

熊本県産業技術センター 第7回技術普及講習会(組み込み技術) 「RTミドルウェアの概要と実習」

日時: 2011年11月25日(金) 11:00~17:00
場所: 熊本県産業技術センター 大会議室



RTミドルウェア講習会



11:00- 11:30	第1部:RTミドルウェアの概略紹介
	担当: 安藤 慶昭(産業技術総合研究所) 概要: RTミドルウェア, RTコンポーネントの概要を説明します。また, Web上で自分の作品を公開できる仕組みについて紹介します。
11:30- 12:00	第2部:RTミドルウェアの概略, 導入方法の紹介
	担当: 栗原真二(産業技術総合研究所) 概要: サンプルシステムを用いた概略紹介. RTミドルウェアの導入方法について紹介します。
13:00- 13:45	第3部: RTミドルウェアを用いたシステム構築方法の紹介
	担当: 坂本武志(株式会社グローバルアシスト) 概要: RTコンポーネントを組み合わせてシステムを構築する方法について説明します。
14:00- 14:45	第4部:RTコンポーネントの作り方
	担当: 坂本武志(株式会社グローバルアシスト) 概要: RTコンポーネントのテンプレート作成ツールRTCBuilderを用いたコンポーネントの設計と実装について説明します。
15:00- 17:00	第5部:RTコンポーネント作成実習
	担当: 栗原真二(産業技術総合研究所) 概要: 参加者全員で実際にコンポーネントを作成してシステムを構築してみます。

