

RL78/G24

タイマ RD2 タイマモード (インプットキャプチャ機能とアウトプットコンペア機能の併用)

要旨

本アプリケーションノートでは、RL78/G24 のタイマ RD2 タイマモードのインプットキャプチャ機能とアウトプットコンペア機能を併用する方法を説明します。

動作確認デバイス

RL78/G24

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
1.1 仕様概要	3
1.2 動作概要	4
2. 動作確認条件	9
3. ハードウェア説明	10
3.1 ハードウェア構成例	10
3.2 使用端子一覧	10
4. ソフトウェア説明	11
4.1 スマート・コンフィグレータの設定	11
4.1.1 システム設定	11
4.1.2 コンポーネントの設定	13
4.2 フォルダ構成	15
4.2.1 Config_TRD0_user.c の変更点	16
4.3 オプション・バイトの設定一覧	18
4.4 定数一覧	18
4.5 変数一覧	18
4.6 関数一覧	19
4.7 関数仕様	19
4.8 フローチャート	20
4.8.1 メイン処理	20
4.8.2 R_Config_TRD0_Create_UserInit 関数	21
4.8.3 r_Config_TRD0_trd0_interrupt 関数	22
5. サンプルコード	23
6. 参考ドキュメント	23
改訂記録	24

1. 仕様

1.1 仕様概要

タイマ RD2 カウンタ 0（以下、タイマ RD20）のインプットキャプチャ機能とアウトプットコンペア機能を併用します。

インプットキャプチャ機能では、TRDIOA0 端子に入力した波形のパルス幅を計算します。

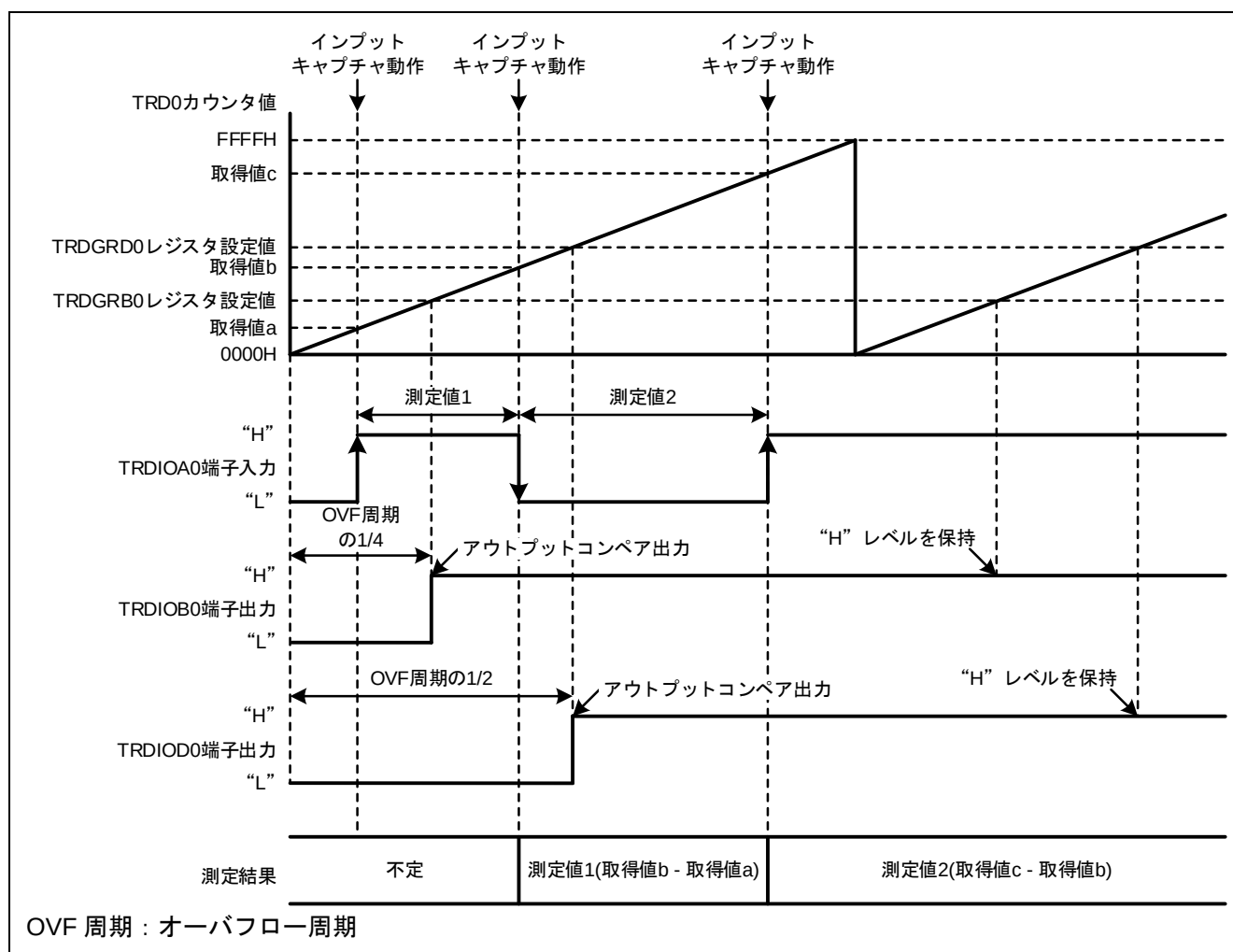
アウトプットコンペア機能では、TRDIOB0 端子／TRDIOD0 端子の出力レベルを“L”から“H”に変更します。

表 1-1 に使用する周辺機能と用途を、図 1-1 に出力波形を示します。

表 1-1 使用する周辺機能と用途

周辺機能	用途
タイマ RD2（タイマ RD20）	パルス幅測定および、波形出力

図 1-1 インプットキャプチャ機能とアウトプットコンペア機能の併用出力波形



1.2 動作概要

タイマ RD20 のインプットキャプチャ機能とアウトプットコンペア機能を併用します。TRDIOA0 端子にインプットキャプチャ機能、TRDIOB0 端子と TRDIOD0 端子にアウトプットコンペア機能を割り当てます。本アプリケーションノートでは、スマート・コンフィグレータのコード生成を使用しています。しかし、2つの機能を併用する設定に対応していないため、インプットキャプチャ機能はスマート・コンフィグレータで、アウトプットコンペア機能は UserInit 関数を使用し、必要な処理を追加することで対応します。

インプットキャプチャ機能は両エッジを指定することで TRDIOA0 端子に入力する立ち上がり/立ち下がりエッジを検出し、検出結果よりパルス幅（ハイ・レベル幅、ロウ・レベル幅）を算出します。

インプットキャプチャ機能の設定を以下に示します。

<設定>

- インプットキャプチャ機能で TRD0 を設定します。
- カウント・ソースに fTRD（96MHz）を設定します。
- カウンタ設定をクリア禁止に設定します。
- レジスタ機能設定は TRDGRC0、TRDGRD0 とともにジェネラル・レジスタに設定します。
- TRDIOA0 端子の入力設定は両エッジに設定します。
- TRDIOA0 端子に、デジタル・フィルタ機能を使用し、サンプリングクロックにカウント・ソース（96MHz）を選択します。
- TRDGRA0 入力キャプチャ割り込みと TRD0 オーバフロー割り込みを有効にします。

