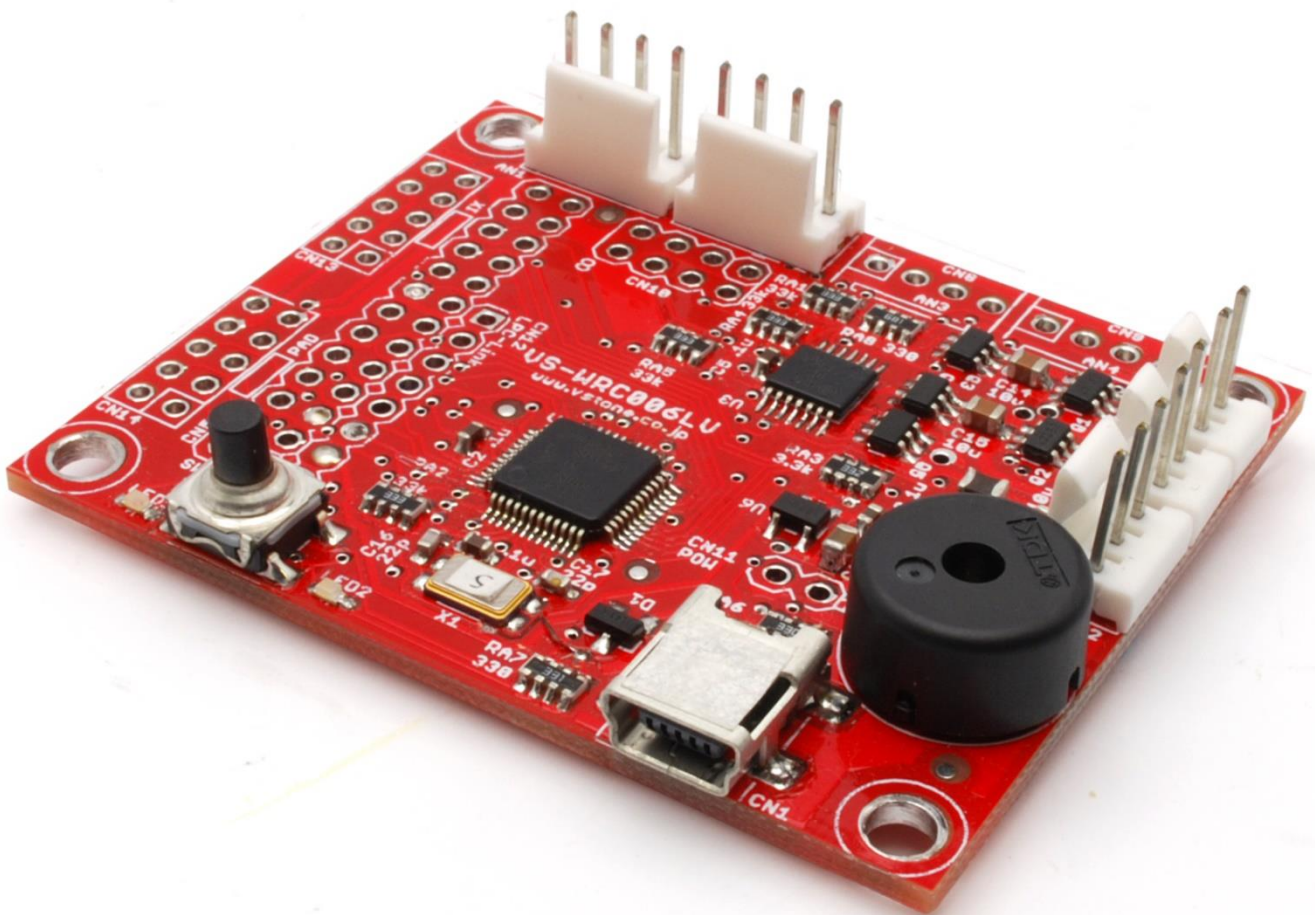


ARM CortexM3 搭載学習用 CPU ボード

VS-WRC103LV

取扱説明書



ヴァイストーン株式会社
(2017.9.22)

目次

1 はじめに	3
2 仕様	3
3 ご注意	4
4 本体外観	4
5 各部詳細	5
5-1 この章の見方	5
5-2 通信コネクタ (CN1)	6
5-3 DC モータ出力 (CN3, 4)	6
5-4 LED	7
5-5 ブザー出力	8
5-6 スイッチ入力 (SW1)	8
5-7 アナログセンサ入力 (CN6, 7, 8, 9)	9
5-8 I2C (CN13)	10
5-9 VS-C1 接続用コネクタ (CN14)	10
5-10 拡張 IO (CN5, CN10)	11
5-11 LPC-Link 接続ポート (CN12)	12
6 開発環境の導入とサンプルプログラムの実行	13
6-1 開発環境 LPCXpresso の入手	13
6-2 LPCXpresso のインストール	18
6-3 LPCXpresso の起動と認証	21
6-4 サンプルプロジェクトのインポートとビルド	26
(1)ワークスペースフォルダの変更	26
(2)サンプルプロジェクトのインポート	29
(3)サンプルプロジェクトのビルド	31
6-5 VS-WRC103LV へのプログラムの書き込み	33
6-6 LED 点滅プログラムについて	36
6-7 エラーの修正	37

1 はじめに

本説明書は、ARM マイコンを搭載した、ロボット制御学習用のマイコンボード「VS-WRC103LV」の使用方法について解説したマニュアルです。ご使用になる前に、かならず本説明書をよくお読みいただき、安全にお使いください。

2 仕様

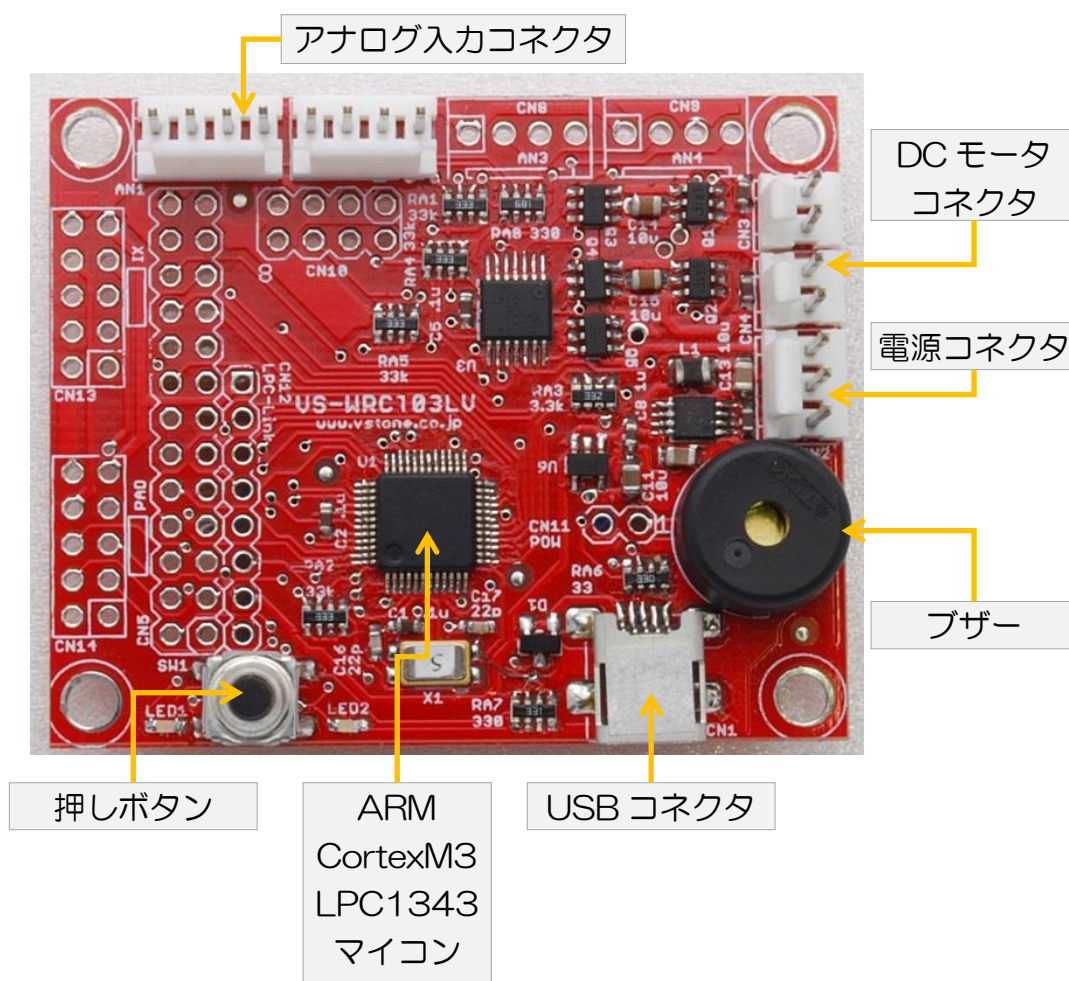
サイズ	W56×D46 (mm)
重量	13g
CPU	ARM CortexM3 LPC1343 (NXP セミコンダクターズ製)
電源	DC 2～4V アルカリ乾電池 2 本 または ニッケル水素充電電池 2～3 本
出力	DC モータ×2 (連続電流 2A まで) LED×2 (オレンジ・緑) 圧電ブザー×1
入力	アナログセンサ入力×2 (最大 4、別途コネクタの取付が必要)
インタフェース	USB (HID 準拠) × 1、シリアルポート (3.3v レベル) × 1 I2C×1、拡張 IO ポート、無線コントローラ接続ポート
オプション	IO 拡張ボード VS-WRC004LV 無線コントローラ VS-C1、ローバー用無線操縦セット I2C 拡張ボード (VS-IX001、VS-IX008 など) ロータリーエンコーダ拡張セット モータアンプ VS-WRC006 8 連赤外線センサ VS-IX010

3 ご注意

本製品を取り扱う際には、注意事項に従い正しくお使いください。

- VS-WRC103LV（以降 本ボード）に強い衝撃を与えないでください。
- 本ボードを水に濡らしたり、湿気やほこりの多い場所で使用したりしないでください。
ショートなどによる故障が発生する恐れがあります。
- 本ボードから煙が発生した場合、すぐに電源をお切りください。
- 本ボードを幼児の近くで使用したり、幼児の手の届くところに保管したりしないでください。
- 動作中、基板上の素子が高温になることがありますので、絶対に触れないでください。
- 基板上の端子（金属部分）に触れると静電気により故障する恐れがあります。かならず基板の縁を触るようにしてください。
- 基板上の端子同士が金属などでショートすると、過電流により故障する可能性があります。
- CN8、CN9 のコネクタは付属しておりません。（拡張オプションとして販売しております。）
- DC モータ出力は FA-130RA タイプモータ（ビュート ローバー用）を使用するためのポートです。
それ以外のモータも使用可能ですが、モータの種類、使用方法によっては基板の破損につながる可能性があります。

4 本体外観



5 各部詳細

ここでは、基板上の各機能について解説します。各部品の ARM マイコンとの接続の詳細につきましては回路図を参照してください。

5-1 この章の見方

○ ピン配置

各入出力端子のピン配置を示します。ARM マイコンとつながっている場合、ポート番号を記載します。ポートと ARM マイコンとの間は直接接続されているだけではなく、なにかしらの回路がある場合がありますので、詳細は回路図をご覧ください。

○ ブロック

ビュートビルダー2 を使用する際に、その機能を使用するためのブロックです。そのブロックをシーケンスエリアに置くことで、簡単に機能を使用することができます。

ビュートビルダー2 の使用方法の詳細は、ビュートローダー付属の CD-ROM 内、またはサポートページにあります「ビュートビルダー2 取扱説明書」をご覧ください。

○ C 言語関数

弊社 Web サイトで公開している C 言語サンプルソースの中で定義されている関数になります。その関数を使用することで、数行のプログラムで動作させることが可能になります。

無償の C 言語開発環境 LPCXpresso、C 言語のサンプルソースの使用方法については本マニュアルの 6 章をご覧ください。

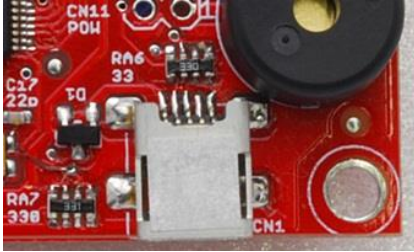
○ ポート

各機能を構成する部品が ARM マイコンのどの IO ポートに接続されているかを示します。C 言語で 1 からプログラムを作成する場合に使用します。

5-2 通信コネクタ (CN1)

通信コネクタは、mini-B タイプの USB 端子となっており、市販の USB mini-B ケーブルなどでパーソナルコンピュータ（以降 PC）と接続できます。

使用している ARM マイコンには USB 機能が搭載されているため、USB で PC と接続すると自動的にドライバがインストールされ、使用できる状態となります。



○ ポート

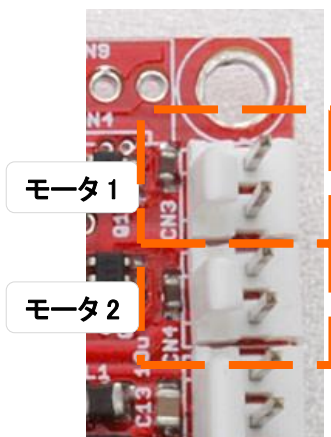
ARM マイコンの USB_DP、USB_DM が接続されています。

本ボードと PC 間の HID での通信仕様、サンプルプログラム、シリアル通信をおこなう方法などは、サポートページをご覧ください。

5-3 DC モータ出力 (CN3, 4)

本ボードには 2ch（拡張ボード VS-WRC004LV を使用すると +4ch）の DC モータ出力を搭載しています。

このモータ出力はビュート ローバー用モータ（FA-130RA タイプモータ）を使用するためのポートになります。それ以外のモータでも、連続電流 2A までのモータであれば接続可能です。（この場合、モータの種類、使用方法によっては基板が破損する可能性がありますので、十分ご注意ください。）



○ ブロック



移動アクションブロック



モータ制御ブロック

○ C 言語関数

`void Mtr_Run_lv ( short mt1 , short mt2 , short mt3 , short mt4 , short mt5 , short mt6 );`

設定した速度でモータを駆動。呼び出した後はその状態を保持し、制御値に 0 を与えるまで停止しない。

引数： モータの制御値

0 : フリー（ブレーキ）

0x8000 : フリー（ブレーキ）

時計回り最大値 : 0x7FFF （32767）

反時計回り最大値 : 0x8001 （-32767）

戻り値： 無し

○ ポート

VS-WRC103LV

- CN3 (M1) PIO2_0 、 PIO2_1 (方向出力) PIO0_8 (PWM、16 bit タイマーBOMR0)
- CN4 (M2) PIO2_2 、 PIO2_3 (方向出力) PIO1_9 (PWM、16 bit タイマーB1MR0)

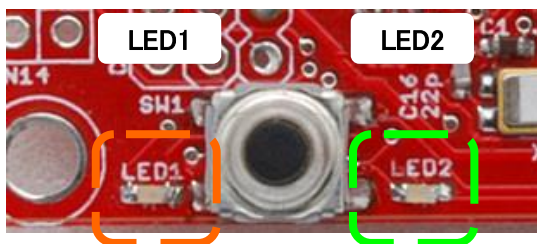
VS-WRC004LV (オプション)

- CN3 (M3) PIO2_4 、 PIO2_5 (方向出力) PIO1_6 (PWM、32 bit タイマーBOMR0)
- CN4 (M4) PIO2_6 、 PIO2_7 (方向出力) PIO1_7 (PWM、32bit タイマーBOMR1)
- CN5 (M5) PIO2_8 、 PIO2_9 (方向出力) PIO0_9 (PWM、16 bit タイマーBOMR1)
- CN6 (M6) PIO2_10、PIO2_11 (方向出力) PIO0_10 (PWM、16 bit タイマーBOMR2)

5-4 LED

本ボードには、オレンジ、緑のLEDが1個ずつ搭載されています。

それぞれのLEDは接続されているポートをLOWレベルにすることで点灯します。



○ ブロック



LED ブロック(オレンジ)



LED ブロック (緑)

