

# Technical Note

TecStar

## Silicon Labs 社 EFM8/C8051 クイックスタートガイド

---

2015 年 6 月

株式会社 **マクニカ**  
テクスター カンパニー

#### 目次

1 はじめに .....	3
2 開発環境のご紹介 .....	4
2-1 ハードウェア .....	4
2-1-1 EFM8 Starter Kit .....	4
2-1-2 C8051 Development Kit .....	5
2-1-3 C8051 ToolStick .....	5
2-2 ソフトウェア .....	6
2-2-1 Simplicity Studio .....	6
3 各種ドキュメント・サンプルコードの入手先 .....	8
3-1 EFM8/C8051 のドキュメント .....	8
3-2 EFM8/C8051 のサンプルコード .....	9
3-3 EFM8 の API 情報 .....	10
4 ソフトウェア・インストール .....	11
4-1 Simplicity Studio のインストール .....	11
4-2 KEIL コンパイラのライセンス設定 .....	11
5 ハードウェア・セットアップ .....	12
5-1 EFM8 Starter Kit のセットアップ .....	12
5-2 C8051 Development Kit のセットアップ .....	12
5-3 C8051 ToolStick のセットアップ .....	13
6 使用方法 .....	14
6-1 サンプルコードを動かしてみる .....	14
6-2 デバッグ機能を使ってみる (Simplicity IDE) .....	19
6-3 消費電流を測定してみる (Energy Profiler) .....	21
6-4 ピン設定やペリフェラル設定を試みる (Configurator) .....	22
6-5 ピン設定やペリフェラル設定を試みる (Configuration Wizard 2) .....	24
6-6 Silicon Labs IDE (レガシーな IDE) から Simplicity IDE (新しい IDE) へ移行する方法 .....	26
改版履歴 .....	27
参考文献 .....	27

## 1 はじめに

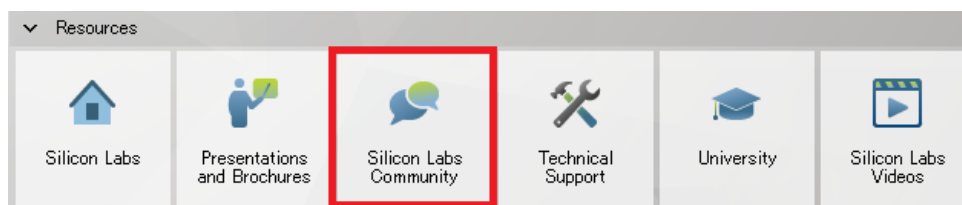
この資料は、Silicon Laboratories(以下、Silicon Labs)社製 MCU EFM8/C8051 ファミリの開発環境について簡易にまとめたものです。内容に誤りがないよう注意は払っておりますが、もし Silicon Labs 社が提供するドキュメント等と差異がございましたら、メーカー提供のものを優先してご参照ください。

また、Silicon Labs 社の ナレッジベース(FAQ)やコミュニティフォーラム(ユーザ同士で問題解決。Silicon Labs のエンジニアも頻繁にコメントしています)には、本資料で取り上げていない様々な情報が記載されております。

製品をご使用頂く過程で疑問や課題が生じることもあると思いますが、他のユーザが既に解決方法を見つけている場合も多々ございます。非常に有益ですので、ぜひご活用下さい。

### ◆ アクセス方法

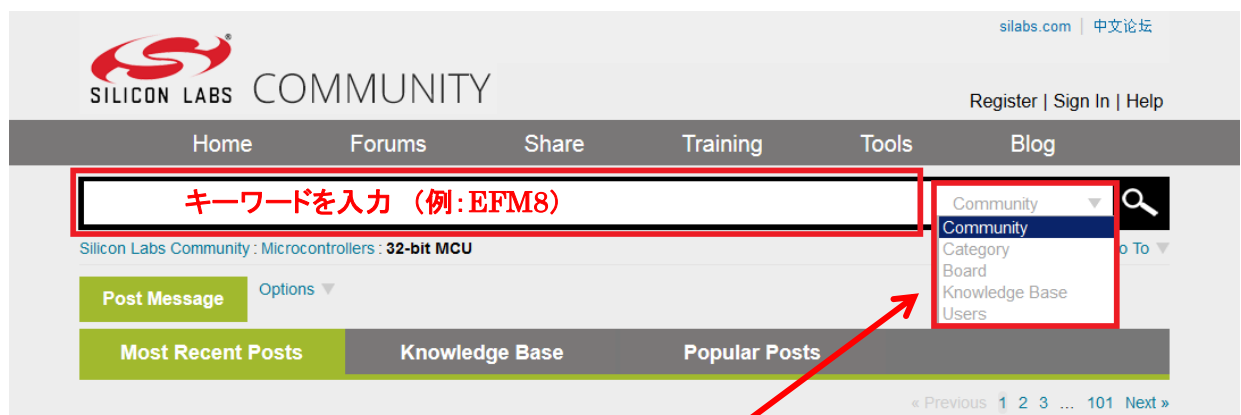
Simplicity Studio から



Web Site から

<http://community.silabs.com/t5/Forum/ct-p/Forum>

### ◆ 使用方法



Community か Knowledge Base を選択

## 2 開発環境のご紹介

EFM8/C8051 の開発環境について、ハードウェアとソフトウェアに分けてご紹介します。

### 2-1 ハードウェア

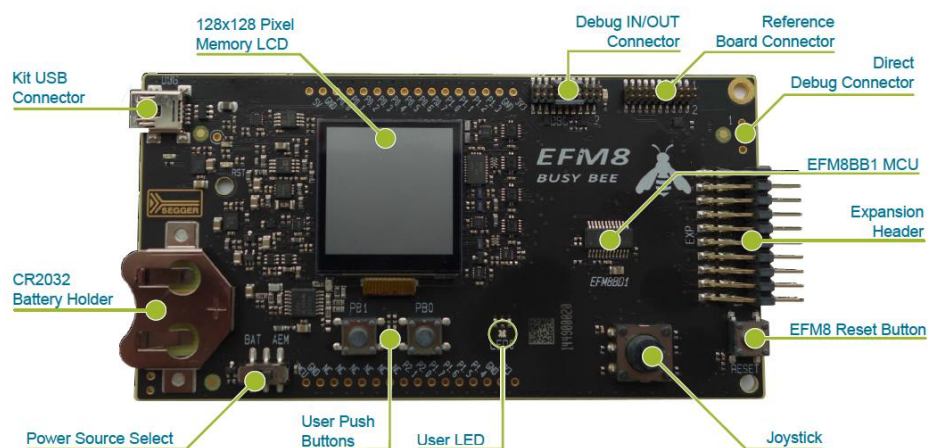
開発環境としては、EFM8 用に Starter Kit を、C8051 用に Development Kit と ToolStick を用意しています。Starter Kit と ToolStick は小型サイズで、お手軽にご評価頂けます。

#### 2-1-1 EFM8 Starter Kit

Starter Kit は、各ファミリに 1 種ずつ用意されています。同一ファミリであっても、ROM/RAM サイズやペリフェラルの数に差異がありますが、Starter Kit にはフルセットの MCU が実装されていますので、これを用いて設計を進めて頂くことが可能です。

ファミリ名	形名	Starter Kit	実装されている型番
Busy Bee	EFM8BB1	SLSTK2020A	EFM8BB10F8G
Busy Bee	EFM8BB2	SLSTK2021A	EFM8BB22F16G
Sleepy Bee	EFM8SB1	SLSTK2010A	EFM8SB10F8G
Sleepy Bee	EFM8SB2	SLSTK2011A	EFM8SB20F64G
Universal Bee	EFM8UB1	SLSTK2000A	EFM8UB10F16G
Universal Bee	EFM8UB2	SLSTK2000A	EFM8UB20F64G

#### ◆ EFM8BB1 : SLSTK2020A



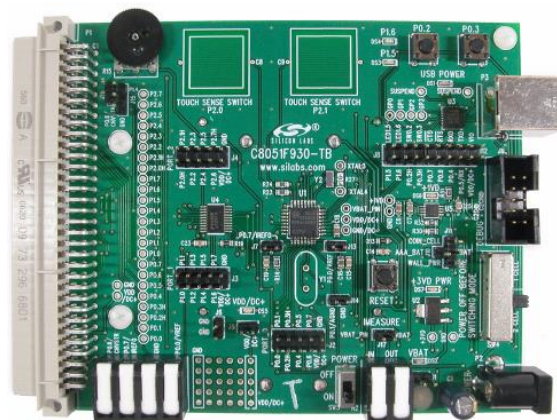
## 2-1-2 C8051 Development Kit

ベーシックな評価基板が入った開発キットです。ターゲットボード、USB Debug Adaptor(PC とターゲットボードとをつなぐ機材)、AC/DC アダプタが同梱されています。MCU ファミリごとに Development Kit が用意されており、ターゲットボードに実装されている MCU が異なります。

ターゲットボードには LED やスイッチが実装されている他、アナログ入力やタッチボタンなど、各 MCU ファミリが持つ特徴的な機能を評価できる作りになっています。また、全ての I/O ピンが引き出されていますので、拡張性にも富んでいます。



Development Kit

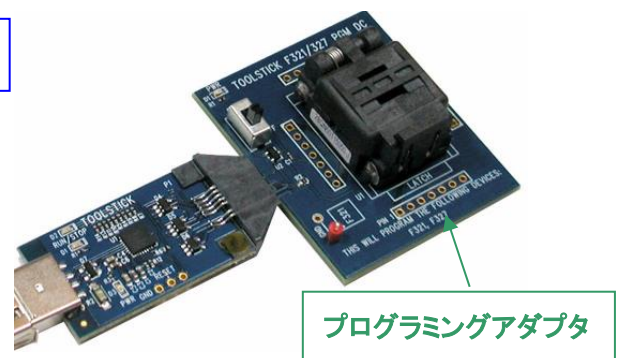
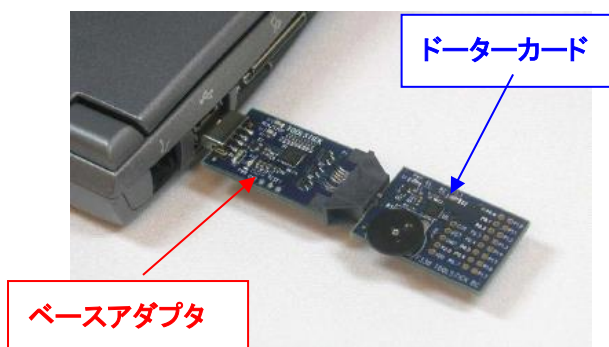


