RA6T1

永久磁石同期モータのセンサレス 120 度通電制御

要旨

本アプリケーションノートは RA6T1 マイクロコントローラで、永久磁石同期モータを駆動するセンサレス 120 度通電制御ソフトウェア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することを目的としています。

本アプリケーションノート対象ソフトウェア(サンプルソフトウェア)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。サンプルソフトウェアを使用する場合、適切な環境で十分な評価をした上で御使用ください。

動作確認デバイス

サンプルソフトウェアの動作確認は下記のデバイスで行っております。

・ RA6T1 (R7FA6T1AD3CFP)

目次

1. 概説 3
1.1 開発環境 3

2. システム概要 4
2.1 ハードウェア構成 4
2.2 ハードウェア仕様 5
2.2.1 ユーザインタフェース 5
2.2.2 周辺機能 7
2.3 ソフトウェア構成 12
2.3.1 ソフトウェア・ファイル構成 12
2.3.2 モジュール構成 13
2.4 ソフトウェア仕様 14
2.5 割込み優先順位 15

3. 制御ソフトウェア説明 16
3.1 制御内容 16
3.1.1 モータ起動/停止 16
3.1.2 A/D 変換 16
3.1.3 速度制御 18
3.1.4 PWM による電圧制御 19
3.1.5 状態遷移 21
3.1.6 センサレス制御時の始動方法 22
3.1.7 システム保護機能 23
3.1.8 AD トリガ 24
3.2 センサレス 120 度通電制御ソフトウェア関数仕様 25
3.3 Contents of control 35

3.3.1 Configuration Options 35
3.3.2 Configuration Options for included modules 35
3.4 制御フロー（フローチャート） 41
3.4.1 メイン処理 41
3.4.2 50 [µs]周期割り込み(キャリア周期割り込み)処理 42
3.4.3 1 [ms]周期割り込み処理 43
3.4.4 過電流検出割り込み処理 44

4. 評価環境説明 45
4.1 動作環境 45
4.2 プロジェクトのインポート 46
4.3 ビルドとデバッグ 47
4.4 サンプルソフトウェアの操作概要 48
4.4.1 クイックスタート 48
4.5 モータ制御開発支援ツール「Renesas Motor Workbench」 49
4.5.1 概要 49
4.5.2 Easy 機能操作例 50
4.5.3 Analyzer 機能用変数一覧 52
4.5.4 Analyzer 機能操作例 53

5. 参考ドキュメント 55

1. 概説

本アプリケーションノートでは、RA6T1 マイクロコントローラで、永久磁石同期モータ(PMSM)を駆動するセンサレス 120 度通電制御ソフトウェアの実装方法及びモータ開発支援ツール「Renesas Motor Workbench」の使用方法について説明します。

なお、このサンプルソフトウェアは「永久磁石同期モータの 120 度通電制御 (アルゴリズム編)」(R01AN2657)のアルゴリズムを使用していますので、アルゴリズムの詳細についてはこちらを参照してください。

1.1 開発環境

サンプルソフトウェアの開発環境を表 1-1、表 1-2 に示します。

表 1-1 ハードウェア開発環境

マイコン	評価ボード 注1	モータ 注2
RA6T1(R7FA6T1AD3CFP)	48V 系インバータボード RA6T1 CPU カード	TG-55L-KA 24V

表 1-2 ソフトウェア開発環境

e²studio バージョン	FSP バージョン	ツールチェーン バージョン
V2021-10	V3.5.0 以降	GCC ARM Embedded : V10.3.1.20210824

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

【注】 1. 48V 系インバータボード(RTK0EM0000B10020BJ) は、キット製品 RTK0EMA170S00020BJ、RTK0EMX270S00020BJ に同梱しており、ルネサスエレクトロニクス株式会社の製品です。
RA6T1 CPU カード(RTK0EMA170C00000BJ)は、ルネサスエレクトロニクス株式会社の製品です。
2. TG-55L-KA,24V は、ツカサ電工株式会社の製品です。
ツカサ電工株式会社(https://www.tsukasa-d.co.jp/)

2. システム概要

サンプルソフトウェアのシステム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。

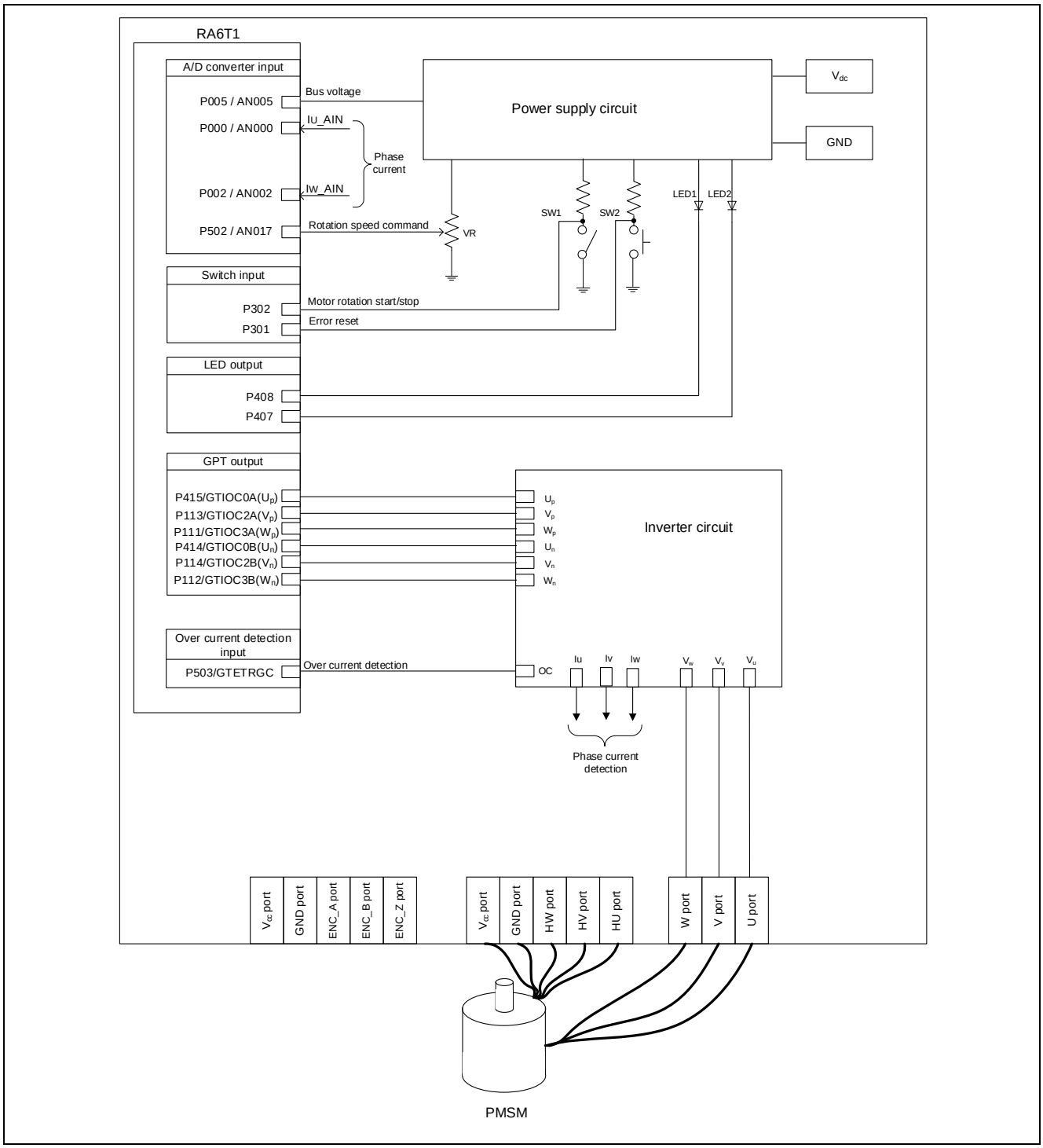


図 2-1 ハードウェア構成図

2.2 ハードウェア仕様

2.2.1 ユーザインタフェース

サンプルソフトウェアのユーザインタフェースを表 2-1 に示します。

表 2-1 インバータボードユーザインタフェース

項目	インタフェース部品	機能
回転速度指令	可変抵抗器	回転速度指令値入力
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
LED1	橙色 LED	・モータ駆動時　：点灯 ・モータ停止時　：消灯
LED2	橙色 LED	・エラー検出時　：点灯 ・通常動作時　：消灯
LED3	橙色 LED	未使用

表 2-2 CPU ボードユーザインタフェース

項目	インタフェース部品	機能
LED1	緑色 LED	・モータ駆動時　：点灯 ・モータ停止時　：消灯
LED2	緑色 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(RESET)	システムリセット

サンプルソフトウェアの端子インタフェースを表 2-3 に示します。

表 2-3 端子インタフェース

R7FA6T2BD3CFP 端子名	機能
P014 / AN005	インバータ母線電圧測定
P502 / AN017	回転速度指令値入力用（アナログ値）
P302	START/STOP トグルスイッチ
P301	ERROR RESET プッシュスイッチ
P408	LED1 点灯／消灯制御
P407	LED2 点灯／消灯制御
P000 / AN000	U 相電流測定
P002 / AN002	W 相電流測定
P504 / AN018	U 相電圧測定
P508 / AN020	V 相電圧測定
P015 / AN006	W 相電圧測定
P415 / GTIOC0A	PWM 出力 (Up)/“Low”アクティブ
P113 / GTIOC2A	PWM 出力 (Vp)/“Low”アクティブ
P111 / GTIOC3A	PWM 出力 (Wp)/“Low”アクティブ
P414 / GTIOC0B	PWM 出力 (Un)/“High”アクティブ
P114 / GTIOC2B	PWM 出力 (Vn)/“High”アクティブ
P112 / GTIOC3B	PWM 出力 (Wn)/“High”アクティブ
P503 / GTETRGC	過電流検出時の PWM 緊急停止入力

2.2.2 周辺機能

サンプルソフトウェアで使用する周辺機能一覧を表 2-4 に示します。

表 2-4 周辺機能対応表

周辺機能	リソース	用途
12 ビット A/D コンバータ	AN000, AN002, AN005, AN006, AN017, AN018, AN020	・回転速度指令値入力 ・各 U/W 相電流 ・各 U/V/W 相電圧 ・インバータ母線電圧測定
AGT	AGT0	1 [ms]インターバルタイマ
GPT	CH0, CH1, CH2, CH3	・相補 PWM 出力 ・回転速度計測用フリーランタイマ
POEG	Group C	PWM 出力端子をハイインピーダンス状態にし、PWM 出力を停止

(1). 12 ビット A/D コンバータ(ADC12)

U 相電流(Iu)、W 相電流(Iw)、U 相電圧(Vu)、V 相電圧(Vv)、W 相電圧(Vw)、およびインバータ母線電圧(V_{dc})と回転速度指令値(VR)を、「シングルスキャンモード」で測定します(ハードウェアトリガを使用)。

A/D 変換は GPT のアンダーフロー(PWM の谷)と連動して動作させています。

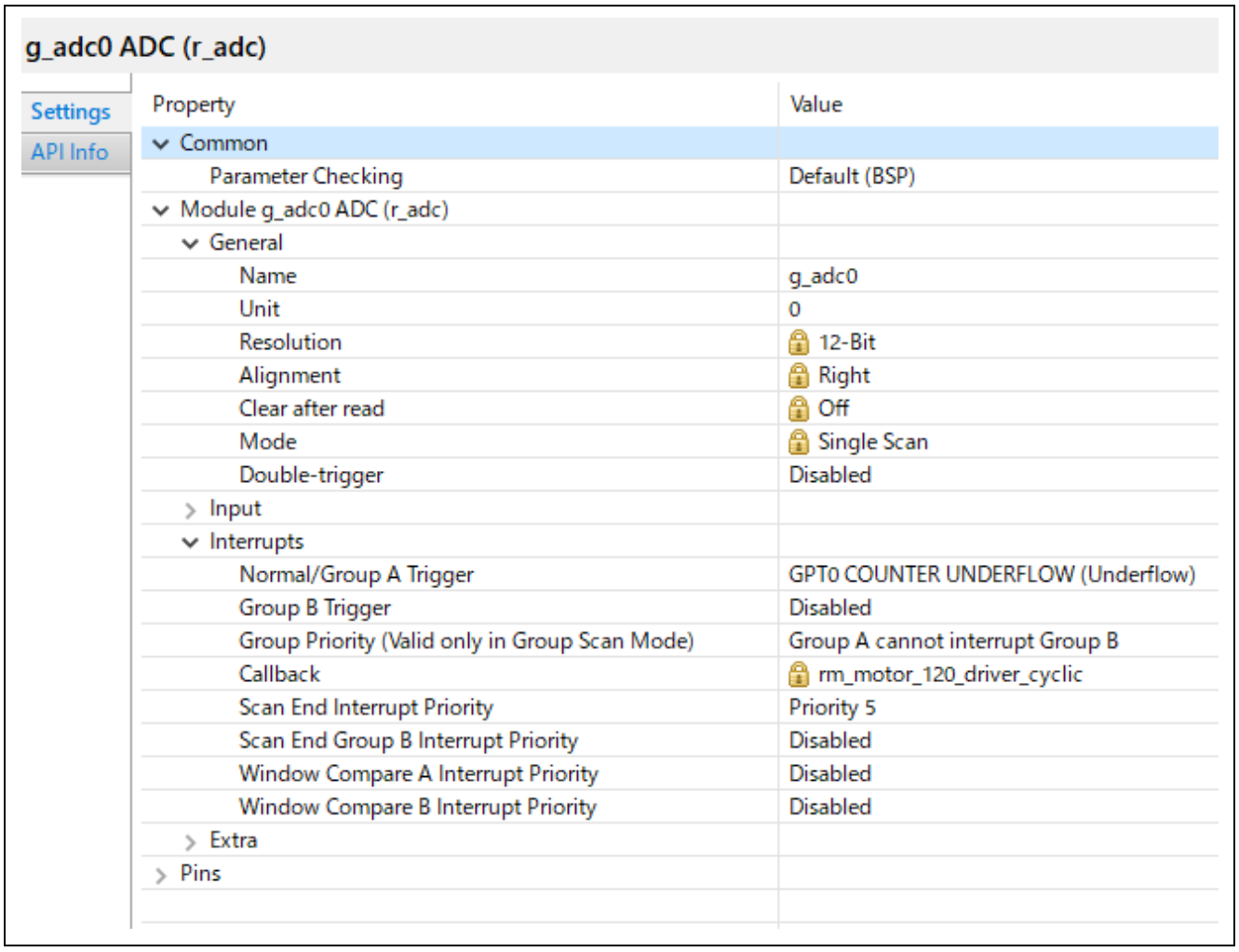


図 2-2 ADC ドライバの FSP コンフィグレーション (FSP3.5.0)

(2). 低消費電力非同期汎用タイマ（AGT）

1 [ms]インターバルタイマとして使用します。

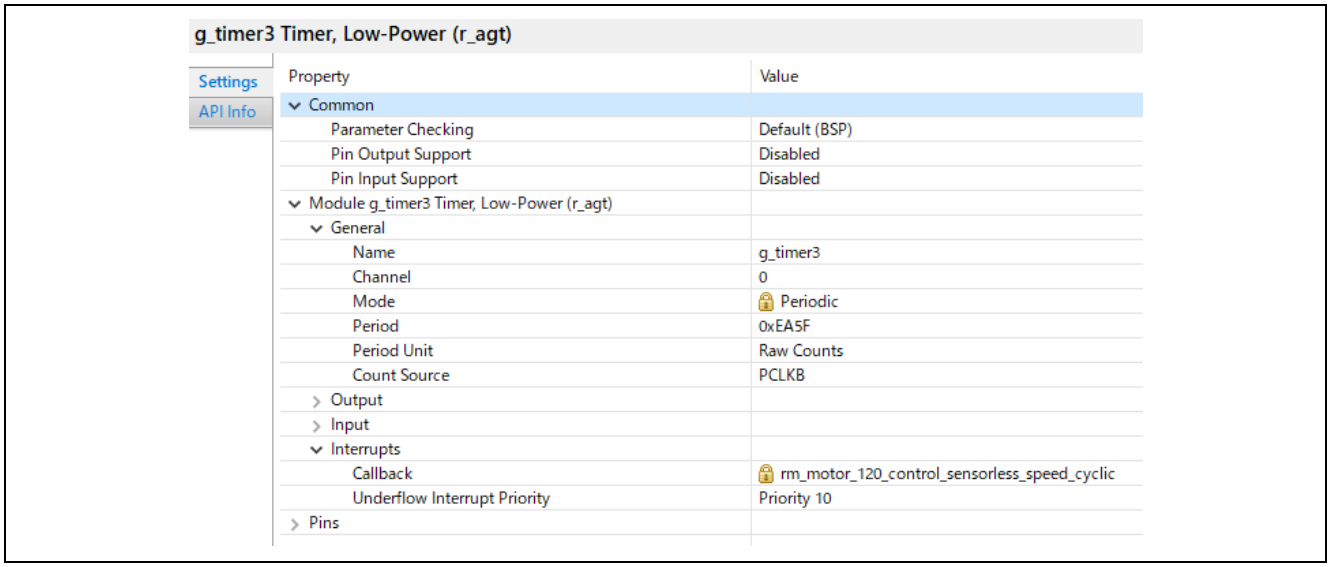


図 2-3 AGT ドライバの FSP コンフィグレーション (FSP3.5.0)

