

RL78/G23

SMS ボタン長押し／短押し判定

要旨

本アプリケーションノートでは、SNOOZE モード・シーケンサを使ったボタン長押し／短押し判定システムを構築する方法を説明します。

動作確認デバイス

RL78/G23

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、そのマイコンの使用に合わせて変更し、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. 関連アプリケーションノート	7
4. ハードウェア説明	8
4.1 ハードウェア構成例	8
4.2 使用端子一覧	8
5. ソフトウェア説明	9
5.1 動作概要	9
5.2 フォルダ構成	10
5.3 オプション・バイトの設定一覧	11
5.4 定数一覧	11
5.5 変数一覧	11
5.6 関数一覧	12
5.7 関数仕様	12
5.8 フローチャート	14
5.8.1 メイン処理	14
5.8.2 ELCL 初期設定処理	15
5.8.3 ELCL 出力開始処理	15
5.8.4 ウェイト処理	16
5.8.5 TAU0 チャンネル 7 割り込み処理	16
5.9 SNOOZE モード・シーケンサの設定	17
6. 応用例	21
6.1 r01an5609_sms_push_jadgement.scfg	21
6.1.1 クロック	22
6.1.2 システム	22
6.1.3 r_bsp	22
6.1.4 Config_LVD0	22
6.1.5 Config_INTC	23
6.1.6 Config_TAU0_7	23
6.1.7 Config_SMS	23
6.1.8 Config_PORT	23
6.2 r01an5609_sms_push_jadgement.sms	24
6.2.1 Start	24
6.2.2 Wait	25
6.2.3 P13_7 入力	25
6.2.4 更新	26
6.2.5 Finish	26
6.2.6 変数の設定	27
6.3 判定時間を変更する方法	28

7. サンプルコード29

8. 参考ドキュメント29

改訂記録30

1. 仕様

本アプリケーションノートでは、SNOOZE モード・シーケンサ（SMS）の処理でボタン（SW1）の押下時間の測定および長押し／短押しを判定し、長押しされた場合に CPU を起動する方法を示します。

SMS には SW1 が指定の時間以上押下された場合は長押し、そうでない場合は短押しと判定する処理をあらかじめ設定しておきます。STOP モードに移行後、SW1 が押下されると INTP0 を検出し、ELCL を経由して SMS が起動します。SMS は SW1 の長押し／短押し判定処理を実行し、長押しされた場合は SMS から WAKEUP 要求（INTSMSE）を出して CPU を起動させます。

図 1-1 にシステム構成例を、図 1-2 にシステム全体のフローチャートを示します。

図 1-1 システム構成

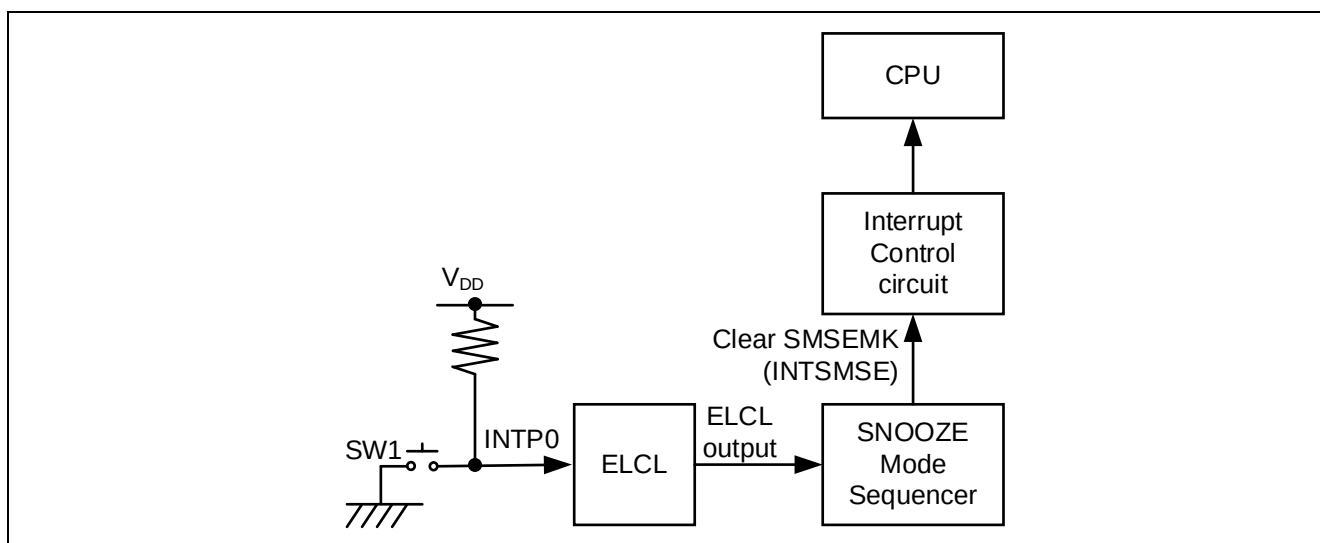
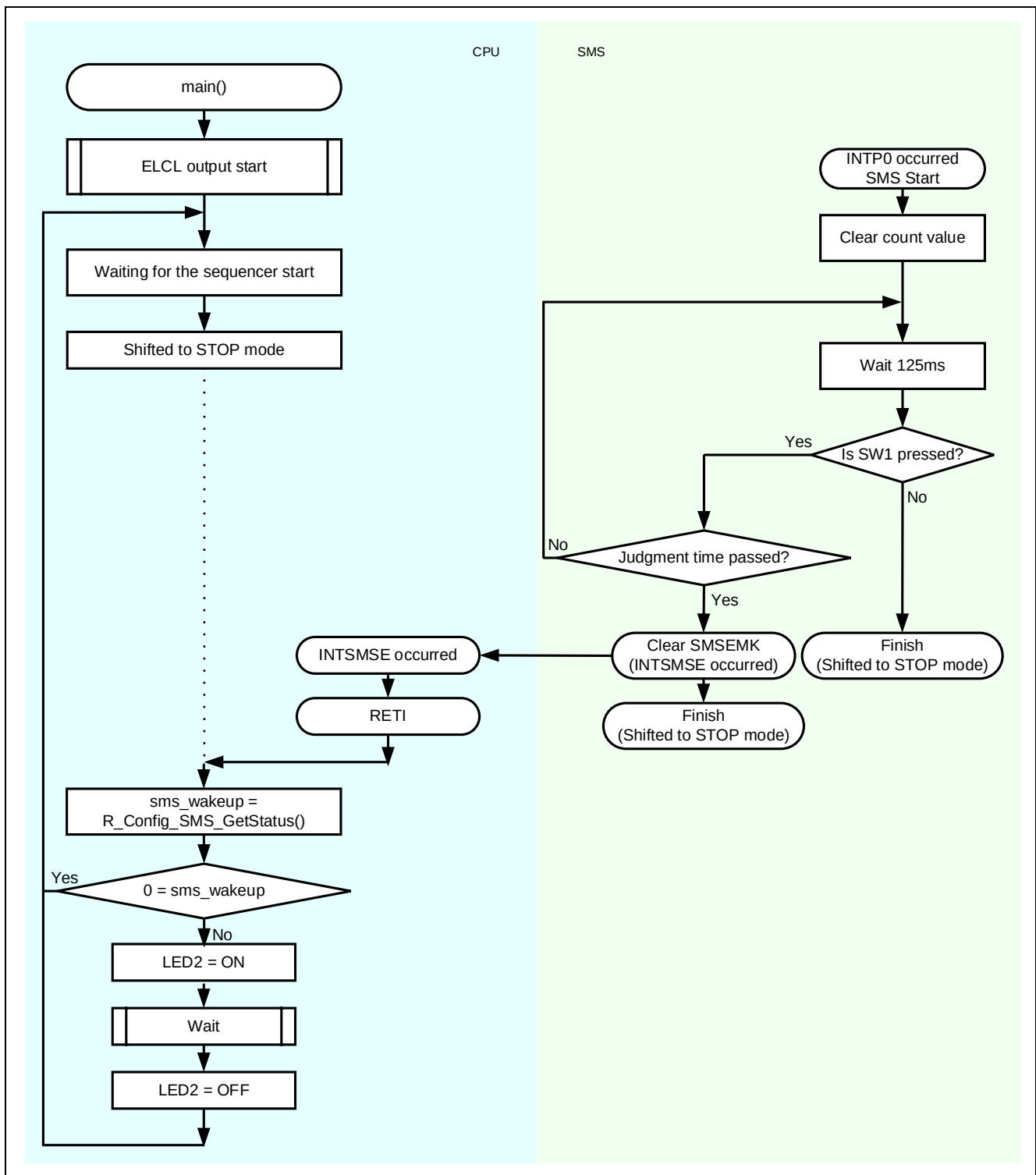


