

RL78/G1F

R01AN3991JJ0200

Rev.2.00

2020.10.20

永久磁石同期モータの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 RL78/G1F マイクロコントローラを使用し、永久磁石同期モータをセンサレスベクトル方式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 (RMW)」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上でご使用ください。

動作確認デバイス

サンプルプログラムの動作確認は以下のデバイスで行っております。

- RL78/G1F(R5F11BLEAFB)

対象サンプルプログラム

本アプリケーションノートの対象サンプルプログラムを下記に示します。

- RL78G1F_MRSSK_LESS_FOC_CSP_CC_V200 (IDE: CS+ for CC)
- RL78G1F_MRSSK_LESS_FOC_E2S_CC_V200 (IDE: e²studio)

24V Motor Control Evaluation System for RX23T と RL78/G1F CPU カード向け

RL78/G1F センサレスベクトル制御サンプルプログラム

参考資料

- RL78/G1F ユーザーズマニュアル ハードウェア編(R01UH0516JJ0110)
- 永久磁石同期モータのセンサレスベクトル制御 アルゴリズム編(R01AN3786JJ0101)
- Renesas Motor Workbench 2.0 ユーザーズマニュアル (R21UZ0004JJ0202 : Renesas-Motor-Workbench-V2-0d)
- Renesas Solution Starter Kit 24V Motor Control Evaluation System for RX23T (Motor RSSK) 取扱説明書(R20UT3697JJ0100)
- RL78/G1F CPU カード 取扱説明書 (R12UZ0014JJ0100)

目次

1. 概説.....	3
2. システム概要	4
3. 制御プログラム説明	11
4. モータ制御開発支援ツール「Renesas Motor Workbench」の利用方法	53

1. 概説

本アプリケーションノートでは、RL78/G1F 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)のセンサレス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「Renesas Motor Workbench」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「永久磁石同期モータのセンサレスベクトル制御 アルゴリズム編」のアルゴリズム一部を使用しています。

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
RL78/G1F (R5F11BLEAFB)	24V 系インバータボード ^{注1} & RL78/G1F CPU カード ^{注2}	TSUKASA ^{注3} TG-55L

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 V8.03.00	ツールチェイン CC-RL V1.09.00
e ² studio バージョン 2020-07	ツールチェイン CC-RL V1.09.00

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下さい。

【注】

- 24V 系インバータボード(RTK0EM0001B00012BJ)は、ルネサスエレクトロニクス株式会社の製品です。
- RL78/G1F CPU カードは、以下の二種類を使用可能です。
RTK0ELML240C03000BJ：ルネサスエレクトロニクス株式会社製
T5103：株式会社デスクトップラボ (<http://desktoplab.co.jp/>)製
- TG-55L は、ツカサ電工株式会社の製品です。
ツカサ電工株式会社 (<http://www.tsukasa-d.co.jp/>)

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 ハードウェア構成

ハードウェア構成を次に示します。

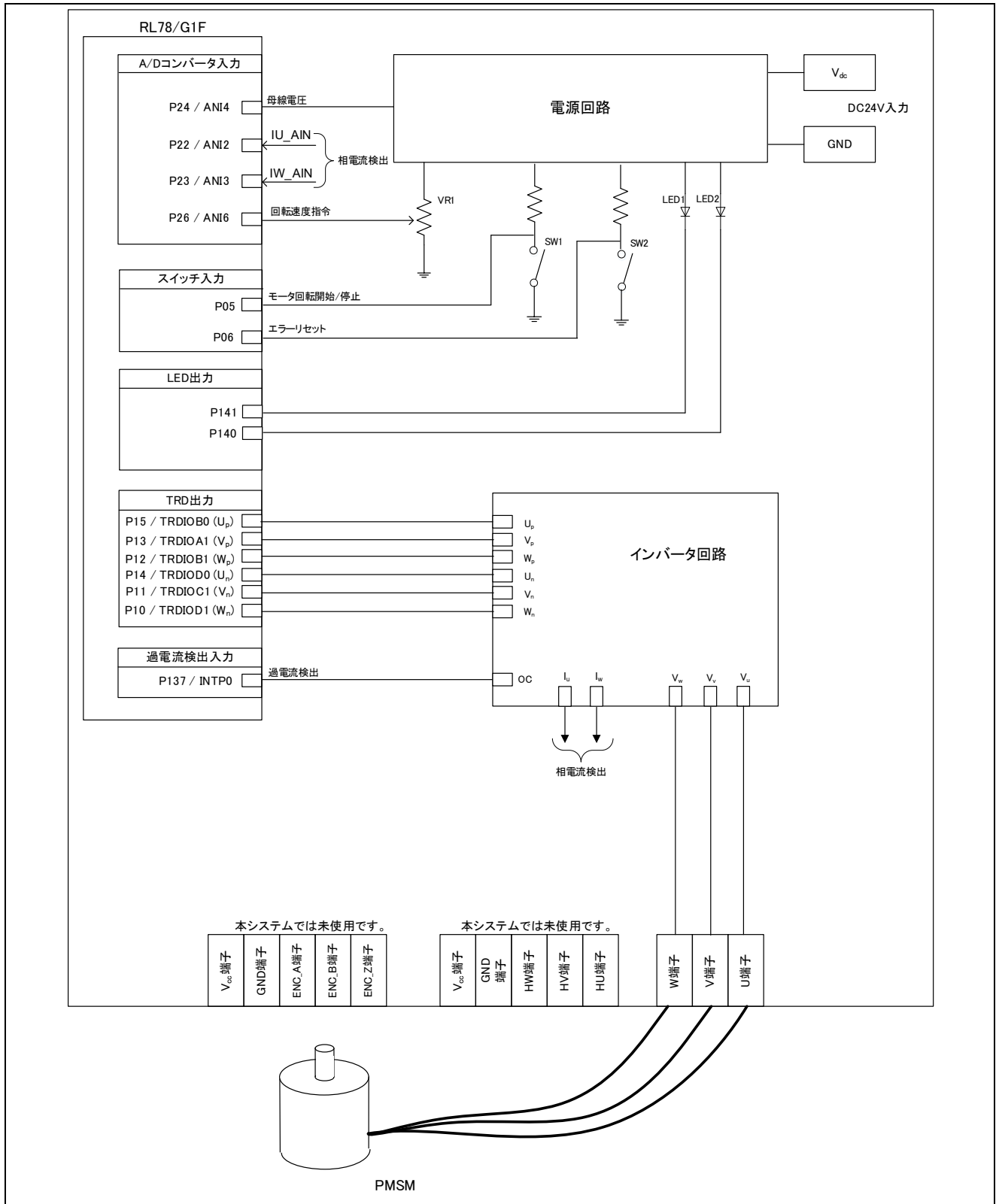


図 2-1 ハードウェア構成図

2.2 ハードウェア仕様

2.2.1 ユーザインタフェース

本システムのユーザインタフェース一覧を表 2-1 に示します。

表 2-1 ユーザインタフェース

項目	インタフェース部品	機能
VR1	可変抵抗	回転速度指令値入力(アナログ値)
SW1	トグルスイッチ	モータ回転開始/停止指令
SW2	トグルスイッチ	エラー状態からの復帰指令
LED1	黄緑色 LED	<ul style="list-style-type: none"> モータ回転時：点灯 停止時：消灯
LED2	黄緑色 LED	<ul style="list-style-type: none"> エラー検出時：点灯 通常動作時：消灯
LED3	黄緑色 LED	本システムでは未使用
RESET	プッシュスイッチ	システムリセット

本システムの RL78/G1F マイクロコントローラ端子のインタフェース一覧を表 2-2 に示します。

表 2-2 端子インタフェース

R5F11BLEAFB 端子名	機能
P24 / ANI4	インバータ母線電圧測定
P26 / ANI6	回転速度指令値入力用(アナログ値)
P05	START/STOP トグルスイッチ
P06	ERROR/RESET トグルスイッチ
P141	LED1 点灯/消灯制御
P140	LED2 点灯/消灯制御
P04	LED3 点灯/消灯制御(本システムでは未使用)
P22 / ANI2	U 相電流検出
P23 / ANI3	W 相電流検出
P15 / TRDIOB0	PWM 出力(U_p)
P13 / TRDIOA1	PWM 出力(V_p)
P12 / TRDIOB1	PWM 出力(W_p)
P14 / TRDIOD0	PWM 出力(U_n)
P11 / TRDIOC1	PWM 出力(V_n)
P10 / TRDIOD1	PWM 出力(W_n)
P137 / INTPO	過電流検出時の PWM 緊急停止入力(外部検出回路使用)

2.2.2 周辺機能

本システムに使用する周辺機能一覧を表 2-3 に示します。

表 2-3 サンプルプログラム別周辺機能対応表

周辺機能	用途
10bit A/D コンバータ(AD)	<ul style="list-style-type: none"> 回転速度指令値入力 インバータ母線電圧測定 U、W 相電流測定 (V 相電流は U、W 相電流から算出)
タイマ・アレイ・ユニット(TAU)	<ul style="list-style-type: none"> 1 [ms] インターバル・タイマ 制御周期タイマ
タイマ RD(TRD)	相補 PWM モード使用の PWM 出力
外部割り込み(INTP0)	過電流検出

(1) 10bit A/D コンバータ(AD)

回転速度指令値入力、U 相電流(Iu)、W 相電流(Iw)、及びインバータ母線電圧(Vdc)を「10bit A/D コンバータ」を使用して測定します。

A/D 変換はチャンネル選択モードを「セレクト・モード」に、変換動作モードを「ワンショット変換モード」に設定します。(ソフトウェアトリガを使用)。

(2) タイマ・アレイ・ユニット(TAU)

a. 1 [ms] インターバル・タイマ

タイマ・アレイ・ユニットのチャンネル 0 を、1 [ms]インターバル・タイマとして使用します。

b. 制御周期タイマ

タイマ・アレイ・ユニットのチャンネル 1 を使用。
100[μs]インターバル・タイマとして使用します。

(3) タイマ RD(TRD)

相補 PWM モードを使用して、デッドタイム付きの 6 相 PWM 出力を行います。

また、パルス出力強制遮断機能を用いて、過電流検出時は PWM 出力端子をハイインピーダンス出力とします。

(4) 外部割り込み(INTP0)

外部回路による過電流検出を行います(パルス出力強制遮断機能使用)。

2.3 ソフトウェア構成

2.3.1 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 2-4 に記します。

表 2-4 サンプルプログラムのフォルダとファイル構成

フォルダ		ファイル	内容
config		r_mtr_config.h	コンフィギュレーション定義
		r_mtr_motor_parameter.h	モータパラメータ定義
		r_mtr_control_parameter.h	制御パラメータ定義
		r_mtr_inverter_parameter.h	インバータパラメータ定義
		r_mtr_scaling_parameter.h	スケーリングパラメータ定義
application	main	main.h	メイン関数
		main.c	
	board	r_mtr_board.h	ハードウェア UI 関数定義
		r_mtr_board.c	
	ics	r_mtr_ics.h	Analyzer ^{注1} UI 関連関数定義
		r_mtr_ics.c	
		ICS_define.h	RMW の CPU 定義
		RL78G1F_vector.c	RMW の割り込みベクタ定義
		ics2_RL78G1F.h	RMW 通信用関数宣言
		ics_RL78G1F_Lx.h	RMW 通信用 MCU シリアル通信定義
ICS2_RL78G1F.obj		RMW 通信用ライブラリ	
driver	auto_generation	cstart.asm	自動生成ファイル
		hdwinit.asm	
		stkinit.asm	
		iodefine.h	
		r_mtr_rl78g1f.h	MCU 固有関数定義
		r_mtr_rl78g1f.c	
middle		r_dsp_cc.h	DSP 定義
		R_dsp_rl78_CC.lib	
		r_mtr_common.h	共通定義
		r_mtr_parameter.h	モータ制御パラメータ定義
		r_mtr_sc_table.h	三角関数テーブル
		r_mtr_ctrl_gain.obj	ゲイン設計関数定義
		r_mtr_driver_access.h	ドライバアクセス関数定義
		r_mtr_driver_access.c	
		r_mtr_statemachine.h	ステートマシン関数定義
		r_mtr_statemachine.c	
		r_mtr_foc_less_speed.h	センサレスベクトル制御関連関数定義
		r_mtr_foc_less_speed.c	
		r_mtr_interrupt.c	割り込みハンドラ関数定義
	r_mtr_est_phase_err.h	軸誤差の推定処理関数定義	
	r_mtr_est_phase_err.obj		

注 1 : モータ制御開発支援ツール「Renesas Motor Workbench」の Analyzer 機能の詳細については、4 章を参照下さい。

また、モータ制御開発支援ツール「Renesas Motor Workbench(RMW)」に関わるフォルダ、ファイル、関数、変数の名前には識別子「ics / ICS(旧ルネサス製モータ制御開発支援ツール「In Circuit Scope」の略)」が付加されている場合があります。

2.3.2 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 2-2 に示します。

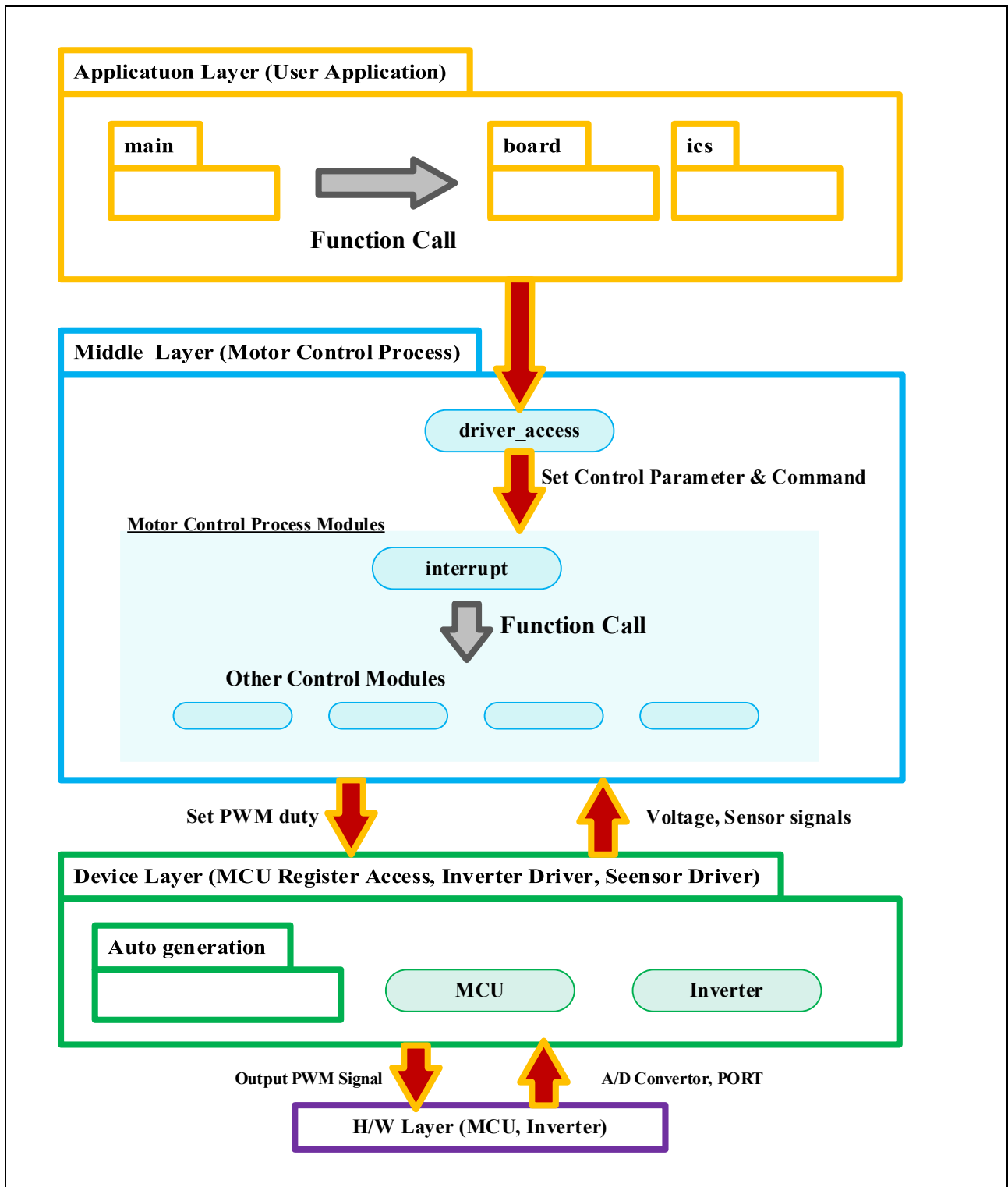


図 2-2 サンプルプログラムのモジュール構成

2.4 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下記に示します。センサレスベクトル制御の詳細に関しては「永久磁石同期モータのセンサレスベクトル制御 アルゴリズム編」を参照してください。