(3). 汎用 PWM タイマ（GPT）

チャネル 0、2、3 の相補 PWM 出力動作モードを使用して、デッドタイム付きの出力を行います。

チャネル 1 を速度計測用フリーランタイマとして使用します。ただし割込みは使用しません。

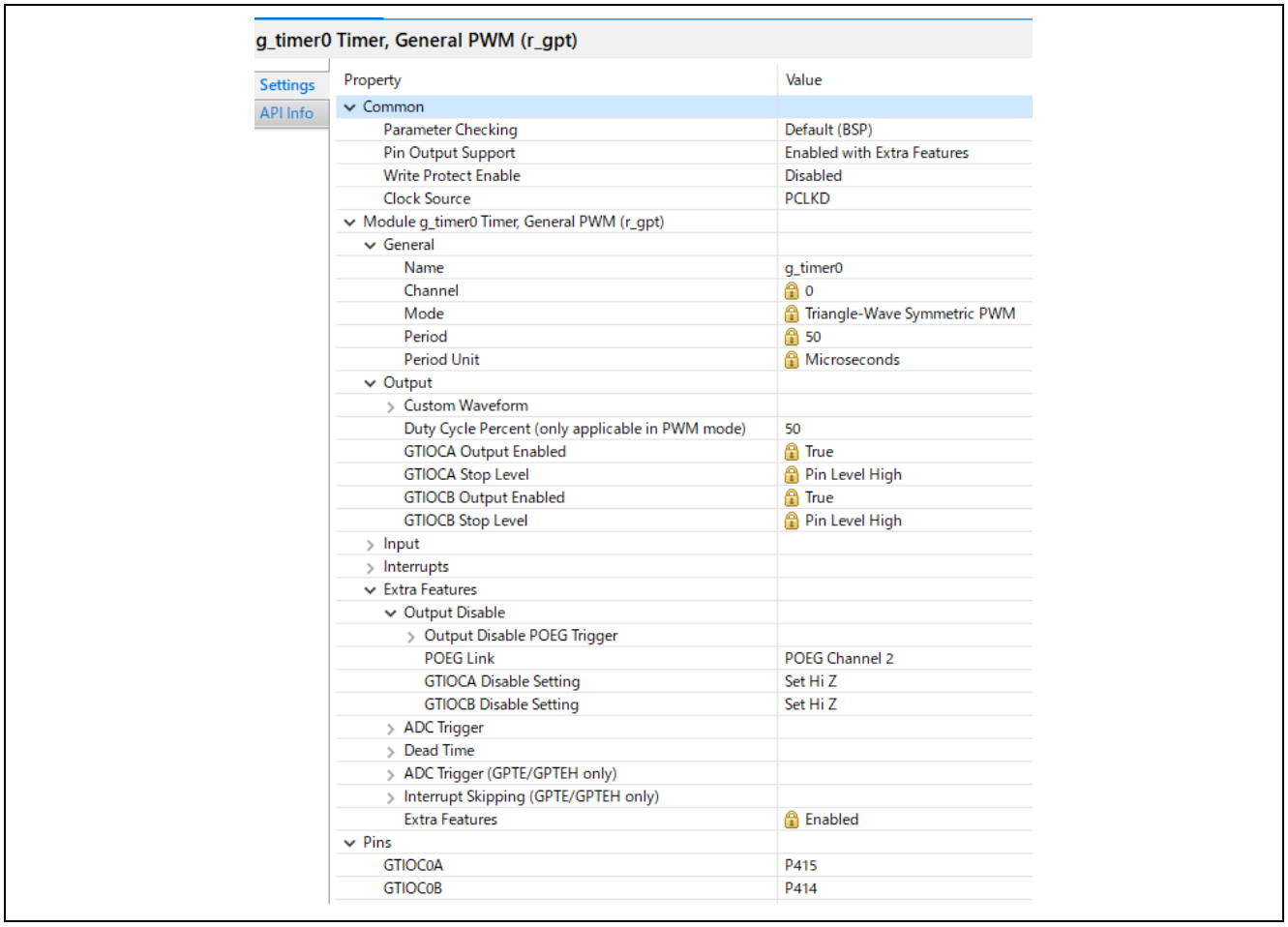


図 2-4 GPT ドライバ(相補 PWM 出力)の FSP コンフィグレーション(FSP3.5.0)

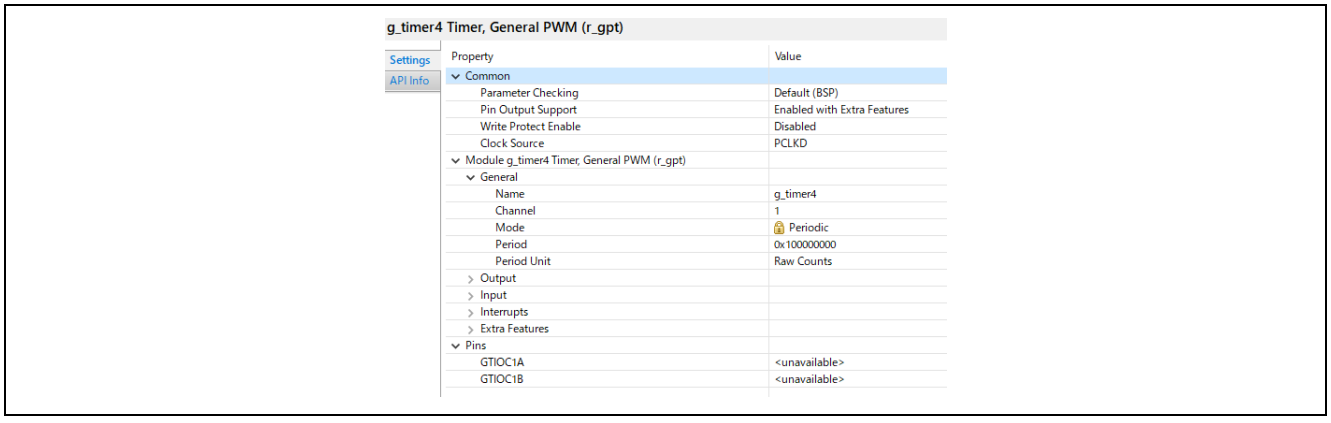


図 2-5 GPT ドライバ(速度計測用フリーランタイマ)の FSP コンフィグレーション(FSP3.5.0)

(4). GPT 用ポートアウトプットイネーブル（POEG）

過電流検出時（GTETRGC 端子の Low レベル検出時）は PWM 出力端子をハイインピーダンス状態にします。

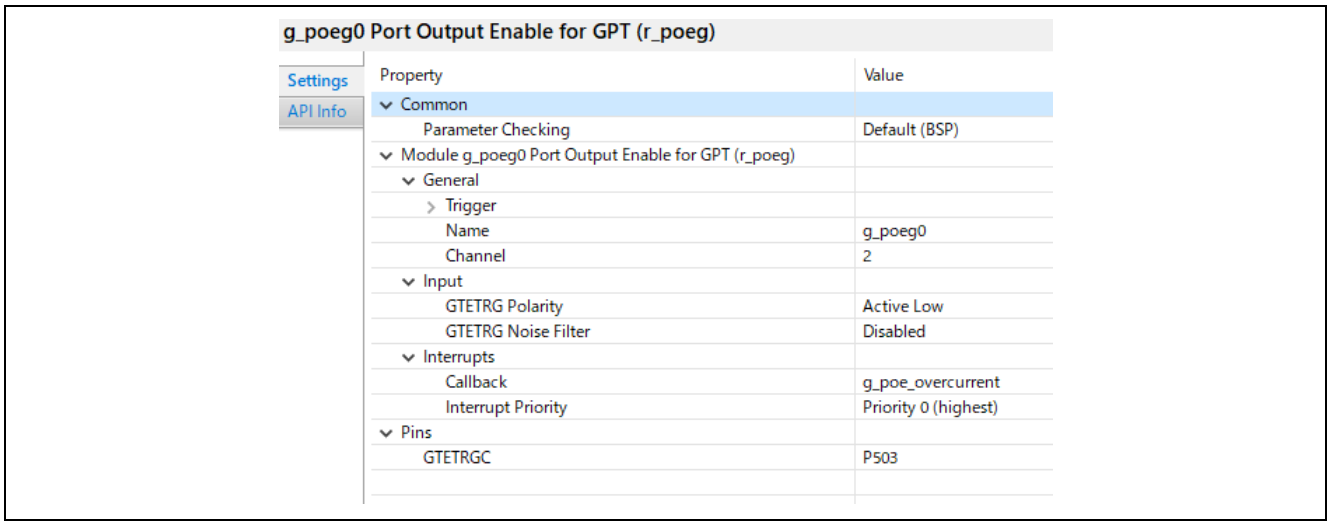


図 2-6 POEG ドライバの FSP コンフィグレーション(FSP3.5.0)

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

サンプルソフトウェアのフォルダとファイル構成を下記に示します。

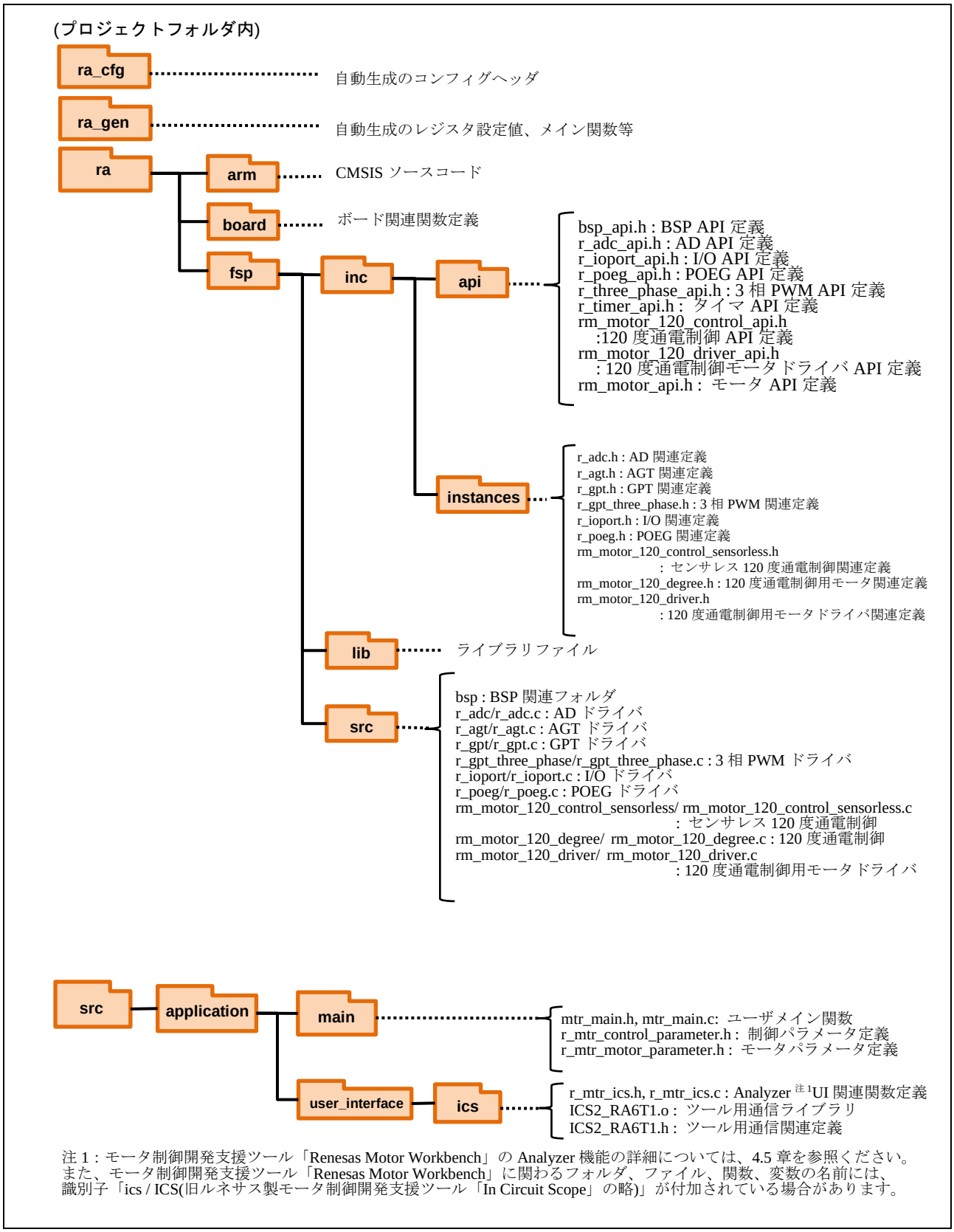


図 2-7 フォルダ・ファイル構成

2.3.2 モジュール構成

サンプルソフトウェアのモジュール構成を図 2-8 に示します。

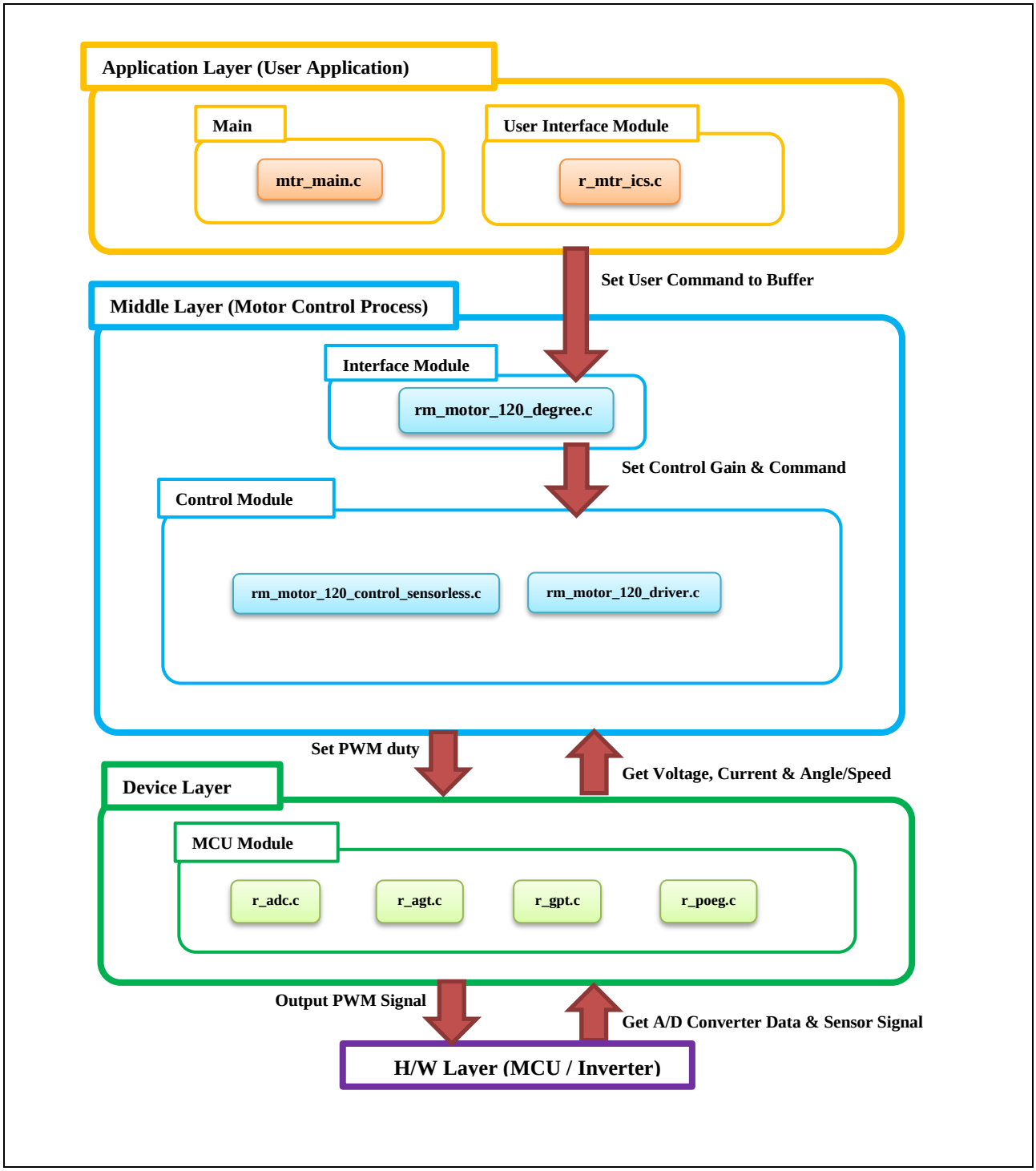


図 2-8 モジュール構成

2.4 ソフトウェア仕様

サンプルソフトウェアの基本仕様を下記に示します。センサレス 120 度通電制御の詳細に関しては「永久磁石同期モータの 120 度通電制御 (アルゴリズム編)」(R01AN2657)を参照してください。

