

センサレス 120 度通電制御

RAJ306000 実装編

要旨

本アプリケーションノートは、RAJ306000 を使用して、3 相ブラシレス DC モータをセンサレス 120 度通電方式で駆動させるサンプルプログラム、及び、開発支援ツール「In Circuit Scope」のライブラリを使用する方法について説明することを目的とする。

サンプルプログラムは、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。サンプルプログラムを使用する場合、適切な環境で十分な評価を行ったうえで使用して下さい。

動作確認デバイス

サンプルプログラムの動作確認は、下記のデバイスで行っている。

- ・ RAJ306000

対象サンプルプログラム

本アプリケーションノートが対象とするサンプルプログラムを下記に示す。

- ・ RAJ306000_LESS_120_OPEN_CSP_CA_V103 (IDE: CS+ for CA, CX)
- ・ RAJ306000_LESS_120_OPEN_CSP_CC_V103 (IDE: CS+ for CC)
- ・ RAJ306000_LESS_120_OPEN_E2S_CC_V103 (IDE: e² studio)

RAJ306000 向けセンサレス 120 度通電制御サンプルプログラム
(相補 PWM 方式)

参考資料

- ・ RL78/G1F ユーザーズマニュアル ハードウェア編 (R01UH0516JJ0110)
- ・ RAJ306000 シリーズ ユーザーズマニュアル ハードウェア編 (R18UZ0066JJ0100)
- ・ In Circuit Scope 取扱説明書、「In Circuit Scope」を使用するための CS+ の設定方法
(下記のサイトからダウンロードして下さい)

ダウンロードサイト: <http://www.desktoplab.co.jp/download.html>

目次

1. 概説	3
1.1 開発環境	3
2. システム概要	5
2.1 ハードウェア構成	6
2.2 ハードウェア仕様	8
2.2.1 ユーザインタフェース	8
2.2.2 周辺機能	11
2.3 ソフトウェア構成	15
2.3.1 ファイル構成	15
2.3.2 モジュール構成	17
2.4 ソフトウェア仕様	18
3. 制御プログラム説明	19
3.1 制御内容	19
3.1.1 モータ回転開始/停止	19
3.1.2 回転方向指令値、回転速度指令値、VM 電圧	19
3.1.3 回転速度演算	20
3.1.4 PWM による電圧制御	21
3.1.5 逆起電圧検出と電流検出	22
3.1.6 状態遷移	23
3.1.7 始動方法	24
3.1.8 システム保護機能	26
3.1.9 システム保護機能(プリドライバ安全機能)	27
3.2 関数仕様	28
3.3 変数仕様	34
3.4 マクロ定義仕様	38
3.5 フローチャート	49
3.5.1 メイン関数	50
3.5.2 プリドライバ初期化処理	51
3.5.3 センサレス処理: インダクティブセンス処理	52
3.5.4 キャリア周波数割り込み処理	53
3.5.5 キャリア周波数アンダフロー割り込み処理	56
3.5.6 500[us] 割り込み処理	57
3.5.7 ALARM 割り込み処理	58
3.5.8 ALARM 復帰処理	59
4. 開発支援ツール「In Circuit Scope」	60
4.1 概要	60
4.2 ライブラリ使用方法	60
4.3 ICS 用変数一覧	61
改訂記録	64

1. 概説

本アプリケーションノートでは、RAJ306000 を使用した 3 相ブラシレス DC モータのセンサレス 120 度通電制御サンプルプログラムを実装する方法、及び、開発支援ツール「In Circuit Scope」^{※1}のライブラリを使用する方法について説明する。

注:

1. 開発支援ツール「In Circuit Scope」(以降、ICS)は、株式会社デスクトップラボの製品です。
株式会社デスクトップラボ: <http://www.desktoplab.co.jp/>

1.1 開発環境

本アプリケーションノートが対象とするサンプルプログラムの開発環境を Table 1-1, Table 1-2 に示す。

Table 1-1 ソフトウェア開発環境

Integrated Development Environment	CS+ for CA, CX V3.02.00 [15 Mar 2016]
Compiler	CA78K0R V1.72
Integrated Development Environment	CS+ for CC V6.01.00 [01 Dec 2017]
Compiler	CC-RL V1.06.00
Integrated Development Environment	e ² studio Version: 5.4.0.015
Compiler	CC-RL V1.06.00

Table 1-2 ハードウェア開発環境

On-chip Debugging Emulator	RENESAS E1 Emulator (R0E000010KCE00)
Operation Checking Device	RAJ306000 注2
RAJ306000 Series Evaluation Board	RTK0EML2A0D00010BJ

注:

- MCU(RL78/G1F)とプリドライバを内蔵した SIP 製品である RAJ306000 の構成を Figure 1-1 に示す。

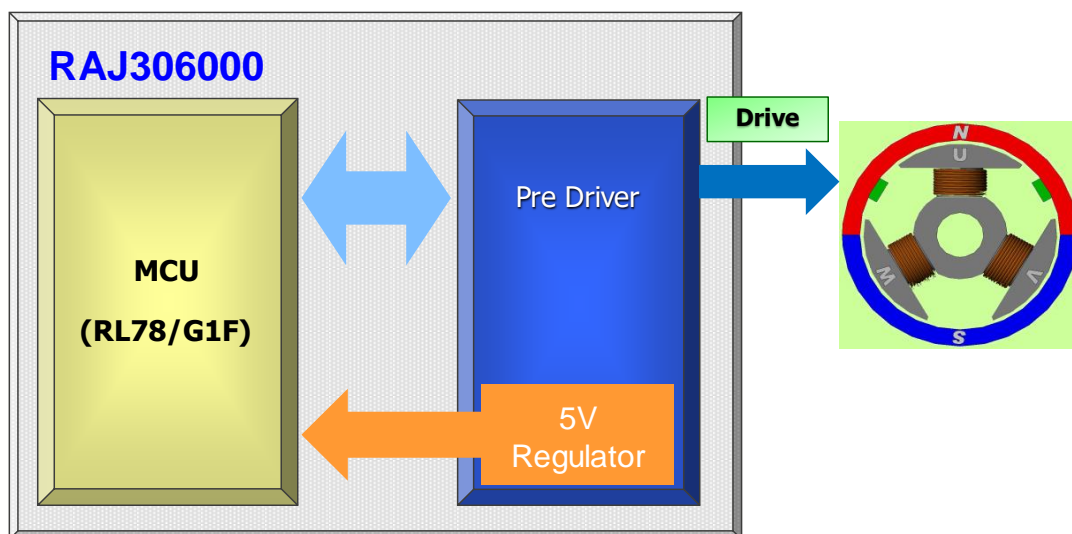


Figure 1-1 RAJ306000

2. システム概要

本システムの概要として、RAJ306000 のシステム構成を Figure 2-1 に示す。

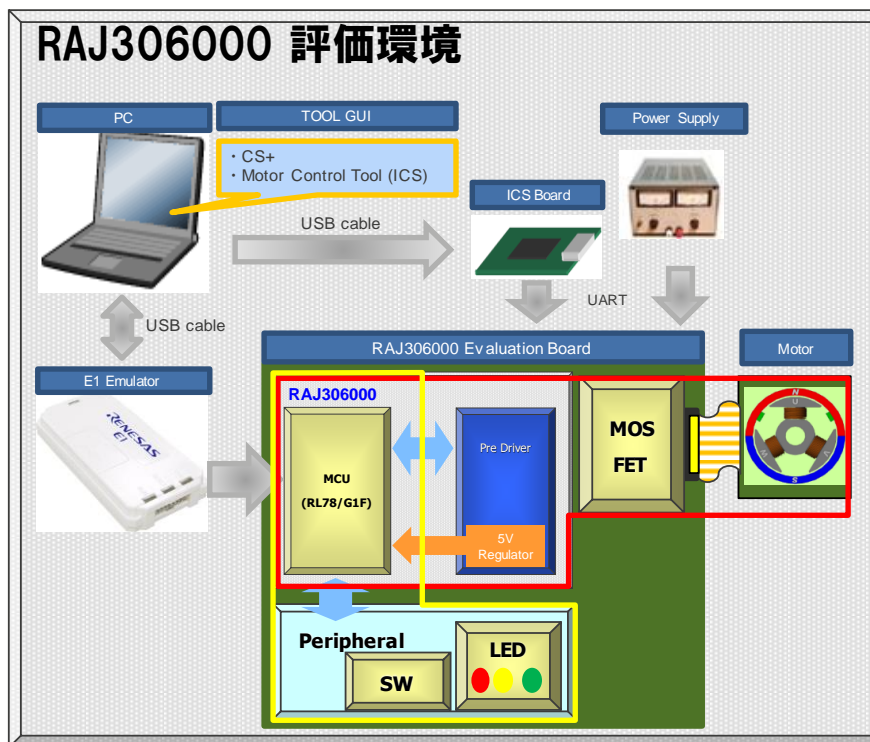


Figure 2-1 システム構成

2.1 ハードウェア構成

ハードウェア構成として、RL78/G1F とプリドライバの接続を Figure 2-2(Figure 2-1 赤線部)に、RL78/G1F と Peripheral との接続を Figure 2-3(Figure 2-1 黄線部)に示す。

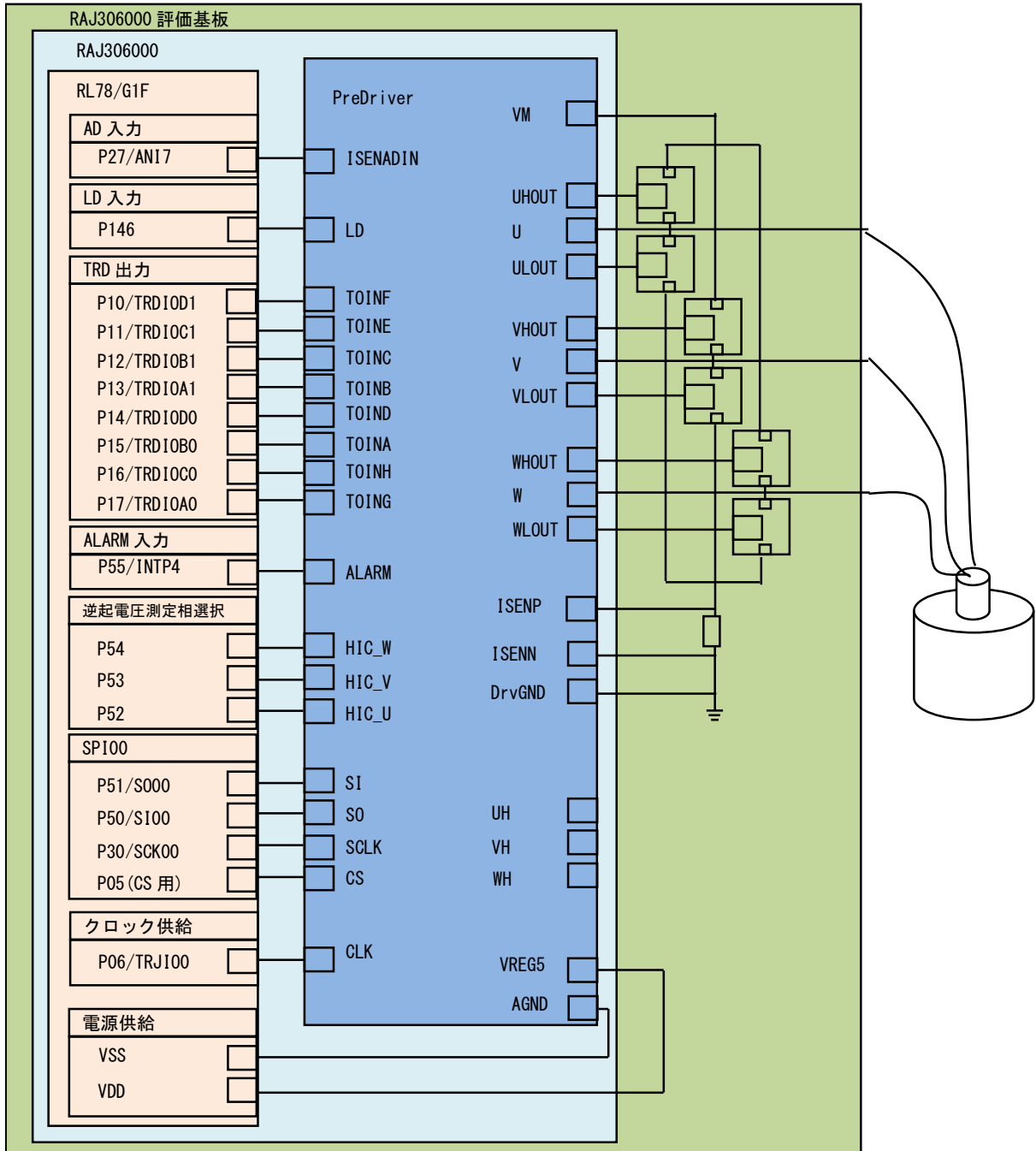


Figure 2-2 ハードウェア構成(RL78/G1F, プリドライバ)

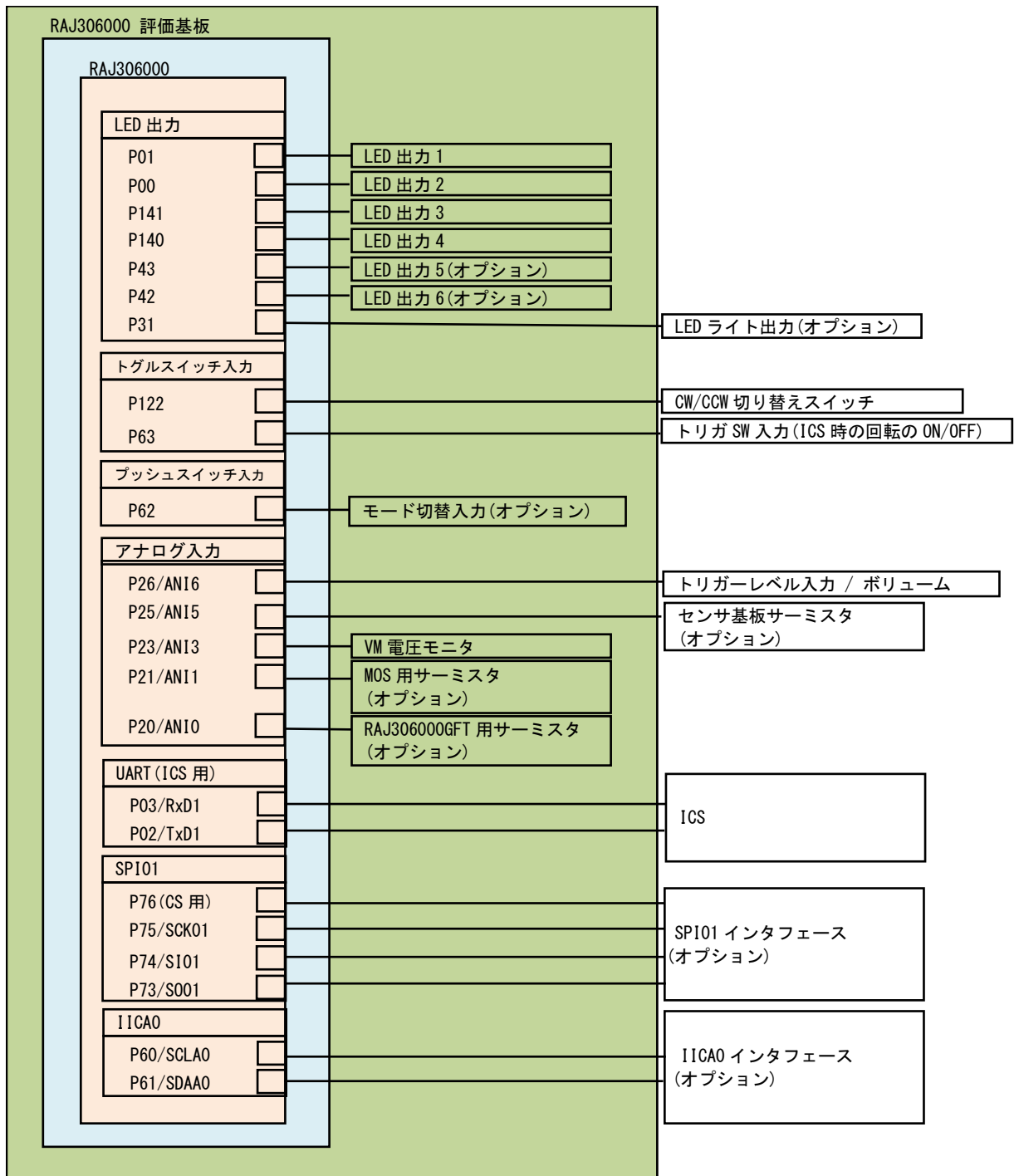


Figure 2-3 ハードウェア構成(RL78/G1F, Peripheral)

2.2 ハードウェア仕様

2.2.1 ユーザインタフェース

本システムのユーザインタフェース一覧を Table 2-1 に示す。

Table 2-1 ユーザインタフェース

項目	インタフェース部品	機能
回転方向	CW/CCW 切り替えスイッチ(SW1) or ICS	回転方向指令値入力(CW/CCW)
回転速度	トリガレベル入力/ボリューム(VR1)	回転速度指令値入力(アナログ値)
START/STOP	トリガレベル入力/ボリューム(VR1) or ICS	モータ回転開始/停止指令
赤色 LED	LED 出力 1	・ 通常動作時: 消灯 ・ エラー検出時: 点灯
	LED 出力 2	・ モータ停止時: 消灯 ・ モータ回転時: 点灯
	LED 出力 3	・ 1500[rpm] 以下: 消灯 ・ 1500[rpm] 超過: 点灯
	LED 出力 4	・ 3000[rpm] 以下: 消灯 ・ 3000[rpm] 超過: 点灯
	LED 出力 5	・ 通常動作時: 消灯 ・ エラー検出時: 点灯
	LED 出力 6	・ 通常動作時: 消灯 ・ エラー検出時: 点灯
過電圧/低電圧検出	VM 電圧モニタ	VM 電圧測定(入力)

本システムの RL78/G1F 端子インタフェース一覧を Table 2-2 に示す。

Table 2-2 端子インタフェース(RL78/G1F)

端子名	機能
P27/ANI7	ブリドライバ電圧測定(入力)
P10/TRDIOD1	ポート出力 or PWM 出力(W_n)
P11/TRDIOC1	ポート出力 or PWM 出力(V_n)
P12/TRDIOB1	ポート出力 or PWM 出力(W_p)
P13/TRDIOA1	ポート出力 or PWM 出力(V_p)
P14/TRDIOD0	ポート出力 or PWM 出力(U_n)
P15/TRDIOB0	ポート出力 or PWM 出力(U_p)
P55/INTP4	ALARM 信号入力
P54	逆起電圧測定相選択 W 相
P53	逆起電圧測定相選択 V 相
P52	逆起電圧測定相選択 U 相
P51/SO00	ブリドライバ制御用データ出力
P50/SI00	ブリドライバ制御用データ入力
P30/SCK00	ブリドライバ制御用クロック出力
P05(CS)	ブリドライバ制御用チップセレクト
P06/TRJIO0	ブリドライバ用システムクロック出力
VSS	グラウンド電位
VDD	正電源
P146, P16/TRDIOC0, P17/TRDIOA0	未使用端子
P01	LED 出力 1 点灯/消灯制御
P00	LED 出力 2 点灯/消灯制御
P141	LED 出力 3 点灯/消灯制御
P140	LED 出力 4 点灯/消灯制御
P43	LED 出力 5 点灯/消灯制御
P42	LED 出力 6 点灯/消灯制御
P122	回転方向指令値入力(CW/CCW)
P26/ANI6	回転速度指令値入力(アナログ値)
	モータ回転開始/停止指令
P23/ANI3	VM 電圧測定(入力)
P03/RxD1	ICS 用 UART 入力
P02/TxD1	ICS 用 UART 出力
P31, P63, P62, P25/ANI5, P21/ANI1, P20/ANI0 P76(CS), P75/SCK01, P74/SI01, P73/SO01 P60/SCLA0, P61/SDLA0	未使用端子

