

開発スターターガイド

- 1. 開発を始める前に
- 2. 開発環境のインストール
- 3. アプリケーションの作成
- 4. アプリケーションの起動とデバッグ

BlueNinjaを利用されるWindows版お客様向けのガイドです。

※開発環境のバージョンアップ手順は「開発環境アップデート手順 (/ja/update_devenv/)」を参照ください。

あわせて「FAQ（よくあるご質問） (../faq/)」もご覧ください。

本ガイドではBlueNinjaにWindowsへのGnu toolchain(GCC、GDB、binutils)のインストールとプログラムのビルド、デバッグを行う手順をまとめています。

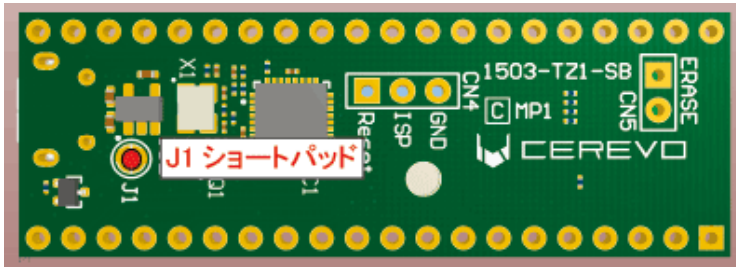
1. 開発を始める前に

本ガイドで解説する開発手順には、下記が必要です。

BlueNinja CDP-TZ01C(モジュール、ブレイクアウトボード)

インターネット接続可能なWindows 7以降が動作するPC

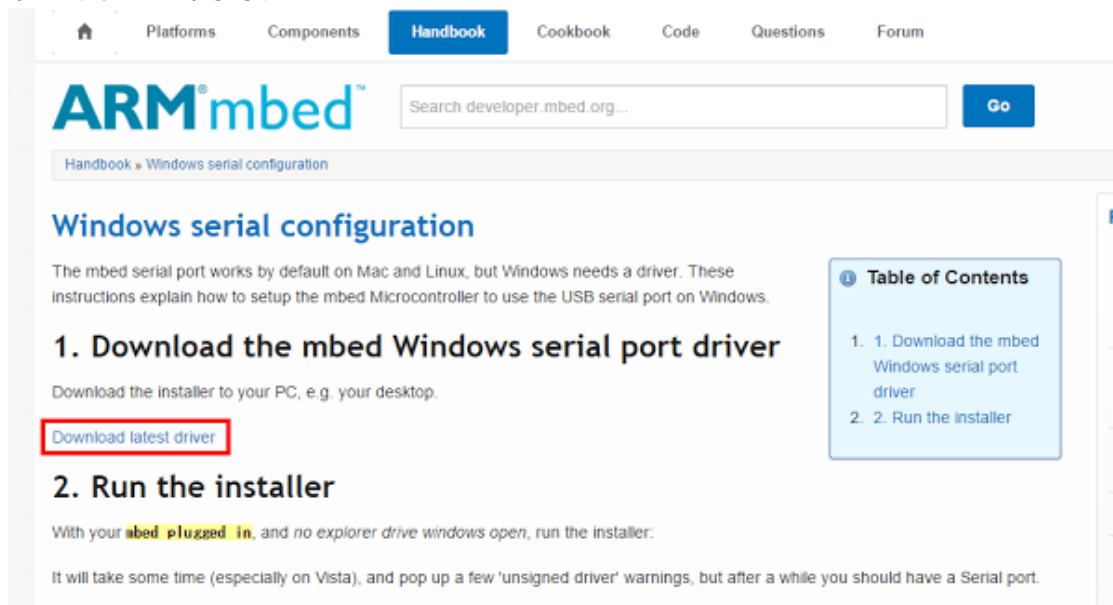
※USBから給電する場合(ブレイクアウトボードのPin17またはBlueNinja本体のCN3から電源供給を行わない場合)はJ1をショートしてください。



2. 開発環境のインストール

2-1.ブレイクアウトボードのデバイスドライバの入手とインストール

1. ブレイクアウトボードに搭載されたmbed相当のインターフェースには、デバイスドライバのインストールが必要です。mbedのサイト (<https://developer.mbed.org/handbook/Windows-serial-configuration>)からデバイスドライバをダウンロードしてください。



