



## Mpression 技術資料

---

### TI CapTIvate MCU Development Kit

---

### クイックスタートガイド

Revision 1.1

2017/05/23

# 目次

1. はじめに	4
2. 開けてみよう	5
2.1 内容物を確認	5
2.2 詳細	6
3. 動かしてみよう	8
3.1 CapTIvate™ Design Center GUI をダウンロード	9
3.2 CapTIvate™ Design Center GUI をインストール	10
3.3 ボードを接続	11
3.4 CapTIvate™ Design Center GUI を起動	11
3.5 プロジェクトをロード	12
3.6 動作を確認	13
3.7 グラフで確認	14
4. 新プロジェクトを作ってみよう	15
4.1 ワークスペースキャンバスを準備	15
4.2 マイコンを設定	16
4.3 センサーを追加	17
4.4 センサーの設定(ボタン)	18
4.5 センサーの設定(ホイール)	19
4.6 センサーの設定(スライダー)	20
4.7 センサーの設定(近接センサー)	21
4.8 キャンバスの完成	22
4.9 マイコンとセンサーの接続設定	23
4.10 ボードを接続	25
4.11 動作を確認	25
5. プログラムを作成してみよう	27
5.1 ベースコードを自動生成	27
5.2 コンパイル環境の準備	29
5.2.1 Code Composer Studio(CCS)をダウンロード	29
5.2.2 Code Composer Studio(CCS)を起動	29
5.2.3 作業ディレクトリ(workspace)を指定	30
5.2.4 Updateを確認	30
5.3 プロジェクトをインポート	31
5.4 ソースコードをコンパイル	32
5.5 ボードにプログラムを書き込んで実行	32

5.6 動作を確認 .....	34
5.7 プログラムを作成 .....	36
5.7.1 既存のプログラムから、LED1の点灯処理部を削除 .....	37
5.7.2 スライダーにタッチ検出時、処理を呼び出すAPIを追加.....	38
5.7.3 スライダーのタッチ検出後の処理を追加 .....	39
5.8 ボードにプログラムを書き込んで実行 .....	40
5.9 動作を確認 .....	40
<b>6. 参考プログラム .....</b>	<b>43</b>
<b>7. 安全上の注意 .....</b>	<b>44</b>
7.1 凡例 .....	44
7.2 注意事項 .....	44
<b>8. 更新履歴 .....</b>	<b>46</b>

# 1. はじめに

---

この資料は、Texas Instruments(以下、TI)製 MSP430 超低消費電力マイコンファミリの静電容量タッチソリューション「MSP CapTivate MCU Development Kit」の開発環境とその基本的な使い方について簡易にまとめたものです。内容に誤りがないように注意は払っておりますが、TI 提供のドキュメントと差異がございましたら、メーカー提供のものを優先してご参照ください。

※ご注意: “Windows 7 (32bit)”で動作の確認を行っております。他 OS での動作確認は行っておりません。

## 2. 開けてみよう

### 2.1 内容物を確認

MSP CapTivate MCU Development Kit のパッケージを開けて内容物を確認しましょう。

- |                       |                       |
|-----------------------|-----------------------|
| 1. MSP430FR2633 搭載ボード | : CAPTIVATE-FR2633    |
| 2. EMI 試験用の絶縁ボード      | : CAPTIVATE-ISO       |
| 3. プログラマ/デバッガ機能ボード    | : CAPTIVATE-PGMR      |
| 4. 自己容量方式センサーボード      | : CAPTIVATE-BSWP      |
| 5. 近接センサーボード          | : CAPTIVATE-PROXIMITY |
| 6. 相互容量方式センサーボード      | : CAPTIVATE-PHONE     |
| 7. USB Cable          |                       |

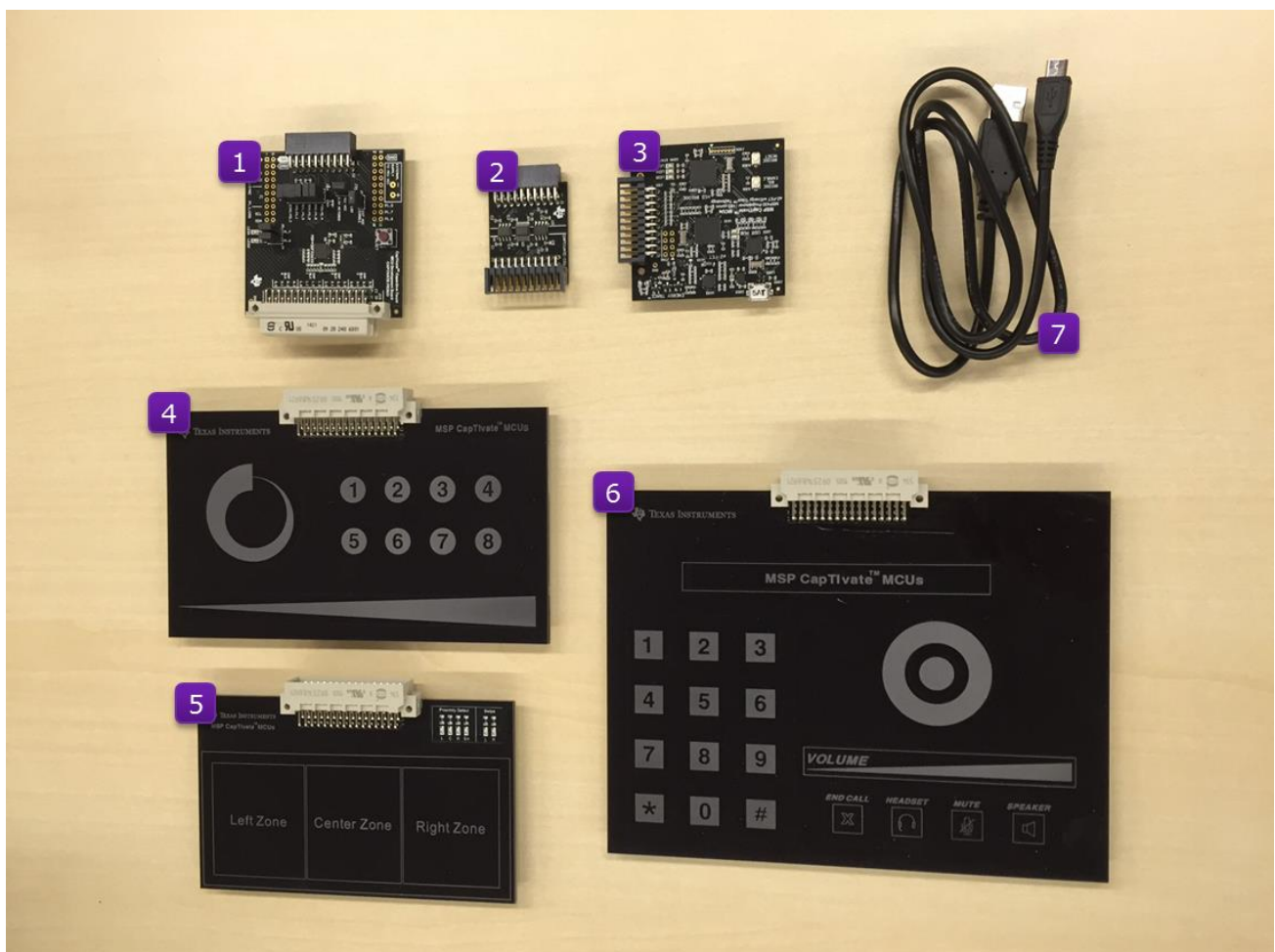


図 1 評価キット内容物

## 2.2 詳細

### 1. MSP430FR2633 搭載ボード: CAPTIVATE-FR2633

TI 製の超低消費電力 16bit マイコン「MSP430FR2633」が搭載されているボードです。

MSP430FR2633 はメカスイッチからの置き換えを想定し、超低消費電力を追及したアーキテクチャになっています。

- 低消費電力かつ SRAM 並みの高速動作が可能な不揮発メモリ (FRAM) を搭載しており、電源を切った状態でもプログラム、データの内容を保持することが可能です。
- CPU が非動作 (スリープ) 状態でも最小電力でタッチ検出を行う専用ハードウェア「CapTIvate-IP」を搭載しており、基本的なノイズフィルタ処理を含め、タッチ検出まですべて「CapTIvate-IP」だけで処理可能な構成となっております。  
タッチを検出すると割り込みで CPU を起こし処理が終了すると CPU は再びスリープすることで、CPU の消費電力を削減します。
- 電極を接続するだけで静電容量式タッチセンサーとして動作可能な IO ピンが 16 個用意されており、1 ピンに 1 つの電極を接続することで 16 個のタッチセンサーの実現はもちろんのこと、8 ピン x 8 ピンのマトリクス構成で使用することで、最大で 64 個のタッチセンサーを実現することが可能です。  
TI からの設計ガイドが以下に公開されております。  
(<http://www.ti.com/tool/tidm-captivate-64-button>)

### 2. EMI 試験用の絶縁ボード: CAPTIVATE-ISO

CapTIvate はノイズ耐性に関する IEC61000-4-6 規格に準拠するノイズ耐性の高いタッチ・ソリューション。実際にノイズ試験を行うことを想定して、EMI 試験用の絶縁ボードも付属しています。

ノイズ耐性に関する技術文書も以下に公開されております。

(<http://www.ti.com/lit/wp/slay045/slay045.pdf>)

### 3. プログラマ/デバッグ機能ボード: CAPTIVATE-PGMR

マイコンと PC をつなぐためのインターフェースボードです。

JTAG デバッグ機能をこのボードに実装しておりますので、JTAG デバッグをご購入いただくことなく、評価を開始いただけます。

### 4. 自己容量方式センサーボード: CAPTIVATE-BSWP

自己容量方式では、一つの電極と接近する導体との間の静電容量変化から、タッチを検出します。

16 個の IO ピンを使って、スライダ、ホイール、近接センサー、ボタン 8 個を実現。CapTIvate の基本機能の評価を手軽に開始いただけます。

### 5. 近接センサーボード: CAPTIVATE-PROXIMITY

一つの電極の面積が大きく作られており、センサーボードから近接する導体 (手や指) が離れていても、微小な容量変化を検出し、導体の近接を検知することが可能です。

本評価ボードでは、手をボードから浮かした状態で横に移動させることで、Left, Center, Right の 3 つの電極が検出した順番から、左右どちらに手が動いたかを判定するジェスチャー動作をご確認いただけます。

## 6. 相互容量方式センサーボード: CAPTIVATE-PHONE

マトリクス構成で多数のタッチ検出電極の実装を想定した構成の評価ボードです。

相互容量方式では、送信電極と受信電極の2つの電極間の静電容量変化を検出し、タッチを検出します。CapTivate では、相互容量方式と自己容量方式を同時に実装することも可能ですので、多数のボタンの実装はマトリクスで実装し、残りのピンは1ピンに1つの電極を接続するという構成も可能です。

このような複雑なピンアサインでも、無償で提供される (MSP CapTivate Design Center GUI (以降、GUI ツールと記載)を使用することで1クリックで自動的にピンアサインされますので、少ない労力でピンアサイン構成を試行することが可能です。

また、この評価ボードには、触覚フィードバックとして「ハプティクスのドライバ IC とアクチュエータ」が実装されており、タッチ時にハプティクスアクチュエータの振動反応動作を確認することも可能です。

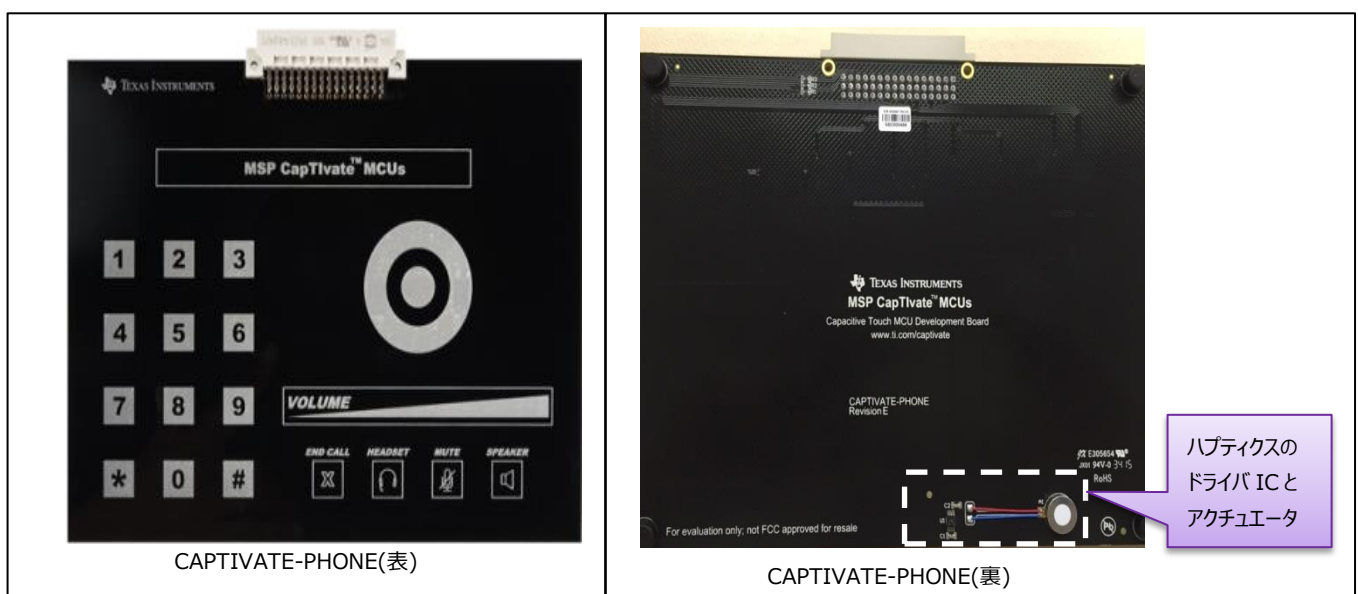


図 2 CAPTIVATE-PHONE

## 7. USB Cable

評価ボードと PC を接続するための USB ケーブルです。

PC 上で動作する開発ツールとの接続に使用されます。

電源供給にも使用しておりますので、AC アダプタ無しでこの USB ケーブル1本で評価を開始頂けます。



## 3. 動かしてみよう

では実際に必要なソフトウェアをインストールして実際にボードの動作を確認しましょう。

CapTIvate を利用するために必要なソフトウェア、技術情報は以下のサイトに集約されています。

<http://www.tij.co.jp/captivate>



Touch the Revolution with MSP MCUs featuring CapTIvate™ technology

**Touch The Revolution**  
**MSP MCUs Featuring CapTIvate™ Technology**

- + ノイズ耐性に関する IEC61000-4-6 規格準拠タッチ・ソリューション
- + 金属タッチ、3D ジェスチャ認識、手袋への対応のほか、さまざまな設定が可能なソリューション
- + 消費電力が世界最小の FRAM 静電容量式タッチ・マイコン
- + 業界最高の分解能を実現するスライダとホイール
- + CapTIvate デザイン・センターにより、5 分以下で設計をセットアップ

**CapTIvate デザイン・センター**

CapTIvate デザイン・センターにより、CapTIvate 技術に関連するツール、技術資料、設計ガイド、サンプル・コードをワンストップで入手可能。CapTIvate デザイン・センターは、最小の手間で、プログラミング・スキル・レベルがさまざま異なるデベロッパーによる静電容量式タッチ・ソリューションの製作を可能にします。また、5 分以内でのセンサの設定とチューニングが可能です。このプログラムは、Microsoft® Windows®、Apple® OS X®、Linux® で利用できます。

クリック [CapTIvate デザイン・センター](#)

