

要旨

本アプリケーションノートでは、ウォッチドッグタイマ (WDTA) のリセット制御を行うサンプルプログラムについて説明します。

サンプルプログラムの特長を以下に示します。

- ウォッチドッグタイマ (WDTA) 動作後、コンペアマッチタイマ (CMT) による定周期 (223.7ms) でリフレッシュ動作させます。
- 外部割り込み (IRQ5) によるソフトウェアウエイトの発生でリフレッシュ動作が停止し、リセットが発生します。リセット発生後に LED2 が点灯します。

動作確認デバイス

RZ/T1 グループ

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

目次

1.	仕様	3
2.	動作環境	4
3.	関連アプリケーションノート	5
4.	周辺機能説明	6
5.	ハードウェア説明	7
5.1	ハードウェア構成例	7
5.2	使用端子一覧	7
6.	ソフトウェア説明	8
6.1	動作概要	8
6.1.1	プロジェクト設定	9
6.2	メモリマップ	9
6.2.1	サンプルプログラムのセクション配置	9
6.2.2	MPU の設定	9
6.2.3	例外処理ベクタテーブル	9
6.3	使用割り込み一覧	10
6.4	固定幅整数一覧	10
6.5	定数／エラーコード一覧	10
6.6	構造体／共用体／列挙型一覧	11
6.7	関数一覧	13
6.8	関数仕様	13
6.8.1	main	13
6.8.2	wdt0_init	13
6.8.3	R_WDT_Open	14
6.8.4	R_WDT_Control	15
6.8.5	R_IRQ9_isr	15
6.8.6	R_IRQ21_isr	16
6.9	フローチャート	17
6.9.1	メイン処理	17
6.9.2	WDT0 初期化処理	18
6.9.3	WDT オープン関数	19
6.9.4	WDT コントロール関数	20
6.9.5	IRQ9 割り込み（IRQ 端子割り込み 5）処理	22
6.9.6	IRQ21 割り込み（コンペアマッチタイマ ch0 割り込み）処理	22
7.	サンプルプログラム	23
8.	参考ドキュメント	24

1. 仕様

表 1.1 に使用する周辺機能と用途を、図 1.1 に動作環境を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
クロック発生回路 (CPG)	CPUクロックおよび低速オンチップオシレータで使用
割り込みコントローラ (ICUA)	外部割り込み入力端子 (IRQ5)、コンペアマッチタイマ割り込み (CMIO) で使用
コンペアマッチタイマ (CMT)	コンペアマッチタイマの周期カウント動作で使用
ウォッチドッグタイマ (WDTA)	タイムアウト期間546.1msで動作し、定周期でリフレッシュ制御する。WDTAカウンタのアンダーフロー/リフレッシュエラーによりECMリセットが発生
エラーコントロールモジュール (ECM)	ERROROUT#端子の初期化、およびウォッチドッグタイマ (WDTA) のアンダーフロー/リフレッシュエラーを使用
汎用入出力ポート	LEDの点灯および消灯のための端子制御に使用

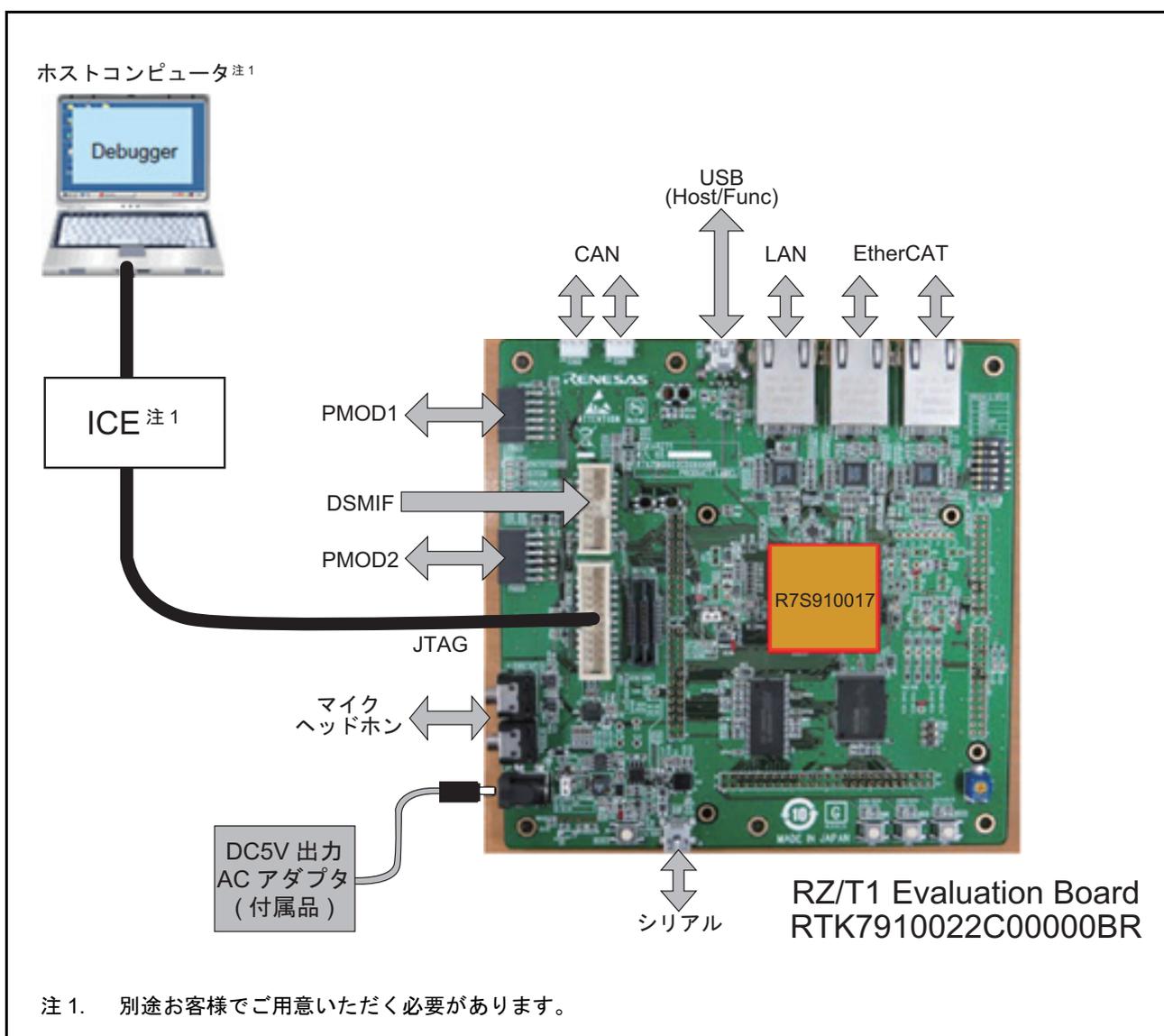


図 1.1 動作環境

2. 動作環境

本アプリケーションノートのサンプルプログラムは、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
- RZ/T1 グループ コンペアマッチタイマ (CMT)