図 1-2 全体フローチャート



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/G23 (R7F100GLG)
動作周波数	- 高速オンチップ・オシレータ・クロック : 32MHz - CPU／周辺ハードウェア・クロック : 32MHz
動作電圧	3.3V - LVD 動作 (LVD0) : リセット・モード 立ち上がり時 TYP. 1.875V 立ち下がり時 TYP. 1.835V
統合開発環境 (CS+)	ルネサスエレクトロニクス製 CS+ for CC V8.11.00
C コンパイラ (CS+)	ルネサスエレクトロニクス製 CC-RL V1.13
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 2024-04 (24.4.0)
C コンパイラ (e ² studio)	ルネサスエレクトロニクス製 CC-RL V1.14
統合開発環境 (IAR)	IAR システム製 IAR Embedded Workbench for Renesas RL78 V5.10.3
C コンパイラ (IAR)	
スマート・コンフィグレータ	V.1.10.0
ボードサポートパッケージ (r_bsp)	V.1.62
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/G23 Fast Prototyping Board (RTK7RLG230CLG000BJ)

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。

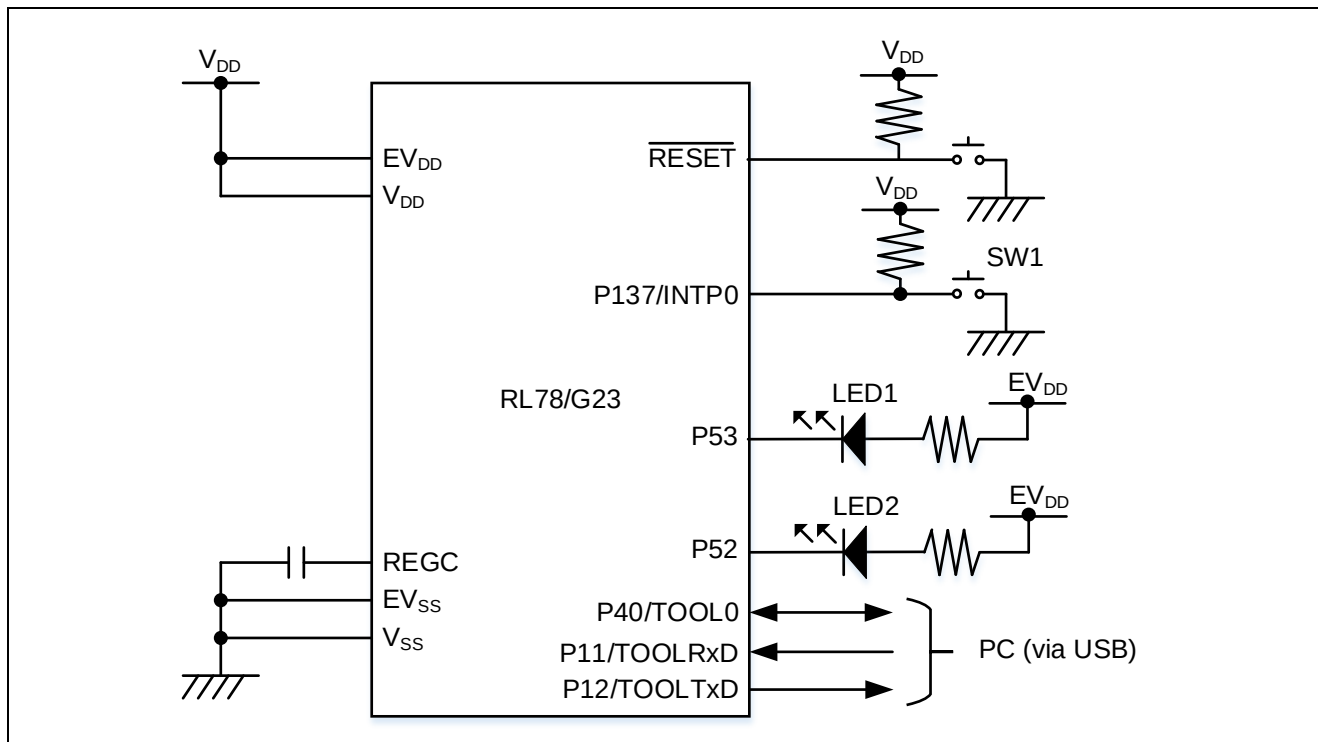
併せて参照してください。

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 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。

4.2 使用端子一覧

表 4-1 に使用端子と機能を示します。

表 4-1 使用端子と機能

端子名	入出力	内容
P53	出力	LED1 点灯（Low Active）
P52	出力	LED2 点灯（Low Active）
P137	入力	SW1（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、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。

5. ソフトウェア説明

5.1 動作概要

本サンプルコードでは、STOP モードに移行後、SW1 押下による INTP0 をトリガに STOP モードから SNOOZE モードに移行し、SW1 の押下時間の測定および長押し／短押し判定を SNOOZE モード・シーケンサ (SMS) で処理します。

SMS は起動トリガとして ELCL 出力を選択します。ELCL は入力に INTP0 を選択することで、ELCL を経由して INTP0 で SMS を起動することができます。SMS は SW1 の押下時間が指定の時間以上押下された場合は長押しと判定し、SNOOZE モードから通常動作に移行して CPU を起動させます。なお、INTP0 割り込みは STOP モードを解除しないようにマスクしておきます。

SMS が起動すると LED1 を点灯、SMS の処理が完了すると LED1 を消灯します。また、SW1 を長押しした場合は LED2 を 1 秒間点灯します。

注意 本アプリケーションノートで示すハードウェア構成のように RL78/G23 と LED を同一電源で使用する場合、LED の順方向電圧の規格を満たせず、LED が点灯しない場合があります。

本サンプルコードで行う処理の概要を以下に示します。

- (1) ELCL、SMS を設定
- (2) STOP モードへ移行
- (3) SW1 押下により INTP0 が ELCL 経由で SMS を起動し、SNOOZE モードへ移行
- (4) LED1 を点灯
- (5) SW1 の押下時間を測定
- (6) 押下時間が指定の時間を超えていない場合 (短押し) は (7) へ、超えた場合 (長押し) は (9) へ
- (7) LED1 を消灯
- (8) (2) へ戻る
- (9) LED1 を消灯し、CPU を起動
- (10) SNOOZE モードから通常動作へ移行
- (11) LED2 を 1 秒間点灯
- (12) (2) へ戻る

