

## RL78/G23

### ELCL 機能を用いたリモコン信号送信（NEC フォーマット）

---

#### 要旨

本アプリケーションノートでは、タイマ・アレイ・ユニット（TAU）とロジック&イベント・リンク・コントローラ（ELCL）を使用してリモコン信号を送信する方法について説明します。

対象リモコン・フォーマットは NEC フォーマット（ヘッダあり、32 ビットデータ、リピートコードあり）です。

#### 動作確認デバイス

評価ボード : RL78/G23-64p Fast Prototyping Board

赤外線リモコン送信モジュール : Ren He V1221

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	3
1.1 仕様概要	3
1.1.1 システム概要	3
1.1.2 信号フォーマット	3
1.1.3 回路構成	5
1.1.4 ELCL を用いたリモコン信号送信	5
1.2 動作詳細	6
2. 動作確認条件	7
3. ハードウェア説明	8
3.1 ハードウェア構成例	8
3.2 使用端子一覧	8
4. ソフトウェア説明	9
4.1 使用する周辺機能	9
4.2 オプション・バイトの設定一覧	10
4.3 フォルダ構成	11
4.4 変数一覧	12
4.5 定数一覧	13
4.6 関数一覧	14
4.7 関数仕様	15
4.8 フローチャート	16
4.8.1 main 関数のフローチャート	16
4.8.2 invert_and_store 関数のフローチャート	20
4.8.3 tx_data_bit 関数のフローチャート	21
4.8.4 r_Config_TAU0_2_interrupt 関数のフローチャート	24
4.9 応用例	25
4.9.1 スマート・コンフィグレータの設定	25
4.9.1.1 ELCL のコンポーネントの設定	25
4.9.1.2 ELCL Flexible Circuit の使用方法	26
4.9.1.3 r01an7636_elcl_remocon_send.scfg	29
4.9.2 送信データの変更方法	32
5. プロジェクトのインポート方法	33
5.1 e <sup>2</sup> studio での手順	33
6. サンプルコード	34
7. 参考ドキュメント	34

## 1. 仕様

### 1.1 仕様概要

#### 1.1.1 システム概要

本アプリケーションノートでは、タイマ・アレイ・ユニット (TAU) とロジック&イベント・リンク・コントローラ (ELCL) を使用し、リモコン信号送信を行います。

#### 1.1.2 信号フォーマット

リモコン信号は、NEC フォーマットで生成されます。このフォーマットでは、940nm の赤外線キャリア波として使用し、38kHz の周波数とデューティ比 1/3 の信号をサブキャリアとして使用します。送信するデータは、サブキャリアのオンとオフの期間を制御することで符号化されます。

NEC フォーマットでは、最小単位の周期を 562us (1 周期) として、サブキャリアの出力を制御することによりデータを送信します。この信号には、フレームやリピートと呼ばれる構造があり、その中に情報が含まれています。送信する情報は、16 ビットのカスタマーコード (メーカー識別コード) と 8 ビットのデータコードです。

フレームは、リモコン信号の基本的な単位であり、送信するデータを構成する一連の信号の集合です。フレームは、次の要素で構成されています。

1. ヘッダ：転送開始を示す部分。16 周期 ON、8 周期 OFF の信号パターンで構成。
2. カスタマーコード：16 ビットのメーカー識別コード。
3. データコード：8 ビットの送信データと、そのビット反転データを合わせた 16 ビットのデータ。
4. ストップビット：転送終了を示す部分。

カスタマーコードとデータコードに含まれるビットデータは信号の長さによって表現されます。信号送信期間の比が 1:1 の場合は「0」、1:3 の場合は「1」を示します。

リピートは、リモコンのボタンなどを長押しすることで送信される信号です。フレームに続いて、108ms の周期で繰り返し送信されます。これにより、ボタンを押している間、受信側は同じ信号が送信されているとみなします。リピートは、次の要素で構成されています。

1. ヘッダ：転送開始を示す部分。16 周期 ON、4 周期 OFF のパターンで構成。
2. ストップビット：転送終了を示す部分。

NEC フォーマットのリモコン信号を図 1-1 に、ボタン長押し時の信号パターンを図 1-2 に示します。

図 1-1 NEC フォーマットのリモコン信号

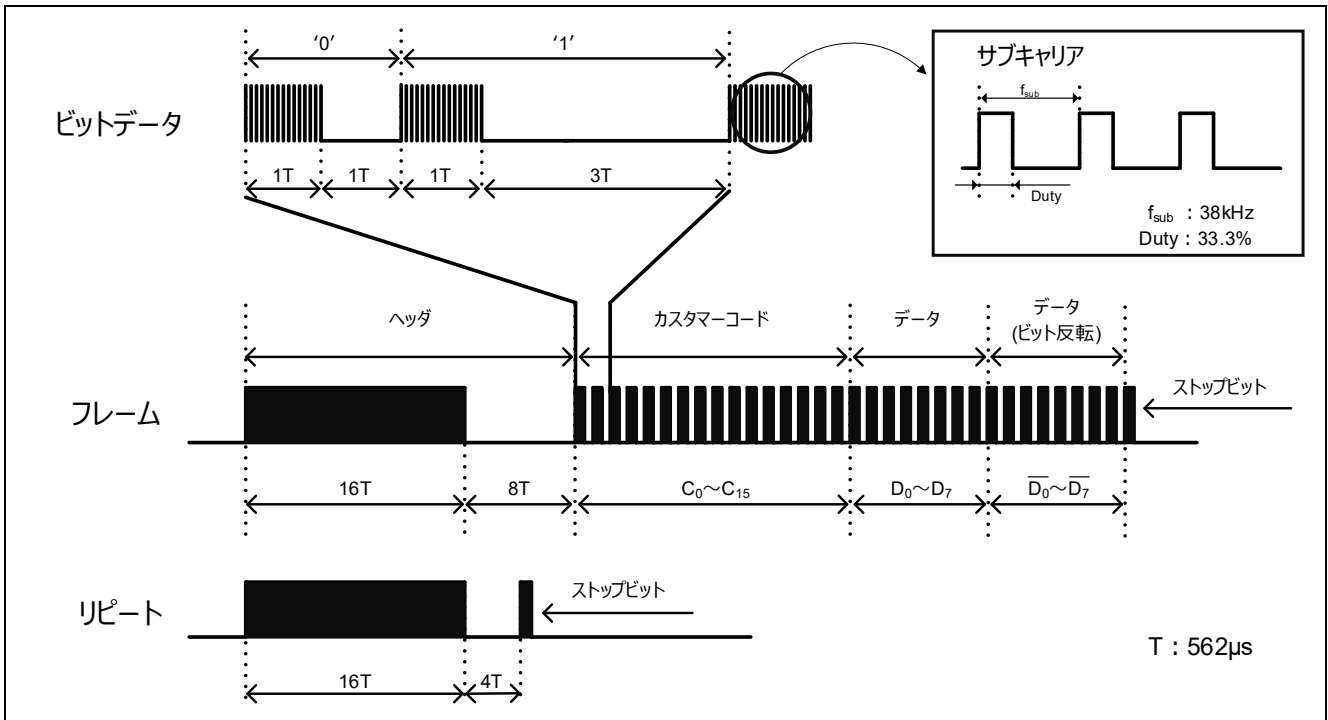
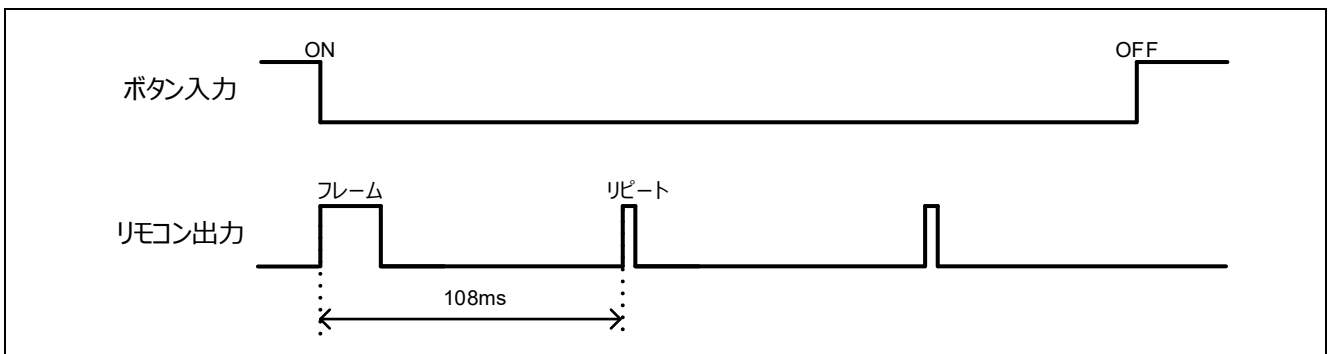


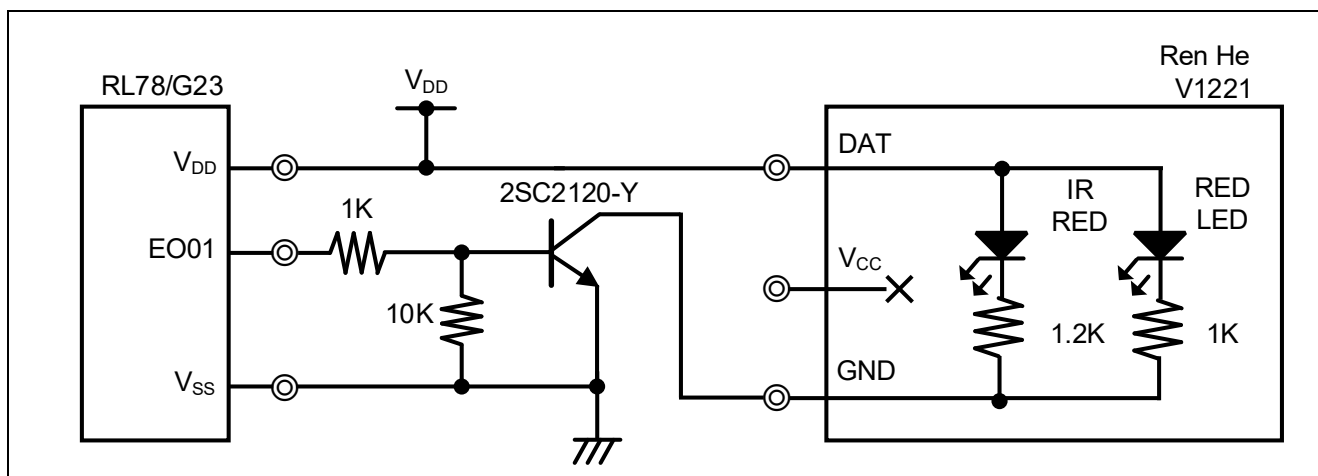
図 1-2 ボタン長押し時の信号パターン



## 1.1.3 回路構成

図 1-3 にリモコン送信時の構成図を示します。

図 1-3 RL78/G23 とリモコン送信モジュールの構成



## 1.1.4 ELCL を用いたリモコン信号送信

図 1-4 に ELCL の構成を示します。

リモコン信号の生成には、タイマ・アレイ・ユニット 0 (TAU0) を使用します。チャンネル 0 (TAU0\_0)、チャンネル 1 (TAU0\_1) では、サブキャリア波形を生成する PWM 出力を行います。チャンネル 2 (TAU0\_2) では、NEC フォーマットの周期  $562\mu\text{s}$  の間隔で、周期の経過をカウントする割り込み処理を行います。TAU0\_2 でカウントした周期を基に、P10 のレベルを制御しマスク波形を生成します。

