

# 第2部：Flipコンポーネント作成

会津大学



# Flipコンポーネントについて

- 入力した画像を反転して出力するコンポーネント



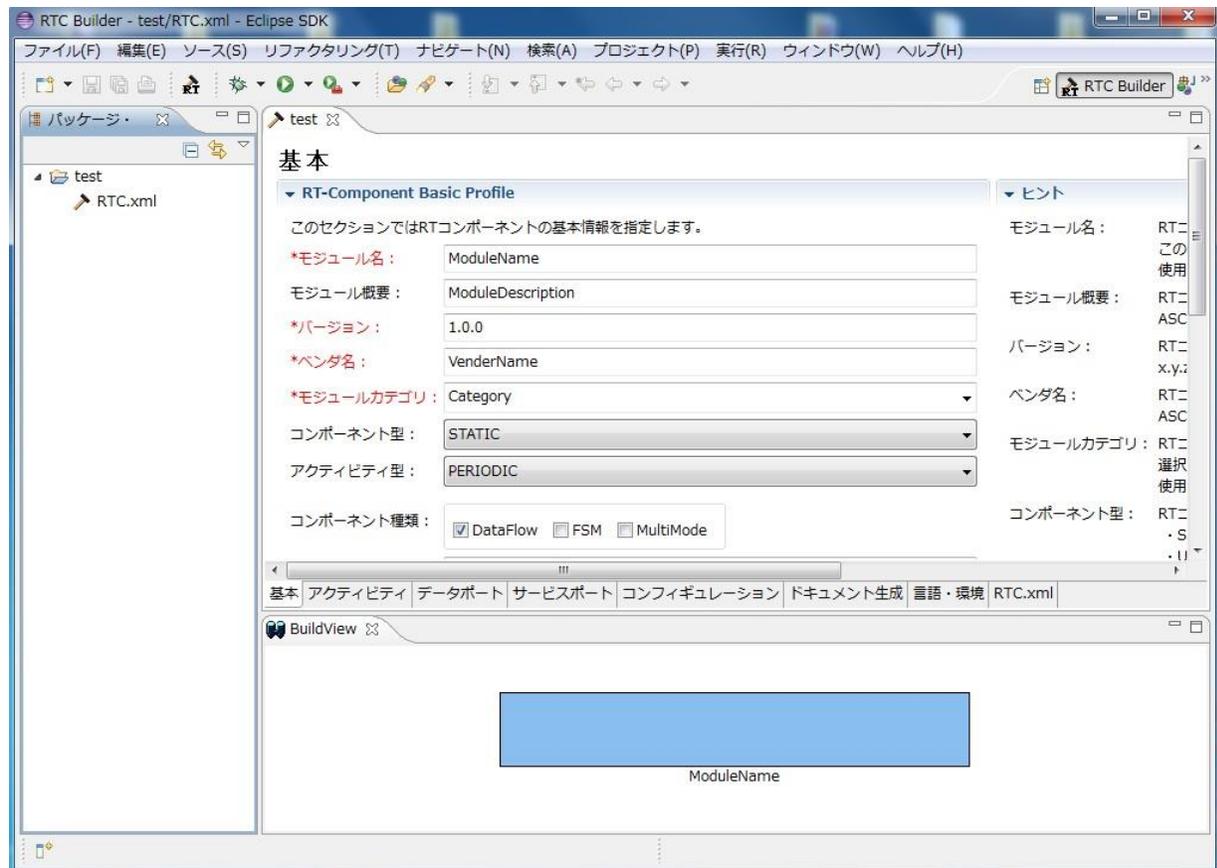
- OpenCVのcvFlip関数を利用して作成します。

# RTCBuilderについて



# RTCBuilder概要

- RTCBuilderとは？
- コンポーネントの情報を入力し雛形を生成するためのツール。
  - コンポーネントの仕様部分を設定
  - 各言語向けRTCの雛型を作成
    - C++
    - JAVA
    - Python



# RTCBuilder起動

- Windows

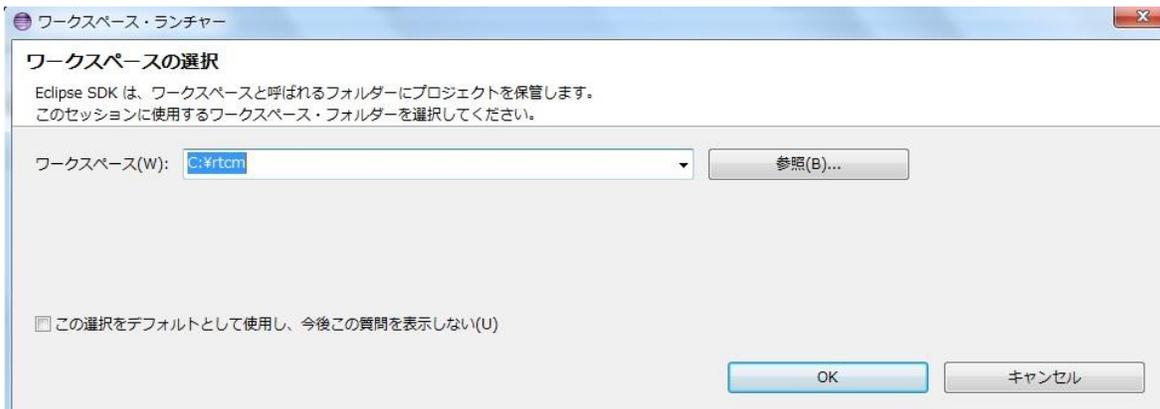
スタートメニューから「OpenRTM-aist x.y」→「tools」の下の[OpenRTP ]から起動

