RZ/T2H

永久磁石同期モータのエンコーダ利用ベクトル制御(9 軸) - アブソリュートエンコーダ

要旨

本書は、ルネサスエレクトロニクス製 MPU の RZ/T2H を搭載した Evaluation Board による 9 軸の永久磁石モータのエンコーダ利用ベクトル制御について説明します。本アプリケーションノート対象ソフトウェアはあくまで参考用途であり、弊社がこの動作を保証するものではありません。本アプリケーションノート対象ソフトウェアを使用する場合、適切な環境で十分な評価をしたうえで御使用ください。

動作確認デバイス

本アプリケーションノート対象ソフトウェアの動作確認は下記のデバイスで行っております。

・RZ/T2H (R9A09G077M44GBG)

目次

1. はじめに

1.1 概要

本ファームウェアはエンコーダを利用した永久磁石モータのベクトル制御を行います。

1.2 機能

本ファームウェアは主な機能として、以下を備えています。

- 永久磁石モータのエンコーダ利用ベクトル制御(9 軸)
- RZ/T2H Motion Control Utility との通信

2. ソフトウェア構成

2.1 基本仕様

本ファームウェアの基本仕様を表 2-1 に示します。

表 2-1 基本仕様

項目	説明
制御方式	ベクトル制御
ロータ位置検出	アブソリュートエンコーダ
入力電圧	DC 24 [V]
PWM 周波数	e2 studio：20 [kHz]
	EWARM：10 [kHz]
デッドタイム	1 [us]
電流制御周期	e2 studio：50 [us]
	EWARM：100 [us]
速度制御周期	100[us]
位置制御周期	100[us]
速度指令範囲	CW：0 [rpm] ～ 3000 [rpm]
	CCW：0 [rpm] ～ 3000 [rpm]
位置指令範囲	-2147483647 [ec] ～ 2147483647 [ec]
位置分解能	0.0055 [degree] (2^{16})
回転方向	CW 方向：位置 [ec] が減少 CCW 方向：位置 [ec] が増加
コンパイラ最適化設定	e2 studio：Optimize (-O1)
	EWARM：Low
保護機能	- POEG - インターロック処理

2.2 開発環境

ソフトウェアの開発環境を表 2-2、開発支援ツールを表 2-3、ハードウェア構成を表 **2-4** に示します。

表 2-2 開発環境

Integrated Development Environment (IDE)	e² studio	IAR Embedded Workbench for Arm
IDE version	2024-10	9.60.2 + patch (EWARM patch for RZ/T2H Rev.1.0)
FSP version	2.2.0	FSP Smart Configurator 2024-10
Toolchain version	GNU Arm Embedded 12.2.1.arm-12-24	-
In Circuit Emulator (ICE)	J-Link OB	IAR I-jet

表 2-3 開発支援ツール

Tool name	Tool Version
RZ/T2H Motion Control Utility	1.0.0.0

表 2-4 ハードウェア構成

Equipment		Model name
RZ/T2H Evaluation Board		RTK9RZT2H0CW1000BJ
	MPU	R9A09G077M44GBG 729-pin FCBGA, RAM 2[MB]
	On-board memory	OctaFlash: 64[MB]
	Operating frequency	Cortex-R52 CPU0: 1000[MHz] Cortex-R52 CPU1, Cortex-A55 は、未使用
	Operating voltage	DC 15[V]/3[A], 24[V]/3[A]
	Operating mode	xSPI0 boot mode (x1 boot serial flash)
Bus Board		RTK0EM0000Z03000BJ
Inverter Board		RTK0EM0000B15010BJ
	Operating voltage	DC 24[V]
Motor /Encoder (manufactured by TAMAGAWA SEIKI)		TSM3101N2001E020 /TS5669N124 (FA-CODER®)

2.3 ファイル構成

ファイル構成を表 2-5 に示します。

表 2-5 ファイル構成

ファイル	説明
rzt_cfg	FSP のコンフィグファイル
rzt_gen	FSP によって生成されたコード
rzt¥arm	CMSIS 関連ファイル
rzt¥board	ボード関連ファイル
rzt¥fsp	FSP 関連ファイル
cg_src¥r_cg_scifa.c	SCI ドライバ
cg_src¥r_cg_systeminit.c	周辺機能初期化処理
inc¥apl	src/apl 以下のソースファイルのヘッダファイル
src¥apl¥m_commands.c	コマンド判別処理
src¥apl¥m_commutation.c	ベクトル制御処理
src¥apl¥m_control.c	制御処理
src¥apl¥m_interlocks.c	インターロック処理
src¥apl¥m_interpreter.c	コマンド判別処理
src¥apl¥m_phasing.c	フェージング処理
src¥apl¥m_pid_calc.c	PID 演算処理
src¥apl¥m_pos_read.c	エンコーダ値取得処理
src¥apl¥m_recorder.c	データ記録処理
src¥drv¥dsm	DSMIF ドライバ
src¥drv¥m_rzt.c	ドライバ関連処理
src¥hal_entry.c	メイン処理
src¥encoder¥FACoder	FACoder ドライバ

