

PM モータのホールセンサベクトル制御

RX ファミリ

要旨

本サンプルプログラムは、ホールセンサが搭載された 3 相の永久磁石型同期モータ (PM モータ) を、停止状態から低速・中高速域までの全速度域でホールセンサを位置・速度センサ (通常はエンコーダ) として用いたベクトル制御 (Hall FOC) 機能を提供します。

ホールセンサから出力される 3 つの信号から、電気角 1 周 6 パルスを生成し、微分積分法を用いて線形補完を行って磁極位置を求め、センサ付ベクトル制御を実現しています。また、速度制御モードにおいては低分解能エンコーダ特有の現象として、速度調節器の積分項が飽和しやすいことから、始動特性や低速域の改善のための摩擦補償を導入し、特性改善を行うことができます。

従来のサンプルプログラムでは、停止状態のみホールセンサの磁極位置検出を行い、低速域ではオープンループによる電流引き込み制御、中高速域は誘起電圧オブザーバによるセンサレスベクトル制御の機能を提供していました。本サンプルプログラムでは、停止状態からのトルク出力性能や始動特性を改善する機能を追加しています。

本書では、RX26T マイクロコントローラを用いて、DC24V 系のインバータ、ならびに PM モータを対象としたサンプルプログラムを解説します。

図 1-1 に、本サンプルプログラムで動作させた際の加減速・正転逆転特性を示します。

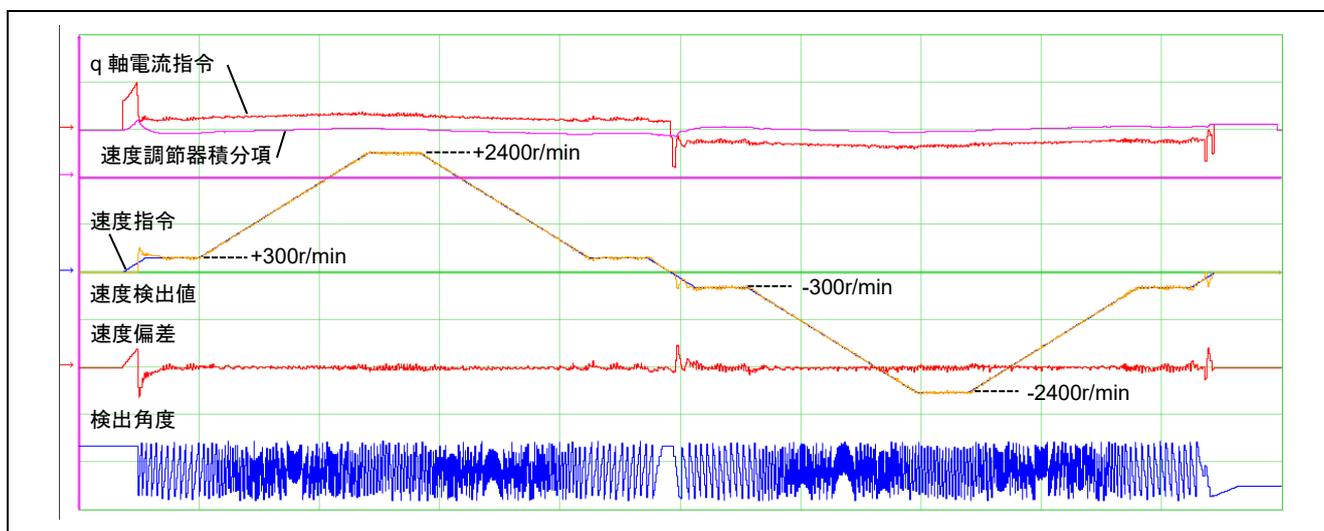


図 1-1 加減速および正転逆転特性

動作確認デバイス

本アプリケーションノート対象ソフトウェアの動作確認は下記のデバイスで行っています。

- ・ RX26T RAM64KB バージョン (R5F526TFCDP)

目次

1.	はじめに	5
2.	用語集	7
3.	使用機材・使用ソフトウェア	8
3.1	使用ハードウェアの一覧	8
3.2	使用ソフトウェアの一覧	8
4.	ハードウェア環境構築方法	9
4.1	ハードウェア環境の概要	9
4.2	電源の準備	9
4.3	モータ及び負荷機の準備	9
4.4	インバータの準備	10
4.5	ホールセンサ線の接続	11
4.6	RX26T CPU ボードのセットアップ	11
4.7	キット(MCK-RX26T)の接続例	12
4.8	オンボードデバッグ	12
4.9	配線方法	13
4.10	測定器の利用	14
5.	ソフトウェア環境構築方法	15
5.1	CS+を使用する場合	15
5.2	e ² Studio を使用する場合	15
6.	運転方法	16
6.1	運転前の注意点	16
6.2	運転までの手順	16
6.3	接続方法	17
6.4	サンプルプログラムの書き込み	17
6.5	RMW の導入方法	18
6.6	Map ファイルの登録更新	19
6.7	RMW の操作に使用する変数	20
6.8	モータ操作方法(RMW)	21
6.9	モータ操作方法(ボード UI)	23
6.10	モータ停止・遮断方法	24
7.	モータ制御アルゴリズム	25
7.1	概要	25
7.2	制御ブロック図	25
7.3	速度制御機能	25
7.4	弱め磁束制御	26
7.5	電流制御機能	27
7.6	非干渉制御	27
7.7	ホールセンサベクトル制御機能	28
7.7.1	概要	28

7.7.2	ホールセンサ信号とホールパターン番号	28
7.7.3	速度検出方法	30
7.7.4	角度検出方法	31
7.7.5	摩擦補償	33
7.8	電圧位相進み補償	36
7.9	電圧誤差補償	36
7.10	PWM 変調方式	37
8.	ソフトウェア仕様・構成	38
8.1	ソフトウェア仕様	38
8.2	ソフトウェア全体構造	39
8.3	タスクの説明	40
8.4	ファイル・フォルダ構成	41
8.5	アプリケーション層	43
8.5.1	機能	43
8.5.2	コンフィグレーション情報	43
8.5.3	構造体・変数情報	44
8.5.4	マクロ定義	46
8.5.5	パラメータ調整・設定	46
8.6	マネージャモジュール・モータ制御モジュール	47
8.6.1	機能	47
8.6.2	モジュール構成図	48
8.6.3	モード管理	49
8.6.4	シーケンスの説明	50
8.6.5	始動	50
8.6.6	保護機能	51
8.6.7	API	52
8.6.8	構造体・変数情報	54
8.6.9	マクロ定義	60
8.7	ドライバモジュール	61
8.7.1	機能	61
8.7.2	モジュール構成図	61
8.7.3	API	62
8.7.4	コンフィグレーション情報	62
8.7.5	構造体・変数情報	63
8.7.6	パラメータ調整・設定	64
9.	パラメータの設定	65
9.1	概要	65
9.2	MCU 関連パラメータ	65
9.3	制御機能の設定パラメータの一覧	66
9.4	保護関連パラメータ	67
9.5	PWM キャリア周波数の変更	67
9.6	パルス変調方法の設定	68
9.7	インバータパラメータ	69
9.7.1	概要	69
9.7.2	電流検出ゲイン	70

9.7.3	電圧検出ゲイン	71
9.7.4	電圧誤差補償パラメータ	71
9.8	モータパラメータ	72
9.9	電流制御パラメータ	75
9.10	速度制御パラメータ	76
9.11	電圧位相進み補償パラメータ	77
9.12	ホールセンサベクトル制御パラメータ	78
10.	スマート・コンフィグレータ設定	80
10.1	クロック設定	80
10.2	コンポーネント設定	80
10.3	AD 設定	81
10.4	モータ設定	81
10.5	割り込み	82
10.6	ユーザコード詳細	82
10.7	POE 設定	83
10.8	端子設定	83
11.	評価結果	84
11.1	モータ制御評価	84
11.1.1	加減速特性	84
11.1.2	負荷特性	85
11.2	CPU 使用率	85
11.3	プログラムサイズ・RAM 使用量	85
12.	FAQ	86
12.1	こんなときは	86
	改訂記録	88

1. はじめに

本アプリケーションノートはルネサス製マイクロコントローラ(MCU)である RX26T を使用し、永久磁石同期モータを停止から低速も含めて全速度域をホールセンサを用いてベクトル制御で駆動するサンプルプログラム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、ルネサス製モータコントロールキット MCK-RX26T を用いて、DC24V 系の PM モータである MOONS'製 R42BLD30L3 を、ホールセンサを用いたベクトル制御でモータ制御することができます。また、モータ制御開発支援ツール「Renesas Motor Workbench」に対応しており MCU の内部データ確認や、モータ制御のユーザインタフェース(UI)として使用可能です。サンプルプログラムの MCU 機能割り当てや、制御の割り込み負荷状況などを参照頂くことで、使用する MCU の選定やソフトウェア開発の参考としてご活用ください。

本アプリケーションノートは、本書内で記載している PM モータ・インバータ環境で開発・評価を行ったものであり、ユーザの使用する PM モータやインバータ環境で動作することを保証するものではありません。電流センサや、ホールセンサの取り付け誤差、信号経路の基板パターン設計、サンプリング・分解能・フィルタの仕様やモータの磁気飽和特性・個体差により、制御性能に制約が生じる場合があります。ユーザの責任の下で、アルゴリズムの改良や、パラメータのチューニングが必要となります。

本アプリケーションノートに記載している機材・機器については、各機器メーカーによる廃盤や改訂等により入手できない場合があります。予めご了承ください。

評価に使用した主な装置・機器

モータコントロールキット：ルネサスエレクトロニクス（株）製 MCK-RX26T (Type-A)
モータ：MOON'S 製 R42BLD30L3

対象ソフトウェア

本アプリケーションノート対象ソフトウェアを下記に示します。

- ・RX26T_MCBA_MCILV1_SPM_HALL_FOC_GSP_V100 (IDE : CS+)
- ・RX26T_MCBA_MCILV1_SPM_HALL_FOC_E2S_V100 (IDE : e²studio)

参考資料

- ・RX26T グループ ユーザーズマニュアル ハードウェア編 (R01UH0979)
- ・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)

本アプリケーションノートを使用いただく際に、よく確認される内容について、対応する章を以下にまとめています。

表 1-1 確認したい内容と対応章の一覧

確認したい内容	対応する章
必要な機材を確認・選定する	3
電源を選定する	4.2
モータを選定する	4.3
インバータを選定する	4.4
配線を確認する	4.9
サンプルプログラムの開発環境を準備する	5
マイコンにサンプルプログラムを書き込む	6.4
PCにモータを運転するソフトウェアを導入する	6.5
サンプルプログラムを変更した後、RMWに変更点を反映する	6.6
インバータの内部情報をPC上で確認する	6.7
モータを運転する	6.8
運転中のモータを停止する	6.9
モータ制御アルゴリズムを調べる	7
サンプルプログラムの構造を調べる	8
インバータパラメータを確認、変更する	9.7、9.4、10.3、10.4、 10.7、10.8
モータパラメータを確認、変更する	9.8、9.4
ホールセンサベクトル制御の制御性能調整を行う	9.12
PWM キャリア周波数の変更	9.5
ホールセンサの設定を変更したい	9.12
マイコンの設定を変更したい	9.2,10
よくある質問を確認する トラブルが起きた場合の対応を確認したい	12

2. 用語集

本書で、使用されている主な用語と、その説明を、以下に示します。

表 2-1 用語集

用語	説明
E2OB	E2 On-Board のこと。E2Lite の機能を CPU ボード上に持つ機能です。
IDE	統合開発環境のこと。e ² studio、CS+等を指す。
MC-COM	波形表示用の接続治具・ツールのことを示します。詳細は、以下の URL を参照ください。 https://www.renesas.com/ja/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rtk0emxc90s00000bj-mc-com-renesas-flexible-motor-control-communication-board
RMW	Renesas Motor Workbench と呼ばれる、モータ制御に特化した操作ソフトウェアのこと。
SPM モータ	SPMSM とも呼ばれる。低速でも滑らかな動きが要求される、サーボモータに使用されます。
インバータ母線電圧	インバータ回路に入力される直流電圧のこと。直流中間電圧とも呼ばれる。
エミュレータ	MCU に書き込むための装置のこと。ICE とも呼ばれます。
オープンループ	位置のフィードバック信号なしで、電圧制御を行うモータ制御方式のこと。
コンポーネント	スマート・コンフィグレータで生成された、MCU 周辺機能を使用しやすくするドライバモジュールのこと。
センサレス	本書では、「磁極位置センサや速度センサがないこと」を示します。位置センサや速度センサは、コスト面や耐環境性等に弱点を持ち、センサを省略する事がメリットとされています。
磁気飽和	一定以上の電流を流すことで、磁氣的にモータが飽和し、磁束が強まらない現象。パラメータが変化するため、インバータによるモータ制御に影響が出ます。
表面磁石型同期モータ	SPM モータのこと
電気角	モータに流れる出力電流の位相角度のこと。モータの極対数で割ると、機械角に換算できる。
機械角	モータ軸の回転角度のこと。軸が 1 分に 1 回転で 1r/min となる。

3. 使用機材・使用ソフトウェア

3.1 使用ハードウェアの一覧

本サンプルプログラムの評価に使用した機器の一覧を以下に示します。

表 3-1 使用機器一覧

機器	メーカー	型式
MCK-RX26T Renesas Flexible Motor Control Kit for RX26T MCU Group	Renesas	RTK0EMXE70S00020BJ
CPU ボード MCB-RX26T Type A (上記キットに同梱)	Renesas	RX26T RAM64KB バージョン (R5F526TFCDFP) / RTK0EMXE70C00000BJ
インバータボード MCI-LV-1 (上記キットに同梱)	Renesas	RTK0EM0000B12020BJ 注：定格電流は 10[A]ですが、5[A]以上を流すときにはインバータに放熱フィンを取り付けることを強く推奨します。
MC-COM (上記キットに同梱)	Renesas	Renesas Flexible Motor Control Communication Board RTK0EMXC90Z00000BJ
モーター	MOONS'	R42BLD30L3
直流電源	COSEL	24V,11A
パワーメータ	横河計測	WT500

3.2 使用ソフトウェアの一覧

本サンプルプログラムの評価で使用したソフトウェアと、そのバージョンを以下に示します。本サンプルプログラムは、弊社開発環境である CS+または e² studio の無償評価版の制限範囲で、ご利用いただけません。

表 3-2 使用ソフトウェア一覧

メーカー	ソフトウェア	バージョン	備考
Renesas	CS+	V8.12.00	無償評価版
Renesas	e ² studio	2024-10	無償版
Renesas	RX スマート・コンフィグレータ	V2.23.0	
Renesas	CC-RX	V3.06.00	無償評価版
Renesas	Renesas Motor Workbench	3.1.2	

4. ハードウェア環境構築方法

4.1 ハードウェア環境の概要

本サンプルプログラムを使用し、PM モータを動かすためのハードウェア環境について説明します。図 4-1 に、ハードウェア構成例を示します。電源(4.2)と、モータ及び負荷機(4.3)、インバータ(4.4), RX26T CPU ボード(4.6)及びモニタリング・書き込み装置(6)について、詳細を次の項から説明します。

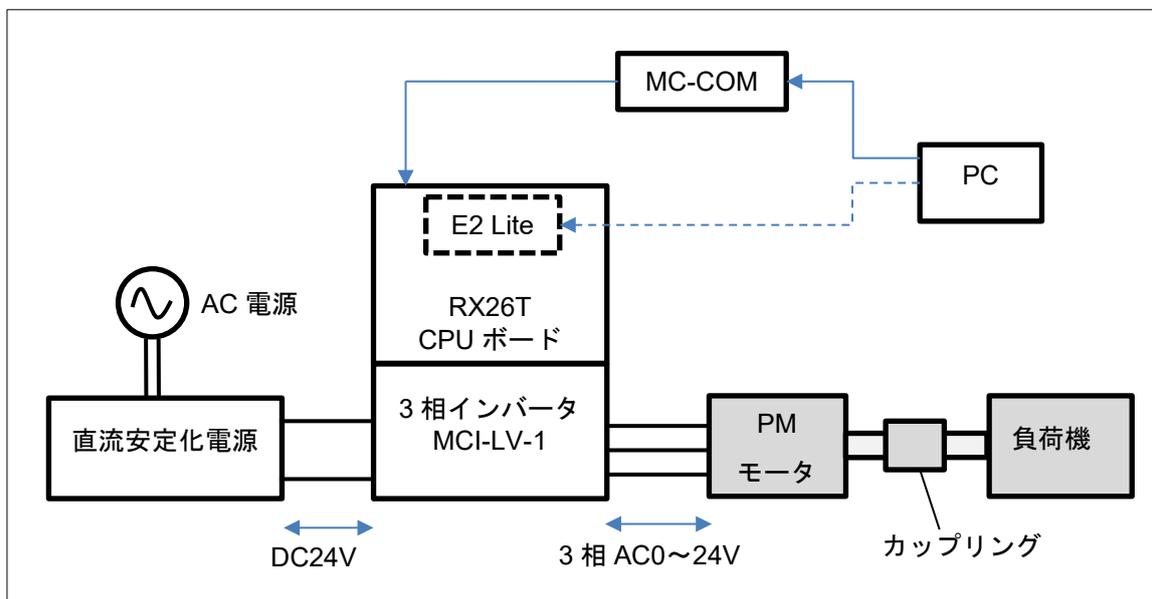


図 4-1 ハードウェア構成例

4.2 電源の準備

本サンプルプログラムでは、直流安定化電源、または AC アダプタ、汎用制御電源(24V,10A 以上の出力が可能なもの)を用いて、単相 AC100V から 3 相インバータ MCI-LV-1 に DC24V を供給しています。このサンプルプログラムでは、低速高トルクに十分な電源容量が必要なため、USB 電源は推奨していません。

インバータに供給する電圧は、使用するモータの誘起電圧や定格条件、最大負荷条件によって変わります。ユーザの実験環境や、使用する電源の制約や条件に応じて、電源の種類を適切に選定してください。なお、ここで紹介するインバータは、出力電流が 10A_{max} となっており、モータを変更した場合、モータの定格条件によっては定格運転ができない場合があります。予めご了承ください。

4.3 モータ及び負荷機の準備

インバータとモータの配線を行う前に、ベクトル制御で動かすために必要となる、PM モータのパラメータ・定数を LCR メータ等の測定器などを用いて取得してください。また必要に応じて、PM モータのメーカーにパラメータの情報を得るために問い合わせをしてください。

モータパラメータを変更した場合、電流調節器や速度調節器、ホールセンサベクトル制御のパラメータを、モータに合わせて変更を行う必要があります。

- 定格値（電流、電圧、速度、極数）
- Ld、Lq、抵抗値
- 誘起電圧、鎖交磁束
- モータ及びモータの軸に繋がっている負荷装置のイナーシャ（慣性モーメント）

弊社で調査を行った、MOONS'製 R42BLD30L3 のモータパラメータを表 4-1 に示します。弊社で独自に測定したものであり、得られるパラメータには測定条件によるばらつきや個体差があります。このパラメータ

タは、正確性や性能を保証するものではありません。また、負荷電流によって生じる磁気飽和現象により、運転中にモータパラメータの値が変化し、位置推定精度や運転性能に影響する場合があります。

制御性能の確認のため、モータの負荷試験を行う場合、PM モータの他に、負荷試験機（負荷機）または、負荷試験が可能な実負荷装置が必要となります。モータの要求負荷容量に応じて、適切な負荷機を選定し、ユーザでご用意ください。

表 4-1 MOONS'製 R42BLD30L3 のモータパラメータ

一次抵抗 R	1.34 Ω
d 軸インダクタンス	1.3 mH
q 軸インダクタンス	1.3 mH
慣性モーメント	0.000003666 kgm ²
磁束鎖交数 Ψ	0.01119 Wb(rms)
極数	8(4 ペア)
定格速度	4000 r/min
最高速度	4500 r/min
定格トルク	0.080 Nm
定格周波数	266.67Hz(電気角) , 66.67Hz(機械角)
定格電圧	36 V
定格電流	1.67Arms
定格出力	30 W

4.4 インバータの準備

インバータを準備される際に、以下の情報を確認してください。本サンプルプログラムでは、MCI-LV-1 に合わせた設定となっており、インバータを変更する場合には変更が必要です。

ベクトル制御の制御性能は、電流センサから入力される電流検出値と、ホールセンサから得られる磁極位置によって左右されます。つまり、センサ自体の性能や、センサから出力される信号の経路となる回路のばらつき・精度に、大きく影響されます。インバータの選定や設計には、十分に配慮を行ってください。

- 定格容量(VA)
- デッドタイム値(us)
- 電流センサの種類、特性、信号仕様
- 電流センサのゲイン値及びオフセット値、電流と電圧の関係性や信号の直線性の特性データ等
- 電圧センサのゲイン値及びオフセット値、信号の直線性の特性データ等

また、MCI-LV-1 上には、ユーザーがモータ制御指令を操作するためのボードユーザインタフェース(ボードUI)を有しています。部品・機能の一覧を表 4-2 に示します。

表 4-2 ボードユーザインタフェースの部品・機能の一覧

項目	インタフェース部品	機能
回転位置/速度	ボリューム(VR1)	回転速度指令値入力(アナログ値)
START/STOP	トグルスイッチ(SW1)	モータ回転開始/停止指令
ERROR RESET	プッシュスイッチ(SW2)	エラー状態からの復帰指令(エラー発生時)
電流オフセット調整	プッシュスイッチ(SW2)	電流オフセット調整(停止時)
LED1	オレンジ色 LED	・モータ回転時：点灯 ・停止時：消灯
LED2	オレンジ色 LED	・エラー検出時：点灯 ・通常動作時：消灯
RESET	プッシュスイッチ(RESET1)	システムリセット

4.5 ホールセンサ線の接続

本サンプルプログラムでは、ホールセンサを使用するため、MCI-LV-1 にホールセンサ信号線を接続する必要があります。MCI-LV-1 と、MOONS'社の R42BLD30L3 との対応表を以下に示します。

R42BLD30L3 のコネクタは、MOLEX 社の 43025-0600 を使用しています。勤合相手は 43020-0600, 43045-0600 が適合します。線径は AWG24 UL1061 となります。端子の準備が困難な場合は、ホールセンサ線からコネクタを切断し、被覆を剥いて MCI-LV-1 の端子台に挿入・固定してください。

表 4-3 ホールセンサとインバータセンサ入力端子の対応

信号線の仕様	MCI-LV-1	MOON'S R42BLD30L3
DGND	CN6-1	5-BLACK
+5V	CN6-2	1-RED
HALL_W	CN6-3	4-BLUE
HALL_V	CN6-4	3-YELLOW/WHITE
HALL_U	CN6-5	6-ORANGE/YELLOW

4.6 RX26T CPU ボードのセットアップ

RX26T CPU ボード(RTK0EMXE70C00000BJ)の装着方法について説明します。MCI-LV-1 の基板に、RX26T CPU ボードを差し込むことができます。また、サンプルプログラムを書き込むための端子、MC-COM 接続用端子、外部エンコーダ接続用 PG 端子が用意されています。

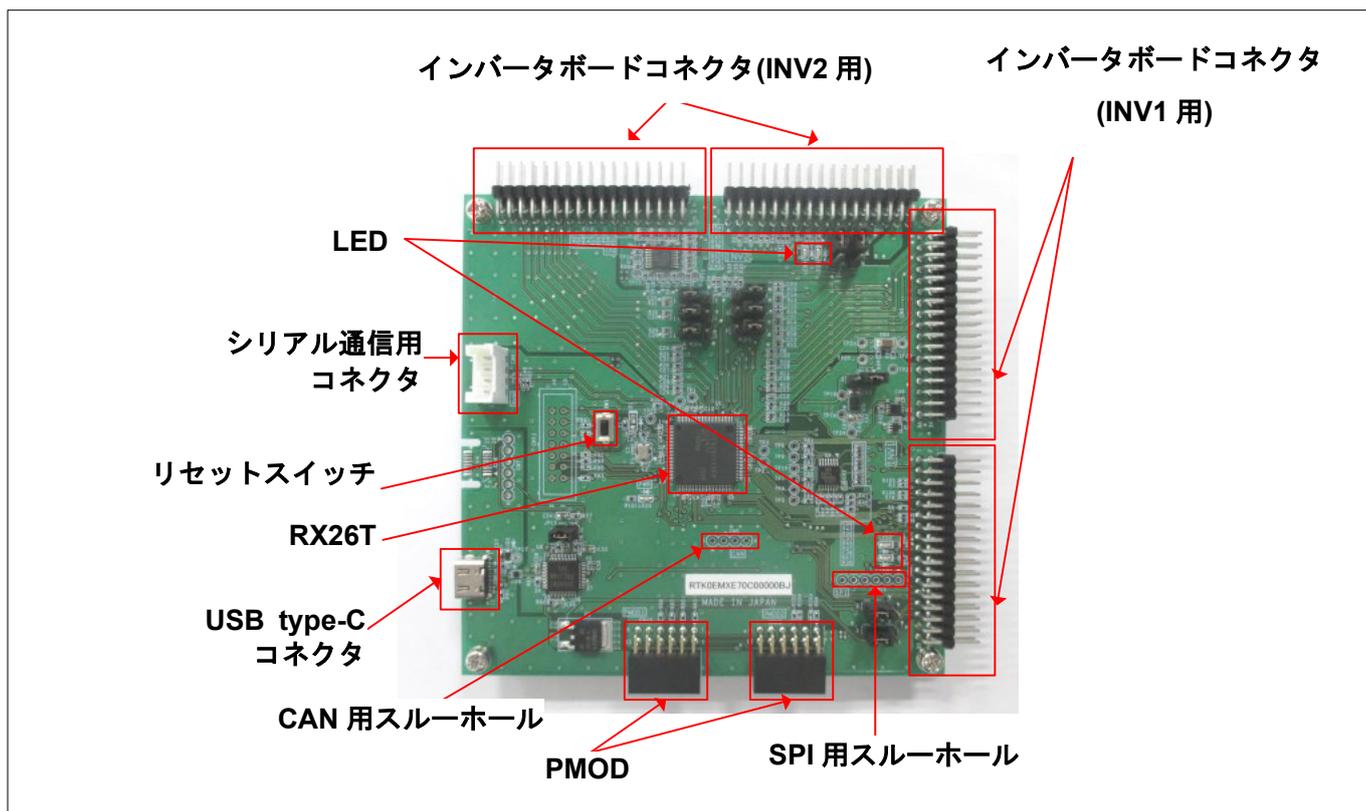


図 4-2 RX26T CPU ボードとインタフェース

4.7 キット(MCK-RX26T)の接続例

CPU ボードをインバータボード(MCI-LV-1、型名: RTK0EM0000B12020BJ)および通信ボード(MC-COM、型名: RTK0EMXC90Z00000BJ)と組み合わせて使用する際の接続例を図 4-3 に示します。

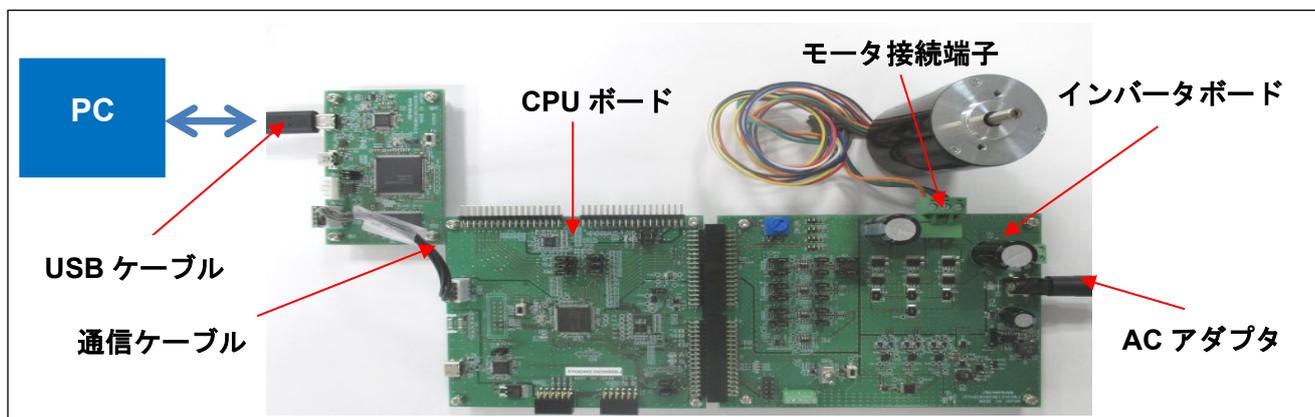


図 4-3 キット (MCK-RX26T) の接続例

4.8 オンボードデバッガ

図 4-4 に示す通り本製品にはオンボードデバッガ回路 E2 On-Board(以下、E2OB)が搭載されており、RX26T のプログラムの書き換えは E2OB を用いて行います。プログラムを書き換える場合、ジャンパ JP11 をオープンにし、CPU ボードと PC を USB ケーブルで接続してください。E2OB は E2 emulator Lite 相当のデバッガとして機能します。統合開発環境(例えば e2studio)あるいはフラッシュプログラミングツール(例えば Renesas flash programmer など)から接続する際には設定時にはデバッガ(ツール)の種類は「E2 emulator Lite」と設定してご利用ください。

