RL78 ファミリ

FPB ボードでスタンドアロン版 QE を用いたタッチアプリケーション開発

要旨

本アプリケーションノートでは、タッチ電極搭載の RL78/G22 FPB (Fast Prototyping Board) を使用し、静電容量タッチセンシングを応用するアプリケーションの作成に必要な手順について説明します。

本アプリケーションノートは、“CS+、スタンドアロン版スマート・コンフィグレータ、スタンドアロン版 QE for Capacitive Touch”を用いた開発ガイドです。スタンドアロン版 QE for Capacitive Touch 注は、シリアル通信を使用することでデバイスや統合開発環境 (IDE) に依存せずに開発を行うことができます。

注　QE for Capacitive Touch は、静電容量式タッチセンサを使用した組み込みシステム開発に必要なタッチインタフェースの初期設定や感度調整に対応した開発支援ツールです。

動作確認デバイス

RL78/G22

静電容量センサユニット (CTSU) をサポートする RL78 ファミリ

目次

1. 概要

本アプリケーションノートでは、RL78 ファミリを使用した静電容量タッチ機能をシステムに組み込む、以下の手順を説明します。

- RL78/G22 FPB を使用したスタンドアロン版スマート・コンフィグレータによるプロジェクト作成
- スタンドアロン版 QE for Capacitive Touch によるタッチインタフェース作成とチューニング及びモニタリング

本アプリケーションノートは、RL78/G22 FPB を使用して説明をしていますが、他の RL78 ファミリの静電容量タッチ IP 搭載のデバイスにも活用することができます。

2. 動作環境

本アプリケーションノートの開発環境を表 2-1、表 2-2 に示します。

付属のサンプルコードは、表 2-1 に示すバージョンの開発環境で開発されています。また、本アプリケーションノートは () 内に示すバージョンでの開発もサポートしています。

本アプリケーションノートでは、スタンドアロン版 QE にて生成したプログラムを CS+で RL78/G22 に書き込み、書き込んだプログラムを RL78/G22 で動作させます。

表 2-1　開発環境 (ソフトウェア)

項目	内容	バージョン
統合開発環境 (IDE)	CS+ for CC	V8.13.00 (V8.09.00 以降)
コンパイラ	CC-RL	V1.15.00 (V1.12.00 以降)
静電容量式タッチセンサ対応開発支援ツール	スタンドアロン版 QE for Capacitive Touch	V4.1.0
スマート・コンフィグレータ	RL78 スマート・コンフィグレータ	V1.12.00 (V1.5.00 以降)
SIS (Software Integration System) モジュール	・Capacitive Sensing Unit driver. (r_ctsu) ・Touch middleware. (rm_touch)	V2.10 (V2.10 以降)

注意　タッチセンサのチューニング時に、CC-RL 無償評価版の V1.12.00 以降のバージョンを使用してコンパイルする場合は、コンパイラの最適化レベルを”デバッグ優先 (-onothing)”に設定してビルドしてください。

表 2-2　開発環境 (ハードウェア)

項目	内容
使用マイコン	RL78/G22 (R7F102GGE2DFB)
ターゲットボード	RL78/G22 Fast Prototyping Board (RTK7RLG220C00000BJ)

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 2-3　動作確認条件

項目	内容
動作電圧	5.0V LVD0 検出電圧：リセット・モード 立ち上がり時 TYP. 2.67V (2.59 V～2.75 V) 立ち下がり時 TYP. 2.62V (2.54 V～2.70 V)
動作周波数	高速オンチップ・オシレータ・クロック (fIH) : 32 MHz

2.1　QE for Capacitive Touch の機能

QE for Capacitive Touch は、静電容量式タッチセンサを使用した組み込みシステム開発に必要なタッチインタフェースの初期設定や感度調整に対応した開発支援ツールです。

QE for Capacitive Touch の主な機能は次のとおりです。

- タッチインタフェース構成の作成機能
  ボタン等のタッチインタフェースの配置とタッチセンサ (電極) の割り当てを視覚的に設定できます。

- チューニング機能
  タッチインタフェースのオフセットや感度を自動的にチューニングできます。

- モニタリングとパラメータ調整機能
  タッチインタフェースの動作をモニタリングでき、さらにパラメータの細かな調整を行うことができます。

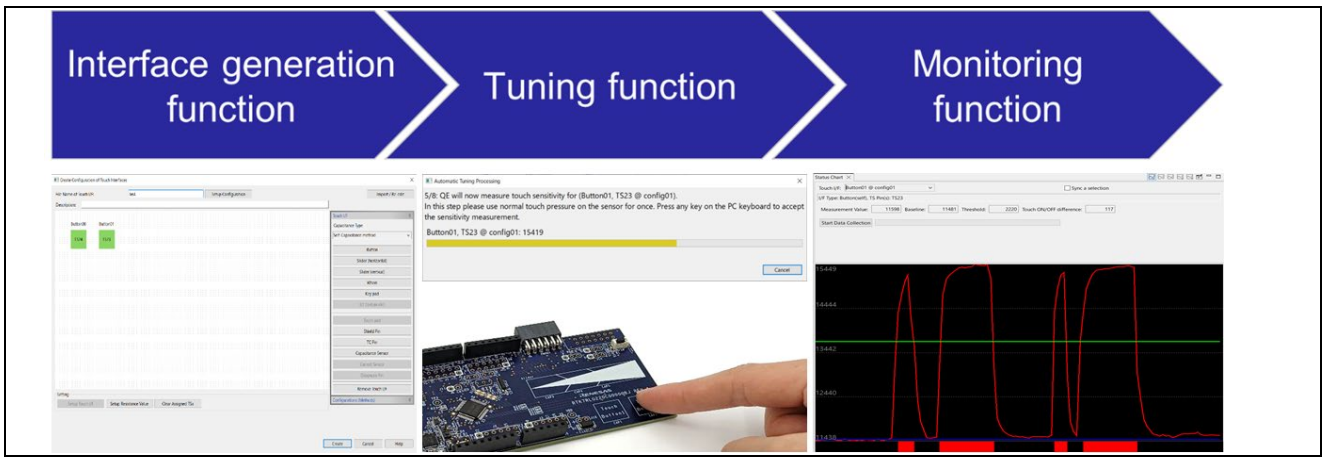


図 2-1　QE for Capacitive Touch の主な機能

3. 開発環境構築

本章では、開発環境のインストール手順およびハードウェアの設定について説明します。

3.1 開発ツールのインストール

本アプリケーション例では、以下のツールを使用します。

- CS+
- スタンドアロン版スマート・コンフィグレータ
- スタンドアロン版 QE for Capacitive Touch

すでに全てのツールをインストール済みの場合は、3.1 節の手順は不要です。

3.1.1 統合開発環境 CS+のインストール手順

1. 下記リンクから、最新の ”統合開発環境 CS+ for CC” のインストーラをダウンロードします。
   統合開発環境 CS+ | Renesas ルネサス
2. ダウンロードした zip ファイルを解凍し、インストーラファイルを実行します。
3. 「CS+のセットアップを開始する」をクリックします。
4. “RL78 ファミリ用ツール”にチェックが入っていることを確認します。
5. インストールが完了後、完了ボタンを押します。

3.1.2 スタンドアロン版スマート・コンフィグレータのインストール手順

1. 下記リンクから、 最新の“RL78 スマート・コンフィグレータ” のインストーラをダウンロードします。
   RL78 スマート・コンフィグレータ | Renesas ルネサス
2. ダウンロードした EXE ファイルを実行し、インストーラを起動します。
3. インストーラを起動後、画面の手順に従ってインストールします。

3.1.3 スタンドアロン版 QE for Capacitive Touch のインストール手順

1. 下記リンクから、最新の ”静電容量式タッチセンサ対応開発ツール QE for Capacitive Touch” のインストーラをダウンロードします。
   静電容量式タッチセンサ対応開発支援ツール QE for Capacitive Touch | Renesas ルネサス
2. ダウンロードした zip ファイルには、プラグイン版 QE とスタンドアロン版 QE が同梱されています。ダウンロードした zip ファイルを展開し、スタンドアロン版 QE をインストールします。

注意　Windows のパス名の文字数制限 (260 文字) を超えないようにドライブのルートに近い場所に展開してください。

展開先の例：C:¥Renesas フォルダ以下

このとき、移動先として Windows フォルダや Program Files フォルダ、名前にスペースが含まれるフォルダなどは避けてください。

3.2　ハードウェア設定

ハードウェアの設定およびボード接続について説明します。表 3-1 に本アプリケーションでのターゲットボードにおけるジャンパの設定を示します。ボードへの電源供給については USB を使用します。図 3-1 のように PC とターゲットボードを USB ケーブルで接続します。

回路接続の変更内容は、ターゲットボードによって異なります。詳細は使用する FPB のユーザーズマニュアルの「QE for Capacitive Touch 使用時の注意事項」をご確認ください。

表 3-1　ボードのジャンパ設定

位置	回路グループ	ジャンパ設定	用途
J16 注	QE シリアル接続切り替えジャンパ	オープン	QE のシリアル接続機能を実行する場合
		ショート	COM Port デバック回路を有効にする場合
J17	電源選択ヘッダ	1-2 ショート	5V 電源を選択

注　本開発手順では、チューニング時およびモニタリング時に J16 の設定を切り替えます。
　　詳細は、8.4　調整の手順 3、手順 4、手順 5 をご確認ください。

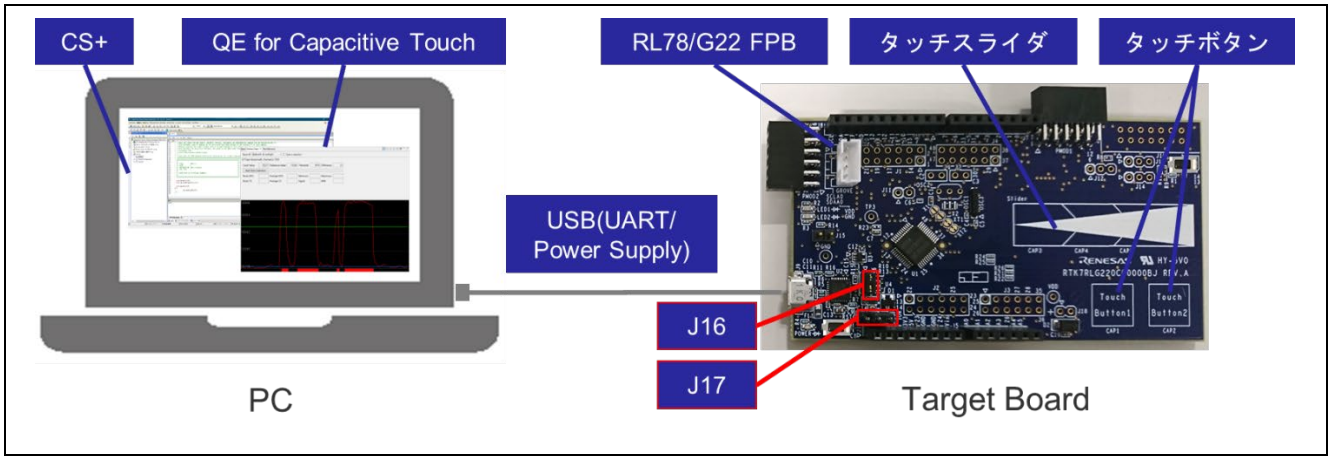


図 3-1　PC とターゲットボードの接続

4.　アプリケーション開発手順

アプリケーション開発の手順を説明します。

開発の流れは QE for Capacitive Touch のワークフローに従います。

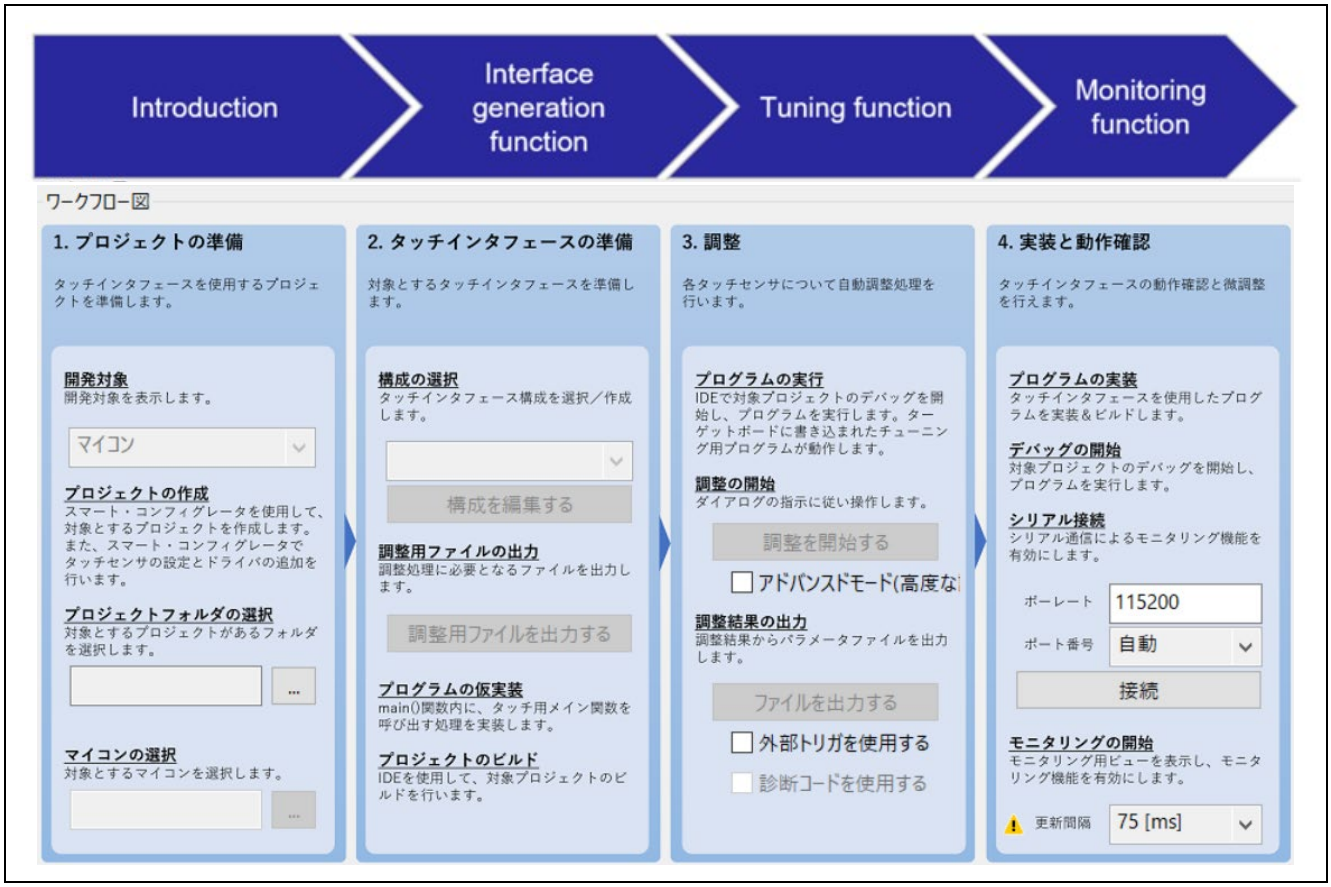


図 4-1　アプリケーション開発手順

ワークフローの各項目を表 4-1 に示します。本表の章番号は、関連章にリンクされています。章番号をクリックして使い方を確認してください。プロジェクトの作成、プログラムの実装、プロジェクトのビルド、デバッグの開始は、IDE やスマート・コンフィグレータを使います。

表 4-1　QE for Capacitive Touch を用いた開発のワークフロー

項目				章番号
プロジェクトの準備	プロジェクトの作成	IDE によるプロジェクトの作成		6
		スマート・コンフィグレータの設定		7
			クロックとシステム	7.2
			CTSU 用ドライバ	7.5.1
			タッチ用ミドルウェア	7.5.2
			シリアルインタフェース (UART 通信)	7.5.3
			電源検出回路 (LVD)	7.5.4
			ポート機能 (PORT)	7.5.5
			ボードサポートパッケージ	7.5.6
			未使用端子の設定	7.6
	プロジェクトフォルダの選択			8.2
	マイコンの選択			
タッチインタフェースの準備	構成の選択			8.3
	調整用ファイルの出力			
	プログラムの仮実装			
	プロジェクトのビルド			
調整	プログラムの実行			8.4
	調整の開始			
	調整結果の出力			
実装と動作確認	プログラムの実装			8.5
	デバッグの開始			
	シリアル接続			
	モニタリングの開始			

5.　アプリケーション例

5.1　アプリケーション例の概要

本アプリケーションノートでは、2 つのボタンおよび 1 つのスライダを使用するアプリケーションを例に説明します。また、シリアル通信を介してタッチ性能のチューニングとモニタリングを行います。

6 章以降で、2 つのボタンおよび 1 つのスライダを使用するアプリケーションを作成し、ボタンまたはスライダをタッチした場合の検出状況をモニタリングする方法を示します。

備考　タッチアプリケーションのタッチ性能のモニタリングは、OCD (On-Chip Debugging) エミュレータを介した通信によっても確認できます。ただし、RL78 ファミリの場合、モニタリングパフォーマンスは、RL78 ファミリのオンチップ・デバッグ機能によって制限されます。
シリアル通信を介してタッチ性能のモニタリングを行うことで、スムーズなモニタリングが可能になります。また、タッチセンサのチューニングもシリアル通信を介すことができます。

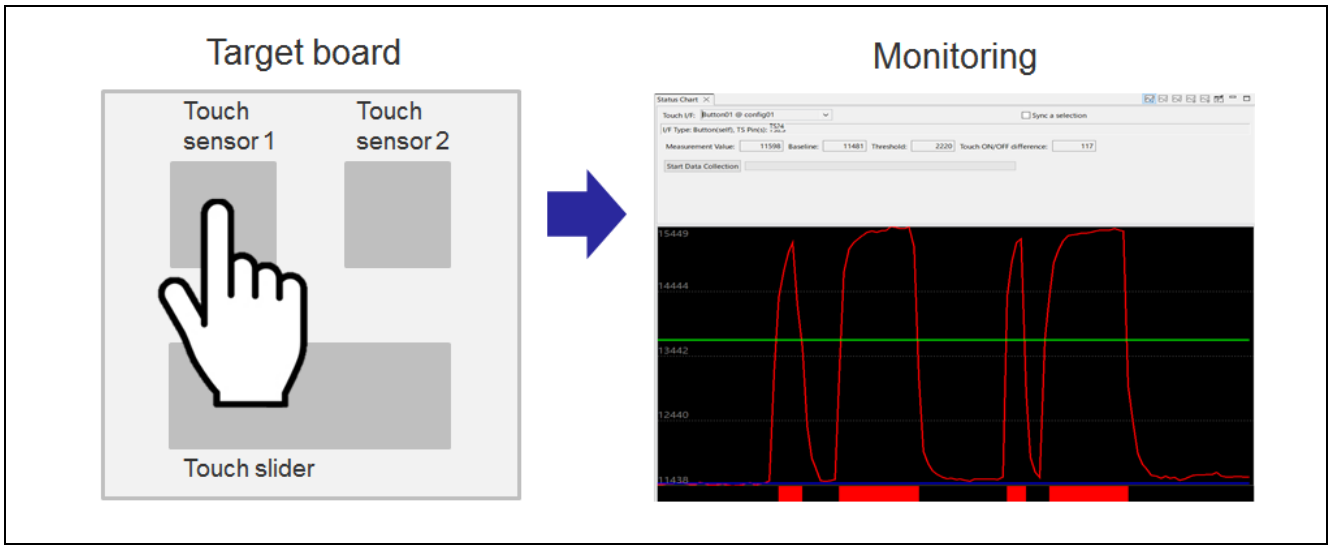


図 5-1　アプリケーション例

本アプリケーションノートには、表 5-1 に示した 2 つのサンプルコードが付属しています。両サンプルコードは、開発手順は共通ですが、スマート・コンフィグレータの設定と qe_touch_sample.c ファイルへ追加するコードが一部異なります。以降の 6 章から 8 章では、ソフトウェアタイマを用いたタッチアプリケーションを例にして、開発手順を説明しています。ハードウェアタイマと LED 制御の実装方法については「9. 応用例」を参照してください。

表 5-1　付属のサンプルコードの概要

ファイル名	タッチ計測周期を生成するタイマ	LED 制御
Capacitive_Touch_Project_Example	ソフトウェアタイマ	無し
Capacitive_Touch_Project_HardwareTimer_Example	ハードウェアタイマ	有り

5.2　使用端子一覧

本アプリケーション例の使用端子を、表 5-2 に示します。

UART 通信およびタッチセンサは、使用するターゲットボードの仕様にあわせて設定する必要があります。

表 5-2　本アプリケーション例の使用端子一覧

項目	端子	用途
UART 通信	RxD0/P11	チューニング、モニタリングを行う
	TxD0/P12	
タッチセンサ①	TS24/P26	ボタンでのタッチを検出する (TS_B1)
タッチセンサ②	TS23/P25	ボタンでのタッチを検出する (TS_B2)
タッチスライダ	TS20/P22	スライダ上の、指が左右に移動した際の位置を検出する (TS_S)
	TS21/P23	
	TS22/P24	

本アプリケーション例で使用するタッチセンサの配置を図 5-2 に示します。

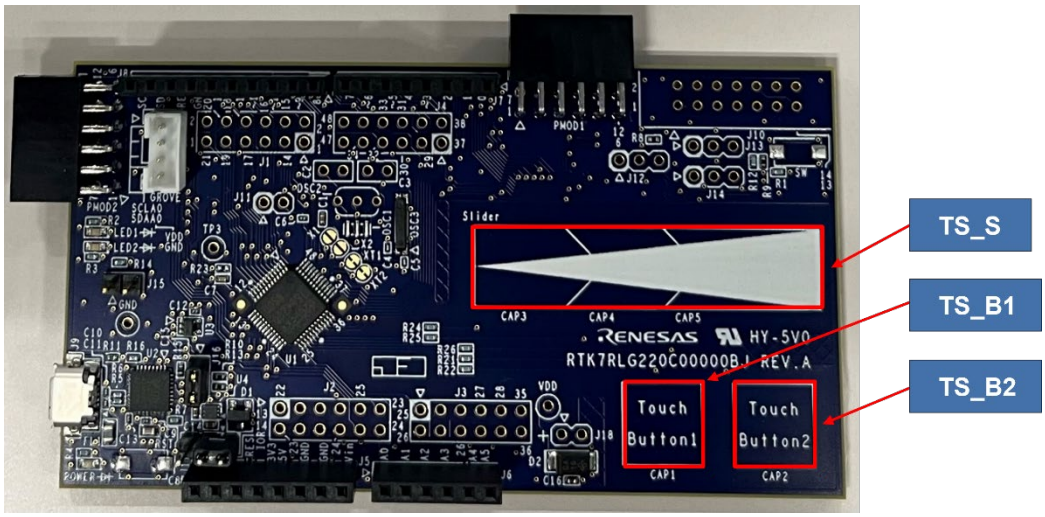


図 5-2　タッチセンサの配置

6.　新規プロジェクトの作成

CS+を起動し、プロジェクトを作成します。

“プロジェクト作成”ダイアログにて、本アプリケーションでは以下を選択します。

- マイクロコントローラ(T)　: RL78
- 使用するマイクロコントローラ(M)　: R7F102GGExFB (48pin)
- プロジェクトの種類(K)　: アプリケーション (CC-RL)
- プロジェクト名(N)　: (任意のプロジェクト名)
- 作成場所(L)　: (任意の作成場所)

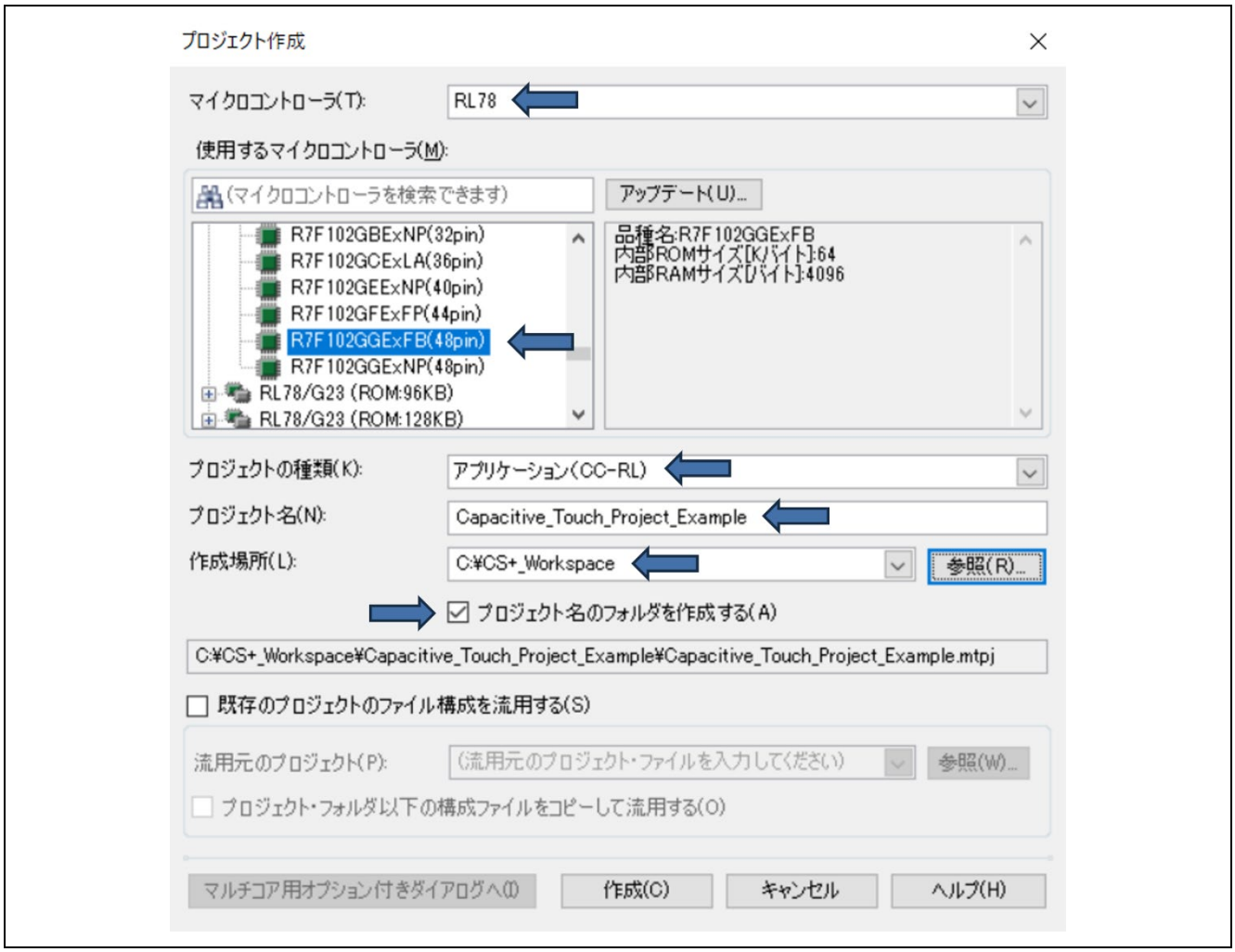


図 6-1　プロジェクトの作成

7.　スマート・コンフィグレータの設定

スマート・コンフィグレータの設定手順を説明します。本アプリケーション例で必要な設定は、以下のとおりです。

- クロックとシステム
- CTSU 用ドライバ
- タッチ用ミドルウェア
- シリアルインタフェース（UART 通信）
- 電圧検出回路（LVD）
- ポート機能（PORT）

7.1　スマート・コンフィグレータの起動

CS+の”プロジェクト・ツリー”で、”スマート・コンフィグレータ”をダブルクリックし、起動します。

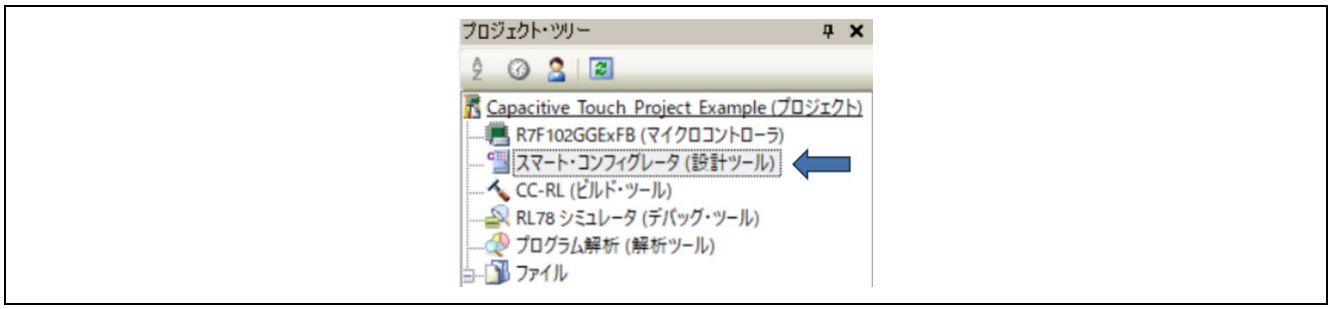


図 7-1　スマート・コンフィグレータを起動

上記方法でスマート・コンフィグレータが起動しない場合は、以下の 2 点を確認してください。

- スマート・コンフィグレータのプロパティで”ファイルパス”が正しく設定されているか
- メニューの [ツール] – [プラグインの管理]から、”RL78 用スマート・コンフィグレータ通信プラグイン”がチェックされているか