本システムのプリドライバ端子インタフェース一覧を Table 2-3 に示す。

Table 2-3 端子インタフェース(プリドライバ)

端子名	機能
ISENADIN	プリドライバ電圧出力
TOINF	モータ制御信号入力(W_n)
TOINE	モータ制御信号入力(V_n)
TOINC	モータ制御信号入力(W_p)
TOINB	モータ制御信号入力(V_p)
TOIND	モータ制御信号入力(U_n)
TOINA	モータ制御信号入力(U_p)
ALARM	ALARM 信号出力
HIC_W	逆起電圧測定相選択 W 相
HIC_V	逆起電圧測定相選択 V 相
HIC_U	逆起電圧測定相選択 U 相
SI	SPI 用データ入力
SO	SPI 用データ出力
SCLK	SPI 用クロック入力
CS	SPI 用チップセレクト
CLK	システムクロック入力
LD, TOINH, TOING	未使用端子
VM	Power Supply
UHOUT	U 相 High-Side Driver(Nch)駆動用出力
U	U 相位相検出用
ULOUT	U 相 Low-Side Driver(Nch)駆動用出力
VHOUT	V 相 High-Side Driver(Nch)駆動用出力
V	V 相位相検出用
VLOUT	V 相 Low-Side Driver(Nch)駆動用出力
WHOUT	W 相 High-Side Driver(Nch)駆動用出力
W	W 相位相検出用
WLOUT	W 相 Low-Side Driver(Nch)駆動用出力
ISENP	シャント抵抗プラス側接続
ISENN	シャント抵抗マイナス側接続
DrvGND	プリドライバ出力段回路 GND
VREG5	Regulator Output (5V)
AGND	プリドライバアナログ回路 GND
UH, VH, WH	未使用端子

2.2.2 周辺機能

本システムで使用する周辺機能の一覧を Table 2-4 に示す。

Table 2-4 周辺機能

周辺機能	機能
A/D コンバータ	回転速度指令値入力(アナログ値)
	電圧測定(逆起電圧測定/VM 電圧測定)
	電流測定
	オプション: 温度測定
汎用ポート	回転方向指令値入力(CW/CCW)
	逆起電圧測定相選択
	モータ制御信号出力: ポート出力
	LED 出力点灯/消灯制御 オプション: LED ライト出力、トグルスイッチ/プッシュスイッチ入力
タイマ・アレイ・ユニット	500[us] インターバルタイマ
	回転速度計測用フリーランタイマ
タイマ RJ	プリドライバ用システムクロック出力
タイマ RD	モータ制御信号出力: 相補 PWM モードを使用した PWM 出力(6 本)
外部割り込み	ALARM 信号検出
通信インタフェース	SPI00(プリドライバ制御用)
	UART1(ICS 用)
	オプション: SPI01, IICA0

(1) A/D コンバータ

回転速度指令値入力(アナログ値)、電圧測定を「A/D コンバータ」を使用して測定する。

A/D 変換は、ソフトウェアトリガを使用して、チャンネル選択モードを「セレクトモード」に、変換動作モードを「ワンショット変換モード」に設定する。

また、A/D 変換の変換時間は、1 チャンネル辺り 2.375[us]で、変換入力値の最小単位を Table 2-5 に示す。

Table 2-5 A/D コンバータ

項目	A/D コンバータ 1 ビット当たりの制御値	チャンネル
回転速度指令値入力 (アナログ値)	5.56[rpm] ステップ (回転速度範囲は、CW/CCW 共に 1100[rpm] ~ 4290[rpm])	ANI6
電圧測定	VM 電圧測定: 45.9[V] / 1024 = 0.045[V]	ANI3
	逆起電圧 ^{注3} 測定 : 5[V] / 1024 = 0.0049[V]	ANI7
電流測定	電流 ^{注3} 測定 : 200[A] / 1024 = 0.195[A]	ANI7

注:

3. 逆起電圧と電流は、プリドライバ側の ADC セレクタ用レジスタ(ADC_SEL)の設定により A/D 変換する信号を切り換えることで測定することができる。

逆起電圧測定は、Hall 信号処理設定レジスタ(HALL_SIG)の BEMF_MODE_SEL に 1(逆起電圧の測定(AD 測定))を設定して、ADC_SEL に 0x03(BEMF レベル検出)を設定した場合、制御値が反映される。

電流測定は、ADC_SEL に 0x01(電流検出 ISENSE)を設定した場合、制御値が反映される。

詳細は、「RAJ306000 シリーズ ユーザーズマニュアル ハードウェア編 (R18UZ0066JJ0100)」を参照して下さい。

(2) 汎用ポート

汎用ポートで逆起電圧を検出する逆起電圧測定相の選択を行う。逆起電圧を検出する場合、選択した逆起電圧測定相と仮想中点(モータの中心)の電位差を A/D 変換し、仮想中点より高いか低いかをプリドライバが判定する。逆起電圧を検出する逆起電圧測定相の汎用ポートを High にして ANI7 の A/D 変換の結果から逆起電圧を得ることができる。

なお、この逆起電圧測定相選択を有効にするには、プリドライバの Hall 信号処理設定レジスタ (HALL_SIG) の HALL_MODE_SEL に 1(センサレス選択)を設定する。

汎用ポートと逆起電圧測定相選択の組み合わせを Table 2-6 に示す。

Table 2-6 汎用ポートと逆起電圧測定相選択

端子名	逆起電圧測定相選択
P52	U 相
P53	V 相
P54	W 相

また、本システムでは、ポート出力機能を使用して、PWM 出力と共に、モータ制御信号を出力する。

汎用ポートとモータ制御信号出力の組み合わせを Table 2-7 に示す。

Table 2-7 汎用ポートとモータ制御信号出力

端子名	モータ制御信号
P10/TRDIOD1	W_n
P11/TRDIOC1	V_n
P12/TRDIOB1	W_p
P13/TRDIOA1	V_p
P14/TRDIOD0	U_n
P15/TRDIOB0	U_p

注:

汎用ポートを入力モードから出力モードに切り替える場合の注意事項については、「RL78/G1F ユーザーズマニュアル ハードウェア編 (R01UH0516JJ0110)」を参照して下さい。

(3) タイマ・アレイ・ユニット

- ・ 500[us] インターバルタイマ

500[us] インターバルタイマは、タイマ・アレイ・ユニットの「インターバルタイマ機能」を使用する。本システムでは、チャンネル 0 を使用する。

- ・ 回転速度計測用フリーランタイマ

回転速度計測用フリーランタイマは、タイマ・アレイ・ユニットの「インターバルタイマ機能」を使用する。ただし、割り込みは使用しない。本システムでは、チャンネル 1 を使用する。

また、本システムでは、チャンネル 2 とチャンネル 3 を使用しない。

(4) タイマ RJ

パルス出力モードを使用して、4[MHz]の矩形波を出力、プリドライバのシステムクロックとして供給する。

(5) タイマ RD

相補 PWM モードを使用して、三角波変調、短絡防止時間ありの三相波形(6 本)を出力する。

本システムでは、周期が 100[us]で、High アクティブの PWM 出力を実現する。また、ALARM 検出時 (INTP4 端子に Low 入力時)は、プリドライバの出力をハイインピーダンス状態(モータ制御信号出力端子を Low 状態)にする。

タイマ出力端子とモータ制御信号出力の組み合わせを Table 2-8 に示す。

Table 2-8 タイマ出力端子とモータ制御信号出力

端子名	モータ制御信号
P10/TRDIOD1	W_n
P11/TRDIOC1	V_n
P12/TRDIOB1	W_p
P13/TRDIOA1	V_p
P14/TRDIOD0	U_n
P15/TRDIOB0	U_p

(6) 割り込み

本システムで使用する割り込みの一覧を Table 2-9 に示す。

Table 2-9 割り込み

端子名	割り込み
P55/INTP4	ALARM 信号検出
INTTM00	500[us] インターバルタイマ
INTTRD0	キャリア周波数(PWM)
INTTRD1	キャリア周波数(アンダフロー)
INTCSI00	SPI00(プリドライバ制御用)通信完了

2.3 ソフトウェア構成

2.3.1 ファイル構成

サンプルプログラムのファイル構成を Table 2-10, Table 2-11 に示す。

Table 2-10 サンプルプログラムのファイル構成(1)

RAJ306000_LESS_120_OPEN_CSP_CA_V103 RAJ306000_LESS_120_OPEN_CSP_CC_V103 RAJ306000_LESS_120_OPEN_E2S_CC_V103		
Inc	control_parameter.h	制御特性依存部ヘッダ
	motor_parameter.h	モータ特性依存部ヘッダ
	mtr_common.h	共通定義用ヘッダ
	mtr_ctrl_rl78g1f.h	RL78/G1F 依存処理部ヘッダ
	mtr_ctrl_rl78g1f_t2001.h	RL78/G1F & ボード依存処理部ヘッダ
	mtr_ctrl_t2001.h	ボード依存処理部ヘッダ
	mtr_main.h	メイン関数、ユーザインタフェース制御ヘッダ
	mtr_spm_less_120_cpm.h	センサレス 120 度通電制御依存部ヘッダ
	r_dsp.h	演算ライブラリヘッダ
	r_less_120_is.h	インダクティブセンス・ライブラリヘッダ
	r_less_120_isw.h	RL78/G1F 依存処理部ヘッダ
	r_stdint.h	型定義用ヘッダ
	version.h	ソフトウェアリビジョン定義用ヘッダ
ics	ICS2_CA_RL78G1F.lib	ICS 用ライブラリ(CA78K0R 用) ^{注4}
	ICS2_CC_RL78G1F.lib	ICS 用ライブラリ(CC-RL 用) ^{注5}
	ics2_RL78G1F.h	ICS 用ヘッダ
	RL78_vector.c	ICS 用割り込みハンドラ
	RL78_vector.h	ICS 用割り込みハンドラヘッダ
lib	R_dsp_rl78_CA.lib	演算ライブラリ(CA78K0R 用) ^{注4}
	R_dsp_rl78_CC.lib	演算ライブラリ(CC-RL 用) ^{注5}
	r_less_120_is_ca.lib	インダクティブセンス・ライブラリ(CA78K0R 用) ^{注4}
	r_less_120_is_cc.lib	インダクティブセンス・ライブラリ(CC-RL 用) ^{注5}
src	mtr_ctrl_rl78g1f.c	RL78/G1F 依存処理部
	mtr_ctrl_rl78g1f_t2001.c	RL78/G1F & ボード依存処理部
	mtr_ctrl_t2001.c	ボード依存処理部
	mtr_interrupt.c	割り込みハンドラ
	mtr_main.c	メイン関数、ユーザインタフェース制御
	mtr_spm_less_120_cpm.c	センサレス 120 度通電制御依存部
	r_less_120_isw.c	RL78/G1F 依存処理部

注:

4. CA78K0R 用は、RAJ306000_LESS_120_OPEN_CSP_CA_V103 にのみ含まれます。
5. CC-RL 用は、RAJ306000_LESS_120_OPEN_CSP_CC_V103 と RAJ306000_LESS_120_OPEN_E2S_CC_V103 にのみ含まれます。

Table 2-11 サンプルプログラムのファイル構成(2)

RAJ306000_LESS_120_OPEN_CSP_CA_V103		
RAJ306000_LESS_120_OPEN_CSP_CC_V103		
RAJ306000_LESS_120_OPEN_E2S_CC_V103		
cg_src	r_cg_adc.c	RL78/G1F A/D コンバータ処理
	r_cg_adc.h	RL78/G1F A/D コンバータ処理ヘッダ
	r_cg_adc_user.c	RL78/G1F A/D コンバータ処理(ユーザ用)
	r_cg_cgc.c	RL78/G1F クロック出力処理
	r_cg_cgc.h	RL78/G1F クロック出力処理ヘッダ
	r_cg_cgc_user.c	RL78/G1F クロック出力処理(ユーザ用)
	r_cg_intp.c	RL78/G1F 割り込み機能処理
	r_cg_intp.h	RL78/G1F 割り込み機能処理ヘッダ
	r_cg_intp_user.c	RL78/G1F 割り込み機能処理(ユーザ用)
	r_cg_macrodriver.h	RL78/G1F エラー定義用ヘッダ
	r_cg_main.c	RL78/G1F メイン処理
	r_cg_main.h	RL78/G1F メイン処理ヘッダ
	r_cg_port.c	RL78/G1F 端子機能処理
	r_cg_port.h	RL78/G1F 端子機能処理ヘッダ
	r_cg_port_user.c	RL78/G1F 端子機能処理(ユーザ用)
	r_cg_predrv.c	ブリドライバ処理
	r_cg_predrv.h	ブリドライバ処理ヘッダ
	r_cg_predrv_prm.h	ブリドライバレジスタパラメータ定義用ヘッダ
	r_cg_predrv_reg.h	ブリドライバレジスタアドレス定義用ヘッダ
	r_cg_predrv_user.c	ブリドライバ処理(ユーザ用)
	r_cg_sau.c	RL78/G1F シリアル・アレイ・ユニット処理
	r_cg_sau.h	RL78/G1F シリアル・アレイ・ユニット処理ヘッダ
	r_cg_sau_user.c	RL78/G1F シリアル・アレイ・ユニット処理(ユーザ用)
	r_cg_systeminit.c	RL78/G1F 初期化処理
	r_cg_tau.c	RL78/G1F タイマ・アレイ・ユニット処理
	r_cg_tau.h	RL78/G1F タイマ・アレイ・ユニット処理ヘッダ
	r_cg_tau_user.c	RL78/G1F タイマ・アレイ・ユニット処理(ユーザ用)
	r_cg_tmr.c	RL78/G1F タイマ RD 処理
	r_cg_tmr.h	RL78/G1F タイマ RD 処理ヘッダ
	r_cg_tmr_user.c	RL78/G1F タイマ RD 処理(ユーザ用)
	r_cg_tmrj.c	RL78/G1F タイマ RJ 処理
	r_cg_tmrj.h	RL78/G1F タイマ RJ 処理ヘッダ
	r_cg_tmrj_user.c	RL78/G1F タイマ RJ 処理(ユーザ用)
	r_cg_userdefine.h	RL78/G1F ユーザ定義用ヘッダ
	r_cg_wdt.c	RL78/G1F ウォッチドッグ・タイマ処理
	r_cg_wdt.h	RL78/G1F ウォッチドッグ・タイマ処理ヘッダ
	r_cg_wdt_user.c	RL78/G1F ウォッチドッグ・タイマ処理(ユーザ用)

2.3.2 モジュール構成

サンプルプログラムの階層構造を Figure 2-4 に、ファイル構成との対応付けを Table 2-12 に示す。

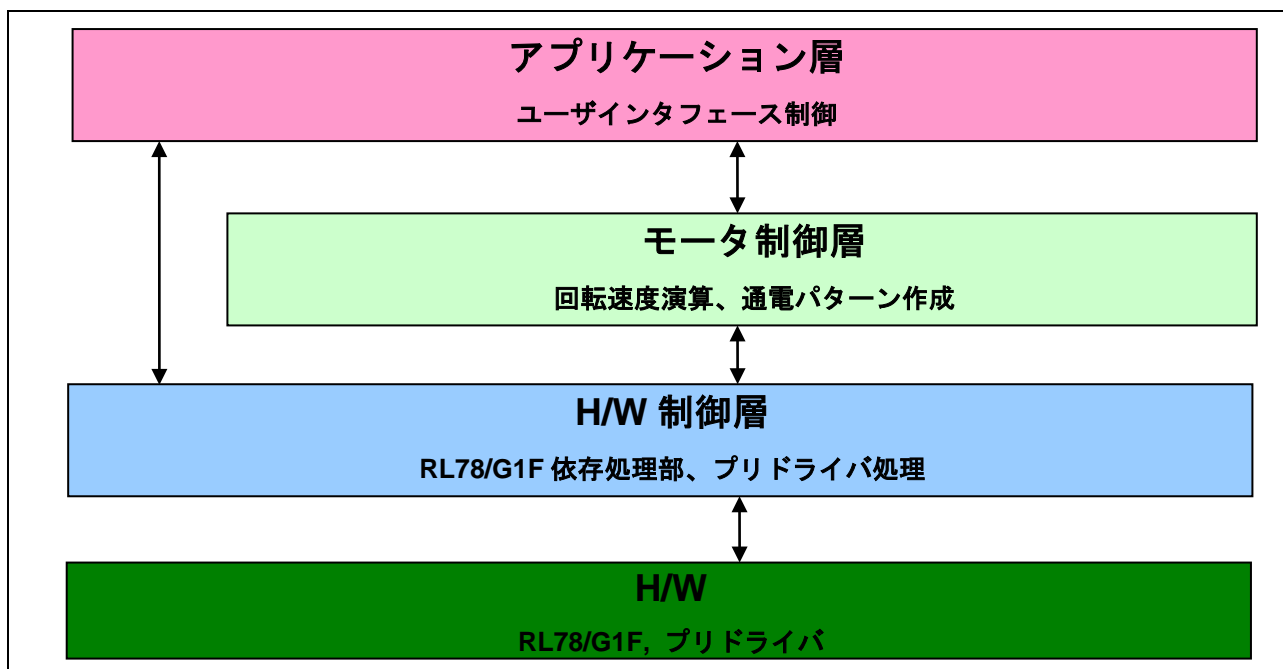


Figure 2-4 サンプルプログラムの階層構造

Table 2-12 サンプルプログラムの階層構造

アプリケーション層	mtr_main.c
モータ制御層	mtr_interrupt.c, mtr_spm_less_120_cpm.c, r_less_120_is*.lib
H/W 制御層	mtr_ctrl_rl78g1f.c, mtr_ctrl_rl78g1f_t2001.c, mtr_ctrl_t2001.c, r_less_120_isw.c, r_cg_adc.c, r_cg_adc_user.c, r_cg_cgc.c, r_cg_cgc_user.c, r_cg_intp.c, r_cg_intp_user.c, r_cg_main.c, r_cg_port.c, r_cg_port_user.c, r_cg_predrv.c, r_cg_predrv_user.c, r_cg_sau.c, r_cg_sau_user.c, r_cg_systeminit.c, r_cg_tau.c, r_cg_tau_user.c, r_cg_tmr.c, r_cg_tmr_user.c, r_cg_tmrj.c, r_cg_tmrj_user.c, r_cg_wdt.c, r_cg_wdt_user.c

2.4 ソフトウェア仕様

本アプリケーションノートが対象とするソフトウェアの基本仕様を Table 2-13, Figure 2-5 に示す。

Table 2-13 ソフトウェアの基本仕様

項目	内容
制御方式	120 度通電方式
モータ回転開始/停止	モータ回転開始/停止指令は、VR1(ANI6 端子)のレベルによって決定する。 ICS からの入力 ^{注6}
回転方向制御	回転方向指令値は、SW1(P122 端子)のレベルによって決定する。 ICS からの入力 ^{注6}
回転速度制御	回転速度指令値は、VR1(ANI6 端子)のレベルによって決定する。 回転速度指令値に比例させた PWM Duty を 0 ~ 100[%]で制御する。
回転速度範囲	CW/CCW 共に 1100[rpm] ~ 4290[rpm]
回転子磁極位置検出	逆起電圧ゼロクロス位置から電気角 30 度の位置毎(電気角 60 度毎)
キャリア周波数(PWM)	10[KHz]
制御周期/回転速度演算	逆起電圧ゼロクロス位置から電気角 30 度の位置毎(電気角 60 度毎)に、 ・ PWM Duty の設定と通電パターンを決定する。 ・ 1 回転(360 度)前のカウンタ値との差分から回転速度を算出する。
保護停止処理	以下のエラー検出時に、モータ制御信号出力端子を Low 状態にする。 ・ ALARM エラー ・ 過電圧エラー ・ 回転速度異常エラー ・ タイムアウトエラー ・ 電流パターンエラー ・ 逆起電圧検出エラー ・ 低電圧エラー

