Renesas Synergy™ プロジェクト

DK-S124用 CTSU スライダ機能サンプルアプリケーション（SSP v1.1.0）

R12AN0043JU0100
Rev.1.00
2016.08.03

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントをご参照ください。

要旨

本アプリケーションノートでは、マイクロコントローラ用Development Kit DK-S124を実例として Renesas Synergy™ MCUシステムの静電容量式タッチセンシングユニット（CTSU）とボタン機能の基本的な利用モデルを紹介します。このアプリケーションノートでは、Synergy ソフトウェアパッケージ（SSP）v1.1.0のCTSUフレームワークを使用したCTSUスライダ機能をどのように作成するかを示します。使用にあたり、Synergy MCU用の e²studio ISDE コンフィグレーションツールの使い方を理解されていることを前提としています。フレームワーク、コールバック機能、SSP API などのSSPの概念も理解しておく必要があります。それらの概念については、SSP ユーザーズマニュアルの 2.1 SSP アーキテクチャ を参照してください。必ずBlinkyプロジェクトのビルドとデバッグを正しく実施した上で、アプリケーションプログラムの演習に進んでください。このアプリケーションノートには、以下の題目を含んでいます。

1. CTSU概要
2. 本アプリケーションノートで使用するハードウェア
3. SSP 静電容量式タッチパネルフレームワーク要旨
4. CTSUアプリケーションプログラムコンフィグレーションデータの概要
5. サンプルプロジェクトの再生成手順
   a. RTOSを含まない新規プロジェクトの生成
   b. CTSUスレッドの生成
   c. CTSUスライダフレームワークの生成
   d. CTSUボタンフレームワークの生成
   e. 静電容量式タッチパネルCTSUコンフィグレーションファイルのコピー
   f. コールバック機能の生成

対象デバイス

S124

対象キット

DK-S124 V2.0

SSP バージョン

SSP v1.1.0

e²Studio バージョン

5.0.0.043

目次

1. CTSU概要

CTSUはタッチセンサの静電容量を測定します。ソフトウェアで静電容量の変化を判定することによって、指などがタッチセンサに接触したことをCTSUが検出できます。静電容量の検出方式には、自己容量方式と相互容量方式があります。このアプリケーションノートは、スライダとボタン機能に自己容量シングルスキャンモードを利用します。

表 1に、S124のCTSU仕様をまとめます。詳細については、S124ユーザーズマニュアルを参照してください。

表 1 CTUS 仕様

項目		説明
動作クロック		PCLKB、PCLKB/2、またはPCLKB/4
端子	静電容量計測	31チャネル（TS00～TS28、TS30、TS31）
	TSCAP	LPF（Low Pass Filter）接続端子
計測モード	自己容量シングルスキャンモード	自己容量方式で1チャネルの静電容量を計測
	自己容量マルチスキャンモード	自己容量方式で複数チャネルの静電容量を連続して計測
	相互容量フルスキャンモード	相互容量方式で複数チャネルの静電容量を連続して計測
ノイズ対策		同期系ノイズ対策、高域ノイズ対策
計測開始条件		• ソフトウェアトリガ • 外部トリガ（ELCからのELC_CTSU）

CTSUは、図 1に示すようにステータス制御部、トリガ制御部、クロック制御部、チャネル制御部、ポート制御部、センサドライブパルス生成部、計測部、割り込み部、 I/0レジスタで構成されます。SSP CTSUフレームワークにより、それらのブロックやレジスタのレベルでモジュールを制御する必要はありません。

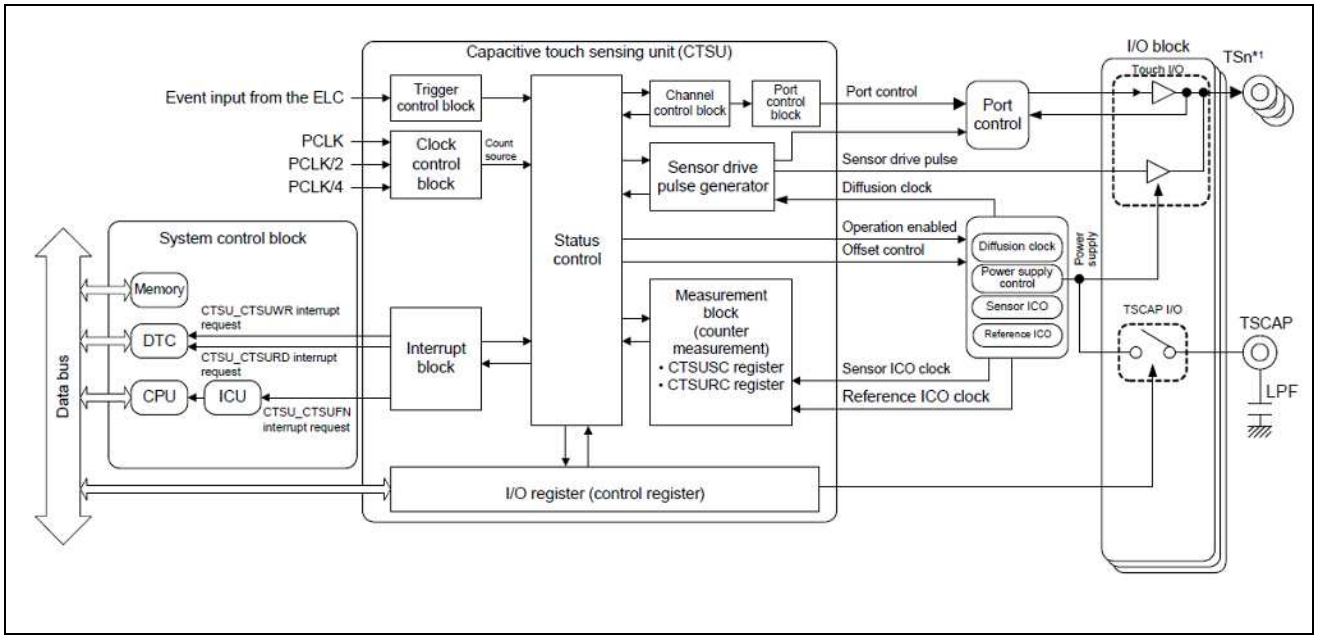

図 1 CTSUのブロック図

スライダ機能は、スライド動作の検出と動作に対しての応答を同時に行います。このアプリケーションノートでは、サンプルプログラムがスライダ上で指をスライドする動作を検出し、ボード上の3つのLEDを点灯して応答します。スライダの先端（左側の細い側、－印）から終端（右側の太い側、＋印）へ指をスライドするとき、ボード上の3つのLEDが暗い状態から明るい状態に変化します。ボタン機能は、ボタンが押されたこ

との検出とそれに対しての応答を行います。サンプルプロジェクトは、ボタン1の押下を検出し、LED2（橙色）の点滅で応答します。また、ボタン2の押下を検出し、LED3（緑色）の点滅で応答します。

圧縮ファイル形式での機能プロジェクトは、本アプリケーションノートに添付されています。圧縮ファイル（CTSU_Slider_Button_Example.zip）をインポート、コンパイル、ダウンロードし、スライダおよびボタン機能を実行してください。また、本アプリケーションノートではSSPスライダ/ボタンフレームワークの機能の使用法について段階に説明し、このアプリケーションプログラムを再利用できるようにします。