3. ファームウェア詳細

3.1 初期化処理

初期化処理はデバイスのリセット後に実行されます。初期化処理の関数を**表 3-1** に示します。

表 3-1 初期化処理

関数	説明
R_Systeminit	R_Systeminit 関数は周辺機能の初期化を行います。初期化される周辺機能は以下の通りです。 ・GPT ・ELC ・DSMIF ・POEG ・SCI ・XSPI 入力：なし 出力：なし
m_startup	m_startup 関数は以下の初期化をおこないます。 ・t_motor 型の初期化 ・ボード上の LED の点灯 ・アブソリュートエンコーダの初期化 入力：なし 出力：なし
m_Restore	m_Restore 関数はフラッシュメモリからモータパラメータを読み出します。読み出したパラメータは t_motor 型構造体のメンバ変数に代入されます。 入力：(t_console *) pc、(t_motor *) pm 出力：なし

3.2 メイン処理

メイン処理は m_control.c の m_foreground 関数として定義されています。m_foreground 関数は hal_entry.c で while ループのなかで呼び出されます。メイン処理の関数を表 **3-2** に示します。

表 3-2 メイン処理

関数	説明
m_interpreter	m_interpreter 関数は RZ/T2H Motion Control Utility から受信したコマンドを実行します。 入力：(t_console *) pc 出力：なし
m_rec_begin	m_rec_begin 関数は RZ/T2H Motion Control Utility の Motion Scope のためのデータ記録を行います。 入力：なし 出力：なし
interlocks	Interlocks 関数はモータの位置偏差、速度、電流の値がリミット値を超えているときに PWM 出力を停止させます。 入力：(t_motor *) pm 出力：なし

3.3 周期割り込み処理

周期割り込み処理は主にエンコーダ利用ベクトル制御のための処理を行います。これらの処理は m_control.c の m_background 関数に実装されています。周期割り込み処理の関数を表 **3-3** に示します。

表 3-3 周期割り込み処理

関数	説明
bootstrap_charge	bootstrap_charge 関数は時間待ち処理と DSMIF のオフセット値を取得します。 入力：(t_motor *) pm 出力：なし
pos_read	pos_read 関数はアブソリュートエンコーダから位置を取得します。 pos_read 関数は m_pos_read.c ファイルに実装されています。 入力：(t_motor *) pm 出力：なし
pos_loop	pos_loop 関数は位置制御を行います。 pos_loop 関数は m_control.c ファイルに実装されています。 入力：(t_motor *) pm 出力：なし
vel_loop	vel_loop 関数は速度制御を行います。 vel_loop 関数は m_control.c ファイルに実装されています。 入力：(t_motor *) pm 出力：なし
crnt_read	crnt_read 関数は DSMIF の値を取得し、電流値への変換を行います。 crnt_read 関数は m_rzt.c ファイルに実装されています。 入力：(t_motor *) pm 出力：なし
crnt_loop	crnt_loop 関数は位置と電流値からベクトル制御を行います。 crnt_loop 関数は m_control.c ファイルに実装されています。 入力：(t_motor *) pm 出力：なし
m_recorder	m_recorder 関数は RZ/T2H Motion Control Utility の Motion Scope 機能のためのデータ取得を行います。 m_recorder 関数は m_recorder.c ファイルに実装されています。 入力：なし 出力：なし

3.4 通信処理

通信処理は RZ/T2H Motion Utility からのコマンドを処理します。通信処理を表 **3-4** に示す。

表 3-4 通信処理

関数	説明
m_rx_interrupt	m_rx_interrupt 関数は受信データフル割り込み発生時に呼び出されます。 受信データレジスタからデータを読みだし、t_console 型変数のメンバ変数を操作し、受信したコマンドのデコードを行います。 m_rx_interrupt 関数は m_rzt.c ファイルに実装されています。 入力：(t_console *) pc 出力：なし
m_tx_interrupt	m_tx_interrupt 関数は送信データエンプティ割り込み発生時に呼び出されます。送信データレジスタに t_console 型変数が持つデータを書き込みます。 m_tx_interrupt 関数は m_rzt.c ファイルに実装されています。 入力：(t_console *) pc 出力：なし

3.5 主要な構造体

主要な構造体を表 3-5 に示します。

表 3-5 主要な構造体

データ型	説明
t_motor	t_motor はモータの位置、速度、電流の指令値と現在値、ゲイン、リミット値、コンペアマッチレジスタのアドレス等を持っています。
t_motor_pars	t_motor_pars はフラッシュメモリへ保存されるパラメータを持っています。
t_console	t_console は RZ/T2H Motion Control Utility との通信に使用する変数、SCI のレジスタのアドレス等をもっています。
t_command	t_command は RZ/T2H Motion Control Utility から送信される ASCII コマンドと変数や関数を対応づけるためテーブルを持っています。
t_trace	t_trace は RZ/T2H Motion Control Utility の Motion Scope 機能に関する変数を持っています。

