

H8 搭載学習用 CPU ボード

# VS-WRC003LV

取扱説明書



# 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
6 開発環境の導入とサンプルプログラムの実行.....	12
6-1 開発環境の入手.....	12
6-2 HEW のインストール.....	15
6-3 HEW の起動とサンプルプロジェクトの読み込み.....	20
6-4 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.....	22
6-5 VS-WRC003LV へのプログラムの書き込み.....	28
6-6 LED 点滅プログラムについて.....	30

# 1 はじめに

本説明書は、H8 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-WRC003LV」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

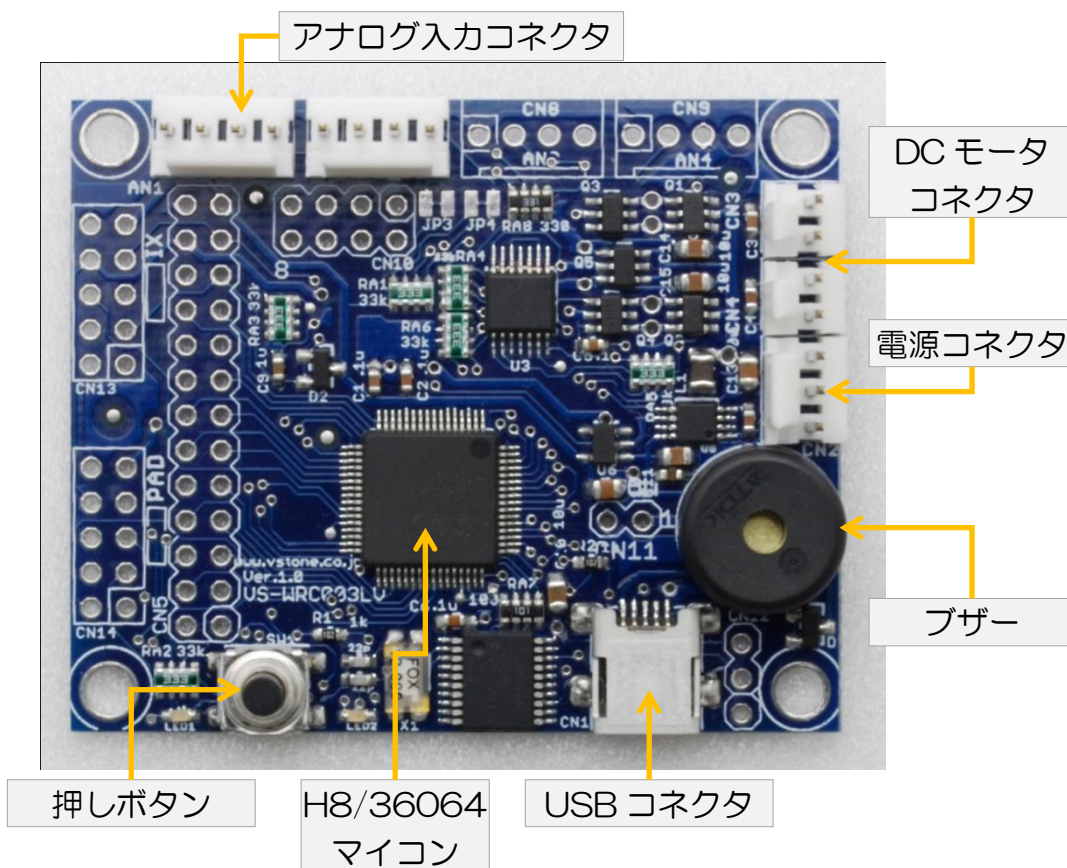
サイズ	W56×D46 (mm)
重量	13g
CPU	H8/36064G
電源	DC 2~4V アルカリ乾電池 2本 または ニッケル水素充電電池 2~3本
出力	DC モータ×2 (連続電流 2A まで) LED×2 (オレンジ・緑) 圧電ブザー×1
入力	アナログセンサ入力×2 (最大 4)
インタフェース	USB (HID 準拠)、シリアルポート (3.3v レベル)、I2C
オプション	VS-WRC004LV 無線コントローラ VS-C1 IXBUS (VS-IX001、VS-IX0004) ロータリーエンコーダ拡張セット など

### 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-WRC003LV（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、基板上的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。
- 基板上的端子（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。
- 基板上的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 of H8 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。H8 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 H8 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、ビュートローバー付属の 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。

無償開発環境 HEW、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。

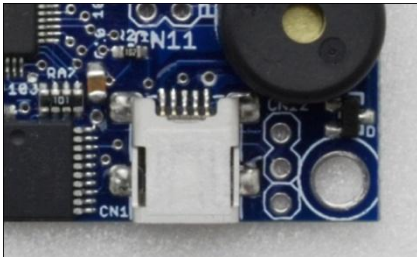
### ○ ポート

各機能を構成する部品が H8 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。

本ボードにはHID-シリアル変換IC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。



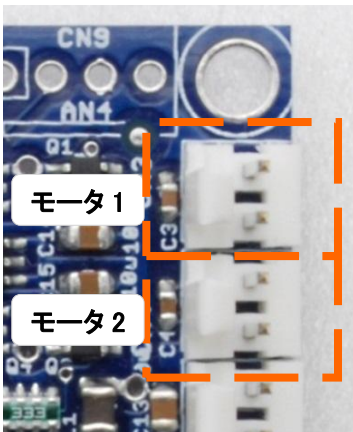
- ポート  
HID-シリアル変換ICとH8のP22/TXD、P21/RXDが接続されています。

本ボードとPC間のHIDでの通信仕様は以下のマニュアルをご確認ください。

## 5-3 DC モータ出力 (CN3, 4)

本ボードには2ch(拡張ボードVS-WRC004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 ( int mt1 , int mt2 , int mt3 , int mt4 , int mt5 , int 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-WRC003LV

- ・ CN3 (M1)      P30、 P31 (方向出力)      P61 (PWM)
- ・ CN4 (M2)      P32、 P33 (方向出力)      P62 (PWM)

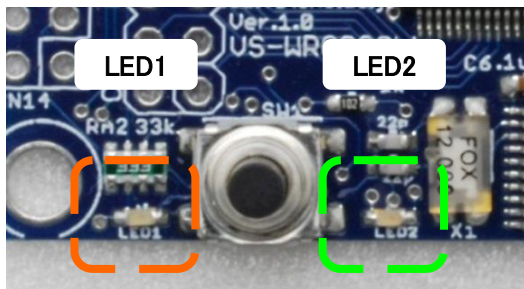
### VS-WRC004LV (オプション)

- ・ CN3 (M3)      P34、 P35 (方向出力)      P63 (PWM)
- ・ CN4 (M4)      P36、 P37 (方向出力)      P65 (PWM)
- ・ CN5 (M5)      P10、 P12 (方向出力)      P66 (PWM)
- ・ CN6 (M6)      P54、 P55 (方向出力)      P67 (PWM)

## 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

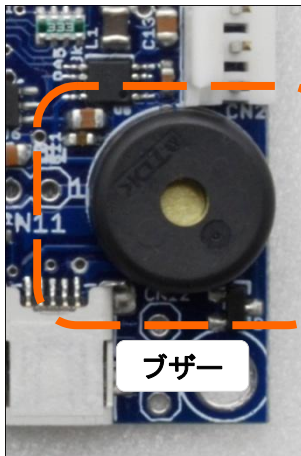
戻り値：        無し

## ○ ポート

- ・ LED1 (オレンジ) : P60
- ・ LED2 (緑)        : P64

## 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。ブザーはタイマーVの出力ポートに接続されているため、H8のPWM出力機能で作動させることができます。



○ ブロック



ブザーブロック

○ ポート

P76/TMOV (タイマーV 出力ポート)

○ 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()関数を呼び出すまでなり続ける。

引数： 無し 戻り値： 無し

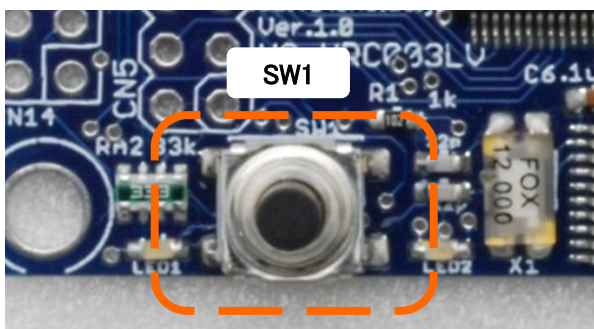
```
void BuzzerStop();
```

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

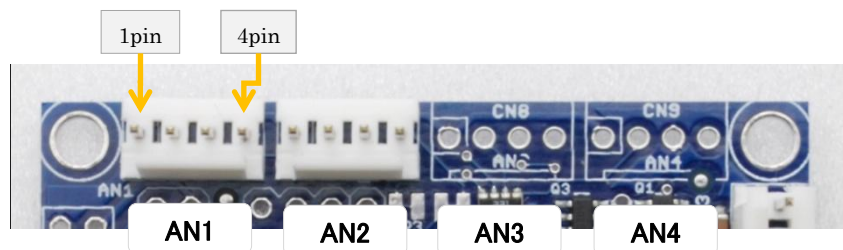
○ ポート P74



## 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-WRC004 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」や VS-WRC003 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)