○ C 言語関数

```
void LED( BYTE LedOn );
```

CPU ボード上の2つのLEDを制御する関数

- 引数：
- 0: 消灯
 - 1: 緑 点灯
 - 2: オレンジ 点灯
 - 3: 両方 点灯

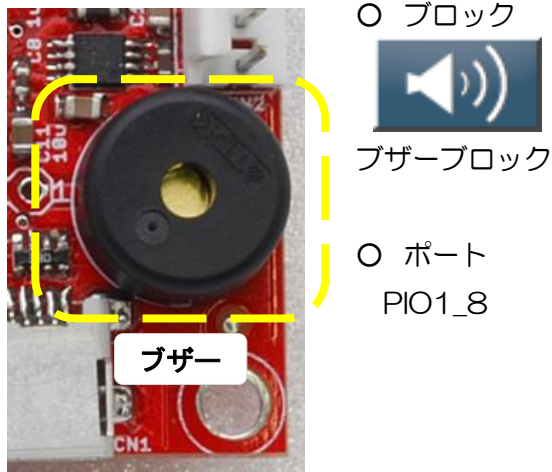
戻り値： 無し

○ ポート

- LED1 (オレンジ) : PIO0_3
- LED2 (緑) : PIO0_7

5-5 ブザー出力

本ボードには、圧電ブザーが搭載されており、単音を出力することが可能です。ブザーは 32bit タイマー1 の割り込みにより PIO ポートをスイッチングして駆動しています。



○ C 言語関数

```
void BuzzerSet( BYTE pitch , BYTE vol );
```

音程、ボリュームの設定。ブザーを鳴らす際の音程とボリュームを設定

引数： pitch：音程の設定（0～255、値が大きいほど低い音）
 vol：ボリュームの設定（0～128）

戻り値： 無し

```
void BuzzerStart();
```

開始。ブザーを鳴らし始める。BuzzerStop()関数を呼びまでなり続ける。

引数： 無し 戻り値： 無し

```
void BuzzerStop();
```

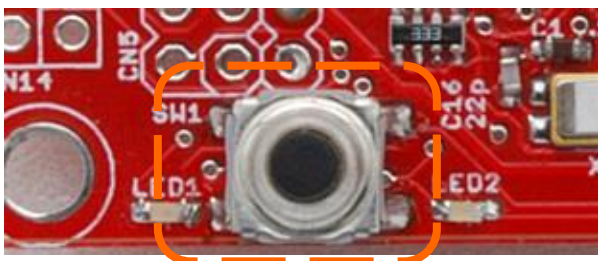
停止。ブザーを止める。

引数： 無し 戻り値： 無し

5-6 スイッチ入力 （SW1）

本ボードには、1つの押しボタンが搭載されています。ビュートビルダー2を使用する場合、このボタンを押すことで、書き込んだシーケンスをスタートさせます。シーケンスを再生中は、スイッチ入力として使用可能です。

○ C 言語関数



```
BYTE getSW();
```

ボタン状態取得。CPU ボード上の押しボタンの状態を取得する

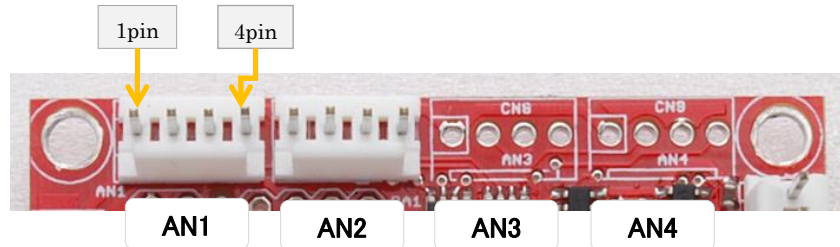
戻り値： 0 : off
 1 : on

○ポート PIO0_1

5-7 アナログセンサ入力 (CN6, 7, 8, 9)

本ボードには、4ch (拡張ボード VS-WRC004LV を使用すると+3ch) のアナログセンサ入力を搭載しています。「ピュート」、「ピュート チェイサー」「ピュートローバー」や VS-WRC003、VS-WRC003LV に対応したセンサデバイスもそのまま接続できます。CN8 (AN3)、CN9 (AN4) には、別途コネクタをはんだ付けする必要があります。

自作のアナログ入力デバイスを接続する場合は、各ピンの仕様に従って作成してください。



○ ピン配置

- 1 pin : +3.3V (電源、赤外エミッタカソード)
- 2 pin : 100Ω-GND (赤外エミッタアノード)
- 3 pin : GND
- 4 pin : 信号入力 (+3.3V プルアップ)

○ C 言語関数

UINT AdRead(BYTE ch);

AD 入力値取得。A/D 変換の入力値を取得。

引数: チャンネル (0~7 = 1~8)

戻り値: A/D 変換の値 (0~1023)

○ ポート

VS-WRC103LV

- CN6 (AN1) PIO0_11
- CN7 (AN2) PIO1_0
- CN8 (AN3) PIO1_1
- CN9 (AN4) PIO1_2

VS-WRC004LV (オプション)

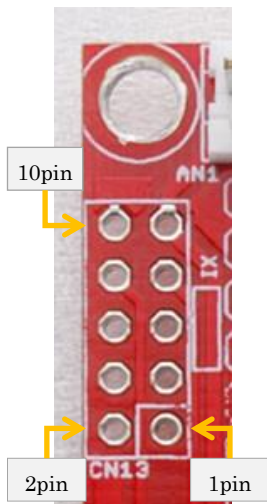
- CN7 (AN5) PIO1_3
- CN8 (AN6) PIO1_4
- CN9 (AN7) PIO1_10

5-8 IXBUS (CN13)

このポートは、IXBUS 拡張基板を接続するためのコネクタです。ビュートビルダー2 で使用する場合は、[ジャイロ/加速度センサ拡張ボード VS-IX001](#)、[アナログ入力拡張ボード VS-IX008](#) を使用することができます。使用する場合必ず基板上的 DIP スイッチの 3, 4 を OFF にしてください。

VS-IX001、VS-IX008 の出力はすべて 0~4095 の間で出力されます。

C 言語でプログラミングする場合、すべてのボードが使用可能です。また、IXBUS 拡張で使用するポートとは別にシリアル通信ポート (SCI3、1,2pin) も備えています。

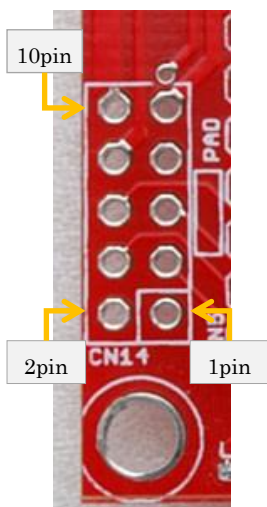


○ピン配置

1 pin	: PIO1_7 (TXD)
2 pin	: PIO1_6 (RXD)
3 pin	: RES
4 pin	: NC
5 pin	: PIO0_4 (SCL)
6 pin	: PIO0_5 (SDA)
7 pin	: +5V
8 pin	: +Vbat
9 pin	: +3.3V
10 pin	: GND

5-9 VS-C1 接続用コネクタ (CN14)

このポートは、無線コントローラ VS-C1 を接続するためのコネクタです。接続するためには別途無線コントローラ接続セットが必要です。



○ブロック



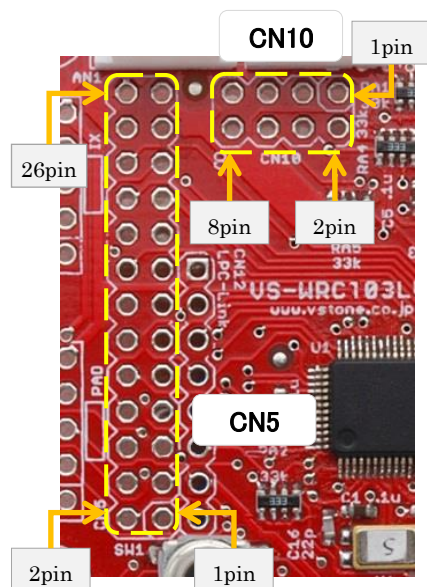
○ピン配置

1 pin	: NC
2 pin	: NC
3 pin	: PIO3_3_
4 pin	: PIO3_0
5 pin	: PIO3_1
6 pin	: PIO3_2
7 pin	: +5V
8 pin	: +Vbat
9 pin	: +3.3V
10 pin	: GND

5-10 拡張IO (CN5、CN10)

このポートは、本ボード内では使用していないIOポートをまとめたコネクタとなります。

このポートにはIO拡張ボード「VS-WRC004LV」が接続することができます。C言語サンプルソースはVS-WRC004LVを使用する前提で記述されていますので、この拡張IOを使用する際はご注意ください。



OCN10 ピン配置

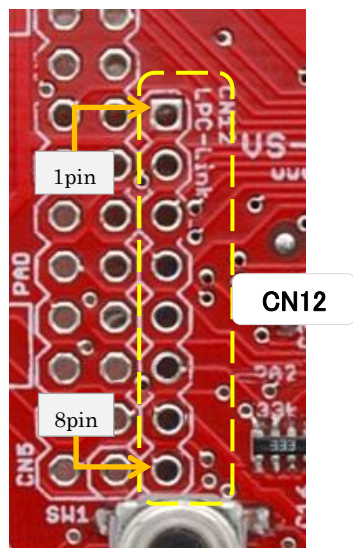
1 pin	: NC
2 pin	: NC
3 pin	: NC
4 pin	: NC
5 pin	: NC
6 pin	: PIO2_9
7 pin	: NC
8 pin	: PIO2_8

OCN5 ピン配置

1 pin	: PIO1_10/AD6	14 pin	: PIO2_11
2 pin	: PIO1_4/AD5	15 pin	: PIO1_5
3 pin	: PIO1_3/AD4	16 pin	: PIO0_2
4 pin	: NC	17 pin	: PIO2_7
5 pin	: NC	18 pin	: PIO2_6
6 pin	: NC	19 pin	: PIO2_5
7 pin	: PIO0_10	20 pin	: PIO2_4
8 pin	: PIO0_9	21 pin	: AES
9 pin	: PIO1_7	22 pin	: NMI
10 pin	: PIO1_6	23 pin	: +5V
11 pin	: PIO2_11	24 pin	: +Vbat
12 pin	: PIO2_10	25 pin	: +3.3V
13 pin	: PIO2_11	26 pin	: GND

5-1 1 LPC-Link 接続ポート (CN12)

このポートは、NXP セミコンダクターズ製の評価ボード LPCXpresso などに付属しているデバッガ「LPC-Link」を接続できるポートです。このポートに「LPC-Link」を接続することで、開発環境 LPCXpressoIDE で、デバッグ（ブレークポイントでのプログラム一時停止、マイコン内部のメモリ、変数の確認など）をおこなうことができるようになります。



○ポート

1pin	:	+3.3v
2pin	:	PIO1_3/SWDIO
3pin	:	PIO0_10/SWCLK
4pin	:	PIO0_9/SWO
5pin	:	NC
6pin	:	RES
7pin	:	+5v
8pin	:	GND

6 開発環境の導入とサンプルプログラムの実行

本項では、「VS-WRC103LV」を C 言語で開発する際に使用可能な、無償配布されている“LPCXpresso”を用いたサンプルプログラムの実行に関して解説をします。LPCXpresso 評価版のご利用は無料ですが、ダウンロードの際にユーザーアカウントを作成、またビルドサイズの制限がかかっていますので、インストール後にネットワークを利用した認証作業をする必要があります。

おおまかな手順は以下の通りです。

- ① 開発環境の入手、インストール
- ② サンプルプロジェクトのダウンロード、ビルド
- ③ プログラムの書き込み

6-1 開発環境 LPCXpresso の入手

- ① LPCXpresso は、NXP からダウンロードします。下記 URL にアクセスしてください。

https://www.nxp.com/ja/webapp/crcl.ccr_register.framework?ACTION_TYPE=registerpage&lang_cd=ja

- ② アカウントの作成では以下のような項目を入力して下さい。

アカウントを作成する

メール * (仕事用のEメール・アドレスをご入力いただければ、優先的に対応いたします。)

メールアドレス

Eメール・アドレスを再入力 *

メールアドレス確認用

パスワード * (Minimum 6 characters, 1 number, 1 letter, no spaces)

パスワード

パスワードを再入力 *

パスワード確認用

名 *

名前

姓 *

苗字

正式会社名 *

所属している企業、学校名など

職種 *

選択してください

所属している企業、学校の職種

地位/役職 *

選択してください

所属している企業、学校での地位、役職

国・地域 *

選択してください

日本なら日本を選択

☒ 技術ニュース、イベント、トレーニング、デモおよびドキュメンテーションに関する最新情報を入手

NXP からのメールマガジン配信可否

☐ 次の事項に同意します。 ([利用規約](#) および [プライバシー・ポリシー](#))

規約への同意（必ずチェック）

☐ 私はロボットではありません

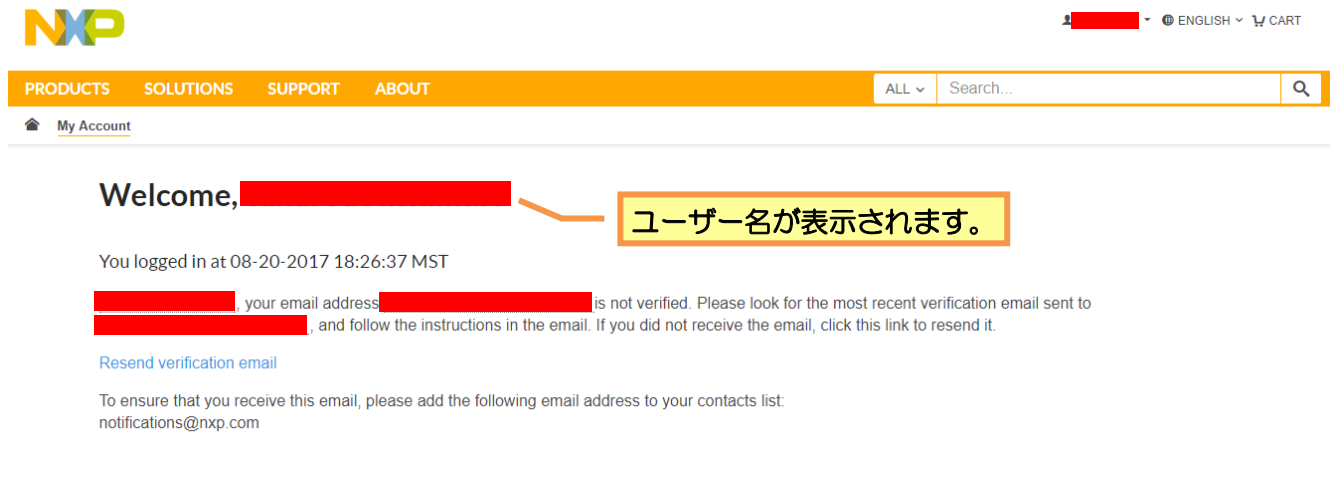
チェックをする

キャンセル

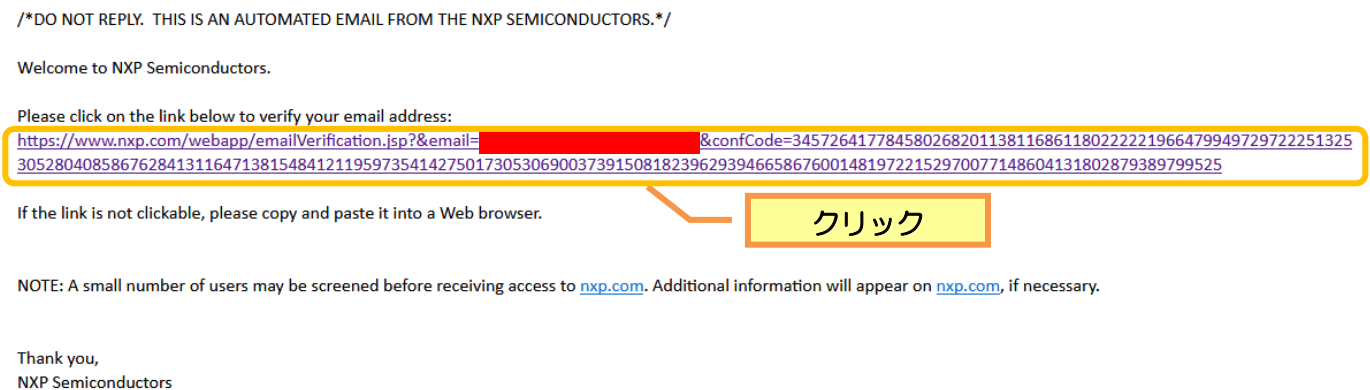
送信

入力が終わったら押す

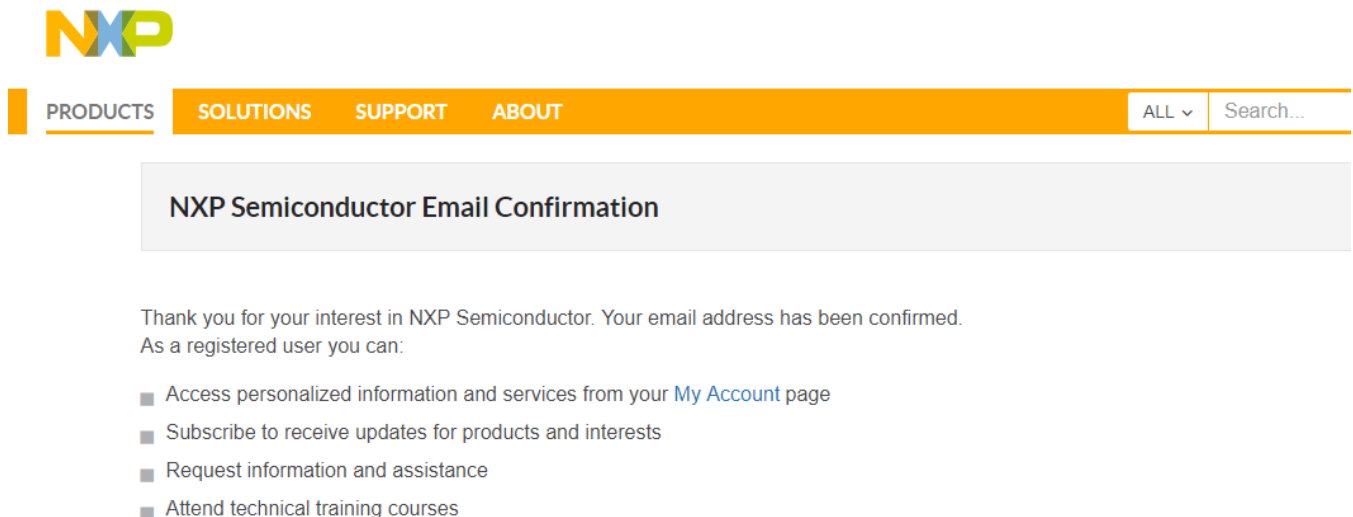
③ 送信を押すと画面が下記のように変わります。この段階ではまだ仮登録です。



④ ③の画面が表示されると、登録したメールアドレス宛にメールアドレス認証メールが届きますので、記載のURL をクリックして下さい。クリックできない場合は該当のURL をコピーして、ウェブブラウザに貼り付けてアクセスしてください。



