
KEIm-08SoM ヘッダーボード スタートアップガイド

Ver.1.0



株式会社近藤電子工業

はじめに

この度は、KEIm 製品をお買い上げいただき誠にありがとうございます。

本製品をご使用になる前に、本マニュアル及び関連資料を十分ご確認ください、使用上の注意を守って正しくご使用ください。



取扱い上の注意

- 本書に記載されている内容は、将来予告なく変更されることがあります。本製品のご使用にあたっては、弊社窓口又は弊社ホームページなどで最新の情報をご確認ください。
- 本製品には一般電子機器用部品が使用されています。極めて高い信頼性を要求する装置(航空、宇宙機器、原子力制御機器、生命維持のための医療機器等)には使用しないでください。
- 本製品は国内使用を前提として開発及び製造を行っています。本製品又は本製品を組み込んだ製品を輸出される場合は、お客様の責任において「外国為替及び外国貿易法」及びその他輸出関連法令等を順守し、必要な手続きを行ってください。
- LAN、USB 以外のコネクタへのケーブルの抜き差しは、必ず電源を OFF にした状態で行ってください。
- 水、湿気、ほこり、油煙等の多い場所では使用しないでください。
- 本製品の関連資料の全部又は一部を弊社に無断で使用または複製することを禁止します。
- 本書及び関連資料で取り上げる会社名及び製品名等は、各メーカーの商標または登録商標です。

お問い合わせ先

- 製品に関するお問い合わせは、下記のメールアドレスよりお願いいたします。

keim-support@kd-group.co.jp

目次

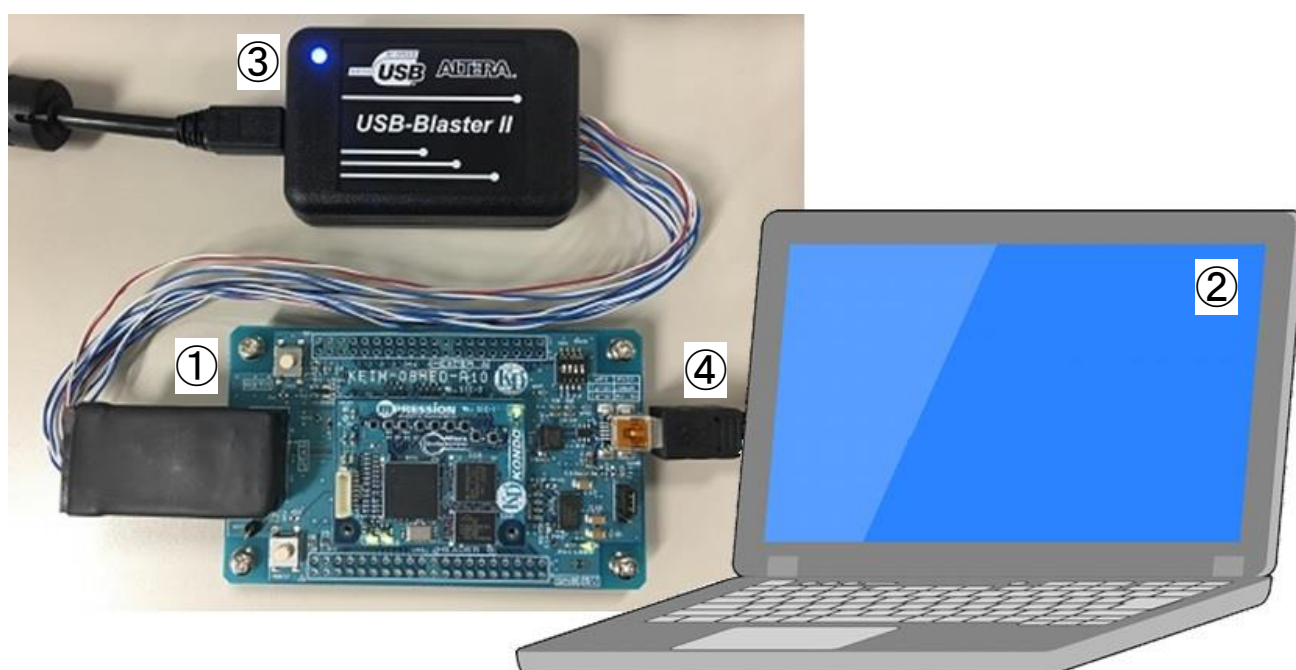
1. 概要.....	4
1.1. 開発環境.....	4
2. 開発準備	5
2.1. Quartus Prime のインストール	5
2.2. KEIm08 用 FPGA サンプルデザインを開発用 PC にコピー	5
3. プログラム作成	6
3.1. Hello World 表示プログラム.....	6
3.1.1. Nios II SBT の起動	6
3.1.2. Nios II Application と BSP を Template より生成	8
3.1.3. sopcinfo ファイルのコピー	8
3.1.4. BSP の設定	9
3.1.5. ソフトウェアプロジェクトのビルド	13
3.1.6. FPGA 内部回路イメージの download	14
3.1.7. ソフトウェアプロジェクトの実行	17
3.2. LED 点滅プログラム	24
3.2.1. 新規 Application プロジェクト作成と BSP の設定	24
3.2.2. hello_led.c をコピー	26
3.2.3. プロジェクトのビルド	27
3.2.4. プログラムの実行	28
4. FPGA デザインの書き込み	33
4.1. Quartus Prime Programmer での書き込み	33
5. プログラムの ROM 化及び Flash ROM 書き込み	34
6. 補足資料	39
6.1. USB シリアルドライバのインストール	39
6.1.1. VCP ドライバのインストール	39
6.1.2. VCP ドライバの設定	40
7. 更新履歴	42

1. 概要

本書は KEIm-08SoM ヘッダーボードを使用してソフトウェアを開発する際のスタートアップガイドです。開発はインテル® FPGA 開発ツール Quartus® Prime® 開発ソフトウェア(以降 Quartus Prime) 及びそれに含まれるツール Nios® II Embedded Design Suite (以降 Nios II EDS) を使用して行います。本書ではこれらのツールを使用して次の内容を実施します。

- ① Hello World 表示プログラム生成及び動作確認
- ② LED 点滅プログラム及び動作確認
- ③ プログラムの ROM 化及び Flash ROM 書き込み

1.1. 開発環境



	名称	補足
①	ヘッダーボード	KEIm-08SoM 搭載済み
②	開発用 PC	OS: Windows7 64bit 開発ツール *2 : Quartus Prime Design Software v18.1.0 以降及び Nios II Embedded Design Suite v18.1.0 以降
③	Intel FPGA Download Cable II™	USB-Blaster II™ 別売り *1
④	USB mini B ケーブル	KEIm SoM 開発キットに同梱

*1 本製品には USB-Blaster II は付属しません。ご利用の際は、取扱販売店又は代理店などを通じ別途ご購入ください。

*2 本書では Quartus Prime Design Software v18.1.0.625 を例として示します。それ以降のバージョンはバージョンを読み替えて参照してください。

2. 開発準備

2.1. Quartus Prime のインストール

KEIm SoM 開発キットを使用してソフトウェア開発を行うためには、インテル社の FPGA 開発ツール Quartus Prime 及び Nios II EDS が必要です。開発用 PC にはあらかじめこれらのツールをインストールする必要があります。

ツールのダウンロード及びインストール方法はインテル社のホームページをご参照ください。

<https://www.intel.co.jp/content/www/jp/ja/software/programmable/quartus-prime/overview.html>

Nios II EDS は Quartus Prime をインストールする際に合わせてインストールされます。

2.2. KEIm08 用 FPGA サンプルデザインを開発用 PC にコピー

弊社 WEB サイトより “keim08_tiny_fast_q181.zip” をダウンロードして開発用 PC にコピーしてください。このデザインは、Nios[®] II/f “Fast” core を使用しています。Nios[®] II/f を含む FPGA デザインをコンパイルする場合は、IPS-EMBEDDED または IP-NIOS のライセンスが必要です。

これらのライセンスがない場合は、Nios[®] II/e “Economy” core を使用した “keim08_tiny_economy_q181.zip” をダウンロードしてください。

Quartus Prime のライセンスに Nios[®] II “Fast” core を例として下記フォルダに格納して解凍した場合にて、以降の説明を進めます。Nios[®] II/e “Economy” core 版をご使用の場合は、フォルダ名を読み替えてください。

Nios[®] II/f “Fast” core の場合

D:\intel\FPGA\keim08\keim08_tiny_fast_q181.zip

解凍後のフォルダ

D:\intel\FPGA\keim08\keim08_tiny_fast_q181\

Nios[®] II/e “Economy” core の場合

D:\intel\FPGA\keim08\keim08_tiny_economy_q181.zip

解凍後のフォルダ

D:\intel\FPGA\keim08\keim08_tiny_economy_q181\

Nios[®] II のコアの違いは下記サイトを参照してください。

https://www.intel.co.jp/content/www/jp/ja/products/programmable/processor/nios-ii.html? ga=2.61131300.928125466.1538442790-1762824604.1533265000&elq_cid=3705377&erpm_id=6734302

ライセンスに付きましては、下記サイトを参照してください。

https://www.intel.co.jp/content/www/jp/ja/products/programmable/processor/nios-ii/embedded-ip-suite.html? ga=2.176649189.1554389885.1538442905-1762824604.1533265000&elq_cid=3705377&erpm_id=6734302



3. プログラム作成

ここでは、Nios II EDS を使用して下記 2 種類のプログラムを作成、その後 KEIm SoM 開発キットを使用して動作確認を行います。作成するプログラムは下記 2 種類とします。

- ①Hello World 表示プログラム
- ②LED 点滅プログラム

3.1. Hello World 表示プログラム

Nios II EDS のコンソールウィンドウ、または開発環境のセットアップで行った UART シリアルコンソール出力に”Hello World”を出力するプログラムを作成します。例としての記述は Nios II EDS のコンソールウィンドウを使用します。

主な作業手順としては下記の通りとなります。

- ① Template から BSP とソースファイルの生成
- ② プロジェクトのビルド
- ③ プログラムの実行

以降の説明に従ってプログラムの作成及び実行を行ってください。

3.1.1. Nios II SBT の起動

プログラムのビルドやデバッグには Nios II Software Build Tools for Eclipse (以降 Nios II SBT) を使用します。スタートメニューより Nios II SBT を立ち上げます。スタートメニューより順に

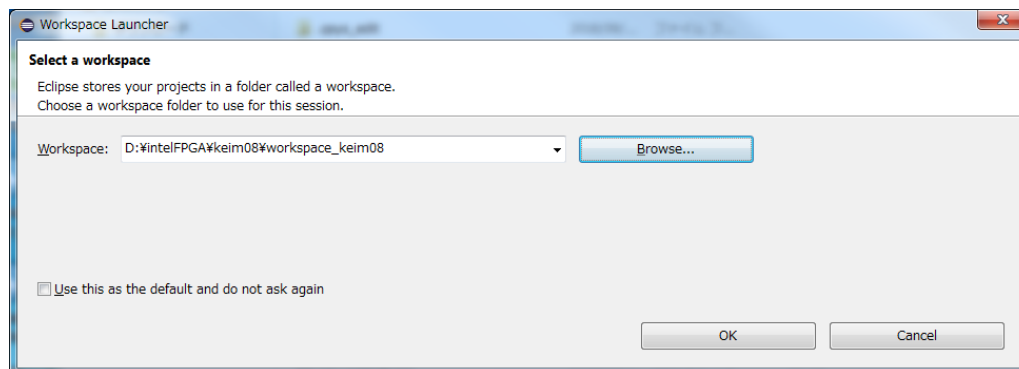
Intel FPGA 18.1.0.625 Lite Edition >

または

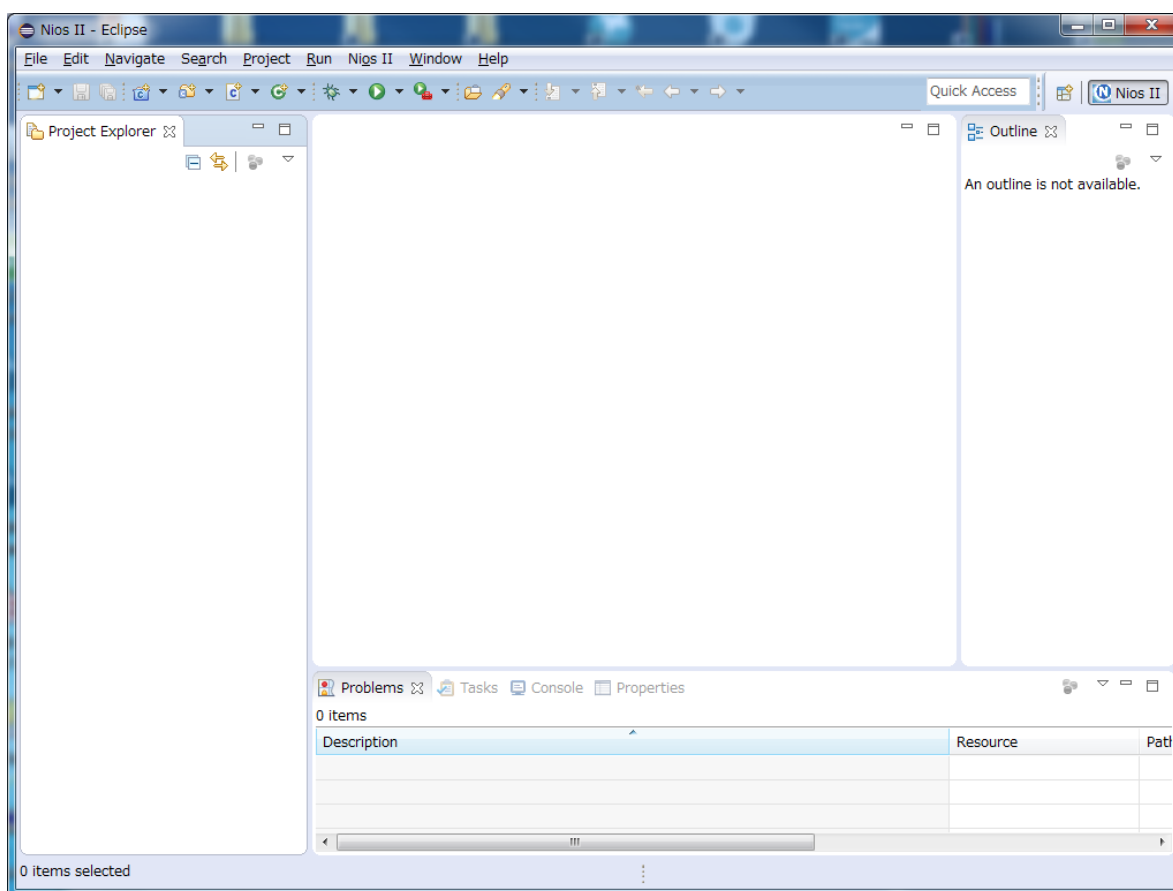
Intel FPGA 18.1.0.625 Standard Edition >

Nios II EDS 18.1.0.625 > Nios II Software Build Tools for Eclipse (Quartus Prime 18.1)
を選択し Nios II SBT を起動します。

Nios II SBT 立ち上げ時に Workspace の設定のダイアログが表示されますので、適切な場所に Workspace のパスを設定してください。本書では例として D:\intelFPGA\keim08\workspace_keim08 をワークスペースフォルダとしています。ワークスペースフォルダの選択をし OK をクリックすると Nios II SBT メイン画面が立ち上がります。



Workspace の指定



Nios II SBT メイン画面

3.1.2. Nios II Application と BSP を Template より生成

Nios II SBT でソフトウェアを開発する場合、BSP プロジェクトとアプリケーション側のプロジェクトの 2 つが必要です。SBT の Template を使って両方を生成します。

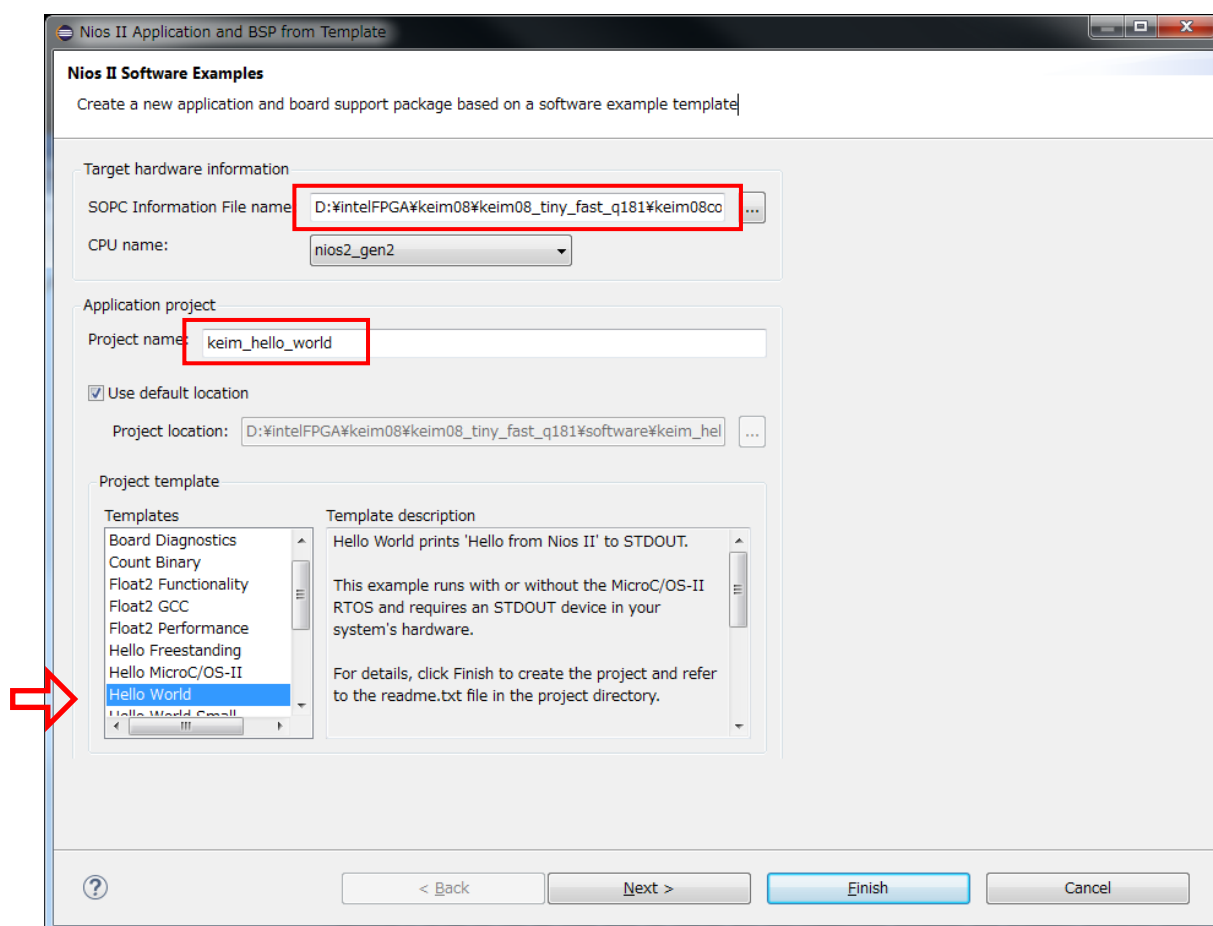
3.1.3. sopcinfo ファイルのコピー

BSP の作成には KEIm の FPGA 構成定義ファイルである”keim08core.sopcinfo”を使用します。

“2.3. KEIm08 用 FPGA サンプルデザインを開発用 PC にコピー” の項目で弊社 Web サイトよりダウンロードしたファイルに含まれています。例として下記に格納されています。

