

Ver.1.0





#### 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いただき、使用上の注意を守って正しく ご使用ください。



- 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
1.1. 開発環境
2. 開発準備
2.1. Quartus Prime のインストール
2.2. KEIm08 用 FPGA サンプルデザインを開発用 PC にコピー
3. プログラム作成
3.1. Hello World 表示プログラム
3.1.1. Nios II SBT の起動
3.1.2. Nios II Application と BSP を Template より生成
3.1.3. sopcinfo ファイルのコピー
3.1.4. BSP の設定
3.1.5. ソフトウェアプロジェクトのビルド1
3.1.6. FPGA 内部回路イメージの download14
3.1.7. ソフトウェアプロジェクトの実行
3.2. LED 点滅プログラム
3.2.1. 新規 Applecation プロジェクト作成とBSP の設定24
3.2.2. hello_led.c をコピー
3.2.3. プロジェクトのビルド
3.2.4. プログラムの実行
4. FPGA デザインの書き込み
4.1. Quartus Prime Programmer での書き込み
5. プログラムの ROM 化及び Flash ROM 書き込み
6. 補足資料
6.1. USB シリアルのドライバインストール
6.1.1. VCP ドライバのインストール
6.1.2. VCPドライバの設定
7. 更新履歴

## 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<sup>®</sup> FPGA 開発ツール Quartus<sup>®</sup> Prime<sup>®</sup> 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<sup>®</sup> 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
		OS: Windows7 64bit
2	開発用 PC	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 <sup>™</sup>	USB-Blaster II <sup>™</sup> 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<sup>®</sup> II/f "Fast" core を使用しています。 Nios<sup>®</sup> 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<sup>®</sup> 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<sup>®</sup> 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<sup>®</sup> 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<sup>®</sup> II/f "Fast" core の場合

D:¥intelFPGA¥keim08¥keim08\_tiny\_fast\_q181.zip

解凍後のフォルダ

D:¥intelFPGA¥keim08¥keim08\_tiny\_fast\_q181¥

Nios<sup>®</sup> II/e "Economy" core の場合

D:¥intelFPGA¥keim08¥ keim08\_tiny\_economy\_q181.zip

解凍後のフォルダ

D:¥intelFPGA¥keim08¥ keim08\_tiny\_economy\_q181¥

Nios<sup>®</sup> 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=2.61131300.928 125466.1538442790-1762824604.1533265000&elg\_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? g a=2.176649189.1554389885.1538442905-1762824604.1533265000&elg\_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 シリアルのコンソール出力 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り ます。

Workspace L	auncher	manue. Trens. To	×
Select a works	space		
Eclipse stores Choose a wor	your projects in a folder called a workspace. rkspace folder to use for this session.		
Workspace:	D:¥intelFPGA¥keim08¥workspace_keim08	- Browse	
Use this as	s the default and do not ask again		
		ОК	Cancel

#### Workspace の指定



Elle Edit Navigate Search Project Run Nigs II Window Help            •          •          •	Nios II
Image:	Vailable.
Project Explorer X C C C C C C C C C C C C C C C C C C	□ □ つ マ vailable.
E 🔄 🐨 🗸 An outline is not a	🌚 ▽ vailable.
Problems S Tasks Console Properties	▼ □ Pati
0 items selected	

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。

Target hardware information	oard support package based on a software example template D:¥intelFPGA¥keim08¥keim08_tiny_fast_q181¥keim08co
CPU name:	nios2_gen2
Application project Project name: keim_hello_ Use default location Project location: D:¥inte	vorld elFPGA¥keim08_tiny_fast_q181¥software¥keim_hel
Project template Templates Board Diagnostics Count Binary	Template description <ul> <li>Hello World prints 'Hello from Nios II' to STDOUT.</li> </ul>
Float2 GCC	RTOS and requires an STDOUT device in your system's hardware.
Float2 Performance Hello Freestanding Hello MicroC/OS-II Hello World	For details, click Finish to create the project and refer to the readme.txt file in the project directory.



#### 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。

🖨 Nios II - Eclipse	
File Edit Navigate Search Project	Run Nios II Window Help
	Graphic Programmer     Ctrl+7       Quick Access     Image: Ctrl+8
Project Explorer 🛛 📃 🗖	BSP Editor Ctrl+9
E 🔄 💱 ⊽	System Console Ctrl+0 An outling is not available
Reim_hello_world_bsn [keim08con]	
keim_neilo_wond_bsp [keimoscon	
	🖹 Problems 🖉 Tasks 📮 Console 🛛 🔲 Properties 🕹 🗘 😭 🔛 👪 🗮 🗮 🗮 🗮 🗮 🖬 🖛 💭 🖛 🗖 🔻 🗖
	CDT Build Console [keim_hello_world]
	make clean  fkeim hello world clean completel
	16:50:04 Duild Einished (back 1: 200ms)
	10.30.04 DUILU FIRISHED (COOK IS.300MS)
4 111	۲ ۲
0 items selected	



BSP Editor	
File Edit Tools Help	
	n Chick Fachle File Connection Transform Directory
SOPC Information file: CPU name: Operating system: BSP target directory:	Version:
	Select BSP Settings File
Information Problems Processing Initializing BSP components Finished initializing BSP components to Searching for BSP components with components of the searching for BSP components with components wit	Look In:       Image: Second Se
0 items selected	Nios II BSP Editor

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 Vertual 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。



BSP Editor - D:¥intelFPGA¥keim08	¥keim08_tiny_fast_q181¥softv	vare¥keim_hello_world	_bsp¥settings.bsp				
File Edit Tools Help							
Main Software Packages Drivers Linker So	cript Enable File Generation Targe	t BSP Directory					
Linker Section Mappings							
Linker Section Name	Linker Region Name		Nemory Device Name		Add		
.bss	sdram		soram		Remove		
.entry	reset		f ash		Restore Default		
.exceptions	onchip_memory		orchip_memory		·		
.heap	sdram		scram				
.rodata	sdram		scram				
.rwdata	sdram		scram				
.stack	onchip memory		orchip_memory				
.text	onchip_flash		scram				
	onchip_memory_BEFORE_	EXCEPTION					
Linker Memory Regions	onchip_memory				-		
Linker Memory Regions	sdram				Add		
Linker Region Name	Addresereset		Size (bytes)	Offset (bytes)	Auu		
flash	0 totash	1.000	4194272	32	Remove		
reset	0x02000000 - 0x0200001F	flash	32	0	Restore Default		
sdram	0×01000000 - 0×017FFFFF	sdram	8388608	0			
onchip_memory	0×00100020 - 0×00103FFF	onchip_memory	16352	32	Add Memory Devi		
onchip_memory_BEFORE_EXCEPTION	0×00100000 - 0×0010001F	onchip_memory	32	0	Denne Menne De		
onchip flash	0×00000000 - 0×00007FFF	onchip flash	32768	0	Remove Memory De		
					Memory Usage		
					Memory Map.		
					J		
Graved out entries are automatically crea	ted at generate time. They are no	nt editable or persisted in	the BSP settings file				
			are ber betange me.				
Information Problems Processing							
Mapped module: "nios2 gen2" to use the open and the op	default driver version.						
Mapped module: "pio_led" to use the defai	ult driver version.						
Mapped module: "timer_100ms" to use the	default driver version						
Mapped module: "itag uart 0" to use the	default driver version						
Mapped module: "Jug_dart_o" to use the d	efault driver version						
Mapped module: sysid_gsys to use the defe	etaut unver version.						
Mapped module: Uart_0 to use the defau	lit driver version.						
Finished loading drivers from ensemble rep	iort.						
UD Loading BSP settings from settings file.							
Changed manned section " stack" from me	mory region "sdram" to memory regio	n "onchin memory".					