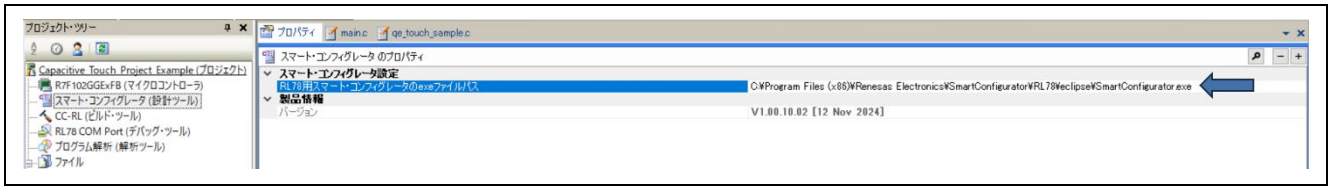


図 7-2　スマート・コンフィグレータのファイルパス

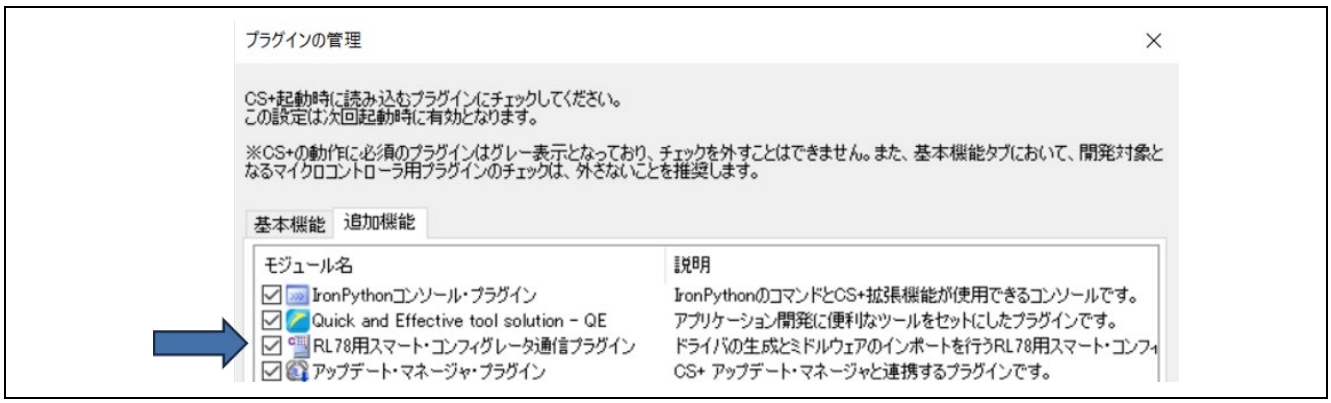


図 7-3　プラグインの管理

7.2　クロックとシステムの設定

クロックとシステムの設定手順を説明します。

1. スマート・コンフィグレータを起動後、Smart Configurator view の下部にある[クロック]タブを選択し、クロックを設定します。EVDD がある MCU の場合は、動作モードに合わせて”EVDD 設定”を行ってください。

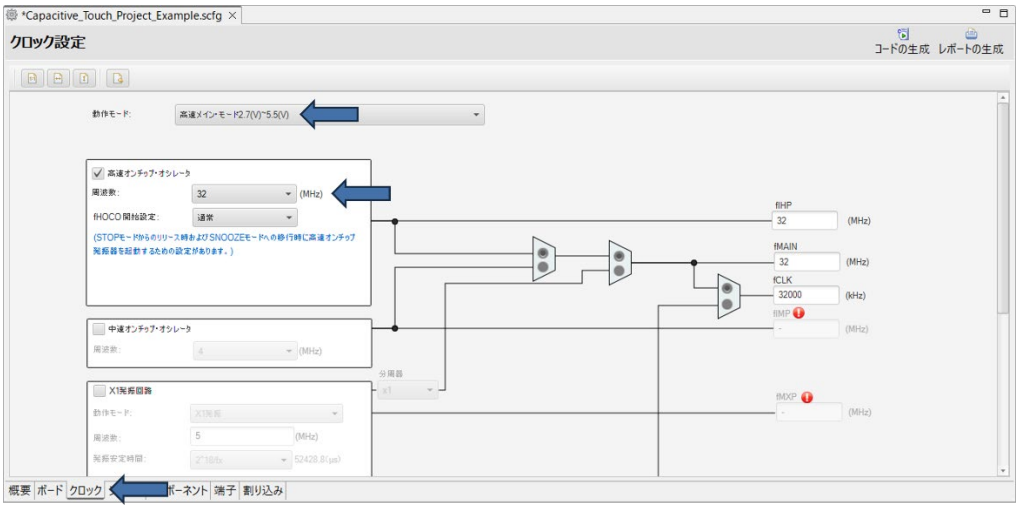


図 7-4　クロックの設定

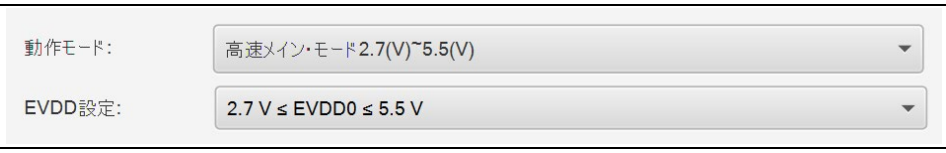


図 7-5　EVDD 設定

2. [システム]タブを選択し、デバッグ環境を設定します。

図 7-6　デバッグの設定

7.3　SIS (Software Integration System) モジュールのダウンロード

タッチアプリケーションの実装に必要な”CTSU 用ドライバ”と”タッチ用ミドルウェア”の 2 つの SIS モジュールのダウンロード方法を説明します。すでにダウンロード済みの場合は、本項は不要です。

1. [コンポーネント]タブを選択し、  アイコンをクリックします。

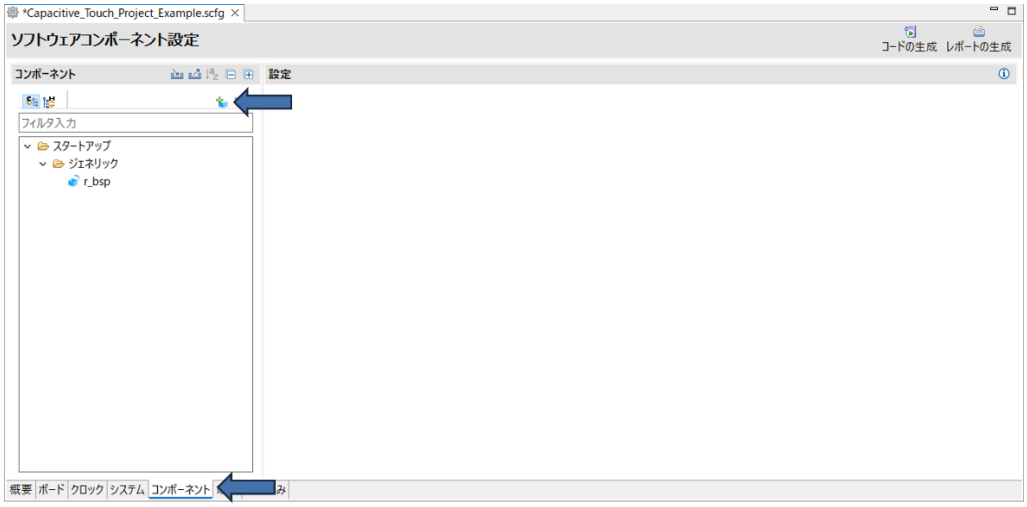


図 7-7　ソフトウェアコンポーネント設定画面

2. “コンポーネントの追加”ダイアログの下側にある”RL78 Software Integration System モジュールをダウンロードする”をクリックします。

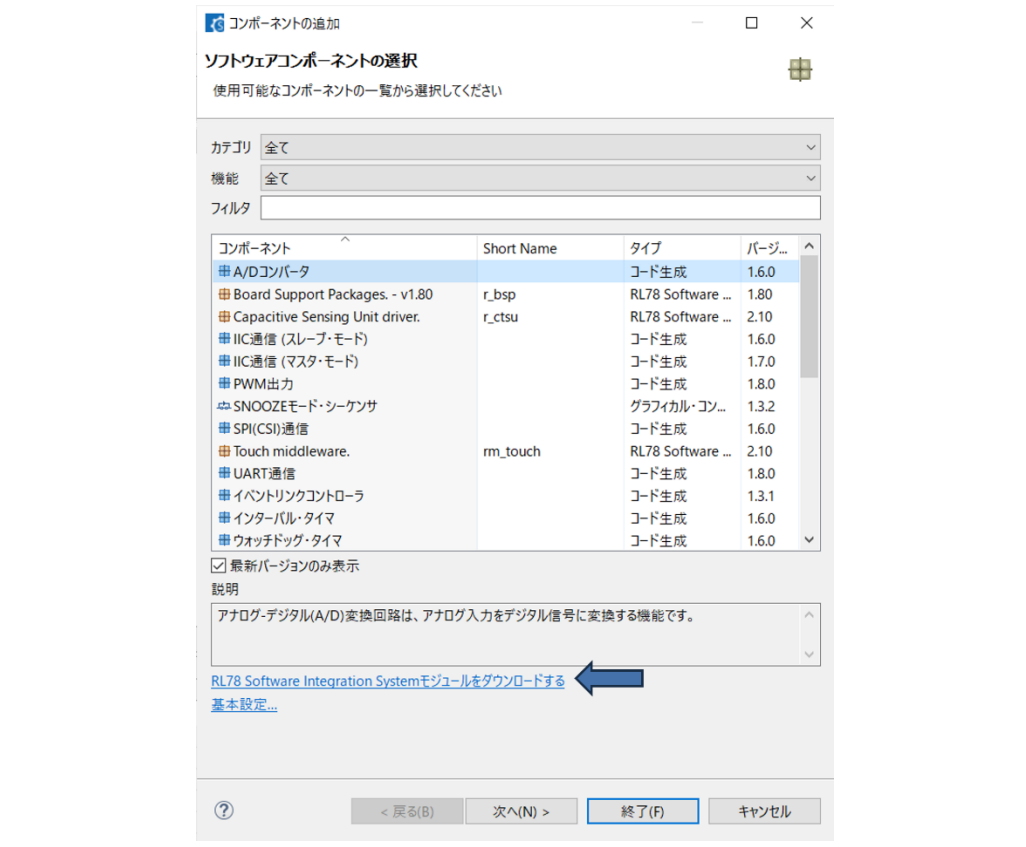


図 7-8　コンポーネントの追加ダイアログ

3. 表示されたダイアログで以下を選択し、ダウンロードをクリックします。
   — RL78 ファミリ CTSU モジュール Software Integration System
   — RL78 ファミリ TOUCH モジュール Software Integration System

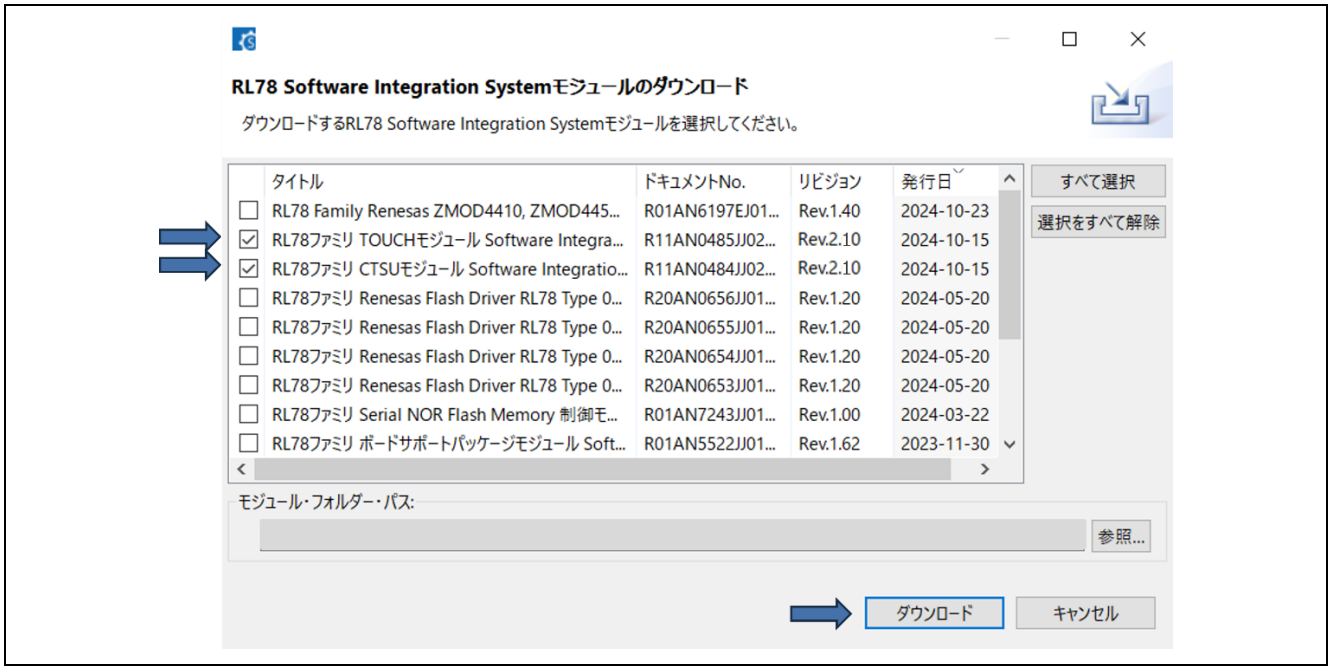


図 7-9　SIS モジュールのダウンロード

注意　上記ダウンロード画面に、TOUCH モジュールおよび CTSU モジュールが表示されない場合は、別の手順でダウンロードする必要があります。
Renesas Web から各モジュールのダウンロードを行い、手順に沿って SIS モジュールのダウンロードフォルダにファイルを追加します。

各モジュールのウェブページおよび適用方法については以下を参照してください。

- CTSU モジュールと TOUCH モジュールのダウンロード

RL78 ファミリ CTSU モジュール Software Integration System
RL78 Family CTSU Module Software Integration System Rev.2.10 - Sample Code | Renesas ルネサス

RL78 ファミリ TOUCH モジュール Software Integration System
RL78 Family TOUCH Module Software Integration System Rev.2.10 - Sample Code | Renesas ルネサス

- 各モジュールの適用方法

Renesas Web 等からダウンロードした Software Integration System モジュールの適用方法 | Renesas Customer Hub

7.4　コンポーネント追加

1.　スマート・コンフィグレータで以下のコンポーネントを選択してください。

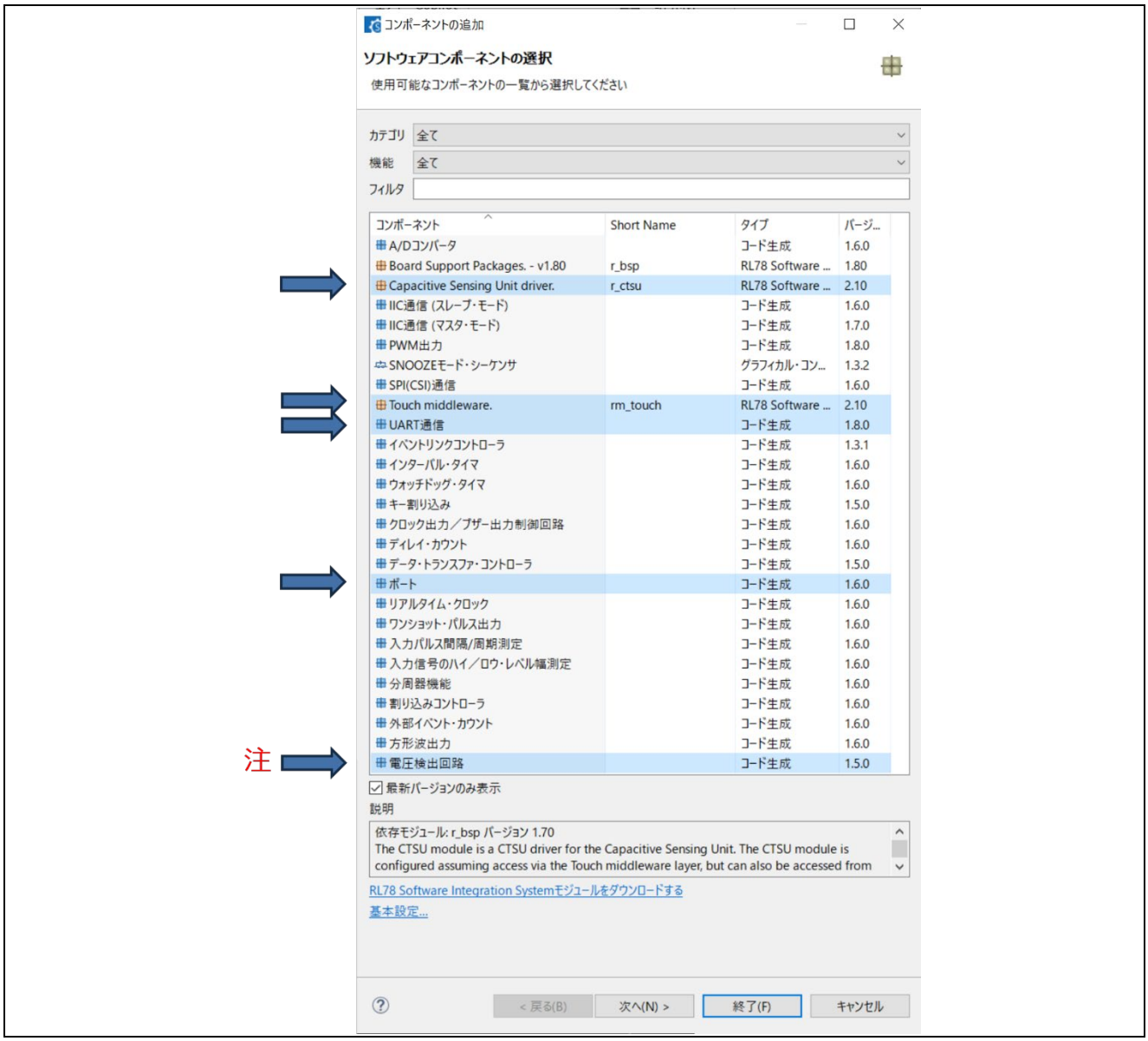


図 7-10　ソフトウェアコンポーネントの選択

注　　RL78/G16 の場合は電圧検出機能の設定方法が異なりますので、コンポーネントで本機能を追加する必要はありません。
RL78/G16 での設定方法は、7.5.4 節をご確認ください。

2. 選択したコンポーネントに対してリソースを設定します。本アプリケーション例では以下の設定で使用します。

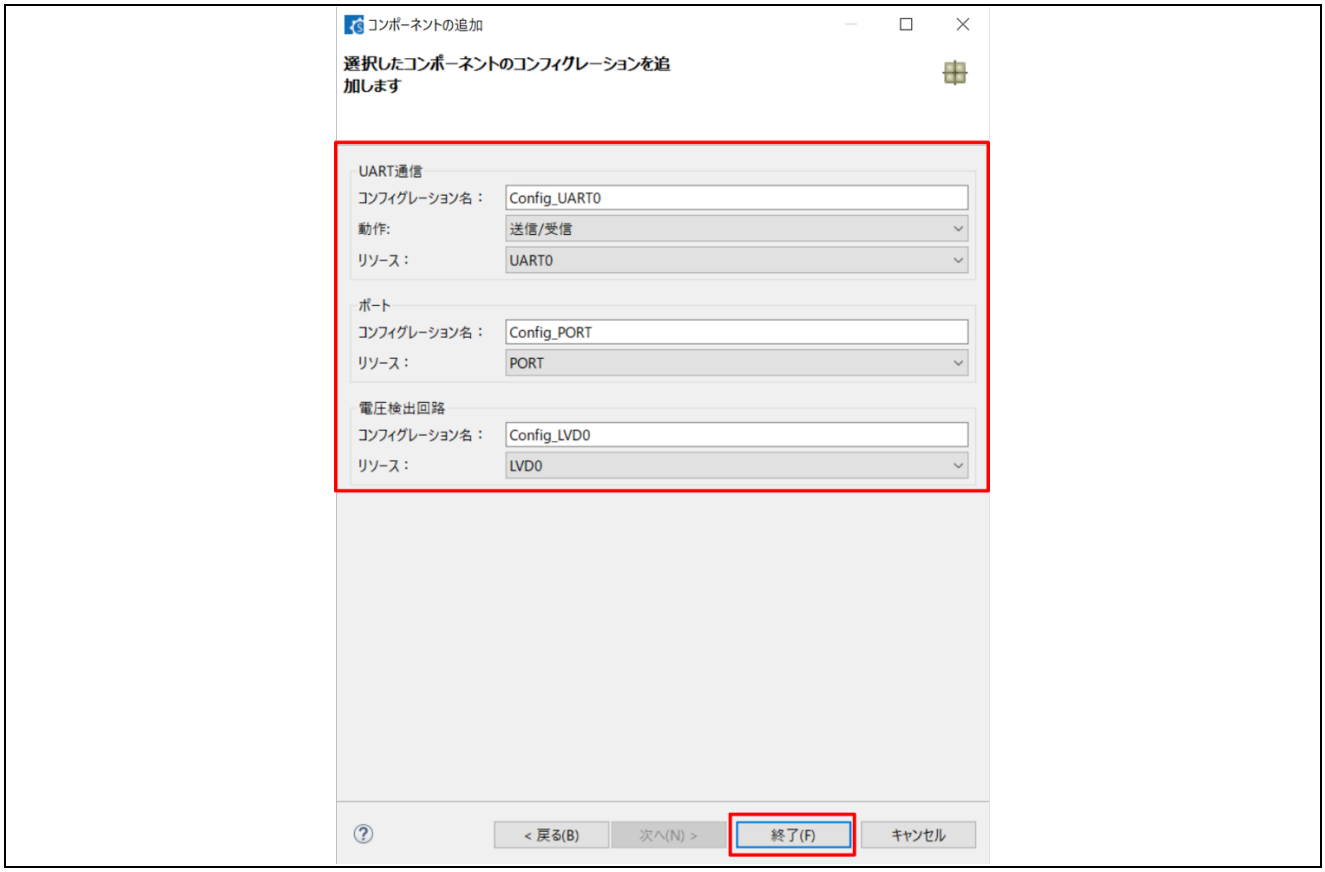


図 7-11　コンポーネントのリソース設定

以下に示すようにコンポーネントが追加されます。

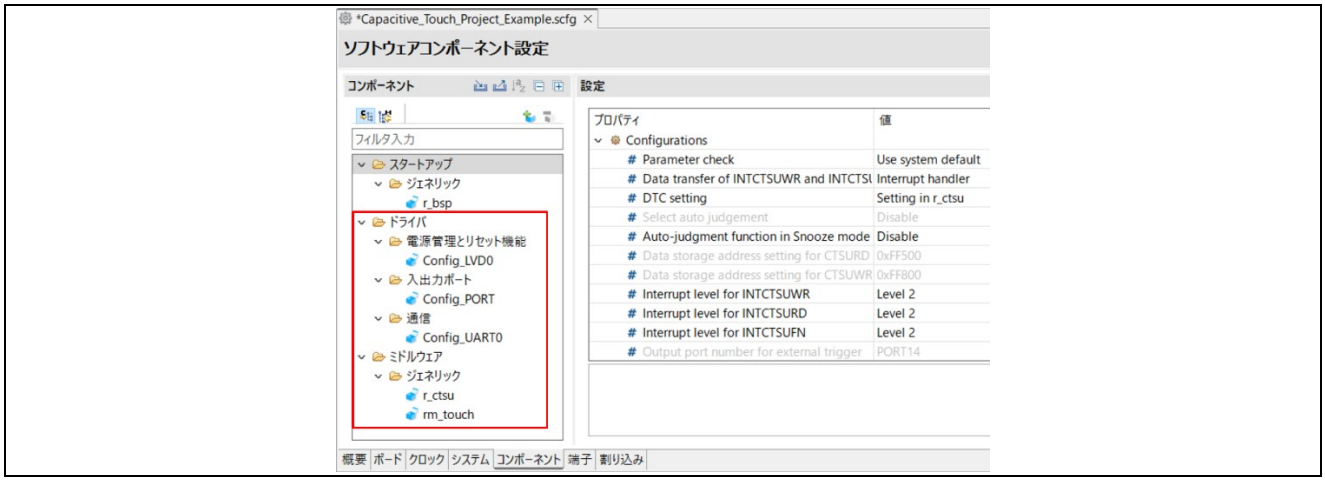


図 7-12　ソフトウェアコンポーネント設定（コンポーネント追加後）

7.5　スマート・コンフィグレータによるコンポーネント設定の変更

追加した各コンポーネントについて設定を行います。

7.5.1　CTSU コンポーネント設定

”r_ctsu”モジュールをクリックします。TSCAP 端子とアプリケーションで使用する 5 つの TS 端子を有効にします。TS 端子とタッチセンサの割り当ては、使用するターゲットボードのユーザーズマニュアルをご確認ください。

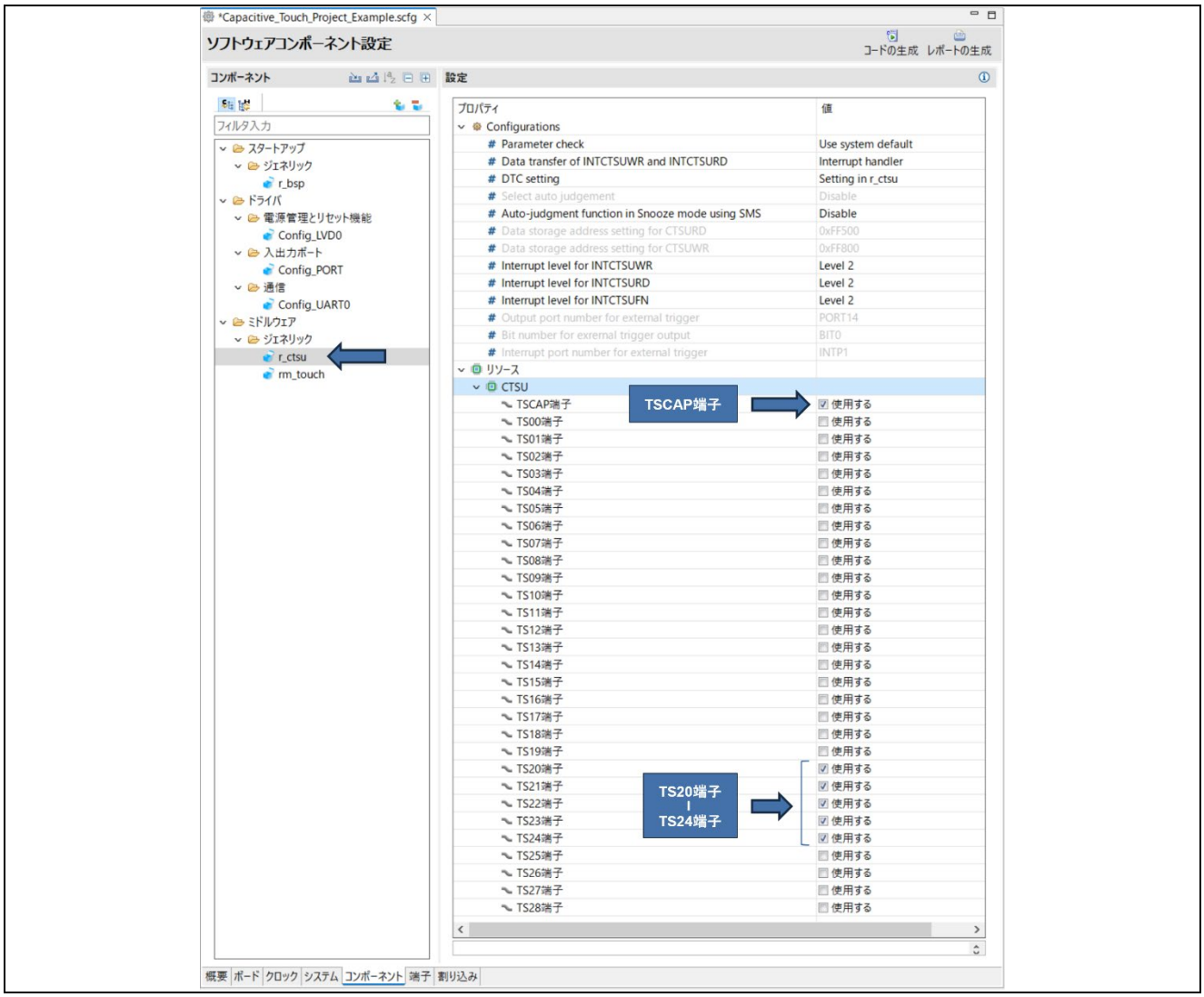


図 7-13　アプリケーションで使用する TS 端子の有効化

また、アプリケーションで使用しない TS 端子については、Low レベル出力に設定することを推奨します。CTSU2 では、アプリケーションで使用しない TS 端子を有効にした場合、非計測端子として、Low レベル出力の設定になります。

そのため、付属の 2 種類のサンプルコードでは、兼用機能 (UART0) を割り当てて使用する TS12 / TS13 端子を除き、アプリケーションで使用しない TS 端子についてもすべて有効にしています。

