

RL78/G23

ELCL を用いた WS2812B LED 制御

要旨

本アプリケーションノートでは、ロジック&イベント・リンク・コントローラ（ELCL）を利用してフルカラーシリアル LED の WS2812B 制御を実現する方法を説明します。ELCL と 3 線シリアル SPI を使うことで WS2812B の規格に対応した送信データに変換することが可能です。

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、そのマイコンの仕様に合わせて変更し、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. ハードウェア説明	7
3.1 ハードウェア構成例	7
3.2 使用端子一覧	7
4. ソフトウェア説明	8
4.1 動作概要	8
4.2 フォルダ構成	11
4.3 オプション・バイトの設定一覧	13
4.4 定数一覧	13
4.5 変数一覧	14
4.6 関数一覧	14
4.7 関数仕様	15
4.8 フローチャート	18
4.8.1 メイン処理	18
4.8.2 LED の色のデータをセット	22
4.8.3 LED 表示データを全て更新	23
4.8.4 LED 表示データの 1 ライン列シフト	23
4.8.5 LED 表示データの列を更新	24
4.8.6 LED 表示データの 1 ライン行シフト	24
4.8.7 LED 表示データの行を更新	25
4.8.8 LED 表示データの送信	26
4.8.9 CSI 送信完了割り込み	27
4.8.10 人感センサ検知割り込み(外部端子)	27
4.8.11 インターバル・タイマ割り込み	28
4.8.12 スイッチ入力検知割り込み(チャタリングフィルタ済み)	28
4.9 LED 制御データ フォーマットイメージ	29
5. スマート・コンフィグレータの設定	30
5.1 ELCL のコンポーネントの設定	30
5.2 r01an7320_elcl.scfg	31
5.2.1 クロック	34
5.2.2 システム	34
5.2.3 r_bsp	34
5.2.4 Config_TAU0_0	34
5.2.5 Config_TAU0_5	34
5.2.6 Config_INTC	35
5.2.7 Config_ITL000_ITL001_ITL012_ITL013	35
5.2.8 Config_ELCL	35
5.2.9 Config_CSI00	36
5.2.10 Config_PORT	36
6. サンプルコード	37

7. 参考ドキュメント 37

1. 仕様

本アプリケーションノートでは ELCL を利用して WS2812B（LED シート）制御を実現します。

WS2812B 通信仕様として、図 1-1 に示す 0 コード、図 1-2 に示す 1 コードと図 1-3 に示す Reset コードで LED を制御します。3 つのコードはタイマ・アレイ・ユニット 0（TAU0）を用いて波形を生成します。0 コード生成はチャンネル 1、1 コード生成はチャンネル 2、Reset コード生成はチャンネル 3 を使用します。TAU0 で生成したコード波形とシリアル・アレイ・ユニット（SAU）の簡易 SPI チャンネル 0（CSI00）で生成するパターン選択信号を ELCL に入力して LED を制御します。

本システムでは、スイッチ押下または、人感センサでの検知により LED シートを点灯させます。スイッチ押下モードでは、スイッチを押下により本モードに応じた点灯パターンで LED を点灯します。人感センサモードでは、人感センサで検知対象の検知により本モードに応じたパターンで LED を点灯します。また、人感センサモードで点灯中にスイッチ押下があれば、スイッチ押下モードの点灯に切り替わります。

図 1-1 0 code

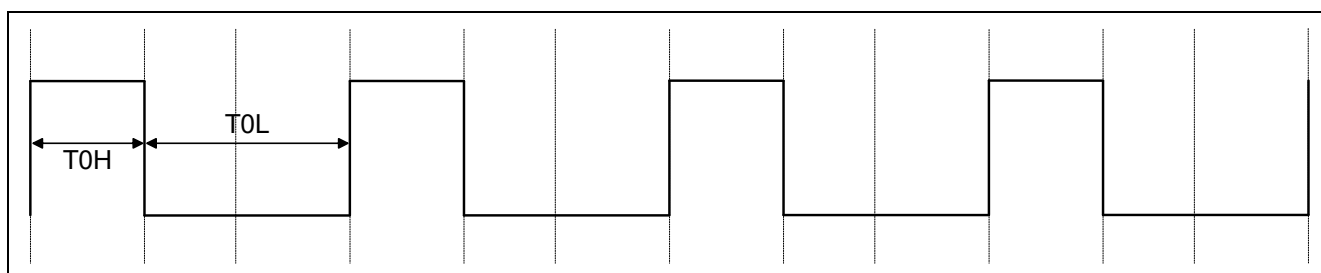


図 1-1 の T0H は 220 ns ～ 380 ns（本サンプルコードでは 342 ns に設定）、T0L は 580 ns ～ 1 μ s（本サンプルコードでは 908 ns に設定）です。

図 1-2 1 code

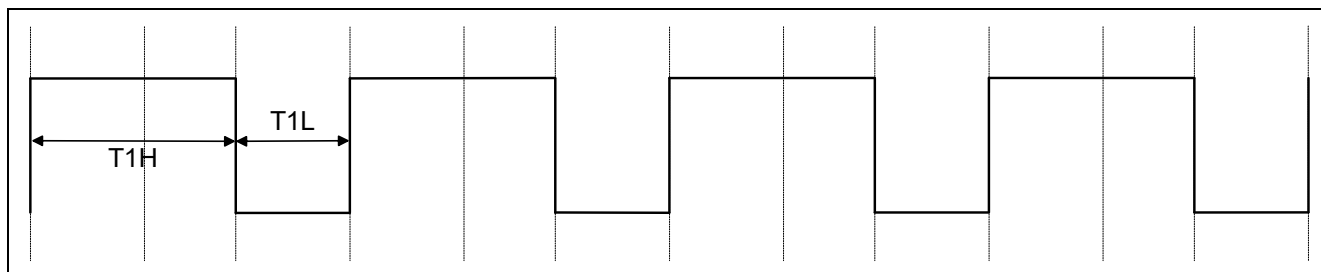


図 1-2 の T1H は 580 ns ～ 1 μ s（本サンプルコードでは 800 ns に設定）、T1L は 580 ns ～ 1 μ s（本サンプルコードでは 450 ns に設定）です。

図 1-3 Reset code



図 1-3 の Reset code は 280 μ s 以上です。

図 1-4 に ELCL を使用して WS2812B 通信の LED 制御を実現するシステム構成を示します。

図 1-4 システム構成

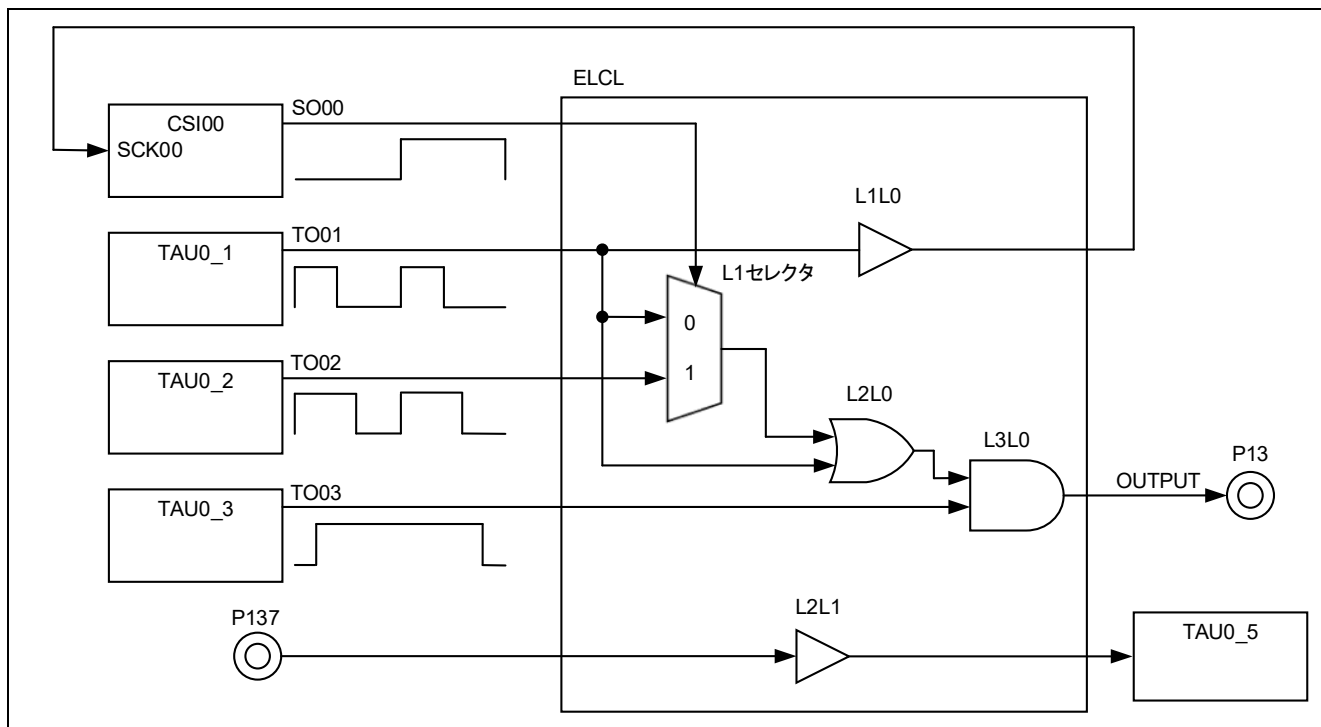
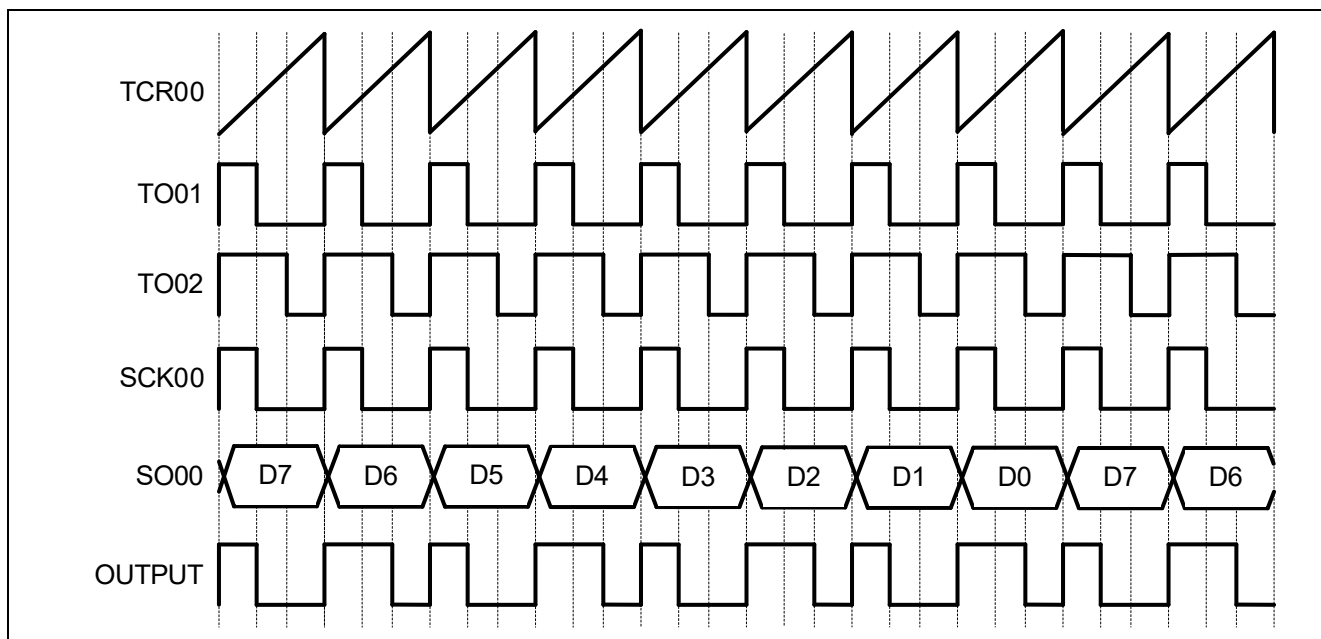


図 1-5 にタイミングチャートを示します。

- (1) TAU0 チャンネル 0 をマスタ・チャンネル、チャンネル 1 とチャンネル 2 をスレーブ・チャンネルとし、多重 PWM 出力機能に設定します
- (2) TO01 は ELCL を介して SCI の SCK00 に入力し、SO00 から LED 点灯パターンに応じたデータを出力します
- (3) SO00 の値に応じて OUTPUT が出力されます

