

RX13T 搭載シーリング FAN インバータボードによる PFC 制御、及びセンサレスベクトル制御

要旨

本アプリケーションノートは 株式会社デスクトップラボ (DTL) 製 RX13T 搭載シーリング FAN インバータボードを使用し、DC ブラシレスモータ (シーリング FAN) をベクトル制御で駆動するサンプルプログラム、及びモータ制御開発支援ツール「Renesas Motor Workbench 2.0」の使用方法について説明することを目的としています。サンプルプログラムはあくまで参考用途であり、弊社がこの動作を保証するものではありません。サンプルプログラムを使用する場合、適切な環境で十分な評価をしたうえで御使用ください。特に、高電圧環境の取り扱いは非常に危険です。各開発環境のユーザーズマニュアルを良くお読みのうえ、安全に御使用ください。本アプリケーションノート掲載開発環境での事故、損害等が発生した場合、一切の責任を負いません。

動作確認デバイス

サンプルプログラムの動作確認は下記のデバイスで行っております。

- RX13T (R5F513T5ADFL)

対象ソフトウェア

本アプリケーションノートの対象ソフトウェアを下記に示します。

シーリングFAN向けRX13Tベクトル制御ソフトウェア

- E1-001_RX13T_CeilingFan_Drive_B009-CSP_RV100 (IDE : CS+版)
- E1-001_RX13T_CeilingFan_Drive_B009-E2S_RV100 (IDE : e2studio 版)

対象ハードウェア

本アプリケーションノートの対象ハードウェアを下記に示します。

株式会社デスクトップラボ製 RX13T搭載シーリングFANインバータボード

- APINV003-RX13T-CEILING

参考資料

- RX13T グループ ユーザーズマニュアル ハードウェア編 (R01UH0822)
- 永久磁石同期モータのエンコーダベクトル制御 (アルゴリズム編) (R01AN3789)
- Renesas Motor Workbench ユーザーズマニュアル (R21UZ0004)
- スマート・コンフィグレータ ユーザーズマニュアル RX API リファレンス編 (R20UT4360)
- RX スマート・コンフィグレータ ユーザーガイド: CS+編 (R20AN0470)
- RX スマート・コンフィグレータ ユーザーガイド: e2 studio 編 (R20AN0451)

目次

1. 概説	3
1.1 開発環境	3
2. システム概要	4
2.1 システム構成	4
2.2 システム特長	4
2.3 基板構成	5
2.4 ハードウェア構成	6
2.5 ハードウェア仕様	7
2.5.1 ユーザインタフェース	7
2.5.2 周辺機能	7
2.6 ソフトウェア構成	8
2.6.1 ソフトウェア・ファイル構成	8
2.6.2 モジュール構成	9
2.7 ソフトウェア仕様	10
3. 制御プログラム説明	10
3.1 制御内容	10
3.1.1 モータ起動／停止	10
3.1.2 モータ回転速度指令値	11
3.1.3 A/D変換	11
3.1.4 速度制御	12
3.1.5 変調	12
3.1.6 PWMによる電圧制御	13
3.1.7 ベクトル制御	13
3.1.8 拾い上げ制御	14
3.1.9 Power factor correction (PFC) 制御	14
3.1.10 状態遷移	16
3.1.11 センサレス制御時の始動方法	16
3.1.12 システム保護機能	17
3.2 関数仕様	18
3.3 変数一覧	20
3.4 マクロ定義	25
3.5 制御フロー（フローチャート）	32
3.5.1 メイン処理	32
3.5.2 キャリア周期割り込み処理	33
3.5.3 1 [ms] 割り込み処理	38
3.5.4 Mathライブラリ処理	43
4. モータ制御開発支援ツール「Renesas Motor Workbench」	47
4.1 概要	47
4.2 Analyzer機能用変数一覧	48
4.3 Analyzer機能操作例	48
5. Appendix「FAQ」	50

1. 概説

本アプリケーションノートでは、RX13T マイクロコントローラを使用した永久磁石同期モータ（PMSM）^{注1}のベクトル制御サンプルプログラムの実装方法及びモータ制御開発支援ツール^{*2}「Renesas Motor Workbench」（以降、RMW）の使用方法について説明します。なお、このサンプルプログラムは株式会社デスクトップラボのベクトル制御アルゴリズムを使用しています。アルゴリズムのご質問については、株式会社デスクトップラボへお問い合わせをお願いいたします。

- 【注】 1. 別称：ブラシレス DC モータ（BLDC）
2. 株式会社デスクトップラボ製 In Circuit Scope(ICS) もご使用頂けます

1.1 開発環境

本アプリケーションノート対象サンプルプログラムの開発環境を表1-1、表1-2に示します。

表1-1 サンプルプログラムの開発環境（H/W）

マイコン	評価ボード	モータ
RX13T (R5F513T5ADFL)	株式会社デスクトップラボ製 RX13T 搭載シーリング FAN インバータボード ^{注1} (APINV003-RX13T-CEILING)	Palkdale ^{注2}

表1-2 サンプルプログラムの開発環境（S/W）

CS+バージョン	e ² studio バージョン	ツールチェーンバージョン
V8.05.00	2021-01	CC-RX : V3.03.00

ご購入、技術サポートにつきましては、弊社営業及び特約店にお問い合わせください。

- 【注】 1. 株式会社デスクトップラボのシーリング FAN 用インバータボードを使用しております。
株式会社デスクトップラボ (<http://www.desktoplab.co.jp/>)
2. Palkdale 社シーリングファン搭載の 4 極モータを使用しております。

2. システム概要

本システムの概要を以下に説明します。

2.1 システム構成

システム構成を以下に示します。

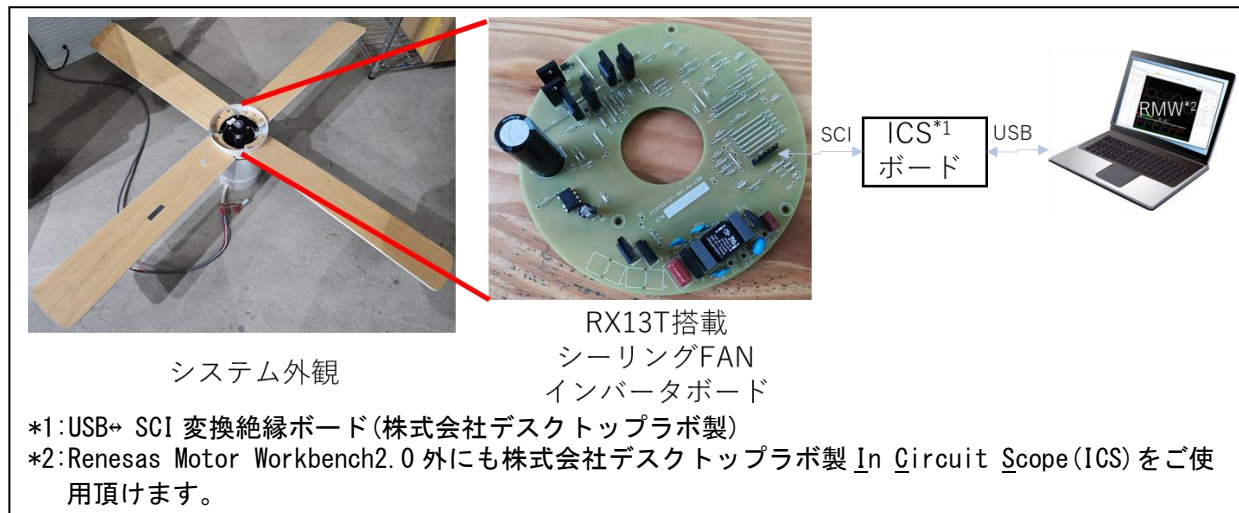


図2-1 システム構成図

2.2 システム特長

株式会社デスクトップラボ製 RX13T 搭載シーリングFAN インバータボードを使用したシーリングFAN システムの特長を以下に示します。

表2-1 システム特長

分類	内容
ハードウェア	基板搭載スペースの少ないシーリング FAN 向けにローコストの片面実装基板を採用した機電一体のシステム構成
	最大 260V の商用電源入力に対し PFC 制御を高周波化 (50KHz) 化する事でインダクタを小さくでき (4mH) 、インダクタ部品のオンボード化を実現 (一般的には基板外付け)
	少ピン小型パッケージ (LQFP48 (7mm□)) により基板レイアウトの自由度が向上
	RX13T 搭載の 3ch 同時サンプル&ホールド回路、及び内蔵プログラマブルゲインアンプ (PGA) を使用し、外付け部品の削減に貢献
ソフトウェア	RX13T@32MHz で 3 シャントセンサレスベクトル制御 (拾い上げ制御含む) + 電流連続モード PFC 制御 (インバータ非同期制御) を 1 チップで実現
	PFC 制御を高周波化 (キャリア周波数: 50KHz) する事で、力率が最大 99% と大幅改善
	拾い上げ制御によりモータを完全停止させずにモータドライブ可能。また 3 シャントセンサレスベクトル制御 (キャリア周波数: 16KHz) で無段階に速度変更可能

* PFC: power factor correction

2.3 基板構成

主要部品にルネサス製部品を使用した RX13T 搭載シーリング FAN インバータボード (APINV003-RX13T-CEILING) の構成を図2-2に、ルネサス製の搭載部品リストを表2-2に示します。

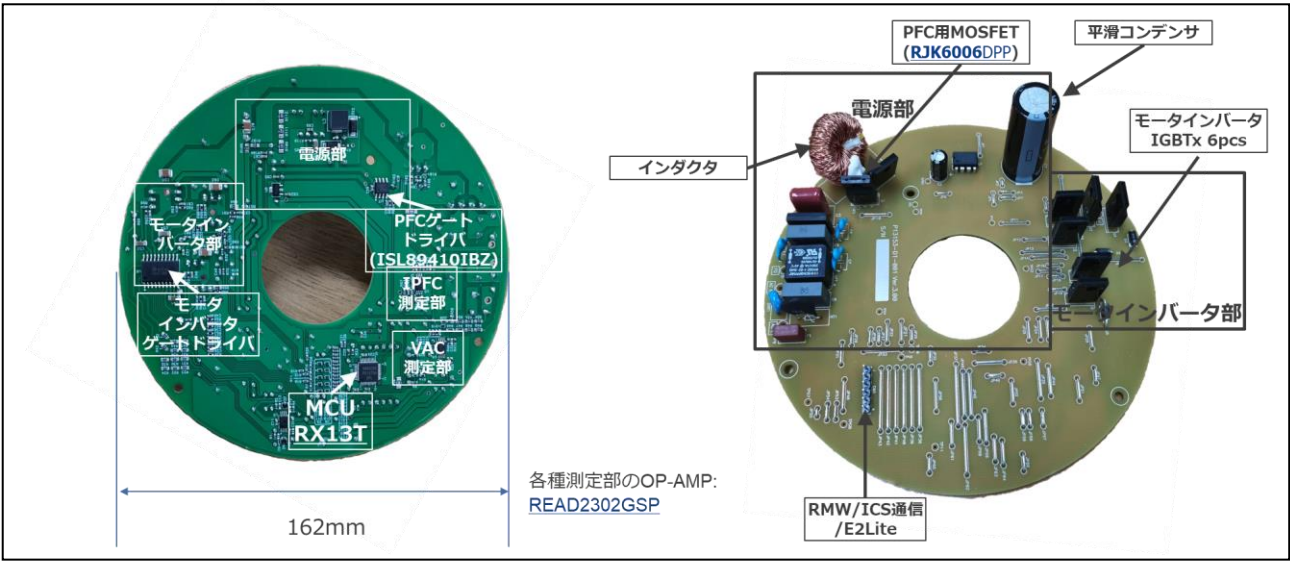


図2-2 基板構成図

表2-2 ルネサス製の搭載部品

Item	Detail	Product Name	Package
PFC MOSFET	Nch Single Power Mosfet 600V 5A 1600Mohm	RJK6006DPP	T0-220FPA
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 ハードウェア構成

ハードウェア構成を次に示します。

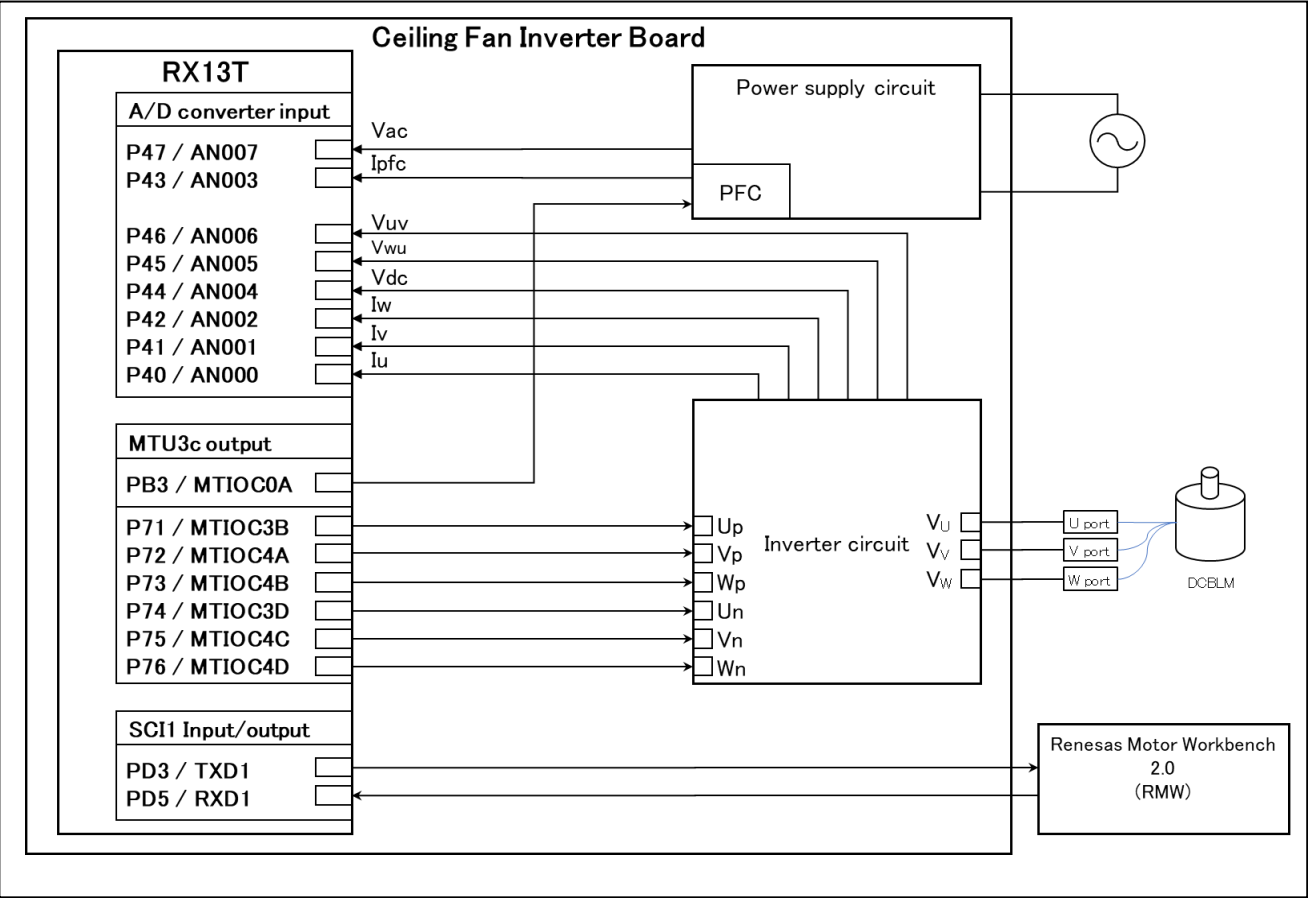


図2-3 ハードウェア構成図

2.5 ハードウェア仕様

2.5.1 ユーザインタフェース

本システムは、モータ制御開発支援ツールの Renesas Motor Workbench2.0(RMW)、もしくは株式会社デスクトップラボ製 In Circuit Scope(ICS)によりシステムコントロールする事を前提としておりますので、ハードウェアのユーザインタフェースはございません。本ツールでシステム動作中にシステム制御変数を直接操作しシステムコントロールをします。詳細は、4章のモータ制御開発支援ツール「Renesas Motor Workbench」をご参照お願いいたします。

本システムの RX13T マイクロコントローラ端子のインタフェース一覧を表2-3に示します。

表2-3 端子インタフェース

R5F513T5ADFL 端子名	信号名	IN/OUT	機 能
P47 / AN007	Vac	IN	AC 電圧
P43 / AN003	Ipfc	IN	PFC 電流測定
P46 / AN006	Vuv	IN	Vuv 線間電圧測定
P45 / AN005	Vwu	IN	Vwu 線間電圧測定
P44 / AN004	Vdc	IN	インバータ母線電圧測定
P42 / AN002	Iw	IN	W 相電流測定
P41 / AN001	Iv	IN	V 相電流測定
P40 / AN000	Iu	IN	U 相電流測定
P71 / MT10C3B	Up	OUT	ポート出力/PWM 出力 (U_p) / “Low” アクティブ
P72 / MT10C4A	Vp	OUT	ポート出力/PWM 出力 (V_p) / “Low” アクティブ
P73 / MT10C4B	Wp	OUT	ポート出力/PWM 出力 (W_p) / “Low” アクティブ
P74 / MT10C3D	Un	OUT	ポート出力/PWM 出力 (U_n) / “High” アクティブ
P75 / MT10C4C	Vn	OUT	ポート出力/PWM 出力 (V_n) / ” High” アクティブ
P76 / MT10C4D	Wn	OUT	ポート出力/PWM 出力 (W_n) / ” High” アクティブ
PB3 / MT10C0A	PFC	OUT	ポート出力/PWM 出力 (PFC 用) / “High” アクティブ
PD3 / TXD1	TX	OUT	Renesas Motor Workbench2.0(RMW) or 株式会社デスクトップ ラボ製 <u>In Circuit Scope</u> (ICS) シリアル通信用
PD5 / RXD1	RX	IN	

2.5.2 周辺機能

本システムに使用する周辺機能一覧を表2-4に示します。

表2-4 サンプルプログラム別周辺機能対応表

12bit A/D	CMT	MTU3	SCI1
- AC 電圧測定 - PFC 電流測定 - インバータ母線電圧測定 - Vuv, Vwu 線間電圧測定 - 各 U/V/W 相電流	1 [ms] インターバルタイマ (回転速度指令用インターバル)	- モータ制御用 相補 PWM 出力 - PFC 制御用 PWM 出力	RMW 用シリアル通信

(1) 12bitA/D コンバータ

- AC 電圧 (V_{ac})、PFC 電流 (I_{pfc})、インバータ母線電圧 (V_{dc})、線間電圧 (V_{uv}, V_{wu})、U 相電流 (I_u)、V 相電流 (I_v)、W 相電流 (I_w)、を「12 ビット A/D コンバータ」を使用し測定します。
- 動作モードは「シングルスキャンモード」に設定します。(ハードウェアトリガを使用)
- U 相電流 (I_u)、V 相電流 (I_v)、W 相電流 (I_w) を、チャンネル専用サンプル&ホールド機能とプログラマブルゲインアンプ (ゲイン設定=5 倍) を使用し「シングルスキャンモード」で測定します (ハードウェアトリガを使用)。

(2) コンペアマッチタイマ (CMT)