3.6 グローバル変数

主要なグローバル変数を表 3-6 に示します。

表 3-6 グローバル変数

グローバル変数	説明
t_motor g_st_m[MOTOR_NUM]	t_motor 型配列（要素数：MOTOR_NUM）
t_console con2	RZ/T2H Motion Control Utiity と通信を行うための t_console 型変数
long g_tick	g_tick は周期割り込み処理のなかでインクリメントします。リアルタイムタスクとメインループ間の処理の調整に使用します。
t_command Commands[]	RZ/T2H Motion Control Utility から送信される ASCII コマンドと変数や関数を対応づけるためテーブル

3.7 主要な列挙型

主要な列挙型を表 3-7 に示します。

表 3-7 主要な列挙型

列挙型	説明
ETYPE	サポートする各種エンコーダタイプを定義します。
CommutationModes	電流ループアルゴリズムの動作を定義します。
PacketCode	受信パケットのタイプを定義します。
PacketError	パケットプロトコルのインタプリタが通知する、起こり得るエラーを定義します。

4. モータ制御処理

4.1 9 軸のモータ制御処理

周期割り込み処理 m_background 関数はモータを制御するために、bootstrap_charge 関数、pos_read 関数、pos_loop 関数、vel_loop 関数、crnt_read 関数、crnt_loop 関数を MOTOR_NUM が指定する回数連続して実行します。MOTOR_NUM は制御するモータの個数を決定する define マクロです。MOTOR_NUM は m_common.h に定義されています。**図 4-1** に 9 軸のモータ制御処理のフローチャートを示します。

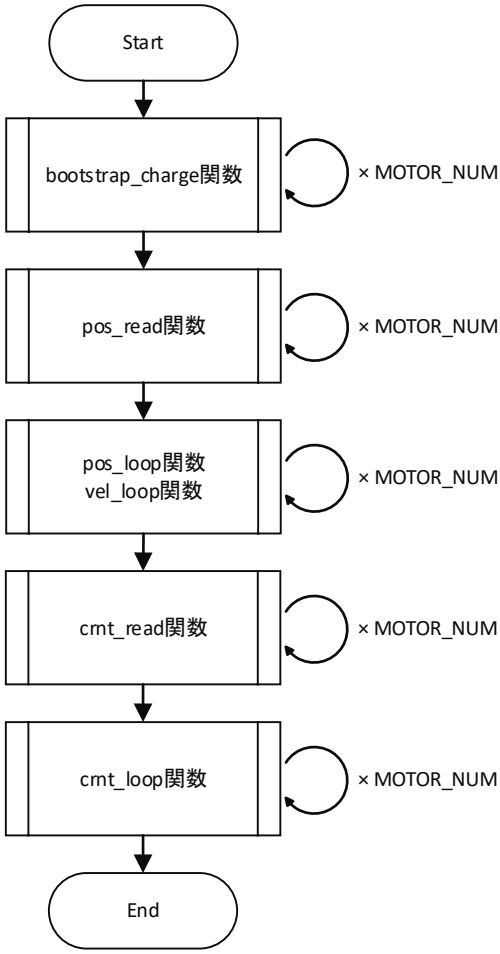

図 4-1 9 軸のモータ制御処理

4.2 m_background 関数

4.2.1 bootstrap_charge 関数

bootstrap_charge 関数は時間待ち処理と DSMIF のオフセット値を取得します。

4.2.2 pos_read 関数

pos_read 関数はアブソリュートエンコーダの位置を取得します。

pos_read 関数に関連する t_motor 型構造体のメンバ変数を表 4-1 に示します。

表 4-1 pos_read 関数に関連する t_motor 型構造体のメンバ変数

変数	説明
encoder_type	エンコーダの種類を指定します。 本ファームウェアはアブソリュートエンコーダとして FACODER を使用するため、encoder_type は ETYPE_APE_FACODER に初期化されています。

4.2.3 pos_loop 関数

pos_loop 関数は位置指令値(cmd_pos64)と現在位置(crnt_pos64)から位置偏差を計算し pid_calc_pos64 関数を呼び出し、速度制御器に入力する速度指令値(in_vel64)を計算します。

pos_loop 関数に関連する t_motor 型構造体のメンバ変数を表 4-2 に示します。

表 4-2 pos_loop 関数に関連する t_motor 型構造体のメンバ変数

変数	説明
crnt_kp	比例ゲイン(位置制御)
crnt_ki	積分ゲイン(位置制御)
crnt_kd	差動ゲイン(位置制御)
integral_limit	積分制限値(位置制御)
crnt_kvff	速度フィードフォワードゲイン(位置制御)
crnt_kaff	加速度フィードフォワードゲイン(位置制御)
crnt_bias	出力加算値(位置制御)
cmd_vel	指令速度
cmd_acc	指令加速度
pos_loop_limit	出力制限値(位置制御)
pos_error2	位置偏差
derivative_err2	位置偏差の微分値
integral_err2	位置偏差の積分値