- Linux

コマンド [openrtp]で起動

起動後はワークスペースの選択を行います。

ここでは[C:¥rtcws]と記入します。※パスに日本語が入らないようにしてください。



# ・初回起動時のみ

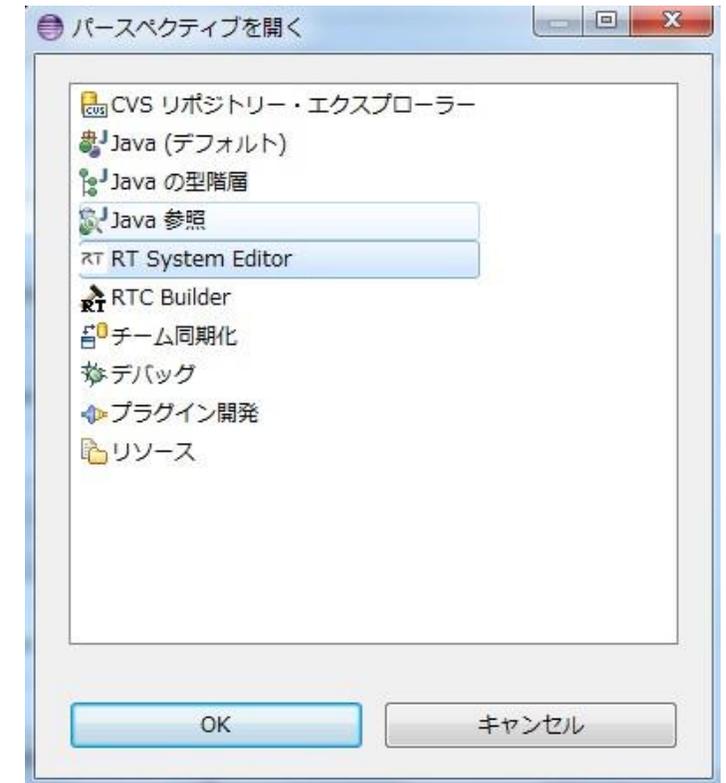


①初回起動時に出てくるようこそ  
の画面を×ボタンを押し、消します。

②画面右上の[パースペクティブを  
開く]を選択し、[その他]を選択



③一覧から[RTCBuilder]を選択して  
ください。



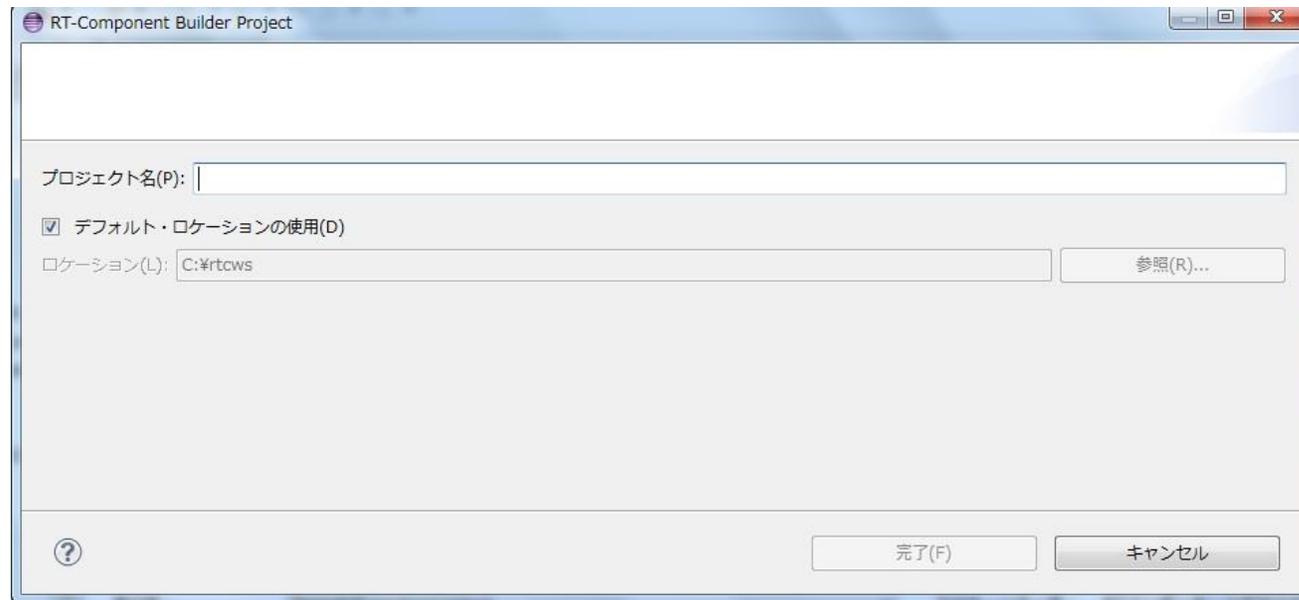
# 新規プロジェクトの作成



① ツールバー内の[RTとカナヅチ]のアイコンをクリック

② プロジェクト名を記入

プロジェクト名: Flip



# 基本プロファイル

## • コンポーネントの基本情報の設定

モジュール名: Flip  
 モジュール概要: 任意(Flip component)  
 バージョン: 1.0.0  
 ベンダ名: 任意(Aizu)  
 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

**▼ RT-Component Basic Profile**

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:

モジュール概要:

\*バージョン:

\*ベンダ名:

\*モジュールカテゴリ:

コンポーネント型:

アクティビティ型:

コンポーネント種類:  DataFlow  FSM  MultiMode

最大インスタンス数:

実行型:

実行周期:

概要:

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。

**▼ プロファイル情報のインポート・エクスポート**

プロファイル情報のインポートおよびエクスポートを行います。

# アクティビティ・プロファイル

## ・コンポーネントで使用予定のアクションコールバックを設定

### アクティビティ

▼ 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

実行コンテキストの起動と停止に関するアクション

onStartup onShutdown

alive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

Mode型コンポーネントのアクション

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ドキュメントを記述します。  
上段のアクションを選択すると、それぞれのドキュメントを記述できます。

アクティビティ名:   ON  OFF

動作概要:

事前条件:

事後条件:

- ①使用するアクティビティを選択する  
※マウスで選択すると文字が赤くなり  
アクティビティ名のところに表示されます。



- ②On/OFFを選択する。  
※選択するとアクティビティの背景が青くなります。

### 選択アクティビティ

- onActivated
- onDeactivated
- onExecute

# データポート・プロファイル

## • RTコンポーネントのデータポートを設定

データポート

▼ DataPortプロファイル

このセクションではRTコンポーネントのDataPort(データポート)の情報を設定します。

*ポート名 (InPort)		*ポート名 (OutPort)	
	Add		Add
	Delete		Delete

▼ Detail

このセクションではデータポート毎の概要を説明するドキュメントを記述します。  
上のデータポートを選択すると、それぞれのドキュメントが記述できます。

ポート名:

\*データ型: RTC::Acceleration2D

変数名:

表示位置: LEFT

Documentation

概要説明:

データ型:

データ数:

意味:

単位:

発生頻度, 周期:

処理頻度, 周期:

基本 アクティビティ データポート サービスポート コンフィギュレーション ドキュメント生成 言語・環境 RTC.xml

*ポート名 (InPort)	
originalImage	

Add

Delete

①追加したいポートの[Add]ボタンを押しポートを追加します。  
その後直接名称を変更します。

②ポートのデータ型と変数名を入力します。

\*データ型: RTC::CameraImage

変数名: originalImage

表示位置: LEFT

• 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

# サービスポート・プロファイル

- RTコンポーネントのサービスポートを設定

サービスポート

**RT-Component Service Ports**

sv\_name

**RT-Component Service Port Profile**

このセクションではRTコンポーネントのServicePortの情報を設定します。

\*ポート名:

表示位置:

▼ Documentation

概要説明:

I/F概要説明:

- 今回は使用しません。

# コンフィギュレーション・プロファイル

## ・RTコンポーネントのコンフィギュレーションを設定

①[Add]ボタンを押し追加します。  
その後直接名称を変更します。

コンフィギュレーション・パラメータ

RT-Component Configuration Parameter Definitions

このセクションではRTコンポーネントのコンフィギュレーション・パラメータを指定します。

*名称	

Add  
Delete

Detail

このセクションでは各コンフィギュレーション・パラメータの詳細情報を指定します。

パラメータ名:

\*データ型: short

\*デフォルト値:

変数名:

単位:

制約条件:

Widget: text

Step:

Documentation

データ名:

デフォルト値:

概要説明:

単位:

データ範囲:

制約条件:

RT-Component Configuration Parameter

このセクションではシステム・コンフィギュレーション・パラメータを指定します。

Configuration	デフォルト値

Add  
Delete

基本 | アクティビティ | データポート | サービスポート | コンフィギュレーション | ドキュメント生成 | 言語・環境 | RTC.xml

*名称	
flipMode	

Add  
Delete

②詳細画面でデータ型、デフォルト値などを記入します。

\*データ型: int

\*デフォルト値: 0

変数名: flipMode

単位:

制約条件: (-1, 0, 1)

Widget: radio

Step:

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

# 言語・環境・プロファイル

- RTコンポーネントの実装する言語, 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++  
 Java  
 Python  
 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

その他OS情報

その他CPU情報

ライブラリ情報

Name	Version	Info.

Add  
Delete

• C++選択

# コード生成

## • 設定内容を確認

### ・基本情報

モジュール名: Flip  
 モジュール概要: 任意(Flip component)  
 バージョン: 1.0.0  
 ベンダ名: 任意(Aizu)  
 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

### ・選択アクティビティ

- onActivated
- onDeactivated
- onExecute

### ・コンフィギュレーション

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

### ・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

・言語  
 C++

## • 全ての設定が終わりましたらコード生成を行います。

### ▼ 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。

コード生成

パッケージ化

①コード生成のボタンを押します。

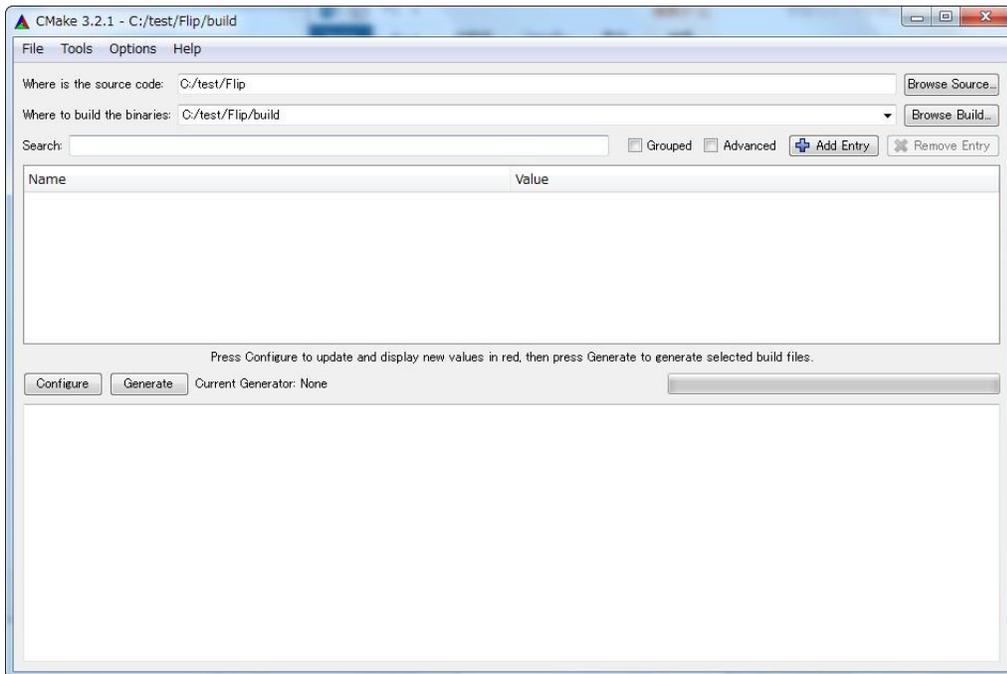
# コンパイルの仕方



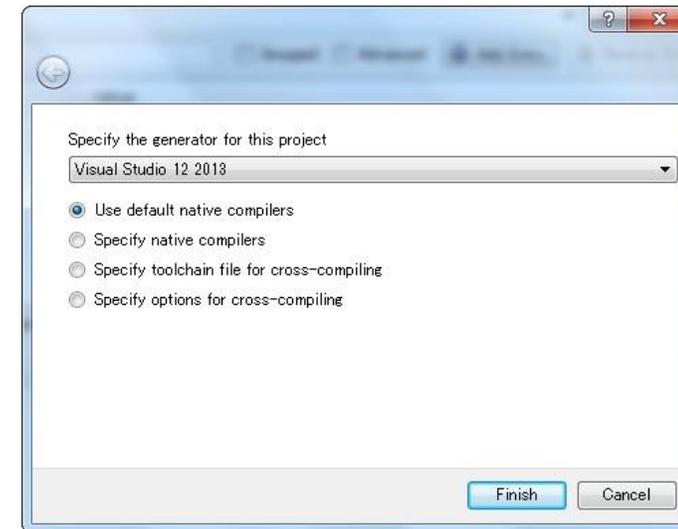
# cmake

- cmakeを使い自分のビルド環境に変更 (Windows版)

