PFC-Controlled and Sensorless Vector-Controlled by Ceiling FAN Inverter Board with RX13T

SUMMARY

This application note describes for DC brushless motor (Ceiling FAN) control using "Ceiling FAN Inverter Board with RX13T" made by Desk Top Laboratories Inc. The algorithm uses vector control (FOC control). This application note describes how to use the sample program and the motor control development support tool (Renesas Motor Workbench 2 0). The sample program does not guarantee this behavior. When using the sample program, thoroughly evaluate it in an appropriate environment. In particular, handling a high-voltage environment is extremely dangerous. Please read the user's manual of each development environment carefully and use it safely. Renesas cannot be held responsible for any accidents or damages that may occur in the development environment listed in this application note.

Operation checking device

The following devices are used to check the operation of the sample program.

- RX13T(R5F513T5ADFL)

Target sample programs

The target sample program of this application note is shown below.

RX13T Vector Control Software for Ceiling FANs
- E1-001_RX13T_CeilingFan_Drive_B009-CSP_RV100 (IDE: CS+ version)
- E1-001_RX13T_CeilingFan_Drive_B009-E2S_RV100 (IDE:e2studio Version)

Target hardware

The target hardware of this application note is shown below.

Ceiling Fan Inverter Board with RX13T from Desk Top Laboratories Inc.
- APINV003-RX13T-CEILING

References

- RX13T Group User's Manual: Hardware (R01UH0822)
- Encoder Vector Control for Permanent-Magnet Synchronous Motors (Algorithms) (R01AN3789)
- Renesas Motor Workbench User's Manual (R21UZ0004)
- Smart Configurator User's Manual, RX API Reference (R20UT4360)
- RX Smart Configurator User Guide: CS+ Edition (R20AN0470)
- RX Smart Configurator User Guide: e2 studio (R20AN0451)

Contents

1. Overview ... 3
1.1 Development environment ... 3

2. System overview ... 4
2.1 System configuration ... 4
2.2 System features ... 4
2.3 Board configuration ... 4
2.4 Hardware configuration ... 6
2.5 Hardware specification ... 7
2.5.1 User interface ... 7
2.5.2 Peripheral functions ... 7
2.6 Software structure ... 8
2.6.1 Software file structure ... 8
2.6.2 Module configuration ... 9
2.7 Software specifications ... 10

3. Control program description ... 10
3.1 Control content ... 10
3.1.1 Start and stop of motor ... 10
3.1.2 Motor rotation speed command value ... 11
3.1.3 A/D conversion ... 11
3.1.4 Speed control ... 12
3.1.5 Modulation ... 12
3.1.6 Voltage control by PWM ... 13
3.1.7 Vector control ... 13
3.1.8 Pick-up control ... 14
3.1.9 Power factor correction (PFC) control ... 14
3.1.10 State transition ... 16
3.1.11 Starting method for sensorless control ... 16
3.1.12 System protection functions ... 17
3.2 Function specifications ... 18
3.3 Variable list ... 20
3.4 Macro definition ... 25
3.5 Control flow (flowchart) ... 31
3.5.1 Main processing ... 31
3.5.2 Carrier period interrupt processing ... 32
3.5.3 1 [ms] interrupt processing ... 37
3.5.4 Math library processing ... 42

4. Motor control support tool "Renesas Motor Workbench" ... 46
4.1 Overview ... 46
4.2 List of variables for analyzer function ... 47
4.3 Operation example for analyzer ... 47

5. Appendix「FAQ」 ... 49

1. Overview

This application note describes how to use RX13T microcontroller to implement the vector control sample program for the Permanent Magnetic Synchronous Motor (PMSM) [*1] and how to use the motor control development support tool [*2] "Renesas Motor Workbench" (RMW). This sample program uses the vector control algorithm of Desk Top Laboratories Inc. For questions regarding algorithms, please contact Desk Top Laboratories Inc.

[Notes] 1. Another name: Brushless DC motor (BLDC)
2. You can also use In Circuit Scope (ICSs) made by Desk Top Laboratories Inc.

1.1 Development environment

Table 1-1 and Table 1-2 show the development environments of the sample programs targeted for this application note.

Table 1-1 : Sample Program Development Environment (H/W)

Microcomputer	Evaluation board	Motor
RX13T (R5F513T5ADFL)	Desk Top Laboratories Inc. Ceiling with RX13T FAN-Inverter Board *[1] (APINV003-RX13T-CEILING)	Palkdale's *[2]

Table 1-2 : Sample Program Development Environment (S/W)

CS+ version	e^2studio Version	Toolchain version
V8.05.00	2021-01	CC-RX:V3.03.00

For purchasing or technical support, please contact our sales office or dealer.

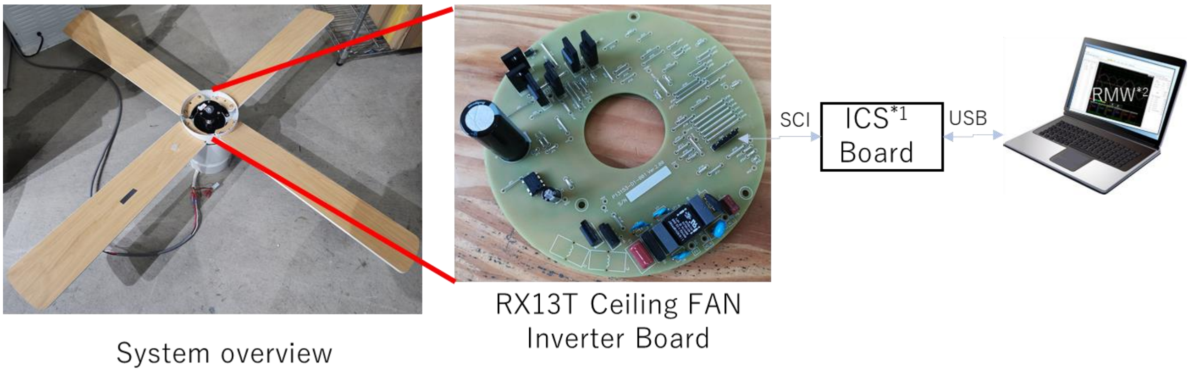
[Notes]1. Desk Top Laboratories Inc., Inc. uses an inverter board for ceiling fans.
Desk Top Laboratories Inc. (http://www.desktoplab.co.jp/)
2. Four-pole motors with Palkdale ceiling fans are used.

2. System overview

This section provides an overview of the system.

2.1 System configuration

The system configuration is shown below.



*1: USB-to-SCI Conversion Isolation Board (Desktop Lab, Inc.)
*2: Renesas Motor Workbench 2.0 You can also use In Circuit Scope (ICSs) made by Desktop Lab, Inc.

Figure 2-1 : System configuration diagram

2.2 System features

The following table describes the features of the Ceiling FAN System using the Ceiling FAN Inverter Board with RX13T manufactured by Desk Top Laboratories Inc.

Table 2-1 : System features

Classification	Content
Hardware	This board is one-sided mounted for the ceiling FAN with narrow mounting space. Suitable for the system configuration of the entire system.
	The PFC control is controlled higher frequency (50KHz) for the commercial power input up to 260V. As a result, the inductor can be made smaller (4mH), and the inductor parts can be made onboard. (Generally external to the board)
	Low-pin count small package (LQFP48 (7mm□)) improves board layout flexibility.
	Contributes to reducing the external components by using the RX13T functions. (3-channel simultaneous sample-and-hold circuit, built-in programmable gain amplifier (PGA), etc.)
Software	Three-shunt sensorless vector control at RX13T@32MHz with one chip. (including pick-up control and current continuous mode PFC control (inverter asynchronous control))
	The power factor improves PFC control to high frequency (carrier frequency: 50KHz) and greatly improves up to 99%.
	Motor drive can be without completely stopping the motor by pick-up control. Velocity can be changed steplessly by 3-shunt sensorless vector control (carrier frequency: 16KHz).

* PFC: power factor correction

2.3 Board configuration

Figure 2-2 shows the configuration of RX13T equipped ceiling FAN inverter board (APINV003-RX13T-CEILING) using Renesas components as main components, and Table 2-2 lists the components manufactured by Renesas.

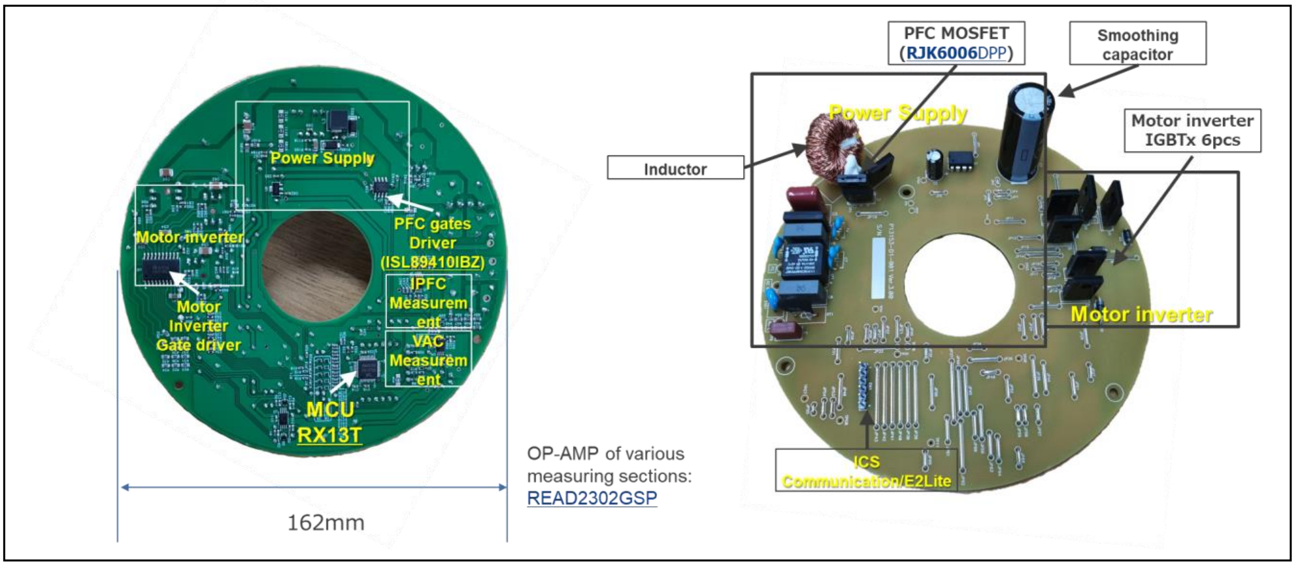


Figure 2-2 : Board configuration diagram

Table 2-2 : Renesas components

Item	Detail	Product Name	Package
PFC MOSFET	Nch Single Power MOSFET 600V 5A 1600Mohm	RJK6006DPP	TO-220FPA
PFC Gate Driver	High Speed, Dual Channel Power MOSFET Drivers	ISL89410IBZ	8-SOIC
Measurement Parts	MSIG CMOS DUAL OPAMP HI-SPEED TS	READ2302GSP#GC3	TSSOP8
Control MCU	RX MCU 32BIT 128KB FLASH 48LFQFP	R5F513T5ADFL	48-LQFP

2.4 Hardware configuration

The hardware configuration is shown below.

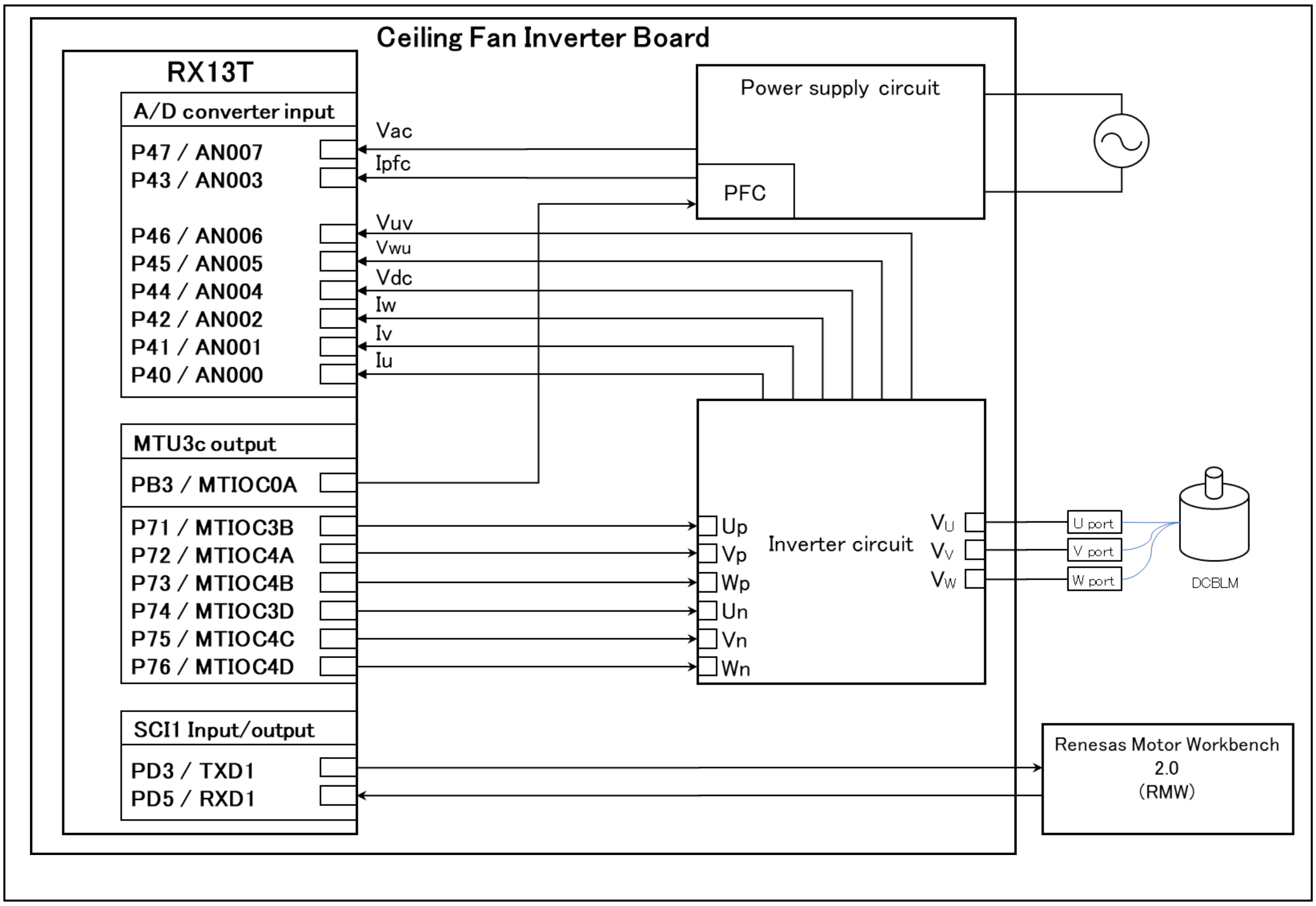


Figure 2-3 : Hardware configuration diagram

2.5 Hardware specification

2.5.1 User interface

This system using Motor Control Development Support Tool Renesas Motor Workbench2 0 (RMW) or Desk Top Laboratories Inc. In Circuit Scope (ICS). Therefore, therefore no hardware user interface. This tool is direct manipulation of system control variables during system operation to control the system. For more details, refer to the Motor Control Development Support Tool "Renesas Motor Workbench" in section 4.

Table 2-3 lists the interfaces of RX13T microcontroller pins of this system.

Table 2-3 : Pin interface

R5F513T5ADFL Pin name	Signal Name	IN/OUT	Function
P47 / AN007	Vac	IN	AC Voltage
P43 / AN003	Ipfc	IN	PFC current measurement
P46 / AN006	Vuv	IN	Vuv line voltage measurement
P45 / AN005	Vwu	IN	Vwu line voltage measurement
P44 / AN004	Vdc	IN	Inverter bus voltage measurement
P42 / AN002	Iw	IN	W-phase current measurement
P41 / AN001	Iv	IN	V-phase current measurement
P40 / AN000	Iu	IN	U-phase current measurement
P71 / MTIOC3B	Up	OUT	Port out/PWM out (U_p)/ "Low" active
P72 / MTIOC4A	Vp	OUT	Port out/PWM out (V_p)/ "Low" active
P73 / MTIOC4B	Wp	OUT	Port out/PWM out (W_p)/ "Low" active
P74 / MTIOC3D	Un	OUT	Port out/PWM out (U_n)/ "High" activated
P75 / MTIOC4C	Vn	OUT	Port out/PWM out (V_n)/ "High" activated
P76 / MTIOC4D	Wn	OUT	Port out/PWM out (W_n)/ "High" activated
PB3 / MTIOC0A	PFC	OUT	Port Out/PWM Out (for PFC)/"High" Active
PD3 / TXD1	TX	OUT	Renesas Motor Workbench2.0(RMW) Or for Desk Top Laboratories Inc. In Circuit Scope (ICS) serial communication
PD5 / RXD1	RX	IN	

2.5.2 Peripheral functions

Table 2-4 lists the peripheral functions used for this system.

Table 2-4 : Correspondence between peripheral functions by sample program

12bit A/D	CMT	MTU3	SCI1
• AC voltage measurement • PFC current measurement • Inverter bus voltage measurement • Voltage measurement between Vuv and Vwu lines • Respective U/V/W phase currents	1 [ms] Interval timer (interval for rotation speed command)	• Complementary PWM output for motor control • PWM output for PFC control	Serial communication for RMW

(1) 12-bit A/D converter

- Use a 12-bit A/D converter to measure AC voltage (Vac), PFC current (Ipfc), inverter bus voltage (V_{dc}), line voltage (Vuv, Vwu), U-phase current (I_u), V-phase current (I_v), and W-phase current (I_w).
- Set the operation mode to "Single scan mode". (Using hardware trigger)
- The U-phase current (I_u), V-phase current (I_v), and W-phase current (I_w) are measured in "single scan mode" using the channel-dedicated sample-and-hold function and the programmable gain amplifier (gain setting = 5 times) (using a hardware trigger).

(2) Compare match timer (CMT)

1 [ms] Interval timer: Channel 0 of the compare match timer is used as the 1 [ms] interval timer.

(3) Multi-function timer pulse unit 3 (MTU3)

The operation mode differs for each channel.

- Complementary PWM mode is used for channels 3 and 4, and complementary PWM output with dead-time for motor control (p side is "low" active and n side is "High" active) is performed.
- PWM mode 1 is used for channel 0, and PWM output for PFC control of "High" is performed.

(4) Serial Communications Interface 1 (SCI1)

Performs communication with the RMW. The baud rate is set to 4Mbps.

2.6 Software structure

2.6.1 Software file structure

Table 2-5 shows the folder and file structure of the sample program.

Table 2-5 : Sample program folders and file structure

E1-001_RX13T_CeilingFan_Drive_B009	src	main.c	Main function, user interface control
		intprg.c	Interrupt handler
		r_mtr_control.c	Motor control, PFC control function
		r_mtr_estimation.obj	Estimated function [*1]
		r_mtr_interrupt.c	Motor control, PFC control, interrupt processing
		r_mtr_sequence.c	Sequence control
		r_mtr_control.h	Motor and Inverter Parameter Related Definitions
		r_mtr_interrupt.h	Control parameter definition
	src¥REL_src	resetprg.c	Power-on processing
		dbsct.c	B, R section setting
		sbrk.c	Memory allocation process
		vecttbl.c	Vector table initialization processing
		iodefine.h	RX13T IO register definitions
		sbrk.h	Allocation size definition
		stacksct.h	Stack area size definition
		typedefine.h	Type definitions
		vect.h	Vector definition
	src¥Motor_IOLIB	r_mtr_IOLIB_AD.c	RX13T S12AD related procedures
		r_mtr_IOLIB_CMT.c	RX13T CMT related procedures
		r_mtr_IOLIB_INV_M	RX13T MTU3 related process

		TU_AD.c	
		r_mtr_MATHLIB.c	Arithmetic processing for vector control
		r_mtr_MATHLIB.h	Arithmetic operation related definition for vector control
		r_mtr_IOLIB.h	Definition of MCU dependent part
	src¥ICS_Lib	ICS2_RX13T.h	Communication-related definitions for tools
		ICS2_RX13T.lib	Communication library for tools

Note 1: The estimation function is an object.

2.6.2 Module configuration

Figure 2-4 and Table 2-6 show the module configuration of the sample program.

