

V850E2/ML4

R01AN1224JJ0100

Rev.1.00

2012.06.22

相補 PWM 出力機能編

要旨

本アプリケーションノートでは、V850E2/ML4の相補 PWM 出力機能の r f 設定方法、およびサンプルコードの動作概要や使用方法を説明します。

[機能・動作]の特長を以下に示します。

- ・ 6層相補 PWM 出力、間引きタイマ使用、デッドタイム付
- ・ 120度(3相)通電によるブラシレス DC モータ制御
- ・ PID(比例要素、積分要素、微分要素)制御による、モータのフィードバック制御(モータの場合、微分要素は使いません。)
- ・ A/D コンバータを用いた VR スイッチによるモータ速度変更

対象デバイス

V850E2/ML4

開発環境

CubeSuite+, GHS MULTI V5.1.7D、IAR for V850 Kickstart V3.80

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様.....	3
2. 動作確認条件.....	5
3. ハードウェア説明.....	6
3.1 ハードウェア構成例	6
3.2 使用端子一覧.....	7
4. ソフトウェア説明.....	8
4.1 動作概要.....	8
4.2 必要メモリサイズ.....	10
4.3 ファイル構成.....	11
4.4 オプション設定メモリ	12
4.5 定数一覧.....	13
4.6 変数一覧.....	14
4.7 関数一覧.....	15
4.8 関数仕様.....	16
4.9 フローチャート	21
4.9.1 メイン処理	21
4.9.2 初期値設定	22
4.9.3 PWM タイマ設定	23
4.9.4 1msec インタバル・タイマ初期化	24
4.9.5 1sec フリーラン・タイマ初期化	24
4.9.6 A/D コンバータ初期化.....	25
4.9.7 割込み処理：キャリア割り込み(50usec)毎のモータ制御	26
4.9.8 割込み処理：ホールセンサ割り込み(電気角 60°)毎の処理	27
4.9.9 割込み処理：フリーラン・タイマのオーバフロー(1sec)毎の処理	27
4.9.10 割込み処理：インタバル・タイマのオーバフロー(1msec)毎の処理.....	28
5. サンプルコード	29
6. 参考ドキュメント.....	29

1. 仕様

このサンプルコードには、次の特徴があります。

- デッドタイム、間引きタイマ付の 6 相相補 PWM 波発生
- 120 度 (3 相) 通電によるブラシレス DC モータ制御
- PID(比例要素、積分要素、微分要素)制御による、モータのフィードバック制御(モータの場合、微分要素は使わない)
- A/D コンバータを用いた VR スイッチによるモータ速度変更

タイマの仕様は以下のとおりです。

タイマ	機能
TAUA0 チャンネル 0	マスタチャンネル、25usec 間隔
TAUA0 チャンネル 1	間引きタイマ、スレーブ、1 回間引き、50usec 間隔で割込み(キャリア割込み)発生。 キャリア割込みで、モータの制御を行います。
TAUA0 チャンネル 2, 4, 6	それぞれ PWM の U,V,W のハイサイド出力端子の値を決定します。 TAUA0CDRn の値によりデューティ比が決まります。 スレーブ、アップダウンカウンタ、マスタのアップダウン出力をトリガに設定します。
TAUA0 チャンネル 3, 5, 7	それぞれ PWM の U,V,W のロウサイド出力端子の値を決定します。 TAUA0CDRn の値によりデッドタイムが決まります。 スレーブ、ワンカウントモード、デッドタイム出力をトリガに設定します。
TAUA0 チャンネル 8	1msec で割り込みを発生させます。
TAUA0 チャンネル 9	フリーランカウンタ、1sec で割り込みを発生させます。

表 1.1 に使用する周辺機能と用途を、図 1.1 に使用例を示します。

表1.1 使用する周辺機能と用途

周辺機能	用途
ポート(P4_3, P4_4)	LED に接続し、LED の点灯を制御
INTP5	U 相ホールセンサからの入力を受け、割込みを発生します。
INTP6	V 相ホールセンサからの入力を受け、割込みを発生します。
INTP7	W 相ホールセンサからの入力を受け、割込みを発生します。
A/D コンバータ	VR スイッチからのアナログ信号を、ANI08 で受け、A/D 変換を行います。

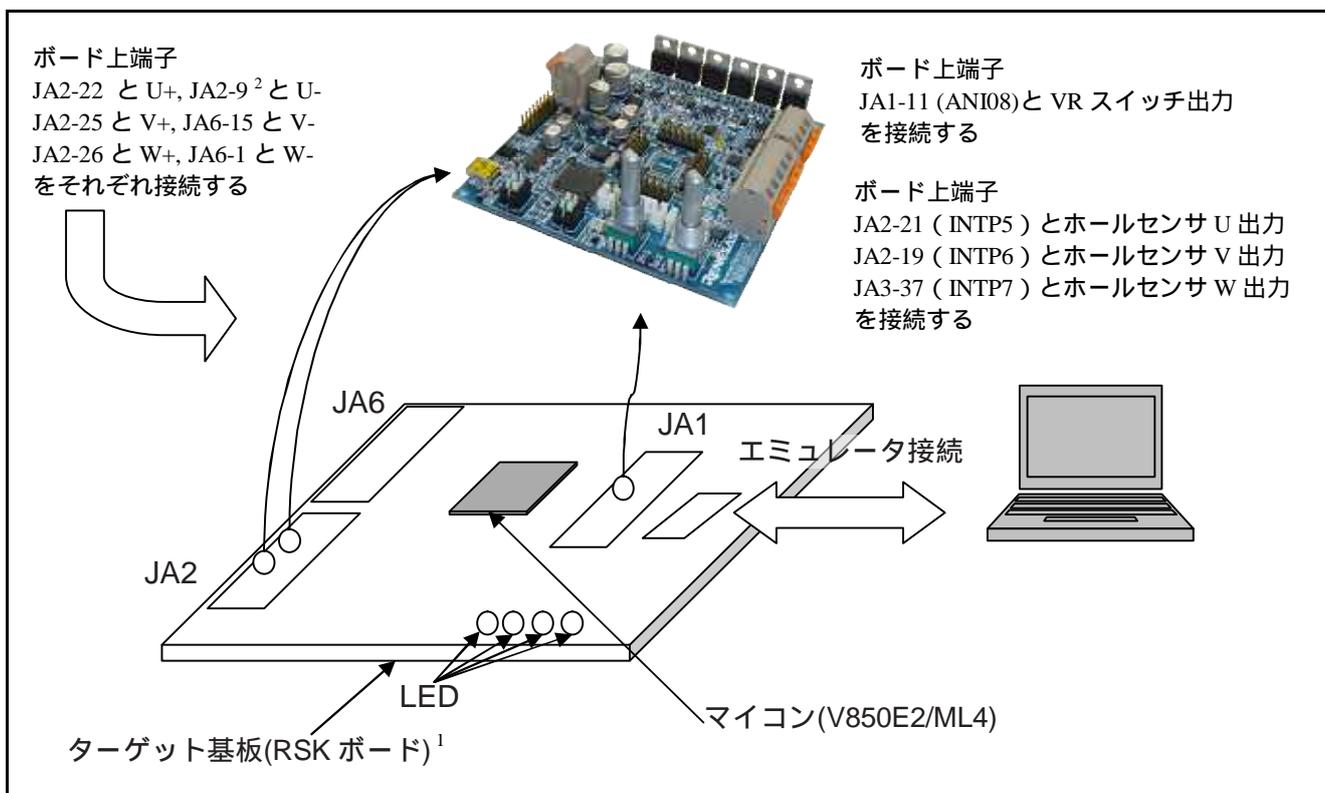


図1.1 使用例

¹ RSK ボードは 2012 年 8 月量産予定

² RSK ボード JA2-9 は出荷時未接続、0 抵抗で接続可

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1 動作確認条件

項目	内容
使用マイコン	V850E2/ML4PWM
動作周波数	200MHz(発振 10MHz × PLL 20 逡倍)
動作電圧	3.3V
統合開発環境	CubeSuite+ V1.00
	GHS MULTI V5.1.7D
	IAR for V850 Kickstart V3.80.1
C コンパイラ	CX V1.20(CubeSuite+)、最適化：デフォルト
	C-V850E 5.1.7 RELEASE(GHS MULTI)、最適化：デフォルト
	IAR C/C++ Compiler for V850 3.80.1 [Kickstart] (3.80.1.30078) 、最適化：デフォルト
動作モード	通常動作モード
サンプルコードのバージョン	V1.00
使用ボード	RSK ボード
使用デバイス	E1 エミュレータもしくは MINICUBE
使用ツール	なし