表 2-5 センサレス 120 度通電制御ソフトウェア基本仕様

項目	内容	
制御方式	120 度通電制御	
回転子磁極位置検出	誘起電圧による位置検出(60 度毎)	
モータ回転開始/停止	SW1 のレベルにより判定 または Renesas Motor Workbench から入力	
入力電圧	DC 24V	
メインクロック周波数	120 [MHz]	
キャリア(PWM)周波数	20 [kHz](キャリア周期：50 [µs])	
デッドタイム	2 [µs]	
制御周期	速度制御：1 [ms]	
回転速度範囲	CW：0 [rpm] ～ 2650 [rpm] CCW：0 [rpm] ～ 2650 [rpm]	
コンパイラ最適化設定	最適化レベル	Optimize more(-O2) (デフォルト設定)
保護停止処理	以下のいずれかの条件の時、モータ制御信号出力(6 本)を非アクティブにする 1.各相の電流が 0.89(=0.42*sqrt(2)*1.5) [A]を超過(50 [µs]毎に監視) 2.インバータ母線電圧が 28 [V]を超過(50 [µs]毎に監視) 3.インバータ母線電圧が 12[V]未満(50 [µs]毎に監視) 4.回転速度が 3000 [rpm]を超過(50 [µs]毎に監視) 5.センサレス駆動時、ゼロクロス検出が 2[s]間未発生 6.仮想ホールセンサパターン(位置情報)の異常検出 外部からの過電流検出信号(GTETRGC 端子に Low レベルを検出)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	

2.5 割込み優先順位

サンプルソフトウェアで使用している割り込みと優先順位を下記に示します。

表 2-6 割込み優先順位

割込みレベル	優先度	処理
15	Min	
14		
13		
12		
11		
10		1[ms]割り込み処理
9		
8		
7		
6		
5		A/D 変換完了割込み
4		
3		
2		
1		
0	Max	過電流異常割込み

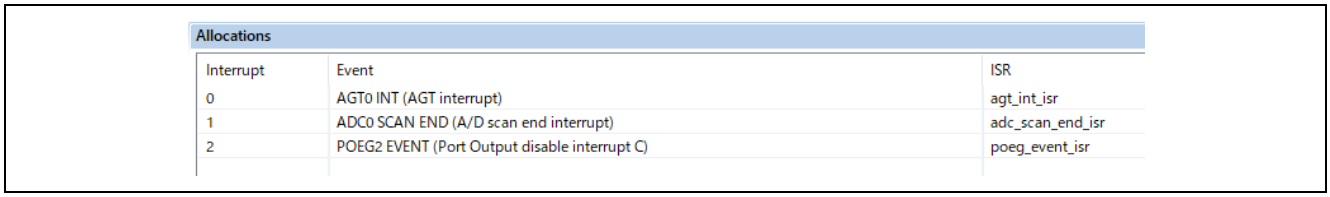


図 2-9 FSP 割り込みコンフィグレーション(FSP3.5.0)

3. 制御ソフトウェア説明

サンプルソフトウェア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、Renesas Motor Workbench からの入力または SW1 から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、”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、”High”レベルのとき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 Renesas Motor Workbench 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 VR 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。

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)		チャネル
回転速度指令値	CW	0 [rpm]～2650[rpm]：0800H～0FFFH	AN017
	CCW	0 [rpm]～2650[rpm]：07FFH～0000H	

(2). インバータ母線電圧

以下の表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)	チャネル
インバータ母線電圧	0 [V]～111 [V]：0000H～0FFFH	AN005

(3). U 相、V 相、W 相電圧

以下の表のように、U 相、V 相、W 相電圧を測定し、ゼロクロス判定に使用します。

表 3-3 U、V、W 相電圧の変換比

項目	変換比 (U 相、V 相、W 相電圧：A/D 変換値)	チャネル
U 相、V 相、W 相電圧	0 [V]～111 [V]：0000H～0FFFH	Vu：AN018 Vv：AN020 Vw：AN006

(4). U 相、W 相電流

以下の表のように、U 相、W 相電流を測定し、過電流検出に使用します。

表 3-4 U、W 相電流の変換比

項目	変換比 (U 相、V 相、W 相電流：A/D 変換値)	チャネル
U 相、W 相電流	-12.5 [A]～12.5 [A]：0000H～0E8BH 注 電流値＝(5.0V－2.5V)÷(0.01Ohm×20)＝12.5A このシステムでは電流検出回路を 5V から 3V にレベルシフトしているため、0E8BH が A/D 変換の上限となります。	Iu：AN000 Iw：AN002

【注】 A/D 変換特性の詳細に関しては、「RA6T1 グループ　ユーザーズマニュアル　ハードウェア編」を参照してください。

3.1.3 速度制御

サンプルソフトウェアのモータ回転速度は、汎用 PWM タイマ(GPT)のチャネル 0 のタイマをフリーランニングさせ、ゼロクロス発生時にタイマ値を取り込み、2π[rad]前の取り込み値との差分から演算します。

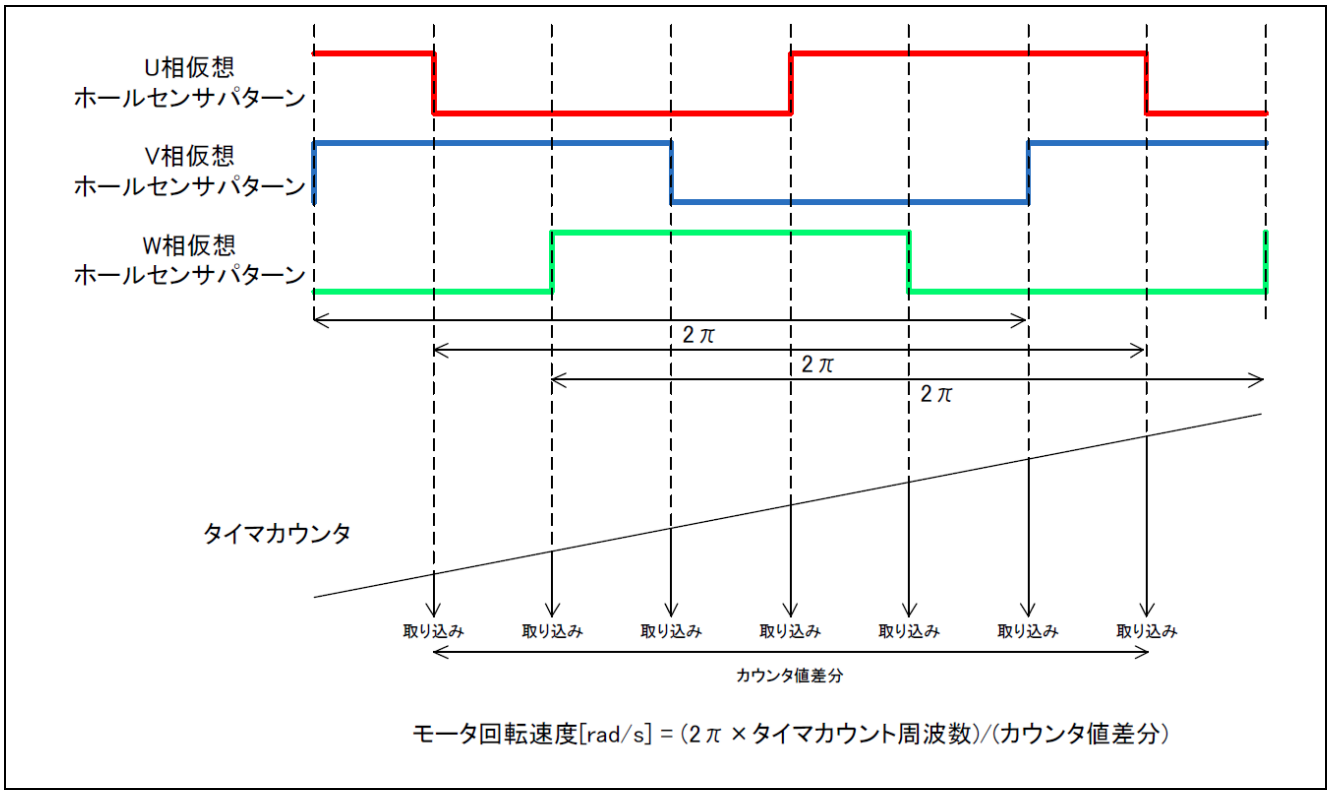


図 3-1 モータ回転速度の演算方法

サンプルソフトウェアでの速度制御は、PI 制御によって行います。下記の速度制御 PI 制御によって電圧指令値を得ます。

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

v^* : 電圧指令値　　ω^* : 速度指令値　　ω : 回転速度

$K_{P\omega}$: 速度PI比例ゲイン　　$K_{I\omega}$: 速度 PI 積分ゲイン　　s : ラプラス演算子

PI 制御の詳細については、専門書を参照してください。

3.1.4 PWM による電圧制御

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

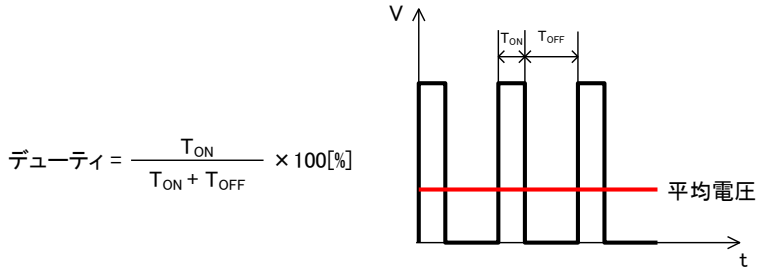


図 3-2 PWM 制御

また、変調率 m を以下のように定義します。

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

m:変調率　　V:指令値電圧　　E:インバータ母線電圧

この変調率を、PWM デューティを決めるレジスタに反映させることで所望の制御を行います。

また、サンプルソフトウェアでは、前半 60 度チョッピングを採用し、出力電圧及び速度の制御を行っています。図 3-3 に、非相補前半 60 度チョッピング時のモータ制御信号出力波形例を示します。図 3-4 に、相補前半 60 度チョッピング時のモータ制御信号出力波形例を示します。

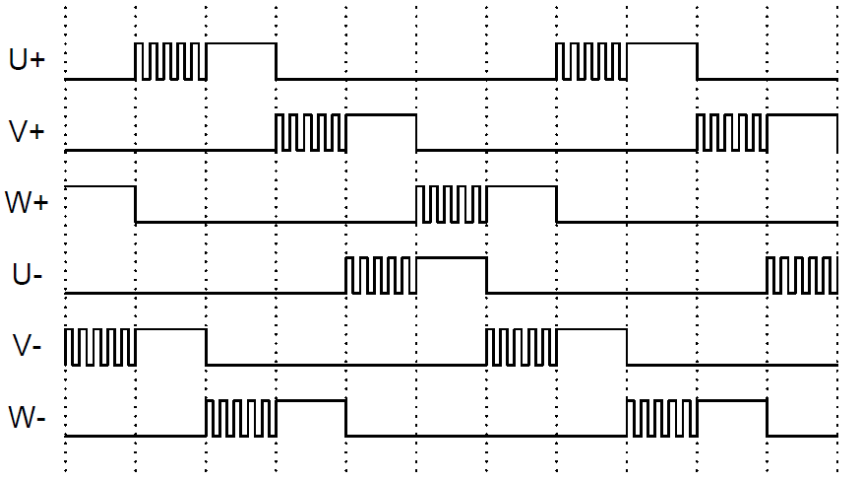


図 3-3　非相補前半 60 度チョッピング

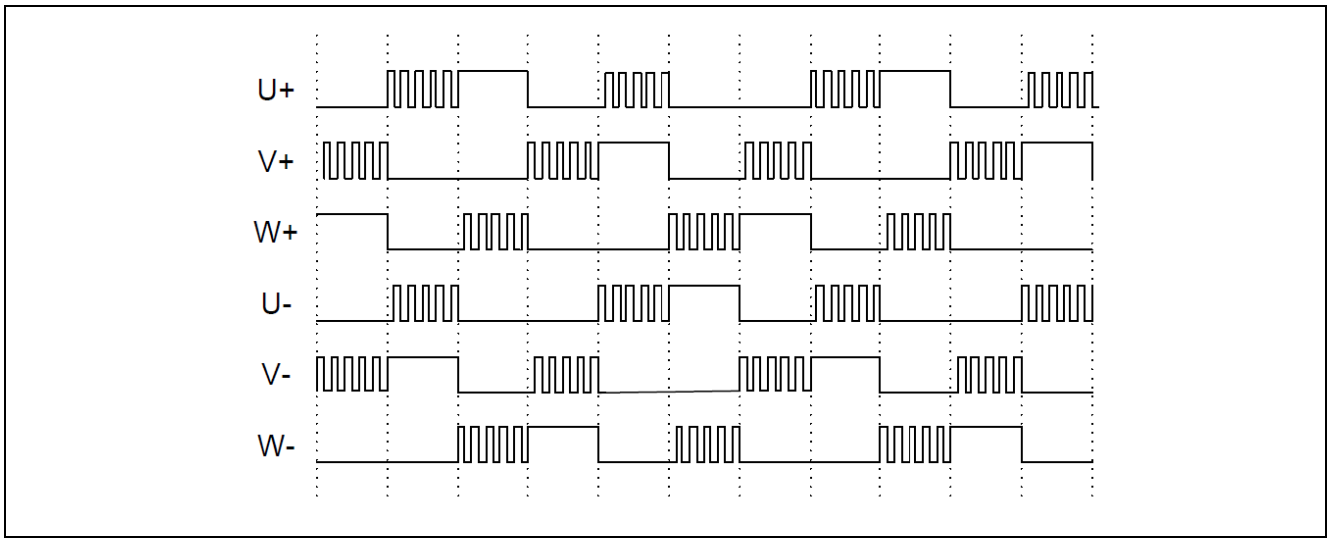
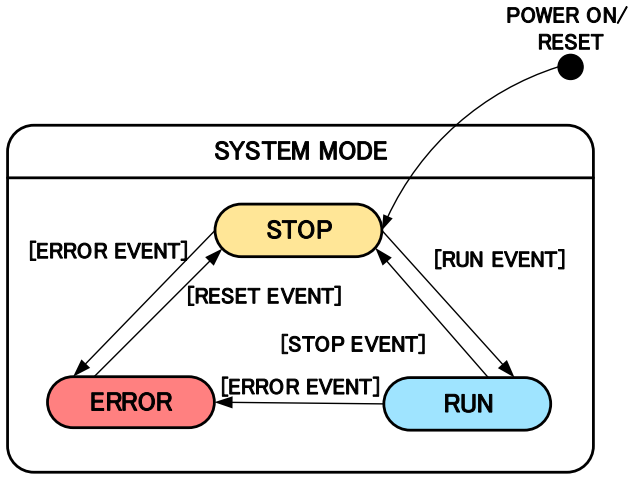


