

# モデルカーを用いたAUTOSAR開発入門 開発環境

名古屋大学 大学院情報科学研究科  
附属組込みシステム研究センター  
人材育成プログラム (NEP)

最終更新日：2015/10/26

---

# 開発環境

- **インストール**
- アプリケーションのビルド
- アプリケーションの実行

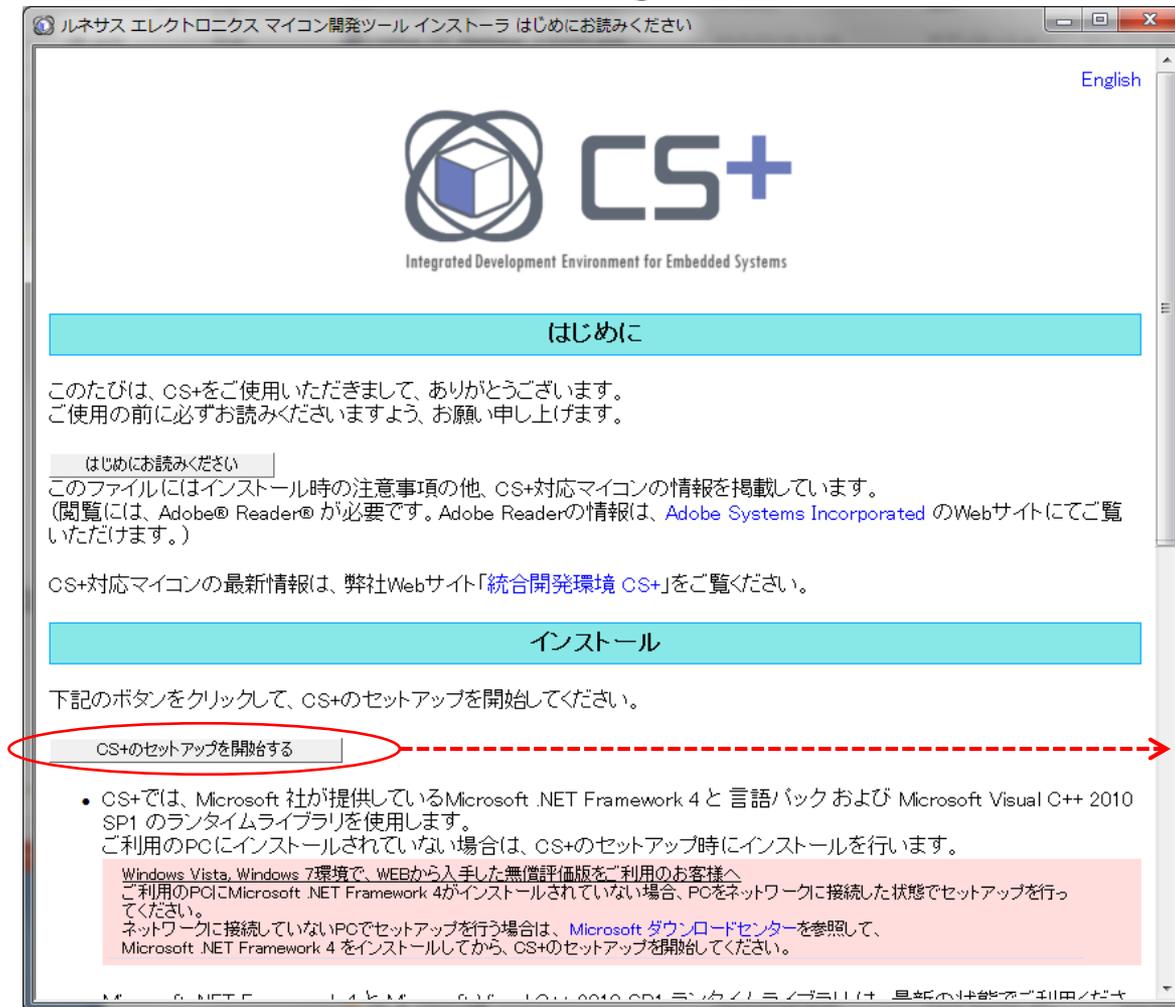
# 必要なソフトウェア環境

---

- 統合開発環境
  - CS+ for CC V3.00.00(無償評価版)
  - ダウンロード先：  
<http://japan.renesas.com/products/tools/ide/csp/downloads.jsp>
  - my renesas への登録が必要
- コンパイラ
  - [H850コンパイラ CC-RH V1.02.00](#)
  - CS+をインストール後にupdateを実施することでインストール
  - 生成可能なプログラムサイズは256KBまで
- E1用ドライバ
  - CS+と同時にインストールされる
- その他のソフトウェア
  - ターミナルソフトウェア(本教材の実行例ではTeraTermを使用)
  - Ruby
    - Ruby Installer(<http://rubyinstaller.org/>)によりインストール

# CS+のインストール

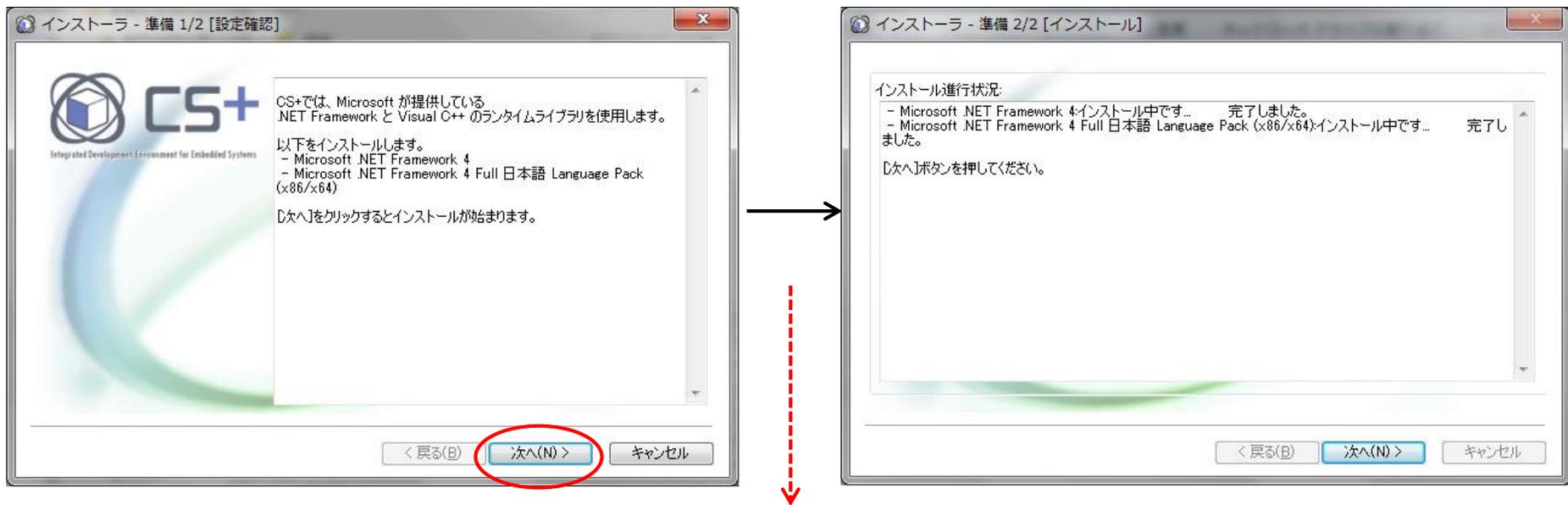
- インストーラの起動
  - CSPlus\_CC\_Package\_V30000.exe



“CS+のセットアップを開始する”を押下する

# CS+セットアップ前準備

- CS+本製品が動作するためにインストールが必要な環境
  - Microsoft .NET Framework 4
  - Microsoft .NET Framework 4 日本語言語パック
- 上記が未インストールの場合は下記のウィンドウが表示される

