

TOPPERS/ATK2プログラミング演習

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)

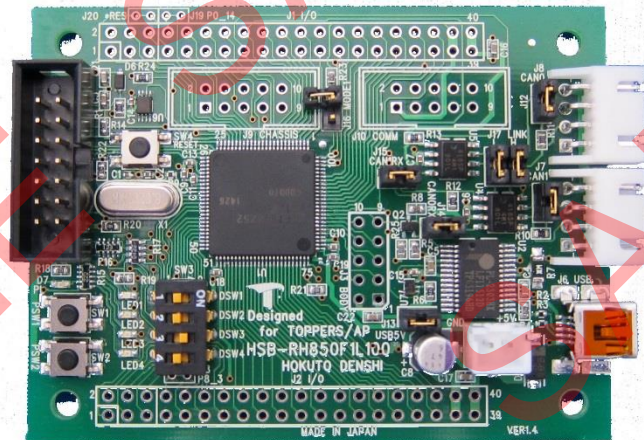
最終更新 : 2016/01/08

TOPPERS/ATK2プログラミング演習

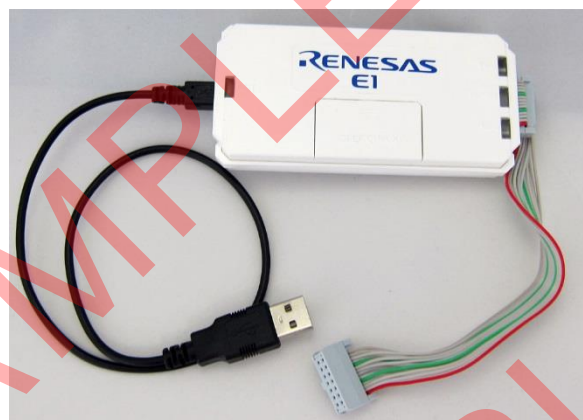
- **開発・実行環境の使用法**
- タスク管理機能
- 割込み管理機能
- 同期機能(イベント, リソース)
- アラーム, カウンタ
- その他

必要なハードウェア環境

- 北斗電子製 HSBRH850F1L100



- E1(オンチップデバッキングエミュレータ)



- USBケーブル(MINI-B) x 2

対象マイコンとプロセッサについて

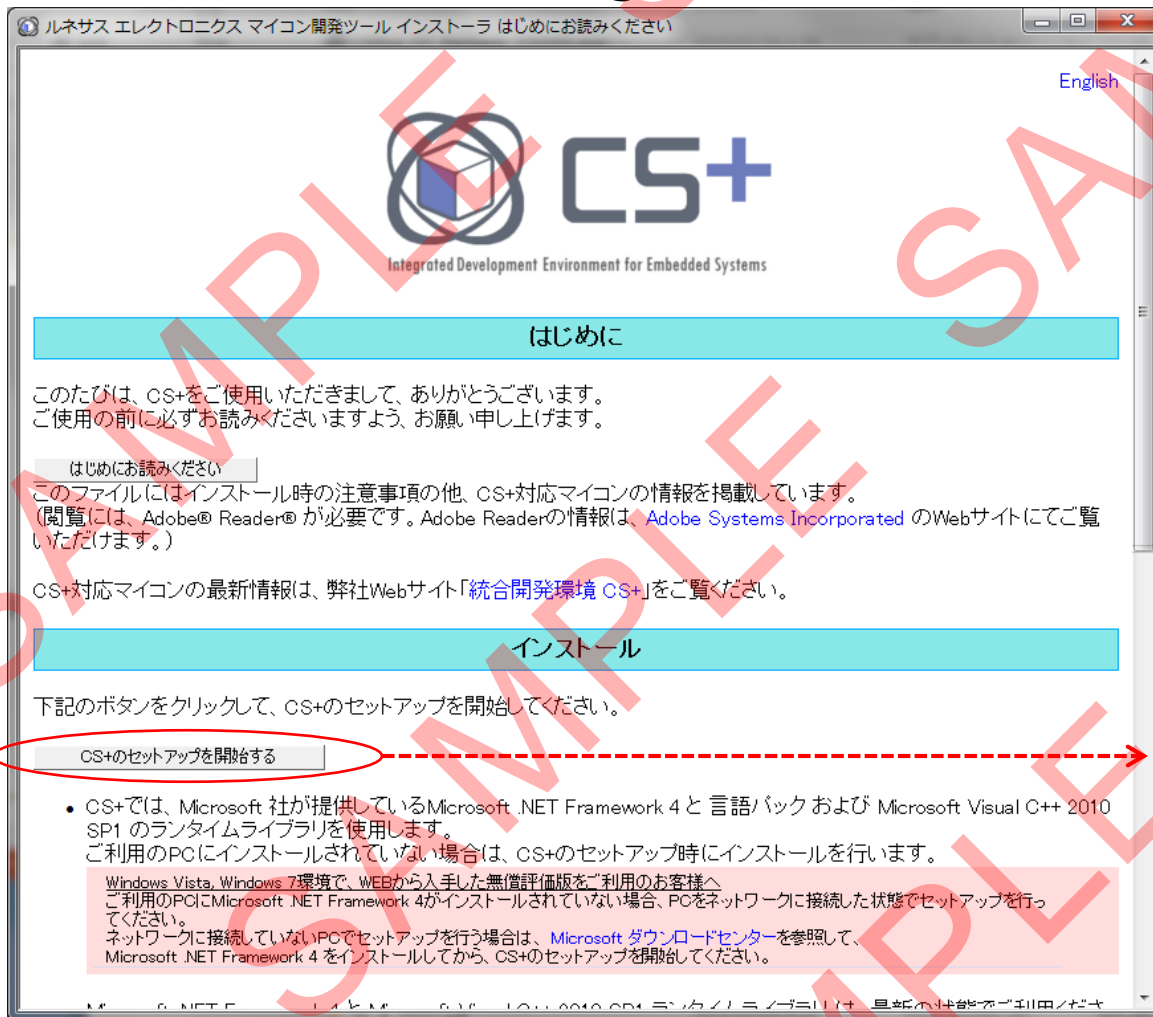
- マイコン
 - ルネサスエレクトロニクス社製 RH850/F1L
 - 車体アプリケーション（ドアモジュール、ライティング、空調システム（HVAC）、BCM (Body Control Module)）等を対象としている。
 - 上位マイコンとしてデュアルコアのRH850/F1Hもラインナップ
 - CAN等はV850世代とは互換性がない
- プロセッサアーキテクチャ RH850(G3K)
 - V850E3v5アーキテクチャを実装したプロセッサ
 - V850をベースに命令セットを拡張しているため、V850とのソフトウェアの互換性は高い。
 - 上位コアとしてG3MやG3MHが存在
 - FPUやロックステップをサポート

必要なソフトウェア環境

- 統合開発環境
 - CS+ for CC V3.01.00(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
 - RH850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
 - Ruby
 - Ruby Installer(<http://rubyinstaller.org/>)よりインストール
 - ターミナルソフトウェア(本教材の実行例ではTera Termを使用)

CS+のインストール

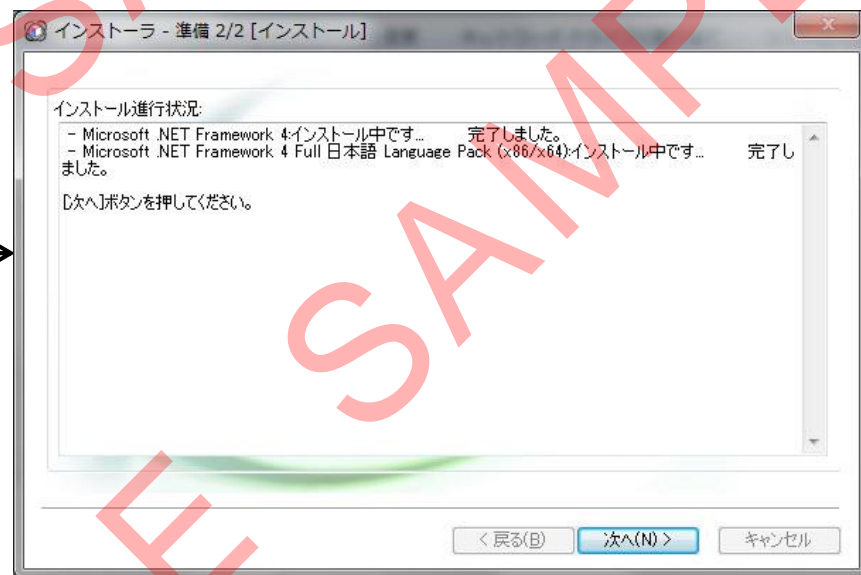
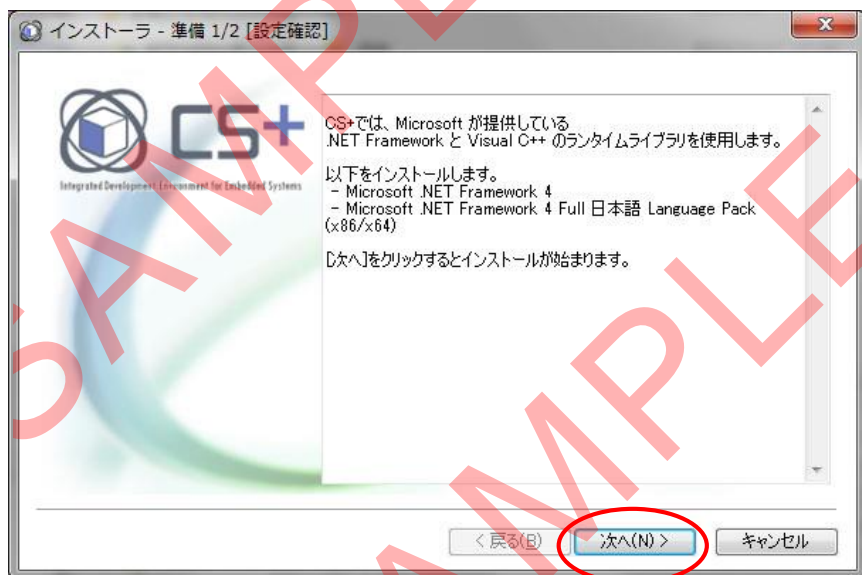
- インストーラの起動
 - CSPlus_CC_Package_V30100.exe