(4) ~ (9) は SMS で処理を行います。

5.2 フォルダ構成

表 5-1 にサンプルコードの使用しているソースファイル／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 5-1 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
¥r01an5609_sms_push_judgment<DIR>	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	
¥elcl<DIR>	ELCL 用プログラム格納フォルダ	
elcl.c	ELCL 用ソースプログラム	
elcl.h	ELCL 用ヘッダファイル	
¥smc_gen<DIR> ^{注4}	スマート・コンフィグレータ生成フォルダ	√
¥Config_INTC<DIR>	割り込み用プログラム格納フォルダ	√
Config_INTC.c	INTP0 (SW1) 用ソースファイル	√
Config_INTC.h	INTP0 用ヘッダファイル	√
Config_INTC_user.c	INTP0 用割り込みソースファイル	√ ^{注1}
¥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	POPT 用割り込みソースファイル	√ ^{注1}
¥Config_SMS<DIR>	SMS 用プログラム格納フォルダ	√
Config_SMS.c	SMS 用ソースプログラム	√
Config_SMS.h	SMS 用ヘッダファイル	√
Config_SMS_ASM.smsasm	SMS 用 ASM ソースファイル	√ ^{注3}
Config_SMS_user.c	SMS 用割り込みソースファイル	√
¥Config_TAU0_7<DIR>	TAU 用プログラム格納フォルダ	√
Config_TAU0_7.c	TAU 用ソースファイル	√
Config_TAU0_7.h	TAU 用ヘッダファイル	√
Config_TAU0_7_user.c	TAU 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 プログラム格納フォルダ	√
¥r_config<DIR>	BSP_CFG 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・コンフィグレータで生成したファイルに割り込み処理ルーチンを追加しています。

注 3. スマート・コンフィグレータで生成したファイルに LED1 の ON/OFF 処理を追加しています。

注 4. 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、r01an5609_sms_push_judgment.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 オプション・バイトの設定一覧

表 5-2 にオプション・バイト設定を示します。

表 5-2 オプション・バイト設定

アドレス	設定値	内容
000C0H/040C0H	11101 111B (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 1110B (FEH)	LVD リセット・モード 検出電圧：立ち上がり 1.875V／立下り 1.835V
000C2H/040C2H	1110 1000B (E8H)	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：32MHz
000C3H/040C3H	1000 0101B (85H)	オンチップ・デバッグ動作許可

5.4 定数一覧

表 5-3 に本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	ファイル
JUDGMENT_COUNT	24	125ms ウエイトの回数 本定数を使用した以下の時間が長押し判定時間となる 長押し判定時間 = 125ms × 本定数 本サンプルプログラムでは左記の設定値により長押し判定時間に 3 秒を設定	main.c
DELAY_MILLSECS	1000	LED2 点灯時間 (単位 : ms)	
LED2	P5_bit.no2	P52	
LED_ON	0	LED を ON するための設定値	
LED_OFF	1	LED を OFF するための設定値	

5.5 変数一覧

表 5-4 にサンプルコードで使用するグローバル変数を示します。

表 5-4 サンプルコードで使用するグローバル変数

型	変数名	内容	使用関数
volatile uint16_t	g_ms_timer	ウェイト処理のカウント値	r_ms_delay, r_Config_TAU0_7_interrupt

5.6 関数一覧

表 5-5 に本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 5-5 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_elcl_create	ELCL 初期設定処理	elcl.c
r_elcl_start	ELCL 出力開始処理	elcl.c
r_ms_delay	LED2 点灯時間のウェイト処理	Config_TAU0_7_user.c
r_Config_TAU0_7_interrupt	TAU0 チャンネル 7 割り込み処理 (LED ウェイト時間カウント用)	Config_TAU0_7_user.c

5.7 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, elcl.h
宣言	void main (void);
説明	ELCL の初期化、出力設定および SMS の長押し判定待ち時間の設定を行い、STOP モードに移行します。 SNOOZE モードから復帰すると LED2 を点灯します。r_ms_delay () 関数で点灯時間をウェイトした後 LED2 を消灯し、再度 STOP モードへ移行します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_elcl_create

概要	ELCL の初期設定処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、elcl.h
宣言	void r_elcl_create (void);
説明	ELCL の初期設定を行います。 ELCL は INTPO を ELCL 出力に選択し、SMS は ELCL 出力を起動トリガに選択します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_elcl_start

概要	ELCL 動作開始処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、elcl.h
宣言	void r_elcl_start (void);
説明	ELCL の出力を許可します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_ms_delay

概要	ウェイト処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_TAU0_7.h
宣言	void r_ms_delay (uint16_t msec);
説明	引数 msec で指定した時間 (ms) ウェイトします。 TAU0 チャンネル 7 を使ってカウントします。g_ms_timer が引数 msec 未満の場合はポーリングし、msec 以上の場合はウェイト処理を完了します。
引数	msec
リターン値	なし
備考	なし

[関数名] r_Config_TAU0_7_interrupt

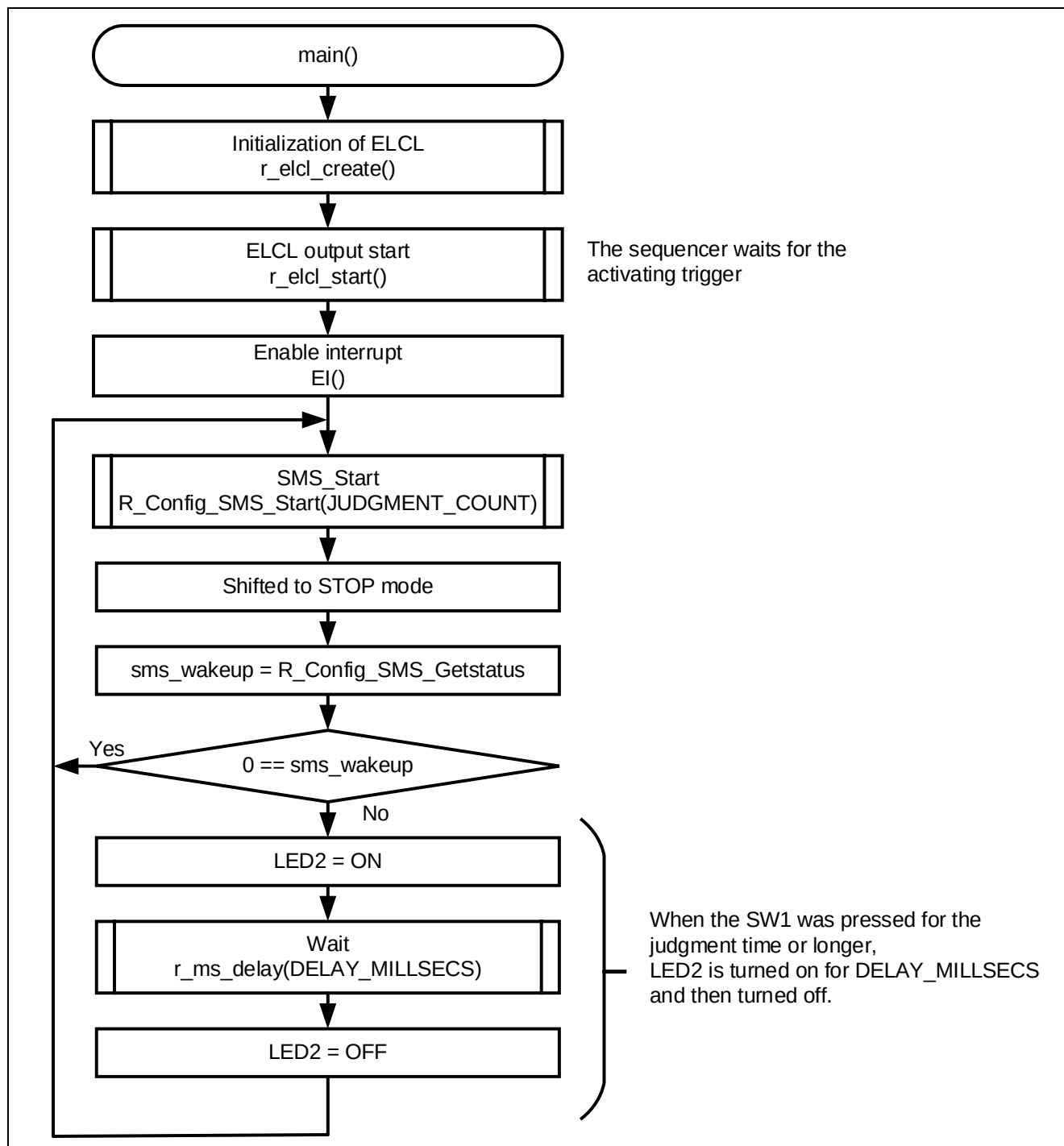
概要	TAU07 割り込み処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_TAU0_7.h
宣言	#pragma interrupt r_Config_TAU0_7_interrupt (vect=INTTM07)
説明	TAU0 チャンネル 7 の INTTM07 による割り込み処理です。 g_ms_timer をカウントアップします。
引数	なし
リターン値	なし
備考	なし

