KEIm SoM 開発キット チュートリアルガイド

Ver.1.1.0





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

E	次	
1.	概要	4
	1.1. 開発環境	4
2.	開発準備	5
	2.1. Quartus のインストール	5
	2.2. USB シリアルのドライバインストール	5
	2.2.1. VCP ドライバのインストール	5
	2.2.2. VCPドライバの設定	6
3.	プログラム作成	8
	3.1. Hello World 表示プログラム	8
	3.1.1. Nios II SBT の起動	8
	3.1.2. BSP の作成	10
	3.1.3. sopcinfo ファイルのコピー	10
	3.1.4. BSP の設定	.11
	3.1.5. SDK ソースファイルのインポート	13
	3.1.6. プログラム作成	15
	3.1.7. プロジェクトのビルド	16
	3.1.8. プログラムの実行	17
	3.1.9. プログラムのデバッグについて	19
	3.2. LED 点滅プログラム	20
	3.2.1. ソースファイル	20
	3.2.2. LED 点滅プログラムの実行	23
4.	プログラムの ROM 化及び Flash ROM 書き込み	24
5.	サンプルアプリケーションについて	27
6.	更新履歴	28



1. 概要

本書は KEIm SoM 開発キットを使用してソフトウェアを開発する際の簡易チュートリアルガイドです。開発は Altera[®] FPGA 開発ツール Quartus[®] II Design Software(以降 Quartus) 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。

Hello World 表示プログラム作成及び動作確認
 (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	ベースボード	KEIm-08SoM 搭載済み
2	開発用 PC	OS: Windows7 64bit 開発ツール *2 : Quartus II Design Software v15.0 以降及び Nios II Embedded Design Suite v15.0 以降 ターミナルソフト: TeraTerm
3	Altera 社 USB-Blaster [™]	別売り *1
4	AC アダプタ	KEIm SoM 開発キットに同梱
5	USB mini B ケーブル	KEIm SoM 開発キットに同梱
6	KEIm SDK	弊社 Web サイト(<u>http://www.kd-group.co.jp/info/</u>)よりダウンロード

*1 本製品には USB-Blaster 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 II Design Software v15.0.0.145 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 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、Altera 社の FPGA 開発ツール Quartus 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 Altera 社のホームページをご参照ください。

http://www.altera.co.jp/products/software/sfw-index.jsp

Nios II EDS は Quartus 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. 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。

2.2.1. 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。

http://www.ftdichip.com/Drivers/VCP.htm

(2015 年 8 月現在 2.12.06 WHQL Certified が Windows 用の最新版ドライバです。)



2.2.2. 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。

当 デバイスマネージャー	x
ファイル(E) 操作(A) 表示(Y) ヘルプ(出)	
4 🛁 HD05	
▶ ● 1394 バスホスト コントローラー	
▶·⑧ Bluetooth 無線	
▶ 🤮 DVD/CD-ROM ドライブ	
P 😋 IDE ATA/ATAPI コントローラー	
🛛 🖓 🔮 Jungo	
▶·· USB 仮想化	
▶ = キーボード	
イスマネージャー ■ <td< td=""></td<>	
▶ 📲 サウンド、ビデオ、およびゲーム コントローラー	
▶ 長 システム デバイス	
▶ 👝 ディスク ドライブ	
▶ 📲 ディスプレイ アダプター	
▶ 🔮 ネットワーク アダプター	
▶ 幅 ヒューマン インターフェイス デバイス	
▶ □ プロセッサ	
▲ 管 ポート (COM と LPT)	
 ⇒ デノイスマネーシャー ファイル(E) 操作(A) 表示(M) ヘルプ(E) ● ● (□) ● (A) ● (A)	
▶	
▶ ● ユニバーサル シリアル バス コントローラー	
▶ ◆ 記憶域コントローラー	

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
全般 ポートの設定 ドライバー 詳細	全般 ボートの設定 ドライバー 詳細
USB Serial Port (COM9) デバイスの種類 ポート (COM と LPT) 製造元: FTDI 場所: USB Serial Converter デバイスの状態 このデバイスは正常に動作しています。	ビット/秒(B) 3600 ・ データビット(D) 8 ・ パリティ(P) なし ・ フロー制御(P) なし ・ 野細設定 (人) 原定値に戻す(R)
ОК ++у/2/	

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	the second se	? ×
COMボート番号(P): USB転送サイズ 低ポ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 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): 4096 ▼ 送信 (パイト): 4096 ▼		OK キャンセル 既定値(0)
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 ブラヴアンドブレイ認識 シリアルグリンター認識 パワーオフキャンセル USB未接続	
タイム/ワト 最小読み込みタイムアウト (msec): 0 	クロース時の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

④OK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。

⑤VCP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。

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 表示プログラム作成

②LED 点滅プログラム

3.1. Hello World 表示プログラム

開発環境のセットアップで行った UART 接続(USB 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。主な作業手順としては下記の通りとなります。

①BSP の作成

②SDK ソースファイルのインポート

③プログラムの作成

④プロジェクトのビルド

⑤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

Altera 15.0.0.145 > Nios II EDS 15.0.0.145 > Nios II 15.0 Software Build Tools for Eclipse 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 C:¥KEIm_SDK¥workspace 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				
elect a work	space				
Eclipse stores	s your projects in a folder called a workspa	ice.			
Choose a wor	rkspace folder to use for this session.				
Workspace:	C:¥KEIm_SDK¥workspace		Browse		
-					
<u>Use this as</u>	s the default and do not ask again				
				ОК	Cancel

