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 (H8/3664)

Introduction

The H8/3664 is used to control a brushless DC motor in the way shown in Figure 1.1.

Target Device

H8/300H Tiny Series H8/3664

Contents

Feb. 2003

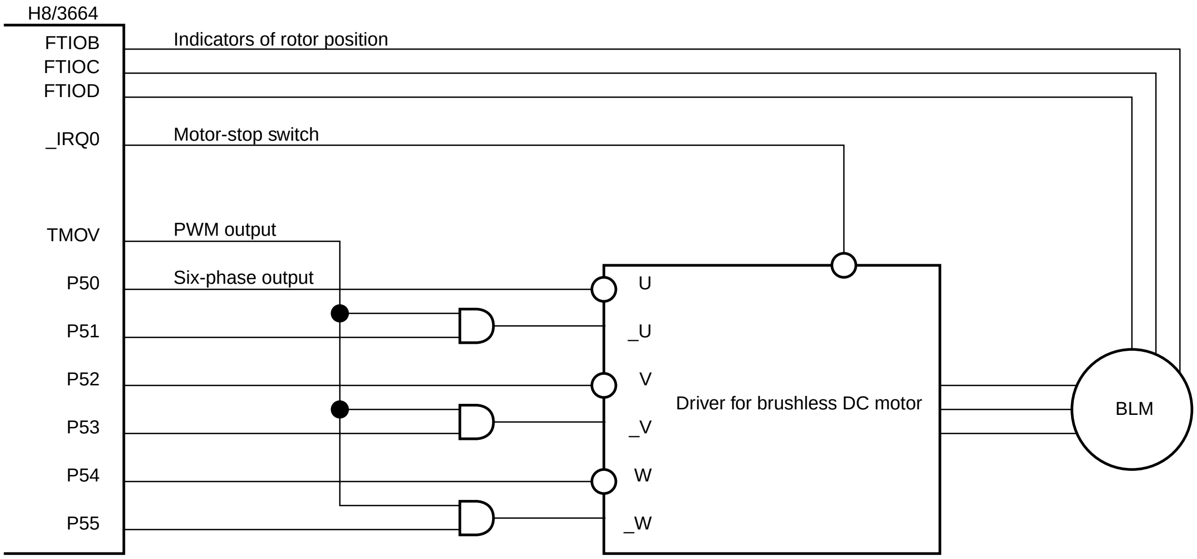
ADE- 502-118

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 H8/3664 is used to control a brushless DC motor in the way shown in Figure 1.1.
2. The H8/3664 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.
3. The H8/3664'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
FTIOB	: Input-capture pin B
FTIOC	: Input-capture pin C
FTIOD	: Input-capture pin D
IRQ0	: $\overline{IRQ0}$ pin (input)
TMOV	: Output pin for timer V waveform
P50 to P55	: Port pins P50 to P55 (outputs)

Figure 1.1 Set-up for Controlling a Brushless DC Motor

2. Design

1. The PWM waveforms for the motor are generated by the timer and output through I/O-port 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.
3. After switching the excited phase six 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 W 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, timers W (input capture/output comparison) and V (compare–match), an I/O port (Port 5) and the IRQ (external interrupt) of the H8/3664 are used to implement the functions required to control a brushless DC motor.

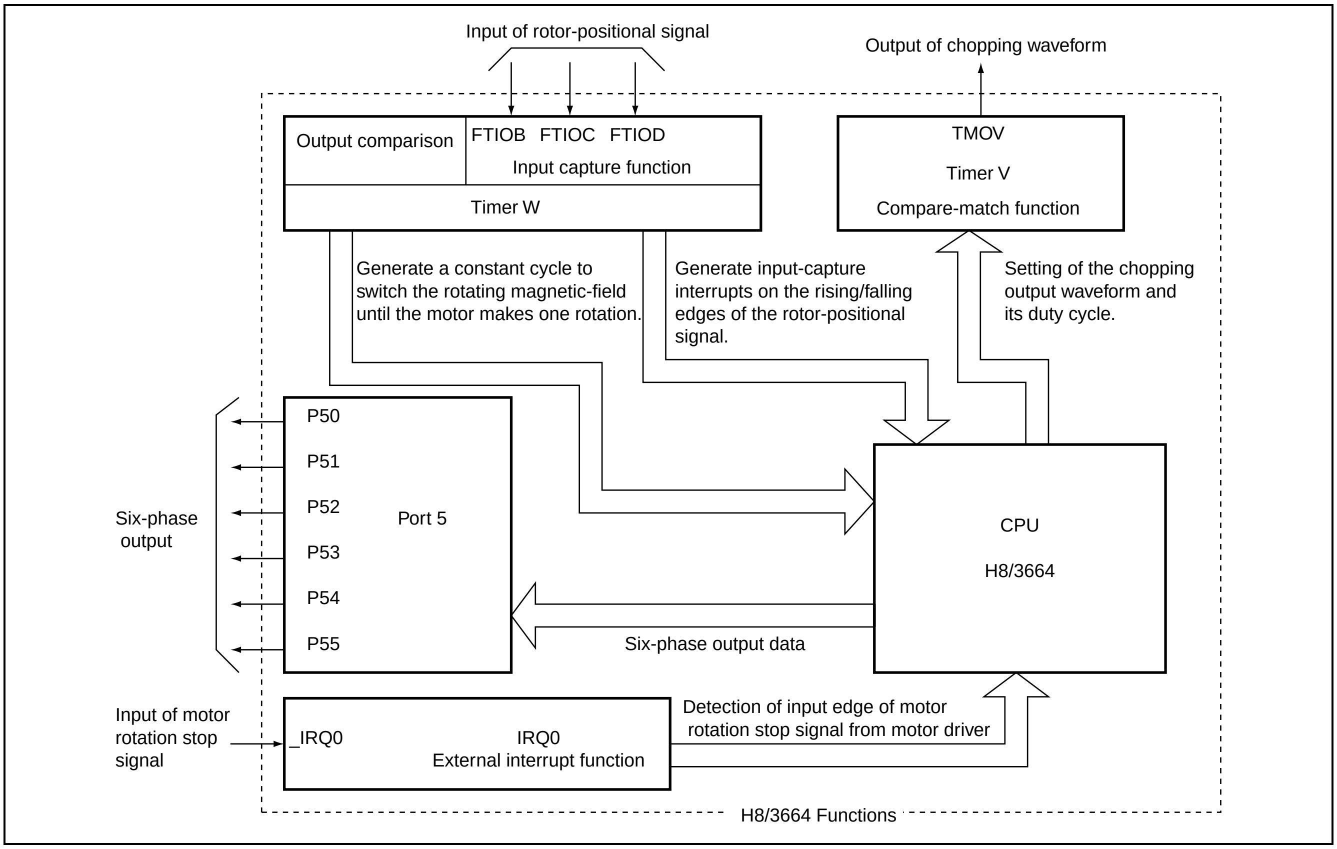

Input of rotor-positional signal
Output of chopping waveform
Output comparison
FTIOB FTIOC FTIOD
Input capture function
Timer W
TMOV
Timer V
Compare-match function
Generate a constant cycle to switch the rotating magnetic-field until the motor makes one rotation.
Generate input-capture interrupts on the rising/falling edges of the rotor-positional signal.
Setting of the chopping output waveform and its duty cycle.
P50
P51
P52
P53
P54
P55
Port 5
Six-phase output
CPU
H8/3664
Six-phase output data
Input of motor rotation stop signal
_IRQ0
IRQ0
External interrupt function
Detection of input edge of motor rotation stop signal from motor driver
H8/3664 Functions

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

The tasks performed by the H8/3664 functional blocks are outlined below.

— Timer W input capture: generates an interrupt request for the CPU on the rising and falling edges of the rotor-positional signal.

— Timer W 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 the motor's frequency of rotation) until the motor has made one rotation.

— Timer V compare–match function: generates the waveform for chopping control when the driver transistor turns on; this is output through the TMOV pin.

— Port 5 function: outputs six-phase data to the motor driver.

— IRQ external interrupt function: stops the motor in response to the external rotation-stop signal.

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

   1) Two functions of Timer W 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 W.

— The system clock (ϕ) is the standard clock that drives the CPU and peripheral functions. Prescaler S (PSS) frequency-divides the system clock to obtain ϕ/8192 to ϕ/2 signals for sending to the various peripheral modules.

— PSS is a 13-bit counter with clock input of ϕ and is incremented per clock cycle.

— The timer counter (TCNT) is a 16-bit readable/writable up-counter. The input clock is selectable from among four signals: those obtained by dividing the system-clock signal by 2, 4, and 8, and an external clock signal. The selection is made in TCRW (described below). In this sample task, the ϕ /8 signal is selected as the TCNT input clock.

— The timer control register (TCRW), an 8-bit readable/writable register, is used to select the input clock for TCNT.

— The timer status register (TSRW), an 8-bit register, controls each of the timer-driven interrupt-request signals.

— The timer interrupt enable register W (TIERW), an 8-bit readable/writable register, controls enabling/disabling of each of the timer-driven interrupt-request signals. In this sample task, interrupt requests by IMFA (IMIA), IMFB (IMIB), IMFC (IMIC), IMFD (IMID) are enabled.

— The timer I/O control registers (TIOR0, TIOR1) are used to select the functions of GRA, GRB, GRC and GRD, along with the edges for input captures B to D.

2) Timer W's input-capture function provides a way to generate interrupts on detecting the rising and falling edges of the rotor-positional signals. 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 B, C, and D (FTIOB, FTIOC, FTIOD).

— Input capture registers B, C, and D (GRB, GRC, GRD) are 16-bit readable/writable registers. When an edge is detected in input signal B, C or D, respectively, the value of TCNT is transferred to GRB, GRC or GRD, and IMFB, IMFC or IMFD of TSRW is set to "1". In this case, when the value of the corresponding IMIEB, IMIEC or IMIED bit in TIERW is "1", an interrupt request is sent to the CPU.