4. 周辺機能説明

クロック発生回路 (CPG)、ウォッチドッグタイマ (WDTA)、コンペアマッチタイマ (CMT)、割り込みコントローラ (ICUA)、エラーコントロールモジュール (ECM)、汎用入出力ポート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

図 5.1 にハードウェア構成例を示します。

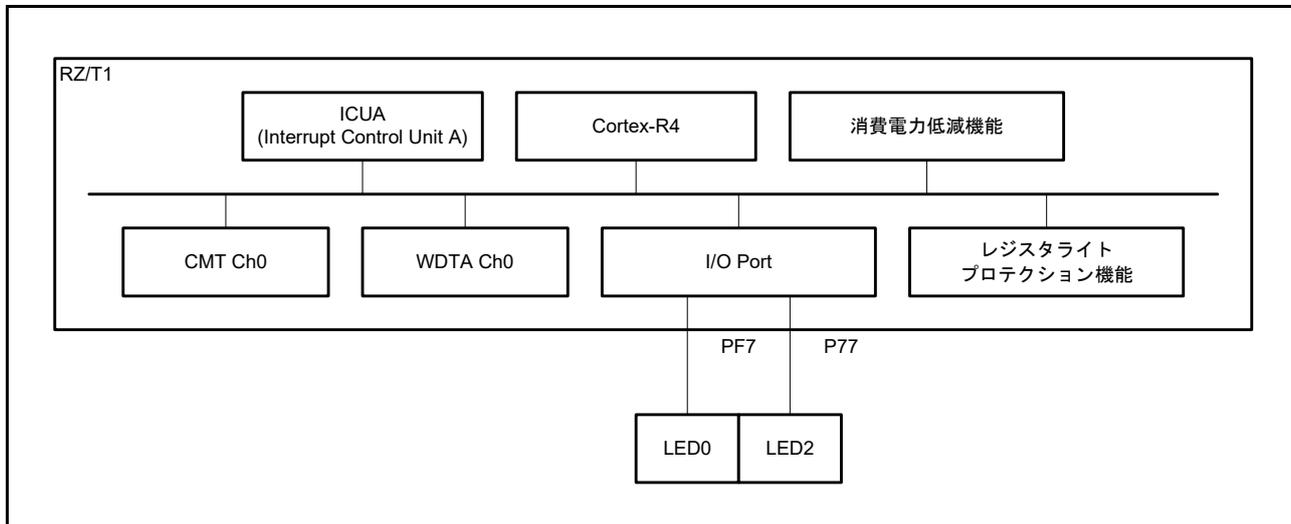


図 5.1 ハードウェア構成例

5.2 使用端子一覧

表 5.1 に使用端子と機能を示します。

表 5.1 使用端子と機能

端子名	入出力	内容
MD0	入力	動作モードの選択 MD0 = "L"、MD1 = "L"、MD2 = "L" (SPI ブートモード) MD0 = "L"、MD1 = "H"、MD2 = "L" (16ビットバスブートモード)
MD1	入力	
MD2	入力	
IRQ5	入力	SW2 (IRQ 端子割り込み)
PF7	出力	LED0の点灯および消灯
P77	出力	LED2の点灯および消灯

6. ソフトウェア説明

6.1 動作概要

本サンプルプログラムでは、ウォッチドッグタイマ (WDTA) の初期設定を行い、以降、コンペアマッチタイマのインターバル割り込みで定期的 (227.3ms) にリフレッシュ動作を行います。

SW2 を押下すると外部端子割り込み 5 が発生し、割り込み内でソフトウェアウエイト処理が実施されます。この間は CMT のインターバル割り込みが保留されるため、ウォッチドッグタイマ (WDTA) のリフレッシュ動作が停止します。そのため、ウォッチドッグタイマのカウンタ値がアンダーフローし、エラーコントロールモジュール (ECM) にエラー通知を行い、ECM リセットが発生します。そしてリセット解除後のリセット判定処理にて LED2 が点灯します。

本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 にタイミング図を示します。

表 6.1 動作概要

機能	概要
チャンネル	チャンネル0
クロック	PCLK/2048 (= 75MHz/2048)
タイムアウト	16384cycle (= 447.4ms)
ウィンドウ	開始 : 75% 終了 : 25%
ECMエラー通知	許可

