

要旨

本アプリケーションノートでは、RZ/T1 グループ MCU を搭載した RSK RZ/T1 評価ボード対応の TPUa (Timer Pulse Unit)、PPG (Programable Pulse Generator) サンプルプログラムについて説明します。

6 本の出力端子に対して、10ms 周期毎に出力信号を切り替えることによりパルス波形を出力します。

- TPUa チャンネル 0 使用
- PPG ユニット 1 (PO23-28 端子) 使用
- TPU0.TGRA のコンペアマッチ割り込み (TGI0A) 使用
- TPU0.TCNT のオーバフロー割り込み (TGI0V) 使用
- PCLKD/64 (PCLKD = 75MHz) 使用
- タイマ動作 (通常モード、TGRA コンペアマッチ動作、PPG トリガ使用) 使用
- FIT 仕様対応 API

対象デバイス

RZ/T1

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	4
2.	動作環境	5
3.	関連アプリケーションノート	6
4.	周辺機能説明	7
5.	ハードウェア説明	8
5.1	ハードウェア構成例	8
5.2	使用端子一覧	8
6.	ソフトウェア説明	9
6.1	動作概要	9
6.1.1	プロジェクト設定	10
6.1.2	使用準備	10
6.2	メモリマップ	11
6.2.1	サンプルプログラムのセクション配置	11
6.2.2	MPU の設定	11
6.2.3	例外処理ベクタテーブル	11
6.3	使用割り込み一覧	11
6.4	固定幅整数一覧	11
6.5	定数／エラーコード一覧	12
6.6	構造体／共用体／列挙型一覧	16
6.7	大域変数一覧	19
6.8	関数一覧	20
6.9	関数仕様	21
6.9.1	main	21
6.9.2	tpu0_a_cmpmatch_callback	23
6.9.3	tpu0_v_callback	24
6.9.4	R_TPUA_Open	24
6.9.5	R_TPUA_Control	26
6.9.6	R_TPUA_Close	27
6.9.7	R_TPUA_GetVersion	27
6.9.8	R_PPG_Open	28
6.9.9	R_PPG_Control	29
6.9.10	R_PPG_Close	30
6.9.11	R_PPG_GetVersion	30
6.10	フローチャート	31
6.10.1	main 処理	31
6.10.2	tpu0_a_cmpmatch_callback 処理	32
6.10.3	tpu0_v_callback 処理	32
6.10.4	R_TPUA_Open 処理	33
6.10.5	R_TPUA_Control 処理	34
6.10.6	R_TPUA_Close 処理	36

6.10.7	R_TPUA_GetVersion 処理	36
6.10.8	R_PPG_Open 処理	37
6.10.9	R_PPG_Control 処理.....	38
6.10.10	R_PPG_Close 処理	39
6.10.11	R_PPG_GetVersion 処理	39
6.11	R_TPUA_Control コマンド一覧	40
6.11.1	TPUA_CMD_TIMER_START	41
6.11.2	TPUA_CMD_TIMER_STOP	41
6.11.3	TPUA_CMD_TIMER_SYNC.....	41
6.11.4	TPUA_CMD_TIMER_ASYNC	42
6.11.5	TPUA_CMD_REG_READ	43
6.11.6	TPUA_CMD_REG_WRITE	45
6.11.7	TPUA_CMD_INTR_A_ENABLE.....	46
6.11.8	TPUA_CMD_INTR_A_DISABLE.....	47
6.11.9	TPUA_CMD_INTR_B_ENABLE.....	47
6.11.10	TPUA_CMD_INTR_B_DISABLE.....	47
6.11.11	TPUA_CMD_INTR_C_ENABLE.....	48
6.11.12	TPUA_CMD_INTR_C_DISABLE.....	48
6.11.13	TPUA_CMD_INTR_D_ENABLE.....	48
6.11.14	TPUA_CMD_INTR_D_DISABLE.....	49
6.11.15	TPUA_CMD_INTR_V_ENABLE.....	49
6.11.16	TPUA_CMD_INTR_V_DISABLE.....	49
6.11.17	TPUA_CMD_INTR_U_ENABLE.....	50
6.11.18	TPUA_CMD_INTR_U_DISABLE.....	50
6.12	R_PPG_Control コマンド一覧	51
6.12.1	PPG_CMD_REG_READ	52
6.12.2	PPG_CMD_REG_WRITE	53
7.	サンプルコード.....	54
8.	参考ドキュメント	55

1. 仕様

周辺機能と動作環境を以下に示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
RZ/T1内蔵 16ビットタイマパルスユニット (TPUa)	コンペアマッチによるタイマ制御 (チャネル0)
RZ/T1内蔵プログラマブルパルスジェネレータ (PPG)	TPUaに同期した出力端子制御 (ユニット1)
RZ/T1内蔵 I/Oポート RZ/T1内蔵マルチファンクションピンコントローラ (MPC)	PO23 (PS6) (端子番号: R19) PO24 (PS7) (端子番号: R20) PO25 (PT0) (端子番号: P19) PO26 (PT1) (端子番号: P20) PO27 (PT2) (端子番号: N19) PO28 (PT3) (端子番号: N20)
RZ/T1内蔵割り込みコントローラ (ICUA)	割り込み制御 (TPUa TPU0) <ul style="list-style-type: none"> コンペアマッチA割り込み要因 (TGI0A) ベクタ番号216 オーバフロー割り込み要因 (TGI0V) ベクタ番号220
RZ/T1内蔵クロック発生回路	TPUa/PPGへのクロック供給制御 (PCLKD:75MHz)
RZ/T1内蔵消費電力低減機能	TPUaの消費電力低減 (TPUaユニット0) PPGの消費電力低減 (PPGユニット1)

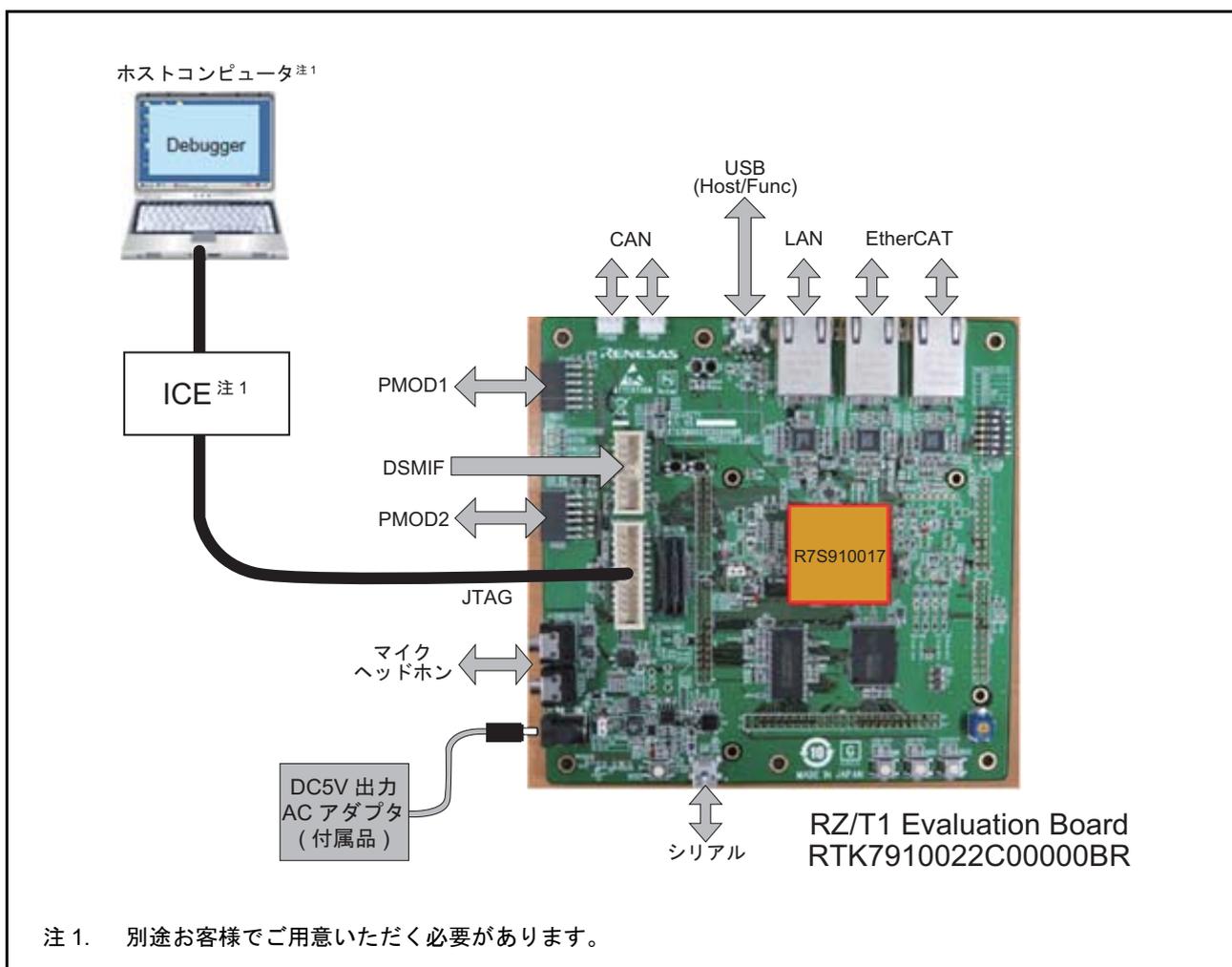


図 1.1 動作環境

2. 動作環境

本アプリケーションノートのサンプルコードは、下記の環境を想定しています。

表2.1 動作環境

項目	内容
使用マイコン	RZ/T1グループ
動作周波数	CPUCLK = 450MHz
動作電圧	3.3V
統合開発環境	IARシステムズ製 Embedded Workbench® for Arm Version 8.20.2 Arm製 DS-5™ 5.26.2 RENESAS製 e2studio 6.1.0
動作モード	SPIブートモード 16ビットバスブートモード
使用ボード	RZ/T1 Evaluation Board (RTK7910022C00000BR)
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none">NORフラッシュメモリ (CS0、CS1空間に接続) メーカー名: Macronix International Co., 型名: MX29GL512FLT2I-10QSDRAM (CS2、CS3空間に接続) メーカー名: Integrated Silicon Solution Inc、型名: IS42S16320D-7TLシリアルフラッシュメモリ メーカー名: Macronix International Co., 型名: MX25L51245G

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/T1 グループ 初期設定 アプリケーションノート (R01AN2554JJ)

注. 本アプリケーションノートで記載しないレジスタに関しては、RZ/T1 グループ 初期設定 アプリケーションノートで設定した値のまま使用します。

4. 周辺機能説明

動作モード、16ビットタイマパルスユニット (TPUa)、プログラマブルパルスジェネレータ (PPG)、I/Oポート、マルチファンクションピンコントローラ (MPC)、割り込みコントローラ (ICUA)、クロック発生回路、消費電力低減機能についての基本的な内容は、RZ/T1グループ・ユーザーズマニュアルハードウェア編を参照してください。

5. ハードウェア説明

5.1 ハードウェア構成例

以下に、TPUa/PPG ハードウェア構成例を示します。(オレンジ枠の PT3 ~ PT0、PS7、PS6 の 6 端子を使用)

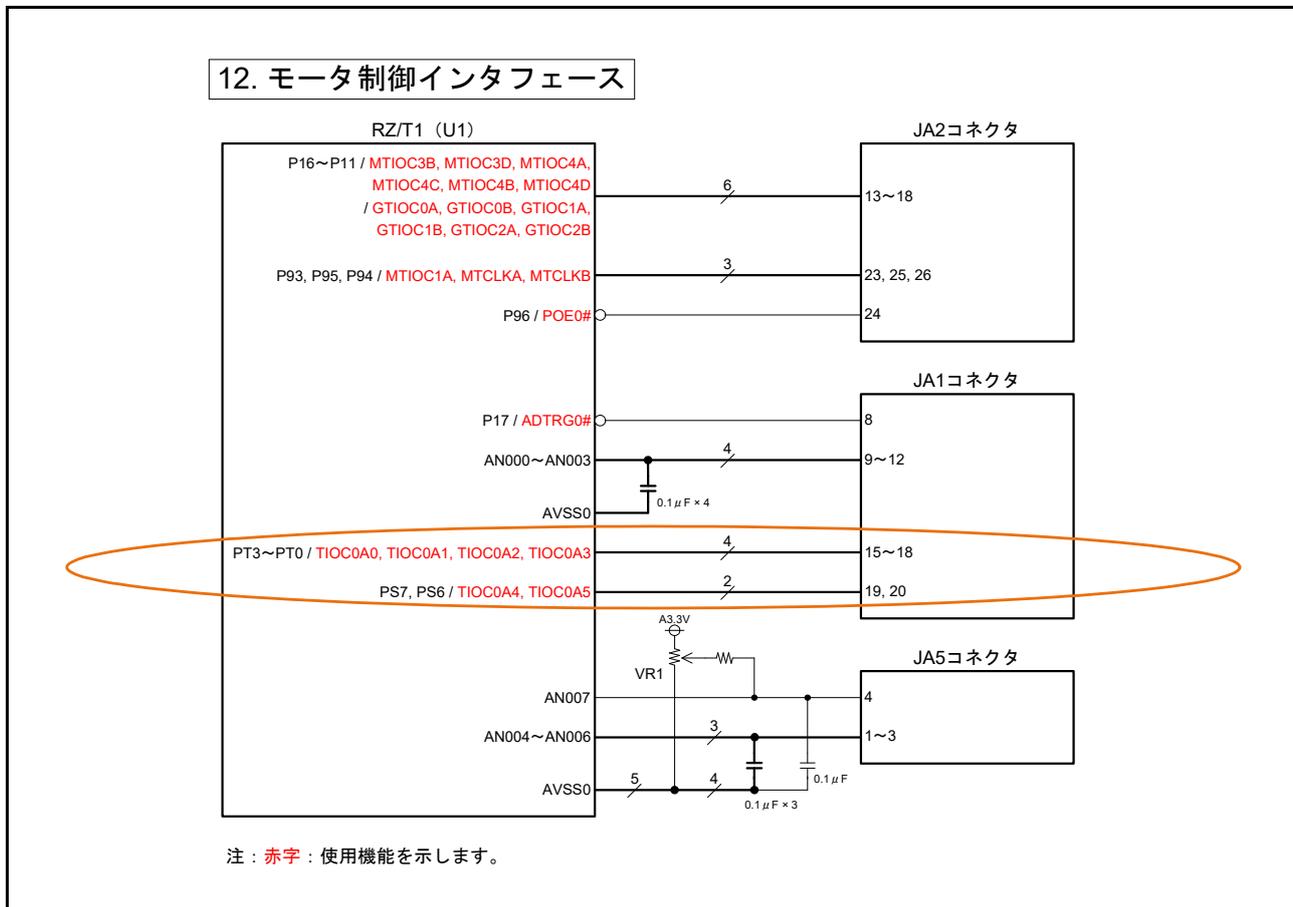


図 5.1 TPUa/PPG ハードウェア構成例

5.2 使用端子一覧

使用端子一覧を以下にを示します。

表 5.1 使用端子と機能

端子名	入出力	内容
PO23 (PS6:R19)	出力	PPGユニット1出力信号
PO24 (PS7:R20)	出力	PPGユニット1出力信号
PO25 (PT0:P19)	出力	PPGユニット1出力信号
PO26 (PT1:P20)	出力	PPGユニット1出力信号
PO27 (PT2:N19)	出力	PPGユニット1出力信号
PO28 (PT3:N20)	出力	PPGユニット1出力信号

6. ソフトウェア説明

6.1 動作概要

TPUa&PPG サンプルプログラムの機能概要を表 6.1 動作概要に示します。また、図 6.1 にシステムブロックを示します。

表6.1 動作概要 (1 / 2)

