



HEW 開発環境の導入と サンプルプログラムのビルド

本項説明書は、「VS-WRC003」を C 言語で開発する際に使用可能な、ルネサスエレクトロニクス社より無料配布されている“High-performance Embedded Workshop”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。

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 プログラム書き込みソフトウェアのダウンロード、インストール
- ④ プログラムの書き込み

目次

1 開発環境の入手	3
2 HEW のインストール.....	6
3 HEW の起動とサンプルプロジェクトの読み込み	15
4 インクルードファイルディレクトリの設定と、ヘッダファイルの編集.....	17
5 VS-WRC003 へのプログラムの書き込み	23
6 LED 点滅プログラム	31
7 トラブル対応：FDT を使用した VS-WRC003 へのプログラム書き込みが失敗する場合	34

1 開発環境の入手

High-performance Embedded Workshop (以下 HEW) とフラッシュ開発ツールキット (以下 FDT) は、ルネサスエレクトロニクスのホームページ (下記の URL) からダウンロードします。(ルネサスへのユーザ登録が必要になります)

http://www.vstone.co.jp/products/beauto_chaser/hew_download

(1) リンク先にある、以下のソフトウェアを選択します。

HEW: 【無償評価版】H8SX,H8S,H8 ファミリー用 C/C++コンパイラパッケージ V.7.00 Release 00 以降

FDT: 【無償評価版】フラッシュ開発ツールキット V.4.02 Release 01 以降

M3T-NC308WA	【無償評価版】NC308WA V.5.42 Release 00	Apr.03.10	High-Performance Embedded Workshopおよびシミュレータデバッグを同梱。
SuperHコンパイラパッケージ	【無償評価版】SuperHファミリー用 C/C++コンパイラパッケージ V.9.03 Release 02	Oct.28.09	無償評価版です。High-Performance Embedded Workshopおよびシミュレータデバッグを同梱。
RXコンパイラパッケージ	【無償評価版】RXファミリー用 C/C++コンパイラパッケージ V.1.00 Release 00	Oct.05.09	無償評価版です。Windows Vista®, Windows® XP, Windows® 2000にのみインストールできます。High-Performance Embedded Workshopおよびシミュレータデバッグを同梱。
MISRACルールチェッカ SQMlint	【無償評価版】MISRA Cルールチェッカ SQMlint V.1.03 Release 00A	Sep.07.09	無償評価版です。
H8SX,H8S,H8コンパイラパッケージ	【無償評価版】H8SX,H8S,H8ファミリー用 C/C++コンパイラパッケージ V.7.00 Release 00	Sep.07.09	無償評価版です。High-Performance Embedded Workshopおよびシミュレータデバッグを同梱。
M3T-CC32R	【無償評価版】M32Rファミリー用 C/C++コンパイラパッケージ M3T-CC32R V.5.01 Release 01	May.08.09	High-Performance Embedded Workshopを同梱。シミュレータデバッグはコンパイラパッケージとは別にダウンロードおよびインストールが必要です。

(2) 規約に同意して「同意する」(Submit) を押します。

流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断して下さい。弊社は、適用可否に対する責任を負いません。

7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません(弊社が自動車用と指定する製品を自動車に使用する場合を除きます)。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。

8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないで下さい。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。

1) 生命維持装置。 2) 人体に埋め込み使用するもの。 3) 治療行為(急部切り出し、薬剤投与等)を行なうもの。 4) その他、直接人命に影響を与えるもの。

9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて製品をご使用された場合の故障および事故につきましては、弊社はその責任を負いません。

10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。弊社製品の故障または誤動作が生じた場合も人身事故、火災事故、社会的損害などを生じさせないよう、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計などの安全設計(含むハードウェアおよびソフトウェア)およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特にマイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願い致します。

11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなまよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。

12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断り致します。

13. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会下さい。

上記事項に

© 2010 Renesas Electronics Corporation. All rights reserved. World Renesas | ご利用に際して | 個人情報保護 | RSS | サイトマップ

(3) [Download]を押します。

【無償評価版】H8SX,H8S,H8ファミリ用C/C++コンパイラパッケージ V.7.00 Release 00

製品概要

H8SX, H8S, H8ファミリ用C/C++コンパイラパッケージの製品概要については [データシート](#) を参照してください。

無償評価版の注意事項

● 使用期限と使用制限について

本無償評価版の使用期限と使用制限については、[無償評価版コンパイラ](#) のページを参照してください。なお、評価期間の起算日は、最初に無償評価版コンパイラを実行した日となります。旧バージョンの無償評価版コンパイラをリビジョンアップされても起算日は変更されませんので、予めご了承ください。

● 動作環境について

ホストマシン名	OS名	ハードディスク容量
IBM PC/AT ※1 互換機	※2 Windows Vista®, Windows® XP, Windows® 2000	240Mバイト以上の空き容量が必要で、

※1. IBM, ATは米国International Business Machines Corporationの登録商標です。

※2. Windows, Windows Vistaは米国Microsoft Corporationの登録商標です。

インストール方法

ダウンロードしたファイルを実行してください。インストーラーが起動します(作業を行うディレクトリは、十分な容量のあるドライブをご使用ください)。その後の作業は、インストーラーの指示に従ってください。

無償評価版ソフトウェアご使用にあたって

- [使用権許諾契約書](#) を必ずお読みください。

ダウンロード

ダウンロード製品名	ファイル名	ファイルサイズ	リンク
【無償評価版】H8SX,H8S,H8ファミリ用C/C++コンパイラパッケージ V.7.00 Release 00	h8v7000_ev.exe	102,344,968 bytes (97.60 Mbytes)	Download

(4) ユーザ登録した ID を入力して送信を押すと、HEW のインストーラがダウンロードできますので、任意の場所（デスクトップなど）に保存してください。



ログイン

処理を続ける為にログインしてください。

メールアドレス

パスワード

取消し/閉じる

送信

パスワードをお忘れの場合[ここをクリック](#)

[My Renesasのご登録がまだのお客様](#)

※FDT も同様にダウンロードしてください。

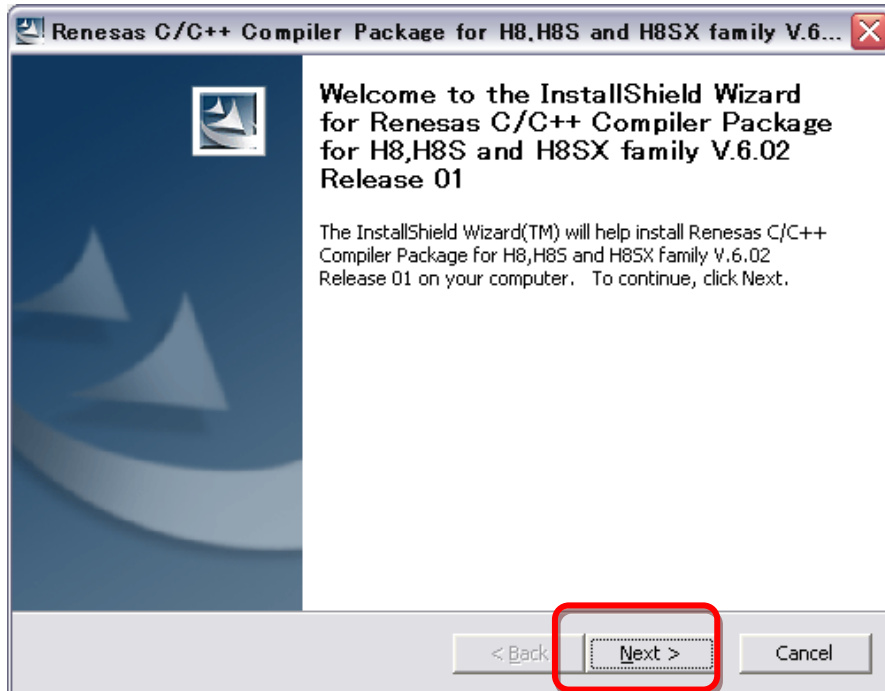
2 HEW のインストール

(1)ダウンロードしたインストーラ[h8v****_ev.exe]を起動します。

(****にはバージョンの数字が入ります)

(2)NEXT を押します。

※以降の画面は V.6.02 Release 01 のものであり、最新のバージョンでは表示が異なる可能性があります。



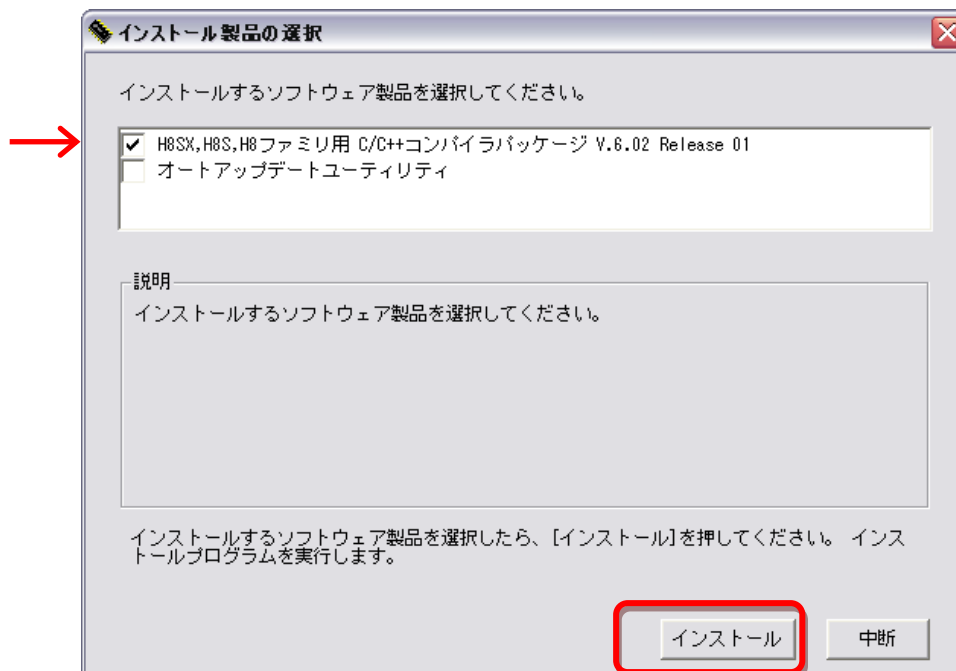
(3)パッケージを展開する任意のフォルダを指定して[Next]を押します。



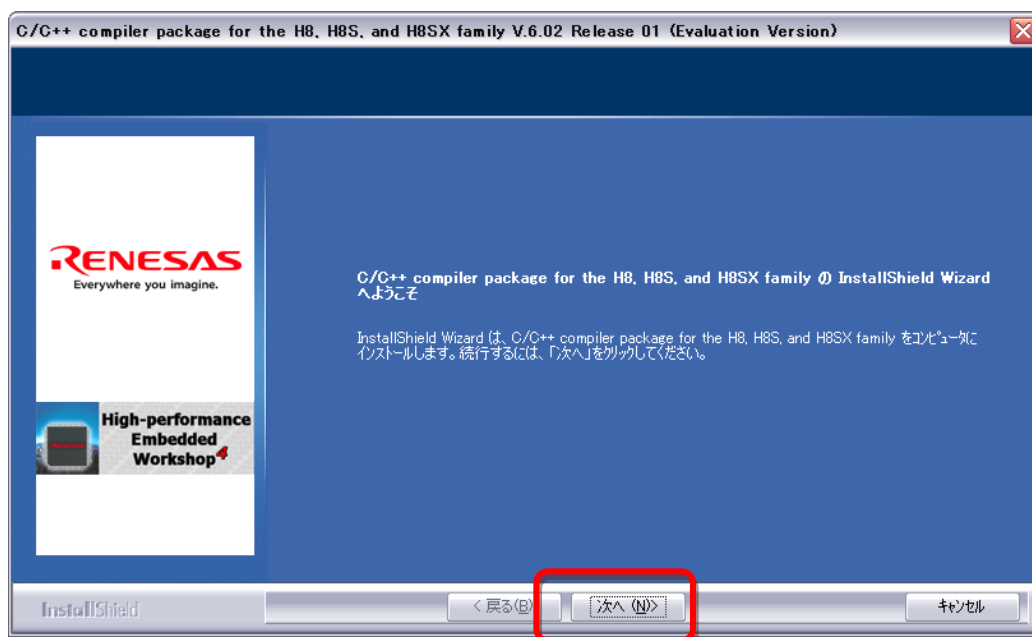
(4) [標準インストール]を押します。



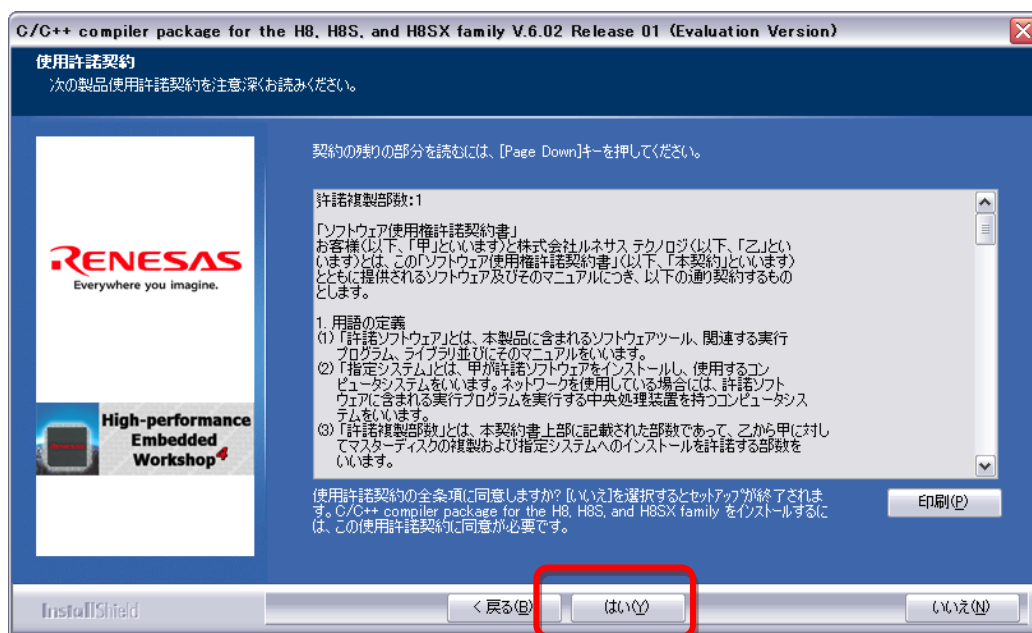
(5) チェックボックスを下の画像と同じように選択し、[インストール]を押します。



