

RL78/L23

LCD 表示（時計デモ）

要旨

本アプリケーションノートでは、RL78/L23 の LCD コントローラ／ドライバで LCD パネルとの制御およびサンプルコードの動作を説明します。

サンプルコードでは RL78/L23 の LCD コントローラ／ドライバを使用し、24 時間の時計表示の使用例を示します。リアルタイム・クロック（以下、RTC）にて計時された時刻を LCD 表示データ・メモリ領域に格納し、RTC の定周期割り込み（1 秒毎）に時刻表示を変更します。また、ユーザスイッチの押下とタッチボタンのタッチ検出により時刻調整を行い、調整された時刻の表示を行います。

動作確認デバイス

RL78/L23

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. 動作確認条件	6
3. 周辺機能説明	7
3.1 RL78/L23 LCD コントローラ／ドライバ機能の基本的な特徴	7
3.2 LCD コントローラ／ドライバ機能の各表示モードについて	8
3.3 駆動電圧生成回路について	10
3.3.1 外部抵抗分割方式	11
3.3.2 内部昇圧方式	13
3.3.3 容量分割方式	14
4. ハードウェア説明	15
4.1 ハードウェア構成例	15
4.2 LCD モジュール	16
4.3 静電容量センサユニット	20
4.4 使用端子一覧	21
5. ソフトウェア説明	23
5.1 動作概要	23
5.2 ファイル構成	25
5.3 スマート・コンフィグレータの設定内容	27
5.4 静電容量タッチ設定	37
5.4.1 タッチインタフェース構成	37
5.4.2 構成（メソッド）の設定	37
5.4.3 チューニング結果	38
5.5 定数一覧	39
5.6 変数一覧	40
5.7 関数一覧	42
5.8 関数仕様	43
5.9 フローチャート	48
5.9.1 main 関数	48
5.9.2 qe_touch_main 関数	49
5.9.3 r_clock_demo_start 関数	50
5.9.4 r_userswitch_callback 関数	51
5.9.5 r_rtc_callback 関数	52
5.9.6 r_tau0_1_blink_callback 関数	53
5.9.7 r_tau0_3_delay_callback 関数	53
5.9.8 r_tau0_4_chattering_callback 関数	54
5.9.9 r_tau0_5_continuous_callback 関数	54
6. プロジェクトのインポート方法	55
6.1 e ² studio での手順	55
6.2 CS+での手順	56
6.3 IAR での手順	57

7. デバッグの設定	58
7.1 e ² studio での COM Port 設定	59
7.2 CS+での COM Port 設定	60
7.3 IAR での COM Port 設定	61
8. サンプルコード	62
9. 参考ドキュメント	62
改訂記録	63

1. 仕様

RL78/L23 の LCD コントローラ／ドライバを使用し、24 時間の時計表示を行います。RTC にて設定された時刻を LCD 表示データ・メモリ領域に格納し、RTC の定周期割り込み（1 秒）毎に時刻表示を変更します。

また、ユーザスイッチ（SW）押下により時調整、分調整、秒調整状態に移行し、タッチボタンのタッチ検出により時刻調整を行い、調整された時刻の表示を行います。

時調整、分調整、秒調整状態時は該当する桁の LCD 表示が点滅します。

表 1-1 に使用する周辺機能と用途を、図 1-1 に動作概要を示します。

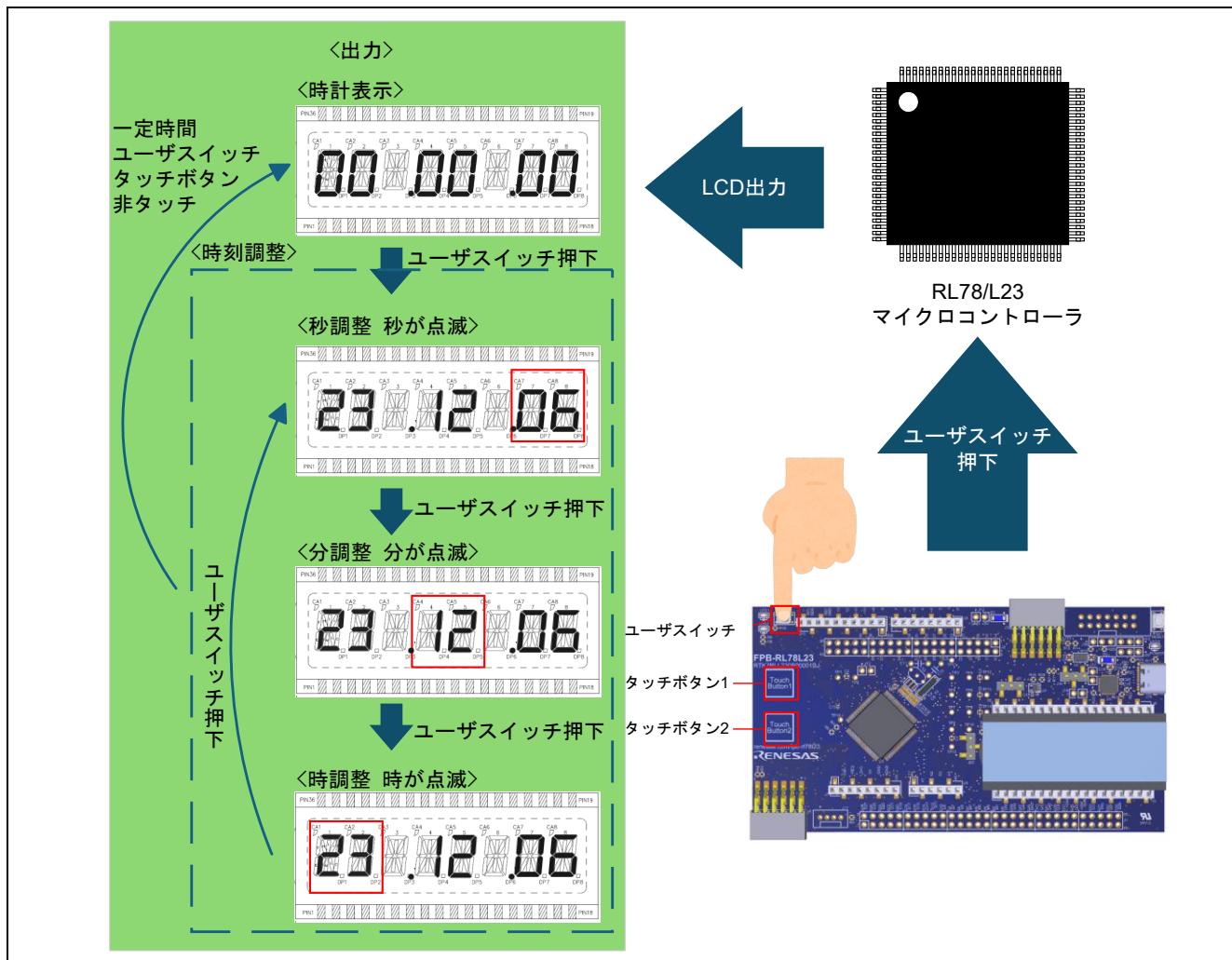
表 1-1 使用する周辺機能と用途

周辺機能	用途
LCD コントローラ／ドライバ	LCD 表示
リアルタイム・クロック（以下、RTC）	時刻のカウント 時計表示状態に遷移する期間の判定
タイマ・アレイ・ユニット 0 チャネル 1（以下、TAU0_1）	LCD の点滅表示
タイマ・アレイ・ユニット 0 チャネル 3（以下、TAU0_3）	待機時間の計測
タイマ・アレイ・ユニット 0 チャネル 4（以下、TAU0_4）	チャタリング対策の時間計測
タイマ・アレイ・ユニット 0 チャネル 5（以下、TAU0_5）	タッチボタンの連続タッチ（長押し）の判定を行うタイマ
静電容量センサユニット (CTSU2La)	タッチ電極に発生する静電容量を計測
外部割り込み INTP0	ユーザスイッチ入力の検出

LCD コントローラ／ドライバは駆動用電源の生成方法として、外部抵抗分割方式、内部昇圧方式、容量分割方式の 3 種類を選択できます。これらの 3 方式の説明は、「3.3 駆動電圧生成回路について」を参照してください。

サンプルコードは LCD 駆動電圧生成回路に容量分割方式を選択しています。

図 1-1 動作概要



ユーザスイッチ押下回数	状態
0	時計表示状態
1	秒調整状態
2	分調整状態
3	時調整状態

タッチボタンのタッチ検出	状態
タッチボタン 1	点滅する桁を+1
タッチボタン 2	点滅する桁を-1

【注】 ユーザスイッチ押下の 4 回目以降は、1 回目からの状態遷移の繰り返しになります。

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/L23(R7F100LPL3CFB)
動作周波数	<ul style="list-style-type: none"> 高速オンチップ・オシレータ・クロック(f_{HOCO}) : 32MHz(標準) CPU/周辺ハードウェア・クロック(f_{CLK}) : 32MHz サブシステム・クロック XR (f_{SX_R}) : 32.768kHz
動作電圧	3.3V LVD動作(V_{LVD0}) : リセット・モード(立ち上がり 2.97V/立ち下がり 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 V8.14.00
Cコンパイラ(CS+)	ルネサス エレクトロニクス製 Renesas CC-RL V1.15.01
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 Version 2025-07
Cコンパイラ(e ² studio)	ルネサス エレクトロニクス製 Renesas CC-RL V1.15.01
統合開発環境(IAR)	IAR システムズ製
Cコンパイラ(IAR)	IAR Embedded Workbench for Renesas RL78 V 5.20.1
スマート・コンフィグレータ (Smart Configurator)	ルネサス エレクトロニクス製 RL78 スマート・コンフィグレータ V1.14.0
静電容量式タッチセンサ対応 開発支援ツール	QE for Capacitive Touch V4.2.0
ボードサポートパッケージ (BSP)	ルネサス エレクトロニクス製 V1.91
TOUCH ドライバ	ルネサス エレクトロニクス製 V2.20
CTSU ドライバ	ルネサス エレクトロニクス製 V2.20
使用ボード	RL78/L23 Fast Prototyping Board (RTK7RLL230S00001BJ)
LCD モジュール	Varitronix VIM-878-DP-FC-S-LV 16 セグ x8 枝 ヘッダ : 36 ピン(18 x 2 列)x1 動作電圧条件 : 3V-4.6V (動作電圧 3.3V 以外で使用する場合は、LCD パネルを取り外した状態でご使用ください)

3. 周辺機能説明

LCD コントローラ／ドライバについて説明を記載します。

詳しい動作は「RL78/L23 ユーザーズマニュアル ハードウェア編 (R01UH1082)」の「第 22 章 LCD コントローラ／ドライバ」を参照してください。

3.1 RL78/L23 LCD コントローラ／ドライバ機能の基本的な特徴

RL78/L23 に内蔵している LCD コントローラ／ドライバには以下のような機能があります。

- A 波形、B 波形の選択が可能
- LCD 駆動電圧生成回路は、内部昇圧／容量分割／外部抵抗分割の切り替えが可能
- 表示データ・レジスタの自動読み出しによるセグメント信号とコモン信号の自動出力が可能
- 昇圧回路動作時に生成する基準電圧を 23 段階から選択可能（コントラスト調整）
- LCD 点滅が可能（点滅表示の切り替えタイミングを選択可）

3.2 LCD コントローラ／ドライバ機能の各表示モードについて

LCD コントローラ／ドライバ機能の各表示モードは、駆動波形、駆動電圧生成回路の組み合わせがあります。表 3-1、表 3-2 に各表示モードにおける最大表示画素数を示します。

表 3-1 A 波形最大表示画素数 (100 ピン製品)

LCD コントローラ／ドライバ用 駆動波形	LCD コントローラ ／ドライバ用 駆動電圧生成回路	バイアス法	時分割	最大表示画素数
A 波形	外部抵抗分割	-	スタティック	56 (56 セグメント × 1 コモン)
			2	112 (56 セグメント × 2 コモン)
		1/2	3	168 (56 セグメント × 3 コモン)
			3	
		1/3	4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)
			8	
	内部昇 圧	V _{L1} リ ファレ ンス	1/3	168 (56 セグメント × 3 コモン)
			4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)
		V _{L2} リ ファレ ンス	1/4	324 (54 セグメント × 6 コモン)
			6	416 (52 セグメント × 8 コモン)
			8	
			1/3	168 (56 セグメント × 3 コモン)
容量分 割	V _{DD} リ ファレ ンス	1/3	4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)
		1/4	6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)
	V _{L4} リ ファレ ンス	1/3	3	168 (56 セグメント × 3 コモン)
			4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)

表 3-2 B 波形最大表示画素数 (100 ピン製品)

LCD コントローラ／ドライバ用 駆動波形	LCD コントローラ ／ドライバ用 駆動電圧生成回路	バイアス法	時分割	最大表示画素数
B 波形	外部抵抗分割	1/3	3	168 (56 セグメント × 3 コモン)
			4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)
		1/4	8	
	内部昇 圧	1/3	3	168 (56 セグメント × 3 コモン)
			4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)
		1/4	8	
	容量分 割	1/3	3	168 (56 セグメント × 3 コモン)
			4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)
		1/4	8	
	V _{L4} リ ファレ ンス	1/3	3	168 (56 セグメント × 3 コモン)
			4	224 (56 セグメント × 4 コモン)
			6	324 (54 セグメント × 6 コモン)
			8	416 (52 セグメント × 8 コモン)

3.3 駆動電圧生成回路について

RL78/L23 は LCD 駆動用電源の生成方法として、外部抵抗分割方式、内部昇圧方式、容量分割方式の 3 種類を選択できます。以降に各方式の特徴を示します

表 3-3 LCD 駆動方式と使用用途について

LCD 駆動方式	特徴／使用方法			使用用途
	ドライブ能力	動作電流	駆動電圧	
外部抵抗分割方式	高い	標準	V_{DD} に依存	大型 LCD/AC 電源セット向け LCD 駆動能力が高く、また駆動電圧を抵抗分割して生成するため低コストで実現できます。
	大型 LCD にも対応			外付けの抵抗を使用して分圧、LCD 駆動電圧を生成します。外部から電圧を入力できるので、外部にて動作電流や駆動能力を、抵抗によって調整できます。
内部昇圧方式	標準	小さい	一定	電池セット向け 動作電流も小さく、電池電圧低下時にも駆動電圧は一定で LCD の表示が薄くなりません。 内部で基準電圧を生成し、外付けのコンデンサを使用して昇圧します。 なお、基準電圧をソフトウェアで調整できるので、LCD のコントラスト調整が可能です。 (RL78/L23 では、23 段階の設定が可能です)
			電池の電圧が低下しても変わらないので、表示が薄くならない。	
容量分割方式	標準	さらに小さい	基準電圧が V_{DD} 時、 V_{DD} に依存	電池セット向け 動作電流が最も小さい方式。電池電圧低下時には、LCD の表示は薄くなります。 電池残量とあわせて、表示を薄くしたい場合はそのままご使用できます。 電池電圧低下時に表示を薄くしたくない場合は、電池電圧低下時に内部昇圧方式に切り替える方法があります。 なお、容量分割方式の外部回路で内部昇圧方式は動作可能です。
			電源電圧の降下にあわせて、表示が薄くなる。	

3.3.1 外部抵抗分割方式

大型 LCD/AC 電源セットに適しています。

LCD 駆動能力が高く、また駆動電圧を抵抗分割して生成するため低コストを実現できます。

外付けの抵抗を使用して分圧し、LCD 駆動電圧を生成します。外部から電圧を入力できるので、外部の抵抗によって動作電流や駆動能力を調整できます。

図 3-1、図 3-2 は、外部抵抗接続方式の接続例です。

図 3-1 外部抵抗接続方式の接続例 (1/2)

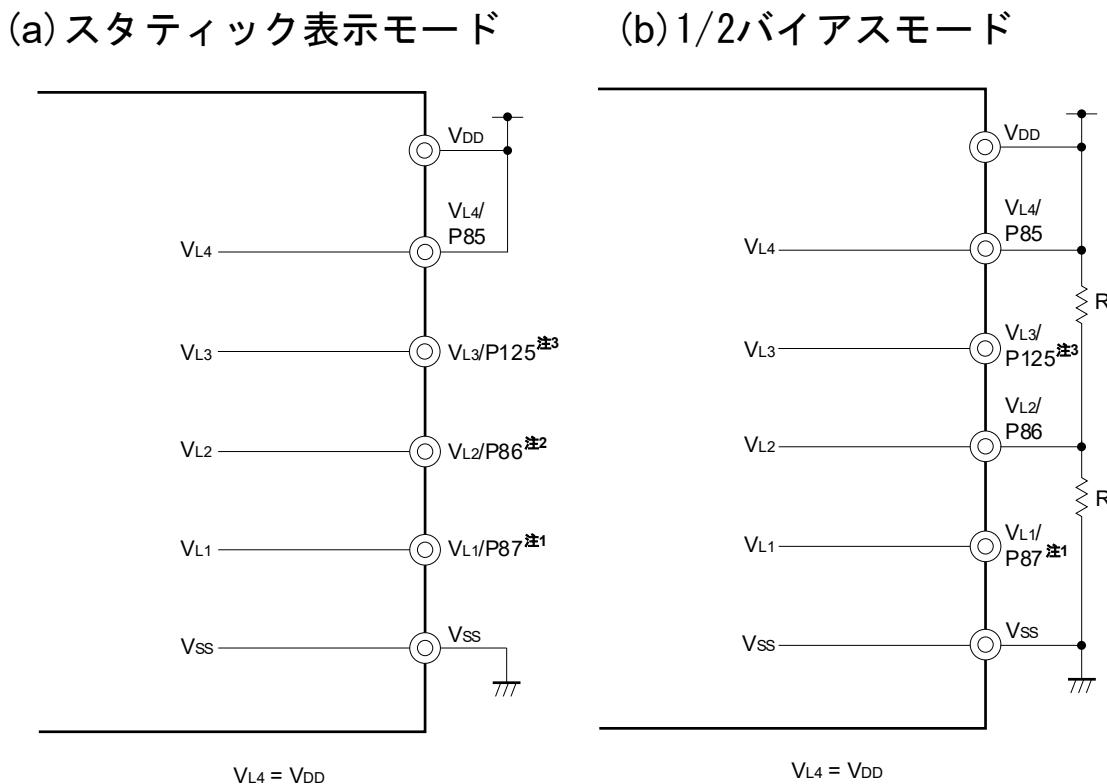
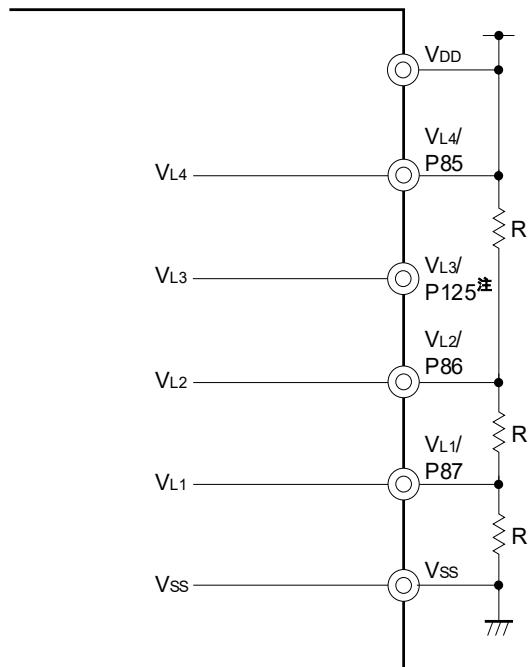