“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

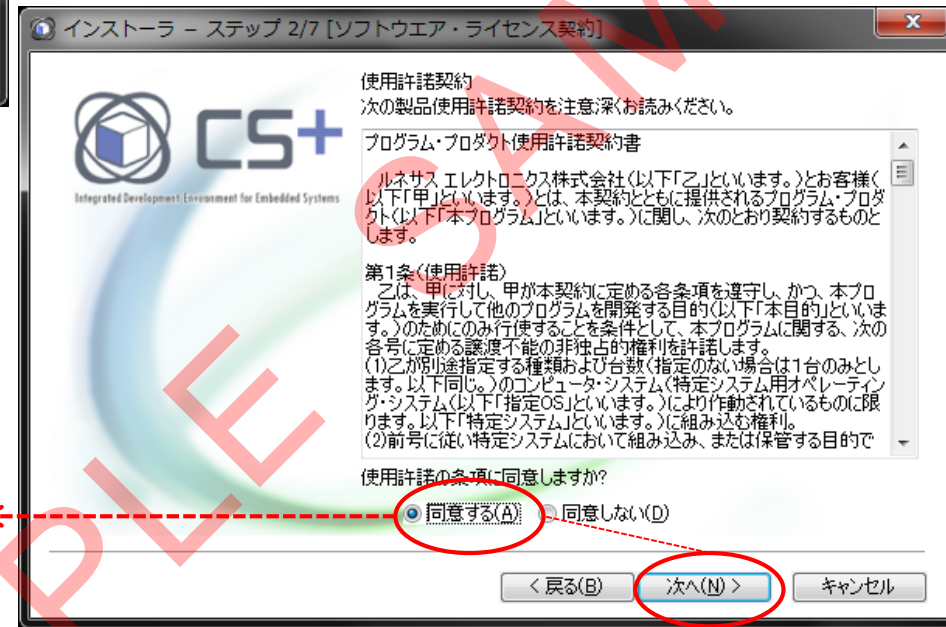


- ウィンドウの手順に従い進め、事前準備のセットアップ完了後にCS+のインストールに移る

CS+のインストール

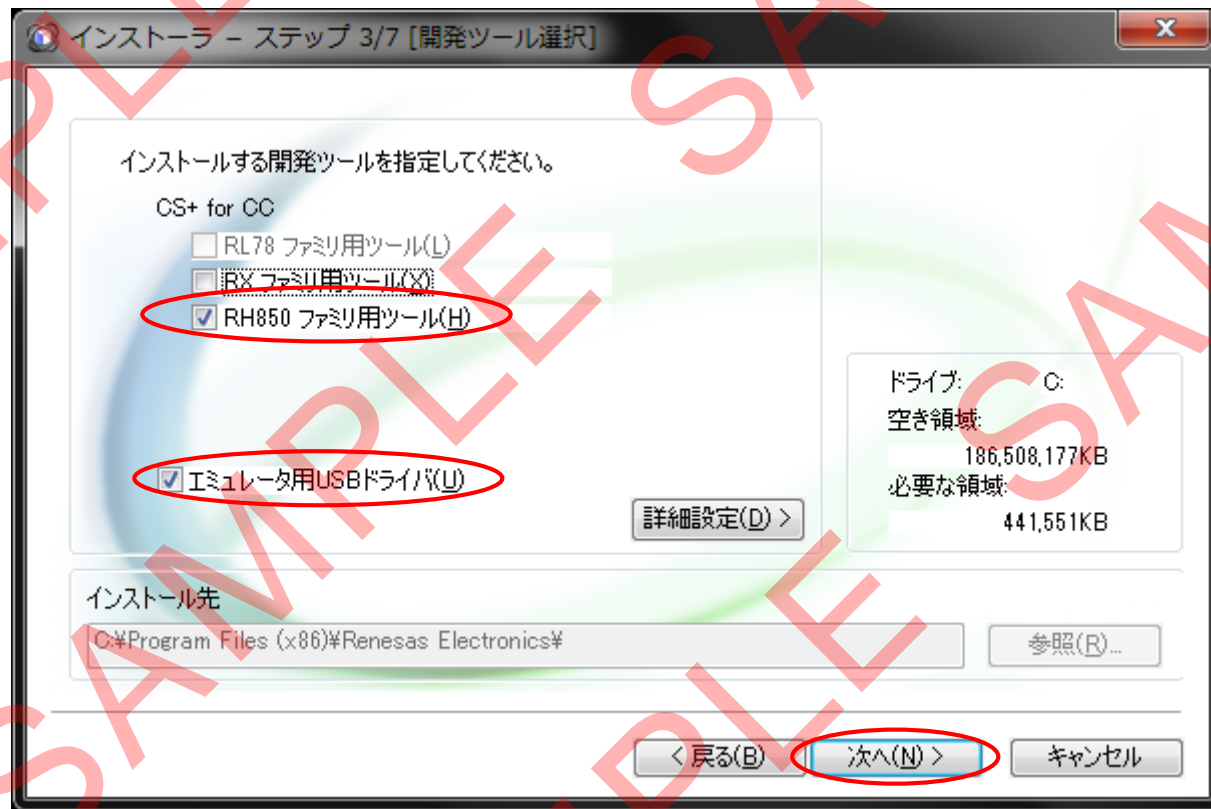


必要な環境がすべてインストール済みの場合、インストーラ起動後に左記のウィンドウから表示される

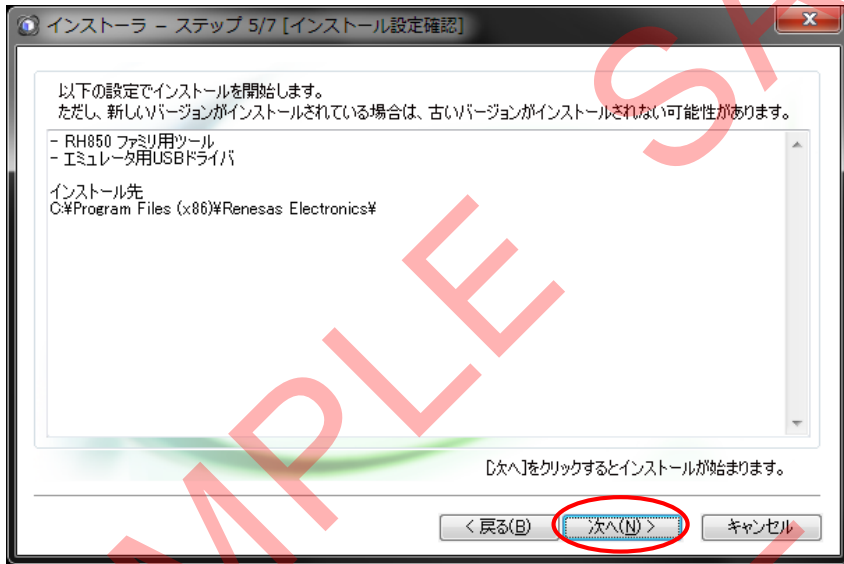


CS+のインストール

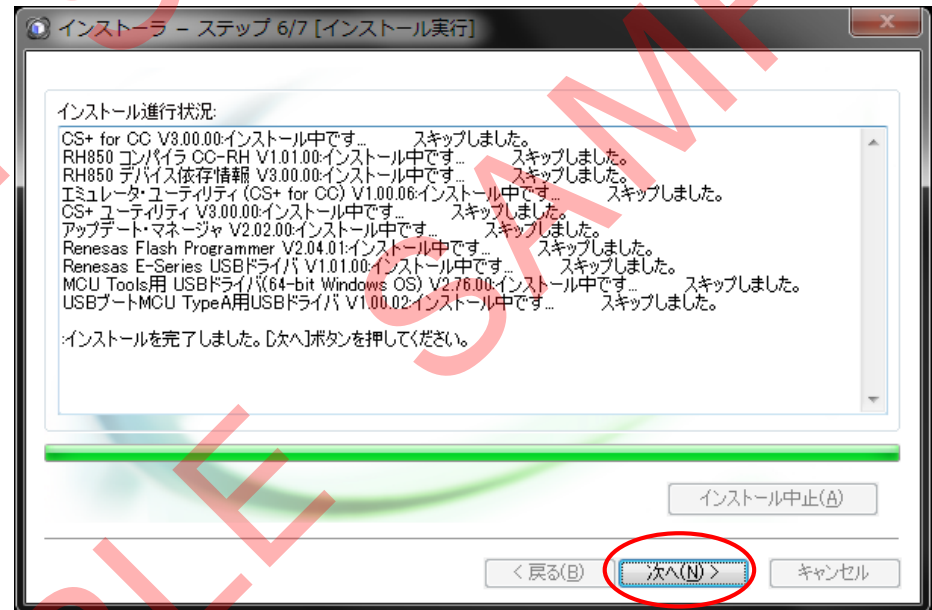
- 本演習ではV850を使用するため“RH850 ファミリ用ツール一式”と“エミュレータ用USBドライバ”にチェックを入れて“次へ”を押下する
 - 開発ツールを細かく指定する場合は[詳細設定]で設定が可能となる
- インストール先を指定する場合は“参照”を押下する



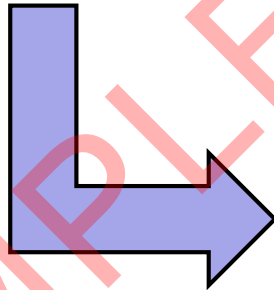