ターゲットボード

## 2-1-3 C8051 ToolStick

USB 給電で動作する、非常にコンパクトな評価基板です。PC に繋がるベースアダプタ、評価対象を搭載したドーターカード、の 2 つで構成されています。

ドーターカードは、MCU ファミリごと、パッケージごとに多種用意しており、評価したい MCU に最適なドーターカードをお選び頂くことができます。またソケットが載ったプログラミングアダプタも用意しており、プログラミングにご使用頂けます。



## 2-2 ソフトウェア

EFM8/C8051 の開発環境である Simplicity Studio を使用して設計を行うこととなります。C・アセンブラのコンパイラについては、KEIL 社のコンパイラ(フルライセンス)を無償提供しています。

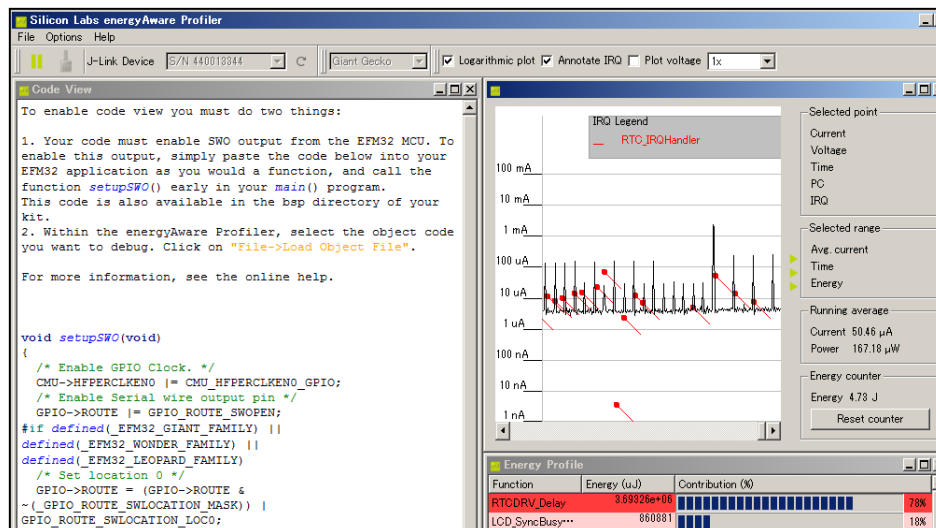
### 2-2-1 Simplicity Studio

Simplicity Studio は、EFM8/C8051 をターゲットとしたコンパイル・デバッグ・プログラミングを 1 つのプラットフォームで提供することができるソフトウェアです。統合開発環境 (IDE) を中心に、非常に便利なツール群が充実しています。同社製の 32bit MCU や無線 MCU も同一プラットフォームで開発が可能です。



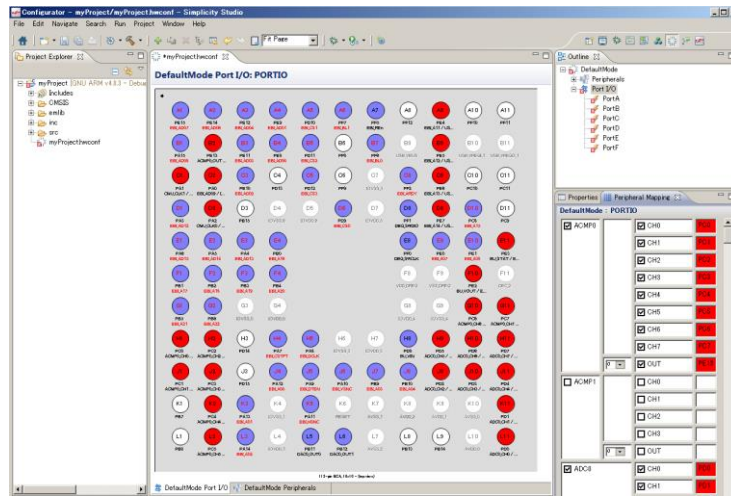
ツール名	機能の概要
Simplicity IDE	統合開発環境 (IDE)。無償の KEIL 社コンパイラを搭載
Energy Profiler	実機の消費電流値を測定することが可能。EFM8 に対応
Configurator	ピン設定やペリフェラル設定を簡単に行うことができる。新しい製品 (EFM8、C8051 の一部) に対応
Configuration Wizard 2	ピン設定やペリフェラル設定を簡単に行うことができる。レガシー製品 (C8051) に対応
Flash Programmer	フラッシュ ROM のライト/イレース

#### ◆ 消費電流が実測できます (Energy Profiler)

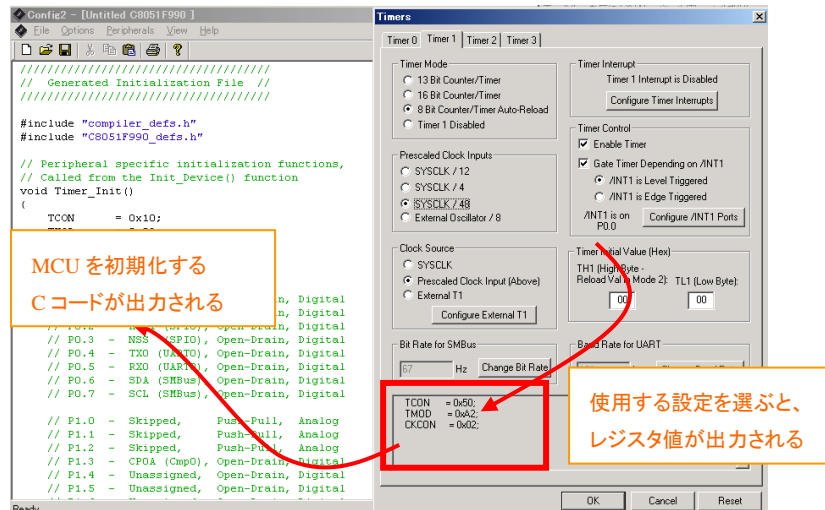




◆ ピン設定やペリフェラル設定を簡単に行えます (Configurator)



◆ ピン設定やペリフェラル設定を簡単に行えます (Configuration Wizard 2)



## 3 各種ドキュメント・サンプルコードの入手先

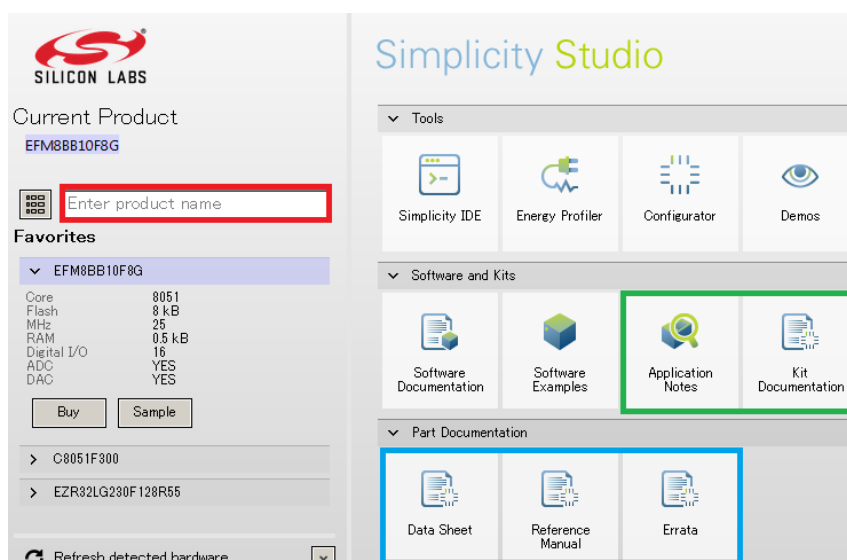
EFM8/C8051 の最新ドキュメント・サンプルコードの入手方法について紹介します。

### 3-1 EFM8/C8051 のドキュメント

EFM8/C8051 のドキュメントは、Simplicity Studio を経由してご入手頂けます。Simplicity Studio の入手先やインストール方法につきましては、「4.ソフトウェア・インストール」をご参照ください。

Simplicity Studio を起動し、Product の欄(下図の赤枠)に使用する製品型番を入力すると、青枠・緑枠から下記情報をご入手頂けるようになります。

- ◆ Datasheet: データシート。スペック、ピン配置情報など
- ◆ Reference Manual: リファレンス・マニュアル。ペリフェラルの使用方法などの解説
- ◆ Errata : エラッタ(バグ情報)
- ◆ Application Notes: アプリケーションノート、それに付随するサンプルコード
- ◆ Kit Documentation: Starter Kit のユーザガイド、回路情報



また、ドキュメントは Silicon Labs 社の Web Site からのご入手可能です。

<http://www.silabs.com/support/pages/document-library.aspx>

製品型番を入力すると、関連ドキュメントがリストアップされます。

#### Document Library

The document library has all of Silicon Labs' technical documents conveniently located in one place. To find the documents you need, start by selecting one of the product categories below.