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。

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。

🖨 Nios II - Eclipse				
File Edit Navigate Search Project	Run Nios II Window Help			
Close	n Project e <b>Project</b>	<b>∦ •   2 • 1 0 • 1 0 • 1</b> → •	Qui	ck Access 📔 😰 🔟 Nios II
Project Explorer  Build Build Build Build Build Build Clear Build C/C+	I All Ctrl+B I Project I Working Set n I Automatically ++ Index			E Outline ☆ □ □
Ргор	erties			
	Problems @ Tasks 📃 C CDT Build Console [keim_hell	Console 🛛 🔲 Properties o_world]		
	make clean [keim_hello_world clean 16:50:04 Build Finished	complete] (took 1s.306ms)		
• • •	•			Þ
😂 keim_hello_world				



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  $\, \sigma \,$  Hardware Setup

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

California Programmer Standard Edition - [Chalifornia]		
File Edit View Processing Tools Window Help		Search altera.com 🚺
		]
Line Hardware Setup USB-Blasterii [USB-1]	Mode: JTAG    Progress:	J
Enable real-time ISP to allow background programming when available		
▶ <sup>™</sup> Start → Hardware Setup	Bit CLAMP	
Hardware Settings JTAG Settings		
Relect a programming hardware setup to use when programming devices. This programming		
X Delete hardware setup applies only to the current programmer window.		
Madd File         Currently selected hardware:         USB-Blasteril [USB-1]		
Change File		
Bave File Hardware Server Port Add Hardware		
Add Device USB-BlasterII Local USB-1 Remove Hardware		
1 <sup>%</sup> up		
1 <sup>th</sup> Down		
Type ID Messa		
ti go ∢	4	
System Processing		

Auto Detect > 10M08SC を選択して[OK]

Quartus Prime Programmer Standard Edition - [Chain1.cdf]		
Pice Edit View Processing Tools window Help		Search altera.com
Hardware Setup USB-Blasteril [USB-1]     Enable real-time ISP to allow background programming when avail	Mode: JTAG    Progress:	
File Device	Checksum Usercode Program/ Verify Blank- Configure Check Check Bit CLAMP	
Auto Detect	Select Device	
X Delete	Found devices with shared JTAG ID for device 1. Please select your device.	
Change File	0 10M085A 0 10M085AE5	
Save File	@ 10M08SC	
Add Device		
↓ <sup>™</sup> Down		
x 7 All O A A < <pre>Siter&gt;&gt;</pre>		
Type ID Message	ОК	
8 8 9		
System Processing		

"keim08\_top.sof"の設定

D:¥intelFPGA¥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]

Quartus Prime Pro	ogrammer Standard Er	dition - [Chain1.cdf	1*														
File Edit View F	Processing Tools W	/indow Help													Search altera.co	om	
-																	
, 									_	_			_				-1
📥 Hardware Setup	USB-BlasterII [USB-1]	1						M	1ode: JTAC	G		•	Progr	ess:			
Enable real-time IS	SP to allow background p	programming when a	vailable														
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP						
Stop	<none></none>	0M08SC	00000000	<none></none>													
Auto Detect			Select Ne	2w Programmi	ing File								x				
× Delete			Look in:	D:\intelF	PGA\keim08\	keim08_ď	jev_fast_q1	181\output_fi	iles 🔻 🕻	3 0	0						
Add File			My C	omputer	Name	*		Size	Туре	Date f	Modified						
Change File			ueshi	ima-h	keim08_top.p	of		314 K	(B pof File	2018/	/10/19 13:	.36:42					
Save File					keim08_top.sc	of		848 K	(B sof File	2018/	/10/19 13:	36:42					^
	Concerns of	N000071															Ξ
Add Device	TDI																
T <sup>m</sup> Up																	
Down	10005																-
×																	
		ter>>															
Type ID ID	Message		4														
a													_				
Br ss			File name:	keim08_top.	.sof						l	Open				_	4
System Proc	essing		Files of type	: Programmin	ng Files (*.sof *	.pof *.jam	i *.jbc *.ekp	*.jic)			•	Cancel					

#### ② Download を実行

Program/Configure にチェックを入れて Start

🔌 Quartus Prim	e Programmer Standard Ed	ition - [Chain1.cdf]*												×
<u>File Edit V</u> ie	w P <u>r</u> ocessing <u>T</u> ools <u>W</u> ir	ndow <u>H</u> elp											Search altera.com	6
-	•													
🚖 Hardware Se	🛔 Hardware Setup USB-Blasteril [USB-1] Mode: JTAG 🔻 Progress:													
Enable real-t	me ISP to allow background pr	ogramming when avai	lable											
Start	File	Device	Checksum	Usercode	Program/	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP			
<sup>™</sup> Stop	D:/intelFPGA/keim08/k.	. 10M08SCU169	005DE34C	005DE34C										
💏 Auto Detec														
X Delete														
Add File														*
隆 Change File														
Save File	TDI													-
Add Device														-
1 <sup>4</sup> Up		1169												
J <sup>™</sup> Down		109												-
× All 🙆	🛕 🛕 🔽 < <filte< td=""><td>1&gt;&gt;</td><td></td><td>68</td><td>Eind</td><td>Find Ne<u>x</u>t</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></filte<>	1>>		68	Eind	Find Ne <u>x</u> t								
Д = Туре	ID Message													
<ul> <li>&gt; ge es</li> </ul>														4
System	Processing													
							_		_					



## 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

File Edit Navigate Search Project R	un Nios II Window Help		
File Edit Navigate Search Project     Project Explorer     Project Explorer     E     E     Keim_hello_world     E     Keim_hello_world_bsp [keim08cor]     Image: Comparison of the second	un       Nios II       Window       Help         Run       Ctrl+F11         Debug       F11         Run History       +         Run As       +         Run Configurations       Debug History       +         Debug History       >       Debug Configurations       Debug Configurations         Debug Configurations       External Tools       +         External Tools       COT Build Console [keim_hello_world]       make clean	** ** ** **         **	Quick Access    Puick Access
✓★ ∞ keim_hello_world	<pre>make clean [keim_hello_world clean complete] 16:50:04 Build Finished (took 1s.306 </pre>	ms) :	E v