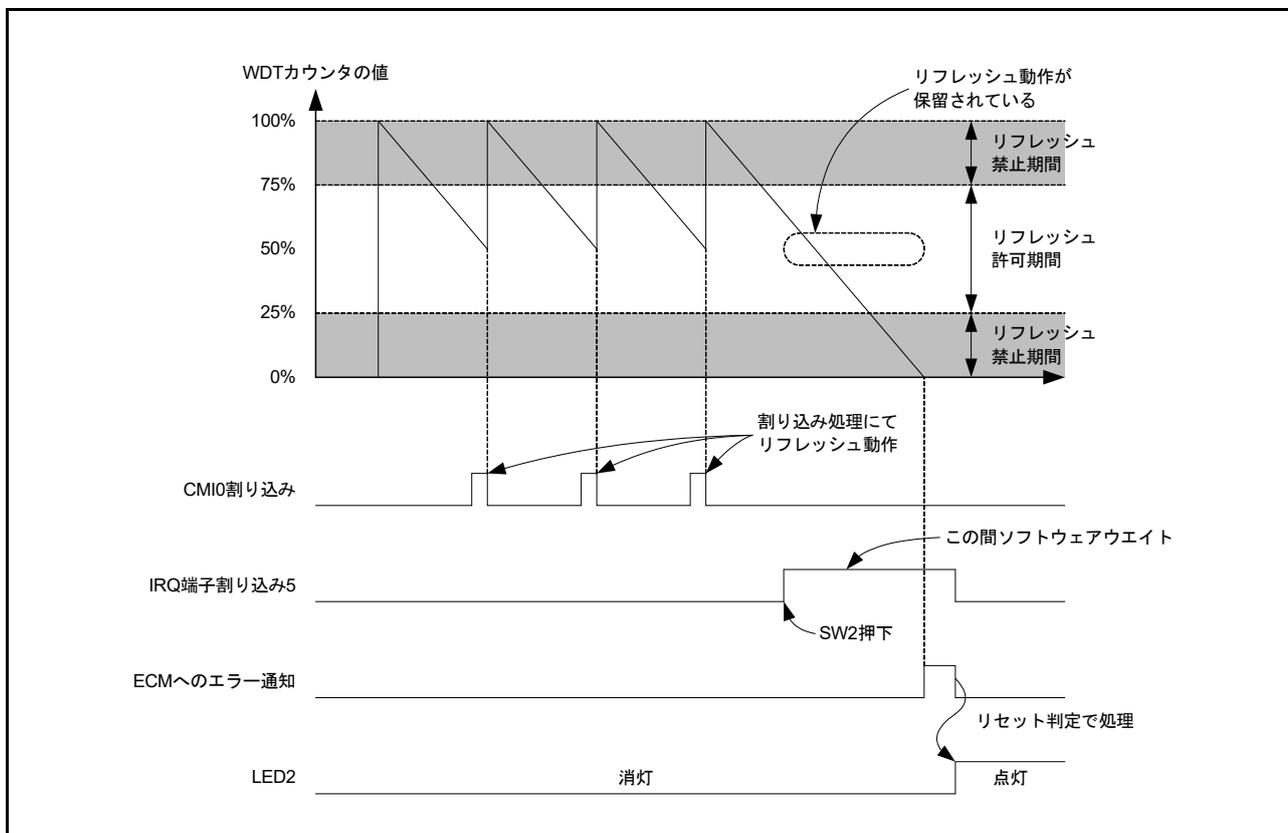


図 6.1 タイミング図

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、アプリケーションノート「RZ/T1 グループ初期設定」に記載しています。

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、アプリケーションノート「RZ/T1 グループ初期設定」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）、スキップローディング機能を使用後のセクション配置（実行ビュー）は、アプリケーションノート「RZ/T1 グループ初期設定」に記載しています。

6.2.2 MPU の設定

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

6.2.3 例外処理ベクタテーブル

例外処理のベクタテーブルは、アプリケーションノート「RZ/T1 グループ初期設定」に記載しています。

6.3 使用割り込み一覧

表 6.2 にサンプルプログラムで使用する割り込みを示します。

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
IRQ 端子割り込み5 (IRQ9)	0	SW2を押下すると、ウォッチドッグタイマがリフレッシュされずにアンダーフローし、ECMリセットが発生します。リセット解除後のリセット判定でLED2が点灯します。
コンペアマッチ割り込み_ch0 (CMIO)	15	インターバル周期 (周期は227.3ms) が発生するたび、ウォッチドッグタイマのリフレッシュ動作を行います。

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 にサンプルプログラムで使用する定数を示します。

表6.4 サンプルプログラムで使用する定数

定数名	設定値	内容
WDT_STAT_REFRESH_ERR_MASK	(0x8000)	WDTステータスからリフレッシュエラーフラグを取得するための定義です。
WDT_STAT_UNDERFLOW_ERR_MASK	(0x4000)	WDTステータスからアンダーフローフラグを取得するための定義です。
WDT_STAT_ERROR_MASK	(0xC000)	WDTステータスからリフレッシュエラーフラグ、アンダーフローフラグを取得するための定義です。
WDT_STAT_COUNTER_MASK	(0x3FFF)	WDTステータスからカウンタ値を取得するための定義です。
WDT_CFG_PARAM_CHECKING_ENABLE	(1)	WDTAのAPI関数でパラメータチェック許可の有効 (1) /無効 (0) を表します。

6.6 構造体／共用体／列挙型一覧

図 6.2 ～ 図 6.4 にサンプルプログラムで使用する構造体／共用体／列挙体を示します。

```
/* API ERROR RETURN CODES */
typedef enum e_wdt_err      // WDT API error codes
{
    WDT_SUCCESS=0,
    WDT_ERR_OPEN_IGNORED,    // The module has already been Open()ed
    WDT_ERR_INVALID_ARG,    // Argument is not valid for parameter
    WDT_ERR_NULL_PTR,       // Received null pointer or missing required argument
    WDT_ERR_NOT_OPENED     // Open function has not yet been called
} wdt_err_t;

typedef enum e_wdt_ch      // WDT channel
{
    WDT_CHANNEL_0=0,        // Ch0
    WDT_CHANNEL_1,         // Ch1
    WDT_CHANNEL_MAX
} wdt_ch_t;

/* Open() DEFINITIONS */
typedef enum e_wdt_timeout // WDT Time-Out Period
{
    WDT_TIMEOUT_1024 =0x0000u,    // 1024 (cycles)
    WDT_TIMEOUT_4096 =0x0001u,    // 4096 (cycles)
    WDT_TIMEOUT_8192 =0x0002u,    // 8192 (cycles)
    WDT_TIMEOUT_16384=0x0003u,    // 16,384 (cycles)
    WDT_NUM_TIMEOUTS
} wdt_timeout_t;
```