ELCL では、TAU0\_1 からサブキャリア波形、P51 ピンからマスク波形の入力を受け、それらを AND 演算することで P01 ピンにリモコン信号を出力します。

図 1-4 ELCL の構成

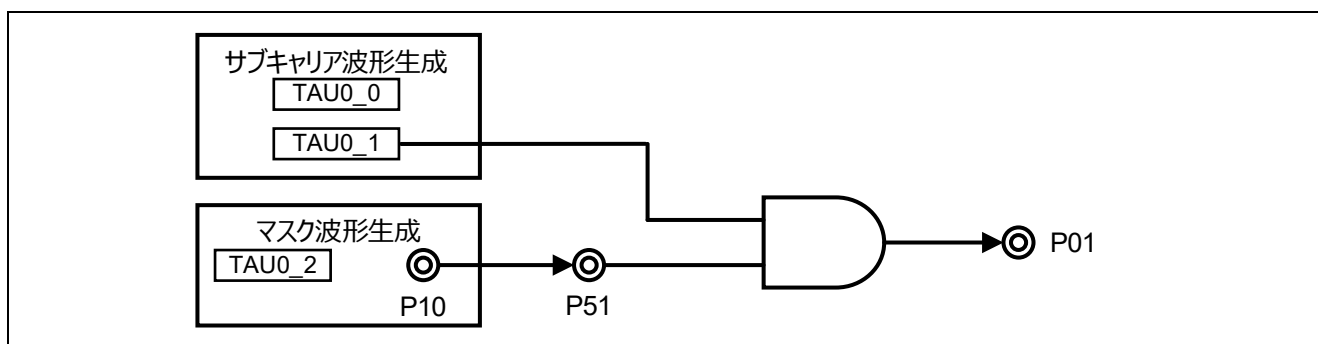
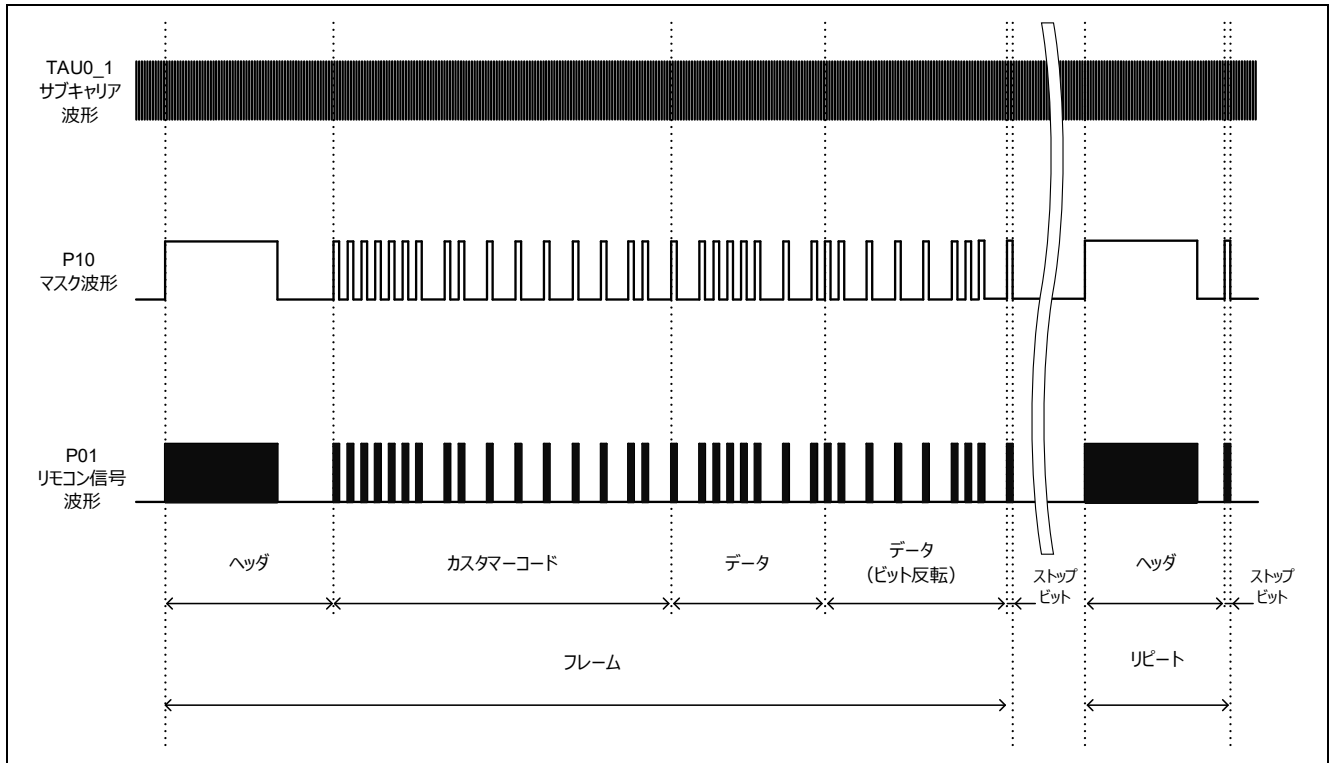


図 1-5 に ELCL を用いたリモコン信号波形を示します。

本アプリケーションノートでは、カスタマーコードを 0x02fd、データコードを 0x86 としてリモコン信号を生成します。

図 1-5 ELCL を用いたリモコン信号波形



## 1.2 動作詳細

本アプリケーションノートでは、リモコン信号送信の使用例を示しています。

ボード上のスイッチ (SW1) を押すことで、リモコン信号を送信します。スイッチを押し続けている間は、リピートコードが繰り返し送信されます。

## 2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2-1 動作確認条件

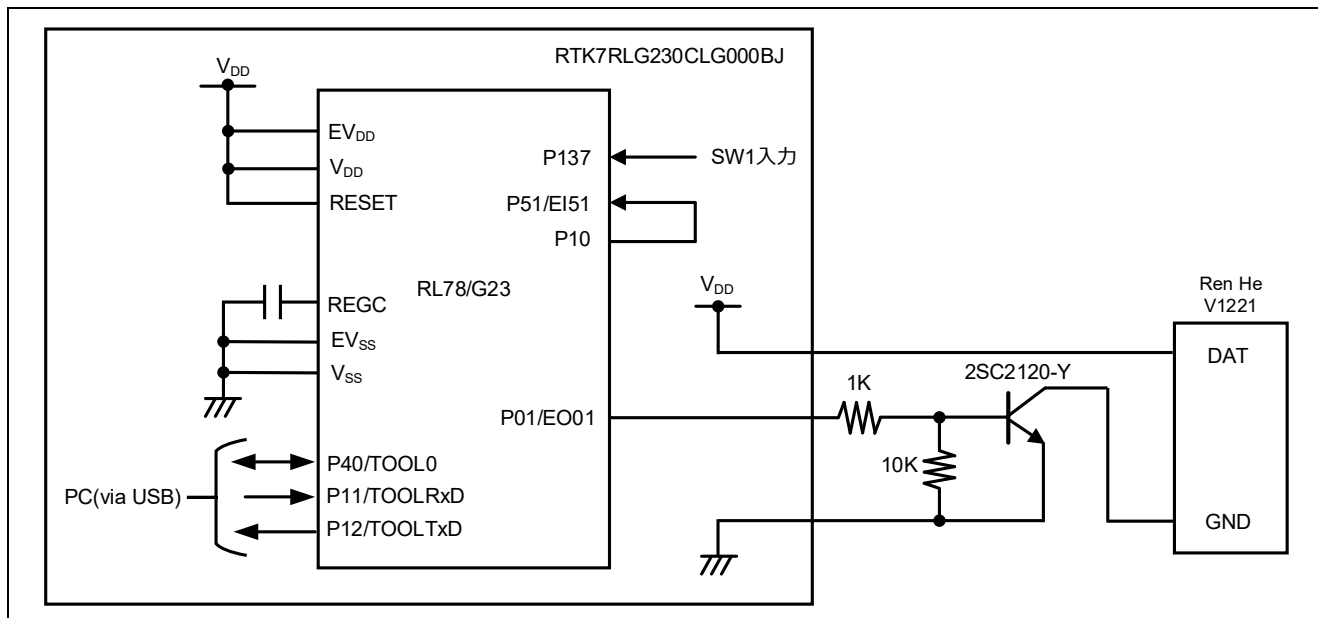
項目	内容
使用マイコン	RL78/G23 (R7F100GLG)
使用ボード	RL78/G23-64p Fast Prototyping Board (RTK7RLG230CLG000BJ)
動作周波数	<ul style="list-style-type: none"> <li>高速オンチップ・オシレータ・クロック : 32MHz</li> <li>CPU/周辺ハードウェア・クロック : 32MHz</li> </ul>
動作電圧	<ul style="list-style-type: none"> <li>5.0V</li> <li>LVD0 動作 (<math>V_{LVD0}</math>) : リセット・モード</li> <li>リセット発生電圧 (<math>V_{LVD0}</math>) : 1.65 (V)</li> </ul>
統合開発環境 (CS+)	ルネサスエレクトロニクス製 CS+ for CC V8.13.0
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 (e <sup>2</sup> studio)	ルネサスエレクトロニクス製 e <sup>2</sup> studio 2025-04 (25.4.0)
C コンパイラ (e <sup>2</sup> 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 IAR Embedded Workbench for Renesas RL78 V 5.20.1
C コンパイラ (IAR)	IAR システム製 IAR C/C++ Compiler for Renesas RL78 5.20.1.2826
スマート・コンフィグレータ	V.1.13.0
ボードサポートパッケージ (r_bsp)	V.1.90
赤外線リモコン送信モジュール	Ren He V1221

### 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
P01/E001	出力	サブキャリア波形と、マスク波形で AND 演算を行い、生成したリモコン信号を出力
TAU0_1	出力	サブキャリア波形を生成し、ELCL に入力
P10	出力	マスク波形を出力
P51/EI51	入力	P10 からマスク波形を入力
P137	入力	SW1 入力