図 2 にDK-S124 V2.0 ボード用のCTSUコンポーネントを示します。

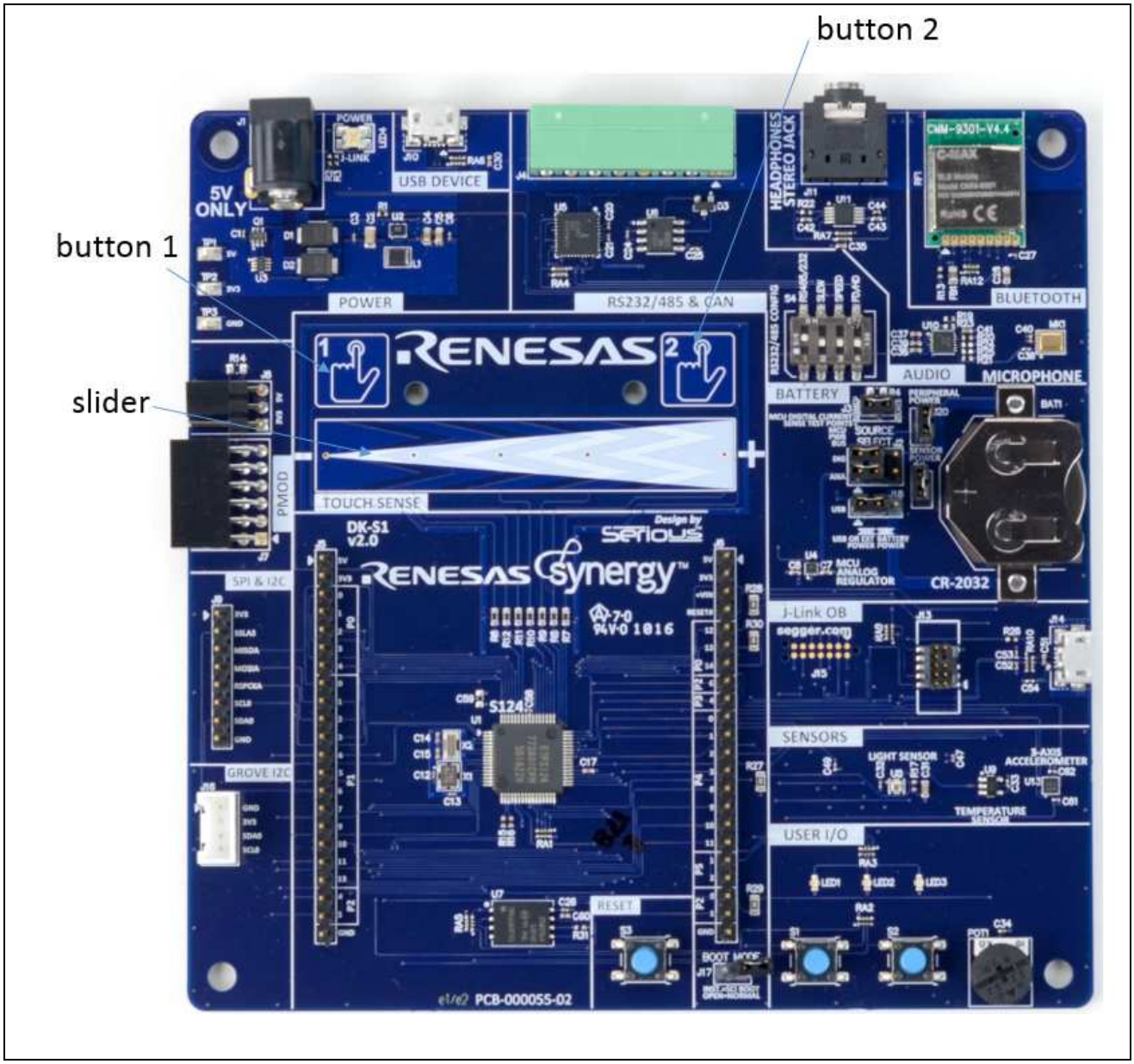

図 2　DK-S124 CTSU コンポーネント

2.　静電容量式タッチパネルSSPフレームワーク概要

静電容量式タッチパネルスライダフレームワークは、システムにより初期化されたすべてのスライダ構成のCTSUデータを処理します。静電容量式タッチパネルスライダフレームワークは、データが処理されるごとにコールされるコールバックをCTSUフレームワーク層に登録します。スライダフレームワークは、このデータ（raw値）を使い、タッチリリースがあったか、あった場合どこで起こったかを判断します。もし状態変化があった時は、フレームワークはスライダ構成テーブルの順に各スライダのイベントと位置のコールバック

をコールします。このアプリケーションノートは、DK-S124ボードで使用できるシングルスライダだけを使用します。

2.1 静電容量式タッチパネルSSPフレームワークで使うリソース

表 2 CTSUフレームワークインターフェースで使うリソース

リソース	ISDEタブ	選択
フレームワークCTSUドライバ	スレッド	Framework>Input>CTSU Framework on sf_touch_ctsu
CTSU処理	ICU	3つのCTSU割り込みを同じプライオリティで有効化します。

SSP v1.1.0は、Synergy MCUのS7G2、S3A7と、S124に対応します。このアプリケーションノートは、DK-S124ボードのS124に実装された静電容量式タッチパネルスライダフレームワークを例にとって説明します。表3と、表4は、スライダフレームワークに使用する追加リソースを示しています。

表3 CTSU スライダフレームワークインターフェースで使用する追加リソース

リソース	ISDEタブ	選択
フレームワークCTSUスライダドライバ	スレッド	Framework>Input>Cap Touch Slider Framework on sf_touch_slider_ctsu
CTSU HAL ドライバ	スレッド	-

表4 CTSU ボタンフレームワークインターフェースで使用する追加リソース

リソース	ISDEタブ	選択
フレームワークCTSUボタンドライバ	スレッド	Framework>Input>Cap Touch Button Framework on sf_touch_button_ctsu
CTSU HAL ドライバ	スレッド	-

2.2 CTSUスライダ/ボタンフレームワークの理解

SSP CTSUスライダフレームワークは、タッチデータ処理の後でイベントを生成します。このサンプルアプリケーションプログラムはSF_TOUCH_CTSU_SLIDER_STATE_HELDをトリガに、応答を生成します。ユーザからのスライダ動作が検出されたとき、通知されたスライダ上の位置を見ることで、コールバック機能で必要な応答を生成することができます。

表 5CTSUスライダの状態

名称	説明
SF_TOUCH_CTSU_SLIDER_STATE_NO_CHANGE	スライダは解放状態
SF_TOUCH_CTSU_SLIDER_STATE_INITIALIZED	スライダは押されている状態
SF_TOUCH_CTSU_SLIDER_STATE_TOUCHED	スライダが押された
SF_TOUCH_CTSU_SLIDER_STATE_RELEASED	スライダは解放された
SF_TOUCH_CTSU_SLIDER_STATE_CLOSED	スライダは無効となり、イベント生成をしない
SF_TOUCH_CTSU_SLIDER_STATE_MULTI_TOUCH	1か所以上のタッチエレメントがタッチされた
SF_TOUCH_CTSU_SLIDER_STATE_DISABLED	スライダはアップデートが無効になっている
SF_TOUCH_CTSU_SLIDER_STATE_HELD	スライダはホールドされている。（押され続けている）

表6　CTSUボタンの状態

名称	説明
TOUCH_BUTTON_STATE_RELEASED	ボタンは解放状態
TOUCH_BUTTON_STATE_PRESSED	ボタンは押されている状態
TOUCH_BUTTON_STATE_LONG_HOLD	ボタンは長時間押されている状態（持続時間は、sf_touch_ctsu_button_config.h）
TOUCH_BUTTON_STATE_STUCK	ボタンは短時間押されている状態（持続時間は、sf_touch_ctsu_button_config.h）
TOUCH_BUTTON_STATE_INITIAL	ボタンが正しく初期化された
TOUCH_BUTTON_STATE_CLOSING	ボタンは無効となり、イベント生成をしない
TOUCH_BUTTON_STATE_MULTI_TOUCH	1か所以上のタッチエレメントがタッチされた
TOUCH_BUTTON_STATE_DISABLED	ボタンはアップデートが無効になっている

2.3　CTSU構成データ

サンプルプロジェクトは、CTSU構成データのセットを以下のディレクトリに格納しています。¥CTSU_Slider_Button_Example¥src¥captouch_configs¥　CTSU構成データはDK-S124ハードウェア専用です。3.3章で、サンプルプロジェクトの再生成を段階的に説明します。ステップ7では、この¥captouch_configフォルダを、これから作成する新しいアプリケーションプログラムにコピーすることが必要です。

3.　CTSUスライダアプリケーションプログラムのビルド

本章では、e^{2}studio ISDEとSynergyソフトウェアパッケージ（SSP）を使ってCTSUスライダとボタンのアプリケーションプログラムを生成する詳細を説明します。SSPパッケージでサポートされるスライダ・ボタンフレームワークにより、アプリケーションプログラムではCTSUイベントからのコールバックを取り扱えば、ほかにCTSUレジスタや基本的な静電容量式タッチパネル処理機能を扱う下位レベルのソフトウェアを必要としません。

3.1章で、サンプルプロジェクトで実装され、所望のアプリケーションプログラムで引用可能なコールバック機能について説明します。3.2章は、本アプリケーションノートで使用しているスライド動作検出方法の詳細を示します。独自に作成するアプリケーションプログラムの要件に基づいて、この方法を調整して使用してください。スライダ・ボタンアプリケーションプログラムの構築過程を図示するため、3.3章ではサンプルプロジェクトの再生成を段階的に説明します。CTSUコンポーネントを既存のアプリケーションプログラムに追加するか、このサンプルプロジェクトを開発するアプリケーションプログラムのスタート地点とすることで、同様の手順をたどることが可能です。