1 [ms] インターバルタイマ: コンペアマッチタイマのチャンネル 0 を 1 [ms] インターバルタイマとして使用します。

(3) マルチファンクションタイマパルスユニット 3 (MTU3)

動作モードはチャンネル毎に異なります。

- チャンネル 3、4 では相補 PWM モードを使用し、モータ制御用デッドタイム付きの相補 PWM 出力 (p 側は“Low” アクティブ、n 側は“High” アクティブ) を行います。
- チャンネル 0 では PWM モード 1 を使用し、“High” アクティブの PFC 制御用 PWM 出力を行います。

(4) シリアルコミュニケーションインターフェース 1 (SCI1)

RMW との通信を行います。ボーレートは 4Mbps に設定しております。

2.6 ソフトウェア構成

2.6.1 ソフトウェア・ファイル構成

サンプルプログラムのフォルダとファイル構成を表2-5に記します。

表2-5 サンプルプログラムのフォルダとファイル構成

E1-001_RX13T_CeilingFan_Drive_B009	src	main.c	メイン関数、ユーザインタフェース制御
		intprg.c	割り込みハンドラ
		r_mtr_control.c	モータ制御、PFC 制御機能
		r_mtr_estimation.obj	推定機能 ^{注1}
		r_mtr_interrupt.c	モータ制御、PFC 制御、割り込み処理
		r_mtr_sequence.c	シーケンス制御
		r_mtr_control.h	モータ、インバータパラメータ関連定義
		r_mtr_interrupt.h	制御パラメータ定義
	src¥REL_src	resetprg.c	パワーオン時処理
		dbset.c	B, R セクション設定
		sbrk.c	メモリアロケーション処理
		vecttbl.c	ベクタテーブル初期化処理
		iodef.h	RX13T I/O レジスタ定義
		sbrk.h	アロケーションサイズ定義
		stackset.h	スタックエリアサイズ定義
		typedef.h	型定義
		vect.h	ベクター定義
	src¥Motor_IOLIB	r_mtr_IOLIB_AD.c	RX13T S12AD 関連処理
		r_mtr_IOLIB_CMT.c	RX13T CMT 関連処理

		r_mtr_IOLIB_INV_MTU_AD.c	RX13T MTU3 関連処理
		r_mtr_MATHLIB.c	ベクトル制御用算術演算処理
		r_mtr_MATHLIB.h	ベクトル制御用算術演算関連定義
		r_mtr_IOLIB.h	MCU 依存部定義
	src¥ICS_Lib	ICS2_RX13T.h	ツール用通信関連定義
		ICS2_RX13T.lib	ツール用通信ライブラリ

注 1: 推定機能はオブジェクト化しております。

2.6.2 モジュール構成

サンプルプログラムのモジュール構成を図2-4、表2-6に示します。

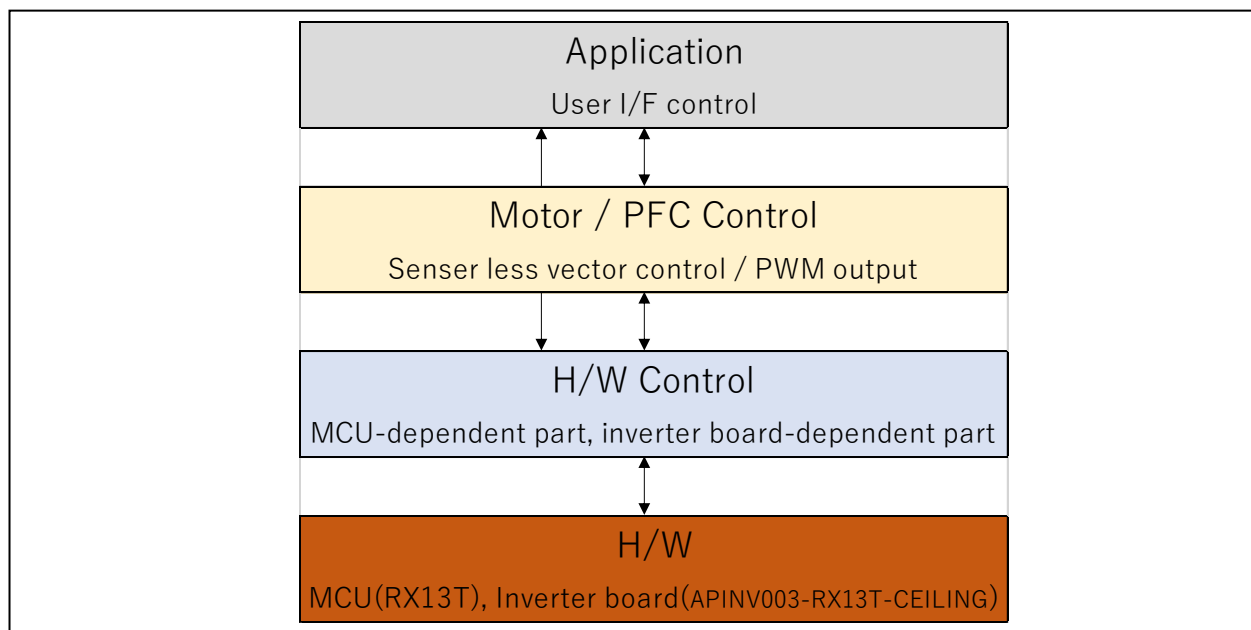


図2-4 サンプルプログラムのモジュール構成

表2-6 サンプルプログラムのモジュール構成

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

- 【注】 1. “r_mtr_sequence.c” は、アプリケーション層、モータ制御層に関わる処理を行っています。
 2. “r_mtr_interrupt.c” は、モータ制御層、H/W 層に関わる処理を行っています。

2.7 ソフトウェア仕様

本システムのソフトウェアの基本仕様を表2-7 センサレスベクトル制御ソフトウェア基本仕様に示します。

表2-7 センサレスベクトル制御ソフトウェア基本仕様

項目	内容
制御方式	センサレスベクトル制御
モータ制御開始/停止	RMW の数値入力により判定 (High(1) : 制御開始、Low(0) : 停止)
回転子磁極位置検出	シャント抵抗による推定
入力電圧	AC100~260V
メインクロック周波数	32 [MHz]
キャリア周波数 (PWM)	16 [kHz] (キャリア周期 : 62.5 [μs])
デッドタイム	2 [μs]
制御周期 (電流)	125 [μs] (キャリア周期の 2 倍)
制御周期 (速度)	1 [ms]
速度指令値管理	CW : 0 [rpm] ~ 250 [rpm] CCW : 0 [rpm] ~ 250 [rpm]
コンパイラ最適化設定	最適化レベル 2 (-optimize=2) (デフォルト設定)
最適化方法	コード・サイズ重視の最適化 (-size) (デフォルト設定)
保護停止処理	以下のいずれかの条件の時、モータ制御信号出力 (6 本) を非アクティブにする 1. 相電流 (Iu, Iv, Iw) が 2.06 [A] を超過 (125 [μs] 毎に監視) 2. インバータ母線電圧 (Vdc) が 300 [V] *1 を超過 (125 [μs] 毎に監視) 3. インバータ母線電圧 (Vdc) が 50 [V] *1 未満 (125 [μs] 毎に監視)

*1: AC100V 入力時の設定。AC260V 入力の際は、保護電圧レベルを引き上げる必要があります。

(AC260V 時の参考値 : 高電圧保護=420 [V]、低電圧保護=200 [V])

3. 制御プログラム説明

本アプリケーションノートの対象サンプルプログラムについて説明します。

3.1 制御内容

3.1.1 モータ起動/停止

モータの起動、停止は、RMW の Analyzer からの入力によって制御します。

CMT0 の割込み周期 (速度制御周期 [1ms]) で変数値 (g_us_mode_system_request) を読み、“High” レベル (1) のときモータを起動すると判断し、“Low” レベル (0) のときモータを停止すると判断します。

表3-1 モータ起動/停止指令値

項 目	指令値		変数名
モータ起動/停止	起動	1	g_us_mode_system_request
	停止	0	

3.1.2 モータ回転速度指令値

モータの回転速度指令値は RMW の Analyzer からの入力によって決定します。

以下の表のように、回転速度指令値として使用します。

表3-2 回転速度指令値

項 目	回転速度: 指令値		変数名
回転速度指令値	CW	0 [rpm] ~ 250 [rpm] : 0 ~ 250	g_f_rpm_ref_request
	CCW	0 [rpm] ~ 250 [rpm] : 0 ~ -250	

3.1.3 A/D 変換

(1) コンバータ AC 電圧

下記のように、コンバータ AC 電圧を測定します。変調率の算出と過電圧検出に使用します。

表3-3 コンバータ AC 電圧の変換比

項 目	変換比 (コンバータ AC 電圧 : A/D 変換値)	チャネル
コンバータ AC 電圧	-750 [V] ~ 750 [V] : 0000H ~ 0FFFH	AN007

(2) コンバータ PFC 電流

下記のように、コンバータ PFC 電流を測定します。変調率の算出と過電圧検出に使用します。

表3-4 コンバータ PFC 電流の変換比

項 目	変換比 (コンバータ PFC 電流 : A/D 変換値)	チャネル
コンバータ PFC 電流	-1.85 [A] ~ 1.85 [A] : 0000H ~ 0FFFH	AN003

(3) インバータ母線電圧

下記のように、インバータ母線電圧を測定します。変調率の算出と過電圧検出に使用します。

表3-5 インバータ母線電圧の変換比

項 目	変換比 (インバータ母線電圧 : A/D 変換値)	チャネル
インバータ母線電圧	0 [V] ~ 300 [V] : 0000H ~ 0FFFH	AN004

(4) UV 相、WU 相 線間電圧

下記のように、U、V、W 相の線間電圧を測定します。

表3-6 UV 相、WU 相線間電圧の変換比

項 目	変換比 (UV 相、WU 相線間電圧 : A/D 変換値)	チャネル
UV 相線間電圧	-750 [V] ~ 750 [V] : 0000H ~ 0FFFH	AN006
WU 相線間電圧		AN005

(5) U 相、V 相、W 相電流

以下の表のように、U 相、V 相、W 相電流を測定し、ベクトル制御に使用します。

表3-7 U、V、W 相電流の変換比

項 目	変換比 (U 相、V 相、W 相電流 : A/D 変換値)	チャネル
U 相電流 (I _u)	-2.06 [A] ~ 2.06 [A] : 0000H ~ 0FFFH ^{注1}	AN000
V 相電流 (I _v)		AN001
W 相電流 (I _w)		AN002

【注】 1. A/D 変換特性の詳細に関しては「RX13T グループ ユーザーズマニュアル ハードウェア編」を参照してください。

3.1.4 速度制御

本アプリケーションノート対象ソフトウェアでの速度制御は、PI 制御によって行います。下記の速度 PI 制御によって電圧指令値を得ます。

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

v^* : 電圧指令値

ω^* : 速度指令値

ω : 回転速度

$K_{p\omega}$: 速度PI比例ゲイン

$K_{I\omega}$: 速度PI積分ゲイン

s : ラプラス演算子

PI 制御の詳細については、専門書を参照してください。

3.1.5 変調

本アプリケーションノート対象ソフトウェアでは、モータへの入力電圧はパルス幅変調（以降、PWM）によって生成し、PWM 波形は三角波比較法によって生成します。三角波比較法とは、指令値電圧を実際に出力する方法の一つとして、キャリア波形（三角波）と指令値電圧波形を比較することで出力電圧のパルス幅を決める方法です。指令値電圧がキャリア波電圧より大きければスイッチをオン、小さければオフにすることで、正弦波状の指令値電圧を擬似的に出力することができます。

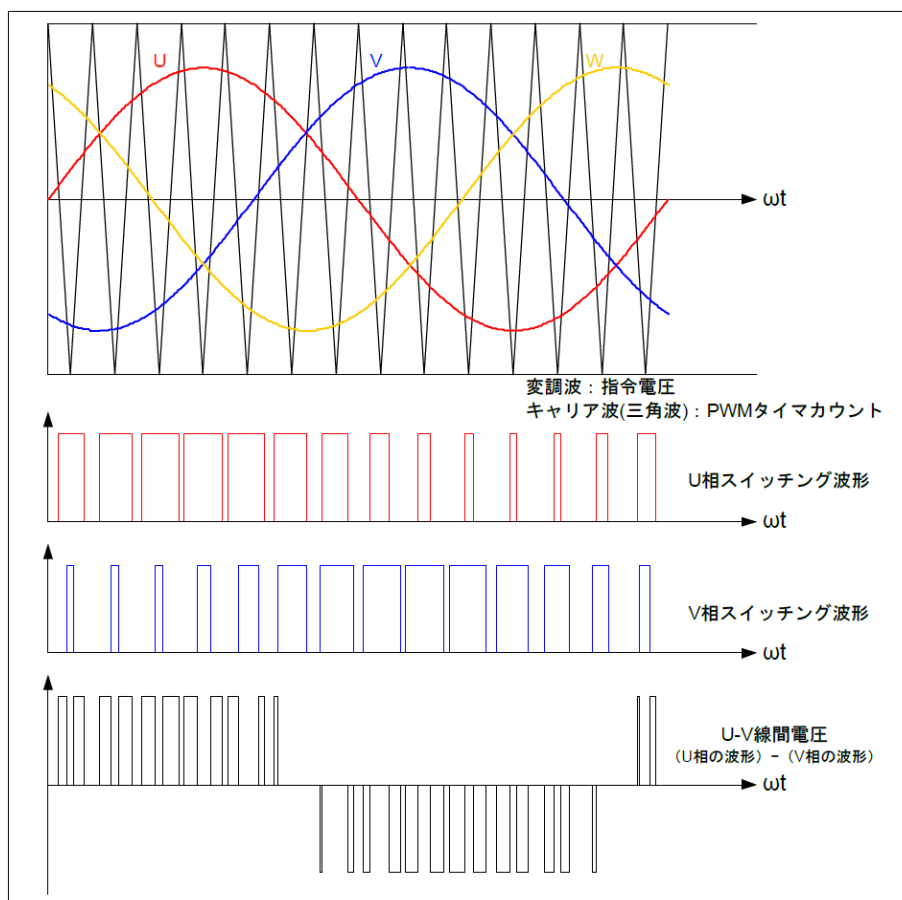


図3-1 三角波比較法の概念図

3.1.6 PWM による電圧制御

出力電圧の制御には PWM 制御を使用しています。PWM 制御とは、図3-2のように、パルスのデューティを変化させることで平均電圧を調整していく制御方式です。

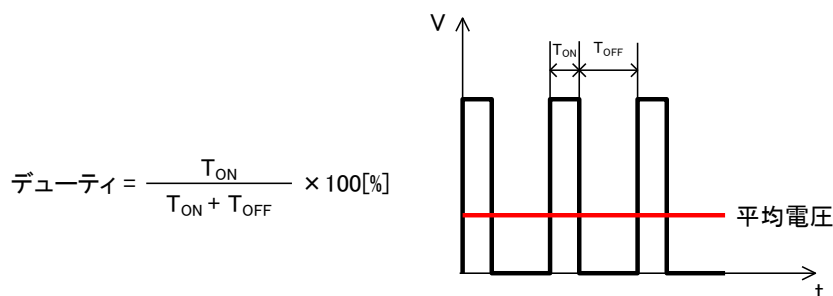


図3-2 PWM 制御

ここで、変調率 m を以下のように定義します。

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

m : 変調率 V : 指令値電圧 E : インバータ母線電圧

この変調率を、PWM デューティを決めるレジスタの設定値に反映させます。

3.1.7 ベクトル制御

図3-3にベクトル制御の概要図を示します。本ソフトウェアでは、株式会社デスクトップラボ製ベクトル制御アルゴリズムを使用しておりますが、参考としてアプリケーションノート「永久磁石同期モータのベクトル制御（アルゴリズム編）」をご参照いただけます。なおアルゴリズムのご質問については、株式会社デスクトップラボへお問い合わせをお願いいたします。

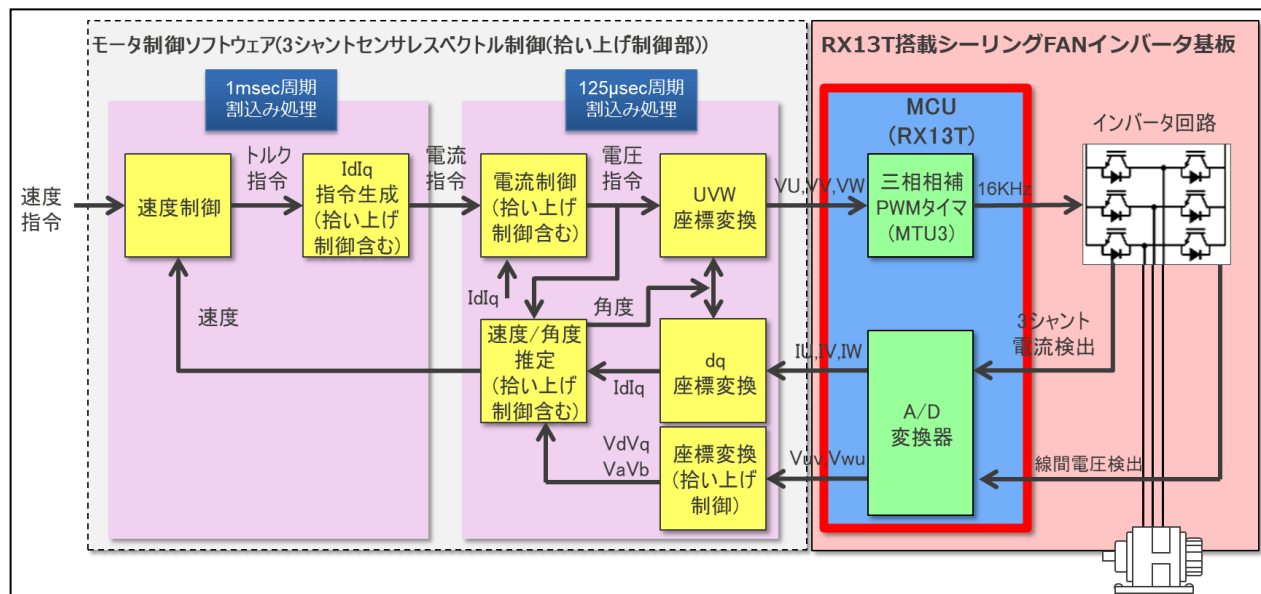


図3-3 ベクトル制御概要

3.1.8 拾い上げ制御

拾い上げ制御とはフリーラン動作しているモータの誘起電圧を測定し、回転角度・回転速度、及び速度指令に対するオフセットを計算する事で、モータの誘起電圧とインバータの出力電圧が同値になるようインバータ出力電圧を調整する事です。これによりフリーラン状態からの駆動でも電流の急変が抑えられ、モータを継続してスムーズに回すことができます。

なお、インバータが停止している時の出力端子 (Vu/Vv/Vw) だけでは角度計算が難しい為、相電圧から線間電圧 (Vuv/Vvw/Vwu) を計算し、角度と速度を求めます。(ベクトル制御の UVW→AB 座標変換を利用し、Va/Vb を求め、角度の計算を行います)

以下に計算手順を示します。

- ① 線間電圧 (u=Vuv、w=Vwu) で Va、Vb を求めます。

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

- ② Va、Vb から角度を計算します。

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

- ③ θ と Va、Vb から Vd、Vq を計算します。

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

- ④ Vd、Vq と誘起電圧と合わせて速度を計算します。

(線間電圧のベクトル値は相電圧より $\sqrt{3}$ 倍大きい為、 $1/\sqrt{3}$ します)

