

Renesas Synergy™ Platform

R20AN0446JU0100

Synergy S124 および S3A7 MCU を使用した相互容量式 タッチソフトウェアのアプリケーション設計

Rev.1.00

2017.05.04

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参照ください。

要旨

このアプリケーションノートは、Synergy MCU と Synergy ソフトウェアパッケージ (SSP) を使用して、相互容量式タッチアプリケーションを作成するためのガイドラインです。

対象ユーザは、すでに e²studio もしくは Renesas Synergy™用 IAR 組込みワークベンチ® (IAR EW for Synergy) を使用しているアプリケーションやシステム開発者です。このアプリケーションノートの読者として、Synergy アプリケーションを e²studio にインポートし、IAR EW for Synergy の新しいワークスペースを開き、Synergy アプリケーションのコンフィグレーター、生成、ビルド、ダウンロード、および実行の経験があることを想定しています。その経験のない新しい Synergy MCU ユーザは、このアプリケーションノートの前に、Synergy ソフトウェアパッケージ (SSP) ユーザマニュアルからブリンキープロジェクトチュートリアル (Blinky Project Tutorial) を実行することをお勧めします。このサンプルプロジェクトを実行するには約2時間かかります。

このアプリケーションノートに記載されているソフトウェアプロジェクトは、Synergy 静電容量式タッチアプリケーションのサンプルキット AE-CAP1 で作られています。このキットは5つの PCB ボードで構成されています。

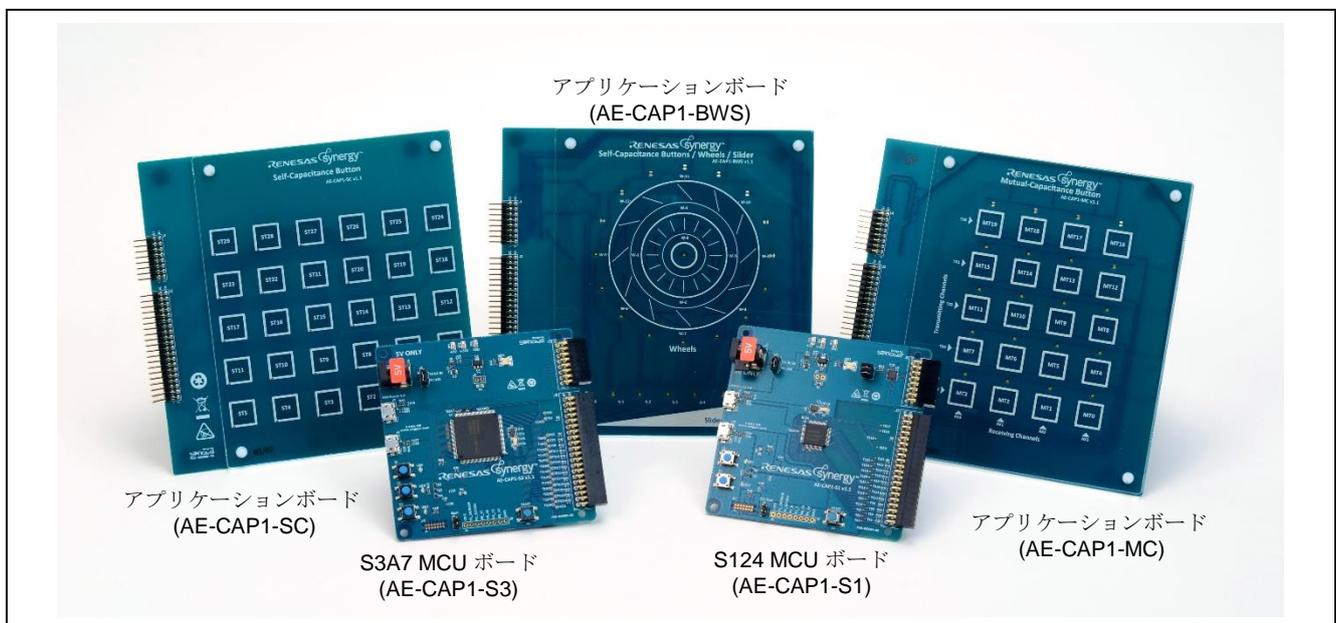


図 1 AE-CAP1 キット

このキットでは Renesas Synergy S124 および S3A7 MCU が使用されています。このアプリケーションノートに記載されている相互容量式タッチアプリケーションの設計と実装方法は、他の Synergy MCU にも適用可能です。

対象デバイス

- S124, S3A7

対象デバイスキット

- AE-CAP1 v1.1

SSP バージョン

- SSP v1.2.0

ISDE バージョン

- e2_studio v5.3.1.002
- Synergy 用 IAR EW v7.71.1.11989
- SSC_5.3.1.002

目次

1. アプリケーションの概要	4
1.1 Synergy MCU による相互容量式タッチ	4
1.2 Synergy ソフトウェアパッケージ (SSP) フレームワーク	4
1.3 AE-CAP1 および Renesas Synergy™ 用静電容量式タッチワークベンチの動作概要	5
1.3.1 AE-CAP1-S124 および AE-CAP1-S3A7 で使用される静電容量式タッチ端子	6
2. 静電容量式タッチセンシングフレームワーク	6
2.1 静電容量式タッチボタンフレームワーク	8
2.2 静電容量式タッチスライダ/ホイールフレームワーク	8
2.2.1 ホイールおよびスライダ位置の報告	8
2.3 静電容量式タッチフレームワーク	8
2.4 CTSU HAL ドライバ	9
3. アプリケーションソフトウェアプロジェクト	9
3.1 プロジェクトのビルドとダウンロード	9
3.1.1 e ² studio	9
3.1.2 IAR EW for Synergy	10
3.2 プロジェクトのスレッド	10
3.3 静電容量式タッチフレームワークとドライバ設定	11
3.3.1 S124 静電容量式タッチ設定	11
3.3.2 S3A7 静電容量式タッチシステムの設定	13
3.4 静電容量式タッチアプリケーションのユーザコールバック	15
3.4.1 ボタンコールバック	15
3.4.2 スライダ/ホイールユーザコールバック	16
3.4.3 静電容量式タッチアプリケーションの自動エラーチェック	18
3.5 Synergy 用 CTW との USB 通信	18

4.	アプリケーションプロジェクトの操作	19
4.1	プロジェクト AE_CAP1_S124_MT_App.....	19
4.1.1	プロジェクト AE_CAP1_S124_MT_App の操作	20
4.2	プロジェクト AE_CAP1_S124_MT_UART_Monitor_App.....	20
4.2.1	プロジェクト AE_CAP1_S124_MT_UART_Monitoring アプリケーションの操作	21
4.3	プロジェクト AE_CAP1_S3_MT_App.....	23
4.3.1	プロジェクト AE_CAP1_S3A7_MT_App の操作	23
5.	付録25	
5.1	CDC_ACM_Template_Win7_64bit.inf:.....	25
5.2	CDC_ACM_Template_Win7_32bit.inf:.....	26
6.	まとめ、および関連資料	28

改訂記録1

1. アプリケーションの概要

1.1 Synergy MCU による相互容量式タッチ

相互静電容量式センシングは、誘電体媒体によって分離された2つの電極が電荷を保持することができるという事実を利用しています。検出される静電容量は、2つの電極（送信側と受信側）間の静電容量です。送信ノードを変調することにより、受信ノードで検出される電界が誘導されます。ユーザが指を使用すると、ユーザの指が電界を妨害または吸収するため、受信ノードで検出されるエネルギーが減少します。相互静電容量方式は、チャンネル数を最適化するためにマトリクス法に適しています。

相互静電容量式タッチ設計のハードウェア設計の詳細については、静電容量式タッチハードウェアの設計とレイアウトガイド（R12AN0060EU0100）を参照してください。

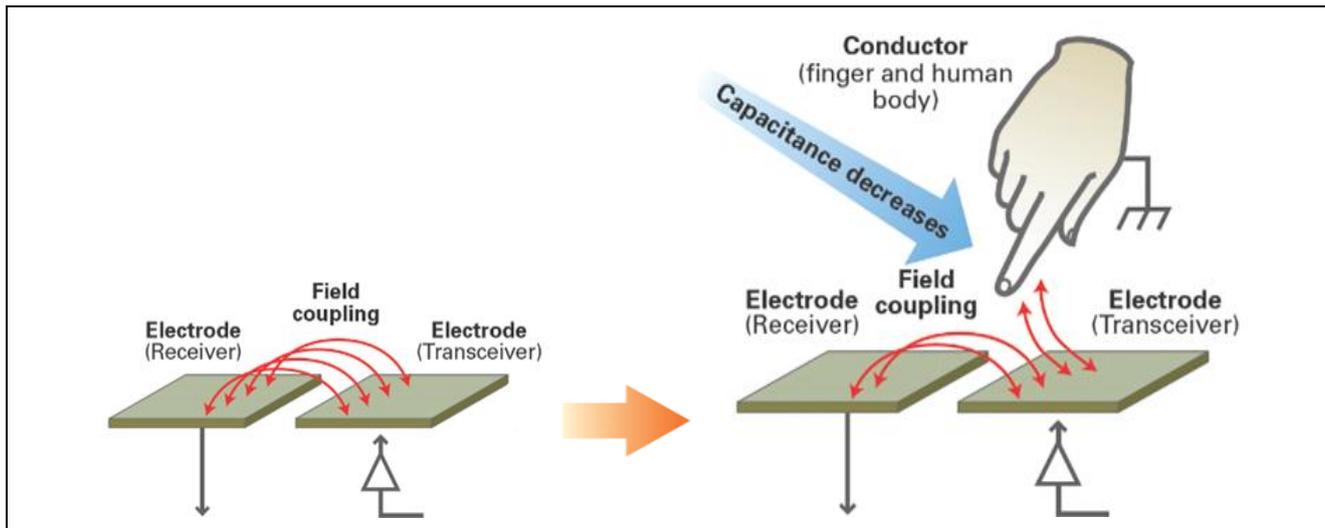


図 2 相互静電容量式センシング

S124 および S3A7 の静電容量式タッチセンシングユニット（CTSU）、RX130 および RX113 などの他の MCU はルネサスタッチ Gen 2 IP を使用しています。静電容量式タッチセンシングユニット（CTSU）周辺装置の内部動作の詳細については、S124 および S3A7 のユーザマニュアルを参照してください。

