

サンプル・モータ制御ソフトウェア

仕様書

ドキュメント番号:Sample

V1.000
2015/3/16

承認	作成
—	—

【目次】

1. 改定記録	- 4 -
2. 概説	- 5 -
2.1 応用システム	- 5 -
2.2 開発環境	- 5 -
3. システム概要	- 6 -
3.1 ハードウェア構成	- 6 -
3.2 ハードウェア仕様	- 7 -
3.2.1 使用端子機能	- 7 -
3.2.2 周辺機能	- 7 -
3.2.3 割り込み機能	- 7 -
3.3 ソフトウェア基本仕様	- 8 -
3.3.1 ベクトル制御ソフトウェア基本仕様	- 8 -
3.3.2 ソフトウェア・ブロック構成図 (全体処理イメージ)	- 10 -
4. ベクトル制御機能	- 11 -
4.1 ベクトル制御	- 11 -
4.2 ベクトル制御ブロック図	- 13 -
4.3 制御内容詳細	- 15 -
4.3.1 角速度およびdq軸角度演算	- 15 -
4.3.2 速度制御演算部	- 16 -
4.3.3 dq軸電流制御及び非干渉制御	- 17 -
4.3.4 電圧ベクトル制限	- 19 -
4.3.4 dq変換/ $\alpha\beta$ 変換	- 20 -
4.3.5 dq逆変換/ $\alpha\beta$ 逆変換	- 20 -
4.3.6 空間ベクトル変調	- 21 -
5. ソフトウェア解説	- 23 -
5.1 システム・リソース	- 23 -
5.2 制御関数一覧	- 23 -
5.3 変数定義	- 26 -
5.5 制御フロー (フロー・チャート)	- 29 -
5.5.1 メインループ系	- 29 -
5.5.2 インターバルタイマ割り込み処理	- 30 -
5.5.3 制御周期割り込み処理	- 31 -

付録	- 32 -
A-1システムの開発環境	- 32 -
B-1 システムハードウェア構成	- 33 -
B-1.1 使用端子機能	- 33 -
B-1.2 周辺機能	- 34 -
B-1.3 割り込み	- 37 -
C-1 製品・モータドライバ制御関数	- 39 -

SAMPLE

2. 概説

本資料は、ルネサスエレクトロニクス製32ビットマイコンV850E2/PJ4を使用した、永久磁石同期モータ（SPM、IPM）のベクトル制御および位相制御に関して説明したものです。

2.1 応用システム

本システム（ソフトウェア）は、マイクロコントローラの内蔵周辺機能を使用してPMSMのベクトル制御および位相制御を実現しています。制御プログラム部と周辺制御（ドライバ部）構成となっており、様々なシステムに応用が可能です。

なお、本システムでは、ユーザI/Fを設けておりませんので、御社環境に合わせて、ソフトウェアの修正を行ってください。

※デフォルト状態では、RAMモニタ（Awatcher）からの操作を行うように設定されています。

※DUTY制御は、システム調整機能の一部として本仕様書への記載はありません。

2.2 開発環境

本システムの開発環境につきましては、付録を参照してください。

3. システム概要

3.1 ハードウェア構成

本システムのハードウェア構成イメージを次に示します。

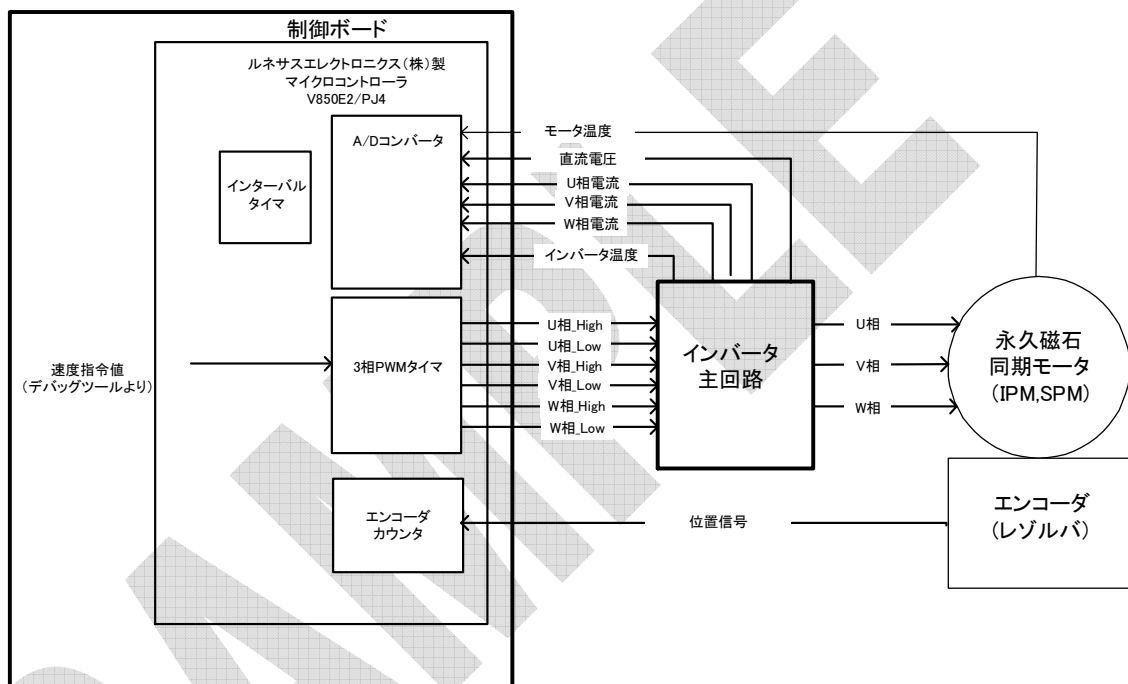


図 3-1 ハードウェア構成図

3.2 ハードウェア仕様

3.2.1 使用端子機能

本システムの使用端子機能については、付録を参照してください。

3.2.2 周辺機能

本システムの使用周辺機能については、付録を参照してください。

3.2.3 割り込み機能

本システムの使用する割り込み機能については、付録を参照してください。

SAMPLE

3.3 ソフトウェア基本仕様

3.3.1 ベクトル制御ソフトウェア基本仕様

ベクトル制御ソフトウェアの基本仕様を表3-1に示します。

表3-1 ベクトル制御ソフトウェア基本仕様

内容	詳細
モータ制御方式	<ul style="list-style-type: none"> ・正弦波駆動（ベクトル制御） ・空間ベクトル変調
磁極位置検出	<ul style="list-style-type: none"> ・レゾルバ ・12ビット分解能
キャリア周波数（PWM）周期	<ul style="list-style-type: none"> ・最大20 [kHz] ※ソフトウェアのデフォルトは10kHz
制御周期	<ul style="list-style-type: none"> ・20kHz（ベクトル制御時、キャリア周波数の2倍） ※ソフトウェアの変更により、キャリア周波数と同じにすることも可能です。
周波数範囲	<ul style="list-style-type: none"> ・1000Hz（電気角） ・CW/CCWに対応
速度制御	<ul style="list-style-type: none"> ・1msタイマ 速度制御よりq軸電流指令値を算出します。ユ
電流制御（PI制御）	<ul style="list-style-type: none"> ・制御周期 電流PI制御よりVd、Vq指令値電圧を算出します。
保護停止処理	<ul style="list-style-type: none"> ・U、V、W相電流を制御周期毎に測定し、異常発生時はPWM出力をソフトウェアによって強制停止する。 ・直流電圧を1ms毎に測定し、異常発生時はPWM出力をソフトウェアによって強制停止する。 ・マイコン外部回路の過電流検知コンパレータの出力を測定し、異常発生時はPWM出力をソフトウェアによって強制停止する。
システム内容	<ul style="list-style-type: none"> ・速度指令値により速度制御を行います。 ・電流制御、速度制御によりPWMのデューティを算出します。