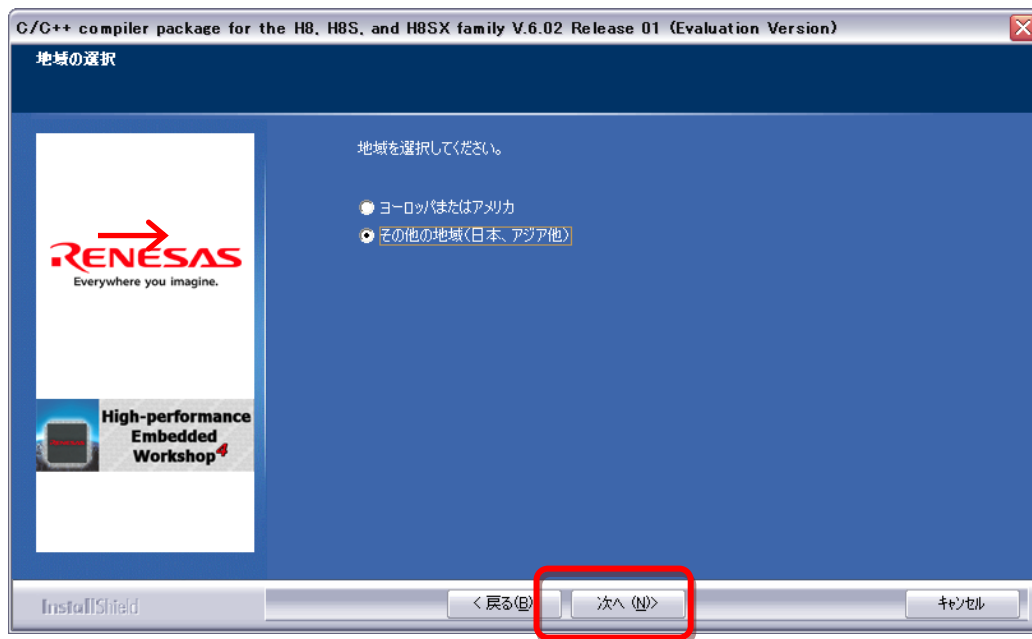
(6) [次へ]を押します。



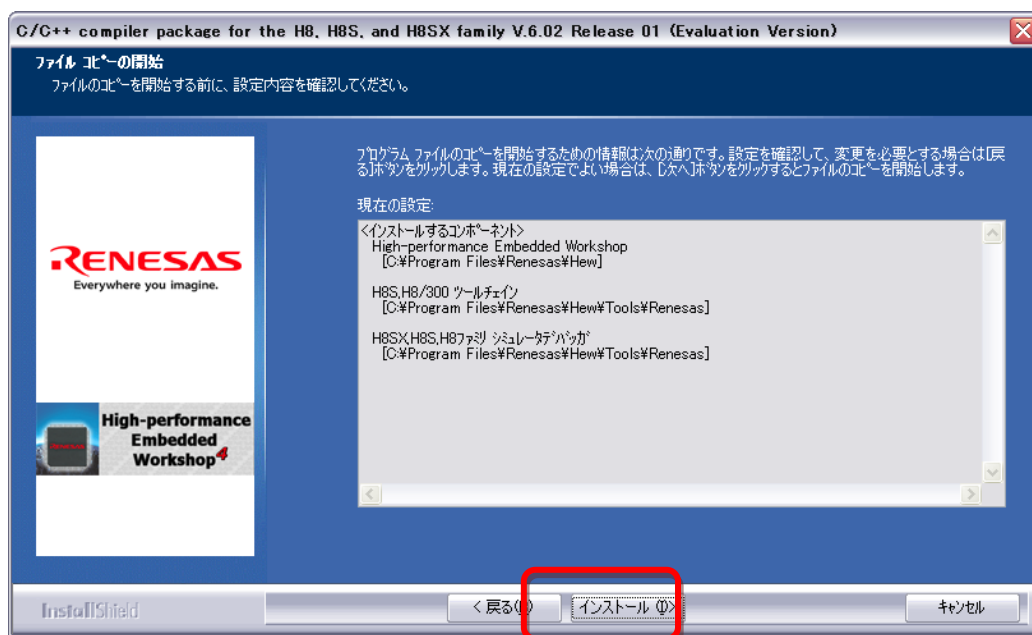
(7) [はい]を押します。



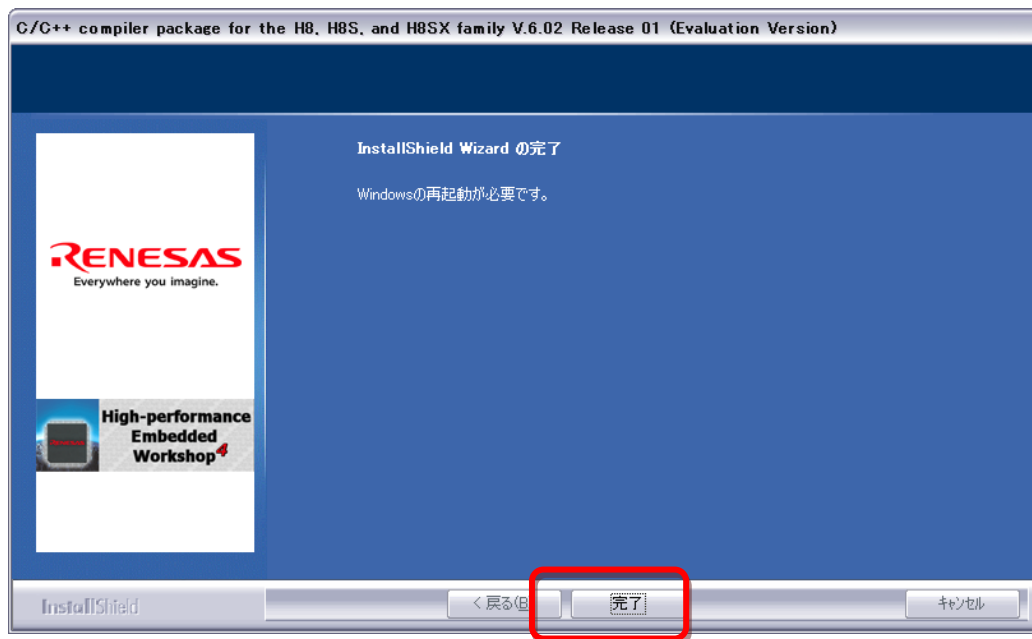
(8) [その他の地域（日本、アジア他）] をチェックし、[次へ] を押します。



(9) [インストール] を押します。

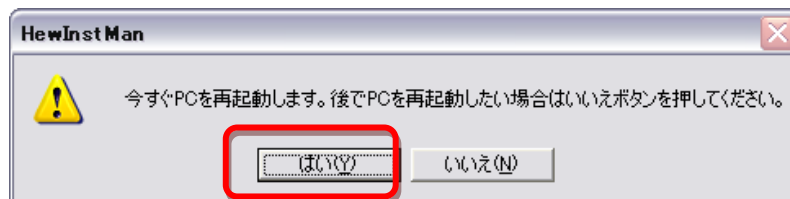


(10) [完了]を押します。



(11) [はい]を押してPCを再起動します。

(PCの環境によって以下のメッセージボックスが出ない場合があります。)



3. FDT のインストール

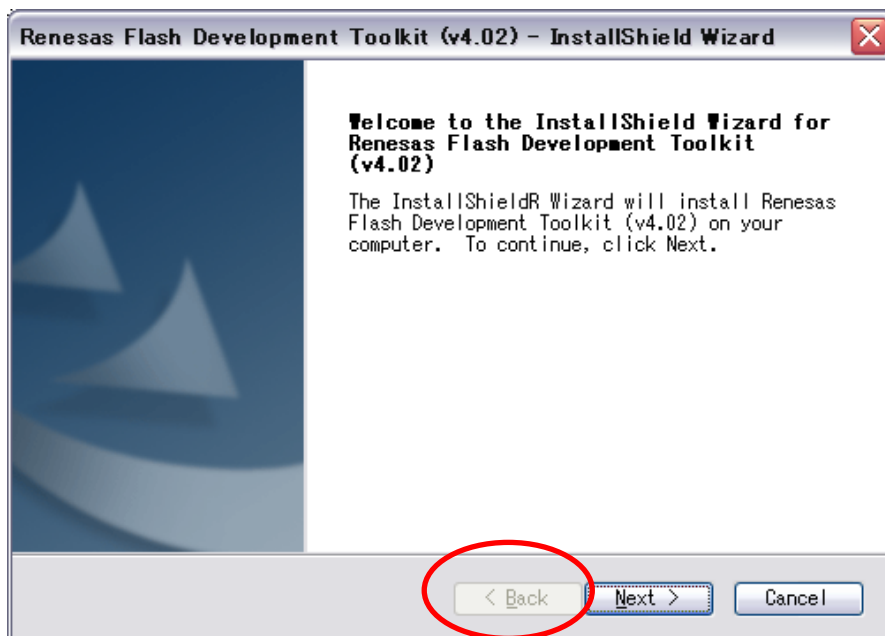
(1) ダウンロードしたインストーラ[fdtv***r**.exe]を起動します。

(※1: ***にはバージョン、**にはリリースの番号が入ります)

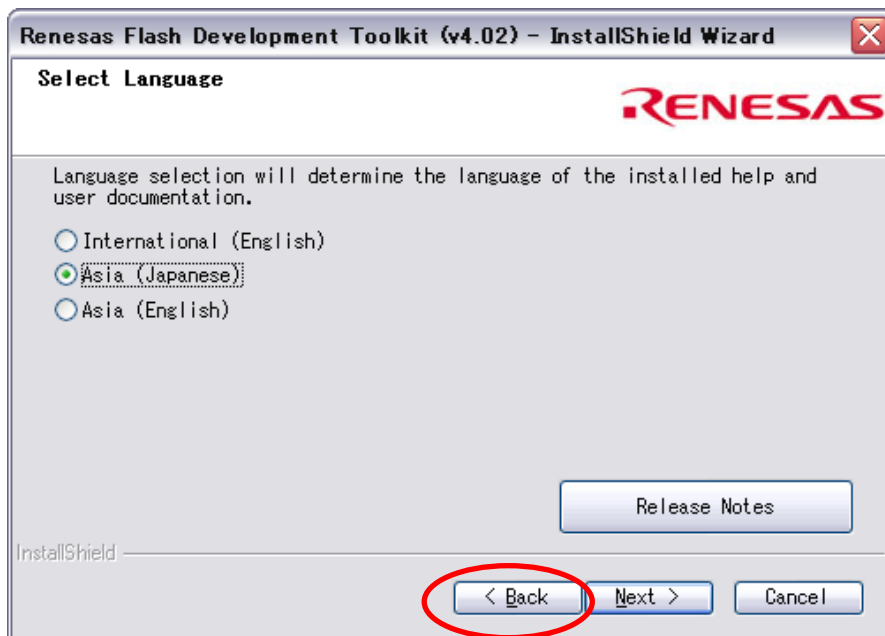
(※2: 起動時に無償評価版の確認画面が表示されることがあります。その場合、OK を押すことで、(2)の画面が表示されます)

(2) [Next]を押します。

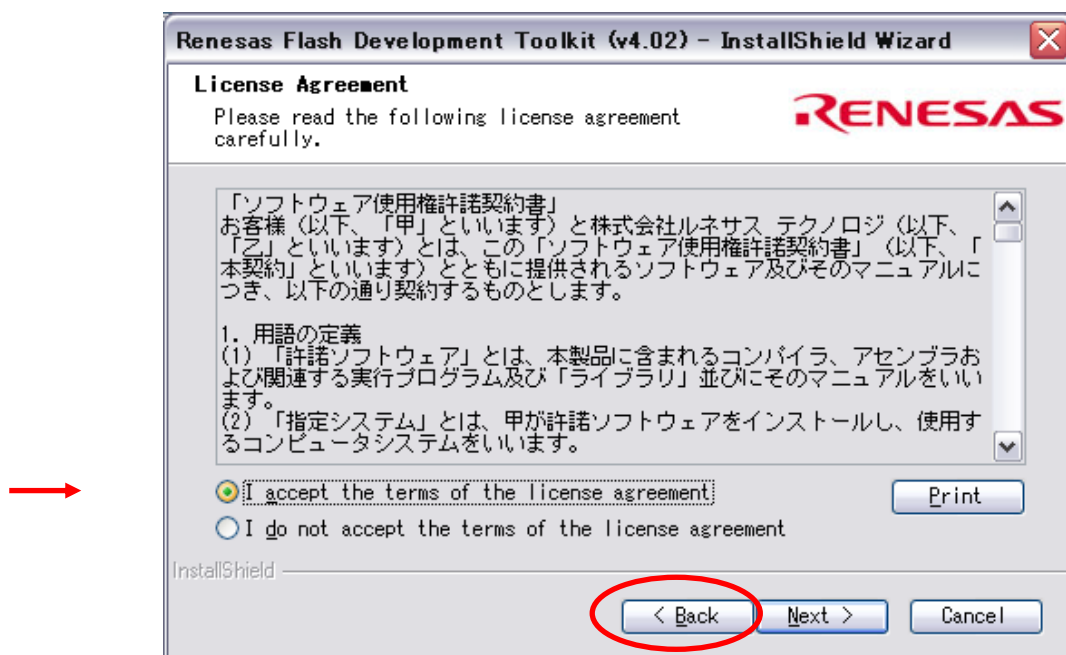
※以降の画面は V.4.02 Release 01 のものであり、最新のバージョンでは表示が異なる可能性があります。