図 1-2 にインプットキャプチャ機能の動作を示します。

<TRD0 レジスタオーバフローなしの場合>

- (1) TRDIOA0 端子への立ち上がりエッジ入力を検出し、TRDGRA0 入力キャプチャ割り込みが発生します。INTTRD0 割り込み処理内で、TRD0 カウンタオーバフローなしの場合の計算式でパルス幅を算出し、IMFA フラグをクリアします。
- (2) TRDIOA0 端子への立ち下がりエッジ入力を検出し、TRDGRA0 入力キャプチャ割り込みが発生します。INTTRD0 割り込み処理内で、TRD0 カウンタオーバフロー無しの場合の計算式でパルス幅を算出し、IMFA フラグをクリアします。

オーバフローなしの場合（図 1-2①の区間）のパルス幅の計算式は以下の通りです。

$$\begin{aligned}\text{パルス幅} &= 1 \text{ カウント当たりの時間 } (1 / 96 [\text{MHz}] = 10.42 [\text{ns}]) \times (\text{今回の測定値} - \text{前回の測定値}) \\ &= 10.42 [\text{ns}] \times (44200 - 25000) \\ &= 200 [\mu\text{s}]\end{aligned}$$

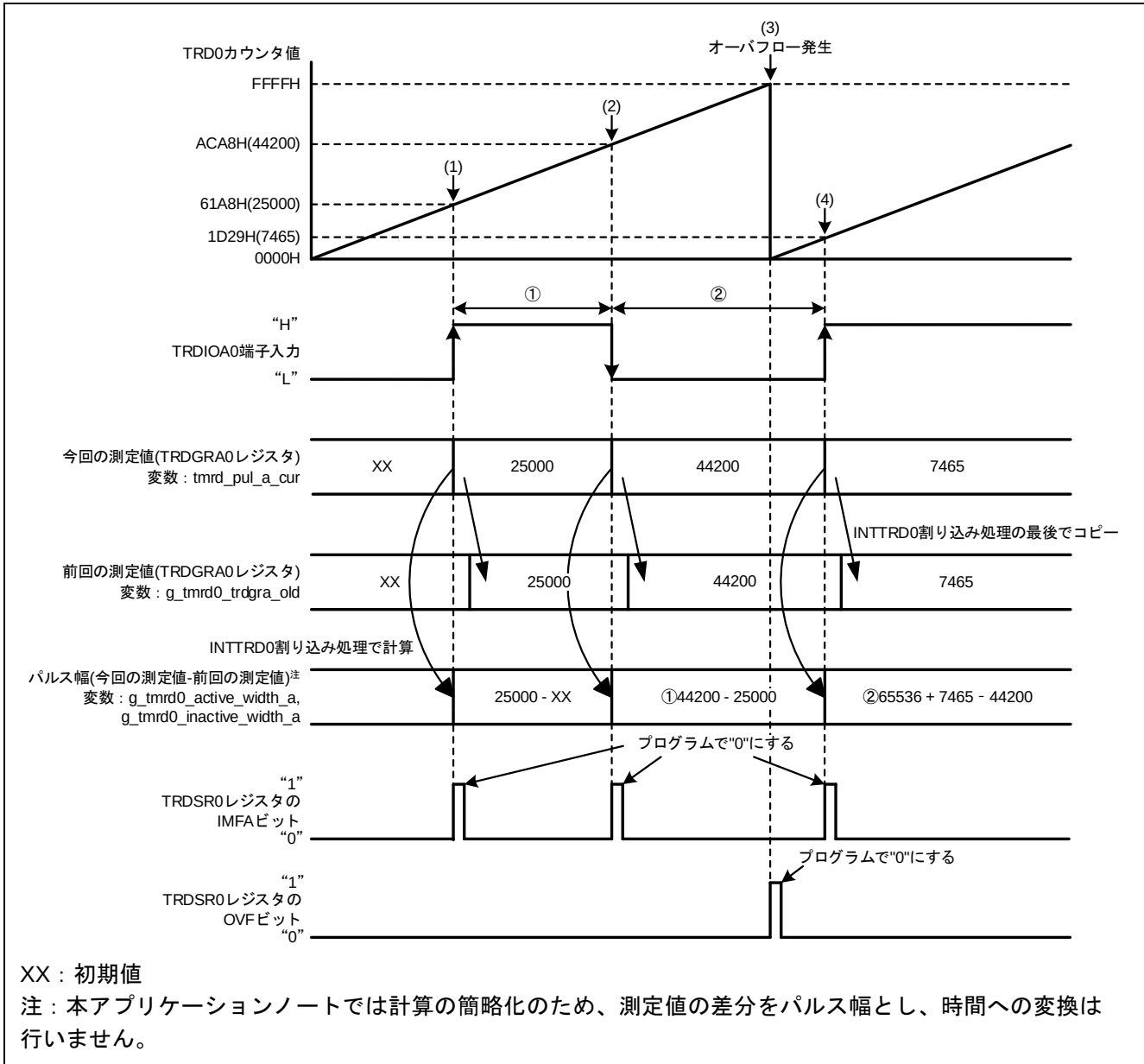
<TRD0 レジスタオーバフローありの場合>

- (3) TRD0 レジスタのオーバフローで、TRD0 オーバフロー割り込みが発生します。INTTRD0 割り込み処理内で、オーバフローカウンタの値をインクリメントし、OVF フラグをクリアします。
- (4) TRDIOA0 端子への立ち上がりエッジ入力を検出し、TRDGRA0 入力キャプチャ割り込みが発生します。INTTRD0 割り込み処理内で、TRD0 カウンタオーバフローありの場合の計算式でパルス幅を算出し、IMFA フラグをクリアします。

オーバフローが 1 回発生した場合（図 1-2②の区間）のパルス幅の計算式は以下の通りです。

$$\begin{aligned}\text{パルス幅} &= 1 \text{ カウント当たりの時間 } (1 / 96 [\text{MHz}] = 10.42 [\text{ns}]) \times \\ &\quad (10000\text{H}(65536) \times \text{オーバフローカウンタ}(1 \text{ 回}) + \text{今回の測定値} - \text{前回の測定値}) \\ &= 10.42 [\text{ns}] \times (65536 + 7465 - 44200) \\ &= 300 [\mu\text{s}]\end{aligned}$$

図 1-2 インプットキャプチャ機能の動作



アウトプットコンペア機能は、タイマ RD0 カウント開始からオーバフロー周期(683μs)の 1/4 経過で TRDIOB0 端子の出力レベルを、オーバフロー周期(683μs)の 1/2 経過で TRDIOD0 端子の出力レベルを"L"から"H"に変更します。以降、TRDIOB0 端子と TRDIOD0 端子は“H”を保持します。