本システムのソフトウェアではPWMキャリアの山谷のタイミングで、AD変換トリガが開始され、各相（U、V、W）の電流を取得します。この相電流を用いて、制御周期処理（電流フィードバック）を行います。

処理タイミングイメージ図を図3-2に示します。

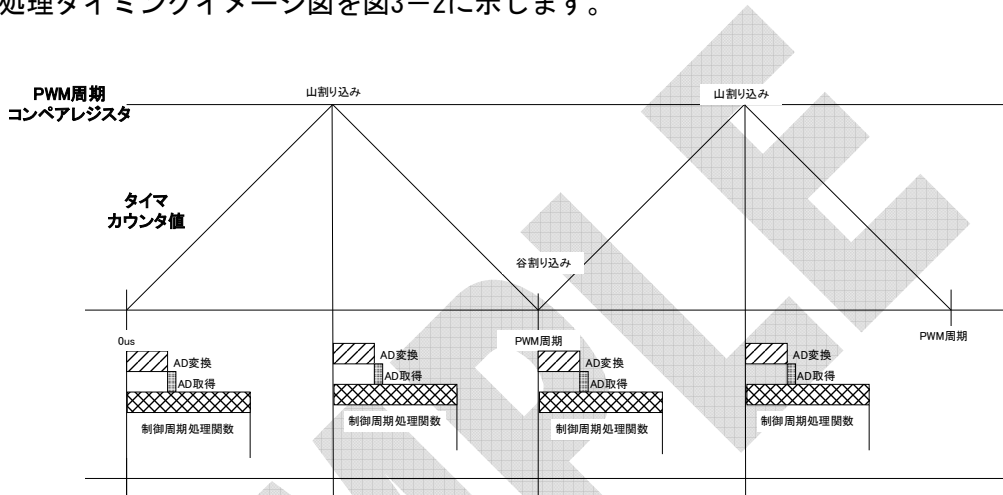


図3-2 PWMキャリアとAD変換のタイミング図

3.3.2 ソフトウェア・ブロック構成図（全体処理イメージ）

本システムのソフトウェア・ブロック構成図を図3-3に示します。また、モータドライバ制御関数（破線）については、付録を合わせて参照してください。

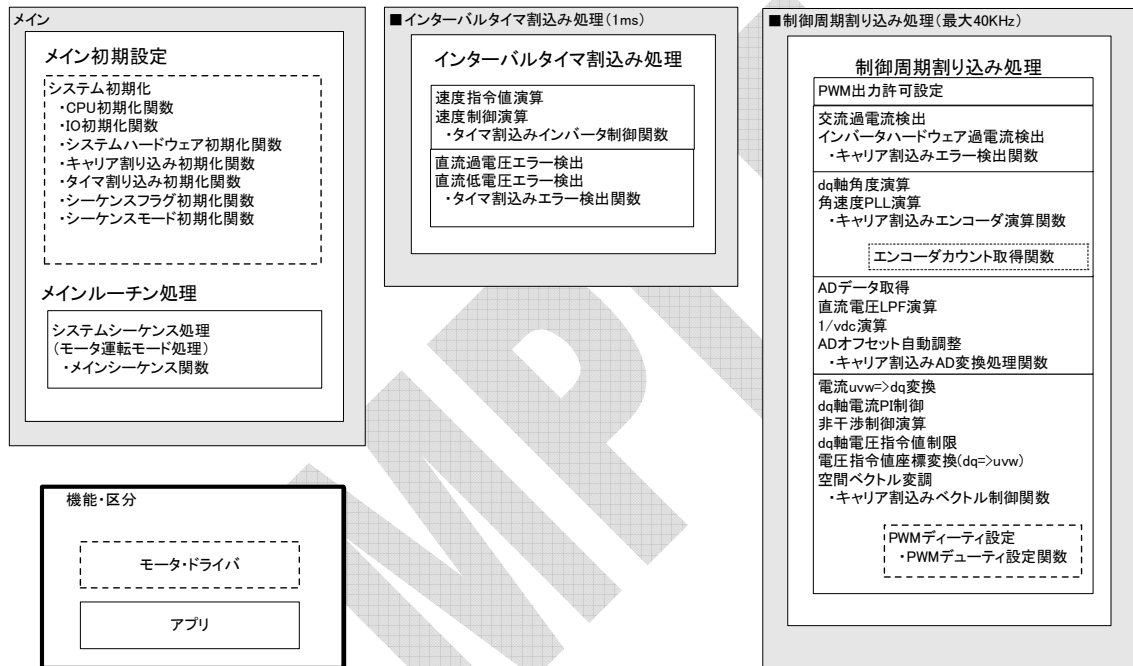


図3-3 ソフトウェア・ブロック構成図（全体仕様）

4. ベクトル制御機能

PMSMのベクトル制御プログラムについて説明します。

- ・ 制御内容
- ・ システム・リソース
- ・ 制御関数一覧
- ・ 定数一覧
- ・ 制御フロー（フロー・チャート）

4.1 ベクトル制御

本制御プログラムは、PMSMをベクトル制御しています。制御プログラムは、付録で示したハードウェア構成に対応するようにインバータに合わせて調整されています。そのため、環境を変更する場合は、その環境に合わせた調整を行う必要があります。

制御プログラムは、ユーザが設定する速度指令値を入力として動作します。速度制御の出力である電流指令値により、電流制御を行います。また、PWM変調方式では、空間ベクトル変調を使用します。

