

EVK-NINA-B1 (BLEモジュール) 開発環境の紹介

～ オフライン環境構築がちょっと楽になりました ～



mbed祭り 2017@2017@秋の虎ノ門

日時：2017年10月14日（土）13:00～17:00

会場：株式会社ウフル本社

富士エレクトロニクス株式会社

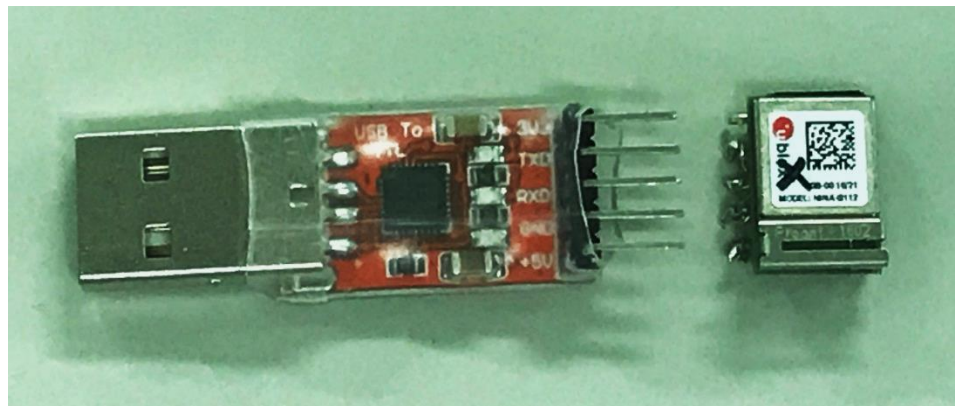
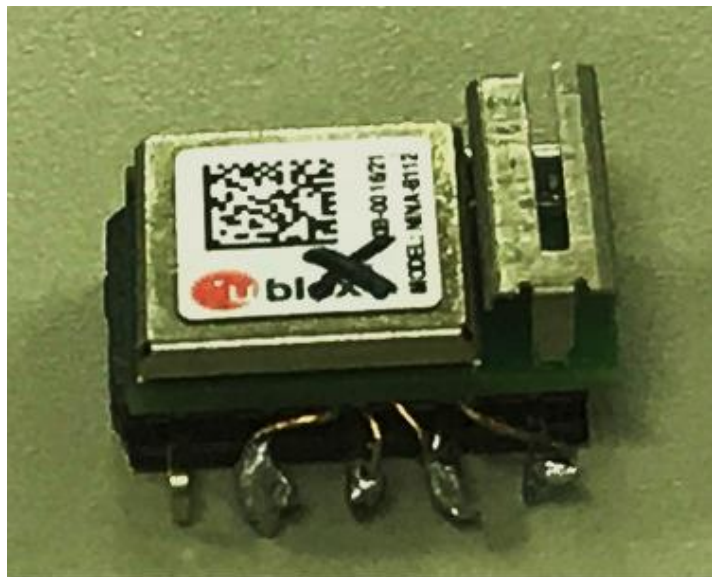
2017年10月14日

- 自己紹介、会社紹介
- NINA-B1の紹介
- 簡単オフライン開発環境
- 動作確認
- 余談



名前など : 小越博昭、47歳、既婚、子供3人
仕事場 : 富士エレクトロニクス(東京都文京区)
仕事内容 : ublox の製品のサポート
経過 : マイコン設計から転職して現在4年目
住居 : 神奈川県鶴見区
mbed歴 : 12ヶ月
休日仕事 : 子供とマイクラ.W10Verが面白くなってきた。
平日趣味 : 秋月などで面白い部品はないか物色する。

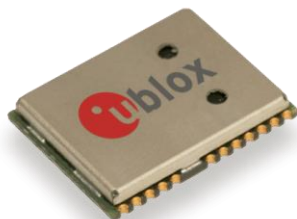
okoshi@fujiele.co.jp



VDD/VCC/TX/RX だけを繋いで動作
するかの実験



<https://www.u-blox.com/ja>



業 種：モジュールメーカー
創 立：1997年（スイス）
従業員数：約700名
所在地：スイス・タルヴィル Thalwil, Switzerland
売上高：400億円@2015年
日本法人：赤坂見附 約20名
主な製品：GNSSモジュール/デバイス ≒GPS
セルラーモジュール ≒携帯
近距離無線モジュール ≒Wifi.Bluetooth

富士エレクトロニクス株式会社 (**mF** Macnica Fuji Electronics)

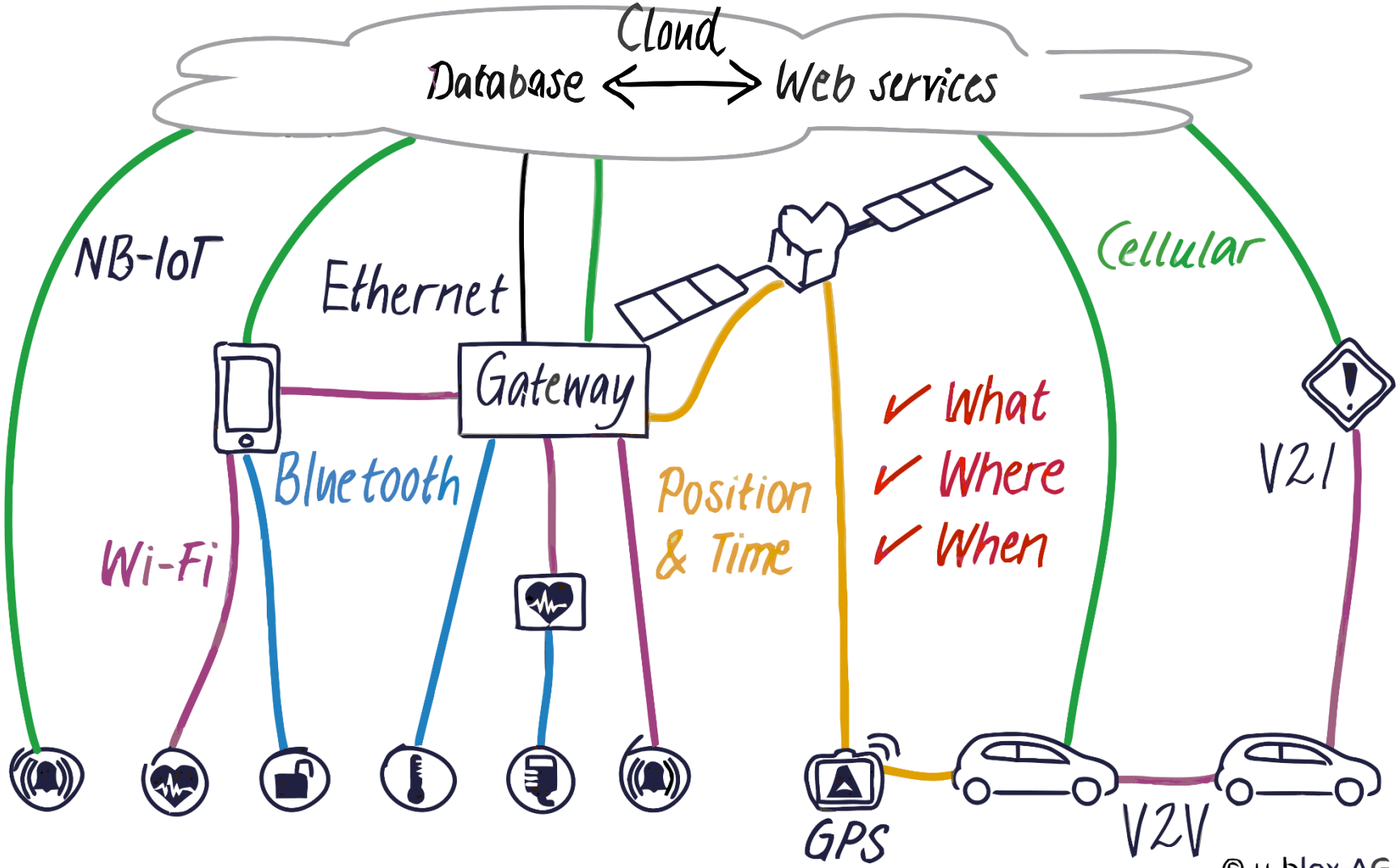
<http://www.macfehd.co.jp>



業 種：半導体商社
創 立：1990年（東京都文京区）
従業員数：約450名
所在地：東京都文京区
売上高：500億円@2015年
主な商品：半導体

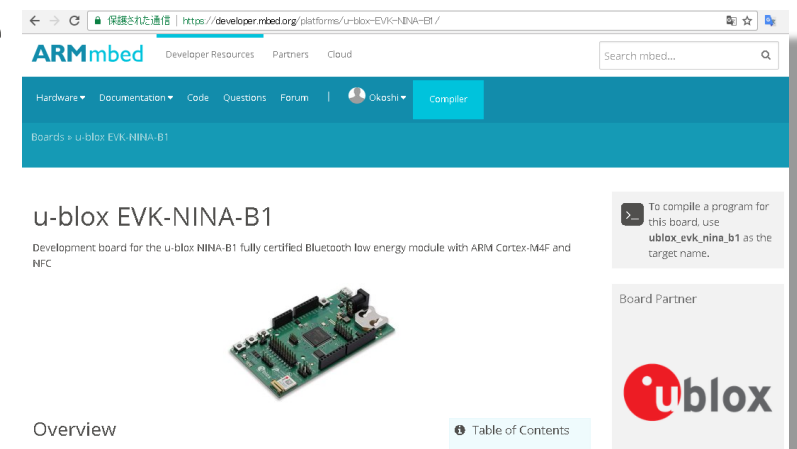
[ODIN-W2 で milkcocoa](#)

https://www.fujiele.co.jp/wp/wp-content/uploads/maker/ublox/FJE_Document/odin-w2%E3%81%A7milkcocoa.pdf