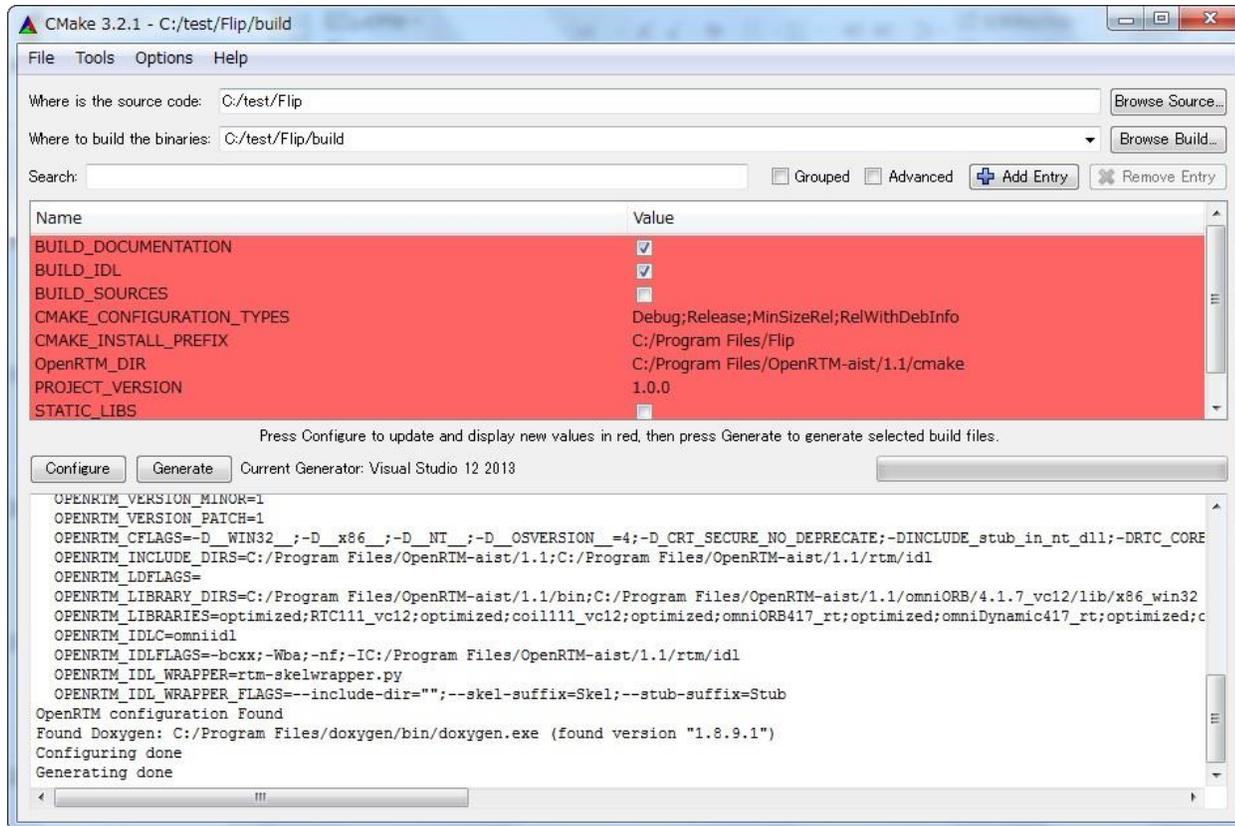
①cmake起動後, source,binaryのディレクトリを指定  
SourceはFlipディレクトリ、binaryはFlip/buildディレク  
トリーを指定。



②[Configure]ボタンを押し、使用するプラットフォーム  
を選択してください。



### ③[Configure Done]と出た後[Generate]ボタンを押す



### ④[Generate Done]と出た後、binaryに指定したディレクトリにソリューションファイルが出来たら完了



- Linux版  
下記コマンドを打ち込む

```

$ cd rtcws/Flip
$ mkdir build
$ cd build
$ cmake ..
  
```

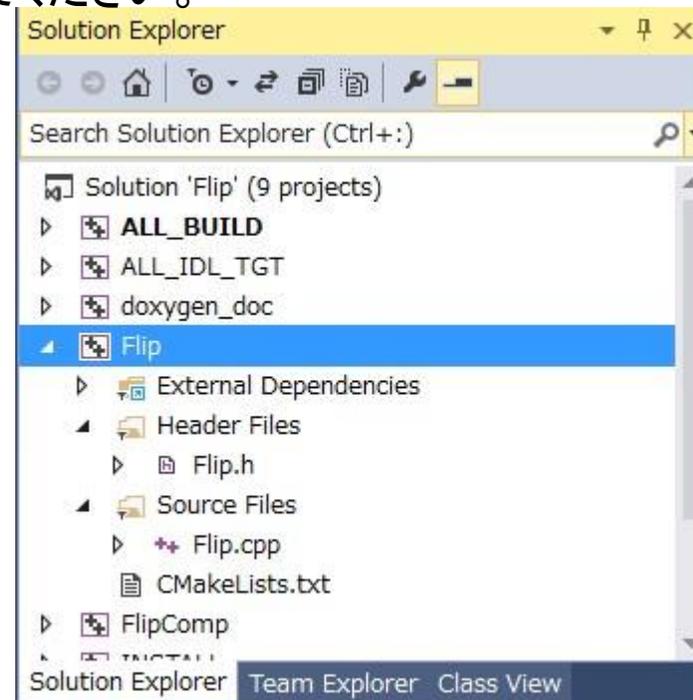
# ヘッダ、ソースの編集

①Cmakeで作成したバイナリーフォルダ内のソリューションファイルを開きます。



- Linux版  
下記ファイルを編集する  
/rtcws/Flip/include/Flip/Flip.h  
/rtcws/Flip/src/Flip.cpp