**Product Line**  
Please select a Product Line

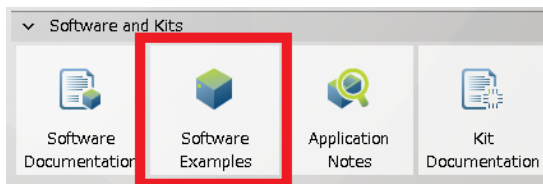
**Family**  
All Families

**Part Number**  
All Part Numbers

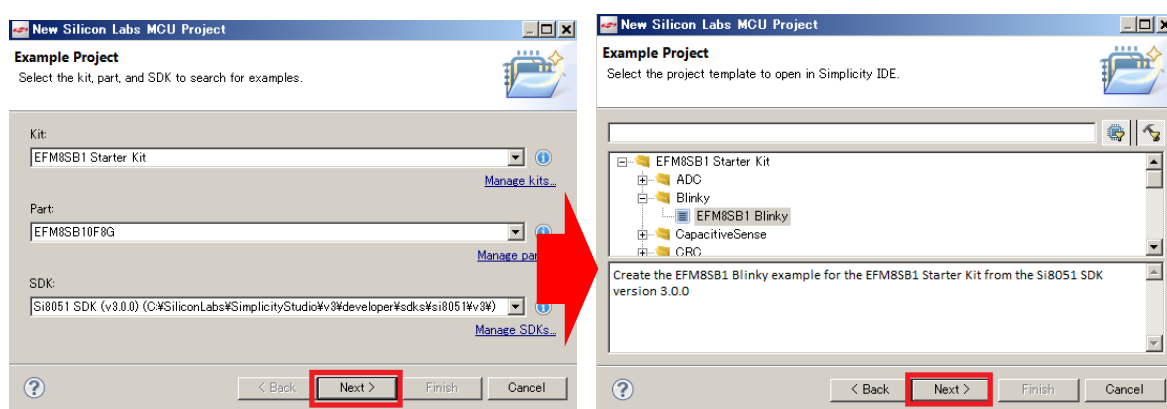


### 3-2 EFM8/C8051 のサンプルコード

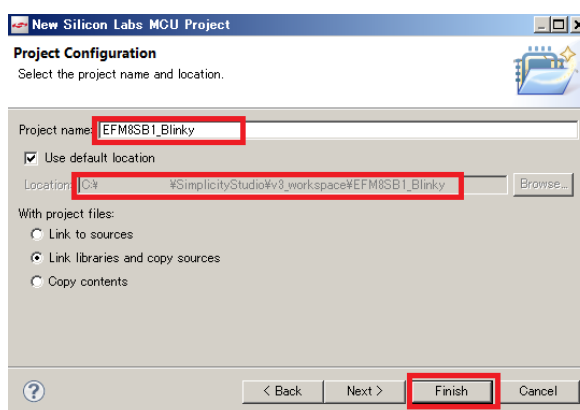
EFM8/C8051 のサンプルコードは、Simplicity Studio を経由してご入手頂けます。Simplicity Studio を起動し、Product の欄に使用する製品型番を入力して、Software Examples をクリックします。



Example Project ウィンドウにて、Kit と Part に、使用する評価キットと MCU 型番が選択されていることを確認し、Next ボタンをクリックします。評価キットをあらかじめ接続しておく、自動で入力されます。次に、生成したいサンプルコードを選択し、Next をクリックします。



プロジェクト名を入力し、サンプルコードの生成フォルダを確認して、Finish ボタンをクリックします。プロジェクトとサンプルコードが生成されます。

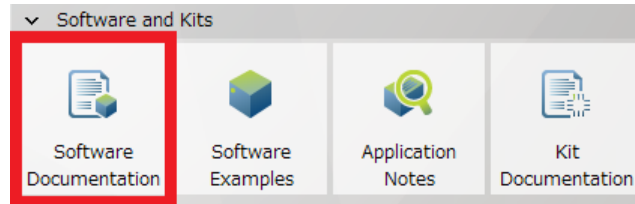


生成が完了すると、プロジェクトが自動でロードされ、コンパイルが行える状態になります。

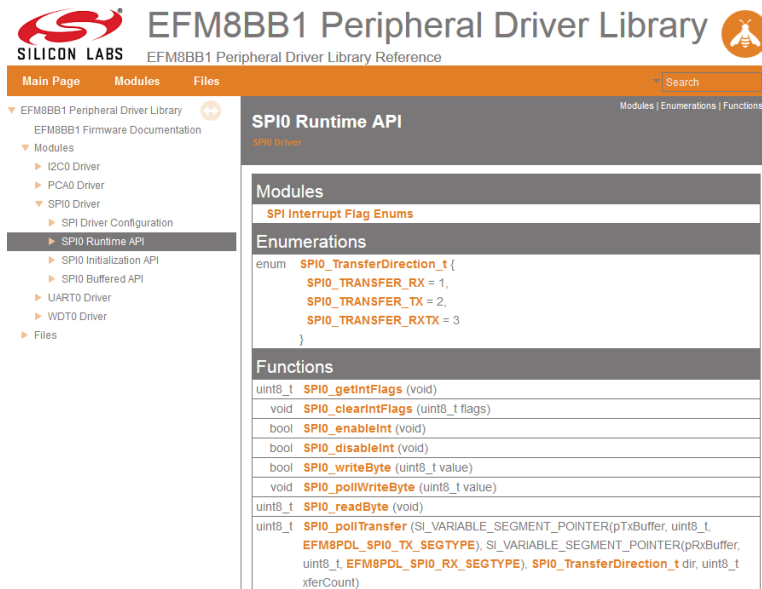
### 3-3 EFM8 の API 情報

EFM8 には、ペリフェラルを制御するためライブラリ(API)が用意されており、それを使用することでソフト設計を円滑に進めて頂くことが可能です。C8051 については用意されていません。

Simplicity Studio を起動し、Product の欄に使用する製品型番を入力して、Software Documentation をクリックします。



ブラウザが起動し、API 情報が表示されます。



## 4 ソフトウェア・インストール

EFM8/C8051 の評価に必要なソフトウェアをインストールします。

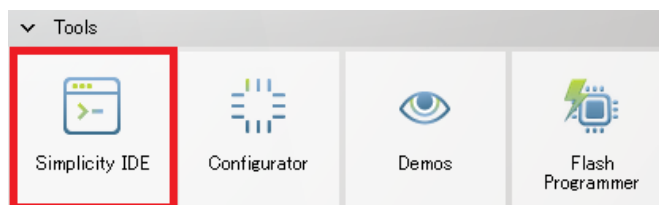
### 4-1 Simplicity Studio のインストール

Simplicity Studio は、下記アドレスからダウンロードして頂けます。ダウンロード後、install-studio.exe を実行して、インストールを行います。

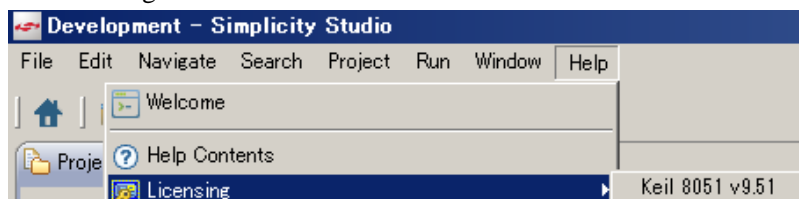
<http://www.silabs.com/products/mcu/Pages/simplicity-studio.aspx>

### 4-2 KEIL コンパイラのライセンス設定

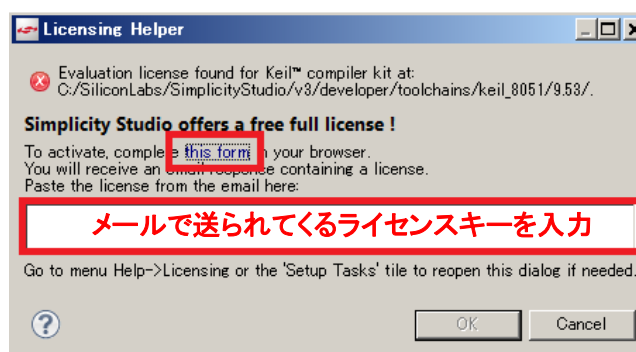
KEIL コンパイラを利用するために、ライセンス登録を行います。Simplicity Studio を起動し Simplicity IDE を選択します。



Help メニュー ⇒ Licensing ⇒ Keil を選択します。



Licensing Helper ウィンドウの中央にある「this form」をクリックし、必要事項を入力します。メールでライセンスキーが送られてきますので、それを Licensing Helper ウィンドウに入力してください。



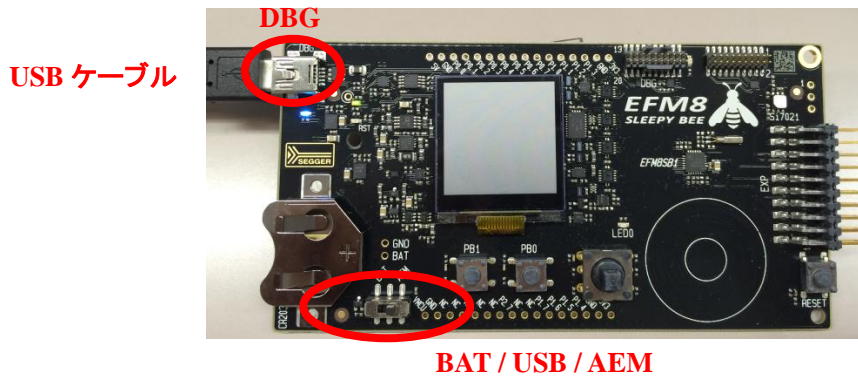
## 5 ハードウェア・セットアップ

EFM8/C8051 の評価に必要なハードウェアの設定を行います。

### 5-1 EFM8 Starter Kit のセットアップ

以下の手順で設定していきます。

1. BAT, USB, AEM の中から、基板に給電する方法を選びます。スイッチを AEM に切り替えます。
2. DBG と PC を USB ケーブルで接続します



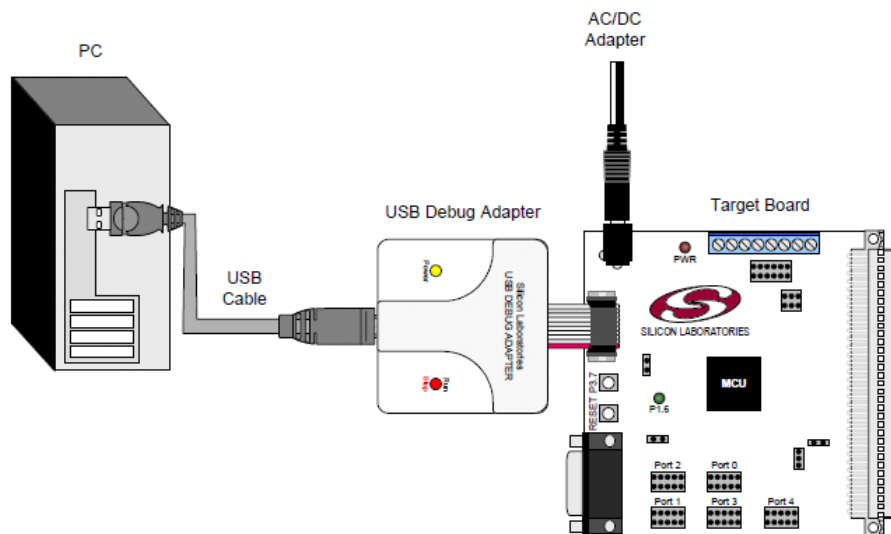
### 5-2 C8051 Development Kit のセットアップ

以下の手順で設定していきます。

1. USB デバッグアダプタを、ターゲットボードの DEBUG ポートに接続します。
2. USB ケーブルを、USB デバッグアダプタに接続します。
3. ターゲットボードにショートブロックがある場合には、ユーザガイドの指示に従って、正しく結線されているか確認してください。