USBケーブルでブレイクアウトボードとPCを接続します。

ダウンロードしたデバイスドライバのインストーラーを実行します。

インストーラ実行中に署名の確認ダイアログが表示されますので、『許可』を選択してください。

ブレイクアウトボードのファームウェア更新手順 ([/ja/breakout_firmware/](#))もあわせて確認してください。

2-2.MinGW(Msys)のインストール

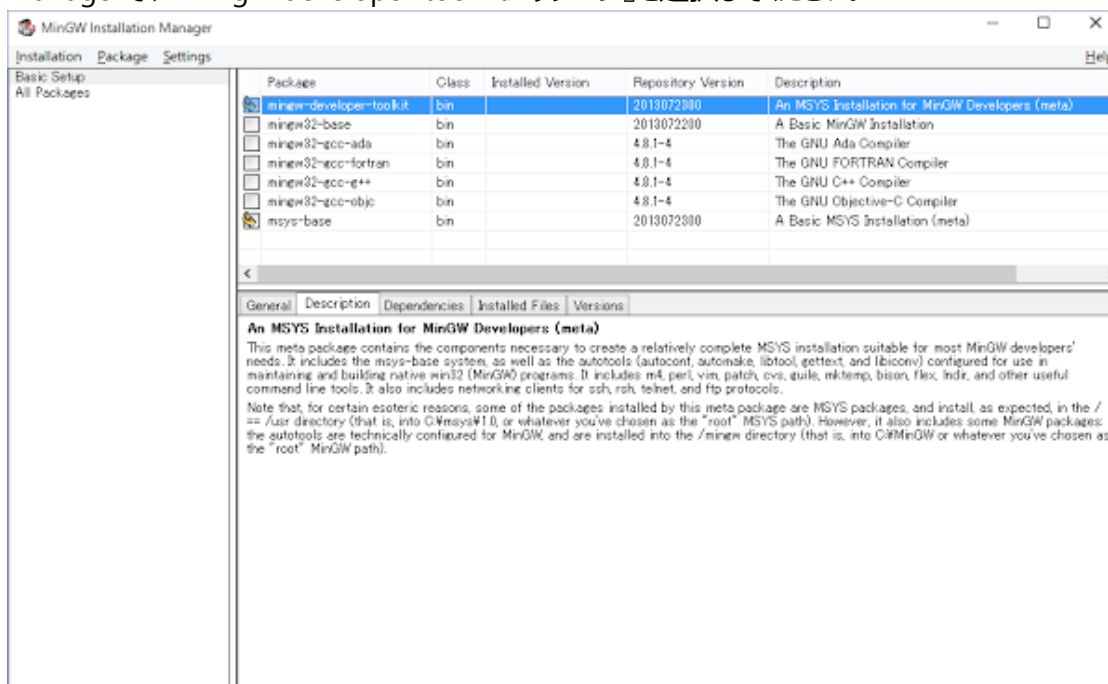
makeなどのプログラムのビルドに必要なツール群を使用するため、MinGW(Msys)をインストールを行います。

1.

MinGWのサイトから (<http://sourceforge.net/projects/mingw/files/latest/download>)mingwのインストーラをダウンロードしてください。

2. ダウンロードしたインストーラを実行します。ウィザードに従って、インストールを進めてください。インストール先などのオプションはデフォルトのままにしてください。

3. mingw-developer-toolkitパッケージを追加します。インストーラの終了後に実行されるMinGW Installation Managerで、『mingw-developer-toolkitパッケージ』を選択してください。



パッケージの選択後、InstallationメニューのApply Changesを実行します。
インストールの完了後、MinGW Installation Managerを終了します。

2-3.Git for Windowsのインストール

BlueNinja BSPはBitbucketで提供 (https://bitbucket.org/cerevo/blueninja_bsp/wiki/Home)しています。入手いただくためには、事前にGit for Windowsをインストールする必要があります。

Git for Windows (<https://msysgit.github.io/>)をダウンロードしてください。

- Git for Windows Version 2.10.2 64bitインストーラ (<https://github.com/git-for-windows/git/releases/download/v2.10.2.windows.1/Git-2.10.2-64-bit.exe>)

- Git for Windows Version 2.10.2 32bitインストーラ (<https://github.com/git-for-windows/git/releases/download/v2.10.2.windows.1/Git-2.10.2-32-bit.exe>)

※Version 2.10.2 (<https://github.com/git-for-windows/git/releases/tag/v2.10.2.windows.1>)でインストールを確認しています。上記のインストーラの使用を推奨します。

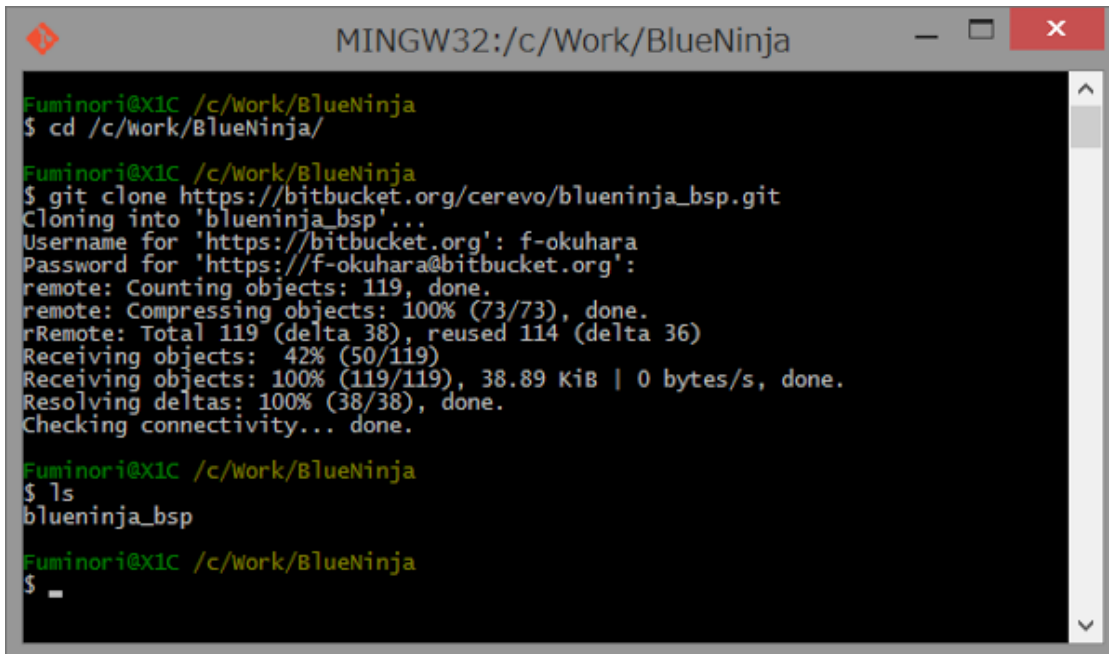
ダウンロードしたインストーラを実行します。

インストール中に表示されるオプションは、すべてデフォルトのまま変更しないでください。

2-4.BlueNinja BSPの入手

BitbucketからBlueNinja BSPリポジトリ (https://bitbucket.org/cerevo/blueninja_bsp/)をクローンします。

```
$ git clone https://bitbucket.org/cerevo/blueninja_bsp.git
```