第4部 RTコンポーネントの作り方

株式会社 グローバルアシスト
坂本 武志



RTCBuilderについて

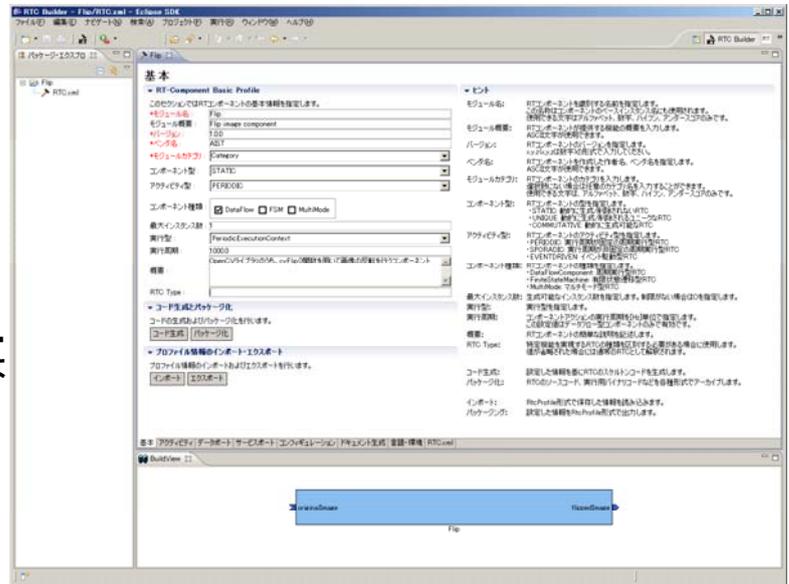


■ RTCBuilderとは？

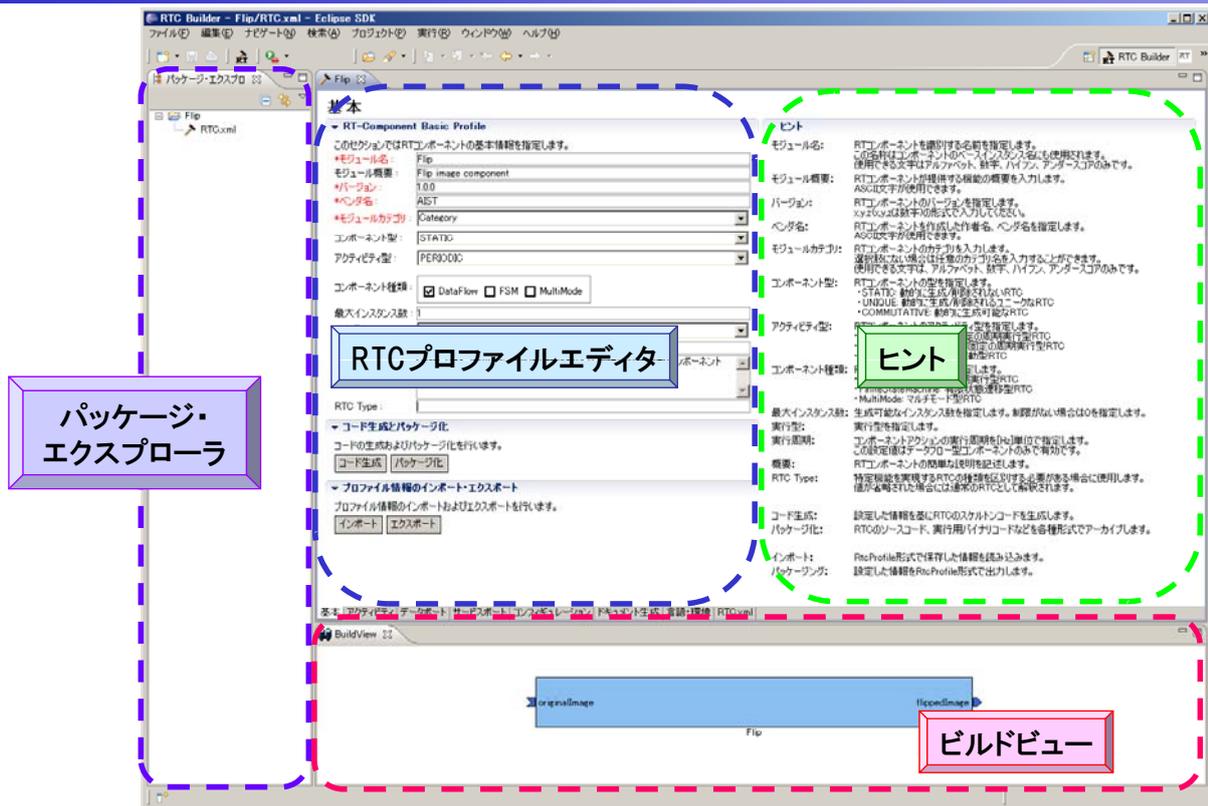
- コンポーネントのプロファイル情報を入力し、ソースコード等の雛形を生成するツール
- 開発言語用プラグインを追加することにより、各言語向けRTCの雛形を生成することが可能

- C++
- Java
- Python

※C++用コード生成機能は RtcBuilder本体に含まれています。
 ※その他の言語用コード生成機能は追加プラグインとして提供されています



画面構成



- 以下から「RTC.xml」をダウンロードします。

<http://www.openrtm.org/openrtm/ja/node/4597#toc5>



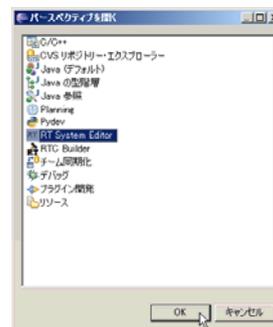
準備

- パースペクティブの切り替え

① 画面右上の「パースペクティブを開く」を選択し、一覧から「その他」を選択



② 一覧画面から対象ツールを選択



※パースペクティブ
Eclipse上でツールの構成を管理する単位
メニュー、ツールバー、エディタ、ビューなど
使用目的に応じて組み合わせる
独自の構成を登録することも可能

プロジェクト作成/エディタ起動

① ツールバー内のアイコンをクリック

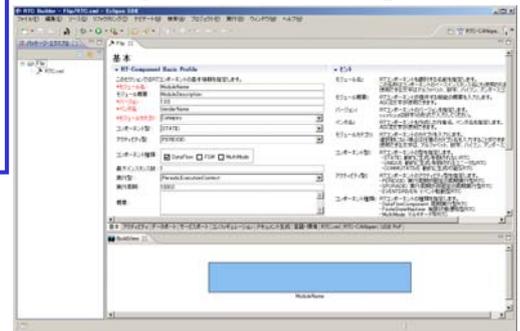
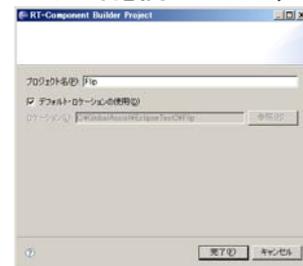


- ※メニューから「ファイル」-「新規」-「プロジェクト」を選択
【新規プロジェクト】画面にて「その他」-「RtcBuilder」を選択し、「次へ」
- ※メニューから「ファイル」-「Open New Builder Editor」を選択

- ※任意の場所にプロジェクトを作成したい場合
②にて「デフォルト・ロケーションの使用」チェックボックスを外す
「参照」ボタンにて対象ディレクトリを選択
→物理的にはワークスペース以外の場所に作成される
論理的にはワークスペース配下に紐付けされる