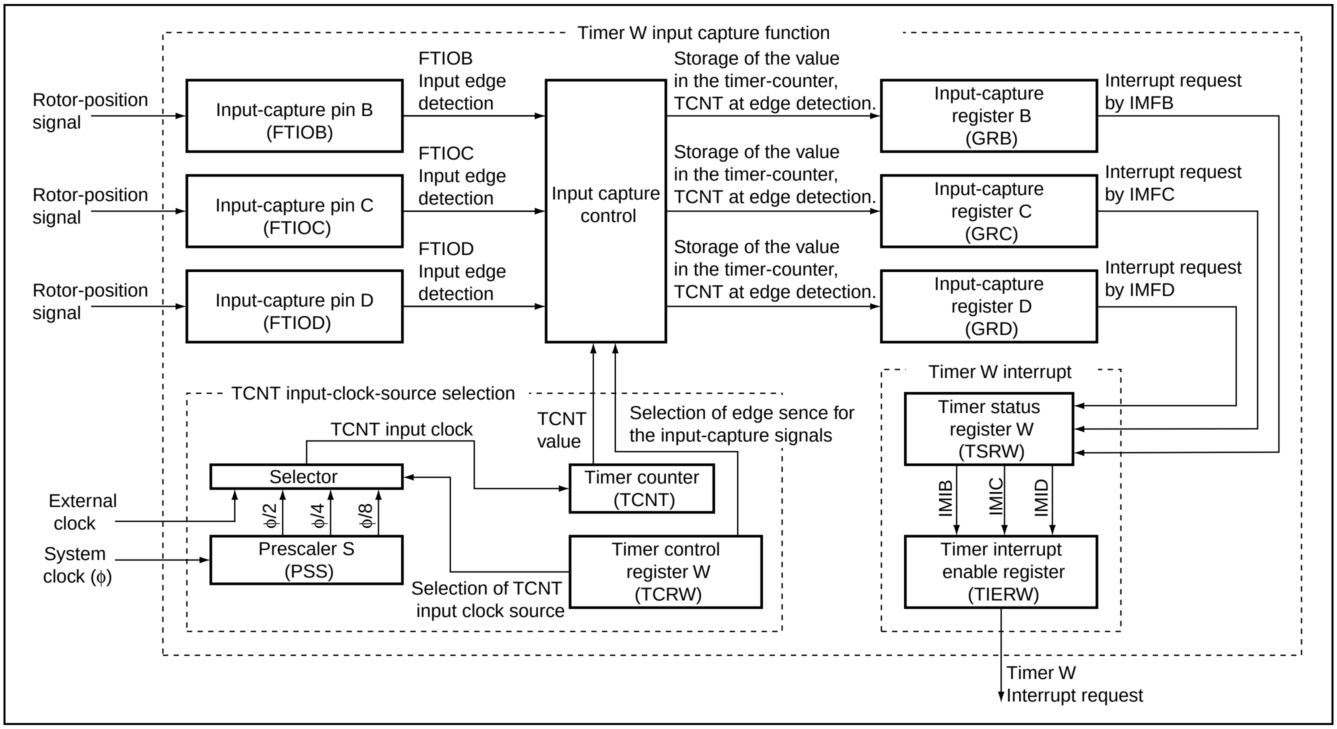

Timer W input capture function
Rotor-position signal
Input-capture pin B (FTIOB)
FTIOB Input edge detection
Rotor-position signal
Input-capture pin C (FTIOC)
FTIOC Input edge detection
Rotor-position signal
Input-capture pin D (FTIOD)
FTIOD Input edge detection
Input capture control
Storage of the value in the timer-counter, TCNT at edge detection.
Storage of the value in the timer-counter, TCNT at edge detection.
Storage of the value in the timer-counter, TCNT at edge detection.
Input-capture register B (GRB)
Input-capture register C (GRC)
Input-capture register D (GRD)
Interrupt request by IMFB
Interrupt request by IMFC
Interrupt request by IMFD
TCNT input-clock-source selection
TCNT input clock
TCNT value
Selection of edge sence for the input-capture signals
Selector
Timer counter (TCNT)
External clock
System clock (ϕ)
ϕ/2
ϕ/4
ϕ/8
Prescaler S (PSS)
Selection of TCNT input clock source
Timer control register W (TCRW)
Timer W interrupt
Timer status register W (TSRW)
IMIB
IMIC
IMID
Timer interrupt enable register (TIERW)
Timer W 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 W's 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 one rotation. Figure 3.3 is a block diagram of the control of these interrupt requests by timer W's output-comparison function. The operation of the blocks is described below.

Output–compare register A (GRA) is a 16-bit readable/writable register. The content of GRA is constantly compared with TCNT. When the two values match, IMFA in TSRW is set to "1". In this case, if the IMIEA bit of TIERW is set to "1", an interrupt request is sent to the CPU.

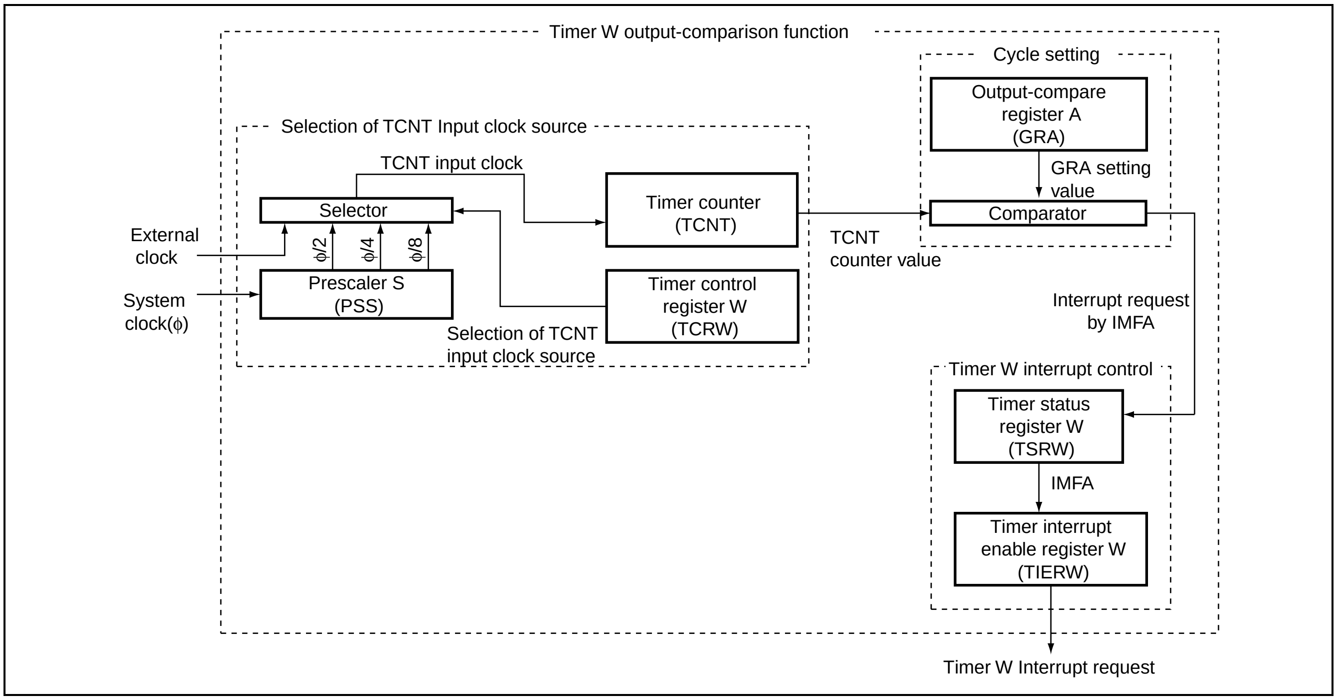

Timer W output-comparison function
Cycle setting
Output-compare register A (GRA)
GRA setting value
Comparator
Selection of TCNT Input clock source
TCNT input clock
Selector
Timer counter (TCNT)
External clock
System clock(ϕ)
ϕ/2
ϕ/4
ϕ/8
Prescaler S (PSS)
Timer control register W (TCRW)
Selection of TCNT input clock source
TCNT counter value
Interrupt request by IMFA
Timer W interrupt control
Timer status register W (TSRW)
IMFA
Timer interrupt enable register W (TIERW)
Timer W Interrupt request

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

4) Timer V's compare–match function is used to generate the chopping waveform for output on the TMOV pin. Figure 3.4 is a block diagram of the control of chopping-waveform output by timer V's compare–match function. The operation of the blocks is described below.

— Timer Counter V (TCNTV), an 8-bit readable/writable up-counter, is incremented by either an internal or external clock input. The clock-source selection is made in bits CKS2 to CKS0 of the TCRV0 register. The value in TCNTV is always readable/writable from the CPU, and is cleared by the input of an external reset signal or by compare–match signal A or B. The clearing signal is selected by bits CCLR1 and CCLR0 of TCRV0.

— Time Constant Registers A and B (TCORA, B) are 8-bit readable/writable registers. The contents of TCORA and B are constantly compared with the value in TCNTV; when either pair of values matches, the corresponding CMFA or B bit in TCSRV is set to "1". In this case, when CMIEA or B of TCSRV0 is "1", an interrupt is requested to the CPU. However, comparison is prohibited during the T3 state in the cycle of writing to the TCORA or B.

— Timer Control Register V0 (TCRV0), an 8-bit readable/writable register, is used to select the input clock, designate the signal that clears the TCNTV and enable interrupt requests. In this sample task, compare–match A is designated as the clearing signal. Selection of the input clock for TCNTV is made in combination with the ICKS0 bit of TCRV1.

— Timer Control/Status Register V (TCSRV), 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 Control Register V1 (TCRV1), an 8-bit readable/writable register, is used to select the input clock for TCNTV.

— The chopping waveform is output from the timer V output pin (TMOV).

