R7F0C807

Stepper Motor Control

R1AN2008EC0100
Rev.1.00
Sep 30, 2014

Introduction

This application note describes the sample program for stepper motor control by using RTO (real-time output controller) function of R7F0C807.

Target Device

R7F0C807

When applying the sample program covered in this application note to another microcomputer with the same SFR (Special Function Register), modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Specifications

This application note describes an example of two 2-phase stepper motors control by using RTO (real-time output controller) function module of microcontroller R7F0C807.

Table 1.1 lists the peripheral functions to be used and their applications.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Use
Real-time output controller	Used to control two stepper motors output
TAU0 channel 0 and channel 1	Used to control stepper motor 1 by using PWM function
TAU0 channel 2 and channel 3	Used to control stepper motor 2 by using PWM function
12-bit interval timer	Used to generate 5 ms timer interrupt in order to start or stop the switch elimination operation
A/D converter	Used to measure the phase current.
External interrupt 0	Used for forced cut-off trigger with the external input signal.
I/O port P14	Used for the start/stop switch.

2. Operating Conditions

The sample code contained in this application note has been tested under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	R7F0C807
Operating frequency	● High-speed on-chip oscillator clock (f_{HOCO}) clock: 20 MHz (typ.) ● CPU/peripheral hardware clock (f_{CLK}): 20 MHz
Operating voltage	5.0 V (Operation enabled from 4.5 to 5.5 V.) SPOR detection operation (V_{SPOR}): rising edge 4.28V(typ.), falling edge 4.00V (min.)
Integrated development environment	Renesas Electronics Corporation CubeSuite+ V2.01.00
C compiler	Renesas Electronics Corporation CA78K0R V1.60

3. Description of the Hardware

3.1 Hardware Configuration Example

In this application note, Eight RTIO pins are used to rotate and stop stepper motors in the forward and reverse directions in 2-phase excitation mode. Starting and stopping the motor are controlled by input from Port P14, which is connected with a start/stop switch. Phase current measurement is realized by using 8-bit A/D converter. Every phase current signal is converted to voltage signal through a sampling resistor and is amplified by the operational amplifier. And then these signals are input to the A/D port of ANI2, ANI3, ANI4 and ANI7 respectively. Meanwhile, when an overcurrent is detected by the current detector circuit, a falling edge signal will be generated and trigger the interrupt of INTP0 to stop the stepper motor by using forced cut-off function.

Figure 3.1 shows an example of hardware configuration that is used for this application note.

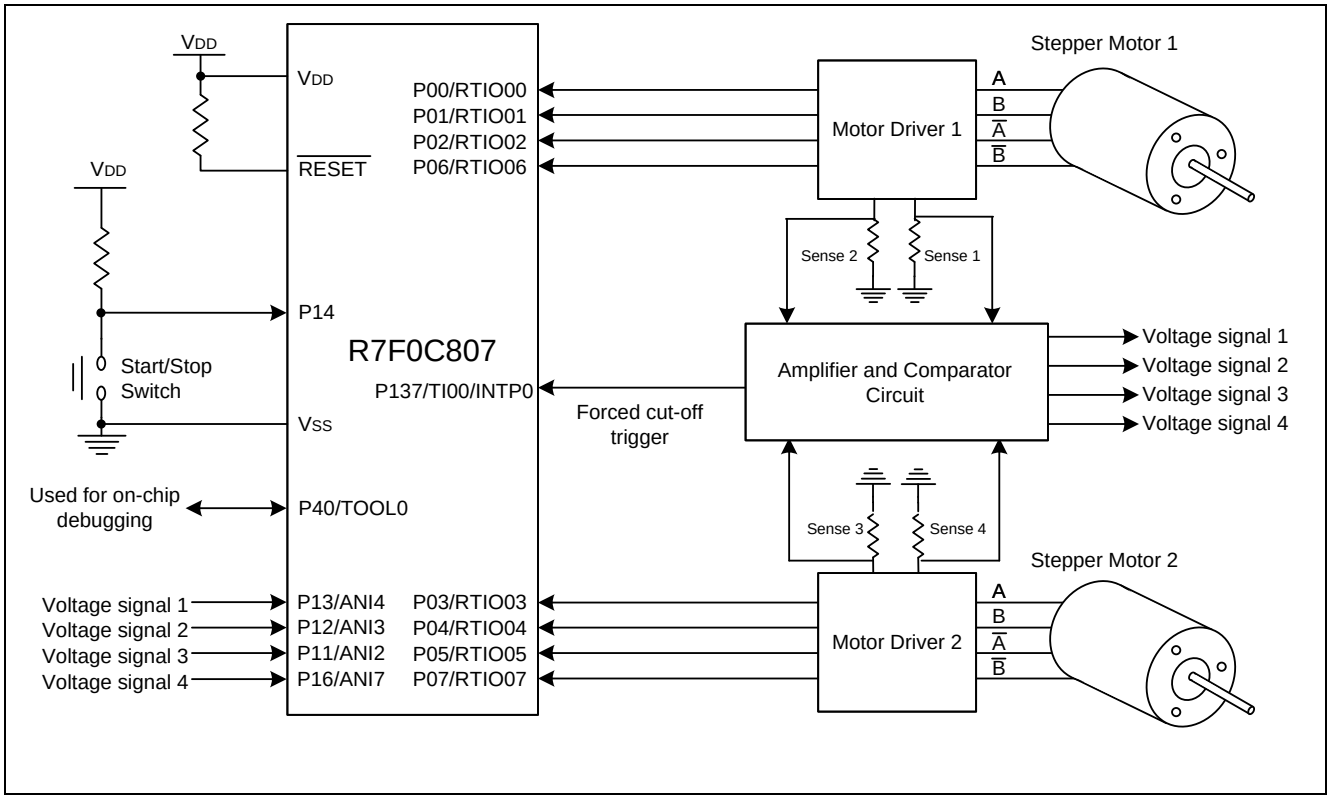


Figure 3.1 Hardware Configuration

Notes: 1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to V_{DD} or V_{SS} via a resistor).

2. V_{DD} must be held at not lower than the reset release voltage (V_{SPOR}) that is specified as SPOR.

3.2 List of Pins to be Used

Table 3.1 lists the pins to be used and their functions.

Table 3.1 Pins to be Used and Their Functions

Pin Name	I/O	Description
P00/RTIO00	Output	Output to phase A of stepper motor 1
P01/RTIO01	Output	Output to phase B of stepper motor 1
P02/RTIO02	Output	Output to phase $\overline{\text{A}}$ of stepper motor 1
P06/RTIO06	Output	Output to phase $\overline{\text{B}}$ of stepper motor 1
P03/RTIO03	Output	Output to phase A of stepper motor 2
P04/RTIO04	Output	Output to phase B of stepper motor 2
P05/RTIO05	Output	Output to phase $\overline{\text{A}}$ of stepper motor 2
P07/RTIO07	Output	Output to phase $\overline{\text{B}}$ of stepper motor 2
P137/INTP0	Input	Forced cut-off trigger signal
P14	Input	Start/stop switch
P13/ANI4	Input	Voltage signal from sampling resistor ‘sense 1’ (for phase B coil current measurement of stepper motor 1)
P12/ANI3	Input	Voltage signal from sampling resistor ‘sense 2’ (for phase A coil current measurement of stepper motor 1)
P11/ANI2	Input	Voltage signal from sampling resistor ‘sense 3’ (for phase B coil current measurement of stepper motor 2)
P16/ANI7	Input	Voltage signal from sampling resistor ‘sense 4’ (for phase A coil current measurement of stepper motor 2)

4. Principles of Stepper Motor Control Operation

4.1 Example of Stepper Motor Control Operation

Figure 4.1 shows an example of operating two 2-phase stepper motors by using two-phase excitation. As shown in figure 4.1, a high pulse causes the corresponding phase to be excited. The operation is outlined below.

- First, phases $\overline{A}$ and $\overline{B}$ are excited simultaneously. At this time, the rotor is positioned halfway between phases $\overline{A}$ and $\overline{B}$. Then, the two-phase excitation method rotates the rotor by exciting two adjacent phases (phases $\overline{B}$ and A, phases A and B, phases B and $\overline{A}$).
- For reverse rotation, the stepper motor is rotated by excitation in the following order: phases $\overline{A}$ and B, phases B and A, phases A and $\overline{B}$, phases $\overline{B}$ and $\overline{A}$.
- For stop operation, the stepper motor is stopped by keeping the last phase of a forward rotation or reverse rotation excited for a certain period of time.

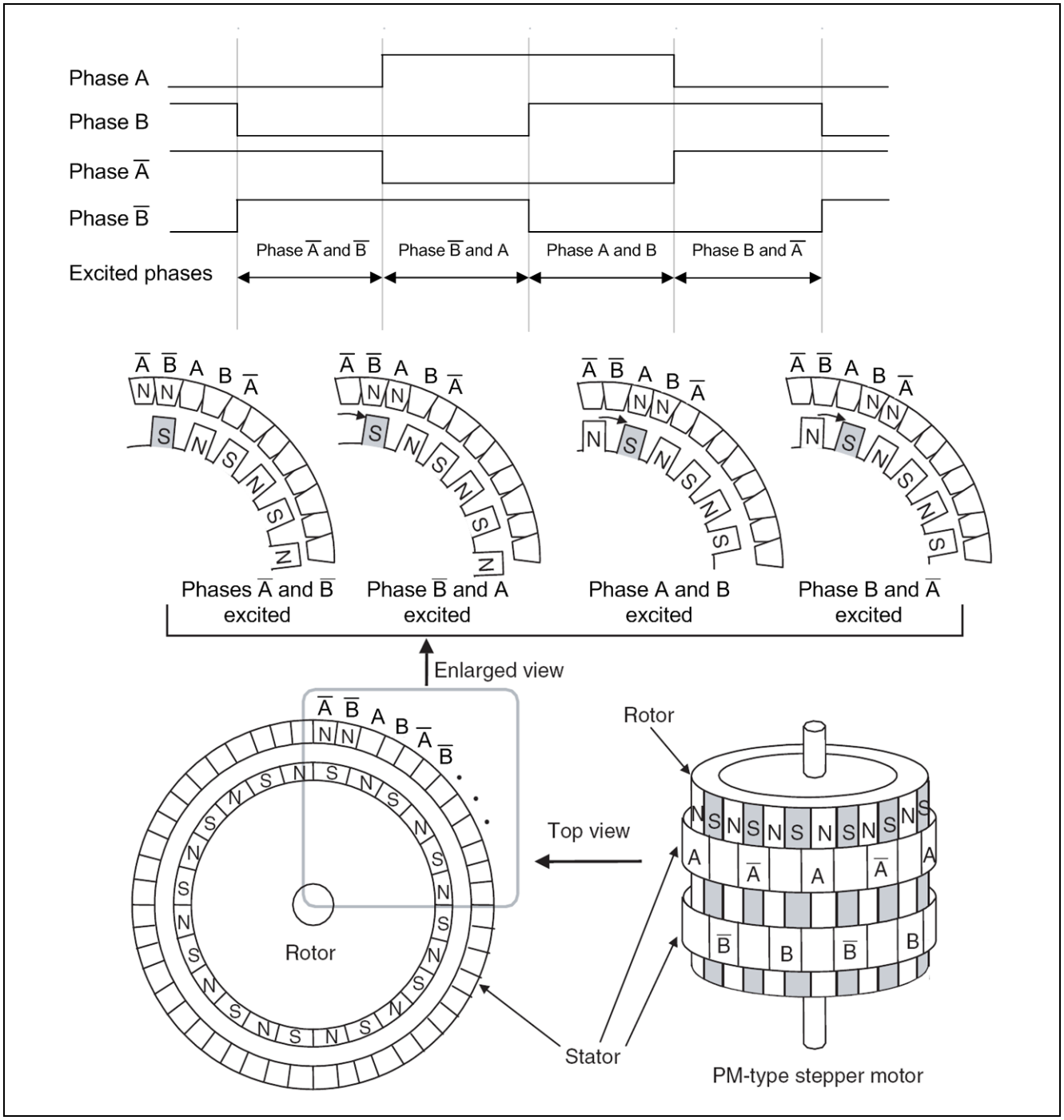


Figure 4.1 Example of Stepper Motor Control Operation

4.2 Slew-up and Slew-down Operation

Slew-up/slew-down operation maintains the synchronization of the motor. Out-of-synchronization means that if a series of short-cycle pulses are suddenly output to operate the motor, the motor may not be able to handle the load and will not rotate. Slew-up and slew-down operation is used to avoid this problem. The following explains the principle of the operation.

- The pulse cycles are gradually shortened to output the specified number of pulses (slew-up operation).
- The specified numbers of pulses are output at a regular pulse cycle (constant-speed operation).
- The pulse cycle is gradually extended to output the specified number of pulses (slew-down operation).

Figure 4.2 shows an example of the slew-up/slew-down operation.

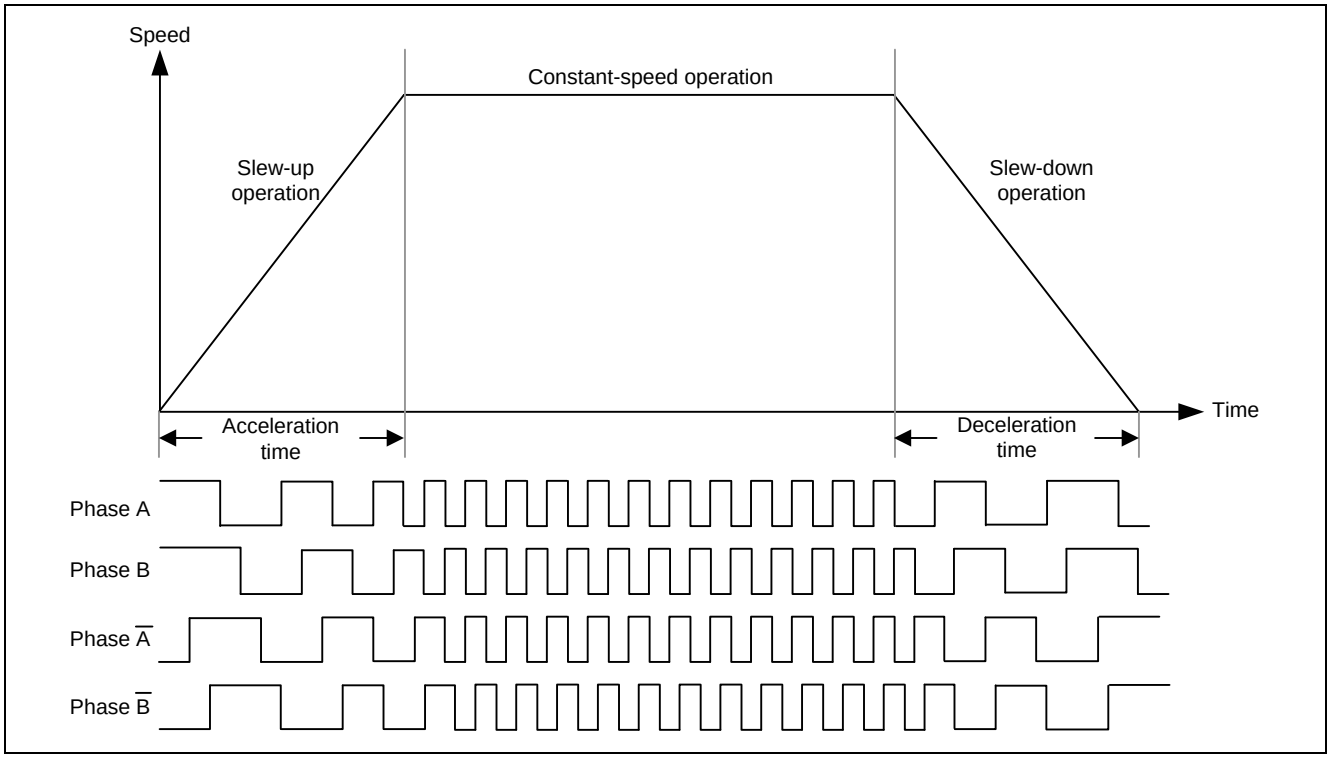


Figure 4.2 Example of Slew-up and Slew-down Operation

4.3 Controlling the Stepper Motors

4.3.1 Stepper Motor Controlling Timing

Table 4.1 shows the control timing for 2-phase stepper motor in this application.

Table 4.1 Control timing for 2-phase stepper motor

Step number	A	B	$\overline{A}$	$\overline{B}$
1	0	0	1	1
2	1	0	0	1
3	1	1	0	0
4	0	1	1	0

Switching the stepper motor excitation phase may produce flip delay, which may be a risk of damage to the driver device. Therefore, a non-overlap time is recommended to be inserted to prevent a through current from flowing when the output pattern is switched.

In this example, PWM output function of the TAU is used to control the rotation speed. TAU00 and TAU01 are used to control stepper motor 1, and TAU02 and TAU03 are used to control stepper motor 2. Pulses of any cycle and duty can be generated by combining the two TAU channels. TAU00 and TAU02 serve as master channels in interval timer mode, and TAU01 and TAU03 serve as slave channels in one-count mode. For example, when TAU00 is underflow (TCR00 reaches 0000H), INTTM00 interrupt generates. In the IMTTM00 interrupt subroutine, phase changing operation is executed by setting RTOOUTC0 and RTOOUTC1 registers. PWM duty (non-overlap control) in this example is set to 95%.

Figure 4.3 shows an example of stepper motor controlling timing waveforms.

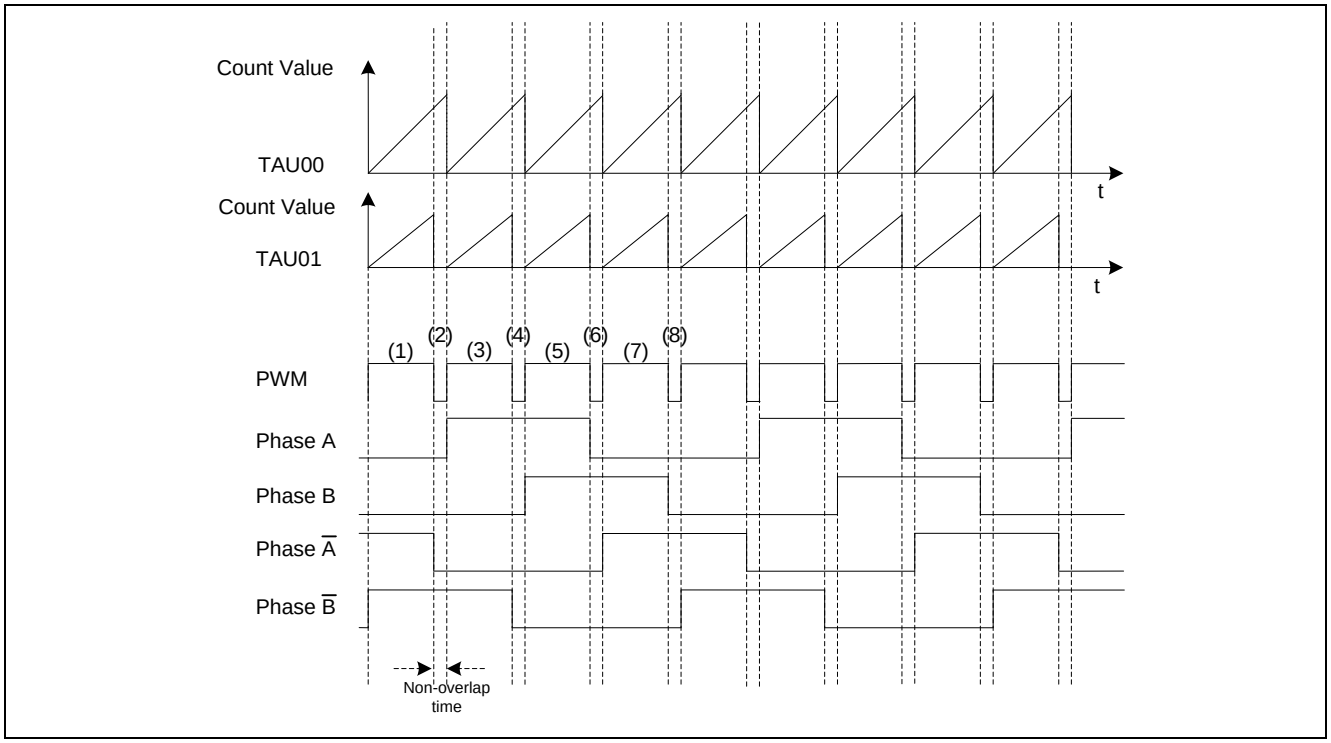


Figure 4.3 Example of Stepper Motor Controlling Timing Waveforms

Note: for detail of step (1) to (8), please refer to table 4.3 and 4.4.