5.8 フローチャート

5.8.1 メイン処理

図 5-1 にメイン処理のフローチャートを示します。

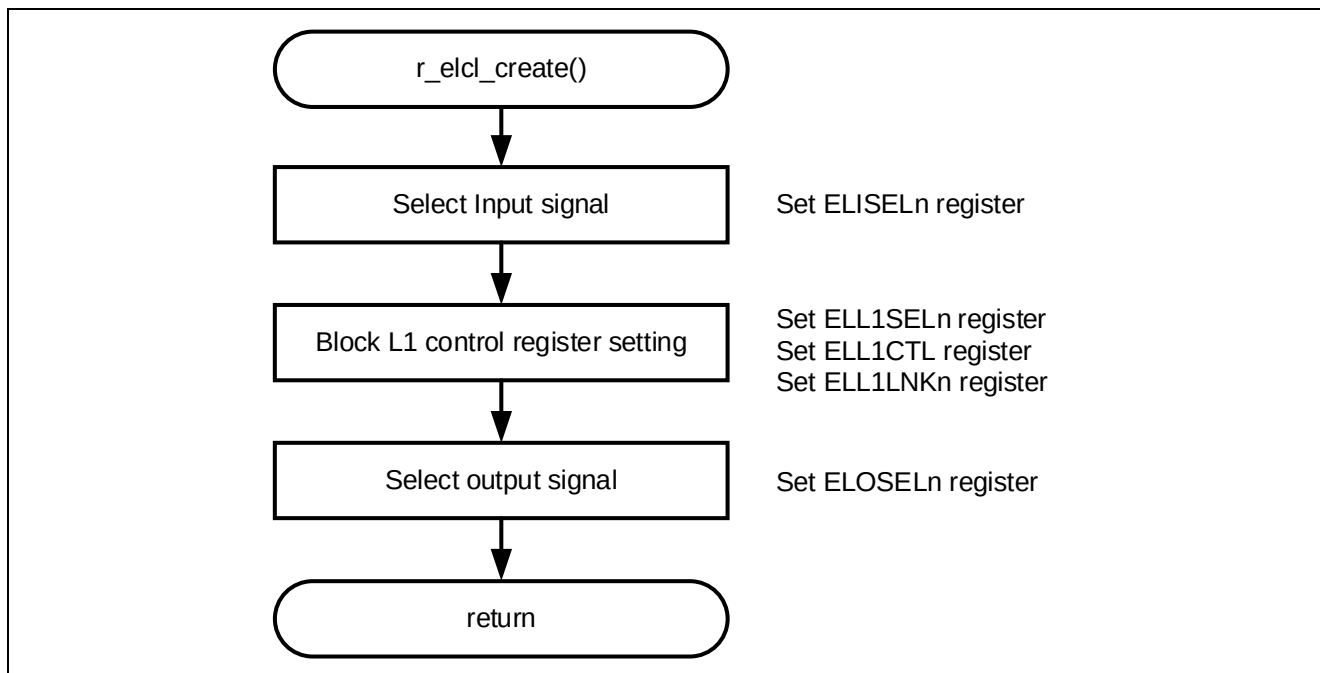
図 5-1 メイン処理



5.8.2 ELCL 初期設定処理

図 5-2 に ELCL 初期設定処理のフローチャートを示します。

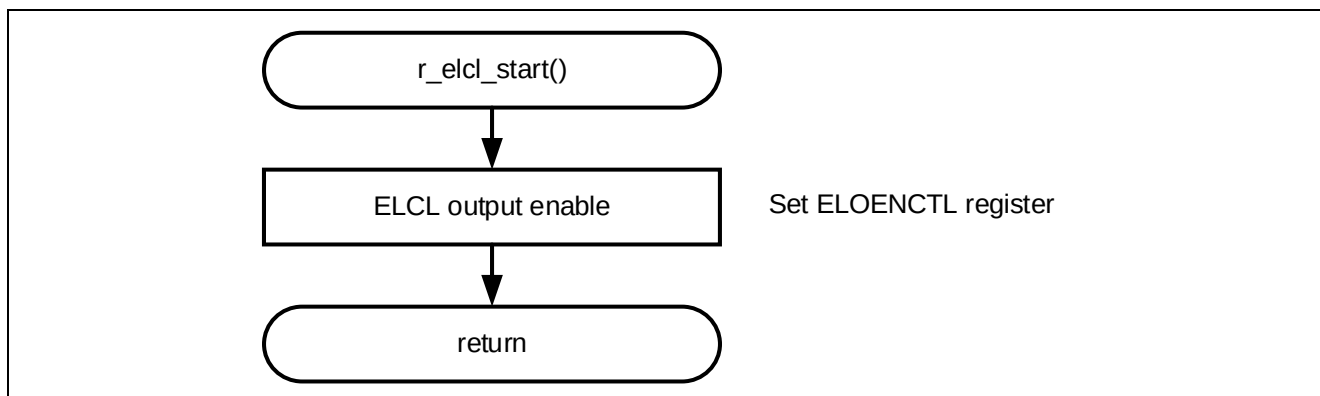
図 5-2 ELCL 初期設定処理



5.8.3 ELCL 出力開始処理

図 5-3 に ELCL 出力開始処理のフローチャートを示します。

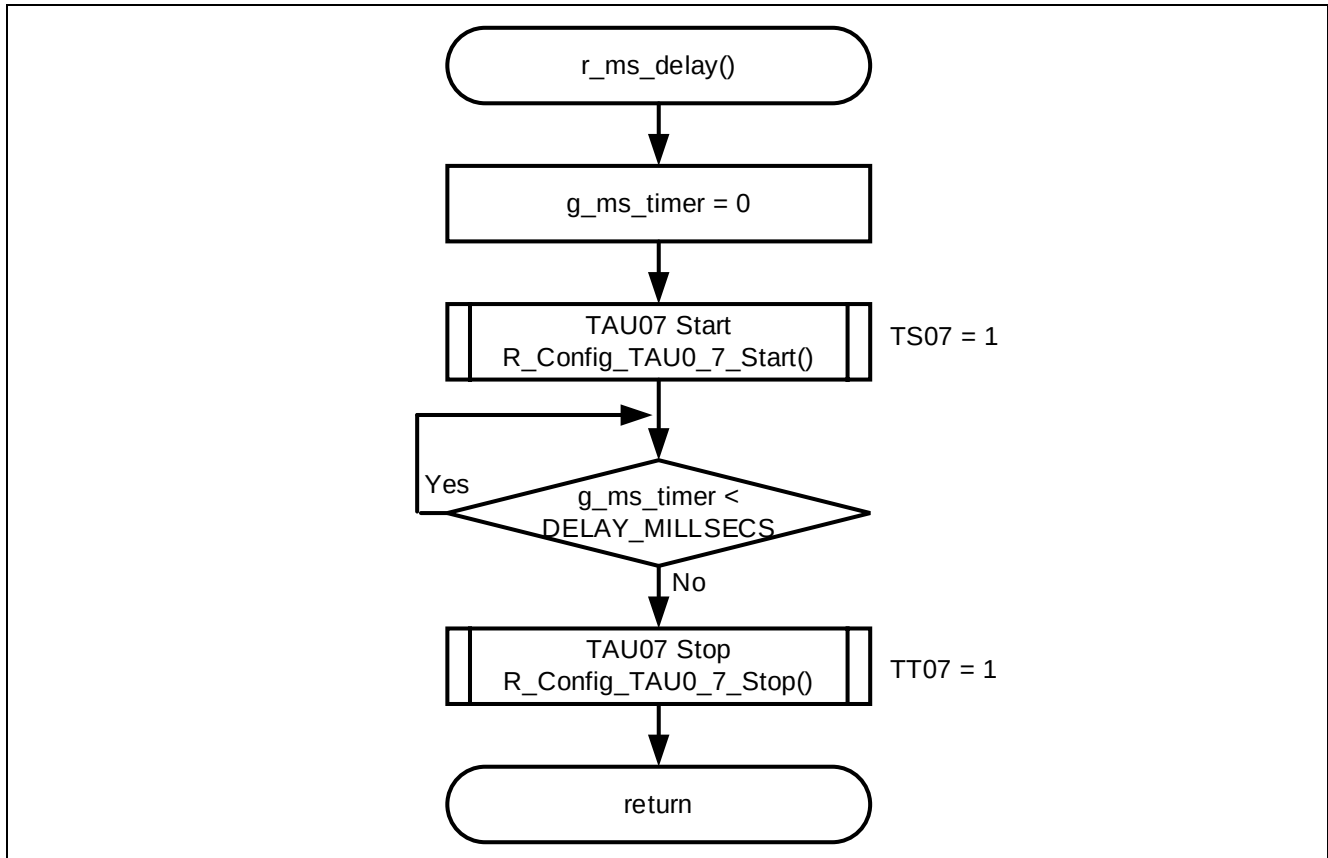
図 5-3 ELCL 出力開始処理



5.8.4 ウェイト処理

図 5-4 にウェイト処理のフローチャートを示します。

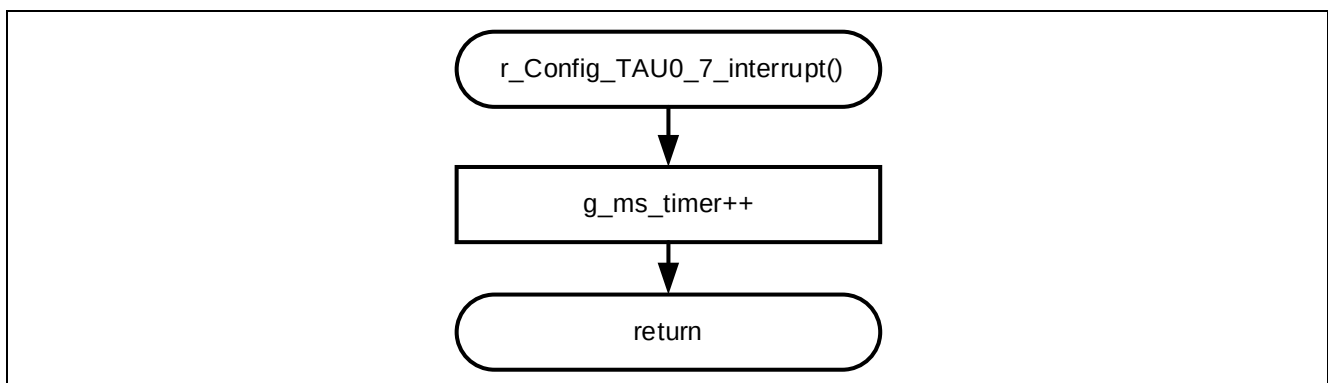
図 5-4 ウェイト処理



5.8.5 TAU0 チャンネル 7 割り込み処理

図 5-5 に TAU0 チャンネル 7 割り込み処理のフローチャートを示します。

図 5-5 TAU0 チャンネル 7 割り込み処理



5.9 SNOOZE モード・シーケンサの設定

起動トリガに設定したイベントが発生すると SMS はシーケンサ・インストラクション・レジスタ (SMSI0-31) に格納された処理コマンドを順次実行します。処理コマンドの実行では、シーケンサ汎用レジスタ (SMMSG0-15) をソース・アドレスやディスティネーション・アドレス、演算データの格納などに使用します。