注:

6. 詳細は、「4 開発支援ツール「In Circuit Scope」」を参照して下さい。

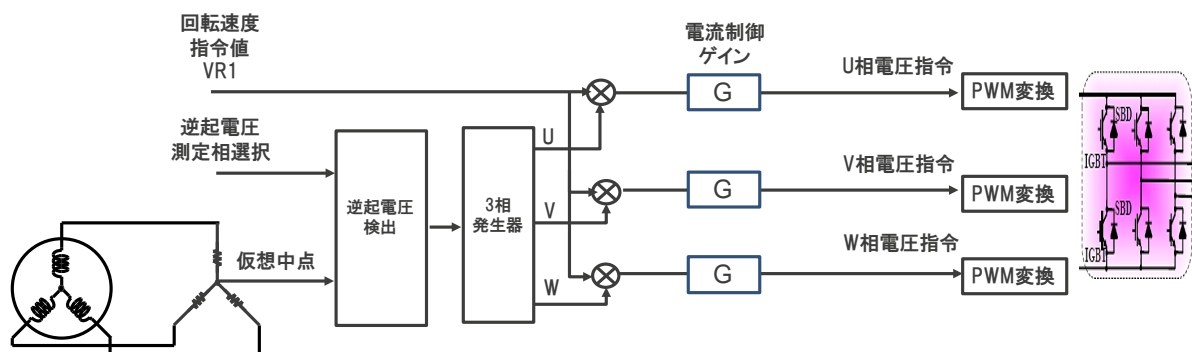


Figure 2-5 ソフトウェアの基本仕様

3. 制御プログラム説明

本アプリケーションノートが対象とするサンプルプログラムについて説明する。

3.1 制御内容

3.1.1 モータ回転開始/停止

モータの回転開始と停止は、VR1 と SW1、または、ICS からの入力によって制御する。

VR1 には、アナログ入力端子(ANI6 端子)が割り当てられ、その入力をメイン・ループ内で A/D 変換して、回転速度指令値を算出する。回転速度指令値が、1200[rpm]以上でモータの回転開始、1100[rpm]以下でモータの停止と判断する。

SW1 には、汎用ポート(P122 端子)が割り当てられ、メイン・ループ内で、P122 端子の High/Low 状態を取得して、回転方向指令値とする。回転方向は、回転方向指令値で判断する。

3.1.2 回転方向指令値、回転速度指令値、VM 電圧

(1) 回転方向指令値

SW1 の High/Low 状態、または、ICS から入力することによってモータの回転方向指令値を設定する。

(2) 回転速度指令値

VR1 の出力値(アナログ値)を A/D 変換することによってモータの回転速度指令値を設定する。A/D 変換された VR1 の値は、下記(Table 3-1)のように、回転速度指令値として使用する。

Table 3-1 回転速度指令値の変換比

項目	変換比 (回転速度指令値: A/D 変換値)	チャネル
回転速度指令値	1100[rpm] ~ 4290[rpm]: 03FFH ~ 0000H	ANI6

(3) VM 電圧

下記(Table 3-2)のように、VM 電圧を測定する。過電圧と低電圧の検出(異常時は、PWM 出力を停止)に使用する。

Table 3-2 VM 電圧の変換比

項目	変換比 (VM 電圧: A/D 変換値)	チャネル
VM 電圧	0.0[V] ~ 45.9[V]: 0000H ~ 03FFH	ANI3

3.1.3 回転速度演算

モータの回転速度は、タイマ・アレイ・ユニットのチャンネル 1 をフリーランニングさせて、逆起電圧ゼロクロス位置から電気角 30 度の位置の時にカウンタ値を取得、1 回転(360 度)前との差分から算出する。また、算出した回転速度に対して、LPF(Low-Pass Filter)処理を行っている。回転速度演算の概念図を Figure 3-1 に示す。

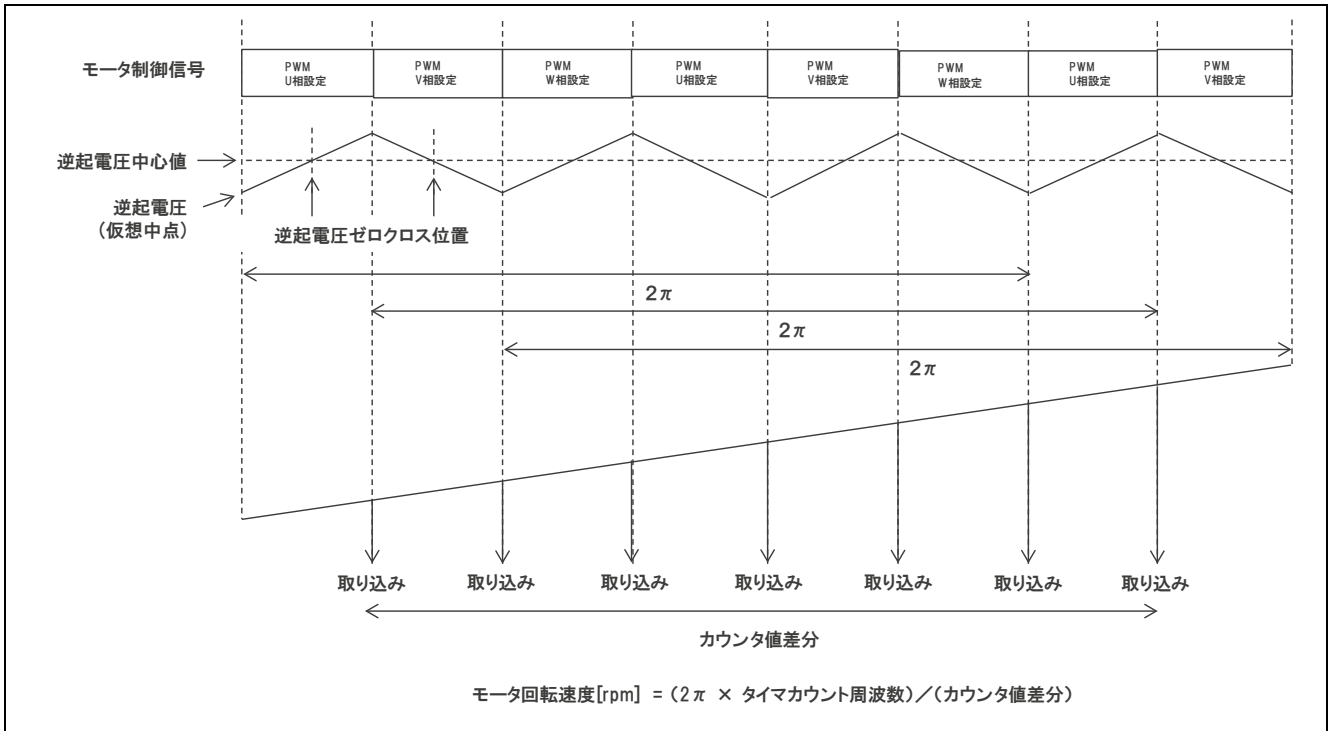


Figure 3-1 回転速度演算

3.1.4 PWM による電圧制御

出力電圧の制御には、PWM 制御を使用する。PWM 制御とは、パルスの Duty 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、回転速度指令値に比例させた PWM Duty を 0 ~ 100[%]で制御する。PWM 制御の概念図を Figure 3-2 に示す。

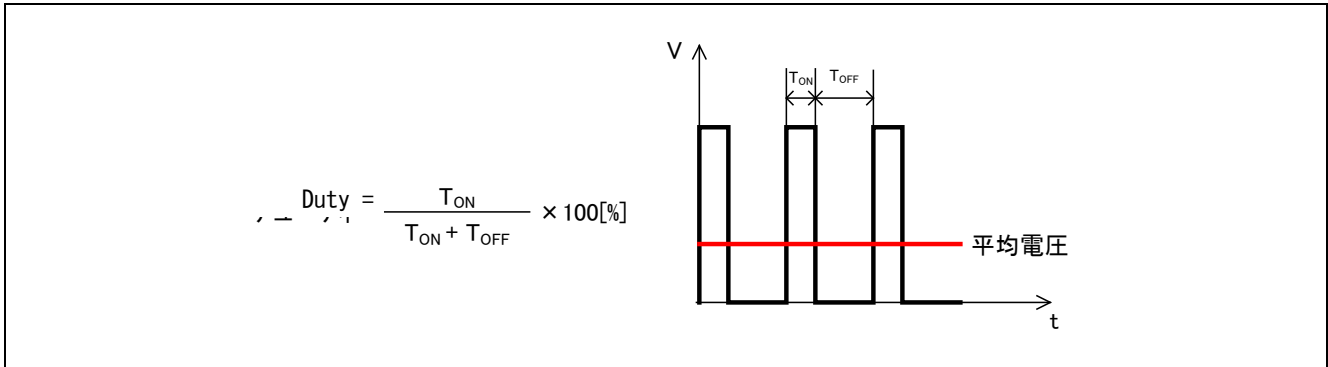


Figure 3-2 PWM 制御

また、本アプリケーションノートが対象とするサンプルプログラムでは、前半 60 度チョッピングを採用して、出力電圧及び回転速度を制御する。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 Figure 3-3 に示す。

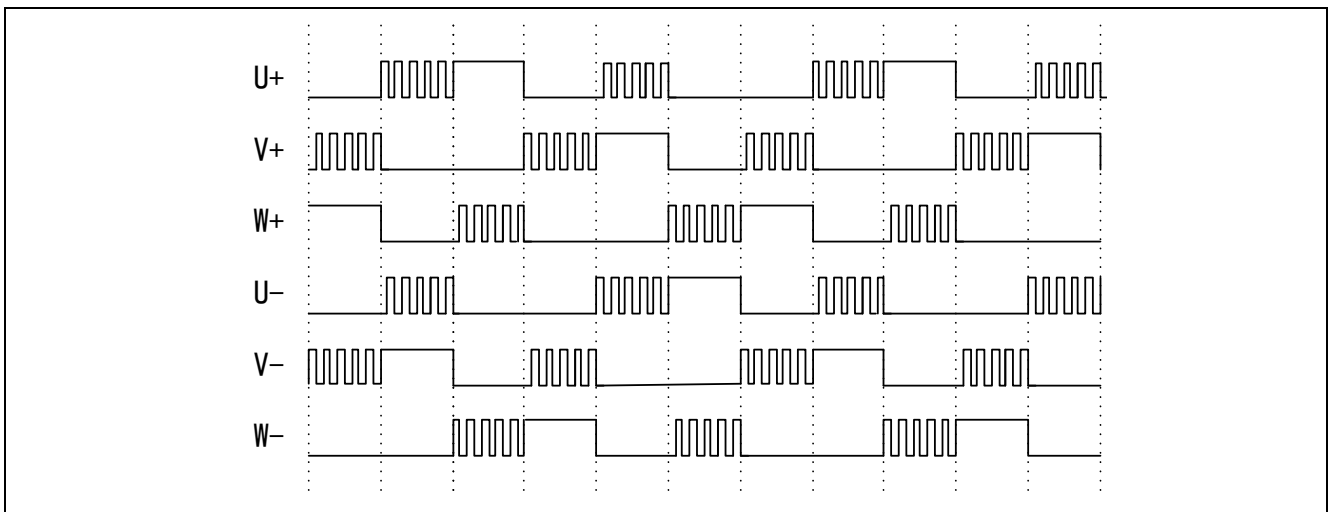


Figure 3-3 相補前半 60 度チョッピング

3.1.5 逆起電圧検出と電流検出

逆起電圧検出は、PWM Duty の状態によって検出タイミングを2種類用意している。PWM Duty が 50[%]未満の場合は、モータの電流が再生している時(回生期間)に逆起電圧検出の A/D 変換を実行する。また、PWM Duty が 50[%]以上の場合は、モータの電流が流れている時(電流期間)に逆起電圧検出の A/D 変換を実行する。逆起電圧検出タイミング図を Figure 3-4, Figure 3-5 に示す。

電流検出は、逆起電圧検出と同じブリドライバ端子を利用している。そのため、ADC_SEL によりブリドライバ端子の回路を切り替える必要がある。ADC_SEL で電流測定に切り替えた後、モータの電流が流れている時(電流期間)に電流検出の A/D 変換を行う。その後、ADC_SEL で逆起電圧測定に切り替え、逆起電圧検出に備える。以後、モータ回転中は、電流検出と逆起電圧検出の ADC_SEL を切り替え続ける。この切り替えは、PWM Duty によらず常に行う。電流検出タイミング図を Figure 3-4, Figure 3-5 に示す。

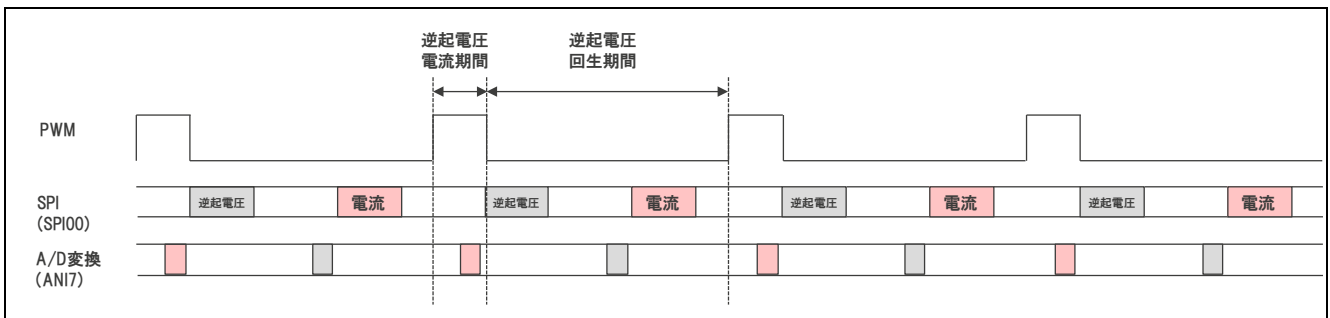


Figure 3-4 逆起電圧検出(回生期間)と電流検出のタイミング (PWM Duty 50[%]未満)

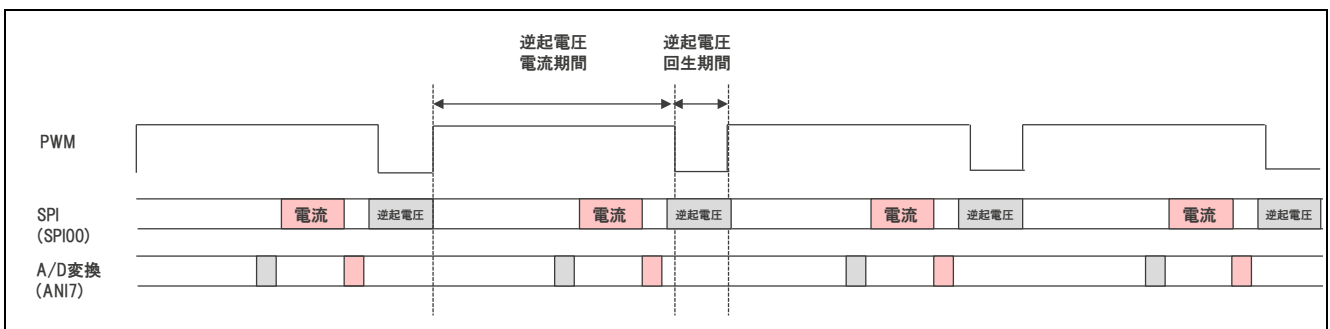


Figure 3-5 逆起電圧検出(電流期間)と電流検出のタイミング (PWM Duty 50[%]以上)

3.1.6 状態遷移

本アプリケーションノートが対象とするサンプルプログラムの状態遷移図を Figure 3-6 に示す。

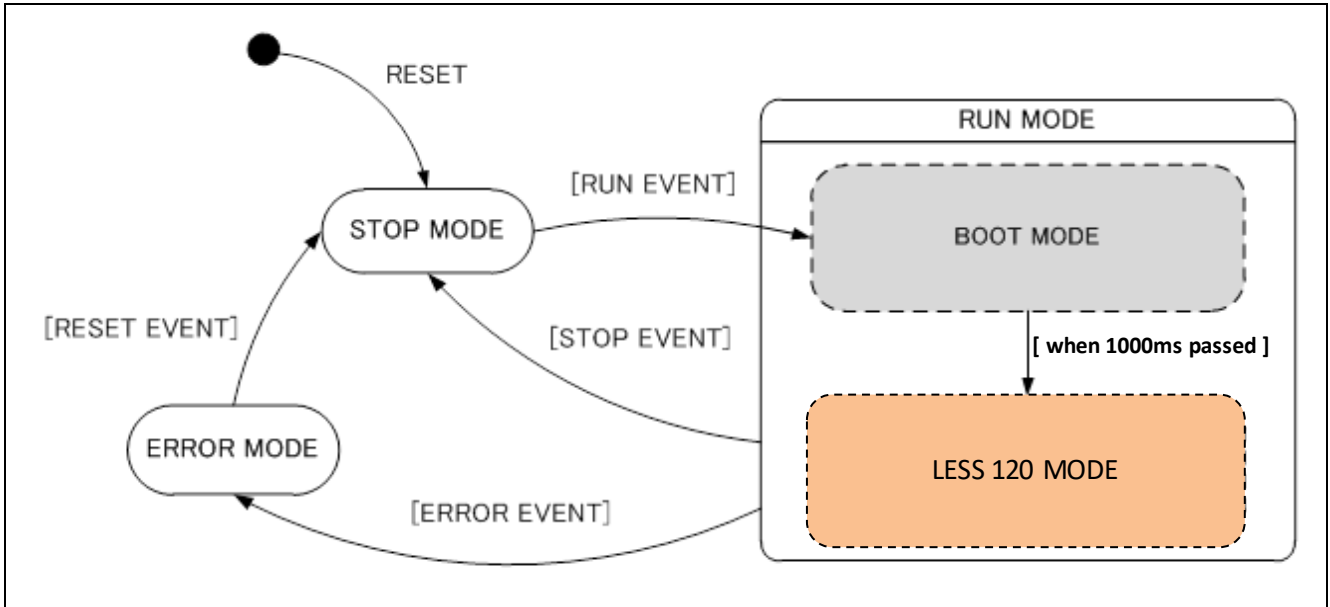


Figure 3-6 状態遷移図

3.1.7 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逆起電圧を利用し、電気角 60 度毎の磁極の位置を検出している。しかし、逆起電圧は回転することで発生するため、始動時は逆起電圧を利用せずに磁極の位置を検出して回転を行うことが必要になる。

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停止中の磁極位置を検出して回転するインダクティブセンスシーケンスを実行する。

次に、逆起電圧シーケンスへ移行し、各相の逆起電圧を検出して Hi-Z 状態の相を判定する Hi-Z 検出を実行して通電パターンを決定する。この通電パターンを起点として逆起電圧検出(モータ駆動)を実行する。

逆起電圧検出(モータ駆動)では逆起電圧と逆起電圧中心値を比較して逆起電圧ゼロクロス位置判定する。そして、逆起電圧ゼロクロス位置から電気角 30 度位置の時間を求めて、この時間経過した後に次の相の通電パターンに切り替える。この一連の動作を常に続けてモータ回転を行う。