より詳細については、サンプルプロジェクトパッケージCTSU_Slider_Button_Example.zipを参照してください。

3.1　スライダ・ボタンコールバック機能の取り扱い

サンプルプロジェクトは、スライダ・ボタンコールバック機能をサンプルプロジェクトフォルダ：¥CTSU_Slider_Button_Example¥src¥ 内の ctsu_thread_entry.c に実装しています。ctsu_thread_entry.c は、Synergyコンフィギュレータで生成します。表7 は、ctsu_thread_entry.cに実装されている機能のまとめです。これらの機能は、標準的なスライダ・ボタンアプリケーションプログラムを構築するためにSSPスライダ・ボタンフレームワークで作成が必要な主な機能を提供します。

3.3章では、サンプルプロジェクトの再生成を段階的に説明します。ctsu_thread_entry.c ファイルを新しいアプリケーションにコピーし、スライド動作・ボタン押下イベントに対して所望の応答となるように修正することができます。

表 7 ctsu_thread_entry.cの機能

機能	説明
g_button_framework_user_callback	ボタン押下コールバック機能の実装（Synergyコンフィギュレータで定義）
g_slider_framework_user_callback	スライダタッチコールバック機能の実装（Synergyコンフィギュレータで定義）
CB_Self_Slider_0	• g_slider_framework_user_callbackから呼び出されるサブファンクション • 表に示したイベントへのレスポンスの実装
CB_Self_Button_TS30	• g_button_framework_user_callback から呼び出されるサブファンクション • ボタン1が押下されると起動 • 表に示したイベントへのレスポンスの実装
CB_Self_Button_TS09	• g_button_framework_user_callback から呼び出されるサブファンクション • ボタン1が押下されると起動 • 表に示したイベントへのレスポンスの実装
pwm_led_brightening	• CB_Self_Slider_0から呼び出されるサブファンクション • 暗い状態から明るい状態にLEDが変化します

CB_Self_Slider_0機能にスライド動作検出機能が実装されています。この機能は、 g_button_framework_user_callback から呼び出されます。表5に示すスライダイベントに応答します。SF_TOUCH_CTSU_SLIDER_STATE_HELD状態の取り扱いにおいて、スライド動作を検出するようなアルゴリズムが実装されています。3.2章でスライド動作の検出について詳細を説明しています。

スライド動作の検出を行うため、ユーザアプリケーションプログラムは、以下の章で説明されているようにわずかな量のスライド位置データ処理を行う必要があります。ボタン押下検出をするためにユーザ側で行うデータ処理はありません。

3.2 ユーザスライド動作の検出

サンプルプロジェクトでは、スライド動作検出に以下のマクロ定義、機能定義、変数を使用します。

図 に示す Update Hz 設定と、スライダ上でどれだけのスライド速度を検出するかによりマクロ定義が決定されます。スライダ位置には、100ｘ[センサ個数]の位置情報が通知されます。DK-S124の場合、センサのナンバリングは左から右に1、2、3、4、5です。そのため、通知されるスライダ位置は0-500の範囲となります。STARTLIMITとENDLIMITは、この計算をもとに決定されます。DETECTIONTIMEは、CTSUセンサ部のSynergyコンフィギュレータで定義されているUpdate Hz に関係しています。3.3章で、サンプルプロジェクトの再生成を段階的に説明します。Update Hzは、ステップ3の図8で設定されます。Update Hzが増えると、スライド動作の検出を妥当な回数実施できるだけの時間になるようにDETECTIONTIMEを増やす必要があります。

```
#define SIZEOFBUFFER     600 /* スライダ位置　一時バッファサイズ*/
#define DETECTIONTIME    8   /* 8回の指の動きが検出されたとき、スライド動作があったと見なす */
#define STARTLIMIT       100 /* スライド動作の開始位置を特定するため、この閾値より小さい必要がある */
#define ENDLIMIT         400 /* スライド動作の終了位置を特定するため、この閾値より大きい必要がある */

void CB_Self_Slider_0(sf_touch_ctsu_slider_callback_args_t * p_args);
static uint32_t detect_count = 0; /* 動きが検出された位置の数を追跡する */
volatile uint32_t result[SIZEOFBUFFER];  /* スライダ位置の一時保管 */
static uint32_t slider_data_index = 0;  /* 結果アレイへのインデックス */
int diff = 0;   /* 隣り合う2個のスライダの位置の違い */
```

上記の detect_count、result、slider_data_index、および diff は、スライド動作検出アルゴリズムの変数です。図 3 は、スライド動作検出の主な動作をまとめるために上記変数を参照する、上位のフローチャートです。

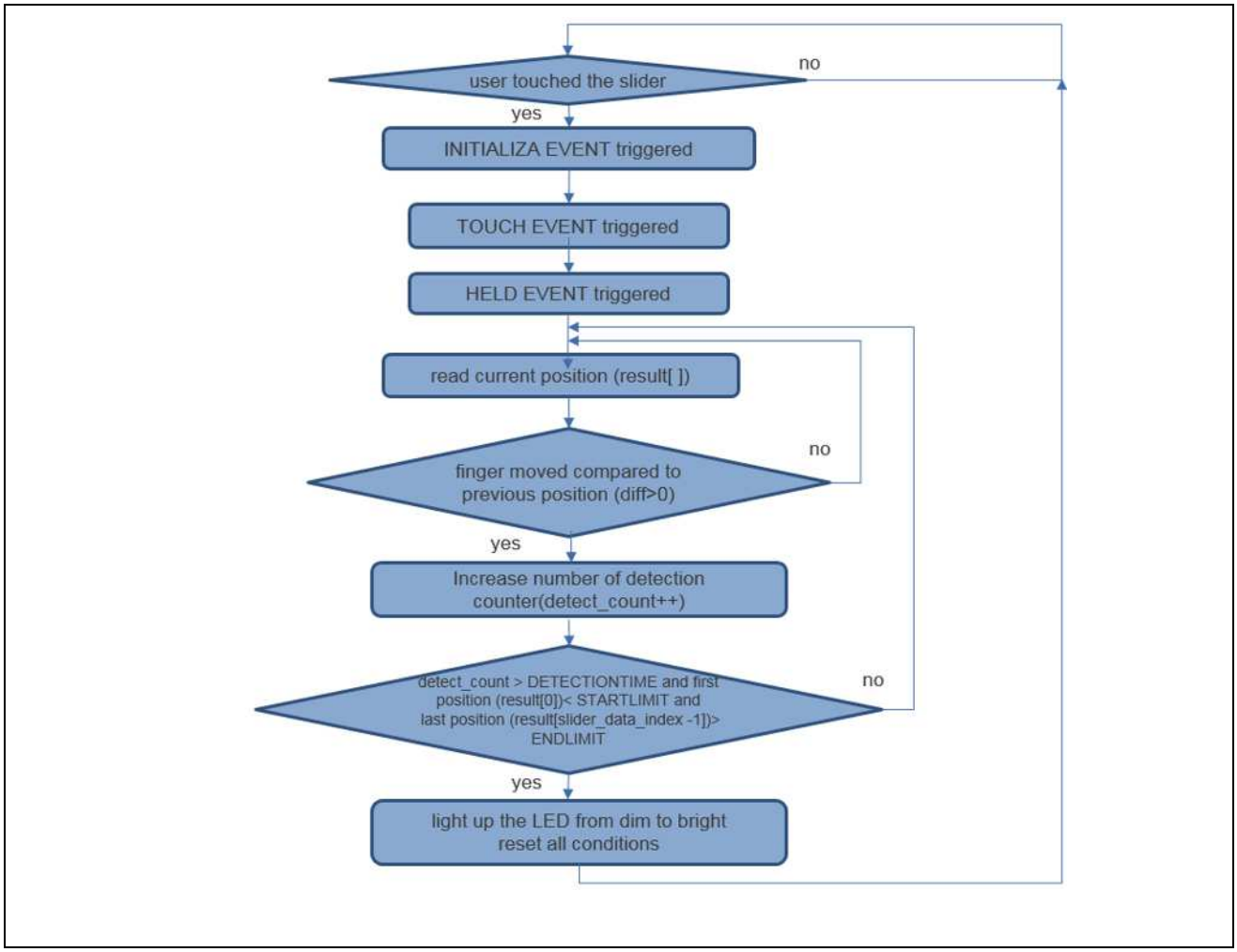

図 3　スライド動作検出

3.3　サンプルプロジェクト再生成の段階的な手順

添付のサンプルプロジェクトを作成する手順を、以下で段階的に説明します。各図で、(1) → (2) → (3) → (4)の順でステップに従ってください。

ステップ 1: RTOSを含まない状態で新規プロジェクトを作成します。

(1) 新規Synergyプロジェクトを作成してください。
(2): プロジェクト名を入力し、Synergyライセンスファイルをセットアップしてください。
(3): S124 DKボードを選択してください。
(4): S124-DK BSPを選択してください。

