" and IOC0 = "0", pin output by GRC0 compare-match is disabled.
			H'F702
			Bit 2
			Bit 1
			Bit 0
				IOC2=0
				IOC1=0
				IOC0=0

TSR0	IMFA0	Timer Status Register 0: (Input capture/output compare-match flag A) When IMFA0 = "0", an input capture/output compare-match has not occurred. When IMFA0 = "1", an input capture/output compare-match results has occurred.	H'F703 Bit 0	0

Table 5.4 Description of Internal Registers (Ctd.) (cont)

Register Name		Function	Address	Setting
TIER0	IMIEA	Timer Interrupt Enable Register 0	H'F704	0
		(Input capture / output compare-match interrupt A enable):	Bit 0	
		When IMIEA = "0", ICFA interrupt requests are disabled.		
		When IMIEA = "1", ICFA interrupt requests are enabled.		
POCR0	POLD	PWM Mode Output Level Control Register 0	H'F705	POLD=1
	POLC	(PWM output control register B to D):	Bit 2	POLC=1
	POLB	When POLD = "0", POLC = "0" and POLB = "0", the FTIOD0, FTIOC0 and FTIOB0 outputs are active low.	Bit 1	POLB=1
			Bit 0	
GRA0		General Register A0	H'F708	H'320
		(Input-capture/output compare-match register):		
		If GRA0 = "H'320", an IMFA0 compare-match is generated and FTIOB0 to FTIOD0 go low when counting by TCNT1 reaches "H'320".		
GRB0		General Register B0	H'F709	H'230
		(Input capture / output compare-match register)		
		: When GRB0 = "H'230", if TCNT1 counts up to "H'230", IMFB0 compare-match is generated and FTIOB is high level.		
GRC0		General Register C0	H'F70A	H'230
		(Input capture / output compare-match register)		
		: When GRC0 = "H'230", if TCNT1 counts up to "H'230", IMFC0 compare-match is generated and FTIOC is high level.		
GRD0		General Register D0	H'F70B	H'230
		(Input capture / output compare-match register)		
		: When GRD0 = "H'230", if TCNT1 counts up to "H'230", IMFD0 compare-match is generated and FTIOD is high level.		
TCR1	CCLR2	Timer Control Register 1	H'F710	CCLR2=1
	CCLR1	(Counter clear 2 to 0):	Bit 7	CCLA1=1
	CCLR0	When CCLR2 = "1", CCLR1 = "1" and CCLR0 = "0", a GRD0 compare-match clears TCNT0.	Bit 6	CCLA0=0
			Bit 5	
	CKEG1	Timer Control Register 1	H'F710	CKEG1=0
	CKEG0	(Clock edge 1 to 0):	Bit 4	CKEG0=0
		When CKEG1 = "0" and CKEG0 = "0", counting is driven by rising edges.	Bit 3	
	TPSC2	Timer Control Register 1	H'F710	TPSC2=0
	TPSC1	(Timer prescaler 2 to 0):	Bit 2	TPSC1=0
	TPSC0	When TPSC2 = "0", TPSC1 = "0" and TPSC0 = "1", counting is done by the divided-by-two internal clock (φ/2) cycles.	Bit 1	TPSC0=1
			Bit 0	

Table 5.4 Description of Internal Registers (Ctd.) (cont)

Register Name		Function	Address	Setting
TIORA1	IOB2	Timer I/O Control Register A1	H'F711	IOB2=1
	IOB1	(I/O control B2 to B0)	Bit 6	IOB1=0
	IOB0	When IOB2 = "1", IOB1 = "0" and IOB0 = "0" input capture to GRB1 is done on the rising edge.	Bit 5	IOB0=0
			Bit 4	
TIORA1	IOA2	Timer I/O Control Register A1	H'F711	IOA2=1
	IOA1	(I/O control A2 to A0)	Bit 2	IOA1=0
	IOA0	When IOA2 = "1", IOA1 = "0" and IOA0 = "0", input capture to GRA1 is done on the rising edge.	Bit 1	IOA0=0
			Bit 0	
TIORC1	IOD2	Timer I/O Control Register C1	H'F712	IOD2=0
	IOD1	(I/O control D2 to D0)	Bit 6	IOD1=0
	IOD0	: When IOD2 = "0", IOD1 = "0" and IOD0 = "0", pin output by GRD0 compare-match is disabled.	Bit 5	IOD0=0
			Bit 4	
TIORC1	IOC2	Timer I/O Control Register C1	H'F712	IOC2=1
	IOC1	(I/O control C2 to C0)	Bit 2	IOC1=0
	IOC0	When IOC2 = "1", IOC1 = "0" and IOC0 = "0", input capture to GRC1 is done on the rising edge.	Bit 1	IOC0=0
			Bit 0	
TSR1	IMFA1	Timer Status Register 1	H'F713	0
		(Input capture/output compare-match flag A):	Bit 0	
		When IMFA1 = "0", an input capture/output compare-match has not occurred.		
		When IMFA1 = "1", an input capture/output compare-match has occurred.		
	IMFB1	Timer Status Register 1	H'F713	0
		(Input capture / output compare-match flag B):	Bit 1	
		When IMFB1 = "0", input capture / output compare-match has not been generated.		
		When IMFB1 = "1", input capture/output compare-match has been generated.		
	IMFC1	Timer Status Register 1	H'F713	0
		(Input capture / output compare-match flag C)	Bit 2	
		: When IMFC1 = "0", input capture/output compare-match has not been generated.		
		: When IMFC1 = "1", input capture/output compare-match has been generated.		

Table 5.4 Description of Internal Registers (Ctd.) (cont)

Register Name		Function	Address	Setting
TSR1 (cont)	IMFD1	Timer Status Register 1 (Input capture / output compare-match flag D) : When IMFD1 = "0", input capture/output compare-match has not been generated. : When IMFD1 = "1", input capture/output compare-match has been generated.	H'F713 Bit 3	0
GRD1		General Register (Input capture / output compare-match register): If GRD1 = "H'9C40", an IMIFD1 compare-match is generated when counting by TCNT1 reaches "H'9C40".	H'F71E	H'9C40
TSTR	STR1	Timer Start Register (Counter start 0, 1): When STR1 = "0", counting operation by TCNT1 is disabled. When STR1 = "1", counting by TCNT1 is enabled.	H'F720 Bit 1	1
	STR0	Timer Start Register (Counter start 0, 1) : When STR0 = "0", counting by TCNT0 is disabled. : When STR0 = "1", counting by TCNT0 is enabled.	H'F720 Bit 0	1
TPMR	PWMD0	Timer PWM Mode Register (PWM mode): When PWMD0 = "0", FTIOD0 operation is in the normal mode. When PWMD0 = "1", FTIOD0 operation is in the PWM mode.	H'F722 Bit 2	1
	PWMC0	Timer PWM Mode Register (PWM mode) : When PWMC0 = "0", FTIOC0 is normal mode. : When PWMC0 = "1", FTIOC0 is PWM mode.	H'F722 Bit 1	1
	PWMB0	Timer PWM Mode Register (PWM mode) : When PWMB0 = "0", FTIOB0 is normal mode. : When PWMB0 = "1", FTIOB0 is PWM mode.	H'F722 Bit 0	1

