

RL78/G23

ELCL を用いたエンコーダ信号カウント機能

要旨

本アプリケーションノートでは、直交エンコーダ信号からカウントパルスを出力する回路をロジック & イベント・リンク・コントローラ（ELCL）を用いて設計します。時計回り用と反時計回り用の回路をそれぞれ構成し、ELCL で生成した信号をタイマ入力に接続します。タイマのカウント値を基にモータの回転数を計算し、その結果を RL78/G23-64p Fast Prototyping Board の USB 経由で Tera Term 上に出力します。

動作確認デバイス

評価ボード	: RL78/G23-64p Fast Prototyping Board（以下 FPB）
モータドライバ	: AE-DRV8835-S
直交エンコーダ付 DC モータ	: 30:1 Metal Gearmotor 37Dx68L mm 12V with 64 CPR Encoder (Helical Pinion)

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	9
3. ハードウェア説明	10
3.1 ハードウェア構成例	10
3.2 使用端子一覧	10
4. ソフトウェア説明	11
4.1 動作概要	11
4.2 フォルダ構成	13
4.3 オプション・バイトの設定一覧	14
4.4 定数一覧	15
4.5 変数一覧	15
4.6 関数一覧	16
4.7 関数仕様	16
4.8 フローチャート	19
4.8.1 メイン処理	19
4.8.2 回転数の概算	21
4.8.3 回転数を送信	22
4.8.4 モータの動作状態変更	23
4.8.5 モータのデューティ比を設定	26
4.8.6 インターバル・タイマ割り込み	27
4.8.7 スイッチ押下検知割り込み	27
4.8.8 PWM 機能のタイマ割り込み	28
4.8.9 UART0 送信完了	28
4.9 モータ制御	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 r01an7635_elcl.scfg	35
5.2.1 クロック	39
5.2.2 システム	39
5.2.3 r_bsp	39
5.2.4 Config_TAU0_0	39
5.2.5 Config_TAU0_1	39
5.2.6 Config_TAU0_2	39
5.2.7 Config_INTC	39
5.2.8 Config_ITL000_ITL001	40
5.2.9 Config_UART0	40
5.2.10 Config_ELCL	40
5.3 エンコーダとモータを変更する場合	41
6. 動作確認時の Tera Term の使用方法	42
6.1 Tera Term の設定	42

7. プロジェクトのインポート方法	44
7.1 e2 studio での手順.....	44
8. サンプルコード	45
9. 参考ドキュメント.....	45

1. 仕様

本アプリケーションノートでは、ELCL を利用して直交エンコーダ信号のカウント機能を実現します。
本システムのシステム構成図を図 1-1 に示します。

図 1-1 システム構成図

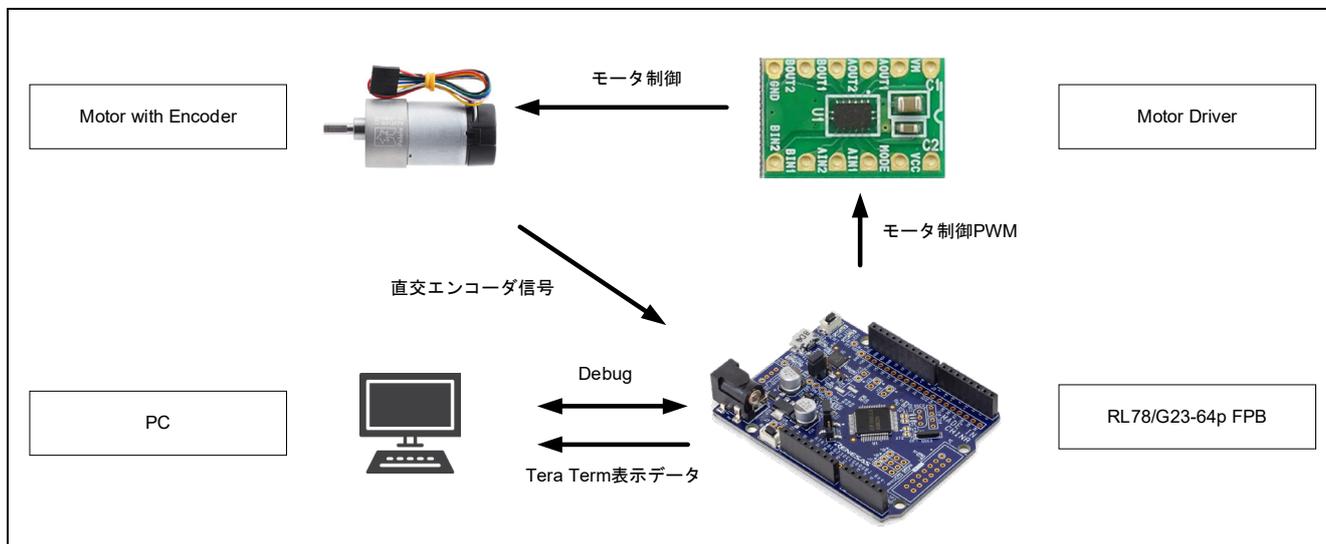
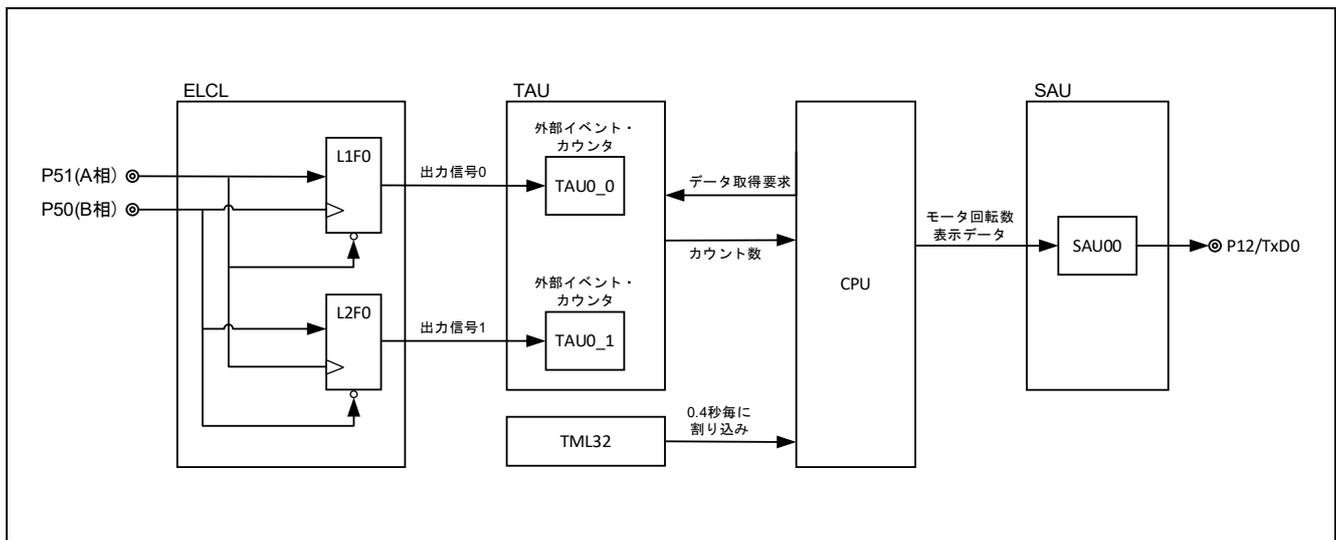


図 1-2 に ELCL を使用して直交エンコーダ信号のカウント機能を実現する構成を示します。直交エンコーダの出力（A 相、B 相）信号を P51 と P50 から ELCL に取り込み、ELCL 内の 2 つのフリップフロップ（L1F0, L2F0）を経由することで、モータの時計回りおよび反時計回りの回転を判別します。L1F0 からの出力は、時計回りの回転信号（出力信号 0）です。この回転信号の立ち上がりエッジをタイマ・アレイ・ユニット（TAU）の外部イベント・カウンタ機能を用いてカウントすることで、モータの回転位置や回転数を計算できます。同様に、L2F0 からの出力である反時計回りの回転信号（出力信号 1）を用いることで、モータの回転位置や回転数を計算できます。

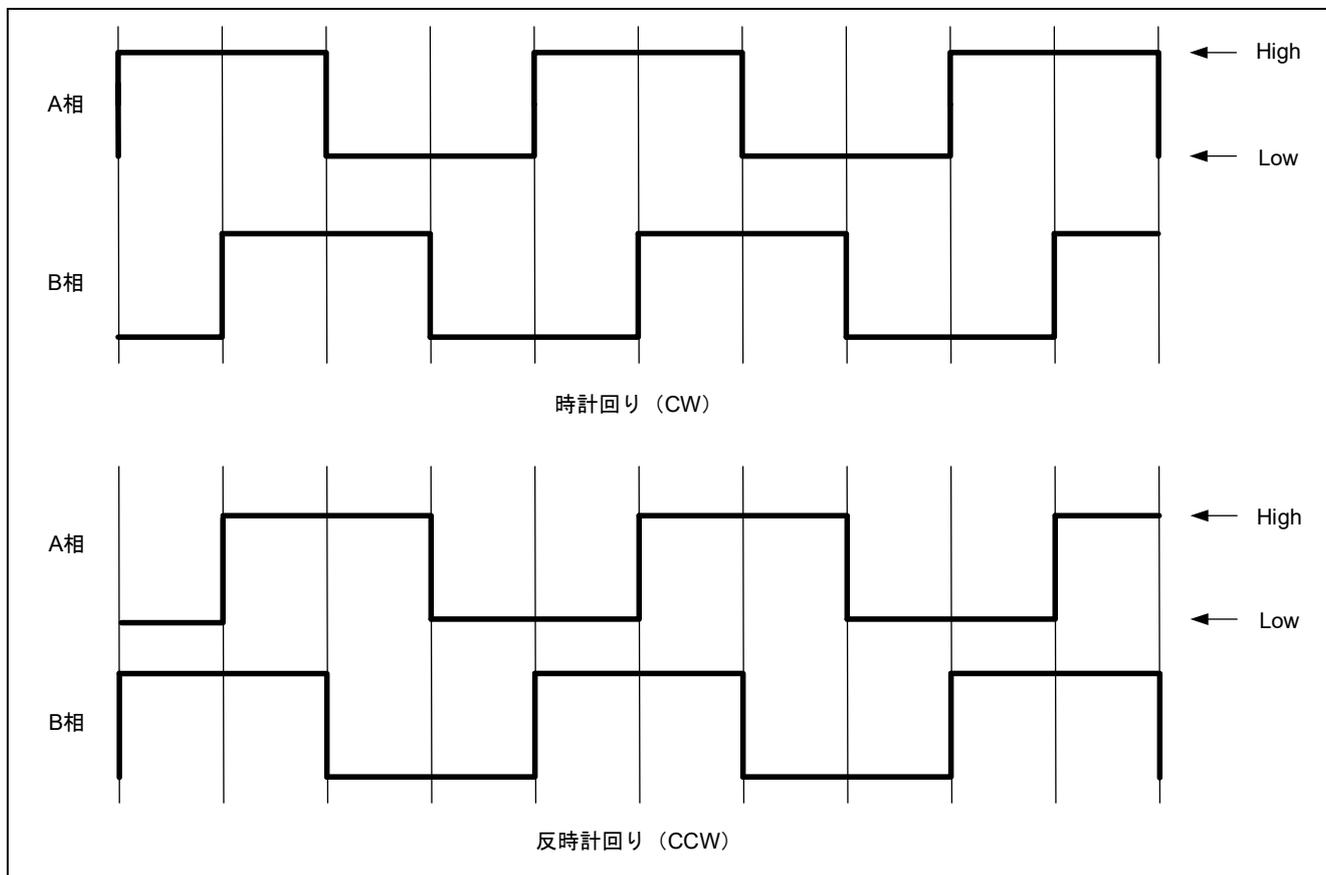
本システムでは、ボード搭載のユーザースイッチの押下により、モータの回転方向および停止が切り替わり、モードに応じて LED の点灯パターンも切り替わります。また、出力信号のカウント数を 1 分間の回転数へ換算し Tera Term 表示データに換算し、TxD0 端子から PC（Tera Term）へ UART 送信します。

図 1-2 直交エンコーダ信号カウント機能の構成



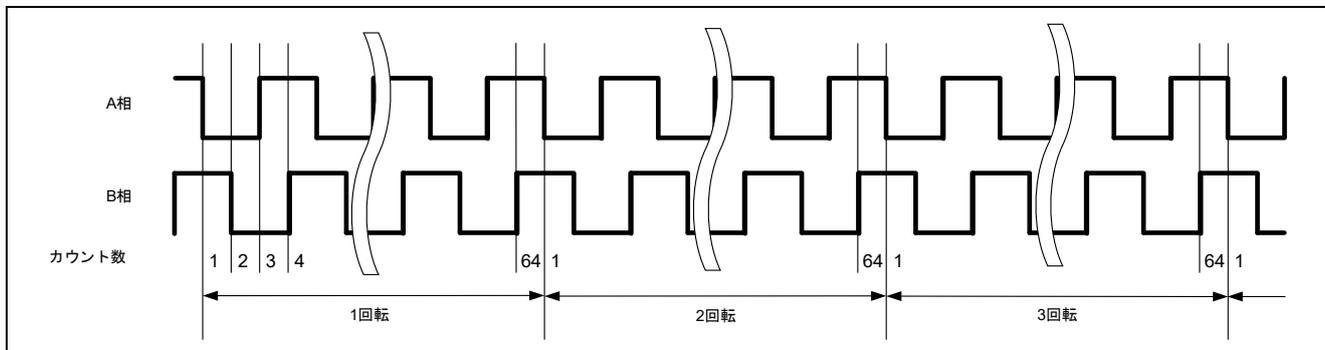
エンコーダとは、回転や水平移動を行うさまざまな機器/装置の移動方向や移動量、角度を検出し、電気信号を出力する電子部品です。長方形の穴（スリット）が開いた回転する円盤に光を当て、スリットを通過した光を信号変換することで 90 度位相差の 2 本の方角波を出力します。図 1-3 に時計回りに回転する場合と反時計回りに回転する場合の直交エンコーダ信号の波形を示します。

図 1-3 直交エンコーダ信号



本アプリケーションノートで使用するエンコーダの分解能は64です。ここでの分解能とは、1回転におけるA相とB相の両波形の両エッジのカウント数です。図1-4に時計回り回転でのカウント例を示します。

図1-4 カウント例



ELCL を用いたエンコーダ信号のカウント方法で、図 1-5 に時計回りの回転をする場合のタイミングチャートを、図 1-6 に反時計回りの回転をする場合のタイミングチャートを示します。本アプリケーションノートでは、片方の波形の立ち上がりエッジでのみカウントするため、分解能の 1/4 になります。