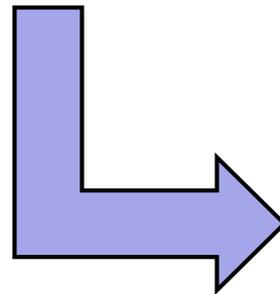


- ウィンドウの手順に従い進め、事前準備のセットアップ完了後にCS+のインストールに移る

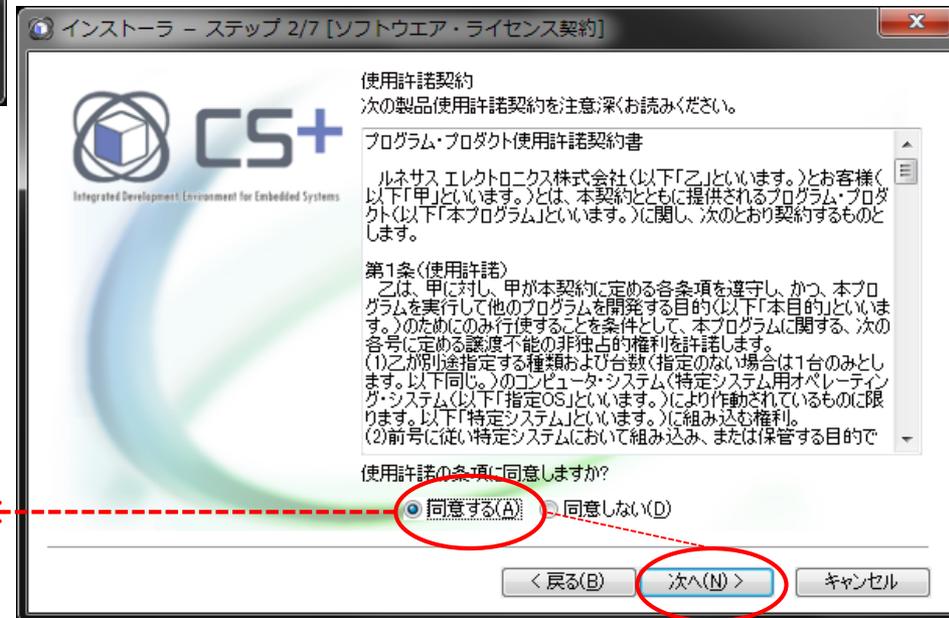
# CS+のインストール



必要な環境がすべてインストール済みの場合、インストーラ起動後に左記のウィンドウから表示される

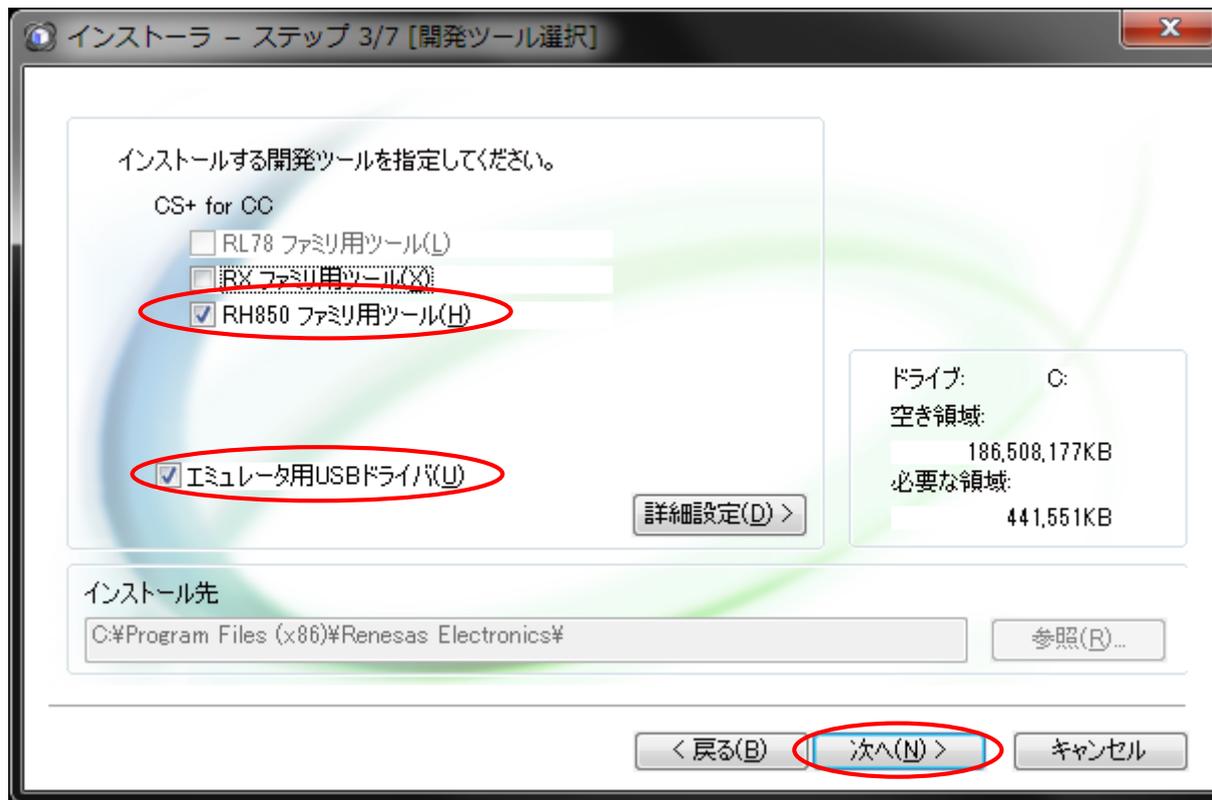


“同意する”にチェックを入れて  
“次へ”を押下する

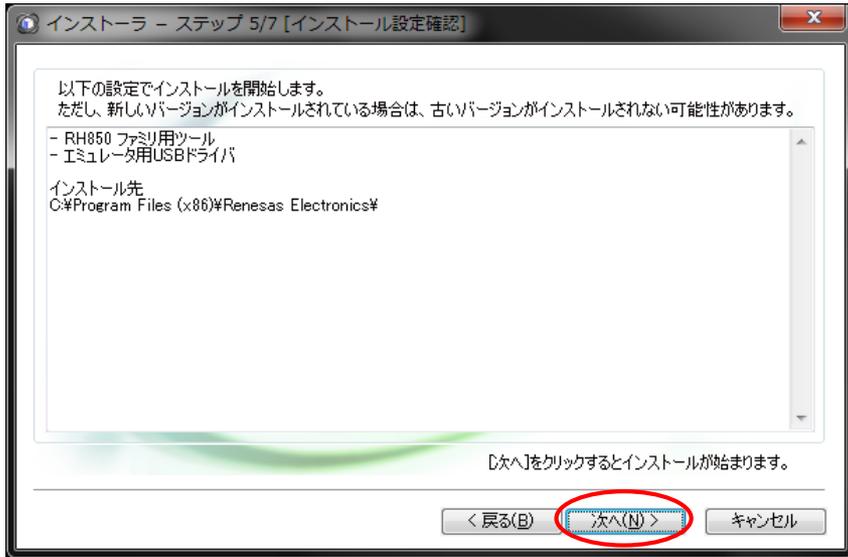


# CS+のインストール

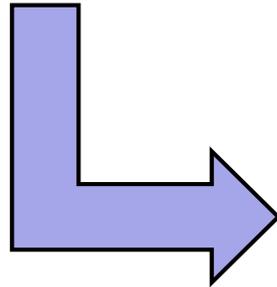
- 本演習ではV850を使用するため“RH850 ファミリ用ツール一式”と“エミュレータ用USBドライバ”にチェックを入れて“次へ”を押下する
  - 開発ツールを細かく指定する場合は[詳細設定]で設定が可能となる
- インストール先を指定する場合は“参照”を押下する