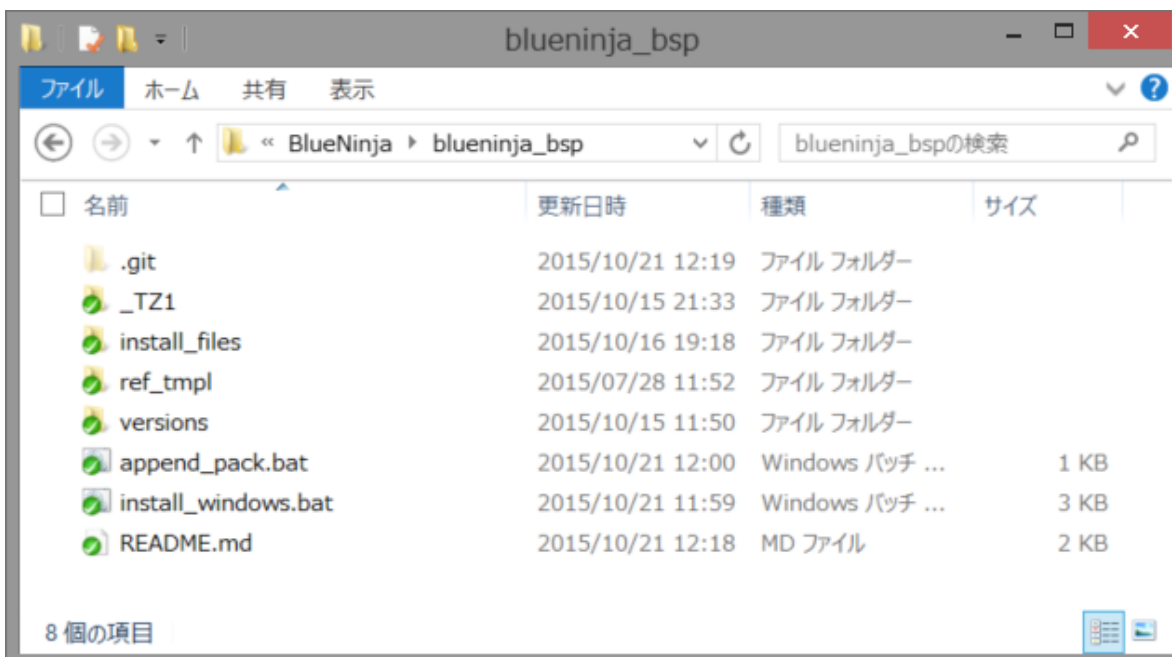
```
Fuminori@X1C /c/Work/BlueNinja
$ cd /c/work/BlueNinja/

Fuminori@X1C /c/Work/BlueNinja
$ git clone https://bitbucket.org/cerevo/blueninja_bsp.git
Cloning into 'blueninja_bsp'...
Username for 'https://bitbucket.org': f-okuhara
Password for 'https://f-okuhara@bitbucket.org':
remote: Counting objects: 119, done.
remote: Compressing objects: 100% (73/73), done.
rRemote: Total 119 (delta 38), reused 114 (delta 36)
Receiving objects: 42% (50/119)
Receiving objects: 100% (119/119), 38.89 KiB | 0 bytes/s, done.
Resolving deltas: 100% (38/38), done.
Checking connectivity... done.

Fuminori@X1C /c/Work/BlueNinja
$ ls
blueninja_bsp

Fuminori@X1C /c/Work/BlueNinja
$
```

インストールスクリプトとBSPのファイル群がダウンロードされます。



2-5.CMSISの入手

1. MDK5 Software Packs (<http://www.keil.com/dd2/pack/>)のページにアクセスします。

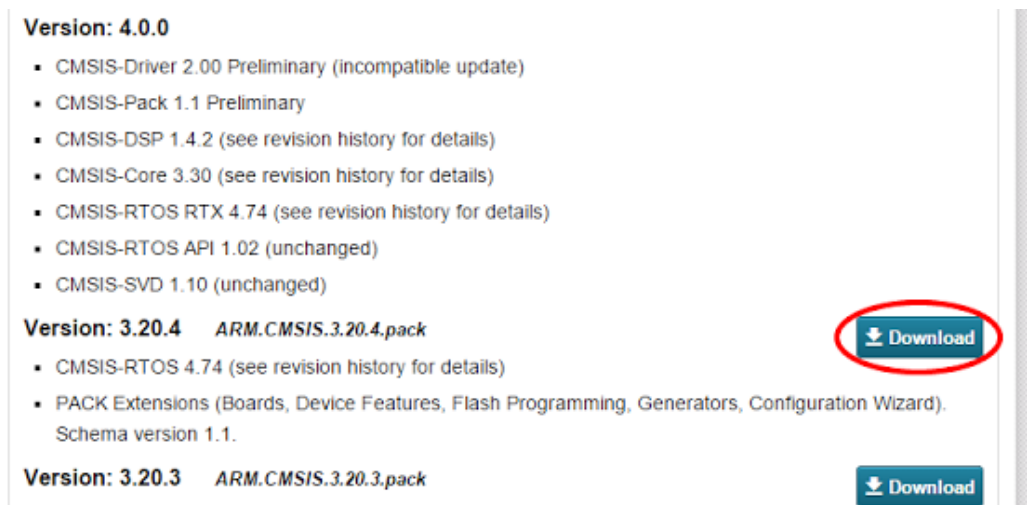
2. “ARM”の“CMSIS”をクリックすると、ダウンロードメニューが展開されます。

KeilのサイトからCMSIS 3.20.4をダウンロードします。

3.