Table 5.4 Description of Internal Registers (Ctd.) (cont)

Register Name		Function	Address	Setting
TOER	ED0	Timer Output Master Enable Register (Master enable FTIOD0): When ED0 = "0", enabling or disabling of the FTIOD0 pin output is according to the settings of TPMRE, TFCR and TIORC1. When ED0 = "1", FTIOD0 pin timer output is disabled, regardless of the setting of TPMR, TFCR and TIORC1 (the FTIOD0 pin functions as an I/O port pin.) In setting of WKP4N input, when inputting low level to WKP4N, "1" is set.	H'F724 Bit 3	0
	EC0	Timer Output Master Enable Register (Master enable FTIOC0) : When EC0 = "0", FTIOC0 pin output is enabled according to the setting of TPMRE, TFCR and TIORC1. : When EC0 = "1", FTIOC0 pin timer output is disabled regardless of the setting of TPMR, TFCR and TIORC1. (FTIOC0 pin functions as I/O port.) When WKP4N is input, if inputting low level to WKP4N, "1" is set.	H'F724 Bit 2	0
	EB0	Timer Output Master Enable Register (Master enable FTIOB0) : When EB0 = "0", FTIOB0 pin output is enabled according to the setting of TPMRE, TFCR and TIORC1. : When EB0 = "1", FTIOB0 pin timer output is enabled regardless of the setting of TPMR, TFCR and TIORC1. (FTIOB0 pin functions as I/O port.) When WKP4N input is set, if inputting low level to WKP4N, "1" is set.	H'F724 Bit 1	0
TOCR	TOD0	Timer Output Control Register (Output level select FTIOD0): When TOD0 = "0", "0" is output on FTIOD0. When TOD0 = "1", "1" is output on FTIOD0.	H'F725 Bit 3	1
	TOC0	Timer Output Control Register (Output level select FTIOC0) : When TOC0 = "0", "0" is output from FTIOC0. : When TOC0 = "1", "1" is output from FTIOC0.	H'F725 Bit 2	1
	TOB0	Timer Output Control Register (Output level select FTIOB0) : When TOB0 = "0", "0" is output from FTIOB0. : When TOB0 = "1", "1" is output from FTIOB0.	H'F725 Bit 1	1

Table 5.4 Description of Internal Registers (Ctd.) (cont)

Register Name		Function	Address	Setting
IEGR1	IEG0	Interrupt Edge Select Register 1 (IRQ0 edge select): When IEG0 = "0", the falling edge of the signal on the IRQ0 pin is detected. When IEG0 = "1", the rising edge of the signal on the IRQ0 pin is detected.	H'FFF2 Bit 0	0
IENR1	IEN0	Interrupt Enable Register 1 (IRQ0 interrupt enable): When IEN0 = "0", IRQ0 interrupt requests are disabled. When IEN0 = "1", IRQ0 interrupt requests are enabled.	H'FFF4 Bit 0	0
IRR1	IRRI0	Interrupt Request Register 1 (IRQ0 interrupt request flag): When IRRI0 = "0", IRQ0 interrupts are not requested. When IRRI0 = "1", IRQ0 interrupts are not is requested.	H'FFF6 Bit 0	0

5.4 Description of RAM

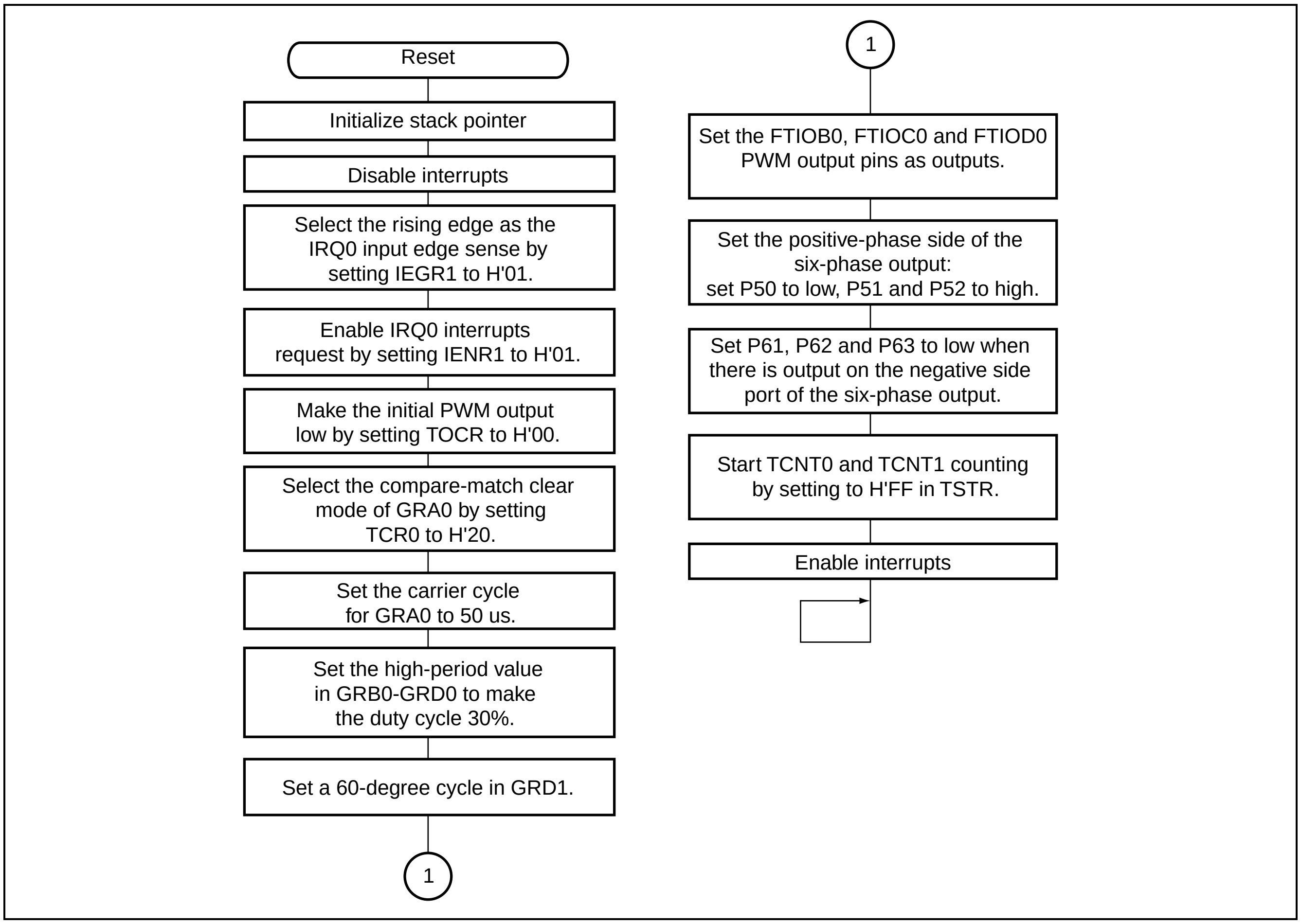
Table 5.5 describes the RAM used in this sample task.