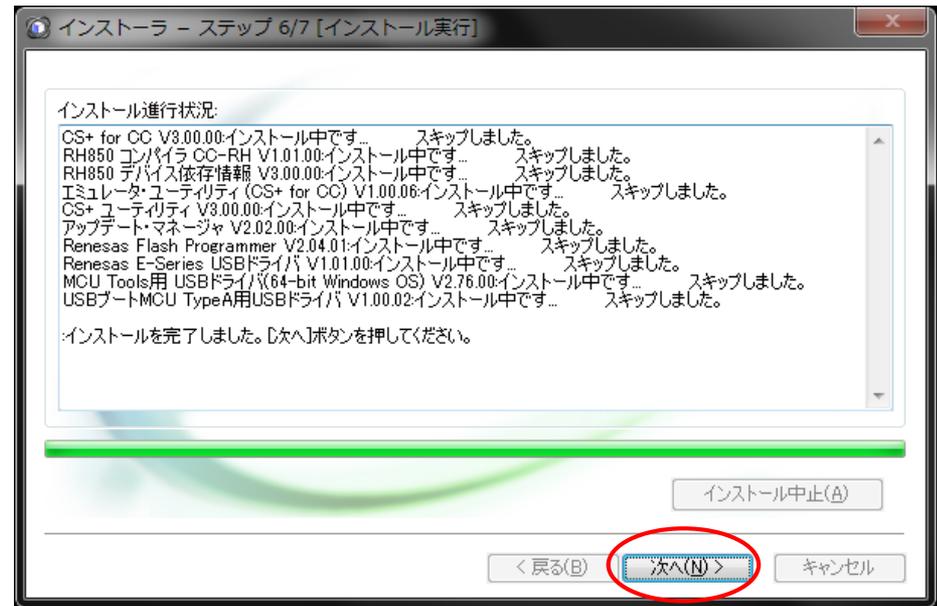
# CS+のインストール



開発ツール選択 ウィンドウ, または  
ツール詳細設定 ウィンドウで設定した  
内容が表示されるため“次へ”を押下  
するとインストールが開始する

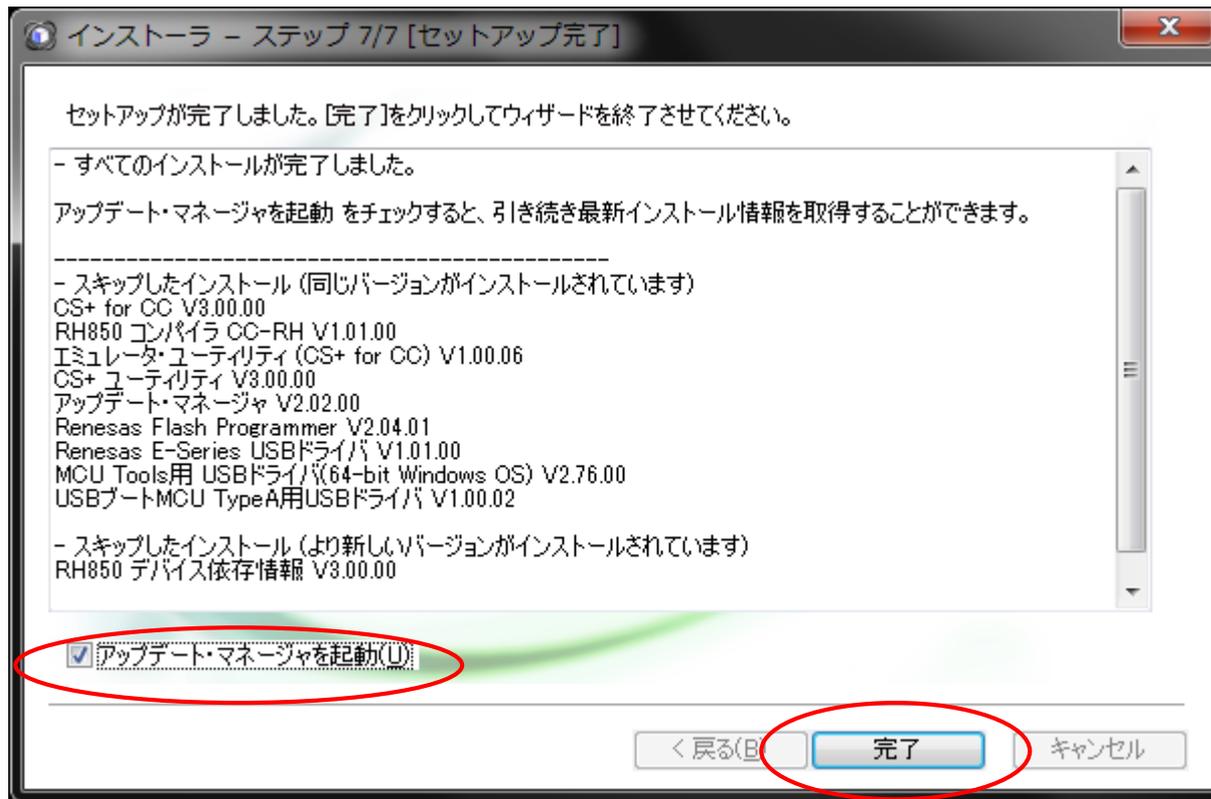


インストール完了後は“次へ”を  
押下する



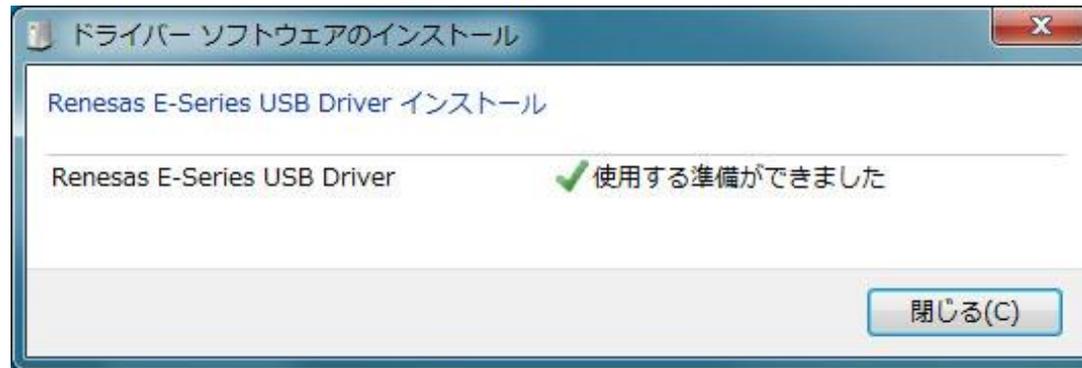
# CS+のインストール

- 下記のウィンドウが表示され, “完了”押下するとインストールは終了する
  - ツールやドキュメントのアップデートが不要な場合は“アップデートマネージャを起動”のチェックを外す



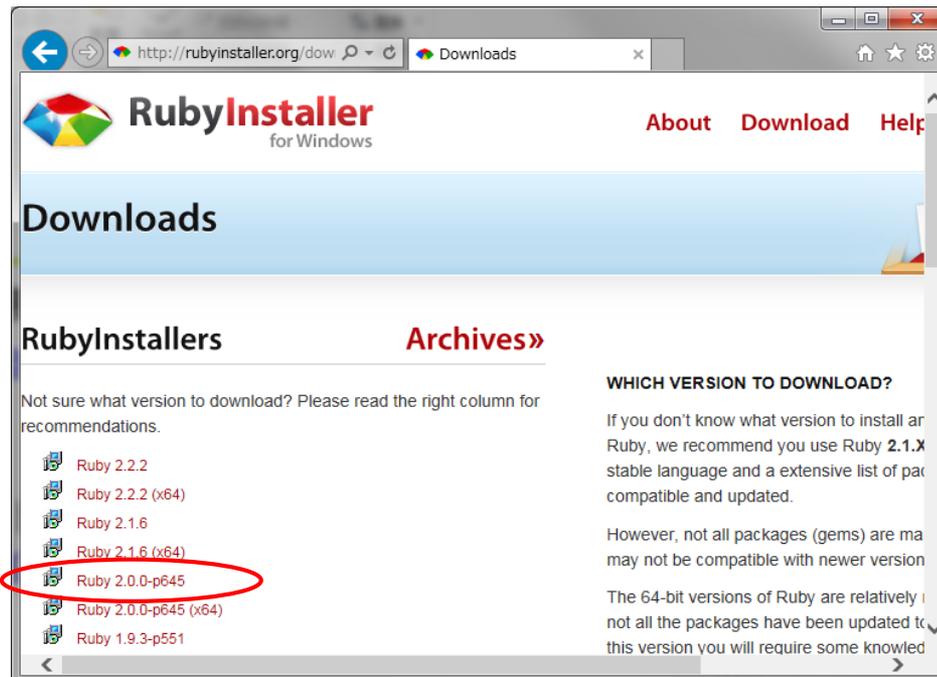
# E1用ドライバ

- E1用ドライバはCS+のインストーラと同時にインストールされる
- E1をPCに接続すると以下のようにドライバが自動的にインストールされる。



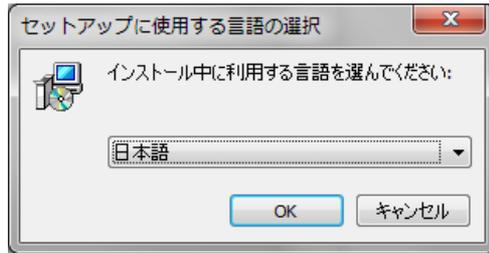
# Rubyのインストール(1/2)

- ビルドの課程においてRubyが必要となる
- Ruby Installer(<http://rubyinstaller.org/>)によりインストール
- インストール手順
  - Ruby Installerのダウンロードサイト(<http://rubyinstaller.org/downloads/>) から2.0.0をダウンロード
  - インストーラを実行