図 1-5 タイミングチャート



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

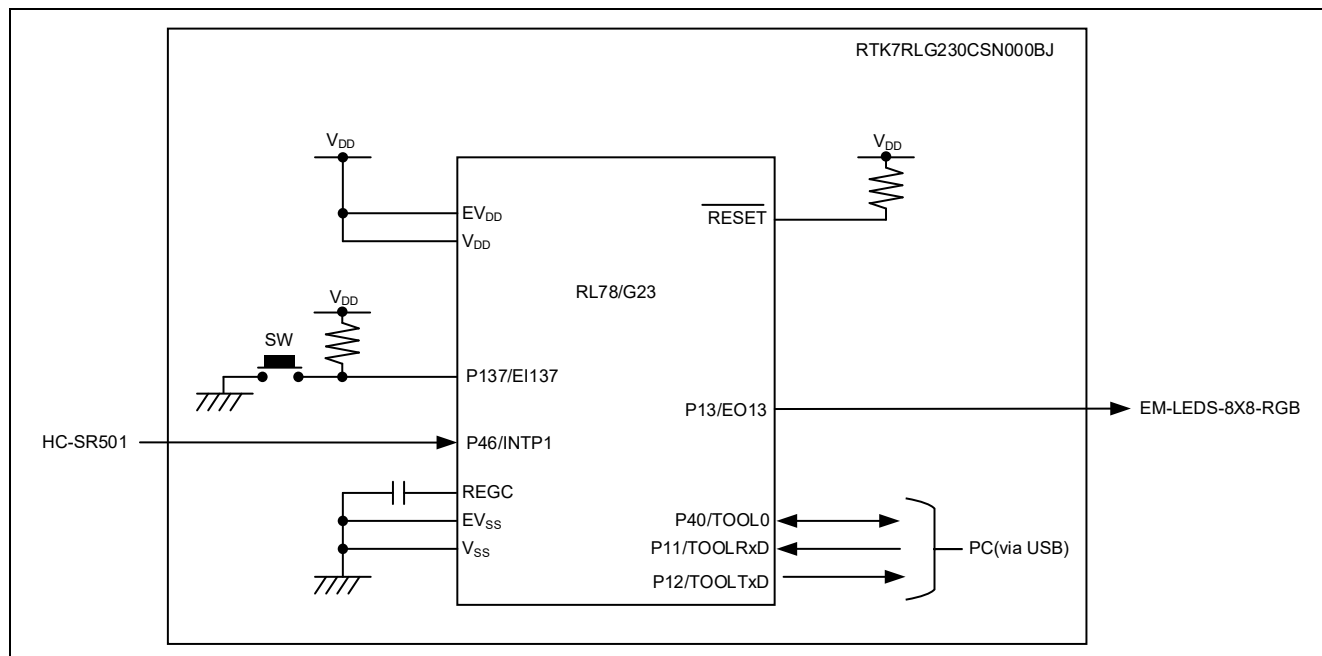
項目	内容
使用マイコン	RL78/G23 (R7F100GSN)
動作周波数	- 高速オンチップ・オシレータ・クロック : 32MHz - CPU/周辺ハードウェア・クロック : 32MHz
動作電圧	5.0V
統合開発環境 (CS+)	ルネサスエレクトロニクス製 CS+ for CC 8.13.00
C コンパイラ (CS+)	ルネサスエレクトロニクス製 CC-RL V1.15
統合開発環境 (e2 studio)	ルネサスエレクトロニクス製 e ² studio 2025-04 (25.04.0)
C コンパイラ (e2 studio)	ルネサスエレクトロニクス製 CC-RL V1.15
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 V 5.20.1
スマート・コンフィグレータ	V1.13.0
ボードサポートパッケージ (r_bsp)	V.1.90
エミュレータ	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 (RTK7RLG230CSN000BJ)

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
P13/EO13	出力	ELCL の出力信号（WS2812B 制御用）
P137/EI137	入力	ELCL の入力信号（スイッチ入力）
P46/INTP1	入力	人感センサ入力
P16/TO01、P15/TO02、P14/TO03	出力	モニタ用
P17/SO00	出力	モニタ用

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 動作概要

本サンプルコードでは、TAU0 から出力される PWM 出力と CSI00 のデータ出力に対して下記(図 4-1 参照) ELCL モジュールを用いて LED シート(WS2812B)の制御信号を生成します。生成した信号は P13 に出力します。

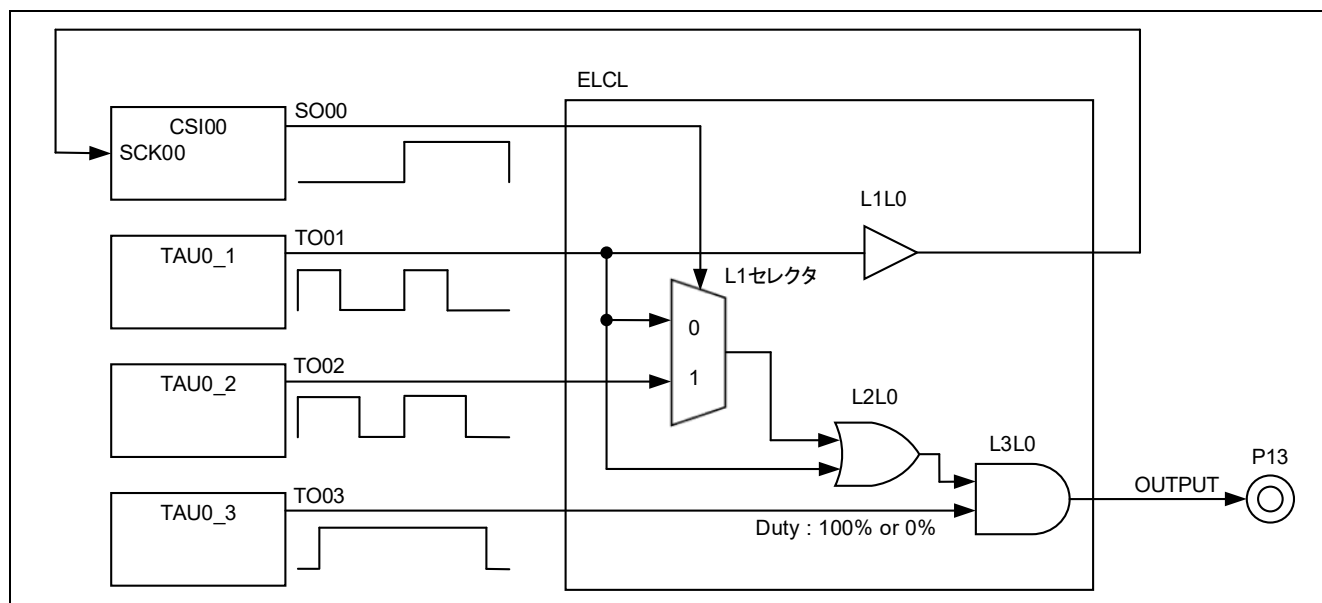
図 4-1 に LED シート(WS2812B)の制御信号生成システム構成を示します。

TAU0 からの 3 つの PWM 出力 (TO01、TO02、TO03) を、ELCL の入力信号とします。また CSI00 の SO00 も ELCL の入力信号とします。TO01 を 0 コード用の PWM、TO02 を 1 コード用の PWM として設定します。また、TO01 は、Through でリンク先として CSI00 のクロック入力に設定します。TO01 と TO02 を ELCL のセクタの入力とし、SO00 をセクタの選択制御信号としてリンク設定します。

上記セクタの出力信号と、TO01 を ELCL の L2 の入力信号とし、OR で結合することでセクタ切り替え時のグリッジノイズリスクへの対策とします。

TO03 を Reset コード用の PWM(デューティ : 100% or 0%)として設定します。上記 OR の出力信号と、TO03 を ELCL の L3 の入力信号とし、AND で結合し、TO03 の PWM デューティを 0%に設定することで、Reset コードを出力できるようにします。AND の出力信号を P13 とリンクさせることで LED シート (WS2812B)の制御信号を P13 から出力します。

図 4-1 LED シート(WS2812B)の制御信号生成システム構成



また、マイコンボード上のスイッチ入力に対しチャタリング除去機能を実装しています。P137 入力に対して、Through リンクで TAU0 の TI5 へ接続します。TAU0 のディレイ・カウンタ機能を用いて、P137 入力のチャタリングを除去します。チャタリング除去後のスイッチ入力に対応する処理は、タイマ割り込み (INTTM05)ベクタで行います。

図 4-2 チャタリング除去機能システム構成

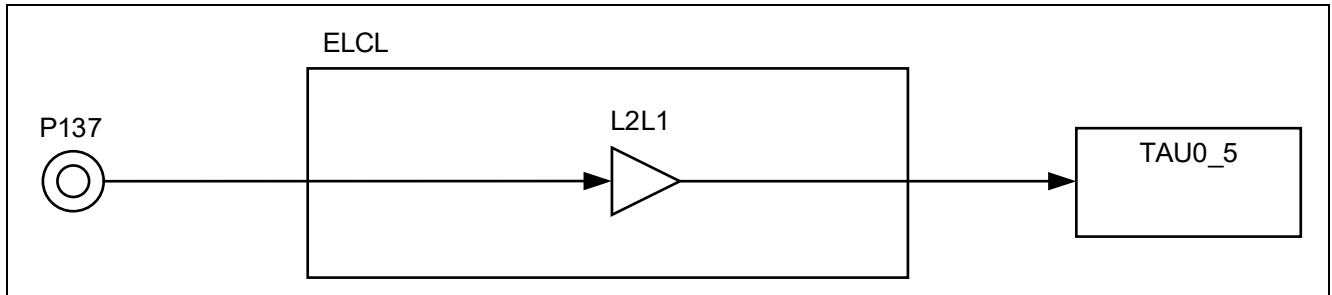
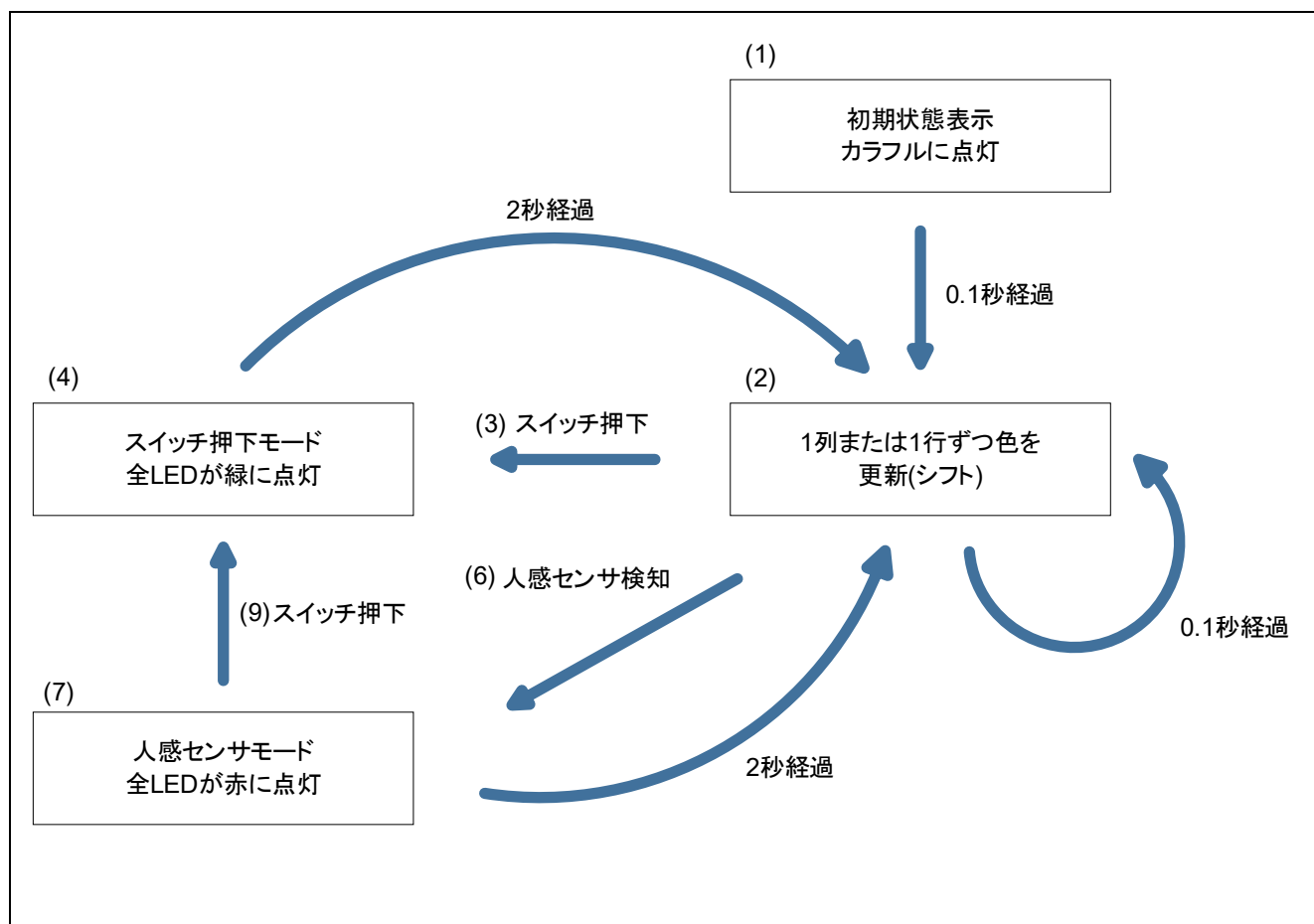