例) F912DK, F930DK, F996DK の場合: J17 をショート、SW5 を ON に設定。

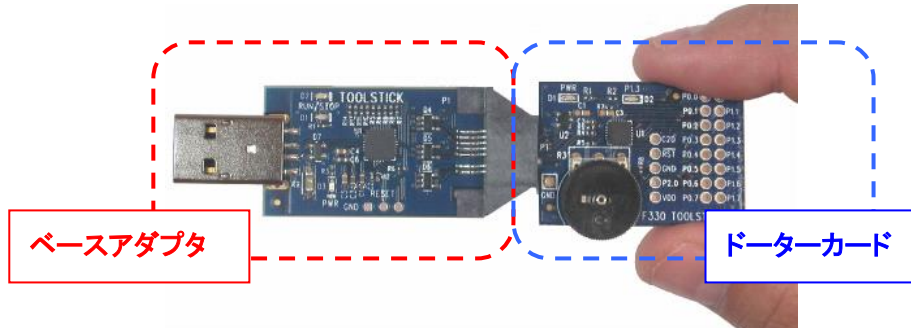
4. USB ケーブルのもう一方を PC に接続します。
5. AD/DC アダプタをターゲットボード、コンセントに接続します。



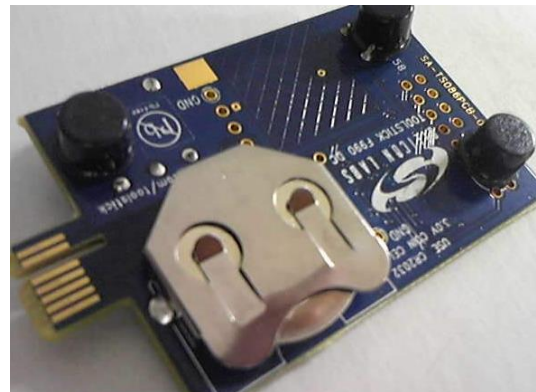
## 5-3 C8051 ToolStick のセットアップ

以下の手順で設定していきます。

1. ベースアダプタとドーターカードとを、下の写真のように接続します。



2. F912DC や F990DC など、いくつかのドーターカードでは電池でも動作するようになっています。USB 給電で動作するか、電池で動作するかを切り変えるスイッチがありますので、USB 給電 (TS PWR) を選択して下さい。



給電用スイッチ

3. USB コネクタを PC に接続します

## 6 使用方法

評価キットとSimplicity Studioを使用した評価手順をご紹介します。ここではSLSTK2010A (Sleepy Bee) を使用しておりますが、他の評価キットでも手順は同じです。

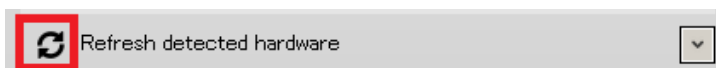
なお、各ツールから Simplicity Studio の Top 画面に戻るには、画面右上のアイコンを使用します。



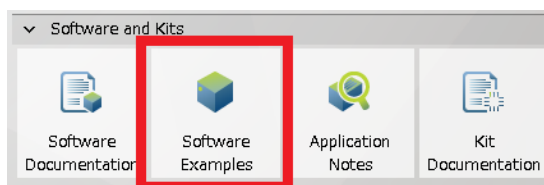
### 6-1 サンプルコードを動かしてみる

Starter Kit 上の LED を点滅させるサンプルコードを、ダウンロードして動作を見てみます。

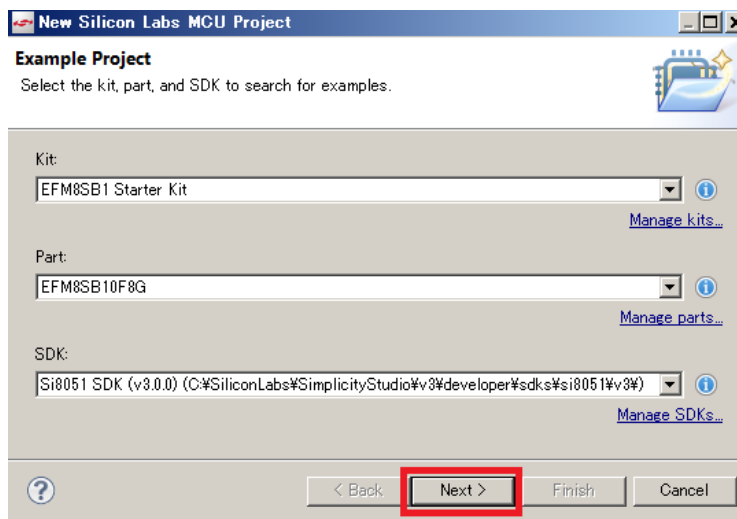
Starter Kit を PC に接続すると、Simplicity Studio が Starter Kit を自動認識します。うまく認識してくれない場合には、Refresh detected hardware アイコンを押してみてください。



Software Examples を選択します。

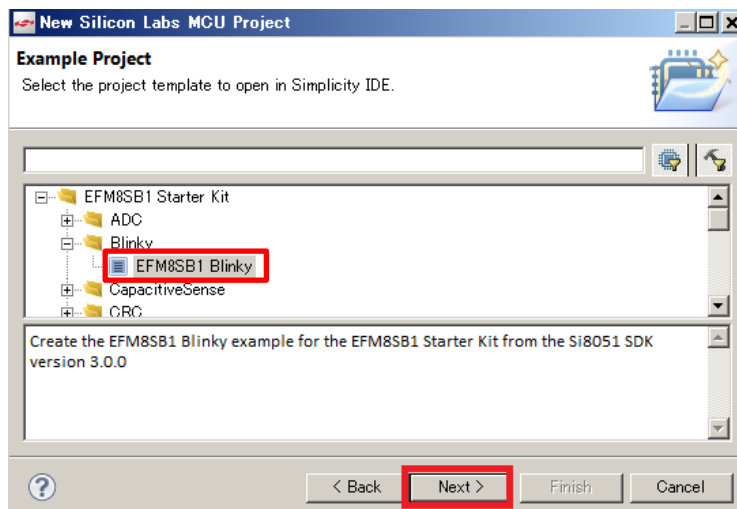


接続した Starter Kit に合わせて、Kit, Part, SDK が自動で選ばれますので、Next をクリックします。

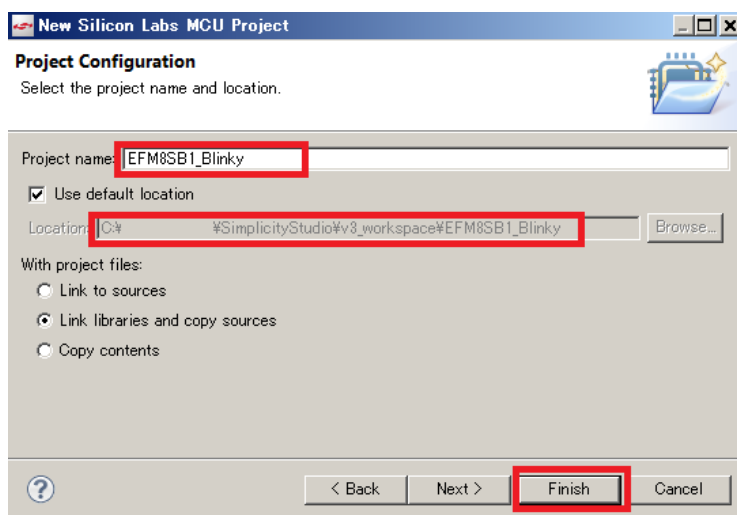




Example Project で EFM8SB1\_Blinky (EFM9xx\_blink) を選択し、Next をクリックします。

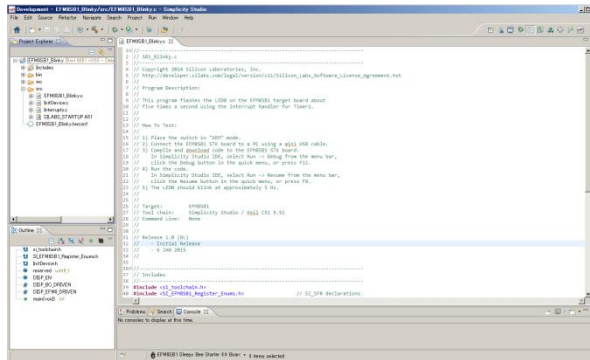


プロジェクト名を入力し、作業フォルダを指定します。With project files では、サンプルコードをローカルにコピーして使うかどうかを指定します。指定が終わったら、Finish をクリックします。