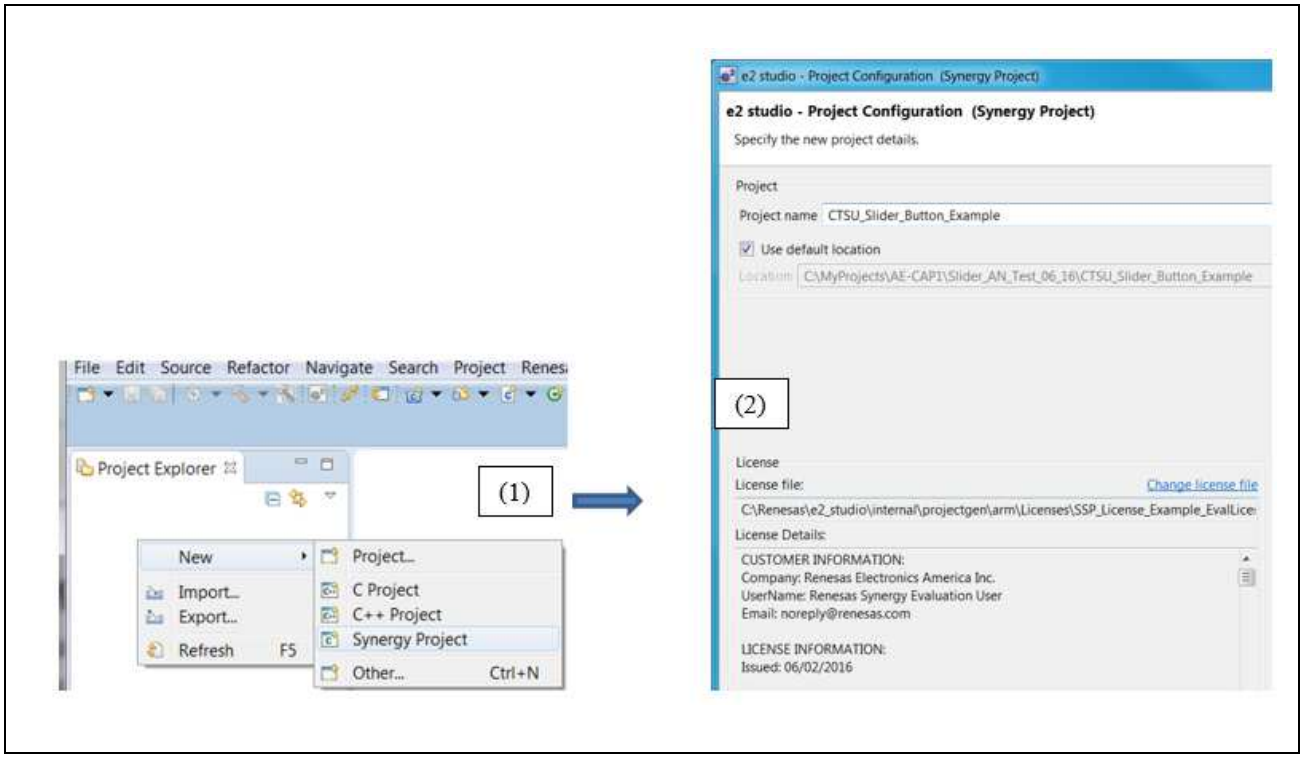

図 4　新規プロジェクトの作成

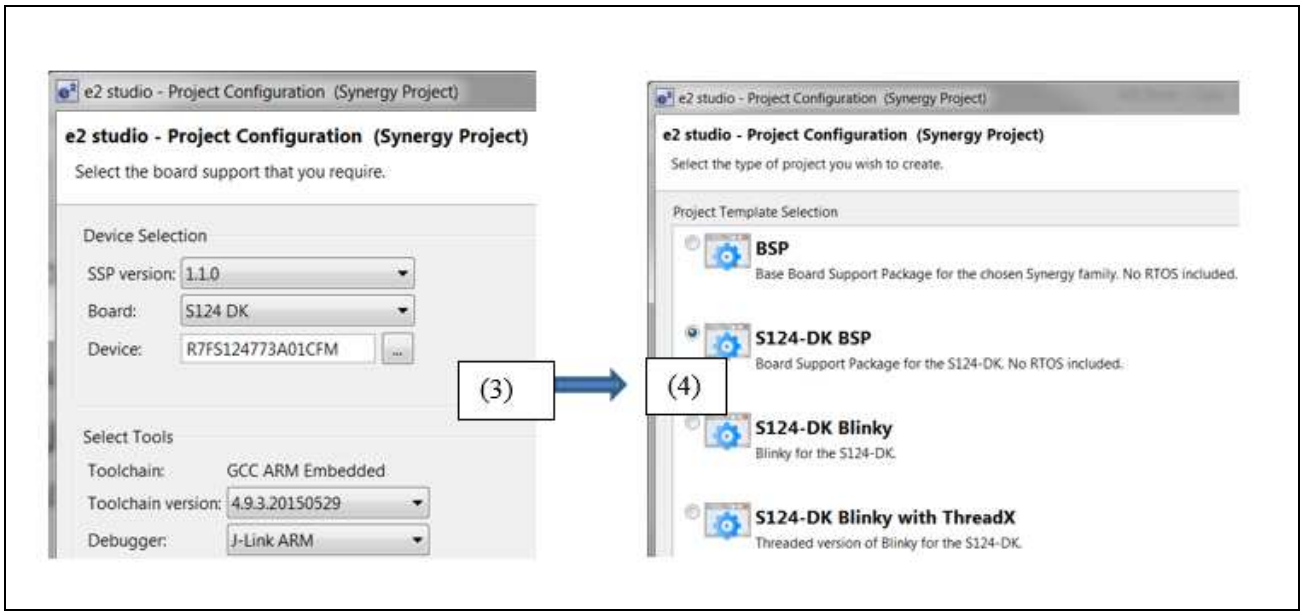

図 5　ボードとBSPパッケージの選択

ステップ 2: CTSUスレッドを作成してください。

(1): BSPプロパティ の“RTOS being used”に、ThreadXを選択してください。
(2): Threadタブの下で、‘+’アイコンをクリックして新規スレッドを作成してください。
(3): この新規スレッドのプロパティを、図 7に示すように設定してください。