図 3-1(a)と(b)の注意事項を以下に示します。

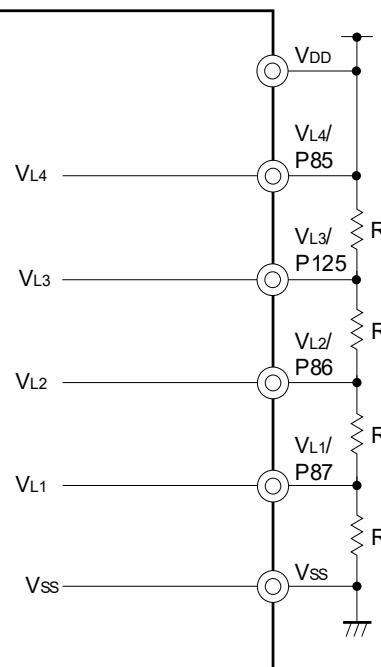
- 【注】 1. VL₁ は、ポート (P87) として使用できます。
 2. VL₂ は、ポート (P86) として使用できます。
 3. VL₃ は、ポート (P125) として使用できます。

図 3-2 外部抵抗接続方式の接続例 (2/2)

(c) 1/3バイアス法



(d) 1/4バイアス法



【注】 VL₃ は、ポート (P125) として使用できます。

外部抵抗分解用抵抗 R の参考値は、10 kΩ～1 MΩ です。また VL₁～VL₄ 端子の電位を安定させる場合には、必要に応じて、VL₁～VL₄ 端子～GND 間にコンデンサを接続してください。これらの参考値は、0.47 μF 程度です。使用する LCD パネル、セグメント端子数、コモン端子数、フレーム周波数、使用環境に依存します。システムにあわせた評価を十分に行った上で、値を調整して決定してください。

3.3.2 内部昇圧方式

電池セットに適しています。

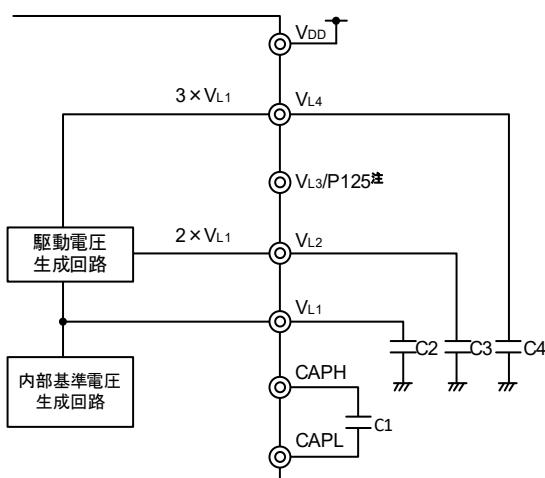
動作電流も小さく、電池電圧低下時にも駆動電圧は一定で LCD の表示が薄くなりません。

内部で基準電圧を生成して、外付けのコンデンサを使用して昇圧します。なお、基準電圧をソフトウェア (LCD 昇圧制御レジスタ (V_{LCD})) で調整できるので、LCD のコントラスト調整が可能です。(RL78/L23 では、23 段階の設定が可能です。)

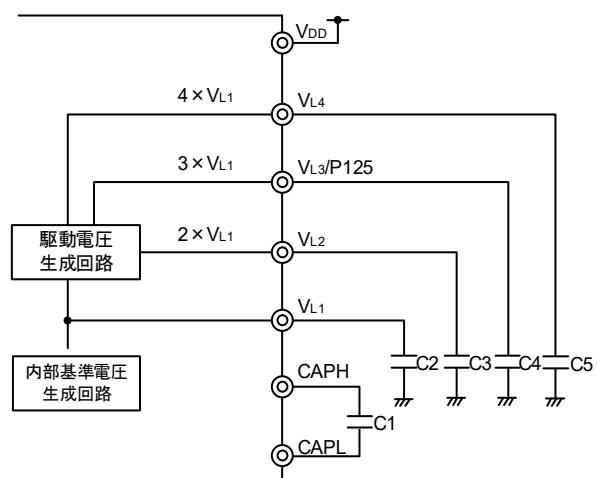
図 3-3 は内部昇圧方式の接続例です。

図 3-3 内部昇圧方式の接続例

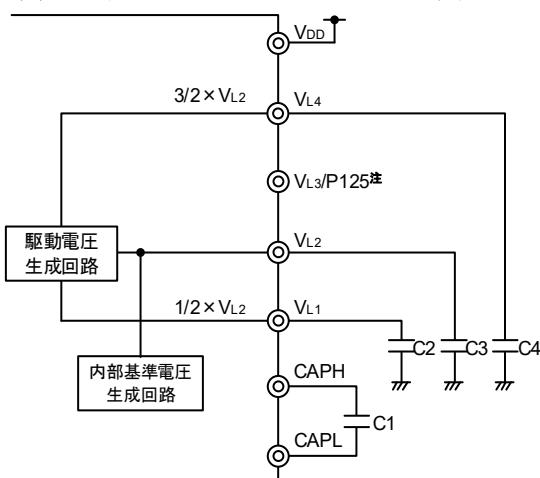
(a) V_{L1} リファレンス 1/3 バイアス法



(b) V_{L1} リファレンス 1/4 バイアス法



(c) V_{L2} リファレンス 1/3 バイアス法



【注】 V_{L3} は、ポート (P125) として使用できます。

【備考】 なるべくリークの少ないコンデンサをご使用ください。なお、C1 は無極性コンデンサにしてください。

3.3.3 容量分割方式

電池セットに適しています。

動作電流が最も小さい方式です。電池電圧低下時には、LCD の表示は薄くなります。

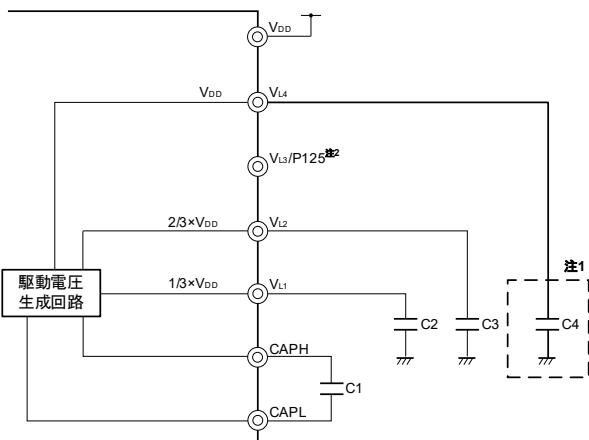
電池残量とあわせて、表示を薄くしたい場合はそのままご使用できます。

電池電圧降下時に表示を薄くしたくない場合は、電池電圧降下時に内部昇圧方式に切り替える方法があります。なお、容量分割方式の外部回路で内部昇圧方式は動作可能です。

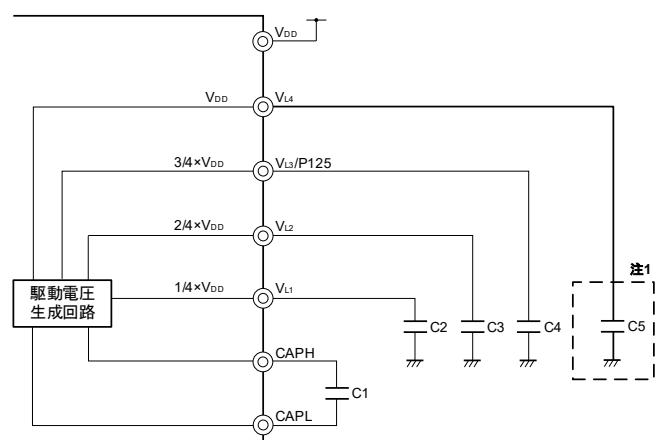
図 3-4 は容量分割方式の接続例です。

図 3-4 容量分割方式の接続例

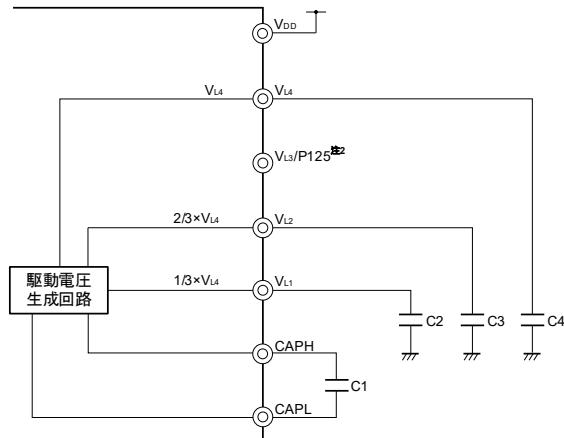
(a) V_{DD} リファレンス 1/3バイアス法



(b) V_{DD} リファレンス 1/4バイアス法



(c) V_{L4} リファレンス 1/4バイアス法



【注】 1. V_{L4} は、内部で V_{DD} と接続されているため、コンデンサは不要です。ただし、内部昇圧方式に切り替えて使用する場合は必要となります。また、容量分割方式のみで使用する場合も、LCD 供給電圧を安定する方法としてコンデンサを接続してもかまいません。
2. V_{L3} は、ポート (P125) として使用できます。

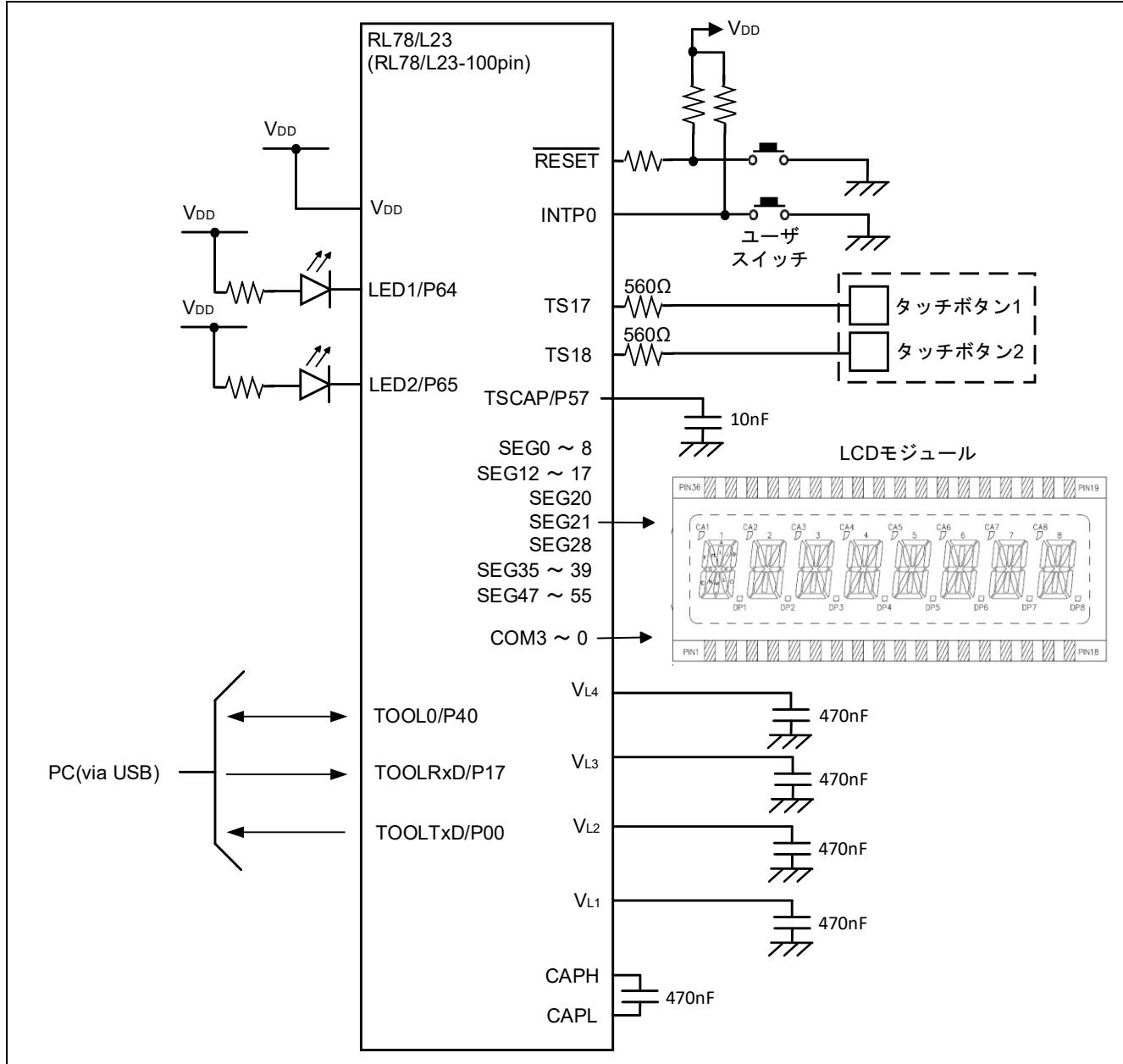
【備考】 なるべくリークの少ないコンデンサをご使用ください。なお、C1 は無極性コンデンサにしてください。

4. ハードウェア説明

4.1 ハードウェア構成例

図 4-1 に本アプリケーションノートで使用するハードウェア構成例を示します。

図 4-1 ハードウェア構成例



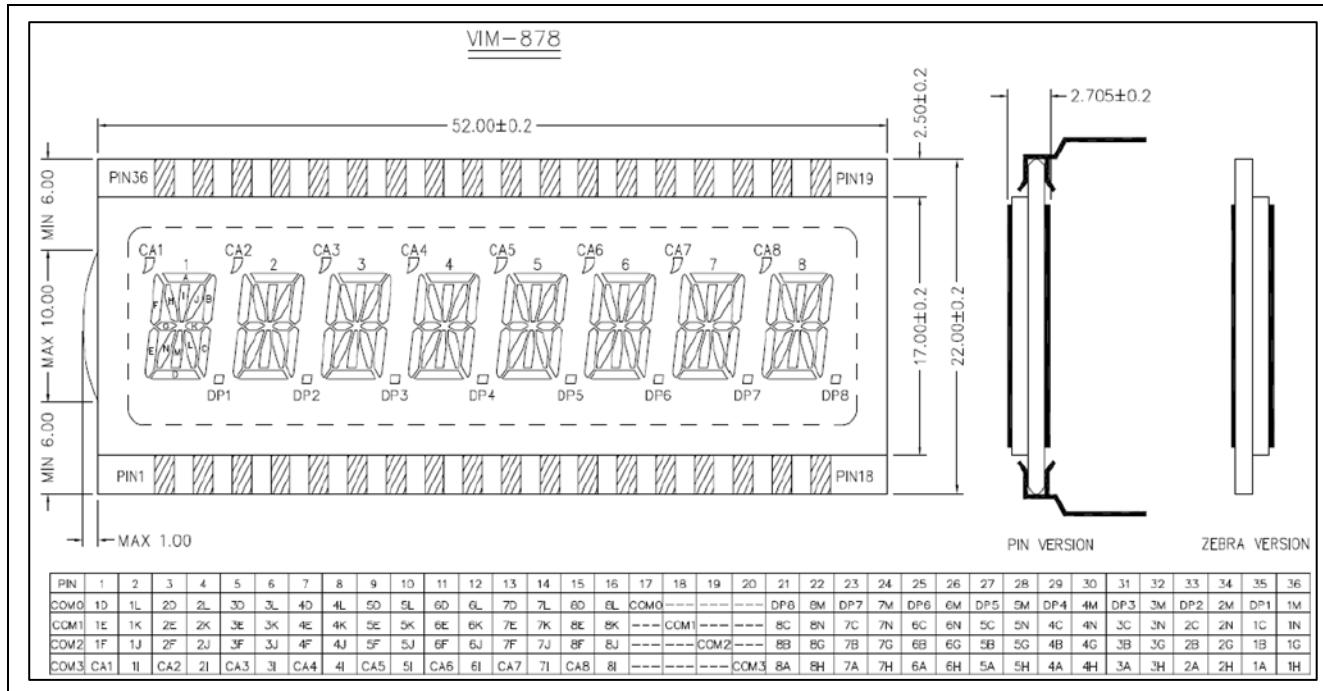
- 【注意】
- この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい）。
 - V_{DD} は LVD にて設定したリセット解除電圧 (V_{LVD}) 以上にしてください。

4.2 LCD モジュール

本サンプルコードで使用する LCD モジュールについて説明します。

RL78/L23 Fast Prototyping Board には、LCD パネル (16 セグメント 8 枝) がソケットにて搭載されております。図 4-2 に LCD パネルのイメージと端子割当表、表 4-1 と表 4-2 に LCD パネルと RL78/L23 の接続を示します。