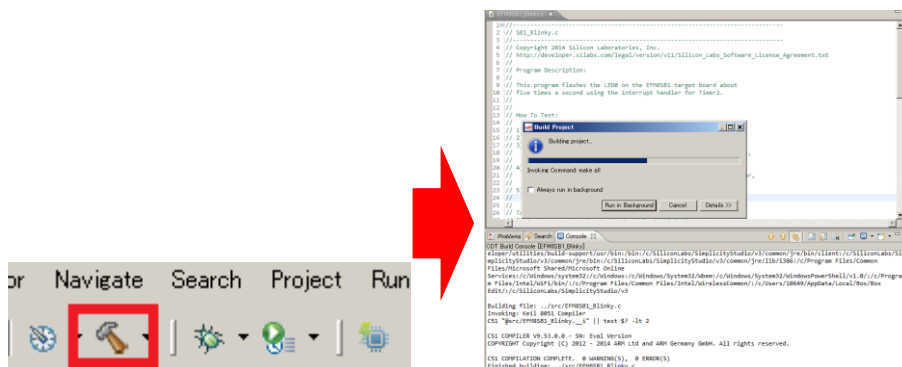


With project files	内容
Link to sources	ライブラリもソースも、オリジナルのものを使う。ライブラリもソースも修正しない人向け。
Link libraries and copy sources	ライブラリはオリジナルのものを参照し、ソースコードはローカルにコピーして使う。
Copy contents	ライブラリもソースも、ローカルにコピーして使う。ライブラリを修正する可能性がある人向け。

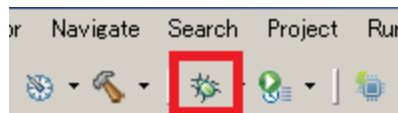
サンプルコードの準備が整うと、Simplicity IDE が起動します。Simplicity IDE の使い方については「6-2 デバッグ機能を使ってみる」で紹介します。



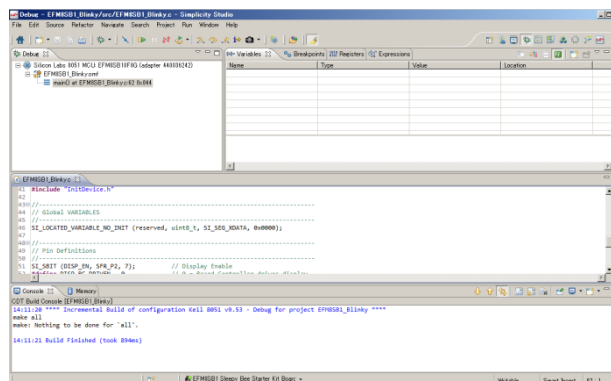
この Simplicity IDE を使用して、サンプルコードをビルドし、Starter Kit にダウンロードします。まずはトナカチのアイコン (Build) をクリックします。コンパイラが走り、サンプルコードがビルドされます。



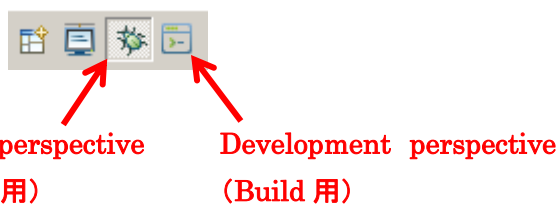
ビルドが完了したら、次に虫のアイコン (Debug) をクリックし、Starter Kit にダウンロードします。



ダウンロードが完了すると、デバッグ用の画面に切り替わります。



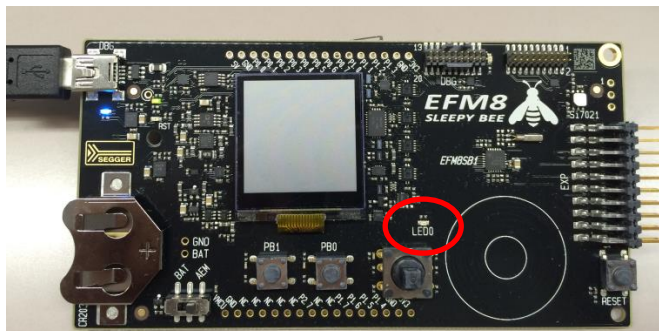
ビルド用の画面と、デバッグ用の画面の切り替えは、ウィンドウ右上のアイコンで行います。



サンプルコードを実行します。下図の実行のアイコン (Resume) をクリックしてください。



Starter Kit 上の LED が、ゆっくりと点滅しているのが確認できます。SLSTK2010A の場合には、LCD 下の LED0 が点滅します。

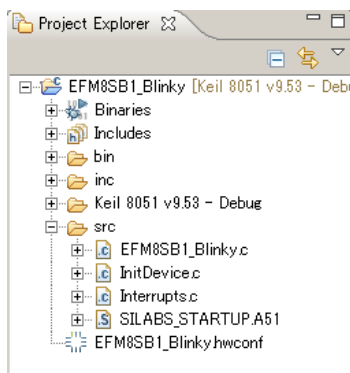


◆ LED の点滅スピードを変更してみましょう。

Development perspective アイコンをクリックして、ビルド用の画面に切り替えます。



画面左に Project Explorer があり、ソースコードの階層が表示されています。



- ◆ EFM8SB1\_Blinky.c: メインルーチン
- ◆ InitDevice.c: 主にペリフェラルの初期化
- ◆ Interrupts.c: 割り込み処理
- ◆ SILABS\_STARTUP.A51: スタートアップルーチン
- ◆ EFM8SB1\_Blinky.hwconf: Configurator のプロジェクトファイル

main()を見ると LED 点滅に関する処理は行われておらず、Interrupts.c を見ると Timer2 割り込みで LED への制御ピンを反転させています。Timer2 のオーバフロー周期を変更すれば点滅スピードを変えることができそうです。

```

//-----
// main() Routine
//-----
int main (void)
{
    //Enter default mode
    enter_DefaultMode_from_RESET();

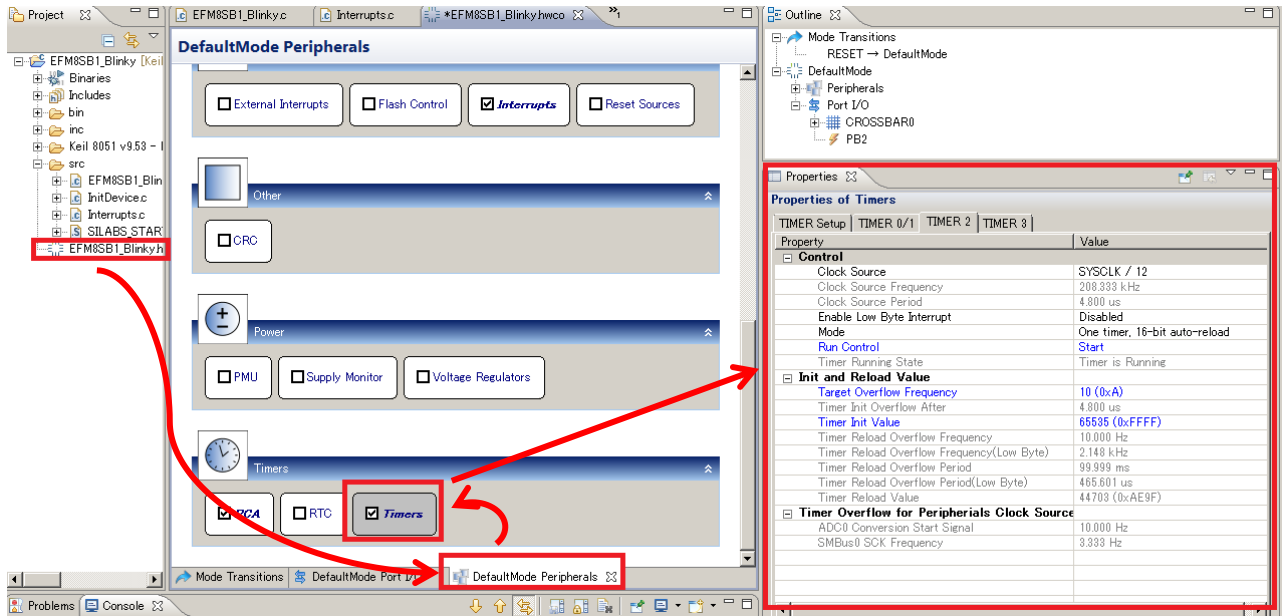
    DISP_EN = DISP_BC_DRIVEN;

    while (1) {} // Spin forever
}

//-----
// TIMER2_ISR
//-----
//
// TIMER2 ISR Content goes here. Remember to clear flag bits:
// TMR2CN0::TF2H (Timer # High Byte Overflow Flag)
// TMR2CN0::TF2L (Timer # Low Byte Overflow Flag)
//-----
SI_INTERRUPT (TIMER2_ISR, TIMER2_IRQn)
{
    TMR2CN0_TF2H = 0; // clear Timer2 interrupt flag
    LED0 = !LED0; // change state of LEDs
}
    
```

もちろんソースコードを追って、設定変更する方法もありますが、ここでは Configurator を使ってみます。FM8SB1\_Blinky.hwconf をダブルクリックすると Configurator が起動します。

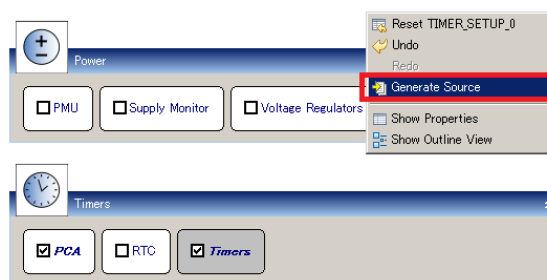
DefaultMode Peripherals タブを選択し、Timers を選択、画面右に Timer 設定が表示されます。