4.3.2 Rotation speed calculation

The calculation formula of rotation speed is shown as below.

Phase changing frequency = 1/ (PWM period) = 1/ {(TDR0n + 1) x (Count clock period)}
(n = 0 or 2)

Motor rotation speed = (Phase changing frequency) x 60 / (360˚/ (stepping angle))

Setting samples are shown in table 4.2.

Table 4.2 Setting samples of stepper motor rotation speed Note

TRD0n value	Phase changing frequency (Hz)	Motor rotation speed (r/min)
49999	400	120
24999	800	240
12499	1600	480

Note: stepping angle is 1.8˚ and count clock period is 50ns in this application.
(n = 0 or 2)

4.3.3 Register Settings

Table 4.3 shows the register settings for rotating the stepper motor 1 forward.

Table 4.3 Example Register Settings for Rotating the Stepper Motor 1 Forward

State		Value Set to RTOSRC	Value Set to RTOOUTC0	Value Set to RTOOUTC1	Phase to be excited
	(1)	0x00	0x40	0x40	$\overline{A}$ $\overline{B}$
	(2)[Note]	0x00	0x00	0x40	$\overline{B}$
	(3)	0x00	0x10	0x40	$\overline{B}$ A
	(4)[Note]	0x00	0x10	0x00	A
	(5)	0x00	0x30	0x00	A B
	(6)[Note]	0x00	0x20	0x00	B
	(7)	0x00	0x60	0x00	B $\overline{A}$
	(8)[Note]	0x00	0x40	0x00	$\overline{A}$

Note: settings in (2), (4), (6), (8) are the register setting for non-overlap control.

Table 4.4 shows the register settings for rotating the stepper motor 2 forward.

Table 4.4 Example Register Settings for Rotating the Stepper Motor 2 Forward

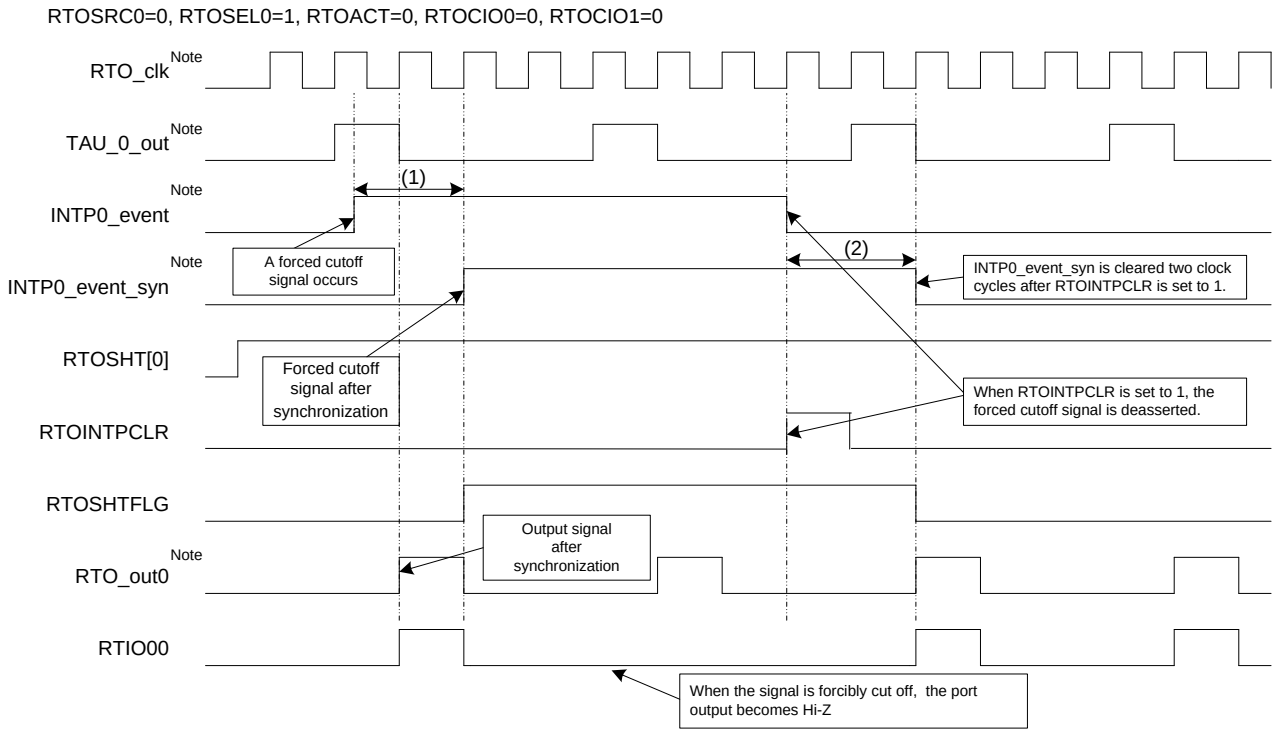
State		Value Set to RTOSRC	Value Set to RTOOUTC0	Value Set to RTOOUTC1	Phase to be excited
	(1)	0x00	0x00	0xa0	$\overline{A}$ $\overline{B}$
	(2)[Note]	0x00	0x00	0x80	$\overline{B}$
	(3)	0x00	0x80	0x80	$\overline{B}$ A
	(4)[Note]	0x00	0x80	0x00	A
	(5)	0x00	0x80	0x10	A B
	(6)[Note]	0x00	0x00	0x10	B
	(7)	0x00	0x00	0x30	B $\overline{A}$
	(8)[Note]	0x00	0x00	0x20	$\overline{A}$

Note: settings in (2), (4), (6), (8) are the register setting for non-overlap control.

For rotating the stepper motor in reverse direction, please use the reversed orders ((7), (6), (5), (4), (3), (2), (1), (8)) described in table 4.3 and table 4.4.

4.4 Forced Cut-off Processing

Figure 4.4 shows the controlling timing of entering and exiting the output cut-off state by using INTP0.



(1) The forced cut-off state is entered one or two clock cycles after INTP0 occurs.
(2) The forced cut-off state is exited two clock cycles after INTP0 is cleared.

Figure 4.4 Controlling Timing of Entering and Exiting the Output Cut-off State by Using INTP0

Note: R7F0C807 internal signal.

In figure 4.4, the output from RTIO00 is the same as TAU00. When the external interrupt INTP0 is input, the output from RTIO00 pin is cut off because RTOSHT0 bit is set to 1 (enabling forced cut-off). The status of the cut-off output is set to low level in this example.

The forced cut-off state can be released by setting RTOINTPCLR bit to 1.

Caution: In the integrated development environment of CubeSuite+, when user program is stopped by a hardware break, the output forced cut-off state is also excited. If user program continues running from the hardware break, the forced cut-off state will be released automatically.

5. Description of the Software

5.1 Operation Overview

The sample program covered in this application note describes an example of how two 2-phase and 4-wire stepper motors are controlled by using eight real-time output controller (RTO) pins.

(1) Execute system initialization. Initialize clock, port, external interrupt 0, 12-bit interval timer, TAU0, A/D converter and RTO module.
(2) Wait until the start switch is pressed.
(3) Execute the switch elimination operation. If the switch elimination is failed, go back to step (2). If the switch elimination is successful, execute the slew-up operation until the constant speed status is gotten and clear the cut-off status flag.
(4) Judge whether the forced cut-off happens. If it happens, go back to step (2).
(5) Measure the phase current of the motor driver.
(6) Judge whether the stop switch is pressed. If it is not pressed, go back to step (5).
(7) Execute the switch elimination operation. If the switch elimination is failed, go back to step (5). If the switch elimination is successful, execute the slew-down operation until the motor Stop.
(8) Go back to step (2).

5.2 List of Option Byte Settings

Table 5.1 summarizes the settings of the option bytes.

Table 5.1 Option Byte Settings

Address	Value	Description
000C0H	11100000B	Watchdog timer operation is stopped. (Count is stopped after reset.)
000C1H	11110011B	SPOR detection voltage: rising edge 4.28V(typ.), falling edge 4.00V(min.) P125/KR1/$\overline{\text{RESET}}$ port works as reset function.
000C2H	11111001B	HOCO: 20 MHz
000C3H	10000101B	On-chip debugging is enabled.

5.3 List of Constants

Table 5.2 lists the constants that are used in this sample program.

Table 5.2 Constants for the Sample Program

Constant	Setting	Description
SWITCH_ELIMINATION_TRUE	0x01	Switch elimination successful status
SWITCH_ELIMINATION_FALSE	0x00	Switch elimination failed status
REVERSE_ADDR	0x08	Base offset address for reverse rotation control value
N_REVERSE_ADDR	0x00	Base offset address for forward rotation control value
_5ms_INTV_TIMER_VALUE	0x4A	5 ms timer setting value for 12-bit interval timer
c_RTOOUTC1_Table2[16]	0x40, 0x40, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x40, 0x40, 0x00	Stepper motor 1 control value (for phase A and phase B) ● Rotating the stepper motor 1 forward: elements from 0 to 7 ● Rotating the stepper motor 1 reversely: elements from 8 to 15
c_RTOOUTC0_Table2[16]	0x40, 0x00, 0x10, 0x10, 0x30, 0x20, 0x60, 0x40, 0x60, 0x20, 0x30, 0x10, 0x10, 0x00, 0x40, 0x40	Stepper motor 1 control value (for phase $\overline{A}$ and phase $\overline{B}$) ● Rotating the stepper motor 1 forward: elements from 0 to 7 ● Rotating the stepper motor 1 reversely: elements from 8 to 15
c_RTOOUTC1_Table3[16]	0xa0,0x80, 0x80,0x00, 0x10,0x10, 0x30,0x20, 0x60,0x20, 0xb0,0x90, 0x90,0x00, 0x40,0x40	Stepper motor 2 control value (for phase A and phase B) ● Rotating the stepper motor 2 forward: elements from 0 to 7 ● Rotating the stepper motor 2 reversely: elements from 8 to 15
c_RTOOUTC0_Table3[16]	0x00,0x00, 0x80,0x80, 0x80,0x00, 0x00,0x00, 0x30,0x10, 0x10,0x00, 0xc0,0xc0, 0xe0,0x20	Stepper motor 2 control value (for phase $\overline{A}$ and phase $\overline{B}$) ● Rotating the stepper motor 2 forward: elements from 0 to 7 ● Rotating the stepper motor 2 reversely: elements from 8 to 15

5.4 List of Variables

Table 5.3 lists the global variables that are used in this sample program.

Table 5.3 Global variables for the Sample Program

Type	Variable Name	Contents	Function Used
uint16_t	g_Invert	Motor control variable for reverse rotation	main(void) Interrupt_INTTM00(void) Interrupt_INTTM01(void) Interrupt_INTTM02(void) Interrupt_INTTM03(void)
uint8_t	g_AD_Converter_Count	A/D conversion counter	main(void)
uint8_t	g_AD_Result_Buffer[4]	A/D conversion results	main(void)
__boolean	g_Button_Flag	Flag for switch being pressed	main(void)
__boolean	g_INTTM00_Flag	Interrupt of TAU0 channel 0 happening flag	stepper_motor_slew_up(uint16_t pace_up) stepper_motor_slew_down(uint16_t pace_down) Interrupt_INTTM00(void)
__boolean	g_INTTM02_Flag	Interrupt of TAU0 channel 2 happening flag	stepper_motor_slew_up(uint16_t pace_up) stepper_motor_slew_down(uint16_t pace_down) Interrupt_INTTM02(void)
uint16_t	g_Loop0	Stepper motor 1 phase changing counter	Interrupt_INTTM00(void) Interrupt_INTTM01(void)
uint16_t	g_Loop1	Stepper motor 2 phase changing counter	Interrupt_INTTM02(void) Interrupt_INTTM03(void)
uint16_t	g_Scan_Count	Switch elimination scan counter	eliminate_buffeting(void)

5.5 List of Functions

Table 5.4 summarizes the functions that are used in this sample program.

Table 5.4 Functions

Function Name	Outline
System_ini	The initial setting of peripheral functions
Port_ini	The initial setting of I/O ports
CLOCK_ini	The initial setting of clock
INTERVAL_TIMER_ini	The initial setting of the 12-bit interval timer
TAU0_ini	The initial setting of TAU0
INTP0_ini	The initial setting of external interrupt 0
AD_ini	The initial setting of A/D converter
RTO_ini	The initial setting of real-time I/O
main	main processing
eliminate_buffeting	Execute switch elimination operation
stepper_motor_slew_up	Execute stepper motor slew-up control
stepper_motor_slew_down	Execute stepper motor slew-down control
TAU00_TAU01_PWM_setting	Setting PWM period for TAU0 channel 0 and 1
TAU02_TAU03_PWM_setting	Setting PWM period for TAU0 channel 2 and 3
current_sampling	Execute motor current sampling
Interrupt_INTTM00	TAU0 channel 0 interrupt subroutine
Interrupt_INTTM01	TAU0 channel 1 interrupt subroutine
Interrupt_INTTM02	TAU0 channel 2 interrupt subroutine
Interrupt_INTTM03	TAU0 channel 3 interrupt subroutine
Interrupt_INTP0	External interrupt 0 interrupt subroutine
user	user program for forced cut-off processing

5.6 Function Specifications

This section describes the specifications for the functions that are used in this sample program.

[Function Name] System_ini

Synopsis	The initial setting of peripheral functions
Header	userdefine.h, r_systeminit.h, r_port.h, r_cgc.h, r_interval_timer.h, r_tau.h r_intp.h, r_ad.h, r_rto.h
Declaration	void System_ini(void);
Explanation	Execute the initialization setting for peripheral functions.
Arguments	None
Return value	None
Remarks	None

[Function Name] PORT_ini

Synopsis	The initial setting of I/O ports
Header	userdefine.h, r_port.h
Declaration	void PORT_ini(void);
Explanation	Execute the initialization setting for I/O ports.
Arguments	None
Return value	None
Remarks	None

[Function Name] CLOCK_ini

Synopsis	The initial setting of clock
Header	userdefine.h, r_cgc.h
Declaration	void CLOCK_ini(void);
Explanation	Execute the initialization setting for clock.
Arguments	None
Return value	None
Remarks	None

[Function Name] INTERVAL_TIMER_ini

Synopsis	The initial setting of 12-bit interval timer
Header	userdefine.h, r_interval_timer.h
Declaration	void INTERVAL_TIMER_ini (void);
Explanation	Execute the initialization setting for 12-bit interval timer.
Arguments	None
Return value	None
Remarks	None

[Function Name] TAU0_ini

Synopsis	The initial setting of TAU0
Header	userdefine.h, r_tau.h
Declaration	void TAU0_ini(void);
Explanation	Execute the initialization setting for TAU0.
Arguments	None
Return value	None
Remarks	None

[Function Name] INTP0_ini

Synopsis	The initial setting of external interrupt 0
Header	userdefine.h, r_intp.h
Declaration	void INTP0_ini(void);
Explanation	Execute the initialization setting for INTP0.
Arguments	None
Return value	None
Remarks	None

[Function Name] AD_ini

Synopsis	The initial setting of A/D converter
Header	userdefine.h, r_ad.h
Declaration	void AD_ini(void);
Explanation	Execute the initialization setting for A/D converter.
Arguments	None
Return value	None
Remarks	None

[Function Name] RTO_ini

Synopsis	The initial setting of RTO
Header	userdefine.h, r_rto.h
Declaration	void RTO_ini(void);
Explanation	Execute the initialization setting for RTO.
Arguments	None
Return value	None
Remarks	None

[Function Name] main

Synopsis	Main processing
Header	userdefine.h, r_systeminit.h, motor.h
Declaration	void main(void);
Explanation	Execute the main processing of stepper motor control.
Arguments	None
Return value	None
Remarks	None

[Function Name] eliminate_buffeting

Synopsis	Execute switch elimination operation.
Header	userdefine.h, r_systeminit.h, motor.h
Declaration	__boolean eliminate_buffeting(void);
Explanation	Start or stop switch elimination operation.
Arguments	None
Return value	[SWITCH_ELIMINATION_TRUE]: Switch elimination operation success. [SWITCH_ELIMINATION_FALSE]: Switch elimination operation fail.
Remarks	None

[Function Name] stepper_motor_slew_up

Synopsis	Execute stepper motor slew-up control
Header	userdefine.h, motor.h, r_tau.h
Declaration	void stepper_motor_slew_up(uint16_t pace_up);
Explanation	Execute slew-up operation of stepper motor 1 and stepper motor 2.
Arguments	pace_up: slew-up step number
Return value	None
Remarks	None

[Function Name] stepper_motor_slew_down

Synopsis	Execute stepper motor slew-down control
Header	userdefine.h, motor.h, r_tau.h
Declaration	void stepper_motor_slew_down(uint16_t pace_down);
Explanation	Execute slew-down operation of stepper motor 1 and stepper motor 2.
Arguments	pace_down: slew-down step number
Return value	None
Remarks	None

[Function Name] TAU00_TAU01_PWM_setting

Synopsis	Setting PWM period for TAU0 channel 0 and 1
Header	userdefine.h, r_tau.h
Declaration	void TAU00_TAU01_PWM_setting(uint16_t period_motor1);
Explanation	Execute PWM period and duty setting for TAU0 channel 0 and 1.
Arguments	period_motor1: PWM period for stepper motor 1
Return value	None
Remarks	None

[Function Name] TAU02_TAU03_PWM_setting

Synopsis	Setting PWM period for TAU0 channel 2 and 3
Header	userdefine.h, r_tau.h
Declaration	void TAU02_TAU03_PWM_setting(uint16_t period_motor2);
Explanation	Execute PWM period and duty setting for TAU0 channel 2 and 3.
Arguments	period_motor2: PWM period for stepper motor 2
Return value	None
Remarks	None

[Function Name] current_sampling

Synopsis	Motor current sampling
Header	userdefine.h, r_systeminit.h, motor.h
Declaration	void current_sampling(void);
Explanation	Execute motor current sampling processing for stepper motor 1 and stepper motor 2.
Arguments	None
Return value	None
Remarks	None

[Function Name] Interrupt_INTTM00

Synopsis	TAU0 channel 0 interrupt subroutine
Header	r_vect.h, userdefine.h
Declaration	__interrupt void Interrupt_INTTM00(void);
Explanation	Execute phase change of phase A and phase B for stepper motor 1.
Arguments	None
Return value	None
Remarks	None

[Function Name] Interrupt_INTTM01

Synopsis	TAU0 channel 1 interrupt subroutine
Header	r_vect.h, userdefine.h
Declaration	__interrupt void Interrupt_INTTM01(void);
Explanation	Execute phase change of phase $\overline{A}$ and phase $\overline{B}$ for stepper motor 1.
Arguments	None
Return value	None
Remarks	None

[Function Name] Interrupt_INTTM02

Synopsis	TAU0 channel 2 interrupt subroutine
Header	r_vect.h, userdefine.h
Declaration	__interrupt void Interrupt_INTTM02(void);
Explanation	Execute phase change of phase A and phase B for stepper motor 2.
Arguments	None
Return value	None
Remarks	None

[Function Name] Interrupt_INTTM03

Synopsis	TAU0 channel 3 interrupt subroutine
Header	r_vect.h, userdefine.h
Declaration	__interrupt void Interrupt_INTTM03(void);
Explanation	Execute phase change of phase $\overline{A}$ and phase $\overline{B}$ for stepper motor 2.
Arguments	None
Return value	None
Remarks	None

[Function Name] Interrupt_INTP0

Synopsis	External interrupt 0 interrupt subroutine
Header	r_vect.h, userdefine.h, user.h
Declaration	__interrupt void Interrupt_INTP0(void);
Explanation	Execute the forced cut-off processing.
Arguments	None
Return value	None
Remarks	None

[Function Name] user

Synopsis	User program for forced cut-off processing
Header	userdefine.h, user.h
Declaration	void user (void);
Explanation	When forced cut-off happens, user can add his own program here to process related status.
Arguments	None
Return value	None
Remarks	None

5.7 Flowcharts

5.7.1 Initial Setting of Peripheral Functions

Figure 5.1 shows the flowchart for the initial setting of peripheral functions.

