

V850E2/PJ4 対応

I/O ライブラリ(デバイスドライバ)関数

(サンプル : モータ制御ソフトウェア用)

説明書

AMDT0XXX-XX-XXX

V1.000

SAMPLE

承認	作成

【目次】

1. 改定記録	- 3 -
2. 概説	- 4 -
2.1 対象CPU	- 4 -
2.2 開発環境	- 4 -
3. 想定システム概要	- 5 -
3.1 想定ハードウェア構成	- 5 -
3.2 デバイスドライバ設定（使用）ハードウェア	- 6 -
3.2.1 設定（使用）端子機能	- 6 -
3.2.2 設定（使用）周辺機能内容	- 7 -
4. デバイスドライバ関数	- 8 -
4.1 関数の種類	- 8 -
4.2 デバイスドライバ関数の説明	- 10 -
4.2.1 A/Dコンバータ関連	- 10 -
4.2.2 CPU関連	- 11 -
4.2.3 ポート設定関連	- 12 -
4.2.4 エンコーダタイマ関連	- 14 -
4.2.5 インターバルタイマ関連	- 21 -
4.2.6 インพุットキャプチャタイマ関連	- 22 -
4.2.7 三相PWM 関連	- 23 -
4.2.8 CAN関連	- 37 -

2. 概説

本資料は、永久磁石同期モータ（以降、PMSMモータ）のベクトル制御に準備した、I/Oライブラリ関数に関して説明したものです。本書では、I/Oライブラリを「デバイスドライバ」と呼びます。