⑤ URL クリック後、下記のページに遷移すればアカウント作成完了です。



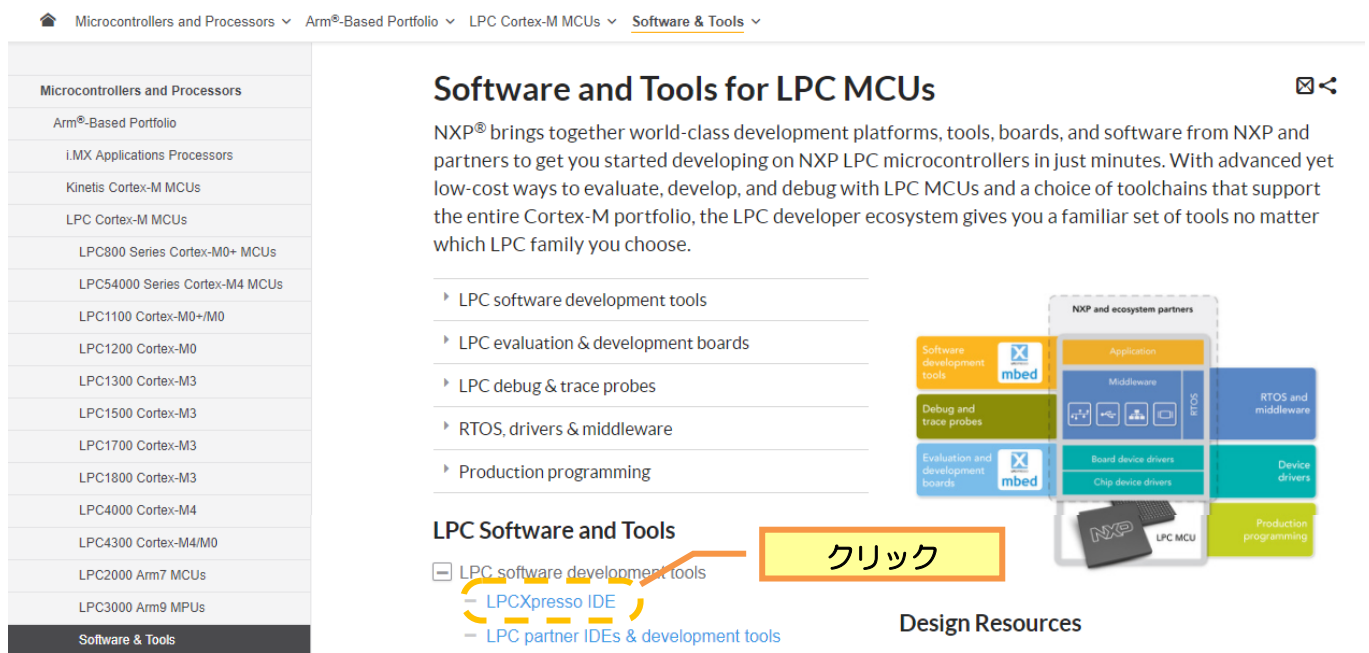
⑥ アカウント作成後、下記 URL をクリックし、ログインしてください。

<https://www.nxp.com/security/login?TARGET=https%3A%2F%2Fwww.nxp.com%2Fruhp%2FmyAccount.html>

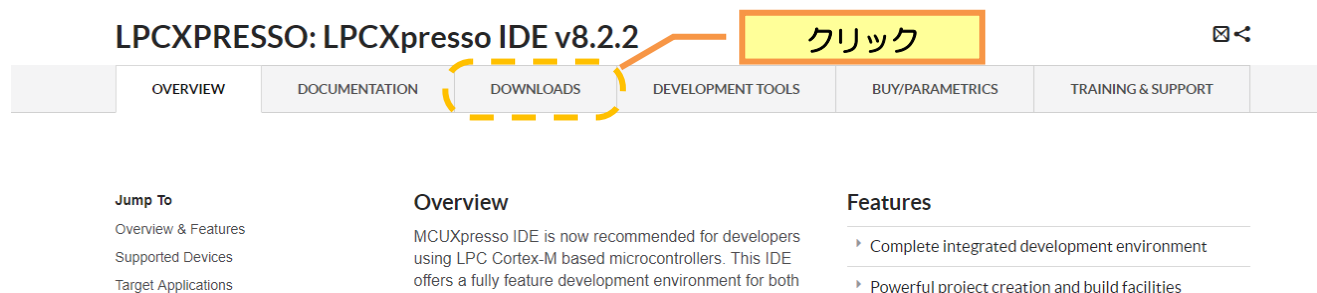


⑦ ログイン後、下記 URL をクリックし、「LPC Software and Tools」の項目にある「LPCXpresso IDE」のリンクをクリックしてください。

<http://www.nxp.com/products/developer-resources/software-development-tools/software-tools:LPC-SW-TOOLS>



- ⑧ クリックすると下記のページに切り替わります。ページ上側にあるタブメニューより「DOWNLOAD」をクリックして画面を切り替えます。



- ⑨ 切り替わったページの「DOWNLOAD」ボタンをクリックしてください。



- ⑩ 「LPCXpresso IDE for Windows v8.x.x」をクリックしてください。

NXP > Software & Support > Product Information : LPCXpresso IDE

Software & Support

Product List
Product Search
Order History
Recent Product Releases
Recent Updates

Licensing

License Lists
Offline Activation

FAQ

Download Help
Table of Contents
FAQs

Product Information

LPCXpresso IDE

To register a New Product please click on the button below

Register

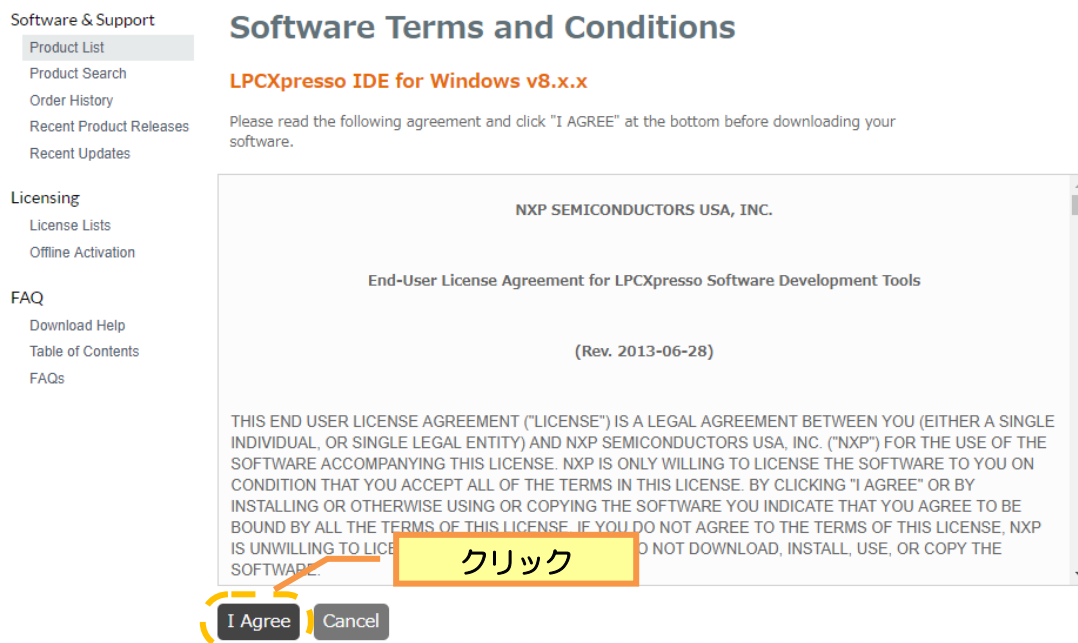
Current

Previous

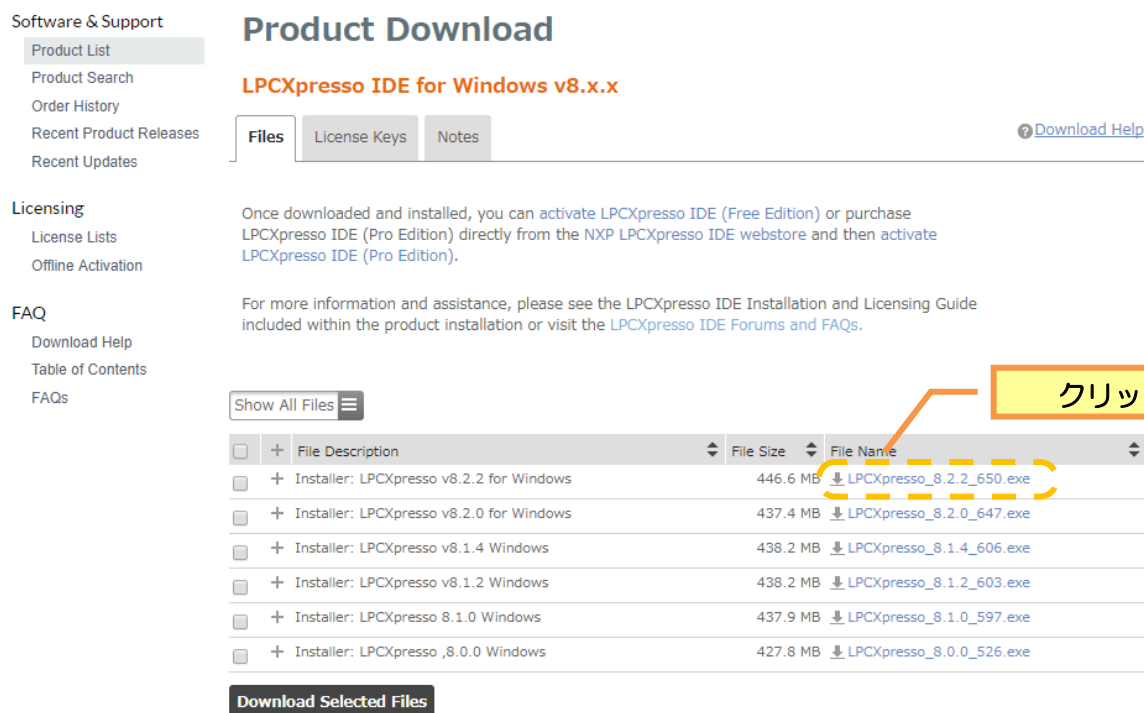
Version	Description	
8.X.X	LPCXpresso IDE for Linux v8.x.x	Download Log
8.X.X	LPCXpresso IDE for MAC v8.x.x	Download Log
8.X.X	LPCXpresso IDE for Windows v8.x.x	Download Log

※本 CPU ボードは Windwos 環境のみの対応とさせていただいております。ここで提供されている他のプラットフォームにつきましては、動作検証・サポート等は行っておりませんのでご了承ください。

- ⑪ 規約に同意する画面に遷移しますので、I Agree をクリックしてください。



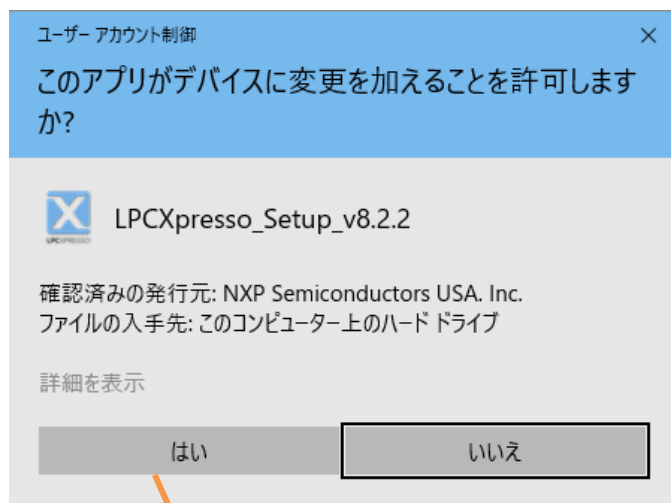
- ⑫ 表示される一覧より「LPCXpresso_8.2.2_650.exe」のリンクをクリックしてインストーラをダウンロードしてください。クリックするとダウンロードが始まります。ファイルのサイズが非常に大きいため、ダウンロードに時間がかかる場合がありますので、ご注意ください。



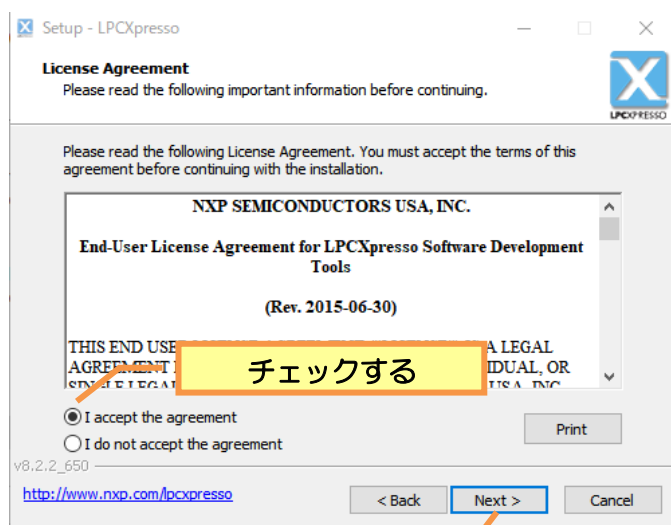
6-2 LPCXpresso のインストール

ダウンロードした LPCXpresso を PC にインストールする方法について説明します。
ダウンロードしたインストーラ（exe ファイル）を実行し、以下の手順に沿ってインストールを進めます。
また、インストール中にネットワークに接続することがあります。その際は、ネットワーク接続を許可して作業を進めてください。