Table 5.5 Description of RAM Usage

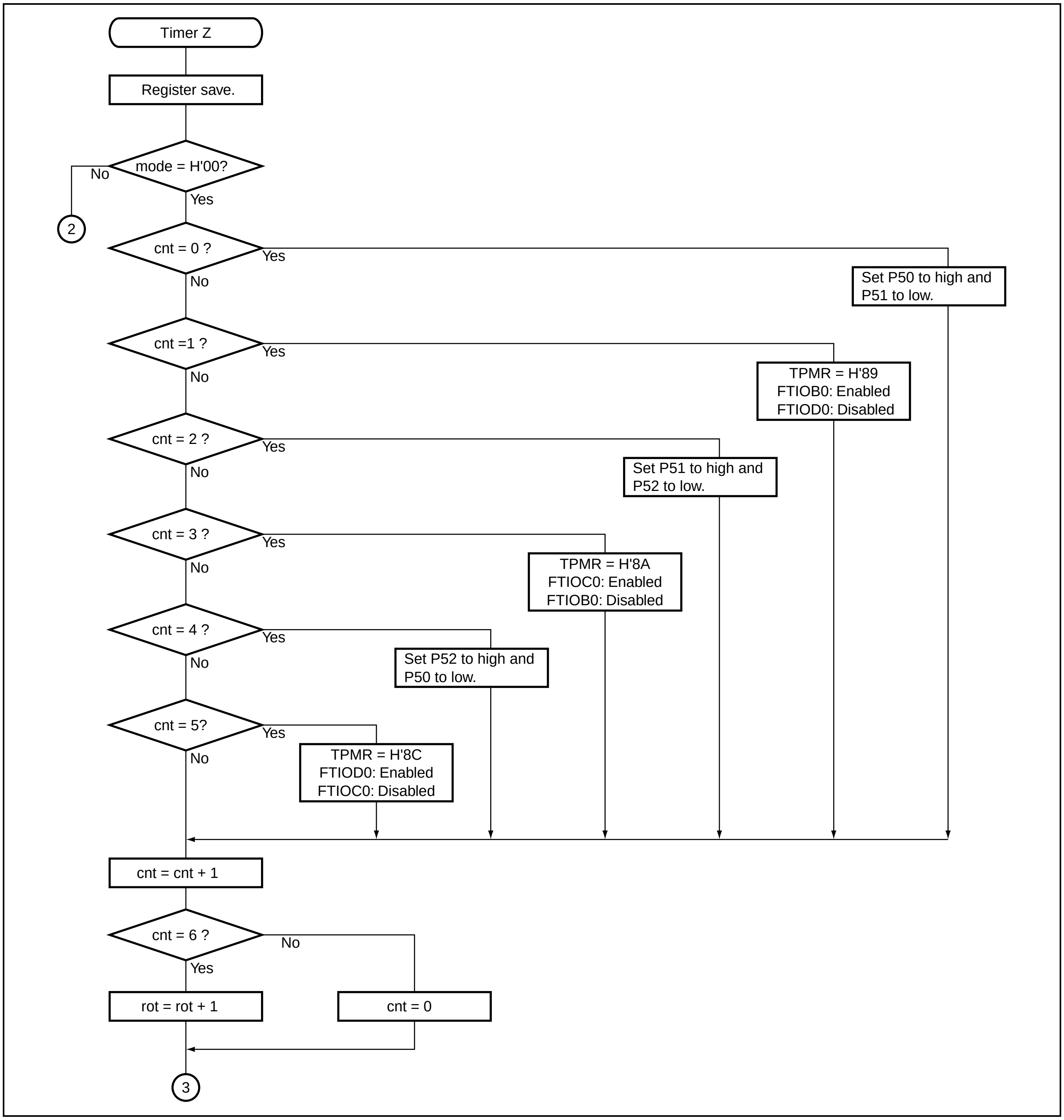
Label name	Function	Used in
cnt	Counter for switching of the output waveform every 60 degrees.	Timer Z
rot	Counter for the motor's initial operation cycle.	Timer Z
mode	Flag to indicate switching of the motor from initial to normal operation.	Timer Z