図 4-2 LCD パネルと端子割り当て表



(出典 : [Datasheet for VIM-878-DP-FC-S-LV Varitronix Optoelectronics | Octopart](#))

表 4-1 J5 ソケットと LCD モジュールの接続表

J5 ソケット	LCD パネル ピン番号	信号名	ポート	信号名とポート	ピン番号	LCD_Header
1	LCD_1	SEG8	P54	P54(SEG8)	54	J1_21
2	LCD_2	SEG7	P53	P53(SEG7)	55	J1_22
3	LCD_3	SEG6	P52	P52(SEG6)	56	J1_23
4	LCD_4	SEG5	P51	P51(SEG5)	57	J1_24
5	LCD_5	SEG4	P50	P50(SEG4)	58	J1_25
6	LCD_6	SEG3	P97	P97(SEG3)	59	J1_26
7	LCD_7	SEG2	P96	P96(SEG2)	60	J1_27
8	LCD_8	SEG1	P95	P95(SEG1)	61	J1_28
9	LCD_9	SEG0	P94	P94(SEG0)	62	J1_29
10	LCD_10	SEG50	P07	P07(SEG50)	69	J1_34
11	LCD_11	SEG49	P06	P06(SEG49)	70	J1_35
12	LCD_12	SEG48	P05	P05(SEG48)	71	J1_36
13	LCD_13	SEG47	P04	P04(SEG47)	72	J1_37
14	LCD_14	SEG39	P14	P14(SEG39)	83	J1_45
15	LCD_15	SEG38	P13	P13(SEG38)	84	J1_46
16	LCD_16	SEG37	P12	P12(SEG37)	85	J1_47
17	LCD_17	COM0	P90	P90(COM0)	66	J1_33
18	LCD_18	COM1	P91	P91(COM1)	65	J1_32

表 4-2 J6 ソケットと LCD モジュールの接続表

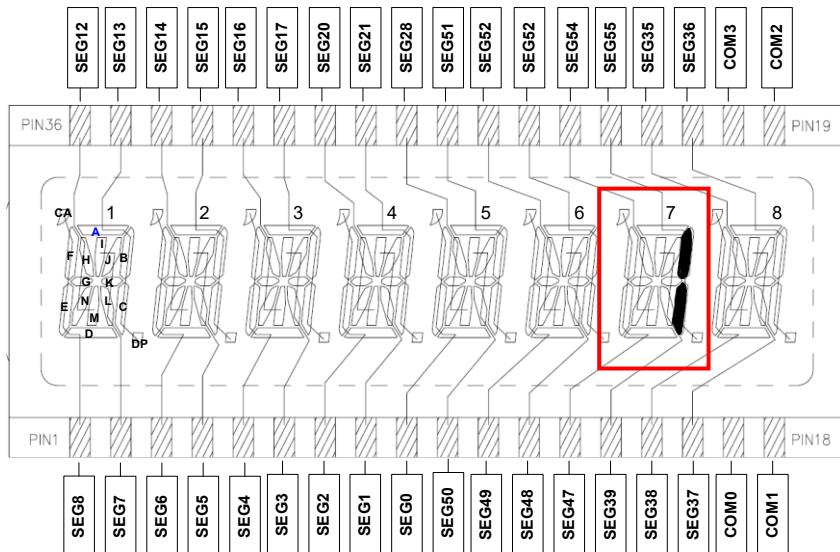
J6 ソケット	LCD パネル ピン番号	信号名	ポート	信号名と ポート	ピン番号	LCD_Header
1	LCD_19	COM2	P92	P92(COM2)	64	J1_31
2	LCD_20	COM3	P93	P93(COM3)	63	J1_30
3	LCD_21	SEG36	P11	P11(SEG36)	86	J1_48
4	LCD_22	SEG35	P10	P10(SEG35)	87	J1_49
5	LCD_23	SEG55	P145	P145(SEG55)	88	J1_50
6	LCD_24	SEG54	P144	P144(SEG54)	89	J1_51
7	LCD_25	SEG53	P143	P143(SEG53)	94	J1_56
8	LCD_26	SEG52	P142	P142(SEG52)	95	J1_57
9	LCD_27	SEG51	P141	P141(SEG51)	96	J1_58
10	LCD_28	SEG28	P130	P130(SEG28)	2	J1_1
11	LCD_29	SEG21	P31	P31(SEG21)	39	J1_8
12	LCD_30	SEG20	P30	P30(SEG20)	40	J1_9
13	LCD_31	SEG17	P75	P75(SEG17)	43	J1_12
14	LCD_32	SEG16	P74	P74(SEG16)	44	J1_13
15	LCD_33	SEG15	P73	P73(SEG15)	45	J1_14
16	LCD_34	SEG14	P72	P72(SEG14)	46	J1_15
17	LCD_35	SEG13	P71	P71(SEG13)	47	J1_16
18	LCD_36	SEG12	P70	P70(SEG12)	48	J1_17

図 4-3 に 8 桁 LCD パネルにおける表示方法の一例と、それを実現するためのセグメント信号とセグメントの対応を示します。

図 4-3 セグメント信号とセグメントの対応

DISIT 7に「1」を表示する例を示します。

表4-1, 表4-2の端子割当表を参考に、次の処理を行います。



LCDモジュールのPIN23にセグメント信号SEG55が接続

端子割当表より、

PIN23の「7C」とCOM1信号が結線

PIN23の「7B」とCOM2信号が結線

SEG55のレジスタに06Hを設定すると「7C」と「7B」がそれぞれ点灯し、DISIT 7の表示は「1」となります。

PIN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
COM0	1D	1L	2D	2L	3D	3L	4D	4L	5D	5L	6D	6L	7D	7L	8D	8L	COM0	---	---	---	DP8	8M	DP7	7M	DP6	6
COM1	1E	1K	2E	2K	3E	3K	4E	4K	5E	5K	6E	6K	7E	7K	8E	8K	---	COM1	---	---	8C	8N	7C	7N	6C	6
COM2	1F	1J	2F	2J	3F	3J	4F	4J	5F	5J	6F	6J	7F	7J	8F	8J	---	---	COM2	---	8B	8G	7B	7G	6B	6
COM3	CA1	1I	CA2	2I	CA3	3I	CA4	4I	CA5	5I	CA6	6I	CA7	7I	CA8	8I	---	---	COM3	8A	8H	7A	7H	6A	6	

(加筆 : [VIM-878.pdf](#))

表 4-3 セグメントとコモンの対応 (DISIT 7)

LCD モジュール PIN 番号 ^注	LCD 表示データ・ レジスタ ^注	COM3	COM2	COM1	COM0
		bit3	bit2	bit1	bit0
PIN23	SEG55	A	B	C	DP
PIN13	SEG47	CA	F	E	D
PIN14	SEG39	I	J	K	L
PIN24	SEG54	H	G	N	M

【注】 他の DISIT については、対応する PIN 番号およびセグメント信号のレジスタ名に読み替えてください。

4.3 静電容量センサユニット

本サンプルコードで使用する静電容量センサユニットについて説明します。

RL78/L23 Fast Prototyping Board には、タッチボタン 1、タッチボタン 2 の 2 つの電極が搭載されています。

CTSU モジュール、TOUCH モジュール、静電容量式タッチセンサ対応開発支援ツール QE for Capacitive Touch を使用することで、タッチボタンを使用可能です。タッチボタンの使用方法の詳細は、参考として「静電容量センサマイコン静電容量タッチ導入ガイド（R30AN0424）」を参照してください。

4.4 使用端子一覧

表 4-4 使用端子と機能 (1/2)

端子名	入出力	機能
P137/INTP0	入力	ユーザスイッチ入力を検出し、時、分、秒の調整状態に移行
P57/TSCAP	-	計測用 2 次電源コンデンサ接続端子
P56/TS17	入出力	タッチボタン 1 の入力を検出し、時、分の LCD 表示をアップカウント
P55/TS18	入出力	タッチボタン 2 の入力を検出し、時、分の LCD 表示をダウンカウント
P94/SEG0	出力	LCD コントローラ／ドライバのセグメント信号
P95/SEG1		
P96/SEG2		
P97/SEG3		
P50/SEG4		
P51/SEG5		
P52/SEG6		
P53/SEG7		
P54/SEG8		
P70/SEG12		
P71/SEG13		
P72/SEG14		
P73/SEG15		
P74/SEG16		
P75/SEG17		
P30/SEG20		
P31/SEG21		
P130/SEG28		
P10/SEG35		
P11/SEG36		
P12/SEG37		
P13/SEG38		
P14/SEG39		
P04/SEG47		
P05/SEG48		
P06/SEG49		
P07/SEG50		
P141/SEG51		
P142/SEG52		
P143/SEG53		
P144/SEG54		
P145/SEG55		
P90/COM0	出力	LCD コントローラ／ドライバのコモン信号
P91/COM1		
P92/COM2		
P93/COM3		
P87/V _{L1}	-	LCD 駆動用電圧
P86/V _{L2}		
P85/V _{L4}		

表 4-5 使用端子と機能 (2/2)

端子名	入出力	機能
P126/CAPL	-	LCD コントローラ／ドライバ用コンデンサ接続
P127/CAPH		
P40/TOOL0	入出力	COM Port デバッグ
P17/TOOLRxD	入力	COM Port デバッグ
P00/TOOLTxD	出力	COM Port デバッグ

5. ソフトウェア説明

5.1 動作概要

このサンプルコードでは、LCD コントローラ／ドライバを使用し、24 時間の時計表示を行います。RTC にて計時された時刻を LCD 表示データ・メモリ領域に格納し、RTC の定周期割り込み（1 秒）毎に時刻表示を変更します。

また、ユーザスイッチ押下により秒調整、分調整、時調整、秒調整の順で調整対象が循環的に切り替わり、タッチボタンのタッチ検出により時刻調整を行い、設定された時刻の表示を行います。秒調整、時調整、分調整状態時は該当する桁の LCD 表示が点滅します。

初期設定では、クロック周波数の選択、入出力ポートの設定、RTC の設定、LCD コントローラ／ドライバの設定などを行います。

初期設定完了後は LCD に「RL78/L23」、「CLK DEMO」と表示され、待機状態に移行します。ユーザスイッチの短押し（INTP0 の両エッジ検出）により待機状態が解除され、時刻表示を行います。RTC の定期割り込みが発生した場合、時刻表示を 1 秒ごとに変更します。ユーザスイッチを押下した場合、任意で表示する時刻の設定が可能です。

タッチボタンは時計表示状態では無効。時調整、分調整、秒調整状態時のみ有効となります。計測周期は 20ms です。

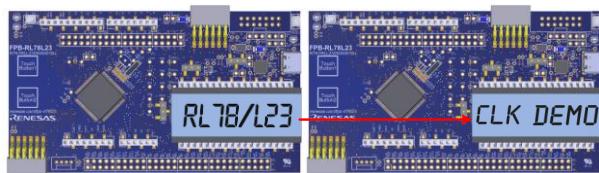
ユーザスイッチ 1 回目の押下で秒調整、ユーザスイッチ 2 回目の押下で分調整、ユーザスイッチ 3 回目の押下で時調整を行い、ユーザスイッチ 4 回目の押下で秒調整へ循環的に切り替わります。ユーザスイッチの押下、及びタッチボタンのタッチ検出が一定時間ないと、時計機能表示に戻り、設定された時刻を表示します。また、時調整状態でタッチボタン 1 のタッチ検出により 1 時間増加、タッチボタン 2 のタッチ検出により 1 時間減少、分調整状態でタッチボタン 1 のタッチ検出により 1 分増加、タッチボタン 2 のタッチ検出により 1 分減少、秒調整状態でタッチボタン 1 のタッチ検出により 1 秒増加、タッチボタン 2 のタッチ検出により 1 秒減少となります。タッチボタンの連続タッチにより、値が連続で変更されます。

詳細については、図 5-1 の状態遷移図（ステート・チャート）に示します。

図 5-1 状態遷移図

①時計デモ開始

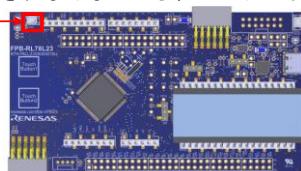
「RL78/L23」→「CLK DEMO」と表示される



②待機状態

LCDが表示されなくなったら、ユーザスイッチの短押しを待機

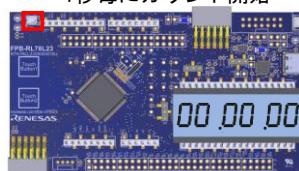
ユーザスイッチ



ユーザスイッチを短押し

③時計表示

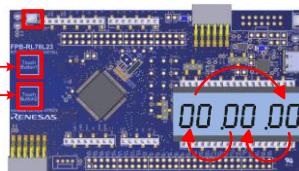
1秒毎にカウント開始



一定時間非タッチ

ユーザスイッチを短押し

④時刻設定：タッチ有効

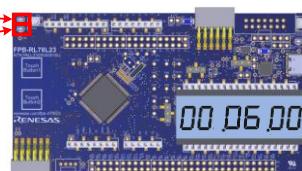
タッチボタン1
タッチボタン2

- 1.ユーザスイッチをタッチすると点滅する時刻表示が秒→分→時→秒と変更される。
- 2.タッチボタン1をタッチ検出すると点滅する時刻表示の値が+1される。タッチボタン2をタッチ検出すると-1される。
- 3.タッチボタン長押しで時刻設定が連続で変更される。
- 4.両タッチボタンをタッチするとLED1,2が点灯し、非タッチ判定されるまで次のタッチ動作をしない。

⑤時刻設定：時刻変更

LED1
LED2

タッチボタンのタッチ検出



- 1.タッチ検出したボタンによってLEDが点灯する。(タッチボタン1:LED1、タッチボタン2:LED2)
- 2.非タッチを判定すると消灯

5.2 ファイル構成

表 5-1、表 5-2 にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成される BSP 環境のファイルは除きます。

表 5-1 ファイル構成 (1/2)

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
¥clock_demo<DIR>	サンプルコードのフォルダ	
¥src<DIR>	ソースフォルダ	
¥clock_demo.c	時計デモソースファイル	
¥clock.c	時計ソースファイル	
¥clock.h	時計ヘッダファイル	
¥lcd_segdata.c	LCD 表示ソースファイル	
¥smc_gen<DIR>	スマート・コンフィグレータ生成コンポーネント格納フォルダ	✓
¥Config_INTC<DIR>	外部割り込みコンポーネントフォルダ	✓
¥Config_LCD<DIR>	LCD コンポーネントフォルダ	✓
¥Config_PORT<DIR>	PORT コンポーネントフォルダ	✓
¥Config_RTC<DIR>	RTC コンポーネントフォルダ	✓
¥Config_TAU0_1<DIR>	TAU0_1 コンポーネントフォルダ	✓
¥Config_TAU0_3<DIR>	TAU0_3 コンポーネントフォルダ	✓
¥Config_TAU0_4<DIR>	TAU0_4 コンポーネントフォルダ	✓
¥Config_TAU0_5<DIR>	TAU0_5 コンポーネントフォルダ	✓
¥r_ctsu<DIR>	CTSU ドライバフォルダ	✓
¥rm_touch<DIR>	TOUCH ドライバフォルダ	✓

表 5-2 ファイル構成 (2/2)

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
¥clock_demo<DIR>	サンプルコードのフォルダ	
¥qe_gen<DIR>	QE for Capacitive Touch 生成ファイル	
¥qe_touch_config.c	QE 生成設定ソースファイル	
¥qe_touch_config.h	QE 生成設定ヘッダファイル	
¥qe_touch_define.h	QE 生成定義ヘッダファイル	
¥qe_touch_sample.c	タッチ動作を含むメイン関数ソースファイル	
¥QE-Touch<DIR>	QE 構成フォルダ	

表 5-1、表 5-2 の注意事項を以下に示します。

【注】 IAR 版のサンプルコードは構成が異なります。詳細は IAR 版のサンプルコードを確認してください。また、clock_demo.ipcf を格納しています。詳細は、「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。