機能	概要
TPU設定	<ul style="list-style-type: none"> • ユニット0のチャンネル0 (TPU0) 使用 • タイマプリスケアラ : PCLKD/64 (注 : PCLKD = 75MHz) TPU0.TCR.TPSC = 001b • 入カクロックエッジ : 立下りでエッジカウント TPU0.TCR.CKEG = 00b • カウンタクリア要因 : TGRAレジスタのコンペアマッチでTCNTカウンタクリア TPU0.TCR.CCLR = 001b • モード選択 : 通常動作 TPU0.TMDR.MD = 0000b • コンペアマッチ動作 : 出力禁止 TPU0.TIORH.IOA = 0000b
TPUaコールバック	TPUaのユニット0チャンネル0のコンペアマッチのコールバック関数を登録し、コールバック関数をパルス出力波形切り替えのトリガとして使用。
PPG設定	<ul style="list-style-type: none"> • PPG1のトリガにTPUaユニット0のチャンネル0を選択 PPG1.PTRSLR.PTRSL = 1b • PPG1.PCR.G0CMS = 00b (グループ0 : TPU0のコンペアマッチ) • PPG1.PCR.G1CMS = 00b (グループ1 : TPU0のコンペアマッチ) • PPG1.PCR.G2CMS = 00b (グループ2 : TPU0のコンペアマッチ) • PPG1.PCR.G3CMS = 00b (グループ3 : TPU0のコンペアマッチ) • PPG1.PMR.G0NOV = 0b (グループ0 : 通常動作) • PPG1.PMR.G1NOV = 0b (グループ1 : 通常動作) • PPG1.PMR.G2NOV = 0b (グループ2 : 通常動作) • PPG1.PMR.G3NOV = 0b (グループ3 : 通常動作) • PPG1.PMR.G0INV = 1b (グループ0 : 直接出力) • PPG1.PMR.G1INV = 1b (グループ1 : 直接出力) • PPG1.PMR.G2INV = 1b (グループ2 : 直接出力) • PPG1.PMR.G3INV = 1b (グループ3 : 直接出力)
使用端子	<p>PT0 ~ 3, PS6, PS7 を PPG機能にて使用</p> <ul style="list-style-type: none"> • PT0 ⇒ PORTT.PMR.B0 = 1 (周辺機能として使用) • PT1 ⇒ PORTT.PMR.B1 = 1 (周辺機能として使用) • PT2 ⇒ PORTT.PMR.B2 = 1 (周辺機能として使用) • PT3 ⇒ PORTT.PMR.B3 = 1 (周辺機能として使用) • PS6 ⇒ PORTS.PMR.B6 = 1 (周辺機能として使用) • PS7 ⇒ PORTS.PMR.B7 = 1 (周辺機能として使用) <p>PO23 ~ 28 を使用</p> <ul style="list-style-type: none"> • PO23 (PS6/TIOCA5/TIOCB5) (端子番号: R19) → PS6PFS = 0x06 • PO24 (PS7/TIOCA4/TIOCB4) (端子番号: R20) → PS7PFS = 0x06 • PO25 (PT0/TIOCA3/TIOCB3) (端子番号: P19) → PT0PFS = 0x06 • PO26 (PT1/TIOCA2/TIOCB2) (端子番号: P20) → PT1PFS = 0x06 • PO27 (PT2/TIOCA1/TIOCB1) (端子番号: N19) → PT2PFS = 0x06 • PO28 (PT3/TIOCA0/TIOCB0) (端子番号: N20) → PT3PFS = 0x06

表6.1 動作概要 (2 / 2)

機能	概要
処理概要	<ul style="list-style-type: none"> TPUaチャンネル0のコンペアマッチの一定周期毎にPO23, PO24, PO25, PO26, PO27, PO28端子からのPPG出力レベルを変更することで、パルス波形を出力する。 (R_PPG_Control APIを介して端子からの出力値を設定)

図 6.1 にシステムブロックを示します。

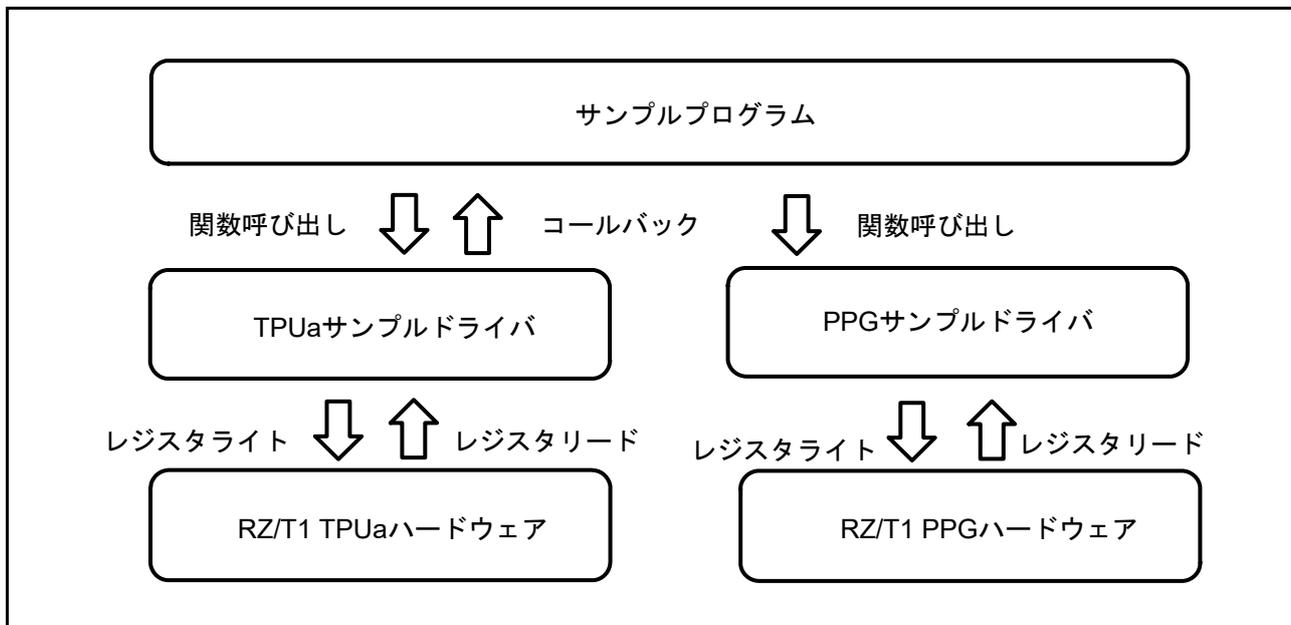


図 6.1 システムブロック

6.1.1 プロジェクト設定

開発環境となる EWARM 上で使用されるプロジェクト設定については、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.1.2 使用準備

本サンプルプログラムの実行準備は必要ありません。

6.2 メモリマップ

RZ/T1 グループのアドレス空間と RZ/T1 評価ボードのメモリマッピングについては、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.2.1 サンプルプログラムのセクション配置

サンプルプログラムで使用するセクションおよびサンプルプログラムの初期状態のセクション配置（ロードビュー）、スキップロード機能を使用後のセクション配置（実行ビュー）は、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.2.2 MPU の設定

MPU の設定は、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.2.3 例外処理ベクタテーブル

例外処理ベクタテーブルについては、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.3 使用割り込み一覧

サンプルプログラムで使用する割り込みを以下に示します。

表6.2 サンプルプログラムで使用する割り込み

割り込み（要因ID）	優先度	処理概要
TPU0.TGRAのインプットキャプチャ/コンペアマッチ割り込み（TG10A）	3	タイマコンペアマッチ処理
TPU0.TCNTのオーバーフロー割り込み（TG10V）	3	オーバーフロー処理

6.4 固定幅整数一覧

表 6.3 にサンプルプログラムで使用する固定幅整数を示します。

表6.3 サンプルプログラムで使用する固定幅整数

シンボル	内容
int8_t	8ビット整数、符号あり（標準ライブラリにて定義）
int16_t	16ビット整数、符号あり（標準ライブラリにて定義）
int32_t	32ビット整数、符号あり（標準ライブラリにて定義）
int64_t	64ビット整数、符号あり（標準ライブラリにて定義）
uint8_t	8ビット整数、符号なし（標準ライブラリにて定義）
uint16_t	16ビット整数、符号なし（標準ライブラリにて定義）
uint32_t	32ビット整数、符号なし（標準ライブラリにて定義）
uint64_t	64ビット整数、符号なし（標準ライブラリにて定義）

6.5 定数／エラーコード一覧

表 6.4 に TPUa、PPG サンプルプログラムにて使用する定数、表 6.5、表 6.6 に TPUa サンプルドライバにて使用する定数、表 6.7、表 6.8 に PPG サンプルドライバにて使用する定数を示します。

表6.4 TPUa、PPGサンプルプログラム定数

定数名	設定値	内容
TPUA_TIME_10MS	0x2DC6	10msのタイマカウント値 (タイマのプリスケアラ設定：PCLKD/64)
PPG_OUTPUTDATA_TBL_WORDSIZE	sizeof(ppg_outputdata) / 2	ppg_outputdataテーブルデータのワードサイズ値

表6.5 TPUaサンプルドライバ定数（グローバル）(1 / 3)

定数名	設定値	内容
TPUA_TCR_REG	0x00000001	R_TPUA_Controlの引数のパラメータとして使用されるTCRレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TMDR_REG	0x00000002	R_TPUA_Controlの引数のパラメータとして使用されるTMDRレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TIORH_REG	0x00000004	R_TPUA_Controlの引数のパラメータとして使用されるTIORHレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TIOR_REG	0x00000008	R_TPUA_Controlの引数のパラメータとして使用されるTIORレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TIORL_REG	0x00000010	R_TPUA_Controlの引数のパラメータとして使用されるTIORLレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TIER_REG	0x00000020	R_TPUA_Controlの引数のパラメータとして使用されるTIERレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TSR_REG	0x00000040	R_TPUA_Controlの引数のパラメータとして使用されるTSRレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TCNT_REG	0x00000080	R_TPUA_Controlの引数のパラメータとして使用されるTCNTレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TGRA_REG	0x00000100	R_TPUA_Controlの引数のパラメータとして使用されるTGRAレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TGRB_REG	0x00000200	R_TPUA_Controlの引数のパラメータとして使用されるTGRBレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TGRC_REG	0x00000400	R_TPUA_Controlの引数のパラメータとして使用されるTGRCレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TGRD_REG	0x00000800	R_TPUA_Controlの引数のパラメータとして使用されるTGRDレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TSTRA_REG	0x00001000	R_TPUA_Controlの引数のパラメータとして使用されるTSTRAレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TSTRB_REG	0x00002000	R_TPUA_Controlの引数のパラメータとして使用されるTSTRBレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TSYRA_REG	0x00004000	R_TPUA_Controlの引数のパラメータとして使用されるTSYRAレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_TSYRB_REG	0x00008000	R_TPUA_Controlの引数のパラメータとして使用されるTSYRBレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_NFCR_REG	0x00010000	R_TPUA_Controlの引数のパラメータとして使用されるNFCRレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_PWMFBSLR_REG	0x00020000	R_TPUA_Controlの引数のパラメータとして使用されるPWMFBSLRレジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_ALL_REG	0x0003FFFF	R_TPUA_Controlの引数のパラメータとして使用される全レジスタを示す情報 (6.11.5、6.11.6参照)
TPUA_CFG_PARAM_CHECKING_ENABLE	1	API引数のパラメータチェック処理有効／無効選択 0：無効／1：有効

表6.5 TPUaサンプルドライバ定数（グローバル）(2 / 3)

定数名	設定値	内容
TPUA_IR_PRIORITY_TPU0_A	3	TPU0Aの割り込み優先度
TPUA_IR_PRIORITY_TPU0_B	3	TPU0Bの割り込み優先度
TPUA_IR_PRIORITY_TPU0_C	3	TPU0Cの割り込み優先度
TPUA_IR_PRIORITY_TPU0_D	3	TPU0Dの割り込み優先度
TPUA_IR_PRIORITY_TPU0_V	3	TPU0Vの割り込み優先度
TPUA_IR_PRIORITY_TPU1_A	3	TPU1Aの割り込み優先度
TPUA_IR_PRIORITY_TPU1_B	3	TPU1Bの割り込み優先度
TPUA_IR_PRIORITY_TPU1_V	3	TPU1Vの割り込み優先度
TPUA_IR_PRIORITY_TPU1_U	3	TPU1Uの割り込み優先度
TPUA_IR_PRIORITY_TPU2_A	3	TPU2Aの割り込み優先度
TPUA_IR_PRIORITY_TPU2_B	3	TPU2Bの割り込み優先度
TPUA_IR_PRIORITY_TPU2_V	3	TPU2Vの割り込み優先度
TPUA_IR_PRIORITY_TPU2_U	3	TPU2Uの割り込み優先度
TPUA_IR_PRIORITY_TPU3_A	3	TPU3Aの割り込み優先度
TPUA_IR_PRIORITY_TPU3_B	3	TPU3Bの割り込み優先度
TPUA_IR_PRIORITY_TPU3_C	3	TPU3Cの割り込み優先度
TPUA_IR_PRIORITY_TPU3_D	3	TPU3Dの割り込み優先度
TPUA_IR_PRIORITY_TPU3_V	3	TPU3Vの割り込み優先度
TPUA_IR_PRIORITY_TPU4_A	3	TPU4Aの割り込み優先度
TPUA_IR_PRIORITY_TPU4_B	3	TPU4Bの割り込み優先度
TPUA_IR_PRIORITY_TPU4_V	3	TPU4Vの割り込み優先度
TPUA_IR_PRIORITY_TPU4_U	3	TPU4Uの割り込み優先度
TPUA_IR_PRIORITY_TPU5_A	3	TPU5Aの割り込み優先度
TPUA_IR_PRIORITY_TPU5_B	3	TPU5Bの割り込み優先度
TPUA_IR_PRIORITY_TPU5_V	3	TPU5Vの割り込み優先度
TPUA_IR_PRIORITY_TPU5_U	3	TPU5Uの割り込み優先度
TPUA_IR_PRIORITY_TPU6_A	3	TPU6Aの割り込み優先度
TPUA_IR_PRIORITY_TPU6_B	3	TPU6Bの割り込み優先度
TPUA_IR_PRIORITY_TPU6_C	3	TPU6Cの割り込み優先度
TPUA_IR_PRIORITY_TPU6_D	3	TPU6Dの割り込み優先度
TPUA_IR_PRIORITY_TPU6_V	3	TPU6Vの割り込み優先度
TPUA_IR_PRIORITY_TPU7_A	3	TPU7Aの割り込み優先度
TPUA_IR_PRIORITY_TPU7_B	3	TPU7Bの割り込み優先度
TPUA_IR_PRIORITY_TPU7_V	3	TPU7Vの割り込み優先度
TPUA_IR_PRIORITY_TPU7_U	3	TPU7Uの割り込み優先度
TPUA_IR_PRIORITY_TPU8_A	3	TPU8Aの割り込み優先度
TPUA_IR_PRIORITY_TPU8_B	3	TPU8Bの割り込み優先度
TPUA_IR_PRIORITY_TPU8_V	3	TPU8Vの割り込み優先度
TPUA_IR_PRIORITY_TPU8_U	3	TPU8Uの割り込み優先度
TPUA_IR_PRIORITY_TPU9_A	3	TPU9Aの割り込み優先度
TPUA_IR_PRIORITY_TPU9_B	3	TPU9Bの割り込み優先度
TPUA_IR_PRIORITY_TPU9_C	3	TPU9Cの割り込み優先度
TPUA_IR_PRIORITY_TPU9_D	3	TPU9Dの割り込み優先度
TPUA_IR_PRIORITY_TPU9_V	3	TPU9Vの割り込み優先度
TPUA_IR_PRIORITY_TPU10_A	3	TPU10Aの割り込み優先度

表6.5 TPUaサンプルドライバ定数（グローバル）(3 / 3)