D:\intelFPGA\keim08\keim08_tiny_fast_q181\keim08core.sopcinfo

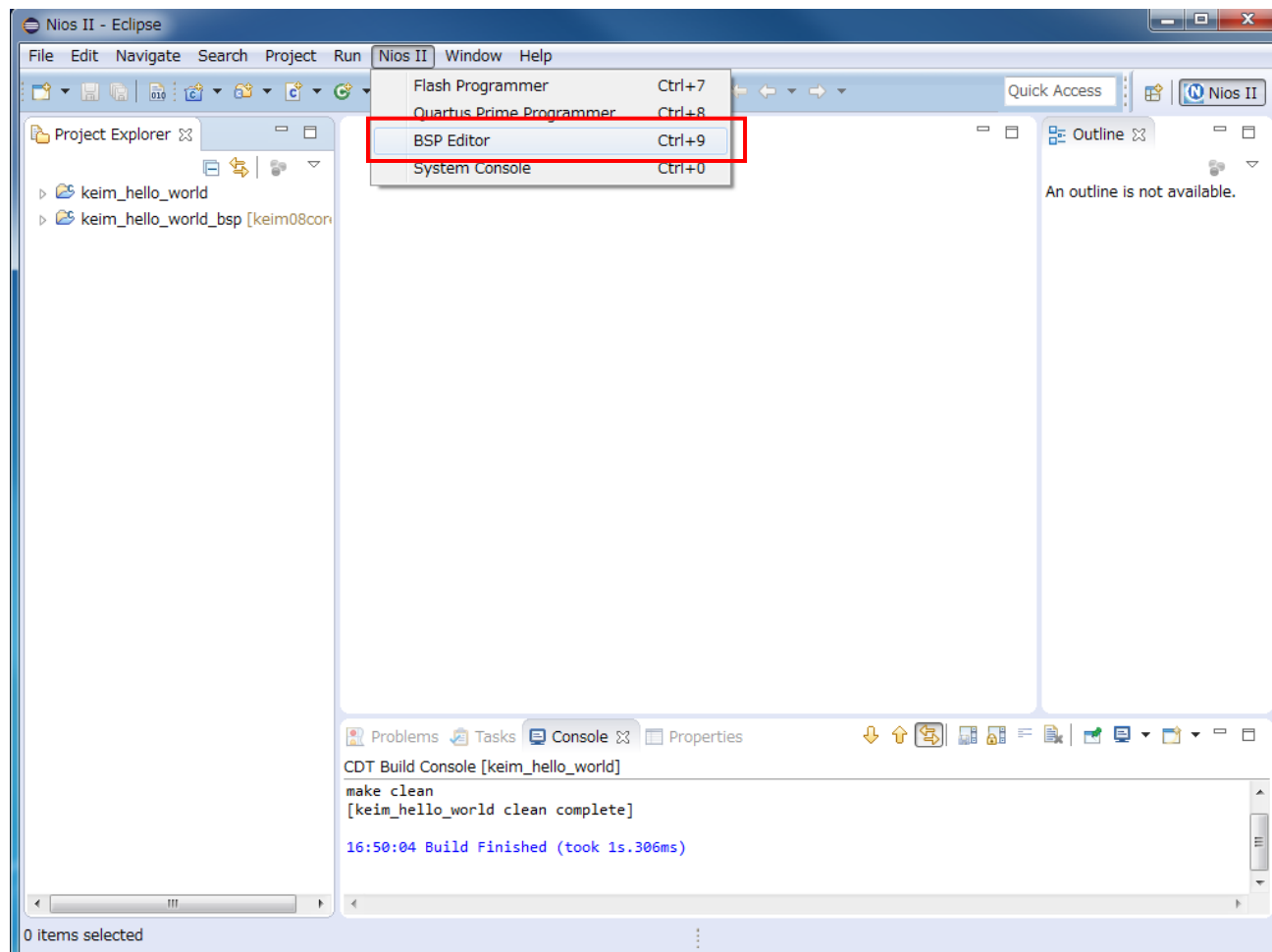
Nios II SBT のメニューから File > New > Nios II Application and BSP from Template を選択し、”keim08core.sopcinfo” ファイルをしていします。また、Templates の中から Hello World を選択します。Project name は任意ですがここでは keim_hello_world とします。下記赤枠部分を設定後、[Finish] をクリックします。

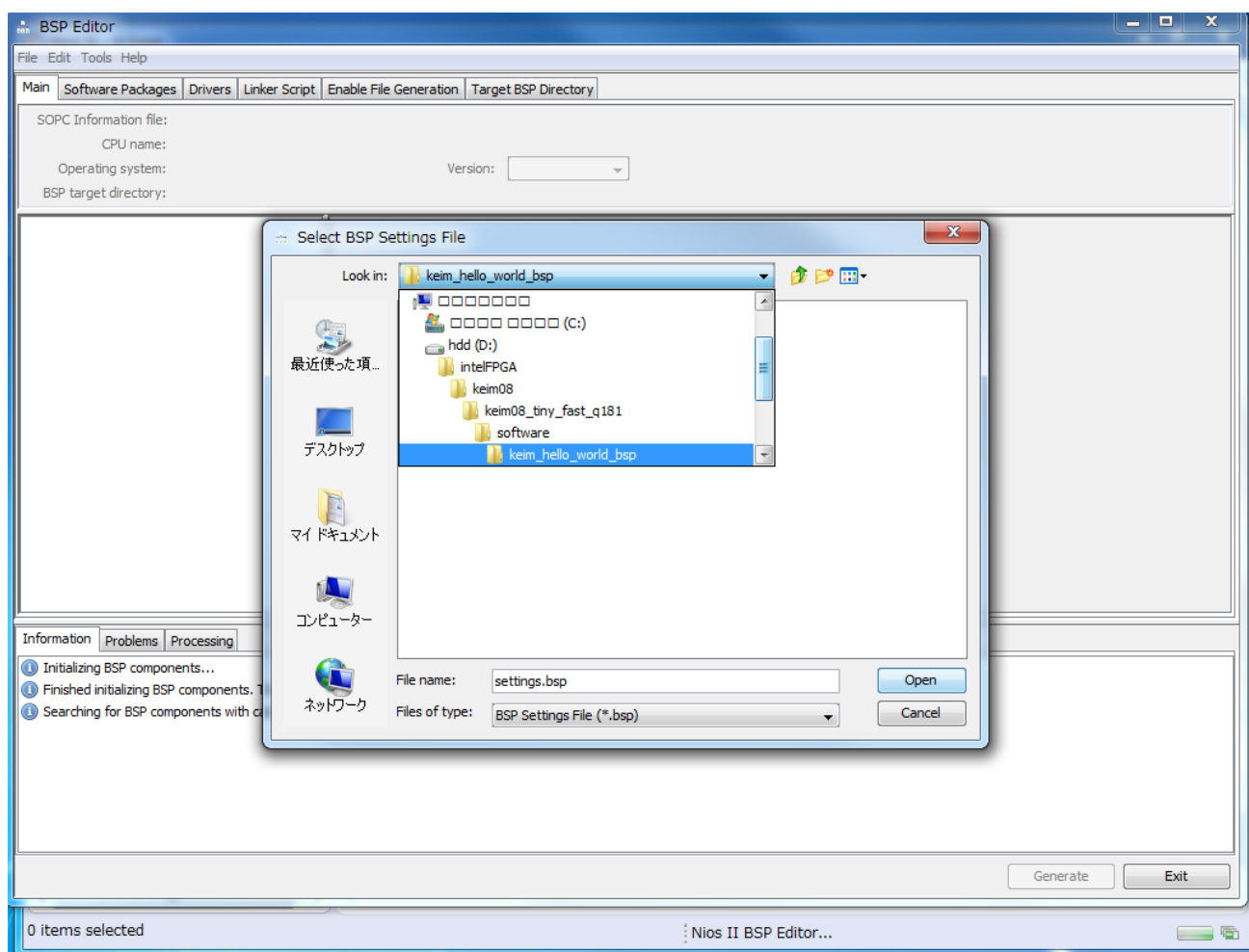


3.1.4. BSP の設定

先程生成された BSP の設定を確認（変更）します。

Nios II > BSP Editor で BSP editor を起動します。





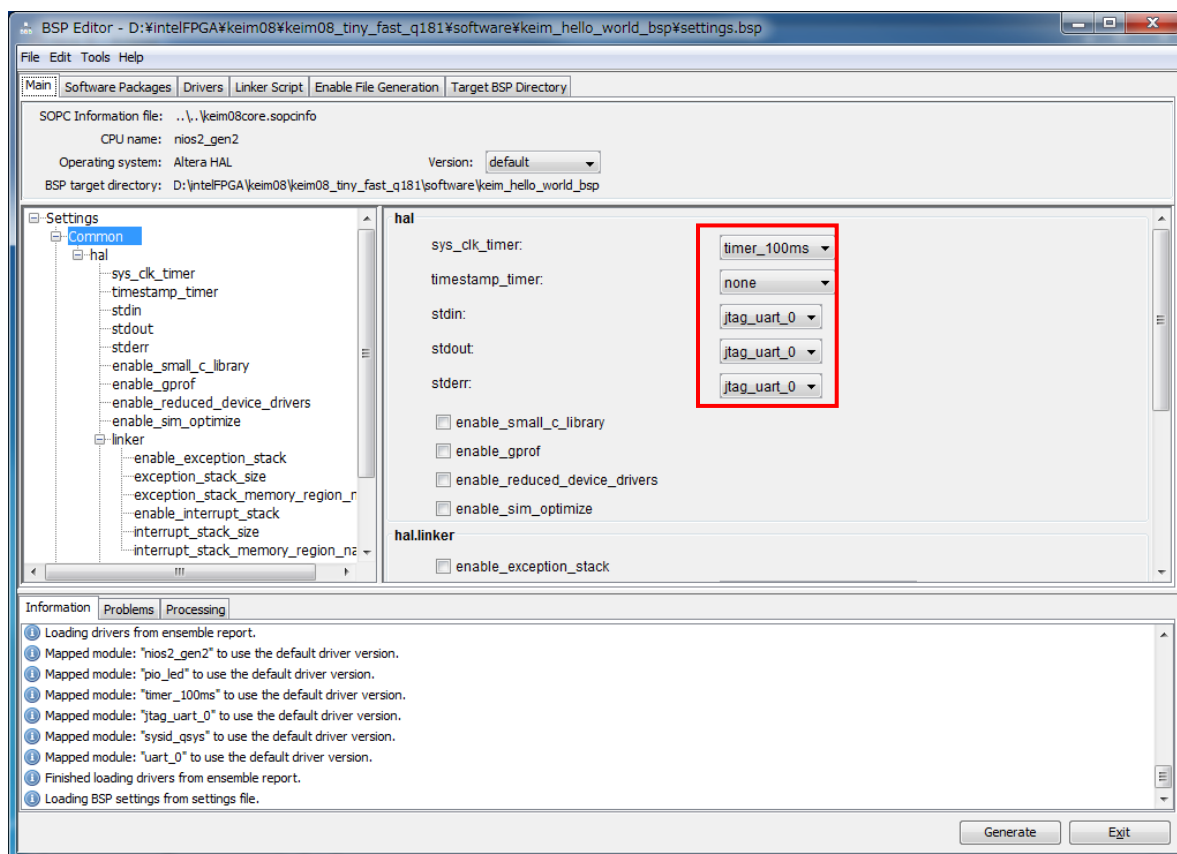
BSP Editor が起動しましたら、作成されたプロジェクトの settings.bsp ファイルを読み込みます。

File > Open >

本例では、

D:\intelFPGA\keim08\keim08_tiny_fast_q181\software\keim_hello_world_bsp\settings.bsp
を指定します。

④ メインの設定



BSP Editor 画面(Main タブ)

Main タブの赤枠の部分を設定することにより、システムクロックタイマーの指定や標準入出力先の選択を行えます。ここではタイマは timer_100ms、標準入出力は jtag_uart_0 とします。(デフォルトの標準出力が jtag_uart_0 になっています)

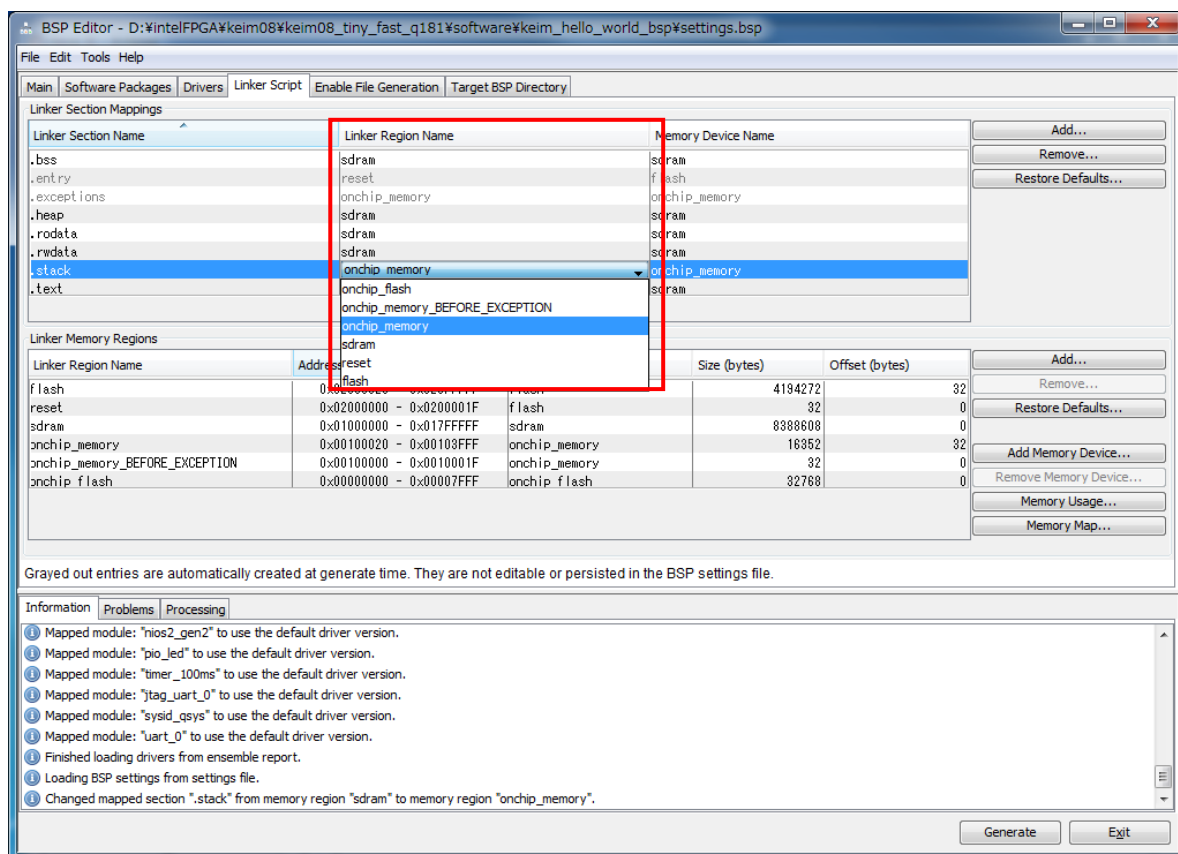
標準入出力の jtag_uart_0 を uart_0 に変更しますと、USB-UART を介して Virtual COM ポートに設定する事が出来ます。

<注意>

但し、JTAG 接続が、"USB-Blaster" の場合は、USB-UART と "USB-Blaster" は同一の PC を使用する事は出来ません(FTDI のドライバが競合する為)。

JTAG 接続が、"USB-Blaster II" の場合は、ドライバの競合がおきませんので、同一の PC で使用可能です。

② 使用メモリの選択



BSP Editor 画面(Linker Script タブ)

Linker Script の赤枠部分を設定することにより各セクションに使用するメモリを指定できます。
ここではスタックメモリを内蔵 onchip_memory に指定しています。

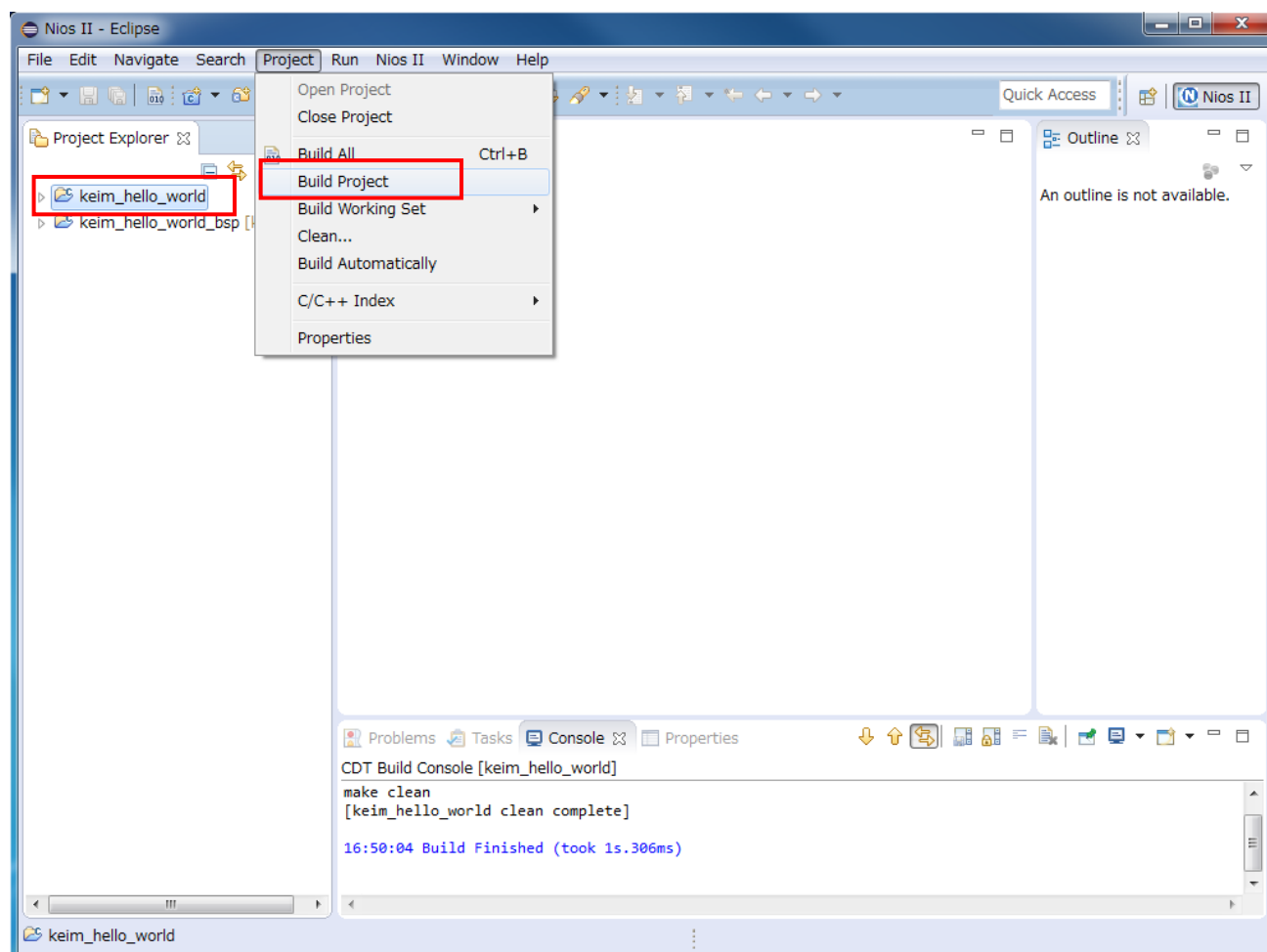
③ BSP の生成

設定が完了しましたら、File > save で上書き保存します。

[Generate] をクリックし BSP を再生成後、[Exit] をクリックし BSP Editor を終了します。

3.1.5. ソフトウェアプロジェクトのビルド

プロジェクト “Keim_hello_world” を選択後、Project> Build Project を実行します。



3.1.6. FPGA 内部回路イメージの download

本プロジェクト用の FPGA 内部回路イメージファイルは”keim08_top.sof” を使用します。先程の”keim08core.sopcinfo”と同様に下記フォルダに格納されています。

D:\¥intelFPGA¥keim08¥keim08_tiny_fast_q181¥output_files¥keim08_top.sof

スタートメニューより Quartus Prime Programmer を立ち上げます。スタートメニューより順に Intel FPGA 18.1.0.625 Lite Edition >

Quartus Prime Lite Edition 18.1.0.625 > Programmer (Quartus Prime 18.1)