表 2-5 ソフトウェア仕様

項目	内容	
制御方式	ベクトル制御	
モータ回転開始/停止	SW1(P05)のレベルにより判定("Low": 回転開始、"High": 停止) もしくはモータ制御開発支援ツールによる操作 ^注	
回転子磁極位置検出	センサレス	
入力電圧	DC24[V]	
メインクロック周波数	CPU クロック : f_{CLK} 32[MHz] TRD クロック : f_{HOCO} 64[MHz]	
キャリア周波数(PWM)	20 [kHz]	
デッドタイム	2 [μ s]	
制御周期	電流制御/位置・速度推定 : 100 [μ s] (キャリア周期の 2 倍) 速度制御 : 1 [ms]	
回転速度範囲	CW : 0 [rpm] ~ 2650 [rpm] CCW : 0 [rpm] ~ 2650 [rpm] ただし、1060 [rpm]以下は速度オープンループで駆動	
最適化設定	デフォルト設定	
ROM/RAM サイズ	ROM	11.6KB
	RAM	0.85KB
保護停止処理	<ul style="list-style-type: none"> ・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 <ol style="list-style-type: none"> 1. インバータ母線電圧が 28 [V]を超過 2. インバータ母線電圧が 15 [V]より下落 3. 回転速度が 3975 [rpm]を超過 4. 各相の電流が 2.0 [A]を超過 ・外部からの過電流検出信号(INTP0 端子にローレベル入力)を検出した場合、PWM 出力端子をハイインピーダンスにする。 	

【注】 詳細に関しては“モータ制御開発支援ツール「Renesas Motor Workbench」の利用方法”を参照してください。

2.5 ユーザ・オプション・バイト

RL78/G1F のフラッシュメモリのユーザ・オプション・バイト領域の設定を示します。

表 2-6 ユーザ・オプション・バイト設定値

設定値	アドレス	値	設定内容
787BF8	000C0H /010C0H	01111000B	<ul style="list-style-type: none"> ・ウォッチドッグ・タイマのインターバル割り込みの使用 : インターバル割り込みを使用しない ・ウォッチドッグ・タイマのウインドウ・オープン期間 : 100% ・ウォッチドッグ・タイマのカウンタの動作制御 : カウンタ動作許可 (リセット解除後、カウント開始) ・ウォッチドッグ・タイマのオーバフロー時間 : 136 [ms] ・ウォッチドッグ・タイマのカウンタ動作制御 : HALT/STOP モード時, カウンタ動作停止
	000C1H /010C1H	01111011B	<ul style="list-style-type: none"> ・LVD の設定(リセット・モード) 立ち上がり : 2.92 [V] 立ち下がり : 2.86 [V]
	000C2H /010C2H	11111000B	<ul style="list-style-type: none"> ・フラッシュの動作モード設定 : HS (高速メイン)モード ・高速オンチップオシレータ・クロックの周波数 fHOCO : 64 [MHz] fIH : 32 [MHz]

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

モータの起動と停止は、モータ制御開発支援ツールからの入力または SW1 と VR1 の入力によって制御します。SW1 には汎用ポートが割り当てられ、メインループ内で、端子を読み、"Low"レベルのときスタートスイッチが押されていると判断し、逆に"High"レベルの時はモータを停止すると判断します。

3.1.2 A/D 変換

(1) モータ回転速度指令値

モータの回転速度指令値はモータ制御開発支援ツールからの入力、または VR1 の出力値(アナログ値)を A/D 変換することによって決定します。A/D 変換された VR1 の値は、以下の表のように回転速度指令値として使用します。ここでの指令値の最大値は、A/D コンバータの分解能で最大回転数が得られる値に設定しています。

表 3-1 回転速度指令値の変換比

項目	変換比(指令値:A/D 変換値)		チャンネル
回転速度指令値	CW	0 [rpm]~2650 [rpm]:0200H~03FFH	ANI6
	CCW	-2650 [rpm]~0[rpm]:0000H~01FFH	

(2) 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・低電圧検出(異常時は PWM 停止)に使用します。

表 3-2 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)	チャンネル
インバータ母線電圧	0 [V]~111[V]:0000H~03FFH	ANI4

(3) U 相、W 相電流

以下の表のように、U 相、W 相電流を測定し、ベクトル制御に使用します。

表 3-3 U、W 相電流の変換比

項目	変換比(U 相、W 相電流 : A/D 変換値)	チャンネル
U 相、W 相電流	-10 [A]~10 [A] : 0000H~03FFH	Iu : ANI2 Iw : ANI3

A/D 変換特性の詳細に関しては「RL78/G1F ユーザーズマニュアル ハードウェア編」を参照して下さい。

3.1.3 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 3-1 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。

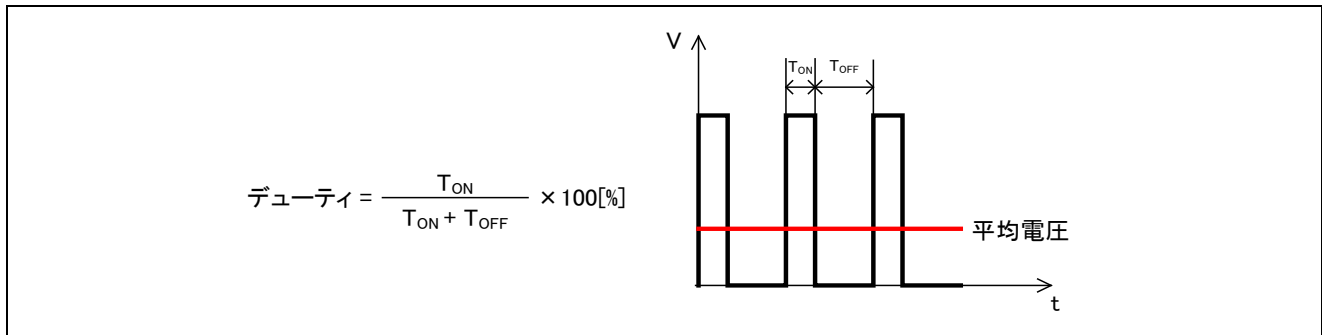


図 3-1 PWM 制御

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

3.1.4 変調

モータへの入力電圧はパルス幅変調（以降、PWM）によって信号生成し印加されます。PWM のパルス幅の生成方法を説明します。

(1). 三角波比較法

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事で出力電圧のパルス幅を決める三角波比較法があります。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事で、正弦波状の指令値電圧を擬似的に出力する事が出来ます。

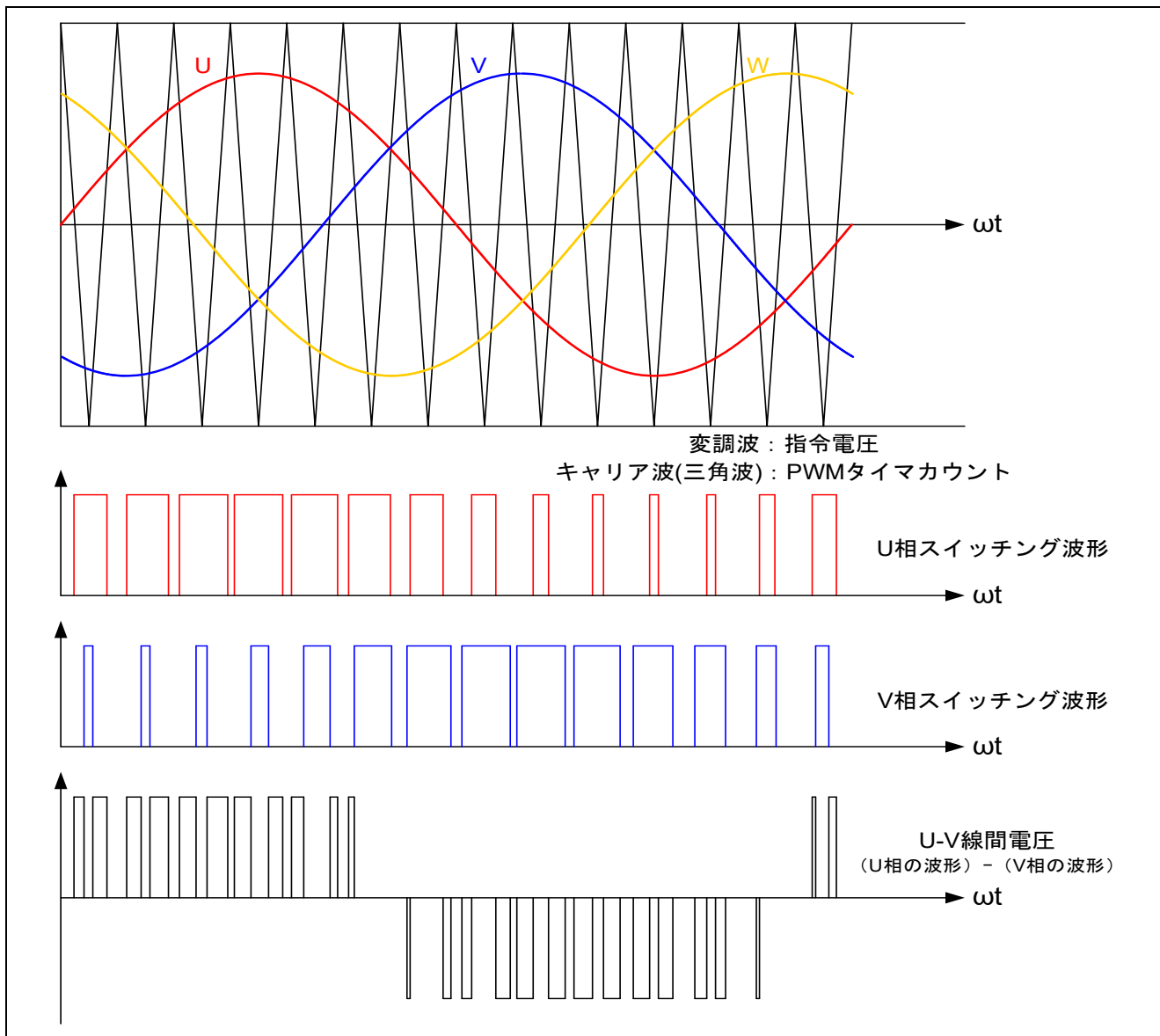


図 3-2 三角波比較法の概念図

(2). 3次高調波重畳法

三角波比較法では線間電圧振幅が入力される直流電圧の約 86.6%までしか使用する事が出来ません。電圧利用効率を向上させるために数多の変調方法がありますが、本プログラムでは3次高調波重畳法を利用する事が出来ます。下記のように指令電圧を計算する事で3次高調波を重畳したのと同じ指令電圧になります。

$$v_o = \frac{\max(v_u^*, v_v^*, v_w^*) + \min(v_u^*, v_v^*, v_w^*)}{2}$$

$$\overline{v_u^*} = v_u^* - v_o$$

$$\overline{v_v^*} = v_v^* - v_o$$

$$\overline{v_w^*} = v_w^* - v_o$$

v_u^*, v_v^*, v_w^* : 元のUVW相の指令電圧、

$\overline{v_u^*}, \overline{v_v^*}, \overline{v_w^*}$: 3次高調波重畳法のUVW相の指令電圧

v_o : 抵抗

上記変調方式は r_mtr_config.h にある MOD_METHOD に下記の値を設定し、コンパイルする事で変更可能です。

MOD_3PH_SPWM	三角波比較法	0
MOD_3PH_TOW	3次高調波重畳法	1: デフォルト設定

3.1.5 状態遷移

図 3-3 に本プログラムにおける状態遷移図を示します。

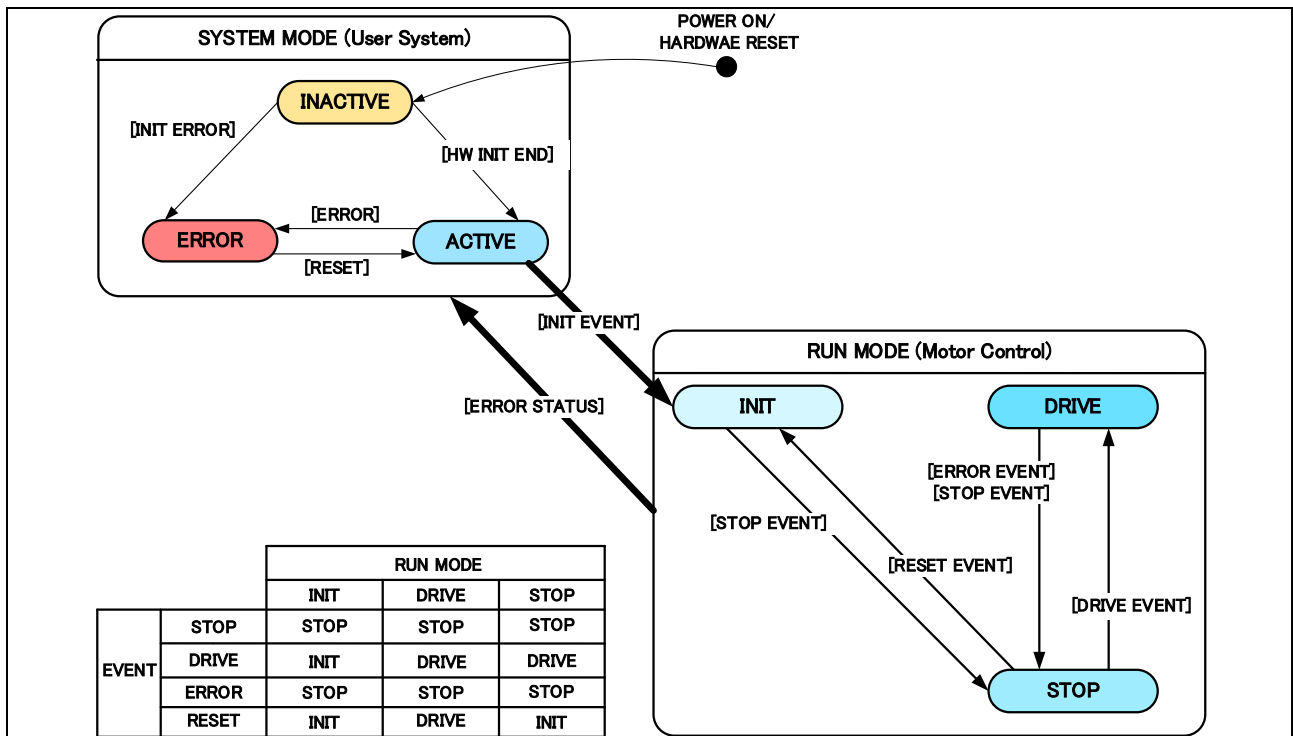


図 3-3 状態遷移図

(1). SYSTEM MODE

システム動作状態を表します。システムの動作状態は、モータ駆動停止(INACTIVE)、モータ駆動(ACTIVE)、異常状態(ERROR)があります。

(2). RUN MODE

モータの駆動状態を表します。各イベント(EVENT)の発生により状態が遷移します。

(3). EVENT

モータの駆動状態の遷移を表します。EVENTが発生すると、RUN MODEが図 3-3 中の表の様に遷移します。各EVENTの発生要因を表 3-4 に示します。

表 3-4 EVENT 一覧

イベント名	発生条件
STOP	ユーザ操作により発生します
DRIVE	ユーザ操作により発生します
ERROR	システムが異常を検出したときに発生します
RESET	ユーザ操作により発生します

3.1.6 始動方法

センサレスベクトル制御ソフトウェアの始動制御内容を図 3-4 に示します。d 軸電流、q 軸電流、速度それぞれの指令値を管理するステータスによってモードをコントロールしています。