プログラムを書き換えた後は、CPU ボードを動作させるために JP11 をショートしてください。

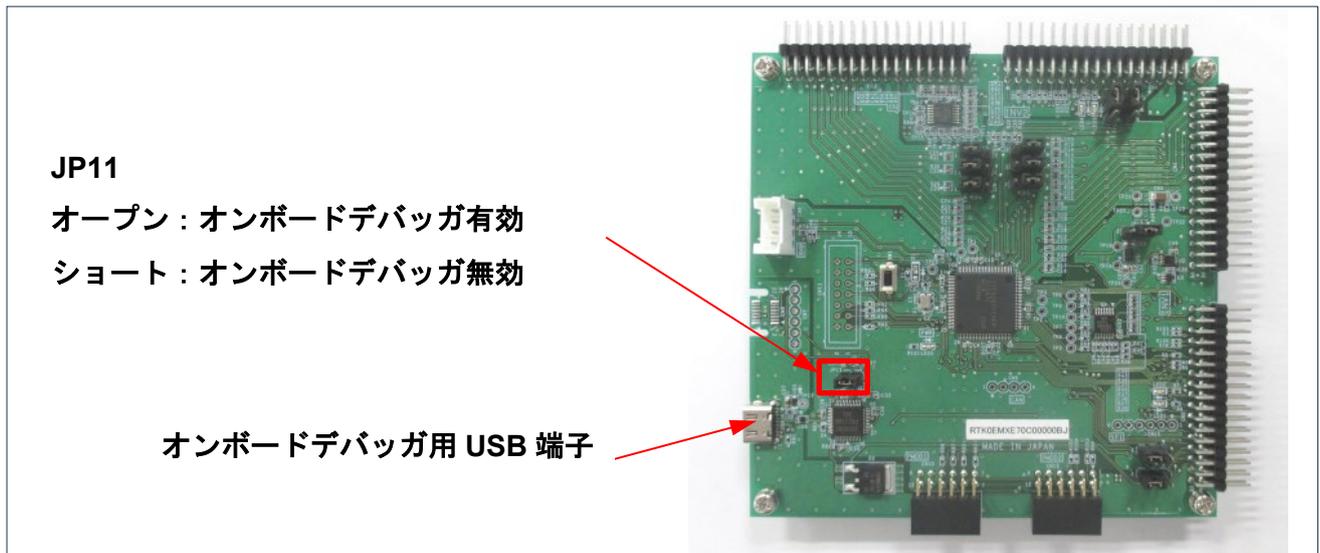


図 4-4 オンボードデバッグ

4.9 配線方法

電源、インバータ、モータの配線方法について説明します。ご使用する装置によって、端子の名称は異なりますので、必ず装置の取扱説明書を参照して内容・仕様を確認の上、配線作業を行ってください。

図 4-5 に、電源～インバータ間の配線例を示します。ここでは、直流安定化電源の出力端子をインバータの P 端子・GND 端子に接続します。極性の間違いにご注意ください。図 4-6 に、インバータ～モータ間の配線例を示します。3 相のモータ線以外に、位置検出用のホールセンサ線がインバータに接続されます。

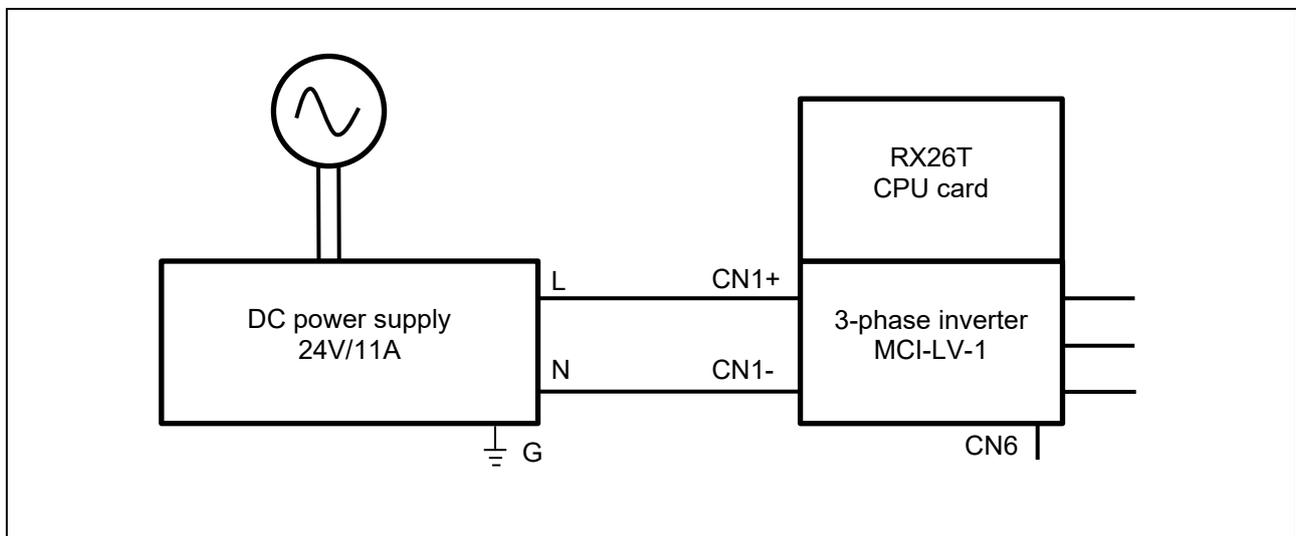


図 4-5 電源～インバータ間の配線

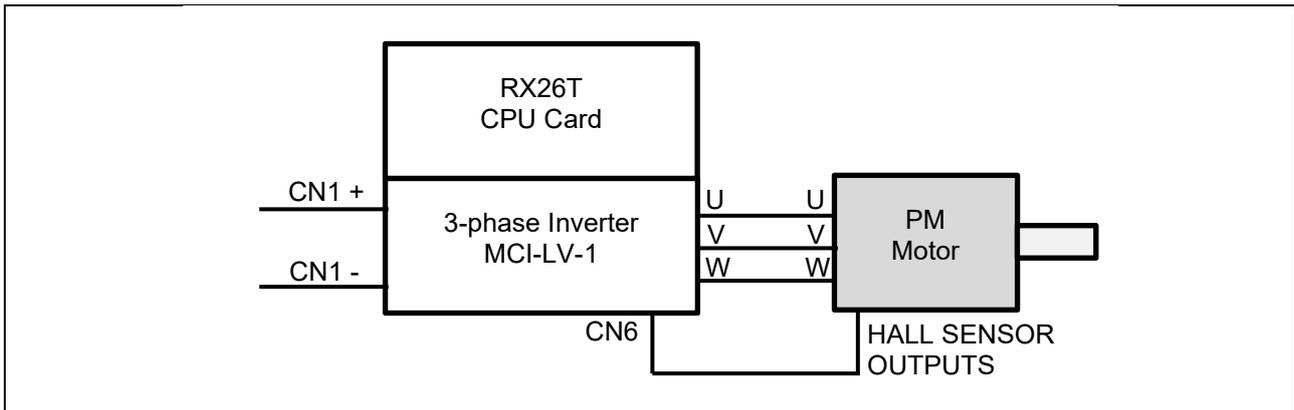


図 4-6 インバータ～モータ間の配線

4.10 測定器の利用

PM モータの制御性能の評価を行う際、パワーメータや、デジタルマルチメータ、トルクメータ、外付けエンコーダを用意することで、詳細なインバータ・モータ制御の分析が可能となります。ユーザ環境や、要求される測定精度、目標性能仕様に合わせて、必要な測定器を検討ください。

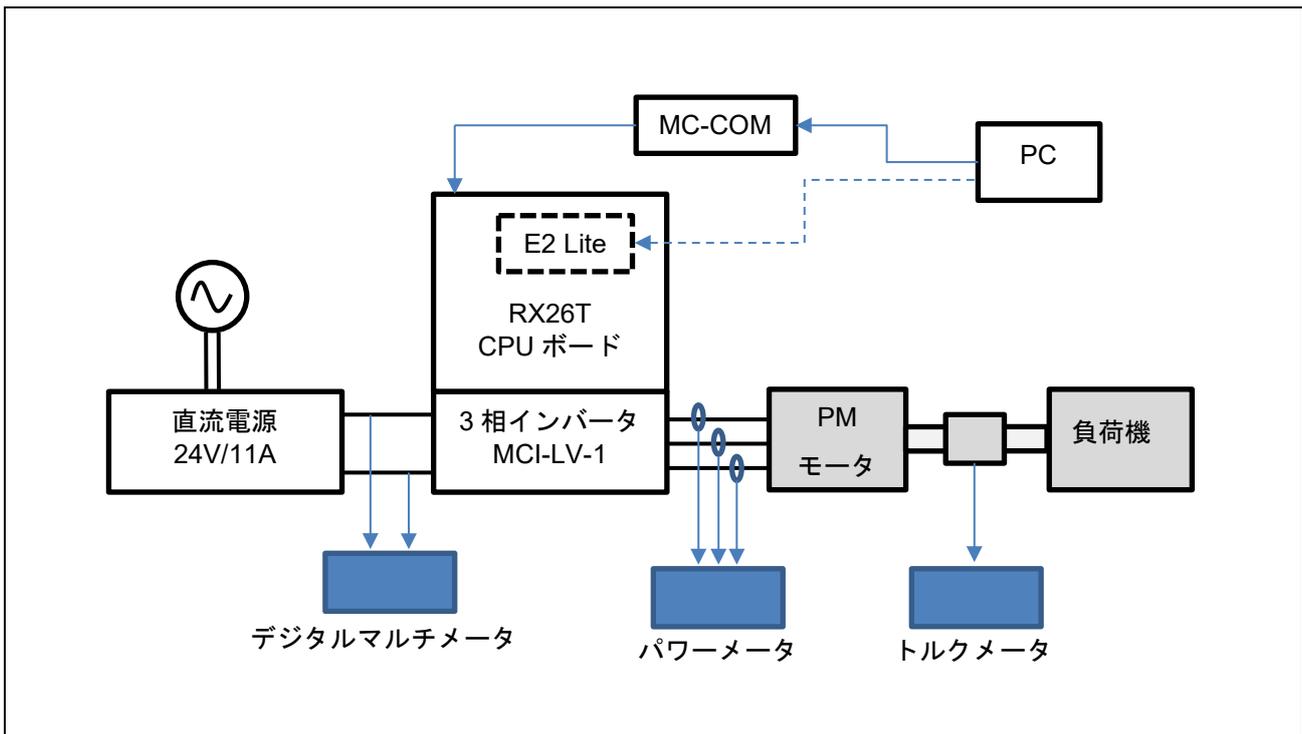


図 4-7 測定器の追加例

5. ソフトウェア環境構築方法

5.1 CS+を使用する場合

CS+は、以下の URL からダウンロードしてください。

<https://www.renesas.com/ja/software-tool/cs>

CC-RX ならびにスマート・コンフィグレータを別途、インストールする必要があります。インストール手順は、CS+付属の PDF マニュアルか、上記 URL 掲載のビデオ「CS+ Quick Start Guide (1/4) – インストール」を参照してください。

詳細な使用方法は、上記 URL でダウンロードが可能な PDF マニュアルや、ビデオを参照してください。

5.2 e² Studio を使用する場合

e² studio は、以下の URL からダウンロードしてください。

<https://www.renesas.com/ja/software-tool/e-studio>

CC-RX ならびにスマート・コンフィグレータを別途、インストールする必要があります。インストール手順は、e² studio 付属の PDF マニュアルか、上記 URL 掲載のビデオ「e² studio チュートリアル RX ファミリ (1/4) - インストール」を参照してください。

詳細な使用方法は、上記 URL でダウンロードが可能な PDF マニュアルや、ビデオを参照してください。

6. 運転方法

6.1 運転前の注意点

モータを動かすにあたって、以下の点にご注意ください。誤った使い方により、感電や機器の故障等を引き起こす場合があります。

- E2OB を使用しながら、トレース実行・ブレークポイントを設定した条件でモータ制御しないでください。不意の停止により、インバータが異常な動作をする場合があります。RMW を使用して、安全機能が正常に動作する条件下で、デバッグを行ってください。
- MC-COM は信号が絶縁されているため、運転中も安全に使用できます。類似品を使用する場合、PC とインバータの GND が共通となる場合があります、GND を介して感電事故の恐れがあります。
- 緊急停止が可能なように、実験設備を構築してください。
- ホールセンサ線の断線に注意してください。ホールセンサ線が断線・接触不良している場合、正常にモータ制御できません。
- モータの軸は回転しますので、必ずガードとなるカバーを取り付けしてください。特にカップリング等の部品は、回転中に損傷した場合、外に飛散する場合があります。
- インバータが停止しても、PM モータが回転している場合、PM モータは誘起電圧を発生させるため、UVW 三相配線に電圧がかかります。露出した導電部に接触すると、感電の恐れがあります。

6.2 運転までの手順

運転を行うための準備手順は、以下の通りです。

表 6-1 運転を行うまでの手順

手順	手順内容	関連する章
1	インバータにあらかじめ、CPU ボードを差し込んでおきます。	-
2	サンプルプログラム、開発環境(CS+または e2 studio など)、エミュレータ (E2OB 等)を使用する PC に導入する。	5
3	PC と、CPU ボードを、エミュレータを介して接続し、CPU ボードに 5V の電源を供給する	-
4	サンプルプログラムを、開発環境でビルドする	6.4
5	ビルドしたサンプルプログラムを CPU ボードに書き込む	
6	エミュレータと CPU ボードをつないだケーブルを取り外す	4.7
7	CPU ボードに MC-COM を接続する	4.7
8	インバータに DC24V の電源を供給する	4.2
9	PC にインストールされた RMW を用いて、CPU ボードに MC-COM を経由して接続し、正しく接続できることを確認する	6.5
10	本サンプルプログラムの変数や、センサ情報などが正常に RMW で表示されていることを確認する	6.7
11	RMW を用いて、モータの運転操作を行う	6.8
	ボード UI を用いて、モータの運転操作を行う	6.9
12	モータを停止・遮断する	6.10

6.3 接続方法

書き込み時と、運転操作時で、CPU ボードと PC の間で使用する機器が異なるため、ご注意ください。以下に、①書き込み時と、②運転操作時についての配線方法を説明します。

① 書き込み時

CPU ボードにオンボードデバッグ回路 E2 On-Board(以下、E2OB)が搭載されており、RX26T のプログラムの書き換えは E2OB を用いて行います。

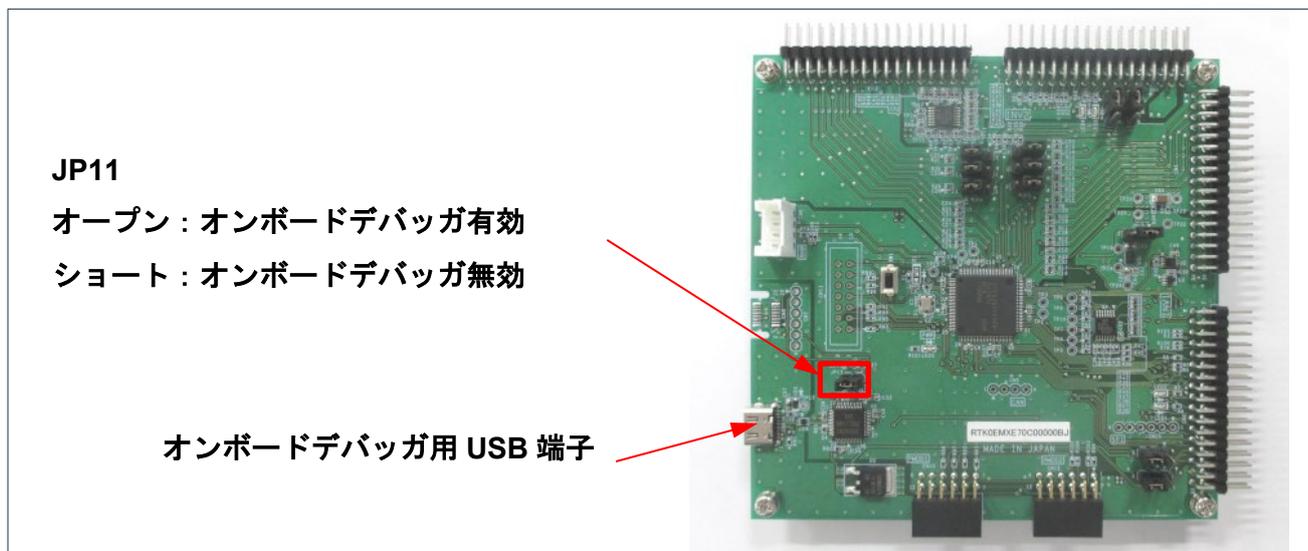


図 6-1 オンボードデバッグ

② 運転操作時

図 6-2 のように MC-COM(RTK0EMXC90Z0000BJ)を用いて、CPU ボードに接続します。PC とは、UART 経由で接続された状態となり、PC からは COM ポートを用いて操作することができます。本サンプルプログラムは、インバータ上の VR・スイッチを用いたボード UI または、RMW を用いて、運転操作を行うことができます。MC-COM はインバータと PC の間を電氣的に絶縁しますので、高電圧環境下でも、安全にご利用いただけます。

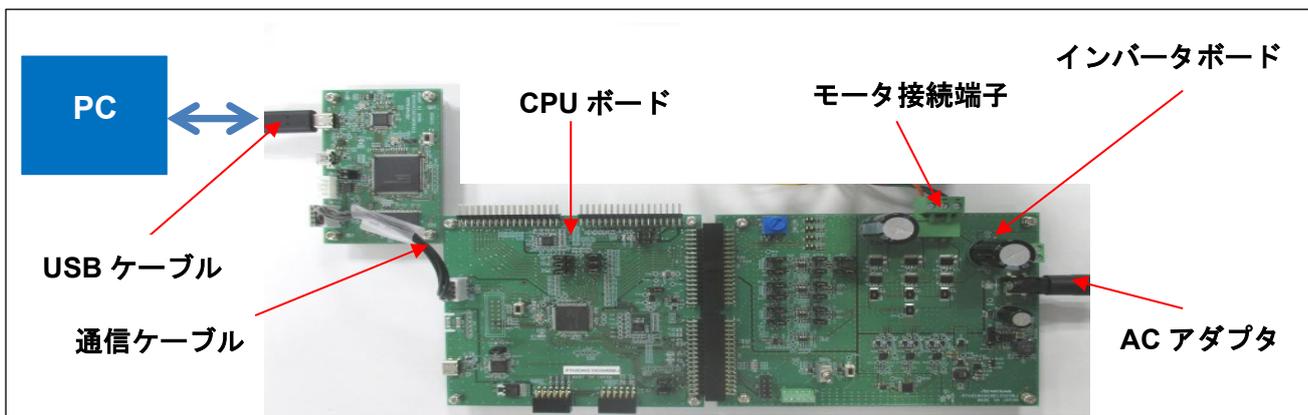


図 6-2 運転操作時の接続例

6.4 サンプルプログラムの書き込み

弊社 WEB サイトからダウンロードしたサンプルプログラムを、CS+や e² Studio、Renesas Flash Programmer を使用して CPU ボードの MCU に書き込んでください。

プログラムの書き込み方法は CS+や e² Studio 及び Renesas Flash Programmer の取扱説明書を参照してください。プログラムを書き込むためには、図 6-1 に示すようにオンボードデバッガを使用します。

6.5 RMW の導入方法

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。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。

<https://www.renesas.com/ja/software-tool/renesas-motor-workbench>

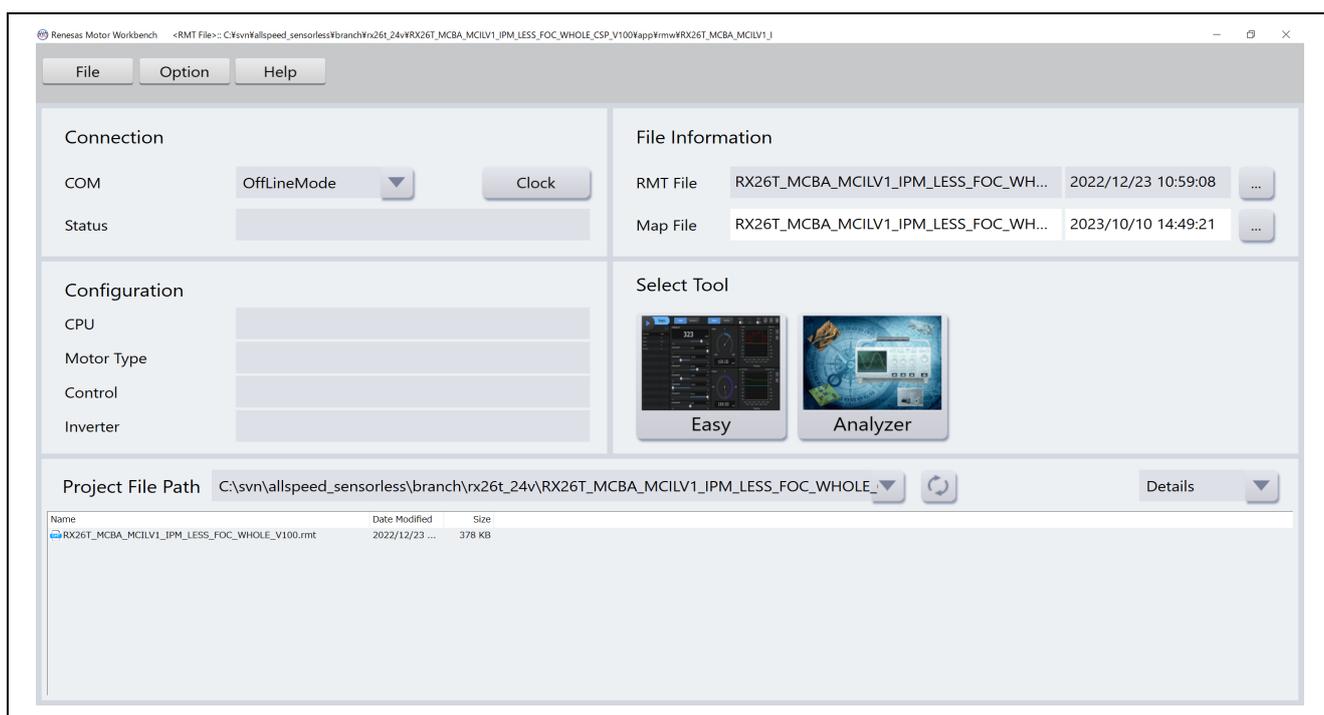


図 6-3 Renesas Motor Workbench 外観

6.6 Map ファイルの登録更新

ユーザがサンプルプログラムの一部を変更した場合、変数などの情報が記載された Map ファイルを RMW に登録更新する作業が必要になります。サンプルプログラムの変更を行っていない場合には、Map ファイルの登録更新作業は不要です。

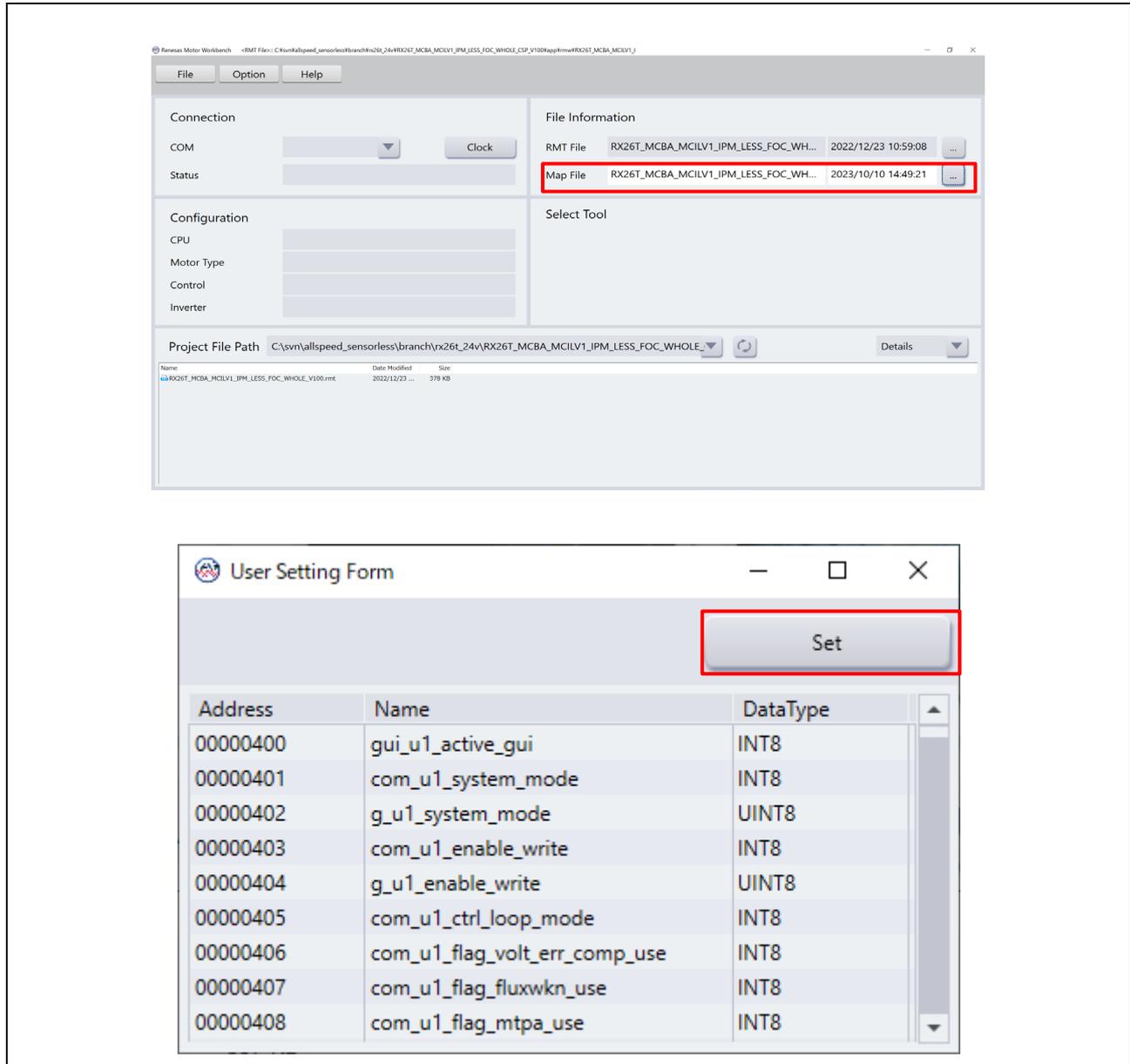


図 6-4 RMW の Map ファイル登録設定箇所（上）と反映画面（下）

6.7 RMW の操作に使用する変数

本サンプルプログラムで、モータを動かす場合には、RMW を用いて制御します。RMW UI 使用時の入力用変数一覧を表 6-2 に示します。なお、これらの変数への入力値は com_u1_enable_write に g_u1_enable_write と同じ値を書き込んだ場合にモータモジュール内の対応する変数へ反映され、モータ制御に使用されます。ただし、(*)が付けられた変数は com_u1_enable_write に依存しません。

一部のモータ制御に用いるパラメータは、停止中に設定を変更できます。詳細は表 8-8 を参照してください。

なお、変数名の接頭辞(u1,f4 等)は変数型の省略形となっています。RMW は、変数名の接頭辞を自動認識して型を自動で選択し、ControlWindow で変数内部の数値の表示を行います。

表 6-2 Analyzer 機能主要入力用変数一覧

Analyzer 機能入力用変数名	型	内容
com_u1_system_mode (*)	uint8_t	ステータス管理 0 : 停止モード 2 : 速度制御モード 3 : トルク制御モード 4 : オープンループ制御モード 5 : エラー解除 8 : 電流オフセット調整
com_f4_ref_speed_rpm (*)	float	速度指令値 (機械角) [r/min]
com_u1_enable_write	uint8_t	ユーザ入力用変数書き換え許可 g_u1_enable_write と変数一致で入力データ反映

次に速度制御の駆動評価を行う際に観測することの多い主要な構造体変数の一覧を表 6-3 に示します。Analyzer 機能で波形表示する際や変数の値を読み込む際に参考にしてください。

表 6-3 速度制御主要変数一覧

速度制御主要変数名	型	内容
g_st_hall_vector.u2_error_status	uint16_t	エラーステータス。 詳細は“止まってしまった場合”を参照
g_st_cc.f4_vdc_ad	float	インバータ母線電圧値[V]
g_st_cc.f4_id_ref	float	d 軸電流指令値 [A]
g_st_cc.f4_id_ad	float	d 軸電流検出値 [A]
g_st_cc.f4_iq_ref	float	q 軸電流指令値 [A]
g_st_cc.f4_iq_ad	float	q 軸電流検出値 [A]
g_st_cc.f4_iu_ad	float	U 相電流検出値 [A]
g_st_cc.f4_iv_ad	float	V 相電流検出値 [A]
g_st_cc.f4_iw_ad	float	W 相電流検出値 [A]
g_st_cc.f4_vd_ref	float	d 軸電圧指令値 [V]
g_st_cc.f4_vq_ref	float	q 軸電圧指令値 [V]
g_st_cc.f4_refu	float	U 相電圧指令値 [V]
g_st_cc.f4_refv	float	V 相電圧指令値 [V]
g_st_cc.f4_refw	float	W 相電圧指令値 [V]
g_st_cc.st_rotor_angle.f4_rotor_angle_rad	float	推定磁極位置[rad]
g_st_sc.f4_ref_speed_rad_ctrl	float	速度指令値 (機械角) [rad/s]
g_st_sc.f4_speed_rad	float	速度検出値 (機械角) [rad/s]