図 3-4　相補前半 60 度チョッピング

3.1.5 状態遷移

図 3-5 にサンプルソフトウェアにおける状態遷移図を示します。サンプルソフトウェアでは、「SYSTEM MODE」により状態を管理します。



		MODE		
		STOP	RUN	ERROR
EVENT	STOP	STOP	STOP	ERROR
	RUN	RUN	RUN	ERROR
	ERROR	ERROR	ERROR	ERROR
	RESET	STOP	RUN	STOP

図 3-5 センサレス 120 度通電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) 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 3-5 中の表の様に遷移します。各 EVENT の発生要因は下記となります。

表 3-5 EVENT 一覧

イベント名	発生要因
STOP	ユーザー操作により発生します
RUN	ユーザー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	ユーザー操作により発生します

3.1.6 センサレス制御時の始動方法

センサレス 120 度通電制御は、永久磁石(回転子)の磁束の変化による誘起電圧を利用し、60 度毎の磁極の位置を推定します。しかし、誘起電圧は回転することで発生するため、始動時は磁極の位置を推定することができず、磁極位置の推定に十分な回転速度が必要です。

そのため、始動方法として、永久磁石の位置にかかわらず、強制的に通電パターンを変化させることで回転磁界を発生させ、同期速度に引き込む方法があります。

図 3-6 ではサンプルソフトウェアでの始動方法を示しています。”MTR_MODE_BOOT”では、磁極位置の引き込みを行い、始動時の過電流を防いでいます。

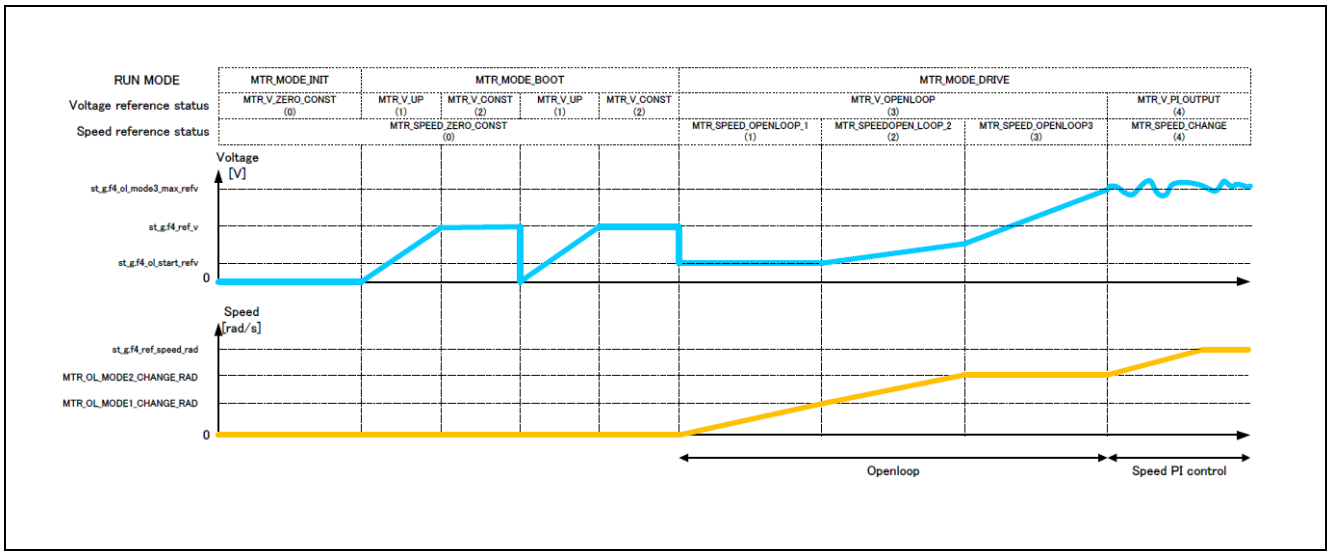


図 3-6 始動方法例

3.1.7 システム保護機能

サンプルソフトウェアは、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 3-6 を参照してください。

・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)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。

・ゼロクロス検出タイムアウトエラー

ゼロクロス検出によるパターン切り替えが一定時間発生しない場合、緊急停止します。

・仮想ホールセンサパターン(位置情報)の異常検出

U、V、W 各相電圧から作る仮想ホールセンサパターン(位置情報)の異常を検出した場合、緊急停止します。

表 3-6 各システム保護機能設定値

エラー	閾値	
過電流エラー	過電流リミット値 [A]	0.89
	監視周期 [μs]	50
過電圧エラー	過電圧リミット値 [V]	28
	監視周期 [μs]	50
低電圧エラー	低電圧リミット値 [V]	14
	監視周期 [μs]	50
回転速度エラー	速度リミット値 [rpm]	3000
	監視周期 [μs]	50
ゼロクロス検出タイムアウトエラー	タイムアウト時間[s]	2

3.1.8 AD トリガ

AD トリガとスキャングループのタイミングを示します。

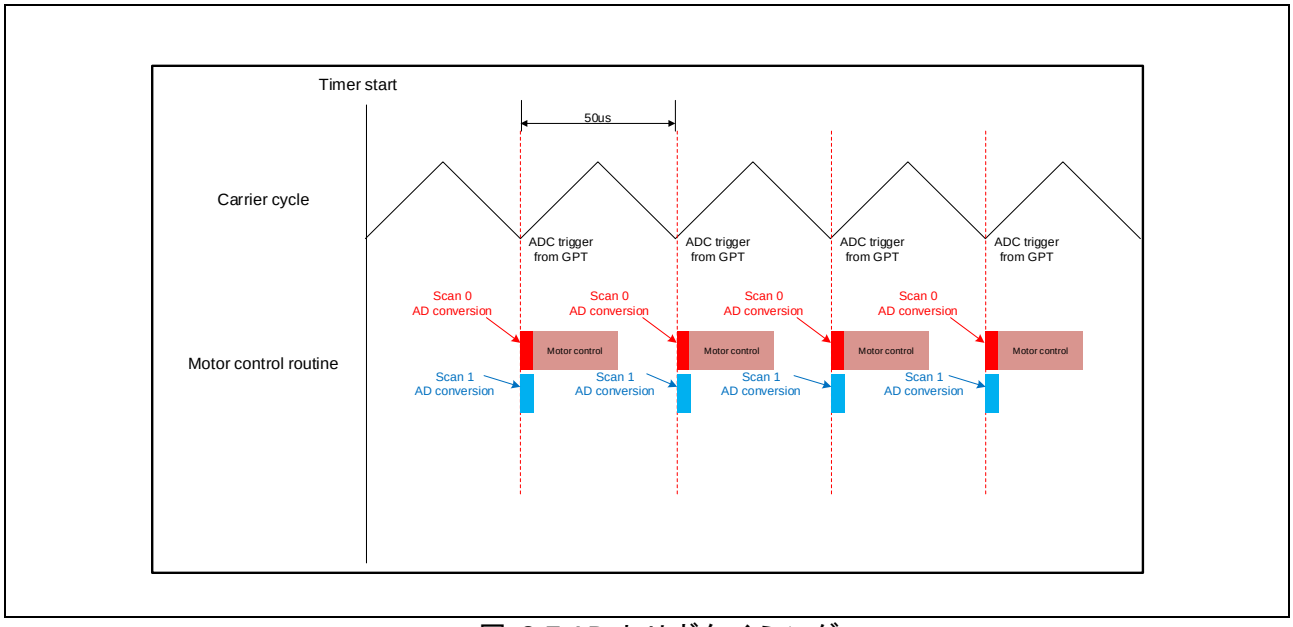


図 3-7 AD トリガタイミング

3.2 センサレス 120 度通電制御ソフトウェア関数仕様

ここでは、各割り込み周期毎に実行される関数についての仕様をまとめます。また各表には、センサレス 120 度通電制御における主要な関数のみ記載しています。各表に記載のない関数の詳細については、ソースコードを参照ください。

表 3-7 “mtr_main.c”関数一覧 [1/2]

ファイル名	関数名	処理概要
mtr_main.c	mtr_init 入力：なし 出力：なし	初期化処理
	mtr_main 入力：なし 出力：なし	メイン処理
	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
	ics_ui 入力：なし 出力：なし	Analyzer ユーザインタフェース使用
	software_init 入力：なし 出力：なし	メイン関数にて使用する変数の初期化
	g_poe_overcurrent 入力：(poeg_callback_args_t *) p_args / コールバック関数パラメータ 出力：なし	POEG 割込み処理
	motor_fsp_init 入力：なし 出力：なし	FSP モジュール初期化処理
	mtr_callback_120_degree 入力：(motor_callback_args_t *) p_args / コールバック関数パラメータ 出力：なし	120 度通電制御コールバック関数
	mtr_board_led_control 入力：(uint8_t) u1_motor_status / モータステータス 出力：なし	LED パターン設定処理
	mtr_remove_sw_chattering 入力：(uint8_t) u1_sw / SW 種別 (uint8_t) u1_on_off / ON/OFF 状態 出力：なし	SW チャタリング除去処理
	get_vr1 入力：なし 出力：なし	VR1 の状態を取得

表 3-8 “rm_motor_120_degree.c”関数一覧 [1/3]

ファイル名	関数名	処理概要
rm_motor_120_degree.c	RM_MOTOR_120_DEGREE_Open 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_cfg_t const * const) p_cfg / コンフィグパラメータ 出力：(fsp_err_t) err / 実行結果	120 度通電制御開始処理
	RM_MOTOR_120_DEGREE_Close 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	120 度通電制御終了処理
	RM_MOTOR_120_DEGREE_Reset 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	エラー状態のリセット処理
	RM_MOTOR_120_DEGREE_Run 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	モータの回転開始処理
	RM_MOTOR_120_DEGREE_Stop 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	モータの回転停止処理
	RM_MOTOR_120_DEGREE_ErrorSet 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_cfg_t const * const) p_cfg / コンフィグパラメータ 出力：(fsp_err_t) err / 実行結果	エラー状態の設定処理
	RM_MOTOR_120_DEGREE_SpeedSet 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ (float const) speed_rpm / 回転指示値[RPM] 出力：(fsp_err_t) err / 実行結果	モータの回転指示値設定処理
	RM_MOTOR_120_DEGREE_StatusGet 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ (uint8_t * const) p_status / モータ制御状態 出力：(fsp_err_t) err / 実行結果	モータ制御状態の取得
	RM_MOTOR_120_DEGREE_SpeedGet 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ (float * const) p_speed_rpm / 回転速度[RPM] 出力：(fsp_err_t) err / 実行結果	モータの回転速度取得処理
	RM_MOTOR_120_DEGREE_WaitStopFlagGet 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ (uint8_t * const) p_flg_wait_stop / モータ停止状態 出力：(fsp_err_t) err / 実行結果	モータ停止状態の取得
	RM_MOTOR_120_DEGREE_ErrorCheck 入力：(motor_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果 (uint16_t * const) p_error / エラー状態	エラーチェック処理

表 3-9 “rm_motor_120_degree.c”関数一覧 [2/3]

ファイル名	関数名	処理概要
rm_motor_120_degree.c	rm_motor_120_degree_active 入力：(motor_120_degree_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：(uint8_t) err / 実行結果	モータの回転開始処理
	rm_motor_120_degree_inactive 入力：(motor_120_degree_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：(uint8_t) err / 実行結果	モータの回転停止処理
	rm_motor_120_degree_nowork 入力：(motor_120_degree_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：(uint8_t) err / 実行結果	ブランク処理
	rm_motor_120_degree_reset 入力：(motor_120_degree_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：(uint8_t) err / 実行結果	エラー状態リセット処理
	rm_motor_120_degree_error 入力：(motor_120_degree_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：(uint8_t) err / 実行結果	エラー時の後処理
	rm_motor_120_degree_statemachine_init 入力：(motor_120_degree_statemachine_t *) p_state_machine / ステートマシン 出力：なし	ステートマシン初期化処理
	rm_motor_120_degree_statemachine_reset 入力：(motor_120_degree_statemachine_t *) p_state_machine / ステートマシン 出力：なし	ステートマシンリセット処理
	rm_motor_120_degree_statemachine_event 入力：(motor_120_degree_instance_ctrl_t *) p_ctrl / インスタンスパラメータ (motor_120_degree_ctrl_event_t) u1_event / イベント 出力：なし	状態遷移処理
	rm_motor_check_over_speed_error 入力：(float) f4_speed_rad / 回転速度[RPM] (float) f4_speed_limit_rad / 回転速度上限[RPM] 出力：(uint16_t) u2_temp0 / エラーフラグ	オーバースピードエラー検出処理
	rm_motor_check_over_voltage_error 入力：(float) f4_vdc / インバータ母線電圧値[V] (float) f4_overvoltage_limit / 上限電圧値[V] 出力：(uint16_t) u2_temp0 / エラーフラグ	過電圧エラー検出処理
	rm_motor_check_low_voltage_error 入力：(float) f4_vdc / インバータ母線電圧値[V] (float) f4_lowvoltage_limit / 下限電圧値[V] 出力：(uint16_t) u2_temp0 / エラーフラグ	低電圧エラー検出処理

表 3-10 “rm_motor_120_degree.c”関数一覧 [3/3]

ファイル名	関数名	処理概要
rm_motor_120_degree.c	rm_motor_check_over_current_error 入力：(float) f4_iu / U 相電流[A] (float) f4_iv / V 相電流[A] (float) f4_iw / W 相電流[A] (float) f4_oc_limit / 上限電流値[A] 出力：(uint16_t) u2_temp0 / エラーフラグ	過電流エラー検出処理
	rm_motor_120_degree_error_check 入力：(motor_120_degree_instance_ctrl_t *) p_ctrl / インスタンスパラメータ (float) f_iu / U 相電流[A] (float) f_iv / V 相電流[A] (float) f_iw / W 相電流[A] (float) f_vdc / インバータ母線電圧[V] (float) f_speed / 回転速度[RPM] 出力：(uint16_t) u2_error_flags / エラーフラグ	エラー検出処理
	rm_motor_120_degree_120_control_callback 入力：(motor_120_control_callback_args_t *) p_args / コールバック関数パラメータ 出力：なし	120contorol モジュールコールバック処理

表 3-11 “rm_motor_120_control_sensorless.c”関数一覧 [1/4]

ファイル名	関数名	処理概要
rm_motor_120_control_sensorless.c	RM_MOTOR_120_CONTROL_SENSORLESS_Open 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_control_cfg_t const * const) p_cfg / コンフィグパラメータ 出力：(fsp_err_t) err / 実行結果	センサレス制御開始処理
	RM_MOTOR_120_CONTROL_SENSORLESS_Close 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	センサレス制御終了処理
	RM_MOTOR_120_CONTROL_SENSORLESS_Run 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	モータ回転開始処理(センサレス制御)
	RM_MOTOR_120_CONTROL_SENSORLESS_Stop 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	モータの回転停止処理(センサレス制御)
	RM_MOTOR_120_CONTROL_SENSORLESS_Reset 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	エラー状態のリセット処理
	RM_MOTOR_120_CONTROL_SENSORLESS_SpeedSet 入力：(motor_120_control_ctrl_t* const) p_ctrl / インスタンスパラメータ (float * const) p_speed_rpm / 回転指示値[RPM] 出力：(fsp_err_t) err / 実行結果	モータの回転指示値設定処理
	RM_MOTOR_120_CONTROL_SENSORLESS_SpeedGet 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (float * const) p_speed_rpm / 回転速度[RPM] 出力：(fsp_err_t) err / 実行結果	モータの回転速度取得処理
	RM_MOTOR_120_CONTROL_SENSORLESS_CurrentGet 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_driver_current_status_t * const) p_current_status / 電流電圧データ 出力：(fsp_err_t) err / 実行結果	電流電圧データ取得処理
	RM_MOTOR_120_CONTROL_SENSORLESS_WaitStopFlagGet 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_control_wait_stop_flag_t * const) p_flg_wait_stop / モータ停止状態 出力：(fsp_err_t) err / 実行結果	モータ停止状態の取得処理
	RM_MOTOR_120_CONTROL_SENSORLESS_TimeoutErrorFlagGet 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_control_timeout_error_flag_t * const) p_timeout_error_flag / ゼロクロス検出タイムアウトエラー状態 出力：(fsp_err_t) err / 実行結果	ゼロクロス検出タイムアウトエラー状態取得処理