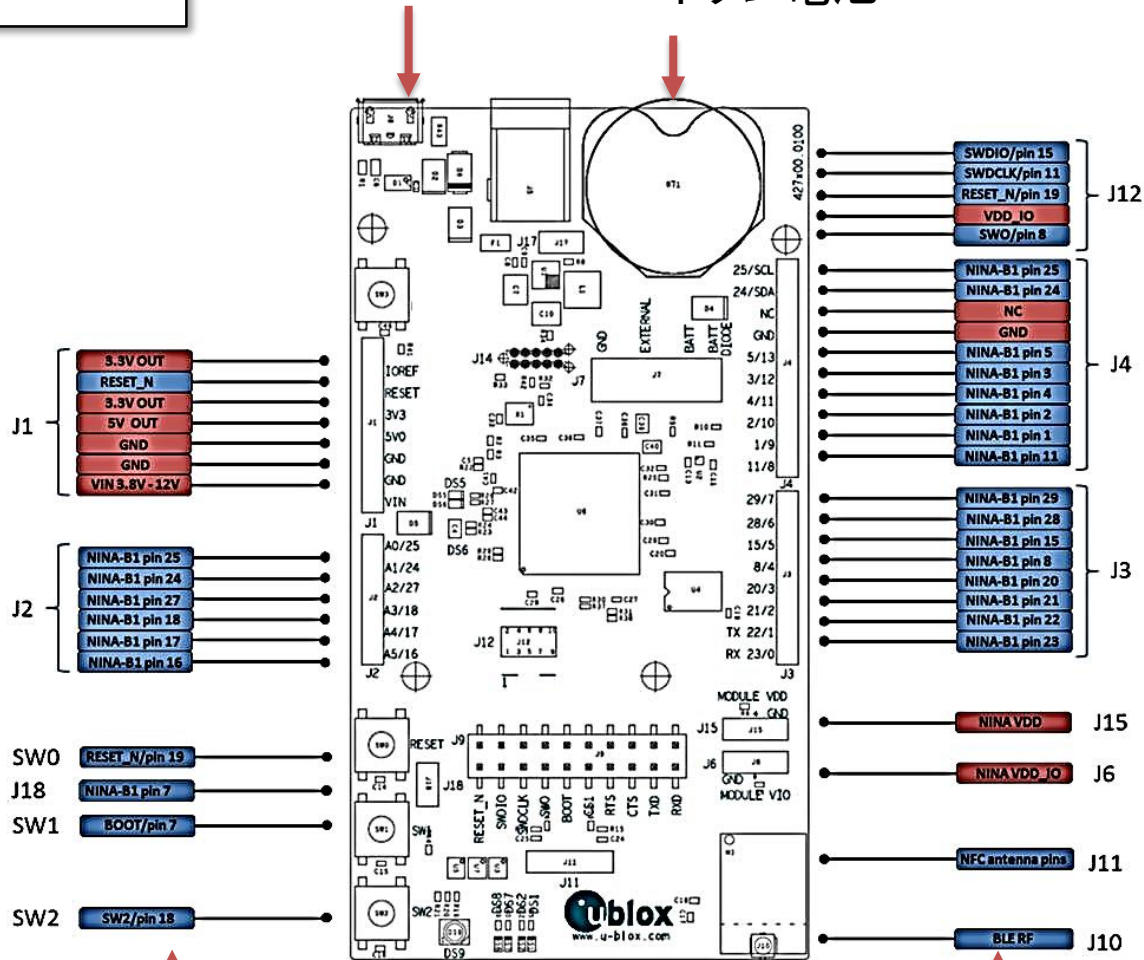
- Development kit for the u-blox NINA-B1 Bluetooth low energy v4.2 module
- Based on Nordic nRF52
- High performance ARM[®] Cortex[™]-M4F Core
 - 64 MHz
 - 512 kB flash
 - 128 kB RAM
- Optimized for ultra-low power operation
- On-chip NFC tag
- Onboard debugger hardware and firmware
- All the NINA-B1 module pins are available at connectors
- Pins for power consumption measurement
- 7 status LEDs
- 4 buttons
- Can be powered through
 - USB
 - External power supply 1.7-3.6VDC
 - CR2032 coin-cell battery, on-board battery holder

<https://developer.mbed.org/platforms/u-blox-EVK-NINA-B1/>



EVK-NINA-B1の端子

USB or external power supply & UART & J-LINK (Micro B) ボタン電池



Bluetooth LE

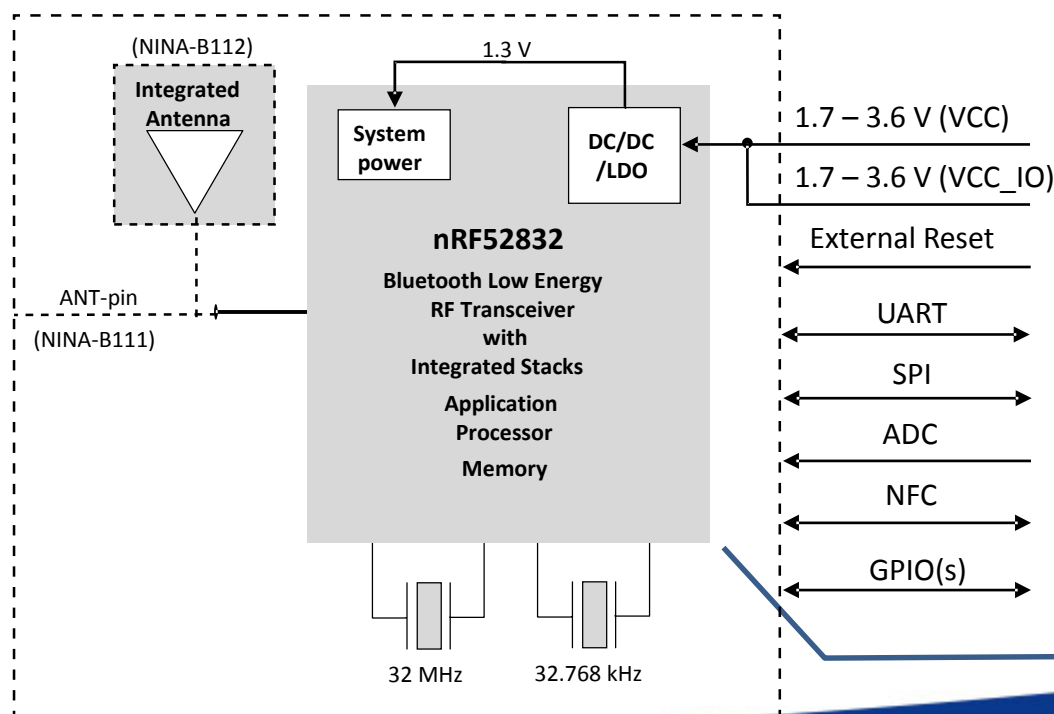


Arduino sockets in 3.3 V I/O domain

NINA-B1の機能・内部構造

- Bluetooth v4.2 with Bluetooth Low Energy (Bluetooth Smart)
- 顧客開発アプリケーション向MCU
- GPIO / SPI / I2 C / UART / ADCインターフェース
- 無線によるファームウェアのアップグレード
- ペリフェラルとセントラル機能の同時実行
- シリアル・データ用シリアル・ポート・アプリケーション

NINA-B1 Block diagram



つまりNordic の
nRF52832

mbed os([mbed OS 5.6.2 released](#))の開発環境

オンライン環境

- 簡単
- すぐに始められる。
- インストールの必要なし
- ライブラリのアップデートも簡単。ワンクリック。

オンライン環境

- 面倒
- 手始めに何をすれば？
- インストールツールが多数
- ライブラリのアップデートもマニュアル。

- デバッガーの利用が可能
- 古いバージョンも簡単に試せる。
- オンラインのトラブルに巻き込まれない。

			判定	備考
コンパイル	Mbed	Eclipse/GCC	○	Mbed CLI で可能(富士エレページで紹介中)
		オンラインコンパイラ	×	なぜか、今NINAを選べません？
	Nordic SDK	Eclipse/GCC	○	無料
		Keil	○	32kB以上有料
		ARM-MKR	○	32kB以上有料
書き込み	ドラックアンドドロップ		×	なぜ？
	SEGGER Flash Writer		○	ライター必要
	OTA-DFU (Over the Air 書き込み)		○	Nodric SDK のみ確認 OTA 対応boot loader 必要
	Serial-DFT(Serial boot loader)		○	Nodric SDK のみ確認 DFU 対応boot loader 必要
デバッグ	- 未調査 -			

SEGGER Flash Witer はソフトは無料だが、ハードが高い(6万円)

コマンドライン・ログ

```
C:\¥L¥src¥BLE¥mbed-os-example-ble¥BLE_LED>mbed --version
```

```
1.2.2
```

```
C:\¥L¥src¥BLE¥mbed-os-example-ble¥BLE_LED>mbed target -S
```

Target	mbed OS 2	mbed OS 5	ARM	GCC_ARM	IAR
ARCH_PRO	Supported	Supported	Supported	Supported	Supported
ARM_BEETLE_SOC	Supported	Supported	Supported	Supported	Supported
~ ~ ~ ~ ~	~ ~ ~ ~ ~	~ ~ ~ ~ ~	~ ~ ~ ~ ~	~ ~ ~ ~ ~	~ ~ ~ ~ ~
SARA_NBIOT_EVK	-	Supported	Supported	Supported	Supported
THUNDERBOARD_SENSE	Supported	Supported	Supported	Supported	Supported
TY51822R3	Supported	Supported	Supported	Supported	Supported
UBLOX_C027	Supported	Supported	Supported	Supported	Supported
UBLOX_EVA_NINA	Supported	Supported	Supported	Supported	Supported
UBLOX_EVK_NINA_B1	Supported	Supported	Supported	Supported	Supported
UBLOX_EVK_UDIN_W2	-	Supported	Supported	Supported	Supported
VK_RZ_A1H	Supported	Supported	Supported	Supported	Supported
XDOT_L151CC	-	Supported	Supported	Supported	Supported

```
Supported targets: 74
```

ね、対応してるでしょ

ツール	内容	URL
mbed-cli	mbed-cli , python , gcc , gbd など	https://docs.mbed.com/docs/mbed-os-handbook/en/latest/dev_tools/cli_install/
eclipse	IDE	https://www.eclipse.org/downloads/
	日本語環境	http://mergedoc.osdn.jp/
ライター	SEGGER	https://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack

1. ツールのインストール・ソースの展開

1.1 mbed-cliなどなどのインストール

1.2 サンプルコードの展開

1.3 Eclipseのインストール

1.4 Eclipseの日本語化

1.5 GNU MCU Eclipse のインストール

2. 実際のコンパイル

2.1 Eclipse形式へのExport

2.2 Build環境の設定とBuild

3. 書き込みと動作確認

3.1 Build結果を書き込み

3.2 アプリを用いて動作確認

1.ツールのインストール・ソースの展開

[ココ](#)から、"mbed_installer_v041.exe"をダウンロードしインストールすることで環境の構築を自動にやってくれます。python , GCC , gdb までインストールしてくれるので非常に楽になりました。

インストール開始画面↓

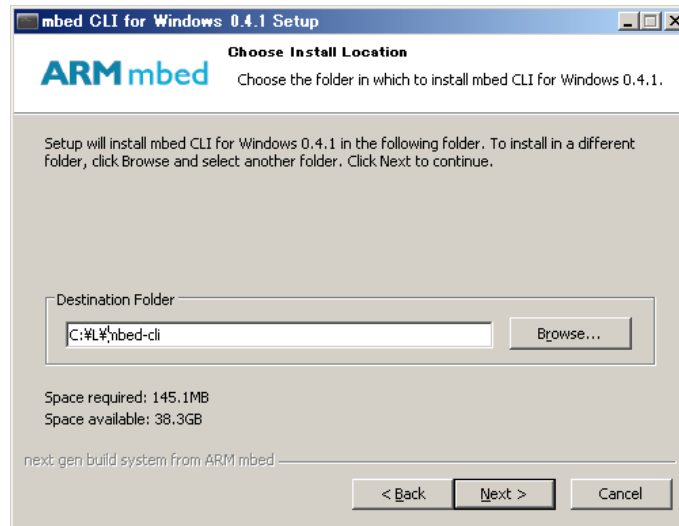


List of components

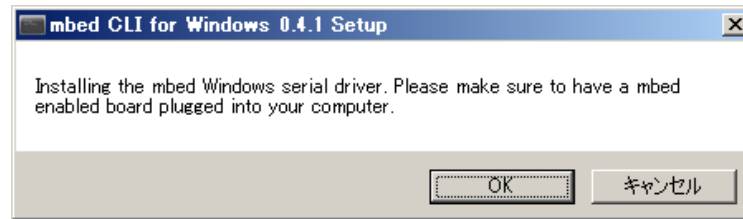
mbed CLI for Windows installs the following components:

- **Python** - mbed CLI is a Python script, so you need Python to use it. Installers installs [version 2.7.13 of Python](#). It is not compatible with Python 3.
- **mbed CLI version 1.1.1** - [mbed CLI](#).
- **Git and Mercurial** - mbed CLI supports both Git and Mercurial repositories. Both Git and Mercurial are being installed. (git and hg) are added to system's PATH.
 - [Git](#) - version 2.12.2.
 - [Mercurial](#) - version 4.1.1.
- **GNU ARM Embedded Toolchain** - [GNU Embedded Toolchain for ARM](#).
- **mbed Windows serial port driver** - [serial port driver](#).