6.8 モータ操作方法(RMW)

RMW の 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、RMW 画面上の“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。

(a) ユーザインタフェースを Analyzer 使用に変更する

- ① “com_u1_sw_userif”の[W?]欄に“チェック”が入っていることを確認します。
- ② [Write]欄に 0 を入力します。
- ③ “Write”ボタンをクリックします。

(b) 電流オフセット調整（手動）

電流オフセット調整は電流投入の 0.5 秒後自動的に開始されます。実施中は LED1 が点灯します。なお、以下の操作を行って手動で再調整することもできます。通常は手動で再調整する必要はありません。

- ① “com_u1_mode_system”の[W?]欄に“チェック”が入っていることを確認します。
- ② [Write]欄に 8 を入力します。
- ③ “Write”ボタンをクリックします。
- ④ “Read”ボタンを押して現在の“com_u1_mode_system”の [Read] 欄を確認します。
- ⑤ “com_u1_mode_system”が 0 に戻ることを確認します。

(c) モータを回転させる

- ① “com_u1_mode_system”, “com_f4_ref_speed_rpm”, “com_u1_enable_write”の [W?] 欄にチェックが入っていることを確認します。
- ② 指令回転速度を“com_f4_ref_speed_rpm”の [Write] 欄に入力します。
- ③ “Write”ボタンをクリックします。
- ④ “Read”ボタンを押して現在の“com_f4_ref_speed_rpm”, “g_u1_enable_write”の [Read] 欄を確認。
- ⑤ MCU 内の変数値へ反映させるため、“com_u1_enable_write” に④で確認した“g_u1_enable_write”と同じ値(“0” or “1”)を入力します。
- ⑥ “com_u1_mode_system”の [Write]欄に“2”を入力します。
- ⑦ “Write”ボタンをクリックします。

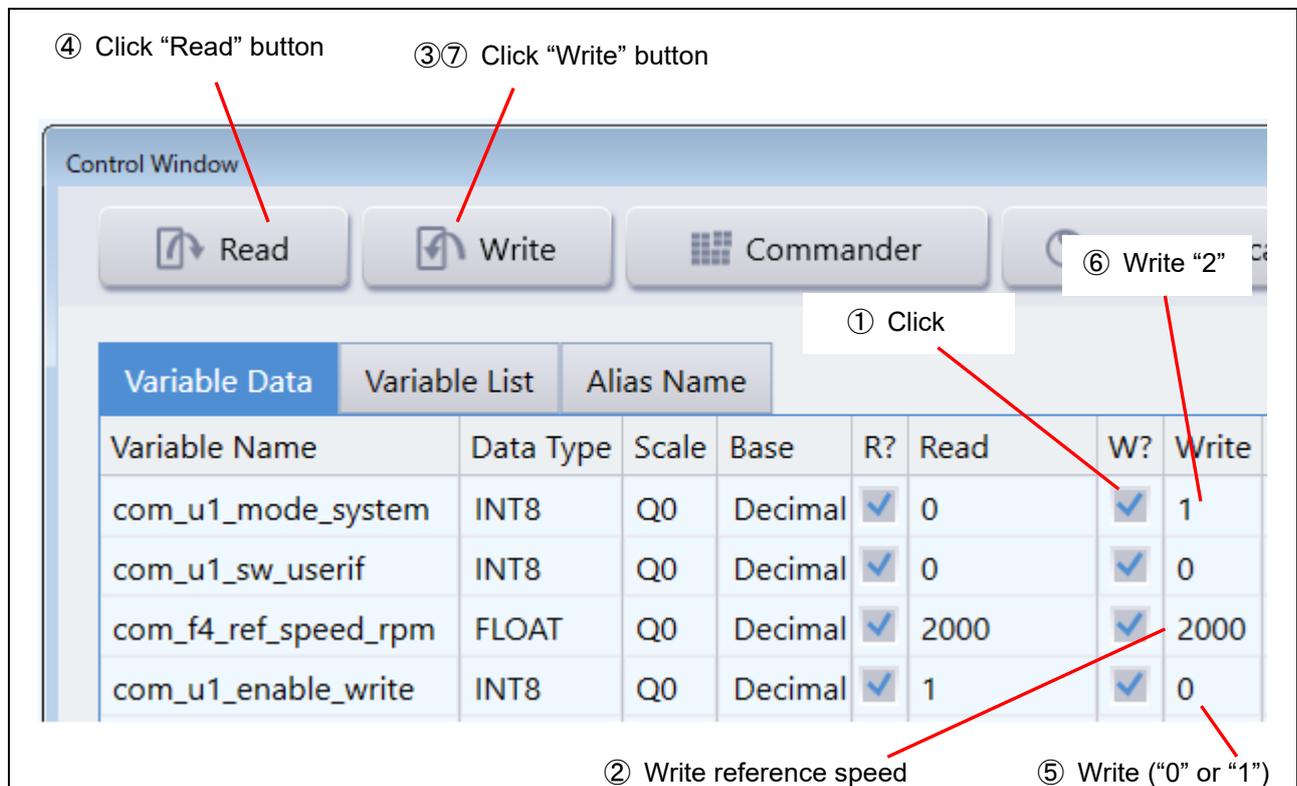


図 6-5 モータ回転の手順

(d) モータを停止させる

- ① "com_u1_mode_system" の [Write] 欄に "0" を入力します。
- ② "Write" ボタンをクリックします。

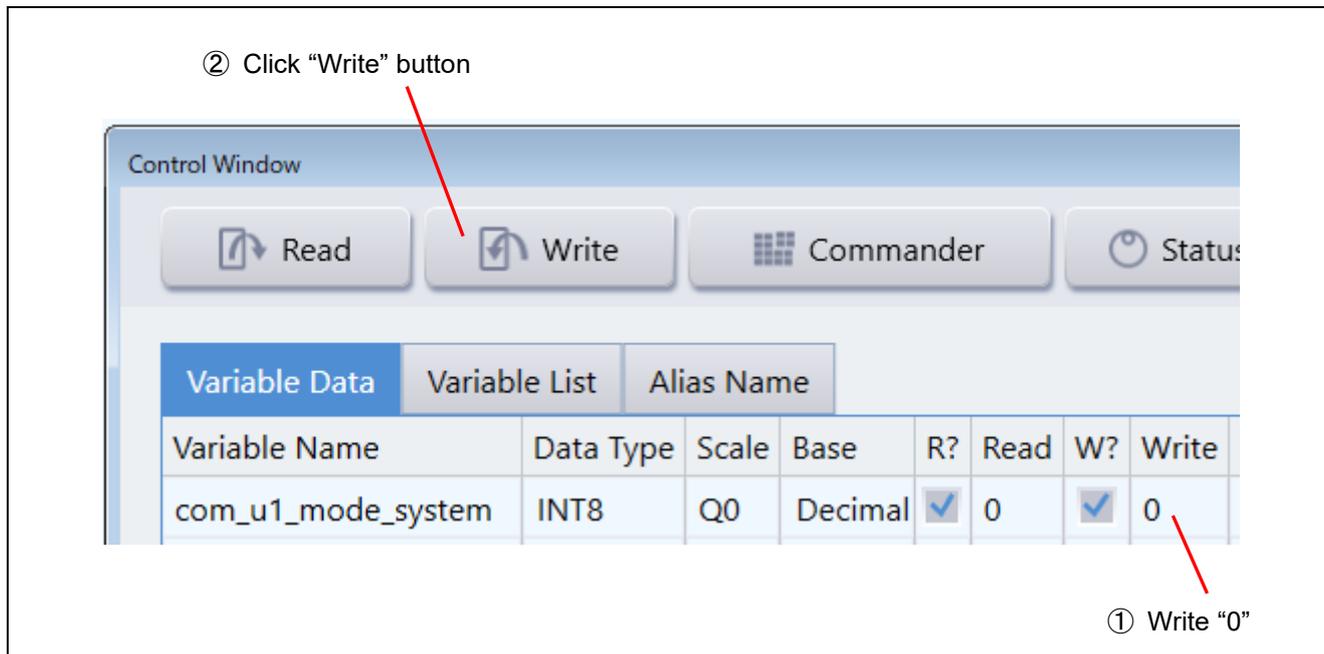


図 6-6 モータ停止の手順

(e) 止まってしまった（エラー）場合の処理

- ① "com_u1_mode_system" の [Write] 欄に "5" を入力する。
- ② "Write" ボタンを押す。

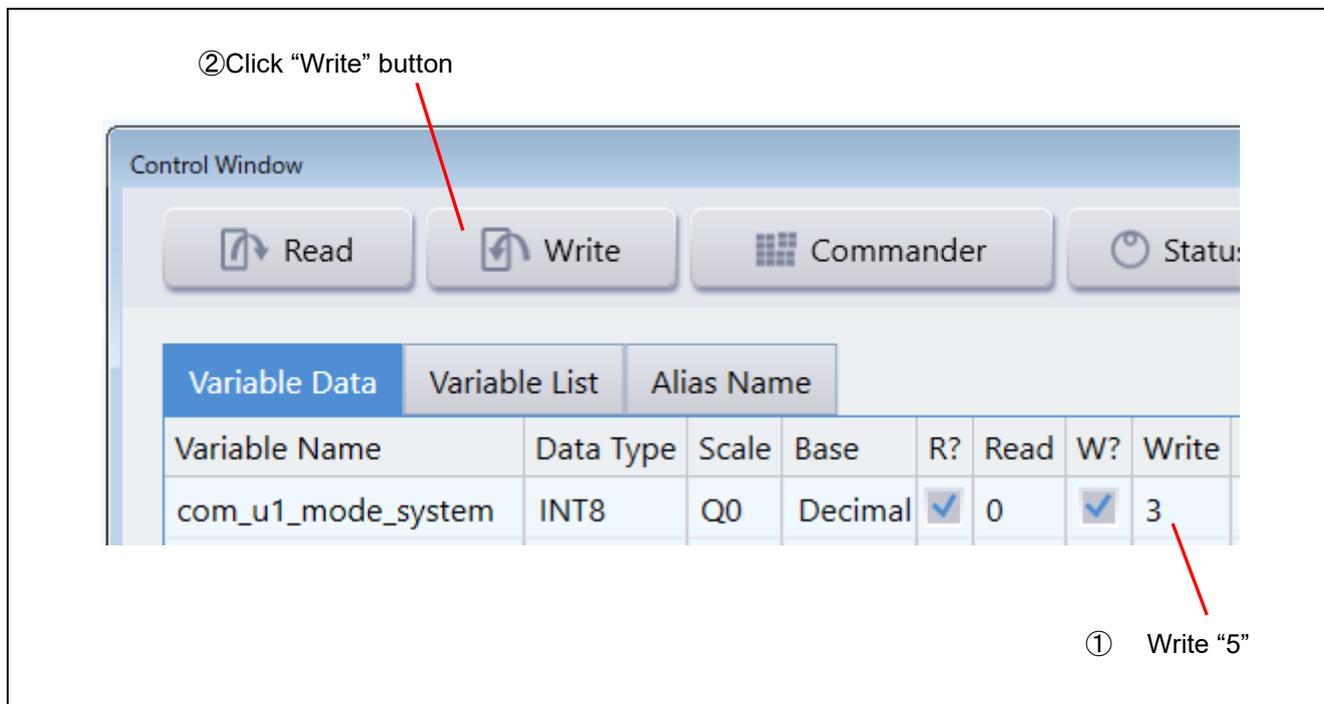


図 6-7 エラー解除の手順

表 6-4 エラーステータスの説明

値	エラー内容	割り当てられているマクロ名
0x0000	エラーなし	MOTOR_ERROR_NONE
0x0001	HW 過電流エラー	MOTOR_ERROR_OVER_CURRENT_HW
0x0002	過電圧エラー	MOTOR_ERROR_OVER_VOLTAGE
0x0004	過速度エラー	MOTOR_ERROR_OVER_SPEED
0x0080	低電圧エラー	MOTOR_ERROR_LOW_VOLTAGE
0x0100	SW 過電流エラー	MOTOR_ERROR_OVER_CURRENT_SW
0x0200	過熱エラー	MOTOR_ERROR_OVER_TEMPERATURE
0xffff	未定義エラー	MOTOR_ERROR_UNKNOWN

6.9 モータ操作方法(ボード UI)

(a) ユーザインタフェースの切り替え

本サンプルプログラムは、RMW UI をユーザインタフェースをデフォルトとして設定しています。ボード UI で使用する場合には、以下の手順に従って切り替え操作を行ってください。この切り替えは、電源入り切りやりセット操作によりデフォルトに戻ります。

“com_u1_sw_userif” の[W?]欄に“チェック”が入っていることを確認し、[Write]欄に“1”を入力する。“Write” ボタンを押します。

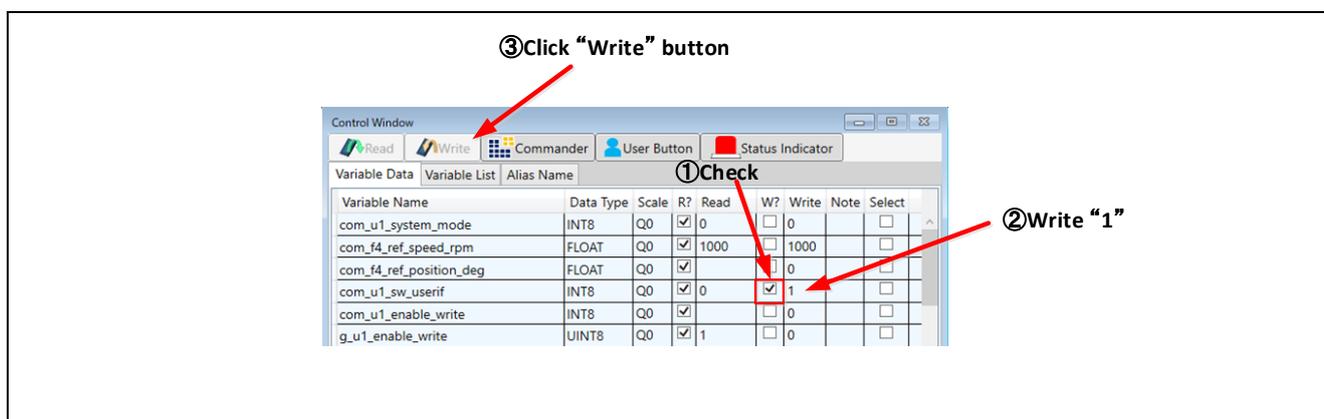


図 6-8 RMW・ボード UI の切り替えの手順

(b) モータ起動／停止

電流オフセット調整は電流投入の 0.5 秒後自動的に実施されます。完了(LED1 が消灯する)までモータを始動しないでください。

ボード UI 時はモータの起動と停止をインバータボードの SW1 からの入力(ボード UI)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”レベルのときはモータを停止すると判断します。

(c) 電流オフセット調整 (手動)

ボード UI 時はモータ停止状態 (LED1 消灯時) で SW2 を押すと、電流オフセット調整が開始します。調整開始時 LED1 が点灯し、電流オフセット調整が完了すると LED1 が消灯します。

(d) モータ回転速度指令値

モータの回転速度指令値はインバータボードの 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。

なお、ボード UI ではトルク制御モードには対応していません。

表 6-5 回転速度指令値の変換比

項目	変換比 (指令値 : A/D 変換値)	
	回転速度指令値	CW
	CCW	0 [rpm]~-2400 [rpm] : 0800H~0FFFH

6.10 モータ停止・遮断方法

運転状態からモータを停止する場合には、以下に示す手順で行ってください。なお、緊急時は、②の DC24V の供給を最優先にして停止させてください。

- ① 6.8(b)6.8(d)のモータ停止手順を行う。
- ② モータが停止するのを確認したら、直流安定化電源を操作し、DC24V の供給を停止する。

7. モータ制御アルゴリズム

7.1 概要

本サンプルプログラムのモータ制御アルゴリズムについて説明します。表 7-1 に、モータ制御機能を示します。

表 7-1 本サンプルプログラムのモータ制御機能

機能項目	機能の内容
制御方式	センサ付ベクトル制御
PWM 変調方法	空間ベクトル変調法（三角波比較法も選択可）
位置・速度検出方法	ホールセンサ用・微分積分法
制御モード	速度制御・トルク制御
制御・補償機能	<ul style="list-style-type: none"> ・ 電圧誤差補償 ・ 電圧位相進み補償 ・ 非干渉制御 ・ 弱め磁束制御 ・ 摩擦補償

7.2 制御ブロック図

ホールセンサベクトル制御のシステム全体のブロック図例を以下に示します。

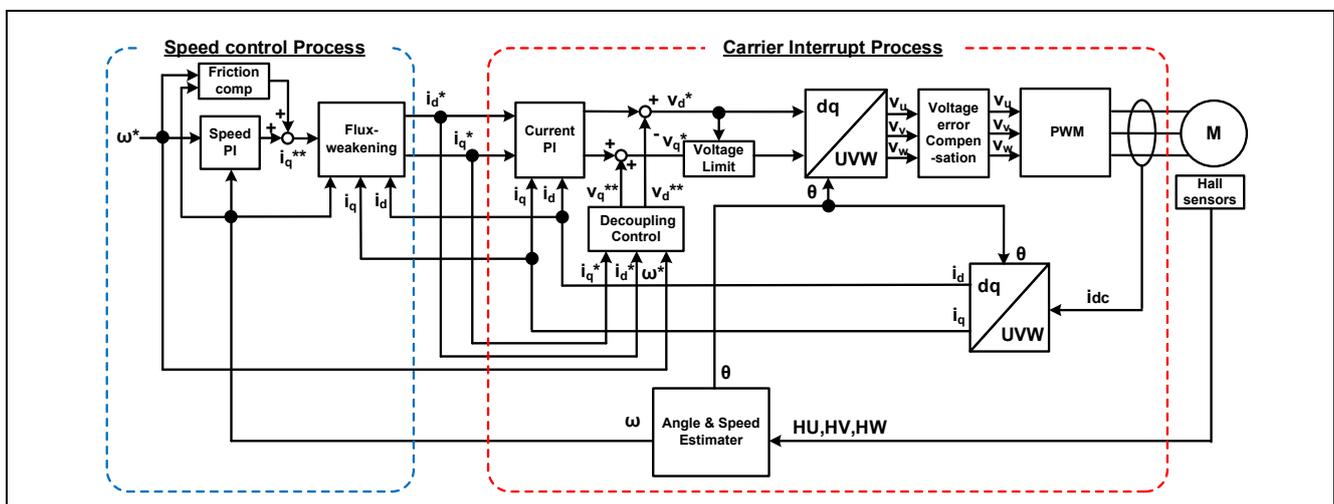


図 7-1 ホールセンサベクトル制御概略ブロック図

7.3 速度制御機能

速度制御機能は、モータが速度指令に追従するよう、PI 制御を行います。速度指令値の入力を受けて、内部の速度調節器が速度推定値との偏差を基に電流指令値を出力します。また、サブモジュールの弱め磁束制御を本モジュールから制御します。

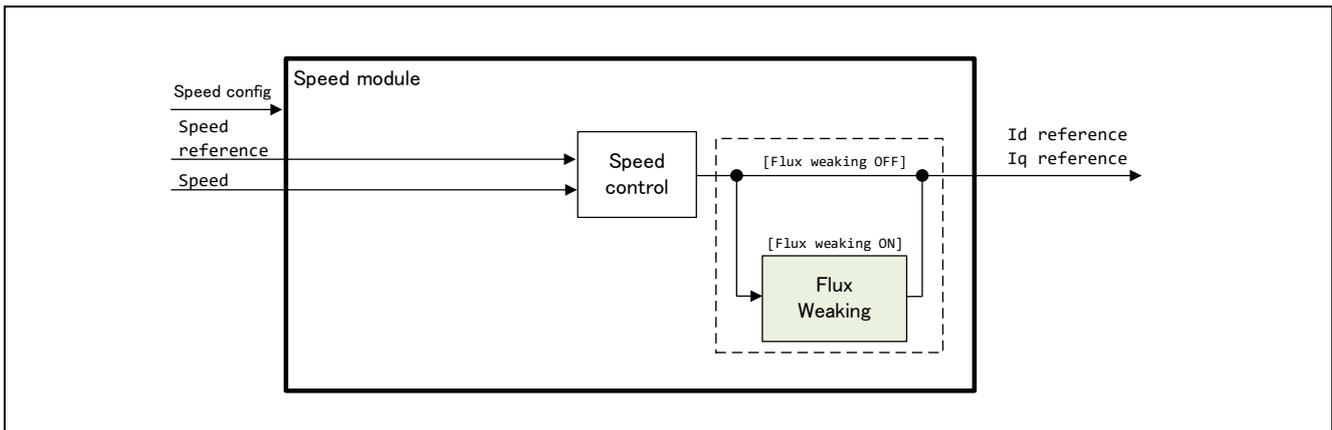


図 7-2 速度制御の機能ブロック図

7.4 弱め磁束制御

弱め磁束制御は、d 軸電流を負方向に制御することで、母線電圧で出力可能な電圧を PM モータの回転に比例して発生する誘起電圧($=\omega\Psi$)が超えてしまう条件下であっても、d 軸電流指令を負の値で増大させ、打ち消す制御を行います(図 7-3)。電圧飽和分を打ち消すことにより、加速に必要な q 軸電流指令を増やすことが可能となり、高速回転化および高速回転域での出力向上を実現します。

弱め磁束制御は、現在の電圧に対して、モータの回転速度が高くなり、電圧の余裕が難しくなった状態を自動的に検知して、 I_d^* の値を負に増大させ、PM モータの電圧方程式に従って誘起電圧をキャンセルする制御を行います。

このため、まず、誘起電圧制限値を以下の計算式から求めます。R はモータの抵抗値、 I_a は I_d, I_q 検出値の二乗和($\sqrt{I_d^2 + I_q^2}$)です。 V_{amax} は、電圧誤差補償や変調処理であらかじめ計算している電圧ベクトルの最大値を使用します。

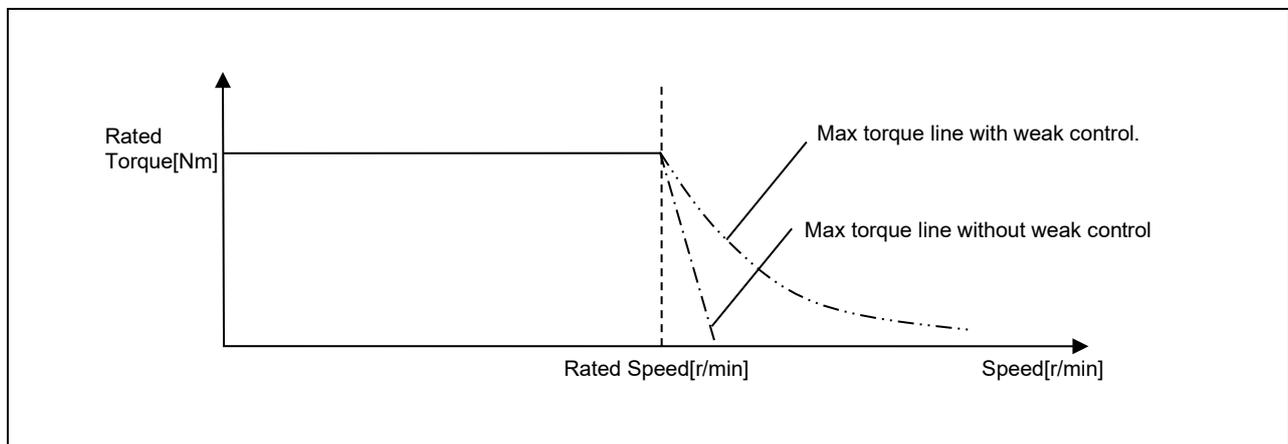


図 7-3 出力可能トルクと速度の関係例

$$V_{om} = V_{amax} - I_a R$$

V_{om} : 誘起電圧制限値 [V], V_{amax} : 電圧ベクトルの最大値 [V]

I_a : 電流ベクトルの大きさ [A]

図 7-4 誘起電圧制限値の計算式

$$I_d = \frac{-\psi_a + \sqrt{\left(\frac{V_{om}}{\omega}\right)^2 - (L_q I_q)^2}}{L_d}$$

$$\because V_{om} = V_{amax} - I_a R$$

V_{om} : 誘起電圧制限値 [V], V_{amax} : 電圧ベクトルの最大値 [V], I_a : 電流ベクトルの大きさ [A]

図 7-5 弱め磁束制御における d 軸電流指令値の計算式

7.5 電流制御機能

電流制御機能は、入力された電流値からベクトル制御に必要な座標変換及びフィードバック制御を行い、PWM として出力する電圧を演算する機能です。また、サブモジュールの非干渉制御、電圧位相進み補償、電圧誤差補償、速度検出・角度検出を本モジュールから制御します。モジュール構成図を図 7-6 に示します。

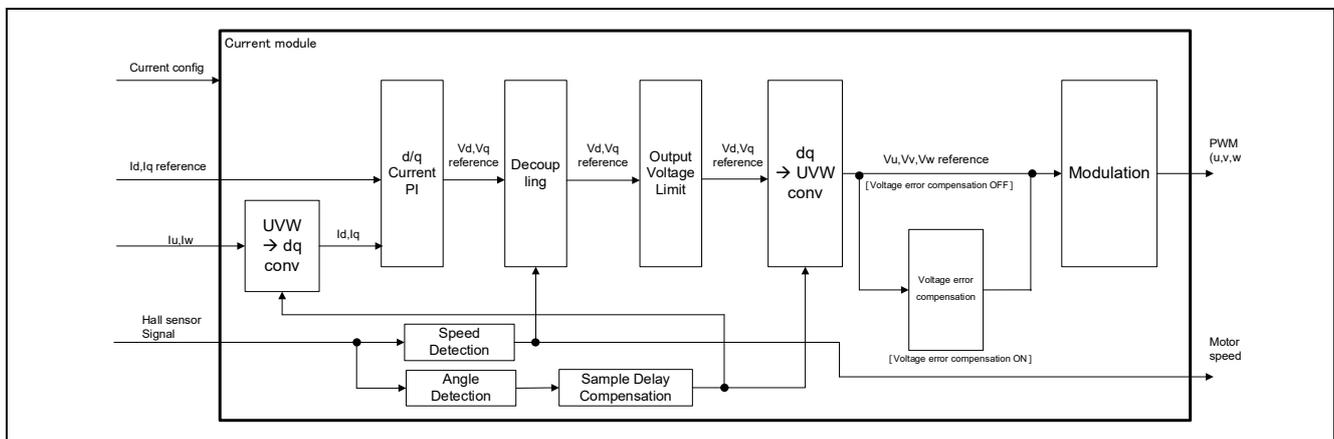


図 7-6 電流制御の機能ブロック図

7.6 非干渉制御

非干渉制御は、電流応答性の向上や d 軸・q 軸の間で電流が互いに干渉し合い、安定性を損なう事を抑制するために使用します。使用する式は、以下となります。一般的な PM モータの電圧方程式となります。

$$V_{d_dec}^* = R I_d^* - \omega L_q I_q^*$$

$$V_{q_dec}^* = R I_q^* + \omega L_d I_d^* + \omega \Psi$$

I_d^*, I_q^* : 電流指令値[A], ω : 回転速度(電気角)[rad/s], R: モータの 1 次抵抗[Ω],

L_d, L_q : モータのインダクタンス[H], Ψ : モータの鎖交磁束数[Wb]

得られた電圧指令値 $V_{d_dec}^*$ と $V_{q_dec}^*$ は、PI 調節器から出力される電圧指令値 V_d^* と V_q^* に加算します。

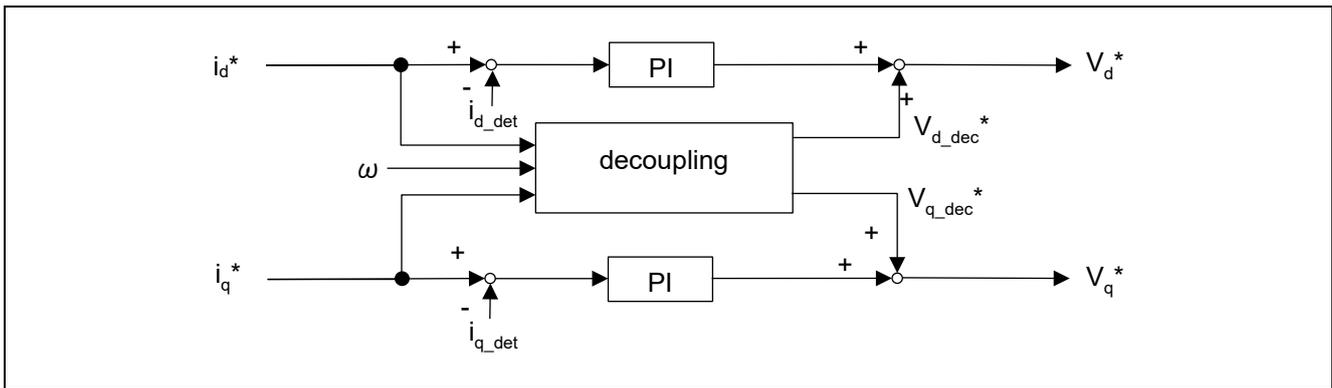


図 7-7 非干渉制御の機能ブロック図

7.7 ホールセンサベクトル制御機能

7.7.1 概要

ホールセンサの信号から、電気角一周あたり 6 パルスを検出することができます。通常、1 周 6 パルスは低分解能であるため、ベクトル制御をおこなう場合には任意のトルクを出力するために角度の補完処理が必要となります。ホールセンサのような低分解能なエンコーダをベクトル制御に用いる場合において、補完処理方法としては、複数の手法が提案されています。本サンプルプログラムでは、平易で実装が容易な微分積分法を用います。回転状態において、パルスの角度の変化量を微分することで、速度を検出します。また、検出した速度で回転していると仮定した場合、サンプリング周期ごとに速度を積分することで、角度を得ることができます。