5.3 スマート・コンフィグレータの設定内容

図 5-2 にスマート・コンフィグレータのクロック設定について示します。

図 5-2 クロック設定

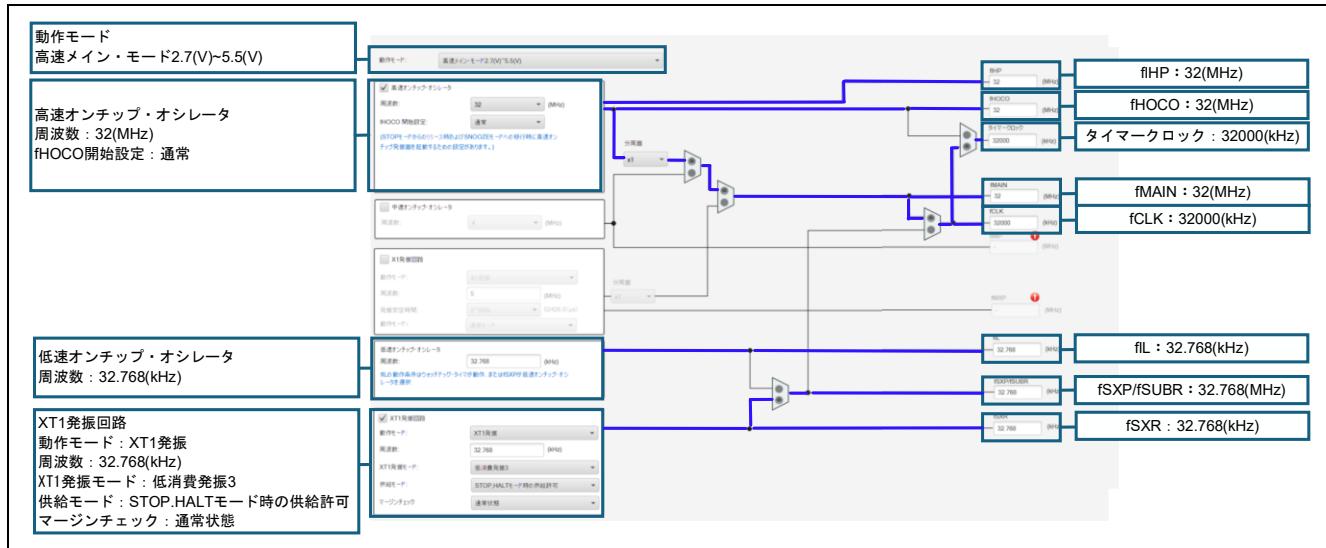


図 5-3 にスマート・コンフィグレータのシステム設定について示します。

図 5-3 システム設定

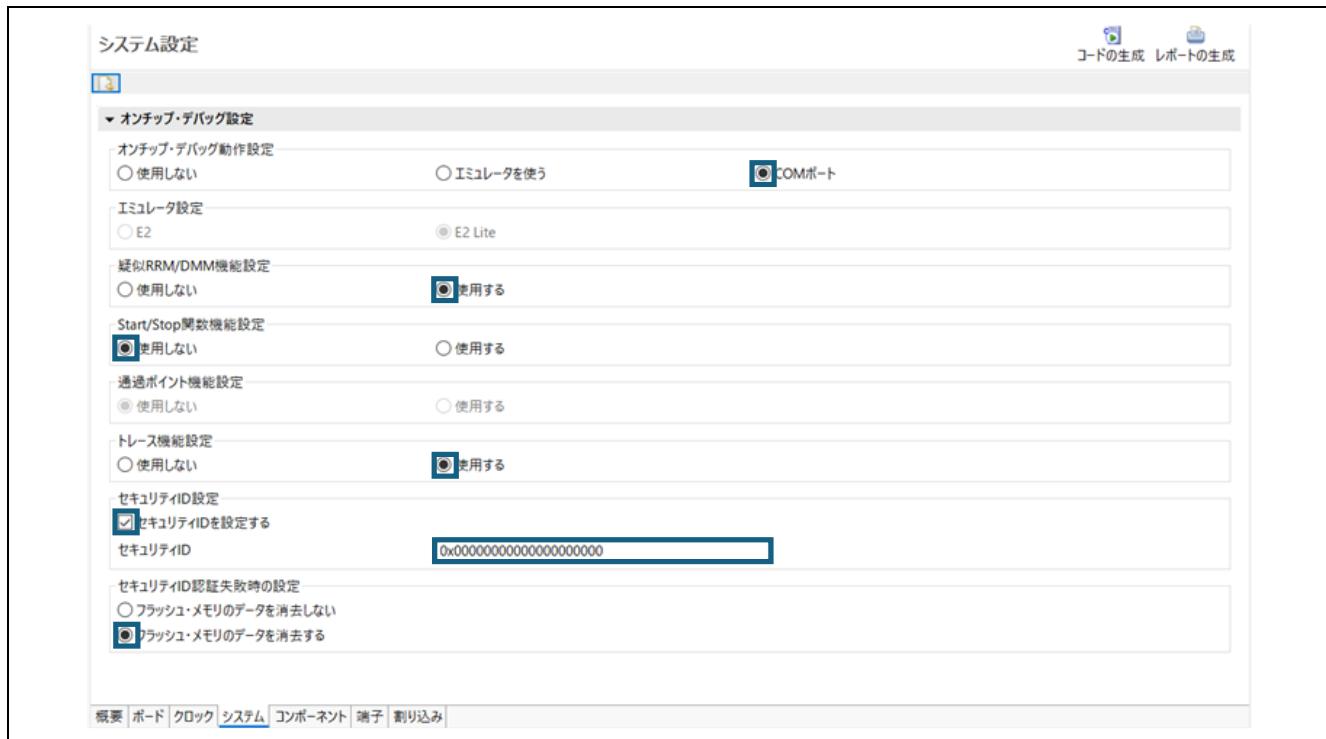


図 5-4 に LVD コンポーネント (Config_LVD0) の設定を示します。

図 5-4 LVD コンポーネント (Config_LVD0) の設定

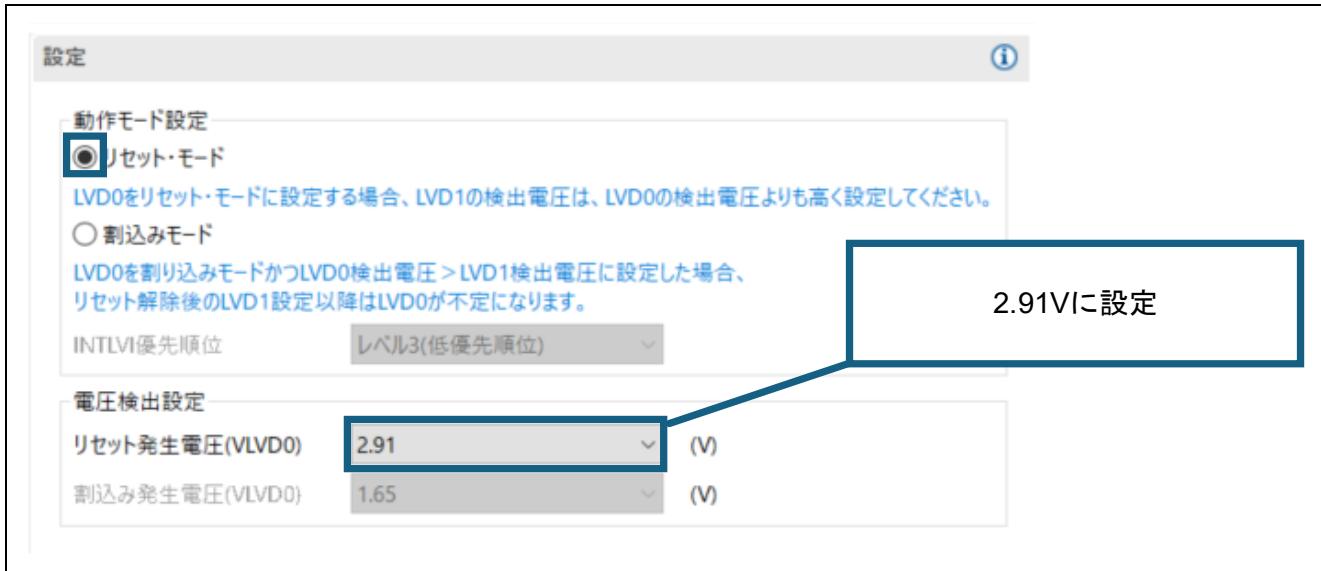


図 5-5 に外部割り込みコンポーネント (Config_INTC) の設定を示します。

図 5-5 外部割り込みコンポーネント (Config_INTC) の設定



図 5-6 に RTC コンポーネント (Config_RTC) の設定を示します。

図 5-6 RTC コンポーネント (Config_RTC) の設定



図 5-7 に TAU0_1 コンポーネント (Config_TAU0_1) の設定を示します。

図 5-7 TAU0_1 コンポーネント (Config_TAU0_1) の設定



図 5-8 に TAU0_3 コンポーネント (Config_TAU0_3) の設定を示します。

図 5-8 TAU0_3 コンポーネント (Config_TAU0_3) の設定



図 5-9 に TAU0_4 コンポーネント (Config_TAU0_4) の設定を示します。

図 5-9 TAU0_4 コンポーネント (Config_TAU0_4) の設定



図 5-10 に TAU0_5 コンポーネント (Config_TAU0_5) の設定を示します。

図 5-10 TAU0_5 コンポーネント (Config_TAU0_5) の設定

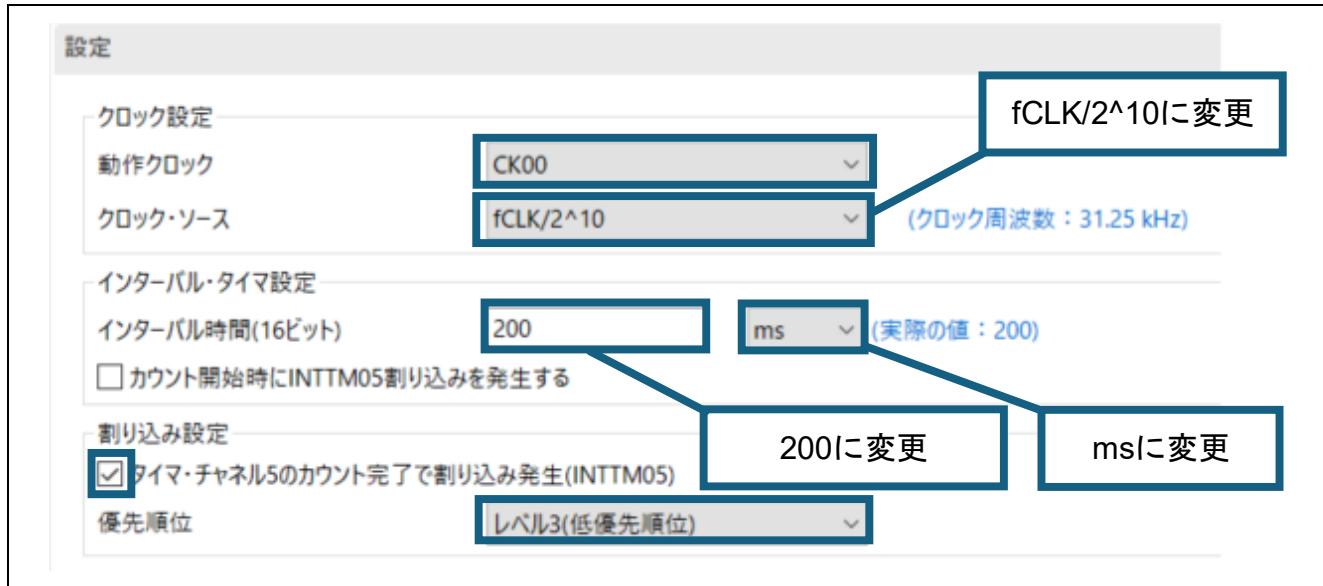


図 5-11 に LCD コンポーネント (Config_LCD) の設定を示します。

図 5-11 LCD コンポーネント (Config_LCD) の設定

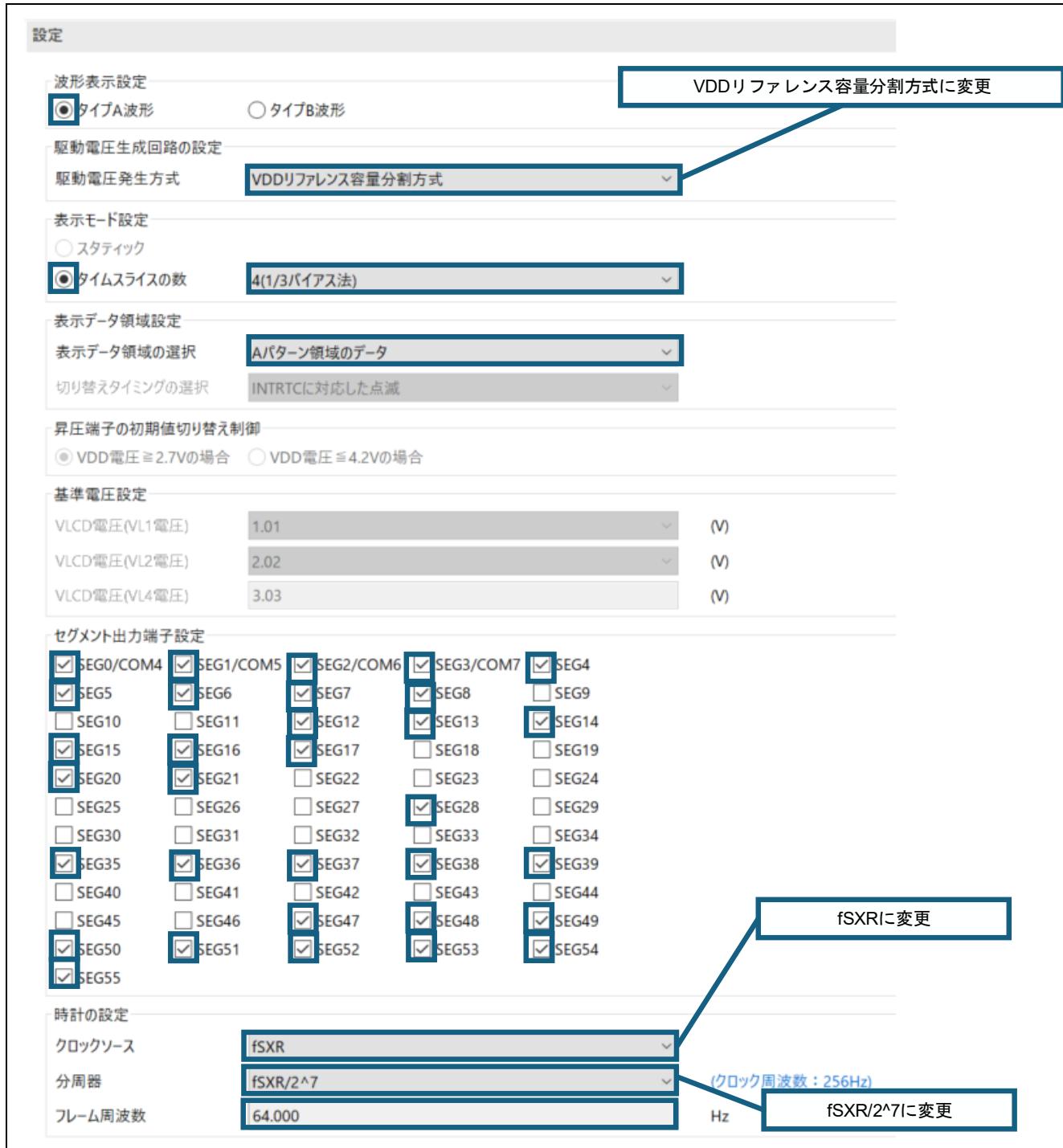


図 5-12、図 5-13 に PORT コンポーネント (Config_PORT) の設定を示します。

図 5-12 PORT コンポーネント (Config_PORT) の設定



図 5-13 PORT6 の設定

ポート	出力	内蔵プルアップ	TTLバッファ	入力バッファオフ	N-ch	1出力	ELCL出力信号を出力する	出力電流	Hi-Z
P60	<input type="radio"/> 使用しない <input checked="" type="radio"/> 出力 <input type="radio"/> 入力 <input type="radio"/> 出力電流	<input type="checkbox"/> 1出力	<input type="checkbox"/> ELCL出力信号を出力する	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z				
P61	<input type="radio"/> 使用しない <input type="radio"/> 入力 <input checked="" type="radio"/> 出力 <input type="radio"/> 出力電流	<input type="checkbox"/> 1出力	<input type="checkbox"/> ELCL出力信号を出力する	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z				
P62	<input type="radio"/> 使用しない <input type="radio"/> 入力 <input checked="" type="radio"/> 出力 <input type="radio"/> 出力電流	<input type="checkbox"/> 1出力	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z					
P63	<input type="radio"/> 使用しない <input type="radio"/> 入力 <input checked="" type="radio"/> 出力 <input type="radio"/> 出力電流	<input type="checkbox"/> 1出力	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z					
P64	<input type="radio"/> 使用しない <input type="radio"/> 入力 <input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> 入力バッファオフ	<input type="checkbox"/> N-ch	<input checked="" type="checkbox"/> 1出力	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z	
P65	<input type="radio"/> 使用しない <input type="radio"/> 入力 <input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> 入力バッファオフ	<input type="checkbox"/> N-ch	<input checked="" type="checkbox"/> 1出力	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z	
P66	<input type="radio"/> 使用しない <input type="radio"/> 入力 <input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> 入力バッファオフ	<input type="checkbox"/> N-ch	<input type="checkbox"/> 1出力	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z	
P67	<input type="radio"/> 使用しない <input type="radio"/> 入力 <input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> 1出力	<input type="checkbox"/> 出力電流	<input type="checkbox"/> Hi-Z				

【備考】 本資料では、LED 用ポートとして P64 および P65 のみを設定しています。それ以外のポートは、浮いた状態による不要な電流消費を防止しています。詳しくは、「RL78/L23 ユーザーズマニュアル ハードウェア編 (R01UH1082)」の「2.3 未使用端子の処理」を参照し、本アプリケーションで使用していない端子の端子処理を適切に行い、電気的特性を満たすように設計して下さい。

図 5-14 に CTSU ドライバ (r_ctsu) の設定を示します。

図 5-14 CTSU ドライバ (r_ctsu) の設定

プロパティ	値
▼ Configurations	
# Parameter check	Use system default
# Data transfer of INTCTSUWR and INTCTSURD	Interrupt handler
# DTC setting	Setting in r_ctsu
# Select auto judgement	Disable
# Data storage address setting for CTSUWR	0xFF300
# Variable address setting for g_ctsu_self_raw.	0xFF400
# Variable address setting for g_ctsu_mutual_raw	0xFF500
# Data storage address setting for CTSUAJTHR.	0xFF600
# Data storage address setting for CTSUAJMMAR.	0xFF700
# Data storage address setting for CTSUAJBLACT.	0xFF800
# Data storage address setting for CTSUAJBLAR.	0xFF900
# Data storage address setting for CTSUAJRR.	0xFFA00
# Data storage address setting for CTSUMCACT1.	0xFFB00
# Data storage address setting for CTSUMCACT2.	0xFFC00
# Auto-judgment function in Snooze mode using SMS	Disable
# Data storage address setting for CTSURD	0xFF500
# Data storage address setting for CTSUWR	0xFF800
# Interrupt level for INTCTSUWR	Level 2
# Interrupt level for INTCTSURD	Level 2
# Interrupt level for INTCTSUFN	Level 2
# Output port number for external trigger	POR14
# Bit number for external trigger output	BIT0
# Interrupt port number for external trigger	INTP1
▼ リソース	
CTSU	
TSCAP端子	<input checked="" type="checkbox"/> 使用する
TS00端子	<input type="checkbox"/> 使用する
TS01端子	<input type="checkbox"/> 使用する
TS02端子	<input type="checkbox"/> 使用する
TS03端子	<input type="checkbox"/> 使用する
TS04端子	<input type="checkbox"/> 使用する
TS05端子	<input type="checkbox"/> 使用する
TS06端子	<input type="checkbox"/> 使用する
TS07端子	<input type="checkbox"/> 使用する
TS08端子	<input type="checkbox"/> 使用する
TS09端子	<input type="checkbox"/> 使用する
TS10端子	<input type="checkbox"/> 使用する
TS11端子	<input type="checkbox"/> 使用する
TS12端子	<input type="checkbox"/> 使用する
TS13端子	<input type="checkbox"/> 使用する
TS14端子	<input type="checkbox"/> 使用する
TS15端子	<input type="checkbox"/> 使用する
TS16端子	<input type="checkbox"/> 使用する
TS17端子	<input checked="" type="checkbox"/> 使用する
TS18端子	<input checked="" type="checkbox"/> 使用する
TS19端子	<input type="checkbox"/> 使用する
TS20端子	<input type="checkbox"/> 使用する
TS21端子	<input type="checkbox"/> 使用する
TS22端子	<input type="checkbox"/> 使用する
TS23端子	<input type="checkbox"/> 使用する
TS24端子	<input type="checkbox"/> 使用する
TS25端子	<input type="checkbox"/> 使用する
TS26端子	<input type="checkbox"/> 使用する
TS27端子	<input type="checkbox"/> 使用する
TS28端子	<input type="checkbox"/> 使用する
TS29端子	<input type="checkbox"/> 使用する
TS30端子	<input type="checkbox"/> 使用する
TS31端子	<input type="checkbox"/> 使用する
TS32端子	<input type="checkbox"/> 使用する
TS33端子	<input type="checkbox"/> 使用する
TS34端子	<input type="checkbox"/> 使用する
TS35端子	<input type="checkbox"/> 使用する