SMSI0-31 と SMMSG0-15 は、SMS 用プログラム (.SMSASM ファイル) をアセンブリ言語で記述することで設定します。SMS 用プログラムは、スマート・コンフィグレータの SNOOZE Mode Sequencer コンポーネントを使い処理ブロックを組み合わせることで作成することも可能です。作成した SMS 用プログラムは SMS 用アセンブラで C 言語ファイルへ変換されプログラムに組み込まれます。

サンプルコードで実行する SMS 処理の仕様を以下に示します。

概要	SMS 処理
説明	SW1 押下による INTP0 を検出すると ELCL を介して SMS が起動し、SW1 の押下時間の測定と長押し／短押し判定を実行します。 SMS は起動後、125ms をウエイトし SW1 が押下状態かを確認します。押下状態でない場合は短押しと判定し、STOP モードに戻ります。押下されている場合、ウエイト回数のカウント値を+1 した後、カウント値が閾値以上かを確認します。閾値未満の場合は再度 125ms ウエイト処理に戻り一連の処理を繰り返します。閾値以上の場合は長押しと判定し、INTSMSE を発生します。 尚、LED1 は SMS 起動時に点灯、SMS 処理完了後に消灯します。
引数 ^{注1}	val_cnt_th
リターン値	なし
備考	本サンプルコードでは、SMS 処理中は LED1 を ON にする処理を追加しています。LED1 の ON/OFF 切り替え処理は、Config_SMS_ASM.smsasm ファイルにおいて .DEFINE 命令で "USE_LED" を定義している場合に有効です。LED1 を使用しない場合は、" .DEFINE USE_LED" をコメントアウトしてください。