本検出方法は、ハードウェアを用いずにソフトウェアでパルス検知する方法を採用しており、サンプリング周期分の検出誤差を速度に含みます。よって、過去複数回の検出結果をもとに移動平均することで、補正を行う方法を選択できるようにパラメータを設けています。

ホールセンサベクトル制御の特有の始動特性として、始動時の速度急変現象があります。停止状態から、最初のホールセンサの出力するパルスが入力されるまで、速度信号が一定時間得られないことから、速度調節器の積分器が飽和し、過剰なトルク電流指令による速度変動が発生する場合があります。

速度変動を低減するために、後述する静止摩擦補償を用いて、モータや負荷に応じて調整を行ってください。なお、補償を行った場合でも、完全に速度変動を除去することは原理上、困難となりますので、予めご了承ください。

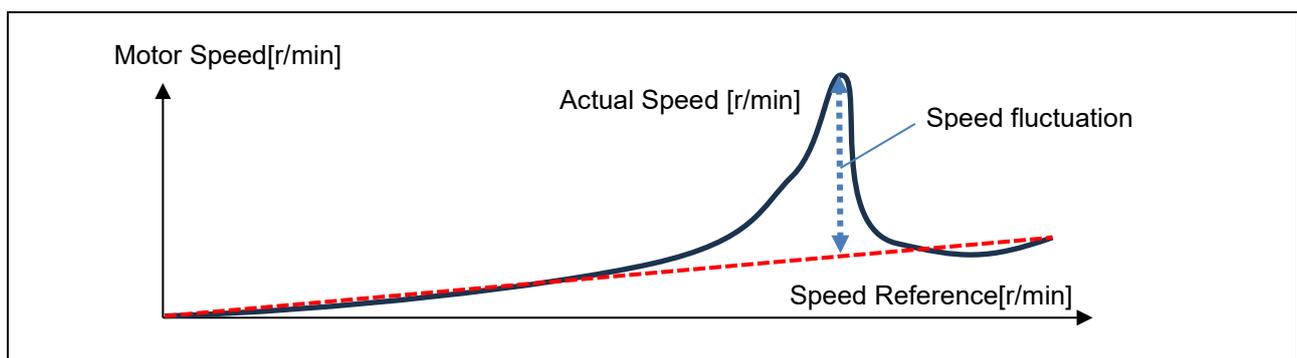


図 7-8 始動時の速度変動

7.7.2 ホールセンサ信号とホールパターン番号

ホールセンサは 3 つで構成され、図 7-9 に示す通り、UVW の巻線に対して対応するように 120 度ずつずれて配置されています。ホールセンサはそれぞれ、HU, HV, HW と定義します。

ホールセンサ信号は、HU,HV,HW の3つの信号で構成されており、この信号の High/Low の違いで、1回転 360 度を6つの磁極位置の状態で表すことができます。つまり、電気角1周に対して6パルスのエンコーダとみなすことができます。

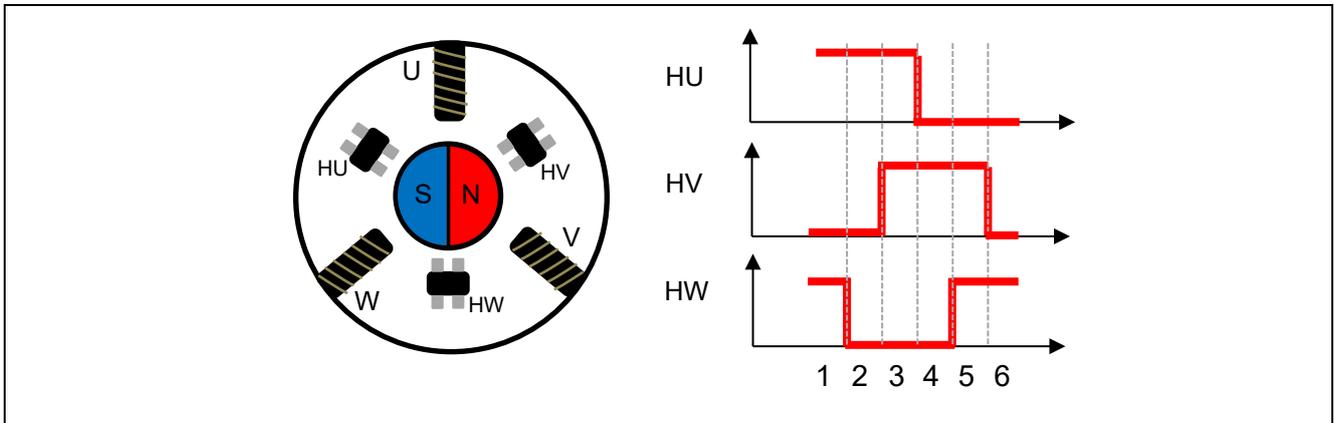


図 7-9 ホールセンサ配置とホールセンサ信号

HU,HV,HW の1ビットの信号から、一意のホール信号値を式で表すと以下で示されます。

$$\text{Hall signal value} = \text{HW} + \text{HV} \times 2 + \text{HU} \times 4$$

HU,HV,HW の信号とホール信号値の関係を表で表すと、以下のような関係があります。

表 7-2 ホールセンサ信号値の関係

HU	HV	HW	ホール信号値
0	0	1	1
1	0	1	5
1	0	0	4
1	1	0	6
0	1	0	2
0	1	1	3

上記のホール信号値と 360 度を 1-6 で表現したホールパターン番号を表で表すと、以下のような関係があります。

表 7-3 ホールセンサ信号とパターン番号の関係

ホールパターン番号	ホール信号値
1	1
2	5

3	4
4	6
5	2
6	3

以上から、ホールパターン番号をもとにして、速度検出および角度検出処理を行います。

7.7.3 速度検出方法

回転速度の検出方法は、ホールセンサ信号の変化を GPIO を介して電流制御周期で確認し、ホール信号の変化の間に何回の周期が発生したかをカウントすることで電気角 $2\pi/6$ [rad](ホール信号変化 1 区間)の期間 T_{60deg} 計測を行います。電流制御周期(50 μ sec)を T_s とすると、以下の式で表すことができます。

$$T_{60deg} = \text{電流制御周期の回数}k \times T_s$$

これにより、以下の式から電気角速度 ω_e [rad/s]を得ることができます。

$$\omega_e = \frac{2\pi}{6} \times \frac{1}{T_{60deg}}$$

ただし、ホール信号変化 1 区間で推定を行う場合、電流制御周期(50 μ s)のジッタに伴う誤差の影響が含まれます。また、3つのホールセンサの取り付け誤差が、検出速度の外乱として現れることがあります。そこで、移動平均を行って演算を行い、ジッタの影響を軽減することもできます。本サンプルプログラムでは、過去 1 回分と、過去 6 回分(1 周分)、自動切り替えの 3つの検出方法を選択できます。

たとえば、6 回の移動平均を用いた場合には、以下の式で 6 回分の平均となる電気角速度 ω_{e6} 求めることができます。

$$\omega_{e6} = \frac{2\pi}{6} \times \frac{6}{T_{360deg}} = \frac{2\pi}{T_{360deg}}$$

ソフトウェアへの実装では角度演算を簡略化するため、以下のような式に置き換えて等価な計算を行っています。

$$\text{キャリア 1 周期電気角変化量[rad]} = 2\pi[\text{rad}] \div \text{キャリア割り込み回数}(2\pi \text{ 期間})$$

$$\text{電気角速度[rad/sec]} = \text{キャリア 1 周期電気角変化量[rad]} \times \text{キャリア周波数[Hz]}$$

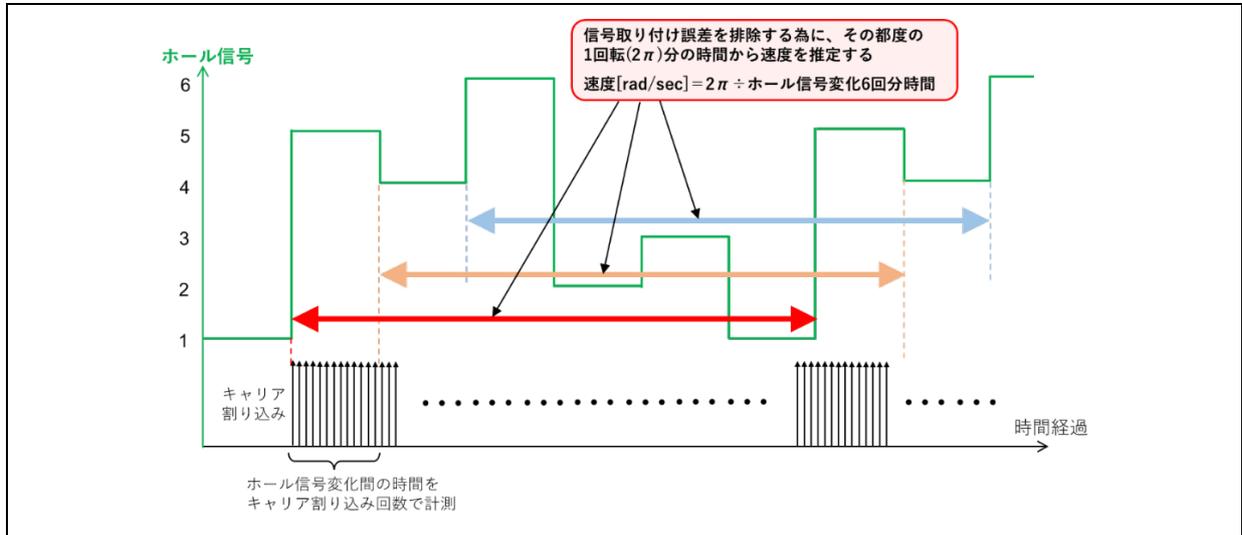


図 7-10 速度推定概念図

7.7.4 角度検出方法

角度検出では、回転方向と推定速度の2つの情報から、電気角を推定します。

回転方向はホールセンサ信号パターンにより検出します。ホールセンサ信号のパターンは使用するモータにより決まっているので、信号が変化した際にその直前のホールセンサ信号と今回のホールセンサ信号を比較することで回転方向を検出することが可能となります。

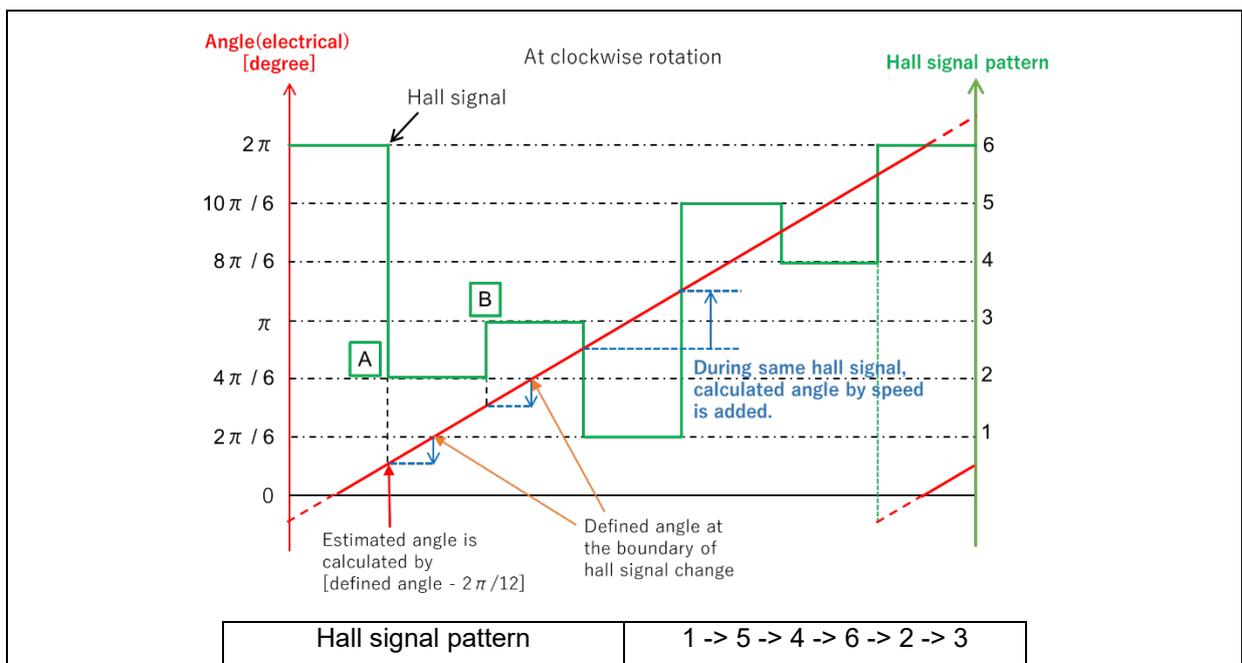


図 7-11 電気角度推定（正方向回転時）

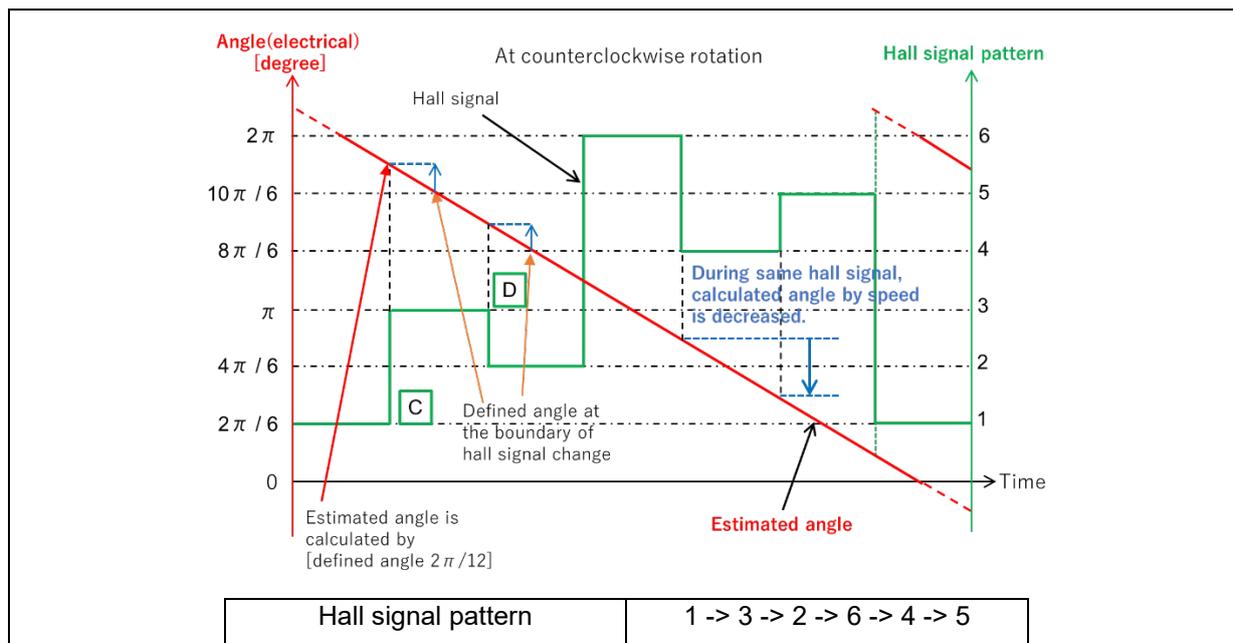


図 7-12 電気角度推定（逆方向回転時）

図 7-11 A 点ではホール信号が 1→5 と変化しています。これからモータの回転方向を正方向と判断できます。ここで電気角を以下のように設定します。

A 点(正方向回転時)

$$\text{電気角[rad]} = 2\pi[\text{rad}] \times \text{基準角度調整値}(1/6) + \text{ホール信号内角度[rad]} + \text{オフセット値[rad]}$$

各ホール信号の変化点において $2\pi/6[\text{rad}]$ の基準角度をもって電気角を推定します。用いる基準角度調整値はホール信号に応じて表 7-4 のように設定します。

表 7-4 基準角度調整値一覧

ホール信号	1	5	4	6	2	3
基準角度調整値	0(0/6)	1/6	2/6	3/6	4/6	5/6

図 7-11 A 点ではホール信号が 1→5 と変化しているため、基準角度調整値はホール信号 5 に対応する $1/6$ を設定します。

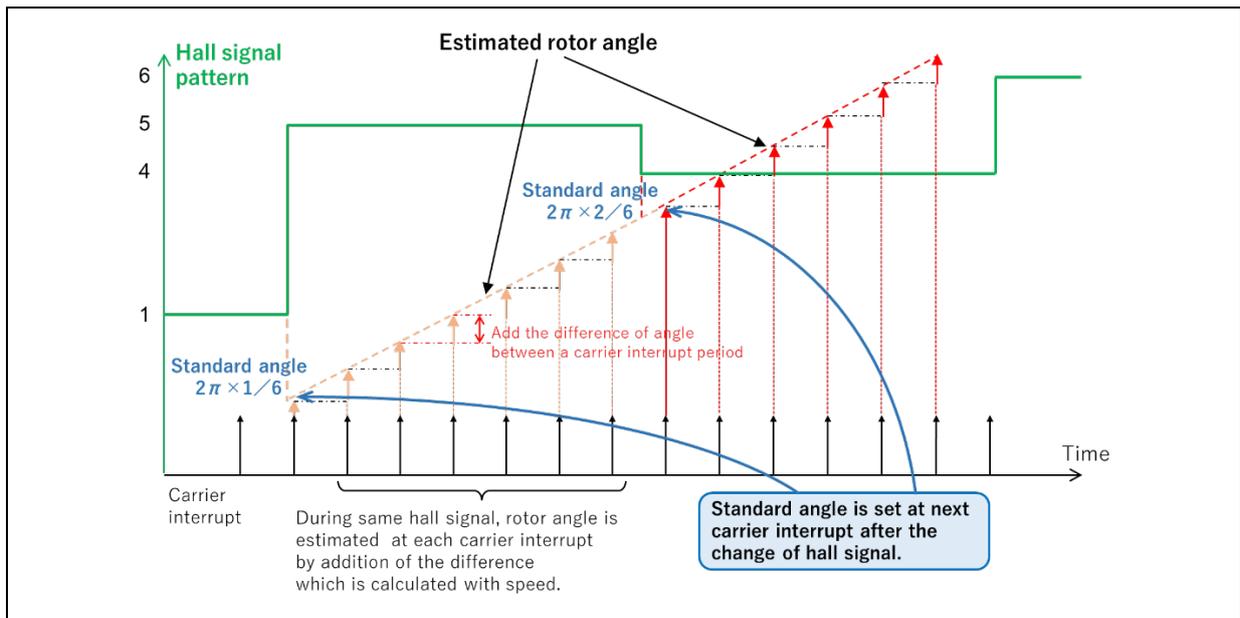


図 7-13 ホール信号内角度推定概念図(正方向回転時)

ホール信号内角度はホール信号変化点において正方向回転であれば $-2\pi/12$ [rad]、逆方向回転であれば $2\pi/12$ [rad]を初期値として設定します。そして、キャリア割り込みが発生するたびに、推定速度情報に基づいて差分値を加算していきます。但し、信号誤差や速度変動を考慮して $-2\pi/12$ [rad]~ $2\pi/12$ [rad]の範囲の値としています。計算された角度は、いずれも $-2\pi/12$ ~ $2\pi/12$ の範囲を超えた場合は丸め込みます。

表 7-5 ホール信号内角度の方向別計算式

方向	ホール信号内角度[rad]
正	初期値($-2\pi/12$) + 推定速度[rad/sec] × T_s × キャリア割り込み回数 または 初期値($-2\pi/12$ [rad]) + キャリア一周期電気角変化量[rad]
逆	初期値($-2\pi/12$) - 推定速度[rad/sec] × T_s × キャリア割り込み回数 または 初期値($2\pi/12$ [rad]) - キャリア一周期電気角変化量[rad]

7.7.5 摩擦補償

始動時において、停止状態から滑らかな加速を行うためには、速度・位置センサの信号を短時間で検出することが重要です。しかし、低分解能であるホールセンサでは、センサの信号が変化する時間間隔が非常に長い特徴があり、高いサンプリング周期での制御に対して、速度検出の応答を高くすることが困難となり、速度調節器が適切なトルクを出力できずに、過大なトルクを出力し、速度が急変する場合があります。

そこで、本サンプルプログラムでは、負荷やモータの特性に合わせて、始動時に加速する際に静止摩擦関連のパラメータをあらかじめ設定し、始動時に過度なトルクを出さないように q 軸電流指令値にフィードフォワード補償する機能を用意しています。なお、静止摩擦が過大な負荷やモータである場合や変化の激しい負荷の場合、速度の急変を所望のレベルに抑制できない場合があります。

摩擦補償について、図 7-14 に例を示します。横軸は速度 ω [rad/s]、縦軸は補償電流値 I_{comp} [A]です。本補償は 4 つのパラメータで構成されます。静止摩擦速度閾値 V_s [rad/s]、静止摩擦補償値 F_s [A]、粘性摩擦係数 F_v [A]、クーロン摩擦補償値 F_c [A]です。補償値の単位はアンペアであり、速度調節器を補償するために用います。

静止摩擦速度閾値 V_s は、停止からモータが回転を始めるときに、摩擦補償値 F_s をどの速度まで適用するかを決めるパラメータです。静止摩擦補償は、回転を始めのために必要な静止摩擦分の電流をあらかじめ設定するものです。

粘性摩擦係数 F_v は、速度に応じて上昇する摩擦量を補償するパラメータです。不明な場合には0を設定してください。0を設定した場合は、速度調節器の積分器が本パラメータの代わりとなって、制御を行います。

クーロン摩擦補償値は、回転時に常にモータの軸にかかる摩擦を補償するものです。例えば、ベアリングや何らかの接触物によって生じるものに対して、補償されます。

これらの補償は、回転方向が逆となった場合には、ソフトウェアで自動的に正負を反転して処理されません。

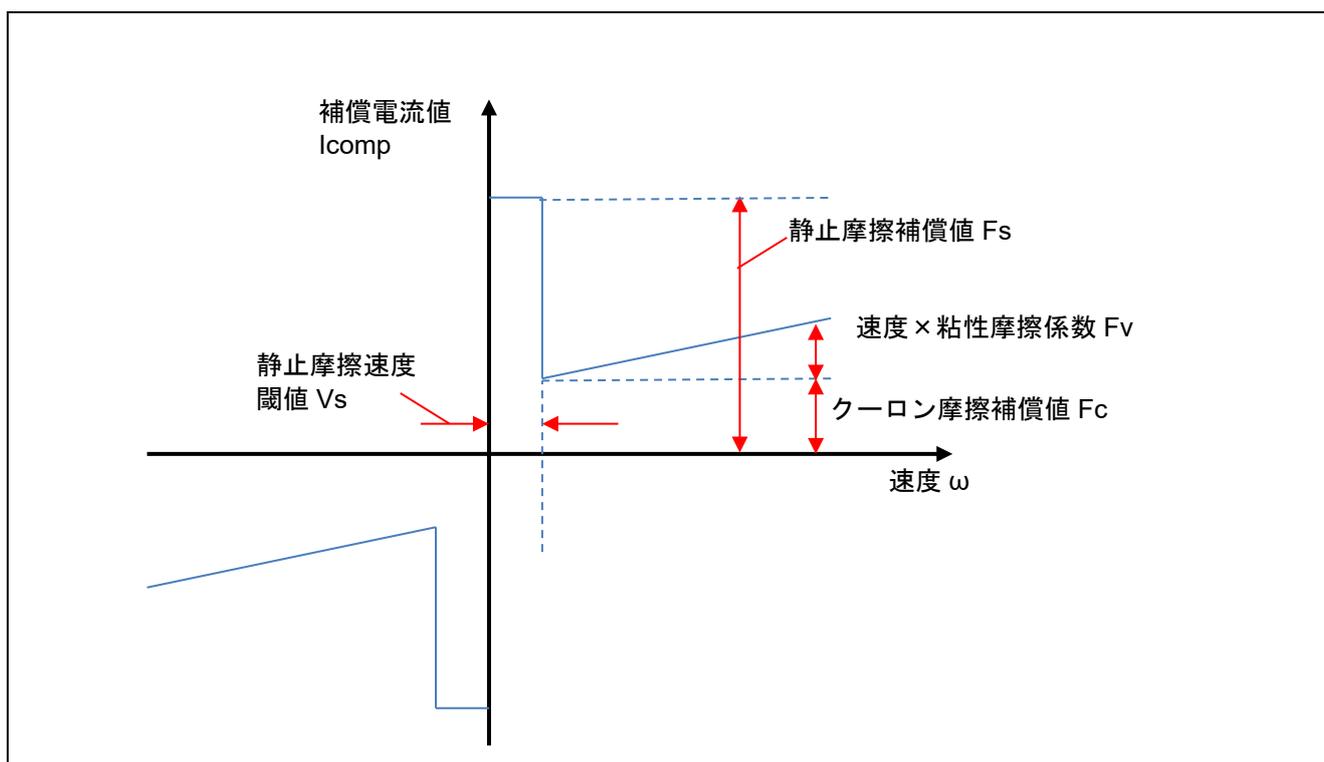


図 7-14 摩擦補償の概念図

摩擦補償処理によって出力される補償電流値 I_{comp} を式で表すと、以下のように表現できます。入力は、速度指令値 $\omega_{ref}[\text{rad/s}]$ と速度検出値 $\omega[\text{rad/s}]$ となり、出力は電流補償値 $[\text{A}]$ となります。

$$I_{comp}(\omega, \omega_{ref}) = \begin{cases} 0, & |\omega_{ref}| < V_s \\ \text{sgn}(\omega)F_s, & |\omega_{ref}| \geq V_s, |\omega| < V_s \\ \text{sgn}(\omega)F_c + F_v\omega & \text{else} \end{cases}$$

以下に、説明した各パラメータと、その設定方法の目安、設定例を示します。理論的に計算することが難しい環境への適用が想定されるため、実機による実験で算定することを想定したパラメータ計算方法としています。

表 7-6 摩擦補償のパラメータ計算方法

パラメータ	設定目安・方法	設定例
静止摩擦補償値 F_s	補償なしで始動する際、モータが動き出したときのトルク指令値(q軸電流指令値)の 80%程度の値を目安に設定してください。	$F_s=0.8 \cdot I_{q_ref}$
クーロン摩擦補償値 F_c	モータに機械系を接続した実際の状況に近い環境において、定格速度の 40%で回転したときの I_q 指令値を目安に設定します。	$F_c=I_{q_ref}$
粘性摩擦係数 F_v	変化が緩やかな負荷条件の場合、速度制御 PI が支配的に動作する場合や、条件がわからない場合は 0 を設定してください。	$F_v=0$
静止摩擦速度閾値 V_s	静止摩擦が発生するしきい値を設定します。目安として、はモータ停止時の速度検出精度を考慮し、小さい値(例: 1.0[rad/s])を設定します。	$V_s=1.0$

7.8 電圧位相進み補償

UVW の三相電圧指令を生成する際に、推定した角度から 0.5 制御周期分、進めた角度で二相三相変換を行います。この処理により、制御の安定性を改善することができます。高速回転用途、PWM キャリア周期が低い場合、間引き処理を行う場合に、改善効果が得られます。

指令演算中、モータの回転が進むことで、角度は常にずれが生じます。このずれを指令演算時間が一定であることを利用し、進む角度を前回の角度移動量から補間する機能となります。