- (1) エンコーダ信号の A 相、B 相を P51、P50 から ELCL へ入力します
- (2) L1F0 から出力信号 0、L2F0 から出力信号 1 が出力されます

図 1-5 時計回り回転のタイミングチャート

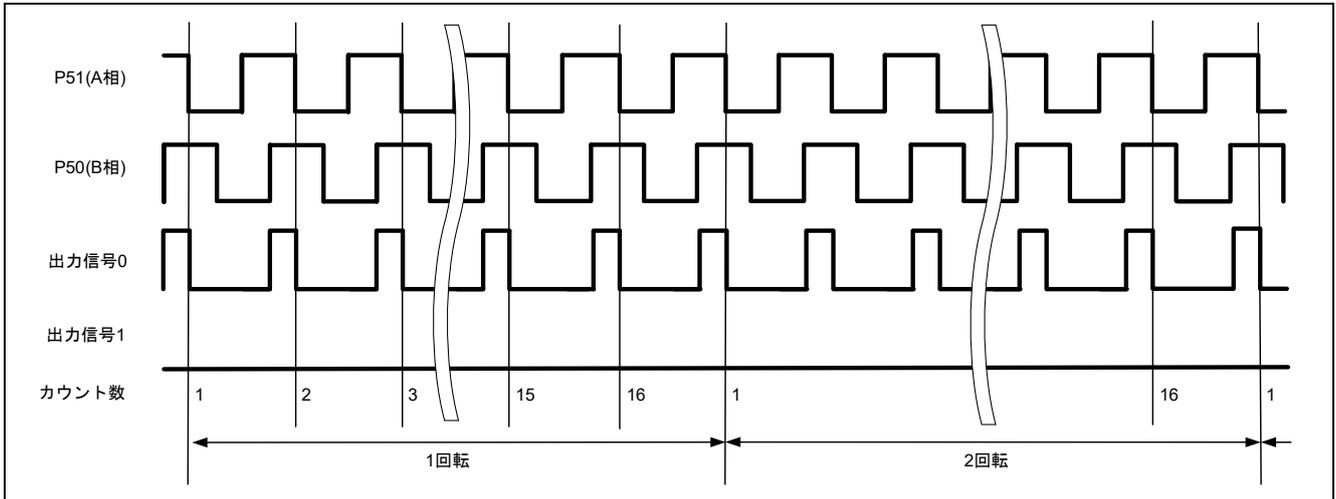
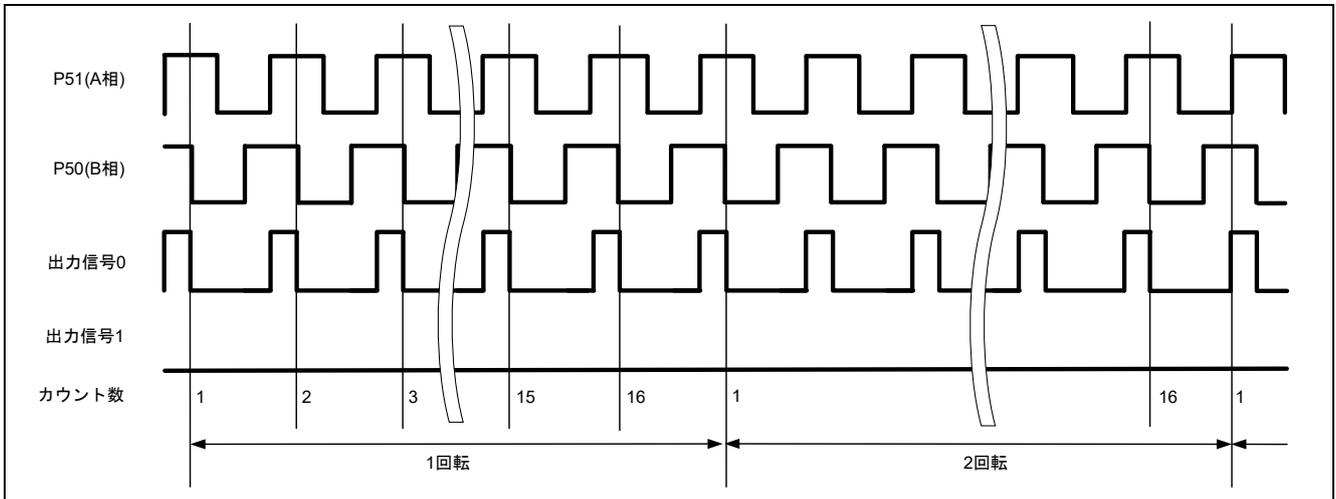


図 1-6 反時計回り回転のタイミングチャート



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

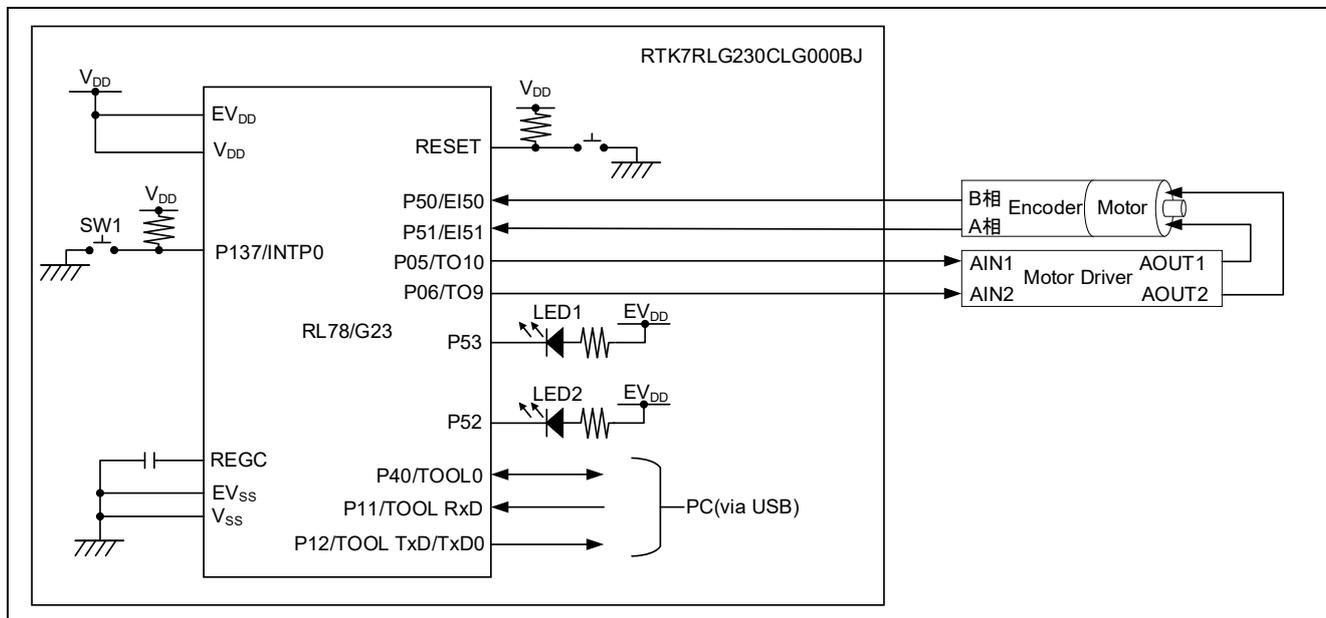
項目	内容
使用ボード	RL78/G23-64p Fast Prototyping Board
動作周波数	<ul style="list-style-type: none"> ● 高速オンチップ・オシレータ・クロック : 32MHz ● CPU/周辺ハードウェア・クロック : 32MHz
動作電圧	<ul style="list-style-type: none"> ● 5.0V
スマート・コンフィグレータ	v1.13.0
統合開発環境 (e ² studio)	ルネサスエレクトロニクス製 e ² studio 2025-04 (25.4.0)
C コンパイラ (e ² studio)	ルネサスエレクトロニクス製 CC-RL V1.15.00
統合開発環境 (IAR)	IAR システム製 IAR Embedded Workbench for Renesas RL78 V 5.20.1
C コンパイラ (IAR)	
総合開発環境(CS+)	ルネサスエレクトロニクス製 CS+ V8.13.00
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 CS+ : COM ポート IAR : E2 エミュレータ Lite
モータドライバ	AE-DRV8835-S
直交エンコーダ付 DC モータ	30:1 Metal Gearmotor 37Dx68L mm 12V with 64 CPR Encoder (Helical Pinion)

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 3-2 に使用端子と機能を示します。

表 3-2 使用端子と機能

端子名	入出力	内容
P05/TO10	出力	モータの動作及びスピードの制御 (High Active)
P06/TO9	出力	モータの動作及びスピードの制御 (High Active)
P51	入力	ELCL の入力信号 (直交エンコーダ信号 A 相)
P50	入力	ELCL の入力信号 (直交エンコーダ信号 B 相)
P137/INTP0	入力	SW1 入力
P52	出力	LED2 点灯 (Low Active)
P53	出力	LED1 点灯 (Low Active)
P12/TxD0	出力	UART 送信

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、P50, P51 の入力信号に対して下記 2 つの ELCL モジュールを用いてカウントパルス信号を生成します。生成した信号は TAU0_0、TAU0_1 へ出力します。

図 4-1 にカウントパルス信号生成システム構成を示します。

P51 を L1L0 の入力信号とします。P50 を Clock 入力、P51 の負論理を Reset 入力に設定します。出力信号 0 を TAU0_0 へ出力します。

P50 を L2L0 の入力信号とします。P51 を Clock 入力、P50 の負論理を Reset 入力に設定します。出力信号 1 を TAU0_1 へ出力します。

TAU0_0, TAU0_1 で立ち上がりエッジをカウントし、カウント値から 1 分間の回転数を算出して UART で Tera Term 上に表示します。

図 4-1 エンコーダ信号カウントパルス生成システム構成

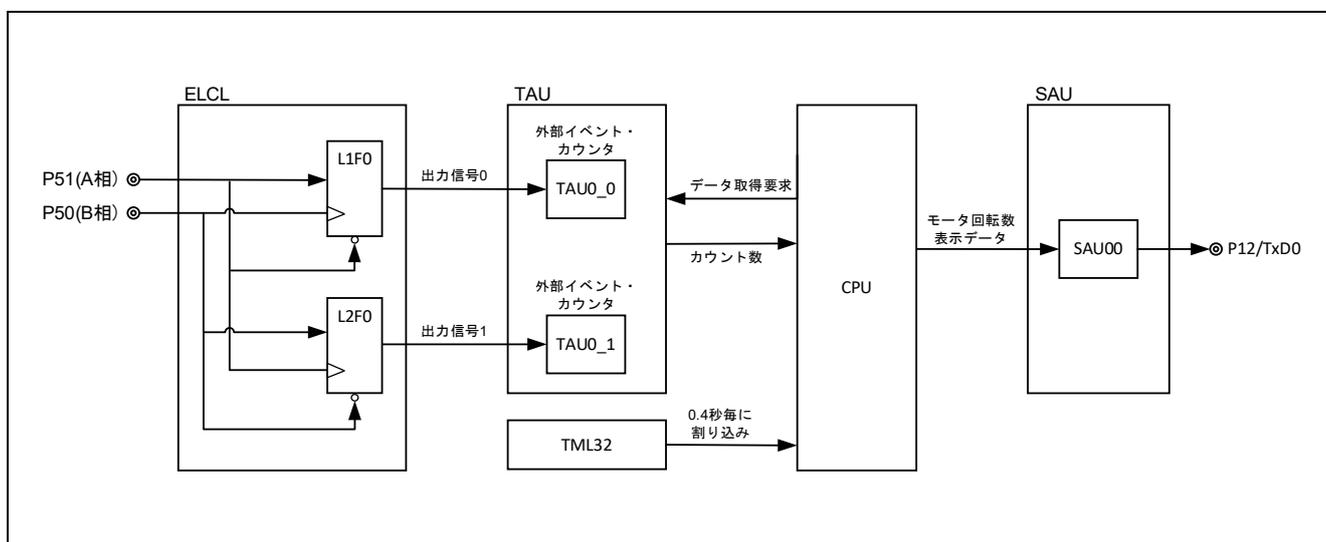
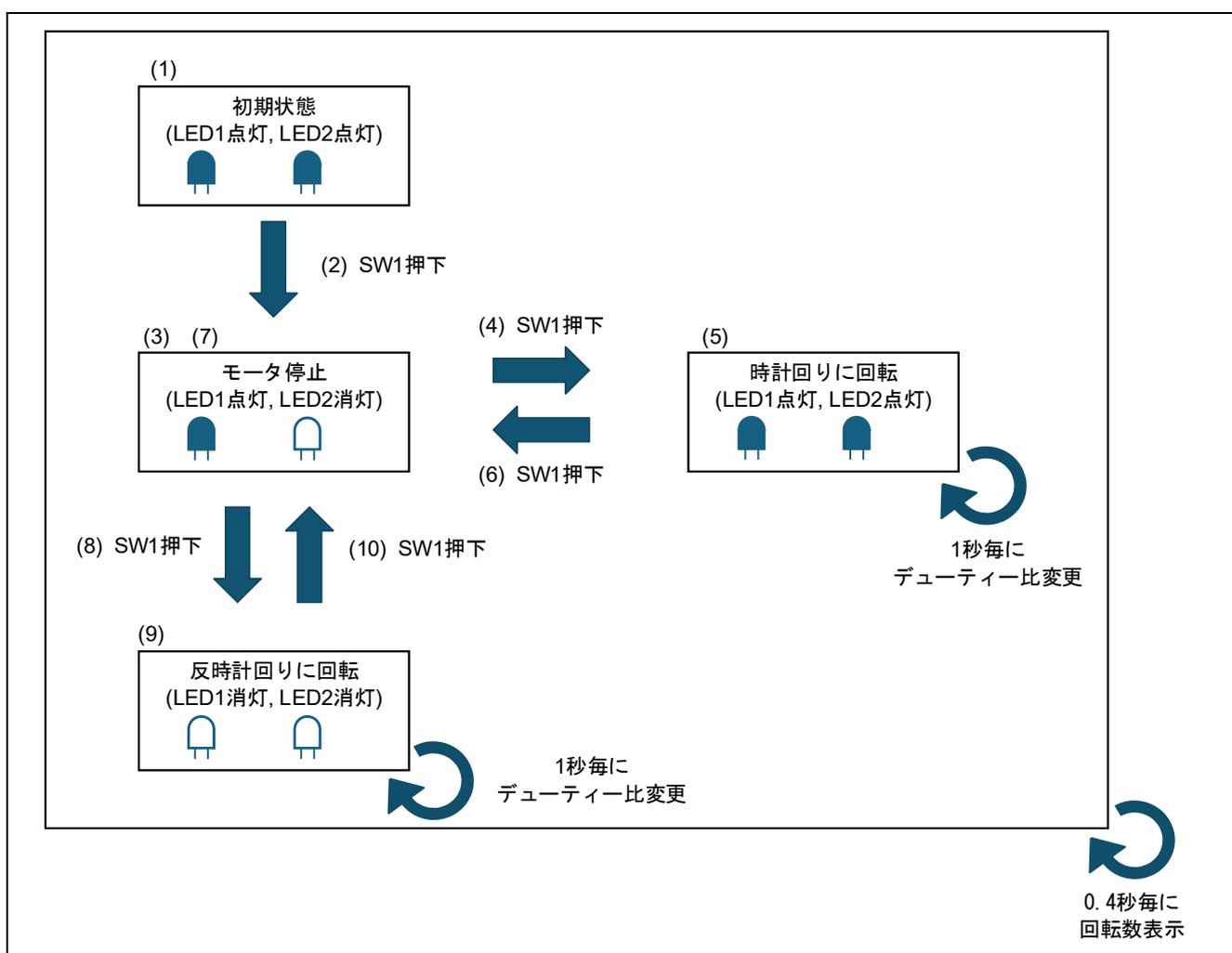