アウトプットコンペア機能の設定を以下に示します。

<設定>

- カウント・ソースに fTRD（96MHz）を設定します。
- カウンタ設定を TRDGRA0 コンペアー一致後もカウント継続に設定し、クリア禁止とします。
- レジスタ機能設定は TRDGRC0、TRDGRD0 とともにジェネラル・レジスタに設定します。
- コンペア値設定は手動で設定するためデフォルトのままコード生成します。
- 出力設定は TRDIOB0 端子、TRDIOD0 端子ともに初期出力 Low、コンペアー一致で High レベル出力に設定します。
- 割り込みは使用しません。

図 1-3 にアウトプットコンペア機能の動作を示します。

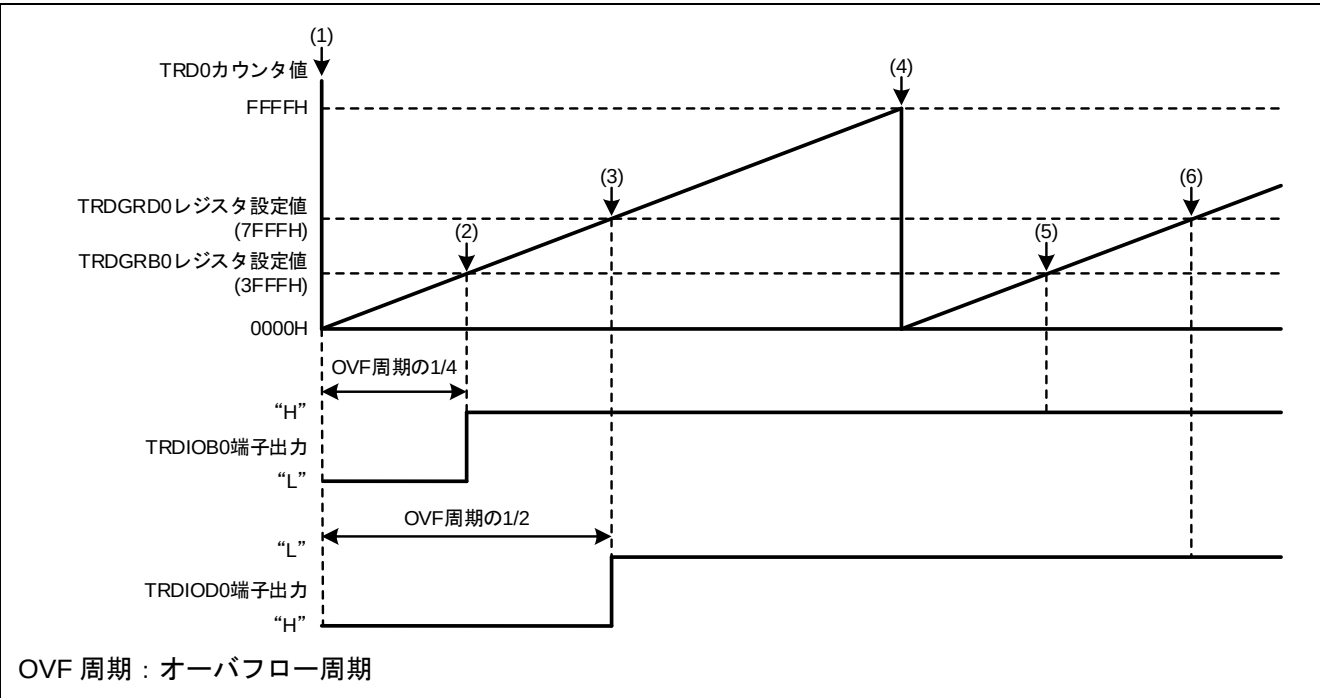
- (1) タイマ RD0 カウント開始します。
- (2) TRD0 レジスタと TRDGRB0 レジスタのコンペアー一致で、TRDIOB0 端子の出力レベルが"L"から"H"に変化します。
- (3) TRD0 レジスタと TRDGRD0 レジスタのコンペアー一致で、TRDIOD0 端子の出力レベルが"L"から"H"に変化します。
- (4) TRD0 レジスタのオーバフローが発生し、TRD0 レジスタが“0000H”にクリアされます。
- (5) TRD0 レジスタと TRDGRB0 レジスタのコンペアー一致が発生しますが、TRDGRB0 端子の出力レベルが“H”のため、出力レベルを保持します。
- (6) TRD0 レジスタと TRDGRD0 レジスタのコンペアー一致が発生しますが、TRDGRD0 端子の出力レベルが“H”のため、出力レベルを保持します。

TRDIOB0 端子と TRDIOD0 端子のロウ・レベル出力幅の計算式は以下の通りです。

$$\begin{aligned} \text{TRDIOB0 端子 : オーバフロー周期(683}\mu\text{s)の } 1/4 &= 1 / 96 [\text{MHz}] \times (\text{TRDGRB0 レジスタ設定値} + 1) \\ &= 10.42 [\text{ns}] \times (16383 + 1) \end{aligned}$$

$$\begin{aligned} \text{TRDIOD0 端子 : オーバフロー周期(683}\mu\text{s)の } 1/2 &= 1 / 96 [\text{MHz}] \times (\text{TRDGRB0 レジスタ設定値} + 1) \\ &= 10.42 [\text{ns}] \times (32767 + 1) \end{aligned}$$

図 1-3 アウトプットコンペア機能の動作



以下の設定を R_Config_TRD0_Create_UserInit 関数に追加することでインプットキャプチャ機能とアウトプットコンペア機能の併用を実現できます。

```
void R_Config_TRD0_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */

    TRDOER1 = 0xF5U;          /* TRDIOD0 output disable : Output enabled
                               TRDIOC0 output disable : Output disabled
                               (TRDIOC0 pin functions as an I/O port.)
                               TRDIOB0 output disable : Output enable
                               TRDIOA0 output disable : Output disabled
                               (TRDIOA0 pin functions as an I/O port.) */

    TRDOCR = 0x00U;          /* TRDIOD0 initial output level select :
                               Low initial output
                               TRDIOB0 initial output level select :
                               Low initial output */

    TRDIOA0 = 0x26U;          /* High output by compare match with TRDGRB0
                               TRDGRB mode select : Output compare function */

    TRDIORC0 = 0xACU;          /* High output by compare match with TRDGRD0
                               TRDGRD mode select : Output compare function */

    TRDGRB0 = 0x3FFFU;          /* Compare value : Set 1/4 of
                               the overflow period(683us) */

    TRDGRD0 = 0x7FFFU;          /* Compare value : Set 1/2 of
                               the overflow period(683us) */

    /* Set TRDIOB0 pin */
    POM1  &= 0xDFU;
    PMCA1 &= 0xDFU;
    P1    &= 0xDFU;
    PM1   &= 0xDFU;

    /* Set TRDIOD0 pin */
    POM1  &= 0xEFU;
    PMCA1 &= 0xEFU;
    P1    &= 0xEFU;
    PM1   &= 0xEFU;

    /* End user code. Do not edit comment generated here */
}
```