注意　回路を作成する際は、端子処理などを適切に行い、電気的特性を満たすようにしてください。

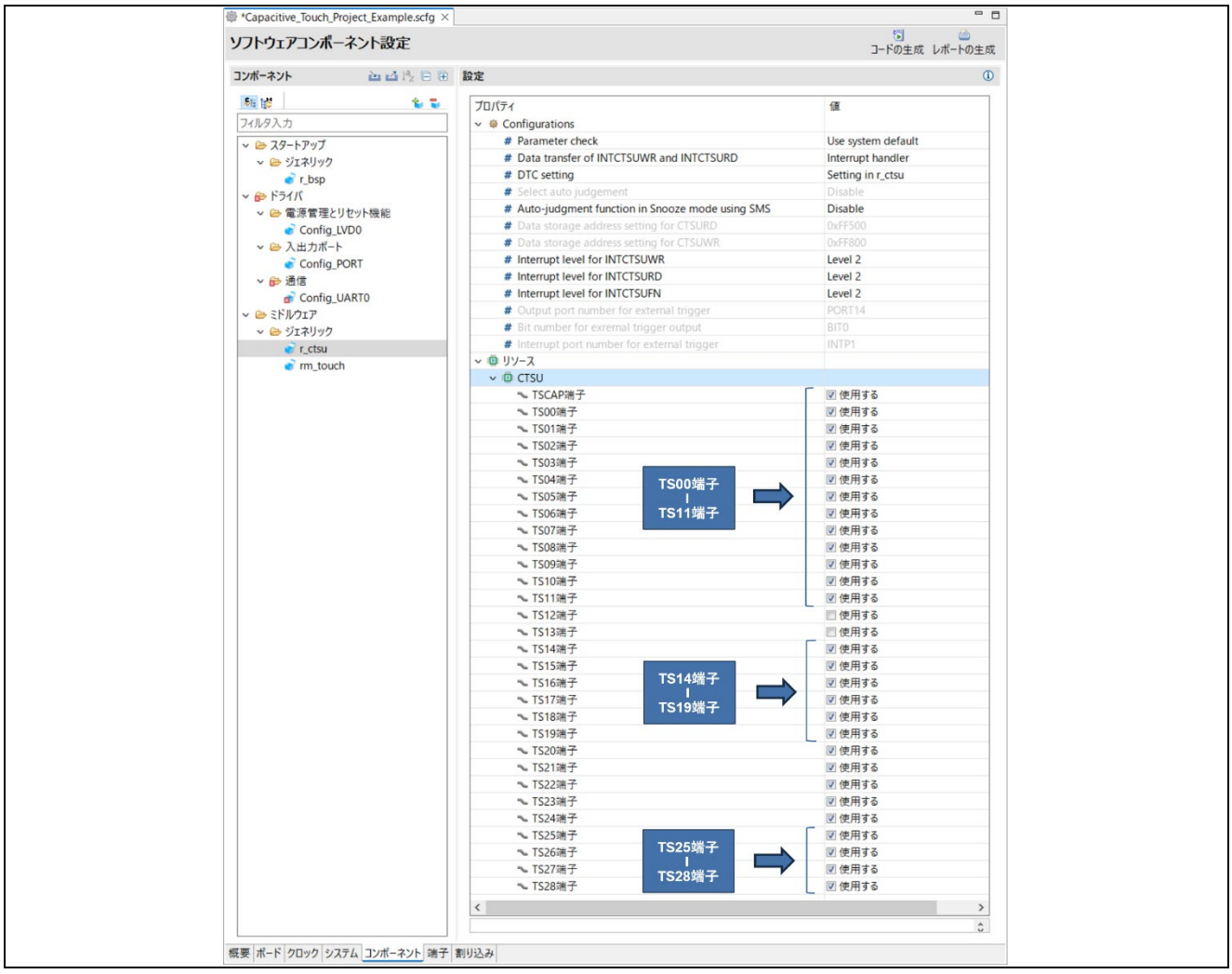


図 7-14　アプリケーションで使用しない TS 端子の有効化

7.5.2　Touch コンポーネント設定

”rm_touch”モジュールをクリックし、以下を設定します。

— Support QE monitor using UART　Enable

— Support QE tuning using UART　Enable

— UART channel　UART0

設定する UART チャネルは、使用するターゲットボードによって異なります。詳細は使用する FPB の回路図をご確認ください。

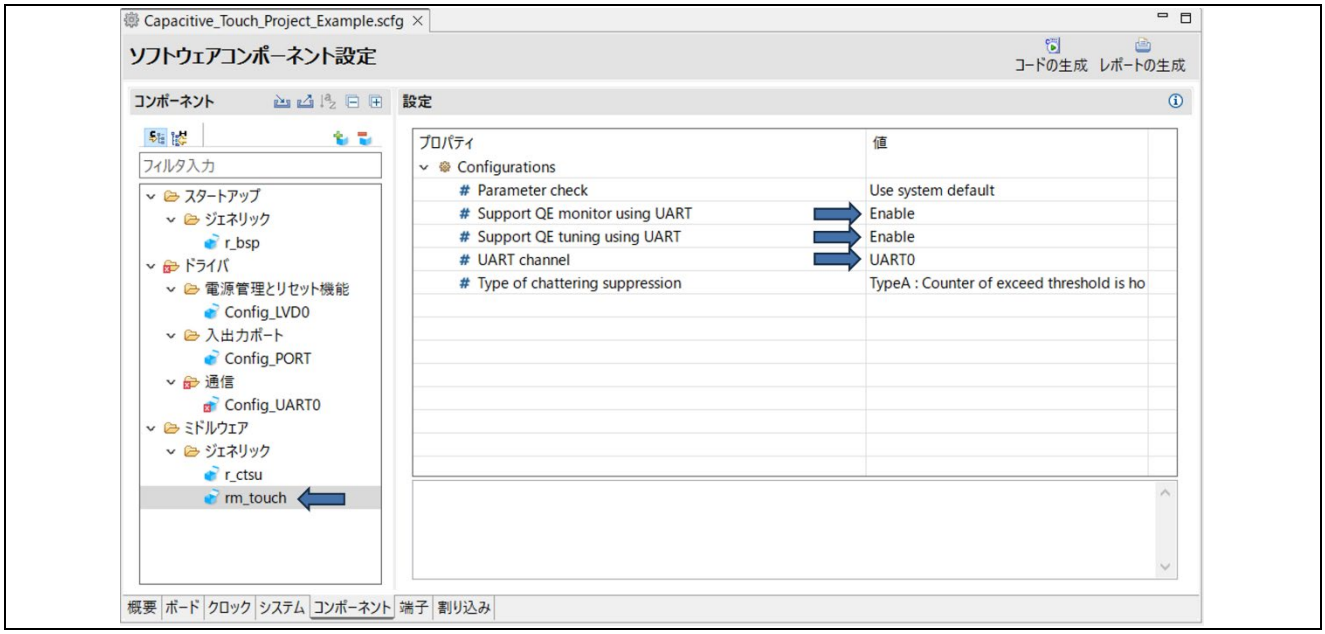


図 7-15　rm_touch の設定

7.5.3　UART 通信コンポーネント設定

タッチセンサのチューニング、モニタリングで使用する UART の設定手順を説明します。

設定する UART チャネルおよびポートは、使用するターゲットボードによって異なります。追加した"UART 通信"モジュールをクリックし、目標の転送レート (ボーレート) に合わせて "送信" および "受信" タブにて動作クロックと転送レートを選択します。

本アプリケーションノートでは、以下のように設定します。

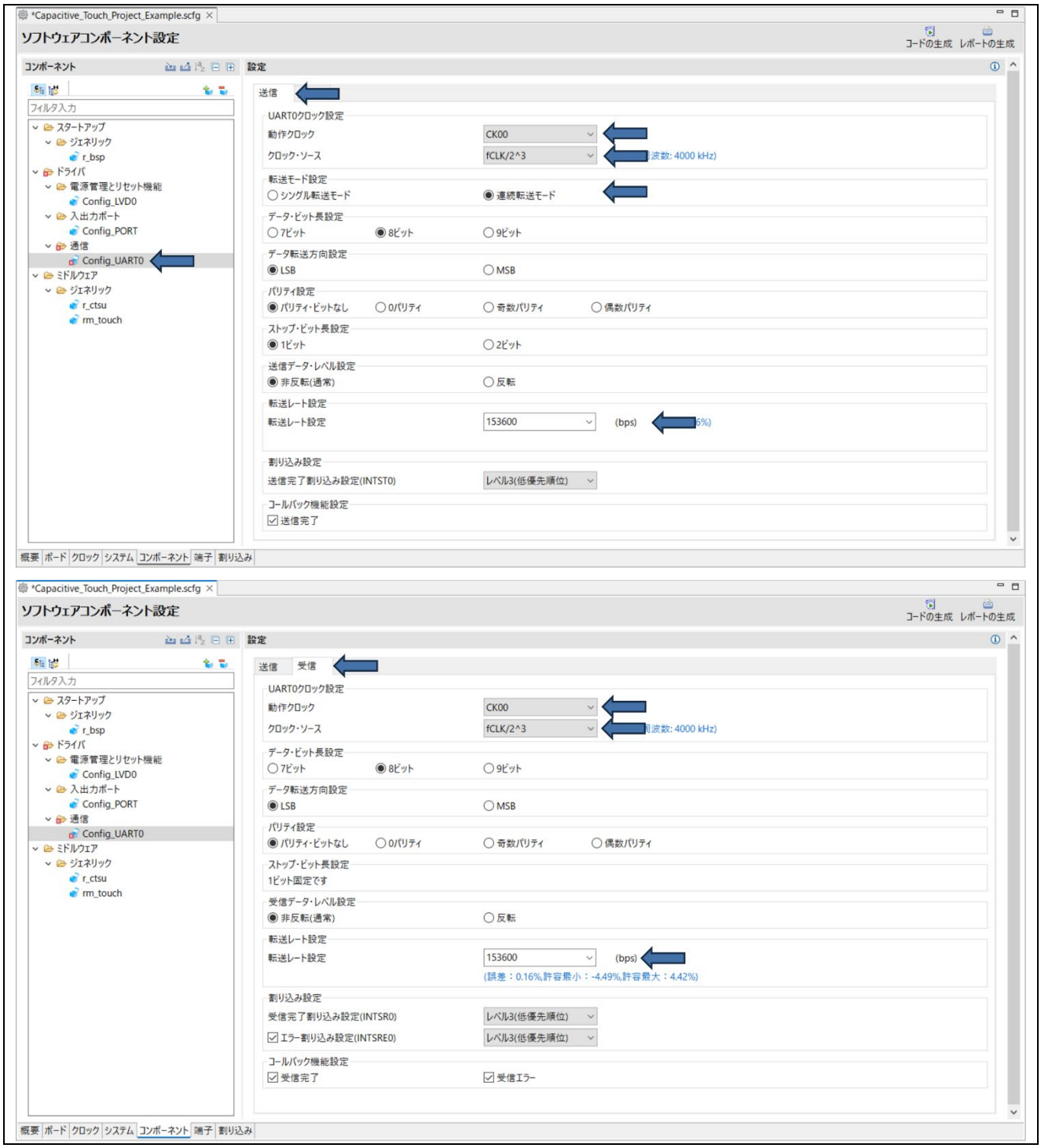


図 7-16　UART 通信コンポーネントの設定 (UART0)

次に、[端子]タブを選択し、UART 通信機能に使用する端子を設定します。

本アプリケーションでは、UART0 (SAU00) に TOOLTxD, TOOLRxD を兼用する端子をそれぞれ割り当てます。RL78/G22 では、以下の端子番号を設定します。

—RxD0　　: 21

—TxD0　　: 20

注意　本設定は使用するデバイスによって異なります。UART 通信機能を利用できる端子は、使用する FPB の回路図にてご確認ください。割り当てる端子番号は、使用する RL78 製品のユーザーズマニュアル ハードウェア編の「1.3 端子接続図」にてご確認ください。

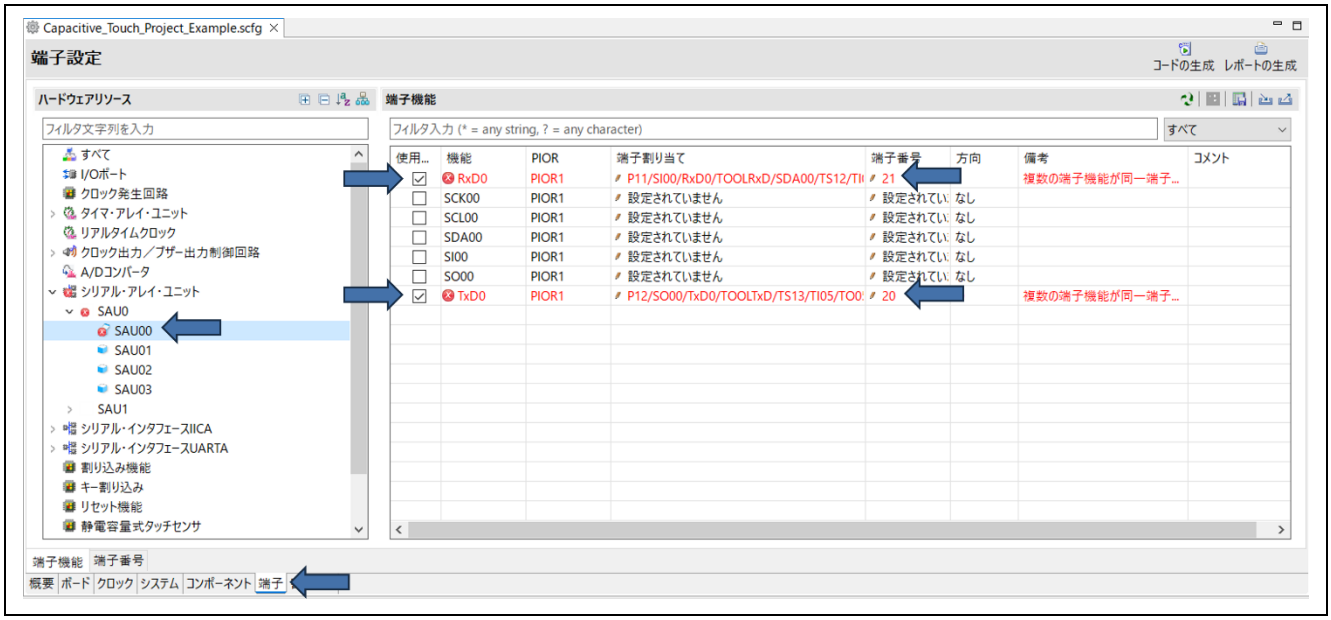


図 7-17　UART チャネル (UART0) の端子割り当て

使用するツールバージョンによっては UART0 の端子割り当てエラーが発生しますが、以下の理由から本エラーは無視してください。
本アプリケーションノートでは、下記の端子をそれぞれ異なる用途で使用します。

① TOOLRxD/TOOLTxD 端子 :　　CS+で作成したプログラムを COM Port 経由で書き込むとき
② RxD0/TxD0 端子 :　　スタンドアロン版 QE を用いたシリアル通信を行うとき

これらの端子機能は同一端子に割り当てられており、兼用機能となるため、スマート・コンフィグレータ上では端子の競合エラーが出力されます。しかし、①および②を同時に使用することはありませんので、実使用上は、端子は競合しません。よってこのまま使用して問題ありません。

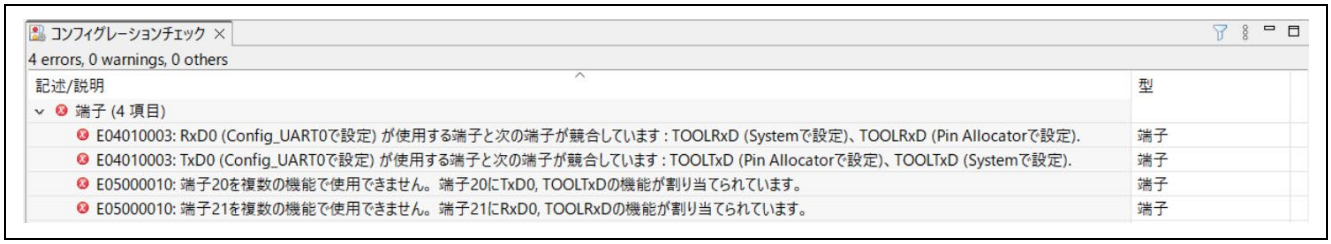


図 7-18　UART0 の端子割り当てエラー

7.5.4　LVD コンポーネント設定

ユーザ・オプション・バイト の電圧検出回路 0（LVD0）を設定します。

”LVD0”モジュールをクリックし、動作モードと検出電圧の設定を行います。リセット発生電圧 (VLVD0) を 2.62 (V)に設定します。

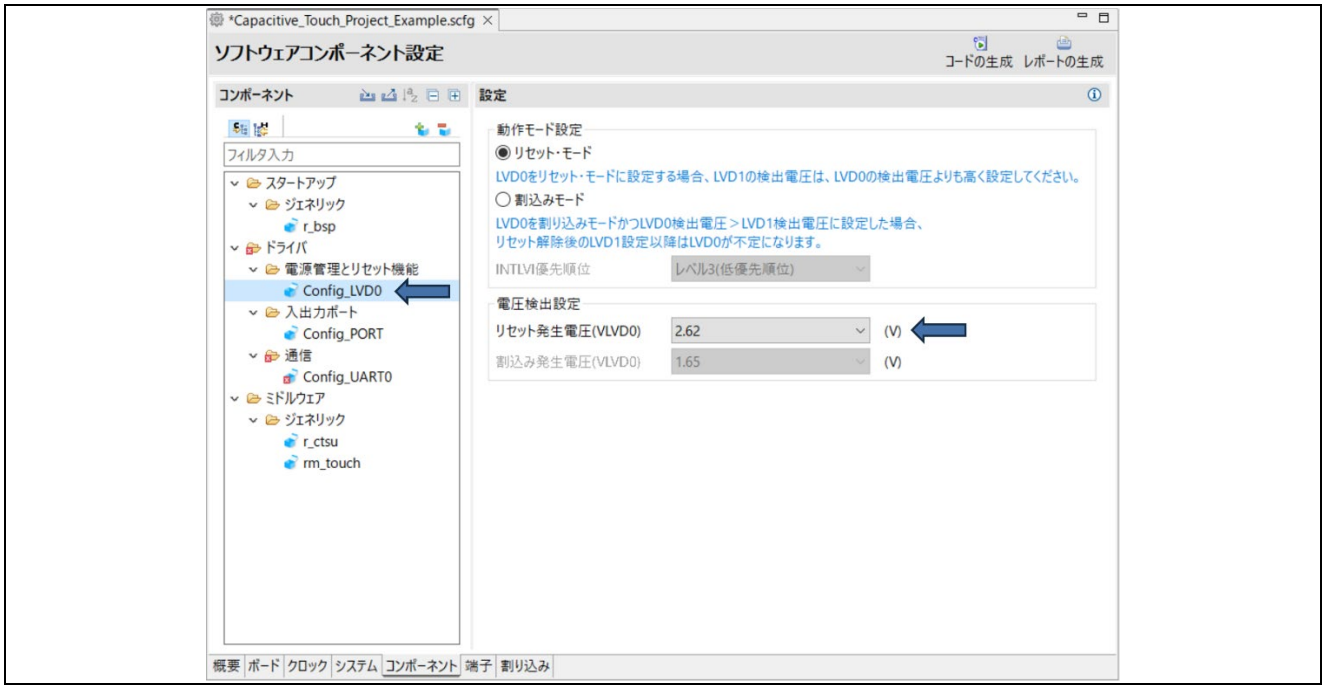


図 7-19　LVD コンポーネントの設定 (LVD0)

注意　RL78/G16 の場合は、電圧検出機能にセレクタブル・パワーオン・リセット回路を使用します。
本機能は [システム] タブからリセット発生電圧を設定します。設定内容を図 7-20 に示します。

図 7-20 リセット発生電圧の設定 (RL78/G16)

7.5.5　PORT コンポーネント設定

LED に接続されるポートの設定を行います。

本アプリケーション例では P62 と P63 を High レベル出力に設定します。

LED を制御するポートについては、使用するターゲットボードの回路図をご確認ください。

1. ”PORT”モジュールをクリックし、”PORT6”にチェックを付けます。

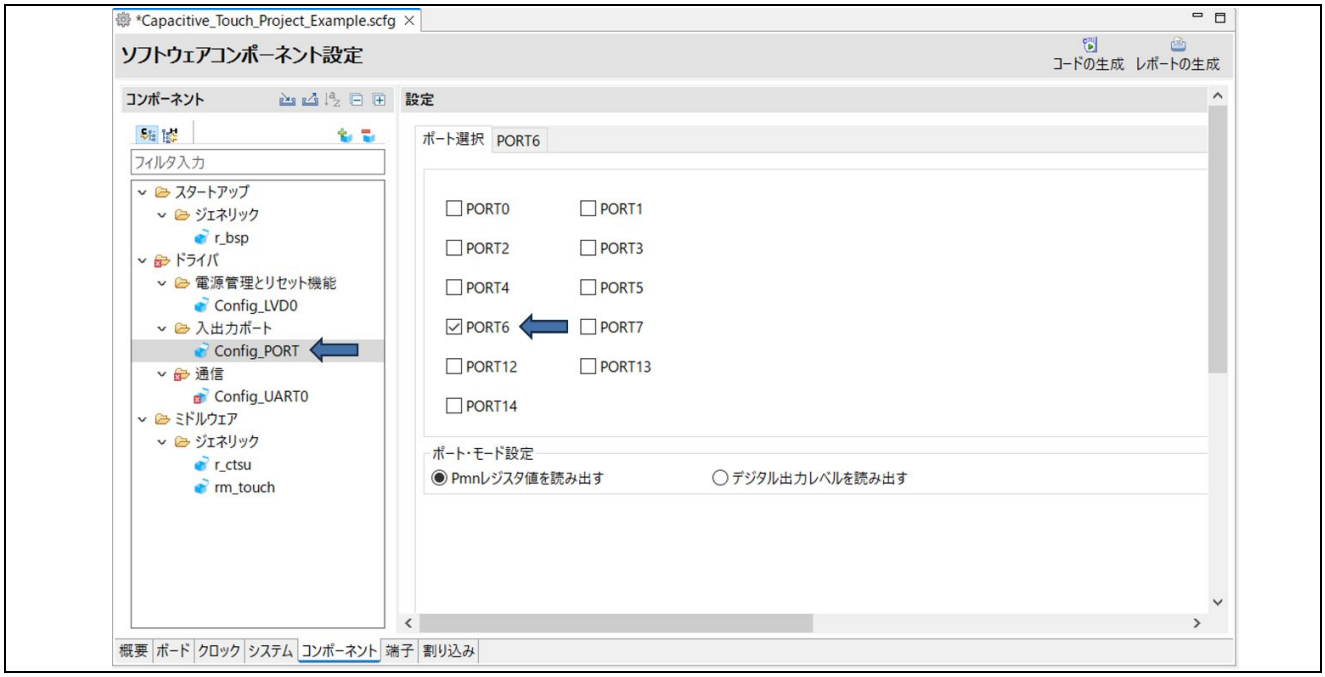


図 7-21　PORT コンポーネントの設定

2. 起動時にターゲットボード上の LED1, LED2 が点灯しないようにポートを設定します。
   [PORT6]タブをクリックし、”P62”と”P63”を High 出力に設定します。

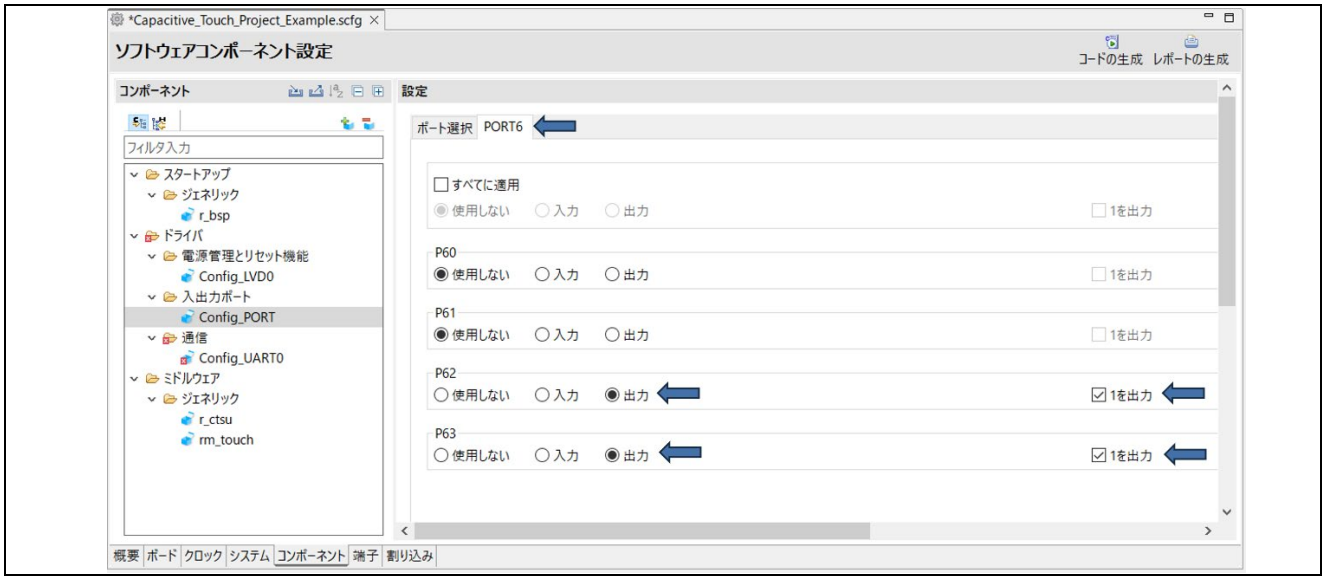


図 7-22　P62 と P63 を High 出力に設定

7.5.6　ボードサポートパッケージ

“r_bsp”モジュールを選択し、”Initialization of peripheral functions by Code Generator/Smart Configurator”が”Enable”に設定されていることを確認してください。

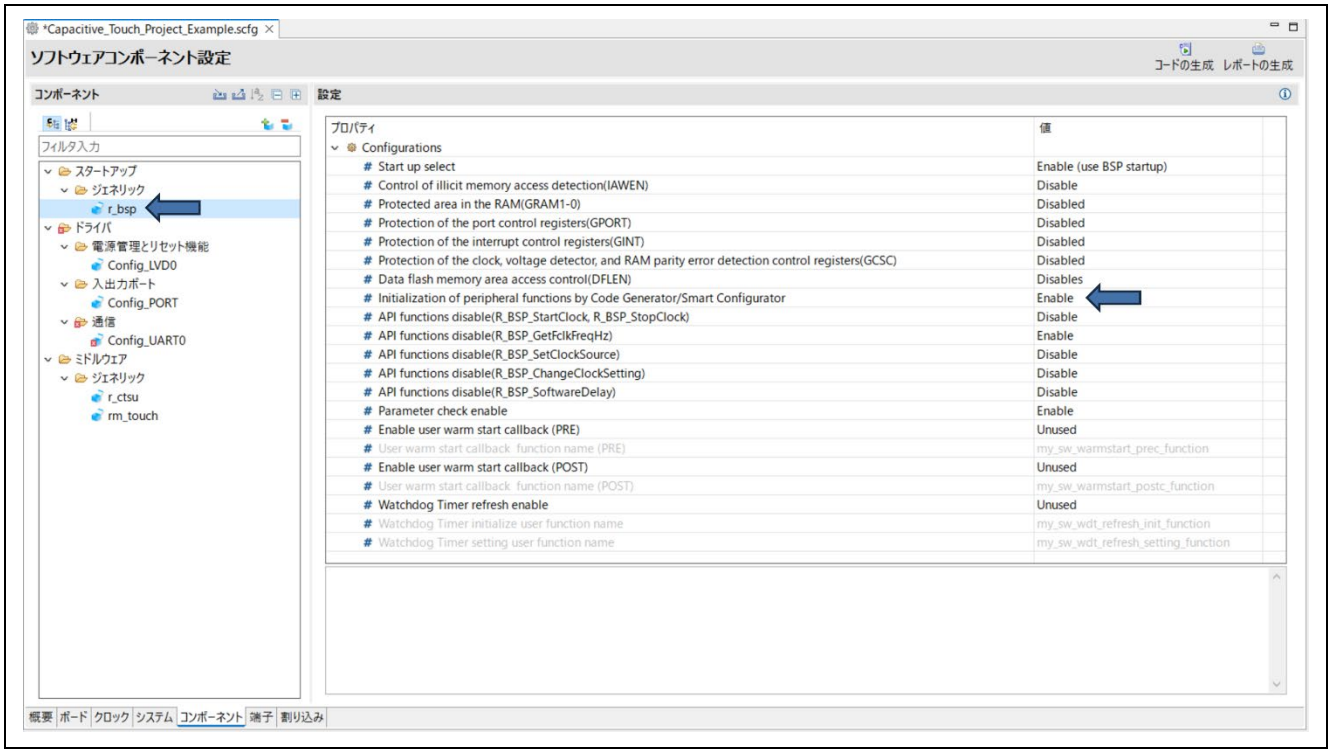


図 7-23　r_bsp の設定

7.6　未使用端子の設定

使用しない端子を Low レベル出力に設定することを推奨します。

本節では、例として”PORT41”を Low レベル出力に設定する手順を説明します。

注意　回路を作成する際は、端子処理などを適切に行い、電気的特性を満たすようにしてください。

1. ”ポート”モジュールをクリックし、”PORT4”にチェックを付けます。

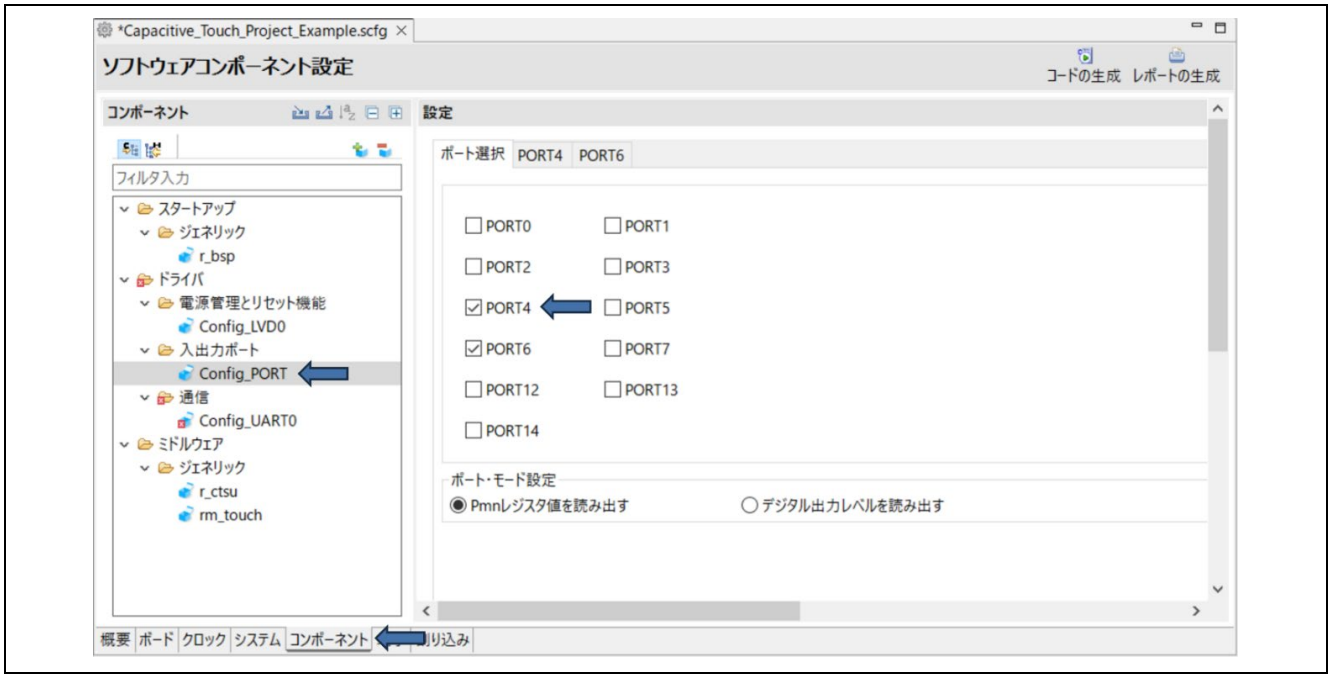


図 7-24　ポートモジュールの設定

2. [PORT4]タブをクリックし、”P41”を Low 出力に設定します。

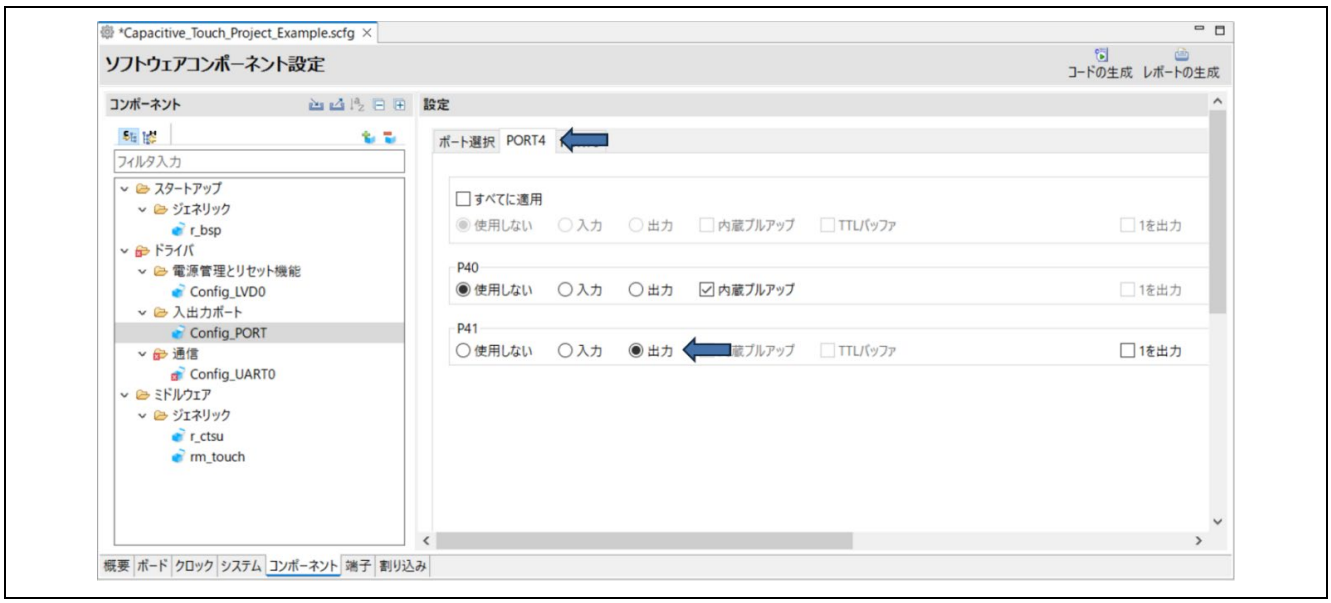


図 7-25　P41 を Low 出力に設定

7.7　コード生成

スマート・コンフィグレータの  アイコンをクリックし、コードの生成を行います。
コード生成時に注意文が表示されますが、無視してコード生成を続行してください。

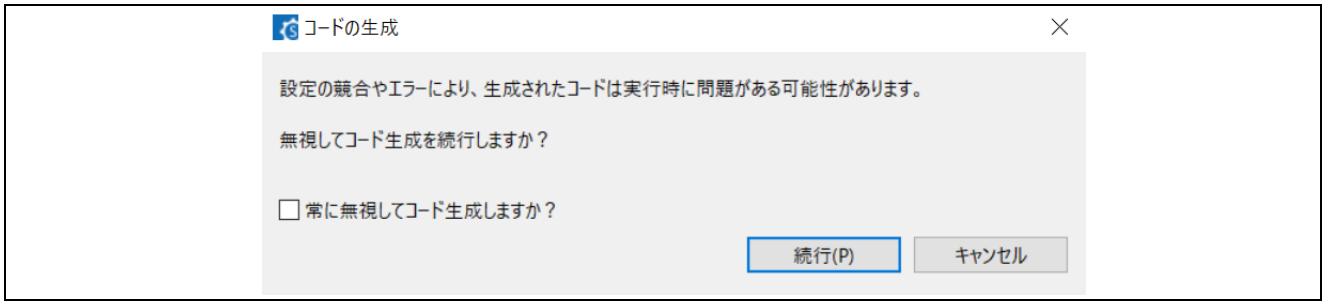


図 7-26　コード生成時の注意画面

スマート・コンフィグレータでオンチップ・デバッグ設定またはオプション・バイト設定を変更した場合、以下のメッセージが表示されることがあります。変更内容を確認した後に、[OK]をクリックします。

