

RL78/G23

SMS LED 点滅調光制御

要旨

本アプリケーションノートでは、SNOOZE モード・シーケンサと 32 ビット・インターバル・タイマを利用して LED を制御する方法を説明します。

32 ビット・インターバル・タイマの割り込み毎に SNOOZE モード・シーケンサを起動し、LED の光量を 100% から 0% に変化させ、LED が徐々に暗くなっていくように制御します。

LED 制御に必要な PWM 出力を SNOOZE モード・シーケンサが制御することで、CPU 制御よりも低消費電力を実現できます。

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	11
5.7 関数仕様	11
5.8 フローチャート	12
5.8.1 メイン処理	12
5.9 SNOOZE モード・シーケンサの設定	13
6. 応用例	16
6.1 r01an5611_sms_led_dimming.scfg	16
6.1.1 クロック	17
6.1.2 システム	17
6.1.3 r_bsp	17
6.1.4 Config_LVD0	18
6.1.5 Config_IT000_ITL001	18
6.1.6 Config_TAU0_0	18
6.1.7 Config_SMS	18
6.1.8 Config_PORT	18
6.2 r01an5611_sms_led_dimming.sms	19
6.2.1 Start	19
6.2.2 TAU デューティ比／周期 設定	20
6.2.3 2byte 計算	20
6.2.4 比較	21
6.2.5 2byte 転送	21
6.2.6 Finish	22
6.2.7 変数の設定	22
6.3 LED 点滅時間を変更する例	23
7. サンプルコード	24
8. 参考ドキュメント	24

改訂記録..... 25

1. 仕様

HALT モード（CPU 動作クロック：停止）期間中に、SNOOZE モード・シーケンサ（SMS）によって PWM 出力を制御し、LED の光量を徐々に下げます。

タイマ・アレイ・ユニット（TAU）の PWM 出力機能で LED の光量を制御します。SMS には PWM 出力のデューティ比を変更する処理をあらかじめ設定しておきます。PWM 出力を制御する間隔は 32 ビット・インターバル・タイマ（TML32）を用いて設定し、SMS は TML32 の割り込み（INTITL）をトリガに起動します。SMS の起動により、PWM 出力のデューティ比を変更することで LED の光量を制御します。