※Windows のバージョンによって、仕様が若干異なる場合があります。

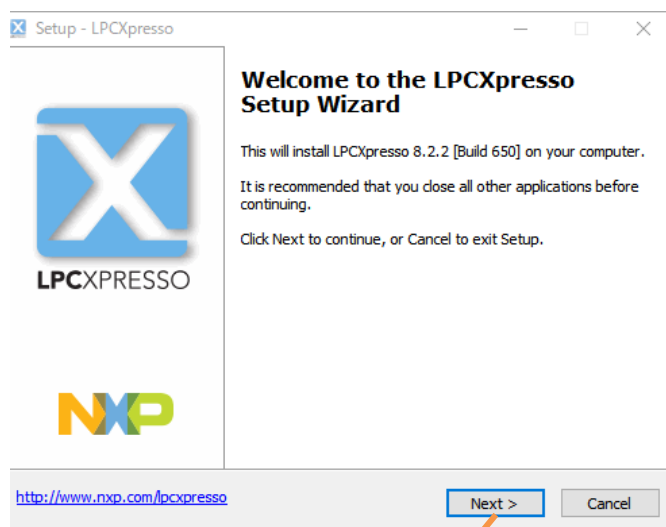


はいを押す

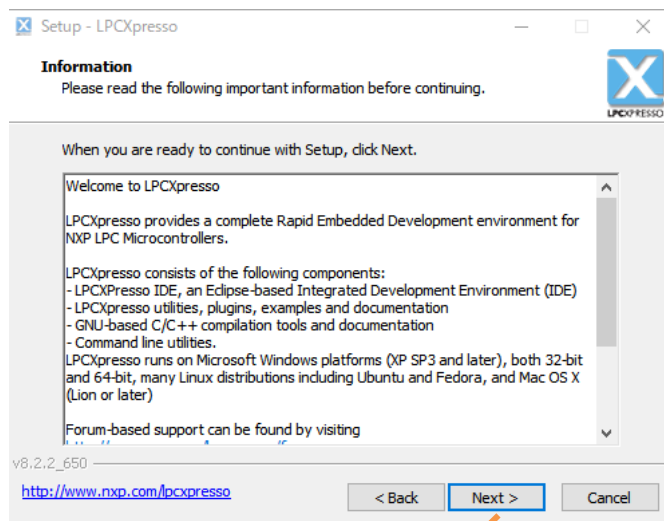


チェックする

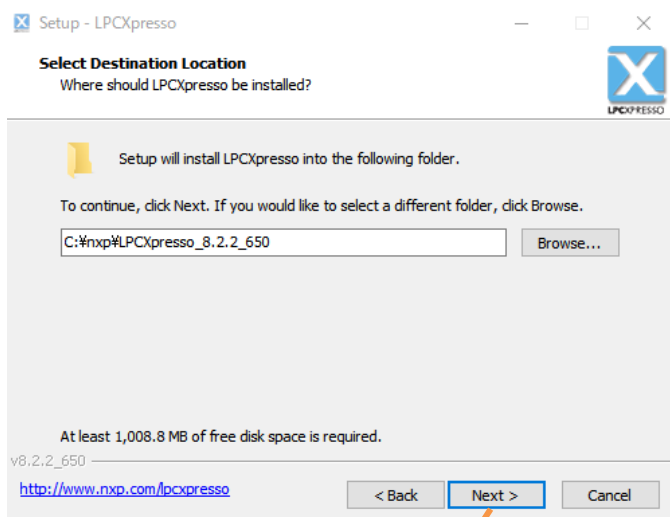
NEXT を押す



NEXT を押す



NEXT を押す



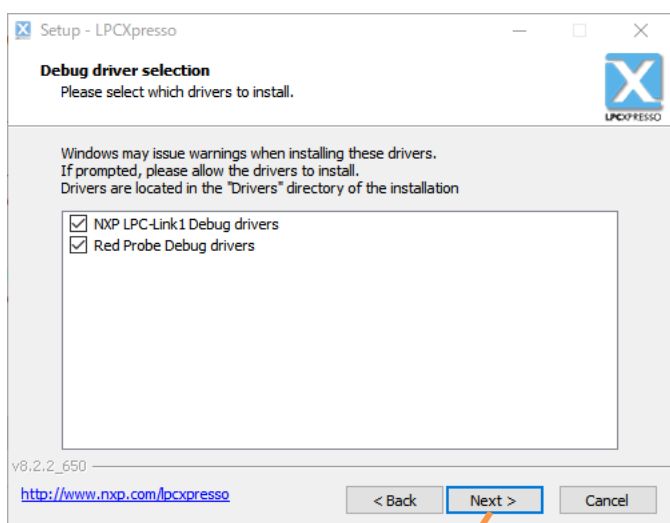
Folder Does Not Exist

The folder:
C:\nxp\LPCXpresso_8.2.2_650
does not exist. Would you like the folder to be created?

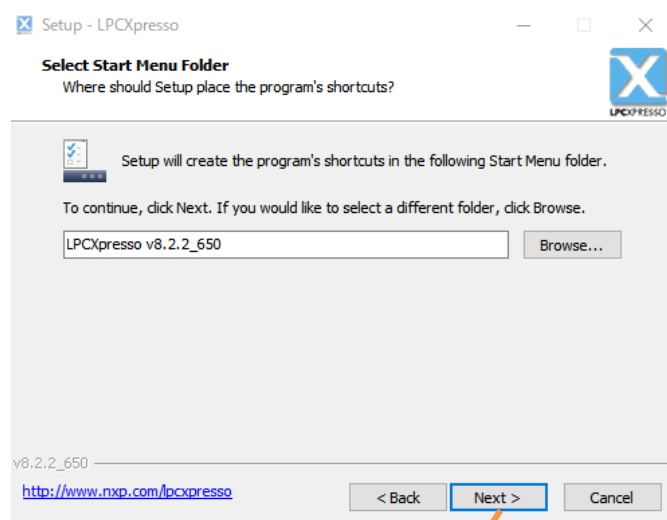
はい(Y)

いいえ(N)

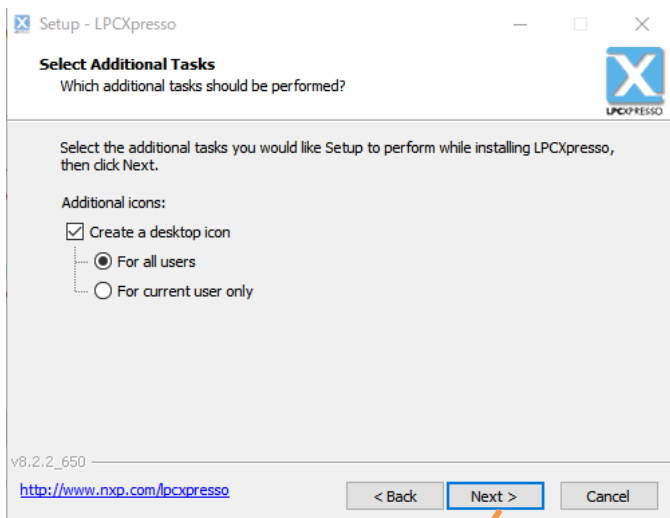
はいを押す



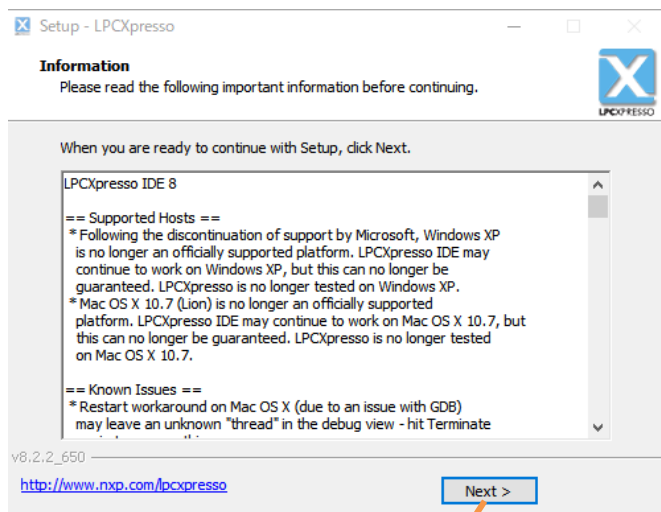
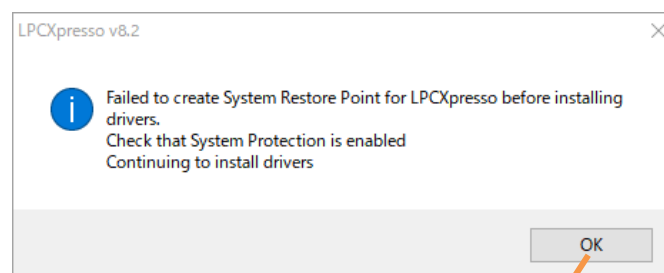
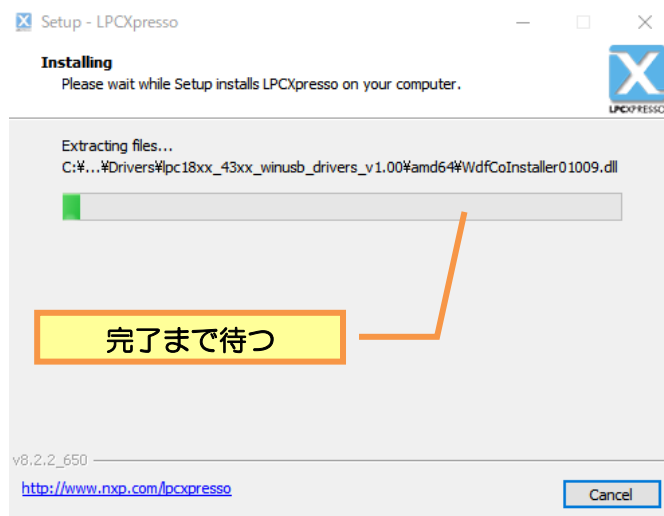
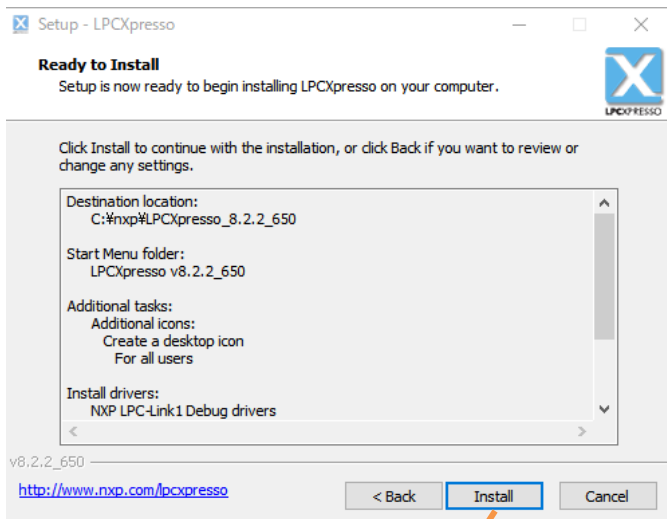
NEXT を押す



NEXT を押す



NEXT を押す



以上で、LPCXpresso のインストールは完了です。続いて次項で説明する認証作業を行ってください。

6-3 LPCXpresso の起動と認証

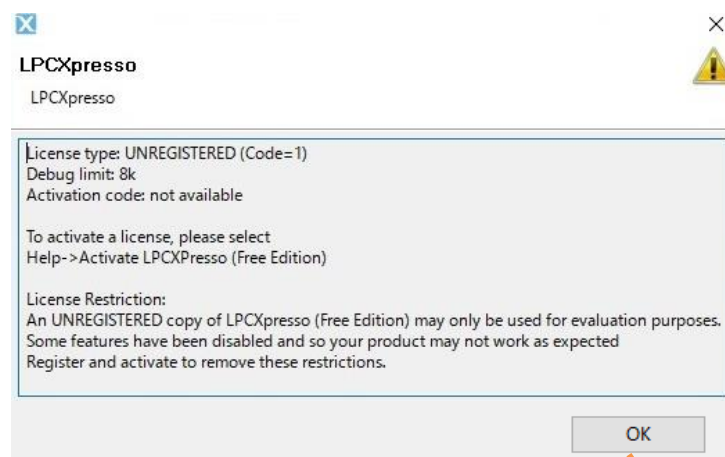
次に、以下の手順に沿って、認証作業を行います。認証作業ではインストールした PC をインターネットに接続する、または、インターネットに接続している PC が必要になります。

① まず、LPCXpresso を起動します。起動するには、デスクトップ上のアイコンをダブルクリックするか、スタートメニューのすべてのプログラムより「LPCXpresso v8.2.2」>「LPCXpresso v8.2.2」をクリックしてください。