Workspace の指定



lios II - Eclipse					
<u>Edit Navigate Search Project Nios II</u>	<u>Run Window Help</u>				
• • • • • • • • • • • • • • •	0 • 9 • 10 # • 11 • 11 • 10			Quick Access	1 1 1 1 N
Project Explorer II			- 0	₿ Outline Is An outline is not ar	ailable.
	ti Problems ≅	e 🗆 Properties	Resource	Path	ş. v
	a and prove		incapor co		
	1				

Nios II SBT メイン画面



3.1.2. BSP の作成

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。ここではまず BSP プロジェクトを作成します。

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。 弊社 Web サイ トより keim08core.sopcinfo ファイルをダウンロードし、KEIm_SDK フォルダにコピーしてください。

Nios II SBT のメニューから File > New > Nios II Board Support Package を選択し、下記赤枠部分を設定後、 Finish をクリックします。Project name は任意ですがここでは keim_sdk_bsp とします。

Nios II Board Support Package	and the set	
Nios II Board Support Package		
create a new Nios 11 Software Build Tools board support package project		
Project name: keim_sdk_bsp		
SOPC Information File name: C:\KEIm_SDK\keim08core.sopcinfo)
Use default location		
Location: C:¥KEIm_SDK¥software¥keim_sdk_bsp		
CPU: nios2_gen2		
BSP type: Altera HAL		
Additional arguments:		
		* *
Command:		
nios2-bsp hal//keim08core.sopcinfocpu-name nios2_gen2		÷
☑ Use relative path		
(?)	Finish	Cancel

Nios II Board Support Package 設定画面



3.1.4. BSP の設定

標準出力や使用メモリの選択をするために BSP の設定を行います。

BSP プロジェクトを作成すると Nios II SBT の Project Explorer 上に Project name で指定したフォルダが作成 されます。そのフォルダを選択し右クリック > Nios II > BSP Editor...を選択し BSP Editor を立ち上げます。

①メインの設定

Street Lotter Street Lotter and		-
Edit Tools Help		
in Software Packages Drivers Linker Script Enable	ile Generation Target BSP Directory	
SOPC Information file:\\keim08core.sopcinfo CPU name: nios2_gen2 Operating system: Altera HAL BSP target directory: C:\KEIm_SOK\poftware\keim_sdk Settings Common Settings Common Store - stdin - stdout - stdout	Version: default Version: default bsp hal sys_clk_timer: timestamp_timer: stdin: stdin: stdout: stdout: stdout: stderr: uart_0 uart_0 uart_0 uart_0	E
Inker enable_exception_stack exception_stack.size exception_stack_memory_region_ enable_interrupt_stack interrupt_stack_size interrupt_stack_memory_region_n	enable_gprof enable_reduced_device_drivers enable_sim_optimize hal.linker	
4		
formation Problems Processing		
Mapped module: "uart_0" to use the default driver versis Mapped module: "uart_1" to use the default driver version. Mapped module: "wdt" to use the default driver version. Mapped module: "time_0" to use the default driver versis Mapped module: "time_2" to use the default driver versis Mapped module: "time_2" to use the default driver versis	n. 	
Finished loading drivers from ensemble report.		(
Loading BSP settings from settings file.		

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_0、標準入出力は uart_0 とします。(デフォルトの標準出力は uart_1 に なっていますが、本チュートリアルでは USB シリアル側に文字表示をさせるためこれを変更します)

②使用メモリの選択

le Edit Tools Help					
Main Software Packages Drivers Linke	r Script Enable File Generation Targe	t BSP Directory			
Linker Section Mappings					
Linker Section Name	Linker Region Name		Memory Device Name	(Add
.bss	sdram		sdram		Remove
.entry	reset		flash	(Restore Defaults
.exceptions	onchip memory		onchip memory		
.heap	sdram		sdram		
.rodata	sdram		sdram		
.rwdata	sdram		sdram		
stack	onchip_memory		onchip_memory		
.text	sdram		sdram		
Linker Memory Regions					
Linker Region Name	Address Range	Memory Device Name	Size (bytes)	Offset (bytes)	Add
flash	0x02000020 - 0x023FFFFF	flash	4194272	32	Remove
reset	0x02000000 - 0x0200001F	flash	32	0	Restore Defaults
sdram	0x01000000 - 0x017FFFFF	sdram	8388608	0	
onchip_memory	0×00100020 - 0×00103FFF	onchip_memory	16352	32	Add Memory Device
onchip_memory_BEFORE_EXCEPTION	0×00100000 - 0×0010001F	onchip_memory	32	0	Remove Memory Device
onchip flash data	0x00000000 - 0x00007FFF	F onchip flash data 32768 0		remove Hendry Devices	
					Memory Usage
					Memory Map
Grayed out entries are automatically c	reated at generate time. They are n	ot editable or persisted in th	te BSP settings file.		
nformation Problems Processing					
Mapped module: "uart_1" to use the de	fault driver version.				
Mapped module: "wdt" to use the defau	ult driver version.				
Mapped module: "timer 0" to use the d	efault driver version.				
Mapped module: "timer_1" to use the d	efault driver version.				
Mapped module: "timer 2" to use the d	efault driver version				
Mapped module: "flash" to use the defa	ault driver version.				
Finished loading drivers from ensemble	report				
Transieurioduring universition ensemble	reports				
I cading BSD sattings from sattings file					

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。

③BSP の生成

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. SDK ソースファイルのインポート

SDK ソースファイルのインポートには keim_sdk_src_verxx.zip(xx:Version)を使用します。 弊社 Web サイトより keim_sdk_src_verxx.zip ファイルをダウンロードし、KEIm_SDK フォルダにコピーしてください。