1.2 Synergy ソフトウェアパッケージフレームワーク

図 3 に、このアプリケーションプロジェクトで使用される主要なソフトウェアシステムを示します。使用する SSP のすべての部品を青枠内に示します。

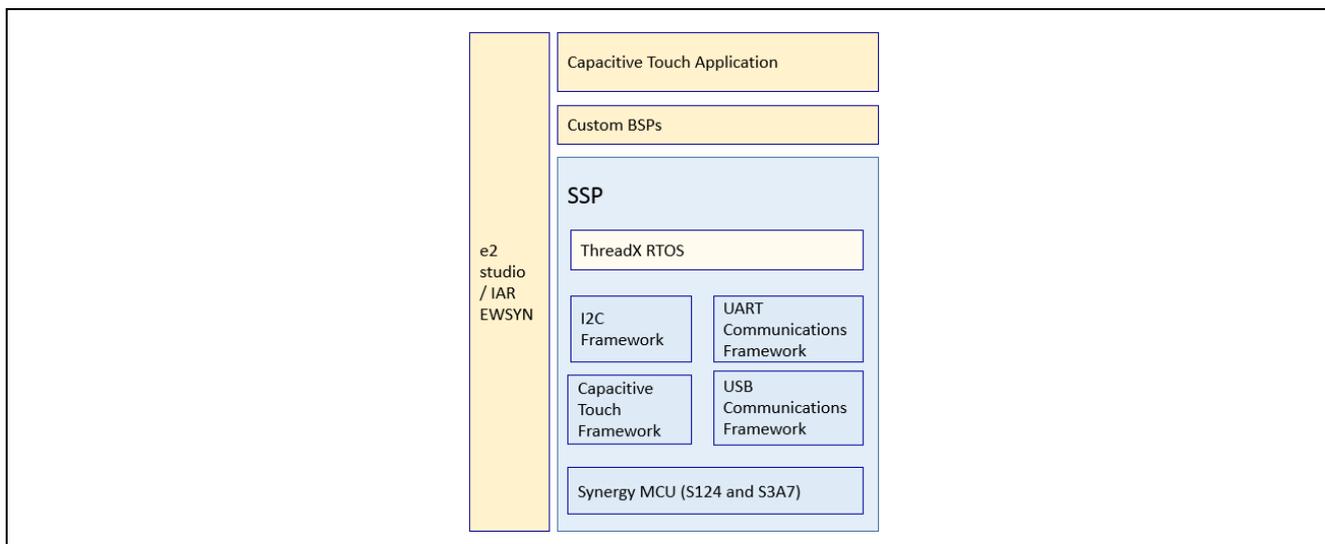


図 3 SSP モジュール

1.3 AE-CAP1 および Renesas Synergy™用静電容量式タッチワークベンチの動作概要

付属の静電容量式タッチソフトウェアプロジェクトの高レベル動作を理解するために、図4および図5のブロック図を参照してください。

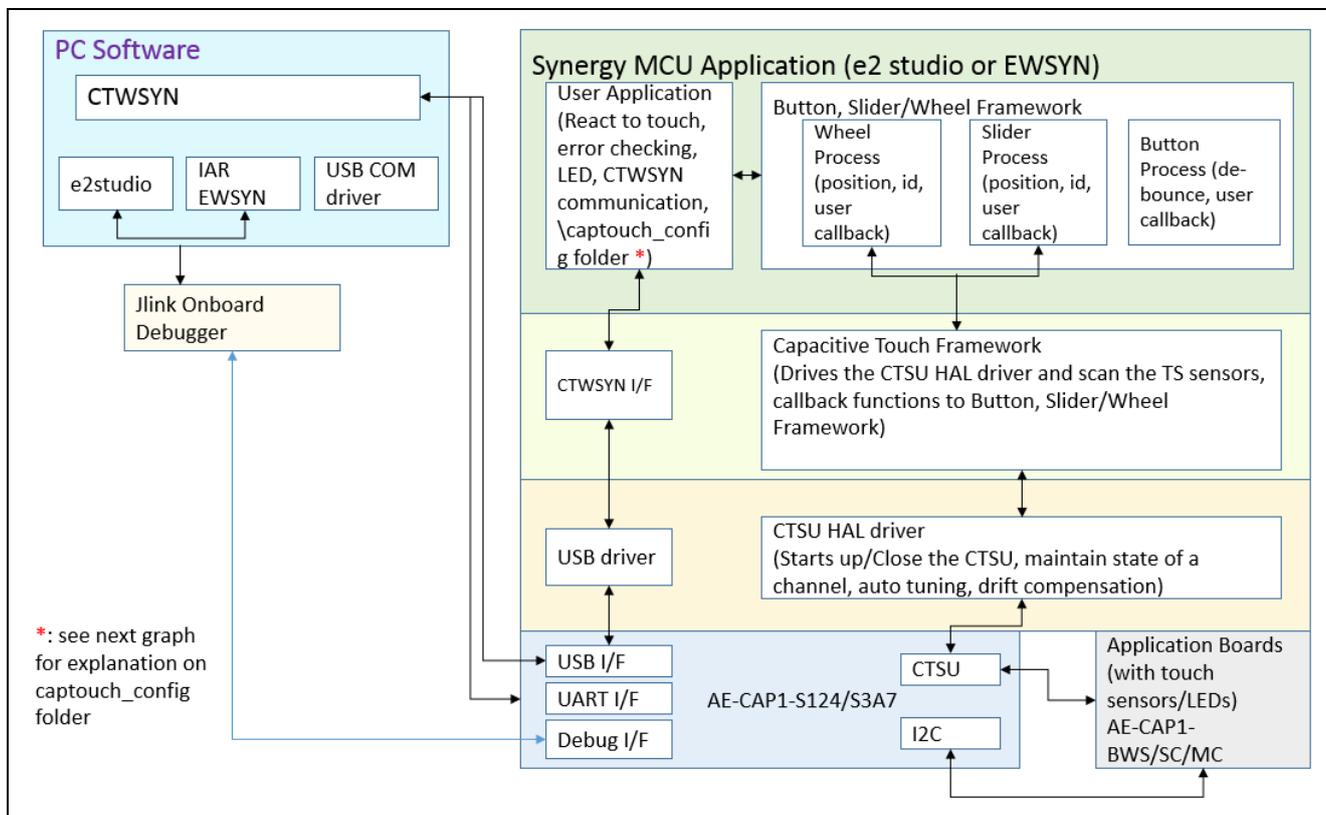


図4 システムブロック図

提供されるソフトウェアプロジェクトのいくつかは、UARTまたはUSBを介して Renesas Synergy™用の静電容量式タッチワークベンチ（Synergy用CTW）と通信する機能を有しています。この機能を有するプロジェクトについては、表1を参照してください。

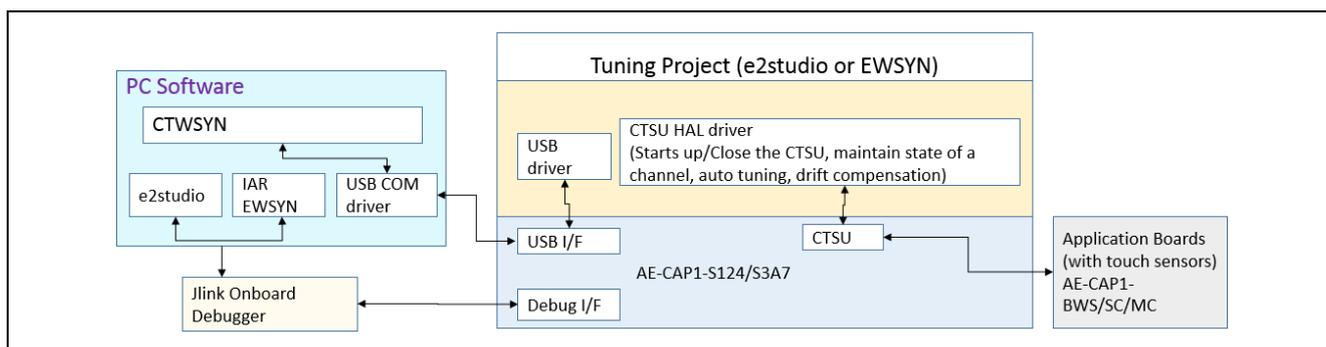


図5 調整プロセス

- 各静電容量式タッチアプリケーションボードに対し、e²studioまたはIAR EW for Synergyのチューニングプロジェクトが必要です。
- このチューニングプロジェクトは、Synergy用CTWと通信して、アプリケーションプロジェクトで使用するされる\captouch_configフォルダを生成します。
- チューニングプロセスは、タッチ閾値を決定し、静電容量式タッチアプリケーションの感度を最大にします。結果のパラメータは\captouch_configフォルダに保存されます。

- チューニングとアプリケーションプロジェクトの両方で、同一の PCLKB 周波数を使用する必要があります。

Synergy 静電容量式タッチシステムのチューニングプロセスを理解するには、静電容量式タッチチューニング (Synergy アプリケーションプロジェクト) (R20AN0448EU0100) を参照してください。このアプリケーションノートでは、チューニングが正常に実行された後のアプリケーションプロジェクトの作成に焦点を当てています。

1.3.1 AE-CAP1-S124 および AE-CAP1-S3A7 で使用される静電容量式タッチ端子

静電容量式タッチの目的のために、AE-CAP1 のハードウェアおよびソフトウェアプロジェクトは、S124 用の静電容量式タッチ端子 28 本と S3A7 用の静電容量式タッチ端子 30 本を実装しています。MCU 静電容量式タッチセンシングサポートの詳細については、各 MCU のハードウェアユーザーマニュアルを参照してください。

表 1 使用端子

S124 端子名	S3A7 端子名	機能
TSCAP	TSCAP	CTSU 電力安定化
TS00-TS02, TS04-TS09, TS12-TS28, TS30-TS31	TS00-TS01, TS04-TS22, TS26-TS27, TS29-TS35	静電容量式タッチセンシング
TS03, TS10, TS11 には他の機能が割り当てられています。	TS03 には他の機能が割り当てられています。	図表参照
TS29 非機能中	TS02, TS23, TS24, TS25 非機能中	ハードウェアユーザーマニュアル参照

相互容量式アプリケーションボードでは、5 個×4 個のマトリックスとして 9 個のタッチセンサーのみが実装され、20 個の相互容量性タッチボタンが作成されます。

2. 静電容量式タッチセンシングフレームワーク

Renesas Synergy™プラットフォームの一部である Renesas Synergy™ソフトウェアパッケージ (SSP) は、組み込みシステム向けに使いやすく、拡張可能で高品質なソフトウェアを提供するための完全統合ソフトウェアパッケージです。

図 6 は、静電容量式タッチセンシングに関連するソフトウェアインターフェースの構造を示しています。

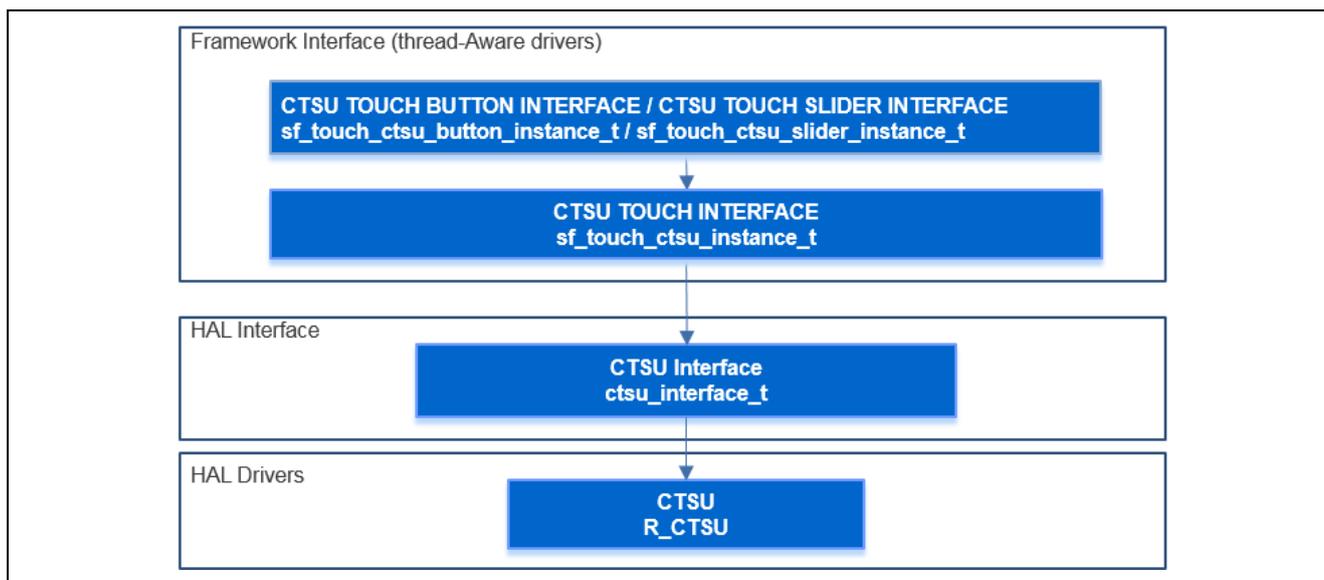


図 6 SSP CTSU フレームワーク

3つの SSP フレームワークが、このソフトウェアプロジェクトで使用されています。

- 静電容量式タッチボタンフレームワーク
- 静電容量式タッチスライダ/ホイールフレームワーク
- 静電容量式タッチフレームワーク

2.1 静電容量式タッチボタンフレームワーク

- 静電容量式タッチボタンフレームワークにより、ユーザが、ボタン数、デバウンスセッティング、およびコールバック機能を設定して、タッチイベントへ対処することが可能になります。
- 静電容量式タッチボタンフレームワークは、システムに存在するすべてのボタンの CTSU データを解釈するために使用されます。また、静電容量式タッチフレームワークのレイヤを初期化します。
- 静電容量式タッチボタンフレームワークは、処理データが利用可能になるたびに呼び出される静電容量式タッチフレームワークレイヤにコールバックを登録します。静電容量式タッチボタンフレームワークは、この処理データを使用して、デバウンスの処理を行い、各ボタンに対してどのようなイベント（プレス、リリースなど）を有効にするかを決定します。
- フレームワークは、各ボタンのユーザコールバックを、ボタン設定テーブルにある順に呼び出します。
- フレームワークレイヤは、静電容量式タッチフレームワークレイヤを使用してボタンインターフェースを実装します。このボタンフレームワークを使用して、ユーザは Synergy 用 CTW から生成された設定構造を使用して複数のボタンを構成して使用することができます。各ボタンの機能により、ボタン ID とイベントタイプを示す引数を持つコールバックが生成されます。