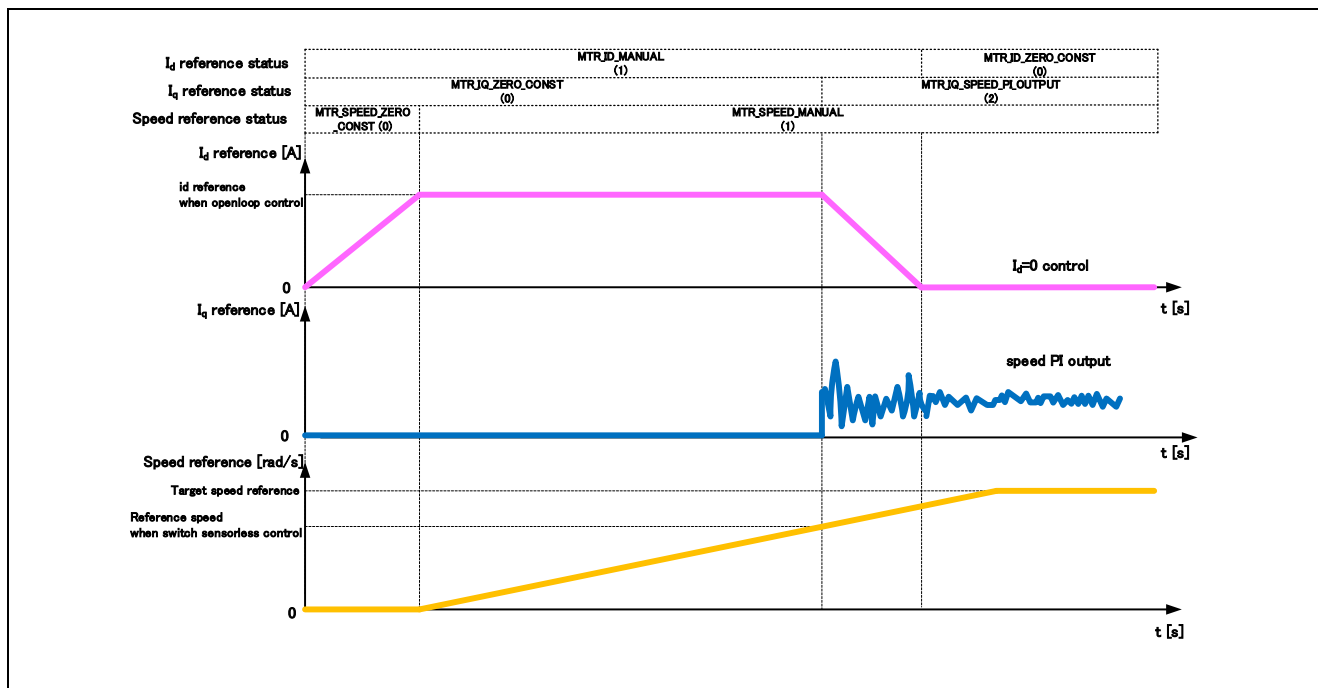


図 3-4 センサレス速度制御ソフトの始動制御内容

3.1.7 制御方法

制御システム全体のブロック図を図 3-5 に示します。制御システムは、座標変換器と非干渉制御器、軸誤差推定器、PLL 制御器、速度制御器 (ASR) と電流制御器 (ACR) から構成されます。また、推定速度に 1 次の LPF を設けています。

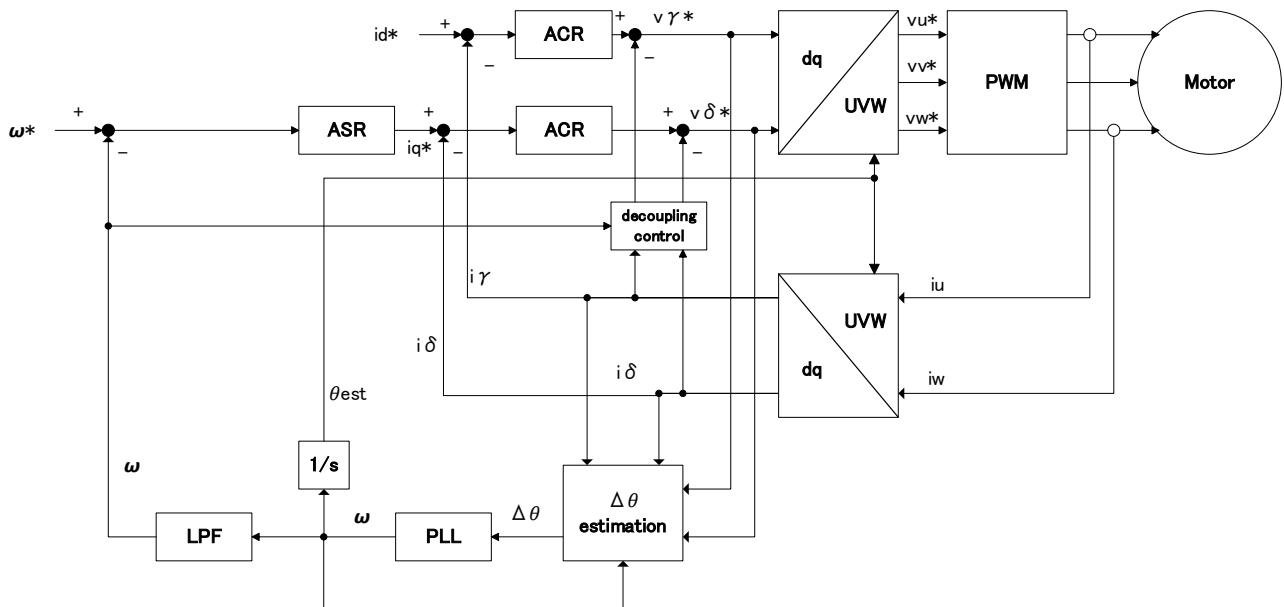


図 3-5 制御システムブロック図

軸誤差推定器は、実軸である dq 軸と推定軸である $\gamma\delta$ 軸との軸誤差 $\Delta\theta$ を推定します。 $\gamma\delta$ 軸における電圧方程式は、次式の dq 軸における電圧方程式の両辺に

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_a + pL_d & -\omega L_q \\ \omega L_d & R_a + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega\psi_a \end{bmatrix}$$

v_d, v_q : d軸電圧, q軸電圧

ω : 角速度

i_d, i_q : d軸電流, q軸電流

L_d, L_q : dq軸インダクタンス

R_a : 抵抗

ψ_a : 永久磁石による電機子鎖交磁束の実効値

次式の回転行列を乗じて得る事が出来ます。

$$\begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{bmatrix}$$

γ 軸と δ 軸の誘起電圧成分 e_γ と e_δ を計算し、次式により軸誤差 $\Delta\theta$ を得ます。誘起電圧の計算の際、 γ 軸電圧と δ 軸電圧をそれぞれ電圧指令値 v_γ^* と v_δ^* で近似して使います。この軸誤差が 0 になる様にフィードバックループを構成する (PLL 制御器) ことで推定速度 ω を求めます。

$$\Delta\theta = \text{atan}\left(\frac{e_\gamma}{e_\delta}\right)$$

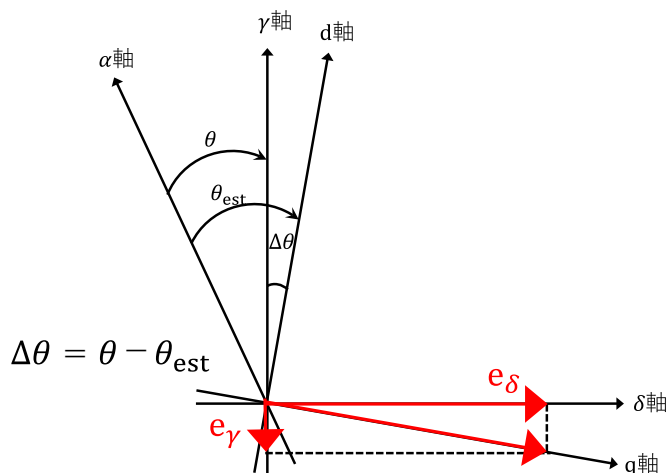


図 3-6 軸誤差とγ軸とδ軸の誘起電圧成分

速度制御器、電流制御器と PLL 制御器は PI 制御器を用いて実現されており、そのゲインは所望の制御に合わせて適宜調整する必要があります。電流 PI 制御ゲイン K_{pACR} 、 K_{iACR} と速度 PI 制御ゲイン K_{pASR} 、 K_{iASR} と PLL 制御ゲイン K_{pPLL} 、 K_{iPLL} はそれぞれ次式のようになります。

$$K_{pACR} = \omega_{CG} L, \quad K_{iACR} = \omega_{CG} R$$

ω_{CG} : 電流 PI 制御固有周波数

L : インダクタンス (d 軸では L_d 、q 軸では L_q)

R : 抵抗

$$K_{pASR} = \frac{\omega_{SG} J}{P_n^2 \psi_a}, \quad K_{iASR} = \frac{\omega_{SG}^2 J}{5 P_n^2 \psi_a}$$

ω_{SG} : 速度 PI 制御固有周波数

J : イナーシャ

ψ_a : 誘起電圧係数

P_n : 極対数

$$K_{pPLL} = \omega_{\Delta\theta}, \quad K_{iPLL} = \frac{\omega_{\Delta\theta}^2}{5}$$

$\omega_{\Delta\theta}$: PLL 制御固有周波数

3.1.8 システム保護機能

本プログラムは、以下のエラー状態を持ち、それぞれの場合に緊急停止する機能を実装しています。システム保護機能に関わる各設定値は表 3-5 を参照して下さい。

- ・ハードウェア過電流エラー

ハードウェアからの緊急停止信号（過電流検出）を検出した時に電圧出力を停止します。

- ・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。

- ・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。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った場合）を検出した時に緊急停止します。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。

- ・回転速度エラー

回転速度監視周期で速度を監視し、速度リミット値を超過した場合、緊急停止します。

表 3-5 各システム保護機能設定値

エラー	閾 値	
ハードウェア過電流エラー	過電流リミット値 [A]	2.0
過電流エラー	過電流リミット値 [A]	2.0
	監視周期 [μs]	100
過電圧エラー	過電圧リミット値 [V]	28
	監視周期 [μs]	100
低電圧エラー	低電圧リミット値 [V]	15
	監視周期 [μs]	100
回転速度エラー	速度リミット値 [rpm]	3975
	監視周期 [μs]	100

3.1.9 単位法 (PU)

固定小数点演算によるモータ制御はダイナミックレンジがコンパイル時に決まります。実際のモータの特性と設計時に想定したモータの特性との違いが大きければ、ダイナミックレンジの差の違いにより、桁あふれ、丸め誤差などの問題が起こり易くなります。本プログラムでは、演算のダイナミックレンジのモータ特性依存度を低減するために単位法 (PU: per-unit) を使用します。任意の物理量の PU 値は基準となる物理量に対する相対的な値であり、以下のように計算できます。

$$PU \text{ 値} = \frac{\text{物理量}}{\text{基準値}}$$

制御に使用される物理量やゲインなどの PU 単位は全て電流、電圧、角周波数、角度の基準値から計算することが出来ます。例えば、抵抗基準値は以下のように電圧基準値と電流基準値で計算します。

$$\text{抵抗基準値} = \frac{\text{電圧基準値}}{\text{電流基準値}} = 1[\text{抵抗PU}]$$

演算ダイナミックレンジがモータの特性から受ける影響を低減するため、電流、電圧、角周波数の基準値はモータ特性を元に設定する必要があります (基準値の取り方は一意ではありません)。本プログラムでは、定格電流、インバータへの入力電圧、最大速度を電流、電圧、角周波数の基準値 (PU 単位) に設定します。各物理量の基準値を表 3-6 に示します。これらの値は `r_mtr_scaling_parameter.h` の中で定義されています。

表 3-6 PU 単位系基準値

カテゴリ	項目	定義	単位
PU 基本物理量	電流	定格電流	[A]
	電圧	入力電圧 (インバータ入力)	[V]
	角周波数	$2\pi \times \text{最大速度}[\text{rpm}] \times \text{極対数}/60$	[Hz]
	角度	1	[rad]
物理量	時間	角度 / 角周波数	[s]
	抵抗	電圧 / 電流	[Ω]
	インダクタンス	抵抗 / 角周波数	[H]
	磁束	電圧 / 角周波数	[Wb]
	イナーシャ	磁束 \times 電流 \times (極対数 / 角周波数) ²	[kgm ² /rad]
電流制御	Kp	抵抗	[Ω]
	Kidt	抵抗	[Ω]
速度制御	Kp	電流 / 角周波数	[A/(rad/s)]
	Kidt	電流 / 角周波数	[A/(rad/s)]
PLL 制御	Kp	角周波数 / 角度	[Hz]
	Kidt	角周波数 / 角度	[Hz]

3.2 センサレスベクトル制御ソフト関数仕様

本プログラムの関数一覧を以下に示します。

表 3-7 “main.c”関数一覧

ファイル名	関数名	処理概要
main.c	main 入力:なし 出力:なし	<ul style="list-style-type: none"> ・ハードウェア初期化関数呼び出し ・ユーザインタフェース初期化関数呼び出し ・メイン処理使用変数初期化関数呼び出し ・状態遷移及びイベント実行関数呼び出し ・母線電圧安定待ち処理呼び出し ・メイン処理 <ul style="list-style-type: none"> ⇒ユーザインタフェース処理呼び出し ⇒ウォッチドッグタイマクリア関数呼び出し
	board_ui 入力:なし 出力:なし	ボードユーザインタフェース使用 <ul style="list-style-type: none"> ・モータステータスの変更 ・回転速度指令値の決定 ・回転方向の決定
	ics_ui 入力:なし 出力:なし	Renesas Motor Workbench 使用 <ul style="list-style-type: none"> ・モータステータスの変更 ・回転速度指令値の決定
	software_init 入力:なし 出力:なし	メイン処理にて使用する変数の初期化

表 3-8 “r_mtr_ics.c” 関数一覧

ファイル	関数	処理概要
r_mtr_ics.c	mtr_set_com_variables 入力:なし 出力:なし	制御変数への入力受け渡し前処理 <ul style="list-style-type: none"> ・ com 変数の値を ICS 変数へ入力 ・ ICS 変数の値を ICS バッファ変数へ入力
	mtr_ics_variables_init 入力:なし 出力:なし	com 変数の初期化
	R_MTR_Limit 入力: int16_t s2_value / 対象の値 int16_t s2_max / 最大値 int16_t s2_min / 最小値 出力: int16_t s2_temp / リミット処理値	最大値、最小値のリミット処理

表 3.9 “ics_RL78G1F.obj” 関数一覧

ファイル名	関数名	処理概要
main.c	ics2_init 入力: unsigned int addr / DTC ベクタテーブル先頭アドレス char pin / SCI 使用ピン char level / 割り込みレベル char num / DTC の構造体の先頭番地 char brr / 通信速度 char mode / 通信モード 出力:なし	通信初期化
	ics2_watchpoint 入力: int16_t s2_ch / 通電パターン 出力:なし	転送関数の呼び出し 250us 以上の間隔で呼び出す必要があります。

表 3-10 “r_mtr_board.c” 関数一覧

ファイル	関数	処理概要
r_mtr_board.c	mtr_board_led_control 入力: uint8_t u1_motor_status / モータステータス uint8_t u1_system_status / システムステータス 出力: なし	LED の制御
	mtr_remove_chattering 入力: (uint8_t) u1_sw / スイッチ入力信号 (uint8_t) u1_on_off / スイッチ状態 出力: (uint8_t) u1_flag_chattering / チャタリングフラグ	スイッチ入力信号のチャタリングの除去

表 3-11 “mtr_ctrl_rl78g1f.c”関数一覧

ファイル名	関数名	処理概要
mtr_ctrl_rl78g1f.c	R_MTR_InitHardware 入力：なし 出力：なし	クロックと周辺機能の初期化
	R_MTR_CtrlStart 入力：なし 出力：なし	タイマ RD PWM 出力許可
	R_MTR_CtrlStop 入力：なし 出力：なし	タイマ RD PWM 出力停止 レジスタ初期化
	R_MTR_GetAdc 入力：uint8_t u1_ad_ch / AD チャネル 出力：uint16_t / AD 変換結果	AD 変換処理
	R_MTR_GetIuwAdc 入力：uint16_t *u2_ad_iuwvdc / UW 相電流ポインタ 出力：なし	UW 相電流検出 AD 変換処理
	R_MTR_GetVdcAdc 入力：なし 出力：なし	電圧検出 AD 変換処理
	R_MTR_ClearOcFlag 入力：なし 出力：なし	過電流フラグのクリア
	mtr_init_clock 【inline function】 入力：なし 出力：uint16_t / クロック設定エラー	クロックの初期化
	mtr_init_ui 【inline function】 入力：なし 出力：なし	ユーザインタフェースの初期化
	mtr_init_tau 【inline function】 入力：なし 出力：なし	タイマ・アレイ・ユニット(TAU)の初期化
	mtr_init_inttm00_interrupt 【inline function】 入力：なし 出力：なし	TAU00 の割り込み設定初期化
	mtr_init_inttm01_interrupt 【inline function】 入力：なし 出力：なし	TAU01 の割り込み設定初期化
	mtr_init_trd 【inline function】 入力：なし 出力：なし	ウォッチドッグタイマ(WDT)のクリア
	mtr_init_ad_converter 【inline function】 入力：なし 出力：なし	AD コンバータの初期化
	mtr_init_pwm_register 【inline function】 入力：なし 出力：なし	PWM 出力レジスタの初期化
mtr_init_intp 【inline function】 入力：なし 出力：なし	PWM ハイインピーダンス状態解除	