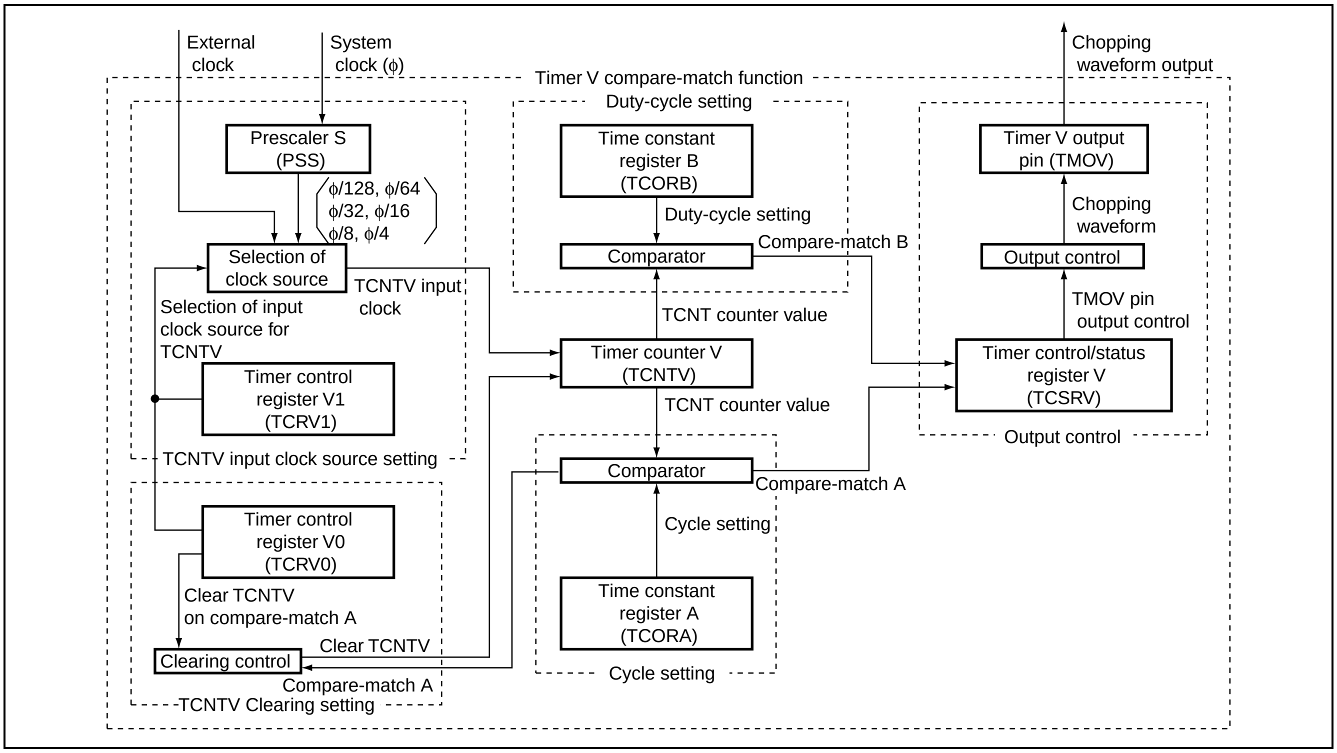


Figure 3.4 Block Diagram of Chopping-Waveform Output through the Timer V Compare–Match Function

5) The six driving signals are output through port 5. Figure 3.5 is a block diagram of six-phase control output through port 5.

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

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

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

— Port pin 53 (P53) acts as the output pin for the _V phase of the motor driver.

— Port pin 54 (P54) acts as the output pin for the W phase of the motor driver.

— Port pin 55 (P55) acts as the output pin for the _W phase of the motor driver.

— Port Register 5 (PDR5), the 8-bit data register for port 5, is used to store the data for each of pins P50 – P56. 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 a bit of PCR5 is "0", the corresponding pin is an input pin 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 P56 on port 5. In this sample task, pins P50 to P56 are set as output pins.

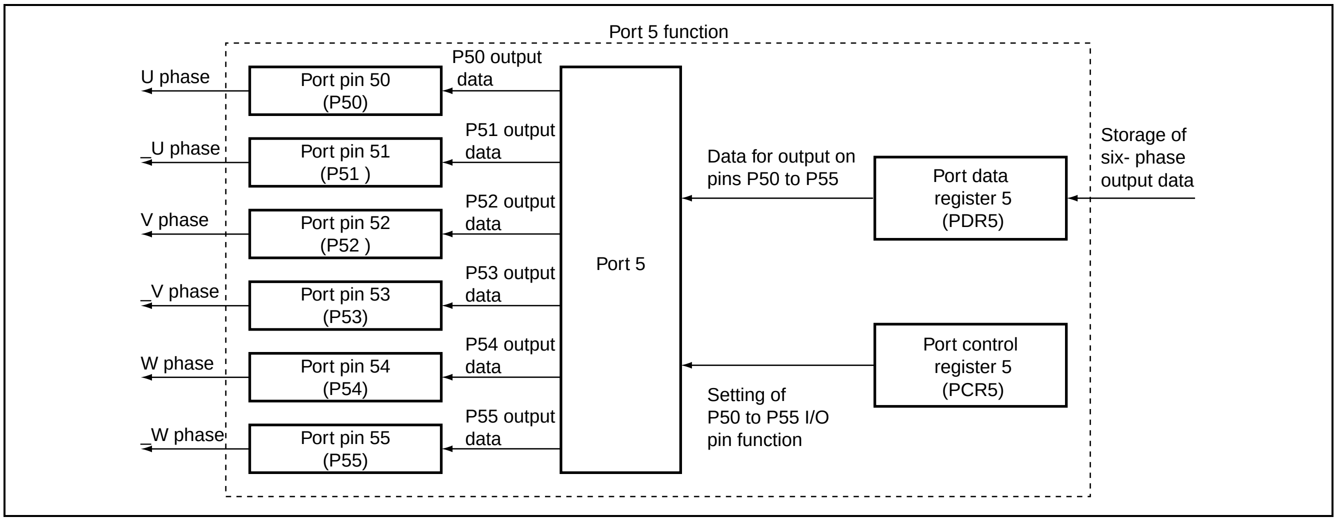


Figure 3.5 Block Diagram of Six-Phase Drive Output on Port 5

6) The IRQ external interrupt line is used to detect the motor-rotation stop 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 motor-rotation stop 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 edge 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 timer A, IRQ1, IRQ2, and IRQ3 interrupts are disabled.

— Interrupt Request Register 1 (IRR1) is an 8-bit readable/writable register of interrupt flags. When a 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.

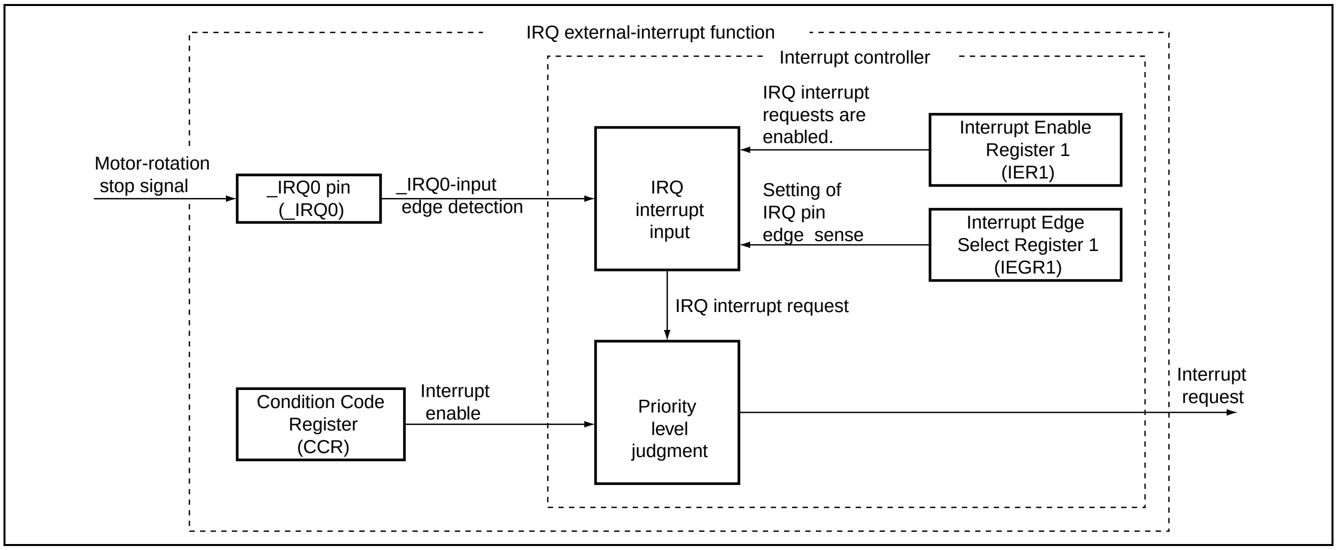

IRQ external-interrupt function
Interrupt controller
Motor-rotation stop signal
_IRQ0 pin (_IRQ0)
_IRQ0-input edge detection
IRQ interrupt input
IRQ interrupt requests are enabled.
Interrupt Enable Register 1 (IER1)
Setting of IRQ pin edge sense
Interrupt Edge Select Register 1 (IEGR1)
IRQ interrupt 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 the Motor-Rotation Stop Signal

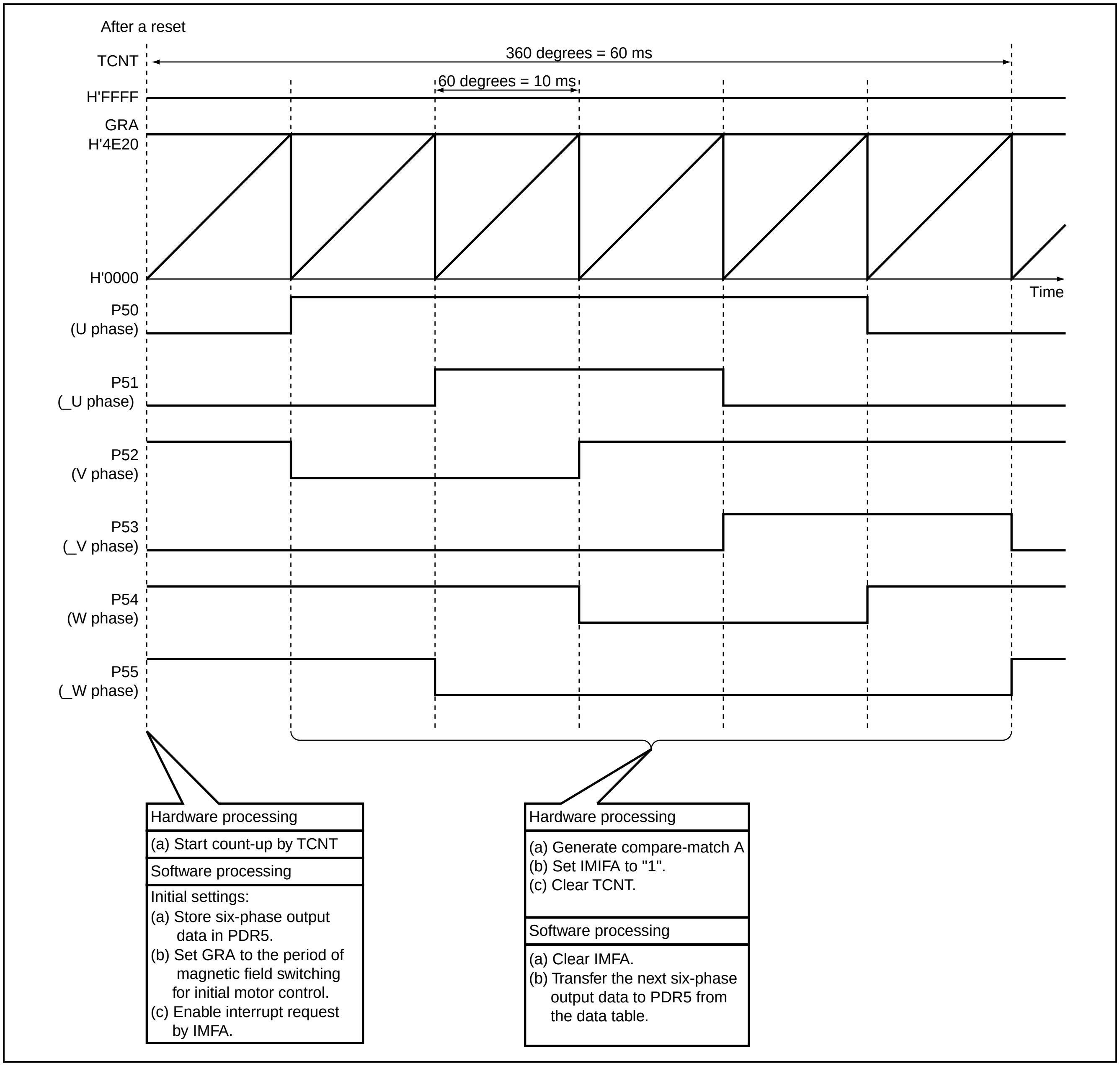
3. Table 3.1 lists the assignment of functions to H8/3664 elements for this sample task, i.e., control of the brushless DC motor.