定数名	設定値	内容
TPUA_IR_PRIORITY_TPU10_B	3	TPU10Bの割り込み優先度
TPUA_IR_PRIORITY_TPU10_V	3	TPU10Vの割り込み優先度
TPUA_IR_PRIORITY_TPU10_U	3	TPU10Uの割り込み優先度
TPUA_IR_PRIORITY_TPU11_A	3	TPU11Aの割り込み優先度
TPUA_IR_PRIORITY_TPU11_B	3	TPU11Bの割り込み優先度
TPUA_IR_PRIORITY_TPU11_V	3	TPU11Vの割り込み優先度
TPUA_IR_PRIORITY_TPU11_U	3	TPU11Uの割り込み優先度

表6.6 TPUaサンプルドライバ定数（ローカル）

定数名	設定値	内容
TPUA_NUM_CHANNELS	12	TPUa全チャンネル数
TPUA_HVA_WRITE_DATA	0x00000000u	HVAへの書き込みデータ
TPUA_CHANNEL0	0	チャンネル番号0
TPUA_CHANNEL1	1	チャンネル番号1
TPUA_CHANNEL2	2	チャンネル番号2
TPUA_CHANNEL3	3	チャンネル番号3
TPUA_CHANNEL4	4	チャンネル番号4
TPUA_CHANNEL5	5	チャンネル番号5
TPUA_CHANNEL6	6	チャンネル番号6
TPUA_CHANNEL7	7	チャンネル番号7
TPUA_CHANNEL8	8	チャンネル番号8
TPUA_CHANNEL9	9	チャンネル番号9
TPUA_CHANNEL10	10	チャンネル番号10
TPUA_CHANNEL11	11	チャンネル番号11
TPUA_TIER_BIT_INVALID	0x00	ビット位置無効
TPUA_TIER_BIT_TGIEA	0x01	TIERレジスタのTGIEAビット位置
TPUA_TIER_BIT_TGIEB	0x02	TIERレジスタのTGIEBビット位置
TPUA_TIER_BIT_TGIEC	0x04	TIERレジスタのTGIECビット位置
TPUA_TIER_BIT_TGIED	0x08	TIERレジスタのTGIEDビット位置
TPUA_TIER_BIT_TGIEV	0x10	TIERレジスタのTGIEVビット位置
TPUA_TIER_BIT_TGIEU	0x20	TIERレジスタのTGIEUビット位置
BITSHIFT_16	16u	16bitシフト数
NULL	0	NULL
TPUA_RZT1_VERSION_MAJOR	1	Major Version
TPUA_RZT1_VERSION_MINOR	0	Minor Version

表6.7 PPGサンプルドライバ定数（グローバル）

定数名	設定値	内容
PPG_PTRSLR_REG	0x00000001	R_TPUA_Controlの引数のパラメータとして使用されるPTRSLRレジスタを示す情報（6.12.1、6.12.2参照）
PPG_NDER_REG	0x00000002	R_TPUA_Controlの引数のパラメータとして使用されるNDERH/Lレジスタを示す情報（6.12.1、6.12.2参照）
PPG_PODR_REG	0x00000004	R_TPUA_Controlの引数のパラメータとして使用されるPODRH/Lレジスタを示す情報（6.12.1、6.12.2参照）
PPG_NDR_REG	0x00000008	R_TPUA_Controlの引数のパラメータとして使用されるNDRH/Lレジスタを示す情報（6.12.1、6.12.2参照）
PPG_PCR_REG	0x00000010	R_TPUA_Controlの引数のパラメータとして使用されるPCRレジスタを示す情報（6.12.1、6.12.2参照）
PPG_PMR_REG	0x00000020	R_TPUA_Controlの引数のパラメータとして使用されるPMRレジスタを示す情報（6.12.1、6.12.2参照）
PPG_ALL_REG	0x0000003F	R_TPUA_Controlの引数のパラメータとして使用される全レジスタを示す情報（6.12.1、6.12.2参照）
PPG_CFG_PARAM_CHECKING_ENABLE	1	API引数のパラメータチェック処理有効/無効選択 0：無効/1：有効

表6.8 PPGサンプルドライバ定数（ローカル）

定数名	設定値	内容
PPG_NUM_UNIT	2u	PPG全ユニット数
BITSHIFT_8	8u	8bitシフト数
BITSHIFT_16	16u	16bitシフト数
NULL	0	NULL
PPG_RZT1_VERSION_MAJOR	1	Major Version
PPG_RZT1_VERSION_MINOR	0	Minor Version

表6.9 TPUaサンプルドライバ（エラーコード）

定数名	設定値	内容
TPUA_SUCCESS	0	正常終了
TPUA_ERR_BAD_CHAN	1	チャンネル番号エラー
TPUA_ERR_CH_NOT_OPENED	2	オープンエラー（オープンされていない）
TPUA_ERR_CH_NOT_CLOSED	3	クローズエラー（クローズされていない）
TPUA_ERR_UNKNOWN_CMD	4	コマンドエラー
TPUA_ERR_NULL_PTR	5	NULLポインタ

表6.10 PPGサンプルドライバ（エラーコード）

定数名	設定値	内容
PPG_SUCCESS	0	正常終了
PPG_ERR_BAD_UNIT	1	ユニット番号エラー
PPG_ERR_UN_NOT_OPENED	2	オープンエラー（オープンされていない）
PPG_ERR_UN_NOT_CLOSED	3	クローズエラー（クローズされていない）
PPG_ERR_UNKNOWN_CMD	4	コマンドエラー
PPG_ERR_NULL_PTR	5	NULLポインタ

6.6 構造体／共用体／列挙型一覧

以下に TPUa/PPG サンプルドライバで使用する構造体／共用体／列挙型を示します。

表6.11 構造体／共用体（TPUaサンプルドライバ）

構造体／共用体定義	概要	定義ファイル
tpua_callback_t	TPUaコールバック関数情報	r_tpua_rzt1_if.h
tpua_handle_t	TPUaハンドル情報（チャンネル、ユニット情報他）	
tpua_reg_t	TPUaリード／ライトのレジスタ情報	
tpua_err_t	関数戻り値のエラー情報	
tpua_cmd_t	R_TPUA_Control関数の第2引数のコマンドコード	

表6.12 構造体／共用体（PPGサンプルドライバ）

構造体／共用体定義	概要	定義ファイル
ppg_handle_t	PPGハンドル情報（チャンネル、ユニット情報他）	r_ppg_rzt1_if.h
ppg_reg_t	PPGリード／ライトのレジスタ情報	
ppg_err_t	関数戻り値のエラー情報	
ppg_cmd_t	R_PPG_Control関数の第2引数のコマンドコード	

表6.13 TPUa構造体／共用体メンバ（1/2）

構造体／共用体定義	メンバ	説明
tpua_callback_t	void (*pintr_a)(void)	TPU 注1A 割り込み時のコールバック関数
	void (*pintr_b)(void)	TPU 注1B 割り込み時のコールバック関数
	void (*pintr_c)(void)	TPU 注2C 割り込み時のコールバック関数
	void (*pintr_d)(void)	TPU 注2D 割り込み時のコールバック関数
	void (*pintr_v)(void)	TPU 注1V 割り込み時のコールバック関数
	void (*pintr_u)(void)	TPU 注3U 割り込み時のコールバック関数
tpua_handle_t	uint8_t channel	TPUaのチャンネル情報（0～11）
	uint8_t unit	TPUaのユニット情報（0～3）
	bool tpua_channel_opened	チャンネルのオープン状態 true：オープン false：未オープン
	tpua_callback_t tpua_callback	tpua_callback_tの説明を参照
tpua_reg_t	uint32_t reg_flag	R_TPUA_Control関数の引数であるパラメータ情報（6.11.5、6.11.6、表6.5参照）
	uint8_t tcr_reg	TCRレジスタ情報 注4
	uint8_t tmdr_reg	TMDRレジスタ情報 注4
	uint8_t tiorh_reg	TIORHレジスタ情報 注4
	uint8_t tior_reg	TIORレジスタ情報 注4
	uint8_t tiorl_reg	TIORLレジスタ情報 注4
	uint8_t tier_reg	TIERレジスタ情報 注4
	uint8_t tsr_reg	TSRレジスタ情報 注4
	uint16_t tcnt_reg	TCNTレジスタ情報 注4
	uint16_t tgra_reg	TGRAレジスタ情報 注4
	uint16_t tgrb_reg	TGRBレジスタ情報 注4
	uint16_t tgrc_reg	TGRCレジスタ情報 注4
	uint16_t tgrd_reg	TGRDレジスタ情報 注4

表6.13 TPUa構造体／共用体メンバ (2/2)

構造体／共用体定義	メンバ	説明
tpua_reg_t	uint8_t tstra_reg	TSTRAレジスタ情報 注4
	uint8_t tstrb_reg	TSTRBレジスタ情報 注4
	uint8_t tsyra_reg	TSYRAレジスタ情報 注4
	uint8_t tsyrb_reg	TSYRBレジスタ情報 注4
	uint8_t nfcr_reg	NFCLレジスタ情報 注4
	uint32_t pwmfbslr_reg	PWMFBSLRレジスタ情報 注4

注1. 0～11

注2. 0, 3, 6, 9

注3. 1, 2, 4, 5, 7, 8, 10, 11

注4. レジスタ詳細に関しては、RZ/T1グループ ユーザーズマニュアル ハードウェア編 (R01UH0483JJ) を参照してください。

表6.14 PPG構造体／共用体メンバ

構造体／共用体定義	メンバ	説明
ppg_handle_t	uint8_t unit	ユニット
	bool ppg_unit_opened	ユニットのオープン状態 true : オープン false : 未オープン
ppg_reg_t	uint32_t reg_flag	R_PPG_Control関数の引数であるパラメータ情報 (6.12.1、6.12.2、表6.7参照)
	uint8_t ptrslr_reg	PTRSLRレジスタ情報 注1
	uint16_t nder_reg	NDERH/Lレジスタ情報 注1
	uint16_t podr_reg	PODRH/Lレジスタ情報 注1
	uint16_t ndr_reg	NDRH/Lレジスタ情報 注1
	uint8_t pcr_reg	PCRレジスタ情報 注1
uint8_t pmr_reg	PMRレジスタ情報 注1	

注1. レジスタ詳細に関しては、RZ/T1グループ ユーザーズマニュアル ハードウェア編 (R01UH0483JJ) を参照してください。

表6.15 TPUa列挙型

列挙型定義	リスト	説明
tpua_err_t	TPUA_SUCCESS	TPUa API実行結果 0 : 正常終了 1 : チャネル番号エラー 2 : オープンエラー (オープンされていない) 3 : クローズエラー (クローズされていない) 4 : コマンドエラー 5 : NULLポインタ
	TPUA_ERR_BAD_CHAN	
	TPUA_ERR_CH_NOT_OPENED	
	TPUA_ERR_CH_NOT_CLOSED	
	TPUA_ERR_UNKNOWN_CMD	
	TPUA_ERR_NULL_PTR	
tpua_cmd_t	TPUA_CMD_TIMER_START	TPUa APIコマンド 0 : タイマスタート 1 : タイマストップ 2 : タイマ同期 3 : タイマ非同期 4 : レジスタリード 5 : レジスタライト 6 : TGRA 割り込み許可 7 : TGRA 割り込み禁止 8 : TGRB 割り込み許可 9 : TGRB 割り込み禁止 10 : TGRC 割り込み許可 11 : TGRC 割り込み禁止 12 : TGRD 割り込み許可 13 : TGRD 割り込み禁止 14 : TGRV 割り込み許可 15 : TGRV 割り込み禁止 16 : TGRU 割り込み許可 17 : TGRU 割り込み禁止
	TPUA_CMD_TIMER_STOP	
	TPUA_CMD_TIMER_SYNC	
	TPUA_CMD_TIMER_ASYNC	
	TPUA_CMD_REG_READ	
	TPUA_CMD_REG_WRITE	
	TPUA_CMD_INTR_A_ENABLE	
	TPUA_CMD_INTR_A_DISABLE	
	TPUA_CMD_INTR_B_ENABLE	
	TPUA_CMD_INTR_B_DISABLE	
	TPUA_CMD_INTR_C_ENABLE	
	TPUA_CMD_INTR_C_DISABLE	
	TPUA_CMD_INTR_D_ENABLE	
	TPUA_CMD_INTR_D_DISABLE	
	TPUA_CMD_INTR_V_ENABLE	
	TPUA_CMD_INTR_V_DISABLE	
	TPUA_CMD_INTR_U_ENABLE	
	TPUA_CMD_INTR_U_DISABLE	

表6.16 PPG列挙型

列挙型定義	リスト	説明
ppg_err_t	PPG_SUCCESS	PPG API実行結果 0 : 正常終了 1 : ユニット番号エラー 2 : オープンエラー (オープンされていない) 3 : クローズエラー (クローズされていない) 4 : コマンドエラー 5 : NULLポインタ
	PPG_ERR_BAD_UNIT	
	PPG_ERR_UN_NOT_OPENED	
	PPG_ERR_UN_NOT_CLOSED	
	PPG_ERR_UNKNOWN_CMD	
	PPG_ERR_NULL_PTR	
ppg_cmd_t	PPG_CMD_REG_READ	PPG APIコマンド 0 : レジスタリード 1 : レジスタライト
	PPG_CMD_REG_WRITE	

6.7 大域変数一覧

表 6.17 に TPUa/PPG のサンプルプログラム／ドライバの static/const 型変数を示します。

表6.17 大域変数一覧

型	変数名	内容	使用関数
static tpu_a_handle_t	gb_tpu_a_handles[12]	TPUaチャンネルのハンドル TPUaの1チャンネルに対し1ハンドル が必要なため、12チャンネル分の12ハ ンドルを配列として確保している。	R_TPUA_Open R_TPUA_Control R_TPUA_Close
static void (* const tpu_a_reg_read_tbl[TPUA_ NUM_CHANNELS])(const uint8_t unit, tpu_a_reg_t * const prddat)	tpu_a_reg_read_tbl	TPUa各チャンネルのレジスタリード関 数テーブル TPU0/6, 1/7, 2/8, 3/9, 4/10, 5/11の各 レジスタリード関数を並べたテーブ ル。	R_TPUA_Control
static void (* const tpu_a_reg_write_tbl[TPUA_ NUM_CHANNELS])(const uint8_t unit, tpu_a_reg_t * const pwrdat)	tpu_a_reg_write_tbl	TPUa各チャンネルのレジスタライト関 数テーブル TPU0/6, 1/7, 2/8, 3/9, 4/10, 5/11の各 レジスタライト関数を並べたテーブ ル。	R_TPUA_Control
static const tpu_a_channel_adr_t	tpu_a_channel_adr_tbl	TPUa各チャンネルのレジスタアドレ ステーブル TPU0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11の レジスタアドレスを並べたテーブル。	R_TPUA_Control
static ppg_handle_t	gb_ppg_handles[2]	PPGユニットのハンドル PPGの1ユニットに対し1ハンドルが 必要なため、2ユニット分の2ハンド ルを配列として確保している。	R_PPG_Open R_PPG_Control R_PPG_Close
ppg_unit_adr_t	ppg_unit_adr_tbl	PPG各ユニットのレジスタアドレ ステーブル PPG0, 1のレジスタアドレスを並べた テーブル。	R_PPG_Control
volatile static bool	tpu_a_intr_a_flag	コンペアマッチ割り込み発生フラグ	main tpu0_a_cmpmatch_callback
volatile static bool	tpu_a_intr_v_flag	オーバーフロー割り込み発生フラグ	main tpu0_v_callback
static bool	tpu_a_main_exit_flag	メイン終了フラグ	main
static const uint16_t	ppg_outputdata[]	PPG出力パルス波形のテーブルデー タ 1端子のレベルを1bitデータとして、 PO31-16端子に出力するレベルを 16bitの1データとし、時系列に並べ た以下テーブル。 const uint16_t ppg_outputdata[] = { 0x0380, 0x0700, 0x0E00, 0x1C00, 0x1880, 0x1180 }	