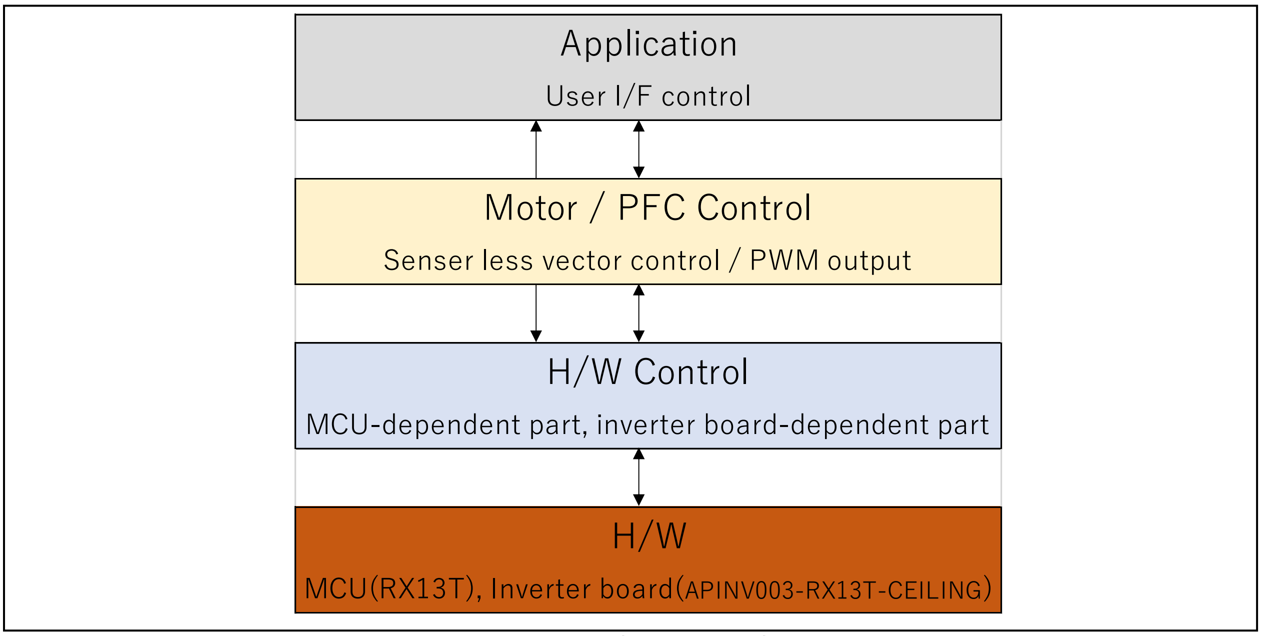


Figure 2-4 : Module configuration of sample program

Table 2-6 : Sample program module configuration

Hierarchy	File
Application	main.c
Application Motor/PFC Control	r_mtr_sequence.c *1
Motor/PFC Control	r_mtr_control.c
	r_mtr_estimation.obj
Motor/PFC Control H/W Control	r_mtr_interrupt.c *2
H/W Control	r_mtr_Custom_IO.c
	r_mtr_IOLIB_AD.c
	r_mtr_IOLIB_CMT.c
	r_mtr_IOLIB_INV_MTU_AD.c
Library	r_mtr_MATHLIB.c
	ICS2_RX13T.lib

[Note] 1. "r_mtr_sequence.c" handles the application layer and the motor control layer.
2. "r_mtr_interrupt.c" handles the motor control layer and the H/W layer.

2.7 Software specifications

The basic specifications of the software of this system are shown in Table 2-7 Basic specifications of the sensorless vector control software.

Table 2-7 : Basic specifications of sensorless vector control software

Item	Content
Control method	Sensorless vector control
Motor control start/stop	Decision by inputting the value of RMW (High (1): Control started, Low(0): Stopped)
Rotor pole position detection	Shunt Resistance Estimation
Input voltage	AC100~260V
Main clock frequency	32[MHz]
Carrier Frequency (PWM)	16[kHz] (Carrier period: 62.5 [μs])
Dead time	2[μs]
Control period (current)	125[μs] (2 times carrier period)
Control period (speed)	1[ms]
Speed command value control	CW:0 [rpm]～250 [rpm] CCW:0 [rpm]～250 [rpm]
Compiler optimization settings	Optimization Level 2 (-optimize = 2) (Default setting)
Optimal manner	Optimize code-size emphasis (-size) (default setting)
Protection stop processing	Deactivate the motor control signal outputs (6) under any of the following conditions:
	1.Phase current (Iu, Iv, Iw) exceeds 2.06 [A] (monitored every 125 [μs])
	2.Inverter bus voltage (Vdc) exceeds 300 [V] [*1] (monitored every 125 [μs])
	3.Inverter bus voltage (Vdc) is less than 50 [V] [*1] (monitored every 125 [μs])

*1: Settings for entering AC100V. When entering AC260V, the protective voltage level must be raised.

(Reference at AC260V: High voltage protection = 420 [V], Under voltage protection = 200 [V])

3. Control program description

This section describes the sample programs covered in this application note.

3.1 Control content

3.1.1 Start and stop of motor

Starting and stopping of the motor is controlled by inputting from Analyzer of RMWs.

Read the variable value (g_us_mode_system_request) at CMT0 interrupt cycle (speed control cycle [1ms]). When High level (1), it is judged that the motor is started. When the speed is low (0), it is judged that the motor is stopped.

Table 3-1 : Motor Start/Stop command values

Item	Command value		Variable Name
Start and stop of motor	Start	1	g_us_mode_system_request
	Stop	0	

3.1.2 Motor rotation speed command value

The motor speed command value is determined by inputting from the RMW Analyzer.

It is used as the rotation speed command value as shown in the following table.

Table 3-2 : Rotation speed command value

Item	Rotational speed: Command value		Variable Name
Rotation speed command value	CW	0 [rpm] to 250 [rpm]：0 to 250	g_f_rpm_ref_request
	CCW	0 [rpm] to 250 [rpm]：0 to -250	

3.1.3 A/D conversion

(1) Converter AC voltage

Measure the converter AC voltage as shown below. Used to calculate the modulation factor and detect the over voltage.

Table 3-3 : Conversion ratio of converter AC voltage

Item	Conversion ratio (converter AC voltage: A/D conversion value)	Channel
Converter AC voltage	-750 [V] to 750 [V]：0000H to 0FFFH	AN007

(2) Converter PFC current

Measure the converter PFC current as shown below. Used to calculate the modulation factor and detect the over voltage.

Table 3-4 : Conversion ratio of converter PFC current

Item	Conversion ratio (Converter PFC current: A/D conversion value)	Channel
Converter PFC current	-1.85 [A] to 1.85 [A]：0000H to 0FFFH	AN003

(3) Inverter bus voltage

Measure the inverter bus voltage as shown below. Used to calculate the modulation factor and detect the over voltage.

Table 3-5 : Conversion ratio of inverter bus voltage

Item	Conversion ratio (inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 300 [V]：0000H to 0FFFH	AN004

(4) UV-phase, WU-phase line voltage

Measure the line voltage of the U, V, and W phases as shown below.

Table 3-6 : Conversion ratio of UV phase and WU phase line voltage

Item	Conversion ratio (UV phase, WU phase line voltage: A/D conversion value)	Channel
UV phase line voltage	-750 [V] to 750 [V]：0000H to 0FFFH	AN006
WU phase line voltage		AN005

(5) U-phase, V-phase, and W-phase currents

As shown in the table below, the U-, V-, and W-phase currents are measured and used for vector control.

Table 3-7 : Conversion ratio of U, V and W phase current

Item	Conversion ratio (U-phase, V-phase, W-phase current: A/D conversion value)	-channel

U phase current (Iu)	-2.06 [A] to 2.06 [A] : 0000H to 0FFFH[*1]	AN000
V phase current (Iv)		AN001
W phase current (Iw)		AN002

[Note] 1. For more details of A/D converter characteristics, refer to RX13T Group User's Manual: Hardware.

3.1.4 Speed control

The target sample software [explained in this application note] uses PI control for speed control. A voltage command value is calculated by the following formula of speed PI control.

$$v^{*} = (K_{P\omega} + \frac{K_{I\omega}}{s})(\omega^{*} - \omega)$$

v^*: Voltage command value, ω^*: Speed command value, ω: Rotation speed

$K_{P\omega}$: Speed PI proportional gain, $K_{I\omega}$: Speed PI integral gain, s : Laplace operator

For more details of PI control, refer to specialized books.

3.1.5 Modulation

In this application note target software, the input voltage to the motor is generated by pulse-width modulation (hereafter PWM), and PWM waveform is generated by the triangle-wave compare method. The triangle waveform comparison method determines the pulse width of the output voltage by comparing the carrier waveform (triangle waveform) with the command value voltage waveform as one of the methods for actually outputting the command value voltage. If the command value voltage is greater than the carrier wave voltage, the switch can be turned on, and if it is smaller, the sinusoidal command value voltage can be pseudo-output.

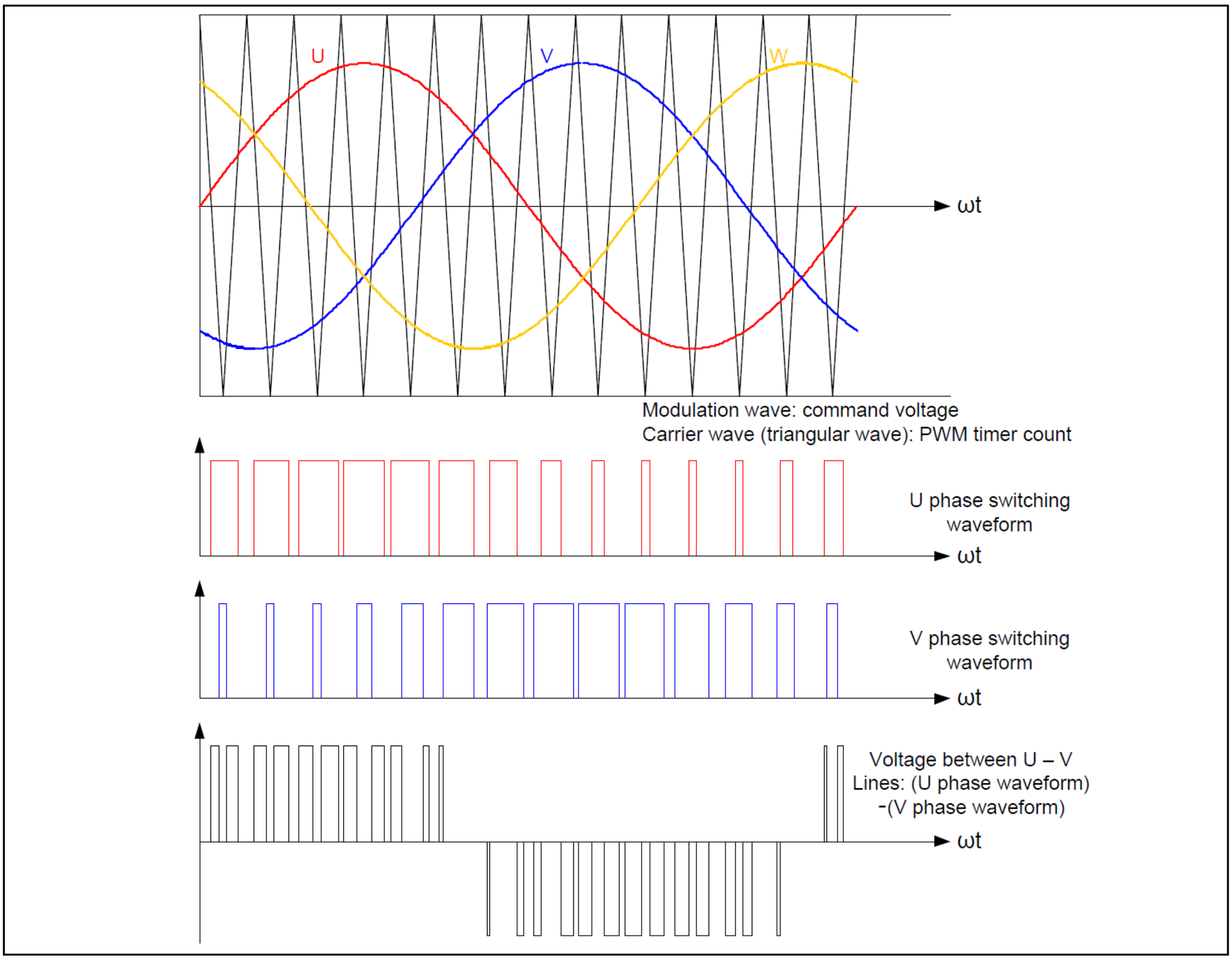


Figure 3-1 : Conceptual diagram of triangle-wave comparison method

3.1.6 Voltage control by PWM

PWM control is used for controlling output voltage. The PWM control is a control method that continuously adjusts the average voltage by varying the duty of pulse, as shown in Figure 3-2.

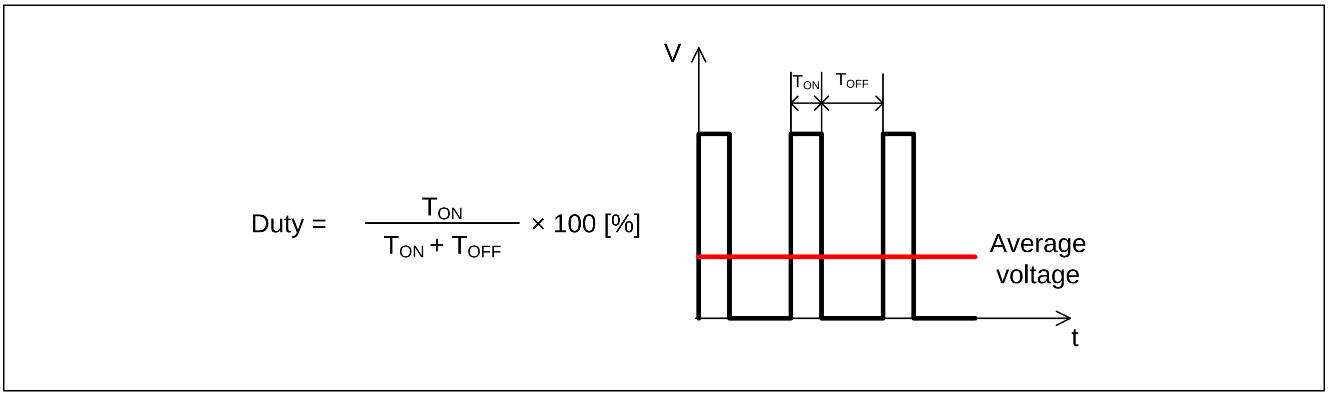


Figure 3-2 : PWM control

The modulation factor m is defined as follows.

$$m = \frac{V}{E}$$

m : Modulation factor V : Command value voltage E : Inverter bus voltage

This modulation ratio is reflected to the setting value of the register that determines the PWM duty.

3.1.7 Vector control

Figure 3-3 shows an overview of sensorless vector control. This software uses a vector control algorithm made by Desk Top Laboratories Inc. However, you can refer to the Application Note "Vector Control of Permanent Magnet Synchronous Motors (Algorithms)" for reference. If you have questions about algorithms, please contact Desk Top Laboratories Inc.

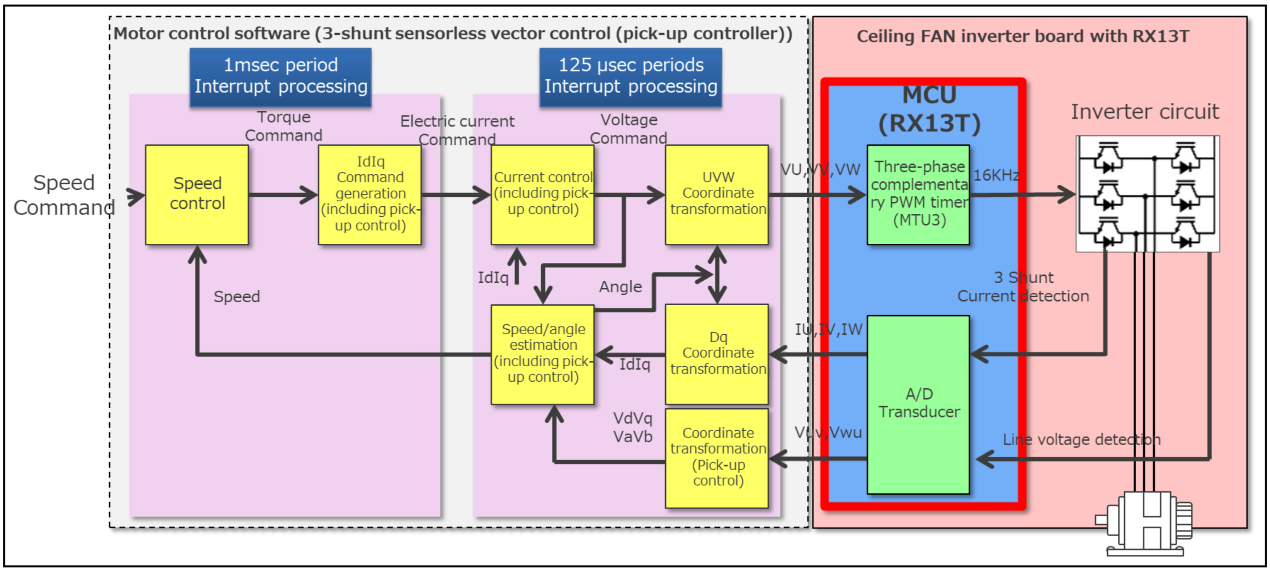


Figure 3-3 : Vector control overview

3.1.8 Pick-up control

The pick-up control is to measure the induced voltage of a coasting motor and adjust the inverter output voltage so that the induced voltage of the motor and the output voltage of the inverter are the same value. To do this, calculate the offset relative to the rotation angle, rotation speed, and speed command. This suppresses a sudden change in the current even when driving from the free-run state, allowing the motor to continue to rotate smoothly.

Since it is difficult to calculate the angle using only the output terminal (Vu/Vv/Vw) when the inverter is stopped, calculate the line voltage (Vuv/Vvw/Vwu) from the phase voltage to determine the angle and speed. (UVWOOBAB coordinate transformation of vector control is used to obtain Va/Vb and calculate the angle.)

The calculation procedure is shown below.

① Calculate Va and Vb with the line voltage (u=Vuv, w=Vwu).

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{3/2} & 0 \\ -\sqrt{2}/2 & -\sqrt{2} \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix}$$

② Calculates the angle from Va and Vb.

$\theta=\mathrm{atan}(Vb/Va)$

③ Calculates Vd and Vq from λ, Va, and Vb.

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

④Calculate the speed by combining Vd and Vq with the induced voltage.

(The vector value of line voltage is $1/\sqrt{3}$ because it is $\sqrt{3}$ times larger than the phase voltage.)

$Spd1=\sqrt{(Vd^2+Vq^2)} / (Ke*\sqrt{3})$

⑤Calculate the speed by performing the differential calculation of theta, and then decide the direction.

$Spd2=\Delta\theta/\Delta t$

Since the angle calculated by equation ② is based on the line voltage (Vuv,Vwu), a phase of 30 degrees is generated from the angle of Vu,Vw. Also, even when the angle is calculated with Vu and Vw, the angle is Va and Vb, so there is a phase of 90 degrees from the angle of Vd and Vq actually used. Consequently, since the angle calculated by the line voltage Vuv/Vvw/Vwu has a phase difference of 120 degrees from the angle in actual current control, the angle used in vector control is offset corrected by 120 degrees as follows.

Theta (true) = theta-120 degrees

3.1.9 Power factor correction (PFC) control

To improve the power factor, this system employs PFC control with the current continuous mode single method. Table 3-8 lists the PFC control specifications, and Figure 3-4 shows the PFC control overview of the current continuous mode single method.

Note that the PFC control section operates asynchronously with the inverter control section. Shunt current sampling does not need to be timed, and software processing is more flexible. However, switching noise may be superimposed during shunt current sampling. For details, see the advantages and disadvantages of asynchronous control in Figure 3-5.

Table 3-8 : PFC control specifications

Item	Content	Remarks
Input voltage	260Vac	In this software, we are checking on 100VAC.
Input current	2A or less	In this software, we are checking on 1A.
Current sampling	Shunt	270mΩ

PFC control cycle	8kHz	
PFC carrier	50kHz	
Synchronization with the Inverter	Asynchronous	The inverter and PFC sections are controlled completely independently, eliminating the need for timing matching, such as shunt current sampling.