Timer2 の初期値設定を見てみると、Timer2 のオーバフロー周期が 10Hz に設定されていました。これを 50Hz に変更してみます。

Init and Reload Value		Init and Reload Value	
Target Overflow Frequency	10 (0xA)	Target Overflow Frequency	50 (0x32)
Timer Init Overflow After	4.800 us	Timer Init Overflow After	4.800 us
Timer Init Value	65535 (0xFFFF)	Timer Init Value	65535 (0xFFFF)
Timer Reload Overflow Frequency	10.000 Hz	Timer Reload Overflow Frequency	49.996 Hz
Timer Reload Overflow Frequency(Low Byte)	2.148 kHz	Timer Reload Overflow Frequency(Low Byte)	2.934 kHz
Timer Reload Overflow Period	99.999 ms	Timer Reload Overflow Period	20.002 ms
Timer Reload Overflow Period(Low Byte)	465.601 us	Timer Reload Overflow Period(Low Byte)	340.801 us
Timer Reload Value	44703 (0xAE9F)	Timer Reload Value	61369 (0xEF89)

設定変更をソースコードに反映させます。DefaultMode Peripherals タブに戻り、右クリックし、Generate Source を選択します。

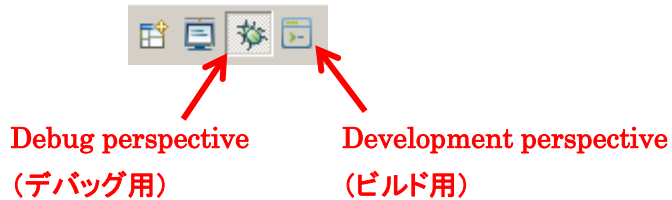


あとは、ビルドして、ダウンロードして、実行します。先ほどと同じ手順で、トンカチのアイコン (Build) ⇒ 虫のアイコン (Debug) ⇒ 実行のアイコン (Resume) の順にクリックします。Starter Kit の LED の点滅が、先ほどよりも早くなったことを確認できるかと思います。

使用する MCU ファミリによってサンプルコードの内容は異なりますが、Blinky.c のようにシンプルなサンプルコードは、制御方法を理解するのに最適です。

## 6-2 デバッグ機能を使ってみる (Simplicity IDE)

ソフトウェア・デバッグの際に使用する、ブレークポイント、ステップ実行などの機能は、Simplicity IDE が提供します。「6-1 サンプルコードを動かしてみる」でも紹介しましたが、Simplicity IDE には、ビルド用の画面と、デバッグ用の画面が用意されています。ソフトウェア・デバッグは、デバッグ用の画面で行います。



### ◆ コードの実行・停止

コードの実行には Resume ボタン、停止には Suspend ボタンを使用します。



### ◆ ハードウェア・リセット

MCU にハードウェア・リセットをかけます。



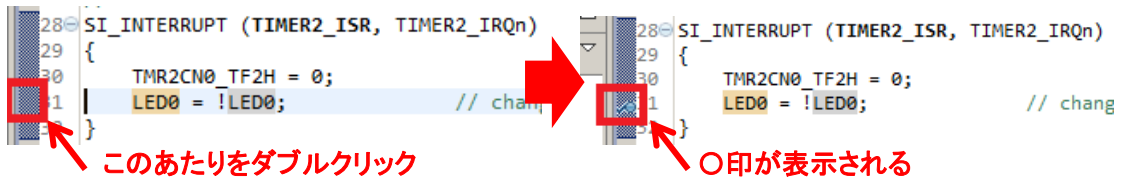
### ◆ デバッグ経路の切断

デバッグモード経路を切断して、デバッグ用の画面を終了します。ビルド用の画面に切り替わります。



### ◆ ブレークポイント

ブレークポイントを設定するには、停止させたい行の左横をダブルクリックします。設定されると、水色の小さな○印が表示されます。再度ダブルクリックすれば解除されます。

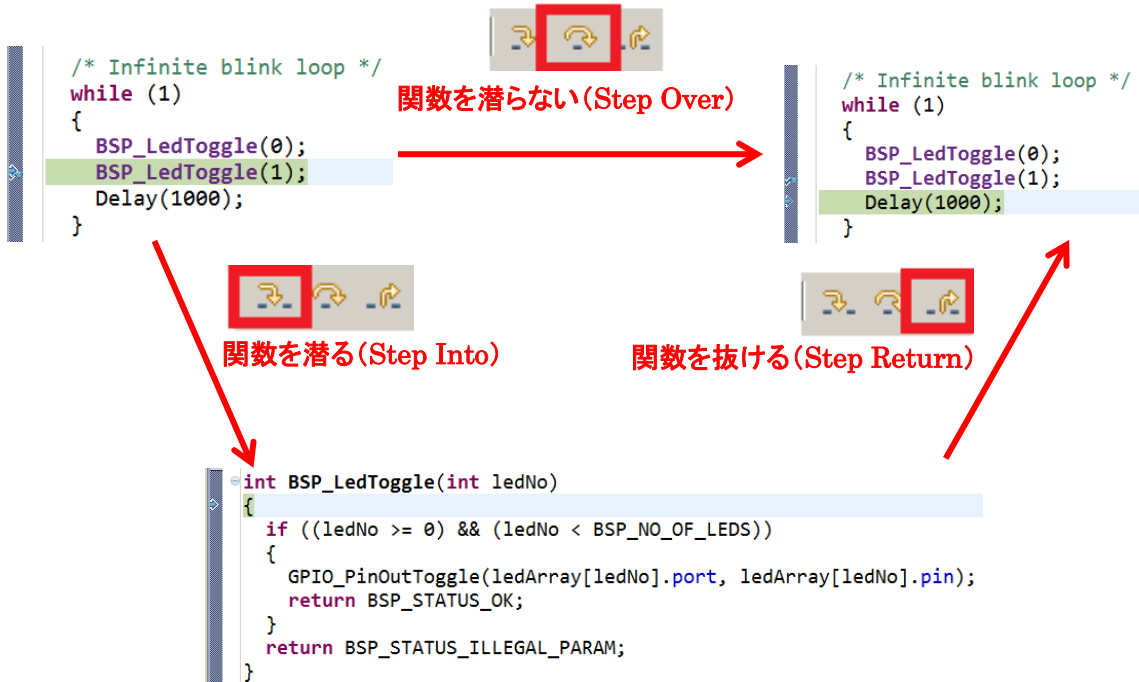


### ◆ ステップ実行

各種ステップ実行に対応しています。



実機で実際に動作を見て頂くのが、判りやすいです。



### ◆ レジスタ値の閲覧・変更

レジスタ・変数の閲覧や変更は、下のウィンドウ (Register ウィンドウなど) で行うことができます。前回の停止から、値が変化した場合には黄色で表示されます。

Name	Value	Description
SMBus 0		SMBus 0 Registers
SPI 0		SPI 0 Registers
Temperature Sensor		Temperature Sensor Registers
TIMER 0/1		TIMER 0/1 Registers
TIMER 2		TIMER 2 Registers
TMR2	0xEFBA	Timer 2 Word
TMR2CN0	0xC4	Timer 2 Control 0
TMR2RL	0xEFB9	Timer 2 Reload Word
TIMER 3		TIMER 3 Registers
TIMER Setup		TIMER Setup Registers
UART 0		UART 0 Registers
Supply Monitor		Supply Monitor Registers
Voltage Reference		Voltage Reference Registers

TIMER 0/1	
TIMER 2	
TMR2	0xEFBA
TMR2CN0	0xC4
TMR2RL	0xEFB9
TIMER 3	

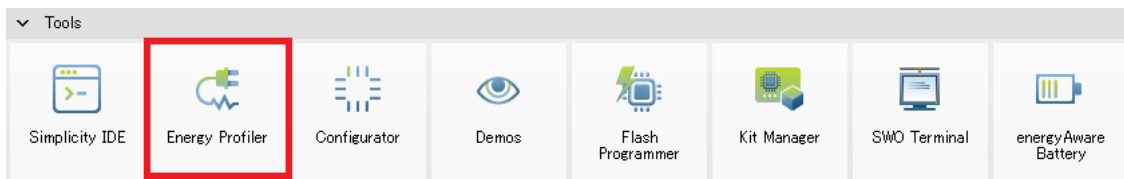


### 6-3 消費電流を測定してみる (Energy Profiler)

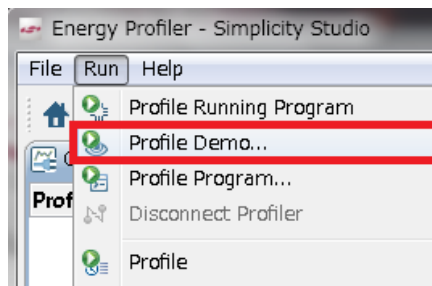
EFM8 Starter Kit には電流センサが搭載されており、消費電流測定ツール (Energy Profiler) と組み合わせることで nA レベルでの電流測定が可能です。Starter Kit には LCD など外部部品も実装されていますが、MCU 単体の消費電流が測定できるように配慮されています。ただし C8051 の評価キットは対応していません。

ここではサンプルコードを使用して、消費電流測定ツール (Energy Profiler) の使用方法をご紹介します。

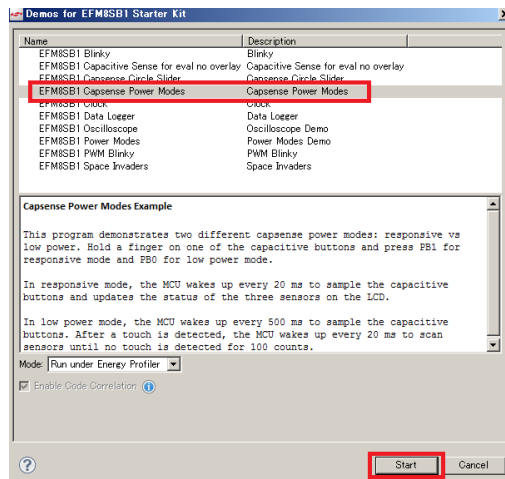
Energy Profiler を起動します。



Run メニューから、Profile Demo を選択します。



EFM8SB1 Capsense Power Modes を選択します。そして Start をクリックします。

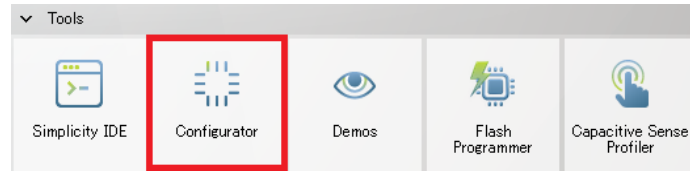


このサンプルコードは、ボタンで EFM8 のパワーモードを切り替えて、その際の消費電流をモニタできる機能です。EFM8 の Starter Kit には、消費電力のモニタ機能がついています。