Motion Control Utility の ASCII 通信プロトコルコマンドの KP、KI、KD、VFF、AFF コマンドは buff_kp、buff_ki、buff_kd、buff_kvff、buff_kaff を変更します。update_ctrl 関数が実行されたときに、これらの変数の値が crnt_kp、crnt_ki、crnt_kd、crnt_kvff、crnt_kaff に代入されます。

4.2.4 vel_loop 関数

vel_loop 関数は速度指令値(in_vel64)と現在速度(crnt_vel64)から速度偏差を計算し pid_calc_vel64 関数を呼び出し、電流制御器に入力する q 軸電流指令値(output_q)を計算します。

vel_loop 関数に関連する t_motor 型構造体のメンバ変数を表 4-3 に示します。

表 4-3 vel_loop 関数に関連する t_motor 型構造体のメンバ変数

変数	説明
crnt_kp_vel	比例ゲイン(速度制御)
crnt_ki_vel	積分ゲイン(速度制御)
crnt_kd_vel	微分ゲイン(速度制御)
integral_limit_vel	積分制限値(速度制御)
PiOut_limit_vel	出力制限値(速度制御)
vel_error2	速度偏差
derivative_err2_vel	速度偏差の微分値
integral_err2_vel	速度偏差の積分値
output_q	q 軸電流指令値 [mA]

Motion Control Utility の ASCII 通信プロトコルコマンドの VKP、VKI、VKD コマンドは buff_kp_vel、buff_ki_vel、buff_kd_vel を変更します。update_ctrl 関数が実行されたときに、これらの変数の値が crnt_kp、crnt_ki、crnt_kd に代入されます。

4.2.5 crnt_read 関数

crnt_read 関数は DSMIF の変換値を取得し、変換値を U 相電流、V 相電流、W 相電流に変換します。

crnt_read 関数に関連する t_motor 型構造体のメンバ変数を表 4-4 に示します。

表 4-4 crnt_read 関数に関連する t_motor 型構造体のメンバ変数

変数	説明
adc1_raw	各ユニットの DSMIF CH0 の変換値
adc2_raw	各ユニットの DSMIF CH1 の変換値
adc3_raw	各ユニットの DSMIF CH2 の変換値
adc_iu	U 相電流 [mA]
adc_iv	V 相電流 [mA]
adc_iw	W 相電流 [mA]
crnt_volt	バスボード電圧 [V]
total_current	U 相電流と V 相電流の絶対値の和 [mA]

4.2.6 crnt_loop 関数

crnt_loop 関数は次の処理を行います。

1. 電気角演算
2. commutate_foc 関数
3. commutate_svm 関数
4. update_pwm 関数

上記の処理に加えて t_motor 型構造体のメンバ変数 aligning が 1 のときにフェージング処理を行います。

4.2.6.1 フェージング処理

t_motor 型構造体のメンバ変数 aligning が 1 にときに、forced_phasing 関数を呼び出し、フェージング処理を行います。フェージング処理は U 相から V 相とW相へ電流を流し、ロータの d 軸を u 軸に合わせます。

フェージング処理に関連する t_motor 型構造体のメンバ変数を表 4-5 に示します。

表 4-5 フェージング処理に関連する t_motor 型構造体のメンバ変数

変数	説明
aligning	フェージング処理実行のフラグ
phasing_mode_crnt	フェージング処理を選択するための変数 forced_phasing 関数を実行するため phasing_mode_crnt は PIM_FORCED(= 0)に初期化されています。
phasing_time	forced_phasing 関数を実行する時間
phasing_power	forced_phasing 関数を実行したときのデューティー比
phasing_origin	forced_phasing 関数実行後のエンコーダ値を格納する変数

4.2.6.2 電気角演算

電気角(phase_angle)を現在位置(crnt_pos)とロータの d 軸が u 軸と一致する位置(phase_origin)から計算し angle_rad へ変換します。angle_rad から sin と cos の値を計算します。

電気角演算に関連する t_motor 型構造体のメンバ変数を表 4-6 に示します。

表 4-6 電気角演算に関連する t_motor 型構造体のメンバ変数

変数	説明
phase_angle	電気角 [ec]
angle_rad	ロータの位置 [rad]
counts2rad	電気角 [ec]をラジアンに変換するための係数
angle_sin	angle_rad から求めた sin 関数の値
angle_cos	angle_rad から求めた cos 関数の値