Nios II SBT のメニューより File > New > Nios II Application を選択し、下記赤枠部分を設定後、

Finish をクリックします。Project name は任意ですがここでは keim_sdk とします。

又、BSP location は先ほど作成した keim_sdk_bsp を選択します。

Nios II Application	
Nios II Application Create a new Nios II Software Build Tools application project	
Project nime: keim_sdk	
	Create
2010/2010/2010/2010/2010/2010/2010/2010	
Additional arguments:	*
Command: nios2-app-oenerate-makefileapp-dirbsp-dir _/keim sdk bspelf-name keim sdk.elfno-src	
☑ Use relative path	*
(?) Finish Co	incel

Nios II Application 画面

Nios II SBT のメニューより File > Import をクリックし Import を立ち上げます。

General > Archive File を選択し Next をクリックします。

Select				
Import resources from an archive file into an existing	project.			
Select an import source:				
type filter text				
🔺 🗁 General				
👰 Archive File				
A Existing Projects into Workspace				
🗀 File System				
Preferences				
▷ 🗁 C/C++				
🖻 🗁 CVS				
o 👝 Git				
🖻 🗁 İnstall				
» Dios II Software Build Tools Project				
Remote Systems				
D 🗁 RPM				
🔋 📂 Run/Debug				
p 🗁 Tasks				
- Toom				
0	< Dack	Next >	Finish	Cancel

Import 画面(Select)



From archive file: keim_sdk_src_verxx.zip ファイルを指定します。 Into folder: keim_sdk を指定し Finish をクリックします。

rom archive file: C:¥KEIm_S	SDK¥keim_sdk_src_ver1.0.0.z	tip		•	Browse
	(
Filter Types	Select All	Deselect All			
Filter Iypes	Select All	Deselect All			Browse
Filter Types hto folder: keim_sdk Qverwrite existing resource	Select All	Deselect All			Bro <u>w</u> se

Import 画面(ファイル指定)



3.1.6. プログラム作成

keim_sdk の main.c の main 関数内 /* User program */ の下に

printf("Hello World¥n");

を記述し、ファイルをセーブします。

Nios II - keim_sdk/keim_sdk_src/main.c	- Eclipse	
Eile Edit Source Refactor Navigate Se	arch Project Nigs II Bun Window Help	
	• • • • • • • • • • • • • • • • •	Quick Access
Noject Explorer 😂	E S P T D B main.c 2	😑 🗆 😫 Outline 🛙 🔛 🞼 💘 🔶 🗰 🍼 🖻
⊿ 😆 keim_sdk	* KEIm SoM SDK main	▲ 🖬 board_init.h
> 🜮 Includes	Himelude "boond init b"	 main() : int
keim_sdk_src	wincidde board_init:n	 timer0_irq_event(void*, alt_u32) :
API-CORE		 timer1_irq_event(void*, alt_u32) :
DRIVER	<pre></pre>	 timer2_irq_event(void*, alt_u32) :
board_init.c	1	 uart_cmd_event(void*) : void
🕞 🔐 board_init.h		 eint0_irq_event(void*, alt_u32) : v
🕞 🕜 config.h	/* board initial */	 eint1_irq_event(void*, alt_u32) : v
▶ 💽 int.c	board_init();	 eint2_irq_event(void*, alt_u32) : v
∍ 🗃 int.h	/* User program */	 eint3_irq_event(void*, alt_u32) : v
Main.c	<pre>printf("Hello World\n");</pre>	 eint4_irq_event(void*, alt_u32) : v
De Makefile	wature Ar	 eint5_irq_event(void*, alt_u32) : v
keim_sdk_bsp [keim08core]	}	 eint6_irq_event(void*, alt_u32) : v
		 eint7_irq_event(void*, alt_u32) : v
	<pre>*vid time*_irq_vent(void* Context, al[us2 id) { /* IRQ Handler TIMER0 disable */ irq_disable(&TIMER0 band_info[0].irq); /* TIMER0 interrupt class */ TIMER_write(&TIMER_bond_info[0], TIMER_STR, 0); /* Interrupt processing */ intq_enable(&TIMER_mand_info[0].irq); irq_enable(&TIMER_bond_info[0].irq); /* Id(#TIMER_bond_info[0].irq /pre>	
	🛣 Problems 🖓 Tasks 🗳 Console 🕮 📼 Properties	
	CDT Build Console [keim_sdk]	
	11:10:33 **** Clean-only build of configuration Nios II for project keim_s make clean [keim_sdk clean complete] 11:10:37 Build Finished (took 367ms)	sdk ****
	Writable Smart Insart I	7 : 26

main.c 画面

default では UART の標準出力は UART1(RS-232C)になっているため変更します。 keim_sdk の config.h の下記部分を変更してファイルをセーブします。

/* STDIO use UART ch */の下の

#define CONFIG_DEBUG_USE_UART

CONFIG_DEBUG_UART1 を CONFIG_DEBUG_UART0

Elle Edit Bource Keractor Bavigate Begrun Eroject Mit	s II Pon Window Help	
3 • 6 9 0 1 6 1 6 • 6 • 6 • 6 • 0 • 9	• @ @ # • [] + # • # • • • • *	Quick Access
• • • • • • • • • • • • • • • • • • •	Company Control Processing devices Control C	Image: Control _ Ell Della _ Ella _
	B Andrems & Task @ Console II = Properties COT Build Console [kem.uk] 11:10:26 **** Clean-only build of configuration Nios II for project keim_sdk **** [keim_sdk.clean [keim_sdk.clean complete] 11:10:37 Build Finished (took 367ms)	• CONFIG_FUNC_EINTS

config.h 画面

- 0 ×

3.1.7. プロジェクトのビルド

Nios II SBT のメニューより Project > Build All を選択し keim_sdk プロジェクト及び keim_sdk_bsp プロジェクト の両方をビルドします。Console 上に Error が表示されず、"Build Finished"が表示されればビルドが完了し、 keim_sdk フォルダ内に keim_sdk.elf ファイルが出力されます。