6.8 関数一覧

表 6.18 に関数一覧を示します。

表 6.18 関数一覧

関数名	ページ番号
main	21
tpu0_a_cmpmatch_callback	23
tpu0_v_callback	24
R_TPUA_Open	24
R_TPUA_Control	26
R_TPUA_Close	27
R_TPUA_GetVersion	27
R_PPG_Open	28
R_PPG_Control	29
R_PPG_Close	30
R_PPG_GetVersion	30

6.9 関数仕様

サンプルコードの関数仕様を示します。

6.9.1 main

main	
概要	サンプルプログラムのメイン処理
ヘッダ	—
宣言	void main(void)
説明	<p>TPUa、PPG の Open/Close/Control 関数を使用して、出力端子からパルス波形パターンを出力する。</p> <p>○主な処理内容</p> <ul style="list-style-type: none"> • 出力端子設定 (PO23, 24, 25, 26, 27, 28) <ul style="list-style-type: none"> - PT0 ~ 3, PS6, PS7 端子を PPG 機能として使用する設定。 PORTT.PMR.B0 = 1 (PT0 端子を周辺機能として使用) PORTT.PMR.B1 = 1 (PT1 端子を周辺機能として使用) PORTT.PMR.B2 = 1 (PT2 端子を周辺機能として使用) PORTT.PMR.B3 = 1 (PT3 端子を周辺機能として使用) PORTS.PMR.B6 = 1 (PS6 端子を周辺機能として使用) PORTS.PMR.B7 = 1 (PS7 端子を周辺機能として使用) - PO23 ~ 28 を PT0 ~ 3, PS6, PS7 端子へ割り当てる設定。 PS6PFS = 0x06 (PO23 を PS6 端子へ割り当てる) PS7PFS = 0x06 (PO24 を PS7 端子へ割り当てる) PT0PFS = 0x06 (PO25 を PT0 端子へ割り当てる) PT1PFS = 0x06 (PO26 を PT1 端子へ割り当てる) PT2PFS = 0x06 (PO27 を PT2 端子へ割り当てる) PT3PFS = 0x06 (PO28 を PT3 端子へ割り当てる) • TPUa チャンネル 0 オープン TGRA コンペアマッチ割り込み、オーバフロー割り込みのコールバック登録 • TPUa チャンネル 0 設定 <ul style="list-style-type: none"> - チャンネル選択 : 0 チャンネル - プリスケーラ設定 : PCLKD/64 (*PCLKD = 75MHz) - 入力クロックエッジ設定 : 立下りエッジ - カウンタクリア要因設定 : TGRA レジスタのコンペアマッチで TCNT カウンタクリア - モード設定 : 通常動作 - コンペアマッチ/インプットキャプチャ動作設定 : コンペアマッチ出力禁止 - コンペアマッチタイマ 10ms 設定

- PPG ユニット 1 オープン
 - ・ PPG ユニット 1 設定
 - PPG1 のトリガに TPUa ユニット 0 のチャンネル 0 を選択
PPG1.PTRSLR.PTRSL = 1b (PPG1 のトリガは TPU0-3)
PPG1.PCR.G0CMS = 00b (グループ 0 : TPU0 のコンペアマッチ)
PPG1.PCR.G1CMS = 00b (グループ 1 : TPU0 のコンペアマッチ)
PPG1.PCR.G2CMS = 00b (グループ 2 : TPU0 のコンペアマッチ)
PPG1.PCR.G3CMS = 00b (グループ 3 : TPU0 のコンペアマッチ)
PPG1.PMR.G0NOV = 0b (グループ 0 : 通常動作)
PPG1.PMR.G1NOV = 0b (グループ 1 : 通常動作)
PPG1.PMR.G2NOV = 0b (グループ 2 : 通常動作)
PPG1.PMR.G3NOV = 0b (グループ 3 : 通常動作)
PPG1.PMR.G0INV = 1b (グループ 0 : 直接出力)
PPG1.PMR.G1INV = 1b (グループ 1 : 通常動作)
PPG1.PMR.G2INV = 1b (グループ 2 : 通常動作)
PPG1.PMR.G3INV = 1b (グループ 3 : 通常動作)
 - PO23-28 端子使用の設定
- TPUa チャンネル 0 コンペアマッチ A 割り込み (コールバック) をトリガとして、10ms 割り込み周期毎に出力パターンのテーブル (表 6.17) を切り替えることで、図 6.2 のパルス波形を出力する。
- TPUa チャンネル 0 オーバフロー割り込み (コールバック) を検出した場合は、エラーとして TPUa チャンネル 0、PPG ユニット 1 をクローズする。

引 数 なし
リターン値 なし

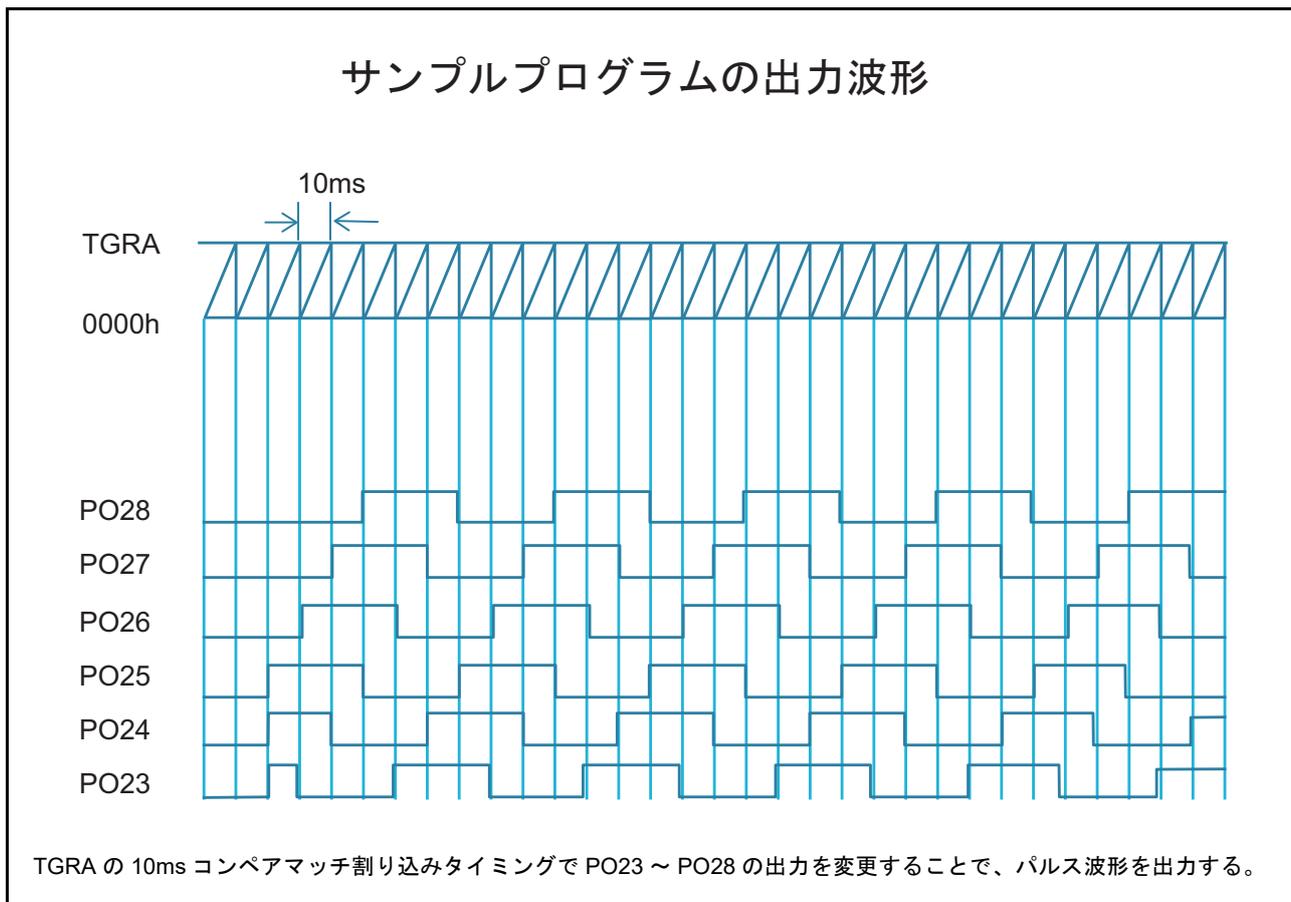


図 6.2 出力波形

6.9.2 tpu0_a_cmpmatch_callback

tpu0_a_cmpmatch_callback

概要	TPU0A（コンペアマッチ）割り込みのコールバック関数
ヘッダ	—
宣言	static void tpu0_a_cmpmatch_callback(void)
説明	コンペアマッチ割り込み発生フラグ（表 6.17）をセットする。 main にて本フラグを監視することでコンペアマッチ割り込みの有無を検出する。
引数	なし
リターン値	なし

6.9.3 tpu0_v_callback

tpu0_v_callback

概要	TPU0V（オーバフロー）割り込みのコールバック関数
ヘッダ	—
宣言	static void tpu0_v_callback(void)
説明	オーバフロー発生フラグ（表 6.17）をセットする。 main にて本フラグを監視することでオーバフロー割り込みの有無を検出する。
引数	なし
リターン値	なし

6.9.4 R_TPUA_Open

R_TPUA_Open

概要	TPUa モジュールのオープン
ヘッダ	r_tpuarzt1_if.h
宣言	tpua_err_t R_TPUA_Open(const uint8_t channel, const tpu_callback_t * const pcallback, tpua_handle_t * const phandle)
説明	<p>TPUa の指定チャンネル（第 1 引数）に対して、TPUa モジュールのオープンおよびパラメータ（第 2 引数）にて指定されるコールバック関数の登録を行い、オープンしたチャンネル情報をハンドル（第 3 引数）へ設定して返す。関数の実行結果をリターン値として返す。</p> <p>○主な処理内容</p> <ul style="list-style-type: none"> 引数チェック <ul style="list-style-type: none"> - チャンネル（第 1 引数）の指定範囲チェック - パラメータ（第 2 引数）のポインタ NULL チェック - ハンドル（第 3 引数）のポインタ NULL チェック - チャンネルのオープン状態チェック <ul style="list-style-type: none"> すでにチャンネルがオープンされているとエラー （すでにオープンされているチャンネルはオープンできない。再度オープンする場合は R_TPUA_Close 関数にて一度クローズする必要あり） オープン状態の設定 コールバック関数の登録 TPUa ユニット 0/1 モジュールストップ状態の解除（消費電力低減機能 OFF） <ul style="list-style-type: none"> 指定チャンネル（第 1 引数）の割り込みベクタ設定および ICU による割り込み許可

○コールバック関数の登録方法

- TGRA 割り込みに callback_a 関数、TGRV 割り込みに callback_b 関数を登録する場合。

⇒ TPUa チャンネル 0 に第 2 引数（表 6.13）callbacktbl の構造体メンバの pintr_a に callback_a 関数を、pintr_v に callback_v 関数を設定する。（未使用のコールバック関数は NULL を指定）

```
callbacktbl.pintr_a = &callback_a;
callbacktbl.pintr_b = NULL;
callbacktbl.pintr_c = NULL;
callbacktbl.pintr_d = NULL;
callbacktbl.pintr_v = &callback_v;
R_TPUA_Open(0, &callbacktbl, &handle)
```

引 数	const uint8_t channel	: チャンネル指定 TPUa のチャンネルを 0 ~ 11 で指定。 ユニット 0 : TPUa 0 ~ 5 チャンネル ユニット 1 : TPUa 6 ~ 11 チャンネル
	const tpu_a_callback_t * const pcallback	: コールバック指定 TGRA/TGRB/TGRC/TGRD/TGRV/TGRU 割り込み発生時に実行するコールバック関数を指定。
	tpua_handle_t * const phandle	: TPUa のハンドルのポインタを指定 phandle のポインタ指定されたエリアへオープンした TPUa のチャンネル情報を返す。 phandle のポインタが指定するエリアは、R_TPUA_Open を呼び出す処理にて確保が必要。
リターン値	TPUA_SUCCESS	: 成功 : TPUa モジュールのオープン成功
	TPUA_ERR_BAD_CHAN	: 失敗 : 不正なチャンネル
	TPUA_ERR_CH_NOT_CLOSED	: 失敗 : すでにオープンされている
	TPUA_ERR_NULL_PTR	: 失敗 : パラメータが指定されていない

6.9.5 R_TPUA_Control

R_TPUA_Control

概要	TPUa モジュールのコマンド実行	
ヘッダ	r_tpua_rzt1_if.h	
宣言	<pre>tpua_err_t R_TPUA_Control(const tpua_handle_t * const phandle, const tpua_cmd_t cmd, tpua_reg_t * const pregdat)</pre>	
説明	<p>ハンドル（第 1 引数）により指定されたチャンネルに対して、指定されたコマンド（第 2 引数）を実行する。実行するコマンドによりパラメータ（第 3 引数）の使用方法は異なり、データを参照する場合とデータを設定して返す場合がある。関数の実行結果をリターン値として返す。</p> <p>○主な処理内容</p> <ul style="list-style-type: none"> 引数チェック <ul style="list-style-type: none"> - 指定ハンドル（第 1 引数）のポインタ NULL チェック - チャンネルのオープン状態チェック チャンネルがオープンされていないとエラー (R_TPUA_Control 関数は、事前に R_TPUA_Open 関数にてチャンネルをオープンした状態で実行する必要あり) コマンド処理 第 2 引数により指定されたコマンドにより、各処理を実行する。 該当しないコマンドの場合は不明なコマンドとしてエラーを返す。 コマンド内容に関しては、6.11 R_TPUA_Control コマンド一覧を参照。 	
引数	<pre>const tpua_handle_t * const phandle</pre> <pre>const tpua_cmd_t cmd</pre> <pre>tpua_reg_t * const pregdat</pre>	<p>: TPUa のハンドルのポインタを指定 R_TPUA_Open にてオープン済みのハンドルを指定する。</p> <p>: 実行するコマンドを指定 (詳細は 6.11 を参照)</p> <p>: レジスタデータのポインタを指定 レジスタリードコマンド実行時には、本引数へレジスタのリード値を格納して返す。 レジスタライトコマンド実行時には、本引数へ設定されている値をレジスタへライトする。 (詳細は 6.11 を参照)</p>
リターン値	<pre>TPUA_SUCCESS</pre> <pre>TPUA_ERR_CH_NOT_OPENED</pre> <pre>TPUA_ERR_NULL_PTR</pre> <pre>TPUA_ERR_UNKNOWN_CMD</pre>	<p>: 成功 : TPUa モジュールのコマンド成功</p> <p>: 失敗 : チャンネルがオープンされていない</p> <p>: 失敗 : パラメータが指定されていない</p> <p>: 失敗 : 不明なコマンドが指定された</p>

6.9.6 R_TPUA_Close

R_TPUA_Close

概要	TPUa モジュールのクローズ	
ヘッダ	r_tpuarzt1_if.h	
宣言	tpua_err_t R_TPUA_Close(const tpuahandle_t * const phandle)	
説明	<p>ハンドル（第1引数）により指定されたチャンネルをクローズする。 関数の実行結果をリターン値として返す。</p> <p>○主な処理内容</p> <ul style="list-style-type: none"> 引数チェック <ul style="list-style-type: none"> 指定ハンドル（第1引数）のポインタ NULL チェック チャンネルのオープン状態チェック チャンネルがオープンされていないとエラー (すでにクローズされているチャンネルはクローズできない。R_TPUA_Open関数にてオープンされた状態のチャンネルしかクローズできない) 指定チャンネル（第1引数）割り込みをICUにて禁止 オープン状態の解除 TPUa ユニット 0/1 モジュールストップ状態への遷移（消費電力低減機能 ON） ユニット中のすべてのチャンネルがクローズされた場合にのみ消費電力低減機能をONする。ユニット中のチャンネルが1つでもオープンされていれば消費電力機能はONしない。 	
引数	const tpuahandle_t * const phandle	: TPUa のハンドルのポインタを指定 R_TPUA_Openにてオープン済みのハンドルを指定する。
リターン値	TPUA_SUCCESS TPUA_ERR_CH_NOT_OPENED TPUA_ERR_NULL_PTR	: 成功 : TPUa モジュールのクローズ成功 : 失敗 : チャンネルがオープンされていない : 失敗 : パラメータが指定されていない