3. ハードウェア説明

3.1 ハードウェア構成例

図 3.1 にハードウェア構成を示します。

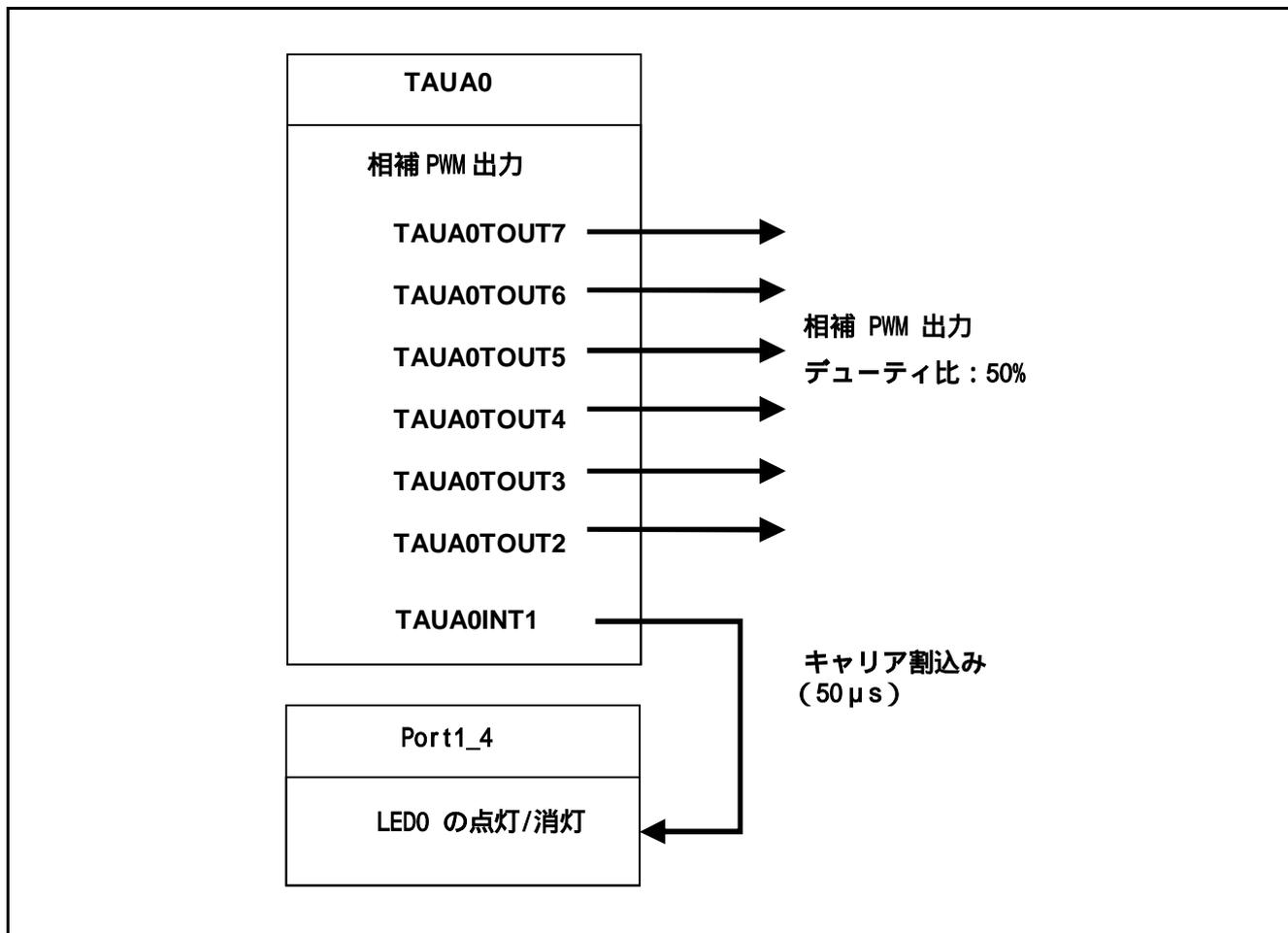


図3.1 ハードウェア構成

3.2 使用端子一覧

表 3.1 に使用端子と機能を示します。

表3.1 使用端子と機能

端子名	入出力	内容
PORT P4_3	出力	ポート・モード、出力、LED2
PORT P4_4	出力	ポート・モード、出力、LED3
P1_2/D18/TA0_I2/TE0_TI1/INTP7/TA0_O2	出力	相補 PWM 出力 (U 相ハイサイド)
P1_3/D19/TA0_I3/INTP8/TA0_O3/PPON	出力	相補 PWM 出力 (U 相ロウサイド)
P1_4/D20/TA0_I4/TE0_AI/INTP9/TA0_O4	出力	相補 PWM 出力 (V 相ハイサイド)
P1_5/D21/TA0_I5/INTP10/TA0_O5	出力	相補 PWM 出力 (V 相ロウサイド)
P1_6/D22/TA0_I6/TE0_BI/INTP11/TA0_O6	出力	相補 PWM 出力 (W 相ハイサイド)
P1_7/D23/TA0_I7/INTP12/TA0_O7	出力	相補 PWM 出力 (W 相ロウサイド)
P1_0/D16/TA0_I0/TE0_TI0/INTP5/TA0_O0	入力	U 相ホールセンサ割込み
P1_1/D17/TA0_I1/OCI/INTP6/TA0_O1	入力	V 相ホールセンサ割込み
P4_0/A16/INTP7/TA1_I8/TA1_O8/CSI0F_CS2	入力	W 相ホールセンサ割込み
P8_2/ANI08	入力	VR スイッチからの入力

4. ソフトウェア説明

4.1 動作概要

ソフトウェアの動作の概要を次の図に示します。main()で各種初期化関数を呼び、割り込み待ち状態に入ります。

図 4.1にシーケンス(初期化)を示します。

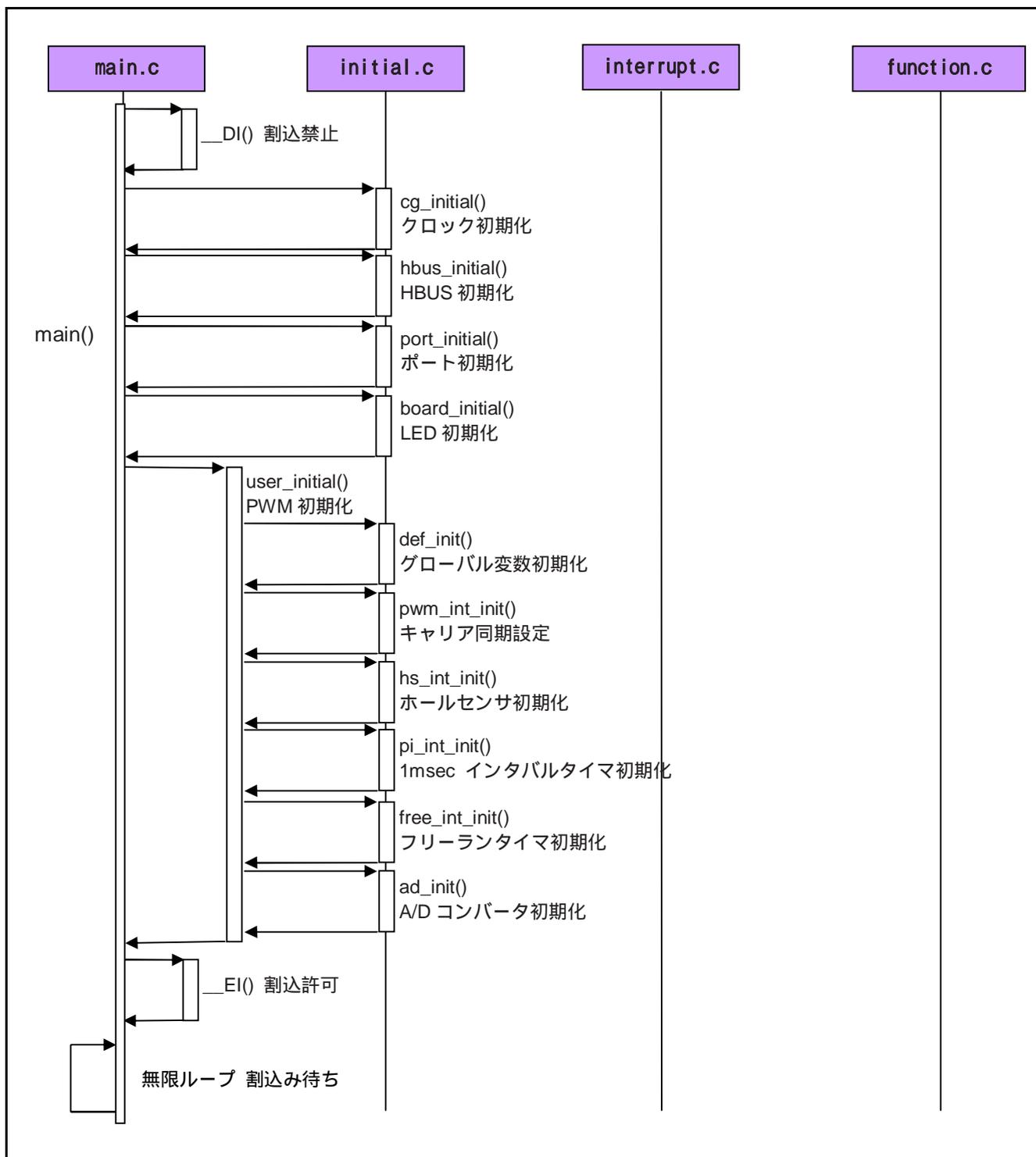


図4.1 シーケンス(初期化)