図 4-3 に動作概要を示します。

- (1) 動作を開始すると LED シートが初期表示状態に点灯します
- (2) 0.1 秒毎に 1 行または 1 列ずつ色のパターンを変更します
- (3) スイッチを押下します
- (4) (3)により、スイッチ押下モードになり 2 秒間全面が緑に点灯します
- (5) (2)と同様に 0.1 秒毎に 1 行または 1 列ずつ色のパターンを変更します
- (6) 人感センサを反応させます
- (7) (6)により、人感センサモードになり 2 秒間全面が赤に点灯します
- (8) (2)と同様に 0.1 秒毎に 1 行または 1 列ずつ色のパターンを変更します
- (9) 人感センサモード時にスイッチ押下を検出した場合、スイッチ押下モードに切り替わります

図 4-3 動作概要



4.2 フォルダ構成

表 4-1 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 4-1 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
¥r01an7320_elcl<DIR> ^{注2}	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	
main.h	ソースコードヘッダファイル	
¥smc_gen<DIR>	スマート・コンフィグレータ生成フォルダ	√
¥Config_CSI00<DIR>	CSI00 用プログラム格納フォルダ	√
Config_CSI00.c	CSI00 用ソースファイル	√
Config_CSI00.h	CSI00 用ヘッダファイル	√
Config_CSI00_user.c	CSI00 用割り込みソースファイル	√ ^{注1}
¥Config_ELCL<DIR>	ELCL 用プログラム格納フォルダ	√
Config_ELCL.c	ELCL 用ソースファイル	√
Config_ELCL.h	ELCL 用ヘッダファイル	√
Config_ELCL_user.c	ELCL 用割り込みソースファイル	√ ^{注1}
¥Config_INTC<DIR>	INTC 用プログラム格納フォルダ	√
Config_INTC.c	INTC 用ソースファイル	√
Config_INTC.h	INTC 用ヘッダファイル	√
Config_INTC_user.c	INTC 用割り込みソースファイル	√ ^{注1}
¥Config_ITL000_ITL001_ITL012_ITL013<DIR>	ITL000_ITL001_ITL012_ITL013 用プログラム格納フォルダ	√
Config_ITL000_ITL001_ITL012_ITL013.c	ITL000_ITL001_ITL012_ITL013 用ソースファイル	√
Config_ITL000_ITL001_ITL012_ITL013.h	ITL000_ITL001_ITL012_ITL013 用ヘッダファイル	√
Config_ITL000_ITL001_ITL012_ITL013_user.c	ITL000_ITL001_ITL012_ITL013 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_TAU0_0<DIR>	TAU00 用プログラム格納フォルダ	√
Config_TAU0_0.c	TAU00 用ソースファイル	√
Config_TAU0_0.h	TAU00 用ヘッダファイル	√
Config_TAU0_0_user.c	TAU00 用割り込みソースファイル	√ ^{注1}
¥Config_TAU0_5<DIR>	TAU05 用プログラム格納フォルダ	√
Config_TAU0_5.c	TAU05 用ソースファイル	√
Config_TAU0_5.h	TAU05 用ヘッダファイル	√
Config_TAU0_5_user.c	TAU05 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 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. IAR 版のサンプルコードは buildinfo.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。

4.3 オプション・バイトの設定一覧

表 4-2 オプション・バイト設定にオプション・バイト設定を示します。

表 4-2 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	0011 1010B (3AH)	LVD0 オフ
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 0100B (84H)	オンチップ・デバッグ動作許可

4.4 定数一覧

表 4-3 にサンプルコードで使用する定数を示します。

表 4-3 サンプルコードで使用する定数

定数名	設定値	内容
LED_W	8	LED シートの幅
LED_H	8	LED シートの高さ
RGB	3	赤、緑、青の 3 色
DISP_NORMAL_TIME	1	通常時の LED 更新時間
DISP_EVENT_TIME	20	イベント発生時の LED 更新時間

4.5 変数一覧

表 4-4 にグローバル変数を示します。

表 4-4 グローバル変数

型	変数名	内容	使用関数
uint8_t	g_buf_csi	CSI 送信用バッファ(配列)	main、shift_row、led_data_send
uint8_t	g_flag_send_end	送信完了フラグ	main、 r_Config_CSI00_callback_sendend、 led_data_send
uint8_t	g_flag_constperiod	一定期間経過フラグ	main、 R_Config_ITL000_ITL001_ITL012_ITL 013_Callback_Shared_Interrupt
uint8_t	g_flag_switch	スイッチ押下フラグ	main、 r_Config_TAU0_5_interrupt
uint8_t	g_flag_sensor	人感センサ反応フラグ	main、 r_Config_INTC_intp1_interrupt
uint8_t	g_flag_send_timing	表示更新フラグ	main
uint16_t	g_cnt_time	LED 表示時間カウンタ	main
uint16_t	g_time_disp	LED 表示時間	main
uint8_t	g_row_column	更新対象(行または列)イン ジケータ	main

4.6 関数一覧

表 4-5 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 4-5 関数一覧

関数名	概要
main()	メイン処理
set_color()	LED の色のデータをセット
change_all()	LED 表示データを全て更新
shift_column()	LED 表示データの 1 ライン列シフト
change_column()	LED 表示データの列を更新
shift_row()	LED 表示データの 1 ライン行シフト
change_row()	LED 表示データの行を更新
led_data_send()	LED 表示データの送信
r_Config_CSI00_callback_sendend()	CSI 送信完了割り込み
r_Config_INTC_intp1_interrupt()	人感センサ検知割り込み(外部端子)
R_Config_ITL000_ITL001_ITL012_ITL 013_Callback_Shared_Interrupt()	インターバル・タイマ割り込み
r_Config_TAU0_5_interrupt()	スイッチ入力検知割り込み(チャタリングフィルタ済み)

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	main.h、r_smc_entry.h
宣言	int main (void);
説明	割り込みの設定を行います 外部割り込み、インターバル・タイマ、CSI00、TAU0 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] set_color

概要	LED の色のデータをセット
ヘッダ	main.h、r_smc_entry.h
宣言	void set_color(E_COLOR e_color, uint8_t *g_val, uint8_t *r_val, uint8_t *b_val)
説明	指定した変数に指定した色データを設定します
引数	e_color, g_val, r_val, b_val
リターン値	なし
備考	なし

[関数名] change_all

概要	LED 表示データを全て更新
ヘッダ	main.h、r_smc_entry.h
宣言	void change_all(uint8_t buf[][LED_W][RGB], E_COLOR e_color);
説明	LED 表示データを指定した色ですべて更新
引数	buf[][LED_W][RGB], e_color)
リターン値	なし
備考	なし

[関数名] shift_column

概要	LED 表示データの 1 ライン列シフト
ヘッダ	main.h、r_smc_entry.h
宣言	void shift_column(uint8_t buf[][LED_W][RGB])
説明	LED 表示データの 1 ライン列シフトします
引数	buf[][LED_W][RGB]
リターン値	なし
備考	なし

[関数名] change_column

概要	LED 表示データの列を更新
ヘッダ	main.h、r_smc_entry.h
宣言	void change_column(uint8_t buf[][LED_W][RGB], uint8_t column, E_COLOR e_color);
説明	LED 表示データの指定した列を指定した色で更新します
引数	なし
リターン値	なし
備考	なし

[関数名] shift_row

概要	LED 表示データの 1 ライン行シフト
ヘッダ	main.h、r_smc_entry.h
宣言	void shift_row(uint8_t buf[][LED_W][RGB])
説明	LED 表示データの 1 ライン行シフトします
引数	buf[][LED_W][RGB]
リターン値	なし
備考	なし

[関数名] change_row

概要	LED 表示データの行を更新
ヘッダ	main.h、r_smc_entry.h
宣言	void change_row(uint8_t buf[][LED_W][RGB], uint8_t row, E_COLOR e_color);
説明	LED 表示データの指定した行を指定した色で更新します
引数	buf[][LED_W][RGB], row, e_color
リターン値	なし
備考	なし

[関数名] led_data_send

概要	LED 表示データの送信
ヘッダ	main.h、r_smc_entry.h
宣言	void led_data_send(void);
説明	LED 表示データを送信します
引数	なし
リターン値	なし
備考	なし

[関数名] r_Config_CSI00_callback_sendend

概要	CSI 送信完了割り込み
ヘッダ	r_cg_macrodriver.h
宣言	static void r_Config_CSI00_callback_sendend(void);
説明	送信完了フラグを"1"セットします
引数	なし
リターン値	なし
備考	なし

[関数名] r_Config_INTC_intp1_interrupt

概要	人感センサ検知割り込み(外部端子)
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_INTC.h
宣言	static void __near r_Config_INTC_intp1_interrupt(void);
説明	人感センサ検知によって、人感センサ反応フラグを"1"セットします
引数	なし
リターン値	なし
備考	なし

[関数名] R_Config_ITL000_ITL001_ITL012_ITL013_Callback_Shared_Interrupt

概要	インターバル・タイマ割り込み
ヘッダ	Config_ITL000_ITL001_ITL012_ITL013.h
宣言	void R_Config_ITL000_ITL001_ITL012_ITL013_Callback_Shared_Interrupt(void);
説明	一定期間経過フラグを"1"セットします
引数	なし
リターン値	なし
備考	なし

[関数名] r_Config_TAU0_5_interrupt

概要	スイッチ入力検知割り込み(チャタリングフィルタ済み)
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_TAU0_5.h
宣言	static void __near r_Config_TAU0_5_interrupt(void);
説明	スイッチ押下によって、スイッチ押下フラグを"1"セットします
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 4-4、図 4-5、図 4-6 および図 4-7 にメイン処理のフローチャートを示します。

図 4-4 メイン処理(1/4)

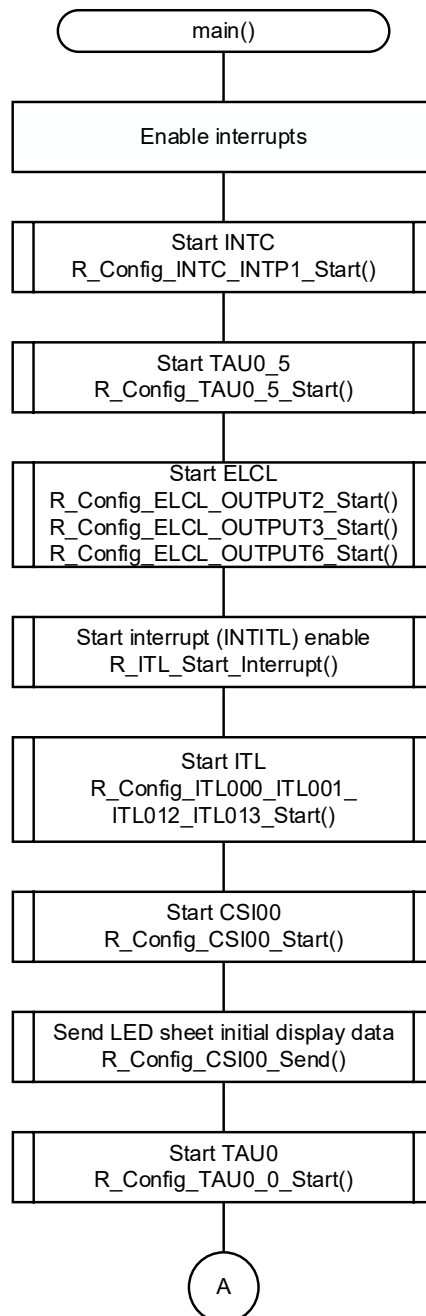


図 4-4 メイン処理(2/4)