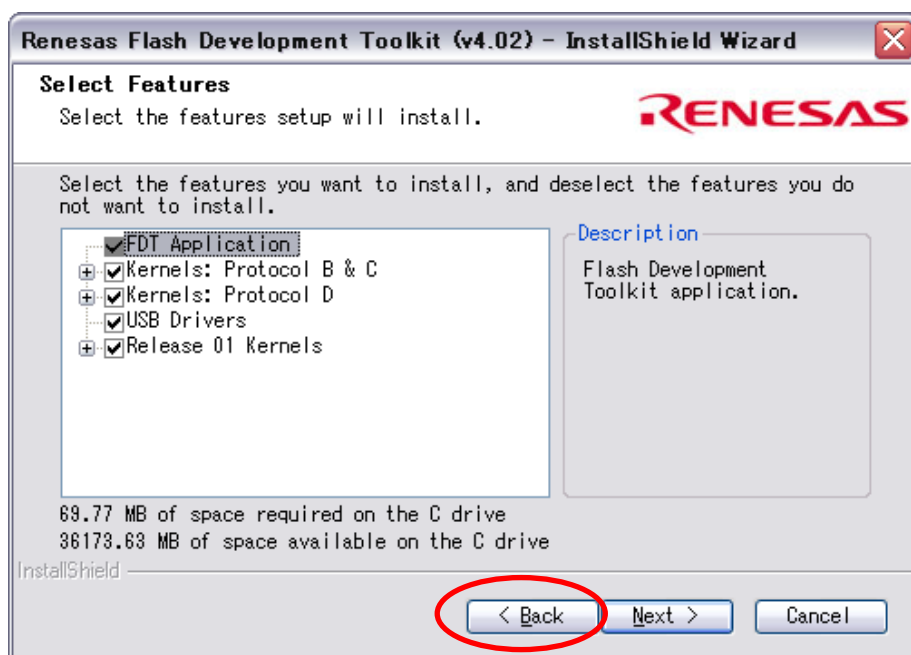
(3) [Asia(Japanese)]を選択し、[Next]を押します。



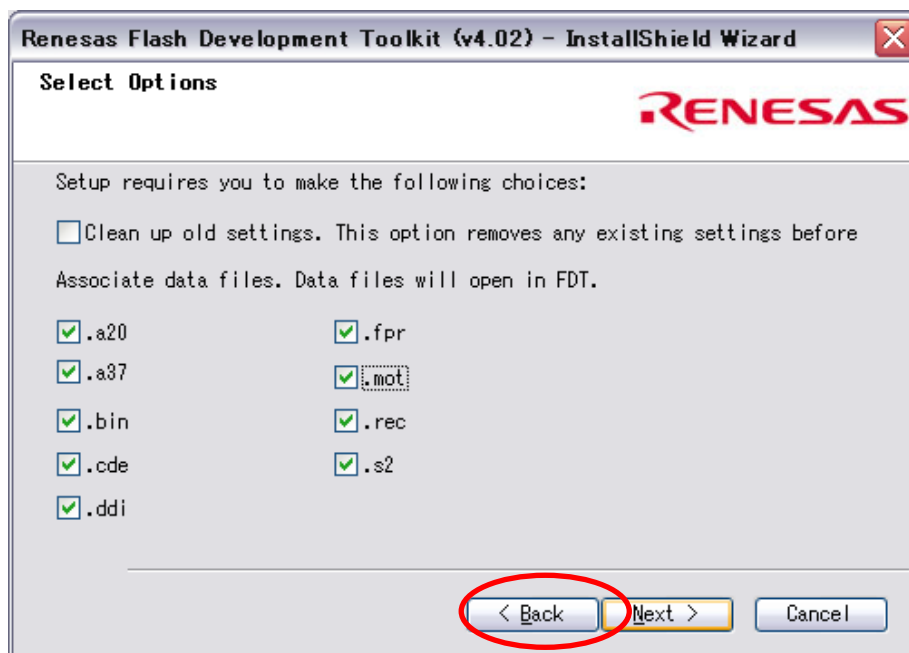
(4) 規約に同意し[Next]を押します。



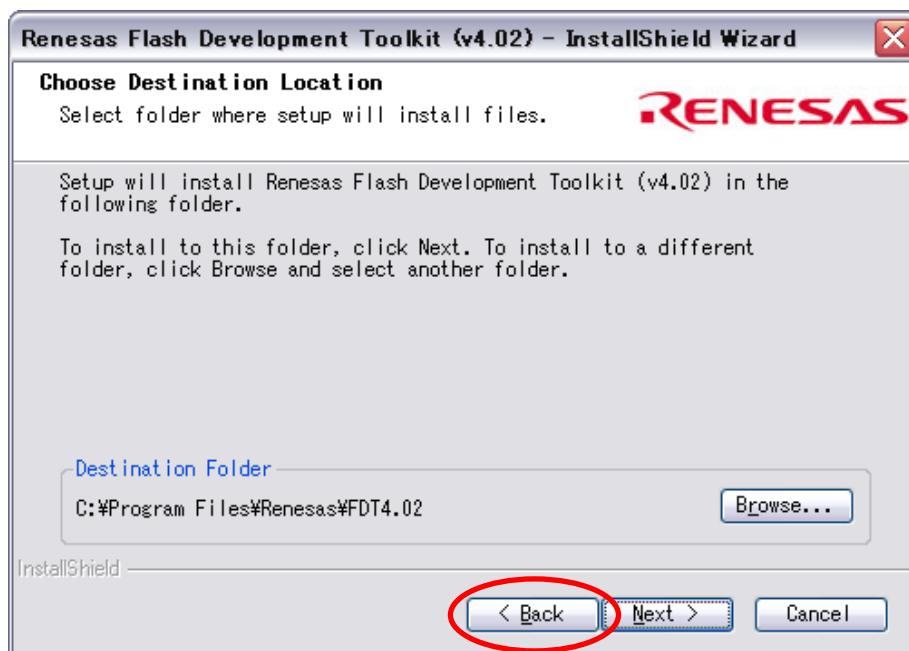
(5) 下のようにすべてにチェックをつけ、[Next]を押します。



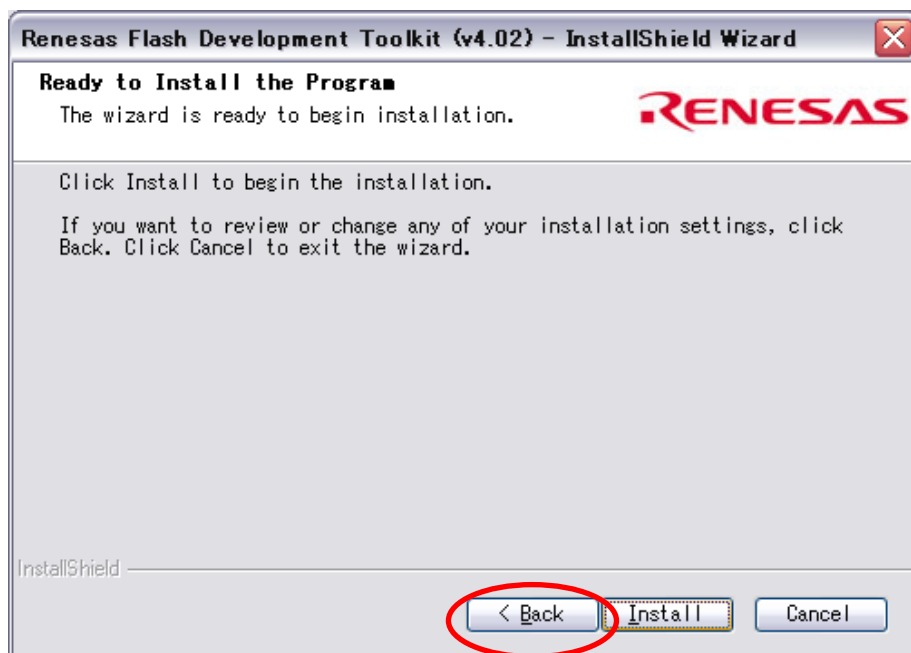
(6) [Next]を押します。



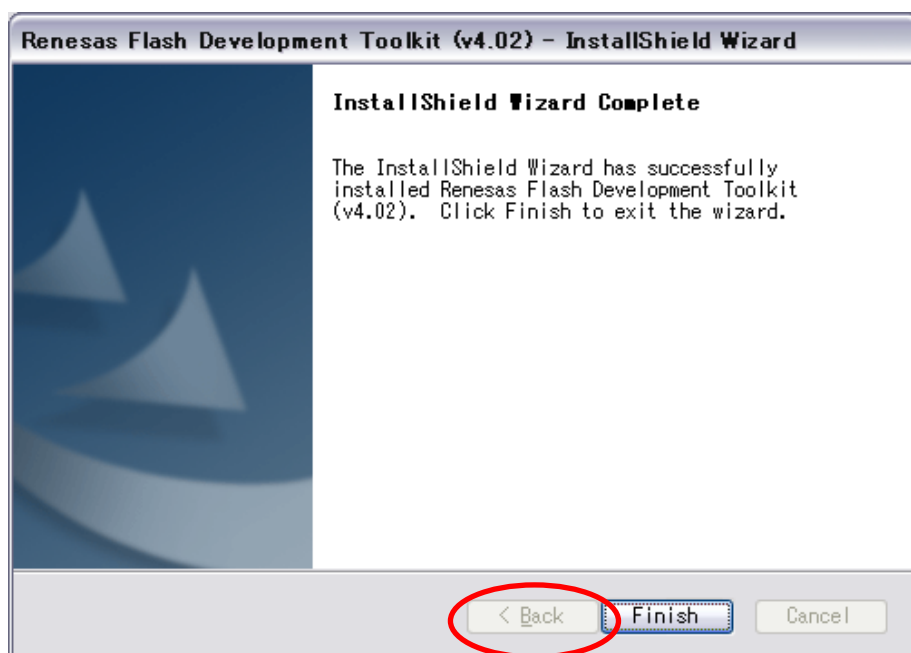
(7) インストール先のフォルダを指定し、[Next]を押します。



(8) [Install] を押し、インストールを開始します。



(9) [finish] を押します。



3 HEW の起動とサンプルプロジェクトの読み込み

- (1) サンプルプロジェクト「WRC003_SampleProject_LED_*****.zip」(**は日付)を弊社 Web ページからダウンロードします。以下のページよりダウンロードできます。

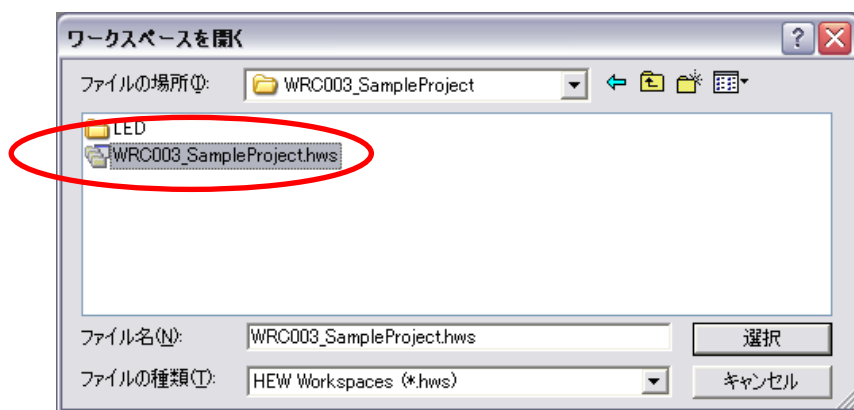
http://www.vstone.co.jp/products/vs_wrc003/download.html

- (2) ダウンロードした Zip ファイルを解凍します。

- (3) デスクトップのショートカット、または、Windows の[スタートメニュー]から
[プログラム]>[Renesas]>[High-performance Embedded Workshop] >[High-performance Embedded Workshop]を起動します。

(※起動時に“ようこそ” 画面が表示されますが、キャンセルしてください)

- (3) メニューの[ファイル]から[ワークスペースを開く]を選択し、解凍したサンプルプロジェクトフォルダ内にある[WRC003LV_SampleProject.hws]を開きます。

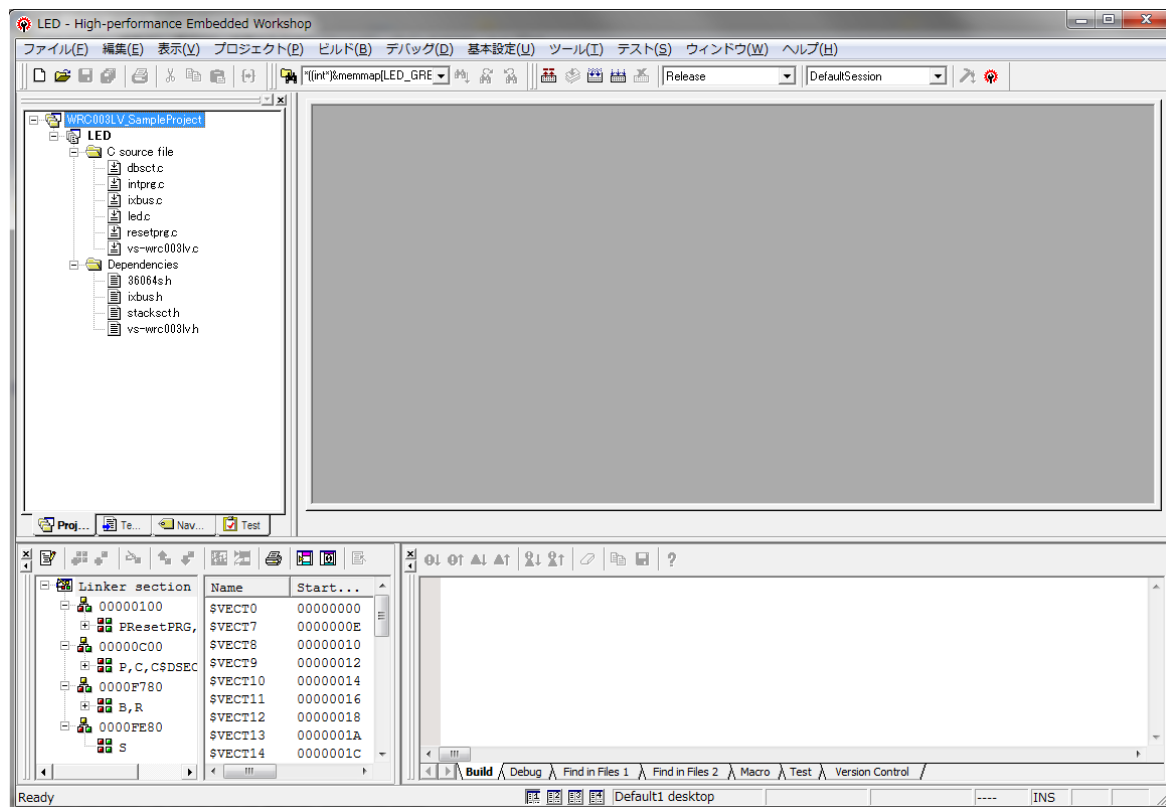


※プロジェクトを開く際に、HEW のバージョンによってツールチェーンが異なる場合があります、その場合、変更を求められることがあります。

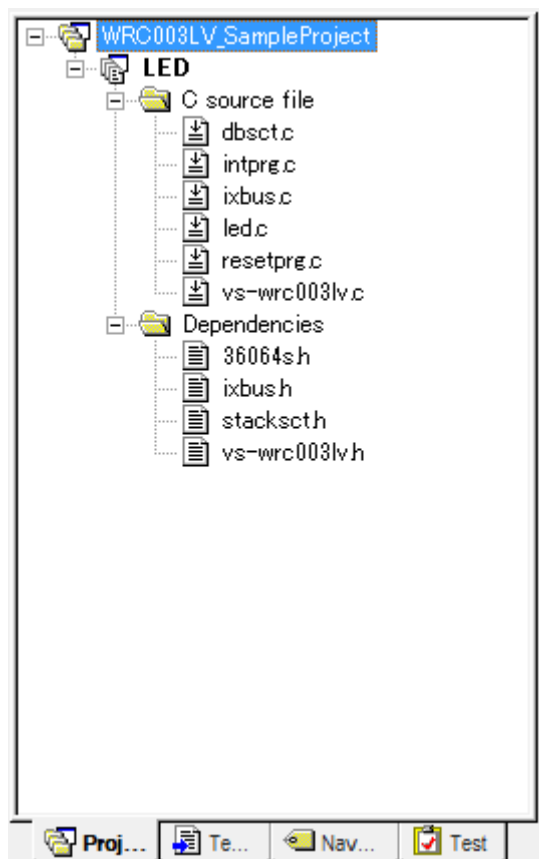
この表示がされた場合、必ずすべて「OK」を押してください。 OK 以外を押すとビルドが正常にできなくなります。