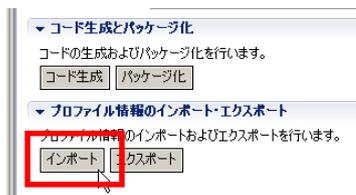
プロジェクト名: Flip

② 「プロジェクト名」欄に入力し, 「終了」

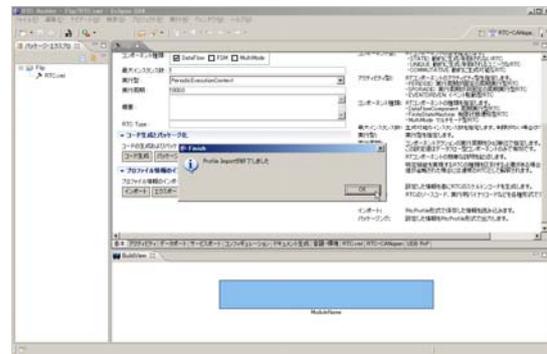


プロファイル インポート

① 「基本」タブ下部の「インポート」ボタンをクリック

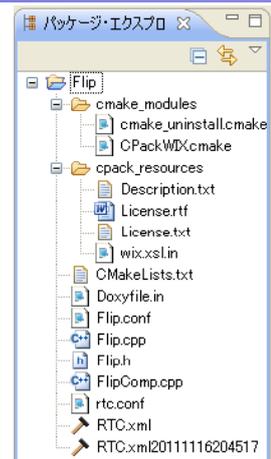
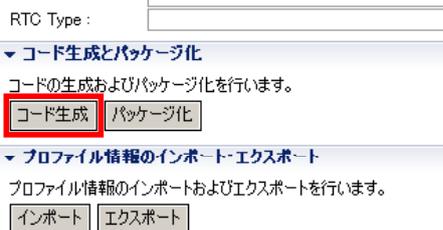


② 【インポート】画面にて対象ファイルを選択



- 作成済みのRTコンポーネント情報を再利用
 - 「エクスポート」機能を利用して出力したファイルの読み込みが可能
 - コード生成時に作成されるRtcProfileの情報を読み込み可能
 - XML形式, YAML形式での入出力が可能

■ コード生成



■ コード生成実行後、パースペクティブを自動切替



※生成コードが表示されない場合には、「リフレッシュ」を実行

C++版RTC → CDT
 Java版RTC → JDT
 (デフォルトインストール済み)
 Python版 → PyDev

画面要素名	説明
基本プロファイル	RTコンポーネントのプロファイル情報など、コンポーネントの基本情報を設定。 コード生成、インポート/エクスポート、パッケージング処理を実行
アクティビティ・プロファイル	RTコンポーネントがサポートしているアクティビティ情報を設定
データポート・プロファイル	RTコンポーネントに付属するデータポートに関する情報を設定
サービスポート・プロファイル	RTコンポーネントに付属するサービスポートおよび各サービスポートに付属するサービスインターフェースに関する情報を設定
コンフィギュレーション	RTコンポーネントに設定するユーザ定義のコンフィギュレーション・パラメータセット情報およびシステムのコンフィギュレーション情報を設定
ドキュメント生成	生成したコードに追加する各種ドキュメント情報を設定
言語・環境	生成対象コードの選択やOSなどの実行環境に関する情報を設定
RTC.xml	設定した情報を基に生成したRTC仕様(RtcProfile)を表示

RTコンポーネントの名称など、基本的な情報を設定

基本

このセクションではRTコンポーネントの基本情報を指定します。

*モジュール名: Flip
*モジュール概要: Flip image component
*バージョン: 1.0.0
*ベンダ名: AIST
*モジュールカテゴリ: Category

コンポーネント型: STATIC
アクティビティ型: PERIODIC

コンポーネント種類: DataFlow FSM MultiMode

最大インスタンス数: 1
実行型: PeriodicExecutionContext
実行周期: 0.0
概要: OpenCVライブラリのうち、cvFlip関数を用いて画像の反転を行うコンポーネント

RTC Type:

コード生成とパッケージ化

コードの生成およびパッケージ化を行います。

プロフィール情報のインポート・エクスポート

プロフィール情報のインポートおよびエクスポートを行います。

イベント

モジュール名: RTコンポーネントを識別する名前を指定します。この名称はコンポーネントのベースインスタンス名にも使用されます。使用できる文字はアルファベット、数字、ハイフン、アンダースコアのみです。
モジュール概要: RTコンポーネントが提供する機能の概要を入力します。ASCII文字が使用できます。
バージョン: RTコンポーネントのバージョンを指定します。

モジュール名: Flip
モジュール概要: 任意(Flip image component)
バージョン: 1.0.0
ベンダ名: 任意(AIST)
モジュールカテゴリ: 任意(Category)
コンポーネント型: STATIC
アクティビティ型: PERIODIC
コンポーネントの種類: DataFlow
最大インスタンス数: 1
実行型: PeriodicExecutionContext
実行周期: 1000.0

- ※エディタ内の項目名が赤字の要素は必須入力項目
- ※画面右側は各入力項目に関する説明

アクティビティ・プロフィール

生成対象RTCで実装予定のアクティビティを設定

アクティビティ

このセクションでは使用するアクションコールバックを指定します。

コンポーネントの初期化と終了処理に関するアクション
onInitialize onFinalize
実行エンタキストの起動と停止に関するアクション
onStartup onShutdown
always状態でのコンポーネントアクション
onActivated onDeactivated onAborting
onError onReset
DataFlow型コンポーネントのアクション
onExecute onStateUpdate onRateChanged
FSM型コンポーネントのアクション
onAction
onModeChanged