表 3-12 “r_mtr_ctrl_gain.obj” 関数一覧

ファイル	関数	処理概要
r_mtr_ctrl_gain.obj	mtr_ctrl_gain 入力: st_mtr_foc_t *st_foc :: FOC 構造体ポインタ const st_mtr_design_parameter_t *st_ctrl_param :: 設計パラメータ構造体ポインタ 出力:なし	ゲイン設計処理

表 3-13 “r_mtr_driver_access.c” 関数一覧

ファイル	関数	処理概要
r_mtr_driver_access.c	R_MTR_InitControl 入力:なし 出力:なし	モータ制御システム初期化 ・モータステータスの初期化 ・制御用変数の初期化
	R_MTR_ExecEvent 入力: uint8_t u1_event :: イベント 出力:なし	モータステータスの変更とイベント処理
	R_MTR_ChargeCapacitor 入力:なし 出力: (uint16_t) u2_charge_cap_error :: タイムアウトエラー	母線電圧安定待ち処理
	R_MTR_SetSpeed 入力: (int16_t) s2_ref_speed_rpm :: 目標回転速度 出力: なし	速度指令値の設定
	R_MTR_GetSpeed 入力: なし 出力: int16_t s2_speed_rpm :: 回転速度	速度の取得
	R_MTR_GetDir 入力:なし 出力: (uint8_t) g_st_120.u1_dir :: 回転方向	回転方向取得
	R_MTR_GetStatus 入力:なし 出力: (uint8_t) mtr_statemachine_get_status(&g_st_120.st_stm) :: モータステータス	モータステータス取得
	R_MTR_GetErrorStatus 入力:なし 出力: (uint16_t) g_st_120.u2_error_status :: エラーステータス	エラーステータス取得
	R_MTR_IcsInput 入力:(mtr_ctrl_input_t) *st_ics_input :: ICS 構造体 出力:なし	ICS 変数の値を ICS バッファ変数へ入力
	R_MTR_SetVariables 【inline function】 入力:なし 出力:なし	ICS バッファ変数の値を制御変数へ入力
	R_MTR_InputBuffParamReset 入力:なし 出力:なし	ICS バッファ変数のリセット
	R_MTR_UpdatePolling 入力:なし 出力:なし	制御変数の設定

表 3-14 “r_mtr_statemachine.c” 関数一覧

ファイル	関数	処理概要
r_mtr_statemachine.c	mtr_statemachine_init 入力:(st_mtr_statemachine_t) *p_state_machine :: モータステータス構造体 出力:なし	モータステータスの初期化
	mtr_statemachine_reset 入力:(st_mtr_statemachine_t) *p_state_machine :: モータステータス構造体 出力:なし	モータステータスのリセット
	mtr_state_machine_event 入力:(st_mtr_statemachine_t) *p_state_machine :: モータステータス構造体 (void) *p_object :: 制御変数用構造体 (uint8_t) u1_event :: イベント 出力:なし	イベントの実行
	mtr_statemachine_get_status 入力:(st_mtr_statemachine_t) *p_state_machine :: モータステータス構造体 出力:(uint8_t) p_state_machine->u1_status :: モータステータス	モータステータスの取得
	mtr_act_none 入力:(st_mtr_statemachine_t) *st_stm :: モータステータス構造体 (void) *p_param :: 制御変数用構造体 出力:なし	処理なし
	mtr_act_init 入力:(st_mtr_statemachine_t) *st_stm :: モータステータス構造体 (void) *p_param :: 制御変数用構造体 出力:なし	制御変数の初期化
	mtr_act_error 入力:(st_mtr_statemachine_t) *st_stm :: モータステータス構造体 (void) *p_param :: 制御変数用構造体 出力:なし	モータの停止
	mtr_act_drive 入力:(st_mtr_statemachine_t) *st_stm :: モータステータス構造体 (void) *p_param :: 制御変数用構造体 出力:なし	制御変数のリセット
	mtr_act_stop 入力:(st_mtr_statemachine_t) *st_stm :: モータステータス構造体 (void) *p_param :: 制御変数用構造体 出力:なし	モータの停止

表 3-15 “r_mtr_foc_less_speed.c” 関数一覧

ファイル名	関数名	処理概要
r_mtr_foc_less_speed.c	mtr_foc_motor_default_init 入力: st_mtr_foc_t*st_foc :: FOC 構造体ポインタ 出力:なし	制御変数の初期化
	mtr_foc_motor_reset 入力: st_mtr_foc_t*st_foc :: FOC 構造体ポインタ 出力:なし	制御変数のリセット
	mtr_ctrl_gain_apply 入力: st_mtr_foc_t*st_foc :: FOC 構造体ポインタ 出力:なし	制御ゲインバッファ変数から制御変数への受け渡し処理

表 3-16 “r_mtr_est_phase_err.obj” 関数一覧

ファイル名	関数名	処理概要
r_mtr_est_phase_err.obj	mtr_est_phase_err 入力: st_mtr_foc_t*st_foc :: FOC 構造体ポインタ 出力:なし	軸誤差推定処理

表 3-17 “mtr_interrupt.c”関数一覧(1/3)

ファイル名	関数名	処理概要
mtr_interrupt.c	mtr_over_current_interrupt 入力：なし 出力：なし	過電流検出処理(外部回路による検出) ・INTP0 割り込み禁止 ・イベント処理選択関数呼び出し(Error イベント発生) ・エラー情報設定(過電流エラー(H/W))
	mtr_100usec_interrupt 入力：なし 出力：なし	周期タイマ割り込み (INTTM01 で呼び出し) 周期: 100 [μs] ・ U、W 相電流検出 ・ 電流検出オフセット補正処理 ・ 過電流エラー監視処理呼び出し ・ ベクトル演算 ・ 非干渉制御処理呼び出し ・ 位置・速度推定演算 ・ 電流 PI 制御処理呼び出し ・ デッドタイム補償処理呼び出し ・ 変調処理呼び出し ・ PWM デューティ設定処理呼び出し ・ 通信処理呼び出し
	mtr_1ms_interrupt 入力：なし 出力：なし	周期タイマ割り込み (INTTM00 で呼び出し) 周期: 1[ms] ・ 母線電圧検出 ・ 電圧の逆数算出 ・ 始動制御 ・ d 軸 q 軸電流、回転速度の指令値設定処理呼び出し ・ 速度 PI 制御処理呼び出し ・ エラー監視処理呼び出し
	mtr_calib_current_offset_uw 【inline function】 入力：st_mtr_tscs_t *st_tscs :: 3 相電流検出構造体ポインタ int16_t *s2_iuw_ad :: UVW 電流ポインタ 出力：uint8_t :: 電流オフセット検出処理完了フラグ	電流オフセット検出処理
	mtr_current_offset_adjustment_uvw 【inline function】 入力：st_mtr_tscs_t *st_tscs :: 3 相電流検出構造体ポインタ int16_t *s2_iuw_ad :: UVW 電流ポインタ int16_t s2_limit_over_current :: 過電流リミット値 出力：uint16_t :: エラーステータス	オフセット除去処理と過電流エラーの検出
	mtr_transform_uvw_dq_abs 【inline function】 入力：int16_t *s2_uvw :: UVW 相ポインタ const st_mtr_ra_t *p_angle :: 角度構造体ポインタ int16_t *s2_dq :: dq 軸ポインタ 出力：なし	UVW → dq 座標変換
	mtr_bemf_magnitude 【inline function】 入力：int16_t s2_bemf_d :: d 軸誘起電圧 int16_t s2_bemf_q :: q 軸誘起電圧 出力：int16_t :: 誘起電圧推定値	誘起電圧の算出処理 オープンループ時の速度算出用

表 3-18 “mtr_interrupt.c”関数一覧(2/3)

ファイル名	関数名	処理概要
mtr_interrupt.c	mtr_rotor_angle_update 【inline function】 入力 : st_mtr_ra_t *p_angle :: 角度構造体ポインタ int16_t s2_angle_rad :: 角度[PU] 出力 : なし	角度の反映処理
	mtr_lpf1_run 【inline function】 入力 : st_mtr_lpf1_t *st_lpf :: LPF 構造体ポインタ int16_t s2_input :: LPF 入力 const uint8_t u1_q :: LPF の Q 値 出力 : なし	1次 LPF 処理
	mtr_current_pi_ctrl 【inline function】 入力 : st_mtr_acr_t *st_acr :: ACR 構造体ポインタ 出力 : なし	電流 PI 処理
	mtr_decoupling 【inline function】 入力 : st_mtr_acr_t *st_acr :: ACR 構造体ポインタ int16_t s2_speed_rad :: 速度 const st_mtr_parameter_t *p_mtr :: モータパラメータ構造体ポインタ 出力 : なし	非干渉制御処理
	mtr_transform_dq_uv_w_abs 【inline function】 入力 : const int16_t *s2_dq :: dq 軸ポインタ const st_mtr_ra_t *p_angle :: 角度構造体ポインタ int16_t *s2_uv_w :: UVW 相ポインタ 出力 : なし	dq → UVW 座標変換
	mtr_deadtime_comp 【inline function】 入力 : st_mtr_deadtime_comp_t *st_dtcomp :: デッドタイム補償構造体ポインタ st_mtr_mod_t *st_mod :: 変調構造体ポインタ int16_t s2_vdc :: 電源電圧 出力 : なし	デッドタイム補償処理
	mtr_uv_w_voltage_limit 【inline function】 入力 : int16_t *s2_ref_v_uv_w :: UVW 相電圧ポインタ int16_t s2_voltage_limit :: 電圧リミット値 出力 : なし	3相電圧の制限処理
	mtr_mod 【inline function】 入力 : st_mtr_mod_t *st_mod :: 変調構造体ポインタ int16_t s2_reci_vdc :: 電圧の逆数 int16_t s2_voltage_limit :: 電圧リミット値 出力 : なし	変調処理
	mtr_pwm_duty_ts 【inline function】 入力 : st_mtr_tscs_t *st_tscs :: 3相電流検出構造体ポインタ int16_t s2_u :: u 相変調率 int16_t s2_v :: v 相変調率 int16_t s2_w :: w 相変調率 出力 : なし	Duty の算出

表 3-19 “mtr_interrupt.c”関数一覧(3/3)

ファイル名	関数名	処理概要
mtr_interrupt.c	mtr_set_speed_ref 【inline function】 入力:st_mtr_foc_t *st_foc :: FOC 構造体ポインタ 出力:int16 s2_speed_rad_ref_buff :: 速度指令値	速度制御用指令値の設定
	mtr_pi_run 【inline function】 入力:st_mtr_pi_t *st_pi :: PI 制御構造体ポインタ int16_t s2_err :: 偏差 const uint8_t u1_kp_q :: 比例ゲインシフト値 const uint8_t u1_kidt_q :: 積分ゲインシフト値 出力:int16 s2_pi_out :: PI 出力	PI 制御処理
	mtr_set_iq_ref 【inline function】 入力:st_mtr_foc_t *st_foc :: FOC 構造体ポインタ 出力:int16 s2_iq_ref_buff :: q 軸電流指令値	q 軸電流指令値の設定
	mtr_set_id_ref 【inline function】 入力:st_mtr_foc_t *st_foc :: FOC 構造体ポインタ 出力:int16 s2_id_ref_buff :: d 軸電流指令値	d 軸電流指令値の設定
	mtr_error_check 入力:st_mtr_foc_t *st_foc :: FOC 構造体ポインタ 出力:なし	エラー処理 ・過電圧検出 ・低電圧検出 ・過速度検出
	mtr_abs 入力:int16_t s2_value :: 入力値 出力:int16_t :: 出力値	入力の絶対値を出力
	mtr_limit_abs 入力:int16_t s2_value :: 入力値 int16_t s2_limit_value :: 制限値 出力:int16_t :: 出力値	入力を絶対値で制限

3.3 センサレスベクトル制御ソフト変数一覧

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、本制御プログラムの制御値は、各値をスケーリングした上で算出しています。Q表記が適用される変数について、スケール欄にあるQnは小数部分がnビットであることを表します。ただし、一部の変数と構造体メンバのQ表記に関してはr_mtr_scaling_parameter.hにある定義で計算されるので、この場合はスケール欄にデフォルトのQ表記を記載しています。PU単位が適用される変数・構造体メンバの単位は[PU([元単位])]のように記述します。

表 3-20 “main.c” 変数一覧

変数	型	Qn	PU	内容	備考
g_u1_system_mode	static uint8_t	Q0	-	モードシステム管理用変数	
g_u1_motor_status	static uint8_t	Q0	-	モータステータス管理用変数	
g_u2_error_status	static uint16_t	Q0	-	エラーステータス管理用変数	
g_u1_reset_req	static uint8_t	Q0	-	リセット要求フラグ	0: リセット要求なし 1: リセット要求あり
g_u1_stop_req	static uint8_t	Q0	-	ストップ要求フラグ	
g_u1_flag_ui_change	static uint8_t	Q0	-	UI 変更フラグ	0: GUI 使用 (デフォルト) 1: ボードユーザインタフェース使用
g_u1_flag_set_values	static uint8_t	Q0	-	変数反映フラグ	
g_u2_conf_hw	uint16_t	Q0	-	RMW configuration 用変数	
g_u2_conf_sw	uint16_t	Q0	-		
g_u2_conf_tool	uint16_t	Q0	-		
gui_u1_active_gui	uint8_t	Q0	-		
g_u2_conf_sw_ver	uint16_t	Q0	-		
com_s2_sw_userif	int16_t	Q0	-	UI 管理用変数	0: ICS_UI 1: BOARD_UI
g_s2_sw_userif	int16_t	Q0	-		
com_u1_run_event	uint8_t	Q0	-	ランモード遷移用変数	0: MTR_EVENT_STOP 1: MTR_EVENT_DRIVE 2: MTR_EVENT_ERROR 3: MTR_EVENT_RESET
g_u1_run_event	uint8_t	Q0	-		
g_u2_system_error	uint16_t	Q0	-	システムエラー管理用変数	

表 3-21 “r_mtr_board.c” 変数一覧

変数	型	Qn	PU	内容	備考
u1_sw_cnt	static uint8_t	Q0	-	チャタリング判別用カウンタ	

表 3-22 “r_mtr_ics.c” 変数一覧

変数	型	Qn	PU	内容	備考
com_u1_direction	uint8_t	Q0	-	回転方向	0 : CW 1 : CCW
com_f4_mtr_r	float	-	-	抵抗 [Ω]	
com_f4_mtr_ld	float	-	-	d 軸インダクタンス[H]	
com_f4_mtr_lq	float	-	-	q 軸インダクタンス[H]	
com_f4_mtr_m	float	-	-	誘起電圧計数 [Vs/rad]	
com_f4_mtr_j	float	-	-	イナーシャ[kg m ²]	
com_u2_mtr_pp	uint16_t	Q0	-	極対数	
com_u2_offset_calc_time	uint16_t	Q0	-	電流オフセット検出時間	
com_s2_ref_speed_rpm	int16_t	Q0	-	目標速度 [rpm]	機械角
com_f4_ramp_limit_speed_rpm	float	-	-	加速度制限値 [rpm/ms]	機械角
com_s2_max_speed_rpm	int16_t	Q0	-	最大速度 [rpm]	機械角
com_f4_acr_nf_hz	float	Q0	-	電流 PI 制御固有周波数 [Hz]	
com_f4_asr_nf_hz	float	Q0	-	速度 PI 制御固有周波数 [Hz]	
com_f4_asr_lpf_cof_hz	float	Q0	-	速度 LPF 固有周波数 [Hz]	
com_f4_pll_nf_hz	float	Q0	-	PLL 固有周波数 [Hz]	
com_s2_cl2ol_speed_rpm	int16_t	Q0	-	センサレスからオープンループへの切り替え速度[rpm]	機械角
com_s2_ol2cl_speed_rpm	int16_t	Q0	-	オープンループからセンサレスへの切り替え速度[rpm]	機械角
com_f4_ol_ref_id	float	-	-	オープンループ d 軸指令電流[A]	
com_f4_init_asr_intg	float	-	-	センサレス移行時の ASR の積分項初期値	
com_f4_ramp_limit_current	float	-	-	電流の上昇制限値[A/ms]	
com_s2_enable_write	int16_t	Q0	-	変数書き換え許可	
g_s2_enable_write	int16_t	Q0	-	変数書き換え許可	
st_ics	mtr_ctrl_input_t	Q0	-	ICS 変数受け渡し構造体	構造体

表 3-23 “r_mtr_driver_access.c” 変数一覧

変数	型	Qn	PU	内容	備考
st_ics_buff	mtr_ctrl_input_t	Q0	-	ICS 変数受け渡しバッファ構造体	構造体
g_u1_trig_enable_write	uint8_t	Q0	-	受け渡し完了フラグ	
g_u1_stop_req	uint8_t	Q0	-	モータ停止フラグ	
g_s2_cnt	int16_t	Q0	-	カウンタ	