4.2.6.3 commutate_foc 関数

commutate_foc 関数は U 相電流と V 相電流をα軸電流とβ軸電流へ変換、αβ軸電流を d 軸電流と q 軸電流へ変換します。d 軸電流と q 軸電流の PI 演算をおこない d 軸電圧と q 軸電圧を得ます。d 軸電圧と q 軸電圧をαβ軸電圧に変換します。

commutate_foc 関数に関連する t_motor 型構造体のメンバ変数を表 4-7 に示します。

表 4-7 commutate_foc 関数に関連する t_motor 型構造体のメンバ変数

変数	説明
p_iu	U 相電流を保持する変数 adc_iu へのポインタ
p_iv	V 相電流を保持する変数 adc_iv へのポインタ
foc_id	d 軸電流 [mA]
foc_iq	q 軸電流 [mA]
foc_id_err	d 軸電流偏差 [mA]
foc_iq_err	q 軸電流偏差 [mA]
foc_id_err_int	d 軸電流偏差の積分値 [mA]
foc_iq_err_int	q 軸電流偏差の積分値 [mA]
output_d	d 軸電流指令値 [mA]
output_q	q 軸電流指令値 [mA]
foc_kp	比例ゲイン(電流制御)
foc_ki	積分ゲイン(電流制御)
foc_vd	d 軸電圧
foc_vq	q 軸電圧
foc_alpha	α軸電圧
foc_beta	β軸電圧

4.2.6.4 commutate_svm 関数

commutate_svm 関数はα軸電圧、β軸電圧を 3 相電圧に変換したのちに空間ベクトル変調を行います。

4.2.6.5 update_pwm 関数

update_pwm 関数は U 相、V 相、W 相の PWM 信号のデューティー比を更新します。

5. インターロック処理

インターロック処理はモータ制御変数を監視し、条件を満たしたときに PWM 出力を停止します。インターロック処理はメイン処理として 1ms ごとに実行されます。インターロック処理の各機能と処理内容を表 5-1 に示します。

表 5-1 インターロック処理

機能	処理
電流超過検出	総電流(total_current)が一定時間(tc_limit_time)以上連続して過電流閾値(tc_limit)を超過したときにステータス(ErrSts)の 26 ビットをセットする。 tc_limit の初期値：TC_LIMIT__VAL_DFLT tc_limit_timer の初期値：TC_LIMIT_TIME_VAL_DFLT
位置誤差超過検出	位置誤差(pos_error)が一定時間(pos_error_timer)以上連続して位置誤差閾値(pos_error_limit)を超過したときにステータス(ErrSts)の 17 ビットをセットする。 pos_error_limit の初期値：POS_ERROR_LIMIT_VAL_DFLT pos_error_time の初期値：POS_ERROR_LIMIT_TIME_VAL_DFLT
バスボード低電圧検出	バスボード電圧(crnt_volt)が低電圧閾値(Lvolt_Val)を下回ったときにステータス(ErrSts)の 28 ビットをセットする。 Lvolt_Val の初期値：LVOLT_VAL_DFLT
バスボード過電圧検出	バスボード電圧(crnt_volt)が過電圧閾値(Hvolt_Val)を上回ったときにステータス(ErrSts)の 27 ビットをセットする。 Hvolt_Val の初期値：HVOLT_VAL_DFLT
インバータ過電流検出(POEG)	POEG0 の POEG0GD0 の PIDF ビットが 1 になったときにステータス(ErrSts)の 25 ビットをセットする。
負荷超過の事前検出	総電流(total_current)が閾値(Ovc_Val)以上になったときにステータス(ErrSts)の 21 ビットをセットする。 Ovc_Val の初期値：OVC_VAL_DFLT
過速度検出	速度(crnt_vel)が閾値(Ovs_Val)以上になったときにステータス(ErrSts)の 20 ビットをセットする。 Ovs_Val の初期値：OVS_VAL_DFLT
速度誤差超過検出	速度誤差(vel_error)が 5 秒間連続して閾値(WOvs_Val)以上になったときにステータス(ErrSts)の 19 ビットをセットする。 WOvs_Val の初期値：WOVS_VAL_DFLT
位置超過検出	位置が位置上限閾値(WPosMax_Val)以上になったときにステータス(ErrSts)の 15 ビットをセットする。 位置が位置下限閾値(WPosMin_Val)以下になったときにステータス(ErrSts)の 13 ビットをセットする。 WPosMax_Val の初期値：WPOSMAX_VAL_DFLT WPosMin_Val の初期値：WPOSMIN_VAL_DFLT 位置超過処理はデフォルトでは無効化されています。

6. データ記録

データ記録機能により、システム動作のリアルタイムでの分析が可能になります。この機能は、アプリケーション固有のコンテキストで、各種制御ループの性能、設定パラメータおよび効率性を分析するために必要不可欠です。データレコーダは、システム稼働中に、ユーザ定義のパラメータを最大 4 個バッファに格納します。サポート関数と設定パラメータにより、記録レートや、記録する変数、記録開始時間および終了予定時間が選択可能です。

記録期間は、サンプル数としてマクロ TRACE_BUFFER_SIZE で定義されています。デフォルトでは 512 に設定されています。この値は、アプリケーションがより長い記録を必要とし、RAM メモリが使用可能な場合に増加できます。全バッファは traceData[]という名前の単一の配列に結合されますが、配列のデータ型は短く、16 ビット整数です。このため、32 ビット変数が記録されるとき、その値は 1 番目と 4 番目のバッファ間で分割されます。データが通知されると、値は適宜結合されます。