6.9.7 R_TPUA_GetVersion

R_TPUA_GetVersion

概要	TPUa モジュールのバージョン情報取得	
ヘッダ	r_tpuarzt1_if.h	
宣言	uint32_t R_TPUA_GetVersion(void)	
説明	TPUa モジュールのバージョン情報をリターン値として返す	
引数	なし	
リターン値	TPUa サンプルドライバのバージョン情報 (32bit) 16-31bit : Major Version 0-15bit : Minor Version	

6.9.8 R_PPG_Open

R_PPG_Open

概 要	PPG モジュールのオープン	
ヘッダ	r_ppg_rzt1_if.h	
宣 言	ppg_err_t R_PPG_Open(const uint8_t unit, ppg_handle_t * const phandle)	
説 明	<p>PPG の指定ユニット（第 1 引数）に対して、PPG モジュールのオープンを行い、オープンしたユニット情報をハンドル（第 2 引数）へ設定して返す。関数の実行結果をリターン値として返す。</p> <p>○主な処理内容</p> <ul style="list-style-type: none"> • 引数チェック <ul style="list-style-type: none"> - ユニット（第 1 引数）の指定範囲チェック - ハンドル（第 2 引数）のポインタ NULL チェック - ユニットのオープン状態チェック • オープン状態の設定 • PPG ユニット 0/1 モジュールストップ状態の解除（消費電力低減機能 OFF） <p>すでにユニットがオープンされているとエラー （すでにオープンされているユニットはオープンできない。再度オープンする場合は R_PPG_Close 関数にて一度クローズする必要あり）</p>	
引 数	const uint8_t unit	: ユニット指定 PPG のユニットを 0, 1 で指定。
	ppg_handle_t * const phandle	: PPG のハンドルのポインタを指定 phandle のポインタ指定されたエリアへ、オープンした PPG のユニット情報を返す。 phandle のポインタが指定するエリアは、R_PPG_Open を呼び出す処理にて確保が必要。
リターン値	PPG_SUCCESS	: 成功 : PPG モジュールのオープン成功
	PPG_ERR_BAD_UNIT	: 失敗 : 不正なチャンネル
	PPG_ERR_UN_NOT_CLOSED	: 失敗 : 既にオープンされている
	PPG_ERR_NULL_PTR	: 失敗 : パラメータが指定されていない

6.9.9 R_PPG_Control

R_PPG_Control

概 要	PPG モジュールのコマンド実行	
ヘッダ	r_ppg_rzt1_if.h	
宣 言	ppg_err_t R_PPG_Control(const ppg_handle_t * const phandle, const ppg_cmd_t cmd, ppg_reg_t * const pregdat)	
説 明	<p>ハンドル（第 1 引数）により指定されたユニットに対して、指定されたコマンド（第 2 引数）を実行する。実行するコマンドによりパラメータ（第 3 引数）の使用方法は異なり、データを参照する場合とデータを設定して返す場合がある。関数の実行結果をリターン値として返す。</p> <p>○主な処理内容</p> <ul style="list-style-type: none"> • 引数チェック（引き数にかかわらず共通な処理） <ul style="list-style-type: none"> - 指定ハンドル（第 1 引数）のポインタ NULL チェック - ユニットのオープン状態チェック ユニットがオープンされていないとエラー (R_PPG_Control 関数は、事前に R_PPG_Open 関数にてユニットをオープンした状態で実行する必要あり) • コマンド処理 第 2 引数により指定されたコマンドにより、各処理を実行する。 該当しないコマンドの場合は不明なコマンドとしてエラーを返す。 コマンド内容に関しては、6.12 R_PPG_Control コマンド一覧を参照。 	
引 数	const ppg_handle_t * const phandle	: PPG のハンドルのポインタを指定 R_PPG_Open にてオープン済みのハンドルを指定する。
	const ppg_cmd_t cmd	: 実行するコマンドを指定 (詳細は 6.12 を参照)
	ppg_reg_t * const pregdat	: レジスタリードコマンド実行時には、本引数へレジスタのリード値を格納して返す。 レジスタライトコマンド実行時には、本引数へ設定されている値をレジスタへライトする。 (詳細は 6.12 を参照)
リターン値	PPG_SUCCESS	: 成功 : PPG モジュールのコマンド成功
	PPG_ERR_UN_NOT_OPENED	: 失敗 : ユニットのオープンされていない
	PPG_ERR_NULL_PTR	: 失敗 : パラメータが指定されていない
	PPG_ERR_UNKNOWN_CMD	: 失敗 : 不明なコマンドが指定された

6.9.10 R_PPG_Close

R_PPG_Close

概要	PPG モジュールのクローズ	
ヘッダ	r_ppg_rzt1_if.h	
宣言	ppg_err_t R_PPG_Close(const ppg_handle_t * const phandle)	
説明	<p>指定ハンドル（第 1 引数）により指定されたユニットをクローズする。 関数の実行結果をリターン値として返す。</p> <p>○主な処理内容</p> <ul style="list-style-type: none"> ・引数チェック <ul style="list-style-type: none"> - 指定ハンドル（第 1 引数）のポインタ NULL チェック - ユニットのオープン状態チェック ユニットがオープンされていないとエラー (すでにクローズされているユニットはクローズできない。R_PPG_Open 関数にてオープンされた状態のユニットしかクローズできない) ・オープン状態の解除 ・PPG ユニット 0/1 モジュールストップ状態への遷移（消費電力低減機能 ON） 	
引数	const ppg_handle_t * const phandle	: PPG のハンドルのポインタを指定 R_PPG_Open にてオープン済みのハンドルを指定する。
リターン値	PPG_SUCCESS PPG_ERR_UN_NOT_OPENED PPG_ERR_NULL_PTR	: 成功 : PPG モジュールのクローズ成功 : 失敗 : ユニットのオープンされていない : 失敗 : パラメータが指定されていない

6.9.11 R_PPG_GetVersion

R_PPG_GetVersion

概要	PPG モジュールのバージョン情報取得	
ヘッダ	r_ppg_rzt1_if.h	
宣言	uint32_t R_PPG_GetVersion(void)	
説明	PPG モジュールのバージョン情報をリターン値として返す	
引数	なし	
リターン値	PPG サンプルドライバのバージョン情報（32bit） 16-31bit : Major Version 0-15bit : Minor Version	