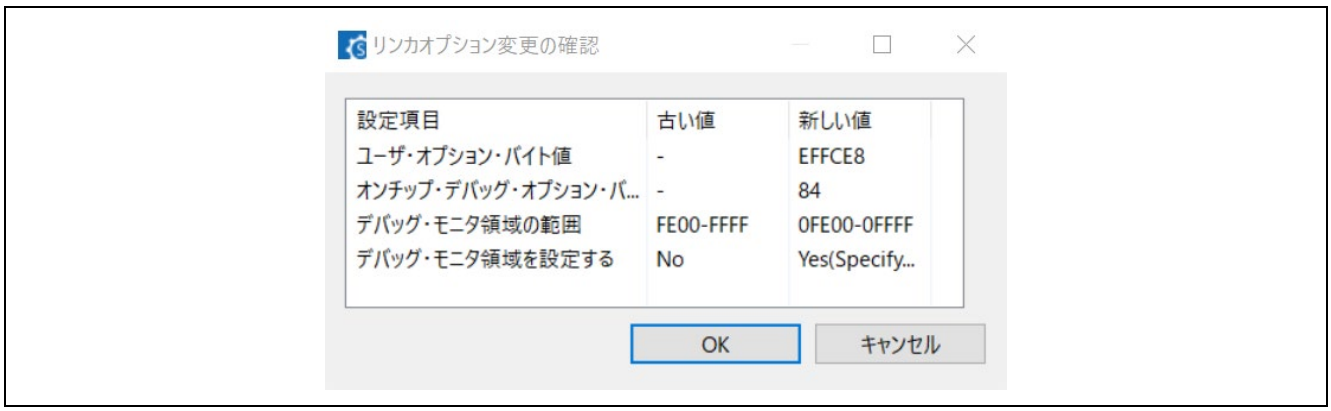


図 7-27　リンカオプション変更の確認

8.　QE for Capacitive Touch の設定

8.1　QE for Capacitive Touch の起動

スタンドアロン版 QE (以下、QE) を起動します。

1. “QE-CapTouch (QE のインストールフォルダ) / eclipse / qe-captouch.exe”より QE を起動します。

2. 起動後の画面を以下に示します。

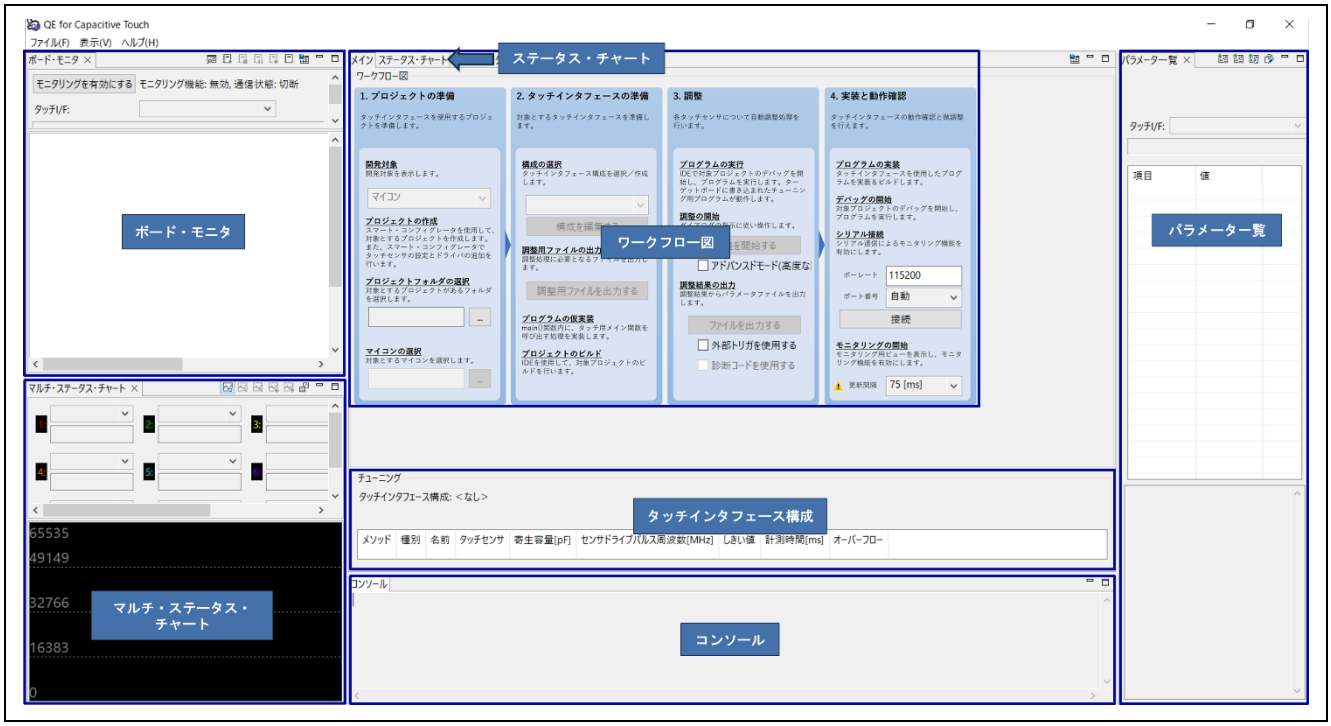


図 8-1　QE 起動後画面

全画面表示にした際に表示が崩れる場合は、Windows 設定よりディスプレイの”拡大と縮小レイアウト”を”100%”に設定してください。

8.2　プロジェクトの準備

タッチインタフェースを使用するプロジェクトを準備します。

QE 起動後画面中央にあるワークフロー図の “プロジェクトの準備” に従い、設定します。

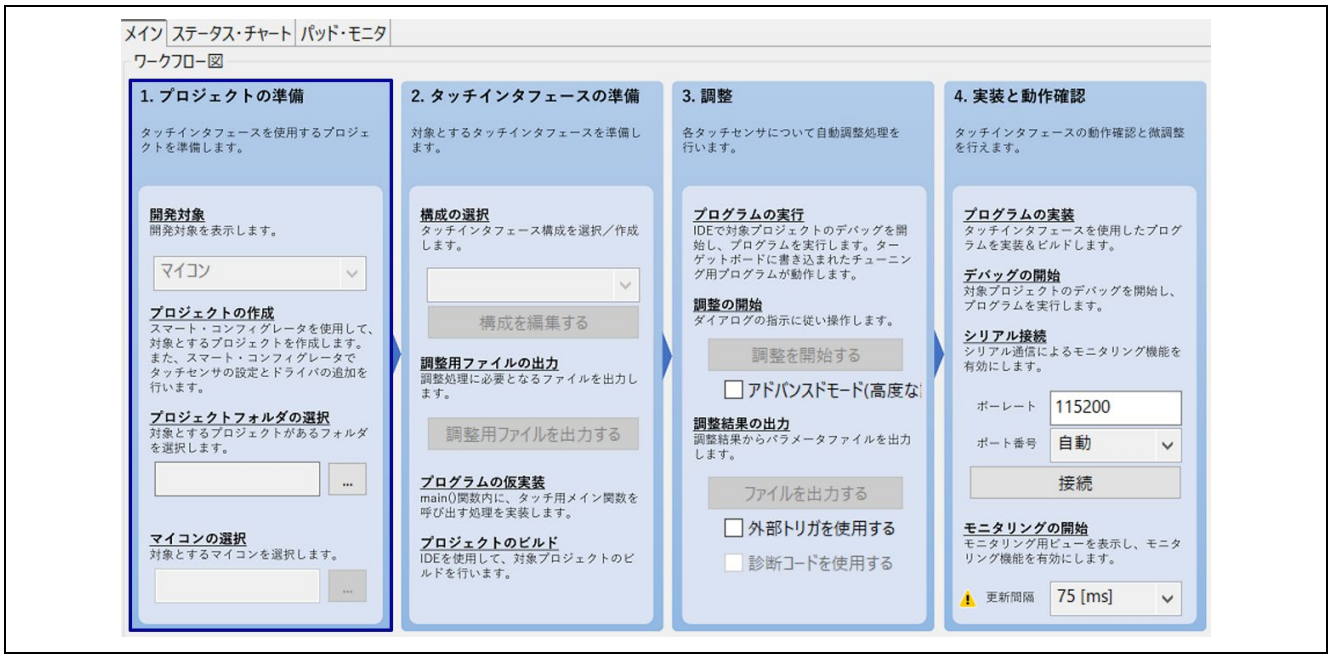


図 8-2　ワークフロー図 (プロジェクトの準備)

1. “プロジェクトフォルダの選択” 下の [···] をクリックし、CS+で作成したプロジェクトフォルダを選択します。

2. “マイコンの選択” 下の [···] をクリックし、使用するマイコンを選択します。

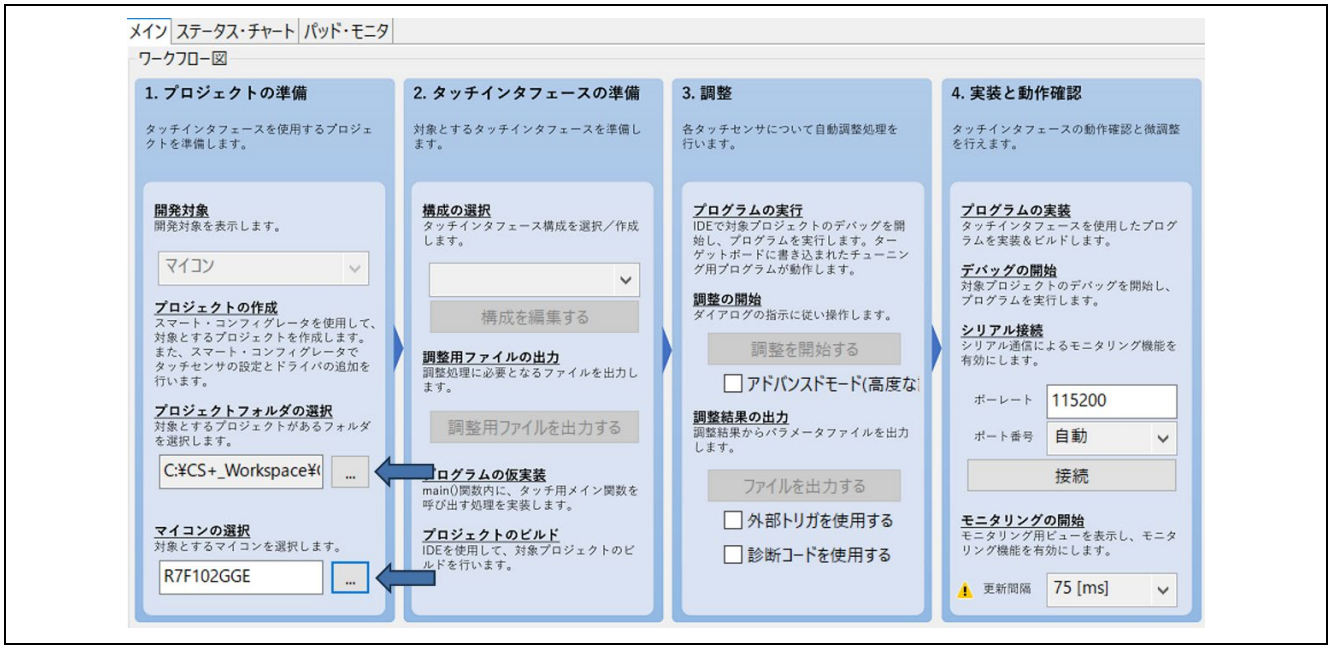


図 8-3　プロジェクトの準備

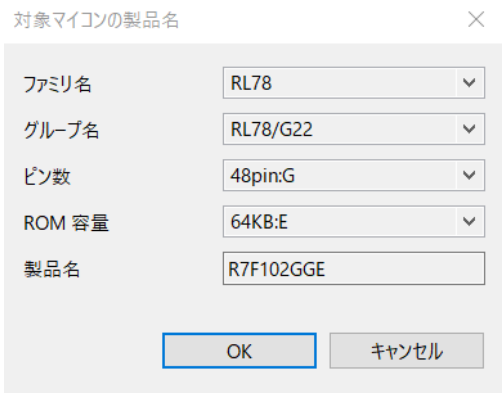


図 8-4　マイコンの選択

“マイコンの選択”で、以下のようなエラーが出た場合は、QE のインストールフォルダの格納場所に問題がある可能性があります。一度 QE を終了し、フォルダを C:¥Renesas フォルダ以下などに移動させたのち、QE を起動させてください。

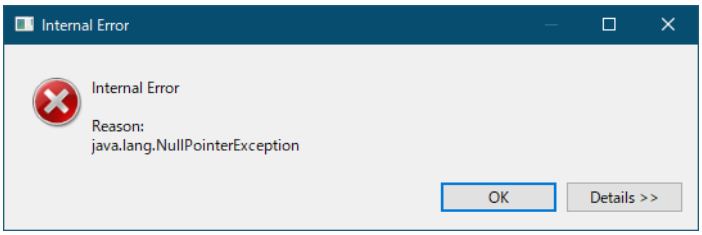


図 8-5　マイコンの選択のエラー

8.3　タッチインタフェースの準備

ワークフロー図の“タッチインタフェースの準備”に従い、設定します。

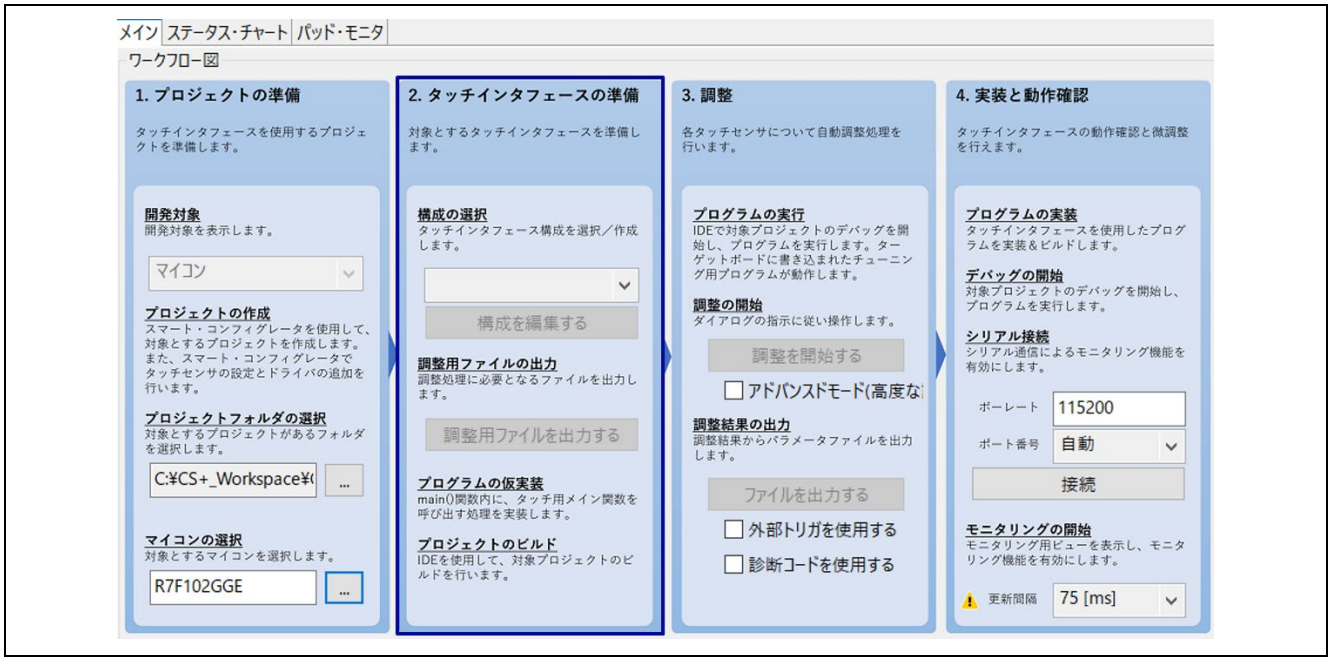


図 8-6　ワークフロー図 (タッチインタフェースの準備)

1. “構成の選択”下の をクリックし、“タッチインタフェース構成の新規作成”を選択します。

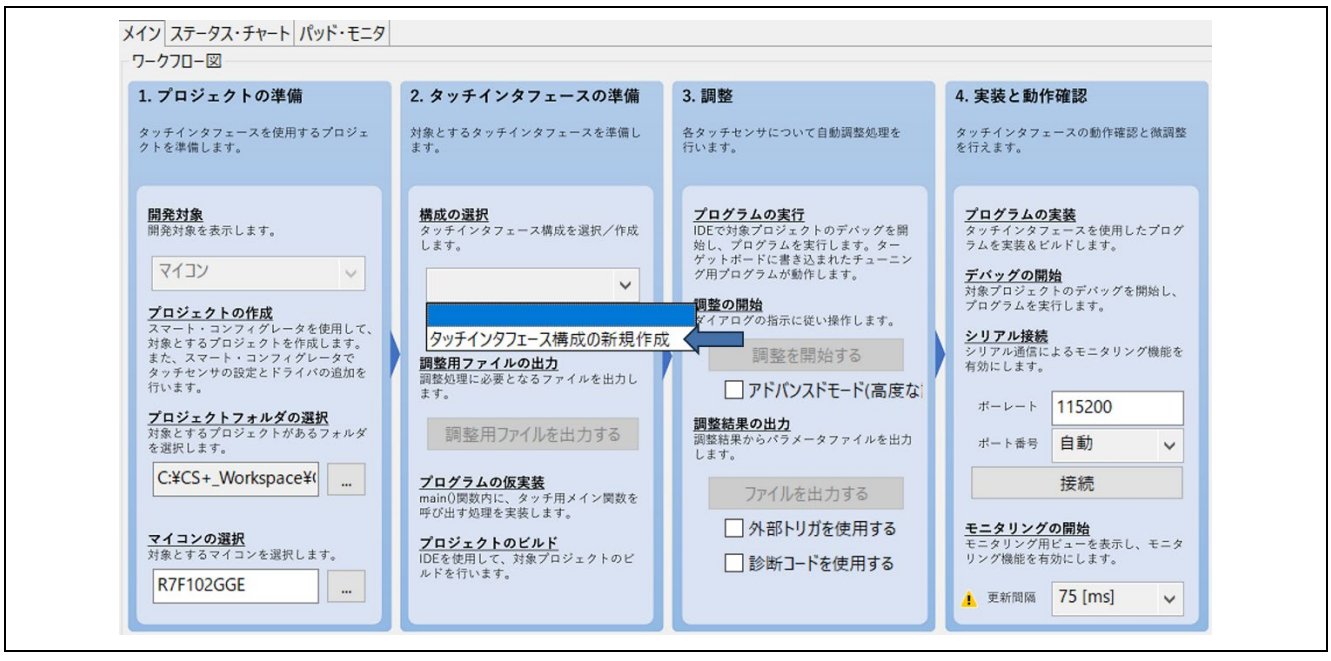


図 8-7　タッチインタフェース構成の新規作成

2. “タッチインタフェース構成の作成”ウィンドウが開き、タッチインタフェースを配置する領域が表示されます。
   右側の“タッチ I/F”パネルから[ボタン]をクリックすると、カーソルがボタンを配置できる状態になり、配置領域でクリックするとボタンを配置できます。
   2 つのボタン (Button00/Button01) を配置し、[ESC]キーを押してボタンの配置を終了します。
   同様に、“タッチ I/F”パネルから[スライダ (横方向)]をクリックして、スライダ (Slider00) を配置します。

   タッチインタフェースの追加を終了すると図 8-8 の状態になります。

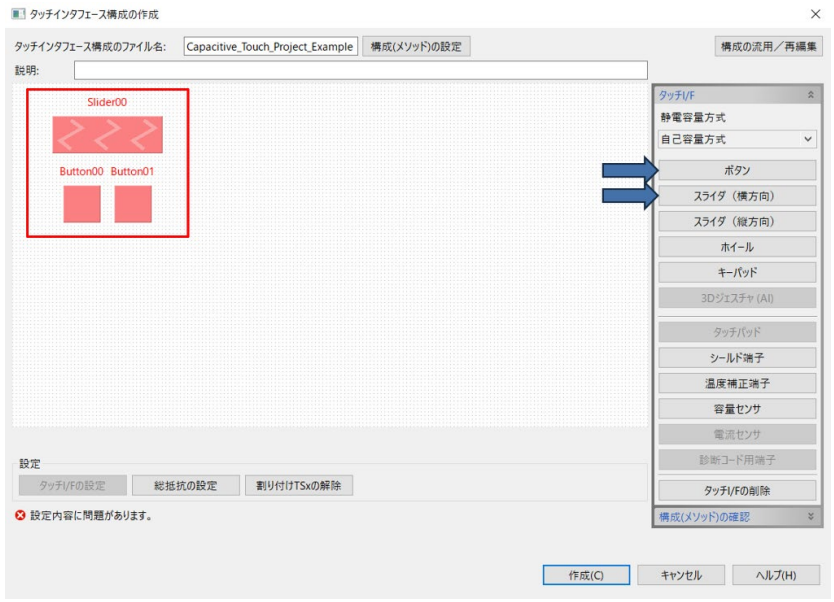


図 8-8　ボタンとスライダの配置

3. 各ボタンの名前とタッチセンサの割り当てを行います。
   配置した[Button00]をダブルクリックし、表示された”タッチインタフェースの設定”ダイアログで、以下のように設定します。
   — タッチセンサ　　: TS24
   — 抵抗値[Ω]　　: 560

   抵抗値は、使用するターゲットボードのユーザーズマニュアル、または回路図をご確認ください。

注意　ボタンにタッチセンサを割り当てる際に TS 端子の表示が消えてしまう場合は、Windows 設定を変更する必要があります。Windows のシステム設定からディスプレイの”拡大と縮小レイアウト”を”100%”に設定後、QE for Capacitive Touch を再起動してください。

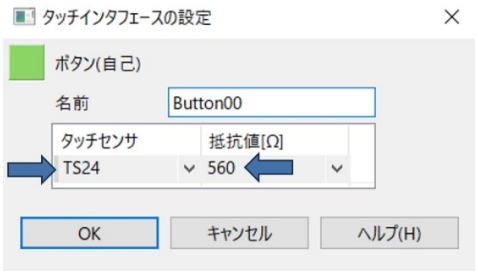


図 8-9　タッチインタフェースの設定 (ボタン)

4. 同様にして[Button01]を以下のように設定します。
   — タッチセンサ　　: TS23
   — 抵抗値[Ω]　　: 560

5. 同様にして[Slider00]を以下のように設定します。
   — タッチセンサ　　: TS20
   　　: TS21
   　　: TS22
   — 抵抗値[Ω]　　: 560

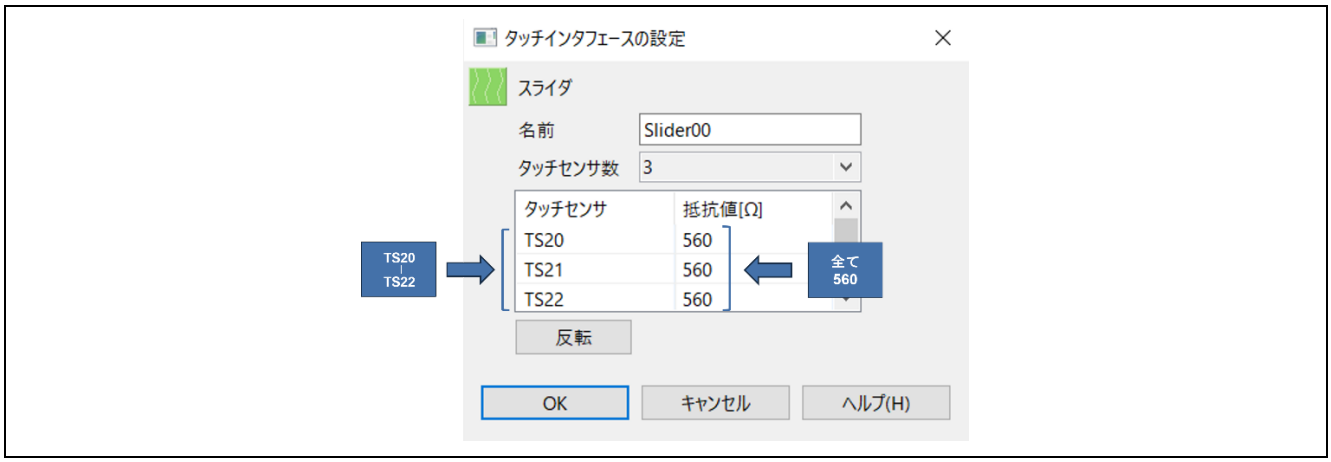


図 8-10　タッチインタフェースの設定 (スライダ)