表 3-12 “rm_motor_120_control_sensorless.c”関数一覧 [2/4]

ファイル名	関数名	処理概要
rm_motor_120_control_sensorless.c	RM_MOTOR_120_CONTROL_SENSORLESS_PatternErrorFlagGet 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_control_pattern_error_flag_t * const) p_pattern_error_flag / 仮想ホールパターンエラー状態 出力：(fsp_err_t) err / 実行結果	仮想ホールセンサパターンエラー状態取得処理
	RM_MOTOR_120_CONTROL_SENSORLESS_VoltageRefGet 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_control_voltage_ref_t * const) p_voltage_ref / 電圧設定状態 出力：(fsp_err_t) err / 実行結果	電圧設定状態取得
	RM_MOTOR_120_CONTROL_SENSORLESS_ParameterUpdate 入力：(motor_120_control_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_control_cfg_t const * const) p_cfg / コンフィグパラメータ 出力：(fsp_err_t) err / 実行結果	パラメータアップデート処理
	rm_motor_120_control_sensorless_speed_cyclic 入力：(timer_callback_args_t *) p_args / コールバック関数パラメータ 出力：なし	速度制御用コールバック関数
	rm_motor_120_control_sensorless_driver_callback 入力：(motor_120_driver_callback_args_t *) p_args / コールバック関数パラメータ 出力：なし	A/D 変換完了コールバック関数
	rm_motor_120_control_sensorless_reset 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	インスタンスパラメータリセット処理
	rm_motor_120_control_sensorless_start_openloop 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	オープンループ開始処理
	rm_motor_120_control_sensorless_ol_signal_set 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	オープンループ時の通電パターン設定処理
	rm_motor_120_control_sensorless_speed_calc 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	回転速度算出処理
	rm_motor_120_control_sensorless_generate_pattern 入力：(float) vu_ad / U 相電圧 (float) vv_ad / V 相電圧 (float) vw_ad / W 相電圧 (float) vn_ad / 中点電圧 出力：(uint8_t) u1_temp / 通電パターン	通電パターン生成処理

表 3-13 “rm_motor_120_control_sensorless.c”関数一覧 [3/4]

ファイル名	関数名	処理概要
rm_motor_120_control_sensorless.c	rm_motor_120_control_sensorless_set_angle_shift 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	位相シフト量の算出
	rm_motor_120_control_sensorless_check_pattern 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	ゼロクロス判定処理
	rm_motor_120_control_sensorless_shift_angle 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	位相シフト処理
	rm_motor_120_control_sensorless_wait_motorstop 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	回転停止チェック処理
	rm_motor_120_control_sensorless_pattern_set 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	通電パターン設定処理
	rm_motor_120_control_sensorless_pattern_first60 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ (uint8_t) u1_signal / 通電パターン 出力：なし	非相補前半 60 度チョッピング処理
	rm_motor_120_control_sensorless_pattern_first60_comp 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ (uint8_t) u1_signal / 通電パターン 出力：なし	相補前半 60 度チョッピング処理
	rm_motor_120_control_sensorless_speed_ref_set 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	速度制御用指令値設定処理
	rm_motor_120_control_sensorless_voltage_ref_set 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	電圧指令値設定処理
	rm_motor_120_control_sensorless_pi_ctrl 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	PI 制御処理
	rm_motor_120_control_sensorless_check_timeout_error 入力：(motor_120_control_sensorless_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	仮想ホールセンサパターンエラー判定処理

表 3-14 “rm_motor_120_control_sensorless.c”関数一覧 [4/4]

ファイル名	関数名	処理概要
rm_motor_120_control_sensorless.c	rm_motor_120_control_sensorless_lpff 入力：(float) f4_lpf_input / LPF 入力値 (float) f4_pre_lpf_output / 前回の LPF 出力値 (float) f4_lpf_k / LPF ゲイン 出力：(float) f4_temp / LPF 出力値	LPF 処理
	rm_motor_120_control_sensorless_limitf 入力：(float) f4_value / 入力値 (float) f4_max / 最大値 (float) f4_min / 最小値 出力：(float) f4_temp / 出力値	上下限リミット処理
	rm_motor_120_control_sensorless_limitf_h 入力：(float) f4_value / 入力値 (float) f4_max / 最大値 出力：(float) f4_temp / 出力値	上限リミット処理
	rm_motor_120_control_sensorless_limitf_l 入力：(float) f4_value / 入力値 (float) f4_min / 最小値 出力：(float) f4_temp / 出力値	下限リミット処理
	rm_motor_120_control_sensorless_limitf_abs 入力：(float) f4_value / 入力値 (float) f4_limit_value / リミット値 出力：(float) f4_temp / 出力値	絶対値リミット処理

表 3-15 “rm_motor_120_driver.c”関数一覧 [1/2]

ファイル名	関数名	処理概要
rm_motor_120_driver.c	RM_MOTOR_120_DRIVER_Open 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_driver_cfg_t const * const) p_cfg / コンフィグパラメータ 出力：(fsp_err_t) err / 実行結果	120 度通電制御用ドライバ開始処理
	RM_MOTOR_120_DRIVER_Close 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	120 度通電制御用ドライバ終了処理
	RM_MOTOR_120_DRIVER_Run 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	モータ回転開始処理
	RM_MOTOR_120_DRIVER_Stop 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	モータの回転停止処理
	RM_MOTOR_120_DRIVER_Reset 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	エラー状態のリセット処理
	RM_MOTOR_120_DRIVER_PhaseVoltageSet 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ (float const) u_voltage / U 相 PWM デューティ (float const) v_voltage / V 相 PWM デューティ (float const) w_voltage / W 相 PWM デューティ 出力：(fsp_err_t) err / 実行結果	PWM デューティ設定処理
	RM_MOTOR_120_DRIVER_PhasePatternSet 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_driver_phase_pattern_t const) pattern / 通電パターン 出力：(fsp_err_t) err / 実行結果	PWM 出力状態切り替え処理
	RM_MOTOR_120_DRIVER_CurrentGet 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_driver_current_status_t * const) p_current_status / 電流電圧データ 出力：(fsp_err_t) err / 実行結果	電流電圧データ取得処理
	RM_MOTOR_120_DRIVER_CurrentOffsetCalc 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	A/D 値のオフセット算出処理
	RM_MOTOR_120_DRIVER_FlagCurrentOffsetGet 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ 出力：(fsp_err_t) err / 実行結果	A/D 値のオフセット算出状態取得処理

表 3-16 “rm_motor_120_driver.c”関数一覧 [2/2]

ファイル名	関数名	処理概要
rm_motor_120_driver.c	RM_MOTOR_120_DRIVER_ParameterUpdate 入力：(motor_120_driver_ctrl_t * const) p_ctrl / インスタンスパラメータ (motor_120_driver_cfg_t const * const) p_cfg / コンフィグパラメータ 出力：(fsp_err_t) err / 実行結果	パラメータアップデート処理
	rm_motor_120_driver_reset 入力：(motor_120_driver_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	インスタンスパラメータリセット処理
	rm_motor_120_driver_output_pwm 入力：(motor_120_driver_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	通常 PWM 出力開始処理
	rm_motor_120_driver_ctrl_start 入力：(motor_120_driver_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	モータ制御用 PWM 出力開始処理
	rm_motor_120_driver_ctrl_stop 入力：(motor_120_driver_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	PWM 出力停止処理
	rm_motor_120_driver_set_uvw_duty 入力：(motor_120_driver_instance_ctrl_t *) p_ctrl / インスタンスパラメータ (float) f_duty_u / U 相 PWM デューティ (float) f_duty_v / V 相 PWM デューティ (float) f_duty_w / W 相 PWM デューティ 出力：なし	PWM デューティ設定処理
	rm_motor_120_driver_current_get 入力：(motor_120_driver_instance_ctrl_t *) p_ctrl / インスタンスパラメータ 出力：なし	A/D 値取得処理
	rm_motor_120_driver_mod_set_max_duty 入力：(motor_120_driver_modulation_t *) p_mod/ PWM 設定 (float) f4_max_duty / 最大デューティ 出力：なし	最大デューティ設定処理
	rm_motor_120_driver_mod_set_min_duty 入力：(motor_120_driver_modulation_t *) p_mod/ PWM 設定 (float) f4_max_duty / 最小デューティ 出力：なし	最小デューティ設定処理
	rm_motor_120_driver_pin_cfg 入力：(bsp_io_port_pin_t) pin / ピン番号 (uint32_t) cfg / 設定値 出力：なし	ピンコンフィグレーション設定処理
	rm_motor_120_driver_cyclic 入力：(adc_callback_args_t *) p_args / コールバック関数パラメータ 出力：なし	A/D 変換完了コールバック関数

3.3 Contents of control

3.3.1 Configuration Options

モータ用センサレス 120 度通電制御モジュールの構成オプションは、RA Configurator を使用して構成できます。 変更されたオプションは、コードの生成時に hal_data.c に自動的に反映されます。 オプション名と設定値は、次の表 3-17 構成オプションに記載しています。

表 3-17 構成オプション

Configuration Options (rm_motor_120_degree.h)	
オプション名	内容
Limit of over current (A) Initial: 0.42	相電流がこの値を超えると、PWM 出力ポートがオフに設定されます。
Limit of over voltage (V) Initial: 28.0	母線電圧がこの値を超えると、PWM 出力ポートがオフに設定されます。
Limit of over speed (rpm) Initial: 3000.0	回転速度がこの値を超えると、PWM 出力ポートがオフに設定されます。
Limit of over speed (rpm) Initial: 14.0	母線電圧がこの値を下回ると、PWM 出力ポートがオフに設定されます。

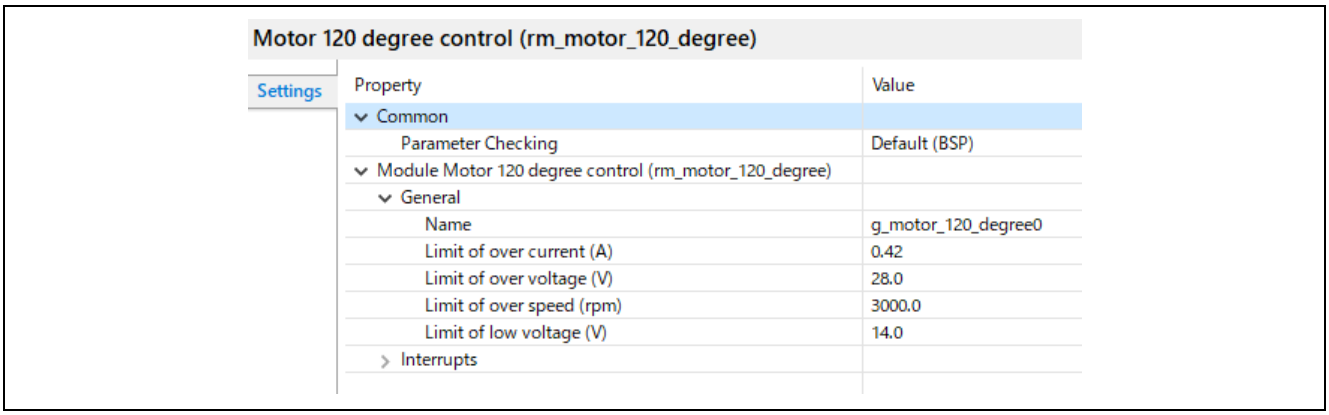


図 3-8 120 度通電制御の FSP コンフィグレーション(FSP3.5.0)

3.3.2 Configuration Options for included modules

モータ用センサレス 120 度通電制御モジュールには、以下のモジュールが含まれます。

・120 control sensorless module

・120 driver module

また、これらのモジュールには、センサレス 120 度通電制御モジュールと同じ各構成パラメータがあります。 オプション名と設定値を以下の表に示します。

表 3-18 構成オプション

Configuration Options (rm_motor_120_control_api.h)	
オプション名	内容
Conduction type Initial: First 60 degree PWM	前半 60 度チョッピング制御切り替え
Timeout counts (msec) Initial: 2000	停止判定カウンタ [ms]
Maximum voltage (V) Initial: 20.0	最大指令電圧 [V]
Minimum voltage (V) Initial: 3.0	最小指令電圧 [V]
Speed PI decimation Initial: 0	速度 PI 制御用割り込み間引き数
Free run timer frequency (MHz) Initial: 120	フリーランタイマ周波数 [Mhz]
Speed LPF K Initial: 1.0	速度 LPF パラメータ
Step of speed change Initial: 0.2	速度指令最大増減幅
PI control KP Initial: 0.02	速度 PI 比例ゲイン
PI control KI Initial: 0.0005	速度 PI 積分ゲイン
PI control limit Initial: 24.0	電圧 PI 制御積分項リミット値 [V]
Motor Parameter | Pole pairs Initial: 2	極対数

表 3-19 構成オプション

Configuration Options (rm_motor_120_control_sensorless.h)	
オプション名	内容
Stop BEMF Initial: 0.5	センサレス駆動停止判定電圧
Maximum voltage for BOOT (V) Initial: 8.0	BOOT モード最大電圧 [V]
Carrier frequency (kHz) Initial: 20.0	キャリア周波数 [khz]
Adjusting angle Initial: 0	パターン切り替えタイミング調整値
Boot reference voltage (V) Initial: 3.0	電圧指令値 [V]
Voltage lamping time Initial: 128	電圧指令値加算カウント回数
Voltage constant adjust time Initial: 64	電圧指令値一定カウント回数
Open loop start speed (rpm) Initial: 150	起動速度 [rpm]
Open loop mode2 speed (rpm) Initial: 185	オープンループモード 1 切り替え速度 [rpm]
Open loop mode3 speed (rpm) Initial: 1000	オープンループモード 2 切り替え速度 [rpm]
Open loop start voltage (V) Initial: 3.0	起動電圧 [V]
Open loop mode1 speed rate Initial: 0.25	オープンループモード 1 での速度増加率 [rpm/control period]
Open loop mode2 voltage rate Initial: 0.00285	オープンループモード 2 での電圧増加率 [V/control period]
Open loop mode2 speed rate Initial: 0.71	オープンループモード 2 での速度増加率 [rpm/control period]
Open loop mode3 voltage rate Initial: 0.002	オープンループモード 3 での電圧増加率 [V/control period]
Open loop maximum voltage (V) Initial: 6.5	オープンループモード 3 での最大電圧 [V]

120-degree conduction control sensorless (rm_motor_120_control_sensorless)