図 4.2にシーケンス(割り込み処理)を示します。

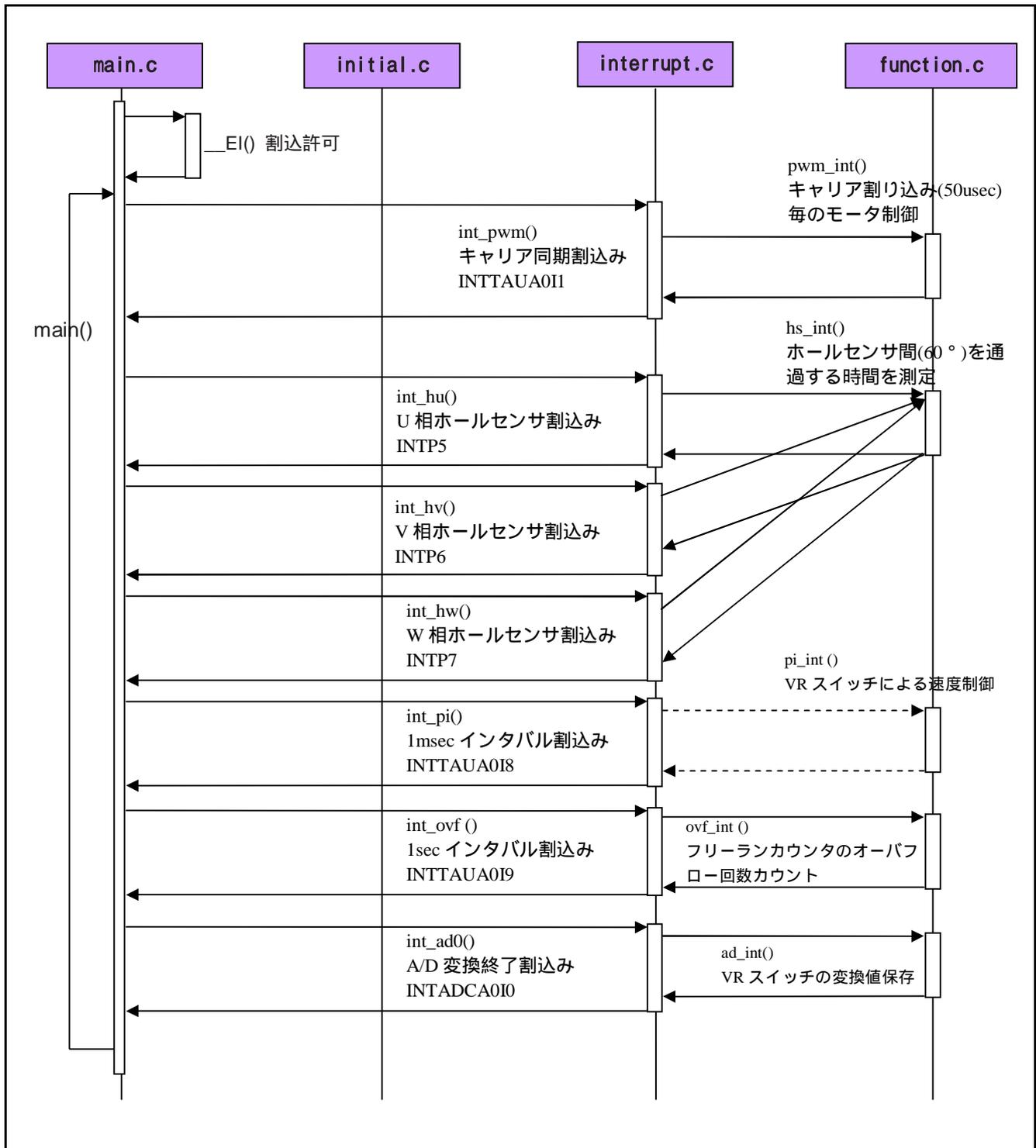


図4.2 シーケンス(割り込み処理)

4.2 必要メモリサイズ

表 4.1に必要メモリサイズを示します。(CubeSuite+、最適化オプション = デフォルトで測定)

表4.1 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	9056	CubeSuite+の生成する map ファイルに出力された ROM 領域で使用するサイズ
RAM	4152	CubeSuite+の生成する map ファイルに出力された RAM 領域で使用するサイズ
最大使用ユーザスタック	8	CubeSuite+のスタック見積もりツールで算出
最大使用割り込みスタック	588	同上

【注】 必要メモリサイズはCコンパイラのバージョンやコンパイルオプションにより異なります。

4.3 ファイル構成

表 4.2にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表4.2 サンプルコードで使用するファイル

ファイル名	概要	備考
crtE.s	ハードウェア初期化処理	CubeSuite+でのみ使用
startup.s		GHS MULTI でのみ使用
V850E2ML4.dir	リンク・ディレクティブ・ファイル	CubeSuite+でのみ使用
V850E2 ML4 CSIH.ld		GHS MULTI でのみ使用
vector.s	ベクタ・テーブル	GHS MULTI でのみ使用
csih.h	変数、関数宣言	
df4022_800.h	V850E2/ML4 用レジスタマクロ宣言	GHS MULTI でのみ使用
V850E2ML4PortRegister.h	portconfig.c 用ヘッダファイル	
main.c	メイン処理	
initial.c	ソフトウェア初期化処理	
function.c	PWM 割り込み処理	
sin_table180.c	Sin 関数テーブル	
interrupt.c	割り込み処理	
portconfig.c	ポート初期化	

4.4 オプション設定メモリ

本サンプルでは、オプション・バイトの設定は行っていません。必要に応じて設定してください。

4.5 定数一覧

表 4.3にサンプルコードで使用する定数を示します。

表4.3 サンプルコードで使用する定数

定数名	設定値	内容
const unsigned short u2_sintbl_U30[]	{ 0, 2, 3, 5, 7, 9,...}	sin テーブル 0° ~ 29°
const unsigned short u2_sintbl_U60[]	{50, 52, 53, 54, 56,...}	sin テーブル 30° ~ 89°

4.6 変数一覧

表 4.4 にグローバル変数を示します。

表4.4 グローバル変数

型	変数名	内容	使用関数
unsigned char	flg_tm9_ovf;	TAUA0_9 のオーバフローフラグ	pwm_int() ovf_int() def_init();
unsigned short	u2_pwm_tm9_cnt;	PWM キャリア割り込みでの TAUA0_9 のカウント値	pwm_int()
unsigned char	u1_hs_st	PWM キャリア割り込みでの ホールセンサのステータス	pwm_int()
unsigned short	u2_swing_pwm	PWM 出力の最大ふり幅	pwm_int() pi_int() def_init()
unsigned short	u2_hs_tm9_cnt	ホールセンサ割り込みでの TAUA0_9 のカウント	pwm_int() hs_int()
unsigned short	u2_hs_tm9_cnt_old	前回のホールセンサ割り込みでの TAUA0_9 のカウント	hs_int() def_init()
unsigned short	u2_hs_tm9_cnt_dif	TAUA0_9 カウントの前回との差	hs_int() pi_int()
unsigned short	u2_vr_st	VR スイッチによる最大ふり幅 カウント減算値	pi_int() ad_int() def_init()
unsigned short	u2_order_rpm	角速度指令値(rpm)	pi_int() def_init()
unsigned short	u2_rpm	角速度(rpm)	pi_int()
unsigned short	u2_ang_us	角速度(angle/usec)*2 ¹⁶ (不動 小数点演算回避用)	def_init() pwm_int() pi_int()
unsigned long	u4_temp_cal_pi	PI 制御演算用変数	pi_int()