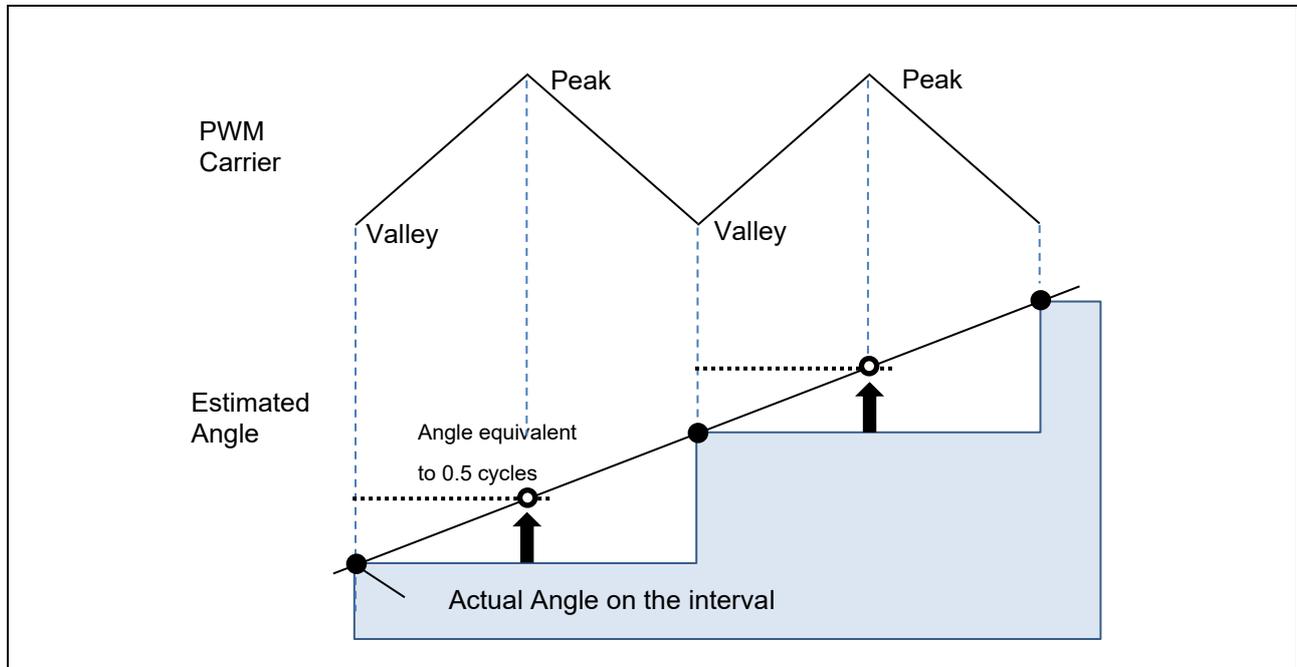


図 7-15 PWM キャリア周期で進む角度量の例

7.9 電圧誤差補償

電圧形 PWM インバータでは、上下アームのスイッチング素子間の短絡を防止するために、上下アーム 2 つの素子が同時にオフとなるデッドタイムを設けています。そのため電圧指令値と実際にモータに印加される電圧には誤差が生じ、制御精度が悪化します。そこでその誤差を低減するため、電圧誤差補償を実装します。

電圧誤差の電流依存性は、電流(向きと大きさ)とデッドタイム、使用するパワー素子のスイッチング特性に依存し、下記のような特性を持ちます。電圧誤差補償では、下記電圧誤差と逆の電圧パターンを電流に応じて電圧指令値に補償します。

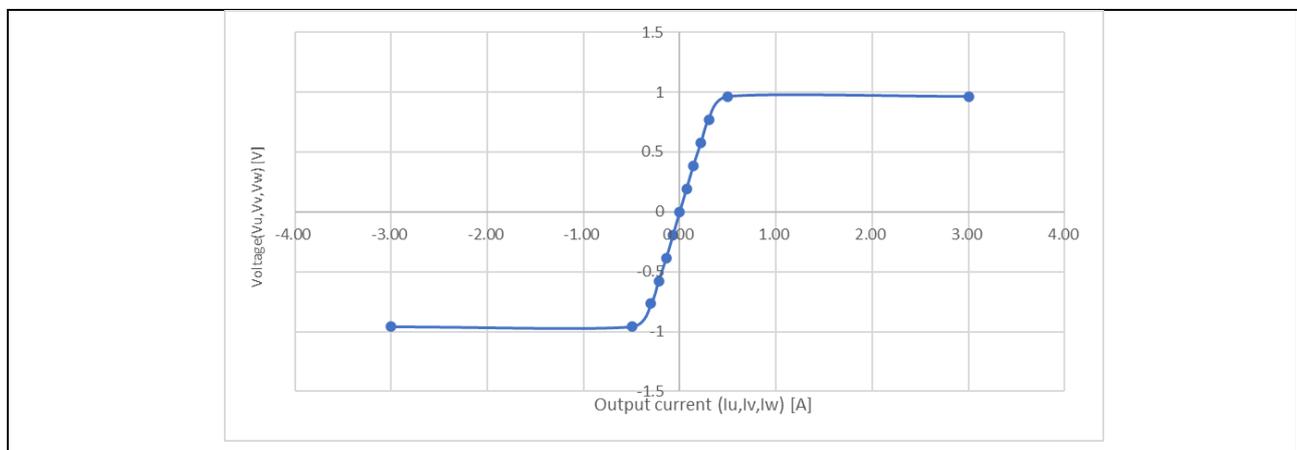


図 7-16 電圧誤差の電流依存性(一例)

7.10 PWM 変調方式

サンプルプログラムでは、モータへの入力電圧はパルス幅変調 (PWM) によって生成します。本モジュールでは、PWM Duty 比の算出を行います。また、電圧利用率を上げるために、変調を行った電圧を出力できます。電流制御モジュールの API を通して変調の動作を設定します。本サンプルプログラムでは、2 種類のパルス幅変調駆動方式から選択できます。

(a) 正弦波変調(MOD_METHOD_SPWM)

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 : PWM 生成用相電圧指令値[V], E :インバータ母線電圧[V]

(b) 空間ベクトル変調(MOD_METHOD_SVPWM)

永久磁石同期モータのベクトル制御において、一般的に所望の各相電圧指令値は正弦波状に生成します。ところが、そのまま PWM 生成のための変調波として使用すると、実際にモータに印加される電圧のインバータ母線電圧に対する電圧利用率は線間電圧換算で最大 86.7[%]となってしまいます。そこで、下記式にあるように各相電圧指令値の最大値と最小値の平均値を算出し、それらを各相電圧指令値から減算したものを変調波として使用します。その結果、変調波の最大振幅は $\sqrt{3}/2$ 倍となり、線間電圧はそのままに電圧利用率は 100[%]となります。

$$\begin{pmatrix} V'_u \\ V'_v \\ V'_w \end{pmatrix} = \begin{pmatrix} V_u \\ V_v \\ V_w \end{pmatrix} + \Delta V \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\therefore \Delta V = -\frac{V_{max}+V_{min}}{2}, \quad V_{max} = \max\{V_u, V_v, V_w\}, \quad V_{min} = \min\{V_u, V_v, V_w\}$$

V_u, V_v, V_w : U,V,W 相電圧指令値

V'_u, V'_v, V'_w : PWM 生成用 U,V,W 相電圧指令値(変調波)

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' : PWM 生成用相電圧指令値[V], E :インバータ母線電圧[V]

8. ソフトウェア仕様・構成

8.1 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。

表 8-1 ホールセンサベクトル制御ソフトウェア基本仕様

項目	内容	
制御方式	ホールセンサベクトル制御	
モータ制御開始/停止	RMW およびボード UI からの入力	
回転子磁極位置検出	ホールセンサ	
入力電圧	DC 24V	
PWM キャリア周波数	20 [kHz]、キャリア周期：50 [μs] (谷割り込み)	
PWM 変調方式	三角波比較法または空間ベクトル変調法	
デッドタイム	2.0 [μs]	
制御周期 (電流)	50 [μs]	
制御周期 (速度)	500 [μs]	
速度指令値管理	CW : 0 [r/min] to 2400 [r/min] CCW : 0 [r/min] to -2400 [r/min]	
各制御系固有周波数	電流制御系 : 300 Hz 速度制御系 : 5 Hz	
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	<p>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</p> <ol style="list-style-type: none"> 1. 各相の電流ピーク値が 10 [A] を超過 (50 [μs] 毎に監視) 2. 定格電流からの計算値(4.723A)を超過 (50 [μs] 毎に監視) 3. インバータ母線電圧が 60 [V] を超過 (50 [μs] 毎に監視) 4. インバータ母線電圧が 8 [V] 未満 (50 [μs] 毎に監視) 5. 回転速度が 2850 [r/min] を超過 (50 [μs] 毎に監視) <p>外部からの過電流検出信号 (POE) 及び出力短絡を検出した場合、PWM 出力端子をハイインピーダンスにする</p>	

8.2 ソフトウェア全体構造

ソフトウェアの全体構成を図 8-1 に示します。

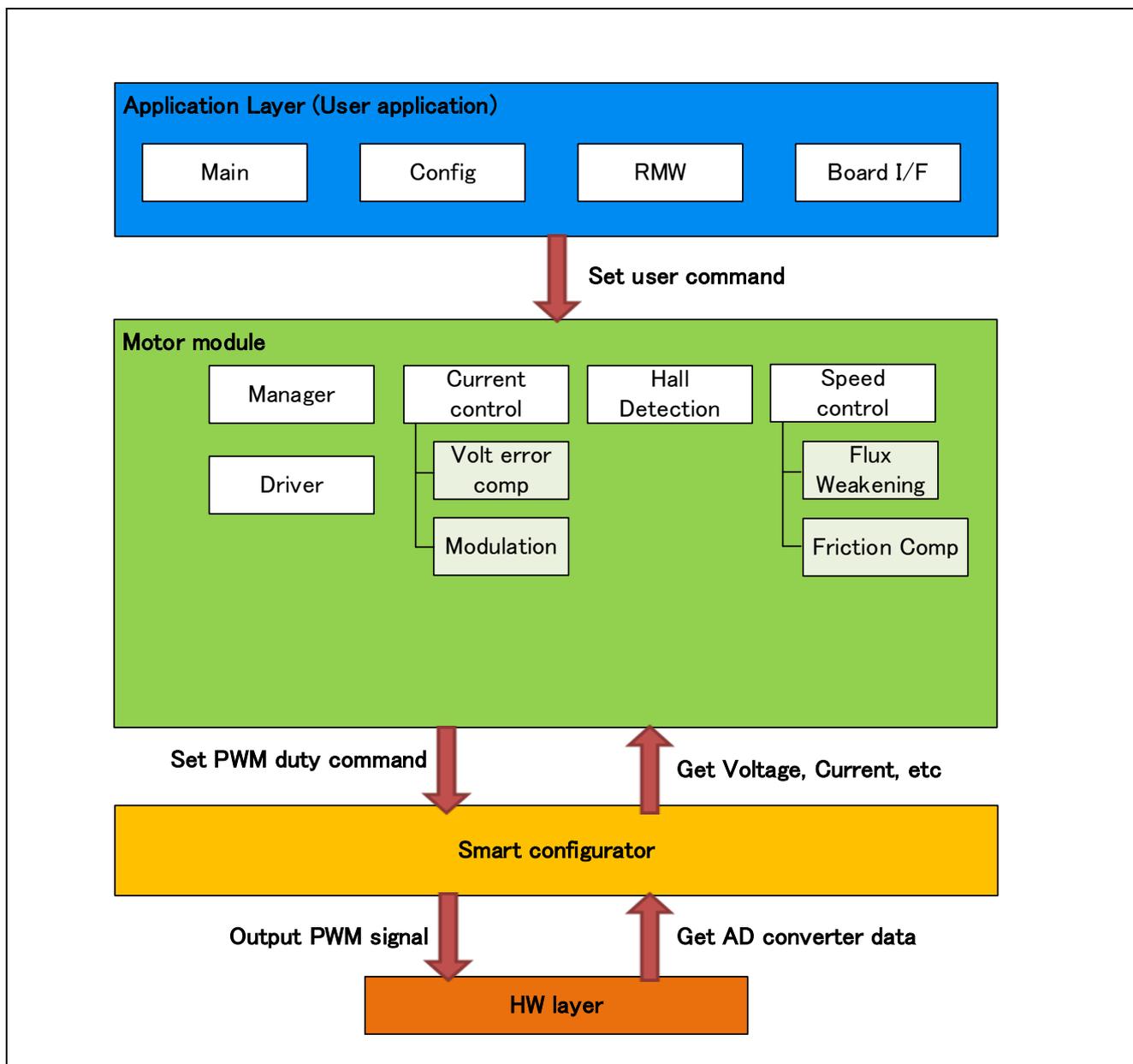


図 8-1 モータ制御ソフトウェアの全体構成

8.3 タスクの説明

モータ制御タスクは、0.5ms 周期の速度制御を行うタスクと、PWM キャリア周期(20kHz, 50us)に同期したタスクの2つに分かれています。

表 8-2 使用する割り込み・タスク

タスク	周辺機能	割り込み関数	コールされる関数
モータ制御 割り込み (速度制御)	CMT0	r_Config_CMT0_cmi0_interrupt	R_MOTOR_HALL_ VECTOR_SpeedInterrupt
モータ制御 割り込み (電流制御)	MTU3,4	r_Config_MOTOR_ad_interrupt	R_MOTOR_HALL_ VECTOR_CurrentInterrupt
アイドル	-	r_app_main_ui_mainloop	R_MOTOR_HALL_VECTOR_StatusGet
起動時 1	-	r_app_main_init_motor_ctrl	R_MOTOR_HALL_VECTOR_Open
起動時 2	-	r_app_main_start_motor_ctrl	(ドライバ関連のみ)
リセット時	-	※エラー復帰時に状態遷移処理 中で実行されます	R_MOTOR_HALL_VECTOR_ MotorReset
RMW 操作	-	r_app_rmw_system_mode	R_MOTOR_HALL_VECTOR_ MotorStart MotorStop MotorErrorCancel CtrlTypeSet

8.4 ファイル・フォルダ構成

サンプルプログラムのフォルダとファイル構成を表 8-3 に示します。

表 8-3 ファイル・フォルダ構成

フォルダ	サブフォルダ	ファイル	備考	
app	main	r_app_main.c/h	ユーザメイン関数	
	board_ui	r_app_board_ui.c/h	ボード UI 関連関数定義	
		r_app_board_ui_ctrl.h	MCU 依存のボード UI 関数定義	
		r_app_board_ui_ctrl_rx26t_mci1v1.c	MCU 依存のボード UI 関数定義	
	rmw	r_app_rmw.c/h	RMW の Analyzer UI 関連関数定義	
		r_app_rmw_mcu.h	RMW の HW 初期化関連マクロ定義	
		r_app_rmw_interrupt.c	RMW の割り込み関数定義	
		ICS2_RX26T.lib/h	RMW の通信用ライブラリ	
	cfg	r_app_control_cfg.h	アプリ層のコンフィグレーション定義	
	motor_module	hall_vector	r_motor_hall_vector_action.c	アクション関数定義
r_motor_hall_vector_api.c/h			マネージャモジュールの API 関数定義	
r_motor_hall_vector_manager.c/h			マネージャモジュールのローカル関数定義	
r_motor_hall_vector_protection.c/h			保護機能の関数定義	
r_motor_hall_vector_statemachine.c/h			状態遷移関連の関数定義	
current		r_motor_current_api.c/h	電流制御モジュールの API 関数定義	
		r_motor_current.c/h	電流制御モジュールのローカル関数定義	
		r_motor_current_modulation.c/h	変調モジュールの関数定義	
		r_motor_current_volt_err_comp.c/h	電圧誤差補償モジュールの関数定義	
		r_motor_current_pi_gain_calc.c	電流制御モジュールの制御ゲイン算出関数定義	
speed		r_motor_speed_api.c/h	速度制御モジュールの API 関数定義	
		r_motor_speed.c/h	速度制御モジュールのローカル関数定義	
		r_motor_speed_fluxwkn.c/h	弱め磁束制御モジュールの関数定義	
		r_motor_speed_pi_gain_calc.c	速度制御モジュールの制御ゲイン算出関数定義	
		r_motor_speed_friction_comp.c/h	摩擦補償モジュールの関数定義	
sensor/hall		r_motor_sensor_hall_api.c/h	ホールセンサモジュールの関数定義	
driver		r_motor_driver.c/h	ドライバモジュールの関数定義	
general		r_motor_filter.c/h	汎用フィルタ関数定義	
		r_motor_pi_control.c/h	PI 制御関数定義	
		r_motor_common.h	共通定義	
cfg		r_motor_inverter_cfg.h	インバータのコンフィグレーション定義	
		r_motor_module_cfg.h	制御モジュールのコンフィグレーション定義	
		r_motor_targetmotor_cfg.h	モータのコンフィグレーション定義	
src		smc_gen	表 8-4 を参照	スマート・コンフィグレータで生成されたドライバ及び API

スマート・コンフィグレータを使用することで、周辺機能ドライバを簡単に生成することができます。

スマート・コンフィグレータは、プロジェクトで使用するマイクロコントローラ、周辺機能、端子機能などの設定情報をプロジェクト・ファイル (*.scfg) に保存し、参照します。本サンプルプログラムの周辺機能設定を確認する場合、以下のファイルを参照してください。

“RX26T_MCBA_MCILV1_SPM_HALL_FOC_xxx_Vyyy.scfg”

(xxx : CSP は CS+版、E2S は e² studio 版を意味します。yyy : リビジョン番号)

スマート・コンフィグレータで生成したフォルダとファイル構成を下記に示します。

表 8-4 スマート・コンフィグレータのフォルダ・ファイル構成

フォルダ	サブフォルダ	サブフォルダ 2	ファイル	備考
src	smc_gen	Config_S12AD2	Config_S12AD2.c/h	12bitADC 関連関数定義
			Config_S12AD2_user.c	12bitADC 関連ユーザ関数定義
		Config_PORT	Config_PORT.c/h	ポート関連関数定義
			Config_PORT_user.c	ポート関連ユーザ関数定義
		Config_CMT0	Config_CMT0.c/h	制御周期用 CMT 関連関数定義
			Config_CMT0_user.c	制御周期用 CMT 関連ユーザ関数定義
		Config_MOTOR	Config_MOTOR.c/h	モータコンポーネント関連関数定義
			Config_MOTOR_user.c	モータコンポーネント関連ユーザ関数定義
		Config_IWDT	Config_IWDT.c/h	IWDT 関連関数定義
			Config_IWDT_user.c	IWDT 関連ユーザ関数定義
		Config_POE	Config_POE.c/h	POE 関連関数定義
			Config_POE_user.c	POE 関連ユーザ関数定義

上記表の他に、スマート・コンフィグレータ使用時に 4 つのフォルダが自動生成されます。

r_bsp : 様々な BSP (BSP : Board Support Package) ファイルを含みます。詳細は “r_bsp” フォルダ内の “readme.txt” ファイルを参照してください。

general : スマート・コンフィグレータ生成ドライバで共通で使用される様々なファイルを含みます。

r_config : MCU パッケージ、クロック、割り込み、R_xxx_Open の名前を持つドライバ初期化関数のコンフィグレーションヘッダファイルを含みます。

r_pincfg : ピン設定に関する様々なファイルを含みます。

8.5 アプリケーション層

アプリケーション層はユーザインタフェース(UI)の選択と RMW を使用したモータモジュールに対する制御の指令値設定や制御モジュールのパラメータ更新を行っています。サンプルプログラムでは、RMW を使用 (RMW UI) して、設定及び処理を行っています。また、これらの UI からモータの駆動/停止や、制御の指令値設定などを行っています。

8.5.1 機能

アプリケーション層で設定している機能一覧を表 8-5 に示します。

表 8-5 アプリケーション層の機能一覧

機能	説明
メイン処理	ユーザの指令に対してシステムを有効 / 無効に設定します。
UI 処理	ボード UI ならびに RMW の管理を行います。
ボード UI 処理	基板上のスイッチ・VR からの速度指令値の取得・設定を行います。
RMW の UI 処理	指令値含むパラメータの取得・設定を行います。

8.5.2 コンフィグレーション情報

アプリケーション層で使用するコンフィグレーション情報を表 8-6 コンフィグレーション情報一覧に示します。

表 8-6 コンフィグレーション情報一覧

ファイル名	マクロ名	説明
r_app_control_cfg.h	APP_CFG_USE_UI	UI の初期値設定 RMW : MAIN_UI_RMW BOARD : MAIN_UI_BOARD
	APP_CFG_FREQ_BAND_LIMIT	電流制御、速度制御の固有周波数が近い値にならないための制限値。
	APP_CFG_MAX_CURRENT_OMEGA	電流制御系固有周波数の上限値[Hz]
	APP_CFG_MIN_OMEGA	固有周波数の下限値[Hz]
	APP_CFG_SCI_CH_SELECT	RMW 用 SCI のチャンネルセレクト

表 8-7 コンフィグレーション情報初期値一覧

マクロ名	設定値
APP_CFG_USE_UI	MAIN_UI_RMW
APP_CFG_FREQ_BAND_LIMIT	3.0f
APP_CFG_MAX_CURRENT_OMEGA	1500.0f
APP_CFG_MIN_OMEGA	1.0f
APP_CFG_SCI_CH_SELECT	ICS_SCI6_P81_P80

8.5.3 構造体・変数情報

アプリケーション層でユーザが使用可能な変数一覧を表 8-8 に示します。また、RMW を使用してモータモジュールのパラメータを更新するための構造体を用意しており、その構造体メンバを表 8-9 に示します。

表 8-8 に示す変数は、RMW から値を設定することで、本アプリケーション層が表 8-9 に示す構造体を介して、各制御モジュールの変数に、変更した値が各モジュールの Update 関数を介して反映されます。

表 8-8 変数一覧

変数	説明
g_st_rmw_input_buffer	RMW 変数更新用構造体
g_u1_update_param_flag	バッファ転送完了フラグ
com_u1_system_mode	ユーザ入力用システムモード切り替え変数 0: 停止モード 2: 速度制御モード 3: トルク制御モード 4: オープンループ制御モード 5: エラー解除 8: 電流オフセット調整
g_u1_system_mode	システムモード 0: 停止モード 2: 速度制御モード 3: トルク制御モード 4: オープンループ制御モード 5: エラー解除 8: 電流オフセット調整
com_u1_enable_write	ユーザ入力用変数書き換え許可
g_u1_enable_write	変数書き換え許可
com_u1_sw_userif	ユーザ入力用 UI 切り替え変数 0: RMW UI 1: BOARD UI
g_u1_sw_userif	UI 切り替え用変数
com_u2_offset_calc_time	電流オフセット値計算時間設定
com_f4_offset_rotor_rad	磁極位置オフセット[rad]
com_u2_mtr_pp	駆動するモータの極対数
com_f4_mtr_r	駆動するモータの抵抗 [Ω]
com_f4_mtr_ld	駆動するモータの d 軸インダクタンス [H]
com_f4_mtr_lq	駆動するモータの q 軸インダクタンス [H]
com_f4_mtr_m	駆動するモータの磁束 [Wb]
com_f4_mtr_j	駆動するモータのロータイナーシャ [kgm^2]
com_f4_nominal_current_rms	駆動するモータの定格電流 [Arms]
com_f4_max_speed_rpm	駆動するモータの速度最大値 (機械角) [r/min]
com_f4_current_omega_hz	電流制御系固有周波数 [Hz]

変数	説明
com_f4_current_zeta	電流制御系減衰係数
com_f4_speed_omega_hz	速度制御系固有周波数 [Hz]
com_f4_speed_zeta	速度制御系減衰係数
com_f4_ref_speed_rpm	速度指令値（機械角）[r/min]
com_f4_speed_rate_limit_rpm	速度指令最大増減幅 [r/min/s]（速度制御時使用,機械角）
com_f4_overspeed_limit_rpm	速度制限値（機械角）[r/min]
com_u1_flag_volt_err_comp_use	電圧誤差補償の設定 0：無効, 1：有効
com_u1_flag_fluxwkn_use	弱め磁束制御の設定 0：無効, 1：有効
com_f4_friction_comp_static	静止摩擦係数[A]
com_f4_friction_comp_brake_speed	摩擦時ブレーキ速度[r/min]
com_f4_friction_comp_coulombs	クーロン摩擦係数[A]
com_f4_friction_comp_viscous	粘性摩擦係数[A/(rad/s)]
com_f4_hall_speed_select	ホールセンサの速度検出方式選択： 0：自動切り替え 1：過去 6 回分(一周)で計算 2：過去 1 回分で計算
com_f4_hall_angle_correct	角度補正[rad]
com_f4_hall_spdmix_dev_th	自動切り替え時の許容する速度偏差率(範囲：0~1.0)
com_f4_hall_spdmix_elec_speed_th	自動切り替え時、過去 6 回分方式に切り替え始める電気角速度 閾値(電気角)[rad/s]
s_u1_cnt_ics	ICS watchpoint のスキップ回数カウンタ
com_f4_torque_ref_nm	トルク指令値[Nm]
com_f4_ol_crnt_ref	オープンループ制御時の電流指令値[A]

表 8-9 RMW によるパラメータ更新用構造体の変数一覧

構造体	変数	説明
st_rmw_param_buffer_t	u2_offset_calc_time	電流オフセットの検出時間設定
RMW 変数更新用構造体	f4_offset_rotor_rad	磁極位置オフセット[rad]
	st_motor	モータパラメータ用の構造体
	f4_max_speed_rpm	最大速度 [r/min] (機械角)
	f4_friction_comp_static	静止摩擦係数[A]
	f4_friction_comp_brake_speed	摩擦時ブレーキ速度[r/min]
	f4_friction_comp_coulombs	クーロン摩擦係数[A]
	f4_friction_comp_viscous	粘性摩擦係数[A/(rad/s)]
	u1_hall_speed_select	ホールセンサの速度検出方式選択： 0：自動切り替え

構造体	変数	説明
		1 : 過去 6 回分(一周)で計算 2 : 過去 1 回分で計算
	f4_hall_angle_correct_rad	角度補正[rad]
	f4_hall_spdmix_elec_speed_th	自動切り替え時、過去 6 回分方式に切り替え始める電気角速度閾値(電気角)[rad/s]
	f4_hall_spdmix_dev_th	自動切り替え時の許容する速度偏差率(範囲 : 0~1.0)
	f4_current_omega_hz	電流制御系固有周波数 [Hz]
	f4_current_zeta	電流制御系減衰係数
	f4_speed_omega_hz	速度制御系固有周波数 [Hz]
	f4_speed_zeta	速度制御系減衰係数
	f4_ref_speed_rpm	速度指令値 [r/min] (機械角)
	f4_speed_rate_limit_rpm	速度の変化量制限 [r/min/s] (機械角)
	f4_overspeed_limit_rpm	速度制限値 [r/min] (機械角)
	f4_torque_ref_nm	トルク指令値[Nm]
	f4_ol_crnt_ref	オープンループモード電流指令値[A]
	u1_flag_volt_err_comp_use	電圧誤差補償使用有無のフラグ
	u1_flag_fluxwkn_use	弱め磁束制御の使用有無フラグ

8.5.4 マクロ定義

マクロ一覧を表 8-10 に示します。

表 8-10 マクロ一覧

ファイル名	マクロ名	定義値	備考
r_app_main.h	MAIN_UI_RMW	0	RMW UI 使用
	MAIN_UI_BOARD	1	ボード UI 使用
	MAIN_UI_SIZE	2	UI 選択可能数
r_app_rmw.h	ICS_DECIMATION	5	RMW watchpoint のスキップ回数
	ICS_INT_LEVEL	6	RMW 割り込みの優先度
	ICS_BRR	251	RMW の通信ボーレート
	ICS_INT_MODE	1	RMW の通信モード選択
	ICS_SCI_CH_SELECT	APP_CFG_SCI_C H_SELECT	使用 SCI チャンネル

【注】 RMW で通信を行うためのチャンネルを定義したマクロが ICS2_RX26T.h に用意されています。

8.5.5 パラメータ調整・設定

アプリケーション層のコンフィグレーション情報は r_app_control_cfg.h で設定する必要があります。

表 8-8 に示す変数の設定・更新は、RMW から行ってください。RMW の操作は 6.7 及び Renesas Motor Workbench V.3.10 ユーザーズマニュアル (R21UZ0004) を参照ください。

8.6 マネージャモジュール・モータ制御モジュール

マネージャモジュールは、モータ制御モジュールに含まれる各制御モジュールを適切に使用してモータ制御を行うモジュールです。各モジュールのインターフェースやモータ制御のシステム全体の管理、システム保護などを行っています。

8.6.1 機能

マネージャモジュールの機能一覧を表 8-11 に示します。モータ制御モジュールの機能の一覧を表 8-12 及び表 8-13 に示します。

表 8-11 マネージャモジュールの機能一覧

機能	説明
モード管理	ユーザの指令に対してシステムを切り替えてモータを制御します。
保護機能	システム保護機能によりエラー処理を行います。
制御方式の管理	速度制御や電流制御の状態の取得・設定を行います。
速度・位置情報の取得	速度制御モジュールや電流制御モジュールから、速度・位置情報の取得を行います。
制御モジュールの指令値設定	電流制御モジュール、速度制御モジュールに対して入力する指令値を制御の状態から選択します。
割り込み処理	スマート・コンフィグレータで設定した割り込みを受けて処理を行い、適切なモジュールへ処理の割り振りを行います。

表 8-12 速度制御モジュールの機能一覧

機能	説明
速度制御	速度指令値に追従するよう演算を行い、電流指令値を出力します。
速度指令設定	速度モジュールに速度指令値を設定します。
弱め磁束制御	d 軸電流を制御し、定格回転速度以上でも運転できるように制御します。

表 8-13 電流制御モジュールの機能一覧

機能	説明
電流制御	電流指令値に追従するよう演算を行い、PWM 出力値を設定します。
電流オフセット調整	AD で検出した電流値のオフセット値を計算します。
電圧誤差補償	出力電圧のデッドタイムによる影響を補償します。
順変換、逆変換	ベクトル制御を行うために検出した電流値に対して、座標変換を行います。演算結果に対して座標の逆変換を行い元の座標軸に戻します。
PWM 変調	PWM 信号に変調して効率を改善します。
非干渉制御	dq 軸の干渉を防ぐために干渉を打ち消す演算を行います。
サンプルディレイ補償	3 相電圧指令値を生成する際に電流制御サイクル分のサンプル遅延を補償します。
摩擦補償	始動時の摩擦を補償し、低分解能なホールセンサによって生じる始動時の速度変動を低減します。