CS+のインストール



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

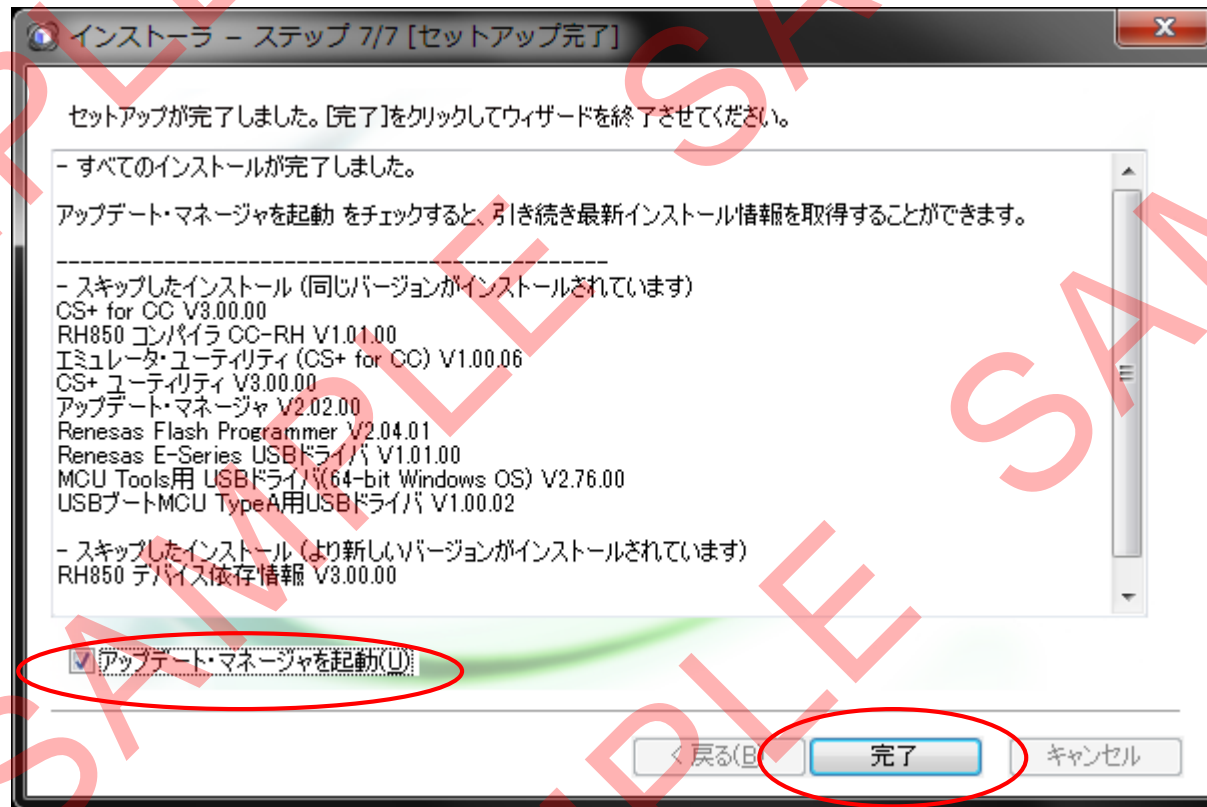


インストール完了後は"次へ"を
押下する



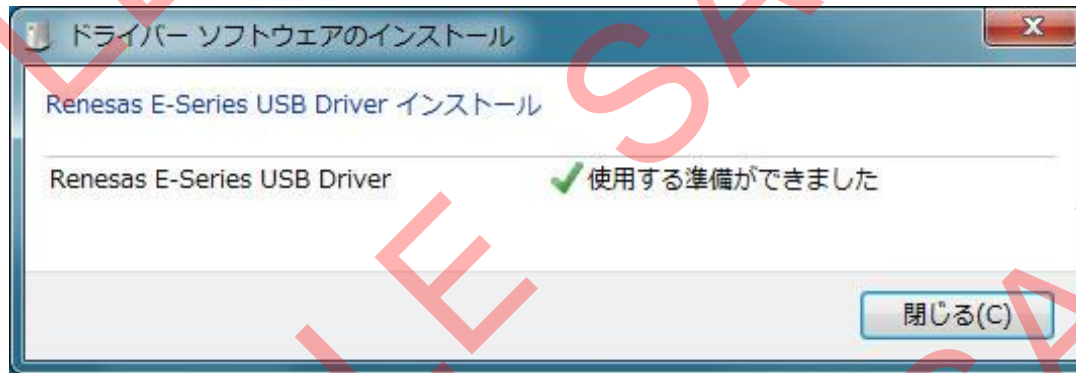
CS+のインストール

- 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。



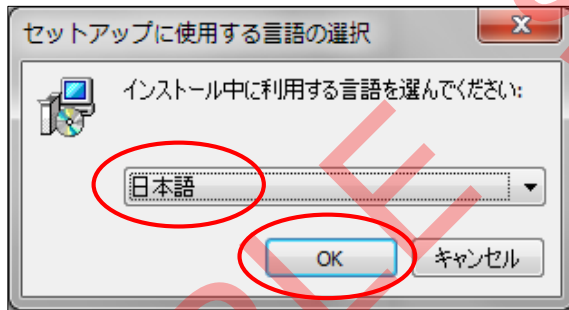
Rubyのインストール(1/3)

- yaml形式のDescriptionファイルからarxmlファイルを生成する際にRubyが必要となる
- Ruby Installer(<http://rubyinstaller.org/>)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



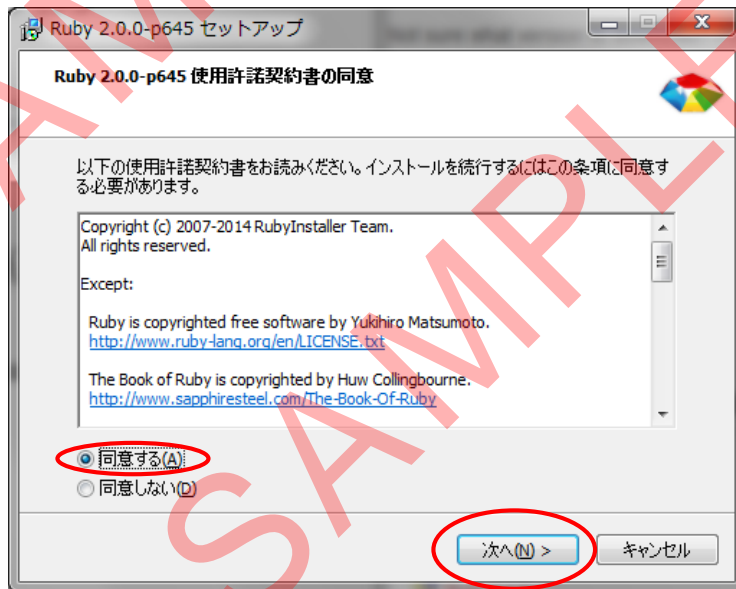
Rubyのインストール(2/3)