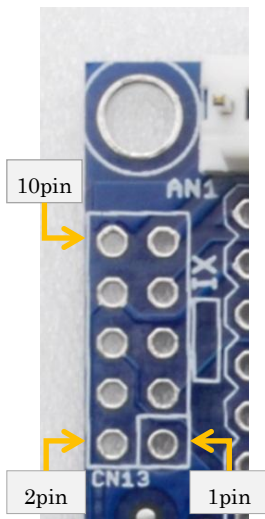
### ○ ポート    P74

## 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。このシリアル通信ポートは、HID-シリアル IC とも接続されていますので、USB 経由の通信と同時に使用することはできません。

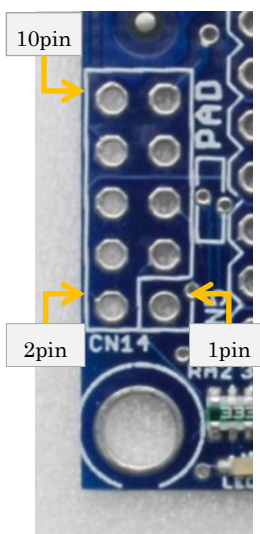


### ○ピン配置

- 1 pin : P22/TXD
- 2 pin : P21/RXD
- 3 pin : RES
- 4 pin : NC
- 5 pin : P57/SCL
- 6 pin : P56/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。また、コントローラで使用するポートとは別にシリアル通信ポート (SCI3\_2、1,2pin) も備えています。



### ○ブロック

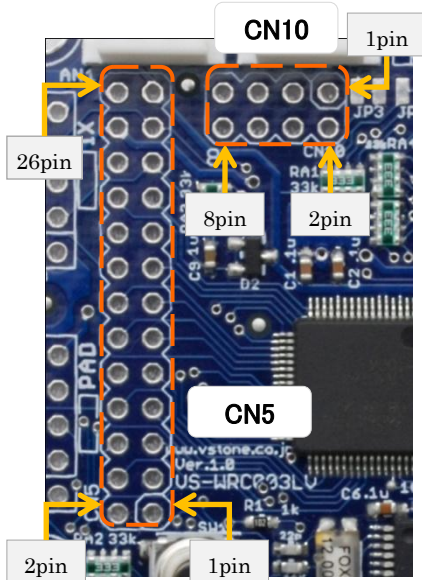


### ○ピン配置

- 1 pin : P72/TXD\_2
- 2 pin : P71/RXD\_2
- 3 pin : P75/TMCIV
- 4 pin : P20/SCK3
- 5 pin : P23/
- 6 pin : P24/
- 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-WRC004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	: P70/SCK3_2
2 pin	: P17/IRQ3/TRGV
3 pin	: P16/IRQ2
4 pin	: P15/IRQ1/TMIB1
5 pin	: P14/IRQ0
6 pin	: P12
7 pin	: P11/PWM
8 pin	: P10

### OCN5 ピン配置

1 pin	: PB6/AN6	14 pin	: P52/WKP2
2 pin	: PB5/AN5	15 pin	: P51/WKP1
3 pin	: PB4/AN4	16 pin	: P50/WKPO
4 pin	: P87	17 pin	: P37
5 pin	: P86	18 pin	: P36
6 pin	: P85	19 pin	: P35
7 pin	: P67/FTIOD1	20 pin	: P34
8 pin	: P66/FTIOC1	21 pin	: RES
9 pin	: P65/FTIOB1	22 pin	: NMI
10 pin	: P63/FTIOD0	23 pin	: +5V
11 pin	: P55/WKP5/ADTRG	24 pin	: +Vbat
12 pin	: P54/WKP4	25 pin	: +3.3V
13 pin	: P53/WKP3	26 pin	: 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-WRC003LV」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

### 6-1 開発環境の入手

High-performance Embedded Workshop（以下HEW）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://japan.renesas.com/products/tools/evaluation\\_software/download\\_search\\_results.jsp](http://japan.renesas.com/products/tools/evaluation_software/download_search_results.jsp)

「ビュート ローバー」をお持ちの場合、付属 CD-ROM の「C 言語開発環境」フォルダに HEW のインストーラがありますので、そちらをご使用ください。

(1)URL リンク先にある、以下のソフトウェアを選択します。

【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

ソフトウェア名	バージョン	リリース日	備考
M3T-CC32R	NC308WA V.5.42 Release 00	Apr.03.09	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	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は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。特にマイクロソフトウェアは、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きよう、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、その他お添付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須です。

※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)	<a href="#">Download</a>

(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	<input type="text"/>
パスワード	<input type="password"/>
<input type="button" value="取消し/閉じる"/>	<input type="button" value="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](#)

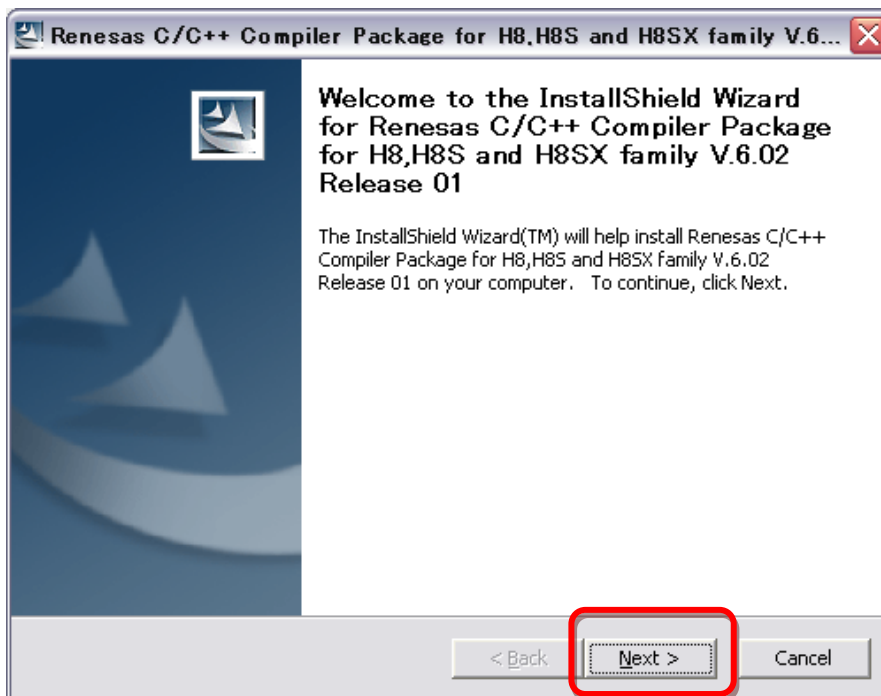
## 6-2 HEW のインストール

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

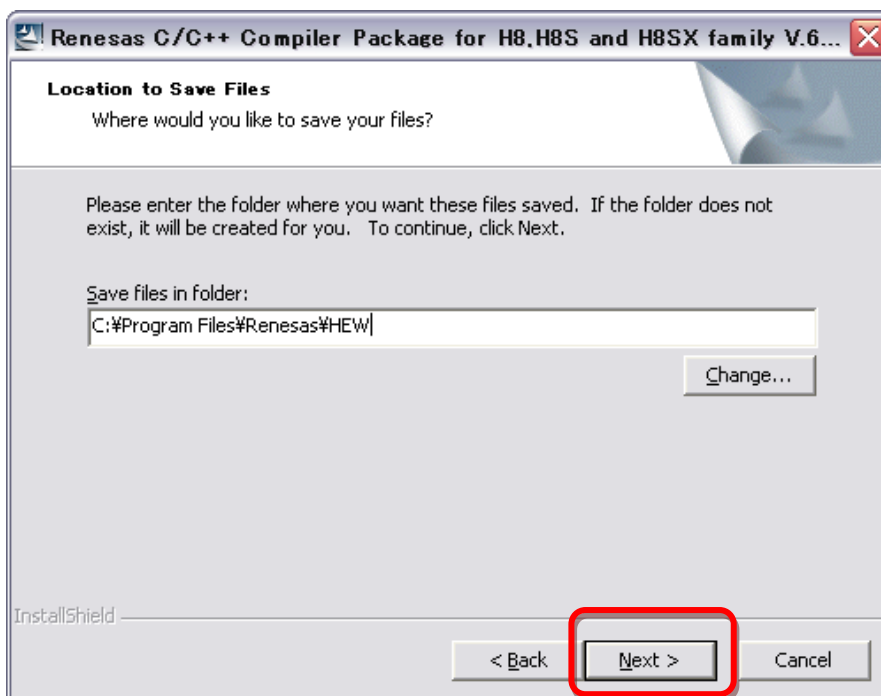
(\*\*\*\*にはバージョンの数字が入ります)