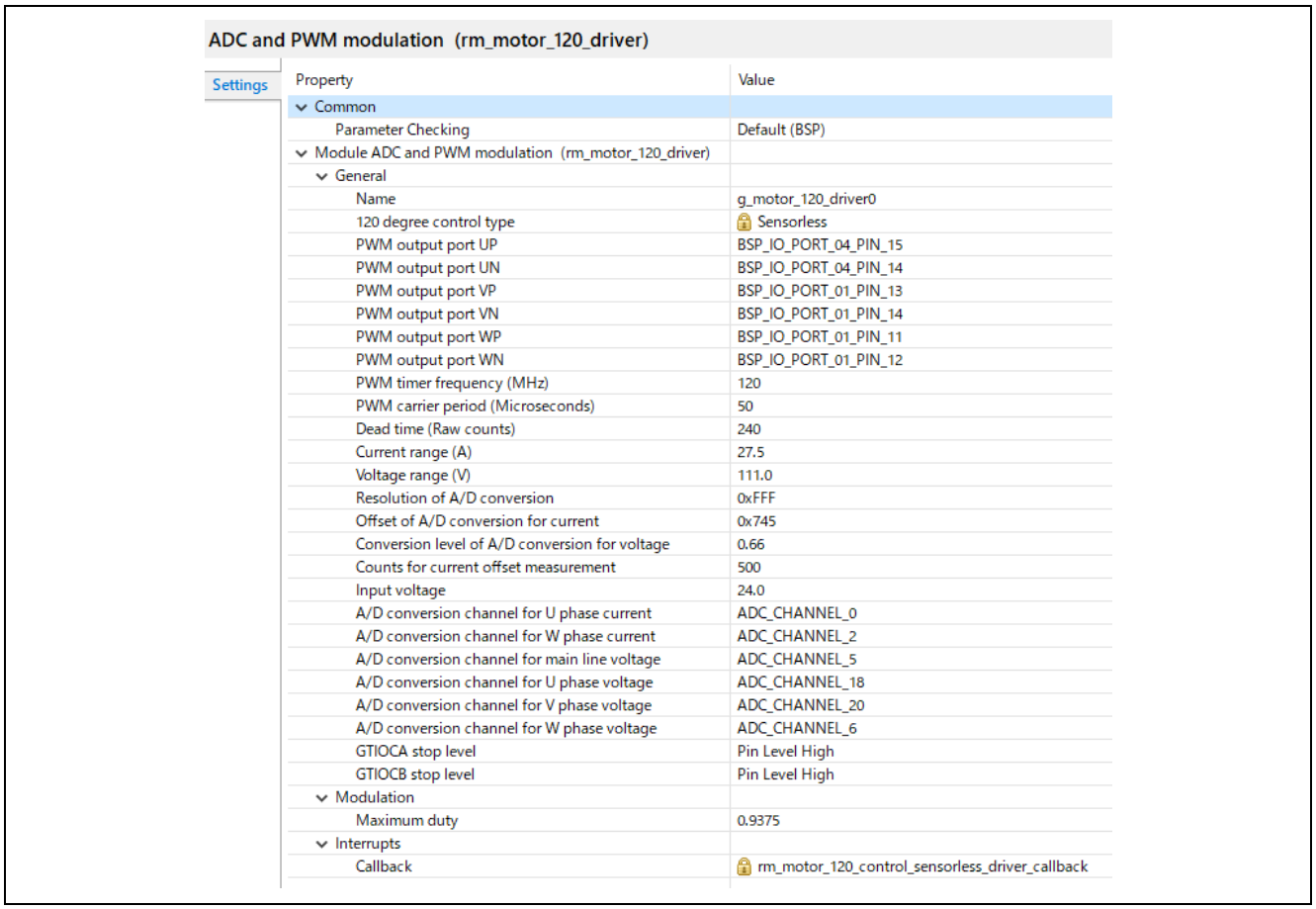
Settings

Property	Value
Common	
Parameter Checking	Default (BSP)
Module 120-degree conduction control sensorless (rm_motor_	
General	
Name	g_motor_120_control_sensorless0
Conduction type	First 60 degree PWM
Stop BEMF	0.5
Timeout counts (msec)	2000
Maximum voltage for BOOT (V)	8.0
Maximum voltage (V)	20.0
Minimum voltage (V)	3.0
Carrier frequency (kHz)	20.0
Adjusting angle	0
Speed PI decimation	0
Free run timer frequency (MHz)	120
Speed LPF K	0.5
Step of speed change	0.2
Boot reference voltage (V)	3.0
Voltage lamping time	128
Voltage constant adjust time	64
Open loop start speed (rpm)	150
Open loop mode2 speed (rpm)	185
Open loop mode3 speed (rpm)	1000
Open loop start voltage (V)	3.0
Open loop mode1 speed rate	0.25
Open loop mode2 voltage rate	0.00285
Open loop mode2 speed rate	0.71
Open loop mode3 voltage rate	0.002
Open loop maximum voltage (V)	6.5
PI control KP	0.02
PI control KI	0.0005
PI control limit	24.0
Motor Parameter	
Pole pairs	2
Resistance (ohm)	6.447
Inductance of d-axis (H)	0.0045
Inductance of q-axis (H)	0.0045
Permanent magnetic flux (Wb)	0.02159
Rotor inertia (kgm^2)	1.8
Interrupts	
Callback	rm_motor_120_degree_120_control_callback

図 3-9 センサレス 120 度通電制御の FSP コンフィグレーション(FSP3.5.0)

表 3-20 構成オプション

Configuration Options (rm_motor_120_driver.h)	
オプション名	内容
120 degree control type Initial: Sensorless	120 度通電制御センサタイプ
PWM output port UP Initial: BSP_IO_PORT_04_PIN_15	PWM 出力(Up)ポート
PWM output port UN Initial: BSP_IO_PORT_04_PIN_14	PWM 出力(Un)ポート
PWM output port VP Initial: BSP_IO_PORT_01_PIN_13	PWM 出力(Vp)ポート
PWM output port VN Initial: BSP_IO_PORT_01_PIN_14	PWM 出力(Vn)ポート
PWM output port WP Initial: BSP_IO_PORT_01_PIN_11	PWM 出力(Wp)ポート
PWM output port WN Initial: BSP_IO_PORT_01_PIN_12	PWM 出力(Wn)ポート
PWM timer frequency (MHz) Initial: 120	PWM タイマ周波数 [Mhz]
PWM carrier period (Microseconds) Initial: 50	PWM キャリア周波数 [Micro seconds]
Dead time (Raw counts) Initial: 240	デッドタイムカウント [Raw counts]
Current range (A) Initial: 27.5	電流検出レンジ [A]
Voltage range (V) Initial: 111.0	電圧検出レンジ [V]
Resolution of A/D conversion Initial: 0xFFF	A/D 変換値
Offset of A/D conversion for current Initial: 0x745	A/D 変換オフセット
Conversion level of A/D conversion for voltage Initial: 0.66	電圧 A/D 変換率
Counts for current offset measurement Initial: 500	オフセット値計算回数
Input voltage Initial: 24.0	母線電圧
A/D conversion channel for U phase current Initial: ADC_CHANNEL_0	U 相電流検出チャネル
A/D conversion channel for W phase current Initial: ADC_CHANNEL_2	W 相電流検出チャネル
A/D conversion channel for main line voltage Initial: ADC_CHANNEL_5	母線電圧検出チャネル
A/D conversion channel for U phase voltage Initial: ADC_CHANNEL_18	U 相電圧検出チャネル
A/D conversion channel for V phase voltage Initial: ADC_CHANNEL_20	V 相電圧検出チャネル
A/D conversion channel for W phase voltage Initial: ADC_CHANNEL_6	W 相電圧検出チャネル
GTIOCA stop level Initial: Pin Level High	上アーム停止時レベル
GTIOCB stop level Initial: Pin Level High	下アーム停止時レベル
Modulation | Maximum duty Initial: 0.9375	PWM 最大デューティ



ADC and PWM modulation (rm_motor_120_driver)

Settings

Property	Value
Common	
Parameter Checking	Default (BSP)
Module ADC and PWM modulation (rm_motor_120_driver)	
General	
Name	g_motor_120_driver0
120 degree control type	Sensorless
PWM output port UP	BSP_IO_PORT_04_PIN_15
PWM output port UN	BSP_IO_PORT_04_PIN_14
PWM output port VP	BSP_IO_PORT_01_PIN_13
PWM output port VN	BSP_IO_PORT_01_PIN_14
PWM output port WP	BSP_IO_PORT_01_PIN_11
PWM output port WN	BSP_IO_PORT_01_PIN_12
PWM timer frequency (MHz)	120
PWM carrier period (Microseconds)	50
Dead time (Raw counts)	240
Current range (A)	27.5
Voltage range (V)	111.0
Resolution of A/D conversion	0xFFF
Offset of A/D conversion for current	0x745
Conversion level of A/D conversion for voltage	0.66
Counts for current offset measurement	500
Input voltage	24.0
A/D conversion channel for U phase current	ADC_CHANNEL_0
A/D conversion channel for W phase current	ADC_CHANNEL_2
A/D conversion channel for main line voltage	ADC_CHANNEL_5
A/D conversion channel for U phase voltage	ADC_CHANNEL_18
A/D conversion channel for V phase voltage	ADC_CHANNEL_20
A/D conversion channel for W phase voltage	ADC_CHANNEL_6
GTIOCA stop level	Pin Level High
GTIOCB stop level	Pin Level High
Modulation	
Maximum duty	0.9375
Interrupts	
Callback	rm_motor_120_control_sensorless_driver_callback

図 3-10 120 度通電制御用 ADC と PWM モジュレーションドライバの FSP コンフィグレーション (FSP3.5.0)