Table 3.1 H8/3664 Function Assignment

H8/3664 Element	Assigned Function
FTIOB, C, D	• Rotor-positional signal input
TCNT	• 16-bit up-counter (input clock source is selected in TCRW)
TCRW	• Clock source setting for TCNT • Selection of the counter-clearing signal for TCNT
GRA	• Selection of the initial switching period for the motor's rotating magnetic field
TSRW	• Reflects the sources of timer W interrupt requests (IMFA, IMFB, IMFC, IMFD)
TIERW	• Enables/disables interrupt requests (IMFA, IMFB, IMFC, IMFD) from timer W
TMOV	• Chopping waveform output
TCNTV	• 8-bit up-counter
TCORA	• Setting of the period for the chopping waveform
TCORB	• Setting of the duty cycle for the chopping waveform
TCRV0	• Set TCNTV input clock source • Set TCNTV clearing condition • Enable/disable of timer V interrupt request (CMIA, CMIB)
TCRV1	• Setting of the input clock source for TCNTV
TCSRV	• Reflects the sources of timer V interrupt requests (CMIA, CMIB) • Selection of the output values for compare–match operations
P50 to P55	• Six-phase waveform output
PCR5	• Setting of pin functions for P50–P55
PDR5	• Storage of data for output on pins P50–P55 • Read to determine pin levels
_IRQ0	• Motor-rotation stop signal input
IEGR1	• Selection of the input edge for detection by _IRQ0
IER1	• Reflects the presence of the IRQ0 interrupt

4. Description of Operation

1. Figure 4.1 shows the principal of operation in initial motor control (until the motor has gone through its first rotation, switching of the rotating magnetic-field takes place at a constant period). Initial control of the brushless DC motor is through hardware and software processing by the H8/3664 as shown in figure 4.1.



Hardware processing

(a) Start count-up by TCNT

Software processing

Initial settings:
(a) Store six-phase output data in PDR5.
(b) Set GRA to the period of magnetic field switching for initial motor control.
(c) Enable interrupt request by IMFA.

Hardware processing

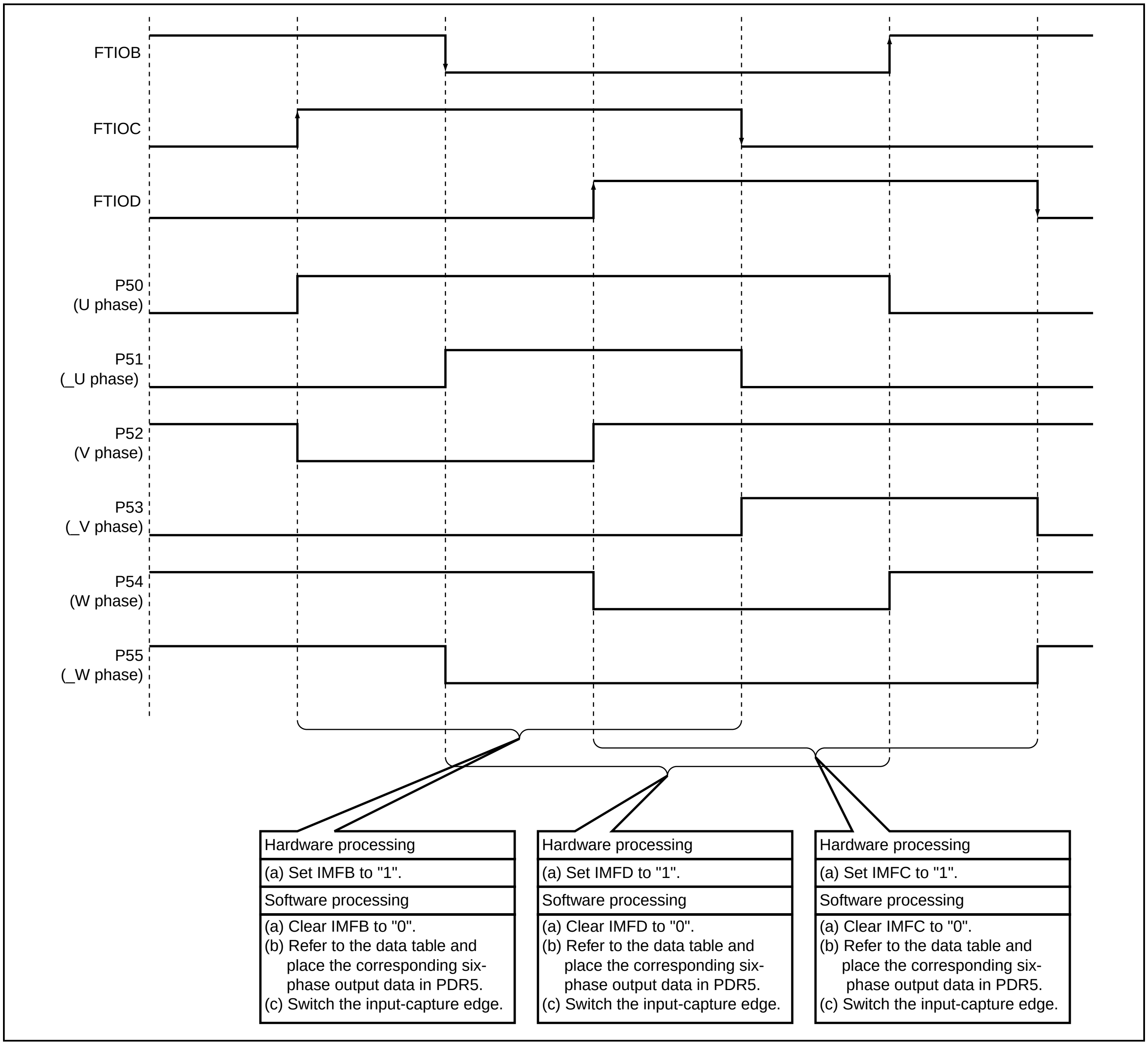
(a) Generate compare-match A
(b) Set IMIFA to "1".
(c) Clear TCNT.

Software processing

(a) Clear IMFA.
(b) Transfer the next six-phase output data to PDR5 from the data table.

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 H8/3664, based on the detected rotor-positional signal, as is shown in figure 4.2.



Hardware processing
(a) Set IMFB to "1".
Software processing
(a) Clear IMFB to "0".
(b) Refer to the data table and place the corresponding six-phase output data in PDR5.
(c) Switch the input-capture edge.

Hardware processing
(a) Set IMFD to "1".
Software processing
(a) Clear IMFD to "0".
(b) Refer to the data table and place the corresponding six-phase output data in PDR5.
(c) Switch the input-capture edge.

Hardware processing
(a) Set IMFC to "1".
Software processing
(a) Clear IMFC to "0".
(b) Refer to the data table and place the corresponding six-phase output data in PDR5.
(c) Switch the input-capture edge.

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. A chopping waveform generated by the timer V compare–match function is output on the TMOV pin. Figure 4.3 shows the principle of operation for output of the chopping waveform.