"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。

<ul> <li>Debug Configurations</li> <li>Create, manage, and run configura</li> <li>[Target Connection]: System time</li> </ul>	tions estamp i	mismatch - conr	nected: "	'0", expected: "1540189039".
Image: Second system       Image: Second system         Image: Secon	Name Project Project Project Project	keim08_samp oject	les et Conne keim_h D:\intelf ile system ne:	ection 🎋 Debugger 🖏 Source 🔲 Common ello_world FPGAlkeim08\keim08_tiny_fast_q181\software\keim_hello_world\keim_hello_world.elf IELF file
Filter matched 8 of 8 items				Re <u>v</u> ert Apply
?				Debug Close



次に"Target Configration"のタブを開き "System IC 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。

Debug Configurations										
Create, manage, and run configurations										
The expected Stdout device name does not match the selected target byte stream device name.										
	]									
	Name: keim08_samples									
type filter text	📄 Project 📕 Target Connection 🛛 🏂 Debugger 🧤 Source 🔲 Common									
C/C++ Application	Connections									
C/C++ Attach to Application	Processors:									
C/C++ Postmortem Debugger	Cable	Device	Device ID	Instance D	Name	Architecture	Refresh Connections			
C/C++ Remote Application	JSB-BlasterII on local	host [USB-1] 10M08SA(.	. 1	0	nios2 O	Nios2:3	Resolve Names			
Launch Group							System ID Properties			
<ul> <li>Nios II Hardware</li> </ul>	Byte Stream Devices:									
New_configuration	Cable	Device	Device ID	Instance ID	Name	Version				
Nios II Hardware v2 (beta)	JSB-BlasterII on local	host [USB-1] 10M08SA(.	. 1	0	jtaguart O	1				
	Disable 'Nice II Console'	view								
		*IC**								
	Quartus Project File name: <	Using default .sopcinfo & .jdi f	iles extracted fro	m ELF >						
	System ID checks									
	gnore mismatched syst	tem ID								
	gnore mismatched syst	em timestamp								
	Download									
	Download ELF to select	ted target system								
	Start processor									
	Reset the selected targ	et system								
						Revert	Apply			
Filter matched 8 of 8 items										
?						<u>D</u> ebug	Close			



Processors: / Bye Stream Devices: の項目 (\_\_\_\_\_\_)がリフレッシュ(一旦消えて再度表示)されましたら [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。

Debug Configurations	
Create, manage, and run configurations The expected Stdout device name does no	ot match the selected target byte stream device name.
Image: Second	Name:       keim08_samples         Project       Target Connection         Connections       Processors:         Cable       Device       Device       Device         Refresh Connections       Processors:       Refresh Connections         Resolve       Device       Device       Name       Architecture         Refresh Connections       Refresh Connections       Refresh Connections         Resolve Names       System D Properties       Resolve Names         Byte Stream Devices:       Cable       Device       Device         Cable       Device       Device       Varian         Byte Stream Devices:       Cable       Device       Device         Cable       Device       Device       Varian         Byte Stream Devices:       Cable       Device       Varian         Cable Tios II Console' view       Outrus Project File name:       Using default.sopcinfo & jdi files extracted from ELF >         System ID Checks       Varian       Varian       Varian         Ø upore mismatched system D       Varian       Varian         Ø upore mismatched system ID       Varian       Varian         Ø upore mismatched system       Varian       Varian         Ø upore mi
Filter matched 8 of 8 items	Revert Apply
0	Debug Close

#### 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。

Confirm Perspective Switch		×
This kind of launch is co	onfigured to open the Nios II Debug perspective when it suspends.	
This perspective is designed view which is particular	gned to support efficient development of a Nios II project. It turns off automatic build and adds the Nios II Co Iy useful.	insole
Do you want to open th	is perspective now?	
Remember my decision	Yes No	



② Breakpoint を設定して実行

Niosll Debug が起動しましたら、 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。





#### KEIm-08SoM ヘッダーボードスタートアップガイド

Nios II Debug - keim_hello_world/hello_world.c - Eclipse	2							
Eile Edit Source Refactor Navigate Search Project Bun Nigs II Window Help								
1 → □ ◎   ◎   物 + 0 → 9 →   ▶ □ ■ № № ◎ 点示 元 ◎ @ Ø → [ / かけれ + 7 → + + + + + + + + + + + + + + + + +								
<b>①</b>		Quick Acces	s 📑 😰 Nios II 🕸 Nios II Debug					
to Debug 🕅	💃 🖉   i+ 🔻 🗖 🗖	🗱 🗣 Variables 🕱 💁 Breakpoints 👯 Reg	jisters 🗻 Memory 📃 🗖					
🔺 📩 keim08_samples [Nios II Hardware]			‱ 🕫 🖂 💕 🗶 🔆 📑 🗠 🔻					
Altera CDI GDB Debugger (2018/10/22 18:24) (S	uspended)	Name	Value					
<ul> <li>Thread [1] (Suspended: Breakpoint hit.)</li> <li>1 main() hello, world c:21 0x01000058</li> </ul>								
<pre>terminated, exit value: 0&gt;nios2-download (201</pre>	8/10/22 18:24)							
nios2-download (2018/10/22 18:24)								
📲 nios2-elf-gdb (2018/10/22 18:24)			<u>^</u>					
<u></u>		•	•					
i hello_world.c ⊠	🗆 🛗 Nios II Console 🛛	x 🛛 🗐 📕 🔽 🗖	🗄 Outline 🛿 🗖 🗖					
2⊕ * "Hello World" example	keim08_samples - cable:	USB-BlasterII on localhost [USB-1] device ID: 1 in	🐉 🖻 🛱 🖉 🖋 🖷 🛧 🗠					
17 #include <stdio.h></stdio.h>			stdio.h					
18 19⊖ int main()			<ul> <li>main() : int</li> </ul>					
20 {								
22								
24 }								
25								
	r							
× +								
E Console 🔀 🔎 Tasks		■ × ※   B, bii 12/   1	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;					
keim08_samples [Nios II Hardware] nios2-elf-gdb (2018/10 _start () at HAL/Src/crt0.5:2/2	/22 18:24)							
272 movui r2, NIOS2_DCACHE_SIZE			Â					
Temporary breakpoint 1, main () at hello_world.c:	21		E					
21 princi ( nerio from wios iii (n );			-					
4 			Þ					