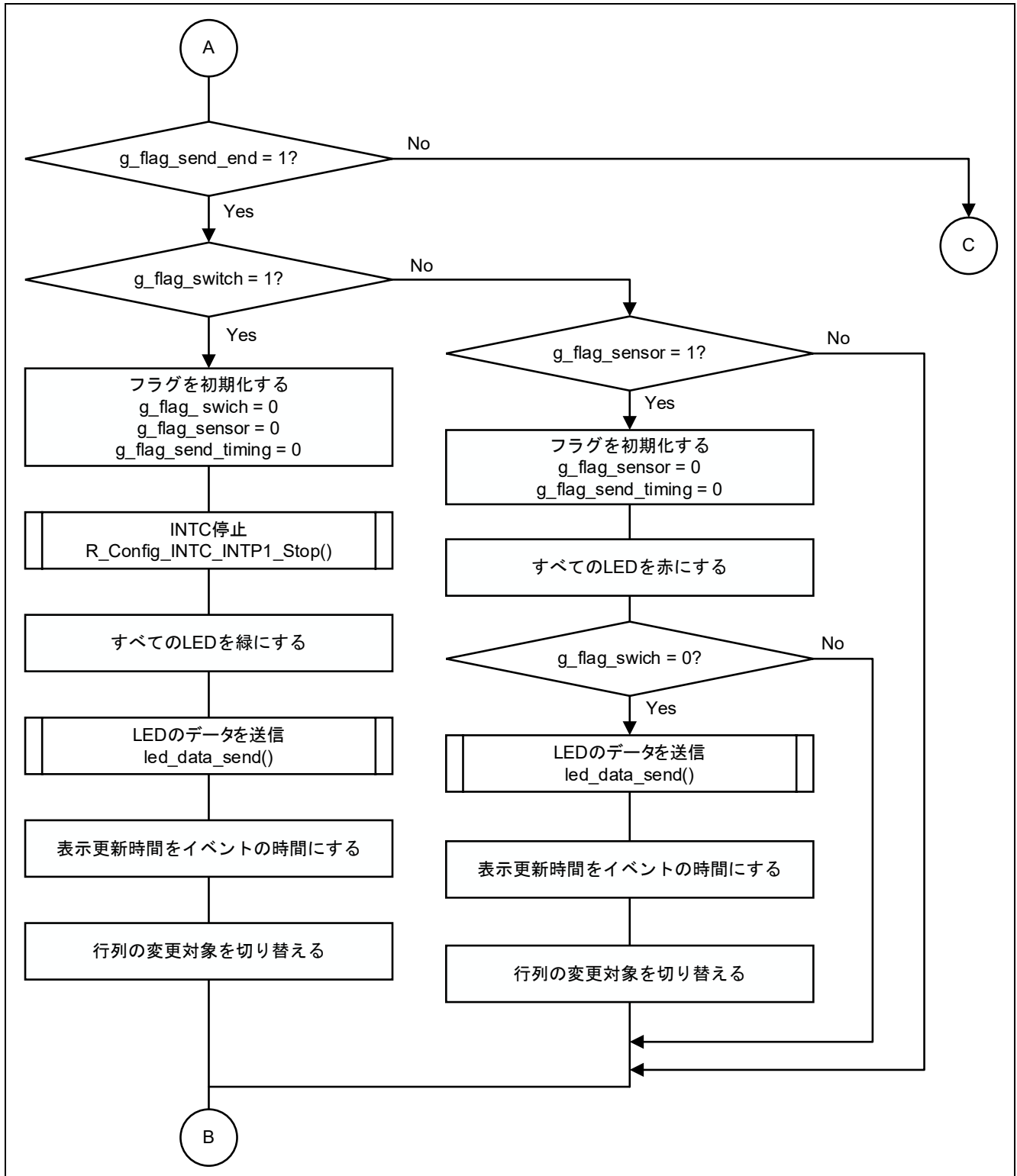


図 4-5 メイン処理(3/4)

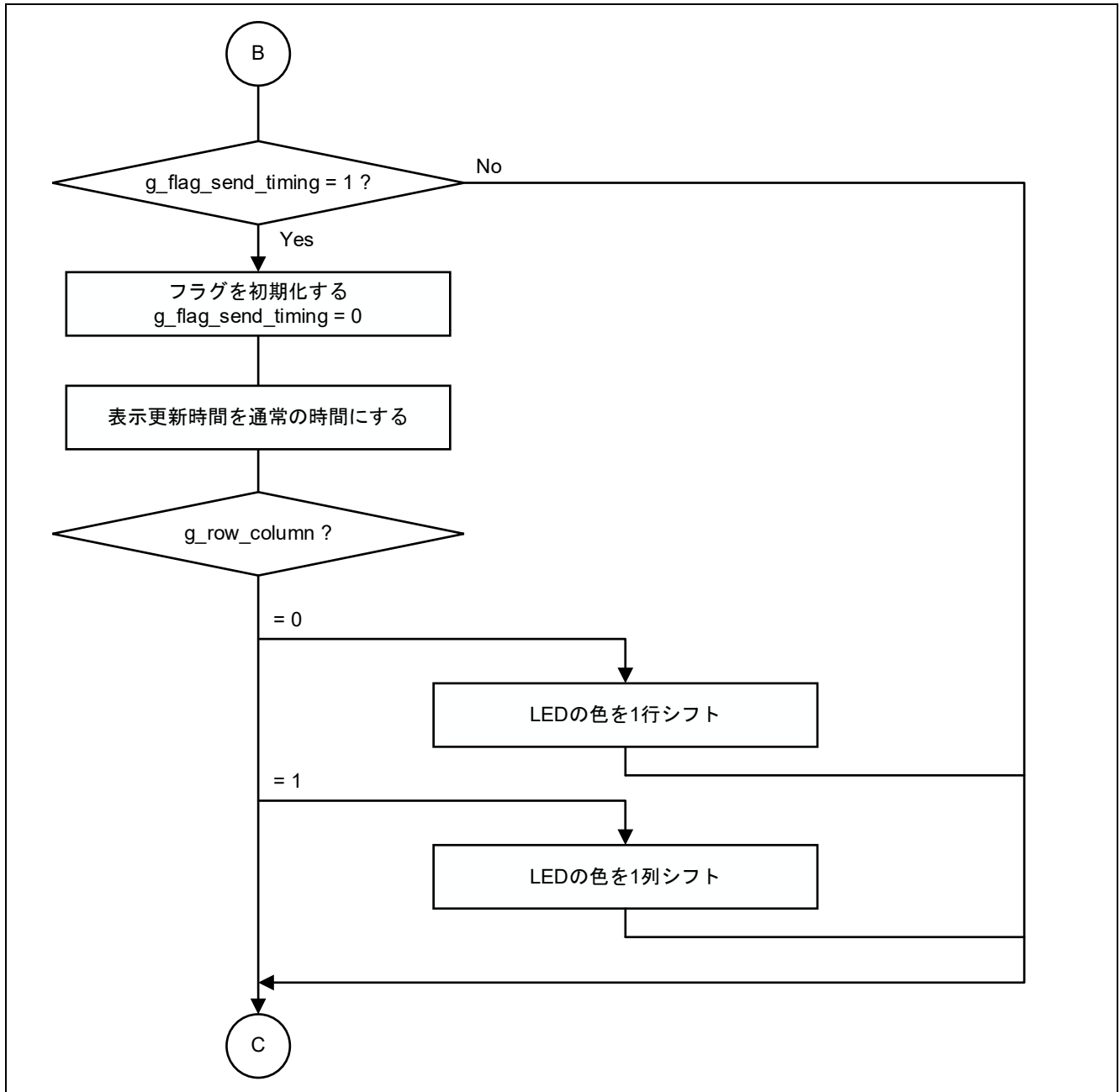
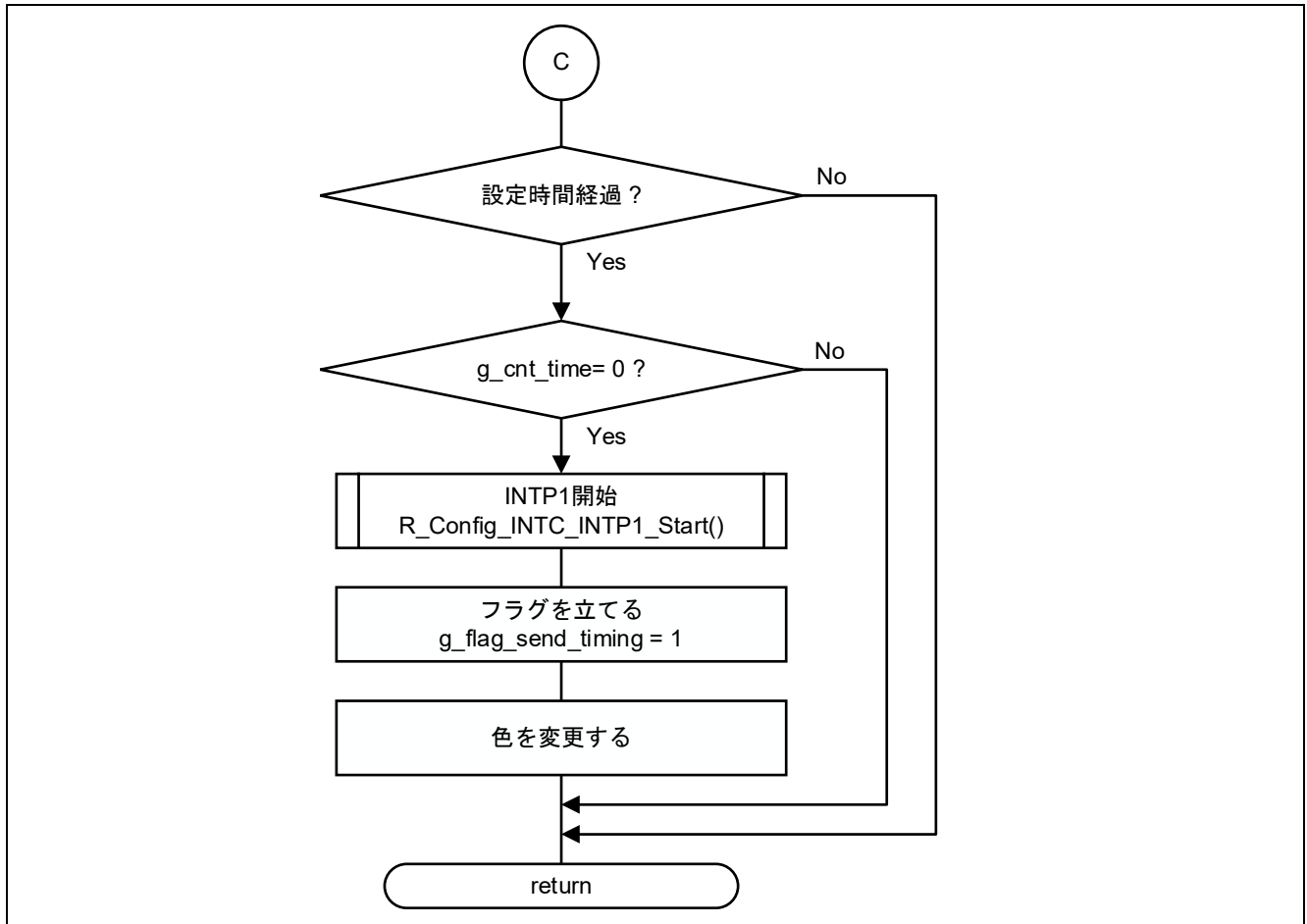