表 3-24 “r_mtr_sc_table.c” 変数一覧

変数	型	Qn	PU	内容	備考
sc_table[1025]	static int16_t	Q0	-	三角関数テーブル	

表 3-25 “r_mtr_statemachine.c” 変数一覧

変数	型	Qn	PU	内容	備考
state_transition_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static uint8_t	Q0	-	状態遷移用のマクロ配列	
action_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static mtr_action_t	Q0	-	状態遷移用の関数配列	

表 3-26 “r_mtr_interrupt.c” 変数一覧

変数	型	Qn	PU	内容	備考
gst_foc	st_mtr_foc_t	Q0	-	ベクトル制御構造体	構造体
g_u1_cnt_ics	static uint8_t	Q0	-	通信処理周期間引き変数	

3.4 センサレスベクトル制御ソフト構造体一覧

本制御プログラムで使用する構造体一覧を次に示します。使用されていない構造体は省略します

表 3-27 “r_mtr_parameter.h / 構造体 : ”st_mtr_parameter_t” 変数一覧

変数	型	Qn	PU	内容	備考
u2_mtr_pp	uint16_t	Q0	-	極対数	
s2_mtr_r	int16_t	Q15	抵抗 (電圧/電流)	抵抗 [PU]	
s2_mtr_ld	int16_t	Q15	インダクタンス (抵抗/角周波数)	d 軸インダクタンス [PU]	
s2_mtr_lq	int16_t	Q15	インダクタンス (抵抗/角周波数)	q 軸インダクタンス [PU]	
s2_mtr_m	int16_t	Q12	誘起電圧係数 (電圧/角周波数)	誘起電圧係数 [PU]	
s2_mtr_j	int16_t	Q7	イナーシャ (誘起電圧係数 × 電流 × (極対数/角周波数) ²)	イナーシャ [PU]	

表 3-28 “r_mtr_parameter.h / 構造体 : ”st_mtr_design_parameter_t” 変数一覧

変数	型	Qn	PU	内容	備考
f4_acr_nf_hz	float	Q0	-	電流 PI 制御固有周波数 [Hz]	
f4_asr_nf_hz	float	Q0	-	速度 PI 制御固有周波数 [Hz]	
f4_asr_lpf_cof_hz	float	Q0	-	速度 LPF 固有周波数 [Hz]	
f4_pll_nf_hz	float	Q0	-	PLL 固有周波数 [Hz]	
f4_dt	float	Q0	-	電流制御周期	
f4_dt_speed	float	Q0	-	速度制御周期	
f4_r	float	Q0	-	抵抗 [Ω]	
f4_ld	float	Q0	-	d 軸インダクタンス [H]	
f4_lq	float	Q0	-	q 軸インダクタンス [H]	
f4_m	float	Q0	-	磁束 [Wb]	
f4_j	float	Q0	-	イナーシャ [kgm^2]	
f4_pu_sf_afreq	float	Q0	-	周波数のスケールファクタ	
u1_q_pll_kp	uint8_t	Q0	-	PLL 比例ゲインの Q 値	
u1_q_pll_kidt	uint8_t	Q0	-	PLL 積分ゲイン * 制御周期の Q 値	
u1_q_acr_kp	uint8_t	Q0	-	電流 PI 比例ゲインの Q 値	
u1_q_acr_kidt	uint8_t	Q0	-	電流 PI 積分ゲイン * 制御周期の Q 値	
u1_q_asr_kp	uint8_t	Q0	-	速度 PI 比例ゲインの Q 値	
u1_q_asr_kidt	uint8_t	Q0	-	速度 PI 積分ゲイン * 制御周期の Q 値	
u1_q_asr_lpf_k	uint8_t	Q0	-	速度 LPF ゲインの Q 値	

表 3-29 “r_mtr_parameter.h / 構造体 : ”st_mtr_ctrl_gain_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_acr_id_kp	int16_t	Q16	抵抗	d 軸電流制御比例ゲイン	
s2_acr_id_kidt	int16_t	Q16	抵抗	d 軸電流制御積分ゲイン*演算周期	
s2_acr_iq_kp	int16_t	Q16	抵抗	q 軸電流制御比例ゲイン	
s2_acr_iq_kidt	int16_t	Q16	抵抗	q 軸電流制御積分ゲイン*演算周期	
s2_asr_pi_kp	int16_t	Q14	電流/角周波数	速度制御比例ゲイン	
s2_asr_pi_kidt	int16_t	Q18	電流/角周波数	速度制御積分ゲイン*演算周期	
s2_acr_lpf_in_k	int16_t	Q14	-	電流 LPF 入力係数	
s2_pll_kp	int16_t	Q13	角周波数/角度	PLL 比例ゲイン	
s2_pll_kidt	int16_t	Q17	1 角周波数/角度	PLL 積分ゲイン*演算周期	

表 3-30 “r_mtr_driver_access.h / 構造体 : ”st_mtr_ctrl_input_t” 変数一覧

変数	型	Qn	PU	内容	備考
u1_direction	uint8_t	Q0	-	回転方向	
u2_offset_calc_time	uint16_t	Q0	-	オフセット検出時間	
s2_ref_speed_rad	int16_t	Q14	角周波数	指令速度 [PU]	電気角
s2_ramp_limit_speed_rad	int16_t	Q14	角周波数	加速度制限値 [PU]	電気角
s2_max_speed_rad	int16_t	Q14	角周波数	最大速度 [PU]	電気角
s2_cl2ol_speed_rad	int16_t	Q14	角周波数	センサレスからオープンループへの切り替え速度[PU]	電気角
s2_ol2cl_speed_rad	int16_t	Q14	角周波数	オープンループからセンサレスへの切り替え速度[PU]	電気角
s2_ol_ref_id	int16_t	Q13	電流	オープンループ d 軸指令電流[PU]	
s2_init_intg	int16_t	Q16	電流	センサレス移行時の ASR の積分項初期値	
s2_ramp_limit_current	int16_t	Q13	電流	電流の上昇制限値[PU/ms]	
st_motor	st_mtr_parameter_t	-	-	モータパラメータ構造体	構造体
st_gain	st_mtr_ctrl_gain_t	-	-	PI 制御パラメータ構造体	

表 3-31 “r_mtr_statemachine.h / 構造体 : ” st_mtr_statemachine_t” 変数一覧

変数	型	Qn	PU	内容	備考
u1_status	uint8_t	Q0	-	モータステータス	
u1_status_next	uint8_t	Q0	-	遷移後モータステータス	
u1_current_event	uint8_t	Q0	-	実行イベント	

表 3-32 “r_mtr_foc_less_speed.h / 構造体 : ” st_mtr_lpf1_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_in_k	int16	電流 : Q14 速度 : Q14	-	LPF 入力ゲイン	
s2_out_k	int16	電流 : Q14 速度 : Q14	-	LPF 前回出力ゲイン	
s2_pre_out	int16	電流 : Q13 速度 : Q14	電流 : 電流 速度 : 角周波数	前回出力値	

表 3-33 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_pi_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_kp	int16_t	電流 : Q16 速度 : Q14 PLL : Q13	-	比例ゲイン	
s2_kidt	int16_t	電流 : Q16 速度 : Q18 PLL : Q17	-	積分ゲイン×制御周期	
s2_intg	int16_t	電流 : Q13 速度 : Q14 PLL : Q14	電流 : 抵抗 速度 : 電流/角周波数 PLL : 角周波数/角度	積分項	
s2_ilimit	int16_t	電流 : Q13 速度 : Q14 PLL : Q14	電流 : 抵抗 速度 : 電流/角周波数 PLL : 角周波数/角度	積分制限 (上下対称)	

表 3-34 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_acr_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_ctrl_period	int16_t	Q18	時間	電流制御周期	
s2_ref_vd	int16_t	Q13	電圧	d 軸出力電圧指令値	
s2_ref_vq	int16_t	Q13	電圧	q 軸出力電圧指令値	
s2_pre_ref_vd	int16_t	Q13	電圧	d 軸出力電圧指令前回値	
s2_pre_ref_vq	int16_t	Q13	電圧	q 軸出力電圧指令前回値	
s2_id_ad	int16_t	Q13	電流	d 軸電流	
s2_iq_ad	int16_t	Q13	電流	q 軸電流	
s2_ref_id	int16_t	Q13	電流	d 軸電流指令	
s2_ref_iq	int16_t	Q13	電流	q 軸電流指令	
s2_ref_id_ctrl	int16_t	Q13	電流	d 軸電流指令制御値	
s2_ref_iq_ctrl	int16_t	Q13	電流	q 軸電流指令制御値	
s2_limit_iq	int16_t	Q13	電流	q 軸電流リミット	
s2_ol_ref_id	int16_t	Q13	電流	オープンループ d 軸電流指令値	
s2_ramp_limit_current	int16_t	Q13	電流	電流の上昇制限値[PU/ms]	
st_pi_id	st_mtr_pi_t	-	-	d 軸電流 PI 構造体	構造体
st_pi_iq	st_mtr_pi_t	-	-	q 軸電流 PI 構造体	

表 3-35 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_pll_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_dt	int16_t	Q18	時間	制御周期	
s2_speed_rad	int16_t	Q14	各周波数	速度	
st_pi	st_mtr_pi_t		-	PI 構造体	構造体

表 3-36 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_ra_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_rotor_angle_rad;	int16_t	Q12	角度	角度	
s2_rotor_angle_rad_ctrl	int16_t	Q12	角度	角度制御値	
s2_sin;	int16_t	Q12	-	Sin	
s2_cos;	int16_t	Q12	-	Cos	

表 3-37 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_deadtime_comp_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_deadtime_error_voltage	int16_t	Q12	電圧	電圧誤差	
s2_deadtime_limit_current	int16_t	Q12	電流	電流リミット	
s2_ref_i_uvw[3]	int16_t	Q12	電流	3相指令電流	
s2_delta_v_uvw[3]	int16_t	Q12	電圧	3相電圧補償値	

表 3-38 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_deadtime_comp_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_deadtime_error_voltage	int16_t	Q12	電圧	電圧誤差	
s2_deadtime_limit_current	int16_t	Q12	電流	電流リミット	
s2_ref_i_uvw[3]	int16_t	Q12	電流	3相指令電流	
s2_delta_v_uvw[3]	int16_t	Q12	電圧	3相電圧補償値	

表 3-39 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_deadtime_comp_t” 変数一覧

変数	型	Qn	PU	内容	備考
u1_ref_dir	int16_t	-	-	回転方向指令	0 : CW 1 : CCW
s2_speed_ctrl_period	int16_t	Q15	時間	速度制御周期	
s2_ref_speed_rad	int16_t	Q14	角周波数	指令速度	
s2_ref_speed_rad_ctrl	int16_t	Q14	角周波数	指令速度制御値	
s2_speed_rad	int16_t	Q14	角周波数	速度	
s2_ramp_limit_speed_rad	int16_t	Q14	角周波数	加速度制限値	
s2_max_speed_rad	int16_t	Q14	角周波数	最大速度	
s2_limit_speed_rad	int16_t	Q14	角周波数	制限速度	
s2_init_intg	int16_t	Q13	電流	センサレス切り替え時積分項初期値	
s2_cl2ol_speed_rad	int16_t	Q14	角周波数	センサレスからオープンループへの切り替え速度	
s2_ol2cl_speed_rad	int16_t	Q14	角周波数	オープンループからセンサレスへの切り替え速度	
st_pi	st_mtr_pi_t	-	-	速度 PI 構造体	構造体
st_lpf	st_mtr_lpf1_t	-	-	速度 LPF 構造体	

表 3-40 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_mod_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_ref_vu	int16_t	Q13	電圧	U 相電圧指令	
s2_ref_vv	int16_t	Q13	電圧	V 相電圧指令	
s2_ref_vw	int16_t	Q13	電圧	W 相電圧指令	
s2_com_v	int16_t	Q13	電圧	電圧オフセット	
s2_mod_u;	int16_t	Q12	-	U 相変調率	
s2_mod_v;	int16_t	Q12	-	V 相変調率	
s2_mod_w;	int16_t	Q12	-	W 相変調率	

表 3-41 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_tscs_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_duty_u	int16_t	Q0	-	U 相 Duty(PWM レジスタ設定値)	
s2_duty_v	int16_t	Q0	-	V 相 Duty(PWM レジスタ設定値)	
s2_duty_w	int16_t	Q0	-	W 相 Duty(PWM レジスタ設定値)	
s2_offset_iu	int16_t	Q13	電流	U 相電流オフセット値	
s2_offset_iw	int16_t	Q13	電流	W 相電流オフセット値	
s4_offset_iu_sum	int32_t	Q13	電流	U 相電流オフセット値積分値	
s4_offset_iw_sum	int32_t	Q13	電流	W 相電流オフセット値積分値	
u2_offset_calc_time	uint16_t	-	-	オフセット電流測定回数	
u2_offset_sample_cnt	uint16_t	-	-	オフセット電流測定サンプル数	
u2_cmt_ad[2]	uint16_t	-	-	UW 相電流 AD 変換値	

表 3-42 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_est_phe_t” 変数一覧

変数	型	Qn	PU	内容	備考
s2_ed	int16_t	Q13	電圧	d 軸誘起電圧	
s2_eq	int16_t	Q13	電圧	q 軸誘起電圧	
s2_e	int16_t	Q13	電圧	誘起電圧	
s2_phase_err_rad	int16_t	Q12	角度	位相誤差	
s2_r_id	int16_t	Q13	電圧	$R \cdot i_d$	
s2_r_iq	int16_t	Q13	電圧	$R \cdot i_q$	
s2_speed_ld_id	int16_t	Q13	電圧	速度 $\cdot L_d \cdot i_d$	
s2_speed_lq_iq	uint16_t	Q13	電圧	速度 $\cdot L_q \cdot i_q$	

表 3-43 “r_mtr_foc_less_speed.h / 構造体 : ” st_mtr_foc_t” 変数一覧 (1/2)

変数	型	Qn	PU	内容	備考
u2_run_mode	uint16_t	-	-	運転モード	0x00 : Init モード 0x01 : Boot モード 0x02 : Drive モード 0x03 : Analysis モード 0x04 : Tune モード
u2_ctrl_conf	uint16_t	-	-	制御入力	0x01 : 電流制御 0x02 : 速度制御 0x04 : 位置制御 0x08 : トルク制御 0x10 : 電圧制御
u2_error_status	uint16_t	-	-	エラーステータス	0x0000 : エラーなし 0x0001 : 過電流エラー(H/W) 0x0002 : 過電圧エラー 0x0004 : 回転速度エラー 0x0008 : ホールタイムアウトエラー 0x0010 : 誘起電圧タイムアウトエラー 0x0020 : ホールパターンエラー 0x0040 : 誘起電圧パターンエラー 0x0080 : 低電圧エラー 0x0100 : 過電流エラー(S/W) 0xFFFF : 未定義エラー
u1_direction	uint8_t	-	-	現在の回転方向	0 : CW 1 : CCW
u1_flag_charge_cap	uint8_t	-	-	電流オフセット値計算 フラグ	0 : オフセット計算処理実行 1 : オフセット計算処理終了
u1_state_offset_calc	uint8_t	-	-	オフセット検出状態降フラグ	0 : オフセット除去中 1 : オフセット除去処理終了 2 : オフセット除去完了
u1_state_ref_id	uint8_t	-	-	d 軸電流指令値生成ステータス	0 : d 軸電流増加 1 : d 軸電流一定 2 : d 軸電流減少 3 : d 軸電流 0
u1_state_ref_iq	uint8_t	-	-	q 軸電流指令値生成ステータス	0 : q 軸電流 0 1 : 速度 PI 出力 2 : q 軸電流減少
u1_state_ref_speed	uint8_t	-	-	速度指令値生成ステータス	0 : 速度 0 1 : 速度変化
u1_flag_down_to_ol	uint8_t	-	-	オープンループ遷移フラグ	0 : 移行なし 1 : 移行実行
s2_iu_ad	int16_t	Q13	電流	U 相電流	
s2_iv_ad	int16_t	Q13	電流	V 相電流	
s2_iw_ad	int16_t	Q13	電流	W 相電流	
s2_vdc_ad	int16_t	Q13	電圧	電源電圧	
s2_reci_vdc	int16_t	Q13	1/電圧	電圧の逆数	
s2_reci_flux	int16_t	Q13	1/磁束	磁束の逆数	
s2_limit_vout	int16_t	Q13	電圧	電圧リミット	
s2_limit_over_current	int16_t	Q13	電流	過電流リミット値	
s2_limit_over_voltage	int16_t	Q13	電圧	過電圧リミット値	
s2_limit_under_voltage	int16_t	Q13	電圧	低電圧リミット値	

表 3-44 “r_mtr_foc_less_speed.h / 構造体 : ”st_mtr_foc_t” 変数一覧 (2/2)