2. 動作確認条件

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

表 2-1 動作確認条件

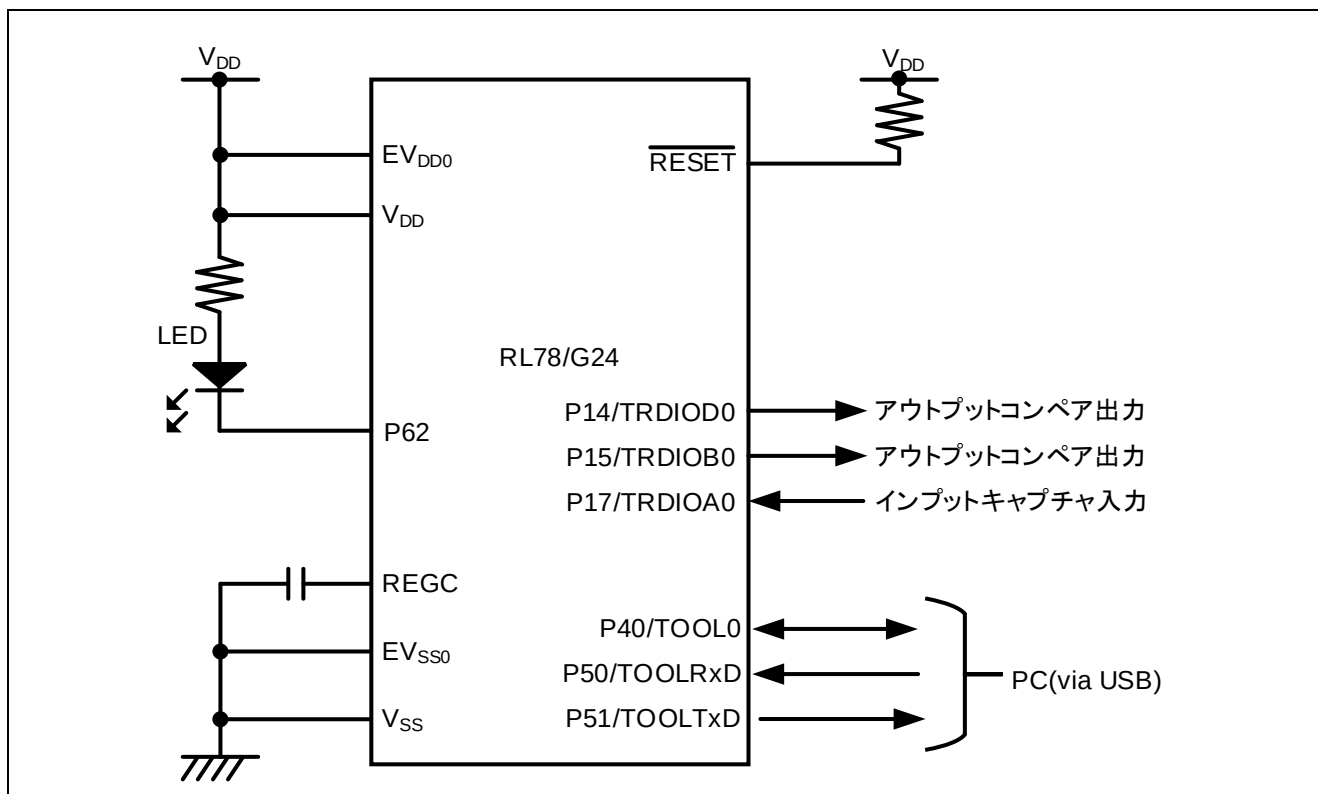
項目	内容
使用マイコン	RL78/G24 (R7F101GLG)
動作周波数	- 高速オンチップ・オシレータ・クロック (f_{HOCO}) : 8MHz - PLL 発振回路出力 (f_{PLL}) : 96MHz - CPU/周辺ハードウェア・クロック (f_{CLK}) : 48MHz
動作電圧	3.3V (2.7V~5.5V で動作可能) - LVD0 動作 (V_{LVD0}) : リセット・モード 立ち上がり時 TYP. 2.97V 立ち下がり時 TYP. 2.91V
統合開発環境 (CS+)	ルネサスエレクトロニクス製 CS+ for CC V8.10.00
C コンパイラ (CS+)	ルネサスエレクトロニクス製 CC-RL V1.12.01
統合開発環境 (e ² studio)	ルネサスエレクトロニクス製 e ² studio 2023-04 (23.4.0)
C コンパイラ (e ² studio)	ルネサスエレクトロニクス製 CC-RL V1.12.00
統合開発環境 (IAR)	IAR システム製
C コンパイラ (IAR)	IAR Embedded Workbench for Renesas RL78 V4.21.1
スマート・コンフィグレータ	V.1.7.0
ボードサポートパッケージ (r_bsp)	V.1.60
エミュレータ	CS+、e ² studio : COM ポート IAR : E2 エミュレータ Lite
使用ボード	RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ)

3. ハードウェア説明

3.1 ハードウェア構成例

図 3-1 に本アプリケーションのサンプルコードで使用するハードウェア構成例を示します。

図 3-1 ハードウェア構成例



- 注意 1. この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい）。
- 注意 2. EV_{SS} で始まる名前の端子がある場合には V_{SS} に、 EV_{DD} で始まる名前の端子がある場合には V_{DD} にそれぞれ接続してください。
- 注意 3. V_{DD} は LVD0 にて設定したリセット解除電圧（ V_{LVD0} ）以上にしてください。

3.2 使用端子一覧

表 3-1 に使用端子と機能を示します。

表 3-1 使用端子と機能

端子名	入出力	内容
P14/TRDIOD0	出力	アウトプットコンペア出力
P15/TRDIOB0	出力	アウトプットコンペア出力
P17/TRDIOA0	入力	インプットキャプチャ入力
P62	出力	LED1 点灯（Low Active）

注意 本アプリケーションノートは、使用端子のみを端子処理しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

4. ソフトウェア説明

4.1 スマート・コンフィグレータの設定

本サンプルプログラムにおけるスマート・コンフィグレータの設定を示します。スマート・コンフィグレータの設定における各表の項目、設定内容は設定画面の表記で記載しています。

4.1.1 システム設定

本サンプルプログラムで使用しているシステム設定を以下に示します。

なお、本サンプルプログラムで使用しているシステム設定は、統合開発環境 e² studio と CS+は同じですが、IAR は異なります。ご使用の環境に合わせて適切な設定を行ってください。

まず、図 4-1 に本サンプルプログラム（e² studio、CS+）で使用しているシステム設定を示します。

RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ) で COM port デバッグを行う場合、統合開発環境（e² studio、CS+）内の設定を適切に行う必要があります。詳細は、「RL78/G24 Fast Prototyping Board ユーザーズマニュアル（R20UT5091J）」の「7.1 e² studio で COM port デバッグを使用する場合」と「7.2 CS+を COM port デバッグを使用する場合」を参照してください。