② 起動すると以下の画面が表示されます。

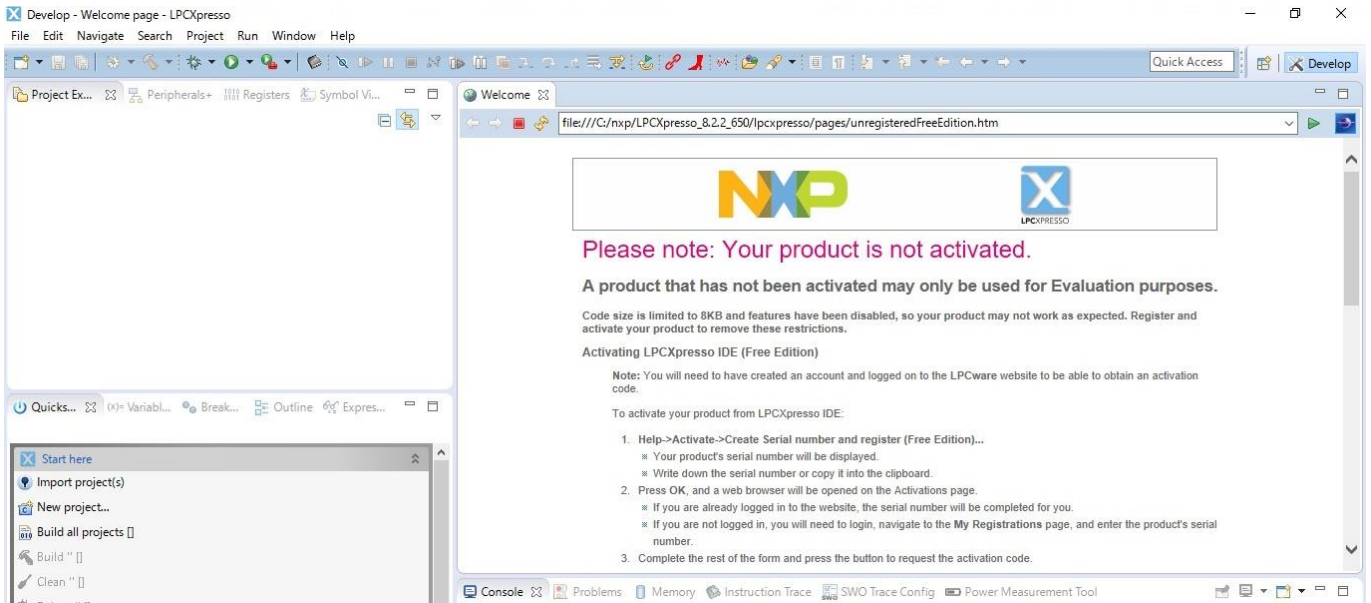


③ 認証を行っていないと以下のような表示がされますので、OK を押します。

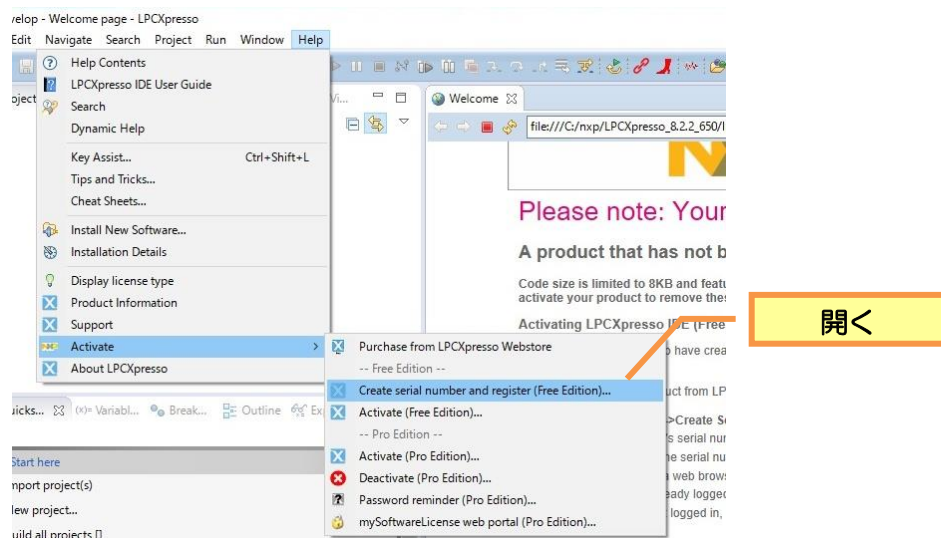


OK を押す

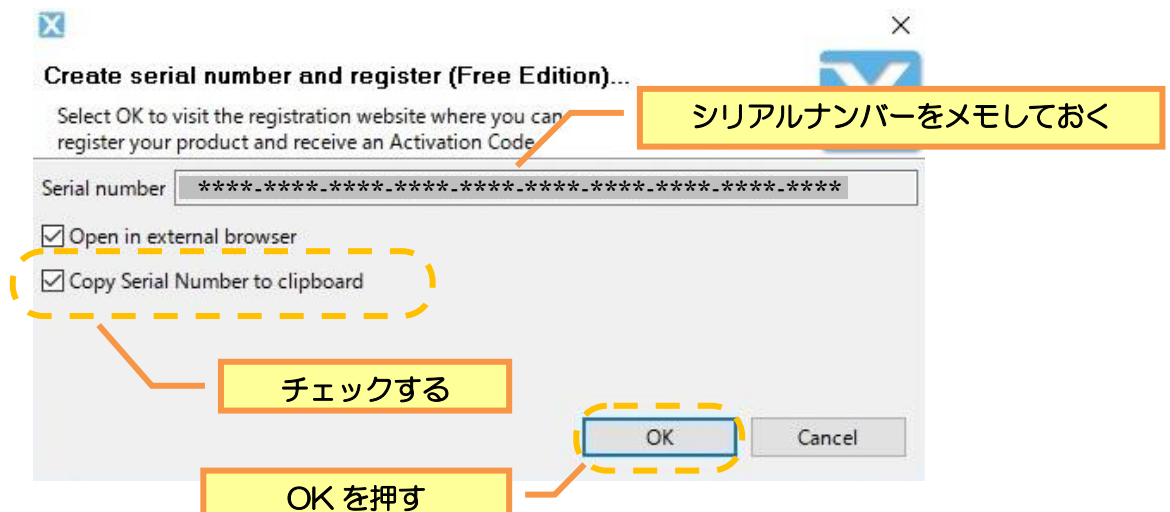
④ 起動すると、以下の編集画面が表示されます。



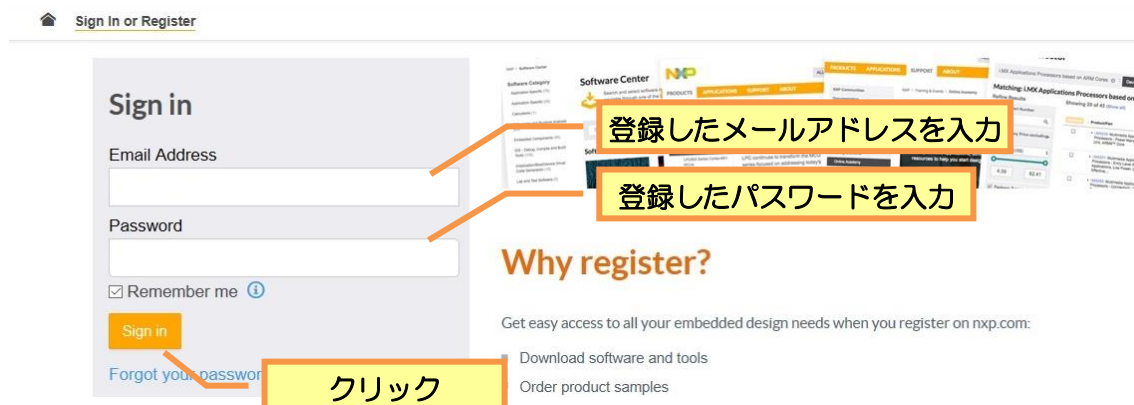
⑤ ツールバーのヘルプから、Help>Activate>Create serial number and register (Free Edition)を開きます。



⑥ 表示されたシリアルナンバーをメモ、コピーなどして、一時的に保存しておき、OK を押します。



- ⑦ OK を押すと、以下のログイン画面が開きます。インストーラをダウンロードした際に登録したアカウントのユーザー名、パスワードを入力し、ログインします。
(ログイン済みの場合、⑨の画面が開きますので、⑦、⑧は飛ばします。)



- ⑧ ログインが完了したら、My Activations のページが表示されるので、Activate をクリックしてください。



- ⑨ Serial Number :の項目に、先ほどメモしたシリアルナンバーを入力し、「Register LPCXpresso」ボタンをクリックしてください。



LPCXpresso IDE Free Edition Activation

Instructions:

- 1) Enter your LPCXpresso IDE serial number, as created by the IDE menu:

Help -> Activate -> Create Serial Number

- 2) Click on Register LPCXpresso

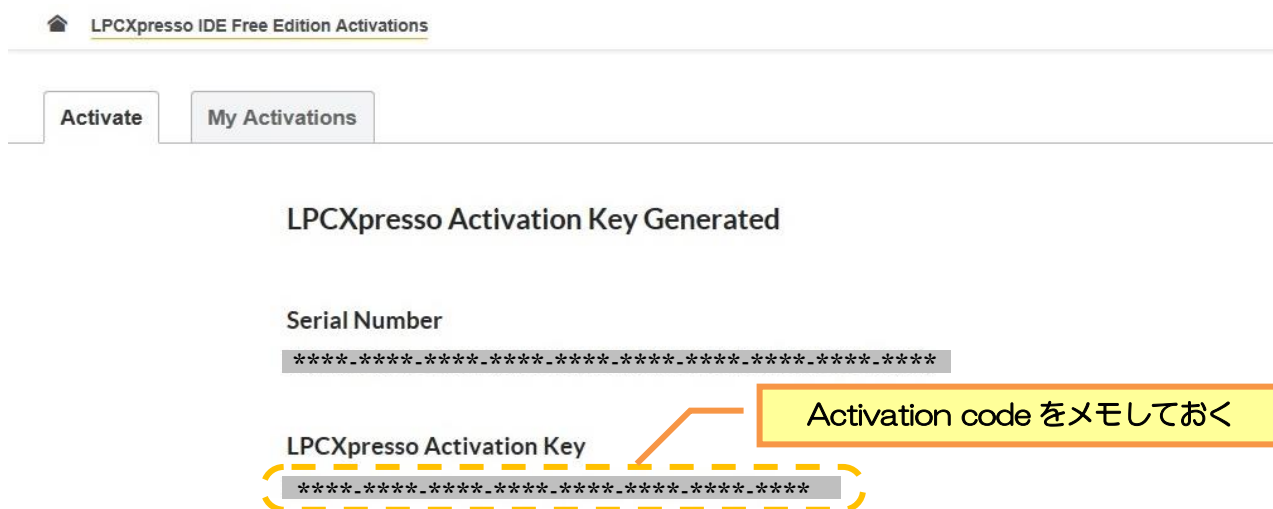
- 3) Copy your generated LPCXpresso IDE Free Edition activation code and paste it into the IDE menu:

Help -> Activate -> Activate (Free Edition)

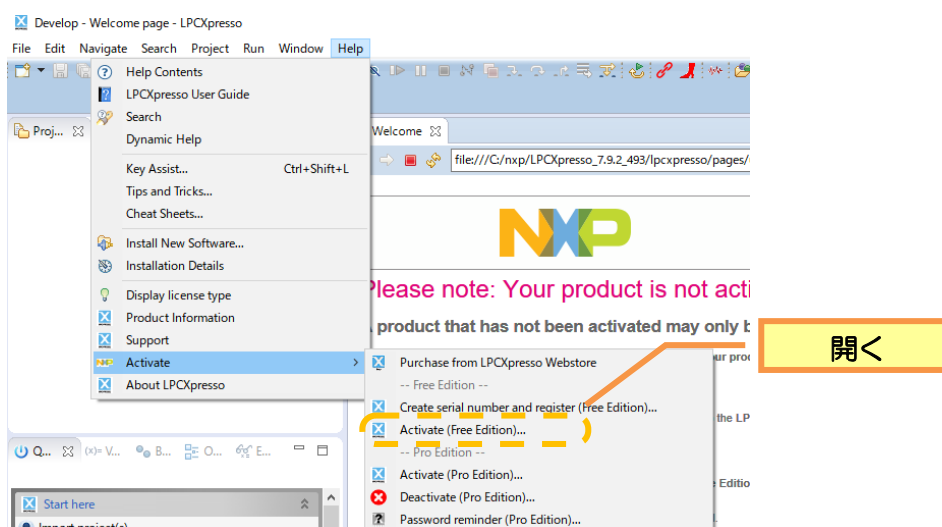


- ⑩ ボタンを押すと、以下のように「activation code」が表示されますので、メモを取ってください。メモを取る前にページを閉じてしまった場合は、NXP メンバーページにログインし、セキュア・アプリケーションの「My LPCXpresso activations」をクリックしてください。「My Activations」タブを選択していただくと「LPCXpresso Activation Key」が確認できます。

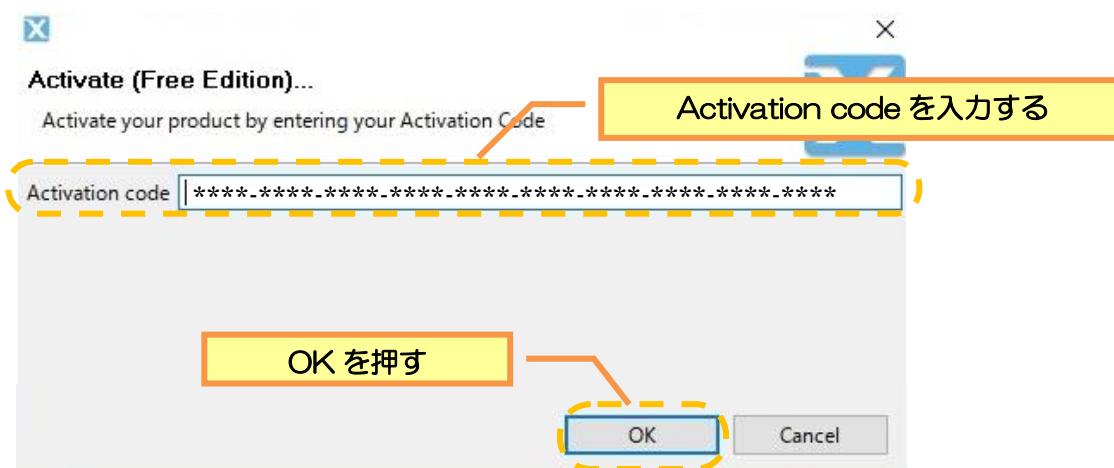
NXP ログインページ：<https://www.nxp.com/security/login>



- ⑪ LPCXpresso のツールバーから Help>Activate>Activate(Free Edition)を選択してください。



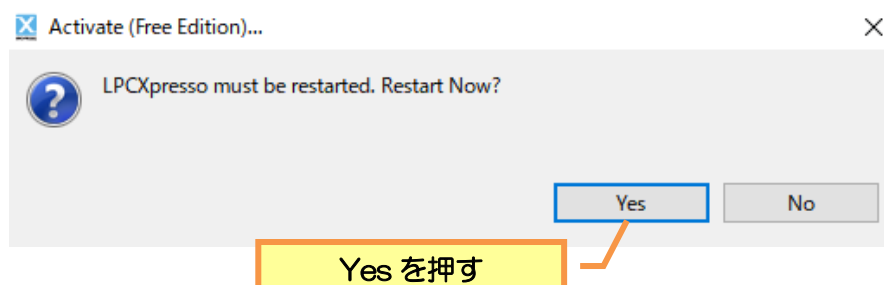
- ⑫ 入力欄が表示されるので、⑩で表示された「activation code」を入力して OK を押します。



- ⑬ 認証が正常に完了すると、以下のように表示され、制限が解除されます。OK を押します。



- ⑭ アクティベーションが完了すると、LPCXpresso を再起動する旨のポップアップが出ますので Yes を選択して再起動してください。



以上で、LPCXpresso の認証作業は完了です。

6-4 サンプルプロジェクトのインポートとビルド

いよいよ、C 言語サンプルプログラムを読み込み、CPU へ書き込みます。ここで使用するサンプルプログラムは、各種設定を行った状態のプロジェクト単位でサンプルを配布していますので、「サンプルプロジェクト」と呼びます。以下の手順に沿って、サンプルプロジェクトのインポートとビルドを行ってください。

(1) ワークスペースフォルダの変更

LPCXpresso をインストールした際に、以下の場所に標準のワークスペースが作られます。このとき、ユーザー名に全角文字（日本語など）が含まれる場合、**正常にビルドできませんので、ワークスペースフォルダの変更を行ってください。**

- WindowsXP の場合

C:\Documents and Settings\[ユーザー名]\My Documents\LPCXpresso_8.2.2_650\workspace

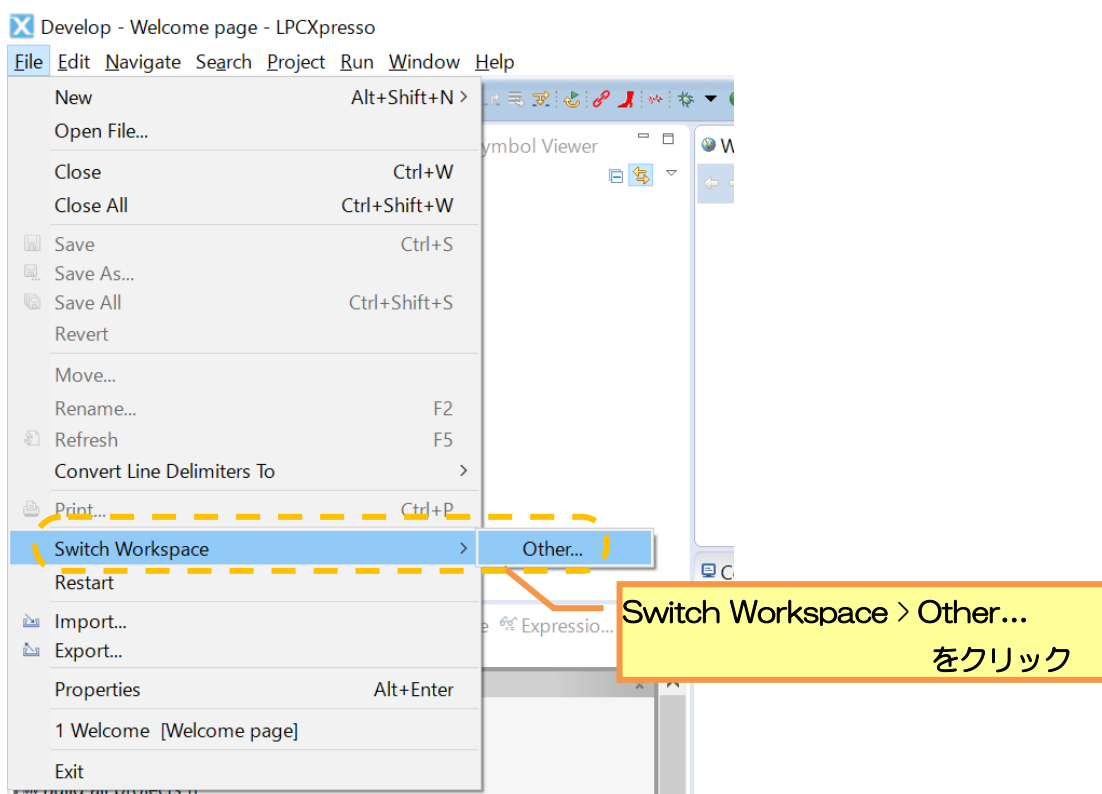
- Windows Vista/7/8/8.1/10 の場合

C:\Users\[ユーザー名]\Documents\LPCXpresso_8.2.2_650\workspace

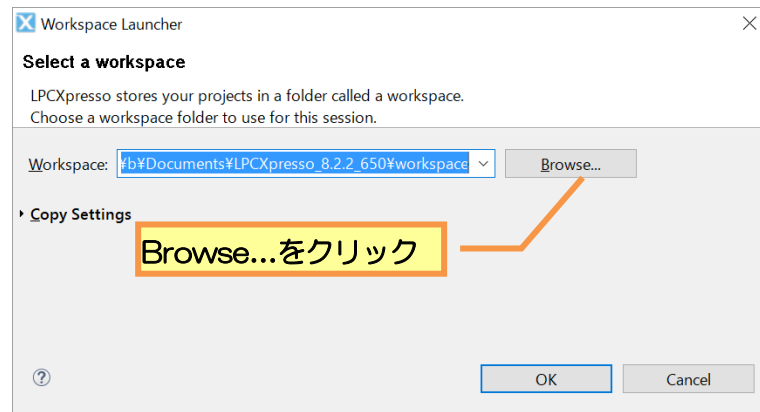
※ユーザー名は Windows にログインしているユーザー名です。

ユーザー名に全角文字が含まれていない場合は変更を行う必要が無いため、(2) サンプルプロジェクトのインポートにお進みください。また、上記説明がよくわからない場合、以下の手順でワークスペースフォルダの変更を行ってください。