図 6.2 サンプルプログラムで使用する構造体／共用体／列挙体

```

typedef enum e_wdt_clock_div      // WDT Clock Division Ratio
{
    WDT_CLOCK_DIV_4  =0x0010u,    // WDTCLK/4
    WDT_CLOCK_DIV_64 =0x0040u,    // WDTCLK/64
    WDT_CLOCK_DIV_128=0x00F0u,    // WDTCLK/128
    WDT_CLOCK_DIV_512=0x0060u,    // WDTCLK/512
    WDT_CLOCK_DIV_2048=0x0070u,   // WDTCLK/2048
    WDT_CLOCK_DIV_8192=0x0080u    // WDTCLK/8192
} wdt_clock_div_t;

typedef enum e_wdt_window_end     // Window End Position
{
    WDT_WINDOW_END_75=0x0000u,    // 75%
    WDT_WINDOW_END_50=0x0100u,    // 50%
    WDT_WINDOW_END_25=0x0200u,    // 25%
    WDT_WINDOW_END_0 =0x0300u     // 0% (window end position is not specified)
} wdt_window_end_t;

typedef enum e_wdt_window_start  // Window Start Position
{
    WDT_WINDOW_START_25 =0x0000u, // 25%
    WDT_WINDOW_START_50 =0x1000u, // 50%
    WDT_WINDOW_START_75 =0x2000u, // 75%
    WDT_WINDOW_START_100=0x3000u  // 100% (window start position is not specified)
} wdt_window_start_t;

typedef enum e_wdt_timeout_control // Signal control when Time-out and Refresh error
{
    WDT_ERROR_ENABLE =0x00u, // Error output is enabled
    WDT_ERROR_DISABLE=0x80u  // Error output is disabled
} wdt_timeout_control_t;

```

図 6.3 サンプルプログラムで使用する構造体/共用体/列挙型

```

typedef struct st_wdt_config      // WDT configuration options used in Open function
{
    wdt_timeout_t      timeout;      // Time-out period
    wdt_clock_div_t    wdtdclk_div;  // WDT clock division ratio
    wdt_window_start_t window_start; // Window start position
    wdt_window_end_t   window_end;   // Window end position
    wdt_timeout_control_t timeout_control; // ERROR output when time-out
} wdt_config_t;

/* Control() DEFINITIONS */

typedef enum e_wdt_cmd           // Command used in Control and GetStatus function
{
    WDT_CMD_GET_STATUS,          // Get WDT status
    WDT_CMD_REFRESH_COUNTING,    // Refresh the counter
    WDT_CMD_NO_ACTION,
} wdt_cmd_t;

```

図 6.4 サンプルプログラムで使用する構造体/共用体/列挙型

6.7 関数一覧

表 6.5 に関数一覧を示します。

表6.5 関数一覧

関数名	ページ番号
main	13
wdt0_init	13
R_WDT_Open	14
R_WDT_Control	15
R_IRQ9_isr	15
R_IRQ21_isr	16

6.8 関数仕様

6.8.1 main

main	
概 要	メイン処理
宣 言	int main(void)
説 明	ポート、ECM、CMT、ICU、WDT の初期設定を行い、CMT0 の動作を開始させます。 その後、メインループにて LED0 の点灯／消灯を繰り返します。
引 数	なし
リターン値	なし
補足	なし

6.8.2 wdt0_init

wdt0_init	
概 要	WDT0 初期化処理
宣 言	void wdt0_init(void)
説 明	WDT0 を初期化し、WDT0 のカウント動作を開始させます。
引 数	なし
リターン値	なし
補足	なし

6.8.3 R_WDT_Open

R_WDT_Open

概要	WDT オープン関数	
ヘッダ	r_wdt_if.h	
宣言	wdt_err_t R_WDT_Open (uint16_t channel, void * const p_cfg)	
説明	指定チャンネルの WDTA 関連レジスタを初期化し、WDT カウンタのオプションを設定します。	
引数	uint16_t channel	WDT チャンネルを指定します。 設定可能範囲 (0、1)
	void * const p_cfg	WDTA 関連レジスタに設定するデータ群を格納したポインタです。 タイムアウト時間 WDT_TIMEOUT_1024 WDT_TIMEOUT_4096 WDT_TIMEOUT_8192 WDT_TIMEOUT_16384 クロック分周比 WDT_CLOCK __ DIV_4 WDT_CLOCK __ DIV_64 WDT_CLOCK __ DIV_128 WDT_CLOCK __ DIV_512 WDT_CLOCK __ DIV_2048 WDT_CLOCK __ DIV_8192 ウィンドウ終了 WDT_WINDOW_END_75 WDT_WINDOW_END_50 WDT_WINDOW_END_25 WDT_WINDOW_END_0 ウィンドウ開始 WDT_WINDOW_START_25 WDT_WINDOW_START_50 WDT_WINDOW_START_75 WDT_WINDOW_START_100 ECM エラー通知 WDT_ERROR_ENABLE WDT_ERROR_DISABLE
リターン値	オープン関数の実行結果を返します。 WDT_SUCCESS : IWDT が初期化されました WDT_ERR_OPEN_IGNORED : モジュールはすでに開かれています WDT_ERR_INVALID_ARG : p_cfg 構造体の要素に無効な値が含まれています WDT_ERR_NULL_PTR : p_cfg ポインタが NULL です	
補足	r_wdt_config.h で定義される WDT_CFG_PARAM_CHECKING_ENABLE を 1 とすることで、引数パラメータのチェック処理を有効にします。	

6.8.4 R_WDT_Control

R_WDT_Control

概要	WDT コントロール関数	
ヘッダ	r_wdt_if.h	
宣言	wdt_err_t R_WDT_Control(uint16_t channel, wdt_cmd_t const cmd, uint16_t* p_status)	
説明	指定したチャンネルの WDT の状態読み出し、WDT のダウンカウンタをリフレッシュします。	
引数	uint16_t channel	WDT チャンネルを指定します。 設定可能範囲 (0、1)
	wdt_cmd_t const cmd	実行されるコマンドを指定します。 IWDT_CMD_GET_STATUS IWDT_CMD_REFRESH_COUNTING
	uint16_t* p_status	カウンタとステータスフラグの格納位置へのポインタ
リターン値	コントロール関数の実行結果を返します。 WDT_SUCCESS : コマンドは正常に完了しました WDT_ERR_INVALID_ARG : 引数の値が無効です WDT_ERR_NULL_PTR : p_status が NULL です WDT_ERR_NOT_OPEN : Open が読み出されていません	
補足	r_wdt_config.h で定義される WDT_CFG_PARAM_CHECKING_ENABLE を 1 とすることで、引数パラメータのチェック処理を有効にします。	

6.8.5 R_IRQ9_isr

R_IRQ9_isr

概要	IRQ9 割り込み (IRQ 端子割り込み 5) 処理
宣言	void R_IRQ9_isr(void)
説明	ソフトウェアウエイト処理 (約 3 秒) を実行します。この間にウォッチドッグタイマをアンダーフローさせ、ECM リセットを発生させます。 リセット解除後のリセット判定にて、LED2 を点灯させます。
引数	なし
リターン値	なし
補足	なし