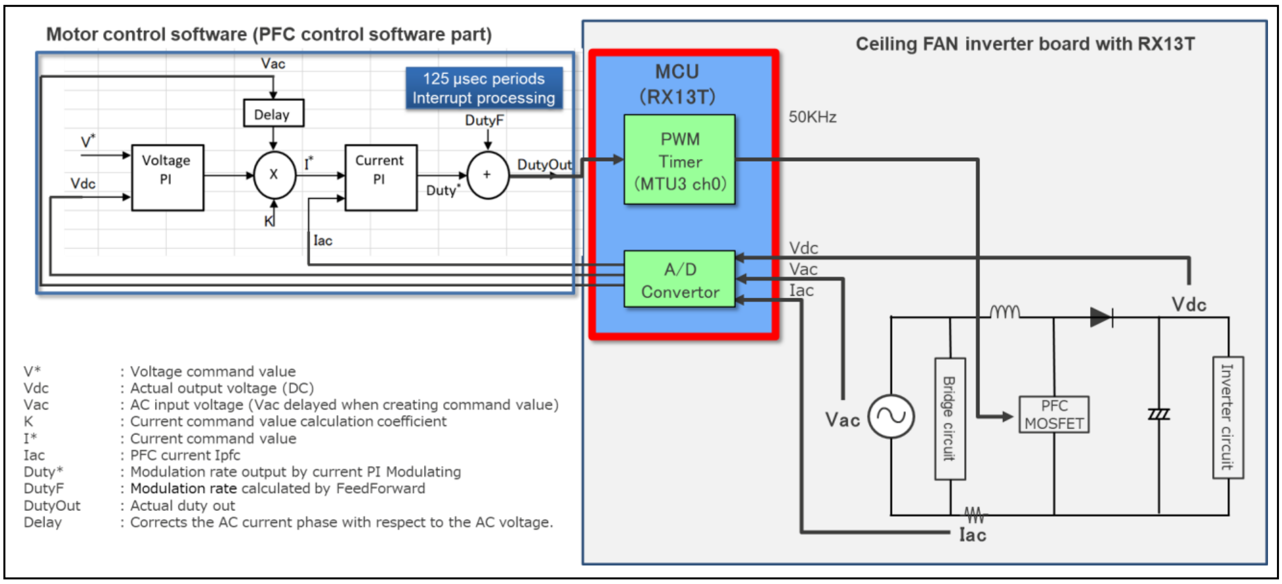
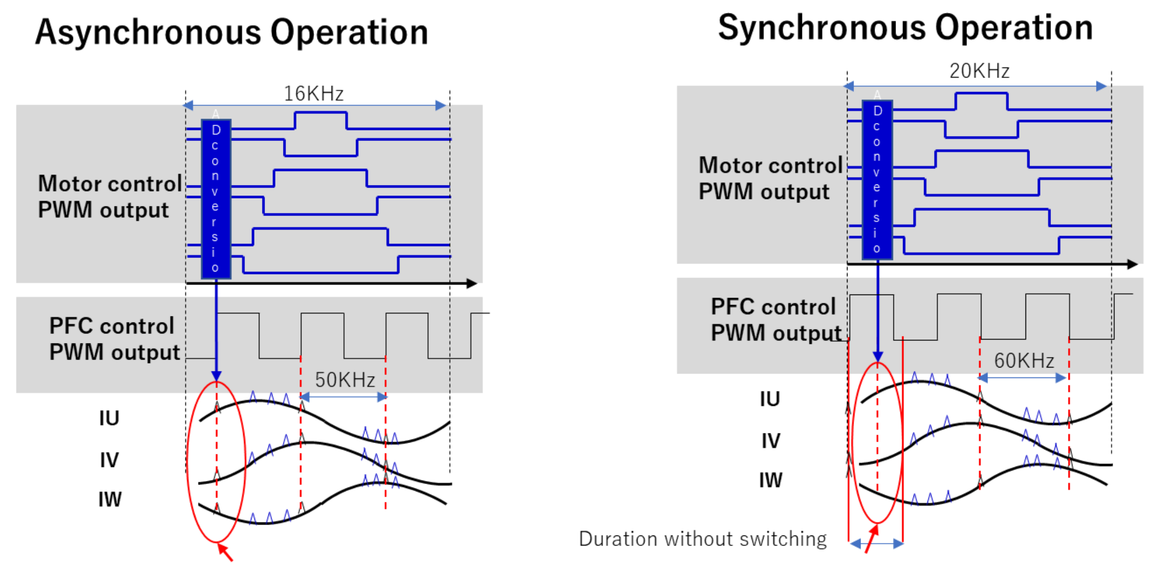


Figure 3-4 : PFC control overview



The PFC-controlled switching noise is superimposed on the motor-controlled UVW shunt measurement line, which may reduce accuracy. Therefore, measures such as enhancing the power supply line and inserting filters may be necessary.

Because PFC control is not switched, noise does not occur and it does not affect the UVW shunt measurement line of motor control. The disadvantage is that the shunt current measurement of the PFC and the inverter section must be carried out in the same period, and the switching frequency of the PFC section must be an integer multiple of the carrier frequency of the inverter section. (PFC part: Inverter part must be an integral multiple as shown in 60KHz:20KHz above.)

Figure 3-5 : Advantages and disadvantages of asynchronous control

3.1.10 State transition

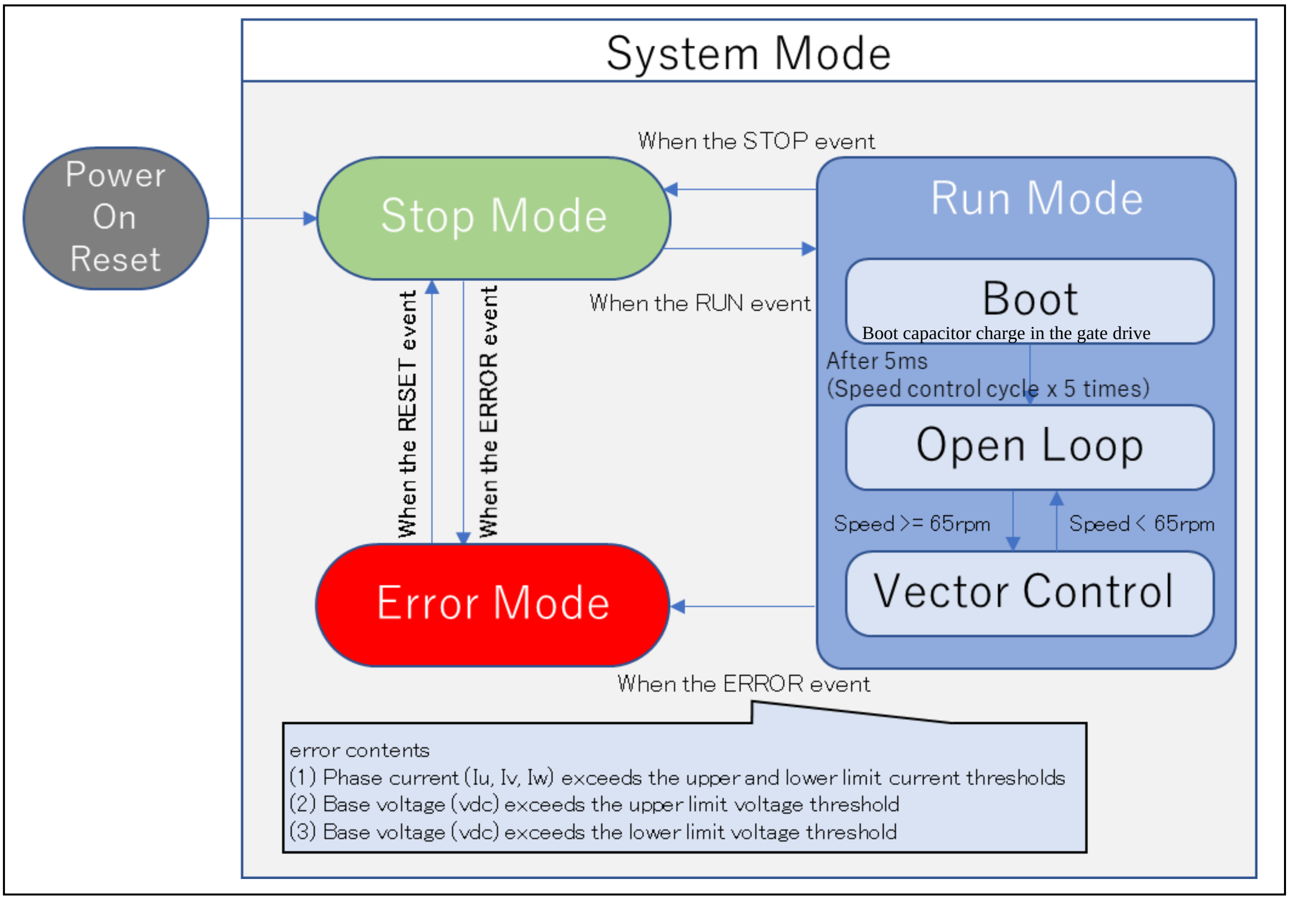


Figure 3-6 : State transition diagram

Table 3-9 : Status table

		Current Mode		
		STOP(0)	RUN(1)	ERROR(2)
EVENT	STOP(0)		STOP	ERROR
	RUN(1)	RUN		ERROR
	ERROR(2)	ERROR	ERROR	
	RESET(3)	STOP	ERROR	STOP

3.1.11 Starting method for sensorless control

As shown in Figure 3-6 Status Transition Diagram, Boot capacitor charge period of the gate drive circuit (5 times of the speed control period (1ms)) is passed before entering Open Loop. Switching between Open Loop and Vector Control is performed by the specified rotation speed. (65rpm in this software). When the specified rotation speed is reached, Id is turned OFF. (85rpm in this software).

Figure 3-7 shows the control switching timing from Open Loop to Vector Control.

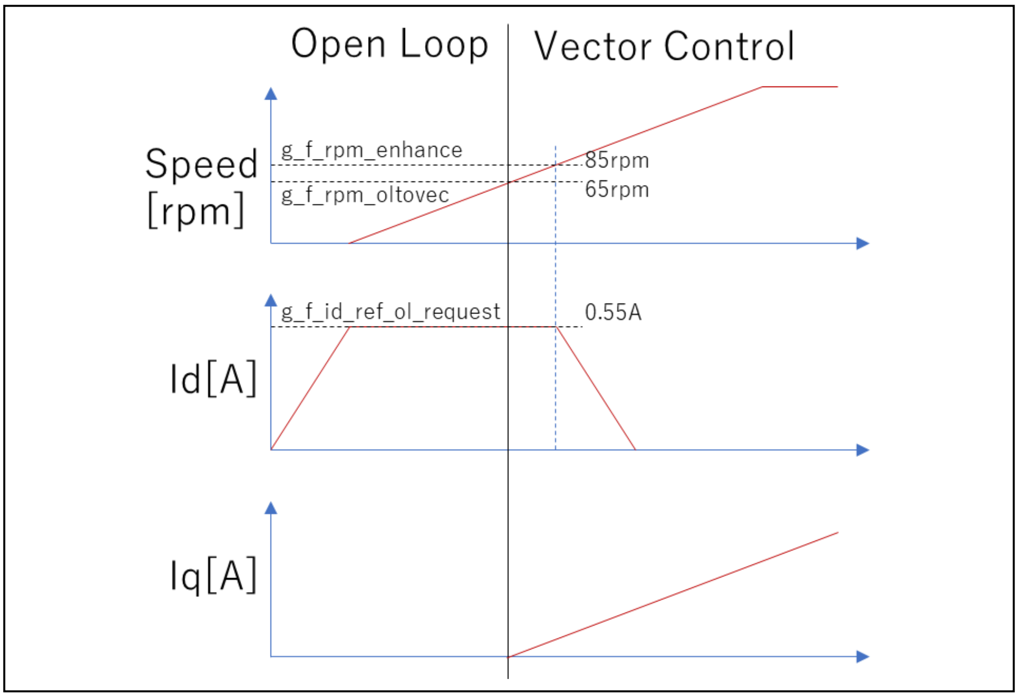


Figure 3-7 : Control switching diagram

3.1.12 System protection functions

This system has the following error conditions, and the emergency stop function is realized in each case. Refer to Table 3-10 for the setting values related to the system protection function.

- Over current error
  Monitors U-, V-, and W-phase currents in the overcurrent monitoring cycle, and makes an emergency stop when an overcurrent (exceeding the overcurrent limit value) is detected. (Software Detection)
- Over voltage error
  The inverter bus voltage is monitored in the over voltage monitoring cycle, and when an over voltage (exceeding the over voltage limit value) is detected, an emergency stop is performed. (Software Detection)
  Here, the over voltage limit value is set by considering the error of the resistance value of the detection circuit, etc.
- Low voltage error
  The inverter bus voltage is monitored in the low voltage monitoring cycle, and an emergency stop is performed when a low voltage (lower than the low voltage limit value) is detected. (Software Detection)
  Here, the low voltage limit value is a value set in consideration of errors in the resistance value of the detection circuit, etc.

Table 3-10 : Setting values of system protection functions

Over current error	Over current limit [A]	2.06
	Watching interval [µs]	125
Over voltage error	Over voltage limit [V]	300(during AC100V)
	Watching interval [us]	125
Low voltage error	Low voltage limit [V]	50(during AC100V)
	Watching interval [us]	125

3.2 Function specifications

The function list of this control program is shown below. Some functions are provided for extension (functions not used in this software) so that they can be easily incorporated by the user. Please refer to Notes in the table below for non-use functions.

Table 3-11 : Function lists

File			Function				Notes
Path	Name	Overview	Name	Overview	Arguments	Return Type	
src	main.c	Main file	main	Main processing	-	void	
	r_mtr_control.c	Motor algorithm	r_mtr_user_init_setting	Initializing process	-	void	
			r_mtr_user_init_control	Control initialization process	-	void	
			user_control_speedloop	Speed control processing	-	void	
			r_mtr_timer_control	Timer control processing	-	void	
			r_mtr_User_CustomIO_init	IO initializing process	-	void	
			r_mtr_Ref_Check	Estimated processing	float mu, float mv, float mw	int16_t	
			r_mtr_check_error_curloop	Checking the reference value	-	void	
			r_mtr_check_error_speedloop	Current loop error check processing	-	void	None Code
			r_mtr_error_stop	Speed loop error check processing	-	void	
			r_mtr_pfc_control	Error stop processing	-	void	
	r_mtr_estimation.obj		r_mtr_Estimation	PFC control processing	-	void	
	r_mtr_interrupt.c	Interrupt processing	interrupt_CMT0	1 ms speed control processing	-	void	
			interrupt_MTU34_carrier	250 us Carrier Cycle Processing	-	void	
	r_mtr_sequence.c	Sequence Processing	seq_act_run	Execution sequence processing	uint8_t ucState	uint8_t	
			seq_act_stop	Stop Sequence Processing	uint8_t ucState	uint8_t	
			seq_act_reset	Reset Sequence Processing	uint8_t ucState	uint8_t	
			seq_act_error	Error Sequence Processing	uint8_t ucState	uint8_t	
			seq_act_none	Unprocessed sequence processing	uint8_t ucState	uint8_t	
			seq_init_start	Sequence initialization processing	-	void	
			r_mtr_seq_init_setting	Initial sequence processing	-	void	
			r_mtr_seq_exec_event	Event Execution Sequence Processing	uint8_t ucEvent	void	
src¥Motor_IOLIB	r_mtr_Custom_IO.c	Hardware configuration	r_mtr_Custom_VRL_init	Hardware initialization (VRL) processing	-	void	For expansion
			r_mtr_Custom_T2001_SW_init	Hardware Initialization (Switch) Processing	-	void	For expansion
			r_mtr_Custom_T2001_LED_init	Hardware Initialization (LED) Processing	-	void	For expansion

	r_mtr_IOLIB_AD.c	ADC Configuration	r_mtr_ad_AD0_init	AD0 initialization process	uint16_t mode	void	
			r_mtr_ad_AD0_set_channel	AD0 channel set process	uint16_t ch_list	void	
			r_mtr_ad_AD0_set_range	AD0 range setting process	int16_t ch, int16_t offset, float range	void	
	r_mtr_IOLIB_CMT.c	CMT settings	r_mtr_interval_CMT0_init	CMT0 initialization process	uint16_t freq	void	
			r_mtr_interval_CMT1_init	CMT1 initializing process	uint16_t freq	void	For expansion
	r_mtr_IOLIB_INV_MTU_AD.c	MTU3 Setting	r_mtr_inverter_MTU34_init	Complementary PWM-output MTU3(ch3, 4) Initialization process	int32_t usFreqCarrier, uint32_t usDeadtime, uint32_t usDecimation	void	
			r_mtr_inverter_MTU34_set_uvw_3shunt	MTU3(ch3, 4) Complementary PWM-output-value Duty setting process	float refu,float refv,float refw	void	For expansion
			r_mtr_inverter_MTU34_set_uvw_3shunt_VariCar	MTU3(ch3, 4) Complementary PWM-output-value Duty setting process	float refu, float refv, float refw, float FreqCar	void	
			r_mtr_inverter_MTU34_boot_mode_lower_on	Complementary PWM output value Low-side ON setting processing	-	void	
			r_mtr_inverter_MTU34_boot_mode_lower_off	Complementary PWM output value Low-side OFF setting processing	-	void	For expansion
			r_mtr_inverter_MTU34_boot_mode_off	Complementary PWM output mode stop processing	-	void	
			r_mtr_pwm_MTU0_set	PWM output value setting process for PFC control	float fDuty	void	
			r_mtr_pwm_MTU0_init	PWM output initialization processing for PFC control	uint16_t usFreq	void	
	r_mtr_IOLIB_IWDT.c	Independent WDT setting	r_mtr_IWDT_init	Independent WDT initialization process	-	void	For expansion
	r_mtr_MATHLIB.c	Coordinate conversion processing	r_mtr_uw2dq	Uw→dq conversion process	float iu, float iw, float theta, float * id, float * iq	void	
			r_mtr_uw2ab	Uw→αβ conversion process	float iu, float iw, float * ia, float * ib	void	
			r_mtr_ab2dq	Process of transforming the alphabeta → dq	float ia, float ib, float theta, float * id, float * iq	void	

			r_mtr_dq2ab	Dq→αβ conversion process	float vd, float vq, float theta, float * va, float * vb	void	
			r_mtr_ab2uvw	Alphabetic → uvw-transform process	float va, float vb, float * u, float * v, float * w	void	
			r_mtr_svpwm	Svpwm output voltage computation	float * refu, float * refv, float * refw	void	
			r_mtr_limit_PN	Data range limit processing	float data, float limitp, float limitn	float	
			r_mtr_limit_P	Positive data range limit processing	float data, float limitp	float	For expansion
			r_mtr_limit	Negative data range limit processing	float data, float limit	float	
src¥REL_src	resetprg.c	Power-on reset	PowerON_Reset_PC	Power-on process	-	void	
src¥REL_src	sbrk.c	Allocation	sbrk	Memory area allocation processing	size_t size	_SBYTE *	

3.3 Variable list

The following table lists the variables used in this control program. However, local variables are not listed. Variables (variables not used in this software) that are prepared for extension are included so that the user can easily incorporate some variables. Please refer to Notes in the table below for information on unused parameters.

Table 3-12 : Variable lists

VariableName	Type	Size	Overview	Notes
g_f_AD0_range	float [8]	32	Range value for S12AD ch0	
g_f_const_oetorpm	float	4	const for rad/s to rpm	
g_f_const_rpmtooe	float	4	const for rpm to rad/s	
g_f_ctrl_time	float	4	Tcarrier:1/f carrier	
g_f_ctrl_time_div_ke	float	4	Tcarrier:1/f carrier / ke	
g_f_ctrl_time_div_ld	float	4	Tcarrier:1/f carrier / id	
g_f_ctrl_time_div_lq	float	4	Tcarrier:1/f carrier / iq	
g_f_current_iac	float	4	iac	
g_f_current_id	float	4	d-axis current	
g_f_current_iq	float	4	q-axis current	
g_f_current_iu	float	4	iu	

g_f_current_iv	float	4	iv	
g_f_current_iw	float	4	iw	
g_f_delta_pe_gamma_lpf	float	4	delta pe Low pass filter gamma	
g_f_duty_ff	float	4	pfc duty ff output	
g_f_duty_ff_buf	float	4	pfc duty ff buffer	
g_f_duty_pfc	float	4	pfc duty output	
g_f_duty_ref	float	4	pfc duty reference	
g_f_duty_ref_i	float	4	pfc duty reference	
g_f_duty_ref_limit	float	4	pfc duty limit	
g_f_emf_est	float	4	Electromotive force estimation value	
g_f_emf_est_factor	float	4	Electromotive force estimation factor	
g_f_est_lpf_factor	float	4	Low pass filter coefficient value for speed	
g_f_factor_vuvw_lpf	float	4	Low pass filter for vu,vv,vw	
g_f_id_ref	float	4	id-axis reference current	
g_f_id_ref_enh_request	float	4	id reference enhance request	
g_f_id_ref_ol_request	float	4	max id reference @openloop	
g_f_id_ref_vec_request	float	4	max id reference @vecloop	
g_f_id_slope_down_vec	float	4	id down slope at down loop	
g_f_id_slope_down_vec_request	float	4	id down slope at vec loop(A/s)	
g_f_id_slope_ol	float	4	id slope(A/ms)	
g_f_id_slope_ol_request	float	4	id slope at open loop(A/s)	
g_f_id_slope_up_vec	float	4	id up slope at vec loop	
g_f_id_slope_up_vec_request	float	4	id up slope at vec loop(A/s)	
g_f_iq_limit	float	4	iq limit	
g_f_iq_ref	float	4	iq-axis reference current	
g_f_iq_ref_i	float	4	iq-axis reference I-current	
g_f_iq_ref_ol_request	float	4	max iq reference @openloop	No use
g_f_iq_ref_over	float	4	iq reference over value	
g_f_iq_slope_ol	float	4	iq slope(A/ms)@openloop	
g_f_iq_slope_ol_request	float	4	iq slope request(A/s)@openloop	
g_f_iref	float	4	pfc current reference	
g_f_iref_i	float	4	pfc current reference	
g_f_iref_limit	float	4	pfc current limit	
g_f_iref_over	float	4	pfc current reference	
g_f_iref_pfc	float	4	pfc current(ac) reference	
g_f_k_duty_ff	float	4	pfc duty ff const	

g_f_ki_factor_ipfc	float	4	pfc cur->duty PI	
g_f_ki_factor_v_pfc	float	4	pfc vol->cur PI	
g_f_ki_id	float	4	d-axis ki	
g_f_ki_iq	float	4	q-axis ki	
g_f_ki_oe	float	4	speed ki	
g_f_kp_factor_ipfc	float	4	pfc cur->duty PI	
g_f_kp_factor_v_pfc	float	4	pfc vol->cur PI	
g_f_kp_id	float	4	d-axis kp	
g_f_kp_iq	float	4	q-axis kp	
g_f_kp_oe	float	4	speed kp	
g_f_motor_ke	float	4	BEMF of motor (Vs/rad)	
g_f_motor_ld	float	4	d-axis inductance (H)	
g_f_motor_lq	float	4	q-axis inductance (H)	
g_f_motor_pp	float	4	motor pole-pairs	
g_f_motor_ra	float	4	res of motor(ohm)	
g_f_mu_ref	float	4	u-axis modulation	
g_f_mv_ref	float	4	v-axis modulation	
g_f_mw_ref	float	4	w-axis modulation	
g_f_oe_est	float	4	actual motor speed [rad/s]	No use
g_f_oe_est_lpf	float	4	speed after filter	
g_f_oe_ref	float	4	motor speed reference [rad/s]	
g_f_oe_vuvw	float	4	pick up speed1	
g_f_oe_vuvw_lpf	float	4	pick up speed1	
g_f_Offset_Cur_Iac	float	4	iac offset	
g_f_offset_cur_iu	float	4	iu offset	
g_f_offset_cur_iv	float	4	iv offset	
g_f_offset_cur_iw	float	4	iw offset	
g_f_offset_lpf_factor	float	4	current ad offset Low pass filter	
g_f_Offset_Vac	float	4	ac voltage offset cal	
g_f_Offset_Vac_lpf	float	4	ac voltage offset cal	
g_f_Offset_Vac_Sum	float	4	ac voltage offset cal	
g_f_Offset_Vac_Temp	float	4	ac voltage offset cal	
g_f_Offset_Vuv	float	4	Vuv voltage offset	
g_f_Offset_Vwu	float	4	Vwu voltage offset	
g_f_over_cur_level	float	4	current protection level	
g_f_over_vol_level	float	4	over voltage protection level	
g_f_pe	float	4	angle of motor[rad]	
g_f_pe_est_factor	float	4	pe estimation factor	