階層を深くしたくないので"C:\¥L"にインストールしています。↓



下記の様に mbed board を接続するように促されますが、cancel でも問題ありませんでした。



念のため mbed-cli のバージョンをアップデートしておきます。

```
> pip uninstall mbed-cli  
> pip install mbed-cli
```

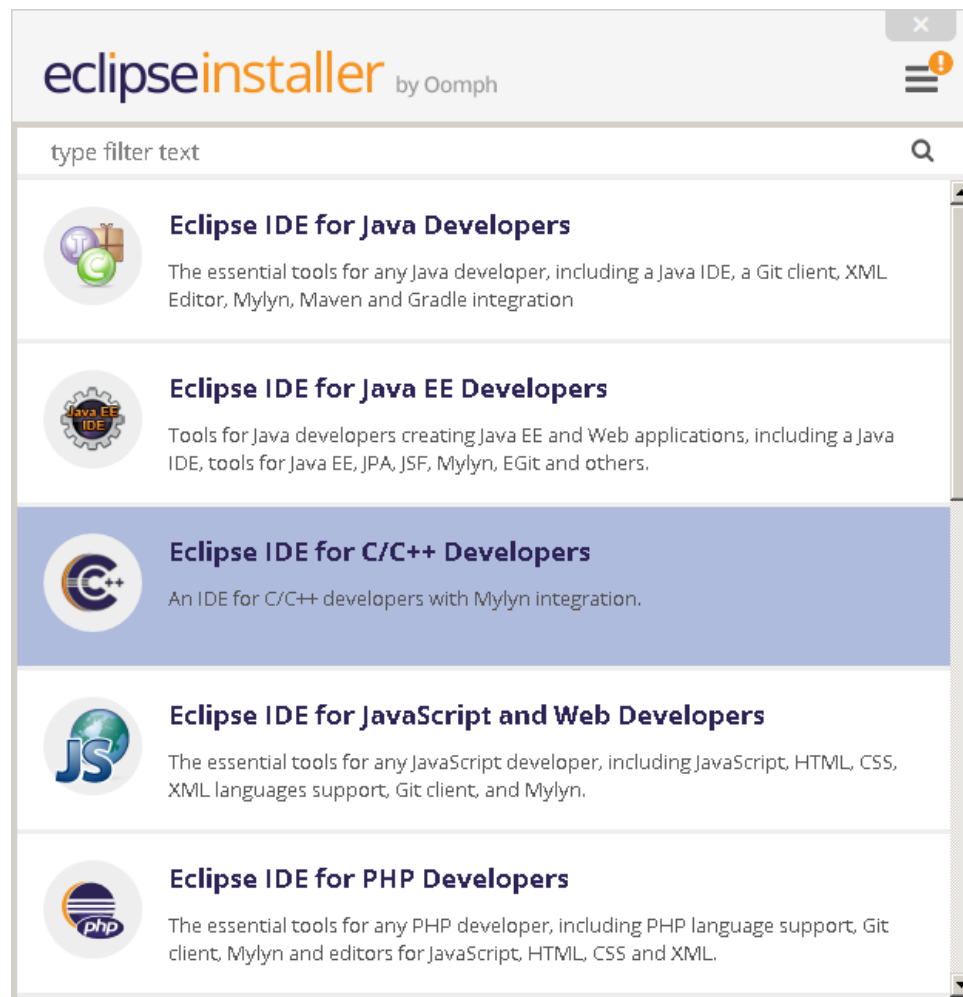

I 1.2 サンプルコードの展開

BLEのサンプルコードを展開します。ここでは c:\src\BLE に展開しています。

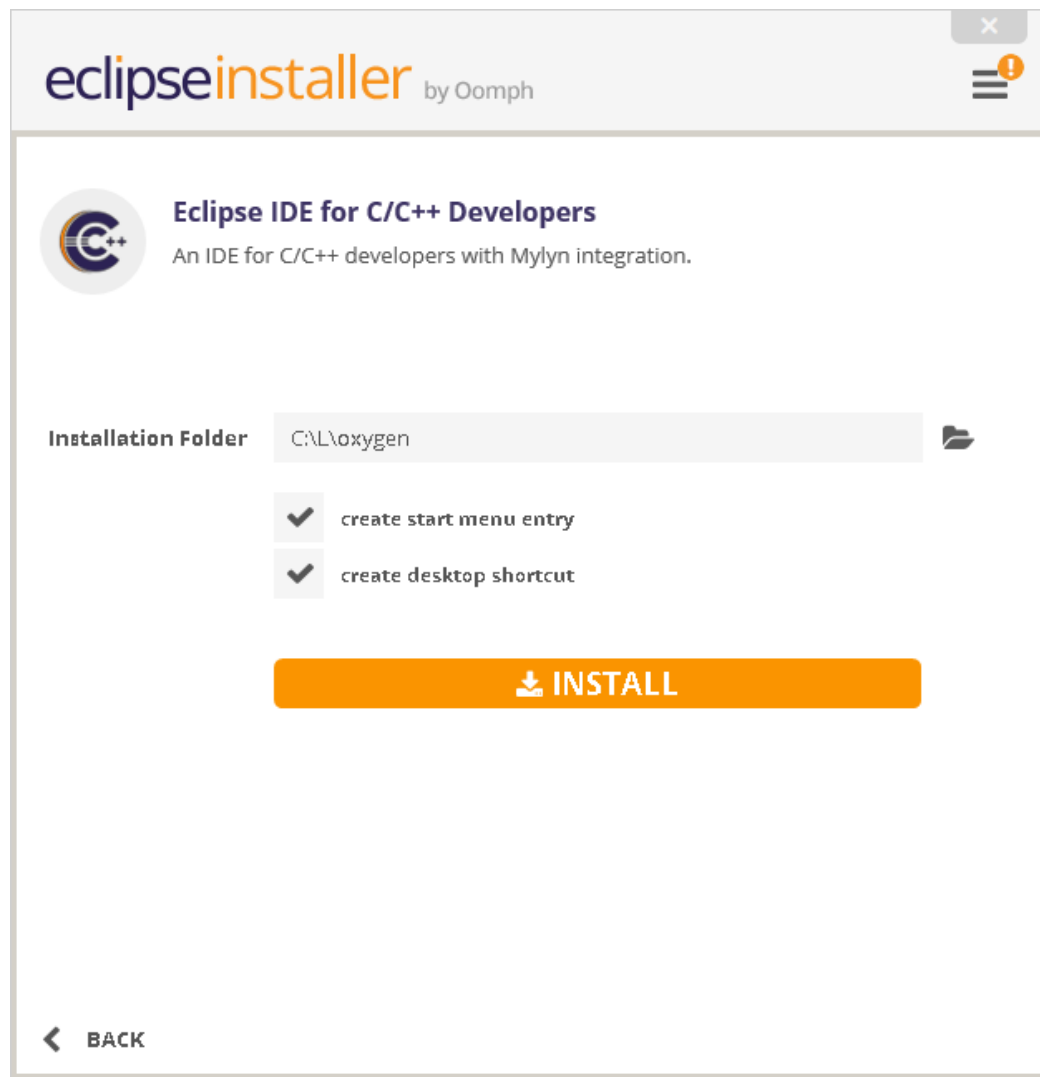
```
> mbed import mbed-os-example-ble
```

I 1.3 Eclipseのインストール

ここでは最新の Eclipse Ver4.7 (Oxygen) をインストールします。[ココ](#)からインストーラーを入手します。OSのBit数に気をつけてください。インストーラを立ち上げ、“Eclipse IDE for C/C++ Developer”を選択し、インストールを開始します。↓



今回はC:/L/oxygen にインストールしました。



このまま 1 回起動しておきます。そして、抜けておきます。

1.4 Eclipse の日本語化

ココから入手したファイルを展開し。



展開したファイル内のREADMEに従い、ファイルをコピーします。
重要な部分だけ下記に引用します。

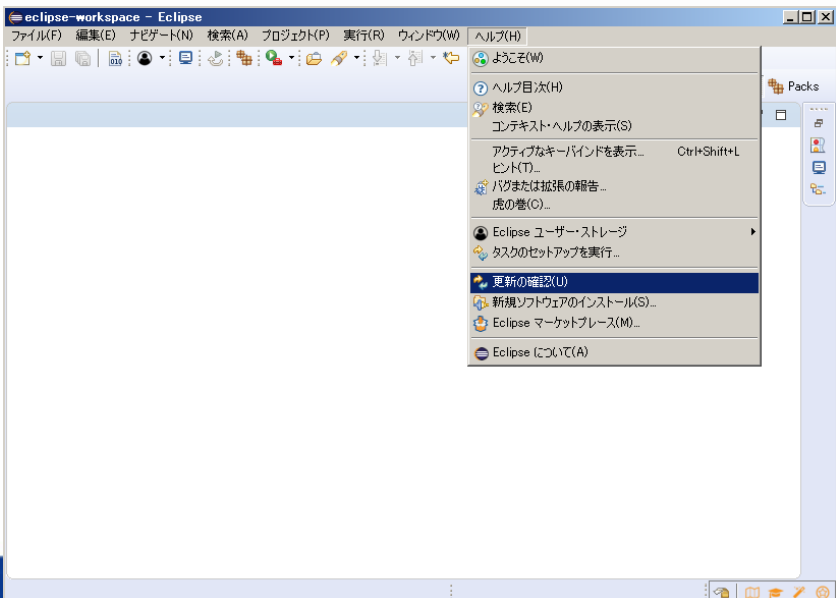
Eclipse へのインストール / アンインストール

1. pleiades_x.x.x.zip を解凍し、plugins、features ディレクトリーを ディレクトリーにコピー。(は Eclipse デフォルトの plugins や features があるディレクトリー)
2. eclipse.ini の最終行に以下の 2 行を追加。Xverify は Eclipse 4.4 以降で必須で、これを指定しないと起動できなかつたり、一部の機能が正常に動作しない可能性があります。
Windows 以外の場合は後述の Eclipse 起動オプション参照してください。
-Xverify:none
-javaagent:plugins/jp.sourceforge.mergedoc.pleiades/pleiades.jar
3. Pleiades スプラッシュ画像を使う場合はeclipse.ini の -showsplash org.eclipse.platform (実際は 2 行) を削除。