4.7 関数一覧

表 4.5に関数を示します。

表4.5 関数

関数名	概要
void main(void)	各初期化処理関数を呼び出したあと、永くループに入る
void port_initial(void)	ポート・モードの設定を行う
void PortConfiguration0(void)	ポートグループ 0 の設定
void PortConfiguration1(void)	ポートグループ 1 の設定
void PortConfiguration2(void)	ポートグループ 2 の設定
void PortConfiguration3(void)	ポートグループ 3 の設定
void PortConfiguration4(void)	ポートグループ 4 の設定
void PortConfiguration5(void)	ポートグループ 5 の設定
void PortConfiguration6(void)	ポートグループ 6 の設定
void PortConfiguration7(void)	ポートグループ 7 の設定
void PortConfiguration8(void)	ポートグループ 8 の設定
void cg_initial(void)	特殊クロック周波数制御レジスタの初期化設定を行う
void hbus_initial(void)	AHB バスの初期化設定を行う
void board_initial(void)	LED 初期状態の設定を行う
void user_initial(void)	モータ制御用割り込み設定
void def_init(void)	モータ制御用変数の初期値設定
void pwm_int_init(void)	PWM 機能初期設定&スタート
void hs_int_init(void)	U,V,W 相ホールセンサ割り込み設定
void pi_int_init(void)	インタバル・タイマ初期設定
void free_int_init(void)	フリーラン・タイマ初期設定
void ad_init(void)	VR スイッチ用 A/D コンバータ初期設定
interrupt void int_pwm(void)	PWM 周期同期割り込み(50us)
interrupt void int_hu(void)	ホールセンサ U 割り込み
interrupt void int_hv(void)	ホールセンサ V 割り込み
interrupt void int_hw(void)	ホールセンサ W 割り込み
interrupt void int_pi(void)	1msec インタバル・タイマ割り込み(1msec)
interrupt void int_ovf(void)	フリーランカウンタオーバーフロー割り込み(1s)
interrupt int_ad0(void)	A/D コンバータ変換終了割り込み
void pwm_int(void)	キャリア割り込み(50usec)毎のモータ制御
void hs_int(void)	速度制御用に電気角 60 度毎の時間を計測
void ovf_int(void)	フリーランカウンタのオーバーフローをカウント
void pi_int(void)	1msec インタバル割り込み
void ad_int(void)	VR1 スイッチのアナログ変換値取得

4.8 関数仕様

サンプルコードの関数仕様を示します。

main()	
概要	メイン関数、最初に呼び出される関数
ヘッダ	-
宣言	void main(void)
説明	各初期化処理関数を呼び出したあと、永久ループに入り、CSIH 割り込みを待つ
引数	-
リターン値	-

port_initial()	
概要	ポート・モードの設定を行う
ヘッダ	pwm.h
宣言	void port_initial (void)
説明	LED、モータホールセンサ、A/D コンバータで使用するポート機能の初期化を行う。
引数	-
リターン値	-

PortConfiguration0() ~ PortConfiguration8()	
概要	ポート・グループごとのポート・モードの設定を行う
ヘッダ	V850E2ML4PortRegister.h
宣言	void PortConfiguration0(void)...void PortConfiguration8(void)
説明	port_initial()より呼ばれ、ポートグループごとの設定を行う。
引数	-
リターン値	-

cg_initial ()	
概要	クロック初期化
ヘッダ	csih.h
宣言	void cg_initial(void)
説明	特殊クロック周波数制御レジスタの初期化設定を行う
引数	-
リターン値	-

hbus_initial()	
概要	H バス初期化
ヘッダ	pwm.h
宣言	void hbus_initial(void)
説明	AHB バスの初期化設定を行う
引数	-
リターン値	-

board_initial()	
概要	ボード初期化
ヘッダ	pwm.h
宣言	void board_initial(void)
説明	LED 初期状態の設定を行う
引数	-
リターン値	-
user_initial()	
概要	モータ制御用割り込み設定
ヘッダ	pwm.h
宣言	void user_initial(void)
説明	モータ制御用の割り込み設定と各種初期化関数を呼び出す。
引数	-
リターン値	-
def_init()	
概要	モータ制御用変数の初期値設定
ヘッダ	pwm.h
宣言	void def_init(void)
説明	モータ制御用変数の初期値設定
引数	-
リターン値	-
pwm_int_init()	
概要	タイマの PWM 設定
ヘッダ	pwm.h
宣言	void pwm_int_init(void)
説明	タイマ TAU0 の各チャンネルを設定し、PWM 波を発生させる。
引数	-
リターン値	-
hs_int_init()	
概要	ホールセンサ割り込み設定
ヘッダ	pwm.h
宣言	void hs_int_init(void)
説明	ホールセンサ割り込み(INTP5, INTP6, INTP7)のマスク解除
引数	-
リターン値	-
pi_int_init()	
概要	1msec インタバル・タイマ初期化
ヘッダ	pwm.h
宣言	void pi_int_init(void)
説明	TAUA0_8 を 1msec インタバルタイマに設定する。

引数	-	-
リターン値	-	

free_int_init()

概要	フリーランカウンタ初期化
ヘッダ	pwm.h
宣言	free_int_init(void)
説明	TAUA0_9 を 1sec フリーランカウンタ設定する。
引数	-
リターン値	-

ad_init()

概要	A/D コンバータ初期化
ヘッダ	pwm.h
宣言	void ad_init(void)
説明	A/D コンバータを初期化し、変換を開始する
引数	-
リターン値	-

int_pwm()

概要	PWM キャリア割り込み(INTTAUA011, 50usec 毎)
ヘッダ	-
宣言	__interrupt void int_pwm(void)
説明	割り込み処理 pwm_int() を呼び出す
引数	-
リターン値	-

int_hu()

概要	ホールセンサ U 割り込み(INTP5)
ヘッダ	-
宣言	__interrupt void int_hu(void)
説明	割り込み処理 hs_int()を呼び出す
引数	-
リターン値	-

int_hv()

概要	ホールセンサ V 割り込み(INTP6)
ヘッダ	-
宣言	__interrupt void int_hv(void)
説明	割り込み処理 hs_int()を呼び出す
引数	-
リターン値	-

int_hw()

概要	ホールセンサ W 割込み(INTP7)
ヘッダ	-
宣言	__interrupt void int_hw(void)
説明	割込み処理 hs_int()を呼び出す
引数	-
リターン値	-

int_pi()

概要	1msec インタバル・タイマ割込み(INTTAUA018)
ヘッダ	-
宣言	__interrupt void int_pi(void)
説明	割込み処理 pi_int()を呼び出す
引数	-
リターン値	-

int_ovf()

概要	1sec インタバル・タイマ割込み(INTTAUA019)
ヘッダ	-
宣言	__interrupt void int_ovf (void)
説明	割込み処理 ovf_int()を呼び出す
引数	-
リターン値	-

int_ad0()

概要	A/D コンバータ変換終了割込み
ヘッダ	-
宣言	__interrupt void int_ad0(void)
説明	割込み処理 ad_int()を呼び出す
引数	-
リターン値	-

pwm_int ()

概要	キャリア割り込み(50usec)毎のモータ制御
ヘッダ	pwm.h
宣言	void pwm_int(void)
説明	ホールセンサからキャリア割込み間の時間から電気角を推定し、PWM の設定を変更して通電パターンを変更する
引数	-
リターン値	-

hs_int ()

概要	速度制御用に電気角 60 度毎の時間を計測
ヘッダ	pwm.h
宣言	void hs_int(void)

説明	U,V,W の各ホールセンサ割り込みにて、タイマにより前回の割り込みからの時間を計測する。
引数	-
リターン値	-

ovf_int ()

概要	フリーランカウンタ割り込み処理
ヘッダ	pwm.h
宣言	void ovf_int(void)
説明	フリーランカウンタのオーバフロー回数 flg_tm9_ovf をインクリメントする。
引数	-
リターン値	-

pi_int ()

概要	1msec インタバル・タイマ割り込み (現在未使用)
ヘッダ	pwm.h
宣言	void pi_int(void)
説明	VR スイッチの A/D 変換値から、角速度指令値(u2_order_rpm)を算出する。
引数	-
リターン値	-

ad_int()