- 言語を選択



"日本語"を選択し, "OK"を押下する

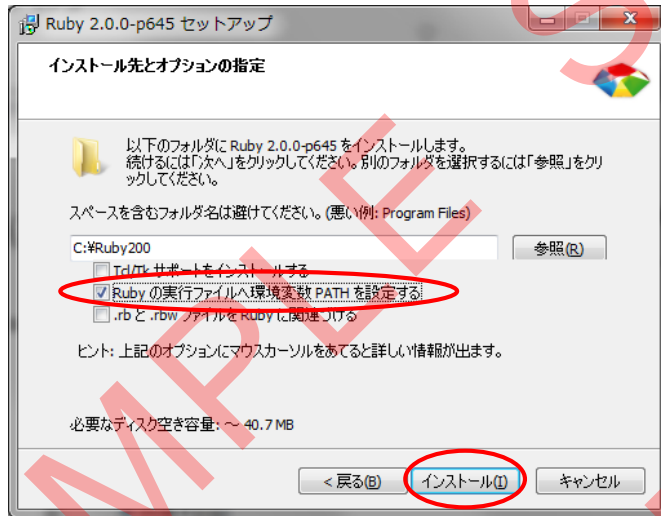
- 仕様許諾契約書に同意



"同意する"にチェックを入れて"次へ"を押下する

Rubyインストール(3/3)