注1 R_Config_SMS_Start 関数設定で指定する引数です。詳細は、6.2.1、6.2.6 を参照してください。

図 5-6 に SMS 処理のフローチャートを示します。

表 5-6～表 5-8 に SNOOZE モード・シーケンサを制御するレジスタの設定値を示します。

図 5-6 SMS 処理

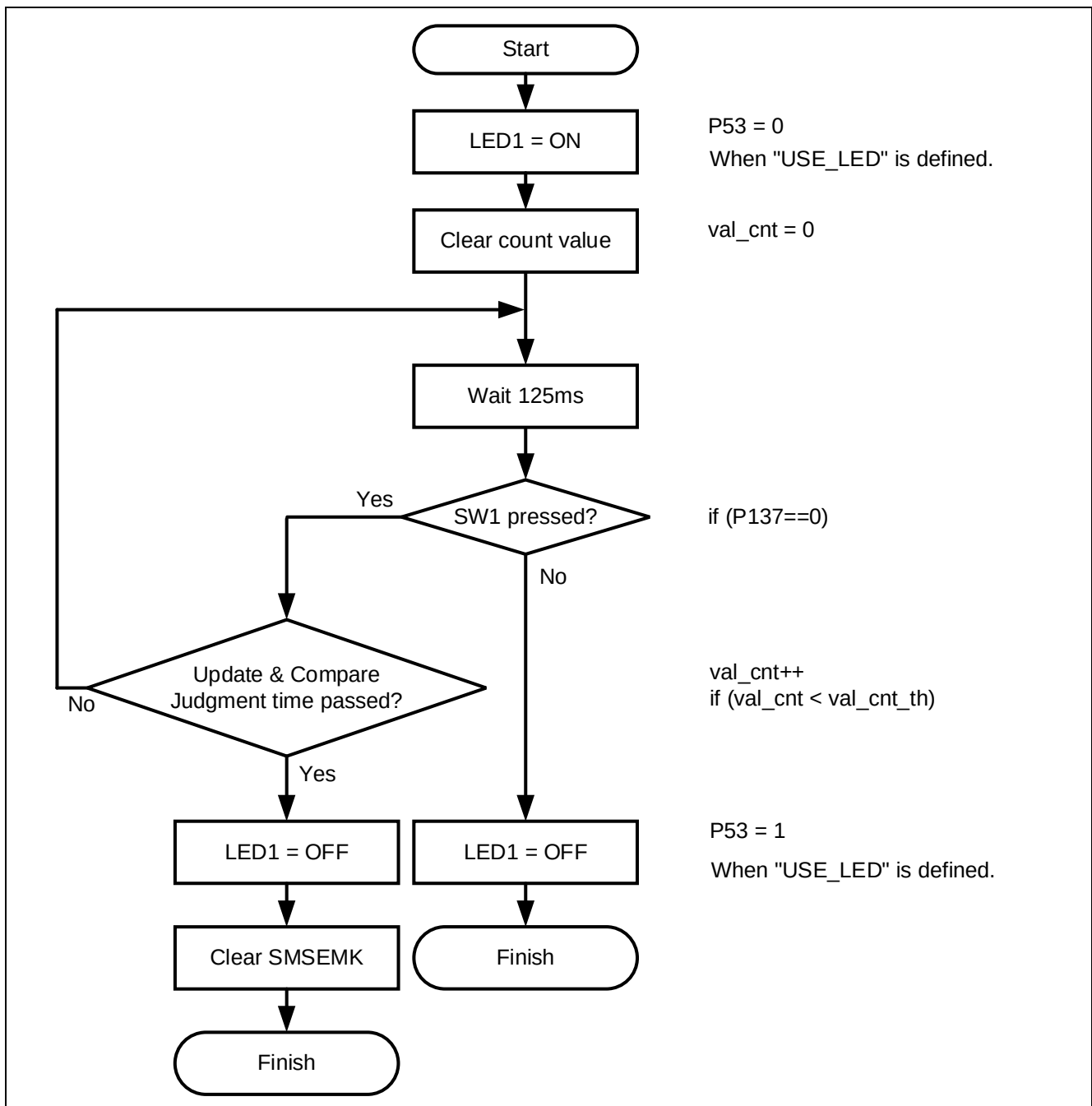


表 5-6 シーケンサ汎用レジスタ 0-15

レジスタ呼称	設定値	備考
SMSG0	0000H	固定値 : 0000H
SMSG1	0	ウェイト回数のカウント : val_cnt
SMSG2	0000H	ウェイト回数のカウント閾値 : val_cnt_th
SMSG3	&P13	P13 アドレス
SMSG4	0xFFE5	MK0H アドレス
SMSG5	1	固定値 : 1
SMSG6	&smsInitValue0	変数初期値格納アドレス (初期値 : 0)
SMSG7	0000H	未使用
SMSG8	0000H	未使用
SMSG9	0000H	未使用
SMSG10	0000H	未使用
SMSG11	0000H	未使用
SMSG12	0000H	未使用
SMSG13	0000H	未使用
SMSG14	FF05H	P5 アドレス (USE_LED 定義時のみ使用)
SMSG15	FFFFH	固定値 : FFFFH

表 5-7 シーケンサ・インストラクション・レジスタ 0-31 (USE_LED 定義時)

レジスタ呼称	設定値	備考
SMSI0	5E30H	CLR1 [SMSG14+0].3
SMSI1	3610H	MOVW SMSG1, [SMSG6+0]
SMSI2	9805H	WAIT 128, 5
SMSI3	6370H	MOV1 SCY, [SMSG3+0].7
SMSI4	8031H	BNC \$__portinbranch_finish
SMSI5	4E30H	SET1 [SMSG14+0].3
SMSI6	F000H	FINISH
SMSI7	7150H	ADDW SMSG1, SMSG5
SMSI8	7122H	CMPW SMSG1, SMSG2
SMSI9	8F90H	BC \$__updbranch_do
SMSI10	4E30H	SET1 [SMSG14+0].3
SMSI11	5440H	CLR1 [SMSG4+0].4
SMSI12	F000H	FINISH
SMSI13-31	0000H	未使用

表 5-8 シーケンサ・インストラクション・レジスタ 0-31 (USE_LED 未定義時)

レジスタ呼称	設定値	備考
SMSI0	3610H	MOVW SMSG1, [MSG6+0]
SMSI1	9805H	WAIT 128, 5
SMSI2	6370H	MOV1 SCY, [MSG3+0].7
SMSI3	8021H	BNC \$ __portinbranch_finish
SMSI4	F000H	FINISH
SMSI5	7150H	ADDW SMSG1, MSG5
SMSI6	7122H	CMPW SMSG1, MSG2
SMSI7	8FA0H	BC \$ __updbranch_do
SMSI8	5440H	CLR1 [MSG4+0].4
SMSI9	F000H	FINISH
SMSI10-31	0000H	未使用

6. 応用例

本アプリケーションノートは、サンプルコードの他に以下のスマート・コンフィグレータの設定ファイルを格納しています。

r01an5609_sms_push_jadgement.scfg

r01an5609_sms_push_jadgement.sms

ファイルの説明と使用する上での設定例および注意事項を以下に示します。

6.1 r01an5609_sms_push_jadgement.scfg

サンプルコードで使用しているスマート・コンフィグレータの設定ファイルです。スマート・コンフィグレータで設定されている全ての機能が含まれています。サンプルコードの設定は以下の通りです。

表 6-1 スマート・コンフィグレータの設定値

タグ名	コンポーネント	内容
クロック	-	動作モード：高速メインモード 2.4 (V) ~5.5 (V) EV _{DD} 設定：1.8V ≤ EV _{DD0} < 5.5V 高速オンチップ・オシレータ：32MHz f _{IHP} ：32MHz f _{CLK} ：32MHz（高速オンチップ・オシレータ） f _{SXP} ：32.768kHz（低速オンチップ・オシレータ）
システム	-	オンチップ・デバッグ動作設定：COM ポート ^{注1} 疑似 RRM/DMM 機能設定：使用する Start/Stop 関数機能設定：使用しない トレース機能設定：使用する セキュリティ ID 設定：セキュリティ ID を設定する セキュリティ ID：0x00000000000000000000 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
コンポーネント	r_bsp	Start up select：Enable (use BSP startup) Control of invalid memory access detection：Disable RAM guard space (GRAM0-1)：Disabled Guard of control registers of port function (GPORT)：Disabled Guard of registers of interrupt function (GINT)：Disabled Guard of control registers of clock control function, voltage detector, and RAM parity error detection function (GCSC)：Disabled Data flash access control (DFLEN)：Disables Initialization of peripheral functions by Code Generator/Smart Configurator：Enable API functions disable：Enable Parameter check enable：Enable Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode：High-speed Enable user warm start callback (PRE)：Unused Enable user warm start callback (POST)：Unused Watchdog Timer refresh enable：Unused
	Config_LVD0	動作モード設定：リセット・モード 電圧検出設定：リセット発生電圧 (V _{LVD0})：1.835 (V)

表 6-2 スマート・コンフィグレータの設定値

タグ名	コンポーネント	内容
コンポーネント	Config_INTC	INTP0 設定：使用する 有効エッジ：立ち下がリエッジ 優先順位：レベル 3
	Config_TAU0_7	コンポーネント：インターバル・タイマ 動作モード：16 ビット・カウンタ・モード リソース：TAU0_7 動作クロック：CK00 クロックソース：f _{CLK} インターバル時間：1ms 割り込み設定：使用する 優先順位：レベル 2
	Config_SMS	コンポーネント：SNOOZE モード・シーケンサ 起動トリガ：ELCL 出力信号
	Config_PORT	コンポーネント：ポート ポート選択：PORT5 P52：出力（1 を出力） P53：出力（1 を出力）

注 1. IAR 使用時は以下の設定にしてください。

オンチップ・デバッグ動作設定：エミュレータを使う

エミュレータ設定：E2 エミュレータ Lite

6.1.1 クロック

サンプルコードで使用するクロックの設定を行います。

6.1.2 システム

サンプルコードのオンチップ・デバッグ設定を行います。

「オンチップ・デバッグ動作設定」、「セキュリティ ID 認証失敗時の設定」は、「表 5-2 オプション・バイト設定」の「オンチップ・デバッグ動作許可」に影響を与えます。設定を変更する際は注意してください。