逆起電圧シーケンスの Hi-Z 検出で U 相検出してモータ回転する例を Figure 3-7 始動方法例に示す。

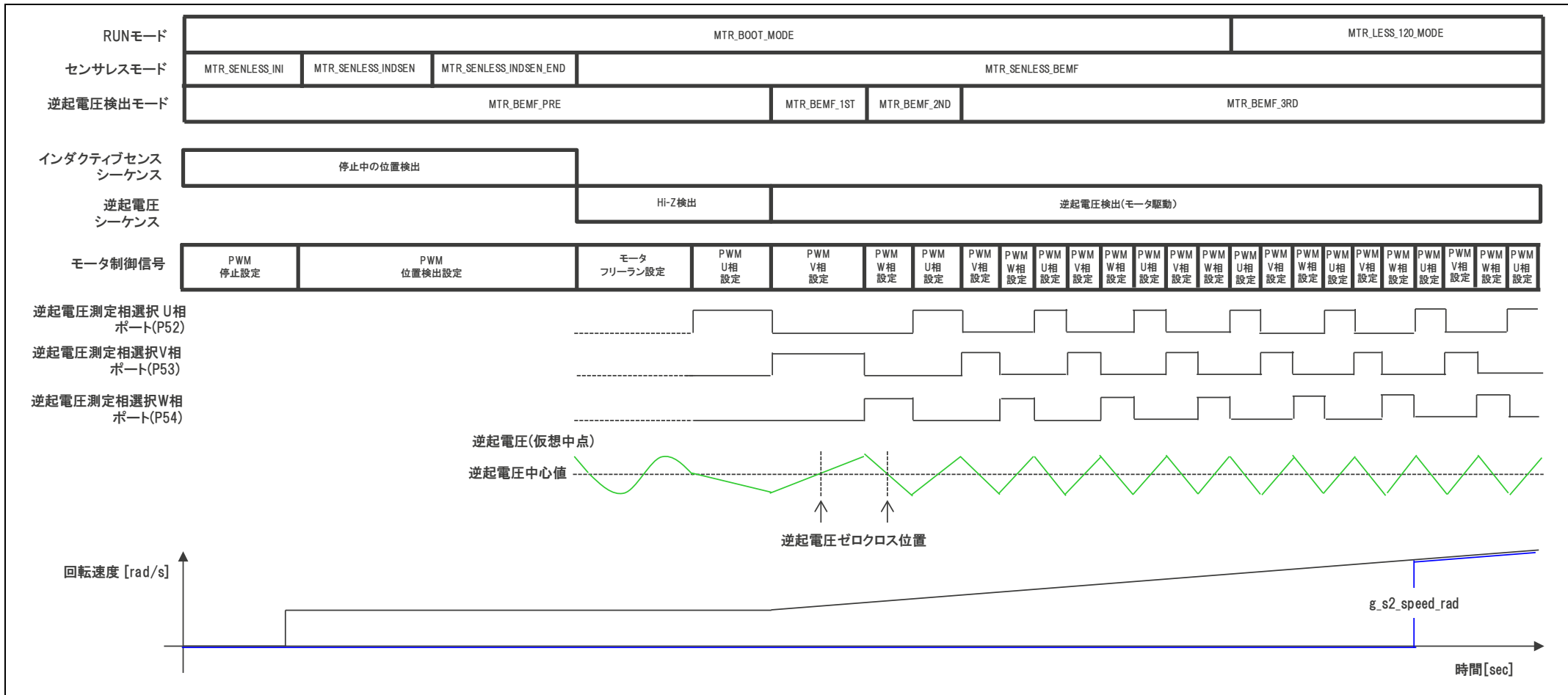


Figure 3-7 始動方法例

3.1.8 システム保護機能

本システムは、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現する。システム保護機能に関わる各設定値を Table 3-3 に示す。

- ・ALARM エラー

プリドライバからの緊急停止信号(ALARM 検出)によって、プリドライバの出力をハイインピーダンス状態(モータ制御信号出力端子を Low 状態)にして、緊急停止する。

- ・過電圧エラー

過電圧監視周期で VM 電圧を監視して、過電圧(過電圧リミット値超過)を検出した場合、緊急停止する。

- ・回転速度異常エラー

回転速度監視周期で回転速度を監視して、回転速度リミット値を超過した場合、緊急停止する。

- ・タイムアウトエラー

タイムアウトエラー監視周期でモータ制御信号の切り替え間隔を監視して、タイムアウト時間を超過してもパターン切り替えが発生していない場合、緊急停止する。

- ・電流パターンエラー

モータ制御信号の切り替え毎に電流パターンを監視して、エラーパターンを検出した場合、緊急停止する。

- ・逆起電圧検出エラー

逆起電圧検出エラー監視周期で逆起電圧シーケンスの処理時間を監視して、タイムアウト時間を超過してもシーケンスが移行しない場合、緊急停止する。

- ・低電圧エラー

低電圧監視周期で VM 電圧を監視して、低電圧(低電圧リミット値未満)を検出した場合、緊急停止する。

Table 3-3 システム保護機能設定値

システム保護機能		設定値
過電圧エラー	過電圧リミット値	30[V]
	監視周期	100[us]
回転速度異常エラー	回転速度リミット値	4290[rpm]
	監視周期	100[us]
タイムアウトエラー	タイムアウト時間	20[ms]
逆起電圧検出エラー	タイムアウト時間	250[ms]
	監視周期	100[us]
低電圧エラー	低電圧リミット値	6[V]
	監視周期	50[us]

3.1.9 システム保護機能(プリドライバ安全機能)

ALARM 動作設定レジスタ(ALMOPE)で、プリドライバ安全機能の有効/無効を設定可能です。

詳細は、データシートを参照して下さい。

3.2 関数仕様

サンプルプログラムの関数一覧を Table 3-4, Table 3-5 に示す。

Table 3-4 関数一覧(1)

ファイル名	関数概要	処理概要
mtr_main.c	main() 入力: なし 出力: なし	<ul style="list-style-type: none"> ・ハードウェア初期化関数呼び出し ・ユーザインタフェース初期化関数呼び出し ・メイン関数使用変数初期化関数呼び出し ・状態遷移、及び、イベント実行関数呼び出し ・メイン関数 ⇒ メイン処理実行関数呼び出し ⇒ センサレス処理の実行 ⇒ ウォッチドッグ・タイマクリア関数呼び出し
	ctrl_ui() 入力: なし 出力: なし	<ul style="list-style-type: none"> ・モータステータスの変更 ・回転速度指令値と回転方向指令値の決定
	ics_ui() 入力: なし 出力: なし	<ul style="list-style-type: none"> ・モータステータスの変更 ・回転速度指令値と回転方向指令値の決定
	ctrl_led() 入力: なし 出力: なし	LED 出力点灯/消灯制御
	ics_predrv_reg_ctrl() 入力: なし 出力: なし	ICS からのブリドライバレジスタの読み込み、書き込み制御
	mcu_sw_init() 入力: なし 出力: なし	F/W の初期化 <ul style="list-style-type: none"> ・インダクティブセンス・ライブラリの初期化 ・F/W 変数の初期化 ・ICS の初期化 ・シーケンス処理の初期化 ・リセットイベントの実行
	software_init() 入力: なし 出力: なし	メイン関数で使用する変数の初期化
	ctrl_sensorless() 入力: なし 出力: なし	センサレス処理 <ul style="list-style-type: none"> ・インダクティブセンス処理 ・逆起電圧シーケンス起動 ・モータ駆動制御設定
mtr_ctrl_rl78g1f.c	clear_wdt() 入力: なし 出力: なし	ウォッチドッグ・タイマクリア
	mtr_clear_oc_flag() 入力: なし 出力: なし	パルス出力強制遮断フラグクリア
	mtr_clear_trd0_imfa() 入力: なし 出力: なし	TRD0 コンペアマッチフラグクリア
	mtr_clear_trd1_udf() 入力: なし 出力: なし	TRD1 アンダフローフラグクリア

mtr_ctrl_rl78g1f_t2001.c	mtr_ctrl_start() 入力: なし 出力: なし	モータ起動処理
	mtr_ctrl_stop() 入力: なし 出力: なし	モータ停止処理
	mtr_change_pattern() 入力: 通電パターン 出力: なし	モータ制御信号出力変更 ・通電パターンの設定 ・逆起電圧測定相選択の設定 ・PWM Duty による通電パターン種別の設定 ・通電パターンエラー時にモータステータス変更 ・イベント処理選択関数呼び出し
	mtr_get_adc() 入力: A/D チャンネル 出力: A/D 変換結果	A/D 変換実行処理
mtr_ctrl_t2001.c	get_vr1() 入力: なし 出力: A/D 変換結果(VR1)	トリガレベルの A/D 変換値取得
	led_on() 入力: LED チャンネル番号 出力: なし	LED の点灯
	led_off() 入力: LED チャンネル番号 出力: なし	LED の消灯
mtr_interrupt.c	mtr_alarm_interrupt() 入力: なし 出力: なし	ALARM 割り込み処理 ・モータステータス変更 ・イベント処理選択関数呼び出し ・パルス出力強制遮断フラグクリア関数呼び出し
	mtr_tau0_interrupt() 入力: なし 出力: なし	500[us] 割り込み処理 ・運転モード切り替え判定処理
	mtr_carrier_interrupt() 入力: なし 出力: なし	キャリア周波数割り込み処理 ・逆起電圧検出/電流検出の A/D 変換 ・逆起電圧検出と電流検出切り替えの SPI 通信 ・逆起電圧シーケンス ・通電パターンの設定 ・モータ回転停止待ち ・エラーチェック関数呼び出し
	mtr_carrier_udf_interrupt() 入力: なし 出力: なし	キャリア周波数アンダフロー割り込み処理 ・逆起電圧検出/電流検出の A/D 変換 ・逆起電圧検出と電流検出切り替えの SPI 通信

mtr_spm_less_120_cpm.c	R_MTR_InitSequence() 入力: なし 出力: なし	シーケンス処理の初期化
	R_MTR_ExecEvent() 入力: 発生イベント 出力: なし	・ステータスの変更を行う ・発生イベントに対して、適切な処理の実行関数を呼び出し
	mtr_act_run() 入力: モータステータス 出力: モータステータス	・モータ起動時変数初期化関数呼び出し ・モータ制御開始関数呼び出し
	mtr_act_stop() 入力: モータステータス 出力: モータステータス	モータ制御終了関数呼び出し
	mtr_act_none() 入力: モータステータス 出力: モータステータス	処理はなし
	mtr_act_reset() 入力: モータステータス 出力: モータステータス	エラー状態から復帰するために必要なグローバル変数の初期化
	mtr_act_error() 入力: モータステータス 出力: モータステータス	エラー発生時のモータ制御終了関数呼び出し
	mtr_pattern_set() 入力: 電流パターン 出力: なし	通電パターン設定 ・回転速度計測関数呼び出し ・通電パターンの決定 ・モータ制御信号出力変更関数呼び出し
	mtr_convert_indsenpat_to_pwmptat() 入力: 電流パターン 出力: 電流パターン(変換値)	電流パターン変換処理
	mtr_get_bemf_threshold() 入力: なし 出力: 逆起電圧中心値	逆起電圧中心値取得処理
	mtr_get_bemf_threshold_direct() 入力: 逆起電圧検出種別、Hi-Z 種別 出力: 逆起電圧中心値	逆起電圧中心値取得処理
	mtr_get_bemf_dir() 入力: なし 出力: 逆起電圧検出方向	逆起電圧検出方向取得処理
	mtr_set_predrv_isense() 入力: なし 出力: なし	電流検出設定処理
	mtr_set_predrv_bemf() 入力: なし 出力: なし	逆起電圧検出設定処理
	mtr_speed_calc() 入力: なし 出力: なし	回転速度計測演算処理
	mtr_start_init() 入力: なし 出力: なし	モータ起動時に必要な変数の初期化
mtr_set_variables() 入力: なし 出力: なし	ICS から入力された値をプロテクト変数に設定	

	R_MTR_IcsInput() 入力: ICS 関連変数の構造体 出力: なし	ICS から入力された値の取得
	R_MTR_SetSpeed() 入力: 回転速度指令値 出力: なし	回転速度の設定
	R_MTR_SetDir() 入力: 回転速度指令値 出力: なし	回転方向の設定
	R_MTR_GetSpeed() 入力: なし 出力: 回転速度情報	回転速度の取得
	R_MTR_GetDir() 入力: なし 出力: 回転方向情報	回転方向の取得
	R_MTR_GetStatus() 入力: なし 出力: モータステータス	モータステータスを取得
	mtr_error_check() 入力: なし 出力: なし	エラーの監視と検出
r_less_120_isw.c	r_isw_set_port() 入力: 汎用ポート設定 出力: 処理結果	汎用ポート(U _p , U _n , V _p , V _n , W _p , W _n)を設定する。
	r_isw_wait() 入力: 指定時間 出力: 処理結果	指定時間[us]待機する。
	r_isw_clear_wdt() 入力: なし 出力: 処理結果	ウォッチドッグ・タイマをクリアする。
	r_isw_get_adc() 入力: なし 出力: 処理結果、A/D 変換結果	A/D 変換結果(電流)を取得する。
r_less_120_is_*.lib	R_IS_Init() 入力: なし 出力: 処理結果	ライブラリを初期化する。
	R_IS_GetRevision() 入力: なし 出力: 処理結果、ライブラリ・リビジョン	ライブラリ・リビジョンを取得する。
	R_IS_Main() 入力: 初期値 出力: 処理結果、停止位置	インダクティブセンス処理 インダクティブセンスを行い、停止位置を検出する。

Table 3-5 関数一覧(2)

ファイル名	関数概要	処理概要
r_cg_adc.c	R_ADC_Create() 入力: なし 出力: なし	A/D 変換器(ADC)初期化
r_cg_adc_user.c	r_adc_interrupt() 入力: なし 出力: なし	SPI 通信 ADC モード SPI 開始判定
r_cg_cgic.c	R_CGIC_Create() 入力: なし 出力: なし	クロック周波数(CGIC)初期化
r_cg_intp.c	R_INTP_Create() 入力: なし 出力: なし	外部割り込み(INTP)初期化
r_cg_main.c	R_MAIN_UserInit() 入力: なし 出力: なし	プリドライバ起動処理
r_cg_port.c	R_PORT_Create() 入力: なし 出力: なし	入出力ポート初期化
r_cg_predrv.c	predriver_hw_init() 入力: なし 出力: なし	プリドライバ初期設定
	R_PREDRV_TRIM_Create() 入力: なし 出力: SPI ステータス	プリドライバトリミングデータ設定
	R_PREDRV_InitSequence() 入力: なし 出力: なし	プリドライバ初期化処理
	R_PREDRV_ErrorRecoverySequence() 入力: ALARM ステータス 出力: なし	ALARM 復帰処理
r_cg_predrv_user.c	R_PreDrvReg_Read() 入力: read address 出力: SPI ステータス, read data	プリドライバレジスタ Read 処理
	R_PreDrvReg_Write() 入力: write address, write data 出力: SPI ステータス	プリドライバレジスタ Write 処理
	R_PreDrvReg_Write_Intr_Mode() 入力: write address, write data 出力: SPI ステータス	プリドライバレジスタ Write 処理 INTR モード
	R_PreDrv_Set_AdcSel() 入力: ADC_SEL 選択指示 出力: なし	ADC_SEL 設定処理
	R_PreDrv_Set_MotEn() 入力: モータ駆動制御指示 出力: なし	モータ駆動制御設定処理

r_cg_sau.c	R_SAU0_Create() 入力: なし 出力: なし	シリアル・アレイ・ユニット(SAU)初期化
	R_UART1_Create() 入力: なし 出力: なし	UART1 初期化
	R_CSI00_Create() 入力: なし 出力: なし	SPI 通信初期化(プリドライバ通信用)
	R_CSI00_Start() 入力: なし 出力: なし	SPI 通信起動(プリドライバ通信用)
	R_CSI00_Send_Receive_SPI_mode() 入力: 送信バッファ バッファサイズ 受信バッファ SPI モード 出力: SPI ステータス	SPI 通信処理
r_cg_sau_user.c	r_csi00_interrupt() 入力: なし 出力: なし	SPI 割り込み処理(プリドライバ通信用)
r_cg_systeminit.c	hwinit() 入力: なし 出力: なし	H/W 初期設定
r_cg_tau.c	R_TAU0_Create() 入力: なし 出力: なし	タイマ・アレイ・ユニット(TAU)初期化
r_cg_tmr.c	R_TMRD0_Create() 入力: なし 出力: なし	タイマ RD(TRD)初期化
	R_TMRD0_Start() 入力: なし 出力: なし	PWM 出力開始
r_cg_tmrj.c	R_TMRJ0_Create() 入力: なし 出力: なし	タイマ RJ(TRJ)初期化
	R_TMRJ0_Start() 入力: なし 出力: なし	プリドライバクロック供給開始
r_cg_wdt.c	R_WDT_Create() 入力: なし 出力: なし	ウォッチドッグ・タイマ初期化

3.3 変数仕様

サンプルプログラムの変数一覧を Table 3-6 に示す。

Table 3-6 変数一覧

変数名	型	内容	備考
g_s2_max_speed	int16_t	回転速度指令最大値	機械角 [rpm]
g_s2_min_speed	int16_t	回転速度指令最小値	機械角 [rpm]
g_s2_margin_min_speed	int16_t	モータ停止用回転速度指令最小値	機械角 [rpm]
g_s2_ref_speed	int16_t	ユーザ設定回転速度	機械角 [rpm]
g_u2_speed_rpm	uint16_t	回転速度演算値	機械角 [rpm]
g_u1_rot_dir	uint8_t	ユーザ設定回転方向	0: CW 1: CCW
g_u1_motor_status	uint8_t	ユーザモータステータス管理	0: 停止 1: 回転中 2: エラー
g_u1_stop_req	uint8_t	モータ停止指令フラグ	回転速度指令値 1100[rpm]以下は停止判定
g_u1_pdrv_status	uint8_t	ブリドライバレジスタ R/W エラーステータス	-
g_u1_err_recovery_req	uint8_t	ALARM 復帰処理要求フラグ	0: 禁止 1: 許可
g_u1_get_alarm_sts1	uint8_t	ブリドライバレジスタ ALMSTS1 取得値	-
g_u1_store_alarm_sts1	uint8_t	ブリドライバレジスタ ALMSTS1 保存値	-
g_u1_get_alarm_sts2	uint8_t	ブリドライバレジスタ ALMSTS2 取得値	-
g_u1_store_alarm_sts2	uint8_t	ブリドライバレジスタ ALMSTS2 保存値	-
g_u2_fw_revision	uint16_t	F/W リビジョン情報	F/W バージョン情報(103)
g_u2_is_lib_revision	uint16_t	インダクティブセンス・ライブラリ・リビジョン情報	インダクティブセンス・ライブラリ・バージョン情報(104)
g_s2_sw_userif	int16_t	ボード UI 使用切り替えフラグ	0: 使用しない 1: 使用する
g_s2_mode_system	int16_t	システムモードフラグ	0: 停止 1: モータ駆動 2: エラー 3: リセット
g_s2_enable_write	int16_t	ICS 書き込み許可フラグ	トルク動作
ics_input	MTR_ICS_INPUT	ICS 入力用構造体	-
g_u2_cnt_boot_mode	uint16_t	BOOT モード時間計測カウンタ	モータ起動後 500[us]をカウント
g_u2_cnt_wait_stop	uint16_t	モータ回転停止待ちカウンタ	モータ停止処理後 10[ms]をカウント
g_u1_flg_wait_stop	uint8_t	モータ回転停止待ちフラグ	モータ停止指令を受けてセットして、モータ停止処理後モータ