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

- ⑤ θ の微分計算を行い、速度を計算してから方向を決めます。

$$\text{Spd2} = \Delta \theta / \Delta t$$

式②で計算した角度は線間電圧 (Vuv、Vwu) がベースのため、Vu、Vw の角度とは 30 度の位相が発生します。また、Vu、Vw で角度を計算した場合においても、Va、Vb の角度になるため、実際に使う Vd、Vq の角度とは 90 度の位相があります。結果的に、線間電圧 Vuv/Vvw/Vwu で計算した角度は実際の電流制御上の角度とは 120 度の位相差がある為、ベクトル制御時に使う角度は以下の通り、120 度オフセット補正します。

$$\theta(\text{真}) = \theta - 120 \text{ 度}$$

3.1.9 Power factor correction(PFC) 制御

本システムでは力率を改善するため、電流連続モードシングル方式の PFC 制御を採用しております。表 3-8 に PFC 制御仕様を、図 3-4 に電流連続モードシングル方式の PFC 制御概要を示します。

なお、PFC 制御部は、インバータ制御部と非同期で動作しておりますので、シャント電流サンプリングのタイミング合わせこみなどが不要でありソフトウェア処理の柔軟性が増しますが、シャント電流サンプリング時にスイッチングノイズが重畳される可能性もあります。詳細は、の非同期制御のメリット・デメリットを参照ください。

表 3-8 PFC 制御仕様

項目	内容	備考
入力電圧	260Vac	本ソフトウェアでは 100VAC で確認しております

RX13T シーリング FAN インバータボードによる PFC 制御、及びセンサレスベクトル制御

入力電流	2A 以下	本ソフトウェアでは 1A で確認しております
電流サンプリング	シャント	270mΩ
PFC 制御周期	8kHz	
PFC キャリア	50kHz	
インバータ部との同期	非同期	インバータ部と PFC 部を完全独立で制御しており、シャント電流サンプリングなどのタイミング合わせこみが不要。

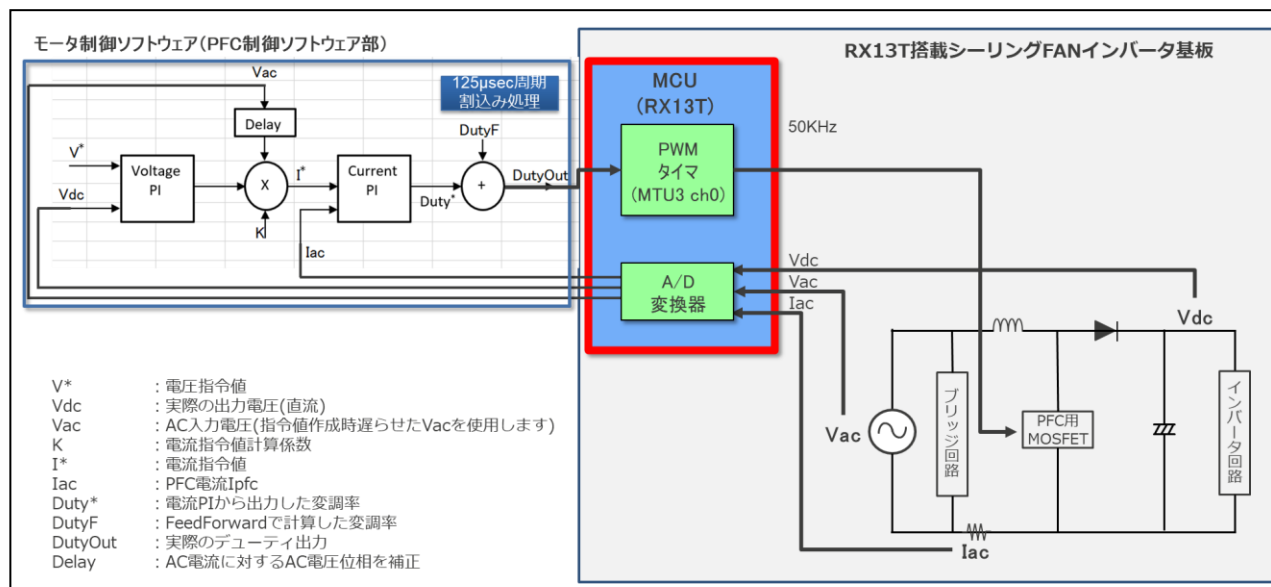


图3-4 PFC 制御概要

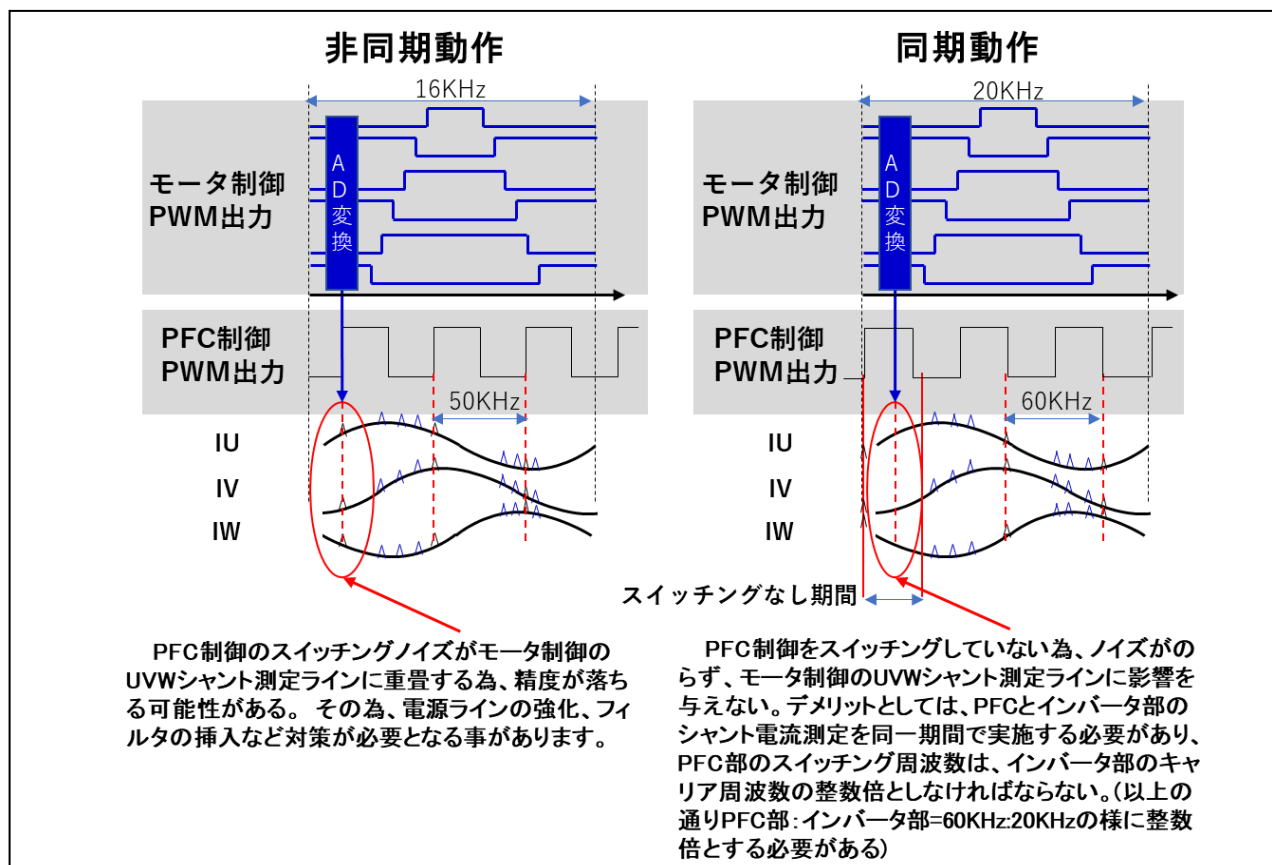


図3-5 非同期制御のメリット・デメリット

3. 1. 10 状態遷移

図3-6にベクトル制御ソフトウェアにおける状態遷移図、表3-9に状態表を示します。

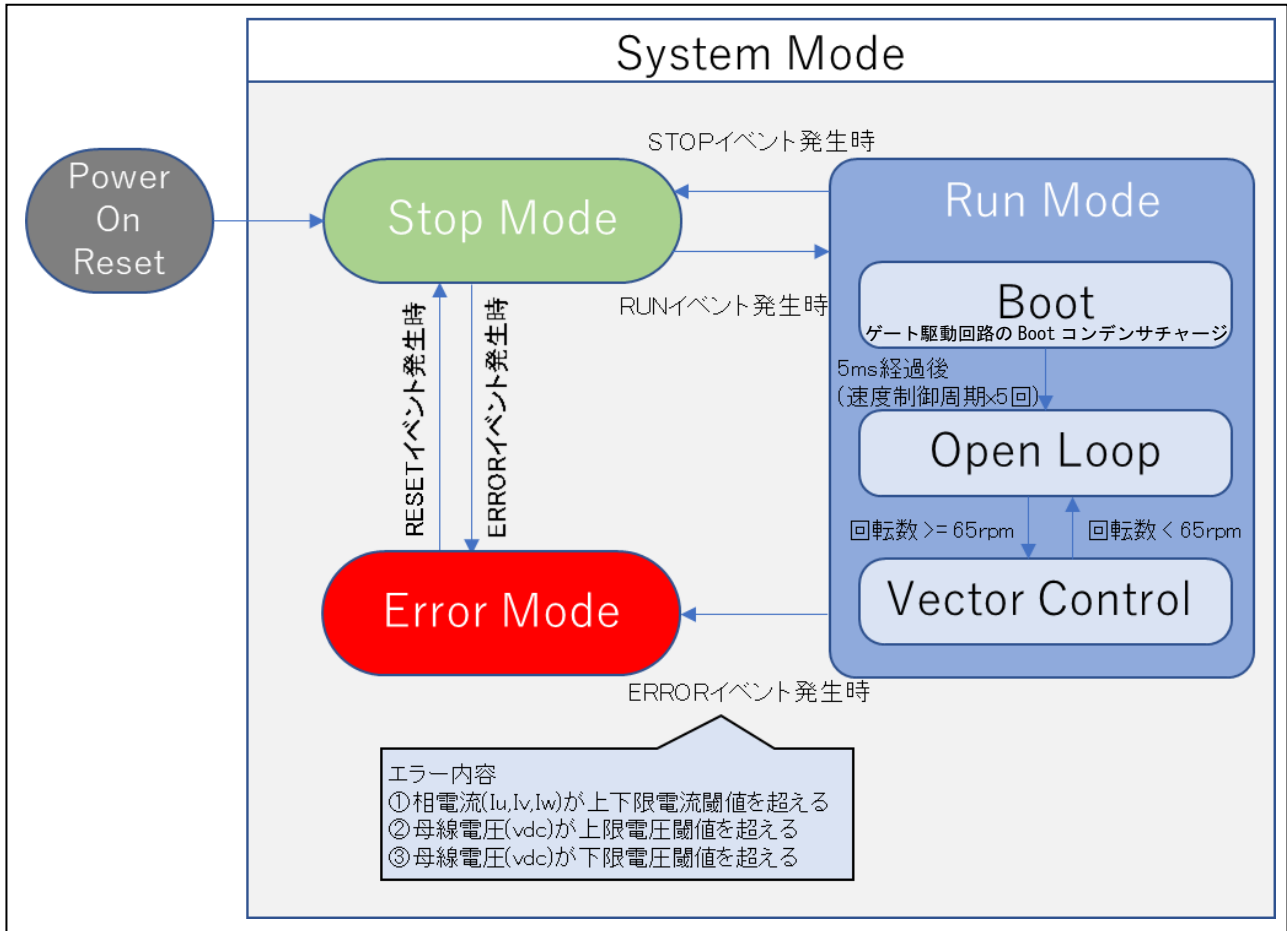


図3-6 状態遷移図

表3-9 状態表

		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 センサレス制御時の始動方法

図3-6 状態遷移図記載のとおり、ゲート駆動回路の Boot コンデンサチャージ期間(速度制御周期(1ms)を5回分)を経て、Open Loop に入ります。Open Loop と Vector Control の切り替えは、指定回転速度(本ソフトウェアでは 65rpm)により切替えをしております。更に指定回転速度(本ソフトウェアでは 85rpm)に達すると、Id を OFF にします。

図3-7に Open Loop から Vector Control への制御切替えタイミングを示します。

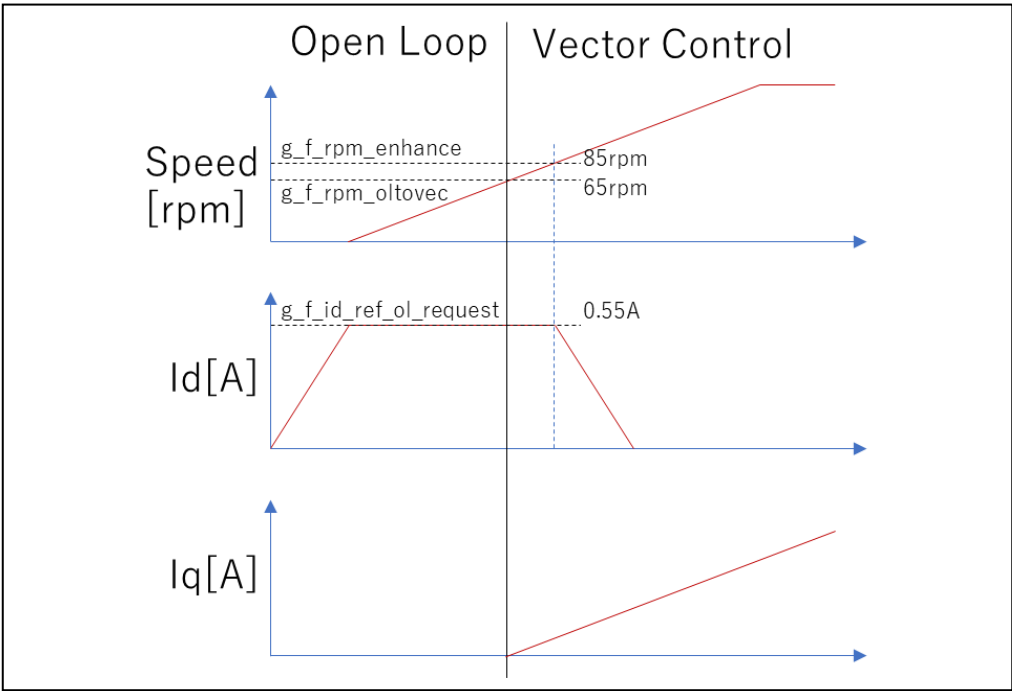


図3-7 制御切替え図

3.1.12 システム保護機能

本システムは、以下のエラー状態を持ち、それぞれの場合に緊急停止機能を実現しています。システム保護機能に関わる各設定値は表3-10を参照してください。

- 過電流エラー
過電流監視周期でU相、V相、W相電流を監視し、過電流（過電流リミット値を超過）を検出した時に、緊急停止します。（ソフトウェア検出）
- 過電圧エラー
過電圧監視周期でインバータ母線電圧を監視し、過電圧（過電圧リミット値を超過）を検出した時に、緊急停止します。（ソフトウェア検出）
ここで、過電圧リミット値は検出回路の抵抗値の誤差等を考慮して設定した値です。
- 低電圧エラー
低電圧監視周期でインバータ母線電圧を監視し、低電圧（低電圧リミット値を下回る）を検出した時に、緊急停止します。（ソフトウェア検出）
ここで、低電圧リミット値は検出回路の抵抗値の誤差等を考慮して設定した値です。

表3-10 各システム保護機能設定値

過電流エラー	過電流リミット値 [A]	2.06
	監視周期 [μ s]	125
過電圧エラー	過電圧リミット値 [V]	300 (AC100V 時)
	監視周期 [μ s]	125
低電圧エラー	低電圧リミット値 [V]	50 (AC100V 時)
	監視周期 [μ s]	125

3.2 関数仕様

本制御プログラムの関数一覧を以下に示します。一部関数はユーザが容易に組み込める様、拡張用として準備している関数(本ソフトウェアでは不使用の関数)も含めております。不使用関数は、以下一覧表の Notes をご参照下さい。

表3-11 関数一覧

File			Function				Notes
Path	Name	Overview	Name	Overview	Arguments	Return Type	
src	main.c	メインファイル	main	メイン処理	-	void	
	r_mtr_control.c	モータアルゴリズム	r_mtr_user_init_setting	初期化処理	-	void	
			r_mtr_user_init_control	制御初期化処理	-	void	
			user_control_speedloop	速度制御処理	-	void	
			r_mtr_timer_control	タイマ制御処理	-	void	
			r_mtr_User_CustomIO_init	IO 初期化処理	-	void	
			r_mtr_Estimation	推定処理	-	void	
			r_mtr_Ref_Check	参照値確認処理	float mu, float mv, float mw	int16_t	
			r_mtr_check_error_current_loop	電流ループエラーチェック処理	-	void	
			r_mtr_check_error_speedloop	速度ループエラーチェック処理	-	void	None Code
			r_mtr_error_stop	エラー停止処理	-	void	
			r_mtr_pfc_control	PFC 制御処理	-	void	
	r_mtr_interrupt.c	割り込み処理	interrupt_CMT0	1ms 速度制御処理	-	void	
			interrupt_MTU34_carrier	250us キャリア周期処理	-	void	
	r_mtr_sequence.c	シーケンス処理	seq_act_run	実行シーケンス処理	uint8_t ucState	uint8_t	
			seq_act_stop	停止シーケンス処理	uint8_t ucState	uint8_t	
			seq_act_reset	リセットシーケンス処理	uint8_t ucState	uint8_t	
			seq_act_error	エラーシーケンス処理	uint8_t ucState	uint8_t	
			seq_act_none	無処理シーケンス処理	uint8_t ucState	uint8_t	
			seq_init_start	シーケンス初期化処理	-	void	
			r_mtr_seq_init_setting	初期シーケンス処理	-	void	
			r_mtr_seq_exec_event	イベント実行シーケンス処理	uint8_t ucEvent	void	
src\Motor_IOLIB	r_mtr_Custom_IO.c	ハードウェア設定	r_mtr_Custom_VRL_init	ハードウェア初期化 (VRL) 処理	-	void	No use
			r_mtr_Custom_T2001_SW_init	ハードウェア初期化 (スイッチ) 処理	-	void	No use
			r_mtr_Custom_T2001_LED_init	ハードウェア初期化 (LED) 処理	-	void	No use