6.1.3 r_bsp

サンプルコードのスタートアップの設定を行います。

6.1.4 Config_LVD0

サンプルコードの電源管理の設定を行います。

「表 5-2 オプション・バイト設定」の「LVD0 の設定」に影響を与えます。設定を変更する際は注意してください。

6.1.5 Config_INTC

サンプルコードで使用する割り込みの設定を行います。

サンプルコードでは、外部マスカブル割り込み（INTP0）を設定します。INTP0 を使用しない場合は削除してください。

6.1.6 Config_TAU0_7

サンプルコードの TAU07 の設定を行います。

サンプルコードでは、LED2 の点灯時間をカウントするのに使用します。INTP0 を使用しない場合は削除してください。

6.1.7 Config_SMS

サンプルコードの SMS の設定を行います。

詳細は、「6.2 r01an5609_sms_push_jadgement.sms」を参照してください。

6.1.8 Config_PORT

サンプルコードのポートの設定を行います。

サンプルコードでは LED1 の制御に P53 を、LED2 の制御に P52 を使用します。

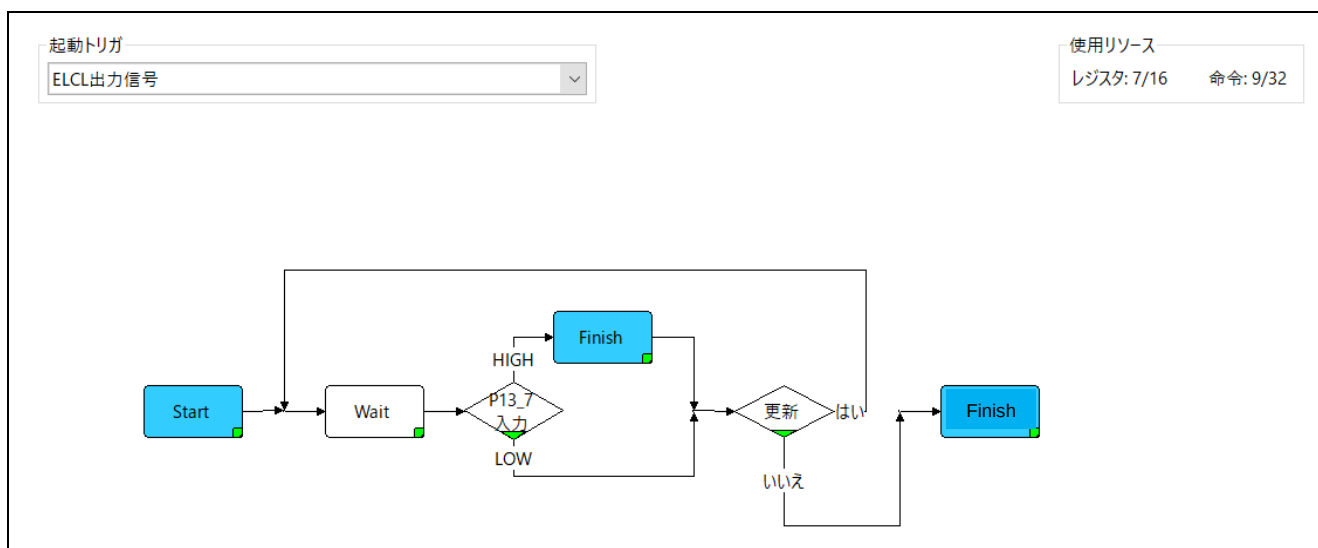
6.2 r01an5609_sms_push_jadgement.sms

Config_SMS 単体のデータです。サンプルコードでは、SMS の起動に ELCL 出力信号を使用します。別途 ELCL の設定を行う必要がありますので注意してください。

また、r01an5609_sms_push_jadgement.sms は、別プロジェクトのスマート・コンフィグレータにインポートすることが可能です。別プロジェクトに SMS コンポーネントを設定後、[インポート] → [ブラウザ] で「r01an5609_sms_push_jadgement.sms」を選択するとインポートします。

スマート・コンフィグレータにインポートすると、図 6-1 のようなフロー図となります。このフロー図は、「図 5-6 SMS 処理」と同じです。

図 6-1 Config_SMS のフロー図

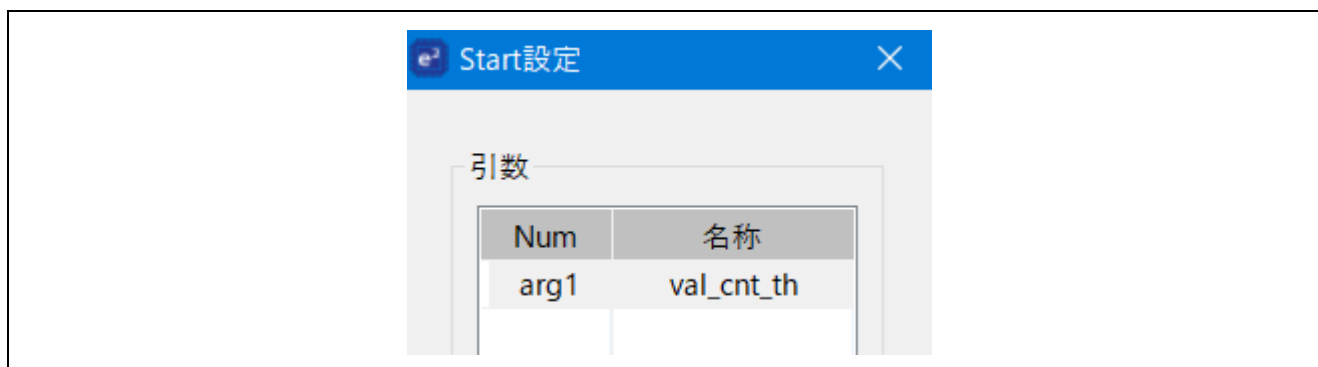


各ブロックの説明を以下に示します。

6.2.1 Start

SMS 起動時に SMS 開始関数 (R_Config_SMS_Start 関数) で引数として渡された JUDGMENT_COUNT の値が val_cnt_th (ウェイト回数のカウント閾値) に設定されます。