			制御信号の切り替えが10[ms]の間未検出の場合クリア
g_u1_enable_write	uint8_t	ICS 入力用構造体書き込み許可フラグ	0: 禁止 1: 許可
g_s2_vdc_ad	int16_t	VM 電圧 A/D 値	[V]
g_s2_reci_vdc_ad	int16_t	VM 電圧 A/D 値の逆数	-
g_s2_pdrv_ad	int16_t	ブリドライバ電圧 A/D 値	[V]
g_s2_pwm_duty	int16_t	タイマ RD コンペアレジスタ設定値	PWM Duty 設定値: 0 ~ 3198 PWM Duty [%]: 0 ~ 100
g_s2_ref_speed_rad	int16_t	回転速度指令値	電気角(Scale: Q2) [rad/s]
g_s2_speed_rad	int16_t	回転速度演算値	電気角(Scale: Q2) [rad/s]
g_s2_speed_lpf_k	int16_t	回転速度 LPF パラメータ	-
g_u1_cnt_ics	uint8_t	ICS 関数呼び出し間隔カウンタ	-
g_u2_run_mode	uint16_t	運転モード管理	0: BOOT モード 3: 通常運転(LESS 120)モード
g_u1_error_status	uint8_t	エラーステータス管理	0x01: ALARM エラー 0x02: 過電圧エラー 0x04: 回転速度異常エラー 0x08: タイムアウトエラー 0x10: 電流パターンエラー 0x20: 逆起電圧検出エラー 0x40: 低電圧エラー (0x80: 未定義エラー)
g_u1_mode_system	uint8_t	ステート管理	0: ストップモード 1: ランモード 2: エラーモード
g_u1_v_pattern	uint8_t	通電パターン	-
g_u2_cnt_timeout	uint16_t	停止判定時間計測カウンタ	通電パターン切り替え時にクリア
g_u1_direction	uint8_t	回転方向管理	0: CW 1: CCW
g_u2_less_timer_cnt	uint16_t	フリーランタイムカウンタ値	TCR01
g_u2_pre_less_timer_cnt	uint16_t	前回のフリーランタイムカウンタ値	-
g_s2_timer_cnt_ave	int16_t	2 π 分の回転速度計測タイムカウンタ差	-
g_u2_timer_cnt_buf	uint16_t	回転速度計測タイムカウンタバッファ	-
g_u2_timer_cnt_num	uint16_t	回転速度計測タイムカウンタバッファ番号	-
ics_input_buff	MTR_ICS_INPUT	ICS 入力変数構造体	-
g_s2_ref_speed_rpm_vr1	int16_t	回転速度指令値	機械角 [rpm]
g_u1_alarm_sts1	uint8_t	ブリドライバレジスタ ALMSTS1 保存値	ICS 表示用
g_u1_alarm_sts2	uint8_t	ブリドライバレジスタ ALMSTS2 保存値	ICS 表示用
g_u1_PreDriver_error	uint8_t	ブリドライバシーケンスエラーステータス	・ブリドライバ初期化シーケンス ・ALARM 復帰シーケンス
g_spi00_comend_flag	uint8_t	SPI 通信状態 フラグ	TURE: 通信終了 FALSE: 通信中

g_spi00_adcend_flag	uint8_t	SPI 通信 ADC End フラグ	TURE : ADC 終了 FALSE: ADC 中
g_spi00_commode	uint8_t	SPI 通信モード	-
gp_csi00_rx_address	uint8_t	SPI 通信受信データアドレス	ブリドライバレジスタ値取得
g_csi00_rx_length	uint16_t	SPI 通信受信データ長	-
g_csi00_rx_count	uint16_t	SPI 通信受信カウンタ	-
gp_csi00_tx_address	uint8_t	SPI 通信送信データアドレス	ブリドライバレジスタアドレス指定
g_csi00_send_length	uint16_t	SPI 通信送信データ長	-
g_csi00_tx_count	uint16_t	SPI 通信送信カウンタ	-
g_u1_sensorless_mode	uint8_t	センサレスモード管理	-
g_u1_bemf_mode	uint8_t	逆起電圧検出モード管理	-
g_u1_bemf_seq	uint8_t	逆起電圧シーケンス番号管理	-
g_u1_indsen_signal	uint8_t	インダクティブセンスパターン位置	電流パターン
g_s2_bemf_ad_pre	int16_t	前回の逆起電圧検出 A/D 値	-
g_s2_bemf_ad_now	int16_t	今回の逆起電圧検出 A/D 値	-
g_s2_bemf_time_cnt	int16_t	逆起電圧検出待ちカウント値	-
g_s2_bemf_time_fg_calc_pre	int16_t	前回の逆起電圧検出電気角 60 度カウント値	-
g_s2_bemf_time_fg_calc_now	int16_t	今回の逆起電圧検出電気角 60 度カウント値	-
g_s2_bemf_time_a_cnt	int16_t	逆起電圧検出 a カウント値	-
g_s2_bemf_time_a_cnt_pre	int16_t	逆起電圧中心値をまたぐ前の逆起電圧検出 a カウント値	-
g_s2_bemf_time_b_cnt	int16_t	逆起電圧検出 b カウント値	-
g_s2_bemf_time_b_cnt_now	int16_t	逆起電圧中心値をまたいだ後の逆起電圧検出 b カウント値	-
g_s2_bemf_time_fg_cnt	int16_t	逆起電圧検出電気角 30 度待ちカウント値	通電パターン切り替えの判定に利用する
g_s2_bemf_fg_th_pre	int16_t	前回の逆起電圧検出電気角 30 度待ちカウント閾値	-
g_s2_bemf_fg_th_now	int16_t	今回の逆起電圧検出電気角 30 度待ちカウント閾値	通電パターン切り替えの判定に利用する
g_s2_bemf_calc_pre	int16_t	前回の逆起電圧検出計算値	-
g_s2_bemf_calc_now	int16_t	今回の逆起電圧検出計算値	-
g_u1_bemf_pwm_phase	uint8_t	通電パターン種別	0 : 上回生 1 : 下回生
g_s2_bemf_timeout_cnt	int16_t	逆起電圧検出タイムアウトカウント値	-
g_s2_bemf_time_pwmchg_cnt	int16_t	Hi-Z 検出 A/D 値取得カウント値	-

g_u1_bemf_chk_hiz	uint8_t	Hi-Z 種別	0: U 相 1: V 相 2: W 相
g_u1_bemf_chk_hiz_status	uint8_t	Hi-Z 検出ステータス管理	0: Hi-Z 検出 未 1: Hi-Z 検出 済
g_u1_bemf_req_moten_on	uint8_t	モータ駆動制御 ON 要求	0: 禁止 1: 許可
g_u1_bemf_req_moten_off	uint8_t	モータ駆動制御 OFF 要求	0: 禁止 1: 許可
g_u1_bemf_mode_cnt	uint8_t	逆起電圧検出モードカウンタ値	-
g_u1_bemf_th_type	uint8_t	逆起電圧検出種別	0: 逆起電圧検出(上回生期間) 1: 逆起電圧検出(電流期間) 2: 逆起電圧検出(下回生期間)
g_u1_bemf_ad_type	uint8_t	A/D 選択種別	0: 電流検出 1: 逆起電圧検出
g_s2_pdrv_ad_isense	int16_t	電流検出 A/D 値	-
g_s2_bemf_fix_pwm_duty_cnt	int16_t	PWM Duty 固定期間カウンタ値	-
g_s2_bemf_timer_cnt_ave	int16_t	PWM Duty クランプ開始電気角	電気角 360 度期間 [us]
g_u1_bemf_limit_pwm_duty_flg	uint8_t	PWM Duty クランプフラグ	0: クランプなし 1: クランプあり
g_u1_bemf_limit_vr_ad_flg	uint8_t	トリガレベル A/D リミットフラグ	0: リミットなし 1: リミットあり
g_u2_bemf_limit_vr_ad_cnt	uint16_t	トリガレベル A/D リミットカウンタ	-
g_u1_bemf_limit_vr_ad	uint8_t	トリガレベル A/D リミット値	-

3.4 マクロ定義仕様

サンプルプログラムのマクロ定義一覧を Table 3-7 に示す。

Table 3-7 マクロ定義一覧

ファイル名	マクロ名	内容	備考
control_parameter.h	CP_MAX_SPEED_RPM	3900	回転速度指令最大値(機械角) [rpm]
	CP_MIN_SPEED_RPM	1200	回転速度指令最小値(機械角) [rpm]
	CP_SPEED_LPF_K	0.35f	回転速度用 LPF パラメータ
mtr_main.h	ICS_UI	0	UI を ICS にセット
	BOARD_UI	1	UI をボードにセット
	M_CW	0	ユーザ回転方向設定値: CW
	M_CCW	1	ユーザ回転方向設定値: CCW
	MAX_SPEED	CP_MAX_SPEED_RPM	回転速度指令最大値(機械角) [rpm]
	MIN_SPEED	CP_MIN_SPEED_RPM	回転速度指令最小値(機械角) [rpm]
	MARGIN_SPEED	100	モータ停止用回転速度指令最小値作用定数(機械角) [rpm]
	MARGIN_MIN_SPEED	MIN_SPEED - MARGIN_SPEED	モータ停止用回転速度指令最小値(機械角) [rpm]
	SPEED_LPF_K	CP_SPEED_LPF_K * 16384	回転速度用 LPF パラメータ
	SW_ON	0	Low アクティブ
	SW_OFF	1	High アクティブ
	REQ_CLR	0	モータ停止指令フラグクリア
	REQ_SET	1	モータ停止指令フラグセット
	LED_ON_1ST_SPEED	1500	LED3 点灯回転数
	LED_ON_2ND_SPEED	3000	LED4 点灯回転数
	REQ_ROT_CCW	0	回転方向ポート取得値: CCW
REQ_ROT_CW	1	回転方向ポート取得値: CW	
motor_parameter.h	MP_POLE_PAIRS	2	極対数補正用定数
mtr_ctrl_r178g1f_t2001.h	MTR_PWM_TIMER_FREQ	64.0f	PWM タイマカウンタ周波数 [MHz]
	MTR_CARRIER_FREQ	10.0f	キャリア周波数 [KHz]
	MTR_DEADTIME	0	デットタイム [ns]
	MTR_DEADTIME_SET	MTR_DEADTIME * MTR_PWM_TIMER_FREQ / 1000	デットタイム設定値
	MTR_CARRIER_SET	(MTR_PWM_TIMER_FREQ * 1000 /	キャリア設定値

		MTR_CARRIER_FREQ / 2) + MTR_DEADTIME_SET - 2	
MTR_START_CARRIER_SET		MTR_CARRIER_SET * 30 / 100	キャリア設定値(初期値)
MTR_VR1_ADC_MAX		802	トリガレベル A/D 変換最大値
MTR_PWM_DUTY_CALC_COEF1		MTR_CARRIER_SET * 100 / MTR_VR1_ADC_MAX	PWM Duty 演算係数 1
MTR_PWM_DUTY_CALC_COEF2		100	PWM Duty 演算係数 2
MTR_RPM_CALC_COEF1		556	目標回転数演算係数 1
MTR_RPM_CALC_COEF2		16888	目標回転数演算係数 2
MTR_RPM_CALC_COEF3		100	目標回転数演算係数 3
MTR_PORT_BEMF_U		P5.2	逆起電圧測定相選択 U 相ポート
MTR_PORT_BEMF_V		P5.3	逆起電圧測定相選択 V 相ポート
MTR_PORT_BEMF_W		P5.4	逆起電圧測定相選択 W 相ポート
MTR_PORT_UP		P1.5	U 相(正相)出力ポート
MTR_PORT_UN		P1.4	U 相(逆相)出力ポート
MTR_PORT_VP		P1.3	V 相(正相)出力ポート
MTR_PORT_VN		P1.1	V 相(逆相)出力ポート
MTR_PORT_WP		P1.2	W 相(正相)出力ポート
MTR_PORT_WN		P1.0	W 相(逆相)出力ポート
MTR_GET_ROT_DIR_REQ		P12.2	回転方向取得ポート
MTR_PORT_LED1		P0.1	LED1 出力ポート
MTR_PORT_LED2		P0.0	LED2 出力ポート
MTR_PORT_LED3		P14.1	LED3 出力ポート
MTR_PORT_LED4		P14.0	LED4 出力ポート
MTR_PORT_LED5		P4.3	LED5 出力ポート
MTR_PORT_LED6		P4.2	LED6 出力ポート
MTR_LED_ON		0	Low アクティブ
MTR_LED_OFF		1	
MTR_OVERVOLTAGE_LIMIT		30 * 128	過電圧エラー判定閾値 [V]
MTR_UNDERVOLTAGE_LIMIT		6 * 128	低電圧エラー判定閾値 [V]
MTR_VDC_SCALING		1471	VM 電圧 A/D 変換値分解能
MTR_RECIVDC_SCALING		256	VM 電圧 A/D 変換値分解能(逆数)
MTR_TAU1_CNT		TCR01	回転速度計測用タイマカウンタレジスタ
MTR_ADCCH_RAJ306000_TEMP		0	RAJ306000 温度測定用 A/D 変換チャンネル
MTR_ADCCH_MOS_TEMP		1	MOS 温度測定用 A/D 変換チャンネル
MTR_ADCCH_VM		3	VM 電圧測定用 A/D 変換チャンネル
MTR_ADCCH_BOARD_TEMP		5	センサ基板温度測定用 A/D 変換チャンネル

	MTR_ADCCH_VR1	6	トリガレベル A/D 変換チャンネル
	MTR_ADCCH_PDRV	7	ブリドライバ電圧測定用 A/D 変換チャンネル
mtr_ctrl_t2001.h	MTR_LED1	1	LED パターン
	MTR_LED2	2	
	MTR_LED3	3	
	MTR_LED4	4	
	MTR_LED5	5	
	MTR_LED6	6	
mtr_spm_less_120_cpm. h	MTR_TWOPI	$2 * 3.14159265f$	2π
	MTR_POLE_PAIRS	MP_POLE_PAIRS	極対数補正用定数
	MTR_RPM_RAD	1716	[rpm]→[rad/s]単位変換用 定数
	MTR_RAD_RPM	2445 / MTR_POLE_PAIRS	[rad/s]→[rpm]単位変換用 定数
	MTR_SPEED_LIMIT_RPM	4290	回転速度リミット値(機械 角) [rpm]
	MTR_SPEED_LIMIT	MTR_SPEED_LIMIT_R PM / 60 * MTR_POLE_PAIRS * MTR_TWOPI * 4	回転速度リミット値(電気 角) [rad/s]
	MTR_SPEED_LPF_K	CP_SPEED_LPF_K * 16384	回転速度用 LPF パラメー タ
	MTR_SPEED_CALC_BASE	383	回転速度計測用定数
	MTR_TIMER_CNT_BUF_NUM	6	回転速度計測タイムカウ ントバッファサイズ
	MTR_TIMEOUT_CNT	200	停止判定時間 (カウント値 * 100[us])
	MTR_START_CNT	2000	BOOT モード時間 (カウント値 * 500[us])
	MTR_PATTERN_CW_V_U	5	CW 電流パターン
	MTR_PATTERN_CW_W_U	4	
	MTR_PATTERN_CW_W_V	6	
	MTR_PATTERN_CW_U_V	2	
	MTR_PATTERN_CW_U_W	3	
	MTR_PATTERN_CW_V_W	1	
	MTR_PATTERN_CCW_V_U	2	CCW 電流パターン
	MTR_PATTERN_CCW_V_W	6	
	MTR_PATTERN_CCW_U_W	4	
	MTR_PATTERN_CCW_U_V	5	
	MTR_PATTERN_CCW_W_V	1	
	MTR_PATTERN_CCW_W_U	3	
	MTR_PATTERN_ERROR	0	通電パターン
	MTR_UP_PWM_VN_ON	1	
	MTR_UP_PWM_WN_ON	2	
	MTR_VP_PWM_UN_ON	3	
	MTR_VP_PWM_WN_ON	4	
	MTR_WP_PWM_UN_ON	5	
	MTR_WP_PWM_VN_ON	6	
MTR_UP_ON_VN_PWM	7		
MTR_UP_ON_WN_PWM	8		
MTR_VP_ON_UN_PWM	9		

MTR_VP_ON_WN_PWM	10	
MTR_WP_ON_UN_PWM	11	
MTR_WP_ON_VN_PWM	12	
MTR_CW	0	回転方向設定値: CW
MTR_CCW	1	回転方向設定値: CCW
MTR_FLG_CLR	0	フラグクリア用定数
MTR_FLG_SET	1	フラグセット用定数
MTR_STOP_WAIT_CNT	100	モータ停止待ち時間 (カウント値 * 100[us])
MTR_ICS_DECIMATION	2	ICS 用関数呼び出し間引き数 (カウント値 * 100[us])
MTR_SENLESS_INIT	0	センサレスモード
MTR_SENLESS_INDSEN	1	
MTR_SENLESS_BEMF	2	
MTR_BEMF_PRE	0	逆起電圧検出モード
MTR_BEMF_1ST	1	
MTR_BEMF_2ND	2	
MTR_BEMF_3RD	3	
MTR_BEMF_UP	0	逆起電圧検出方向
MTR_BEMF_DOWN	1	
MTR_BEMF_WAIT_1ST_1	25	モータ制御信号停止時間 (カウント値 * 100[us])
MTR_BEMF_WAIT_1ST_2	10	逆起電圧検出マスク時間 (カウント値 * 100[us])
MTR_BEMF_WAIT_1ST_3	0	逆起電圧検出(電流期間) マスク時間 (カウント値 * 100[us])
MTR_BEMF_WAIT_2ND	5	Hi-Z 検出開始マスク時間 (カウント値 * 100[us])
MTR_BEMF_WAIT_3RD	50	Hi-Z 検出終了マスク時間 (カウント値 * 100[us])
MTR_BEMF_PWMCHG_WAIT	1	Hi-Z 検出 A/D 値取得カウンタ (カウント値 * 100[us])
MTR_BEMF_TIMEOUT	2500	逆起電圧検出タイムアウト (カウント値 * 100[us])
MTR_BEMF_HIZ_KICK_NUM	1	Hi-Z 検出電流パターン加算値
MTR_BEMF_TH_LIMIT_MAX	1024	逆起電圧検出レベル最大値 (レベル値 * 0.0049[V])
MTR_BEMF_TH_LIMIT_MIN	0	逆起電圧検出レベル最小値 (レベル値 * 0.0049[V])
MTR_BEMF_TH_MARGIN	300	逆起電圧検出レベル制限値 (レベル値 * 0.0049[V])
MTR_PWM_PHASE_P	0	通電パターン種別
MTR_PWM_PHASE_N	1	
MTR_BEMF_MODE_CNT	2	逆起電圧検出モードカウント値

MTR_BEMF_TIMING_TH	50	逆起電圧検出種別切り替え閾値 PWM Duty [%]
MTR_BEMF_FIX_PWMDUTY	35	PWM Duty 固定値 [%]
MTR_BEMF_FIX_PWMDUTY_CNT_LIMIT	32	PWM Duty 固定期間 (カウント値 * 電気角 60 度)
MTR_BEMF_LIMIT_PWMDUTY	55	PWM Duty クランプ値 [%]
MTR_BEMF_LIMIT_PWMDUTY_ELEROT_TIME	1875	PWM Duty クランプ開始する電気角 360 度期間 [us]
MTR_BEMF_LIMIT_VRAD_CNT_LIMIT	1000	トリガレベル A/D リミット実行時間 (カウント値 * 500[us])
MTR_BEMF_LIMIT_VRAD_COEF1	8	トリガレベル A/D リミット計算係数 1
MTR_BEMF_LIMIT_VRAD_COEF2	10	トリガレベル A/D リミット計算係数 2
MTR_WAITTIME_100_US	100	100[us]待ち
MTR_WAITTIME_25_MS	25000	25[ms]待ち
MTR_PATTERN_W_V	0	電流パターン
MTR_PATTERN_W_U	1	
MTR_PATTERN_V_U	2	
MTR_PATTERN_V_W	3	
MTR_PATTERN_U_W	4	
MTR_PATTERN_U_V	5	
MTR_PATTERN_MAX	6	
MTR_BEMF_TH_HSIDE	0	逆起電圧検出種別
MTR_BEMF_TH_HIZ	1	
MTR_BEMF_TH_LSIDE	2	
MTR_BEMF_TH_MAX	3	
MTR_BEMF_HIZ_U	0	Hi-Z 種別
MTR_BEMF_HIZ_V	1	
MTR_BEMF_HIZ_W	2	
MTR_BEMF_HIZ_MAX	3	
MTR_BEMF_AD_ISENSE	0	A/D 選択種別
MTR_BEMF_AD_BEMF	1	
MTR_BEMF_AD_MAX	2	
MTR_BEMF_SEQ_INIT	0	逆起電圧シーケンス定義
MTR_BEMF_SEQ_WAIT	1	
MTR_BEMF_SEQ_SET_MOTEN_OFF	2	
MTR_BEMF_SEQ_WAIT_2ND	3	
MTR_BEMF_SEQ_CHK_HIZ	4	
MTR_BEMF_SEQ_SET_MOTEN_ON	5	
MTR_BEMF_SEQ_WAIT_3RD	6	
MTR_BEMF_SEQ_CHK_TH_RANGE	7	
MTR_BEMF_SEQ_DET_BEMF_LEVEL	8	
MTR_BEMF_SEQ_SET_PWM	9	
MTR_BEMF_SEQ_OVF_COUNT	10	
MTR_BEMF_SEQ_NUM_MAX	11	
MTR_BOOT_MODE	0x00	BOOT モード
MTR_LESS_120_MODE	0x03	通常運転(LESS 120)モード
MTR_ALARM_ERROR	0x01	ALARM エラー