RX13T シーリング FAN インバータボードによる PFC 制御、及びセンサレスベクトル制御

	r_mtr_IOLIB_AD.c	ADC 設定	r_mtr_ad_ADO_init	ADO 初期化处理	uint16_t mode	void	
			r_mtr_ad_ADO_set_channel	ADO チャンネルセット処理	uint16_t ch_list	void	
			r_mtr_ad_ADO_set_range	ADO レンジセット処理	int16_t ch, int16_t offset, float range	void	
	r_mtr_IOLIB_CMT.c	CMT 設定	r_mtr_interval_CMT0_init	CMT0 初期化处理	uint16_t freq	void	
			r_mtr_interval_CMT1_init	CMT1 初期化处理	uint16_t freq	void	No use
	r_mtr_IOLIB_INV_MTU_AD.c	MTU3 設定	r_mtr_inverter_MTU34_init	相補 PWM 出力用 MTU3 (ch3, 4) 初期化处理	int32_t usFreqCarrier, uint32_t usDeadtime, uint32_t usDecimation	void	
			r_mtr_inverter_MTU34_set_uvw_3shunt	MTU3 (ch3, 4) 相補 PWM 出力値 Duty 設定処理	float refu, float refv, float refw	void	No use
			r_mtr_inverter_MTU34_set_uvw_3shunt_VariCar	MTU3 (ch3, 4) 相補 PWM 出力値 Duty 設定処理	float refu, float refv, float refw, float FreqCar	void	
			r_mtr_inverter_MTU34_boot_mode_lower_on	相補 PWM 出力値 Low 側 ON 設定処理	-	void	
			r_mtr_inverter_MTU34_boot_mode_lower_off	相補 PWM 出力値 Low 側 OFF 設定処理	-	void	No use
			r_mtr_inverter_MTU34_boot_mode_off	相補 PWM 出力モード停止処理	-	void	
			r_mtr_pwm_MTU0_set	PFC 制御用 PWM 出力値設定処理	float fDuty	void	
			r_mtr_pwm_MTU0_init	PFC 制御用 PWM 出力初期化处理	uint16_t usFreq	void	
	r_mtr_IOLIB_IWDT.c	独立 WDT 設定	r_mtr_IWDT_init	独立 WDT 初期化处理	-	void	No use
	r_mtr_MATHLIB.c	座標変換処理	r_mtr_uw2dq	uw→dq 変換処理	float iu, float iw, float theta, float * id, float * iq	void	
			r_mtr_uw2ab	uw→ $\alpha\beta$ 変換処理	float iu, float iw, float * ia, float * ib	void	

			r_mtr_ab2dq	$\alpha \beta \rightarrow dq$ 変換処理	float ia, float ib, float theta, float * id, float * iq	void	
			r_mtr_dq2ab	$dq \rightarrow \alpha \beta$ 変換処理	float vd, float vq, float theta, float * va, float * vb	void	
			r_mtr_ab2uvw	$\alpha \beta \rightarrow uvw$ 変換処理	float va, float vb, float * u, float * v, float * w	void	
			r_mtr_svpwm	svpwm 出力電圧計算処理	float * refu, float * refv, float * refw	void	
			r_mtr_limit_PN	データレンジ制限処理	float data, float limitp, float limitn	float	
			r_mtr_limit_P	正データレンジ制限処理	float data, float limitp	float	No use
			r_mtr_limit	負データレンジ制限処理	float data, float limit	float	
src¥REL_src	resetprg.c	パワーオンリセット	PowerON_Reset_PC	パワーオン処理	-	void	
src¥REL_src	sbrk.c	アロケーション	sbrk	メモリエリアアロケーション処理	size_t size	_SBYTE *	

3.3 変数一覧

本制御プログラムで使用する変数一覧を次に示します。ただし、ローカル変数は記載していません。また、一部変数はユーザが容易に組み込める様、拡張用として準備している変数(本ソフトウェアでは不使用の変数)も含めております。不使用変数は、以下一覧表の Notes をご参照下さい。

表3-12 変数一覧

VariableName	Type	Size	Overview	Notes
g_f_AD0_range	float [8]	32	Range value for S12AD ch0	

RX13T シーリング FAN インバータボードによる PFC 制御、及びセンサレスベクトル制御

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/fcarrier	
g_f_ctrl_time_div_ke	float	4	Tcarrier:1/fcarrier / ke	
g_f_ctrl_time_div_id	float	4	Tcarrier:1/fcarrier / id	
g_f_ctrl_time_div_iq	float	4	Tcarrier:1/fcarrier / 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 buf	
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	

RX13T シーリング FAN インバータボードによる PFC 制御、及びセンサレスベクトル制御

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]	

RX13T シーリング FAN インバータボードによる PFC 制御、及びセンサレスベクトル制御

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	

RX13T シーリング FAN インバータボードによる PFC 制御、及びセンサレスベクトル制御

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 マクロ定義

本制御プログラムで使用するマクロ定義一覧を次に示します。一部マクロ定義はユーザが容易に組み込める様、拡張用として準備しているマクロ定義(本ソフトウェアでは不使用のマクロ定義)も含めております。不使用マクロ定義は、以下一覧表の Notes をご参照下さい。

表3-13 マクロ定義一覧

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	0hm
	OFFSET_IU	$AD_MAX * PGA_GAIN / 10$	Current offset	
	RANGE_IU	$CPU_VOLTAGE * 10 / (SHUNT_R * 9 * PGA_GAIN)$	Current range	0hm
	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	$CPU_VOLTAGE / (OPA_GAIN_IAC * SHUNT_R_IAC)$	Range value for AC current	
	RANGE_VDC	$CPU_VOLTAGE * 453.3f / 3.3f$	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	$(2.0f * \text{MATH_PI})$	$2 * \pi$	
	MATH_PI_3	$(\text{MATH_PI} / 3.0f)$	$\pi / 3$	
	MATH_TWOPI_3	$(\text{MATH_TWOPI} / 3.0f)$	$2 * \pi / 3$	No use
	MATH_TWOPI2_3	$(\text{MATH_TWOPI} * 2 / 3.0f)$	$4 * \pi / 3$	
	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	$(1.0f / 60.0f * \text{MATH_TWOPI})$	$1 / 60 * (2 * \pi)$	rpm to radian, No use
	MATH_RAD2RPM	$(1.0f / \text{MATH_TWOPI} * 60.0f)$	$1 / (2 * \pi) * 60$	radian to rpm, No use
	MATH_DEG2RAD	$(\text{MATH_TWOPI} / 360.0f)$	$(2 * \pi) / 360$	Degree to radian, No use
	MATH_SQRT_3	$(1.73205081f)$	$\sqrt{3}$	
	MATH_1div_SQRT3	$(1 / \text{MATH_SQRT_3})$	$1 / \sqrt{3}$	
CMT0 register set value (1ms speed command cycle)	r_mtr_interval_CMT0_set_int_level(level)	$(\text{ICU_IPR}[4].\text{BIT.IPR} = (\text{level}) \& 15)$	Set interrupt level for CMT0	
	r_mtr_interval_CMT0_start_timer()	$(\text{CMT.CMSTRO.BIT.STRO} = 1)$	Start operation for CMT0	
	r_mtr_interval_CMT0_stop_timer()	$(\text{CMT.CMSTRO.BIT.STRO} = 0)$	Stop operation for CMT0	No use
	r_mtr_interval_CMT0_start_int()	$(\text{ICU_IER}[3].\text{BIT.IEN4} = 1, \text{CMT0.CMCR.BIT.CMIE} = 1)$	Enable interrupt for CMT0	
	r_mtr_interval_CMT0_stop_int()	$(\text{ICU_IER}[3].\text{BIT.IEN4} = 0, \text{CMT0.CMCR.BIT.CMIE} = 0)$	Disable interrupt for CMT0	No use
	r_mtr_interval_CMT0_clear_flag()	$(\text{ICU_IR}[28].\text{BIT.IR} = 0)$	Clear interrupt flag for CMT0	
CMT1 register set value (Not used)	r_mtr_interval_CMT1_set_int_level(level)	$(\text{ICU_IPR}[5].\text{BIT.IPR} = (\text{level}) \& 15)$	Set interrupt level for CMT1	No use
	r_mtr_interval_CMT1_start_timer()	$(\text{CMT.CMSTRO.BIT.STR1} = 1)$	Start operation for CMT1	No use
	r_mtr_interval_CMT1_stop_timer()	$(\text{CMT.CMSTRO.BIT.STR1} = 0)$	Stop operation for CMT1	No use
	r_mtr_interval_CMT1_start_int()	$(\text{ICU_IER}[3].\text{BIT.IEN5} = 1, \text{CMT1.CMCR.BIT.CMIE} = 1)$	Enable interrupt for CMT1	No use
	r_mtr_interval_CMT1_stop_int()	$(\text{ICU_IER}[3].\text{BIT.IEN5} = 0, \text{CMT1.CMCR.BIT.CMIE} = 0)$	Disable interrupt for CMT1	No use

		IT. CMIE=0)		
	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	
	r_mtr_inverter_MTU34_stop_int()	(ICU. IER[0x11]. BIT. IEN2=0, MTU4. TIER. BIT. TCIEV=0)	Disable interrupt for MTU3 ch3, 4	
	r_mtr_inverter_MTU34_start_timer()	(MTU. TSTRA. BYTE = (0xC0))	Start operation for MTU3 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. CS TO = (0x01))	Start operation for MTU3 ch0	
	r_mtr_pwm_MTU0_stop_timer()	(MTU. TSTRA. BIT. CS TO &= (~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. POEOF)	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 register setting (Not used)	r_mtr_ABZ_MTU1_P33_P32_PA5	(0x00)	Select MTU3 ch1 function	No use
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_CONTINUOUS_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 - g_ss_r_mtr_AD0_offset[2]))*g_f_r_mtr_AD0_range[2]/4096.0f)	Get S12AD ch2 machining value	
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	

		$\text{fset}[3])) * g_f_r_m_tr_AD0_range[3] / 4096.0f)$		
	<code>r_mtr_ad_AD0_get_ad004()</code>	$((\text{float})((\text{int16_t})S12AD_ADDR4 - g_ss_r_mtr_AD0_of_fset[4])) * g_f_r_m_tr_AD0_range[4] / 4096.0f)$	Get S12AD ch4 machining value	
	<code>r_mtr_ad_AD0_get_ad005()</code>	$((\text{float})((\text{int16_t})S12AD_ADDR5 - g_ss_r_mtr_AD0_of_fset[5])) * g_f_r_m_tr_AD0_range[5] / 4096.0f)$	Get S12AD ch5 machining value	
	<code>r_mtr_ad_AD0_get_ad006()</code>	$((\text{float})((\text{int16_t})S12AD_ADDR6 - g_ss_r_mtr_AD0_of_fset[6])) * g_f_r_m_tr_AD0_range[6] / 4096.0f)$	Get S12AD ch6 machining value	
	<code>r_mtr_ad_AD0_get_ad007()</code>	$((\text{float})((\text{int16_t})S12AD_ADDR7 - g_ss_r_mtr_AD0_of_fset[7])) * g_f_r_m_tr_AD0_range[7] / 4096.0f)$	Get S12AD ch7 machining value	
	<code>r_mtr_ad_AD0_get_ad000_raw()</code>	S12AD_ADDR0	Get S12AD ch0 value	No use
	<code>r_mtr_ad_AD0_get_ad001_raw()</code>	S12AD_ADDR1	Get S12AD ch1 value	No use
	<code>r_mtr_ad_AD0_get_ad002_raw()</code>	S12AD_ADDR2	Get S12AD ch2 value	No use
	<code>r_mtr_ad_AD0_get_ad003_raw()</code>	S12AD_ADDR3	Get S12AD ch3 value	No use
	<code>r_mtr_ad_AD0_get_ad004_raw()</code>	S12AD_ADDR4	Get S12AD ch4 value	No use
	<code>r_mtr_ad_AD0_get_ad005_raw()</code>	S12AD_ADDR5	Get S12AD ch5 value	No use
	<code>r_mtr_ad_AD0_get_ad006_raw()</code>	S12AD_ADDR6	Get S12AD ch6 value	No use
	<code>r_mtr_ad_AD0_get_ad007_raw()</code>	S12AD_ADDR7	Get S12AD ch7 value	No use
IWDT register setting	<code>r_mtr_IWDT_refresh()</code>	IWDT_IWDTRR=0, IWDT_IWDTRR=0xFF	Refresh for IWDT	No use
	<code>r_mtr_IWDT_underflow_status()</code>	IWDT_IWDTSR.BIT.U NDFF	Get underflow status for IWDT	No use
	<code>r_mtr_IWDT_underflow_clear()</code>	IWDT_IWDTSR.BIT.U NDFF=0	Set underflow status for IWDT	No use