g_f_pfc_on_rpm_setting	float	4	setting pfc rpm	
g_f_rad_delta	float	4	delta rad	
g_f_rad_vuvw	float	4	pick up rad new	
g_f_rad_vuvw_offset	float	4	pick up rad offset	
g_f_rad_vuvw_old	float	4	pick up rad old	
g_f_ref_limit	float	4	max for modulation	
g_f_rev_vdc	float	4	voltage for modulation	
g_f_rpm_enhance	float	4	speed of enhance mode	
g_f_rpm_est	float	4	actual motor speed [rpm]	No use
g_f_rpm_oltovec	float	4	speed of vector mode switchover	
g_f_rpm_ref	float	4	motor speed reference [rpm]	
g_f_rpm_ref_request	float	4	motor speed reference of user input[rpm]	
g_f_rpm_slope	float	4	speed reference slope [rpm/ms]	
g_f_rpm_slope_request	float	4	acceleration of vector mode [rpm/s]	
g_f_under_vol_level	float	4	under voltage protection level	
g_f_va_ref	float	4	alfa-axis voltage	
g_f_vac_plus_temp	float [25]	100	pfc phase shift buf	
g_f_Vac_Temp_New	float	4	ac voltage check	
g_f_Vac_Temp_Old	float	4	ac voltage check	
g_f_vac_value	float	4	ac voltage	
g_f_vac_value_plus	float	4	ac voltage	
g_f_vb_ref	float	4	beta-axis voltage	
g_f_vd_ref	float	4	d-axis reference voltage	
g_f_vd_ref_i	float	4	d-axis I-voltage	
g_f_vd_ref_over	float	4	d-axis reference voltage	
g_f_vd_ref_p	float	4	d-axis P-voltage	No use
g_f_vdc_value	float	4	dc-bus voltage	
g_f_vdq_limit	float	4	voltage limit	
g_f_vec_vuvw	float	4	Pick up speed2	
g_f_vec_vuvw_lpf	float	4	Pick up speed2	
g_f_vq_ref	float	4	q-axis reference voltage	
g_f_vq_ref_i	float	4	q-axis I-voltage	
g_f_vq_ref_over	float	4	q-axis reference voltage	
g_f_vq_ref_p	float	4	q-axis P-voltage	No use
g_f_vref_pfc	float	4	pfc voltage reference	
g_f_vref_pfc_temp	float	4	pfc voltage reference temp	
g_f_vu_ref	float	4	u-axis voltage	

g_f_Vuv_Value	float	4	Vuv voltage	
g_f_vuvw_a	float	4	pick up voltage	
g_f_vuvw_b	float	4	pick up voltage	
g_f_vuvw_d	float	4	pick up voltage	
g_f_vuvw_q	float	4	pick up voltage	
g_f_vuvw_temp	float	4	Pick up speed2	
g_f_vv_ref	float	4	v-axis voltage	
g_f_vw_ref	float	4	w-axis voltage	
g_f_Vwu_Value	float	4	Vwu voltage	
g_sl_main_cnt	int32_t	4	Main loop count	
g_ss_AD0_offset	int16_t [8]	16	Offset value for S12AD ch0	
g_ss_oe_direction	int16_t	2	pick up direction	
g_uc_decimation	uint8_t	1	Decimation count	
g_uc_error_status	uint8_t	1	error status of mode system	
g_uc_ics_cnt	uint8_t	1	ics control	
g_uc_mode_system	uint8_t	1	actual mode of system	
g_ul_dtc_table	uint32_t [256]	1024	DTC table	
g_us_Clock	uint16_t	2	RX13T operation frequency	
g_us_cnt_boot	uint16_t	2	count for boot charge	
g_us_ControlCnt	uint16_t	2	decimation count	
g_us_deadtime	uint16_t	2	Deadtime setting : ns	No use
g_us_Deadtime_C	uint16_t	2	deadtime setting for motor	
g_us_error_flag_cur_loop	uint16_t	2	deadtime setting : ns	
g_us_error_flag_speed_loop	uint16_t	2	flag of error in current loop	
g_us_freq_current	uint16_t	2	current loop control frequency: Hz	
g_us_freq_speed	uint16_t	2	speed loop control frequency: Hz	
g_us_HalfCarrier_C1	uint16_t	2	Half carrier for motor	
g_us_HalfCarrier_Pfc	uint16_t	2	Half carrier for PFC	No use
g_us_Lib_Version	uint16_t	2	Library version	No use
g_us_mode_system_request	uint16_t	2	input of mode system	
g_us_Offset_Vac_Cnt	uint16_t	2	ac voltage offset cal count	
g_us_Offset_Vac_lpf_Cnt	uint16_t	2	ac voltage offset cal count	
g_us_pfc_on	uint16_t	2	pfc on/off	
g_us_run_mode	uint16_t	2	mode of driving	
g_us_run_mode_temp	uint16_t	2	mode of driving (temp)	
g_us_time_cnt_offset	uint16_t	2	ad adjust cnt	
g_us_time_setting_offset	uint16_t	2	ad adjust setting	
g_us_TopCarrier_C1	uint16_t	2	Top carrier for motor	

g_us_TopCarrier_Pfc	uint16_t	2	Top carrier for PFC	
g_us_Vac_Offset_Enable	uint16_t	2	ac voltage offset cal enable	
g_us_vac_plus	uint16_t	2	Plus ac voltage	
g_us_vac_plus_delay	uint16_t	2	pfc phase shift setting	
idDelay1	float	4	id delay	
iqDelay1	float	4	iq delay	
s_uc_DefAction	uint8_t (*[4][3])(uint8_t)	48	Set action mode	
s_uc_DefState	uint8_t [4][3]	12	Status	

3.4 Macro definition

The following table lists the macro definitions used in this control program. Macro definitions (macro definitions that are not used in this software) are also included in the macro definitions that are prepared for extension so that users can easily incorporate some macro definitions. Please refer to Notes in the table below to define unused macros.

Table 3-13 : Macro Definition Lists

Overview	Definition Name	Definition Contents	Detail	Notes
Inverter settings	CARRIER_FREQ	16000	Motor carrier frequency	16KHz
	DEADTIME	2000	Motor dead time	2000ns
	TIMER_FREQ	1000	Motor speed loop interval	1kHz
	DECIMATION	1	Decimate number	
	ICS_DECI	1	ICS decimate number	
	CARRIER_PFC_FREQ	50000	PFC carrier frequency	50KHz
	ERROR_IAC_OC_LEVEL	5.0f	AC over current limit level	No use
	ERROR_VDC_OV_LEVEL	28.0f	DC over voltage limit level	No use
	ERROR_VDC_UV_LEVEL	17.0f	DC under voltage limit level	No use
	CPU_VOLTAGE	5.0f	RX13T operation voltage	
	AD_MAX	4096	AD maximum value(12bit)	
	PGA_GAIN	5.0f	PGA gain(magnification)	
	SHUNT_R	0.25f	Shunt resistance value	Ohm
	OFFSET_IU	AD_MAX * PGA_GAIN / 10	Current offset	
	RANGE_IU	CPU_VOLTAGE*10 /(SHUNT_R*9*PGA_GAIN)	Current range	Ohm
	SHUNT_R_IAC	0.25f	Shunt resistance value for AC current	
	OFFSET_IAC	AD_MAX / 2	Offset value for AC current	
	OPA_GAIN_IAC	5.0f	PGA gain value for AC current(magnification)	
	RANGE_IAC	CPU_VOLTAGE / (OPA_GAIN_IAC *	Range value for AC current	

		SHUNT_R_IAC)		
	RANGE_VDC	CPU_VOLTAGE*45 3.3f/3.3f	Range value for DC voltage	
	OFFSET_V_PHASE	AD_MAX / 2	Offset for V-phase	
	RANGE_V_PHASE	CPU_VOLTAGE * 300	Range for V-phase	Gain = 300
	OFFSET_VAC	AD_MAX / 2	Offset for AC voltage	
	RANGE_VAC	CPU_VOLTAGE * 300	Range for AC voltage	Gain = 300
System Setting(System mode)	SEQ_MODE_STOP	(0x00)	Stop sequence mode	
	SEQ_MODE_RUN	(0x01)	Run sequence mode	
	SEQ_MODE_ERROR	(0x02)	Error sequence mode	
	SEQ_SIZE_STATE	(3)	Sequence mode number	Number of state
System Setting(Event)	SEQ_EVENT_STOP	(0x00)	Stop sequence event	
	SEQ_EVENT_RUN	(0x01)	Run sequence event	
	SEQ_EVENT_ERROR	(0x02)	Error sequence event	
	SEQ_EVENT_RESET	(0x03)	Reset sequence event	
	SEQ_SIZE_EVENT	(4)	Sequence event number	Number of event
System Setting(Error status)	ERROR_OVER_CURRENT	(0x0001)	Over current error	
	ERROR_OVER_VOLTAGE	(0x0002)	Over voltage error	
	ERROR_UNDER_VOLTAGE	(0x0004)	Under voltage error	
	ERROR_SEQ_UNKNOWN	(0xff)	Undefined error	
System Setting(Mode)	MODE_BOOT	(0U)	Boot mode	Boot mode
	MODE_RUN_OPENLOOP	(1U)	Openloop mode	Open loop mode
	MODE_RUN_FOC	(2U)	FOC mode	Vector control mode
Parameter Setting(User setting)	MOTOR_RA	(117)	Motor resistance	ohm
	MOTOR_LD	(0.2)	d-axis self-inductance	H
	MOTOR_LQ	(0.36)	q-axis self-inductance	H
	MOTOR_KE	(0.465)	Back electromotive force constant	Vs/rad
	MOTOR_PP	(4)	Number of poles	
	IDQ_LIMIT	(0.6)	id/iq limit value	A
	ID_OPENLOOP	(0.55)	id value at open loop	A
	ID_ENHANCE	(0.1)	id enhance value	A
	SPEED_SLOPE	(0.005)	Speed slope value	rpm/ms
	KP_SPEED	(0.01)	Proportional for speed value	

	KI_SPEED	(0.0003)	Integral for speed value	
	KP_ID	(160)	Proportional for id	
	KI_ID	(0.08)	Integral for id	
	KP_IQ	(200)	Proportional for iq	
	KI_IQ	(0.08)	Integral for iq	
	K_ANGLE_EST	(0.7)	Angle estimation coefficient value	
	K_EMF_EST	(20)	Electromotive force estimation coefficient value	
	K_SPEED_LPF	(0.03)	Low pass filter coefficient value for speed	
	SPEED_OL_VEC	(65)	Switching speed to VEC or open loop	
	SPEED_ENHANCE	(85)	Switching speed enhance value	
Coordinate conversion fixed value	MATH_ZERO	(0.0f)	0 float value	
	MATH_ONE	(1.0f)	1 float value	No use
	MATH_TWO	(2.0f)	2 float value	No use
	MATH_PI	(3.1415926535f)	π	
	MATH_TWOPI	(2.0f*MATH_PI)	2 * π	
	MATH_PI_3	(MATH_PI/3.0f)	π / 3	
	MATH_TWOPI_3	(MATH_TWOPI/3.0f)	2 * π / 3	No use
	MATH_TWOPI2_3	(MATH_TWOPI*2/3.0f)	4 * π / 3	
	MATH_SQRT_3d2	(1.224745f)	sqrt(3/2)	
	MATH_SQRT_2d3	(0.816497f)	sqrt(2/3)	
	MATH_SQRT_3_2	(0.866025f)	sqrt(3)/2	
	MATH_1d2	(0.5f)	1/2	
	MATH_SQRT_2_2	(0.707107f)	sqrt(2)/2	
	MATH_SQRT_2	(1.414214f)	sqrt(2)	
	MATH_RPM2RAD	(1.0f/60.0f*MATH_TWOPI)	1 / 60 * (2 * π)	rpm to radian, No use
	MATH_RAD2RPM	(1.0f/MATH_TWOPI*60.0f)	1 / (2 * π) * 60	radian to rpm, No use
	MATH_DEG2RAD	(MATH_TWOPI/360.0f)	(2 * π) / 360	Degree to radian, No use
	MATH_SQRT_3	(1.73205081f)	sqrt(3)	
	MATH_1div_SQRT3	(1/MATH_SQRT_3)	1 / sqrt(3)	
CMT0 register set value (1ms speed command cycle)	r_mtr_interval_CMT0_set_int_level(level)	(ICU.IPR[4].BIT.IPR=(level)&15)	Set interrupt level for CMT0	
	r_mtr_interval_CMT0_start_timer()	(CMT.CMSTR0.BIT.STR0=1)	Start operation for CMT0	
	r_mtr_interval_CMT0_stop_timer()	(CMT.CMSTR0.BIT.STR0=0)	Stop operation for CMT0	No use

	r_mtr_interval_CMT0_start_int()	(ICU.IER[3].BIT.IEN4=1,CMT0.CMCR.BIT.CMIE=1)	Enable interrupt for CMT0	
	r_mtr_interval_CMT0_stop_int()	(ICU.IER[3].BIT.IEN4=0,CMT0.CMCR.BIT.CMIE=0)	Disable interrupt for CMT0	No use
	r_mtr_interval_CMT0_clear_flag()	(ICU.IR[28].BIT.IR=0)	Clear interrupt flag for CMT0	
CMT1 register set value (Not used)	r_mtr_interval_CMT1_set_int_level(level)	(ICU.IPR[5].BIT.IPR=(level)&15)	Set interrupt level for CMT1	No use
	r_mtr_interval_CMT1_start_timer()	(CMT.CMSTR0.BIT.STR1=1)	Start operation for CMT1	No use
	r_mtr_interval_CMT1_stop_timer()	(CMT.CMSTR0.BIT.STR1=0)	Stop operation for CMT1	No use
	r_mtr_interval_CMT1_start_int()	(ICU.IER[3].BIT.IEN5=1,CMT1.CMCR.BIT.CMIE=1)	Enable interrupt for CMT1	No use
	r_mtr_interval_CMT1_stop_int()	(ICU.IER[3].BIT.IEN5=0,CMT1.CMCR.BIT.CMIE=0)	Disable interrupt for CMT1	No use
	r_mtr_interval_CMT1_clear_flag()	(ICU.IR[29].BIT.IR=0)	Clear interrupt flag for CMT1	No use
MTU3 (Ch3,4) register setting value (complementary PWM output)	r_mtr_inverter_MTU34_set_int_level(level)	(ICU.IPR[138].BIT.IPR=((level)&15))	Set interrupt level for MTU3 ch3,4	
	r_mtr_inverter_MTU34_start_pwm()	(MTU.TOERA.BYTE=0xFF)	Enable PWM output for MTU3 ch3,4	
	r_mtr_inverter_MTU34_stop_pwm()	(MTU.TOERA.BYTE=0xC0)	Disable PWM output for MTU3 ch3,4	
	r_mtr_inverter_MTU34_start_int()	(ICU.IER[0x11].BIT.IEN2=1,MTU4.TIER.BIT.TCIEV=1)	Enable interrupt for MTU3 ch3,4 ch3,4	
	r_mtr_inverter_MTU34_stop_int()	(ICU.IER[0x11].BIT.IEN2=0,MTU4.TIER.BIT.TCIEV=0)	Disable interrupt for MTU3 ch3,4 ch3,4	
	r_mtr_inverter_MTU34_start_timer()	(MTU.TSTRA.BYTE |= (0xC0))	Start operation for MTU3 ch3,4 ch3,4	
	r_mtr_inverter_MTU34_clear_int_flag()	(ICU.IR[138].BIT.IR=0)	Clear interrupt flag for MTU3 ch3,4	
MTU3 (Ch0) register setting value (PWM output for PFC)	r_mtr_pwm_MTU0_start_timer()	(MTU.TSTRA.BIT.CST0 |= (0x01))	Start operation for MTU3 ch0	
	r_mtr_pwm_MTU0_stop_timer()	(MTU.TSTRA.BIT.CST0 &= (~0x01))	Stop operation for MTU3 ch0	No use
	r_mtr_pfc_MTU0_start_pwm()	(PORTB.PMR.BIT.B3 = 1)	Select MTU3 ch0 function	
	r_mtr_pfc_MTU0_stop_pwm()	(PORTB.PMR.BIT.B3 = 0, PORTB.PODR.BIT.B3 = 0, PORTB.PDR.BIT.B3 = 1)	Select I/O port function	
POE (Port output enable) register setting (not used)	r_mtr_inverter_POE0_get_status(void)	(POE.ICSR1.BIT.POE0F)	get POE0# status	No use
	r_mtr_inverter_POE0_clear_status(void)	(POE.ICSR1.WORD, POE.ICSR1.BIT.POE0F=0)	Clear POE0# status	No use
MTU3 (Ch1) phase count mode	r_mtr_ABZ_MTU1_P33_P32_PA5	(0x00)	Select MTU3 ch1 function	No use

register setting (Not used)				
S12AD register setting	r_mtr_AD000	(0)	Specify S12AD ch0	
	r_mtr_AD001	(1)	Specify S12AD ch1	
	r_mtr_AD002	(2)	Specify S12AD ch2	
	r_mtr_AD003	(3)	Specify S12AD ch3	
	r_mtr_AD004	(4)	Specify S12AD ch4	
	r_mtr_AD005	(5)	Specify S12AD ch5	
	r_mtr_AD006	(6)	Specify S12AD ch6	
	r_mtr_AD007	(7)	Specify S12AD ch7	
	r_mtr_AD_SINGLE_SCAN	(0)	Select single scan mode	No use
	r_mtr_AD_GROUP_SCAN	(1)	Select group scan mode	No use
	r_mtr_AD_CONTINOUS_SCAN	(2)	Select continuous scan mode	No use
	r_mtr_SEL_AD000	(0x00001)	Select S12AD ch0 function	
	r_mtr_SEL_AD001	(0x00002)	Select S12AD ch1 function	
	r_mtr_SEL_AD002	(0x00004)	Select S12AD ch2 function	
	r_mtr_SEL_AD003	(0x00008)	Select S12AD ch3 function	
	r_mtr_SEL_AD004	(0x00010)	Select S12AD ch4 function	
	r_mtr_SEL_AD005	(0x00020)	Select S12AD ch5 function	
	r_mtr_SEL_AD006	(0x00040)	Select S12AD ch6 function	
	r_mtr_SEL_AD007	(0x00080)	Select S12AD ch7 function	
	r_mtr_ad_AD0_start()	(S12AD.ADCSR.BIT.ADST=1)	Start operation for S12AD	No use
	r_mtr_ad_AD0_set_int_level(level)	(ICU.IPR[102].BIT.IPR=((level)&15))	Set interrupt level for S12AD	
	r_mtr_ad_AD0_start_int()	(ICU.IER[0x0C].BIT.IEN6=1,S12AD.ADCSR.BIT.ADIE = 1)	Enable interrupt for S12AD	
	r_mtr_ad_AD0_stop_int()	(ICU.IER[0x0C].BIT.IEN6=0,S12AD.ADCSR.BIT.ADIE = 0)	Disable interrupt for S12AD	No use
	r_mtr_ad_AD0_get_status()	(S12AD.ADCSR.BIT.ADST)	Get status for S12AD	
	r_mtr_ad_AD0_get_ad000()	(((float)((int16_t)S12AD.ADDR0 - g_ss_r_mtr_AD0_offset[0]))*g_f_r_mtr_AD0_range[0]/4096.0f)	Get S12AD ch0 machining value	
	r_mtr_ad_AD0_get_ad001()	(((float)((int16_t)S12AD.ADDR1 - g_ss_r_mtr_AD0_offset[1]))*g_f_r_mtr_AD0_range[1]/4096.0f)	Get S12AD ch1 machining value	
	r_mtr_ad_AD0_get_ad002()	(((float)((int16_t)S12AD.ADDR2 -	Get S12AD ch2 machining value	

		g_ss_r_mtr_AD0_offset[2]))*g_f_r_mtr_AD0_range[2]/4096.0f)		
	r_mtr_ad_AD0_get_ad003()	(((float)((int16_t)S12AD.ADDR3 - g_ss_r_mtr_AD0_offset[3]))*g_f_r_mtr_AD0_range[3]/4096.0f)	Get S12AD ch3 machining value	
	r_mtr_ad_AD0_get_ad004()	(((float)((int16_t)S12AD.ADDR4 - g_ss_r_mtr_AD0_offset[4]))*g_f_r_mtr_AD0_range[4]/4096.0f)	Get S12AD ch4 machining value	
	r_mtr_ad_AD0_get_ad005()	(((float)((int16_t)S12AD.ADDR5 - g_ss_r_mtr_AD0_offset[5]))*g_f_r_mtr_AD0_range[5]/4096.0f)	Get S12AD ch5 machining value	
	r_mtr_ad_AD0_get_ad006()	(((float)((int16_t)S12AD.ADDR6 - g_ss_r_mtr_AD0_offset[6]))*g_f_r_mtr_AD0_range[6]/4096.0f)	Get S12AD ch6 machining value	
	r_mtr_ad_AD0_get_ad007()	(((float)((int16_t)S12AD.ADDR7 - g_ss_r_mtr_AD0_offset[7]))*g_f_r_mtr_AD0_range[7]/4096.0f)	Get S12AD ch7 machining value	
	r_mtr_ad_AD0_get_ad000_raw()	S12AD.ADDR0	Get S12AD ch0 value	No use
	r_mtr_ad_AD0_get_ad001_raw()	S12AD.ADDR1	Get S12AD ch1 value	No use
	r_mtr_ad_AD0_get_ad002_raw()	S12AD.ADDR2	Get S12AD ch2 value	No use
	r_mtr_ad_AD0_get_ad003_raw()	S12AD.ADDR3	Get S12AD ch3 value	No use
	r_mtr_ad_AD0_get_ad004_raw()	S12AD.ADDR4	Get S12AD ch4 value	No use
	r_mtr_ad_AD0_get_ad005_raw()	S12AD.ADDR5	Get S12AD ch5 value	No use
	r_mtr_ad_AD0_get_ad006_raw()	S12AD.ADDR6	Get S12AD ch6 value	No use
	r_mtr_ad_AD0_get_ad007_raw()	S12AD.ADDR7	Get S12AD ch7 value	No use
IWDT register setting	r_mtr_IWDT_refresh()	IWDT.IWDTRR=0,IWDT.IWDTRR=0xFF	Refresh for IWDT	No use
	r_mtr_IWDT_underflow_status()	IWDT.IWDTSR.BIT.UNDFF	Get underflow status for IWDT	No use
	r_mtr_IWDT_underflow_clear()	IWDT.IWDTSR.BIT.UNDFF=0	Set underflow status for IWDT	No use