変数	型	Qn	PU	内容	備考
st_stm	st_mtr_statemachine_t	-	-	ステートマシン構造体	
st_motor	st_mtr_parameter_t	-	-	モータパラメータ構造体	
st_ra	st_mtr_ra_t	-	-	ロータ角度構造体	
st_phe	st_mtr_est_phe_t	-	-	位相誤差推定構造体	
st_tscs	st_mtr_tscs_t	-	-	3相電流検出構造体	
st_acr	st_mtr_acr_t	-	-	ACR 構造体	電流 PI 制御
st_asr	st_mtr_asr_t	-	-	ASR 構造体	速度 PI 制御
st_mod	st_mtr_mod_t	-	-	変調構造体	
st_pll	st_mtr_pll_t	-	-	PLL 制御構造体	
st_gain_buf	st_mtr_ctrl_gain_t	-	-	制御ゲイン構造体	
st_dt_comp	st_mtr_deadtime_comp_t	-	-	デッドタイム補償構造体	

3.5 センサレスベクトル制御ソフトマクロ定義一覧

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。

表 3-45 “r_mtr_config.h” マクロ定義一覧

マクロ	定義値	内容	備考
IP_MRSSK	-	インバータボードの選択	
MP_TG55L	-	モータパラメータの選択	
CP_TG55L	-	制御パラメータの選択	
ICS_UI	0	RMW UI	デフォルト設定
BOARD_UI	1	RSSK ボード UI	
MTRCONF_DEFAULT_UI	0:1	UI の選択	ボード UI / ICS UI
USE_DEADTIME_COMP	0:1	デッドタイム補償の選択	デフォルト設定 0
USE_SPEED_LPF	0:1	速度 LPF の選択	デフォルト設定 1
MOD_3PH_SPWM	0	正弦波変調	
MOD_3PH_TOW	1	3 次高調波加算	
MOD_METHOD	0:1	変調方式	デフォルト設定 1
SC_LIB	0	三角関数ライブラリ	
SC_TABLE	1	三角関数テーブル	
SC_METHOD	0:1	三角関数演算方式	デフォルト設定 0

表 3-46 “motor_parameter.h”マクロ定義一覧

マクロ名	定義値	備考	備考
MP_POLE_PAIRS	2	極対数	
MP_RESISTANCE	9.125f	抵抗 [Ω]	
MP_D_INDUCTANCE	0.003844f	d 軸インダクタンス [H]	
MP_Q_INDUCTANCE	0.004315f	q 軸インダクタンス [H]	
MP_MAGNETIC_FLUX	0.02144f	磁束 [Wb]	
MP_ROTOR_INERTIA	0.000002050f	イナーシャ [kgm^2]	
MP_NOMINAL_CURRENT_RMS	0.42f	定格電流 [A]	
MP_RATED_SPEED	2650	定格速度 [rpm]	

表 3-47 “control_parameter.h”マクロ定義一覧

マクロ名	定義値	備考	備考
CP_ACR_NF_HZ	300.0f	電流 PI 制御固有周波数 [Hz]	
CP_ASR_NF_HZ	15.0f	速度 PI 制御固有周波数 [Hz]	
CP_PLL_NF_HZ	30.0f	PLL 制御固有周波数 [Hz]	
CP_MAX_SPEED_RPM	MP_RATED_SPEED	最大速度(機械角)[rpm]	
CP_SPEED_LIMIT_RPM	MP_RATED_SPEED*1.5	制限速度(機械角)[rpm]	
CP_OC_LIMIT	1.5	過電流リミット値 [A]	
CP_OL_REF_ID	MP_NOMINAL_CURRENT_RMS	d 軸電流指令値 [A]	
CP_INIT_ASR_INTEG	0.2f	q 軸電流 PI 積分項初期値 [A]	
CP_LAMP_LIMIT_CURRENT	0.01f	電流の上昇制限値[PU/ms]	
CP_OL2CL_SPEED_RPM	MP_RATED_SPEED*0.4	センサレスからオープンループへの切り替え速度(機械角)[rpm]	
CP_CL2OL_SPEED_RPM	MP_RATED_SPEED*0.3	オープンループからセンサレスへの切り替え速度(機械角)[rpm]	
CP_LAMP_LIMIT_SPEED_RPM	1	加速度制限値 [rpm/ms]	
CP_OFFSET_CALC_TIME	384.0f	電流オフセット値計算回数	

表 3-48 “r_mtr_inverter_parameter.h” マクロ定義一覧

マクロ	定義値	内容	備考
IP_DEADTIME	2.0f	デッドタイム	
IP_CURRENT_RANGE	20.0f	電流スケーリングレンジ [A]	
IP_VDC_RANGE	111.0f	電圧スケーリングレンジ [V]	
IP_INPUT_V	24.0f	入力電圧 [V]	
IP_CURRENT_LIMIT	2.0f	電流リミット値 [A]	
IP_OVERVOLTAGE_LIMIT	28.0f	過電圧リミット [V]	
IP_UNDERVOLTAGE_LIMIT	15.0f	低電圧リミット [V]	

表 3-49 “r_mtr_scaling_parameter.h” マクロ定義一覧

マクロ	定義値	内容	備考
FP_SF_VOLTAGE	37	電圧 PU 変換値 (((IP_VDC_RANGE/1023)*PU_SF_VOLTAGE) *(1<<MTR_Q_VOLTAGE))の値を設定	
FP_SF_CURRENT	381	電流 PU 変換値 (((IP_CURRENT_RANGE/1023)*PU_SF_CUR RENT) * (1<<MTR_Q_CURRENT)) の値を設定	
PU_BASE_CURRENT_A	MP_NOMINAL_CURRENT_RMS	電流基準値 [A]	
PU_BASE_VOLTAGE_V	IP_INPUT_V	電圧基準値 [V]	
PU_BASE_FREQ_Hz	MTR_TWOPole*CP_MAX_SPEED_RPM *MP_POLE_PAIRS/60	周波数基準値 [Hz]	
PU_BASE_ANGLE_Rad	1.0f	角度基準値 [rad]	
PU_SF_CURRENT	1.0f / PU_BASE_CURRENT_A	電流スケール [PU/A]	
PU_SF_VOLTAGE	1.0f / PU_BASE_VOLTAGE_V	電圧スケール [PU/V]	
PU_SF_AFREQ	1.0f / PU_BASE_FREQ_Hz	角周波数スケール [PU/(rad/s)]	
PU_SF_ANGLE	1.0f / PU_BASE_ANGLE_Rad	角度スケール[PU/rad]	
PU_SF_TIME	PU_SF_ANGLE / PU_SF_AFREQ	時間スケール [PU/s]	
PU_SF_RES	PU_SF_VOLTAGE / PU_SF_CURRENT	抵抗スケール[PU/ohm]	
PU_SF_IND	PU_SF_RES / PU_SF_AFREQ	インダクタンススケール[PU/H]	
PU_SF_FLUX	PU_SF_VOLTAGE / PU_SF_AFREQ	磁束スケール[PU/Wb]	
PU_SF_INERTIA	PU_SF_FLUX * PU_SF_CURRENT / (MP_POLE_PAIRS * MP_POLE_PAIRS * PU_SF_AFREQ * PU_SF_AFREQ)	イナーシャスケール [PU/(rad/kgm ²)]	
PU_SF_RPM_RAD	1.0f / CP_MAX_SPEED_RPM	[rpm]から[rad/s]への変換スケール	
PU_SF_RAD_RPM	CP_MAX_SPEED_RPM	[rad/s]から[rpm]への変換スケール	
PU_SF_ACR_KP	PU_SF_RES	電流 PI 比例ゲインスケール	
PU_SF_ACR_KIDT	PU_SF_RES	電流 PI 積分ゲインスケール	
PU_SF_ASR_KP	PU_SF_CURRENT / PU_SF_AFREQ	速度 PI 比例ゲインスケール	
PU_SF_ASR_KIDT	PU_SF_CURRENT / PU_SF_AFREQ	速度 PI 積分ゲインスケール	
PU_SF_PLL_KP	PU_SF_AFREQ / PU_SF_ANGLE	PLL 比例ゲインスケール	
PU_SF_PLL_KIDT	PU_SF_AFREQ / PU_SF_ANGLE	PLL 積分ゲインスケール	
MTR_Q_ANGLE	12	角度の Q 値	
MTR_Q_CTRL_TIME	17	FOC 制御周期の Q 値	
MTR_Q_CTRL_TIME SPEED	14	速度制御周期の Q 値	
MTR_Q_CURRENT	13	電流の Q 値	
MTR_Q_VOLTAGE	13	電圧の Q 値	
MTR_Q_AFREQ	14	角周波数の Q 値	
MTR_Q_RESISTANC E	17	抵抗の Q 値	
MTR_Q_INDUCTANC E	17	インダクタンスの Q 値	
MTR_Q_FLUX	15	磁束の Q 値	
MTR_Q_RECI_FLUX	13	磁束の逆数の Q 値	
MTR_Q_INERTIA	9	イナーシャの Q 値	
MTR_Q_VMOD	12	PWM 変調率の Q 値(変更不可)	
MTR_Q_RECIV	13	電圧の逆数の Q 値	
MTR_Q_DIV_DSP	16	DSP 関数割り算の Q 値(変更不可)	
MTR_Q_SIN_COS_D SP	12	DSP 関数の三角関数の Q 値(変更不可)	

表 3-50 “r_mtr_scaling_parameter.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_Q_ACR_KP	18	電流 PI 比例ゲインの Q 値	
MTR_Q_ACR_KIDT	14	電流 PI 積分ゲイン * 制御周期の Q 値	
MTR_Q_ASR_KP	12	速度 PI 比例ゲインの Q 値	
MTR_Q_ASR_KIDT	15	速度 PI 積分ゲイン * 制御周期の Q 値	
MTR_Q_PLL_KP	14	PLL 比例ゲインの Q 値	
MTR_Q_PLL_KIDT	20	PLL 積分ゲイン * 制御周期の Q 値	
MTR_Q_SPEED_LPF_CO	14	速度 LPF ゲインの Q 値	

表 3-51 “main.h” マクロ定義一覧

マクロ	定義値	内容	備考
MODE_INACTIVE	0x00	インアクティブモード	
MODE_ACTIVE	0x01	アクティブモード	
MODE_ERROR	0x02	エラーモード	
SIZE_STATE	3	モード状態数	

表 3-52 “ICS_define.h” マクロ定義一覧

マクロ	定義値	内容	備考
RL78	-	CPU 定義	

表 3-53 “r_mtr_ics.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_ICS_DECIMATION	2	ICS 処理間引き数	
ICS_ADDR	0xFE00	ICS 用アドレス指定	
ICS_INT_LEVEL	2	ICS 用割り込みレベル設定	
ICS_NUM	0x40	ICS 通信データサイズ	
ICS_BRR	15	ICS ビットレートレジスタ選択	
ICS_MODE	0	ICS 割り込みモード設定	

表 3-54 “r_mtr_board.h” マクロ定義一覧

マクロ	定義値	内容	備考
SW_CHATTERING_CNT	10	チャタリング除去判別カウンタ数	
VR1_MARGIN	400	VR1 マージン値	
VR1_SCALING	$(CP_MAX_SPEED_RPM + VR1_MARGIN) / 0x0200$	VR1 速度指令値計算用定数	
VR1_OFFSET	0x1FF	VR1 オフセット調整用定数	
VR1_MIN_SPEED	100	VR1 の最低速度[rpm]	

表 3-55 “r_mtr_ctrl_rl78g1f.h” マクロ定義一覧 [1/2]

マクロ	定義値	内容	備考
MTR_INT_DECIMATION	1	割り込み処理キャリア間引き数	
MTR_PWM_TIMER_FREQ	64.0f	PWM タイマ周波数[kHz]	
MTR_INTVAL_TIMER_FREQ	32.0f	インターバル・タイマ周波数[kHz]	
MTR_CARRIER_FREQ	20.0f	キャリア割り込み周波数[kHz]	
MTR_INVTVAL_PERIOD	$(MTR_INT_DECIMATION + 1) * 1000.0f / (MTR_CARRIER_FREQ)$	インターバル・タイマ周期 [μs]	
MTR_DEADTIME	IP_DEADTIME	デッドタイム [μs]	
MTR_DEADTIME_CNT	$(int16_t)(MTR_DEADTIME * MTR_PWM_TIMER_FREQ)$	デッドタイム設定値	
MTR_CARRIER_CNT	$(uint16_t)(MTR_PWM_TIMER_FREQ * 1000 / MTR_CARRIER_FREQ * 0.5f)$	キャリア設定値	
MTR_HALF_CARRIER_CNT	$(uint16_t)(MTR_CARRIER_SET * 0.5f)$	キャリア設定値(中間値)	
MTR_CURRENT_ADCONV_TIME	6.0f	A/D 変換開始を進める時間[μs]	
MTR_CURRENT_SETTLING	0.42f	下アーム ON 後の電流安定時間	*
MTR_AD_TIME_ADJUST	2.55f	AD タイミング調整時間	
MTR_AD_TIME_CNT	$(uint16_t)(MTR_PWM_TIMER_FREQ * (MTR_PWM_LA_MIN_ONTIME * 0.5f))$	A/D 変換時間カウンタ値 g	
MTR_CENTER_AMPLITUDE_CNT	$(uint16_t)((MTR_CARRIER_CNT - (MTR_AD_TIME_CNT + MTR_DEADTIME_CNT)) * 0.5f + MTR_AD_TIME_CNT + MTR_DEADTIME_CNT)$	PWM タイマの振幅中心	
MTR_VOLTAGE_LIMIT_OFFSET	$(int16_t)((MTR_CURRENT_ADCONV_TIME + MTR_DEADTIME * 2) / (1000/MTR_CARRIER_FREQ)) * 0.5f * (1 << MTR_Q_VOLTAGE)$	電圧のオフセット制限値[PU(V)]	
MTR_VOLTAGE_ERROR	$(MTR_DEADTIME/1000000) * (MTR_CARRIER_FREQ * 1000)/2$	デッドタイム補償係数	
MTR_DEADTIME_CURRENT_LIMIT	$MP_NOMINAL_CURRENT_RMS * 0.1f$	電流制限値	
MTR_CTRL_PERIOD	$(MTR_INT_DECIMATION + 1) / (MTR_CARRIER_FREQ * 1000)$	電流制御周期	
MTR_SPEED_CTRL_PERIOD	0.001f	速度制御周期	
MTR_PORT_UP	P1_bit.no5	U 相 (正相) 電圧出力ポート	
MTR_PORT_UN	P1_bit.no4	U 相 (逆相) 電圧出力ポート	
MTR_PORT_VP	P1_bit.no3	V 相 (正相) 電圧出力ポート	
MTR_PORT_VN	P1_bit.no1	V 相 (逆相) 電圧出力ポート	
MTR_PORT_WP	P1_bit.no2	W 相 (正相) 電圧出力ポート	
MTR_PORT_WN	P1_bit.no0	W 相 (逆相) 電圧出力ポート	
MTR_PORT_ENC_A	P0_bit.no0	エンコーダ A 相入力ポート	
MTR_PORT_ENC_B	P0_bit.no1	エンコーダ B 相入力ポート	
MTR_PORT_ENC_Z	P5_bit.no0	エンコーダ Z 相入力ポート	
MTR_PORT_HALL_U	P1_bit.no2	ホールセンサ U 相入力ポート	
MTR_PORT_HALL_V	P1_bit.no3	ホールセンサ V 相入力ポート	
MTR_PORT_HALL_W	P1_bit.no4	ホールセンサ W 相入力ポート	
MTR_PORT_SW1	P12_bit.no5	SW1 入力ポート	
MTR_PORT_SW2	P13_bit.no7	SW2 入力ポート	
MTR_PORT_LED1	P14_bit.no1	LED1 出力ポート	
MTR_PORT_LED2	P14_bit.no0	LED2 出力ポート	
MTR_PORT_LED3	P0_bit.no4	LED3 出力ポート	

表 3-56 “r_mtr_ctrl_rl78g1f.h” マクロ定義一覧 [2/2]

マクロ	定義値	内容	備考
MTR_DUTY_U	TRDGRD0	タイマ RD ジェネラルレジスタ	
MTR_DUTY_V	TRDGRC1	タイマ RD ジェネラルレジスタ	
MTR_DUTY_W	TRDGRD1	タイマ RD ジェネラルレジスタ	
MTR_ADCCH_VR1	6	VR1 の A/D コンバータチャネル	
MTR_ADCCH_VDC	4	母線電圧の A/D コンバータチャネル	
MTR_ADCCH_VU	16	U 相電圧の A/D コンバータチャネル	
MTR_ADCCH_VV	0	V 相電圧の A/D コンバータチャネル	
MTR_ADCCH_VW	1	W 相電圧の A/D コンバータチャネル	
MTR_ADCCH_IU	2	U 相電流の A/D コンバータチャネル	
MTR_ADCCH_IV	19	V 相電流の A/D コンバータチャネル	
MTR_ADCCH_IW	3	W 相電流の A/D コンバータチャネル	
MTR_ADC_DATA_SHIFT	6	AD 変換値シフト量	
MTR_ADC_OFFSET	0x1FF	AD 変換値オフセット	
ERROR_NONE	0x00	エラーなし	
ERROR_CHANGE_CLK_TIMEOUT	0x01	クロック設定タイムアウトエラー	
ERROR_CHARGE_CAP_TIMEOUT	0x02	コンデンサ充電タイムアウトエラー	