3.5 制御フロー（フローチャート）

3.5.1 メイン処理

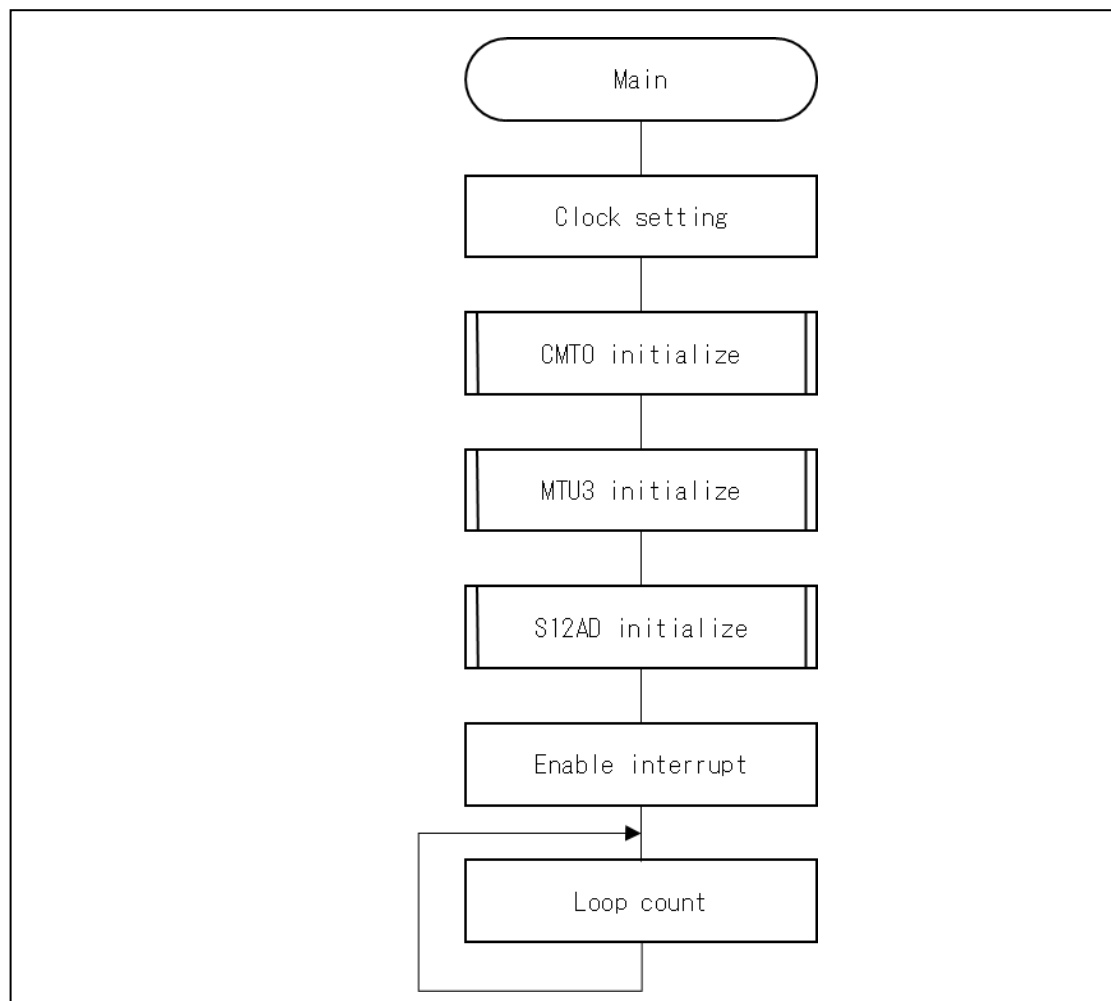


図3-8 メイン処理フローチャート

3.5.2 キャリア周期割り込み処理

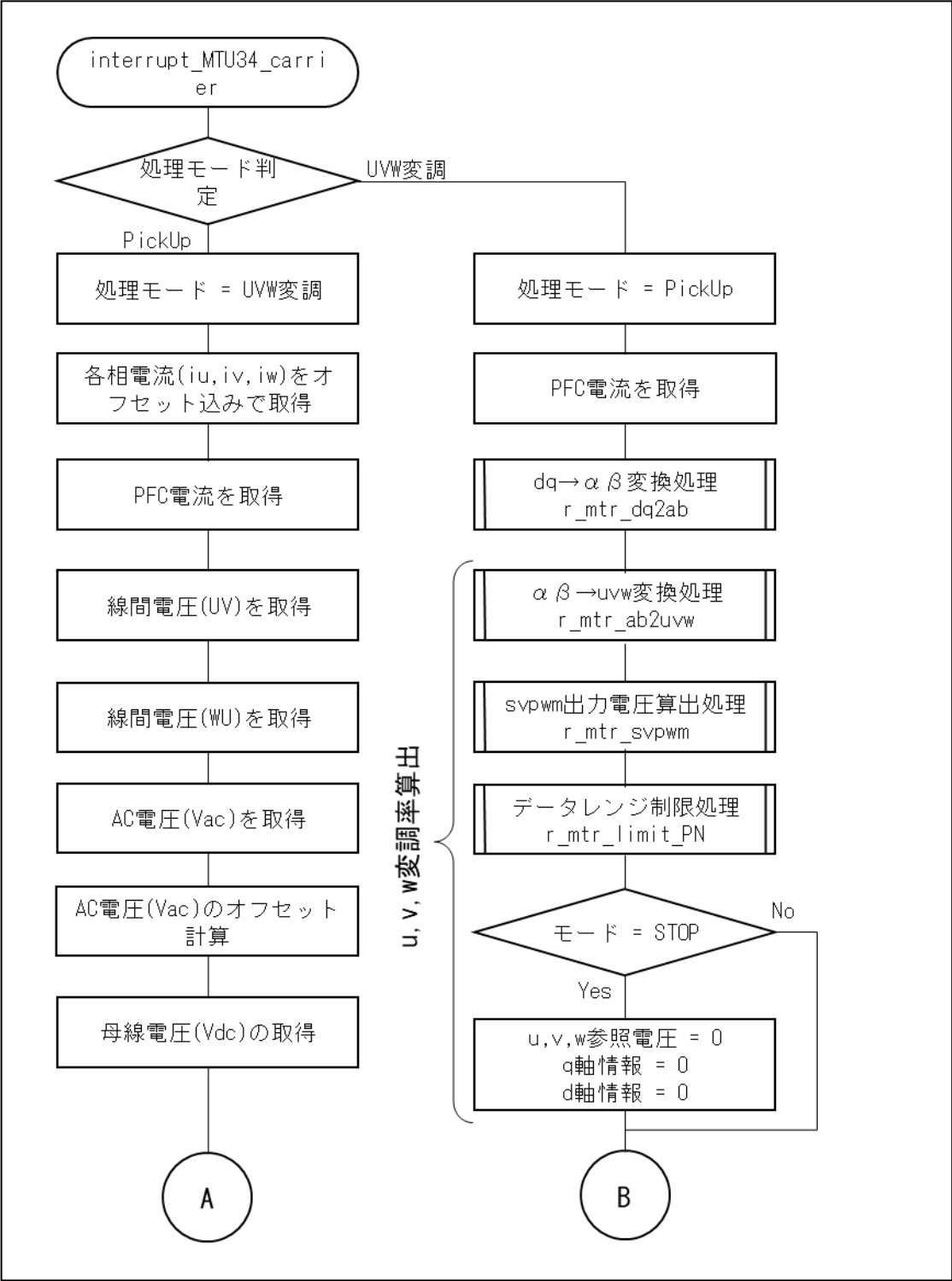


図3-9 キャリア周期割り込み処理フローチャート 1/5

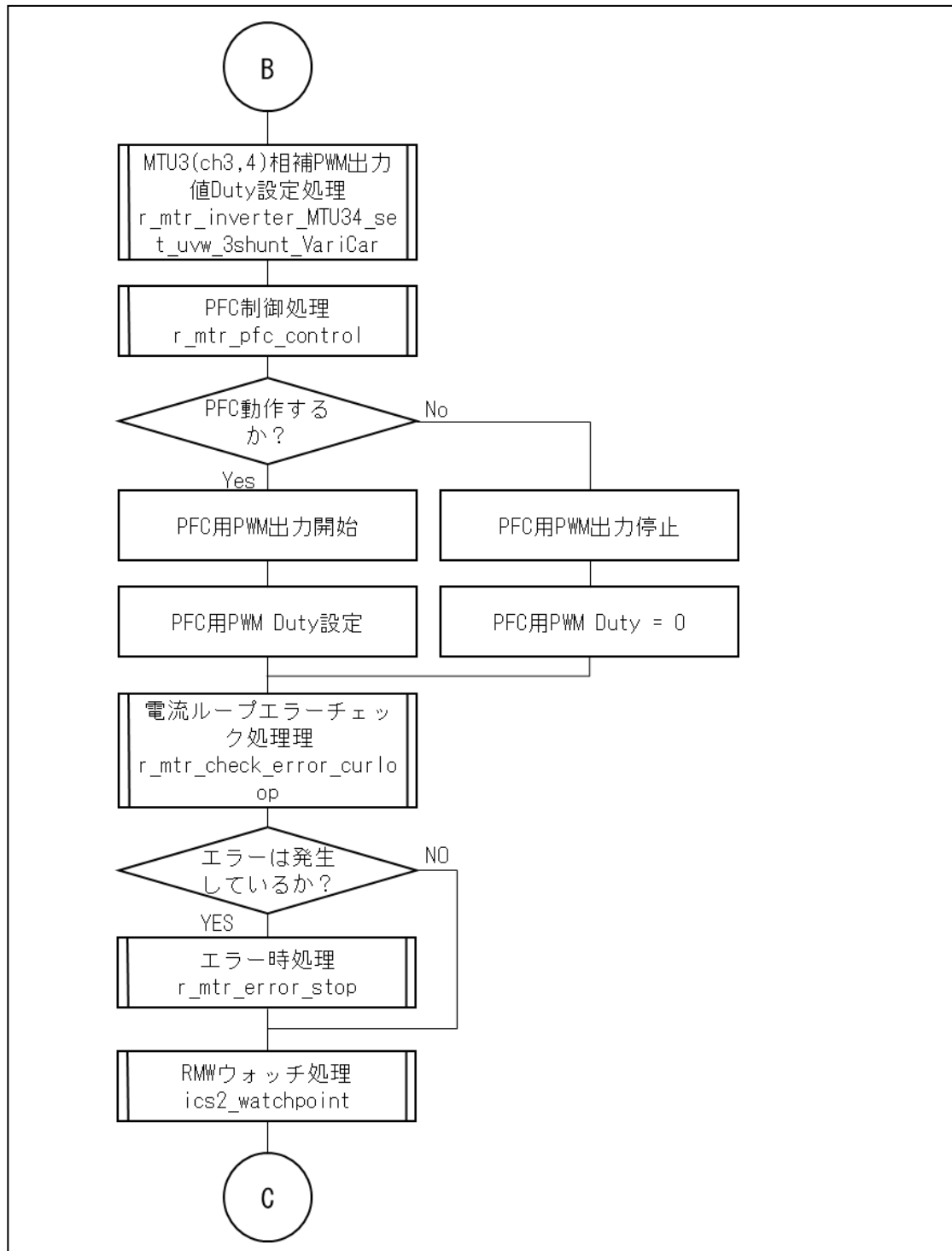


図3-10 キャリア周期割り込み処理フローチャート 2/5

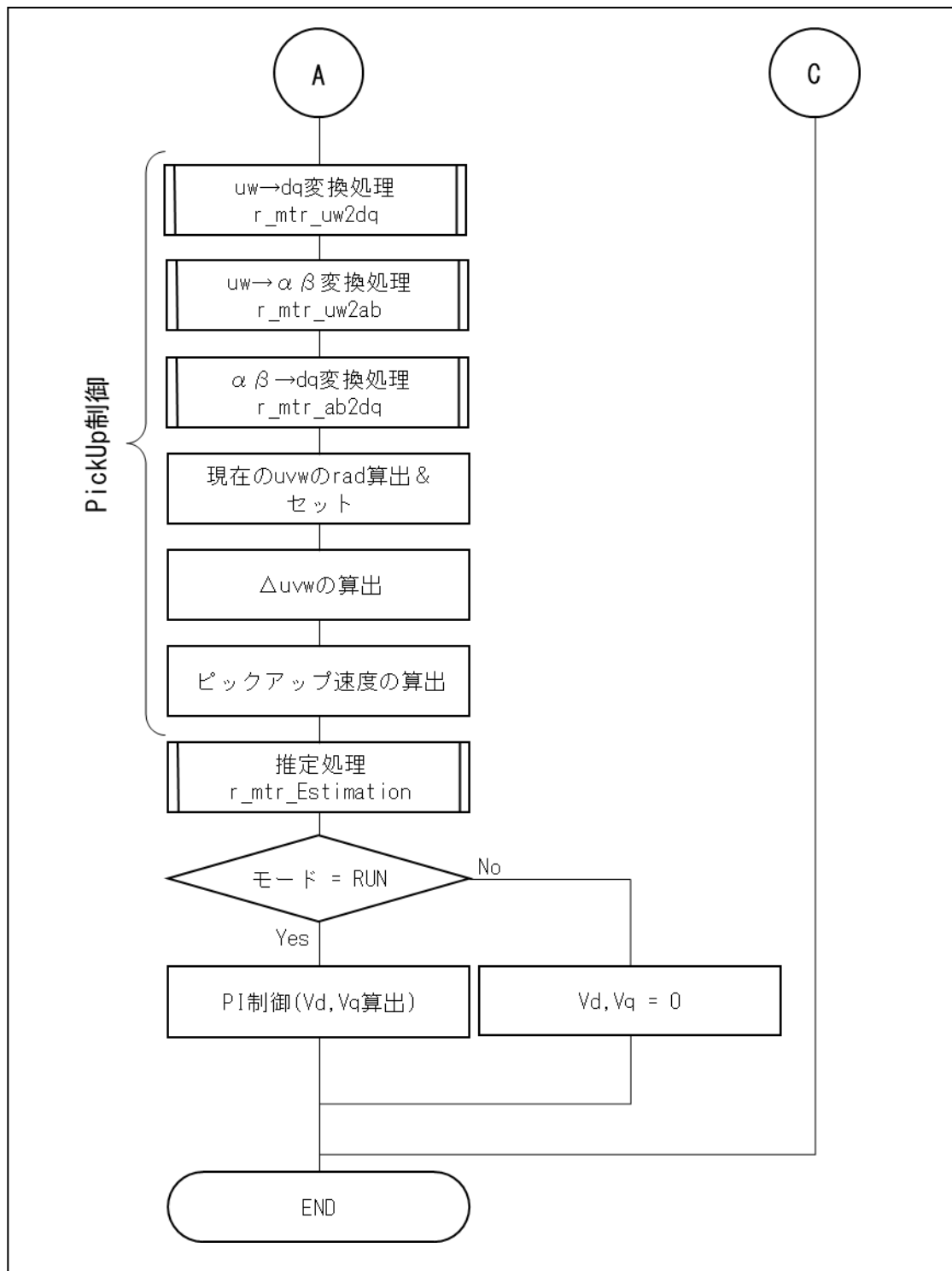


図3-11 キャリア周期割り込み処理フローチャート 3/5

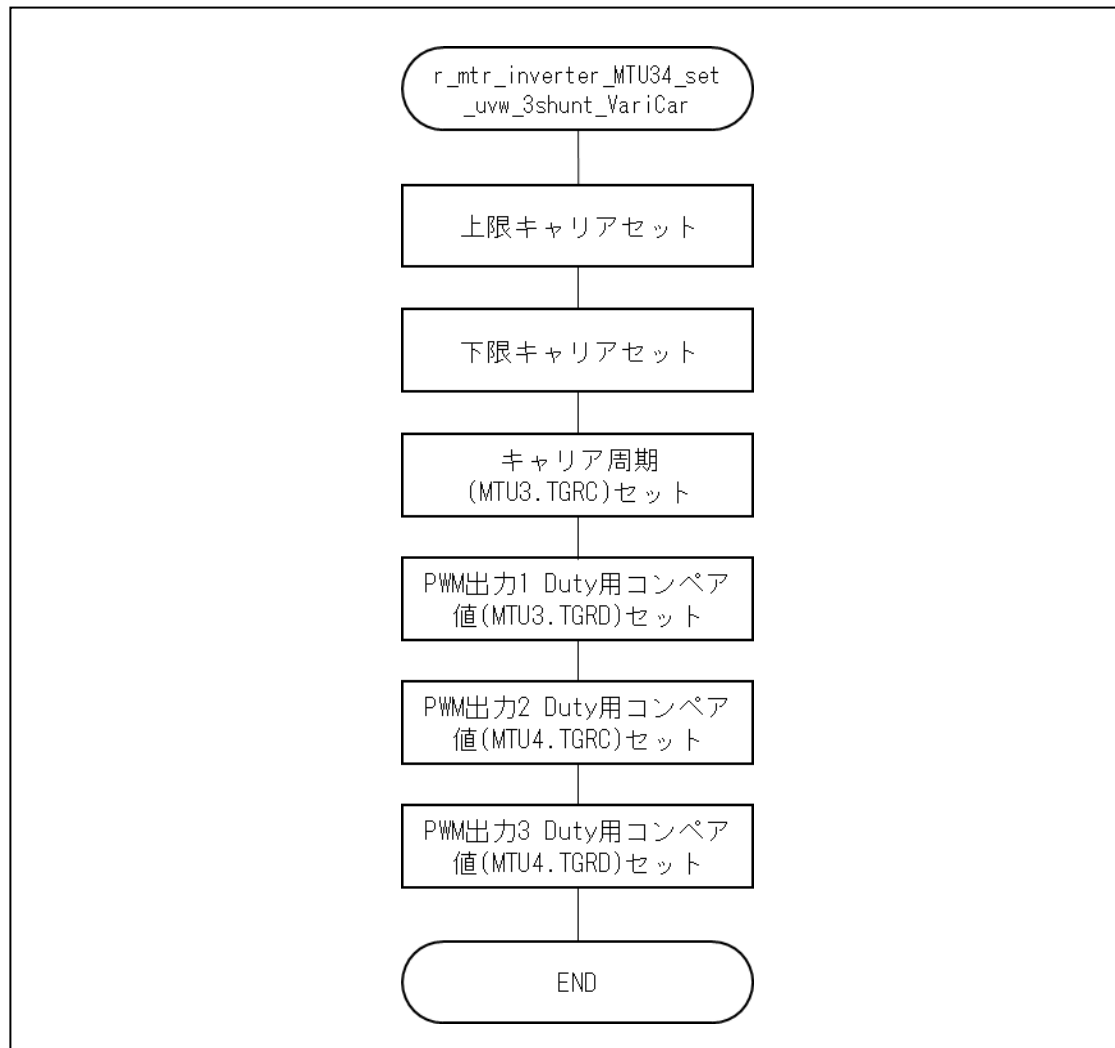


図3-12 キャリア周期割り込み処理フローチャート 4/5

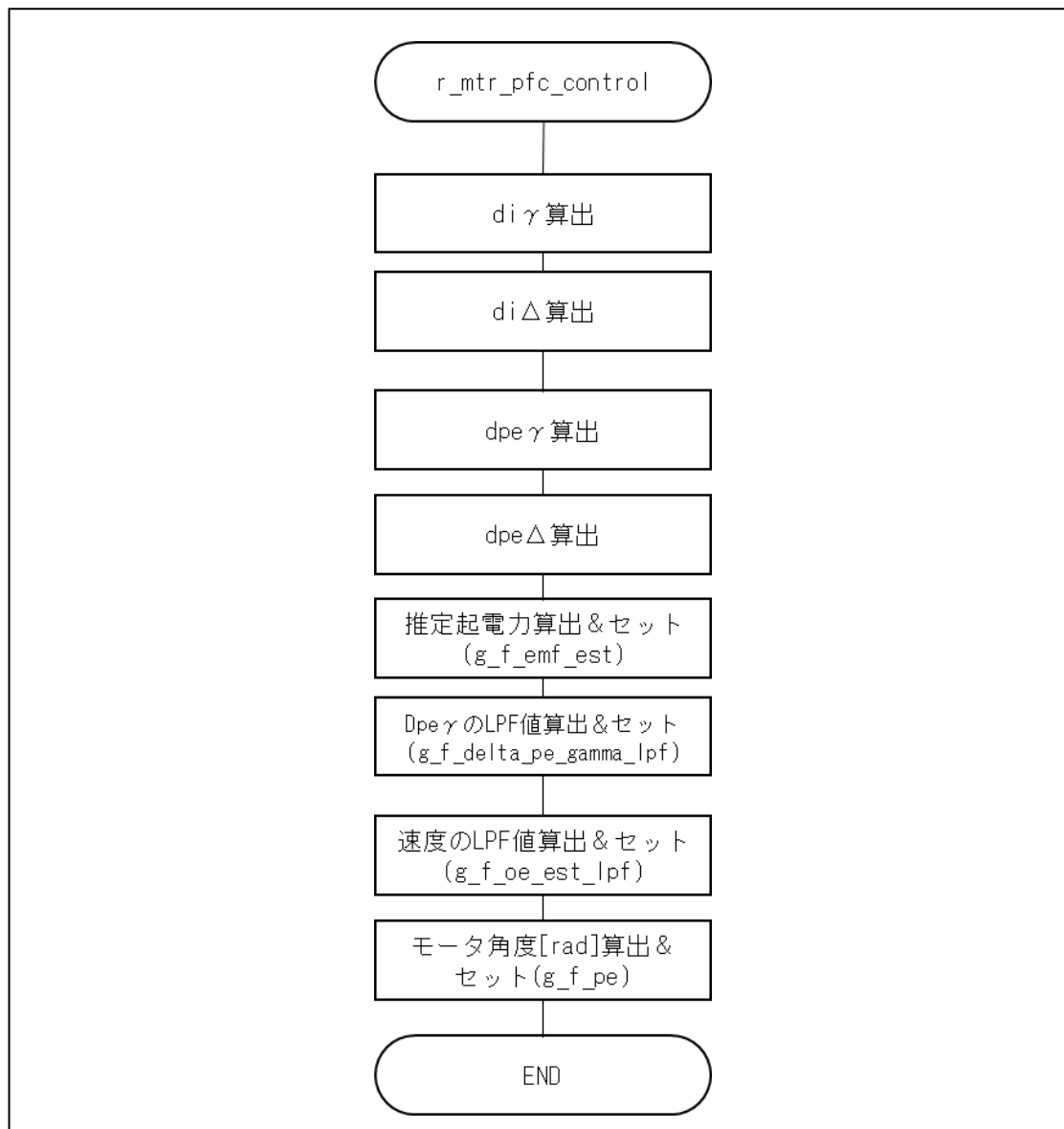


図3-13 キャリア周期割り込み処理フローチャート 5/E

3.5.3 1 [ms] 割り込み処理