ホストは RVAL コマンドを使用してデータを取得します。表 6-1 に RVAL コマンドのコードと取得するデータ変数の対応を示します。

表 6-1 データ変数とコードの対応

コード	データ変数
0	crnt_pos
1	crnt_vel
2	crnt_acc
3	I2t_integral
4 - 7	Reserved
8	pos_error
9	output_q
10	Reserved
11	foc_id
12	foc_iq
13	foc_id_err
14	foc_iq_err
15	adc1_raw
16	adc2_raw
17	pvt_points
18	foc_vd
19	foc_vq
20	g_counter
21	phase_angle
22	adc3_raw
23	captured_pos
24	pos_error2
25	Integral_err2
26	vel_error2
27	Integral_err2_vel
28	foc_id_err_int

29	foc_iq_err_int
30	est_trq
31	angle_rad

データレコーダの開始・停止条件は、ASCII コマンド TRACE（変数 trace.Trigger）で設定します。レコーダの各トリガ条件を表す使用可能な設定を**表 6-2** に示します。

表 6-2 各トリガ条件

TRACE コード	開始トリガ条件	停止トリガ条件
0	N/A	データ記録を停止
1	即時開始	動作完了時に停止。このトリガは動作終了を確認するのに有効。
2	即時開始	バッファフル時に停止。このトリガは動作開始を確認するのに有効
3	動作開始時に開始	動作終了時に停止
4	即時開始	入力変化時に停止。入力ビットマスクは trace.Level 変数で定義
5	即時開始	trace.Level 変数で定義した閾値を超過した時
6	即時開始	trace.Level 変数で定義した閾値を下回った時
7	PWM 出力変化時に開始	バッファフル時に停止
8	入力変化時に開始。入力マスクは trace.Level 変数で定義	バッファフル時に停止
9	trace.Level 変数で定義した閾値を超過した時	バッファフル時に停止
10	trace.Level 変数で定義した閾値を下回った時	バッファフル時に停止

レコーダは、50us 間隔で実行されるリアルタイムタスクに同期して動作します。記録レートはこの時間間隔の倍数で表されます。乗数は ASCII コマンド TRATE で設定され、変数 trace.RateMult に格納されます。たとえば、記録レートを 1ms にしたい場合、TRATE は 20 に設定されている必要があります。

トリガ条件発生時に評価されるもう一つの変数は、ASCII コマンド TLEVEL（変数 trace.Level）です。この変数の値は、記録された変数と比較する閾値となります。トリガのコードによっては、記録開始条件で、値が閾値より大きいかまたは小さいかをテストすることできます。

開始トリガ条件をテストするために定期的に呼び出される関数は m_rec_begin()です。

停止トリガ条件をテストし、レコーディングを実施するために定期的に呼び出される関数は m_recorder()です。レコーダの動作モードは、ASCII コマンド TMODE（変数 trace.Mode）で通知します。格納されているコードの意味を**表 6-3** に示します。

表 6-3 TMODE コード

TMODE コード	動作モード／ステータス
0	アイドル状態。記録を開始した場合は停止。
1	レコーダが作動可能：動作開始時に記録を開始可能
2	記録中。バッファフルで記録停止。
3	循環バッファに記録中。動作完了で記録停止。

データ記録が完了すると、ホストはコマンド PLAY を使用して記録バッファの内容を取得できます。この要求を実装する関数は m_Play()です。この関数は、起動コンテキストがパケットコマンドハンドラである場合は、バイナリパケット応答の設定も行います。

7. ASCII 通信プロトコル

ASCII プロトコルは、復帰文字（CR、ASCII 13）で終わる ASCII 文字列で構成されるコマンドをベースとしています。コントローラはプロンプト直後のオプションデータ列を返します。

ASCII プロトコルは、以下の通信パラメータを使用します。

115,200 bps、8 データビット、1 ストップビット、パリティ無し

コマンドを受け付けると、コマンドプロンプトには CR、改行文字（LF、ASCII 10）、大なり記号（>）が表示されます。コマンドが拒否されると、コマンドプロンプトには > の代わりに疑問符（?）が表示されます。

例）

```
POS          ; CR で終わるホストコマンド
120          ; 応答データ列
>            ; 応答プロンプト
```

コマンドプロンプトに変数名を入力すると、参照先の変数の値が返されます。変数名の後にパラメータを続けて入力すると、その変数を新しい値に設定するリクエストとして認識されます。いくつかの変数は読み取り専用となっていますので、それらの変数に値を設定しようとすると、無効なコマンドとして返されます。

例）

```
>POS              ; 変数 POS の値をリクエスト
2100
>POS 2000    ; POS を新しい値に設定
>POS              ; 新しい値を通知
2000
>
```

ASCII 通信プロトコルのコマンドの一覧を**表 7-1** に示します。

表 7-1 ASCII 通信プロトコルコマンド

コマンド名	サイズ	R/W	説明
STA	2H	R	ステータスワード short act_state
ERR	2	R	位置誤差 [ec] short pos_error
ADC1	2	R	U 相電流 [A/D 変換値] short adc1_raw
ADC2	2	R	V 相電流 [A/D 変換値] short adc2_raw
TC	2U	R	総電流(U, V 相電流の絶対値の和) [mA] unsigned short total_current
CV	4	R	回転速度 [rpm] long crnt_vel