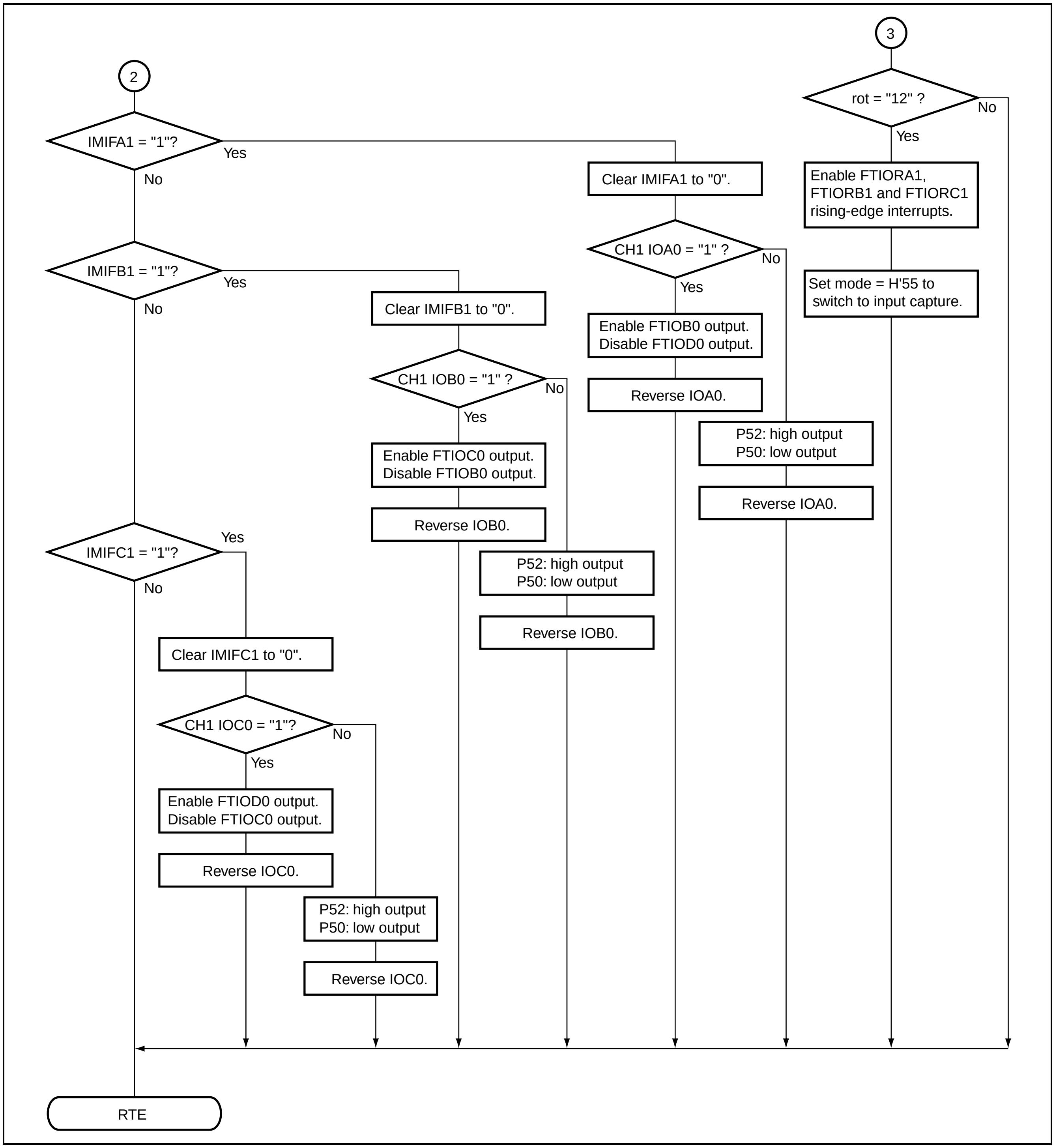
6. Flowchart

1. Main routine

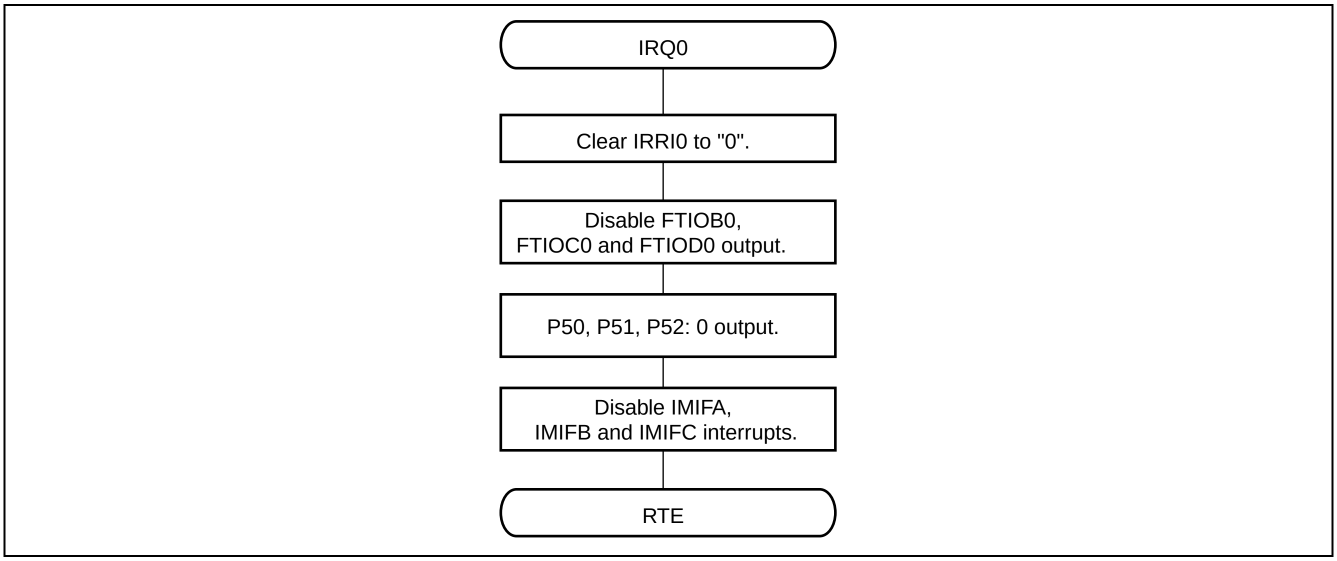

Reset
Initialize stack pointer
Disable interrupts
Select the rising edge as the IRQ0 input edge sense by setting IEGR1 to H'01.
Enable IRQ0 interrupts request by setting IENR1 to H'01.
Make the initial PWM output low by setting TOCR to H'00.
Select the compare-match clear mode of GRA0 by setting TCR0 to H'20.
Set the carrier cycle for GRA0 to 50 us.
Set the high-period value in GRB0-GRD0 to make the duty cycle 30%.
Set a 60-degree cycle in GRD1.
1
1
Set the FTIOB0, FTIOC0 and FTIOD0 PWM output pins as outputs.
Set the positive-phase side of the six-phase output: set P50 to low, P51 and P52 to high.
Set P61, P62 and P63 to low when there is output on the negative side port of the six-phase output.
Start TCNT0 and TCNT1 counting by setting to H'FF in TSTR.
Enable interrupts