6. タッチインタフェースが設定されると、図 8-11 のように電極が緑色の表示へと変化します。このまま [作成] をクリックします。

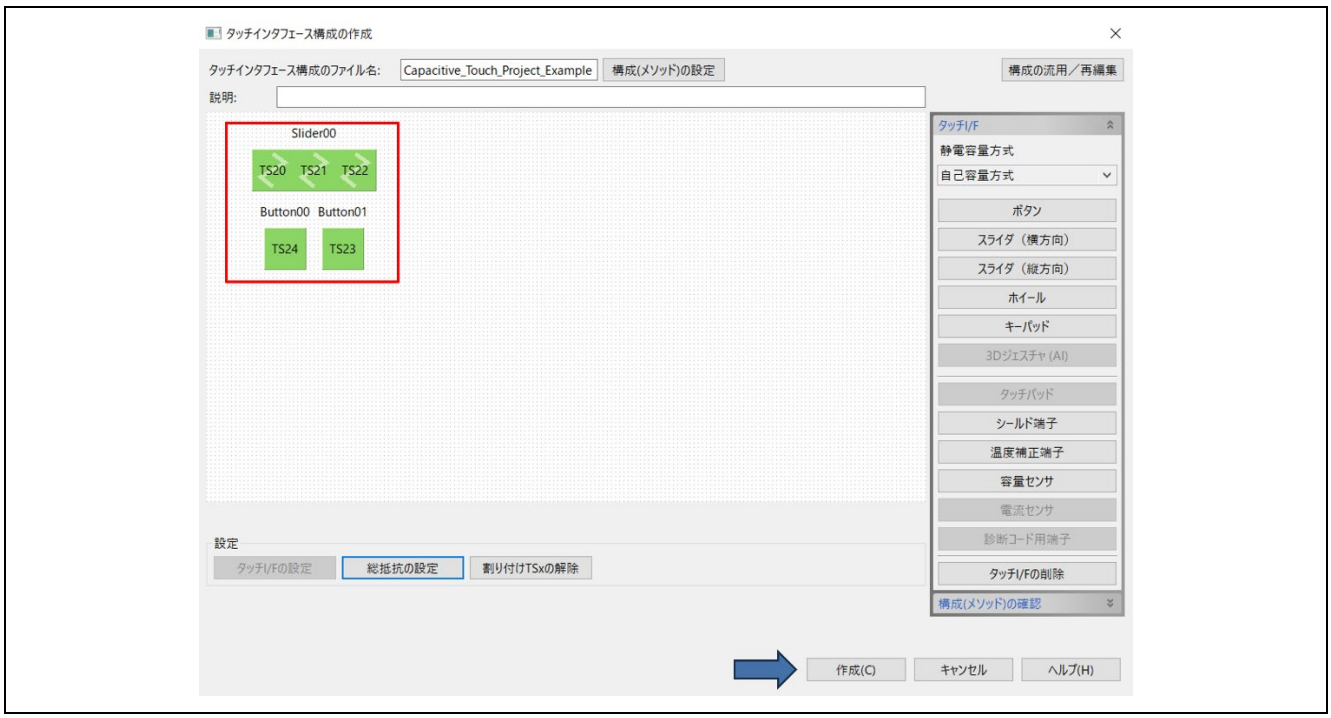
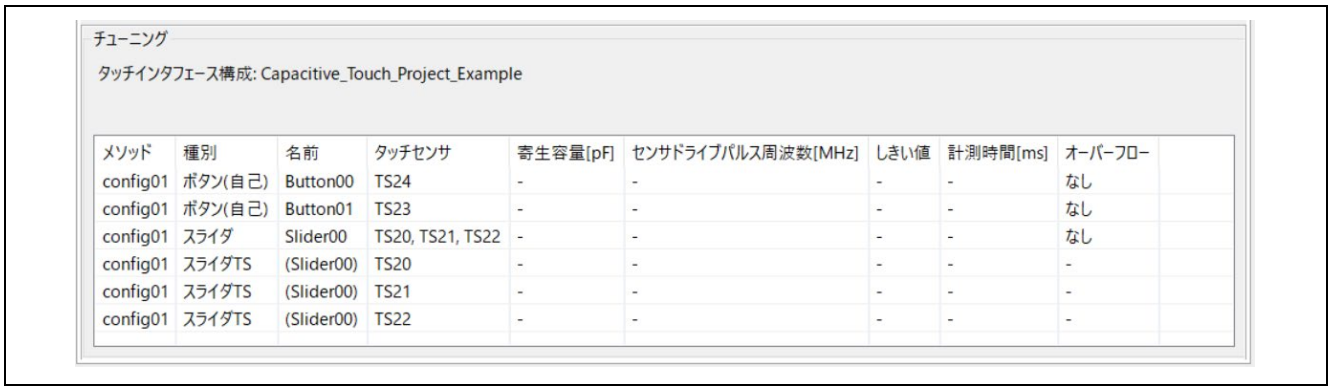


図 8-11　タッチインタフェース設定後の構成

7. “チューニング”パネルに”タッチインタフェースの構成”が表示されます。



メソッド	種別	名前	タッチセンサ	寄生容量[pF]	センサドライブパルス周波数[MHz]	しきい値	計測時間[ms]	オーバーフロー
config01	ボタン(自己)	Button00	TS24	-	-	-	-	なし
config01	ボタン(自己)	Button01	TS23	-	-	-	-	なし
config01	スライダ	Slider00	TS20, TS21, TS22	-	-	-	-	なし
config01	スライダTS	(Slider00)	TS20	-	-	-	-	-
config01	スライダTS	(Slider00)	TS21	-	-	-	-	-
config01	スライダTS	(Slider00)	TS22	-	-	-	-	-

図 8-12　タッチインタフェース構成とチューニングパネル

8. 調整処理に必要なファイルを出力するためのフォルダを作成してください。本アプリケーション例では “Capacitive_Touch_Project_Example/src” の下に、新規で “qe_gen”という名前でフォルダを作成しています。QE ワークフロー図中の[調整用ファイルを出力する]をクリックした後、出力先のフォルダに作成したフォルダを選択してください。

図 8-13　“qe_gen”フォルダ新規作成

以下が出力されるファイルを含めたフォルダ構成です。

```
Capacitive_Touch_Project_Example      ← CS+プロジェクトフォルダ (6.新規プロジェクトの作成
  |- src                                 章で設定した任意のプロジェクト名)
    |- smc_gen
    |- qe_gen                         ← 新規作成フォルダ
      |- qe_touch_config.c            ← 出力ファイル
      |- qe_touch_config.h            ← 出力ファイル
      |- qe_touch_define.h            ← 出力ファイル
      |- qe_touch_sample.c            ← 出力ファイル
```

9. CPU および周辺ハードウェアの周波数を設定します。出力先のフォルダを選択すると、ダイアログが表示されますので、CPU/周辺ハードウェア・クロック周波数 (fCLK) を設定し[OK]をクリックします。

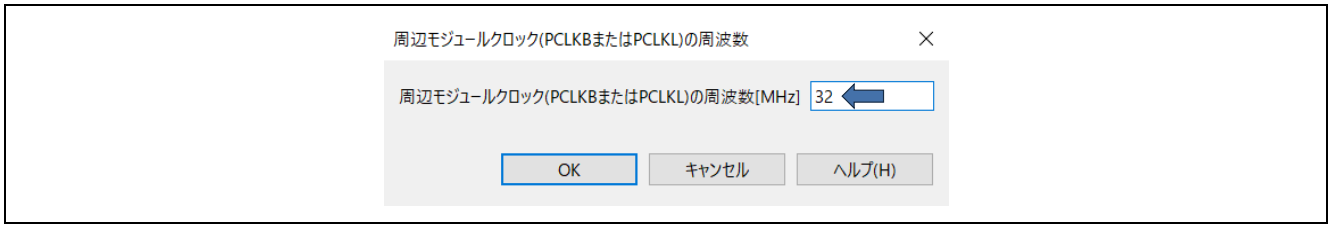


図 8-14　周辺モジュールクロックの周波数の設定

10. マイコンへ供給する電圧を選択します。表示された“マイコンへの供給電圧”ダイアログで、電圧値を設定して[OK]をクリックします。
使用するマイコンの電気的特性をご確認ください。
EVDD がある MCU の場合は、VDD を EVDD に読み替えて設定を行ってください。

図 8-15　マイコンへの供給電圧の設定

11. “QE for Capacitive Touch”ダイアログが表示されます。
ダイアログの内容は、QE 画面下部の“コンソール”パネルにも表示されます。

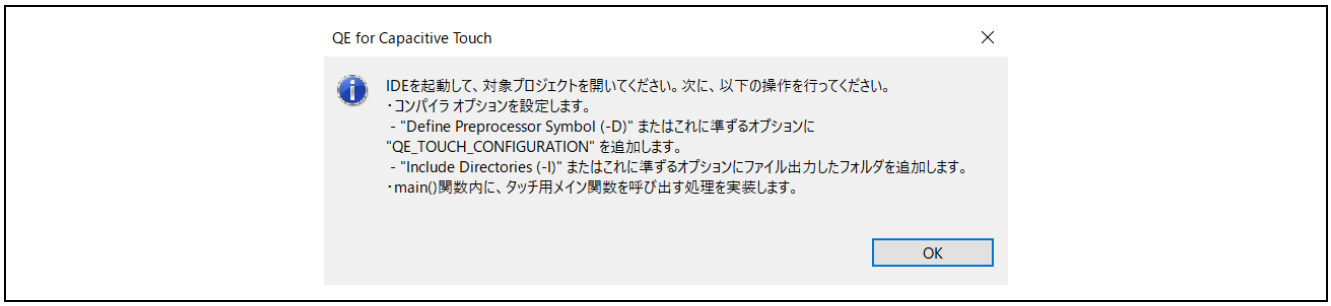


図 8-16　QE for Capacitive Touch ダイアログ

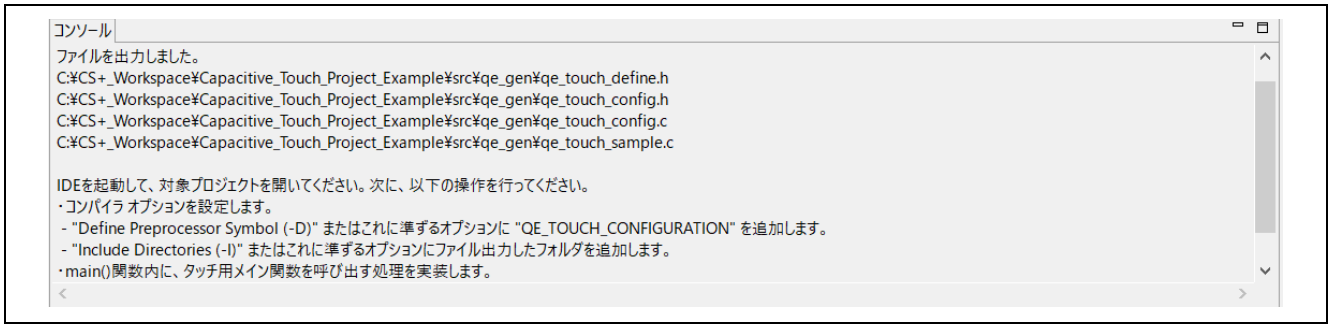


図 8-17　コンソールパネル

12. コンパイル・オプションを設定します。CS+を開き、“プロジェクト・ツリー” から “CC-RL(ビルド ツール)” を選択します。プロパティ” の “共通オプション” タブのうち、“よく使うオプション (コンパイル) ” の下の “定義マクロ” を選択し、右側に表示された [···] をクリックします。

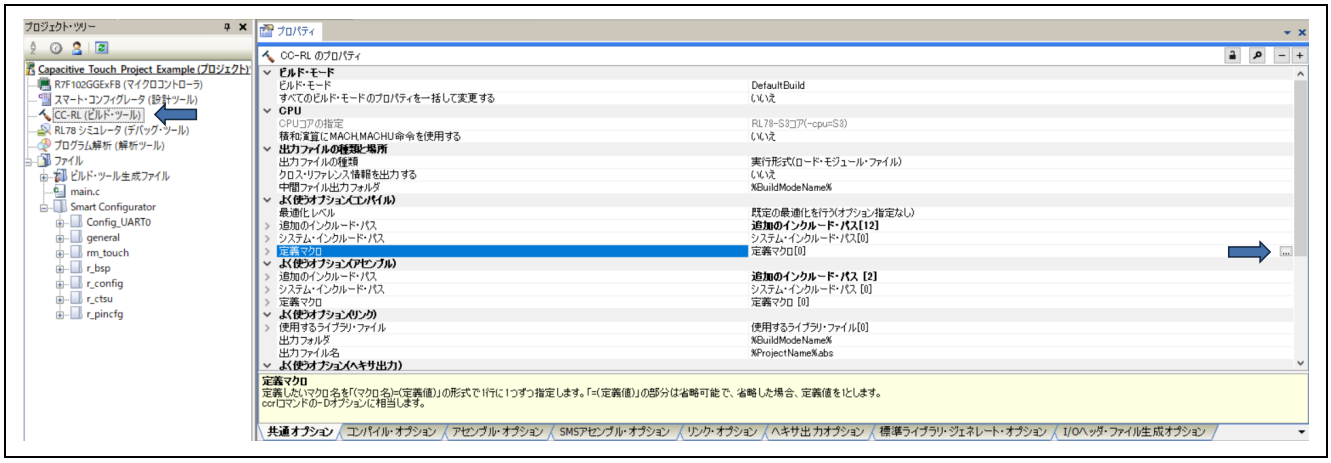


図 8-18　定義マクロの選択

13. 表示された “テキスト編集”ダイアログの“テキスト”フィールド内に“QE_TOUCH_CONFIGURATION” を追加し、[OK]をクリックします。

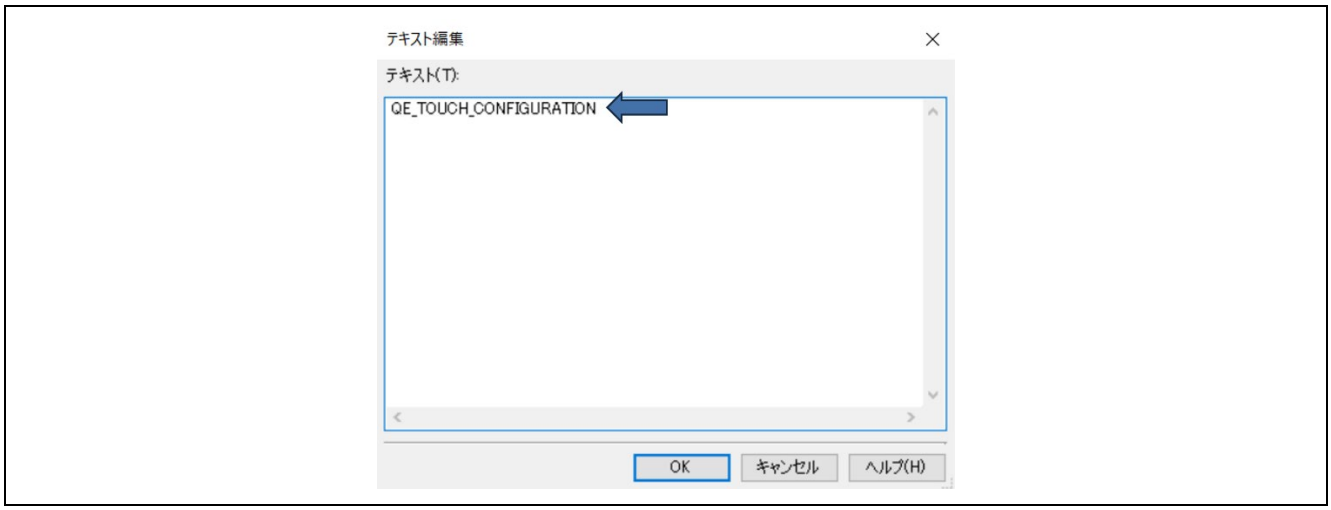


図 8-19　定義マクロの編集

14. サンプルプロジェクトへ”qe_gen”フォルダを追加します。CS+の“プロジェクト・ツリー”にエクスプローラから”qe_gen”フォルダをドラッグアンドドロップして追加します。

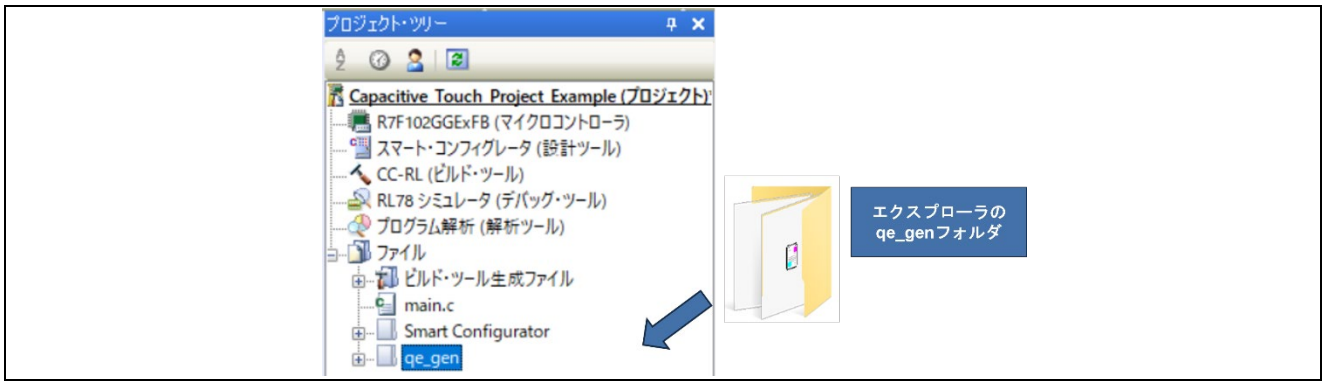


図 8-20　qe_gen フォルダの追加

15. “qe_gen”フォルダのインクルード・パスを追加します。プロパティ” の “共通オプション” タブのうち、”よく使うオプション (コンパイル) ” の下の ”追加のインクルードパス” を選択し、右側に表示された[…]をクリックします。
表示された “パス編集” ダイアログの “パス” フィールドに ”src¥qe_gen” が追加されていることを確認して、[OK]をクリックします。

図 8-21　コンパイラのインクルードパスの追加

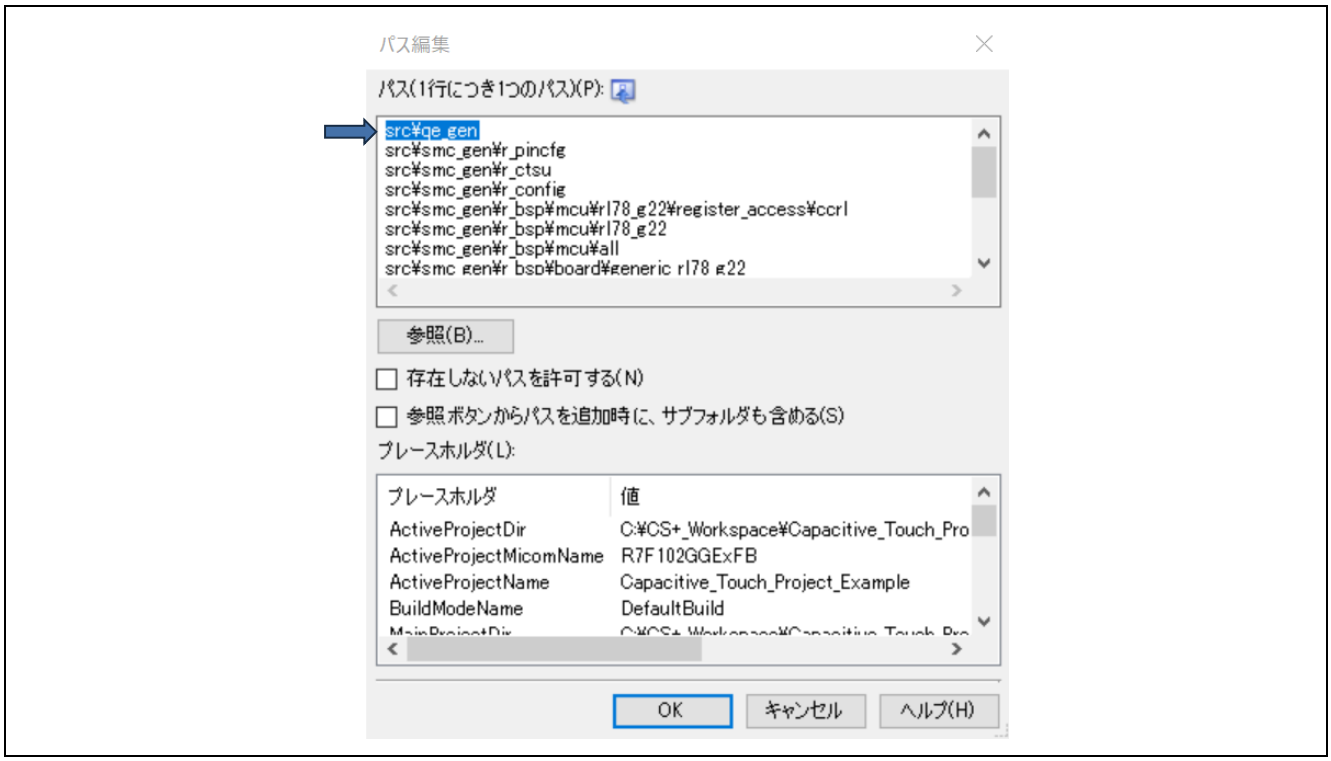


図 8-22　追加したインクルードパスの確認

16. C ソース・ファイルの言語を選択します。
”コンパイル・オプション” のタブにある ”ソース” をクリックし、”C ソース・ファイルの言語”をクリックします。右側に表示された をクリックし、“C99 (-lang=c99) ”を選択します。

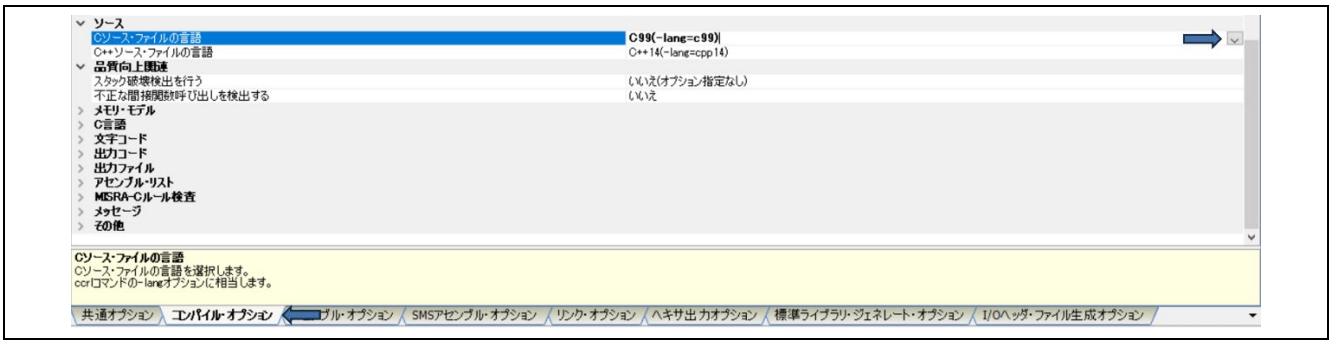


図 8-23　C 言語の規格の選択

17. オンチップ・デバック・オプション・バイトとユーザ・オプション・バイトを設定します。
”リンク・オプション”のタブにある”デバイス”をクリックし、以下を設定します。設定するオプション・バイト値は、使用するマイコンのユーザーズマニュアルをご確認ください。

- オンチップ・デバック・オプション・バイト制御値 : 84
- デバック・モニタ領域を設定する : はい (範囲指定) (-DEBUG_MONITOR=<アドレス範囲>)
- ユーザ・オプション・バイト値 : EFFCE8

図 8-24　オプション・バイトの設定

18. CC-RL 無償評価版の V1.12.00 以降のバージョンを使用してコンパイルする場合は、コンパイラの最適化レベルを” デバッグ優先 (-onothing)”を設定してビルドしてください。
”コンパイル・オプション”のタブにある”最適化”をダブルクリックし、”最適化レベル”に”デバッグ優先 (-onothing)”を選択します。

備考　本操作はチューニング時のみ必要となります。チューニング後は、任意の最適化設定で使用可能です。

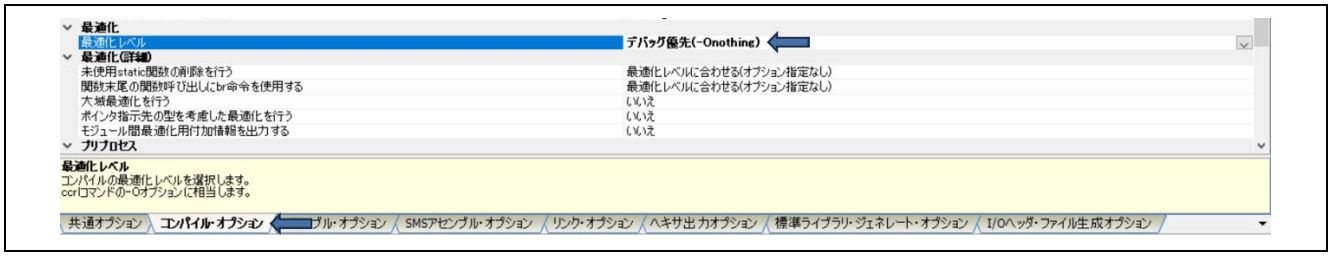


図 8-25　コンパイラの最適化レベルの設定

19. main()関数内に、タッチ用メイン関数を呼び出す処理を実装します。main()関数から qe_touch_main()関数をコールします。
”main.c”ファイルへ以下のコードを追加します。
    - extern void qe_touch_main(void);
    - qe_touch_main();

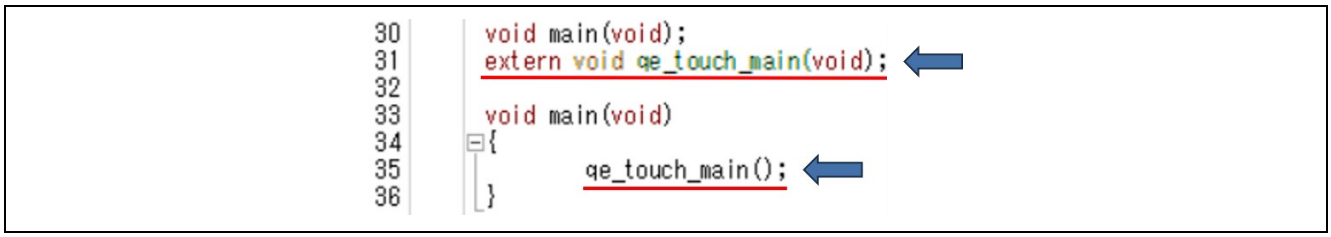


図 8-26　main.c

20. “Config_UART0_user.c”にシリアル通信用の関数を追加します。
    以下のコードをそれぞれ追加します。
    — extern void touch_uart_callback(uint16_t event);
    — touch_uart_callback(0);
    — touch_uart_callback(1);

```
52      /* Start user code for global. Do not edit comment generated here */
53      extern void touch_uart_callback(uint16_t event);
54      /* End user code. Do not edit comment generated here */


74      static void r_Config_UART0_callback_sendend(void)
75      {
76          /* Start user code for r_Config_UART0_callback_sendend. Do not edit comment generated here */
77          touch_uart_callback(0);
78          /* End user code. Do not edit comment generated here */
79      }


87      static void r_Config_UART0_callback_receiveend(void)
88      {
89          /* Start user code for r_Config_UART0_callback_receiveend. Do not edit comment generated here */
90          touch_uart_callback(1);
91          /* End user code. Do not edit comment generated here */
92      }
```

図 8-27　Config_UART0_user.c

21. CS+でプロジェクトをビルドします。CS+のメニュー下にある  アイコンをクリックし、ビルドします。エラーまたはワーニングなしで終了することを確認します。

ビルド時に下図のワーニング(W0511187)が出た場合は、本書 40 ページの図 8-25 を参考にコンパイラ最適化設定を「デバッグ優先 (-onothing) 」に変更して、再ビルドしてください。

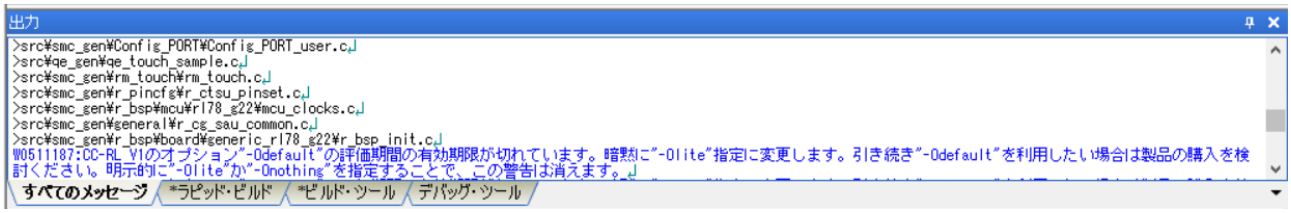


図 8-28　ビルド時のワーニング (W0511187)

8.4　調整

ワークフロー図の“調整”に従い、設定します。

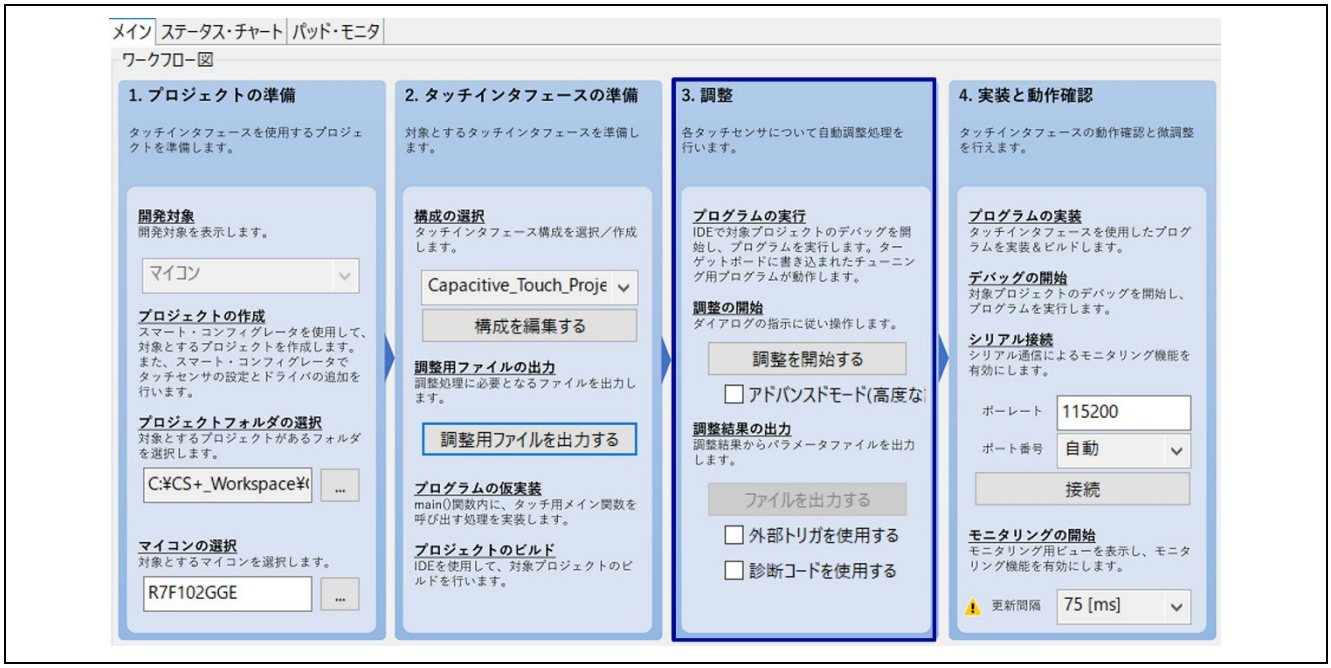


図 8-29　ワークフロー図 (調整)

1. 使用するデバック・ツールを選択します。CS+の“プロジェクト・ツリー”で“デバッグ・ツール”を右クリックし、”使用するデバッグ・ツール”の”RL78 COM Port(C)”を選択します。