再度 Eclipseを立ち上げると日本語化されています。



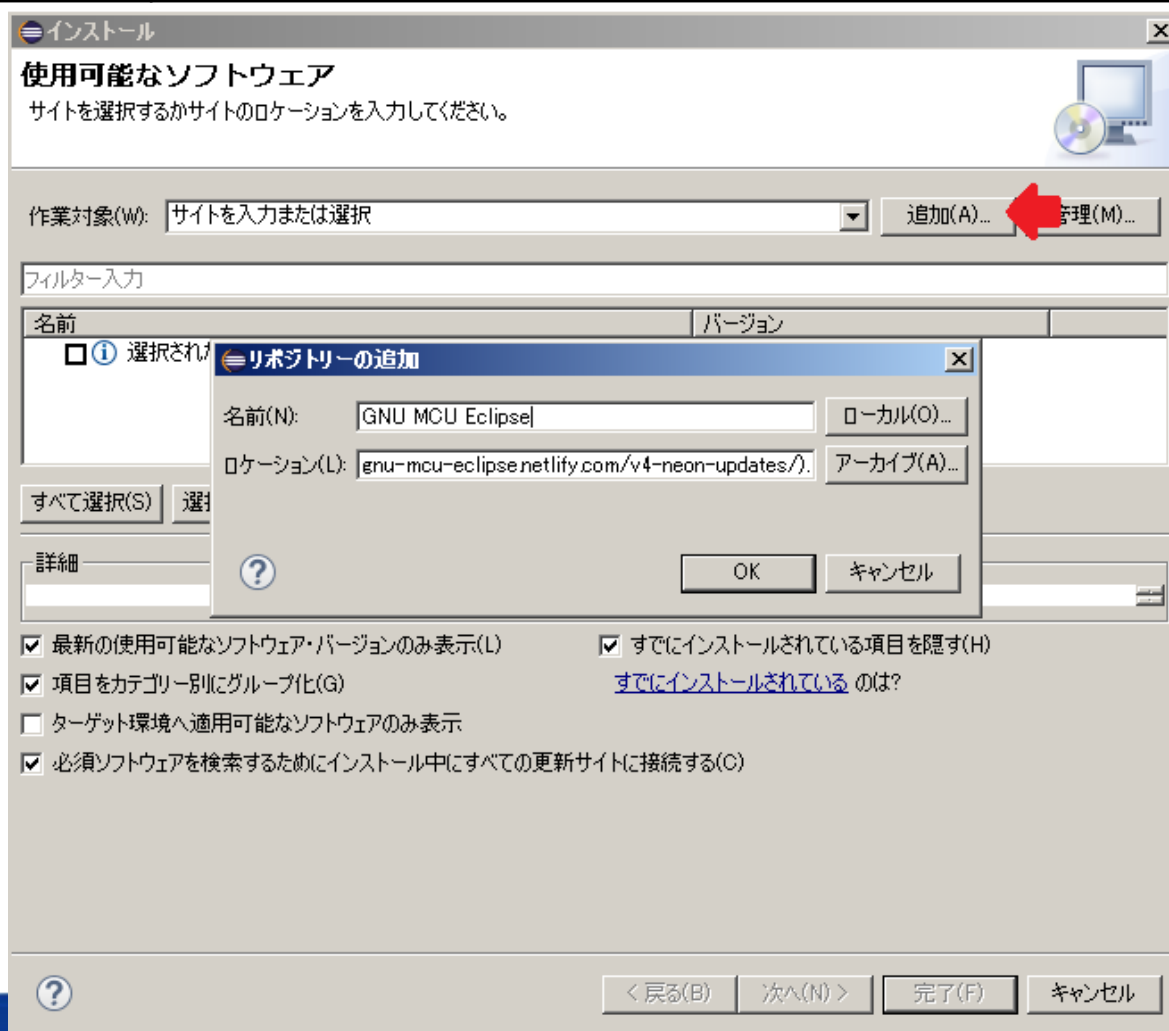
ここで一度HELP→更新の確認から、更新の確認をしておいてください。↓



I 1.5 GNU MCU Eclipse のインストール

Eclipse上でhelp → [新規ソフトウェアのインストール]の追加から、インストールします。

名前(N) :	GNU MCU Eclipse
ロケーション(L):	http://gnu-mcu-eclipse.netlify.com/v4-neon-updates/



I 1.7 SEGGERツールのインストール

[ココ](#)からダウンロードし普通にインストールします。Windows版を普通に選んでください。

The screenshot shows the SEGGER website's download page for the J-Link Software and Documentation Pack. The page includes a navigation menu, a list of features, and a table of available software versions.

J-Link Software and Documentation Pack

- All-in-one debugging solution
- Can be downloaded and used free of charge by any owner of a SEGGER [J-Link](#), [J-Trace](#) or [Flasher](#) model. Not all features of it may be available on all J-Link / J-Trace / Flasher models.
- Updated frequently
- [Release Notes](#)
- [More information](#)

[Click for downloads](#)

	Version	Date	File size	
<input checked="" type="checkbox"/> J-Link Software and Documentation pack for Windows Installing the software will automatically install the J-Link USB drivers and offers to update applications which use the J-Link DLL. Multiple versions of the J-Link software can be installed on the same PC without problems; they will co-exist in different directories.	V6.20e Older versions	[2017-10-06]	26,147 KB	DOWNLOAD

2.実際のコンパイルとデバッガーの起動

I 2.1 Eclipse形式へのExport

まず、ソースコードは取り込んであるのでフォルダを変更して

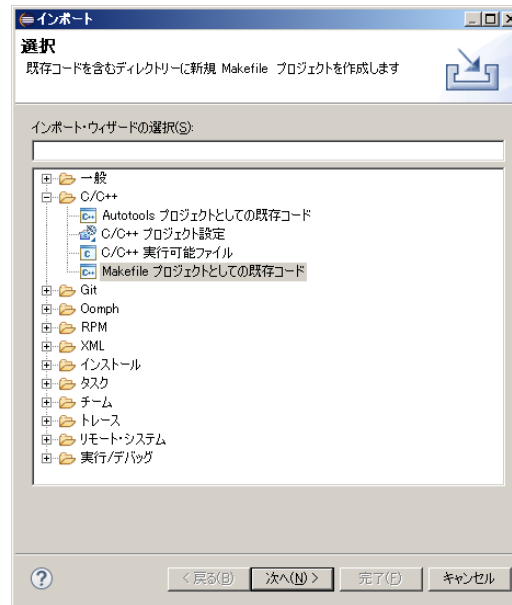
```
> cd mbed-os-example-ble¥BLE_HeartRate
```

下記コマンドで、eclipse で読み込める形式に export します。

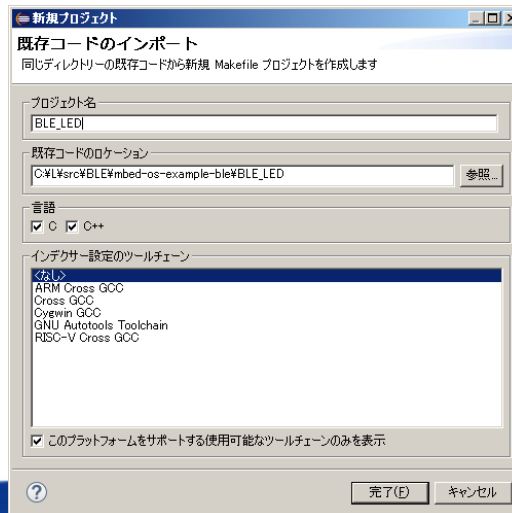
```
> mbed export -i eclipse_gcc_arm -m UBLOX_EVK_NINA_B1 --profile mbed-os/tools/profiles/debug.json
```

Eclipseを立ち上げ

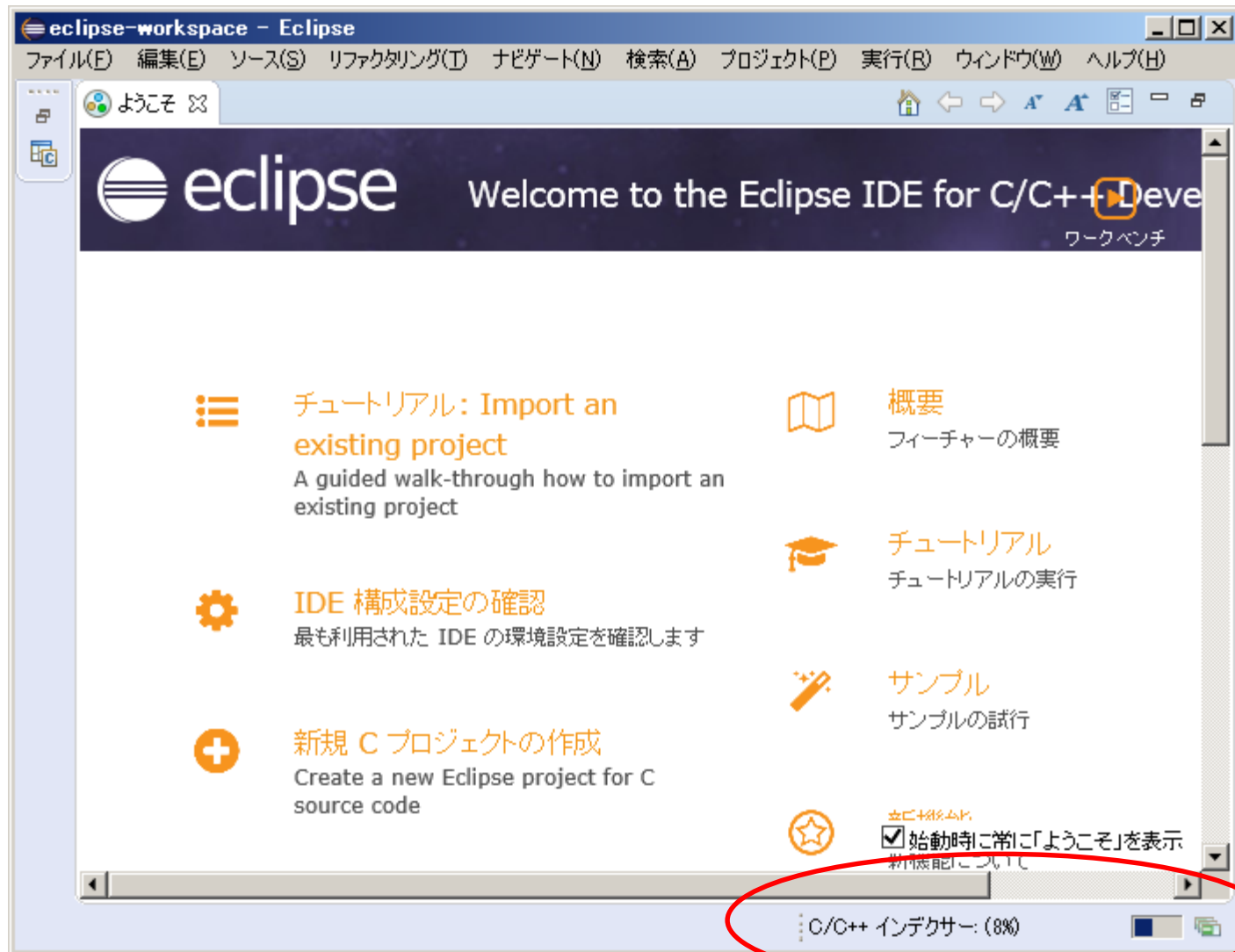
[ファイル]→[インポート]→[C/C++]→[Makefile プロジェクトとしての既存コード]
で"mbed-os-example-blinky"のソースを読み込みます。