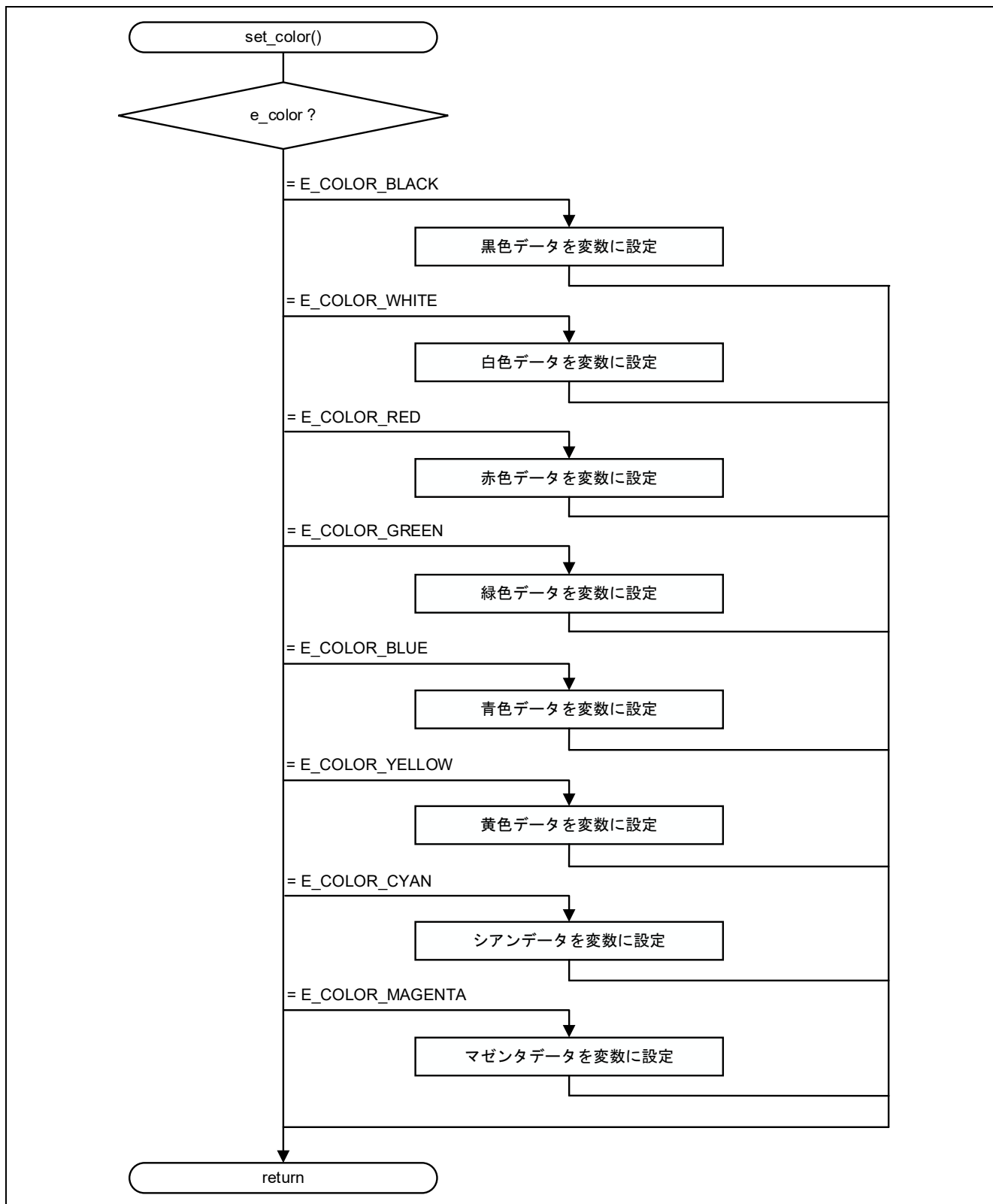
図 4-6 メイン処理(4/4)