表 8-14 センサモジュールの機能一覧

機能	説明
ホールセンサ角度・速度推定	ホールセンサの信号から角度と速度を計算・推定します。
速度推定方式選択	API で速度推定の方式を選択できます。

8.6.2 モジュール構成図

モジュール構成図を図 8-2 に示します。

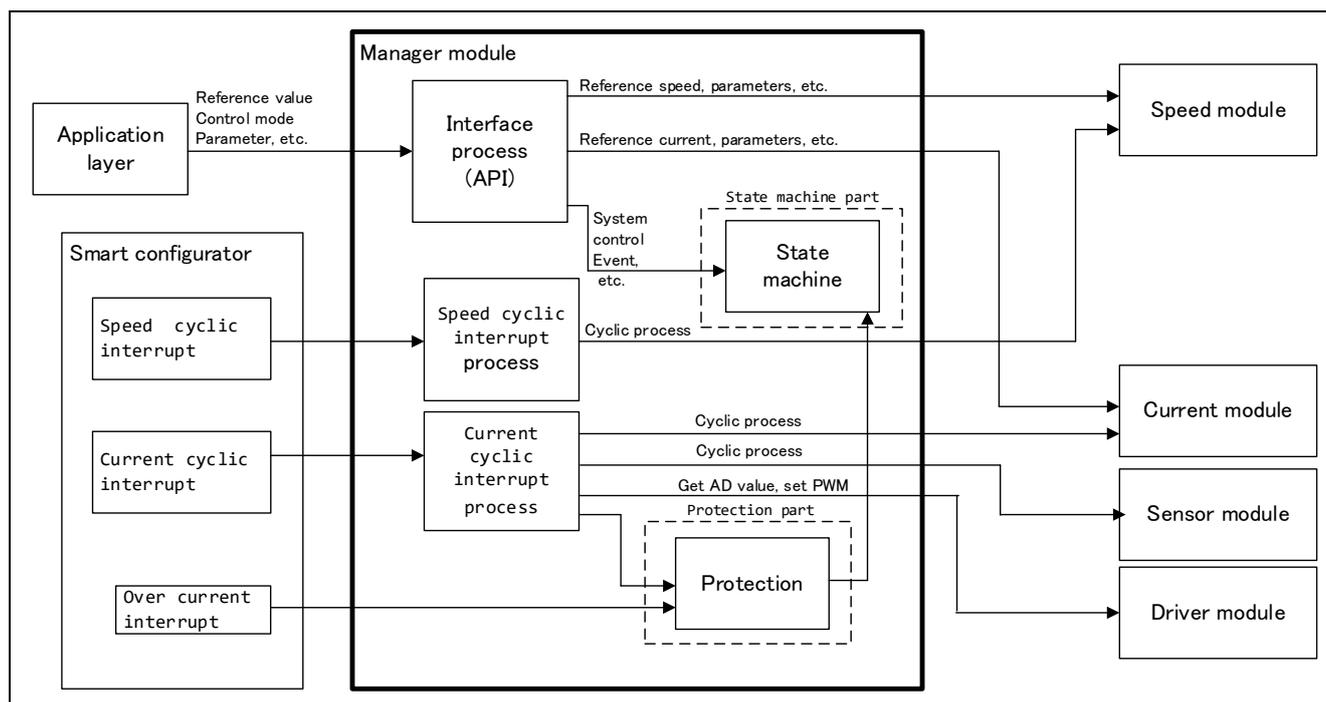


図 8-2 マネージャモジュール構成図

8.6.3 モード管理

図 8-3 に本サンプルプログラムにおける状態遷移図を示します。本サンプルプログラムでは、「SYSTEM MODE」と、「RUN MODE」により状態を管理します。

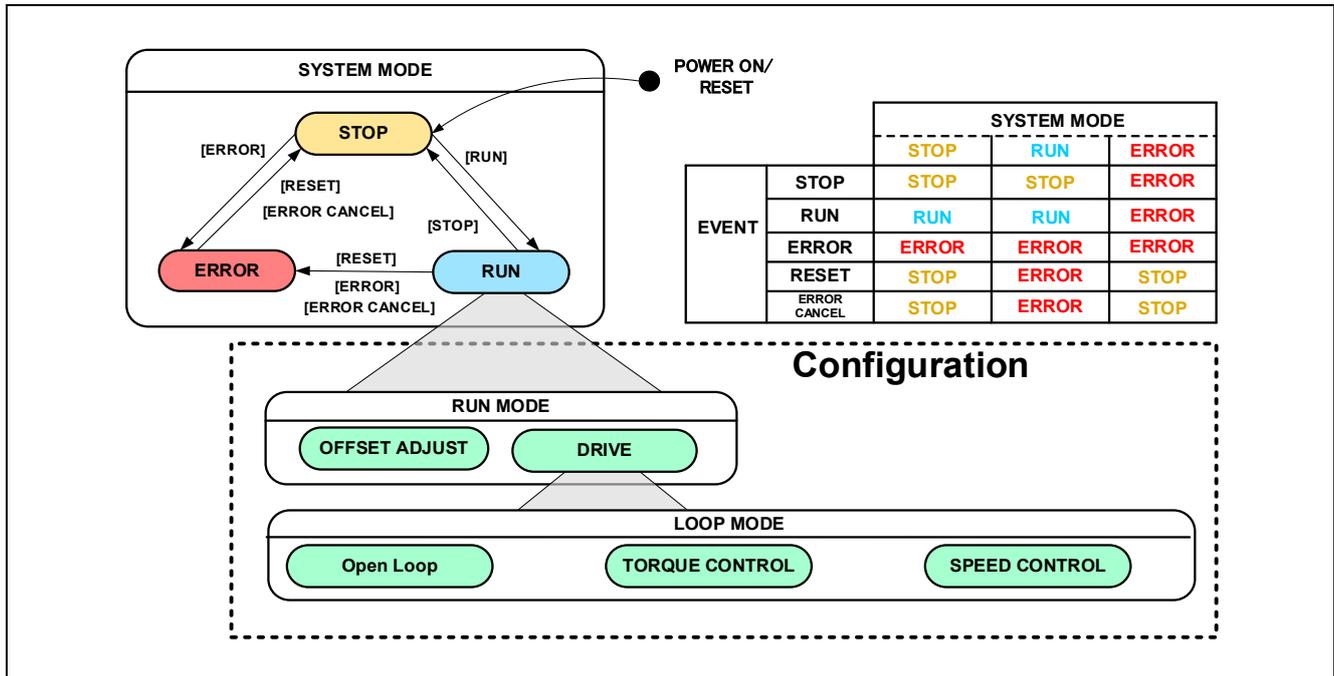


図 8-3 ホールセンサベクトル制御ソフトウェアの状態遷移図

(1) SYSTEM MODE

システム動作状態を表します。各イベント（EVENT）の発生により、状態が遷移します。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・LOOP MODE

モータの制御状態を表します。システムの状態が ACTIVE になると、モータは図 8-3 の Configuration 設定に従って駆動または校正が実施されます。

(3) EVENT

各 SYSTEM MODE 中に EVENT が発生すると、その EVENT に従って、システム動作状態が図 8-3 中の表のように遷移します。各 EVENT の発生要因は下記となります。

表 8-15 EVENT 一覧

イベント名	発生要因
INACTIVE	ユーザ操作により発生します
ACTIVE	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
ERRORCANCEL	ユーザ操作により発生します

8.6.4 シーケンスの説明

本サンプルプログラムでは、モードと呼ばれる運転・停止・エラーを管理する状態遷移に加えて、ホールセンサベクトル制御を行うための運転状態を管理するシーケンスの2つを有しています。モードは、8.6.3を参照してください。ここでは、後者の運転状態を管理するシーケンスを説明します。以下に示します。

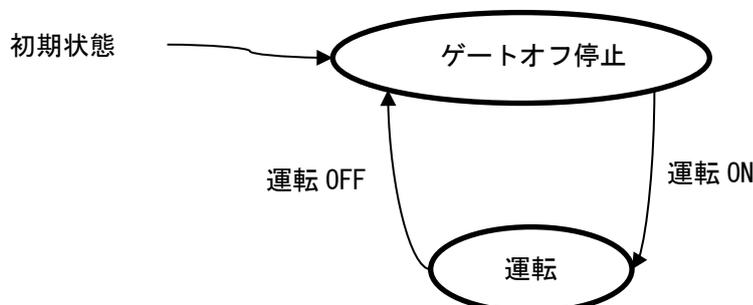


図 8-4 運転シーケンスの状態遷移図

表 8-16 運転シーケンスの状態とその説明

状態	状態の説明
初期状態	CPU が初期化される前の状態です。
ゲートオフ停止	CPU ボードの電源が ON の状態です。インバータの電源が OFF の場合、運転 ON には推移しません。運転 ON した場合、低電圧保護により、エラーが発生します。
運転	ホールセンサ制御アルゴリズムを用いてベクトル制御を行います。

8.6.5 始動

マネージャモジュールでは R_MOTOR_HALL_VECTOR_CtrlTypeSet の API を用いて、回転時に使用する制御方法や始動時のセンサの調整を行います。この設定は停止時のみ変更可能です。

上記の API 設定で RUN MODE と LOOP MODE が設定され、各制御モジュールの入力選択が決まり、次の始動からそれによってモータを制御します。例に速度制御の始動の動きを図 8-5 に示します。

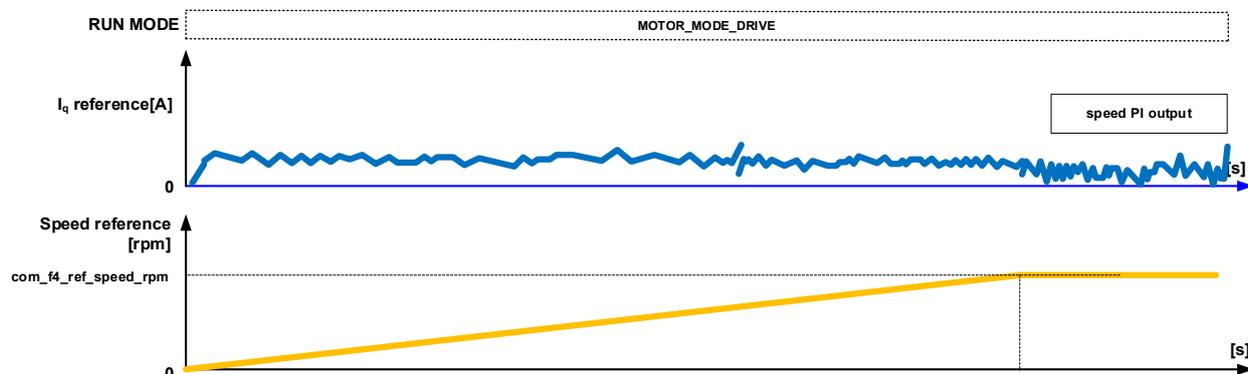


図 8-5 ホールセンサベクトル制御の速度制御始動制御内容

8.6.6 保護機能

本制御プログラムは、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 8-17 を参照してください。

- 過電流エラー
過電流エラーはハードウェア及びソフトウェア両方で検出されます。
ハードウェアからの緊急停止信号（過電流検出）により、PWM 出力端子をハイインピーダンス状態にします。また、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）。
過電流リミット値はモータの定格電流 (MOTOR_CFG_NOMINAL_CURRENT_RMS) から自動で計算されます。
- 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 8-17 各システム保護機能の動作条件・設定値

過電流エラー	過電流リミット値 [A]	10
	監視周期 [μ s]	電流制御周期*1
過電圧エラー	過電圧リミット値 [V]	60
	監視周期 [μ s]	電流制御周期*1
低電圧エラー	低電圧リミット値 [V]	8
	監視周期 [μ s]	電流制御周期*1
回転速度エラー	速度リミット値 [r/min]	2850
	監視周期 [μ s]	電流制御周期*1

【注】 1. 表 8-1 ホールセンサベクトル制御ソフトウェア基本仕様参照

8.6.7 API

マネージャモジュールの API 一覧を表 8-18 に示します。

表 8-18 API 一覧

API	説明
R_MOTOR_HALL_VECTOR_Open	本モジュールと使用するモジュールのインスタンスを生成します。
R_MOTOR_HALL_VECTOR_Close	本モジュールをリセット状態にします。
R_MOTOR_HALL_VECTOR_Reset	モジュールの初期化を行います。
R_MOTOR_HALL_VECTOR_ParameterUpdate	本モジュールの制御パラメータを更新します。また、関連するモジュールの制御パラメータ更新を行います。
R_MOTOR_HALL_VECTOR_MotorStart	モータ駆動状態にします。
R_MOTOR_HALL_VECTOR_MotorStop	モータ停止状態にします。
R_MOTOR_HALL_VECTOR_MotorReset	システムのエラー状態を解除し、パラメータをリセットします。
R_MOTOR_HALL_VECTOR_MotorCancel	システムのエラー状態を解除します、パラメータリセットなし。
R_MOTOR_HALL_VECTOR_ErrorSet	システムにエラー状態を設定します。
R_MOTOR_HALL_VECTOR_SpeedSet	速度指令値を設定します。速度制御時に有効になります。
R_MOTOR_HALL_VECTOR_SpeedGet	速度情報を取得します。
R_MOTOR_HALL_VECTOR_StatusGet	ステータスの状態を取得します。
R_MOTOR_HALL_VECTOR_ErrorStatusGet	エラー状態を取得します。
R_MOTOR_HALL_VECTOR_CtrlTypeSet	制御方式を設定します。制御方式を変更する場合は、モータを停止状態にしてください。制御方式は枚挙型 e_mtr_ctrl_type_t で定義されます。 0 : 位置制御(Not available) 1 : 速度制御 2 : トルク制御 3 : 電圧制御(Not available) 4 : 電流制御(位置オープンループ制御) 5 : 電流オフセット調整
R_MOTOR_HALL_VECTOR_LoopModeStatusGet	制御方式を取得します。 結果は枚挙型 e_mtr_ctrl_type_t の定義に従います。
R_MOTOR_HALL_VECTOR_RotorAngleOffsetSet	角度オフセットの設定を行います。
R_MOTOR_HALL_VECTOR_CurrentOffsetSet	AD 電流オフセット設定を行います。
R_MOTOR_HALL_VECTOR_CurrentOpenLoopSet	オープンループの電流指令値の設定を行います。
R_MOTOR_HALL_VECTOR_TorqueSet	トルク制御モードのトルク指令値の設定を行います。
R_MOTOR_HALL_VECTOR_OffsetCalibrationStatusGet	オフセット調整の状態を取得します。

API	説明
R_MOTOR_HALL_VECTOR_SpeedInterrupt	速度制御を行うための割り込み処理を行います。
R_MOTOR_HALL_VECTOR_CurrentInterrupt	電流制御を行うための割り込み処理を行います。
R_MOTOR_HALL_VECTOR_OverCurrentInterrupt	過電流が発生した際の割り込み処理を行います。

表 8-19 電流制御モジュールの API 一覧

API	説明
R_MOTOR_CURRENT_Open	電流制御モジュールのインスタンスを生成します。
R_MOTOR_CURRENT_Close	電流制御モジュールをリセット状態にします。
R_MOTOR_CURRENT_Reset	電流制御モジュールの初期化をします。
R_MOTOR_CURRENT_Run	電流制御モジュールをアクティブ状態にします。
R_MOTOR_CURRENT_ParameterSet	電流制御に使用する変数情報を入力します。
R_MOTOR_CURRENT_ParameterGet	電流制御結果の出力を取得します。
R_MOTOR_CURRENT_ParameterUpdate	電流制御モジュールの制御パラメータを更新します。
R_MOTOR_CURRENT_CurrentCyclic	電流制御を行います。
R_MOTOR_CURRENT_OffsetCalibration	電流検出のオフセット調整を行います。
R_MOTOR_CURRENT_CurrentOffsetRemove	電流検出オフセット値を除いた値を返します。
R_MOTOR_CURRENT_VoltErrCompParamSet	電圧誤差補償パラメータ設定を行います。
R_MOTOR_CURRENT_CurrentOffsetSet	AD 電流オフセット値設定を行います。

表 8-20 速度制御モジュールの API 一覧

API	説明
R_MOTOR_SPEED_Open	速度モジュールのインスタンスを生成します。
R_MOTOR_SPEED_Close	モジュールをリセット状態にします。
R_MOTOR_SPEED_Reset	モジュールの初期化します。
R_MOTOR_SPEED_Run	モジュールをアクティブ状態にします。
R_MOTOR_SPEED_ParameterSet	速度制御に使用する変数情報を入力します。
R_MOTOR_SPEED_ParameterGet	速度制御結果の出力を取得します。
R_MOTOR_SPEED_ParameterUpdate	モジュールの制御パラメータを更新します。
R_MOTOR_SPEED_SpdRefSet	速度指令値を設定します。
R_MOTOR_SPEED_SpeedCyclic	速度制御を行います。
R_MOTOR_SPEED_FrictionParameterUpdate	摩擦補償のパラメータを更新します。

8.6.8 構造体・変数情報

マネージャモジュールの構造体・変数一覧を表 8-21 に示します。マネージャモジュールは API のインスタンス確保にて、マネージャモジュール用構造体(g_st_hall_vector)を定義します。電流制御モジュールで使用する構造体・変数一覧を表 8-22 に示します。速度制御モジュールの構造体・変数一覧を表 8-23 に示します。電流制御モジュールと、速度制御モジュールは API のインスタンス確保にて、電流制御モジュール用構造体(g_st_cc)と、速度モジュール用構造体(g_st_sc)を定義します。

表 8-21 マネージャモジュール用構造体・変数一覧

構造体	変数	説明
st_hall_vector_control_t マネージャモジュール用構造体	u1_state_id_ref	d 軸電流指令値状態 0 : ゼロ指令 1 : Id 指令入力 2 : マニュアル指令入力
	u1_state_iq_ref	q 軸電流指令値状態 0 : ゼロ指令 1 : Iq 指令入力 2 : マニュアル指令入力
	u1_state_speed_ref	速度指令値状態 0 : ゼロ指令 1 : 速度指令入力 2 : マニュアル指令入力
	u1_state_open_loop	オープンループ状態 0 : 位置センサからの位置で電流制御の制御位相を計算(Closed-loop) 1 : 指令速度で電流制御の制御位相を計算(Open-loop)
	u1_direction	回転方向 0 : CW 1 : CCW
	u1_ctrl_loop_mode	制御ループモード 0 : d 軸電流制御 1 : q 軸電流制御 2 : 速度制御 3 : 位置制御
	u1_offset_adjust_mode	オフセット除去モード 0 : 電流オフセット 1 : 位置オフセット 2 : 完了
	u1_flag_offset_calc	オフセット算出フラグ
	u2_error_status	エラー状態
	u2_error_status_mask	エラービットマスク、ビットが 0 の場合、該当エラー無視されます
	u2_run_mode	モータ駆動モード 0 : オフセット除去 1 : モータ駆動
	u1_flag_charge_cap	電圧安定フラグ
	f4_time_elapsed_power_on	電源投入後の経過時間[s]
	f4_vdc_ad	母線電圧 [V]
	f4_iu_ad	u 相電流 [A]

構造体	変数	説明
	f4_iv_ad	v 相電流 [A]
	f4_iw_ad	w 相電流 [A]
	f4_overcurrent_limit	過電流制限値 [A]
	f4_oversvoltage_limit	過電圧制限値 [V]
	f4_undervoltage_limit	低電圧制限値 [V]
	f4_overspeed_limit_rad	過速度制限値 [rad/s]
	f4_user_open_loop_current	ユーザオープンループ電流指令[A]
	f4_user_torque_nm	ユーザトルク指令[Nm]
	f4_open_loop_angle_rad	オープンループ制御時の電流制御位相[rad]
	f4_rotor_angle_rad	電流制御位相[rad]
	f4_rotor_angle_offset_rad	位置オフセット[rad]
	HallSignalGet	ホール信号を取得する関数へのポインタ、 関数例： <code>void HallSignalGetFunction (uint8_t * u1_bit_u, uint8_t * u1_bit_v, uint8_t * u1_bit_w);</code>
	st_current_output	電流モジュールの出力用構造体
	st_speed_output	速度モジュールの出力用構造体
	st_sensor_output	センサモジュール用出力構造体
	st_stm	ステータマシンの構造体
	st_motor	モータパラメータ構造体
	*p_st_driver	ドライバモジュールの生成インスタンス
	*p_st_cc	電流モジュールの生成インスタンス
	*p_st_sc	速度モジュールの生成インスタンス
*p_st_sensor	センサモジュール構造体ポインタ	
st_hall_vector_cfg_t マネージャ モジュール制御 パラメータ設定用 構造体	f4_overspeed_limit_rpm	速度制限値 [r/min] (機械角)
	st_motor	モータパラメータ構造体

表 8-22 電流制御モジュール用構造体・変数一覧

構造体	変数	説明
st_current_control_t 電流制御モジュール用 構造体	u1_active	電流制御モジュールのアクティブ状態
	u1_flag_volt_err_comp_use	電圧誤差補償機能の有効/無効
	u1_state_id_ref	d 軸入力ステータス
	u1_state iq_ref	q 軸入力ステータス
	u1_flag_offset_calc	電流オフセット計算のフラグ
	u2_offset_calc_time	電流オフセット調整時の測定時間設定
	u2_offset_calc_wait	電流オフセット調整タイミングカウンタ
	u2_crnt_offset_cnt	電流オフセット調整時の測定回数
	f4_ctrl_period	電流制御周期(期間)[s]
	f4_refu	u 相指令電圧[V]
	f4_refv	v 相指令電圧[V]
	f4_refw	w 相指令電圧[V]
	f4_vd_ref	d 軸電圧指令値[V]
	f4_vq_ref	q 軸電圧指令値[V]
	f4_id_ref	d 軸電流指令値[A]
	f4_iq_ref	q 軸電流指令値[A]
	f4_id_ad	d 軸電流値[A]
	f4_iq_ad	q 軸電流値[A]
	f4_lim iq	q 軸電流制限値[A]
	f4_offset_iu	u 相オフセット電流値[A]
	f4_offset_iw	w 相オフセット電流値[A]
	f4_sum_iu_ad	u 相電流合計値[A]
	f4_sum_iw_ad	w 相電流合計値[A]
	f4_vdc_ad	母線電圧値[V]
	f4_iu_ad	u 相電流値[A]
	f4_iv_ad	v 相電流値[A]
	f4_iw_ad	w 相電流値[A]
	f4_modu	u 相デューティ比
	f4_modv	v 相デューティ比
	f4_modw	w 相デューティ比
	f4_speed_rad	速度[rad/s]
	f4_rotor_angle_input_rad	ロータ角度[rad]
f4_id_ref_manual	d 軸用電流ステータス: d 軸指令固定モード時の d 軸電流指令値[A]	
f4_iq_ref_manual	q 軸用電流ステータス: q 軸指令固定モード時の q 軸電流指令値[A]	

構造体	変数	説明
	f4_ref_id_ctrl	d 軸電流指令値 [A]
	f4_ref_iq_ctrl	q 軸電流指令値[A]
	f4_current_rate_limit	電流の上昇速度制限値
	f4_va_max	dq 軸上の最大電圧 [V]
	st_mod	変調用構造体
	st_volt_comp	電圧誤差補償
	st_pi_id	d 軸の PI 制御用構造体
	st_pi_iq	q 軸の PI 制御用構造体
	st_rotor_angle	ロータ情報の構造体
	st_rotor_angle_phasecomp	ロータ情報の構造体(進み補償)
	st_motor	モータパラメータの構造体
st_current_cfg_t 電流制御モジュール制御パラメータ設定用構造体	u2_offset_calc_time	オフセット計算時間設定
	f4_ctrl_period	制御周期[s]
	f4_current_omega_hz	電流制御系固有周波数[Hz]
	f4_current_zeta	電流制御系減衰係数
	u1_flag_volt_err_comp_use	電圧誤差補償有効/無効
	st_motor	モータパラメータの構造体
st_current_output_t 電流制御モジュール出力用構造体	u1_flag_offset_calc	電流オフセットフラグ
	f4_modu	u 相デューティ比
	f4_modv	v 相デューティ比
	f4_modw	w 相デューティ比
	f4_neutral_duty	オフセット測定時のデューティ比
	f4_va_max	dq 軸上の最大電圧[V]
st_current_input_t 電流制御モジュール入力用構造体	u1_state_id_ref	d 軸ステータス
	u1_state_iq_ref	q 軸ステータス
	f4_rotor_angle_rad	ロータ角度[rad]
	f4_iu_ad	U 相電流値[A]
	f4_iv_ad	V 相電流値[A]
	f4_iw_ad	W 相電流値[A]
	f4_vdc_ad	母線電圧値[V]
	f4_speed_rad	速度[rad/s]
	f4_id_ref	d 軸電流指令値[A]
	f4_iq_ref	q 軸電流指令値[A]

構造体	変数	説明
	f4_id_ref_manual	d 軸用電流ステータス: d 軸指令固定モード時の d 軸電流指令値[A]
	f4_iq_ref_manual	q 軸用電流ステータス: q 軸指令固定モード時の q 軸電流指令値[A]

表 8-23 速度制御モジュール用構造体・変数一覧 1

構造体	変数	説明
st_speed_control_t	u1_active	モジュールの有効/無効選択
速度モジュール用構造体	u1_state_speed_ref	速度指令値を決定する状態管理。本節のマクロに記載する状態を管理する。
	u1_flag_fluxwkn_use	弱め磁束制御の使用有無フラグ
	f4_speed_ctrl_period	速度ループの周期 [s]
	f4_ref_speed_rad_ctrl	制御用の速度指令値 [rad/s]
	f4_ref_speed_rad	速度制御時の速度モジュール出力の速度指令値 [rad/s]
	f4_ref_speed_rad_manual	速度制御時のユーザの速度指令値設定値 [rad/s]
	f4_speed_rad_ctrl	速度制御モジュール内で演算する速度 [rad/s]
	f4_speed_rad	入力された速度 [rad/s]
	f4_max_speed_rad	最大速度 [rad/s]
	f4_speed_rate_limit_rad	速度の変化量の制限値 [rad/s]
	f4_id_ref_output	d 軸電流指令値 [A]
	f4_iq_ref_output	q 軸電流指令値 [A]
	f4_va_max	dq 軸上の最大電圧 [V]
	f4_id_ad	d 軸電流値 [A]
	f4_iq_ad	q 軸電流値 [A]
	st_motor	モータパラメータ構造体
	st_pi_speed	PI 制御用構造体
	st_fluxwkn	弱め磁束制御用構造体
	st_friction_comp	摩擦補償用構造体

表 8-24 構造体・変数一覧 2

構造体	変数	説明
st_speed_cfg_t	st_motor	モータパラメータ構造体
速度モジュール制御パラメータ設定用構造体	u1_flag_fluxwkn_use	弱め磁束制御の使用有無フラグ
	f4_max_speed_rpm	最大速度 [rpm]
	f4_speed_ctrl_period	速度制御の周期 [s]
	f4_speed_rate_limit_rpm	速度の変化量の制限値 [rpm]
	f4_speed_omega_hz	速度制御系固有周波数 [Hz]
	f4_speed_zeta	速度制御系減衰係数
st_speed_input_t	u1_state_speed_ref	速度指令ステータス
速度モジュール入力用構造体	f4_ref_speed_rad	速度指令値 [rad/s]
	f4_speed_rad	入力する速度 [rad/s]
	f4_va_max	dq 軸における最大電圧 [V]
st_speed_output_t	f4_id_ref	d 軸電流指令値 [A]
速度モジュール出力用構造体	f4_iq_ref	q 軸電流指令値 [A]
	f4_ref_speed_rad_ctrl	PI 制御に使用する速度 [rad/s]

8.6.9 マクロ定義

マネージャモジュールのマクロ一覧を表 8-25 に示します。

表 8-25 マクロ一覧

ファイル名	マクロ名	定義値	備考
r_motor_hall_vector_api.h	MOTOR_LOOP_CURRENT_ID	0	磁束制御モード。
	MOTOR_LOOP_CURRENT_IQ	1	トルク制御モード。
	MOTOR_LOOP_SPEED	2	速度制御モード。
	MOTOR_LOOP_POSITION	3	位置制御モード。本サンプルプログラムでは使用できません。
	MOTOR_ERROR_NONE	(0x0000)	エラーステータス。エラーなし状態。
	MOTOR_ERROR_OVER_CURRENT_HW	(0x0001)	エラーステータス HW 過電流エラー状態。
	MOTOR_ERROR_OVER_VOLTAGE	(0x0002)	エラーステータス。過電圧エラー状態。
	MOTOR_ERROR_OVER_SPEED	(0x0004)	エラーステータス。過速度エラー状態。
	MOTOR_ERROR_LOW_VOLTAGE	(0x0080)	エラーステータス。低電圧エラー状態。
	MOTOR_ERROR_OVER_CURRENT_SW	(0x0100)	エラーステータス。SW の過電流エラー状態。
	MOTOR_ERROR_OVER_TEMPERATURE	(0x0200)	エラーステータス。インバータ過熱エラー状態。
	MOTOR_ERROR_UNKNOWN	(0xffff)	エラーステータス。エラーコード不明のエラー状態。
r_motor_hall_vector_manager.h	MOTOR_HALL_VECTOR_MODE_OFFSET_ADJUST	(0x00)	初期化を行う動作モード。
	MOTOR_HALL_VECTOR_MODE_DRIVE	(0x01)	モータ駆動状態の動作モード。
	MOTOR_HALL_VECTOR_OFAJ_CURRENT_OFFSET	(0x00)	The operation mode for current offset
	MOTOR_HALL_VECTOR_OFAJ_FIN	(0x01)	Offset adjustment completed
r_motor_hall_vector_api.h	MOTOR_CTRL_TYPE_POS	0	制御方式切り替え用マクロ。位置制御モード。本サンプルプログラムでは使用できません。
	MOTOR_CTRL_TYPE_SPEED	1	制御方式切り替え用マクロ。速度制御モード。
	MOTOR_CTRL_TYPE_TORQUE	2	制御方式切り替え用マクロ。トルク制御モード。
	MOTOR_CTRL_TYPE_VOLTAGE	3	未使用
	MOTOR_CTRL_TYPE_CURRENT	4	制御方式切り替え用マクロ。オープンループ制御モード。
	MOTOR_CTRL_TYPE_ADJ_CRNT_OFFSET	5	制御方式切り替え用マクロ。電流オフセット調整モード。