図 4-1 システム設定（e² studio、CS+）



次に、図 4-2 に本サンプルプログラム（IAR）で使用しているシステム設定を示します。

図 4-2 システム設定（IAR）

The screenshot shows the 'システム設定' (System Settings) dialog box in IAR. The 'オンチップ・デバッグ設定' (On-chip Debug Settings) section is expanded. The following settings are highlighted with blue boxes and arrows pointing to a 'チェックをつける' (Check) label:

- オンチップ・デバッグ動作設定:** ☒ エミュレータを使う (Use emulator) and ☐ COMポート (COM port).
- エミュレータ設定:** ☒ E2 Lite.
- 疑似RRM/DMM機能設定:** ☒ 使用する (Use).
- Start/Stop関数機能設定:** ☒ 使用しない (Do not use).
- 通過ポイント機能設定:** ☒ 使用しない (Do not use).
- トレース機能設定:** ☒ 使用しない (Do not use).
- セキュリティID設定:** ☒ セキュリティIDを設定する (Set security ID). The security ID field below contains '0x00000000000000000000000000000000'.
- セキュリティID認証失敗時の設定:** ☒ フラッシュ・メモリのデータを消去しない (Do not erase flash memory data) and ☐ フラッシュ・メモリのデータを消去する (Erase flash memory data).

4.1.2 コンポーネントの設定

本サンプルプログラムで使用しているコンポーネントの設定を以下に示します。

表 4-1 コンポーネントの設定（タイマ RD2）

項目	内容
コンポーネント	インプットキャプチャ機能
コンフィグレーション名	Config_TRD0
リソース	TRD0

図 4-3 タイマ RD20 の設定

設定

クロック・ソース設定

クロック・ソース

fTRD

(クロック周波数: 96000 kHz, fPLLをfTRDとして選択します)

外部クロックエッジ選択

立ち上がりエッジ

カウンタ設定

カウンタクリア

クリア無効

レジスタ機能設定

TRDGRCO

一般レジスタ

TRDGRD0

一般レジスタ

入力設定

☒ TRDIOA0端子

両エッジ

☐ TRDIOB0端子

立ち上がりエッジ

☐ TRDIOC0端子

立ち上がりエッジ

☐ TRDIOD0端子

立ち上がりエッジ

☐ ELC 入力

チェックをはずす

デジタル・フィルタ機能設定

☒ TRDIOA0デジタル・フィルタが有効

☐ TRDIOB0デジタル・フィルタが有効

☐ TRDIOC0デジタル・フィルタが有効

☐ TRDIOD0デジタル・フィルタが有効

チェックをつける

サンプリング時計

クロック・ソース

(サンプリング周波数: 96000 kHz)

割り込み設定

☒ TRDGRA0入力キャプチャ割り込みを有効にする

☒ TRDGRB0入力キャプチャ割り込みを有効にする

☒ TRDGRCO入力キャプチャ割り込みを有効にする

☒ TRDGRD0入力キャプチャ割り込みを有効にする

☒ TRD0オーバーフロー割り込みを有効にする

優先順位

レベル3(低優先順位)

表 4-2 コンポーネントの設定（ポート）

項目	内容
コンポーネント	ポート
コンフィグレーション名	Config_PORT
リソース	PORT

図 4-4 ポートの設定

設定

ポート選択 PORT6

☐ PORT0

☐ PORT1

☐ PORT2

☐ PORT3

☐ PORT4

☐ PORT5

☒ PORT6

☐ PORT7

☐ PORT12

☐ PORT13

☐ PORT14

チェックをつける

ポート・モード設定

☒ Pmnレジスタ値を読み出す

☐ デジタル出力レベルを読み出す

設定

ポート選択 PORT6

☐ すべてに適用

☒ 使用しない

☐ 入力

☐ 出力

☐ 出力電流

☐ 内蔵プルアップ

P60

☒ 使用しない

☐ 入力

☐ 出力

☐ 出力電流

P61

☒ 使用しない

☐ 入力

☐ 出力

☐ 出力電流

P62

☐ 使用しない

☐ 入力

☒ 出力

☐ 出力電流

☐ 内蔵プルアップ

チェックをつける

4.2 フォルダ構成

表 4-3 にサンプルコードの使用するソースファイル／ヘッダファイルの構成を示します。なお、統合開発環境で自動生成されるファイル、bsp 環境のファイルは除きます。

表 4-3 フォルダ構成

フォルダ、ファイル名	説明	スマート・コンフィグレータを使用
¥r01an6892_trd2_timer<DIR> ^{注3}	サンプルコードのフォルダ	
¥src<DIR>	プログラム格納用フォルダ	
main.c	サンプルコードソースファイル	
¥smc_gen<DIR>	スマート・コンフィグレータ生成フォルダ	√
¥Config_PORT<DIR>	PORT 用プログラム格納フォルダ	√
Config_PORT.c	PORT 用ソースファイル	√
Config_PORT.h	PORT 用ヘッダファイル	√
Config_PORT_user.c	PORT 用割り込みソースファイル	√ ^{注1}
¥Config_TRD0<DIR>	TRD0 用プログラム格納フォルダ	√
Config_TRD0.c	TRD0 用ソースファイル	√
Config_TRD0.h	TRD0 用ヘッダファイル	√
Config_TRD0_user.c	TRD0 用割り込みソースファイル	√ ^{注2}
¥general<DIR>	初期化、共通プログラム格納フォルダ	√
¥r_bsp<DIR>	BSP 用プログラム格納フォルダ	√
¥r_config<DIR>	プログラム格納フォルダ	√

補足 ” <DIR> ” は、ディレクトリを意味します。

注1. 本サンプルコードでは使用しません。

注2. スマート・コンフィグレータの出力に対して変更を行っています。
変更内容の詳細は「4.2.1 Config_TRD0_user.c の変更点」を参照してください。

注3. IAR 版のサンプルコードは r01an6892_trd2_timer.ipcf を格納しています。ipcf ファイルについては、「RL78 スマート・コンフィグレータ ユーザーガイド：IAR 編（R20AN0581）」を確認してください。

4.2.1 Config_TRD0_user.c の変更点

本アプリケーションノートでは、スマート・コンフィグレータで生成された Config_TRD0_user.c を変更しています。R_Config_TRD0_Create_UserInit 関数では「1.2 動作概要」に記載の設定を追記し、r_Config_TRD0_trd0_interrupt 関数では、下記の黄色で色付けした箇所を変更しています。