6.10 フローチャート

6.10.1 main 処理

図 6.3 に main 処理のフローチャートを示します。

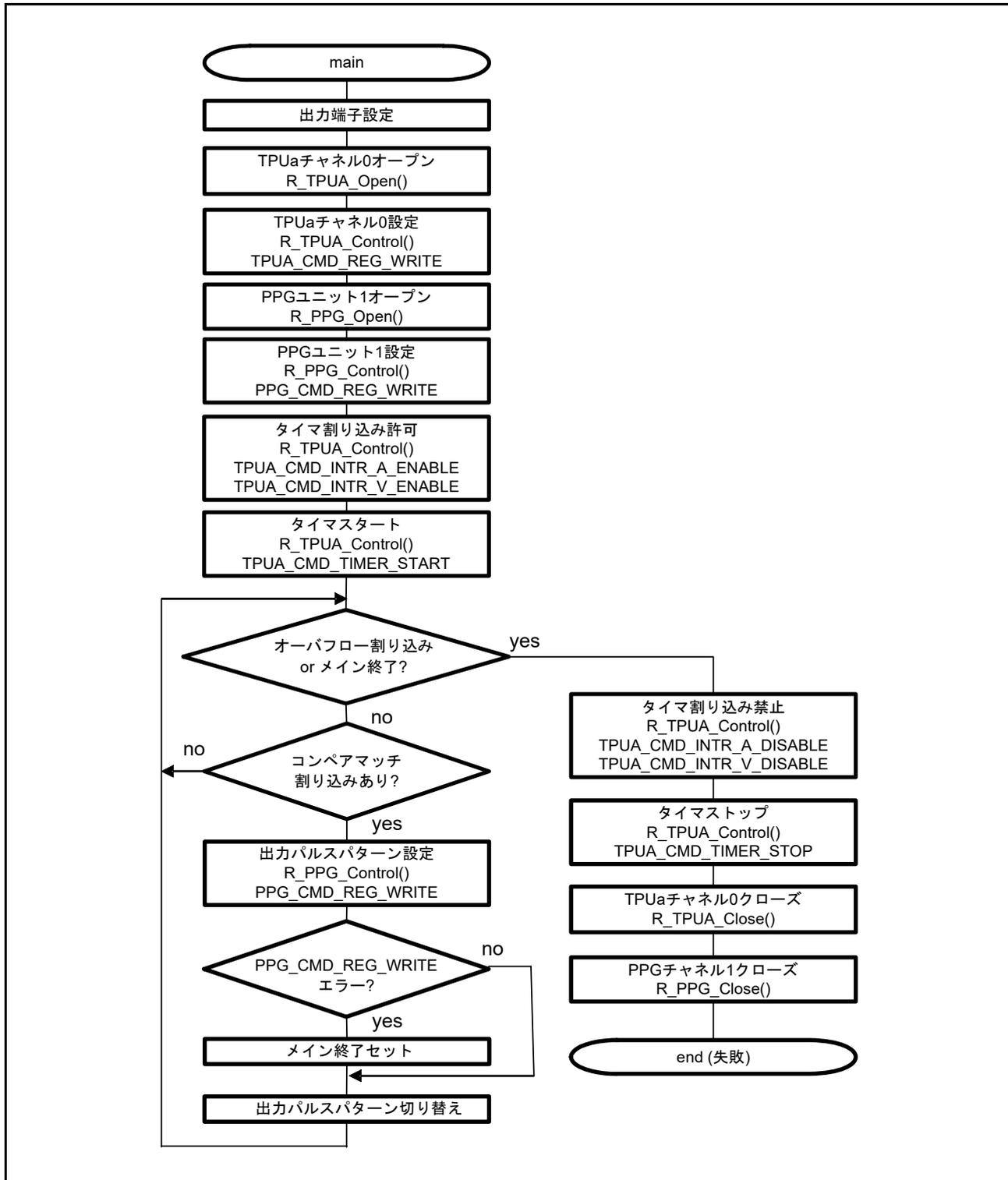


図 6.3 main 処理

6.10.2 tpu0_a_cmpmatch_callback 処理

図 6.4 に tpu0_a_cmpmatch_callback のフローチャートを示します。

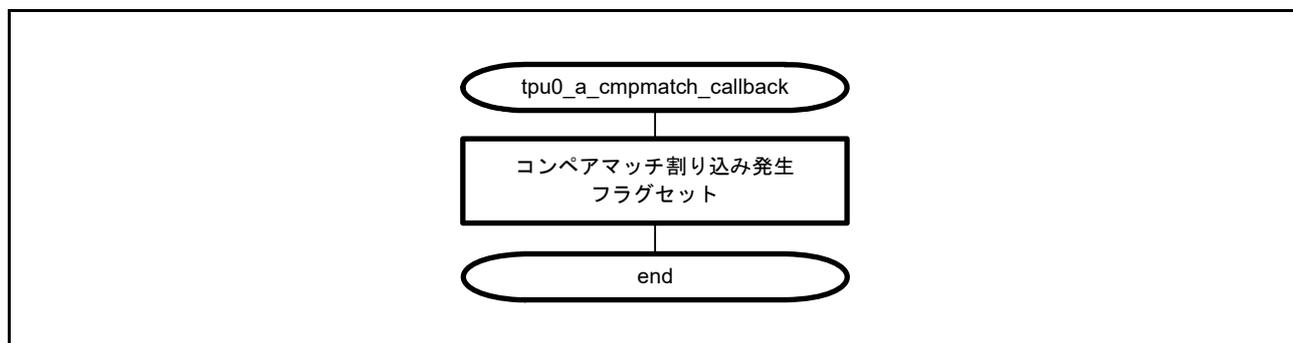


図 6.4 tpu0_a_cmpmatch_callback 処理

6.10.3 tpu0_v_callback 処理

図 6.5 に tpu0_v_callback のフローチャートを示します。

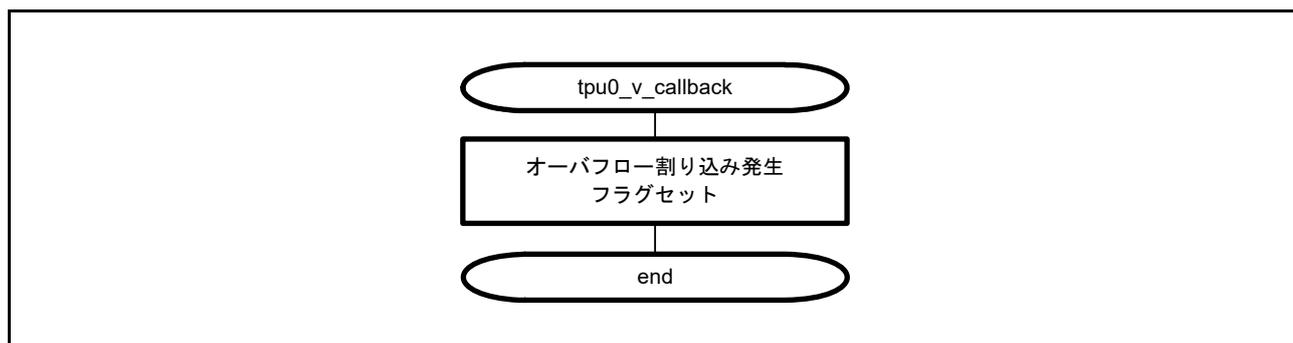


図 6.5 tpu0_v_callback 処理

6.10.4 R_TPUA_Open 処理

図 6.6 に R_TPUA_Open 処理のフローチャートを示します。

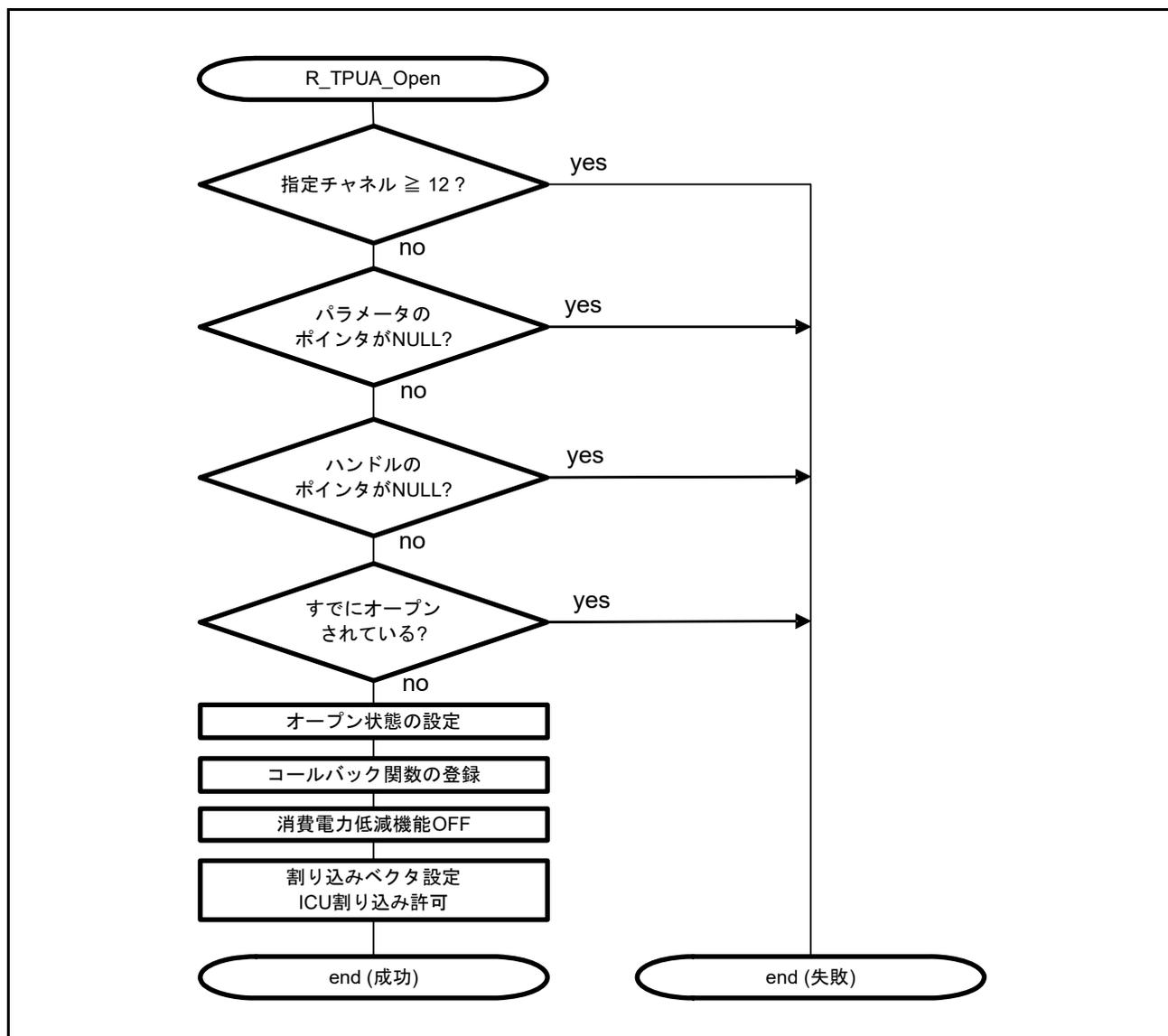


図 6.6 R_TPUA_Open 処理

6.10.5 R_TPUA_Control 処理

図 6.7、図 6.8 に R_TPUA_Control のフローチャートを示します。

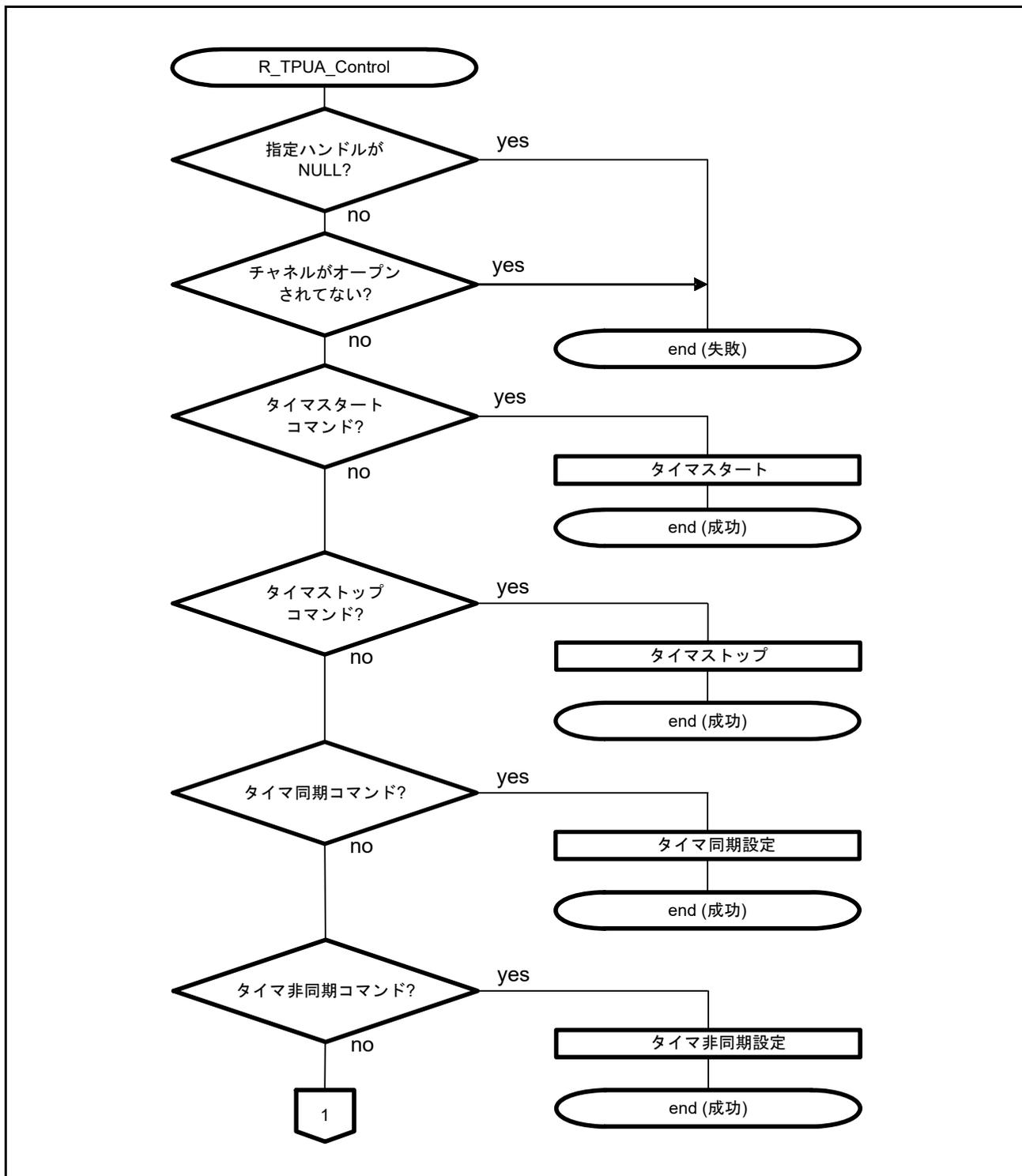


図 6.7 R_TPUA_Control 処理

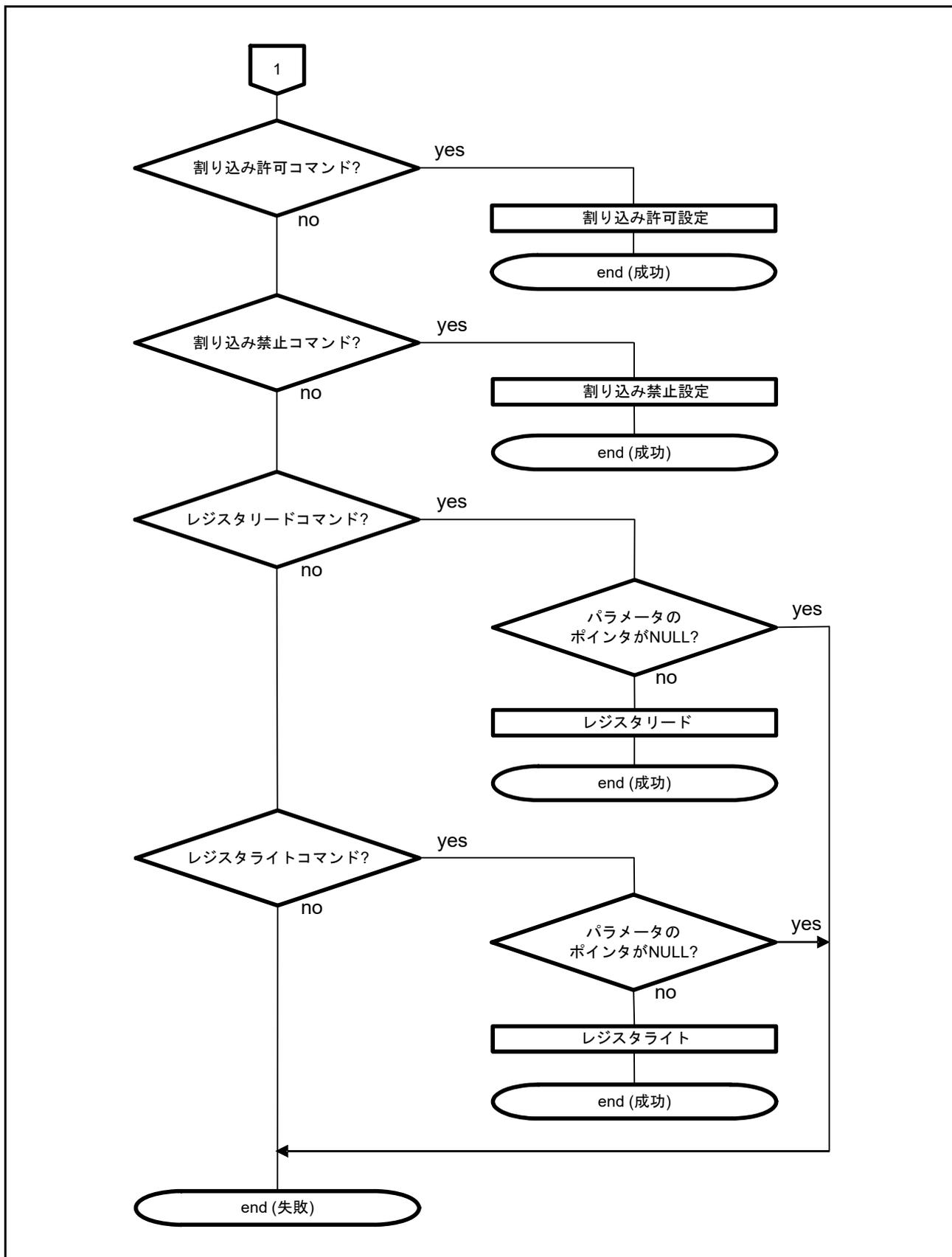


図 6.8 R_TPUA_Control 処理

6.10.6 R_TPUA_Close 処理

図 6.9 に R_TPUA_Close のフローチャートを示します。

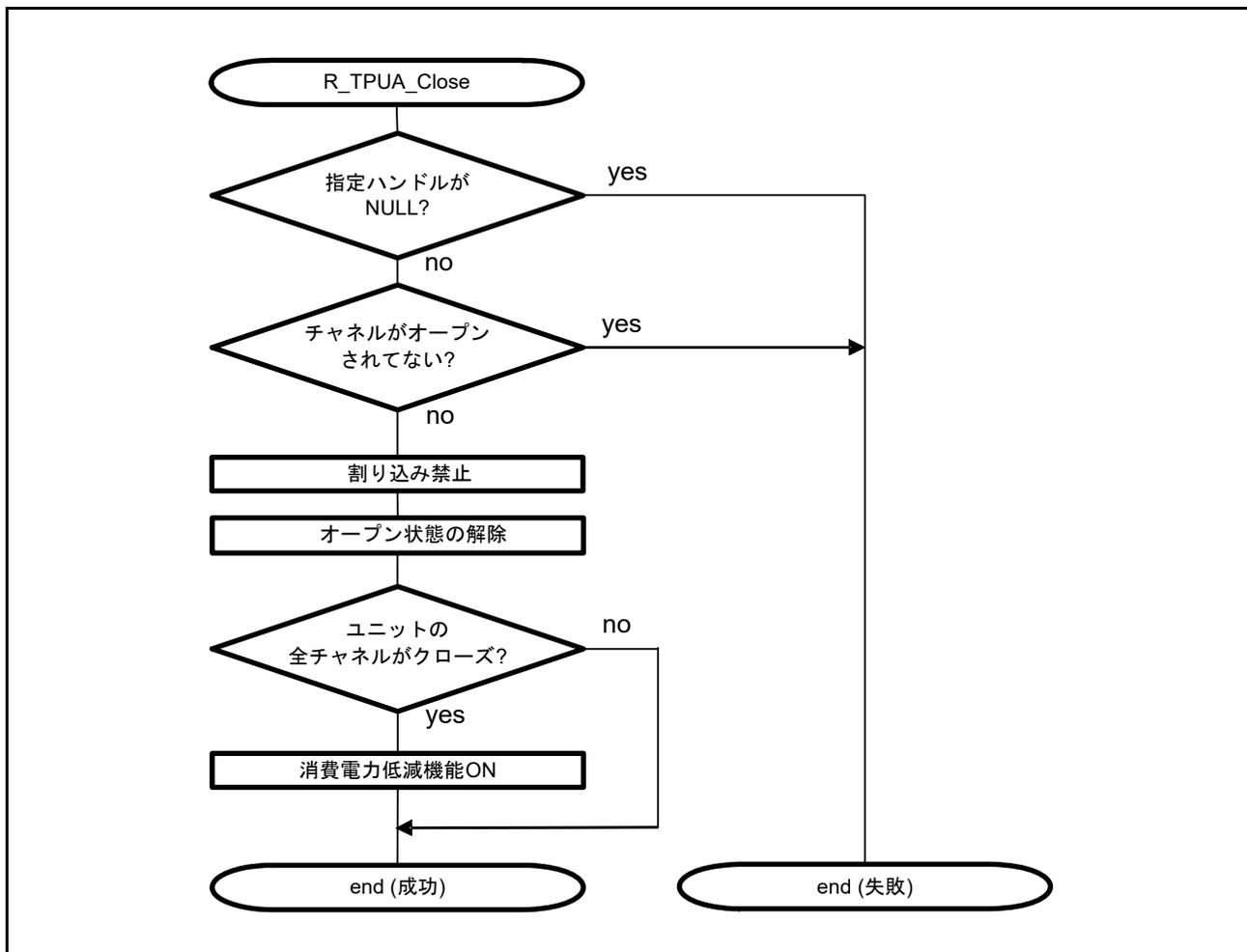


図 6.9 R_TPUA_Close 処理

6.10.7 R_TPUA_GetVersion 処理

図 6.10 に R_TPUA_GetVersion のフローチャートを示します。

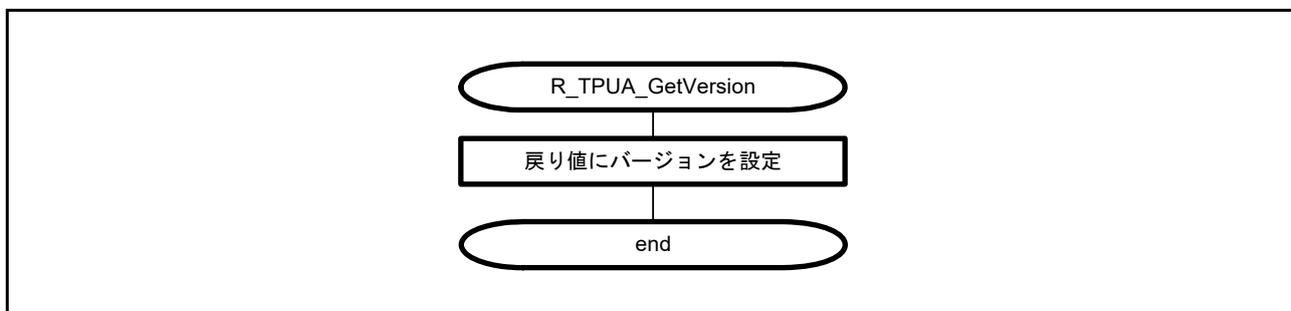


図 6.10 R_TPUA_GetVersion 処理

6.10.8 R_PPG_Open 処理

図 6.11 に R_PPG_Open 処理のフローチャートを示します。

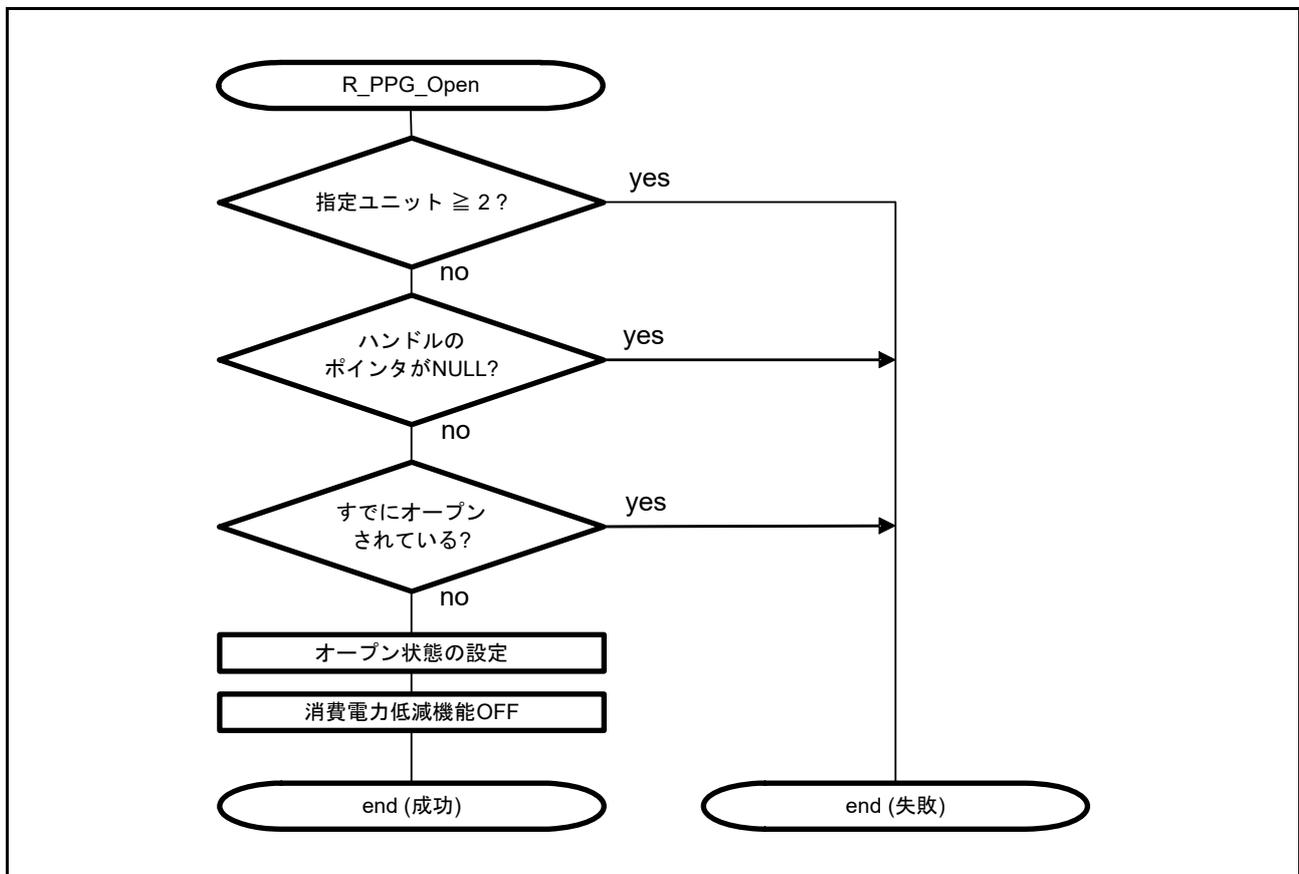


図 6.11 R_PPG_Open 処理

6.10.9 R_PPG_Control 処理

図 6.12 に R_PPG_Control のフローチャートを示します。

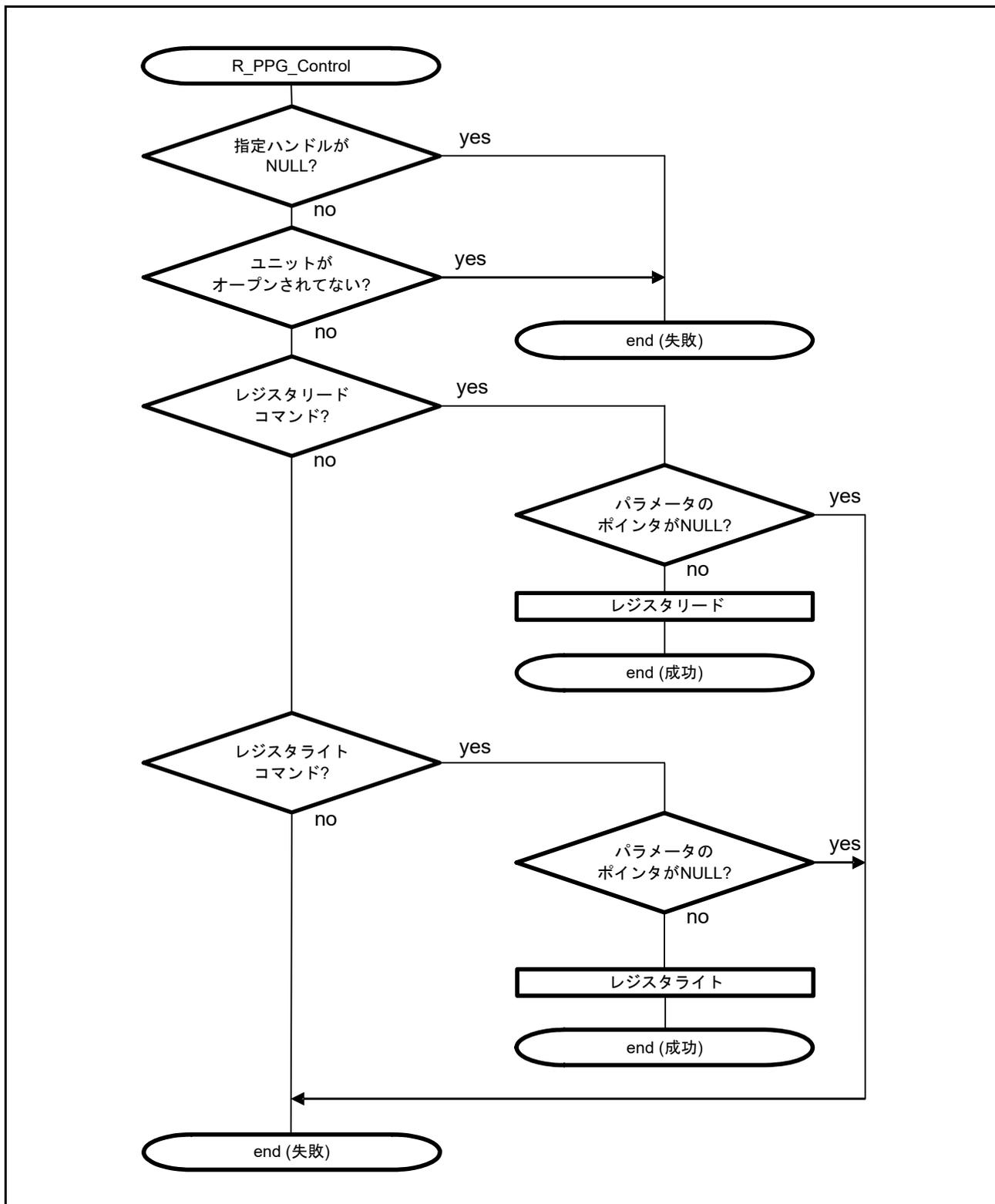


図 6.12 R_PPG_Control 処理

6.10.10 R_PPG_Close 処理

図 6.13 に R_PPG_Close のフローチャートを示します。

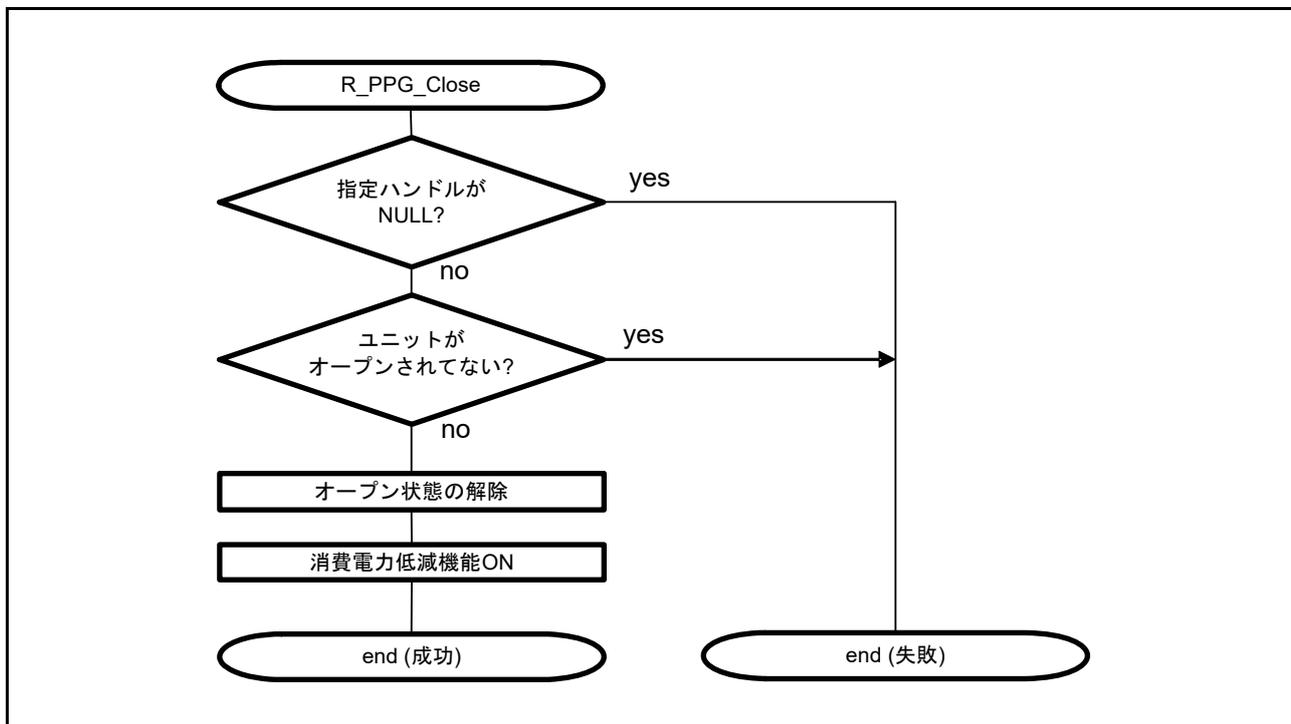


図 6.13 R_PPG_Close 処理

6.10.11 R_PPG_GetVersion 処理

図 6.14 に R_PPG_GetVersion のフローチャートを示します。

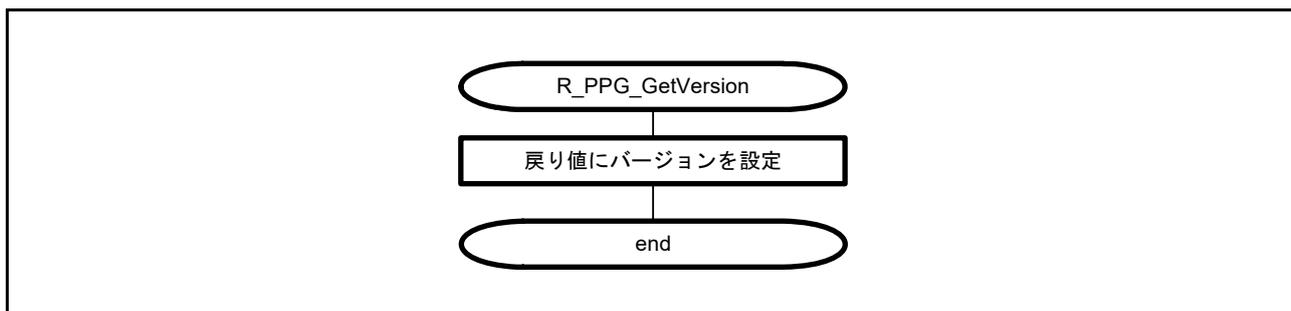


図 6.14 R_PPG_GetVersion 処理

6.11 R_TPUA_Control コマンド一覧

R_TPUA_Control 関数で使用するコマンド一覧を以下に示します。

表6.19 R_TPUA_Controlコマンド一覧

コマンド	概要
TPUA_CMD_TIMER_START	指定チャンネルのタイマ動作を開始します。注1
TPUA_CMD_TIMER_STOP	指定チャンネルのタイマ動作を停止します。注1
TPUA_CMD_TIMER_SYNC	指定チャンネルのタイマを同期動作に指定します。注1
TPUA_CMD_TIMER_ASYNC	指定チャンネルのタイマを単独（非同期）動作に指定します。注1
TPUA_CMD_REG_READ	指定チャンネルのレジスタ値を読み出します。注2
TPUA_CMD_REG_WRITE	指定チャンネルのレジスタへ指定値を書き込みます。注3
TPUA_CMD_INTR_A_ENABLE	指定チャンネルのTGRA割り込みを許可します。注1
TPUA_CMD_INTR_A_DISABLE	指定チャンネルのTGRA割り込みを禁止します。注1
TPUA_CMD_INTR_B_ENABLE	指定チャンネルのTGRB割り込みを許可します。注1
TPUA_CMD_INTR_B_DISABLE	指定チャンネルのTGRB割り込みを禁止します。注1
TPUA_CMD_INTR_C_ENABLE	指定チャンネルのTGRC割り込みを許可します。注1
TPUA_CMD_INTR_C_DISABLE	指定チャンネルのTGRC割り込みを禁止します。注1
TPUA_CMD_INTR_D_ENABLE	指定チャンネルのTGRD割り込みを許可します。注1
TPUA_CMD_INTR_D_DISABLE	指定チャンネルのTGRD割り込みを禁止します。注1