その後、 🍉 ボタンを押してソフトウェアを実行します。

Niosll 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。



#### KElm-08SoM ヘッダーボードスタートアップガイド

Nios II Debug - keim_hello_world/hello_world.c - Eclipse								
Eile Edit Source Refactor Navigate Search Project Run Nigs II Window Help								
: 🖆 ▾ 🔚 🗞   🖓 = 🌘 ▼ 🎧 ▼ 🕪 🗉 📾 🖉 🎘 🧐 📖 🗮 🛒 (2) 🥵 🖉 ▼ 🤃 🖓 ▼ 🔛 🖉 ▼ 🖓 ▼ 🖓 ▼ 🖓 ▼ 🔶 ▼								
<b>企</b>	Quick Access 🛛 😰 🛛 🔞 Nios II 🗱 Nios II Debug							
🎋 Debug 🛛 🦌 🥳 🖬 🚺 🔻 🖓	Image: Comparison of the second se							
<ul> <li>keim08_samples [Nios II Hardware]</li> <li>Altera CDI GDB Debugger (2018/10/22 18:24) (Suspended)</li> <li>Phread [1] (Suspended: Breakpoint hit.)</li> <li>1 main() hello_world.c:23 0x01000064</li> <li><terminated, 0="" exit="" value:="">nios2-download (2018/10/22 18:24)</terminated,></li> <li>nios2-download (2018/10/22 18:24)</li> <li>nios2-elf-gdb (2018/10/22 18:24)</li> </ul>	Name Value							
<pre>     hello_world.c      world.c      w</pre>	nsole 3 - cable: USB-BlasterII on localhost [USB-1] device ID: 1 in os II!							
📮 Console 🛿 🧔 Tasks	■ X ¾   № 2 №   3 U   EE							
keim08_samples [Nios II Hardware] nios2-elf-gdb (2018/10/22 18:24)								
<pre>21 printf("Hello from Nios II!\n"); Breakpoint 2, main () at hello_world.c:23 23 return 0;</pre>	* *							
(* [	• • •							



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 Applecation 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 Applecation 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⊐ピーe
- 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 Applecation 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 Applecation 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。

⊜ N	ios II - keim_hello_world/hello_world.	c - Eclipse						
File	Edit Source Refactor Navigate	Search Project	Run	Nios II Window Help				
	New	Alt+Shift+N ►	<b>≣</b>	Nios II Application and BSP from Template	1			
	Open File		C#	Nios II Application		ess	R	🚺 Nios II 🕸 Nios II Debug
	Close	Ctrl+W	C+i	Nios 11 Board Support Package			-	
	Close All	Ctrl+Shift+W	R	Nios II Library				🗄 Outline 🛛 👘 🗆
	Save	Ctrd - C		Project			*	🐉 🖻 🚏 🖉 🖋 🛛 🗰
	Save As	Cul+5		Other Ctrl+	N			$\bigtriangledown$
	Save All	Ctrl+Shift+S	Γ		_			stdio.h
	Revert	ourronnero		a from Nics III) o").				main() : int
	Maria		F11	0 170m (103 11: (n );				
-2	Move	52	E					
	Rendme	F2	ι.					
5	Convert Line Delimiters To	15	ι.					
			ι.					
9	Print	Ctrl+P						
	Switch Workspace	+	ι.					
	Restart							
2	Import							
2	Export		ι.					
	Properties	Alt+Enter	L					
	1 hello_world.c [keim_hello_world]							
	Exit		Ŀ			4	*	
_		📳 Problems 🧔 T	Tasks	🕒 Console 🙁 🔚 Nios II Console 🔲 Properties				
					£			
		CDT Build Console	[kein	n_hello_world]	~			
[BSP build comple				]				*
[Keim_neiio_worid_pniid_compiete]								
4		18:23:56 Build	Finis	shed (took 3s.255ms)				

Project name :	をここでは	keim_hello_led	とします。
----------------	-------	----------------	-------

Nios II Application		
BSP location is empty		
roject name keim_hello_led		
ISP location:		
		Create
☑ Use default location		
Location:		
dditional arguments:		
ommand:		
Use relative path		
(?)	Einish	Cancel

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。

Project Selection		
Select a BSP project		
Ekeim_hello_world_bsp		
1		
		Correct
	UK	Cancel

下記設定になりましたら [Finish] します。

😑 Nios II Appli	cation	
Nios II Applic	ation	
Create a new	Nios II Software Build Tools application project	
Project name	: keim_hello_led	
BSP location:	D:¥intelFPGA¥keim08¥keim08_tiny_fast_q181¥software¥keim_hello_world_bsp	····
		Create
🔽 Use default	location	
Location:	D:¥intelFPGA¥keim08¥keim08_tiny_fast_q181¥software¥keim_hello_led	
Additional arg	uments:	
		* *
Command:		
nios2-app-ge	nerate-makefileapp-dirbsp-dir/keim_hello_world_bspelf-name keim_hello_led.elfno-src	A 7
🔽 Use relativ	e path	
(?)	Einish	Cancel
J		

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:\fintelFPGA\formation keim08\_tiny\_fast\_q181\formation formation keim08\_tiny\_fast\_q181\formation formation keim08 for a formation keim08 formation keim08 formation keim08 formation keim08 formation keim08

elp		
Quick Access	🖹 🔝 Nios II	🕸 Nios II Debug
E	🖹 🗖 📴 Outline	x - 0
		, 🔊 🔊 e 💥
keim08_tiny_fast_q181 > software	<ul> <li>examples</li> </ul>	👻 🐓 exampi
ルダー		
ń	更新日時	種類
keim08_hello_led	2018/10/22 15:01	ファイル フォル…
keim08_hello_world	2018/10/22 15:01	ファイル フォル
keim08_hello_world_bsp	2018/10/22 15:01	ファイル フォル…
hello_led.c	2018/09/27 13:16	Cファイル
	p   Quick Access     Quick Access      (y) <	p Quick Access Quick Access Quick Access P Outline D Outline P Outline Outline P Outline Outline Outline Outline Outline Outline Outli

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

File Operation	x
Select how files should be imported into the project:	
© Copy files	
© <u>L</u> ink to files	
✓ Create link locations <u>r</u> elative to: PROJECT_LOC →	
Configure Drag and Drop Settings	
OK   Cancel	

# 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

Nios II - keim_hello_world/hello_world	.c - Eclipse	
File Edit Source Refactor Navigate	Search Project Run Nios II Window Help	
	ⓒ ▼ 🖏 Open Project 🕴 🖢 ▼ 🎘 ▼ 🏷 ♥	⇒ <b>*</b>
	Close Project Qui	ck Access 📑 🔯 Nios II 🕸 Nios II Debug
Project Explorer 🕅 🗖	Build All Ctrl+B	C C Cutline 🕅 🗆 🗆
	20 Helle Build Project	
↓ 🗠 keim hello led	16 Build Working Set	
▷ 🗊 Includes	17 #. Clean 18	stdio.h
▶ 🔝 hello_led.c	190 i Build Automatically	• main() : int
🚡 Makefile	21 C/C++ Index ►	
keim_hello_world	Properties	
keim_heiio_world_bsp [keim08cord]	24 }	
	23	
	4	<u>▼</u> 4
	Ducklama , 🔿 Taska 🖸 Caracle 🕅 👘 Nice II Canaela 📼 Duccartica	
	Problems Val Tasks 📮 Console 🐹 🕅 Nios II Console 📋 Properties	
	CDT Ruild Cancele [keim helle led]	🔶 🕆 🔄 🖬 🚮 🗧 🐘 🛛 🖬 🕇 🗸
	18:33:18 Build Finished (took 1s.414ms)	
		-
(	( •	4
😂 keim_hello_led		