4.8.2 LED の色のデータをセット

図 4-8 に LED の色のデータをセットのフローチャートを示します。

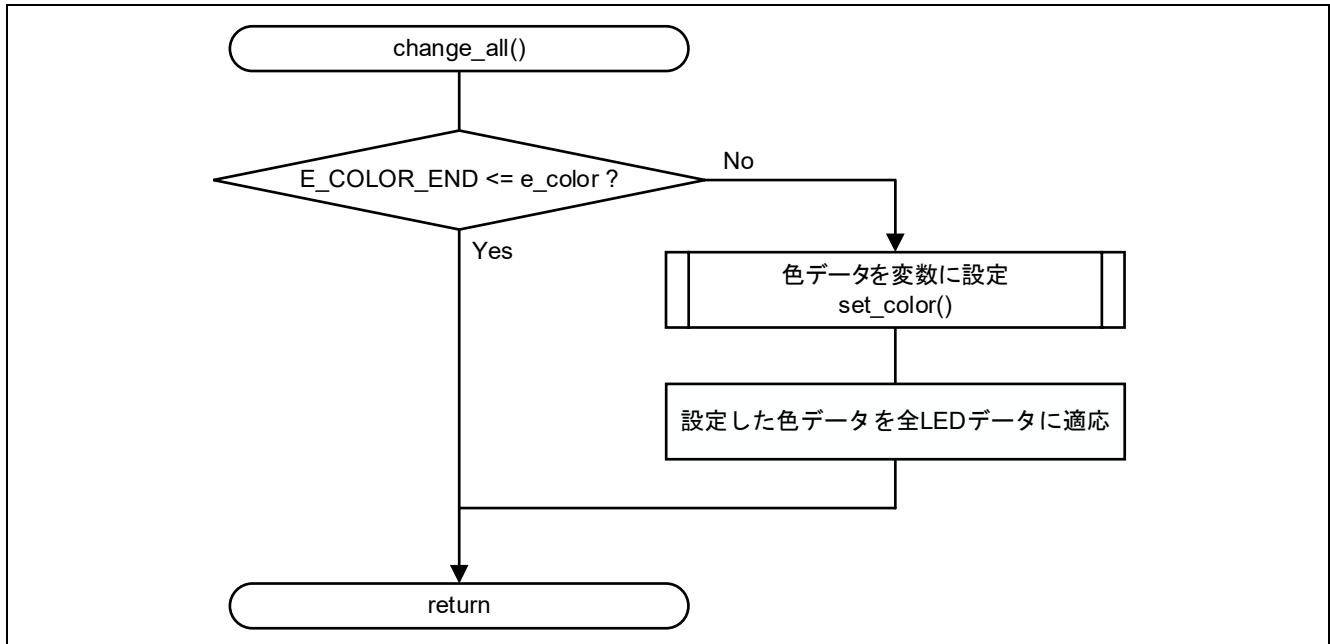
図 4-7 LED の色のデータをセット



4.8.3 LED 表示データを全て更新

図 4-9 に LED 表示データを全て更新のフローチャートを示します。

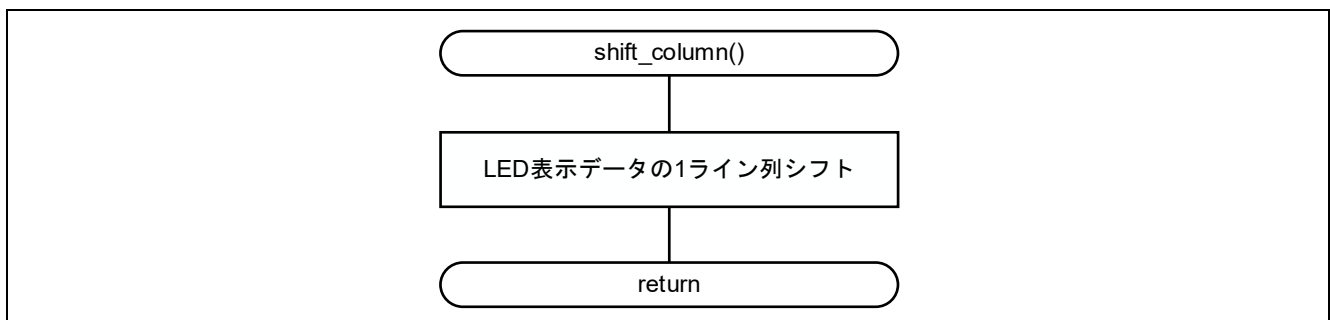
図 4-8 LED 表示データを全て更新



4.8.4 LED 表示データの 1 ライン列シフト

図 4-10 に LED 表示データの 1 ライン列シフトのフローチャートを示します。

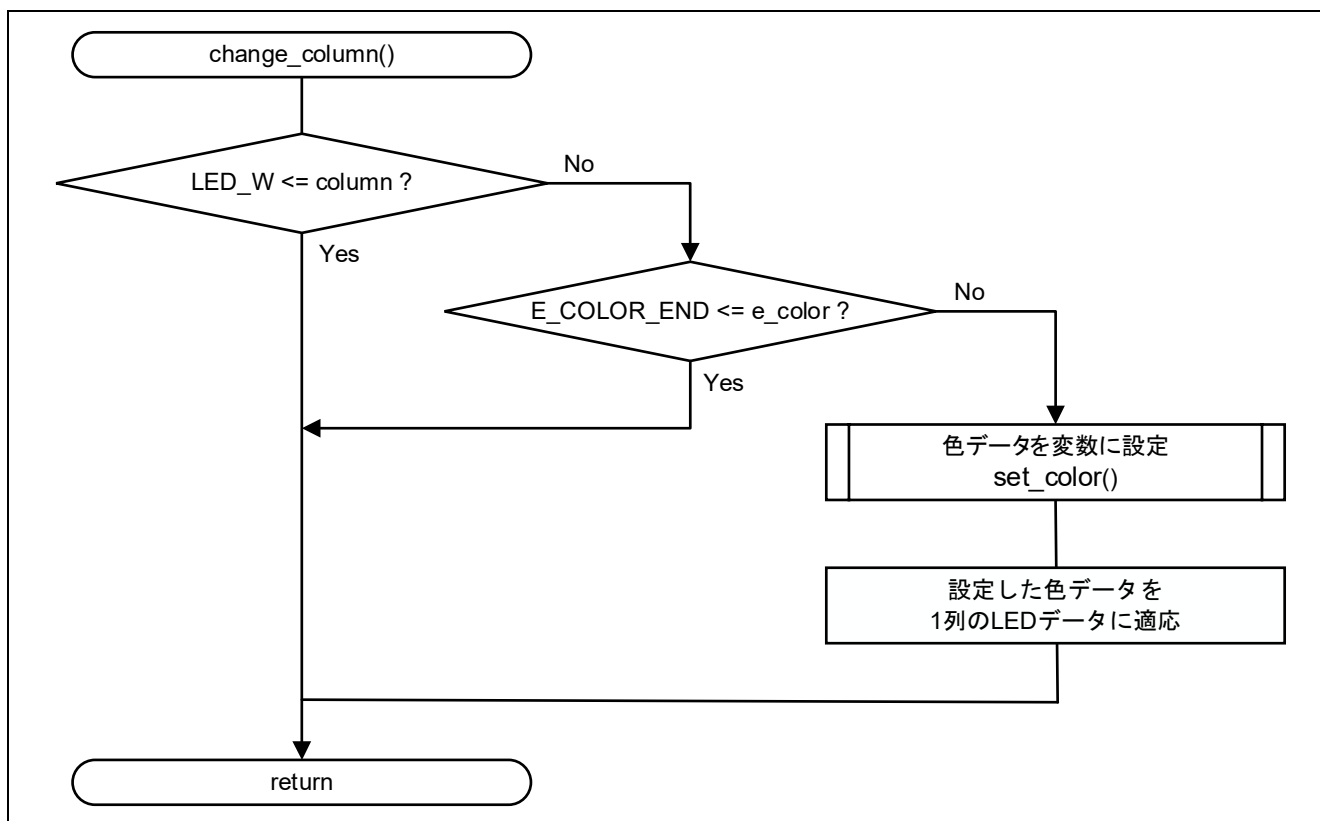
図 4-9 LED 表示データの 1 ライン列シフト



4.8.5 LED 表示データの列を更新

図 4-11 に LED 表示データの列を更新のフローチャートを示します。

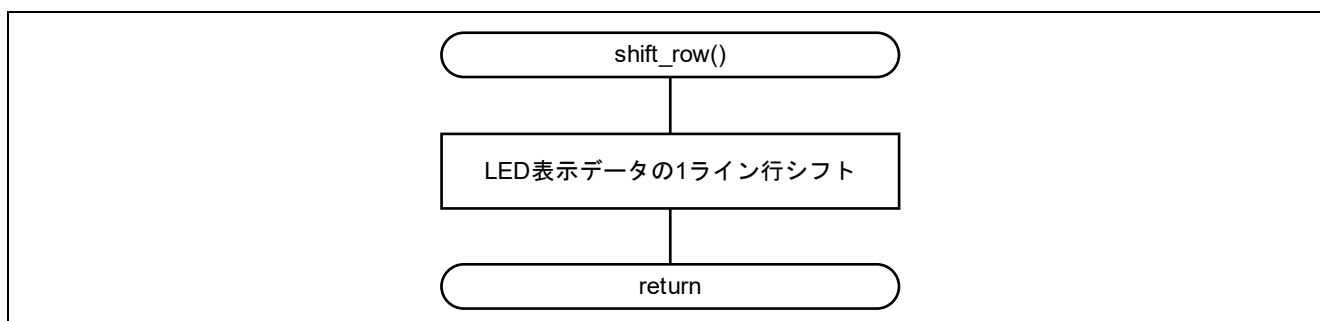
図 4-10 LED 表示データの列を更新



4.8.6 LED 表示データの 1 ライン行シフト

図 4-12 に LED 表示データの 1 ライン行シフトのフローチャートを示します。

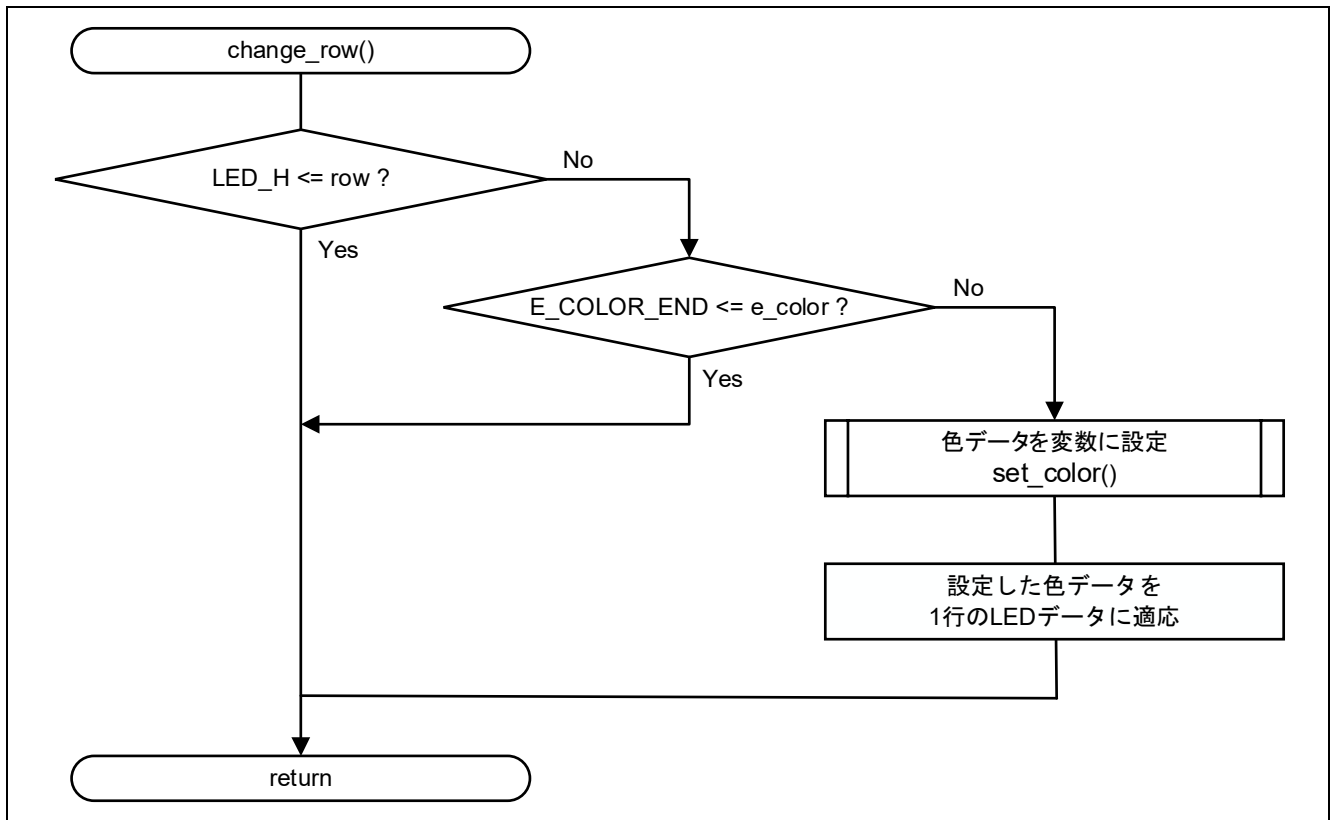
図 4-11 LED 表示データの 1 ライン行シフト



4.8.7 LED 表示データの行を更新

図 4-13 に LED 表示データの行を更新のフローチャートを示します。

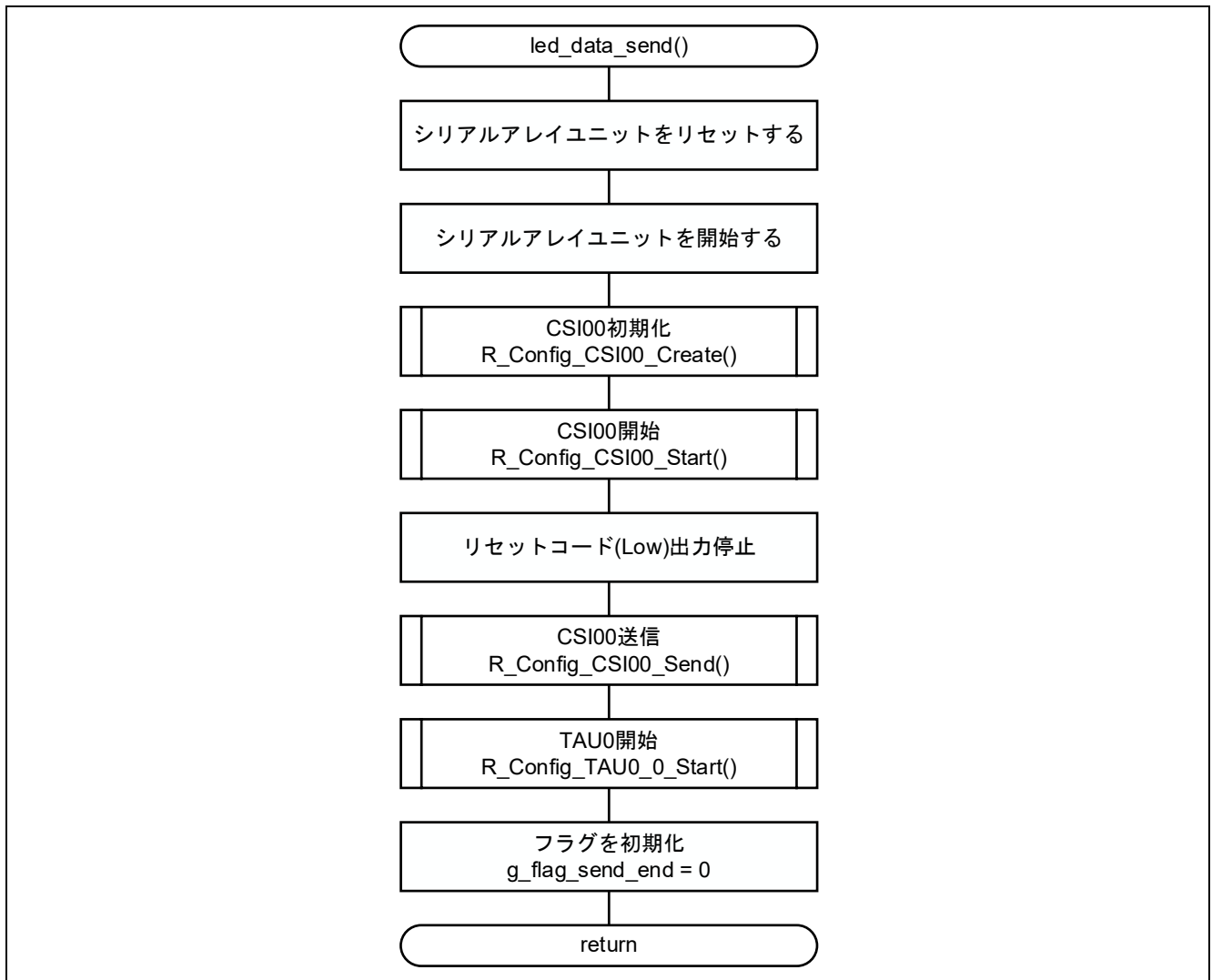
図 4-12 LED 表示データの行を更新



4.8.8 LED 表示データの送信

図 4-14 に LED 表示データの送信のフローチャートを示します。

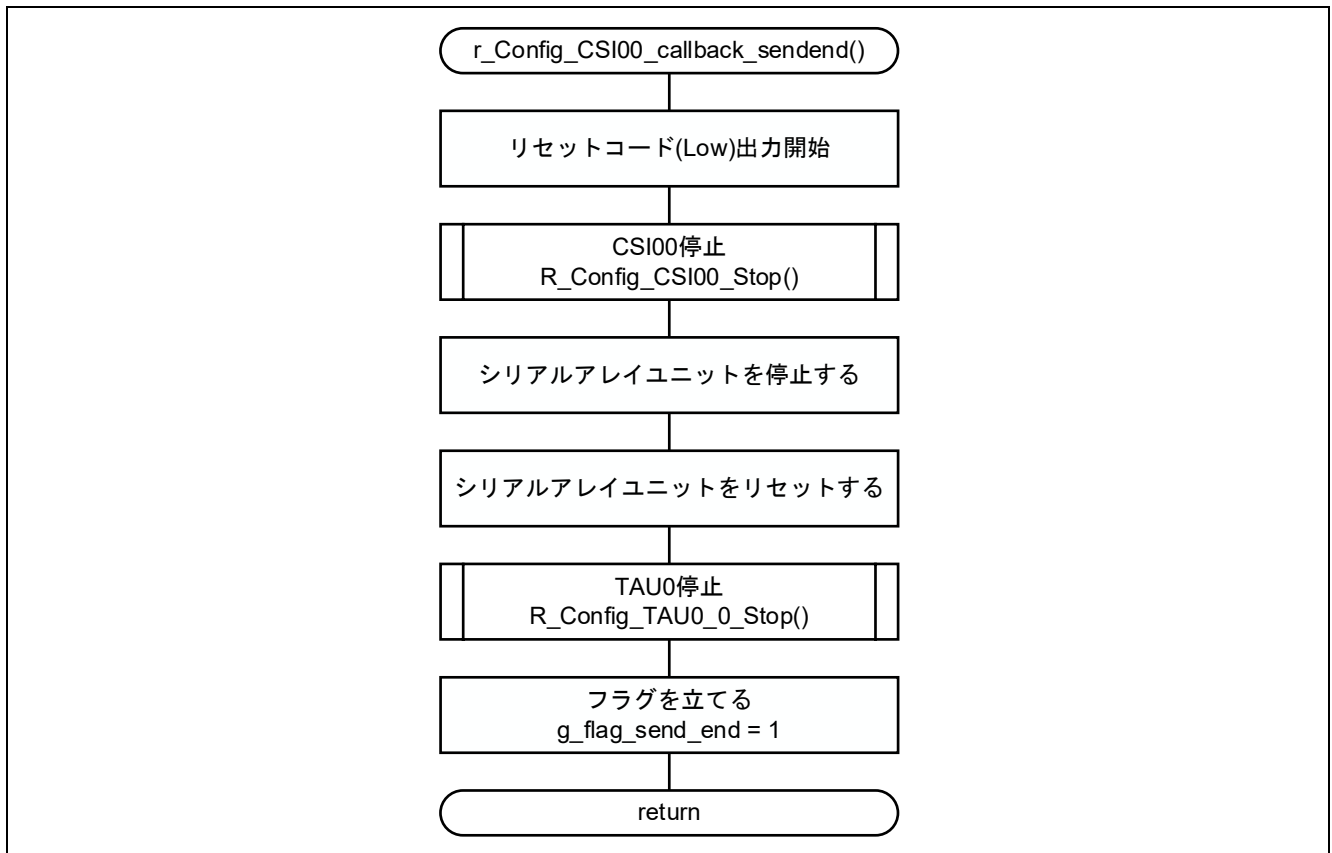
図 4-13 LED 表示データの送信



4.8.9 CSI 送信完了割り込み

図 4-14 に CSI 送信完了割り込みのフローチャートを示します。

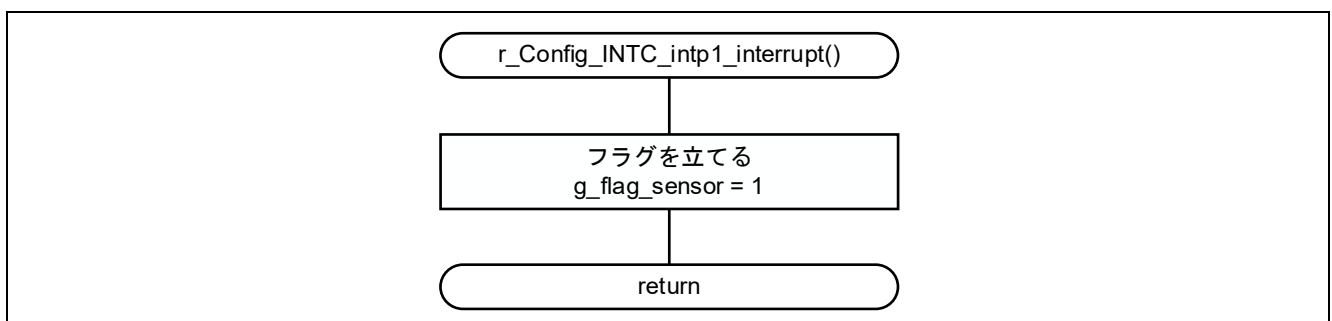
図 4-14 CSI 送信完了割り込み



4.8.10 人感センサ検知割り込み(外部端子)