### 6-4 ピン設定やペリフェラル設定を試みる (Configurator)

レジスタ設定を補助するツールとして Configurator (新しい製品向け) および Configuration Wizard2 (レガシー製品向け) が用意されています。ここでは、Configurator の使用方法を簡単にご紹介します。

Configurator を選択します。

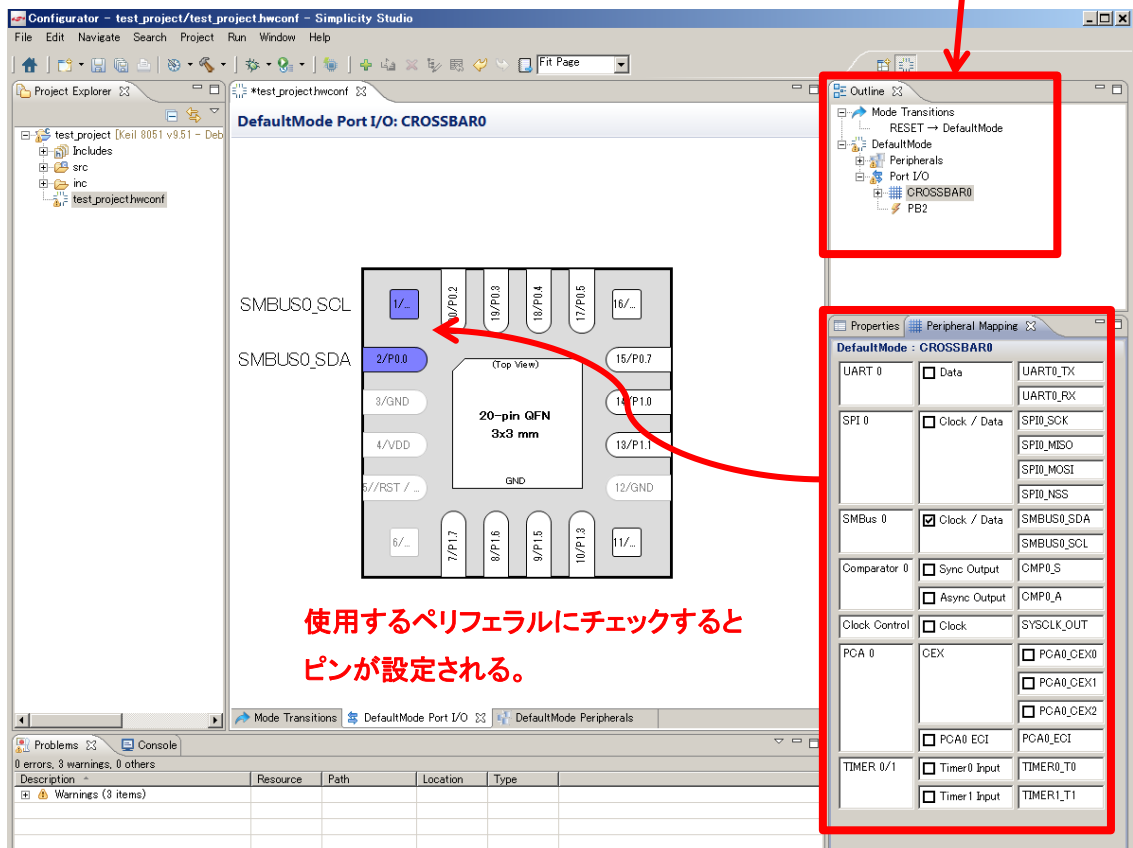


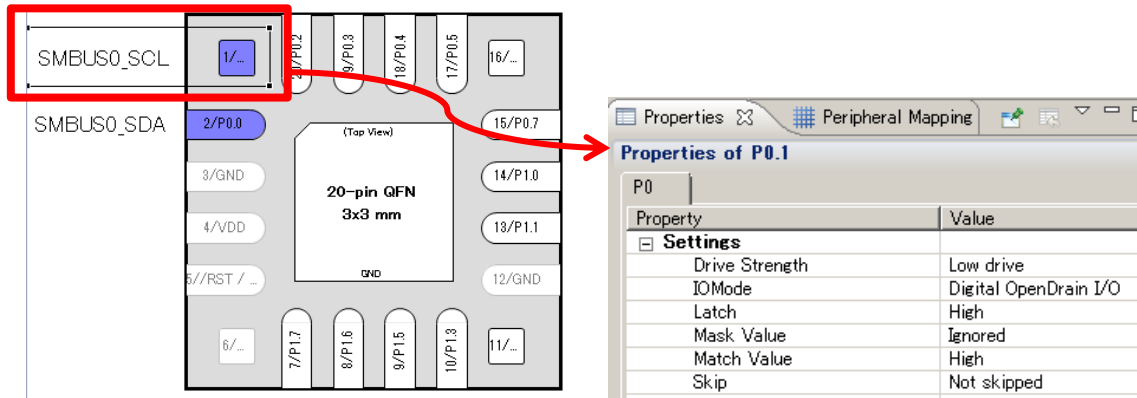
使用する Kit、型番などを指定してプロジェクトを作成します。プロジェクトを作成済みであれば、使用するプロジェクトを選択します。完了すると Configurator が起動します。

#### ◆ ピン設定

使用するペリフェラルにチェックを入れると、使用するピンが紫色に変わります。ピンの上でクリックすると、IO モード (Open Drain、Push-pull)、ドライブ・ストレンクス、Pin Skip 設定などを変更できます。

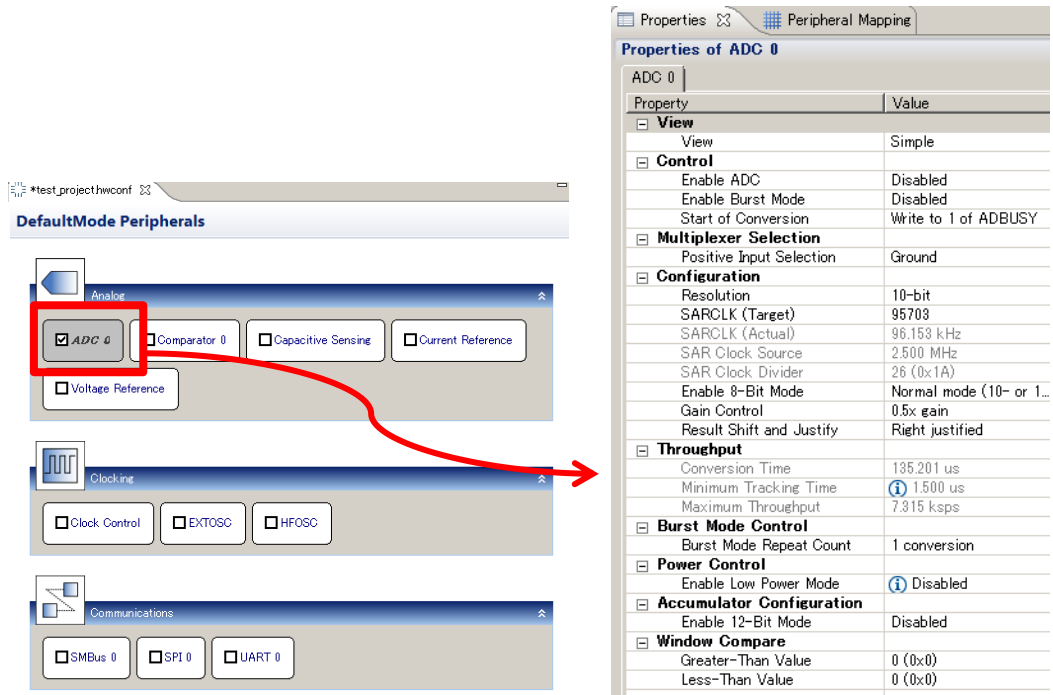
ピン設定とペリフェラル設定の切り替え





◆ ペリフェラル設定

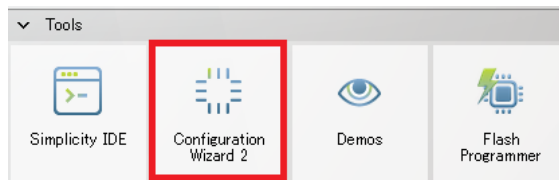
使用するペリフェラルにチェックを入れると、細かな設定が行えます。



### 6-5 ピン設定やペリフェラル設定を試みる (Configuration Wizard 2)

Configuration Wizard 2 の使用方法を簡単にご紹介します。

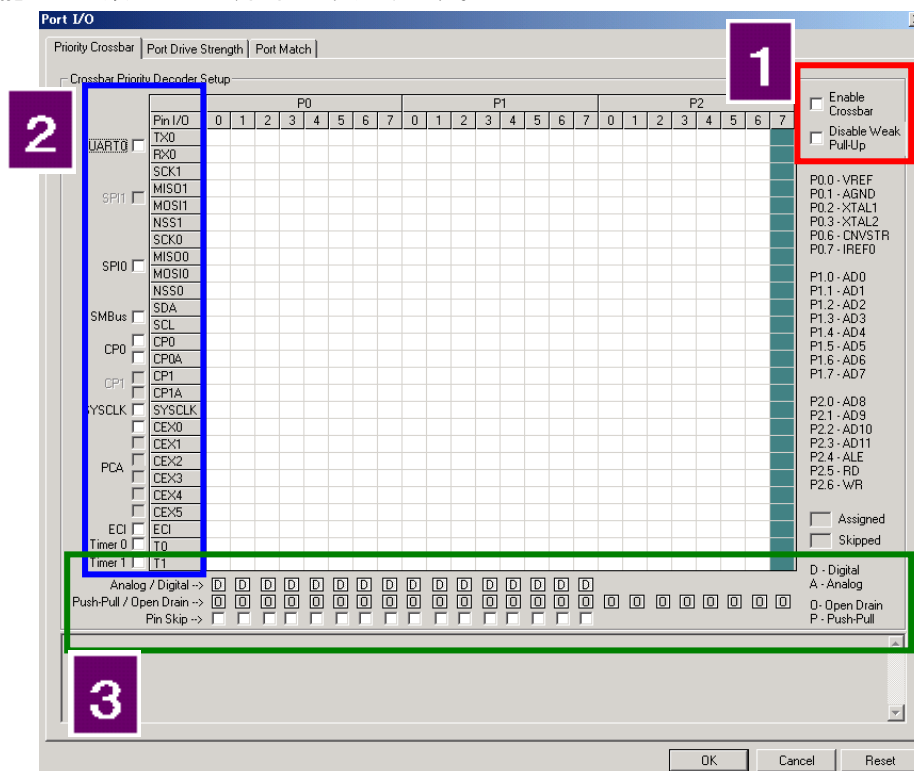
Configuration Wizard 2 を選択します。



New Project ウィンドウで MCU の型番を選択すると、Configuration Wizard 2 が起動します。

#### ◆ ピン設定

Peripherals メニューから Port I/O を選択すると、下図のようなウィンドウが表示されます。使用する MCU の機能やピン数によって、画面が異なります。