2.2 静電容量式タッチスライダ/ホイールフレームワーク

- 静電容量式タッチスライダフレームワークは、ThreadX RTOS を使用する静電容量式タッチスライダおよびホイールアプリケーション用に `sf_touch_ctsu_slider` に実装された汎用 API です。
- 静電容量式タッチスライダフレームワークには、Synergy マイクロコントローラハードウェア上で動作する静電容量式タッチフレームワークと CTSU ドライバモジュールが必要です。静電容量式タッチスライダフレームワークは、システムによって初期化されたすべてのスライダ構成の CTSU データを解釈するために使用されます。また、静電容量式タッチフレームワークのレイヤを初期化します。
- 静電容量式タッチスライダフレームワークは、処理されたデータが利用可能になるたびに呼び出される静電容量式タッチフレームワークレイヤにコールバックを登録します。
- スライダフレームワークは、このデータ（生の値）を使用して、タッチまたはリリースが発生したかどうかを判断し、発生した場合は発生した場所を判断します。状態の変化がある場合、フレームワークは、スライダ/ホイール構成テーブルに存在する順序で、各スライダ/ホイールのコールバックをイベントと位置で呼び出します。
- スライダフレームワークは、タッチイベントとリリースイベントの間で更新レート (`sf_touch_ctsu configuration update_hz`) でコールバックを実行します。

2.2.1 ホイールおよびスライダ位置の報告

SSP から読み取られるスライダ位置は、スライダ静電容量式タッチコンポーネント上に 0 から 500 まで、直線的に分散しています。ホイールの形状はスライダの概念と似ていますが、直線的な解釈ではなく、0~360 度のレイアウトを持つ円とみなされます。

2.3 静電容量式タッチフレームワーク

- 静電容量式タッチフレームワークは、CTSU HAL ドライバの駆動に使用する ThreadX 対応 CTSU インターフェースです。これは、CTSU ハードウェアを実行し、スキャンの結果を読み取るために使用します。
- 静電容量式タッチフレームワークインターフェースは、静電容量式タッチパネルのハードウェアスキャンを駆動しパネルを定期的に更新するプライベートスレッドを作成します。
- 静電容量式タッチフレームワークは、HAL レイヤの CTSU ドライバを使用してスキャンした結果を読み取ります。スキャンが完了すると、アプリケーションレイヤによって登録されたコールバックが呼び出されます。複数の上位レイヤがこのフレームワーク（たとえば、ボタン、スライダ、ホイール）を使用している場合、このレイヤは、このレイヤで初期化された順にそれらのレイヤの到達範囲のコールバックを呼び出します。

2.4 CTSU HAL ドライバ

- CTSUドライバを使用して、CTSU周辺装置を初期化して、任意の設定（およびイネーブルした）チャンネルの容量変化を検出し、必要なフィルタリングを実行し、ボタン、ホイール、およびスライダなどの上位レベルのウィジェット層で使用できるさまざまなデータを生成します。
- これらのレイヤで必要とされるさまざまなタイプのデータをサポートするため、実装では上位レベルのレイヤが必要に応じてさまざまなタイプの処理データを読み取ることができるRead()関数を提供します。
- ドライバはまた、各スキャンの完了時、および新しい処理データが利用可能な際にコールバックを提供します。これらのコールバックは上位レイヤによるデータの読み取りに使用されます。
- CTSUドライバを使用すると、相互および自己容量を含むサポートされているすべての動作モードでCTSUチャンネルを設定することができます。
- ドライバは設定チャンネルをスキャンし、DTCを使用してデータを移動し、フィルタリング、ドリフト補正、オートチューニングを実行し、各繰り返しが完了するとコールバックを介してユーザーに通知します。
- ドライバは一度に1つの設定のみをサポートすることができますが、ユーザーはアプリケーションの必要に応じて複数のチャンネル設定でドライバを開くことができます。

3. アプリケーションソフトウェアプロジェクト

表 2に、このアプリケーションプロジェクトに含まれる付属のソフトウェアプロジェクトの概要をまとめます。すべてのプロジェクトは、以下に記されていない限り、e²studio および Synergy 用 IAR EW for Synergy で Windows10 / Windows7 に対応しています。

表 2 アプリケーションプロジェクト概要

ソフトウェアプロジェクト	プロジェクトの説明
AE_CAP1_S3A7_MT_App	AE-CAP1-MC の S3A7 アプリケーションプロジェクト <ul style="list-style-type: none"> • 20 個の相互静電容量式タッチボタン • AE-CAP1-MT ボードの LED 表示 • USB 経由の Synergy 用 CTW との通信
AE_CAP1_S124_MT_App	AE-CAP1-MC 用 S124 アプリケーションプロジェクト <ul style="list-style-type: none"> • 20 個の相互容量性タッチボタン • AE-CAP1-MC ボードの LED 表示
AE_CAP1_S124_MT_UART_Monitor_App	AE-CAP1-MC ボード用 S124 アプリケーションプロジェクト <ul style="list-style-type: none"> • 20 個の相互静電容量式タッチボタン • AE-CAP1-S124 ボードの LED 表示 • UART 経由の Synergy 用 CTW との通信

3.1 プロジェクトのビルドとダウンロード

3.1.1 e²studio

1. 2つのBSPファイル (Renesas.s124_ae_cap1.1.2.0およびRenesas.s3a7_ae_cap1.1.2.0) をe²studioのインストールフォルダ\`<your e2_studio folder>\internal\projectgen\arm\Packs`内に配置します。
2. 完成したプロジェクトをe²studioにインポートしてプロジェクトをビルドする手順については、「Synergy プロジェクトインポートガイド」 (r11an0023eu0116-synergy-ssp-import-guide.pdf) を参照してください。付属のSelf-Capacitive Touch Software Application Design with Synergy MCUs.zip ファイルは完成したプロジェクトを提供します。

適切にインポートすると、以下のファイル構造が表示されます。

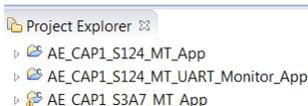


図7 e²studio におけるインポートファイル

3. サンプルプロジェクトには、デバッグプロセスに役立つ適切な起動ファイルが付属しています。

3.1.2 Synergy 用 IAR EW

1. Renesas.s124_ae_cap1.1.2.0 と Renesas.s3a7_ae_cap1.1.2.0 の2つの BSP ファイルを、Synergy SSC フォルダ \- 2. IAR EW ワークスペースを開く前に、SSP を SSC にインストールする必要があります。IAR EW ワークスペースを開いたら、**RenesasSynergy> Settings** に移動します。IAR EW は、SSC v5.3.1.002 がインストールされた場所と関連 SSP ライセンスファイルを表示します。

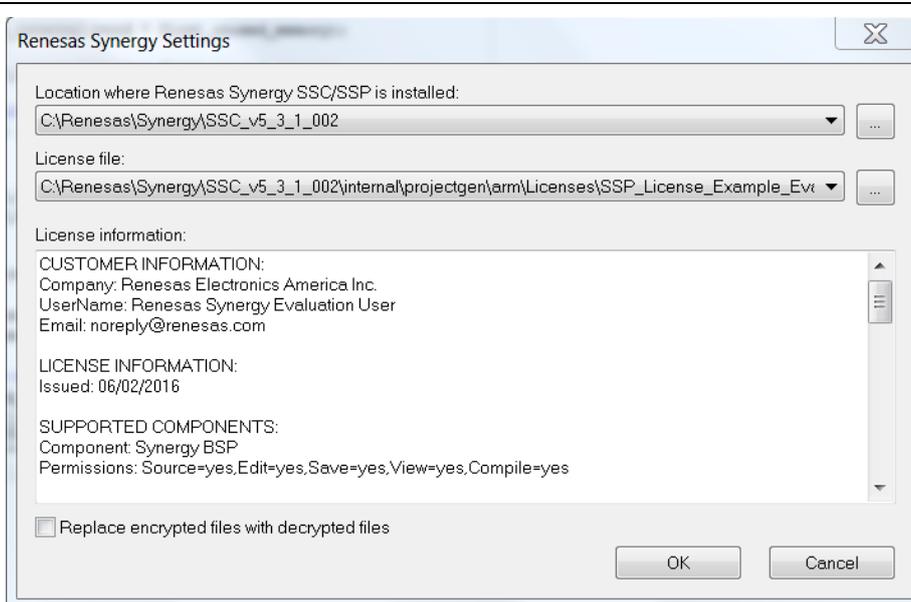


図8 SSC ロケーション

3. IAR WE で‘Project->Rebuild All’に移動します。プロジェクトをダウンロードしてデバッグするには、緑色の矢印をクリックします。



図9 ダウンロードとデバッグ

3.2 プロジェクトのスレッド

すべてのサンプルプロジェクトには、静電容量式タッチアプリケーション用の ctsu スレッドが含まれています。S3A7 用の USB 通信スレッドと S124 用の UART スレッドは、Synergy 用 CTW と通信するために使用されます。