図 3 CapTIvate web ページ

上記「CapTIvate デザイン・センター」をクリックすることでダウンロードサイトに移動できます。

このサイトには、online video や製品ラインナップ、関連 document 等の掲載もございますので、あわせてご参照ください。

CapTIvate 技術採用の MSP マイコン				
	MSP430FR2532	MSP430FR2632	MSP430FR2533	MSP430FR2633
FRAM/RAM	8.5K/1K	8.5K/2K	15.5K/2K	15.5K/4K
ボタン数	自己容量:最大 8 個、相互容量:最大 8 個	自己容量:最大 8 個、相互容量:最大 16 個	自己容量:最大 16 個、相互容量:最大 16 個	自己容量:最大 16 個、相互容量:最大 64 個
パッケージ	24-RGE(QFN)	24-RGE(QFN)	32-DA(TSSOP) 32-RHB(QFN)	32-DA(TSSOP) 32-RHB(QFN)

図 4 製品ラインナップ



## 3.1 CapTIvate™ Design Center GUI をダウンロード

CapTIvate™ Design Center GUI のダウンロードをクリックしてください。



図 5 ダウンロードボタン

ご希望の OS に対応したツールをダウンロードしてください。

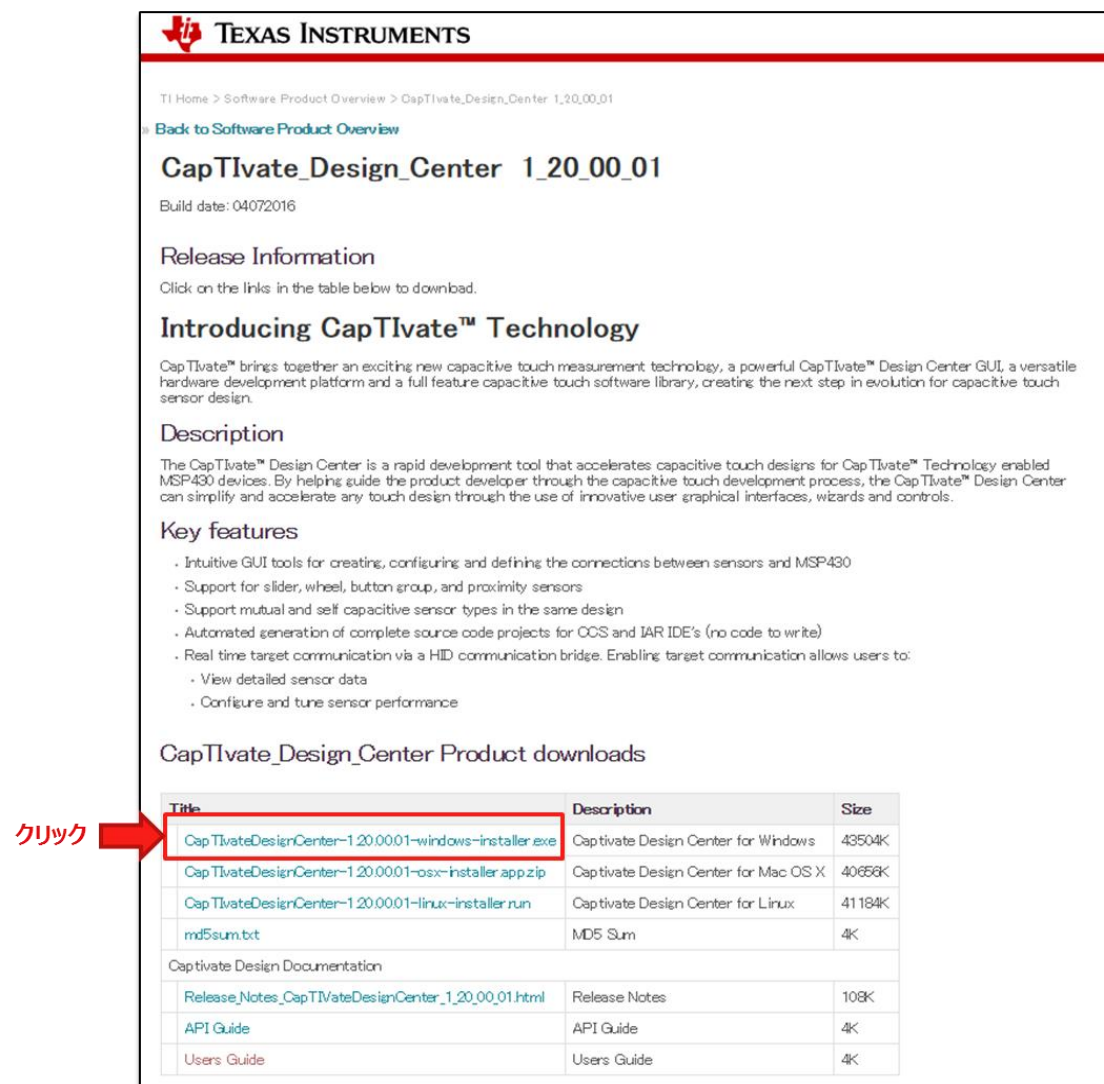


図 6 ダウンロードサイト

## 3.2 CapTivate™ Design Center GUI をインストール

License Agreement では“I accept...”を選択して Next を押してください。

Installation Directory ではデフォルトのまま(C:\ti\msp のまま)、Next を押してください。

Install が完了するまで待ちましょう。



図 7 Setup のポップアップ

### 3.3 ボードを接続

ボードを以下のように接続し、付属の USB ケーブルで PC に接続しましょう。

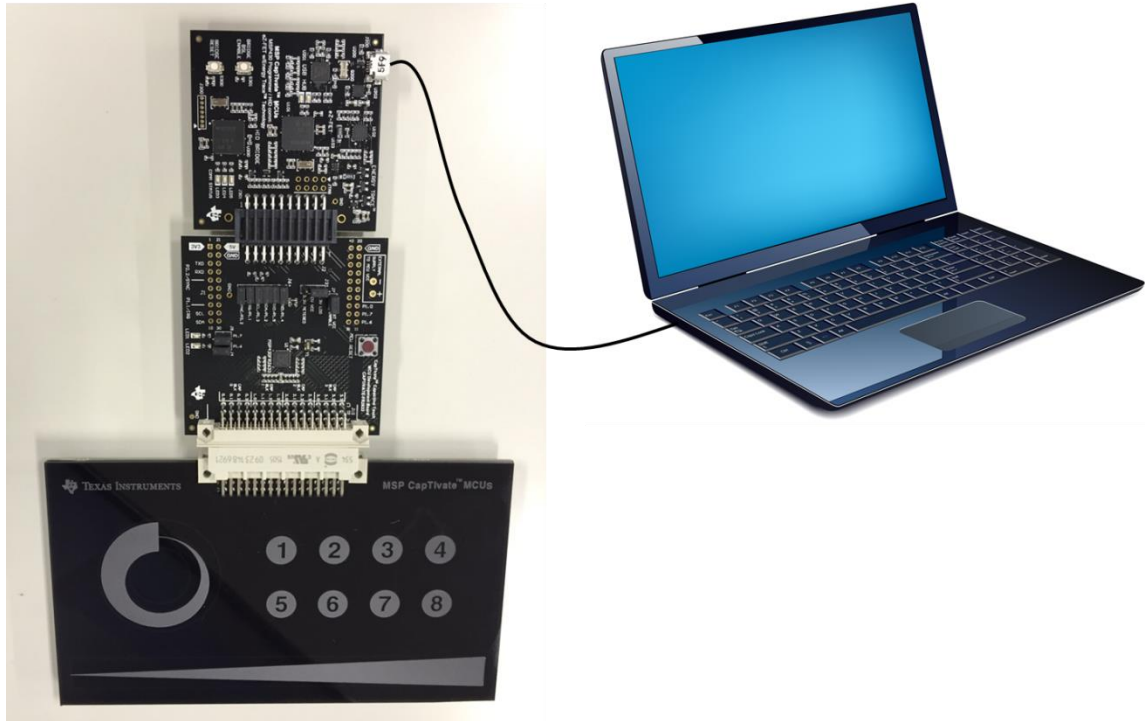


図 8 ボード接続

### 3.4 CapTivate™ Design Center GUI を起動

デスクトップにある CapTivate™ Design Center のアイコンを実行してください。  
もし、デスクトップのショートカットを削除してしまった場合、スタートメニューから以下のようにたどって CapTivate™ Design Center を起動してください。

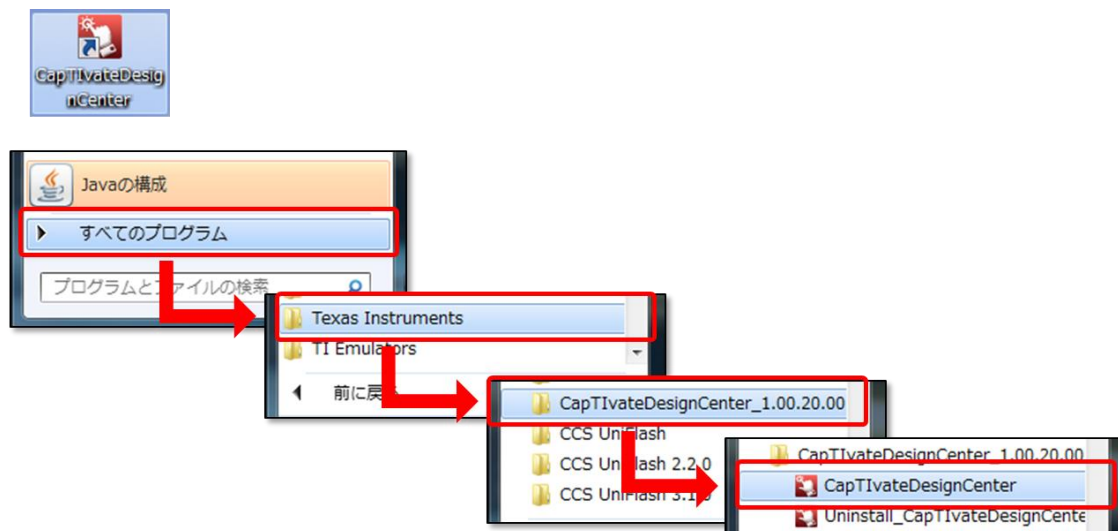


図 9 GUI アイコン

## 3.5 プロジェクトをロード

出荷時に書き込まれているサンプルコードを実行するためのプロジェクトをロードします。

- ・メニューバーの File → Project Open をクリックしてください。
- ・CAPTIVATE-BSWP を選択して Select を押してください。

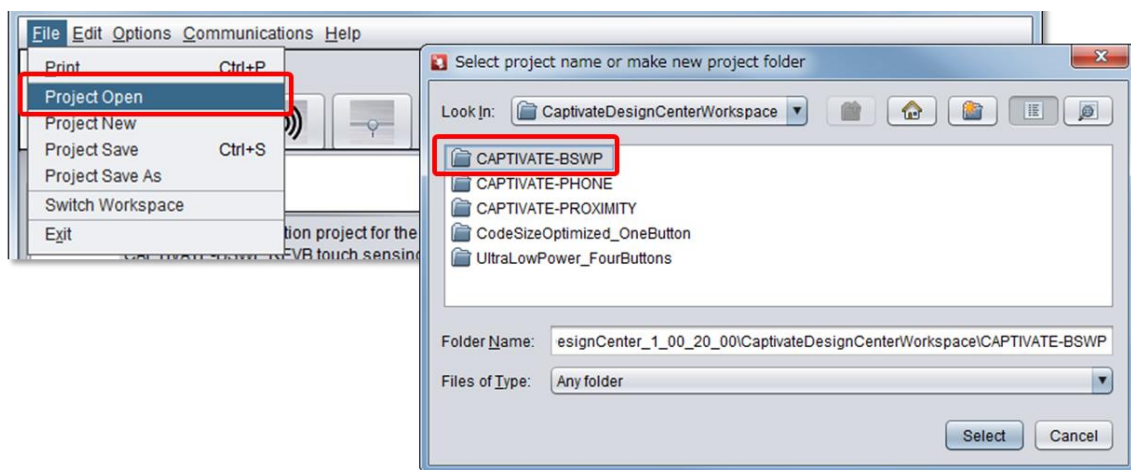


図 10 ロード画面

プロジェクトのロード後、図 11 のような画面が表示されます。

このサンプルプロジェクトでは、スライダー、ホイール、近接センサー、ボタン 8 個の動作を確認できます。

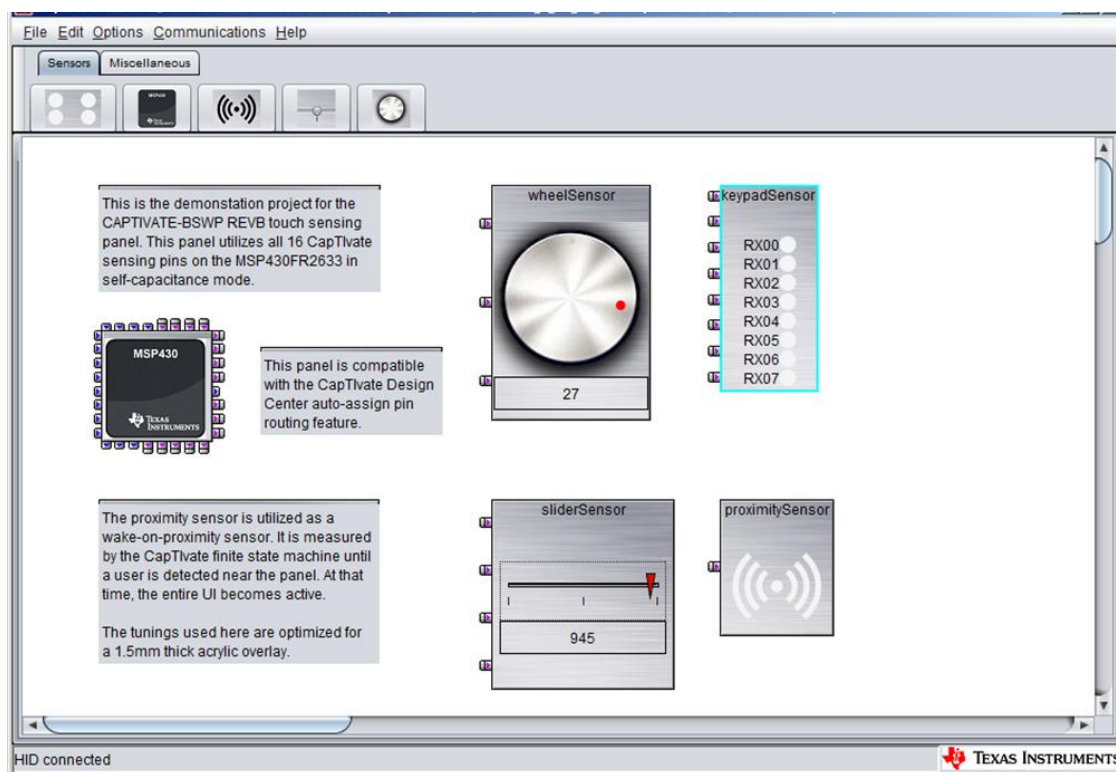
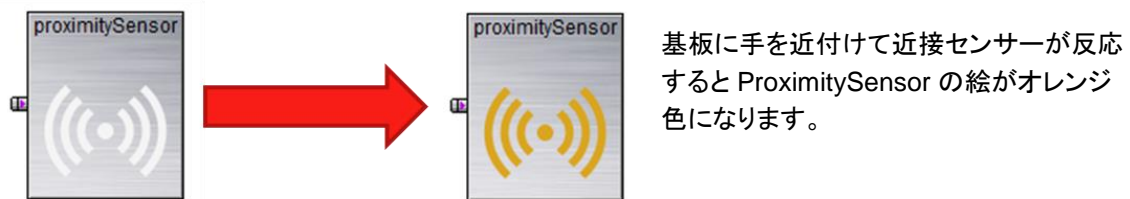
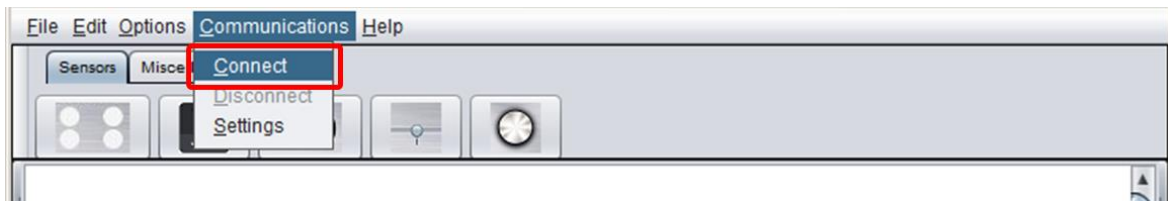