#### 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。

Programmer - D:	/intelFPGA/keim08/keim08_tiny_fast_q	181/keim08 - keim08_t	top - [keim08_1	top.cdf]	_				• X
<u>F</u> ile <u>E</u> dit <u>V</u> iew	P <u>r</u> ocessing <u>T</u> ools <u>W</u> indow <u>H</u> elp						Se	arch altera.co	om 🤇
Aardware Setup.	USB-Blasterii [USB-1]		Mode: JTAG		•	Progress:	1	00% (Succes	sful)
🔲 Enable real-time I	SP to allow background programming whe	n available							
▶ <sup>™</sup> Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit
Stop	output_files/keim08_top.sof	10M08SCU169	003F815B	003F815B	<b>v</b>				
💏 Auto Detect									
X Delete									
💾 Add File									
Change File			m						
Save File									
Add Device									

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

Debug Configurations	
Create, manage, and run configuration Nios II Hardware Tab Group	ins to the second se
Image: Second system         Image: Second system <th>Name: keim08_samples  Project I Target Connection % Debugger Common % Source  Project name: keim_helio_led  Project ELF file name: keim_helio_led.elf  Enable browse for file system ELF file  File system ELF file name:  Advanced</th>	Name: keim08_samples  Project I Target Connection % Debugger Common % Source  Project name: keim_helio_led  Project ELF file name: keim_helio_led.elf  Enable browse for file system ELF file  File system ELF file name:  Advanced
Filter matched 8 of 8 items	Revert Apply
?	Debug Close

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

Debug Configurations							×
Create, manage, and run configurations The expected Stdout device name does n	ot match the selected targ	et byte stream device na	ame.				Ť.
Image: The second s	Name: keim08_sample:	a onnection %> Debugge Device nost (USB-1) 10M0834(.  . Device nost (USB-1) 10M0834(.  . View Using default .sopcinfo &.jdi em D em timestamp ed target system at system	Perice ID Device ID Device ID	Instance D mstance D 0 mELF >	Name nics2 cen2 Name jtes uart .	Architecture Nios219 Version	Refresh Connections Resolve Names System D Properties
Filter matched 8 of 8 items						Revert	Apply
0						Debug	Close

Refresh が完了後、[Apply] > [Debug] を実行

#### > Confirm Perspective Switch [Yes] を実行



#### ⑤ Debug が起動しましたら、

36 usleep(wait\_time);の行に breakpoint を設定

Nios II Debug - keim_hello_led/hello_led.c - Eclipse		
<u>File E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un Ni <u>o</u> s II <u>V</u>	/indow <u>H</u> elp	
🔁 🕶 🔚 🕞 💩 💠 🕶 💽 🕶 💁 🕶 💷 📾 🖄 💀 🕫 📰	😕 🗀 🖋 🕶 🚺 💝 🐓 🖛 😓	▼ ⇒ ▼
<b>☆</b>	Quick Ac	cess 📔 😰 Nios II 🕸 Nios II Debug
🗱 Debug 🛛 🧏 🙀 i i i 🗸 🖓	🗆 (x)= Variables 💥 💁 Breakpoints 🛅	Registers 🔋 Memory 📃 🗖
a 📩 keim08_samples [Nios II Hardware]		🗄 📲 🖻 🥤 🗶 🔆 🗖 🖄
a 🚱 Altera CDI GDB Debugger (2018/10/22 18:48) (Suspended)	Name	Value
Thread [1] (Suspended: Breakpoint hit.)	(x)= wait_time	16789056
1 main() hello_led.c:25 0x01000058	(x)= i	16789056
<terminated, 0="" exit="" value:="">nios2-download (2018/10/22 18:48)</terminated,>	(×)= led_data	-559038737
nios2-download (2018/10/22 18:48)		
📓 nios2-elf-gdb (2018/10/22 18:48)		÷
	•	۶. ۲
🖻 hello_led.c 🛛 🗈 👘 Nios II Cor	nsole 🛛 🔳 💽 🗎 🗸 🖓 🗖 E	🗆 📴 Outline 🕱 👘 🗖
<pre>20 Int 1; 27 volatile int led_data = 0; 28 29 // printf("Hello from Nios II!\n"); 30 while (1){ 31 while (1){ 32 for (i = 0; i &lt;= 3; i++ ) { 33 led_data = i &amp; 0x3; 34 IOWR_32DIRECT(PIO_LED_BASE, 0x0, le 35 // printf("led = %x\n", led_data); 36 usleep(wait_time); 37 } 37 } 39 return 0; 4</pre>	- cable: USB-Blasterii on localhost [USB-1] device ID: 1 i	n P □ I <sup>4</sup> <sub>2</sub> ≷ ≷ ● ₩ ▼ U unistd.h System.h I alt_types.h io.h main() : int
🔁 Console 🔀 🧔 Tasks	■ × 🔆   🗟 🚮 🕬	📰 🖳 🕞 😁 🚽 🖳 🗸 📄 👘
<pre>keim08_samples [Nios II Hardware] nios2-elf-gdb (2018/10/22 18:48) _start () at HAL/SrC/Crt0.5:2/2 272 movui r2, NIOS2_DCACHE_SIZE Temporary breakpoint 1, main () at hello_led.c:25 25 int wait_time = 500000; </pre>		* = *
Writable Smart Insert	2	

# 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



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

Nios II Debug - keim_hello_led/hello_led.c - Eclipse				x
<u>File Edit Source Refactor Navigate Search Project</u>	<u>R</u> un Ni <u>o</u> s II <u>W</u> indow	<u>H</u> elp		
📑 🕶 🖶 💼 🔯 🕶 💽 🕶 🎴 💌 💷 🛤 🗈	- 0- 1.e. 🗟 🔊 🙆 🖨	s 🔗 🕶 📝 💝 🐓 🖛 🖓 👻 🦕 🕶	⇒ -	
		Quick Acces	s 👔 👔 🔯 Nios II 🕸 Nios II De	bug
🎋 Debug 🔀	;∦{ i→ ▽□□	(x)= Variables 🔀 💁 Breakpoints 👫 Reg	isters 📋 Memory 🗖	
🔺 📩 keim08_samples [Nios II Hardware]			🏝 📲 🖃 💕 🗶 🔆 🖆 🖆	$\bigtriangledown$
4 🥁 Altera CDI GDB Debugger (2018/10/22 18:48)				
Thread [1] (Running)	(4.0.(0.0.4.0.)			
<pre>cterminated, exit value: U&gt;nios2-download (2018)</pre>	/10/22 18:48)			
nios2-elf-adb (2018/10/22 18:48)				
		4		Ψ. 
	Nios II Console 2		E Outline ⊠	
<pre>27 volatile int led_data = 0;</pre>	keimuo_sampies - cable:	USB-blasterii on localnost [USB-1] device ID: 1 In		~
<pre>29 // printf("Hello from Nios II!\n");</pre>			system.h	
30 31 while (1){			alt_types.h	
32 for $(i = 0; i <= 3; i++)$ {			🖬 io.h	
I I I I I I I I I I I I I I I I I I I			main() : int	
35 // printf("led = %x\n", led_data);				
37 }				
38 } 39 return 0; ▼				
( ) ) ( )				
📮 Console 🕱 🧔 Tasks		🔲 🗙 🔆 🖹 🖬 🔛 🖕	] U (FF) 🚽 🗉 🕶 🗖 🗸 🖻	
keim08_samples [Nios II Hardware] nios2-elf-gdb (2018/10/2	2 18:48)			
<pre>36 usleep(wait_time);</pre>				*
Breakpoint 2, main () at hello_led.c:36				
36 usleep(wait_time);				
4				Þ.
Writable	Smart Insert	3		
		1.4		