## 4. ソフトウェア説明

### 4.1 使用する周辺機能

本アプリケーションノートのサンプルコードで使用する周辺機能を以下に示します。

表 4-1 周辺機能一覧

周辺機能	用途
タイマ・アレイ・ユニット (TAU0_0)	PWM マスタ (周期設定)
タイマ・アレイ・ユニット (TAU0_1)	PWM スレーブ (デューティ比制御)、PWM 出力
タイマ・アレイ・ユニット (TAU0_2)	NEC フォーマットの周期 (562 $\mu$ s) カウント用
ポート機能	P10 : マスク波形の作成に使用 P137 : SW1 の入力に使用
ロジック&イベント・リンク・コントローラ(ELCL)	PWM スレーブ出力とポート出力の AND 演算を行い、リモコン信号を生成

## 4.2 オプション・バイトの設定一覧

オプション・バイト設定を以下に示します。

表 4-2 オプション・バイト設定

アドレス	設定値	内容
000C0H / 020C0H	1110 1111B (0xEF)	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 / 020C1H	1111 1100B (0xFC)	LVD0 検出電圧 : リセット・モード リセット発生電圧 ( $V_{LVD0}$ ) : 1.65 (V)
000C2H / 020C2H	1110 1000B (0xE8)	HS モード、 高速オンチップ・オシレータ・クロック : 32MHz
000C3H / 020C3H	1000 0100B (0x84)	オンチップ・デバッグ許可

### 4.3 フォルダ構成

サンプルコードで使用しているソースファイル/ヘッダファイルの構成を以下に示します。  
 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 4-3 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
¥ r01an7636_elcl_remocon_send <DIR> <sup>注2</sup>	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	√ <sup>注1</sup>
¥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	√ <sup>注1</sup>
¥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	√ <sup>注1</sup>
¥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	√

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 r01an7636\_elcl\_remocon\_send.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。

#### 4.4 変数一覧

サンプルコードで使用する変数一覧を以下に示します。

表 4-4 変数一覧

型	変数名	内容	ファイル
uint8_t	g_data_bit[4]	送信データを格納する配列	main.c
uint8_t	g_bit_index	現在処理中のビットのインデックス	main.c
uint8_t	g_byte_index	現在処理中のバイトのインデックス	main.c
uint8_t	g_tm02_count	周期の経過を示す変数	main.c Config_TAU0_2_us er.c
uint8_t	g_repeat_count	リピートコード出力用の周期カウント 変数	main.c Config_TAU0_2_us er.c
uint16_t	g_cusutomer_code	カスタマーコードを格納する変数	main.c
uint8_t	g_data_code	データコードを格納する変数	main.c

#### 4.5 定数一覧

サンプルコードで使用する定数一覧を以下に示します。

これらの定数は main.h で定義されています。

表 4-5 定数一覧

定数名	設定値	内容
SW1_PRESSED	0	SW1 が押下された状態を示す値
SIGNAL_HIGH	1	P10 の出力を HIGH に設定
SIGNAL_LOW	0	P10 の出力を LOW に設定
HEADER_HIGH_CYCLES	16	ヘッダの HIGH 周期
HEADER_LOW_CYCLES_FRAME	8	ヘッダの LOW 周期 (フレーム)
HEADER_LOW_CYCLES_REPEAT	4	ヘッダの LOW 周期 (リピート)
CYCLE_HIGH_0	1	ビットデータの HIGH 周期 (0 の場合)
CYCLE_LOW_0	1	ビットデータの LOW 周期 (0 の場合)
CYCLE_HIGH_1	1	ビットデータの HIGH 周期 (1 の場合)
CYCLE_LOW_1	3	ビットデータの LOW 周期 (1 の場合)
STOP_CYCLES	1	ストップビットの HIGH 周期
REPEAT_CYCLES	192	リピートが出力される周期間隔 リピートの間隔 (108ms) を 1 周期 (562 $\mu$ s) で除算して算出

## 4.6 関数一覧

サンプルコードで使用する関数一覧を以下に示します。

表 4-6 関数一覧

関数名	概要	ソースファイル
main	メイン処理	main.c
invert_and_store	データを変換し配列に格納する処理	main.c
tx_data_bit	ビットデータ出力処理	main.c
r_Config_TAU0_2_interrupt	TAU0_2 割り込み	Config_TAU0_2_user.c

## 4.7 関数仕様

サンプルコードの関数仕様を以下に示します。

## [関数名]main

---

概要	メイン処理
ヘッダ	r_smc_entry.h
宣言	int main (void)
説明	スイッチ入力を受けるとリモコン信号を出力する処理を行います。
引数	なし
リターン値	なし
備考	なし

## [関数名] invert\_and\_store

---

概要	データを変換し配列に格納する処理
ヘッダ	main.h
宣言	void invert_and_store(uint16_t customer, uint8_t data)
説明	カスタマーコードとデータコードをリモコン信号用に変換し、g_data_bit[4] に格納します。
引数	customer : カスタマーコード data : データコード
リターン値	なし
備考	なし

## [関数名] tx\_data\_bit

---

概要	ビットデータ出力処理
ヘッダ	main.h
宣言	void tx_data_bit(void)
説明	g_data_bit[4]に格納した値を、NEC フォーマットの規則で出力します。
引数	なし
リターン値	なし
備考	なし

## [関数名] r\_Config\_TAU0\_2\_interrupt

---

概要	TAU0_2 割り込み
ヘッダ	Config_TAU0_2.h
宣言	static void __near r_Config_TAU0_2_interrupt(void)
説明	NEC フォーマットの周期(562μs)毎にカウントする処理を行います。
引数	なし
リターン値	なし
備考	なし