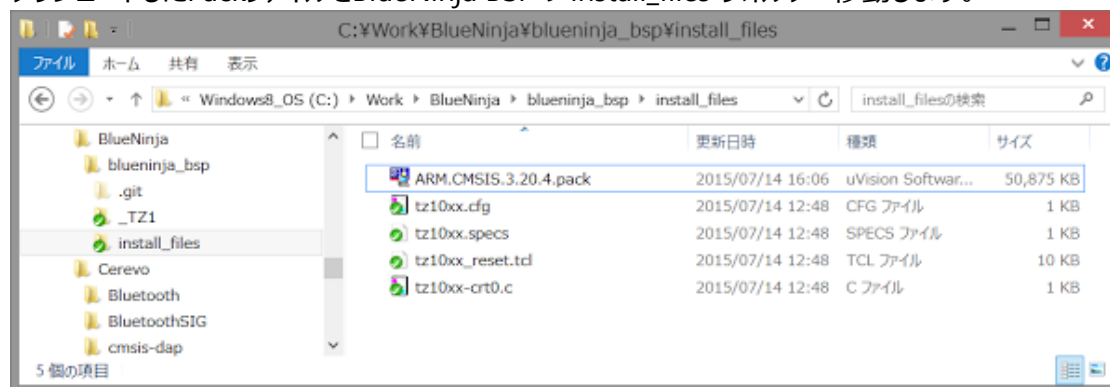
※一部環境ではpackファイルではなくzip ファイルがダウンロードされる場合がございます。その場合には下記の手順中のpack ファイルをzipファイルに読み替えて手順を実行してください。

CMSIS 3.20.4をダウンロードします。Version: 3.20.4の『Download』をクリックしてください。
異なるバージョンをダウンロードしないようご注意ください。



EULAが表示されるので『Accept』を選んでください。

ダウンロードしたPackファイルをBlueNinja BSPの“install_files”フォルダへ移動します。



2-6.TZ1000 BSPの入手

東芝のTZ1000シリーズの製品ページより (<http://toshiba.semicon-storage.com/jp/product/assp/applite/tz1000.html>)ApP Lite™ Technical CenterにログインしTZ1000のBSPを入手します。ダウンロードにはユーザー登録が必要です。

» TOSHIBA ApP Lite™ Technical Center 登録手順 (/ja/altc/)

1. ユーザー登録を行います。

2. ユーザー登録が完了したら、『お客様サポートサイト』へログインします。

3. マイページの「ApP Lite(TM) (Application Processor Lite)」リンクからApP Lite™ Technical Centerのページへ移動します。

4. ApP Lite™ Technical Centerの“BSP”から、Ver.1.31.1とVer.1.35.0、Ver.1.39.0をダウンロードします。
▶ [TZ1000_Usage_note_errata_IDE_20151224.pdf](#) [update:2016.1.]

BSP

Cerevo / BlueNinja

Download TZ1000 BSP non OS Version for MDK-ARM

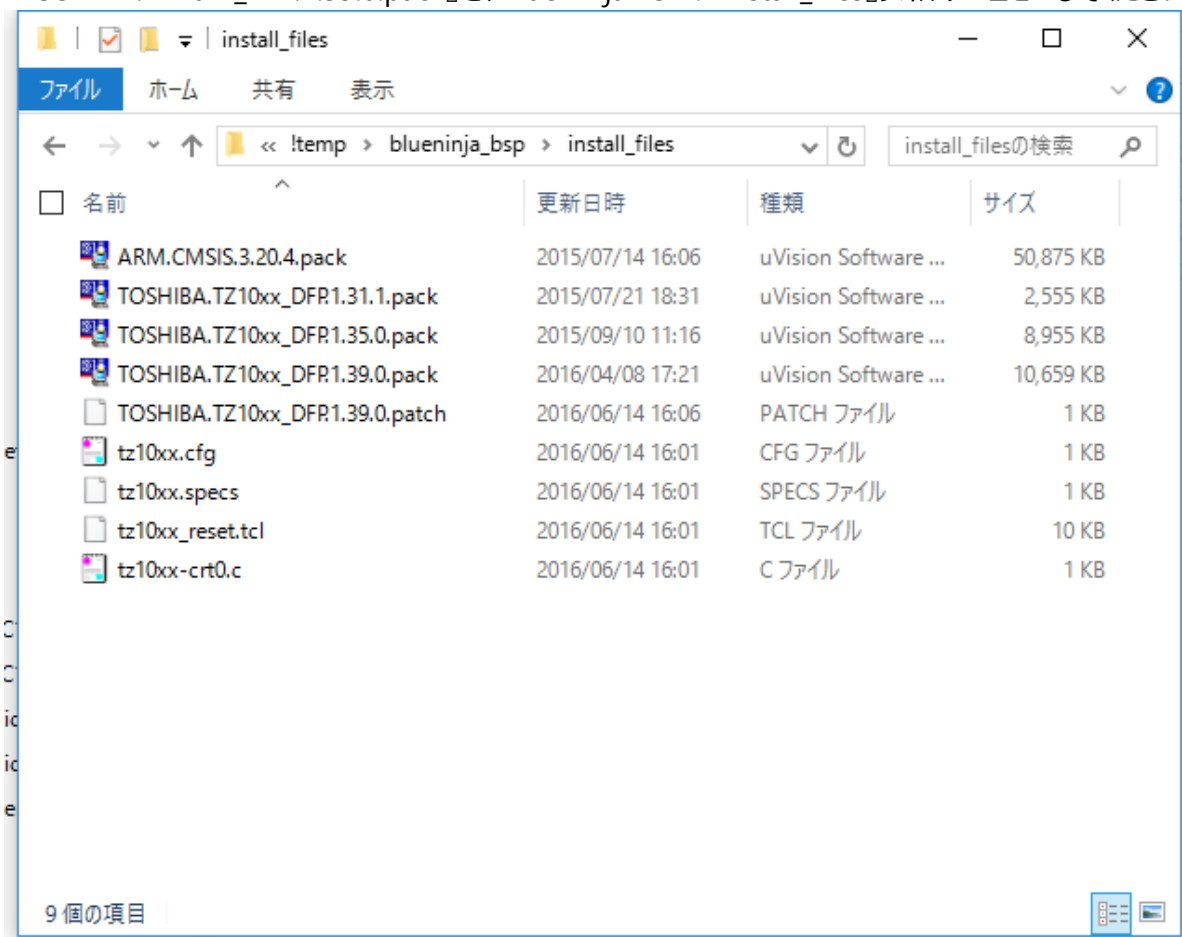
Package Name	Release Date	Note
Ver. 1.39.0	29 January, 2016	Latest version
Ver. 1.35.0	27 August, 2015	Previous version
Ver. 1.31.1	13 July, 2015	