2. Timer Z interrupt processing routine




3
rot = "12" ?
No
Yes
Enable FTIORA1, FTIORB1 and FTIORC1 rising-edge interrupts.
Set mode = H'55 to switch to input capture.
2
IMIFA1 = "1"?
Yes
No
Clear IMIFA1 to "0".
CH1 IOA0 = "1" ?
No
Yes
Enable FTIOB0 output. Disable FTIOD0 output.
Reverse IOA0.
P52: high output P50: low output
Reverse IOA0.
IMIFB1 = "1"?
Yes
No
Clear IMIFB1 to "0".
CH1 IOB0 = "1" ?
No
Yes
Enable FTIOC0 output. Disable FTIOB0 output.
Reverse IOB0.
P52: high output P50: low output
Reverse IOB0.
IMIFC1 = "1"?
Yes
No
Clear IMIFC1 to "0".
CH1 IOC0 = "1"?
No
Yes
Enable FTIOD0 output. Disable FTIOC0 output.
Reverse IOC0.
P52: high output P50: low output
Reverse IOC0.
RTE

3. IRQ0 interrupt processing routine



7. Program Listing

INIT. SRC (program listing)

```
        .EXPORT    _INIT
        .IMPORT    _main
    ;
        .SECTION  P,CODE
    _INIT:
        MOV.W      #H'FF80,R7
        LDC.B      #B'10000000,CCR
        JMP        @_main
    ;
        .END
```

```
/***********************************************************/
/*                                                         */
/*  H8/300HN Series -H8/3687-                              */
/*  Application Note                                       */
/*                                                         */
/*'DC brushless motor control function'                    */
/*                                                         */
/*  Function                                               */
/*  :Timer Z PWM mode &input capture                       */
/*                                                         */
/*  External Clock :16MHz                                  */
/*  Internal Clock :16MHz                                  */
/*  Sub Clock :32.768kHz                                   */
/*                                                         */
/***********************************************************/
#include <machine.h>
/***********************************************************/
/*Symbol Definition                                        */
/***********************************************************/
struct BIT {
unsigned char b7:1;     /*bit7 */
unsigned char b6:1;     /*bit6 */
unsigned char b5:1;     /*bit5 */
unsigned char b4:1;     /*bit4 */
unsigned char b3:1;     /*bit3 */
unsigned char b2:1;     /*bit2 */
unsigned char b1:1;      /*bit1 */
unsigned char b0:1;      /*bit0 */
#define TCR0 *(volatile unsigned char *)0xF700             /*Timer Control Register 0                        */
#define TCR0_BIT (*(struct BIT *)0xF700)                   /*Timer Control Register 0                        */
#define CCLR2_0 TCR0_BIT.b7                                /*Counter Clear 2                                 */
#define CCLR1_0 TCR0_BIT.b6                                /*Counter Clear 1                                 */
#define CCLR0_0 TCR0_BIT.b 5                               /*Counter Clear 0                                 */
#define CKEG1_0 TCR0_BIT.b4                                /*Clock Select 1                                  */
#define CKEG0_0 TCR0_BIT.b3                                /*Clock Select 0                                  */
#define TPSC2_0 TCR0_BIT.b2                                /*Timer Prescaler 2                               */
#define TPSC1_0 TCR0_BIT.b1                                /*Timer Prescaler 1                               */
#define TPSC0_0 TCR0_BIT.b0                                /*Timer Prescaler 0                               */
#define TIORA0 *(volatile unsigned char *)0xF701           /*Timer I/O Control Register A0                   */
#define TIORA0_BIT (*(struct BIT *)0xF701)                 /*Timer I/O Control Register A0                   */
#define IOB2_0 TIORA0_BIT.b6                               /*I/O Control B2                                  */
#define IOB1_0 TIORA0_BIT.b5                               /*I/O Control B1                                  */
#define IOB0_0 TIORA0_BIT.b4                               /*I/O Control B0                                  */
#define IOA2_0 TIORA0_BIT.b2                               /*I/O Control A2                                  */
#define IOA1_0 TIORA0_BIT.b1                               /*I/O Control A1                                  */
#define IOA0_0 TIORA0_BIT.b0                               /*I/O Control A0                                  */
#define TIORC0 *(volatile unsigned char *)0xF702           /*Timer I/O Control Register C0                   */
#define TIORC0_BIT (*(struct BIT *)0xF702)                 /*Timer I/O Control Register C0                   */
#define IOD2_0 TIORC0_BIT.b6                               /*I/O Control B2                                  */
#define IOD1_0 TIORC0_BIT.b5                               /*I/O Control B1                                  */
#define IOD0_0 TIORC0_BIT.b4                               /*I/O Control B0                                  */
#define IOC2_0 TIORC0_BIT.b2                               /*I/O Control A2                                  */
#define IOC1_0 TIORC0_BIT.b1                               /*I/O Control A1                                  */
#define IOC0_0 TIORC0_BIT.b0                               /*I/O Control A0                                  */
#define TSR0 *(volatile unsigned char *)0xF703             /*Timer Status Register 0                         */
#define TSR0_BIT (*(struct BIT *)0xF703)                   /*Timer Status Register 0                         */
#define UDF_0 TSR0_BIT.b5                                  /*Under Flow Flag                                 */
#define OVF_0 TSR0_BIT.b4                                  /*Over Flow Flag                                  */
#define IMIFD_0 TSR0_BIT.b3                                /*Input Caputure/Compare-match Flag D             */
#define IMIFC_0 TSR0_BIT.b2                                /*Input Caputure/Compare-match Flag C             */
#define IMIFB_0 TSR0_BIT.b1                                /*Input Caputure/Compare-match Flag B             */
#define IMIFA_0 TSR0_BIT.b0                                /*Input Caputure/Compare-match Flag A             */
#define TIER0 *(volatile unsigned char *)0xF704            /*Timer Interrupt Enable Register 0               */
#define TIER0_BIT (*(struct BIT *)0xF704)                  /*Timer Interrupt Enable Register 0               */
#define OVIE_0 TIER0_BIT.b4                                /*Over Flow Interrupt Enable                      */
#define IMIED_0 TIER0_BIT.b3                               /*Input Capture/compare-match Interrupt Enable D  */
```

```
#define IMIEC_0 TIER0_BIT.b2                               /*Input Capture/compare-match Interrupt Enable C  */
#define IMIEB_0 TIER0_BIT.b1                               /*Input Capture/compare-match Interrupt Enable B  */
#define IMIEA_0 TIER0_BIT.b0                               /*Input Capture/compare-match Interrupt Enable A  */
#define POCR0 *(volatile unsigned char *)0xF705            /*Port Output Level Control Register              */
#define POCR0_BIT (*(struct BIT *)0xF705)                  /*Port Output Level Control Register              */
#define POLD_0 POCR0_BIT.b2                                /*PWM Mode Outputr Level Control Register D       */
#define POLC_0 POCR0_BIT.b1                                /*PWM Mode Outputr Level Control Register C       */
#define POLB_0 POCR0_BIT.b0                                /*PWM Mode Outputr Level Control Register B       */
#define TCNT0 *(volatile unsigned short *)0xF706           /*Timer Counter 0                                 */
#define GRA0 *(volatile unsigned short *)0xF708            /*General Register A0                             */