	O TRADE A TO VISION OF A TO VISION	Quick Access
roject Explorer 🕴 😑 📚 🖓	😤 🖻 🗟 main.c 🔒 config.h 🛱	🐃 🗆 🔡 Outline 🗈 😰 🖻 🖧 💘 🔹 🗰 💆
≝ keim_sdk	<pre>* * config.h[]</pre>	CONFIG_H_
🛛 🐮 Binaries		# CONFIG_DEBUG_UARTO
s fi Includes	#indef_CONFIG_H_	# CONFIG DEBUG UART1
e keim sdk src	Warthe _contra_n_	E CONFIG DEBUG USE UART
APL-CORE	#define CONFIG DEBUG UART0 0	CONFIG DEV 12C NUM
	#define CONFIG_DEBUG_UART1 1	CONFIG DEV LIC RTC 8564
Character and the second		
B board_init.c	/* STDIO use UART ch */	# CONFIG_DEV_12C_BR24G32
⊫ init.h	#define CONFIG_DEBUG_USE_DART CONFIG_DEBUG_DART0	CONFIG_DEV_I2C_ADT7410
⊧ @ config.h	/* Configuration for using devices */	# CONFIG_DEV_I2C_MCP23017
⊭ 📓 int.c	#define CONFIG_DEV_I2C_NUM 4	# CONFIG_DEV_SPI_W5500
> 🕼 int.h	#define CONFIG_DEV_I2C_RTC_8564 1	CONFIG_DEV_SPI_ADC1085022
> 🕼 main.c	#define CONFIG_DEV_I2C_BR24G32 1	# CONFIG_DEV_SPI_DAC0825085
e obi	#define CONFIG_DEV_I2C_ADT7410 1	# CONFIG DEV SPI TSC2046
#2 keim_sdk.elf - [alteranios2/le]	Wdetine courid_dev_12C_ncP23017 1	CONFIG DEV PWM BUZZER
N keim odk man	#define CONFIG DEV SPI W5500 1	CONFIG DEV DWM SERVO
in heim adh shideren	#define CONFIG_DEV_SPI_ADC1085022 1	- CONFIG DEV_PHIN_DECKIO
keim_sak.objaump	#define CONFIG_DEV_SPI_DAC082S085 1	CONFIG_DEV_PWM_BACKLIGHT
Sa Makefile	#define CONFIG_DEV_SPI_TSC2046 0	CONFIG_DEV_UART_RS232C
keim_sdk_bsp [keim08core]	Reading CONFIG DEN DIA DITTER 1	# CONFIG_DEV_UART_USB
	#define CONFIG DEV_PWH_BOZZER 1	# CONFIG_DEV_LCD
	#define CONFIG DEV PWM BACKLIGHT 0	CONFIG_FUNC_EINTO
		# CONFIG_FUNC_EINT1
	#define CONFIG_DEV_UART_RS232C 1	# CONFIG FUNC EINT2
	#define CONFIG_DEV_UART_USB 1	CONFIG FUNC FINT3
	Hadina CONFIG DEV LCD	- CONFIG FUNC FINTA
	derine confid_bev_ccb	
		CONFIG_FONC_EINTS
	🔃 Problems 🖉 Tasks 🗳 Console 🗱 🔲 Properties	🕹 😌 🖼 🔝 🖛 😁 🖛 📑 🖛
	CDT Build Console [keim_sdk_bsp]	
	11:31:54 **** Build of configuration Nios II for project keim sdk bsp ****	
	make all	
	[BSP build complete]	
	11-31-55 Build Simiched (took 1c 171mc)	
	11:51:55 BUILG FIRISREG (LOOK 15:1/185)	

Build 完了後画面



3.1.8. プログラムの実行

ここでは、作成した Hello World 表示プログラムを KEIm SoM ヘダウンロードし、プログラムを実行する手順を 記載します。

①ベースボードの電源を ON

SW7をONにし、電源を入れます。電源が入るとベースボード上のLED9が点灯します。

②ターミナルソフトの起動

TeraTerm を起動し、通信ポート(COM 番号)及び通信フォーマットを設定します。

TeraTermのメニューより設定 > シリアルポート(E)…を選択し下記画面を立ち上げます。

このうちポートは使用するポートを選択し、その他の通信フォーマットは下記の通り設定してください。

設定が終われば OK をクリックします。

Tera Term: シリアルポー	ト設定		X
ポート(P):	COM9	•	ОК
ホー・レート(<u>B)</u> : データ(<u>D</u>):	8 bit	•	キャンセル
パリティ(<u>A)</u> : ストップ(S):	none	•	ヘルプ(н)
ストッフ(<u>s</u>). フロー制御(<u>F)</u> :	none	•	
送信遅延 0 ミリ利	》/字(<u>C</u>) 0	щ	リ秒/行(_)

Tera Term シリアルポート設定画面

③Run Configurations

Nios II SBT のメニューより Run > Run Configurations...を選択し、Run Configurations 画面(下図)を 立ち上げる。次に、左側に表示されている Configuration Type 中 Nios II Hardware を選択し 右クリック > New を選択します。