概要	A/D 変換割り込み
ヘッダ	pwm.h
宣言	void ad_int(void)
説明	VR1 スイッチのアナログ変換値取得
引数	-
リターン値	-

4.9 フローチャート

4.9.1 メイン処理

図 4.3にメイン処理のフローチャートを示します。

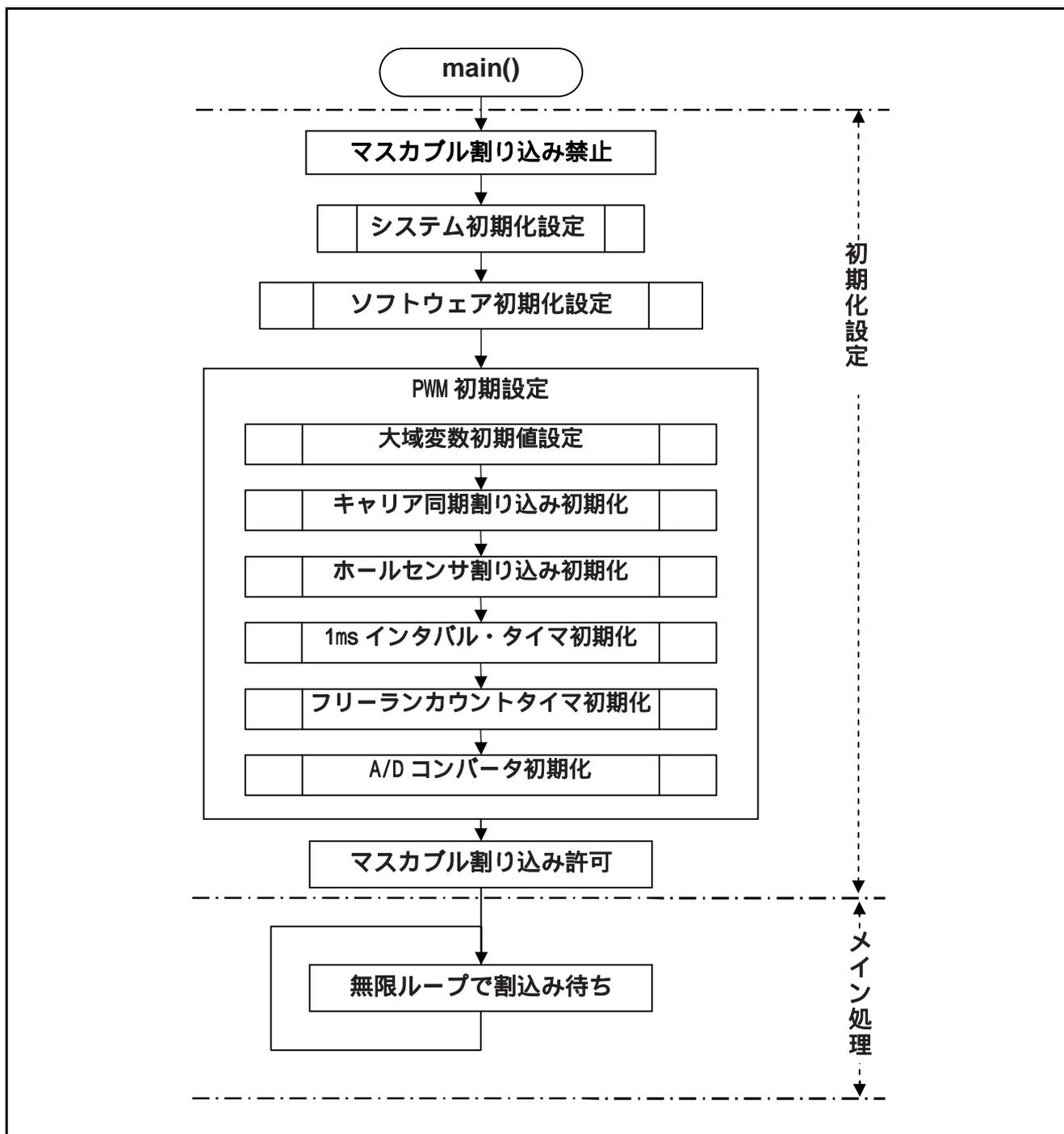


図4.3 メイン処理

4.9.2 初期値設定

関数 `def_init()`にて、タイマのクロック設定、各変数の初期値を設定します。

図 4.4に初期値設定のフローチャートを示します。

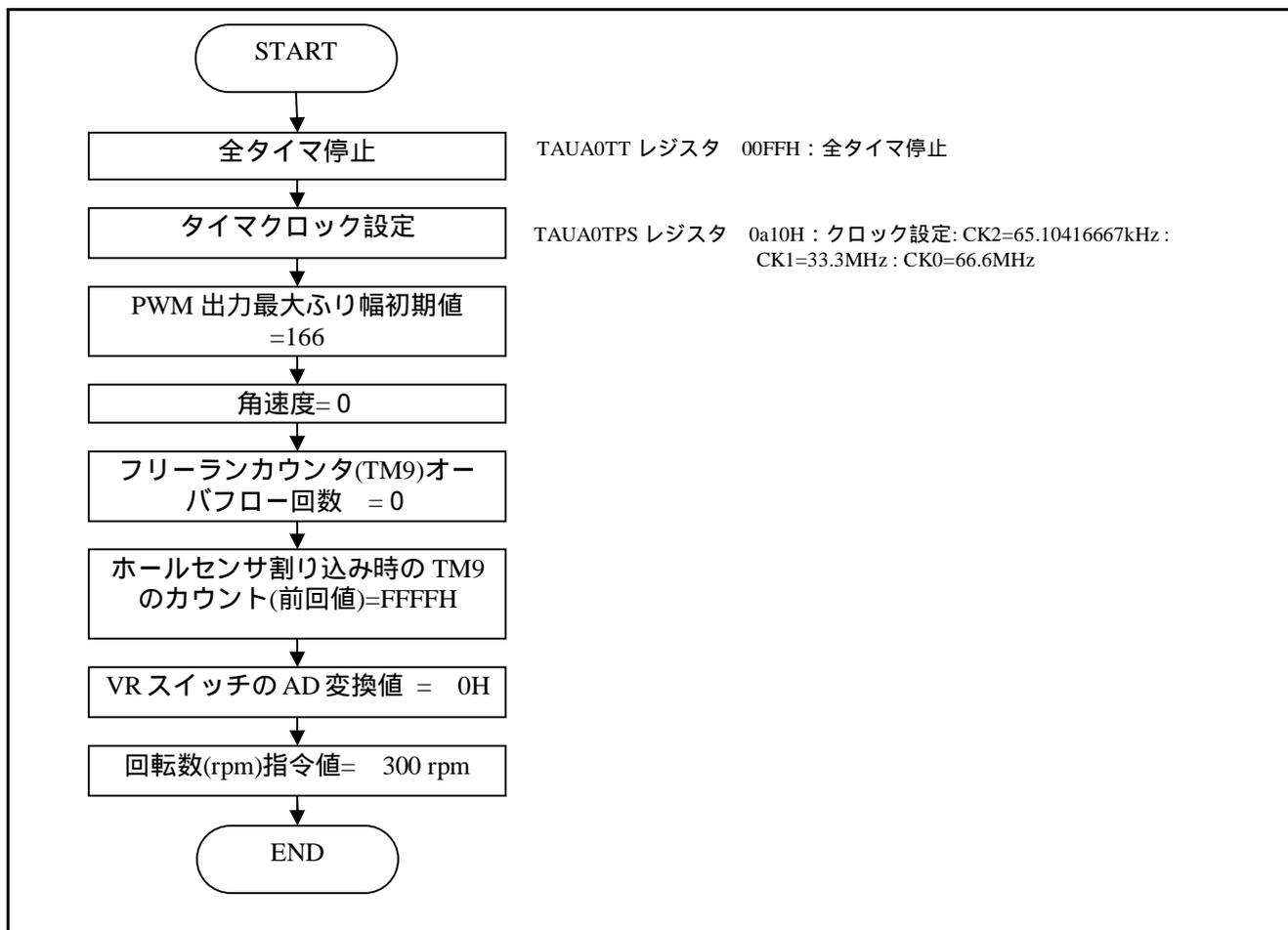


図4.4 初期値設定

4.9.3 PWM タイマ設定

関数 `pwm_int_init()` にて、PWM タイマを設定します。

図 4.5 に PWM タイマ設定のフローチャートを示します。

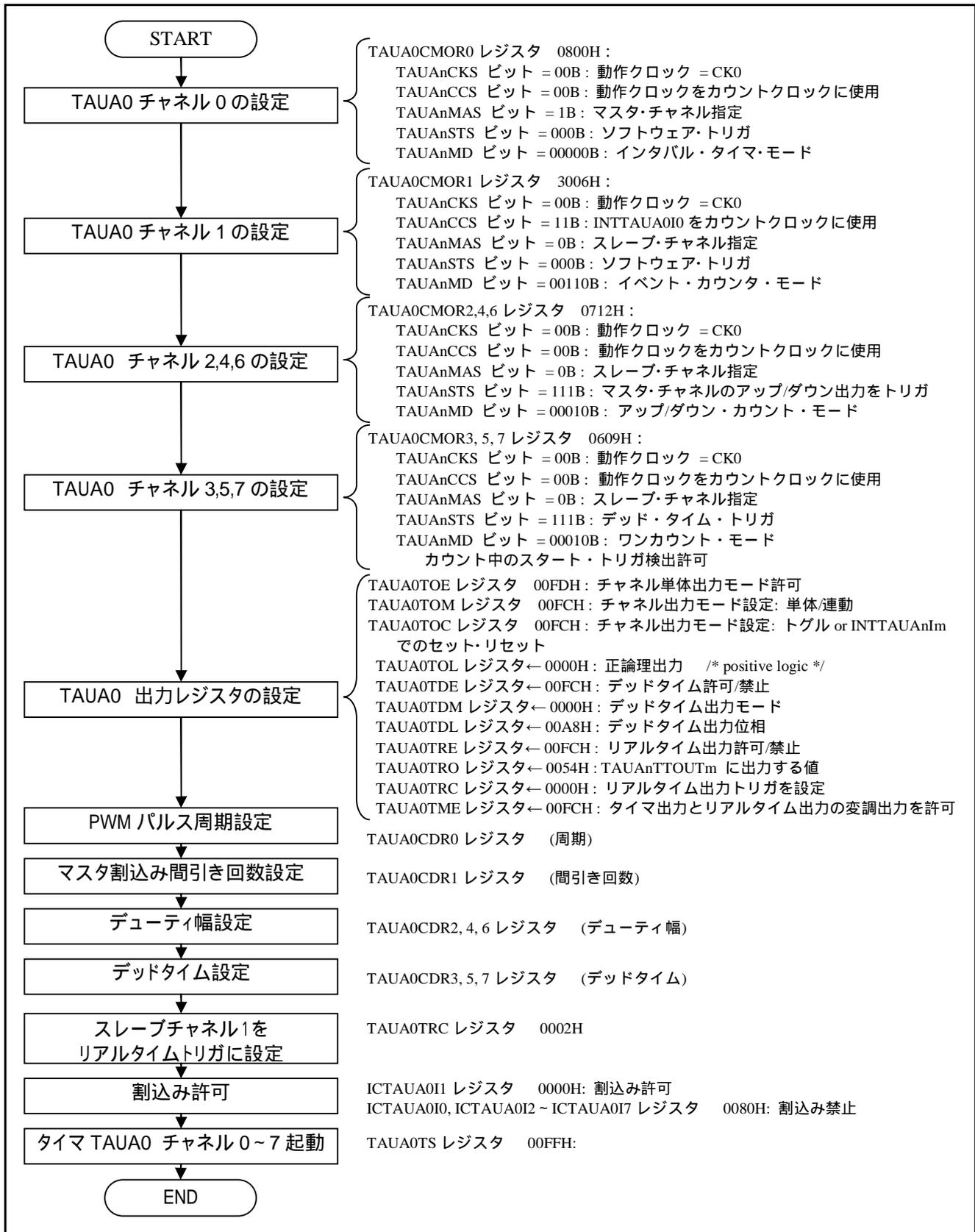


図4.5 PWM タイマ設定

4.9.4 1msec インタバル・タイマ初期化

TAUA0 タイマのチャンネル 8 を 1msec ごとに割り込みが発生するように設定します。

図 4.6 に 1msec インタバル・タイマ初期化のフローチャートを示します。

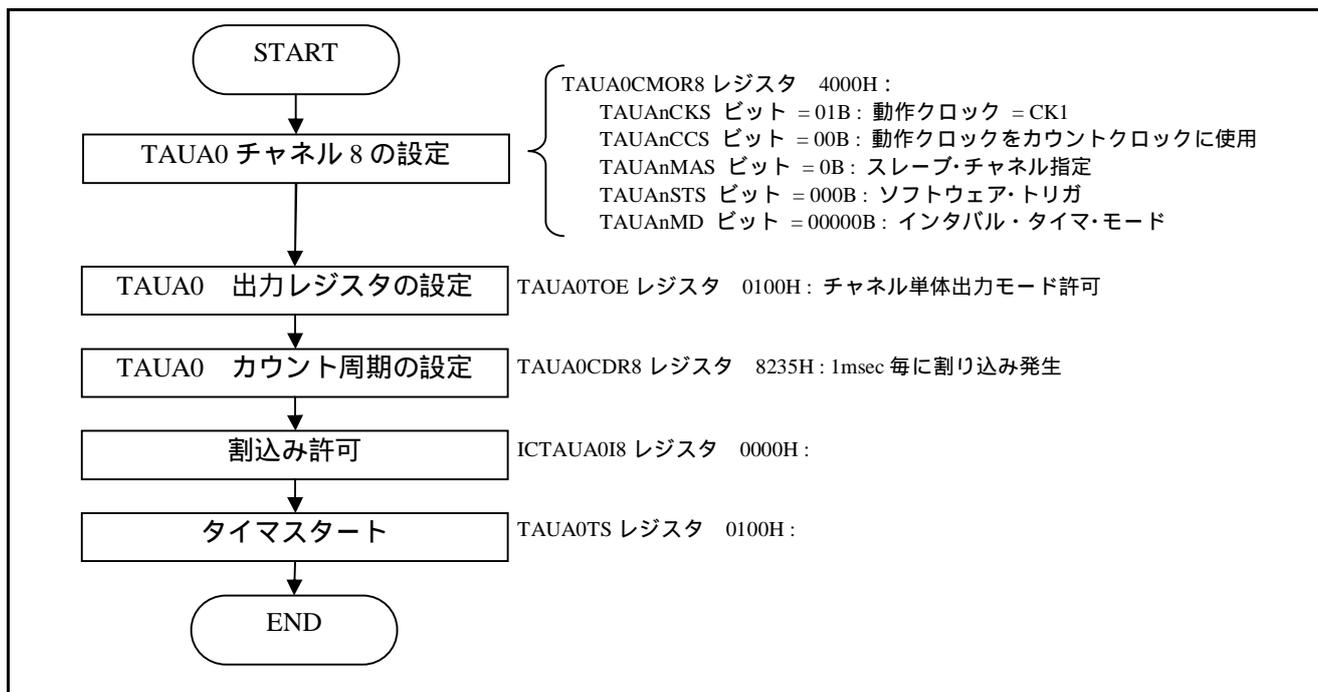