- インストール先を選択



インストール先にパスを通すように指定する

インストールボタンを押すとインストールが始まる

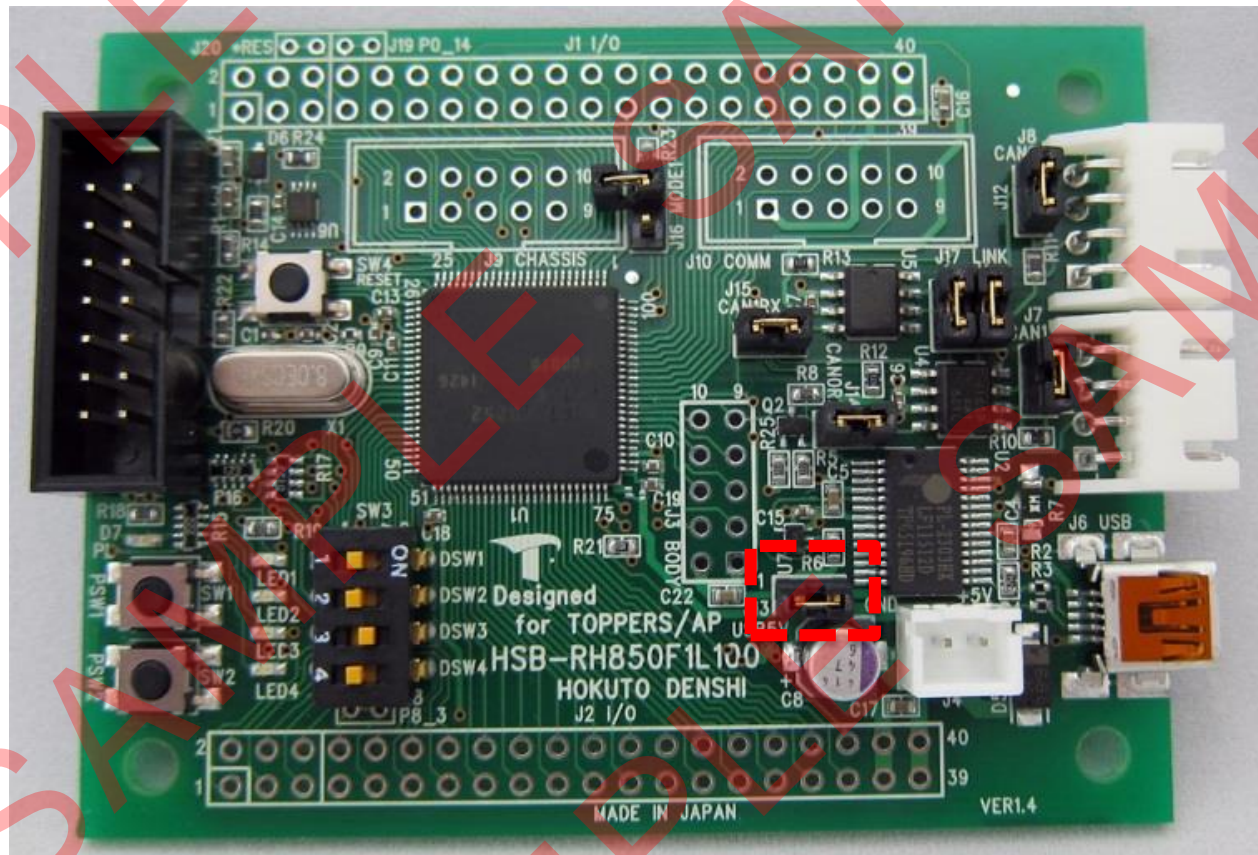
- 終了



完了ボタンを押すとインストール完了となる

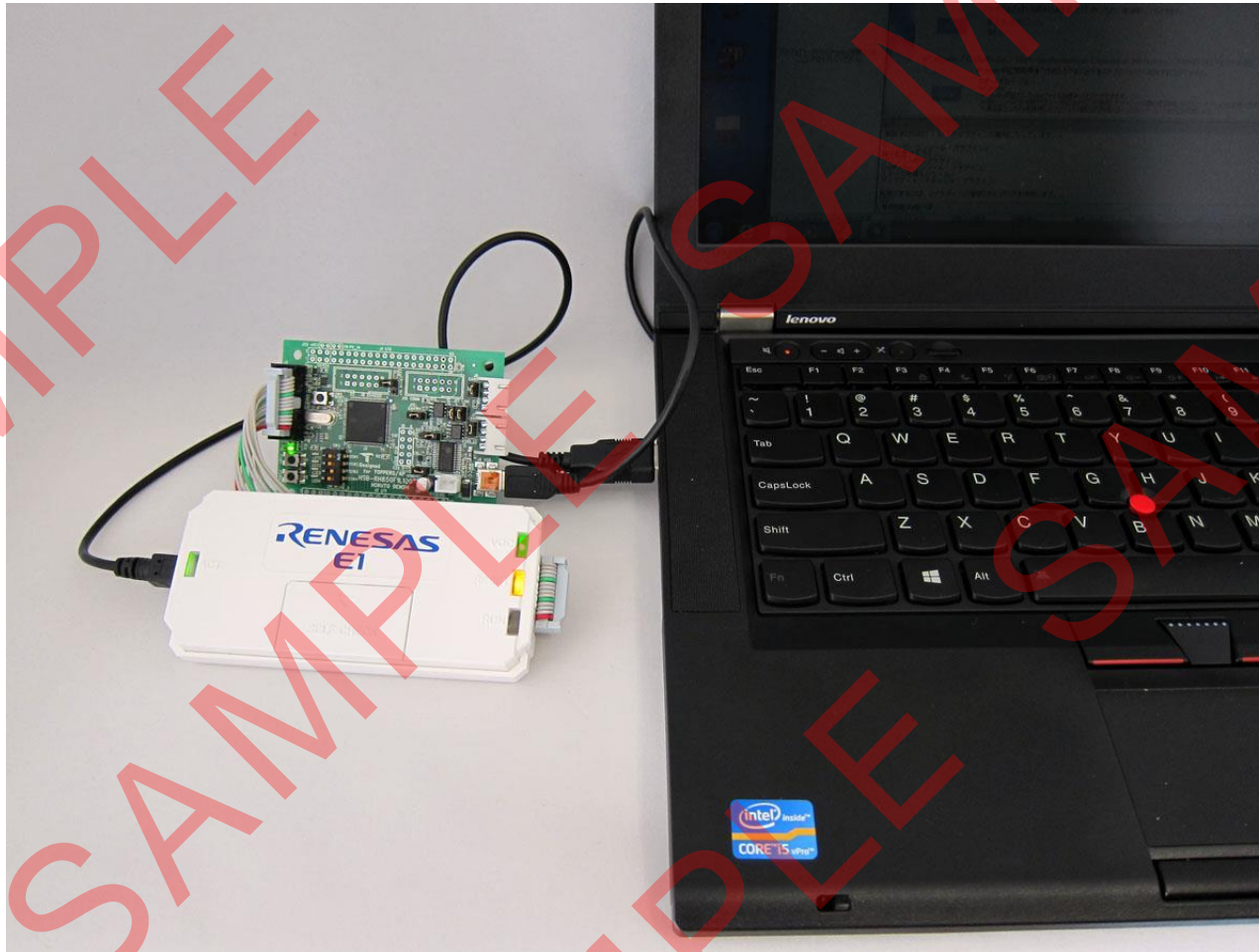
マイコンボードのセットアップ

- DIPスイッチの設定
 - 出荷時の設定から以下を変更する
 - J13 : ショート



マイコンボードのセットアップ

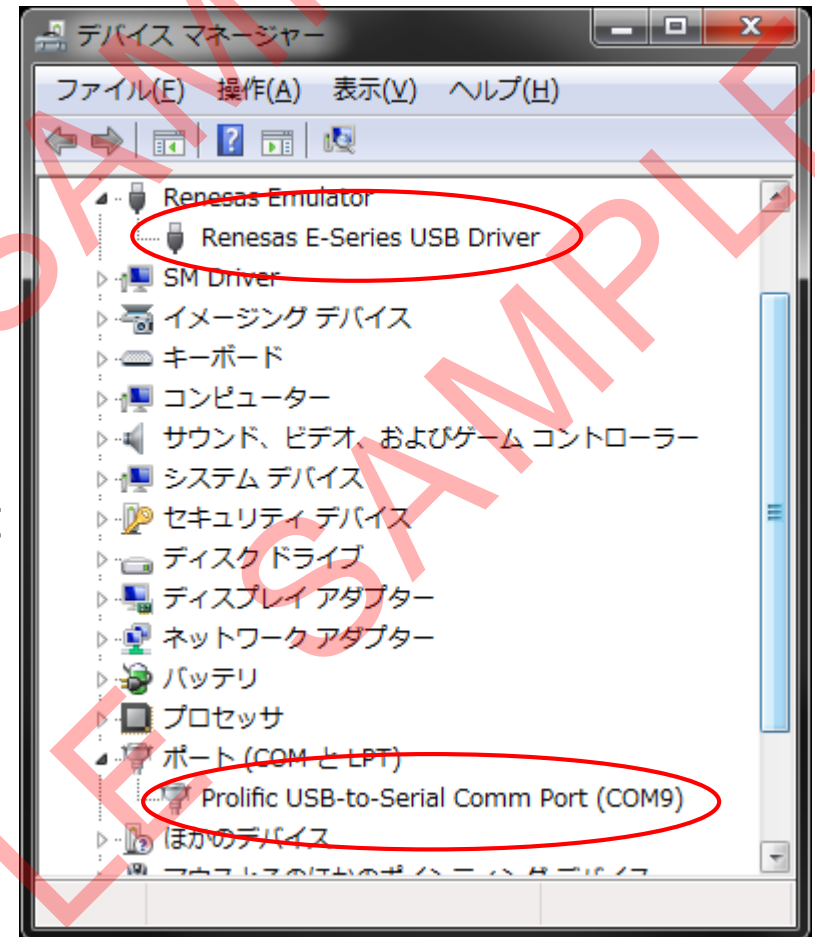
- マイコンボードとE1をJTAGケーブルで接続する
- マイコンボードとE1をPCとUSBケーブルにより接続する



マイコンボードのセットアップ

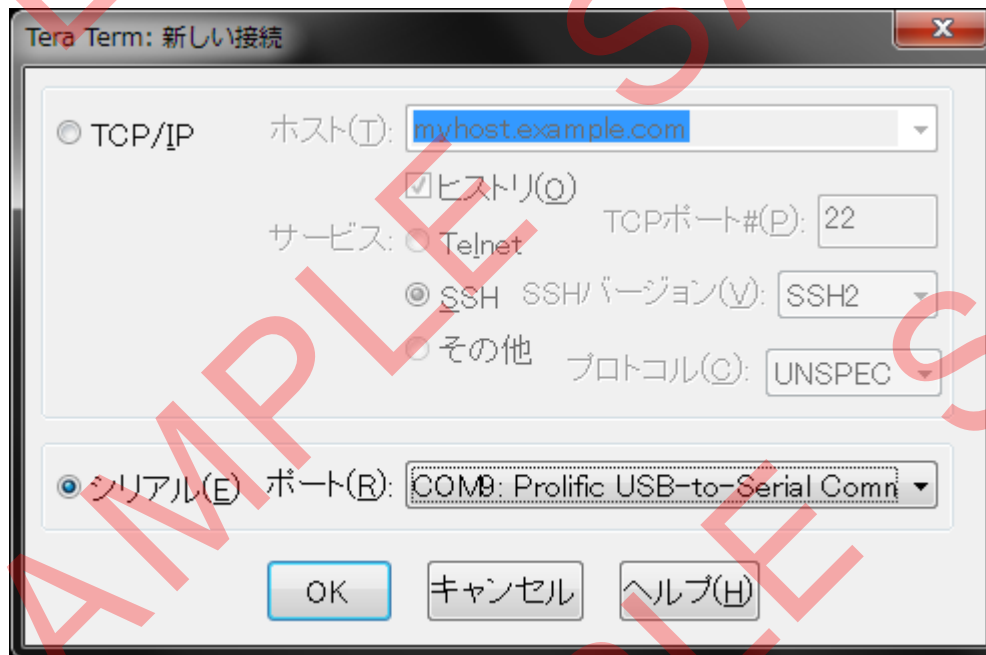
- CPUボード接続時にprolific usb-to-serial comm portと認識される
- E1接続時にデバイスマネージャにて“Renesas E-Series USB Driver”と認識されていればOK

- CPU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



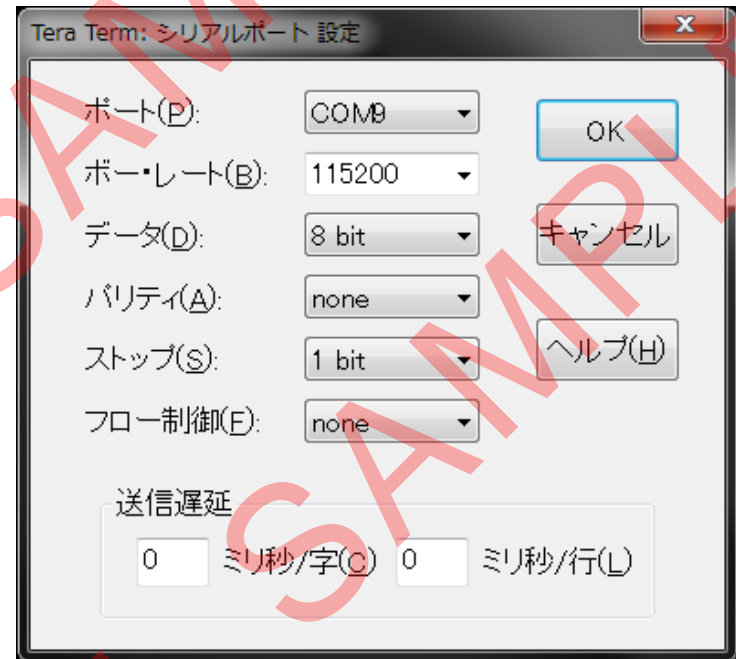
CS+の実行方法：TeraTermの実行

- 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



CS+の実行方法：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



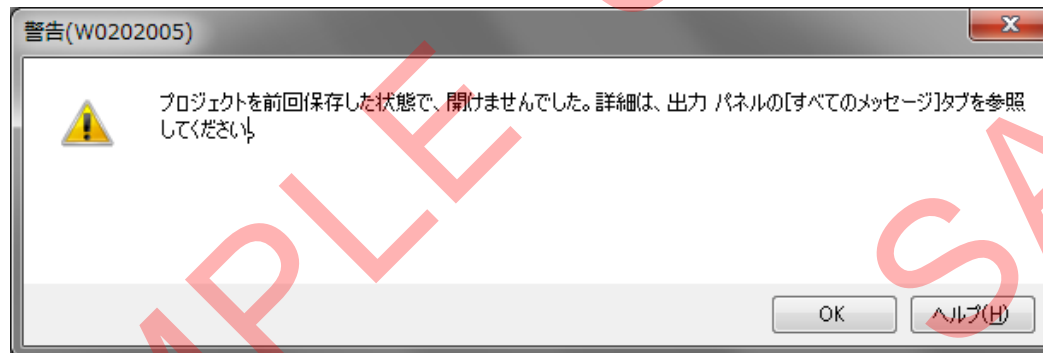
- 設定後メニューの“設定” → “設定の保存”で次回から設定の省略が可能

演習プログラムの準備

- 赤文字のファイルは, TOPPERSのコントリビューティッドソフトウェアであるため, 以下のサイトからダウンロードする(画面下のDownload in other formats: Zip Archive をクリックすることによりZIPファイルがダウンロード可能)
 - http://dev.toppers.jp/trac_user/contrib/browser/autosar_os_training/trunk
 - README.txtに書かれているバージョンのATK2-SC1をダウンロードして同じフォルダに展開する.
-
- **README.txt** : ユーザードキュメント
 - **program/** : 演習プログラム
 - **driver/** : LEDとスイッチのドライバプログラム
 - **atk2-sc1/** : ATK2-SC1ソースコード