(4) プロジェクトを開くと以下のような画面が表示されます。



(5) プロジェクトには以下のファイルが含まれています。
このとき、36064.h は表示されない場合があります。



※複数のプロジェクトがある場合、サンプルのワークスペース内の[LED]を右クリックし、プロジェクトをアクティブにします。

4 インクルードファイルディレクトリの設定と、ヘッダファイルの編集

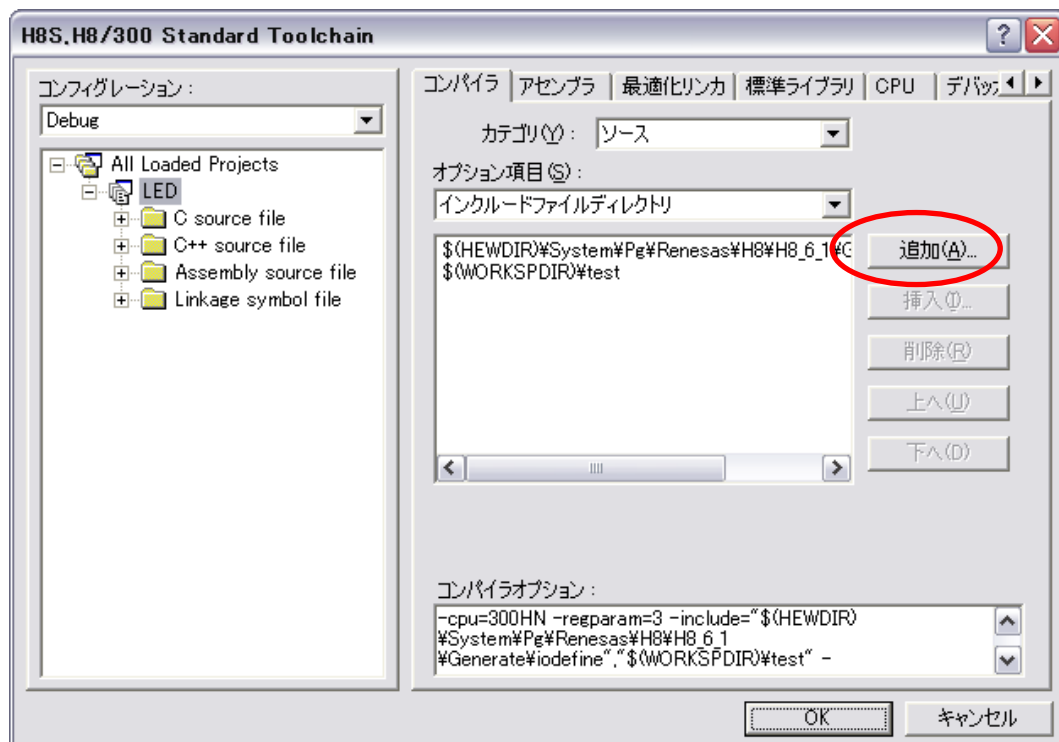
- (1) 外部から持ってきたプロジェクトそのままでは、必要なヘッダファイル（拡張子.h のファイル）が読み込まれていないので、インクルードファイルディレクトリを設定します。
- (2) メニューの[ビルド]内の[H8S,H8/300 Standard Toolchain..]からツールチェーン（コンパイラ、リンカなどのこと）の設定をします。

※ビルドが表示されていない場合、プロジェクト読み込み時のダイアログで OK 以外を選択したため、プロジェクトの一部に不具合が起きている可能性があります。6-3 のサンプルプロジェクトのダウンロードからやりなして、プロジェクトを読み込む際に表示されるダイアログでは、すべて OK を押してください。



- (3) ツールチェーンの設定を開くと以下のようなウィンドウが表示されます。

ここで、[コンパイラ]タブ内のカテゴリで[ソース]、オプション項目で[インクルードファイルディレクトリ]を選択し、[追加]を押します。

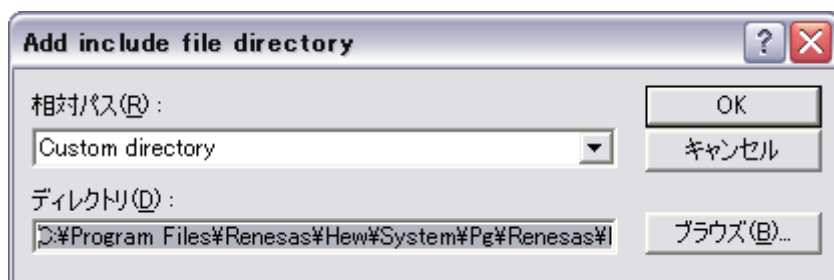


(4) インクルードディレクトリ追加用のウィンドウが開いたら、相対パスで[Custom directory]を選択し、ディレクトリに各 CPU 用のインクルードファイルがある[iodefine フォルダ](※)のパスを設定し、[OK]を押します。フォルダのパスはPCによりこととなりますので、必ず確認をしてください。

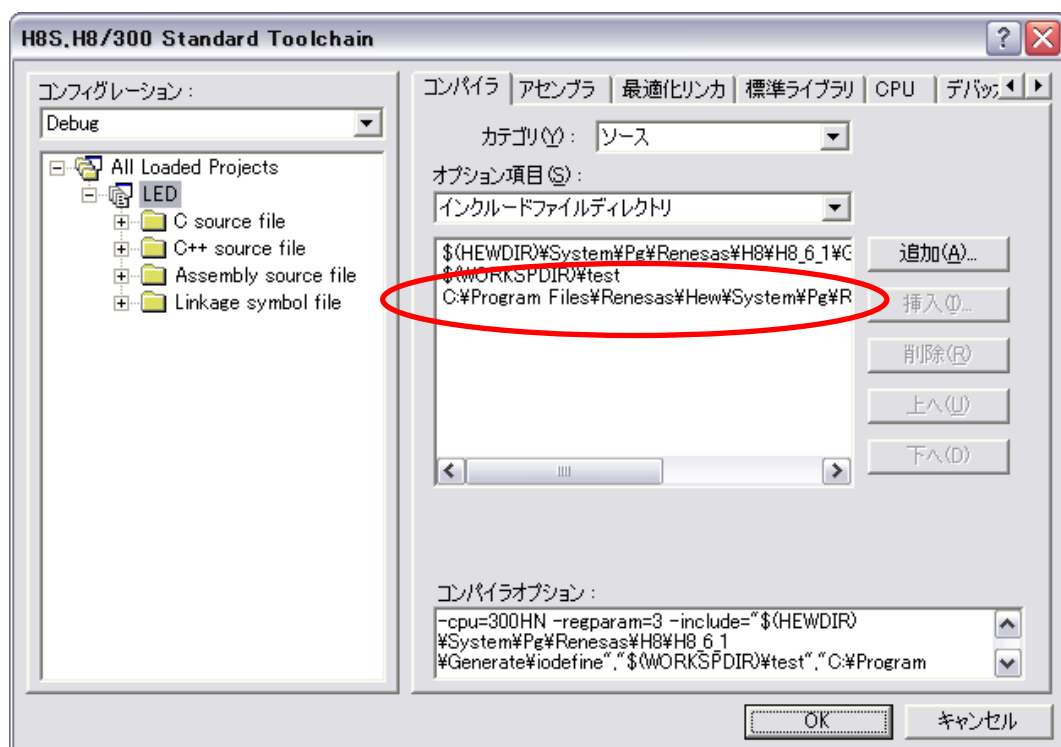
※ インストール時にインストール先のフォルダを変更していない場合、[iodefine フォルダ]は以下のようになります。(Windows が入っているドライブが C:の場合)

[C:¥Program Files¥Renesas¥Hew¥System¥Pg¥Renesas¥H8¥H8_*_*¥Generate¥iodefine]

※「*」にはツールチェーンのバージョンが入ります。

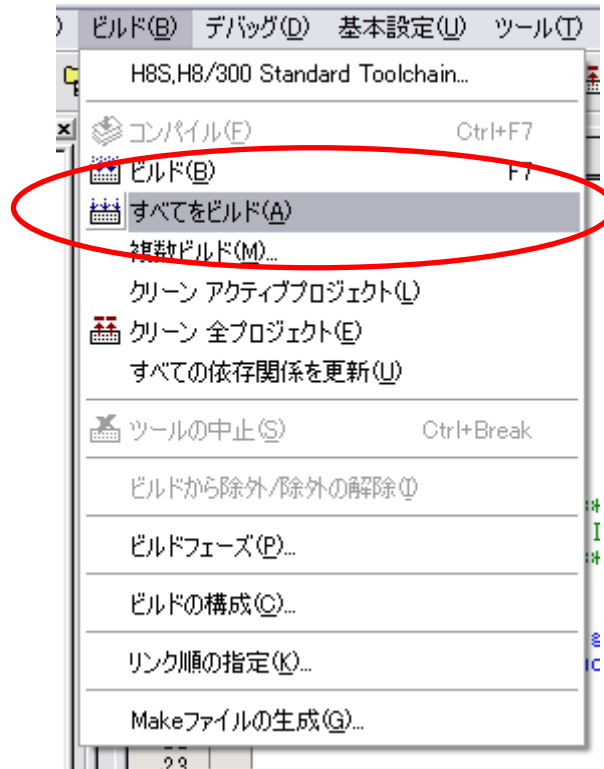


(5) 設定したフォルダが追加されていたら[OK]をおして、ツールチェーンの設定を終了します。



(6) ここで、一旦ビルドをします。ビルドはメニューの[ビルド]内の[すべてをビルド]で行えます。

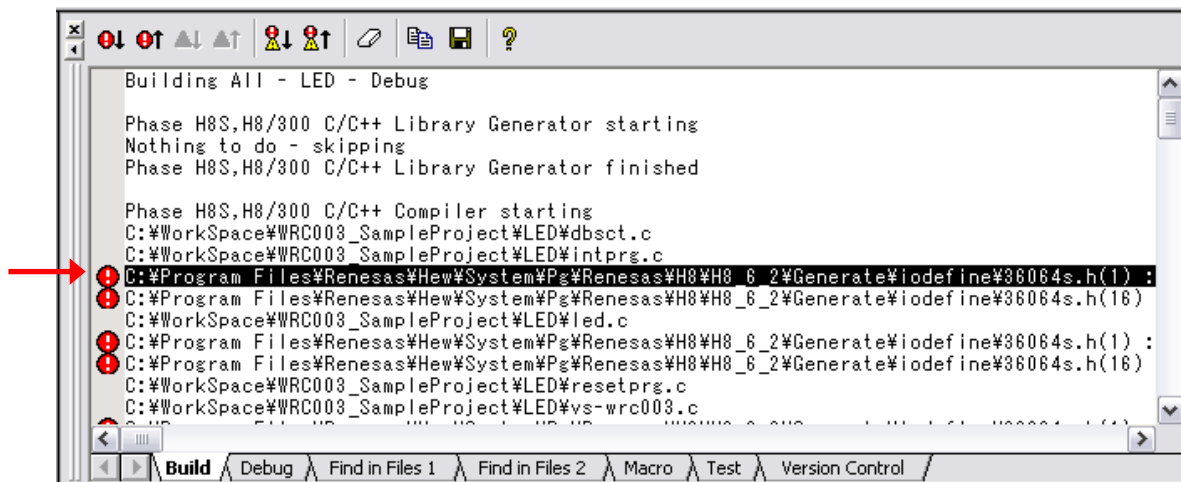
(2回目以降は[ビルド(B)]を選択 または F7 キーを押すことでもビルドできます)



(7)ビルドを行うと画面右下(アウトプットウィンド)にエラーが表示されます。

エラーの原因がどこかを探したいときは、エラーの表示されている行(赤い!マークではなく、以下の画像で黒くハイライトされている箇所)をダブルクリックすると、エディタにエラーがある部分が表示されます。

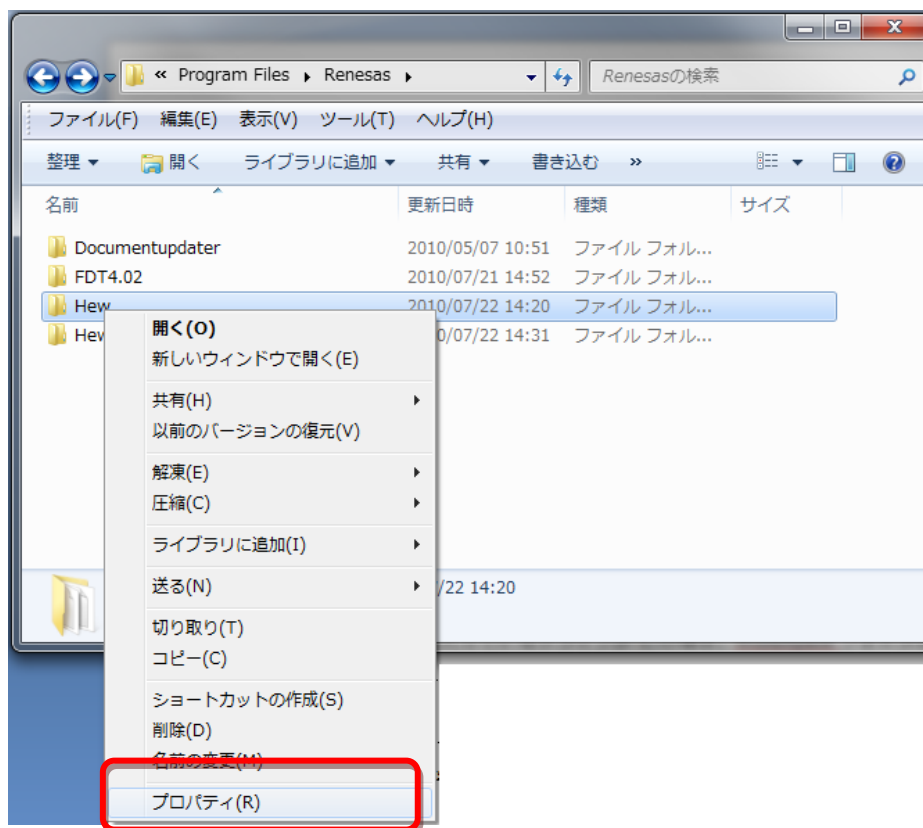
ここでは、エラーのうち[(HEW のインストールフォルダ)...\define\36064s.h(1):****] (一番初めに出力されたエラー)をダブルクリックして表示させます。