Documentation

このセクションでは各アクションの概要を説明するコメントを記述します。上記のアクションを参照すると、それぞれのコメントを記述できます。

アクティビティ名: onInitialize ON OFF

動作概要: コンポーネント自身の各種初期化処理
事前条件: なし
事後条件: コンポーネントの初期化処理が正常に完了している

イベント

初期化と処理です。コンポーネントライフサイクル開始時に一度だけ呼び出されます。常に有効。終了処理です。コンポーネントライフサイクル終了時に一度だけ呼び出されます。常に有効。ExecutionContextが実行を開始する時に一度だけ呼び出されます。ExecutionContextが実行を停止するときに一度だけ呼び出されます。非アクティブ状態からアクティブ化されると一度だけ呼び出されます。アクティブ状態から非アクティブ化されると一度だけ呼び出されます。ERROR状態に入る前に一度だけ呼び出されます。ERROR状態にいる間複数回呼び出されます。ERROR状態から非アクティブ状態に移行するときに一度だけ呼び出されます。アクティブ状態から非アクティブ化されると一度だけ呼び出されます。ExecutionContextが変更されたときに一度だけ呼び出されます。対応する状態に応じた動作を実行するために一度だけ呼び出されます。モードが変更された時に呼び出されます。

動作概要: アクティビティの概要を説明記述します。
事前条件: アクティビティを実行する前に成立すべき事前条件を記述します。
事後条件: アクティビティを実行した後に成立すべき事後条件を記述します。

①設定対象のアクティビティを選択

onActivated
onError

②使用/未使用を設定

ON OFF

以下をチェック:
onActivated
onDeactivated
onExecute

- ※現在選択中のアクティビティは、一覧画面にて赤字で表示
- ※使用(ON)が選択されているアクティビティは、一覧画面にて背景を水色で表示
- ※各アクティビティには、「動作概要」「事前条件」「事後条件」を記述可能
→記述した各種コメントは、生成コード内にDoxygen形式で追加される

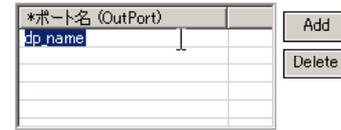
■ 生成対象RTCに付加するDataPortの情報を設定

データポート



① 該当種類の欄の「Add」ボタンをクリックし、ポートを追加後、直接入力で名称設定

（ポート）の情報を設定します。

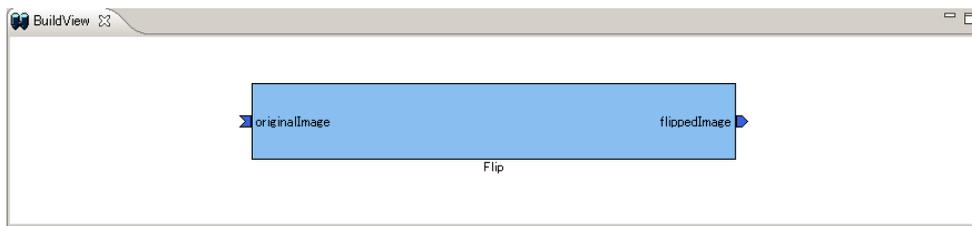


② 設定する型情報を一覧から選択



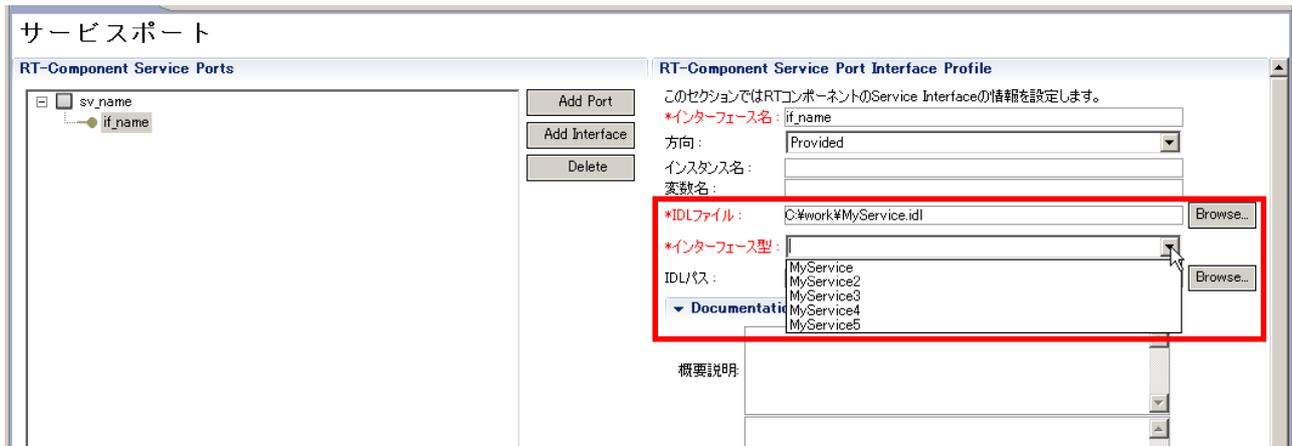
- ※データ型は、型定義が記載されたIDLファイルを設定画面にて追加することで追加可能
- ※OpenRTM-aistにて事前定義されている型については、デフォルトで使用可能
→ [RTM_Root]rtm/idl 以下に存在するIDLファイルで定義された型
- ※各ポートに対する説明記述を設定可能
→ 記述した各種コメントは、生成コード内にDoxygen形式で追加される