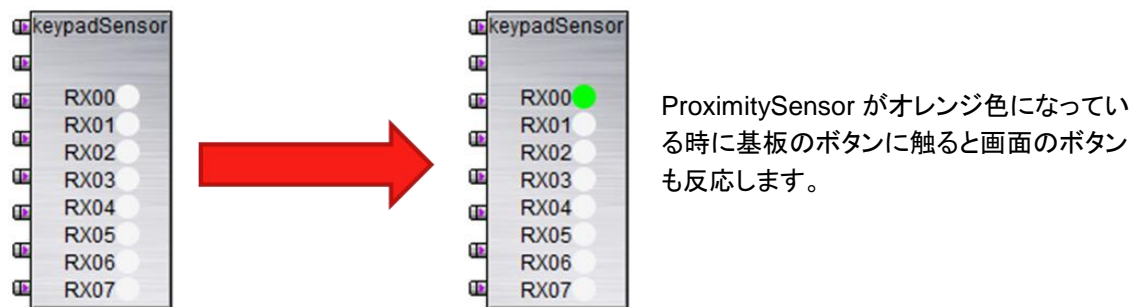
図 11 GUI 画面

## 3.6 動作を確認

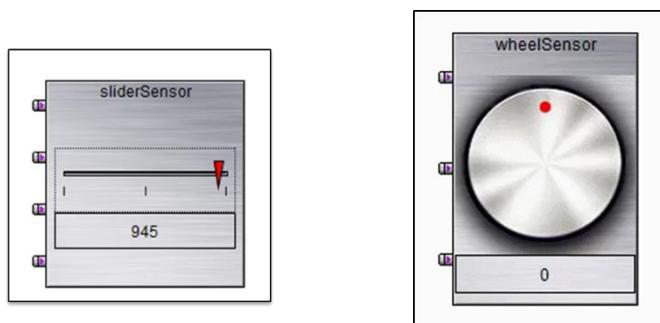
プロジェクトがロードできたらメニューバーの Communication → Connect をクリックして接続しましょう。



基板に手を近付けて近接センサーが反応すると ProximitySensor の絵がオレンジ色になります。



ProximitySensor がオレンジ色になっている時に基板のボタンに触ると画面のボタンも反応します。



同様にスライダーやホイールも触っている位置に合わせて画面のスライダー、ホイールが動きます。

図 12 各種センサー

## 3.7 グラフで確認

ボタンなどの反応をグラフで確認することができます。  
画面の部品(例: ボタン)を右クリックして Properties をクリックしましょう。  
すると、図 13 のような画面が表示されます。  
ボタンを触ると対応したセンサーの棒グラフが変化します。

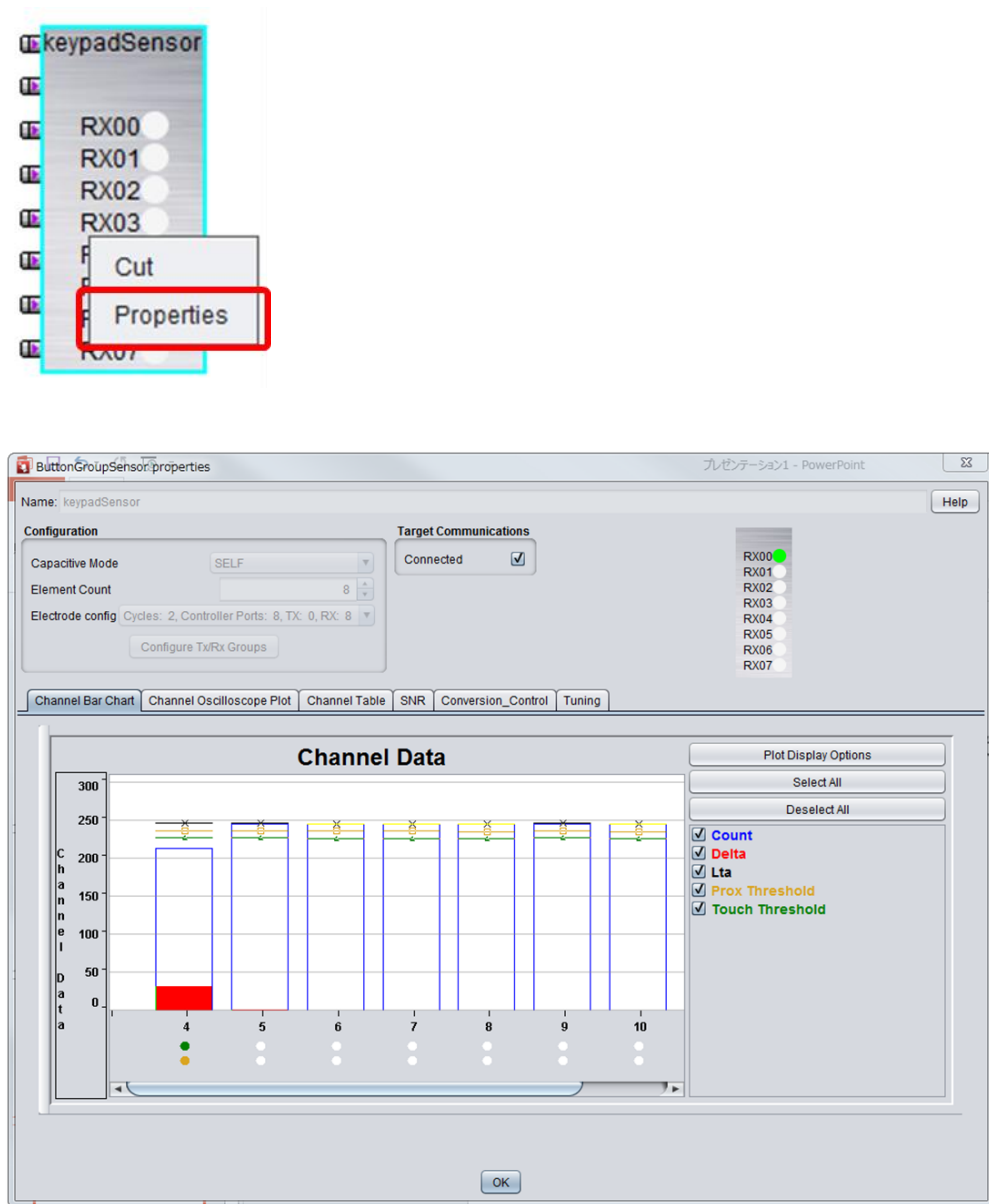


図 13 チューニング画面



## 4. 新プロジェクトを作ってみよう

次は、実際に新規プロジェクトを作成していきましょう。

付属の自己容量方式センサーボード(CAPTIVATE-BSWP)を利用してその流れを説明します。

では、CapTivate™ Design Center GUI を起動しましょう。

デスクトップにある CapTivate™ Design Center のアイコンを実行してください。

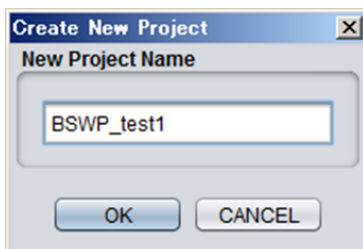


図 14 アイコン

### 4.1 ワークスペースキャンバスを準備

次に、新規のプロジェクト(ワークスペースキャンバス)を準備しましょう。

File->Project New :プロジェクト名として「BSWP\_test1」とタイプし OK をクリックしてください。



以下のように新しいキャンバスが準備されます。

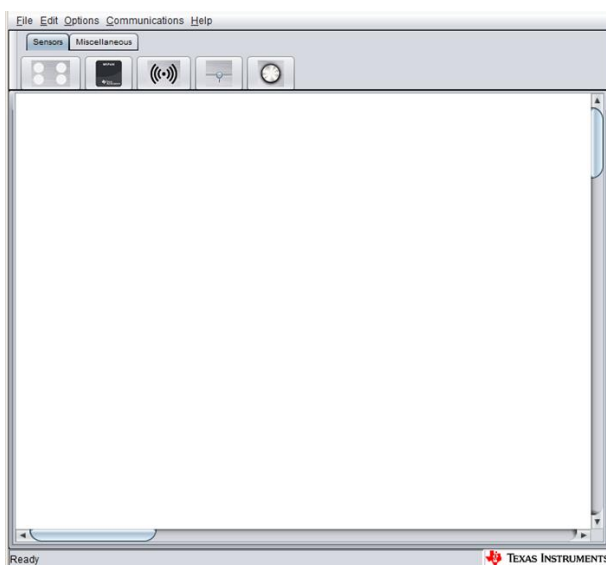


図 15 キャンバス



## 4.2 マイコンを設定

最初に以下の手順でマイコンの設定をします。

- 1: ワークスペースキャンバスにコントローラを Drag&Drop します。
- 2: マイコンの上で右クリックしプロパティビューを Open します。
- 3: コンフィギュレーション・デバイスは、ドロップダウンから MSP430FR2633IRHB を選択します。
- 4: 通信インターフェースのプルダウンから BULK\_I2C を選択し OK をクリックします。

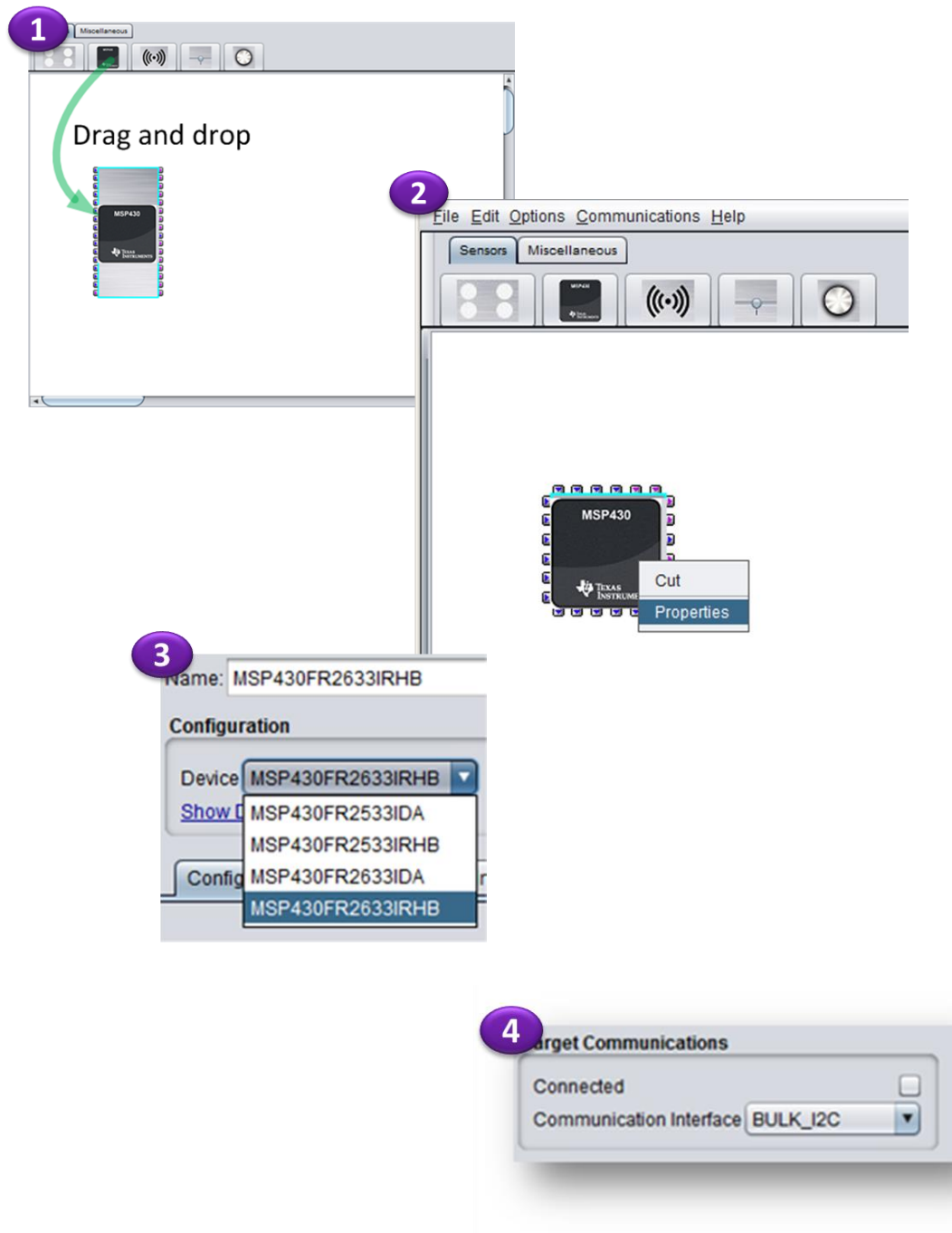


図 16 マイコン設定画面

## 4.3 センサーを追加

次に以下の手順でセンサーを追加します。

マイコンと同様に、ボタン、ホイール、スライダー、近接センサーをワークスペースキャンバスに Drag&Drop してください。

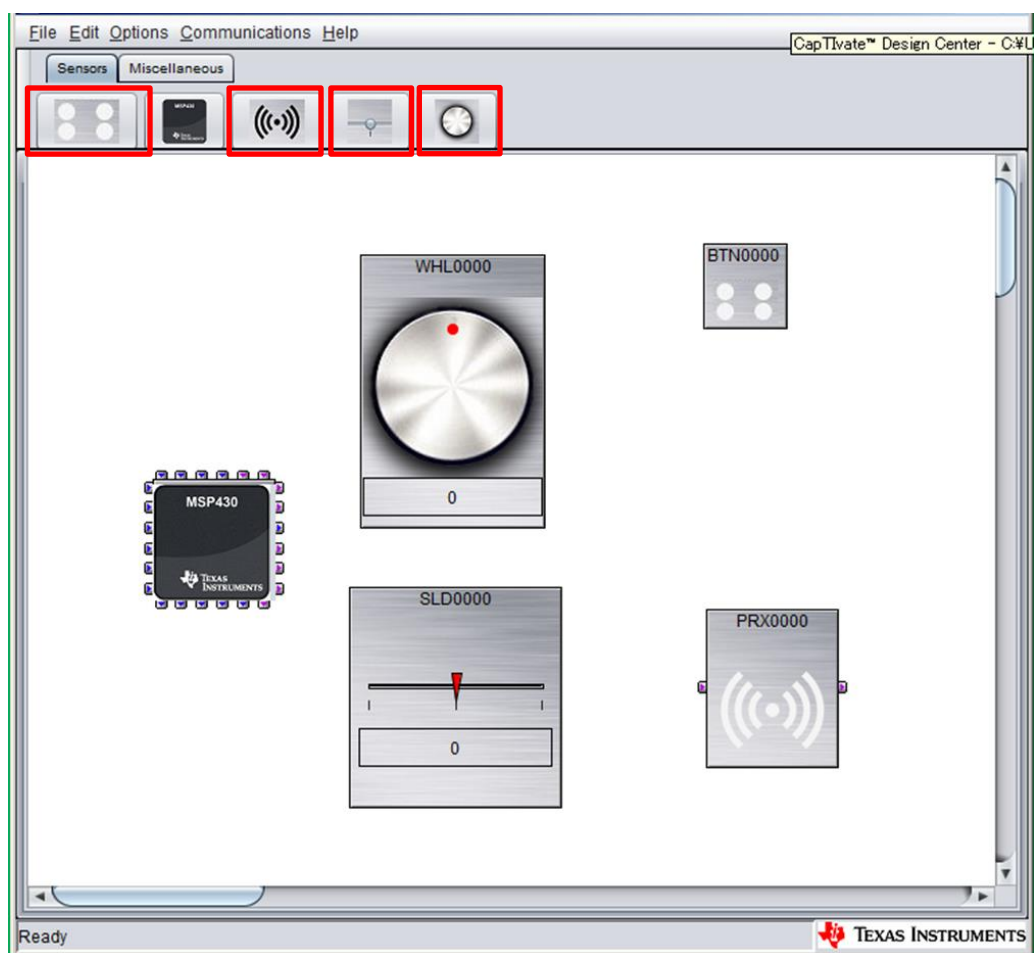
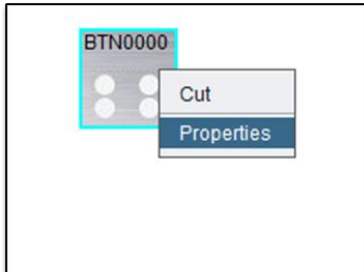