表 3-57 “r_mtr_common.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_TWOPI	2*3.14159265359f	2π	
MTR_SQRT_3	1.7320508f	√3	
MTR_CW	0	CW	
MTR_CCW	1	CCW	
MTR_ON	0	オン	
MTR_OFF	1	オフ	
MTR_CLR	0	フラグクリア	
MTR_SET	1	フラグセット	

表 3-58 “r_mtr_parameter.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_PWM_DUTY_RANGE	4095	Duty レンジ	
MTR_INPUT_V	IP_INPUT_V	入力電圧	
MTR_HALF_VDC	MTR_INPUT_V * 0.5f	電圧の 50%	
MTR_MCU_ON_V	MTR_INPUT_V * 0.8f	電圧の 80%	
MTR_OVERVOLTAGE_LIMIT	IP_OVERVOLTAGE_LIMIT	過電圧リミット値	
MTR_UNDERVOLTAGE_LIMIT	IP_UNDERVOLTAGE_LIMIT	低電圧リミット値	
MTR_ANGLE_RANGE	(int16_t)(MTR_TWOPi * PU_SF_ANGLE * 4096)	角度レンジ 2π	
MTR_ANGLE_HALF_RANGE	(int16_t)(MTR_ANGLE_RANGE/2)	角度レンジ π	
MTR_ANGLE_QUAT_RANGE	(int16_t)(MTR_ANGLE_RANGE/4)	角度レンジ $\pi/2$	
MTR_OVERCURRENT_LIMIT	CP_OC_LIMIT	電流リミット値	
MTR_I_LIMIT_VD	IP_INPUT_V * 0.5f	Vd の電流 PI リミット	
MTR_I_LIMIT_VQ	IP_INPUT_V * 0.5f	Vq の電流 PI リミット	
MTR_RPM_RAD	(MP_POLE_PAIRS * MTR_TWOPi) / 60.0f	[rpm]から[rad/s]への変換	
MTR_SPEED_LIMIT_RAD	CP_SPEED_LIMIT_RPM * MTR_RPM_RAD	速度制限値[rad/s]	
MTR_MAX_SPEED_RAD	CP_MAX_SPEED_RPM * MTR_RPM_RAD	最大速度[rad/s]	
MTR_LIMIT_IQ	MP_NOMINAL_CURRENT_RMS * MTR_SQRT_3	速度 PI の出力制限値	
MTR_I_LIMIT_IQ	MP_NOMINAL_CURRENT_RMS * MTR_SQRT_3	速度 PI の積分項の出力制限値	
MTR_CL2OL_SPEED_RAD	CP_LESS2OL_SPEED_RPM * MTR_RPM_RAD	センサレスからオープンループ への切り替え速度 [rad/s]	
MTR_OL2CL_SPEED_RAD	CP_OL2LESS_SPEED_RPM * MTR_RPM_RAD	オープンループからセンサレス への切り替え速度[rad/s]	
MTR_RECI_FLUX	1.0f/(MP_MAGNETIC_FLUX * PU_SF_FLUX)	磁束の逆数[PU]	

表 3-59 “r_mtr_sc_table.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_PI_2	1024	$\pi/2$	
MTR_PI	2048	π	
MTR_3PI_2	3072	$3\pi/2$	

表 3-60 “r_mtr_statemachine.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_MODE_INIT	0x00	初期化モード	
MTR_MODE_DRIVE	0x01	ドライブモード	
MTR_MODE_STOP	0x02	ストップモード	
MTR_SIZE_STATE	3	ステート数	
MTR_EVENT_STOP	0x00	ストップイベント	
MTR_EVENT_DRIVE	0x01	ランイベント	
MTR_EVENT_ERROR	0x02	エラーイベント	
MTR_EVENT_RESET	0x03	リセットイベント	
MTR_SIZE_EVENT	4	イベント数	

表 3-61 “r_mtr_foc_less_speed.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_CONTROL_CURRENT	0x01	電流制御	
MTR_CONTROL_SPEED	0x02	速度制御	
MTR_CONTROL_POSITION	0x04	位置制御	
MTR_CONTROL_TORQUE	0x08	トルク制御	
MTR_CONTROL_VOLTAGE	0x10	電圧制御	
MTR_ERROR_NONE	0x00	エラーなし	
MTR_ERROR_OVER_CURRENT	0x01	過電流エラー	
MTR_ERROR_OVER_VOLTAGE	0x02	過電圧エラー	
MTR_ERROR_OVER_SPEED	0x04	過速度エラー	
MTR_ERROR_HALL_TIMEOUT	0x08	ホールタイムアウトエラー	
MTR_ERROR_BEMF_TIMEOUT	0x10	誘起電圧タイムアウトエラー	
MTR_ERROR_HALL_PATTERN	0x20	ホールパターンエラー	
MTR_ERROR_BEMF_PATTERN	0x40	誘起電圧パターンエラー	
MTR_ERROR_UNDER_VOLTAGE	0x80	低電圧エラー	
MTR_ERROR_UNKNOWN	0xff	未定義エラー	
MTR_ID_ZERO_CONST	0	d 軸電流 0 制御	
MTR_ID_MANUAL	1	d 軸電流マニュアル制御	
MTR_IQ_ZERO_CONST	0	q 軸電流 0 制御	
MTR_IQ_MANUAL	1	q 軸電流マニュアル制御	
MTR_IQ_SPEED_PI_OUTPUT	2	速度 PI 制御出力	
MTR_SPEED_ZERO_CONST	0	速度 0 制御	
MTR_SPEED_MANUAL	1	速度マニュアル制御	
MTR_OFFSET_CALC_EXE	0	オフセット除去実行	
MTE_OFFSET_CALC_END	1	オフセット除去終了	
MTR_OFFSET_CALC_DONE	2	オフセット除去処理完了	

表 3-62 “r_mtr_est_phase_err.h” マクロ定義一覧

マクロ	定義値	内容	備考
MTR_TWOPI_Q	(int16_t)(4096.0f * MTR_TWOPI)	$2\pi \times (\text{角度の Q 値})$	
MTR_INV_TWOPI_Q	(int16_t)(4096.0f / MTR_TWOPI)	$1/2\pi \times (\text{角度の Q 値})$	