	MTR_OVER_VOLTAGE_ERROR	0x02	過電圧エラー
	MTR_OVER_SPEED_ERROR	0x04	回転速度異常度エラー
	MTR_TIMEOUT_ERROR	0x08	タイムアウトエラー
	MTR_LESS_ERROR	0x10	電流パターンエラー
	MTR_BEMF_ERROR	0x20	逆起電圧検出エラー
	MTR_UNDER_VOLTAGE_ERROR	0x40	低電圧エラー
	MTR_UNKNOWN_ERROR	0x80	未定義エラー
	MTR_MODE_STOP	0x00	停止状態
	MTR_MODE_RUN	0x01	回転中
	MTR_MODE_ERROR	0x02	エラー状態
	MTR_SIZE_STATE	3	状態数
	MTR_EVENT_STOP	0x00	モータ停止イベント
	MTR_EVENT_RUN	0x01	モータ起動イベント
	MTR_EVENT_ERROR	0x02	モータエラーイベント
	MTR_EVENT_RESET	0x03	モータリセットイベント
	MTR_SIZE_EVENT	4	イベント数
r_less_120_is.h	IS_DIRECTION_CW	MTR_CW	回転方向: CW
	IS_DIRECTION_CCW	MTR_CCW	回転方向: CCW
	IS_SECTION_TIME_A	2	a 区間時間 (* 時間分解能[us])
	IS_SECTION_TIME_B	4	b 区間時間 (* 時間分解能[us])
	IS_SECTION_TIME_C	4	c 区間時間 (* 時間分解能[us])
	IS_TIME_RESOLUTION	20	時間分解能[us]
	IS_PWM_OUT_TIME	125	モータ制御信号出力時間 (* 時間分解能[us])
	IS_PWM_OFF_TIME	5	モータ制御信号停止時間 (* 時間分解能[us])
	IS_KICK_NUM	1	電流パターン加算値
	IS_TRIAL_NUM	4	インダクティブセンス試 行回数
	IS_STATE_INIT	0	初期値
	IS_STATE_NORMAL	1	正常終了
	IS_STATE_ERROR_PORT	2	異常状態: 汎用ポート設 定
	IS_STATE_ERROR_WAIT	3	異常状態: 時間指定待機
	IS_STATE_ERROR_WDT	4	異常状態: ウォッチドッ グ・タイマ クリア
	IS_STATE_ERROR_ADC	5	異常状態: A/D 変換結果取 得
	IS_STATE_MAX	6	最大値
	IS_PATTERN_1	0	電流パターン 1
	IS_PATTERN_2	1	電流パターン 2
	IS_PATTERN_3	2	電流パターン 3
	IS_PATTERN_4	3	電流パターン 4
	IS_PATTERN_5	4	電流パターン 5
	IS_PATTERN_6	5	電流パターン 6
	IS_PATTERN_MAX	6	最大値
r_less_120_isw.h	IS_PORT_OFF	0x00U	汎用ポート設定: all OFF
	IS_PORT_P_ON	0x2CU	汎用ポート設定: U_p , V_p , $W_p = ON$

	IS_PORT_N_ON	0x13U	汎用ポート設定: U _n , V _n , W _n = ON
	ISW_STATE_INIT	0	初期値
	ISW_STATE_NORMAL	1	正常終了
	ISW_STATE_ERROR	2	異常終了
	ISW_STATE_MAX	3	最大値
version.h	FW_REVISION	103	F/W リビジョン情報

ファイル名	マクロ名	内容	備考
r_cg_userdefine.h	SPI00_CS_H	P0 = P0 0x20	SPI 通信 Chip Select 信号 = H
	SPI00_CS_L	P0 = P0 & 0xDF	SPI 通信 Chip Select 信号 = L
	SPI_WAIT_MODE	0x01	SPI 通信 Wait モード
	SPI_INTR_MODE	0x02	SPI 通信 Interrupt モード
	SPI_ADC_MODE	0x03	SPI 通信 ADC モード

ファイル名	マクロ名	内容	備考	
r_cg_predrv.h	REG_BUFF_SIZE	2	プリドライバレジスタバッファサイズ	
	SPI_CHK_MAX	100	プリドライバSPI通信チェック回数	
	PREDRV_NORMAL	0	プリドライバシーケンスなし	
	PREDRV_SPI_ERROR	1	プリドライバシーケンス SPI 通信エラー	
	PREDRV_ALARM_ERROR	2	プリドライバシーケンス ALARM エラー	
	PREDRV_REGRW_ERROR	4	プリドライバシーケンス レジスタ R/W エラー	
	PREDRV_SPI_ACCESS_OK	0x6A	プリドライバSPI通信判定	
	PREDRV_ALMRAW1_OK	0xEF	プリドライバ ALMRAW1 判定	
	HALL_SIG_MASK	0x7F	HALL_SIG マスク	
	ALMSTS1_TSD_N	0x01	ALARM ステータス 1 判定	
	ALMSTS1_OCP_N	0x02		
	ALMSTS1_VGB_UVP_N	0x04		
	ALMSTS1_VGB_OVP_N	0x08		
	ALMSTS1_VGT_UVP_N	0x10		
	ALMSTS1_VGT_OVP2_N	0x20		
	ALMSTS1_VGT_OVP1_N	0x40		
	ALMSTS1_VREG5_OVP_N	0x80		
	ALMSTS1_NO_ERROR	0xEF		
	ALMSTS1_VGT_UVP_MASK	0xEF		
	ALMSTS2_VM_UVP_N	0x01		ALARM ステータス 2 判定
	ALMSTS2_DI_SEL_W_CMP_N	0x20		
	ALMSTS2_DI_SEL_V_CMP_N	0x40		
	ALMSTS2_DI_SEL_U_CMP_N	0x80		
	ALMSTS2_NO_ERROR	0xFF		
	WHO_AM_I_MASK	0xFE	WHO_AM_I マスク	
	INIT_PS_ALL	0x01	PS_ALL 初期値	
	INIT_PS_1ST	0x3E	PS 初期値 1st	
	INIT_PS_2ND	0x3F	PS 初期値 2nd	
	INIT_PS_3RD	0xBF	PS 初期値 3rd	
	INIT_SELSIG_U	0x03	SELSIG_U 初期値	
	INIT_SELSIG_V	0x14	SELSIG_V 初期値	
	INIT_SELSIG_W	0x25	SELSIG_W 初期値	
	INIT_HALL_SIG	0xA0	HALL_SIG 初期値	
	INIT_ALMOPE1	0x10	ALMOPE1 初期値	
	INIT_ALMOUT1	0x10	ALMOUT1 初期値	
	INIT_CS_SET2	0x60	CS_SET2 初期値	
	INIT_ERROR_WAIT	0x00	ERROR_WAIT 初期値	
	INIT_CS_SET1	0x08	CS_SET1 初期値	
	INIT_HAIC_TH	0x00	HAIC_TH 初期値	
	INIT_LD_WAIT	0x00	LD_WAIT 初期値	
INIT_DRIVE_SET	0x01	DRIVE_SET 初期値		

INIT_IDRCNT_H	0x00	IDRCNT_H 初期値
INIT_IDRCNT_L	0x00	IDRCNT_L 初期値
INIT_TRCNT_P	0x00	TRCNT_P 初期値
INIT_CPSET1	0x01	CPSET1 初期値
INIT_CPSET2	0x02	CPSET2 初期値
IINIT_CP_TRIM	0x00	CP_TRIM 初期値
INIT_VREG5_TRIM	0x20	VREG5_TRIM 初期値
INIT_CSAMP_TRIM	0x20	CSAMP_TRIM 初期値
INIT_TRIM_PT	0x00	TRIM_PT 初期値 プロテクト設定
INIT_TRIM_PT_UP	0x95	TRIM_PT 初期値 プロテクト解除
INIT_TRIM_EN	0x00	TRIM_EN 初期値
INIT_TRIM_EN_EFWD	0x01	TRIM_EN 初期値トリ ミングデータ有効設定 値
INIT_BGR_TRIM	0x00	BGR_TRIM 初期値
INIT_BFAMP_TRIM	0x00	BFAMP_TRIM 初期値
ERRRCV_PS_1ST	0x3C	PS ALARM 復帰値 1st
ERRRCV_PS_2ND	0x3E	PS ALARM 復帰値 2nd
ERRRCV_PS_3RD	0x3F	PS ALARM 復帰値 3rd
ERRRCV_PS_4TH	0xBF	PS ALARM 復帰値 4th
ERRRCV_MOT_EN_CLR	0x00	DRIVE_SET ALARM 復帰値 モータ回転禁止
ERRRCV_MOT_EN_SET	0x01	DRIVE_SET ALARM 復帰値 モータ回転許可
ERRRCV_ALM_LATCH_CLR	0x40	DRIVE_SET ALARM 復帰値 ALARM ラッチクリア
WAITTIME_1_MS	0x11F8	1[ms]待ち
WAITTIME_3_MS	0x35E8	3[ms]待ち
REQ_MOT_EN_OFF	0x00	モータ駆動制御 OFF
REQ_MOT_EN_ON	0x01	モータ駆動制御 ON
SET_MOT_EN_CLR	0x00	DRIVE_SET モータ回転禁止
SET_MOT_EN_SET	0x01	DRIVE_SET モータ回転許可
REQ_ADC_SEL_ISEN	0x00	電流設定指示
REQ_ADC_SEL_BEMF	0x01	逆起電圧設定指示
SET_ADC_SEL_ISEN	0x01	ADC_SEL 電流設定
SET_ADC_SEL_BEMF	0x03	ADC_SEL 逆起電圧 設定
INIT_ICS_PS_ALL	INIT_PS_ALL	PS_ALL ICS 変数初期 値
INIT_ICS_PS	INIT_PS_3RD	PS ICS 変数初期値
INIT_ICS_SW_RESET	0x00	SW_RESET ICS 変数 初期値
INIT_ICS_ADC_SEL	0x00	ADC_SEL ICS 変数初 期値

INIT_ICS_SELSIG_U	0x03	SELSIG_U ICS 変数初期値
INIT_ICS_SELSIG_V	0x14	SELSIG_V ICS 変数初期値
INIT_ICS_SELSIG_W	0x25	SELSIG_W ICS 変数初期値
INIT_ICS_HALL_SIG	INIT_HALL_SIG	HALL_SIG ICS 変数初期値
INIT_ICS_ALMSTS1	0xFF	ALMSTS1 ICS 変数初期値
INIT_ICS_ALMOPE1	INIT_ALMOPE1	ALMOPE1 ICS 変数初期値
INIT_ICS_ALMOUT1	INIT_ALMOUT1	ALMOUT1 ICS 変数初期値
INIT_ICS_ALMSTS2	0xFF	ALMSTS2 ICS 変数初期値
INIT_ICS_CS_SET2	INIT_CS_SET2	CS_SET2 ICS 変数初期値
INIT_ICS_ALMOUT2	0x00	ALMOUT2 ICS 変数初期値
INIT_ICS_ERROR_WAIT	0x00	ERROR_WAIT ICS 変数初期値
INIT_ICS_CS_SET1	INIT_CS_SET1	CS_SET1 ICS 変数初期値
INIT_ICS_HAIC_TH	0x00	HAIC_TH ICS 変数初期値
INIT_ICS_PDDSTS	0xF0	PDDSTS ICS 変数初期値
INIT_ICS_LD_WAIT	0x00	LD_WAIT ICS 変数初期値
INIT_ICS_DRIVE_SET	INIT_DRIVE_SET	DRIVE_SET ICS 変数初期値
INIT_ICS_DI_TIME	0x00	DI_TIME ICS 変数初期値
INIT_ICS_IDRCNT_H	0x00	IDRCNT_H ICS 変数初期値
INIT_ICS_IDRCNT_L	0x00	IDRCNT_L ICS 変数初期値
INIT_ICS_TRCNT_P	0x00	TRCNT_P ICS 変数初期値
INIT_ICS_CPSET1	0x01	CPSET1 ICS 変数初期値
INIT_ICS_CPSET2	0x02	CPSET2 ICS 変数初期値
INIT_ICS_CP_TRIM	INIT_CP_TRIM	CP_TRIM ICS 変数初期値
INIT_ICS_VREG5_TRIM	INIT_VREG5_TRIM	VREG5_TRIM ICS 変数初期値
INIT_ICS_CSAMP_TRIM	INIT_CSAMP_TRIM	CSAMP_TRIM ICS 変数初期値
INIT_ICS_ALMRAW1	0xFF	ALMRAW1 ICS 変数初期値
INIT_ICS_TOIN_MONI	0x00	TOIN_MONI ICS 変数初期値

INIT_ICS_WHO_AM_I	0x6A	WHO_AM_I ICS 変数初期値
INIT_ICS_TRIM_PT	INIT_TRIM_PT	TRIM_PT ICS 変数初期値
INIT_ICS_TRIM_EN	INIT_TRIM_EN	TRIM_EN ICS 変数初期値
INIT_ICS_BGR_TRIM	INIT_BGR_TRIM	BGR_TRIM ICS 変数初期値
INIT_ICS_BFAMP_TRIM	INIT_BFAMP_TRIM	BFAMP_TRIM ICS 変数初期値
SEQ_INIT	0	プリドライバ初期化シーケンス定義
SEQ_CHK_SPI	1	
SEQ_CHK_TSD_N	2	
SEQ_SET_HALL_SIG_PRM	3	
SEQ_SET_5VTRIM	4	
SEQ_SET_ALMOPE1_PRM	5	
SEQ_SET_ALMOUT1_PRM	6	
SEQ_SET_CS_SET2_PRM	7	
SEQ_SET_CS_SET1_PRM	8	
SEQ_SET_SEQINIT_PRM	9	
SEQ_SET_PS_ALL_PRM	10	
SEQ_SET_PS_1ST_PRM	11	
SEQ_SET_PS_2ND_PRM	12	
SEQ_CHK_ALMRAW1	13	
SEQ_SET_PS_3RD_PRM	14	
SEQ_CHK_ALMSTS	15	
SEQ_SET_MOT_EN	16	
SEQ_END	17	
SEQ_NUM_MAX	18	
ERR_RCV_SEQ_INIT	0	プリドライバALARM復帰シーケンス定義
ERR_RCV_SEQ_CHK_STS	1	
ERR_RCV_SEQ_CLR_MOT_EN	2	
ERR_RCV_SEQ_SET_PS_1ST	3	
ERR_RCV_SEQ_CHK_ALMSTS_1ST	4	
ERR_RCV_SEQ_SET_ALM_LATCH_CLR	5	
ERR_RCV_SEQ_SET_PS_2ND	6	
ERR_RCV_SEQ_SET_PS_3RD	7	
ERR_RCV_SEQ_CHK_ALMRAW1	8	
ERR_RCV_SEQ_SET_PS_4TH	9	
ERR_RCV_SEQ_CHK_ALMSTS_2ND	10	
ERR_RCV_SEQ_SET_MOT_EN	11	
ERR_RCV_SEQ_END	12	
ERR_RCV_SEQ_NUM_MAX	13	

3.5 フローチャート

Figure 3-8 に全体フローチャートと初期化関数のフローチャートを示す。

また、サンプルプログラムの主要な処理のフローチャートを Figure 3-8 から Figure 3-18 に示す。

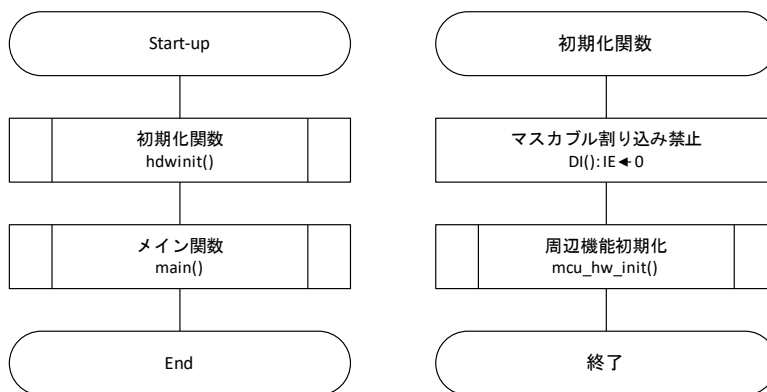


Figure 3-8 フローチャート(全体、初期化関数)

3.5.1 メイン関数

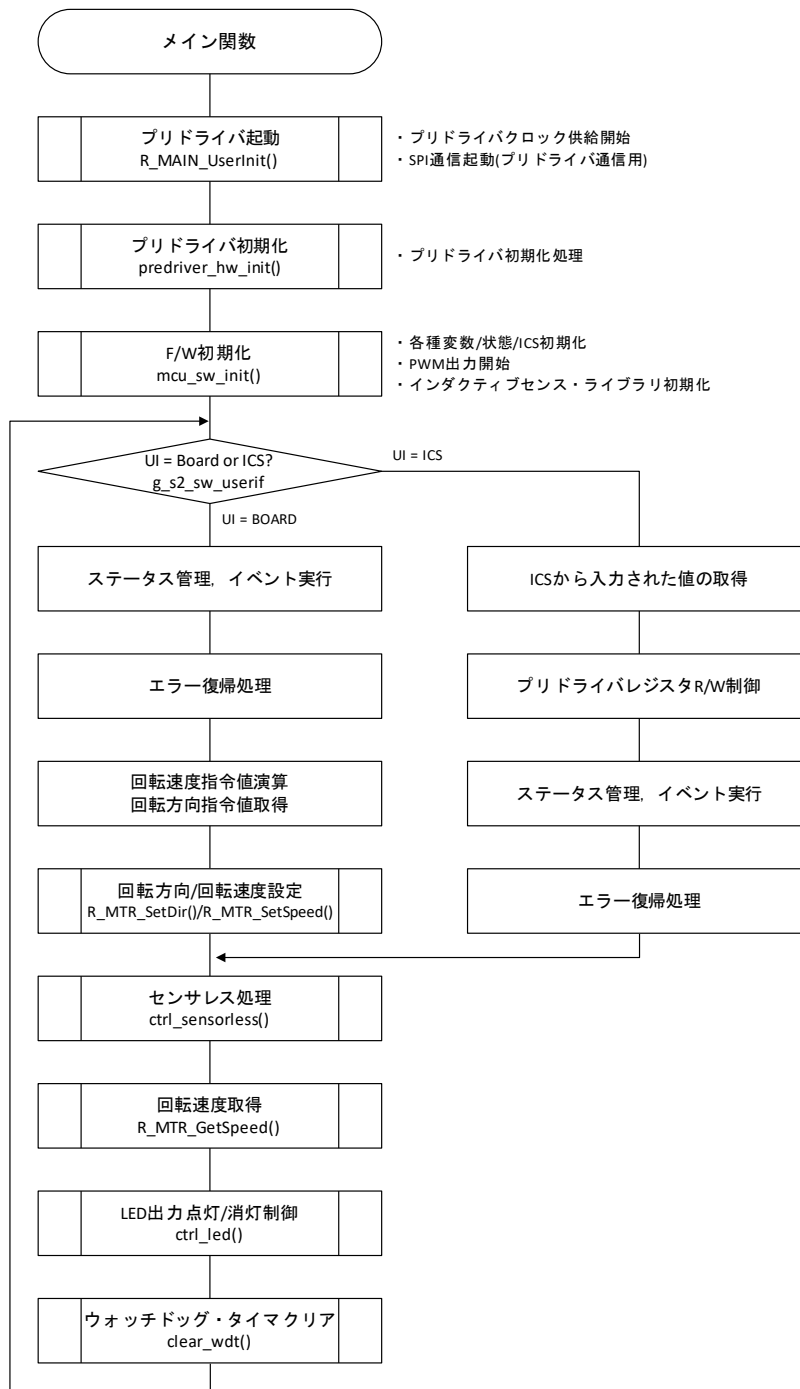


Figure 3-9 フローチャート(メイン関数)