図 6-2 Start 設定

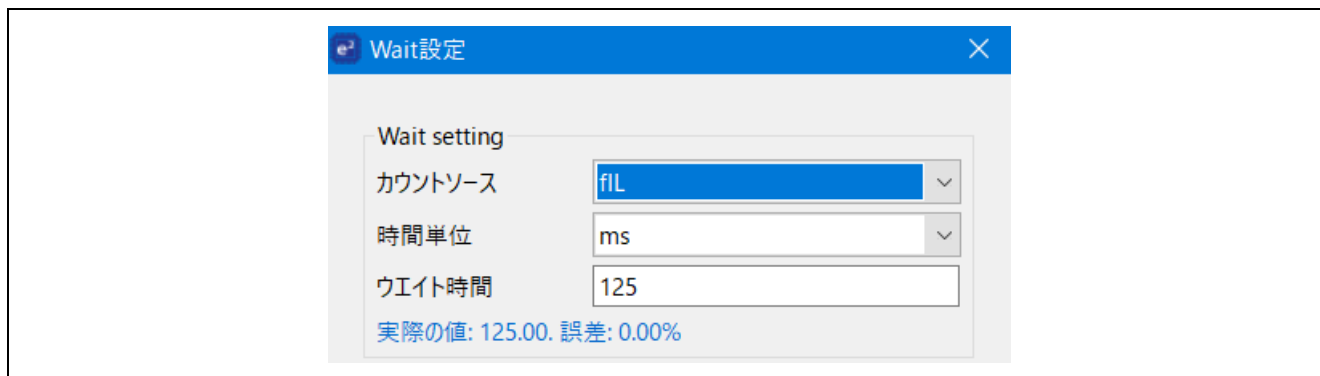


6.2.2 Wait

SMS の処理を設定したウェイト時間分待ちます。サンプルコードの場合は、カウントソース（fIL）で 125ms の間、処理を待ちます。

待ち時間を変更することもできますが、設定できない値を設定した場合は赤字となります。変更する場合は注意してください。

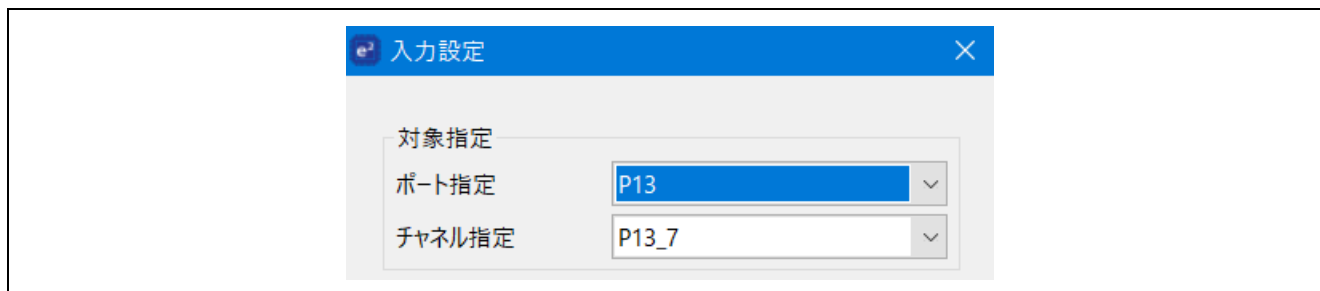
図 6-3 Wait 設定



6.2.3 P13_7 入力

対象指定したポートの入力値を確認して処理を分岐します。サンプルコードの場合は、P13_7 の値を対象端子としています。ポートの設定（例えば入力モードへの変更）は、Config_Port のコンポーネントで行ってください。サンプルコードで使用している P13_7 は入出力モードが入力固定のため設定不要です。

図 6-4 入力設定



6.2.4 更新

変数 val_cnt に設定された値 "0"（初期値）に固定値 "1" を加算し、この値を比較値 "val_cnt_th" と比較式 ($\text{val_cnt} < \text{val_cnt_th}$) で比較します。値が比較と一致するときは元に戻り、この時、val_cnt には更新された値 "1" が上書きされます。値が一致する場合は次に進みます。

図 6-5 更新設定

6.2.5 Finish

STOP モードに移行します。サンプルコードでは、戻り値を使用しません。

図 6-6 Finish 設定

6.2.6 変数の設定

SMS で使用している変数の設定を以下に示します。

表 6-3 SMS で使用している変数

データ名	初期化	初期値	説明
val_cnt	SMS で毎回初期化	0	ウェイト回数のカウント値を格納します。
val_cnt_th	SMS の開始関数で引数渡し	-	ウェイト回数のカウント閾値を格納します。 R_Config_SMS_Start 関数で JUDGMENT_COUNT の値が引数として設定されます

6.3 判定時間を変更する方法

以下の計算式は、サンプルコードの判定時間です。判定時間を変更する方法を説明します。

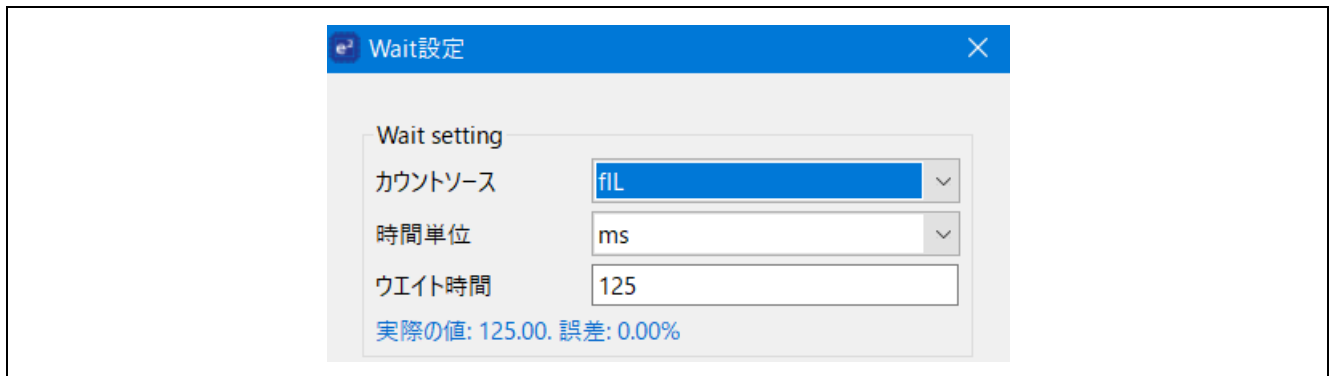
$$\text{判定時間} = 125\text{ms} \times \text{JUDGMENT_COUNT (ms)}$$

定数「JUDGMENT_COUNT」は、以下のように設定されています。

```
#define JUDGMENT_COUNT    (24U)
```

図 6-7 に示すように、125ms は SMS の Wait ブロックで設定しています。

図 6-7 Wait 設定



JUDGMENT_COUNT の値と Wait 設定の設定値を変更することで判定時間を変更できます。

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

RL78/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896J)

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

SMS アセンブラ ユーザーズマニュアル【暫定版】 (R20UT4792J)

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)

(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。)

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	Apr.13.21	-	初版発行
1.10	Jun.1.21	6	ツールバージョンを更新 表 2-1 動作確認条件 統合開発環境 (CS+) : CS+ for CC V E8.05.00f ⇒ V8.05.01 C コンパイラ (CS+) : CC-RL V1.09 ⇒ V1.10 統合開発環境 (e ² studio) : e ² studio 2021-01 (21.1.0) ⇒ 2021-04 (21.4.0) C コンパイラ (e ² studio) : CC-RL V1.09 ⇒ V1.10 統合開発環境 (IAR) : V4.20.1 ⇒ V4.21.1 スマート・コンフィグレータ : V.1.0.0 ⇒ V.1.0.1 ボードサポートパッケージ (r_bsp) : V.1.0.0 ⇒ V.1.10
		6, 8 21,22	COM ポート対応に伴う変更 表 2-1 動作確認条件 エミュレータ : E2 エミュレータ Lite ⇒ CS+, e ² studio : COM ポート IAR : E2 エミュレータ Lite 図 4-1 ハードウェア構成例 P11/TOOLRxD、P12/TOOLTxD 追加 表 6-1 スマート・コンフィグレータの設定 注 1 の追加
		10, 12	IAR 版サンプルコード更新に伴う変更 表 5-1 フォルダ構成、注 4 フォルダ構成の参照ドキュメントに関する注意書き追加 5.7 関数仕様 [関数名]main, ヘッダ e ² studio, CS+ : r_smc_entry.h, elc.h IAR : ior7f100g.h, ior7f100g_ext.h, r_cg_macrodriver.h, Config_SMS.h, Config_TAU0_7.h, elc.h ⇒ r_smc_entry.h, elc.h
		21, 22	クロック略称変更 表 6-1 スマート・コンフィグレータの設定値 クロック : f _{SXL} ⇒ f _{SXP}
		28	RL78 スマート・コンフィグレータ ユーザーガイドの追記 8. 参考ドキュメント 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 : e ² 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)
1.20	Feb.15.22	5	図 1-2 全体フローチャートを修正
		18	図 5-6 SMS 処理を修正
1.30	Jan.9.24	-	SMS 処理のフロー変更
1.40	Jul.24.24	6	表 2-1 動作確認条件を修正
		19	表 5 7 シーケンサ・インストラクション・レジスタ 0-31 (USE_LED 定義時) を修正
		20	表 5 8 シーケンサ・インストラクション・レジスタ 0-31 (USE_LED 未定義時) を修正

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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

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

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因し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。