図 1-1 にシステム構成を、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

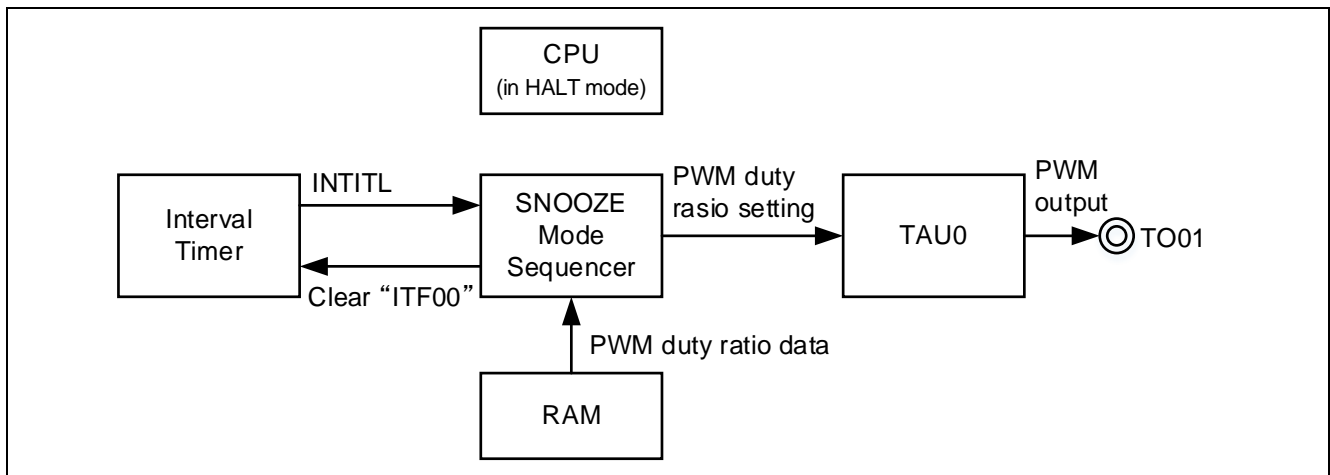
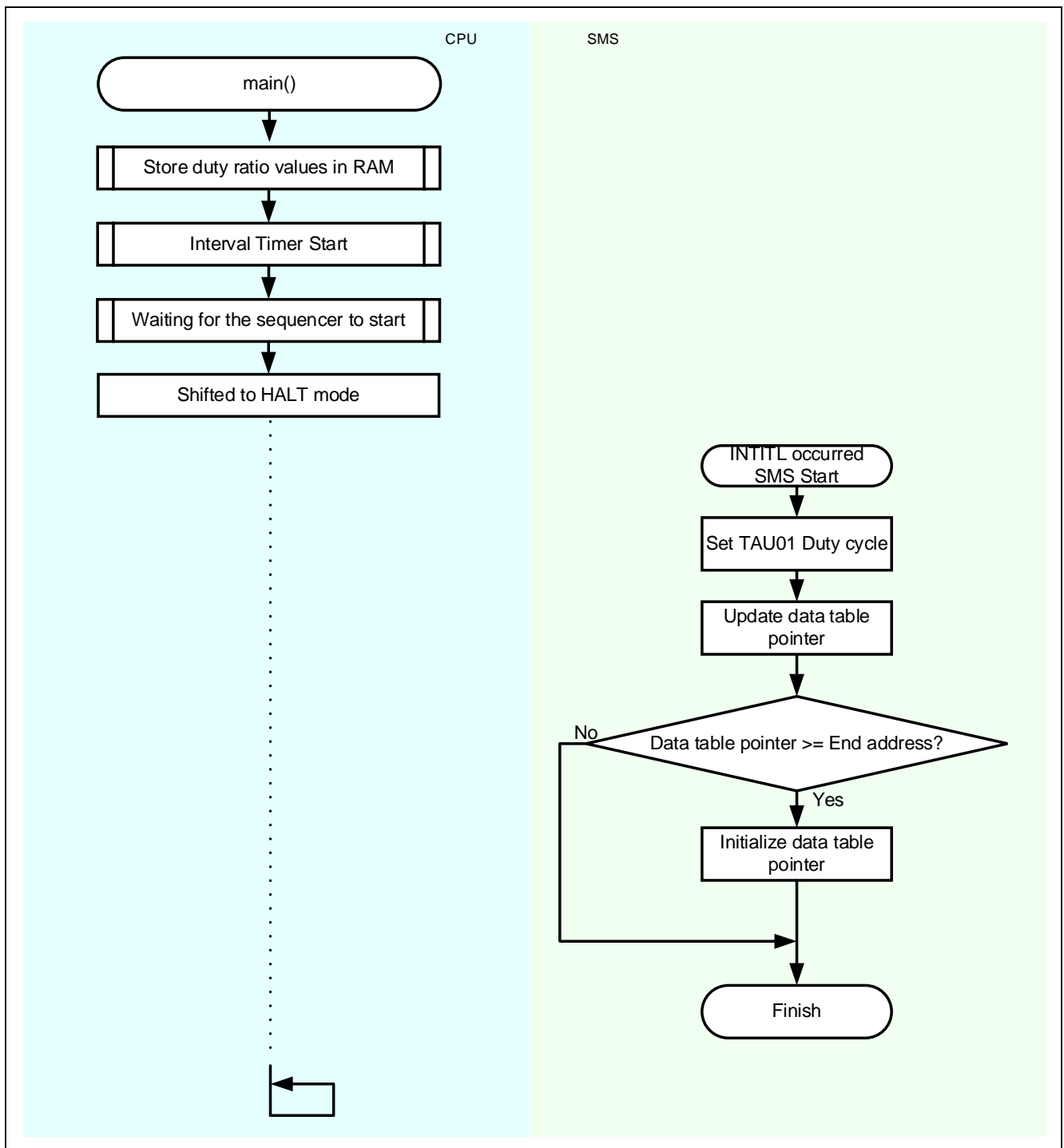


図 1-2 全体フローチャート



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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 3.3V LVD0 動作 (V_{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 V8.05.00
C コンパイラ (CS+)	ルネサスエレクトロニクス製 CC-RL V1.10
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 2021-04 (21.4.0)
C コンパイラ (e ² studio)	ルネサスエレクトロニクス製 CC-RL V1.10
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.1.0.1
ボードサポートパッケージ (r_bsp)	V.1.10
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