関数名 : r_Config_TRD0_trd0_interrupt	
変更前	変更後
<pre>static void __near r_Config_TRD0_trd0_interrupt(void) { uint16_t tmrdr_pul_a_cur = TRDGRA0; uint8_t trdier0_temp = TRDIER0; TRDIER0 = 0x00U; /* overflow process */ if ((TRDSR0 & _10_TRD_INTTOV_GENERATE_FLAG) == _10_TRD_INTTOV_GENERATE_FLAG) { TRDSR0 &= (uint8_t)~_10_TRD_INTTOV_GENERATE_FLAG; g_tmrdr0_ovf_a += 1U; } /* TRDGRA0 input capture interrupt */ if ((TRDSR0 & _01_TRD_INTA_GENERATE_FLAG) == _01_TRD_INTA_GENERATE_FLAG) { TRDSR0 &= (uint8_t)~_01_TRD_INTA_GENERATE_FLAG; if (0U == g_tmrdr0_ovf_a) { if ((P1 & 0x80U) == 0x80U) { g_tmrdr0_inactive_width_a = (uint32_t)((uint32_t)tmrdr_pul_a_cur - (uint32_t)g_tmrdr0_trdgra_old); } else { </pre>	<pre>static void __near r_Config_TRD0_trd0_interrupt(void) { uint16_t tmrdr_pul_a_cur = TRDGRA0; uint8_t trdier0_temp = TRDIER0; TRDIER0 = 0x00U; /* overflow process */ if ((TRDSR0 & _10_TRD_INTTOV_GENERATE_FLAG) == _10_TRD_INTTOV_GENERATE_FLAG) { TRDSR0 &= (uint8_t)~_10_TRD_INTTOV_GENERATE_FLAG; g_tmrdr0_ovf_a += 1U; /* Start user code */ g_capture_flg = 0x01U; /* End user code */ } /* TRDGRA0 input capture interrupt */ if ((TRDSR0 & _01_TRD_INTA_GENERATE_FLAG) == _01_TRD_INTA_GENERATE_FLAG) { TRDSR0 &= (uint8_t)~_01_TRD_INTA_GENERATE_FLAG; if (0U == g_tmrdr0_ovf_a) { if ((P1 & 0x80U) == 0x80U) { g_tmrdr0_inactive_width_a = (uint32_t)((uint32_t)tmrdr_pul_a_cur - (uint32_t)g_tmrdr0_trdgra_old); /* Start user code */ g_capture_flg = 0x02U; /* End user code */ } else { </pre>

<pre> g_tmr0_active_width_a = (uint32_t)((uint32_t)tmrd_pul_a_cur - (uint32_t)g_tmr0_trdgra_old); } } else { if ((P1 & 0x80U) == 0x80U) { g_tmr0_inactive_width_a = (uint32_t)(((0x10000UL * (uint32_t)g_tmr0_ovf_a) + (uint32_t)tmrd_pul_a_cur) - (uint32_t)g_tmr0_trdgra_old); } else { g_tmr0_active_width_a = (uint32_t)(((0x10000UL * (uint32_t)g_tmr0_ovf_a) + (uint32_t)tmrd_pul_a_cur) - (uint32_t)g_tmr0_trdgra_old); } g_tmr0_ovf_a = 0U; } g_tmr0_trdgra_old = tmrd_pul_a_cur; } TRDIER0 = trdier0_temp; } </pre>	<pre> g_tmr0_active_width_a = (uint32_t)((uint32_t)tmrd_pul_a_cur - (uint32_t)g_tmr0_trdgra_old); /* Start user code */ g_capture_flg = 0x04U; /* End user code */ } } else { if ((P1 & 0x80U) == 0x80U) { g_tmr0_inactive_width_a = (uint32_t)(((0x10000UL * (uint32_t)g_tmr0_ovf_a) + (uint32_t)tmrd_pul_a_cur) - (uint32_t)g_tmr0_trdgra_old); /* Start user code */ g_capture_flg = 0x02U; /* End user code */ } else { g_tmr0_active_width_a = (uint32_t)(((0x10000UL * (uint32_t)g_tmr0_ovf_a) + (uint32_t)tmrd_pul_a_cur) - (uint32_t)g_tmr0_trdgra_old); /* Start user code */ g_capture_flg = 0x04U; /* End user code */ } g_tmr0_ovf_a = 0U; } g_tmr0_trdgra_old = tmrd_pul_a_cur; } TRDIER0 = trdier0_temp; } </pre>
--	---

備考 r_Config_TRD0_trd0_interrupt 関数には任意の位置にユーザーコードを追加できる指定タグ（/* Start user code */ ... /* End user code */）を使用しています。そのため、再度スマート・コンフィグレータでプログラムを自動生成しても、出力値（変更前）に上書きされず、追加されたユーザーコードはマージ（保護）されます。指定タグの詳細は、「RL78 スマート・コンフィグレータ V1.5.0 リリースノート」の「2.3.3 ユーザーコード保護機能のサポート」を参照してください。

4.3 オプション・バイトの設定一覧

表 4-4 にオプション・バイト設定を示します。

表 4-4 オプション・バイト設定

アドレス	設定値	内容
000C0H/040C0H	1110 1111B (EFH)	ウォッチドッグ・タイマ動作停止 (リセット解除後、カウント停止)
000C1H/040C1H	1111 1011B (FBH)	LVD0 リセット・モード 検出電圧：立ち上がり 2.97V／立下り 2.91V
000C2H/040C2H	1110 1010B (EAH)	フラッシュ動作モード：高速メインモード 高速オンチップ・オシレータの周波数：8MHz
000C3H/040C3H	1000 0101B (85H)	オンチップ・デバッグ動作許可

4.4 定数一覧

本サンプルコードでは定数は使用しません。

4.5 変数一覧

表 4-5 に本サンプルコードで使用する変数一覧を以下に示します。

以下の変数はスマート・コンフィグレータで生成した変数です。

表 4-5 サンプルコードで使用する変数

変数名	型	内容	使用関数
g_tmrd0_active_width_a	uint32_t	ハイ・レベルのパルス幅算出値 格納	r_Config_TRD0_trd0_interrupt
g_tmrd0_inactive_width_a	uint32_t	ロウ・レベルのパルス幅算出値 格納	r_Config_TRD0_trd0_interrupt
g_tmrd0_trdgra_old	uint16_t	前回の測定値 (TRDGRA0 レジスタ値)	r_Config_TRD0_trd0_interrupt
g_tmrd0_ovf_a	uint8_t	オーバフローカウンタ	r_Config_TRD0_trd0_interrupt
tmrd_pul_a_cur	uint16_t	今回の測定値 (TRDGRA0 レジスタ値)	r_Config_TRD0_trd0_interrupt
g_capture_flg	uint8_t	パルス幅算出完了フラグ	r_Config_TRD0_trd0_interrupt

4.6 関数一覧

表 4-6 にサンプルコードで使用する関数を示します。ただし、スマート・コンフィグレータで生成された関数の内、変更を行っていないものは除きます。

表 4-6 関数一覧

関数名	概要	ソースファイル
main	メイン処理	main.c
R_Config_TRD0_Create_UserInit	アウトプットコンペア機能初期設定	Config_TRD0_user.c
r_Config_TRD0_trd0_interrupt	タイマ RD20 割り込み処理	Config_TRD0_user.c

4.7 関数仕様

サンプルコードの関数仕様を示します。

[関数名] main

概要	メイン処理
ヘッダ	r_smc_entry.h
宣言	void main (void);
説明	タイマ RD20 の動作を開始します
引数	なし
リターン値	なし
備考	なし