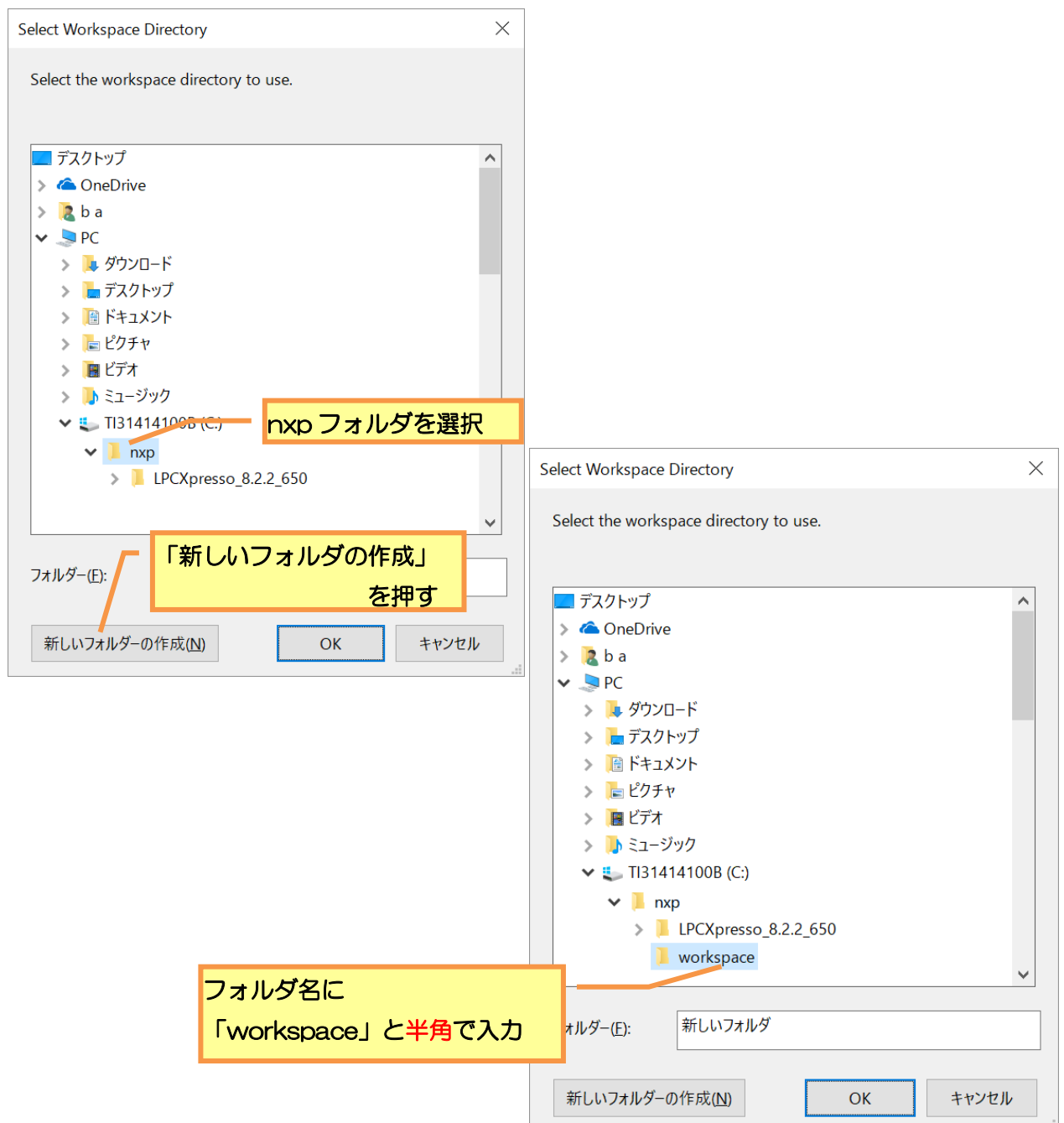
- ① LPCXpresso のファイルメニューから「Switch Workspace > Other...」を選択しクリックします。



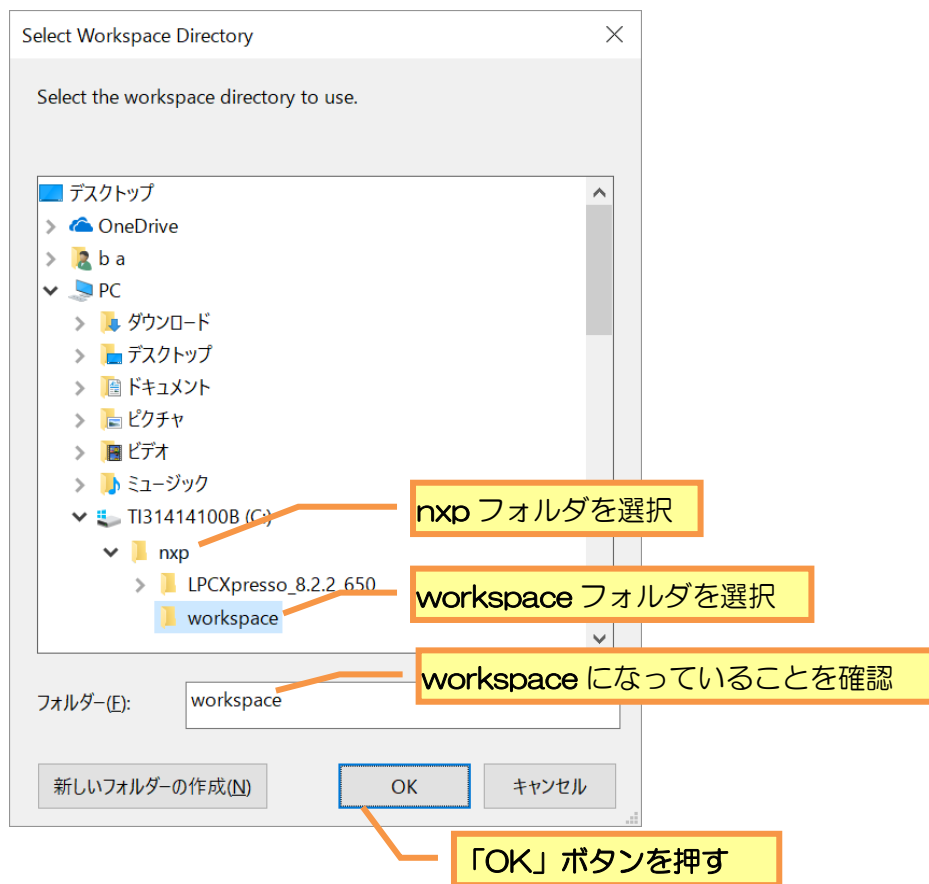
② 「Brows...」をクリックします。



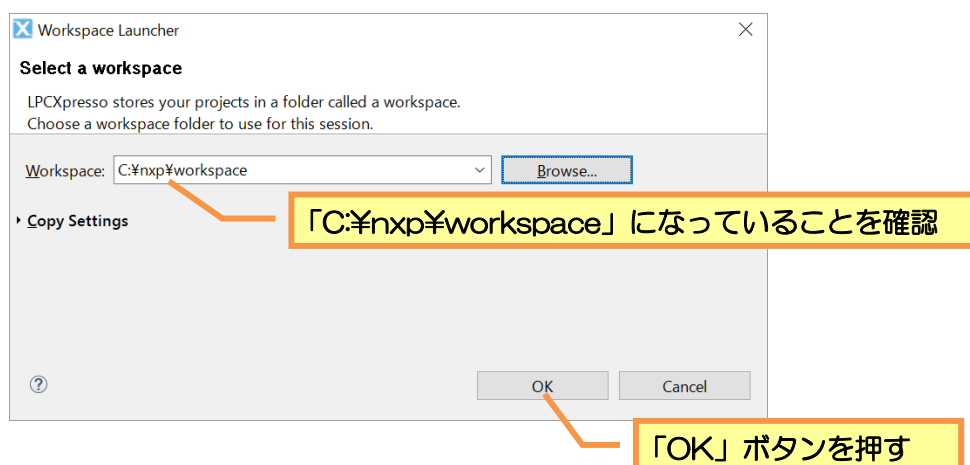
③ マイコンピュータ内の「ローカルディスク(C:) > nxp」フォルダを選択し、「新しいフォルダの作成」ボタンをクリックします。作成された「新しいフォルダ」の名前を「workspace」に変更します。この時、必ず半角文字で入力してください。



- ④ このままでは、フォルダが反映されていないので、フォルダ名に直接「workspace」と入力するか、以下の手順で、一度別のフォルダをクリックした後に、再度「workspace」フォルダをクリックし、フォルダ名が「worksoace」になっていることを確認し、「OK」ボタンを押します。



- ⑤ workspace フォルダが、「C:¥nxp¥workspace」になっていることを確認し、「OK」ボタンを押します。



以上で、ワークスペースフォルダの変更は完了です。

(2) サンプルプロジェクトのインポート

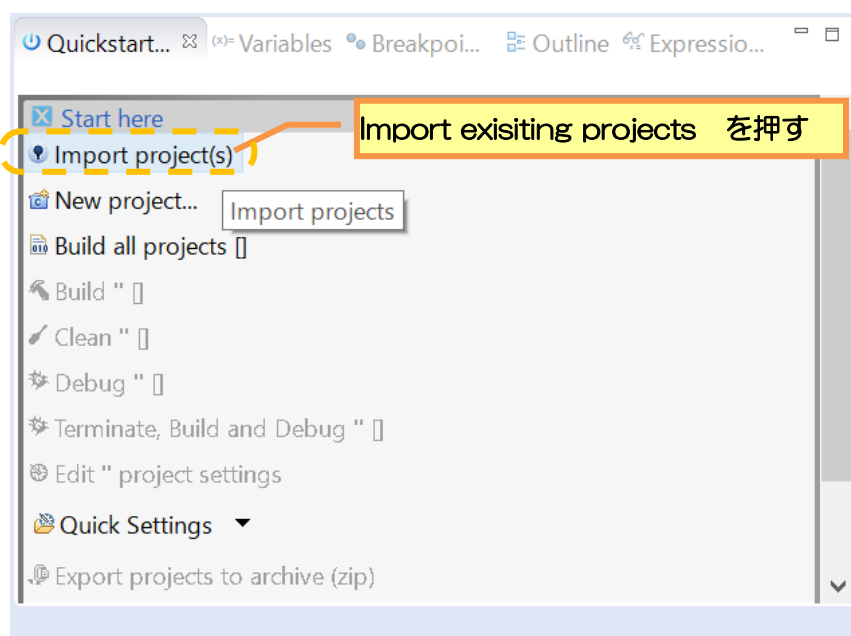
次にワークスペースにサンプルプロジェクトをインポートします。

- ① 以下の URL より、VS-WRC103LV サポートページにアクセスし、LED 点滅サンプルプロジェクト「VS-WRC103LV_Sample_LED_*****.zip」（****はバージョン）をダウンロードします。

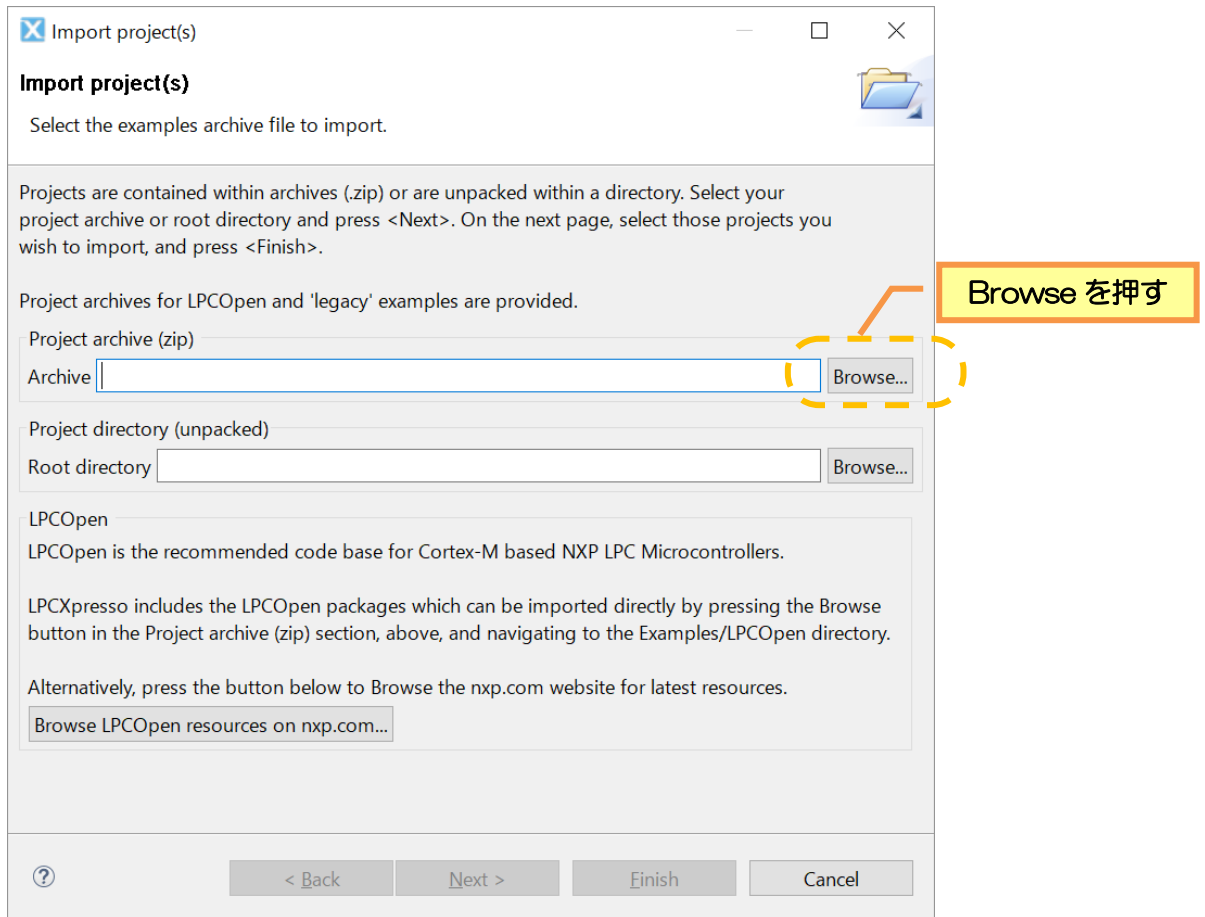
<http://www.vstone.co.jp/top/products/vs-wrc103lv>

※ サポートページで公開しているものが最新になりますが、 ビュートローバーの CD-ROM をお持ちの場合、CD-ROM 内の「C 言語_ARM¥サンプルプロジェクト」フォルダ内にも、各種サンプルを同封しています。

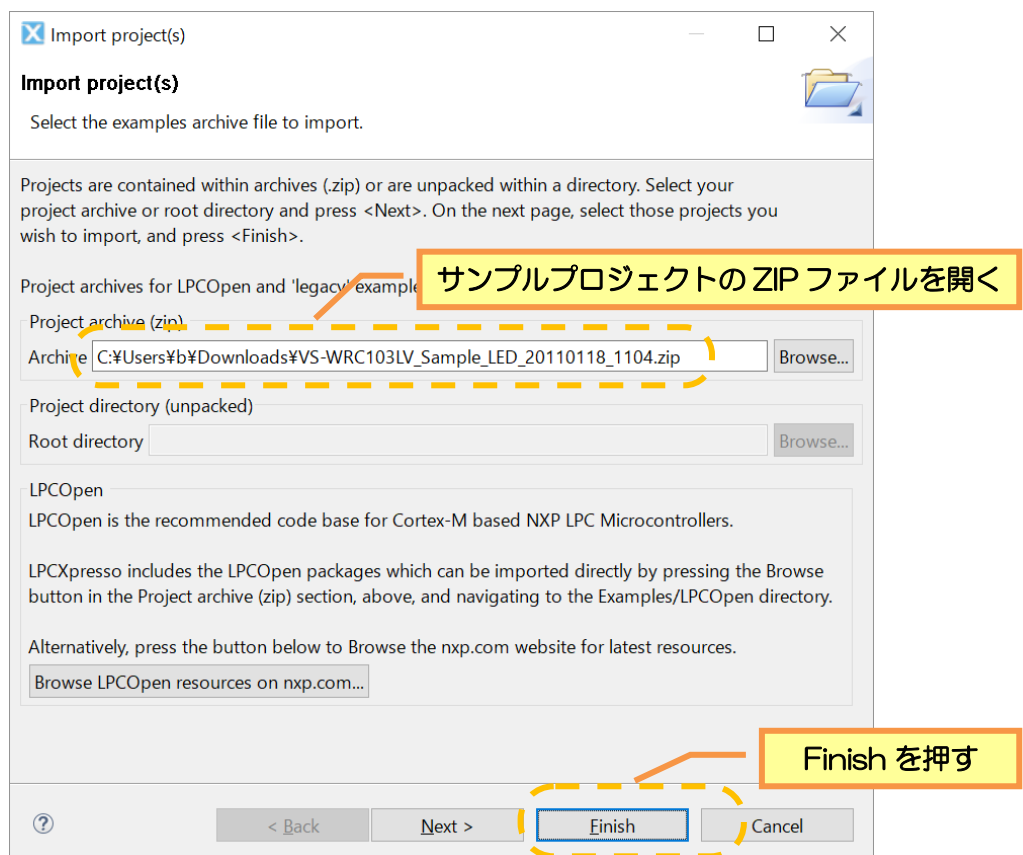
- ② LPCXpresso を起動し、画面左下の「Quickstart Panel」の「Start here」より「Import project(s)」を開きます。