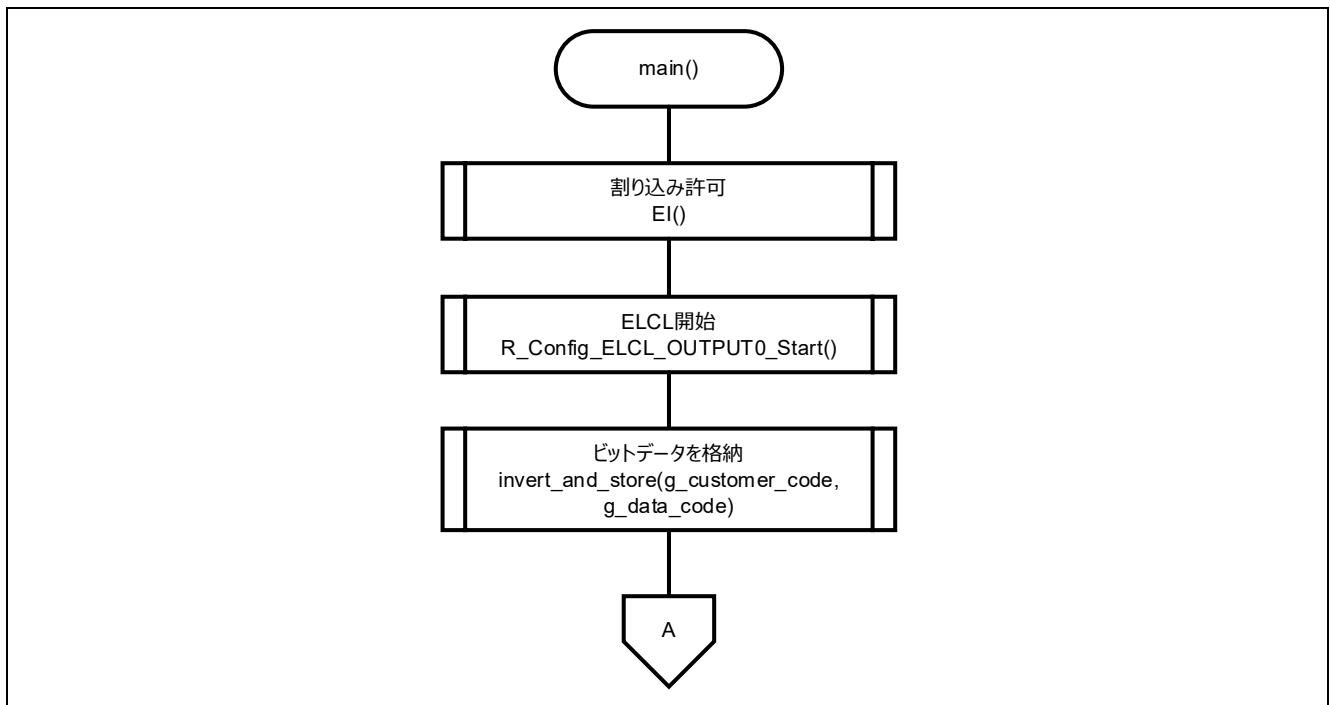
## 4.8 フローチャート

### 4.8.1 main 関数のフローチャート

main 関数のフローチャートを以下に示します。

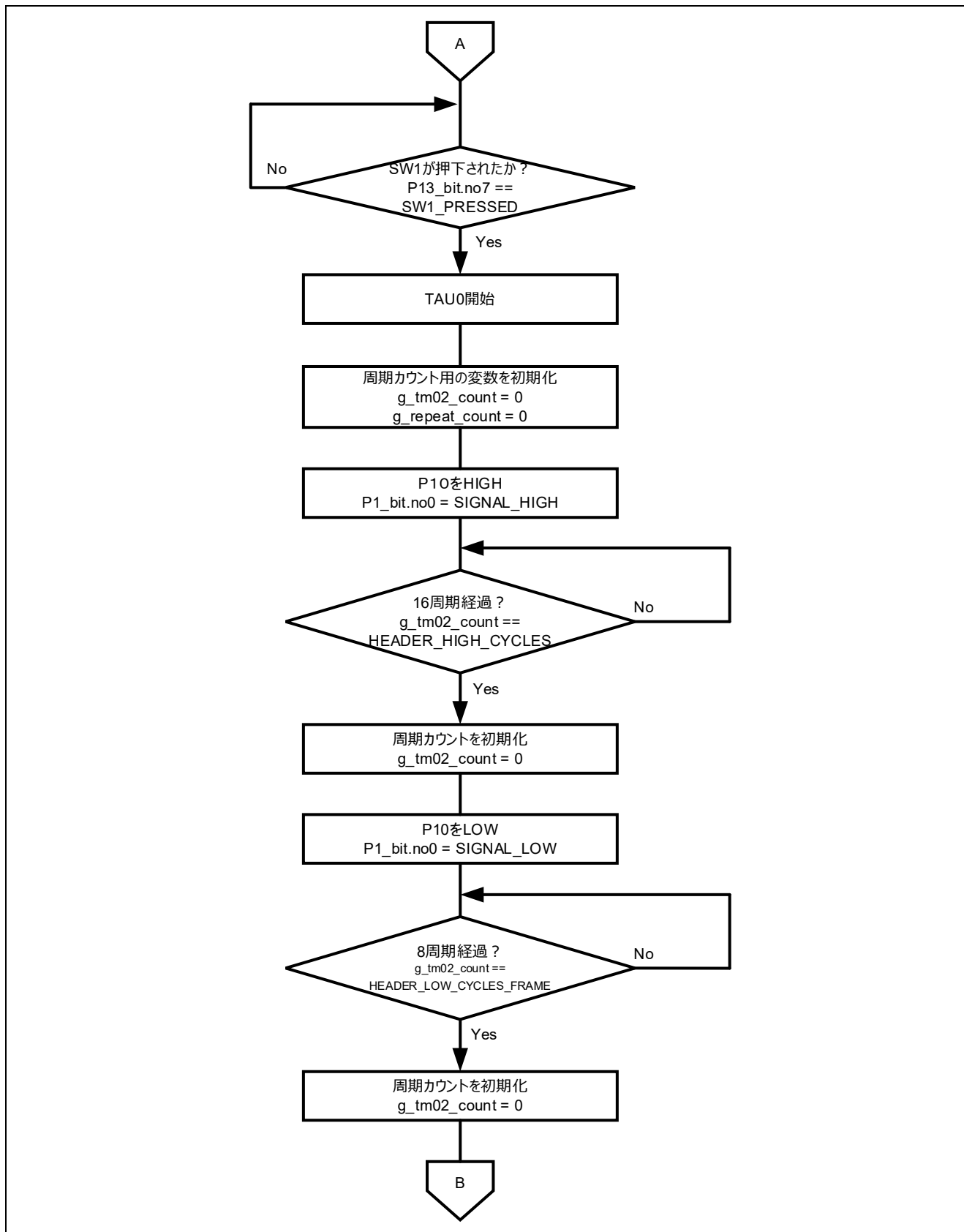
始めにプログラムの初期化を行います。

図 4-1 main 関数フローチャート (1/4)



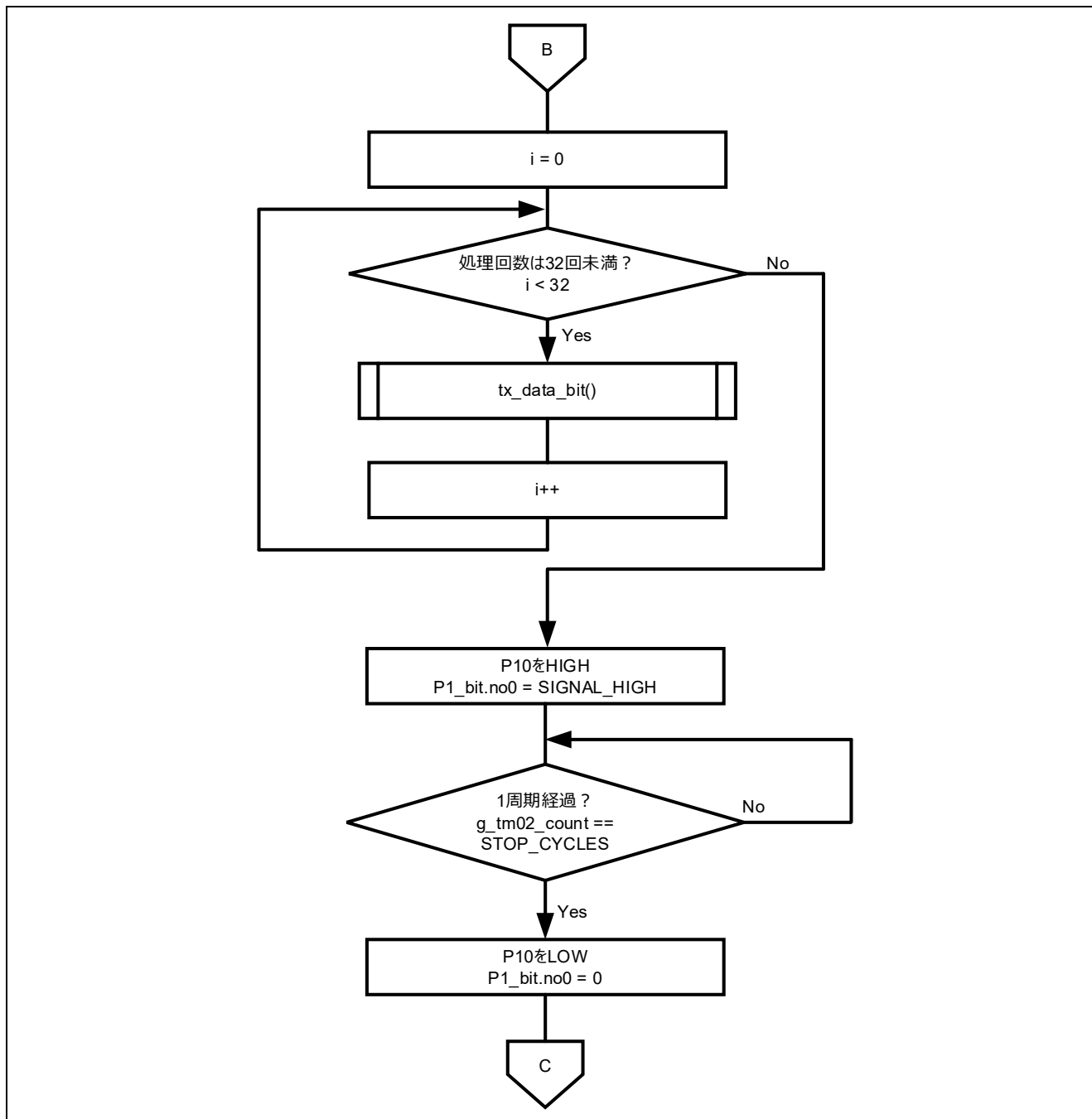
SW1 の押下を検出し、フレームのヘッダを送信する処理を行います。

図 4-2 main 関数フローチャート (2/4)



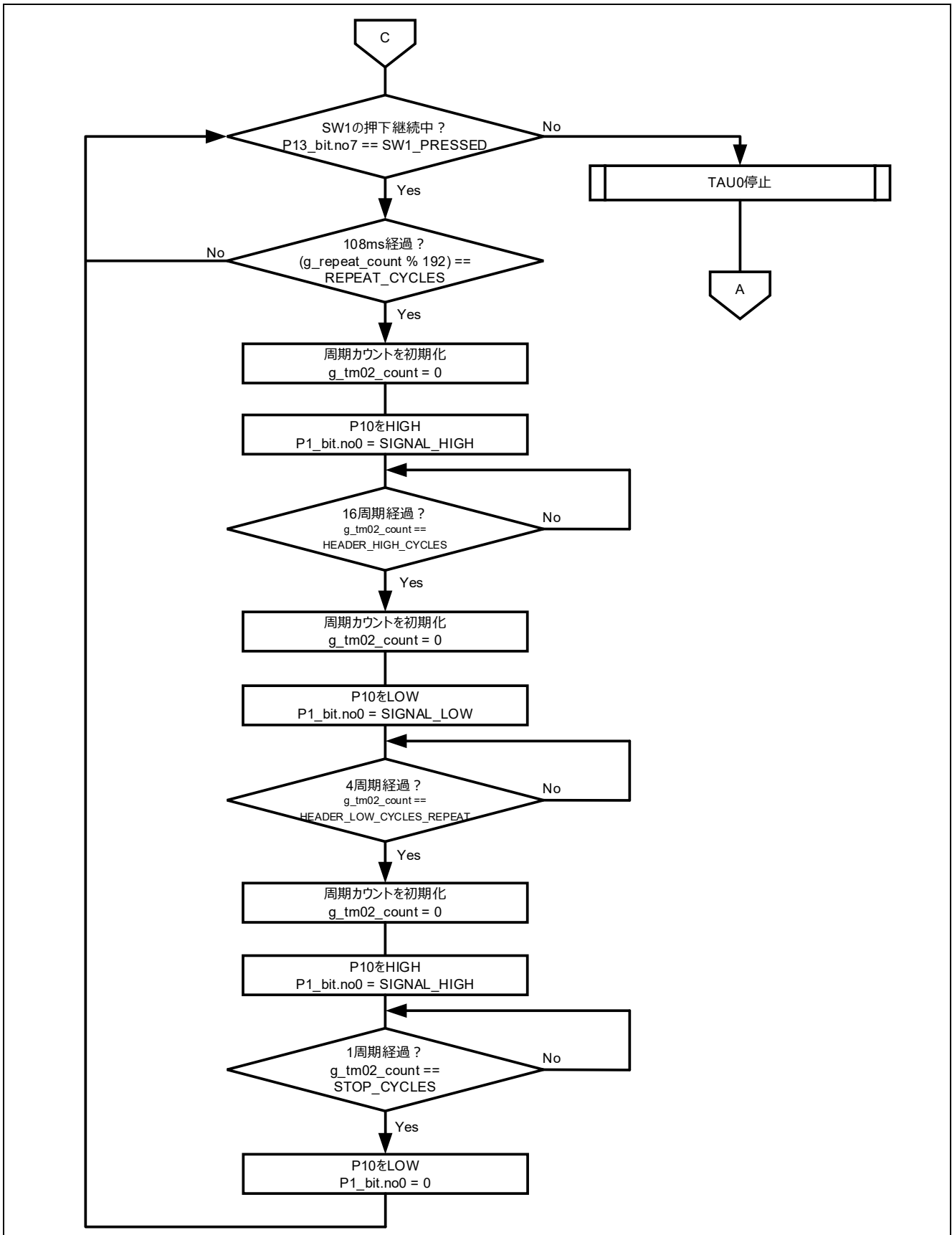
カスタマーコードとデータコードの計 32 ビット分の送信処理を行い、ストップビットを出力します。

図 4-3 main 関数フローチャート (3/4)