3.5 Control flow (flowchart)

3.5.1 Main processing

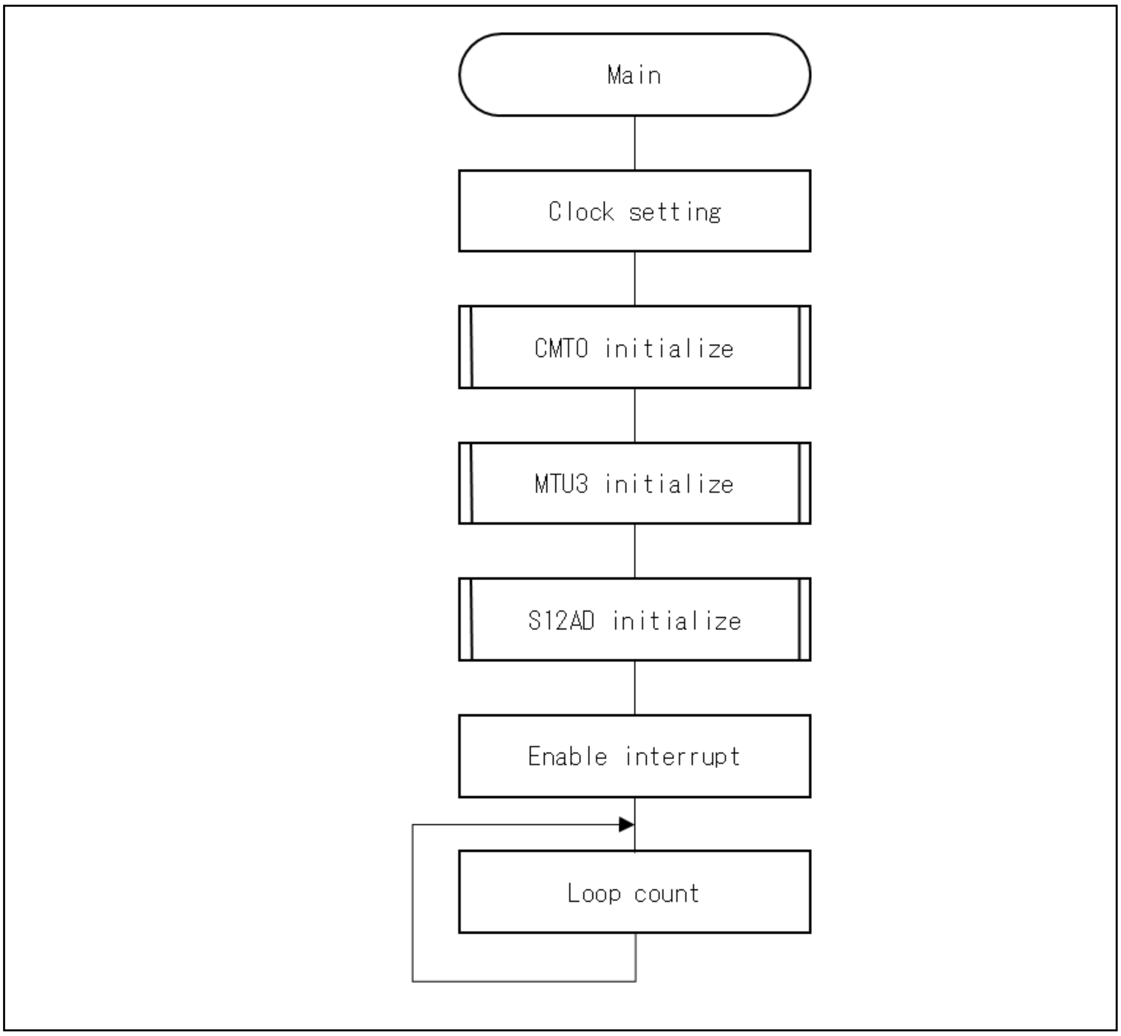


Figure 3-8 : Main processing flowchart

3.5.2 Carrier period interrupt processing

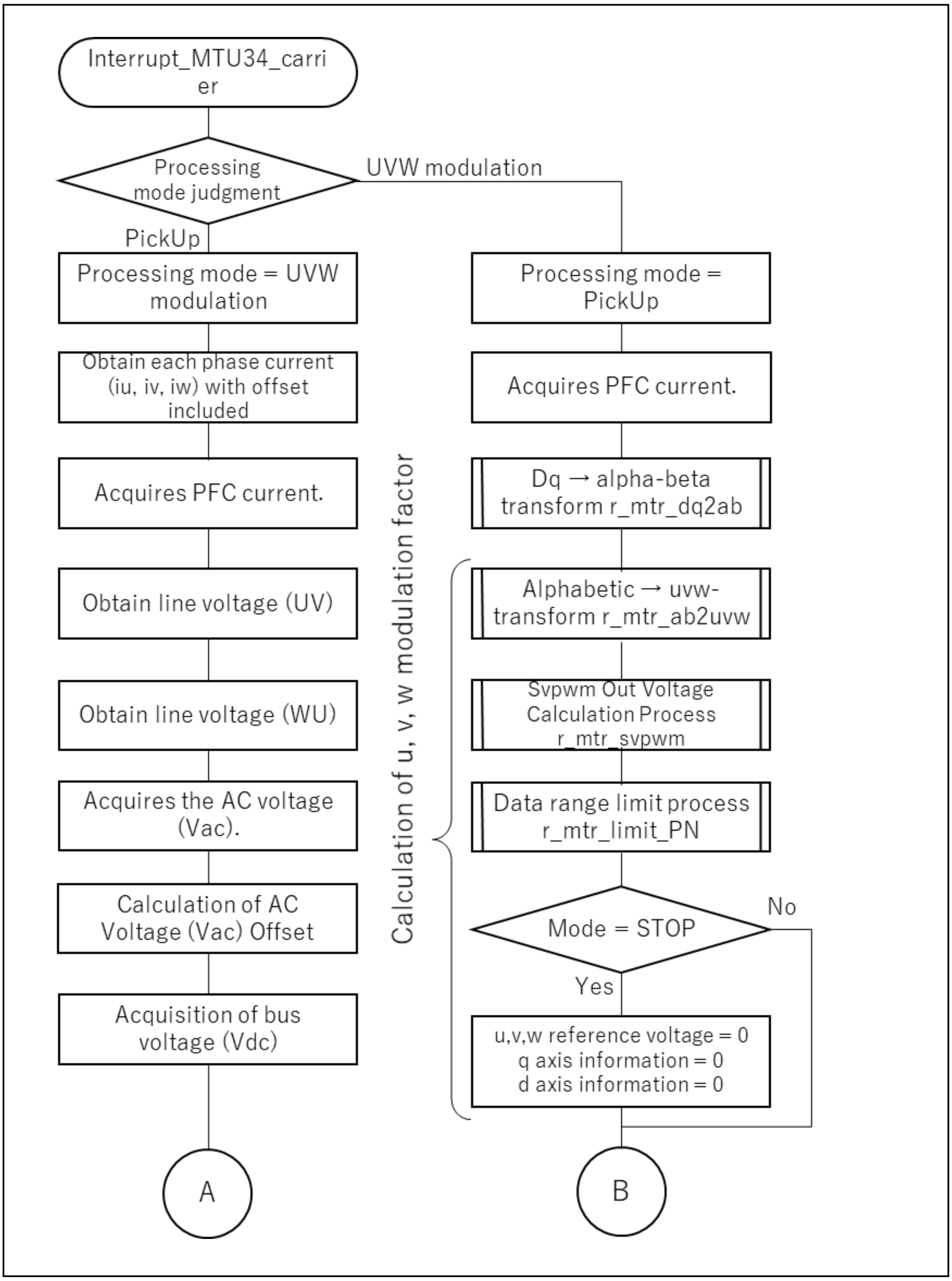


Figure 3-9 : Carrier cycle interrupt processing flowchart 1 of 5

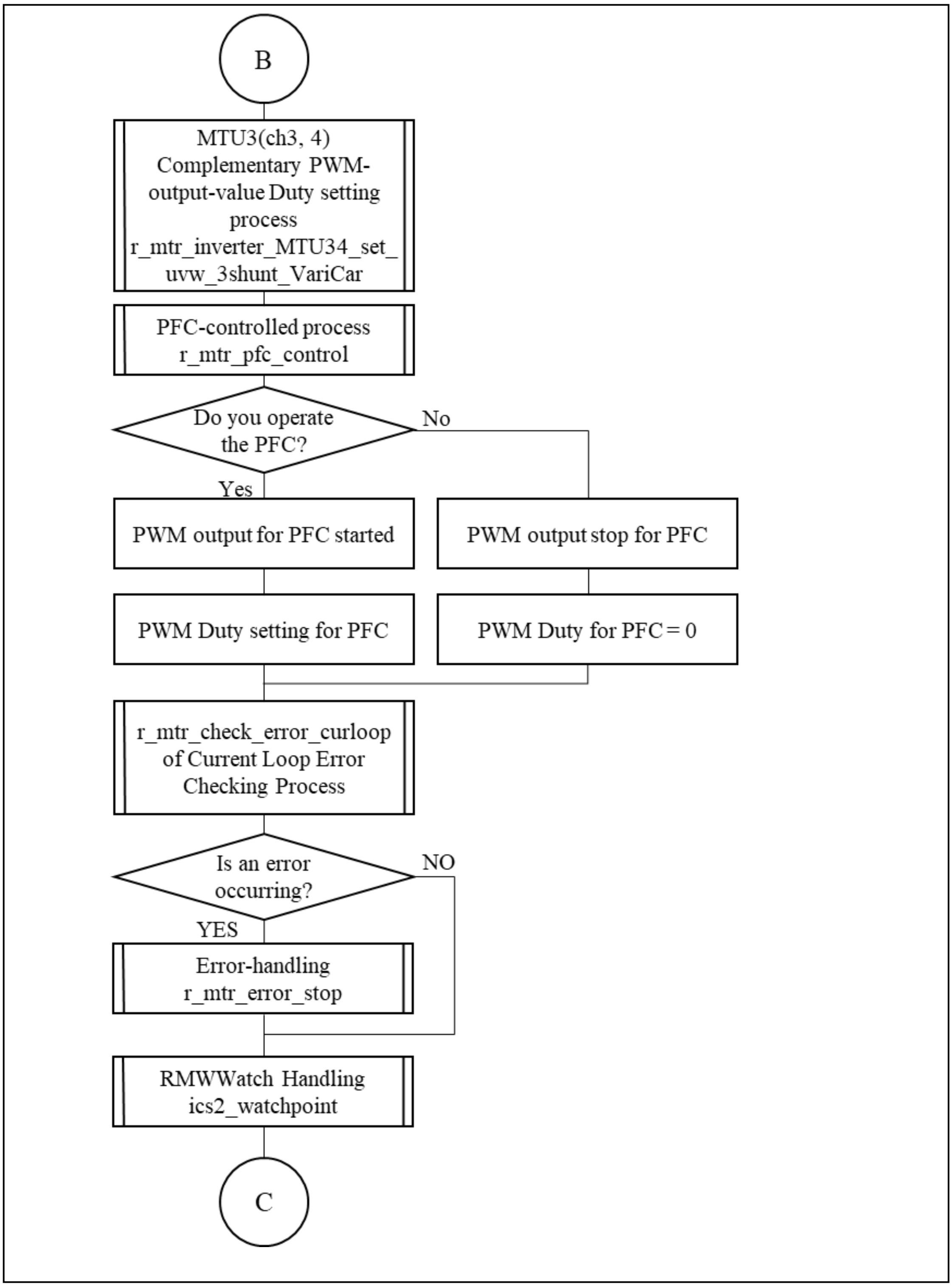


Figure 3-10 : Carrier cycle interrupt processing flowchart 2 of 5

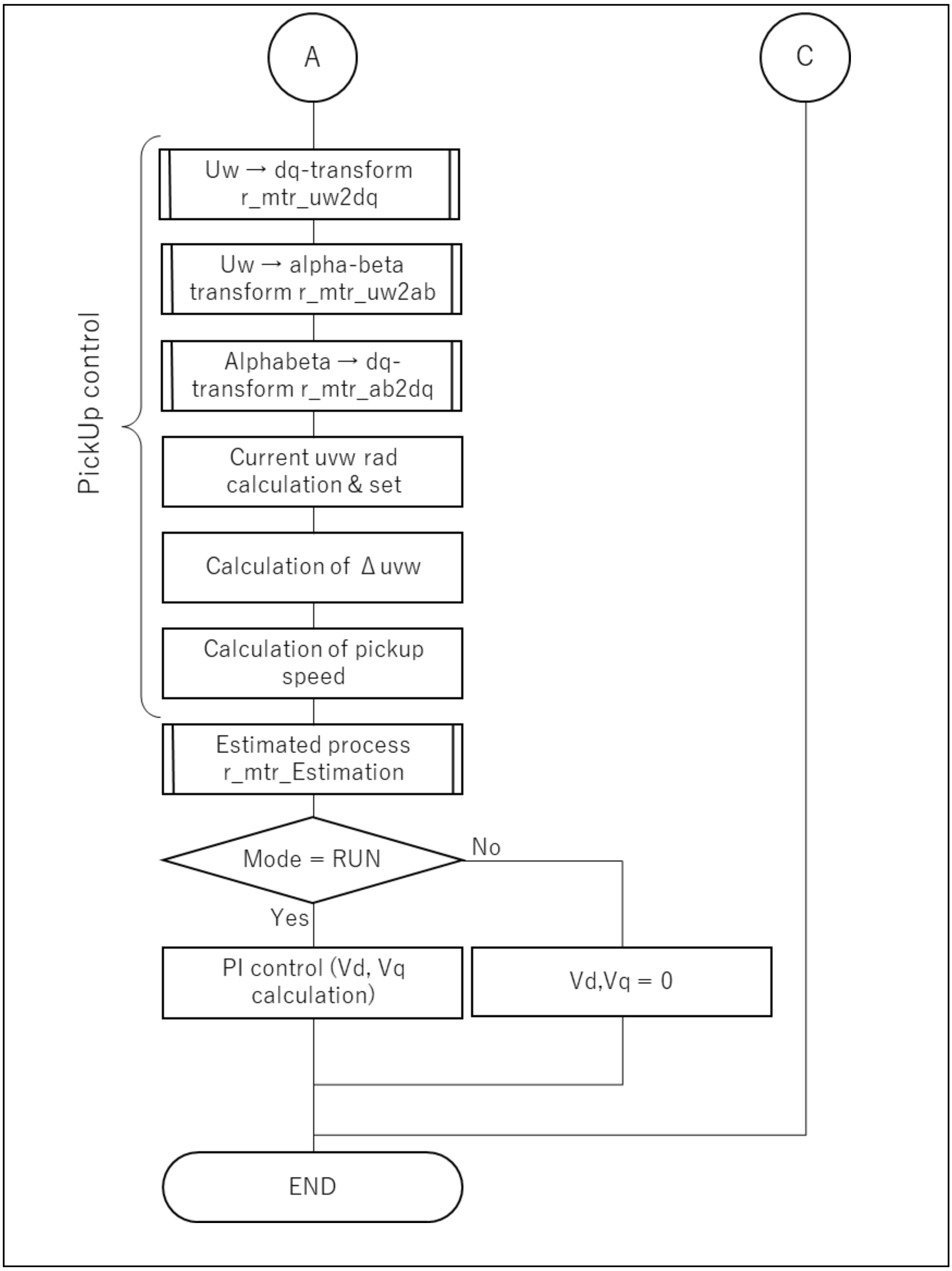


Figure 3-11 : Carrier cycle interrupt processing flowchart 3 of 5

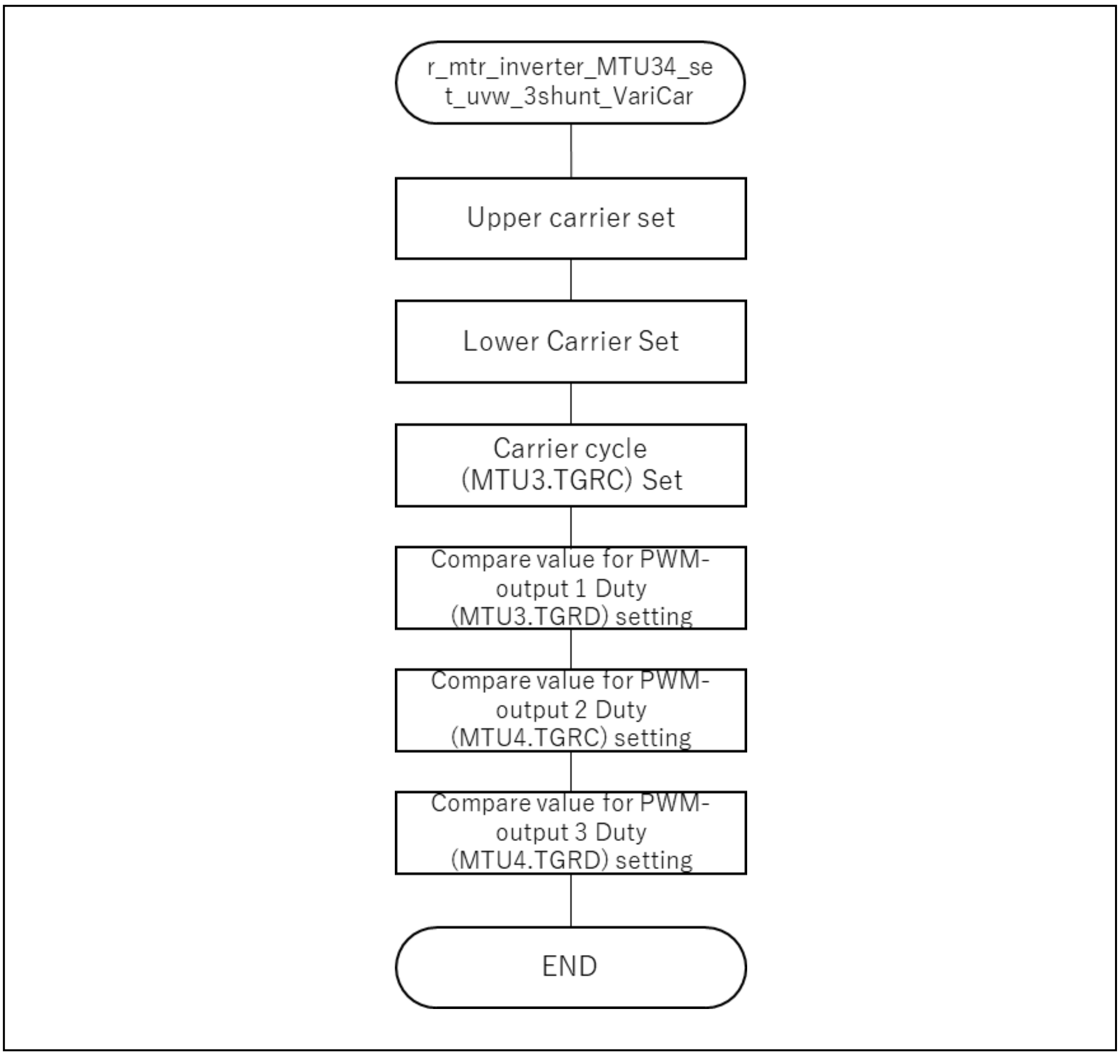


Figure 3-12 : Carrier cycle interrupt processing flowchart 4 of 5

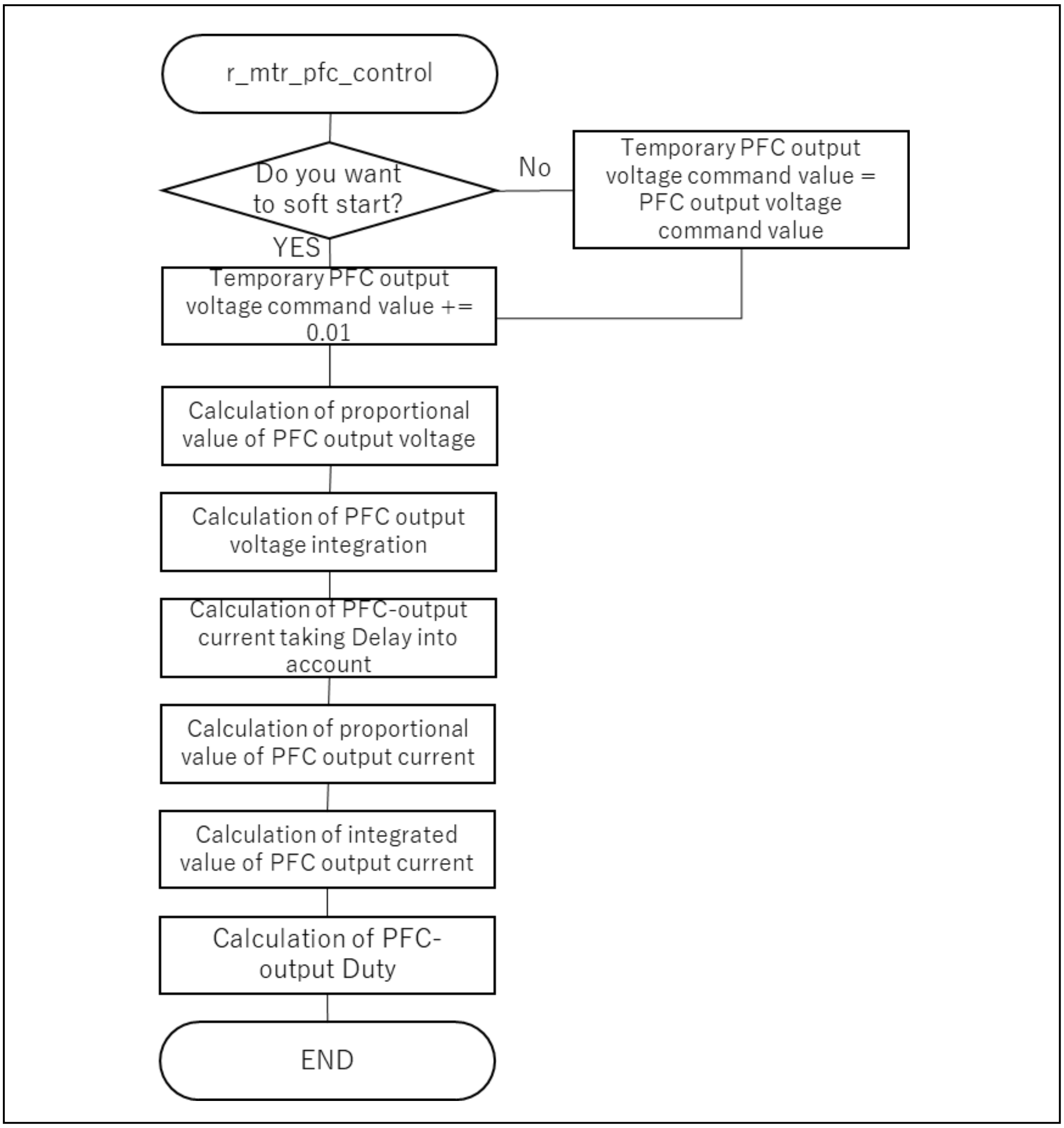


Figure 3-13 : Carrier cycle interrupt processing flowchart 5 of 5

3.5.3 1 [ms] interrupt processing

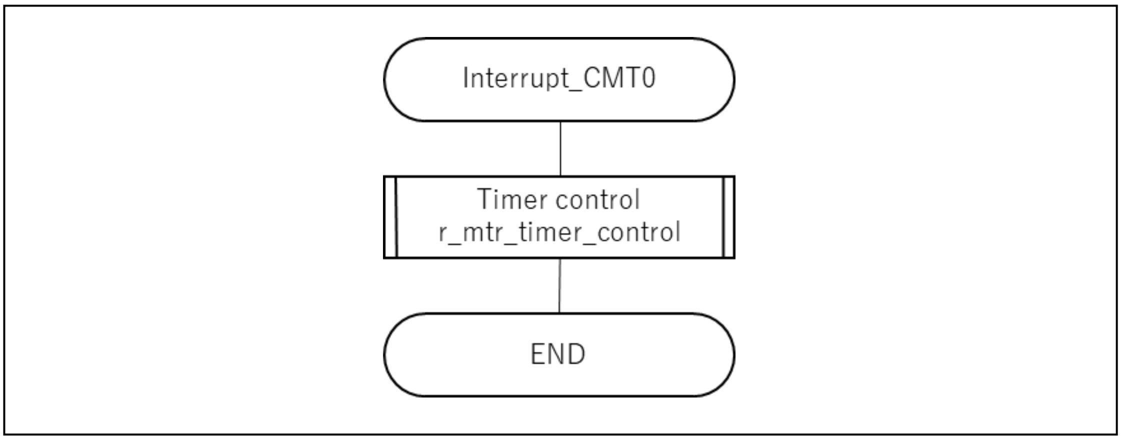


Figure 3-14 : 1 [ms] interrupt processing flowchart 1 of 6

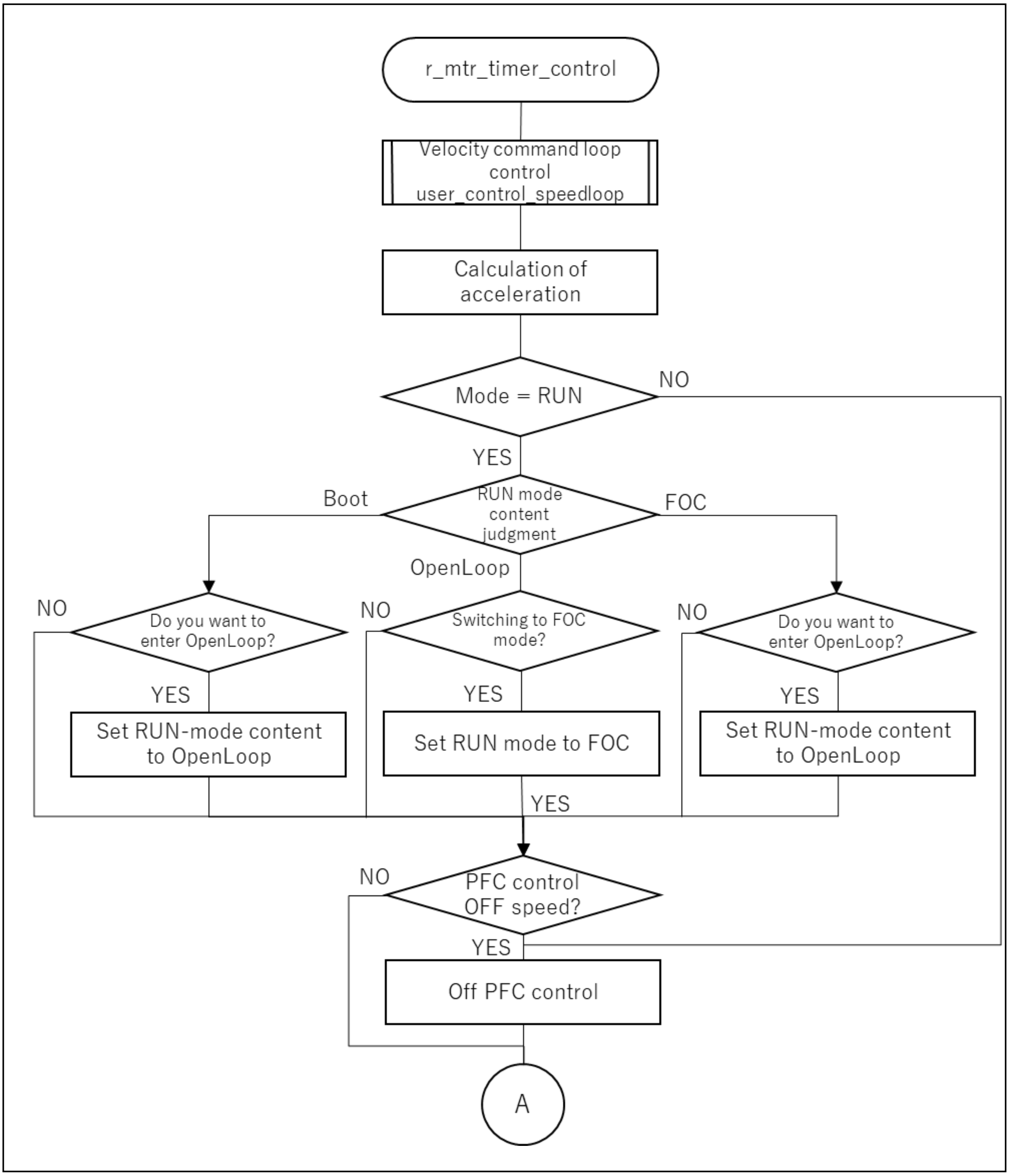


Figure 3-15 : 1 [ms] interrupt processing flowchart 2 of 6

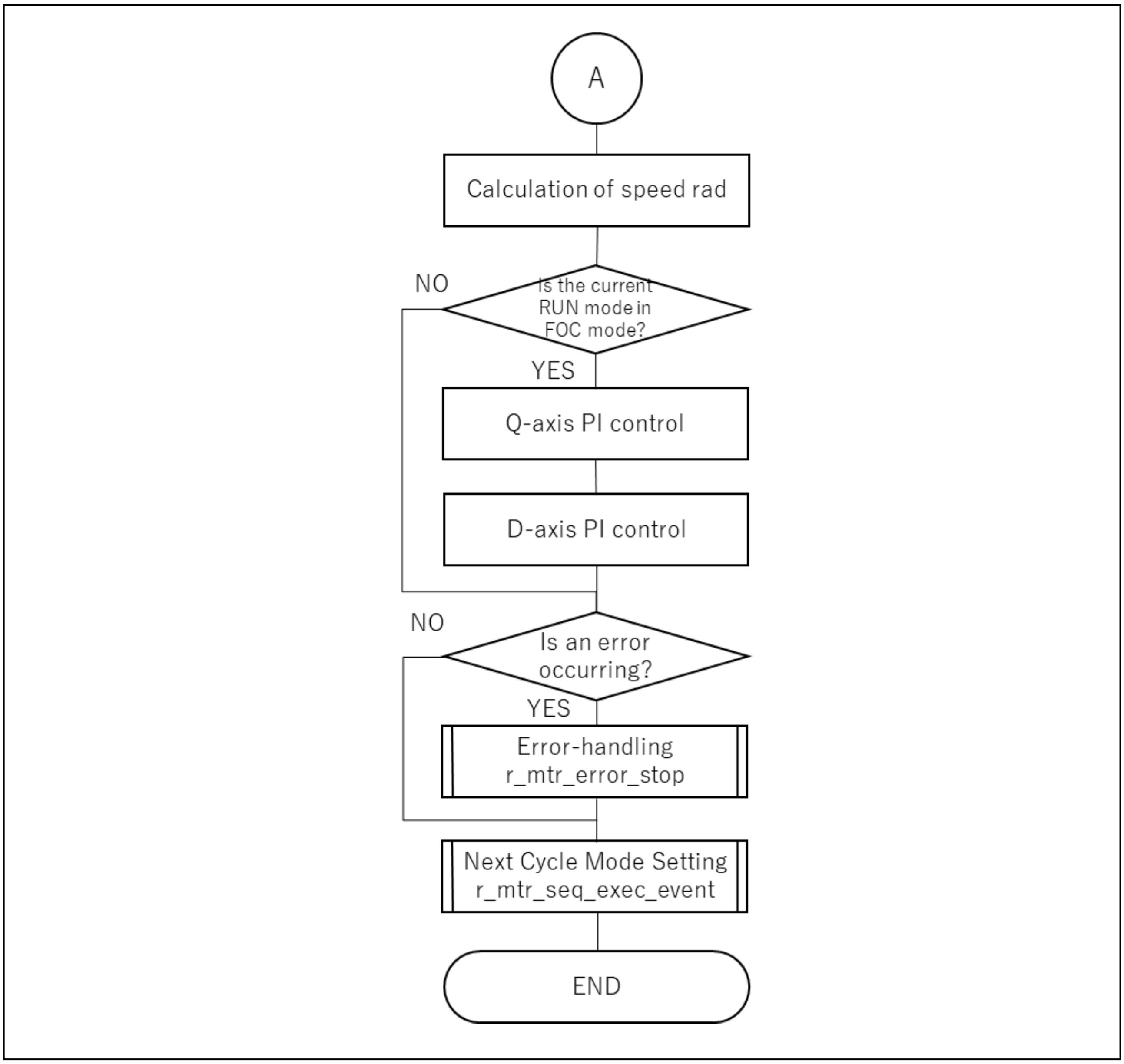


Figure 3-16 : 1 [ms] interrupt processing flowchart 3 of 6

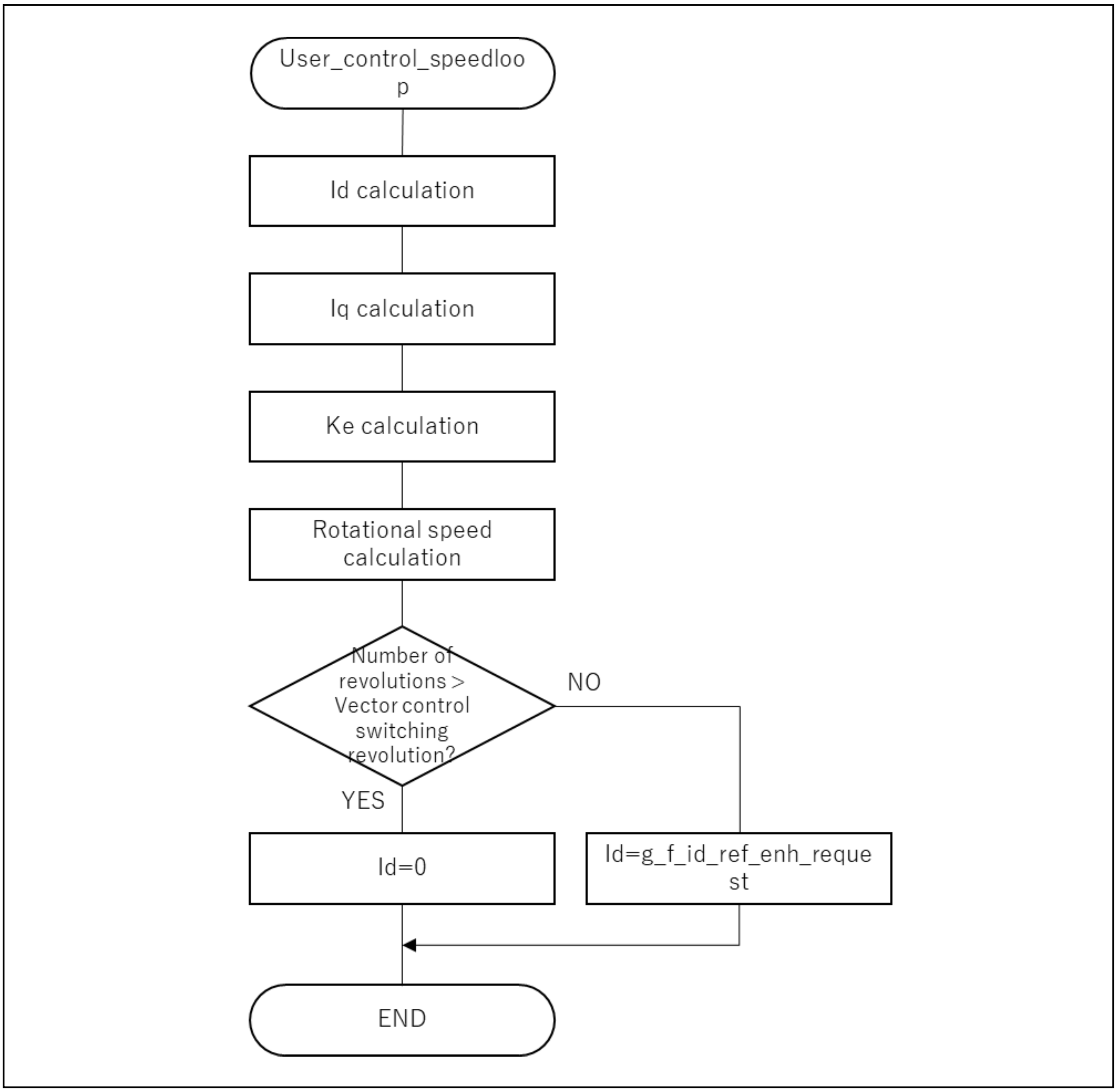


Figure 3-17 : 1 [ms] interrupt processing flowchart 4 of 6

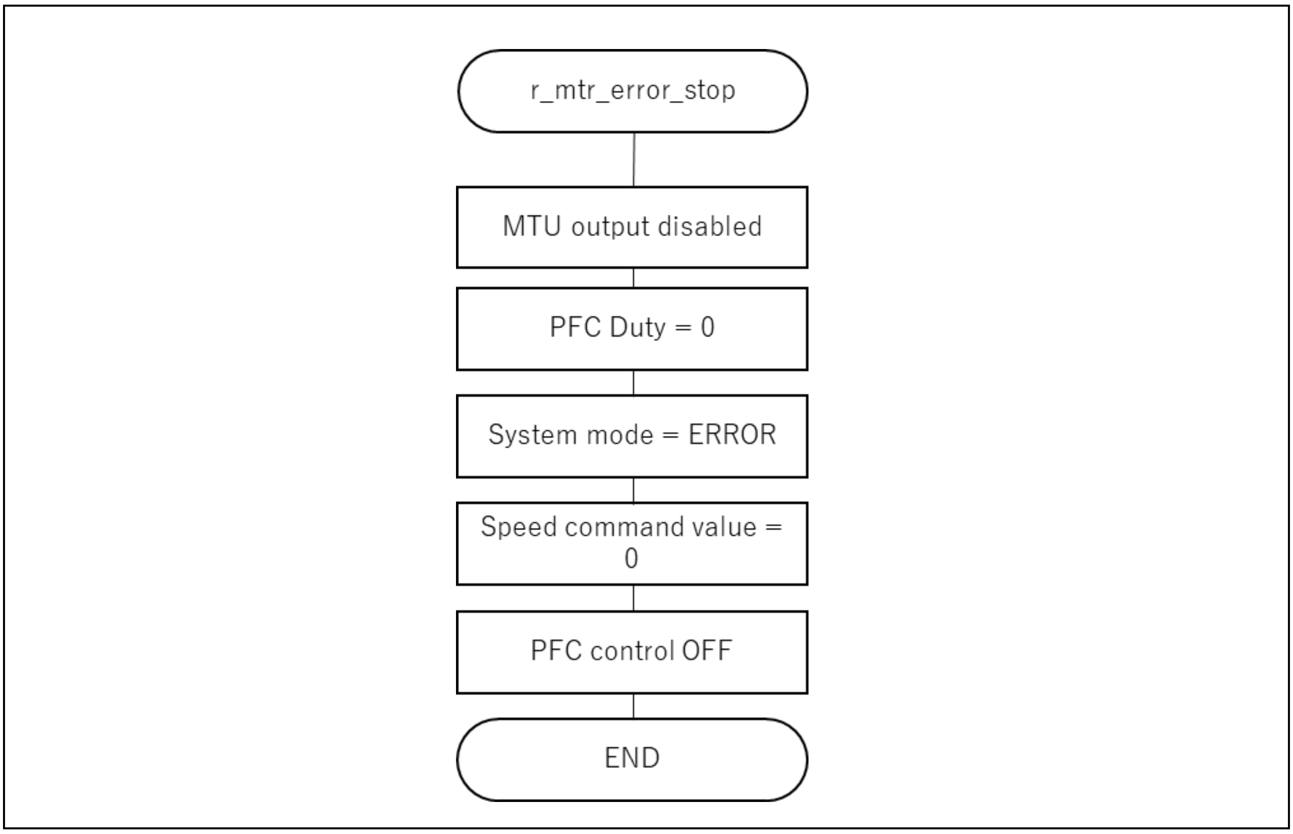


Figure 3-18 : 1 [ms] interrupt processing flowchart 5 of 6

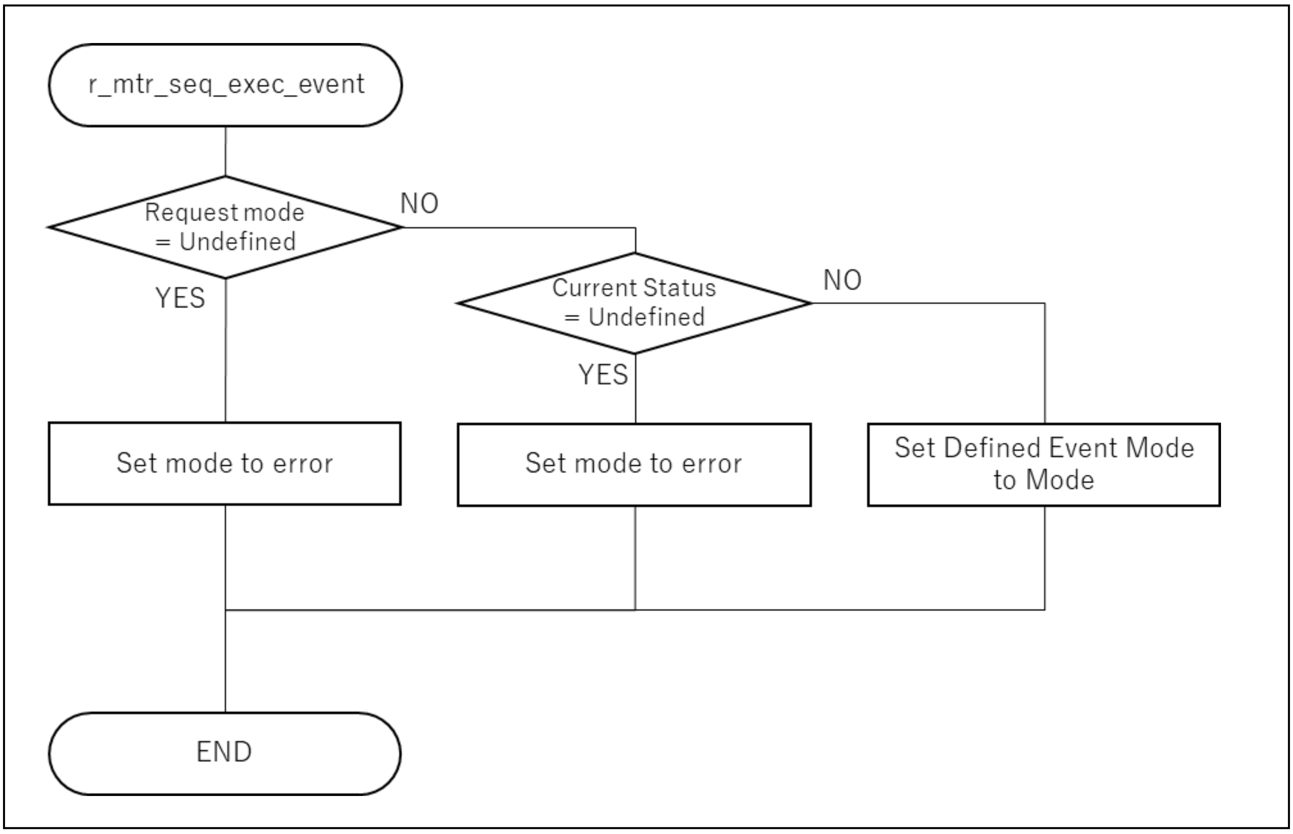


Figure 3-19 : 1 [ms] interrupt processing flowchart 6 of 6

3.5.4 Math library processing

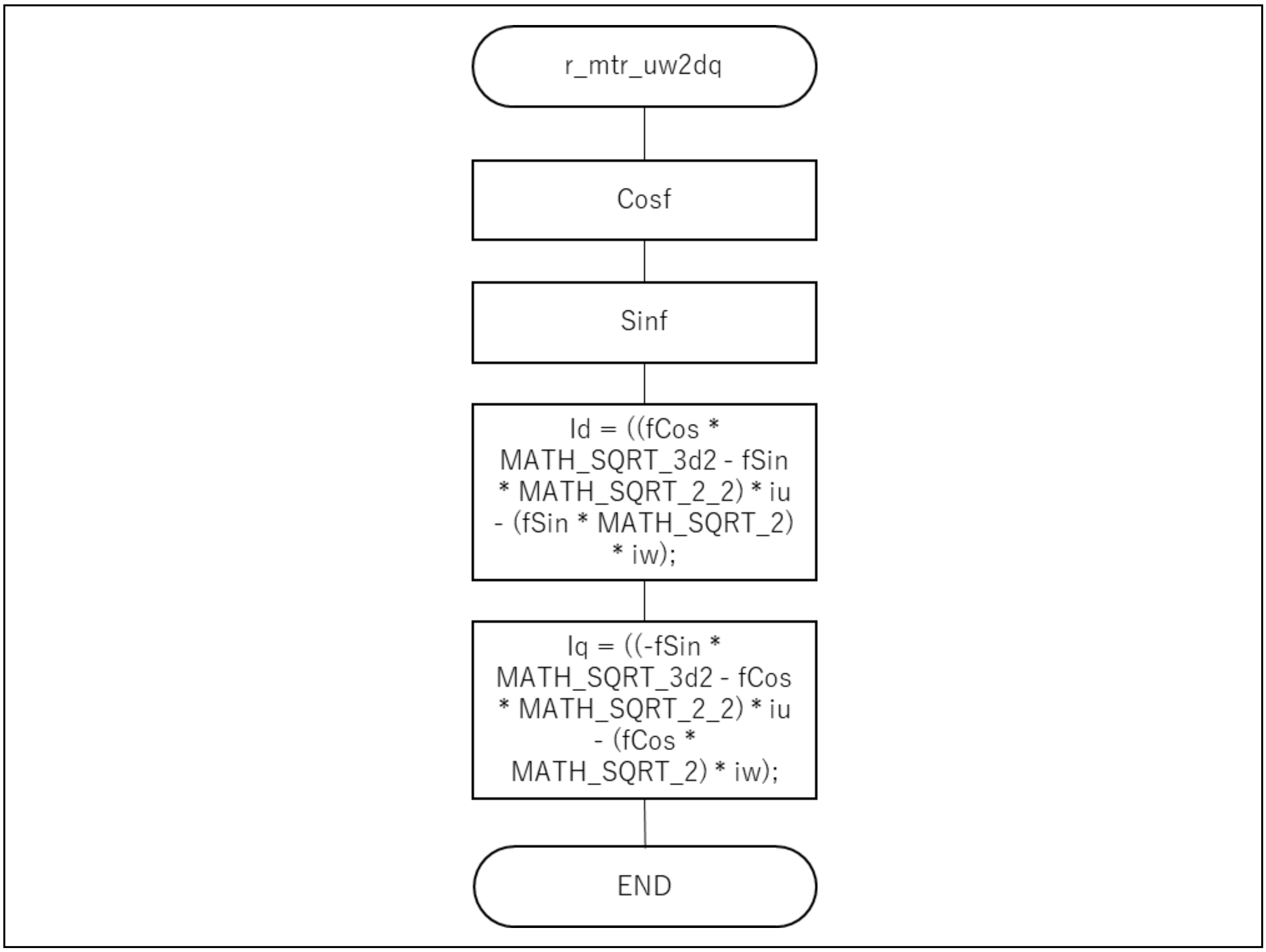


Figure 3-20 : uw → dq conversion processing flowchart

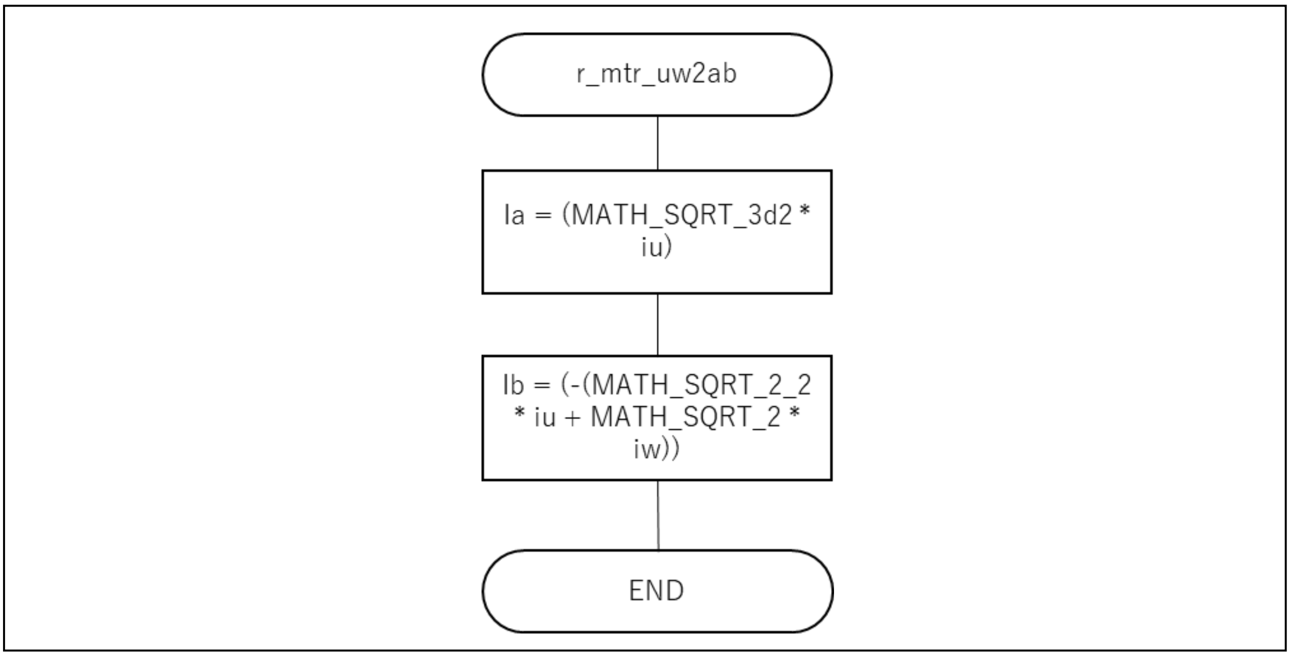


Figure 3-21 : uw → Alpha-Beta conversion processing flowchart

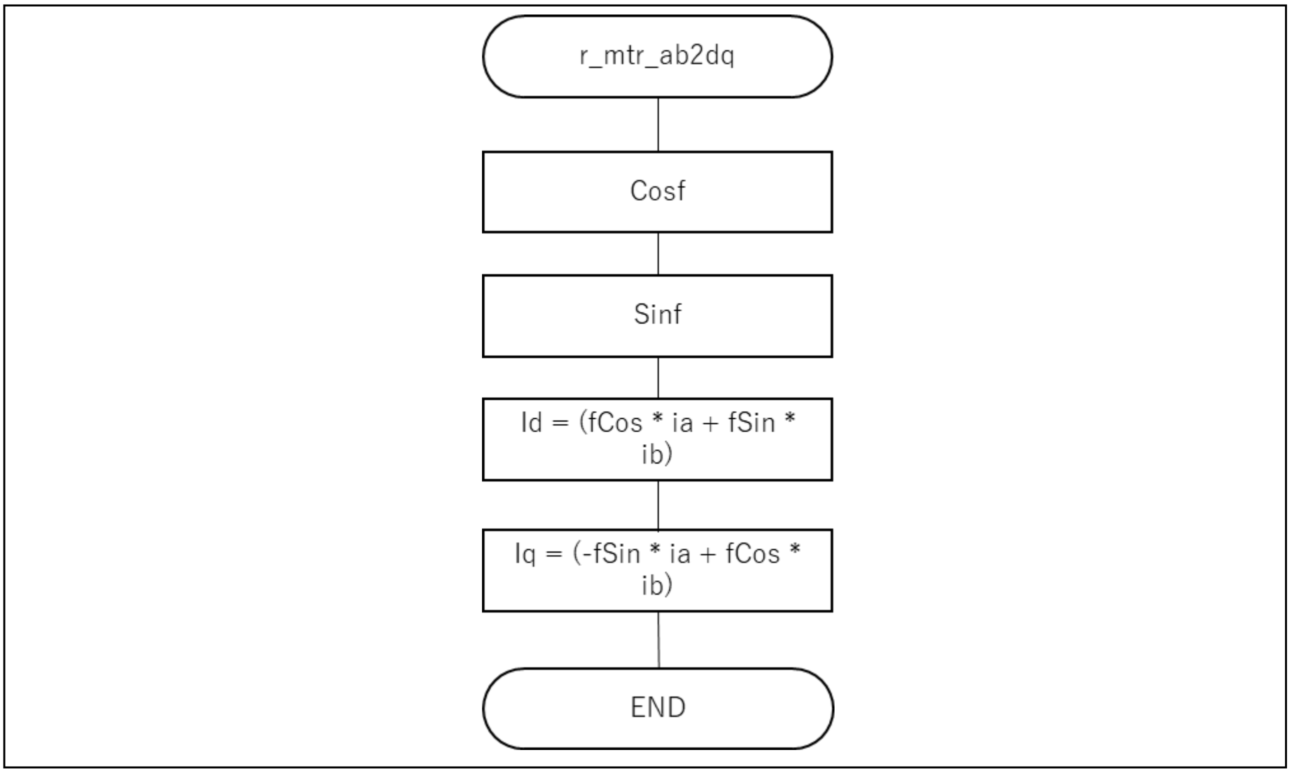


Figure 3-22 : Alpha-Beta → dq conversion processing flowchart

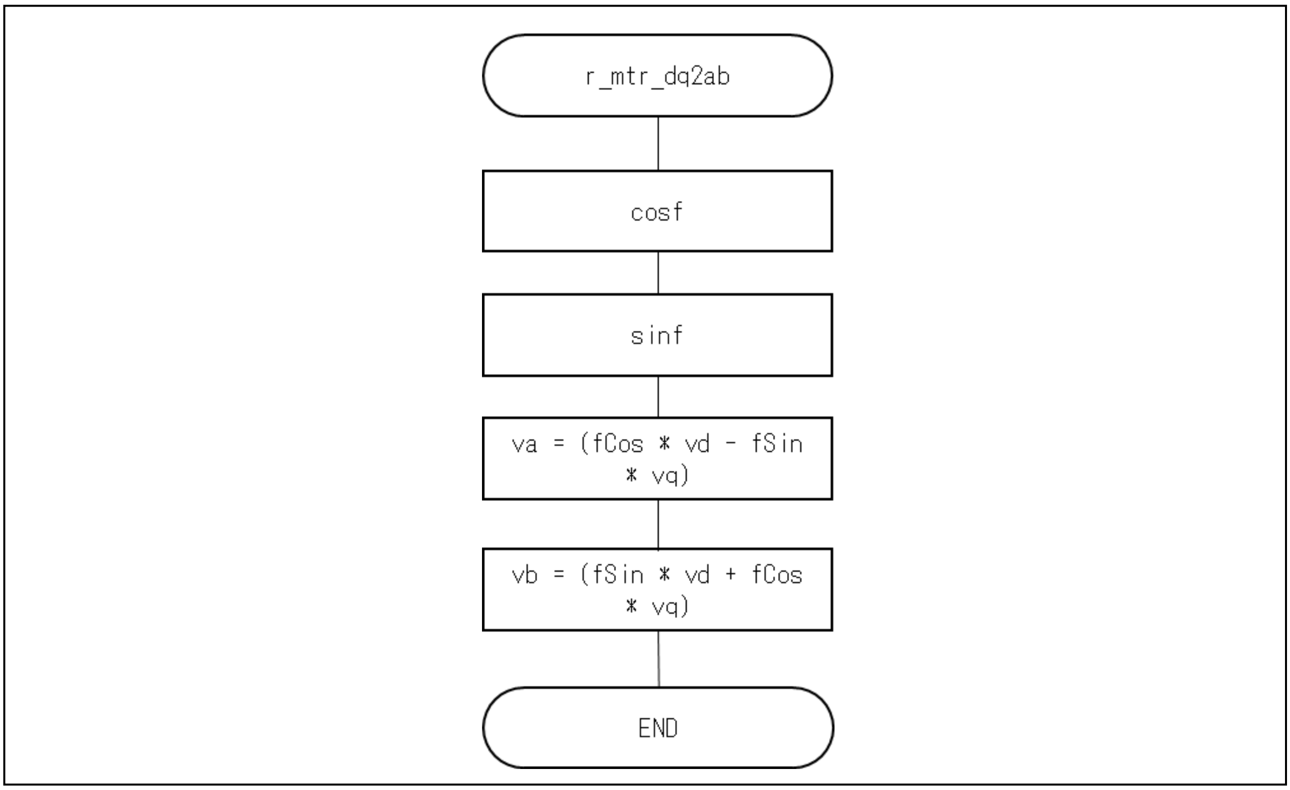


Figure 3-23 : dq → Alpha-Beta conversion processing flowchart

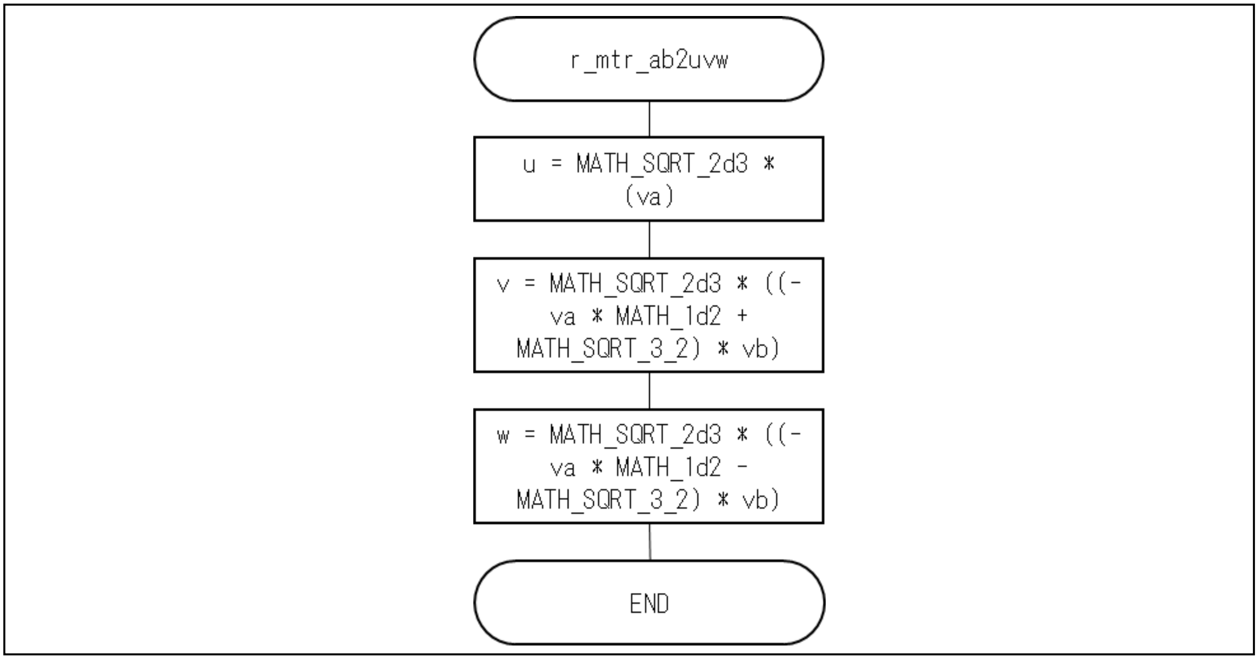


Figure 3-24 : Alpha-Beta → uvw conversion processing flowchart

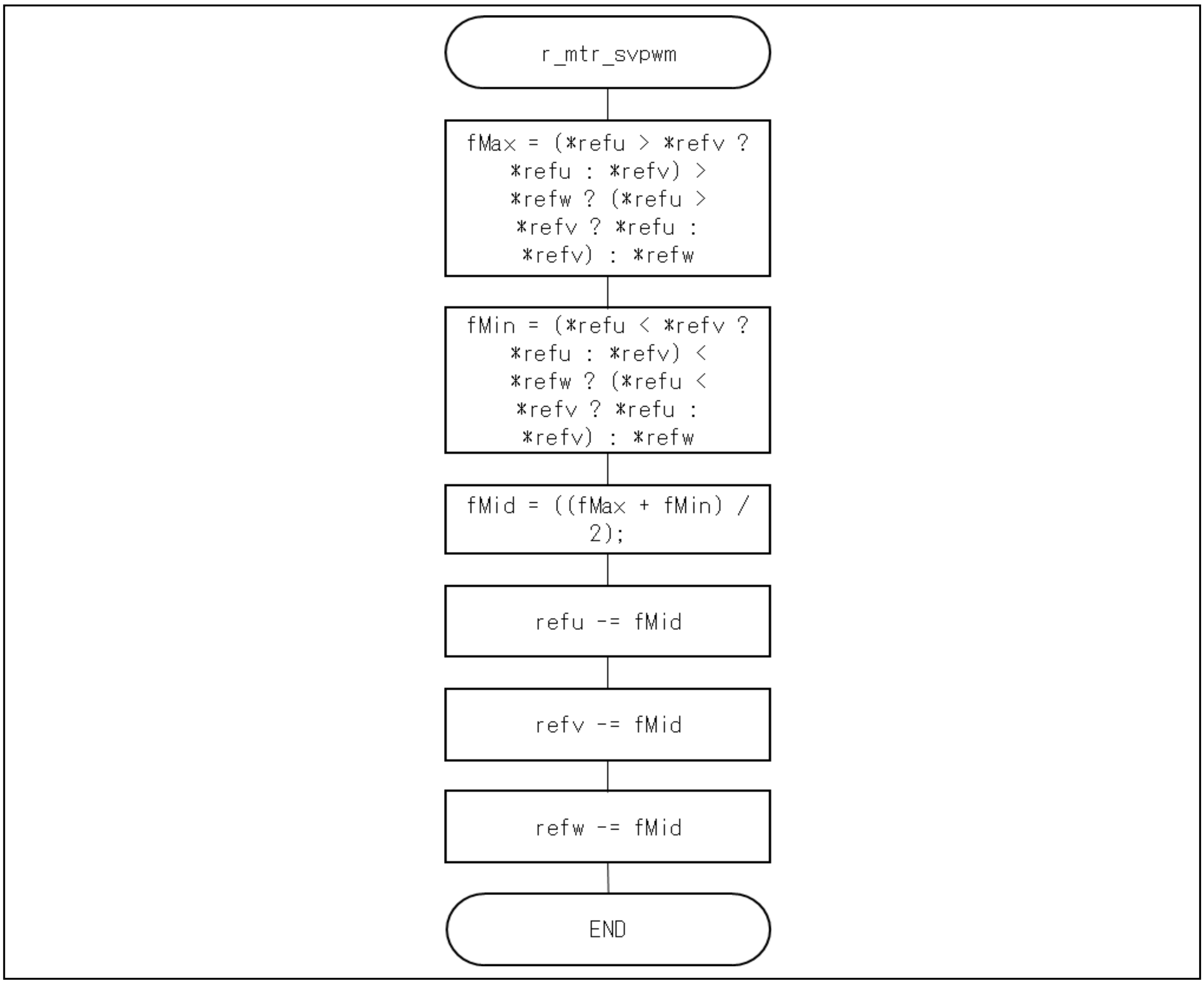


Figure 3-25 : pwm output-voltage calculation processing flowchart

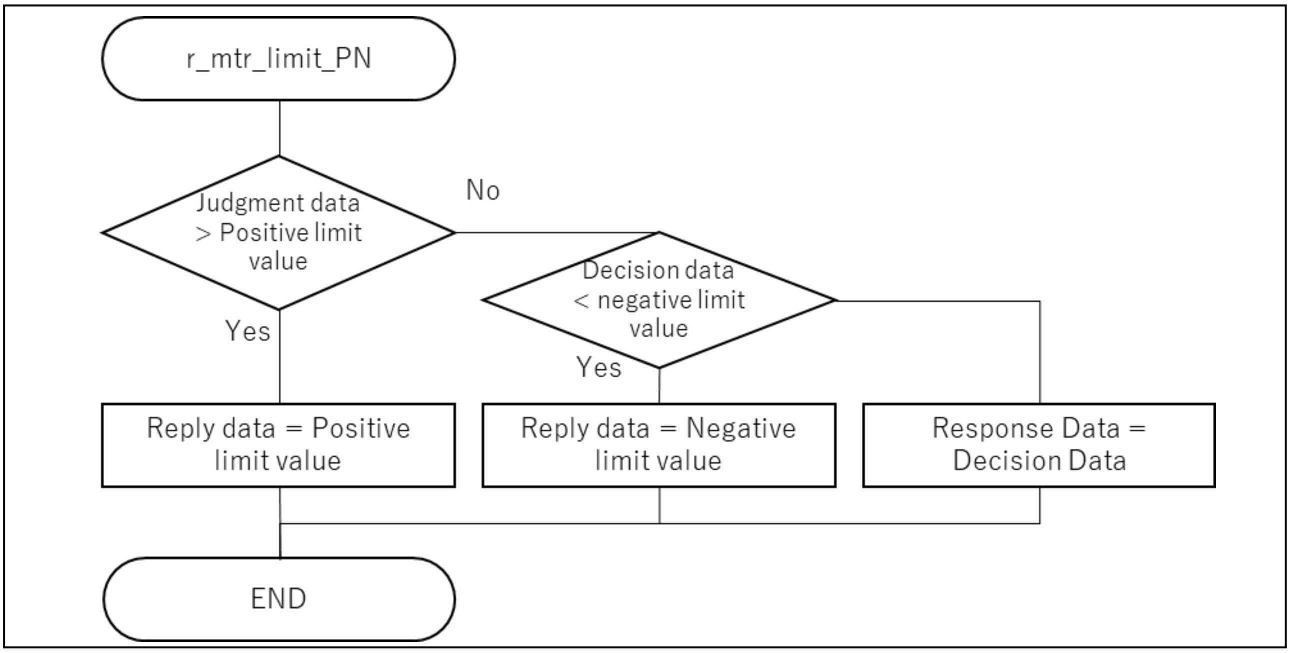


Figure 3-26 : Positive/Negative data range limit processing flowchart

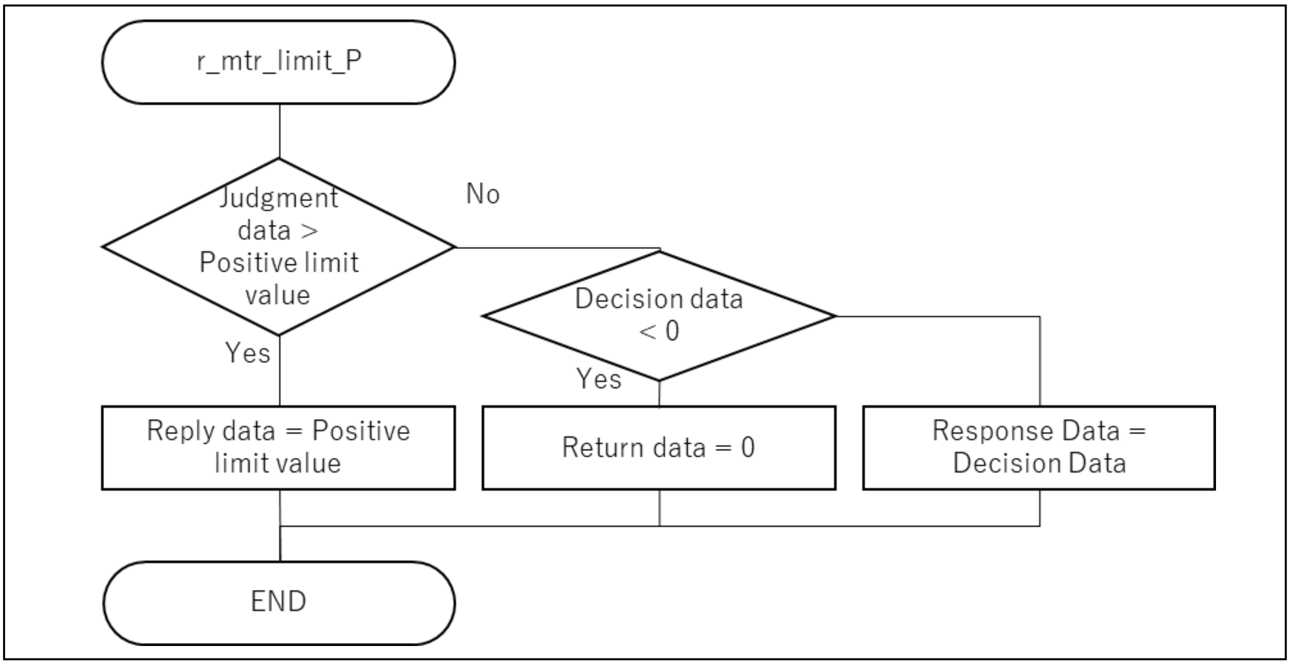


Figure 3-27 : Positive data range limit processing flowchart