図4.6 1msec インタバル・タイマ初期化

4.9.5 1sec フリーラン・タイマ初期化

TAUA0 タイマのチャンネル 9 を 1sec ごとに割り込みが発生するように設定します。

図 4.7 に 1sec フリーラン・タイマ初期化のフローチャートを示します。

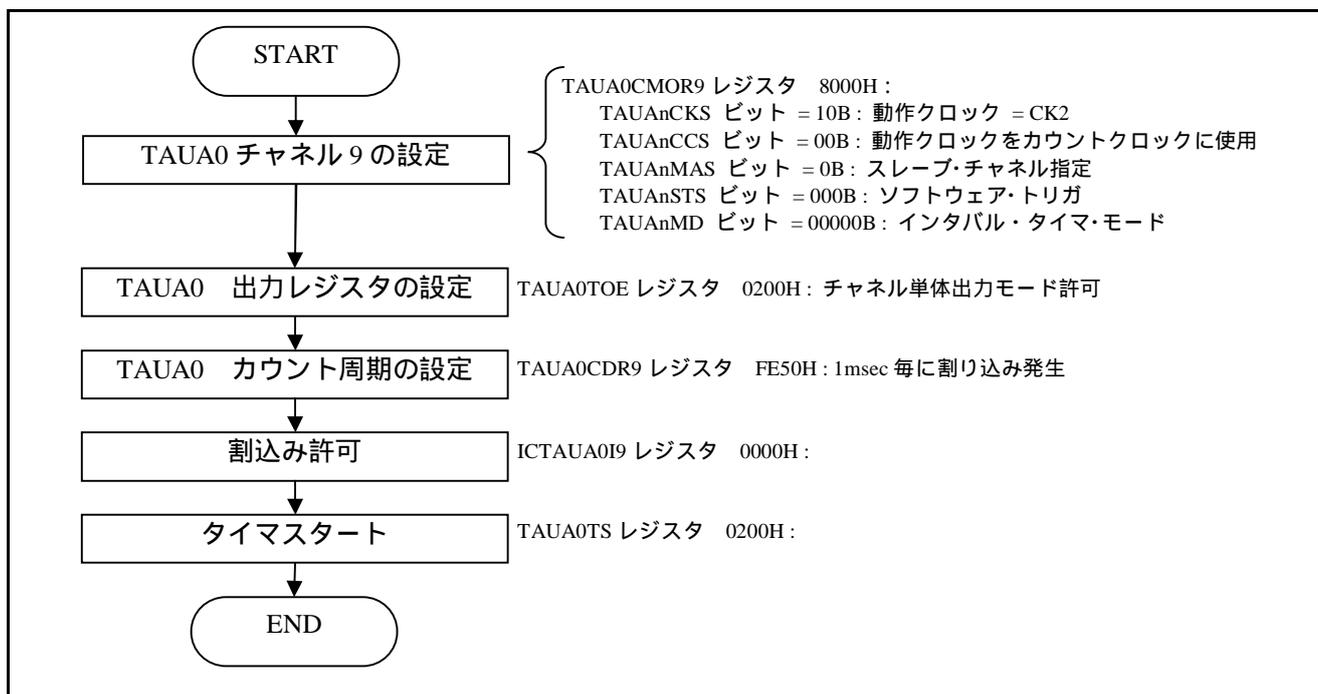


図4.7 1sec フリーラン・タイマ初期化

4.9.6 A/D コンバータ初期化

VR スイッチ(ANI08)からの入力値を A/D 変換するよう設定します。

図 4.8にA/D コンバータ初期化のフローチャートを示します。

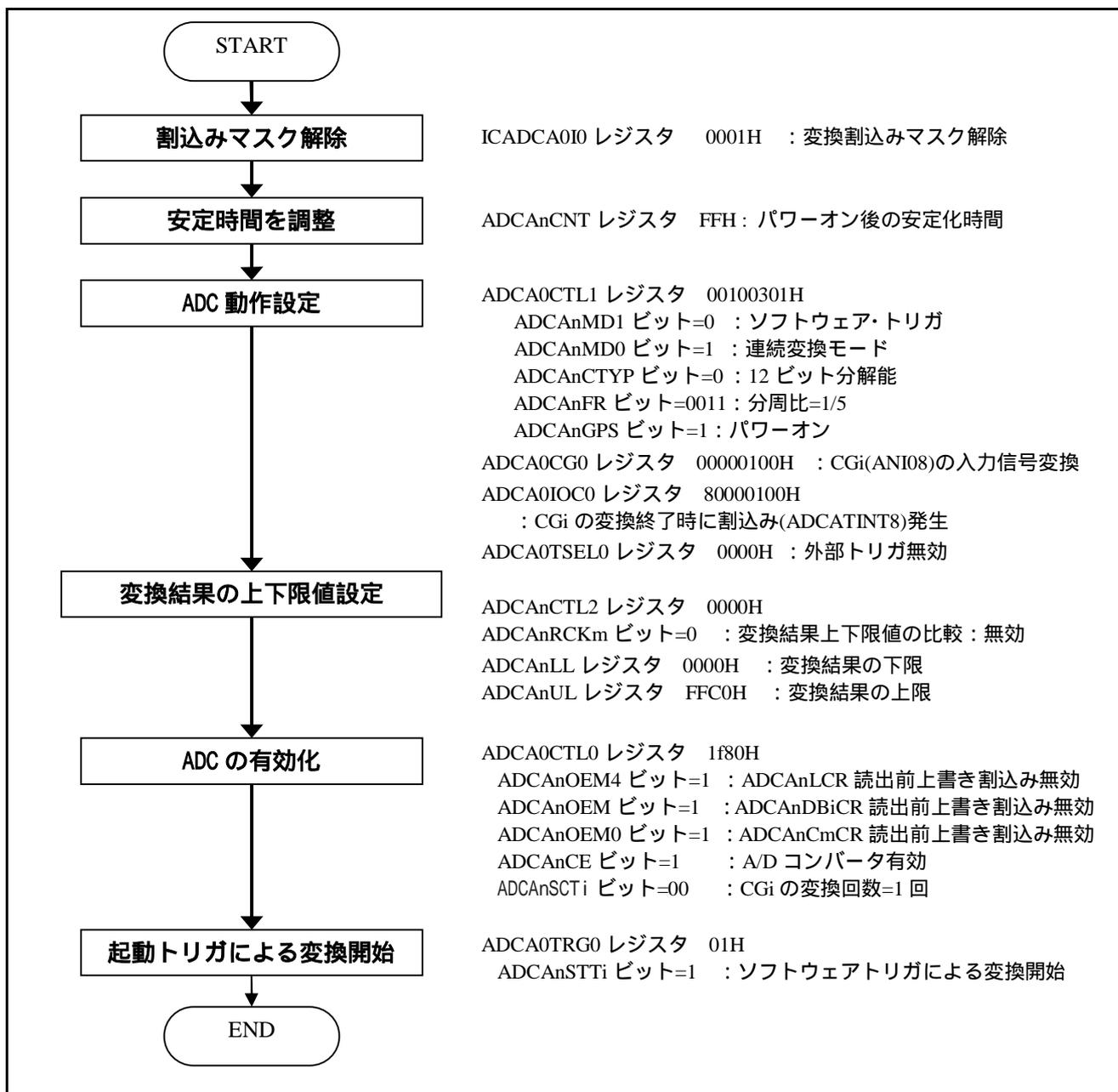


図4.8 A/D コンバータ初期化

4.9.7 割込み処理：キャリア割り込み(50usec)毎のモータ制御

関数 `pwm_int()` にて、キャリア割り込みごとに、PWM 変調の処理を行います。

図 4.9にキャリア割り込み(50usec)毎のモータ制御のフローチャートを示します。

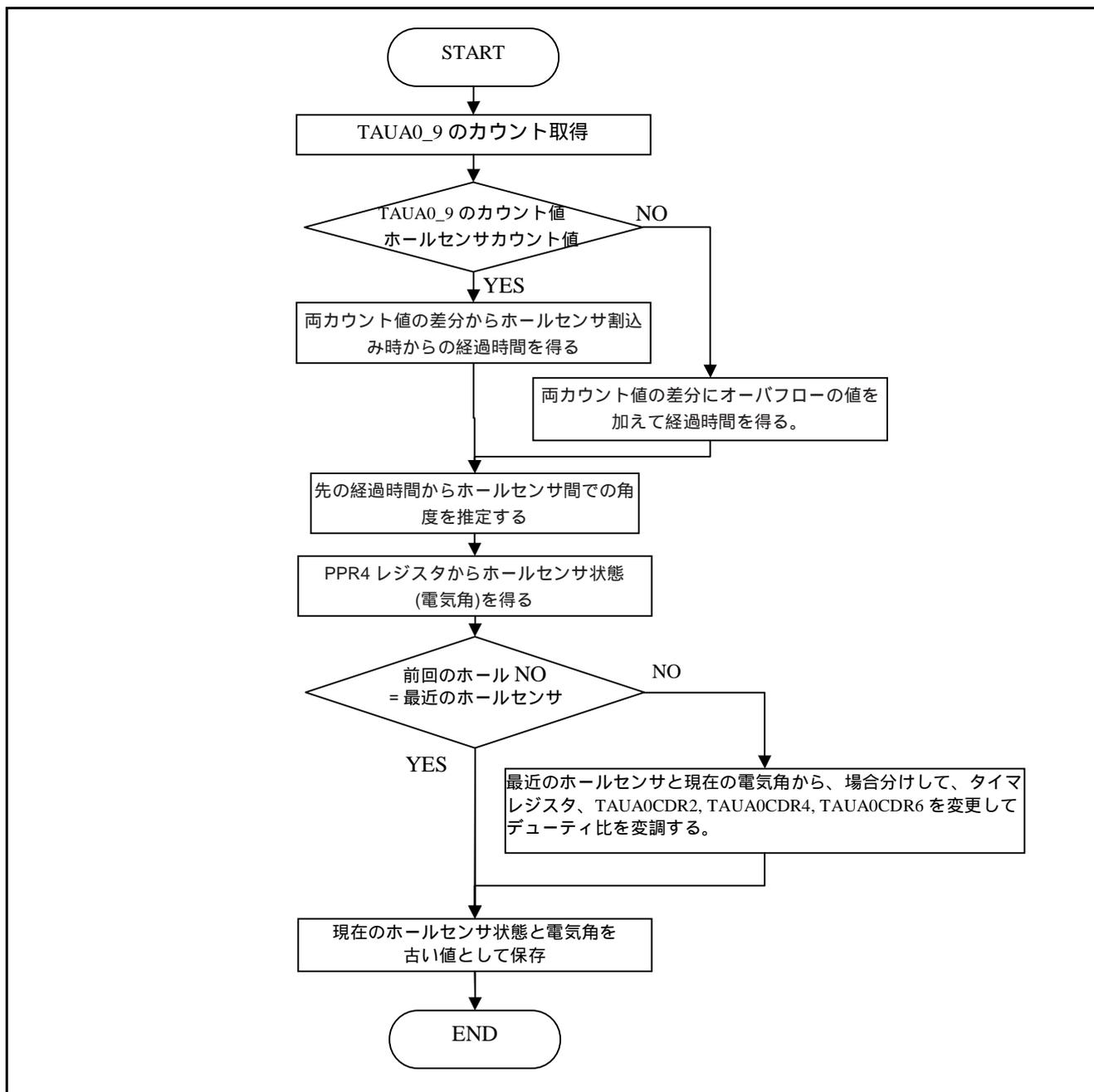


図4.9 キャリア割り込み(50usec)毎のモータ制御

4.9.8 割込み処理：ホールセンサ割り込み(電気角 60°)毎の処理

関数 `hs_int()` にて、角速度調整のため、ホールセンサ割り込みごとの経過時間を測定します。