Run Configurations	
Create, manage, and run configuration	ns
Image: State	Configure launch settings from this dialog: • Press the 'New' button to create a configuration of the selected type. • Press the 'Duplicate' button to copy the selected configuration. • Press the 'Delete' button to remove the selected configuration. • Press the 'Delete' button to configure filtering options. • Edit or view on existing configuration by selecting it. Configure launch perspective settings from the <u>Perspectives</u> ' preference page.
Filter matched 7 of 7 items	
?	Bun Close

Run Configurations 画面



④Project タブの Project name	を使用するプロジェクトに設定します。
---------------------------	--------------------

Create, manage, and run configu Nios II Hardware Tab Group	ations	
ĵ 🖹 ¥ 🖻 ≱ ▾ ype filter text		
C/C++ Application C/C++ Remote Application Launch Group	Project name: kem_adk Project Elif fie name: [CKEIm_SOKWorkspacekem_adk.eff	
 m Nios II Hardware New_configuration Nios II Hardware v2 (beta) Nios II ModelSim Nios II ModelSim v2 (beta) 	Chable browse for file system ELF file File system ELF file Adva Adva	Iced
ilter matched 8 of 8 items	Apply Revert	
(?)	Run Close	

⑤Target Connection タブの Refresh Connections をクリックし、Processors 欄に下図の様に SoM 内の Nios II 情報が表示されることを確認します。

Run Configurations	-				1000	1000	×
Create, manage, and run configuration The expected CPU name does not ma	ons atch the selected target CPU n	ame.					
📑 🗎 🗶 📄 🐡 🔹	New_configuration	n					
type filter text	Project	nection 🌾 Debugge	Source)	Common			
C/C++ Application C/C++ Remote Application	Connections Processors:			-			
Launch Group	Cable	Device	Device ID	Instance D	Name	Architecture	Refresh Connections
Nios II Hardware	JSB-Blaster on localhost	[USB-1] 10M08SA(.]	1	0	nios2 0	Nios2:3	Resolve Names
Nios II Hardware v2 (beta)							System ID Properties
Nios II ModelSim	Dute Observe Davisate						
📕 Nios II ModelSim v2 (beta)	Cable	Device	Device ID	Instance D	Name	Version	
	Disable Nios I Console' v Quartus Project File name < U System ID checks Janore mismatched system Janore mismatched system Download Download Download Staft processor Reset the selected target	ew Ising defauit .sopcinfo & .jc n D n timestamp I target system system	li files extracted f	rom ELF >			
	•						•
Filter matched 8 of 8 items					Арр	μ	Re <u>v</u> ert
(?)					<u>R</u> ui	n	Close

上記の設定が完了後、Apply をクリックした後、Run をクリックするとプログラムが KEIm SoM へ ダウンロードされ、その後実行されます。プログラムを実行した結果、ターミナルへ"Hello World"が 表示されます。



実行後のターミナル画面

プログラムを再度実行する場合、Project Explorer 上で keim_sdk フォルダを選択し、 右クリック > Run As > Nios II Hardware を選択すれば実行できます。

3.1.9. プログラムのデバッグについて

デバッグする際は、Project Explorer 上で keim_sdk フォルダを選択し、右クリック > Debug As > Nios II Hardware を選択するとデバッガが起動します。詳しいデバッガの使い方は、Altera 社のホームページより ご確認ください。



3.2. LED 点滅プログラム

LED の点滅を行うプログラムを作成します。

3.2.1. ソースファイル

LED の点滅を行うサンプルアプリケーション LedApplSampleSoft.zip を弊社 Web サイトよりダウンロードして御 使用ください。LedApplSampleSoft.zip を解凍すると以下のフォルダ構成になっております。

LedApplSampleSoft	doc	LED_sample_application_manual_v1.0.0.pdf
	src	LedSampleApl.c
		LedSampleApl.h
		main.c

LED のサンプルアプリケーションの詳細は解凍後の LedApplSampleSoft フォルダ > doc にある アプリケーションマニュアルを参照ください。

①Nios II SBT のメニューより File > New > Nios II Application を選択し、下記赤枠部分を設定後、 Finish をクリックします。

Project name は任意ですがここでは LEDApplSampleSoft とします。

又、BSP location は keim_sdk_bsp 選択します。

Nios II Application	
Nios II Application	
Create a new Nios II Software Build Tools application project	
Project name LEDApp SampleSoft	
BSP location: C:¥KEIm_SDK¥software¥keim_sdk_bsp	
	Create
☑ Use default location	
Location: C:¥KEIm_SDK¥software¥LEDApplSampleSoft	
Additional arguments:	
	Q.
Command:	
nios2-app-generate-makefileapp-dirbsp-dir/keim_sdk_bspelf-name LEDApplSampleSo	oft.elf no-src
✓ Use relative path	
0	Einish Cancel

Nios II Application 画面



```
②Nios II SBT のメニューから File > Import をクリックし Import を立ち上げます。
```

General > Archive File を選択し Next をクリックします。

Import					(100)	□ X
Select Import resources from an archive	file into an existing pro	ject.				Ľ
Select an import source:						
type filter text						
🔺 🗁 General						
Archive File						
🥩 Existing Projects into Work	space					
🚞 File System						
, Preferences						
⊳ 🗁 C/C++						
CVS						=
👂 📂 Git						
🛛 🗁 Install						
🕞 🗁 Nios II Software Build Tools F	Project					
Remote Systems						
» 🗁 RPM						
🕞 📂 Run/Debug						
🕫 🗁 Tasks						
- Toom						
			· · · · · · · · · · · · · · · · · · ·			
(?)		< <u>B</u> ack	Next >	Einish	Cancel	
•		- 2004		Ennan	Curre	