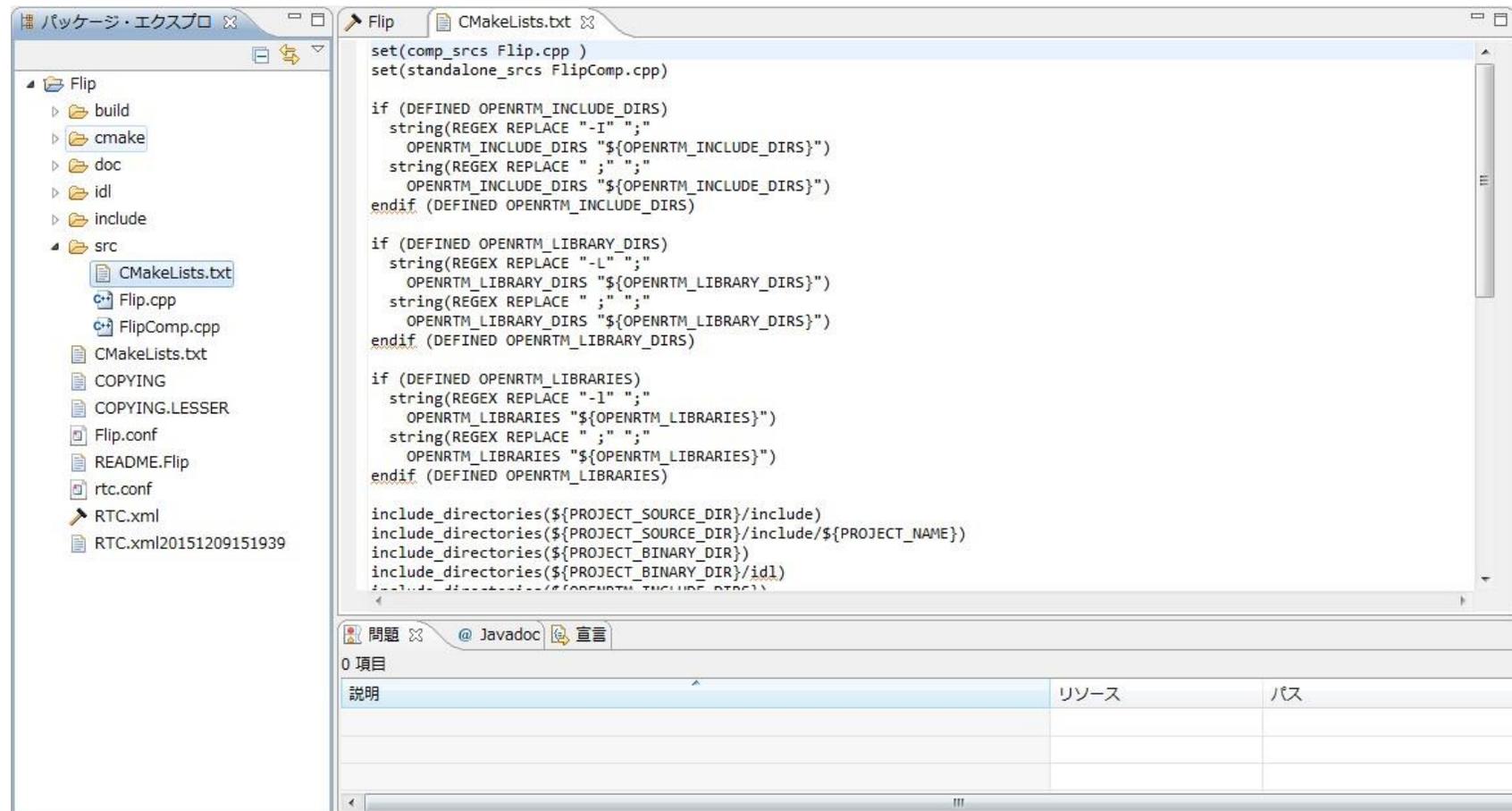
②ソリューションを開き、[Flip.h]と[Flip.cpp]を開き、編集をしてください。