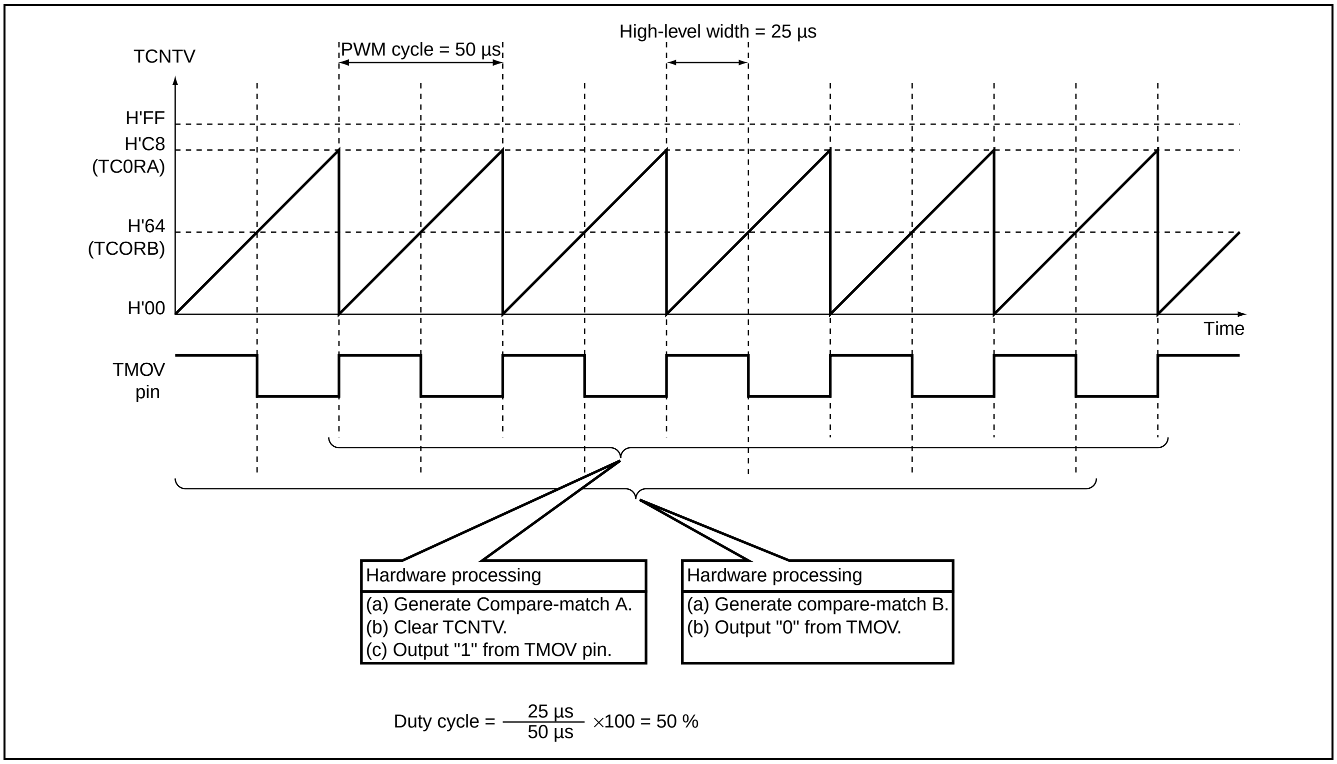
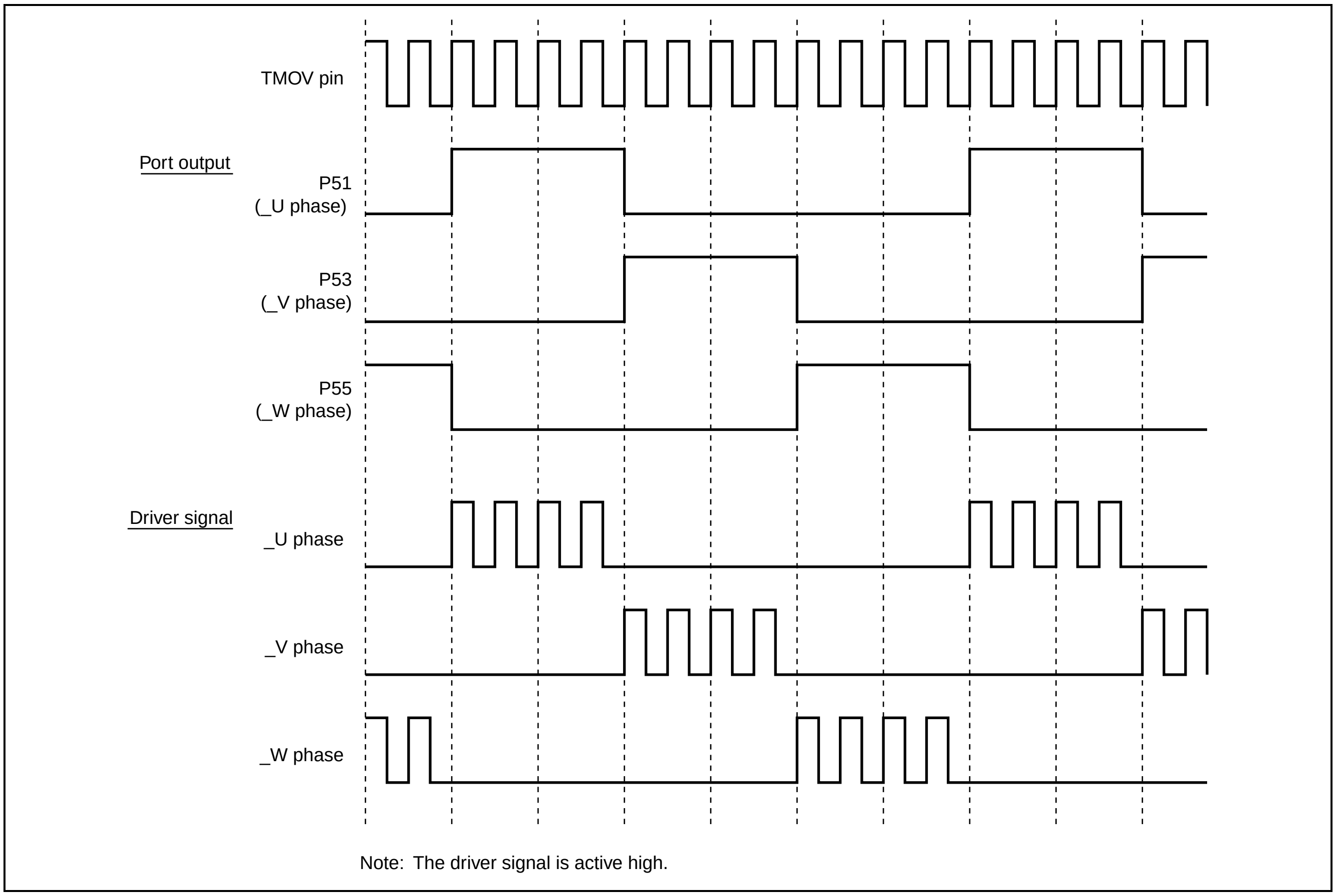

TCNTV
PWM cycle = 50 µs
High-level width = 25 µs
H'FF
H'C8
(TC0RA)
H'64
(TCORB)
H'00
Time
TMOV
pin
Hardware processing
(a) Generate Compare-match A.
(b) Clear TCNTV.
(c) Output "1" from TMOV pin.
Hardware processing
(a) Generate compare-match B.
(b) Output "0" from TMOV.
Duty cycle = 25 µs / 50 µs ×100 = 50 %

Figure 4.3 Output of the Chopping Waveform: Principle of Operation

In this sample task, a duty cycle of 50% is selected by the period setting of H'C8 in TCORA (50 µs) and the high-level width setting of H'64 in TCORB (25 µs).

4. The external AND circuits on the inverse-phase signals of the six-phase control output obtain the logical AND of the chopping waveform (on the TMOV pin) and magnetic-field rotation waveforms (port output) for output to the driver. Figure 4.4 shows the principle of operation applied to obtain the AND of the chopping and rotating-field waveforms.



Note: The driver signal is active high.

Figure 4.4 Taking the AND of the Inverse-Phase Port-Output Signals and the TMOV Output: Principle of Operation

5. Description

5.1 Description of Modules

Table 5.1 describes the software used in this sample task.

Table 5.1 Description of Modules

Module Name	Label Name	Function
Main routine	main	Initializes the stack pointer, sets up the IRQ_0 interrupt, starts counting by the timer, and sets up the individual interrupts.
Initialization	IO_INIT	Initializes the registers to be used, sets up timer W output comparison, timer V output comparison and the RAM used.
Timer W interrupt processing routine	intpt_tmw	Checks the source of a timer W interrupt and calls the appropriate routine for six-phase output control.
Initial motor control routine	tmw_3	Switches the rotating magnetic field on a constant cycle until the motor has gone through one rotation.
Feedback-driven routine to drive switching to produce the rotating-field signals (FTIOB)	SA_1	Switches the excited phase according to the rotor position detection signal which is output to the FTIOB pin from the motor.
Rotating-field switching routine by feedback signal (FTIOC)	SB_1	Switches the excited phase by rotor position detection signal which is output to the FTIOC pin from the motor.
Rotating-field switching routine by feedback signal (FTIOD)	SC_1	Switches the excited phase by rotor position detection signal which is output to the FTIOD pin from the motor.
IRQ_0 interrupt processing routine	r_stop	As the IRQ_0 interrupt handler, stops the motor by clearing the interrupt request flag and setting the error flag.

5.2 Description of Arguments

No arguments are used in this sample task.

5.3 Description of Internal Registers

Table 5.2 describes the internal registers used in this sample task.

Table 5.2 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'00
PCR5		Port Control Register 5: When PCR5n is "0", pin P5n acts as an input pin. (n=0 to 7) When PCR5n is "1", pin P5n acts as an output pin. (n=0 to 7)	H'FFE8	H'FF
TSRW	IMFA	Timer Control/Status Register (Compare-match Flag A): When IMFA is "0", compare-match A has not been generated. When IMFA is "1", compare-match A has been generated.	H'FF83 Bit 0	0
	IMFB	Timer Control/Status Register (Input Capture Flag B): When IMFB is "0", compare-match B has not been generated. When IMFB is "1", compare-match B has been generated.	H'FF83 Bit 1	0
	IMFC	Timer Control / Status Register (Input Capture Flag C): When IMFC is "0", compare-match C has not been generated. When IMFC is "1", compare-match C has been generated.	H'FF83 Bit 2	0
	IMFD	Timer Control / Status Register (Input Capture Flag D): When IMFD is "0", compare-match D has not been generated. When IMFD is "1", compare-match D has been generated.	H'FF83 Bit 3	0

Table 5.2 Description of Internal Registers (cont)

Register Name		Function	Address	Setting
TIERW	IMIEA	Timer Interrupt Enable Register (Compare-match Interrupt Enable A): When IMIEA is "0", IMFA interrupt requests are disabled. When IMIEA is "1", IMFA interrupt requests are enabled.	H'FF82 Bit 0	1
	IMIEB	Timer Interrupt Enable Register (Input Capture Interrupt Enable B): When IMIEB is "0", IMFB interrupt requests are disabled. When IMIEB is "1", IMFB interrupt requests are enabled.	H'FF82 Bit 1	0
	IMIEC	Timer Interrupt Enable Register (Input Capture Interrupt Enable C): When IMIEC is "0", IMFC interrupt requests are disabled. When IMIEC is "1", IMFC interrupt requests are enabled.	H'FF82 Bit 2	0
	IMIED	Timer Interrupt Enable Register (Input Capture Interrupt Enable D): When IMIED is "0", IMFD interrupt requests are disabled. When IMIED is "1", IMFD interrupt requests are enabled.	H'FF82 Bit 3	0

TIOR0	IOA0
	IOA1
		I/O Control A1 to 0
		If IOA2 is "0":
		When A1 and A0 are "00", no output on compare match.
			H'FF84
			Bit 0
			Bit 1
				IOA0=0
				IOA1=0

	IOA2	I/O Control A2: When IOA2 is "0", GRA functions as an output-compare register. When IOA2 is "1", GRA functions as an input-capture register.	H'FF84 Bit 2	0

	IOB0
	IOB1
		I/O Control B1 to 0
		If IOB2 is "1":
		When B1 and B0 are "00", input capture is made by GRB on the rising edge of the FTIOB signal.
		When B1 and B0 are "01", input capture is made by GRB on the rising edge of the FTIOB signal.
			H'FF84
			Bit 4
			Bit 5
				IOB0=1
				IOB1=0