3.5.2 プリドライバ初期化処理

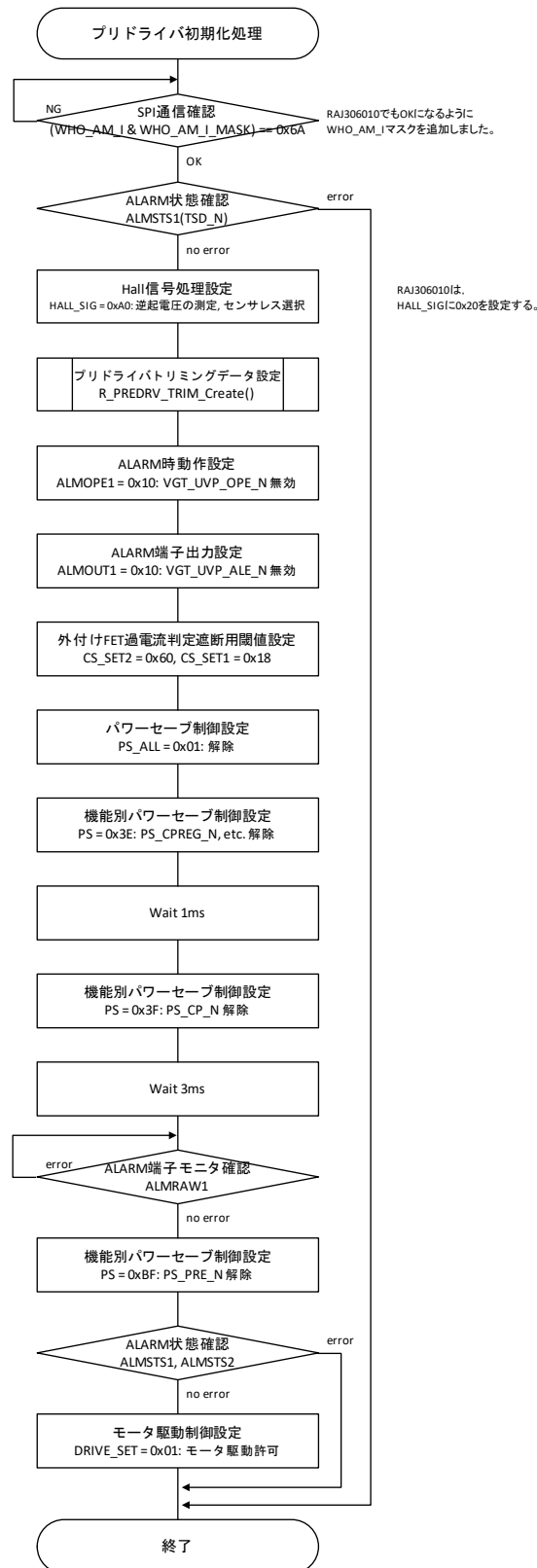


Figure 3-10 フローチャート(プリドライバ初期化処理)

3.5.3 センサレス処理: インダクティブセンス処理

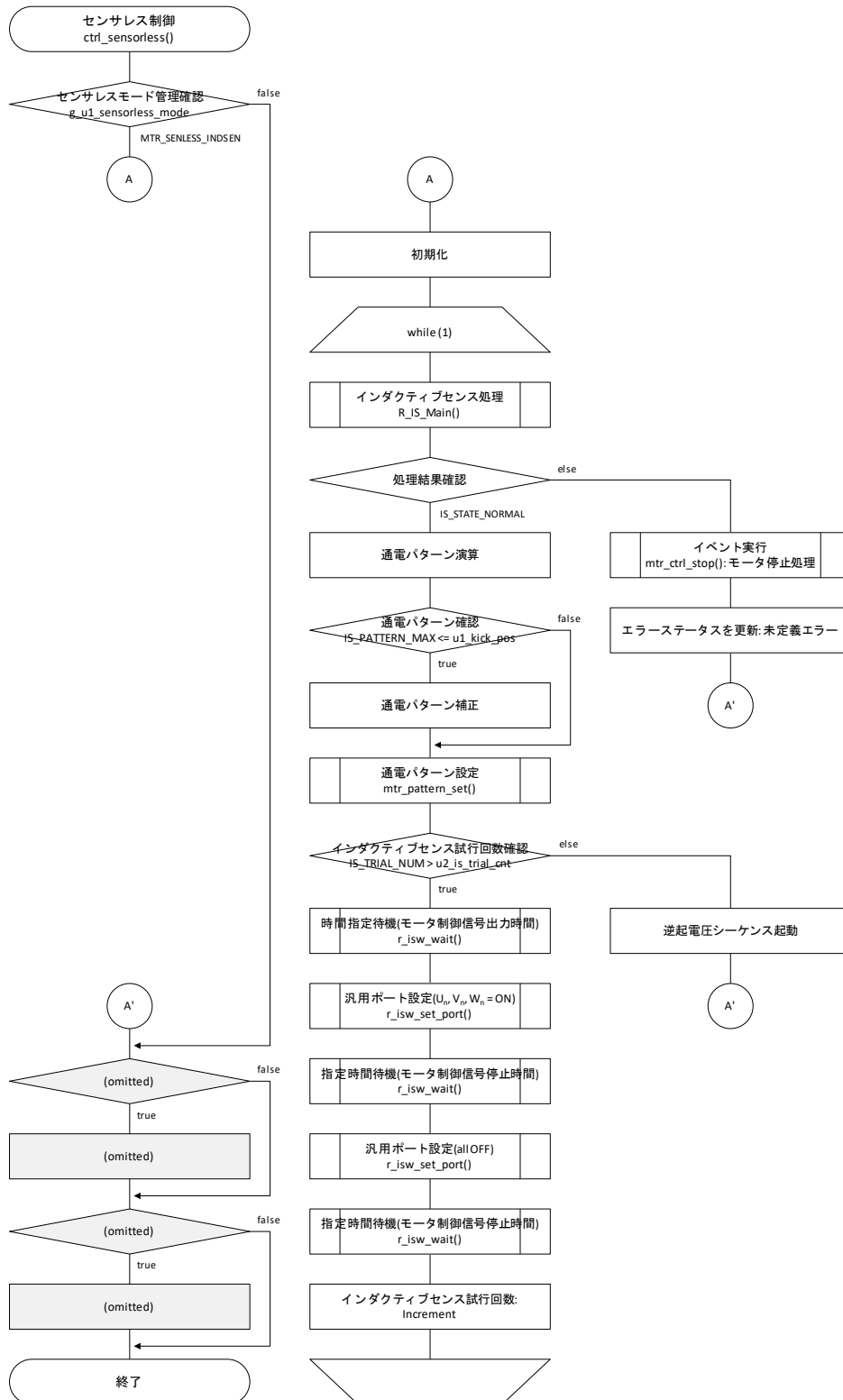


Figure 3-11 フローチャート(センサレス処理: インダクティブセンス処理)

3.5.4 キャリア周波数割り込み処理

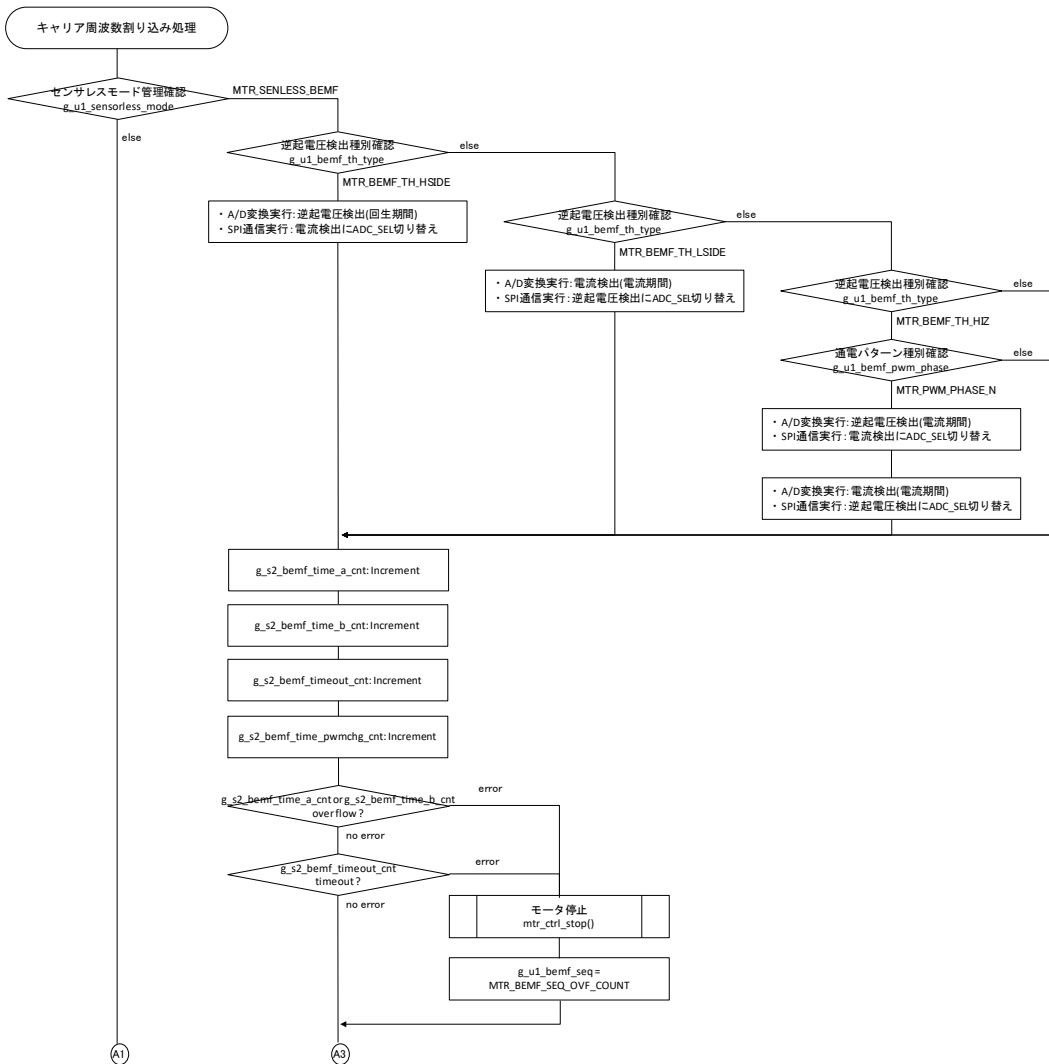


Figure 3-12 フローチャート(キャリア周波数割り込み処理)

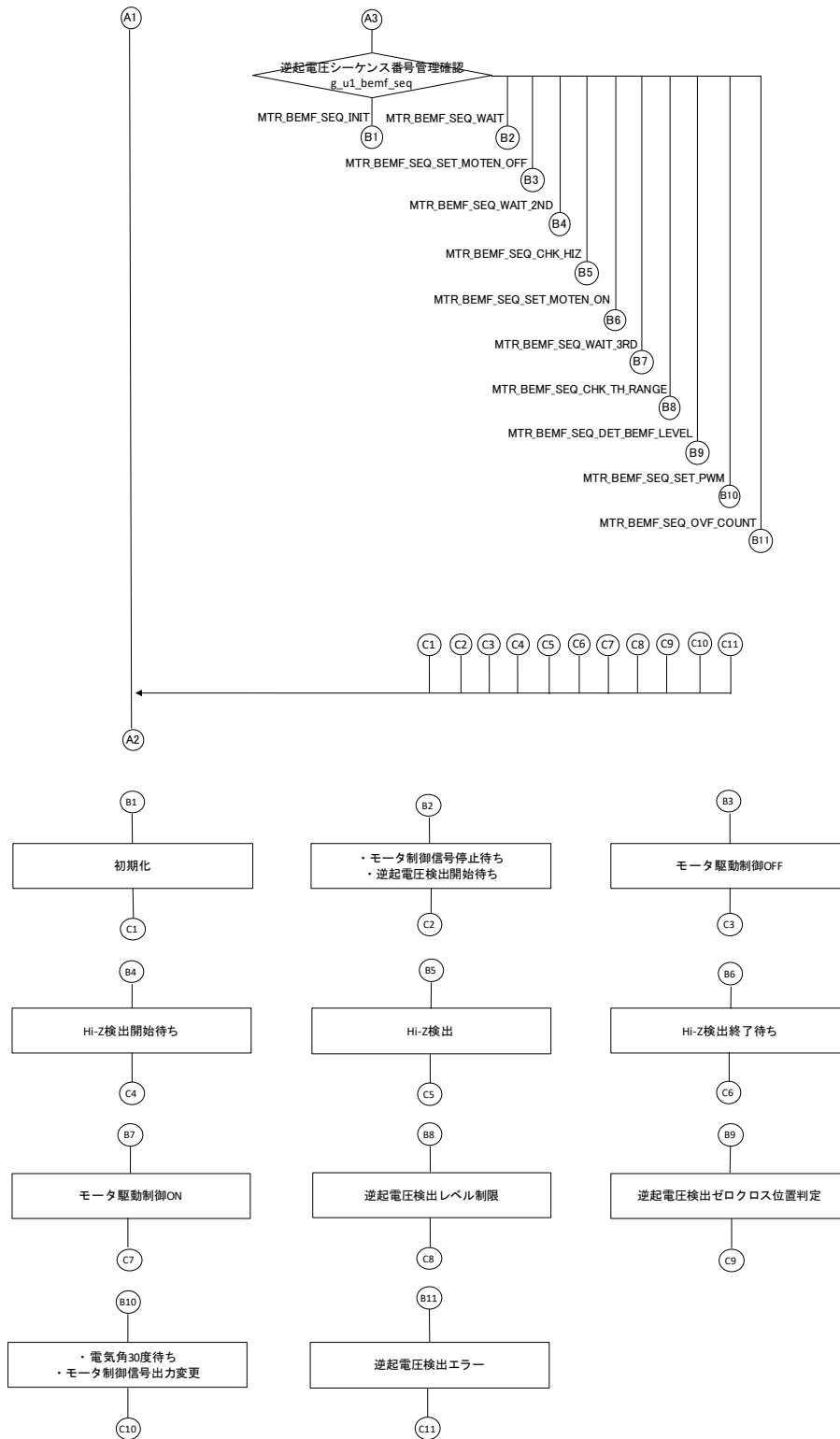


Figure 3-13 フローチャート(キャリア周波数割り込み処理)

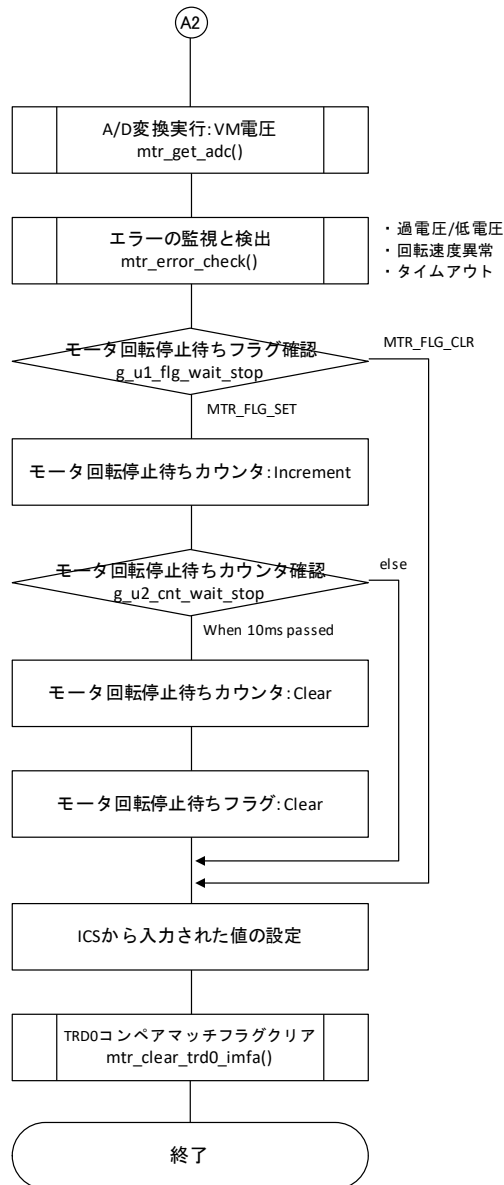


Figure 3-14 フローチャート(キャリア周波数割り込み処理)

3.5.5 キャリア周波数アンダフロー割り込み処理

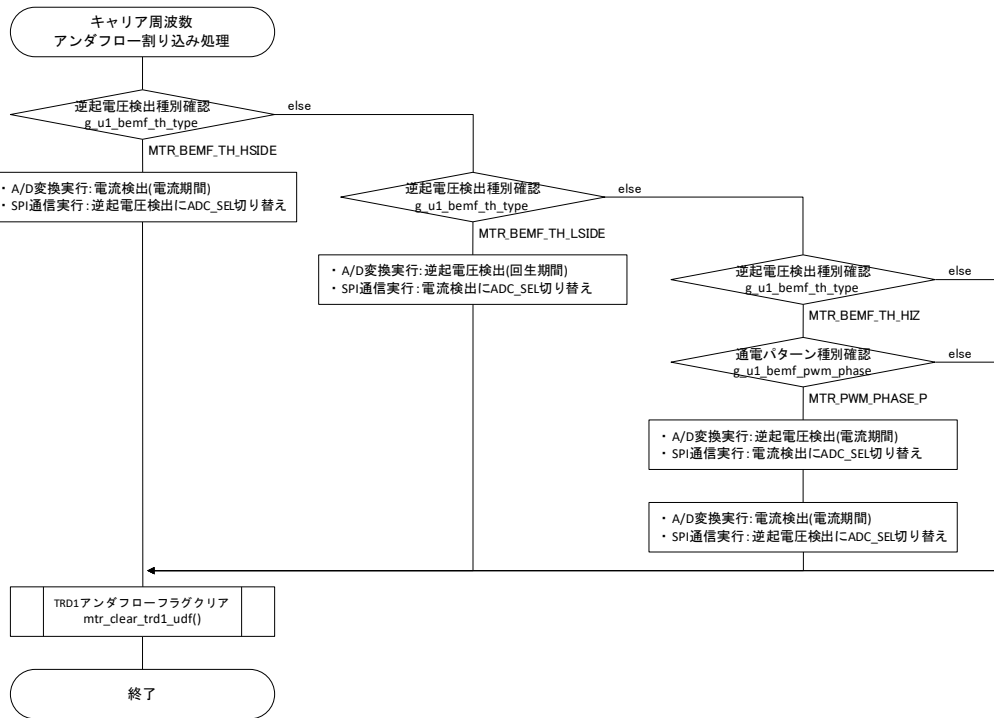


Figure 3-15 フローチャート(キャリア周波数アンダフロー割り込み処理)