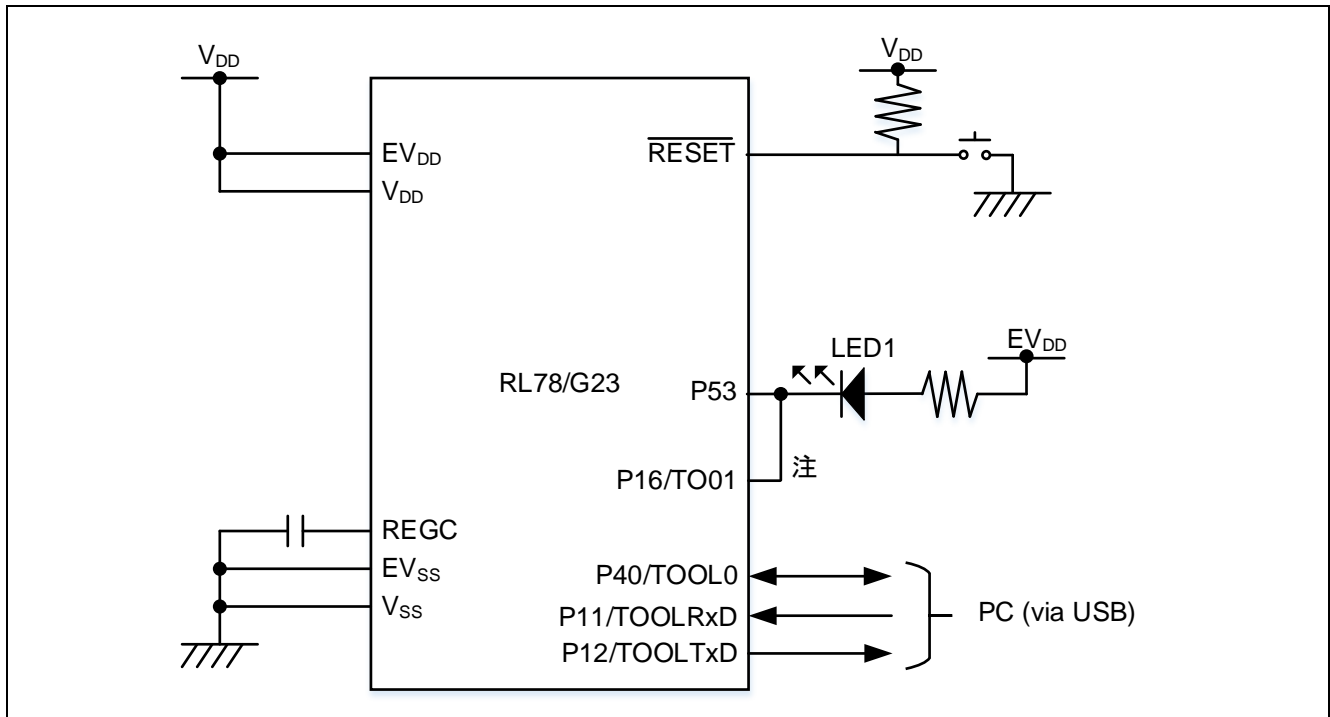
本アプリケーションノートに関連するアプリケーションノートを以下に示します。
併せて参照してください。

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 本アプリケーションノートでは、ボード (RTK7RLG230CLG000BJ) 上の LED を使用するため、P16 と P53 を外部で接続しています。必要に応じて、回路を変更してください。

- 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
P16	出力	PWM 出力 (LED1 制御)
P53	入力	P53 出力を禁止

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、32 ビット・インターバル・タイマ (TML32) の割り込み要求 (INTITL) によって SNOOZE モード・シーケンサ (SMS) を起動し、SMS でタイマ・アレイ・ユニット (TAU) の PWM 出力のデューティ比を変更します。

PWM 出力のデューティ比を変えることで LED の光量を変更します。PWM 出力の周期を 1ms に固定し、デューティ比を 0% から 100% へ 32 ステップで上げていくことにより、LED の光量を徐々に下げます。デューティ比の値は、あらかじめ RAM に格納しておきます。SMS では、RAM に格納された値を読み出し、タイマ・データ・レジスタ (TDR) に書き込みます。

また、初期設定後、消費電力低減のために HALT モードへ移行します。

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) PWM 出力のデューティ比設定用データテーブルを作成
- (2) TAU のカウント開始
- (3) TML32 のカウント開始
- (4) HALT モードへ移行
- (5) TML32 のコンペアー致で SMS 起動
- (6) ポインタが示すデータテーブルの値をタイマ・データ・レジスタへ書き込む
- (7) ポインタを進めてデータテーブルの最後なら (8) へ、最後でなければ (9) へ
- (8) ポインタをデータテーブルの先頭アドレスにセット
- (9) SMS は待機状態へ移行し (5) へ

(6) ~ (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
¥r01an5611_sms_led_dimming<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	
¥smc_gen<DIR> ^{注2}	スマート・コンフィグレータ生成フォルダ	√
¥Config_ITL000_ITL001<DIR>	TML32 用プログラム格納フォルダ	√
Config_ITL000_ITL001.c	TML32 用ソースファイル	√
Config_ITL000_ITL001.h	TML32 用ヘッダファイル	√
Config_ITL000_ITL001_user.c	TML32 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	PORT 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	√
Config_SMS_user.c	SMS 用割り込みソースファイル	√ ^{注1}
¥Config_TAU0_0<DIR>	TAU 用プログラム格納フォルダ	√
Config_TAU0_0.c	TAU 用ソースファイル	√
Config_TAU0_0.h	TAU 用ヘッダファイル	√
Config_TAU0_0_user.c	TAU 用割り込みソースファイル	√ ^{注1}
¥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. 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、r01an5611_sms_led_dimming.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	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 1110B (FEH)	LVD0 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
BRIGHTNESS_RANK	32	デューティ比変更のステップ数	main.c

5.5 変数一覧

本サンプルコードではグローバル変数を使用しません。

5.6 関数一覧

表 5-4 にサンプルコードで使用する関数を示します。

表 5-4 関数一覧

関数名	概要	ソースファイル
main	メイン処理	main.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