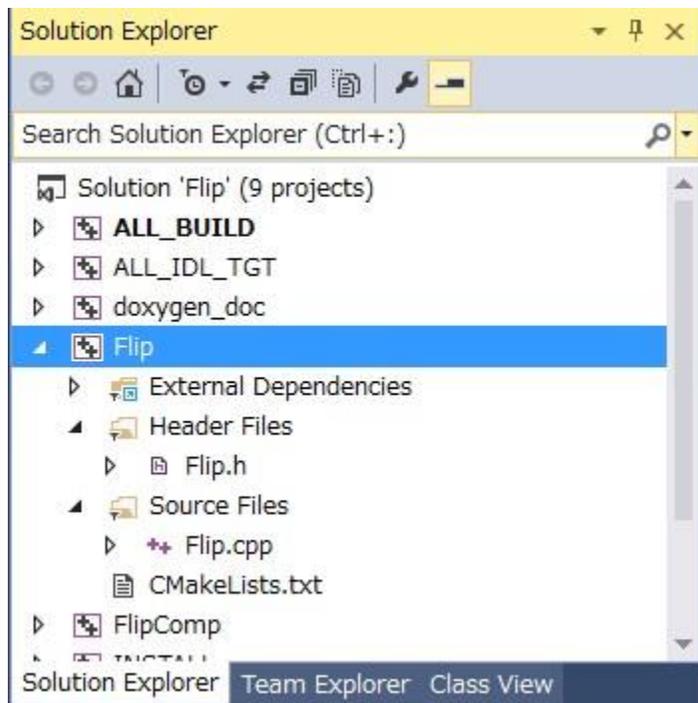
# CMakeLists.txt

OpneCVを使用するためにCMakeList.txtを編集します。

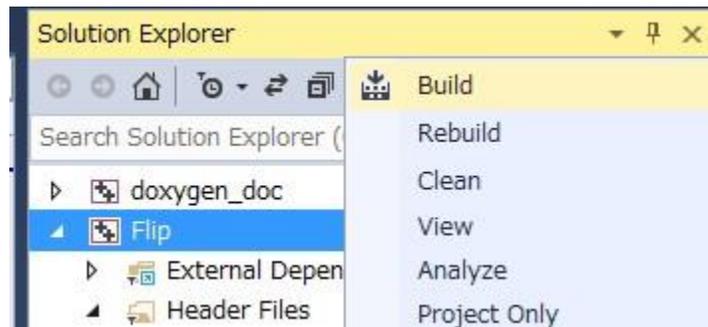
①Flip/src内の  
CMakeList.txtを編集します。



# ビルドの実行



①右クリックでメニューを開きビルドをしてください。エラーが出なければ完了です。



- Linux版  
下記コマンドを打ち込む

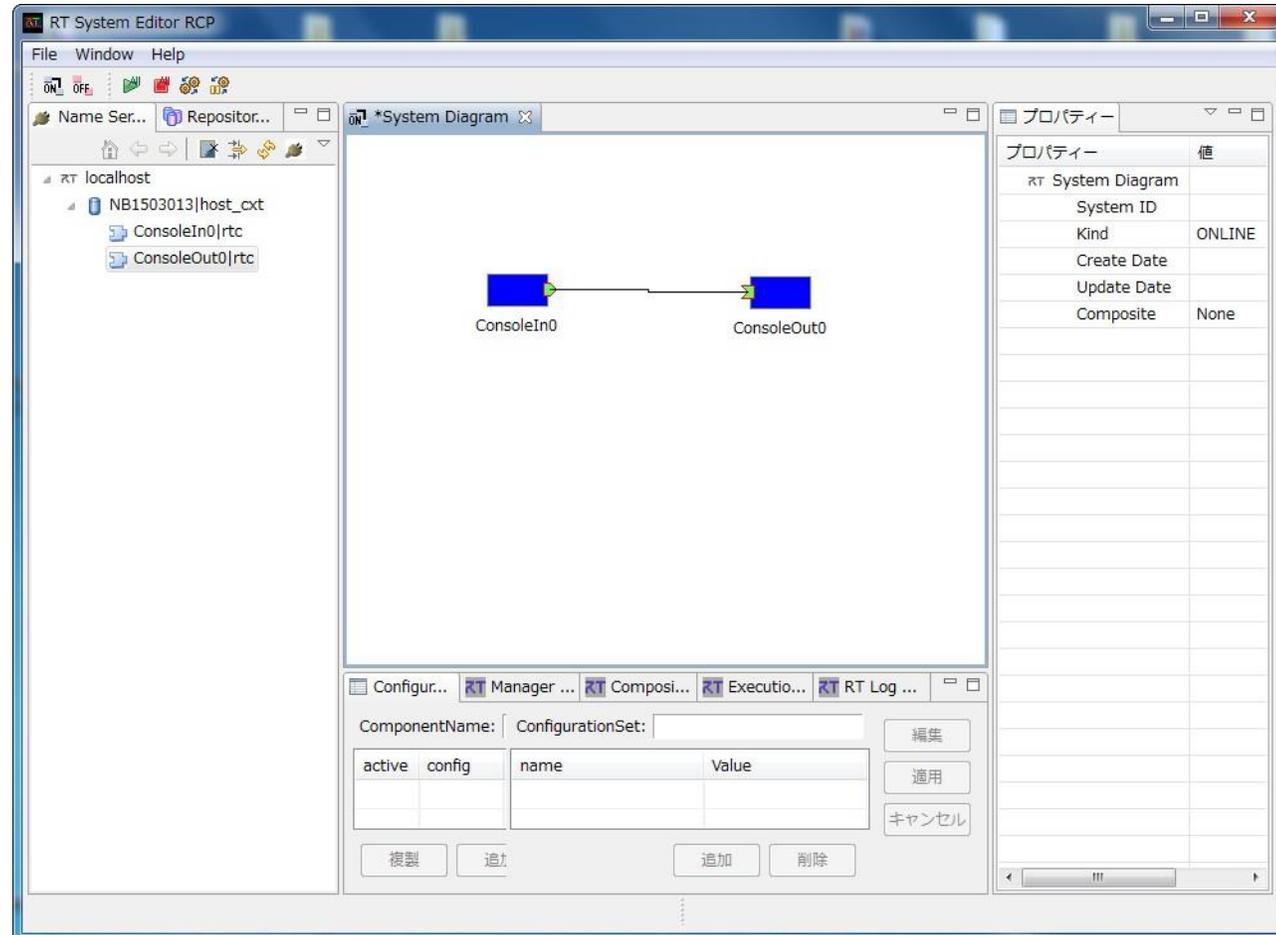
```
$ cd rtcws/Flip/build
$ make
```

# RTSystemEditorについて



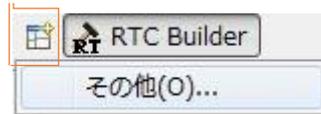
# RTSystemEditor概要

- RTコンポーネントを接続してRTシステムを構築するためのツール



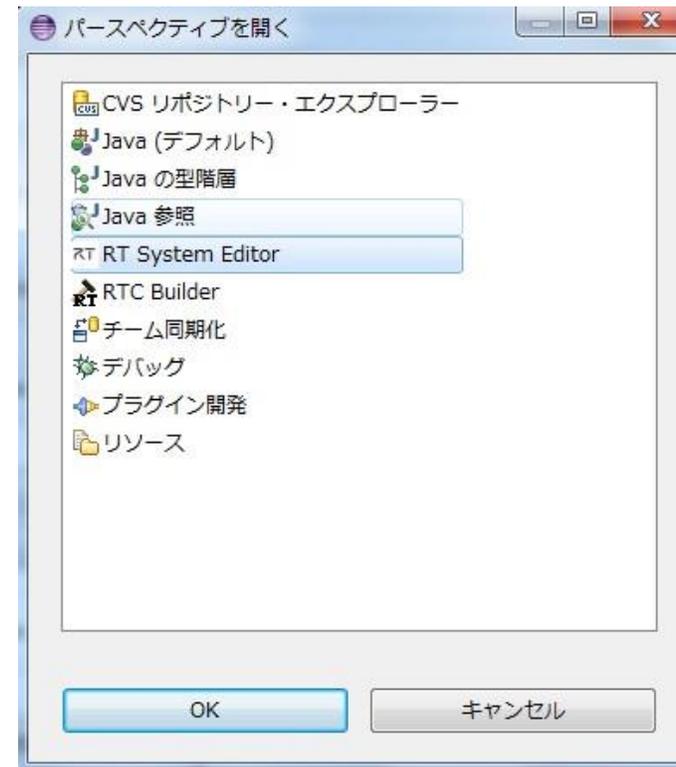
# RTSystemEditor起動

## ・ 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、さらに「その他」を選択

②一覧の中から[RT System editor]を選択



# カメラ関連コンポーネント起動

- Naming Serviceの起動

[スタート]メニューから

[プログラム]→[OpenRTM-aist x.y]→ [tools]→[Start Naming Service]

- CameraViewerCompの起動

[スタート]メニューから起動

[プログラム]→[OpenRTM-aist x.y]→[C++]→[Components]→[OpenCV-Examples]→ [CameraViewerComp.exe]

- OpenCVCameraCompの起動

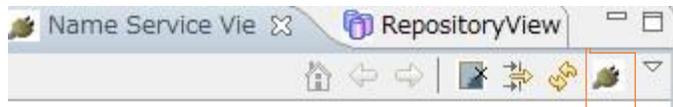
[スタート]メニューから起動

[プログラム]→[OpenRTM-aist x.y]→[C++]→[Components]→[OpenCV-Examples]→ [OpenCVCameraComp.exe]