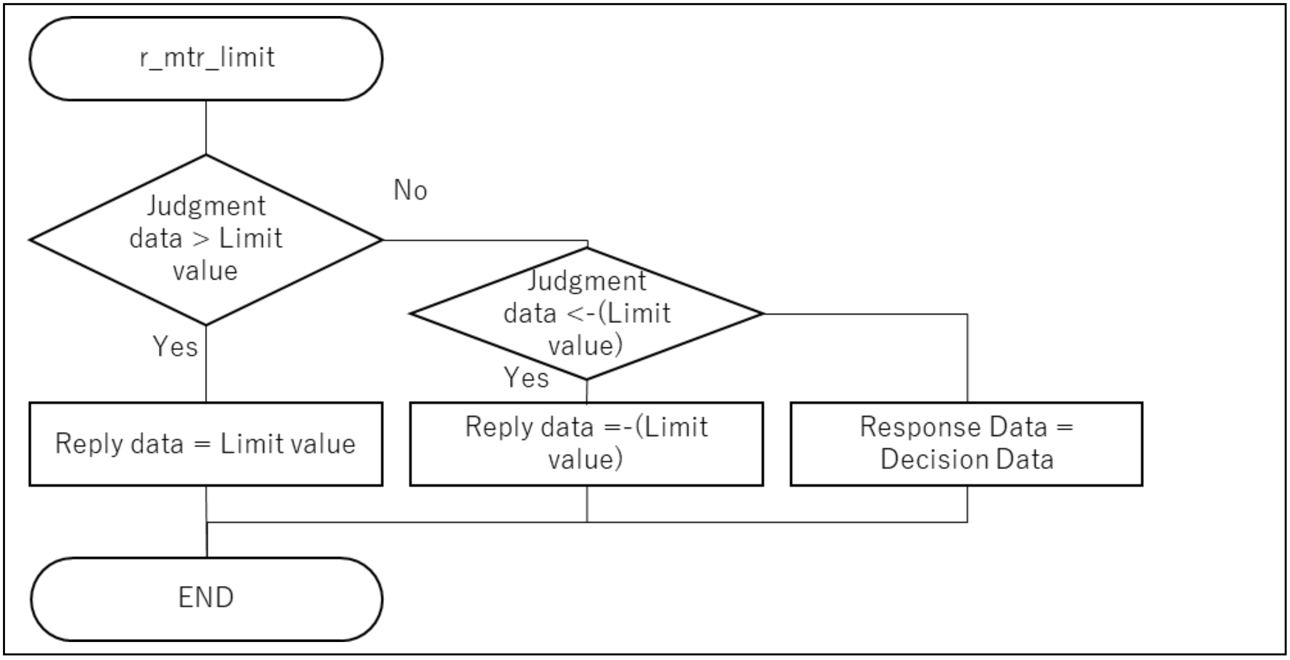


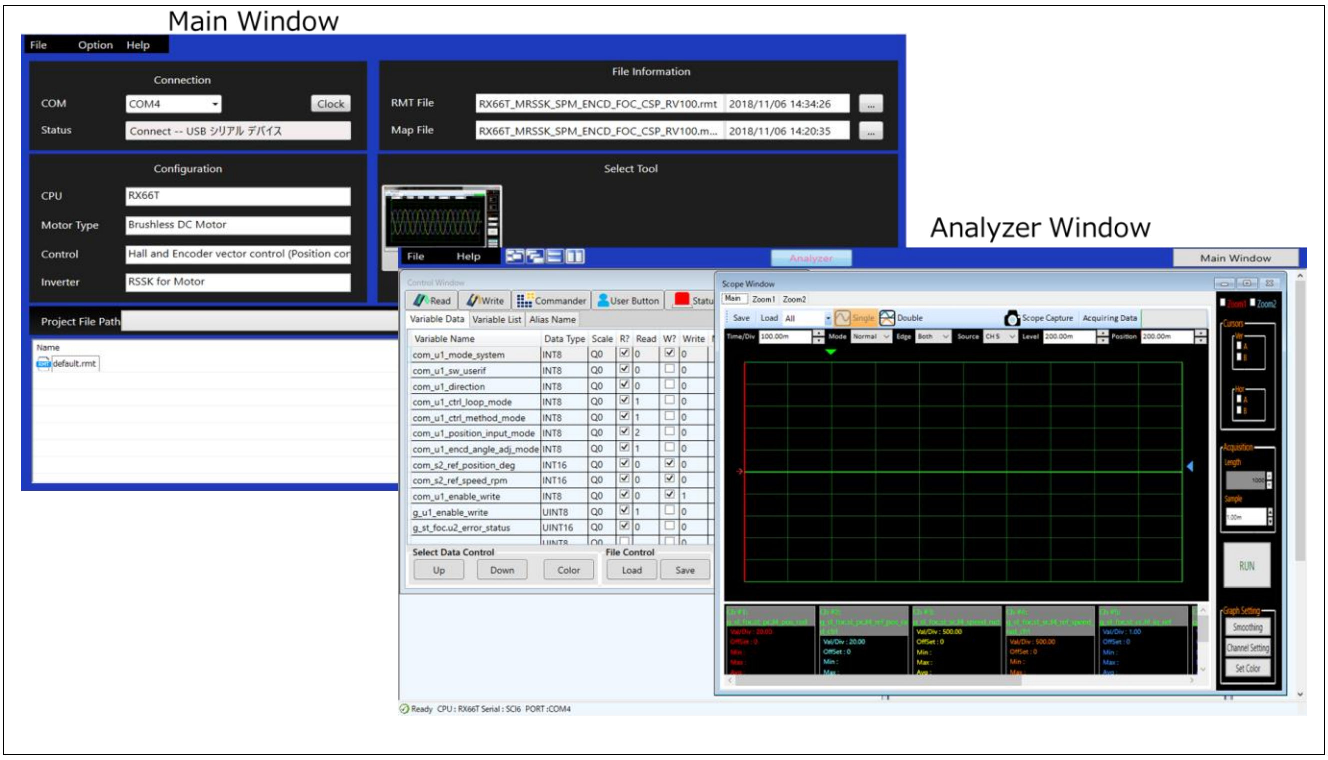
Figure 3-28 : Data range limit processing flowchart

4. Motor control support tool "Renesas Motor Workbench"

4.1 Overview

In the target sample programs described in this application note, user interfaces (rotating/stop command, rotation speed command, etc.) are based on the motor control development support tool, ‘Renesas Motor Workbench’ can be used. Please refer to ‘Renesas Motor Workbench User’s Manual’ for usage and more details.

You can find ‘Renesas Motor Workbench’ on Renesas Electronics Corporation website.



COM: COM4
Status: Connect -- USB シリアル デバイス
RMT File: RX66T_MRSSK_SPM_ENCD_FOC_CSP_RV100.rmt 2018/11/06 14:34:26
Map File: RX66T_MRSSK_SPM_ENCD_FOC_CSP_RV100.m... 2018/11/06 14:20:35
CPU: RX66T
Motor Type: Brushless DC Motor
Control: Hall and Encoder vector control (Position cor
Inverter: RSSK for Motor

Variable Name	Data Type	Scale	R?	Read	W?	Write
com_u1_mode_system	INT8	Q0	☑	0	☑	0
com_u1_sw_userif	INT8	Q0	☑	0	☐	0
com_u1_direction	INT8	Q0	☑	0	☐	0
com_u1_ctrl_loop_mode	INT8	Q0	☑	1	☐	0
com_u1_ctrl_method_mode	INT8	Q0	☑	1	☐	0
com_u1_position_input_mode	INT8	Q0	☑	2	☐	0
com_u1_encd_angle_adj_mode	INT8	Q0	☑	1	☐	0
com_s2_ref_position_deg	INT16	Q0	☑	0	☑	0
com_s2_ref_speed_rpm	INT16	Q0	☑	0	☑	0
com_u1_enable_write	INT8	Q0	☑	0	☑	1
g_u1_enable_write	UINT8	Q0	☑	1	☐	0
g_st_foc.u2_error_status	UINT16	Q0	☑	0	☐	0

Time/Div: 100.00m
Mode: Normal
Edge: Both
Source: CH 5
Level: 200.00m
Position: 200.00m
Length: 1000
Sample: 1.00m

Figure 4-1 : Appearance of Renesas Motor Workbench

Using the Motor control development support tool "Renesas Motor Workbench"

① Click the tool icon to launch the tool.
② From MENU of Main Panel, select [RMTFile] → [Open RMT File(O)].
Load the RMT file in the ics folder of the sample software folder.
③ Select the COM of the "Connection" COM-connected kit.
④ Click "Analyzer"' in the upper-right corner of Select Tool to open Analyzer function window.
⑤ Driving the motor based on "4.3Analyzer Function Operating Example".

4.2 List of variables for analyzer function

For the variables that can be used (displayed) in Analyzer, all the variables listed in Table 3-12 Variable list can be used. Table 4-1 shows the user interface input variables related to direct control. The following variables immediately reflect the input value.