機能構成は、以下の通りです。

(1) dq軸角度演算

エンコーダカウント値から、dq軸角度を演算します。

(2) 速度演算

エンコーダカウント値を取得し、速度演算を行います。角速度演算は、PLL演算で行い、速度制御および非干渉制御に使用します。

(3) 速度制御演算

速度指令値と速度演算値を用いて、PI制御を行います。速度PI制御の出力値は、q軸電流指令値となります。

(4) dq軸電流制御

dq軸電流指令値が、検出したdq軸電流値と一致するようにPI制御を行います。

(5) 電圧ベクトル制限演算

ddq軸電圧制限は、dq軸電圧指令値の振幅が電圧制限円を超える場合にd軸電圧指令値を優先してq軸電圧指令値を制限します。

(6) 非干渉制御演算

角速度検出値、電流指令値、モータ定数から非干渉制御を行います。非干渉制御は、モータ電圧をフィードフォワード成分としてインバータ電圧指令値に加算します。

(7) dq (逆) 変換 / $\alpha\beta$ (逆) 変換

三相電流検出値をdq座標に変換 (dq軸電流検出値) します。また、dq軸電圧指令値を三相電圧指令値に変換します。

(8) 空間ベクトル(過変調)PWM制御

三相電圧指令値から、空間ベクトル変調によりPWM変調率指令値を演算します。

4.2 ベクトル制御ブロック図

ベクトル制御のブロック図を、図4-1に示します。

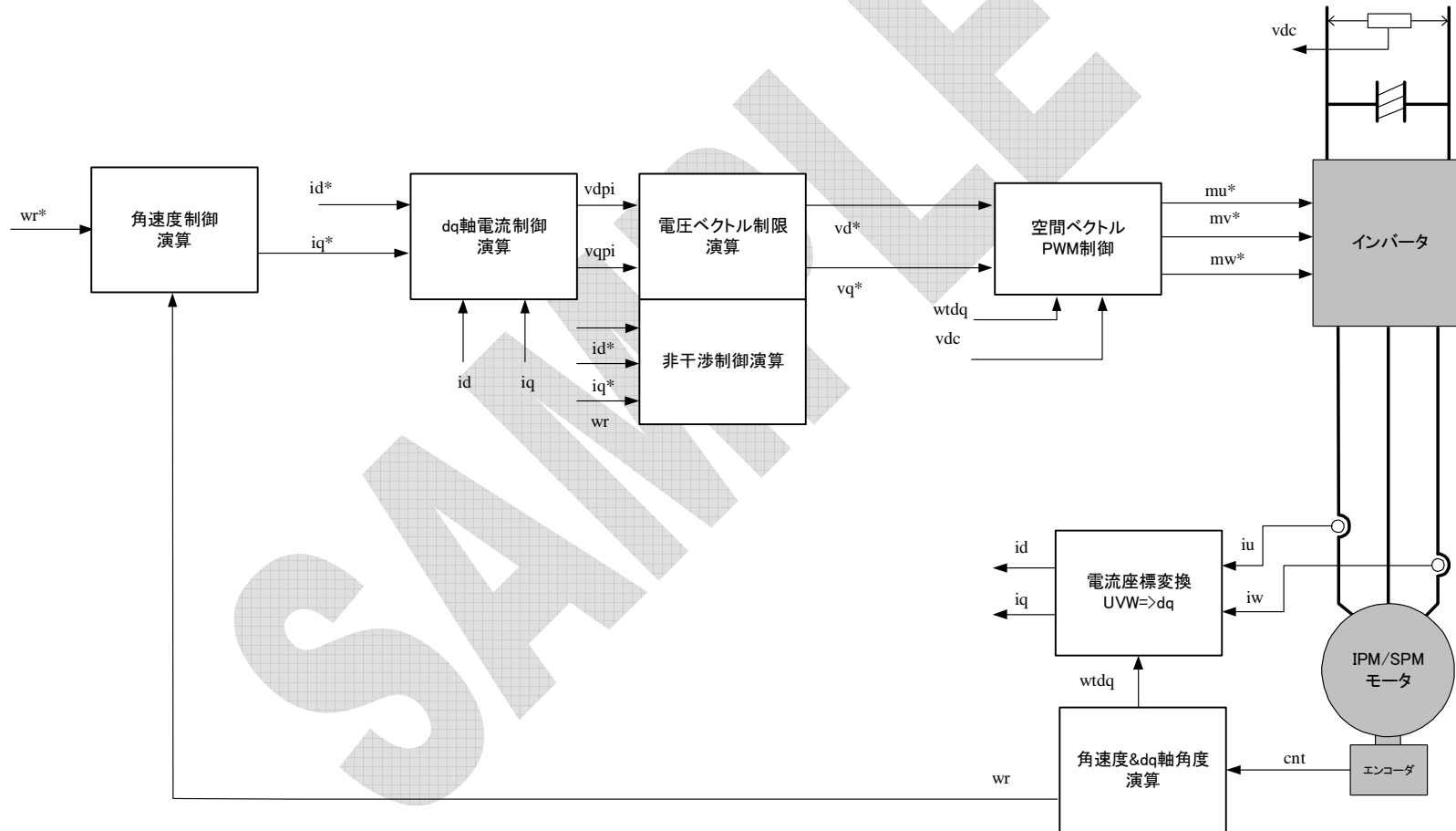


図 4-1 ベクトル制御ブロック図

表 4-1 ベクトル制御ブロック図パラメータ

名称	意味	備考
wr*	角速度指令値(rad/s)	ユーザが設定する角速度制御の入力です。
id*	d 軸電流指令値(A)	電流指令値、進角指令値から算出される d 軸電流指令値と弱め磁束制御から設定されます。
iq*	q 軸電流指令値(A)	電流指令値、進角指令値から算出される q 軸電流指令値です。
wr	角速度検出値(rad/s)	エンコーダカウンタから PLL 制御により算出した角速度です。
wtdq	dq 軸角度(rad)	エンコーダカウンタから算出した dq 軸角度です。
vdc	直流電圧(V)	
id	d 軸電流検出値(A)	相電流検出値を dq 座標変換した値です。
iq	q 軸電流検出値(A)	
vdpi	d 軸電流 PI 制御出力(V)	dq 軸電流制御の出力値です。非干渉制御に加算されます。
vqpi	q 軸電流 PI 制御出力(V)	
vd*	d 軸電圧指令値(V)	空間ベクトル変調に入力する dq 軸電圧指令値です。
vq*	q 軸電圧指令値(V)	
iu	U 相電流検出値(A)	
iw	W 相電流検出値(A)	
mu*	U 相変調率	空間ベクトル変調を行った三相変調率です。
mv*	V 相変調率	
mw*	W 相変調率	
cnt	エンコーダカウント値	エンコーダカウント値です。

4.3 制御内容詳細

4.3.1 角速度および dq 軸角度演算

図4-2に角速度およびdq軸角度演算ブロック図を示します。dq軸角度はエンコーダカウンタにゲインを乗じてラジアン(rad)に変換します。角速度はPLLを用いて演算します。

表 4-2 角速度および dq 軸角度演算ブロック図パラメータ

名称	意味	備考
cnt	エンコーダカウンタ値	
wtdq	dq 軸角度(rad)	
wr	角速度検出値(rad/s)	エンコーダカウンタから演算した角速度に LPF 処理を行ったデータです。
wrpll	角速度 PLL 演算値(rad/s)	モータ電流制御部では使用しません。