#define GRB0 *(volatile unsigned short *)0xF70A            /*General Register B0                             */
#define GRC0 *(volatile unsigned short *)0xF70C            /*General Register C0                             */
#define GRD0 *(volatile unsigned short *)0xF70E            /*General Register D0                             */
#define TCR1 *(volatile unsigned char *)0xF710             /*Timer Control Register 1                        */
#define TCR1_BIT (*(struct BIT *)0xF710)                   /*Timer Control Register 1                        */
#define CCLR2_1 TCR1_BIT.b7                                /*Counter Clear 2                                 */
#define CCLR1_1 TCR1_BIT.b6                                /*Counter Clear 1                                 */
#define CCLR0_1 TCR1_BIT.b5                                /*Counter Clear 0                                 */
#define CKEG1_1 TCR1_BIT.b4                                /*Clock Select 1                                  */
#define CKEG0_1 TCR1_BIT.b3                                /*Clock Select 0                                  */
#define TPSC2_1 TCR1_BIT.b2                                /*Timer Prescaler 2                               */
#define TPSC1_1 TCR1_BIT.b1                                /*Timer Prescaler 1                               */
#define TPSC0_1 TCR1_BIT.b0                                /*Timer Prescaler 0                               */
#define TIORA1 *(volatile unsigned char *)0xF711           /*Timer I/O Control Register A1                   */
#define TIORA1_BIT (*(struct BIT *)0xF711)                 /*Timer I/O Co ntrol Register A1                  */
#define IOB2_1 TIORA1_BIT.b6                               /*I/O Control B2                                  */
#define IOB1_1 TIORA1_BIT.b5                               /*I/O Control B1                                  */
#define IOB0_1 TIORA1_BIT.b4                               /*I/O Control B0                                  */
#define IOA2_1 TIORA1_BIT.b2                               /*I/O Control A2                                  */
#define IOA1_1 TIORA1_BIT.b1                               /*I/O Control A1                                  */
#define IOA0_1 TIORA1_BIT.b0                               /*I/O Control A0                                  */
#define TIORC1 *(volatile unsigned char *)0xF712           /*Timer I/O Control Register C1                   */
#define TIORC1_BIT (*(struct BIT *)0xF712)                 /*Timer I/O Control Register C1                   */
#define IOD2_1 TIORC1_BIT.b6                               /*I/O Control D2                                  */
#define IOD1_1 TIORC1_BIT.b5                               /*I/O Control D1                                  */
#define IOD0_1 TIORC1_BIT.b4                               /*I/O Control D0                                  */
#define IOC2_1 TIORC1_BIT.b2                               /*I/O Control C2                                  */
#define IOC1_1 TIORC1_BIT.b1                               /*I/O Control C1                                  */
#define IOC0_1 TIORC1_BIT.b0                               /*I/O Control C0                                  */
#define TSR1 *(volatile unsigned char *)0xF713             /*Timer Status Register 1                         */
```

```
  #define TSR1_BIT (*(struct BIT *)0xF713)                   /*Timer Status Register 1                         */
  #define UDF_1 TSR1_BIT.b5                                  /*Under Flow Flag                                 */
  #define OVF_1 TSR1_BIT.b4                                  /*Over Flow Flag                                  */
  #define IMIFD_1 TSR1_BIT.                                  /*Input Caputure/Compare-match Flag D             */
  #define IMIFC_1 TSR1_BIT.b2                                /*Input Caputure/Compare-match Flag C             */
  #define IMIFB_1 TSR1_BIT.b1                                /*Input Caputure/Compare-match Flag B             */
  #define IMIFA_1 TSR1_BIT.b0                                /*Input Caputure/Compare-match Flag A             */
  #define TIER1 *(volatile unsigned char *)0xF714            /*Timer Interrupt Enable Register 0               */
  #define TIER1_BIT (*(struct BIT *)0xF714)                  /*Timer Interrupt Enable Register 0               */
  #define OVIE_1 TIER1_BIT.b4                                /*Over Flow Interrupt Enable                      */
  #define IMIED_1 TIER1_BIT.b3                               /*Input Capture/compare-match Interrupt Enable D  */
  #define IMIEC_1 TIER1_BIT.b2                               /*Input Capture/compare-match Interrupt Enable C  */
  #define IMIEB_1 TIER1_BIT.b1                               /*Input Capture/compare-match Interrupt Enable B  */
  #define IMIEA_1 TIER1_BIT.b0                               /*Input Capture/compare-match Interrupt Enable A  */
  #define POCR1 *(volatile unsigned char *)0xF715            /*Port Output Level Control Register              */
  #define POCR1_BIT (*(struct BIT *)0xF715)                  /*Port Output Level Control Register              */
  #define POLD_1 POCR1_BIT.b2                                /*PWM Mode Outputr Level Control Register D       */
  #define POLC_1 POCR1_BIT.b1                                /*PWM Mode Outputr Level Control Register C       */
  #define POLB_1 POCR1_BIT.b0                                /*PWM Mode Outputr Level Control Register B       */
  #define TCNT1 *(volatile unsigned short *)0xF716           /*Timer Counter 1                                 */
  #define GRA1 *(volatile unsigned short *)0xF718            /*General Register A1                             */
  #define GRB1 *(volatile unsigned short *)0xF71A            /*General Register B1                             */
  #define GRC1 *(volatile unsigned short *)0xF71C            /*General Register C1                             */
  #define GRD1 *(volatile unsigned short *)0xF71E            /*General Register D1                             */
  #define TSTR *(volatile unsigned char *)0xF720             /*Timer Start Register                            */
  #define TSTR_BIT (*(struct BIT *)0xF720)                   /*Timer Start Register                            */
  #define STR1 TSTR_BIT.b1                                   /*Counter Start 1                                 */
  #define STR0 TSTR_BIT.b0                                   /*Counter Start 0                                 */
  #define TMDR *(volatile unsigned char *)0xF721             /*Timer Mode Register                             */
  #define TMDR_BIT (*(struct BIT *)0xF721)                   /*Timer Mode Register                             */
  #define BFD1 TMDR_BIT.b7                                   /*Buffer Register D1                              */
  #define BFC1 TMDR_BIT.b6                                   /*Buffer Register D0                              */
  #define BFD0 TMDR_BIT.b5                                   /*Buffer Register C1                              */
  #define BFC0 TMDR_BIT.b4                                   /*Buffer Register C0                              */
  #define SYNC TMDR_BIT.b0                                   /*timer Synchonus Mode                            */
  #define TPMR *(volatile unsigned char *)0xF722             /*Timer PWM Mode Register                         */
  #define TPMR_BIT (*(struct BIT *)0xF722)                   /*Timer PWM Mode Register                         */
  #define PWMD1 TPMR_BIT.b6                                  /*PWM Mode D1                                     */
  #define PWMC1 TPMR_BIT.b5                                  /*PWM Mode C1                                     */