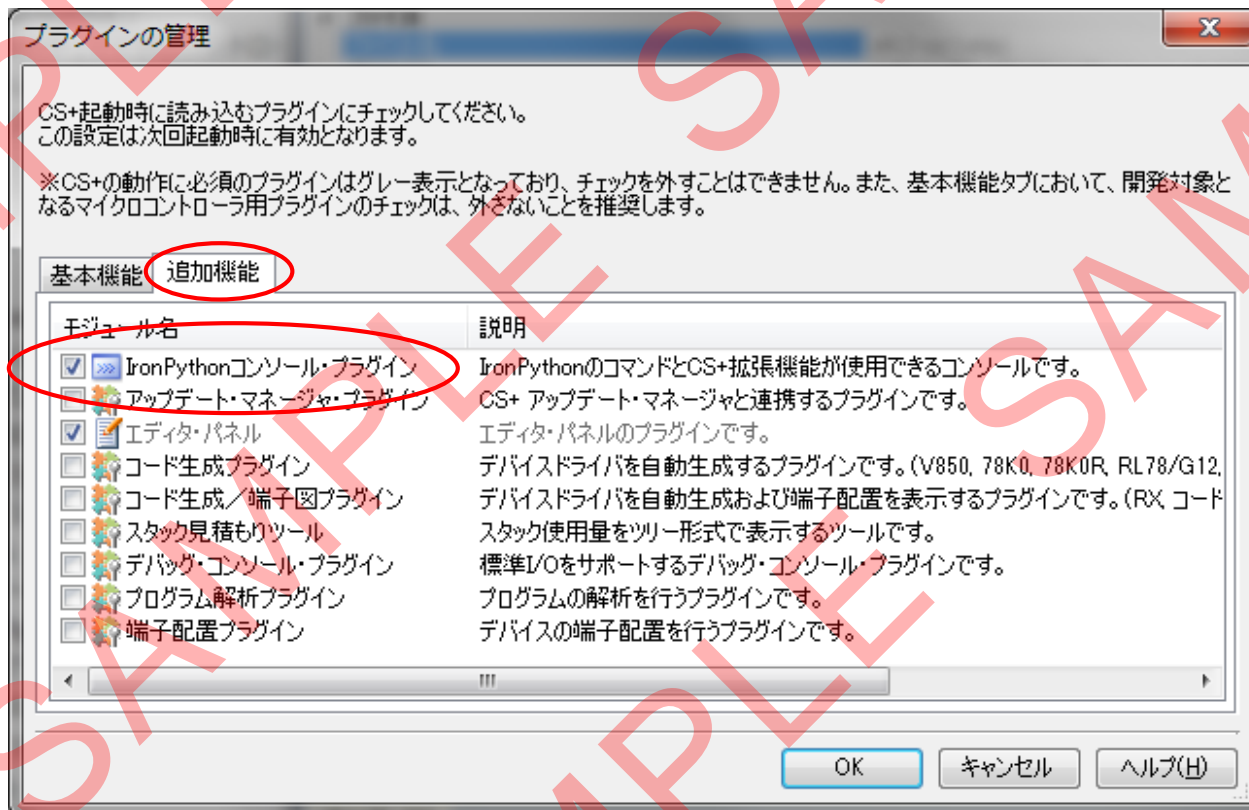
CS+の実行方法：サンプルプログラム実行

- TOPPERS/ATK2-SC1のサンプルをCS+で動作させる
 - program/sample/atk2-sc1.mtpjをCS+で開く
- 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りますが、動作に影響はないため"OK"を押下する



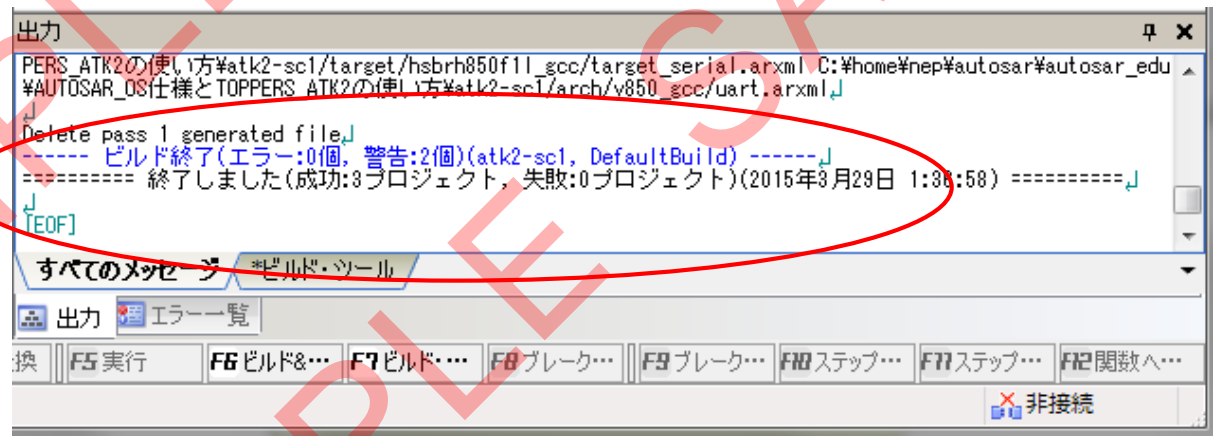
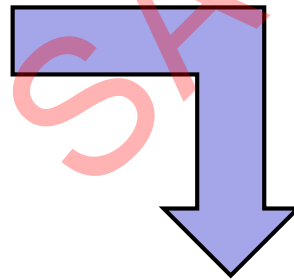
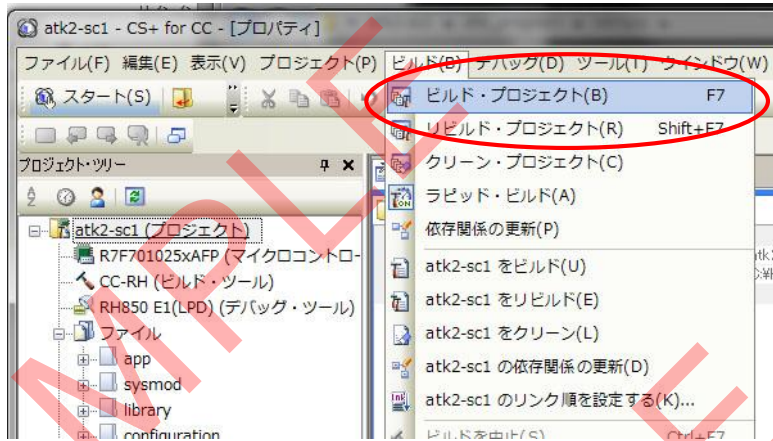
CS+の実行方法：サンプルプログラム実行