下図のように、既存のソースコードロケーションを指定すれば、プロジェクト名は自動的に挿入されます。

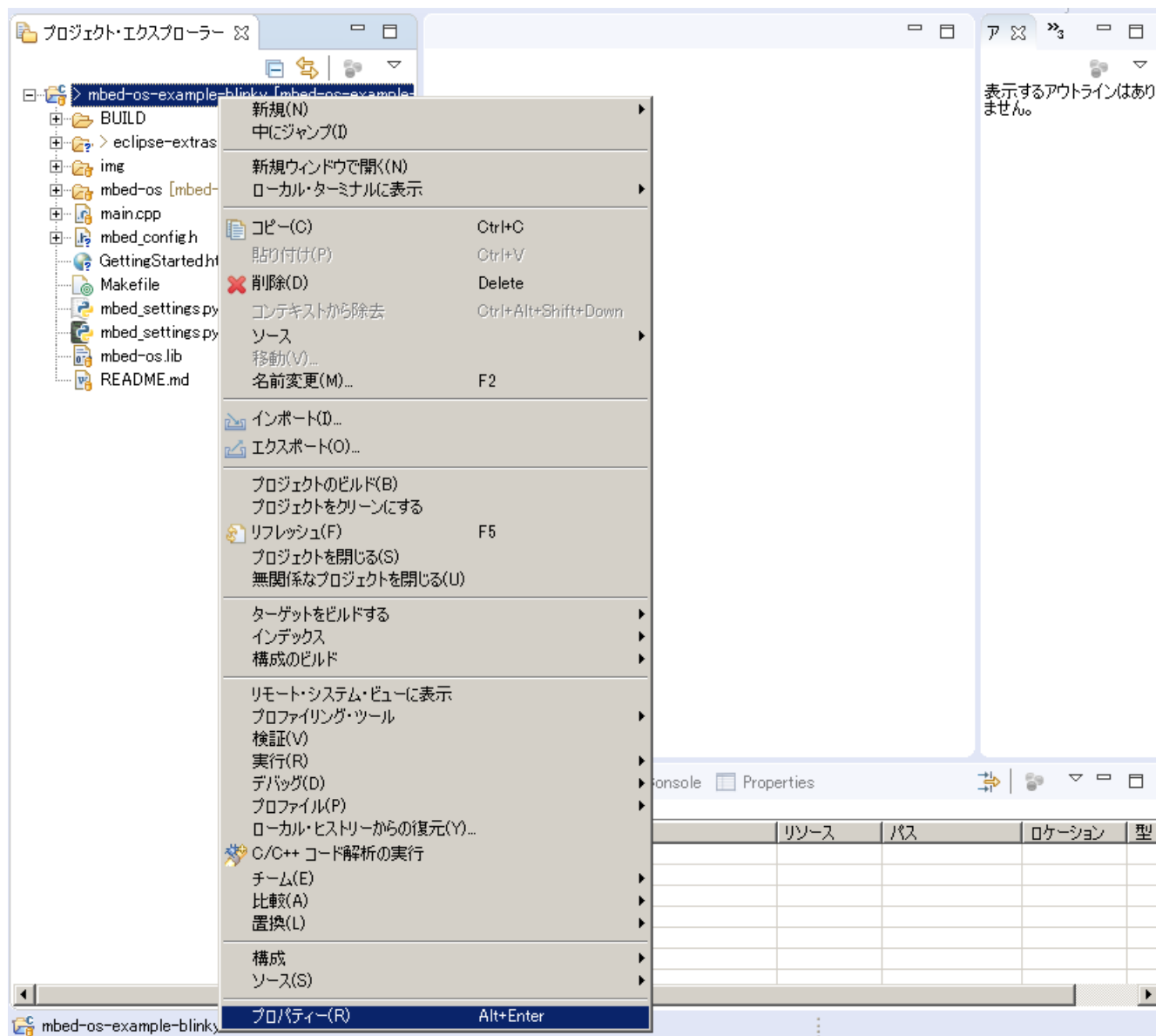


読み込みに時間がかかります。Windowsの右下のインジケーターが100% になるまで気長にお待ち下さい。

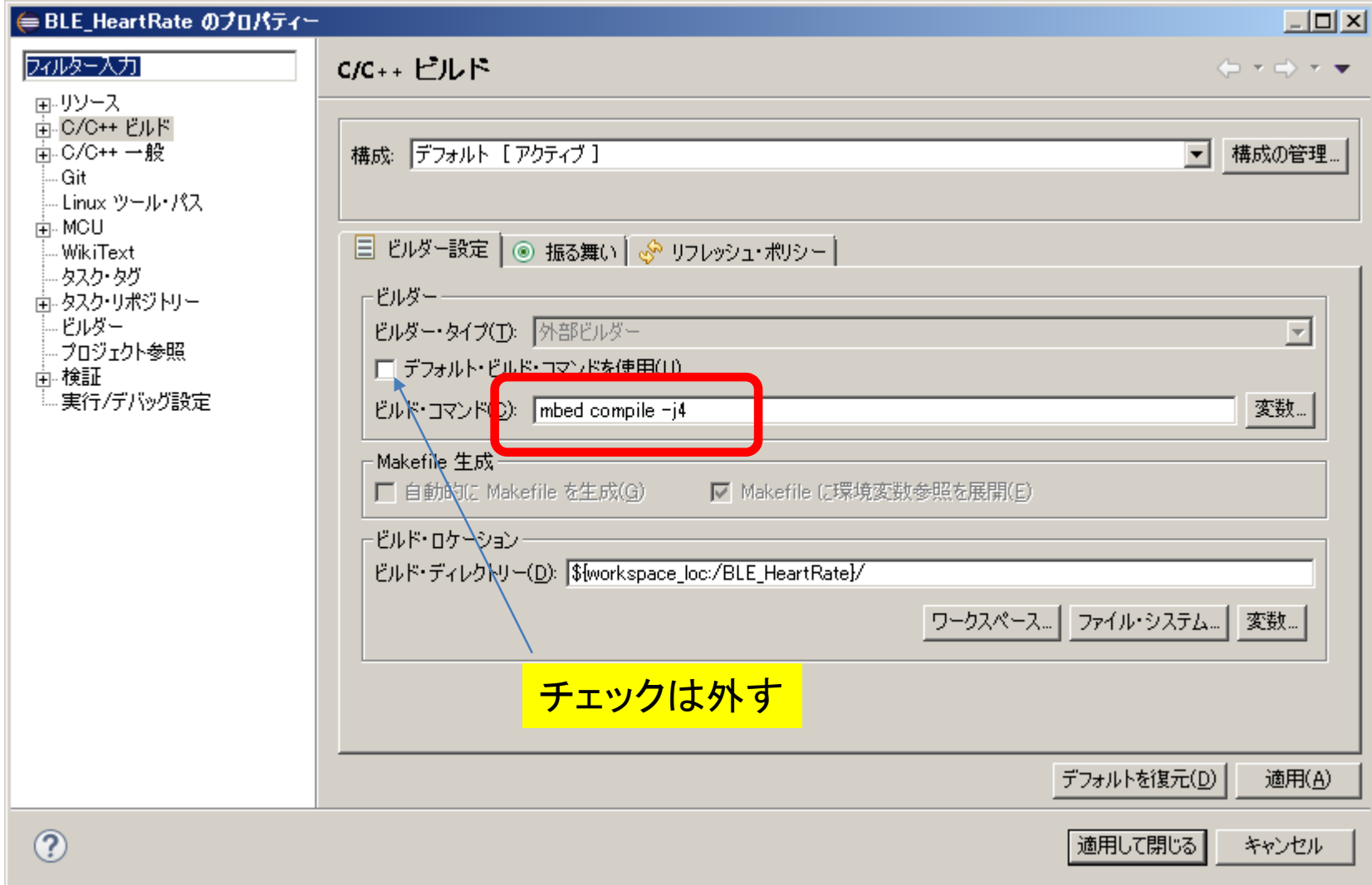


I 2.2 Build環境の設定とBuild

下図の様にEclipseのプロジェクト名をから、プロパティーの設定を選択します。



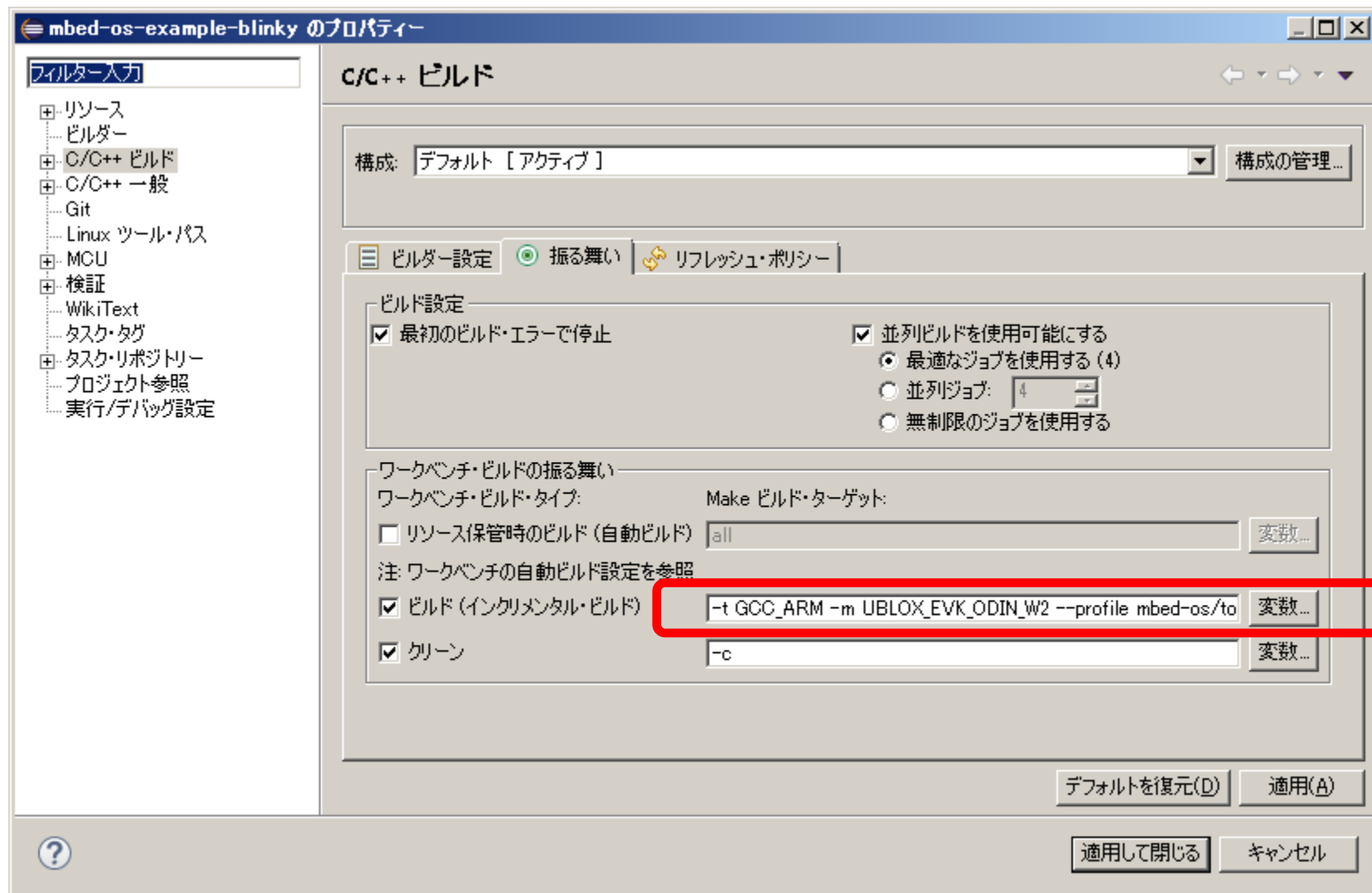
C/C++ ビルドを選択し、[ビルダー設定]中の[ビルド・コマンド(C)]を "mbed compile"に書き換えます。(余談ですが、make の環境が整っていればmake のままでも build は可能です。)