図 17 キャンバス

## 4.4 センサーの設定(ボタン)

まず、ボタンを設定しましょう。  
ボタンの上で右クリックしプロパティビューを Open してください。



以下の画像のように設定し OK をクリックします。

- ・Name を記入: 今回は"keypad"とします。
- ・Capacitive Mode : BSWP は自己容量方式 (Self Capacitance)なので"SELF"を選択します。
- ・Element Count: 今回は8個すべてのボタンを使うので、"8"と入力します。

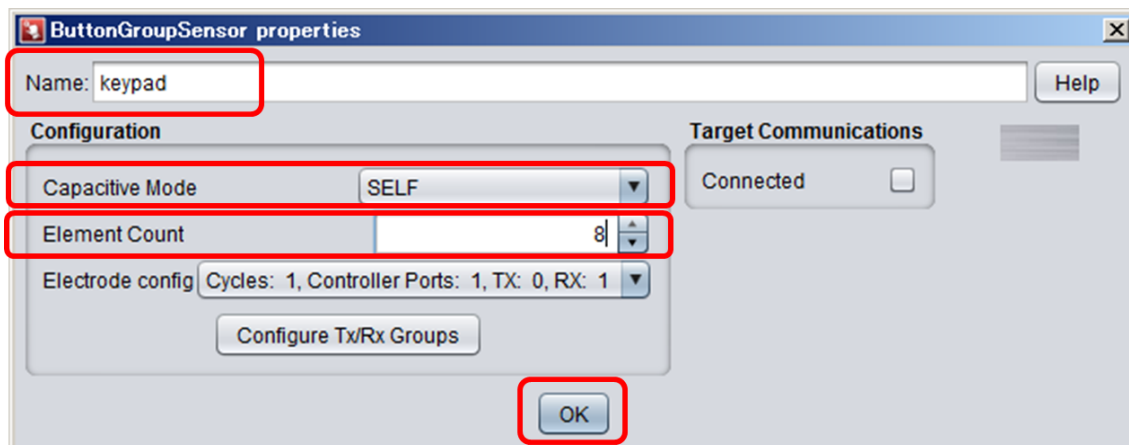
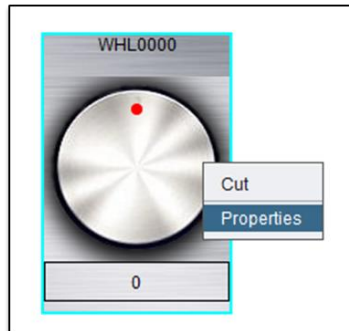


図 18 センサー設定(ボタン)

## 4.5 センサーの設定(ホイール)

次にホイールを設定しましょう。

ホイールの上で右クリックしプロパティビューを Open してください。



以下の画像のように設定し OK をクリックします。

- ・Name を記入: 今回は"wheel"とします。
- ・Capacitive Mode :BSWP は自己容量方式 (Self Capacitance)なので"SELF"を選択します。
- ・Element Count:BSWP のホイールは3個の電極で設計されていますので、"3"と入力します。

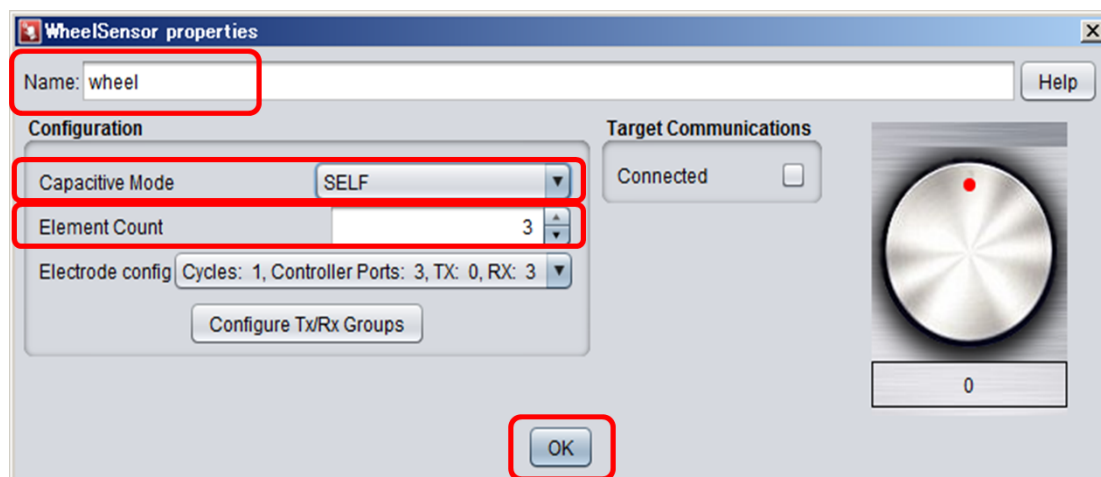


図 19 センサー設定(ホイール)

## 4.6 センサーの設定(スライダー)

同様にスライダーを設定しましょう。

スライダーの上で右クリックしプロパティビューを Open してください。



以下の画像のように設定し OK をクリックします。

- Name を記入:今回は"slider"とします。
- Capacitive Mode :BSWP は自己容量方式 (Self Capacitance)なので"SELF"を選択します。
- Element Count:BSWP のスライダーは 4 個の電極で設計されていますので、"4"と入力します。

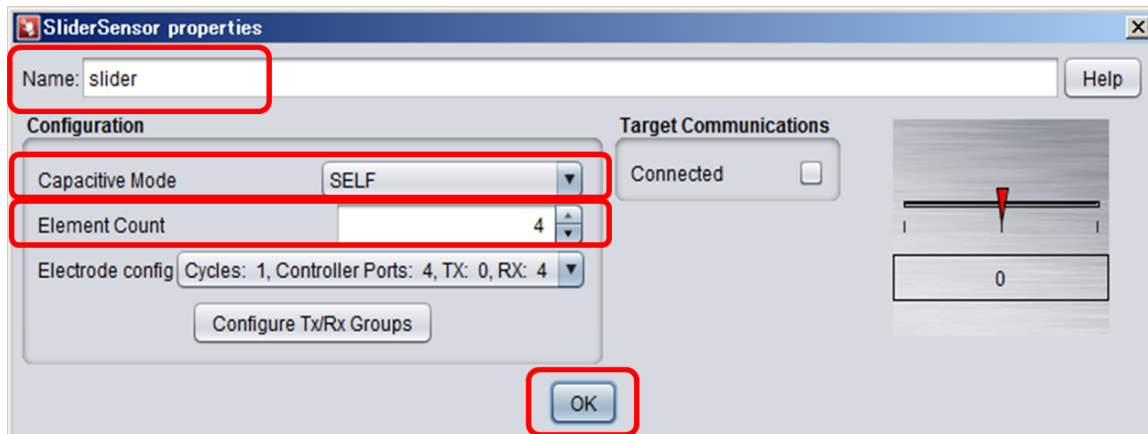
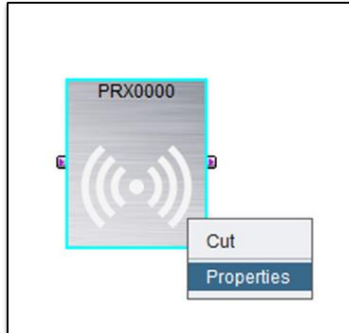


図 20 センサー設定(スライダー)

## 4.7 センサーの設定(近接センサー)

次に近接センサーを設定しましょう。

近接センサーの上で右クリックしプロパティビューを Open してください。



以下の画像のように設定し OK をクリックします。

- Name を記入: 今回は"proximity"とします。
- Capacitive Mode :BSWP は自己容量方式 (Self Capacitance)なので"SELF"を選択します。
- Element Count: BSWP の近接センサーは 1 個の電極で設計されていますので、"1"と入力します。

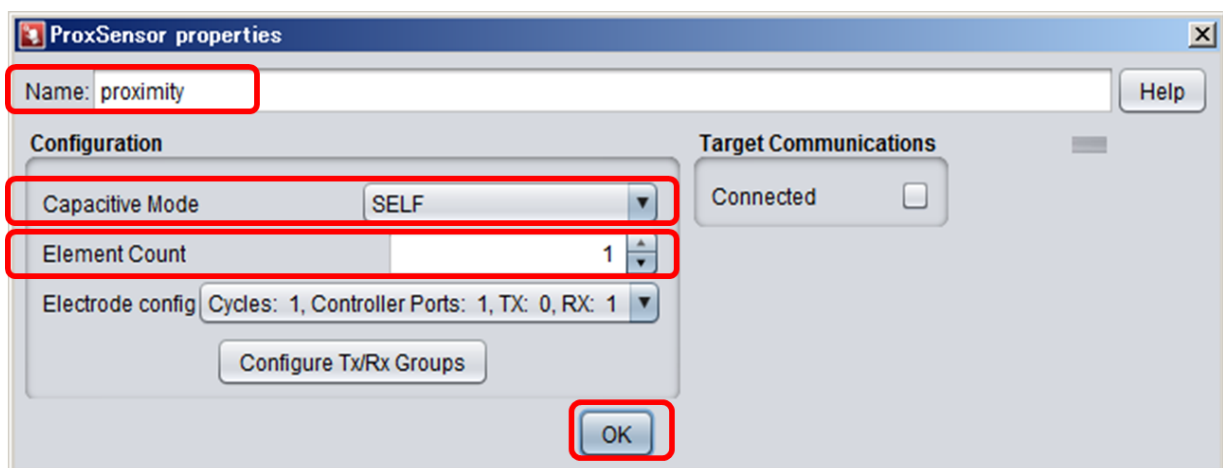


図 21 センサー設定(近接センサー)