デバイスドライバ（CPU周辺I/O）を準備しておくことで、CPUを変更した際、制御側のソフトウェアの変更を最小限に押さえることができます。

2.1 対象 CPU

対象CPU : V850E2/PJ4

参考ユーザーズマニュアル : R01UH0098JJ0102

2.2 開発環境

統合開発環境 : CubeSuite+ V2.00

コンパイラ : CX V1.21

3. 想定システム概要

3.1 想定ハードウェア構成

本デバイスドライバ関数の想定ハードウェア構成イメージを次に示します。

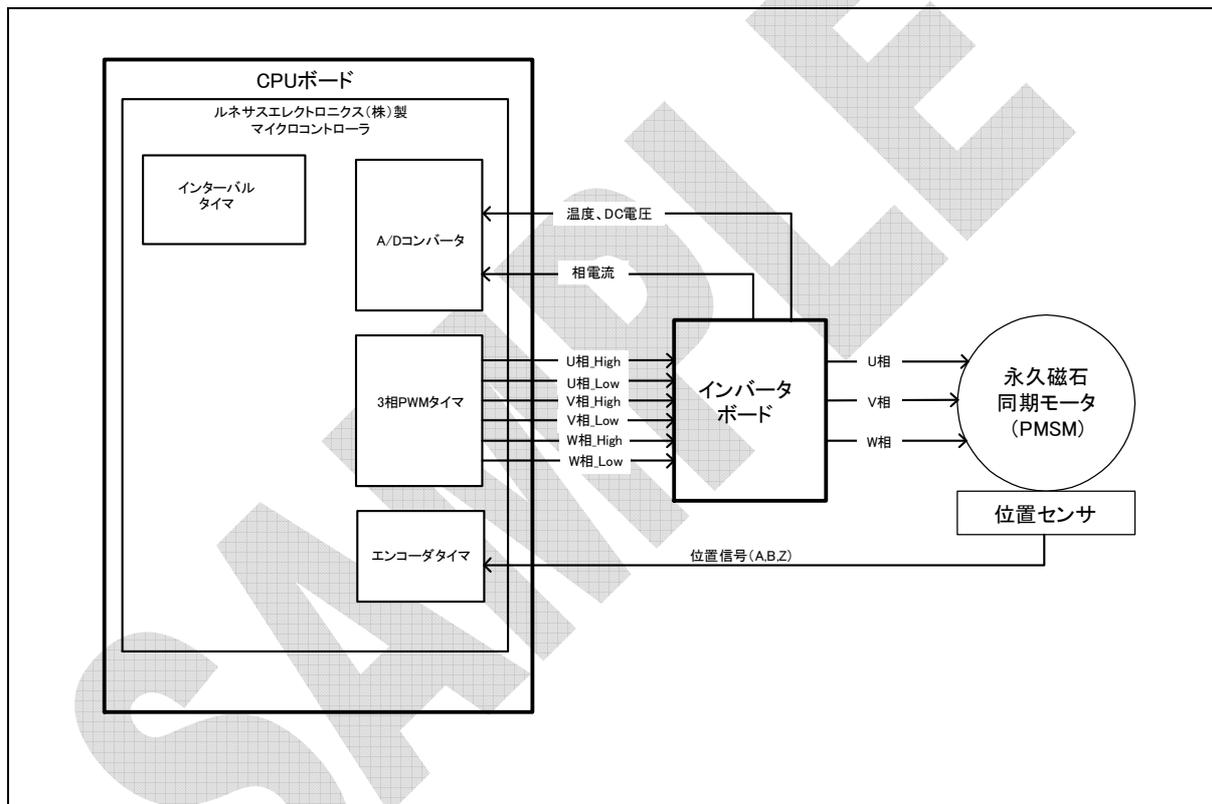


図 3.1 ハードウェア構成イメージ図

3.2 デバイスドライバ設定（使用）ハードウェア

本デバイスドライバで設定（使用）可能な周辺機能を次に示します。

3.2.1 機能（使用）端子

本デバイスドライバでの機能端子を次に示します。

表 3.2.1 設定端子機能

端子名	機能	備考
P0_3/FCN0TX	CAN送信	
P0_4/FCN0RX	CAN受信	
P1_1/TSG2001	相補PWM出力（U相ハイサイド）	インバータ0用
P1_2/TSG2002	相補PWM出力（U相ロウサイド）	インバータ0用
P1_3/TSG2003	相補PWM出力（V相ハイサイド）	インバータ0用
P1_4/TSG2004	相補PWM出力（V相ロウサイド）	インバータ0用
P1_5/TSG2005	相補PWM出力（W相ロウサイド）	インバータ0用
P1_6/TSG2006	相補PWM出力（W相ロウサイド）	インバータ0用
P1_7/CSIG0SI	シリアルデータ入力	RDCIC用等
P1_8/CSIG0SO	シリアルデータ出力	RDCIC用等
P1_9/CSIG0SC	シリアルクロック出力	RDCIC用等
P2_2/TSG2101	相補PWM出力（U相ハイサイド）	インバータ1用
P2_3/TSG2102	相補PWM出力（U相ロウサイド）	インバータ1用
P2_4/TSG2103	相補PWM出力（V相ハイサイド）	インバータ1用
P2_5/TSG2104	相補PWM出力（V相ロウサイド）	インバータ1用
P2_6/TSG2105	相補PWM出力（W相ロウサイド）	インバータ1用
P2_7/TSG2106	相補PWM出力（W相ロウサイド）	インバータ1用
ADCA0I1-6	同時S/H用アナログ入力	相電流等
ADCA0I7-22	アナログ入力（同時S/Hは無）	温度等
P3_0/TAUA1I0	インプットキャプチャ入力	
P3_2/CSIG1SI	シリアルデータ入力	RDCIC用等
P3_3/CSIG1SO	シリアルデータ出力	RDCIC用等
P3_4/CSIG1SC	シリアルクロック出力	RDCIC用等
P7_4/ENCA1E0	エンコーダ入力（A相）	
P7_5/ENCA1E1	エンコーダ入力（B相）	
P7_6/ENCA1EC	エンコーダ入力（Z相）	

3.2.2 設定（使用）周辺機能内容

本デバイスドライバでの主な周辺機能設定を次に示します。

表 3.2.2 主な周辺機能設定

周辺機能	機能設定			
A/Dコンバータ (ADCA0)	・12ビット分解能 / ・ディスチャージ有効 / ・バッファアンプ有効 / ・エラー割り込み発生無し CG2はPWMキャリア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
	同時サンプル・アンド・ホールド	無し	—	あり
	変換チャンネル内容	ch 7-22 : タイマと非同期	—	ch1-6: タイマ同期 (相電流等)
	AD変換の優先順位 CG2 > CG1 > CG0			
TSG2 (TSG20,TSG21)	HT-PWMモード（デッドタイム付加PWM） AD起動トリガ出力			
タイマ・アレイ・ユニットA (TAUA0)	インターバルタイマ設定（1MHz分解能） チャンネル0			
タイマ・アレイ・ユニットA (TAUA1)	キャプチャタイマ設定（1MHz分解能） チャンネル0			
エンコーダタイマ (ENCA1)	A,B相両エッジカウント、Z相の立ち上がりエッジでカウンタクリア			
シリアルI/F (CSIG0,1)	データ長 12Bit LSBファースト（送受信）			
CANコントローラ (FCN0)	送受信設定、IDマスク機能クリア、拡張IDは未対応 メッセージBOXアドレス（0xFF480000）			