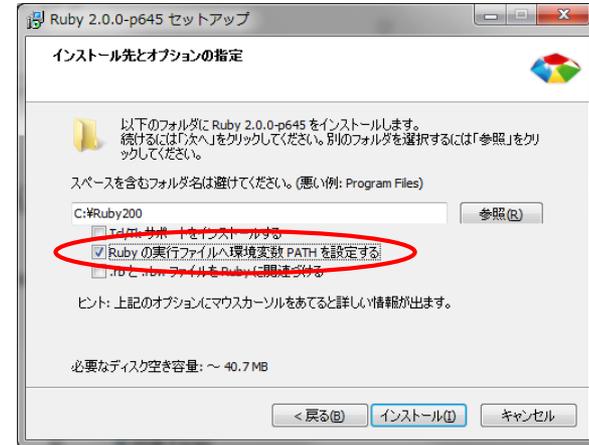


# Rubyのインストール(2/2)

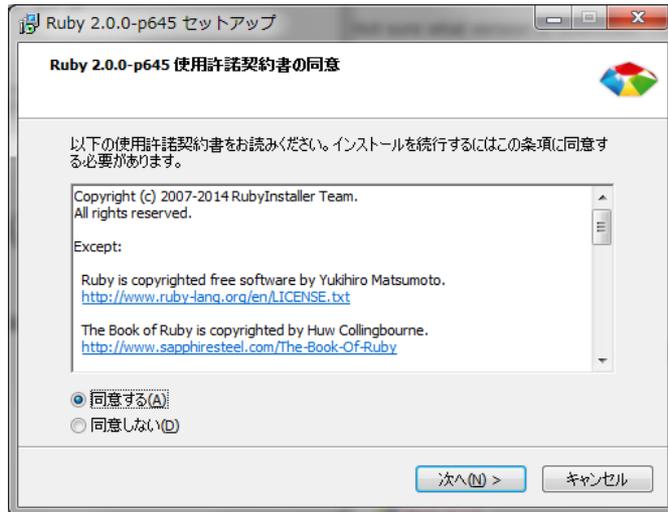
- 言語を選択



- インストール先にパスを通すように指定する



- 利用許諾に合意



- 終了



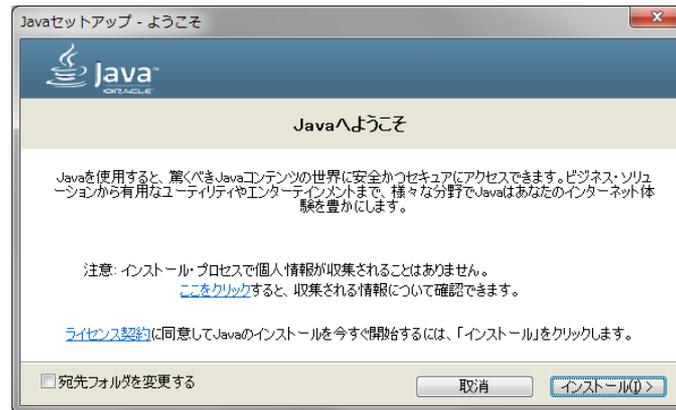
# Java Runtime Environment(JRE)のインストール

- ビルドの課程においてJREが必要となる
  - 対応バージョン：1.7.0\_45以降
- インストーラのダウンロード
  - <https://java.com/ja/download/>にアクセス。
  - 利用条件に合意してインストーラをダウンロード

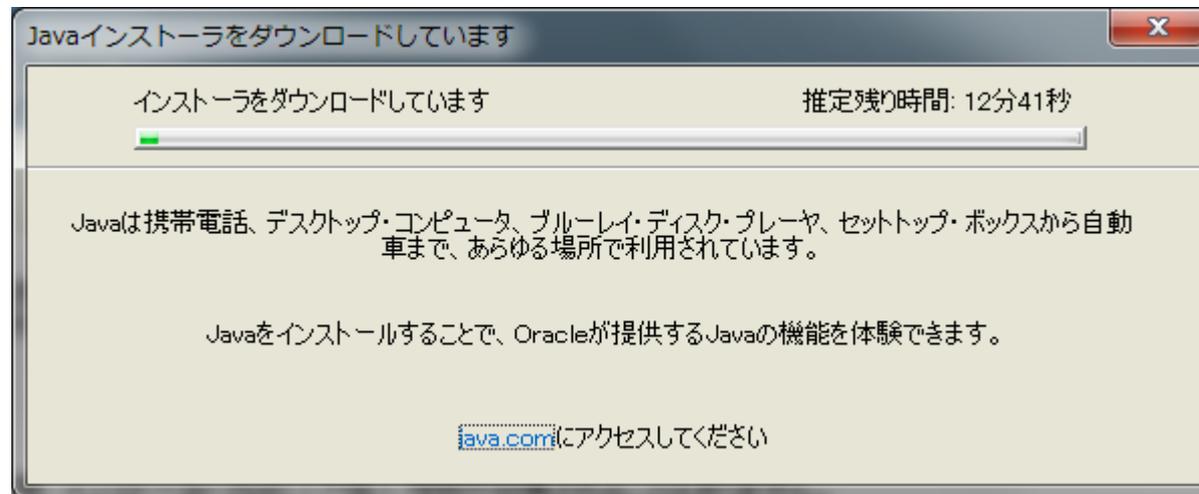


# Java Runtime Environment(JRE)のインストール

- インストーラを実行



- インストーラ本体がダウンロードされインストールされる



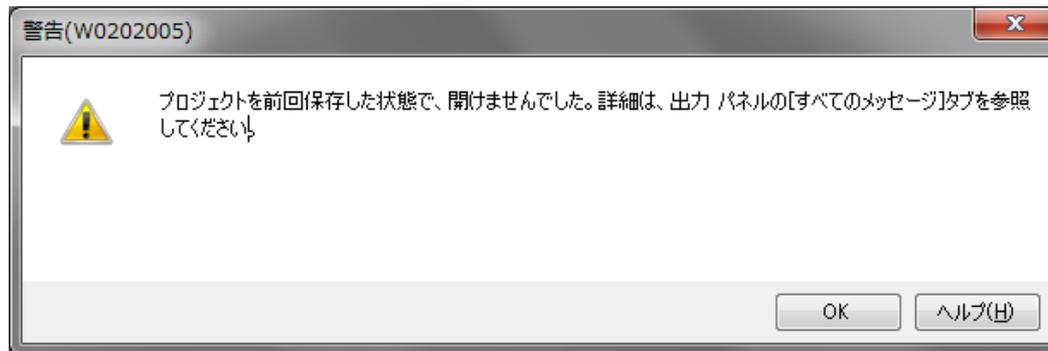
---

# 開発環境

- インストール
- **アプリケーションのビルド**
- アプリケーションの実行

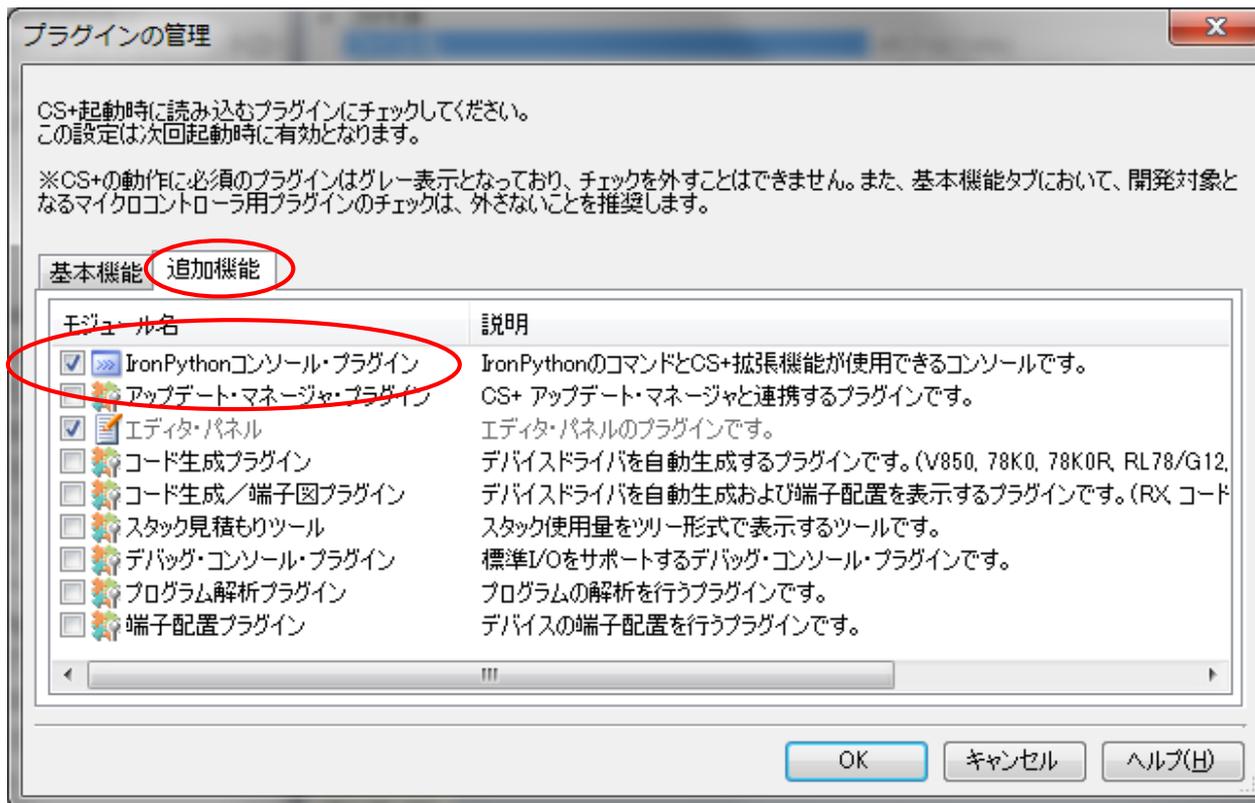
# CS+の起動

- AUTOSAR OS対応のモデルカー制御プログラムをビルドする
  - ¥os-application¥obj¥configure.mtpjをCS+で開く
  - CS+が起動し, atk2-sc1.mtpjが作成され開かれる
    - 次回以降は, atk2-sc1.mtpjを開けば良い.
    - 本スライドはCS+の実行方法に関する内容のため, 本プログラムの詳細は省略する
- CS+の起動時に, まれに下記の警告のポップアップが出現する場合があるが, 動作に影響はないため"OK"を押下する