(2)NEXT を押します。

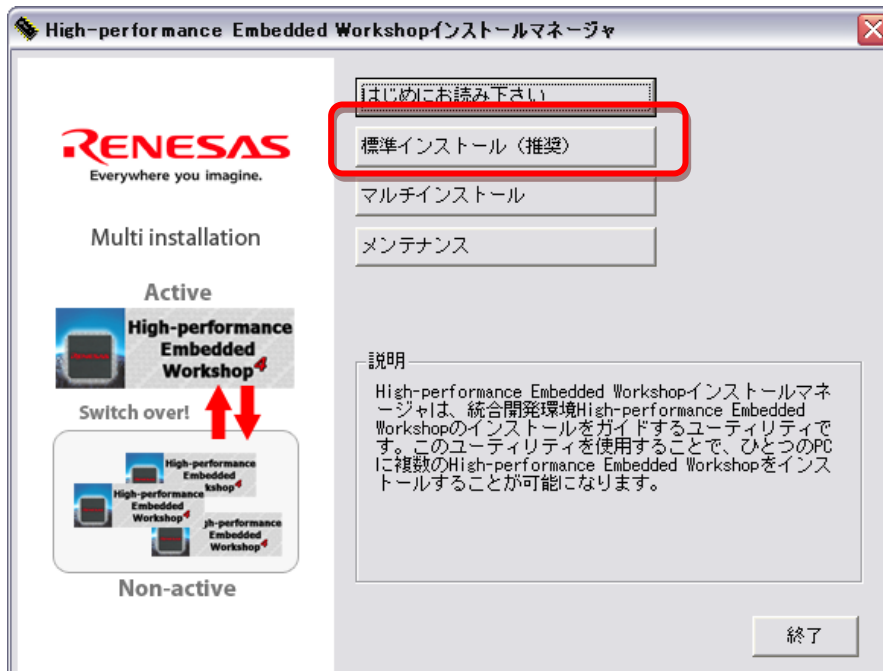
※以降の画面は V.6.02 Release 01 のものであり、最新のバージョンでは表示が異なる可能性があります。