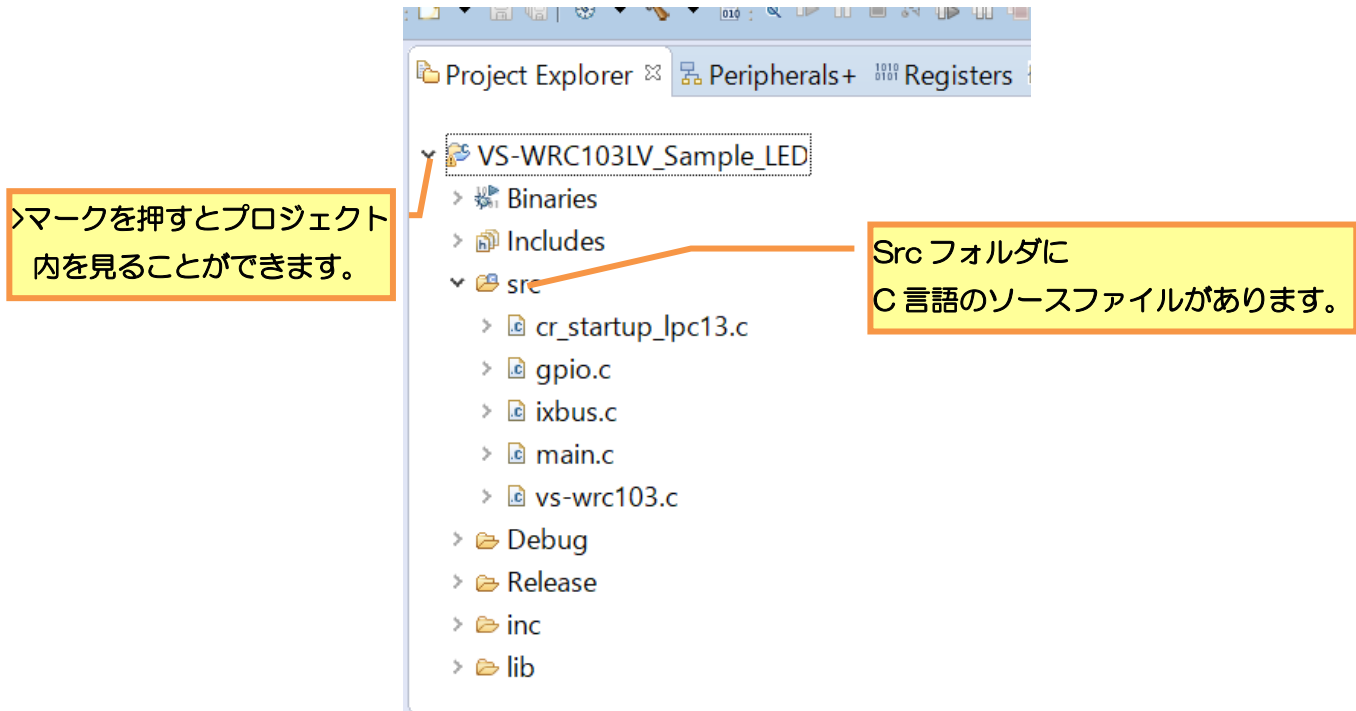
- ③ 以下のダイアログが表示されるので、「Project archive(zip)」内の「Browse」ボタンを押します。



- ④ インポートするプロジェクト（ここでは、「VS-WRC103LV_Sample_LED_*****_****」（*****はサンプルのバージョンの数字が入ります））の ZIP ファイルを開いて、「Finish」を押します。



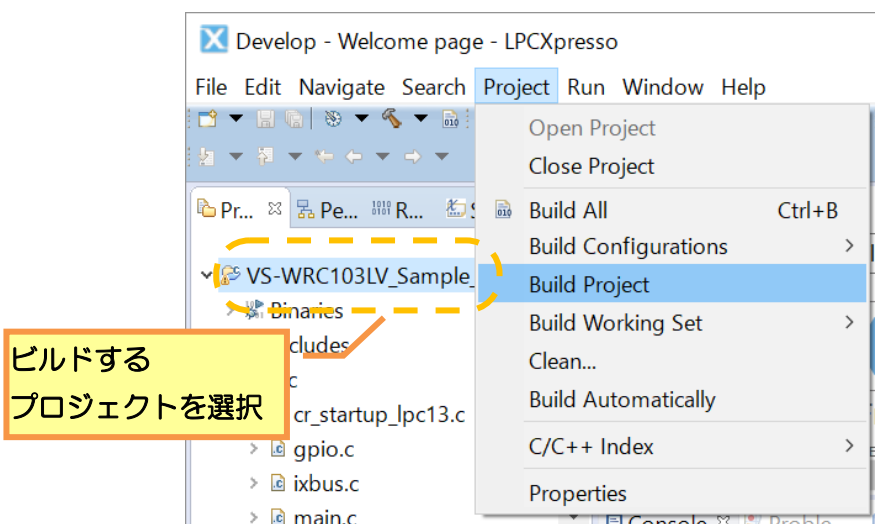
- ⑤ 「Finish」をクリックすると、画面左上の「Project Ex」にプロジェクトが追加されます。これでプロジェクトの読み込みは完了です。



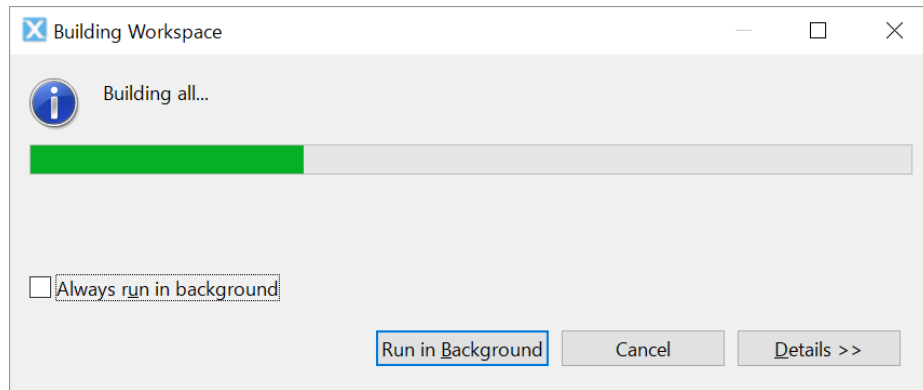
(3) サンプルプロジェクトのビルド

次に、プログラムをCPUボードに形式に変換する作業のビルドを行います。以下の手順に従って、ビルドを行ってください。

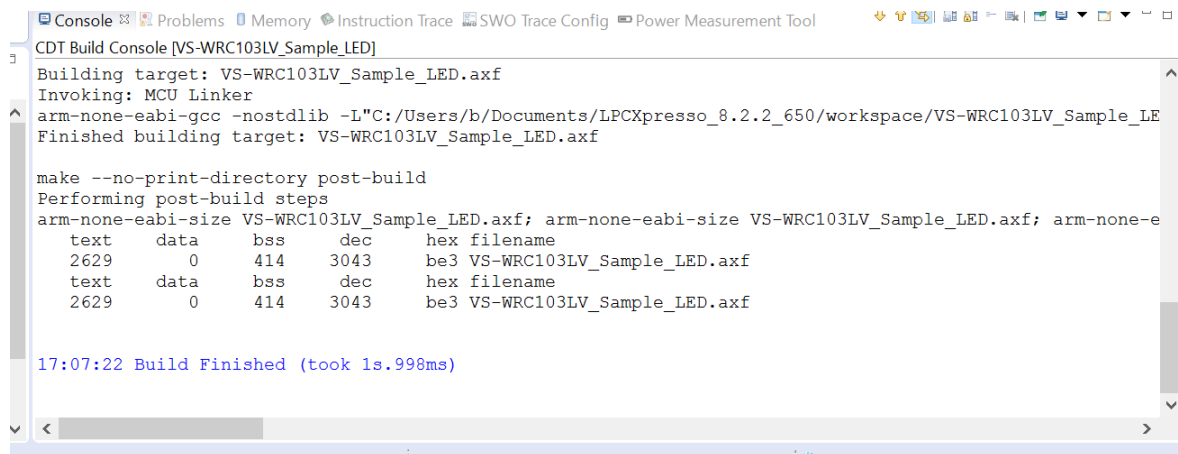
- ① 「Project Ex」でビルドしたいプロジェクト（ここでは、VS-WRC103LV_Sample_LED）を選択し、Project>Build Project をクリックしてビルドを開始します。



② ビルドを開始すると以下のダイアログが表示されますので、完了するまで待ちます。



③ 完了すると、ダイアログが閉じ、以下のように表示されます。



※ **ここで Error が出た場合**、(1)のワークスペースフォルダの変更が、正常に行えてない可能性があります。
再度ワークスペースフォルダの変更を行ってください。

(新しいフォルダの名前を変更で、「workspace」と入力するときは必ず半角で入力してください)

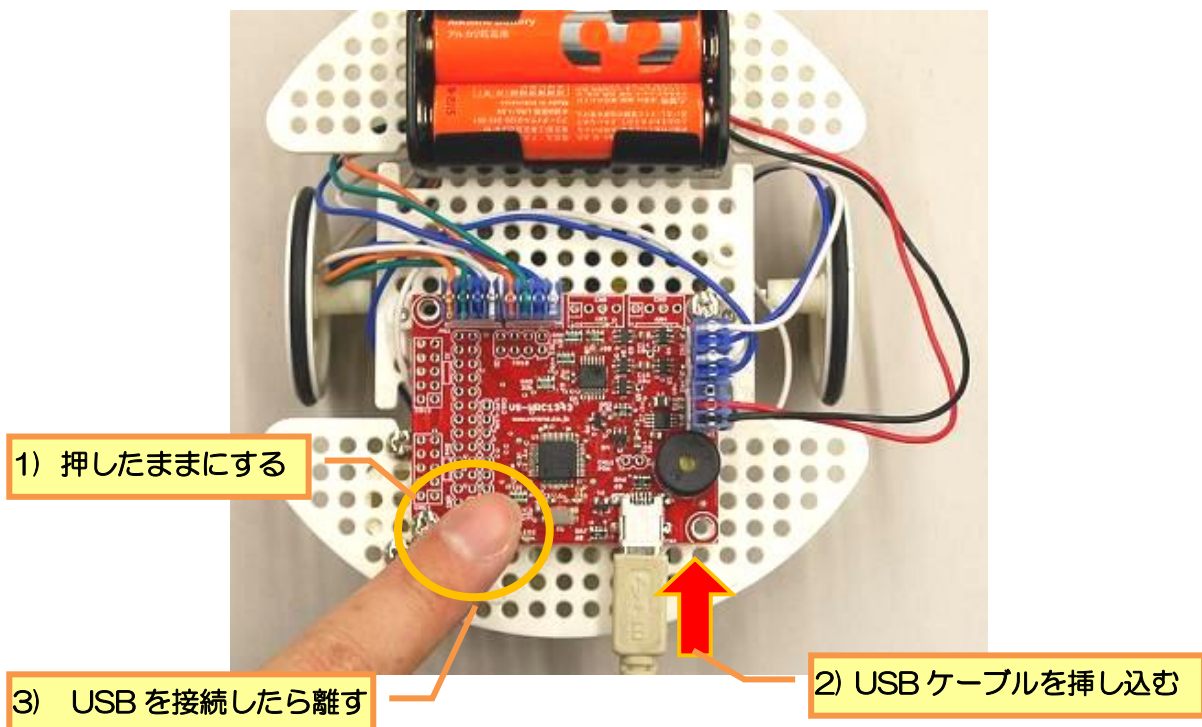
※ C 言語のソースを編集した後に、Error と表示された場合の対処方法は「6-7 エラーの修正」をご参考下さい。

6-5 VS-WRC103LV へのプログラムの書き込み

サンプルプロジェクトをビルドすると、ワークスペースフォルダ ¥ VS-WRC103LV_Sample_LED ¥ Debug 内に、「VS-WRC103LV_Sample_LED.bin」というファイルが作成されます。

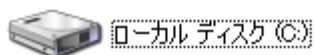
このファイルを CPU ボードに書き込むと、作成したプログラムを CPU ボードに実行させることができますので、以下の手順で、書き込みを行ってください。

- ① USB ケーブルを外し、CPU ボードの**全ての電源を OFF** にします。
- ② **プッシュボタンを押したまま**、PC と USB ケーブルで接続します。
- ③ USB を接続したらプッシュボタンは離しても OK です。
- ④ 20 秒程度待つと、PC が CPU ボードを USB メモリなどと同じ「マスストレージデバイス」として認識します。（初めて CPU ボードを PC に接続した場合、認識にしばらく時間がかかります。）

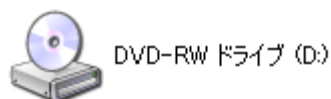


- ⑤ CPU ボードが PC に認識されると、「CRP DISABLE」という名称のドライブ名で表示されます。CPU ボードを接続したら「マイコンピュータ」を開いて、この名称のドライブが表示されることを確認し、ダブルクリックしてドライブを開きます。