  #define PWMB1 TPMR_BIT.b4                                  /*PWM Mode B1                                     */
```

```
#define PWMD0 TPMR_BIT.b2                                  /*PWM Mode D0                                     */
#define PWMC0 TPMR_BIT.b1                                  /*PWM Mode C0                                     */
#define PWMB0 TPMR_BIT.b0                                  /*PWM Mode B0                                     */
#define TFCR *(volatile unsigned char *)0xF723             /*Timer Function Control Register                 */
#define TFCR_BIT (*(struct BIT *)0xF723)                   /*Timer Function Control Register                 */
#define STCLK TOER_BIT.b0                                  /*External Clock Select                           */
#define ADEG TOER_BIT.b2                                   /*A/D Egde Trigger Select                         */
#define ADTRG TOER_BIT.b1                                  /*A/D Trigger Disable                             */
#define OLS1 TOER_BIT.b0                                   /*Output Level Select 1                           */
#define OLS0 TOER_BIT.b2                                   /*Output Level Select 0                           */
#define CMD1 TOER_BIT.b1                                   /*Conbination Mode 1                              */
#define CMD0 TOER_BIT.b0                                   /*Conbination Mode 0                              */
#define TOER *(volatile unsigned char *)0xF724             /*Timer Output Master Enable Register             */
#define TOER_BIT (*(struct BIT *)0xF724)                   /*Timer Output Master Enable Register             */
#define ED1 TOER_BIT.b7                                    /*Master Enable D1                                */
#define EC1 TOER_BIT.b6                                    /*Master Enable C1                                */
#define EB1 TOER_BIT.b5                                    /*Master Enable B1                                */
#define EA1 TOER_BIT.b4                                    /*Master Enable A1                                */
#define ED0 TOER_BIT.b3                                    /*Master Enable D0                                */
#define EC0 TOER_BIT.b2                                    /*Master Enable C0                                */
#define EB0 TOER_BIT.b1                                    /*Master Enable B0                                */
#define EA0 TOER_BIT.b0                                    /*Master Enable A0                                */
#define TOCR *(volatile unsigned char *)0xF725             /*Timer Output Master Enable Register             */
#define TOCR_BIT (*(struct BIT *)0xF725)                   /*Timer Output Master Enable Register             */
#define TOD1 TOCR_BIT.b7                                   /*Output Level Select D1                          */
#define TOC1 TOCR_BIT.b6                                   /*Output Level Select C1                          */
#define TOB1 TOCR_BIT.b5                                   /*Output Level Select B1                          */
#define TOA1 TOCR_BIT.b4                                   /*Output Level Select A1                          */
#define TOD0 TOCR_BIT.b3                                   /*Output Level Select D0                          */
#define TOC0 TOCR_BIT.b2                                   /*Output Level Select C0                          */
#define TOB0 TOCR_BIT.b1                                   /*Output Level Select B0                          */
#define TOA0 TOCR_BIT.b0                                   /*Output Level Select A0                          */
#define IEGR1 *(volatile unsigned char *)0xFFF2            /*Interrupt Edge Select Register 1                */
#define IEGR1_BIT (*(struct BIT *)0xFFF2)                  /*Interrupt Edge Select Register 1                */
#define IEG0 IEGR1_BIT.b0                                  /*IRQ0 Edge Select                                */
#define IENR1 *(volatile unsigned char *)0xFFF4            /*Interrupt Enable Register 1                     */
#define IENR1_BIT (*(struct BIT *)0xFFF4)                  /*Interrupt Enable Register 1                     */
#define IEN0 IENR1_BIT.b0                                  /*IRQ0 Interrupt Enable                           */
#define IRR1 *(volatile unsigned char *)0xFFF6             /*Interrupt Flag Register 1                       */
#define IRR1_BIT (*(struct BIT *)0xFFF6)                   /*Interrupt Flag Register 1                       */
```

```
#define IRRI0 IRR1_BIT.b0                                  /*IRQ0 Interrupt Request Flag                     */
#define PMR1 *(volatile unsigned char *)0xFFE0             /*Port Mode Register                              */
#define PMR1_BIT (*(struct BIT *)0xFFE0)                   /*Port Mode Register                              */
#define IRQ0 PMR1_BIT.b4                                   /*IRQ0 Pin Select                                 */
#define PDR5 *(volatile unsigned char *)0xFFD8             /*Port5 Data Register                             */
#define PDR5_BIT (*(struct BIT *)0xFFD8)                   /*Port5 Data Register                             */
#define P5B2 PDR5_BIT.b2                                   /*Output Level Select D1                          */
#define P5B1 PDR5_BIT.b1                                   /*Output Level Select C1                          */
#define P5B0 PDR5_BIT.b0                                   /*Output Level Select B1                          */
#define PDR6 *(volatile unsigned char *)0xFFD9             /*Port6 Data Register                             */
#define PCR5 *(volatile unsigned char *)0xFFE8             /*Port5 Control Register                          */
#define PCR6 *(volatile unsigned char *)0xFFE9             /*Port6 Control Register                          */
#pragma interrupt (timerz)
#pragma interrupt (irq0)
/**********************************************************/
/* Function Definition                                                                                       */
/**********************************************************/
extern void INIT (void );                                  /*SP Set                                          */
void main (void );
void timerz (void );
void irq0 (void );
/**********************************************************/
/*RAM define                                              */
/**********************************************************/
unsigned char mode;                                       /*User Flag Erea                                   */
unsigned char cnt;                                        /*User Flag Erea                                   */
unsigned char rot;                                        /*User Flag Erea                                   */
extern void _INITSCT();
/**********************************************************/
/*Vector Address                                          */
/**********************************************************/
#pragma section V1                                         /*VECTOR SECTOIN SET                              */
void (*const VEC_TBL1[])(void)={
                                                           /*0x00 -0x0f                                      */
INIT                                                       /*00 Reset                                        */
}
#pragma section V2                                         /*VECTOR SECTOIN SET                              */
void (*const VEC_TBL2[])(void)={
irq0                                                       /*1C IRQ0 Interrupt                               */