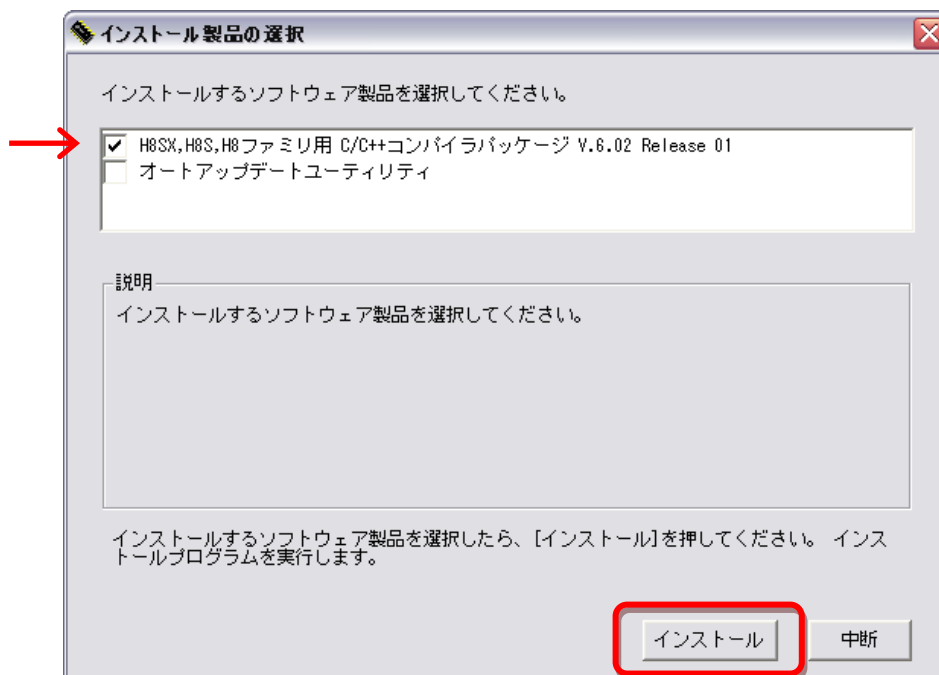
(3)パッケージを展開する任意のフォルダを指定して[Next]を押します。



(4) [標準インストール]を押します。

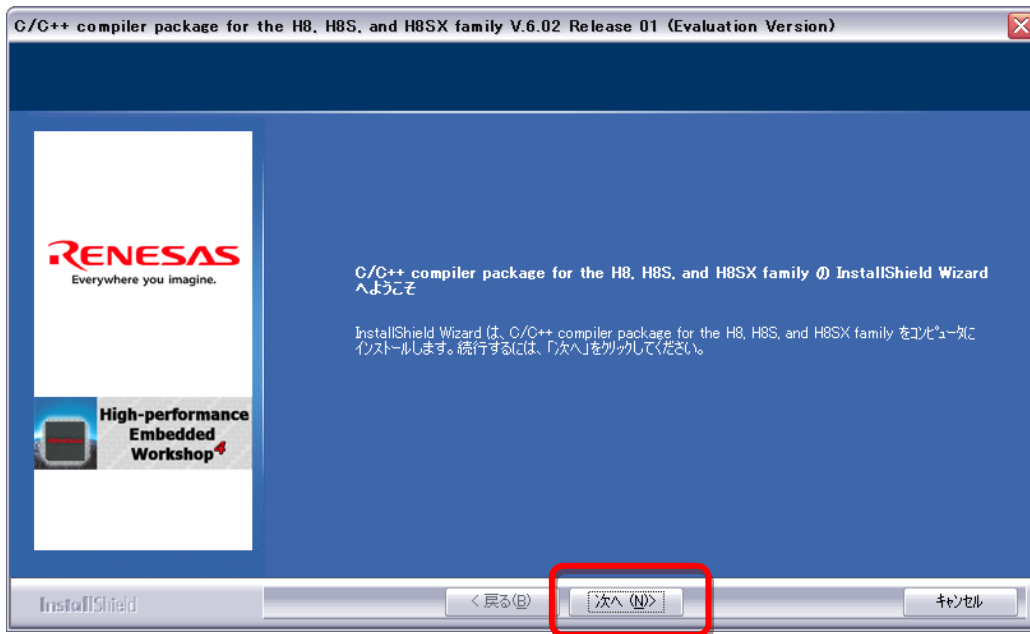


(5) チェックボックスを下の画像と同じように選択し、[インストール]を押します。

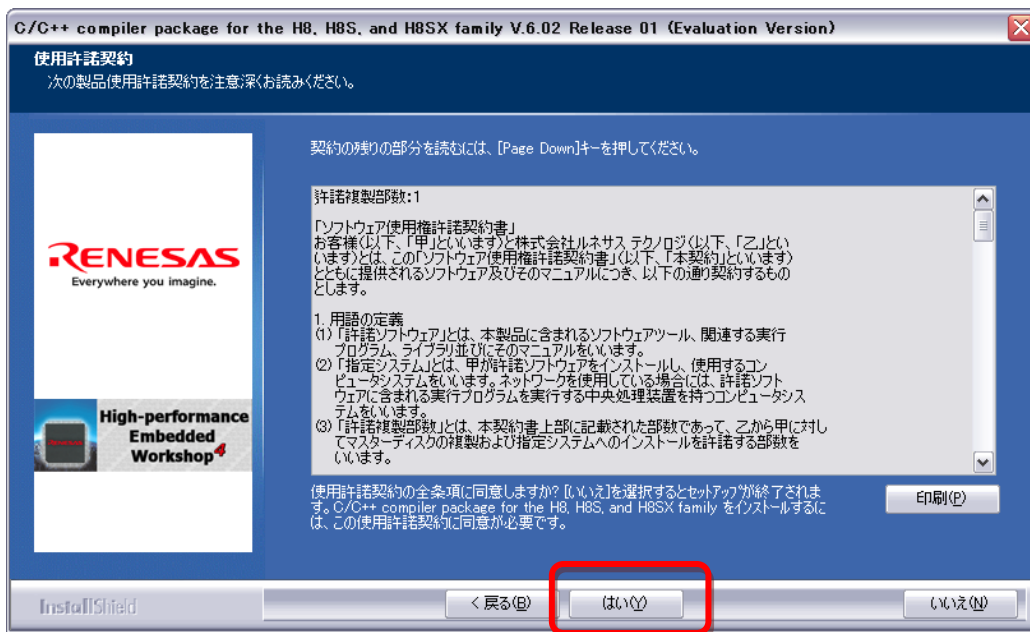




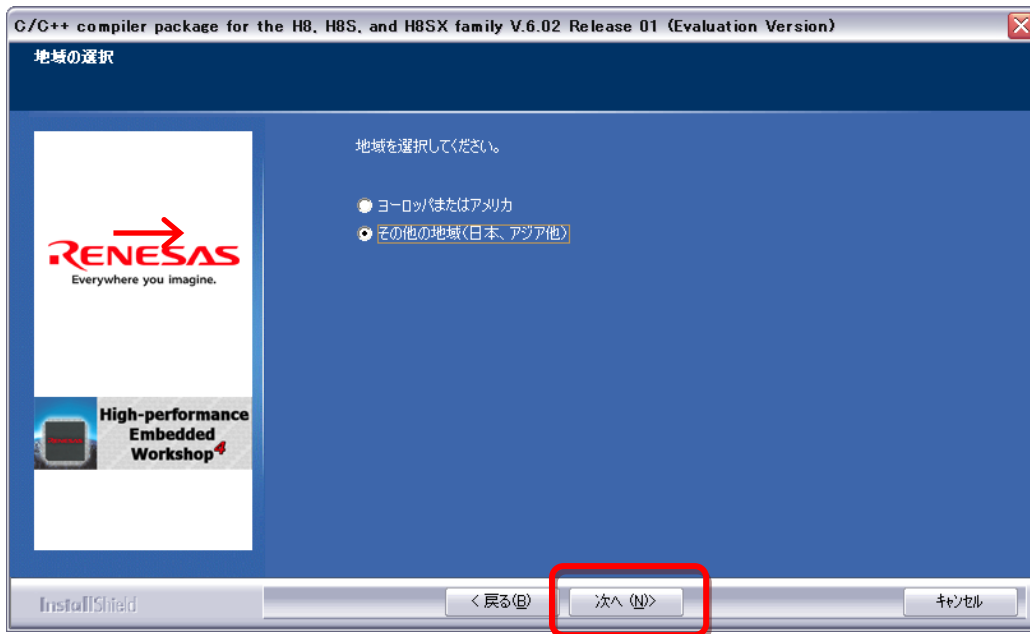
(6) [次へ]を押します。



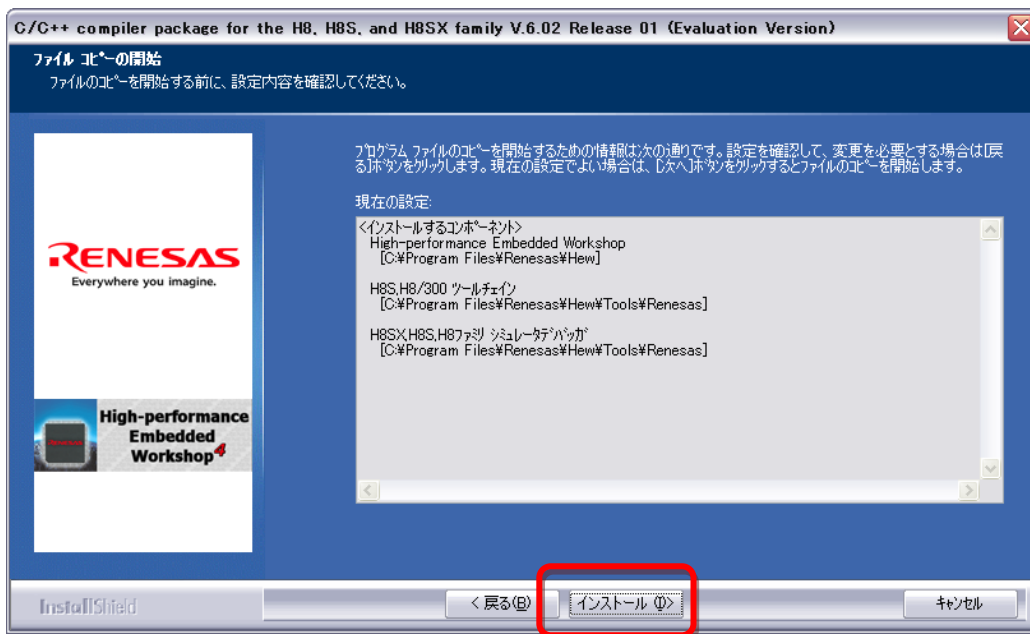
(7) [はい]を押します。



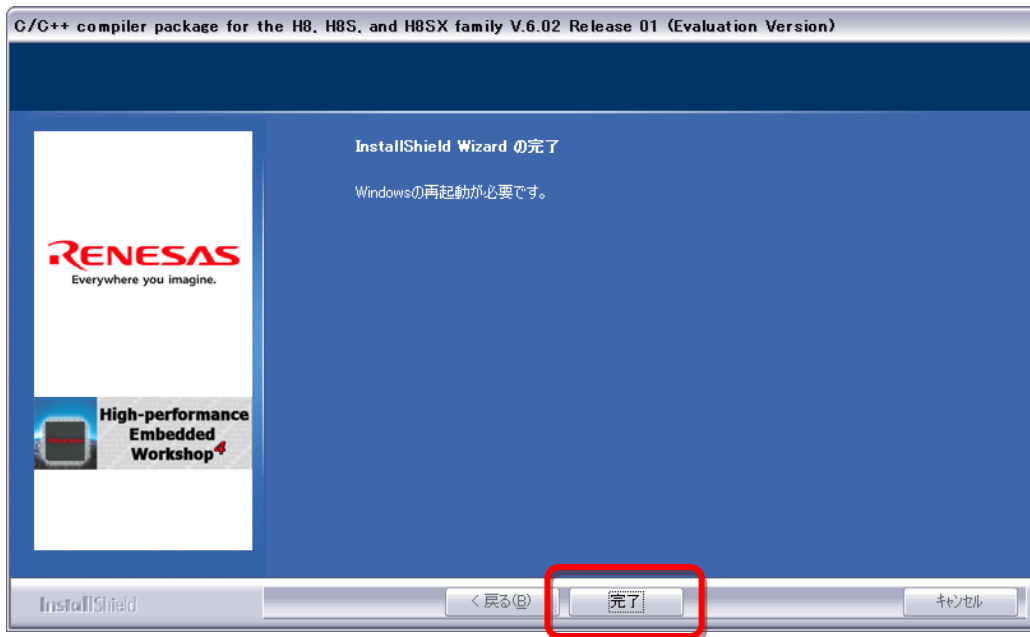
(8) [その他の地域 (日本、アジア他)] をチェックし、[次へ] を押します。



(9) [インストール] を押します。

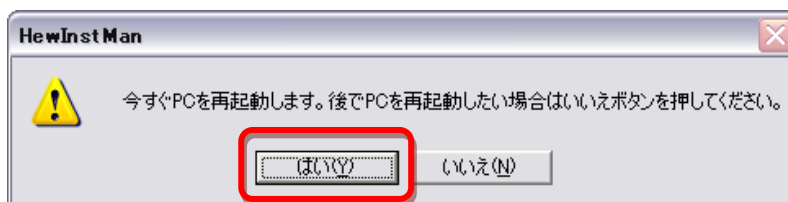


(10) [完了]を押します。



(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。)



## 6-3 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「WRC003LV\_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。

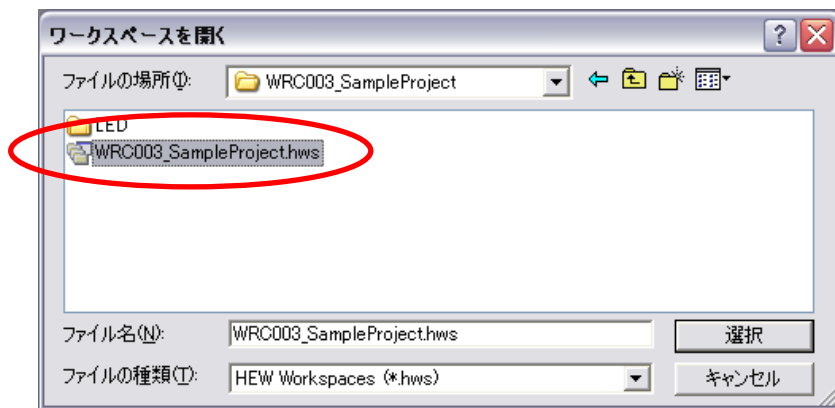
<http://www.vstone.co.jp/top/products/vs-wrc003/>