[関数名] R_Config_TRD0_Create_UserInit

概要	アウトプットコンペア機能の初期設定
ヘッダ	Config_TRD0.h
宣言	void R_Config_TRD0_Create_UserInit(void);
説明	アウトプットコンペア機能の初期設定を行います。
引数	なし
リターン値	なし
備考	なし

[関数名] r_Config_TRD0_trd0_interrupt

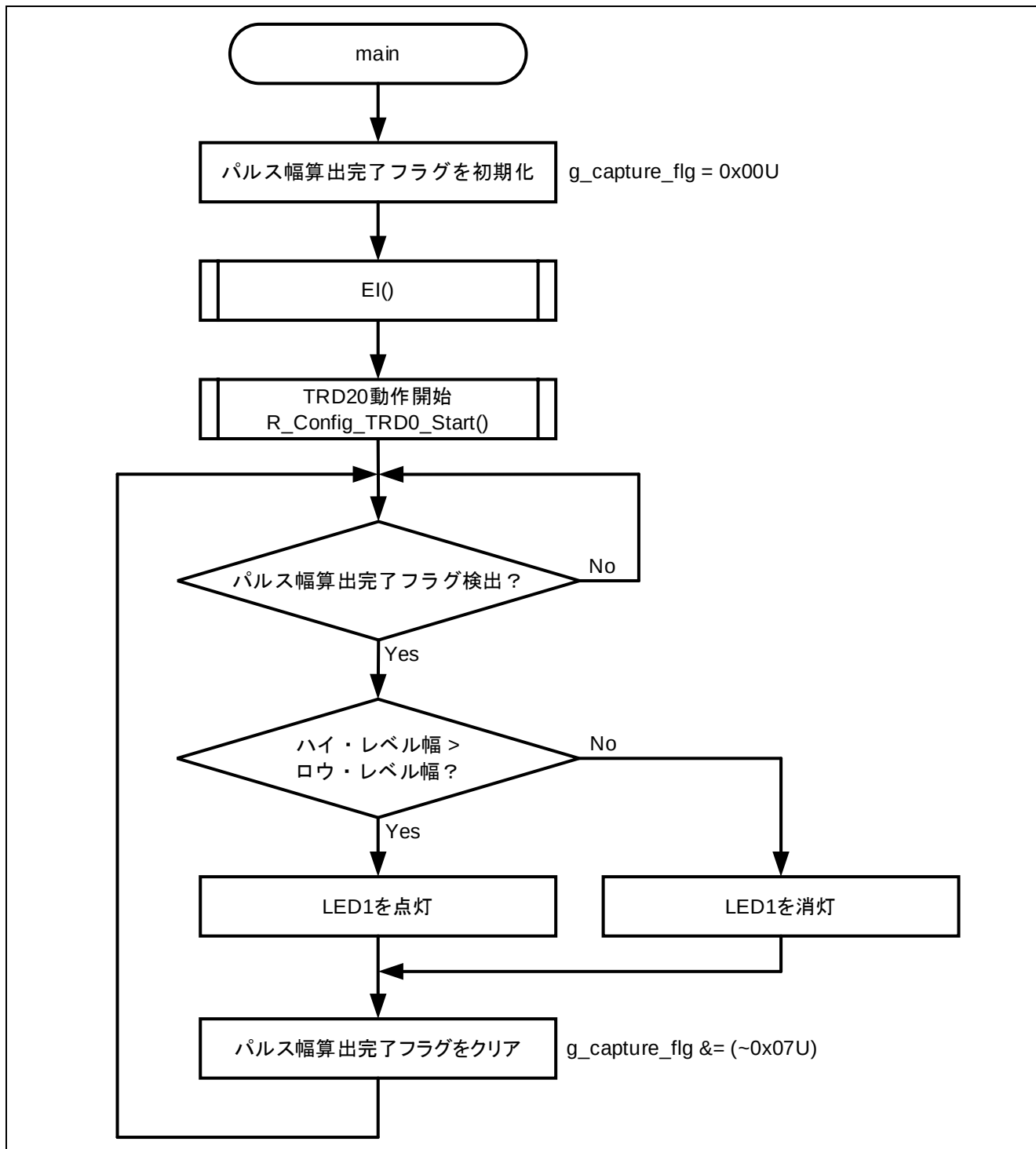
概要	タイマ TD20 割り込み処理
ヘッダ	Config_TRD0.h
宣言	static void __near r_Config_TRD0_trd0_interrupt(void)
説明	TRDIOA0 端子にエッジが入力された時、パルス幅の算出を行います。 また、パルス幅算出完了フラグ g_capture_flg を以下の条件で設定します。 ・オーバフローが検出された時：0 ビット目を 1 (0x01U) ・ハイ・レベル幅の計算結果が格納された時：1 ビット目を 1 (0x02U) ・ロウ・レベル幅の計算結果が格納された時：2 ビット目を 1 (0x04U)
引数	なし
リターン値	なし
備考	なし