または

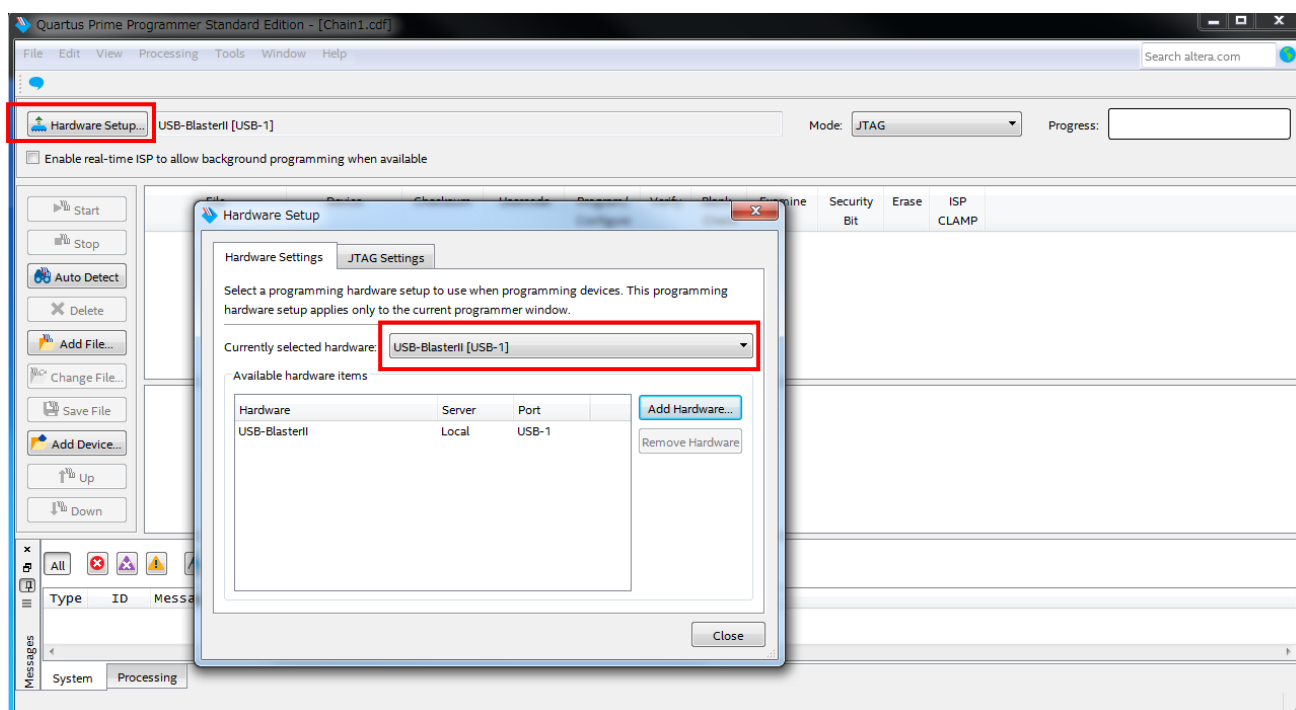
Intel FPGA 18.1.0.625 Standard Edition >

Quartus Prime Standard Edition 18.1.0.625 > Programmer (Quartus Prime 18.1)

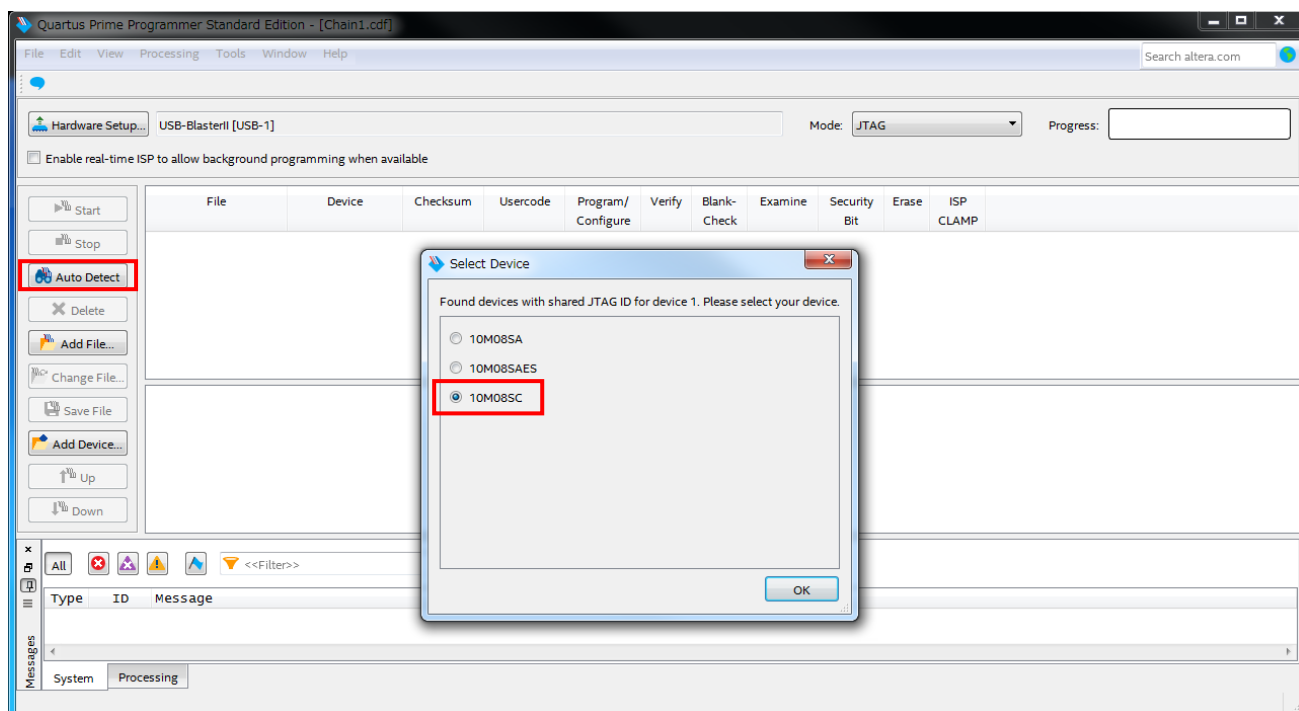
を選択し Quartus Prime Programmer を起動します。

① Intel FPGA download cable II の Hardware Setup

Hardware Setup > USB-Blaster II (USB-1) を選択します。



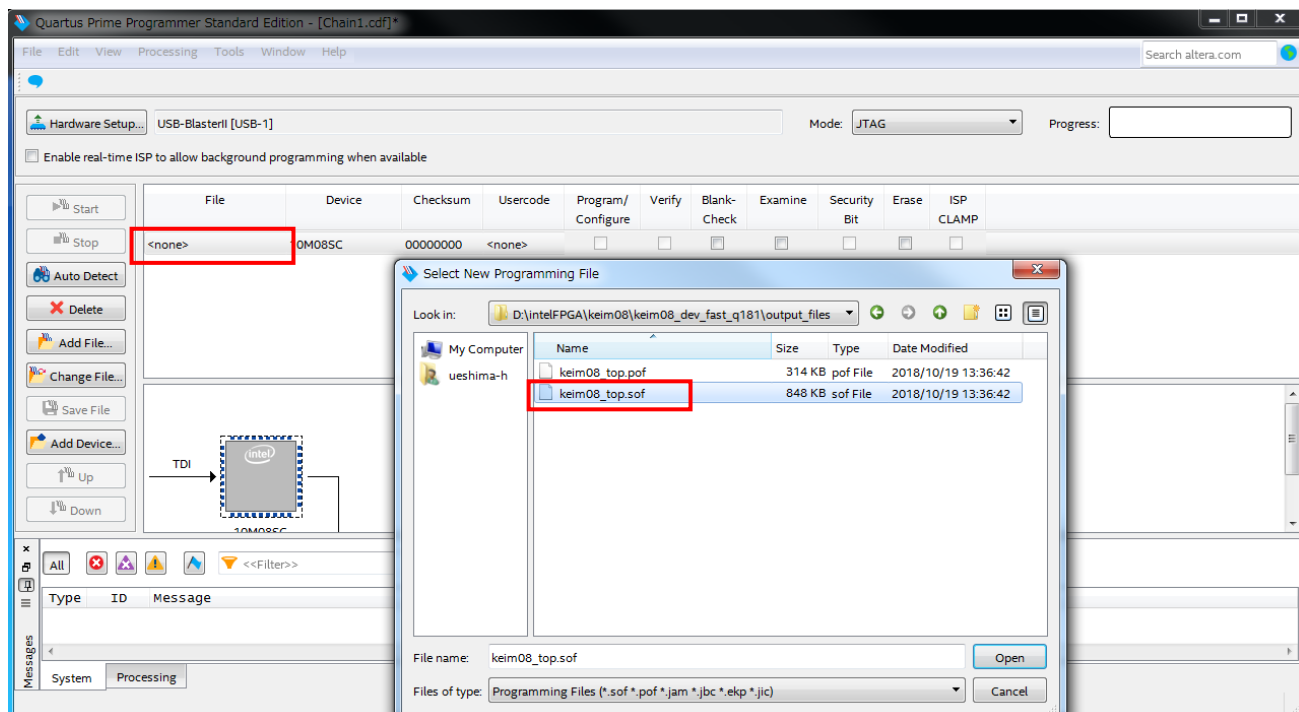
Auto Detect > 10M08SC を選択して[OK]



”keim08_top.sof”の設定

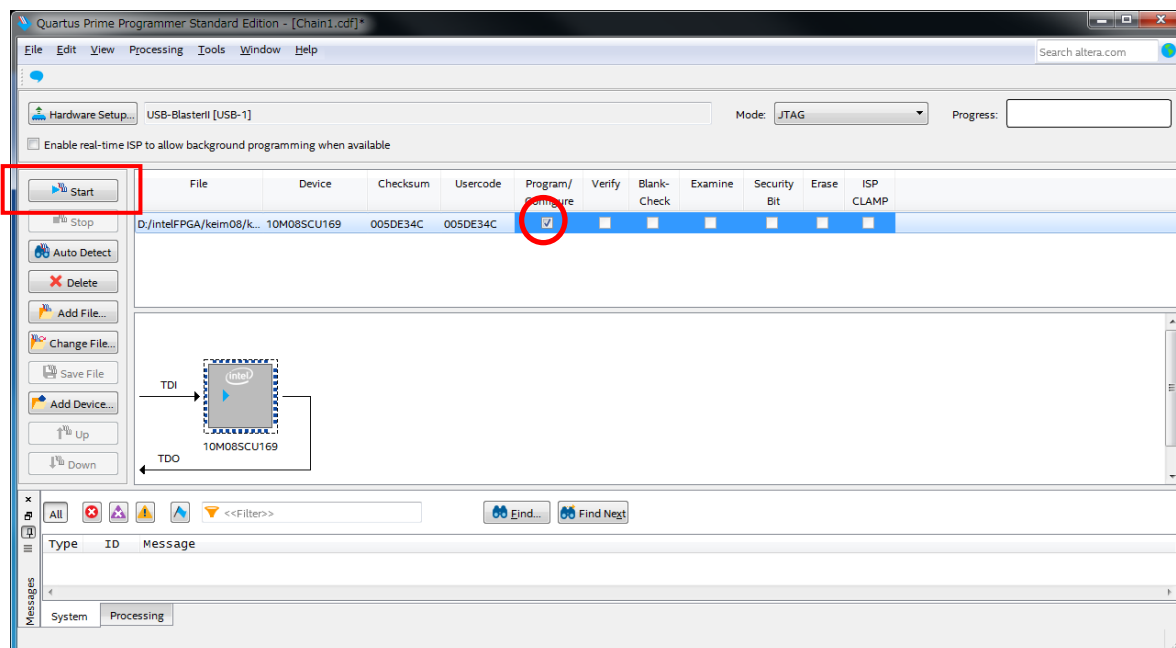
D:\intel\FPGA\keim08\keim08_tiny_fast_q181\output_files\keim08_top.sof

File<none> をダブルクリックして”keim08_top.sof”ファイルを選択して[Open]



② Download を実行

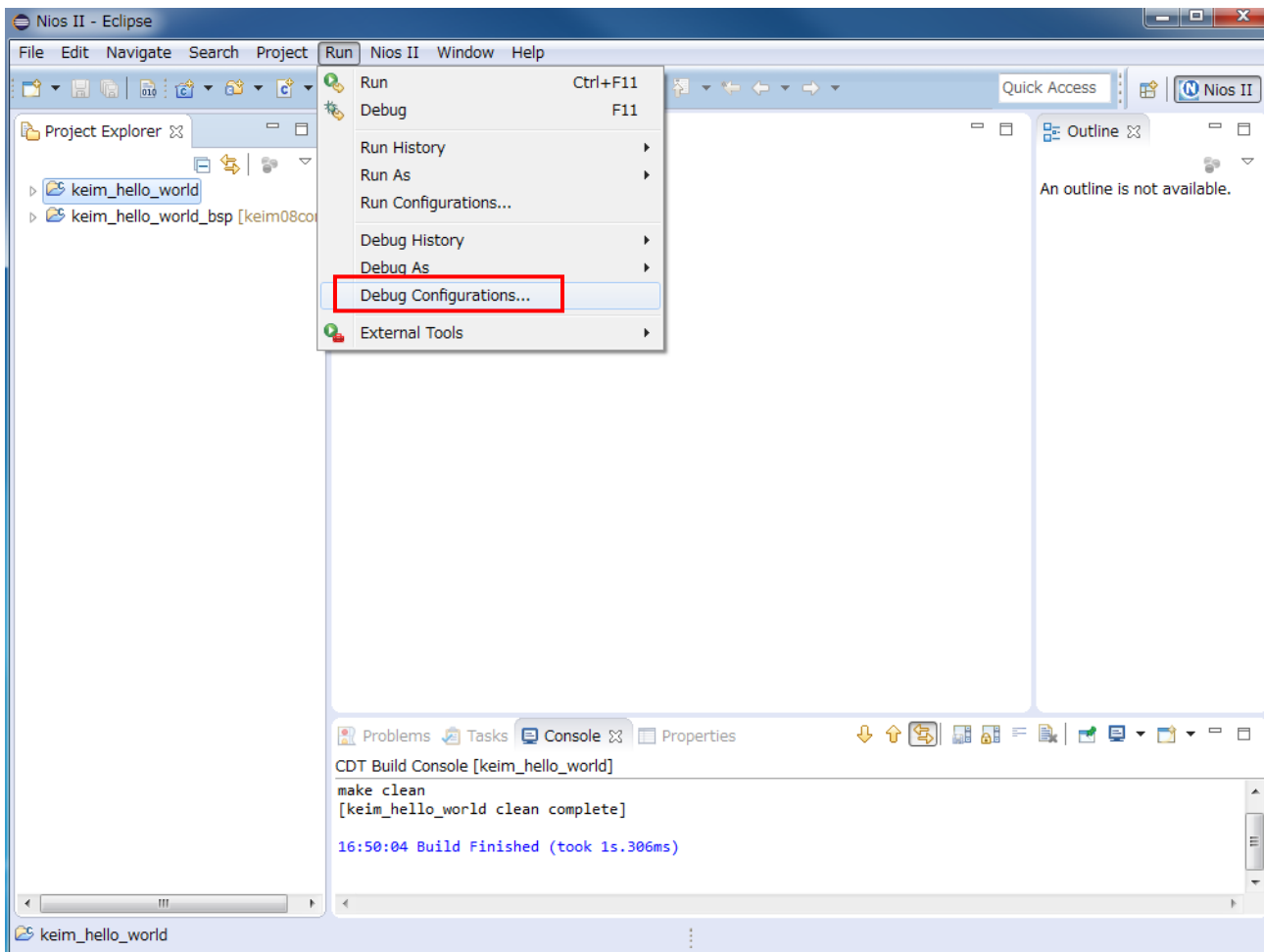
Program/Configure にチェックを入れて Start



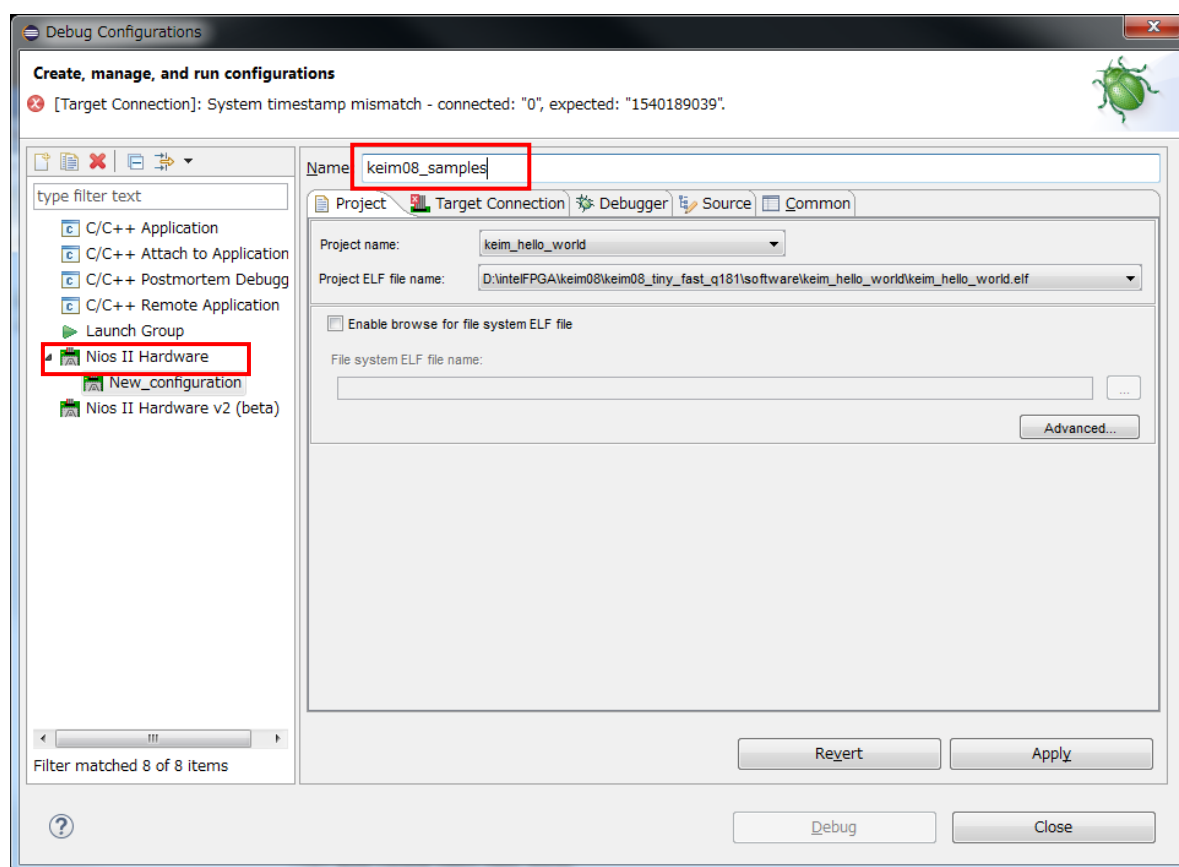
3.1.7. ソフトウェアプロジェクトの実行

3.1.5 項でビルドしたソフトウェアプロジェクトをメモリに download して実行します。

① Debug Configurations の設定



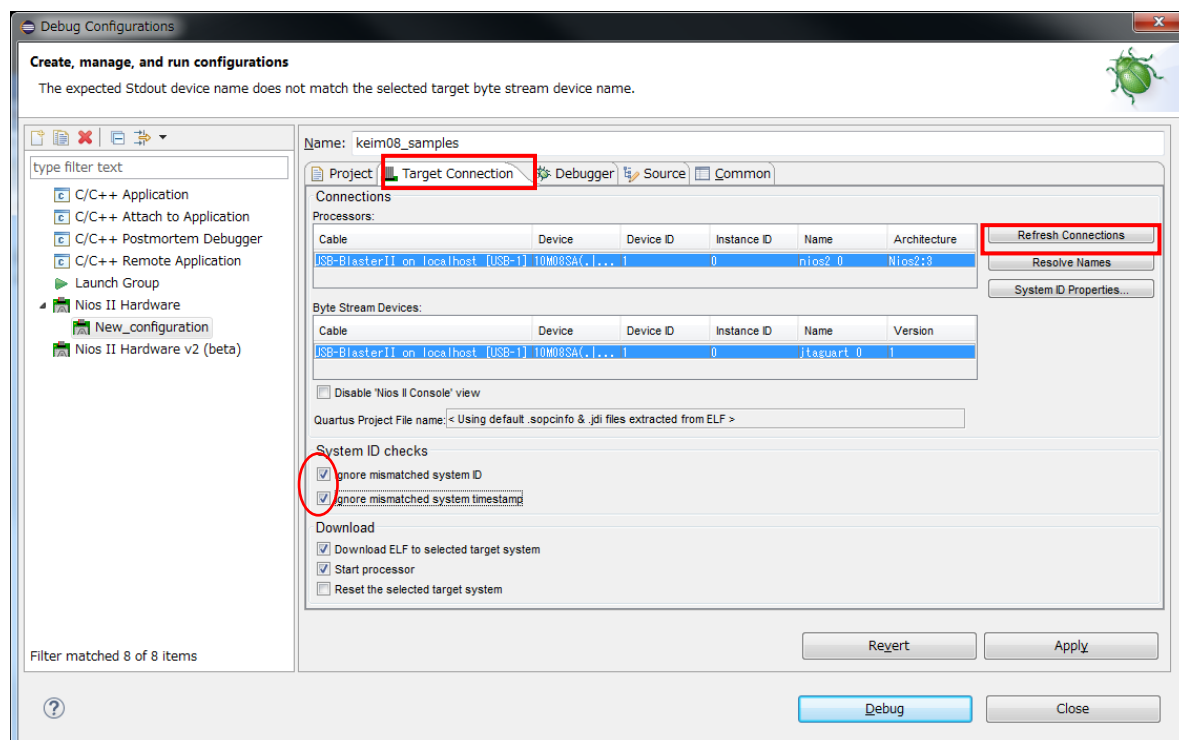
“Nios II Hardware” をダブルクリックしますと、New_configuration が現れますので、ここでは Name 例として keim08_samples とします。