8.7 ドライバモジュール

ドライバモジュールは、サンプルプログラムのミドルウェアに相当するマネージャモジュールと MCU のペリフェラルにアクセスするためのスマート・コンフィグレータを接続するインターフェースの役割を持つモジュールです。ドライバモジュールを適切に設定することで、MCU の機能割り当てや使用するボード仕様の差分をモータモジュールの変更無く使用することが可能になります。

8.7.1 機能

ドライバモジュールの機能一覧を表 8-26 に示します。

表 8-26 ドライバモジュールの機能一覧

機能	説明
A/D 変換値の取得	スマート・コンフィグレータ関数経由で相電流やインバータボードの母線電圧など AD 値を取得します。
PWM の duty 設定	スマート・コンフィグレータ関数経由で UVW 相へ出力する PWM Duty 値を設定します。
PWM の開始、停止	スマート・コンフィグレータ関数経由で PWM 出力の開始、停止を制御します。

8.7.2 モジュール構成図

ドライバモジュールのモジュール構成図を図 8-6 に示します。

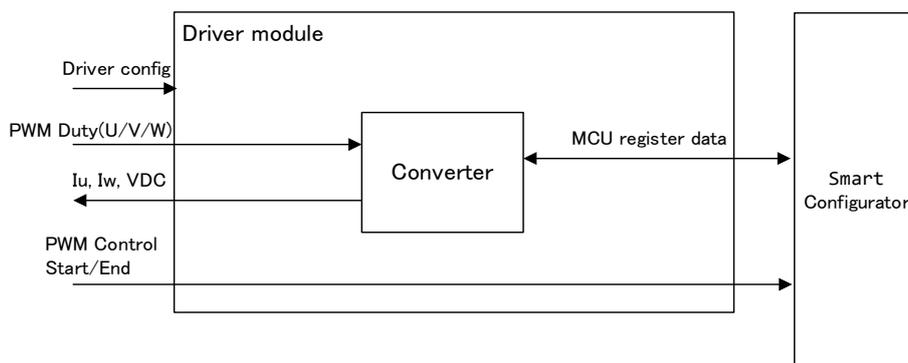


図 8-6 ドライバモジュール構成図

8.7.3 API

ドライバモジュールの API 一覧表と各 API の説明を表 8-27 に示します。

表 8-27 ドライバモジュールの API 一覧

API	説明
R_MOTOR_DRIVER_Open	ドライバモジュールのインスタンスを生成します。
R_MOTOR_DRIVER_Close	モジュールをリセット状態にします。
R_MOTOR_DRIVER_ParameterUpdate	モジュール内部で使用する変数情報を入力します。
R_MOTOR_DRIVER_BldcAnalogGet	AD 変換結果を取得します。
R_MOTOR_DRIVER_BldcDutySet	PWM Duty の設定を行います。
R_MOTOR_DRIVER_PWMControlStop	PWM 制御を停止します。
R_MOTOR_DRIVER_PWMControlStart	PWM 制御を開始します。

8.7.4 コンフィグレーション情報

ドライバモジュールのコンフィグレーション情報一覧を表 8-28 コンフィグレーション情報一覧に示します。使用する機能や各種パラメータを設定してください。

表 8-28 コンフィグレーション情報一覧

ファイル名	マクロ名	設定	説明
r_motor_module_cfg.h	DRIVER_CFG_FUNC_PWM_OUTPUT_START	R_Config_xxx_StartTimerCtrl (スマート・コンフィグレータ関数) *1 *2	PWM 出力許可関数設定
	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	R_Config_xxx_StopTimerCtrl (スマート・コンフィグレータ関数) *1 *2	PWM 出力禁止関数設定
	DRIVER_CFG_FUNC_ADC_DATA_GET	R_Config_xxx_AdcGetConvVal (スマート・コンフィグレータ関数) *1 *2	AD 変換結果取得関数設定
	DRIVER_CFG_FUNC_DUTY_SET	R_Config_xxx_UpdDuty (スマート・コンフィグレータ関数) *1 *2	Duty Cycle 設定関数設定
r_motor_inverter_cfg.h	INVERTER_CFG_ADC_REF_VOLTAGE	5.0f	AD 変換基準電圧設定
r_motor_module_cfg.h	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	AD オフセット値設定

- 【注】
1. 設定値に記載した関数については、10 スマート・コンフィグレータ設定を参照してください。
 2. スマート・コンフィグレータのモータコンポーネントを使用する場合は、“xxx”は“MOTOR”と設定しています。モータコンポーネントを使用しない場合は、PWM に使用するモジュール名が入ります。

8.7.5 構造体・変数情報

ドライバモジュールで使用する構造体一覧を表 8-29 に示します。ドライバモジュールは API のインスタンス確保にて、ドライバモジュール用構造体(g_st_driver)を定義します。

表 8-29 構造体・変数一覧

構造体	変数	説明
st_motor_driver_t ドライバモジュール用構造体	*ADCCDataGet	スマート・コンフィグレータ関数へのポインタ (AD 変換結果取得関数を設定)
	*BLDCDutySet	スマート・コンフィグレータ関数へのポインタ (PWM 出力許可関数を設定)
	*PWMOutputStop	スマート・コンフィグレータ関数へのポインタ (PWM 出力禁止関数を設定)
	*PWMOutputStart	スマート・コンフィグレータ関数へのポインタ (Duty Cycle 設定関数を設定)
	f4_ad_crnt_per_digit	電流 AD 変換用スケール
	f4_ad_vdc_per_digit	電圧 AD 変換用スケール
	f4_pwm_period_cnt	PWM カウンター周期のカウンタ数(Duty 設定用情報)
	f4_pwm_dead_time_cnt	デッドタイムのカウンタ数(Duty 設定用情報)
st_motor_driver_cfg_t ドライバモジュール制御 パラメータ設定用構造体	*ADCCDataGet	スマート・コンフィグレータ関数へのポインタ
	*BLDCDutySet	スマート・コンフィグレータ関数へのポインタ
	*PWMOutputStop	スマート・コンフィグレータ関数へのポインタ
	*PWMOutputStart	スマート・コンフィグレータ関数へのポインタ
	f4_shunt_ohm	シャント抵抗値[ohm] (f4_ad_crnt_per_digit 計算用)
	f4_volt_gain	電圧変換ゲイン係数(f4_ad_vdc_per_digit 計算用)
	f4_crnt_amp_gain	電流変換ゲイン係数(f4_ad_crnt_per_digit 計算用)
	f4_pwm_period_cnt	PWM カウンター周期のカウンタ数(Duty 設定用情報)
f4_pwm_dead_time_cnt	デッドタイムのカウンタ数(Duty 設定用情報)	

8.7.6 パラメータ調整・設定

ドライバモジュールでは、制御パラメータ設定(R_MOTOR_DRIVER_ParameterUpdate)から入力されたパラメータを使用して、モータモジュールとスマート・コンフィグレータとの関連付け、データ変換を行います。ドライバモジュール制御パラメータ設定用構造体(st_speed_config_t)を使って入力します。サンプルプログラムでは、コンフィグレーションとして定義されているものをパラメータ設定値として使用しています。設定内容を表 8-30 に示します。

表 8-30 サンプルプログラム設定例

変数名	マクロ名	ファイル名
*ADCDataGet	DRIVER_CFG_FUNC_ADC_DATA_GET	r_motor_module_cfg.h
*BLDCDutySet	DRIVER_CFG_FUNC_DUTY_SET	
*PWMOutputStop	DRIVER_CFG_FUNC_PWM_OUTPUT_START	
*PWMOutputStart	DRIVER_CFG_FUNC_PWM_OUTPUT_STOP	
f4_shunt_ohm	INVERTER_CFG_SHUNT_RESIST	r_motor_inverter_cfg.h
f4_volt_gain	INVERTER_CFG_VOLTAGE_GAIN	
f4_crnt_amp_gain	INVERTER_CFG_CURRENT_AMP_GAIN	
f4_pwm_period_cnt	MOTOR_COMMON_CARRIER_SET_BASE	r_motor_module_cfg.h
f4_pwm_dead_time_cnt	MOTOR_COMMON_DEADTIME_SET	

9. パラメータの設定

9.1 概要

本サンプルプログラムでは、パラメータは以下のヘッダファイル内でマクロ定義されています。マクロ定義されたパラメータは、起動時の初期化ルーチンで、各機能モジュールで管理される変数・構造体に設定され、各々の処理に使用されます。

一部のパラメータは、RMW 等から動的に変更が可能です。変更を行った場合には、パラメータアップデートの関数をコールし、反映させる必要があります。詳細は、各機能モジュールの説明を参照してください。

表 9-1 パラメータ設定ファイルの一覧

マクロ名	説明
r_motor_module_cfg.h	モータ制御に関するパラメータの初期値を定義しています。
r_motor_inverter_cfg.h	インバータに関するパラメータの初期値を定義しています。
r_motor_targetmotor_cfg.h	モータに関するパラメータの初期値を定義しています。

9.2 MCU 関連パラメータ

MCU の周辺機能に関連するパラメータ一覧を表 9-2 に示します。マイコンのペリフェラル設定を変更した場合、これらのパラメータで、該当する箇所は変更を行う必要があります。

表 9-2 MCU 関連パラメータの一覧

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	MOTOR_MCU_CFG_PWM_TIMER_FREQ	120.0f	PWM のタイマ周波数 [MHz]
	MOTOR_MCU_CFG_CARRIER_FREQ	20.0f	キャリア周波数 [kHz]
	MOTOR_MCU_CFG_INTR_DECOMINATION	0	キャリア割り込みの間引き回数
	MOTOR_MCU_CFG_AD_FREQ	60.0f	ADC の動作周波数 [MHz]
	MOTOR_MCU_CFG_AD_SAMPLING_CYCLE	2.0f * (7.25f + 63.0f)	ADC のサンプリング周期 [cycle]
	MOTOR_MCU_CFG_AD12BIT_DATA	4095.0f	ADC の分解能
	MOTOR_MCU_CFG_ADC_OFFSET	0x7FF	ADC のオフセット値

9.3 制御機能の設定パラメータの一覧

モータ制御プログラムに備わっている機能の有効無効を設定するパラメータを、表 9-3、表 9-4、表 9-5 に示します。モータ制御の内部で使用される、モータ定数や設定に関する項目は、後述します。

表 9-3 動作パラメータの一覧(全般)

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	MOTOR_TYPE_BLDC	MOTOR_TY E_BLDC	デフォルトのまま使用ください。
	MOTOR_COMMON_CFG_LOOP_MODE	MOTOR_LOOP_SPEED	デフォルトのまま使用ください。
	MOTOR_COMMON_CFG_OVERCURRENT_MARGIN_MULT	2.0f	過電流のリミット係数
	MOTOR_COMMON_CFG_IAX_CALC_MULT	MTR_SQRT_3	過電流リミット値計算用係数。 √3 を設定してください。
	MOTOR_COMMON_CFG_AUTO_OFFSET_ADJUST_DELAY	(0.5f)	電源投入後、自動電流オフセット調整を実施するまでの遅延時間[s]
	MOTOR_MCU_CFG_TFU_OPTIMIZE	MTR_ENABLE	TFU 専用関数処理の設定 CC-RX の TFU コンパイラオプション(-tfu)に応じて自動的に設定される

表 9-4 動作パラメータの一覧(速度制御関連)

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	SPEED_CFG_FLUX_WEAKENING	MTR_ENABLE	弱め磁束制御の有効無効設定。 有効 : MTR_ENABLE 無効 : MTR_DISABLE
	SPEED_CFG_CTRL_PERIOD	0.0005f	速度制御周期[sec]の設定。 0.5ms とするため、0.0005f を設定してください。

表 9-5 動作パラメータの一覧(電流制御関連)

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	CURRENT_CFG_VOLT_ERR_COMP	MTR_ENABLE	電圧誤差補償機能の有効・無効設定。MTR_ENABLE を設定してください。
	CURRENT_CFG_MODULATION_METHOD	MOD_METHOD_SVPWM	9.5 を参照してください。 通常、MOD_METHOD_SVPWM を設定してください。
	CURRENT_CFG_OFFSET_CALC_TIME	512.0f	電流オフセットの測定時間設定。
	CURRENT_CFG_OFFSET_CALC_WAIT	4000	電流オフセットの検出タイミング

9.4 保護関連パラメータ

モータを運転する際に、安全性を担保するための保護機能のパラメータを以下に示します。

表 9-6 モータパラメータ、インバータパラメータ設定

ファイル名	マクロ名	設定値	説明
r_motor_inverter_cfg.h	INVERTER_CFG_CURRENT_LIMIT	21.4f	インバータボードの過電流の制限値 [A]
	INVERTER_CFG_OVERVOLTAGE_LIMIT	60.0f	過電圧制限 [V]
	INVERTER_CFG_UNDERVOLTAGE_LIMIT	8.0f	低電圧制限 [V]

INVERTER_CFG_CURRENT_LIMIT

インバータが出力可能な最大の電流値から、安全マージンをとった電流値を設定します。

INVERTER_CFG_OVERVOLTAGE_LIMIT

過電圧保護が動作する電圧を設定します。インバータ母線電圧が、設定した電圧超となると、エラーとなり、モータの動作が停止します。ご使用される電源環境に合わせて設定してください。

INVERTER_CFG_UNDERVOLTAGE_LIMIT

低電圧保護が動作する電圧を設定します。インバータ母線電圧が、設定した電圧未滿となると、エラーとなり、モータの動作が停止します。ご使用される電源環境に合わせて設定してください。

9.5 PWM キャリア周波数の変更

PWM キャリア周波数は、スマート・コンフィグレータによる設定と、r_motor_module_cfg.h で定義されている MOTOR_MCU_CFG_CARRIER_FREQ の定数で設定されています。PWM キャリア周波数を変更した場合、表 9-7 に示す変更箇所を修正してください。PWM キャリア周波数の設定値に合わせて、パラメータの調整が必要となるパラメータがあります。

本サンプルプログラムのデフォルト PWM キャリア周波数は、20kHz です。

表 9-7 PWM キャリア周波数を変更した場合に変更を行う箇所

項目	変更箇所
デッドタイム値	9.7 インバータパラメータを参照
キャリア周波数	<ul style="list-style-type: none"> スマート・コンフィグレータのモータ設定で PWM キャリア周波数を設定 9.2 に記載の MOTOR_MCU_CFG_CARRIER_FREQ
モータ制御関連	電流調節器のパラメータ ホールセンサ制御のパラメータ

9.6 パルス変調方法の設定

本サンプルプログラムでは、パルス幅変調駆動方式を2種類から設定することができます。デフォルトは空間ベクトルPWM(MOD_METHOD_SVPWM)となります。変調機能のコンフィグ情報一覧を表9-8に示します。

パルス幅変調駆動方式を、正弦波PWMに変更した場合、電圧利用率が86%に制約され、モータに適切な電圧が出力できず、所望の電圧を得るにはインバータ母線電圧を高く設定する必要があります。空間ベクトルPWMを使用した場合、電圧利用率はインバータ母線電圧に対して100%利用できます。

表 9-8 コンフィグレーション情報一覧

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	CURRENT_CFG_MODULATION_METHOD	(MOD_METHOD_OD_SVPWM)	パルス幅変調駆動方式

表 9-9 パルス幅変調駆動方式の設定項目

パルス幅変調駆動方式の設定項目	値	パルス幅変調駆動方式
MOD_METHOD_SPWM	0	正弦波PWM
MOD_METHOD_SVPWM	1	空間ベクトルPWM

パルス変調には、以下の設定項目があります。通常は、デフォルト値のままご使用ください。

表 9-10 マクロ一覧

ファイル名	マクロ名	設定値	説明
r_motor_current_modulation.h	MOD_DEFAULT_MAX_DUTY	1.0f	最大PWM デューティ比。通常は1.0fのままとしてください。
	MOD_VDC_TO_VAMAX_MULT	0.6124f	インバータ母線電圧で出力可能な最大電圧を得るための変換係数
	MOD_SVPWM_MULT	1.155f	空間ベクトルPWMを使用した場合のみ。空間ベクトルPWM係数

9.7 インバータパラメータ

9.7.1 概要

サンプルプログラムを使用する際に、インバータの情報を正しく設定する必要があります。サンプルプログラムで設定されているインバータパラメータを表 9-11 に示します。

表 9-11 インバータパラメータ設定

ファイル名	マクロ名	設定値	説明
r_motor_inverter_cfg.h	INVERTER_CFG_SHUNT_RESIST	0.010f	シャント抵抗値 [ohm]
	INVERTER_CFG_DEADTIME	2.0f	デッドタイム [us]
	INVERTER_CFG_VOLTAGE_GAIN	22.2766f	電圧検出用係数
	INVERTER_CFG_CURRENT_AMP_GAIN	20.0f	電流検出用アンプのゲイン
	INVERTER_CFG_INPUT_V	24.0f	入力電圧 [V]
	INVERTER_CFG_ADC_REF_VOLTAGE	5.0f	MCU のアナログ電源電圧 [V]
	INVERTER_CFG_COMP_V0	0.564f	電圧誤差補償用係数 [V]
	INVERTER_CFG_COMP_V1	0.782f	電圧誤差補償用係数 [V]
	INVERTER_CFG_COMP_V2	0.937f	電圧誤差補償用係数 [V]
	INVERTER_CFG_COMP_V3	1.027f	電圧誤差補償用係数 [V]
	INVERTER_CFG_COMP_V4	1.058f	電圧誤差補償用係数 [V]
	INVERTER_CFG_COMP_I0	0.022f	電圧誤差補償用係数 [A]
	INVERTER_CFG_COMP_I1	0.038f	電圧誤差補償用係数 [A]
	INVERTER_CFG_COMP_I2	0.088f	電圧誤差補償用係数 [A]
	INVERTER_CFG_COMP_I3	0.248f	電圧誤差補償用係数 [A]
INVERTER_CFG_COMP_I4	0.865f	電圧誤差補償用係数 [A]	

INVERTER_CFG_DEADTIME

インバータの仕様書・設計書に記載された、デッドタイム時間を us(マイクロ秒)単位で指定してください。インバータキット MCK-RX26T では、2.0us が指定されています。

INVERTER_CFG_INPUT_V

DC 電圧値 24V をデフォルトとしています。

INVERTER_CFG_ADC_REF_VOLTAGE

MCU のアナログ電圧を指定します。RX26T CPU ボードは 5.0V となります。

INVERTER_CFG_COMP_Vx, INVERTER_CFG_COMP_Ix

9.7.4 を参照してください。

9.7.2 電流検出ゲイン

MCI-LV-1 インバータでは、表 9-12 に示すように、電流の大きさによって、ADC に入力される電圧値が規定されています。

本サンプルプログラムで、電流の検出ゲインを設定するには、`INVERTER_CFG_CURRENT_AMP_GAIN` と、`INVERTER_CFG_SHUNT_RESIST` を使用します。このとき、`INVERTER_CFG_ADC_REF_VOLTAGE` は、5V のままとします。

`INVERTER_CFG_CURRENT_AMP_GAIN`

ADC で入力する電圧 1V あたり、何 A に相当するかを求める係数を設定します。

`INVERTER_CFG_SHUNT_RESIST`

シャント抵抗で使用している抵抗値を設定します。シャント抵抗を使わずにホール CT を使用する場合には、1.0 を指定してください。

`INVERTER_CFG_ADC_REF_VOLTAGE`

`INVERTER_CFG_SHUNT_RESIST` の逆数で割った値に 1V あたりに換算した電流値で割ります。0-5V で±12.5A(Peak to Peak で 25A)の換算となっている MCI-LV-1 の仕様では、1V あたり 5A となります。シャント抵抗値 0.01Ω とすると、その逆数は 100 となります。計算すると、 $(1/100) \times (1/20) = 5$ となります。`INVERTER_CFG_ADC_REF_VOLTAGE` には、5 を設定してください。

`INVERTER_CFG_ADC_REF_VOLTAGE` を求めるために一般化した式を以下に示します。

$$INVERTER_CFG_ADC_REF_VOLTAGE = \frac{1}{INVERTER_CFG_SHUNT_RESIST[\Omega]} \times \frac{1}{1V \text{ あたりの電流値}[A/V]}$$

表 9-12 MCI-LV-1 の電流信号仕様

3 相出力電流値	ADC 入力電圧値	ADC 変換値
+12.5A	5V	4095
0A	2.5V	2048
-12.5A	0V	0

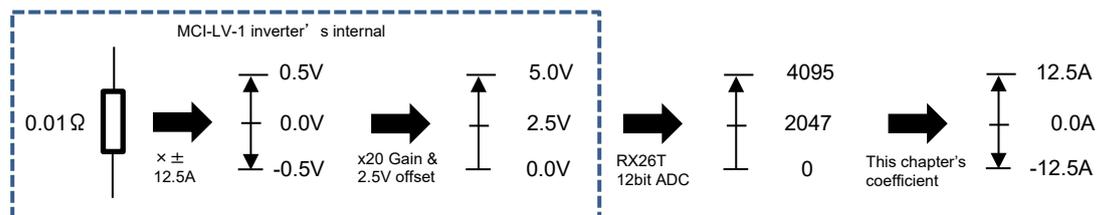


図 9-1 電流検出の計算の流れ

9.7.3 電圧検出ゲイン

電圧検出ゲインは、INVERTER_CFG_VOLTAGE_GAIN で設定します。

ADC で入力する電圧 1V あたり、インバータ母線電圧で何 V に相当するかを求める係数を設定します。ADC に入力される電圧 5V で 111.383V に相当する場合は、 $111.383/5=22.2766$ となるため、INVERTER_CFG_VOLTAGE_GAIN には、22.2766 を設定します。

$$\text{INVERTER_CFG_VOLTAGE_GAIN} = \frac{\text{インバータ母線電圧の基準}}{\text{基準となるときのADC入力電圧}} = \frac{111.383}{5} = 22.2766$$

表 9-13 MCI-LV-1 のインバータ母線電圧信号仕様

インバータ母線電圧値	ADC 入力電圧値	ADC 変換値
0V	0V	0
111.383V	5V	4095

9.7.4 電圧誤差補償パラメータ

電圧誤差補償の機能の使用及び設定方法について説明します。以下の 3 点の設定が必要となります。

① デッドタイム値の選定

インバータに使用されているパワー半導体の特性により、デッドタイム値は決定されます。本サンプルソフトでは、キットに同梱するインバータボード(RTK0EM0000B12020BJ)の特性に則して $2.0\mu\text{s}$ に設定しています。スマート・コンフィグレータのモータ設定に入力場所が用意されていますので、選定したデッドタイム値を反映させてください。

② 電圧誤差機能有効フラグの設定

電流制御モジュールの制御パラメータ設定(R_MOTOR_CURRENT_ParameterUpdate)呼び出し時に、電圧誤差補償機能の有効/無効使用有無フラグ(u1_flag_volt_err_comp_use)を MTR_FLG_SET に設定することで機能が有効になります。無効にする場合は、上記フラグを MTR_FLG_CLR に設定してください。

③ 電圧補償テーブルの設定

実機のインバータで、電流を流したスイッチング試験を行うか、デッドタイムとキャリア周期の関係から求まるデッドタイム分の電圧誤差値を用いて、電圧補償テーブルを作成します。スイッチング試験で得られた電流と電圧の関係が求めると、より効果的な電圧補償テーブルに設定可能な値が得られます。

また、補償電圧値のリミットは以下の式で計算できます。

$$\text{補償電圧リミット} = (\text{キャリア周期} [\text{kHz}] \times \text{デッドタイム時間} [\mu\text{s}] \div 1000) \times \text{母線電圧値}$$

ゼロクロス付近の傾斜は、 I_u と V_u (I_v-V_v , I_w-V_w) の関係が実験的に得られない場合は、上記の式と、主回路の特性を考慮して机上計算し、テーブルを求める必要があります。

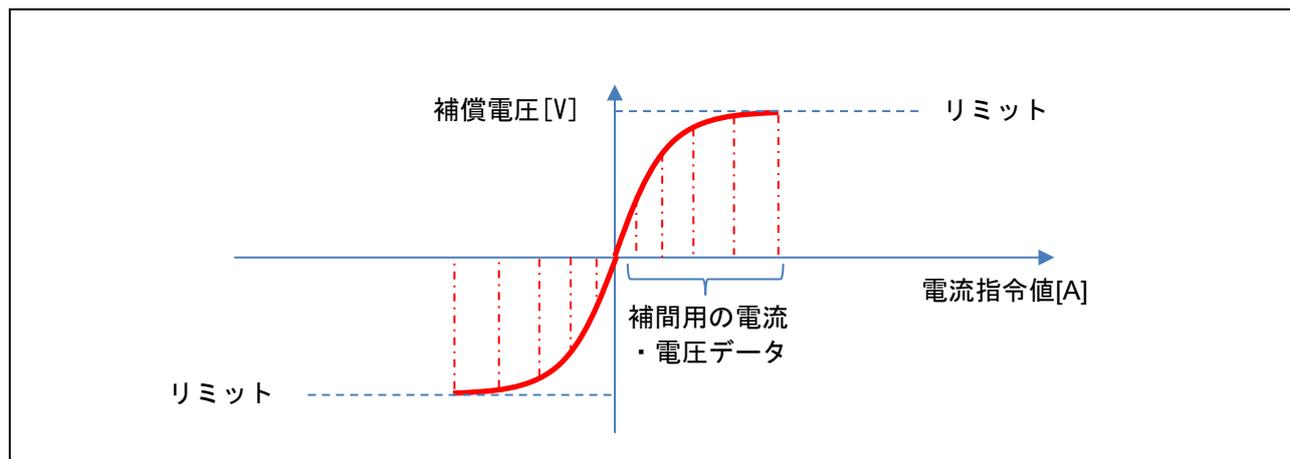


図 9-2 補償電圧値とリミット、電流指令値の関係

表 9-14 キャリア周期と補間用の電流・電圧データ

キャリア周期		20kHz
	lu	ΔV_u
0	0.00	0.00
1	0.022	0.564
2	0.038	0.782
3	0.088	0.937
4	0.248	1.027
5	0.865	1.058

9.8 モータパラメータ

モータの製造メーカーから、モータパラメータの情報が得られない場合、LCR メータを用いて R,Ld,Lq のモータパラメータを簡易的に得ることができます。また、オシロスコープを用いることで、簡易的に誘起電圧を得ることができます。ここで説明した方法は、磁気飽和などを考慮せず、またモータを速やかに回す事を考慮した、簡易的な方法であり、個体差や測定誤差を含んでいます。このため、実際の製品開発でパラメータを使用する際には、精度を担保した測定設備を用いて、測定を行ってください。

LCR メータは、定期的に校正をされたもので、電源を起動して 30 分以上経過させたウォーミングアップ完了状態で測定してください。また、4 端子法を用いて、プローブの誤差を低減するため、オープン補正とショート補正をあらかじめ行ってください。詳細は、LCR メータの取扱説明書を参照してください。

サンプルプログラムを使用する際に、インバータの情報と使用するモータの情報を正しく設定する必要があります。サンプルプログラムの設定値を表 9-15 に示します。

表 9-15 モータパラメータ設定

ファイル名	マクロ名	設定値	説明
r_motor_targetmotor_cfg.h	MOTOR_CFG_POLE_PAIRS	4	極対数
	MOTOR_CFG_MAGNETIC_FLUX	0.01119f	磁束 [wb]
	MOTOR_CFG_RESISTANCE	1.3f	抵抗 [ohm]
	MOTOR_CFG_D_INDUCTANCE	0.0013f	d 軸のインダクタンス [H]
	MOTOR_CFG_Q_INDUCTANCE	0.0013f	q 軸のインダクタンス [H]
	MOTOR_CFG_ROTOR_INERTIA	0.000003666f	ロータのイナーシャ [kg m ²]
	MOTOR_CFG_NOMINAL_CURRENT_RMS	1.67f	定格電流 [A]
	MOTOR_CFG_MAX_SPEED_RPM	2400.0f	最大速度 [r/min]