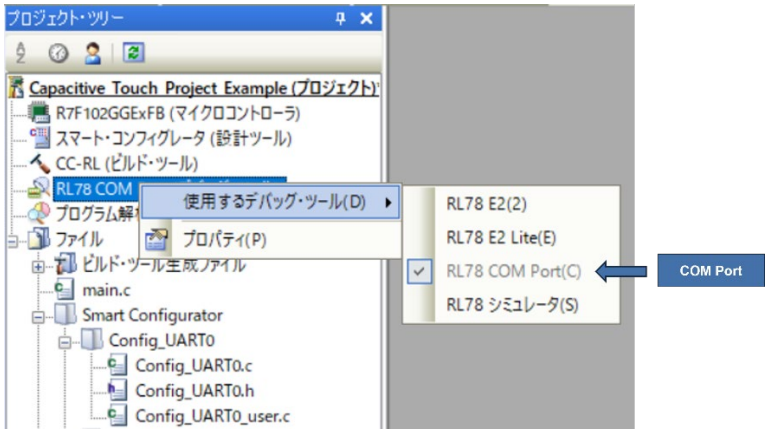


図 8-30　デバッグ・ツールの選択

2. “デバッグ・ツール”のプロパティで”通信ポート”を設定します。
本アプリケーション例では、COM24 を使用します。

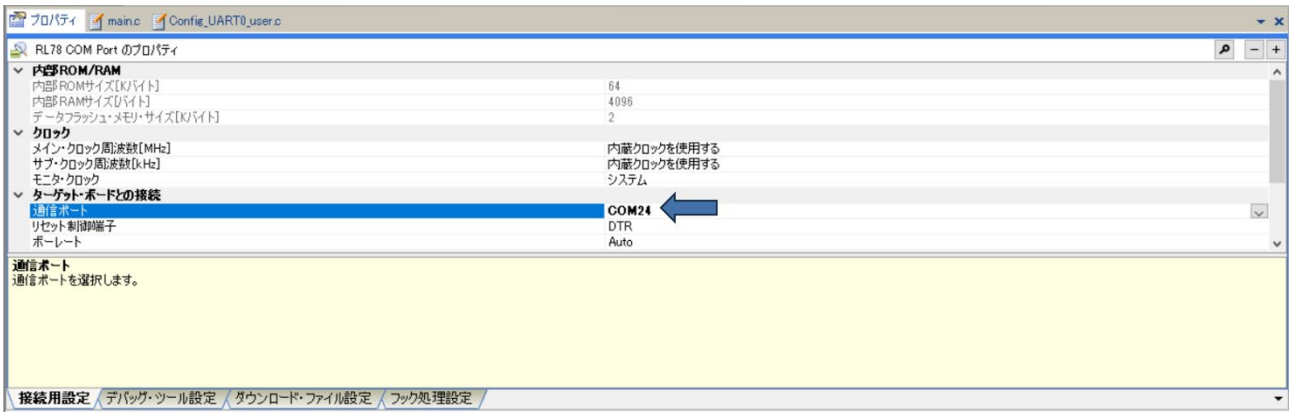


図 8-31　デバッグ・ツールのプロパティ

通信ポートの COM 番号は、デバイス マネージャーで確認できます。

図 8-32　デバイス マネージャー

3. COM Port デバック回路を有効にします。ターゲットボードの QE シリアル接続切り替えジャンパ (J16) がショートされていることを確認してください。

4. ビルドとプログラムの書き込みを行います。PC とターゲットボードが USB ケーブルで接続されていることを確認し、CS+の  アイコンをクリックします。プログラム書き込み後のダウンロードが完了したら  アイコンをクリックしてプログラムを停止させ、続いて  アイコンをクリックして切断します。

5. QE のシリアル接続機能を実行します。デバック・ツールから切断後、PC とターゲットボードを接続している USB ケーブルを外し、QE シリアル接続切り替えジャンパ (J16) をオープンにします。
この後に QE と接続するため、PC とターゲットボードを USB ケーブルで再度接続します。この時にターゲットボードは、書き込んだプログラムが動作し、QE との接続待機状態となります。

QE シリアル接続切り替えジャンパ (J16) は、ターゲットボードのユーザーズマニュアルをご確認ください。また、USB ケーブルはデータ転送対応のケーブルをご使用ください。

6. QE で“シリアル接続”の”ボーレート”を 7.5.3 節で設定した値に設定します。

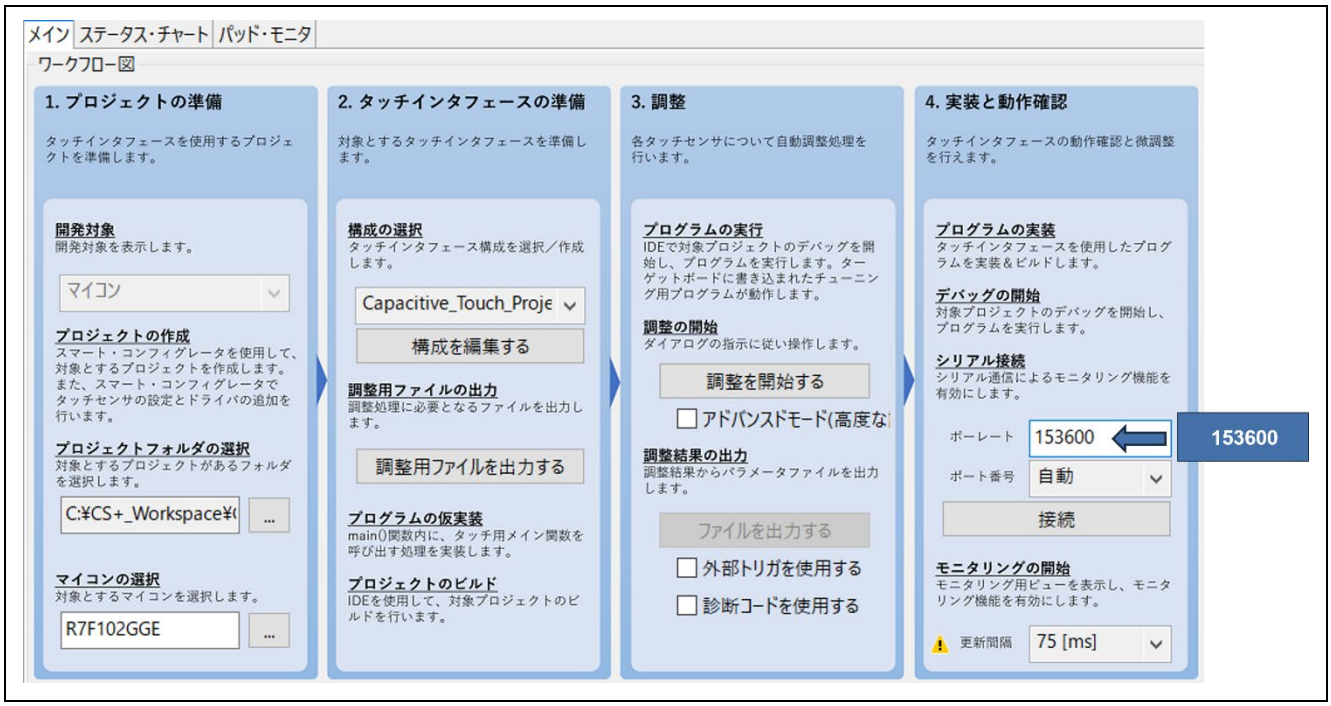


図 8-33　ボーレートの設定

7. [調整を開始する]をクリックし、自動チューニングを開始します。

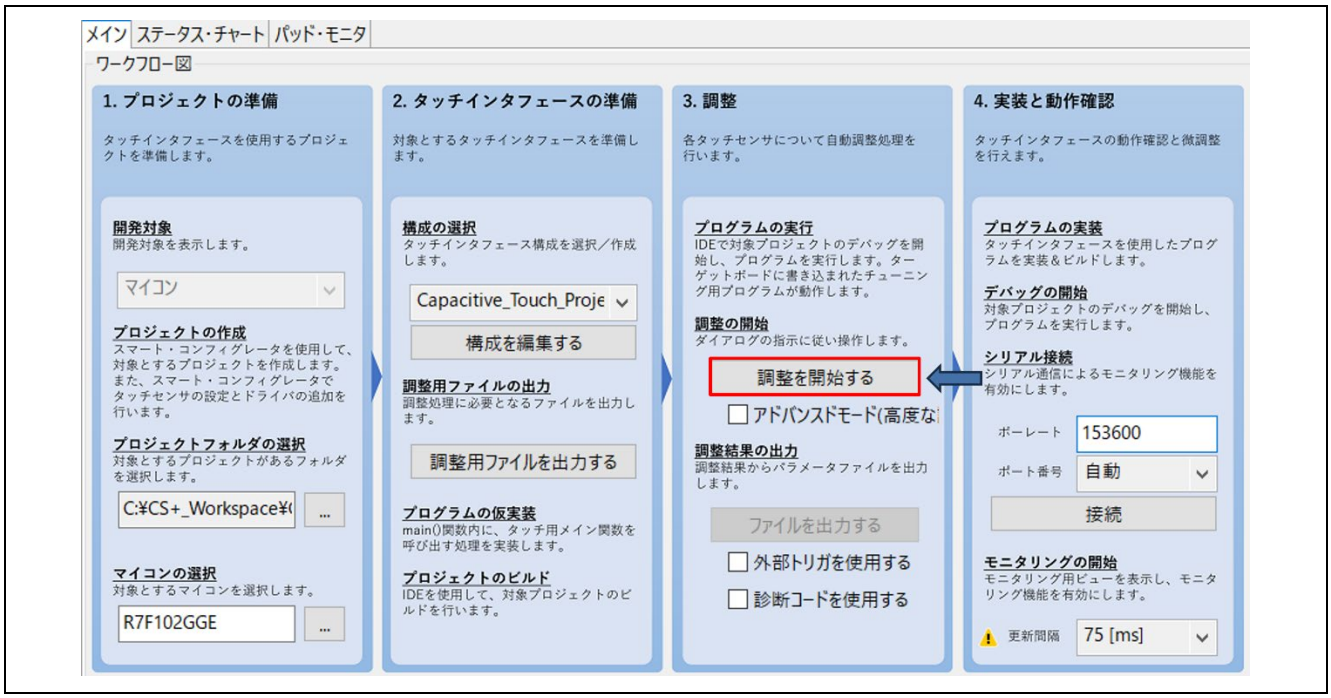


図 8-34　自動チューニング

8. 表示されたダイアログで”ボーレート”を設定し、[接続]をクリックします。

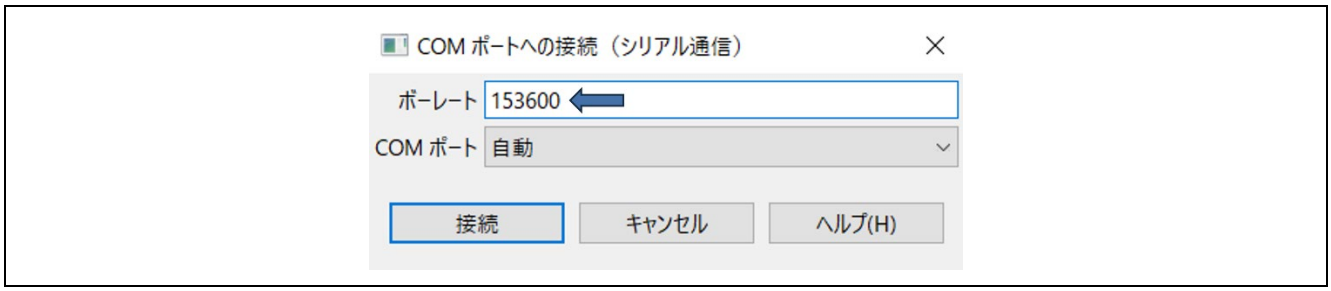


図 8-35　ボーレートの設定

9. 続いて表示されたダイアログで CPU/周辺ハードウェア・クロック周波数を設定し、[OK]をクリックします。

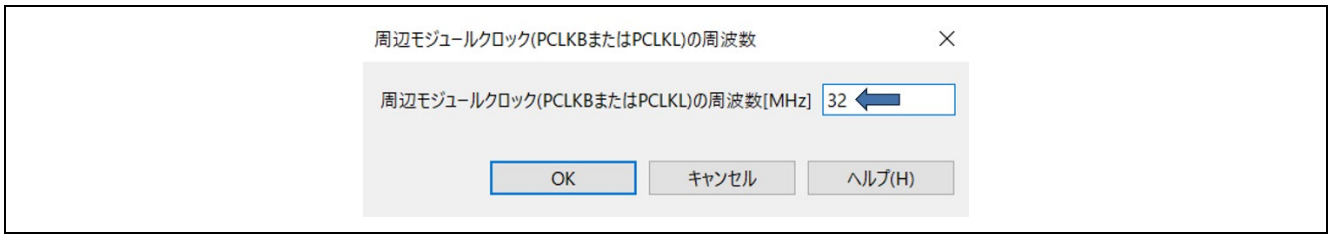


図 8-36　周辺モジュールクロックの周波数の設定

10. 自動チューニングが開始されます。チューニングプロセスをガイドする[自動調整処理中]ダイアログを適宜確認し、ダイアログ中の指示に従い操作を進めていきます。

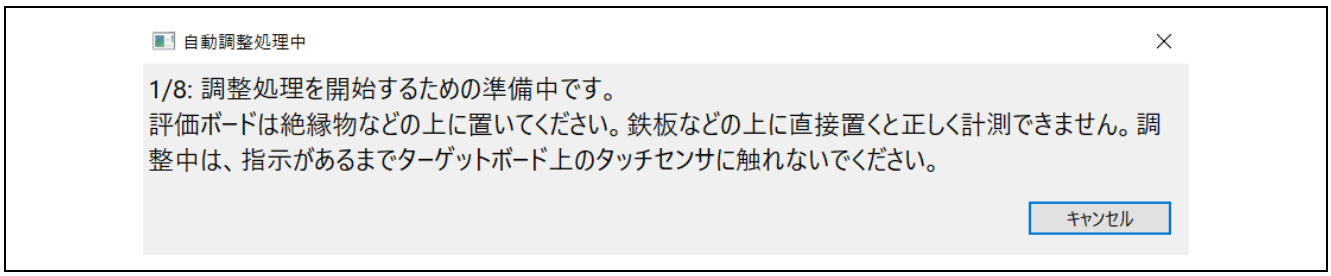


図 8-37　自動調整処理中ダイアログ

いくつかのプロセスを経て、以下のようなダイアログが表示されます。
ここでは、タッチ感度を計測します。ダイアログで表示されている Button01 を通常の圧力でタッチします。タッチセンサに触れているとき、バーグラフは右に増加し、数値で示すタッチカウント値が大きくなります。
タッチセンサに触れたまま、PC キーボードのいずれかのキーを押して計測を確定します。

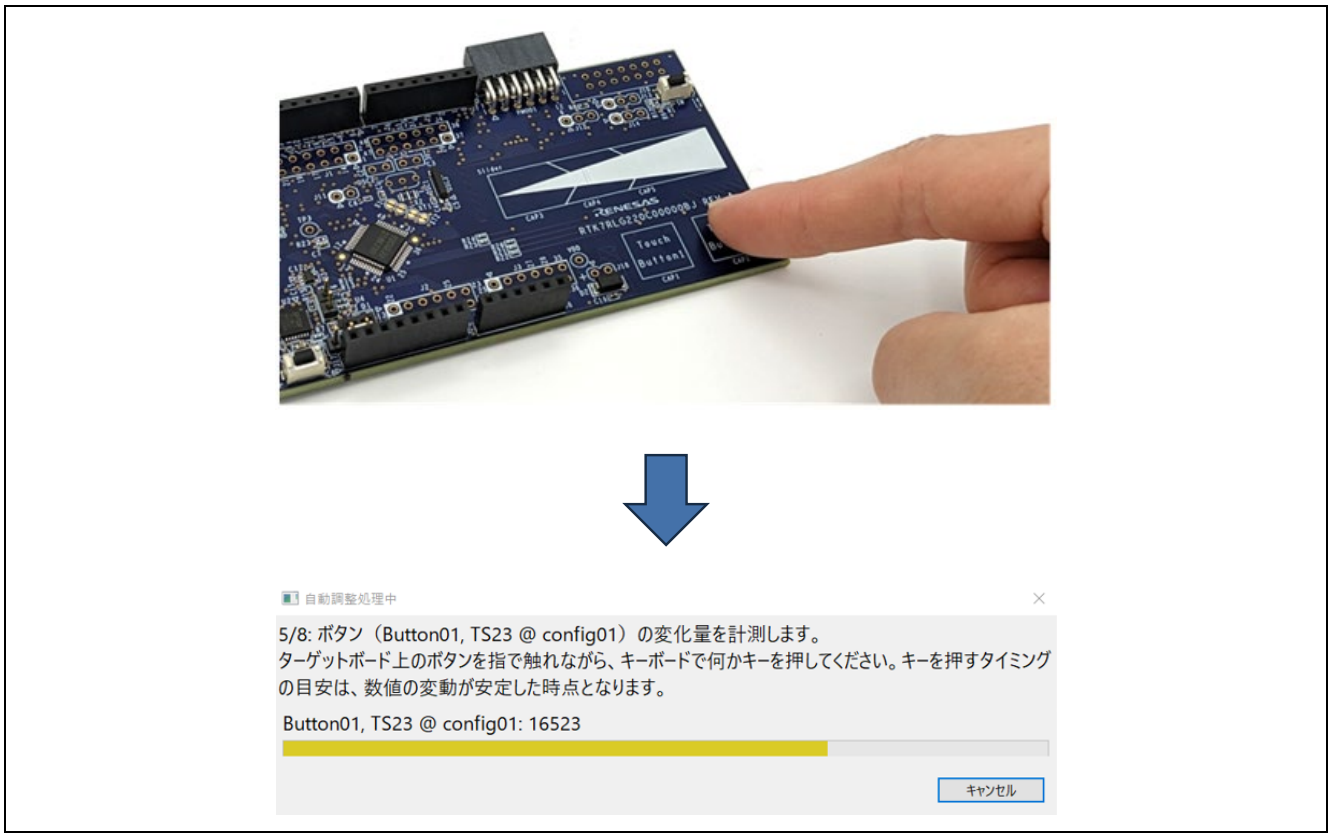


図 8-38　タッチ感度の計測 (ボタン)

11. もう一方のタッチセンサについても同様に計測します。

12. スライダのタッチセンサのタッチ感度を計測します。ターゲットボード上にあるスライダを通常の圧力で上下または左右に 3～4 回なぞったあと、PC キーボードのいずれかのキーを押して計測を確定します。

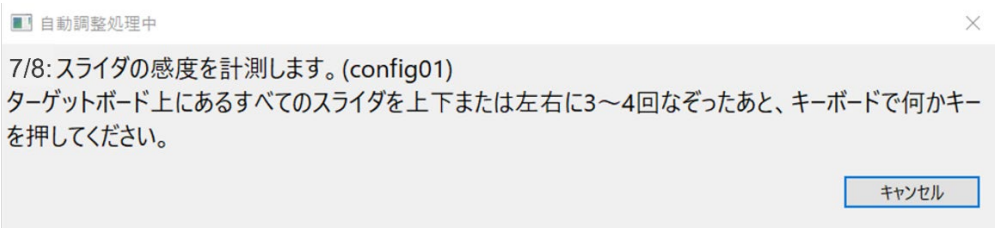


図 8-39　タッチ感度の計測 (スライダ)

13. チューニングが完了すると、以下のようなダイアログが表示され、しきい値を確認できます。このしきい値はミドルウェアでタッチのイベント判定で使用されます。
[調整処理の継続]をクリックします。これでチューニングは終了です。

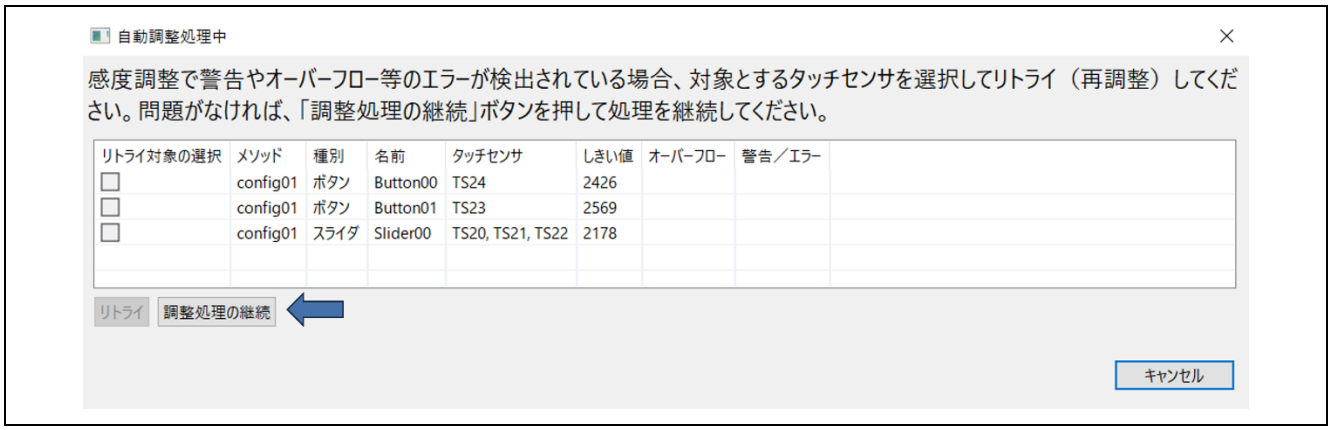


図 8-40　タッチセンサのしきい値

14. [ファイルを出力する]をクリックし、調整結果が反映されたパラメータファイルを出力します。ファイルの出力先フォルダは 8.3 節で新規作成した”qe_gen”を選択し、ファイルを上書きします。
出力されるファイル群は、8.3 節の[調整用ファイルを出力する]で出力された以下のファイル群と同じファイル名です。

|- qe_touch_config.c　← 出力ファイル

|- qe_touch_config.h　← 出力ファイル

|- qe_touch_define.h　← 出力ファイル

|- qe_touch_sample.c　← 出力ファイル

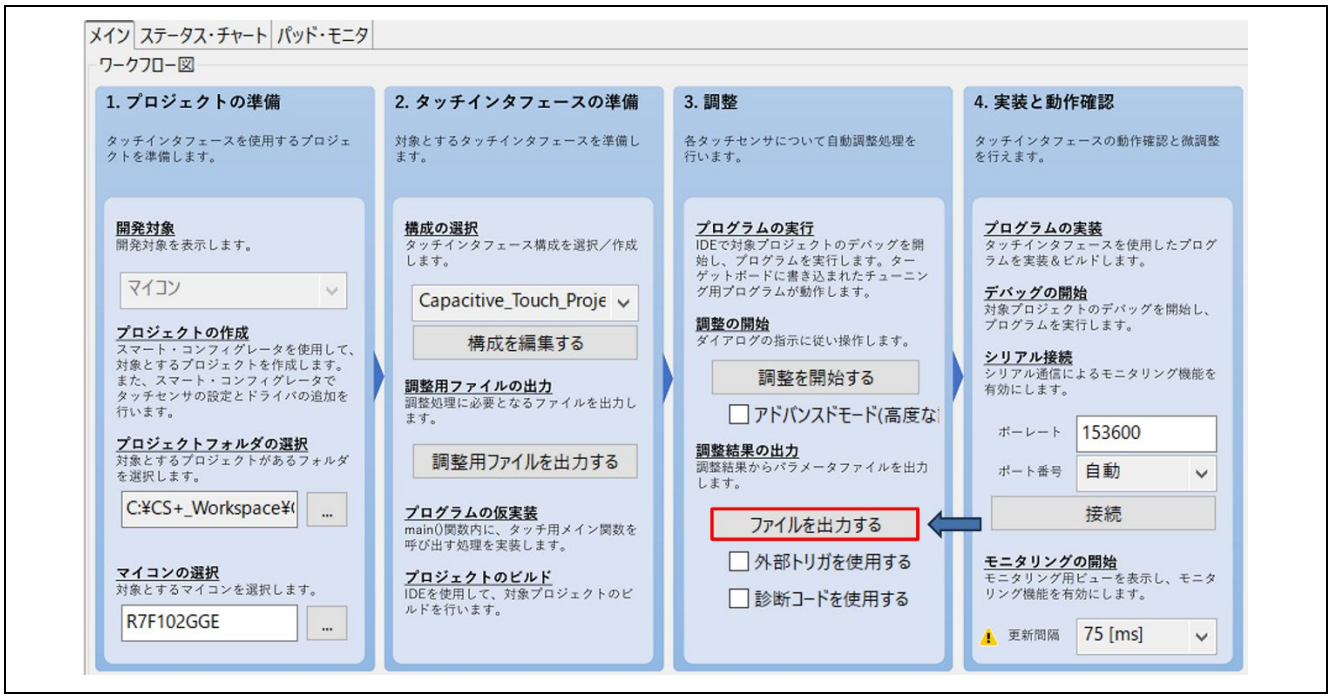


図 8-41　パラメータファイルの出力

8.5　実装と動作確認

8.5.1　モニタリング

ワークフロー図の“実装と動作確認”に従い、モニタリングを実施します。

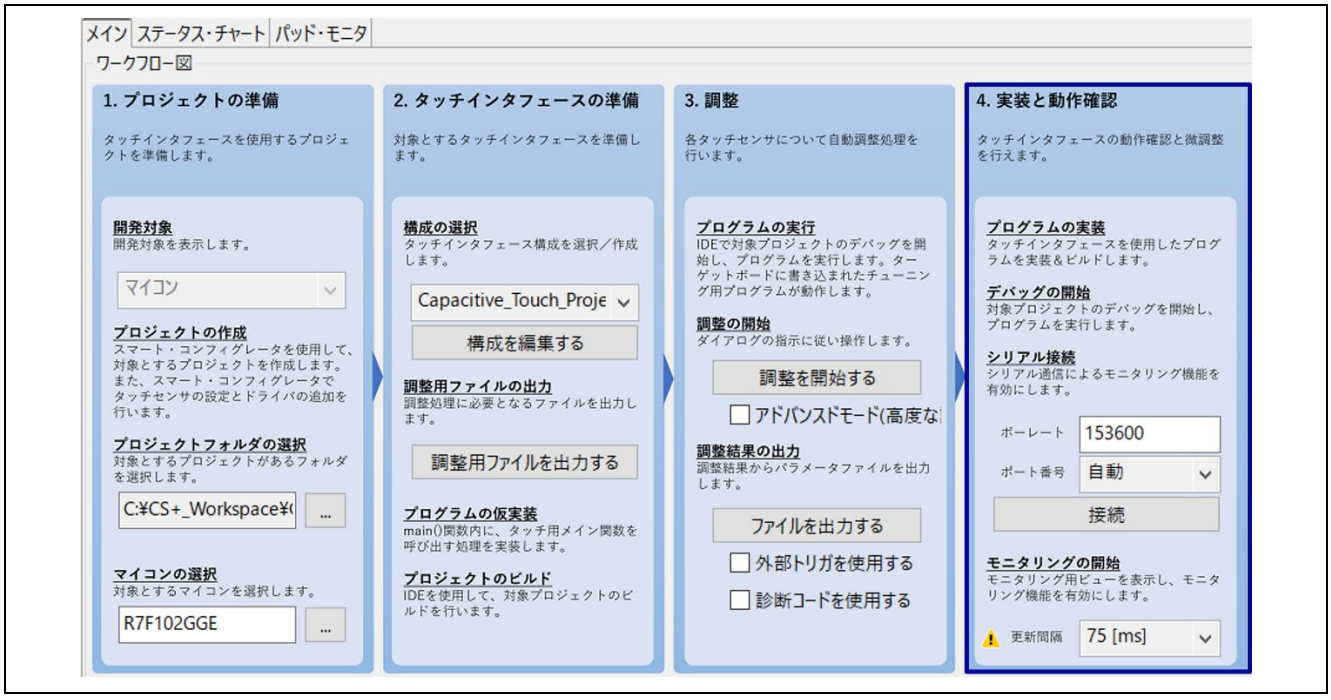


図 8-42　ワークフロー図 (実装と動作確認)

1. COM Port デバック回路を有効にします。PC とターゲットボードを接続している USB ケーブルを外し、QE シリアル接続切り替えジャンパ (J16) をショートします。次に CS+と接続するため、PC とターゲットボードを USB ケーブルで再度接続します。

2. ビルドとプログラムの書き込みを行います。CS+の  アイコンをクリックし、ビルドとプログラムの書き込みを行います。プログラム書き込み後のダウンロードが完了したら、  アイコンをクリックしてプログラムを停止させ、続いて  アイコンをクリックして切断します。

3. QE のシリアル接続機能を実行します。デバック・ツールから切断後、PC とターゲットボードを接続している USB ケーブルを外し、QE シリアル接続切り替えジャンパ (J16) をオープンにします。
この後に QE と接続するため、PC とターゲットボードを USB ケーブルで再度接続します。この時にターゲットボードは、書き込んだプログラムが動作し、QE との接続待機状態となります。