※Portの設定内容に応じて、下部のBuildViewの表示が変化



- InPort
ポート名: **originalImage**
データ型: **RTC::CameraImage**
変数名: **originalImage**
表示位置: **left**
- OutPort
ポート名: **flippedImage**
データ型: **RTC::CameraImage**
変数名: **flippedImage**
表示位置: **right**

■ 生成対象RTCに付加するServicePortの情報を設定



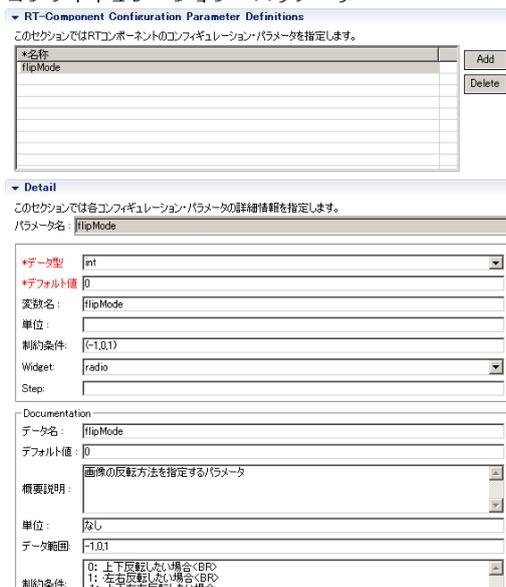
■ サービスインターフェースの指定

- IDLファイルを指定すると、定義されたインターフェース情報を表示

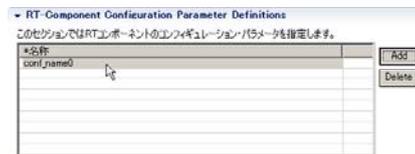
今回のサンプルでは未使用

■ 生成対象RTCで使用する設定情報を設定

コンフィギュレーション・パラメータ



- ①「Add」ボタンをクリックし、追加後、直接入力で名称設定



- ②詳細画面にて、型情報、変数名などを設定

名称: flipMode
データ型: int
デフォルト値: 0
変数名: flipMode
制約条件: (-1, 0, 1)
Widget: radio

- ※データ型は、short,int,long,float,double,stringから選択可能(直接入力も可能)
- ※制約情報とWidget情報を入力することで、RTSystemEditorのコンフィギュレーションビューの表示を設定することが可能

■ 制約条件について

- データポートとコンフィギュレーションに設定可能
- チェックはあくまでも**コンポーネント開発者側の責務**
 - ミドルウェア側で検証を行っているわけではない

■ 制約の記述書式

- 指定なし: 空白
- 即値: 値そのもの
 - 例) 100
- 範囲: <, >, <=, >=
 - 例) 0<=x<=100
- 列挙型: (値1, 値2, ...)
 - 例) (val0, val1, val2)
- 配列型: 値1, 値2, ...
 - 例) val0, val1, val2
- ハッシュ型: { key0:値0, key1:値1, ... }
 - 例) { key0:val0, key1:val1 }

■ Widget

- text(テキストボックス)
 - デフォルト
 - slider(スライダ)
 - **数値型**に対して**範囲指定**の場合
 - 刻み幅をstepにて指定可能
 - spin(スピナ)
 - **数値型**に対して**範囲指定**の場合
 - 刻み幅をstepにて指定可能
 - radio(ラジオボタン)
 - 制約が**列挙型**の場合に指定可能
- ※指定したWidgetと制約条件がマッチしない場合は、テキストボックスを使用

言語・環境・プロファイル

■ 生成対象RTCを実装する言語, 動作環境に関する情報を設定

言語・環境

▼言語
このセクションでは使用する言語を指定します

C++
 Python
 Java
 Ruby

Use old build environment.