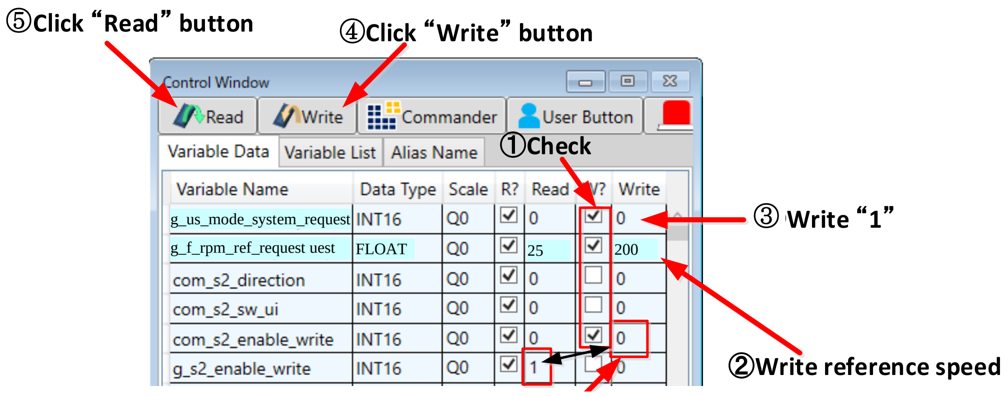
Table 4-1 : List of parameters for entering 1 analyzer function

Variable Name	Type	Content
g_us_mode_system_request	Int16	State Management 0:Stop mode 1:Run mode 3:Reset
g_f_rpm_ref_request	Int16	Speed command value (mechanical angle) [rpm]
g_f_rpm_slope_request	Float32	Velocity command [rpm/s]
g_f_rpm_oltovec	Float32	Speed for switching from open loop control to vector control [rpm]
g_f_rpm_enhance	Float32	Rate at which Id is decayed [rpm]
g_f_id_ref_ol_request	Float32	Maximum id value for open loop [A]

4.3 Operation example for analyzer

Following is an example demonstrating the motor driving operation using Analyzer. Operation is using “Control Window”. Refer to ‘In Circuit Scope manual’ for ‘Renesas Motor Workbench User’s Manual’.

① Confirm that "CHECK" is displayed in the [W?] field of "g_us_mode_system_request" and "g_f_rpm_ref_request".
② Enter the command speed in the [Write] field of "g_f_rpm_ref_request".
③ Enter "1" in the [Write] field of "g_us_mode_system_request".
④ Press "Write".
⑤ Press "Read" to check the [Read] field of "g_f_rpm_ref_request".

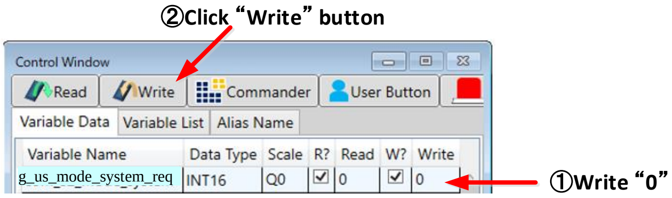


Variable Name	Data Type	Scale	R?	Read	W?	Write
g_us_mode_system_request	INT16	Q0	☑	0	☑	0
g_f_rpm_ref_request uest	FLOAT	Q0	☑	25	☑	200
com_s2_direction	INT16	Q0	☑	0	☐	0
com_s2_sw_ui	INT16	Q0	☑	0	☐	0
com_s2_enable_write	INT16	Q0	☑	0	☑	0
g_s2_enable_write	INT16	Q0	☑	1	☐	0

Figure 4-2 : Procedure for motor rotation

Stop motors

① Enter "0" in the [Write] field of "g_us_mode_system_request".
② Press "Write".

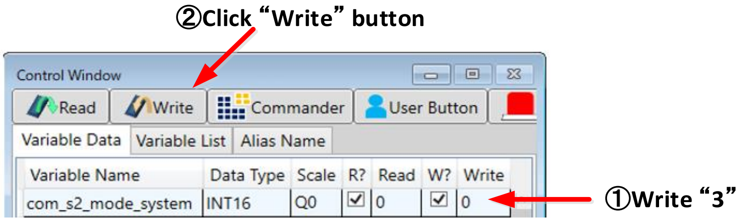


Variable Name	Data Type	Scale	R?	Read	W?	Write
g_us_mode_system_req	INT16	Q0	☑	0	☑	0

Figure 4-3 : Procedure for stopping the motor

Processing when stopped (error)

① Enter "3" in the [Write] field of "com_s2_mode_system".
② Press "Write".



Variable Name	Data Type	Scale	R?	Read	W?	Write
com_s2_mode_system	INT16	Q0	☑	0	☑	0

Figure 4-4 : Error clearing procedure

5. Appendix「FAQ」

■About this system

1. What equipment is required to handle this system?