3.5.6 500[us] 割り込み処理

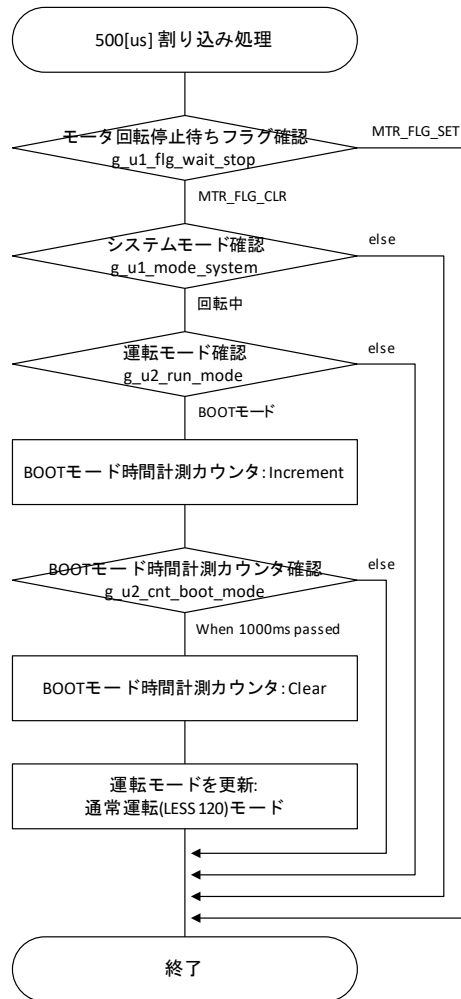


Figure 3-16 フローチャート(500[us] 割り込み処理)

3.5.7 ALARM 割り込み処理

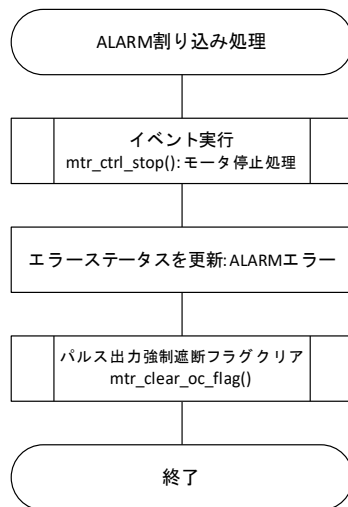


Figure 3-17 フローチャート(ALARM 割り込み処理)

3.5.8 ALARM 復帰処理

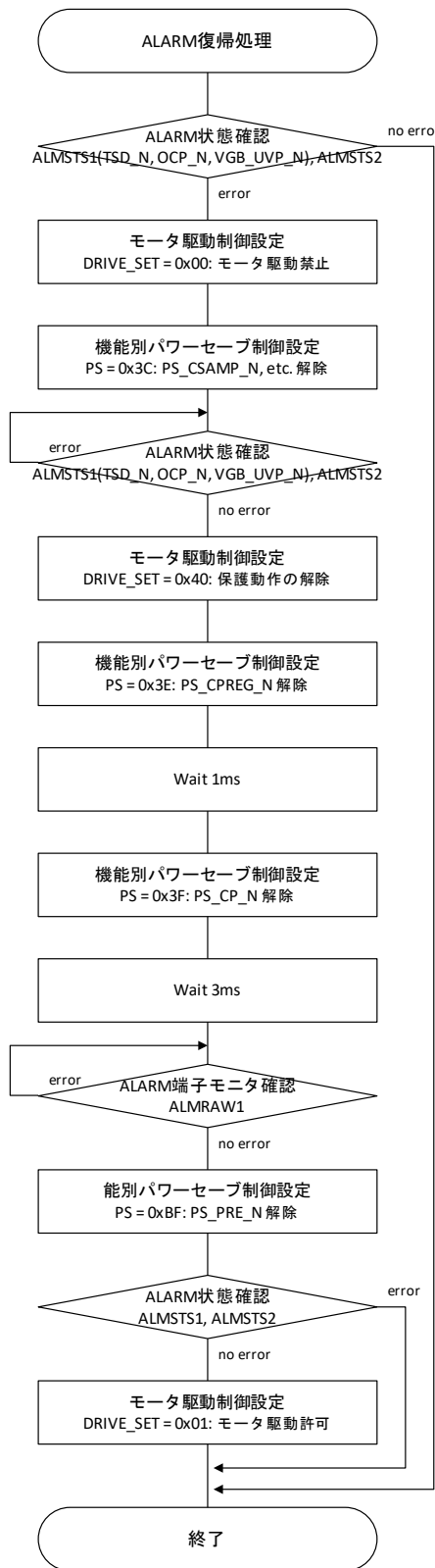


Figure 3-18 フローチャート(ALARM 復帰処理)

4. 開発支援ツール「In Circuit Scope」

4.1 概要

本アプリケーションノート対象サンプルプログラムでは、ICSによるユーザインタフェース(回転/停止指令、回転速度指令等)が使用可能です。ICSはターゲットシステム上で実行されるプログラムのグローバル変数値をリアルタイムにパソコン上に波形表示することができるツールです。使用方法などの詳細は「In Circuit Scope 取扱説明書」を参照して下さい。

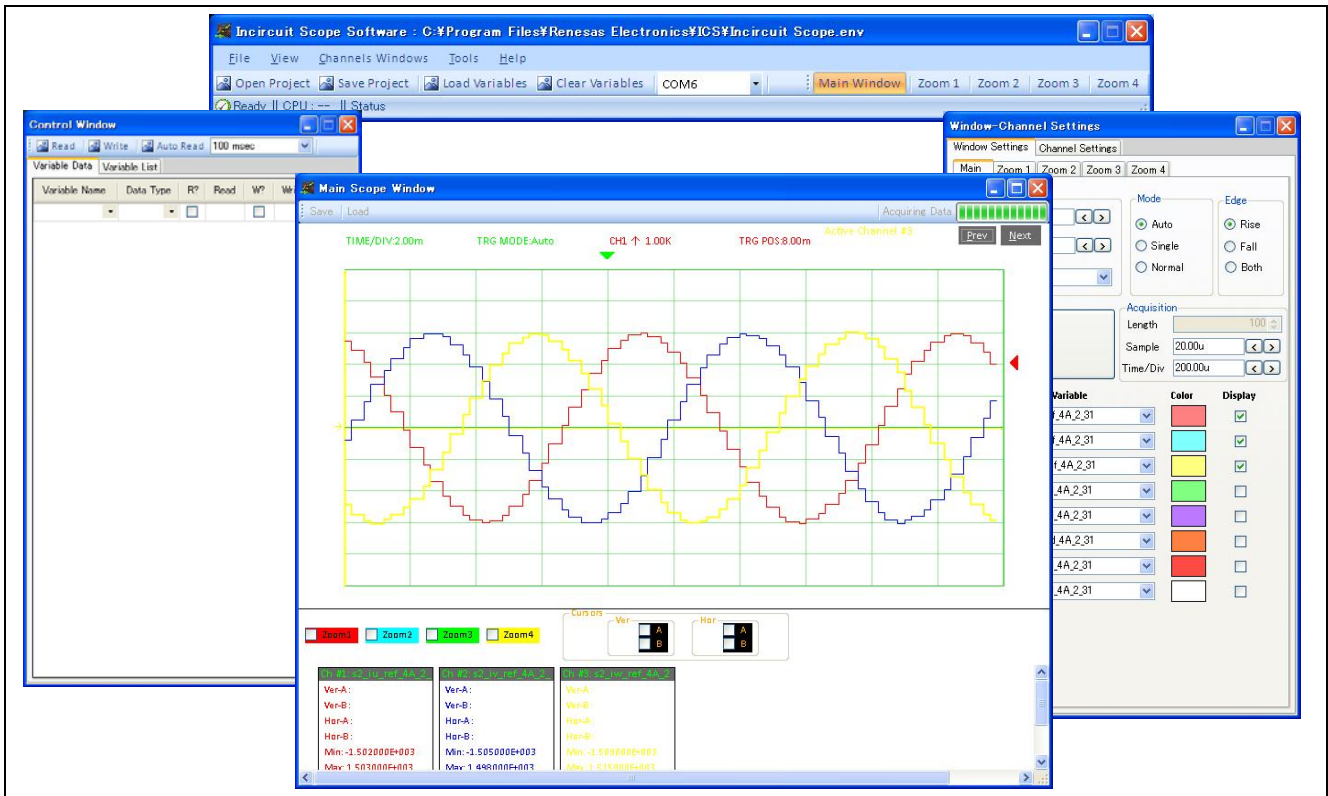


Figure 4-1 ICS 外観

4.2 ライブラリ使用方法

ICSを使用する場合、ICSに関連する関数を呼び出す必要があります。ICS関連関数は条件コンパイル(#ifdef-#endif)で設定しています。ICSを使用する場合は下記の通り設定してください。

【ファイル名】 mtr_common.h

【変更箇所】 下記の宣言を加えてください。

```
#define ICS_USE
```

4.3 ICS 用変数一覧

ICS 用変数の一覧を Table 4-1, Table 4-2 に示す。Table 4-1 の変数の値は com_s2_enable_write に g_s2_enable_write と同じ値を書込んだ場合にプロテクト変数へ反映される。また、Table 4-2 の変数は com_s2_enable_write に依存しない。

Table 4-1 ICS 用変数一覧

変数名	型	内容	備考 (【 】:プロテクト変数名)
com_s2_direction	int16_t	回転方向 0 : CW 1 : CCW	【g_u1_direction】
com_s2_ref_speed_rpm	int16_t	回転速度指令値 (機械角) [rpm] ※ 未使用	【g_s2_ref_speed】
com_s2_speed_lpf_k	int16_t	速度 LPF パラメータ	【g_s2_speed_lpf_k】
com_s2_enable_write	int16_t	変数書き換え許可	-

Table 4-2 ICS 用変数一覧

変数名	型	内容	備考
com_s2_sw_userif	int16_t	ユーザインタフェーススイッチ 0 : ICS ユーザインタフェース使用 1 : ボードユーザインタフェース使用	-
com_s2_mode_system	int16_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_s2_pwm_duty	int16_t	タイマ RD コンペアレジスタ設定値	PWM Duty 設定値: 0 ~ 3198 PWM Duty [%]: 0 ~ 100
com_u1_pdrvreg_ctrl	uint8_t	ブリドライバレジスタ R/W コントロール フラグ	0: R/W 禁止 1: R/W 許可
com_u1_pdrvreg_ps_all_pre	uint8_t	ブリドライバレジスタ PS_ALL 前回値	Read 値
com_u1_pdrvreg_ps_all_now	uint8_t	ブリドライバレジスタ PS_ALL 現在値	Write 値
com_u1_pdrvreg_ps_pre	uint8_t	ブリドライバレジスタ PS 前回値	Read 値
com_u1_pdrvreg_ps_now	uint8_t	ブリドライバレジスタ PS 現在値	Write 値
com_u1_pdrvreg_sw_reset_pre	uint8_t	ブリドライバレジスタ SW_RESET 前回 値	Read 値
com_u1_pdrvreg_sw_reset_now	uint8_t	ブリドライバレジスタ SW_RESET 現在 値	Write 値
com_u1_pdrvreg_adc_sel_pre	uint8_t	ブリドライバレジスタ ADC_SEL 前回値	Read 値
com_u1_pdrvreg_adc_sel_now	uint8_t	ブリドライバレジスタ ADC_SEL 現在値	Write 値
com_u1_pdrvreg_selsig_u_pre	uint8_t	ブリドライバレジスタ SELSIG_U 前回値	Read 値
com_u1_pdrvreg_selsig_u_now	uint8_t	ブリドライバレジスタ SELSIG_U 現在値	Write 値
com_u1_pdrvreg_selsig_v_pre	uint8_t	ブリドライバレジスタ SELSIG_V 前回値	Read 値
com_u1_pdrvreg_selsig_v_now	uint8_t	ブリドライバレジスタ SELSIG_V 現在値	Write 値
com_u1_pdrvreg_selsig_w_pre	uint8_t	ブリドライバレジスタ SELSIG_W 前回 値	Read 値
com_u1_pdrvreg_selsig_w_now	uint8_t	ブリドライバレジスタ SELSIG_W 現在 値	Write 値
com_u1_pdrvreg_hall_sig_pre	uint8_t	ブリドライバレジスタ HALL_SIG 前回値	Read 値
com_u1_pdrvreg_hall_sig_now	uint8_t	ブリドライバレジスタ HALL_SIG 現在値	Write 値
com_u1_pdrvreg_almsts1_pre	uint8_t	ブリドライバレジスタ ALMSTS1 前回値	Read 値 (ALMSTS1 Read Only)
com_u1_pdrvreg_almope1_pre	uint8_t	ブリドライバレジスタ ALMOPE1 前回値	Read 値

com_u1_pdrvreg_almope1_now	uint8_t	ブリドライバレジスタ ALMOPE1 現在値	Write 値
com_u1_pdrvreg_almout1_pre	uint8_t	ブリドライバレジスタ ALMOUT1 前回値	Read 値
com_u1_pdrvreg_almout1_now	uint8_t	ブリドライバレジスタ ALMOUT1 現在値	Write 値
com_u1_pdrvreg_almsts2_pre	uint8_t	ブリドライバレジスタ ALMSTS2 前回値	Read 値 (ALMSTS2 Read Only)
com_u1_pdrvreg_cs_set2_pre	uint8_t	ブリドライバレジスタ CS_SET2 前回値	Read 値
com_u1_pdrvreg_cs_set2_now	uint8_t	ブリドライバレジスタ CS_SET2 現在値	Write 値
com_u1_pdrvreg_almout2_pre	uint8_t	ブリドライバレジスタ ALMOUT2 前回値	Read 値
com_u1_pdrvreg_almout2_now	uint8_t	ブリドライバレジスタ ALMOUT2 現在値	Write 値
com_u1_pdrvreg_error_wait_pre	uint8_t	ブリドライバレジスタ ERROR_WAIT 前回値	Read 値
com_u1_pdrvreg_error_wait_now	uint8_t	ブリドライバレジスタ ERROR_WAIT 現在値	Write 値
com_u1_pdrvreg_cs_set1_pre	uint8_t	ブリドライバレジスタ CS_SET1 前回値	Read 値
com_u1_pdrvreg_cs_set1_now	uint8_t	ブリドライバレジスタ CS_SET1 現在値	Write 値
com_u1_pdrvreg_haic_th_pre	uint8_t	ブリドライバレジスタ HAIC_TH 前回値	Read 値
com_u1_pdrvreg_haic_th_now	uint8_t	ブリドライバレジスタ HAIC_TH 現在値	Write 値
com_u1_pdrvreg_pddsts_pre	uint8_t	ブリドライバレジスタ PDDSTS 前回値	Read 値 (PDDSTS Read Only)
com_u1_pdrvreg_ld_wait_pre	uint8_t	ブリドライバレジスタ LD_WAIT 前回値	Read 値
com_u1_pdrvreg_ld_wait_now	uint8_t	ブリドライバレジスタ LD_WAIT 現在値	Write 値
com_u1_pdrvreg_drive_set_pre	uint8_t	ブリドライバレジスタ DRIVE_SET 前回値	Read 値
com_u1_pdrvreg_drive_set_now	uint8_t	ブリドライバレジスタ DRIVE_SET 現在値	Write 値
com_u1_pdrvreg_di_time_pre	uint8_t	ブリドライバレジスタ DI_TIME 前回値	Read 値
com_u1_pdrvreg_di_time_now	uint8_t	ブリドライバレジスタ DI_TIME 現在値	Write 値
com_u1_pdrvreg_idrcnt_h_pre	uint8_t	ブリドライバレジスタ IDRCNT_H 前回値	Read 値
com_u1_pdrvreg_idrcnt_h_now	uint8_t	ブリドライバレジスタ IDRCNT_H 現在値	Write 値
com_u1_pdrvreg_idrcnt_l_pre	uint8_t	ブリドライバレジスタ IDRCNT_L 前回値	Read 値
com_u1_pdrvreg_idrcnt_l_now	uint8_t	ブリドライバレジスタ IDRCNT_L 現在値	Write 値
com_u1_pdrvreg_trcnt_p_pre	uint8_t	ブリドライバレジスタ TRCNT_P 前回値	Read 値
com_u1_pdrvreg_trcnt_p_now	uint8_t	ブリドライバレジスタ TRCNT_P 現在値	Write 値
com_u1_pdrvreg_cpset1_pre	uint8_t	ブリドライバレジスタ CPSET1 前回値	Read 値
com_u1_pdrvreg_cpset1_now	uint8_t	ブリドライバレジスタ CPSET1 現在値	Write 値
com_u1_pdrvreg_cpset2_pre	uint8_t	ブリドライバレジスタ CPSET2 前回値	Read 値
com_u1_pdrvreg_cpset2_now	uint8_t	ブリドライバレジスタ CPSET2 現在値	Write 値
com_u1_pdrvreg_cp_trim_pre	uint8_t	ブリドライバレジスタ CP_TRIM 前回値	Read 値
com_u1_pdrvreg_cp_trim_now	uint8_t	ブリドライバレジスタ CP_TRIM 現在値	Write 値
com_u1_pdrvreg_vreg5_trim_pre	uint8_t	ブリドライバレジスタ VREG5_TRIM 前回値	Read 値
com_u1_pdrvreg_vreg5_trim_now	uint8_t	ブリドライバレジスタ VREG5_TRIM 現在値	Write 値
com_u1_pdrvreg_csamp_trim_pre	uint8_t	ブリドライバレジスタ CSAMP_TRIM 前回値	Read 値
com_u1_pdrvreg_csamp_trim_now	uint8_t	ブリドライバレジスタ CSAMP_TRIM 現在値	Write 値
com_u1_pdrvreg_almraw1_pre	uint8_t	ブリドライバレジスタ ALMRAW1 前回値	Read 値 (ALMRAW1 Read Only)
com_u1_pdrvreg_toin_moni_pre	uint8_t	ブリドライバレジスタ TOIN_MONI 前回値	Read 値 (TOIN_MONI Read Only)
com_u1_pdrvreg_who_am_i_pre	uint8_t	ブリドライバレジスタ WHO_AM_I 前回値	Read 値 (WHO_AM_I Read Only)

com_u1_pdrvreg_trim_pt_pre	uint8_t	ブリドライバレジスタ TRIM_PT 前回値	Read 値
com_u1_pdrvreg_trim_pt_now	uint8_t	ブリドライバレジスタ TRIM_PT 現在値	Write 値
com_u1_pdrvreg_trim_en_pre	uint8_t	ブリドライバレジスタ TRIM_EN 前回値	Read 値
com_u1_pdrvreg_trim_en_now	uint8_t	ブリドライバレジスタ TRIM_EN 現在値	Write 値
com_u1_pdrvreg_bgr_trim_pre	uint8_t	ブリドライバレジスタ BGR_TRIM 前回値	Read 値
com_u1_pdrvreg_bgr_trim_now	uint8_t	ブリドライバレジスタ BGR_TRIM 現在値	Write 値
com_u1_pdrvreg_bfamp_trim_pre	uint8_t	ブリドライバレジスタ BFAMP_TRIM 前回値	Read 値
com_u1_pdrvreg_bfamp_trim_now	uint8_t	ブリドライバレジスタ BFAMP_TRIM 現在値	Write 値

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Dec.22.17	-	新規発行
1.01	Feb.28.19	-	RAJ306000_LESS_120_OPEN_V101 インターバルタイマのタイミングを 5[us]から 20[us]に修正 キャリア周波数両エッジ割り込みをアンダフローに変更 RSSK(TG-55L-KA)に合わせてパラメータを変更 誤記修正 Figure 2-5, 3-4, 5, 9, 12, 13, 14 Table 3-6: g_u2_fw_revision, g_u2_lib_revision Table 3-7: SPI_INTR_MODE, POLE_PAIR, etc.
1.02	Jun.05.19	-	RAJ306000_LESS_120_OPEN_*_*_V102 IDE: CS+ for CC, e ² studio に対応 Table 1-1, 2-10, 11 ICS 用ライブラリを変更 Table 2-10: ics インダクティブセンス・ライブラリを更新 Figure 3-11, Table 2-10, 12, 3-4, 6, 7 20[us] 割り込み処理(インターバルタイマ)を削除 Figure 3-11, Table 2-4, 9, 3-4 誤記修正 Figure 3-12, Table 3-6, etc.
1.03	Apr.08.20	-	RAJ306000_LESS_120_OPEN_*_*_V103 マクロ定義(HALL_SIG_MASK, WHO_AM_I_MASK)を追加 Figure 3-10, Table 3-7 誤記修正 Table 4-2: TRIM_PT, TRIM_EN, BGR_TRIM, BFAMP_TRIM, etc.

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えば、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。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まれます。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。
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、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なります。
 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。