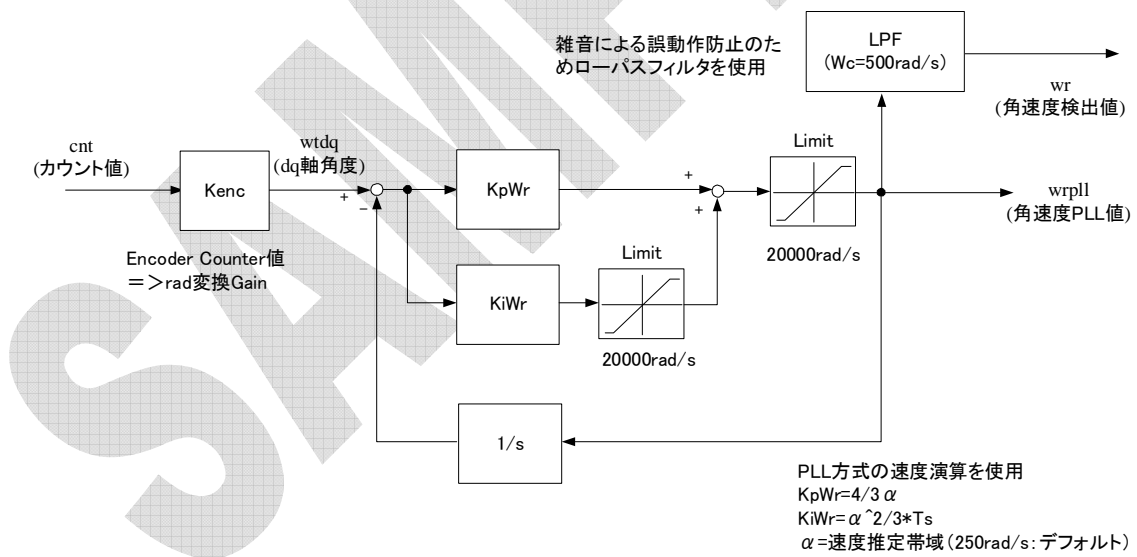


図4-2 角速度およびdq軸角度演算ブロック図

4.3.2 速度制御演算部

- ・ 図 4-3 に速度制御演算ブロック図を示します。速度制御演算部は、速度指令値演算および速度制御から構成されます。
- ・ 角速度制御では、角速度指令値と角速度演算値を用いて、PI 制御を行い q 軸電流指令値が算出されます。

表 4-3 速度制御演算ブロック図パラメータ

名称	意味	備考
w_r^*	角速度指令値(rad/s)	ユーザが設定する角速度制御の入力です。
w_r	角速度検出値(rad/s)	エンコーダカウンタから PLL 制御により算出した角速度です。
i_{q^*}	q 軸電流指令値(A)	電流指令値、進角指令値から算出される q 軸電流指令値です。

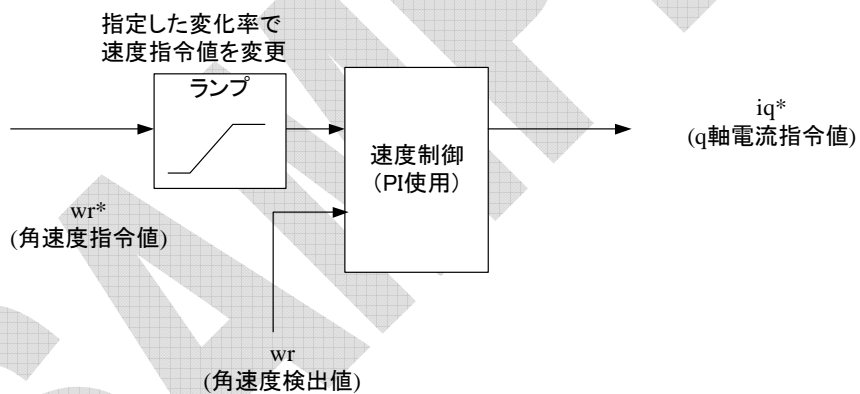


図 4-3 速度制御演算ブロック図

4.3.3 dq 軸電流制御及び非干渉制御

- ・ 図 4-4 に dq 軸電流制御及び非干渉制御ブロック図を示します。非干渉制御は、PMSM の電圧方程式に従って演算します。dq 座標系における PMSM の電圧方程式は以下の式になります。

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = R \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \frac{d}{dt} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \omega \begin{bmatrix} 0 & -L_q \\ L_d & 0 \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \omega \begin{bmatrix} 0 \\ \phi \end{bmatrix}$$

備考	L_d	: d軸インダクタンス (H)
	L_q	: q軸インダクタンス (H)
	I_d	: d軸電流 (A)
	I_q	: q軸電流 (A)
	ω	: 角速度 (rad/s)
	ϕ	: 永久磁石磁束 (Wb)
	V_d	: d軸電圧 (V)
	V_q	: q軸電圧 (V)

- ・ 上式で定常状態 ($d/dt=0$) として、非干渉制御電圧を求めます。このとき演算に使用する dq 軸電流は、指令値を使用します。角速度には、検出値を使用します。
- ・ dq 軸電流制御の PI ゲインは次式から演算します。
 $K_p = (L_{dq} \times \text{制御周波数} \times 0.1)$
 $K_i = (K_P) \times 0.1$

表 4-4 dq 軸電流制御および非干渉制御ブロック図パラメータ

名称	意味	備考
id*	d 軸電流指令値(A)	0A に指定します。
iq*	q 軸電流指令値(A)	速度制御から設定されます。
id	d 軸電流検出値(A)	
iq	q 軸電流検出値(A)	
wr	角速度検出値(rad/s)	エンコーダカウンタから演算した角速度に LPF 処理を行ったデータです。
vdpi	d 軸電流 PI 制御出力(V)	
vqpi	q 軸電流 PI 制御出力(V)	
vdc*	d 軸電圧指令値(V)	電圧ベクトル制限を行う前のdq軸電圧指令値です。
vqc*	q 軸電圧指令値(V)	
R	固定子抵抗(Ω)	
Ld	d 軸インダクタンス(H)	dq 軸インダクタンス飽和特性から演算したインダクタンス値です。
Lq	q 軸インダクタンス(H)	
ϕ_m	永久磁石鎖交磁束(Wb)	