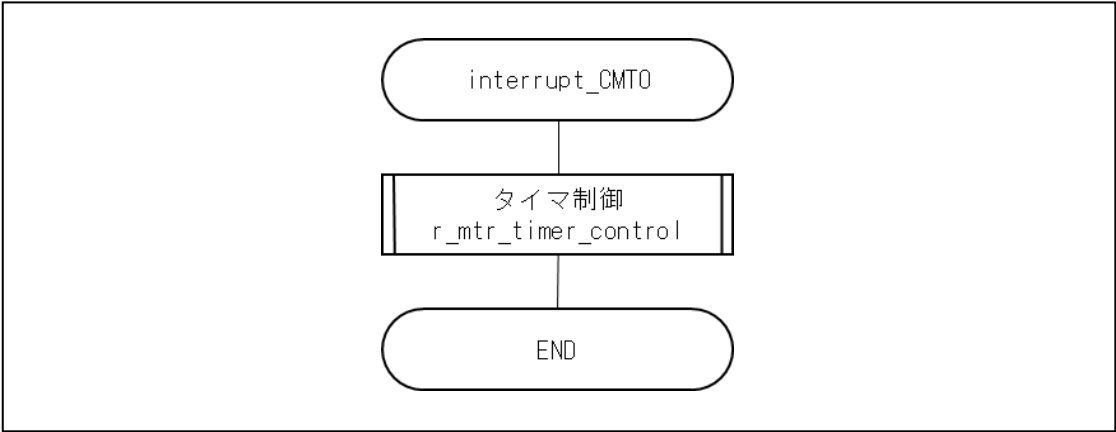


図3-14 1 [ms] 割り込み処理フローチャート 1/6

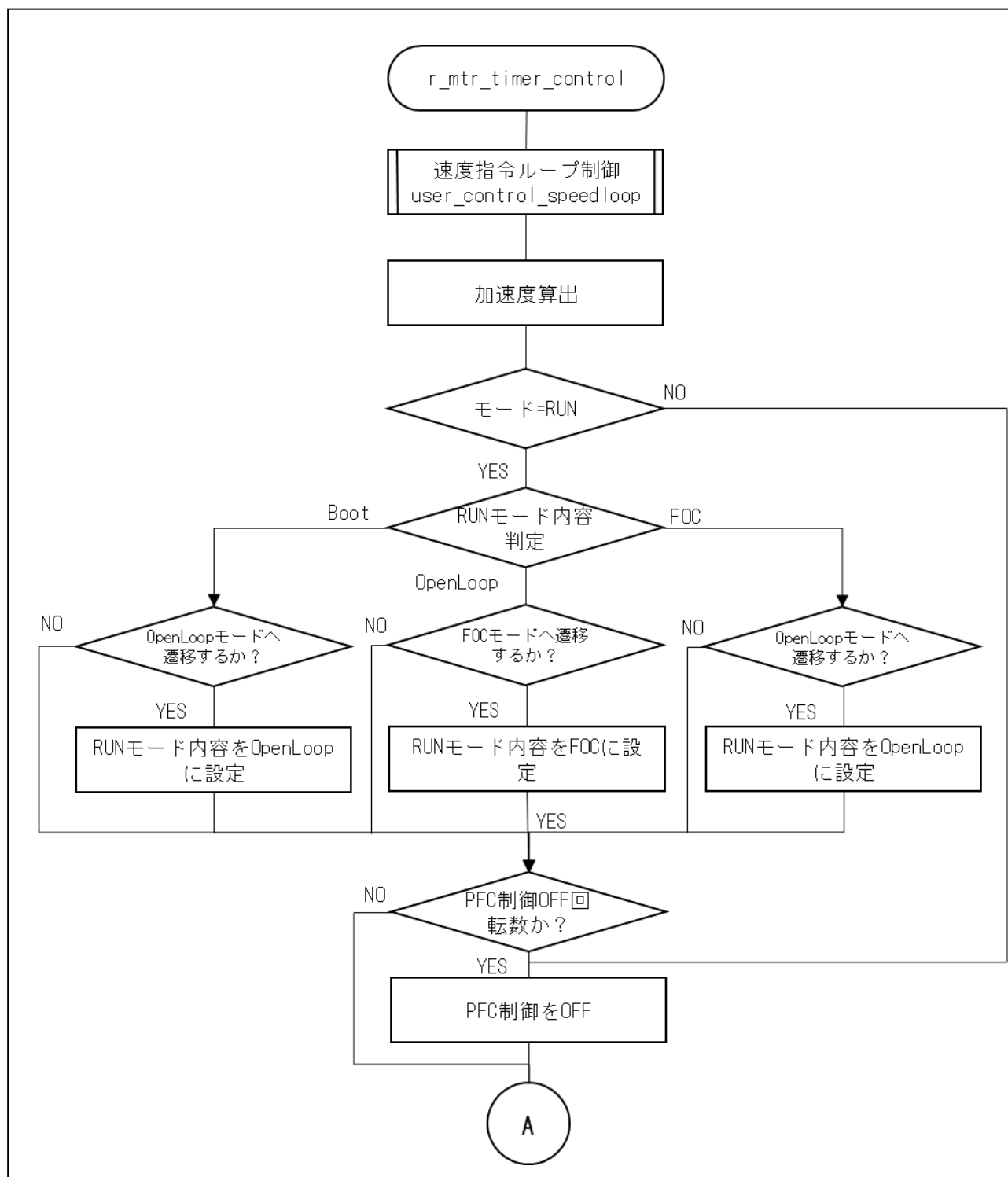


図3-15 1 [ms]割り込み処理フローチャート 2/6

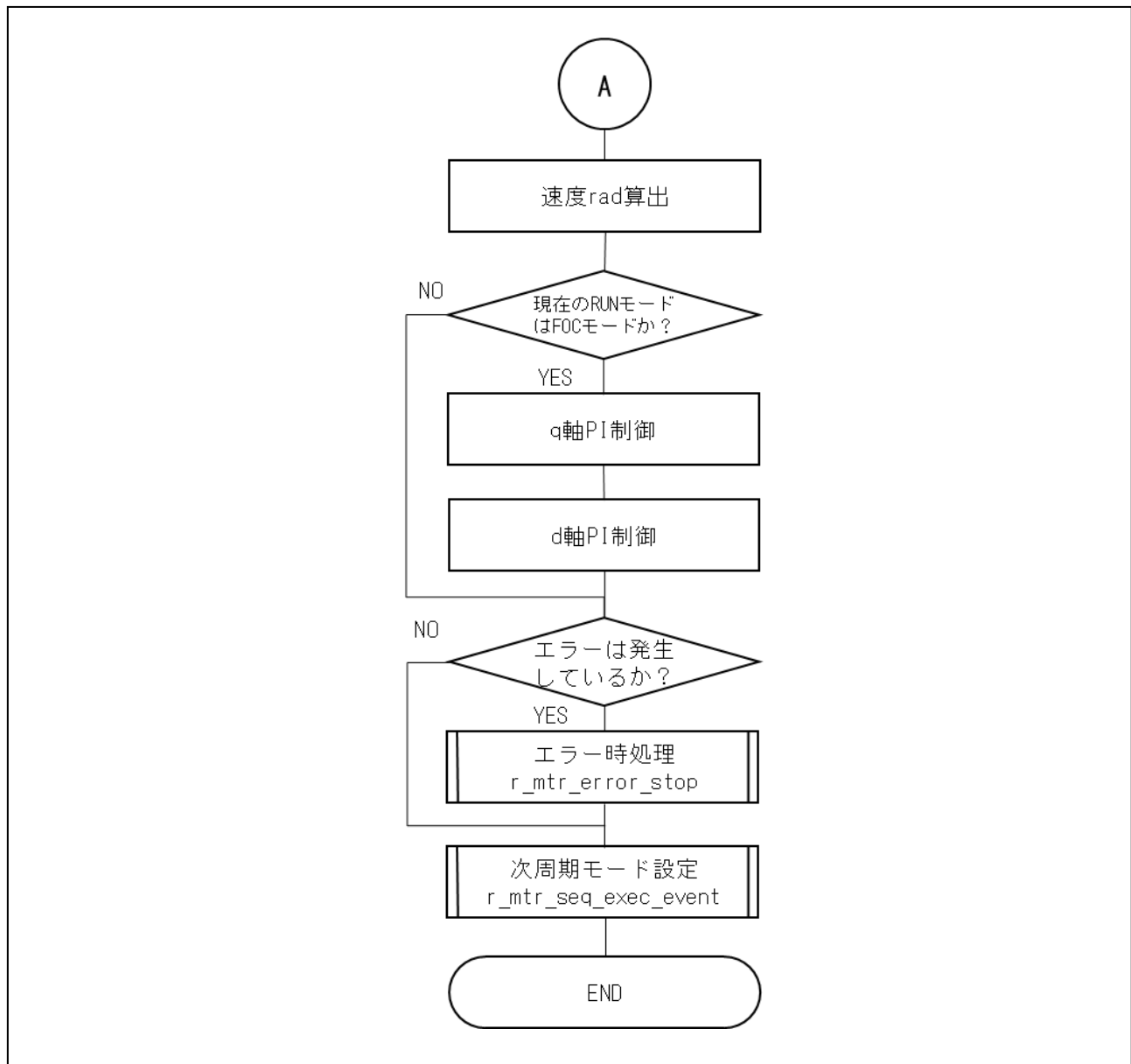


図3-16 1 [ms]割り込み処理フローチャート 3/6

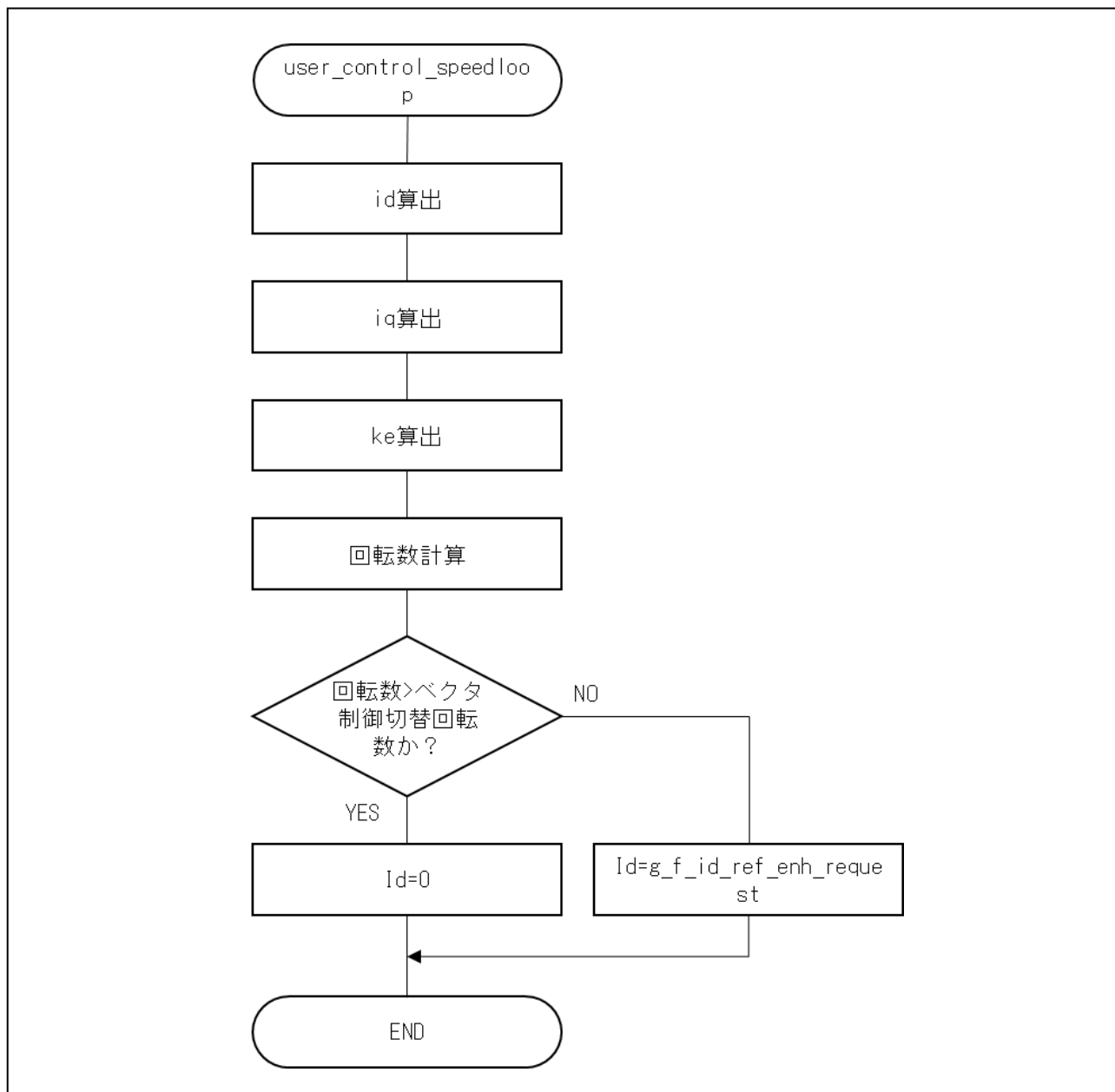


図3-17 1 [ms]割り込み処理フローチャート 4/6

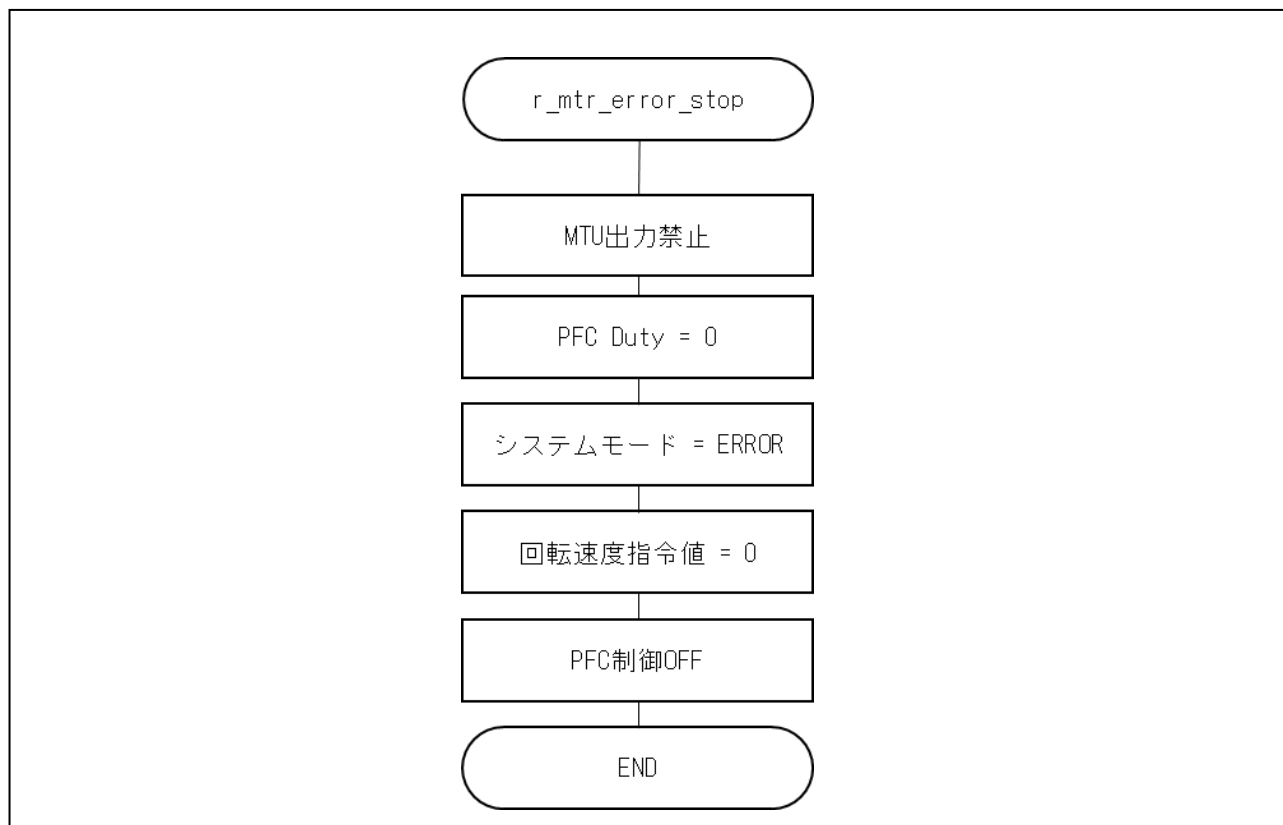


図3-18 1 [ms]割り込み処理フローチャート 5/6

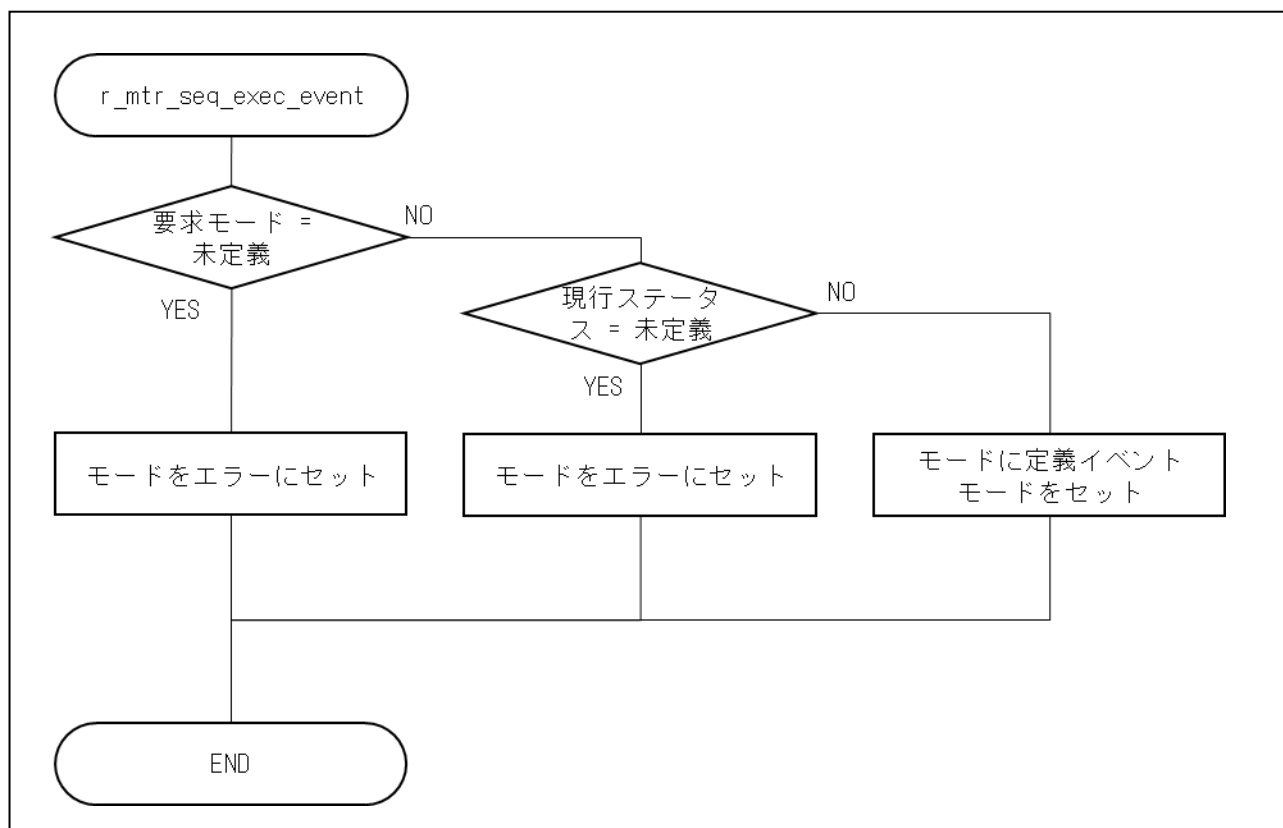
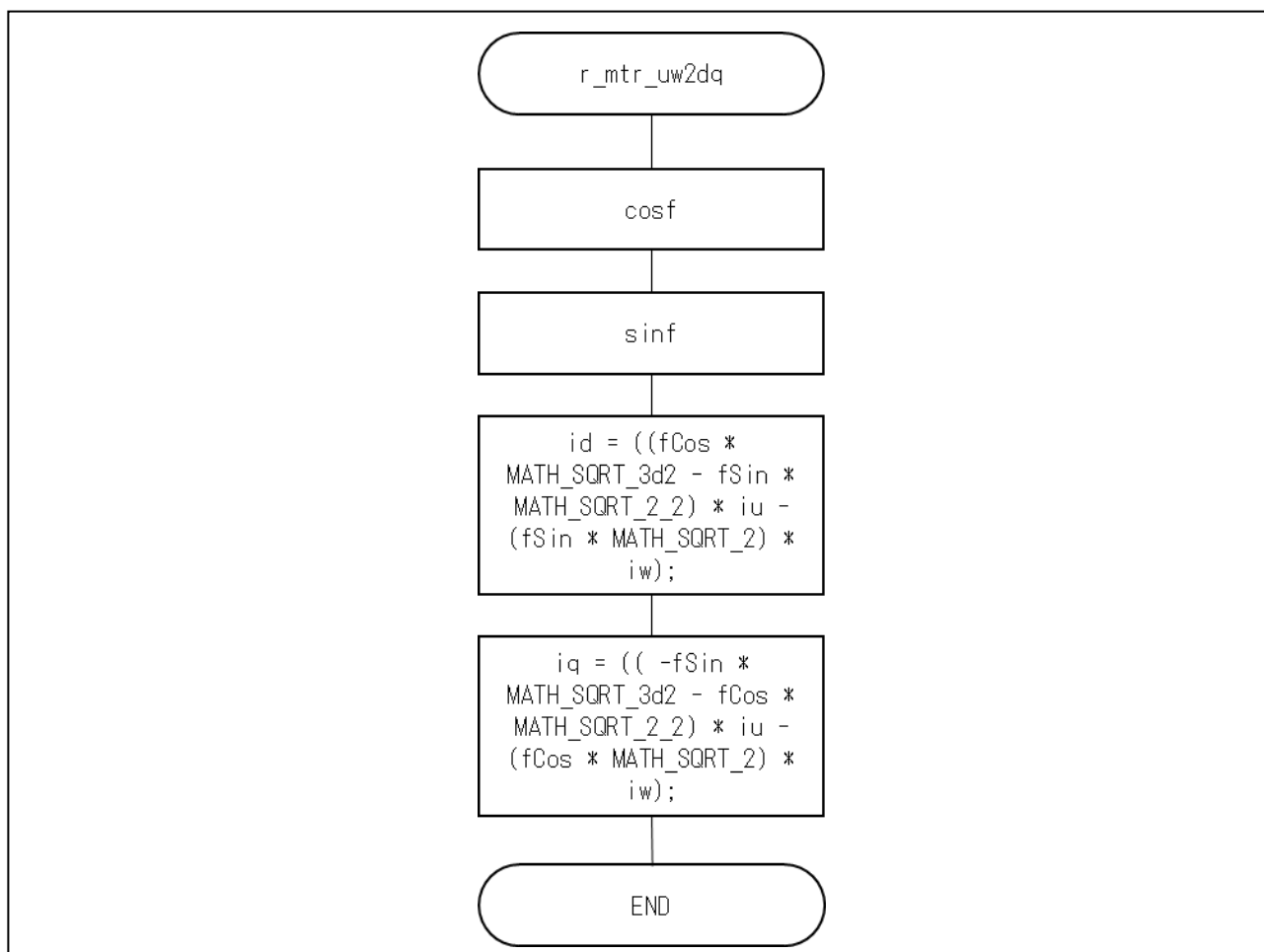
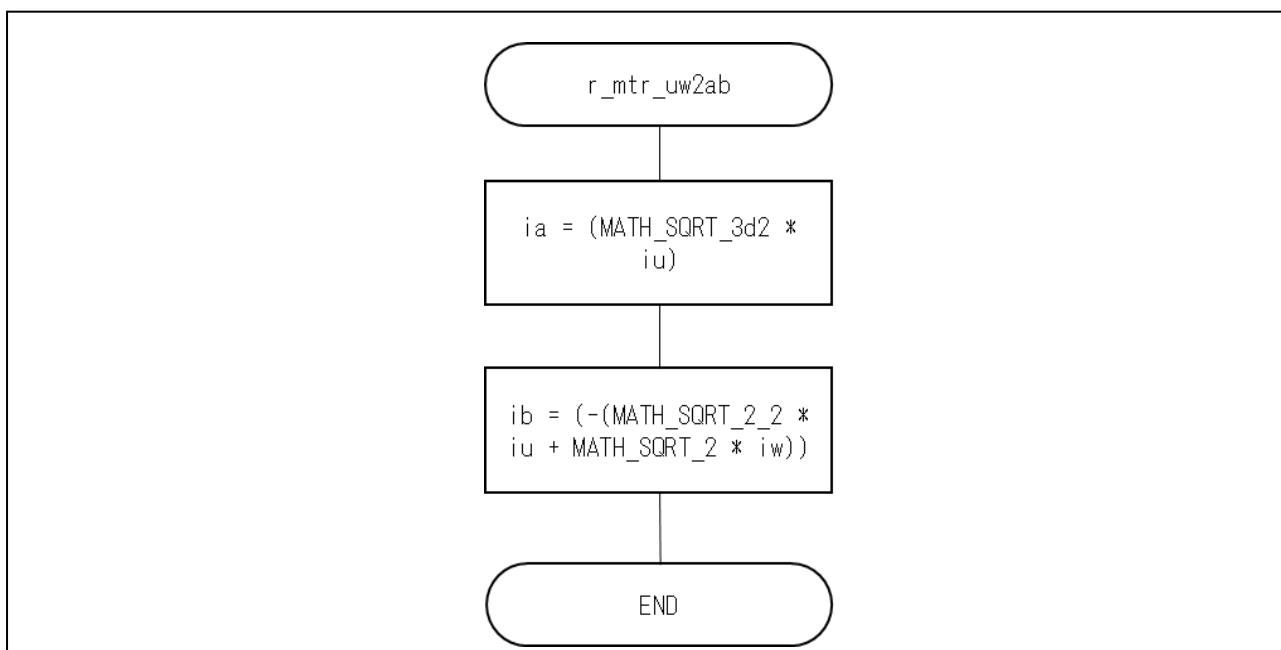