図 5-15 に TOUCH ドライバ (rm_touch) の設定を示します。

図 5-15 TOUCH ドライバ (rm_touch) の設定

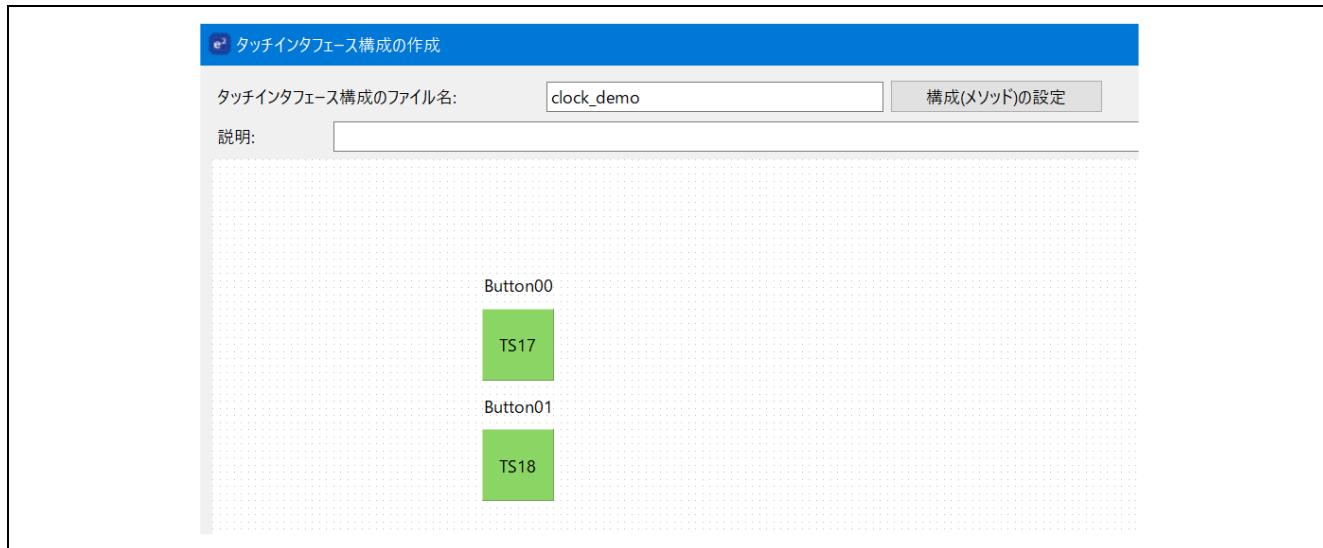
設定	
フィルタ入力 (* = any string, ? = any character)	
プロパティ	値
▼  Configurations	
# Parameter check	Use system default
# Support QE monitor using UART	Disable
# Support QE tuning using UART	Disable
# UART channel	UART0
# Type of chattering suppression	TypeA : Counter of exceed threshold is hold within hysteresis range.

5.4 静電容量タッチ設定

5.4.1 タッチインターフェース構成

図 5-16 にタッチインターフェース構成を示します。TS17、TS18 を自己容量方式で計測します。

図 5-16 タッチインターフェース構成



5.4.2 構成 (メソッド) の設定

図 5-17 にタッチインターフェースの構成 (メソッド) の設定を示します。

図 5-17 メソッドの設定



5.4.3 チューニング結果

図 5-18 にタッチインターフェースの QE チューニングでの調整結果を示します。本サンプルコードは以下に示される設定値で動作します。

調整結果は QE チューニング時の動作環境に依存するため、再度 QE チューニングを行うとこれらの値が変化する可能性があります。

図 5-18 QE チューニング結果

チューニング ジェスチャ									
タッチインターフェース構成: clock_demo									
メソッド	種別	名前	タッチセンサ	寄生容量[pF]	センサドライバパルス周波数[MHz]	しきい値	計測時間[ms]	オーバーフロー	リセット
config01	ボタン(自己)	Button00	TS17	7.5	5.436	3990	0.576	なし	
config01	ボタン(自己)	Button01	TS18	7.312	5.547	3967	0.576	なし	

5.5 定数一覧

表 5-3 にサンプルコードで使用する定数を示します。

表 5-3 サンプルコードで使用する定数

定数名	設定値	内容
MAX_SEC_MIN	59	時計の分、秒の最大値
MAX_HOUR	23	時計の時の最大値
MIN_SEC_MIN_HOUR	0	時計の時、分、秒の最小値(0)
UNIT_SEC	0	時計の秒の単位
UNIT_MIN	1	時計の分の単位
UNIT_HOUR	2	時計の時の単位
TIME_SETTING_FINISH	5	時刻設定の終了時間(秒)
COUNTINUOUS_START_NUM	3	連続タッチ検出回数

5.6 変数一覧

表 5-4、表 5-5 にサンプルコードで使用する変数を示します。

表 5-4 サンプルコードで使用する変数（1/2）

Type	Variable Name	Contents	Function Used
uint8_t	g_RTC_interrupt_flag	RTC 割り込みフラグ	r_RTC_callback r_change_time
uint8_t	g_show_segdata[4][4]	LCD に表示される文字を格納する配列	r_lcd_show
uint8_t	g_digit_segdata[8][4]	LCD の表示桁数を格納する配列	r_lcd_show
uint8_t	g_touch_button_flag	タッチ機能を有効にするフラグ	qe_touch_main r_RTC_callback, r_time_unit_change
uint8_t	s_lcd_seconds	時計の秒の値を格納する変数	r_RTC_callback r_time_show r_change_time
uint8_t	s_lcd_minutes	時計の分の値を格納する変数	r_RTC_callback r_time_show r_change_time
uint8_t	s_lcd_hours	時計の時の値を格納する変数	r_RTC_callback r_time_show r_change_time
uint8_t	sp_time_address[3]	時計の秒、分、時の変数を格納する配列	r_touch_plus_set_time r_touch_minus_set_time
uint8_t	s_unit	時計の単位を格納する変数	r_touch_plus_set_time r_touch_minus_set_time r_blink_set_time r_time_unit_change
uint8_t	s_clock_mode	時計の状態を格納するフラグ	r_RTC_callback r_time_unit_change
uint8_t	s_blink_flag	LCD の選択した単位を点滅するためのフラグ	r_tau0_1_blink_callback
uint8_t	s_set_time_limit_count	時刻設定状態を終了する時間をカウントする変数	r_userswitch_callback r_RTC_callback r_touch_plus_set_time r_touch_minus_set_time r_touch_both_buttons
uint8_t	s_clock_start_flag	LCD の時刻表示開始を示すフラグ	r_clock_demo_start r_userswitch_release r_userswitch_callback
uint8_t	s_100ms_count	100ms 単位で待機時間をカウントする変数	r_tau0_3_delay_callback r_delay_100ms
uint8_t	s_chattering_flag	ユーザスイッチを長押しした時間をカウントするフラグ	r_userswitch_callback r_tau0_4_chattering_callback
uint8_t	s_prev_button1_flag	タッチボタン 1 の以前の状態を格納するフラグ	r_touch_plus_set_time r_touch_both_buttons r_no_touch

表 5-5 サンプルコードで使用する変数（2/2）

Type	Variable Name	Contents	Function Used
uint8_t	s_prev_button2_flag	タッチボタン 2 の以前の状態を格納するフラグ	r_touch_minus_set_time r_touch_both_buttons r_no_touch
uint8_t	s_continuous_touch_flag	タッチボタンの連続タッチの間隔を判定するフラグ	r_tau0_5_continuous_callback r_touch_plus_set_time r_touch_minus_set_time r_touch_both_buttons r_no_touch
uint8_t	s_continuous_touch_count	タッチボタンの連続タッチの回数をカウントする変数	r_tau0_5_continuous_callback r_touch_plus_set_time r_touch_minus_set_time
uint8_t	s_userswitch_on_of_flag	ユーザスイッチの ON/OFF 状態を判定するフラグ	r_userswitch_release r_userswitch_callback

5.7 関数一覧

表 5-6 にサンプルコード（clock.c）で使用する関数を示します。

表 5-6 サンプルコード（clock.c）で使用する関数

関数名	概要
void r_clock_demo_start(void);	時計デモの開始処理
void r_userswitch_release(void);	ユーザスイッチ検出処理
void r_change_to_sub(void);	サブシステム・クロック動作遷移
void r_change_to_main(void);	メイン・システム・クロック動作遷移
void r_userswitch_callback(void);	ユーザスイッチ押下時割り込みコールバック
void r_rtc_callback(void);	RTC 割り込みコールバック
void r_tau0_1_blink_callback(void);	TAU0_1 割り込みコールバック
void r_tau0_3_delay_callback(void);	TAU0_3 割り込みコールバック
void r_tau0_4_chattering_callback(void);	TAU0_4 割り込みコールバック
void r_tau0_5_continuous_callback(void);	TAU0_5 割り込みコールバック
void r_touch_plus_set_time(void);	タッチボタン1タッチ検出処理
void r_touch_minus_set_time(void);	タッチボタン2タッチ検出処理
void r_touch_both_buttons(void);	タッチボタン1と2の同時タッチ検出処理
void r_no_touch(void);	タッチ解除検出処理
static void r_lcd_show(uint8_t value, uint8_t digit);	LCD に値表示処理
static void r_time_show(void);	LCD への時刻表示処理
static void r_change_time(void);	RTC レジスタから時刻取得処理
static void r_blink_set_time(void);	時刻設定中の選択単位点滅処理
static void r_array_show(uint8_t* array, uint8_t num, uint8_t delay);	配列データを LCD に順次表示処理
static void r_clear_show(void);	LCD 表示クリア処理
static void delay_100ms(uint8_t num);	遅延発生処理
static void r_time_unit_change(void);	時刻設定箇所変更処理

5.8 関数仕様

サンプルコード（clock.c）の関数仕様を示します。

r_clock_demo_start

概要	時計デモの開始処理
ヘッダ	clock.h
宣言	void r_clock_demo_start(void)
説明	デモ開始時に LCD に「RL78/L23」、「CLK DEMO」を表示
引数	なし
リターン値	なし
備考	なし

r_userswitch_release

概要	ユーザスイッチ検出処理
ヘッダ	clock.h
宣言	void r_userswitch_release(void)
説明	ユーザスイッチのリリースに応じて対応するフラグを変更
引数	なし
リターン値	なし
備考	なし

r_change_to_sub

概要	サブシステム・クロック動作遷移
ヘッダ	clock.h
宣言	void r_change_to_sub(void)
説明	現在のクロックをメイン・システム・クロックからサブシステム・クロックへ切り替える
引数	なし
リターン値	なし
備考	なし

r_change_to_main

概要	メイン・システム・クロック動作遷移
ヘッダ	clock.h
宣言	void r_change_to_main (void)
説明	現在のクロックをサブシステム・クロックからメイン・システム・クロックへ切り替える
引数	なし
リターン値	なし
備考	なし

r_userswitch_callback

概要	ユーザスイッチ押下時割り込みコールバック
ヘッダ	clock.h
宣言	void r_userswitch_callback(void)
説明	短押しの場合は時刻設定を開始し、r_time_unit_change をコールアップする
引数	なし
リターン値	なし
備考	なし

r_RTC_callback

概要	RTC 割り込みコールバック
ヘッダ	clock.h
宣言	void r_RTC_callback(void)
説明	割り込み発生毎に時刻をカウントアップする ユーザスイッチを押して時刻設定を行う場合は時計に戻るカウントを行う
引数	なし
リターン値	なし
備考	なし

r_tau0_1_blink_callback

概要	TAU0_1 割り込みコールバック
ヘッダ	clock.h
宣言	void r_tau0_1_blink_callback(void)
説明	時刻設定を行う箇所の LCD 点滅を行う
引数	なし
リターン値	なし
備考	なし

r_tau0_3_delay_callback

概要	TAU0_3 割り込みコールバック
ヘッダ	clock.h
宣言	void r_tau0_3_delay_callback(void)
説明	タイマ (TAU0 チャネル 3) によってディレイのカウント変数をインクリメントする
引数	なし
リターン値	なし
備考	なし

r_tau0_4_chattering_callback

概要	TAU0_4 割り込みコールバック
ヘッダ	clock.h
宣言	void r_tau0_4_chattering_callback(void)
説明	チャタリング対策に外部割り込みを一定時間無効にする
引数	なし
リターン値	なし
備考	なし

r_tau0_5_continuous_callback

概要	TAU0_5 割り込みコールバック
ヘッダ	clock.h
宣言	void r_tau0_5_continuous_callback(void)
説明	タッチボタンの連続タッチを判定する
引数	なし
リターン値	なし
備考	なし

r_touch_plus_set_time

概要	タッチボタン1タッチ検出処理
ヘッダ	clock.h
宣言	void r_touch_plus_set_time(void)
説明	タッチボタン1をタッチ検出した際に、時刻設定中の対象単位のLCD表示を+1する
引数	なし
リターン値	なし
備考	なし

r_touch_minus_set_time

概要	タッチボタン2タッチ検出処理
ヘッダ	clock.h
宣言	void r_touch_minus_set_time(void)
説明	タッチボタン2をタッチ検出した際に、時刻設定中の対象単位のLCD表示を-1する
引数	なし
リターン値	なし
備考	なし

r_touch_both_buttons

概要	タッチボタン1と2の同時タッチ検出処理
ヘッダ	clock.h
宣言	void r_touch_both_buttons(void)
説明	タッチボタン1、2が同時にタッチ検出された際にタイマの停止や各種制御変数のリセットを行う
引数	なし
リターン値	なし
備考	なし

r_no_touch

概要	タッチ解除検出処理
ヘッダ	clock.h
宣言	void r_no_touch(void)
説明	タッチボタンがいずれもタッチ検出されていない場合、内部状態を初期化し、タイマなどの制御を停止する
引数	なし
リターン値	なし
備考	なし

r_lcd_show

概要	LCDに値表示処理
ヘッダ	なし
宣言	static void r_lcd_show(uint8_t value, Uint8_t digit)
説明	LCDに表示する際に、グローバル変数に格納している value(数字・文字の番号)と digit(LCDの桁)を設定し、表示する
引数	uint8_t value uint8_t digit
	表示する数や文字 表示する桁
リターン値	なし
備考	なし

r_time_show

概要	LCD への時刻表示処理
ヘッダ	なし
宣言	static void r_time_show(void)
説明	グローバル関数 s_lcd_hours、s_lcd_minutes、s_lcd_seconds に格納された値を LCD に表示する
引数	なし
リターン値	なし
備考	なし

r_change_time

概要	RTC レジスタから時刻取得
ヘッダ	なし
宣言	static void r_change_time(void)
説明	グローバル変数の値を RTC 割り込みのたびに、1 秒+する
引数	なし
リターン値	なし
備考	なし

r_blink_set_time

概要	時刻設定中の選択単位点滅処理
ヘッダ	なし
宣言	static void r_blink_set_time(void)
説明	時刻設定を行う際に、選択されている箇所の点滅を行う
引数	なし
リターン値	なし
備考	なし

r_array_show

概要	配列データを LCD に順次表示処理	
ヘッダ	なし	
宣言	static void r_array_show(uint8_t * array, uint8_t num, uint8_t delay)	
説明	配列に格納した番号に対応する数字・文字を LCD に表示する	
引数	uint8_t * array uint8_t num uint8_t delay	表示する値のポインタ 表示する要素数 各表示後に待機する時間 (100ms 単位)
リターン値	なし	
備考	なし	

r_clear_show

概要	LCD 表示クリア処理
ヘッダ	なし
宣言	static void r_clear_show(void)
説明	LCD の 8 行すべてに空白を表示させ、表示内容を全てクリア状態にする
引数	なし
リターン値	なし
備考	なし

r_delay_100ms

概要	遅延発生処理
ヘッダ	なし
宣言	static void r_delay_100ms(uint8_t num)
説明	100ms 単位の遅延を発生させる関数
引数	uint8_t num
	待機する 100ms 単位の時間 num=3 の場合、300ms 待機
リターン値	なし
備考	なし

r_time_unit_change

概要	時刻設定箇所変更処理
ヘッダ	なし
宣言	static void r_time_unit_change(void)
説明	時刻設定を行う箇所を秒→分→時→秒の変更を行う
引数	なし
リターン値	なし
備考	なし