	IOB2	I/O Control B2: When IOB2 is "0", GRB functions as an output-compare register. When IOB2 is "1", GRB functions as an input-capture register.	H'FF84 Bit 6	1*

Table 5.2 Description of Internal Registers (cont)

Register Name		Function	Address	Setting

TIOR1	IOC0
	IOC1
		I/O Control C1 to 0
		When IOC2 is "1":
		When IOC1 and IOC0 are "00", input capture is made by GRC on the rising edge of the FTIOC pin.
		When IOC1 and IOC0 are "01", input capture is made by GRC on the falling edge of the FTIOC pin.
			H'FF85
			Bit 0
			Bit 1
				IOC0=0
				IOC1=0

	IOC2	I/O Control C2: When IOC2 is "0", GRC functions as an output-compare register. When IOC2 is "1", GRC functions as an input-capture register.	H'FF85 Bit 2	1*

	IOD0
	IOD1
		I/O Control D1 to 0
		When IOD2 is "1":
		When IOD1 and IOD0 are "00", input capture is made by GRD on the rising edge of the FTIOD pin.
		When IOD1 and IOD0 are "00", input capture is made by GRD on the falling edge of the FTIOD pin.
			H'FF85
			Bit 4
			Bit 5
				IOD0=1
				IOD1=0

	IOD2	I/O Control D2: When IOD2 is "0", GRD functions as an output-compare register. When IOD2 is "1", GRD functions as an input-capture register.	H'FF85 Bit 6	1*
GRA		General Register A The register is used as the output-compare register and is constantly compared with the value in TCNT. When the two values match (compare-match), the IMFA flag of TSRW is set to "1".	H'FF88	H'4E20
GRB		General Register B The register is used as an input-capture register. When the input capture signal from FTIOB is detected, the value of TCNT is stored; the IMFB flag of TSRW is then set to "1".	H'FF8A	H'0000
GRC		General Register C The register is used as an input capture register. When the input capture signal is detected from FTIOC, the value of TCNT is stored, then IMFC flag of TSRW is set in "1".	H'FF8C	H'0000
GRD		General Register D The register is used as an input capture register. When the input capture signal is detected from FTIOD, the value of TCNT is stored, then IMFD flag of TSRW is set in "1".	H'FF8E	H'0000

Table 5.2 Description of Internal Registers (cont)

Register Name		Function	Address	Setting
TCSRV	OS0	Timer Control/Status Register V	H'FFA1	OS1=1
	OS1	(Output select 1 to 0):	Bit 0	OS0=0
		When OS1 and OS0 are "01", "0" is output from the TMOV pin when comparison of TCORA and TCNTV produces a match.	Bit 1	
		When OS1 and OS0 are "01", "1" is output from the TMOV pin when comparison of TCORA and TCNTV produces a match.		
	OS2	Timer Control/Status Register V	H'FFA1	OS2=1
	OS3	(Output select 3 to 2):	Bit 2	OS3=0
		When OS3 and OS2 are "01", "0" is output from the TMOV pin when comparison of TCORB and TCNTV produces a match.	Bit 3	
		When OS3 and OS2 are "01", "1" is output from the TMOV pin when comparison of TCORB and TCNTV produces a match.		
TCRV0	CCLR0	Counter Clear 1, 0:	H'FFA0	CCLR0=0
	CCLR1	When CCLR1 and CCLR0 are "01", TCNTV is cleared by compare-match A.	Bit 0 Bit 1	CCLA1=1
		When CCLR1 and CCLR0 are "10", TCNTV is cleared by compare-match B.		
TCORA		Time Constant Register A:	H'FFA2	H'C8
		When TCORA is "H'C8", compare-match A occurs when TCNTV is incremented to H'C8.		
TCORB		Time Constant Register B:	H'FFA3	H'64
		When TCORB is "H'64", compare-match B occurs when TCNTV is incremented to H'64.		
IEGR1	IEG0	Interrupt edge select register 1	H'FFF2	0
		(IRQ0 edge select):	Bit 0	
		When IEG0 is "0", falling edges in the IRQ0 pin's input are detected.		
		When IEG0 is "1", rising edges in the IRQ0 pin's input are detected.		
IENR1	IEN0	Interrupt enable register 1	H'FFF4	0
		(IRQ0 interrupt enable):	Bit 0	
		When IEN0 is "0", IRQ0 interrupt requests are disabled.		
		When IEN0 is "1", IRQ0 interrupt requests are enabled.		
IRR1	IRRI0	Interrupt request register 1	H'FFF6	0
		(IRQ0 interrupt request flag):	Bit 0	
		"0" indicates that an IRQ0 interrupt request has not been generated.		
		"1" indicates that an IRQ0 interrupt request has been generated.		

Note: * This value is initially "0".

5.4 Description of RAM Usage

Table 5.3 describes the RAM locations used in this sample task.

Table 5.3 Description of RAM Usage

Label Name	Function	Address	Label Name of Using Module
x	Counter for reference data table in initial motor control	H'FB80	IO_INIT
y	Counter for change of timer function	H'FB81	IO_INIT

5.5 Description of Data Table

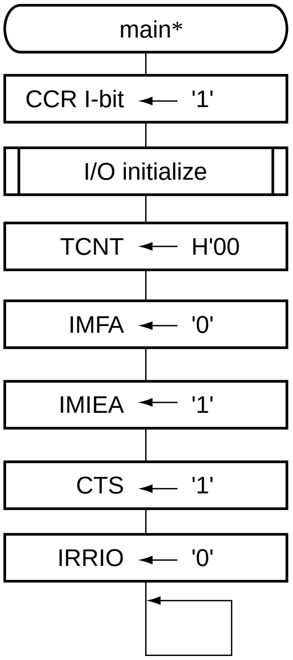
In this sample task, six patterns data on the table are output from port 5 at initial motor control. Table 5.4 shows the description of data table.

Table 5.4 Description of Data Table (data_0)

Data Name	Data	Output Pattern	Size	Address
data_0[0]	H'34	U = L, _U = L, V = H, _V = L, W = H, _W = H	1 byte	H'02D6
data_0[1]	H'31	U = H, _U = L, V = L, _V = L, W = H, _W = H	1 byte	H'02D7
data_0[2]	H'13	U = H, _U = H, V = L, _V = L, W = H, _W = L	1 byte	H'02D8
data_0[3]	H'07	U = H, _U = H, V = H, _V = L, W = L, _W = L	1 byte	H’02D9
data_0[4]	H'0D	U = H, _U = L, V = H, _V = H, W = L, _W = L	1 byte	H'02DA
data_0[5]	H'1C	U = L, _U = L, V = H, _V = H, W = H, _W = L	1 byte	H'02DB

6. Flowchart

1. Main routine



...... Set interrupt mask bit (I) of CCR to "1", masking interrupt request.

...... Initialize (IO_IINIT) the registers and RAM to be used.

...... Initialize TCNT.

...... Clear compare-match flag A to "0".

...... Enable interrupt requests by IMFA.

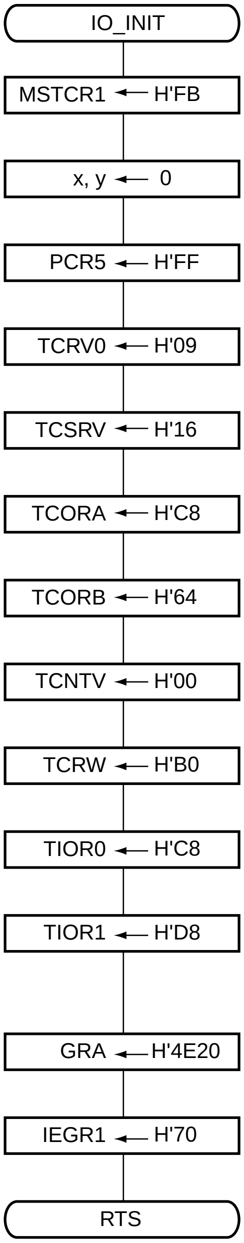
...... Start timer counting.

...... Clear the interrupt mask bit (I) of CCR to "0".

IRQ0 interrupt-request flag clear?

Note: * In this sample task, the stack pointer is set in INIT.SRC (assembly language).

2. Register initialization routine



...... Take timer W out of module standby mode.

...... Clear the counter "x" for data-table reference and
the counter "y" for initial motor control.

...... Set port 5 up as an output.

...... Set TCNTV to ϕ /4.
Select counter clearing by compare-match A.

...... Output 1 on a compare-match between TCORA and TCNTV.
Output 0 on a compare-match between TCORB and TCNTV.

...... Set the TMOV output pulse peiod to 50 µs.

...... Set the TMOV output pulse duty cycle to 50 %.

...... Clear TCNTV (start counting).

...... Set TCNT to ϕ / 8.
Select counter clearing by compare-match A.

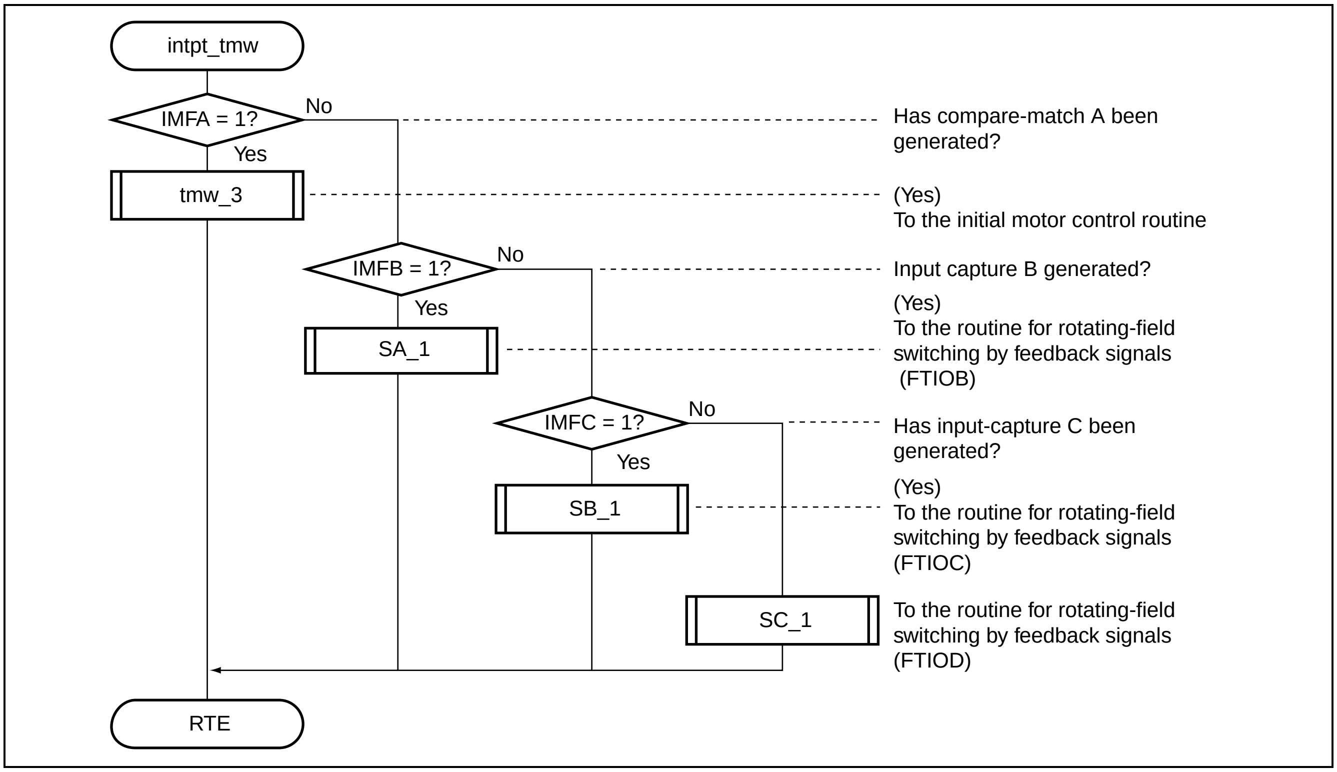
...... Set GRA as an output-compare register.
Set GRB as an input-capture register.
(input capture on rising edge of FTIOB)