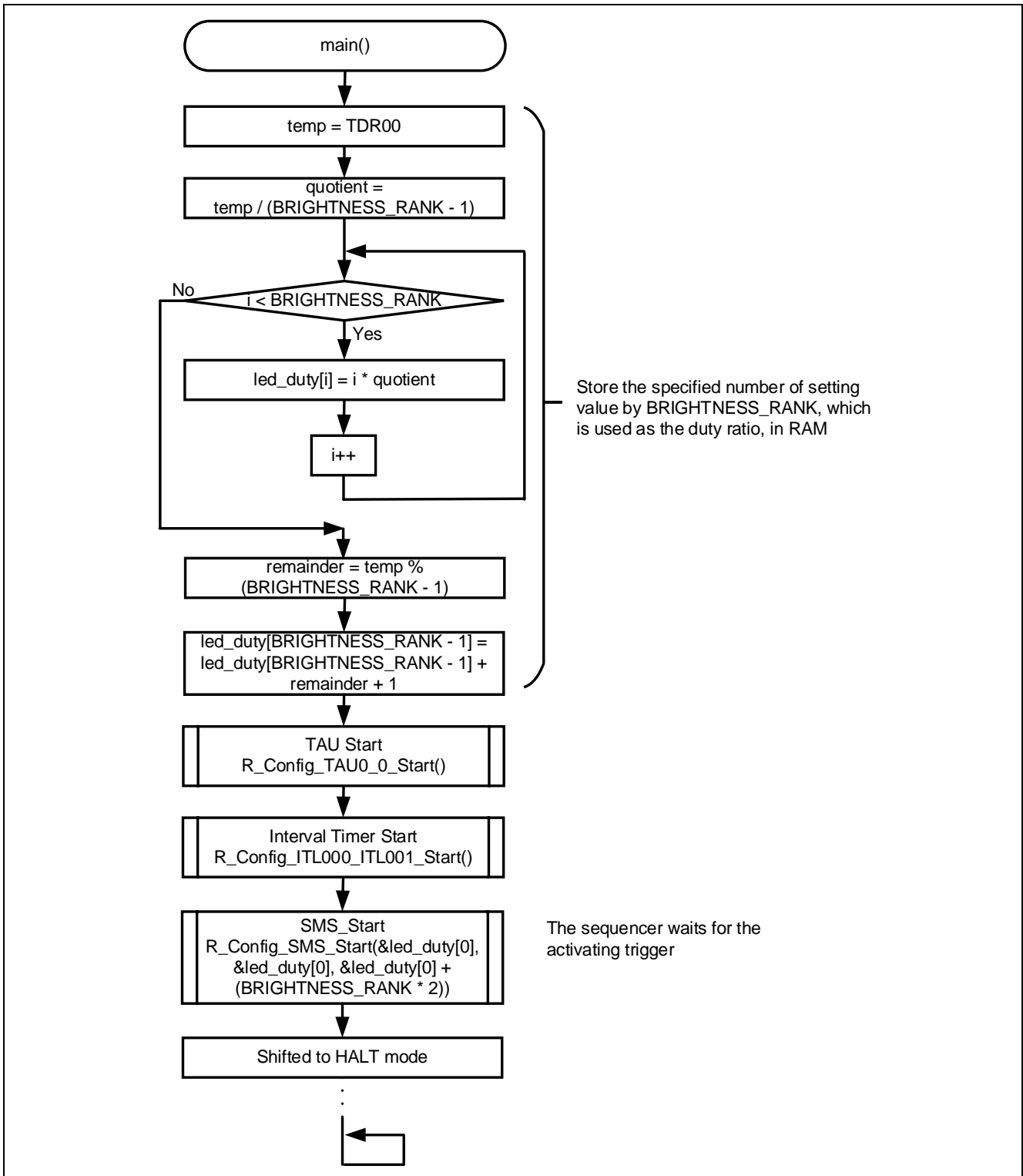
概要	メイン処理
ヘッダ	r_smc_entry.h
宣言	void main (void);
説明	デューティ比の値をデータテーブルとして定義します。その後、TML32 と TAU の動作を開始して HALT モードへ移行します。
引数	なし
リターン値	なし
備考	なし

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



5.9 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 (SMSIO-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 (SMSGO-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。

SMSIO-31 と SMSGO-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	TML32 割り込みにより SMS が起動し、PWM 出力のデューティ比を定義しているデータテーブルからデータを取り出し、TAU のタイマ・データ・レジスタへ書き込みます。
引数 ^{注1}	val_ram_d, val_ram_s, val_ram_e
リターン値	なし
備考	なし