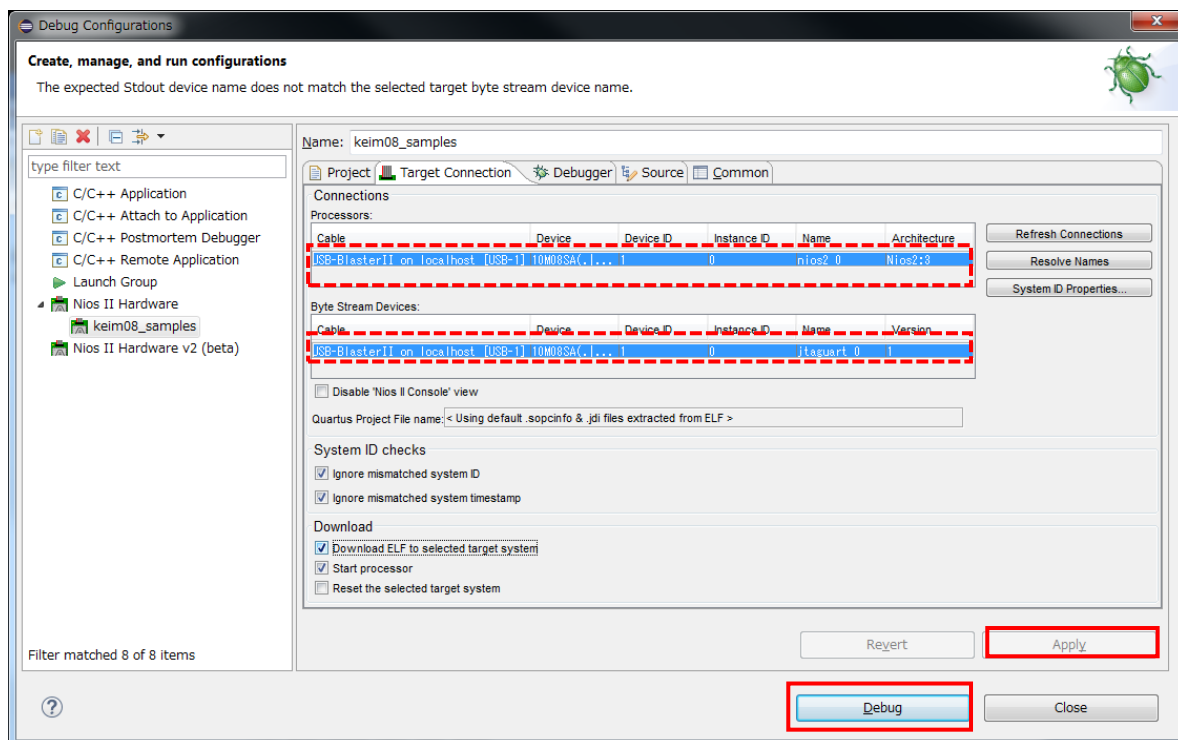
次に”Target Configuration” のタブを開き “System ID checks” の下記 2 項にチェックマークを付けます。

- Ignore mismatched system ID
- Ignore mismatched system timestamp

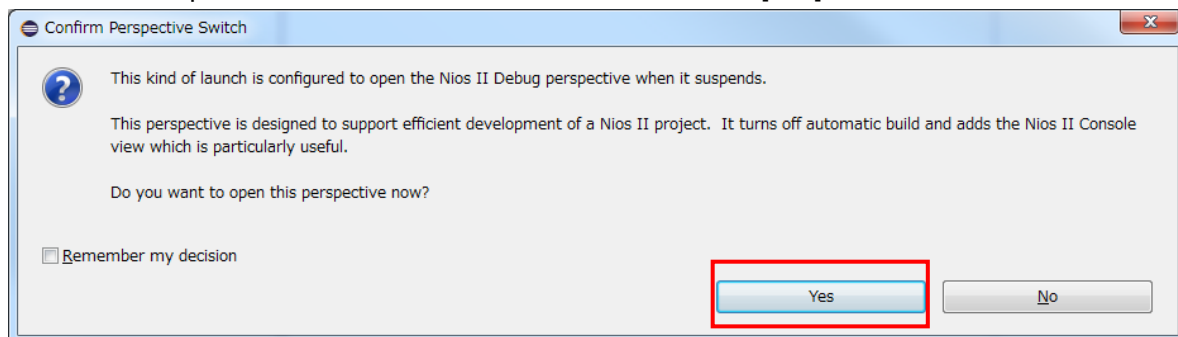
その後、[Refresh Connection] のボタンを押します。



Processors: / Bye Stream Devices: の項目 () がリフレッシュ (一旦消えて再度表示) されたら [Apply] ボタンを押してから [Debug] ボタンを押して Debugger を起動します。



下記 Confirm Perspective Switch のダイアログが現れましたら [Yes] を押してください。

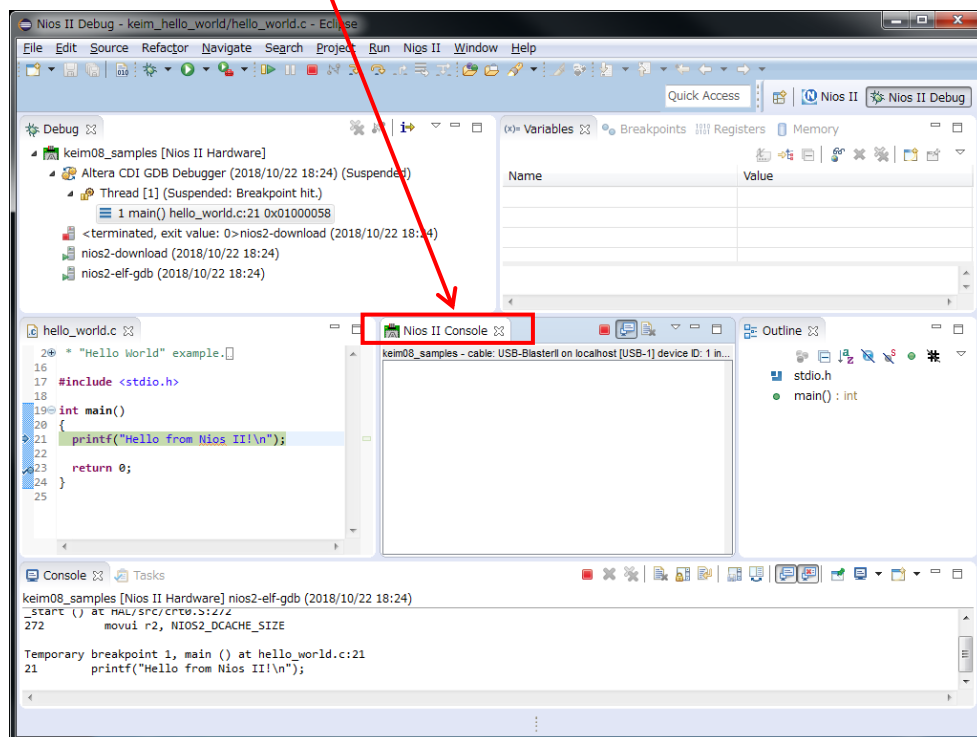
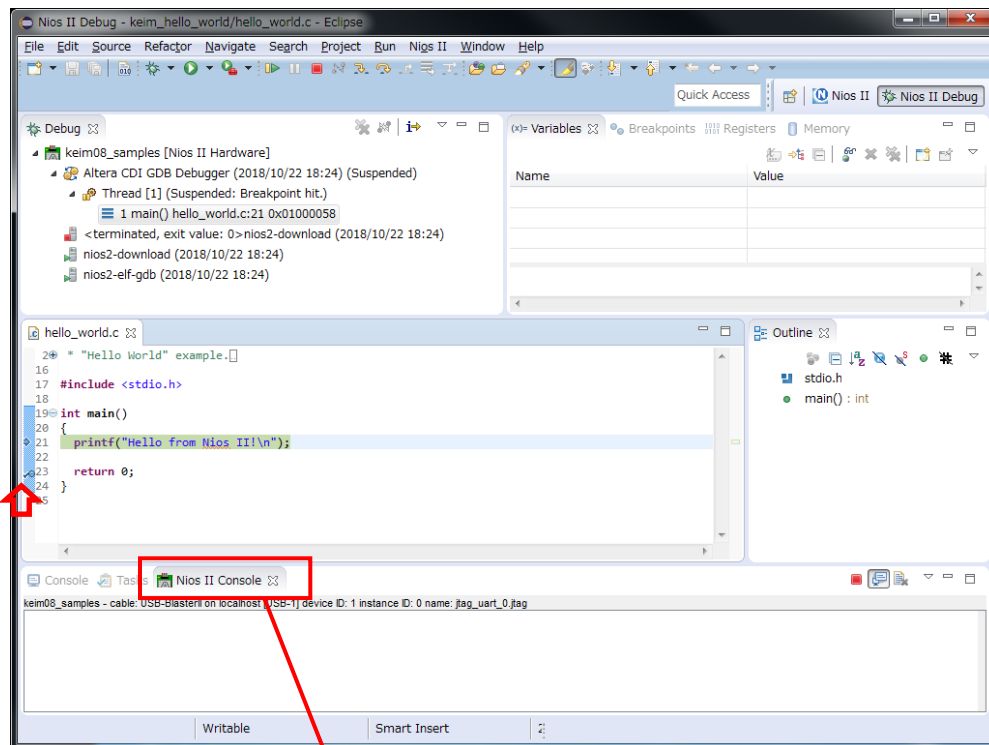


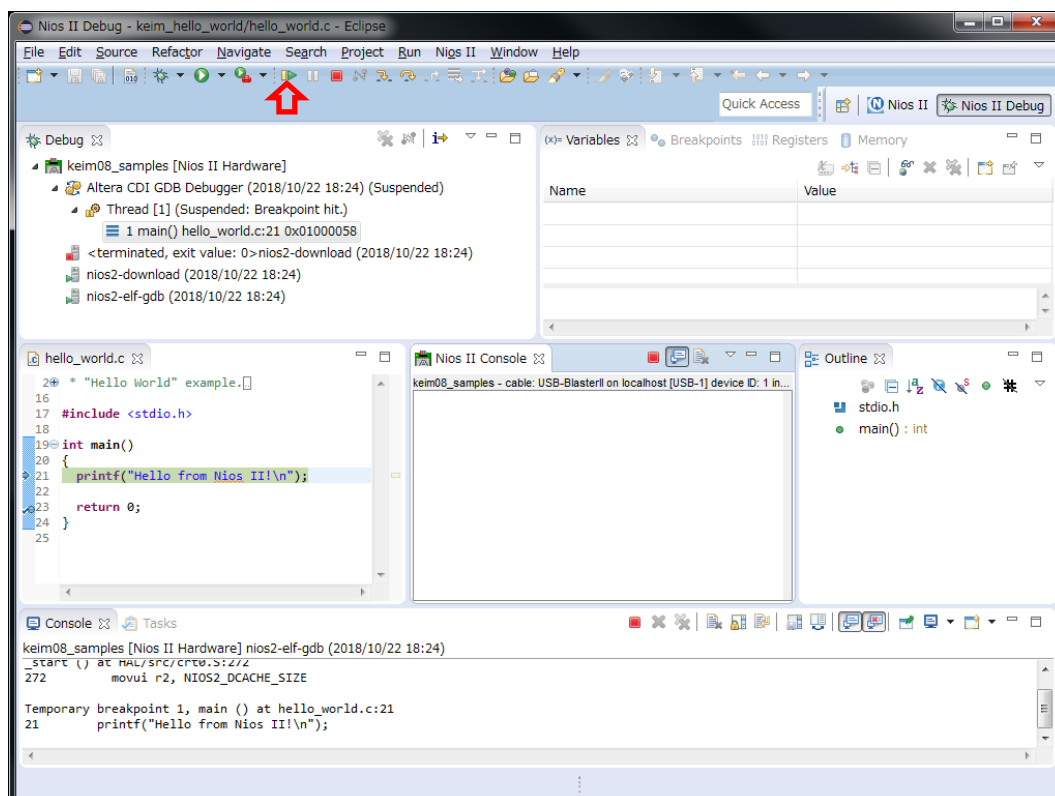
② Breakpoint を設定して実行


NiosII Debug が起動したら、 hello_world.c の

23 return 0;

の行の 23 の文字の少し左側にマウスカーソルを持っていきダブルクリックしますと Breakpoint のマークが付きます。次に Nios II Console をドラッグアンドドロップで見やすい位置に移動しておきます。






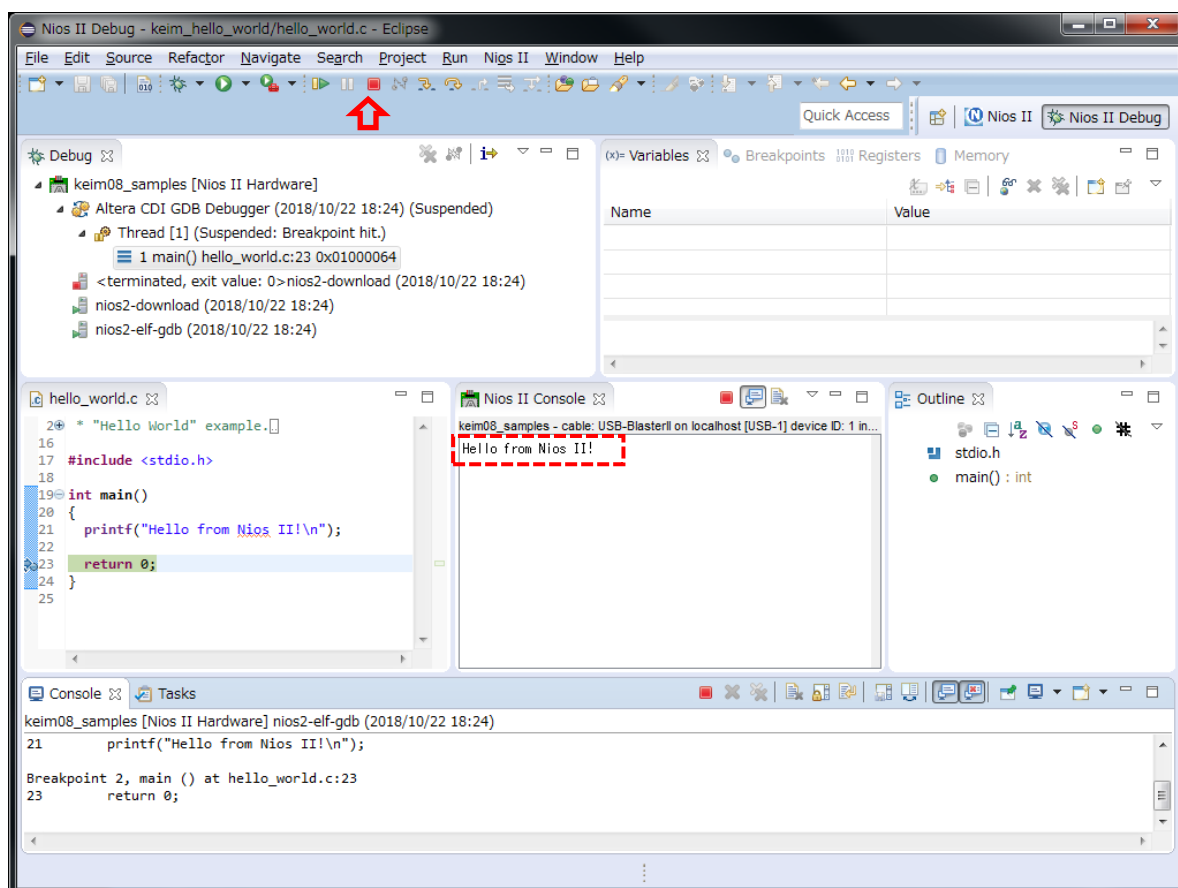
その後、 ボタンを押してソフトウェアを実行します。

NiosII Console ウィンドウに

Hello from Nios II!