## 4.8 キャンバスの完成

以下のようなキャンバスが完成しました。

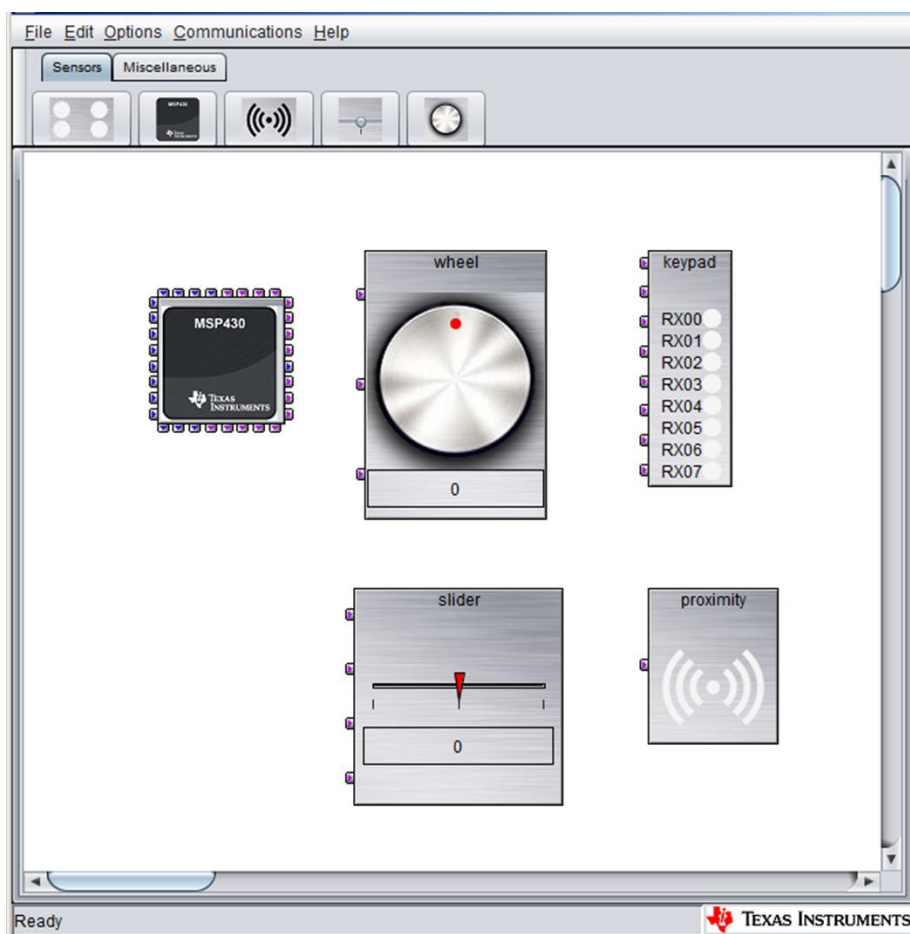


図 22 完成したキャンバス

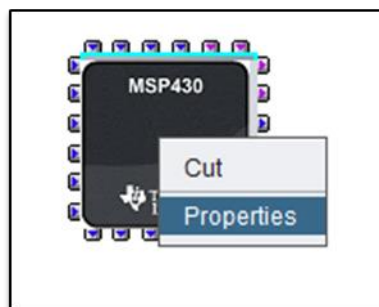


## 4.9 マイコンとセンサーの接続設定

次に、マイコンの I/O ピンとセンサーを接続します。

1クリックで自動的に接続設定が完了します。では、やってみましょう。

マイコンの上で右クリックしプロパティビューを Open してください。



Auto Assign ボタンをクリックします。

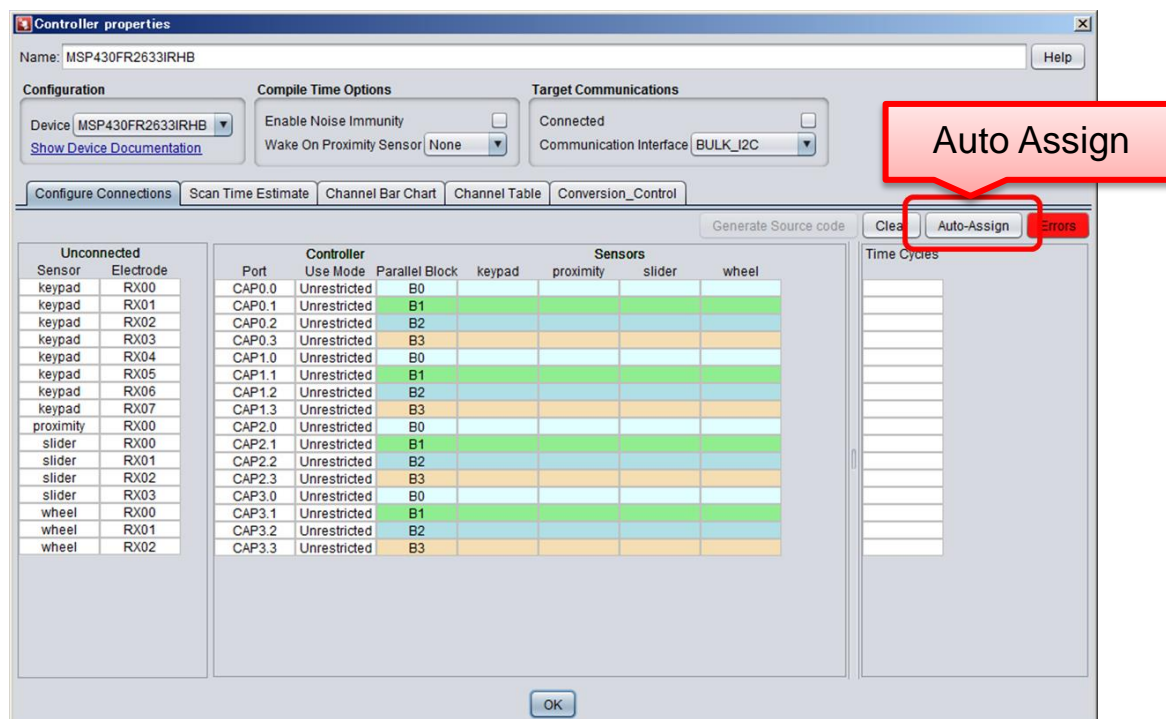
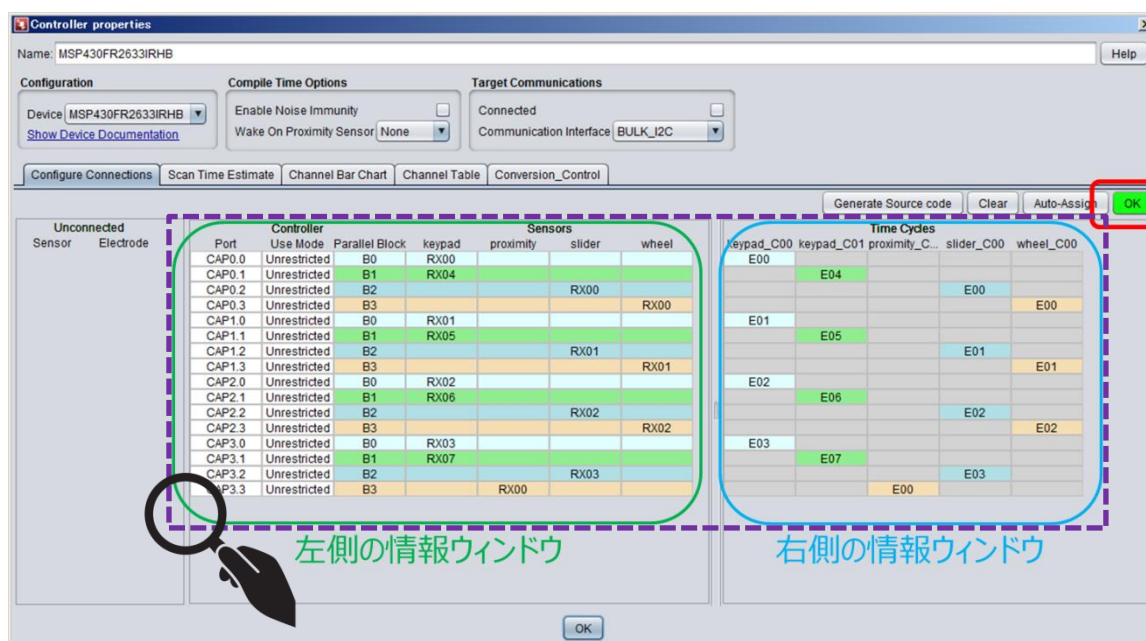


図 23 マイコン設定(センサー接続設定)

正しく接続された場合、以下のように OK(緑)のインジケータが確認できます。



Controller				Sensors			Time Cycles				
Port	Use Mode	Parallel Block	keypad	proximity	slider	wheel	keypad_C00	keypad_C01	proximity_C...	slider_C00	wheel_C00
CAP0.0	Unrestricted	B0	RX00				E00	E04		E00	
CAP0.1	Unrestricted	B1	RX04				E01	E05		E01	
CAP0.2	Unrestricted	B2			RX00		E02	E06		E02	
CAP0.3	Unrestricted	B3				RX00	E03	E07		E03	
CAP1.0	Unrestricted	B0	RX01								E00
CAP1.1	Unrestricted	B1	RX05								E01
CAP1.2	Unrestricted	B2			RX01						E02
CAP1.3	Unrestricted	B3				RX01					E03
CAP2.0	Unrestricted	B0	RX02								
CAP2.1	Unrestricted	B1	RX06								
CAP2.2	Unrestricted	B2			RX02						
CAP2.3	Unrestricted	B3				RX02					
CAP3.0	Unrestricted	B0	RX03								
CAP3.1	Unrestricted	B1	RX07								
CAP3.2	Unrestricted	B2			RX03						
CAP3.3	Unrestricted	B3		RX00							

図 24 マイコン設定(センサー接続設定完了)

さあ、どのように接続されたのか確認してみましょう。

まず、左側の情報ウィンドウで、マイコンの IO ピンは CAP0.0 から CAP3.3 まで16個の存在が確認できます。各センサーの接続は次のようになっています。

keypad: CAP0.0,CAP0.1,CAP1.0,CAP1.1,CAP2.0,CAP2.1,CAP3.0,CAP3.1

proximity: CAP3.3

slider: CAP0.2,CAP1.2,CAP2.2,CAP3.2

wheel: CAP0.3,CAP1.3,CAP2.3

では、右側の情報ウィンドウは何を示しているのでしょうか。

実は、CapTIvate では「1サイクルで最大4個の IO をパラレルに同時検出」する仕組みを備えており、センサーノードスキャンの高速化を実現しています。

このウィンドウから、Keypad で2サイクル、proximity、slider、wheel で各1サイクル、結果として、5サイクルで16個のセンサーノードをスキャンするように設定されたことが読み取れます。

では、最下部の OK ボタンをクリックして、次に進みましょう。

## 4.10 ボードを接続

ここまでの手順で、CAPTIVATE-BSWP のプロジェクトが作成できました。  
では、以下のように、ボードと PC を接続してください。

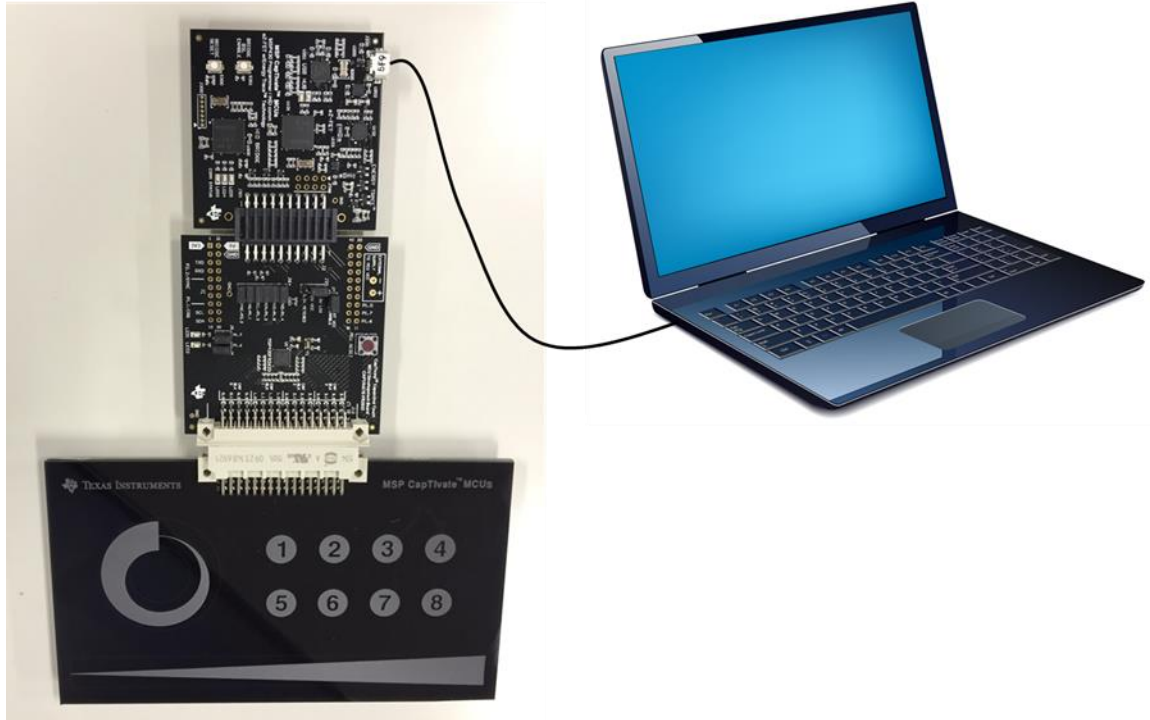
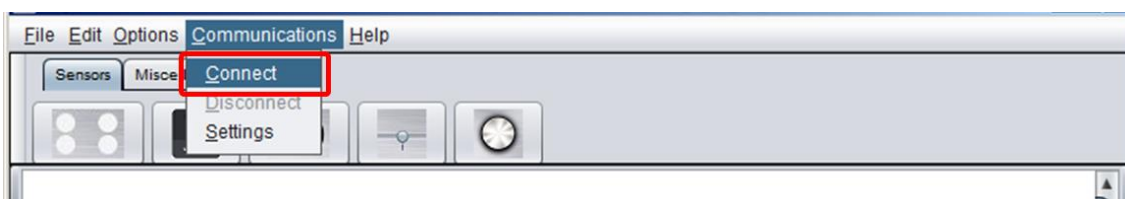


図 25 ボード接続

## 4.11 動作を確認

次に、Communication → Connect をクリックして接続しましょう。



ボタン、ホイール、スライダー、近接センサーの動作を確認してみてください。  
動作しましたか？

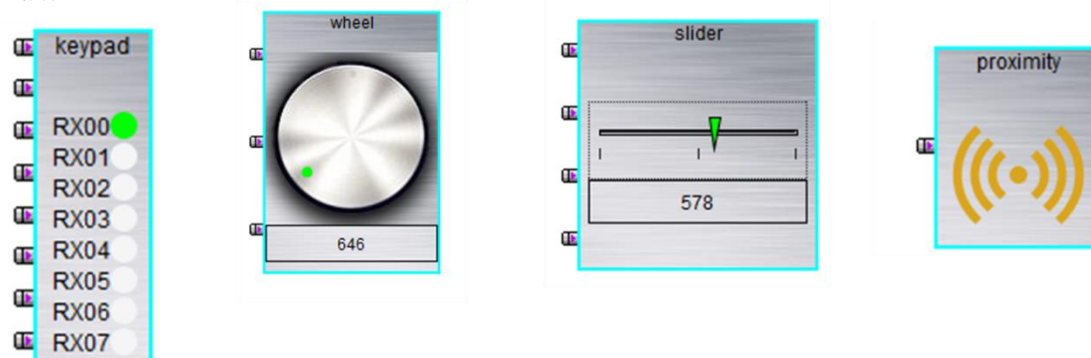
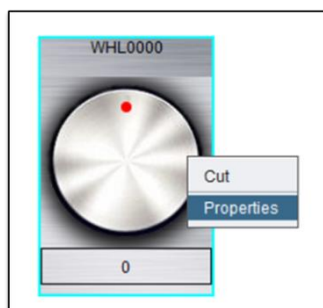


図 26 各種センサー

<補足>

実は、私の場合、ホイールがうまく動作しないという現象が見られましたが、以下のように Touch Threshold の調整で動作するようになりました。

ホイールの上で右クリックしプロパティビューを Open してください。



Tuning タブをクリックし、Touch Threshold を設定、ホイールの全てのセンサーノード(E00,E01,E02) を 20 程度に設定し、Apply をクリックし、OK をクリックしてプロパティビューを閉じてください。

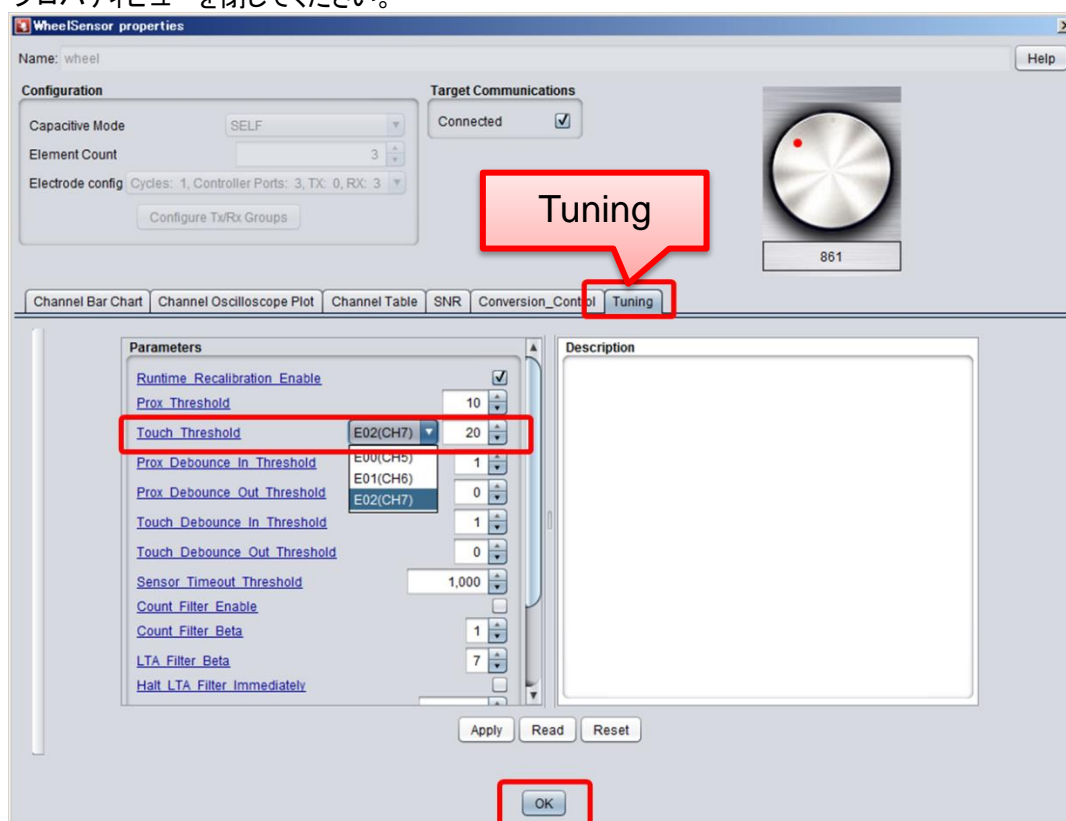


図 27 センサー設定(ホイール)

## 5. プログラムを作成してみよう

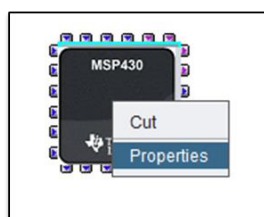
ここまでは評価キットのマイコンに書き込まれたプログラムを使って、基本的な機能の確認を行ってきましたが、いよいよ、自作のプログラムを作成してみましょう。

まず、GUI ツールでプログラムを自動生成し、ボードへ書き込むまでの手順から説明します。

### 5.1 ベースコードを自動生成

では、コードの自動生成から始めましょう。

マイコンの上で右クリックしプロパティビューを Open してください。



Generate Source code ボタンをクリックします。

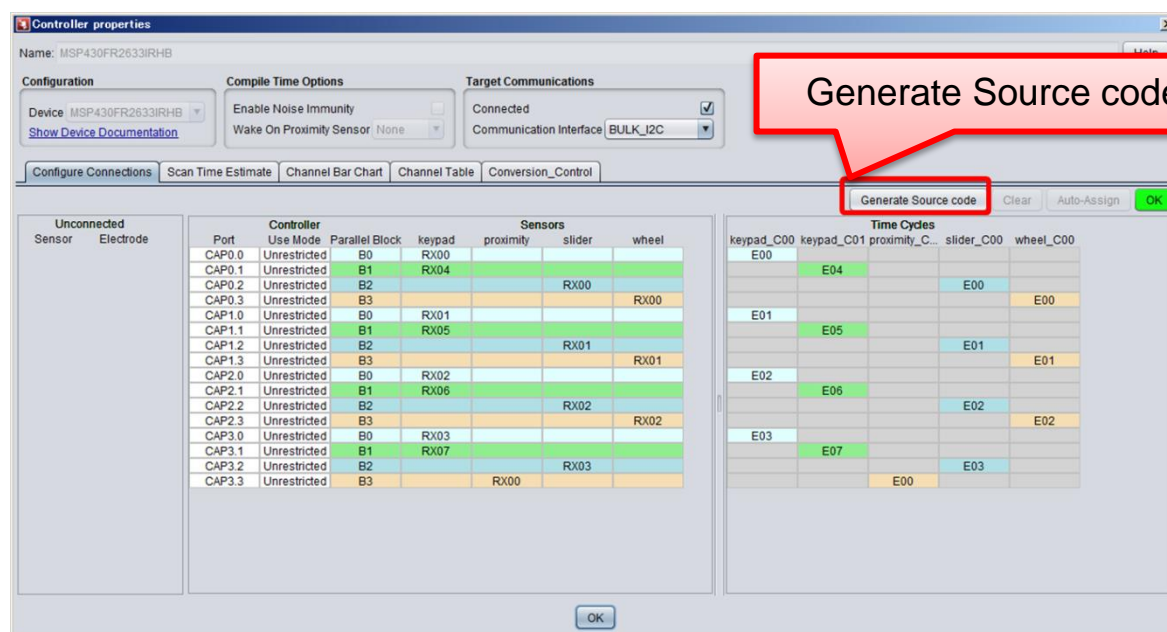


図 28 コード自動生成



Create new project を選択し OK ボタンをクリックしてください。

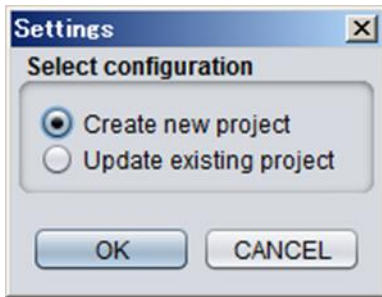


図 29 Setting のポップアップ

格納先ディレクトリを指定し、OK ボタンをクリックしてください。



図 30 File の選択

これでプログラムが自動生成され、上記指定のディレクトリに格納されました。

## 5.2 コンパイル環境の準備

では、生成されたコードをコンパイルしてマイコンに書き込みましょう。

TI のマイコン、プロセッサのプログラム、コンパイル、デバッグには、CodeComposerStudio (以降 CCS と記載) というツールが用意されています。