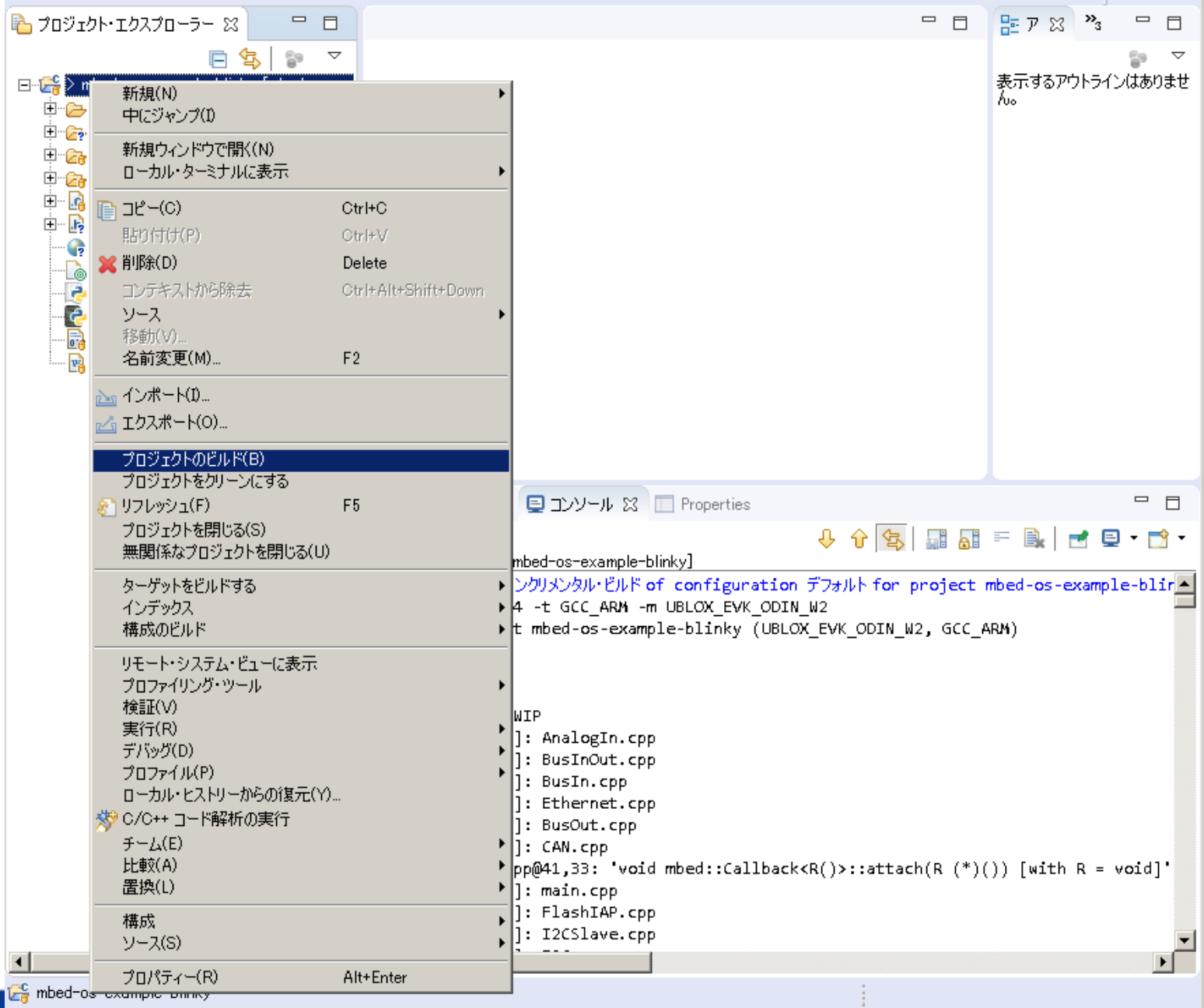
同様に[振る舞い]設定中の[ビルド(インクリメンタル・ビルド)]に

"-t GCC_ARM -m UBLOX_EVK_NINA_B1"

を設定します。並列ビルドを有効にしておくと、PCの複数のコアを用いでBuildするので、実行が早くなります。



下図のように[プロジェクトのビルド]を選択し、build します。



実際の実行ログを下記に示します。

```
mbed compile -j4 -c -t GCC_ARM -m UBLOX_EVK_NINA_B1 --profile mbed-os/tools/profiles/debug.json
Building project BLE_HeartRate (UBLOX_EVK_NINA_B1, GCC_ARM)
Scan: .
Scan: env
Scan: mbed
Scan: FEATURE_BLE
Compile [ 0.5%]: BusIn.cpp
Compile [ 0.9%]: BusOut.cpp
Compile [ 1.4%]: BusInOut.cpp
Compile [ 1.9%]: AnalogIn.cpp
.....
Compile [ 99.1%]: us_ticker.c
Compile [ 99.5%]: test_env.cpp
Compile [100.0%]: main.cpp
Link: BLE_HeartRate
Elf2Bin: BLE_HeartRate
```

Module	.text	.data	.bss
[fill]	80	10	57
[lib]/libc.a	22786	2204	56
[lib]/libgcc.a	3728	0	0
[lib]/libm.a	88	0	0
[lib]/libnosys.a	32	0	0
[lib]/misc	236	12	28
mbed-os/drivers	1742	0	0
mbed-os/events	2784	0	93
mbed-os/features	39621	5	572
mbed-os/hal	1262	0	48
mbed-os/platform	2713	4	265
mbed-os/rtos	17365	180	5984
mbed-os/targets	47236	140	2200
source/main.o	5590	5	101
test_env.o	492	0	0
Subtotals	145755	2560	9404

```
Total Static RAM memory (data + bss): 11964 bytes
Total Flash memory (text + data): 148315 bytes

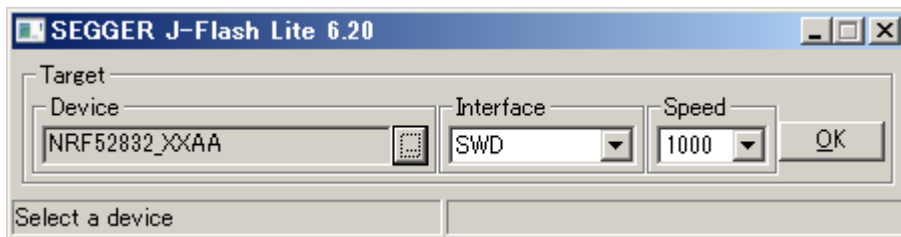
Image: .¥BUILD¥UBLOX_EVK_NINA_B1¥GCC_ARM¥BLE_HeartRate.hex

20:00:30 Build Finished (took 1m:44s.540ms)
```

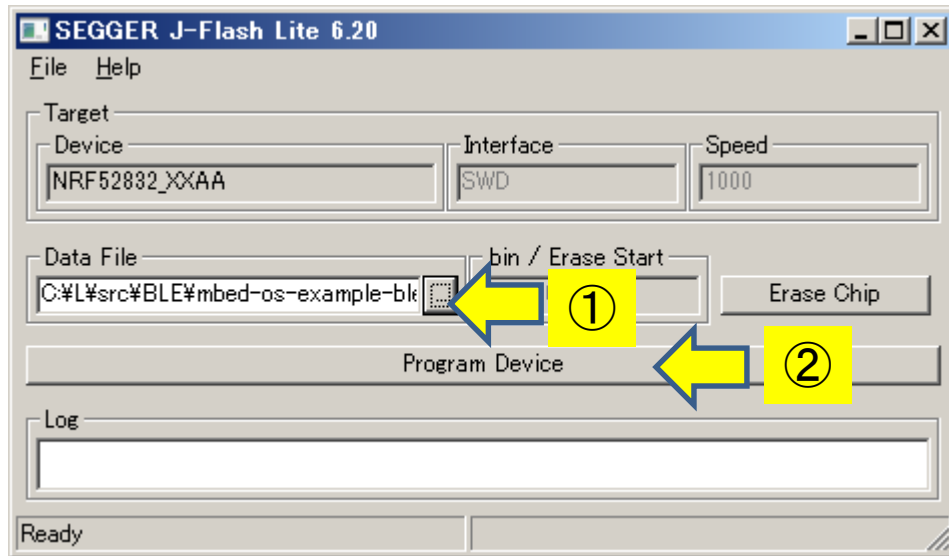
3.書き込みと動作確認

I 3.1 Build結果を書き込み

ここでは SEGGER J-Flash Lite を使用します。
Device に “NRF52832_XXAA” を選んで “OK” を押します。



下記の①ボタンを押して、BuildしたHEXを選択します。場所↓
C:\¥L¥src¥BLE¥mbed-os-example-ble¥BLE_HeartRate¥BUILD¥UBLOX_EVK_NINA_B1¥GCC_ARM



②で書き込みます。

SEGGER J-Link V6.20 - Flash download (260 KB)		
Compare	100.0%	0.033s
Erase	100.0%	0.000s
Program	26.1%	1.238s
Verify	0.0%	
Programming range 0x00011000 - 0x000117FF (2 KB)		1.271s

1 3.2 アプリを用いて動作確認

iOS/Android アプリケーションを用いて、動作を確認します。HeartRateなので心拍数を擬似的に電波で飛ばしています。

ここでは、nRF Tool アプリケーションを用います。

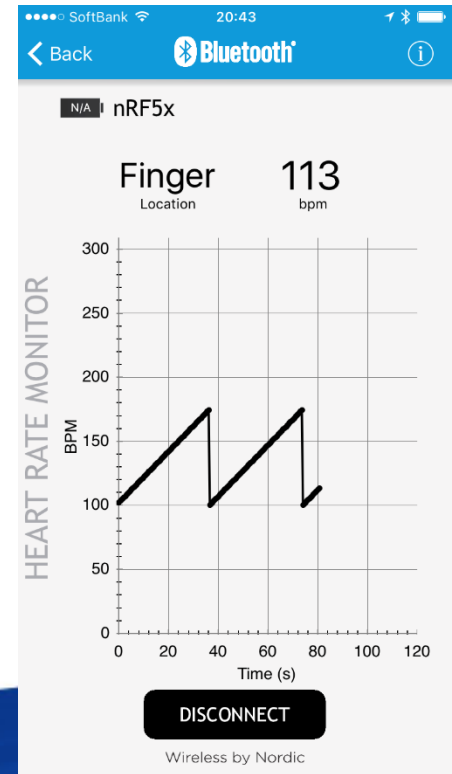
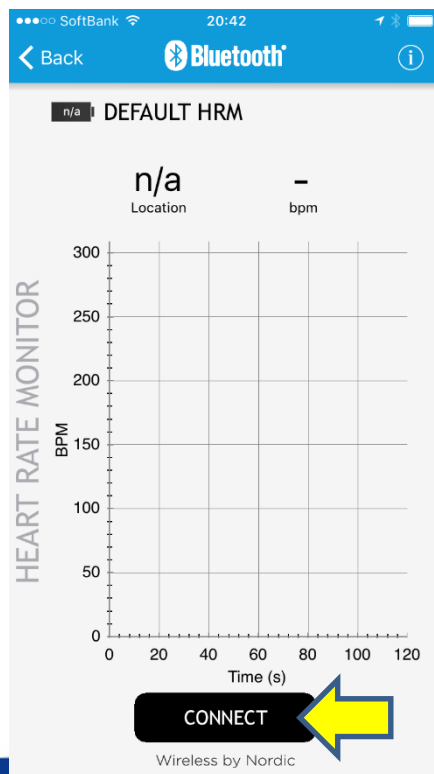
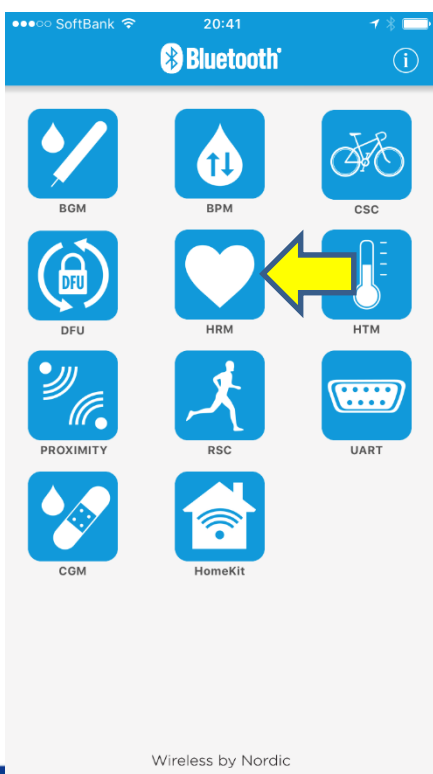


iOS

<https://itunes.apple.com/us/app/nrf-toolbox/id820906058?mt=8>

Android

<https://play.google.com/store/apps/details?id=no.nordicsemi.android.nrftoolbox&hl=ja>