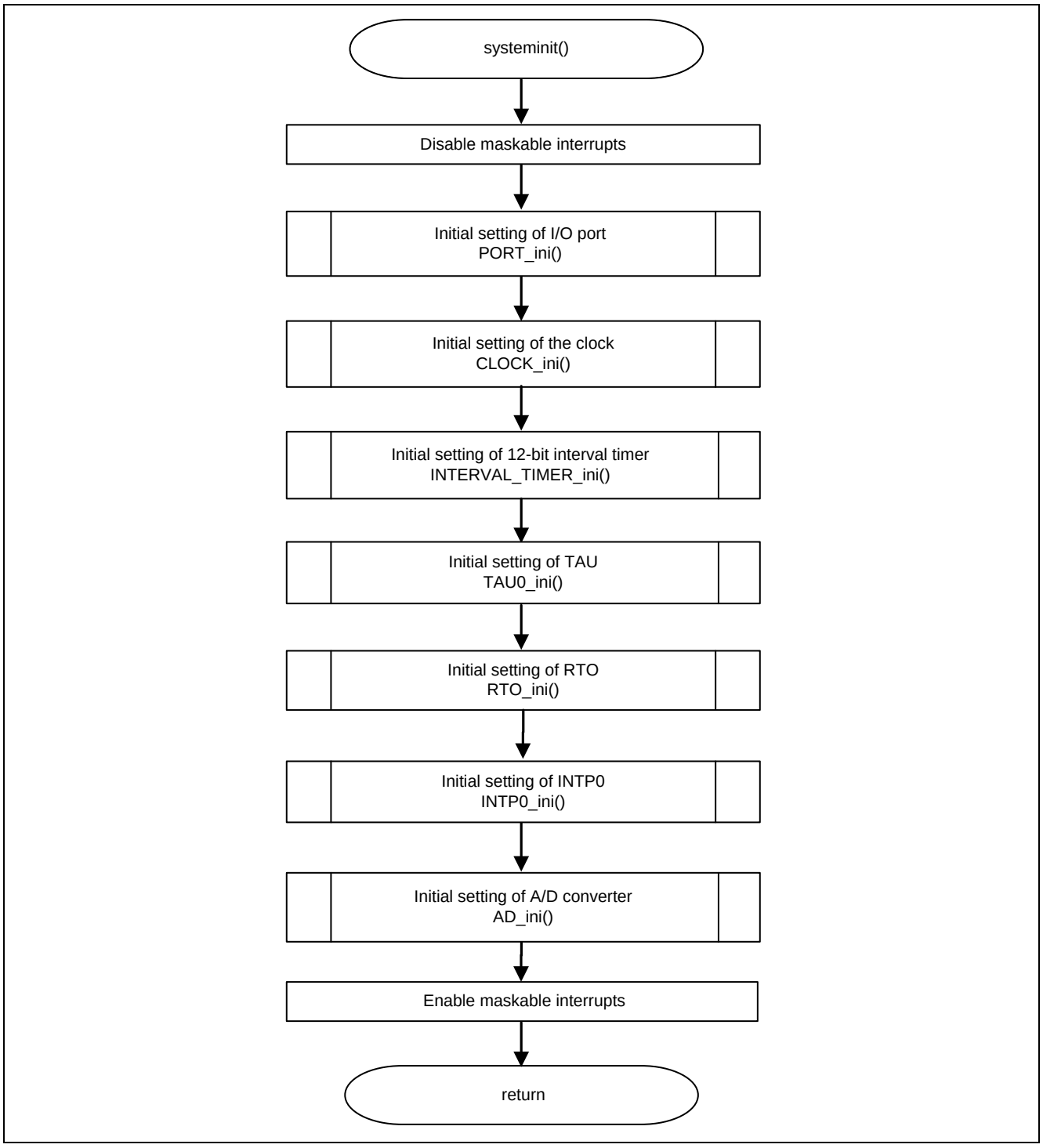


Figure 5.1 Initial Setting of Peripheral Functions

5.7.2 Initial Setting of I/O ports

Figure 5.2 shows the flowchart for initial setting of I/O ports.

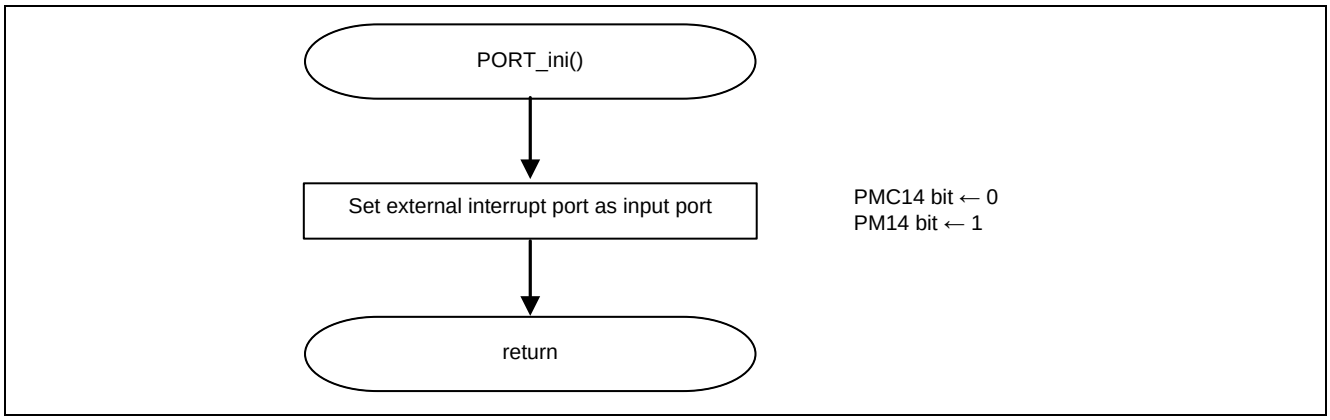


Figure 5.2 Initial Setting of I/O ports

Set the port registers

- Port mode control register 1 (PMC1)
  Set ports as digital I/O ports.

Symbol: PMC1

7	6	5	4	3	2	1	0
1	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10
-		x	**0**				x

Bit 4

PMC14	P14 pin digital I/O / analog input selection
0	**Digital I/O (alternate function other than analog input)**
1	Analog input

- Port mode register 1 (PM1)
  Set P14 as input mode.

Symbol: PM1

7	6	5	4	3	2	1	0
1	PM16	PM15	PM14	PM13	PM12	PM11	PM10
-		x	**1**				x

Bit 4

PM14	P14 pin I/O mode selection
0	Output mode (output buffer on)
1	**Input mode (output buffer off)**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.3 Initial Setting of Clock

Figure 5.3 shows the flowchart for initial setting of clock.

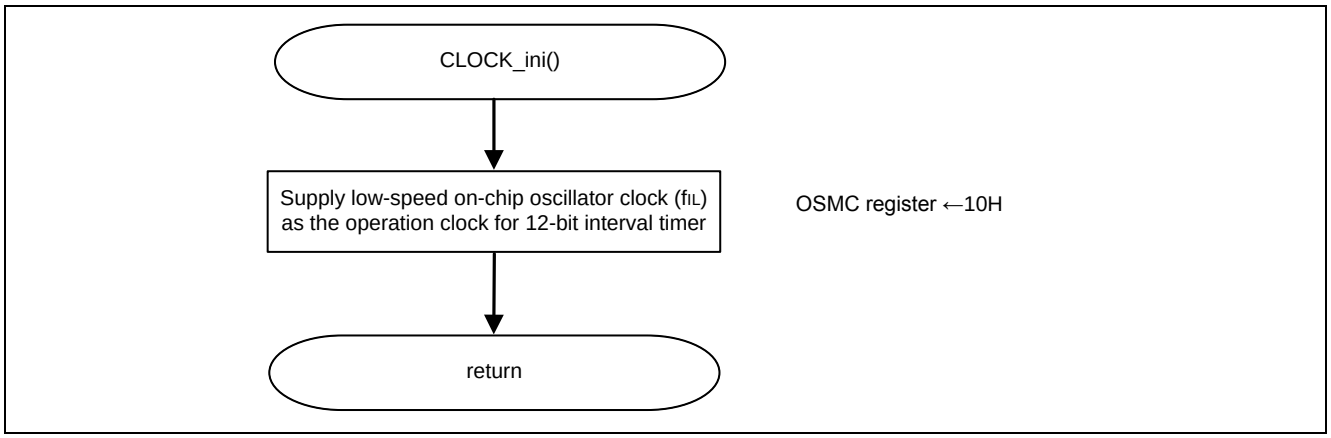


Figure 5.3 Initial Setting of Clock

Set supply of the count clock for 12-bit interval timer

- Operation speed mode control register (OSMC)
  Supply low-speed on-chip oscillator (f_{IL}) for 12-bit interval timer.

Symbol: OSMC

7	6	5	4	3	2	1	0
0	0	0	WUTMMCK0	0	0	0	0
-	-	-	**1**	-	-	-	-

Bit 4

WUTMMCK0	Supply of count clock for 12-bit interval timer
0	Clock supply stop
1	**Low-speed on-chip oscillator clock (f_{IL}) supply**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.4 Initial Setting of 12-bit Interval Timer

Figure 5.4 shows the flowchart for initial setting of 12-bit interval timer.

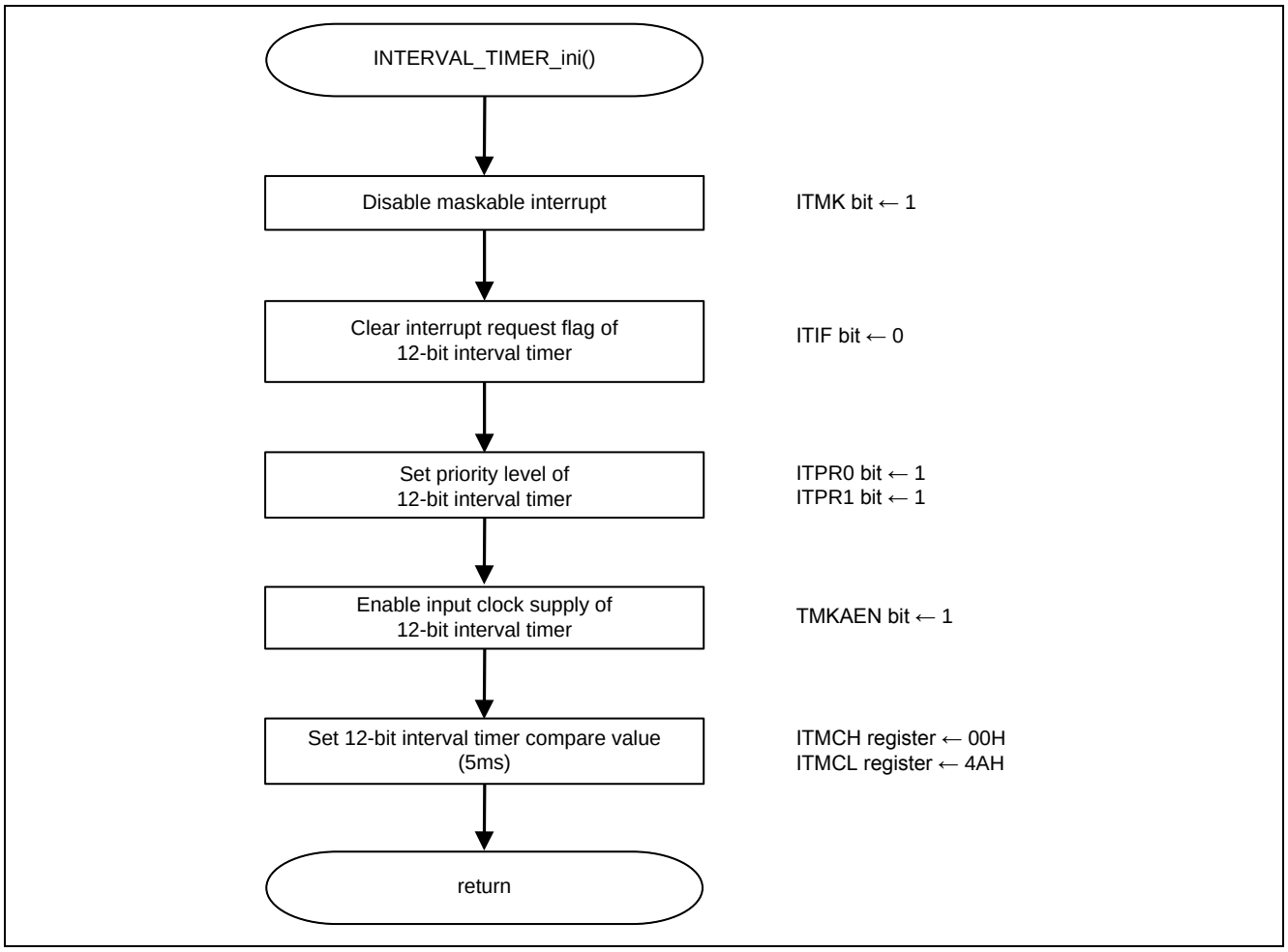


Figure 5.4 Initial Setting of 12-bit Interval Timer

Set interrupt for 12-bit interval timer.

- Interrupt mask flag register (MK1L)
  Disable interrupt servicing.

Symbol: MK1L

7	6	5	4	3	2	1	0
1	1	1	1	PMK5	PMK4	ITMK	TMMK03
-	-	-	-	x	x	**1**	

Bit 1

ITMK	Interrupt servicing control
0	Interrupt servicing enabled
1	**Interrupt servicing disabled**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

• Interrupt request flag register (IF1L)
Clear interrupt request flag.

Symbol: IF1L

7	6	5	4	3	2	1	0
0	0	0	0	PIF5	PIF4	ITIF	TMIF03
-	-	-	-	x	x	**0**	

Bit 1

ITIF	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

• Priority specification flag registers (PR01L, PR11L)
Set priority level.

Symbol: PR01L

7	6	5	4	3	2	1	0
1	1	1	1	PPR05	PPR04	ITPR0	TMPR003
-	-	-	-	x	x	**1**	

Symbol: PR11L

7	6	5	4	3	2	1	0
1	1	1	1	PPR15	PPR14	ITPR1	TMPR103
-	-	-	-	x	x	**1**	

Bit 1

ITPR1	ITPR0	Priority level selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	**1**	**Specifying level 3 (low priority)**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Enable input clock supply for 12-bit interval timer.

- Peripheral enable register 0 (PER0)
  Enable input clock supply.

Symbol: PER0

7	6	5	4	3	2	1	0
TMKAEN	RTOEN	ADCEN	0	0	SAU0EN	0	TAU0EN
1			-	-	x	-	

Bit 7

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stop input clock supply.
1	**Enable input clock supply.**

Set 12-bit interval timer's compare value.

- Interval timer control registers (ITMCH, ITMCL)
  Set 12-bit interval timer's compare value as 5ms.

Symbol: ITMCH

7	6	5	4	3	2	1	0
RINTE	0	0	0	ITCMP11 to ITCMP8			
0	-	-	-	**0**	**0**	**0**	**0**

Symbol: ITMCL

7	6	5	4	3	2	1	0
ITCMP7 to ITCMP0							
0	**1**	**0**	**0**	**1**	**0**	**1**	**0**

Bit 7

RINTE	12-bit interval timer operation control
0	**Count operation stopped (count clear)**
1	Count operation started

ITCMP11 to ITCMP0	12-bit interval timer operation control
001H	These bits generate an interrupt at the fixed cycle (count clock cycles × (ITCMP setting + 1)).
•••	
FFFH	
000H	Setting prohibited
Example interrupt cycles when 04AH is specified for ITCMP11 to ITCMP0 (count clock f_{IL} = 15 kHz) • 1/15 [kHz] × (74 + 1) = 5 [ms]	

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.5 Initial Setting of TAU

Figure 5.5 shows the flowchart for initial setting of TAU.