MOTOR_CFG_POLE_PAIRS

PM モータの極対数を設定します。極対数は、極数を 1/2 した値となります。PM モータの仕様書を参照してください。

MOTOR_CFG_RESISTANCE

LCR メータで測定する際の配線は、モータの三相出力線 U,V,W のうち、2つを選び、プローブをつなげてください。抵抗値を求める場合は、直流抵抗(DCR)のモードを用いて、測定します。得られた抵抗値は、2相分の合成抵抗となっていますので、1/2 をすることで、1相分のモータの抵抗値を得ることができます。得られた抵抗 R は、r_motor_targetmotor_cfg.h の MOTOR_CFG_RESISTANCE に設定してください。単位はΩとなります。

MOTOR_CFG_D_INDUCTANCE, MOTOR_CFG_Q_INDUCTANCE

LCR メータで測定する際の配線は、モータの三相出力線 U,V,W のうち、2つを選び、プローブをつなげてください。計測モードは、直列等価回路モード(Ls)で行います。詳細な測定方法は、LCR メータの取扱説明書を参照ください。

軸をゆっくり回し、表示されるインダクタンスの最大値と最小値をメモします。このとき、最大値の 1/2 が、Lq となり、最小値の 1/2 の値が Ld となります。

得られた Ld 及び Lq は、r_motor_targetmotor_cfg.h の MOTOR_CFG_D_INDUCTANCE, MOTOR_CFG_Q_INDUCTANCE に設定してください。単位は H(ヘンリー)です。

MOTOR_CFG_ROTOR_INERTIA

モータの回転子・軸のイナーシャ（慣性モーメント）を設定します。単位は、kg m²です。通常、モータに添付された資料に記述があります。負荷を取り付ける場合には、負荷側のイナーシャも加えて設定してください。

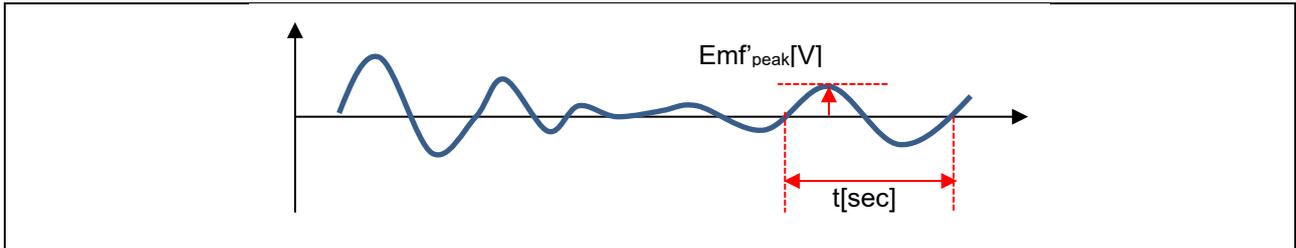
MOTOR_CFG_NOMINAL_CURRENT_RMS

モータの定格電流(実効値)を設定してください。単位はアンペアです。モータの銘板または添付資料に記載されています。

MOTOR_CFG_MAGNETIC_FLUX

モータの三相出力線 U,V,W のうち、2つを選び、オシロスコープにつなげてください。例えば、U 相と V 相を、オシロスコープのプローブを当てて、電圧を測れるようにします。モータの軸の先には、定格速度で回転できるモータを繋げて定格速度で回転させると、U-V 相の線間電圧値が得られます。線間電圧値を $\sqrt{3}$ で割ることで、相あたりの誘起電圧のピーク値が得られます。鎖交磁束数 Ψ は、誘起電圧 $=\omega\Psi$ の式から求められますから、定格速度を電気角速度の周波数 f [Hz]に換算し、 $\omega=2\pi f$ に置き換え、誘起電圧 $=2\pi f\Psi$ となり、式を変形し、値を代入することで鎖交磁束 Ψ [Wb]が得られます。

軸の先にモータを取り付けできない等の場合には、精度は保証されず、試運転目的のみでの利用となりますが、手で素早く回転させ、電圧波形を取得して簡易的に求める手法も使用できます。手で回した際に、以下のようなイメージで電圧波形が得られますが、このとき、正弦波で一定速に近い周期を選び、電圧のピークと周期を求めます。



本アルゴリズムではピーク値を実効値に換算する必要があるため $\sqrt{2}$ で割って実効値 Emf'_{rms} を得ます。

$$Emf'_{rms}[V] = Emf'_{peak}[V] \times \frac{1}{\sqrt{2}}$$

得られた時間 t [sec]を Hz に直すため、 $f=1/t$ の式にあてはめます。得られた f [Hz]と、この PM モータの定格速度から得られる電気角周波数[Hz]の比を求め、同時に得られた電圧 Emf_{rms} [V]に比を掛け算します。

$$Emf[V] = Emf'_{rms}[V] \times \frac{\text{電気角周波数[Hz]}}{f'[Hz]}$$

この結果、この PM モータの定格速度で回転した時に発生する、誘起電圧[V]が簡易的に求められます。実際に誘起電圧を求める場合には、負荷試験装置を使い、定格速度でモータの軸を回転させて測定する必要があります。

次に誘起電圧から、磁束鎖交数 Ψ [Wb]を求めます。一般的に、誘起電圧と磁束鎖交数には以下のような関係式があります。 f は、定格速度時の電気角周波数[Hz]です。

$$Emf[V] = \omega\Psi = 2\pi f\Psi$$

式を変形し、上記で得られた誘起電圧 Emf [V]と、定格速度運転時の電気角周波数[Hz]を代入することで、磁束鎖交数 Ψ [Wb]を求められます。

$$\Psi = \frac{Emf[V]}{2\pi f}$$

得られた磁束鎖交数 Ψ は、`r_motor_targetmotor_cfg.h` の `MOTOR_CFG_MAGNETIC_FLUX` に設定してください。

9.9 電流制御パラメータ

電流制御パラメータを、表 9-16 に示します。モータのパラメータや PWM キャリア周波数、所望の電流応答性能によって、電流制御のパラメータを算出します。

表 9-16 に示す電流制御パラメータのマクロは、起動時に内部の変数に設定・反映されますが、起動後に調整が必要な場合には、RMW から調整することが可能なパラメータが以下の 2 点、用意されています。表 6-2 を参照してください。すべてのパラメータを変更できるものではありませんので、ご注意ください。

電流制御系固有周波数 : `com_f4_current_omega_hz`

電流制御系減衰係数 : `com_f4_current_zeta`

表 9-16 電流制御パラメータの一覧

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	CURRENT_CFG_OFFSET_CALC_TIME	512.0f	電流オフセットの測定時間設定
	CURRENT_CFG_OMEGA	300.0 f	電流制御系固有周波数[Hz]
	CURRENT_CFG_ZETA	1.0f	電流制御系減衰係数
	CURRENT_CFG_RATE_LIMIT	0.0004f	電流指令変化率制限[A/周期]

CURRENT_CFG_OFFSET_CALC_TIME

起動時に、電流検出のオフセットを測定するときに、オフセット値の測定回数を指定します。通常はデフォルトのままをご利用ください。

CURRENT_CFG_OMEGA, CURRENT_CFG_ZETA

電流制御系固有周波数と電流制御系減衰係数を調整して制御のゲインを調整します。電流制御系固有周波数は、電流制御を行う頻度に比例して設定してください。電流制御周波数（PWM キャリア周波数）の約 1/10 まで設定できますが、位置検出と電流検出のノイズなどを考慮し、マージンを設けて低く設定する場合があります。

たとえば、電流制御周波数が 20kHz(50us 間隔で電流制御が動作)のときは、1/10 まで設定できますので、電流制御系固有周波数は 2kHz を指定できます。しかし、実際には、モータのパラメータの電気定数に起因して、固有周波数が高いと敏感に反応しすぎる場合があります、2kHz よりも下の周波数(たとえば 500Hz~1kHz)程度に設定することが多いです。

電流制御系減衰係数は、0.7~1.0 が常用範囲です。1.0 に近いほど安定で緩やかな応答になります。

CURRENT_CFG_RATE_LIMIT

トルク制御時に、電流指令値が急激に増大しないように、変化率を制限します。1 電流制御周期あたりに変化してもよい電流値[A]を指定します。デフォルトでは、電流制御周期は 50us となっているため、この値を 0.0004[A]とした場合、1 秒あたり 8A の変化まで許容されます。

9.10 速度制御パラメータ

速度制御系のパラメータを、表 9-17 に示します。設定した値が初期値となり、システム起動時に適用されます。表 9-17 に示す速度制御パラメータのマクロは、起動時に内部の変数に設定・反映されますが、起動後に調整が必要な場合には、RMW から調整することが可能なパラメータが以下のように、用意されています。表 6-2 を参照してください。RMW 上で、すべての速度制御パラメータを変更できるものではありませんので、ご注意ください。

速度制御系固有周波数 : `com_f4_speed_omega_hz`

速度制御系減衰係数 : `com_f4_speed_zeta`

摩擦補償パラメータ: `com_f4_friction_comp_brake_speed`, `com_f4_friction_comp_static`,
`com_f4_friction_comp_coulombs`, `com_f4_friction_comp_viscous`

表 9-17 速度制御パラメータの一覧

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	SPEED_CFG_CTRL_PERIOD	0.0005f	制御周期設定 [s]
	SPEED_CFG_OMEGA	5.0f	速度制御系固有周波数 [Hz]
	SPEED_CFG_ZETA	1.0f	速度制御系減衰係数
	SPEED_CFG_SPEED_LIMIT_RPM	2850.0f	速度制限値[r/min] (機械角)
	SPEED_CFG_RATE_LIMIT_RPM	1000.0f	加速度制限 [r/min/s]

SPEED_CFG_CTRL_PERIOD

0.0005s(0.5ms)としてください。変更する場合には、Config_CMT0 のタイマ設定値を変更し、速度制御周期を変更することとなります。

SPEED_CFG_OMEGA, SPEED_CFG_ZETA

速度制御モジュールでは、速度制御系固有周波数と速度制御系減衰係数を調整して制御のゲインを調整します。速度制御系固有周波数を高くすると、応答性が向上し指令速度に対する速度の追従性が向上します。速度制御系固有周波数は電流制御との干渉を防ぐため、設定できる上限が電流制御系の固有周波数の 1/3 となっています。電流制御系の固有周波数が 500Hz であった場合には、 $500\text{Hz}/3=166\text{Hz}$ となります。

デフォルト値は、ホールセンサが低分解能であることを考慮し、制御系の発散を防ぐために追従性を高くせず、マージンを取った値を設定しております。たとえば、外乱が固有周波数より高めで振動する場合には、固有周波数の値を外乱に合わせて増やすことで、外乱の振動への追従性がよくなり、デフォルト設定よりも安定して動く場合があります。

速度制御系減衰係数は 0.7~1.0 は常用範囲とし、値 1 に近いほど安定で緩やかな応答になります。速度の応答を確認しながら調整を行ってください。

SPEED_CFG_RATE_LIMIT_RPM

速度指令値を設定した時に、速度が上昇するスピード（加速度）を設定します。値を大きくすると、早く速度が上昇します。100 を指定した時、1 秒当たり 100r/min、上昇します。停止から 2000r/min まで 20 秒で到達します。

表 9-18 摩擦補償パラメータの一覧

ファイル名	マクロ名	設定値	単位	com 変数・説明
r_motor_module_cfg.h	SPEED_CFG_FRICTION_BRAKE_SPEED	1.0f	rad/s	com_f4_friction_comp_brake_speed
	SPEED_CFG_FRICTION_COULOMBS	0.15f	A	com_f4_friction_comp_coulombs
	SPEED_CFG_FRICTION_STATIC	0.3f	A	com_f4_friction_comp_static
	SPEED_CFG_FRICTION_VISCOUS_COEF	0.0f	A/(rad/s)	com_f4_friction_comp_viscous

SPEED_CFG_FRICTION_BRAKE_SPEED

静止摩擦が発生するしきい値を設定します。目安としてはモータ停止時の速度検出精度を考慮し、小さい値(例：1.0[rad/s])を設定します。

SPEED_CFG_FRICTION_STATIC

補償なしで始動する際、モータが動き出したときのトルク指令値(q 軸電流指令値)の 80%程度の値を目安に設定してください。

SPEED_CFG_FRICTION_COULOMBS

モータに機械系を接続した実際の状況に近い環境において、定格速度の 40%で無負荷状態で回転したときの Iq 指令値を目安に設定します。

SPEED_CFG_FRICTION_VISCOUS_COEF

変化が緩やかな負荷条件の場合、速度制御 PI が支配的に動作する場合や、条件がわからない場合は 0 を設定してください。

9.11 電圧位相進み補償パラメータ

電流検出タイミングを基準として、実際に PWM が出力されるタイミングまで角度を進ませるための補償値です。本サンプルプログラム及びインバータ構成では、0.5 サンプル進ませることで、PWM 出カタイミングを一致させることができます。

表 9-19 コンフィグレーション情報一覧

ファイル名	マクロ名	設定値	説明
r_motor_module_cfg.h	CURRENT_CFG_PERIOD_MAG_VALUE	0.5f	進み補償を行うサンプル数を設定します。0.5 を設定してください。

9.12 ホールセンサベクトル制御パラメータ

ここでは、ホールセンサベクトル制御に必要なパラメータの設定方法について説明します。ホールセンサベクトル制御は、パラメータが不適切であると、始動特性や低速運転において所望の性能を発揮できない場合があります。

ホールセンサベクトル制御に用いる、パラメータ一覧を表 9-20 に示します。なお、速度制御パラメータに関連している SPEED_CFG_FRICTION_BRAKE_SPEED 等の摩擦補償に関するパラメータは、9.10 で説明しています。

表 9-20 ホールセンサベクトル制御の設定パラメータ

ファイル名	マクロ名	設定値	単位	com 変数・説明
r_motor_module_cfg.h	SPEED_CFG_FRICTION_BRAKE_SPEED	1.0f	rad/s	com_f4_friction_comp_brake_speed
	SPEED_CFG_FRICTION_STATIC	0.3f	A	com_f4_friction_comp_static
	SPEED_CFG_FRICTION_COULOMBS	0.15f	A	com_f4_friction_comp_coulombs
	SPEED_CFG_FRICTION_VISCOUS_COEF	0.0f	A/(rad/s)	com_f4_friction_comp_viscous
	SENSOR_HALL_CFG_ANGLE_CORRECT	0.0f	rad	com_f4_hall_angle_correct
	SENSOR_HALL_CFG_SPEED_DEVIATION_TOLERANCE	0.1f	-	com_f4_hall_spdmix_dev_th 0~1.0 の間の数値を入れてください。
	SENSOR_HALL_CFG_SPEED_MIX_TH	自動計算	rad/s	com_f4_hall_spdmix_elec_speed_th (SPEED_CFG_OMEGA * MTR_TWOPI * 30.0f / 6.0f)
	SENSOR_HALL_CFG_DEFAULT_COUNTS	4000	-	
	SENSOR_HALL_CFG_MAXIMUM_PERIOD	5000	-	
	SENSOR_HALL_CFG_HALL_POLEPAIRS	自動計算		
	SENSOR_HALL_CFG_PATTERN_1	1	-	
	SENSOR_HALL_CFG_PATTERN_2	5	-	
	SENSOR_HALL_CFG_PATTERN_3	4	-	
	SENSOR_HALL_CFG_PATTERN_4	6	-	
	SENSOR_HALL_CFG_PATTERN_5	2	-	
SENSOR_HALL_CFG_PATTERN_6	3	-		

SENSOR_HALL_CFG_ANGLE_CORRECT

ホール検出角度の修正値、明らかにトルク出力が仕様より低いや正回転と逆回転で出力できるトルクが大きくなった場合、この値を調整してください。その他の場合は0としてください。

SENSOR_HALL_CFG_SPEED_DEVIATION_TOLERANCE

ホールセンサの速度検出の自動切り替え時許容するリップル率（範囲：0.0~1.0）。定格速度における速度リップル率を見ながら 0 から調整してください。

SENSOR_HALL_CFG_DEFAULT_COUNTS

SENSOR_HALL_CFG_MAXIMUM_PERIOD の 80%~100%を設定してください。

SENSOR_HALL_CFG_MAXIMUM_PERIOD

速度を 0（停止）と判定するカウント数で、カウントの頻度によりますので、約 0.25 秒に相当するカウント数を設定してください。

SENSOR_HALL_CFG_SPEED_MIX_TH

自動切り替えモードで速度を高精度の速度に切り替えはじめる速度閾値、この速度における実際のサンプリングレートを速度制御の帯域から十分離すように設定してください。

SENSOR_HALL_CFG_PATTERN_1, SENSOR_HALL_CFG_PATTERN_2,

SENSOR_HALL_CFG_PATTERN_3, SENSOR_HALL_CFG_PATTERN_4,

SENSOR_HALL_CFG_PATTERN_5, SENSOR_HALL_CFG_PATTERN_6

ホールセンサのパターン番号を設定します。検出回路の High・Low の定義やホールセンサ信号と電気角の配置関係による仕様に合わせて設定してください。

10. スマート・コンフィグレータ設定

サンプルプログラムでは、スマート・コンフィグレータを使用してプロジェクトを作成しています。使用しているコンポーネントとユーザ領域に追加した関数を説明します。

10.1 クロック設定

クロック設定を表 10-1 に示します。

表 10-1 MCU クロック設定

クロックの種類	設定クロック
メインクロック	240Mz
システムクロック (ICLK)	120MHz
周辺モジュールクロック (PCLKA)	120MHz
周辺モジュールクロック (PCLKB)	60MHz
周辺モジュールクロック (PCLKC)	120MHz
周辺モジュールクロック (PCLKD)	60MHz
FlashIF クロック (FCLK)	60MHz
IWDTCLK	120kHz

10.2 コンポーネント設定

使用するコンポーネント情報と機能割り当てを表 10-2 に示します。

表 10-2 スマート・コンフィグレータのコンポーネントと機能割り当て

機能	コンポーネント
3相 PWM 出力、 電流検出の AD 変換	Config_MOTOR
A/D 変換処理 (インバータ母線電圧検出)	Config_S12AD2
使用ポートの設定	Config_PORT
位置速度制御割り込みタイマ	Config_CMT0
独立ウォッチドックタイマ	Config_IWDT
過電流検出	Config_POE

10.3 AD 設定

MCU 内蔵の 12bit AD コンバータ(S12AD)を用いて、U 相出力電流と、W 相出力電流、インバータ母線電圧を測定します。割り当てチャンネルと、検出タイミングを、表 10-3 に示します。

MCI-LV-1 はシャント抵抗による電流検出方式のため、キャリアの谷の条件で電流を取得するように設定しています。また、AD 検出開始から変換終了までを行った後、AD 変換終了割り込みを発生させ、r_Config_MOTOR_ad_interrupt 割り込み関数を通じて電流制御が実行されます。

表 10-3 AD のチャンネルと検出タイミング設定

機能	割り当てチャンネル	変換開始トリガ
インバータ母線電圧測定	AN003	TRG4AN (MTU4.TCNT と MTU4.TADCOBRA のコンペアマッチ)
U 相電流測定	AN000	
V 相電流測定	AN001	
W 相電流測定	AN002	

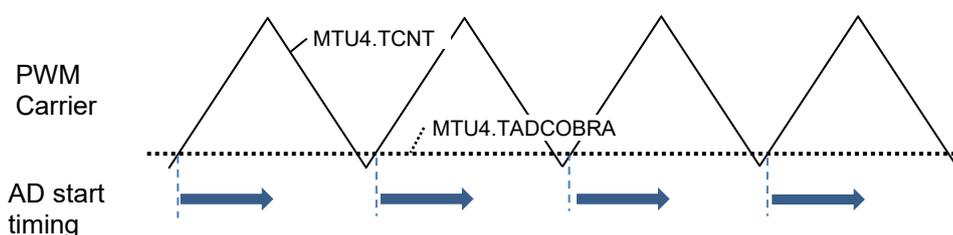


図 10-1 AD 検出開始タイミング

10.4 モータ設定

本サンプルプログラムでは、MTU3 を用いて PWM を出力します。Duty の反映は、キャリアの谷で行われます。設定は、コンポーネントから Config_MOTOR を選択することで、行えます。

表 10-4 MTU の設定

タイマ設定		
周期設定	タイマ動作周期	50us
	カウントクロック分周比	1
	TGRA レジスタ値	3240
	デッドタイム	2.0us
出力パルスおよび A/D 変換トリガ設定	A/D 変換開始トリガ間引き	間引きしない
	Up	ハイレベル
	Un	ハイレベル
	Vp	ハイレベル
	Vn	ハイレベル
	Wp	ハイレベル
	Wn	ハイレベル
タイマパルス出力端子設定	U 相	MTU4 B-D
	V 相	MTU4 A-C
	W 相	MTU3 B-D
A/D コンバータ設定		
A/D 変換設定	計測対象端子 Iu	AN000
	計測対象端子 Iv	AN001
	計測対象端子 Iw	AN002
	計測対象端子 Vdc	AN003

10.5 割り込み

モータコンポーネントを使用した MCU の割り込み情報を表 10-5 に示します。

表 10-5 割り込み一覧

コンポーネント	割り込み関数	説明
Config_MOTOR	r_Config_MOTOR_ad_interrupt	AD 変換終了割り込み 割り込みレベル：12 多重割り込み：許可
Config_S12AD0/1	なし	なし
Config_PORT	なし	なし
Config_CMT0	r_Config_CMT0_cmi0_interrupt	速度制御割り込み 割り込みレベル：11 多重割り込み：許可
Config_IWDT	なし	なし
Config_POE	r_Config_POE_oei1_interrupt	HW 過電流の割り込み 割り込みレベル：15 多重割り込み：禁止

10.6 ユーザコード詳細

ユーザコード領域に作成した関数一覧を表 10-6 に示します。

表 10-6 ユーザ領域の関数一覧

コンポーネント	関数	説明
Config_PORT	R_Config_PORT_GetSW1	SW1 状態の取得
	R_Config_PORT_GetSW2	SW2 状態の取得
	R_Config_PORT_Led1_on	LED1 点灯
	R_Config_PORT_Led2_on	LED2 点灯
	R_Config_PORT_Led1_off	LED1 消灯
	R_Config_PORT_Led2_off	LED2 消灯
	R_Config_PORT_GetHallSignal	ホール信号取得

10.7 POE 設定

Config_POE で設定が可能な、POE 設定を表 10-7 に示します。出力ピン設定は、インバータ仕様によって異なりますので、ご使用のインバータの信号仕様を確認ください。

表 10-7 POE 設定

機能			設定
POE0#	要求受付条件	POE0#入力の立ち下がリエッジで 要求を受け付ける	✓
	サンプリング回数		-
	割り込み設定	アウトプットイネーブル割り込み 1 許可	✓
		優先順位(グループ BL1)	レベル 15 (最高)
MTU3/MTU4	出力短絡時に端子の出力を停止する		✓
	アクティブレベル設定有効		✓
	出力ピン設定 MTIOC3B.MTIOC3D	MTIOC3B アクティブレベル MTIOC3D アクティブレベル	High High
	出力ピン設定 MTIOC4A.MTIOC4C	MTIOC4A アクティブレベル MTIOC4C アクティブレベル	High High
	出力ピン設定 MTIOC4B.MTIOC4D	MTIOC4B アクティブレベル MTIOC4D アクティブレベル	High High

10.8 端子設定

端子のインタフェース情報を表 10-8 に示します。

表 10-8 端子インタフェース

機能	RX26T
インバータ母線電圧測定	P43 / AN003
U 相電流測定	P40 / AN000
V 相電流測定	P41 / AN001
W 相電流測定	P42 / AN002
VR	P50 / AN204 / CMPC42
PWM 出力 (U _p) / “High” アクティブ	P76 / MTIOC4D / GTIOC2B
PWM 出力 (V _p) / “High” アクティブ	P75 / MTIOC4C / GTIOC1B
PWM 出力 (W _p) / “High” アクティブ	P74 / MTIOC3D / GTIOC0B
PWM 出力 (U _n) / “High” アクティブ	P73 / MTIOC4B / GTIOC2A
PWM 出力 (V _n) / “High” アクティブ	P72 / MTIOC4A / GTIOC1A
PWM 出力 (W _n) / “High” アクティブ	P71 / MTIOC3B / GTIOC0A
過電流検出時の PWM 緊急停止入力	P70 / POE0#
LED2	P20
LED1	P21
SW2	P22
SW1	P23
ホールセンサ W	P24
ホールセンサ V	P27
ホールセンサ U	P30

11. 評価結果

11.1 モータ制御評価

11.1.1 加減速特性

停止状態から下記の速度指令シーケンスで運転した特性データを以下に示します。

【速度指令シーケンス】

0 → 300r/min → 2400r/min → 300r/min → -300r/min → -2400r/min → -300r/min → 0

※加減速度制限：1500(r/min)/s、回転速度(正=CW、負=CCW)、無負荷条件



図 11-1 加減速および正転逆転特性

11.1.2 負荷特性

2000r/min 運転時の負荷特性波形を以下に示します。なお、200r/min から 2000r/min まで、負荷試験を行い、定格負荷で運転できることを確認しています。



図 11-2 負荷特性(2000r/min 時)

11.2 CPU 使用率

各制御周期の CPU 処理時間と負荷率を以下に示します。

表 11-1 制御ループと CPU 負荷率

制御ループ種類	制御周期	処理時間	CPU 負荷率
電流制御ループ	50 us (間引き 0 回)	15.44us	30.9%
速度制御ループ	500 us	4.90us (多重割り込みで最大 17.14us)	1.0%

11.3 プログラムサイズ・RAM 使用量

本サンプルプログラムでのプログラムサイズ(ROM)と、RAM 使用量は以下の通りです。コンパイラの最適化設定において、最適化レベル 2 (-optimize = 2)に設定し、最適化方法をコード・サイズ重視の最適化(-size)に設定しています。

表 11-2 プログラムサイズと RAM 使用量

プログラムサイズ(ROM)	22647 [Bytes]
RAM 使用量	9704 [Bytes]
スタック解析結果の最大値	300 [Bytes]
スタックサイズの IDE 環境の設定値	5120 [Bytes]

12. FAQ

12.1 こんなときは

代表的な現象と、その解決例を表 12-1 に示します。

表 12-1 現象と回答の一覧

現象	回答
モータが動かない 過電流エラーになる	<ul style="list-style-type: none"> ・ U,V,W の断線・接触不良がないか確認してください。 ・ ホールセンサ線の断線に注意してください。ホールセンサ線が断線・接触不良している場合、正常にモータ制御できません。 ・ 軸がロックされているか、過負荷状態になっていないか確認してください。
モータ運転時に母線電圧が低下する	モータの負荷に対して電源の出力性能不足や、モータ側でショートしている場合があります。
始動時に速度変動が発生します。速度変動をなくすことはできますか	<p>速度制御において、停止状態からベクトル制御をおこなうため、ホールセンサのパルスが入力されるまでは速度がゼロとみなされ、速度調節器が飽和することにより、速度変動が生じます。この速度変動を完全に消すことは、ホールセンサが電気角 1 周 6 パルスの低分解能エンコーダという仕様であることから、除去は困難です。</p> <p>なお、トルク制御で運転する場合には、このような問題は生じません。</p>
200r/min 以下の低速で回転できない	低分解能なホールセンサは、検出信号変化の発生間隔が低速時には長くなります。よって、速度検出の更新周期が著しく遅くなり、速度調節器が正常に速度を制御できなくなる場合があります。ホールセンサベクトル制御の速度制御モードは、低速域での回転が必要なアプリケーションには不適です。トルク制御モードおよび外部の別の速度検出方法を併用することを検討してください。
エラーで停止後、モータを回転できない	6.8(c)を参照してください。エラーからの復帰方法が説明されています。
運転開始を行っても、エラーで停止してしまう	<ul style="list-style-type: none"> ・ 6.8(c)を参照してエラー要因を確認してください ・ インバータ母線電圧の過電圧または低電圧、出力が短絡していないか確認してください ・ センサ関連の信号設定が正しいか確認してください。 ・ スマート・コンフィグレータの POE 設定やモータ設定（MTU の端子・アクティブレベル設定）等が、インバータ回路仕様に合わせて適切に設定されているか、確認してください。
RMW に表示される変数の値が異常となってしまう	サンプルプログラムの変更を行った場合、ビルド後に HardwareDebug フォルダ内に生成される Map ファイルを RMW に登録し、サンプルプログラムの変数状態を更新する作業が必要です。この作業を省略した場合、変数が正しく表示できない場合があります。詳細は、6.6 を参照ください。
RMW から値を設定しても、反映されない	com_u1_enable_write の変数操作で、パラメータを書き換えます。com_u1_enable_write への数値の書き換えタイミングが、パラメータの書き込みより先の場合、内部の反映処理が先に動作します。以下のように対応してください。

	<ul style="list-style-type: none">・ com_u1_enable_write を最後の行に置く・ com_u1_enable_write の書き込み(1回目 : 1、二回目 : 0)を2回行う、または、トグル書き込みする
スムーズに始動にはどのようなモータ・機械システムを選べばよいか	<ul style="list-style-type: none">・ 極対数の多いモータを選択することで、ホールセンサの分解能が高くなり、低速域の性能が向上され、よりスムーズに始動できる・ 無負荷より、イナーシャ負荷あったほうがスムーズに始動できる

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2024/12/26	-	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しないでください。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、リセットを解除してください。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っていると、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、ノイズ輻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。

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、これに限りません。）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た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て、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としたします。
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/