(無償でご使用いただけます。)

### 5.2.1 Code Composer Studio(CCS)をダウンロード

今回のマイコン(MSP430FR2633)も CCS に対応していますので、CCS を使用します。

では、以下のサイトから CCS をダウンロードしましょう。

[http://processors.wiki.ti.com/index.php/Download\\_CCS](http://processors.wiki.ti.com/index.php/Download_CCS)

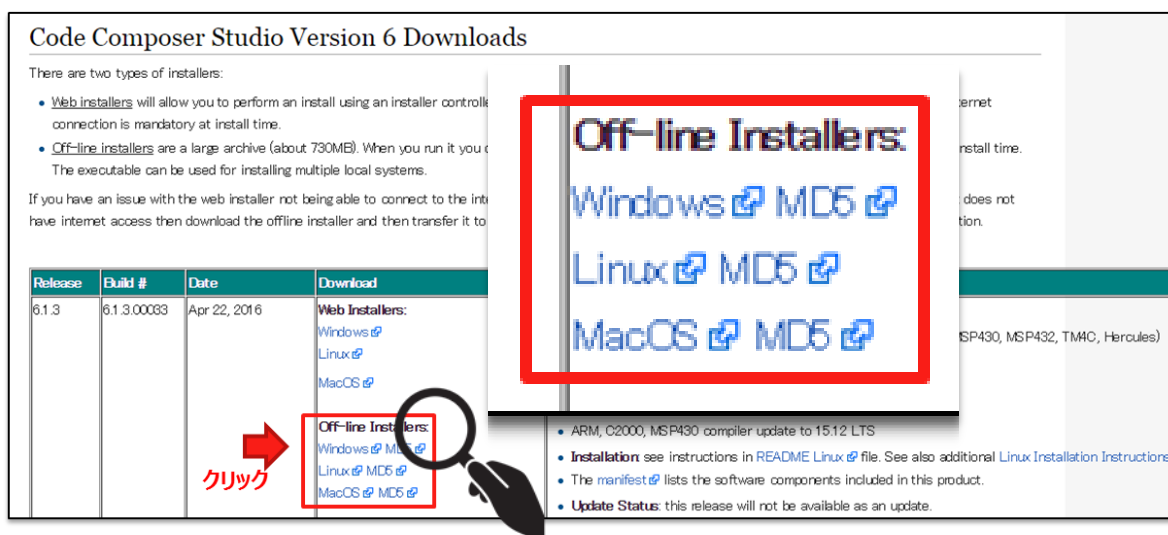


図 31 ダウンロードサイト

ご希望の OS に対応した最新バージョンのツールをダウンロードしてください。

CCS 等、各種ソフトウェアはご案内の URL より入手可能ですが、myTI にご登録頂く必要がございます。  
※myTI に関する情報や登録方法は下記の URL をご参照ください。

[http://www.tij.co.jp/lstds/ti\\_ja/general/myTI/overview.page](http://www.tij.co.jp/lstds/ti_ja/general/myTI/overview.page)

### 5.2.2 Code Composer Studio(CCS)を起動

Code Composer Studio(CCS)を起動しましょう

デスクトップにある CCS のアイコンを実行してください。

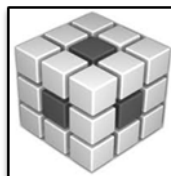


図 32 CCS アイコン



### 5.2.3 作業ディレクトリ(workspace)を指定

CCS が起動したら作業ディレクトリ(workspace)のディレクトリを指定します。  
今回は、C:\%capture\_workspace と指定して、OK をクリックしました。

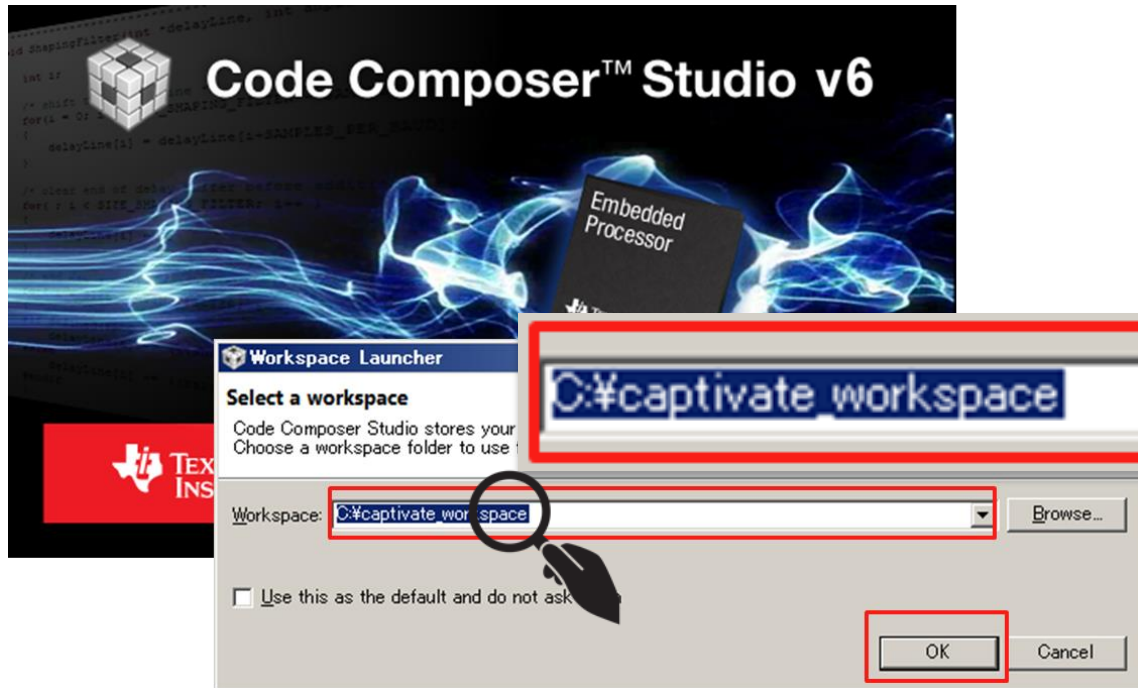


図 33 CCS 起動画面

### 5.2.4 Update を確認

Update を確認します。(初回以降は、必要に応じて実行してください)  
Help -> Check for Update を選択します。

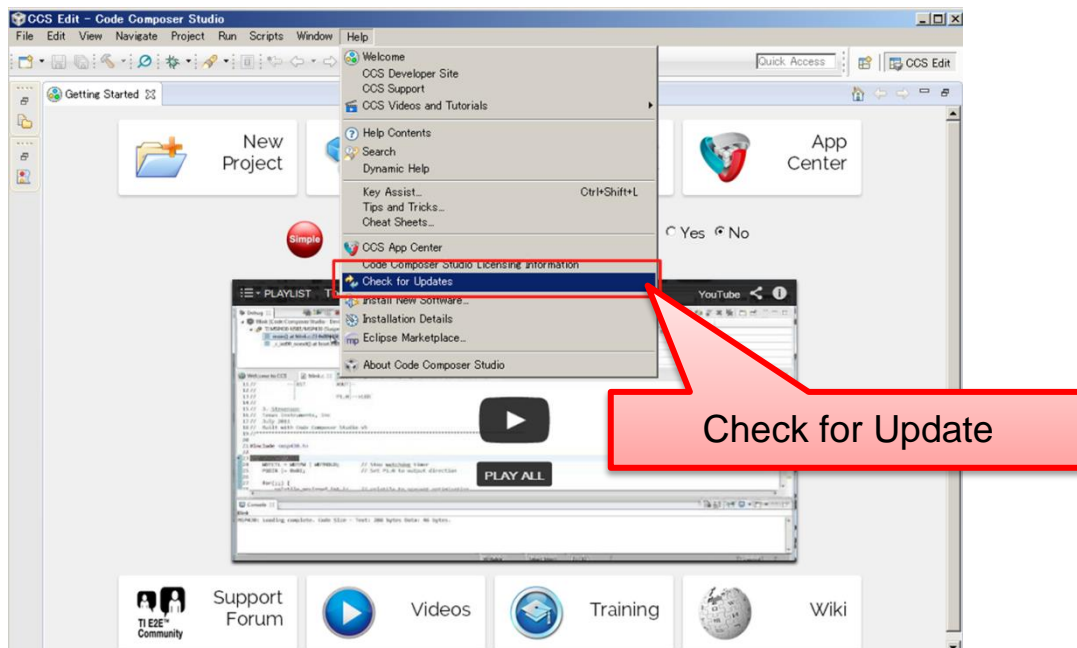


図 34 CCS Update 画面

## 5.3 プロジェクトをインポート

GUI ツールで自動生成されたプログラムを読み込みましょう。  
project->Import CCS Projects を選択します。

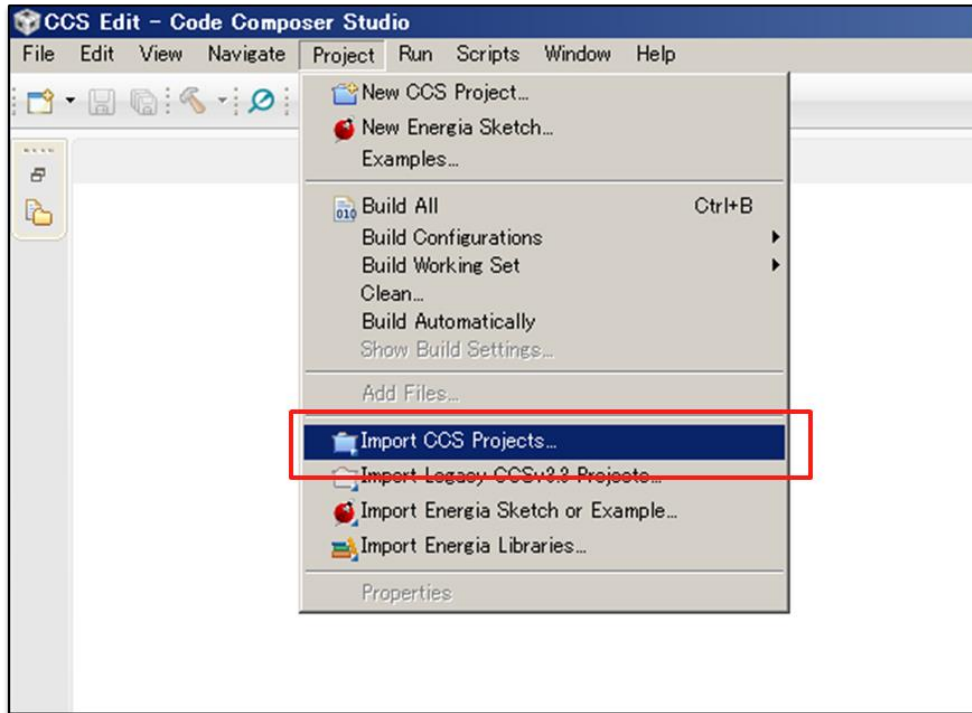


図 35 CCS インポート画面

次に、Browse ボタンをクリックして、GUI ツールの "Generate Source code" で指定したフォルダを選択し、読み込むプロジェクト (今回は "BSWP\_test1") をチェックし、Finish をクリックします。