→This board (Ceiling FAN Inverter Board with RX13T by Desk Top Laboratories Inc.), ICS Board by Desk Top Laboratories Inc., Target Motor, and Motor Control Development Support Tool "Renesas Motor Workbench"/In Circuit Scope by Desk Top Laboratories Inc. (ICS). E2Lite+ Integrated Development Environment (CS+/E2Studio) is required to modify the software and check the operation of tracing.

2. I would like to purchase this system.

→The Ceiling FAN Inverter Board with RX13T can be purchased from Desk Top Laboratories Inc. Contact to Desk Top Laboratories Inc.

3. Can I provide sample software and other design data for reference, such as schematics?

→It can be provided by us. It can also be provided from Desk Top Laboratories Inc.

4. Is it design data paid for?

→It is provided free of charge.

5. I would like to ask for support in order to carry out modifications, etc.

→Consulting and engineering work are contracted at Desk Top Laboratories Inc. Contact to Desk Top Laboratories Inc.

■Control Software

1. What is the CPU load factor?

→PFC control (current continuous mode) + pick-up control + vector control (3-shunt sensorless vector control + other processing) is about 92% (PFC:17us, pick-up: 30us, vector control: 63us).

2. Can I set PFC control to low frequency?

→It is possible. Since Ch0 of MTU3 is used for PFC control, reduce the count frequency or set the cycle in the count cycle register. In addition, the circuit constants, including hardware indictors, must be revised.

3. I want to increase the carrier frequency of the inverter control.

→It is possible. In this case, you will need to increase the control skipping count. The current software is a carrier-frequency 16KHz with one control decimation (the control cycle is 8KHz).

4. Can I change the drive control from 3 shunt control to 1 shunt control?

→It is possible. In this case, the control software must be modified. If necessary, Contact to Desk Top Laboratories Inc.

5. Does the PFC control algorithms require the delay of Vac?

→Phase difference may occur due to reactance and capacitor constant. Use this as a fine adjustment factor.

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
01.00	Sep.09.21	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit 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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2021 Renesas Electronics Corporation. All rights reserved.