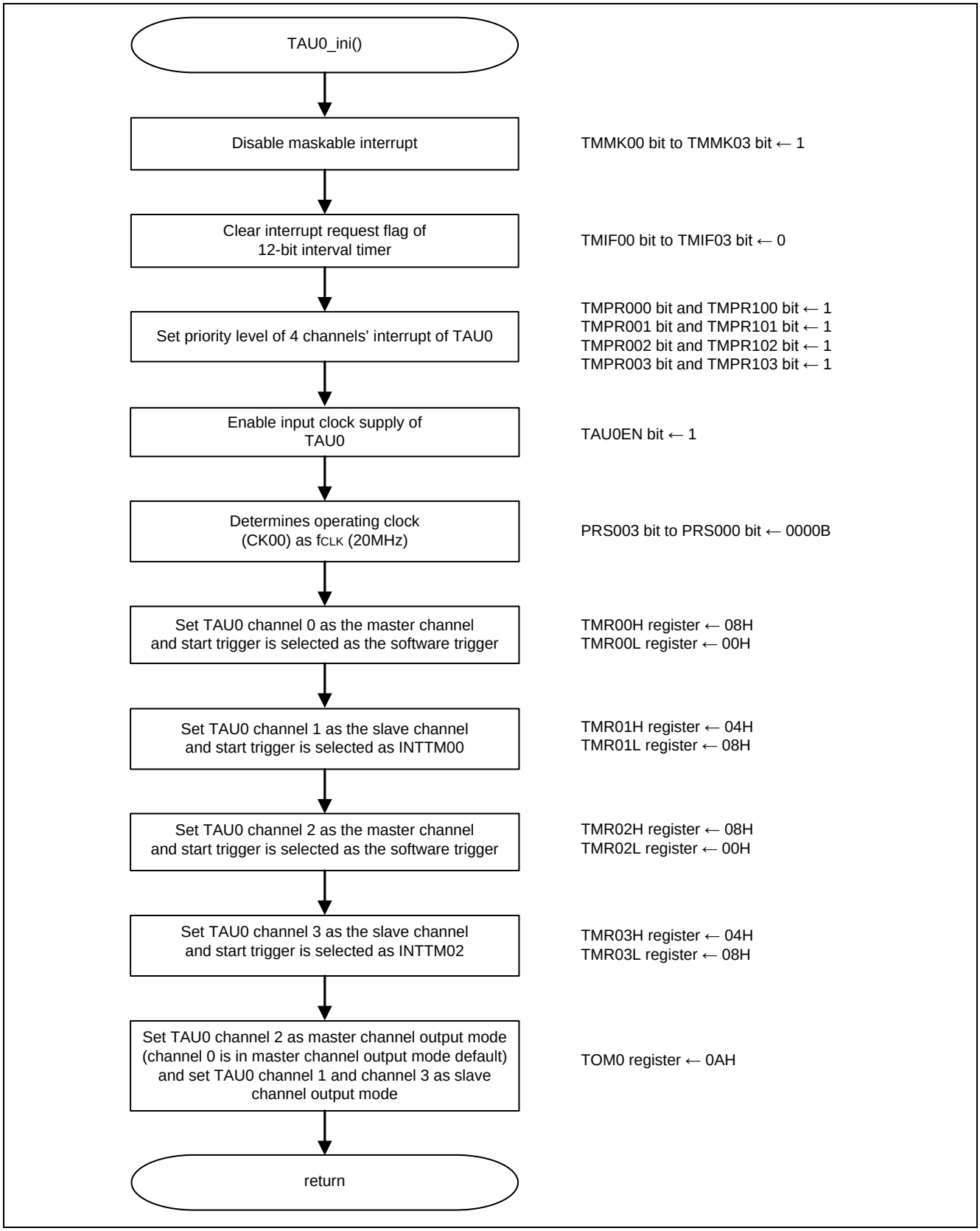


Figure 5.5 Initial Setting of TAU

Set interrupt for TAU0.

- Interrupt mask flag register (MK0L, MK0H and MK1L)
  Disable interrupt servicing.

Symbol: MK0L

7	6	5	4	3	2	1	0
TMMK00	TMMK01H	SREMK0	SRMK0	STMK0 CSIMK00	PMK1	PMK0	WDTIMK
1	x	x	x	x	x		x

Bit 7

TMMK00	Interrupt servicing control
0	Interrupt servicing enabled
1	**Interrupt servicing disabled**

Symbol: MK0H

7	6	5	4	3	2	1	0
TMMK02	1	TMMK03H	PMK3	PMK2	KRMK	ADMK	TMMK01
1	-	x	x	x	x		**1**

Bit 0

TMMK01	Interrupt servicing control
0	Interrupt servicing enabled
1	**Interrupt servicing disabled**

Bit 7

TMMK02	Interrupt servicing control
0	Interrupt servicing enabled
1	**Interrupt servicing disabled**

Symbol: MK1L

7	6	5	4	3	2	1	0
1	1	1	1	PMK5	PMK4	ITMK	TMMK03
-	-	-	-	x	x	x	**1**

Bit 0

TMMK03	Interrupt servicing control
0	Interrupt servicing enabled
1	**Interrupt servicing disabled**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

- Interrupt request flag register (IF0L, IF0H and IF1L)
  Clear interrupt request flag.

Symbol: IF0L

7	6	5	4	3	2	1	0
TMIF00	TMIF01H	SREIF0	SRIF0	STIF0 CSIIF00	PIF1	PIF0	WDTIF
0	x	x	x	x	x		x

Bit 7

TMIF00	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Symbol: IF0H

7	6	5	4	3	2	1	0
TMIF02	0	TMIF03H	PIF3	PIF2	KRIF	ADIF	TMIF01
0	-	x	x	x	x		**0**

Bit 0

TMIF01	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Bit 7

TMIF02	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Symbol: IF1L

7	6	5	4	3	2	1	0
0	0	0	0	PIF5	PIF4	ITIF	TMIF03
-	-	-	-	x	x		**0**

Bit 1

TMIF03	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

- Priority specification flag registers (PR00L, PR10L, PR00H, PR10H, PR01L, and PR11L)
  Set interrupt priority level.

Symbol: PR00L

7	6	5	4	3	2	1	0
TMPR000	TMPR001H	SREPR00	SRPR00	STPR00 CSIPR000	PPR01	PPR00	WDTIPR0
1	x	x	x	x	x		x

Symbol: PR10L

7	6	5	4	3	2	1	0
TMPR100	TMPR101H	SREPR10	SRPR10	STPR10 CSIPR100	PPR11	PPR10	WDTIPR1
1	x	x	x	x	x		x

Bit 7

TMPR100	TMPR000	Priority level selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	**1**	**Specifying level 3 (low priority)**

Symbol: PR00H

7	6	5	4	3	2	1	0
TMPR002	1	TMPR003H	PPR03	PPR02	KRPR0	ADPR0	TMPR001
1	-	x	x	x	x		**1**

Symbol: PR10H

7	6	5	4	3	2	1	0
TMPR102	1	TMPR103H	PPR13	PPR12	KRPR1	ADPR1	TMPR101
1	x	x	x	x	x		**1**

Bit 7

TMPR102	TMPR002	Priority level selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	**1**	**Specifying level 3 (low priority)**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bit 0

TMPR101	TMPR001	Priority level selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	**1**	**Specifying level 3 (low priority)**

Symbol: PR01L

7	6	5	4	3	2	1	0
1	1	1	1	PPR05	PPR04	ITPR0	TMPR003
-	-	-	-	x	x		**1**

Symbol: PR11L

7	6	5	4	3	2	1	0
1	1	1	1	PPR15	PPR14	ITPR1	TMPR103
-	-	-	-	x	x		**1**

Bit 0

TMPR103	TMPR003	Priority level selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	**1**	**Specifying level 3 (low priority)**

Enable input clock supply for TAU0.

- Peripheral enable register 0 (PER0)
  Enable input clock supply.

Symbol: PER0

7	6	5	4	3	2	1	0
TMKAEN	RTOEN	ADCEN	0	0	SAU0EN	0	TAU0EN
			-	-	x	-	**1**

Bit 0

TAU0EN	Control of TAU0 input clock supply
0	Stop input clock supply.
1	**Enable input clock supply.**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Select operation clock for TAU0.

- Timer clock select register 0 (TPS0)
  Select operation clock.

Symbol: TPS0

7	6	5	4	3	2	1	0
PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000
x	x	x	x	**0**	**0**	**0**	**0**

Bits 3 to 0

PRS 003	PRS 002	PRS 001	PRS 000		Selection of operation clock (CK00)				
					f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	**0**	**0**	**0**	**f_{CLK}**	1.25 MHz	2.5 MHz	5 MHz	10 MHz	**20 MHz**
0	0	0	1	$f_{CLK}/2$	625 kHz	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	313 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	156 kHz	313 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	78.1 kHz	156 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	39.1 kHz	78.1 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	0	0	$f_{CLK}/2^{12}$	305 Hz	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	153 Hz	305 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	76.3 Hz	153 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	38.1 Hz	76.3 Hz	153 Hz	305 Hz	610 Hz

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set operation mode for TAU0 channel 0.

- Timer mode register 00 (TMR00H, TMR00L)
  Select operation clock (f_{MCK}).
  Select count clock.
  Set start trigger as software trigger.
  Set operation mode.

Symbol: TMR00H

7	6	5	4	3	2	1	0
CKS001	0	0	CCS00	0	STS002	STS001	STS000
0	-	-	**0**	-	**0**	**0**	**0**

Bit 7

CKS001	Selection of operation clock (f_{MCK}) of channel 0
0	**Operation clock CK00 set by timer clock select register 0 (TPS0)**
1	Operation clock CK01 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS00 bit.	

Bit 4

CCS00	Selection of count clock (f_{TCLK}) of channel 0
0	**Operation clock (f_{MCK}) specified by the CKS001 bit**
1	Valid edge of input signal input from the T00 pin
Count clock (f_{TCLK}) is used for the timer/counter, output controller, and interrupt controller.	

Bits 2 to 0

STS 002	STS 001	STS 000	Setting of start trigger or capture trigger of channel 0
0	**0**	**0**	**Only software trigger start is valid (other trigger sources are unselected).**
0	0	1	Valid edge of the TI00 pin input is used as the start trigger and capture trigger.
0	1	0	Both the edges of the TI00 pin input are used as a start trigger and a capture trigger.
1	0	0	When the channel is used as a slave channel with the one-shot pulse output, PWM output function, or multiple PWM output function: The interrupt request signal of the master channel (INTTM00) is used as the start trigger.
1	1	0	When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM00) is used as the start trigger.
Other than above			Setting prohibited

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Symbol: TMR00L

7	6	5	4	3	2	1	0
CIS001	CIS000	0	0	MD003	MD002	MD001	MD000
0	**0**	-	-	**0**	**0**	**0**	**0**

Bit 7 and bit 6

CIS001	CIS000	Selection of TI00 pin input valid edge
0	**0**	**Falling edge**
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS002 to STS000 bits is other than 010B, set the CIS001 to CIS000 bits to 10B.		

Bits 3 to 1

MD 003	MD 002	MD 001	Setting of operation mode of channel 0	Corresponding function	Count operation of TCR
0	**0**	**0**	**Interval timer mode**	**Interval timer/Square wave output/Divider function/PWM output (master)**	**Down count**
0	1	0	Capture mode	Input pulse interval measurement/Two-channel input with one-shot pulse output function (slave)	Up count
0	1	1	Event counter mode	External event counter	Down count
1	0	0	One-count mode	Delay counter/One-shot pulse output/Two-channel input with one-shot pulse output function (master)/PWM output (slave)	Down count
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of each mode changes depending on the operation of MD000 bit (see the table below).					

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Operation mode (Value set by the MD003 to MD001 bits)	MD 000	Setting of starting counting and interrupt
• **Interval timer mode (0, 0, 0)**	**0**	**Timer interrupt is not generated when counting is started (timer output does not change, either).**
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
	1	Start trigger is valid during counting operation. At that time, a timer interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
Other than above		Setting prohibited

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set operation mode for TAU0 channel 1.

- Timer mode register 01 (TMR01H, TMR01L)
  Select operation clock (f_{MCK}).
  Select count clock.
  Set start trigger as INTTM00 trigger.
  Set operation mode.

Symbol: TMR01H

7	6	5	4	3	2	1	0
CKS011	0	0	CCS01	SPLIT01	STS012	STS011	STS010
0	-	-	**0**	**0**	**1**	**0**	**0**

Bit 7

CKS011	Selection of operation clock (f_{MCK}) of channel 1
0	**Operation clock CK00 set by timer clock select register 0 (TPS0)**
1	Operation clock CK01 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS01 bit.	

Bit 4

CCS01	Selection of count clock (f_{TCLK}) of channel 1
0	**Operation clock (f_{MCK}) specified by the CKS011 bit**
1	Valid edge of input signal input from the T01 pin
Count clock (f_{TCLK}) is used for the timer/counter, output controller, and interrupt controller.	

Bit 3

SPLIT01	Selection of 8 or 16-bit timer operation for channels 1
0	**Operates as 16-bit timer.**
1	Operates as 8-bit timer.

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bits 2 to 0

STS012	STS011	STS010	Setting of start trigger or capture trigger of channel 1
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TI01 pin input is used as the start trigger and capture trigger.
0	1	0	Both the edges of the TI01 pin input are used as a start trigger and a capture trigger.
1	**0**	**0**	**When the channel is used as a slave channel with the one-shot pulse output, PWM output function, or multiple PWM output function:** **The interrupt request signal of the master channel (INTTM00) is used as the start trigger.**
1	1	0	When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM00) is used as the start trigger.
Other than above			Setting prohibited

Symbol: TMR01L

7	6	5	4	3	2	1	0
CIS011	CIS010	0	0	MD013	MD012	MD011	MD010
0	**0**	-	-	**1**	**0**	**0**	**0**

Bit 7 and bit 6

CIS011	CIS010	Selection of TI01 pin input valid edge
0	**0**	**Falling edge**
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS012 to STS010 bits is other than 010B, set the CIS011 to CIS010 bits to 10B.		

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bits 3 to 1

MD 013	MD 012	MD 011	Setting of operation mode of channel 1	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer/Square wave output/Divider function/PWM output (master)	Down count
0	1	0	Capture mode	Input pulse interval measurement/Two-channel input with one-shot pulse output function (slave)	Up count
0	1	1	Event counter mode	External event counter	Down count
1	**0**	**0**	**One-count mode**	**Delay counter/One-shot pulse output/Two-channel input with one-shot pulse output function (master)/PWM output (slave)**	**Down count**
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of each mode changes depending on the operation of MD010 bit (see the table below).					

Operation mode (Value set by the MD013 to MD011 bits)	MD 010	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• **One-count mode (1, 0, 0)**	**0**	**Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.**
	1	Start trigger is valid during counting operation. At that time, a timer interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
Other than above		Setting prohibited

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set PWM period.

- Timer data register 00 (TDR00H, TDR00L)
- Timer data register 01 (TDR01H, TDR01L)

  Select operation clock.

Symbol: TDR00H

7	6	5	4	3	2	1	0

Symbol: TDR00L

7	6	5	4	3	2	1	0

Pulse period = {Set value of TDR00 (master) + 1} × Count clock period

Symbol: TDR01H

7	6	5	4	3	2	1	0

Symbol: TDR01L

7	6	5	4	3	2	1	0

Duty factor [%] = {Set value of TDR01 (slave)}/{Set value of TDR00 (master) + 1} × 100

Note: For the stepper motor rotation speed calculation, please refer to ‘4.3.2 Rotation speed calculation’.

Refer to the R7F0C806-809 user’s manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set operation mode for TAU0 channel 2.

- Timer mode register 02 (TMR02H, TMR02L)
  Select operation clock (f_{MCK}).
  Select count clock.
  Set start trigger as software trigger.
  Set operation mode.

Symbol: TMR02H

7	6	5	4	3	2	1	0
CKS021	0	0	CCS02	MASTER02	STS022	STS021	STS020
0	-	-	**0**	**1**	**0**	**0**	**0**

Bit 7

CKS021	Selection of operation clock (f_{MCK}) of channel 2
0	**Operation clock CK00 set by timer clock select register 0 (TPS0)**
1	Operation clock CK01 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS02 bit.	

Bit 4

CCS02	Selection of count clock (f_{TCLK}) of channel 2
0	**Operation clock (f_{MCK}) specified by the CKS021 bit**
1	Valid edge of input signal input from the T02 pin
Count clock (f_{TCLK}) is used for the timer/counter, output controller, and interrupt controller.	

Bit 3

MASTER02	Selection of independent channel operation/simultaneous channel operation (slave/master) of channel 2
0	Operates as the slave channel in the independent channel operation function or the simultaneous channel operation function.
1	**Operates as the master channel in the simultaneous channel operation function.**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bits 2 to 0

STS022	STS021	STS020	Setting of start trigger or capture trigger of channel 2
0	**0**	**0**	**Only software trigger start is valid (other trigger sources are unselected).**
0	0	1	Valid edge of the TI02 pin input is used as the start trigger and capture trigger.
0	1	0	Both the edges of the TI02 pin input are used as a start trigger and a capture trigger.
1	0	0	When the channel is used as a slave channel with the one-shot pulse output, PWM output function, or multiple PWM output function: The interrupt request signal of the master channel (INTTM02) is used as the start trigger.
1	1	0	When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM02) is used as the start trigger.
Other than above			Setting prohibited

Symbol: TMR02L

7	6	5	4	3	2	1	0
CIS021	CIS020	0	0	MD023	MD022	MD021	MD020
0	0	-	-	0	0	0	0

Bit 7 and bit 6

CIS021	CIS020	Selection of TI02 pin input valid edge
0	**0**	**Falling edge**
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS022 to STS020 bits is other than 010B, set the CIS021 to CIS020 bits to 10B.		

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bits 3 to 1

MD 023	MD 022	MD 021	Setting of operation mode of channel 0	Corresponding function	Count operation of TCR
0	**0**	**0**	**Interval timer mode**	**Interval timer/Square wave output/Divider function/PWM output (master)**	**Down count**
0	1	0	Capture mode	Input pulse interval measurement/Two-channel input with one-shot pulse output function (slave)	Up count
0	1	1	Event counter mode	External event counter	Down count
1	0	0	One-count mode	Delay counter/One-shot pulse output/Two-channel input with one-shot pulse output function (master)/PWM output (slave)	Down count
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of each mode changes depending on the operation of MD020 bit (see the table below).					

Operation mode (Value set by the MD023 to MD021 bits)	MD 020	Setting of starting counting and interrupt
• **Interval timer mode (0, 0, 0)**	**0**	**Timer interrupt is not generated when counting is started (timer output does not change, either).**
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
	1	Start trigger is valid during counting operation. At that time, a timer interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
Other than above		Setting prohibited

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set operation mode for TAU0 channel 3.

- Timer mode register 03 (TMR03H, TMR03L)
  Select operation clock (f_{MCK}).
  Select count clock.
  Set start trigger as INTTM02 trigger.
  Set operation mode.

Symbol: TMR03H

7	6	5	4	3	2	1	0
CKS031	0	0	CCS03	SPLIT03	STS032	STS031	STS030
0	-	-	**0**	**0**	**1**	**0**	**0**

Bit 7

CKS031	Selection of operation clock (f_{MCK}) of channel 3
0	**Operation clock CK00 set by timer clock select register 0 (TPS0)**
1	Operation clock CK01 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS01 bit.	

Bit 4

CCS03	Selection of count clock (f_{TCLK}) of channel 3
0	**Operation clock (f_{MCK}) specified by the CKS031 bit**
1	Valid edge of input signal input from the T013 pin
Count clock (f_{TCLK}) is used for the timer/counter, output controller, and interrupt controller.	