が表示されて breakpoint で止まります。

Terminate  ボタンを押して Debug を終了します。



```
20 * "Hello World" example.[]  
16  
17 #include <stdio.h>  
18  
19 int main()  
20 {  
21     printf("Hello from Nios II!\n");  
22  
23     return 0;  
24 }  
25
```

```
keim08_samples - cable: USB-BlasterI on localhost [USB-1] device ID: 1 in...  
Hello from Nios II!
```

Outline

- stdio.h
- main() : int

```
Console Tasks  
keim08_samples [Nios II Hardware] nios2-elf-gdb (2018/10/22 18:24)  
21     printf("Hello from Nios II!\n");  
  
Breakpoint 2, main () at hello_world.c:23  
23     return 0;
```

3.2. LED 点滅プログラム

2.2 項で解凍した下記フォルダに、LED 点滅プログラム “hello_led.c” がありますので、それを新規作成する Application プロジェクトにコピーします。

D:\intelFPGA\keim08\keim08_tiny_fast_q181\software\examples\hello_led.c

主な作業手順としては下記の通りとなります。

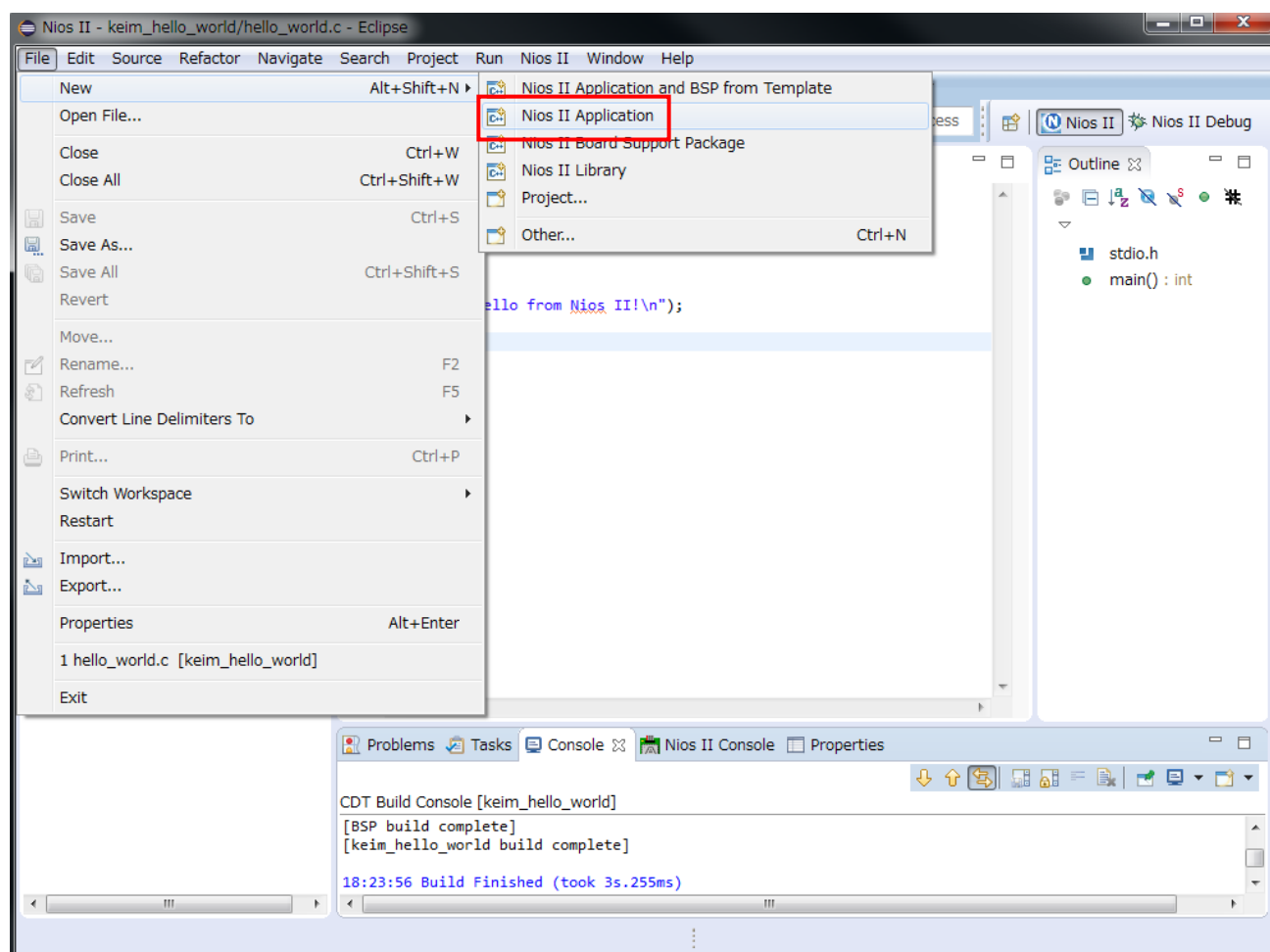
- ① 新規 Application プロジェクトの作成し、BSP を Hello World 表示プログラムで作成した BSP に設定
- ② hello_led.c をコピー
- ③ プロジェクトのビルド
- ④ プログラムの実行

以降の説明に従ってプログラムの作成及び実行を行ってください。

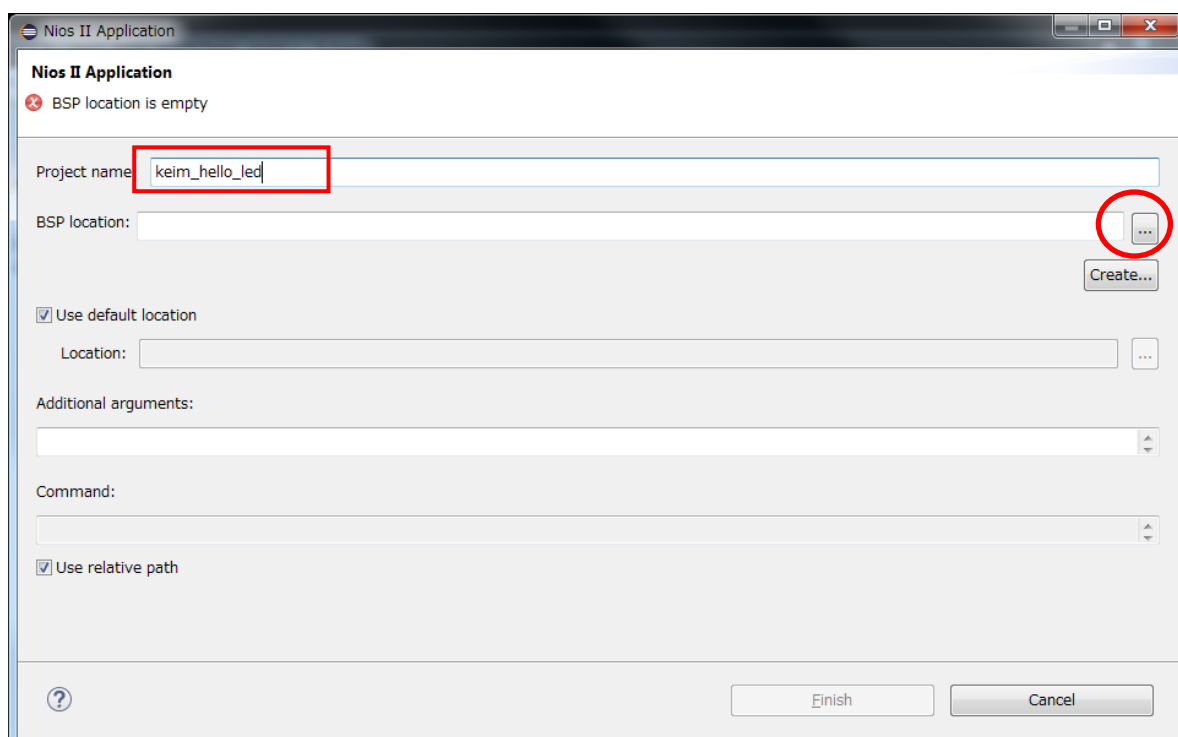
3.2.1. 新規 Application プロジェクト作成と BSP の設定


LED 点滅プログラム用に新規 Application プロジェクトを作成し、BSP は先程の hello_world の BSP を設定します。 Nios II SBT のウィンドウに戻り、

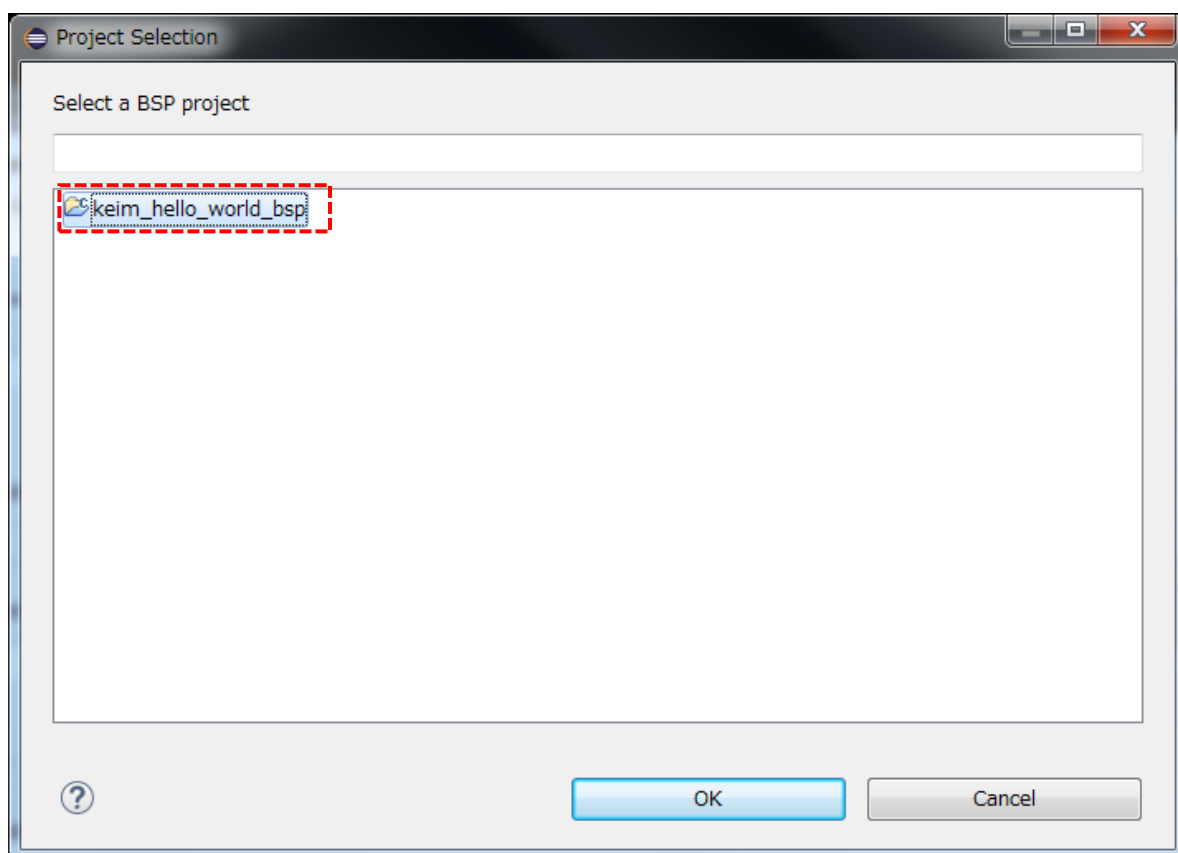
File > New > Nios II Application を選択します。



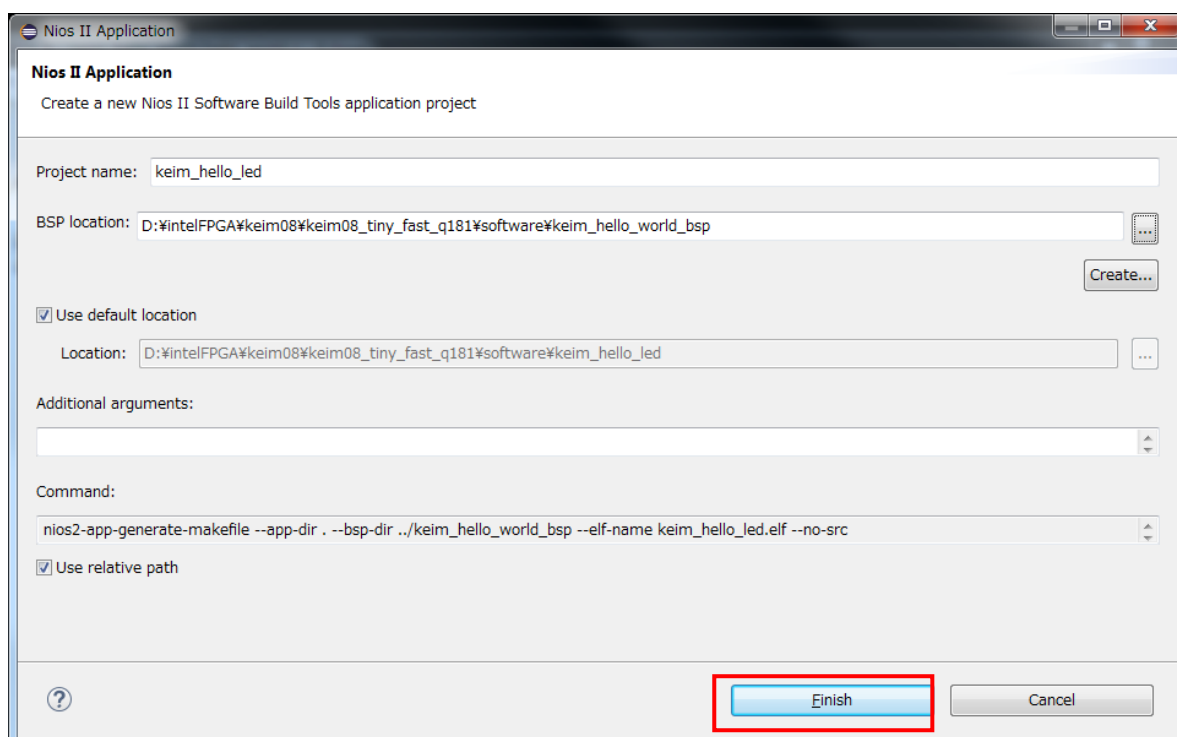
Project name : ここでは keim_hello_led とします。



BSP location: の右端の  をクリックして先程の keim_Hello_world_bsp を選択します。



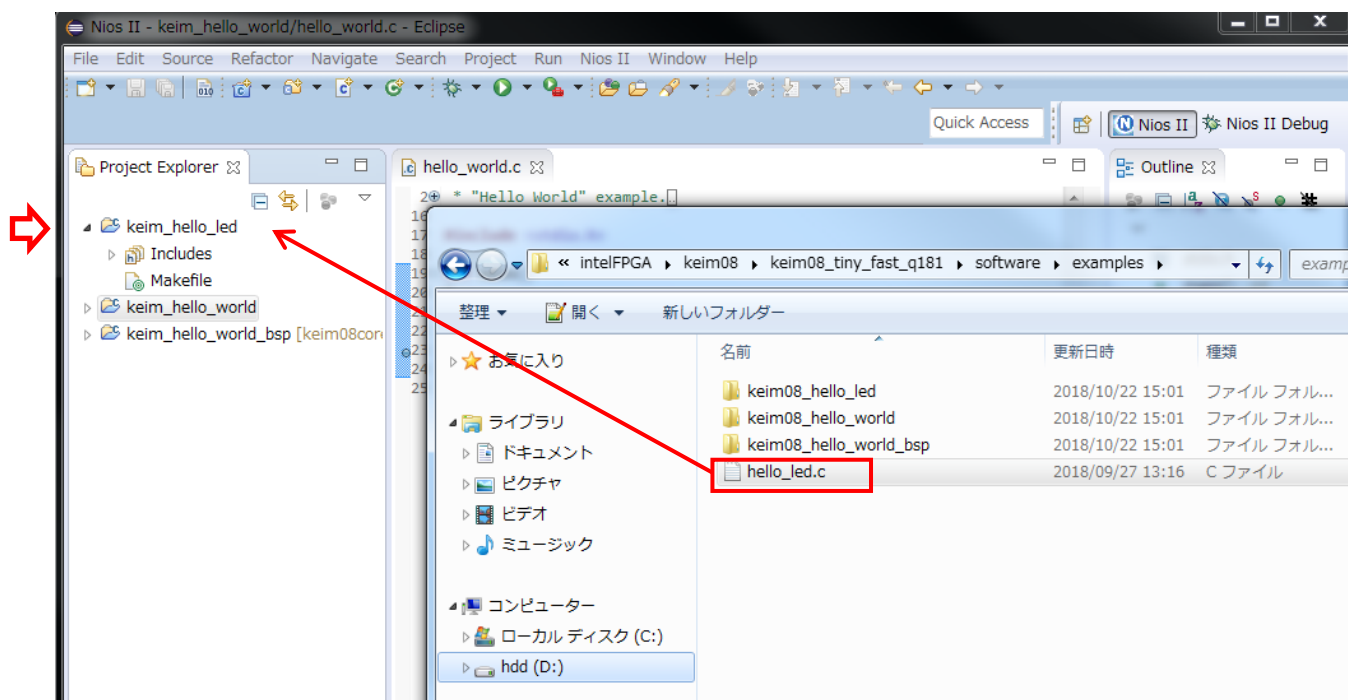
下記設定になりましたら [Finish] します。



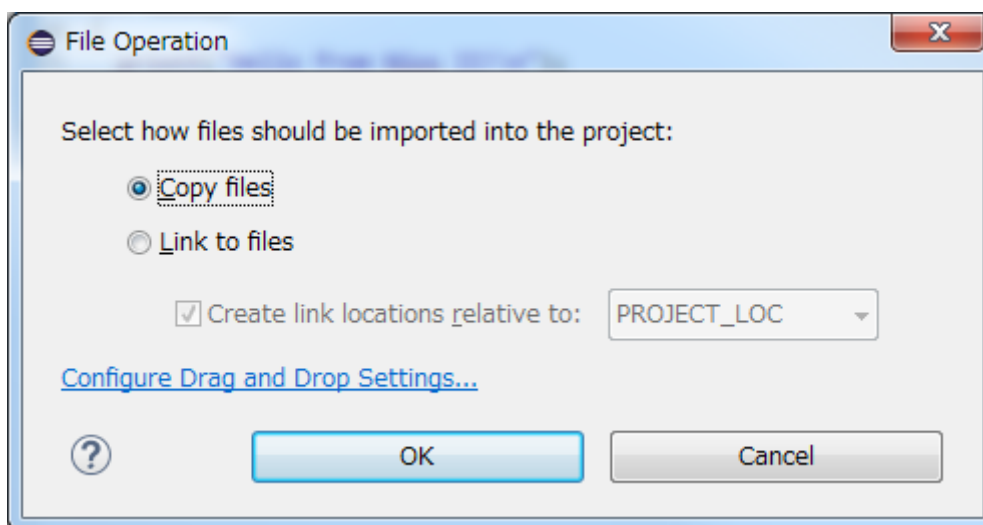
3.2.2. hello_led.c をコピー

エクスプローラで下記 “hello_led.c” を選択し、Drag and drop で keim_hello_led のプロジェクトにコピー

D:\intelFPGA\keim08\keim08_tiny_fast_q181\software\examples\hello_led.c

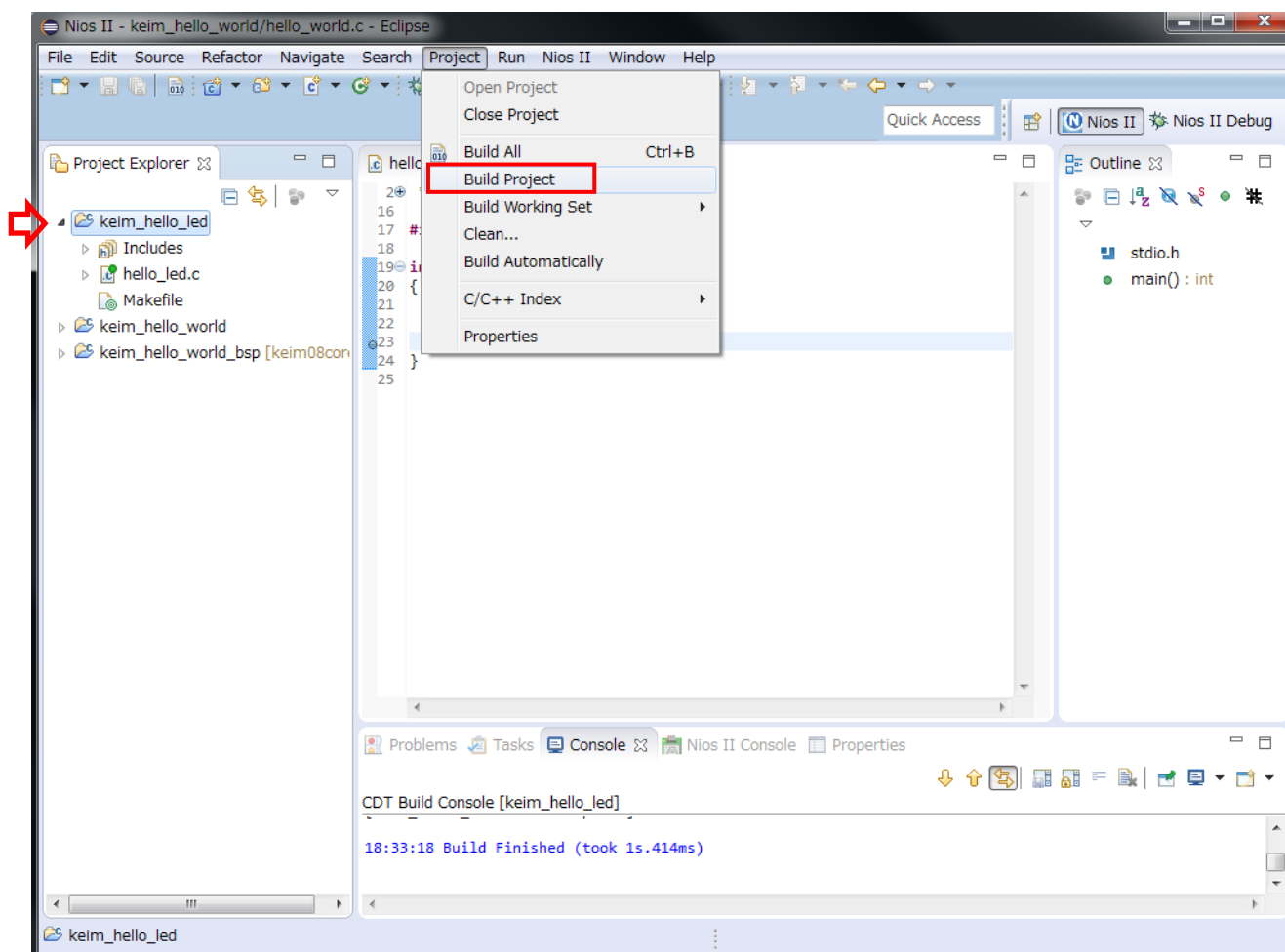


File Operation のダイアログボックスが現れましたら、Copy file を選択して[OK]