図3-19 1 [ms]割り込み処理フローチャート 6/E

3.5.4 Math ライブラリ処理

図3-20 $uw \rightarrow dq$ 変換処理フローチャート図3-21 $uw \rightarrow \alpha \beta$ 変換処理フローチャート

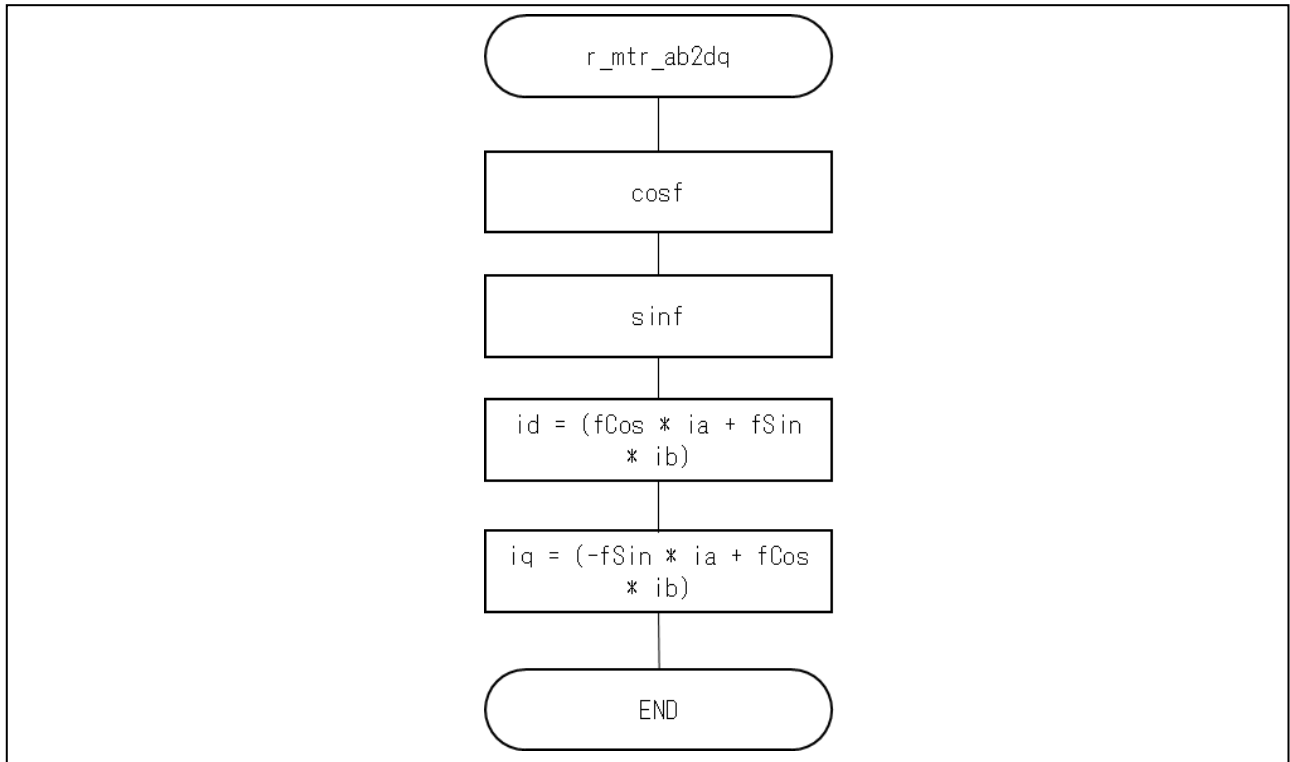


図3-22 $\alpha\beta \rightarrow dq$ 変換処理フローチャート

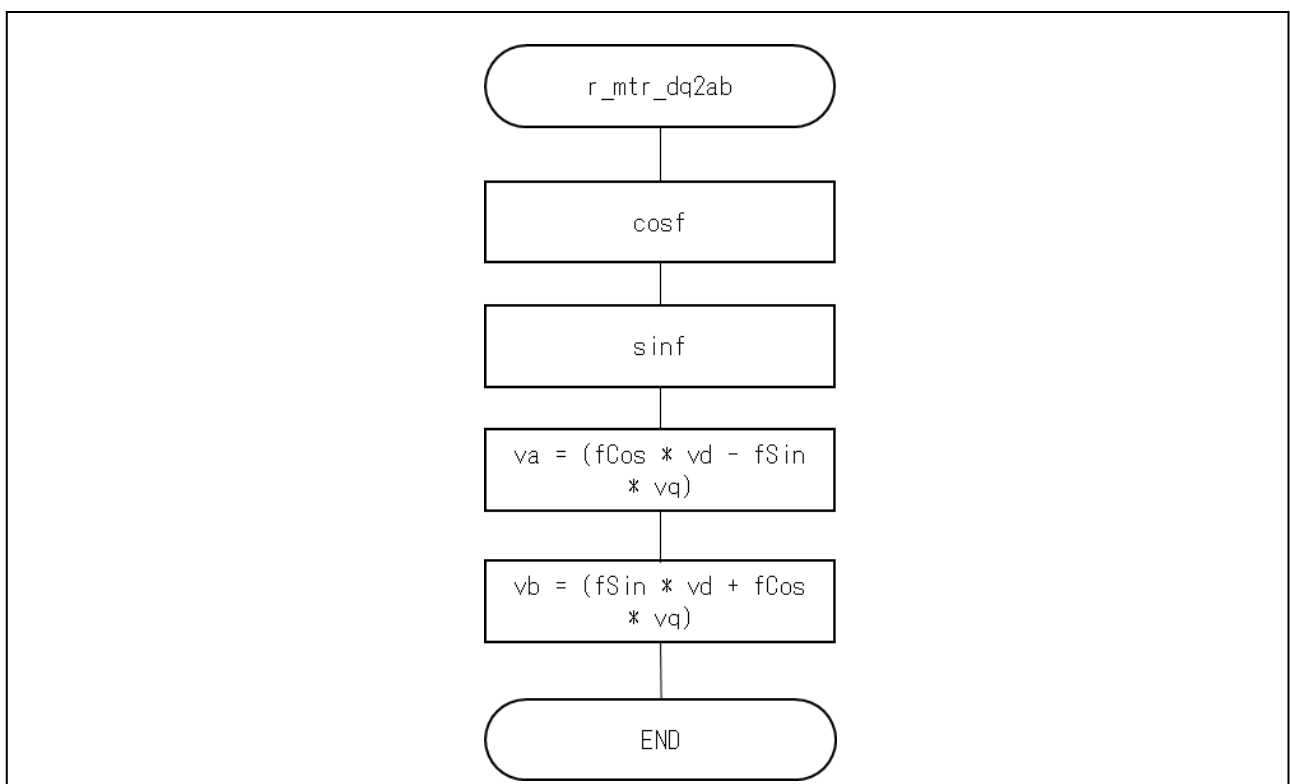


図3-23 $dq \rightarrow \alpha\beta$ 変換処理フローチャート

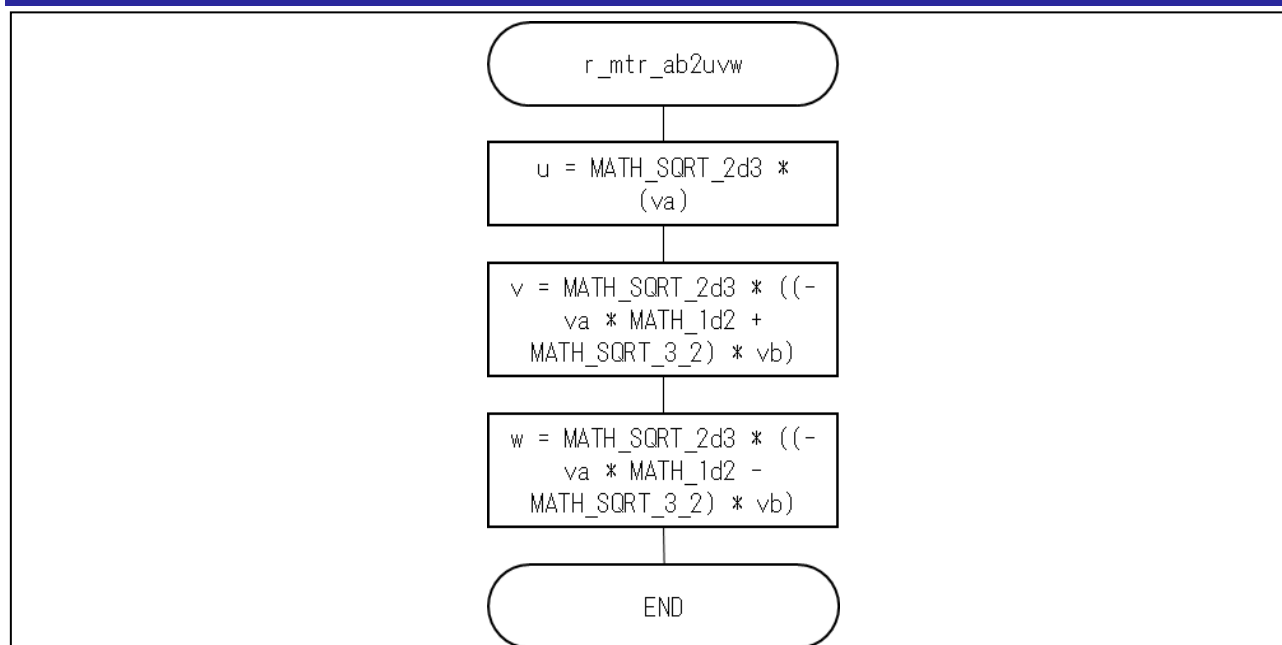
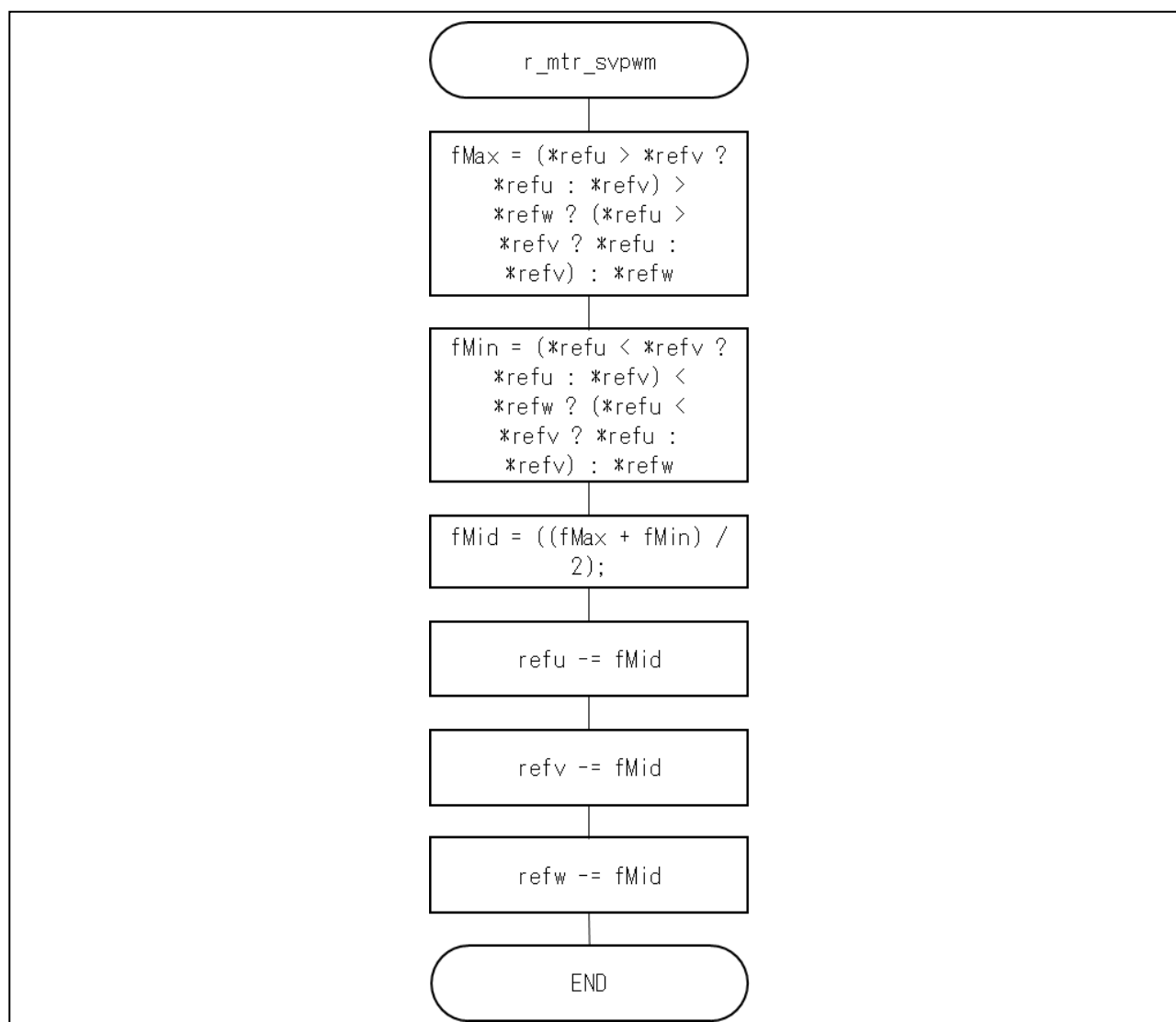
図3-24 $\alpha\beta \rightarrow uvw$ 変換処理フローチャート