Bit 3

SPLIT03	Selection of 8 or 16-bit timer operation for channels 3
0	**Operates as 16-bit timer.**
1	Operates as 8-bit timer.

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bits 2 to 0

STS032	STS031	STS030	Setting of start trigger or capture trigger of channel 3
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TI00 pin input is used as the start trigger and capture trigger.
0	1	0	Both the edges of the TI0n pin input are used as a start trigger and a capture trigger.
1	**0**	**0**	**When the channel is used as a slave channel with the one-shot pulse output, PWM output function, or multiple PWM output function:** **The interrupt request signal of the master channel (INTTM02) is used as the start trigger.**
1	1	0	When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM02) is used as the start trigger.
Other than above			Setting prohibited

Symbol: TMR03L

7	6	5	4	3	2	1	0
CIS031	CIS030	0	0	MD033	MD032	MD031	MD030
0	**0**	-	-	**1**	**0**	**0**	**0**

Bit 7 and bit 6

CIS031	CIS030	Selection of TI03 pin input valid edge
0	**0**	**Falling edge**
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS032 to STS030 bits is other than 010B, set the CIS031 to CIS030 bits to 10B.		

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bits 3 to 1

MD 033	MD 032	MD 031	Setting of operation mode of channel 3	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer/Square wave output/Divider function/PWM output (master)	Down count
0	1	0	Capture mode	Input pulse interval measurement/Two-channel input with one-shot pulse output function (slave)	Up count
0	1	1	Event counter mode	External event counter	Down count
1	**0**	**0**	**One-count mode**	**Delay counter/One-shot pulse output/Two-channel input with one-shot pulse output function (master)/PWM output (slave)**	**Down count**
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of each mode changes depending on the operation of MD030 bit (see the table below).					

Operation mode (Value set by the MD033 to MD031 bits)	MD 030	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• **One-count mode (1, 0, 0)**	**0**	**Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.**
	1	Start trigger is valid during counting operation. At that time, a timer interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
Other than above		Setting prohibited

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set PWM period.

- Timer data register 02 (TDR02H, TDR02L)
- Timer data register 03 (TDR03H, TDR03L)

  Select operation clock.

Symbol: TDR02H

7	6	5	4	3	2	1	0

Symbol: TDR02L

7	6	5	4	3	2	1	0

Pulse period = {Set value of TDR02 (master) + 1} × Count clock period

Symbol: TDR03H

7	6	5	4	3	2	1	0

Symbol: TDR03L

7	6	5	4	3	2	1	0

Duty factor [%] = {Set value of TDR03 (slave)}/{Set value of TDR02 (master) + 1} × 100

Note: For the stepper motor rotation speed calculation, please refer to ‘4.3.2 Rotation speed calculation’.

Refer to the R7F0C806-809 user’s manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set output mode for each channel of TAU0.

- Timer output mode register 0 (TOM0)
  Select output mode for each channel of TAU0.

Symbol: TOM0

7	6	5	4	3	2	1	0
0	0	0	0	TOM03	TOM02	TOM01	0
-	-	-	-	**1**	**0**	**1**	-

Bit 3 and bit 1

TOM0n	Control of timer output mode of channel n
0	Used as the independent channel operation function (to produce toggle output by the interrupt request signal (INTTM0n))
1	**Slave channel output mode (output is set by the interrupt request signal (INTTM00, INTTM02) of the master channel, and reset by the timer interrupt request signal (INTTM0n) of the slave channel)**

n=1 or 3

Bit 2

TOM02	Control of timer output mode of channel 2
0	**Used as the independent channel operation function (to produce toggle output by the interrupt request signal (INTTM02))**
1	Slave channel output mode (output is set by the interrupt request signal (INTTM00, INTTM02) of the master channel, and reset by the timer interrupt request signal (INTTM01 or INTTM03) of the slave channel)

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.6 Initial Setting of RTO

Figure 5.6 shows the flowchart for initial setting of RTO.

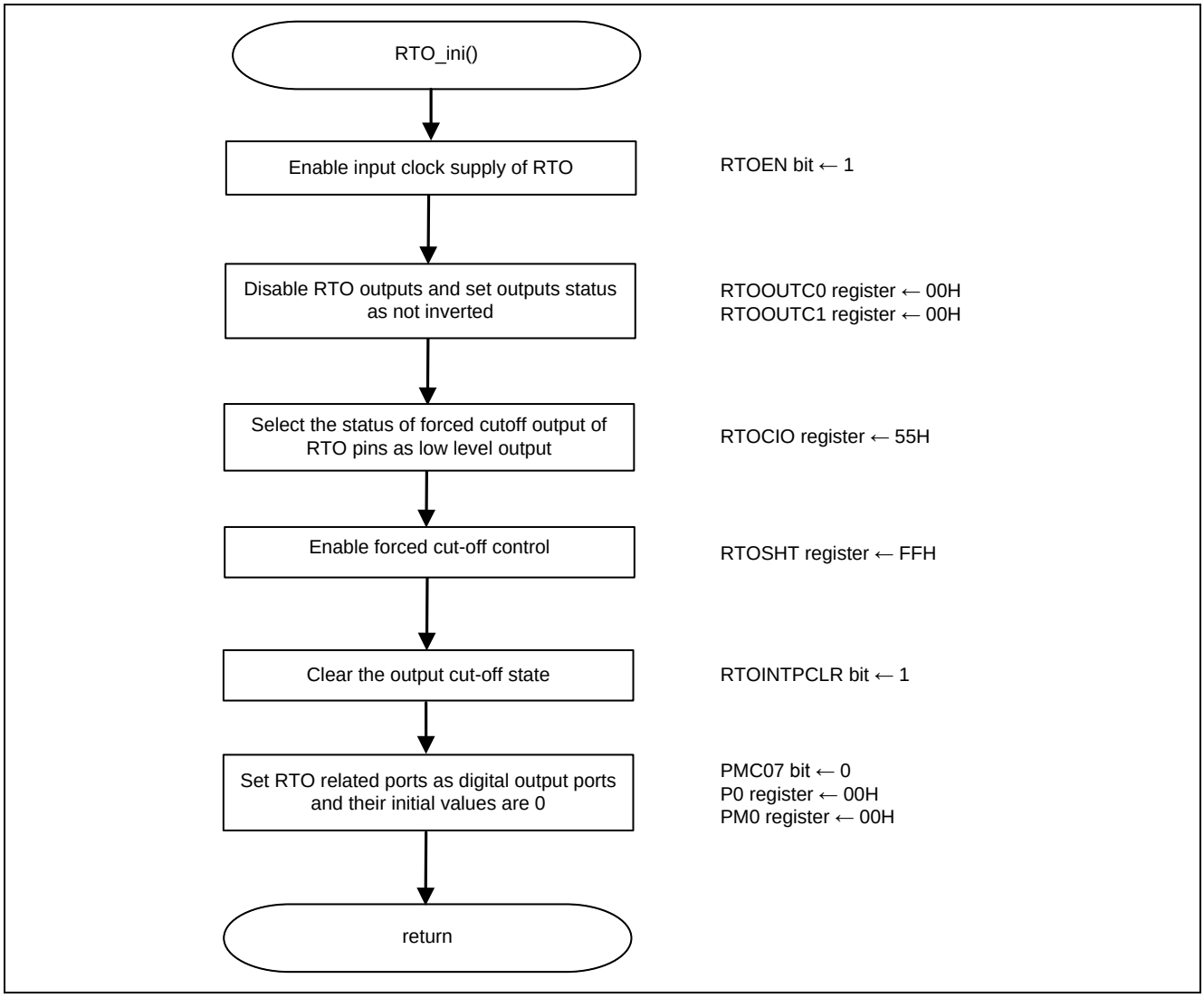


Figure 5.6 Initial Setting of RTO

Enable input clock supply for RTO

- Peripheral enable register 0 (PER0)
  Enable input clock supply.

Symbol: PER0

7	6	5	4	3	2	1	0
TMKAEN	RTOEN	ADCEN	0	0	SAU0EN	0	TAU0EN
	1		-	-	x	-	

Bit 6

RTOEN	Control of RTO input clock supply
0	Stop input clock supply.
1	**Enable input clock supply.**

Set RTO output inverting control

- RTO control register 0 and 1 (RTOOUTC0 and RTOOUTC1)
  Set the output inverting control of RTIO00 to RTIO07 pins.

Symbol: RTOOUTC0

7	6	5	4	3	2	1	0
RTOACT3	RTOACT2	RTOACT1	RTOACT0	RTOSEL3	RTOSEL2	RTOSEL1	RTOSEL0
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Bits 3 to 0

RTOSELn	RTIO0n output control
0	**Disable output.**
1	Enable output.

n = 0 to 3

Bits 7 to 4

RTOACTn	RTIO0n output inverting control
0	**Do not invert**
1	Invert

n = 0 to 3

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Symbol: RTOOUTC1

7	6	5	4	3	2	1	0
RTOACT7	RTOACT6	RTOACT5	RTOACT4	RTOSEL7	RTOSEL6	RTOSEL5	RTOSEL4
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Bits 3 to 0

RTOSELn	RTIO0n output control
0	**Disable output.**
1	Enable output.

n = 4 to 7

Bits 7 to 4

RTOACTn	RTIO0n output inverting control
0	**Do not invert**
1	Invert

n = 4 to 7

Set RTO output inverting control

- RTO forced cut-off output selection register (RTOCIO)
  Select the status of the RTO output that has forcibly been cut off.

Symbol: RTOCIO

7	6	5	4	3	2	1	0
RTOCIO7	RTOCIO6	RTOCIO5	RTOCIO4	RTOCIO3	RTOCIO2	RTOCIO1	RTOCIO0
0	**1**	**0**	**1**	**0**	**1**	**0**	**1**

Bit 7 and bit 6

RTOCIO7	RTOCIO6	Selection of the status of forced cut-off output from RTIO07
0	0	Hi-Z output
0	**1**	**Low-level output**
1	0	High-level output
1	1	Cut-off invalidated

Bit 5 and bit 4

RTOCIO5	RTOCIO4	Selection of the status of forced cut-off output from RTIO05 to RTIO03
0	0	Hi-Z output
0	**1**	**Low-level output**
1	0	High-level output
1	1	Cut-off invalidated

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Bit 3 and bit 2

RTOCIO3	RTOCIO2	Selection of the status of forced cut-off output from RTIO06
0	0	Hi-Z output
0	**1**	**Low-level output**
1	0	High-level output
1	1	Cut-off invalidated

Bit 1 and bit 0

RTOCIO1	RTOCIO0	Selection of the status of forced cut-off output from RTIO02 to RTIO00
0	0	Hi-Z output
0	**1**	**Low-level output**
1	0	High-level output
1	1	Cut-off invalidated

Enable RTO forced cut-off control

- RTO forced cut-off control register (RTOSHT)
  Enable forced cut-off.

Symbol: RTOSHT

7	6	5	4	3	2	1	0
RTOSHT7	RTOSHT6	RTOSHT5	RTOSHT4	RTOSHT3	RTOSHT2	RTOSHT1	RTOSHT0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Bits 7 to 0

RTOSHTn	RTIO0n output forced cut-off control
0	Disable forced cut-off.
1	**Enable forced cut-off.**

n=0 to 7

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Clear the forced cut-off status

- RTO forced cut-off status register (RTOSTR)
  Clear the forced cut-off state.

Symbol: RTOSTR

7	6	5	4	3	2	1	0
0	0	0	0	0	0	RTOSHTFLG	RTOINTPCLR
-	-	-	-	-	-	**0**	**1**

Bit 1

RTOSHTFLG	Cut-off status flag
0	**The timer output is output normally.**
1	The timer output is cut off.

Bit 0

RTOINTPCLR	Clearing the output cut-off state
0	**--**
1	**Clear the output cut-off state.**

Set the port registers

- Port mode control register 0 (PMC0)
  Set ports as digital I/O ports.

Symbol: PMC0

7	6	5	4	3	2	1	0
PMC07	1	1	1	1	1	1	1
0	-	-	-	-	-	-	-

Bit 7

PMC07	P07 pin digital I/O/analog input selection
0	**Digital I/O (alternate function other than analog input)**
1	Analog input

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

• Port mode register 0 (PM0)
Set input or output mode for the ports.

Symbol: PM0

7	6	5	4	3	2	1	0
PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Bit 7 to 0

PM0n	P0n pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

n=0 to 7

• Port register 0 (P0)
Set port output latch.

Symbol: P0

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Bit 7 to 0

P0n	Output data control (in output mode)	Input data read (in input mode)
0	**Output 0**	Input low level
1	Output 1	Input high level

n=0 to 7

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.7 Initial Setting of INTP0

Figure 5.7 shows the flowchart for initial setting of INTP0.

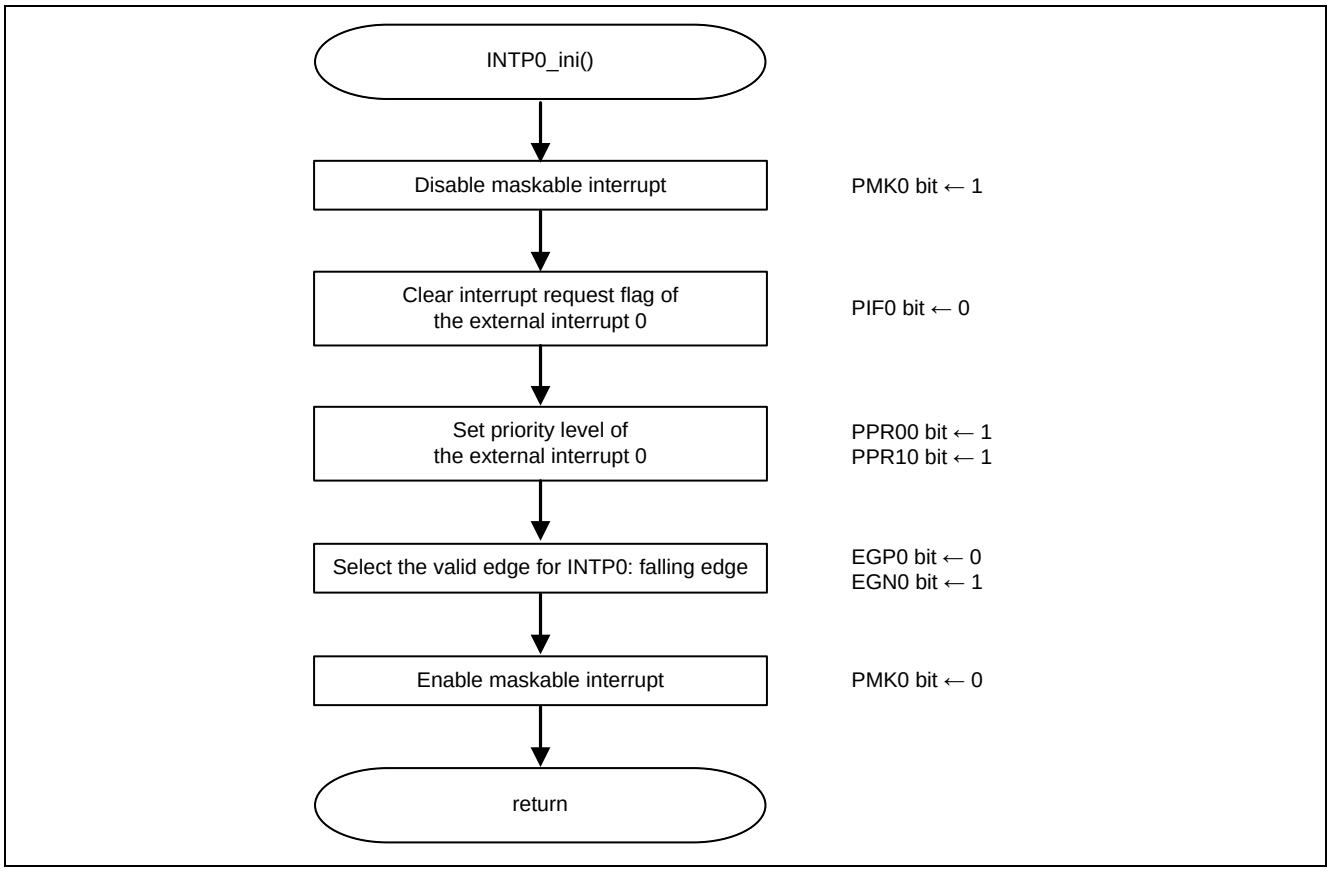


Figure 5.7 Initial Setting of INTP0

Set interrupt for external interrupt 0 (INTP0)

- Interrupt mask flag register (MK0L)
  Disable or enable interrupt servicing.

Symbol: MK0L

7	6	5	4	3	2	1	0
TMMK00	TMMK01H	SREMK0	SRMK0	STMK0 CSIMK00	PMK1	PMK0	WDTIMK
	x	x	x	x	x	**1/0**	x

Bit 1

PMK0	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

• Interrupt request flag register (IF0L)
Clear interrupt request flag.

Symbol: IF0L

7	6	5	4	3	2	1	0
TMIF00	TMIF01H	SREIF0	SRIF0	STIF0 CSIIF00	PIF1	PIF0	WDTIF
	x	x	x	x	x	**0**	x

Bit 1

PIF0	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

• Priority specification flag registers (PR00L, PR10L)
Set priority level.

Symbol: PR00L

7	6	5	4	3	2	1	0
TMPR000	TMPR001H	SREPR00	SRPR00	STPR00 CSIPR000	PPR01	PPR00	WDTIPR0
	x	x	x	x	x	**1**	x

Symbol: PR10L

7	6	5	4	3	2	1	0
TMPR100	TMPR101H	SREPR10	SRPR10	STPR10 CSIPR100	PPR11	PPR10	WDTIPR1
	x	x	x	x	x	**1**	x

Bit 1

PPR10	PPR00	Priority level selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	**1**	**Specifying level 3 (low priority)**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set INTP0 pin valid edge

- External interrupt rising edge enable register 0 (EGP0) and External interrupt falling edge enable register 0 (EGN0)
  Select falling edge as the valid edge for INTP0.

Symbol: EGP0

7	6	5	4	3	2	1	0
0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0
-	-	x	x	x	x	x	**0**

Symbol: EGN0

7	6	5	4	3	2	1	0
0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0
-	-	x	x	x	x	x	**1**

Bit 0

EGP0	EGN0	INTP0 pin valid edge selection
0	0	Edge detection disabled.
0	**1**	**Falling edge.**
1	0	Rising edge.
1	1	Both rising and falling edges.

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.8 Initial Setting of A/D Converter

Figure 5.8 shows the flowchart for initial setting of A/D converter.