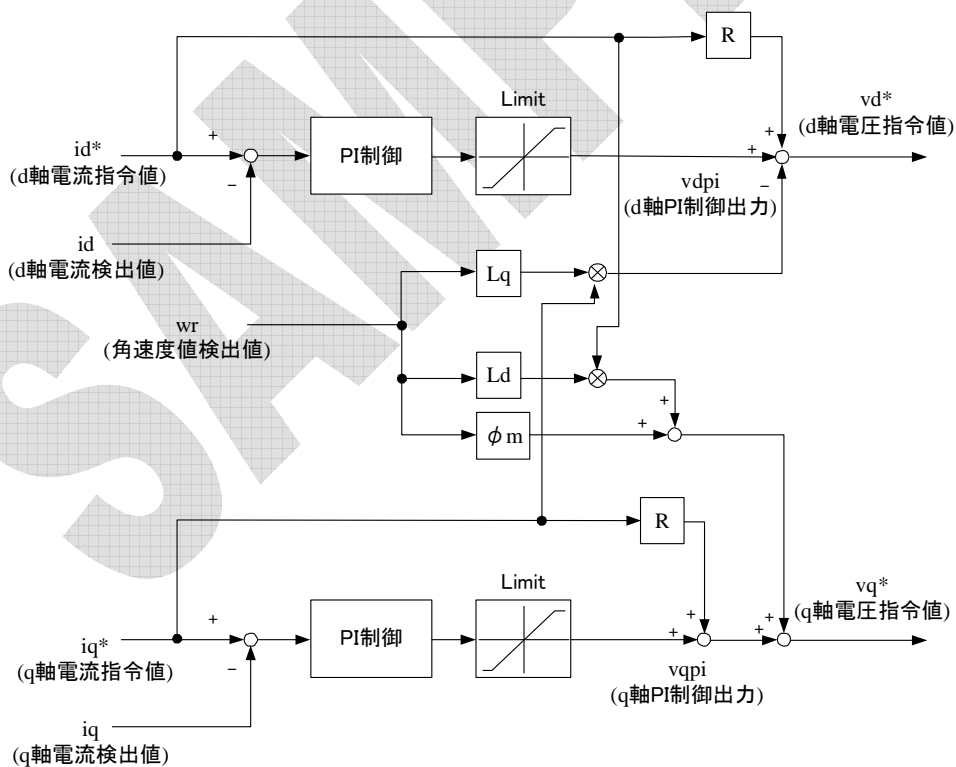


図 4-4 dq 軸電流制御及び非干渉制御ブロック図

4.3.4 電圧ベクトル制限

図4-5に電圧ベクトル制限のブロック図を示します。電圧ベクトル制限では、出力電圧ベクトルが最大出力電圧設定値を超えないように、dq軸電圧ベクトルを制限します。最大出力電圧設定値は、直流電圧のK倍を選択します。ここでは、最大出力電圧設定値を超えた場合、d軸電圧指令値を優先して、q軸電圧を制限するように処理します。

表 4-5 電圧ベクトル制限ブロック図パラメータ

名称	意味	備考
vdc*	d 軸電圧指令値(V) (テンポラリ)	電圧ベクトル制限を行う前のdq軸電圧指令値です。
vqc*	q 軸電圧指令値(V) (テンポラリ)	
vd*	d 軸電圧指令値(V)	
vq*	q 軸電圧指令値(V)	
vdc	直流電圧(V)	
Vref_max	INV 電圧制限(V)	直流電圧の K 倍
Vref*	INV 電圧指令電圧(V)	$\sqrt{(vdc^*^2+vqc^*^2)}$ 演算値です。

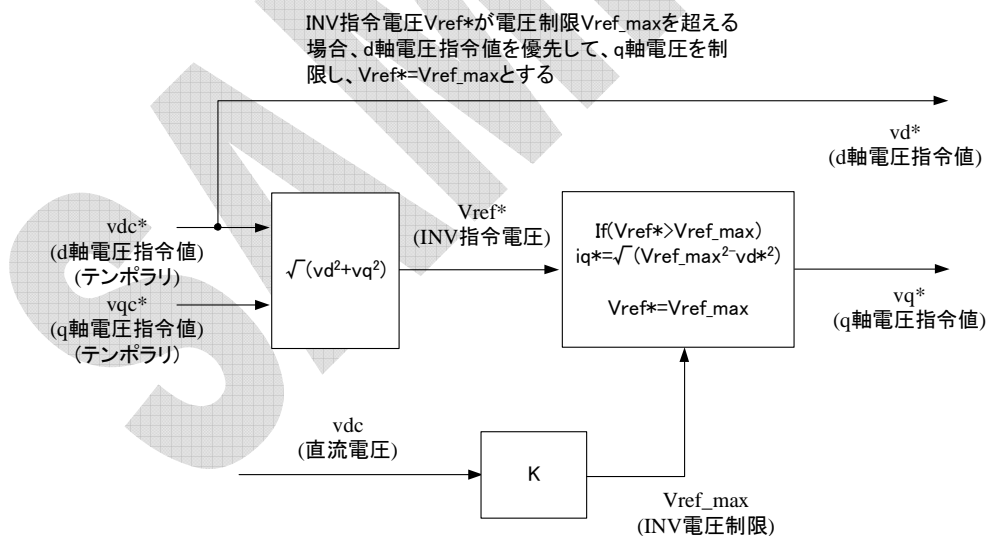


図 4-5 電圧ベクトル制限ブロック図

4.3.4 dq 変換 / $\alpha\beta$ 変換

相電流から変換行列を使い、dq軸電流を求めます。本ソフトウェアでは、 $\alpha\beta$ 変換とdq変換を順番に行っています。

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} I_u \\ I_v \\ I_w \end{bmatrix}$$

↑
dq 軸電流

← 相電流

I_α, I_β

4.3.5 dq 逆変換 / $\alpha\beta$ 逆変換

dq軸電圧指令値から三相指令値電圧を求めます。本ソフトウェアでは、dq逆変換と $\alpha\beta$ 逆変換を順番に行っています。

$$\begin{bmatrix} V_u \\ V_v \\ V_w \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} V_d \\ V_q \end{bmatrix}$$

↑
出力電圧指令

← dq 軸指令値電

V_α, V_β

4.3.6 空間ベクトル変調

図4-6に空間ベクトル変調ブロック図を示します。dq軸電圧指令値を三相電圧指令値に変換後、三相変調率を演算します。三相電圧指令値に変換する際は、サンプリング遅れ補償を行います。演算式を次に示します。

$\begin{aligned} \text{変調率}\mu &= V_u \times 2 / \text{インバータ電圧} \\ \text{変調率}\mu_v &= V_v \times 2 / \text{インバータ電圧} \\ \text{変調率}\mu_w &= V_w \times 2 / \text{インバータ電圧} \end{aligned}$
--

空間ベクトル変調から算出される変調率を μ' 、 μ_v' 、 μ_w' とし、 μ 、 μ_v 、 μ_w の最大値を $\max(\mu, \mu_v, \mu_w)$ 、最小値を $\min(\mu, \mu_v, \mu_w)$ であらわしています。

$\begin{aligned} \mu' &= \mu - \{\max(\mu, \mu_v, \mu_w) + \min(\mu, \mu_v, \mu_w)\} / 2 \\ \mu_v' &= \mu_v - \{\max(\mu, \mu_v, \mu_w) + \min(\mu_v, \mu_v, \mu_w)\} / 2 \\ \mu_w' &= \mu_w - \{\max(\mu, \mu_v, \mu_w) + \min(\mu_w, \mu_w, \mu_w)\} / 2 \end{aligned}$

表 4-6 空間ベクトル（過変調）PWM 制御ブロック図パラメータ

名 称	意 味	備 考
vd*	d 軸電圧指令値(V)	
vq*	q 軸電圧指令値(V)	
vdc	直流電圧(V)	
wr	角速度(rad/s)	
wtdq	dq 軸角度(rad)	
vu*	U 相電圧指令値(V)	
vv*	V 相電圧指令値(V)	
vw*	W 相電圧指令値(V)	
mu*	U 相変調率	
mv*	V 相変調率	
mw*	W 相変調率	