6.8.6 R_IRQ21_isr

R_IRQ21_isr

概要	IRQ21 割り込み (コンペアマッチタイマ (CMI0)) 処理
宣言	void R_IRQ21_isr(void)
説明	ウォッチドッグタイマのリフレッシュ動作を行います。
引数	なし
リターン値	なし
補足	なし

6.9 フローチャート

6.9.1 メイン処理

図 6.5 にメイン処理のフローチャートを示します。

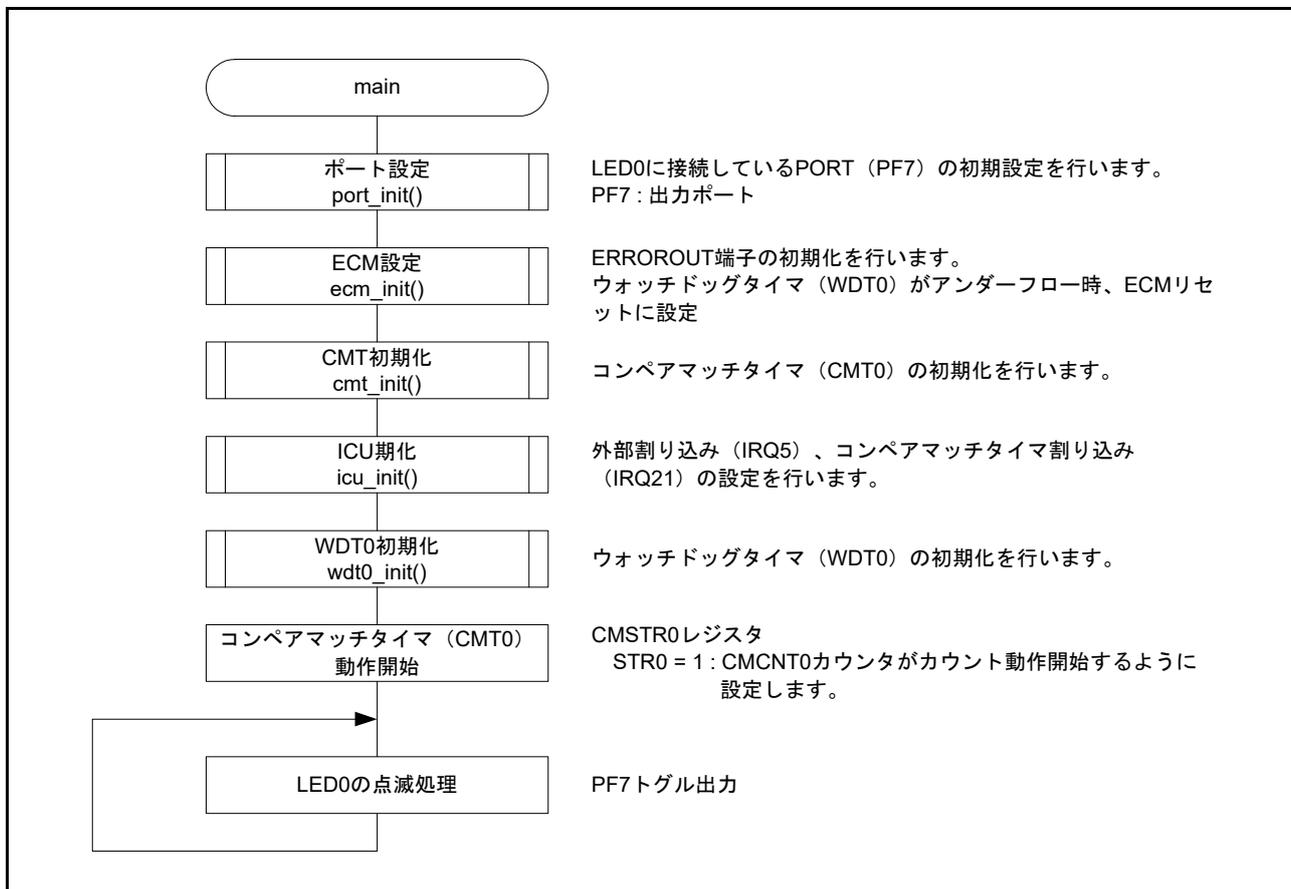


図 6.5 メイン処理

6.9.2 WDT0 初期化処理

図 6.6 に WDT0 初期化処理のフローチャートを示します。

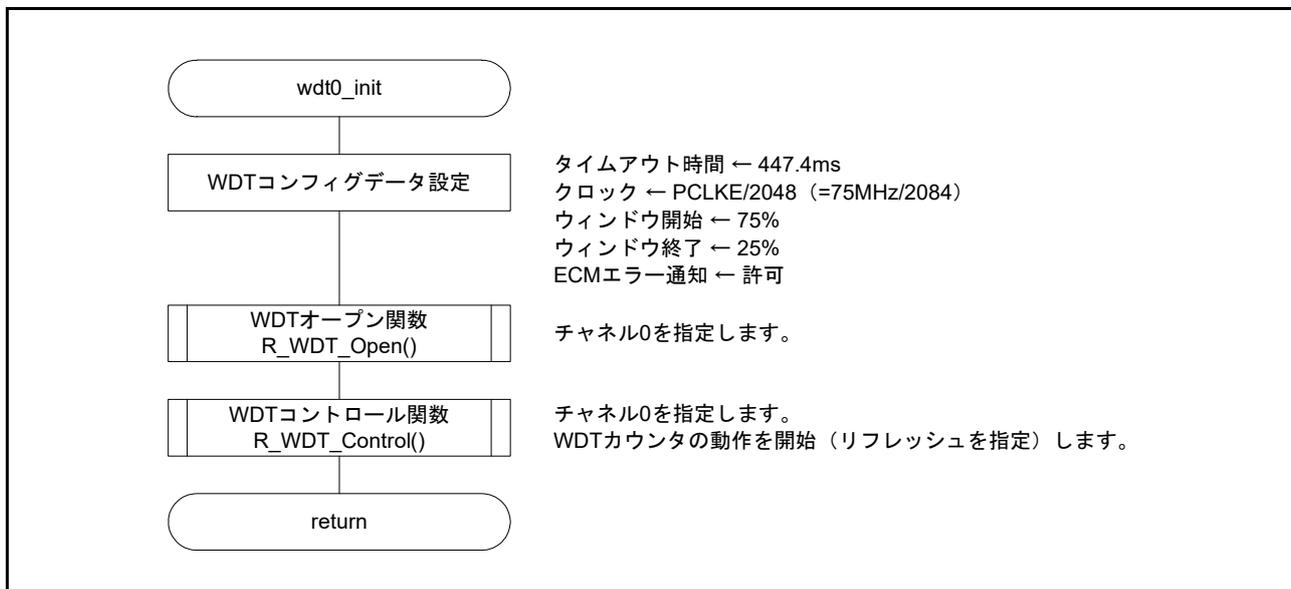


図 6.6 WDT0 初期化処理

6.9.3 WDT オープン関数

図 6.7 に WDT オープン関数のフローチャートを示します。