(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。

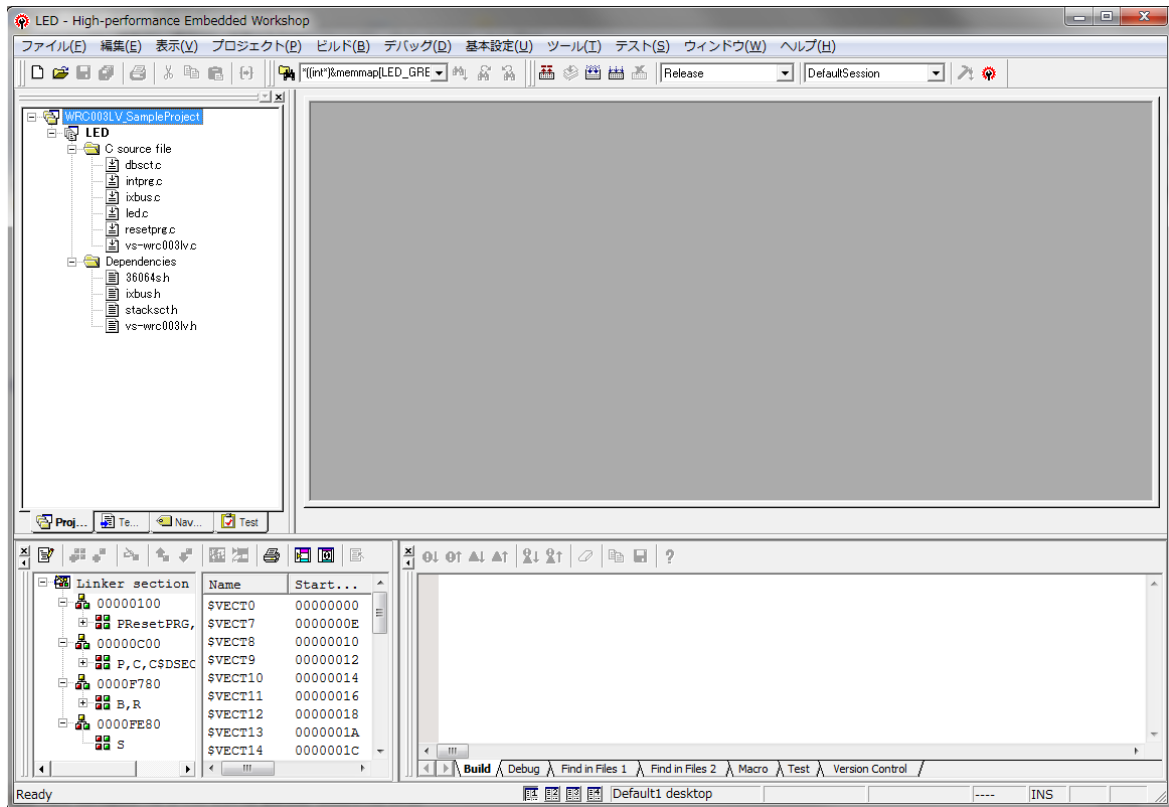


※プロジェクトを開く際に、HEW のバージョンによってツールチェーンが異なる場合があります、その場合、変更を求められることがあります。

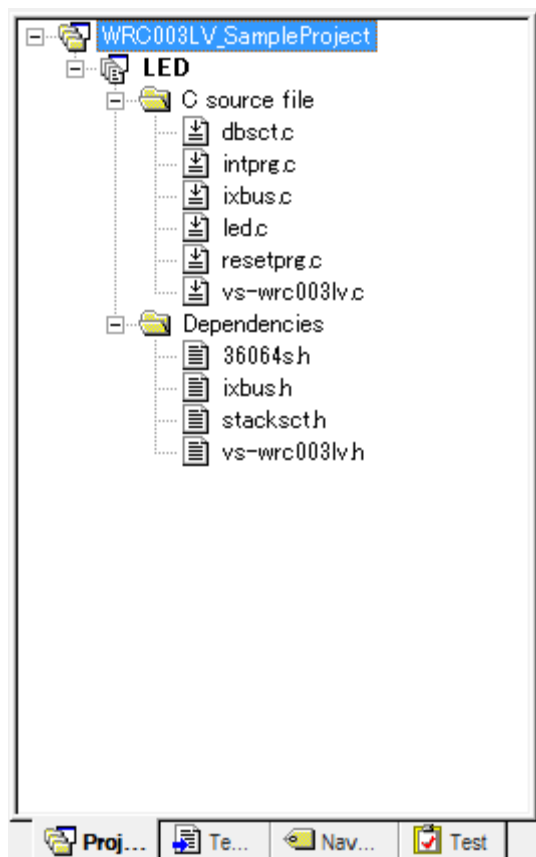
この表示がされた場合、必ずすべて「OK」を押してください。OK 以外を押すとビルドが正常にできなくなります。



(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。

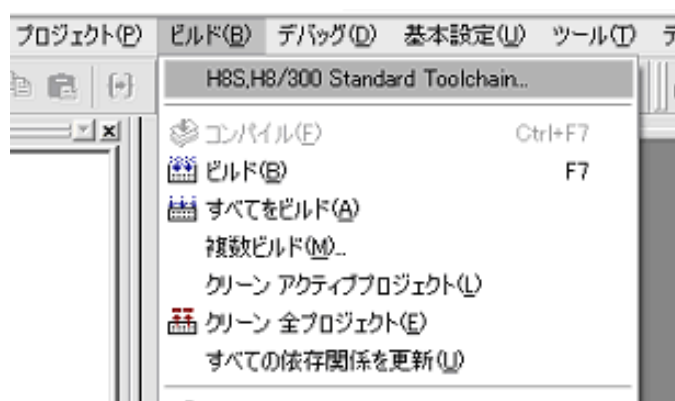


※複数のプロジェクトがある場合、サンプルのワークスペース内の[LED]を右クリックし、プロジェクトをアクティブにします。

## 6-4 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

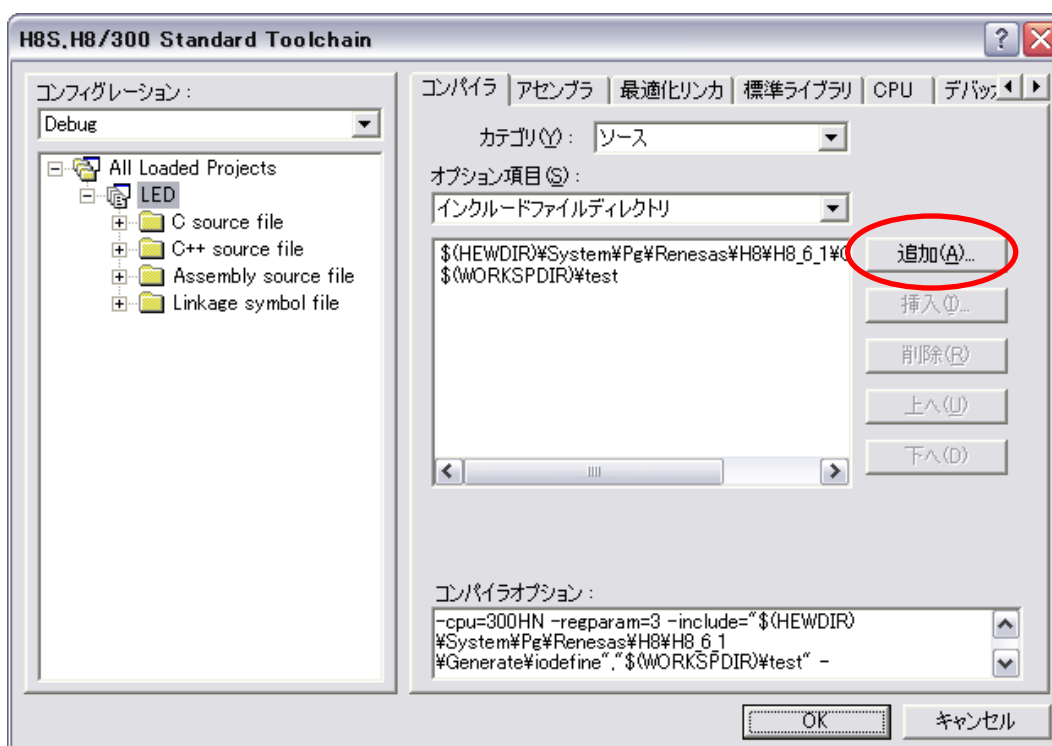
- (1) 外部から持ってきたプロジェクトそのままでは、必要なヘッダファイル（拡張子.hのファイル）が読み込まれていませんので、インクルードファイルディレクトリを設定します。
- (2) メニューの[ビルド]内の[H8S,H8/300 Standard Toolchain...]からツールチェーン（コンパイラ、リンカなどのこと）の設定をします。

※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。

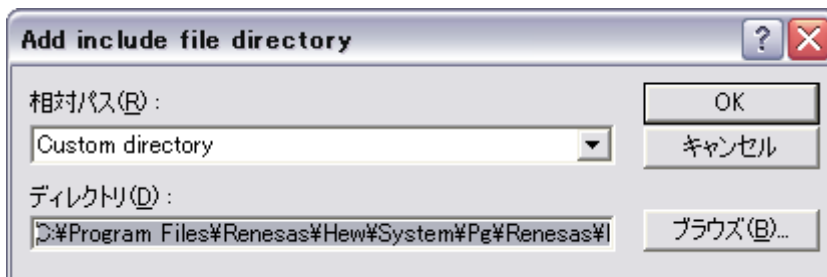


(4) インクルードディレクトリ追加用のウィンドウが開いたら、相対パスで[Custom directory]を選択し、ディレクトリに各 CPU 用のインクルードファイルがある[iodefne フォルダ](※)のパスを設定し、[OK]を押します。フォルダのパスは PC によりこととなりますので、必ず確認をしてください。

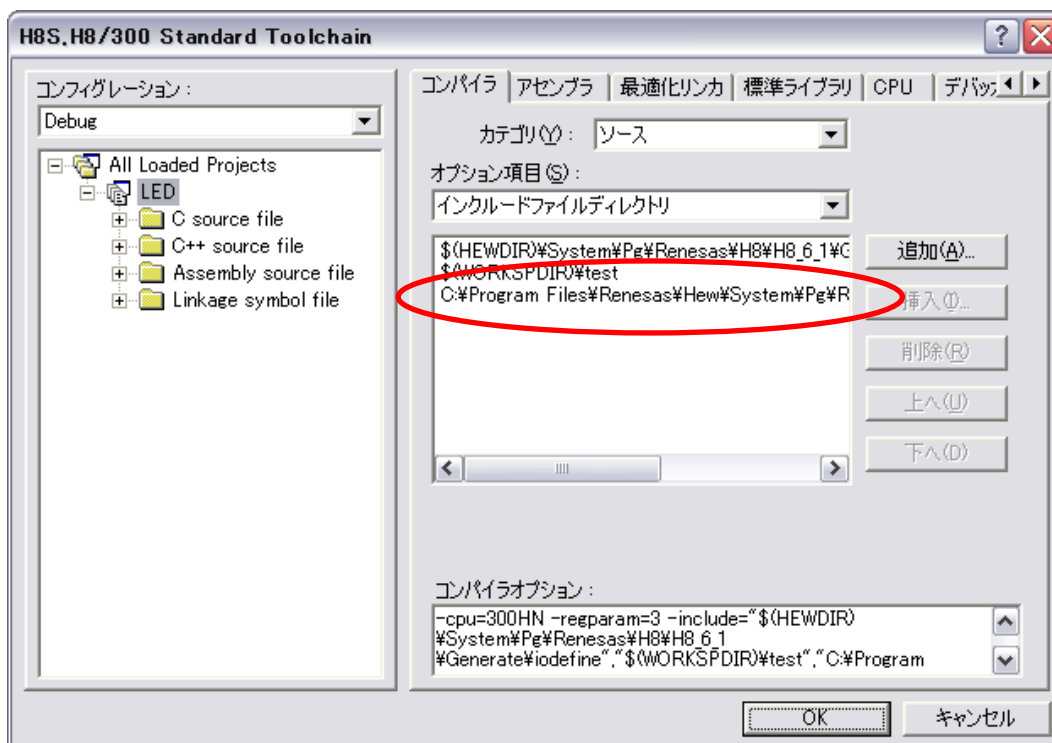
※ インストール時にインストール先のフォルダを変更していない場合、[iodefne フォルダ]は以下のようになります。(Windows が入っているドライブが C:の場合)