...... Set GRC as an input-capture register.
(input capture on rising edge of FTIOC)
Set GRD as an input-capture register.
(input capture on falling edge of FTIOD)

...... Set interrupt generation cycle of timer W to10 ms.

...... Set up stopping of motor rotation on a falling edge of
the IRQ0 signal.

3. Timer W interrupt process



Has compare-match A been generated?

(Yes)
To the initial motor control routine

Input capture B generated?

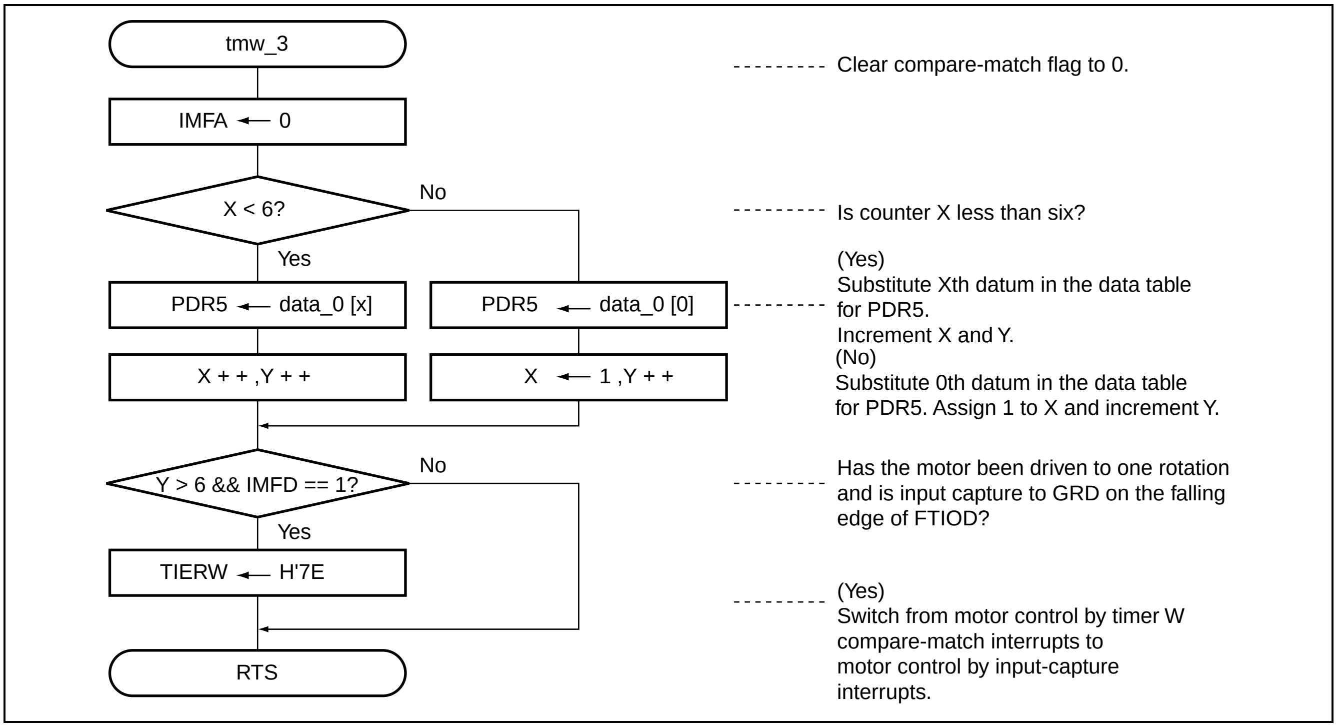
(Yes)
To the routine for rotating-field switching by feedback signals (FTIOB)

Has input-capture C been generated?

(Yes)
To the routine for rotating-field switching by feedback signals (FTIOC)

To the routine for rotating-field switching by feedback signals (FTIOD)

4. Initial motor control



Clear compare-match flag to 0.

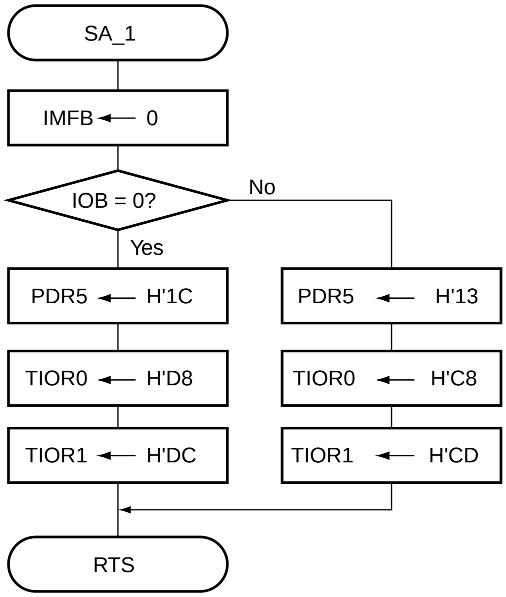
Is counter X less than six?

(Yes)
Substitute Xth datum in the data table for PDR5.
Increment X and Y.
(No)
Substitute 0th datum in the data table for PDR5. Assign 1 to X and increment Y.

Has the motor been driven to one rotation and is input capture to GRD on the falling edge of FTIOD?

(Yes)
Switch from motor control by timer W compare-match interrupts to motor control by input-capture interrupts.

5. Rotating-field switching routine by feedback signal (FTIOB)



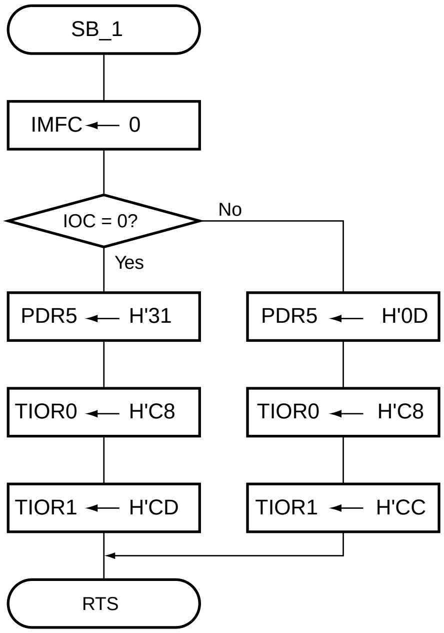
Clear input-capture flag B to "0".

Was input capture set for the rising edge of FTIOB?

(Yes)
Set the corresponding field-excitation data in PDR5.
Set the falling edge of FTIOB, rising edge of FTIOC and falling edge of FTIOD as the next condition for input capture.

(No)
Set the corresponding field-excitation data in PDR5.
Set the rising edge of FTIOB, falling edge of FTIOC and rising edge of FTIOD as the next condition for input capture.

6. Rotating-field switching routine by feedback signal (FTIOC)



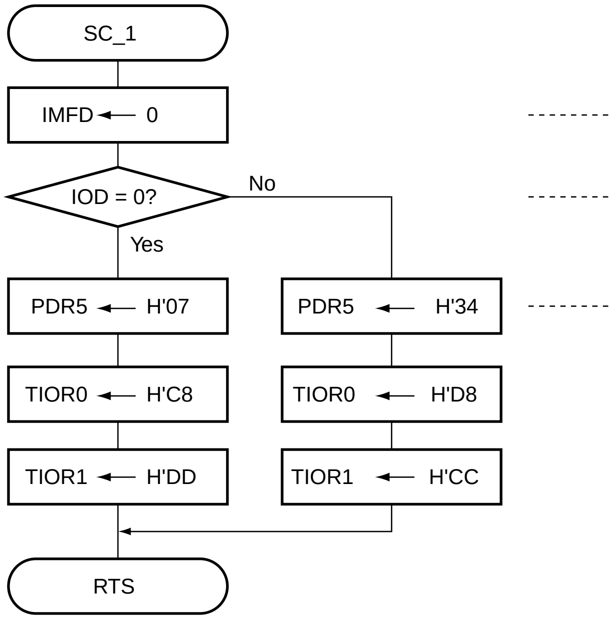
Clear input-capture flag C to "0".

Was input capture set for the rising edge of FTIOC?

(Yes)
Set the corresponding field-excitation data in PDR5.
Set the rising edge of FTIOB, falling edge of FTIOC and rising edge of FTIOD as the next condition for input capture.

(No)
Set the corresponding field-excitation data in PDR5.
Set the rising edge of FTIOB, rising edge of FTIOC and rising edge of FTIOD as the next condition for input capture.

7. Rotating-field switching routine by feedback signal (FTIOD)



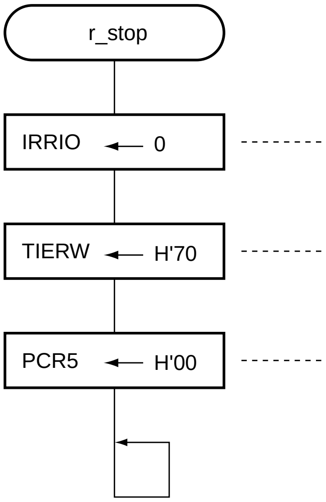
Clear input-capture flag D to "0".