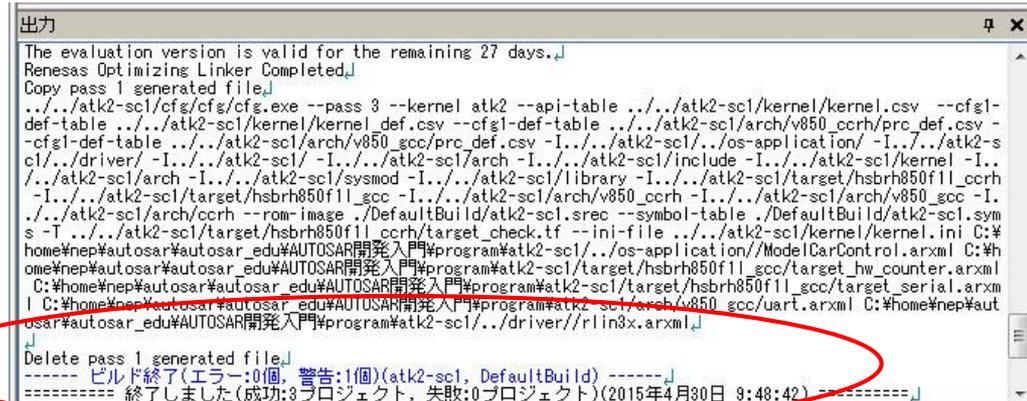
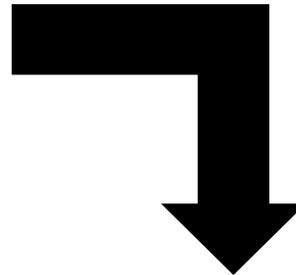
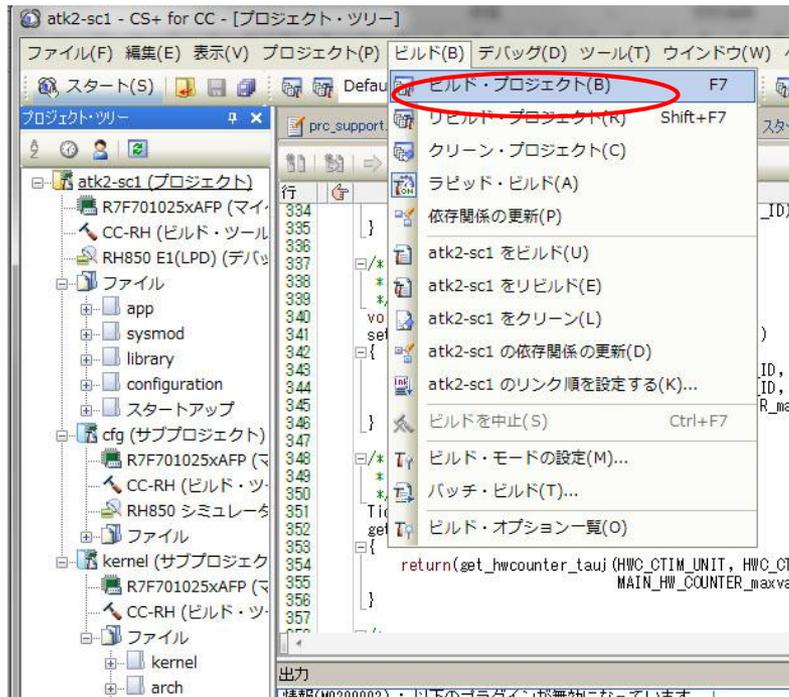
# CS+の起動：プラグインのチェック

- TOPPERS/ATK2のビルドには、CS+のIronPythonコンソール・プラグインが有効になっている必要がある
- CS+のメニューの“ツール”→“プラグインの管理”で下記の画面を表示してIronPythonコンソール・プラグインのチェックが入っていればよい。チェックが入っていない場合はチェックしてOKを押す。



# CS+によるビルド

- CS+実行後にビルドのメニューから“ビルド・プロジェクト”を選択する
- 出力ウィンドウにてビルドが成功したことを確認する



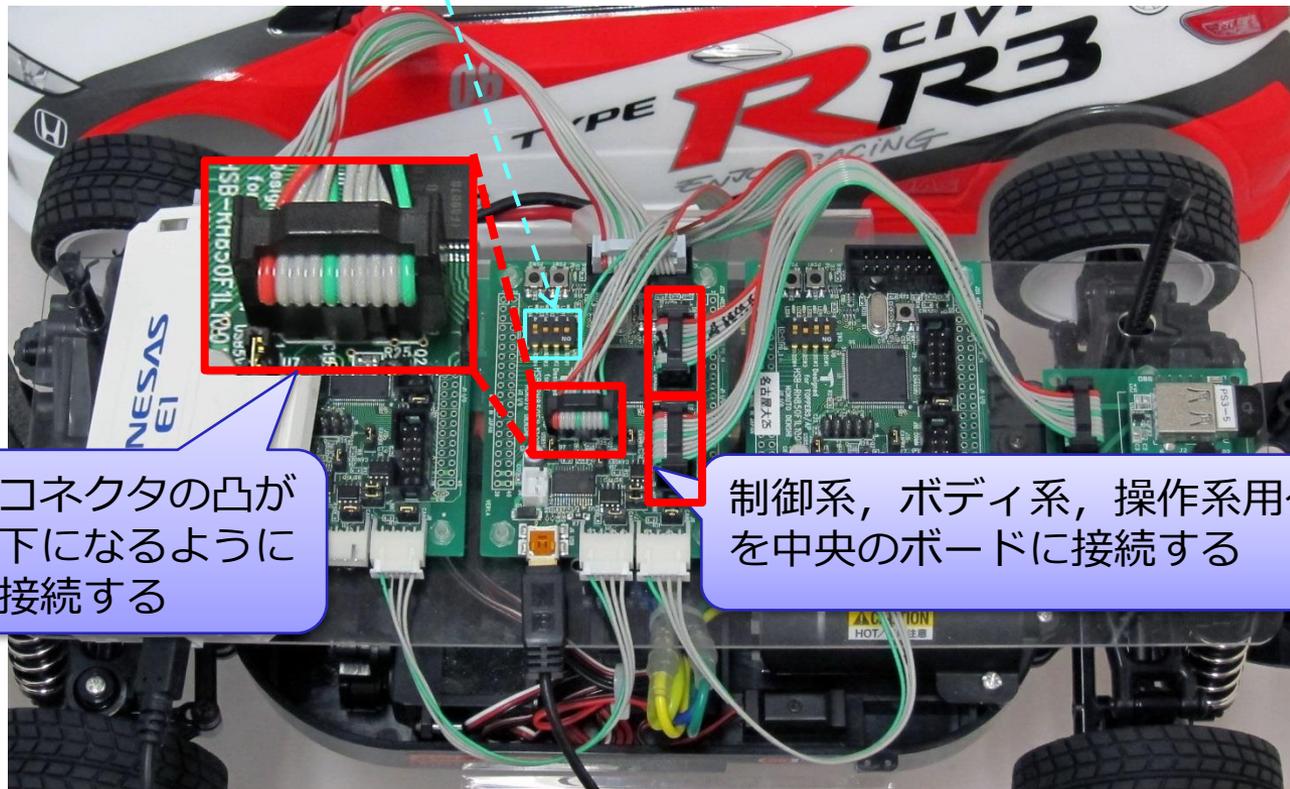
---

# 開発環境

- インストール
- アプリケーションのビルド
- **アプリケーションの実行**

# マイコンボードのセットアップ

- 使用するマイコンボードの接続
  - 3台搭載しているマイコンボードのうち、中央の1台に制御系、ボディ系、操作系用ケーブルを接続する
  - DSW1~3をONにしておく