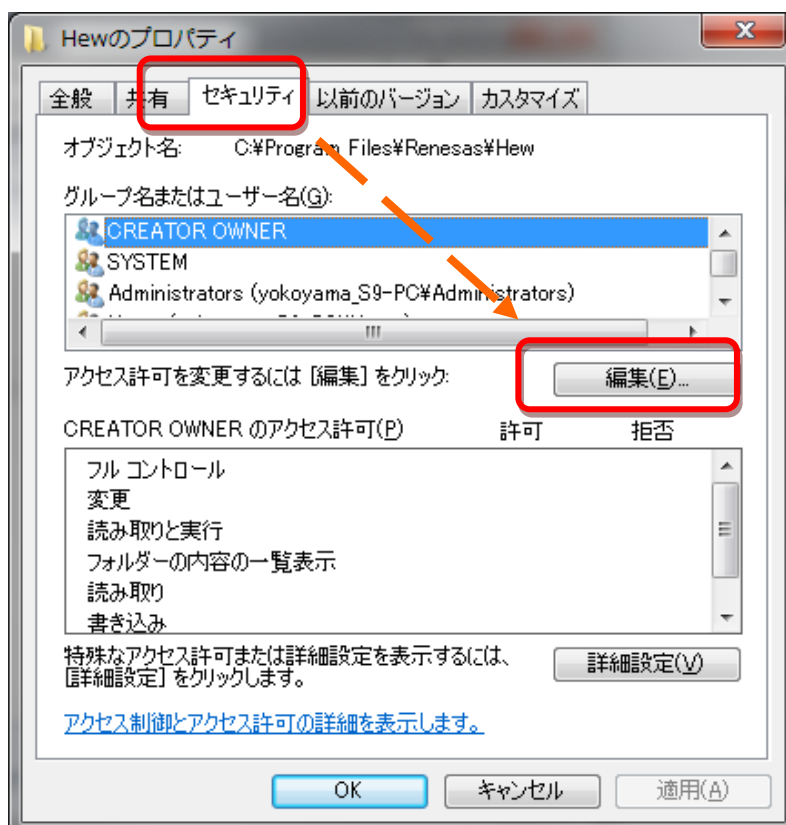
※Windows Vista または Windows7 の場合、セキュリティの都合上、Program Files フォルダ内のファイルが変更できない場合があります。以下の手順で、使用されているユーザに対して HEW フォルダに対するフルコントロールを与えるか、Program Files 以外のフォルダにインストールすることで、36064s.h が編集可能になります。

※以下の変更は、Windows のセキュリティに影響を及ぼす可能性があります。

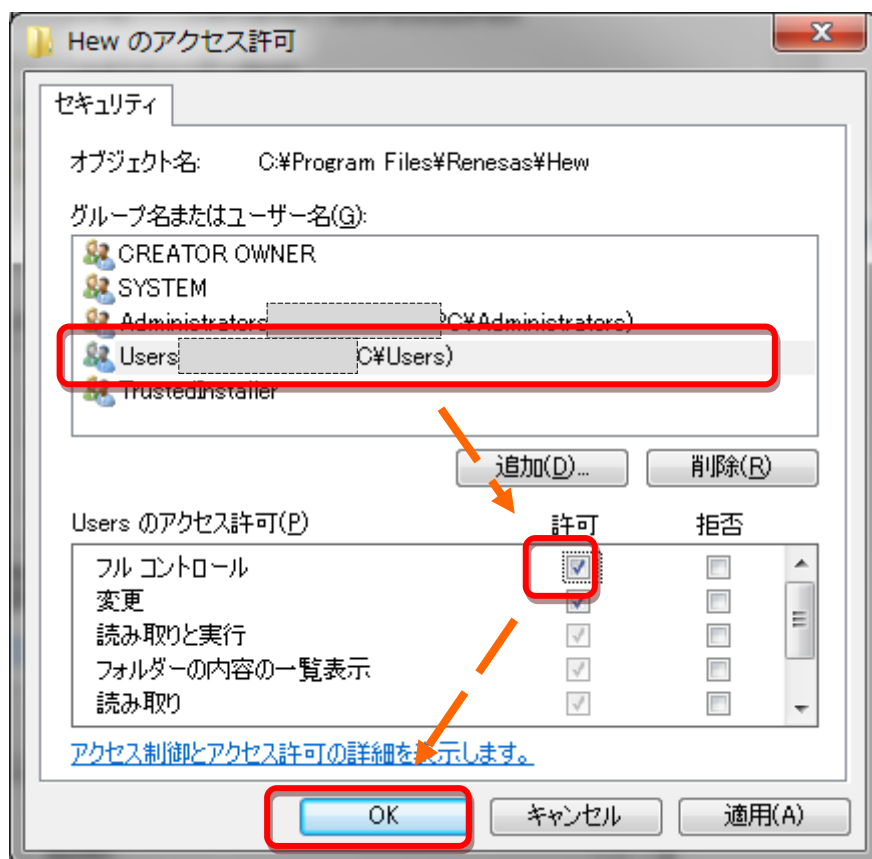
① 「C:\Program Files\Renesas\Hew」フォルダのプロパティを開きます。



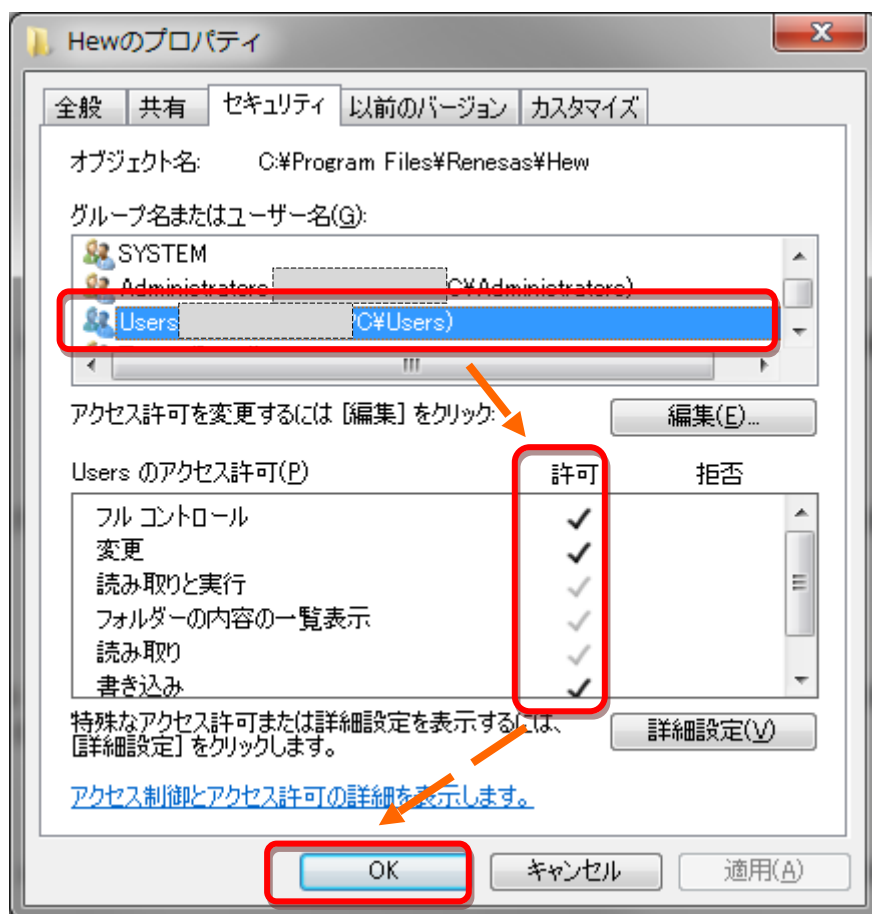
② セキュリティタブを開き、編集ボタンを押します。



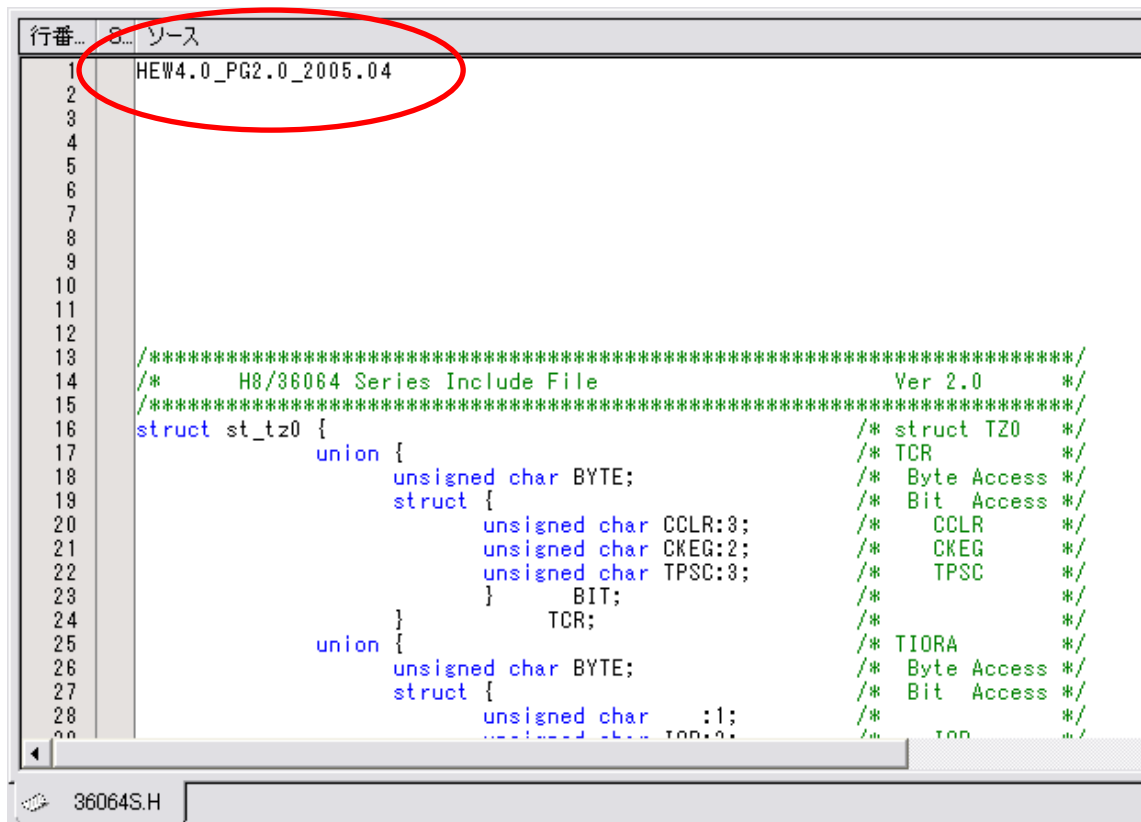
- ③ 現在使用しているユーザ（Users など）を選択し、フルコントロールの許可にチェックを入れ、OK をおします。



- ④ 現在使用しているユーザ（Users など）を選択し、フルコントロールにチェックが入っていることを確認して、OK を押します。以上で、変更は終了です。



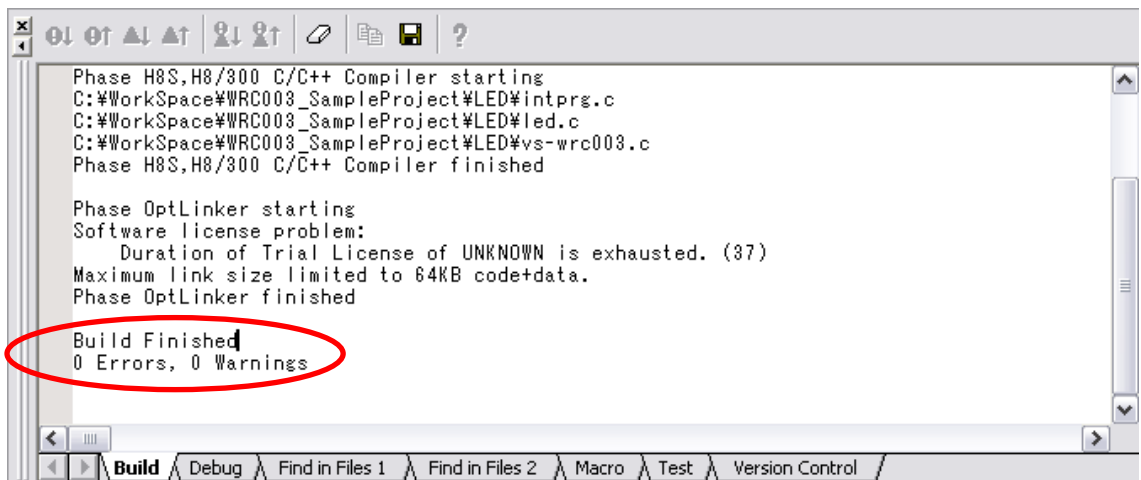
(8) エディタに 36064s.h が表示されると思います。この 1 行目にコメントアウトされていない一文がありますので、先頭に[/]/を入力（コメントアウト）するか、1 行目を削除してください。



```
行番... 8... ソース
1 HEW4.0_PG2.0_2005.04
2
3
4
5
6
7
8
9
10
11
12
13 /*****
14 /*      H8/36064 Series Include File                      Ver 2.0      */
15 /*****
16 struct st_tz0 {
17     union {
18         unsigned char BYTE;
19         struct {
20             unsigned char CCLR:3;
21             unsigned char CKEG:2;
22             unsigned char TPSC:3;
23             BIT;
24         } TCR;
25     }
26     union {
27         unsigned char BYTE;
28         struct {
29             unsigned char :1;
30             unsigned char TCR:3;
31         } TCR;
32     }
33 };
```