[C:¥Program Files¥Renesas¥Hew¥System¥Pg¥Renesas¥H8¥H8\_\*\_\*¥Generate¥iodefne]

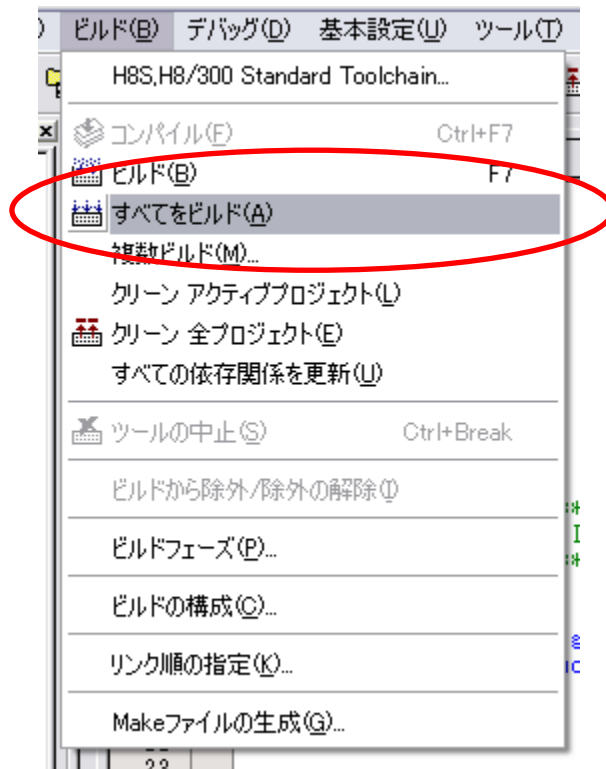
※ 「\*」にはツールチェーンのバージョンが入ります。



(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。



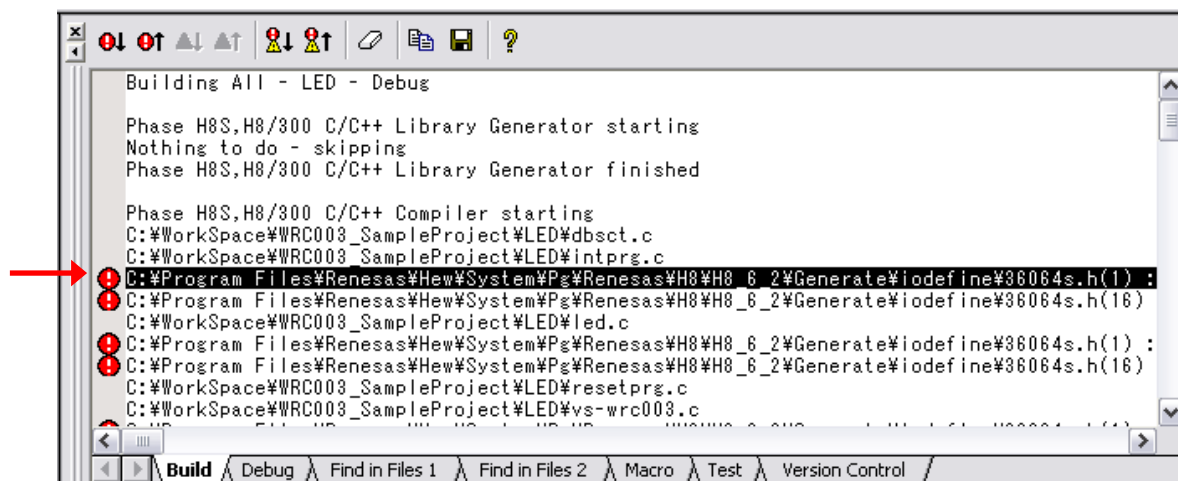
- (6) ここで、一旦ビルドをします。ビルドはメニューの[ビルド]内の[すべてをビルド]で行えます。  
(2回目以降は[ビルド(B)]を選択 または F7 キーを押すことでもビルドできます)



- (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、以下の化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)...\#ifndef\#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。

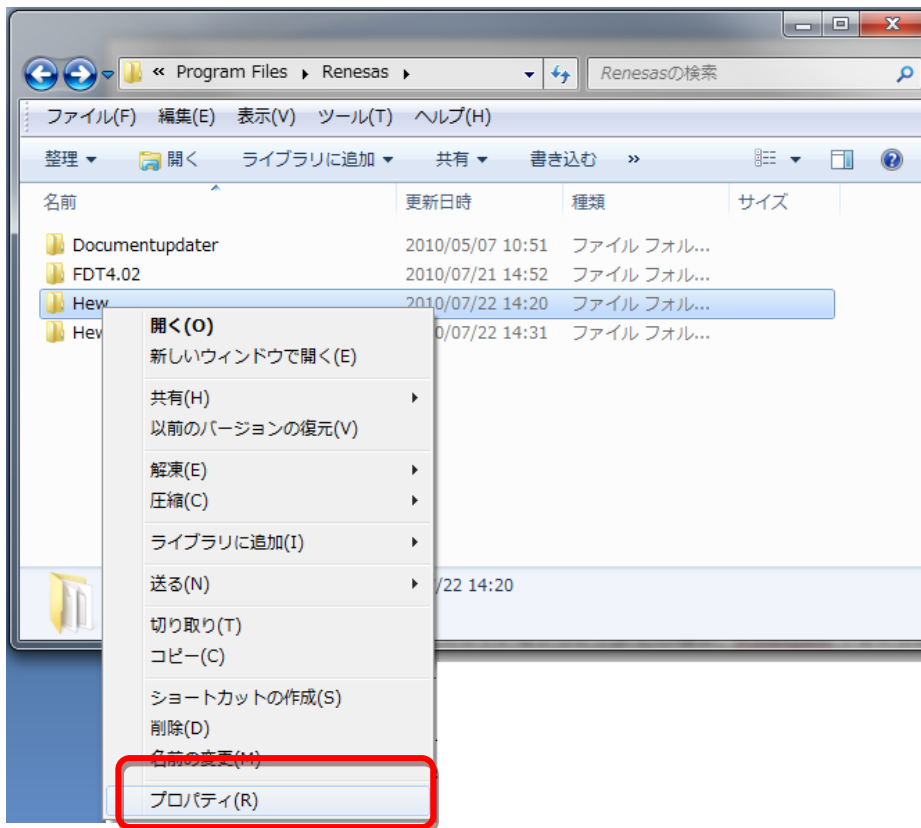




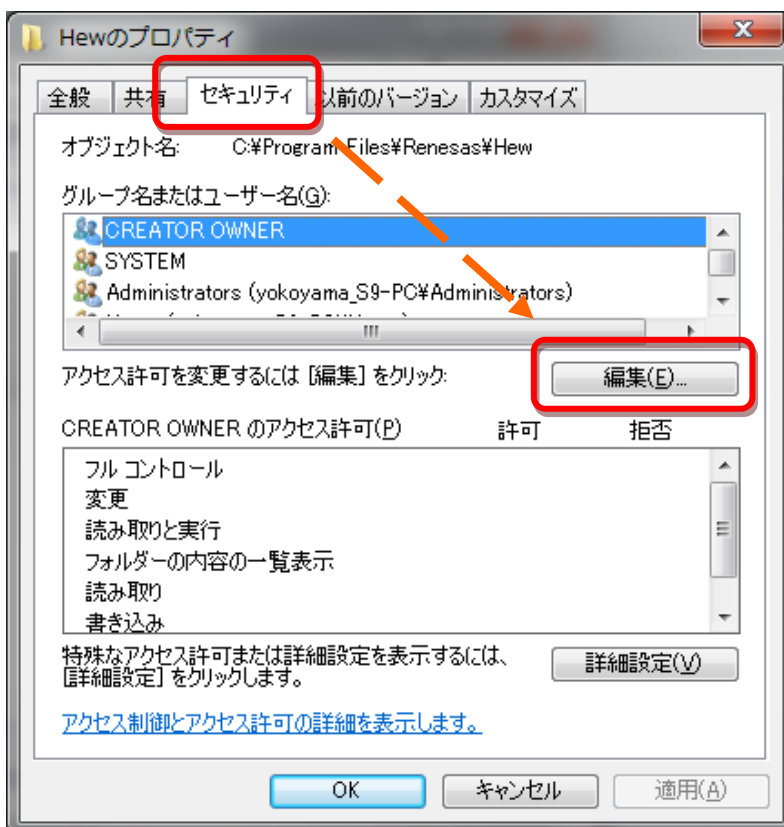
※Windows Vista または Windows7 の場合、セキュリティの都合上、Program Files フォルダ内のファイルが変更できない場合があります。以下の手順で、使用されているユーザーに対してHEW フォルダに対するフルコントロールを与えるか、Program Files 以外のフォルダにインストールすることで、36064s.h が編集可能になります。