▼環境
このセクションでは依存するライブラリや使用するOSなどを指定します

Version	OS

Add
Delete

詳細情報

OS Version	CPU

Add
Delete

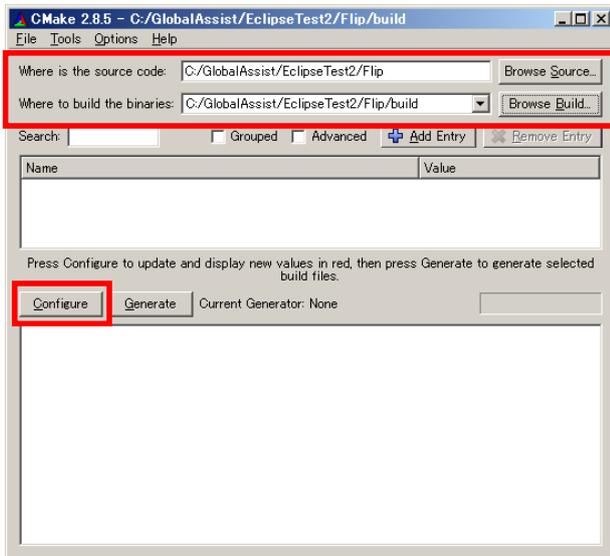
言語: RTコンポーネントを作成する言語を選択します。リスト中の言語から選択可能です。
環境: 言語ごとのライブラリの依存関係や、使用するOSなどの環境を選択します。
詳細情報で設定した内容(OS情報、ライブラリ情報などは、プロファイル内にも保存されます。

このチェックボックスをONにすると, 旧バージョンと同様なコード(Cmakeを利用しない形式)を生成

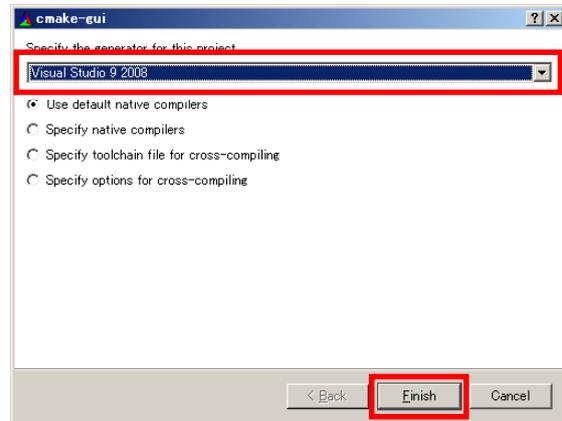
「C++」を選択

コンパイル(Windows,CMake利用)

① GUI版Cmakeを起動し, source, binaryのディレクトリを指定



② 「Configure」を実行し, 使用するプラットフォームを選択

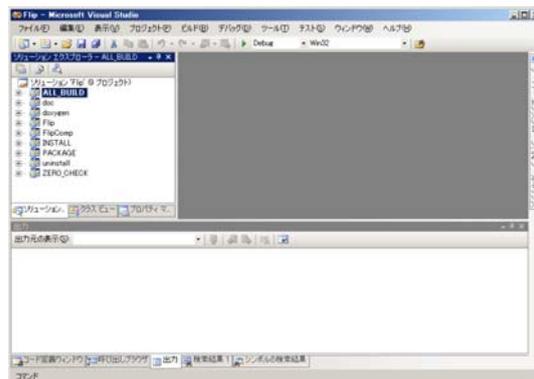
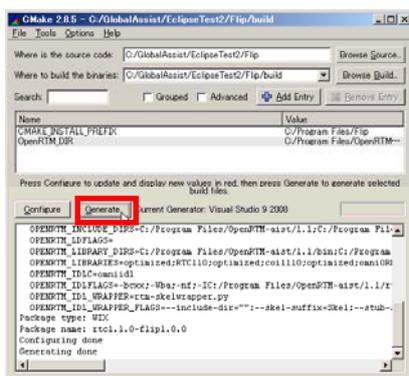


※binaryには, sourceとは別のディレクトリを指定する事を推奨

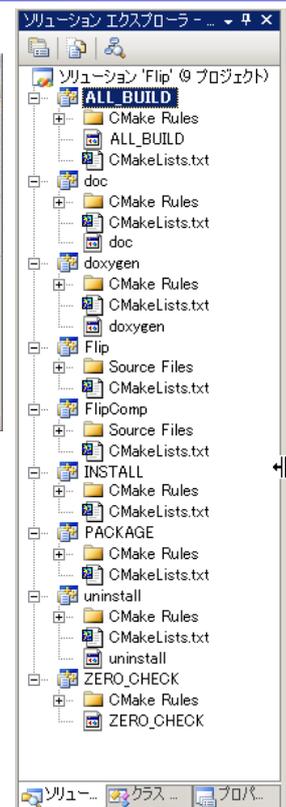
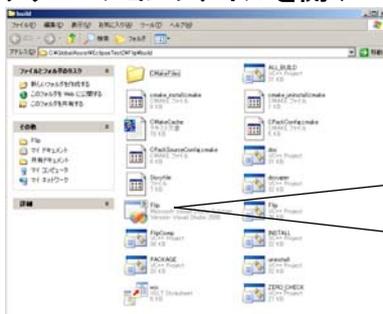
※日本語は文字化けしてしまうため英数字のみのディレクトリを推奨