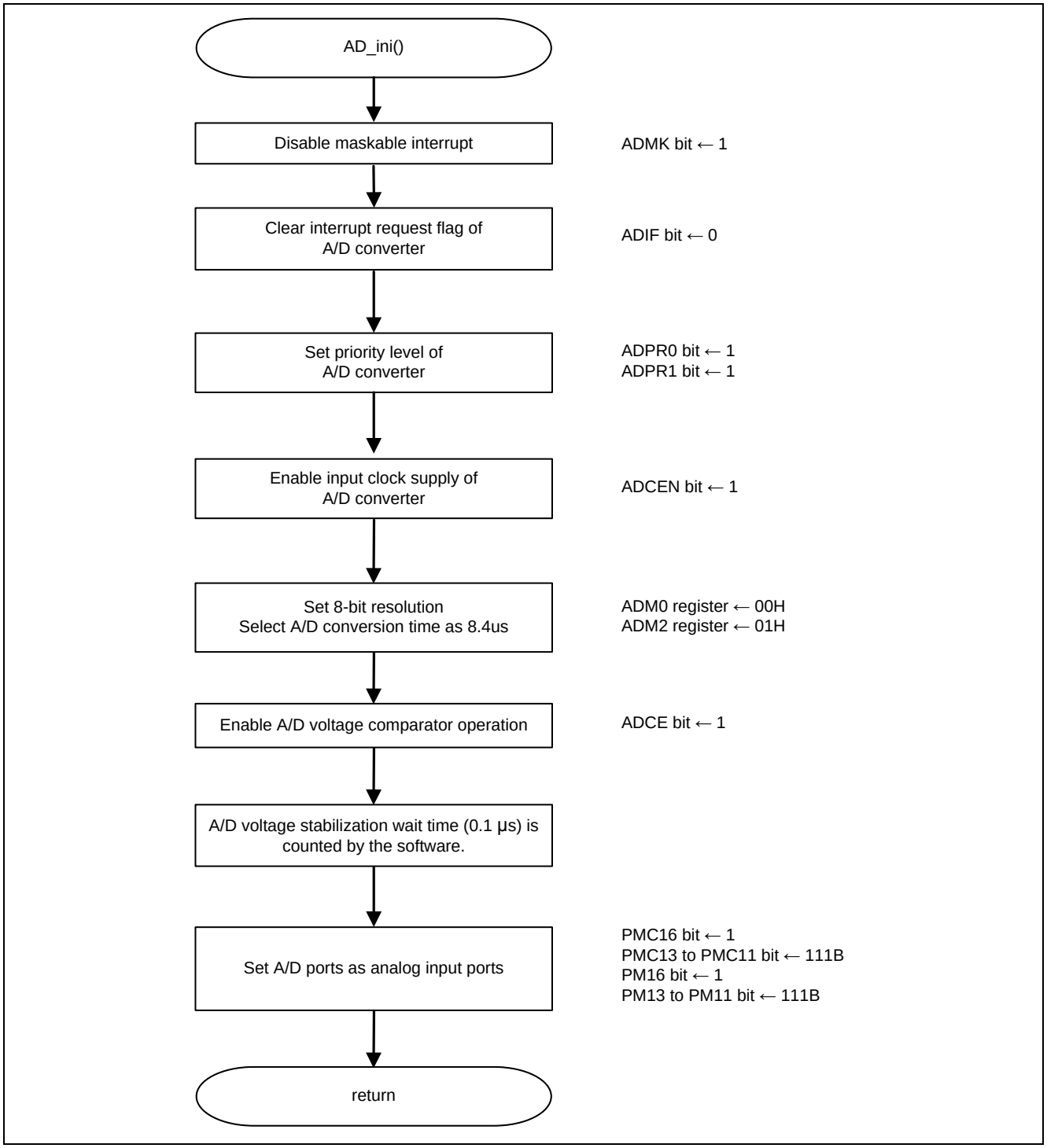


Figure 5.8 Initial Setting of A/D Converter

Set interrupt for A/D converter (INTAD)

- Interrupt mask flag register (MK0H)
  Disable interrupt servicing.

Symbol: MK0H

7	6	5	4	3	2	1	0
TMMK02	1	TMMK03H	PMK3	PMK2	KRMK	ADMK	TMMK01
	-	x	x	x	x	**1**	

Bit 1

ADMK	Interrupt servicing control
0	Interrupt servicing enabled
1	**Interrupt servicing disabled**

- Interrupt request flag register (IF0H)
  Clear interrupt request flag.

Symbol: IF0H

7	6	5	4	3	2	1	0
TMIF02	0	TMIF03H	PIF3	PIF2	KRIF	ADIF	TMIF01
	-	x	x	x	x	**0**	

Bit 1

ADIF	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

- Priority specification flag registers (PR00H, PR10H)
  Set priority level.

Symbol: PR00H

7	6	5	4	3	2	1	0
TMPR002	1	TMPR003H	PPR03	PPR02	KRPR0	ADPR0	TMPR001
	-	x	x	x	x	**1**	

Symbol: PR10H

7	6	5	4	3	2	1	0
TMPR102	1	TMPR103H	PPR13	PPR12	KRPR1	ADPR1	TMPR101
	-	x	x	x	x	**1**	

Bit 1

ADPR10	ADPR00	Priority level selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	**1**	**Specifying level 3 (low priority)**

Enable input clock supply for A/D converter

- Peripheral enable register 0 (PER0)
  Enable input clock supply.

Symbol: PER0

7	6	5	4	3	2	1	0
TMKAEN	RTOEN	ADCEN	0	0	SAU0EN	0	TAU0EN
		1	-	-	x	-	

Bit 5

ADCEN	Control of A/D converter input clock supply
0	Stop input clock supply.
1	**Enable input clock supply.**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

Set A/D conversion operation mode

- A/D converter mode register 0 (ADM0)
  Set A/D conversion operation mode.

Symbol: ADM0

7	6	5	4	3	2	1	0
ADCS	0	0	FR1	FR0	0	LV0	ADCE
0	-	-	**0**	**0**	-	**0**	**0**

Bit 7

ADCS	A/D conversion operation control
0	**Stop conversion operation (conversion stopped/standby status)**
1	Enable conversion operation (conversion operation status)

Bit 0

ADCE	A/D voltage comparator operation control
0	**Stop A/D voltage comparator operation**
1	Enable A/D voltage comparator operation

Bit 4 to 3, and bit 1

A/D Converter Mode Register 0 (ADM0)			Conversion Clock	Number of Conversion Clock	Conversion Time	Conversion Time Selection (μs)				
FR1	FR0	LV0				f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	**0**		**f_{CLK}/8**	**21 f_{AD} (Number of sampling clock: 9 f_{AD})**	**168/f_{CLK}**	Setting prohibited	Setting prohibited	Setting prohibited	16.8	**8.4**
0	1	**0**	f_{CLK}/4		84/f_{CLK}			16.8	8.4	4.2
1	0		f_{CLK}/2		42/f_{CLK}		16.8	8.4	4.2	Setting prohibited
1	1		f_{CLK}		21/f_{CLK}	16.8	8.4	4.2	Setting prohibited	
0	0		f_{CLK}/8	15 f_{AD} (Number of sampling clock: 3 f_{AD})	120/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	12.0	6.0
0	1	1	f_{CLK}/4		60/f_{CLK}			12.0	6.0	3.0
1	0		f_{CLK}/2		30/f_{CLK}		12.0	6.0	3.0	Setting prohibited
1	1		f_{CLK}		15/f_{CLK}	12.0	6.0	3.0	Setting prohibited	

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

• A/D converter mode register 2 (ADM2)
Set A/D conversio resolution.

Symbol: ADM2

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ADTYP
-	-	-	-	-	-	-	**1**

Bit 0

ADTYP	Resolution of A/D conversion
0	10-bit resolution
1	**8-bit resolution**

Set the port registers

• Port mode control register 1 (PMC1)
Set ports as analog input ports.

Symbol: PMC1

7	6	5	4	3	2	1	0
1	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10
-	**1**	x		**1**	**1**	**1**	x

Bit 6 and bits 3 to 1

PMC1n	P1n pin digital I/O / analog input selection
0	Digital I/O (alternate function other than analog input)
1	**Analog input**

n=1 to 3, 6

• Port mode register 1 (PM1)
Set ports as input mode.

Symbol: PM1

7	6	5	4	3	2	1	0
1	PM16	PM15	PM14	PM13	PM12	PM11	PM10
-	**1**	x		**1**	**1**	**1**	x

Bit 6 and bits 3 to 1

PM1n	P1n pin I/O mode selection
0	Output mode (output buffer on)
1	**Input mode (output buffer off)**

n=1 to 3, 6

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.9 Main Processing

Figure 5.9 and figure 5.10 show the flowchart for main processing.

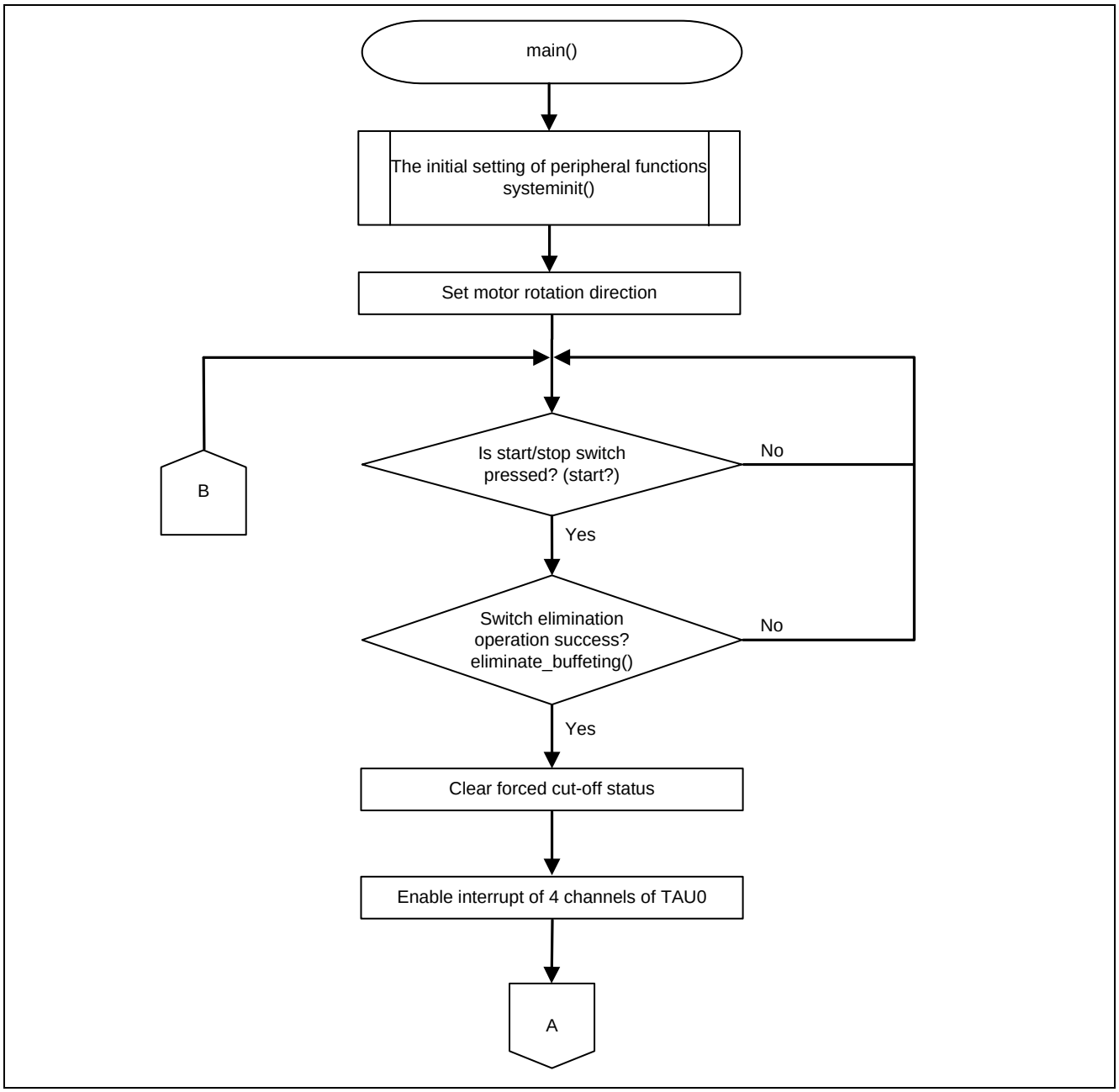


Figure 5.9 Main Processing (1/2)

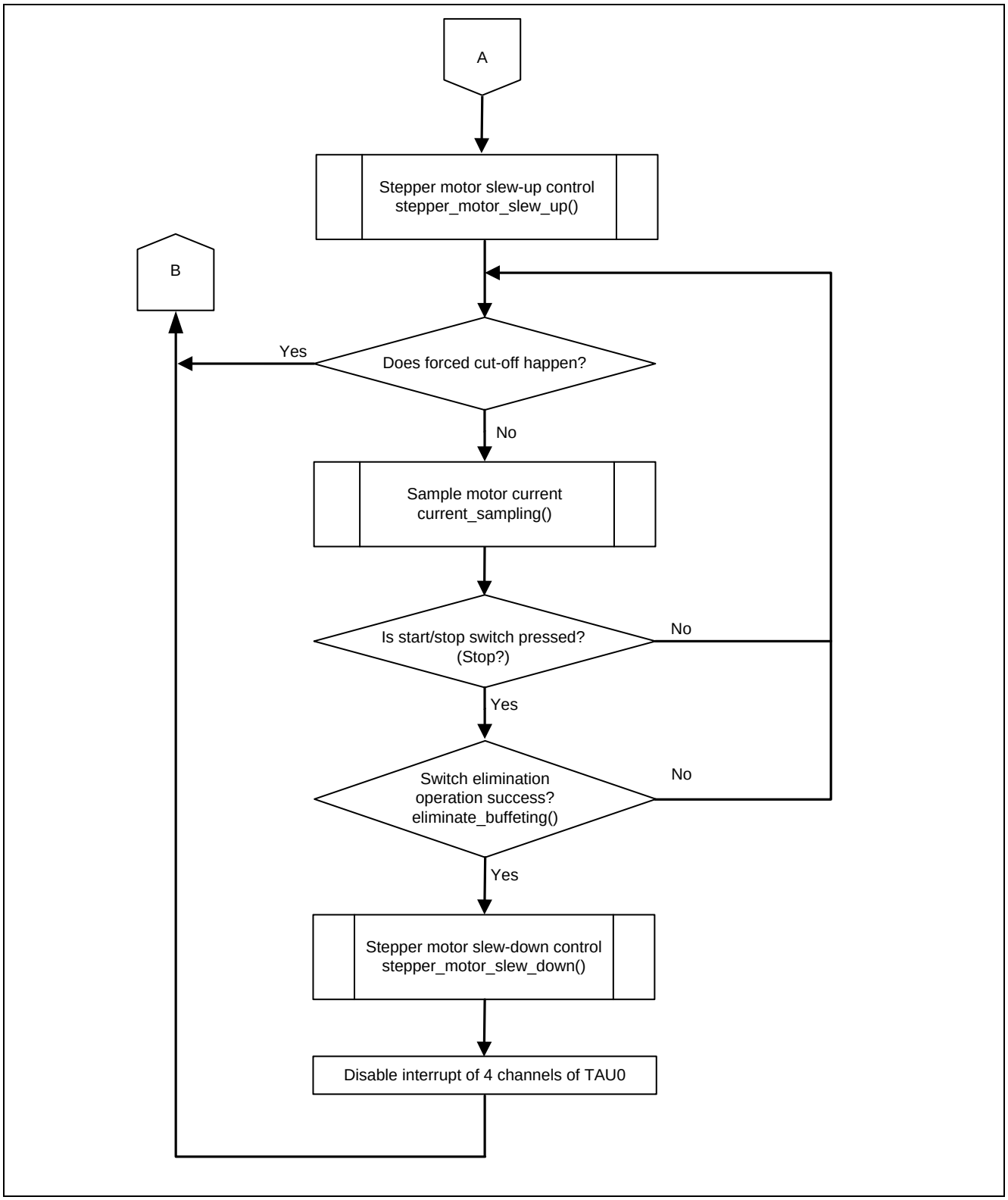


Figure 5.10 Main Processing (2/2)

5.7.10 Switch Elimination Function

Figure 5.11 shows the flowchart for switch elimination function.

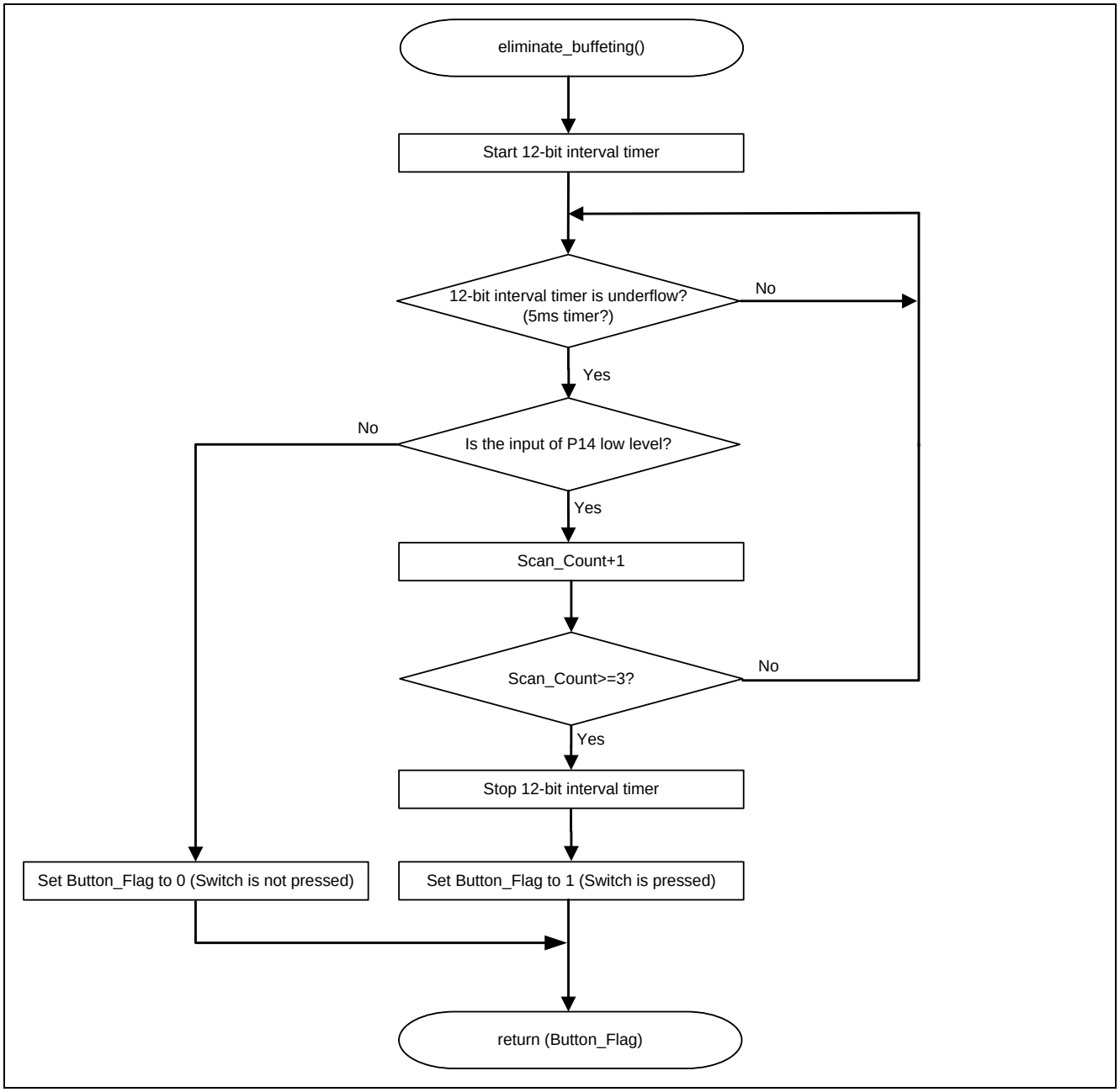


Figure 5.11 Switch Elimination Function

5.7.11 Stepper Motor Slew-up Function

Figure 5.12 and figure 5.13 show the flowchart for stepper motor slew-up function.