コマンド名	サイズ	R/W	説明
VEL	4	R/W	目標回転速度 [ec/cycle time] long buffMotion.velocity
ACC	4	R/W	目標加速度 [ec/cycle time^2] long buffMotion.acceleration
DEC	4	R/W	目標減速度 [ec/cycle time^2] long buffMotion.deceleration
PRO	2U	R/W	速度プロファイルモード(0 固定) short dflt_vgp_mode
KP	2U	R/W	比例ゲイン(位置制御) short buff_kp
KI	2U	R/W	積分ゲイン(位置制御) short buff_ki
KD	2U	R/W	微分ゲイン(位置制御) short buff_kd
IL	2U	R/W	積分制限値(位置制御) short integral_limit16
VFF	2U	R/W	速度フィードフォワードゲイン(位置制御) short buff_kvff
AFF	2U	R/W	加速度フィードフォワードゲイン(位置制御) short buff_kaff
MAX	2U	R/W	位置誤差検出閾値[ec] short buff_err_limit
ETIME	2U	R/W	位置誤差検出指定時間(連続検出回数) short pos_error_time
DS	2U	R/W	位置・速度制御間隔(2 固定) short crnt_ds
MLIMIT	2U	R/W	出力制限値(位置制御) long pos_loop_limit
BIAS	2	R/W	出力加算値(位置制御) short crnt_bias
ASTOP	2U	R/W	自動停止モード(0 固定) short auto_stop_mode
PIMODE	2U	R/W	フェージングモード(0 固定) short phasing_mode
PITIME	2U	R/W	フェージング期間 short phasing_time
PIOUT	2U	R/W	フェージング PWM 出力 short phasing_power
PMAP	2U	R/W	PWM 出力相選択(0 固定) short phase_config
PORIGIN	4	R	位相原点[ec] long phase_origin
PCMODE	2U	R/W	位相整流モード(4 固定) short *commutation_mode*
PPAIRS	2U	R/W	極対数(5 固定) short pole_pairs
PCOUNTS	4	R/W	電気角 1 回転辺りのエンコーダカウント[ec] long ec_per_ecycle
ECPR	4	R/W	機械角 1 回転辺りのエンコーダカウント[ec] long ec_per_rev

コマンド名	サイズ	R/W	説明
PANGLE	2	R	磁束位置[ec] short phase_angle
CLIMIT	2U	R/W (Func)	過電流検出閾値[mA] m_CurrentLimit() unsigned short tc_limit
CTIME	2U	R/W	過電流検出指定時間(連続検出回数) unsigned short tc_limit_time
IDM	2	R	d 軸電流[mA] short foc_id
IQM	2	R	q 軸電流[mA] short foc_iq
IQERR	2	R	q 軸電流誤差[mA] short foc_iq_err
QKP	2U	R/W	比例ゲイン(電流制御) short foc_kp
QKI	2U	R/W	積分ゲイン(電流制御) short foc_ki
ECP	4	R	位置誤差が指定時間連続して位置誤差検出閾値を超過したときの位置指令値[ec] long mecmd_pos
ECV	4	R	位置誤差が指定時間連続して位置誤差検出閾値を超過したときの回転速度指令値 long mecmd_vel
EPO	4	R	位置誤差が指定時間連続して位置誤差検出閾値を超過したときの現在位置[ec] long mecrnt_pos
U	2	R/W	U 相 PWM 出力 unsigned short PhaseU
V	2	R/W	V 相 PWM 出力 unsigned short PhaseV
W	2	R/W	W 相 PWM 出力 unsigned short PhaseW
PHASES	2U	R/W	モータタイプ(3 固定) short motor_type
TMODE	2U	R/W	データ記録動作モード short trace.Mode
TRATE	2U	R/W	データ記録レート(50[us]間隔) short trace.RateMult
TLEVEL	4	R/W	データ記録閾値 float trace.Level
ABS	4	R/W (Func)	絶対目標位置[ec] m_Abs() long buffMotion.position
REL	4	R/W (Func)	相対目標位置[ec] m_Rel() long buffMotion.position_rel
POS	4	R/W (Func)	現在位置[ec] m_Position() volatile long crnt_pos