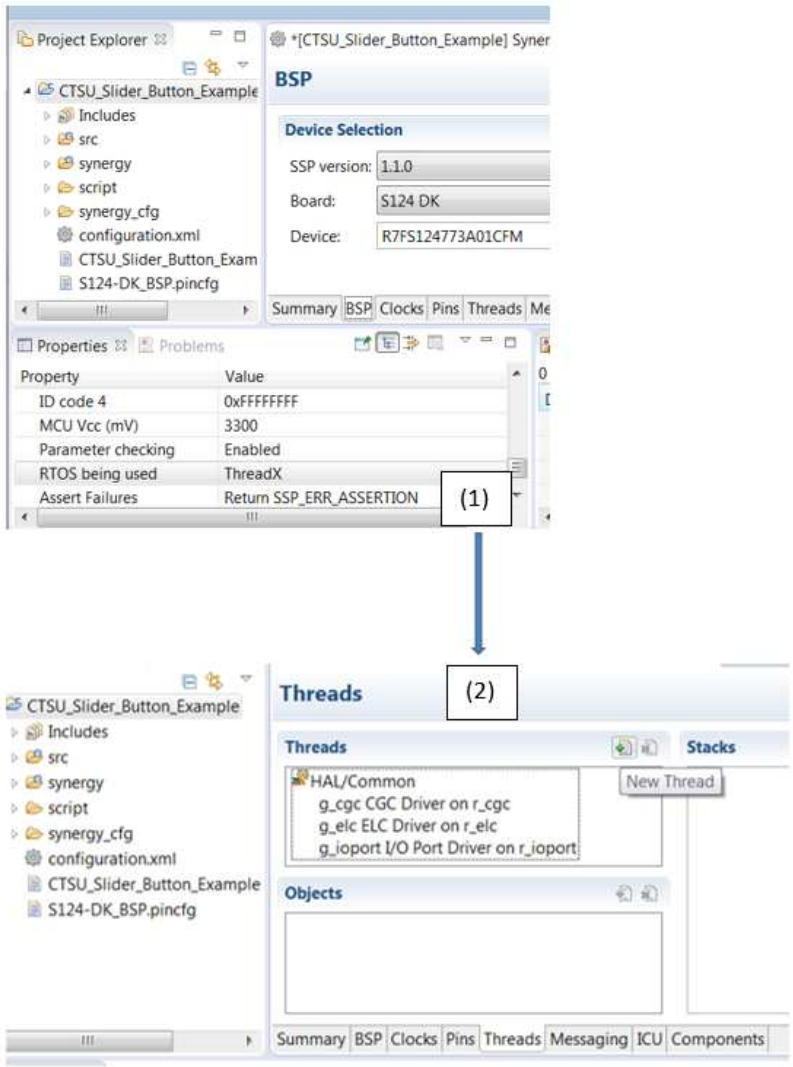

図 6　CTSUスレッドの作成

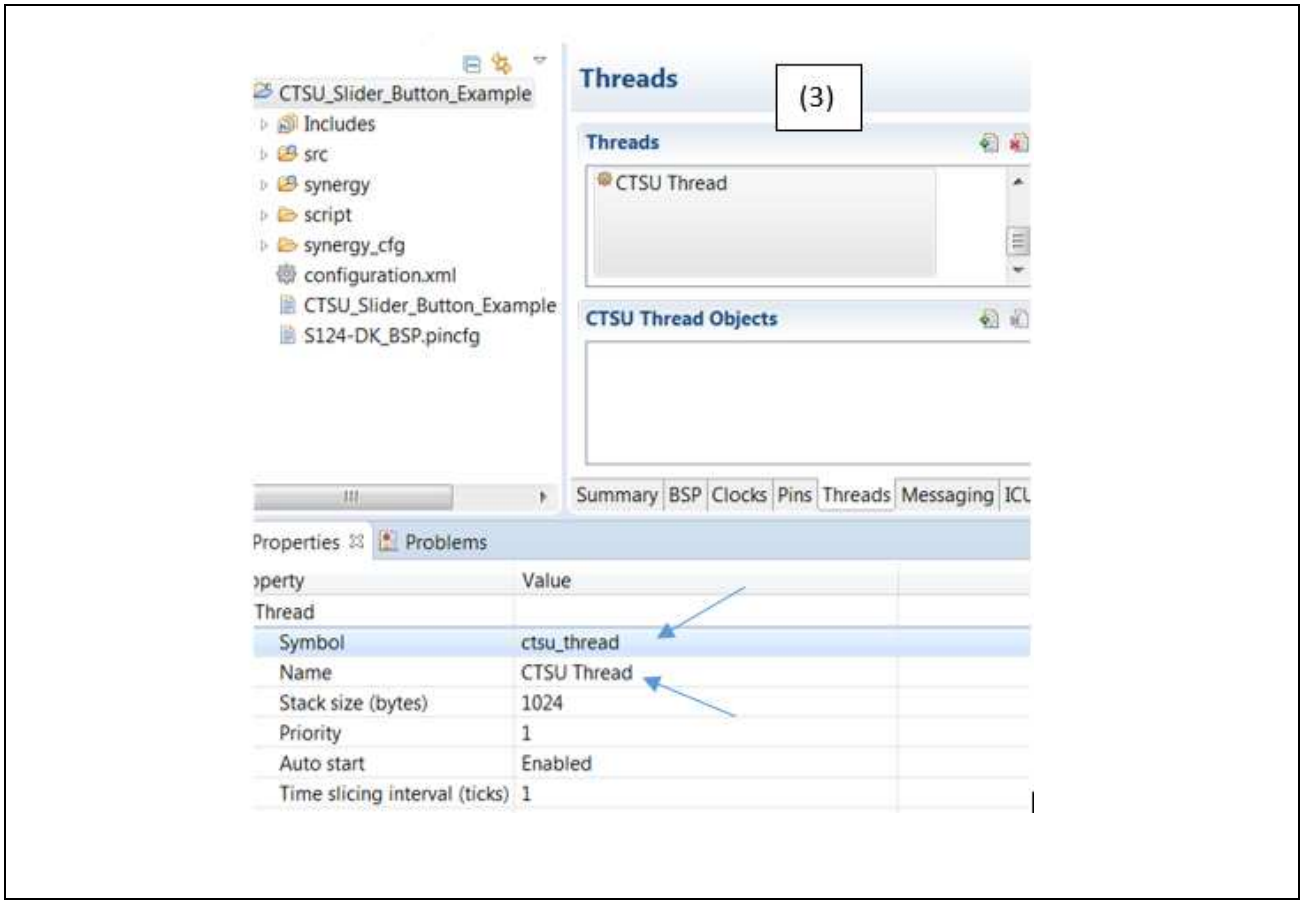

図 7　CTSUスレッドの構成

ステップ 3: CTSUスライダフレームワークを作成してください。

(1): “CTSU Thread”スレッドを新規作成する、をクリックしてください。“CTSU Thread Stacks”ウィンドウで、‘+’アイコンをクリックして“Cap Touch Slider/Wheel Framework on sf_touch_ctsu_slider”を追加します。

(2): ブルーボックスの“g_sf_touch_ctsu0 Cap Touch Framework on sf_touch_ctsu”をクリックし、Update Hzを20に設定します。

(3): 赤文字の“g_ctsu0 CTSU Driver on r_ctsu”をクリックし、このスクリーンショットに示す内容にプロパティを設定してください。Max. active channels ：‘7’、CTSU WRITE、CTSU READ、と、CTSU ENDのプライオリティ：Priority 1、モジュール名称：“g_ctsu”、CTSU configuration used：“g_ctsu_config_self”、にそれぞれ設定してください。全部のプロパティを設定したら、赤字の部分は普通の黒文字に変わります。