その他


データシートの技適マークをコピーして完成品に貼り付けてください。

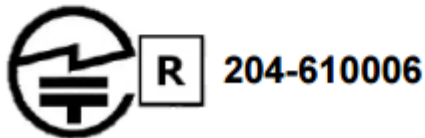
6.5 Japan radio equipment compliance

The NINA-B1 series modules comply with the Japanese Technical Regulation Conformity Certification of Specified Radio Equipment (ordinance of MPT N°. 37, 1981), Article 2, Paragraph 1:

- Item 19 "2.4 GHz band wide band low power data communication system"

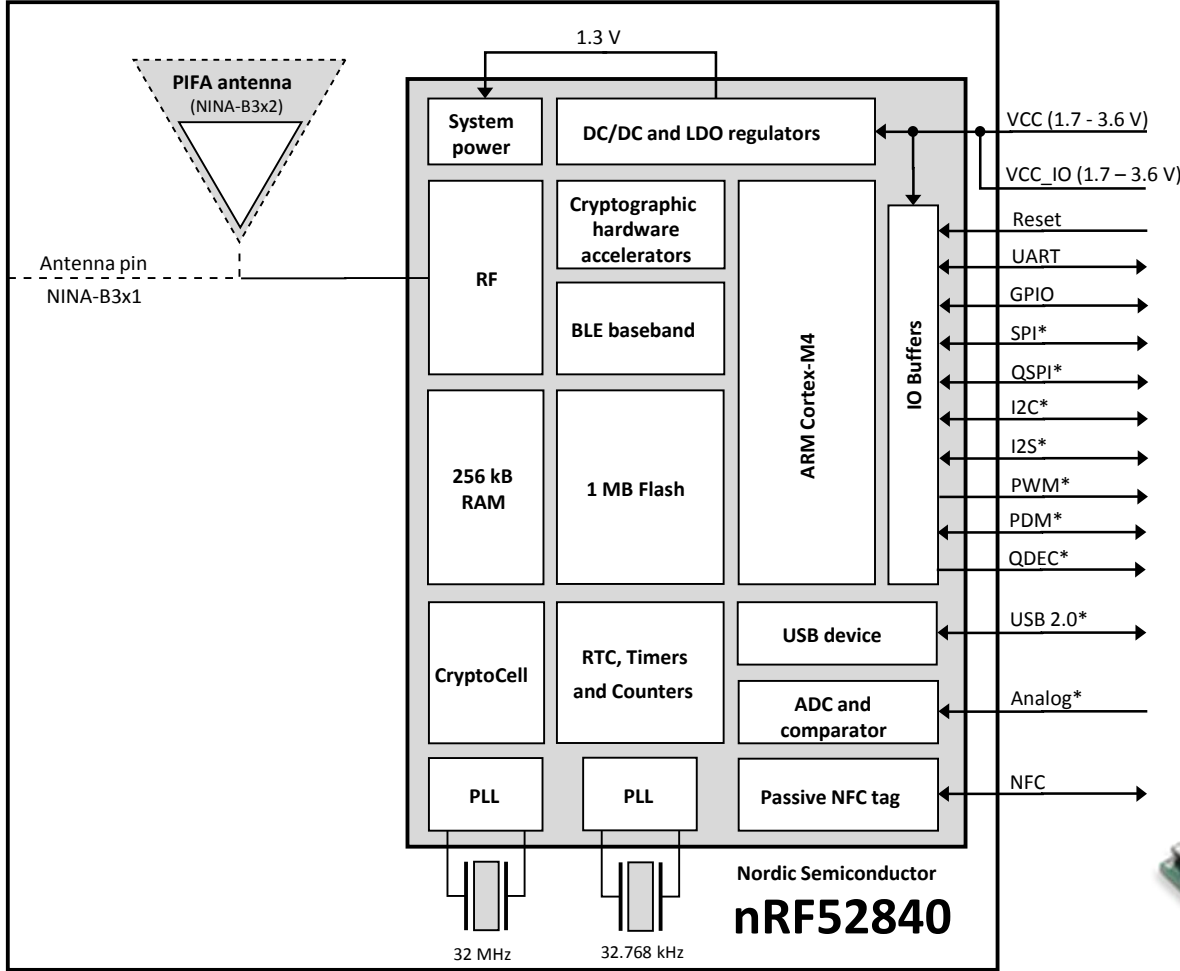
When a product integrated with a NINA-B1 module is placed on the Japanese market, either:

- The ODIN-W2 module be affixed with a label with the Giteki marking below.
In this case is it recommended that the product is marked with "Contains MIC ID:  204-610006".
- The product be affixed with a label with the Giteki marking below. The marking must be visible for inspection.



https://www.u-blox.com/sites/default/files/NINA-B1_DataSheet_%28UBX-15019243%29.pdf

ブロック図



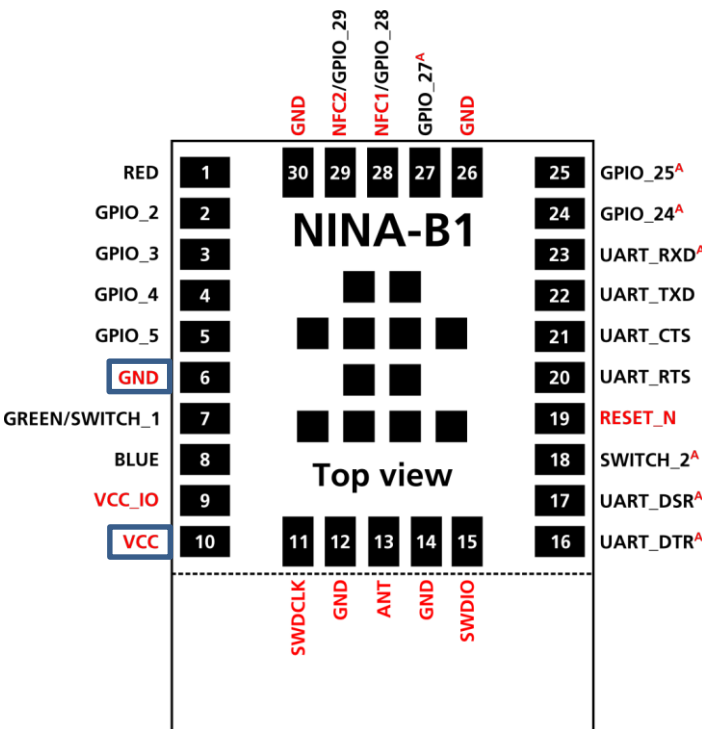
* Only on NINA-B301 and NINA-B302

モジュール



評価ボード



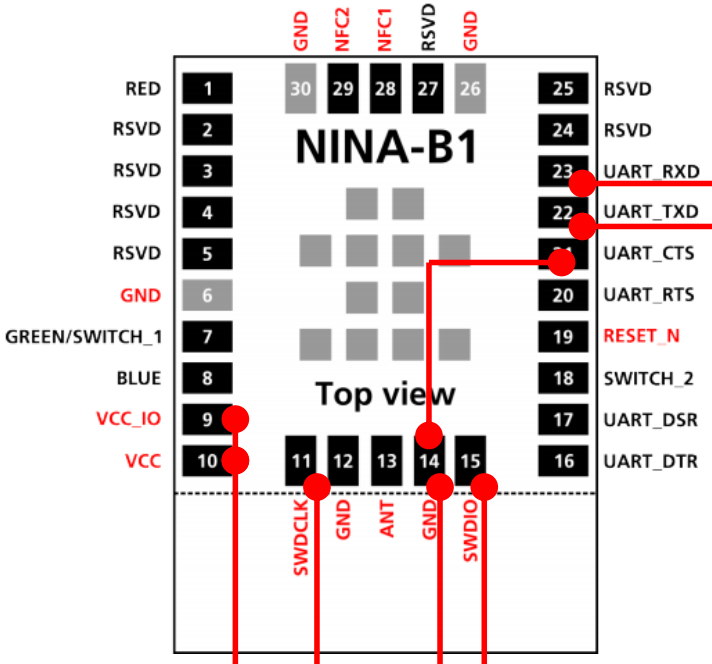


A = Analog function capable pin

No.	Name	I/O	Description	Remarks
1	RED	O	RED systemstatus signal	Active low , should be routed to an RGB LED
2	GPIO_2	I/O	General purpose I/O	
3	GPIO_3	I/O	General purpose I/O	
4	GPIO_4	I/O	General purpose I/O	
5	GPIO_5	I/O	General purpose I/O	
6	GND	-	Ground	
7	GREEN/SWITCH_1	I/O	This signal is multiplexed: GREEN: System status signal. SWITCH_1: Enter bootloader, restore UART serial settings.	Active low . GREEN: Should be routed to an RGB LED. SWITCH_1: See section 2.7.3 for more information.
8	BLUE	O	BLUE systemstatus signal	Active low , should be routed to an RGB LED
9	VCC_IO	I	Module I/O level voltage input	Must be connected to VCC on NINA-B1
10	VCC	I	Module supply voltage input	1.7-3.6 V range
11	RSVD	-	RESERVED pin	Leave unconnected
12	GND	-	Ground	
13	ANT	I/O	Tx/Rx antenna interface	50 Ω nominal characteristic impedance, only used with NINA-B111 modules
14	GND	-	Ground	
15	RSVD	-	RESERVED pin	Leave unconnected
16	UART_DTR	O	UART data terminal ready signal	Used to indicate system status
17	UART_DSR	I	UART data set ready signal	Used to change systemmodes
18	SWITCH_2	I	Connect to peripheral device, enter bootloader and restore UART serial settings.	Active low , see section 2.7.3 for more information.
19	RESET_N	I	External system reset input	Active low
20	UART_RTS	O	UART request to send control signal	Used only when hardware flow control is enabled
21	UART_CTS	I	UART clear to send control signal	Used only when hardware flow control is enabled
22	UART_TXD	O	UART data output	
23	UART_RXD	I	UART data input	Pin is analog capable
24	GPIO_24	I/O	General purpose I/O	Pin is analog capable
25	GPIO_25	I/O	General purpose I/O	Pin is analog capable
26	GND	-	Ground	
27	GPIO_27	I/O	Analog function enabled GPIO	Pin is analog capable
28	NFC1/GPIO_28	I/O	NFC pin 1 (default)	May be used as a GPIO
29	NFC2/GPIO_29	I/O	NFC pin 2 (default)	May be used as a GPIO
30	GND	-	Ground	