SW1 が押下され続けている場合、108ms 間隔でレポートを送信します。

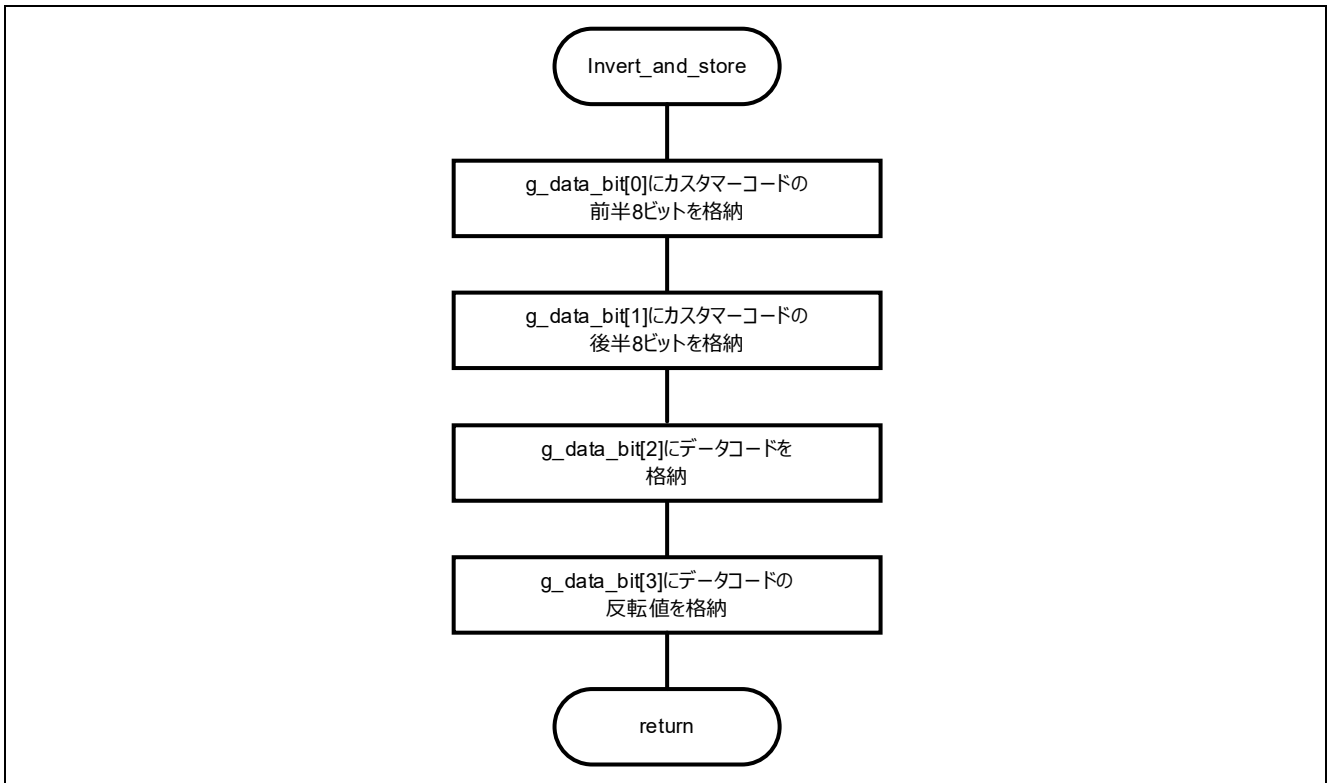
図 4-4 main 関数フローチャート (4/4)



## 4.8.2 invert\_and\_store 関数のフローチャート

invert\_and\_store 関数のフローチャートを以下に示します。

図 4-5 invert\_and\_store 関数フローチャート



4.8.3 tx\_data\_bit 関数のフローチャート

tx\_data\_bit 関数のフローチャートを以下に示します。

図 4-6 tx\_data\_bit 関数フローチャート (1/3)

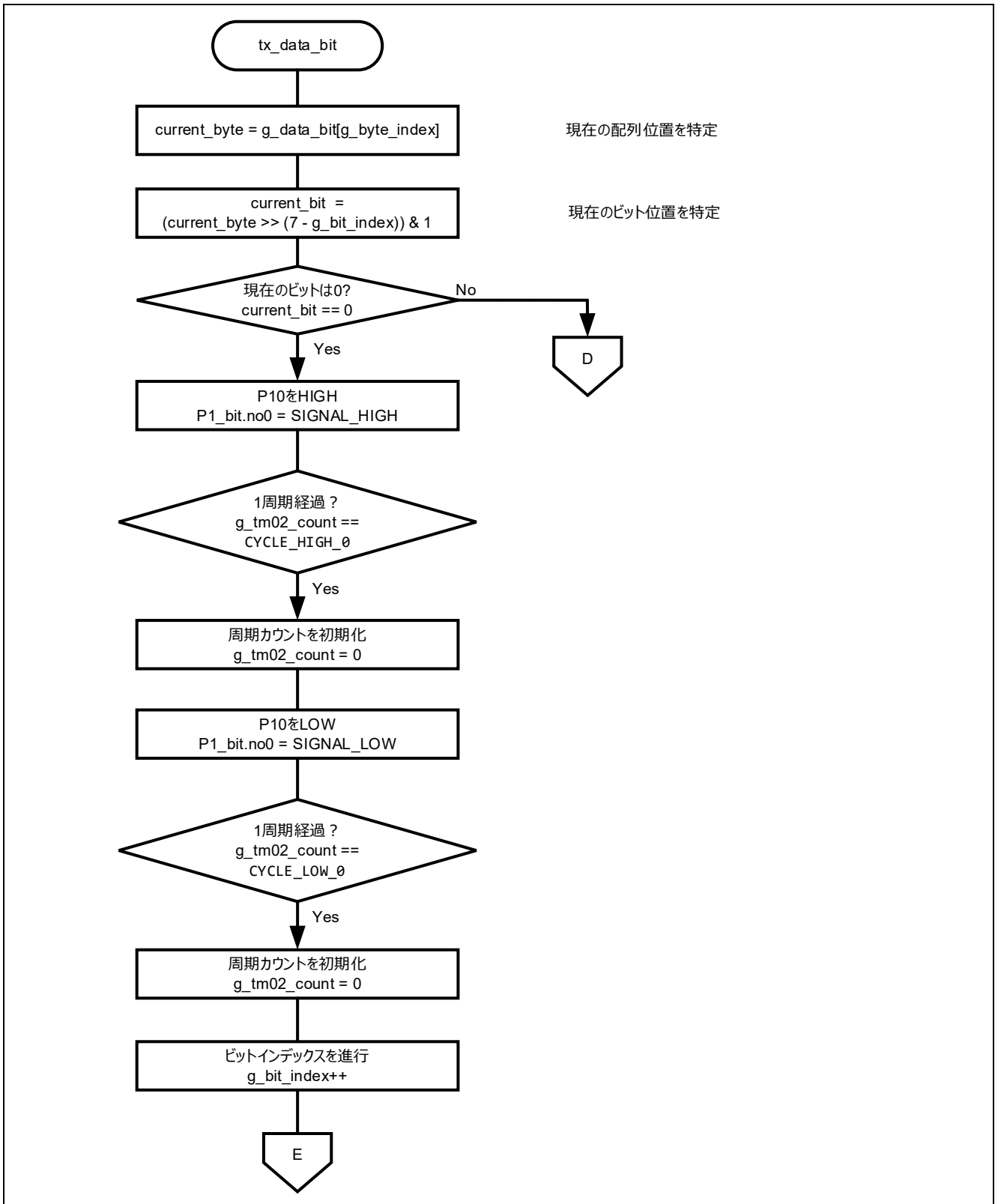


図 4-7 tx\_data\_bit 関数フローチャート (2/3)

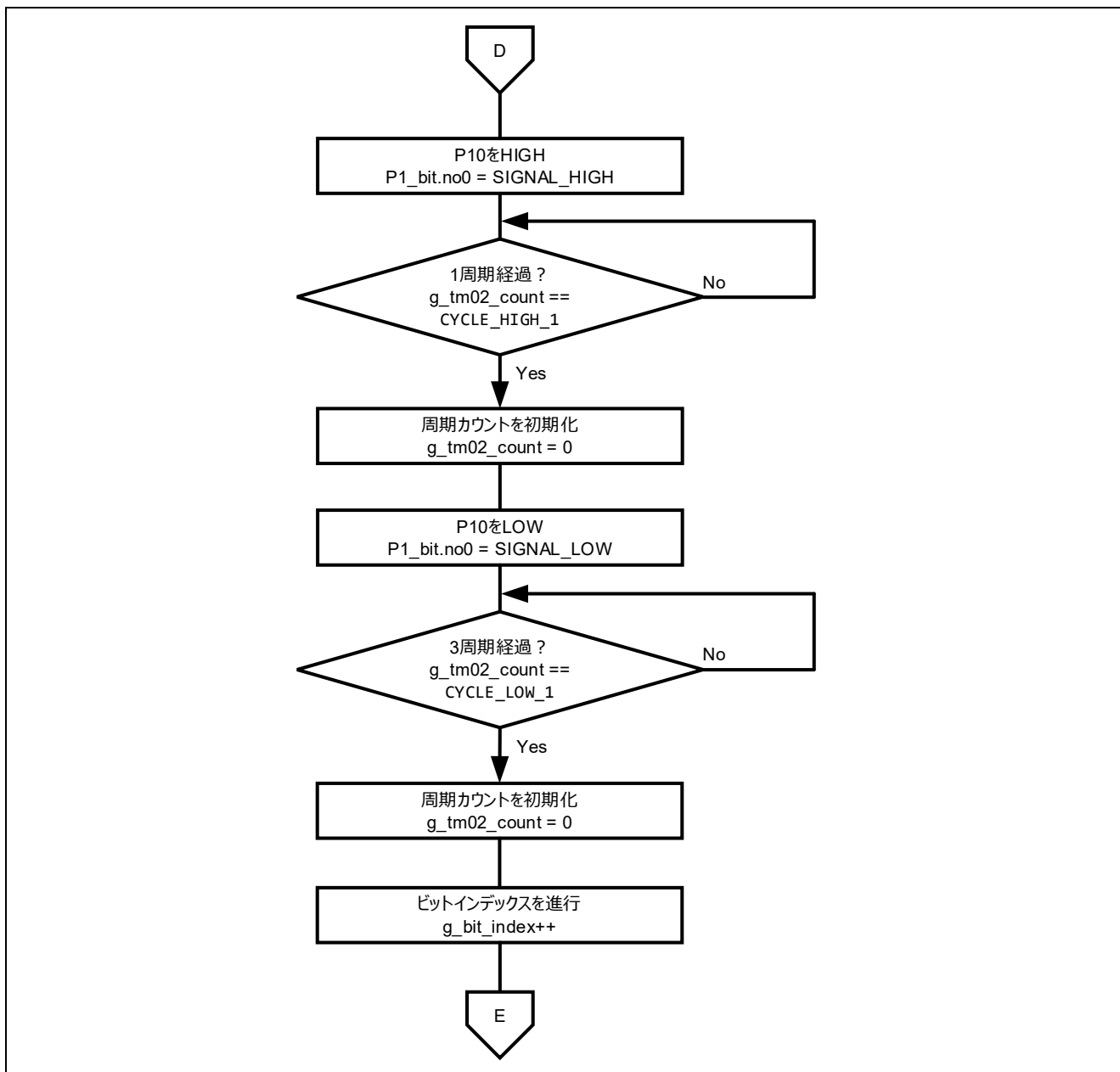
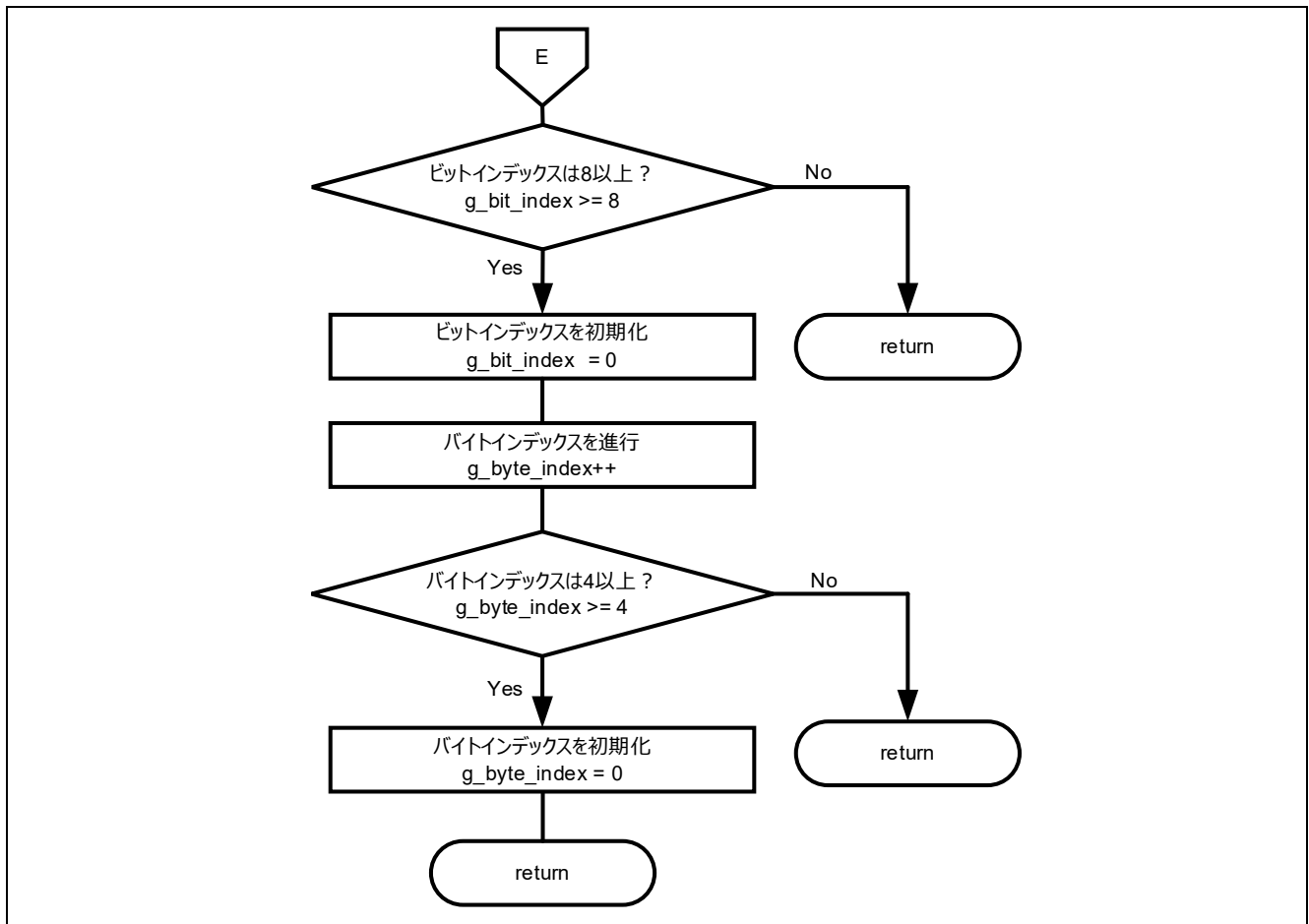