※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。

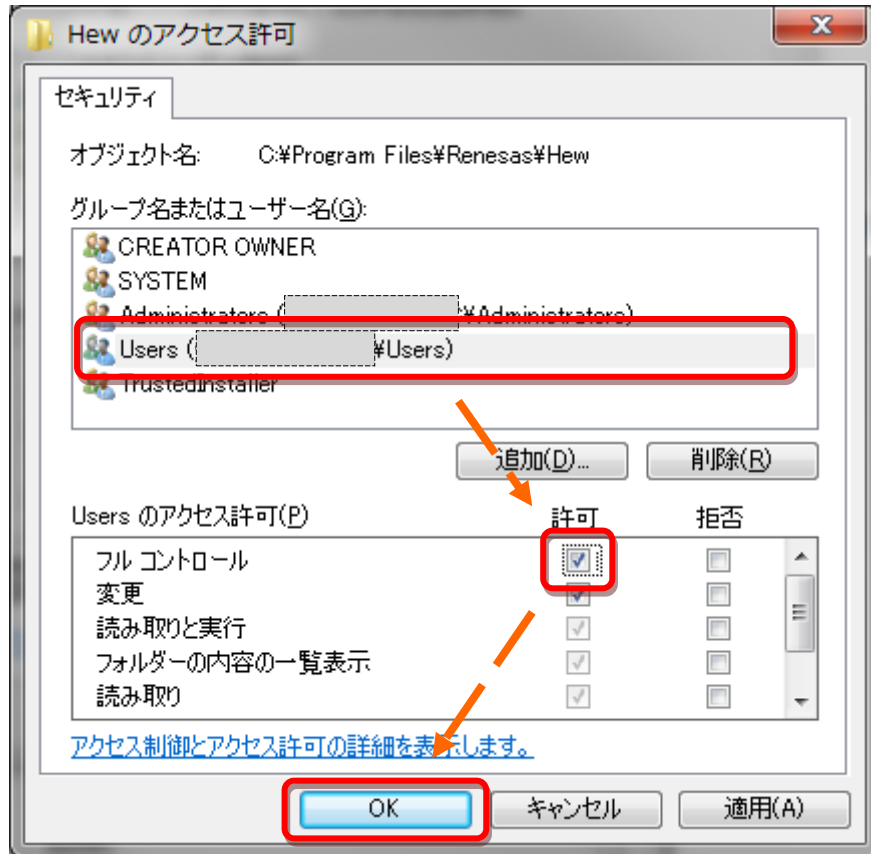
① 「C:\Program Files\Renesas\Hew」フォルダのプロパティを開きます。



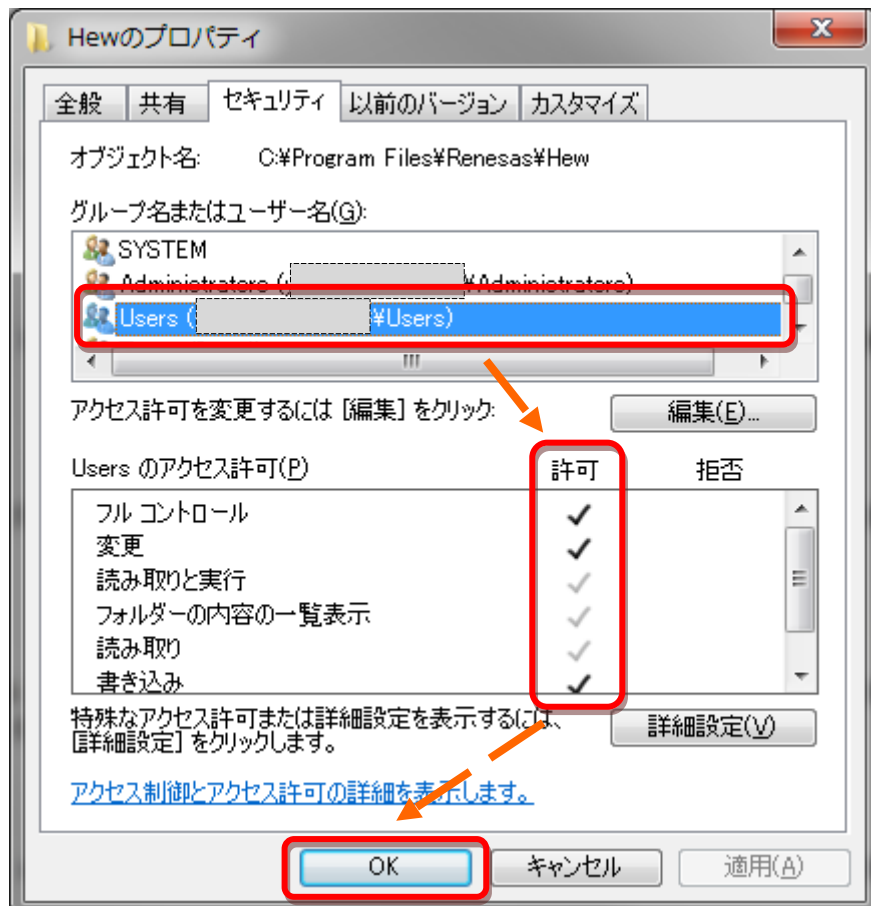
② セキュリティタブを開き、編集ボタンを押します。



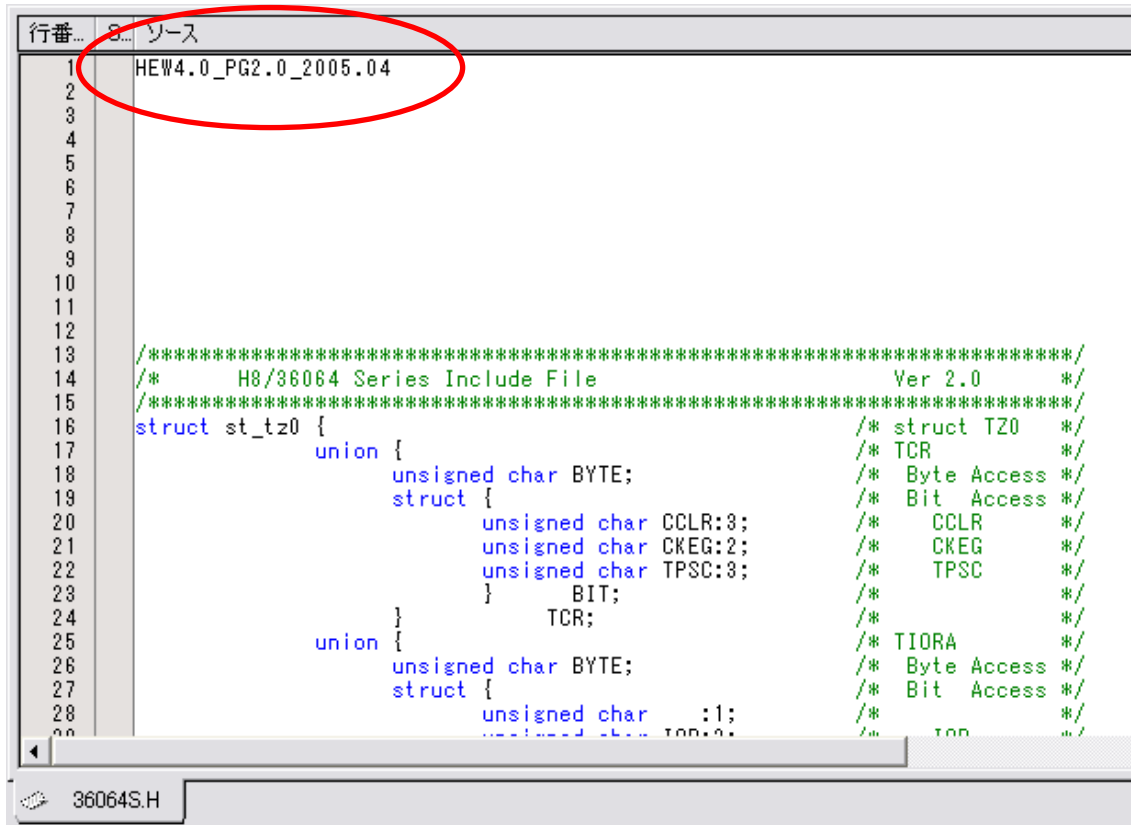
- ③ 現在使用しているユーザ（Users など）を選択し、フルコントロールの許可にチェックを入れ、OK をおします。