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、6.2.7 を参照してください。

図 5-2 に SMS 処理のフローチャートを示します。

表 5-5～表 5-6 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 5-2 SMS 処理

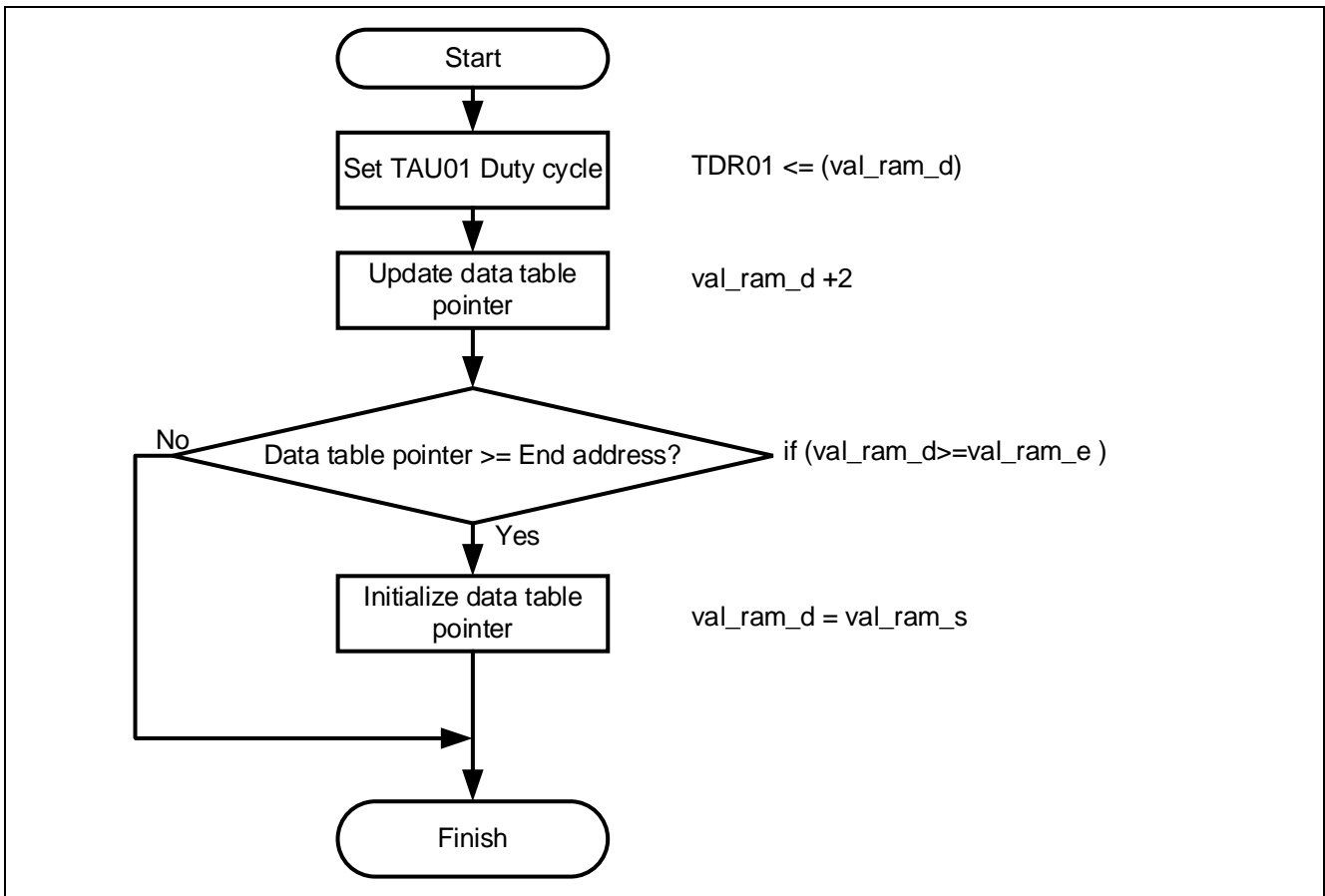


表 5-5 シーケンサ汎用レジスタ 0-15

レジスタ呼称	設定値	備考
SMSG0	0000H	固定値 : 0000H
SMSG1	0000H	データテーブルポインタ : val_ram_d
SMSG2	0000H	データテーブル終了アドレス : val_ram_e
SMSG3	0000H	データテーブル開始アドレス : val_ram_s
SMSG4	&ITLS0	ITLS0 アドレス
SMSG5	&SMSG1	SMSG1 アドレス
SMSG6	&TDR01	TDR01 アドレス
SMSG7	2	固定値
SMSG8	0000H	SMS 内部変数
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	0000H	未使用
SMSG15	FFFFH	固定値 : FFFFH

表 5-6 シーケンサ・インストラクション・レジスタ 0-31

レジスタ呼称	設定値	備考
SMSI0	0400H	MOV [SMSG4+0], SMSG0
SMSI1	3180H	MOVW SMSG8, [SMSG1+0]
SMSI2	2680H	MOVW [SMSG6+0], SMSG8
SMSI3	7170H	ADDW SMSG1, SMSG7
SMSI4	7122H	CMPW SMSG1, SMSG2
SMSI5	8020H	BC \$2
SMSI6	2530H	MOVW [SMSG5+0], SMSG3
SMSI7	F000H	FINISH
SMSI8-31	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。

r01an5611_sms_led_dimming.scfg

r01an5611_sms_led_dimming.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 r01an5611_sms_led_dimming.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

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} ：32000kHz（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