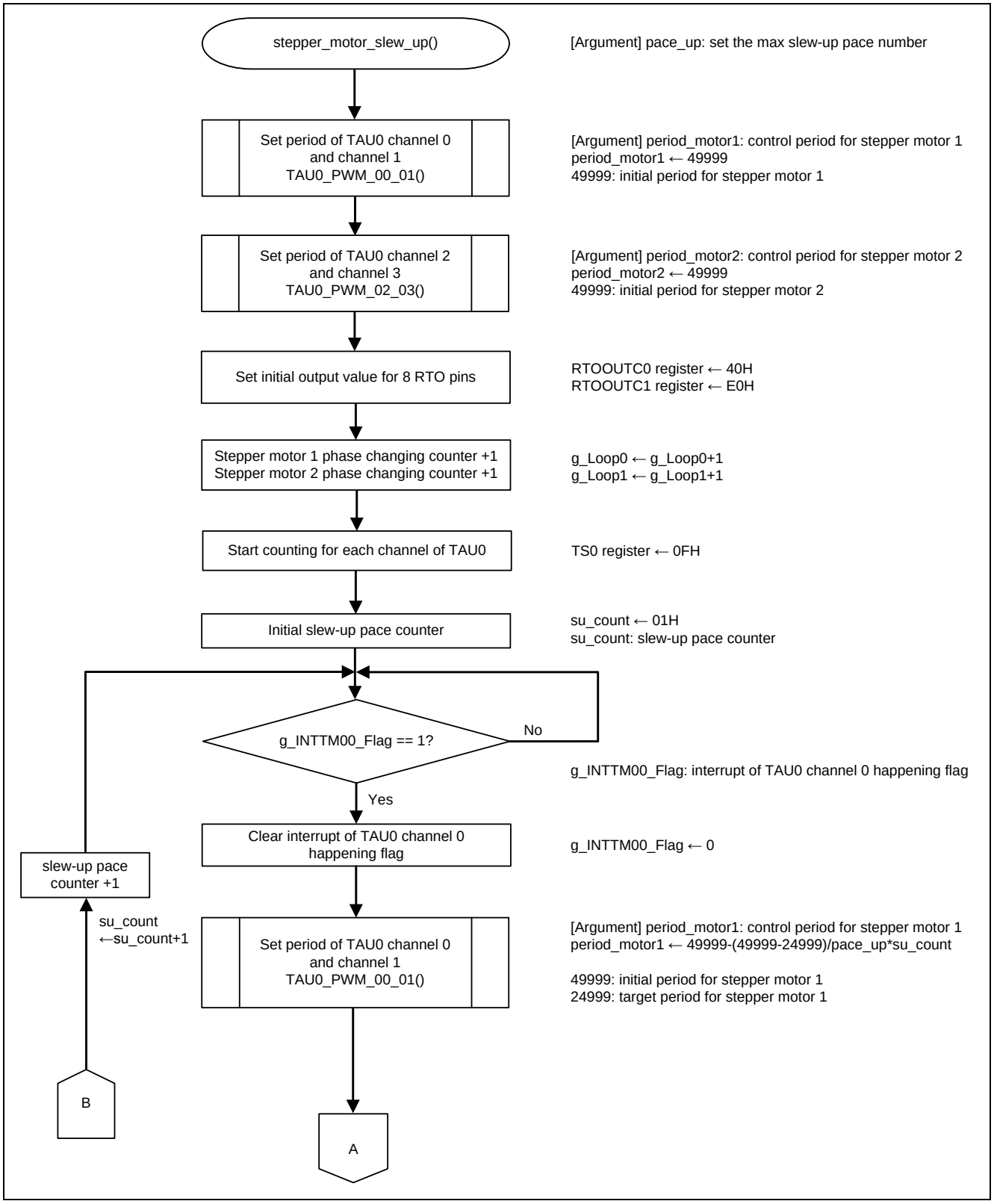


Figure 5.12 Stepper Motor Slew-up Function (1/2)

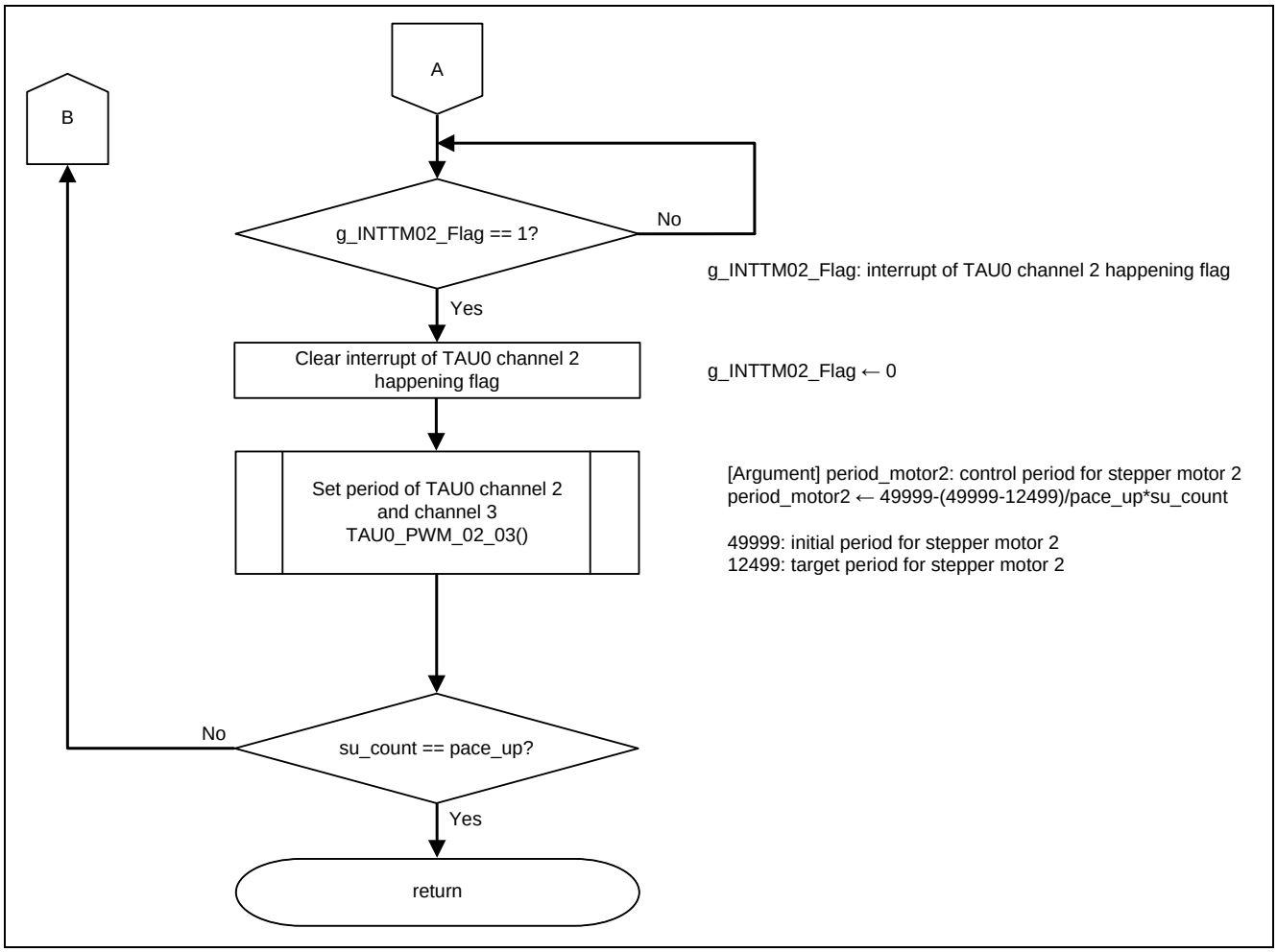


Figure 5.13 Stepper Motor Slew-up Function (2/2)

5.7.12 Stepper Motor Slew-down Function

Figure 5.14 shows the flowchart for stepper motor slew-down function.

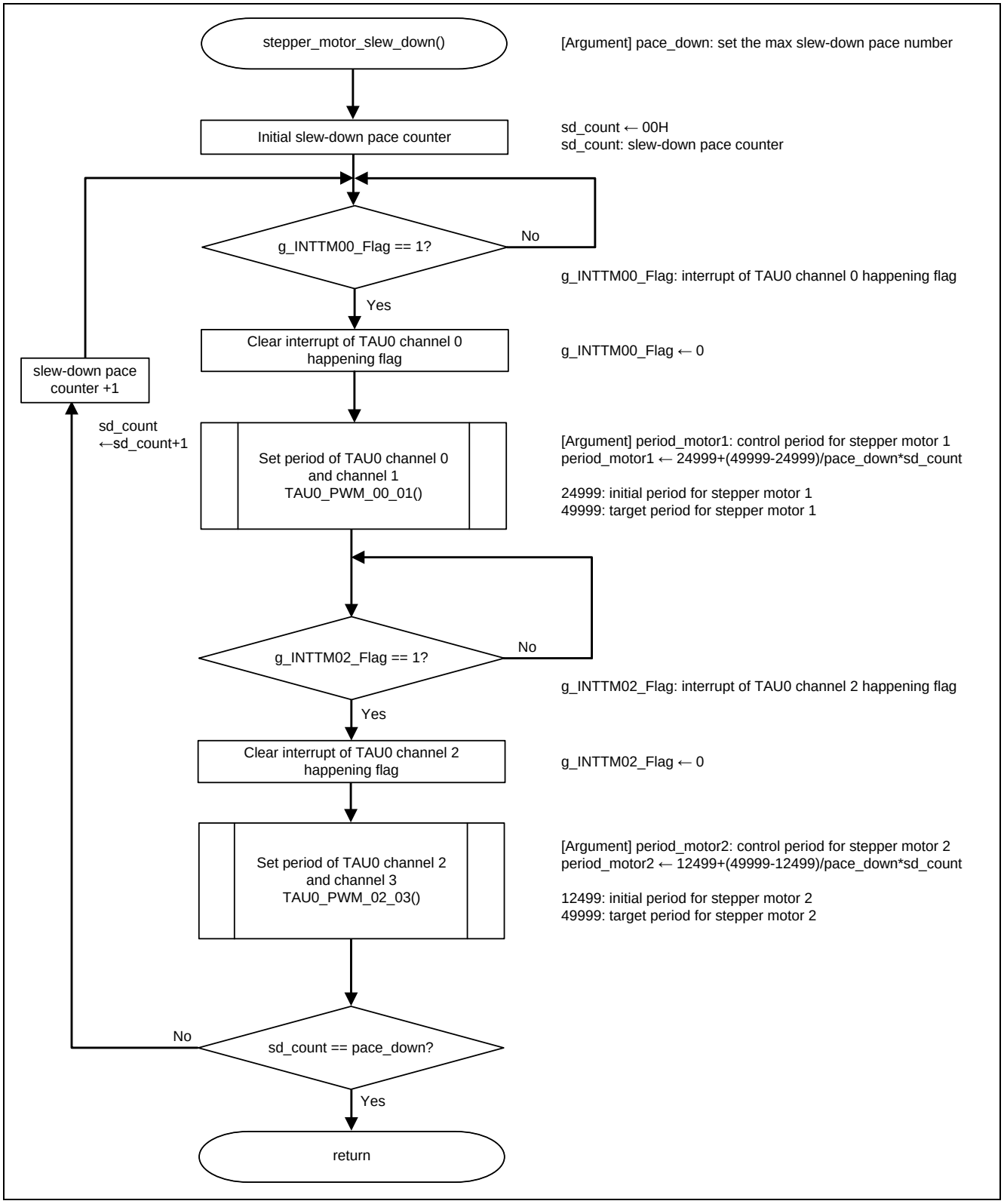


Figure 5.14 Stepper Motor Slew-down Function

5.7.13 TAU0 PWM Period Setting

Figure 5.15 shows the flowchart for TAU0 channel 0 and 1 PWM period setting.

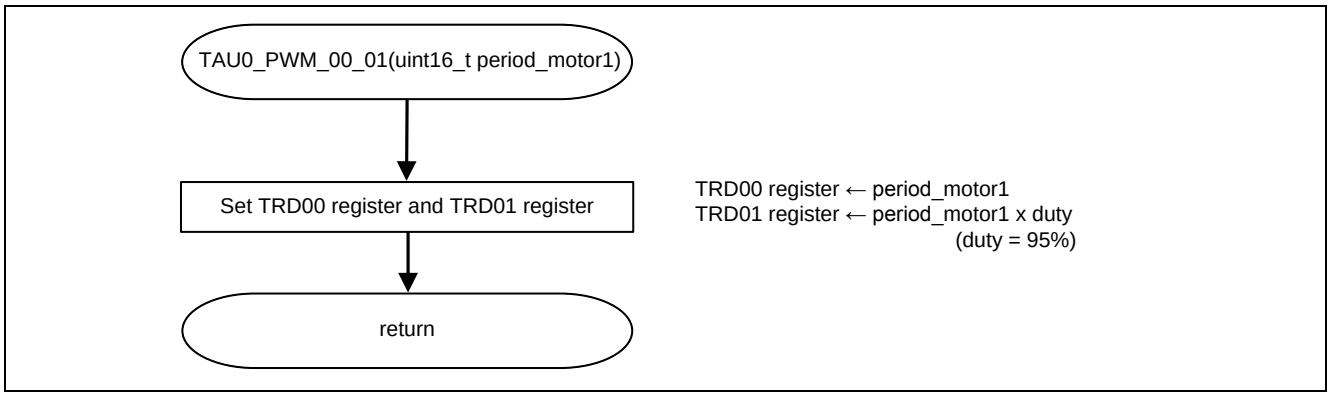


Figure 5.15 TAU0 channel 0 and 1 PWM Period Setting

Figure 5.16 shows the flowchart for TAU0 channel 2 and 3 PWM period setting.

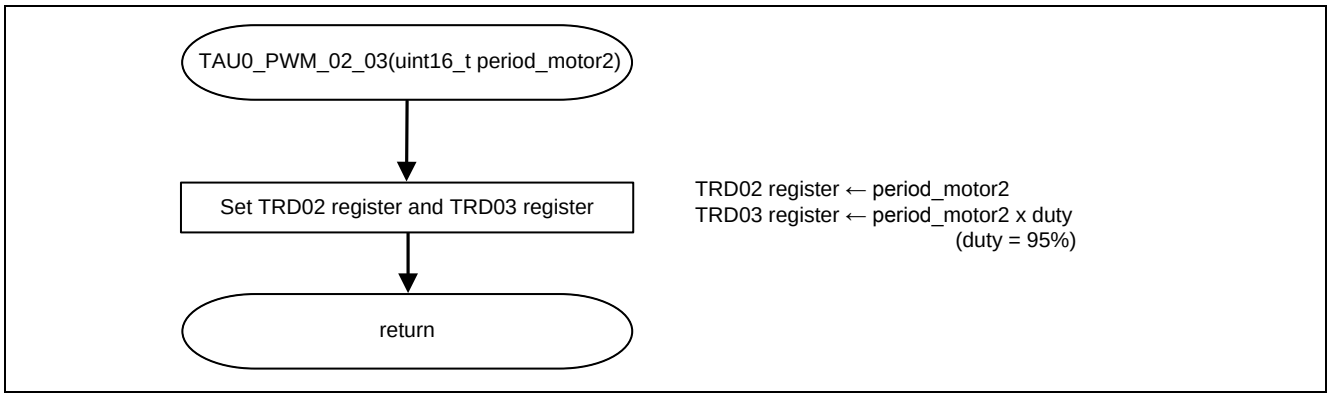


Figure 5.16 TAU0 channel 2 and 3 PWM Period Setting

5.7.14 TAU0 Interrupt Sub Routine

Figure 5.17 to figure 5.20 show the flowchart for TAU0 interrupt sub routine.