Import 画面(Select)

From archive file: keim_sdk_src_verxx.zip ファイルを指定します。 Into folder: LEDApplSampleSoft を指定し Finish をクリックします。

Import			
Archive file			D.
Import the contents of an archive file in zip or tar format from the local file system.			~ 4
From archive file: C:\KEIm_SDK\keim_sdk_src_ver1.0.0.zip		-	Browse
• • • •			
Filter Types Select All Deselect All			
Into folder: LEDApplSampleSoft			Browse
Qverwrite existing resources without warning			
(V) < Back	Next >	Finish	Cancel

Import 画面(ファイル指定)

③Nios II SBT のメニューから File > Import をクリックし Import を立ち上げます。

General を選択すると General の下にサブメニューが表示されますので、"File System"を選択した状態で Next をクリックします。

Select	e local file system int	o an existing pro	iect.		P
Select an import source:	,				
type filter text					
Archive File Existing Projects File System File System C/C++ CVS GIt	into Workspace				
0		< <u>B</u> ack	Next >	Einish	Cancel

Import 画面(Select)

④From directory:で先ほど解凍したフォルダを選択してください。

出てきたフォルダにチェックを入れてください。

IntoFolder:LEDApplSampleSoft/keim_sdk_src を指定し、Finish をクリックします。

Import				
File system Import resources from the local file system.				
From directory: C:¥KEIm_SDK¥ApplSampleSoft¥Led	ApplSampleSoft¥src			Browse
		[7]]		
Filter Iypes Select All Into folder: LEDApplSampleSoft/keim_sdk_src Options	Deselect All			Browse
Qverwrite existing resources without warning Create top-level folder Advanced >>				
3	< <u>B</u> ack	Next >	Einish	Cancel

Import 画面(ファイル指定)

全て上書きをします。

Question		- Roman	
Overwrite 'n	ain.c' in folder 'LEDApplSamp	leSoft¥keim_sdk_src'?	



3.2.2. LED 点滅プログラムの実行

プログラムを実行するために下記の手順を実施してください。

Hello World 表示プログラムで実施したのと同じ手順でプロジェクトのビルド〜実行をします。

①Nios II SBT のメニューより Project > Build All を選択しビルドを実行します。

②Nios II SBT の Project Explorer 上で LedApplSampleSoft フォルダを選択し、

右クリック > Run As > Nios II Hardware の順に選択します。

プログラムが実行されるとLED が全点灯します。ベースボードの SW1 を押すたびに LED の点灯状態が 変わります。



4. 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ベースボードの電源を ON

SW7をONにし、電源を入れます。電源が入るとベースボード上のLED9が点灯します。

②Flash 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		X
File Options Tools Help		
Target hardware information		
BSP Settings File name:		
SOPC Information File name:		
CPU to program flash:		
Hardware connection:	Connection: USB-Blaster on localhost [USB-2] Device: 10M08SA(. JES) (10M08S(C JF) @ 1 Device ID: 1 CPU Instance ID: 0 CPU Name: nios2_0 CPU Architecture: Nios2	Connections
Nios II Flash Programme	ır	
This tool is the GUI version of t	he command line tool: nios2-flash-programmer-generate.	
It parses a SOPC Information I	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 or open an existing project usi	from the top level menu bar to create a new flash settings project, ng the File->Open menu item.	
Information Problems Processi	no.	
Information Problems	3	
		Start Exit

Flash Programmer メイン画面



③New 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。

🛃 New Flash Programmer Sett	tings File		×
🔘 Get flash programmer system deta	ails from BSP Settings File		
Get flash programmer system deta	ails from SOPC Information File		
BSP Settings File name:			
SOPC Information File name:	C:\KEIm_SDK\keim08core.sopcinfo		
Master CPU name:	nios2_gen2 👻		
Flash memory:	onchip_flash_data, flash		
		OK Can	cel

New Flash Programmer Settings 画面

④Hardware 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。表示されない場合、ベースボードの電源が入っているか再度確認して ください。

onnections						
Cable	Device	Device ID	Instance ID	Name	Architecture	Refresh Connections
SB-Blaster on localhost [USB-2]	10M08SA(.	1	0	nios2 0	Nios2:3	Resolve Names
						System ID Properties
Quartus Project File name:						
uartus Project File name:						
Quartus Project File name: System ID checks						

Hardware Connections 画面



⑤ROM 書き込みファイルの設定

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

🔬 Nios II Flash Programmer	er	- 0 ×
File Options Tools Help		
Target hardware information		
BSP Settings File name:	а 	
SOPC Information File name:	: C: \KEIm_SDK\keim08core.sopcinfo	
CPU to program flash:	: nios2_gen2	
Hardware connection:	: Connection: USB-Blaster on localhost [USB-2] Device: 10M08SA(-,IES) 10M08S(C(F)@1 Device ID: 1 CPU Instance ID: 0 CPU Name: nios2_0 CPU Architecture: Nios2	Connections
Elacht flach Interfer anders flack		
ridshi hoshi riash: onchip_hash_		
Base address: 0x2000000	0 Memory span: 0x400000	
Master CPU: nios2_gen2	.zp file system offset in BSP:	
Hies for flash conversion:		664
Presidence	Concessorrype real-onset	Remove
pranein sonevorkspaceakern s	SURFACHI SURFACHI SURFACHI SURFACHI	
File generation command:		
elf2flash innut="C*/KEIm S	STM (workenerg/baim edw/defoutput="f" (NETE STM (flash/baim edw flash flash"	D
boot="nios2eds/components/	Saltera_nics2/boot_loader_ofi.srco*base=0.5200000end=0x2400000verbose	Propercies
	•	
File programming command:		
nios2-flash-programmer "C:/M	/KEIn_S0K/flash/kein_sdk_flash.flash"base=0x2000000sidp=0x200040id=0x1timestamp=1438844781device=1instance=0	
'cable=USB-Blaster on loca	calhost [USB-2]'programverbose	
Information Problems Processin	shq	
A The expected CPU name does i	s not match the selected target CPU name.	
	Start	Ežit