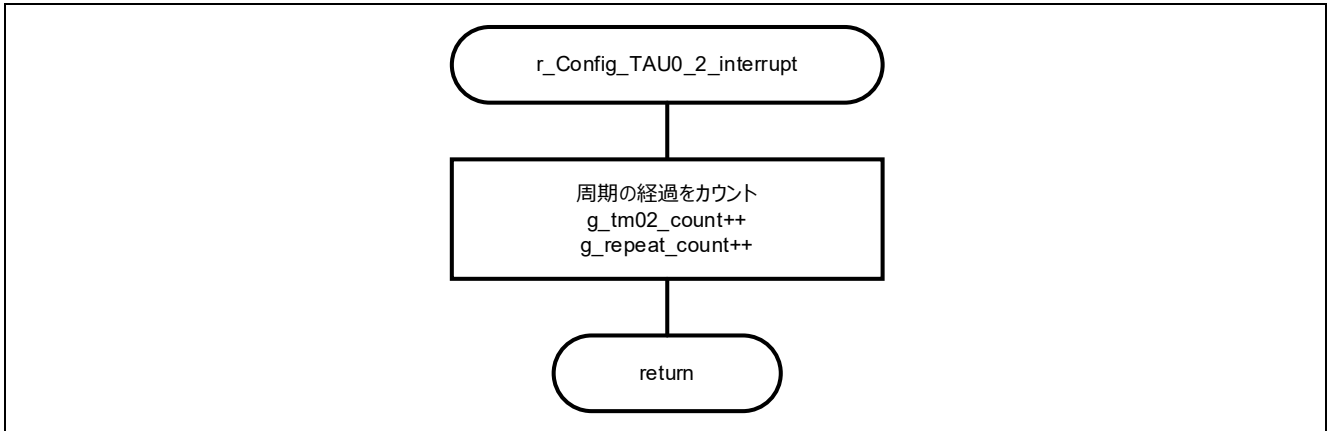
図 4-8 tx\_data\_bit 関数フローチャート (3/3)



## 4.8.4 r\_Config\_TAU0\_2\_interrupt 関数のフローチャート

r\_Config\_TAU0\_2\_interrupt 関数のフローチャートを以下に示します。

図 4-9 r\_Config\_TAU0\_2\_interrupt 関数フローチャート



## 4.9 応用例

### 4.9.1 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。

r01an7636\_elcl\_remocon\_send.scfg

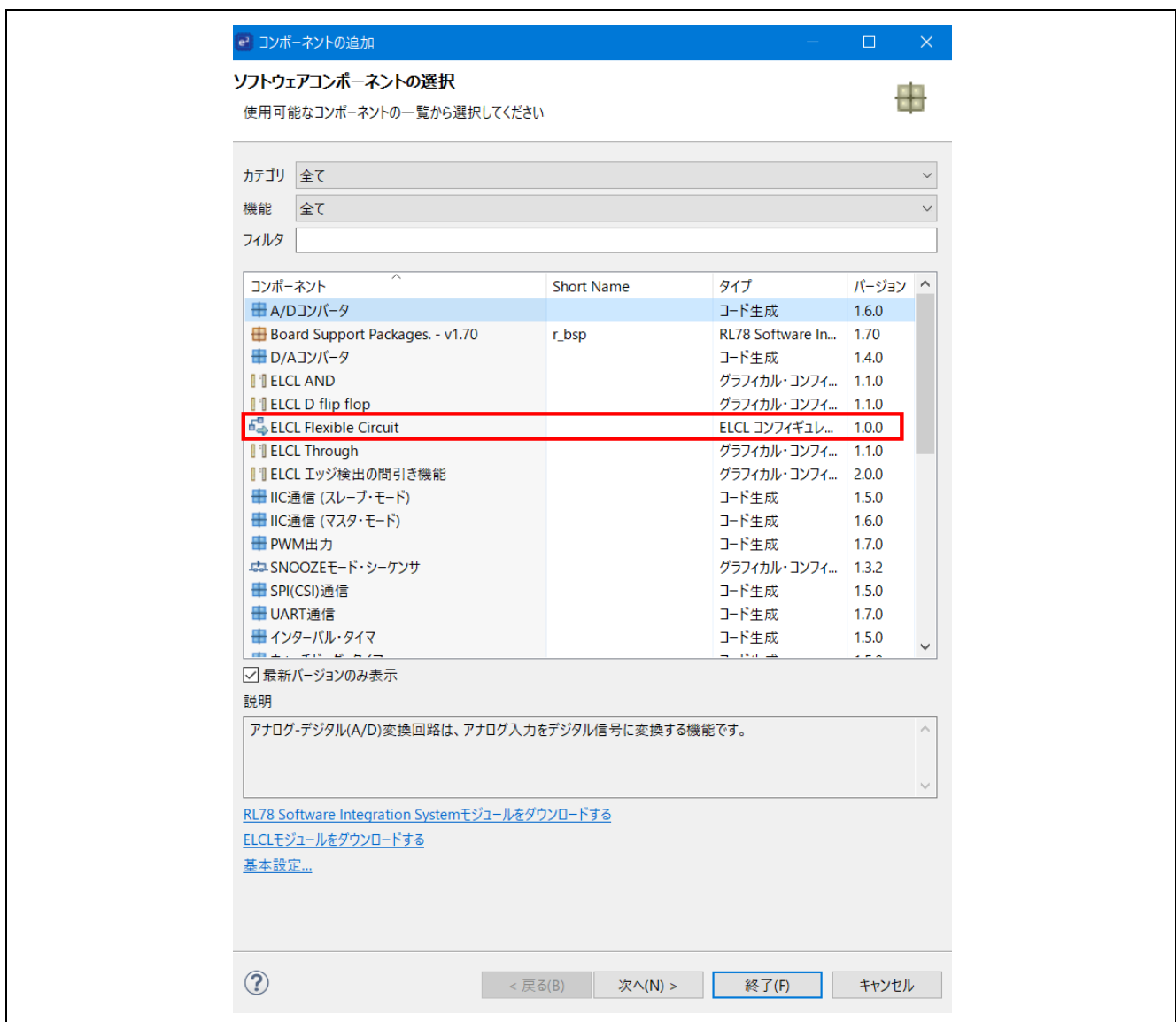
ファイルの説明と使用する上での設定例および注意事項を以下に示します。

#### 4.9.1.1 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. 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」タグをクリックし、「コンポーネントの追加」をクリックしてください。
3. 「ELCL Flexible Circuit」を選択してください。