TPUA_CMD_INTR_V_ENABLE	指定チャンネルのオーバフロー割り込みを許可します。注1
TPUA_CMD_INTR_V_DISABLE	指定チャンネルのオーバフロー割り込みを禁止します。注1
TPUA_CMD_INTR_U_ENABLE	指定チャンネルのアンダフロー割り込みを許可します。注1
TPUA_CMD_INTR_U_DISABLE	指定チャンネルのアンダフロー割り込みを禁止します。注1

- 注1. 指定チャンネルは、R_TPUA_Control関数の第1引数にて指定します。
TPUA_CMD_REG_WRITEコマンドにて同一動作を指定することが可能です。
- 注2. 指定チャンネルは、R_TPUA_Control関数の第1引数にて指定します。
読み出した値は、R_TPUA_Control関数の第3引数に格納されます。
- 注3. 指定チャンネルは、R_TPUA_Control関数の第1引数にて指定します。
書き込む指定値は、R_TPUA_Control関数の第3引数に設定します。

6.11.1 TPUA_CMD_TIMER_START

TPUA_CMD_TIMER_START

概要	TPUa タイマのカウンタ動作開始
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマのカウンタ動作を開始します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.2 TPUA_CMD_TIMER_STOP

TPUA_CMD_TIMER_STOP

概要	TPUa タイマのカウンタ動作停止
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマのカウンタ動作を停止します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.3 TPUA_CMD_TIMER_SYNC

TPUA_CMD_TIMER_SYNC

概要	TPUa タイマの同期動作指定
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの同期動作を指定します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.4 TPUA_CMD_TIMER_ASYNC

TPUA_CMD_TIMER_ASYNC

概要	TPUa タイマの単独（非同期）動作指定
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの単独動作を指定します。注1
パラメータ	なし
リターン値	なし
補足	—

- 注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.5 TPUA_CMD_REG_READ

TPUA_CMD_REG_READ

概要	TPUa レジスタ読み出し	
ヘッダ	r_tpua_rzt1_if.h	
説明	tpua_reg_t 構造体メンバの reg_flag により指定されたレジスタの読み出し値を、tpua_reg_t 構造体のメンバに格納します。 TPUA_CMD_REG_READ コマンド実行前に reg_flag で読み出すレジスタを指定し、コマンド実行後に tpua_reg_t のメンバを参照することでレジスタ値の読み出しが可能です。	
パラメータ	tpua_reg_t uint32_t reg_flag	以下のパラメータにより読み出すレジスタを指定します。 <ul style="list-style-type: none"> • TPUA_TCR_REG : TCR レジスタを指定 • TPUA_TMDR_REG : TMDR レジスタを指定 • TPUA_TIORH_REG : TIORH レジスタを指定 • TPUA_TIOR_REG : TIOR レジスタを指定 • TPUA_TIORL_REG : TIORL レジスタを指定 • TPUA_TIER_REG : TIER レジスタを指定 • TPUA_TSR_REG : TSR レジスタを指定 • TPUA_TCNT_REG : TCNT レジスタを指定 • TPUA_TGRA_REG : TGRA レジスタを指定 • TPUA_TGRB_REG : TGRB レジスタを指定 • TPUA_TGRC_REG : TGRC レジスタを指定 • TPUA_TGRD_REG : TGRD レジスタを指定 • TPUA_TSTRA_REG : TSTRA レジスタを指定 • TPUA_TSTRB_REG : TSTRB レジスタを指定 • TPUA_TSYRA_REG : TSYRA レジスタを指定 • TPUA_TSYRB_REG : TSYRB レジスタを指定 • TPUA_NFCR_REG : NFCR レジスタを指定 • TPUA_PWMFBSLR_REG : PWMFBSLR レジスタを指定 • TPUA_ALL_REG : 全レジスタを指定 <p>注. 複数レジスタを指定する場合は、上記パラメータを OR にて指定します。</p>
リターン値	TPUA_SUCCESS	: 成功 : レジスタリード成功