3.6 制御フロー（フローチャート）

3.6.1 メイン処理

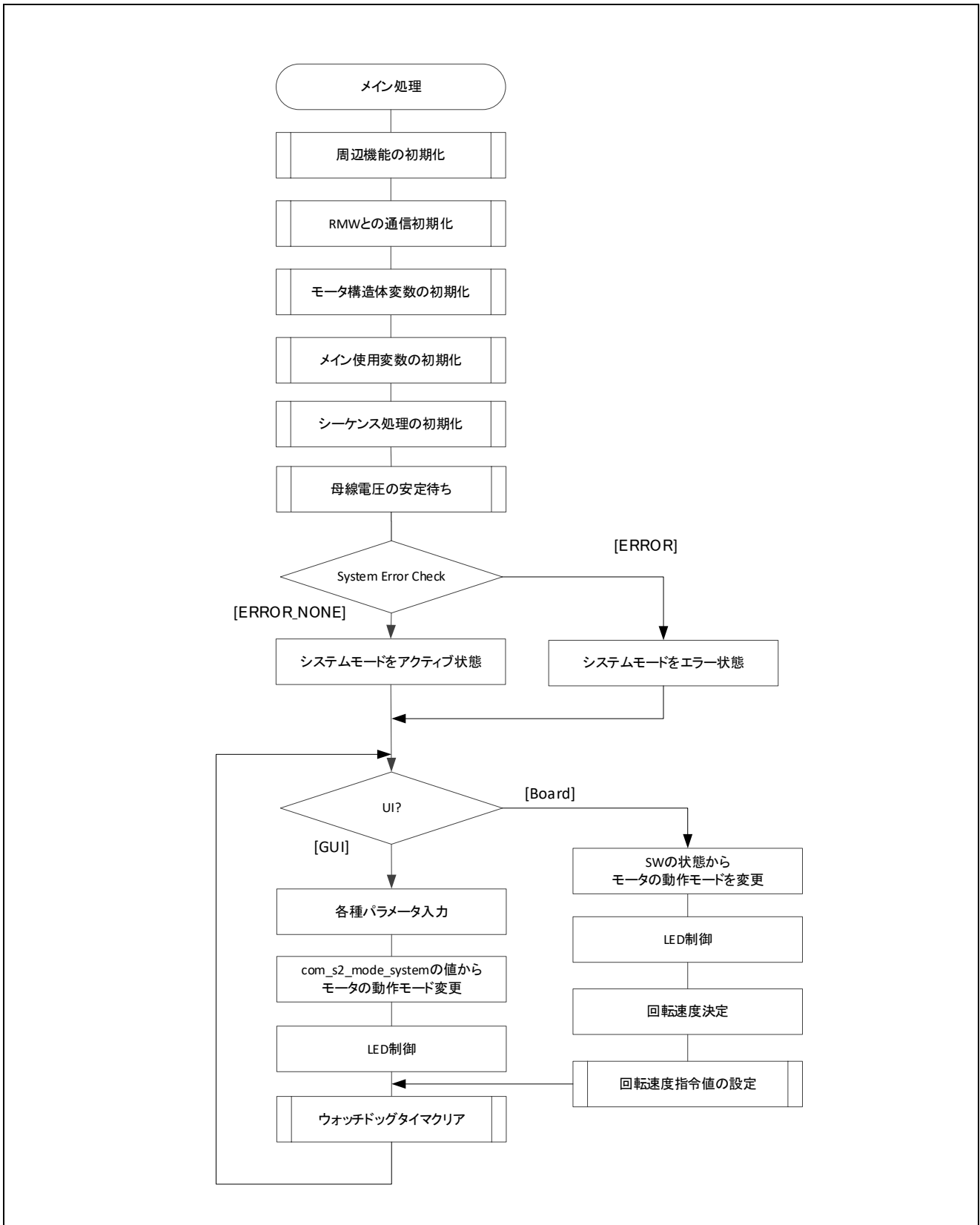


図 3-7 メイン処理フローチャート

3.6.2 100 [μs]周期割り込み処理

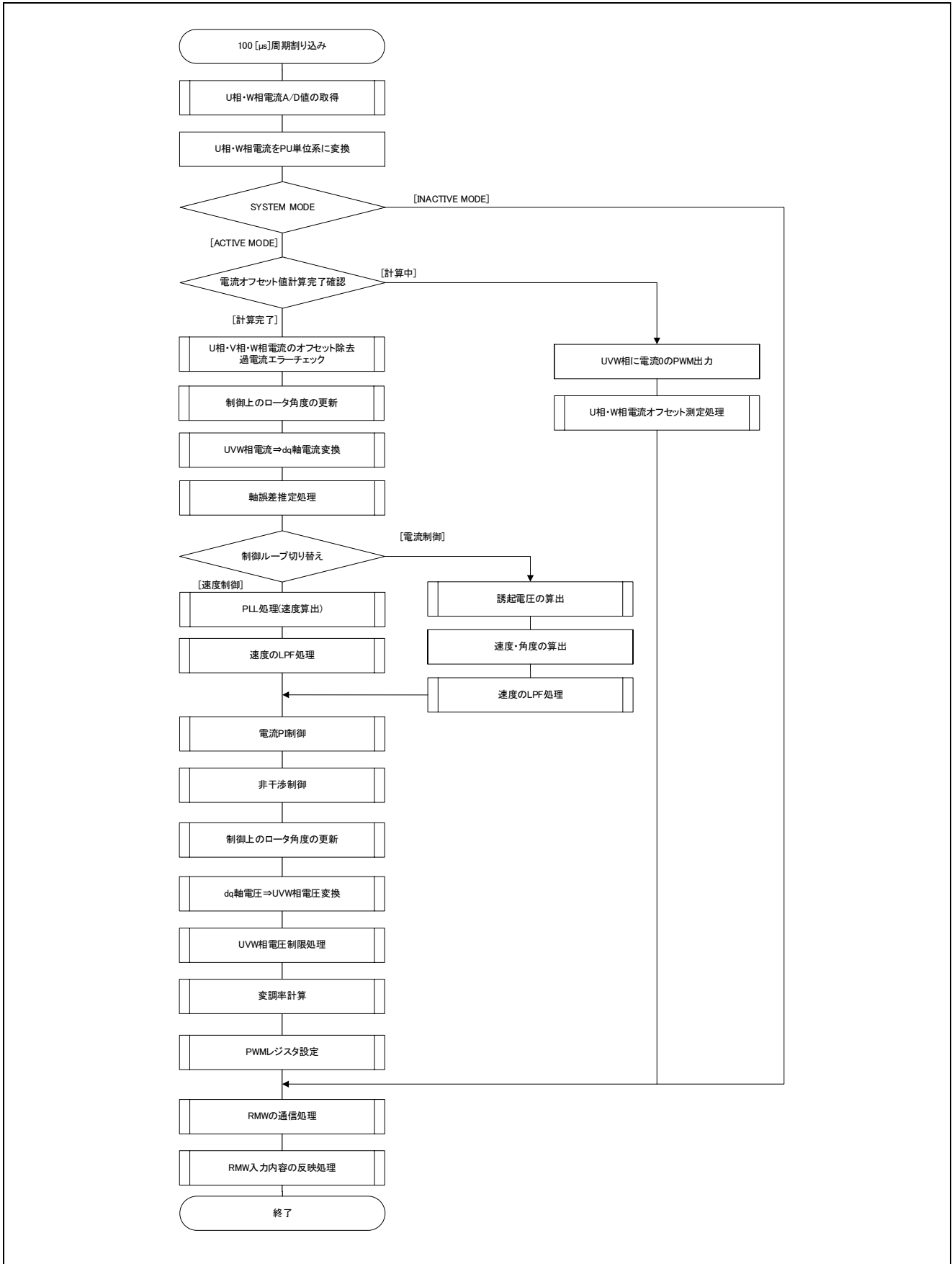


図 3-8 キャリア周期割り込み処理フローチャート

3.6.3 1 [ms]割り込み処理

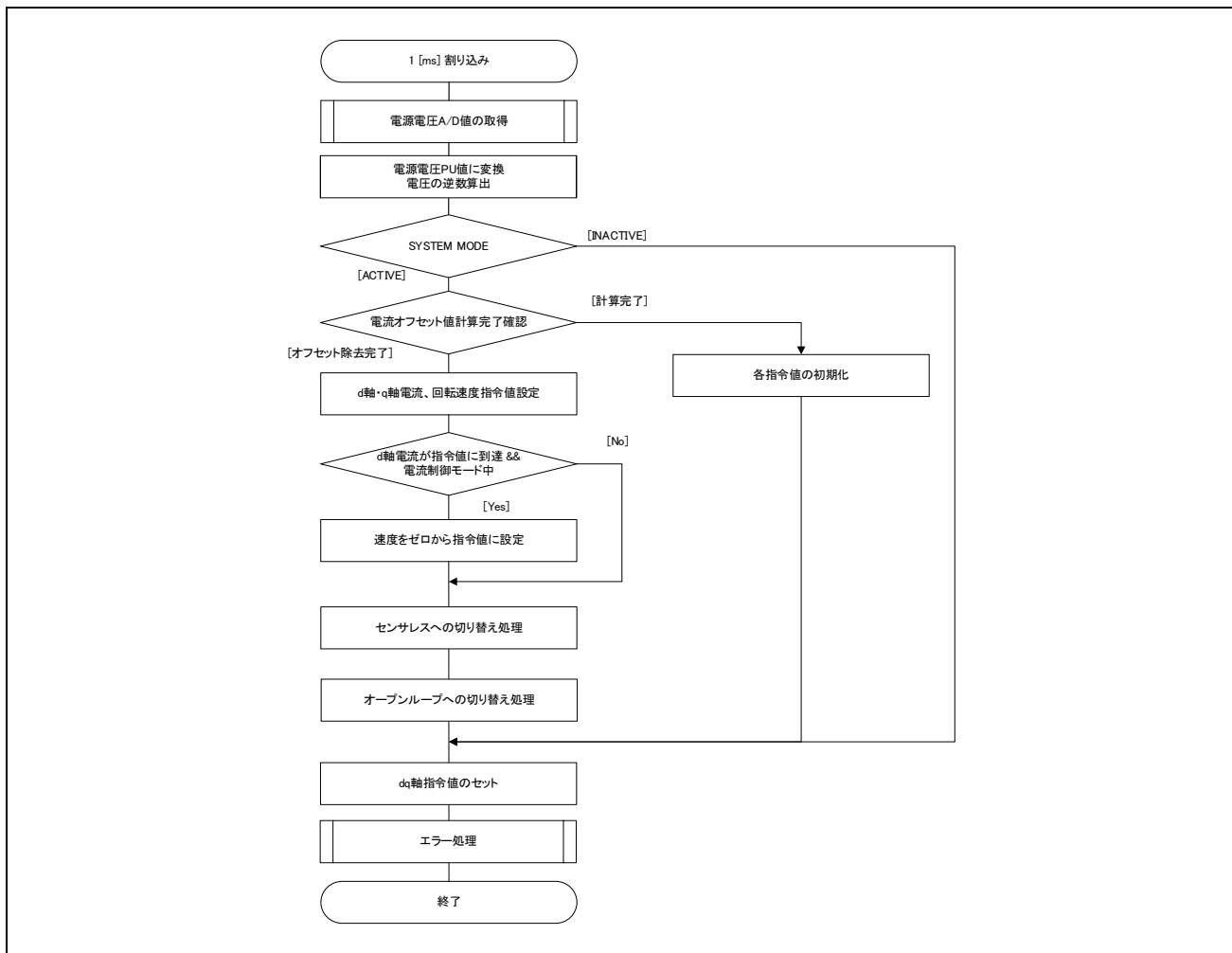


図 3-9 1 [ms]割り込み処理フローチャート

3.6.4 過電流割り込み処理

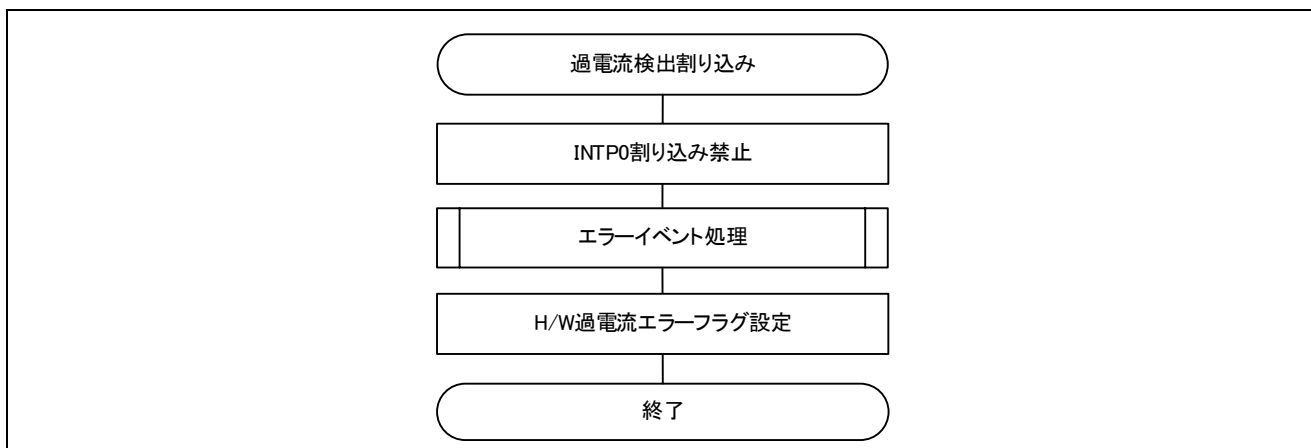


図 3-10 過電流検出割り込み処理フローチャート

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

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 V 2.0 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。

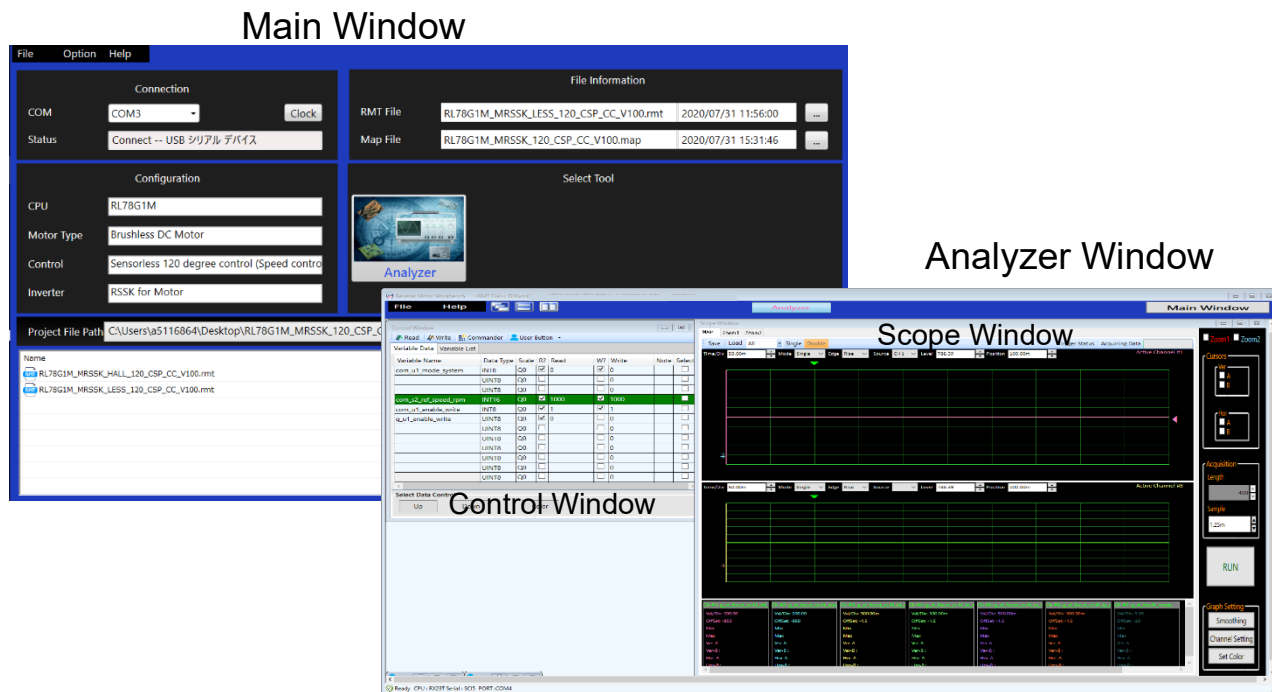
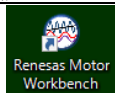


図 4-1 Renesas Motor Workbench 外観

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 Window の MENU バーから、[File] → [Open RMT File(O)]を選択。
” [プロジェクトフォルダ]/application/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 右側の Select Tool の Analyzer ボタンをクリックし、Analyzer 機能を起動する。
(起動すると Analyzer Window 画面に切り替わります。)
- ⑤ ” Analyzer 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 ユーザインタフェース使用時の入力用変数一覧を表 4-1 に示します。なお、これらの変数の値は com_s2_enable_write に g_s2_enable_write と同じ値を書き込んだ際に反映されます。ただし、(*)が付けられた変数は com_s2_enable_write に依存しません。

表 4-1 Analyzer 機能入力用変数一覧

変数名	型	内容	備考 ([]: 反映先の変数名)
com_u1_run_event (*)	uint8_t	ランモード遷移用変数 0: ストップイベント 1: ドライブイベント 2: エラーイベント 3: リセットイベント	[g_u1_run_event]
com_s2_sw_userif (*)	int16_t	UI 管理用変数 0: ICS UI (default) 1: Board UI	[g_s2_sw_userif]
com_u1_direction	uint8_t	回転方向 0 : CW 1 : CCW	[gst_foc.st_asr.u1_ref_dir]
com_f4_mtr_r	float	抵抗 [Ω]	[gst_foc.st_motor.s2_mtr_r]
com_f4_mtr_ld	float	d 軸インダクタンス[H]	[gst_foc.st_motor.s2_mtr_ld]
com_f4_mtr_lq	float	q 軸インダクタンス[H]	[gst_foc.st_motor.s2_mtr_lq]
com_f4_mtr_m	float	誘起電圧計数 [Vs/rad]	[gst_foc.st_motor.s2_mtr_m]
com_f4_mtr_j	float	イナーシャ[kg m ²]	[gst_foc.st_motor.s2_mtr_j]
com_u2_mtr_pp	uint16_t	極対数	[gst_foc.st_motor.s2_mtr_pp]
com_u2_offset_calc_time	uint16_t	電流オフセット検出時間	[gst_foc.st_tscs.u2_offset_calc_time]
com_s2_ref_speed_rpm	int16_t	目標速度 [rpm]	[gst_foc.st_asr.s2_ref_speed_rad]
com_f4_ramp_limit_speed_rpm	float	加速度制限値 [rpm/ms]	[gst_foc.st_asr.s2_ramp_limit_speed_rad]
com_s2_max_speed_rpm	int16_t	最大速度 [rpm]	[gst_foc.st_asr.s2_max_speed_rad]
com_f4_acr_nf_hz	float	電流 PI 制御固有周波数 [Hz]	[gst_foc.st_acr.st_pi_id.s2_kp] [gst_foc.st_acr.st_pi_id.s2_kidt] [gst_foc.st_acr.st_pi_iq.s2_kp] [gst_foc.st_acr.st_pi_iq.s2_kidt]
com_f4_asr_nf_hz	float	速度 PI 固有周波数 [Hz]	[gst_foc.st_asr.st_pi.s2_kp] [gst_foc.st_asr.st_pi.s2_kidt]
com_f4_asr_lpf_cof_hz	float	速度 LPF 固有周波数 [Hz]	[gst_foc.st_acr.st_iq_lpf.s2_in_k] [gst_foc.st_acr.st_iq_lpf.s2_out_k]
com_f4_pll_nf_hz	float	PLL 固有周波数 [Hz]	[gst_foc.st_pll.st_pi.s2_kp] [gst_foc.st_pll.st_pi.s2_kidt]
com_s2_cl2ol_speed_rpm	int16_t	センサレスからオープンループへの切り替え速度[rpm]	[gst_foc.st_asr.s2_less2ol_speed_rad]
com_s2_ol2cl_speed_rpm	int16_t	オープンループからセンサレスへの切り替え速度[rpm]	[gst_foc.st_asr.s2_ol2less_speed_rad]
com_f4_ol_ref_id	float	オープンループ d 軸指令電流[A]	[gst_foc.st_acr.s2_ol_ref_id]
com_f4_init_asr_intg	float	センサレス移行時の ASR の積分項初期値	[gst_foc.st_asr.s2_init_intg]
com_f4_ramp_limit_current	float	電流の上昇制限値[A/ms]	[gst_foc.st_acr.s2_ramp_limit_current]
com_s2_enable_write	int16_t	変数書き換え許可	[g_s2_enable_write]

4.3 Analyzer 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、図 4-1 で示す “Control Window” で行います。“Control Window” の詳細は、「Renesas Motor Workbench V 2.0 ユーザーズマニュアル」を参照して下さい。

- モータを回転させる
 - ① “com_u1_run_event”, “com_s2_ref_speed_rpm”, “com_s2_enable_write” 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 “com_u2_ref_speed_rpm” の [Write] 欄に入力する。
 - ③ “Write” ボタンを押す。
 - ④ “Read” ボタンを押して現在の “com_s2_ref_speed_rpm”, “g_s2_enable_write” の [Read] 欄を確認する。
 - ⑤ MCU 内の変数値へ反映させるため、“com_s2_enable_write” に④で確認した “g_s2_enable_write” と同じ値を入力する。
 - ⑥ “com_u1_run_event” の [Write]欄に “1” を入力する。
 - ⑦ “Write” ボタンを押す。

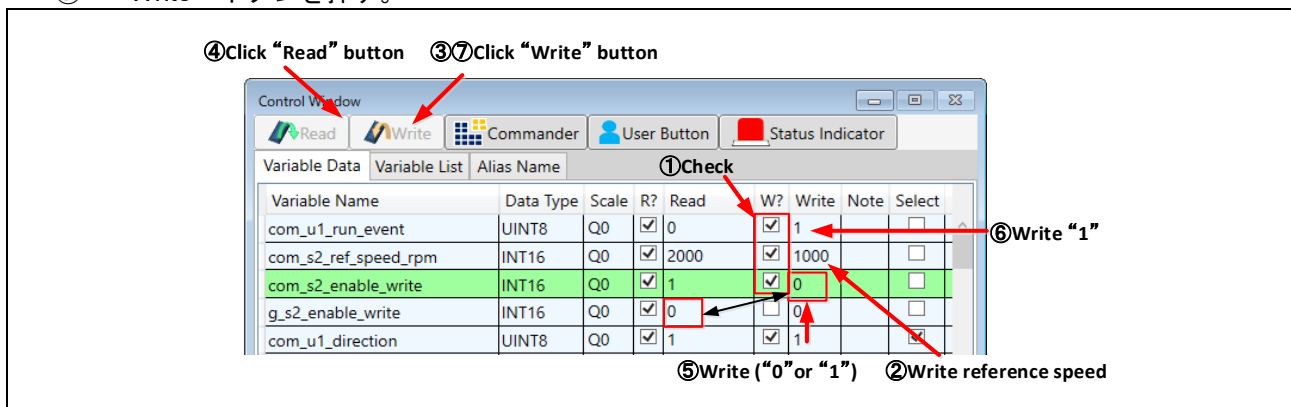


図 4-2 モータ回転の手順

- モータを停止させる
 - ① “com_u1_run_event” の [Write]欄に “0” を入力する。
 - ② “Write” ボタンを押す。

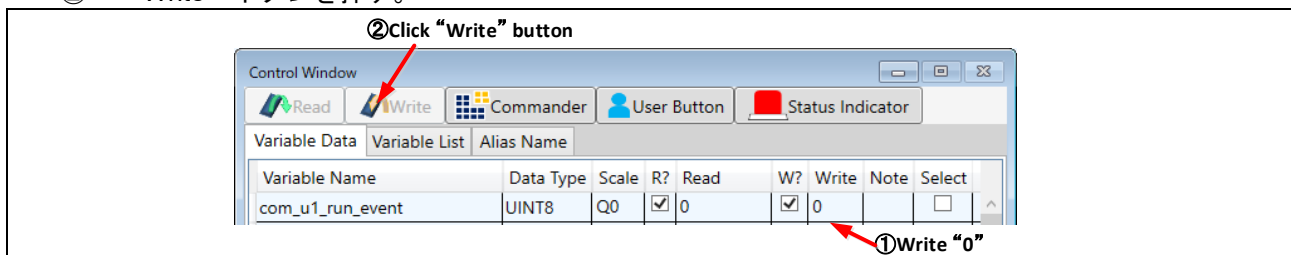


図 4-3 モータ停止の手順

- 止まってしまった（エラー）場合の処理
 - ① “com_u1_run_event” の [Write]欄に “3” を入力する。
 - ② “Write” ボタンを押す。

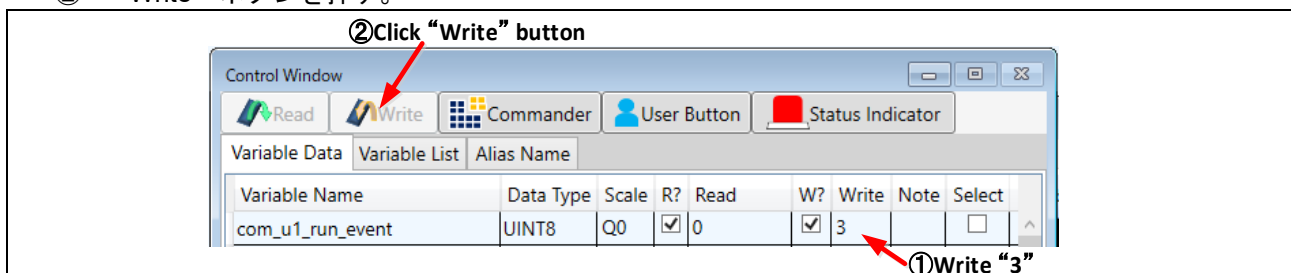


図 4-4 エラー解除の手順

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017/07/31	—	新規発行
2.00	2020/10/20	—	制御アルゴリズムとスケーリングを変更

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ーやマガジンケース、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、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プ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、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えば、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、ノイズ輻射量などが異なる場合があります。型名が違えば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。お客様の機器・システムの設計において、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には、お客様の責任において行ってください。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まれます。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、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、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。
5. 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）、交通管制（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も、当社は一切その責任を負いません。
6. 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。
 7. 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。
 8. 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。
 9. 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。
 10. 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となります。
 11. 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。
 12. 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.4.0-1 2017.11)

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

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/

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。