コマンド名	サイズ	R/W	説明
IND	4	R (Func)	インデックス位置[ec] m_Index() volatile long index_pos
GO	4	W (Func)	目標位置まで回転する。 m_Go() long buffMotion.position
FWD	0	W (Func)	PWM コマンドの設定に応じて正回転する。 m_Forward()
REV	0	W (Func)	PWM コマンドの設定に応じて逆回転する。 m_Reverse()
ON	0	W (Func)	サーボ制御を有効にする。 m_ServoOn()
OFF	0	W (Func)	サーボ制御を無効にする。 m_ServoOff()
ENABLE	0	W (Func)	PWM 出力を有効にする。 m_PowerOn()
DISABLE	0	W (Func)	PWM 出力を無効にする。 m_PowerOff()
STOP	0	W (Func)	回転を目標減速度で停止する。 m_SmoothStop()
ABORT	0	W (Func)	回転を最大減速度で停止する。 m_AbruptStop()
ALIGN	0	W (Func)	フェージングを実行する。 m_AlignPhase()
VER	string	R (Func)	ファームウェアバージョン m_Version() const char *s_version
PWM	2	R/W (Func)	PWM 出力(目標 q 軸電流[mA]) m_PosLoopCmd() short pos_loop_cmd
IQCMD	2	R/W (Func)	目標 q 軸電流[mA] m_OutputIQ() short output_q
IDCMD	2	R/W (Func)	目標 d 軸電流[mA] m_OutputID() short output_d
CH1	2	R/W (Func)	記録するデータ選択(チャンネル 1) m_LogChannel0()
CH2	2	R/W (Func)	記録するデータ選択(チャンネル 2) m_LogChannel1()
CH3	2	R/W (Func)	記録するデータ選択(チャンネル 3) m_LogChannel2()
CH4	2	R/W (Func)	記録するデータ選択(チャンネル 4) m_LogChannel3()
TRACE	2	R/W (Func)	データ記録の開始/停止条件 m_Trace() short trace.Trigger
PLAY	4 x 4	R (Func)	記録したデータを取得する。 m_Play()

コマンド名	サイズ	R/W	説明
PINVERT	2	R/W (Func)	現在位置の符号反転(0 固定) m_PosInvert() short pos_inv_mode
SAVE	0	W (Func)	モータパラメータをフラッシュメモリに保存する。 ※ EWARM 版は、未サポート m_Save()
RESTORE	0	W (Func)	モータパラメータをフラッシュメモリから復元する。 ※ EWARM 版は、未サポート m_Restore()
ETYPE	2	R/W (Func)	エンコーダタイプ(3 固定) m_EncoderType() short encoder_type
EBAUDRATE	4	R/W (Func)	エンコーダ通信ボーレート(2500[kHz]固定) m_EncBaudrate() long enc_baudrate
ESTATUS	2H	R (Func)	エンコーダステータス m_EncStatus() unsigned short enc_status
TBSIZE	2U	R	データ記録サイズ short trace.buff_size
QKD	2U	R/W	微分ゲイン(電流制御) short foc_kd
VKP	2U	R/W	比例ゲイン(速度制御) short buff_kp_vel
VKI	2U	R/W	積分ゲイン(速度制御) short buff_ki_vel
VKD	2U	R/W	微分ゲイン(速度制御) short buff_kd_vel
ELVOLT	4	R/W (Func)	低電圧検出閾値[V] m_Elvolt() long Lvolt_Val
EHVOLT	4	R/W (Func)	過電圧検出閾値[V] m_Ehvolt() long Hvolt_Val
EWPOSMIN	4	R/W (Func)	位置異常検出閾値(下限) m_EwposMin() long WPosMin_Val
EWPOSMAX	4	R/W (Func)	位置異常検出閾値(上限) m_EwposMax() long WPosMax_Val
EOVS	4	R/W (Func)	過速度検出閾値[rpm] m_Eovs() long Ovs_Val
EWOVS	4	R/W (Func)	指示速度差検出閾値[ec/cycle time] m_Ewovs() long WOvs_Val
ERRMASK	4H	R/W (Func)	異常状態マスク m_Emask() unsigned long ErrMsk

コマンド名	サイズ	R/W	説明
EVOLT	2	R	バスボード母線電圧[V] long crnt_volt
EQUERY	4H	R	異常状態 unsigned long ErrSts
ERESET	0	W (Func)	異常状態をリセット(0)する。 m_Ereset() unsigned long ErrSts
EOVC	4	R/W (Func)	過負荷の事前検出閾値[mA] m_Eovc() long Ovc_Val
CTRLMODE	2U	R/W (Func)	制御モード 0: 位置制御 1: 速度制御 m_CtrlMode() unsigned short ctrl_mode
COMDIR	2U	R/W (Func)	回転方向指令値(CW/CCW) m_CommandDirection() unsigned short cmd_dir
COMVEL	2	R/W (Func)	回転速度指令値[rpm] m_CommandVelocity() short cmd_vel_rpm
RVAL	2, 4, 8, 4H	R/W (Func)	記録したデータを取得する。 m_ReadValue()

8. 関連ドキュメント

・RZ/T2H スタートアップマニュアル(RZ/T2H Motion Control Utility 編) (R01AN7334)

・RZ/T2H シリアルフラッシュメモリへのプログラム書き込みガイド (R01AN7335)

・e^2 studio 統合開発環境 ユーザーズマニュアル 入門ガイド RZ ファミリ (R20UT4535)

改訂履歴

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2024.11.26	-	初版作成

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 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。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 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。
14. 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.5.0-1　2020.10)

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