- 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。



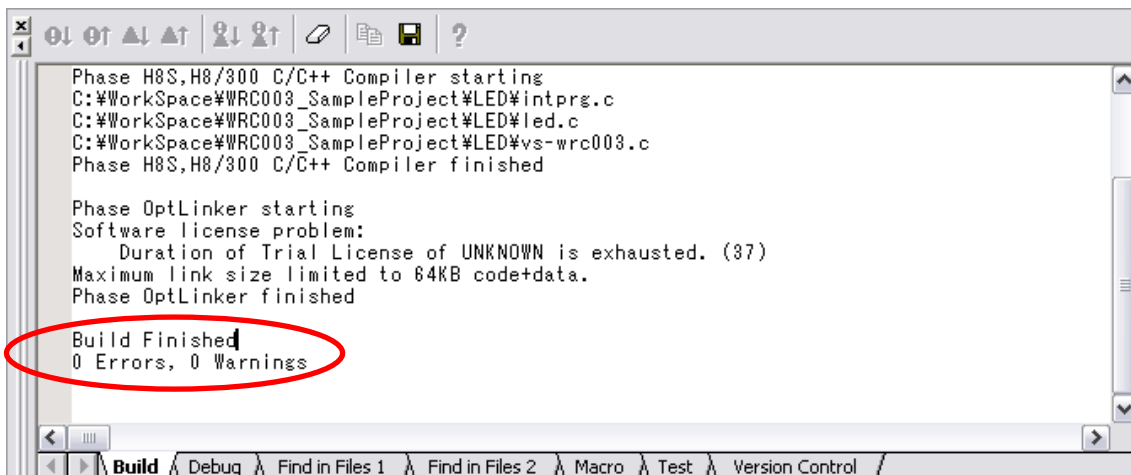
(8) エディタに 36064s.h が表示されると思います。この 1 行目にコメントアウトされていない一文がありますので、先頭に[/]を入力（コメントアウト）するか、1 行目を削除してください。



```
行番... 3... ソース
1 HEW4.0_PG2.0_2005.04
2
3
4
5
6
7
8
9
10
11
12
13
14 /******
15 /*      H8/36064 Series Include File                      Ver 2.0      */
16 /******
17 struct st_tz0 {                                           /* struct TZ0 */
18     union {                                              /* TCR */
19         unsigned char BYTE;                             /* Byte Access */
20         struct {                                         /* Bit Access */
21             unsigned char CCLR:3;                       /* CCLR */
22             unsigned char CKEG:2;                       /* CKEG */
23             unsigned char TPSC:3;                       /* TPSC */
24             } BIT;                                       /* */
25         } TCR;                                           /* */
26     union {                                              /* TIORA */
27         unsigned char BYTE;                             /* Byte Access */
28         struct {                                         /* Bit Access */
29             unsigned char :1;                            /* */
30             unsigned char TOP:3;                         /* TOP */
31         }
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-WRC003LV 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
```

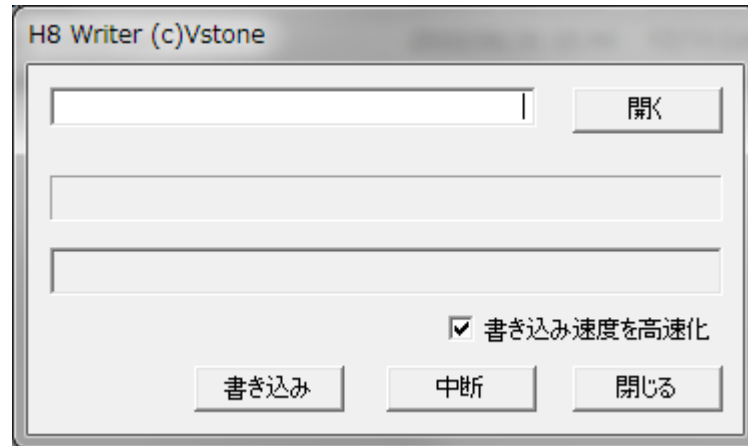
## 6-5 VS-WRC003LV へのプログラムの書き込み

VS-WRC003LV では書き込みに「H8Writer」という専用ソフトウェアを使用します。

ここでは、プログラムを書き込むための一通りの流れを解説します。

ソフトウェアの詳細については「「H8Writer」取扱説明書」をご覧ください。

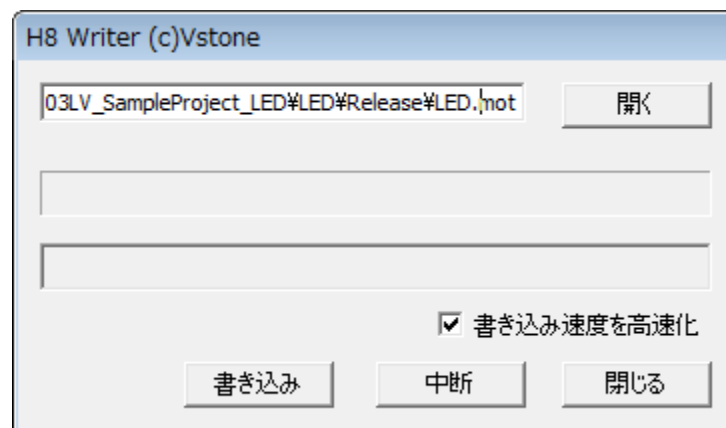
- (1) ビュートローバーのCD-ROM、または VS-WRC003LV サポートページから、「H8Writer.zip」をダウンロードし、任意の場所に解凍してください。
- (2) 解凍したフォルダを開き、H8Writer.exe を起動します。起動するといかのウィンドウを開きます。



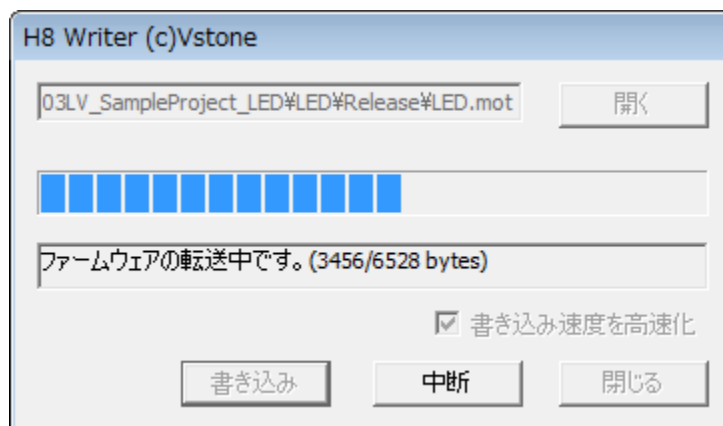
- (3) USB ケーブルで VS-WRC003LV と PC を接続します。

はじめて接続する場合、自動的にドライバがインストールされますので、インストールが完了するまで待ちます。

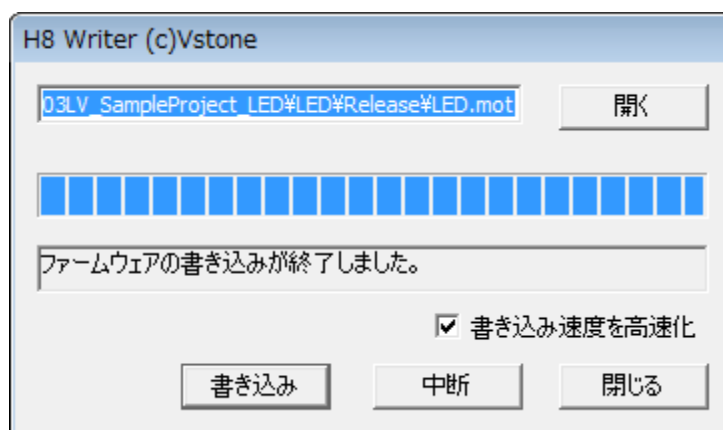
- (4) 「開く」をクリックし、VS-WRC003LV に書き込むバイナリファイルを選択するウィンドウを開きます。このウィンドウでサンプルプロジェクトフォルダ内の「¥LED¥Release¥LED.mot」を選択します。左の欄にファイルパスが表示されていることを確認してください。



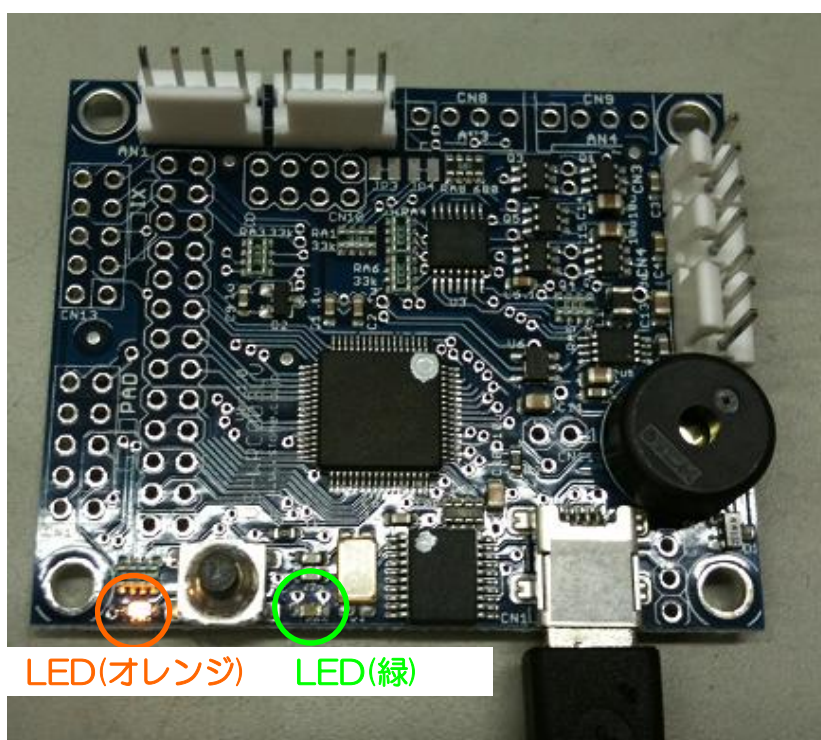
(5) 「書き込み」をクリックすると、選択したバイナリファイルを VS-WRC003LV に書き込みます。書き込みの実行中は下図のように書き込みの進行状況をウィンドウに表示します。



(6) 書き込みが完了すると、以下のような表示になります。



(7) 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。



## 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-WRC003LV.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 BYTE MainCycle = 60;
5:
6:     Init((BYTE)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-wrc003lv.h」で簡単に解説していますので、そちらをご覧ください。

## ■オプションパーツ、関連商品のご購入は・・・

No.1 の品揃え！ 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