図 4.10 にホールセンサ割り込み(電気角 60°)毎の処理のフローチャートを示します。

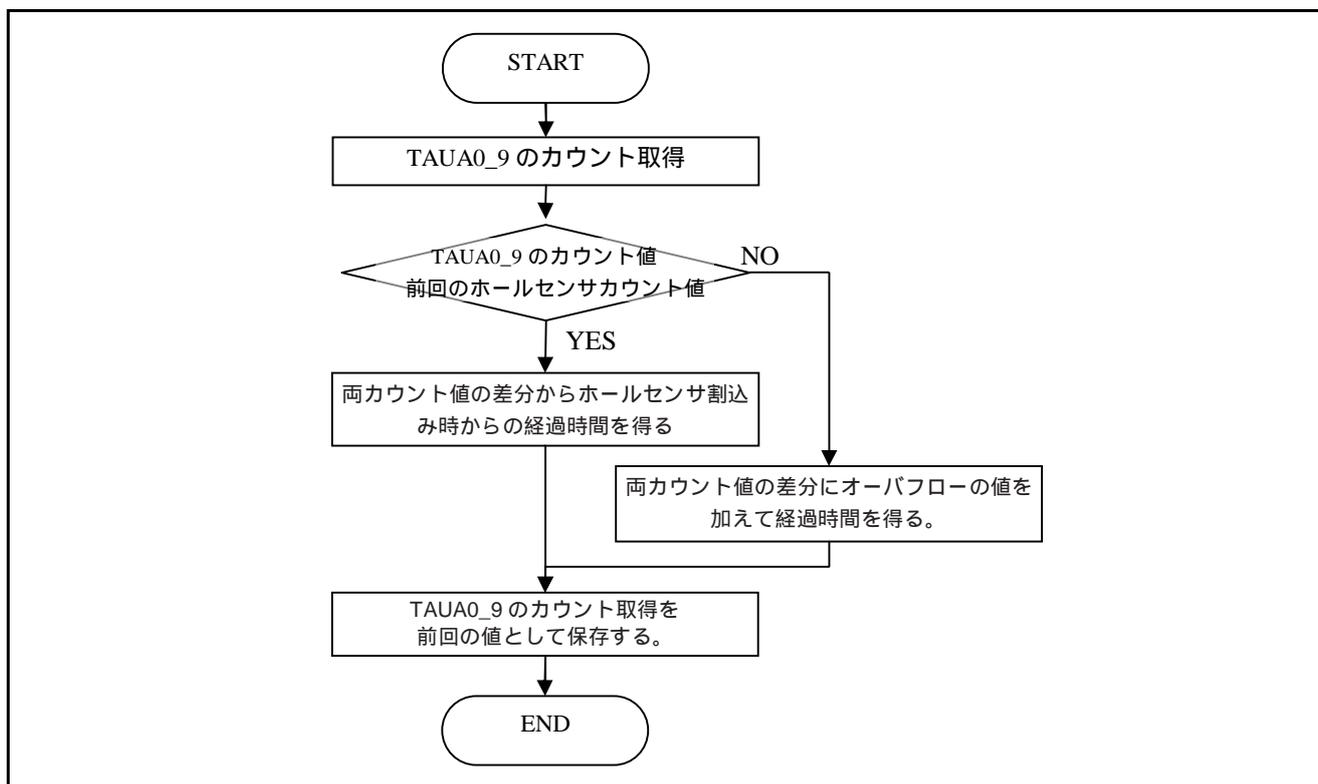


図4.10 ホールセンサ割り込み(電気角 60°)毎の処理

4.9.9 割込み処理：フリーラン・タイマのオーバーフロー(1sec)毎の処理

関数 `ovf_int()` にて、オーバーラン回数のカウンタを 1 増やす。

図 4.11 にフリーラン・タイマのオーバーフロー(1sec)毎の処理のフローチャートを示します。

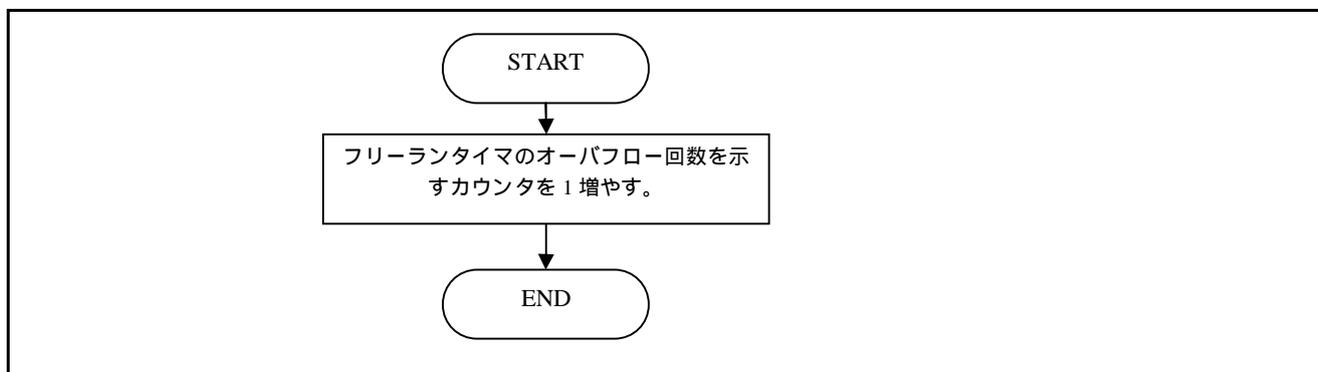


図4.11 フリーラン・タイマのオーバーフロー(1sec)毎の処理

4.9.10 割込み処理：インタバル・タイマのオーバフロー(1msec)毎の処理

関数 pi_int() にて、VR スイッチの値から指令値(300~2000rpm)を求め、シング角を算出します

図 4.12にインタバル・タイマのオーバフロー(1msec)毎の処理のフローチャートを示します。

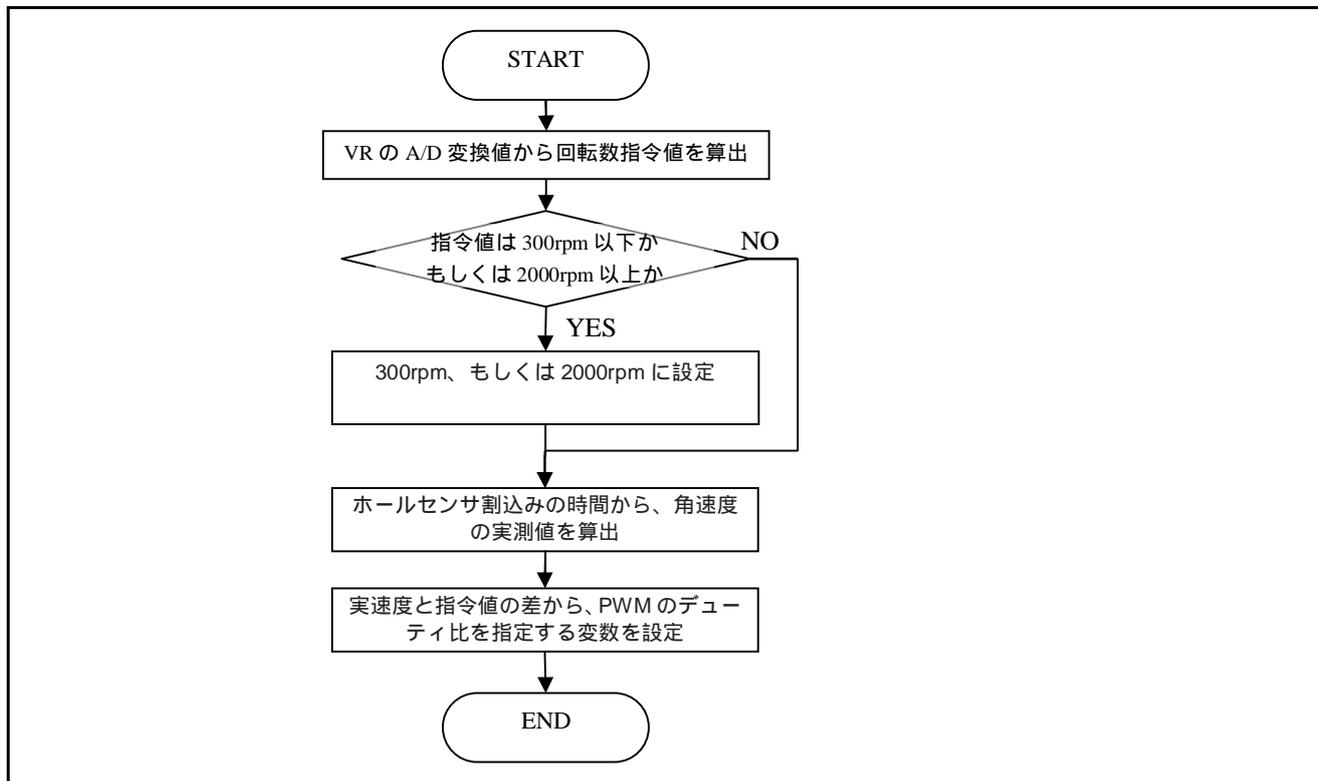


図4.12 インタバル・タイマのオーバフロー(1msec)毎の処理

5. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

6. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

V850E2/ML4 ユーザーズマニュアル ハードウェア編 (R01UH0262JJ)

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com>

お問合せ先

<http://japan.renesas.com/inquiry>

改訂記録	V850E2/ML4 アプリケーションノート 相補 PWM 出力機能編
------	-------------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.06.22	—	初版発行

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。

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。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。

4. クロックについて

【注意】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。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