■ 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<sup>®</sup> MAX<sup>®</sup> 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いますので、 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 ヤ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。

👋 Programmer - D	/intelFPGA/keim08/keim08_tiny_fast_q	181/keim08 - keim08_	top - [keim08_	top.cdf]*					
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Processing Tools Window Help						Se	earch altera.co	om 🌖
🔔 Hardware Setup	USB-Blasterii [USB-1]		Mode: JTAG		•	Progress:	1	00% (Succes	sful)
Enable real-time	ISP to allow background programming wher	n available							
	Eile	Device	Checksum	Usercode	Program /	Verify	Blanks	Evamine	Security
► <sup>™</sup> Start	File	Device	Checksum	osercode	Configure	veniy	Check	examine	Bit
■ <sup>™</sup> Stop	output_files/keim08_top.pof	10M08SCU169	02984886	003F815B					
💏 Auto Detect	CFM0					<b>V</b>			
× Delete	UFM								
Add File						_			
Change File	•					_			
Save File									
Add Device									
J <sup>™</sup> Down	10M085CU169								
	TDO								
	•								

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。

Nios II - keim_hello_led/hello_led.c - E	clipse
File Edit Source Refactor Navigate	Search Project Run Nios II Window Help
📑 👻 🔚 🕞 📠 🔂 🕶 🗳 🕶 🗳 🕶	🞯 🔻 🗱 🔻 🚺 👻 🖌 🐨 🐨 🐨 🐨 🐨 🐨 🐨 🐨 🐨 🐨 🐨 🐨 🐨
	Quartus Prime Programmer Ctrl+8 Juick Access
	BSP Editor Ctrl+9
Project Explorer 🛛 📃 🗖	ic hello_led.c ☆ System Console Ctrl+0
E 🔄 🗊 🗸	2⊕ * "Hello World" example.∐ 16
▶ ₩ Binaries	17 //#include <stdio.h></stdio.h>
⊳ 🔊 Includes	19 <b>#include</b> "system.h"
> 🕞 obi	20 #include <alt_types.h> 21 #include <io.h></io.h></alt_types.h>
⊳ 🛃 hello_led.c	22
keim_hello_led.elf - [alteranios]	23⊖ int main() 24 {
😰 keim_hello_led.map	25 int wait_time = 500000;
📓 keim_hello_led.objdump	26 int i ; 27 volatile int led data = 0:
🗋 Makefile	28
keim_hello_world	<pre>29 // printf("Hello from Nios II!\n"); 30</pre>
keim_hello_world_bsp [keim08core	31 while (1){
	32 for (i = 0; i <= 3; i++) { 33 led data = i & 0x3:
	<pre>34 IOWR_32DIRECT(PIO_LED_BASE, 0x0, led_data);</pre>
	<pre>35 // printf("led = %x\n", led_data); 36 usleen(wait time);</pre>
	37 }
	38 } 39 return 0:
	40 }

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。

🔬 Nios II Flash Programmer	- • ×
File Options Tools Help	
Target hardware information	
BSP Settings File name:	
SOPC Information File name:	
CPU to program flash:	
Hardware connection: Connection: USB-BlasterII on localhost [USB-1] Device: 10M08SA(. [ES) 10M08SC@1 Device ID: 1 CPU Instance ID:	Connections
Nios II Flash Programmer	
This tool is the GUI version of the command line tool: nios2-flash-programmer-generate.	
It parses a SOPC Information File or Board Support Package (BSP) Settings File, and dynamically creates a tab for each flash device found.	
Each Flash tab allows multiple files to be added to a list for eventual conversion to flash and programming to its target flash device.	
Use the File->New menu item from the top level menu bar to create a new flash settings project, or open an existing project using the File->Open menu item.	
Information Problems Processing	
Start	E <u>x</u> it
	0



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。

실 Nios II Flash Programmer			Contra California		
File Options Tools Help					
Target hardware information					
BSP Settings File name:					
SOPC Information File name:					
CPU to program flash:					
Hardware connection:	Connection: USB-Blas	sterII on localhos	t [USB-1] Device: 10M08SA(.  ES) 10	0M08SC@1 Device ID: 1 C	CPU Instance ID: .
Nios II Flash Programme This tool is the GUI version of t	er the command line to	ool: nios2-flash	-programmer-generate.		
It parses a SOPC Inform	New Flash Progran	nmer Settings	File	makes a fait for each fact	23
Each Elach tab allows m	Cet flack programmer	ovetem detaile fra	m RSD Sattings File		
Each Flash tab allows III	Get flash programmer	system details fro	om SOPC Information File		
Use the File->New menu or open an existing proje	BSP Settings	File name:			
	CODC Information	Eile annas			
	SOPC Information	riie name;			
	Master C	CPU name:	<b>T</b>		
	Select SOPC	Information [	Design File		
	Look in:	📗 keim08_tir	ny_fast_q181	- 🤣 🔊 🛄 -	
Information Problems Pr	最近使った項 デスクトップ マイドキュメント コンピューター	.qsys_edit db flash incrementa ip upgrade keim08core output_file rtl script software keim08core	il_db e_port_diff_reports e s e.sopcinfo		
	く ネットワーク	File name: Files of type:	keim08core.sopcinfo SOPC Information File (.sopcinfo)		Select Cancel

SOPC Information File name: に

を選択します。

New Flash Programmer Settings File				
◯ Get flash programmer system details from BSP Settings File				
Get flash programmer system deta	ilis from SOPC Information File			
BSP Settings File name:				
SOPC Information File name:	D: \intelFPGA \keim08 \keim08 \tem08_tiny_fast_q181 \keim08core.sopcinfo			
Master CPU name:	nios2_gen2 🔻			
Flash memory:	onchip_flash, flash			
	OK Car	icel		

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 Processrors 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。