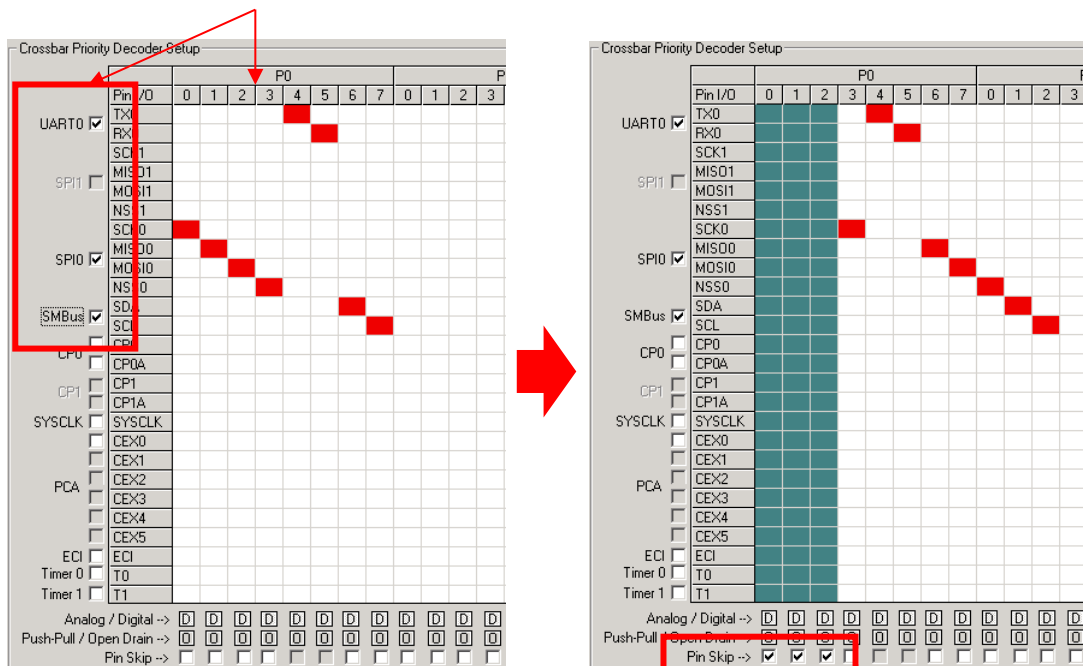
番号	概要
1	Crossbar の有効・無効 (必ず有効にして下さい) 内蔵 pull-up の有効・無効
2	使用するペリフェラルの選択
3	アナログピン、デジタルピンの切り替え 出力モード(push-pull, open-drain)の切り替え ピンスキップの設定

I/O を使用するペリフェラル(UART, SPI など)を使用する場合、ポートはそれらペリフェラルに優先的に割り振られます。特定のポートを GPIO として使用したい場合には、ピンスキップの設定を行うことで優先的に確保できます。

下図(左)は、UART, SPI0, SMBus を使用する場合のポート設定です。P0.0~P0.7 が、これらペリフェラルに割り当てられています。

それに対し、下図(右)は、P0.0~P0.2 をピンスキップ設定した場合です。ペリフェラルのポート割り当てが P0.3~P1.2 へ変更されたことが判ります。

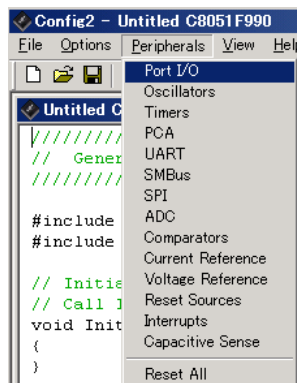
UART, SPI0, SMBus を選択。 P0.0~P0.7 が優先的に占有された。



ピンスキップ設定

◆ ペリフェラル設定

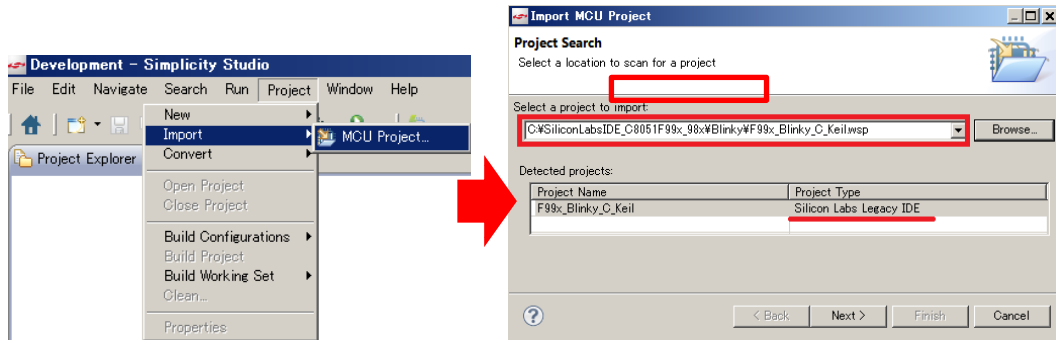
各ペリフェラルの設定は、Peripherals メニューを使用して行っていきます。一つ設定が終るたびに、Cコードのヘッダーファイルが更新されます。このヘッダーファイルには、設定に準じたペリフェラル初期化ルーチンが記述されていますので、ソフトウェア設計の際にご使用頂けます。



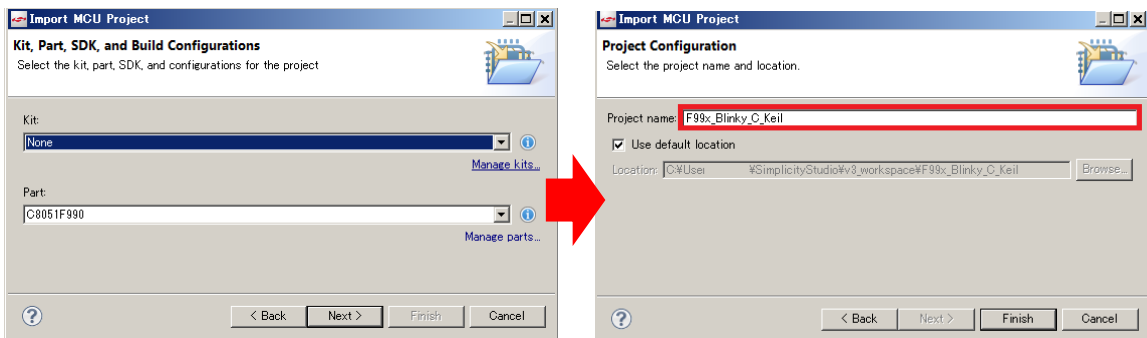
### 6-6 Silicon Labs IDE(レガシーな IDE)から Simplicity IDE(新しい IDE)へ移行する方法

C8051 の開発環境であった Silicon Labs IDE から、新しい開発環境である Simplicity IDE へ移行する方法をご紹介します。

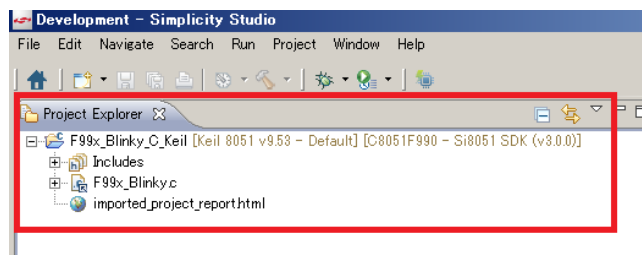
Simplicity IDE を起動し、Project ⇒ Import ⇒ MCU Project を選択します。Silicon Labs IDE のプロジェクトである.wsp を選択し、Next をクリックします。



評価キット名(必要であれば)、および使用する MCU 型番を選択し、Next をクリックします。最後にプロジェクト名を入力して Finish をクリックします。



これでプロジェクトの移行は完了です。Project Explorer にプロジェクトが移行され、コンパイルする準備が整いました。





## 改版履歴

Version	改定日	改定内容
1.0	2011年07月	・新規作成
1.3	2015年06月	・EFM8に対応。マクニカオンラインで公開

## 参考文献

- Silicon Labs 社 各種ドキュメント
- Silicon Labs 社 ナレッジベース、コミュニティフォーラム

### 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. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不審な点や誤り、記載漏れなどお気づきの点がありましたら、弊社までご一報いただければ幸いです。
4. 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。
5. 本資料は製品を利用する際の補助的なものとしてかかれたものです。製品をご使用になる場合は、メーカーリリースの資料もあわせてご利用ください。

本社

〒222-8561 横浜市港北区新横浜 1-6-3 TEL 045-470-9841 FAX 045-470-9844