3.3 静電容量式タッチフレームワークとドライバ設定

3.3.1 S124 静電容量式タッチ設定

(1) 静電容量式タッチボタンフレームワークの設定

図 10 の画面キャプチャに、S124 相互静電容量式ソフトウェアプロジェクトの静電容量式タッチボタンフレームワークの共通パラメータ設定の一部を示します。

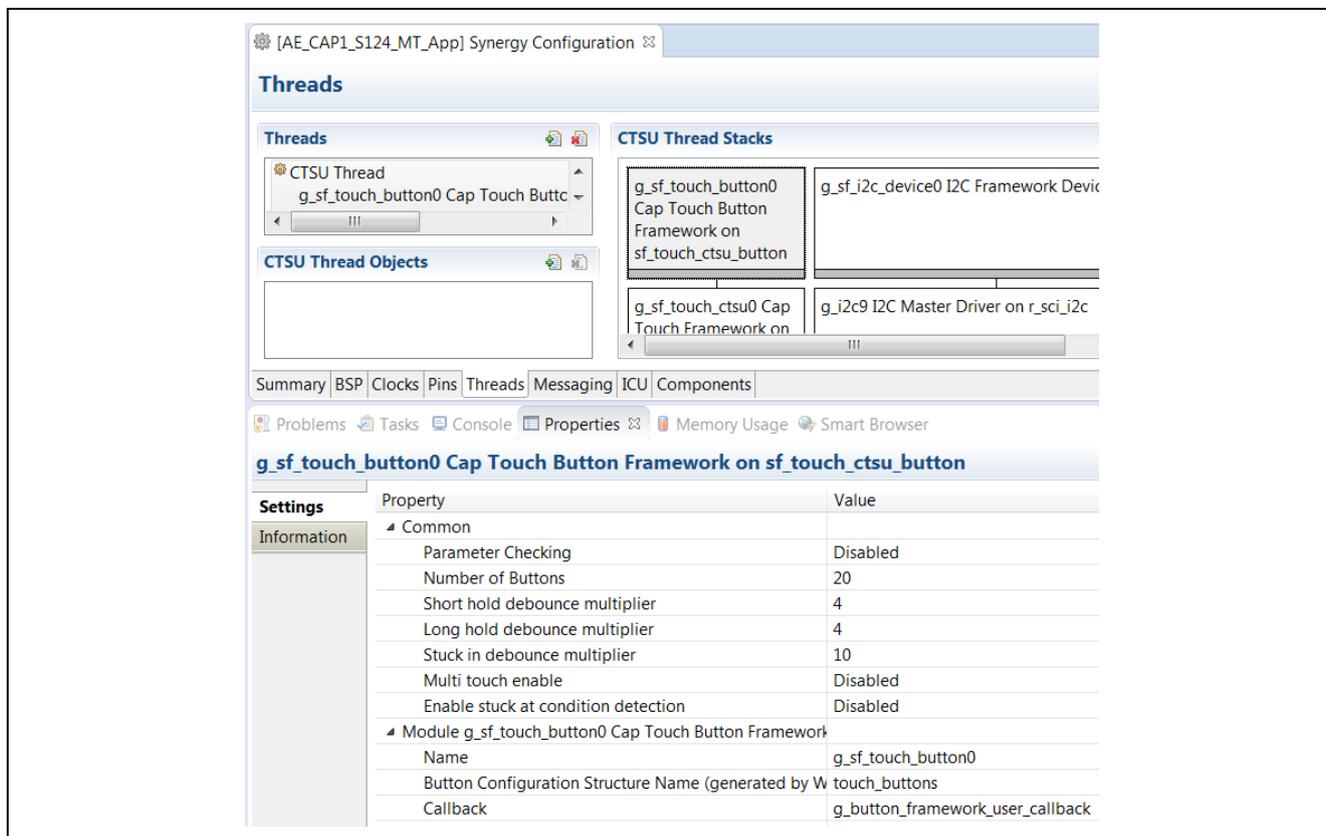


図 10 相互静電容量式アプリケーションにおける S124 の静電容量式タッチボタンフレームワーク設定

- AE-CAP1-MC アプリケーションボードは 5 行 4 列のマトリクス設定を使用し、各交差点は相互静電容量式のタッチボタンとして構成されています。したがって、**ボタン数**は 20 に設定されます。
- ユーザは、アプリケーションコードにコールバック関数 `g_button_framework_user_callback` を実装する必要があります。
- デバウンス乗数はアプリケーションで内部的に 7 倍になります。

(2) 静電容量式タッチスライダのフレームワーク設定

AE-CAP1 キットの最初のリリースでは、相互静電容量式スライダ/ホイールは実装されていません。

(3) 静電容量式タッチのフレームワーク設定

図 11 の画面キャプチャに、S124 相互静電容量式タッチソフトウェアプロジェクトにおける静電容量式タッチフレームワークの共通パラメータ設定の一部を示します。

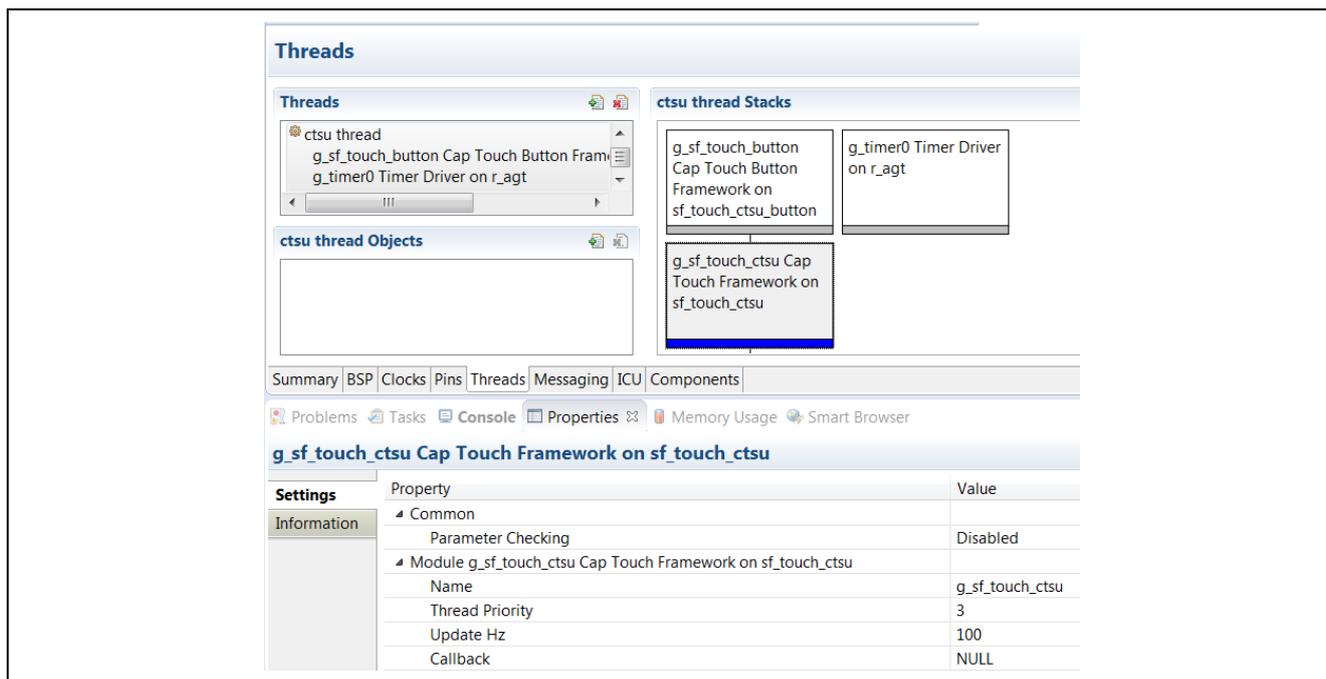


図 11 S124 および S3A7 の静電容量式タッチスライダ/ホイールフレームワークの設定

- ユーザは、ソフトウェアアプリケーションで適切なスレッド優先度を設定することができます
- **Update Hz** は、アプリケーションで 100 Hz に設定されている RTOS ティックレートより少なくする必要があります

(4) CTSU HAL ドライバの設定

図 12 の画面キャプチャに、S124 ソフトウェアプロジェクトの CTSU HAL ドライバの共通パラメータ設定の一部を示します。

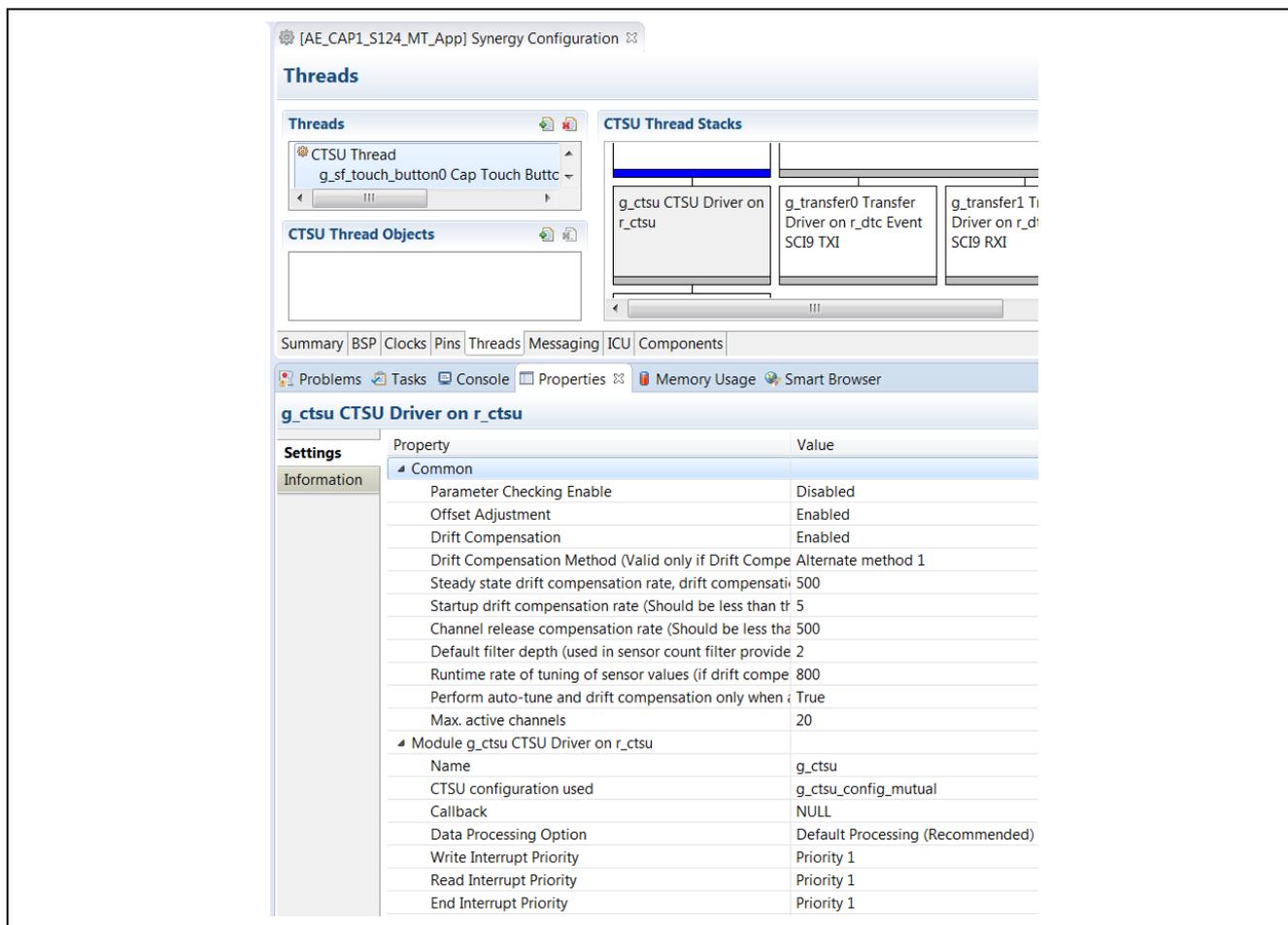


図 12 相互静電容量式アプリケーションにおける S124 の CTSU HAL ドライバ設定

- プロパティ **Max. active channels** の設定は、図 10 の **Number of Channels** と同一です。
- 使用される CTSU 設定が `g_ctsu_config_mutual` に設定されていることに注意してください。これは、相互静電容量式調整データを保持するチューニングプロセスから生成されたデータ構造の名前です
- ユーザは、アプリケーションに基づいて、**Write, Read, End Interrupt Priority** を設定できます

3.3.2 S3A7 静電容量式タッチシステムの設定

(1) 静電容量式タッチボタンフレームワークの設定:

図 13 の画面キャプチャに、S3A7 ソフトウェアプロジェクトの静電容量式タッチボタンフレームワークの共通パラメータ設定の一部を示します。

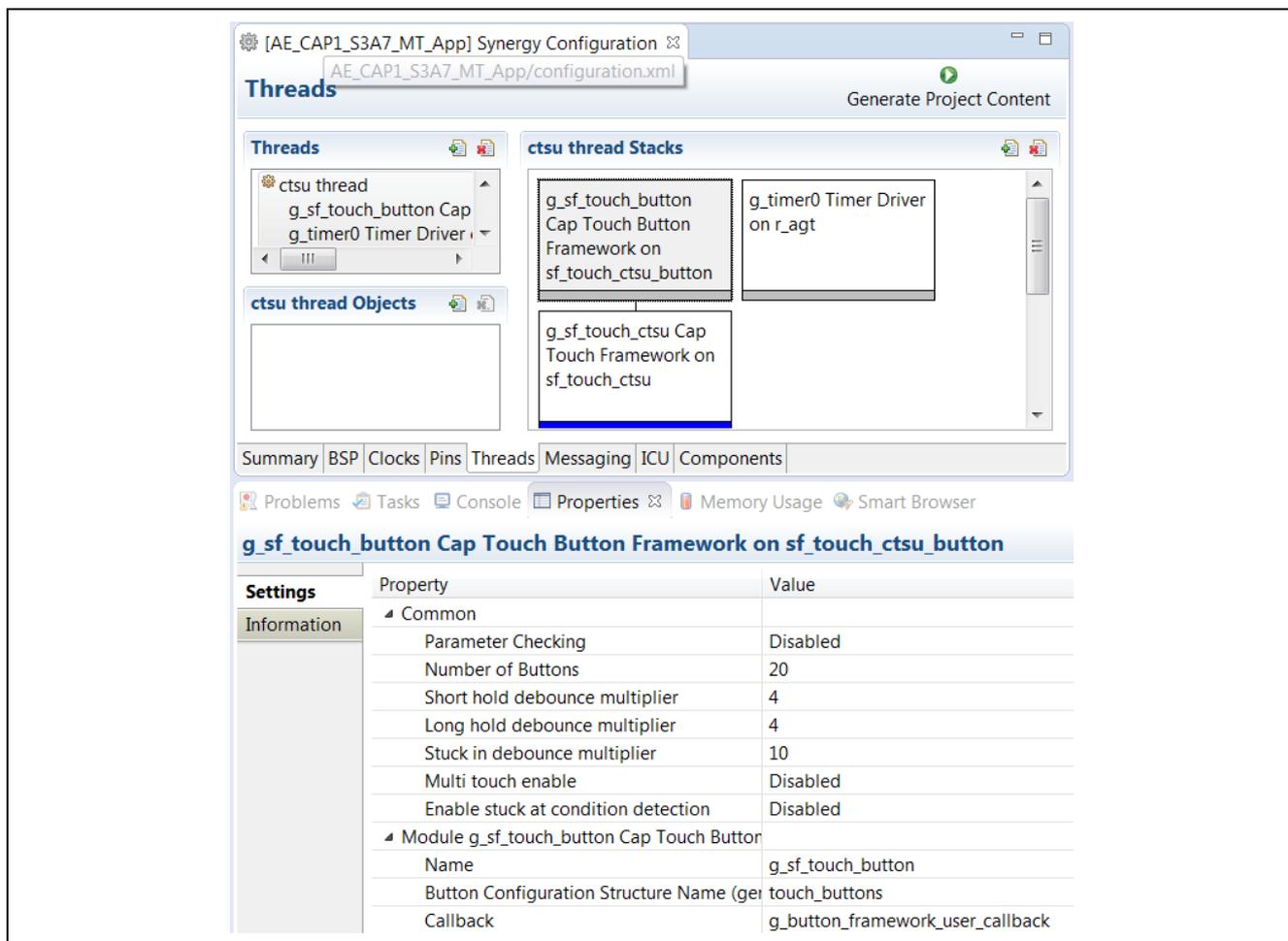


図 13 相互静電容量式タッチアプリケーションにおける S3A7 用静電容量式タッチボタンフレームワーク

- S124 と同様に、AE-CAP1-MC の相互静電容量式タッチボタンは、クロスポイントで 20 個のボタンを備えた 5 行×4 列のマトリックスとして編成されています。
- ユーザは、アプリケーションコードにコールバック関数 `g_button_framework_user_callback` を実装する必要があります。
- デバウンス乗数はアプリケーションで内部的に 7 倍になります。

(2) 静電容量式タッチスライダのフレームワーク設定

AE-CAP1 キットの最初のリリースには、相互静電容量式スライダ/ホイールの実装はありません。静電容量式タッチフレームレイヤーは、S124 と S3A7 の間の共通設定を共有します。関連設定については、図 11 を参照してください。

(3) CTSU HAL ドライバの設定

図 14 の画面キャプチャに、S3A7 相互静電容量式タッチソフトウェアプロジェクトにおける CTSU HAL ドライバ設定の共通パラメータ設定の一部を示します。

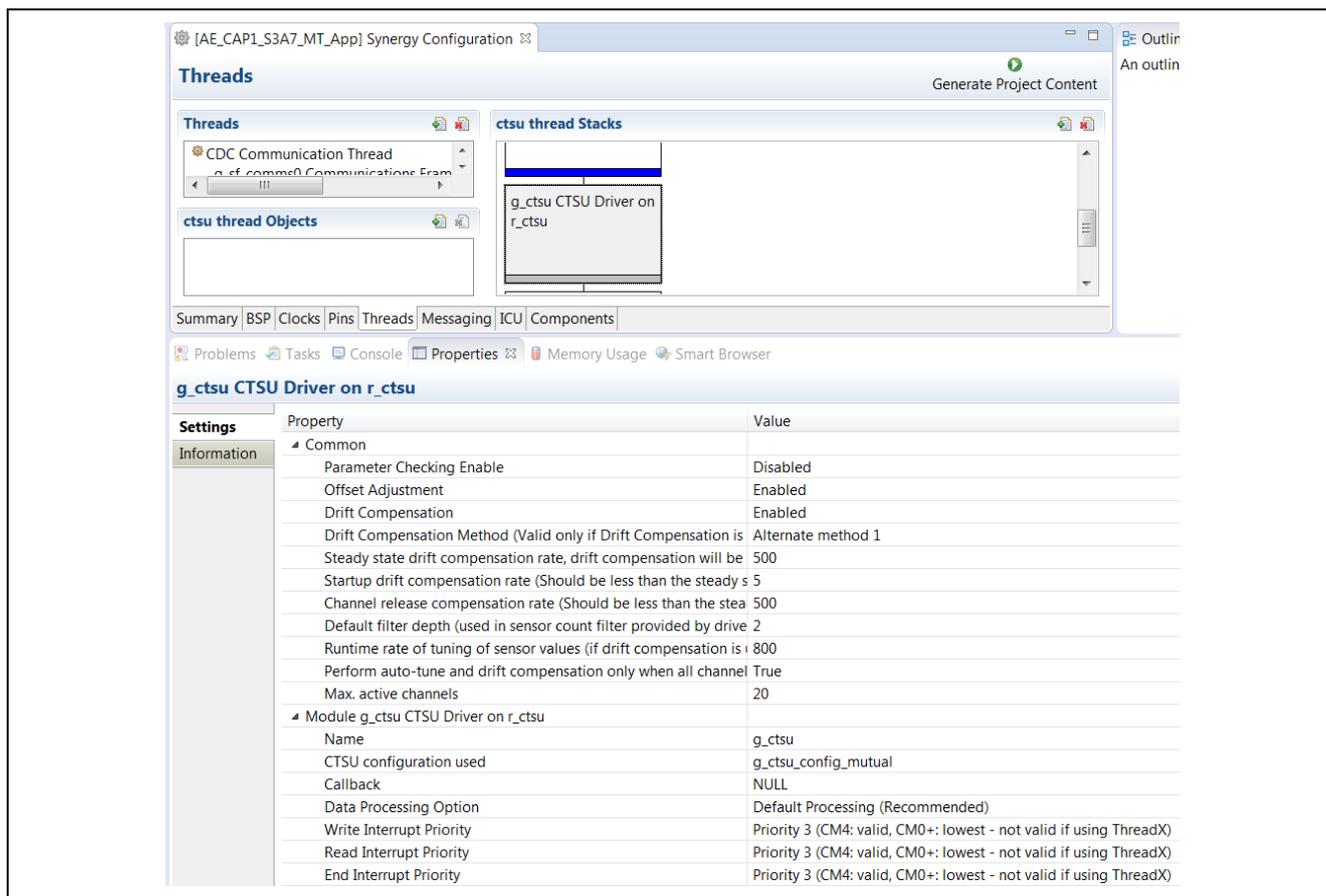


図 14 相互静容量式タッチアプリケーションにおける S3A7 用 CTSU HAL ドライバ設定

- プロパティ **Max. active channels** の設定は、図 13 の **Number of Channels** の設定と一致します
- 使用する CTSU 設定が `g_ctsu_config_mutual` に設定されていることに注意してください。これは、相互静容量式チューニングデータを保持するチューニングプロセスから生成されたデータ構造の名前です
- ユーザは、アプリケーションに基づいて、**Write, Read, End Interrupt Priority** を設定できます

3.4 静容量式タッチアプリケーションのユーザコールバック

3.4.1 ボタンコールバック

```
void g_button_framework_user_callback (sf_touch_ctsu_button_callback_args_t * p_args)
{
    switch(p_args->id)
    {
        case TS11:
            CB_Self_Button_TS11(p_args);
            break;
    }
}
```

図 15 ボタンフレームワークのユーザコールバック

ボタンコールバックから生成される可能性のあるイベントを表示するには、図 16 を参照してください。

```
void CB_Self_Button_TS11 (sf_touch_ctsu_button_callback_args_t * p_args);  
void CB_Self_Button_TS11 (sf_touch_ctsu_button_callback_args_t * p_args)  
{  
    switch(p_args->event)  
    {  
        case TOUCH_BUTTON_STATE_INITIAL:  
            break;  
        case TOUCH_BUTTON_STATE_PRESSED:  
            SetLED (3,3); /* W0 */  
            break;|  
        case TOUCH_BUTTON_STATE_LONG_HOLD:  
            break;  
        case TOUCH_BUTTON_STATE_SHORT_HOLD:  
            break;  
        case TOUCH_BUTTON_STATE_STUCK:  
            break;  
        case TOUCH_BUTTON_STATE_RELEASED:  
            offLEDs ();  
            break;  
        case TOUCH_BUTTON_STATE_CLOSING:  
            break;  
        case TOUCH_BUTTON_STATE_DISABLED:  
            break;  
        case TOUCH_BUTTON_STATE_MULTI_TOUCH:  
            break;  
        default:  
            break;  
    }  
}
```

アプリケーションボー
ド上の LED 表示

図 16 ボタンイベント処理

3.4.2 スライダー/ホイールユーザコールバック

```
void g_slider_framework_user_callback(sf_touch_ctsu_slider_callback_args_t * p_args)  
{  
    switch(p_args->id)  
    {  
        case SLIDER0:  
            CB_Self_Slider_0(p_args);  
            break;  
        case WHEEL0:  
            CB_Self_Wheel_0(p_args);  
            break;  
        case WHEEL1:  
            CB_Self_Wheel_1(p_args);  
            break;|  
        default:  
            break;  
    }  
}
```