コンパイル(Windows,CMake利用)

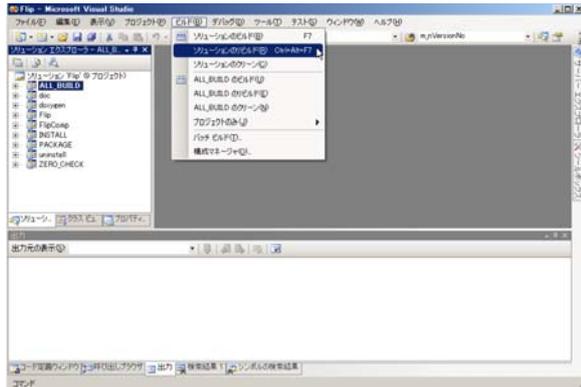
③ 正常終了後, 「Generate」を実行



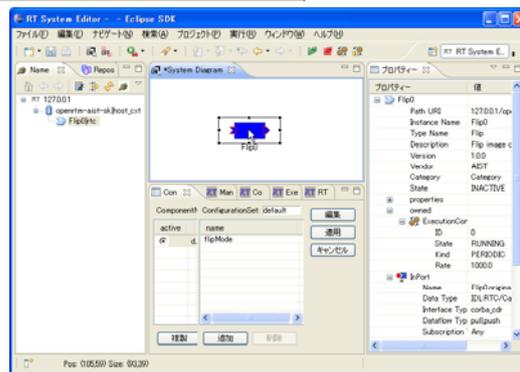
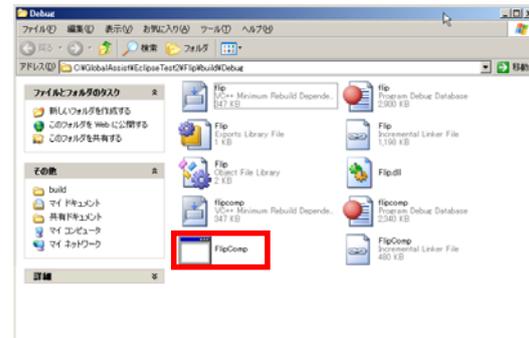
④ binaryとして指定したディレクトリ内にあるソリューションファイルを開く



⑤ ソリューションをビルド



⑥ binaryにて指定したディレクトリ以下の Debug内のFlipComp.exeを起動



ドキュメント作成 (Windows, CMake 利用)

※binaryにて指定したディレクトリ以下のdoc/html以下にdoxygenにて生成したドキュメント



■ 生成されたドキュメントの例

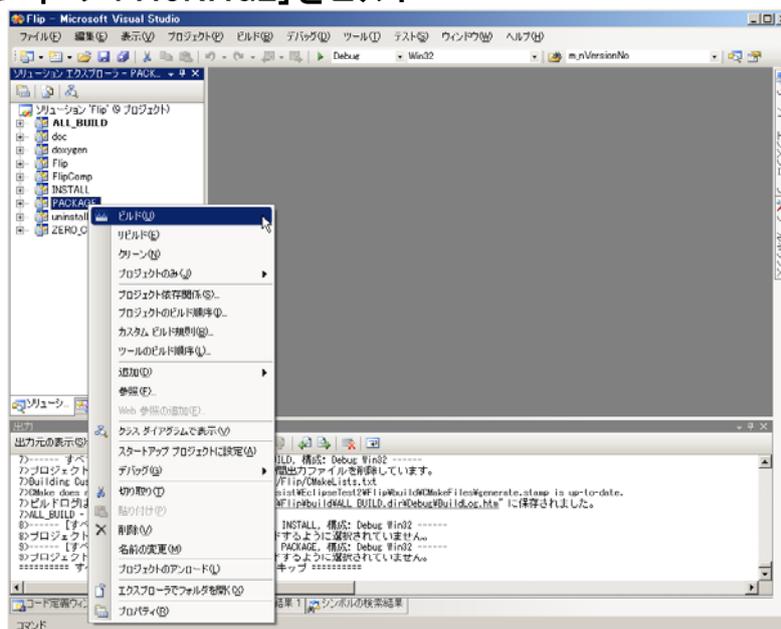
flip 1.0.0



flip 1.0.0



■ ソリューション中の「PACKAGE」をビルド



- binaryにて指定したディレクトリ直下にmsi形式のインストールパッケージを生成
 - コンポーネントのインストール先
C:¥Program Files¥OpenRTM-aist¥1.1¥components¥<言語>/<パッケージ名>