図 4-2 に動作概要を示します。

- (1) 動作を開始すると LED1, LED2 が点灯します
- (2) SW1 を押下します
- (3) モータは停止し LED1 は点灯、LED2 は消灯します
- (4) SW1 を押下します
- (5) モータは時計回りに回転し LED1, LED2 が点灯します
- (6) SW1 を押下します
- (7) モータは停止し LED1 は点灯、LED2 は消灯します
- (8) SW1 を押下します
- (9) モータは反時計回りに回転し LED1, LED2 が消灯します
- (10) (3)へ移動します

図 4-2 動作概要



4.2 フォルダ構成

表 4-1、表 4-2 にサンプルコードの使用するソースファイル/ヘッダファイルの構成を示します。
 なお、統合開発環境で自動生成されるファイル、bsp 環境のファイルは除きます。

表 4-1 フォルダ構成 (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
¥CS+/e2stdio/IAR<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	
main.h	サンプル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_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	√
¥Config_ITL000_ITL001<DIR>	ITL000_ITL001 用プログラム格納フォルダ	√
Config_ITL000_ITL001.c	ITL000_ITL001 用ソースファイル	√
Config_ITL000_ITL001.h	ITL000_ITL001 用ヘッダファイル	√
Config_ITL000_ITL001.c	ITL000_ITL001 用割り込みソースファイル	√
¥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}

表 4-2 フォルダ構成 (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
¥CS+/e2studio/IAR<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	
main.h	サンプル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_TAU0_0 <DIR>	TAU0_0 用プログラム格納フォルダ	√
Config_TAU0_0.c	TAU0_0 用ソースファイル	√
Config_TAU0_0.h	TAU0_0 用ヘッダファイル	√
Config_TAU0_0_user.c	TAU0_0 用割り込みソースファイル	√
¥Config_TAU0_1 <DIR>	TAU0_1 用プログラム格納フォルダ	√
Config_TAU0_1.c	TAU0_1 用ソースファイル	√
Config_TAU0_1.h	TAU0_1 用ヘッダファイル	√
Config_TAU0_1_user.c	TAU0_1 用割り込みソースファイル	√
¥Config_TAU0_2 <DIR>	TAU0_2 用プログラム格納フォルダ	√
Config_TAU0_2.c	TAU0_2 用ソースファイル	√
Config_TAU0_2.h	TAU0_2 用ヘッダファイル	√
Config_TAU0_2_user.c	TAU0_2 用割り込みソースファイル	√
¥Config_UART0 <DIR>	UART0 用プログラム格納フォルダ	√
Config_UART0.c	UART0 用ソースファイル	√
Config_UART0.h	UART0 用ヘッダファイル	√
Config_UART0_user.c	UART0 用割り込みソースファイル	√

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。

4.3 オプション・バイトの設定一覧

表 4-3 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 4-3 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 4-4 にサンプルコードで使用する定数を示します。

表 4-4 サンプルコードで使用する定数

定数名	設定値	内容
SCALE_FACTOR	0x00000064	オーバーフロー抑止のスケール補正值
AIN1_HIGH	0x20	モータ制御用の信号 (AIN1 を HIGH に設定)
AIN2_HIGH	0x40	モータ制御用の信号 (AIN2 を HIGH に設定)
AIN1_LOW	0xDF	モータ制御用の信号 (AIN1 を LOW に設定)
AIN2_LOW	0xBF	モータ制御用の信号 (AIN2 を LOW に設定)
LED1_ON	0xF7	LED1 を点灯状態に設定
LED2_ON	0xFB	LED2 を点灯状態に設定
LED1_OFF	0x08	LED1 を消灯状態に設定
LED2_OFF	0x04	LED2 を消灯状態に設定
BRAKE1	0	モータ制御モード (ブレーキ 1)
CW	1	モータ制御モード (時計回り)
BRAKE2	2	モータ制御モード (ブレーキ 2)
CWW	3	モータ制御モード (反時計回り)
DUTY_10	0x1900	モータ制御用のデューティサイクル (10%)
DUTY_90	0xE100	モータ制御用のデューティサイクル (90%)

4.5 変数一覧

表 4-5 にグローバル変数を示します。ただし、スマート・コンフィグレータで生成された変数の内、変更を行っていないものは除きます。

表 4-5 グローバル変数

型	変数名	内容	使用関数
uint8_t	g_flag_itl_periodic	インターバル・タイマでの一定期間経過フラグ	main R_Config_ITL000_ITL001_Callback_Shared_Interrupt
uint8_t	g_flag_switch	スイッチ押下フラグ	main、__near r_Config_INTC_intp0_interrupt
uint8_t	g_flag_pwm_periodic	タイマの PWM 機能での一定期間経過フラグ	main、__near r_config_TAU0_2_channel5_interrupt
MD_STATUS	g_uart0_tx_end	UART0 の送信完了を示すフラグ	main r_Config_UART0_callback_sendend __near r_Config_UART0_interrupt_send
uint16_t	g_max_count	最大カウント数	main、__near R_Config_TAU0_0_Create_UserInit R_Config_TAU0_1_Create_UserInit

4.6 関数一覧

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

表 4-6 関数一覧

関数名	概要
main ()	メイン処理
calc_rpm ()	パルス数のカウント値から回転数を算出
send_rpm ()	計算された CW および CWW の回転数を UART 送信
change_motor_action ()	モータの動作状態を変更
set_motor_dutycycle ()	モータの Duty 比を設定
R_Config_ITL000_ITL001_Callback_Shared_Interrupt ()	インターバル・タイマ割り込み
r_Config_INTC_intp0_interrupt ()	スイッチの外部割り込み
r_Config_TAU0_2_channel5_interrupt ()	PWM 機能のタイマ割り込み
r_Config_UART0_interrupt_send ()	UART0 送信完了割り込み

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 関数

概要	メイン処理
ヘッダ	Stdint.h、stdio.h、string.h、r_smc_entry.h、main.h
宣言	int main (void);
説明	ELCL、インターバル・タイマ、タイマ、INTC の動作を開始します フラグを確認し、必要な関数を呼び出します
引数	なし
リターン値	なし
備考	なし

[関数名] calc_rpm

概要	モータの CW および CWW 方向の回転速度を計算
ヘッダ	Stdint.h、stdio.h、string.h、r_smc_entry.h、main.h
宣言	void calc_rpm(uint16_t * rpm_cw, uint16_t * rpm_cww);
説明	指定した変数から 1 分間の回転速度を概算します
引数	rpm_cw, rpm_cww
リターン値	なし
備考	なし

[関数名] send_rpm

概要	計算された時計回り及び反時計回りの回転速度を UART 送信
ヘッダ	Stdint.h、stdio.h、string.h、r_smc_entry.h、main.h
宣言	void send_rpm(uint16_t rpm_cw, uint16_t rpm_cww);
説明	指定した変数を UART 送信します
引数	rps_cw, rpm_cww
リターン値	なし
備考	なし

[関数名] change_motor_action

概要	モータの動作状態を変更
ヘッダ	Stdint.h、stdio.h、string.h、r_smc_entry.h、main.h
宣言	void change_motor_action(void);
説明	モータの動作状態を変更します
引数	なし
リターン値	なし
備考	なし

[関数名] set_motor_dutycycle

概要	モータの Duty 比を設定
ヘッダ	Stdint.h、stdio.h、string.h、r_smc_entry.h、main.h
宣言	void set_motor_dutycycle(void);
説明	モータのデューティ比を設定します
引数	なし
リターン値	なし
備考	なし

[関数名] R_Config_ITL000_ITL001_Callback_Shared_Interrupt

概要	ITL000 および ITL001 チャンネルの共通割り込み処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_ITL000_ITL001.h
宣言	void R_Config_ITL000_ITL001_Callback_Shared_Interrupt(void);
説明	コンペアー一致検出フラグを"0"にクリアします インターバル・タイマでの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_INTC_intp0_interrupt

概要	INTP0（外部割り込み 0）の割り込みサービスルーチン（ISR）
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_INTC.h
宣言	static void __near r_Config_INTC_intp0_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

[関数名] r_Config_TAU0_2_channel5_interrupt

概要	INTTM05（チャンネル5）の割り込みサービスルーチン（ISR）
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_TAU0_2.h
宣言	static void __near r_Config_TAU0_2_channel5_interrupt(void);
説明	PWM 機能のタイマでの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_UART0_interrupt_send

概要	UART0 の送信割り込みサービスルーチン（ISR）
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h、Config_UART0.h
宣言	static void __near r_Config_UART0_interrupt_send(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

4.8 フローチャート

4.8.1 メイン処理

図 4-3、図 4-4 にメイン処理のフローチャートを示します。

図 4-3 メイン処理 (1/2)

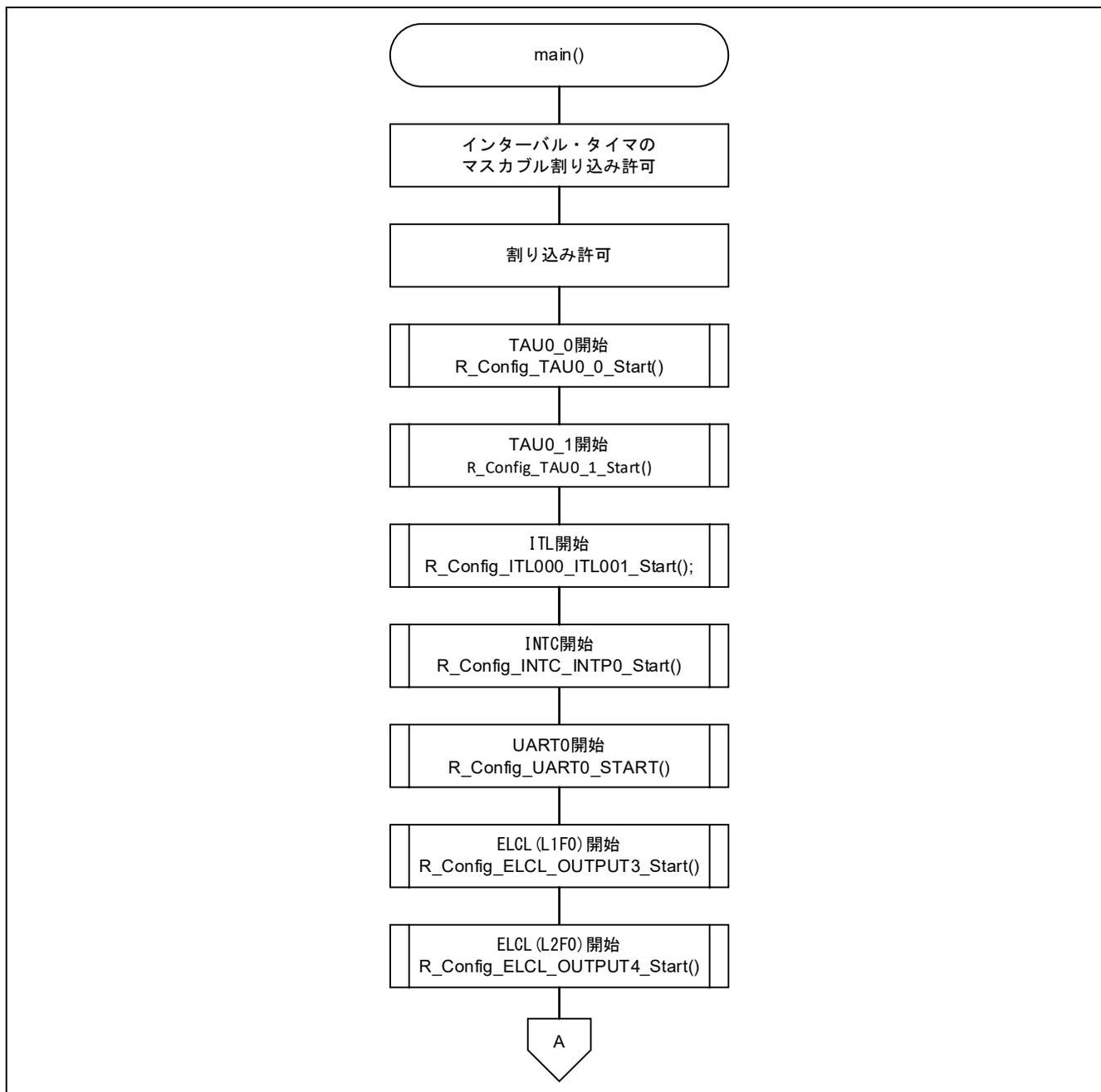
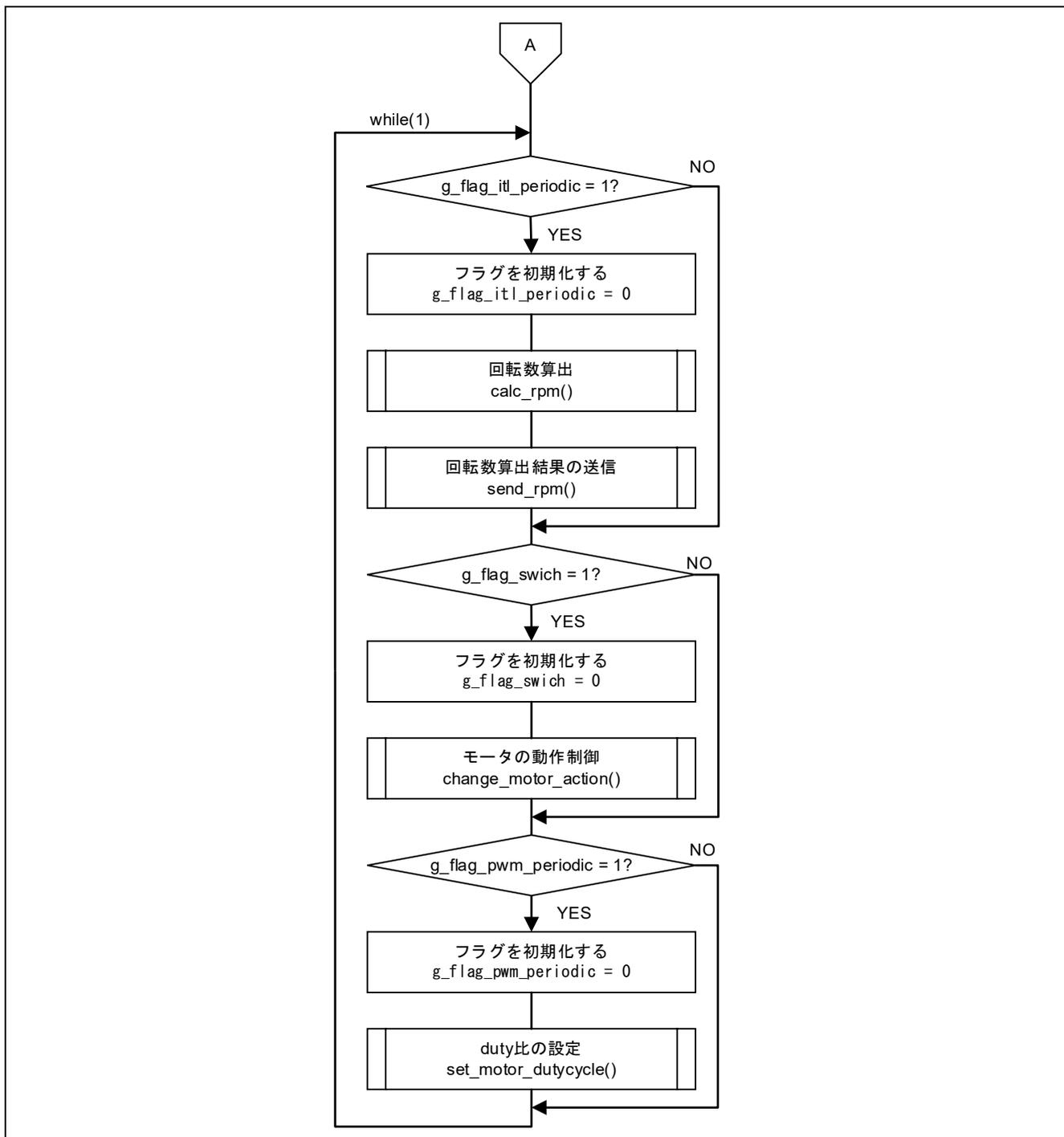