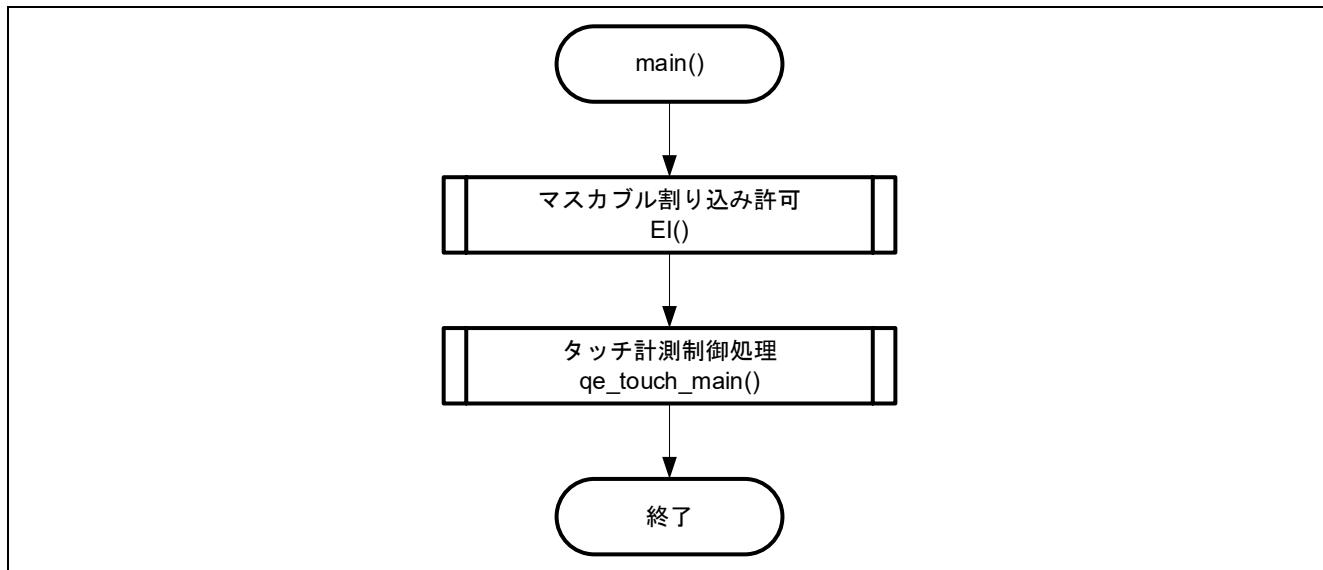
5.9 フローチャート

本サンプルコードの流れを示す主要関数のフローチャートを記載します。

5.9.1 main 関数

図 5-19 に main 関数のフローを示します。

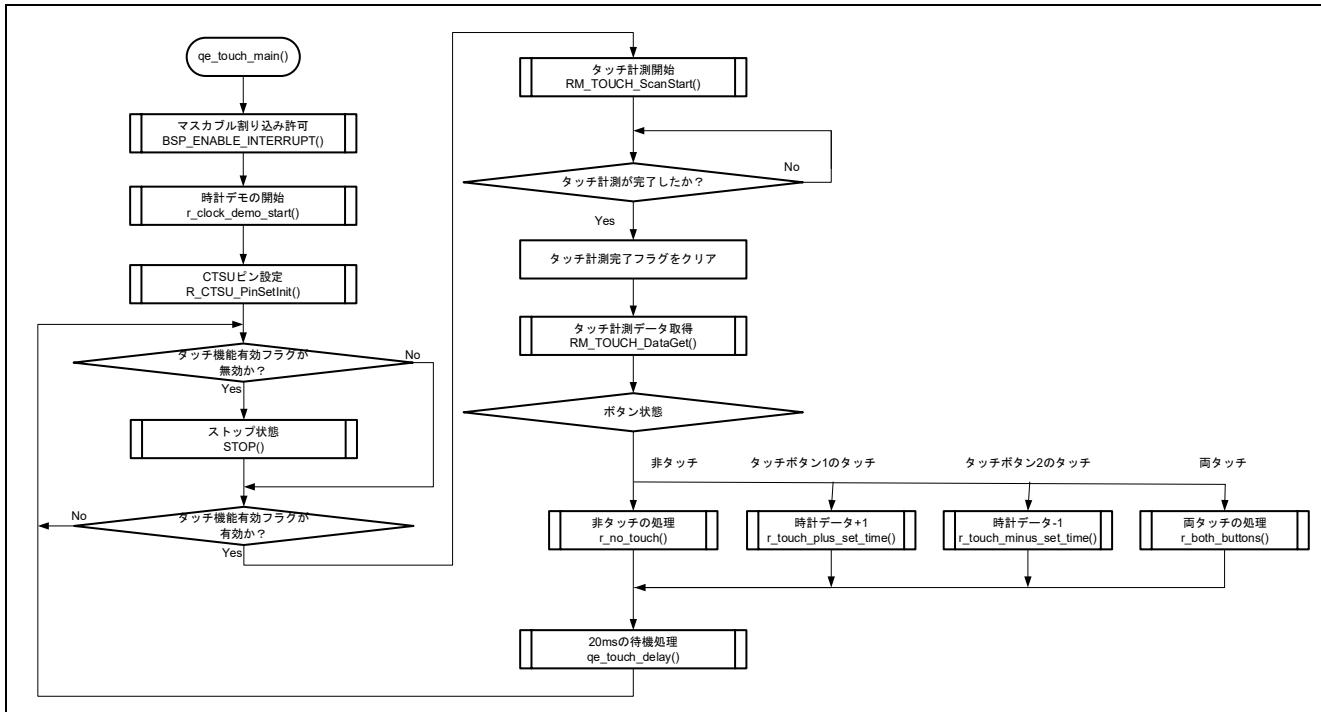
図 5-19 main 関数フローチャート



5.9.2 qe_touch_main 関数

図 5-20 に qe_touch_main 関数のフローチャートを示します。

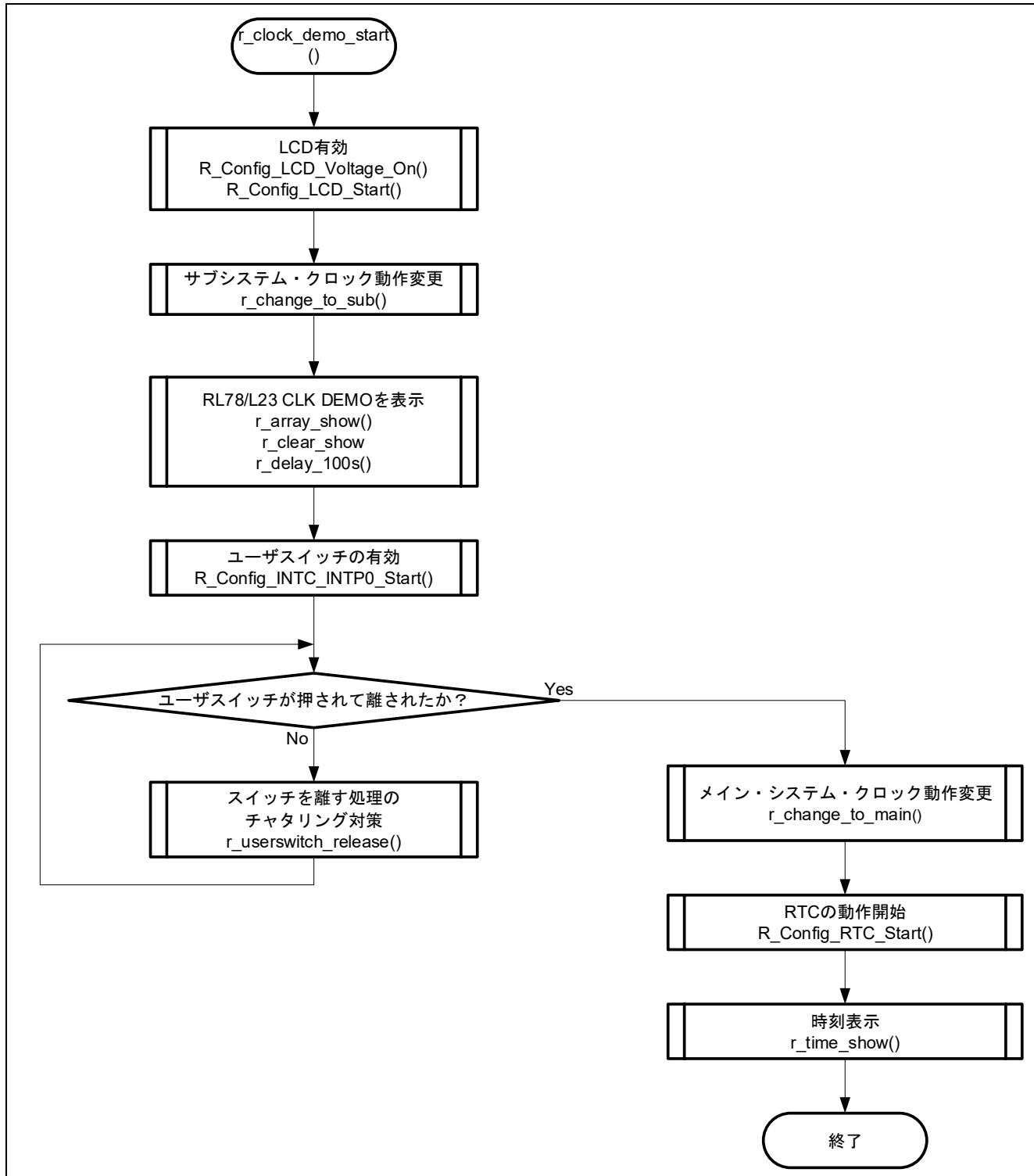
図 5-20 qe_touch_main 関数のフローチャート



5.9.3 r_clock_demo_start 関数

図 5-21 に r_clock_demo_start 関数のフローチャートを示します。

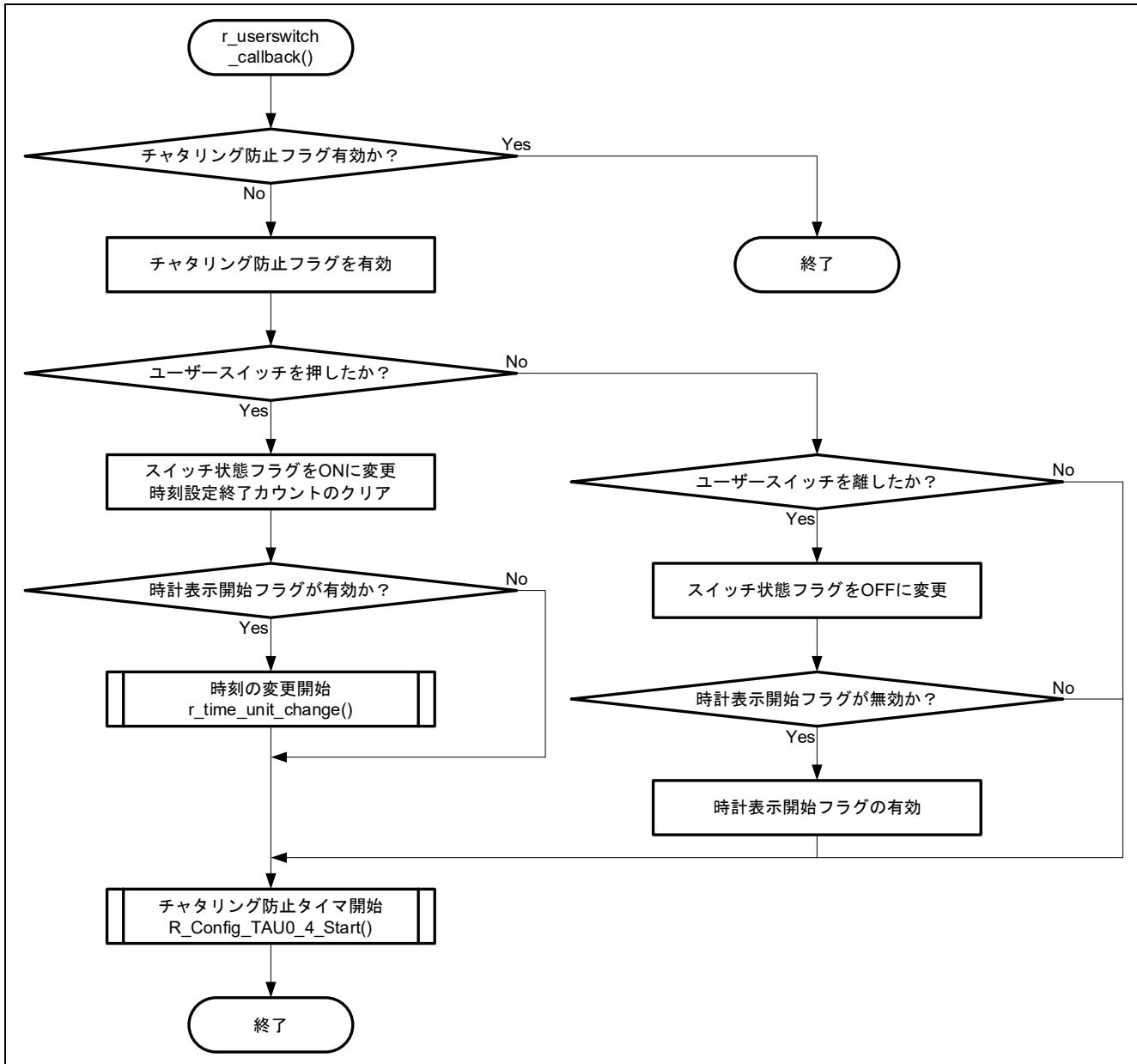
図 5-21 r_clock_demo_start 関数のフローチャート



5.9.4 r_userswitch_callback 関数

図 5-22 に r_userswitch_callback 関数のフローチャートを示します。

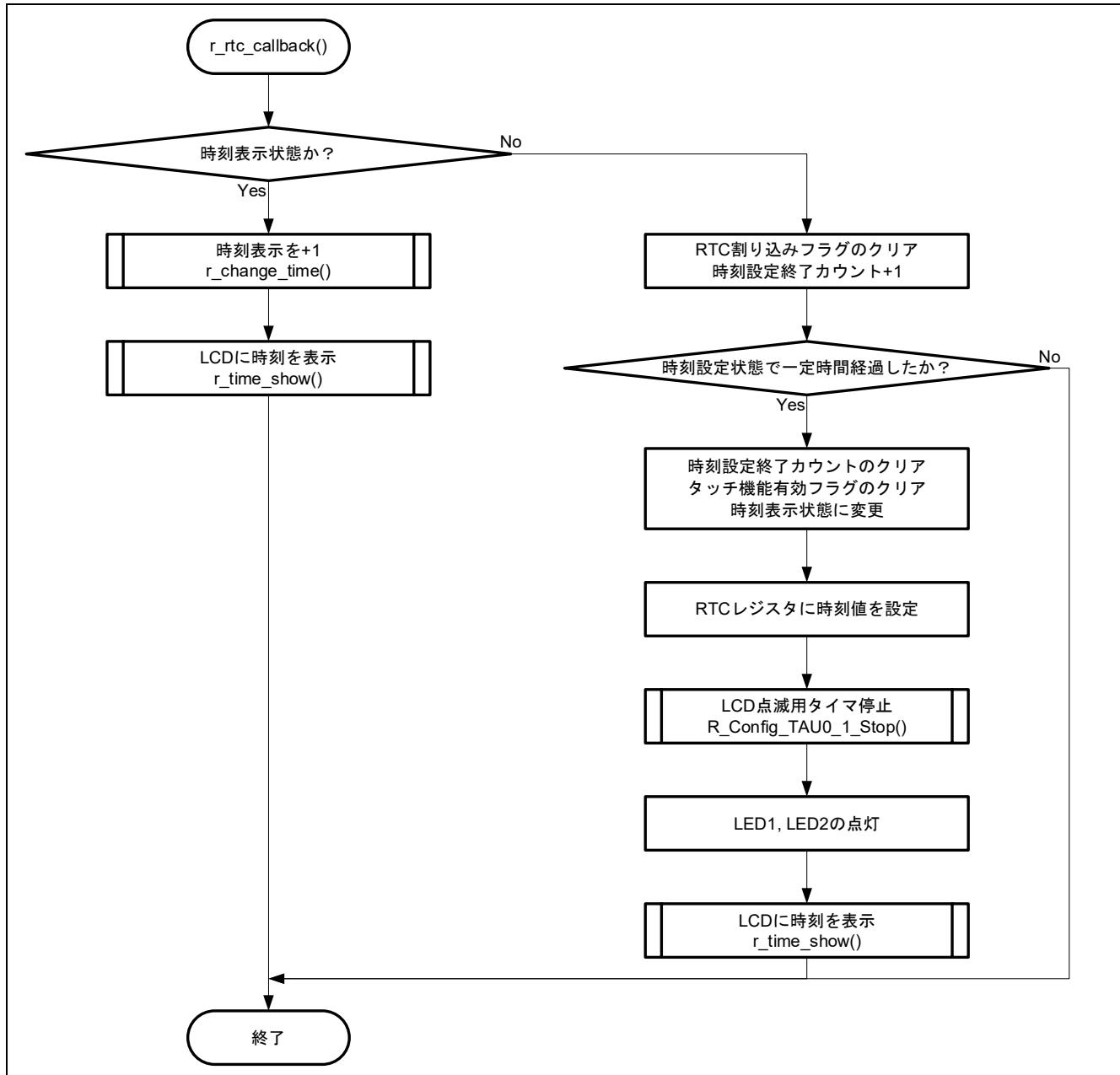
図 5-22 r_userswitch_callback 関数のフローチャート



5.9.5 r_RTC_callback 関数

図 5-23 に r_RTC_callback 関数のフローチャートを示します。

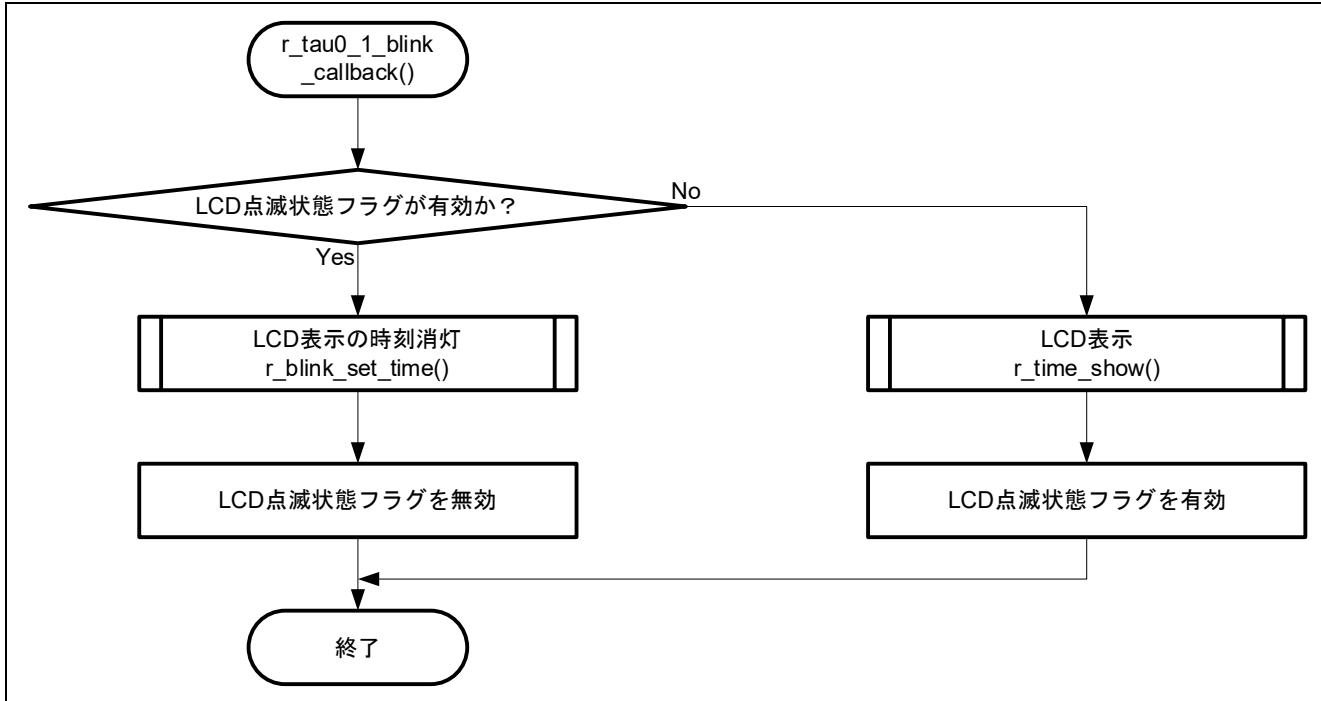
図 5-23 r_RTC_callback 関数のフローチャート



5.9.6 r_tau0_1_blink_callback 関数