タグ名	コンポーネント	内容
コンポーネント	Config_ITL000_ITL001	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：ITL000_ITL001 動作クロック：f _{SXP} クロックソース：f _{ITL0} /128 インターバル時間：125ms 割り込み設定：使用しない
	Config_TAU0_0	コンポーネント：PWM 出力 リソース：TAU0_0 動作クロック：CK00 クロックソース：f _{CLK} /2 ⁸ 周期設定：1ms 割り込み設定：使用しない (INTTM00) PWM スレーブ選択設定：チャンネル 1 スレーブ PWM ディューティ設定：0% 初期出力値：0 出力レベル：アクティブ・ハイ 割り込み設定：使用しない (INTTM01)
	Config_SMS	コンポーネント：SNOOZE モード・シーケンサ 起動トリガ：インターバル検出割り込み (INTITL)
	Config_PORT	コンポーネント：ポート ポート選択：PORT5 P53：入力

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_IT000_ITL001

サンプルコードのインターバル・タイマの初期設定を行います。

サンプルコードの SMS の起動にインターバル・タイマ割り込み (INTITL) を使用します。そのため、「割り込み設定」を「使用しない」に設定しています。「割り込み設定」を「使用する」に変更することも可能です。R_Config_SMS_Start 関数で INTITL はマスクされますので、HALT モード、SNOOZE モード中に INTITL が発生しても CPU が起動することはありません。HALT モード、SNOOZE モードから復帰後は、INTITL はマスクされた状態ですので、必要なときは INTITL のマスクを解除してください。

6.1.6 Config_TAU0_0

サンプルコードの TAU0 の初期設定を行います。

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 r01an5611_sms_led_dimming.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 を使用します。

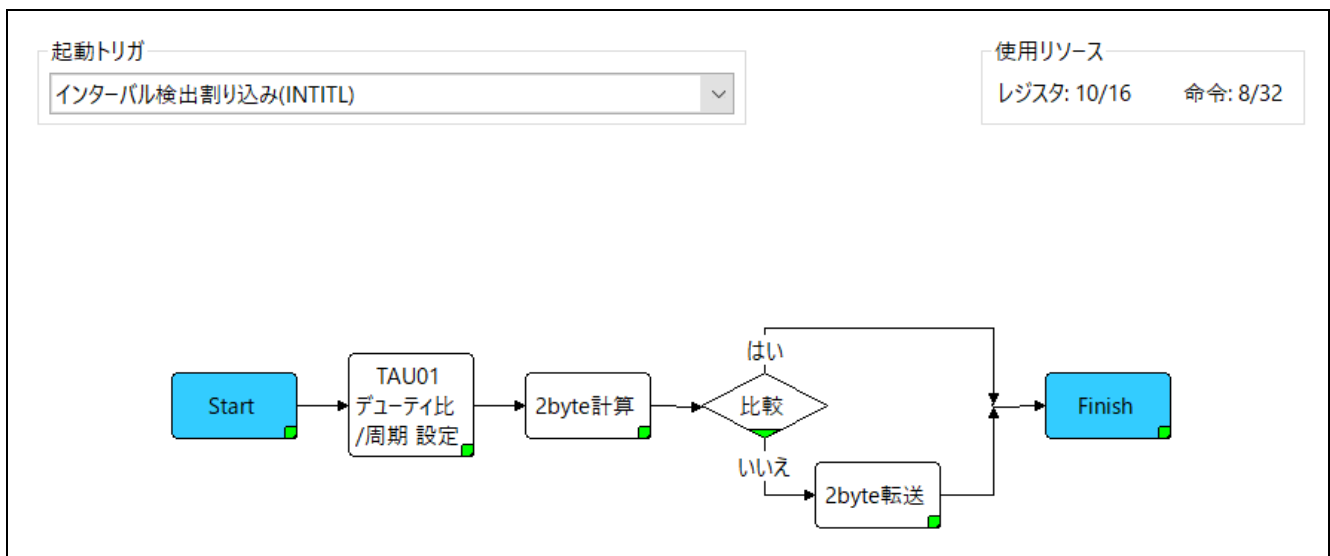
6.2 r01an5611_sms_led_dimming.sms

Config_SMS 単体のデータです。サンプルコードでは、SMS の起動にインターバル・タイマの割り込みを使用し、SMS の動作で TAU を使用します。別途インターバル・タイマと TAU 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、r01an5611_sms_led_dimming.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] → [ブラウズ] で「r01an5611_sms_led_dimming.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 5-2 SMS 処理」と同じです。