図 17 スライダーユーザコールバック

- スライダーユーザコールバックはスライダーコールバックとホイールコールバックの両方を処理します
- ボタンコールバックから生成される可能性のあるイベントを見るには、図 18 を参照してください

```
void CB_Self_Slider_0(sf_touch_ctsu_slider_callback_args_t * p_args);  
void CB_Self_Slider_0(sf_touch_ctsu_slider_callback_args_t * p_args)  
{  
    extern slider_info_t g_sliderInfo[];  
  
    switch(p_args->event)  
    {  
        case SF_TOUCH_CTSU_SLIDER_STATE_INITIALIZED:  
  
            break;  
        case SF_TOUCH_CTSU_SLIDER_STATE_TOUCHED:  
        case SF_TOUCH_CTSU_SLIDER_STATE_TOUCHED |  
SF_TOUCH_CTSU_SLIDER_STATE_MULTI_TOUCH:  
            if (0 != (p_args->event &  
(uint32_t)SF_TOUCH_CTSU_SLIDER_STATE_MULTI_TOUCH))  
            {  
            }  
            else  
            {  
                g_sliderInfo[0].value = (uint16_t)p_args->current_position;  
                slider0_led_response(p_args);  
            }  
            break;  
        case SF_TOUCH_CTSU_SLIDER_STATE_HELD:  
        case SF_TOUCH_CTSU_SLIDER_STATE_HELD |  
SF_TOUCH_CTSU_SLIDER_STATE_MULTI_TOUCH:  
            if (0 != (p_args->event &  
(uint32_t)SF_TOUCH_CTSU_SLIDER_STATE_MULTI_TOUCH))  
            {  
            }  
            else  
            {  
                g_sliderInfo[0].value = (uint16_t)p_args->current_position;  
                slider0_led_response(p_args);  
            }  
            break;  
        case SF_TOUCH_CTSU_SLIDER_STATE_RELEASED:  
            offLEDs ();  
  
            break;  
        case SF_TOUCH_CTSU_SLIDER_STATE_CLOSED:  
  
            break;  
        case SF_TOUCH_CTSU_SLIDER_STATE_DISABLED:  
  
            break;  
        default:  
            break;  
    }  
}
```

CTWSYN との通信

図 18 スライダイベント処理

3.4.3 静電容量式タッチアプリケーションの自動エラーチェック

```
void g_sf_touch_button_err_callback(void * p_instance, void * p_data)
{
    ssp_err_t ssp_err_g_sf_touch_button;
    SSP_PARAMETER_NOT_USED (p_instance);
    ssp_err_g_sf_touch_button = *((ssp_err_t*)p_data);

    switch(ssp_err_g_sf_touch_button){
        case SSP_ERR_CTSU_OFFSET_ADJUSTMENT_FAILED:
            Blink_Red_Led = 1;
            break;
        case SSP_ERR_CTSU_SC_OVERFLOW:
            break;
        case SSP_ERR_CTSU_RC_OVERFLOW:
            break;
        case SSP_ERR_CTSU_SAFETY_CHECK_FAILED:
            break;
        default:
            break;
    }
}
```

図 19 自動エラーチェック

ユーザは、他の使用可能なイベントに基づいてエラー処理を追加することができます。

3.5 Synergy 用 CTW との USB 通信

S3A7 ソフトウェアは USB を使用して、J9 の Synergy 用 CTW と通信します (図 28 参照)。Windows 10 で作業する場合は、USB ドライバをインストールする必要はありません。USB シリアルデバイスはサンプルプロジェクトで列挙されます。

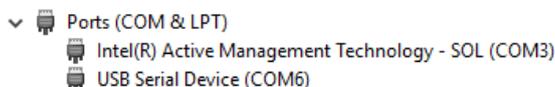


図 20 Windows 10 の USB シリアルデバイス COM ポート

Windows 7 で作業する場合は、USB CDC / ACM デバイスドライバをインストールする必要があります。この 32 ビット版と 64 ビット版のドライバを、この文書最後の付録に示します。

1. 付録で指定した 2 つのファイルを PC 上のフォルダに作成し、その内容を各ファイルにコピー/ペーストします。

CDC / ACM デバイスは、**Universal Serial Bus Controller** グループの下の PC のデバイスマネージャに **UNKNOWN DEVICE** として表示されます。

2. このデバイスを右クリックし、[Update Driver Software] を選択します。
3. ドライバの場所の入力を求められたら、前の手順で作成した 2 つのファイルの場所を参照します。ドライバが更新されると、新しい COM デバイスがデバイスマネージャに表示されます。

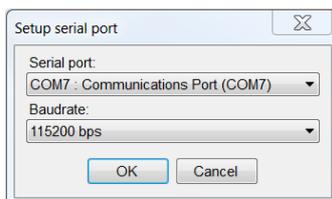


図 21 Windows 7 の通信ポート

4. アプリケーションプロジェクトの操作

4.1 プロジェクト AE_CAP1_S124_MT_App

図 22 のハードウェア設定に従って、AE-CAP1-S124 および AE-CAP1-BWS を立ち上げます。

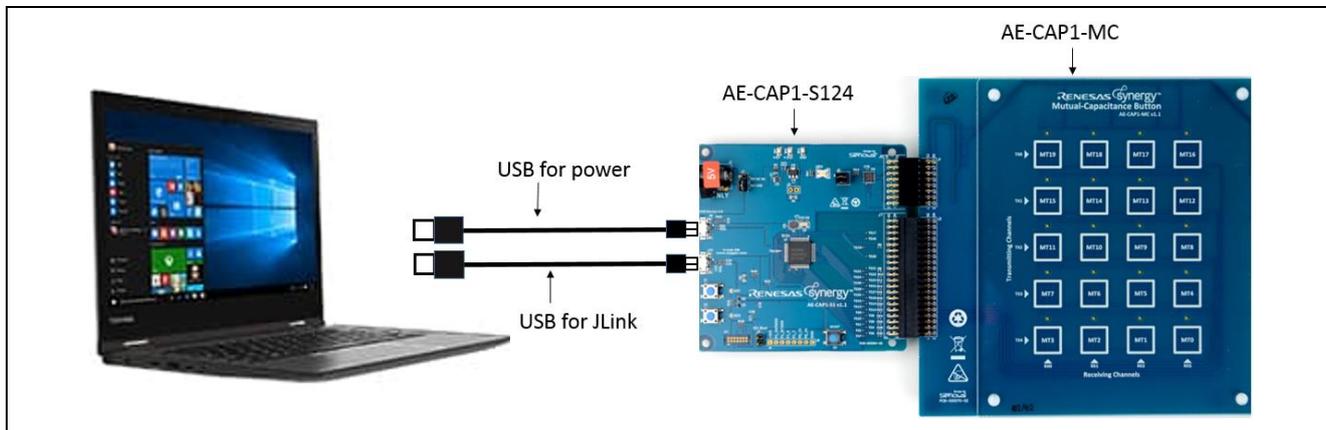


図 22 プロジェクト AE_CAP1_S124_MT_App ハードウェアのセットアップ

図 23 に示すように、ジャンパ J4 をオープンにし、ジャンパ J2 を USB デバイス側にして、J5 にジャンパを取り付けます。

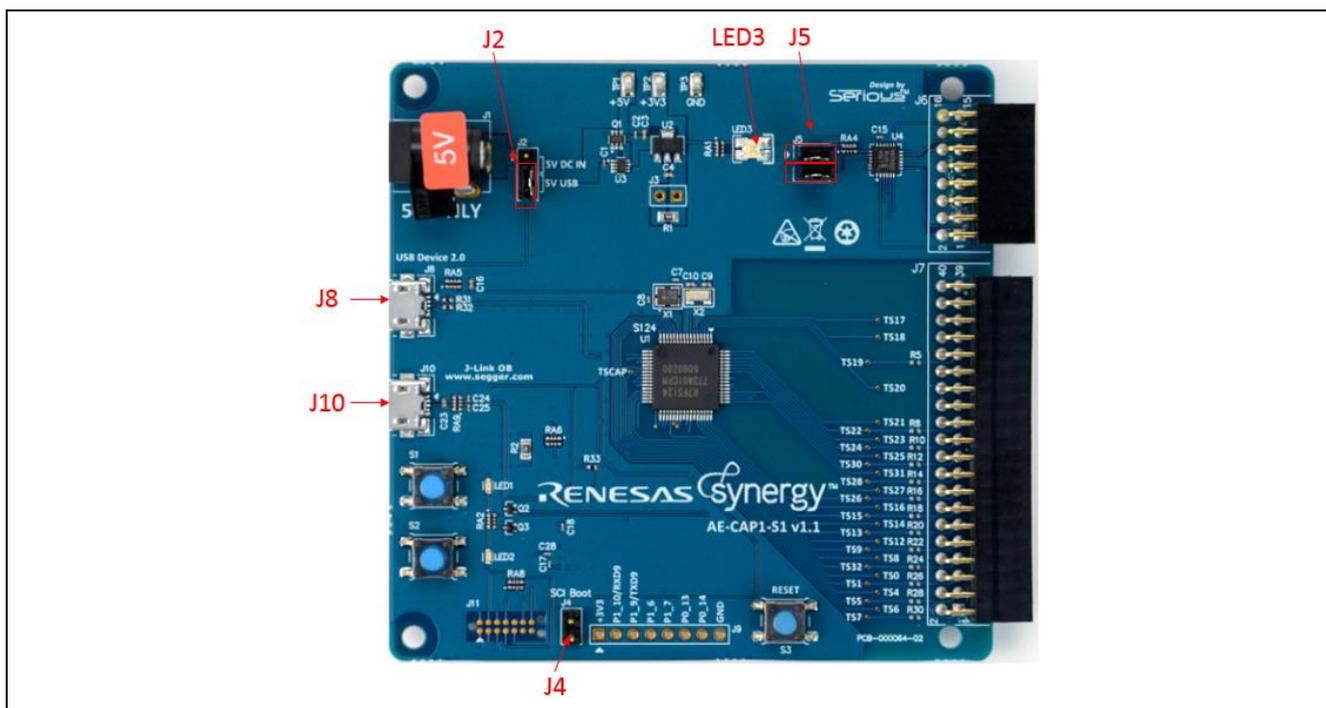


図 23 アプリケーションボード LED 制御用の AE-CAP1-S1 設定

1. システムに電源を供給するために、付属の USB ケーブルを使用して、AE-CAP1-S1 を J8 から PC に接続します。正常に電力接続されると、LED3 が点灯します。
2. J-Link 接続のために、他の付属 USB ケーブルを使用して、AE-CAP1-S124 を J10 から PC に接続します。
3. プロジェクトをビルドしてダウンロードして、実行します。

4.1.1 プロジェクト AE_CAP1_S124_MT_App の操作

ダウンロード後に実行すると、システムで次のイベントが順番に発生します。

1. ボタン/ホイール/スライダボードでオートチューニングテストを実行します (約 2 秒間)。
2. このオートチューニングが終了すると、AE-CAP1-S1 ボードの LED1 (赤) と LED2 (緑) が 5 回点滅します。**LED1 と LED2 が点滅し始める前にボードに触れないでください。**
3. オートチューニングが成功すると、LED2 (緑) が点滅し続け、ボードはユーザのタッチを受け入れる準備ができていることを示します。押すと、ボタンの上にある LED が点灯し、リリースを押すと消灯します。
4. オートチューニングが失敗すると、LED1 (赤) だけが点滅し、ハードウェア不良を示します。ハードウェア不良が発生した場合：
 - a. 正しいアプリケーションボード AE-CAP1-MT を使用していることを確認してください。
 - b. 使用しているボードが絶縁物の上に置かれていることを確認してください。