🛓 Nios	s II Flash Programmer	3
File Opt	otions Tools Help	
Target	t hardware information	
	BSP Settings File name:	
SOF	DPC Information File name: D:\intelFPGA\keim08\keim08\tiny_fast_q181\keim08core.sopcinfo	
	CPU to program flash: nios2_gen2	
	Hardware connection: Connection: USB-BlasterII on localhost [USB-1] Device: 10M08SA(.]ES) 10M08SC@1 Device ID: 1 CPU Instance ID: Connections	
Flash:		
Bac	Ardware Connections	
M	Connections	
Files fo	Processors:	
File N	ISR-Blaster II on localbost [ISR-1] 10M0854(	ן ר
		5 I
	System ID Properties	
File ger		
		ר ו
	Quartus Project File name: D: \intelFPGA \keim08 \keim08 \teim08_tiny_fast_q181 \keim08.qpf	
File pro	System ID checks	
	Ignore mismatched system ID	
	Ignore mismatched system timestamp	
Inform	Close	
🛕 Obta	taining system ID information for selected processor	
Ҏ No f	files have been added for conversion to target flash files.	
<u> </u>		
	Start Exit	
		0
		~

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。

 $D: \texttt{``HintelFPGA`+keim08`+keim08\_tiny\_fast\_q181`+software`+keim\_hello\_led`+keim\_hello\_led.elf$ 

🛓 Nios II Flash Programm	ner	
File Options Tools Help		
Target hardware information		
BSP Settings File nam	ie:	
SOPC Information File nam	e: D:\intelFPGA\keim08\keim08_tiny_fast_q181\keim08core.sopcinfo	
CPU to program flas	h: nios2_gen2	
Hardware connect	Select File for Flash Conversion	Connections
Flash: flash Flash: onchip_f	Look in: 👔 keim_hello_led 🔹 🗸 🍺 🗁 🖽 🗸	
Base address: 0x2000	🕞 hdd (D:)	
Master CPU: nios2_(	lintelFPGA	
Files for flash conversion:	最近使った項 keim08 tiny fast g181	
File Name	↓ software	Add
	keim_hello_led	Remove
File generation command:	Makenie	
	71 142421	Properties
		-
File programming command:		
	コンピューター	
	File name: keim hello led.elf	Select
Information Problems Prod	ネットワーク Files of type: All Files	Cancel
▲ Obtaining system ID inform	All Files	
		Start Exit

Flash ROM 書き込みファイルの選択

File Ontions Too	le Halo					
Target bardware	information					
BSP Set	tings File name:					
SOPC Inform	ation File name:	D:\intelEPGA\keim08\keim09_tinv	fact o 181\keim08core cor	rcinfo		
CDLL to	program flash	pies2 con2	rost_d to t kennopeoi ersol	Juino -		
CPU to	program hash:	Connections LICB Blockett on Local	hant DICR 11 Devices 10		Device ID: 1 CDU Jackson II	
Haruw	are connection:	connection, 030-bidsterii on local	ioscioso-ij Device: Iu	HUOSA(, [ES] [10HUOSCI@1	Device 10. 1 CPU Instance It	
Flash: flash Flas	sh: onchip_flash					
Base address:	0x2000000	Memory span:	0x400000			
Master CPU:	nios2_gen2	.zip file system offset in BSP:				
Files for flash con	version:					
File Name				Conversion Type	Flash Offset	
File generation cor elf2flashin output="D:/i	nmand: put="D:/intelF ptelFPGA/keim0	PGA/keim08/keim08_tiny_fast_ 18/keim08 tiny_fast_n181/flas	q181/software/keim_he b//eim_bello_led_fla	:llo_led/keim_hello_lec	.elf″	
File generation con elf2flashin output="D:/i boot="nios2e	nmand: put="D:/intelF ntelFPGA/keim0 ds/components/	PGA/keim08/keim08_tiny_fast_ 18/keim08_tiny_fast_q181/flas %altera_nios2/boot_loader_cfi	q181/software/keim_he h/keim_hello_led_flas .srec″base=0x2000(	∶llo_led/keim_hello_leo h.flash″ 00end=0x2400000r	.elf″ eset=0x2000000verbose	^ -
File generation cor elf2flash in output="D:/i boot="nios2e File programming o	mmand: put="D:/intelF ntelFPGA/keim0 ds/components/ command:	PGA/keim08/keim08_tiny_fast_ 18/keim08_tiny_fast_q181/flas altera_nios2/boot_loader_cfi	q181/software/keim_h h/keim_hello_led_flas .srec"base=0x20000	: lo_led/keim_hello_lec h.flash" 00end=0x2400000r	.elf″ eset=0×2000000verbose	•
File generation cor elf2flashin output="D;/i boot="nios2e File programming o nios2-flash-pr sidp=0x20020 programve	nmand: put="D:/intelF GA/keim0 ds/components/ command: ogrammer "D:/i 0id=0x0t rbose	PGA/keim08/keim08_tiny_fast_ 8/keim08_tiny_fast_a181/flas %ltera_nios2/boot_loader_cfi ntelFPGA/keim08/keim08_tiny_ imestamp=1540188489device	q181/software/keim_ht h/keim_hello_led_flat .srec"base=0x20000 fast_q181/flash/keim =1instance=0 'ce	illo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB-BlasterII on l	.elf" eset=0x2000000verbose base=0x2000000 ocalhost [USB-1]"	•
File generation con elf2flash in output="D:/i boot="nios2e File programming o nios2-flash-pr sidp=0x20020 program ve Information Pro	nmand: put="D:/intelF ntelFPGA/keim0 ds/components/ command: ogrammer "D:/i 0id=0x0t rbose blems Processin	PGA/keim08/keim08_tiny_fast_ 8/keim08_tiny_fast_a181/flas %ltera_nios2/boot_loader_cfi ntelFPGA/keim08/keim08_tiny_ imestamp=1540188489device	q181/software/keim_ht h/keim_hello_led_flat .srec"base=0x2000( fast_q181/flash/keim =1instance=0 'ce	illo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB-BlasterII on l	.elf" eset=0x2000000verbose base=0x2000000 ocalhost [USB-1]"	•
File generation con el f2f lash in output="0:/i boot="nios2e File programming o nios2-f lash-pr sidp=0x20020 programve Information Pro	nmand: put="D:/intelF ntelFPGA/keim0 ds/components/ command: ogrammer "D:/i 0id=0x0t rbose blems Processin	PGA/keim08/keim08_tiny_fast_ 8/keim08_tiny_fast_a181/flas %altera_nios2/boot_loader_efi ntelFPGA/keim08/keim08_tiny_ imestamp=1540188489device	q181/software/keim_h h/keim_hello_led_flas .srec"base=0x20000 fast_q181/flash/keim =1instance=0'cc	illo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB=BlasterII on l	.elf" eset=0x2000000verbose base=0x2000000 ocalhost [USB-1]'	
File generation con elf2flash in output="Distribute" boot="rilos2e File programming o nios2-flash-pr sidp=0x20020 program ve Information Pro	nmand: put="D:/intelFPGA/keim0 ds/components/ command: ogrammer "D:/i 0id=0x0t rbose processin Processin	PGA/keim08/keim08_tiny_fast_ 18/keim08_tiny_fast_a181/flas %altera_nios2/boot_loader_cfi ntelFPGA/keim08/keim08_tiny_ imestamp=1540188488device 	q181/software/keim_h h/keim_hello_led_flas .srec <sup>°</sup> base=0x2000( fast_q181/flash/keim =1instance=0 'ce	:llo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB=BlasterII on l	.elf" eset=0x2000000verbose base=0x2000000 ocalhost [USB-1]'	
File generation con elf2f1ash in output="0:/i boot="nios2e File programming o nios2-f1ash-pr sidp=0x20020 programve Information Pro	nmand: put="D:/intelF PCA/keim0 ds/components/ command: ogrammer "D:/i 0id=0x0t rbose blems Processin	PGA/keim08/keim08_tiny_fast_ 8/keim08_tiny_fast_q181/flas altera_nios2/bootloader_cfi ntelFPGA/keim08/keim08_tiny_ imestamp=1540188488device	q181/software/keim_hd h/keim_hello_led_flas .srec <sup>°</sup> base=0x2000( fast_q181/flash/keim_ =1instance=0 'ce	:llo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB-BlasterII on 1	.elf" eset=0x2000000verbose base=0x2000000 ocalhost [USB-1]'	
File generation con elf2f1ash in output="0:/i boot="nios2e File programming o nios2-f1ash-pr sidp=0x20020 programve Information Pro	nmand: put="0:/intelF intelFPGA/keim0 ds/components/ command: ogrammer "D:/i 0id=0x0t rbose blems Processin	PGA/keim08/keim08_tiny_fast_ 8/keim08_tiny_fast_q181/flas altera_nios2/boot_loader_cfi ntelFPGA/keim08/keim08_tiny_ imestamp=1540198489device	q181/software/keim_hr h/keim_hello_led_flat .srec"base=0x20000 fast_q181/flash/keim, -linstance=0 'ce	ilo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB-BlasterII on l	.elf" eset=0x2000000verbose base=0x2000000 ocalhost [USB-1]"	
File generation con elf2f1ash inoutput="0:/i boot="nios2e File programming o nios2-f1ash-pr sidp=0x20020 programve Information Pro	mmand: put=""D:/intelF intelFPGA/keim0 de/components/ command: corrammer "D:/i0 0id=0x0t rbose blems Processin	PGA/keim08/keim08_tiny_fast_ 8/keim08_tiny_fast_0181/flas altera_nics2/boot_loader_cfi ntelFPGA/keim08/keim08_tiny_ imestamp=1540188489device	q181/software/keim_ht h/keim_hello_led_flat .srec"base=0x20000 fast_q181/flash/keim_ =1instance=0 'ce	illo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB-BlasterII on l	.elf" eset=0x2000000verbose base=0x2000000 ocelhost [USB-1]"	
File generation cor elf2flash in output="D:/i boot="nios2e File programming o nios2-flash-or programve Information Pro	mmand: put="D:/intelF irommand: command: ogrammer "D:/i 0id=0x0t rbose blems Processin	PGA/keim08/keim08_tiny_fast_ 8/keim08_tiny_fast_q181/flas %ltera_nios2/boot_loader_cfi ntelFPGA/keim08/keim08_tiny_ imestamp=1540198489device	q181/software/keim_ht h/keim_hello_led_flat .srec"base=0x20000 fast_q181/flash/keim =1instance=0 'ce	illo_led/keim_hello_lec h.flash" 00end=0x2400000r hello_led_flash.flash" ble=USB-BlasterII on l	.elf" eset=0x2000000verbose base=0x2000000 ocalhost [USB-1]"	A C