図3-25 pwm 出力電圧算出処理フローチャート

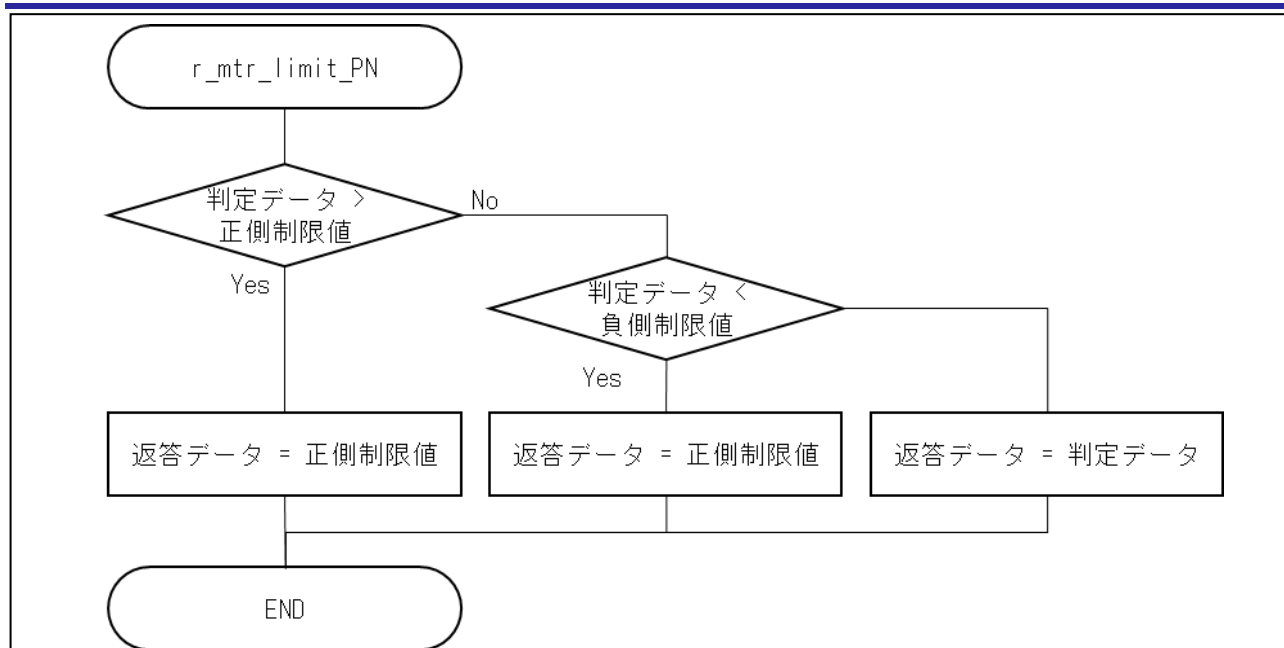


図3-26 正負データレンジ制限処理フローチャート

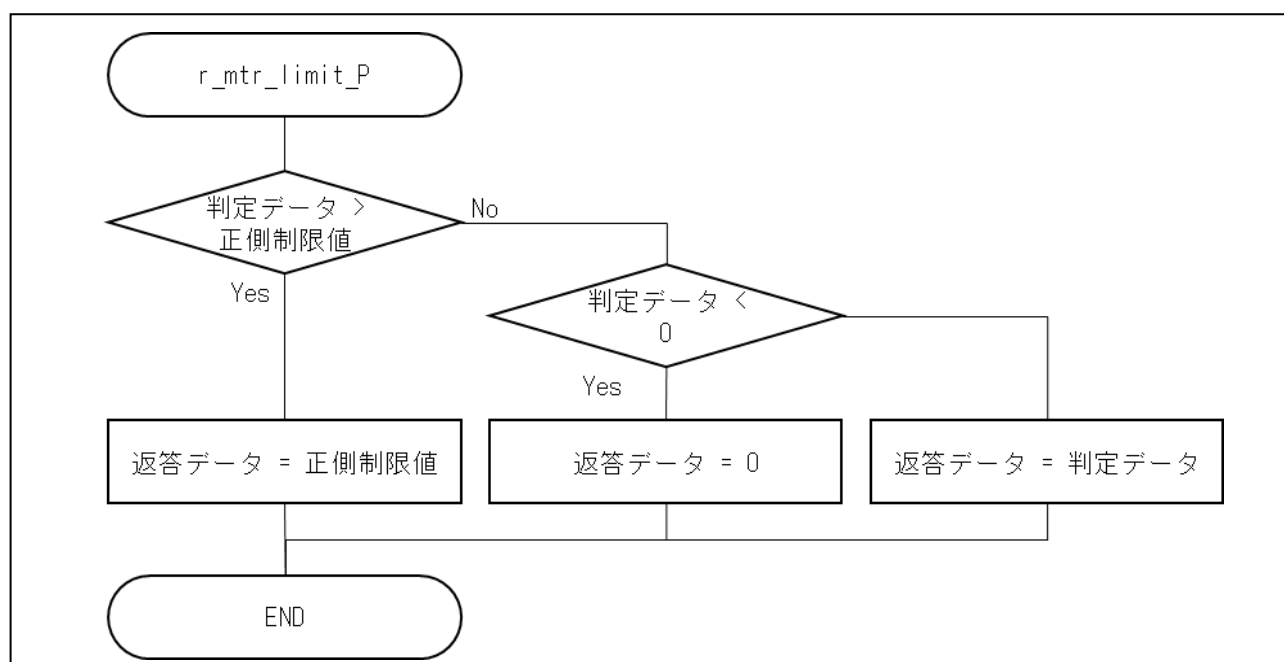


図3-27 正データレンジ制限処理フローチャート

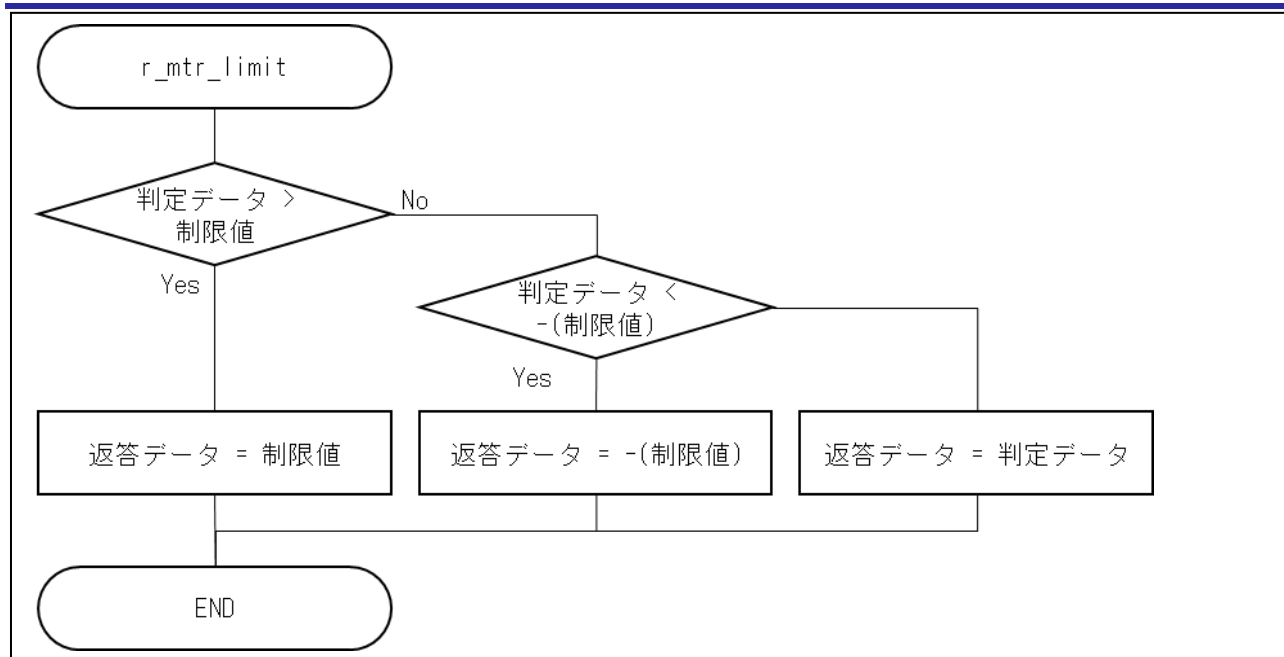


図3-28 データレンジ制限処理フローチャート

4. モータ制御開発支援ツール「Renesas Motor Workbench」

4.1 概要

本アプリケーションノート対象サンプルプログラムでは、モータ制御開発支援ツール「Renesas Motor Workbench」をユーザインタフェース（回転／停止指令、回転速度指令等）として使用します。使用方法などの詳細は「Renesas Motor Workbench ユーザーズマニュアル」を参照してください。

モータ制御開発支援ツール「Renesas Motor Workbench」は弊社 WEB サイトより入手してください。

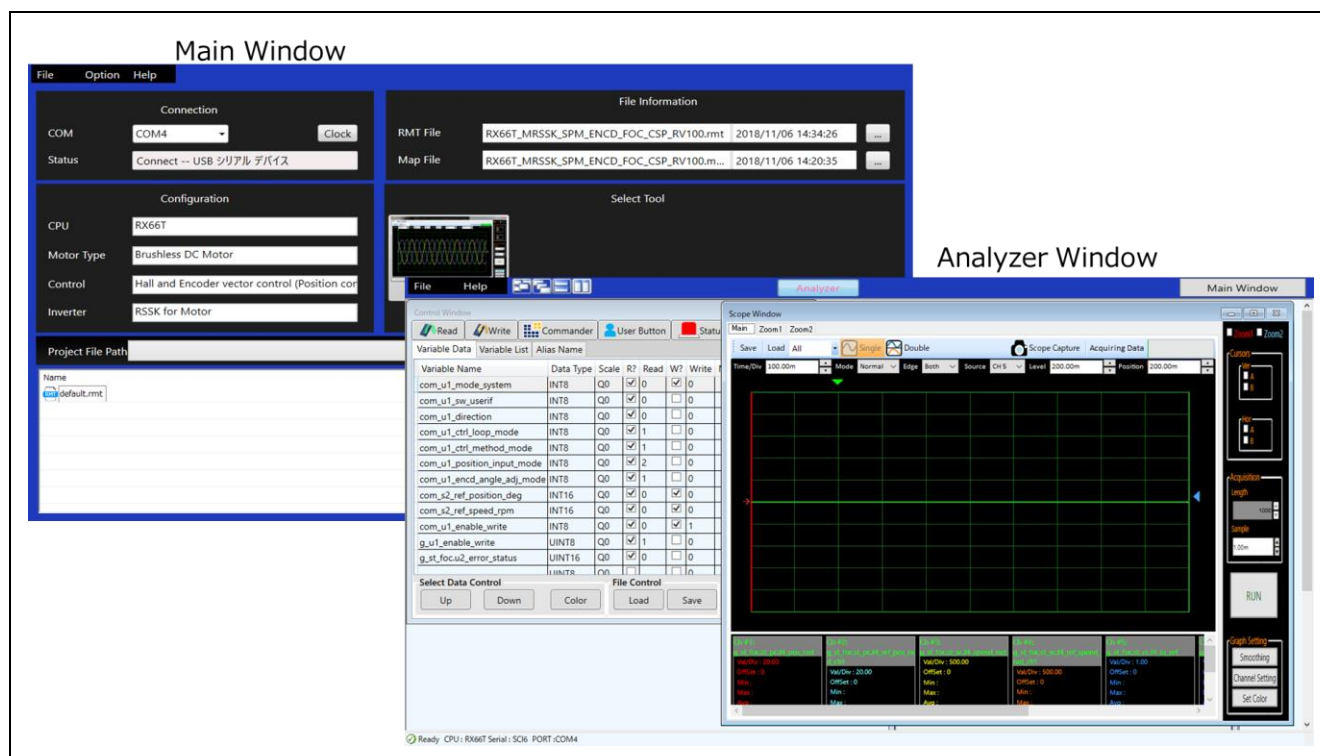


図4-1 Renesas Motor Workbench 外観

モータ制御開発支援ツール「Renesas Motor Workbench」の使い方

- ① ツールアイコン  をクリックしツールを起動する。
- ② Main Panel の MENU バーから、[RMTFile] → [Open RMT File(0)] を選択。
サンプルソフトウェアフォルダの“ics”フォルダ内にある RMT ファイルを読み込む。
- ③ “Connection” の COM で接続されたキットの COM を選択する。
- ④ Select Tool 右上の ‘Analyzer’ ボタンをクリックし、Analyzer 機能画面を表示する。
- ⑤ “4.3Analyzer機能操作例” を元にモータを駆動させる。

4.2 Analyzer 機能用変数一覧

Analyzer で使用（表示）できる変数は、表3-12 変数一覧に記載の変数を全て利用できます。本項では直接制御に関わるユーザインターフェース入力変数を表4-1に示します。以下変数は、入力値が即座に反映されます。

表4-1 Analyzer 機能入力用変数一覧

変数名	型	内容
g_us_mode_system_request	int16	ステート管理 0：ストップモード 1：ランモード 3：リセット
g_f_rpm_ref_request	int16	速度指令値（機械角）[rpm]
g_f_rpm_slope_request	float32	速度Δ指令値 [rpm/s]
g_f_rpm_oltovec	float32	オープンループ制御からベクトル制御へ切替る速度 [rpm]
g_f_rpm_enhance	float32	Idを減衰する速度 [rpm]
g_f_id_ref_ol_request	float32	オープンループ時の id 最大値 [A]

4.3 Analyzer 機能操作例

Analyzer 機能を使用し、モータを操作する例を以下に示します。操作は、“Control Window”で行います。“Control Window”の詳細は、「Renesas Motor Workbench ユーザーズマニュアル」を参照してください。モータを回転させる

- ① “g_us_mode_system_request”, “g_f_rpm_ref_request” の[W?]欄に“チェック”が入っていることを確認する。
- ② 指令回転速度を“g_f_rpm_ref_request”の[Write]欄に入力する。
- ③ “g_us_mode_system_request”の[Write]欄に“1”を入力する。
- ④ “Write”ボタンを押す。
- ⑤ “Read”ボタンを押して現在の“g_f_rpm_ref_request”の[Read]欄を確認する。

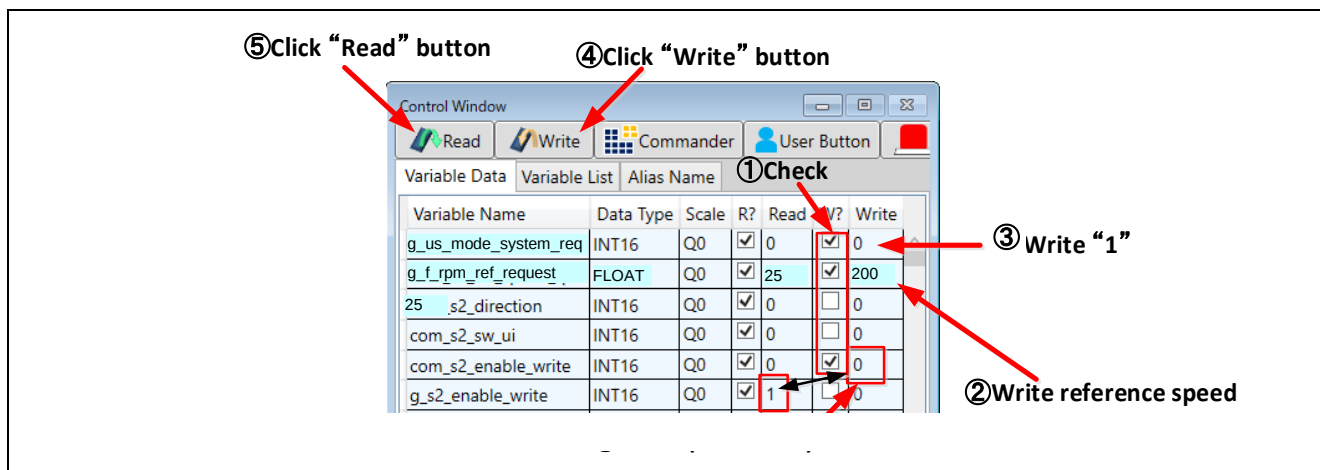


図4-2 モータ回転の手順

モータを停止させる

- ① “g_us_mode_system_request” の[Write]欄に “0” を入力する。
- ② “Write” ボタンを押す。

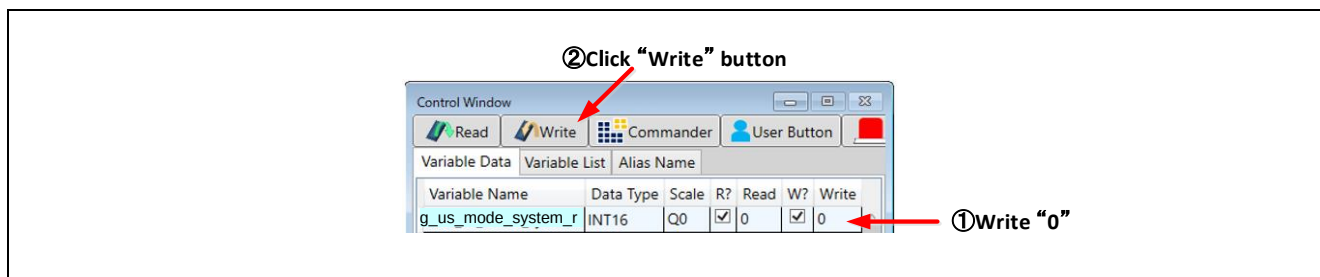


図4-3 モータ停止の手順

止まってしまった（エラー）場合の処理

- ① “com_s2_mode_system” の[Write]欄に “3” を入力する。
- ② “Write” ボタンを押す。

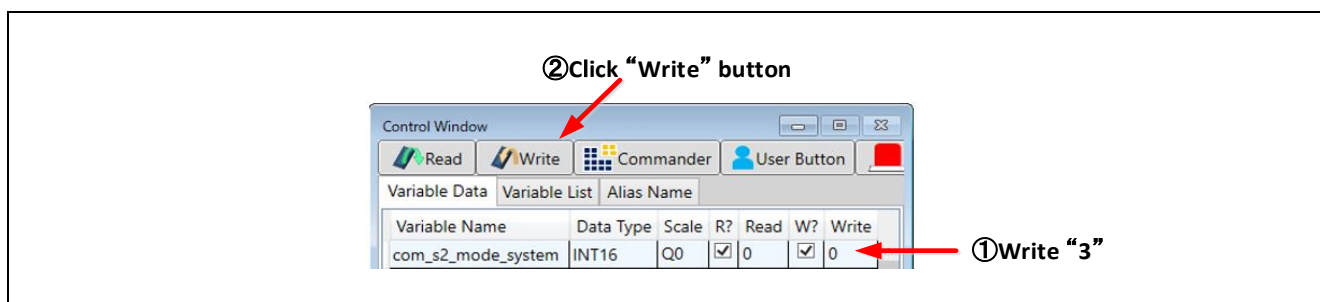


図4-4 エラー解除の手順

5. Appendix「FAQ」

■本システムについて

1. 本システムを使用する上で必要となる機材は何がありますか？
→本ボード(株式会社デスクトップラボ製 RX13T 搭載シーリング FAN インバータボード)、株式会社デスクトップラボ製 ICS ボード、ターゲットとなるモータ、及びモータ制御開発支援ツール「Renesas Motor Workbench」/株式会社デスクトップラボ製 In Circuit Scope(ICS)となります。 ソフトウェアの改造、及び動作トレースなどの確認を実施する場合は、E2Lite+統合開発環境(CS+/E2Studio)が必要となります。
2. 本システムを購入したいのですが、購入可能でしょうか？
→株式会社デスクトップラボより RX13T 搭載シーリング FAN インバータボードを購入可能です。 株式会社デスクトップラボにお問い合わせください。
3. サンプルソフトウェアをはじめ回路図など参考となる設計データは提供頂けますでしょうか？
→弊社から提供可能です。 また、株式会社デスクトップラボからも提供可能です。
4. 設計データは有償でしょうか？
→無償で提供しております。
5. 改造等を実施する上で、サポートをお願いしたいのですが？
→株式会社デスクトップラボにてコンサルティング、エンジニアリング業務を請け負っていますので、株式会社デスクトップラボにお問い合わせください。

■制御ソフトウェアについて

1. CPU 負荷率は、どの位でしょうか？
→PFC 制御(電流連続モード)+拾い上げ制御+ベクトル制御(3 シャントセンサレスベクトル制御+その他処理)で 92% (PFC:17us、拾い上げ:30us、ベクトル制御:63us)程度です。
2. PFC 制御を低周波数にする事は出来ますか？
→可能です。 PFC 制御は、MTU3 の Ch0 を使用しておりますので、カウント周波数を下げるか、カウント周期レジスタで周期設定をして下さい。 なお、ハードウェアのインダクタを含めた回路定数の見直しが必要です。
3. インバータ制御のキャリア周波数を上げたいのですが、対応可能でしょうか？
→可能です。その場合、制御間引き数を増やして頂く事になります。 現状ソフトウェアはキャリア周波数 16KHz で制御間引きを 1 回(制御周期は 8KHz)としております。
4. インバータ制御は 3 シャントの電流センサ、位置センサレスとの事ですが、1 シャントに変更する事は可能でしょうか？
→可能です。その場合、制御ソフトウェアの改造が必要です。 必要に応じて株式会社デスクトップラボにご相談ください。
5. PFC 制御アルゴリズムで Vac の Delay は、必要でしょうか？
→リアクタンス、コンデンサの定数にて位相差が発生する事がございますので、微調整の調整係数としてお使い下さい。

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
01.00	2021. 9. 21	-	新規発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレイやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力ブルアップ電源を入れないでください。入力信号や入出力ブルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因してまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
 11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
 13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/