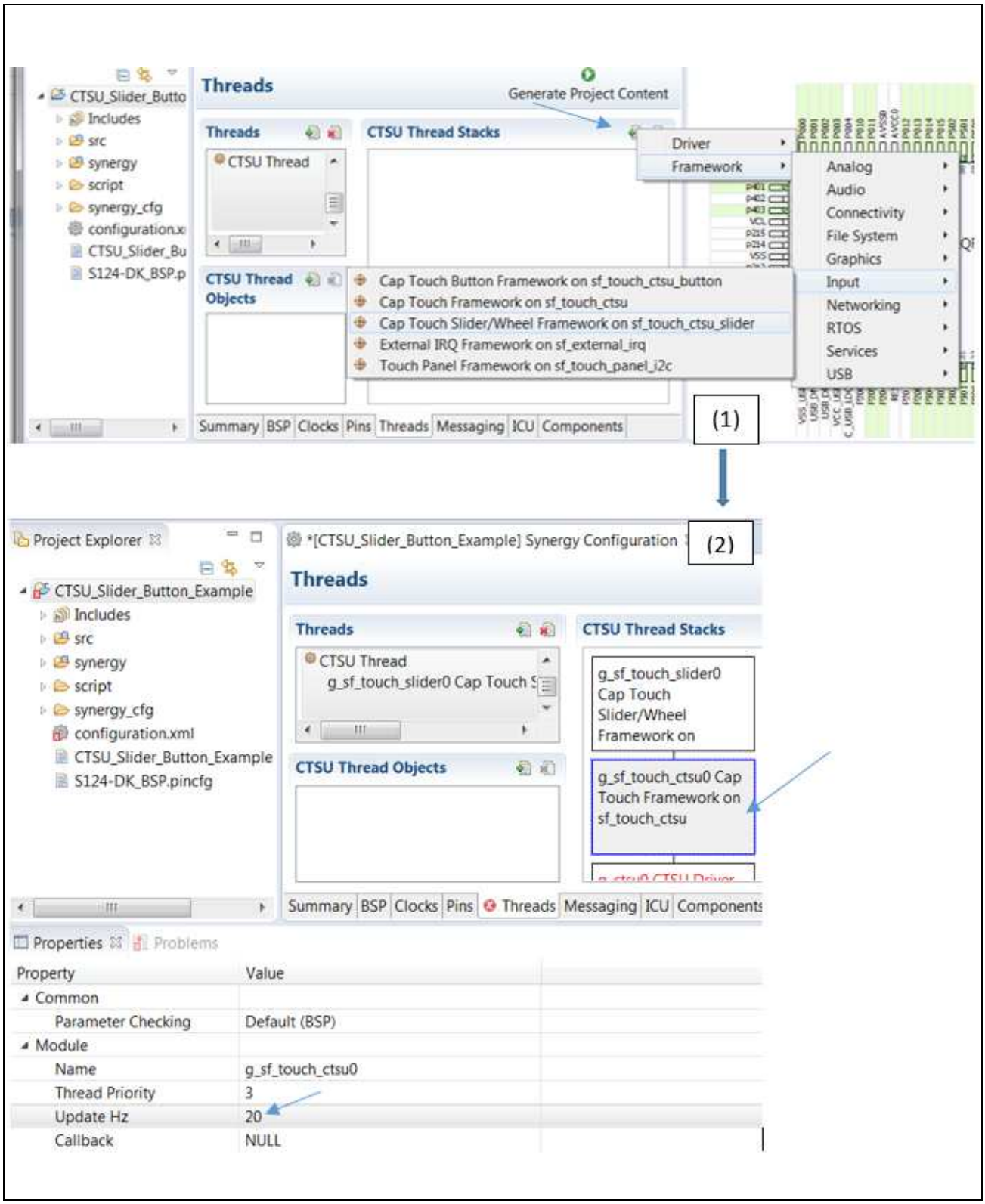

図 8　CTSUスライダフレームワークの生成

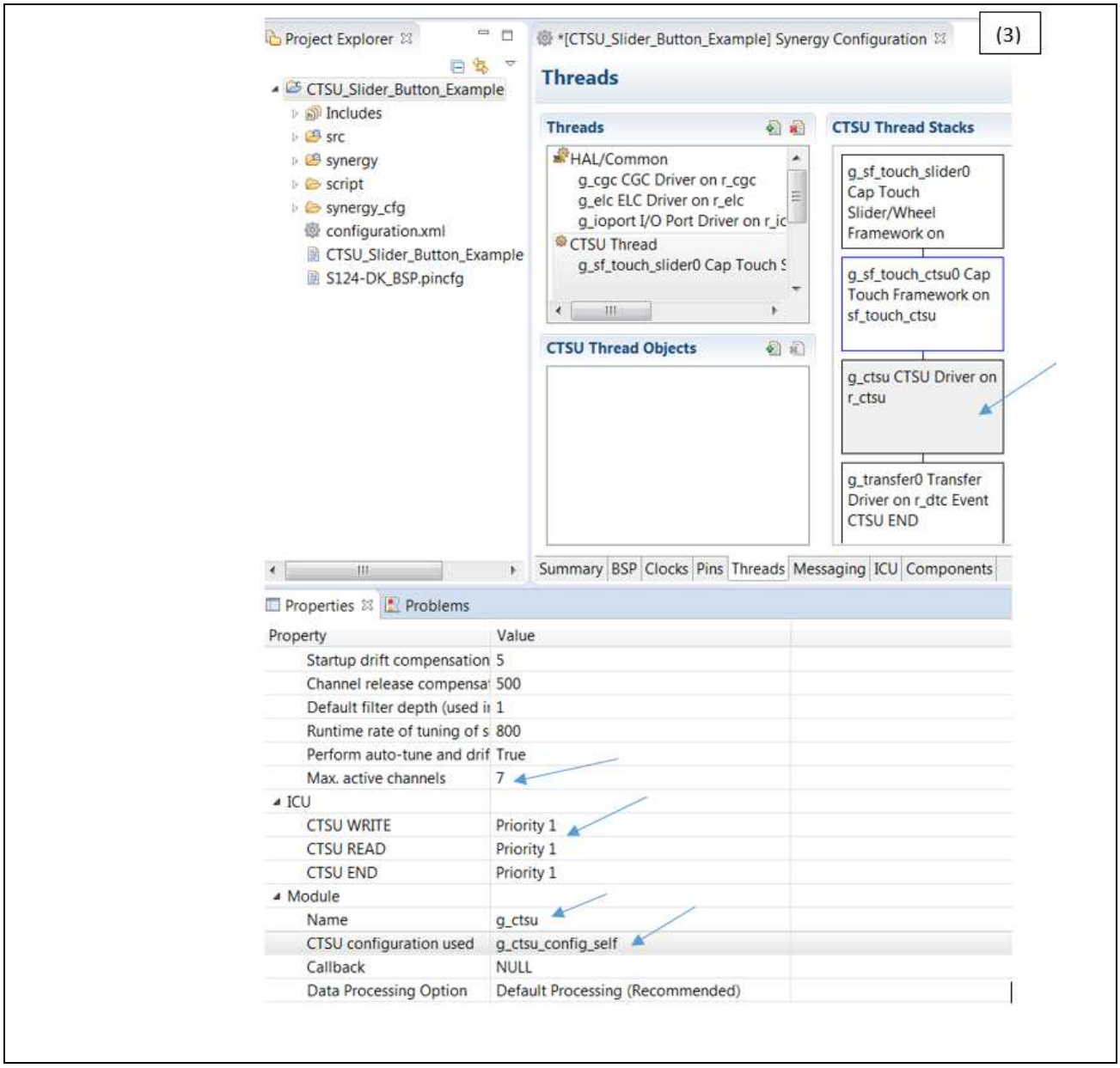

図 9　CTSUスライダドライバの構成

ステップ 5: CTSUボタンフレームワークを生成してください。

(1): “CTSU Thread Stacks”ウィンドウで、‘+’アイコンをクリックして“Cap Touch Button Framework on sf_touch_ctsu_slider”を追加します。

(2): ボックスの“g_sf_touch_button0 Cap Touch Button Framework on sf_touch_ctsu_button”をクリックし、Number of Buttonsを2に変更します。

(3): ピンクのボックスをクリックし、Use→g_sf_touch_ctsu0 Cap Touch Framework on sf_touch_ctsuを選択してください。