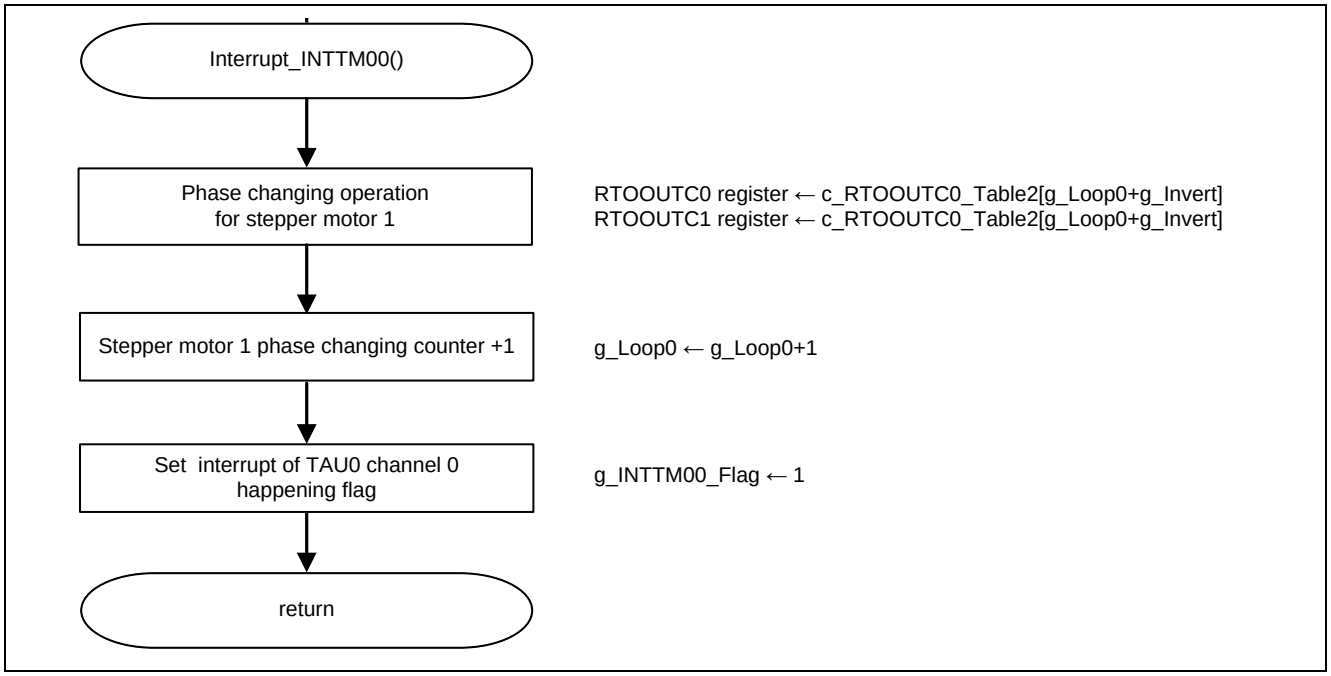


Figure 5.17 TAU0 channel 0 Interrupt Sub Routine

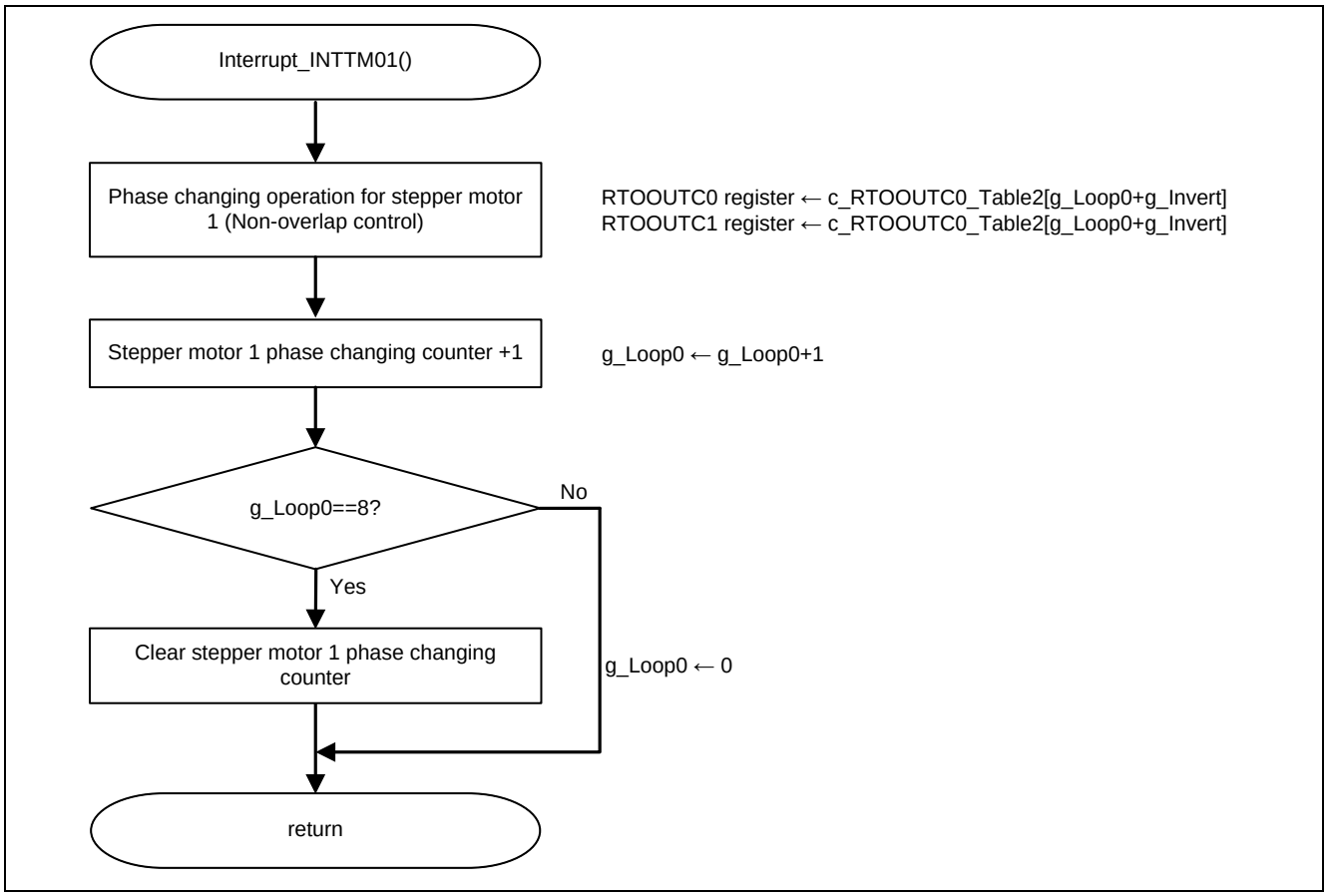


Figure 5.18 TAU0 channel 1 Interrupt Sub Routine

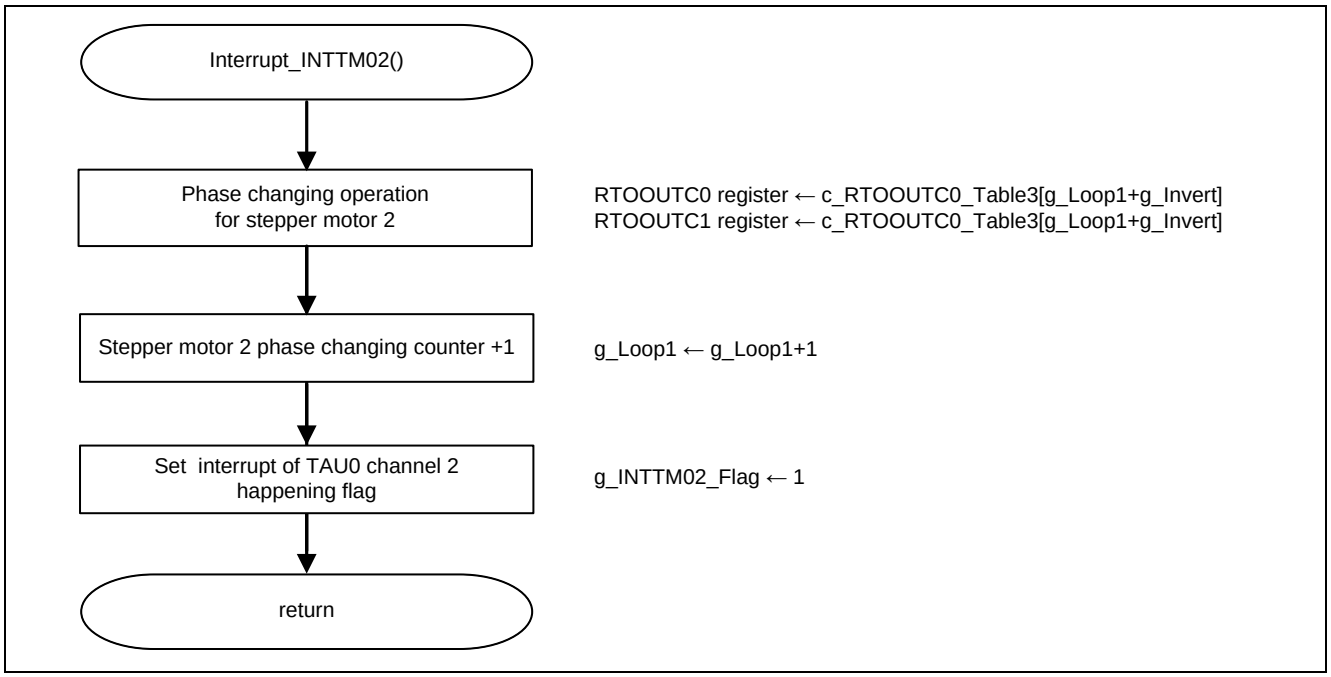


Figure 5.19 TAU0 channel 2 Interrupt Sub Routine

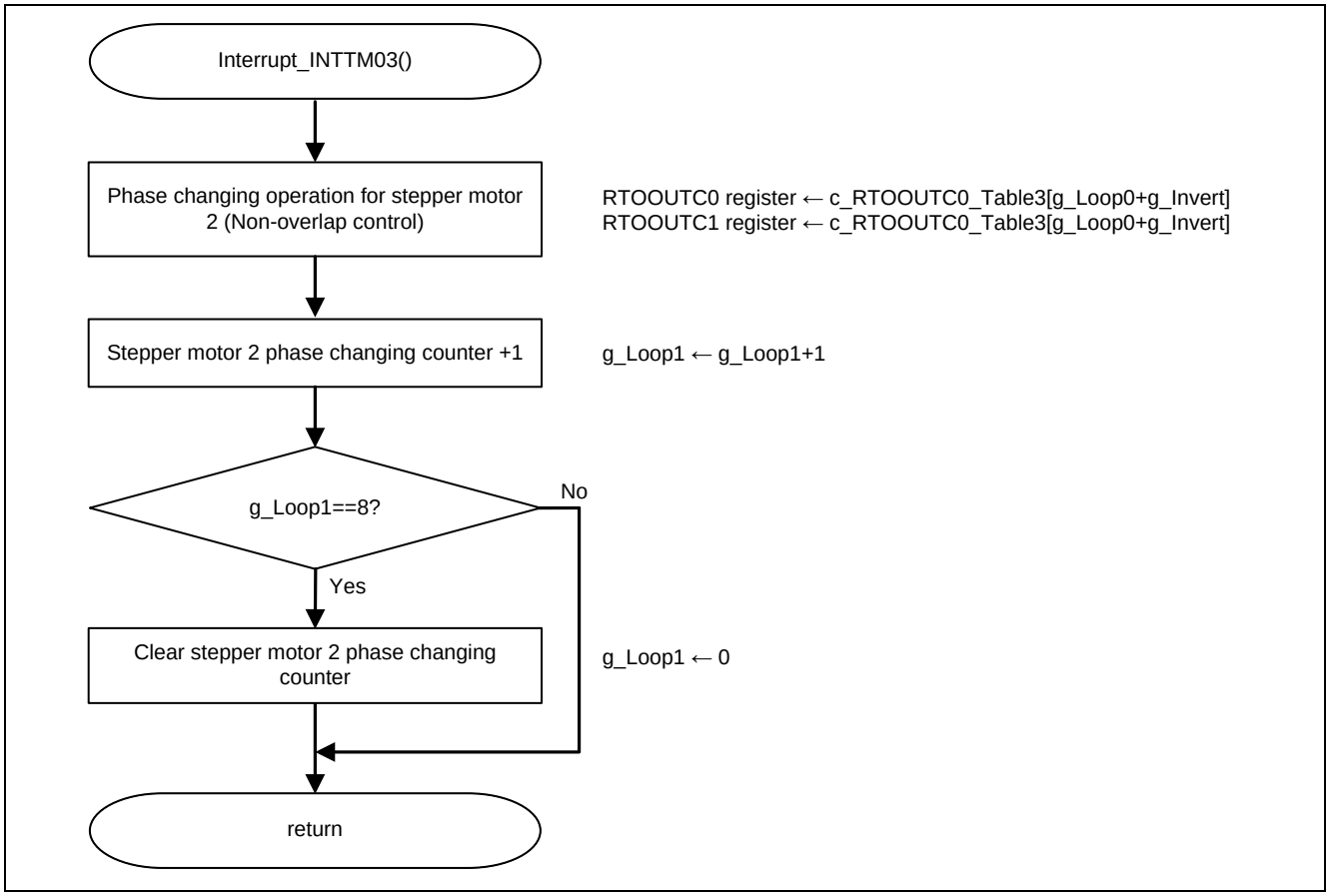


Figure 5.20 TAU0 channel 3 Interrupt Sub Routine

5.7.15 Motor Current Sampling

Figure 5.21 shows the flowchart for motor current sampling.

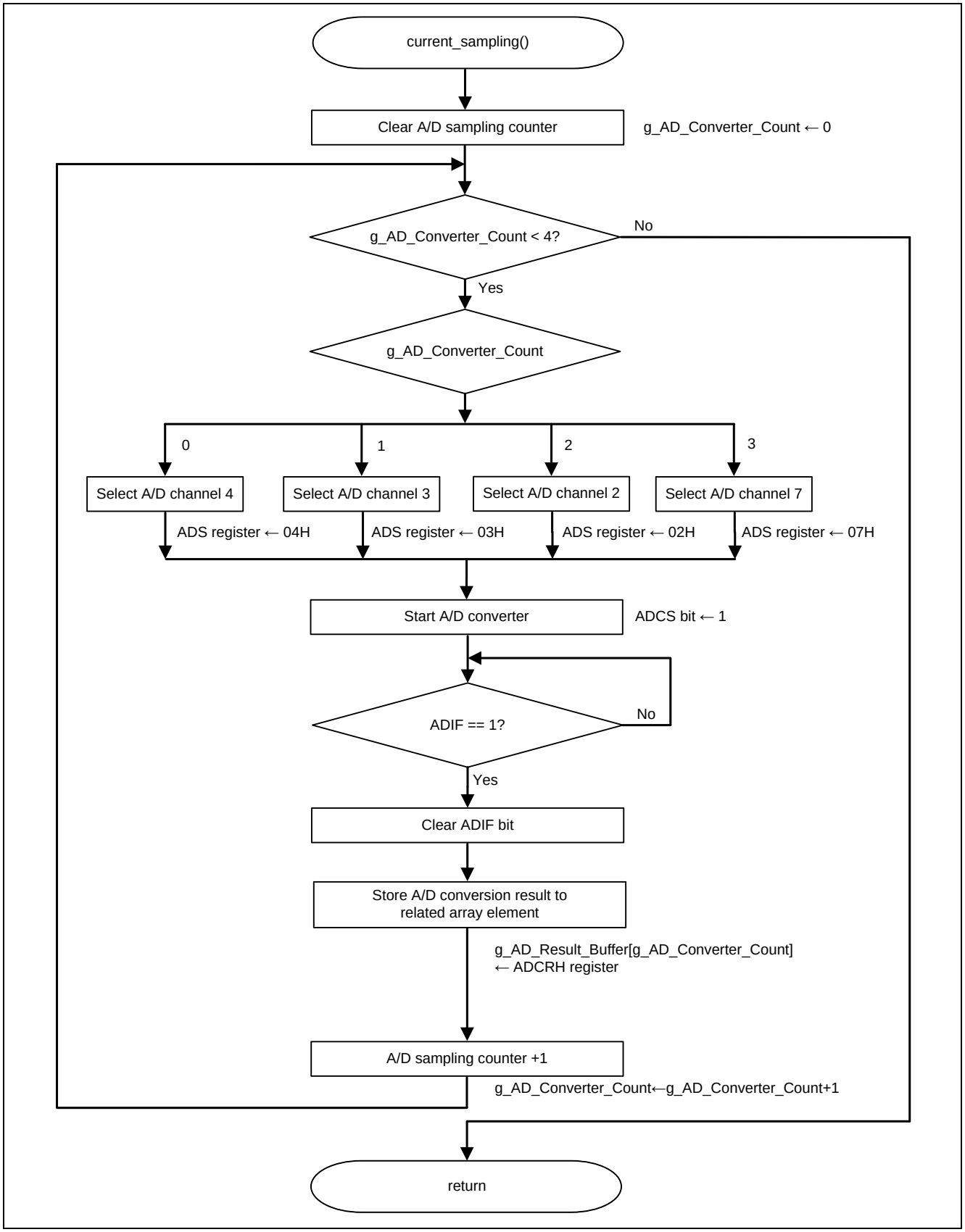


Figure 5.21 Motor Current Sampling

Set A/D conversion channel

- Analog input channel specification register (ADS)
  Specify the input channel of the analog voltage to be A/D converted.

Symbol: ADS

7	6	5	4	3	2	1	0
0	0	0	0	0	ADS2	ADS1	ADS0
-	-	-	-	-			

Bits 2 to 0

ADS2	ADS1	ADS0	Target of A/D conversion	Analog input pin
0	0	0	AN0	P07/ANI0 pin
0	0	1	ANI1	P10/ANI1 pin
0	**1**	**0**	**ANI2**	**P11/ANI2 pin**
0	**1**	**1**	**ANI3**	**P12/ANI3 pin**
1	**0**	**0**	**ANI4**	**P13/ANI4 pin**
1	0	1	ANI5	P14/ANI5 pin
1	1	0	ANI6	P15/ANI6 pin
1	**1**	**1**	**ANI7**	**P16/ANI7 pin**

Start A/D conversion

- A/D converter mode register 0 (ADM0)
  Start A/D conversion.

Symbol: ADM0

7	6	5	4	3	2	1	0
ADCS	0	0	FR1	FR0	0	LV0	ADCE
1	-	-			-		

Bit 7

ADCS	A/D voltage comparator operation control
0	Conversion standby state
1	**Conversion-in-progress state**

Refer to the R7F0C806-809 user's manual (hardware) for details on individual registers.
Initial values of individual bits
x: Bits not used in this application; blank spaces: bits that do not change; -: reserved bits or bits that have nothing assigned.

5.7.16 INTP0 Interrupt Sub Routine

Figure 5.22 shows the flowchart for INTP0 interrupt sub routine.

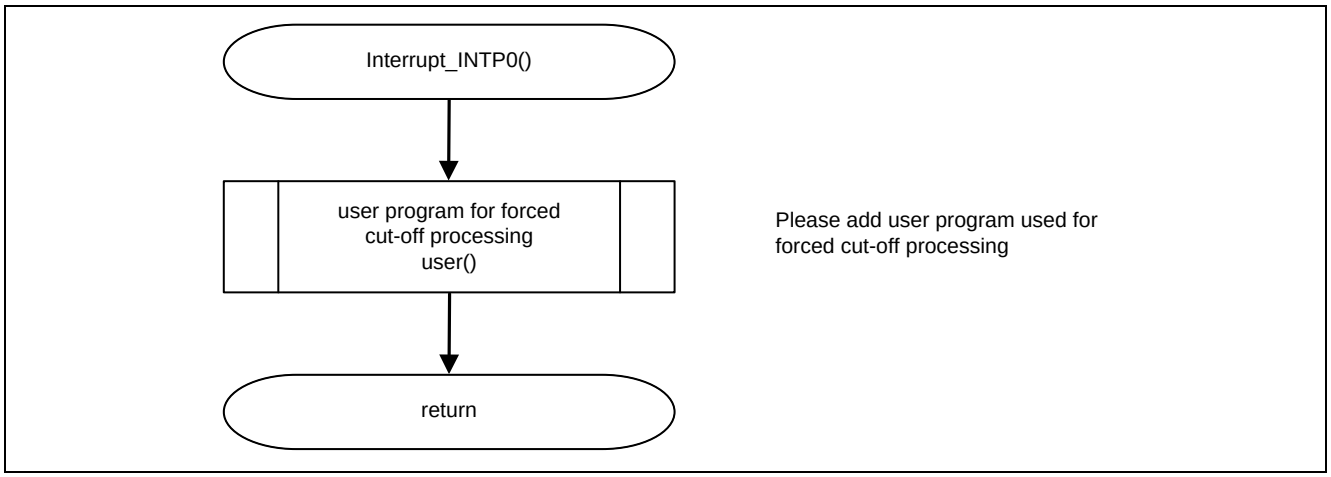


Figure 5.22 INTP0 Interrupt Sub Routine

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Reference Documents

User's Manual

R7F0C806-809 User's Manual: Hardware (R01UH0481E)

RL78 Family User's Manual: Software (R01US0015E)

The latest versions of the documents are available on the Renesas Electronics Website.

Technical Updates/Technical News

The latest versions of the documents are available on the Renesas Electronics Website.

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep.30, 2014	77	First edition issued.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "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 etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0