4.2 プロジェクト AE_CAP1_S124_MT_UART_Monitor_App

以下のハードウェア設定に従って、AE-CAP1-S124 および AE-CAP1-MT を立ち上げます。

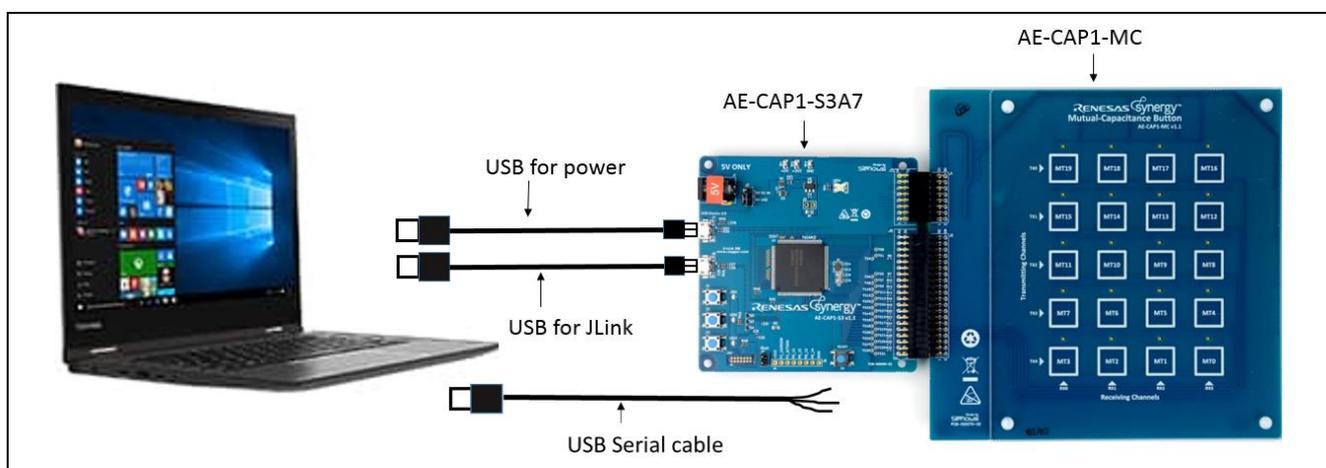


図 24 プロジェクト AE_CAP1_S124_MT_UART_Monitor_App ハードウェアの設定

1. AE-CAP1-S1 では、ジャンパ J4 をオープンにして、ジャンパ J2 を USB デバイス側にして、**ジャンパを J5 から取り外します (J5 をオープンにします)**。J4、J2、J5 の位置については、図 23 を参照してください。
2. 付属の USB ケーブルを使用して AE-CAP1-S1 を J8 から PC に接続し、システムに電源を供給します。
3. J-Link 接続のために、他の付属 USB ケーブルを使用し、AE-CAP1-S124 を J10 から PC に接続します。
4. 図 25 のように、AE-CAP1-S124 J9 を Serial-to-USB 変換ケーブルを介して PC に接続します。**Serial-to-USB 変換ケーブルはキットに含まれていません。**

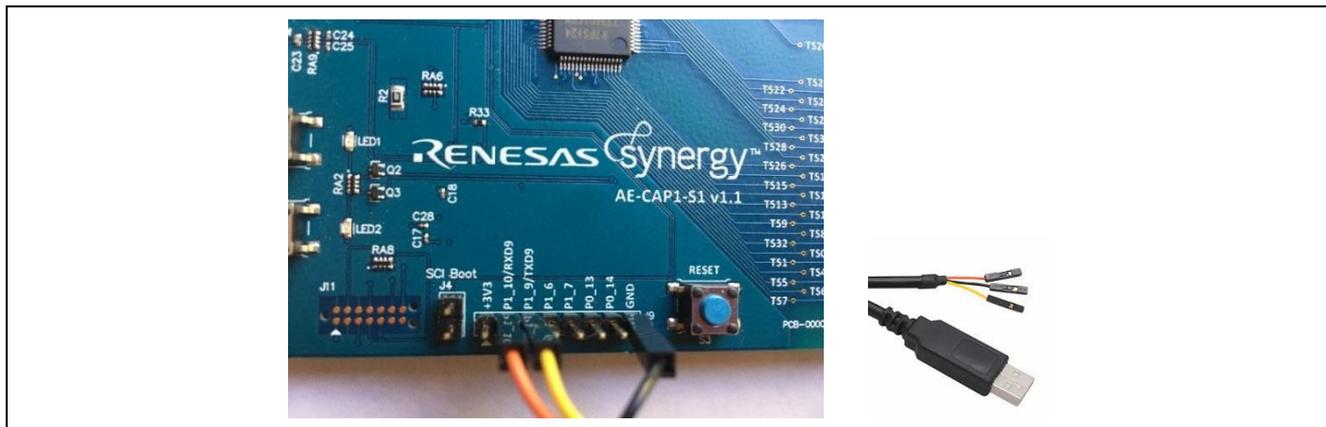


図 25 直列接続

4. プロジェクトをビルド、ダウンロード、実行します。

4.2.1 プロジェクト AE_CAP1_S124_MT_UART_Monitoring アプリケーションの操作

(1) スタンドアロンモード

デバッガに接続せずにシステムを実行している場合は、システムの電源をオンにした後に、[Reset]ボタンを押してシステムを起動することをお勧めします。実行時には、次のイベントがシステム内で順番に発生します

1. システムは、ボタン/ホイール/スライダボードでオートチューニングテストを実行します（約2秒間）。
2. オートチューニングが終了すると、AE-CAP1-S1 ボードの LED1（赤）と LED2（緑）が5回点滅します。**LED1 と LED2 が点滅し始める前に、ボードに触れないでください。**
3. オートチューニングが成功すると、LED2（緑）が点滅し続け、ボードはユーザのタッチ感知を受け入れる準備ができたことを示します。**AE-CAP1-MT ボードの LED はデモモードではアクティブではありませんのでご注意ください。**
4. オートチューニングが失敗すると、LED1（赤）だけが点滅し、ハードウェア不良を示します。ハードウェア不良が発生した場合:
 - a. 正しいターゲットボード AE-CAP1-MT を使用していることを確認してください。
 - b. 使用しているボードが絶縁物の上に置かれていることを確認してください。

(2) Renesas Synergy 用静電容量式タッチワークベンチ（Synergy 用 CTW）との通信

Synergy 用 CTW と通信するには、UART のボーレートの 115200 を選択して通信を確立します。

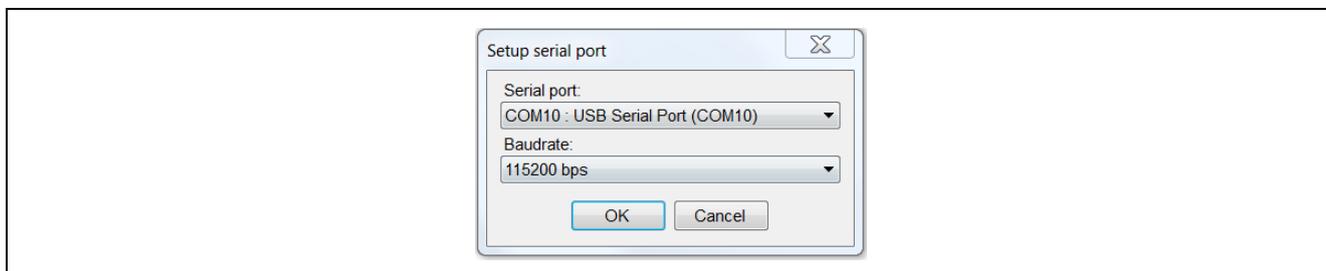


図 26 UART 通信速度設定

AE-CAP1-MC のマーキングを CTW の S124 用 Synergy のタッチボタン割り当てにマッピングするためには、表 3 を参照してください。

Table 3 S124 相互静電容量式ボタン用 CTW への AE-CAP1-MC ボードシルクスクリーンマーキングのマッピング

Synergy 割り当てのための CTW	AE-CAP1-MC マーク	Synergy 割り当てのための CTW	AE-CAP1-MC マーク	Synergy 割り当てのための CTW	AE-CAP1-MC マーク
Mtx0	MT16	Mtx7	MT9	Mtx14	MT2
Mtx1	MT12	Mtx8	MT5	Mtx15	MT19
Mtx2	MT8	Mtx9	MT1	Mtx16	MT15
Mtx3	MT4	Mtx10	MT18	Mtx17	MT11
Mtx4	MT0	Mtx11	MT14	Mtx18	MT7
Mtx5	MT17	Mtx12	MT10	Mtx19	MT3
Mtx6	MT13	Mtx13	MT6		