4.8 フローチャート

4.8.1 メイン処理

図 4-5 にメイン処理のフローチャートを示します。

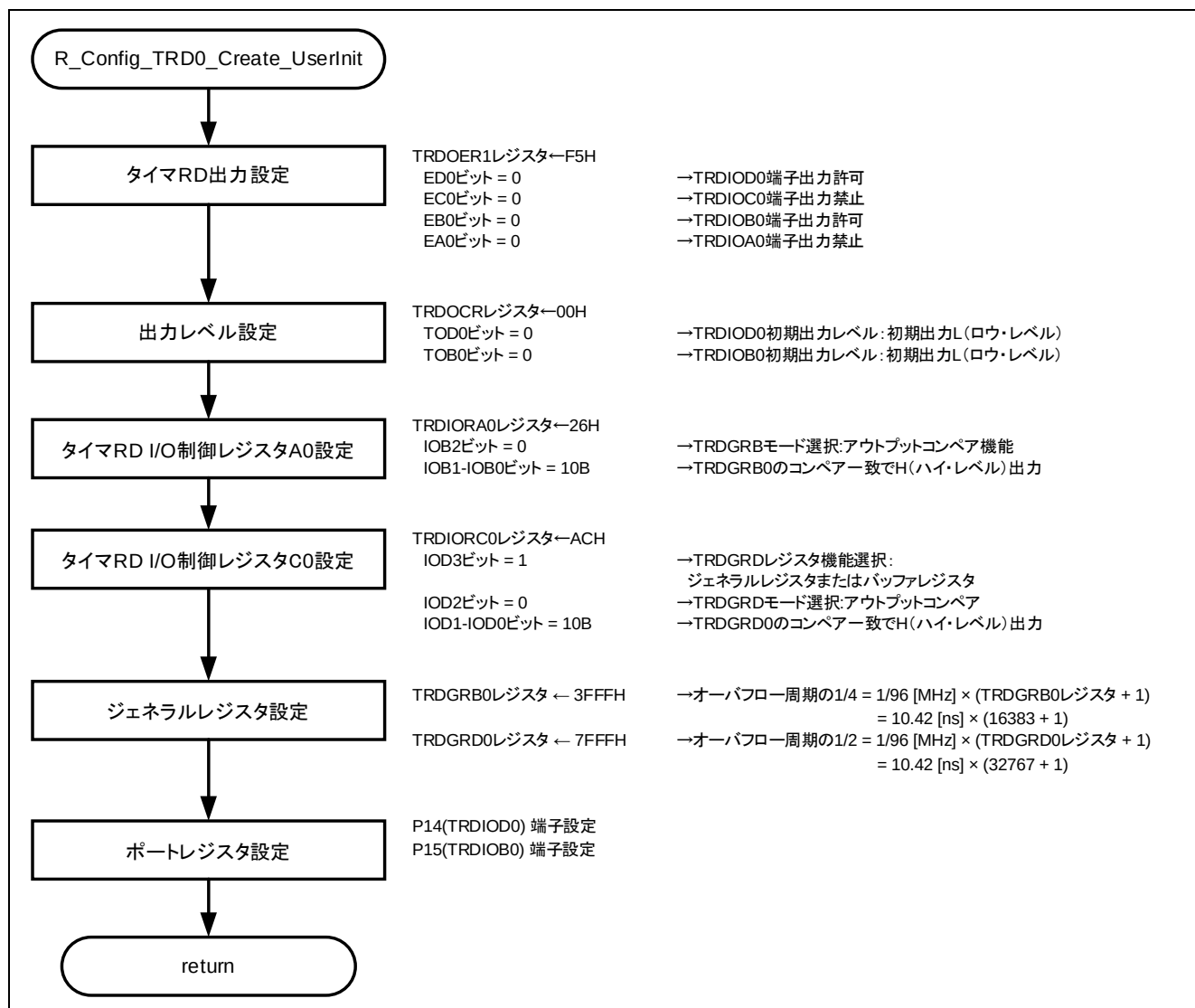
図 4-5 メイン処理



4.8.2 R_Config_TRD0_Create_UserInit 関数

図 4-6 に R_Config_TRD0_Create_UserInit 関数のフローチャートを示します。

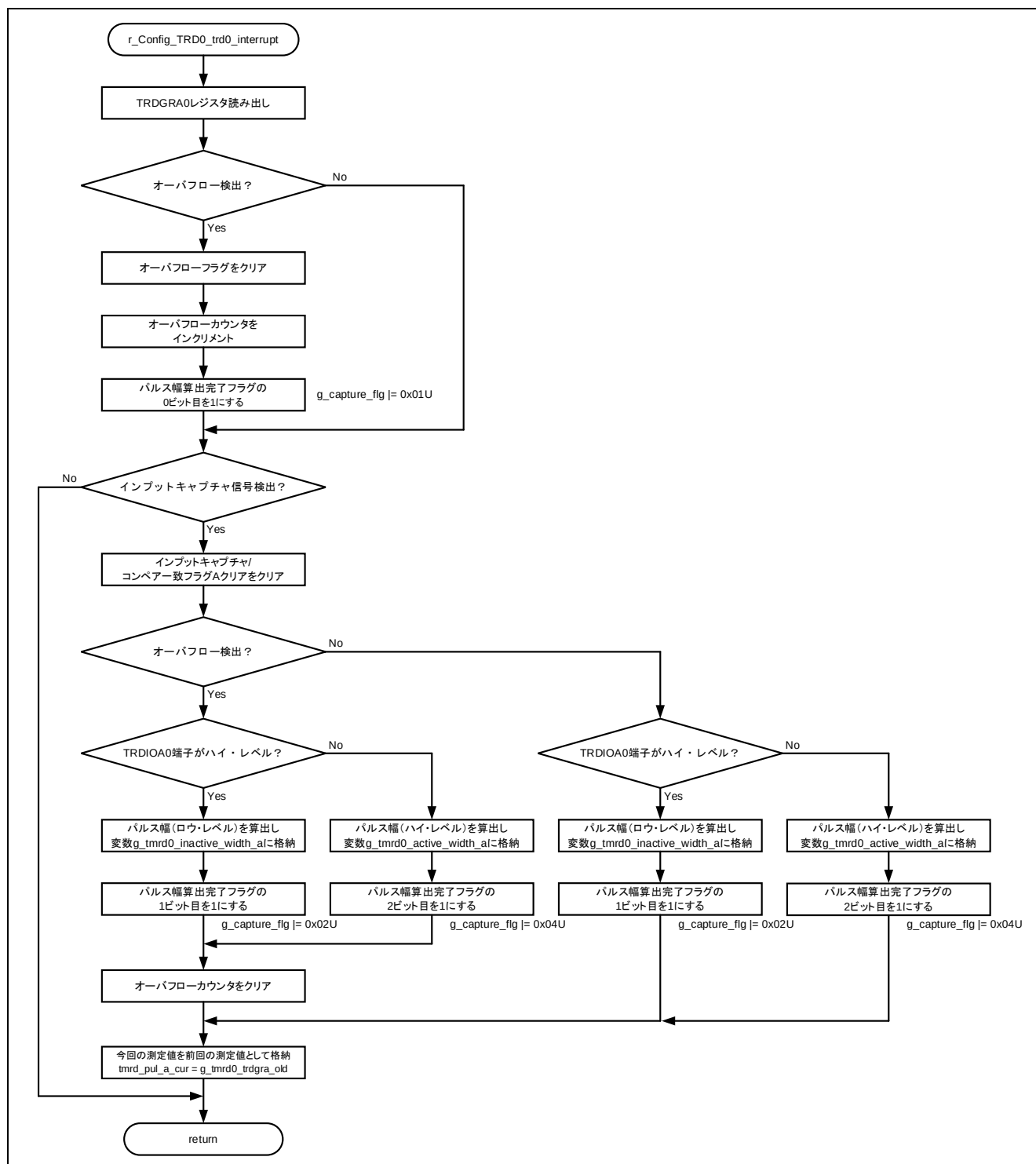
図 4-6 R_Config_TRD0_Create_UserInit 関数



4.8.3 r_Config_TRD0_trd0_interrupt 関数

図 4-7 に r_Config_TRD0_trd0_interrupt 関数のフローチャートを示します。

図 4-7 r_Config_TRD0_trd0_interrupt 関数



5. サンプルコード

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

6. 参考ドキュメント

RL78/G24 ユーザーズマニュアル ハードウェア編（R01UH0961J）

RL78 ファミリ ユーザーズマニュアル ソフトウェア編（R01US0015J）

RL78/G24 Fast Prototyping Board ユーザーズマニュアル（R20UT5091J）

RL78 スマート・コンフィグレータ ユーザーガイド：CS+編（R20AN0580J）

RL78 スマート・コンフィグレータ ユーザーガイド：e² studio 編（R20AN0579J）

RL78 スマート・コンフィグレータ ユーザーガイド：IAR 編（R20AN0581J）

RL78 スマート・コンフィグレータ V1.5.0 リリースノート（R20UT5253J）

（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

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

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問合せ先

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

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023.09.07	－	初版発行

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、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。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）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

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、これに限られません。）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た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しします。
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

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。