・本製品をご使用頂く場合は、本製品に関する最新の情報（本資料、取扱

別添付ファイルをご覧ください

ダウンロードしたZIPファイルを解凍します。

解凍したファイルの中にある『TOSHIBA.TZ10xx_DFP.1.31.1.pack』、『TOSHIBA.TZ10xx_DFP.1.35.0.pack』、『TOSHIBA.TZ10xx_DFP.1.39.0.pack』を、BlueNinja BSPの『install_files』フォルダへコピーしてください。



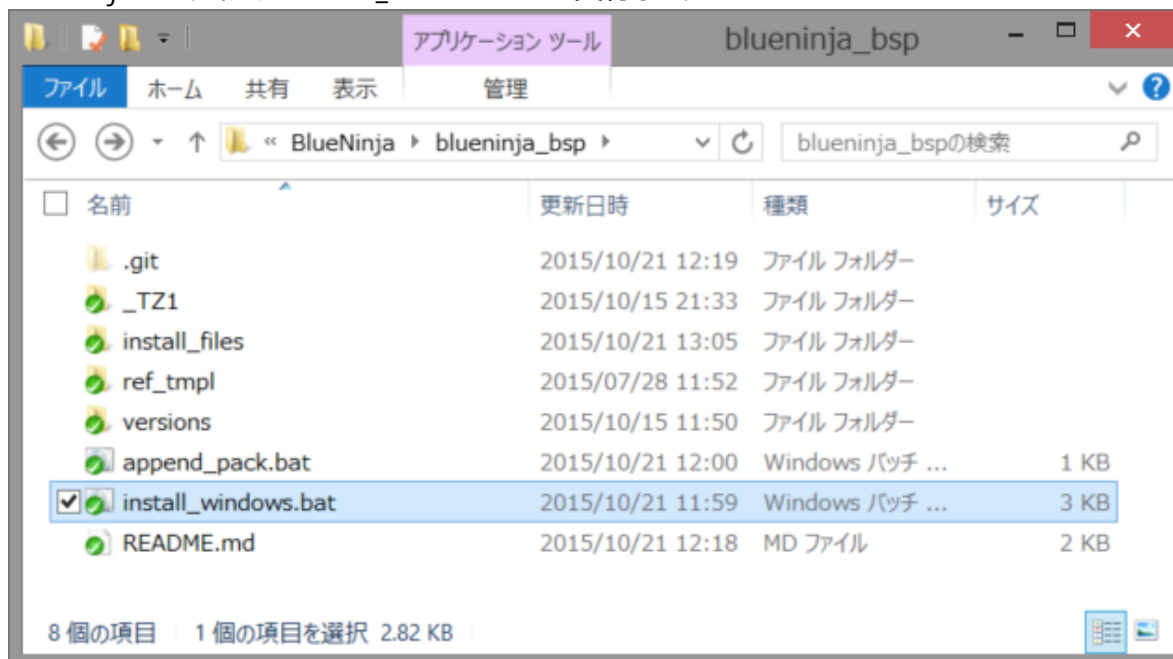
2-7.開発環境セットアップスクリプトの実行

BlueNinja BSPの開発環境をセットアップします。