4. デバイスドライバ関数

IPM モータ制御ソフトウェア用のデバイスドライバ関数について説明します。

4.1 関数の種類

デバイスドライバ関数は、次に示す関数準備しています。

表4.1 デバイスドライバ関数(1/2)

ファイル名 (暫定)	関数名	主な機能
io_ad.c	dpad_init	ADコンバータの初期化
io_CPU.c	cpu_init	CPU初期化 (SGA設定, 全割り込みマスク)
io_PIO.c	pio_init	IOポート、外部割り込み初期化
	intpx_status_read	外部割り込みステータス読み込み、外部割り込みフラグクリア
io_CSIG(n).c ((n)=0,1)	csig(n)_init	シリアルI/F初期化 (3線シリアルI/F)
	csig(n)_stop	シリアルI/F送受信禁止&レゾルバIC用 CS_O設定
	csig(n)_start	シリアルI/F送受信許可
	resolver_pos_read	レゾルバIC通信データリード (初期位置取得用)
io_ENCA1.c	enca1_init	エンコーダタイマ初期化
	u2_enca1_cnt_read	エンコーダカウント値取得
	enca1_stop	エンコーダタイマカウント停止
	enca1_start	エンコーダタイマカウント開始
	enca1_cnt_set	エンコーダカウント値設定 (カウント初期値設定)
	enca1_init_interrupt	エンコーダコンペアー一致割り込みレベル設定&割り込み許可
io_TAU0.c	taua0_init	TAU0ユニットチャンネル0初期化&カウント開始 インターバルタイマ機能設定
	taua1_cap_init	TAU1ユニットチャンネル0初期化&カウント開始 インプットキャプチャ機能設定
io_TSG2(n).c ((n)=0,1)	tsg2(n)_init	3相インバータ用タイマ初期化
	tsg2(n)_start	3相インバータ用タイマカウント開始
	tsg2(n)_stop	3相インバータ用タイマカウント停止
	dpinv(n)_start	3相PWM出力開始 (タイマは動作)
	dpinv(n)_stop	3相PWM出力停止Hi-Z出力 (タイマは動作)
	dpinv(n)_set_addelay	ADサンプルタイミングの設定 (Delayのみ)
	dpinv(n)_setuvwf	U,V,W相Duty, キャリア周波数の設定 (固定小数点用)
	dpinv(n)_setuvwf_f	U,V,W相Duty, キャリア周波数の設定 (浮動小数点用)
dpinv(n)_extgb_init	外部ゲートブロック (緊急PWM停止) 用周辺初期化	

4.2 デバイスドライバ関数の説明

4.2.1 A/D コンバータ関連

関数名：

`void dpad_init (void)`

【概要】

ADコンバータの初期化を行います。

【記述例】

```
dpad_init();
```

【事前設定】

特に必要なし

【引数】

なし

【戻り値】

なし

【備考】

各チャンネルの変換タイミングは、

- ・ ADCA0I1~ADCA0I6 : PWM の割り込みに同期、6CH 同時サンプリング
 - ・ ADCA0I7~ADCA0I22 : 連続スキャンモードにて常時変換
- 分解能は 12bit、変換時間は 1.563us に設定する。

4.2.2 CPU 関連

関数名:

```
void cpu_init( void )
```

【概要】

セーフティガーディアンの設定、及び、マスカブル割り込みを禁止（マスク）設定を行います。

【記述例】

```
cpu_init();
```

【事前設定】

特に必要なし

【引数】

なし

【戻り値】

なし

【備考】

本関数は main()ルーチンの先頭でコールすること。

4.2.3 ポート設定関連

関数名:

void pio_init (void)

【概要】

デジタルポートの入出力、外部割り込み設定を行います。

【記述例】

```
pio_init();
```

【事前設定】

特に必要なし

【引数】

なし

【戻り値】

なし

【備考】

ハードウェア（基板）により、修正が必要。

PWM出力等の周辺I/Oの設定は、各周辺設定で行う事。

4.2.4 エンコーダタイマ関連

関数名:

```
void enca1_init( unsigned short u2_enclcnt_max )
```

【概要】

エンコーダタイマ（ENCA1）の初期化を行います。

【記述例】

```
enca1_init(2048);
```

【事前設定】

特に必要なし

【引数】

u2_enclcnt_max:エンコーダカウント値のMAX値

【戻り値】

なし

【備考】

Z相によるカウンタクリア信号以外に、引数値とエンコーダカウント値が一致した場合もカウンタ値はクリアされます。また、アンダーフロー時は、引数値がカウンタ値にセットされます。