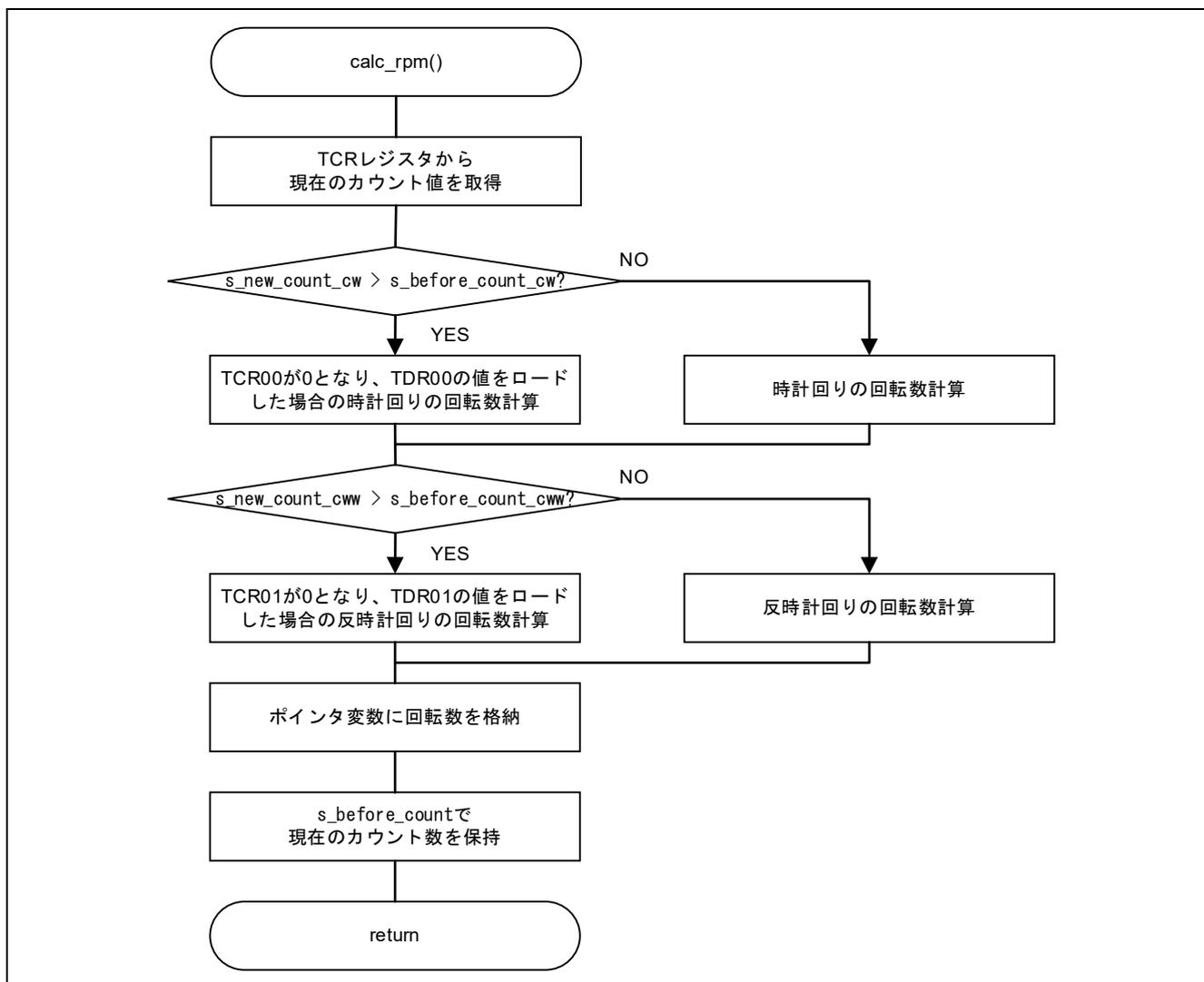
図 4-4 メイン処理 (2/2)



4.8.2 回転数の概算

図 4-5 に回転数の概算のフローチャートを示します。

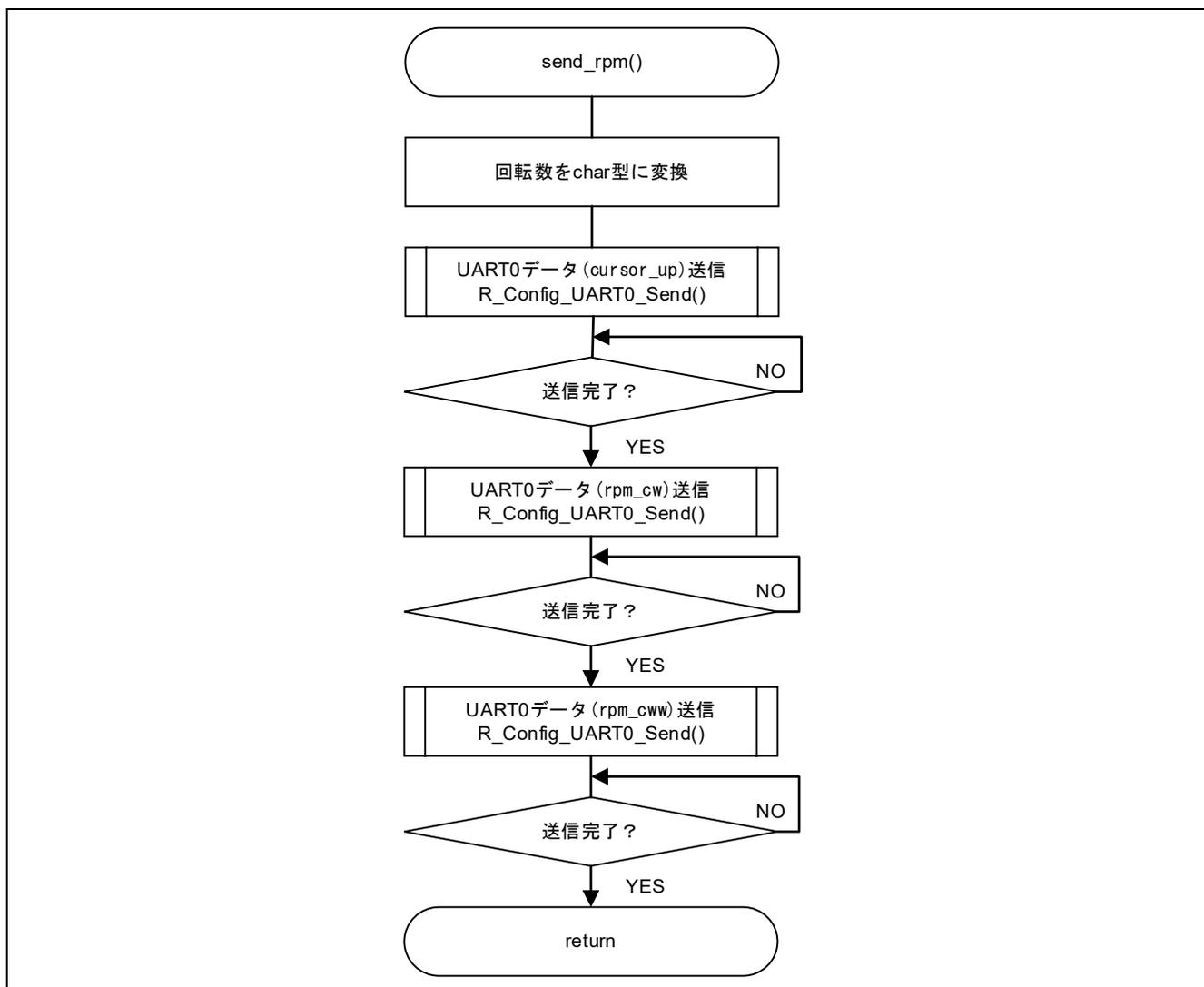
図 4-5 回転数の概算



4.8.3 回転数を送信

図 4-6 に回転数を送信のフローチャートを示します。

図 4-6 回転数を送信



4.8.4 モータの動作状態変更

図 4-7、図 4-8、図 4-9、図 4-10、図 4-11 にモータの動作状態変更のフローチャートを示します。

図 4-7 モータの動作状態変更 (1/5)

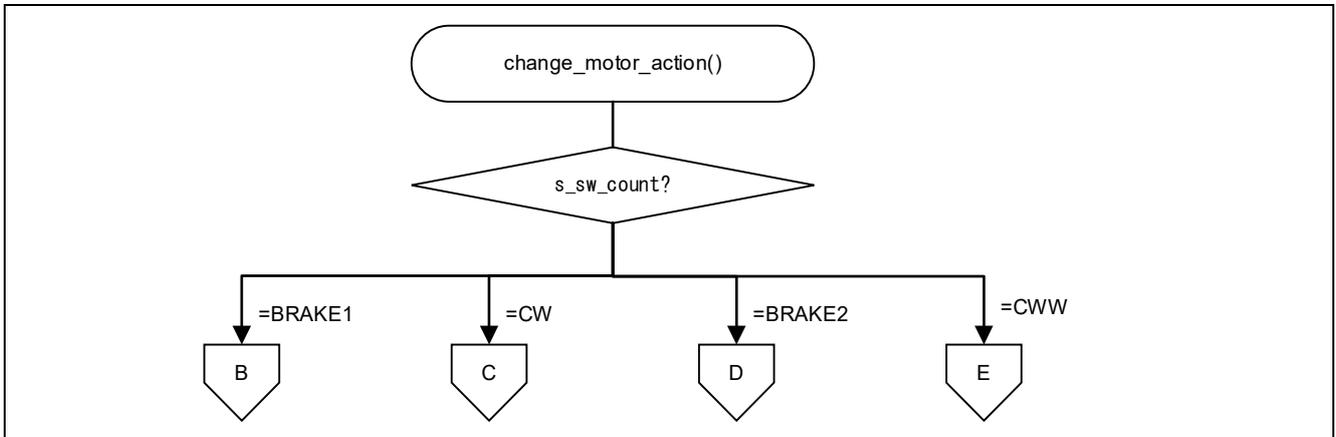


図 4-8 モータの動作状態変更 (2/5)

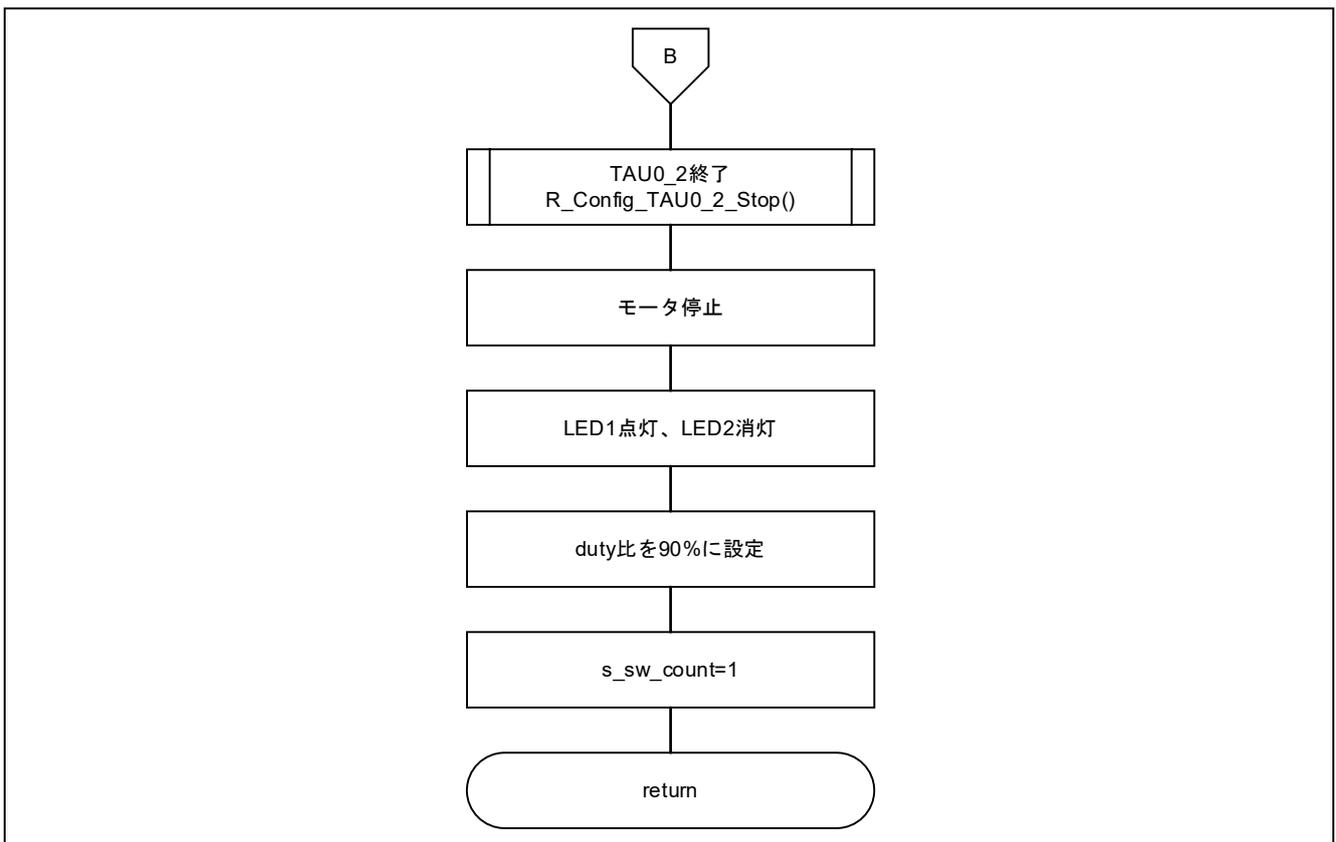


図 4-9 モータの動作状態変更 (3/5)