4.3 プロジェクト AE_CAP1_S3_MT_App

図 27 のハードウェア設定に従って、AE-CAP1-S3A7 および AE-CAP1-BWS を立ち上げます。

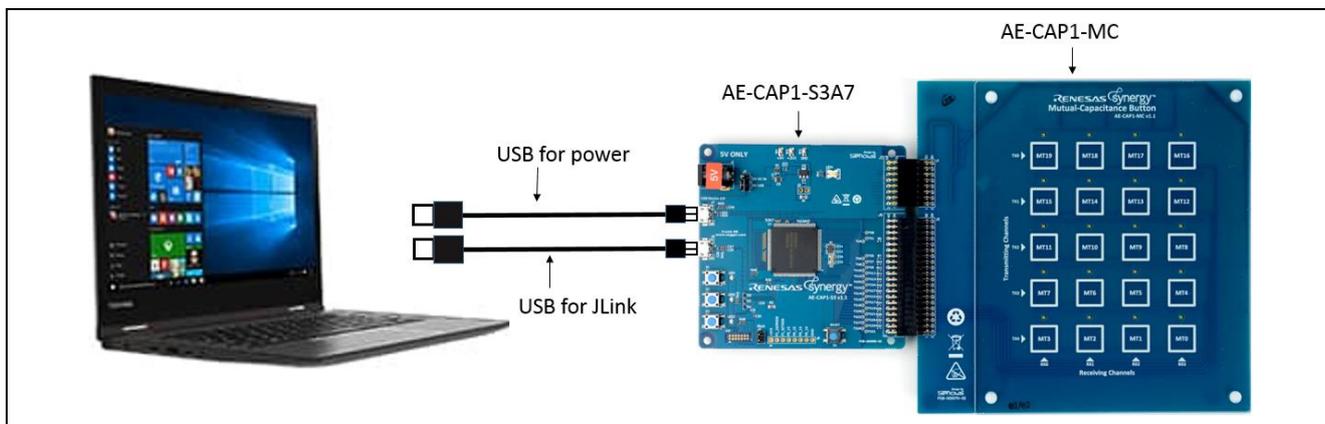


図 27 プロジェクト AE_CAP1_S3A7_MT_App ハードウェアセットアップ

ジャンパ J4 をオープンして、ジャンパ J2 を USB デバイス側にします

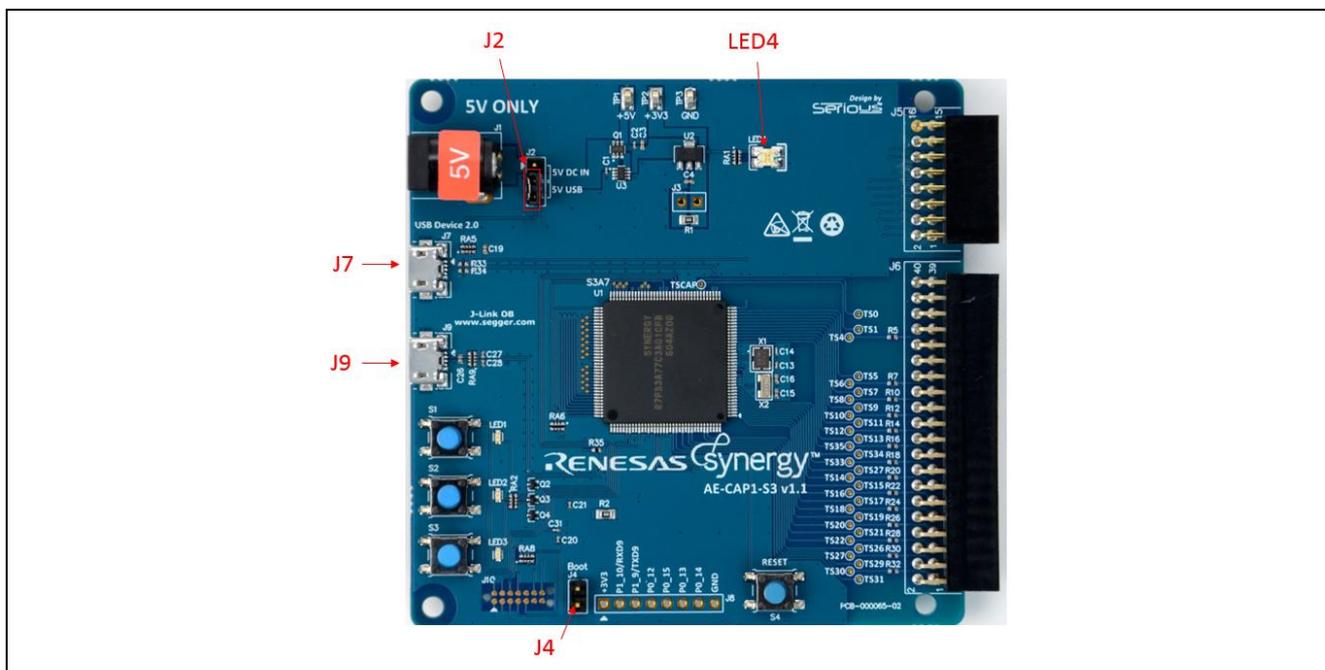


図 28 AE-CAP1-S3 設定

1. システムに電源を供給するために、付属の USB ケーブルを使用して、AE-CAP1-S3 を J7 から PC に接続します。
2. A-CAP1-S3A7 を J9 から PC に接続し、付属の他の USB ケーブルを使用して J-Link 接続を行います。
3. プロジェクトをビルド、ダウンロード、実行します。

4.3.1 プロジェクト AE_CAP1_S3A7_MT_App の操作

(1) スタンドアロンモード

デバッガに接続せずにシステムを実行している場合は、システムの電源をオンにした後に、[Reset] ボタンを押してシステムを起動することをお勧めします。実行時には、次のイベントがシステム内で順番に発生します

1. システムは、ボタン/ホイール/スライダボードでオートチューニングテストを実行します（約 2 秒）

- 間)。
2. オートチューニングが終了すると、AE-CAP1-S3 ボードの LED1 (赤)、LED2 (黄)、LED3 (緑) が 5 回点滅します。LED1、LED2、LED3 が点滅し始める前にボードに触れないでください。
 3. オートチューニングが成功すると、LED3 (緑) が点滅し続け、ボードはユーザのタッチを受け入れる準備ができたことを示します。押すと、ボタンの上にある LED が点灯し、リリースを押すと消灯します。
 4. オートチューニングが失敗すると、LED1 (赤) だけが点滅し、ハードウェア不良を示します。ハードウェア不良が発生した場合:
 - a. 正しいターゲットボード AE-CAP1-MT を使用していることを確認してください
 - b. 使用しているボードが絶縁物の上に置かれていることを確認してください

(2) Renesas Synergy 用静電容量式タッチワークベンチ (Synergy 用 CTW) との通信

Synergy 用 CTW との通信方法を確認するには、AE-CAP1 クイックスタートガイドの第 3.1 章と第 5 章を参照してください。このプログラムを使用するときは、**J7 USB デバイスポートを PC に接続してください**。表 4 を参照して、AE-CAP1-MC のマーキングを S3A7 のタッチセンサーチャンネルにマッピングします。

表 4 S3A7 相互静電容量式ボタン用 CTW への AE-CAP1-MC ボードシルクスクリーンマーキングのマッピング

Synergy 割り当て用 CTW	AE-CAP1-MC マーク	Synergy 割り当て用 CTW	AE-CAP1-MC マーク	Synergy 割り当て用 CTW	AE-CAP1-MC マーク
Mtx0	MT4	Mtx7	MT13	Mtx14	MT18
Mtx1	MT0	Mtx8	MT9	Mtx15	MT7
Mtx2	MT12	Mtx9	MT17	Mtx16	MT3
Mtx3	MT8	Mtx10	MT6	Mtx17	MT15
Mtx4	MT16	Mtx11	MT2	Mtx18	MT11
Mtx5	MT5	Mtx12	MT14	Mtx19	MT19
Mtx6	MT1	Mtx13	MT10		

5. 付録

5.1 CDC_ACM_Template_Win7_64bit.inf:

```
*****
; Windows USB CDC ACM Setup File
; Copyright (c) 2000 Microsoft Corporation

[Version]
Signature="$Windows NT$"
Class=Ports
ClassGuid={4D36E978-E325-11CE-BFC1-08002BE10318}
Provider=%MFGNAME%
LayoutFile=layout.inf
CatalogFile=%MFGFILENAME%.cat
DriverVer=11/15/2007,5.1.2600.0

[Manufacturer]
%MFGNAME%=DeviceList, NTamd64

[DestinationDirs]
DefaultDestDir=12

;-----
; Windows 2000/XP/Vista-32bit Sections
;-----

[DriverInstall.nt]
include=mdmcpq.inf
CopyFiles=DriverCopyFiles.nt
AddReg=DriverInstall.nt.AddReg

[DriverCopyFiles.nt]
usbser.sys,,0x20

[DriverInstall.nt.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,%DRIVERFILENAME%.sys
HKR,,EnumPropPages32,,"MsPorts.dll,SerialPortPropPageProvider"

[DriverInstall.nt.Services]
AddService=usbser, 0x00000002, DriverService.nt

[DriverService.nt]
DisplayName=%SERVICE%
ServiceType=1
StartType=3
ErrorControl=1
ServiceBinary=%12%\%DRIVERFILENAME%.sys

;-----
; Vista-64bit Sections
;-----

[DriverInstall.NTamd64]
include=mdmcpq.inf
```

RenesasSynergy™ プラットフォーム SynergyS124 および S3A7 MCU を使用した相互容 量式タッチソフトウェアのアプリケーション設計

```
CopyFiles=DriverCopyFiles.NTamd64
AddReg=DriverInstall.NTamd64.AddReg

[DriverCopyFiles.NTamd64]
%DRIVERFILENAME%.sys,,0x20

[DriverInstall.NTamd64.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,%DRIVERFILENAME%.sys
HKR,,EnumPropPages32,, "MsPorts.dll,SerialPortPropPageProvider"

[DriverInstall.NTamd64.Services]
AddService=usbser, 0x00000002, DriverService.NTamd64

[DriverService.NTamd64]
DisplayName=%SERVICE%
ServiceType=1
StartType=3
ErrorControl=1
ServiceBinary=%12%\%DRIVERFILENAME%.sys

;-----
; Vendor and Product ID Definitions
;-----
; When developing your USB device, the VID and PID used in the PC side
; application program and the firmware on the microcontroller must match.
; Modify the below line to use your VID and PID. Use the format as shown below.
; Note: One INF file can be used for multiple devices with different VID and PIDs.
; For each supported device, append ",USB\VID_xxxx&PID_yyyy" to the end of the line.
;-----
[SourceDisksFiles]
[SourceDisksNames]
[DeviceList]
%DESCRIPTION%=DriverInstall, USB\VID_0000&PID_0000

[DeviceList.NTamd64]
%DESCRIPTION%=DriverInstall, USB\VID_0000&PID_0000

;-----
; String Definitions
;-----
;Modify these strings to customize your device
;-----
[Strings]
MFGFILENAME="CDC_vista"
DRIVERFILENAME ="usbser"
MFGNAME="ExpressLogic"
DESCRIPTION="Communications Port"
SERVICE="USB RS-232 Emulation Driver"
```

5.2 CDC_ACM_Template_Win7_32bit.inf:

```
; CDC_ACM.inf
;
; INF file for ExpressLogic simple CDC/ACM class
;
; 1) Replace VID/PID to your own in [MYCORP] section
```

RenesasSynergy™ プラットフォーム SynergyS124 および S3A7 MCU を使用した相互容 量式タッチソフトウェアのアプリケーション設計

```
; VID_vvvv&PID_pppp
; vvvv, pppp: four digit hex number of VID and PID, respectively
;
; 2) Replace 'MYCORP' to your own abbreviated one (without space)
; ex ExpressLogic
; - Replace all MYCORP in this inf file
;
; 3) Replace 'MYDEV000' to your device model number (without space)
; ex CDC ACM Example
; - Replace all MYDEV000 in this inf file
;
; 4) Edit the strings in [Strings] section
;
[Version]
Signature="$Windows NT$"
Class=Ports
ClassGuid={4D36E978-E325-11CE-BFC1-08002BE10318}

Provider=%MYCORP%
LayoutFile=layout.inf
DriverVer=08/04/2004,5.1.2600.2180

[Manufacturer]
%MYCORP%=MYCORP

[MYCORP]
%MYDEV000%= MYDEV000,USB\VID_0000&PID_0000

[DestinationDirs]
FakeModemCopyFileSection=12
DefaultDestDir = 12

[MYDEV000.NT]
include=mdmcpq.inf
CopyFiles=FakeModemCopyFileSection
AddReg=MYDEV000.NT.AddReg

[MYDEV000.NT.Services]
AddService = usbser, 0x00000002, Service_Inst

[Service_Inst]
DisplayName = %Serial.SvcDesc%
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
StartType = 3 ; SERVICE_DEMAND_START
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
ServiceBinary = %12%\usbser.sys
LoadOrderGroup = Base

[MYDEV000.NT.AddReg]
HKR,,NTMPDriver,*ntkern
HKR,,NTMPDriver,usbser.sys
HKR,,EnumPropPages32,"MsPorts.dll,SerialPortPropPageProvider"
HKR,,PortSubClass,1,01

[Strings]
MYCORP = "ExpressLogic" ; Your company name
MYDEV000 = "CDC ACM example device" ; Device description
Serial.SvcDesc = "CDC ACM Driver" ; Device driver description
```

6. まとめ、および関連資料

1. 開発ツールとユーティリティの詳細については、<http://www.renesassynergy.com/tools> を参照してください。それらをダウンロードするには <http://www.renesassynergy.com/gallery> をご覧ください。
2. <http://www.renesassynergy.com/docs> アプリケーションノート「*Touch IP Gen. 2 Design and Layout Guide*」をダウンロードし、Renesas の静電容量式タッチのハードウェア設計ガイドラインをご覧ください。
3. <http://www.renesassynergy.com/docs> からアプリケーションノート「*Synergy Capacitive Touch Tuning Usage Note*」の使用上の注意をダウンロードし、SynergyCTW の静電容量式タッチチューニングをご覧ください。
4. 詳細について:
 - a. Synergyキット : <http://www.renesassynergy.com/kits>
 - b. Synergyマイクロコントローラ : <http://www.renesassynergy.com/microcontrollers>
 - c. AE-CAP1キットで実行される他のSynergyアプリケーションプロジェクトを調べて変更する:
<http://www.renesassynergy.com/kits/ae-cap1>
 - d. Synergyソフトウェア <http://www.renesassynergy.com/m/software>
 - e. Synergyソリューション <http://www.renesassynergy.com/solutions>

ホームページとサポート窓口

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

テクニカルサポート:

- アメリカ: https://renesas.zendesk.com/anonymous_requests/new
- ヨーロッパ: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

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	2017.05.04	-	初版

ご注意書き

- 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生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
 - 当社製品、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。
 - 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。
 - 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。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。
 - 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、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、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損害が生じて、当社は一切その責任を負いません。
 - 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。
 - 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。
 - 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。
 - 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、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
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。
 - お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 - 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。
 - 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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。

(Rev.3.0-1 2016.11)



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：<https://www.renesas.com/contact/>