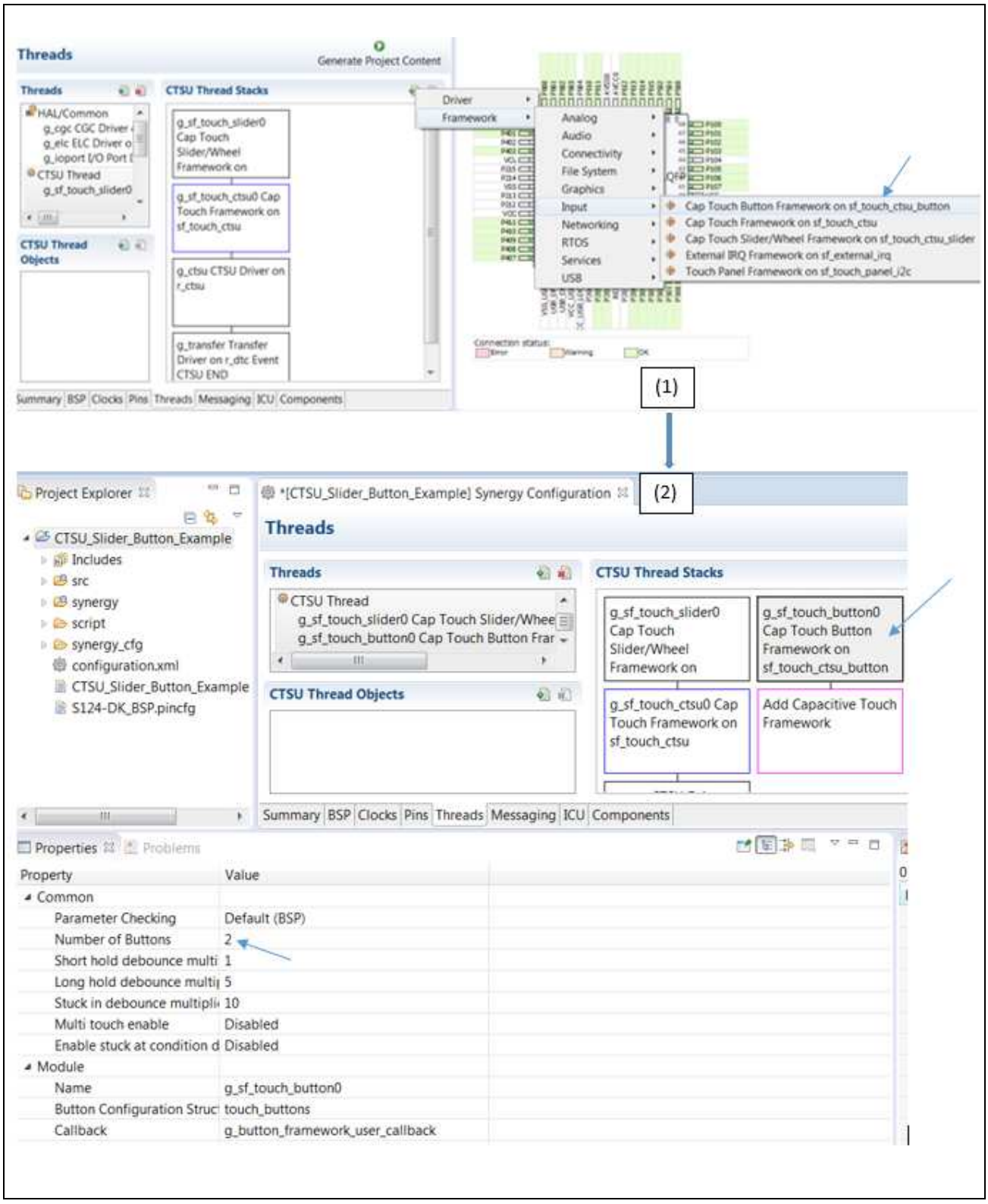

図 10　CTSUボタンフレームワークの生成

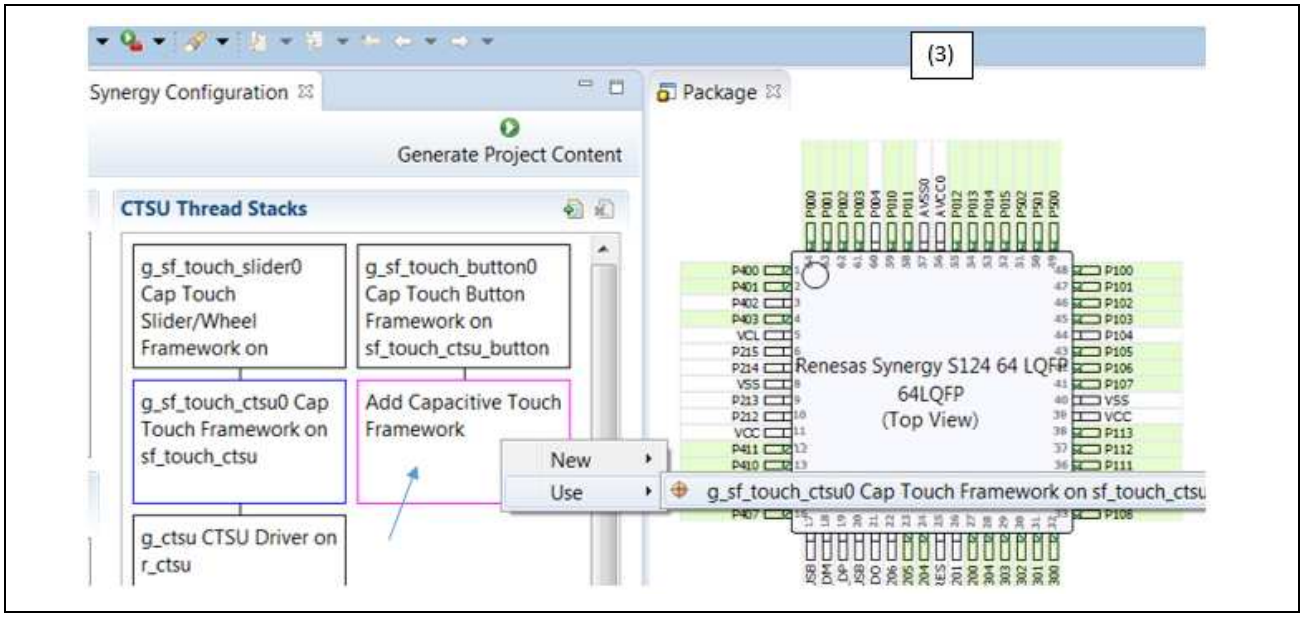

図 11　CTSUボタンフレームワークの構成

ステップ 6: この時、CTSUスレッドスタックパネルの内容は、図 12のようになっている必要があります。このステップでは、構成の変更はありません。

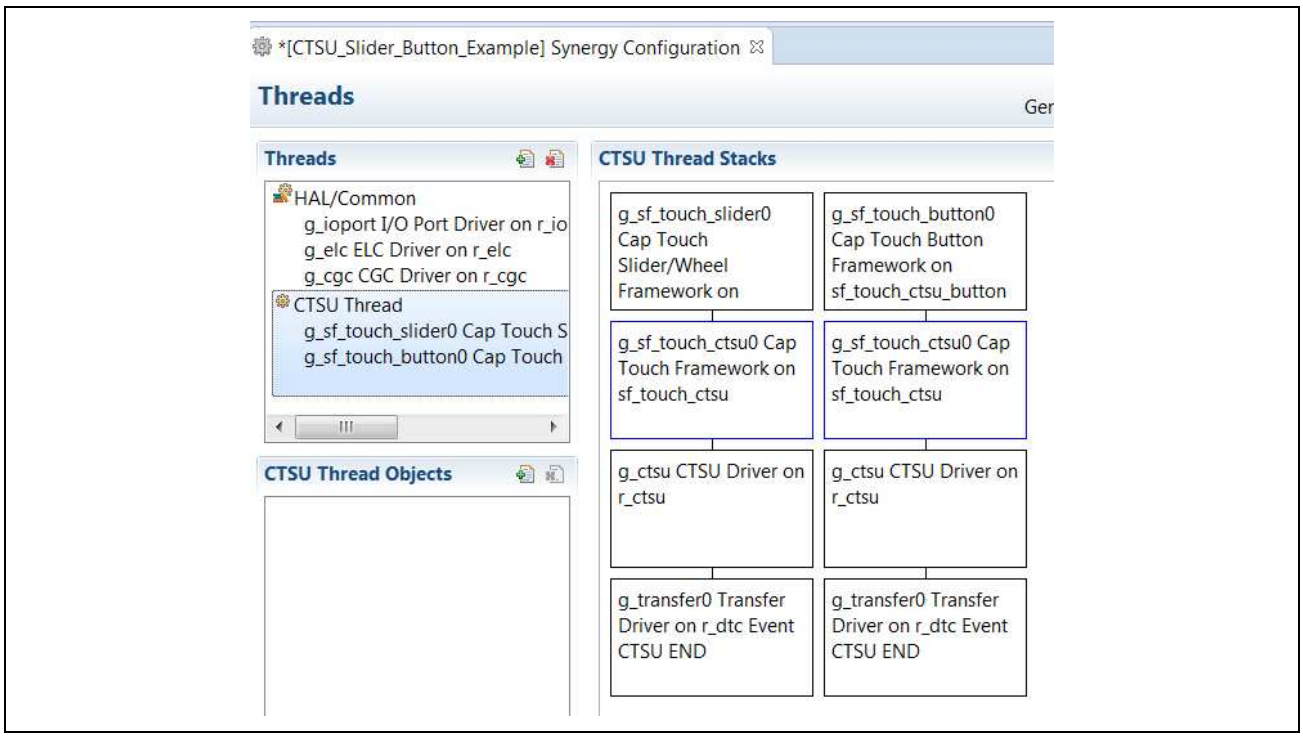

図 12　CTSUフレームワークの概要

ステップ 7: 図 13に示すように、¥CTSU_Slider_Button_Example¥src¥captouch_configsフォルダのインポートした機能サンプルプロジェクトから、新規作成したプロジェクトにcaptouch_configsフォルダを上書きしてください。