(9) コメントアウトし、ファイルを保存したらビルドします。

エラーがなく、ビルドが完了している場合は[Build Finished 0 Errors, 0 Warnings]と表示されます。これで、VS-WRC003 に書き込むためのファイル（.mot ファイル）が生成されました。



```
Phase H8S,H8/300 C/C++ Compiler starting
C:\Workspace\WRC003_SampleProject\LED\intprg.c
C:\Workspace\WRC003_SampleProject\LED\led.c
C:\Workspace\WRC003_SampleProject\LED\vs-wrc003.c
Phase H8S,H8/300 C/C++ Compiler finished

Phase OptLinker starting
Software license problem:
Duration of Trial License of UNKNOWN is exhausted. (37)
Maximum link size limited to 64KB code+data.
Phase OptLinker finished

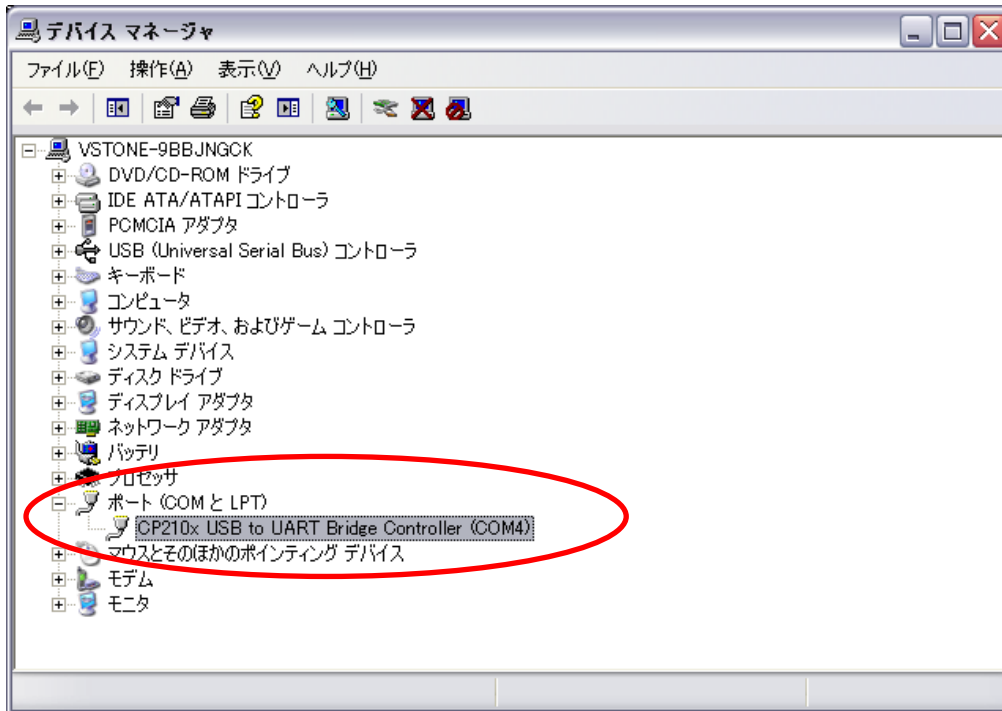
Build Finished
0 Errors, 0 Warnings
```

5 VS-WRC003 へのプログラムの書き込み

(1) VS-WRC003(以下 CPU ボード)には USB シリアル変換 IC が内蔵されています。ですので、書き込みには COM ポートを使用します。まず、CPU ボードと PC を USB ケーブルで接続します。この状態で、**デバイスマネージャ**(※1)を開き **COM ポートの番号**を確認します。(下の画像では COM4) COM9 以降の場合は 19 ページの記載に従って番号を COM8 より小さいものに変更してください。

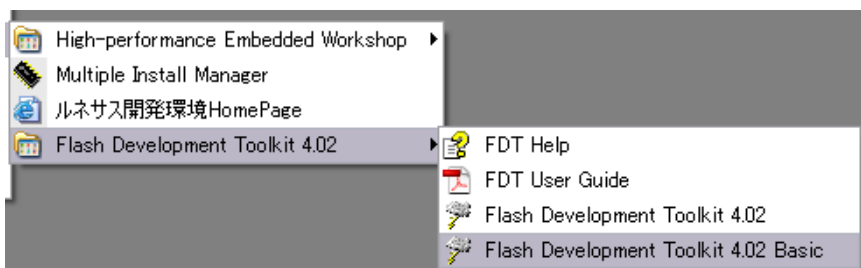
(※1：デバイスマネージャはマイコンピュータのプロパティで[ハードウェア]タブ内の[デバイスマネージャ]を押すと表示されます。)

(※2CP210x のドライバがインストールされていない(デバイスマネージャで以下のような表示がない) 場合は、Beauto ビルダー-NEO のインストールを行ってください)



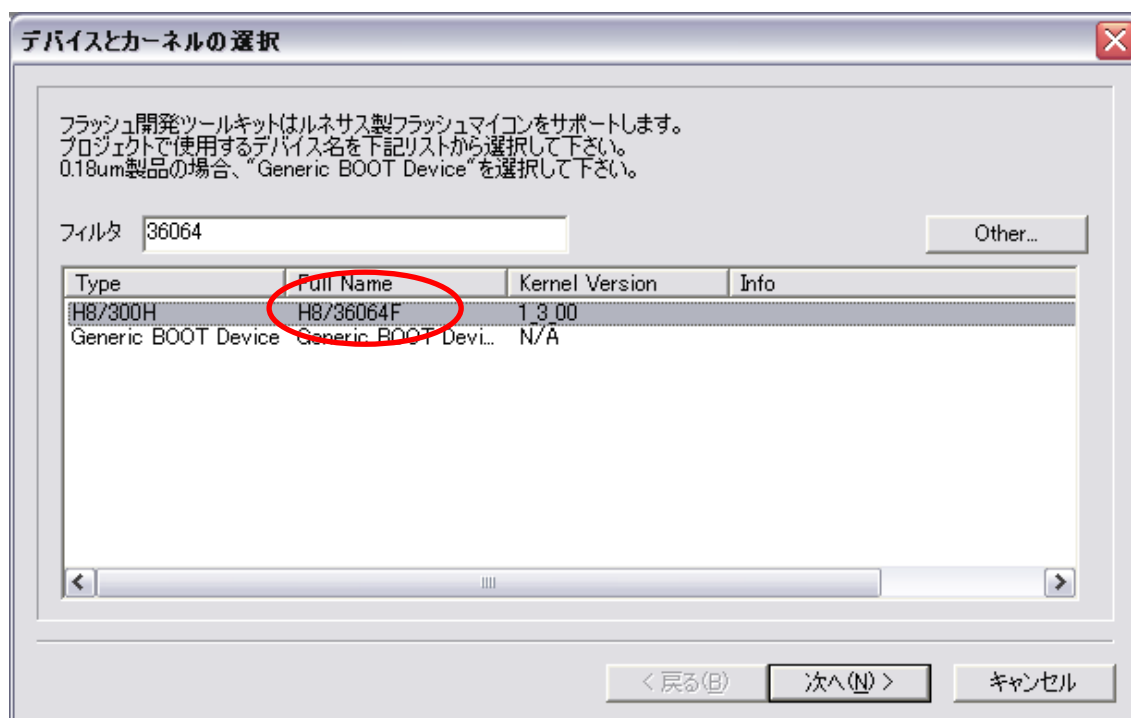
(2) 次に、書き込み用のツールを開きます。

スタートメニューの[プログラム]>[Renesas]>[Flash Development Toolkit *.*]>[Flash Development Toolkit *.* Basic]を開きます。(*.*はバージョンの番号)



(3) 初回起動時のみ、以下のオプション画面が表示されます。

CPU ボードが PC に接続されている状態で、フィルタに[36064]と入力し、一番上の[H8/36064F]を選択し、[次へ]を押します。



(4) [Select port:]にデバيسマネージャで調べた COM ポートを設定し次へを押します。



(5) [入力クロック]を[14.7456]に設定し、[次へ]を押します。

デバイス設定

デバイス設定用の値を入力してください:
[H8/36064F] using [Protocol B]

外部クロックまたは内部クロックを選択してください: External Clock

入力クロック: 14.7456 Mhz

クロックモード: NONE

メインクロックの通倍比 (CKM): 1

周辺クロックの通倍比 (CKP):

< 戻る(B) 次へ(N) > キャンセル

(6) そのままの設定で[次へ]を押します。

接続タイプ

フラッシュ開発ツールキットは多くの異なる方法でユーザのデバイスに接続することができます。

Select Connection:
☒ BOOT Mode ☐ USER Program Mode

☐ Kernel already running

ブートモードで接続した場合は、フラッシュメモリのデータはあらかじめ消去されています。(デバイスの種類に依存します)

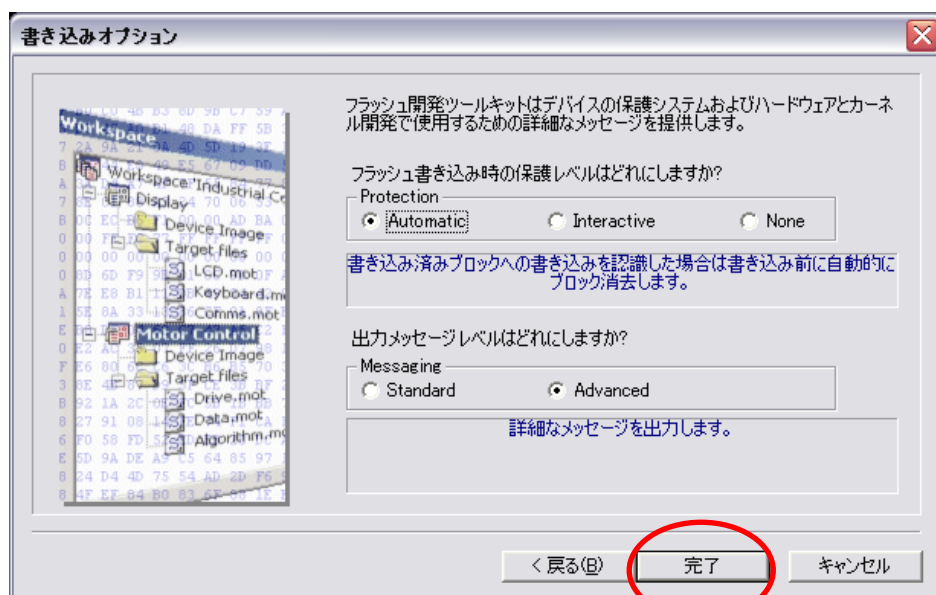
通信は調歩同期式で行っております。そのため通信速度はデバイスの動作クロックおよび接続機器により設定できる通信速度が異なります。

☒ ボーレート(推奨): 19200 ☒ Use Default

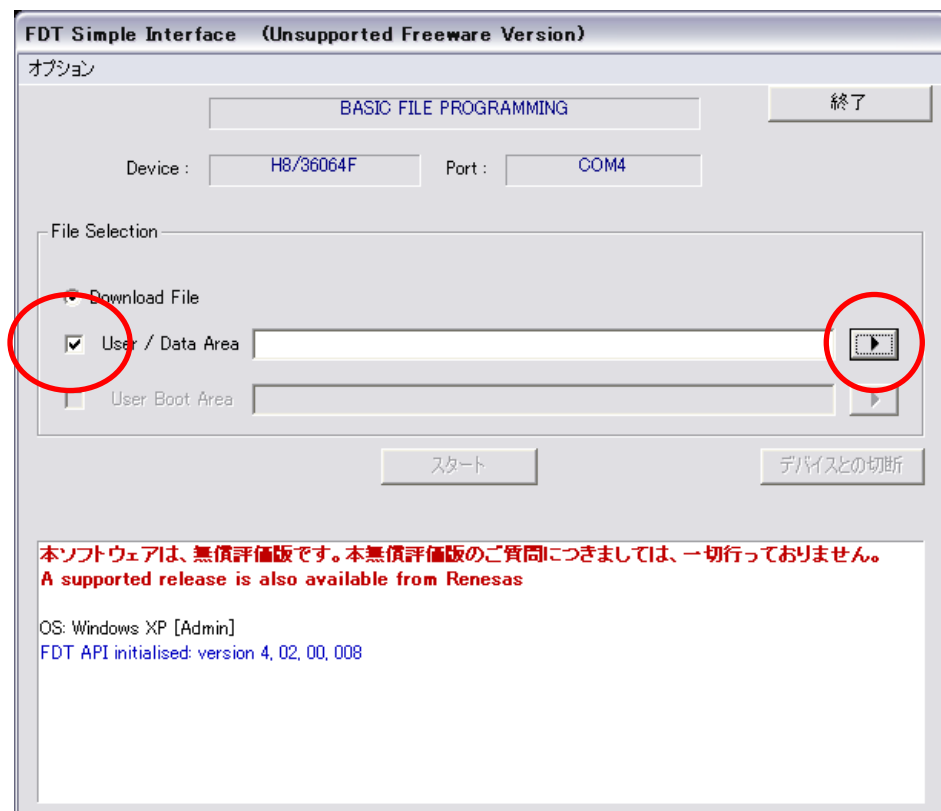
☐ ボーレート(ユーザ入力):