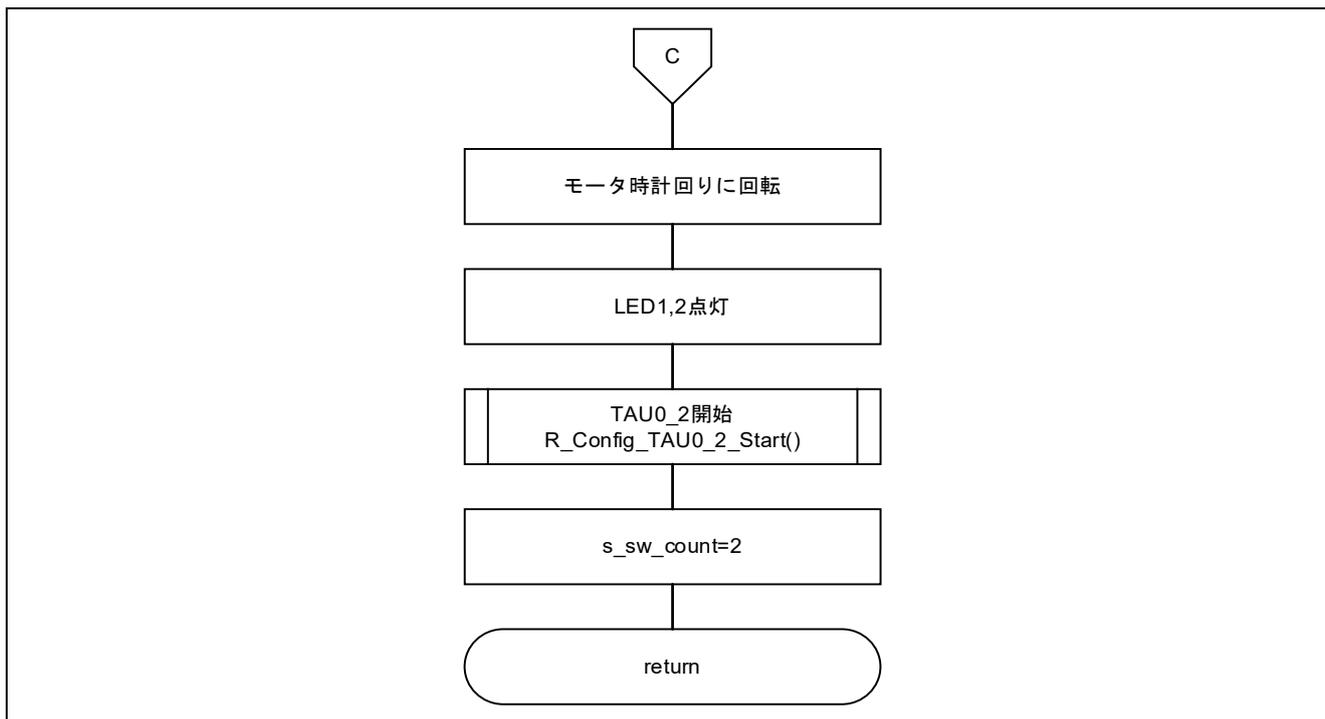


図 4-10 モータの動作状態変更 (4/5)

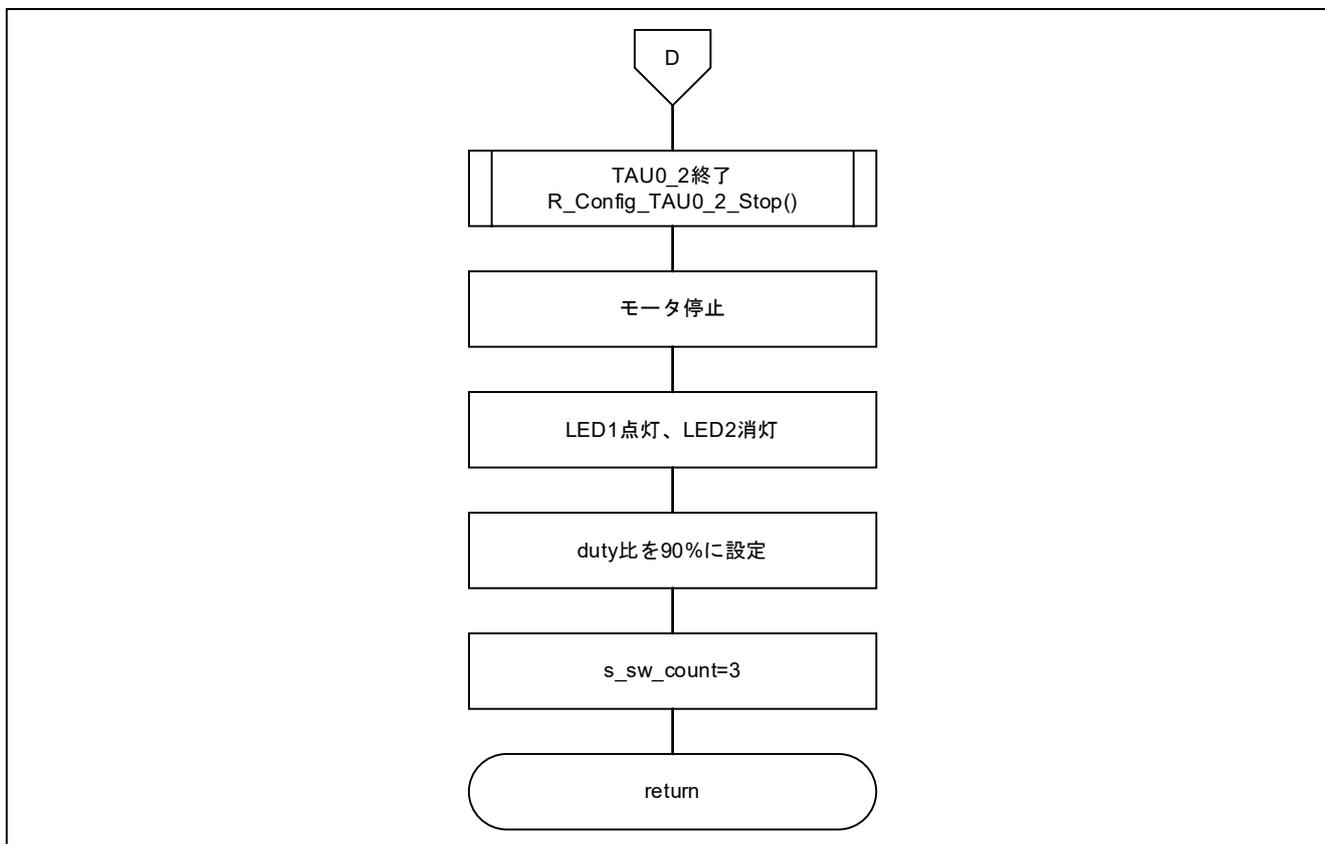
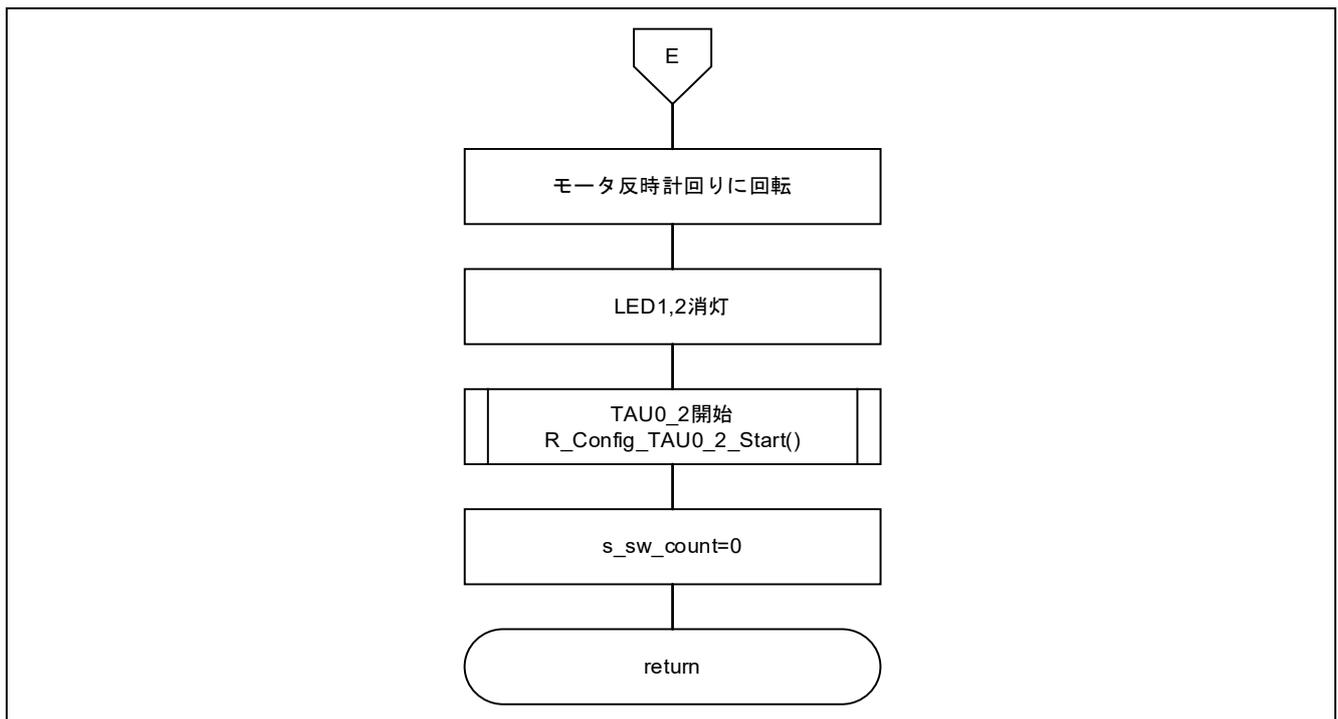


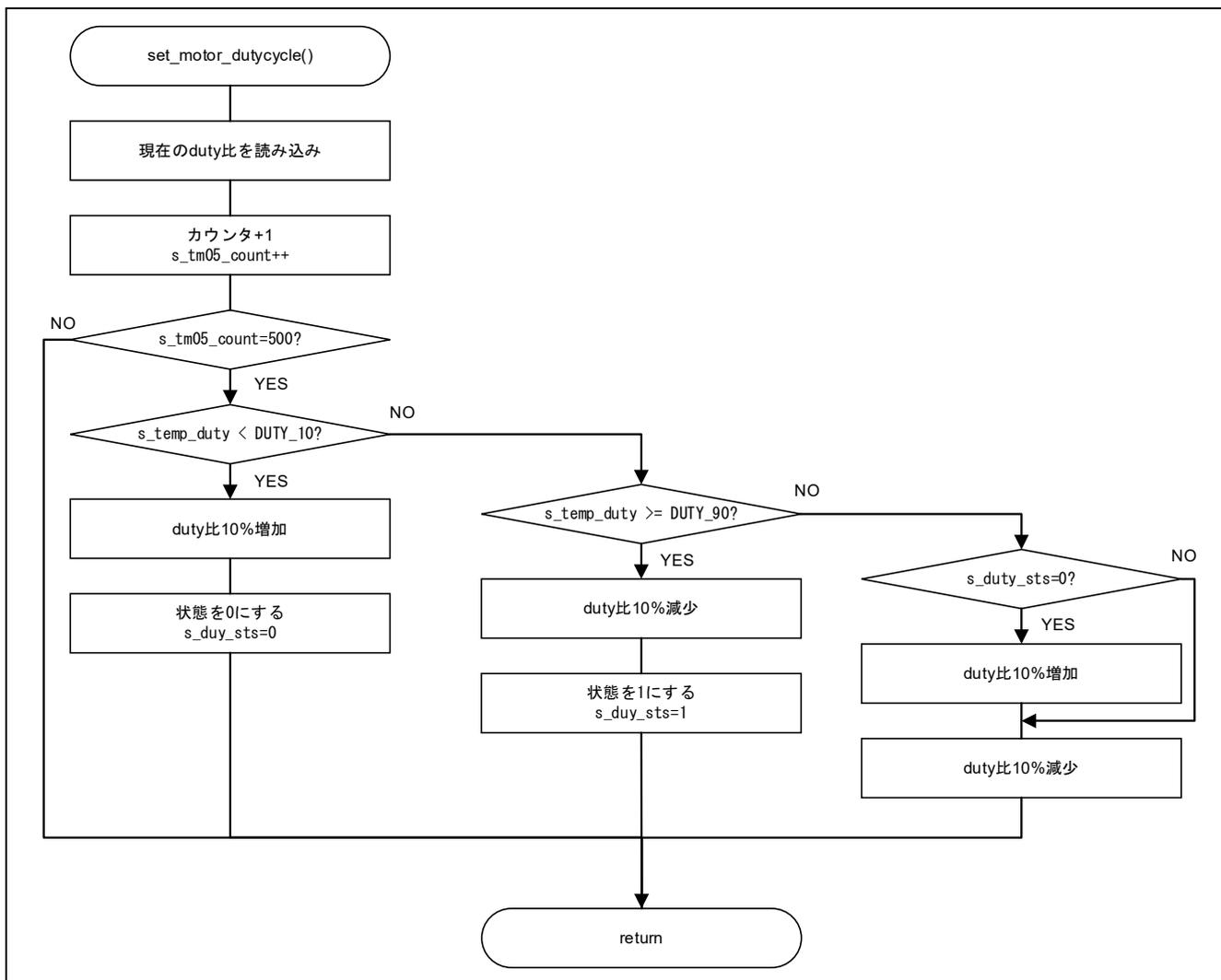
図 4-11 モータの動作状態変更 (5/5)