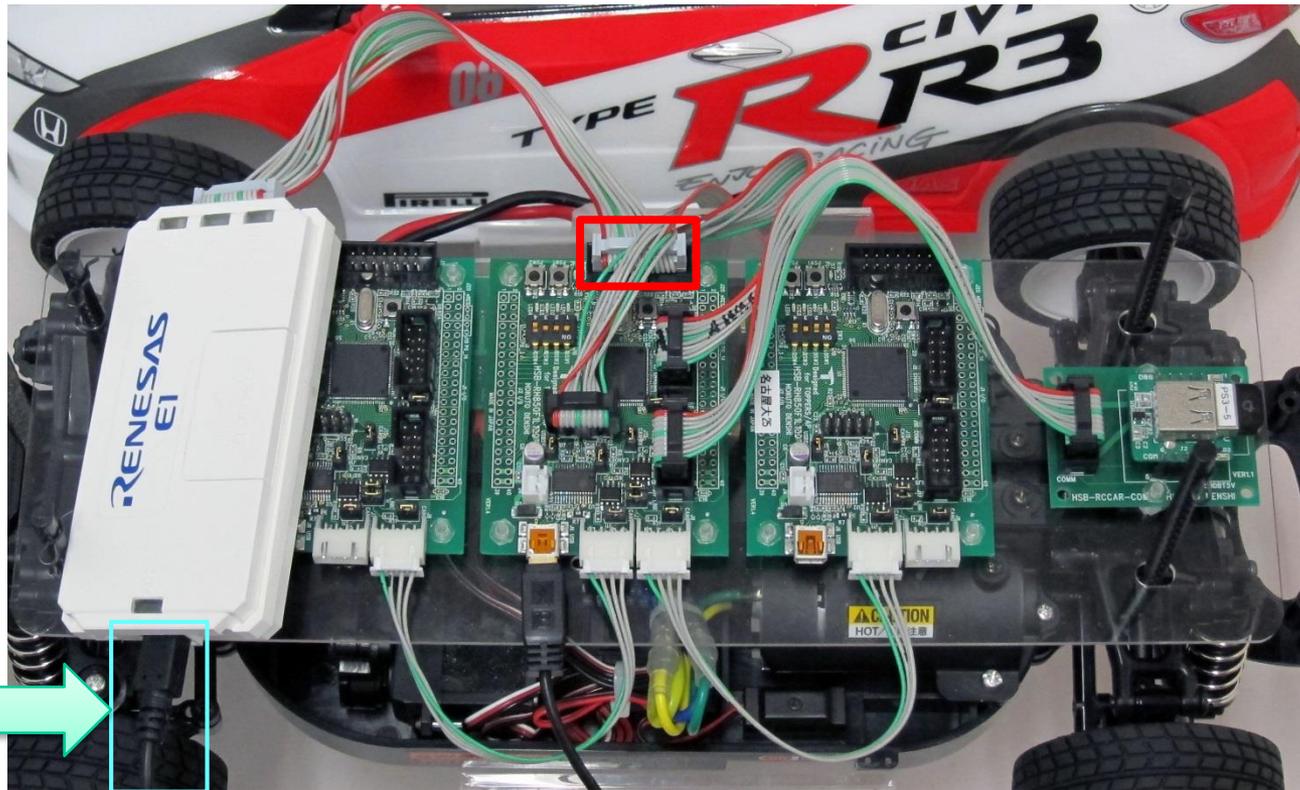


コネクタの凸が下になるように接続する

制御系、ボディ系、操作系用ケーブルを中央のボードに接続する

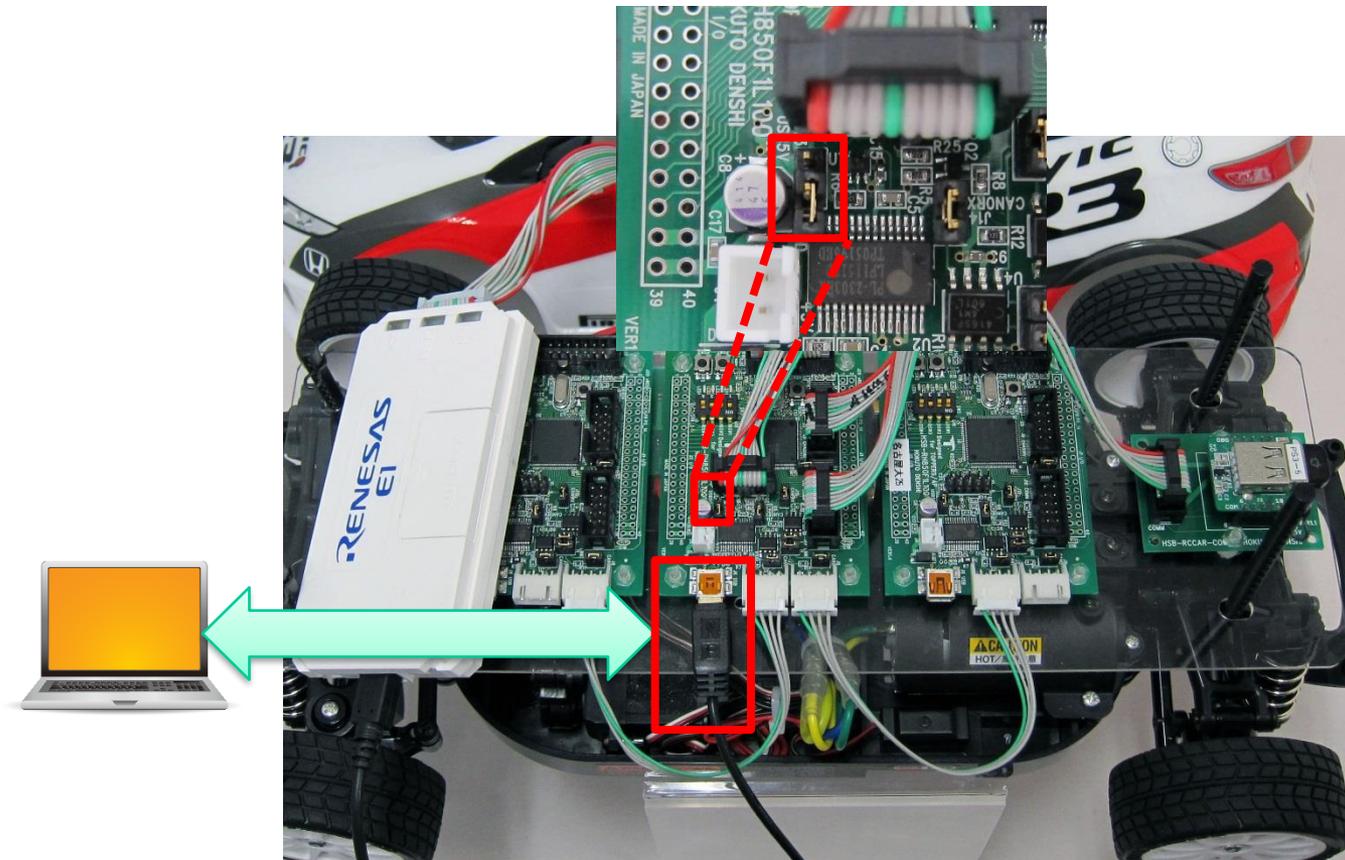
# マイコンボードのセットアップ

- E1とボードをインターフェースケーブルで接続する
- E1側のUSBミニB端子とPC間をUSBケーブルで接続する



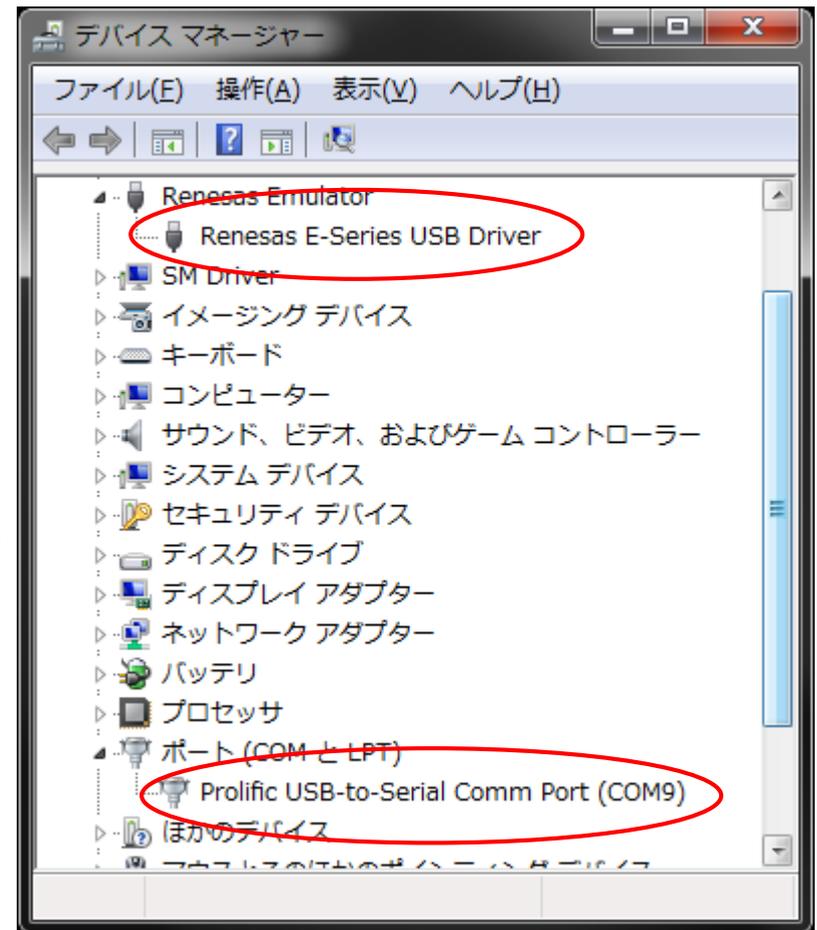
# マイコンボードのセットアップ

- J13のジャンパをショートするとUSBケーブル経由で給電されるため、オープンにしておく。
- CPUボード上のUSBミニB端子をUSBケーブルでPCと接続する