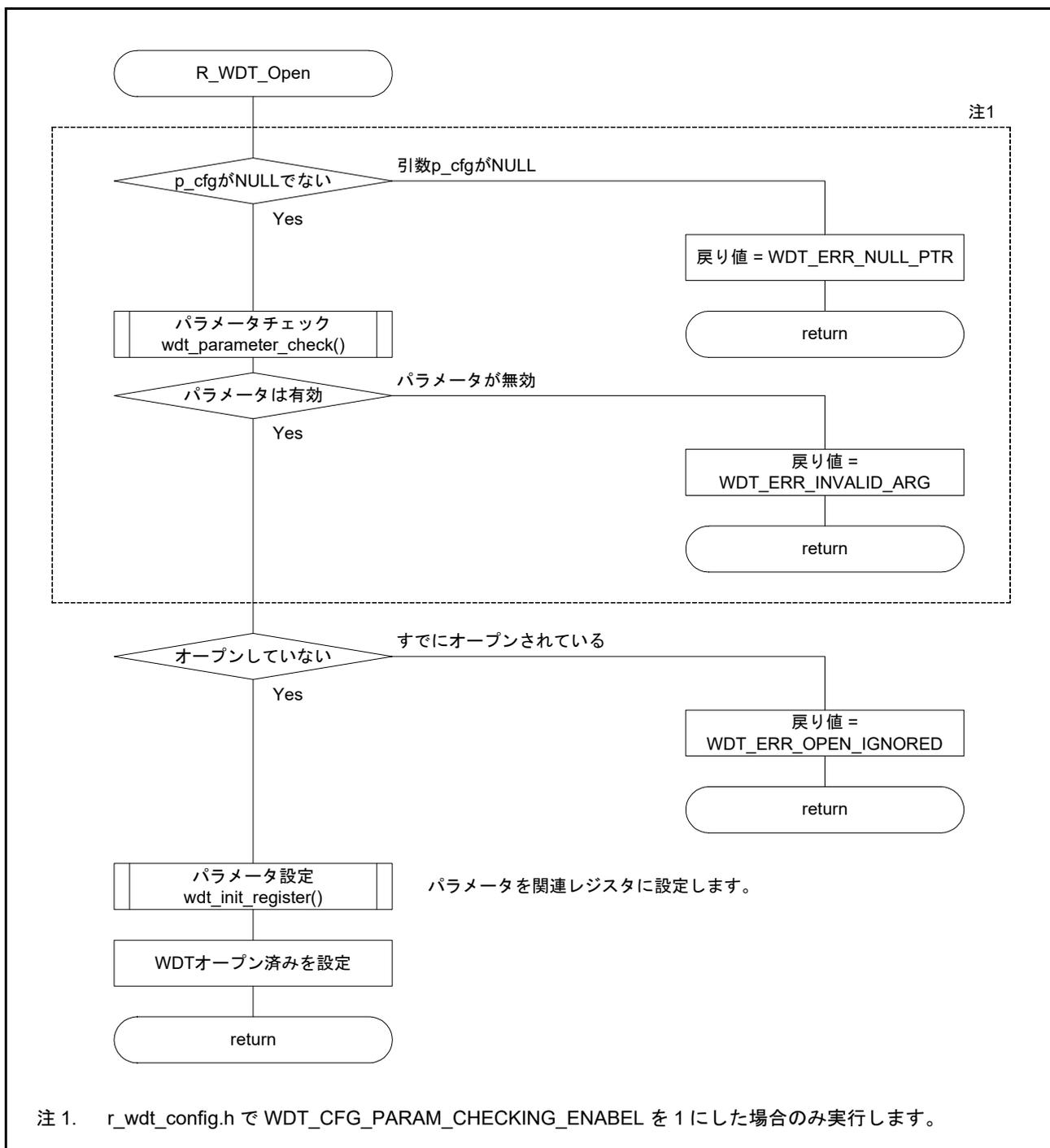


図 6.7 WDT オープン関数

6.9.4 WDT コントロール関数

図 6.8、図 6.9 に WDT コントロール関数のフローチャートを示します。

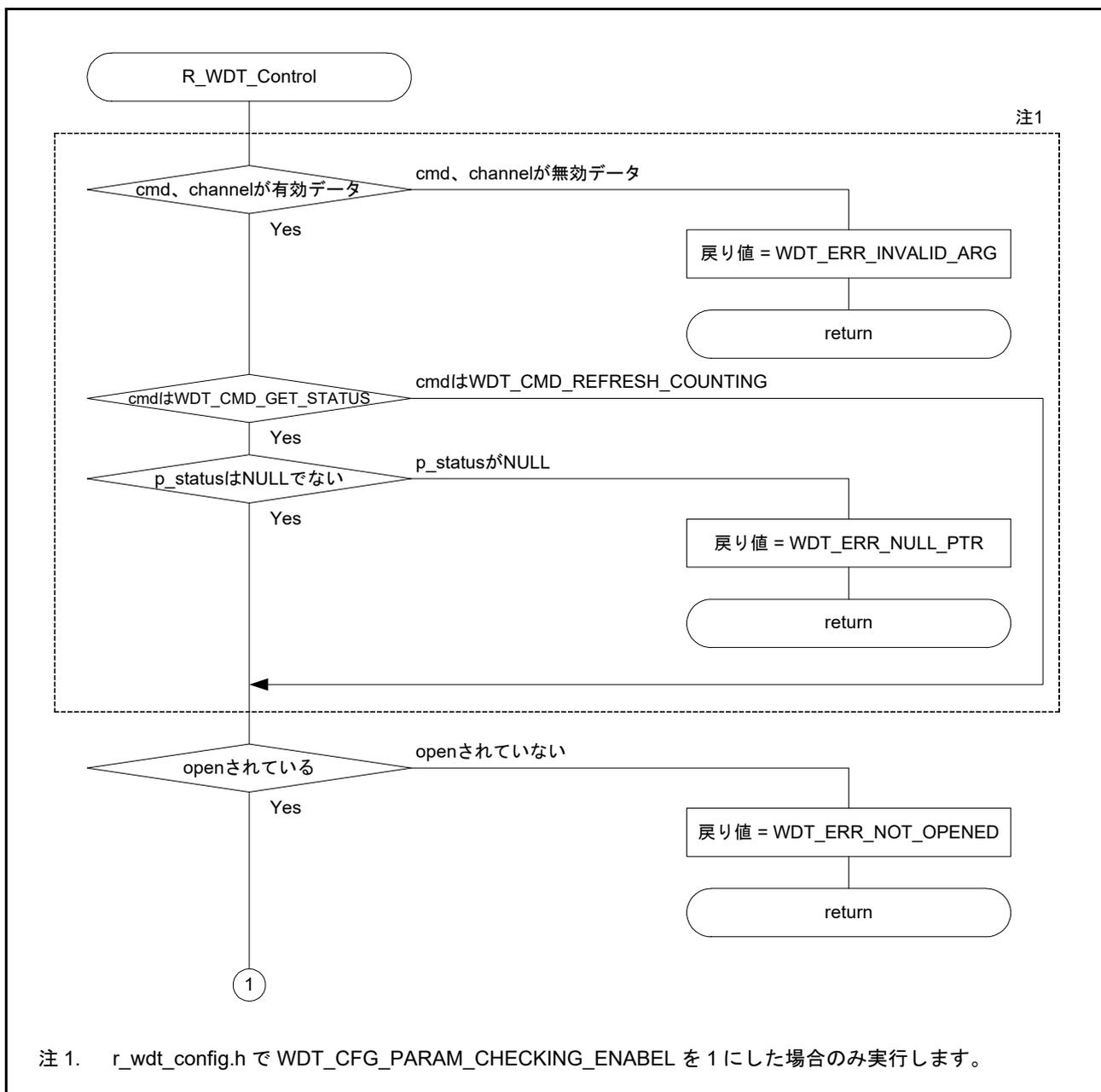


図 6.8 WDT コントロール関数

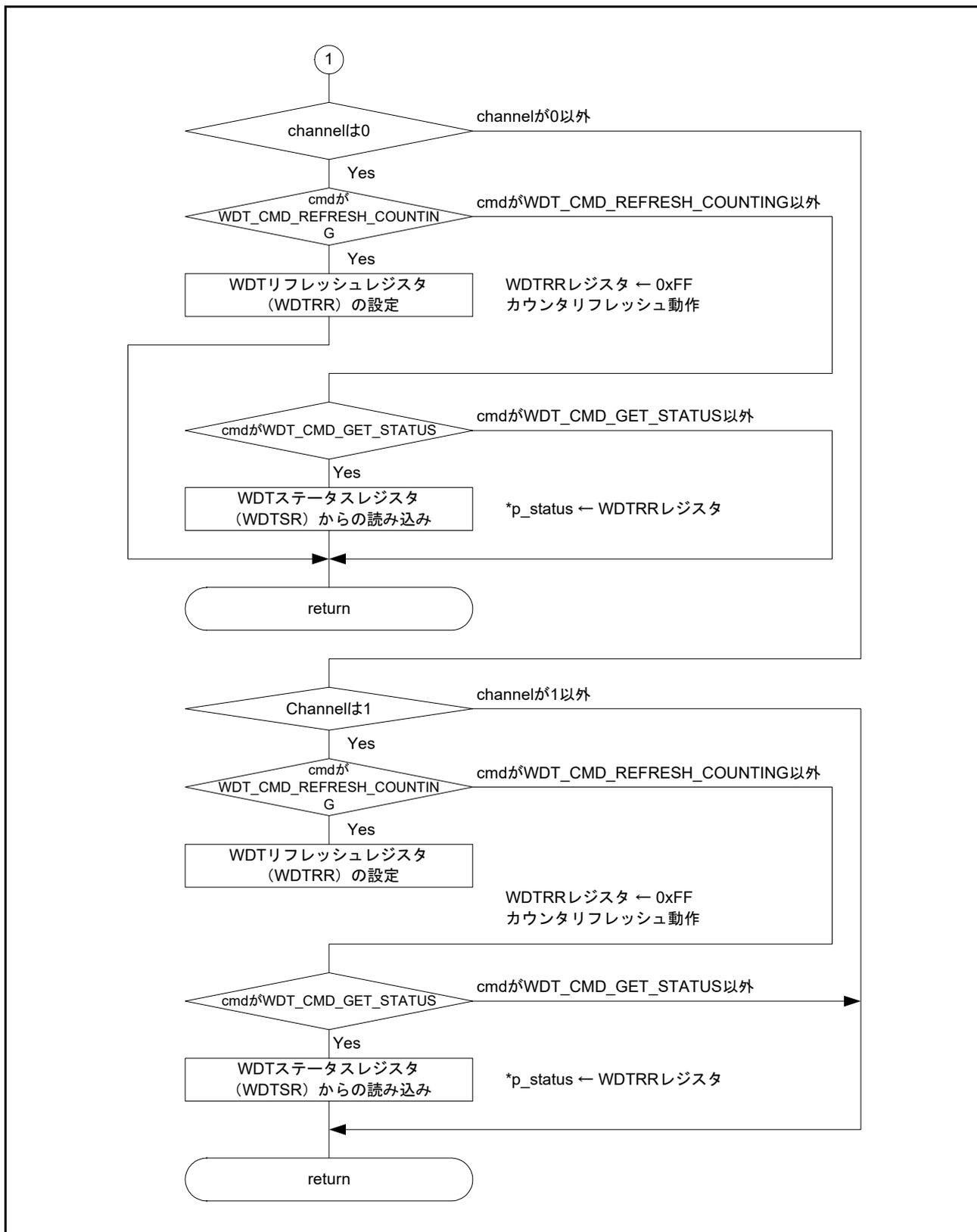


図 6.9 WDT コントロール関数

6.9.5 IRQ9 割り込み (IRQ 端子割り込み 5) 処理

図 6.10 に IRQ9 割り込み (IRQ 端子割り込み 5) 処理のフローチャートを示します。

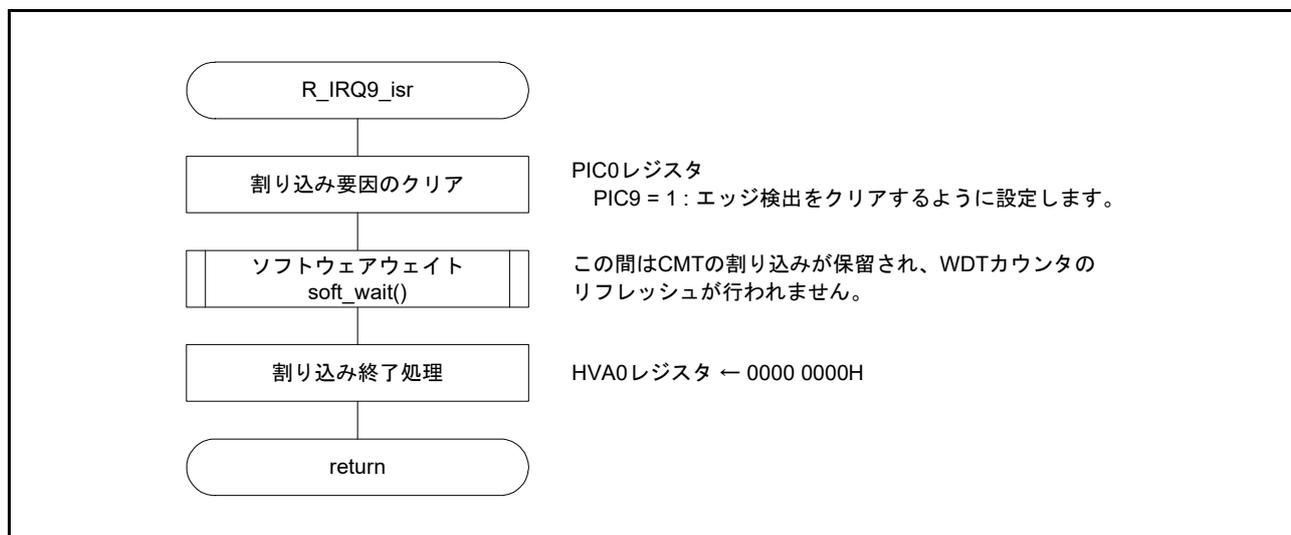


図 6.10 IRQ9 割り込み (IRQ 端子割り込み 5) 処理

6.9.6 IRQ21 割り込み (コンペアマッチタイマ ch0 割り込み) 処理