図 6-1 Config_SMS のフロー図

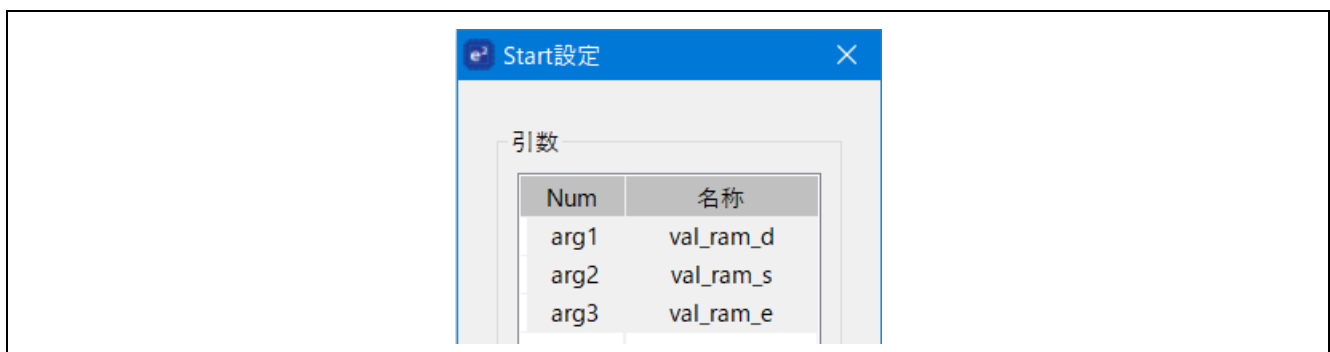


各ブロックの説明を以下に示します。

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値が val_ram_d (データテーブルポインタ)、val_ram_s (データテーブル開始アドレス)、val_ram_e (データテーブル終了アドレス) に設定されます。

図 6-2 Start 設定



6.2.2 TAU デューティ比／周期 設定

TAU のデューティ比を変更します。チャンネル指定された TAU01 の TDR レジスタに val_ram_d（データテーブルポインタ）の値を格納します。

サンプルコードでは、TAU00 と TAU01 を使って PWM 出力を行っているので、チャンネル指定に TAU01 を設定していますが、チャンネル指定は TAU00～TAU07 の中から設定することができます。

図 6-3 TAU01 デューティ比／周期 設定



6.2.3 2byte 計算

デューティ比が格納された val_ram_d に"2"を足し、結果を val_ram_d に格納します。これにより次のデューティ比が格納されたデータテーブルにポインタが変わります。

図 6-4 2byte 計算設定



6.2.4 比較

val_ram_d に格納されたデータテーブルポインタが val_ram_e (データテーブルの最終アドレス) に格納された閾値より小さいか比較します。val_ram_d が小さい場合は、HALT モードに移行します。val_ram_d が大きい場合は、次の 2byte 転送を行い、val_ram_d を更新します。

図 6-5 比較設定



6.2.5 2byte 転送

val_ram_s (データテーブル開始アドレス) の値を転送対象 val_ram_d に格納します。

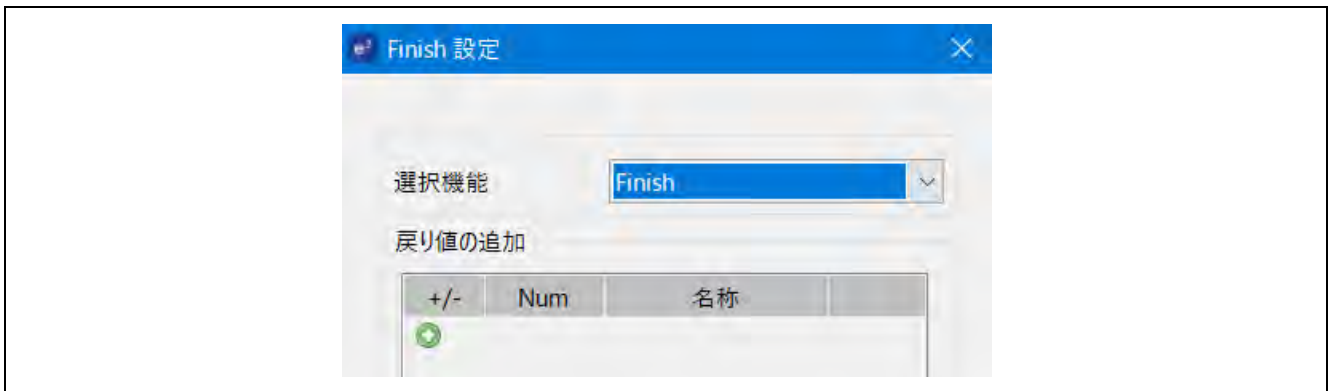
図 6-6 2byte 転送設定



6.2.6 Finish

HALT 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 6-7 Finish 設定



6.2.7 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_ram_d	SMS の開始関数で引数渡し	-	デューティ比を格納するデータテーブルのポインタを格納します。 R_Config_SMS_Start 関数で led_duty[0]のポインタが引数として設定されます。
val_ram_s	SMS の開始関数で引数渡し	-	デューティ比を格納するデータテーブルの開始アドレスを格納します。 R_Config_SMS_Start 関数で led_duty[0]のアドレスが引数として設定されます。
val_ram_e	SMS の開始関数で引数渡し	-	デューティ比を格納するデータテーブルの最終アドレスを格納します。 R_Config_SMS_Start 関数で &led_duty[0] + (BRIGHTNESS_RANK * 2) のアドレスが引数として設定されます。