3. [接続]をクリックして、ターゲットボードとシリアル接続します。図 8-43 の赤枠内の表示が[接続]から[切断]に切り替わります。

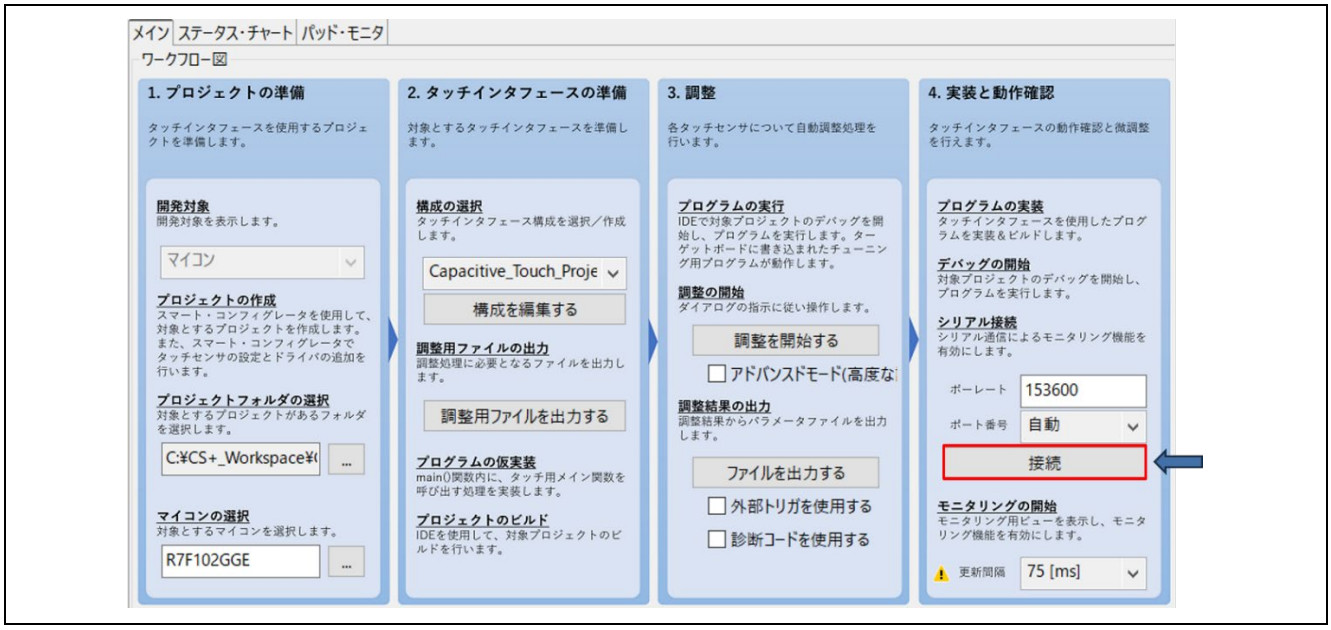


図 8-43　シリアル接続

4. QE 画面の左上にある“ボード・モニタ”パネルの[モニタリングを有効にする]をクリックします。”モニタリング機能: 無効”が“モニタリング機能: 有効”に切り替わります。

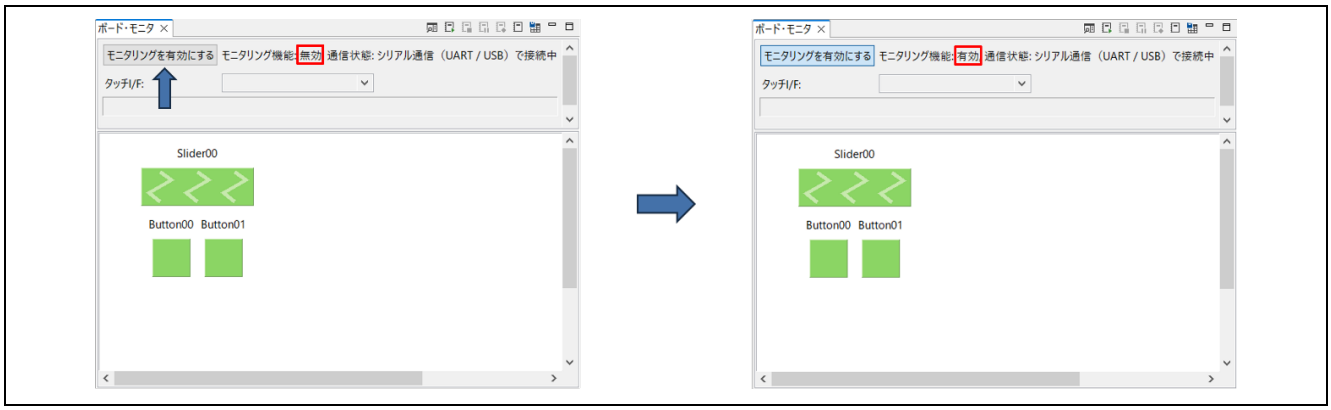


図 8-44　モニタリングの有効化

タッチセンサに触れると、その状態が指のアイコンで表されます。

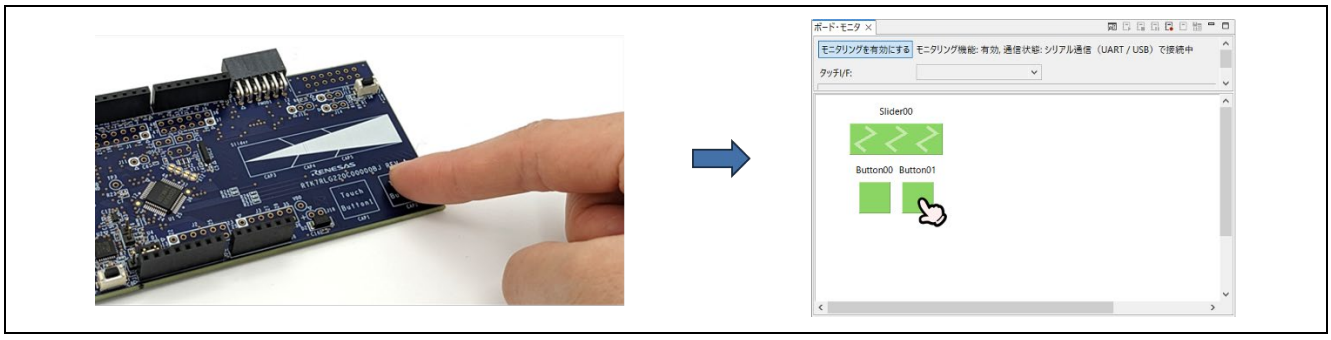


図 8-45　タッチセンサに触れた状態の表示

6. タッチカウント値をステータス・チャートへグラフィカルに表示します。
   A. [ステータス・チャート]タブをクリックします。

   B. 表示されたステータス・チャート画面の“タッチ I/F” の をクリックし、タッチインタフェースを選択します。

グラフには実行中のタッチカウント値が表示されます。選択したタッチセンサに触れると、タッチカウント値がグラフ上で変化することを確認できます。
緑のラインは、しきい値を表し、rm_touch ミドルウェアはボタンが操作/タッチされているかどうかを判断するために使用されます。
グラフ下部の赤い矩形は、タッチカウント値がしきい値を超えてタッチが検出されたことを表示しています。

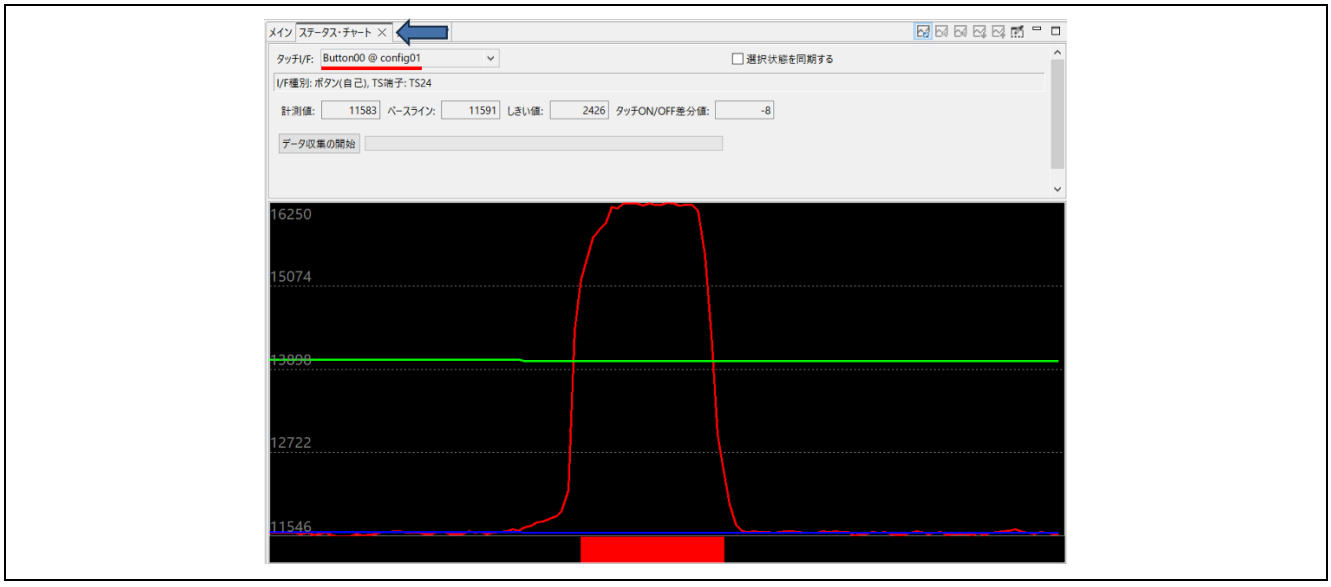


図 8-46　タッチカウント値のグラフ表示 (ボタン)

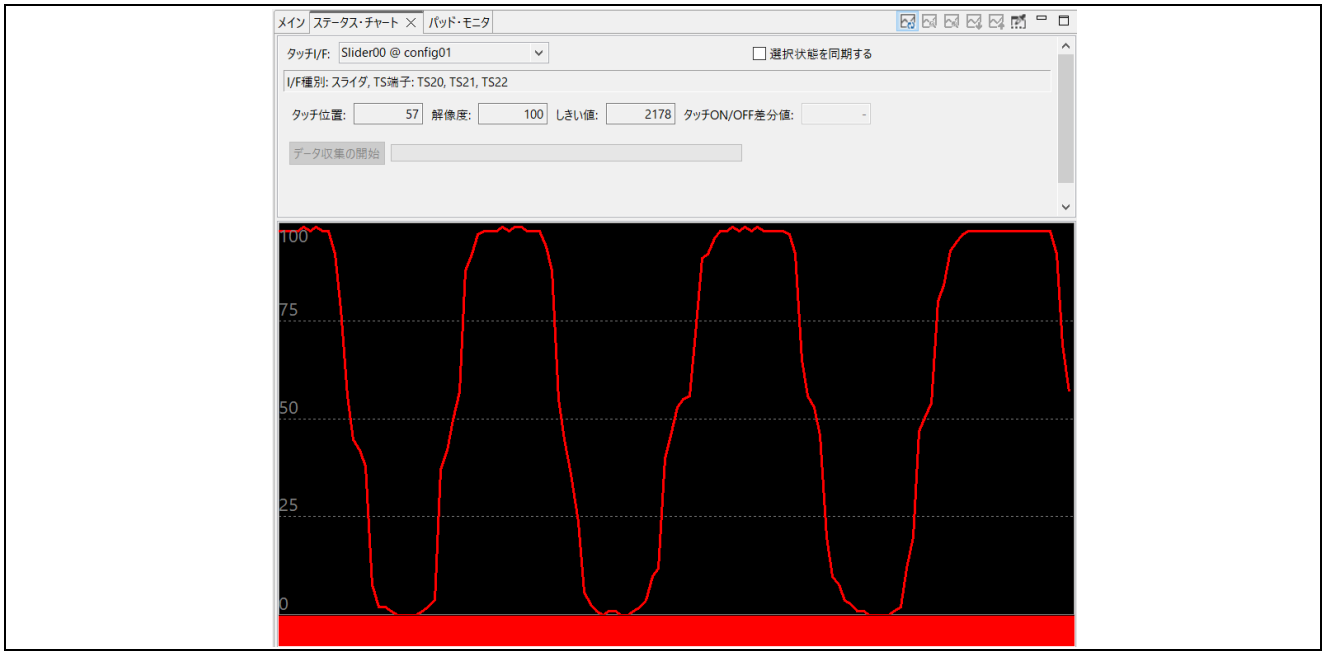


図 8-47　タッチカウント値のグラフ表示 (スライダ)

7. 必要に応じて、SNR 値を測定します。

A. [ステータス・チャート]タブ上の“データ収集の開始”をクリックします。

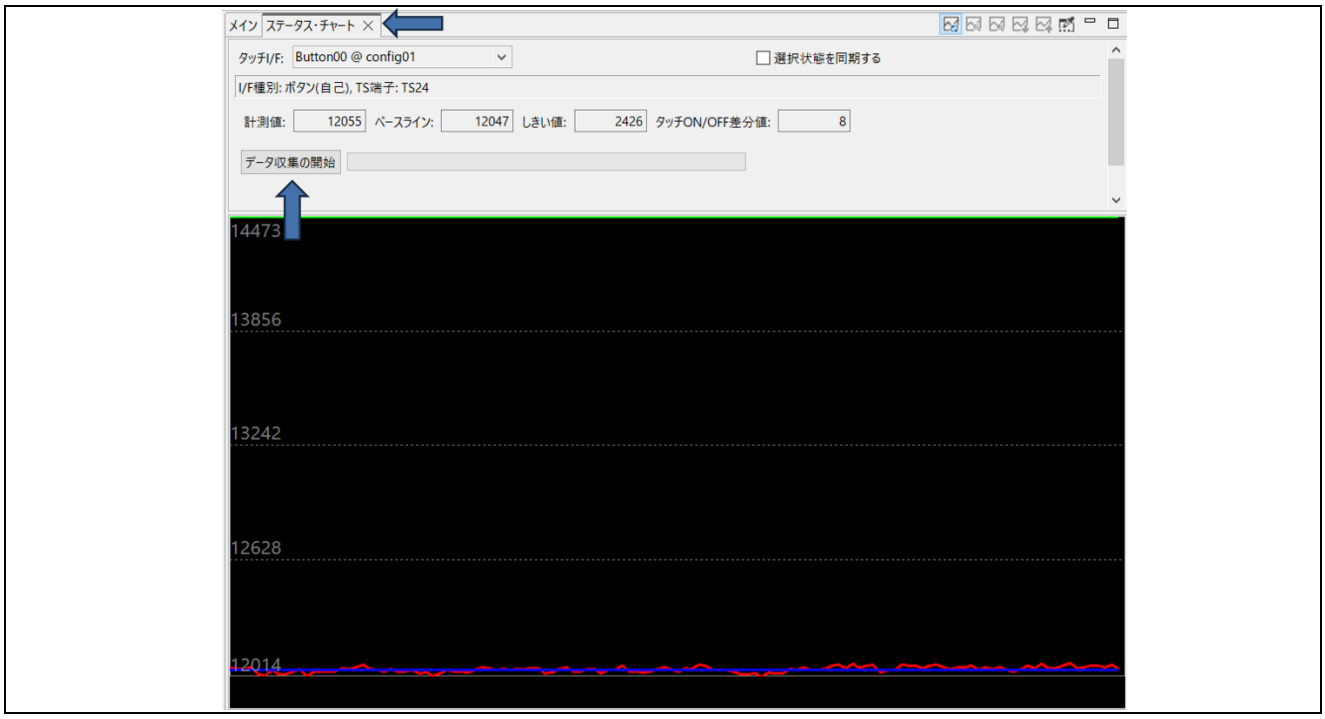


図 8-48　タッチオフ状態のデータ収集

B. データ収集設定を行い、“データ収集の開始”をクリックします。
タッチオフ状態を収集している間は電極に触れないでください。また、緑色のバーはデータの収集率を表しています。緑色のバーが右端まで達すると。データの収集率は 100%となりタッチオフ状態の収集は完了します。

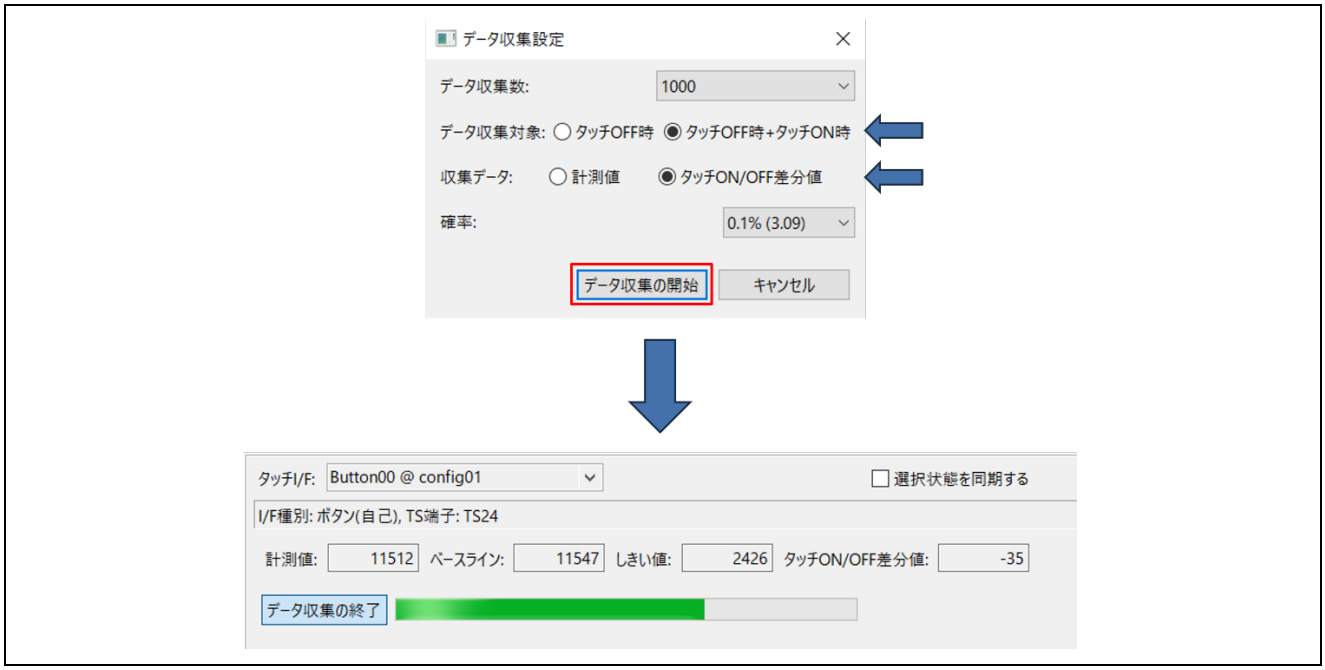


図 8-49　データ収集の開始

C. 同様の手順でタッチ ON 時のデータを収集します。指が電極に触れていることを確認してから、”データ収集の開始”をクリックします。緑色のバーが右端まで達するとタッチオン状態の収集は完了します。

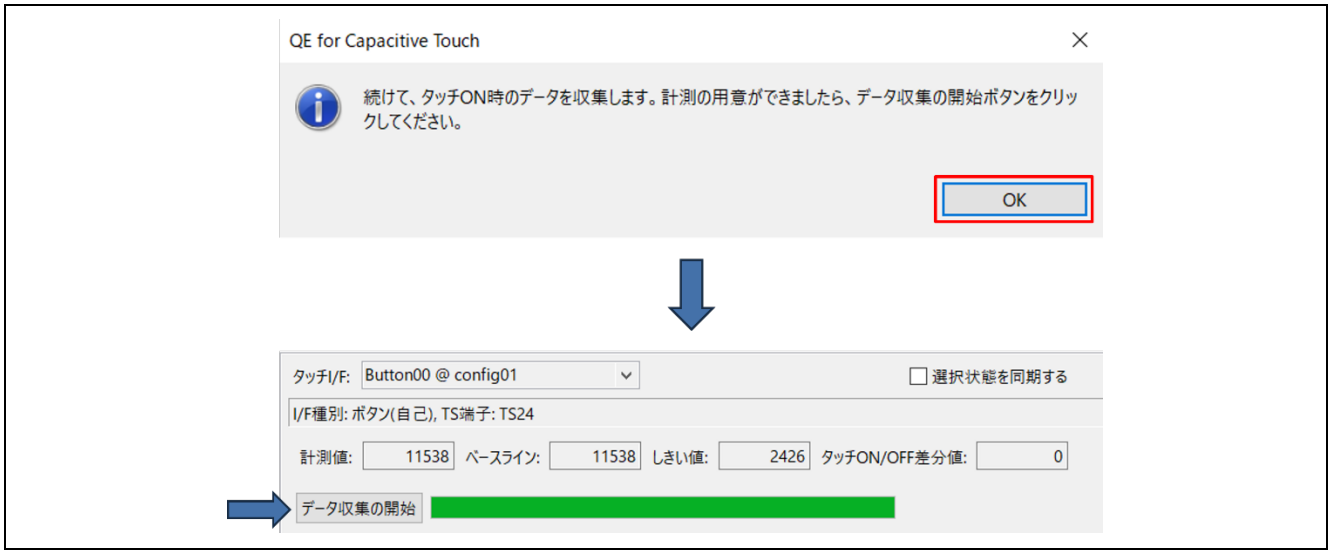


図 8-50　タッチ ON 時のデータ収集の開始

D. データ収集が完了すると SNR 値が表示されます。

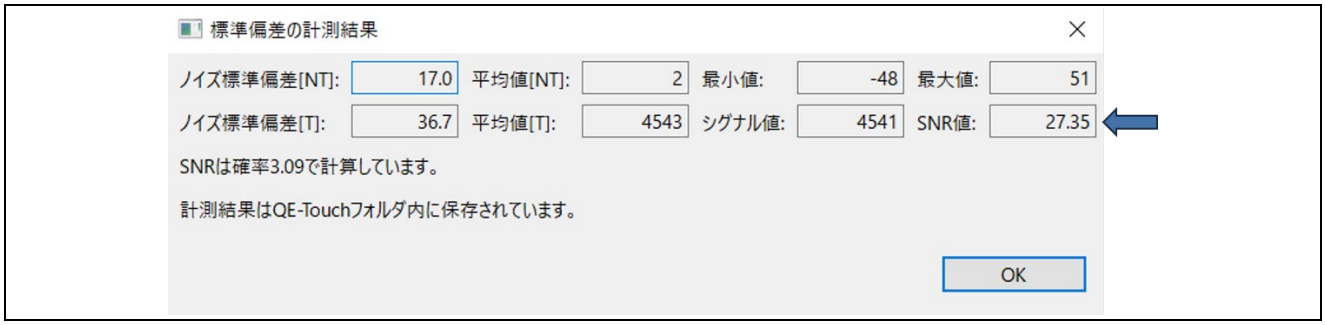


図 8-51　SNR 値

8. 複数のタッチセンサのタッチカウント値をマルチ・ステータス・チャートへグラフィカルに表示します。
   QE 画面の左下にある[マルチ・ステータス・チャート]タブにて、表示するタッチセンサを選択します。