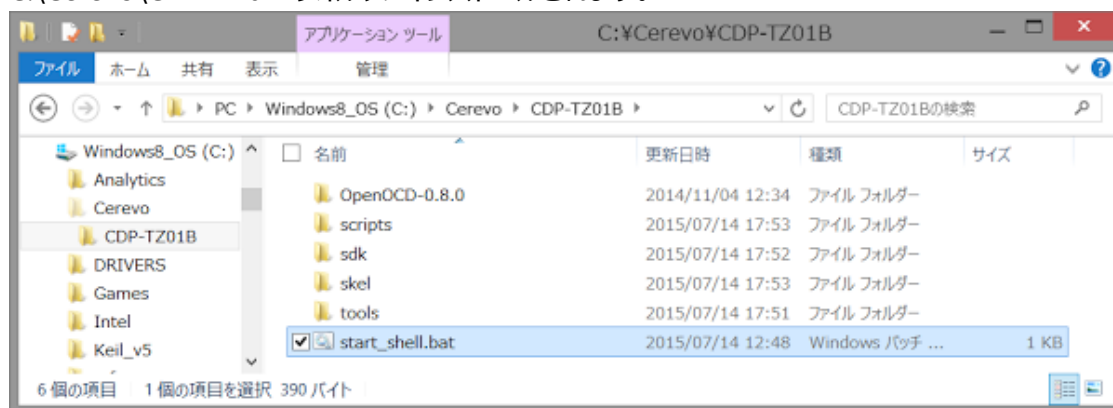
1.

同梱のセットアップスクリプトを実行すると、C:\以下にGCC ARM Embedded、OpenOCDのダウンロードと展開、BSPの展開、BlueNinja向けのパッチの適用、開発支援スクリプトのセットアップまでを自動で行います。

BlueNinja BSPフォルダのinstall_windows.batを実行します。



C:\Cerevo\CDP-TZ01Bフォルダにインストールされます。



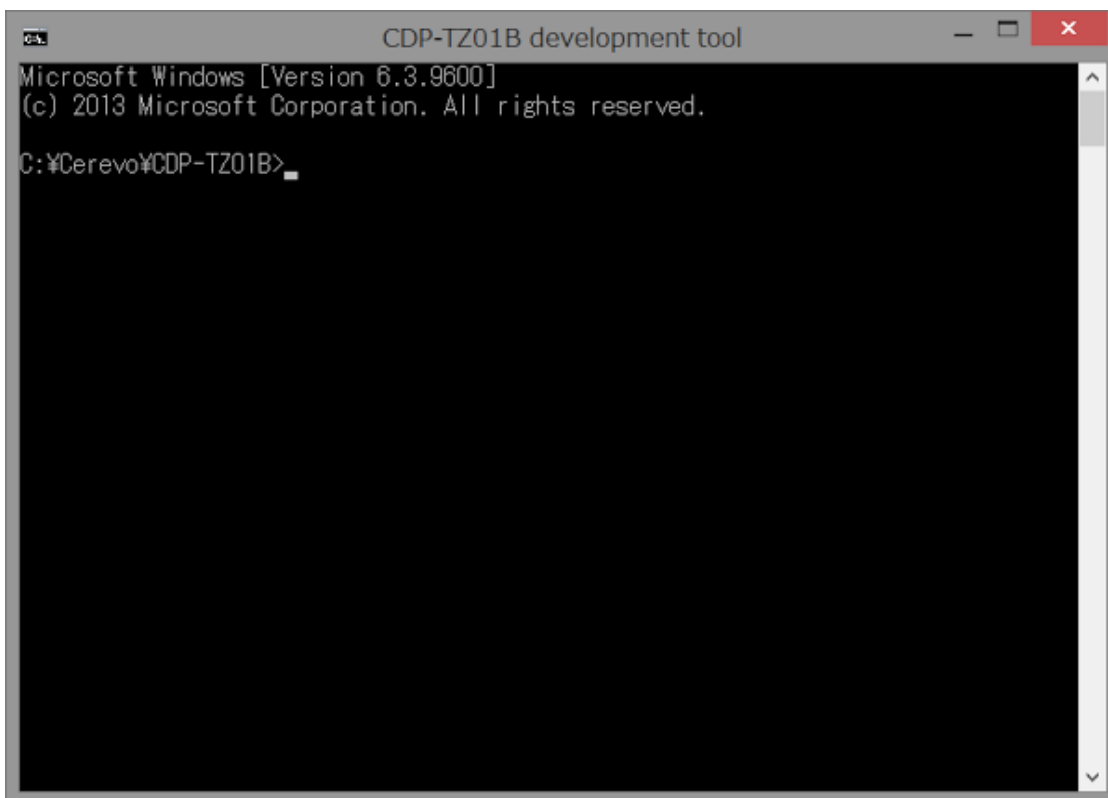
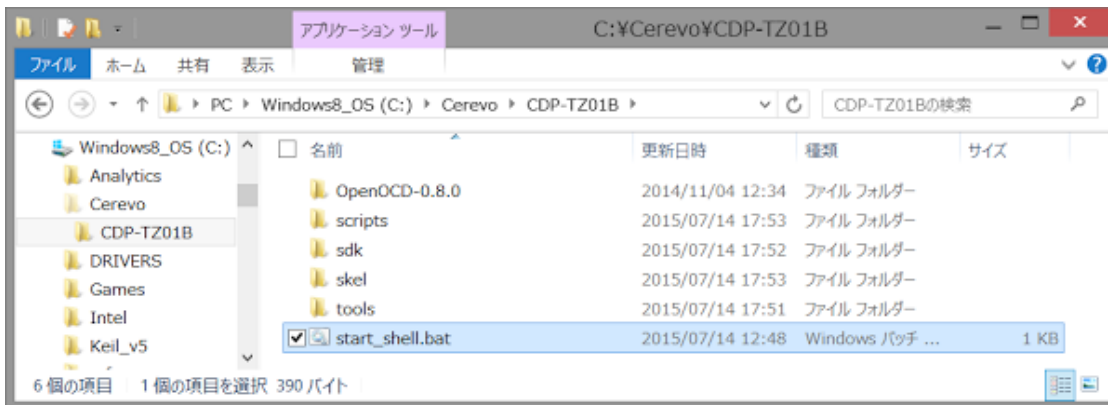
以上で開発環境の構築は完了です。