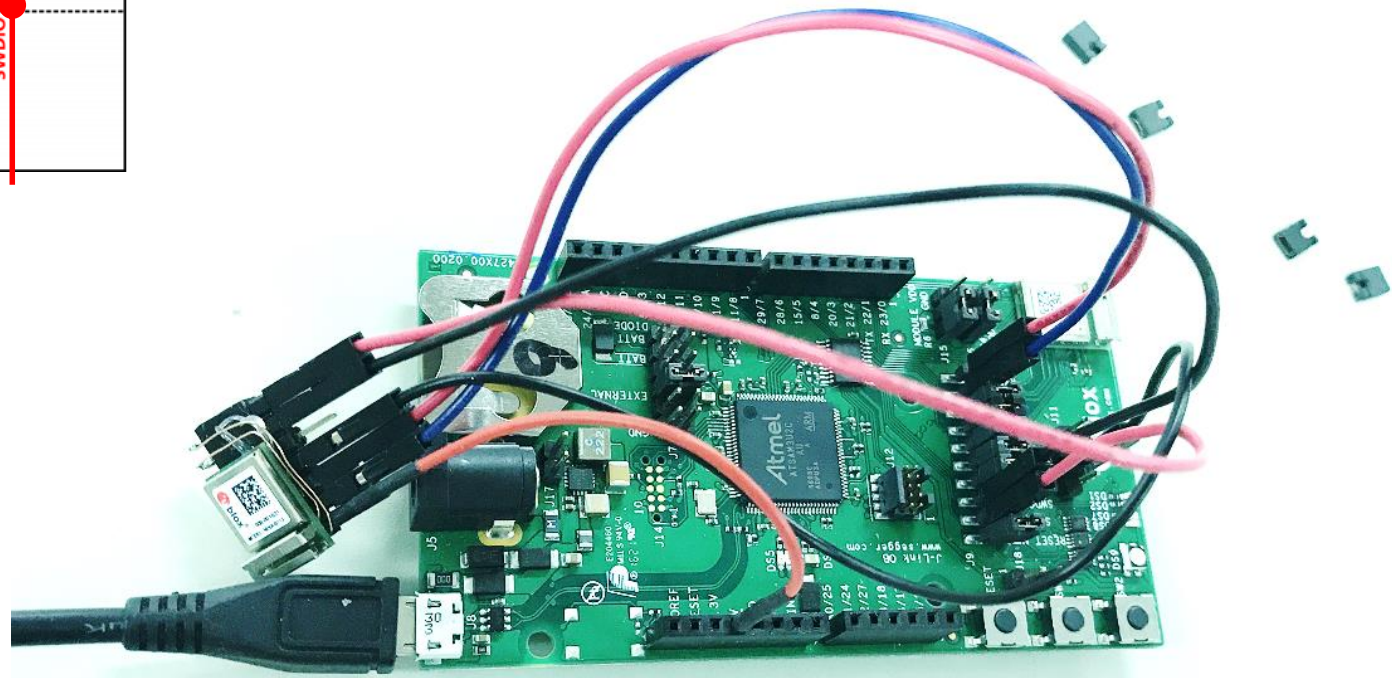
Table 8: NINA-B1 series NINA-B11x firmware pin-out

EVK-NINA-B1をご購入してください。(SEGGER正規品は高いです。)



EVKからこんな感じに配線を飛ばして書き込みます。

NINA-B1の最低限の配線はこうです。



I モジュールにはどう書き込むの？【2】

念のためEVKでの接続を詳しく。

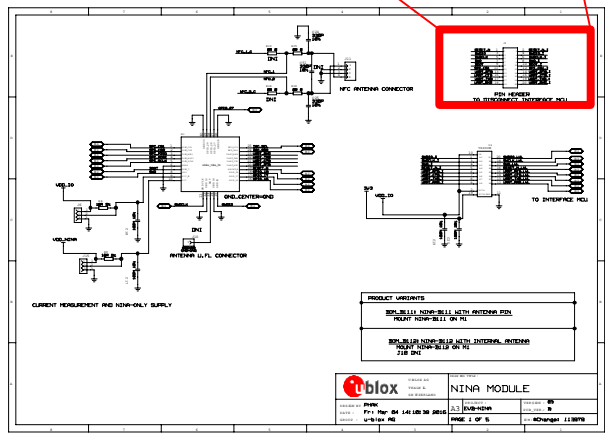
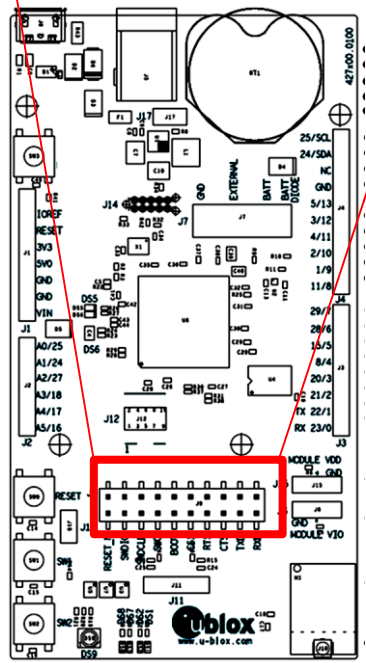
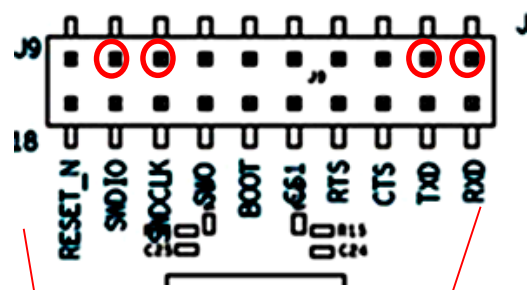
■ 回路図ではココ

RESET_N	1	2	RESET_N_I
SWDIO	3	4	SWDIO_I
SWCLK	5	6	SWCLK_I
SWD	7	8	SWD_I
BOOT	9	10	BOOT_I
SPI_CS1	11	12	SPI_CS1_I
UART_RTS	13	14	UART_RTS_I
UART_CTS	15	16	UART_CTS_I
UART_TXD	17	18	UART_TXD_I
UART_RXD	19	20	UART_RXD_I

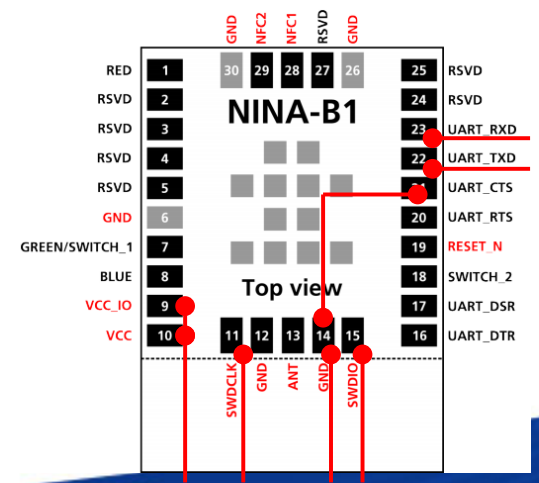
PIN HEADER
TO DISCONNECT INTERFACE MCU

■ 基板イメージではココ

ジャンパーは4個抜く



- から
- SWDIO
 - SWCLK
 - TXD
 - RXD
- を引っ張り出して
NINA-B1に繋いでください。
- VDD(3V)
 - GND
- も忘れずに



※名前そのままでもいいです。クロス済の端子名です。

I モジュールにはどう書き込むの? 【3】

念のためnRF52 DKでの接続を詳しく。

http://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52%2Fdita%2Fnr52%2Fdevelopment%2Fpreview_dev_kit%2Fhw_debug_out.html

に書いてありますが、下記に翻訳を貼っておきます。

デバッグ出力

nRF52 Preview Development Kitボードは、外部ボードのプログラミングとデバッグをサポートしています。外部ボードをデバッグするには、デバッグ出力コネクタ (P19) に10ピンケーブルで接続します。

nRF52 Preview DK v1.1.x

P19, Debug out

P20

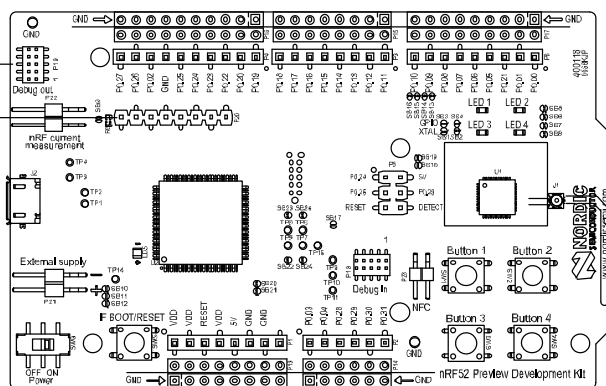


Figure 1. Debug output connector

外部ボードに電源が供給されると、インタフェースMCUはボードの電源電圧を検出し、オンボード nRF52832の代わりに外部ボード上のターゲットチップをプログラム/デバッグします。

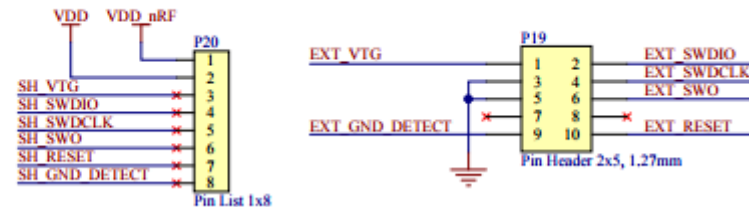
重要:外部デバッグ/プログラミングによってサポートされる電圧は3.0Vです。

また、P20をデバッグアウト接続として使用してシールド実装ターゲットをプログラムすることもできます。デバッグ出力ヘッダ (P19) の場合、インタフェースMCUは実装シールドの電源電圧を検出し、シールドターゲットをプログラム/デバッグします。

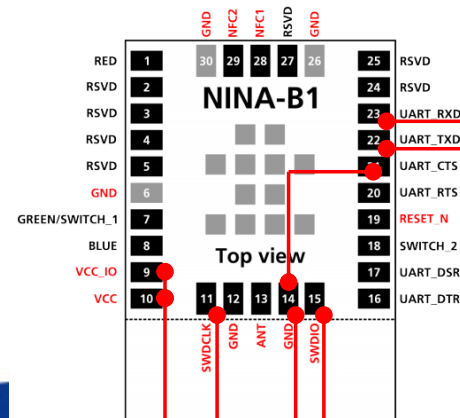
インタフェースMCUがP19とP20の両方でターゲット電源を検出すると、P19に接続されているターゲットをデフォルトでプログラム/デバッグします。

↓回路図より。

Shield Debug and Current measurement Connector Debug OUT Connector



つまり、P19 or P20 から SWDIO/SWCLKを引っ張り出せば良いようです。(持っていないので実験できませんが) それをNINA-B1 の相当端子に繋いでください。



- **mbedページでのEVK-NINA-B1紹介**
<https://developer.mbed.org/platforms/u-blox-EVK-NINA-B1/>
- **NINA-B1 開発環境構築(Eclipse を用いて mbed-os-example-ble の実行まで)**
<https://www.fujiele.co.jp/u-blox-open-contents/>
- **EVK-NINA-B1 ユーザーズガイド**
https://www.u-blox.com/sites/default/files/EVK-ODIN-W2_UserGuide_%28UBX-16007132%29.pdf
- **u-blox NINA-B1 紹介ページ**
<https://www.u-blox.com/ja/product/nina-b1-series>
- **富士エレクトロニクス u-blox社製品 紹介ページ**
<http://www.fujiele.co.jp/semiconductor/ublox/>
- **mbed オフラインの開発環境**
<https://os.mbed.com/users/MACRUM/notebook/mbed-offline-development/>

thank you for your attention

Best Choice for Customers.

いつもお客様のそばに、ベストチョイスをお届けする為に



confidential