3.4 制御フロー（フローチャート）

3.4.1 メイン処理

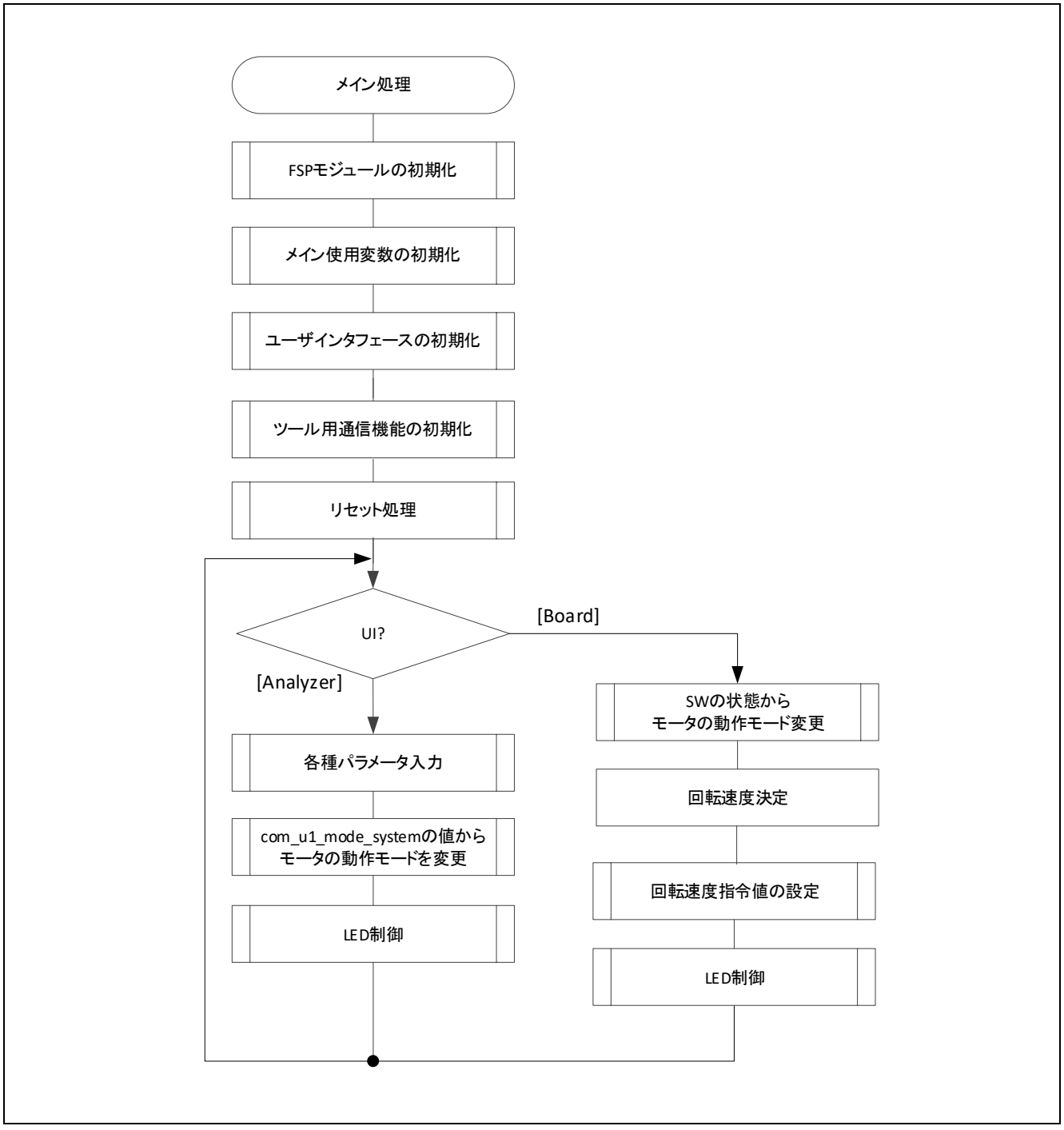


図 3-11 メイン処理フローチャート

3.4.2 50 [µs]周期割り込み(キャリア周期割り込み)処理

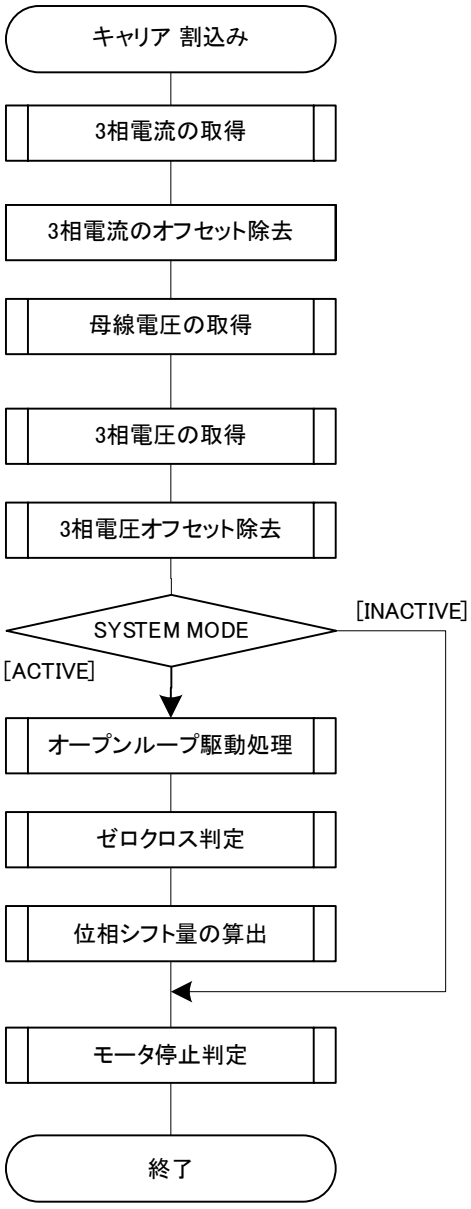


図 3-12 50[us]周期割り込み処理フローチャート

3.4.3 1 [ms]周期割り込み処理

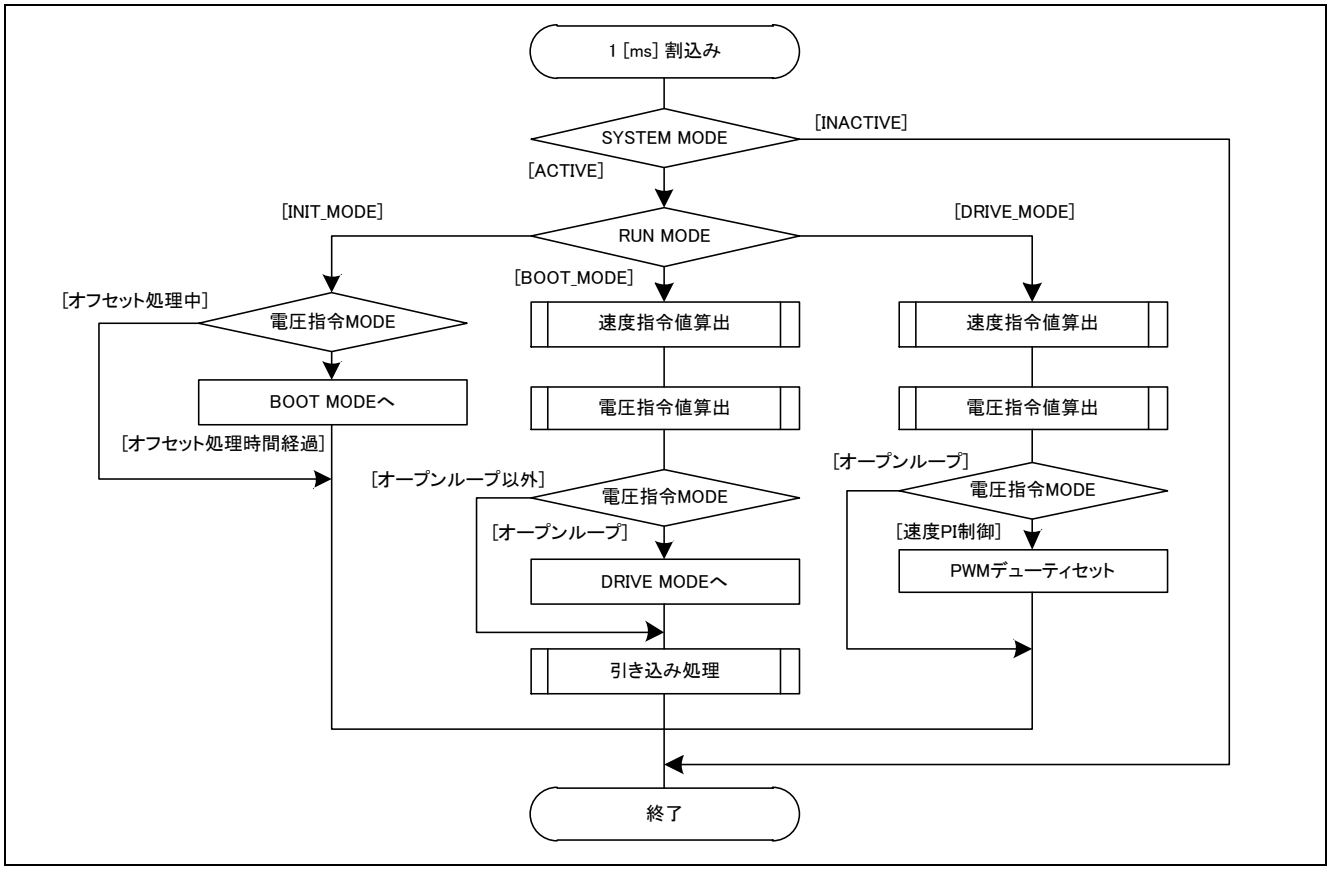


図 3-13 1 [ms]周期割り込み処理フローチャート

3.4.4 過電流検出割り込み処理

過電流検出割り込みは、サンプルソフトウェアにおける PWM 出力端子のハイインピーダンス制御条件である GTETRGC 出力レベル比較動作による出力短絡検出時に発生する割り込みです。そのため、本割り込み処理の実行開始時点では既に PWM 出力端子はハイインピーダンス状態になっており、モータへの出力は停止しています。

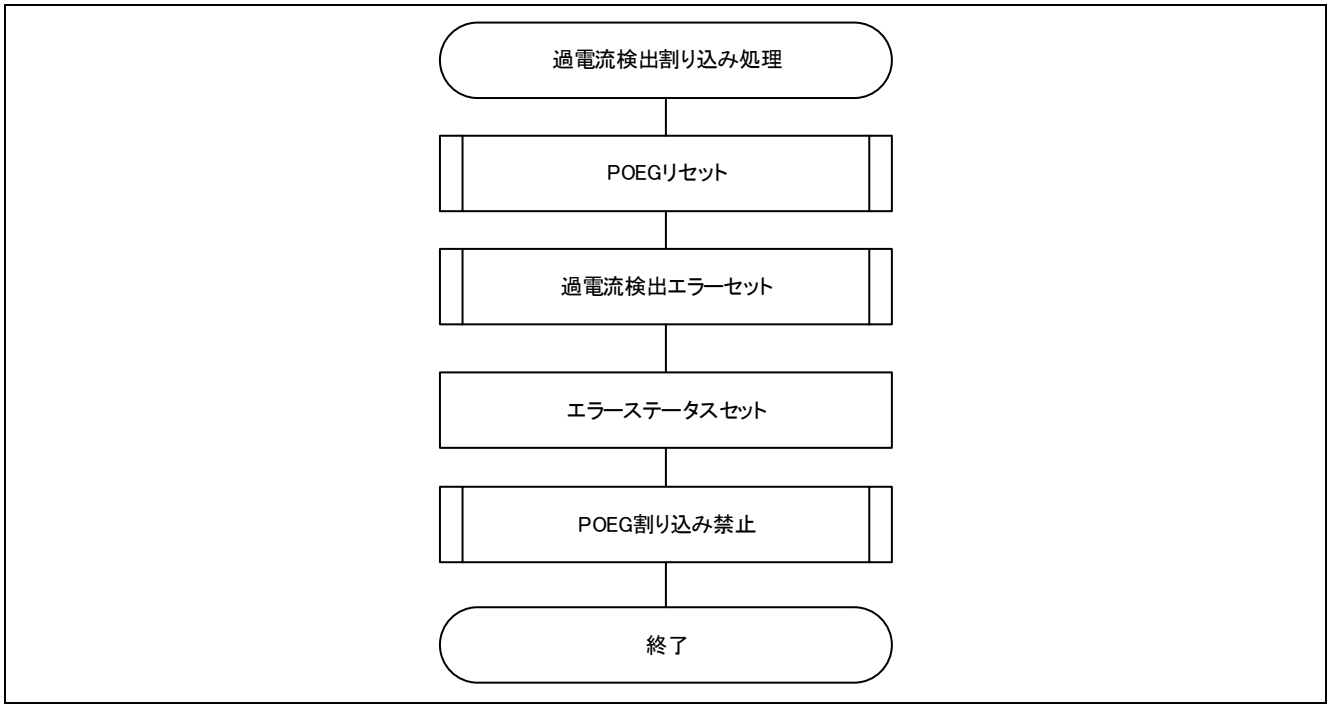


図 3-14 過電流検出割り込み処理フローチャート

4. 評価環境説明

サンプルソフトウェアの評価環境について説明します。

4.1 動作環境

表 4-1 に、サンプルソフトウェアをビルドおよびデバッグするためのハードウェア要件を示します。

表 4-1 センサレス 120 度通電制御ハードウェア基本仕様

項目	内容
インバータボード	RA6T1-RSSK [RTK0EM0000B10020BJ]
CPU ボード	RA6T1 CPU Card [RTK0EM0013C01201BJ]
モータ	ブラシレス DC モータ (TG-55L-KA 24V)
オンチップデバッグエミュレータ	RA6T1 CPU カードには、オンボードデバッガ（J-Link OB）が搭載されているため、エミュレータを用意する必要はありません。

表 4-2 に、サンプルソフトウェアをビルドおよびデバッグするためのソフトウェア要件を示します。

表 4-2 センサレス 120 度通電制御ソフトウェア基本仕様

項目		バージョン	内容
GCC	e2 studio	2021-10	Integrated development environment (IDE) for Renesas devices.
	GCC ARM Embedded	V10.3.1.20210824	C/C++ Compiler. (Download from e2 studio installer)
	Renesas Flexible Software Package (FSP)	V3.5.0 以降	Software package for writing applications for the RA microcontroller series.

4.2 プロジェクトのインポート

サンプルソフトウェアは、以下の手順で e2 studio にインポートできます。

1. File → Import

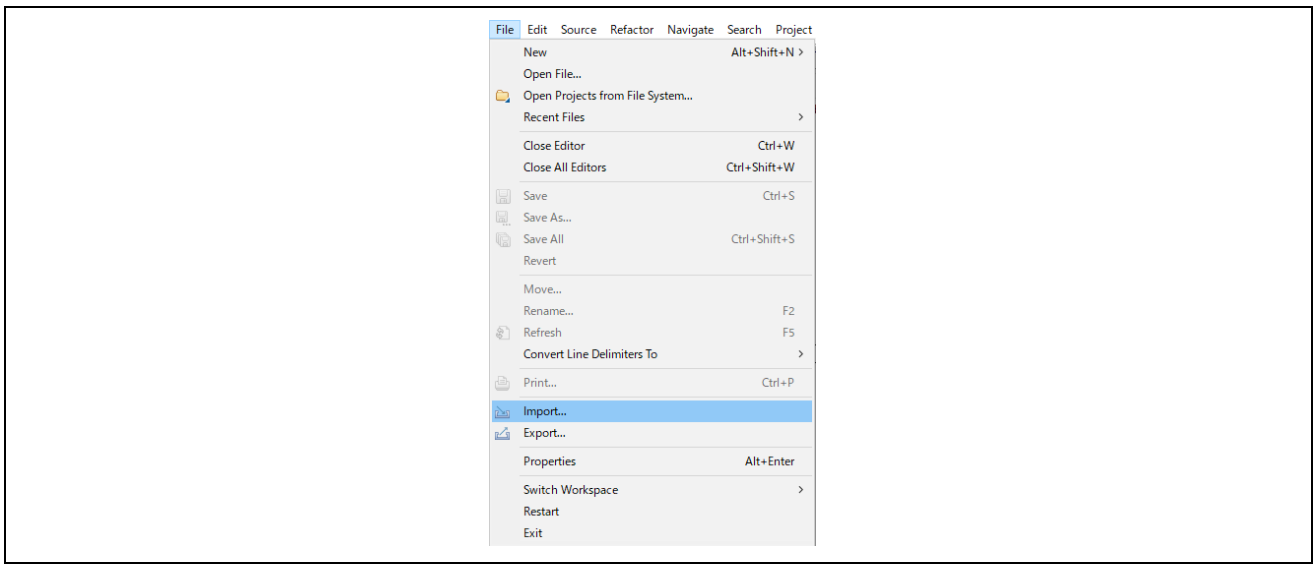


図 4-1 ファイルメニュー

2. 「Existing Projects into Workspace」を選択し、[次へ]ボタンをクリックします。

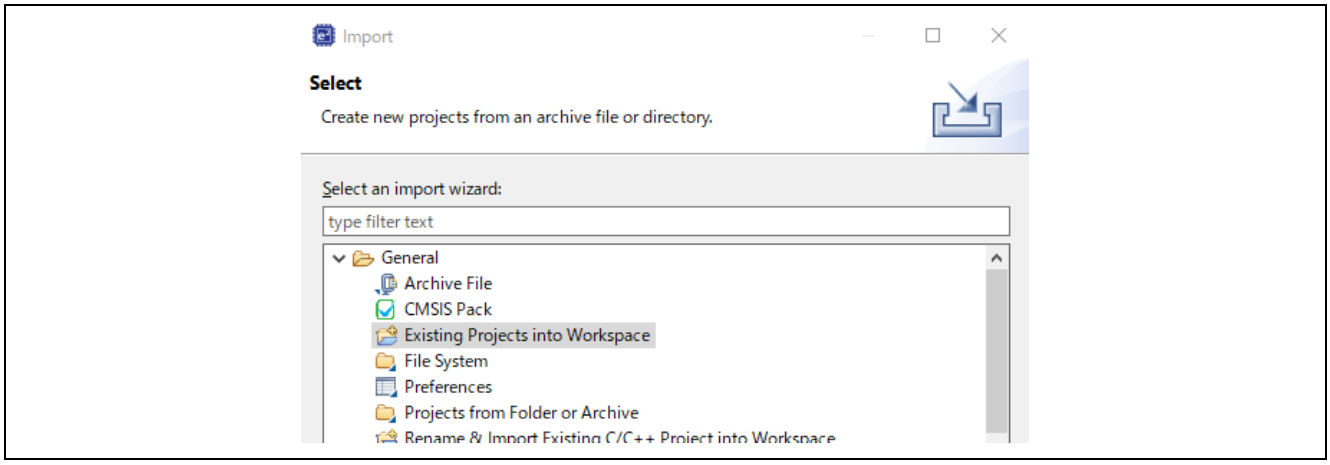


図 4-2 インポートメニュー

3. プロジェクトファイルを選択します。 Finish ボタンをクリックすると、プロジェクトがインポートされます。

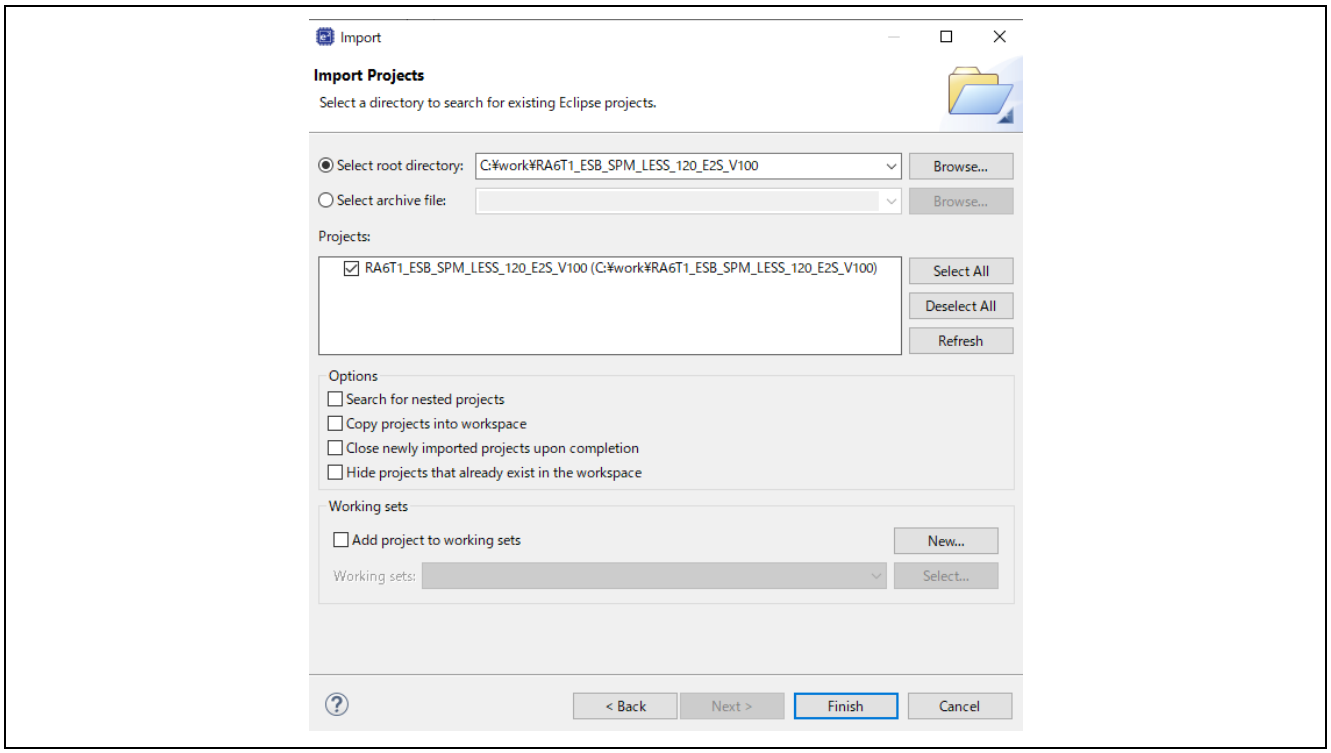


図 4-3　プロジェクトのインポート

4.3　ビルドとデバッグ

「e2 studio ユーザーズマニュアル入門ガイド（R20UT4204）」を参照してください。

4.4 サンプルソフトウェアの操作概要

4.4.1 クイックスタート

モータ制御開発支援ツール「Renesas Motor Workbench」を使用せずにサンプルソフトウェアを動作させる場合、下記の手順でクイックスタートサンプルプロジェクトを実行します。

(1) 安定化電源投入後、またはリセット後はインバータボード上の LED1、LED2 はすべて消灯状態で、モータは停止しています。

(2) インバータボード上のトグルスイッチ(SW1)を ON にするとモータが回転します。トグルスイッチ(SW1)を切り替えるごとにモータの回転開始/停止を繰り返します。モータが正常に回転している場合はインバータボード上の LED1 が点灯します。このとき、インバータボード上の LED2 が点灯している場合はエラーが発生しています。

(3) モータの回転方向を変更する場合は、インバータボード上のボリューム抵抗(VR1)で調整します。

- ボリューム抵抗(VR1)を右に回す：モータが時計回りに回転
- ボリューム抵抗(VR1)を左に回す：モータが反時計回りに回転

(4) エラーが発生した場合、インバータボード上の LED2 が点灯し、回転が停止します。復帰するためにはインバータボードのトグルスイッチ(SW1)を OFF にした上でプッシュスイッチ(SW2)を押してください。