補足 レジスタ詳細に関しては、RZ/T1グループ ユーザーズマニュアル ハードウェア編 (R01UH0483JJ) を参照してください。

レジスタリードの実施例

・ TSR レジスタ、TCNT レジスタをリードする場合。

⇒ R_TPUA_Control 関数の第3引数の regdata.reg_flag に、リードしたいレジスタをパラメータで指定する。コマンド実行後に regdata メンバにリード値が格納されている。

```
regdata.reg_flag = TPUA_TSR_REG | TPUA_TCNT_REG;
```

```
R_TPUA_Control(handle, TPUA_CMD_REG_READ, &regdata)
```

6.11.6 TPUA_CMD_REG_WRITE

TPUA_CMD_REG_WRITE

概要	TPUa レジスタ読み出し	
ヘッダ	r_tpua_rzt1_if.h	
説明	tpua_reg_t 構造体メンバの reg_flag により指定されたレジスタに、tpua_reg_t 構造体メンバで指定された値を書き込みます。 TPUA_CMD_REG_WRITE コマンド実行前に reg_flag に書き込むレジスタを指定し、同時に tpua_reg_t のメンバに書き込む値を設定してコマンドを実行することで、レジスタへの書き込みが可能です。	
パラメータ	tpua_reg_t uint32_t reg_flag	以下のパラメータにより書き込むレジスタを指定します。

- TPUA_TCR_REG : TCR レジスタを指定
- TPUA_TMDR_REG : TMDR レジスタを指定
- TPUA_TIORH_REG : TIORH レジスタを指定
- TPUA_TIOR_REG : TIOR レジスタを指定
- TPUA_TIORL_REG : TIORL レジスタを指定
- TPUA_TIER_REG : TIER レジスタを指定
- TPUA_TSR_REG : TSR レジスタを指定
- TPUA_TCNT_REG : TCNT レジスタを指定
- TPUA_TGRA_REG : TGRA レジスタを指定
- TPUA_TGRB_REG : TGRB レジスタを指定
- TPUA_TGRC_REG : TGRC レジスタを指定
- TPUA_TGRD_REG : TGRD レジスタを指定
- TPUA_TSTRA_REG : TSTRA レジスタを指定
- TPUA_TSTRB_REG : TSTRB レジスタを指定
- TPUA_TSYRA_REG : TSYRA レジスタを指定
- TPUA_TSYRB_REG : TSYRB レジスタを指定
- TPUA_NFCR_REG : NFCR レジスタを指定
- TPUA_PWMFBSLR_REG : PWMFBSLR レジスタを指定
- TPUA_ALL_REG : 全レジスタを指定

注. 複数レジスタを指定する場合は、上記パラメータを OR にて指定します。

uint8_t tcr_reg	TCR レジスタに書き込む値を設定します。
uint8_t tmdr_reg	TMDR レジスタに書き込む値を設定します。
uint8_t tiorh_reg	TIORH レジスタに書き込む値を設定します。
uint8_t tior_reg	TIOR レジスタに書き込む値を設定します。
uint8_t tiorl_reg	TIORL レジスタに書き込む値を設定します。
uint8_t tier_reg	TIER レジスタに書き込む値を設定します。
uint8_t tsr_reg	TSR レジスタに書き込む値を設定します。

```

uint16_t tcnt_reg  TCNT レジスタに書き込む値を設定します。
uint16_t tgra_reg  TGRA レジスタに書き込む値を設定します。
uint16_t tgrb_reg  TGRB レジスタに書き込む値を設定します。
uint16_t tgrc_reg  TGRC レジスタに書き込む値を設定します。
uint16_t tgrd_reg  TGRD レジスタに書き込む値を設定します。
uint8_t  tstra_reg TSTRA レジスタに書き込む値を設定します。
uint8_t  tstrb_reg TSTRB レジスタに書き込む値を設定します。
uint8_t  tsyra_reg TSYRA レジスタに書き込む値を設定します。
uint8_t  tsyrb_reg TSYRB レジスタに書き込む値を設定します。
uint8_t  nfcr_reg  TNFCR レジスタに書き込む値を設定します。
uint32_t
pwmfbslr_reg      PWMFBSLR レジスタに書き込む値を設定します。

```

リターン値 TPUA_SUCCESS : 成功 : レジスタライト成功
 補足 レジスタ詳細に関しては、RZ/T1 グループ ユーザーズマニュアル ハードウェア編 (R01UH0483JJ) を参照してください。

レジスタライトの指定方法例

・ TCR レジスタに 0x01、TMDR レジスタに 0x02 をライトする場合。

⇒ R_TPUA_Control 関数の第 3 引数の regdata.reg_flag に、ライトしたいレジスタをパラメータで指定し、ライトしたい値を regdata メンバに設定する。

```
regdata.reg_flag = TPUA_TCR_REG | TPUA_TMDR_REG;
```

```
regdata.tcr_reg = 0x01;
```

```
regdata.tmdr_reg = 0x02;
```

```
R_TPUA_Control(handle, TPUA_CMD_REG_WRITE, &regdata)
```

6.11.7 TPUA_CMD_INTR_A_ENABLE

TPUA_CMD_INTR_A_ENABLE

概要	TPUa タイマの TGRA 割り込み許可
ヘッダ	r_tpuarzt1_if.h
説明	TPUa タイマの TGRA 割り込みを許可します。注 1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.8 TPUA_CMD_INTR_A_DISABLE

TPUA_CMD_INTR_A_DISABLE

概要	TPUa タイマの TGRA 割り込み禁止
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの TGRA 割り込みを禁止します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.9 TPUA_CMD_INTR_B_ENABLE

TPUA_CMD_INTR_B_ENABLE

概要	TPUa タイマの TGRB 割り込み許可
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの TGRB 割り込みを許可します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.10 TPUA_CMD_INTR_B_DISABLE

TPUA_CMD_INTR_B_DISABLE

概要	TPUa タイマの TGRB 割り込み禁止
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの TGRB 割り込みを禁止します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.11 TPUA_CMD_INTR_C_ENABLE

TPUA_CMD_INTR_C_ENABLE

概要	TPUa タイマの TGRC 割り込み許可
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの TGRC 割り込みを許可します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.12 TPUA_CMD_INTR_C_DISABLE

TPUA_CMD_INTR_C_DISABLE

概要	TPUa タイマの TGRC 割り込み禁止
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの TGRC 割り込みを禁止します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.13 TPUA_CMD_INTR_D_ENABLE

TPUA_CMD_INTR_D_ENABLE

概要	TPUa タイマの TGRD 割り込み許可
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの TGRD 割り込みを許可します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.14 TPUA_CMD_INTR_D_DISABLE

TPUA_CMD_INTR_D_DISABLE

概要	TPUa タイマの TGRD 割り込み禁止
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマの TGRD 割り込みを禁止します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.15 TPUA_CMD_INTR_V_ENABLE

TPUA_CMD_INTR_V_ENABLE

概要	TPUa タイマのオーバフロー割り込み許可
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマのオーバフロー割り込みを許可します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.16 TPUA_CMD_INTR_V_DISABLE

TPUA_CMD_INTR_V_DISABLE

概要	TPUa タイマのオーバフロー割り込み禁止
ヘッダ	r_tpua_rzt1_if.h
説明	TPUa タイマのオーバフロー割り込みを禁止します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.17 TPUA_CMD_INTR_U_ENABLE

TPUA_CMD_INTR_U_ENABLE

概要	TPUa タイマのアンダフロー割り込み許可
ヘッダ	r_tlua_rzt1_if.h
説明	TPUa タイマのアンダフロー割り込みを許可します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.11.18 TPUA_CMD_INTR_U_DISABLE

TPUA_CMD_INTR_U_DISABLE

概要	TPUa タイマのアンダフロー割り込み禁止
ヘッダ	r_tlua_rzt1_if.h
説明	TPUa タイマのアンダフロー割り込みを禁止します。注1
パラメータ	なし
リターン値	なし
補足	—

注1. 各コマンドは、第1引数にて指定されたチャンネルに対してのみ機能するため、1チャンネル毎の動作しか指定できません。そのため、複数チャンネルを同時に機能させたい場合（カスケード動作等）には、TPUA_CMD_REG_WRITEコマンドにより直接レジスタを操作する必要があります。

6.12 R_PPG_Control コマンド一覧

R_PPG_Control 関数で使用するコマンド一覧を以下に示します。

表6.20 R_PPG_Controlコマンド一覧

列挙型定義名	内容
PPG_CMD_REG_READ	指定チャンネルのレジスタ値を読み出します。注1
PPG_CMD_REG_WRITE	指定チャンネルのレジスタへ指定値を書き込みます。注2

- 注1. 指定ユニットは、R_PPG_Control関数の第1引数にて指定します。
読み出した値は、R_PPG_Control関数の第3引数に格納されます。
- 注2. 指定チャンネルは、R_PPG_Control関数の第1引数にて指定します。
書き込む指定値は、R_PPG_Control関数の第3引数に設定します。

6.12.1 PPG_CMD_REG_READ

PPG_CMD_REG_READ

概要	PPG レジスタ読み出し	
ヘッダ	r_ppg_rzt1_if.h	
説明	ppg_reg_t 構造体メンバの reg_flag により指定されたレジスタの読み出し値を、ppg_reg_t 構造体のメンバに格納します。 PPG_CMD_REG_READ コマンド実行前に reg_flag で読み出すレジスタを指定し、コマンド実行後に ppg_reg_t のメンバを参照することでレジスタ値の読み出しが可能です。	
パラメータ	ppg_reg_t uint32_t reg_flag	以下のパラメータにより読み出すレジスタを指定します。 <ul style="list-style-type: none"> • PPG_PTRSLR_REG : PTRSLR レジスタを指定 • PPG_NDER_REG : NDER レジスタを指定 • PPG_PODR_REG : PODR レジスタを指定 • PPG_NDR_REG : NDR レジスタを指定 • PPG_PCR_REG : PCR レジスタを指定 • PPG_PMR_REG : PMR レジスタを指定 • PPG_ALL_REG : 全レジスタを指定 <p style="text-align: center;">注 . 複数レジスタを指定する場合は、上記パラメータを OR にて指定します。</p>
リターン値	PPG_SUCCESS	: 成功 : レジスタリード成功
補足	レジスタ詳細に関しては、RZ/T1 グループ ユーザーズマニュアル ハードウェア編 (R01UH0483JJ) を参照してください。	

レジスタリードの実施例

・ PTRSLR レジスタ、PODR レジスタをリードする場合。

⇒ R_PPG_Control 関数の第 3 引数の regdata.reg_flag に、リードしたいレジスタをパラメータで指定する。コマンド実行後に regdata メンバにリード値が格納されている。

```
regdata.reg_flag = PPG_PTRSLR_REG | PPG_PODR_REG;
R_PPG_Control(handle, PPG_CMD_REG_READ, &regdata)
```

6.12.2 PPG_CMD_REG_WRITE

PPG_CMD_REG_READ

概要	PPG レジスタ読み出し	
ヘッダ	r_ppg_rzt1_if.h	
説明	ppg_reg_t 構造体メンバの reg_flag により指定されたレジスタに、ppg_reg_t 構造体メンバで指定された値を書き込みます。 PPG_CMD_REG_WRITE コマンド実行前に reg_flag に書き込むレジスタを指定し、同時に ppg_reg_t のメンバに書き込む値を設定してコマンドを実行することで、レジスタへの書き込みが可能です。	
パラメータ	ppg_reg_t uint32_t reg_flag	以下のパラメータにより書き込むレジスタを指定します。

- PPG_PTRSLR_REG : PTRSLR レジスタを指定
- PPG_NDER_REG : NDER レジスタを指定
- PPG_PODR_REG : PODR レジスタを指定
- PPG_NDR_REG : NDR レジスタを指定
- PPG_PCR_REG : PCR レジスタを指定
- PPG_PMR_REG : PMR レジスタを指定
- PPG_ALL_REG : 全レジスタを指定

注． 複数レジスタを指定する場合は、上記パラメータを OR にて指定します。

uint8_t ptrslr_reg	PTRSLR レジスタに書き込む値を設定します。
uint16_t nder_reg	NDERH/L レジスタに書き込む値を設定します。
uint16_t podr_reg	PODRH/L レジスタに書き込む値を設定します。
uint16_t ndr_reg	NDRH/L レジスタに書き込む値を設定します。
uint8_t pcr_reg	PCR レジスタに書き込む値を設定します。
uint8_t pmr_reg	PMR レジスタに書き込む値を設定します。

リターン値 PPG_SUCCESS : 成功 : レジスタライト成功

補足 レジスタ詳細に関しては、RZ/T1 グループ ユーザーズマニュアル ハードウェア編 (R01UH0483JJ) を参照してください。

レジスタリード／ライトの指定方法

・ PTRSLR レジスタに 0x01、NDR レジスタに 0x0002 をライトする場合。

⇒ R_PPG_Control 関数の第 3 引数の regdata.reg_flag に、ライトしたいレジスタをパラメータで指定し、ライトしたい値を regdata メンバに設定する。

```
regdata.reg_flag = PPG_PTRSLR_REG | PPG_NDR_REG;
```

```
regdata.tcr_reg = 0x01;
```

```
regdata.tmdr_reg = 0x0002;
```

```
R_PPG_Control(handle, PPG_CMD_REG_WRITE, &regdata)
```

7. 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。

8. 参考ドキュメント

- ユーザーズマニュアル：ハードウェア

RZ/T1 グループ ユーザーズマニュアルハードウェア編

(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。)

RZ/T1 Evaluation Board RTK7910022C00000BR ユーザーズマニュアル

(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。)

- ユーザーズマニュアル：開発環境

IAR 統合開発環境 (IAR Embedded Workbench® for Arm) に関しては、IAR ホームページから入手してください。

(最新版を IAR 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/contact/>

改訂記録

TPUa/PPG サンプルプログラム アプリケーションノート

Rev.	発行日	改訂内容	
		ページ	ポイント
0.10	2015.03.06	—	初版発行
1.00	2015.04.10	—	Web掲載に際しRevのみ変更
1.10	2015.07.06	2. 動作環境	
		5	表2.1 動作環境 統合開発環境 表記一部修正、追加
		6. ソフトウェア説明	
		11	6.2.4 説明文 参照を追加
		11	表6.2 タイトル、サイズを一部修正
		12	表6.3 追加
1.20	2015.12.03	2. 動作環境	
		5	表2.1 動作環境 統合開発環境 一部修正
		6. ソフトウェア説明	
		12	表6.4 追加
1.30	2017.04.05	2. 動作環境	
		5	表2.1 動作環境 統合開発環境の内容変更
		6. ソフトウェア説明	
1.40	2018.06.07	2. 動作環境	
		5	表2.1 動作環境 統合開発環境の内容変更
		8. 参考ドキュメント	
		55	IAR 統合開発環境名変更

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. 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（予約領域）のアクセス禁止

【注意】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、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）がありません。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、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っていると、内部 ROM、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。型名が異なる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、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、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て、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いたします。
 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)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>