図 4-15 に人感センサ検知割り込み(外部端子)のフローチャートを示します。

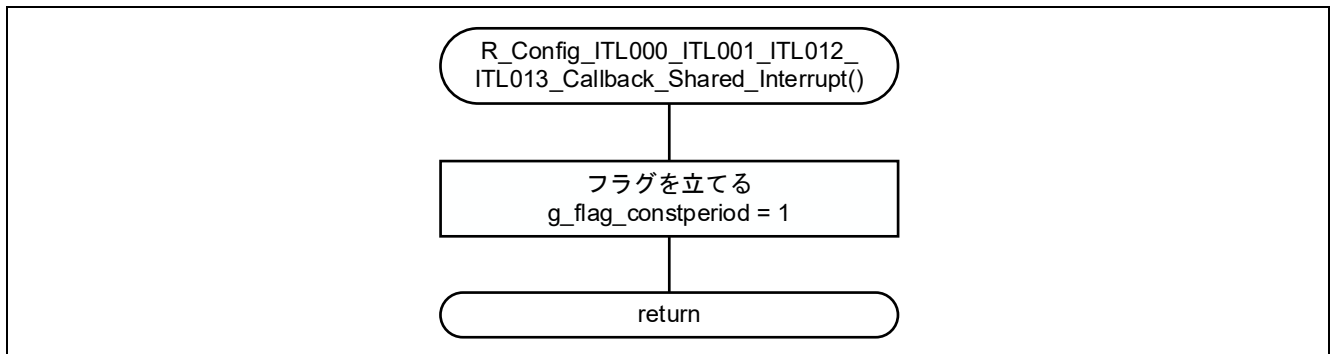
図 4-15 人感センサ検知割り込み(外部端子)



4.8.11 インターバル・タイマ割り込み

図 4-17 にインターバル・タイマ割り込みのフローチャートを示します。

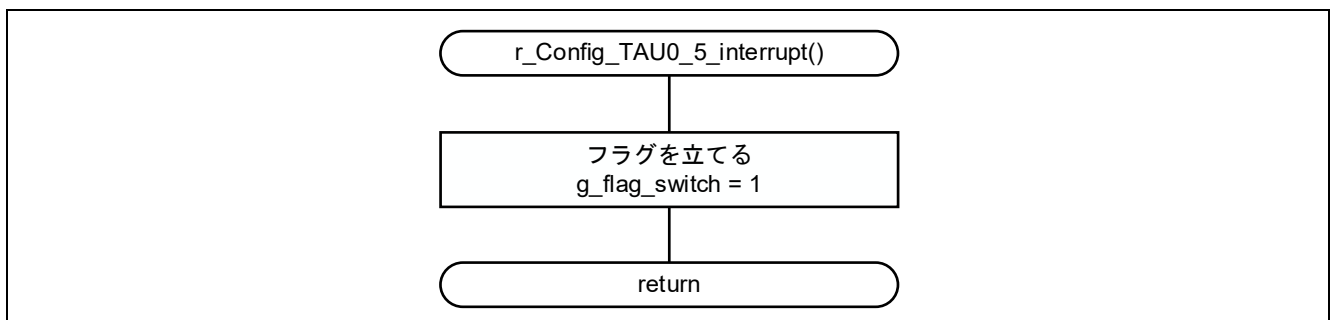
図 4-16 インターバル・タイマ割り込み



4.8.12 スイッチ入力検知割り込み(チャタリングフィルタ済み)

図 4-18 にスイッチ入力検知割り込み(チャタリングフィルタ済み)のフローチャートを示します。

図 4-17 スイッチ入力検知割り込み(チャタリングフィルタ済み)

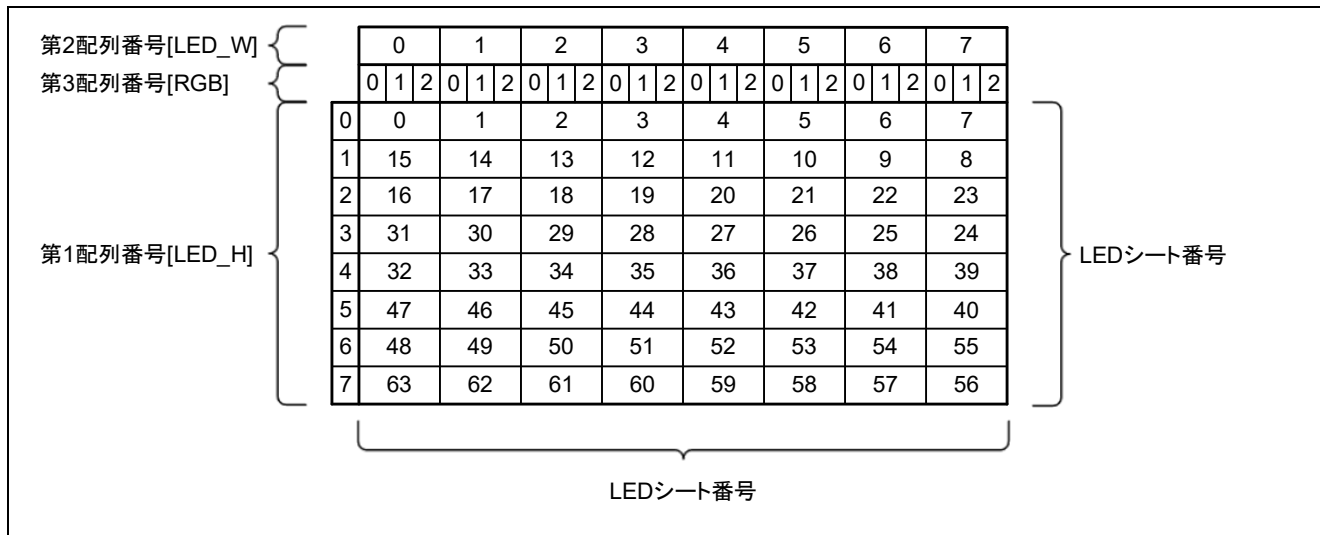


4.9 LED 制御データ フォーマットイメージ

図 4-18 に WS2812B(LED シート)を制御するシリアルデータ・フォーマットイメージを示します。

シリアルデータは CSI 送信用バッファ変数(g_buf_csi[LED_H][LED_W][RGB])に格納しています。

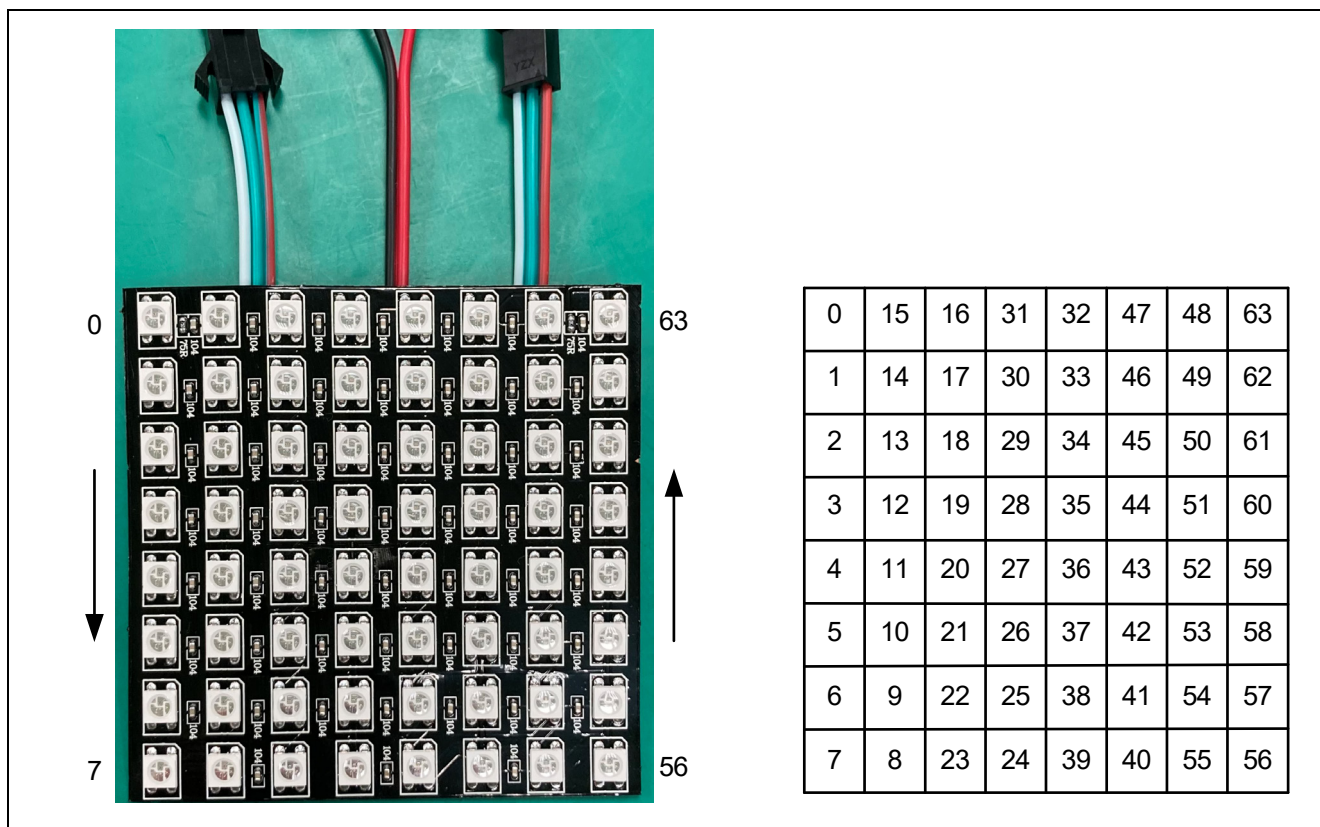
図 4-18 g_buf_csi[LED_H][LED_W][RGB]に格納される LED 制御データ・フォーマットイメージ



詳細は WS2812B のデータシートを参照してください。

図 4-19 に LED シート番号を示します。

図 4-19 LED シート番号



5. スマート・コンフィグレータの設定

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。

r01an7320_elcl.scfg

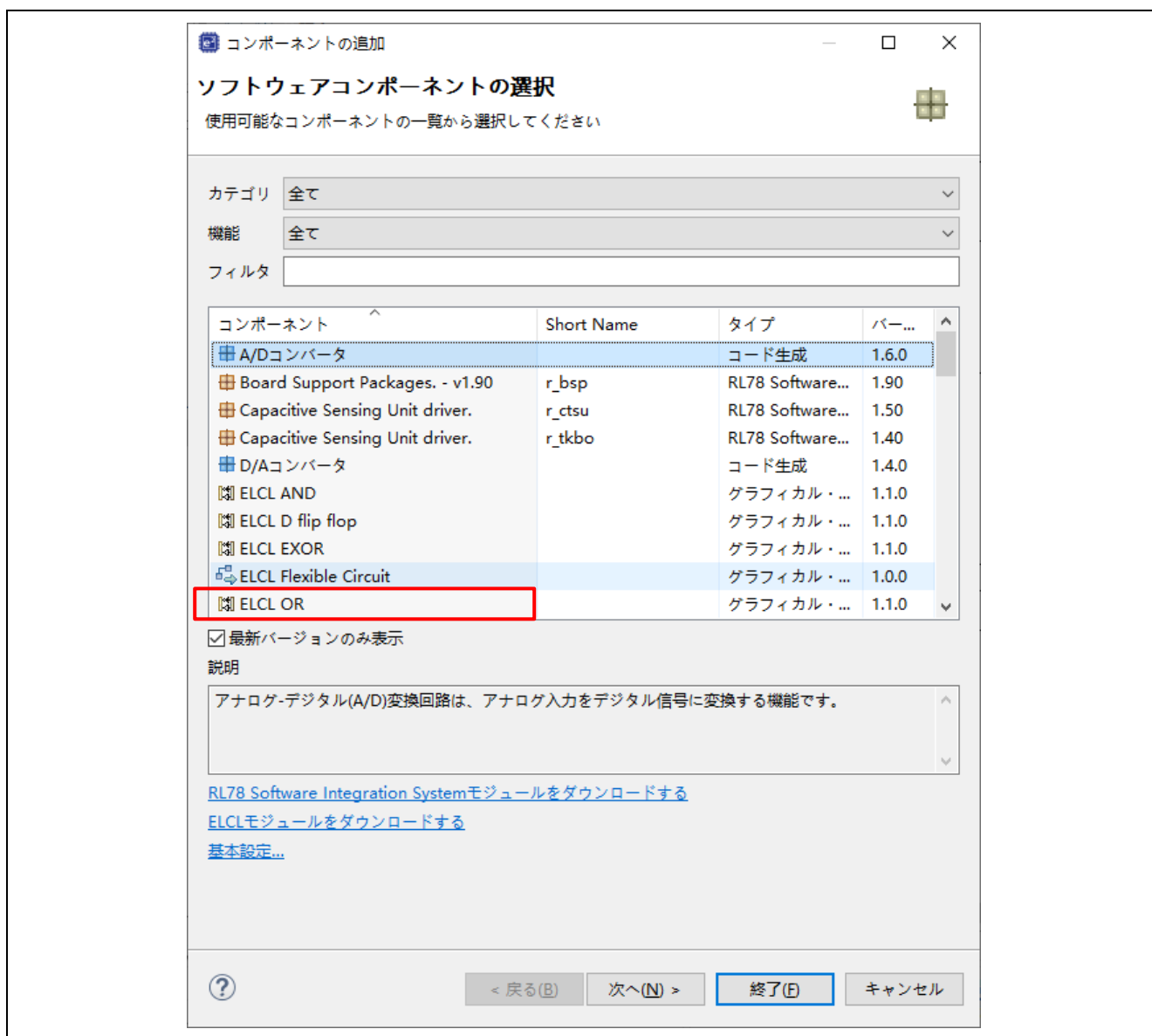
ファイルの説明と使用する上での設定例および注意事項を以下に示します。

5.1 ELCL のコンポーネントの設定

ELCL コンポーネントを使用する手順を以下に示します。

1. スマート・コンフィグレータを起動してください。
2. 「コンポーネント」タブをクリックし、「コンポーネントの追加」をクリックしてください。
3. 図 5-1 に示す「コンポーネントの追加」のウィンドウが開きますので、「ELCL Flexible Circuit」を選択し「終了」をクリックしてください。