- 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。



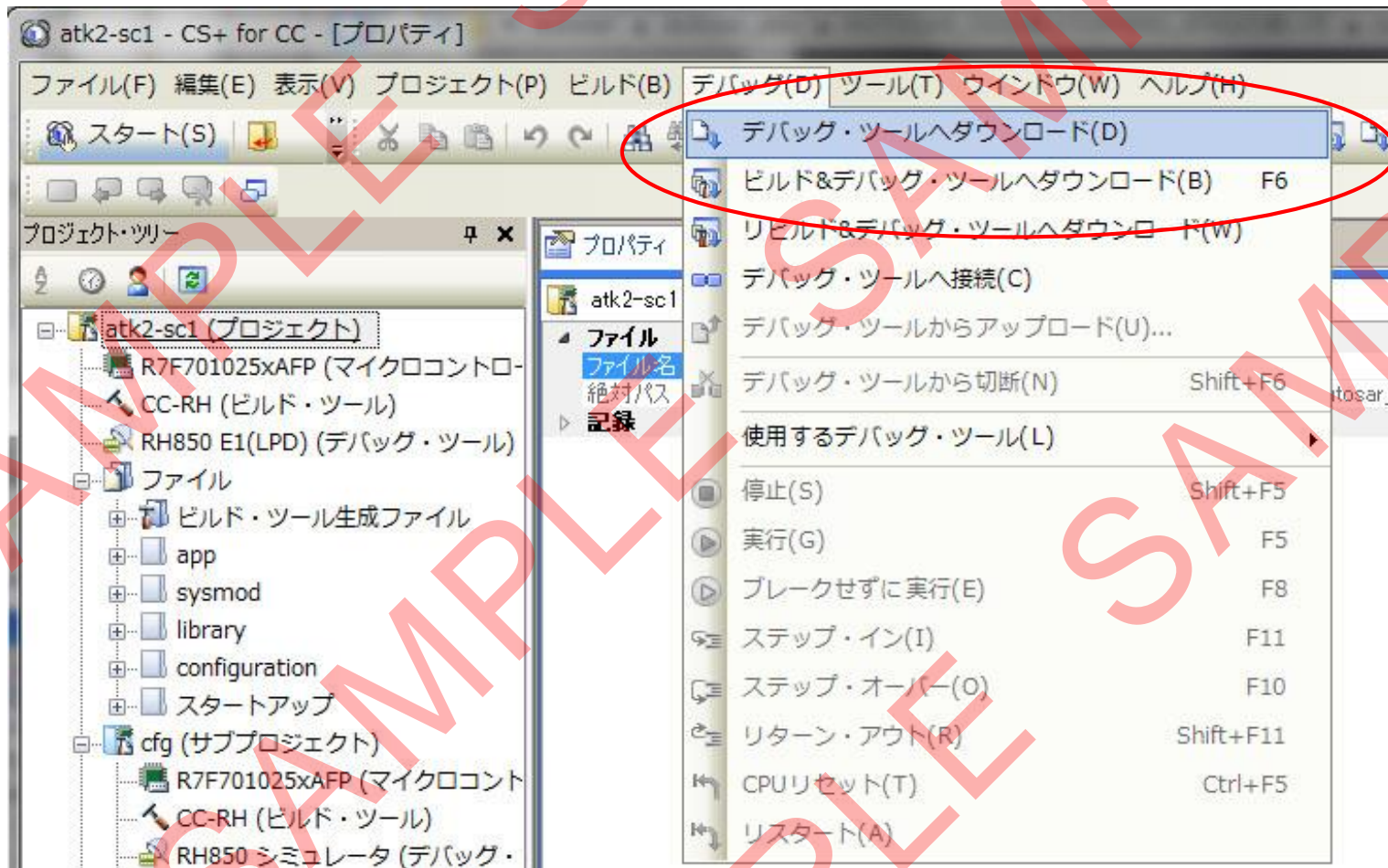
CS+の実行方法：サンプルプログラム実行

- ビルドから“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




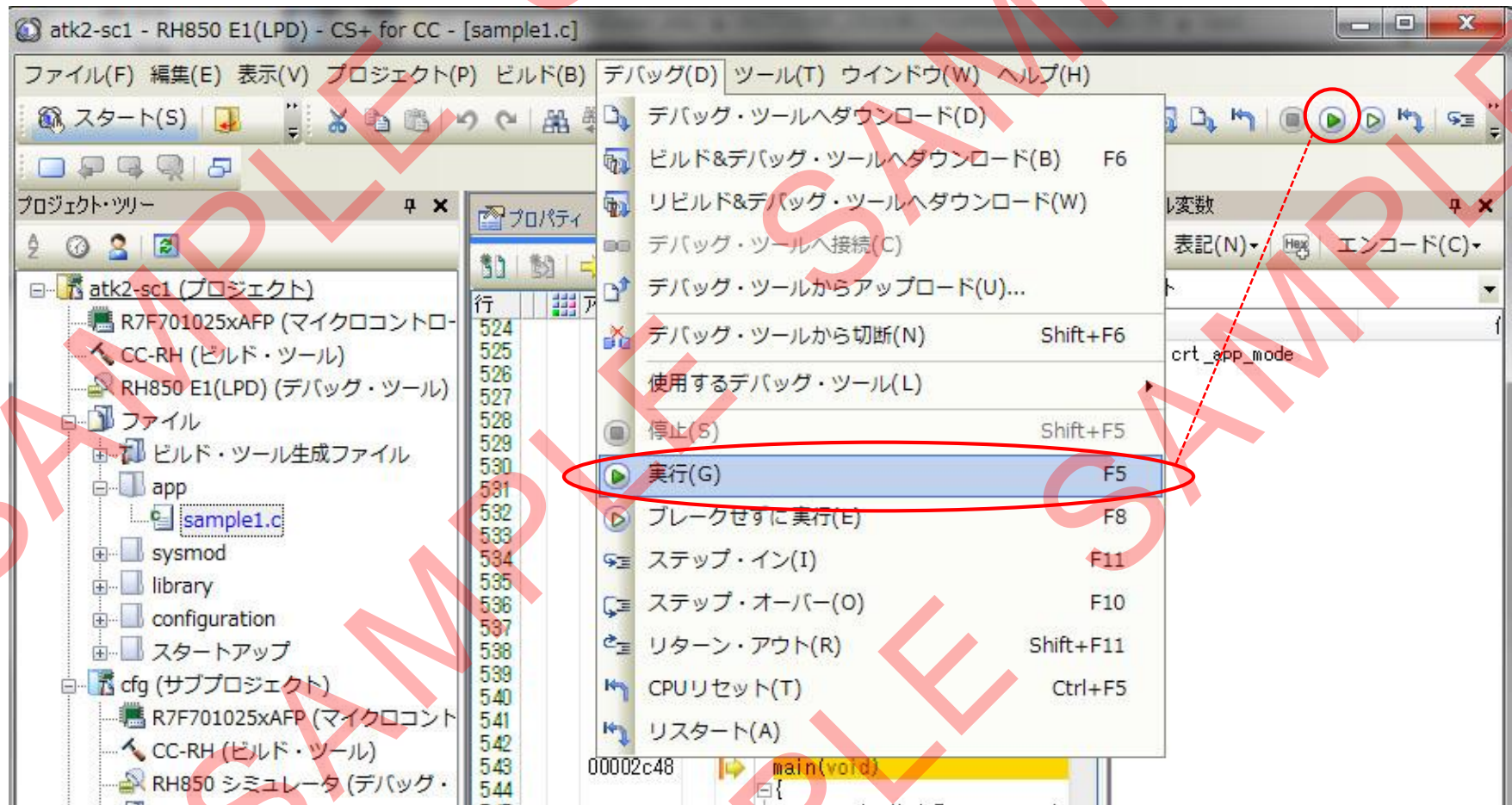
CS+の実行方法：サンプルプログラム実行

- デバッグから“デバッグ・ツールヘダウンロード”を選択する



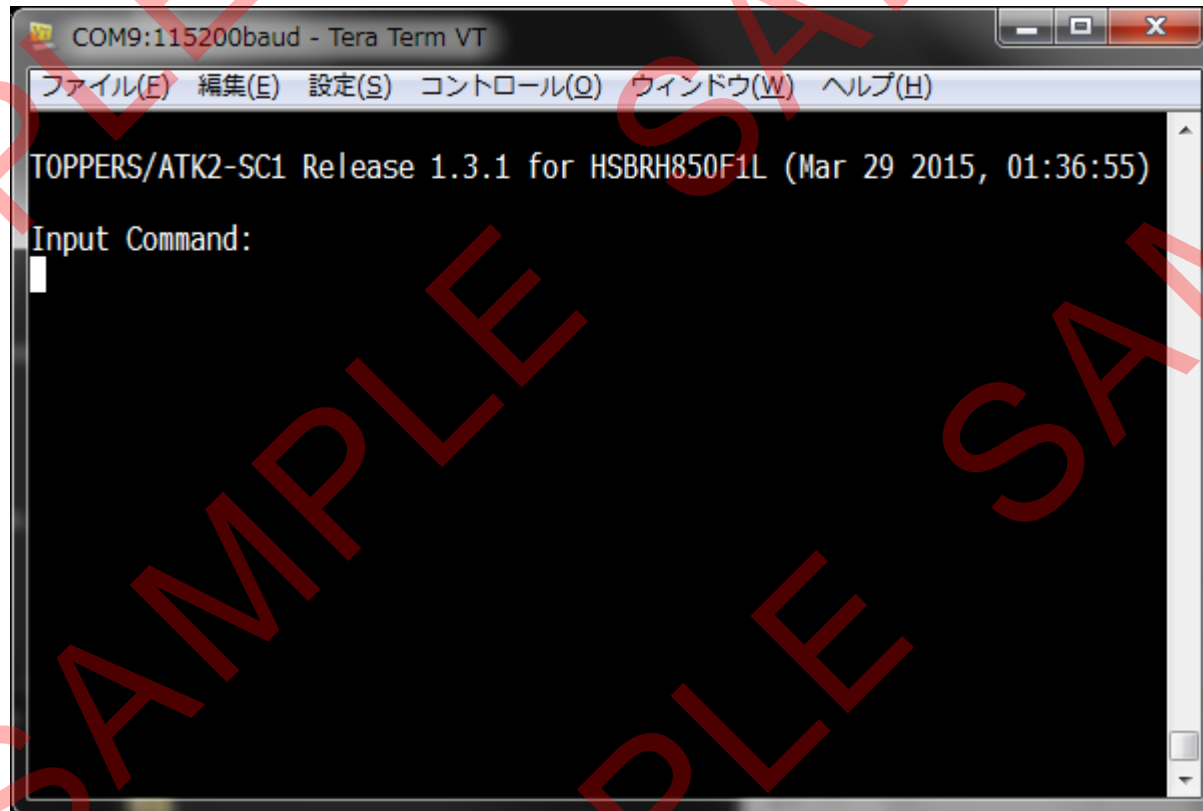
CS+の実行方法：サンプルプログラム実行

- デバッグから“実行”もしくは  を押下することでサンプルプログラムの実行が可能となる



CS+の実行方法：サンプルプログラム実行

- プログラムが実行時はCS+のウィンドウ下部には、
 **RUN**  **実行中** が表示され、TeraTermには出力ログが表示される



ATK2サンプルプロジェクトの新規作成方法

- program/sampleと同等のサンプルプロジェクトを新規で作成する場合の手順を紹介する
 - program内に任意の名称のフォルダを用意する
 - 容易したにフォルダ”にatk2-sc1/arch/ccrh/configure内にある下記のファイルをコピーする
 - configure.mtpj
 - configure.py
 - def.py
 - コピーしたconfigureのdef.py記載されている項目を編集する
 - SRCDIR : ソースコードのトップ(atk2-sc1/)との相対位置
 - TARGET = "hsbrh850f1l_ccrh"
 - TARGET_MCU = 'R7F701025xAFP'
 - configure.mtpjをCS+で開く
 - atk2-sc1.mtpjが作成され開かれる

ATK2サンプルプロジェクトの新規作成方法

- 用意したフォルダにsample1.c/sample1.h等のファイルが生成できていればプロジェクトの作成は成功となる
 - 生成できてい場合はSRCDIRの設定が誤っている場合がある
- 以後、プロジェクトファイルはatk2-sc1.mtpjを使用するためコピーしてきたconfigure.mtpj/configure.pyは削除可能となる
- def.pyで編集可能なパラメータについて
 - SRCDIR : ソースコードのトップとの相対位置
 - CFGNAME : コンフィギュレーションファイル名の指定
 - TARGET : ターゲットの指定(./target 以下いずれかのフォルダ名)
 - TARGET_MCU : ターゲットMCU名の指定
 - app_app_files : アプリケーションファイル名の指定
 - USER_INCLUDE : インクルードパスの指定
 - COPY_SAMPLE1 : ./sampleからsample1ファイルをコピーするか指定

TOPPERS/ATK2プログラミング演習

- 開発・実行環境の使用方法
- **タスク管理機能**
- 割込み管理機能
- 同期機能(イベント, リソース)
- アラーム, カウンタ
- その他

タスク管理機能：実習内容

- タスクの生成方法, スケジューリング規則及び, システムの起動方法について学ぶ
 - タスクの生成/システムの起動方法
 - 1タスクによるLED点滅プログラムの解説
 - デバイスドライバ/コンソール出力
 - LED/プッシュスイッチ/ディップスイッチを扱うプログラムの解説
 - マルチタスク(優先度)
 - フルプリエンパティブタスクから高優先度タスクを起動するプログラムを作成
 - マルチタスク(ノンプリエンパティブ, 起動要求のキューイング)
 - ノンプリエンパティブタスクから高優先度タスクを起動するプログラムを作成
 - マルチタスク(ChainTask)
 - ChainTaskによるタスク起動プログラムの作成

タスク管理機能：プログラムファイル

- 演習プログラムファイルの置き場所
 - program

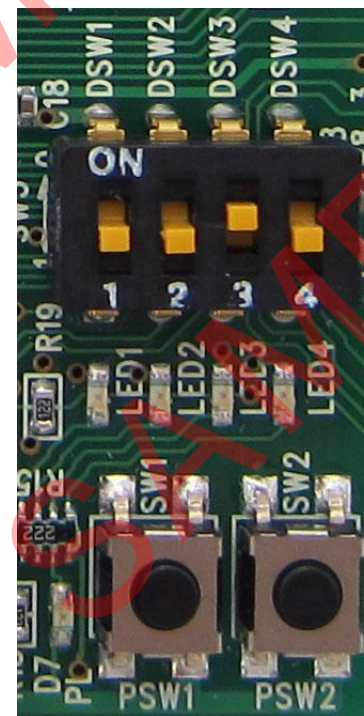
フォルダ名	プログラム内容
led_1task	1タスクによるLED点滅プログラム
device_log_1task	LED/プッシュスイッチ/ディップスイッチ操作プログラム
multi_task_active	フルプリエンパティブルタスクから高優先度タスクを起動するプログラム
multi_task_nonpre	ノンプリエンパティブタスクから高優先度タスクを起動するプログラム
multi_actqueu	タスク起動のキューイング確認プログラム
multi_task_chain	ChainTaskによるタスク起動プログラム

タスクの生成/システムの起動方法

- タスク(led_1task)によりLED1を一定周期で点滅させる
- 学習内容
 - プログラムの全体像の理解
 - システム起動方法
 - タスクの記述方法