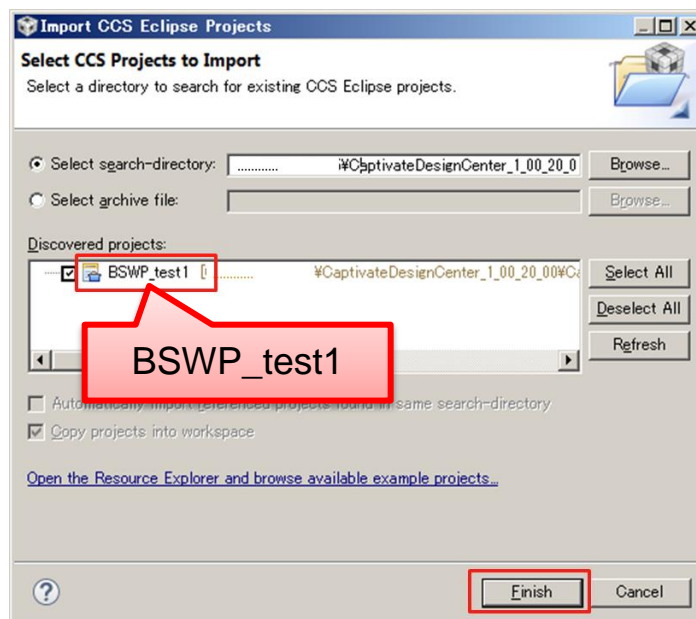


図 36 File の選択

これでプロジェクトの読み込みができました。

## 5.4 ソースコードをコンパイル

ではソースコードをコンパイルしましょう。



プロジェクトを選択して、ハンマーのアイコン(Build) をクリックしてください。



上記のように、エラーなしで"Build Finished"が確認できたらコンパイル成功です。

図 37 CCS プロジェクト画面

## 5.5 ボードにプログラムを書き込んで実行

ボードにプログラムを書き込んで動作をさせてみましょう。  
ボードと PC が USB ケーブルで接続されていることを確認した後、



虫のアイコン(Debug) をクリックしてください。



図 38 デバッグアイコン

もし、途中で、以下のようなメッセージボックスが出現した場合、Update を押してください。

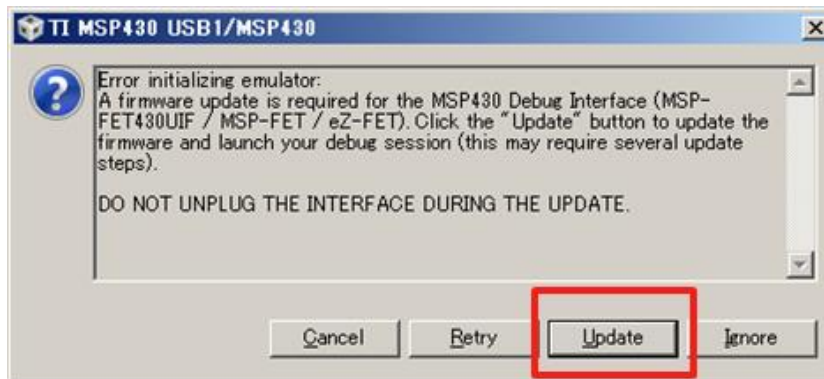


図 39 エラーメッセージ

成功すると、以下のようなデバッグウィンドウが表示されます。

三角のアイコン(Resume) をクリックすると、プログラムの実行が開始されます。

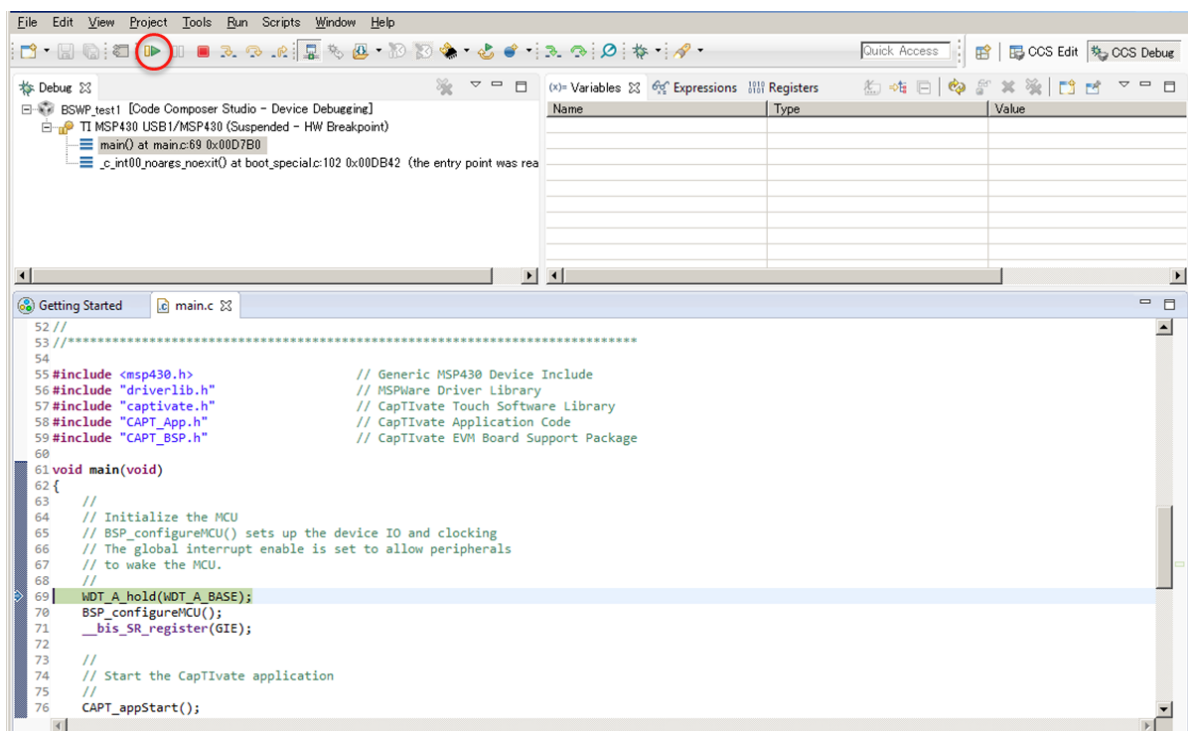


図 40 プログラム実行



黄色のアイコン(Suspend)をクリックすると、プログラムが一時停止します。  
(以下のように、停止した行がハイライトされます。)

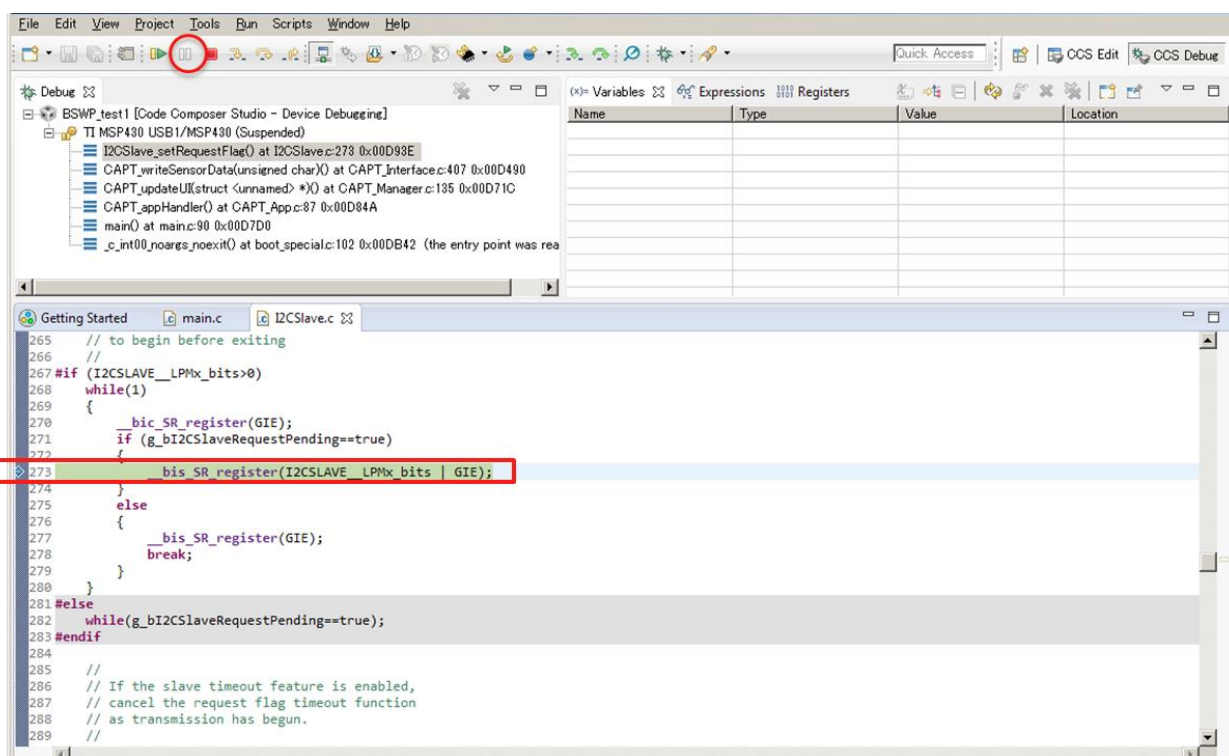


図 41 プログラム一時停止

プログラムはマイコンの不揮発メモリ(FRAM)に書き込まれたので、デバッグを終了しましょう。



赤色のアイコン(Stop)をクリックしてください。

File -> Exit CCS を終了してウィンドウを閉じてください。

## 5.6 動作を確認

では、GUI ツールでタッチセンサーの動作を確認してみましょう。

デスクトップにある CapTIvate™ Design Center のアイコンを実行してください。



図 42 アイコン



次にプロジェクトを読み込みましょう。

- ・メニューバーの File → Project Open をクリックしてください。
- ・BSWP\_test1 を選択して Select を押してください。

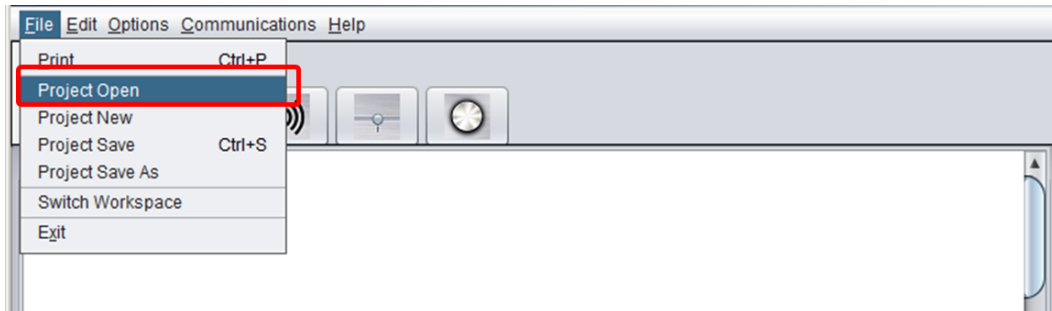


図 43 プロジェクトロード画面

次に、Communication → Connect をクリックして接続しましょう。



ボタン、ホイール、スライダー、近接センサーの動作を確認してみてください。  
動作しましたか？

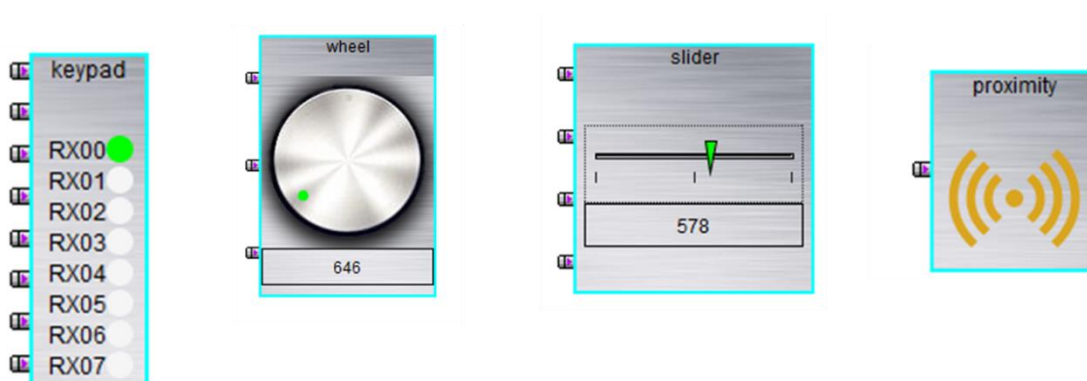


図 44 各種センサー

## 5.7 プログラムを作成

ここまでで、GUI ツールでプログラムを自動生成し、ボードに書きこんで動作することを確認しました。  
今回は、実際にコードを書いていきましょう。

デスクトップにある CCS のアイコンを実行してください。

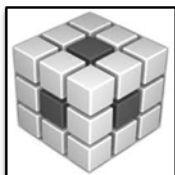


図 45 CCS アイコン

CCS が起動したら作業ディレクトリ(workspace)のディレクトリを指定します。C:\%captivate\_workspace と指定して、OK をクリックしてください。



図 46 CCS 起動画面

今回は、次のような処理を追加します。

- ・スライダーが 500 を超えたら、LED1 を点灯
- ・スライダーが 500 以下なら、LED1 を消灯

そのための手順は以下の3つになります。

- ・既存のプログラムから、LED1 の点灯処理部を削除
- ・スライダーにタッチ検出時、処理を呼び出す API を追加
- ・スライダーのタッチ検出後の処理を追加



## 5.7.1 既存のプログラムから、LED1 の点灯処理部を削除

では、まず LED1 の点灯処理部の削除を行いましょう。

- main.c から、以下のように LED 制御処理部を削除(コメントアウト)してください。  
(後程、別の用途で使いますので、LED2 の処理も削除してしまいます。)
- CAPT\_appHandler(); を忘れずに移動しておきます。

```
void main(void)
{
    WDT_A_hold(WDT_A_BASE);
    BSP_configureMCU();
    __bis_SR_register(GIE);
    //
    // Start the CapTivate application
    //
    CAPT_appStart();
    while(1)
    {
        /*
            LED1_ON;
            if(CAPT_appHandler()==true)
                LED2_ON;
            else
                LED2_OFF;
            LED1_OFF;
        */

        CAPT_appHandler();

        __no_operation();
        CAPT_appSleep();

    } // End background loop
} // End main()
```

図 47 プログラム

## 5.7.2 スライダーにタッチ検出時、処理を呼び出す API を追加

MAP\_CAPT\_registerCallback(&slider, &my\_slider\_callback);  
という API を main の CAPT\_appStart(); の前に追加します。

```
void main(void)
{
    WDT_A_hold(WDT_A_BASE);
    BSP_configureMCU();
    __bis_SR_register(GIE);

    MAP_CAPT_registerCallback(&slider, &my_slider_callback);

    //
    // Start the CapTivate application
    //
    CAPT_appStart();
    while(1)
    {
        /*
            LED1_ON;
            if(CAPT_appHandler()==true)
                LED2_ON;
            else
                LED2_OFF;
            LED1_OFF;
        */

        CAPT_appHandler();

        __no_operation();
        CAPT_appSleep();

    } // End background loop
} // End main()
```

図 48 プログラム

### 5.7.3 スライダーのタッチ検出後の処理を追加

my\_slider\_callback(tSensor\* pSensor) を、main の前に追加します。

```
void my_slider_callback(tSensor* pSensor)
{
    tSliderSensorParams *pSliderParams;
    if (pSensor->bSensorTouch == true)
    {
        pSliderParams = (tSliderSensorParams*)(pSensor->pSensorParams);
        if (pSliderParams->SliderPosition.ui16Natural > 500)
        { LED1_ON;}
        else
        { LED1_OFF;}
    }
}
```

slider 値が 500 を超えたら LED1=ON

slider 値が 500 以下なら LED1=OFF

```
void main(void)
{
    WDT_A_hold(WDT_A_BASE);
    BSP_configureMCU();
    __bis_SR_register(GIE);
    MAP_CAPT_registerCallback(&slider, &my_slider_callback);

    //
    // Start the CapTIvate application
    //
    CAPT_appStart();
    while(1)
    {
        /*
            LED1_ON;
            if(CAPT_appHandler()==true)
                LED2_ON;
            else
                LED2_OFF;
            LED1_OFF;
        */

        CAPT_appHandler();

        __no_operation();
        CAPT_appSleep();

    } // End background loop
} // End main()
```

図 49 プログラム

## 5.8 ボードにプログラムを書き込んで実行

ボードにプログラムを書き込んで動作をさせてみましょう。  
ボードと PC が USB ケーブルで接続されていることを確認した後、

虫のアイコン(Debug) をクリックしてください。  
(実は、このボタンは、build も実行してくれますので、  
今回は、ハンマーのアイコンはクリックしなくても OK です。)



図 50 デバッグアイコン

## 5.9 動作を確認

では、実行して、動作を確認してみましょう。  
スライダーでタッチが検出されると、my\_slider\_callback(tSensor\* pSensor)  
の処理が実行されるはずですから、  
**if (pSliderParams->SliderPosition.ui16Natural > 500)**  
にブレークポイントを設定して、実行してみます。  
ブレークポイントを設定したい行の左の部分(以下)をダブルクリックしてみてください。  
丸い記号でマーキングされます。