(5) 動作確認を終了する場合は、モータの回転が停止していることを確認し、安定化電源の出力を OFF にします。

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

4.5.1 概要

サンプルソフトウェアでは、モータ制御開発支援ツール「Renesas Motor Workbench」をユーザインタフェース(回転/停止指令、回転速度指令等)として使用します。使用方法などの詳細は「Renesas Motor Workbench ユーザーズマニュアル」を参照してください。

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

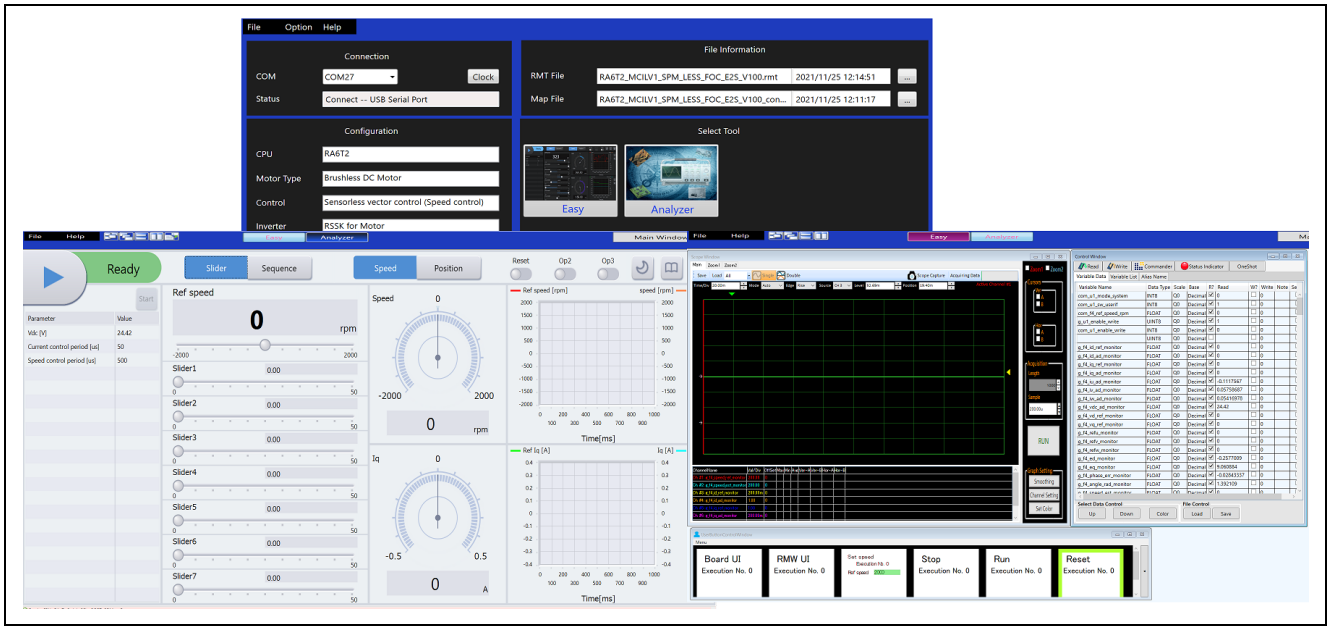
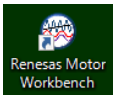


図 4-4 Renesas Motor Workbench 外観

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

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の“src/application/user_interface/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 Easy または Analyzer ボタンをクリックする。

⑤ ”4.5.2 Easy 機能操作例”または”4.5.4 Analyzer 機能操作例”を元にモータを駆動させる。

4.5.2 Easy 機能操作例

Easy 機能を使用し、モータを操作する例を以下に示します。

- ユーザインタフェースを Renesas Motor Workbench 使用に変更する
  ① “RMW UI”を ON にする。

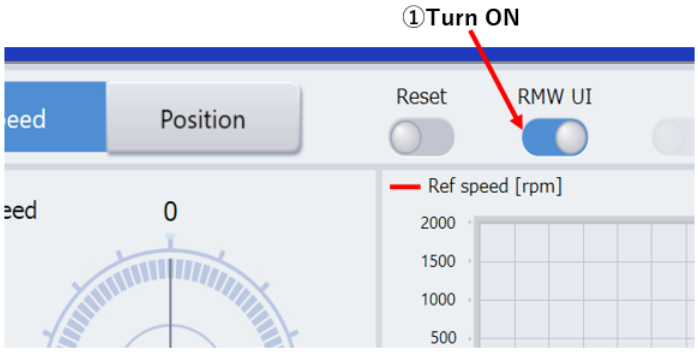


図 4-5 Renesas Motor Workbench 使用に変更する手順

- モータを回転させる
  ① “Run”ボタンを押す。
  ② 指令回転速度を“Ref speed”スライダで入力する。

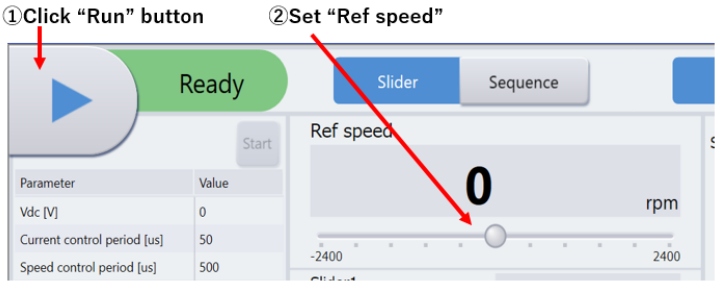


図 4-6 モータ回転の手順

- モータを停止させる
  ① “Stop”ボタンを押す。

図 4-7 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 “Reset”スイッチを ON にする。
  ② “Reset”スイッチを OFF にする。

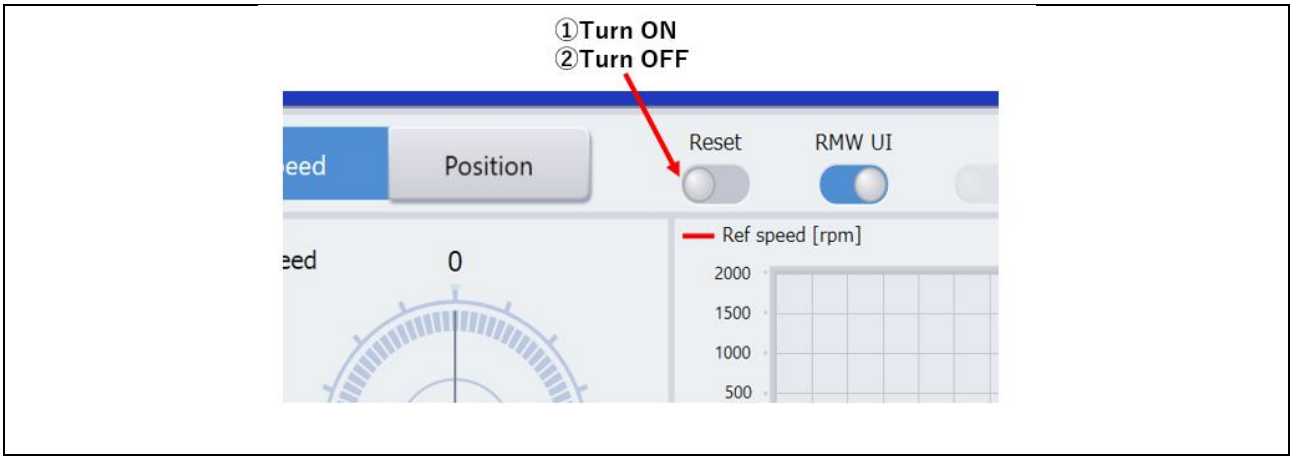


図 4-8 エラー解除の手順

4.5.3 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 4-3 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 と同じ値を書込んだ場合に対応する変数へ反映されます。ただし、(*)が付けられた変数は com_u1_enable_write に依存しません。

表 4-3 Analyzer 機能入力用変数一覧 [1/2]

変数名	型	内容
com_u1_mode_system1(*)	uint8_t	ステート管理 0：ストップモード、1：ランモード、3：リセット
com_f4_ref_speed_rpm	float	速度指令値（機械角）[rpm]
com_f4_overcurrent_limit	float	過電流エラー閾値 [A]
com_f4_overvoltage_limit	float	過電圧エラー閾値 [V]
com_f4_overspeed_limit_rpm	float	速度超過エラー閾値(機械角) [rpm]
com_f4_lowvoltage_limit	float	低電圧エラー閾値 [V]
com_u4_timeout_cnt	uint32_t	停止判定時間カウンタ
com_f4_max_drive_v	float	最大指令電圧 [V]
com_f4_min_drive_v	float	最小指令電圧 [V]
com_s4_angle_shift_adjust	int32_t	パターン切り替えタイミング調整値
com_f4_speed_lpf_k	float	速度 LPF パラメータ
com_f4_limit_speed_change	float	速度指令最大増減幅
com_f4_boot_ref_v	float	電圧指令値
com_u4_v_up_time	uint32_t	電圧指令値加算カウント回数
com_u4_v_const_time	uint32_t	電圧指令値一定カウント回数
com_s4_ol_start_rpm	int32_t	始動速度
com_s4_ol_mode1_change_rpm	int32_t	モード 1 切り替え速度
com_s4_ol_mode2_change_rpm	int32_t	モード 2 切り替え速度
com_f4_ol_start_refv	float	始動電圧
com_f4_ol_mode1_rate_rpm	float	モード 1 での速度増加率
com_f4_ol_mode2_rate_refv	float	モード 2 での電圧増加率
com_f4_ol_mode2_rate_rpm	float	モード 2 での速度増加率
com_f4_ol_mode3_rate_refv	float	モード 3 での電圧増加率
com_f4_ol_mode3_max_refv	float	モード 3 での最大電圧
com_f4_pi_ctrl_kp	float	比例ゲイン
com_f4_pi_ctrl_ki	float	積分ゲイン
com_f4_pi_ctrl_ilimit	float	電圧 PI 制御積分項リミット値 [V]
com_u4_mtr_pp	uint32_t	極対数
com_u1_enable_write	uint8_t	変数書き換え許可 (g_u1_enable_write と同じ値を書き込んだ場合に書き込み許可)

4.5.4 Analyzer 機能操作例

Analyzer 機能を使用し、モータを操作する例を以下に示します。操作は、図 4-4 で示す”Control Window”で行います。”Control Window”の詳細は、「Renesas Motor Workbench V.1.00 ユーザーズマニュアル」を参照してください。

- 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する。
  ② [Write]欄に 0 を入力する。
  ③ “Write”ボタンを押す。

- 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する。
  ② 指令回転速度を“com_f4_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_f4_ref_speed_rpm”,”g_u1_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_u1_enable_write” に④で確認した、”g_u1_enable_write”と同じ値を入力する。
  ⑥ “com_u1_mode_system”の [Write]欄に”1”を入力する。
  ⑦ “Write”ボタンを押す。

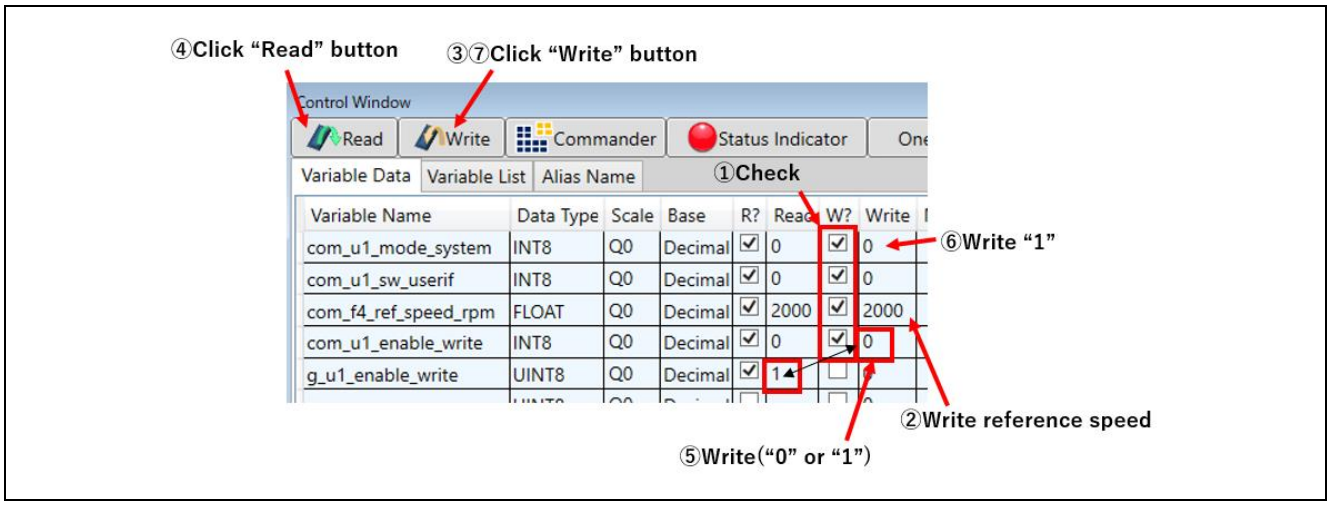


図 4-9 モータ回転の手順

- モータを停止させる
  ② “com_u1_mode_system”の[Write]欄に”0”を入力する。
  ③ ”Write”ボタンを押す。

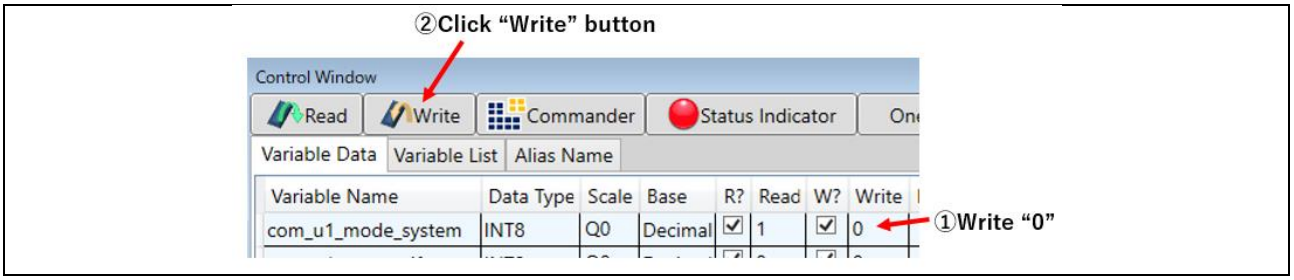


図 4-10 モータ停止の手順

- 止まってしまった（エラー）場合の処理
  ③ “com_u1_mode_system”の[Write]欄に”3”を入力する。
  ④ “Write”ボタンを押す。

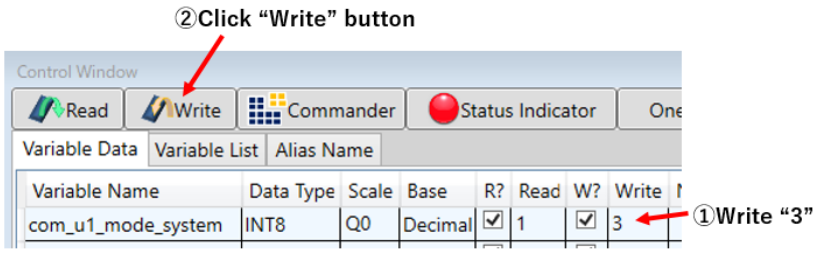


図 4-11 エラー解除の手順

5. 参考ドキュメント

RA6T1 グループ　ユーザーズマニュアル　ハードウェア編（R01UH0897）

RA Flexible Software Package Documentation

永久磁石同期モータの 120 度通電制御 (アルゴリズム編) (R01AN2657)

Renesas Motor Workbench ユーザーズマニュアル（R21UZ0004）

Evaluation System for BLDC Motor ユーザーズマニュアル（R12UZ0062）

Motor Control Evaluation System for RA Family (R12UZ0078)

RA6T1 CPU カード ユーザーズマニュアル（R12UZ0077）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2021.12.17	—	初版発行

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に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。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、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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、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て、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いたします。
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。

© 2021 Renesas Electronics Corporation. All rights reserved.