 - LEDドライバの使用方法
 - yamlファイル
- 構成ファイル
 - toppers_atk2.c : Cソースファイル
 - toppers_atk2.h : ヘッダファイル
 - toppers_atk2.yaml : yamlファイル
 - abrex.bat : abrex実行用バッチファイル
 - clean.bat : クリーン用バッチファイル
 - toppers_atk2.arxml : コンフィギュレーションファイル
 - atk2-sc1.mtpj : CS+のプロジェクトファイル

デバイス操作：DSWスイッチ, PSWスイッチ

- 演習プログラムにて使用するディップスイッチ(DSW), プッシュスイッチ(PSW)は拡張I/Oボードの下記を操作する
- ディップスイッチ(DSW1~4)
 - 0 : OFFの状態
 - 1 : ONの状態
- プッシュスイッチ(PSW1, PSW2)



デバイス操作関数群：LED

- 拡張I/OボードのLEDとスイッチ用のドライバ関数

- driver/device.c
- driver/device.h

- LED関連

- マクロ

```
#define LED1      0x01
#define LED2      0x02
#define LED3      0x04
#define LED4      0x08
#define LED_OFF   0x00
```

- 関数

- void led_init(void)
 - 初期化(出力ポートに設定し, LED消灯)
- void led_out(uint8 led_data)
 - 引数にマクロで点灯パターンを指定

デバイス操作関数群：ディップスイッチ(DSW)

- マクロ

```
#define DSW1      0x01
#define DSW2      0x02
#define DSW3      0x04
#define DSW4      0x08
```

- 関数

- void switch_dip_init(void)
 - 初期化(入力モードに設定)
- uint8 switch_dip_sense(void)
 - DSWの状態取得
 - 戻り値にON状態スイッチのビットを'1'とした値を返す

デバイス操作関数群：プッシュスイッチ(PSW)

- マクロ

```
#define PSW1      0x1  
#define PSW2      0x2
```

- 関数

- void switch_push_init(void)
 - 初期化(入力モードに設定)
- uint8 switch_push_sense(void)
 - 戻り値にON状態スイッチのビットを'1'とした値を返す

デバイスドライバ

- I/O関連の機能()のデバイス関連のAPIを使用する
 - LOCAL_INLINE uint16 sil_reh_mem(void *mem)
 - 16ビット単位の読出し
 - LOCAL_INLINE void sil_wrh_mem(void *mem, uint16 data)
 - 16ビット単位の書込み

```
/* LED接続ポート書き込み */  
void  
led_out(unsigned char led_data) {  
    uint16 wk;  
  
    led_data &= 0x0f;  
    wk = sil_reh_mem((void *) P(4));  
    wk &= ~0x0f;  
    wk |= led_data;  
    sil_wrh_mem((void *) P(4), wk);  
}
```

コンフィギュレーション

- コンフィギュレーション時、静的に設定するパラメータの集合体
コンテナ名称と0個以上の属性を持つ
また、入れ子形式でコンテナ内にコンテナを含むことができ、
これをサブコンテナという
- 次頁以降から“コンテナ”、“サブコンテナ”をOSオブジェクトの
集合体を示す
 - 例：OsOS, OsTask, OsAlarmAutostart 等
- 次頁以降から“パラメータ”をコンフィギュレーションパラメータの
設定を行うものとして示す
 - 例：OsOsStackSize, OsAlarmSetEventRef 等

コンテナ : Os

- 以下のサブコンテナを設定することで、使用する機能の設定が出来る

- OsOS
- OsAppMode
- OsTask
- OsIsr
- OsEvent
- OsResource
- OsCounter
- OsAlarm

Ecuc:

Os:

OsInclude:

OsIncludeFileName : toppers_atk2.h

AppMode1:

DefinitionRef : OsAppMode

OsOS:

OsStackMonitoring : true

OsUseGetServiceId : true

...

<toppers_atk2.yaml >

コンテナ

サブコンテナ

- 本コンテナが記述されていない場合、ジェネレータはエラーを検出する