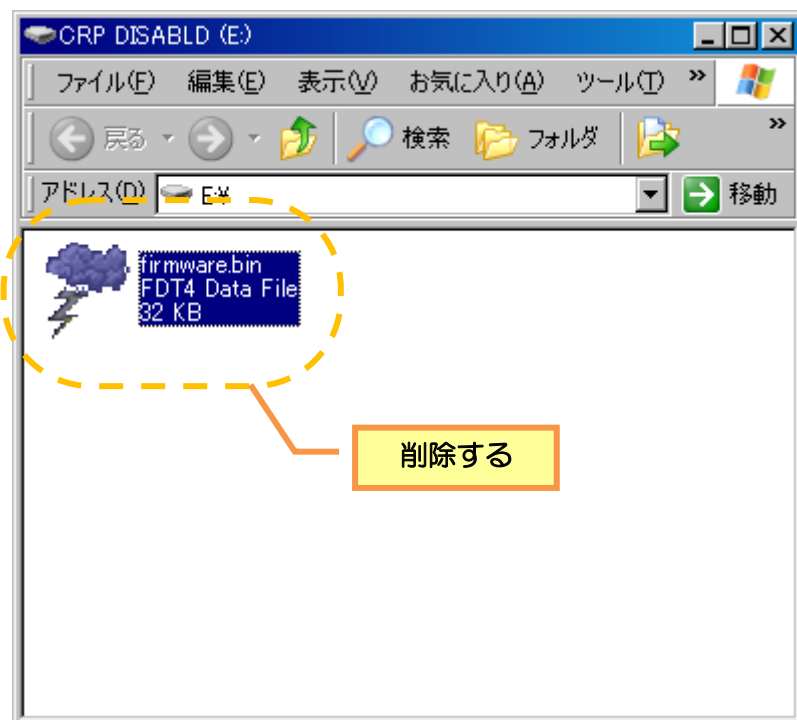
ハードディスクドライブ



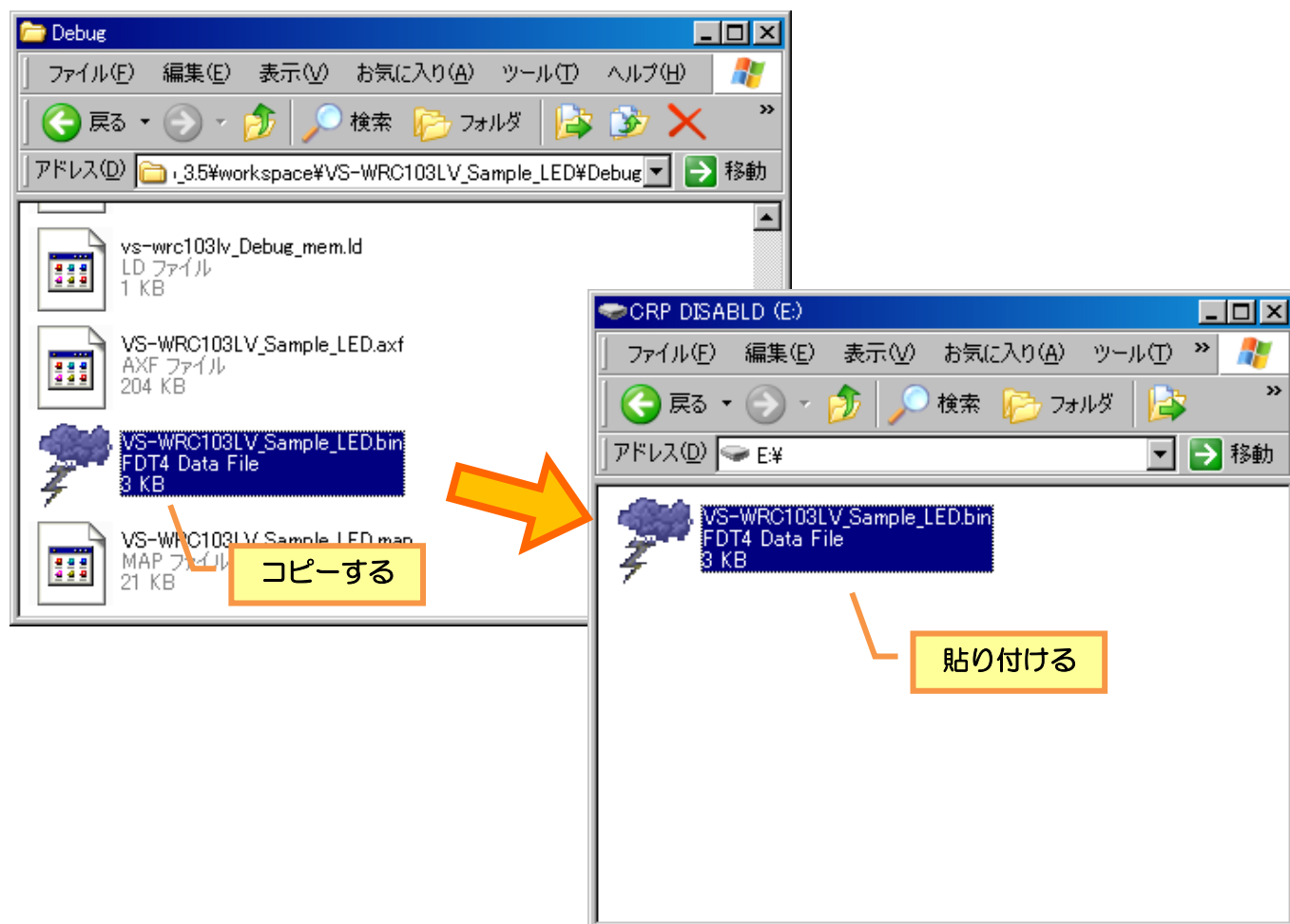
リムーバブル記憶域があるデバイス



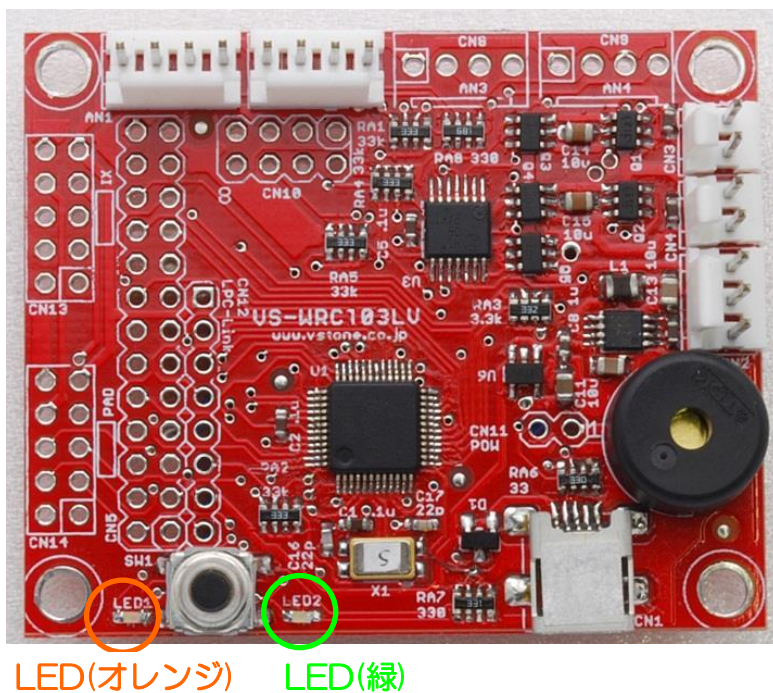
- ⑥ 「CRP DISABLE」を開くと、中に「firmware.bin」というファイルが入っていますので、このファイルを削除します。



- ⑦ ワークスペースフォルダ ¥ VS-WRC103LV_Sample_LED ¥ Debug 内の、「VS-WRC103LV_Sample_LED.bin」をコピーして、「CRP DISABLE」内に貼り付けます。



- ⑧ PC から CPU ボードを取り外し、再度電源を入れると、書き込んだプログラムに従って CPU ボードが動作します。LED サンプルプログラムの書き込みが正常に完了していると、オレンジ、緑の LED が交互に点灯します。



以上で、プログラムの書き込み手順は完了です。

6-6 LED 点滅プログラムについて

サンプルの LED 点滅プログラムは、関数 LED()、Wait()などを使って LED を制御しています。これらの関数はヘッダファイル VS-WRC103LV.h 内で定義され、プログラム内で利用することが出来ます。

LED 点滅プログラムの main()関数を以下に示します。プログラムを実行する場合、まずこの main()関数から実行されます。LED 点滅プログラムでは、main()は led.c 内にあります。

```
1: void main(void)
2: {
3:     //制御周期の設定 [単位 : Hz 範囲 : 30.0~]
4:     const unsigned short MainCycle = 60;
5:
6:     Init(MainCycle);          //CPU の初期設定
7:     //I2C_init(void); //I2C 初期化
8:
9:     //ループ
10:    while(1){
11:        LED(1);               //緑の LED 点灯
12:        Wait(1000);           //1000msec 待つ
13:        LED(2);               //オレンジの LED 点灯
14:        Wait(1000);           //1000msec 待つ
15:    }
16: }
```

main 関数の各行について説明します。

0 : 関数の宣言

4 ~ 7 : 各機能の初期設定

8 : メインループ

while 文で書かれた無限ループ内に実行したい処理を記述します。

9 ~ 12 : 実行する処理

LED()関数と Wait()関数を利用して LED を交互に点滅させています。

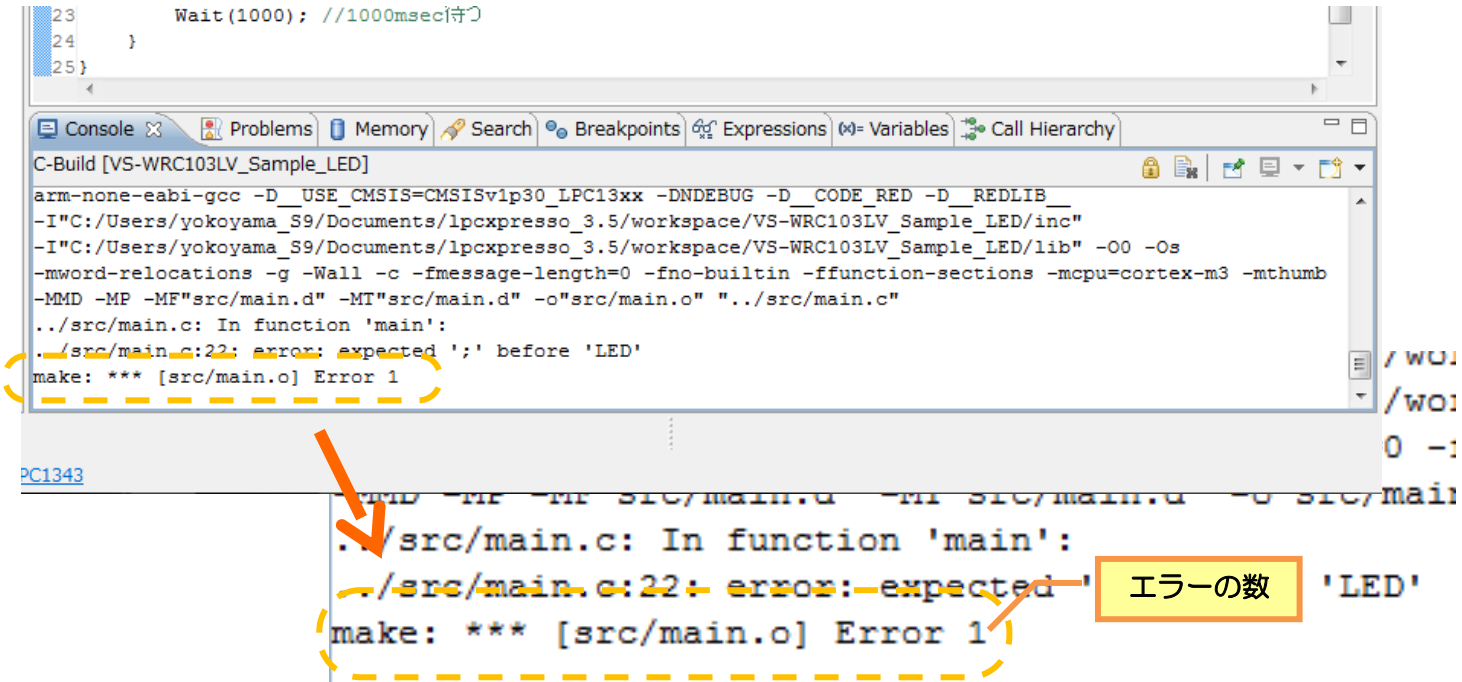
各関数については、本マニュアルの 5 章、またはサンプルプロジェクト内のヘッダファイル「vs-wrc103lv.h」で簡単に解説していますので、そちらをご覧ください。

6-7 エラーの修正

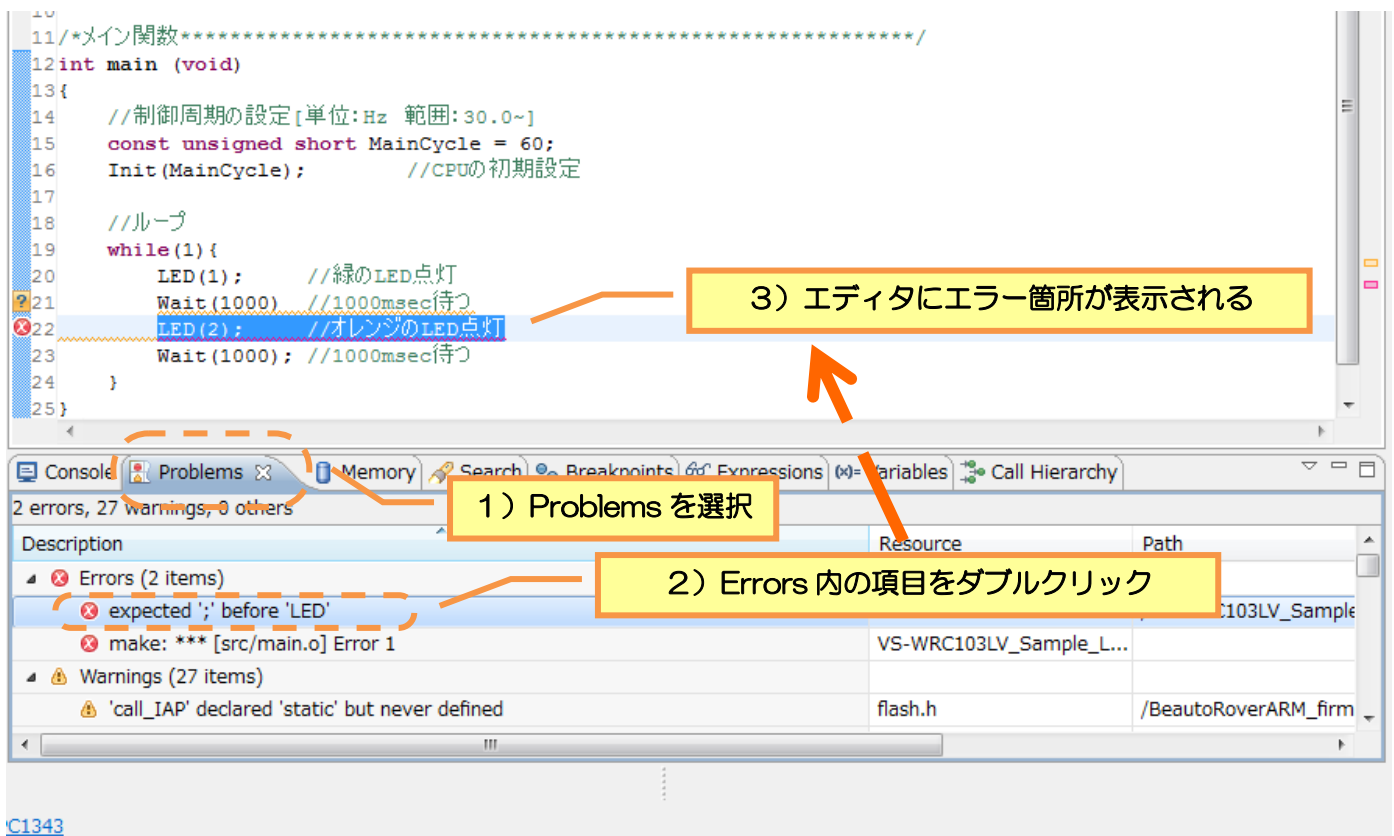
サンプルプロジェクトのプログラムを変更し、ビルドすると構文の間違いなどでビルドエラーが発生することがあります。その場合、以下の手順で、エラーの発生した箇所を確認し、修正してください。

①プログラムにエラーがあった場合、コンソールに以下のようなビルド結果が表示されます。

Error の後の数値は、いくつエラーがあったかその数が表示されます。



②エラーの位置を確認するためには、「Problems」タブを選択し、Errors 内のいずれかのエラーをダブルクリックすると、エディタ上で、エラー箇所が表示されます。



③ここでのエラーは、LED(2);の前のWait(1000);に「; (セミコロン)」がないことによるものですので、そこを修正して再度ビルドします。

```
12 int main (void)
13 {
14     //制御周期の設定[単位:Hz 範囲:30.0~]
15     const unsigned short MainCycle = 60;
16     Init(MainCycle);          //CPUの初期設定
17
18     //ループ
19     while(1){
20         LED(1);               //緑のLED点灯
21         Wait(1000);           //1000msec待つ
22         LED(2);               //オレンジのLED点灯
23         Wait(1000);          //1000msec待つ
24     }
25 }
```

④エラーがなくなり、正常にビルドが完了すると以下のように表示されます。

```
12 int main (void)
13 {
14     //制御周期の設定[単位:Hz 範囲:30.0~]
15     const unsigned short MainCycle = 60;
16     Init(MainCycle);          //CPUの初期設定
17
18     //ループ
19     while(1){
20         LED(1);               //緑のLED点灯
21         Wait(1000);          //1000msec待つ
22         LED(2);               //オレンジのLED点灯
23         Wait(1000);          //1000msec待つ
24     }
25 }
```

Console Problems Memory Search Breakpoints Expressions Variables Call Hierarchy

C-Build [VS-WRC103LV_Sample_LED]

arm-none-eabi-size VS-WRC103LV_Sample_LED.axf; arm-none-eabi-size VS-WRC103LV_Sample_LED.axf;
arm-none-eabi-objcopy -O binary VS-WRC103LV_Sample_LED.axf VS-WRC103LV_Sample_LED.bin; checksum
VS-WRC103LV_Sample_LED.bin;

	text	data	bss	dec	hex	filename
	2753	0	490	3243	cab	VS-WRC103LV_Sample_LED.axf
	text	data	bss	dec	hex	filename
	2753	0	490	3243	cab	VS-WRC103LV_Sample_LED.axf

Smart Insert 23 : 33

[PC1343](#)

以上で、エラーの修正は完了です。エディタに表示した箇所に関連する箇所で問題が起こっている可能性がありますので、その点にも注意してエラーを修正しましょう。

■オプションパーツ、関連商品のご購入は・・・

No.1 の品揃え！ ビュートローバーのオプション、ロボット関連製品のご購入はコチラ

<http://www.vstone.co.jp/robotshop/>

東京、福岡の各ロボットセンター店頭でもロボット関連商品をお買い求めいただけます。

ロボットセンター東京秋葉原店（東京支店）

〒101-0021

東京都千代田区外神田 1-9-9 内田ビル 4F

TEL：03-3256-6676

FAX：03-3256-6686

営業時間： 平日：10:30～20:00

土日祝：10:30～19:00

年末年始を除き、年中無休

ロボットセンターロボスクエア店(福岡支店)

〒814-0001

福岡市早良区百道浜 2-3-2

TNC 放送会館 2F ロボスクエア内

TEL：092-821-4111

営業時間： 9:30～18:00

定休日：第2水曜日、年末年始

商品に関するお問い合わせ

TEL: 06-4808-8701

FAX: 06-4808-8702

E-mail: infodesk@vstone.co.jp

受付時間：9:00～18:00（土日祝日は除く）

ヴイストーン株式会社

〒555-0012 大阪市西淀川区御幣島 2-15-28

www.vstone.co.jp