図 6.11 に IRQ21 割り込み (コンペアマッチタイマ ch0 割り込み) 処理のフローチャートを示します。

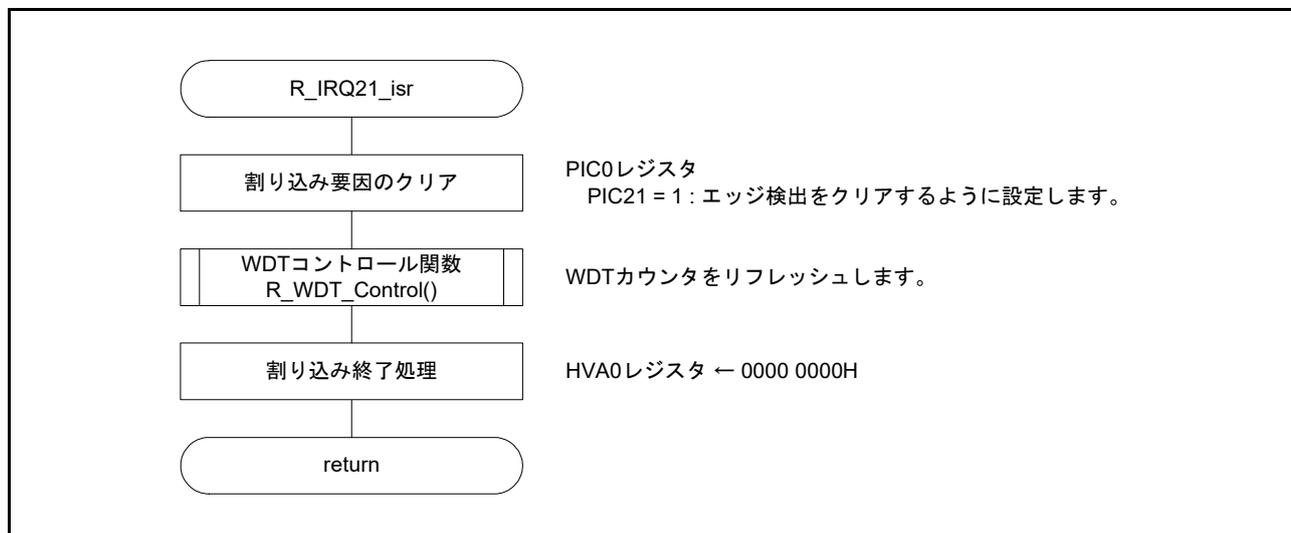


図 6.11 IRQ21 割り込み (コンペアマッチタイマ ch0 割り込み) 処理

7. サンプルプログラム

サンプルプログラムは、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	ウォッチドッグタイマ (WDTA) 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	2015.02.03	—	初版発行
0.20	2015.03.10	9	6.2.4 必要メモリサイズ 表6.2 必要メモリサイズ 修正
		10	6.4 固定幅整数一覧 表6.4 サンプルプログラムで使用する固定幅整数 修正
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	
		4	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	
		9	6.2.4 説明文 参照を追加
		9	表6.2 タイトルを一部追加
1.20	2015.12.03	2. 動作環境	
		4	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	
1.30	2017.04.05	2. 動作環境	
		4	表2.1 動作環境 統合開発環境の内容変更
		—	6.2.4 必要メモリサイズ 削除
1.40	2018.06.07	2. 動作環境	
		4	表2.1 動作環境 統合開発環境の内容変更
		5. ハードウェア説明	
		7	図5.1 ハードウェア構成例 モジュール名変更
		8. 参考ドキュメント	
		24	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/>