# ネームサービスへ接続

## • 接続方法

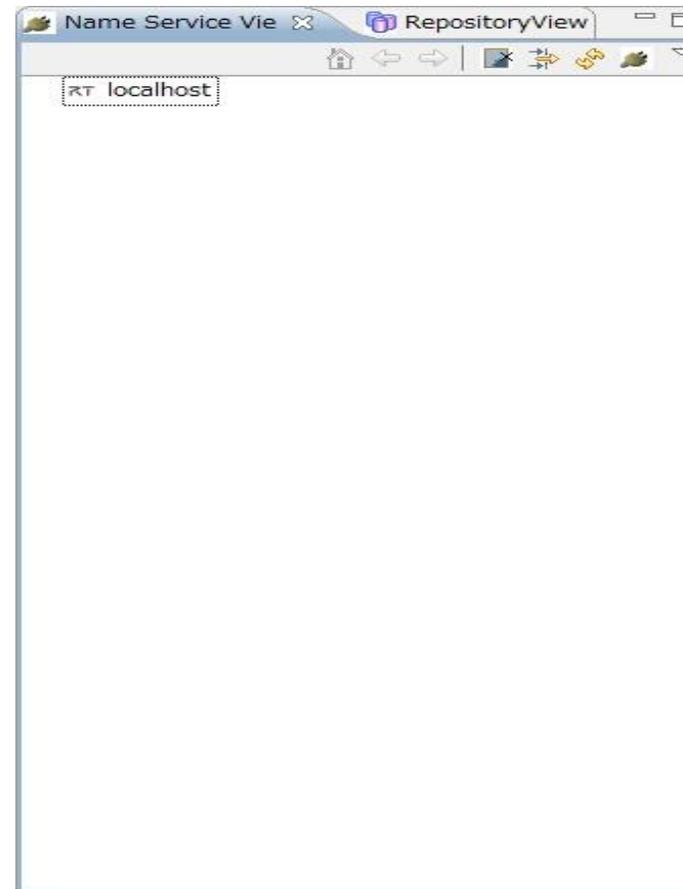
③Name Service Viewに記入したものが記載されれば完了



①Name Service View のコンセントのアイコンをクリック

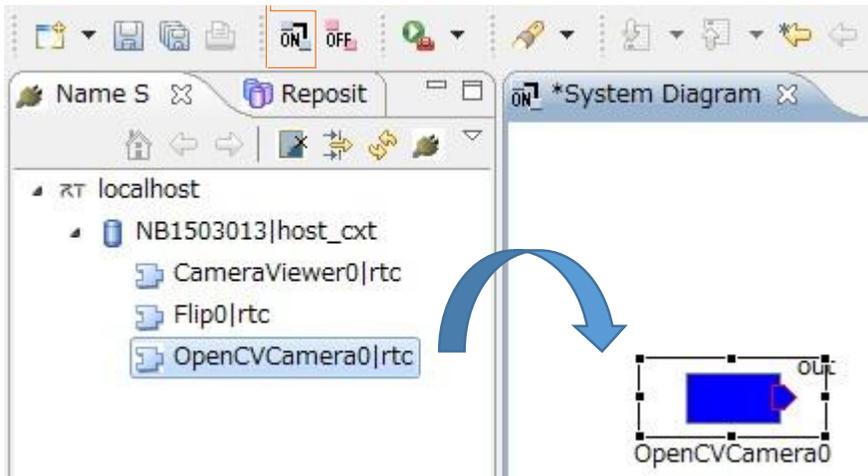


②ネームサーバのアドレスを聞かれるので、ホスト名かIPアドレスを記入



# コンポーネント配置

## • コンポーネントの配置

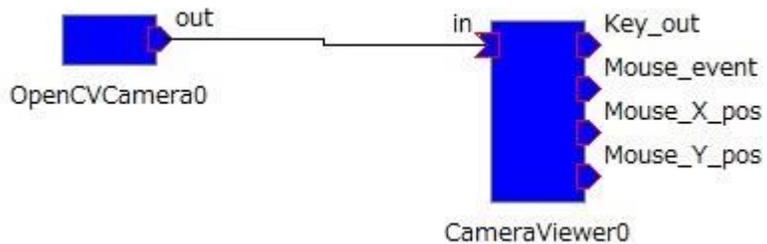


※System Diagramが開いてない場合は[ON]のボタンをクリックしてください。

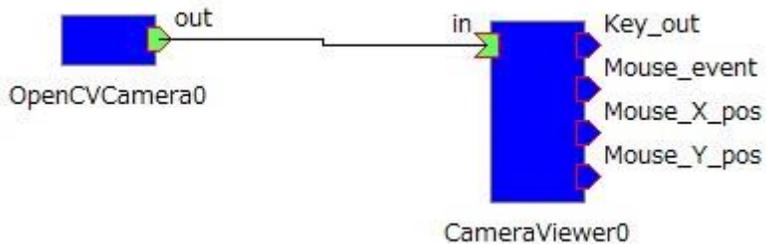
①対象コンポーネントをドラックアンドドロップでコンポーネントを配置