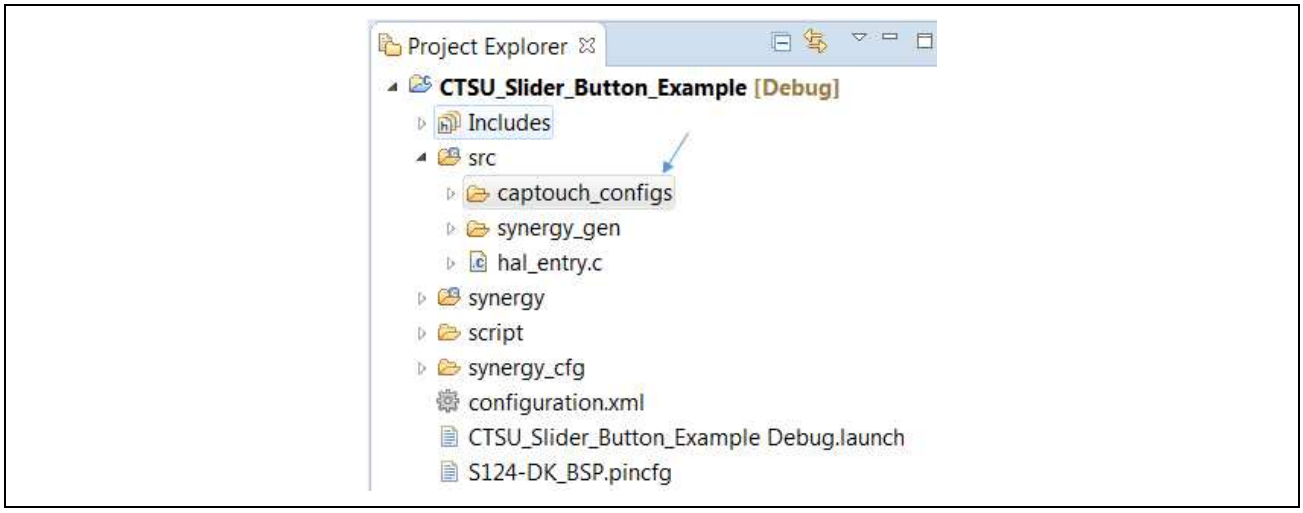

図 13 CTSU構成ファイルの上書き

ステップ 8: 新規コールバック機能を作成してください。

図 14に示すように、このステップで、サンプルプロジェクト ¥CTSU_Slider_Button_Example¥src¥から、新規作成したプロジェクトにctsu_thread_entry.c を上書きできます。

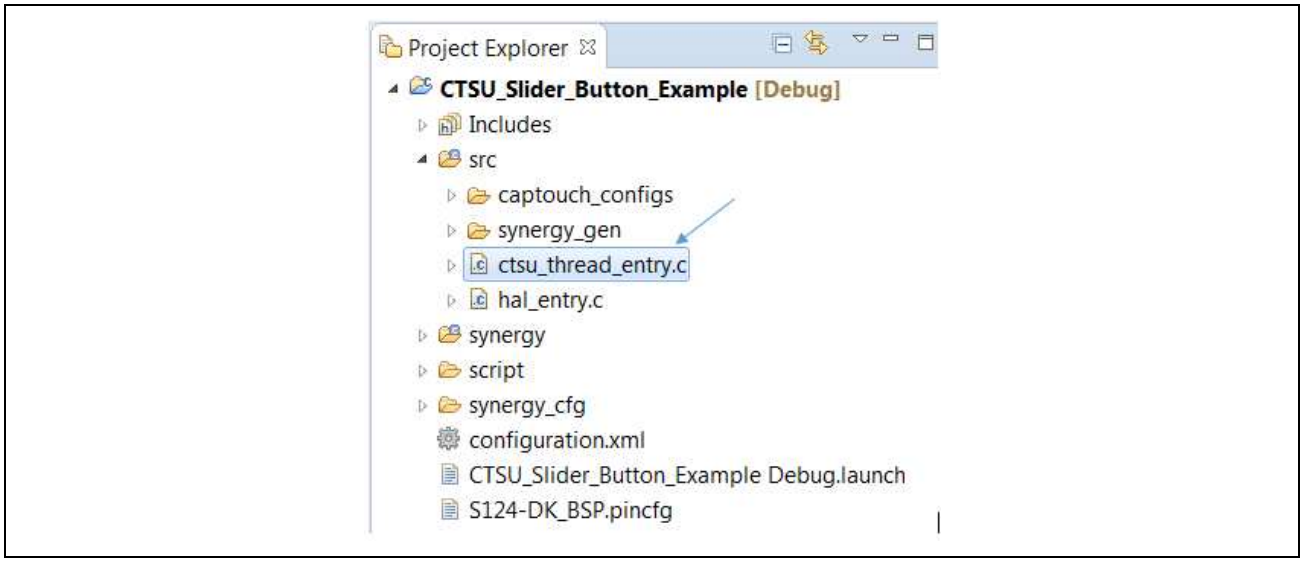

図 14 コールバック機能例の上書き

ステップ 9: プロジェクトを実行してください。

Synergyコンフィギュレータで“Generate Project Content”をクリックし、それからプロジェクトをコンパイルしてください。エラーなしでコンパイルされる必要があります。

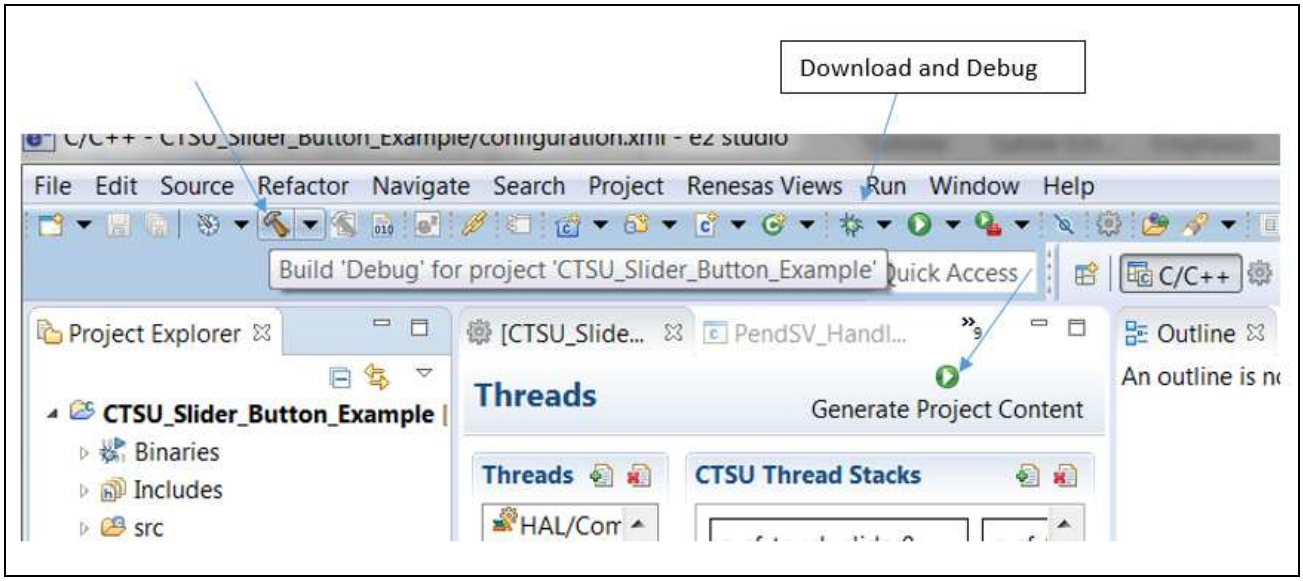

図 15　プロジェクトの生成とプロジェクトのビルド

プロジェクトをダウンロードし、プロジェクトを実行します。タッチスライダの白線の細い方（左側、－印）から太い方（右側、＋印）に指をスライドさせると、3つのLEDが暗い状態から明るい状態（太い方まで来たとき）に変化します。このサンプルアプリケーションプログラムで、LED2（橙色）はボタン1の押下イベントでトグル動作し、LED3（緑色）は、ボタン2の押下イベントでトグル動作します。

4.　まとめ

SSP CTSUフレームワークと e2studio コンフィギュレータを使って、Renesas開発キット上のスライダアプリケーションを簡単にビルドできます。SSPパッケージには、CTSUデータ処理機能とコールバックスキームが組み込まれ、迅速なスライダ・ボタンアプリケーションプログラムの開発を可能にしています。

ホームページとサポート窓口

サポート　https://synergygallery.renesas.com/support
https://synergygallery.renesas.com/support

技術的な質問の窓口の詳細

米国:　https://renesas.zendesk.com/anonymous_requests/new

ヨーロッパ:　http://www.renesas.eu/support/index.jsp
http://www.renesas.eu/support/index.jsp

日本:　http://japan.renesas.com/contact/index.jsp
http://japan.renesas.com/contact/index.jsp

改訂履歴

		説明	
Rev.	日付	ページ	まとめ
1.00	2016.08.03	-	初版

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. 未使用端子の処理
   【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。
2. 電源投入時の処置
   【注意】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. リザーブアドレス（予約領域）のアクセス禁止
   【注意】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、将来の機能拡張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。
4. クロックについて
   【注意】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。
5. 製品間の相違について
   【注意】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、内部ROM、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。お客様の機器・システムの設計において、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には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの責任を負うものではありません。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. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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。
5. 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）、交通用信号機器、
   防災・防犯装置、各種安全装置等
   当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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。
7. 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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、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。
8. 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。お客様が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。
9. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 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。

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その総株主の議決権の過半数を直接または間接に保有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。

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：http://japan.renesas.com/contact/