< 戻る(B) 次へ(N) > キャンセル

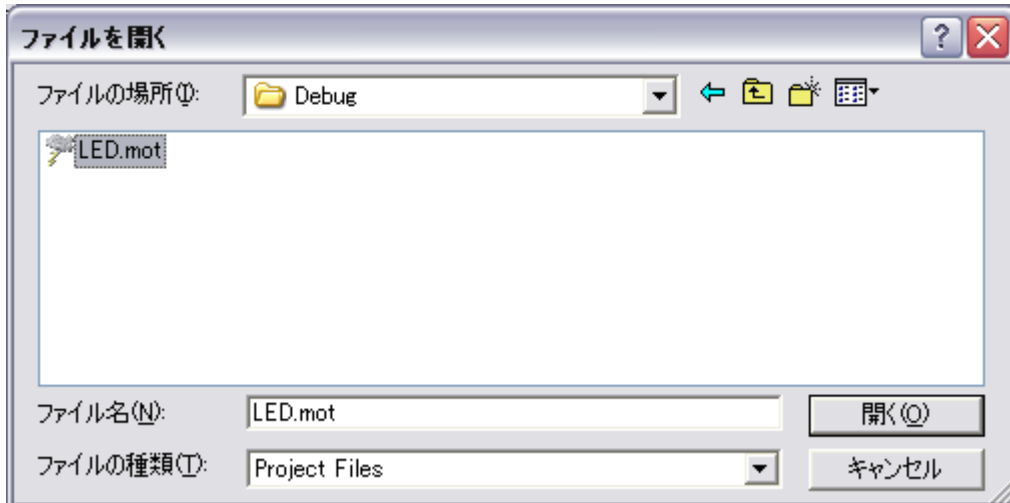
(7) そのままの設定で[完了]を押します。



(8) 設定が完了したら、[User/DataArea]をチェックし、右向きの三角のボタンを押し、書き込むファイルを選択します。



(9) 書き込みファイルの[(プロジェクト保存先)¥WRC003_SampleProject¥LED¥Debug]内の[LED.mot]を開きます。

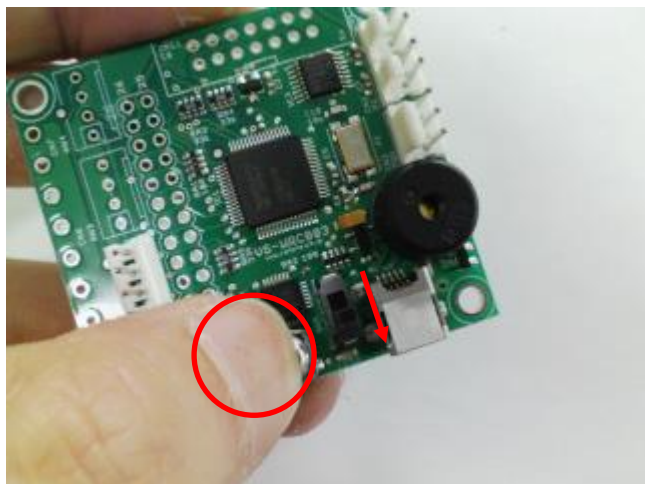


(10)CPU ボードから一度 USB ケーブルを抜き、以下の手順で書き込んでください。

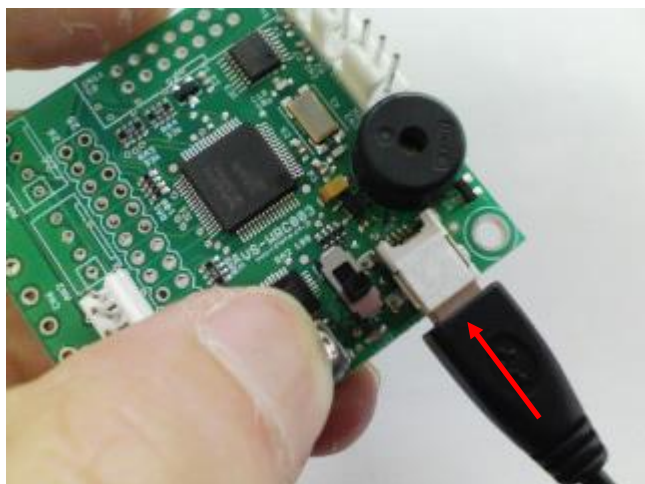
・ 書き込みの手順

①電源スイッチを切ります。

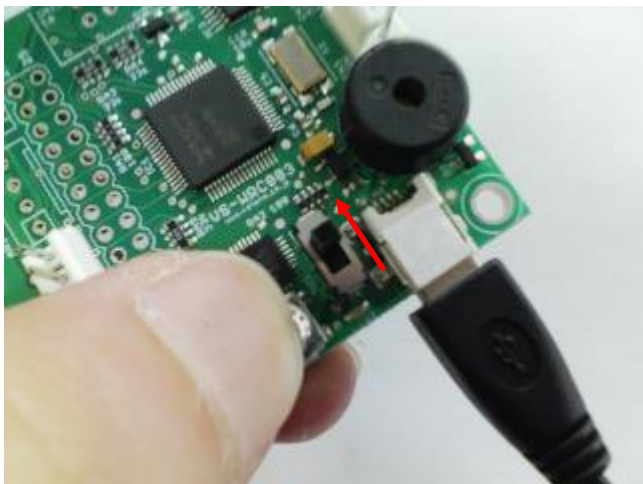
②スイッチを押します。



③スイッチを押したまま USB コネクタを差し込みます。

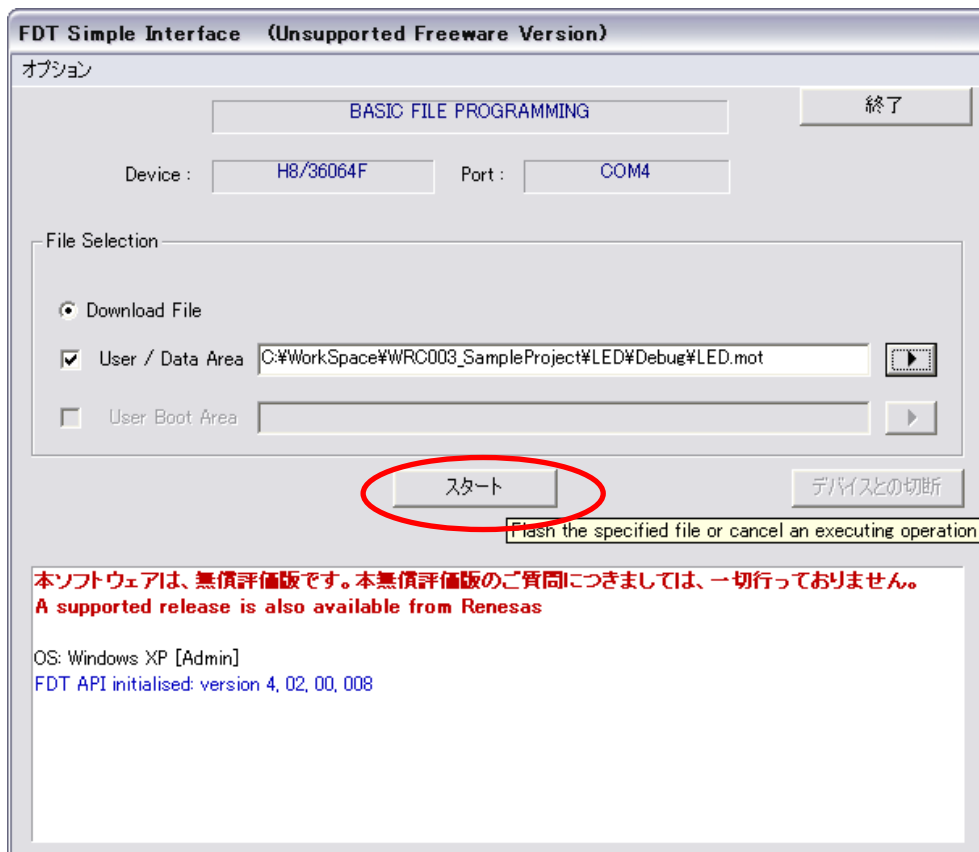


④スイッチを押したまま、電源スイッチを入れます。



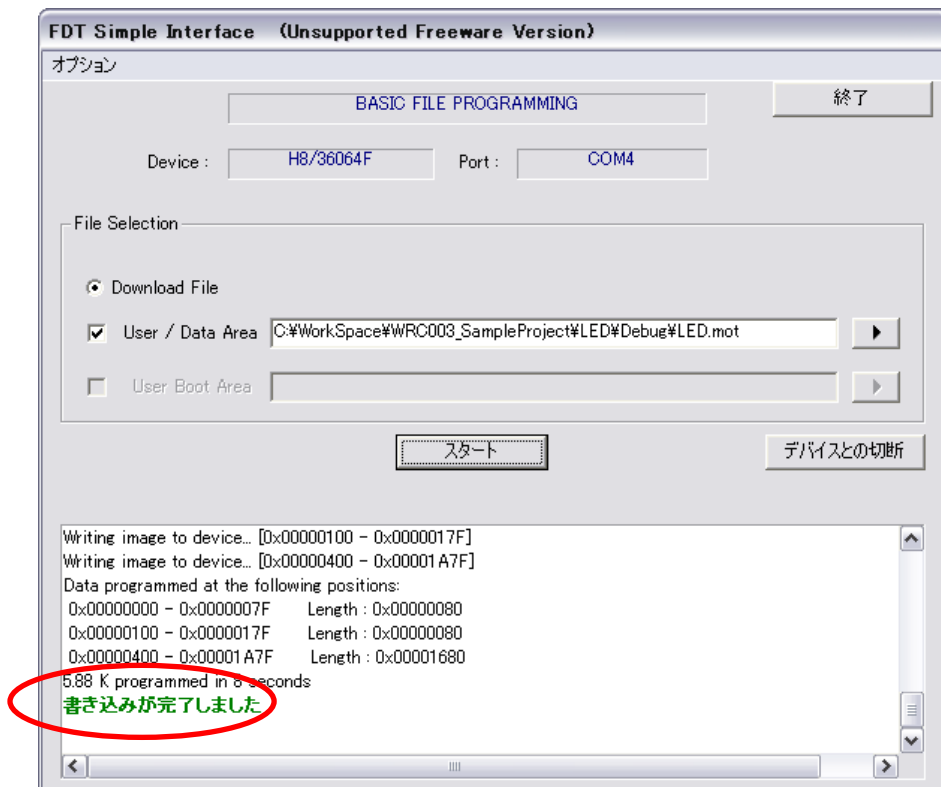
④この状態のまま、FDT のスタートボタンを押します。

書き込み中はスイッチから手を離さないでください。



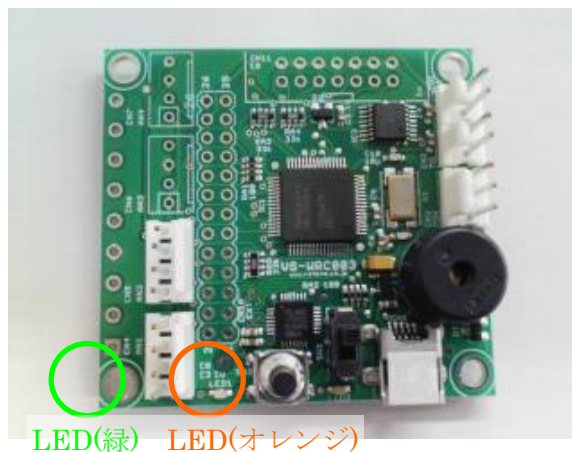
(11) 下のように[書き込みが完了しました]と表示されれば、書き込み完了です。

表示されない場合は[デバイスと切断]を押した後、再度 (10) の手順で書き込みを行ってください。



(12) CPU ボードの電源を入れると、書き込んだプログラムが実行されます。

きちんと書き込めている場合、CPU 上のオレンジと緑の LED が交互に点滅します。



※ 設定は次回起動時も保存されていますので、(10)～(11)の手順のみで書き込むことが出来ます。

※ 違うファイルを書き込みたい場合は、(8)～(10)と同じ手順で行ってください。

6 LED 点滅プログラム

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-WRC003.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。

```
0: void main(void)
1: {
2:     //制御周期の設定[単位：Hz 範囲：30.0~]
3:     const BYTE MainCycle = 60;
4:     Init((BYTE)MainCycle); //CPUの初期設定
5:     InitSci3(CBR_115200,even,1); //シリアル通信の設定
6:     BuzzerSet(0x80,0x80); //ブザーの設定
7:     //無限ループ
8:     while(1){
9:         LED(1); //緑のLED点灯
10:        Wait(1000); //1000msec待つ
11:        LED(2); //オレンジのLED点灯
12:        Wait(1000); //1000msec待つ
13:    }
14: }
```

main関数の各行について説明します。

0：関数の宣言

2～6：各機能の初期設定

ブザー、シリアル通信などの設定をしています。これらについては、各機能を使用するときに再度説明します。

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

上記のプログラムの場合、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。各関数について以下で説明します。

Ovoid LED(BYTE LedOn);

LED()関数はCPUボード上にある2つのLEDの点灯、消灯を、引数[LedOn]に数値をあたえることで制御することができます。

[LedOn]に与える数値は2進数でbit0が緑、bit1がオレンジのLEDに対応し、1のとき点灯、0のとき消灯します。

各数値を指定した時のLEDの状態は以下のとおりです。

数値 (10進 数)	bit1	bit0	LED (オレンジ)	LED (緑)
0	0	0	消灯	消灯
1	0	1	消灯	点灯
2	1	0	点灯	消灯
3	1	1	点灯	点灯

Ovoid Wait(int msec);

Wait ()関数は、引数[msec]で与えた時間(単位は msec)だけ待つ関数です。

たとえば、LED点滅プログラムの中にあるWait(1000)は1秒(=1000msec)だけ待つ処理になります。

例題①

スイッチが押されたときに緑のLEDを光らせるプログラムを作ってみましょう。
スイッチが押されているか、いないかの状態の取得には getSW() 関数を利用します。

○ BYTE getSW()

getSW() 関数を実行すると、スイッチが押されているかどうかを戻り値で判定できます。押されている場合は 1、押されていない場合は 0 を返します。

getSW() 関数の戻り値を if、else 文で判定し、LED を点灯、消灯することで実現できます。Main 関数は以下のようになります。

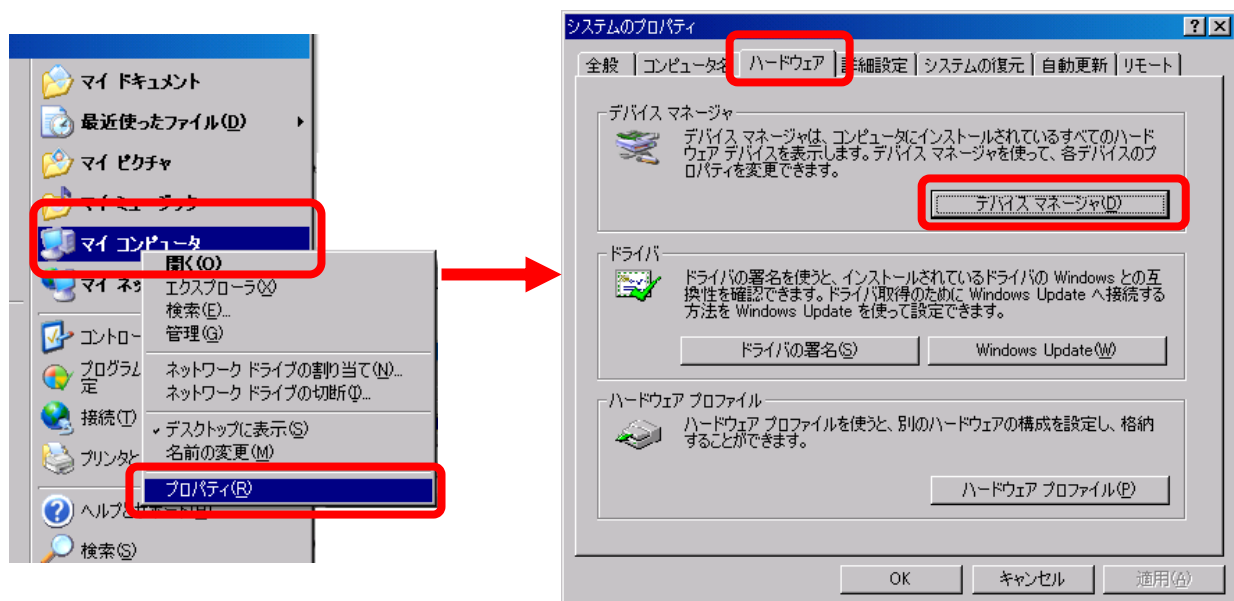
```
0: void main(void)
1: {
2:     //制御周期の設定[単位：Hz 範囲：30.0~]
3:     const BYTE MainCycle = 60;
4:     Init((BYTE)MainCycle);          //CPU の初期設定
5:     InitSci3(CBR_115200,even,1);    //シリアル通信の設定
6:     BuzzerSet(0x80,0x80);          //ブザーの設定
7:     //無限ループ
8:     while(1){
9:         if(getSW() == 1){           //スイッチが押されていたら
10:             LED(1);                //緑のLED 点灯
11:         }
12:         else{                       //押されていなかったら
13:             LED(0);                //すべてのLED 消灯
14:         }
15:     }
16: }
```