4.8.5 モータのデューティ比を設定

図 4-12 にモータのデューティ比設定のフローチャートを示します。

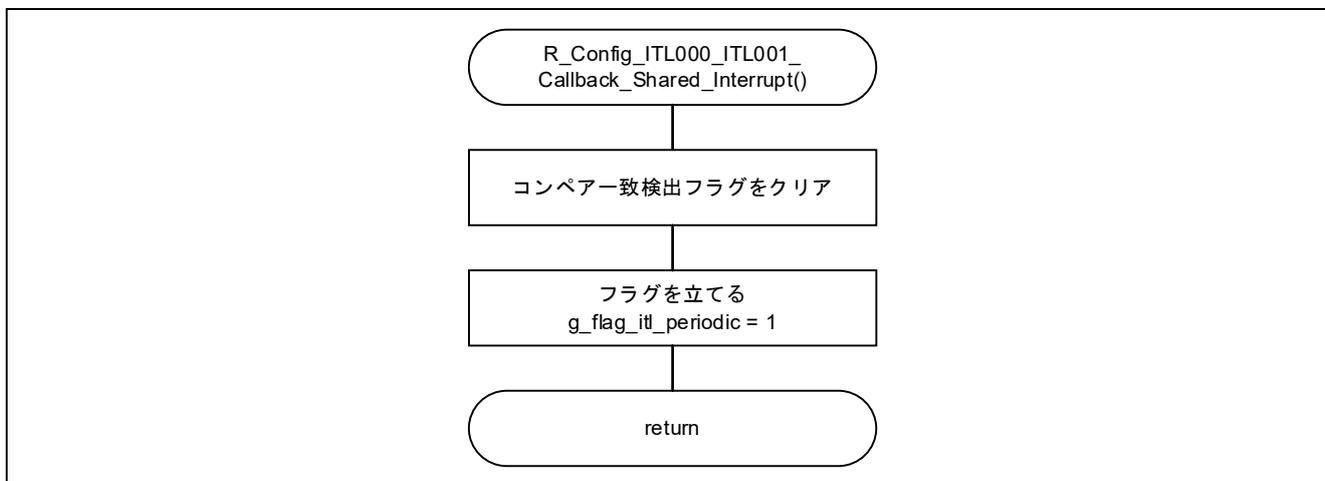
図 4-12 モータのデューティ比設定



4.8.6 インターバル・タイマ割り込み

図 4-13 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。

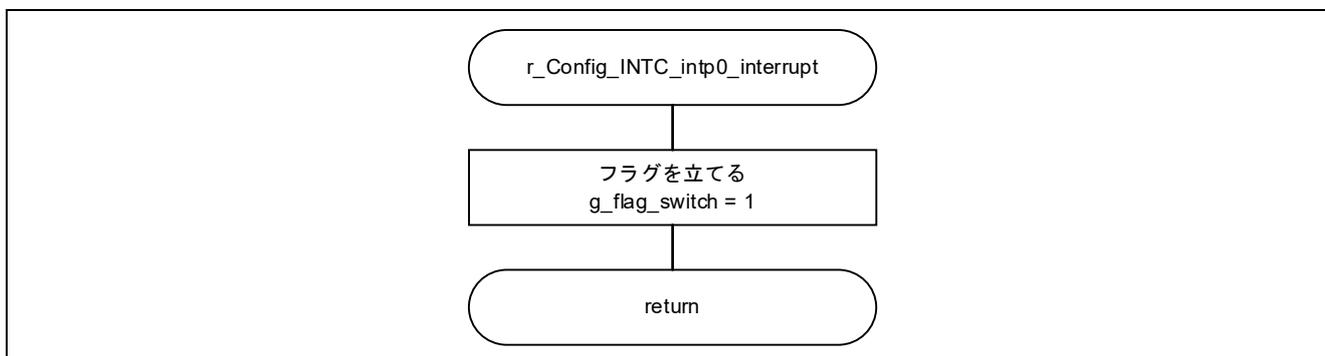
図 4-13 インターバル・タイマ割り込み



4.8.7 スイッチ押下検知割り込み

図 4-14 にスイッチ押下検知割り込みのフローチャートを示します。

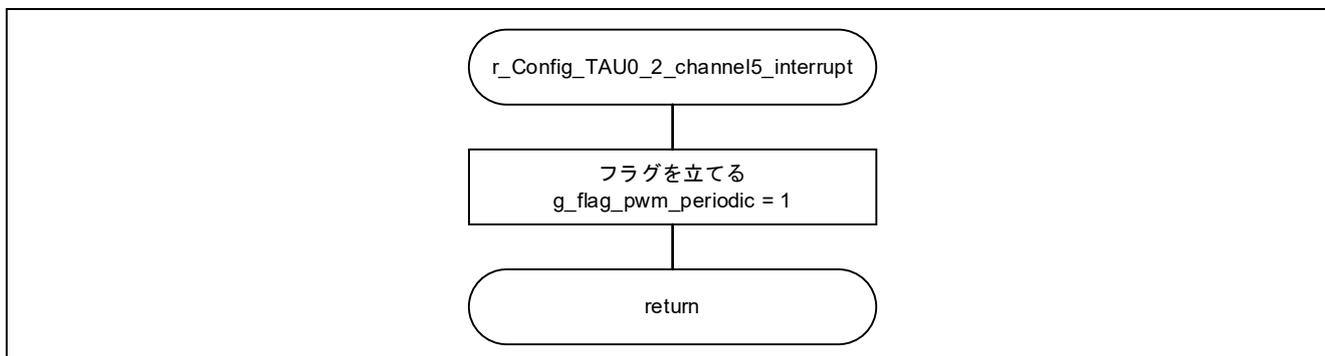
図 4-14 スイッチ押下検知割り込み



4.8.8 PWM 機能のタイマ割り込み

図 4-15 に PWM 機能のタイマ割り込みのフローチャートを示します。

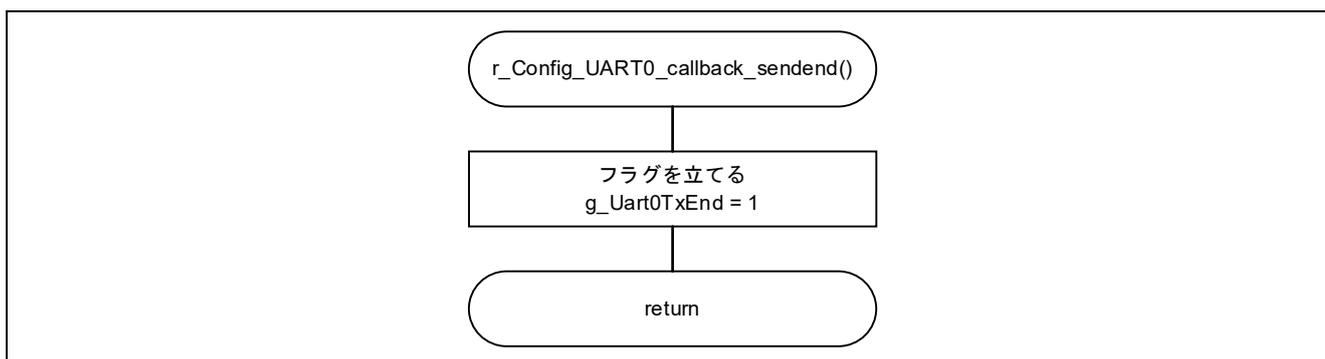
図 4-15 PWM 機能のタイマ割り込み



4.8.9 UART0 送信完了

図 4-16 に UART0 送信完了のフローチャートを示します。

図 4-16 UART0 送信完了



4.9 モータ制御

表 4-7 にモータを制御する端子と動作について示します。

表 4-7 モータを制御する端子と動作

P05(AIN1)	P06(AIN2)	AOUT1	AOUT2	動作
0	0	HiZ	HiZ	空転
0	1	L	H	逆転
1	0	H	L	正転
1	1	L	L	ブレーキ