#### 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 processer pused"が表示されれば書き込み完了です。

Nios II Flash Programmer	2 * 0 * 0 * <b>0</b> *	BU 11 03.	1000		
File Options Tools Help					
Target hardware information					
BSP Settings File name:					
SOPC Information File name:	D:\intelFPGA\keim08\keim08_tiny	_fast_q181\keim08core.sop	tinfo		
CPU to program flash:	nios2_gen2				
Hardware connection:	Connection: USB-BlasterII on loca	alhost [USB-1] Device: 10M	108SA(. ES) 10M08SC@1 E	evice ID: 1 CPU Instance ID:	Connections
Flash: flash Flash: onchip_flash					
Base address: 0x2000000	Memory span:	0x400000			
Master CPU: nios2_gen2	.zip file system offset in BSP:				
Files for flash conversion:					
File Name			Conversion Type	Flash Offset	Add
D:¥intelFPGA¥keim08¥keim08	tiny fast q181¥software¥keim	hello led¥keim hell	ELF	<pre><no offset=""></no></pre>	Remove
output="D:/intelFPGA/keim boot="nios2eds/components, Ele scoscomping compand:	08/keim08_tiny_fast_q181/fla /altera_nios2/boot_loader_cf	sh/keim_hello_led_flash i.srec"base=0x20000(	n.flash″ 10end=0x2400000re:	set=0×2000000verbose 💂	
nico?-flack-programmand.	into IEDCé /kaie00 /kaie00 tinu	foot a191/floot/kain b	alla lad flaab flaab".	hass-0-2000000	1
sidp=0x200200id=0x0i	timestamp=1540198489devic	e=1instance=0 'cat	)le=USB-BlasterII on lo	calhost [USB-1]'	
programverbose				*	
Information Problems Processin	p				
00000000 ( 0): Erasing					
Erased 64kB in 0.0s					
00000000 (0): Programming Programmed 19KB ±45KB in 0.0c					ſ
Bevice contents chedroammed OK	-				
Leaving target processor paused					
				Star	t E <u>x</u> it

#### 書き込み完了画面

## 6. 補足資料

6.1. USB シリアルのドライバインストール

KEIm SoM 開発キットの USB 通信は、USB シリアル IC FT232RQ (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 シリアルの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 (COM9)のプロパティ	USB Serial Port (COM9)のプロパティ
USB Serial Port (COM9)のプロパティ     本       全般     ホートの設定       ドライバー       USB Serial Port (COM8)       デバイスの種類       ポート (COM と LPT)       製造元       FTDI       場所:       場所:       場所:       ごのデバイスは正常に動作しています。	USB Serial Port (COM9)のプロパティ 全般 ホートの設定 ドライバー 詳細 ビット/秒(型) [15200 ・ データビット(型) 個 パリティ(型) なし フロー制御(型) なし 製造設定 (ム) 歴史値に厚す(四)
OK ##2/2/L	ок <b>*</b> туди

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。

COM9 の詳細設定	ि <mark>र</mark>
COMボート番号(P):     COM9       USB転送サイズ     低ボーレートでのパフォーマンスを修正する場合は、低い設定にして       高速パフォーマンスの場合は、高い設定にしてください。     受信 (パイト):       受信 (パイト):     4096 ・       送信 (パイト):     4096 ・	) OK キャンセル 気定値の
BMオブション レスポンス問題を修正するには、低い設定にしてください 待ち時間 (msec):	その他のオブション フラヴアンドプレイ記録 シリアルプリンター記録 パワーオフ キャンセル USB未接続 クローズ時のRTS設定 起動時のモデムコントロール無効設定 Enable Selective Suspend Selective Suspend Idle Timeout (secs): 5 ▼

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			