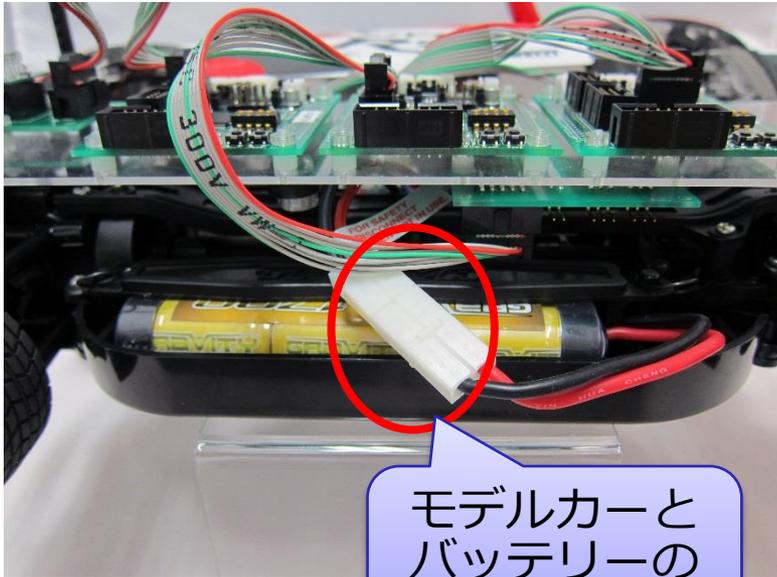


# マイコンボードのセットアップ

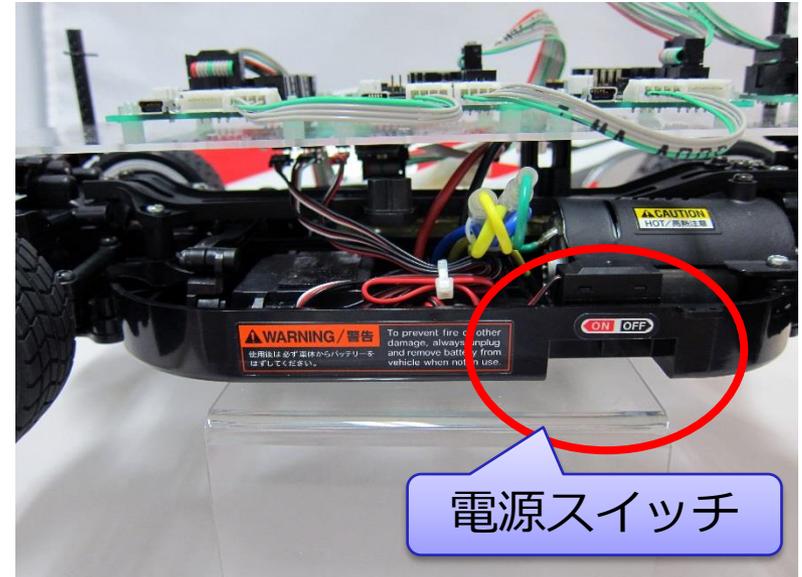
- マイコンボードを接続すると, prolific usb-to-serial comm portと認識される
- E1接続時にデバイスマネージャにて“Renesas E-Series USB Driver”と認識されていればOK
- マイコンボードを接続した際に, デバイスマネージャで認識されない (prolific usb-to-serial comm portが表示されない)場合
  - 使用しているPCにUSB Serial Portのドライバをインストールする



# モデルカーへの電源投入



モデルカーと  
バッテリーの  
接続



電源スイッチ

- モデルカーの下部(シャシー側中央)にある電源ケーブルをバッテリーに接続する (コネクタは使用しない場合は取り外す)
- 電源スイッチをONにする

# PS3コントローラの接続



“1”の箇所が赤く点灯していれば接続状態  
全消灯の場合は接続解除状態



- PS3コントローラとモデルカーの接続
  - モデルカーの電源投入後，PS3コントローラのPSボタンを押下する
  - コントローラ上部にあるLED (4つ)が点滅中はモデルカーとの接続同期待ちで，“1”のLEDが点灯したら接続が完了となる
    - 接続中にPS3コントローラの充電はできないので注意
- PS3コントローラとモデルカーの接続解除
  - PSボタンを長押し(10秒程度)でPS3のコントローラのみOFFとなる
  - コントローラ上部にあるLEDが全て消灯していれば接続解除となる

# TeraTermの実行

- TeraTermは、シリアルポートから送られてくるデータを表示する
- ソフトウェア側の設定(スピード[BPS]やデータフォーマット等)に合わせる必要がある
- ポートの選択は、ボードが接続しているCOMポート番号に設定する



# TeraTermの実行

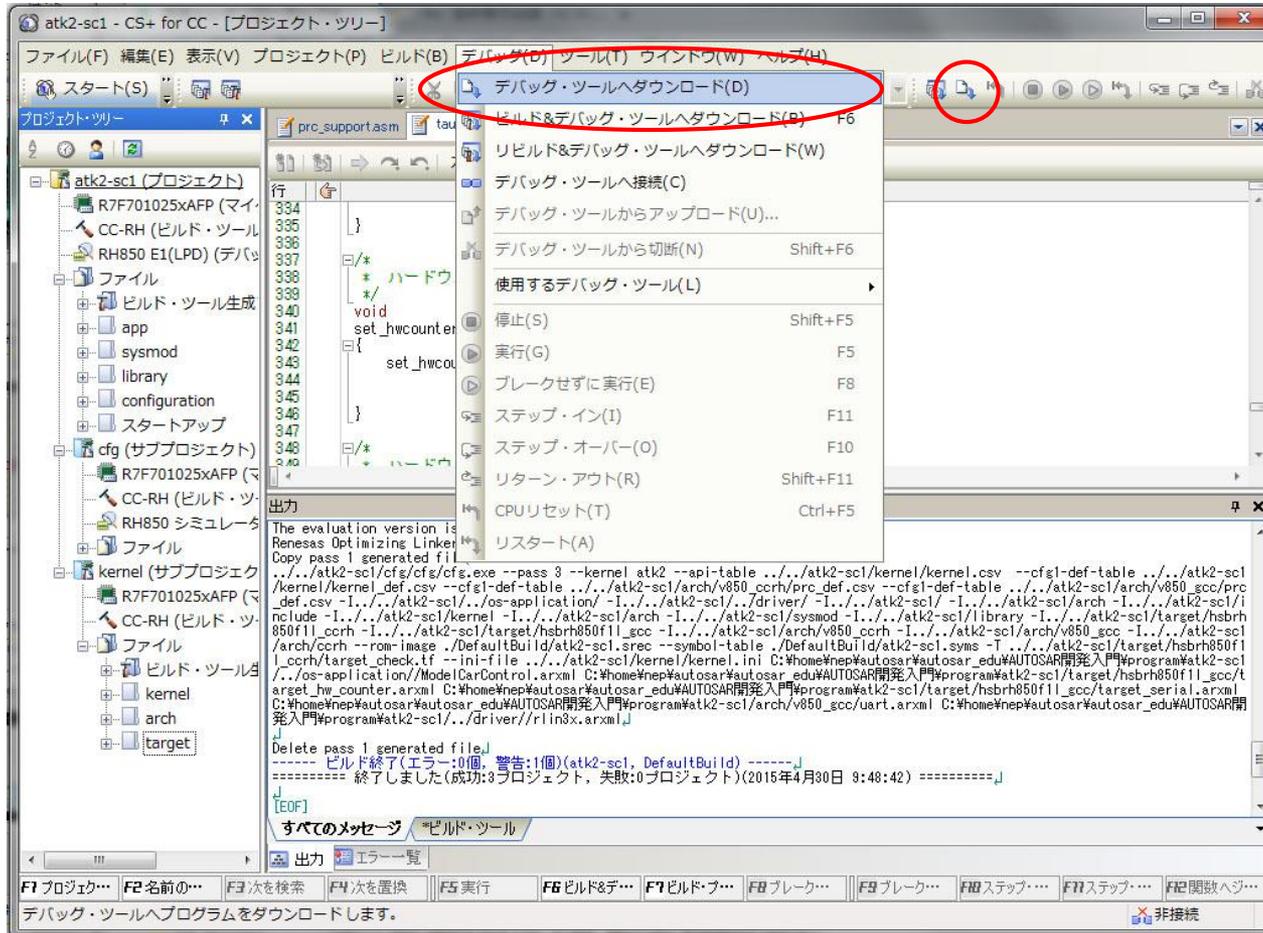
- シリアルポートの設定：メニューの“設定” → “シリアルポート”
  - ポート : COMポートの番号に設定
  - ボーレート : 115200(bps)
  - データ : 8 bit
  - パリティ : none,
  - ストップ : 1 bit,
  - フロー制御 : none