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。

r01an7635_elcl.scfg

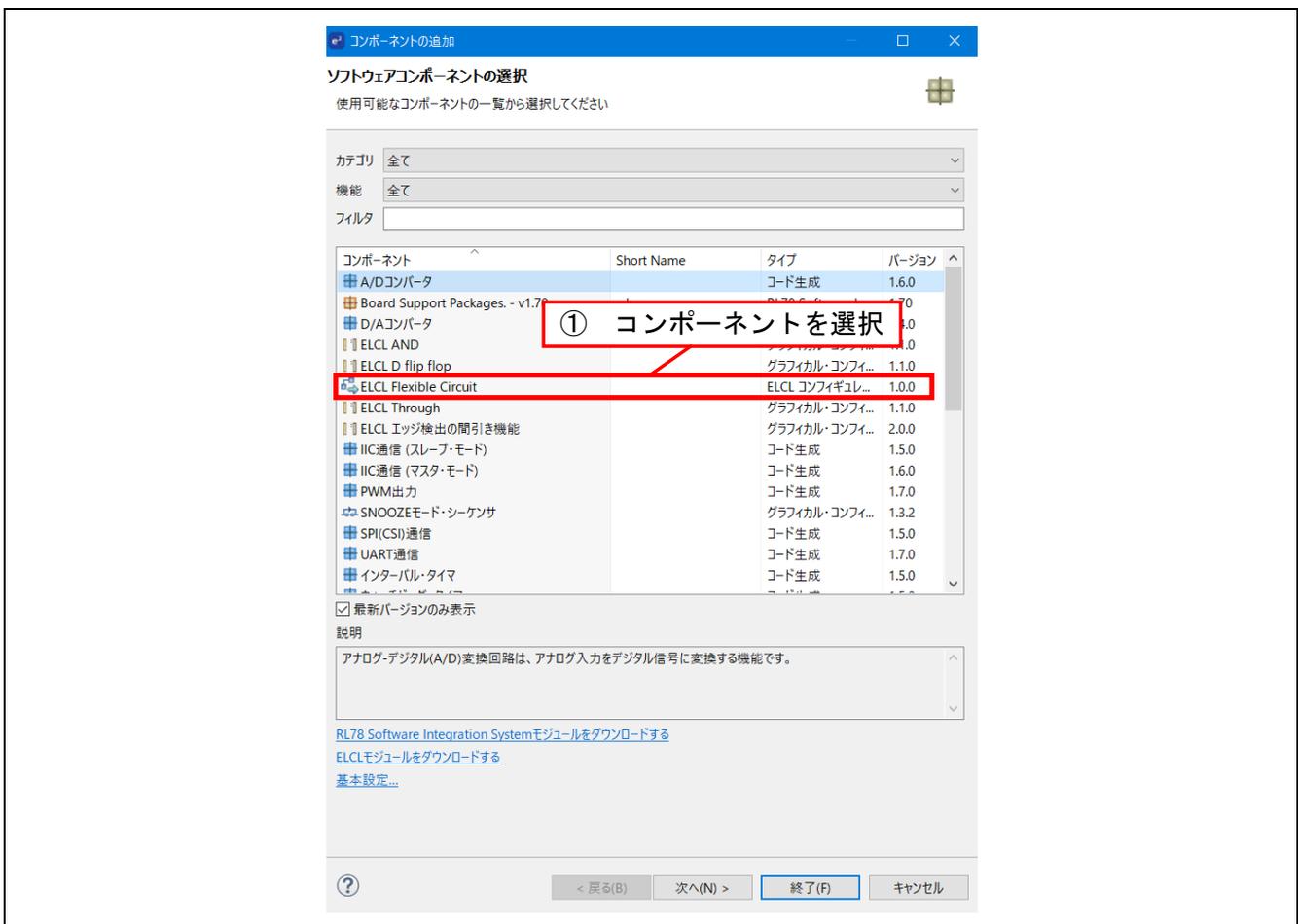
ファイルの説明と使用する上での設定例および注意事項を以下に示します。

5.1 ELCL のコンポーネントの設定

ELCL のコンポーネントを設定する手順を以下に示します。

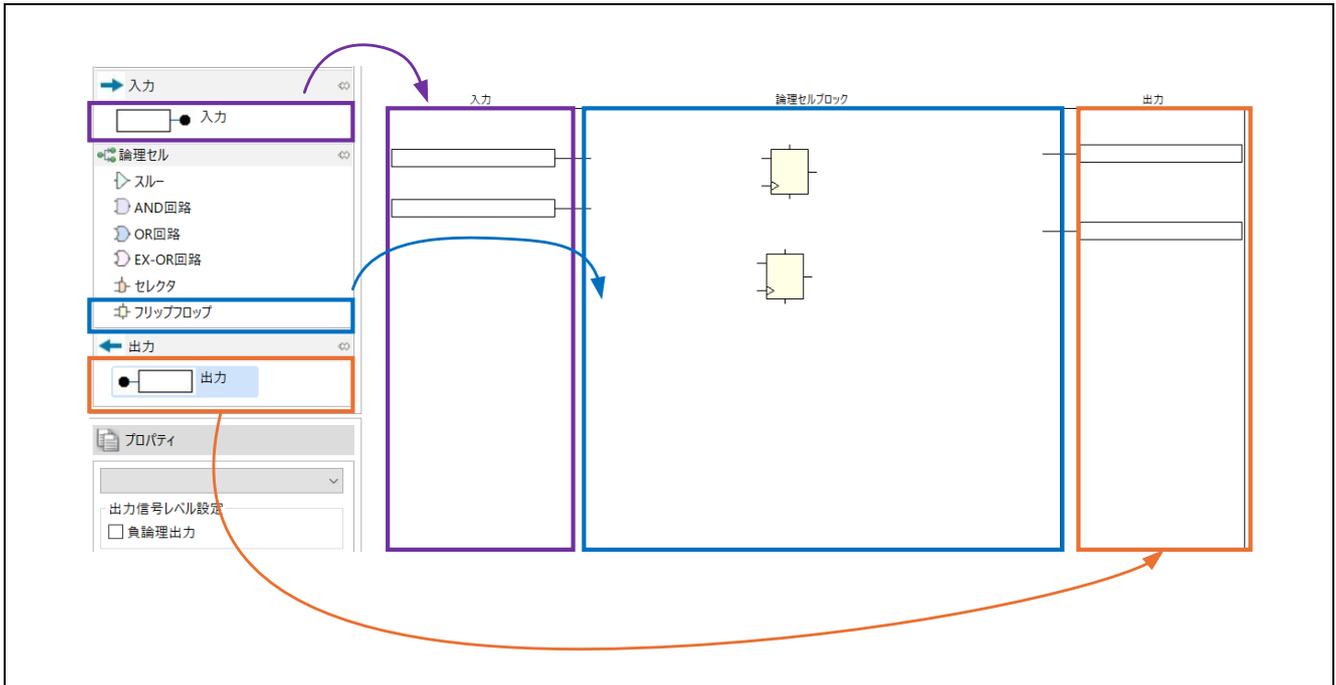
1. 「ELCL Flexible Circuit」を選択してください。

図 5-1 コンポーネントの選択



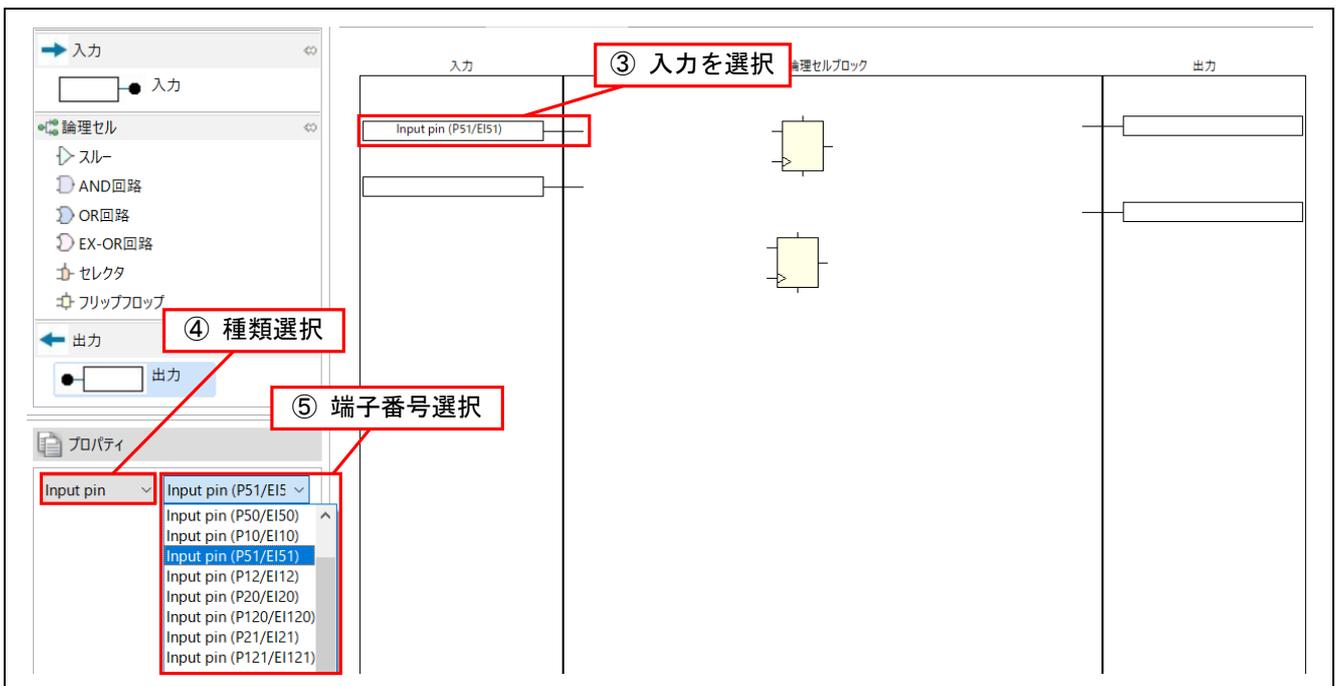
2. 左側にある入力、フリップフロップ、出力を右側の対応するエリアにドラッグ&ドロップします。

図 5-2 ブロックの選択



3. 入力エリアのプロパティ設定したい入力をクリックして選択してください。
4. 入力先の種類を選択してください。
5. 入力先の端子番号を選択してください。

図 5-3 入力エリアのプロパティ設定（詳細は表 5-5 プロパティの設定値）



6. 論理セルブロックエリアのプロパティ設定したいフリップフロップをクリックして選択してください。
7. リソースを選択してください。
8. イベント信号出力レベル設定のリセットのみ負論理出力にチェックしてください。

図 5-4 論理セルブロックエリアのプロパティ設定 (1/2) (詳細は表 5-5 プロパティの設定値)

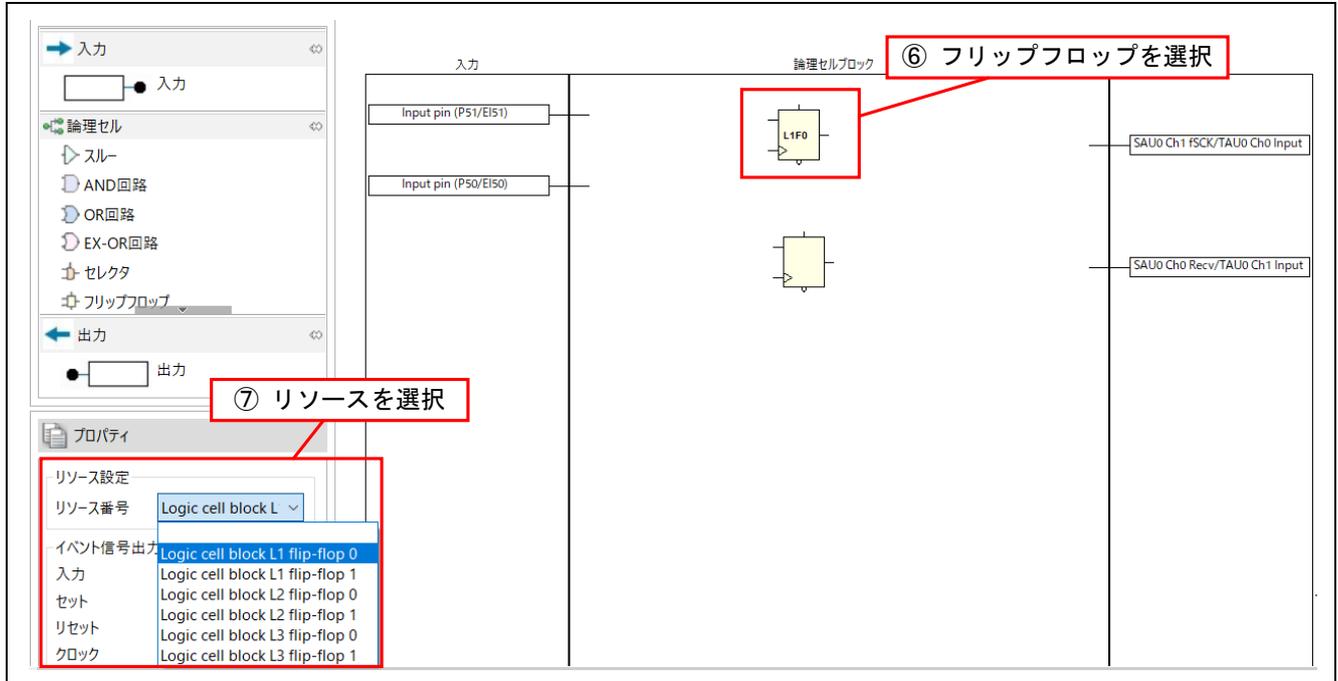
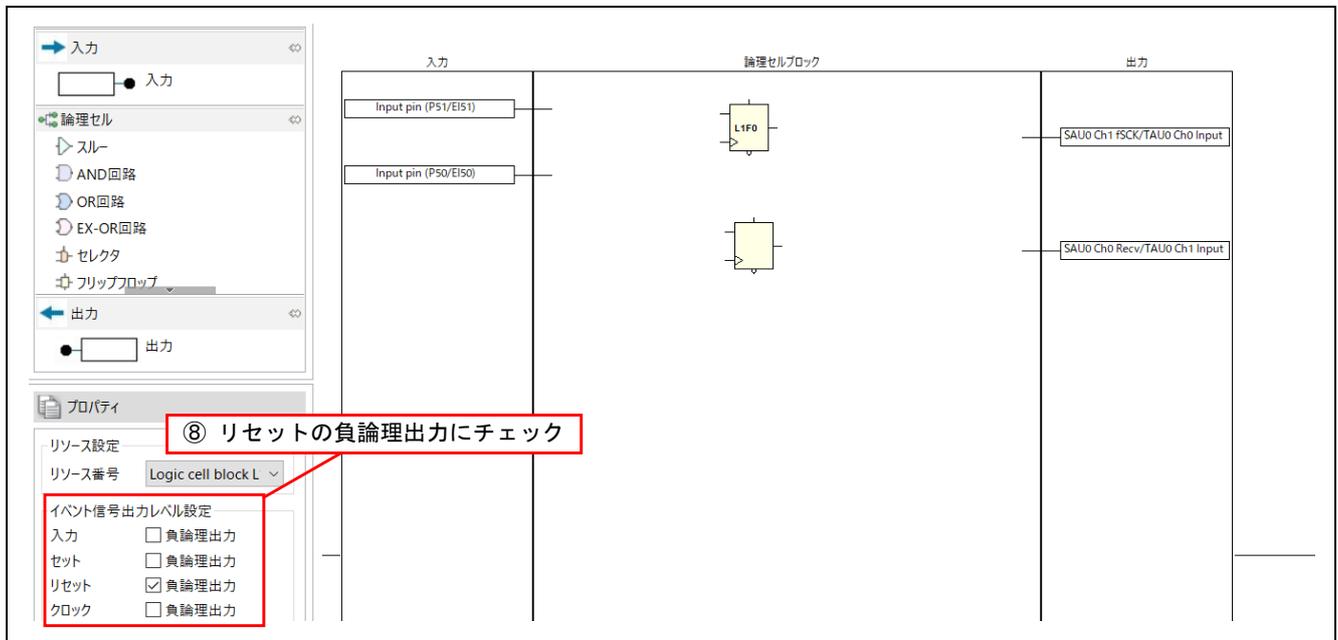
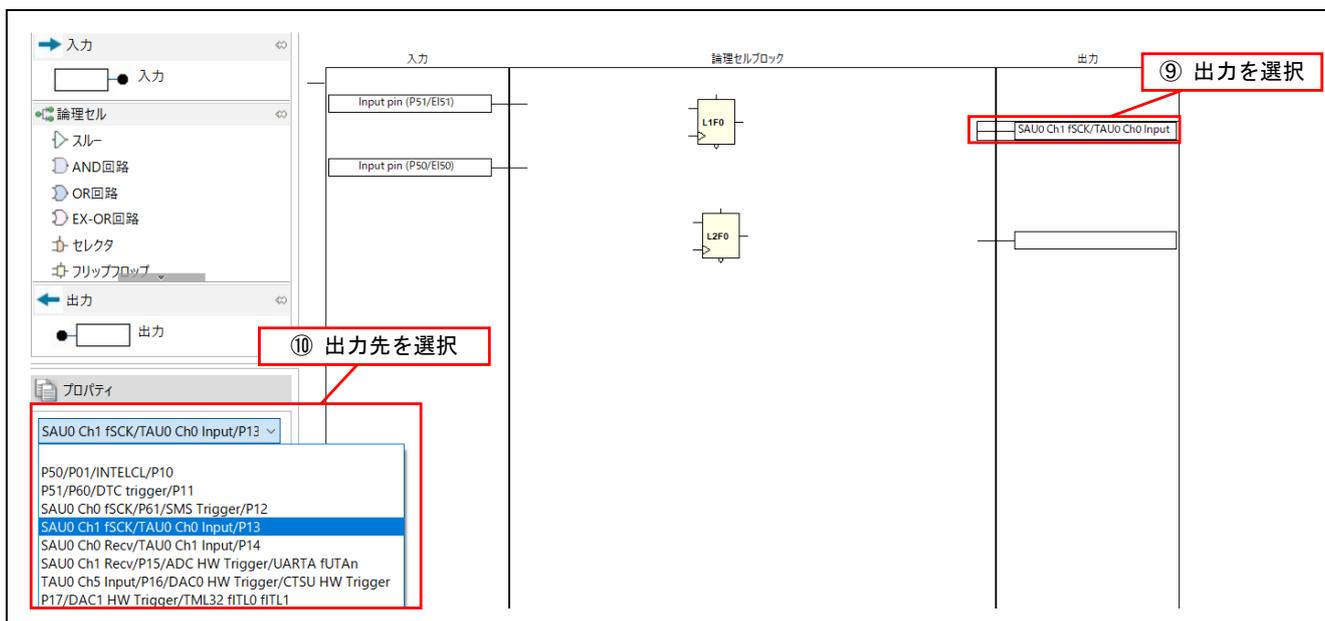


図 5-5 論理セルブロックエリアのプロパティ設定 (2/2)



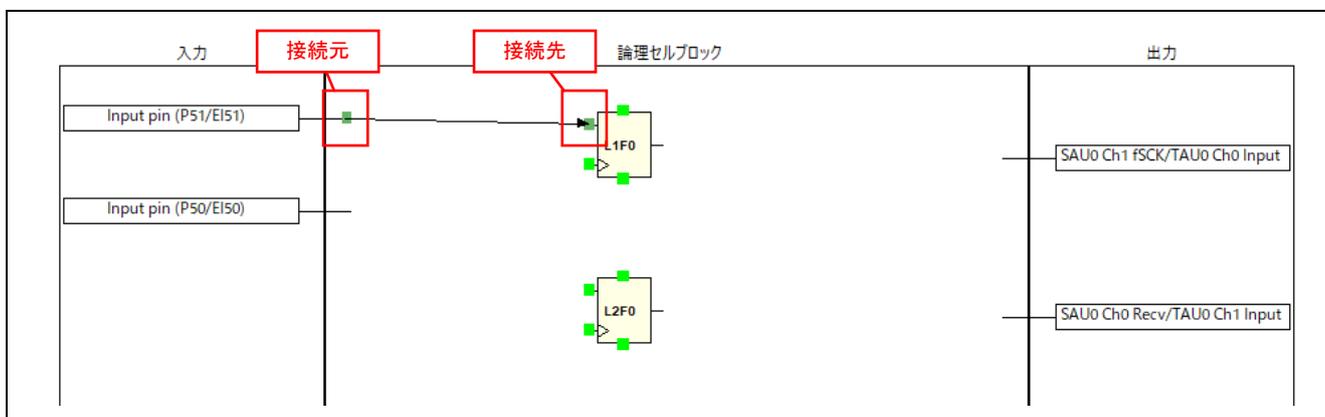
9. 出力エリアのプロパティ設定したい出力をクリックして選択してください。
10. 出力先を選択してください。