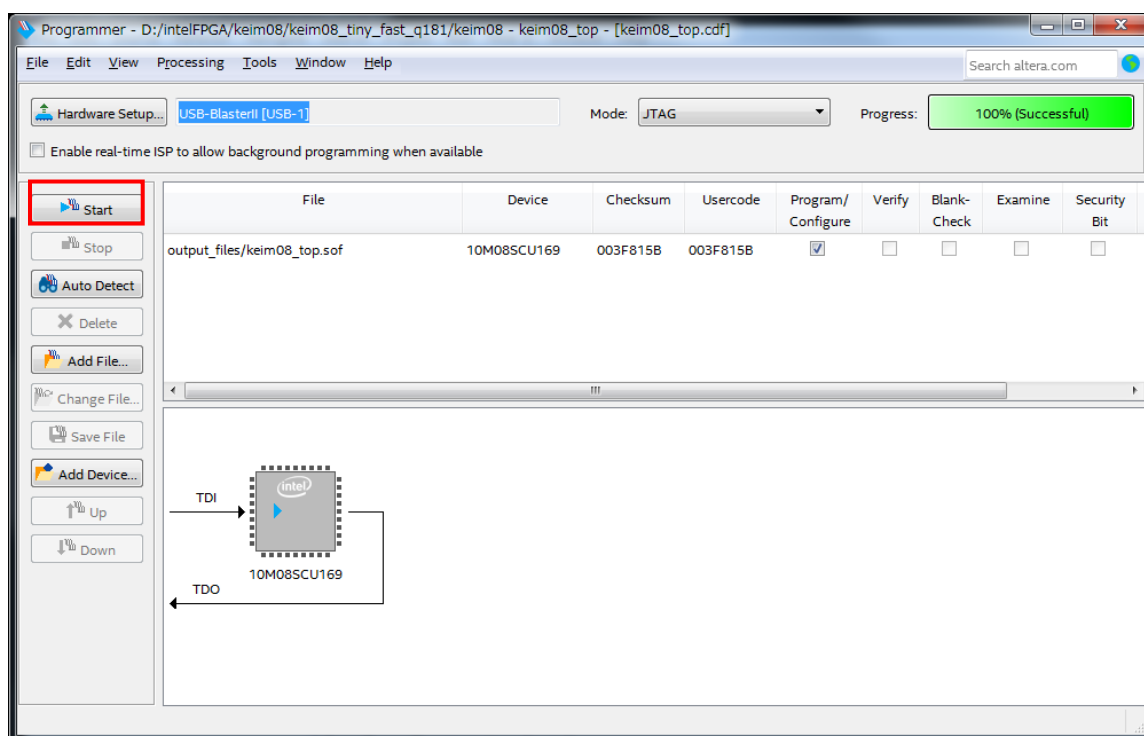
3.2.3. プロジェクトのビルド

keim_hello_led のプロジェクトを選択し、Project > Build Project を実行

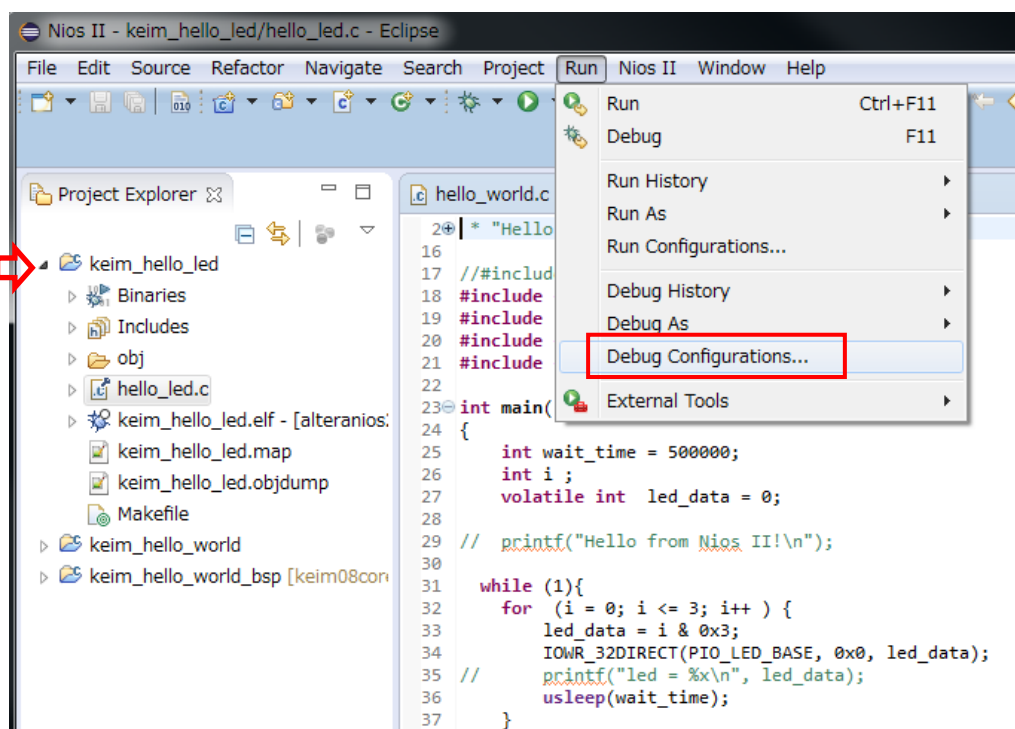


3.2.4. プログラムの実行

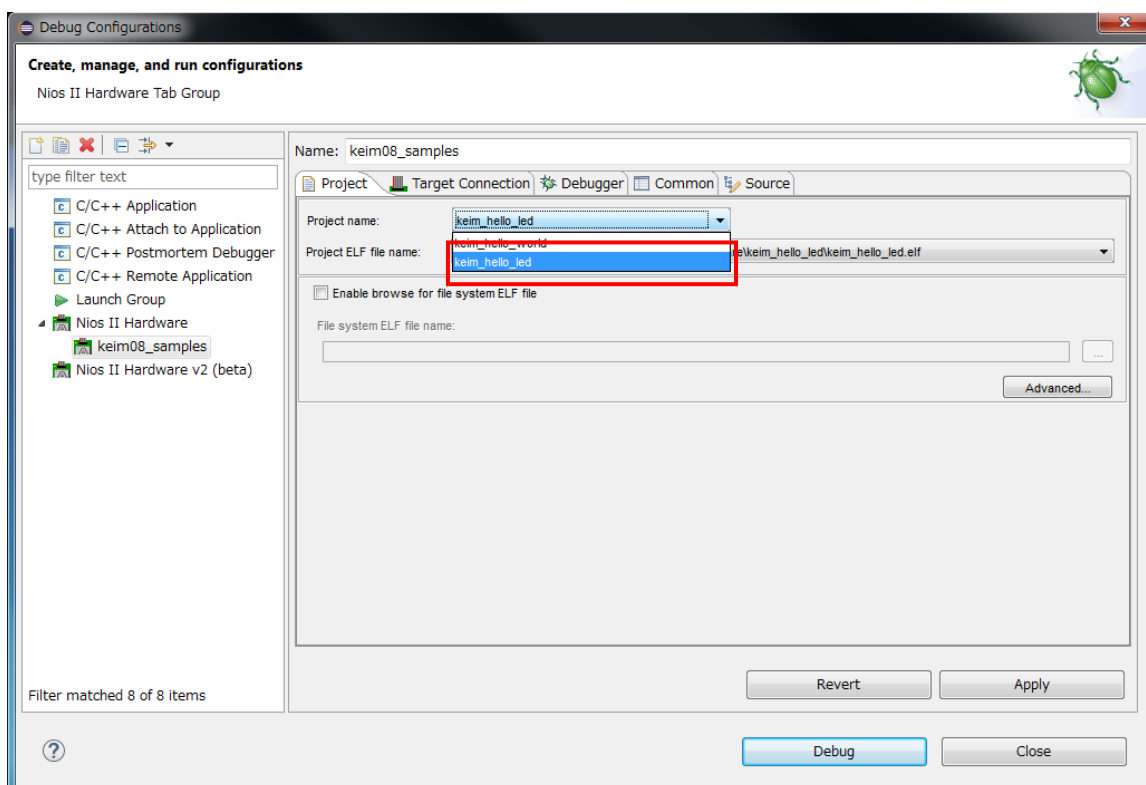
- ① hello_world と同様、Quartus Prime Programmer (Hello_world の時の設定のまま) "keim08_top.sof" を download します。



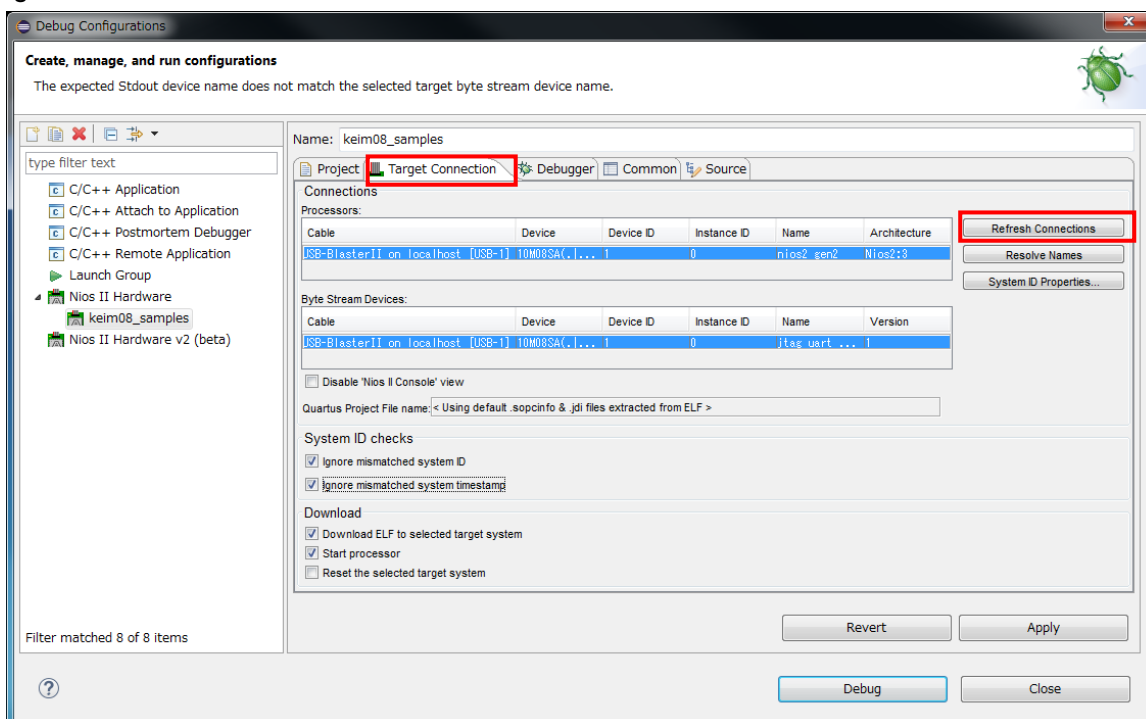
- ② keim_hello_led を選択して Run > Debug Configurations... を選択



③ Project name: を keim_hello_led を選択

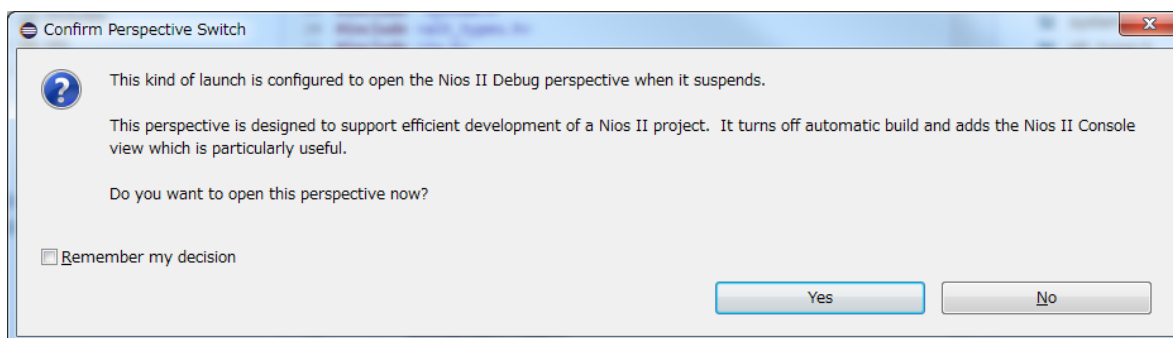


④ Target Connection のタブを開いて、Refresh Connections を実行

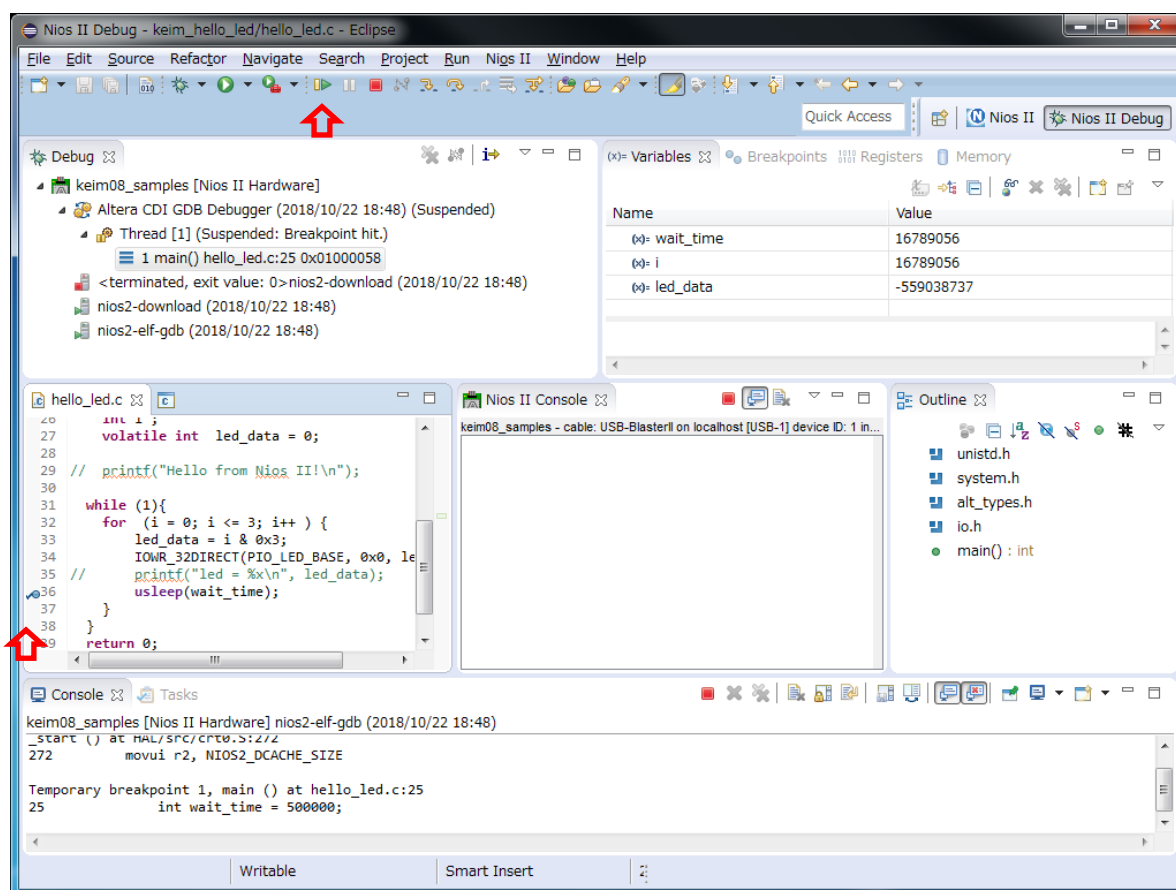


Refresh が完了後、[Apply] > [Debug] を実行


> Confirm Perspective Switch [Yes] を実行



- ⑤ Debug が起動したら、
36 usleep(wait_time); の行に breakpoint を設定



その後、  ボタンを押してソフトウェアを実行します。

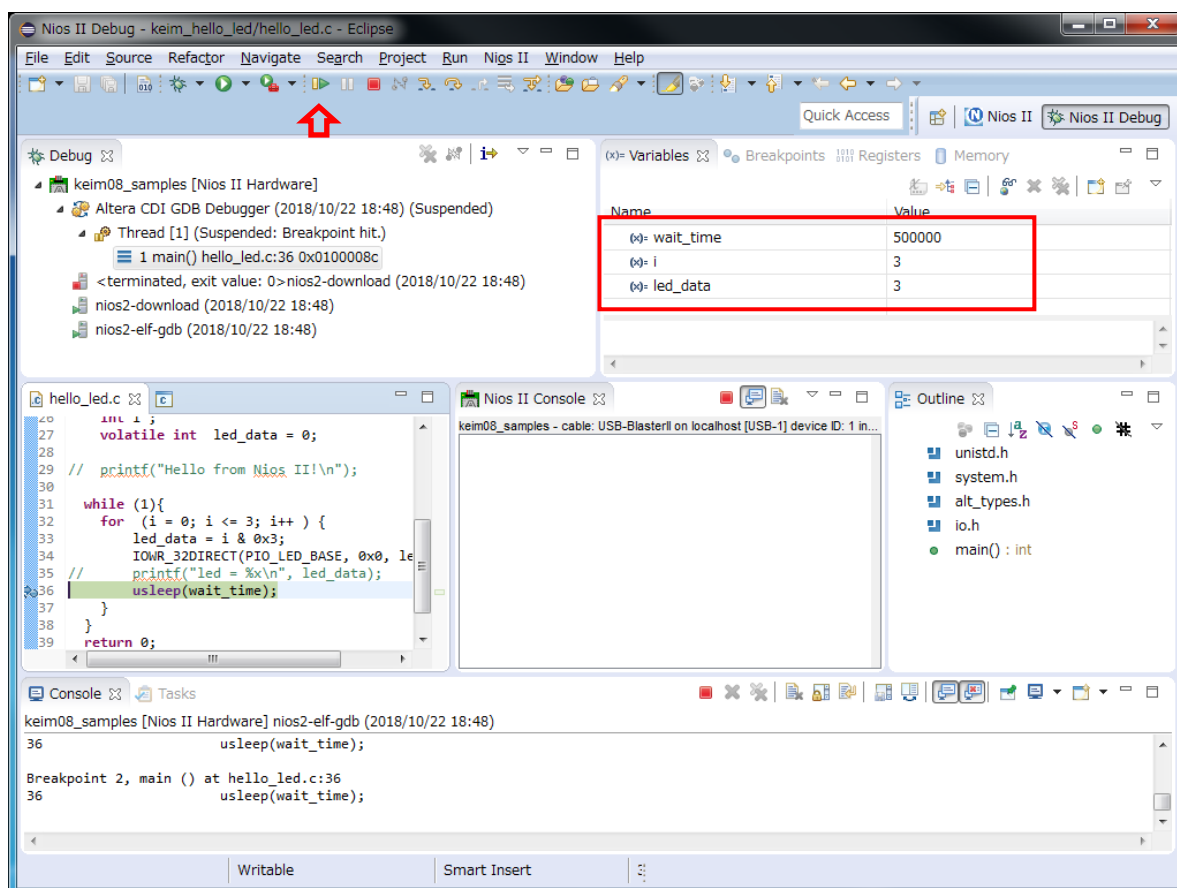
Breakpoint で停止する毎に、LED の点灯と NiosII console へのメッセージを確認し、 ボタンを押してソフトウェアを継続実行繰り返します。