3. アプリケーションの作成

インストールした開発環境を使用して実際にプログラムをビルドします。

3-1.開発環境の起動

BlueNinja BSPのインストールフォルダ(C:\Cerevo\CDP-TZ01B)のstart_shell.batを実行してください。
プログラムのビルドに必要な設定を行った後、コマンドプロンプトが起動します。



3-2.アプリケーションのひな形の作成

任意の作業フォルダに移動します。

(このガイドでは、例としてC:\Work\BlueNinjaを使用します。任意の作業フォルダを作成する場合は、日本語やスペースを含まないパスをおすすめします。)

```
C:\Cerevo\CDP-TZ01B>cd \Work\BlueNinja  
C:\Work\BlueNinja>
```

hello_blueninja_san という名前でアプリケーションを作成します。

app_createスクリプトの引数にアプリケーション名を指定して実行します。

```
C:\Work\BlueNinja>app_create hello_blueninja_san  
"Create application directory: hello_blueninja_san"  
"Copy skeleton files."  
C:\Cerevo\CDP-TZ01B\skel\CDP_TZ01B_BSP.mk  
C:\Cerevo\CDP-TZ01B\skel\CMSIS.mk  
C:\Cerevo\CDP-TZ01B\skel\Makefile  
C:\Cerevo\CDP-TZ01B\skel\RTE.mk  
C:\Cerevo\CDP-TZ01B\skel\RTE.patch  
C:\Cerevo\CDP-TZ01B\skel\TZ10xx_DFP.mk  
C:\Cerevo\CDP-TZ01B\skel\src\main.c  
:  
省略  
:  
C:\Cerevo\CDP-TZ01B\sdk\TOSHIBA.TZ10xx_DFP\Middleware\TWiC\Config\tz1sm  
_hal_nonos.h  
27 個のファイルをコピーしました  
1 個のファイルをコピーしました。  
patching file RTE/Device/TZ1001MBG/RTE_Device.h  
patching file RTE/Middleware/TZ1001MBG/tz1sm_config.h  
patching file RTE/Middleware/TZ1001MBG/tz1sm_hal.c  
patching file RTE/Middleware/TZ1001MBG/tz1sm_hal.h  
patching file RTE/Middleware/TZ1001MBG/tz1sm_hal.h  
C:\Work\BlueNinja\hello_blueninja_san>
```

スクリプトの実行が完了すると、アプリケーションのひな形であるHelloWorld(LED点滅)プログラムがビルド可能な状態になります。

3-3.ビルド

makeを実行します。

ビルドの結果、hello_blueninja_san.elfとhello_blueninja_san.binが生成されます。

```
C:\Work\BlueNinja\hello_blueninja_san>make

C:\Cerevo\CDP-TZ01B\tools\bin\arm-none-eabi-gcc -c -MMD -MP -mcpu=cortex
-m4 -mthumb -mthumb-interwork -march=armv7e-m -mfloat-abi=softfp -mfpu=fpv4-sp-d
16 -spe
cs=tz10xx.specs -std=c99 -g -O0 -IRTE -IRTE/Device/TZ1001MBG -Isrc/boards/CD
P-TZ
01B/include -Isrc/config -Isrc/drivers/include -Isrc/utils/include -IC:\Cerevo\C
DP-TZ01B\sdk\ARM.CMSIS\CMSIS\Include -IC:\Cerevo\CDP-TZ01B\sdk\ARM.CMS
IS\CMSIS\D
river\Include -IRTE -IRTE/Device/TZ1001MBG -IC:\Cerevo\CDP-TZ01B\sdk\TOSHIB
A.TZ1
0xx_DFP/Device/Include -IC:\Cerevo\CDP-TZ01B\sdk\TOSHIBA.TZ10xx_DFP\RTE_D
river -
o build/src/main.o src/main.c
:
省略
:
C:\Cerevo\CDP-TZ01B\tools\bin\arm-none-eabi-objcopy -O binary hello_bluenin
j
a_sa
n.elf hello_blueninja_san.bin
C:\Work\BlueNinja\hello_blueninja_san>
```

3-4.書き込み

ビルドで生成されたhello_blueninja_san.binをBlueNinjaへ書き込みます。

ブレイクアウトボードにはmbed相当の書き込み機能があるので、hello_blueninja_san.binファイルをブレイクアウトボードへコピーします。

ブレイクアウトボードがEドライブとして認識している場合は、以下のように表示されます。

```
C:\Work\BlueNinja\hello_blueninja_san>copy hello_blueninja_san.bin e:
1 個のファイルをコピーしました。
C:\Work\BlueNinja\hello_blueninja_san>
```

ブレイクアウトボードへファイルのコピーができれば手段は問いません。エクスプローラーでコピーしても構いません。

書き込み中はブレイクアウトボードの緑色LEDが点滅します。

(最初の約5秒間、コピーは進捗しません。内蔵FlashROMのイレース処理の待ち時間ですので異常ではありません。)