図 5-24 に r_tau0_1_blink_callback 関数のフローチャートを示します。

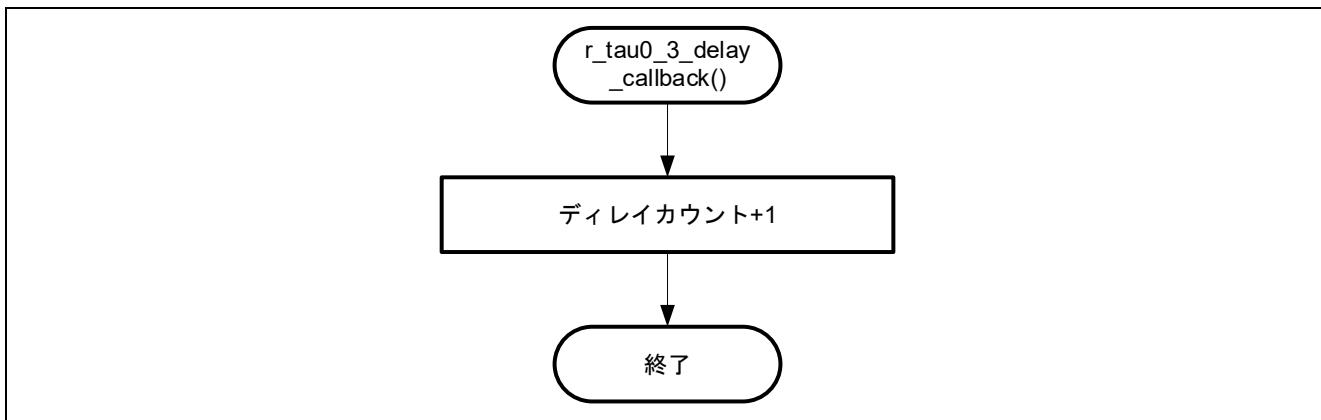
図 5-24 r_tau0_1_blink_callback 関数のフローチャート



5.9.7 r_tau0_3_delay_callback 関数

図 5-25 に r_tau0_3_delay_callback 関数のフローチャートを示します。

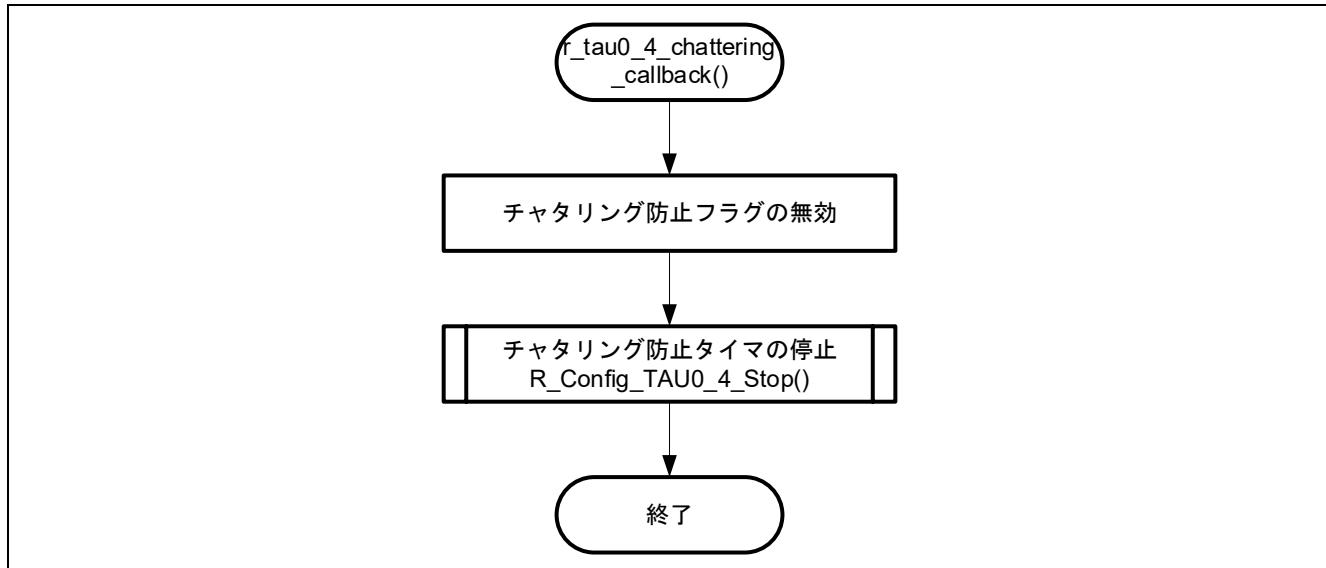
図 5-25 r_tau0_3_delay_callback 関数のフローチャート



5.9.8 r_tau0_4_chattering_callback 関数

図 5-26 に r_tau0_4_chattering_callback 関数のフローチャートを示します。

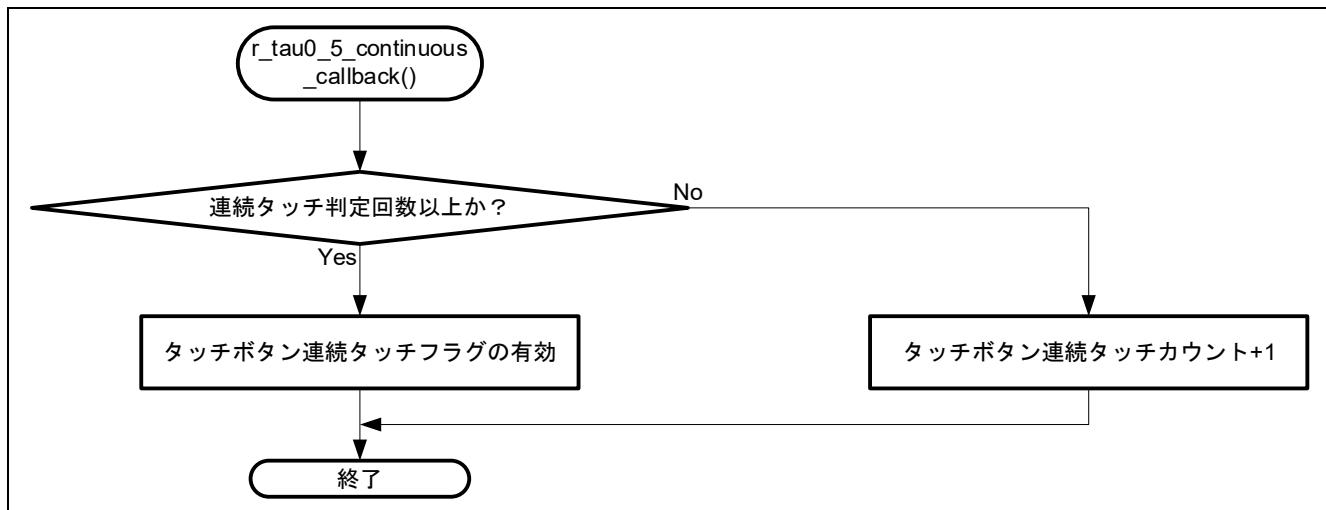
図 5-26 r_tau0_4_chattering_callback 関数のフローチャート



5.9.9 r_tau0_5_continuous_callback 関数

図 5-27 に r_tau0_5_continuous_callback 関数のフローチャートを示します。

図 5-27 r_tau0_5_continuous_callback 関数のフローチャート



6. プロジェクトのインポート方法

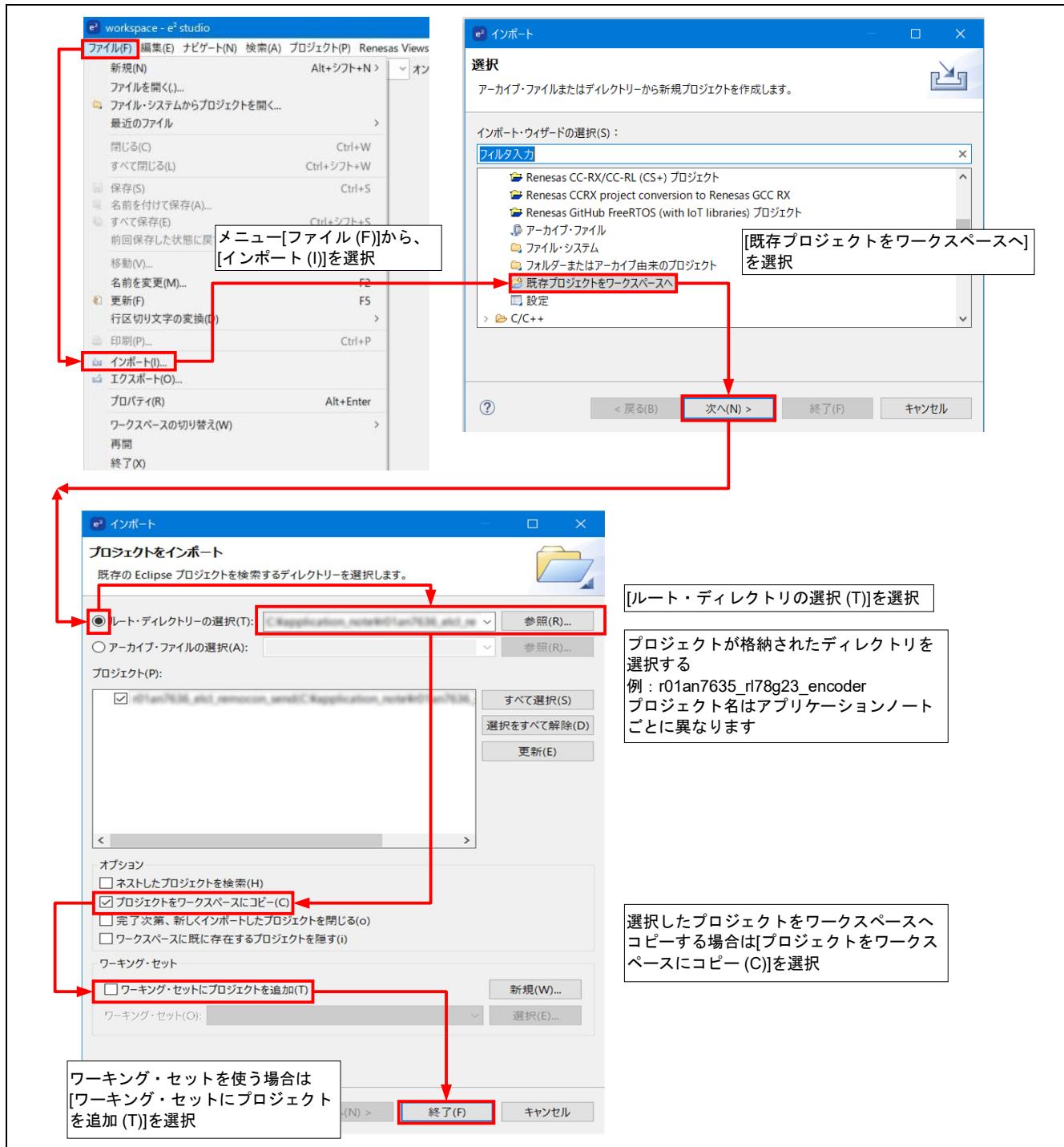
6.1 e² studio での手順

e² studio でご使用になる際は、以下の手順で e² studio にインポートしてください。

なお、e² studio で管理するプロジェクトのフォルダ名、およびそのフォルダに至るファイルパスには、空白文字の他、半角カナ文字、全角文字、半角記号（特に'\$'、'#'、 '%'）が混じらないようにしてください。