4.2.5 インターバルタイマ関連

関数名:

```
void taua0_init ( unsigned long int interval_us, unsigned char intlv )
```

【概要】

インターバルタイマの初期化を行います。

【記述例】

```
taua0_init(1000, 8);
```

インターバル時間 1000 μ S

割り込み優先レベル 8

【事前設定】

特に必要なし

【引数】

interval_us : タイマ周期[μ s]

intlv : 割り込み優先レベル

【戻り値】

なし

【備考】

分解能 1 μ S

4.2.6 インพุットキャプチャタイマ関連

関数名:

```
void tau1_cap_init ( void )
```

【概要】

インพุットキャプチャタイマの初期化を行います。

【記述例】 tau1_cap_init();

【事前設定】

特に必要なし

【引数】

なし

【戻り値】

なし

【備考】

分解能 1 μ S

SAMPLE

4.2.7 三相 PWM 関連

関数名:

```
void tsg20_init ( unsigned long int freq, unsigned long int deadtime_on,
                unsigned long int deadtime_off, unsigned char logic,
                unsigned char trigsrsrc, unsigned char intlv)
```

【概要】

3相PWM出力用タイマの初期設定を行います。

【記述例】

```
ts20_init(10000, 3500, 3500, 0, 3, 1)
```

キャリア周波数:10KHz, デッドタイム:3.5uS, ハイアクティブ, 山谷割り込み,
割り込み優先レベル:1

【引数】

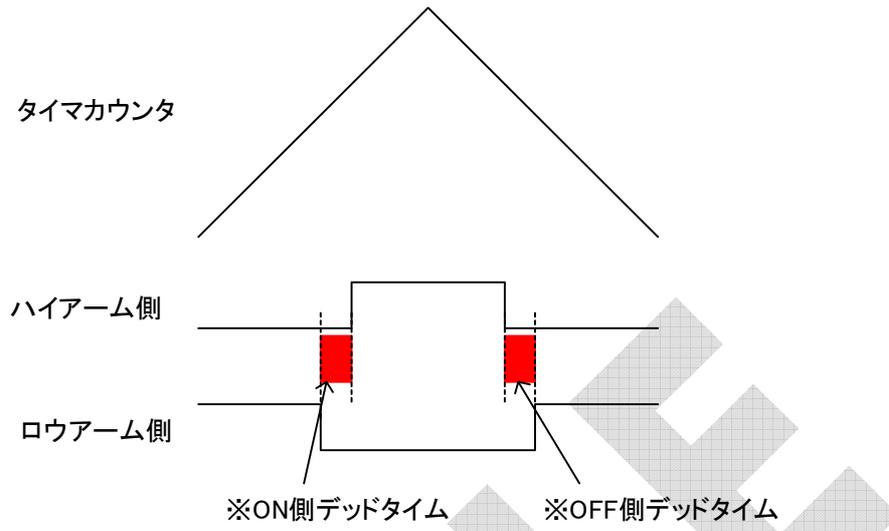
freq	:	PWMキャリア周波数[Hz]
deadtime_on	:	ON側デッドタイム[nS]
deadtime_off	:	OFF側デッドタイム[nS]
logic	:	アクティブレベル 0:ハイアクティブ/1:ロウアクティブ
trigsrsrc	:	割り込み位置 0:割り込みなし (リロードは、山谷) 1:山割り込み (リロードは山) 2:谷割り込み (リロードは谷) 3:山谷割り込み (リロードは、山谷)
intlv	:	割り込み優先レベル

【戻り値】

なし

【備考】

tsg21_init関数も同様 (出力端子TSG2101-6)



SAMPLE

4.2.8 CAN 関連

関数名:

```
unsigned char can_init(unsigned long int ch, unsigned long int brp,  
                      unsigned long int tseg1, unsigned long int tseg2,  
                      unsigned long int sjw)
```

【概要】

指定チャンネルのCAN機能の初期化を行います。

【記述例】

```
status = can_init(0, 128, 8, 4, 2);
```

【事前設定】

特に必要なし

【引数】

ch : CAN チャンネルの指定
brp : ビットレートプリスケアラ
tseg1 : タイムセグメント 1 設定
tseg2 : タイムセグメント 2 設定
sjw : 同期ジャンプ幅の長さ

【戻り値】

0 : 正常終了
1 : デバイスタイムアウト。または、パラメータ設定範囲外

【備考】

入力引数範囲は以下の値とする。

$0 \leq ch \leq 1$

$1 \leq brp \leq 255$

$0 \leq tseg1 \leq 15$

$0 \leq tseg2 \leq 7$

$0 \leq sjw \leq 3$