led_data = 0 : 両方消灯

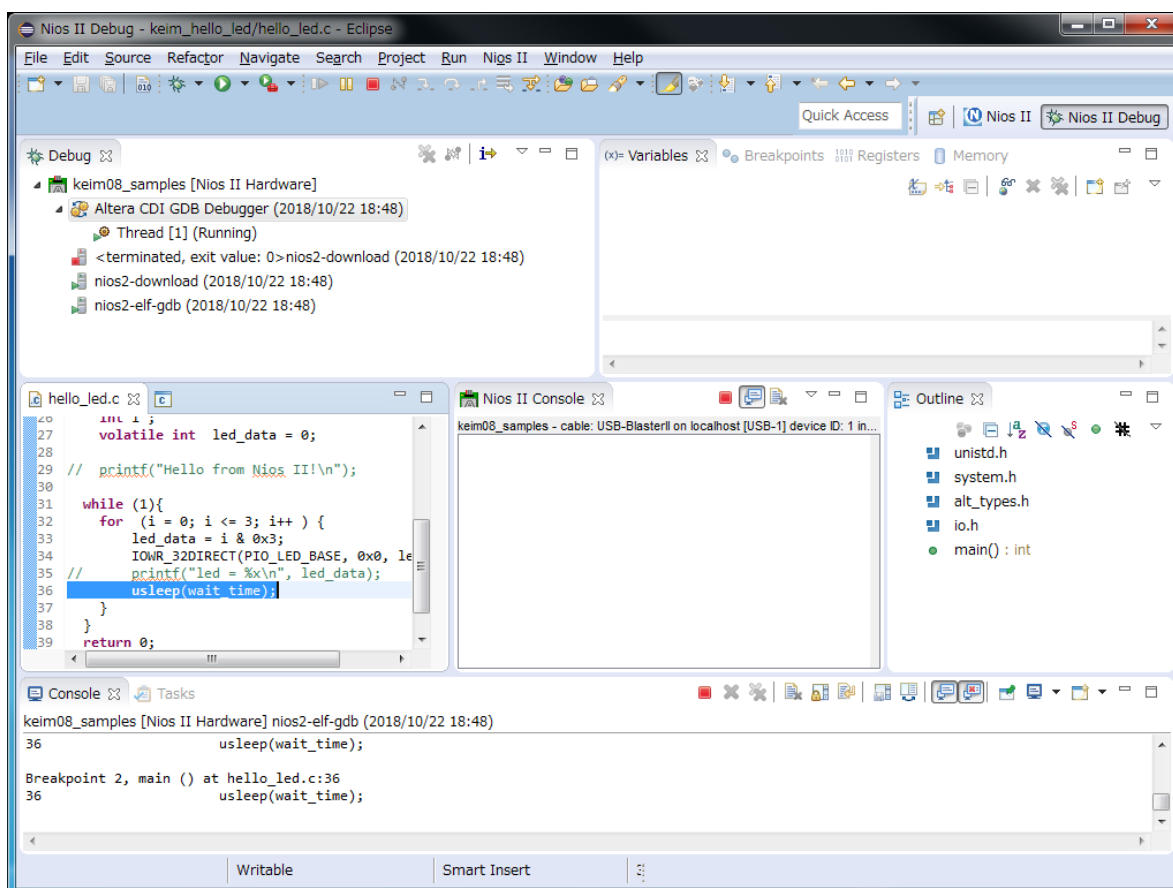
led_data = 1 : LED2 だけ点灯

led_data = 2 : LED1 だけ点灯

led_data = 3 : 両方点灯



一通り動作が確認出来ましたら、Breakpoint を外して、 ボタンを押してソフトウェアを連続動作させます。



ボタンで一時停止



ボタンで terminate 出来ます。

4. FPGA デザインの書き込み

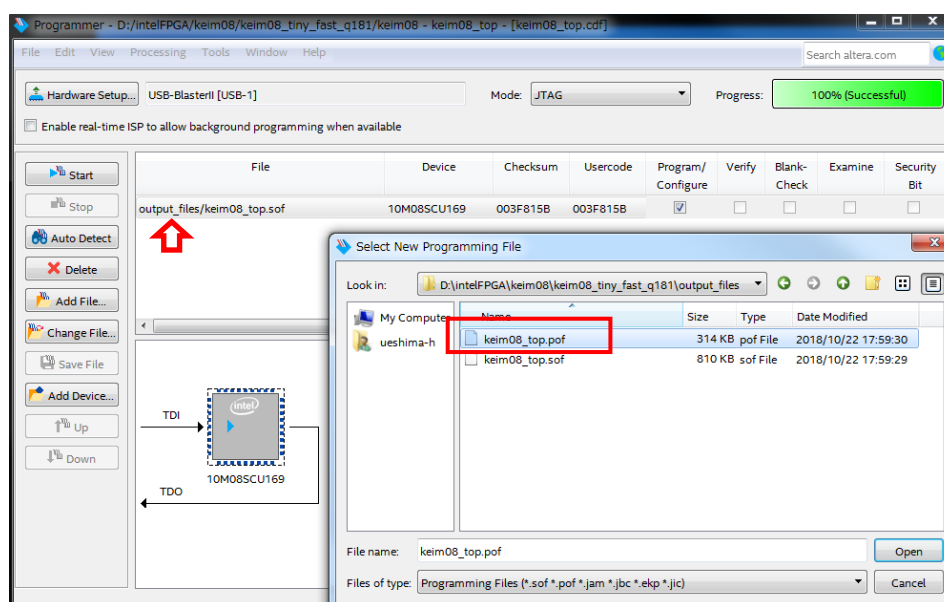
FPGA デザインをインテル® MAX® 10 FPGA 内蔵の Flash ROM へ書き込みます。

“2.3 KEIm08 用 FPGA サンプルデザインを開発用 PC にコピー” 項でコピーしたファイルの中に書き込みファイルが含まれていますので、それを Quartus Prime Programmer を使用して書き込みます。

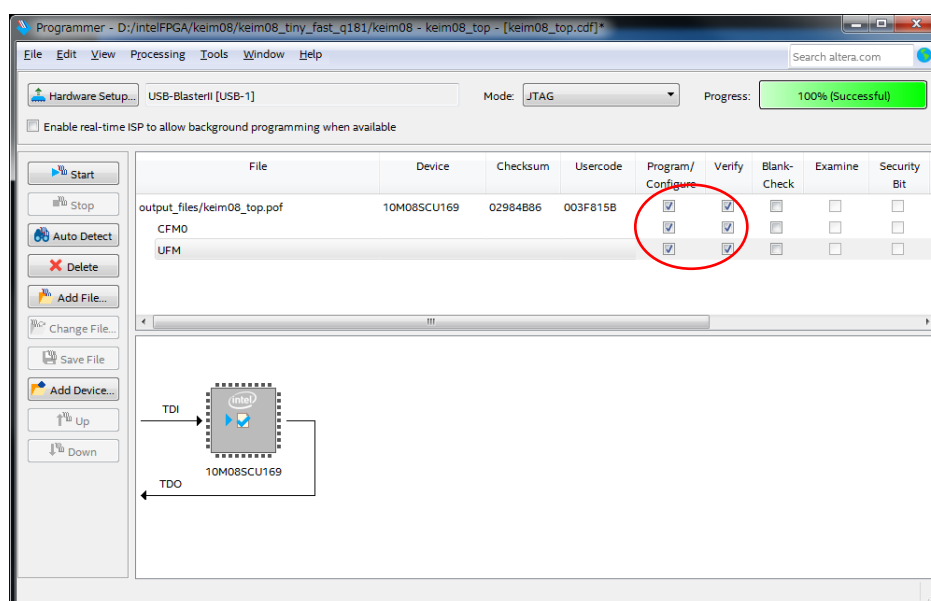
4.1. Quartus Prime Programmer での書き込み

Quartus Prime Programmer の設定は、download する .sof ファイルになっているかと思しますので、これを Flash memory 書き込みファイルの .pof ファイルに変更します。

“keim08_top.sof”ファイルをダブルクリックしますと下図の様に Select New Programming File のダイアログボックスが現れますので、“keim08_top.pof”を選択し、[Open] を押します。



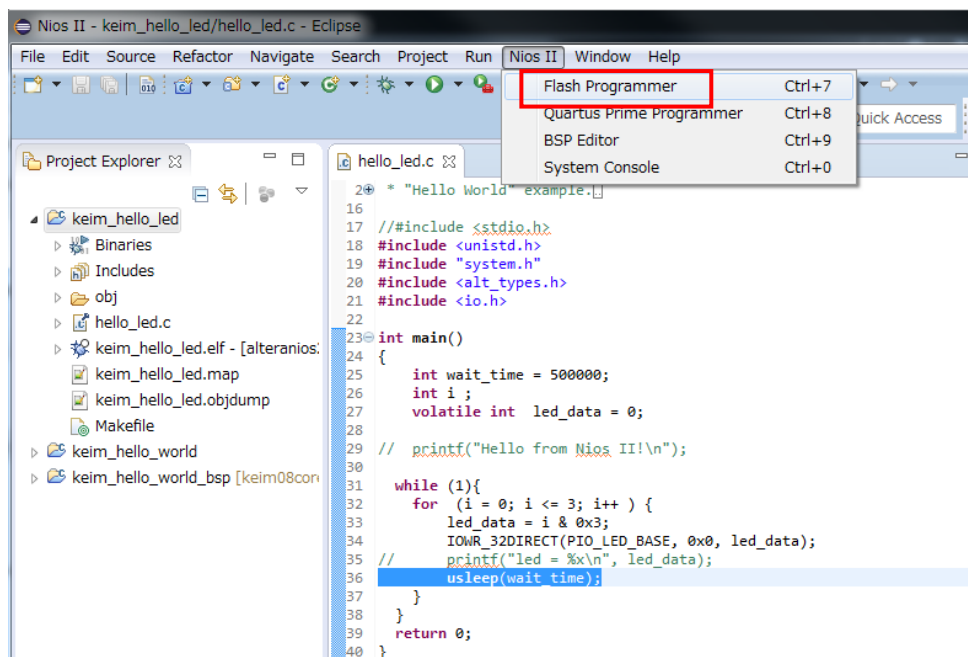
書き込みファイルが設定されますので、Program/Configure と Verify にチェックマークを設定し、[Start] を押しますと書き込みが実行されます。



Progress が 100%(Successful) となれば完了です。

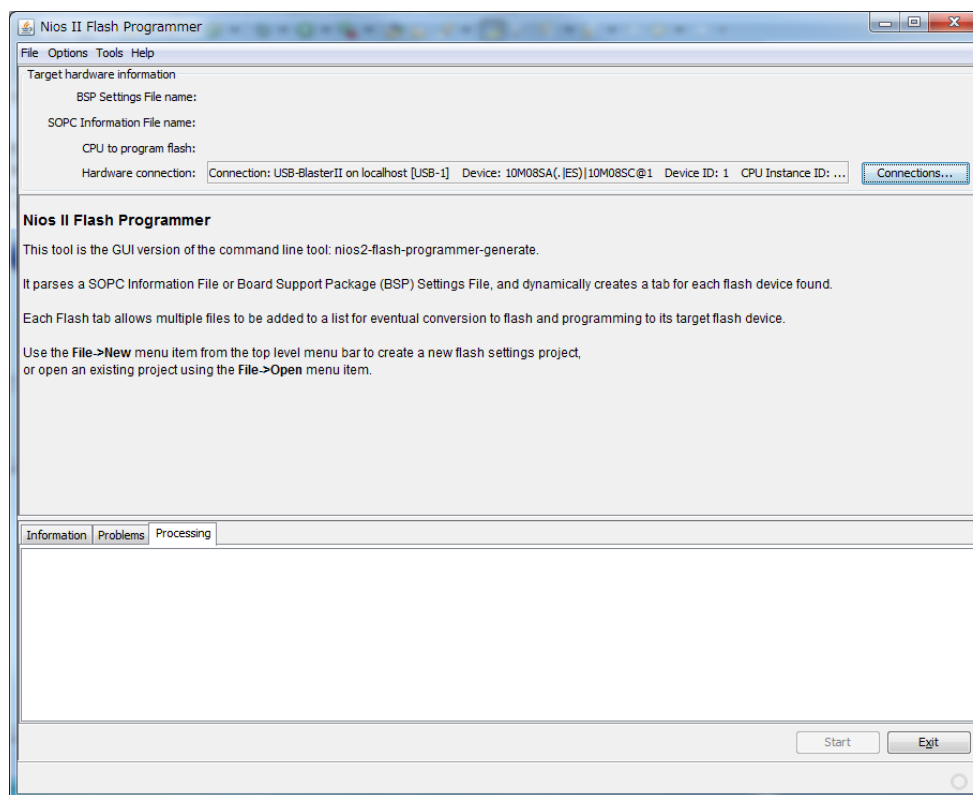
5. プログラムの ROM 化及び Flash ROM 書き込み

プログラムの ROM 化及び Flash ROM への書き込みは Nios II EDS の Flash Programmer で行います。下記の手順に沿って実行してください。