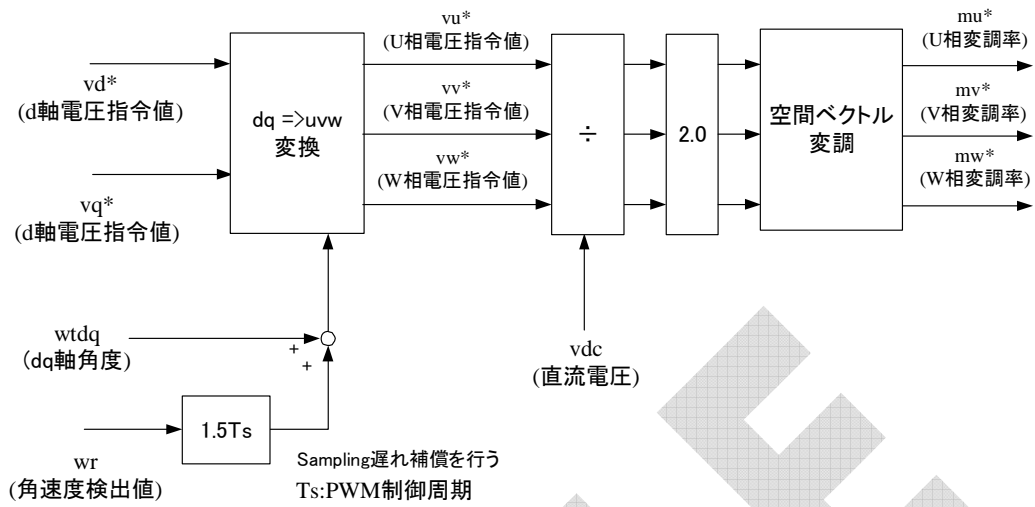


図 4-6 空間ベクトル PWM 制御ブロック図

5. ソフトウェア解説

5.1 システム・リソース

本ソフトウェアで使用するシステム・リソースは、付録を参照してください。

5.2 制御関数一覧

本ソフトウェアでは、複数の制御関数を使用しています。制御関数の一覧を次に示します。より詳細な処理については、フロー・チャート、またはプログラムを参照してください。

表5-1 制御関数の主な機能

機能	組み込まれる関数/名称	主な機能	ファイル名
メイン 初期設定	void main(void)/メイン関数	<ul style="list-style-type: none"> ハードウェアの初期化 システムシーケンス処理 	fpmsm_main.c
	void fb_main_init(void)/初期化関数	<ul style="list-style-type: none"> システム初期化処理 	fpmsm_mian.c
	void cpu_init(void)/CPU初期化関数	<ul style="list-style-type: none"> 割り込みマスク 	io_CPU_r.c
	void pio_init(void)/IO初期化関数	<ul style="list-style-type: none"> 汎用ポート/外部割り込み設定 	io_PIO.c
	void fb_hw_ctrl_init(void)/ システムハードウェア初期化関数	<ul style="list-style-type: none"> PWMタイマ初期化 AD初期化 1mSタイマ初期化 エンコーダ機能初期化 	fb_hw_ctrl.c
	void fb_pwm_int_init(void)/ キャリア割込み処理初期化関数	<ul style="list-style-type: none"> キャリア割込みインバータ制御初期化 キャリア割込みAD変換処理初期化 キャリア割込みエンコーダ処理初期化 キャリア割込みエラー検出初期化 	fi_pwm_int.c
	void fb_pwm_invctrl_init(void)/ キャリア割込みインバータ制御初期化関数	<ul style="list-style-type: none"> キャリア割込みインバータ制御変数初期化 	fi_pwm_invctrl.c
void fb_pwm_adctrl_init(void)/ キャリア割込みAD変換処理初期化関数	<ul style="list-style-type: none"> キャリア割込みAD変換処理変数初期化 	fi_pwm_adctrl.c	

5.3 変数定義

本ソフトウェアで使用する変数一覧を次に示します。ただし、ローカル変数は記載していません。

変数の型名の定義 (vals_typedef.hにて定義)

- ・ u1 : 符号無し8ビットデータ
- ・ s1 : 符号付き8ビットデータ
- ・ u2 : 符号無し16ビットデータ
- ・ s2 : 符号付き16ビットデータ
- ・ u4 : 符号無し32ビットデータ
- ・ s4 : 符号付き32ビットデータ
- ・ f4 : 浮動小数32ビットデータ

本システムの変数一覧についてはソースファイルを直接参照してください。

※RAMモニタ (Awatcher) を使用するためには、変数はグローバル定義する必要があります。

5.4 定数一覧

本制御プログラムで使用する定数一覧を次に示します。ヘッダファイルfpmsm_user.hに記載されています。

表5-2 定数一覧

項目	型	定数名	備考	単位
PWM論理設定 (0: 正論理 1: 負論理)	u4	PWM_LOGIC	PWM論理/割込みソース設定	—
PWM割込みソース設定 (1:000(山)2:111(谷)3: 両方)	u4	PWM_SOURCE		—
PWMキャリア周波数設定 サンプリング周波数は、インバータPWM周波数の2倍	u4	FC_PWM	PWMキャリア周波数設定/インバータ	Hz
デッドタイム(ns)	u4	DEAD_TIME_NS	デッドタイム&電圧制御誤差設定	nsec
モータ電流レンジ	f4	R_IINV_AD	AD変換レンジ設定	A
直流電圧レンジ	f4	R_VDC_AD		V
dq軸電圧ベクトル最大値比率	f4	K_VDQ_MAX	インバータ電圧&変調率制限設定	—
インバータ変調率制限	f4	M_MAX		—
交流過電流レベル	f4	LEV_OC_IAC	システムSW保護レベル設定	A
直流過電流レベル	f4	LEV_OC_IDC		A
直流過電圧レベル	f4	LEV_OV_VDC		V
直流低電圧レベル	f4	LEV_UV_VDC		V
交流過電流エラー時限	u2	CNT_OC_IAC_MAX	システムSW保護検出時限設定	PWM 周期
直流過電流エラー時限	u2	CNT_OC_IDC_MAX		PWM 周期
直流過電圧エラー時限	u2	CNT_OV_VDC_MAX		PWM 周期
直流低電圧エラー時限	u2	CNT_UV_VDC_MAX		1ms
極対数	f4	PF	モータパラメータ	—
固定子抵抗	f4	RS		Ω
永久磁石鎖交磁束	f4	FM		Wb
d軸インダクタンス定常値	f4	LD_CONST		H
q軸インダクタンス定常値	f4	LQ_CONST		H
電流ベクトル制限値	f4	IDQ_MAX		電流リミット設定
レゾルバオフセット値	f4	WT_DQ_OFFSET	レゾルバ設定	rad
角速度PLL比例ゲイン	f4	KP_WRPLL	角速度PLLパラメータ設定	—
角速度PLL積分ゲイン	f4	KI_WRPLL		—
角速度PLL	f4	WC_LPF_WRPLL		rad/s