図 5-6 出力エリアのプロパティ設定（詳細は表 5-5 プロパティの設定値）



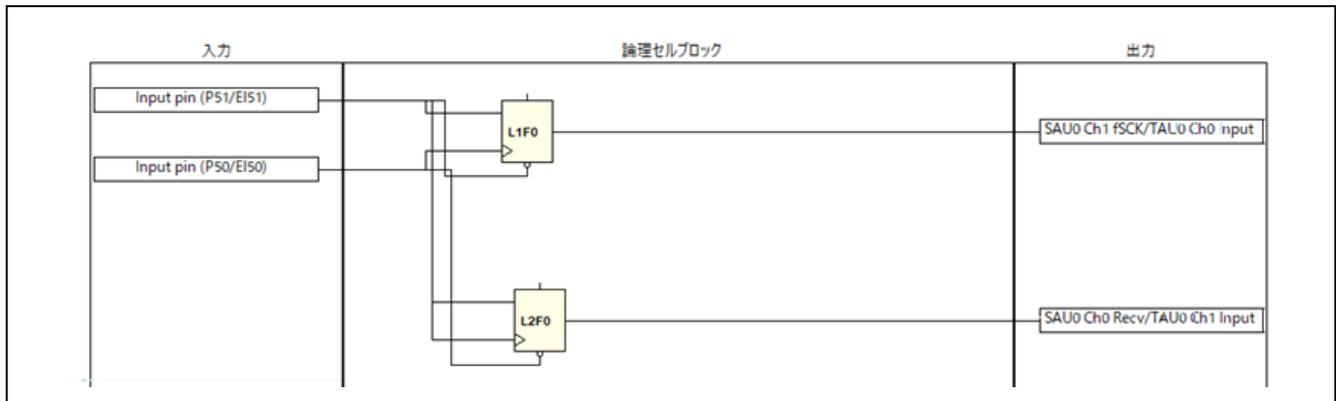
11. 入力、フリップフロップ、出力を接続してください。接続元にカーソルを合わせると緑色の点が出ます。緑色の点をクリックし、接続先までカーソルを引っ張って離します。

図 5-7 ブロックの接続（詳細は表 5-6 各ブロックの接続設定）



12. 本アプリケーションノートでの設定完了イメージ図を図 5-8 に示します。

図 5-8 設定完了イメージ図



5.2 r01an7635_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 (1/4)

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：0x000000000000000000000000 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

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 (2/4)

タグ名	コンポーネント	内容
コンポーネント	r_bsp	Start up select : Enable (use BSP startup) Control of illicit 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
	Config_INTC	コンポーネント : 割り込みコントローラ リソース : INTC INTP0 設定 : INTP0 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	コンポーネント : インターバル・タイマ 動作 : 16 ビット・カウンタ・モード リソース : ITL000_ITL001 動作クロック (f _{ITL0}) : f _{SXP} クロック・ソース : f _{ITL0} インターバル時間 : 400ms 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	コンポーネント : 外部イベント・カウント リソース : TAU0_0 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(TI00) : 立ち上がりエッジ カウント値 : 1 割り込み設定 : 使用しない

表 5-3 スマート・コンフィグレータの設定値 (3/4)

タグ名	コンポーネント	内容
	Config_TAU0_1	コンポーネント：外部イベント・カウント リソース：TAU0_1 動作クロック：CK00 クロック・ソース：f _{CLK} 入力ソース設定：ELCL 動作モード設定：16 ビット 外部イベント・エッジ選択(TI01)：立ち上がりエッジ カウント値：1 割り込み設定：使用しない
コンポーネント	Config_TAU0_2	コンポーネント：PWM 出力 リソース：TAU0_2 動作クロック：CK00 クロック・ソース：f _{CLK} 周期設定：2ms（モータのデータシートから決定） 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 5 スレーブ、チャンネル 6 スレーブ (スレーブ 5) デューティ：90% 初期出力値：0 出力レベル：アクティブ・ハイ 割り込み設定：使用する 優先順位：レベル 3（低優先順位） (スレーブ 6) デューティ：90% 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
	Config_PORT	コンポーネント：ポート リソース：PORT ポート選択：PORT5 P52：出力（0 を出力） P53：出力（0 を出力） ポート・モード設定：Pmn レジスタ値を読みだす

表 5-4 スマート・コンフィグレータの設定値 (4/4)

タグ名	コンポーネント	内容
コンポーネント	Config_UART0 ^注	コンポーネント : UART 通信 リソース : UART0 動作 : 送信 動作クロック : CK00 クロック・ソース : f _{CLK} 転送モード設定 : シングル転送モード データ・ビット長設定 : 8 ビット データ転送方向設定 : LSB パリティ設定 : パリティ・ビットなし ストップビット長 : 1 ビット 転送レート設定 : 153600bps 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
	Config_ELCL	詳細は 5.2.10Config_ELCL に記載

注. CS+プロジェクトおよび e2 studio プロジェクトではデバッグに COM ポート接続を選択しています。そのため、デバッグで使用する通信線とシリアル通信(UART0)の通信線が競合し、スマート・コンフィグレータ上で×が表示されます。本サンプルでは、デバッグとシリアル通信を同時に使用しないことで問題は発生しません。(USB-シリアル変換経由でデバッグする際は必ず Tera Term を切断してください。また、Tera Term でシリアル通信をする場合は、必ずデバッグを終了しデバッグを切断してください。)

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、パルスカウント機能として、外部イベント・カウンタ機能を使用します。割り込みは使用しません。

5.2.5 Config_TAU0_1

サンプルコードの TAU01 の設定を行います。

サンプルコードでは、パルスカウント機能として、外部イベント・カウンタ機能を使用します。割り込みは使用しません。

5.2.6 Config_TAU0_2

サンプルコードの TAU02 の設定を行います。

サンプルコードでは、モータのスピード制御用の 2 つの PWM 出力を設定します。割り込みベクタ処理にて、デューティ比変更タイミングを計る処理のトリガとなるフラグをセットします。

5.2.7 Config_INTC

サンプルコードの INTP0 の設定を行います。

サンプルコードでは、スイッチ押下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、スイッチを押下した際の処理のトリガとなるフラグをセットします。

5.2.8 Config_ITL000_ITL001

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、1 分間の回転数概算タイミグ、概算結果送信タイミグを生成するため 400ms 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。

5.2.9 Config_UART0

サンプルコードの SAU00 の設定を行います。

本サンプルコードでは、UART 通信で COM ポートを通じて Tera Term へ送信しています。

5.2.10 Config_ELCL

サンプルコードの ELCL の設定を行います。

本サンプルコードでは L1F0, L2F0 を使用し、直交エンコーダ信号を入力信号として選択します。直交エンコーダ信号の A 相を入力信号 0, 直交エンコーダ信号の B 相を入力信号 1 とします。

図 5-9 ELCL 設定

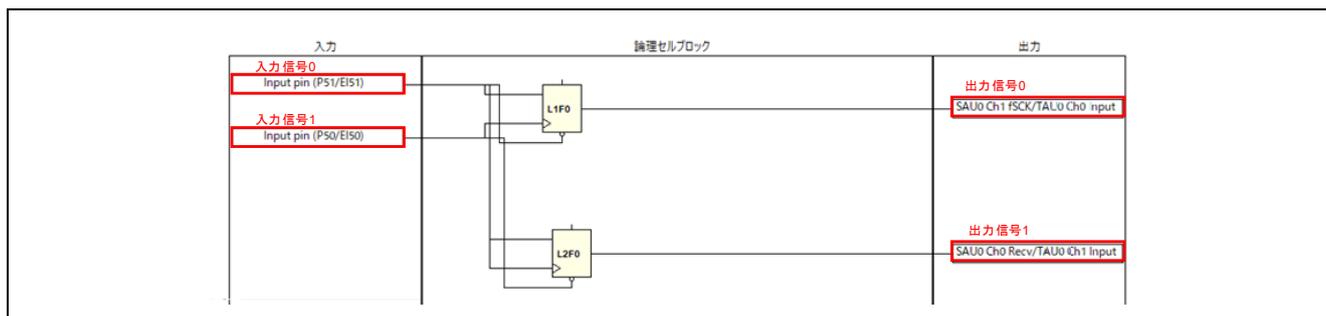


表 5-5 プロパティの設定値（設定方法は図 5-3 から図 5-6 に記載）

項目	プロパティ
入力信号 0	P51
入力信号 1	P50
L1F0	リセット：負論理出力
L2F0	リセット：負論理出力
出力信号 0	TAU0 Ch0 Input
出力信号 1	TAU0 Ch1 Input

表 5-6 各ブロックの接続設定（設定方法は図 5-7 に記載）

接続元	接続先
入力信号 0	L1F0 の入力
	L1F0 のリセット
	L2F0 のクロック
入力信号 1	L1F0 のクロック
	L2F0 の入力
	L2F0 のリセット
L1F0	出力信号 0
L2F0	出力信号 1

5.3 エンコーダとモータを変更する場合

カウント数から 1 分間の回転数を概算する calc_rpm 関数では、エンコーダの分解能とモータのギア比を考慮しています。また、0.4 秒間のカウント数が大きい場合、カウント数の上限を変更する必要があります。

本サンプルコードは以下の設定をしています。

使用するエンコーダとモータを変更する場合は以下の定数及び変数の値を変更してください。

表 5-7 エンコーダ・モータ設定

変数名	値	内容
RESOLUTION ^注	0x00000040	エンコーダの分解能の 1/4
GEAR_RATIO ^注	0x0000001E	モータのギア比
g_max_count	0x2710	カウント数の最大値

注. オーバーフロー抑止のため、値は 32 ビットにしてください。

6. 動作確認時の Tera Term の使用方法

本アプリケーションノートは、カウント結果の確認に Tera Term を使用しています。
Tera Term を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 Tera Term の設定

Tera Term を使用する上での設定例を以下に示します。

表 6-1 Tera Term 設定

名称	設定値
ポート	使用している COM ポートの番号に変更してください。
スピード	153600
データ	8bit
パリティ	none
ストップビット	1bit
フロー制御	none
改行コード (受信)	CR

注. FPB に搭載されている USB-シリアル変換器は、デバッグで使用する通信線とシリアル通信(UART0)の通信線が競合しています。
そのため、USB-シリアル変換経由でデバッグする際は必ず Tera Term を切断してください。
また、Tera Term でモータの回転数を表示する場合は、必ずデバッグを終了しデバッグを切断してください。

図 6-1 Tera Term 設定画面



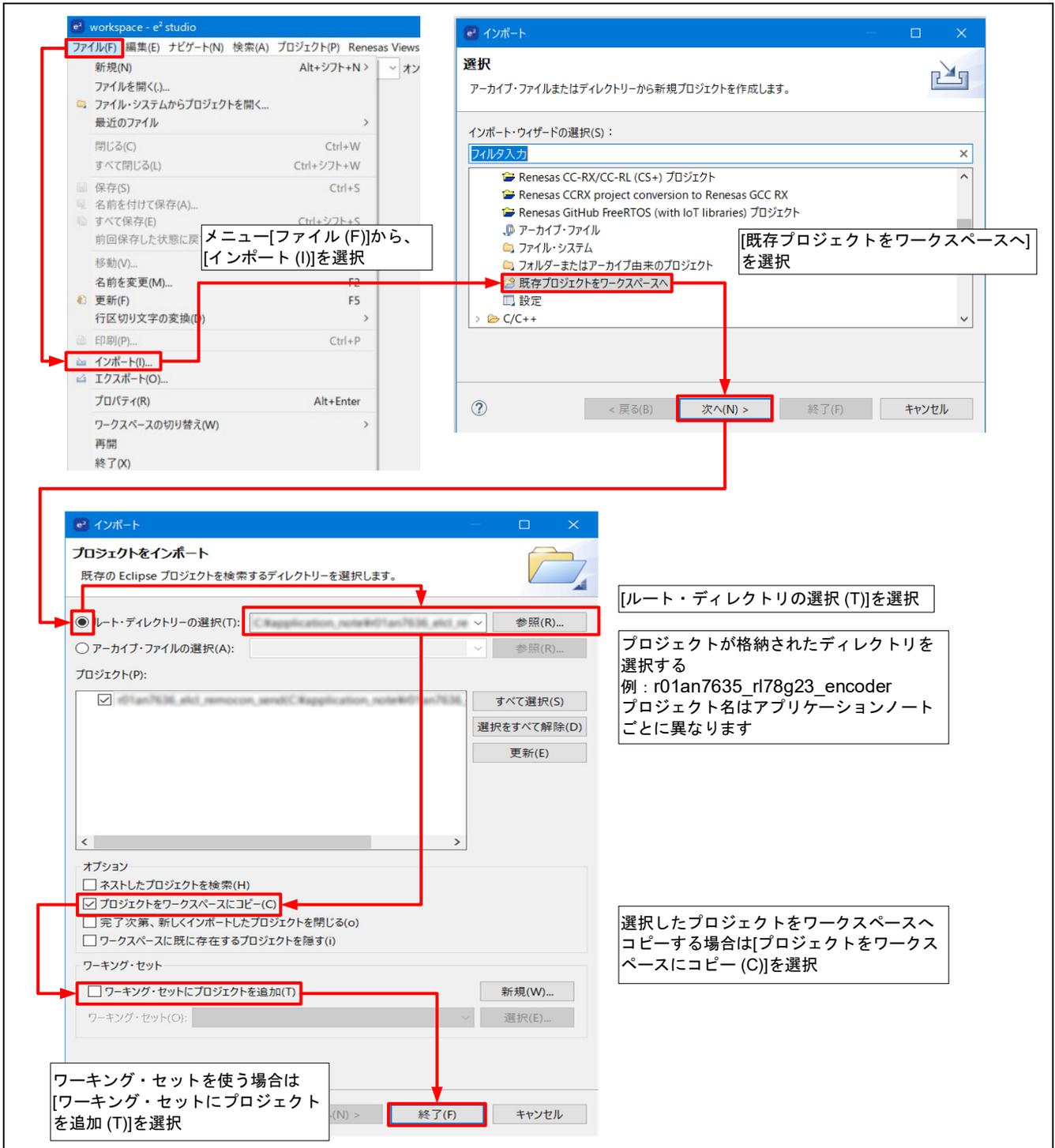
7. プロジェクトのインポート方法

7.1 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 7-1 プロジェクトを e² studio にインポートする方法



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/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)

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	Jun.9.25	-	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。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くと、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。