① Flash Programmer の起動

Nios II SBT のメニューから Nios II > Flash Programmer を選択し、Flash Programmer を起動します。

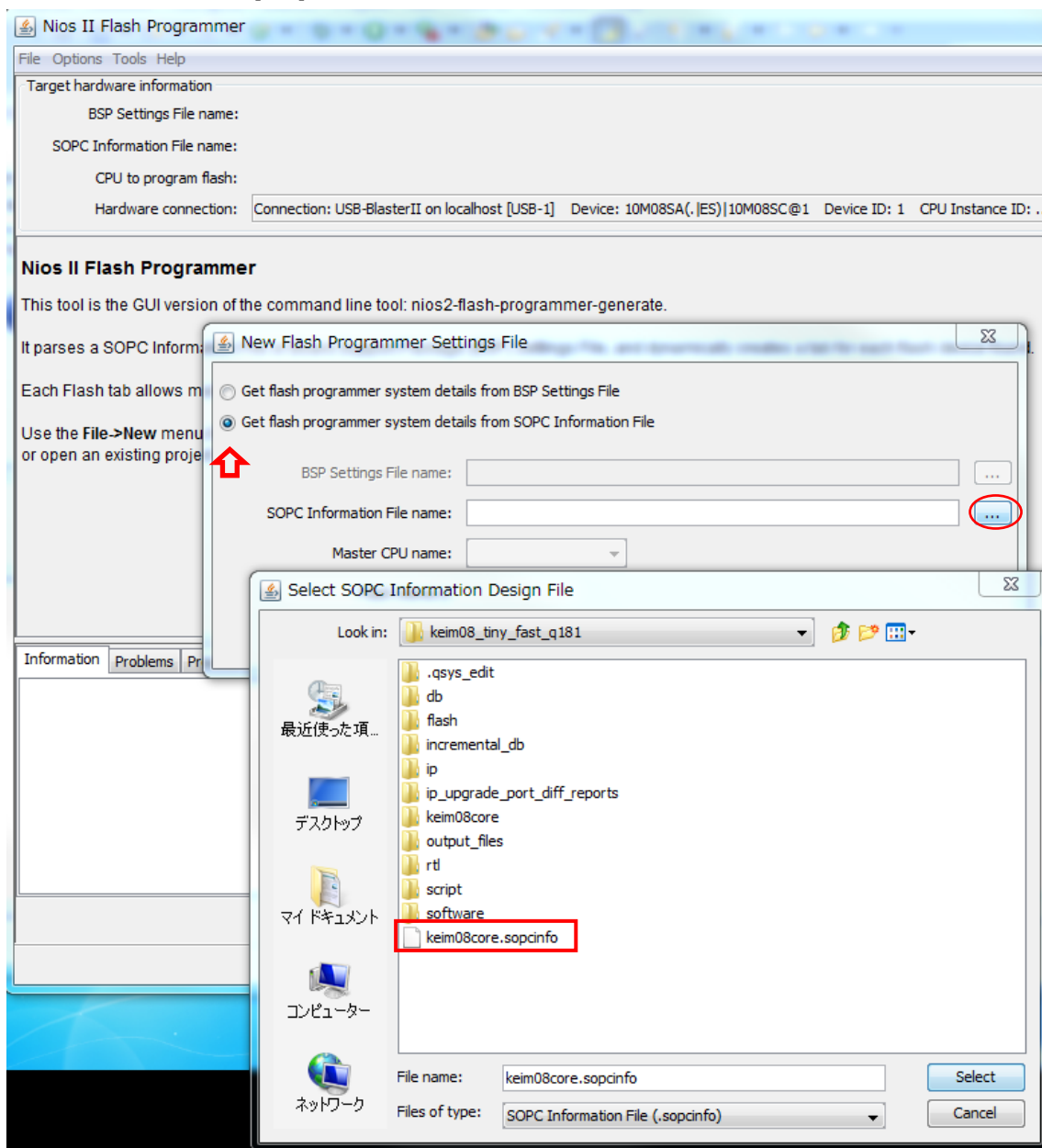


Flash Programmer メイン画面

② New Flash Programmer Settings

メニューより File > New...を選択すると New Flash Programmer Settings File 画面が現れます。

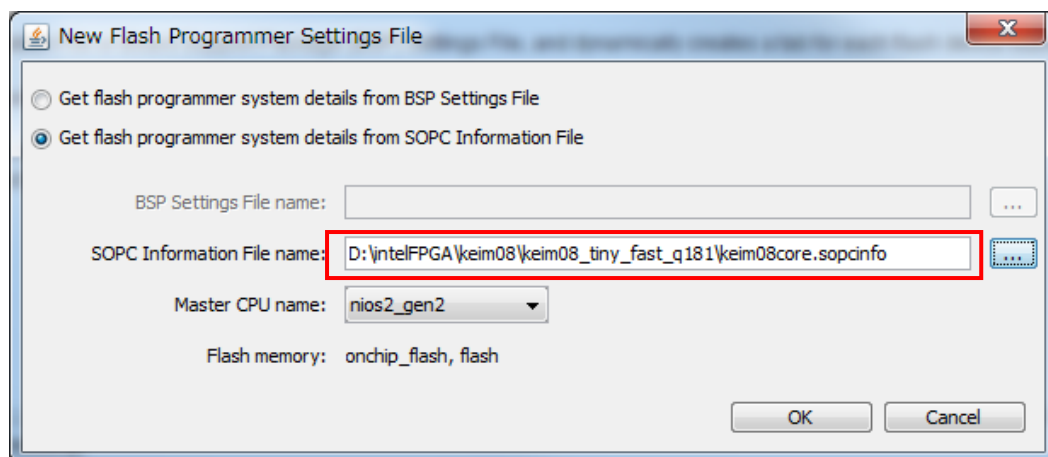
下図に従い赤枠部分を設定し[OK] を選択します。



SOPC Information File name : に

D:\intel\FPGA\keim08\keim08_tiny_fast_q181\keim08core.sopcinfo

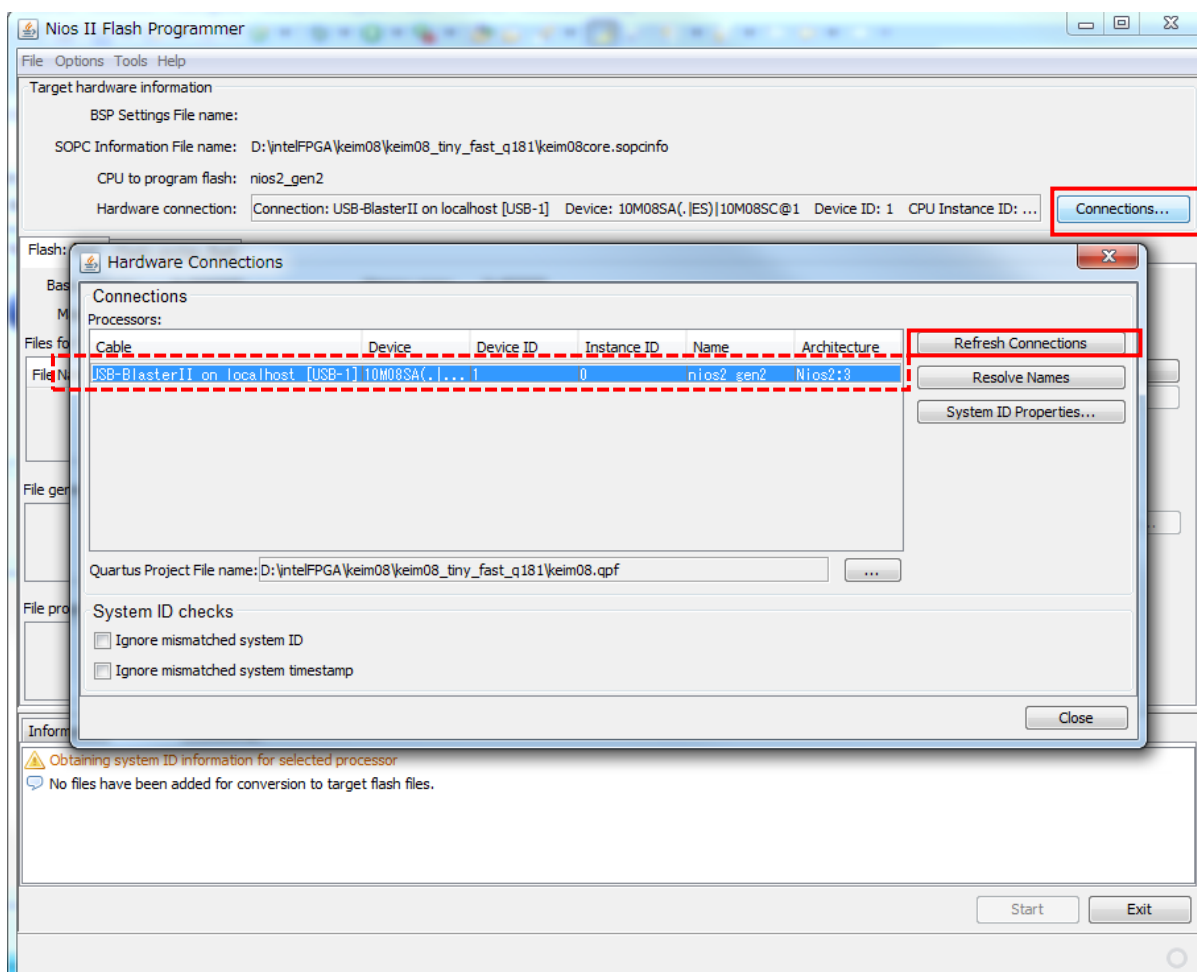
を選択します。



New Flash Programmer Settings 画面

③ Hardware Connections

Flash Programmer メイン画面で Connections... を選択すると Hardware Connections 画面が表示されます。Hardware Connections 画面で Refresh Connections を選択し Processors 欄 に下記表示がされていることを確認し [Close] を選択してください。



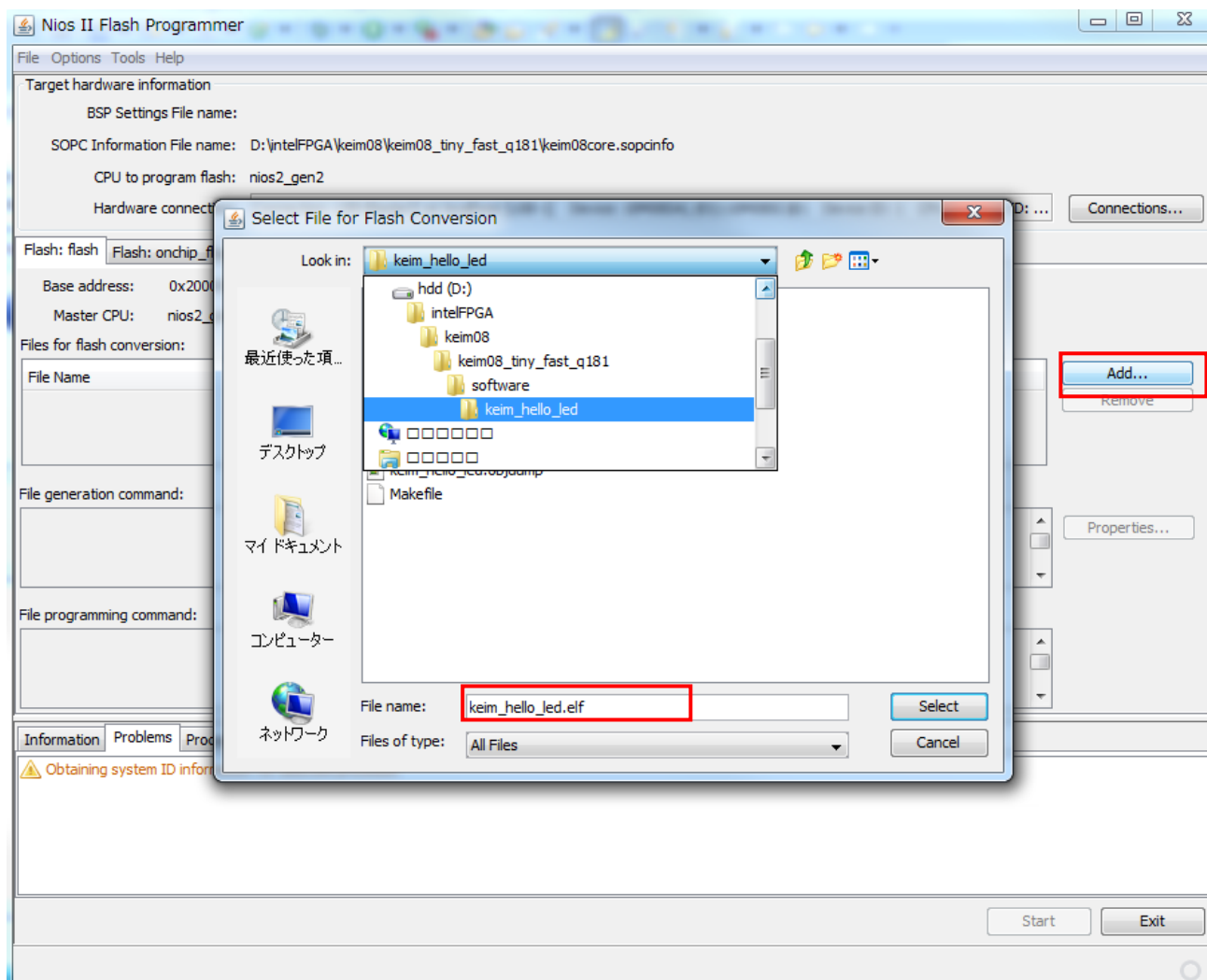
Hardware Connections 画面

④ Flash ROM 書き込みファイルの設定

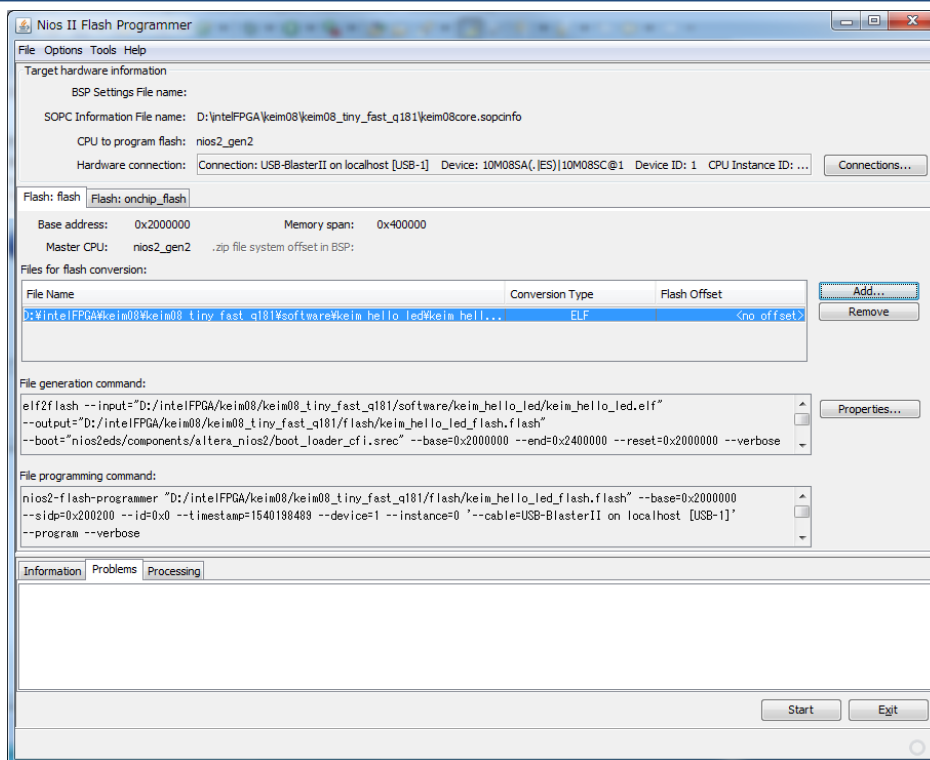
[Add...] を選択し、Flash ROM へ書き込むためのファイル(.elf)を選択してください。

本例では、下記フォルダより “keim_hello_led.elf”を選択します。

D:\intelFPGA\keim08\keim08_tiny_fast_q181\software\keim_hello_led\keim_hello_led.elf



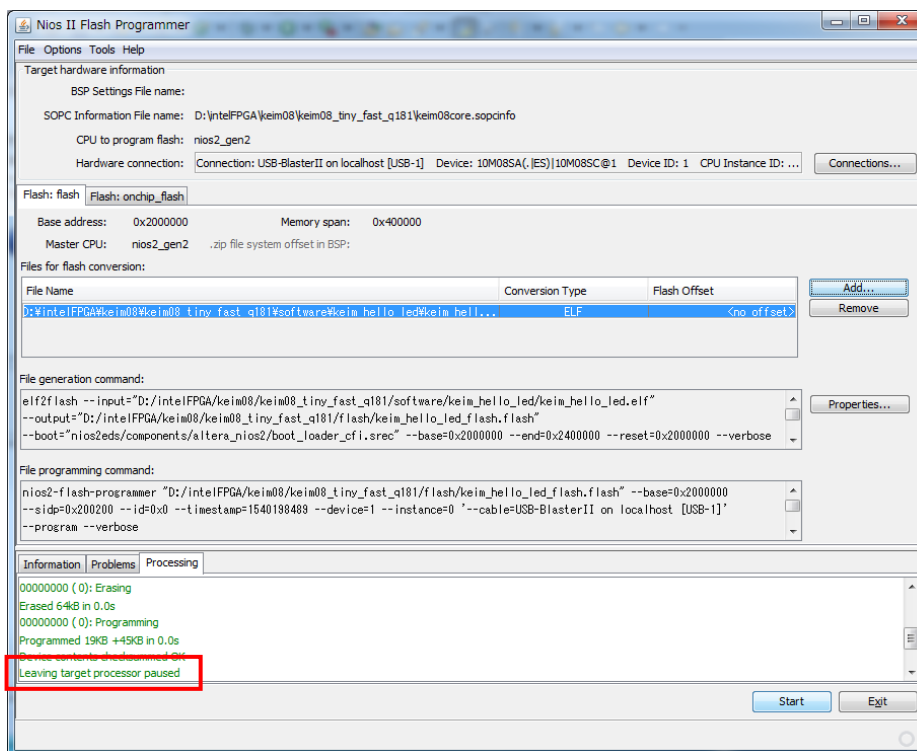
Flash ROM 書き込みファイルの選択



書き込みファイル設定完了

⑤ Flash ROM 書き込みの実行

[Start] を選択すると、Flash ROM への書き込みが実行され、Processing ウィンドウに "Leaving target processor paused" が表示されれば書き込み完了です。



書き込み完了画面

6. 補足資料

6.1. USB シリアルドライバのインストール

KEIm SoM 開発キットの USB 通信は、USB シリアル IC FT232RL (FTDI) を介して行われます。これを使用するためには FTDI 社製の Virtual COM Port ドライバ(以降 VCP ドライバと称す) のインストールが必要です。以降でそのインストール方法及び VCP ドライバの設定方法を記載します。

6.1.1. VCP ドライバのインストール

既にインストールしたことのある PC であれば、ベースボードと開発用 PC を USB mini B ケーブルで接続すると USB シリアルドライバのインストールが始まります。もし自動的にインストールされない場合は、FTDI 社サイトの下記 URL より Windows 用の VCP ドライバをインストールしてください。

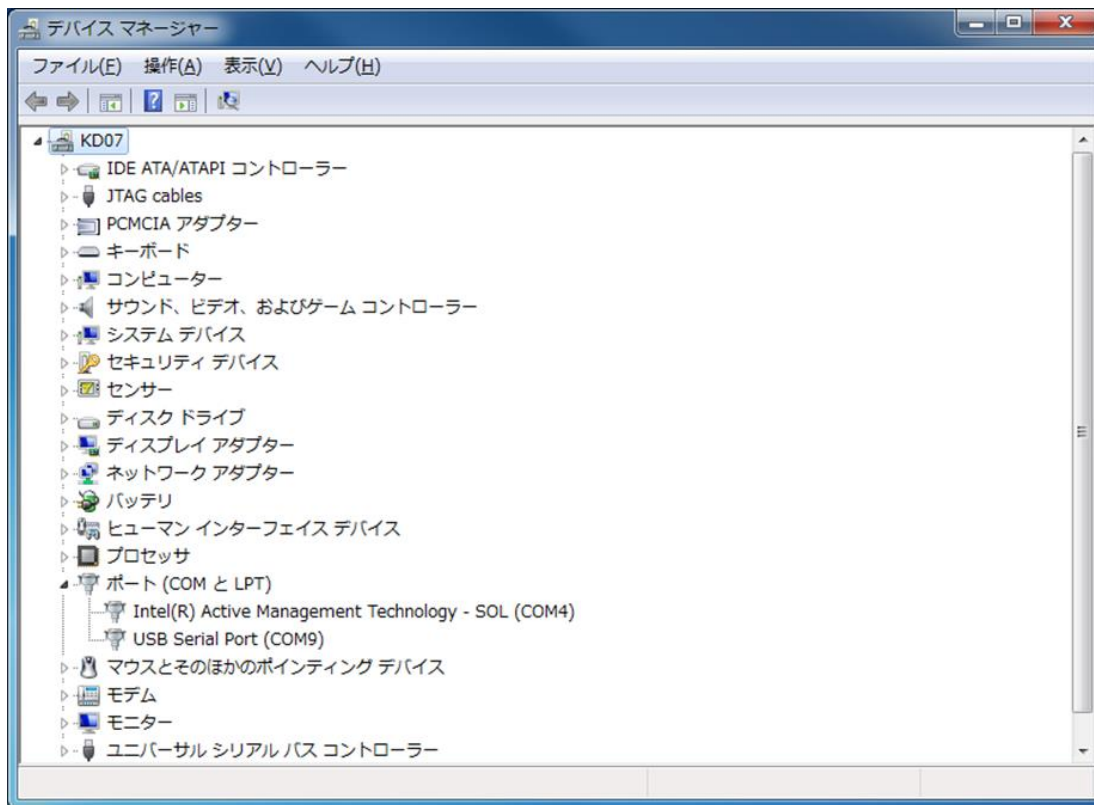
<https://www.ftdichip.com/Drivers/VCP.htm>

(2018 年 9 月現在 2.12.28 WHQL Certified が Windows 用の最新版ドライバです。)

6.1.2. VCP ドライバの設定

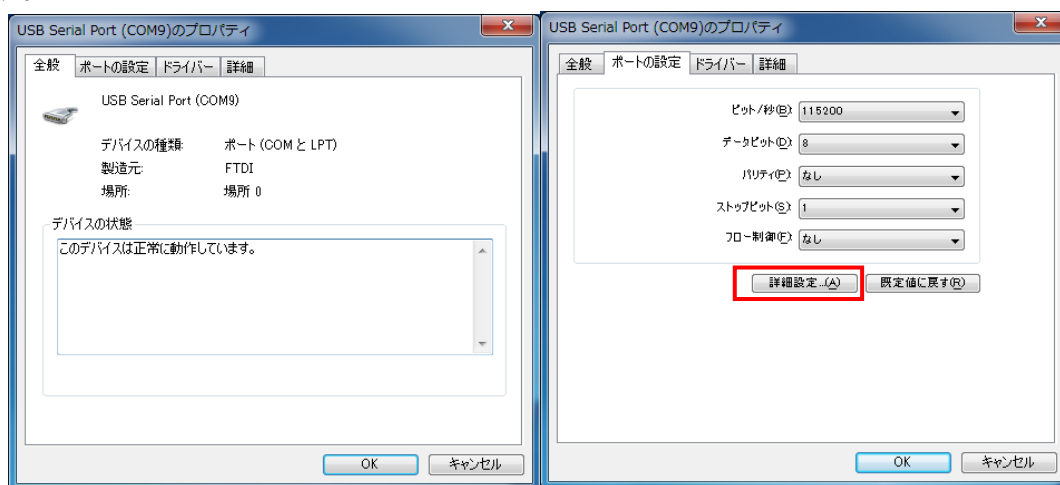
VCP ドライバのデフォルトの設定は、データの待ち時間が長めに設定されており、このチュートリアル の用途に 対してレスポンスが悪いため、VCP ドライバの設定を行います。

- ① Windows のデバイスマネージャーを立ち上げます。



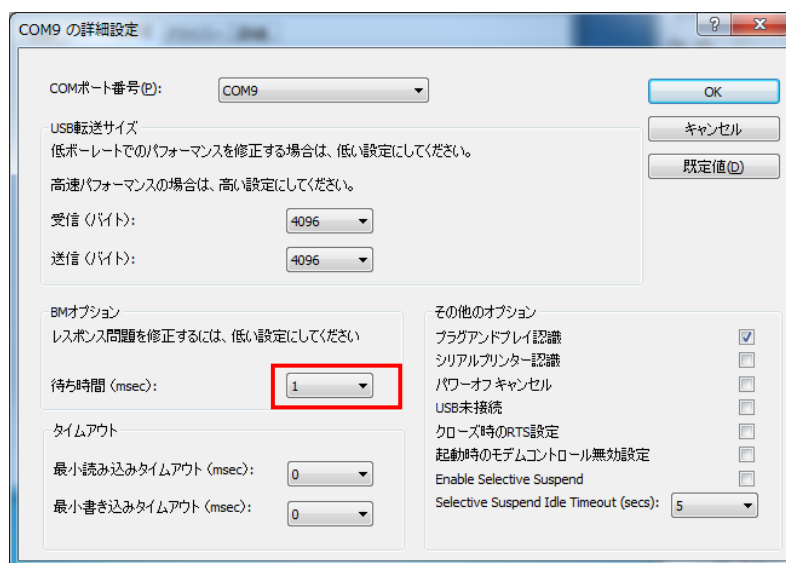
デバイスマネージャー画面

- ② 設定を行う USB Serial Port (COM**)を選択しプロパティを表示し、ポート設定タブ>詳細設定を選択します。



USB Serial Port (COM**) のプロパティ画面

- ③ 詳細設定画面の BM オプションをデフォルトの 16→1 に変更します。



COM**の詳細設定画面

- ④ [OK] を選択し設定画面を閉じます。これで VCP ドライバの設定完了ですので、プロパティ画面も[OK] を選択し設定画面を閉じます。デバイスマネージャーも終了します。この時、場合によっては Windows の再起動を要求されますので、その場合は再起動を実行します。

- ⑤ VCP ドライバの設定を反映するために一度ベースボードに接続している USB ケーブルを抜き、再度差し込みます。これで、VCP ドライバの設定が USB シリアル IC に反映されます。

7. 更新履歴

Ver.	更新日付	内容
1.0.0	2018/10/22	新規作成