ROM 書き込みファイルの選択

⑥プログラムの実行

Startを選択すると、ROM へのプログラムが実行され、Processing ウィンドウに

"Leaving target processer pused"が表示されればプログラム完了です。

Nios II Flash Programm	er	the state of the second		
ile Options Tools Help				
Target hardware information				
BSP Settings File name	:			
SOPC Information File name	: C:\KEIm_SDK\keim08core.sopcinfo			
CPU to program flash	: nias2 gen2			
Hardware connection	Connection: LISB.Blacter on locabost (LISB.2)	Device: 10M08SA(JES) 10M08S(CJE)@1 Device ID: 1 CPU Insta	ance ID: 0 CPU Name: pige? 0 CPU Architecture: Ni	connections
	connection and blaster of focuriost [add 1]	j bener to obskilles/to not (i) jet bener to i a o tote		
Flash: flash Flash: onchip_flas	n_data			
Base address: 0x20000	0 Memory span: 0x40000	00		
Master CPU: nios2_ger	12 .zip file system offset in BSP:			
Files for flash conversion:				
File Name		Conversion Type	Flash Offset	Add
::WEIn SOK#workspace¥keia	adk¥kein sdk.elf	FI F	4	Remove
le generation command: f2f ash input="C:/KEIa -boot="nios2eds/component	_SOK/workspace/keim_sdk/keim_sdk.elf" _\$/altera_nios2/boot_loader_cfi.srec"	output="C:/KEIm_SDK/flash/keim_sdk_flash.flash" -base=0x2000000end=0x2400000 -reset=0x2000000verbc		Properties
ile generation command: el f2 f1 s.sh i.neut="C; /KE1 boot = "ni os2eds/component ile programming command: ni os2-f1 s.sh. programmer "C; bit i.n. p. p	_SOK/norkssace/keis_sdk/keis_sdk.keif" s/siters_nios2/boot_loader_ofi.sree" //Eis_SOK/fish/keis_sdk_fish.fish"	oxtoxt="C=//Elm_SDM/flash/Keim_gdd_flash.flash" -base=0x200000end=0x240000reset=0x200000verba base=0x2000000sid=0x20040id=0x1timestame=145	sse 18844781device=1instance=0	Properties
ile generation command: slf2flashinput="C:/KEIa boot="nios2eds/component ile programming command: nios2-flash-programmer "C: cable=USB-Blaster on to	_SOM/vorkspace/koim_sdk/koim_sdk.neff* s/alters_nios2/boot_loader_ofisree*** /NEIm_SOM/flash/keim_sdk_flash.flash* calhost [USB-2]*programverbose	output="C://EIm_SDK/flash/kein_sdc_flash.flash" base=0x200000end=0x2400000reset=0x2000000verbc base=0x2000000sidp=0x200040id=0x1timestamp=145	sse 18844781device=1instance=0	Properties
lle generation command: If 27 (1 sth i - ruput="0; //ET Li boot="n i os2eds/consonent lle programming command: i os2=f1 ssh=programmer "0; cable=USB=B1 aster on Li information Problems Proces	LSW/workseace/kois_sdk/koin_sdk.eff* 	oxdaxd="C=/KETa_SDK/f1aab/Noim_gdd_f1aab.f1aab" -base=0x2000000enx=0x2400000reset=0x2000000verbs base=0x2000000sidg=0x200040id=0x1timestamp=14S	sse 8844781device=1instance=0	Properties
le generation command: If??[Iathireµct="C://IEI ino22ds/command: ico2=f1ath-comment cable=USB=Blaster on Ic Information Problems Proces Information 25/8 Plaster 2.28	_SDK/vorkspace/koim_sdk/keim_sdk.eff* _sletren_nios2/boot_loader_cfl.eree* /kEIm_SDK/flash/keim_sdk_flash calhost [USB-2]*programverbose sing	output="C://EIm_SOK/f Hash/keim_sok_f Hash.f Hash" output="C://EIm_SOK/f Hash/keim_sok_f Hash.f Hash" base=0x2000000end=0x200040id=0x1t imestamp=145	sse 8844781device=1Instance=0	Properties
lle generation command: If 2f 1ashincut="0:2//EIs boot="nioszeds/comeonent lle programming command: cableUSB-Blaster on 1c information [Problems] Process cableUSB-Blaster on 1c information [Problems] Process cableUSB-Blaster on 1c cableUSB-Blaster on 1c	_SDM/workspace/koim_sdk/koim_sdk.eff" _saltera_niesZ/boot_loader_off.eree" /KEIm_SDM/flash/keim_sdk_flash.flash" -a lhoat (USB-2)"programverboe	output="C:/VEIm_SOK/flash/kein_sdc_flash.flash" -base=0x2000000end=0x2400000reset=0x2000000verbc base=0x2000000sidp=0x200040id=0x1timestamp=145	sse 18844781device=1instance=0	Properties
lle generation command: si f2f lash i neut="C://kElk boot="n i no2dsi/concoment lle programming command: inder_flash=programmer" cable=USB=Blaster on lo information_Problems_Process hecksummed/end \$38 m 2.25 race not require same not require composition (1%). Programming 000000 (1%). Programming	_SDK/workspace/keim_sdk/keim_sdk.eff** s/aitera_nicos/hoot_loader_off.arec*** /KEIm_SDK/flash/keim_sdk_flash* calhost [USB-2]**-programverbose	output="C-/KEIn_SDK//lish/Nein_sdk_flish/lish/lish/ base=0x200000end=0x240000reset=0x200000verb base=0x200000side=0x200040id=0x1timestame=145	sse 8844781device=1instance=0	Proper tes
lle generation command: If (21 (ash i-ruput="C:)//ELI in (ozedas/command: ile programming command: ile programming command: in (ozed f ash brogrammer "C: information Problems Proces information Proces information Problems Proces information	_SDK/vorkspace/koim_sdk/keim_sdk.eff** /altera_nico27bool_loader_cfi.arec*** /KEIm_SDK//leath/keim_sdk_fiesh.fiesh* calhost [USB-2]*programverbose	output="C://EIm_SOK/f Hash/keim_sok_f Hash.f Hash" -base=b:2000000end=D:2400000reset=b:2000000verbs base=D:2000000side=D:200040id=D:1t imestame=143	sse 8844781device=1Instance=0	Properties
ile generation command: If [21] ash i ruput="C:///EI boott" in los2eds/conservent lie programming command: in los2-11 ash-programmer "C: cobl e=USB-Blaster on 1c information	LSDK/horitssaca/koia_sdk/hein_sdk.elf* s/sitera_nios2/boot_loader_ofl.srec* /REIn_SDK/flash/kein_sdk_flash.flash* calhost (USD-2)*programverbose	oxtoxt="C-/KEIn_SDK//Isbh/Kein_ydk_/Isbh/Fish" -base=0x200000end=0x2400000reset=0x200000verb base=0x2000000side=0x200040id=0x1timestame=145	see 8844781device=1instance=0	Proper tes
ile generation command: If 27 lash irevel = "C2//IEL ile programming command: ile programming command: ile programming command: index lash programming index lash programming 0000000 (0%): Programming 000000 (0%): Programming 00000 (0%): Progra	_SDK/workspace/koim_sdk.heim_sdk.eff* =valtera_nice2/boot_loader_off.ere* =valtera_nice2/boot_loader_off.ere* /KEIn_SDK/flash/keim_sdk_flash* calhost [USB-2]*programverbose =ng	output="C:/KEIn_SDK/f1sh/kein_sdk_f1sh.f1sh" -base=0x200000en=0x240000reset=0x200000verb base=0x200000sids=0x20040id=0x1timestame=145	sse 8844781device=1instance=0	Proper tes
File generation command: If (2f1 sah i rpeut="02/XIE14 boot t="i noized=/commonent File programming command: Information [Problems] Proces cable=USB=Blaster on 1c Information [Problems] Proces cable=USB=Blaster on 1c Information [Problems] Proces cable=USB=Blaster on 1c Information [Problems] Proces 	_SDK/workspace/keim_sdk/keim_sdk.eff** _valtera_nied2/bool_loader_off.aree*** /KEIm_SDK/flash/keim_sdk_flash*flash* calhost [USB+2]*programverbose	output="C:/KEIm_SOK/f lash/keim_sok_f lash.f lash" -base=bx2000000end=Dx240000reset=bx2000000verbx base=Dx2000000side=Dx200040id=Dx1t imestame=143	see 8844781device=1instance=0	Proper See
ile generation command: el f2f lash input="Cz/I/ELI boot="nio2dst/concoment ile programming command: cable=USB-Blaster on lo 	_SDK/workspace/keim_sdk/keim_sdk.eff" - s/altera_nios//boot_loader_ofl.erec" /REIm_SDK/flash/feim_sdk_flash" - calhost [USB-2]"programverbose		sse 8844781device=1instance=0	Proper bes