（使用する e² studio のバージョンによっては画面が異なる場合があります。）

図 6-1 プロジェクトを e² studio にインポートする方法

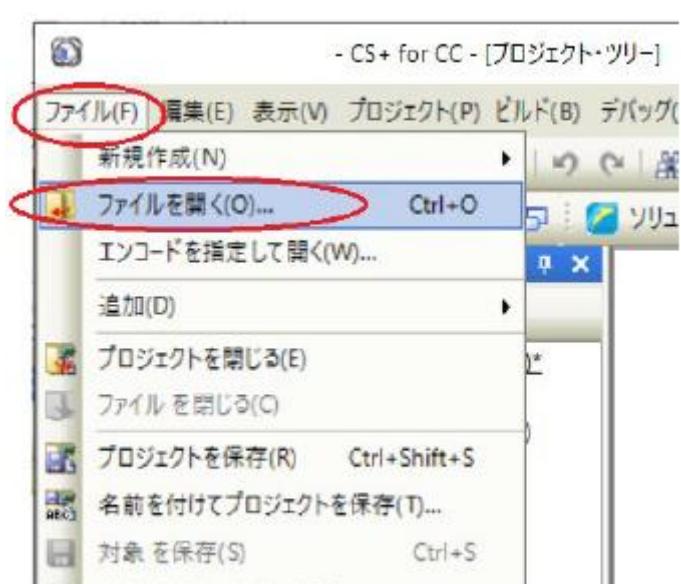


6.2 CS+での手順

CS+でご使用になる際は、「ファイル」 - 「ファイルを開く」メニューを選択し、プロジェクトの mtpj ファイルを読み込みます。

（使用する CS+ のバージョンによっては画面が異なる場合があります。）

図 6-2 プロジェクトを CS+ にインポートする方法

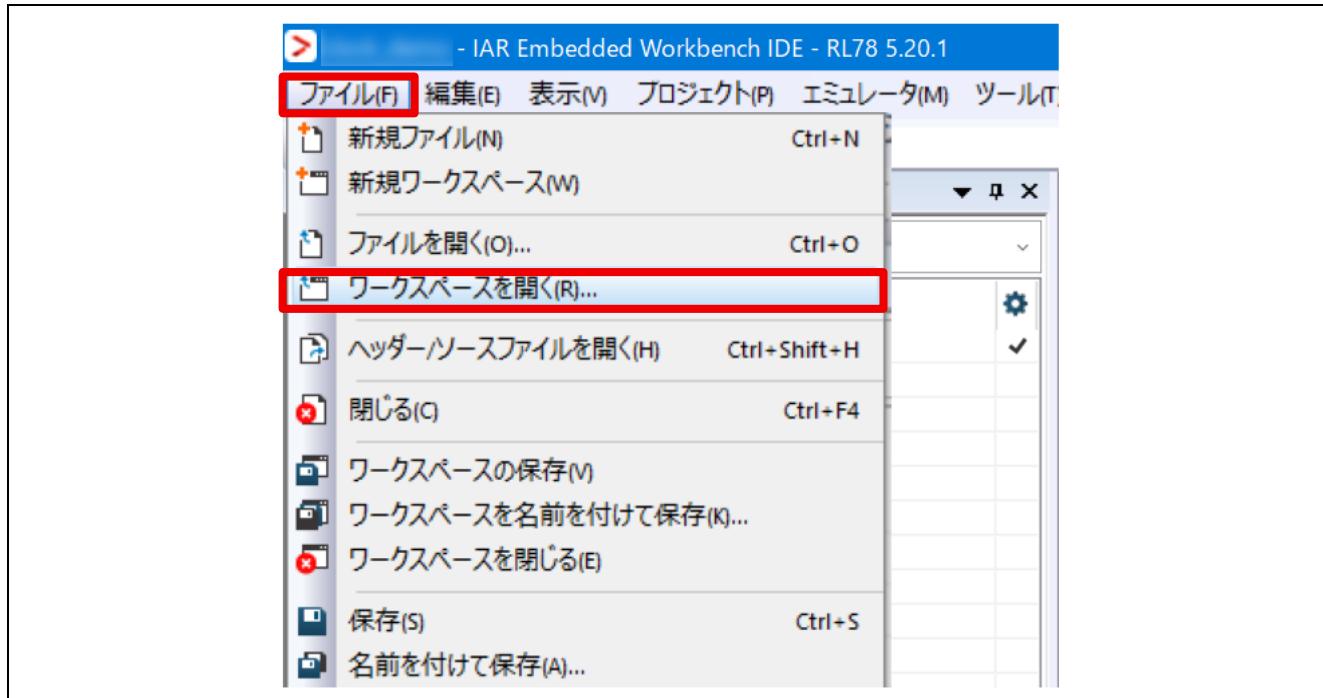


6.3 IAR での手順

IAR でご使用になる際は、「ファイル」 – 「ワークスペースを開く」メニューを選択し、プロジェクトの eww ファイルを読み込みます。

（使用する IAR のバージョンによっては画面が異なる場合があります。）

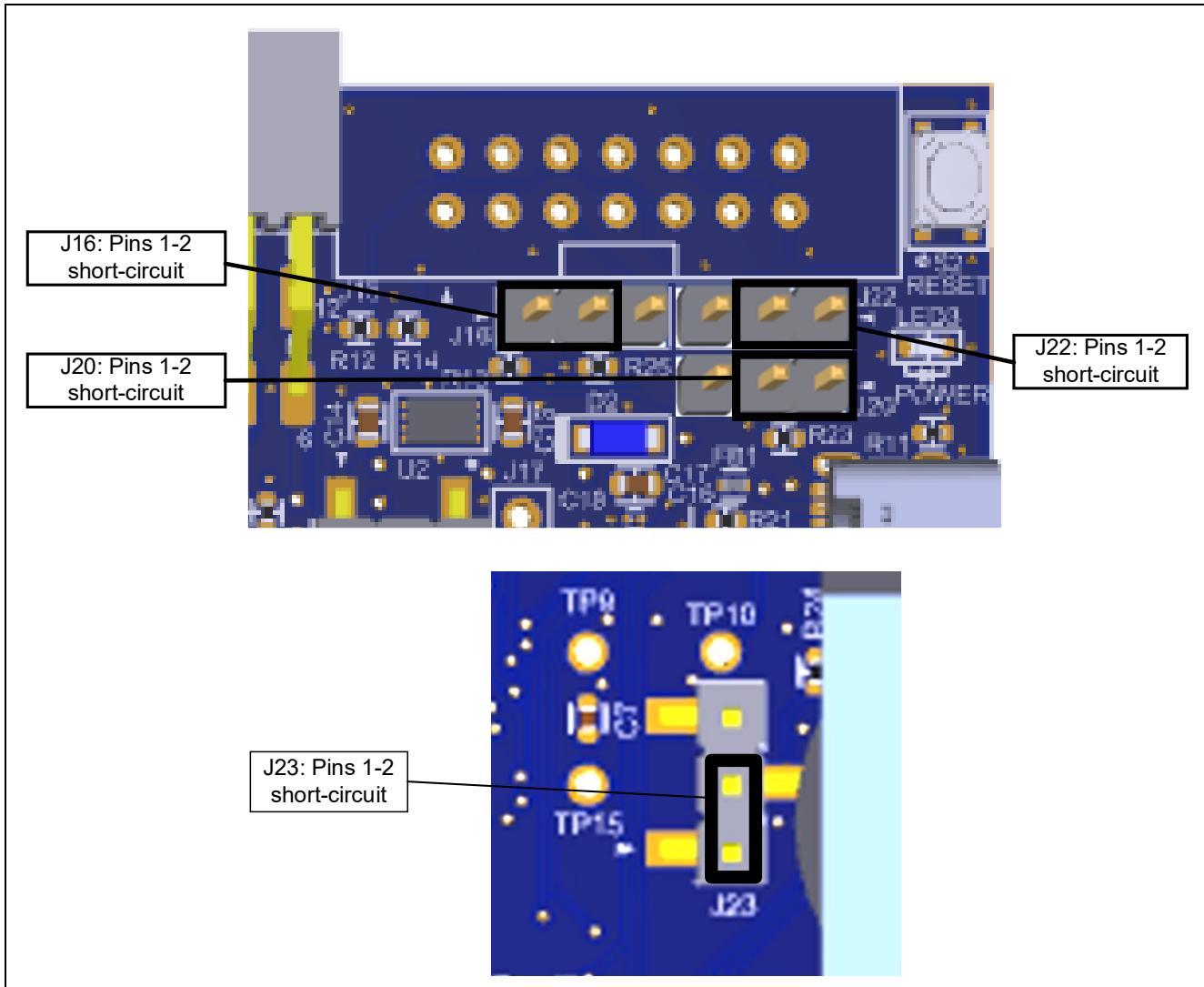
図 6-3 プロジェクトを IAR にインポートする方法



7. デバッグの設定

本サンプルコードでは、RL78/L23 Fast Prototyping Board (RTK7RLL230S000001BJ) の USB-C ポートを使用し、COM Port 経由でデバッグを行います。COM Port でのデバッグを行う際は、RL78/L23 のジャンパ設定が図 7-1 のとおりになっていることをご確認ください。詳細については、「RL78/L23 Fast Prototyping Board ユーザーズマニュアル (R20UT5544)」の「5.15 USB-シリアル変換器」と「5.21 エミュレータ用コネクタ」をご参照ください。

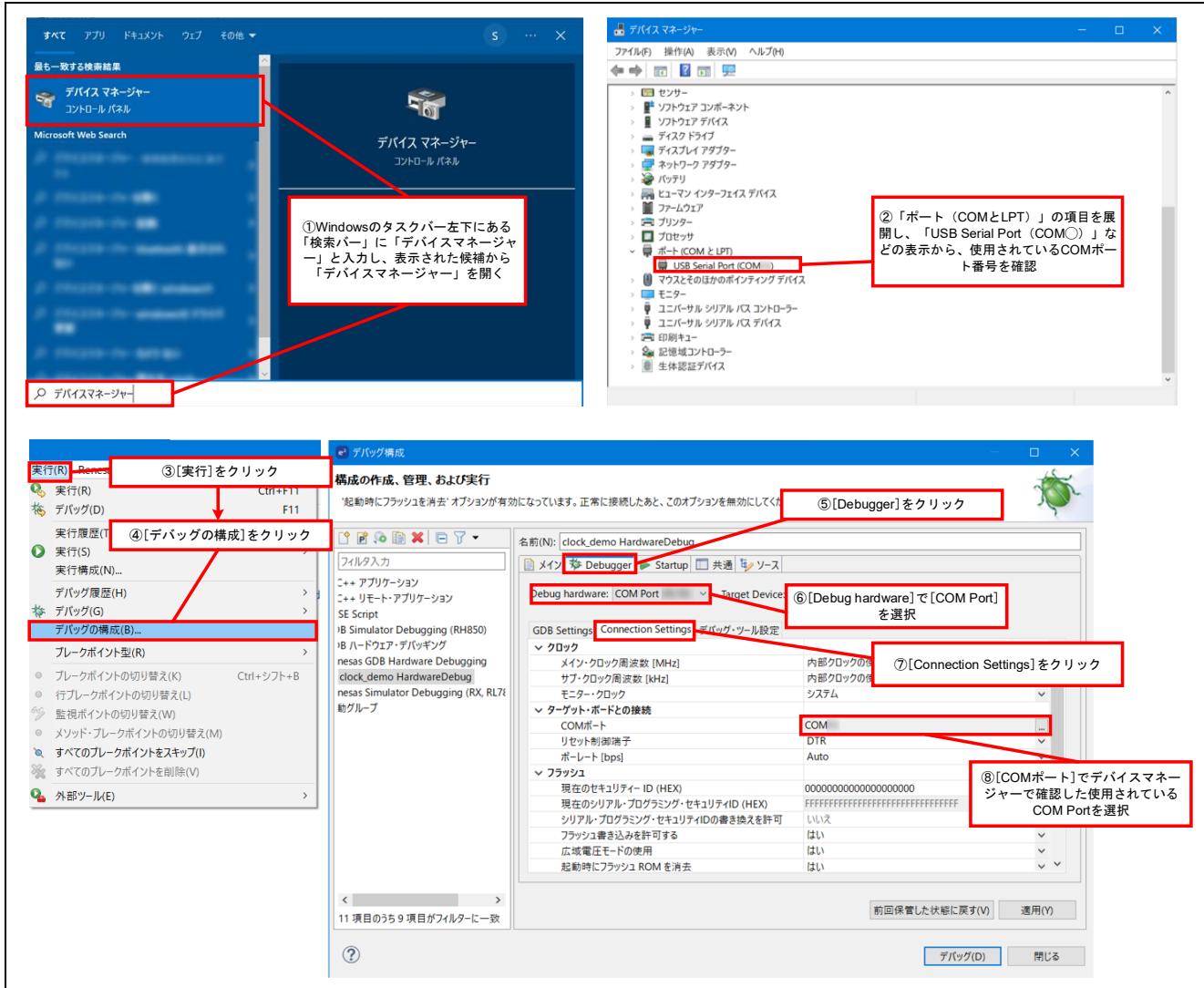
図 7-1 COM Port デバッグ使用時設定



7.1 e² studio での COM Port 設定

パソコンと RL78/L23 ボードを USB-C ケーブルで接続した後、図 7-2 を参考にしてデバッグ時に使用する COM Port を選択してください。

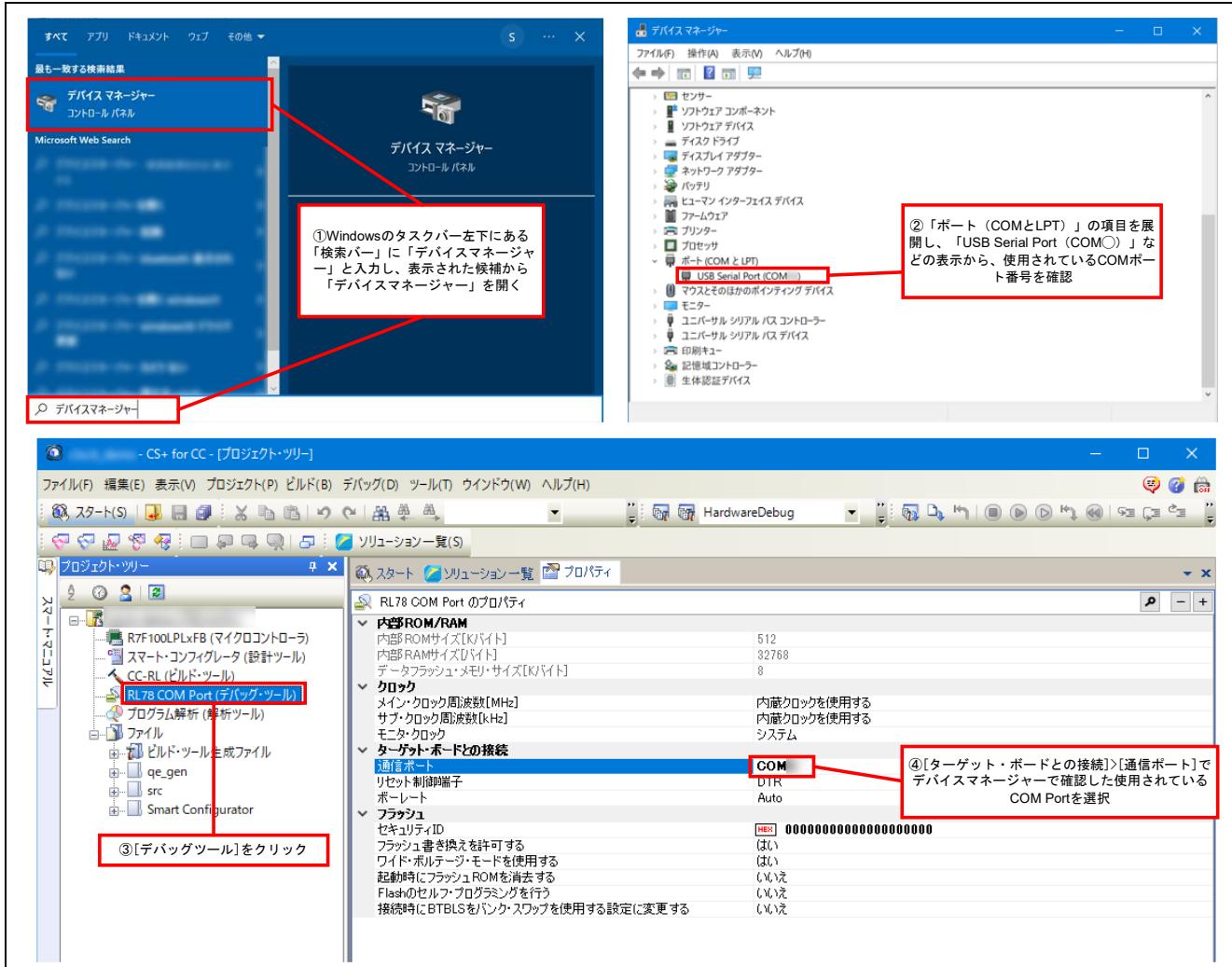
図 7-2 e² studio での COM Port 設定



7.2 CS+での COM Port 設定

パソコンと RL78/L23 ボードを USB-C ケーブルで接続した後、図 7-3 を参考にしてデバッグ時に使用する COM Port を選択してください。

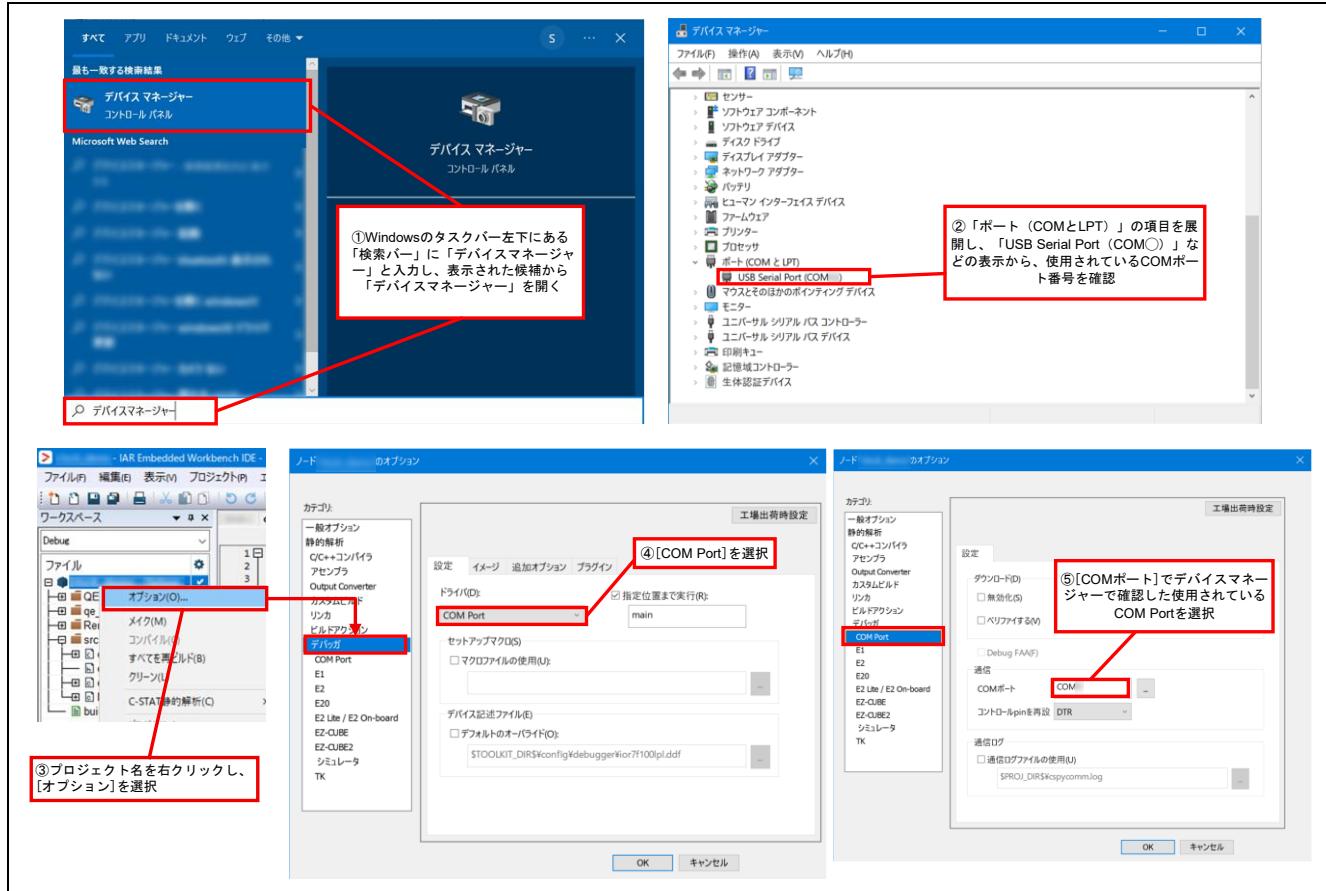
図 7-3 CS+での COM Port 設定



7.3 IAR での COM Port 設定

パソコンと RL78/L23 ボードを USB-C ケーブルで接続した後、図 7-4 を参考にしてデバッグ時に使用する COM Port を選択してください。

図 7-4 IAR での COM Port 設定



8. 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。

9. 参考ドキュメント

RL78/L23 ユーザーズマニュアル ハードウェア編 (R01UH1082)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015)

RL78/L23 Fast Prototyping Board ユーザーズマニュアル (R20UT5544)

静電容量センサマイコン静電容量タッチ導入ガイド (R30AN0424)

RL78 スマート・コンフィグレータ ユーザーガイド : IAR 編 (R20AN0581)

(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。)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2025.08.18	-	初版発行
1.10	2025.10.24	6	2 動作確認条件 IAR 環境の記載を追加
		25-26	表 5 1, 表 5 2 表 5 1 を分割し、表 5 2 を作成 IAR 版のファイル構成に関する注意書きを追加
		49	図 5 20 フローチャートの変更
		57	6.3 IAR での手順 記載を追加
		61	7.3 IAR での COM Port 設定 記載を追加

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

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。