7__トラブル対応：FDT を使用した VS-WRC003 へのプログラム書き込みが

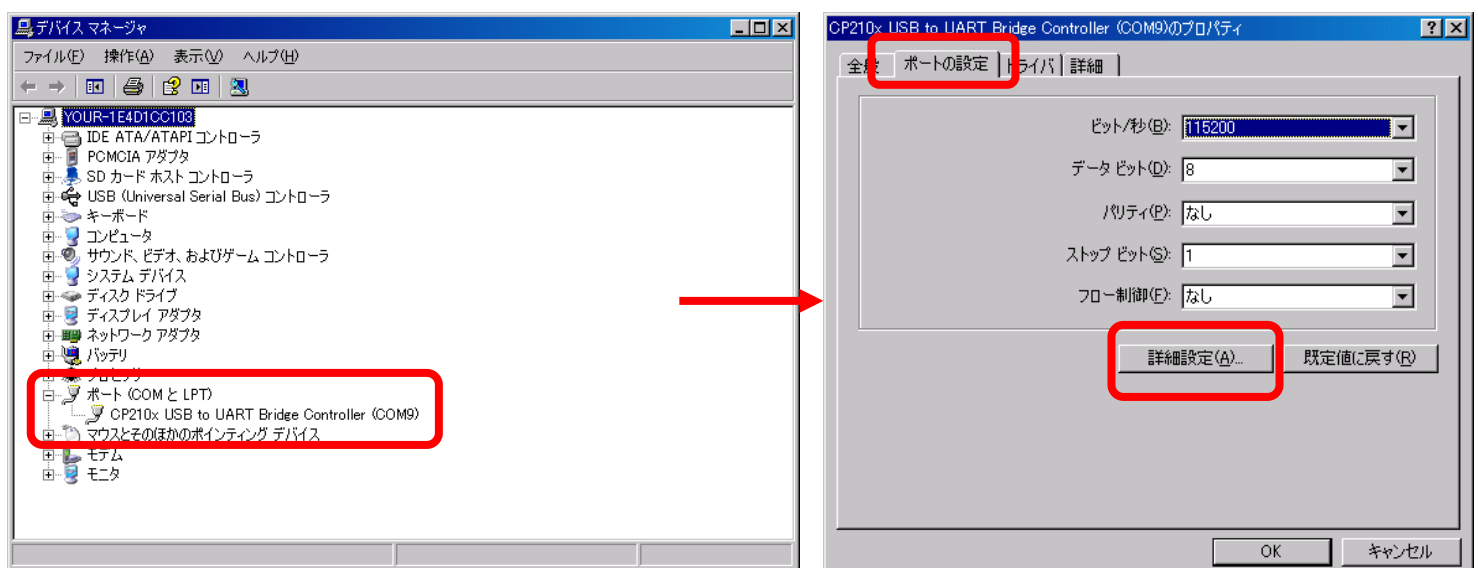
失敗する場合

本説明書に従い、ロボットにプログラムの書き込みを行なった際、ロボットとの接続やボタンの操作などが正しく行われているにもかかわらず書き込みに失敗する場合は、お使いの PC のシリアルポート番号設定を変更することで改善できる場合があります。 シリアルポート番号の変更は、以下の手順で行ないます。

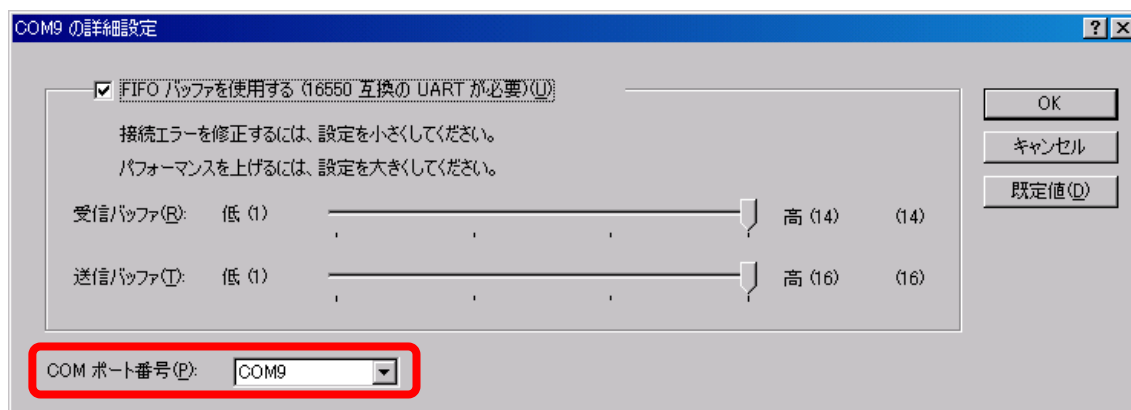
まず、OS のスタートメニューをクリックし、表示されたメニューより「マイコンピュータ」のアイコンを右クリックし、メニュー中の「プロパティ」をクリックしてください（左下図）。右下図のウィンドウを表示されるので、「ハードウェア」のタブをクリックし、「デバイスマネージャ」のボタンをクリックしてください。



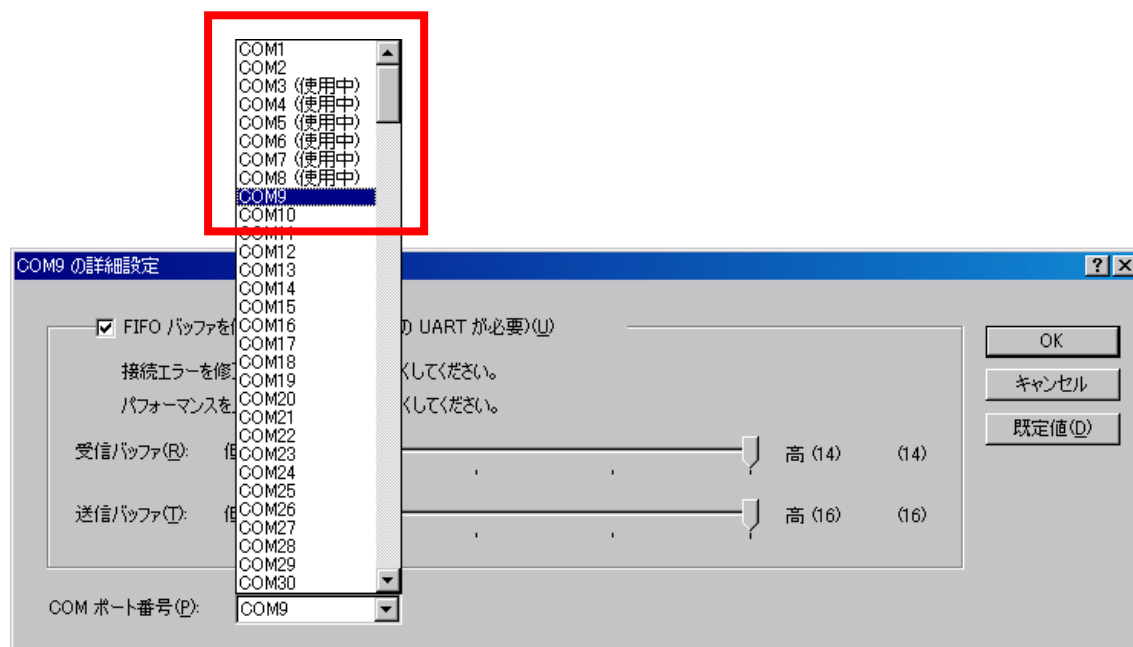
デバイスマネージャのウィンドウを画面に表示されたら、VS-WRC003 と PC を接続し、VS-WRC003 の電源を ON にしてください。ウィンドウの「ポート (COM と LPT)」の項目に「CP210x USB to UART Bridge Controller (COM?)」という行が追加されます。この行をダブルクリックしプロパティを表示します。「ポートの設定」のタブインデックスをクリックし、「詳細設定」を開きます。



詳細設定が表示されたら、下部にある「COM ポート番号(P)」より、シリアルポート番号を変更することができます。

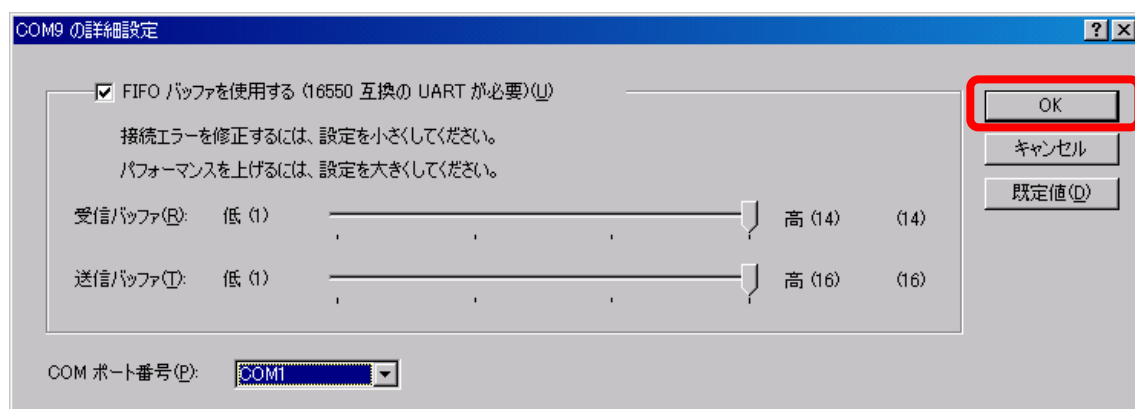


「COM ポート番号(P)」の項目をクリックするとシリアルポート番号の選択が表示されます。こちらより、現在の設定より小さいシリアルポート番号を選択します。

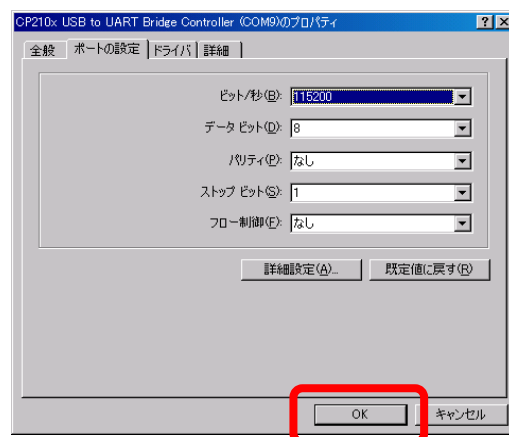


項目に「(使用中)」の表示が無い番号のシリアルポートを選択してください。また、現在の設定より小さいシリアルポート全てに「(使用中)」の表記がある場合、現在実際に使用していない番号を選択しても問題なく利用できる場合があります。

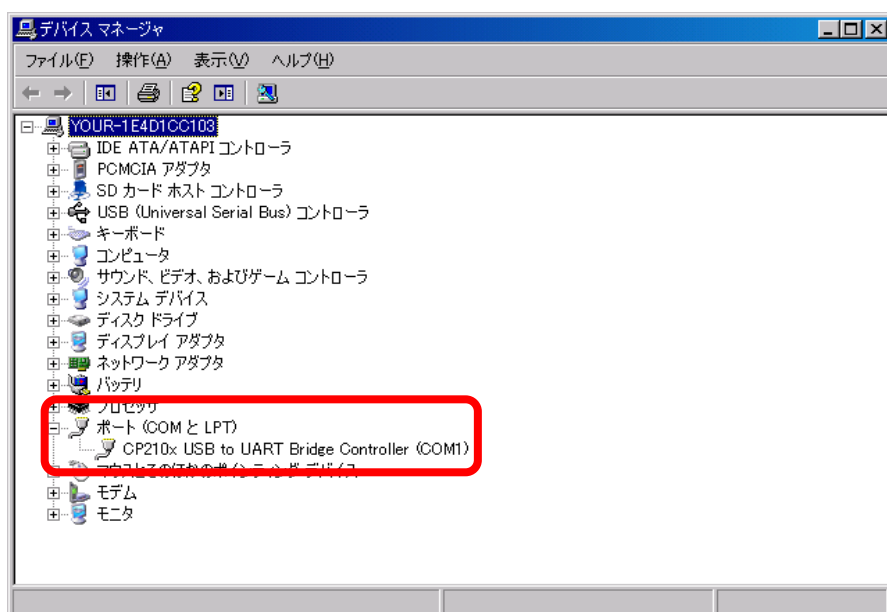
設定を行ったら、「OK」ボタンをクリックして設定を適用しダイアログを閉じてください。



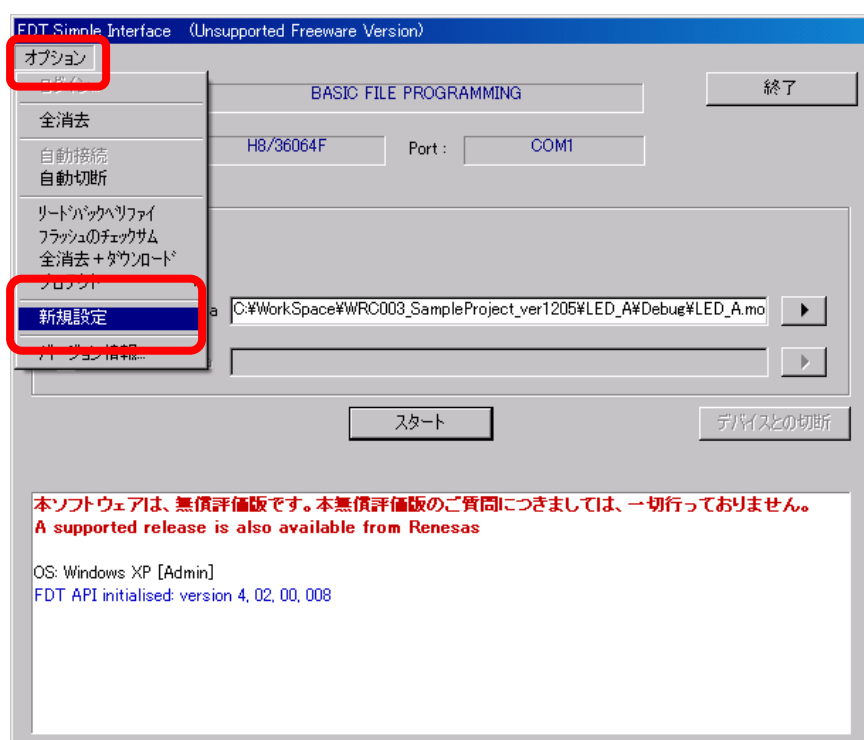
続いて以下のダイアログも「OK」ボタンをクリックし、設定を適用してダイアログを閉じてください。



最後に、一度 VS-WRC003 を PC から抜き差しして、PC に VS-WRC003 を再認識させてください。設定が正しく行なわれていたら、シリアルポート番号が変更した番号に切り替わります。



シリアルポート番号の変更は、FDT の設定にも反映させる必要があります。FDT の設定を変更する場合は、メニューより「オプション」→「新規設定」をクリックしてください。クリックすると、本説明書 9 ページからの記述と同じ画面が表示されるので、説明書の記述に従い、改めて設定を行なってください。



■オプションパーツ、関連商品のご購入は・・・

No.1 の品揃え！ 各種オプションパーツ、ロボット関連製品のご購入はコチラ

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

楽天・Amazon・Yahoo の各 Web 店舗、または東京、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

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

商品に関するお問い合わせ

商品の技術的なご質問は、問題・症状・ご使用の環境などを記載の上メールにてお問い合わせください。

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

TEL: 06-4808-8701 FAX: 06-4808-8702

VstoneTM
www.vstone.co.jp