Was input capture set for the rising edge of FTIOD?

(Yes)
Set the corresponding field-excitation data in PDR5.
Set the rising edge of FTIOB, falling edge of FTIOC and falling edge of FTIOD as the next condition for input capture.

(No)
Set the corresponding field-excitation data in PDR5.
Set the falling edge of FTIOB, rising edge of FTIOC and rising edge of FTIOD as the next condition for input capture.

8. Motor-rotation stop signal detection



Clear the IRQ0 interrupt request flag.

Disable the timer W compare-match and input-capture interrupts .

Set port 5 as an input port.

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
```

```
/* H8/300H TIny Series -H8/3664- Application Note                                                            */
/* Applied chapter-3                                                                                         */
/* Control of Brushless DC Motor                                                                             */
#include <machine.h>



/* Symbol Defnition                                                                                          */

struct BIT {
  unsigned char          b7:1;
  unsigned char          b6:1;
  unsigned char          b5:1;
  unsigned char          b4:1;
  unsigned char          b3:1;
  unsigned char          b2:1;
  unsigned char          b1:1;
  unsigned char          b0:1;
 };
#define        TMRW        *(volatile unsigned char  *)0xFF80        /* Timer W                              */
#define        TMRW_BIT    (*(struct BIT  *)0xFF80)
#define        CTS         TMRW_BIT.b7
#define        TCRW        *(volatile unsigned char  *)0xFF81
#define        TIERW       *(volatile unsigned char  *)0xFF82
#define        TIERW_BIT    (*(struct BIT  *)0xFF82)
#define        IMIED        TIERW_BIT.b3
#define        IMIEC        TIERW_BIT.b2
```

```
#define        IMIEB        TIERW_BIT.b1
#define        IMIEA        TIERW_BIT.b0
#define        TSRW         *(volatile unsigned char  *)0xFF83
#define        TSRW_BIT     (*(struct BIT  *)0xFF83)
#define        IMFD         TSRW_BIT.b3
#define        IMFC         TSRW_BIT.b2
#define        IMFB         TSRW_BIT.b1
#define        IMFA         TSRW_BIT.b0
#define        TIOR0        *(volatile unsigned char  *)0xFF84
#define        TIOR0_BIT    (*(struct BIT  *)0xFF84)
#define        IOB0         TIOR0_BIT.b4
#define        TIOR1        *(volatile unsigned char  *)0xFF85
#define        TIOR1_BIT    (*(struct BIT  *)0xFF85)
#define        IOD0         TIOR1_BIT.b4
#define        IOC0         TIOR1_BIT.b0
#define        TCNT        *(volatile unsigned int  *)0xFF86
#define        GRA         *(volatile unsigned int  *)0xFF88

#define        TCRV0       *(volatile unsigned char  *)0xFFA0       /* Timer V                              */
#define        TCSRV       *(volatile unsigned char  *)0xFFA1
#define        TCORA       *(volatile unsigned char  *)0xFFA2
#define        TCORB       *(volatile unsigned char  *)0xFFA3
#define        TCNTV       *(volatile unsigned char  *)0xFFA4
#define        TCRV1       *(volatile unsigned char  *)0xFFA5
#define        PUCR1       *(volatile unsigned char  *)0xFFD0       /* I/O Port                             */
#define        PUCR5       *(volatile unsigned char  *)0xFFD1
#define        PDR1        *(volatile unsigned char  *)0xFFD4
#define        PDR2        *(volatile unsigned char  *)0xFFD5
#define        PDR5        *(volatile unsigned char  *)0xFFD8
#define        PDR7        *(volatile unsigned char  *)0xFFDA
#define        PDR8        *(volatile unsigned char  *)0xFFDB
#define        PMR1        *(volatile unsigned char  *)0xFFE0
#define        PMR1_BIT    (*(struct BIT  *)0xFFE0)
#define        IRQ0        PMR1_BIT.b4
#define        PMR5       *(volatile unsigned char  *)0xFFE1
#define        PCR1       *(volatile unsigned char  *)0xFFE4
#define        PCR2       *(volatile unsigned char  *)0xFFE5
#define        PCR5       *(volatile unsigned char  *)0xFFE8
#define        PCR7       *(volatile unsigned char  *)0xFFEA
#define        PCR8       *(volatile unsigned char  *)0xFFEB
```

```
#define        IEGR1       *(volatile unsigned char  *)0xFFF2        /* IRQ0                                 */
#define        IENR1       *(volatile unsigned char  *)0xFFF4
#define        IENR_BIT    (*(struct BIT  *)0xFFF4
#define        IEN0        IENR_BIT.b0
#define        IRR1        *(volatile unsigned char  *)0xFFF6
#define        IRR1_BIT     (*(struct BIT  *)0xFFF6)
#define        IRRI0        IRR1_BIT.b0
#define        MSTCR1      *(volatile unsigned char  *)0xFFF9

#pragma interrupt   (intpt_tmw)                                      /* Timer W interrupt                    */
#pragma interrupt   (r_stop)                                         /* IRQ0                                 */
/* Function definition                                                                                       */
extern  void INIT (void);
void  main(void);
void  IO_INIT(void);
void  intpt_tmw(void);
void  r_stop(void);
void  tmw_3(void);
void  SA_1(void);
void  SB_1(void);
void  SC_1(void);



/* Data Table                                                                                                */
const unsigned char data_0[6]=
{
    0x34,
    0x31,
    0x13,
    0x07,
    0x0d,
    0x1c
};

 unsigned char x;                                                    /* Counter for data table reference     */
 unsigned char y;                                                    /*  Counter for timer function change   */
/* Vector Address                                                                                            */
#pragma section V1
void (*const VEC_TBL1[])(void) = {
    INIT                                                             /* H'0000 Reset vector                  */
 };
#pragma section V2                                                   /* Timer W interrupt                     */
void (*const VEC_TBL2[])(void) = {
        intpt_tmw
 };
#pragma section V3                                                   /* IRQ0 interrupt                        */
void (*const VEC_TBL3[])(void) = {
        r_stop
 };


#pragma section                                                      /* P                                     */
/* main routine                                                                                               */

void main(void)
{
   set_imask_ccr(1);                                                 /* CCR -Ibit = 1                         */
   IO_INIT();                                                        /* initialize                            */
   TCNT=0x00;
   IMFA = 0;
   IMIEA = 1;                                                        /* enable IMIA                           */
   CTS = 1;                                                          /* Start Timer count                     */
   IRRI0= 0;
   IEN0=1;
   set_imask_ccr(0);                                                 /* CCR -Ibit = 0                         */
   while(1)   {
       ;
   }

}


void IO_INIT(void)
{
    MSTCR1 = 0xf9;
    x=0;                                                             /* clear counter                         */
    y=0;

    PCR2 = 0x03;
    PDR2 = 0x03;
    PMR5 = 0x00;
    PUCR5 = 0x00;
    PDR5 = 0x00;
    PCR5 = 0xff;                                                     /* Output  Port5                        */

    PDR7 = 0x70;
    PCR7 = 0x70;

    PDR8 = 0xff;
    PCR8 = 0xf1;

    TCRV0 = 0x09;                                                    /*  Timer V initialize                  */
    TCRV1 = 0x02;                                                    /* clock = ϕ/4                          */
    TCSRV = 0x16;
    TCORA = 0xc8;                                                    /* PWM period = 50μs                    */
    TCORB = 0x64;                                                    /* PWM duty = 50%                       */
    TCNTV = 0x00;                                                    /* TCNTV start                          */

    TMRW = 0x48;                                                     /* Timer W initialize                   */
    TCRW = 0xb0;                                                     /* Clock = ϕ/8                          */
    TIERW = 0x70;
    TSRW = 0x70;
    TIOR0 = 0xc8;
    TIOR1 = 0xd8;
    GRA = 0x4e20;                                                    /* Port output period = 10.0ms          */

    IEGR1 =0x70;
 }
void r_stop(void)                                                    /*  IRQ0 interrupt routine              */
{
      IRRI0 = 0;
      TIERW = 0x70;
      PCR5 = 0x00;
      }
      while(1){
         ;
      }
 }
 void intpt_tmw(void)                                                /* Timer W interrupt routine            */
 {
   if(IMFA==1){                                                          /* Compare-match A ?                   */
     tmw_3();
   }
    if(IMFB==1){                                                         /* FTIOB Input capture ?               */
     SA_1();
         }
    if(IMFC==1){                                                         /* FTIOC Input capture ?               */
     SB_1();
        }
    if(IMFD==1){                                                         /* FTIOD Input capture ?               */
     SC_1();
         }
  }
void tmw_3(void)                                                         /* Compare-match A interrupt routine   */
 {

    IMFA =0;
    if(x < 6){
        PDR5 = data_0[x];
        x++;
        y++;
      }
      else{
       PDR5 = data_0[0];                                                 /* If  x > 6 = clear  x                */
        x=1;
        y++;
     }

   if(y > 0x06  && IMFB == 1 ){                                          /* 1rotated & input capture A ?        */
       TIERW = 0x7e;                                                     /* disable IMIA                        */
      }
  }


  void SA_1(void)                                                        /* FTIOB input capture routine         */
  {
    IMFB = 0;
    if(IOB0 == 0){
      PDR5 = 0x1c;
      TIOR0 = 0xd8;
      TIOR1 = 0xdc;
     }
     else{
      PDR5 = 0x13;
      TIOR0 = 0xc8;
      TIOR1 = 0xcd;
   }
 }
  void SB_1(void)                                                    /*FTIOC input capture routine            */
  {
   IMFC = 0;
   if(IOC0 == 0){
     PDR5 = 0x31;
     TIOR0 = 0xc8;
     TIOR1 = 0xcd;
    }
   else{
     PDR5 = 0x0d;
     TIOR0 = 0xc8;
     TIOR1 = 0xcc;
   }
 }
void SC_1(void)                                                      /* FTIOD input capture routine           */
  {
   IMFD = 0;
   if(IOD0 == 0){
     PDR5 = 0x07;
     TIOR0 = 0xc8;
     TIOR1 = 0xdd;
     }
   else{
     PDR5 = 0x34;
     TIOR0 = 0xd8;
     TIOR1 = 0xcc;
   }
  }
```

Link Address Setting:

Section Name	Address
CV1	H'0000
CV2	H'002A
CV3	H'001C
P	H'0100
B	H'FB80