5.5 制御フロー（フロー・チャート）

5.5.1 メインループ系

(1) メイン処理（ソースファイル名：fpmsm_main.c）

- ・ H/Wの初期設定
- ・ メインループ

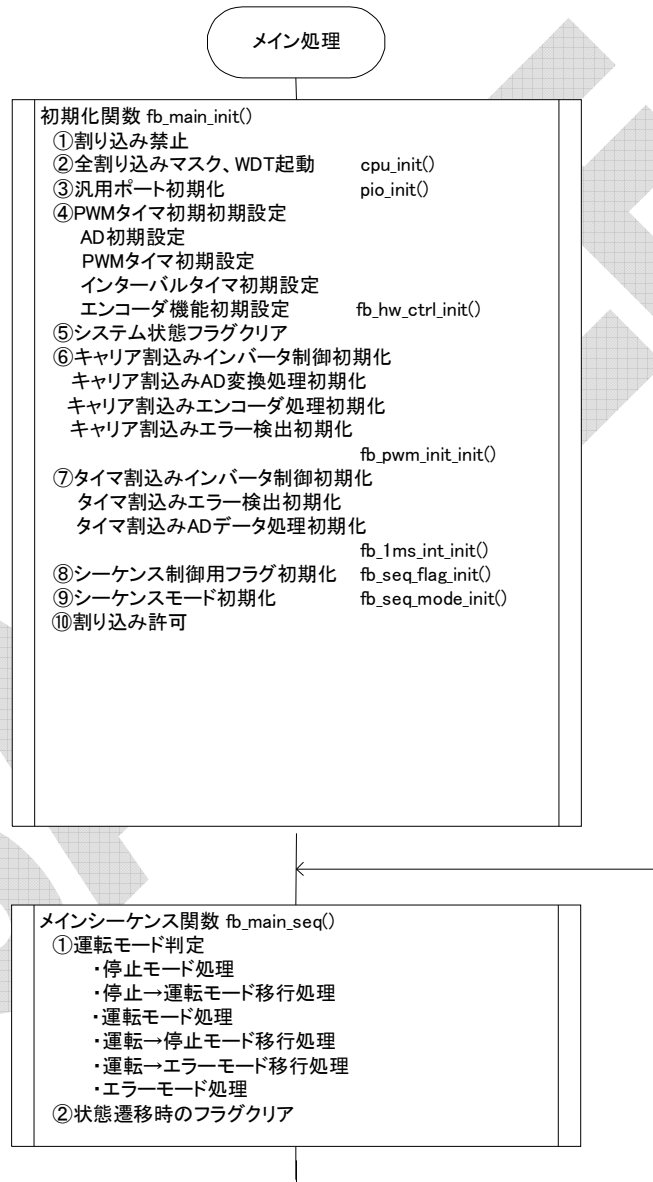


図5-1 メイン処理

付録

A-1 システムの開発環境

本システムの開発環境を次に示します。

(1) ソフトウェア開発環境

- ・開発統合環境 : CubeSuite+ V1.03
- ・Cコンパイラ : CXコンパイラ V1.21

SAMPLE

B-1 システムハードウェア構成

B-1.1 使用端子機能

本システムのV850E2/PJ4マイクロコントローラの使用端子を表B-1に示します。
ハードウェア依存（LED1，外付けウォッチドックタイマ等）等は除外しております。

表B-1 使用端子端子一覧

端子名	機能
P0_0/ URTH0RXD	AWatcher I/F
P0_1/ URTH0TXD	AWatcher I/F
P1_1/TSG2001	相補PWM出力（U相ハイサイド）
P1_2/TSG2002	相補PWM出力（U相ロウサイド）
P1_3/TSG2003	相補PWM出力（V相ハイサイド）
P1_4/TSG2004	相補PWM出力（V相ロウサイド）
P1_5/TSG2005	相補PWM出力（W相ロウサイド）
P1_6/TSG2006	相補PWM出力（W相ロウサイド）
P3_1	レゾルバIC用シリアル入力設定用チップセレクト
P3_0	レゾルバIC用シリアル出力用チップセレクト
P7_4/ENCA1E0	エンコーダ入力A
P7_5/ENCA1E1	エンコーダ入力B
P7_6/ENCA1EC	エンコーダ入力Z
P3_2/CSIG1SI	シリアルデータ（レゾルバICとの通信用）
P3_3/CSIG1SO	シリアルデータ（レゾルバICとの通信用）
P3_4/CSIG1SC	シリアルデータ（レゾルバICとの通信用）
ADCA0I1	U相電流測定
ADCA0I2	V相電流測定
ADCA0I3	W相電流測定
ADCA0I4	直流電流測定
ADCA0I5	インバータ直流電圧（VDC）
ADCA0I7	U相主回路温度
ADCA0I8	V相主回路温度
ADCA0I9	W相主回路温度
ADCA0I10	モータ温度

B-1.2 周辺機能

本システムに使用する周辺機能一覧を表B-2に示します。

表B-2 周辺機能一覧(1/2)

周辺機能	機能設定			
A/D Converter (ADCA0)	12ビット分解能 / ディスチャージ無効 / バッファアンプ無効 / エラー割り込み発生無し CG2は、PWMキャリアの山谷のタイミングに同期して変換開始			
	チャンネル・グループ設定			
		CG0	CG1	CG2
	変換モード	連続スキャンモード	—	ワンショット変換モード
	スキャンリスト	7~22	—	1~6
	変換終了割り込み	なし	—	なし
	起動トリガ	ソフトウェアトリガ	—	ハードウェアトリガ
	サンプル&ホールド	無し	—	あり
変換チャンネル内容	ch 7 : インバータ 直流電圧 ch 10-12 : IGBT温度 ch 8-9,13-22 : —	—	インバータ ch 1 : U相電流 ch 2 : V相電流 ch 3 : W相電流 ch4-6 : —	
AD変換の優先順位 CG2 > CG1 > CG0				
TSG2 (TSG20)	PWM (キャリア) 周期 : 10KHz (制御周期は、PWM周期の倍 : 20KHz) 相補PWM出力 (3相) アクティブレベル : Hi Level AD起動トリガ出力			
タイマ・アレイ・ユニットA (TAUA0)	使用CH CH0 : インターバルタイマ機能 (周期1mS)			

(2) TSG2 (TSG20)