図 5-1 「ELCL Flexible Circuit」の追加



5.2 r01an7320_elcl.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 5-1 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 4.0 (V) ~ 5.5 (V) EV _{DD} 設定：4.0V ≤ 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） (XT1 発振回路) 動作モード：XT1 発振 周波数：32.768kHz XT1 発振モード：低消費発振 1 供給モード：STOP.HALT モード時の供給許可
システム	-	オンチップ・デバッグ動作設定：COM ポート注 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：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する
コンポーネント	r_bsp	Start up select：Enable (use BSP startup) Control of invalid memory access detection(IAWEN)：Disable Protected area in the RAM (GRAM0-1)：Disabled Protection of the port control registers (GPORT)：Disabled Protection of the interrupt control registers (GINT)：Disabled Protection of the clock, voltage detector, and RAM parity error detection control registers (GCSC)：Disabled Data flash memory area/extra area access control (DFLEN)：Disables Initialization of peripheral functions by Code Generator/Smart Configurator：Enable API functions disable (R_BSP_StartClock, R_BSP_StopClock)：Disable API functions disable (R_BSP_GetFclkFreqHz)：Enable API functions disable (R_BSP_SetClockSource)：Disable API functions disable (R_BSP_ChangeClockSetting)：Disable API functions disable (R_BSP_SoftwareDelay)：Disable Parameter check enable：Enable Enable user warm start callback (PRE)：Unused Enable user warm start callback (POST)：Unused Watchdog Timer refresh enable：Unused

注. 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

表 5-2 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	INTP1 設定 : INTP1 有効エッジ : 立ち上がりエッジ 優先順位 : レベル 3(低優先順位)
	Config_ITL000_ITL001_ITL012_ITL013	動作クロック (f _{ITL0}) : f _{IHP} クロックソース : f _{ITL0} インターバル時間 : 100ms 割り込み設定 : 使用する 優先順位 : レベル 3(低優先順位)
	Config_TAU0_0	動作クロック : CK00 クロックソース : f _{CLK} 周期設定 : 1250μs 割り込み設定 : 使用しない PWM スレーブ選択設定 : チャンネル 1 スレーブ、チャンネル 2 スレーブ、チャンネル 3 スレーブ (スレーブ 1) デューティ : 27% 初期出力値 : 0 出力レベル : アクティブ・ハイ 出力設定 : タイマ出力は ELCL への入力にのみ使用します。TO01 ピンに出力しません。 割り込み設定 : 使用しない (スレーブ 2) デューティ : 64% 初期出力値 : 0 出力レベル : アクティブ・ハイ 出力設定 : タイマ出力は ELCL への入力にのみ使用します。TO02 ピンに出力しません。 割り込み設定 : 使用しない (スレーブ 3) デューティ : 100% 初期出力値 : 1 出力レベル : アクティブ・ハイ 出力設定 : タイマ出力は ELCL への入力にのみ使用します。TO03 ピンに出力しません。 割り込み設定 : 使用しない
	Config_TAU0_5	動作クロック : CK00 クロックソース : f _{CLK} 入力ソース設定 : ELCL 外部イベント・エッジ選択(TI05) : 両エッジ ディレイ : 2ms 割り込み設定 : 使用する 優先順位 : レベル 3(低優先順位)

表 5-3 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_ELCL	コンポーネント : ELCL Flexible Circuit (Detail setting : Input) Transmit data output from SAU0 channel0 Output from TAU0 channel 1 Output from TAU0 channel 2 Output from TAU0 channel 3 Input pin (P137/EI137) (Detail setting : L1L0, Pass-through) Input: Output from TAU0 channel 1 Output: SAU0 Ch0 fSCK/P61/SMS trigger/P12 (Detail setting : L1, Selector) Input 0: Output from TAU0 channel 1 Input 1: Output from TAU0 channel 2 Selector: Transmit data output from SAU0 channel0 Output: L2L0 input 0 (Detail setting : L2L0, OR) Input 0: L1 selector output Input 1: Output from TAU0 channel 1 Output: L3L0 input 0 (Detail setting : L2L1, Pass-through) Input: Input pin (P137/EI137) Output: TAU0 Ch5 input/P16/DAC0 HW trigger/CTSU HW trigger (Detail setting : L3L0, AND) Input 0: L2L0 output Input 1: Output from TAU0 channel 3 Output: SAU0 Ch1 fSCK/TAU0 Ch0 input/P13 (Detail setting : Output) Output of signal 2: SAU0 Ch0 fSCK/P61/SMS trigger/P12 Output of signal 3: SAU0 Ch1 fSCK/TAU0 Ch0 input/P13 Output of signal 6: TAU0 Ch5 input/P16/DAC0 HW trigger/CTSU HW trigger
	Config_PORT	コンポーネント : Ports ポート選択 : PORT1 P13 : 出力(ELCL 出力信号を出力する) P137 : 入力
	Config_CSI00	転送クロックモード : 外部クロック(スレーブ) 転送モード設定 : 連続転送モード データ・ビット長設定 : 8 ビット データ転送方向設定 : LSB データ送受信タイミング設定 : タイプ 2 転送レート設定 : 使用しない 割り込み設定 : レベル 3(低優先順位) コールバック機能設定 : 送信完了

5.2.1 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。

5.2.2 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 4-3 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。

5.2.3 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。

5.2.4 Config_TAU0_0

サンプルコードの TAU00 の設定を行います。

サンプルコードでは、0 コード、1 コードデータ送信用のデューティ比の異なる 2 つの PWM 出力、Reset コード送信用のデューティが 100%または 0%の PWM 出力を設定します。割り込みは使用しません。

5.2.5 Config_TAU0_5

サンプルコードの TAU05 の設定を行います。

サンプルコードでは、スイッチ入力のチャタリング除去機能として、ディレイカウント機能を使用します。割り込みベクタ処理にて、有効なスイッチ入力が行われた際の処理のトリガとなるフラグをセットします。

5.2.5.1 Config_TAU0_5.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出力されたコードに対して、下記の通り、黄色で色づけした箇所を変更しています。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たときは、スマート・コンフィグレータの出力値（変更前）に上書きされます。自動生成する度に更新してください。

関数名	R_Config_TAU0_5_Create
変更前	<pre>TMR05 = _0000_TAU_CLOCK_SELECT_CKM0 _0000_TAU_CLOCK_MODE_CKS _0100_TAU_TRIGGER_TIMN_VALID _0080_TAU_TIMN_EDGE_BOTH_LOW _0008_TAU_MODE_ONE_COUNT _0001_TAU_START_INT_USED;</pre>
変更後	<pre>TMR05 = _0000_TAU_CLOCK_SELECT_CKM0 _0000_TAU_CLOCK_MODE_CKS _0200_TAU_TRIGGER_TIMN_BOTH _0080_TAU_TIMN_EDGE_BOTH_LOW _0008_TAU_MODE_ONE_COUNT _0001_TAU_START_INT_USED;</pre>

5.2.6 Config_INTC

サンプルコードの INTP1 の設定を行います。

サンプルコードでは、人感センサの信号を立ち上がりエッジで検知します。割り込みベクタ処理にて、人感センサが感知した際の処理のトリガとなるフラグをセットします。

5.2.7 Config_ITL000_ITL001_ITL012_ITL013

サンプルコードの TML32 の設定を行います。

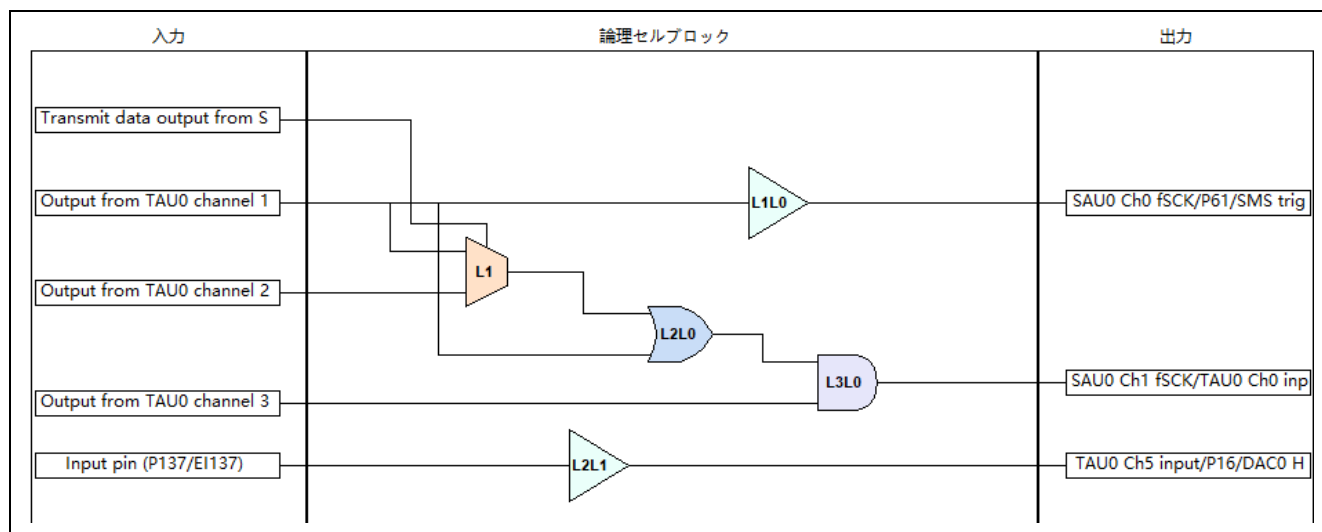
サンプルコードでは、LED の表示更新タイミングを生成するため 100ms のインターバル・タイマとして設定します。割り込みベクタ処理にて、一定周期ごとの処理のトリガとなるフラグをセットします。

5.2.8 Config_ELCL

サンプルコードの ELCL の設定を行います。

サンプルコードでは、L3L0(AND)を使用し、入力信号として LED 制御信号(L2L0 OR 出力)とデューティ 100%または 0%の PWM を選択します。

図 5-2 ELCL 設定



5.2.9 Config_CSI00

サンプルコードの CSI00 の設定を行います。

本サンプルコードでは、転送クロックモードを外部クロック(スレーブ)モードとし、データ送受信タイミングをタイプ 2 とすることで、LED 制御信号の 0 コード、1 コードを切り替えるセクタの選択制御の切り替えを 0 コード用 PWM の立ち下りエッジで行うように設定しています。

5.2.10 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、WS2812 制御のために使用する P13 を ELCL 出力信号に設定し、SW に使用する P137 を入力に設定します。

6. 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。

7. 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 (R01UH0896)

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

RL78 スマート・コンフィグレータ ユーザーガイド : CS+編 (R20AN0580)

RL78 スマート・コンフィグレータ ユーザーガイド : e² studio 編 (R20AN0579)

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	Jul.16.24	-	初版発行
2.00	Jul.22.25	-	Update to use Smart Configurator "ELCL Flexible Circuit" component to make application design instead of using "ELCL AND", "ELCL OR", "ELCL Selector" and "ELCL Through" component. ・「ELCL AND」、「ELCL OR」、「ELCL Selector」、「ELCL Through」コンポーネントの代わりにスマート・コンフィグレータの「ELCL Flexible Circuit」を使用してアプリケーションを設計するよう更新 ・設定ファイルの「ELCL AND」、「ELCL OR」、「ELCL Selector」、「ELCL Through」コンポーネントを削除 ・「ELCL Flexible Circuit」コンポーネントで作成された Config_ELCL を追加 ・メイン処理のフローチャートを更新 ・ELCL 入力に使用する Config_TAU0_0 の設定項目の追加 ・ELCL 入出力に使用する Config_PORT の設定項目の追加
		6	表 2-1 動作確認条件を更新

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。