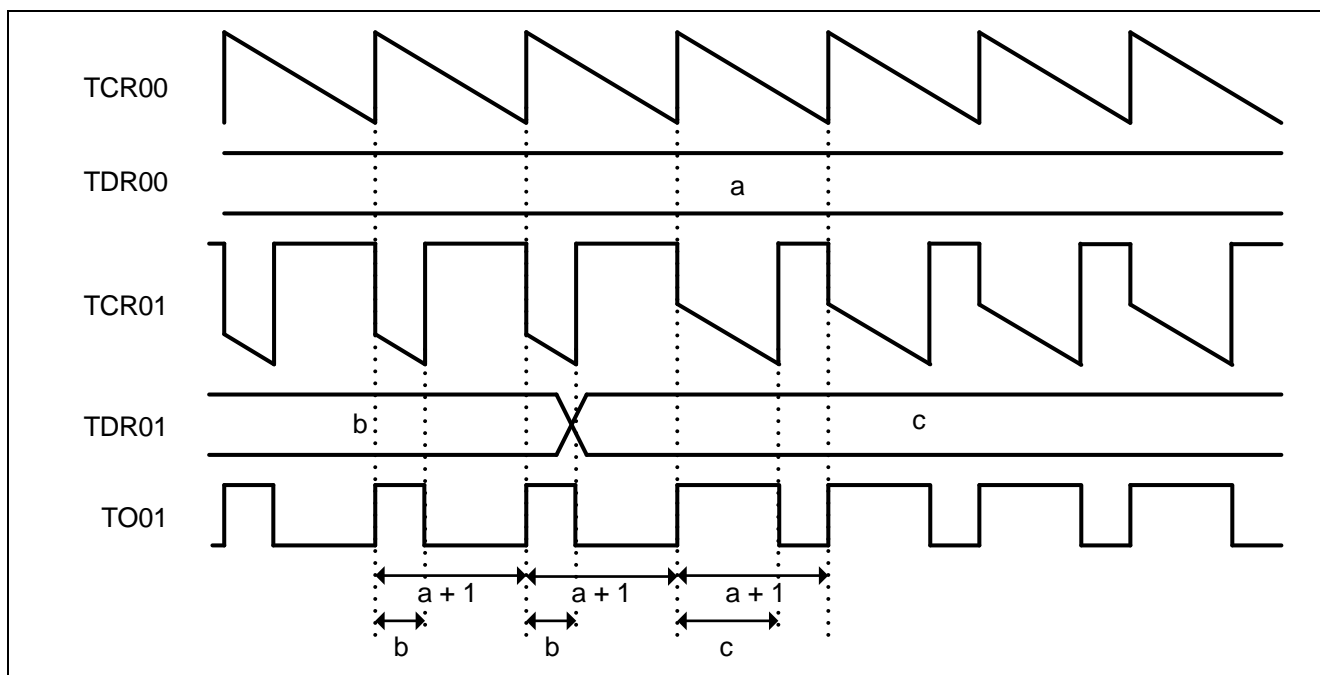
6.3 LED 点滅時間を変更する例

図 6-8 にサンプルコードの PWM 動作のタイミング例を示します。SMS 動作時に TDR01 の設定値を変更して、TO01 のデューティ比を変更します。Config_TAU0_0 にて初期設定値を決定した後は、TDR00 を変更していません。TDR01 に設定する値は、main.c で周期 (TDR00) を何分割 (BRIGHTNESS_RANK) するかで決めています。

サンプルコードの場合、以下のように 32 が設定されていますので、32 分割された値が設定されます。この値を変更することで、LED 点滅時間を変えることができます。

```
#define BRIGHTNESS_RANK    (32U)
```

図 6-8 PWM 動作のタイミング例



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。

また、SC でコードを再生成した場合は、R_Config_SMS_Start 関数内を下記赤枠の様に変更してください。

```
void R_Config_SMS_Start(uint16_t val_ram_d, uint16_t val_ram_s, uint16_t val_ram_e)
{
    /* Set the sms data from arguments */
    SMSG1 = val_ram_d;
    SMSG3 = val_ram_s;
    SMSG2 = val_ram_e;
    /* Disable related interrupts */
    ITLMK = 1U;
    TMMK01 = 1U;
    /* Start sms */
    //SMSEIF = 0U; /* clear INTSMSE interrupt flag */
    //SMSEMK = 0U; /* enable INTSMSE interrupt */
    SMSEMK = 1U; /* disable INTSMSE interrupt */
    SMSEIF = 1U; /* set INTSMSE interrupt flag */
    g_sms_wakeup_flag = 0U;
    ITLS0 = _00_INTITL_CLEAR;
    SMSSTART = 1U;
}
```

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 (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 ⇒ V8.05.00 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 17, 18	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 2 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 IAR : ior7f100g.h, ior7f100g_ext.h, r_cg_macrodriver.h, Config_SMS.h, Config_ITL000_ITL001.h, Config_TAU0_0.h ⇒ r_smc_entry.h
		17, 18	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 クロック : f _{SXL} ⇒ f _{SXP} 表 6-2 スマート・コンフィグレータの設定 動作クロック : f _{SXL} ⇒ f _{SXP}
		25	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	Jan.9.24	-	SMS 処理のフロー変更

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しないでください。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えば、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まれます。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。