図 4-10 コンポーネントの選択

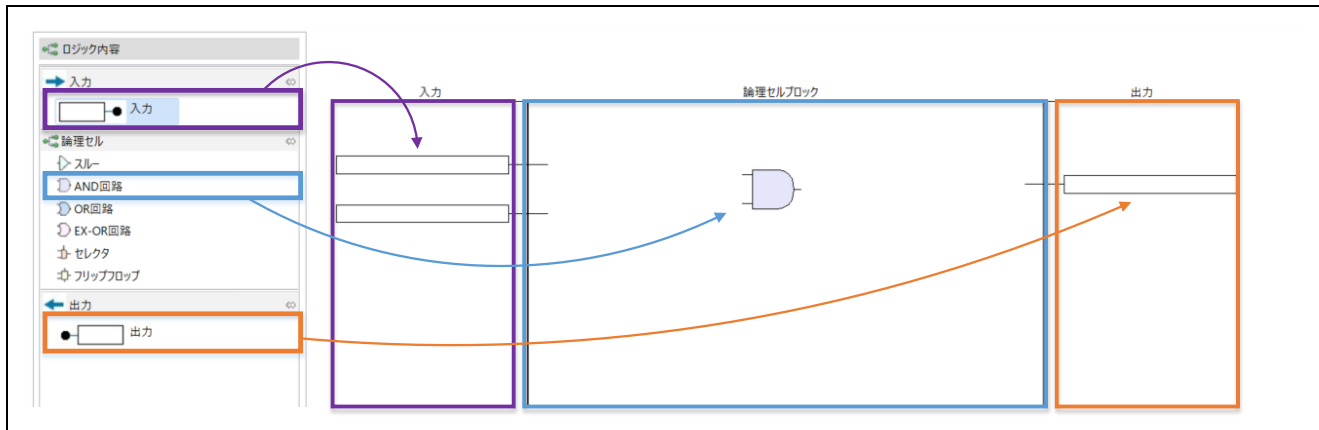


## 4.9.1.2 ELCL Flexible Circuit の使用方法

## 1. ELCL で使用するブロックを選択します。

「ロジック内容」の欄にある入力、論理セル、出力のブロックを対応するエリアにドラッグ&ドロップします。

図 4-11 ブロックの選択



## 2. プロパティを設定します。

設置したブロックをクリックすると、画面左下にある「プロパティ」の欄から、入出力に使用する端子、リソースなどを設定することができます。

図 4-12 プロパティの設定 (入力)

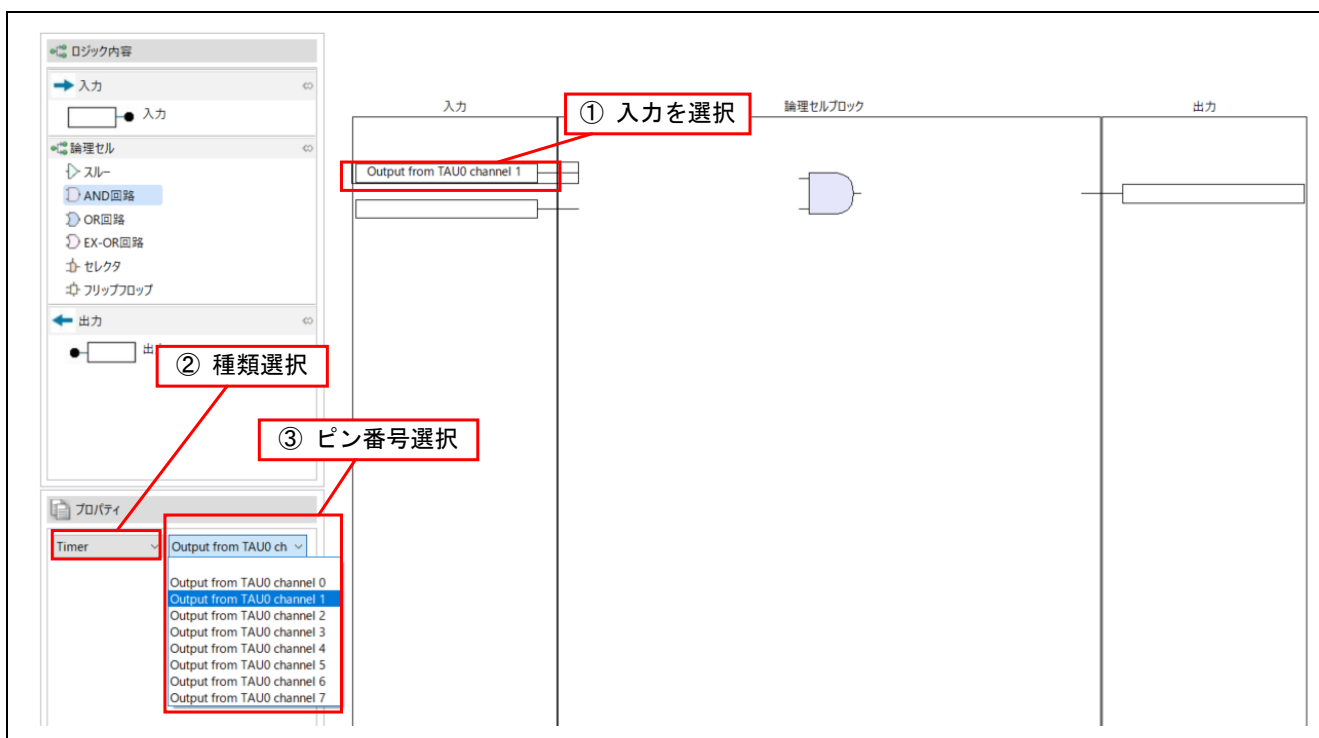


図 4-13 プロパティの設定 (論理セル)

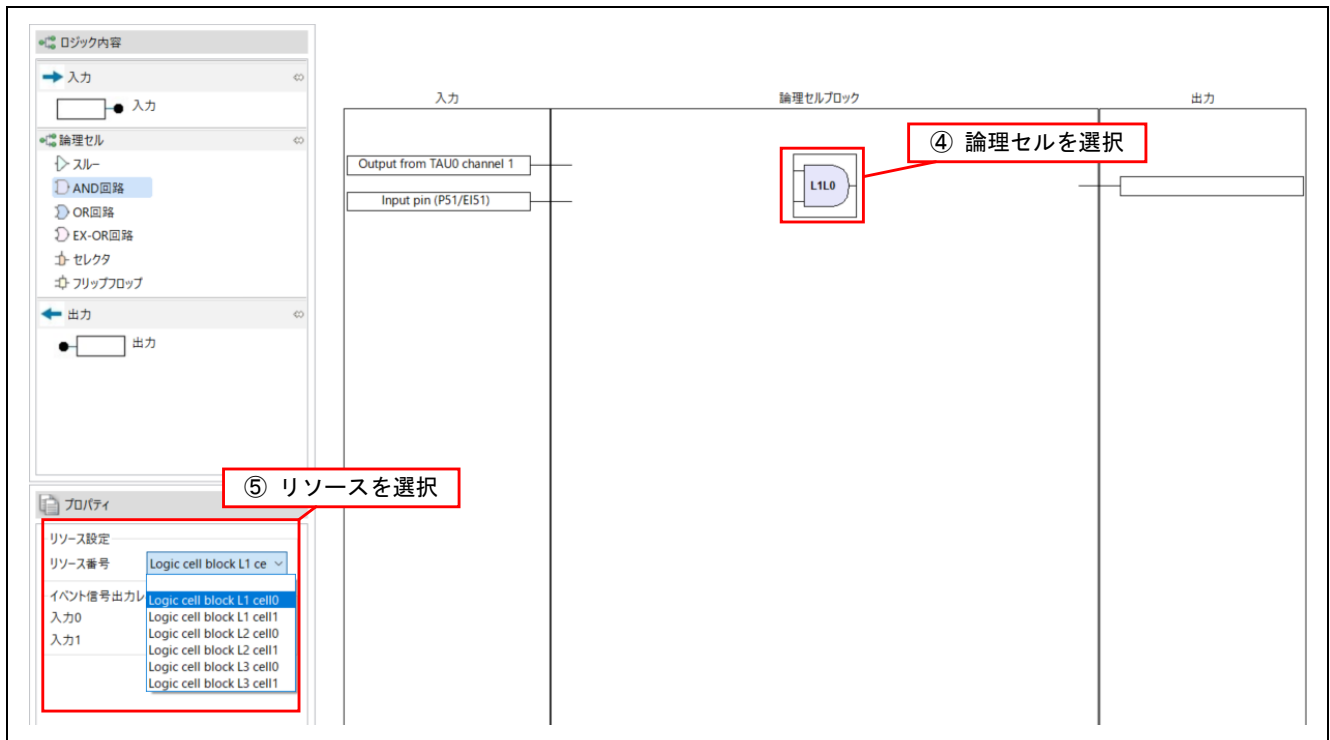
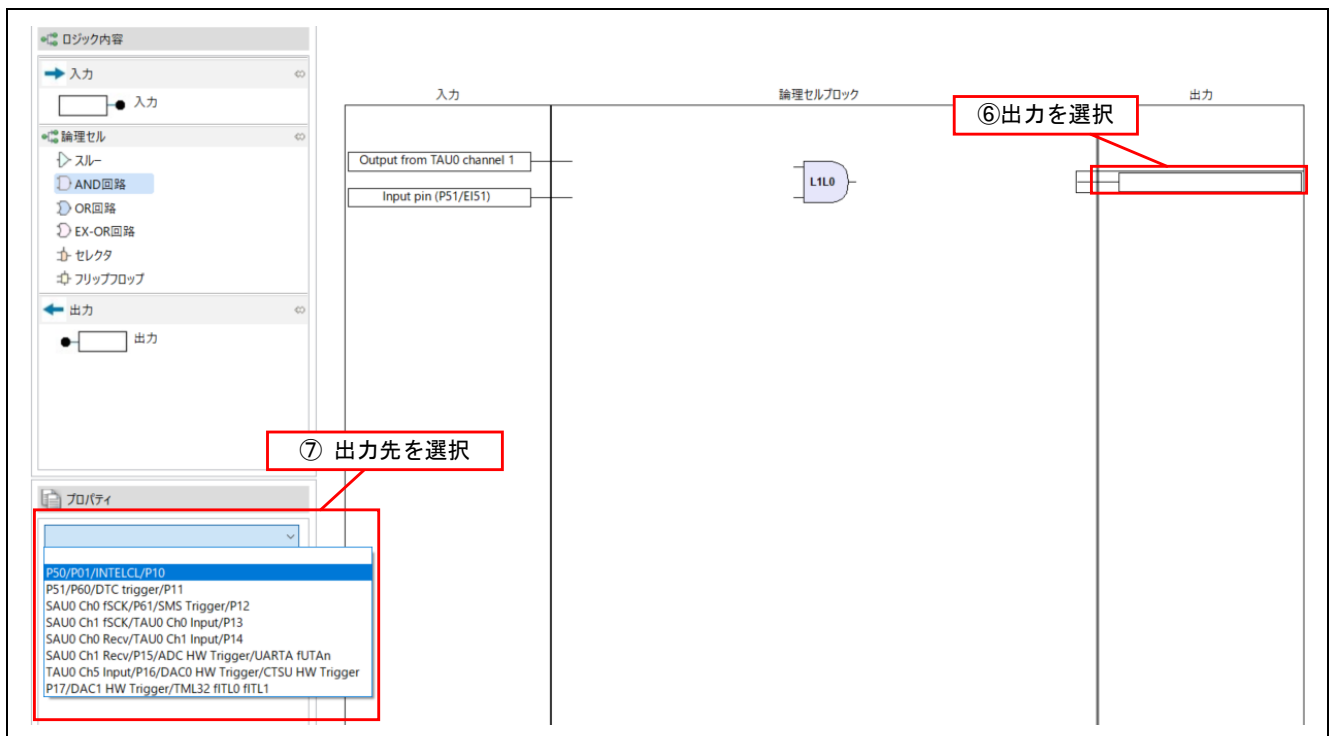