三角のアイコン(Resume) をクリックすると、プログラムの実行が開始されます。

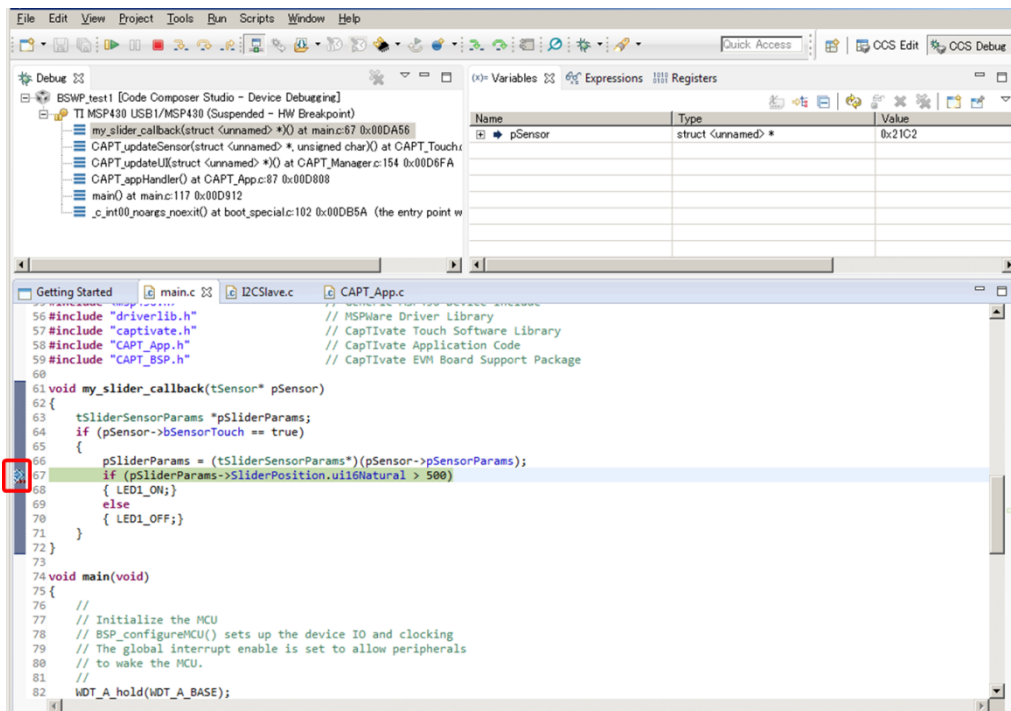


図 51 プログラム実行

スライダーをタッチしてみてください。

スライダーがタッチを検出し、先ほど設定したブレークポイントでのプログラム停止が確認できます。

次に、ブレークポイントを解除して、動作の確認を進めましょう。

先ほどのブレークポイントの丸いマーカをダブルクリックして、ブレークポイントを解除してください。

三角のアイコン(Resume) をクリックして、プログラムを再実行しましょう。



スライダーの上で指を左右に移動させてみてください、スライダーの中心付近で LED1 の点灯、消灯が確認できます。

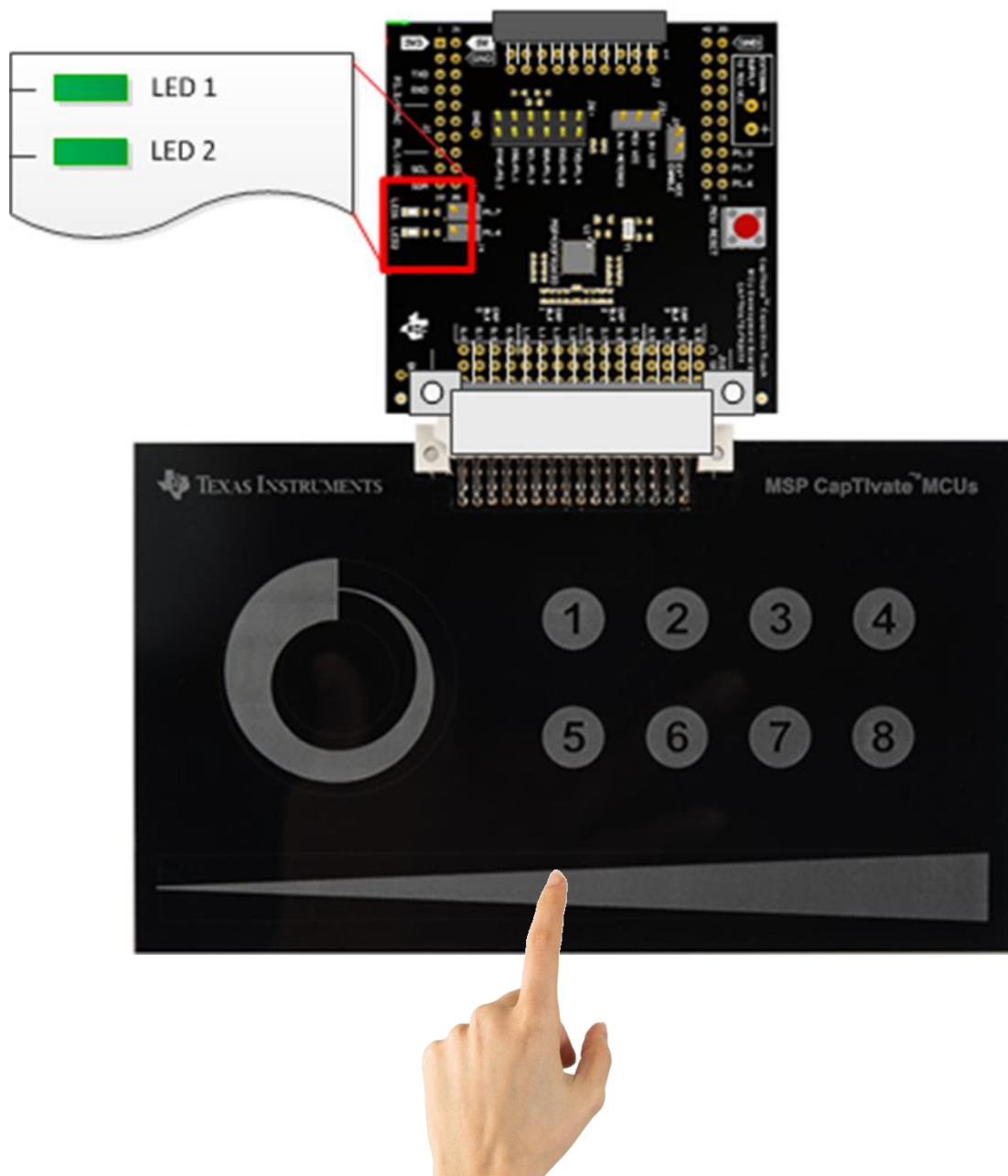


図 52 スライダー動作確認

GUI ツールを起動し、Communication → Connect をクリックして接続すると、スライダーの値が 500 で LED1 が消灯、501 で LED1 が点灯することが確認できます。

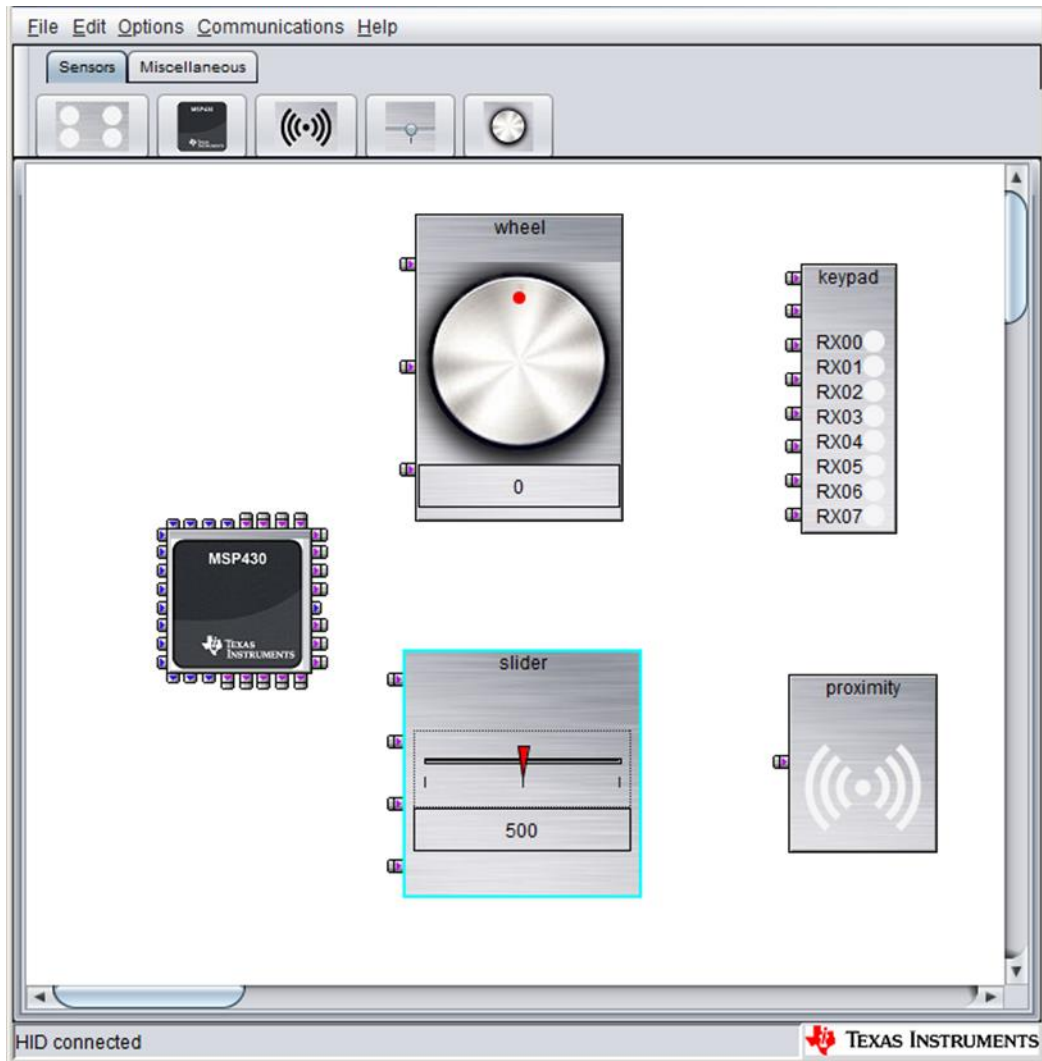


図 53 キャンバス

## 6. 参考プログラム

参考として、ボタンのタッチを検出したら LED2 を点灯するように修正したコードを紹介します。  
 以下のようにプログラムを修正して、動作を確認してみてください。  
 (詳細説明は、割愛させていただきます。)

```
void my_keypad_callback(tSensor* pSensor)
{
    if ((pSensor->bSensorTouch == true) && (pSensor->bSensorPrevTouch == false))
    {
        LED2_ON;
    }
    else if ((pSensor->bSensorTouch == false) && (pSensor->bSensorPrevTouch == true))
    {
        LED2_OFF;
    }
}

void my_slider_callback(tSensor* pSensor)
{
    tSliderSensorParams *pSliderParams;
    if (pSensor->bSensorTouch == true)
    {
        pSliderParams = (tSliderSensorParams*)(pSensor->pSensorParams);
        if (pSliderParams->SliderPosition.ui16Natural > 500)
        { LED1_ON;}
        else
        { LED1_OFF;}
    }
}

void main(void)
{
    WDT_A_hold(WDT_A_BASE);
    BSP_configureMCU();
    __bis_SR_register(GIE);

    MAP_CAPT_registerCallback(&keypad, &my_keypad_callback);

    MAP_CAPT_registerCallback(&slider, &my_slider_callback);
    // Start the CapTivate application
    CAPT_appStart();
    // Background Loop
    while(1)
    {
        CAPT_appHandler();
        __no_operation();
        CAPT_appSleep();
    } // End background loop
} // End main()
```




図 54 プログラム





## 7. 安全上の注意



ここに示した注意事項は、お使いになる人や、他の人への危害、財産への損害を未然に防ぐための内容を記載していますので、必ずお守りください。

### 7.1 凡例

 <b>危険</b>	この表示は、取り扱いを誤った場合、「死亡また重傷を負う危険が切迫して生じることが想定される」内容です。
 <b>警告</b>	この表示は、取り扱いを誤った場合、「死亡または重傷を負う可能性が想定される」内容です。
 <b>注意</b>	この表示は、取り扱いを誤った場合、「傷害を負う可能性が想定される場合および物的損害のみの発生が想定される」内容です。

### 7.2 注意事項

 <b>危険</b>	本技術資料で指定された仕様に基づく AC アダプタ (梱包品) を使用してください。指定の仕様を満たさない AC アダプタを使用した場合は、製品の発熱、破裂、発火の原因となります。
 <b>警告</b>	強い衝撃を与えたり、投げつけたりしないでください。 発熱、破裂、発火や機器の故障、火災の原因となります。
	電子レンジなどの加熱調理機器や高压容器に、製品や AC アダプタを入れないでください。製品や AC アダプタの発熱、破裂、発火、発煙、部品の破壊、変更などの原因となります。
	使用中の製品を布など熱のこもりやすいもので包んだりしないでください。 熱がこもり、発火、故障の原因となることがあります。
	製品を廃棄する時は、他の一般ゴミと一緒に捨てないで下さい。 火中に投げると破裂する恐れがあります。廃棄方法については、廃棄物に関する各種法律・法令・条例等に従ってください。
	極端な高温、低温、また温度変化の激しい場所で使用しないで下さい。 故障の原因となります。周囲温度は 5℃ ～ 35℃、湿度は 0% ～ 85% の範囲でご使用ください。
	電源コードを強く引っ張ったり、重いものを乗せたりしないでください。 電源コードを傷つけたり、破損したり、束ねたり、加工したりしないでください。 傷ついた部分から漏電して、火災・感電の原因になります。
	濡れた手で電源プラグを抜き差ししないでください。 感電による怪我や故障の原因になります。
	電源プラグはコンセントの奥までしっかりと差し込んでください。 しっかり差し込まないと、感電や発電による火災の原因となります。
	タコ配線を行ったり、AC アダプタの規格電圧以外の電源に接続したりしないでください。 故障や感電、発熱による火災の原因となります。

 <b>警告</b> (前項から継続)	<p>電源プラグのほこりを定期的に拭き取り、コンセント周辺のたまったほこりを取り除いてください。ほこりがたまったままで使用していると湿気などで、絶縁不良となり、火災の原因になります。</p> <p>電源プラグやコンセント周辺のほこりは、乾いた布で拭き取ってください。</p> <p>製品にコップや花瓶など、水や液体が入った容器を置かないでください。</p> <p>製品に水や液体が入ると、故障や感電の原因になります。水などをこぼした場合は、使用を中止し、電源を切って電源プラグを抜いてください。修理や技術的な相談は購入元へお問い合わせください。</p> <p>製品を使用する際は、防護メガネ、防護手袋などを装着し、身体を保護してください。</p>
 <b>注意</b>	<p>ぐらついた台の上や傾いた場所等、不安定な場所には置かないでください。</p> <p>落下して、けがや故障の原因になります。</p> <p>直射日光の強い場所や炎天下の車内など高温の場所で使用、放置しないでください。</p> <p>発熱、破損、発火、暴走、変形、故障の原因になります。また、機器の一部が熱くなり、火傷の原因となる場合もあります。</p> <p>製品を組み込んだ装置の保守中は、電源を抜いて作業してください。</p> <p>感電の危険性があります。</p> <p>製品に無理な力がかかるような場所に置かないでください。</p> <p>基板の変形により、基板の破損、部品の脱落、故障の原因となります。</p> <p>拡張ボードや他の周辺機器と一緒にお使いの場合には、それぞれ個別の取り扱い説明書をよく読んで適正にお使いください。</p> <p>本技術資料に記載されているもの、また別途動作を確認できていることを公表しているものの他は、特定の拡張ボードや周辺機器の相互動作は保証いたしかねます。</p> <p>製品を移動・接続するときは、電源スイッチを切ってください。</p> <p>電源をいれたまま移動・接続すると、故障や感電の原因になります。</p> <p>ベンジンやシンナーなど化学薬品を含んだ雑巾で手入れしないでください。</p> <p>製品が変質する可能性があります。科学雑巾を使用するときは、その注意書きに従ってください。</p> <p>製品を箱から取り出した際、機器製品に結露が発生した場合は、すぐに電源を入れないでください。</p> <p>製品を箱から取り出す際、冷えたボード製品が部屋の暖かい空気により結露が発生することがあります。</p> <p>結露があるまま電源を入れると、製品が破損したり、部品の寿命が短くなる場合があります。</p> <p>製品を取り出したら室温になじませてください。結露が発生した場合は、水滴が蒸発してから設置や接続を行ってください。</p> <p>カスタマイズ可能と明示している部分以外の分解、解体、改変、改造、再生はしないでください。</p>

## 8. 更新履歴

日付	版	更新概要
2016 年 5 月 30 日	1.0	<ul style="list-style-type: none"> <li>初版</li> </ul>
2017 年 5 月 23 日	1.1	<ul style="list-style-type: none"> <li>CCS のバージョン変更に伴う修正</li> </ul>
		<ul style="list-style-type: none"> <li></li> </ul>

### 免責、及び、ご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料の内容は、予告無く変更することがございますので、最新のものとすることをご確認の上ご使用下さい。
3. 本資料に記載された回路、デバイス、ファームウェア、ソフトウェア、及びこれらに付随する情報をお客様の機器に使用される場合には、お客様の責任において実施して下さい。これらの使用に起因するお客様若しくは第三者の損害に対して、当社は一切その責任を負いません。
4. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、下記までご一報いただければ幸いです。

株式会社マクニカ

Mpression 推進部

〒222-8561 横浜市港北区新横浜 1-6-3 HP: <https://service.macnica.co.jp/contact>

5. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
6. 製品をご使用になる場合は、各デバイス・メーカの最新資料もあわせてご利用ください。