書き込み完了画面

5. サンプルアプリケーションについて

本書で紹介しました LED 点滅アプリケーションの他に、以下のサンプルアプリケーションを用意しております。詳細に関しましては各々のアプリケーションマニュアルを参照ください。

	アプリ名称	概要
1	ADCApplSampleSoft	DAC から SIN 波を出力し、ADC からループバックされたデータを取得し UART0 へ出力します。 *1
2	BuzzerApplSampleSoft	ボタンが押下されるとブザーが鳴ります。
3	EtherApplSampleSoft	ネットワークから入力されたコマンドに伴い、ADC、Buzzer、LED、 RTC、サーボモータ及びステッピングモータを動作させます。 *2
4	RtcApplSampleSoft	日付(固定データ)をRTC に書き込み、年月日時分秒を標準出力へ 出力します。
5	ServoApplSampleSoft	ボタンが押下されるとサーボモータが動作します。*2
6	SteppingApplSampleSoft	ボタンが押下されるとステッピングモータが動作します。*2

*1 ループバックは DAC 出力と ADC 入力をベースボード外部でショートして行っています。

*2 サーボモータ及びステッピングモータ及びステッピングモータのドライブ回路は KEIm SoM 開発キット には付属しません。



6. 更新履歴

Ver.	更新日付	内容
1.0.0	2015/09/14	新規作成
1.1.0	2016/04/14	 ・1. 概要に Quartus Prime を追記 ・各種データ入手元の記載を"CD-ROM"から"弊社 Web サイトよりダウンロ ード"に変更