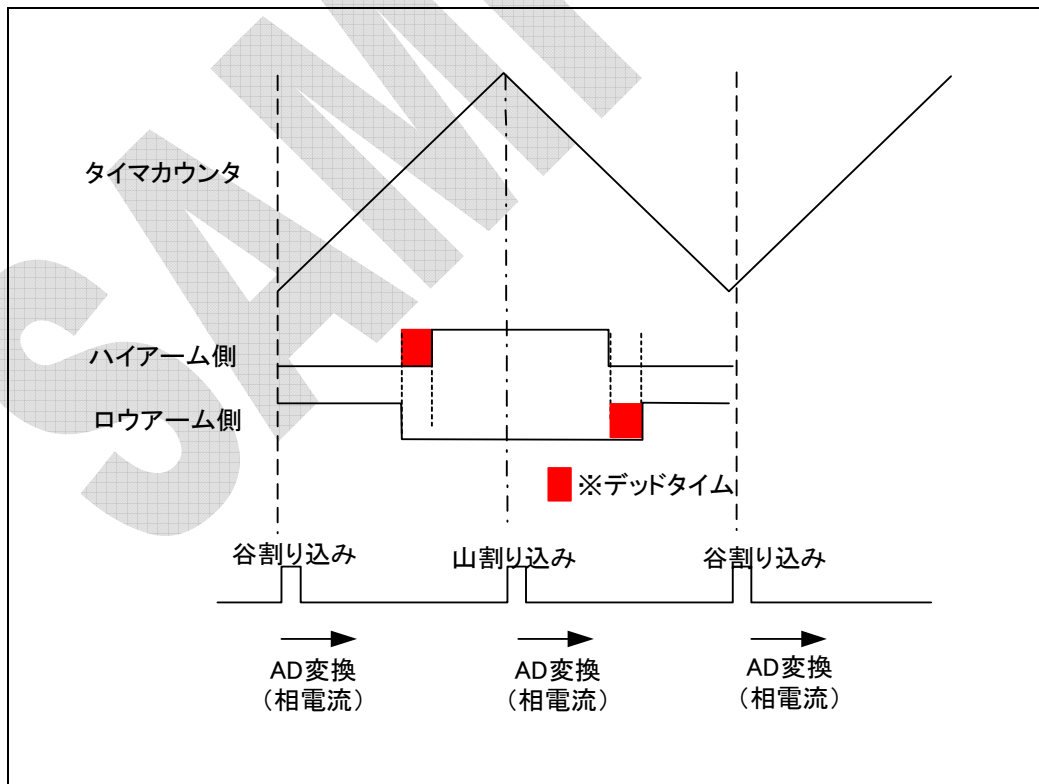
TSG20を使用し、デッドタイム付き3相PWMを出力します。

PWM出力端子の一覧を次の表B-4に示します。

表B-4 PWM出力端子一覧

端子名	機能	備考
TSG2001	インバータPWMゲート信号 (UP)	論理設定はハイアクティブ
TSG2002	インバータPWMゲート信号 (UN)	
TSG2003	インバータPWMゲート信号 (VP)	
TSG2004	インバータPWMゲート信号 (VN)	
TSG2005	インバータPWMゲート信号 (WP)	
TSG2006	インバータPWMゲート信号 (WN)	

また、下記図B-1の様に、TSG20の山谷割り込みに同期してAD変換（相電流）を開始します。



図B-1 相補PWM出力波形（デッドタイム付きPWM出力）

(3) タイマ・アレイ・ユニットA (TAUA0)

TAUA0のチャンネル0を使用して、1mSのインターバル割り込みを発生させ、インターバル割り込み処理を行います。

(4) エンコーダ・タイマ (ENCA1)

エンコーダ入力信号により、カウントのアップ/ダウンを行う機能（エンコーダ・コンペアモード）に設定します。

このエンコーダのカウント値からdq軸角度にソフトウェアで変換します。

また、エンコーダの割り込みは、1パルス制御に利用します。

(5) クロック同期シリアルI/F (CSIG1)

レゾルバICとの通信を行い、エンコーダカウンタの初期値取得の為に使用します。

(6) 調歩同期シリアルI/F (UART0)

Awatcher用のI/Fになります。

B-1.3 割り込み

本システムで使用する割り込みの一覧を次に示します。

表B-5 使用割り込み一覧

割り込み名称	機能
INTTAUA0I0	インターバルタイマ周期割り込み (1ms)
INTTSG20IVLY	PWM周期 (谷割り込み) 同期割り込み (制御周期割り込み処理)
INTTSG20IPEK	PWM周期 (山割り込み) 同期割り込み (制御周期割り込み処理)
INTENCA1I1	エンコーダコンペア一致割り込み

また、本制御プログラムで使用する割り込みの主な機能を次に示します。

表B-6 割り込みリソース

割り込み	ソースファイル名/関数名	周期	主な機能
谷割り込み (INTTSG20IVLY) 山割り込み (INTTSG20IPEK)	void fi_pwm_int(void) void fi_pwm_int(void)にて 以下の関数を読み出し void fi_pwm_invadj(void) void fi_pwm_errctrl(void), void fi_pwm_encctrl(void) void fi_pwm_vctrler(void) void fi_pwm_adctrl(void), void fi_pwm_vecctrl(void) void fi_pwm_anglectrl(void)	Min 50 μ s PWM(キャリア)周期/2 (山谷割り込み)	<ul style="list-style-type: none"> ・パラメータ測定処理 ・システム保護エラー処理 ・エンコーダデータ処理 ・電圧制御誤差補償 ・ADデータ処理 ・インバータ制御処理 等
インターバル・タイマ割り込み (INTTAUA010)	void fi_1ms_int(void) void fi_1ms_int(void)にて 以下の関数を呼び出し void fi_1ms_vecctrl(void), void fi_1ms_anglectrl(void), void fi_1ms_errctrl(void) void fi_1ms_adctrl(void) void fi_1ms_sysctrl(void)	1 ms (1 kHz) / 固定	<ul style="list-style-type: none"> ・速度制御 ・Ld,Lq飽和特性演算 ・システム保護エラー処理 ・システムI/F処理 等
エンコーダ・タイマ割り込み (INTENCA111)	void fi_eed_int(void)	-	<ul style="list-style-type: none"> ・矩形波処理

C-1 製品・モータドライバ制御関数

ドライバ制御関数の主な機能を次に示します。

表C-1 モータドライバ制御関数の主な機能 (1/1)

ファイル名	組み込まれる関数	名称	主な機能
io_TSG20.c	unsigned short tsg20_init(unsigned long 第1引数, unsigned long 第2引数, unsigned long 第3引数, unsigned char第4引数, unsigned char第5引数, unsigned short第6引数)	3相相補PWM出力 初期化関数	<ul style="list-style-type: none"> 3相相補PWM出力用タイマの初期化 第1引数: INVキャリア周波数 [Hz] 第2引数: デッドタイム設定ON側 [ns] 第3引数: デッドタイム設定OFF側 [ns] 第4引数: INVロジックレベル 0 = 正論理, 1 = 負論理 第5引数: 割り込みソース 1=山, 2=谷, 3=山谷 第6引数: 割り込みレベル
	void dpin0_setuvwf_f(float 第1引数, float 第2引数, float 第3引数, float 第4引数)	PWMデューティ 設定関数	<ul style="list-style-type: none"> 引数 (変調率) からPWMのデューティを設定 第1引数: U相変調率(±1.0) 第2引数: V相変調率(±1.0) 第3引数: W相変調率(±1.0) 第4引数: キャリア周波数
	void tsg20_start(void)	3相PWMタイマ開始 関数	・TSG20スタート
	void tsg20_stop(void)	3相PWMタイマ停止 関数	・TSG20停止
	void dpin0_start(void)	3相PWM出力開始 関数	・3相PWM出力端子出力許可
	void dpin0_stop(void)	3相PWM出力停止 関数	・3相PWM出力端子出力停止 (Hi-Z)
	void dpin0_set_addeley(signed long 第1引数)	ADトリガディレイ 値 設定関数	・ADトリガディレイ値を設定
	void dpin0_setuvw_1pulse (unsigned char 第1引数)	1パルスゲート 設定関数	<ul style="list-style-type: none"> 1パルス・ゲートパターンを設定 第1引数: ゲートパターン
Io_ENCA1.c	void enca1_init(unsigned short 第1引数)	ENCA1初期化関数	<ul style="list-style-type: none"> ENCA1初期化関数 第1引数: エンコーダカウンタMAX値
	void enca1_cnt_set(unsigned short 第1引数)	ENCA1カウンタ設 定 関数	<ul style="list-style-type: none"> ENCA1のカウンタ値の初期設定 第1引数: エンコーダカウンタ値
	void enca1_start	ENCA1開始関数	・ENCA1スタート
	void enca1_stop	ENCA1停止関数	・ENCA1停止