- 設定後メニューの“設定” → “設定の保存”で次回から設定の省略が可能

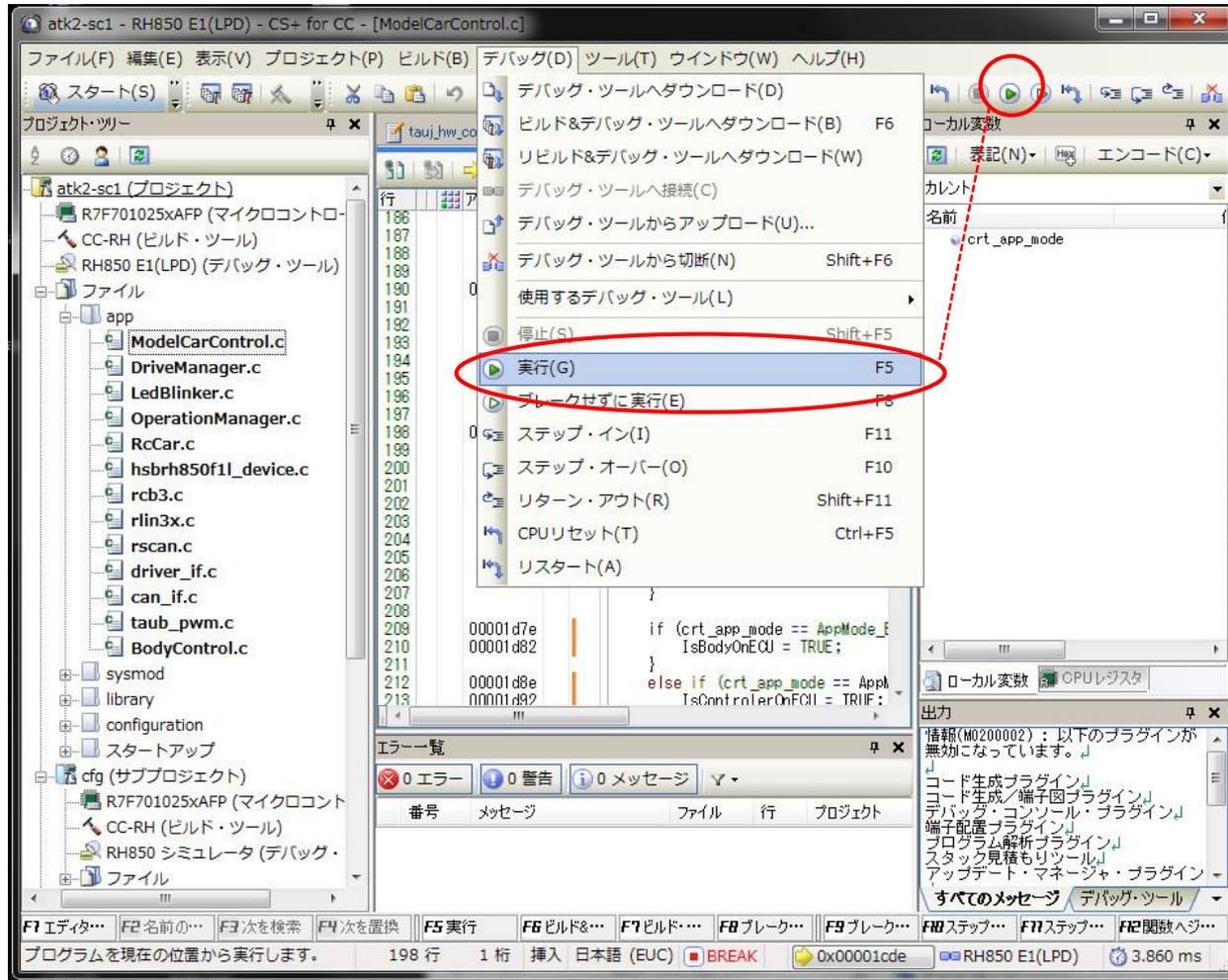
# プログラムのダウンロード

- デバッグから“デバッグ・ツールへダウンロード”を選択する
- メニューバーからも可能



# プログラムの実行

- デバッグから“実行”もしくは  を押下することでプログラムの実行が開始される

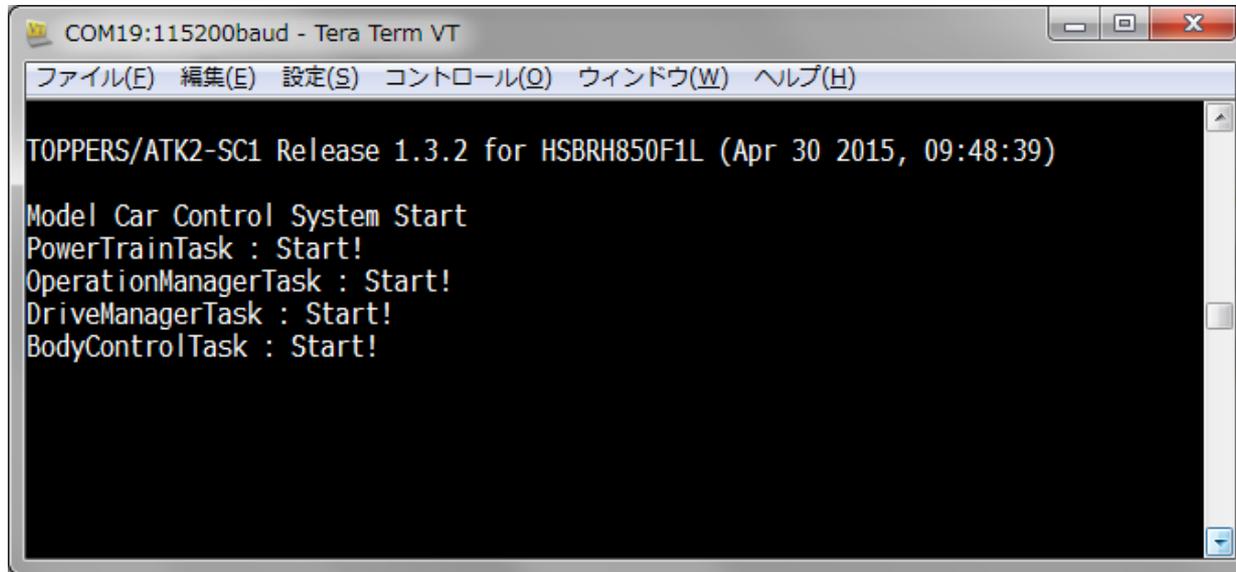


# プログラム実行と動作確認

- プログラム実行時にCS+のウィンドウ下部には、



が表示され、TeraTermには出力ログが表示される



- 動作確認
  - PS3コントローラを操作して仕様通りモデルカーが動作するか確認する
  - DSW1~3全てをOFFしている等、設定が正しくない場合は、LEDが全点灯してバーナーが表示されないため注意すること

# パラメータ調整

- ESCの種類によってはESCのパラメータ調整が必要
  - 動作確認済みはタミヤTLU-105BK/TEU-104BK
  - 同じESCであっても個体差によりパラメータが異なる場合がある
- ./driver/rc.h の以下のマクロを設定する
  - ブレーキが効かずにバックする場合は以下のマクロを増加してみる
    - ESC\_DTH\_100NS : 100単位で増加
    - ESC\_WAIT\_D2B\_WAIT\_10MS : 1単位で増加

```
/* TLU104BK用のパラメータ */                                     <rc.h>
#ifdef ESC_TLU104BK
#define ESC_N_100NS                15000
#define ESC_DMAX_100NS             14000
#define ESC_RMAX_100NS             17000
#define ESC_DTH_100NS              14700
#define ESC_RTH_100NS              15300
#define ESC_D2R_100NS              16500
#define ESC_BREAK_100NS            18000
#define ESC_WAIT_N2R_WAIT_10MS     4 /* ニュートラルからリバースまでの待ち時間(10MS単位) */
#define ESC_WAIT_D2B_WAIT_10MS     10 /* ドライブからブレーキまでの待ち時間(10MS単位) */
#define ESC_WAIT_D2NR_WAIT_10MS    8 /* ドライブからリバースする際のニュートラルまでの待ち時間(10MS単位) */
#endif /* ESC_TLU104BK */
```