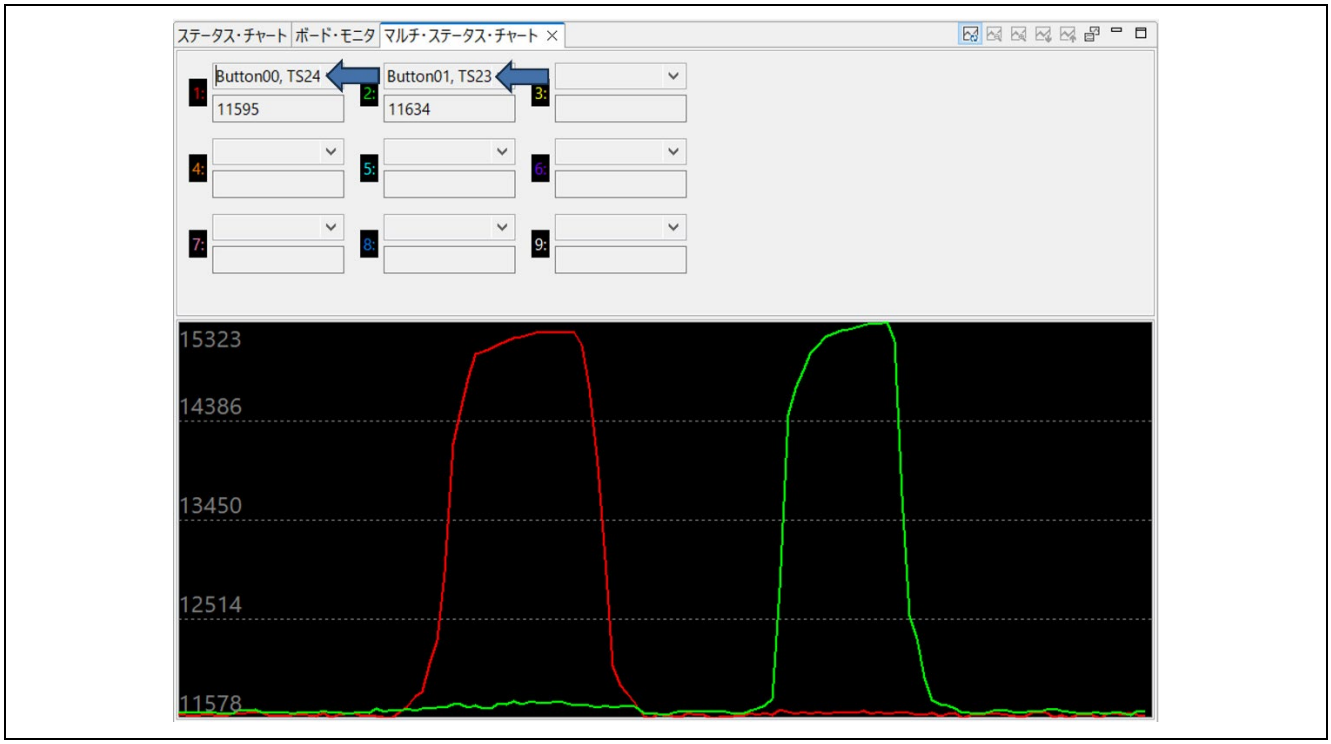


図 8-52　マルチ・ステータス・チャート

9. 必要に応じて、パラメータを手動で調整します。
   QE 画面の右側にある“パラメーター覧”パネルでパラメータを調整します。

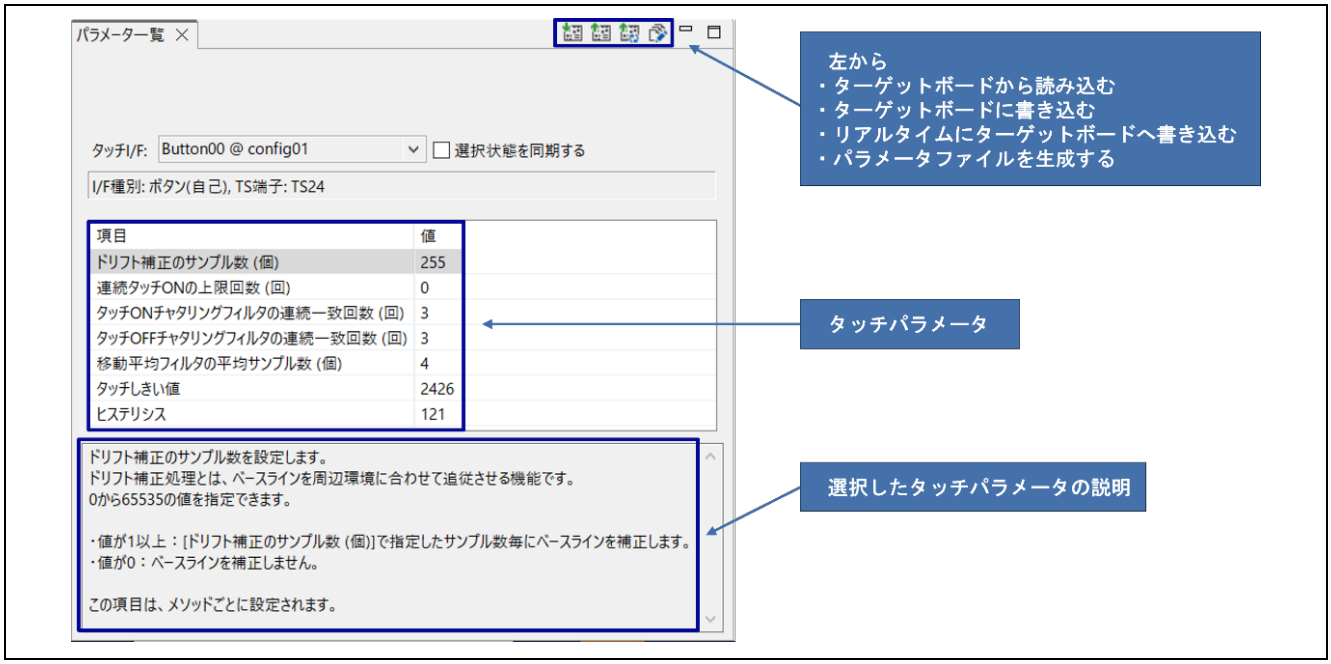


図 8-53　パラメータの調整

10.“モニタリング機能: 有効”の状態で [モニタリングを有効にする] をクリックして、モニタリングを終了します。

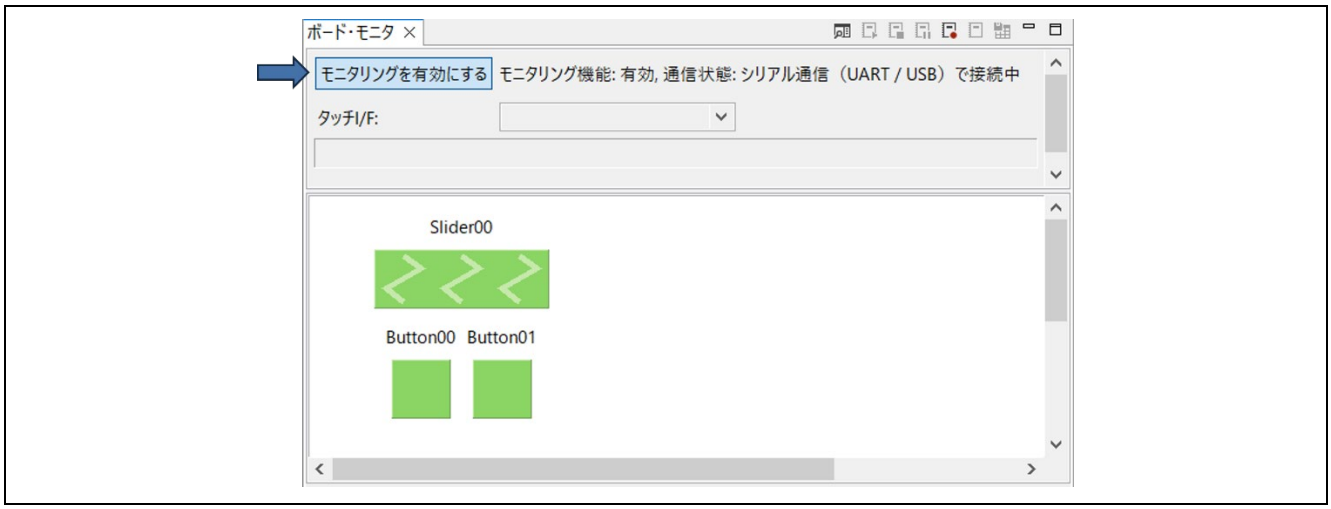


図 8-54　モニタリングを終了する

11.[切断] をクリックし、シリアル接続を切断します。

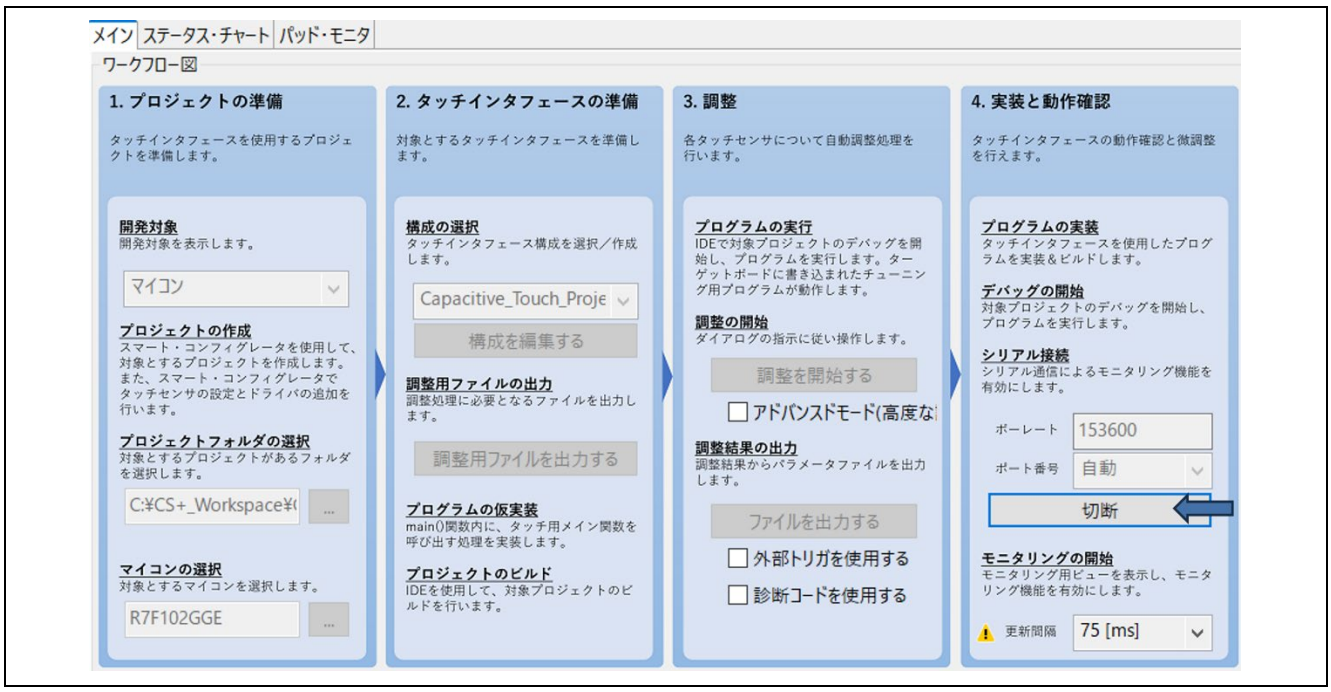


図 8-55　シリアル接続を切断

8.6　フローチャート（ソフトウェアタイマ）

図 8-56 にソフトウェアタイマを使用したタッチ計測制御処理のフローチャートを示します。

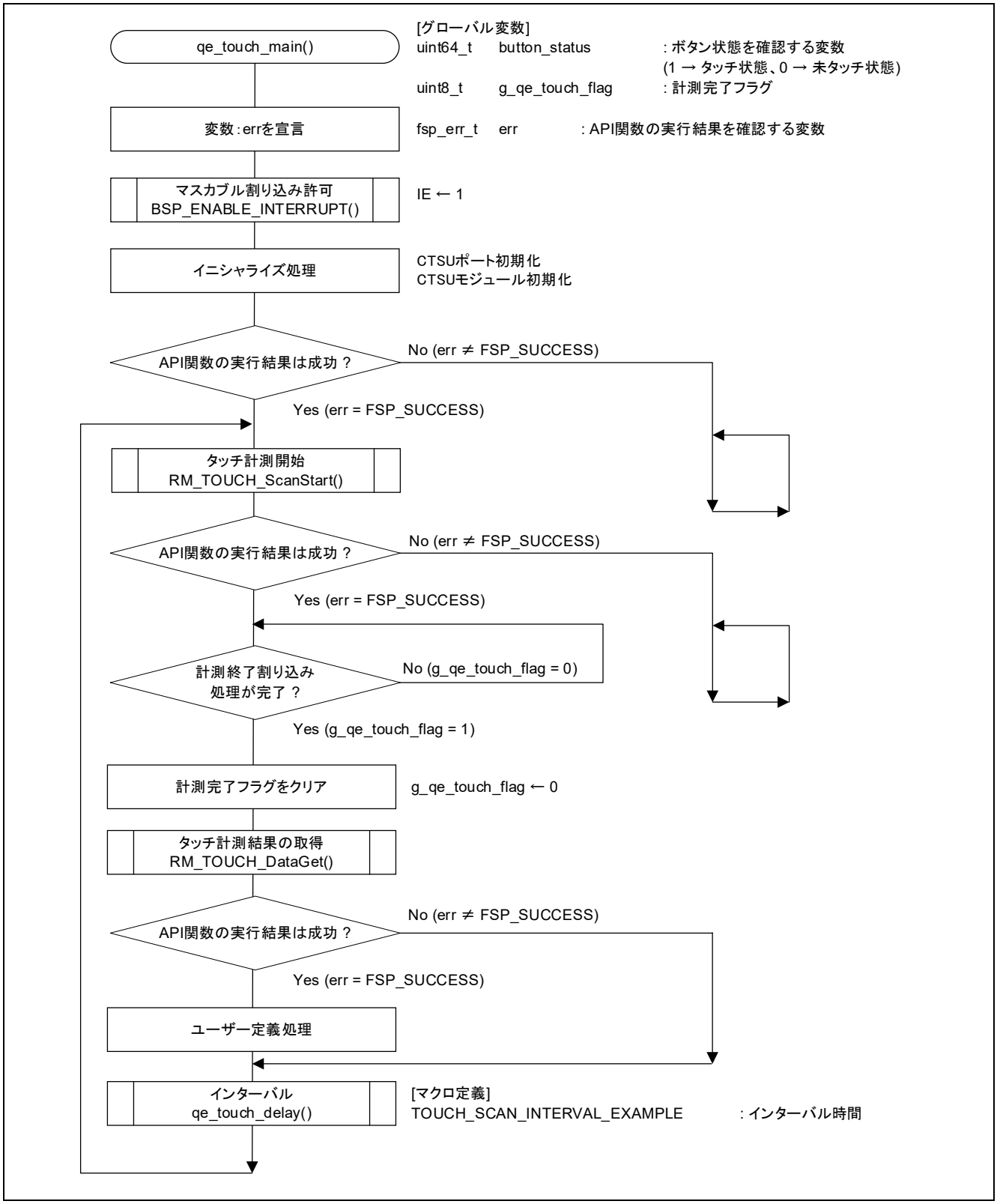


図 8-56　ソフトウェアタイマを使用したタッチ計測制御処理

9.　応用例

9.1　ハードウェアタイマでのタッチ計測

本章では、ハードウェアタイマを使用したタッチ計測周期の実装例を説明します。本アプリケーション例では、32 ビット・インターバル・タイマの 8 ビット・カウンタ・モードによるインターバル・タイマ機能を使用します。また、動作確認のため、タッチセンサ (ボタン) でのタッチ判定結果に応じてターゲットボード上の LED を点灯あるいは消灯させます。本アプリケーション例では、タッチセンサ① (TS_B1) に指が触れ、タッチ判定結果が ON になると、LED1 が点灯します。

アプリケーションノート本編の「7. スマート・コンフィグレータの設定」の内容に加えて、以下の設定を行ってください。

備考　32 ビット・インターバル・タイマの代わりに、タイマ・アレイ・ユニットや 12 ビット・インターバル・タイマを使用することも可能です。

9.1.1　スマート・コンフィグレータの設定 (ハードウェアタイマ)

1. スマート・コンフィグレータの [クロック] タブを選択し、インターバル・タイマに使用するクロックを設定します。
   本アプリケーション例では低速周辺クロック (fSXP) を使用します。また、XT1 発振回路のチェックを外します。

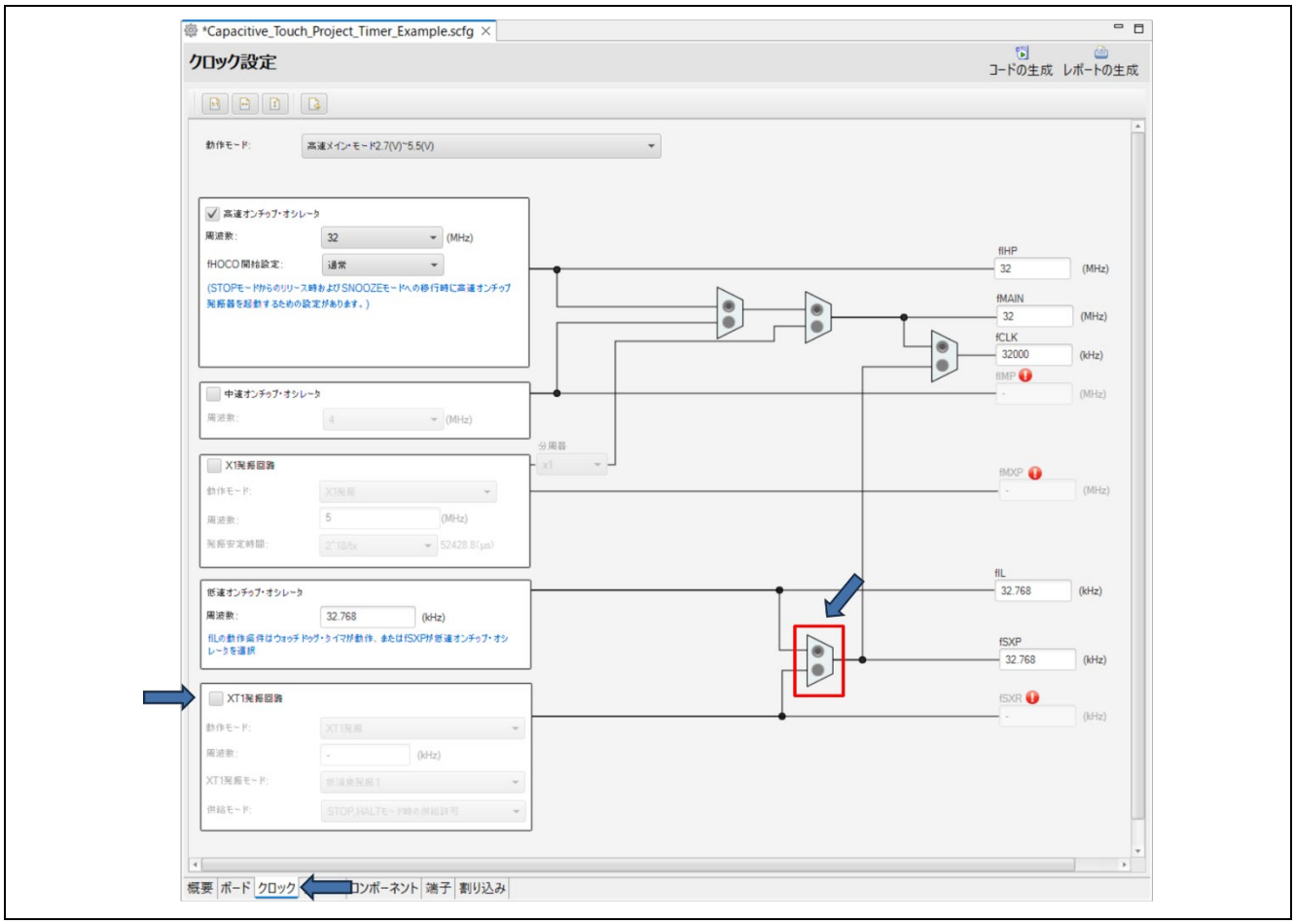


図 9-1　クロックの設定

2. ハードウェアタイマを使用したタッチ計測および LED の制御に必要な周辺機能を追加します。
   [コンポーネント] タブを選択し、  クリックして”コンポーネントの追加”ダイアログを開きます。“インターバル・タイマ”モジュールと“ポート“モジュールを選択し、[次へ] をクリックします。
   続いて、選択したコンポーネントに対してリソースを設定します。本アプリケーション例では以下の設定で使用します。

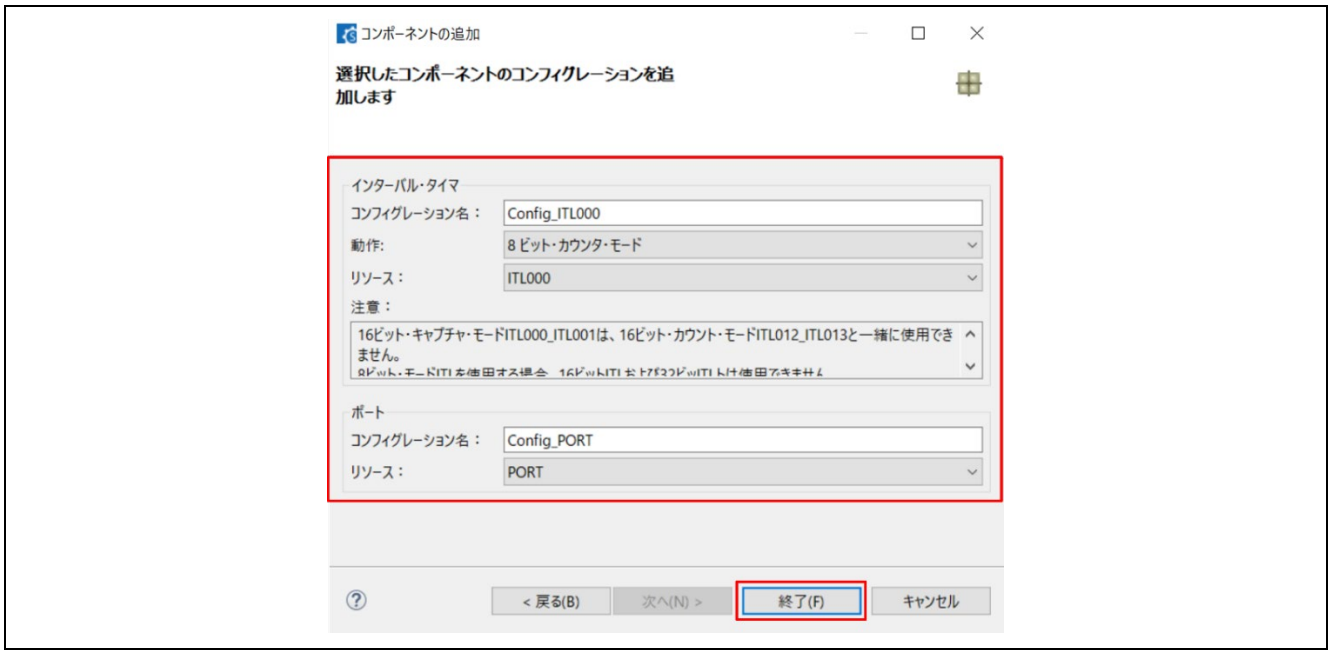


図 9-2　インターバル・タイマとポートのリソース設定

3. インターバル・タイマの設定を行います。”Config_ITL000”コンポーネントを選択して、以下のように設定します。

図 9-3　Config_ITL000 の設定

4. LED に使用する端子の設定を行います。”ポート”モジュールで”P62”を High レベル出力に設定します。

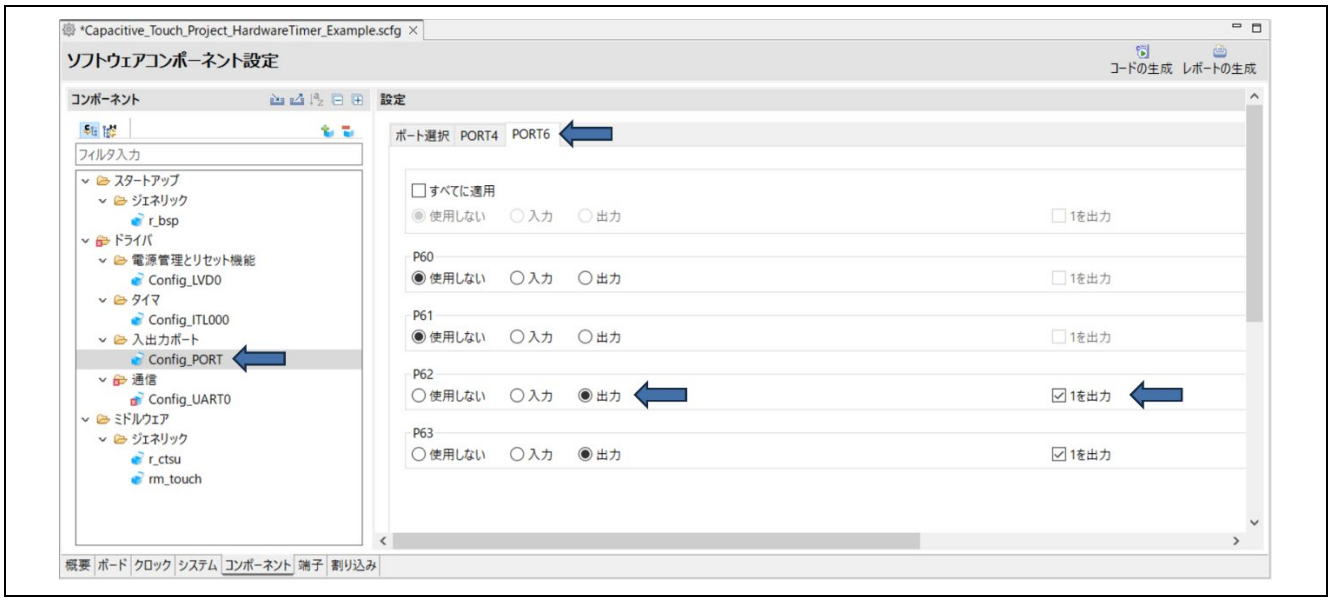


図 9-4　P62 の設定

5. スマート・コンフィグレータの右上の アイコンをクリックして、コードの生成を行います。

以降は「8. QE for Capacitive Touch の設定」に従い設定を行ってください。

9.1.2　フローチャート（ハードウェアタイマ）

図 9-5 にハードウェアタイマを使用したタッチ計測制御処理のフローチャートを示します。

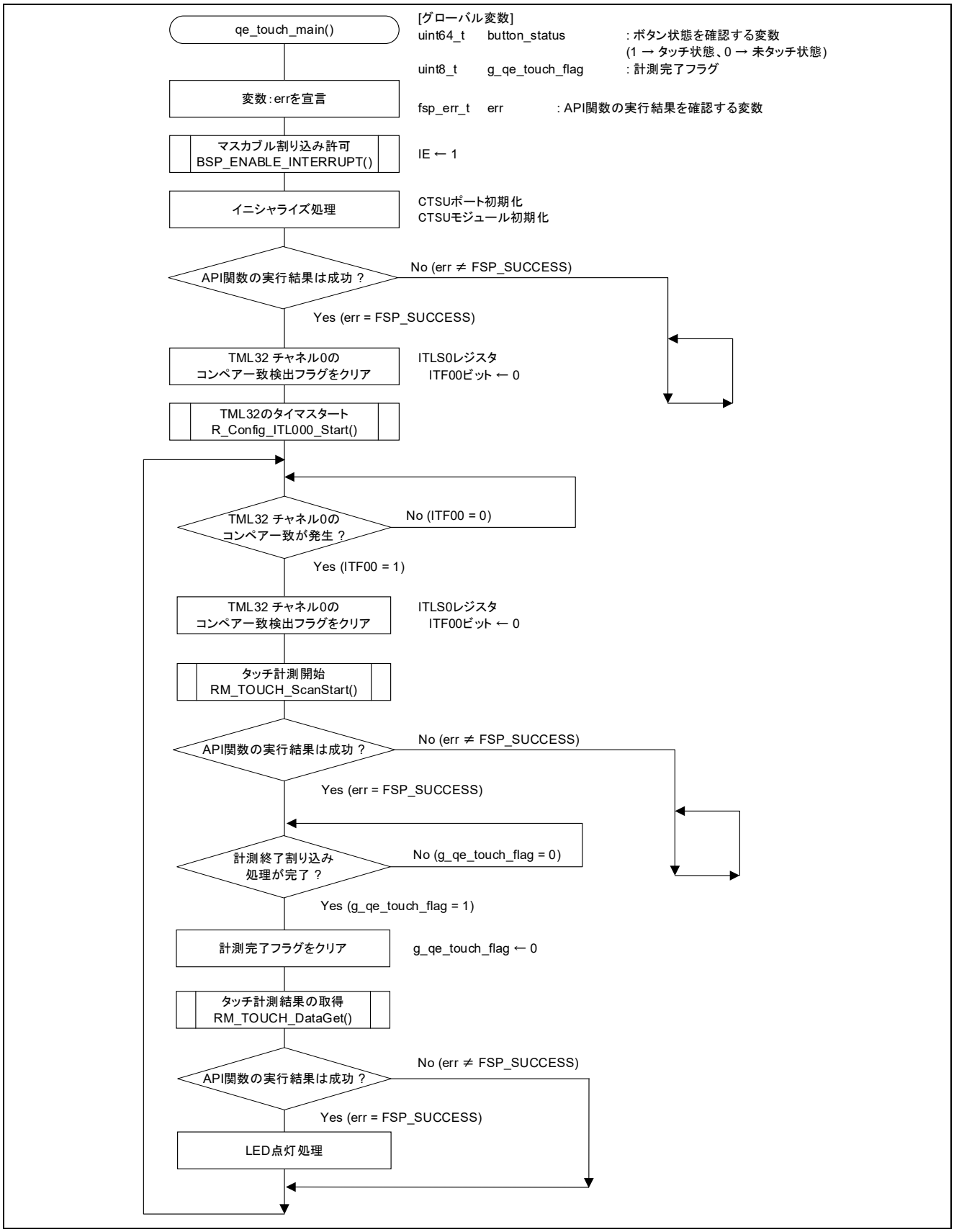


図 9-5　ハードウェアタイマを使用したタッチ計測制御処理

9.1.3　サンプルコード（ハードウェアタイマ）

ハードウェアタイマでのタッチ計測のプログラム実装例 (qe_touch_sample.c) を以下に示します。

```
/***********************************************************************
*
* FILE : qe_sample_sample.c
* DATE : 2025-02-25
* DESCRIPTION : CTSU2L Program for RL78
*
* NOTE:THIS IS A TYPICAL EXAMPLE.
*
***********************************************************************/
#include "qe_touch_config.h"
#include "Config_ITL000.h"

void R_CTSU_PinSetInit(void);
void qe_touch_main(void);

uint64_t button_status;
#if (TOUCH_CFG_NUM_SLIDERS != 0)
uint16_t slider_position[TOUCH_CFG_NUM_SLIDERS];
#endif
#if (TOUCH_CFG_NUM_WHEELS != 0)
uint16_t wheel_position[TOUCH_CFG_NUM_WHEELS];
#endif

void qe_touch_main(void)
{
    fsp_err_t err;
    BSP_ENABLE_INTERRUPT();
    /* Initialize pins (function created by Smart Configurator) */
    R_CTSU_PinSetInit();
    /* Open Touch middleware */
    err = RM_TOUCH_Open(g_qe_touch_instance_config01.p_ctrl,
g_qe_touch_instance_config01.p_cfg);
    if (FSP_SUCCESS != err)
    {
        while (true) {}
    }
    ITLS0 &= ~_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE;

    R_Config_ITL000_Start();

    /* Main loop */
    while (true)
    {
        while (_00_ITL_CHANNEL0_COUNT_MATCH_NOT_DETECTE == (ITLS0 &
_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE)) {}
        ITLS0 &= ~_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE;
```

```
        /* for [CONFIG01] configuration */
        err = RM_TOUCH_ScanStart(g_qe_touch_instance_config01.p_ctrl);
        if (FSP_SUCCESS != err)
        {
            while (true) {}
        }

        while (0 == g_qe_touch_flag) {}

        g_qe_touch_flag = 0;

        err = RM_TOUCH_DataGet(g_qe_touch_instance_config01.p_ctrl,
&button_status, slider_position, NULL);

if (FSP_SUCCESS == err)
        {
            /* TODO: Add your own code here. */
            if (0 != button_status)
            {
                P6_bit.no2 = 0;
            }
            else
            {
                P6_bit.no2 = 1;
            }
        }
    }
}
```

10. 参考ドキュメント

〇ユーザーズマニュアル
- RL78/G22 ユーザーズマニュアル ハードウェア編 (R01UH0978)
- RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015)

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

〇テクニカルアップデート／テクニカルニュース
(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

〇ユーザーズマニュアル：開発環境
- RL78/G22 Fast Prototyping Board ユーザーズマニュアル (R20UT5121)

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

〇アプリケーションノート
- 静電容量センサマイコン 静電容量タッチ導入ガイド (R30AN0424)
- RL78 ファミリ スタンドアロン版 QE を使用した静電容量タッチアプリケーションの開発 (R01AN6574)
- シリアルポートを使用した RL78 デバッグ機能 (R20AN0632)

- RL78 ファミリ CTSU モジュール Software Integration System (R11AN0484)
- RL78 ファミリ TOUCH モジュール Software Integration System (R11AN0485)
- 静電容量センサマイコン 静電容量タッチ電極デザインガイド (R30AN0389)
-
- RL78 ファミリ QE と SIS を使用した静電容量タッチアプリケーションの開発 (R01AN5512)

(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)

ホームページ
- ルネサス エレクトロニクスホームページ
  https://www.renesas.com/
- Fast Prototyping Board 関連ページ
  https://www.renesas.com/fast-prototyping-board
- QE for Capacitive Touch 関連ページ
  https://www.renesas.com/qe-capacitive-touch
- 静電容量センサユニット関連ページ
  https://www.renesas.com/solutions/touch-key

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Mar.20.23	-	初版
2.00	May.28.25	1	要旨を更新
		1	QE for Capacitive Touch の説明を注として追加
		4	1.　概要 章 を追加
		4	本書の対応デバイスを修正
		4	表 2-1 を更新
		4	表 2-1 に SIS モジュールを追加
		4	表 2-3 動作確認条件 を追加
		5	2.1 QE for Capacitive Touch の機能 章を追加(内容は Rev.1.00 の 1.システム概要章と同等)
		5	図 2-1 を更新
		6	3.1 章に CS+とスマート・コンフィグレータのインストール手順を追加 3.1 章のタイトルを “開発ツールのインストール手順” に変更
		6	スタンドアロン版 QE for Capacitive Touch をインストール時の注意事項を追加
		7	3.2 章に文章および注を追加
		7	表 3-1 ボードのジャンパ設定 を追加
		9	表 4-1 を更新
		10	本アプリケーションノートのチューニングおよびモニタリング時の通信方法を追記
		10	Rev1.00 の 7.3.3 章内のモニタリング時の制限事項を 5.1 章に移動し、備考として記載
		10	図 5-1 を更新
		10	5.1 章に付属サンプルコードの説明を追加
		10	表 5-1 付属のサンプルコードの概要 を追加
		11	表 5-2 を更新
		13	7 章に “電源検出回路 (LVD)” と ”ポート機能 (PORT)” の設定項目を追加
		14	7.2 章に図 7-5 と EVDD 設定の手順を追加
		15	7.3 章の文章を更新
		16	SIS モジュールのダウンロード方法について注意を追加
		17	RL78/G16 の電圧検出機能の設定方法について注意を追加
		17-18	7.4 コンポーネント追加 章を追加し、使用する全コンポーネントの追加手順と各リソースの設定内容を記載
		17-18	図 7-10, 図 7-11, 図 7-12 を追加
		19-26	7.5 スマート・コンフィグレータによるコンポーネント設定の変更 章を追加し、各コンポーネントの設定内容を集約して記載
		20	付属サンプルコードでの CTSU コンポーネントの設定を更新
		20	回路作成時の注意事項を記載
		20	図 7-14 を更新

		21	Touch コンポーネントの設定内容を図 7-15 中の表記に統一
		22	7.5.3 章の文章更新
		23	UART 通信端子設定時の注意事項を追加
		24	“7.5.4 LVD コンポーネント設定” を追加
		24	図 7-19 LVD コンポーネントの設定 を追加
		25	“7.5.5 PORT コンポーネント設定” を追加
		25	図 7-21 PORT コンポーネントの設定 を追加
		25	図 7-22 P62 と P63 を High 出力に設定 を追加
		26	7.5.6 ボードサポートパッケージ 章を追加 (内容は Rev.1.00 の 7.6 コード生成 章の項目 1 と同等)
		27	図 7-24, 図 7-25 を更新
		28	図 7-26 コード生成時の注意画面 を追加
		28	図 7-27 のユーザ・オプション・バイト値を修正
		33	8.3 章 項目 2 の文章を更新
		33	8.3 章 項目 3 に文章および注意を追加
		33	図 8-8 を更新
		34	8.3 章 項目 6 に文章を追加
		34	図 8-11 を更新
		35	8.3 章 項目 8 の文章を更新
		35	図 8-13 “qe_gen”フォルダ新規作成 を追加
		35	フォルダ名の誤記を修正
		36	8.3 章 項目 9 に文章を追加
		36	8.3 章 項目 10 に供給電圧の設定に関する留意点を追加
		36	図 8-15 を更新
		38-40	開発手順 (項目 14 から 18 まで) の変更 ・項目 14 に qe_gen フォルダ追加方法の説明を移動。 ・標準・数学ライブラリの設定内容及び図 8-21 (Rev.1.00 の P.35 に記載) を削除
		38-39	8.3 章 項目 14,15,17 の文章を更新
		39	ユーザ・オプション・バイト値を修正
		39	図 8-24 のユーザ・オプション・バイト値を修正
		43	図 8-31, 図 8-32 を更新
		43	8.4 章 項目 3,4,5 の文章を更新
		45	8.4 章 項目 9 の文章を更新
		47	図 8-40 を更新
		48	8.5.1 章 項目 1,2,3 の文章を更新
		49	8.5.1 章 項目 4 に文章を追加
		49	図 8-44, 図 8-45 を更新
		50	8.5.1 章 項目 6 のステータス・チャートへ表示するタッチインタフェースの設定方法を修正
		50	図 8-46, 図 8-47 を更新
		51	8.5.1 章 項目 7 の A の文章を更新
		51	図 8-48 を更新
		51	8.5.1 章 項目 7 の B にデータ収集時の留意点を追加

		51	図 8-49 データ収集の開始 を追加
		52	8.5.1 章 項目 7 の C の文章を更新
		52	図 8-50 データ収集の終了 を追加
		52-54	図 8-51, 図 8-52, 図 8-53, 図 8-54 を更新
		54	ソフトウェアタイマのサンプルコード (Rev.1.00 の P.54, P55 に記載) を削除
		55	図 8-56 の CTSU ポート初期化、CTSU モジュール初期化をイニシャライズ処理に変更
		55	図 8-56 に図のタイトルを追加
		56	9.1 章の文章を更新
		56	9.1 章に応用例での開発時の留意点を追加
		57	9.1.1 章 項目 2 の文章を更新
		57	図 9-2 を更新
		57	9.1.1 章 項目 3 の文章を更新
		58	9.1.1 章に項目 5 以降の設定手順の補足を追加
		59	図 9-5 を変更 ・CTSU ポート初期化、CTSU モジュール初期化をイニシャライズ処理に変更 ・LED 点灯処理部を LED 点灯処理に変更
		59	図 9-5 に図のタイトルを追加
		60	9.1.3 章に文章を追加
		62	10 章に参考ドキュメント (R01AN5512) を追加 (Rev.1.00 の要旨から移動)
		62	参考ドキュメント (ホームページ) に Fast Prototyping Board 関連ページを追加

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うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因し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。