}
```

```
#pragma section V3                                         /*VECTOR SECTOIN SET                              */
void (*const VEC_TBL3[])(void)={
timerz                                                     /*38 Timer Z Interrupt                            */
}
#pragma section                                            /*P                                               */
/**********************************************************/
/*Main Program                                            */
/**********************************************************/
void main (void )
{
    _INITSCT();
    set_imask_ccr(1);                                      /*Interrupt Disable                               */
    mode =0x00;
    cnt =0x00;
    rot =0x00;
    IEG0 =1;                                               /*_IRQ   Falling Edge Detection                   */
    IEN0 =1;                                               /*_IRQ   Interrupt Reguest Enable                 */
    IRQ0 =1;
    TSTR =0xFC;                                            /*Timer Stop                                      */
    TOCR =0x00;                                            /*PWM Initial Output "0"                          */
    TCR0 =0x21;                                            /*GRA Compare-match Clear Mode,φ=1/2              */
    POCR0 =0xFF;                                           /*FTIOB0,FTIOC0,FTIOD0 High Active                */
    GRA0 =0x320;                                           /*Carieer Cycle 50us                              */
    GRB0 =0x230;                                           /*Duty 30%                                        */
    GRC0 =0x230;                                           /*Duty 30%                                        */
    GRD0    =0x230;                                        /*Duty 30%                                        */
    TCR1 =0xC2;                                            /*GRD1 Compare-match Clear,φ=1/4                  */
    TIER1 =0x08;                                           /*TGRD1 Compare-match Interrupt Enable            */
    GRD1 =0x9c40;                                          /*60 Degree cycle =10ms                           */
    TOE R =0xF1;                                           /*FTIOB0,FTIOC0,FTIOD0 Output Enable              */
    PDR5 =0x00;                                            /*P50,P51,P52 =Low                                */
    PDR6 =0x00;                                            /*P61,P62,P63 =Low                                */
    PCR5 =0x07;                                            /*P50,P51,P52 Output                              */
    PCR6 =0x0E;                                            /*P61,P62,P63 Output                              */
    PDR5 =0x06;                                            /*P50 =Low,P51 =High,P52 =High                    */
    TSTR =0xFF;                                            /*TCNT0,TCNT1 Start                               */
    set_imask_ccr(0);                                      /*Interrupt Enable                                */
    while(1){
    ;
    }
```

```
}
/**********************************************************/
/*Timer Z Interrupt                                       */
/**********************************************************/
void timerz (void )
{
    if (mode ==0x00 )
      {
    IMIFD_1 =0;                                            /*Clear IMIFD to 0                */
    switch(cnt){
    case 0x00:
         P5B0 =1;                                          /*U Phase High Output             */
         P5B1 =0;                                          /*V Phase Low Output              */
         break;

    case 0x01:
         TPMR =0x89;                                       /*_U Phase(FTIOB0)Output Enable   */
         break;

    case 0x02:
          P5B1 =1;                                         /*V Phase High Output             */
          P5B2 =0;                                         /*W Phase Low Output              */
          break;
    case 0x03:
          TPMR =0x8A;                                      /*_V Phase(FTIOC0)Output Enable   */
          break;

    case 0x04:
          P5B2 =1;                                         /*W Phase High Output             */
          P5B0 =0;                                         /*U Phase Low Output              */
          break;
    case 0x05:
          TPMR =0x8C;                                      /*_W Phase(FTIOD0)Output Enable   */
          break;
}

    cnt =cnt +1;
    if (cnt ==0x06)
      {
          cnt =0x00;
      }
    rot =rot +1;
    if (rot ==0x12)
    {
          TIORA1 =0xCC;                                    /*FTIOA1,FTIOB1 Rising Edge Interrupt Enable       */
          TIORC1 =0x8C;                                    /*FTIOC1 Rising Edge Interrupt Enable              */
          TIER1 =0xE7;

          mode =0x5 5;                                     /*Change initial rotation to nornal rotation       */
      }
}
else{
    if (IMIFA_1 ==1){
          IMIFA_1 =0;                                      /*Interrupt Flag Clear                             */
          if(IOA0_1 ==1 ){
                  TPMR =0x89;                              /*FTIOD0:disable,FTIOB0:enable                     */
                  IOA0_1 =0;                               /*Falling Edge Select                              */
                  }
          else{
                         P5B2 =1;                          /*W phase high,U phase low                         */
                         P5B0 =0;
                         IOA0_1 =1;                        /*Rising Edge Select                               */
                  }
          }
    else if (IMIFB_1 ==1){
                  IMIFB_1 =0;                              /*Interrupt Flag Clear                             */
                  if(IOB0_1 ==1 ){
                         TPMR =0x8A;                       /*FTIOB0:disable,FTIOC0:enable                     */
                         IOB0_1 =0;                        /*Falling Edge Select                              */
                         }
                  else{
                         P5B0 =1;                          /*U phase high,V phase low                         */
                         P5B1 =0;
                         IOB0_1 =1;                        /*Rising Edge Select                               */
                  }
          }
    else if (IMIFC_1 ==1){
                  IMIFC_1 =0;                              /*Interrupt Flag Clear                             */
                  if(IOC0_1 ==1 ){
                         TPMR =0x8C;                       /*FTIOC0:disable,FTIOD0:enable                     */
                         IOC0_1 =0;                        /*Falling Edge Select                              */
                         }
                  else{
                         P5B1 =1;                          /*V phase high,W phase low                         */
                         P5B2 =0;
                         IOC0_1 =1;                        /*Rising Edge Select                               */
                  }
          }
    }
}
/**********************************************************/
/*IRQ0 Interrupt                                          */
/**********************************************************/
void irq0 (void )
{
    IRRI0 =0;
    TOER =0xFF;                                            /*FTIOB0,FTIOC0,FTIOD0 Output Disable              */
    PDR5 =0x00;
    TIER1 =0xE0;
    while(1){
    ;
    }
}
```

Link Address Setting:

Section Name	Address
CV1	H'0000
CV2	H'001C
CV3	H'0036
P	H'0100
B	H'FDB80