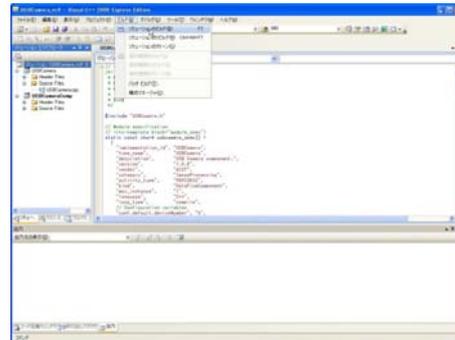
RTCBuilderの補足

コンパイル・実行(Windows)

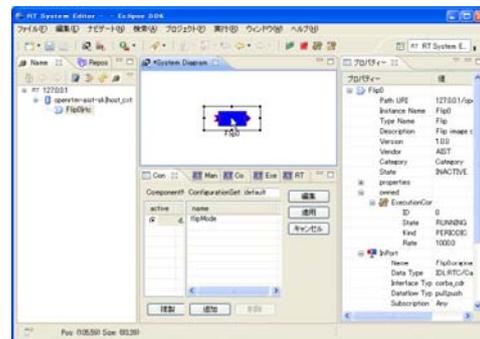
- ①コード生成先ディレクトリ内の「copyprops.bat」をダブルクリックして、設定ファイルをコピー



- ②VisualStudioを用いたビルド

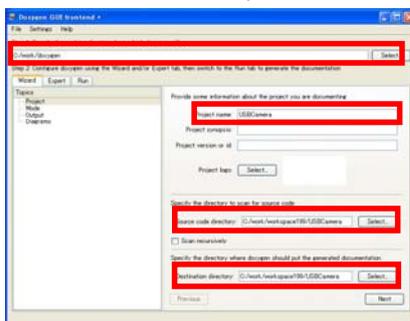


- ③FlipComp¥¥Debug内のFlipComp.exeを起動

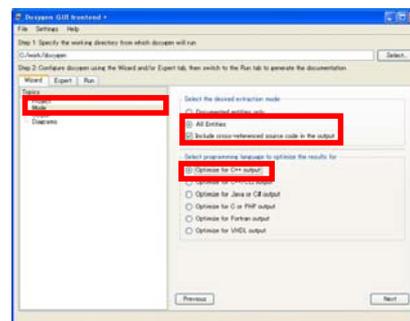


ドキュメント作成(Windows)

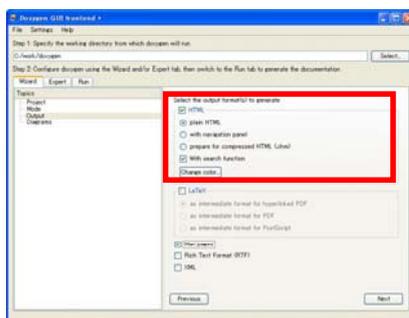
- ①Doxygen用GUIツールを起動
作業用ディレクトリ,ソース格納場所,
生成ファイル出力先,プロジェクト名を指定



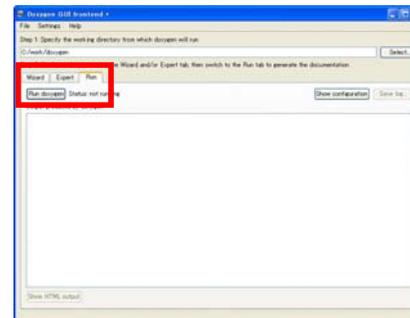
- ②「Mode」セクションにて,
出力内容,使用言語を指定



- ③「Output」セクションにて,html出力を指定



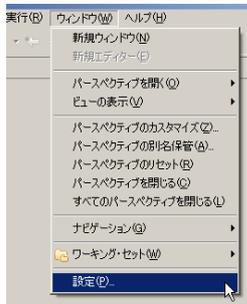
- ③「Run」タブにて,「Run doxygen」を実行



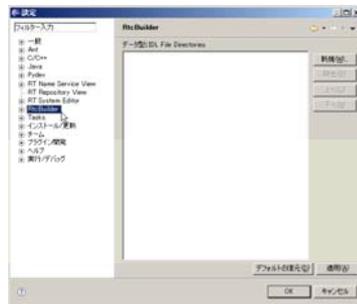
■ DataPortにて利用するデータ型の指定

→データ型を定義したIDLファイルが格納されているディレクトリを指定

①メニューから
「ウインドウ」→「設定」



②「RtcBuilder」を選択



③「新規」ボタンにて表示される
ディレクトリ選択ダイアログ
にて場所を指定



※独自に定義したデータ型を使用する場合のみ必要な設定

OpenRTM-aistにて標準で用意されている型のみを使用する場合には設定不要

・標準型の定義内容格納位置：[RTM_Root]rtm/idl

→BasicDataType.idl, ExtendedDataTypes.idlなど

→デフォルト設定では、[RTM_Root]=C:/Program Files/OpenRTM-aist/1.1/

熊本県産業技術センター
第7回技術普及講習会(組み込み技術)
「RTミドルウェアの概要と実習」