書き込みが完了すると一旦ブレイクアウトボードのドライブが切断され、その後再認識します。

※書き込みが正常に行えない場合(FAIL.TXTが生成される/一瞬で書き込み処理が完了してしまった等)は下記を確認してください。

- ・モジュールに電源供給が正しく供給されていることを確認する

モジュールのCN3から5Vを給電、モジュールのCN2にバッテリー接続し給電、ブレイクアウトボードのJ1をショートしてUSBから給電のいずれかの手段で供給する必要があります。

- ・ファームウェアを強制的にイレースを試す

ブレイクアウトボードのERASE端子をショートした状態で電源ONを行います。ファームウェアの一部が書き込まれてしまい、TZ1000のデバッグインターフェース(CoreSight)が動作できない状態が解消されます。

詳しくは [フラッシュ強制イレース方法 \(/ja/guide/flash_erase/\)](/ja/guide/flash_erase/)を参照してください。

4. アプリケーションの起動とデバッグ

書き込みを行ったプログラムの起動とデバッグを使用したデバッグの方法を説明します。

4-1. アプリケーションの起動

ブレイクアウトボードの電源ボタン(USBコネクタと横並びのボタン)を押してください。アプリケーションが動作すると、モジュール上の赤色LEDが点滅し始めます。電源ボタンを長押し(約2秒)することで電源が切れます。

電源の管理はBlueNinja BSPに含まれるTZ01_systemライブラリで行っています。TZ1001のリセットなどでプログラムの動作が停止すると電源が切れます。

4-2. デバッグ

ブレイクアウトボードのCMSIS-DAPインターフェースとGDBとOpenOCDによりオンチップデバッグができます。

デバッグの開始

ブレイクアウトボードの電源ボタンを押したまま、引数にhello_blueninja_san.elfを指定しdebuggerスクリプトを実行します。

```
C:\Work\BlueNinja\hello_blueninja_san>debugger hello_blueninja_san.elf
```

別ウィンドウでGDBが起動しTZ1001のリセットを行った後、コマンド入力待ちになります。

```
hello_blueninja_san.elf - GDB
and "show warranty" for details.
This GDB was configured as "--host=i686-w64-mingw32 --target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from hello_blueninja_san.elf...done.
Remote debugging using localhost:3333
0x00000000 in __InitialVectors ()
adapter speed: 100 kHz
Only resetting the Cortex-M core, use a reset-init event handler to reset any pe
ripherals or configure hardware srst support.
target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000044 msp: 0x20008000
adapter speed: 4000 kHz
requesting target halt and executing a soft reset
soft_reset_halt is deprecated, please use 'reset halt' instead.
target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000044 msp: 0x20008000
(gdb)
```

以下では、GDBの代表的なコマンドを使用して実際にデバッグを行ってみます。

TZ01_system_init()の実行が終わるまで、電源ボタンは押しっぱなしにしてください。

ブレイクポイントの設定

“b”コマンドで、main()の先頭にブレイクポイントを設定します。

宣言等を除いたコードの先頭、TZ01_system_init()の呼び出しの行に設定されます。

```
(gdb) b main
Breakpoint 1 at 0x100001a4: file src/main.c, line 36.
(gdb)
```

プログラム実行の再開、ブレイクポイントによる実行中断

“cont”コマンドでプログラムの実行を再開します。

ブレイクポイントを設定しているTZ01_system_init()の呼び出し前で実行が停止します。

```
(gdb) cont
Continuing.
Breakpoint 1, main () at src/main.c:36
36 TZ01_system_init();
(gdb)
```

ステップオーバー実行

“n”コマンドでステップオーバー実行します。

TZ01_system_init()が実行されTZ01_console_init()の呼び出し前で実行が停止します。

(電源のEnableが完了していますので電源ボタンを離してください)

コマンドを入力せずにEnterすると直前のコマンドの“n”を再度実行します。

TZ01_console_init()が実行されif (...の行で実行が停止します。

```
(gdb) n
37 TZ01_console_init();
(gdb)
40 if (TZ01_system_run() == false) {
(gdb)
```

ステップイン実行

“s”コマンドでステップイン実行します。

TZ01_system_run()関数に入った先頭の行で実行が停止します。

```
(gdb) s
TZ01_system_run () at src/boards/CDP-TZ01B/TZ01_system.c:255
255 if (TZ01_system_tick_check_timeout(SYSTICK_NO_PWSW_CHECK)) {
(gdb)
```

プログラム実行の再開

“cont”コマンドでプログラムの実行を再開します。

```
(gdb) cont
Continuing.
```

プログラムの実行停止

キーボードからCtrl+Cキーを押すことでプログラムの実行が停止します。

停止した箇所のソースコードが表示されます。

```
Program received signal SIGINT, Interrupt.  
0x10000648 in TZ01_system_tick_check_timeout (tim_no=SYSTICK_NO_LED_BLINK)  
at src/boards/CDP-TZ01B/TZ01_system.c:181  
181 {  
(gdb)
```

デバッグの終了

“quit”コマンドでデバッグを終了します。

```
(gdb) quit  
A debugging session is active.  
Inferior 1 [Remote target] will be detached.  
Quit anyway? (y or n) y
```

GDBの終了後、バググラウンドで動作しているOpenOCDも終了します。

関連URL

BlueNinja BSP
(https://bitbucket.org/cerevo/blueninja_bsp/)

TZ1000シリーズ製品ページ
(<http://toshiba.semicon-storage.com/jp/product/assp/applite/tz1000.html>)

GCC ARM Embedded
(<https://launchpad.net/gcc-arm-embedded>)

Download OpenOCD for Windows
(<http://gnutoolchains.com/arm-eabi/openocd/>)