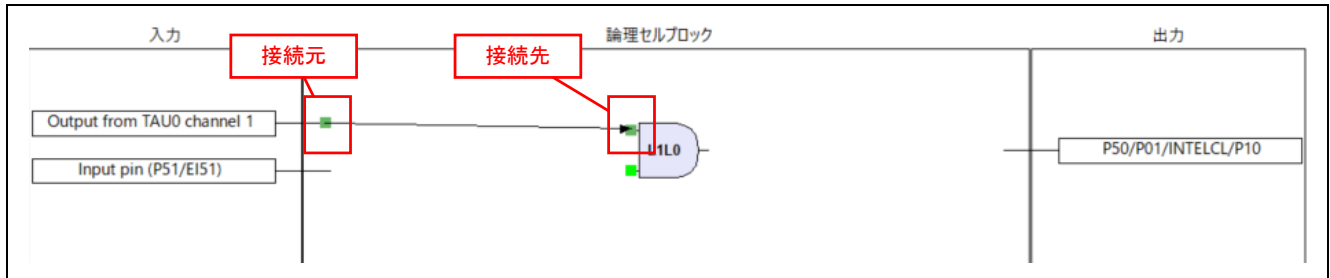
図 4-14 プロパティの設定 (出力)



## 3. ブロックを接続します。

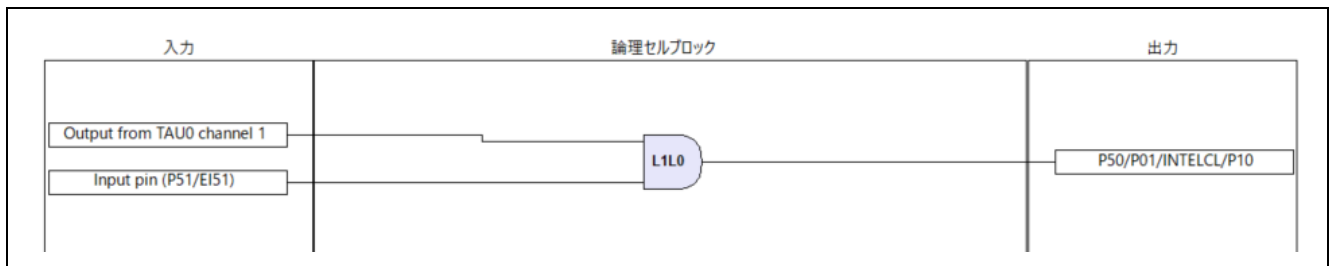
接続元にカーソルを合わせると緑色の点が表示されます。その状態でクリックし、接続先までカーソルをドラッグすることで、ブロック同士を接続することができます。

図 4-15 ブロックの接続



## 4. 本アプリケーションノートでの設定完了イメージ図を以下に示します。

図 4-16 設定完了イメージ図



## 4.9.1.3 r01an7636\_elcl\_remocon\_send.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 4-7 スマート・コンフィグレータの設定値 (1/2)

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 <sub>DD</sub> 設定：4.0V ≤ EV <sub>DD0</sub> < 5.5V 高速オンチップ・オシレータ：32MHz f <sub>IHP</sub> ：32MHz f <sub>CLK</sub> ：32000kHz (高速オンチップ・オシレータ) f <sub>SXP</sub> ：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 <sup>注</sup> 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する
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_LVD0	コンポーネント：電源検出回路 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 <sub>LVD0</sub> )：1.65 (V)

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 4-8 スマート・コンフィグレータの設定値 (2/2)

タグ名	コンポーネント	内容
コンポーネント	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 <sub>CLK</sub> 周期設定 : 26.32μs 割り込み設定 : 使用する 優先順位 : レベル 3 PWM スレーブ選択設定 : チャンネル 1 スレーブ  (スレーブ 1) デューティ : 33.3% 初期出力値 : 0 出力レベル : アクティブ・ハイ タイマ出力は ELCL への入力にのみ使用する 割り込み設定 : 使用しない
	Config_TAU0_2	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_2 動作クロック : CK00 クロックソース : f <sub>CLK</sub> インターバル時間 : 562μs 割り込み設定 : 使用する 優先順位 : レベル 3
	Config_ELCL	コンポーネント : ELCL Flexible Circuit Common setting : L1L0  Detail setting : L1L0 Input signal selector : TO01 , P51 Application : AND Output signal selector : P01
	Config_PORT	コンポーネント : ポート ポート選択 : PORT0 , PORT1 , PORT13 P01 : 出力 (ELCL 出力信号を出力) P10 : 出力 P137 : 入力 (SW1 に 0 を入力)

## (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。

## (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、「4.2 オプション・バイトの設定一覧」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。

## (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。

## (4) Config\_LVD0

サンプルコードの電源管理の設定を行います。

「4.2 オプション・バイトの設定一覧」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。

## (5) Config\_TAU0\_0

サンプルコードの TAU0\_0 の設定を行います。

本サンプルコードでは、周波数 38kHz のサブキャリア波形生成に使用するため、周期を  $26.32\mu\text{s}$  ( $= 1 / 38,000\text{Hz}$ )、デューティを 33.3% の PWM 出力として設定します。

## (6) Config\_TAU0\_2

サンプルコードの TAU0\_2 の設定を行います。

本サンプルコードでは、NEC フォーマットの周期である、 $562\mu\text{s}$  のインターバル・タイマとして設定します。割り込み処理にて、周期の経過をカウントします。

## (7) 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 L1L0 の AND 演算回路を使用し、サブキャリア波形とマスク波形を入力信号として選択します。

#### 4.9.2 送信データの変更方法

NEC フォーマットでは、16 ビットのカスタマーコードと 8 ビットのデータコードを送信します。

本サンプルコードでは以下の送信データが設定されています。

送信データを設定する場合は以下の変数の値を変更してください。

表 4-9 送信データ

データの種類 (データ長)	変数名	値
カスタマーコード (16 ビット)	g_customer_code	0x02fd
データコード (8 ビット)	g_data_code	0x86

## 5. プロジェクトのインポート方法

サンプルコードは e<sup>2</sup> studio のプロジェクト形式で提供しています。本章では、e<sup>2</sup> studio および CS+ へプロジェクトをインポートする方法を示します。インポート完了後、ビルドおよびデバッガの設定を確認してください。

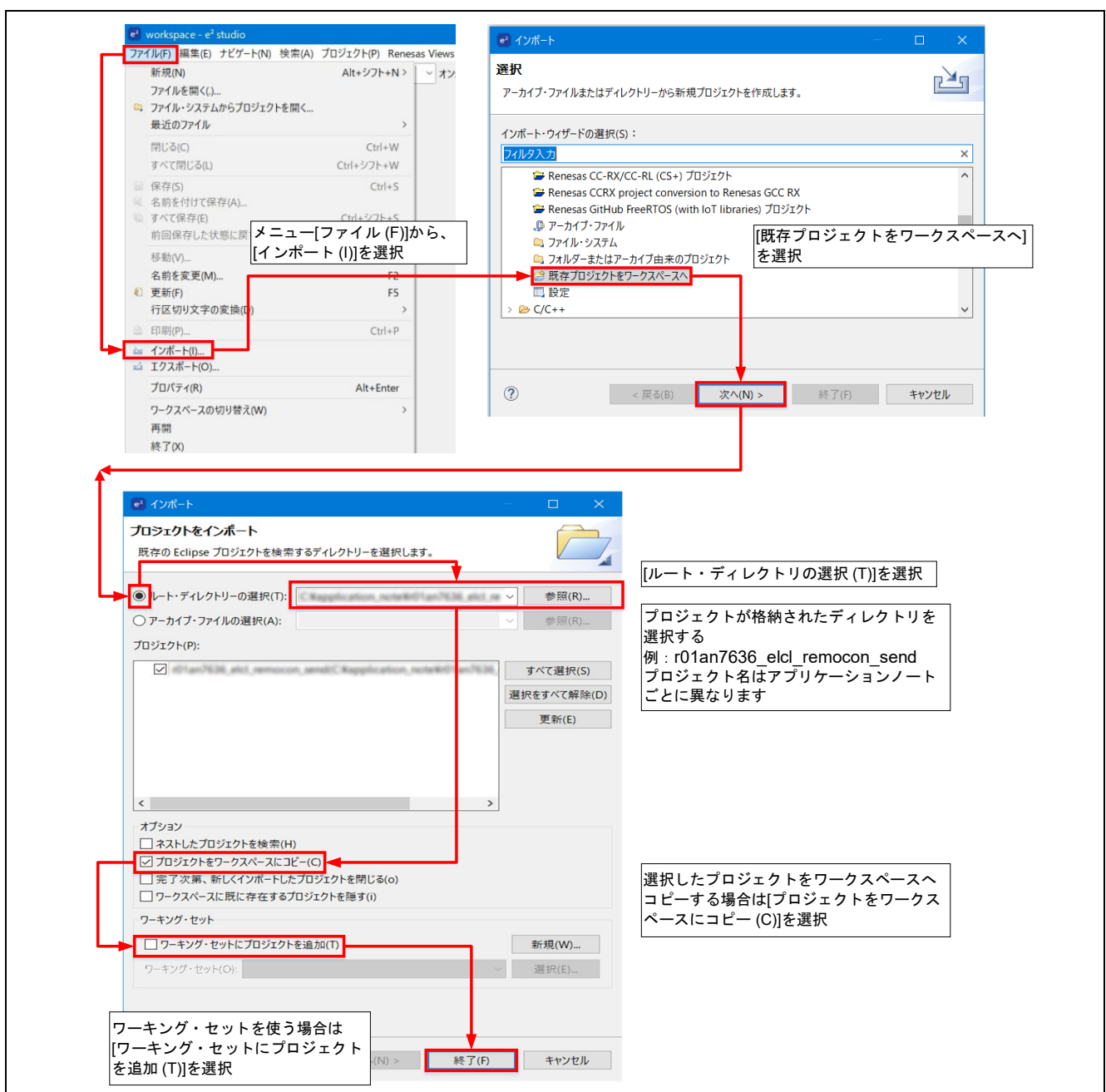
### 5.1 e<sup>2</sup> studio での手順

e<sup>2</sup> studio でご使用になる際は、以下の手順で e<sup>2</sup> studio にインポートしてください。

なお、e<sup>2</sup> 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 e<sup>2</sup> 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 5-1 プロジェクトを e<sup>2</sup> studio にインポートする方法



## 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 (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 : e<sup>2</sup> 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)

アプリケーションノート RL78/G23

リモコン信号受信 (NEC フォーマット、STOP モード使用) (R01AN6001)  
(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.14.2025	—	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、リセットを解除してください。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っていると、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）、交通管制（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](http://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/](http://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。