# データポート接続

## • データポートを接続



①接続元のポートから接続先のポートまでドラックアンドドロップ



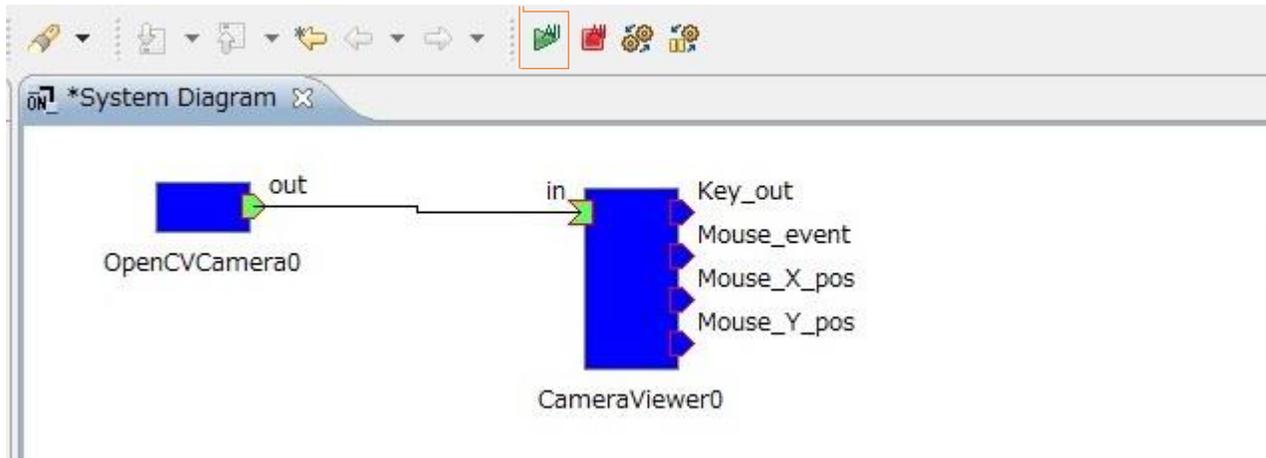
③接続しているポートが緑になれば完了

②接続プロファイルを記入してOKをクリック

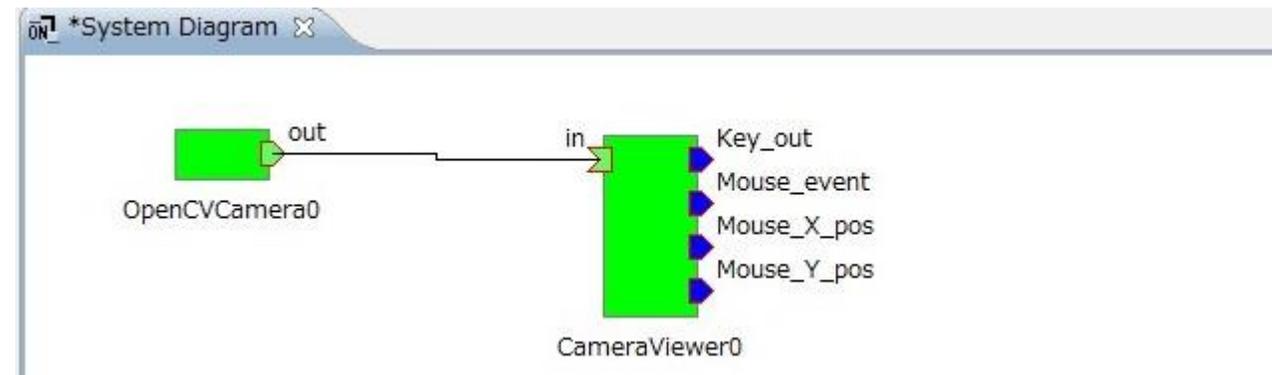


# コンポーネントアクティブ

- コンポーネントの起動方法



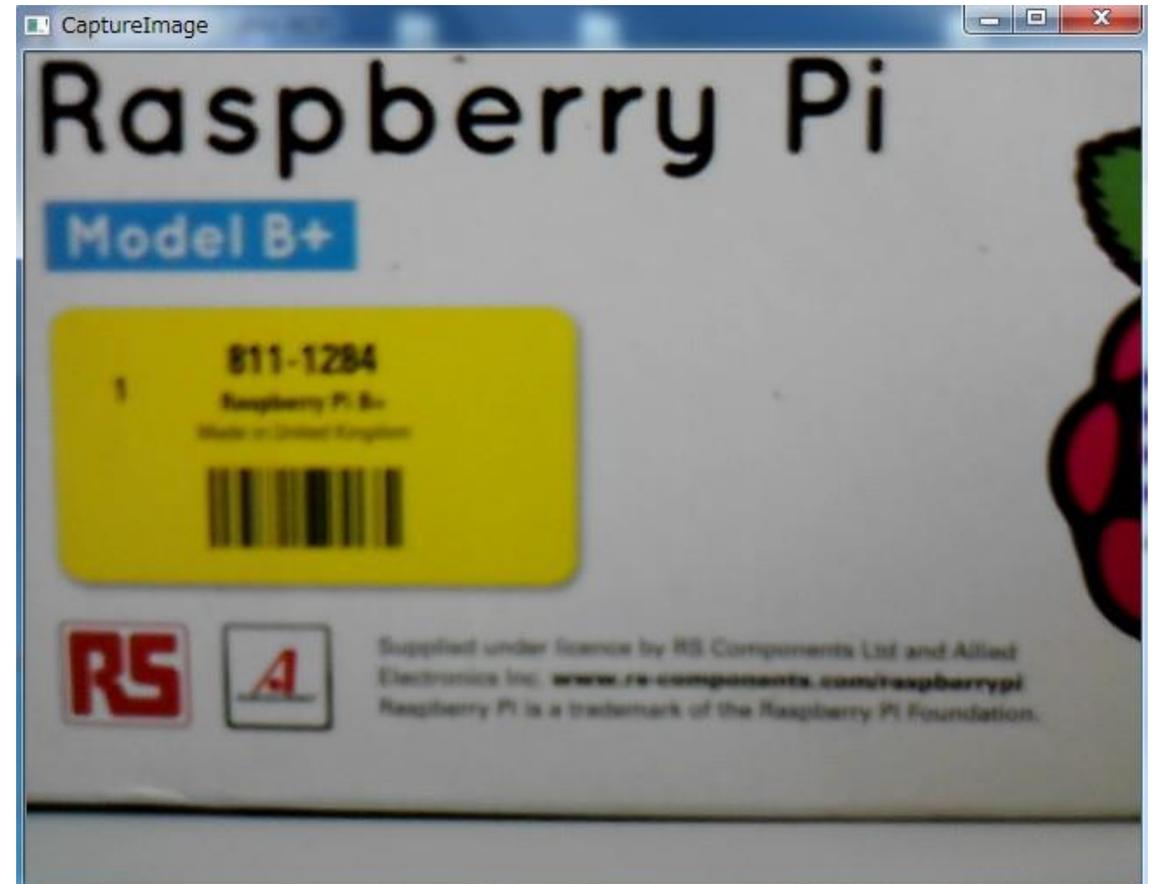
①画面の緑の矢印をクリックか、右クリックから[All Activate]を選択。  
コンポーネントが全て緑になれば完了です。



# コンポーネント動作確認

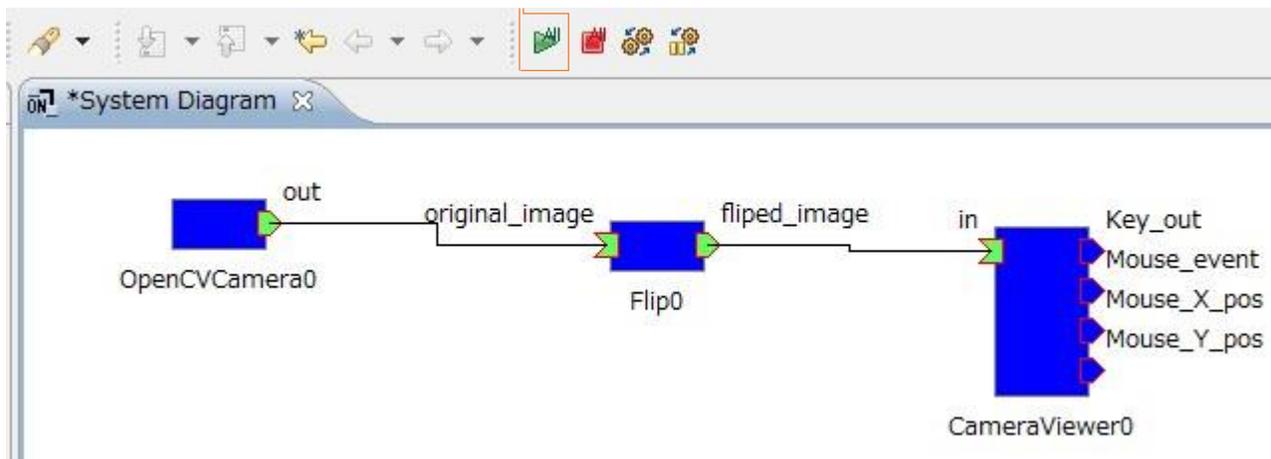
- 確認方法

ウィンドウが出てきてカメラから取得した画像が表示されることを確認してください。

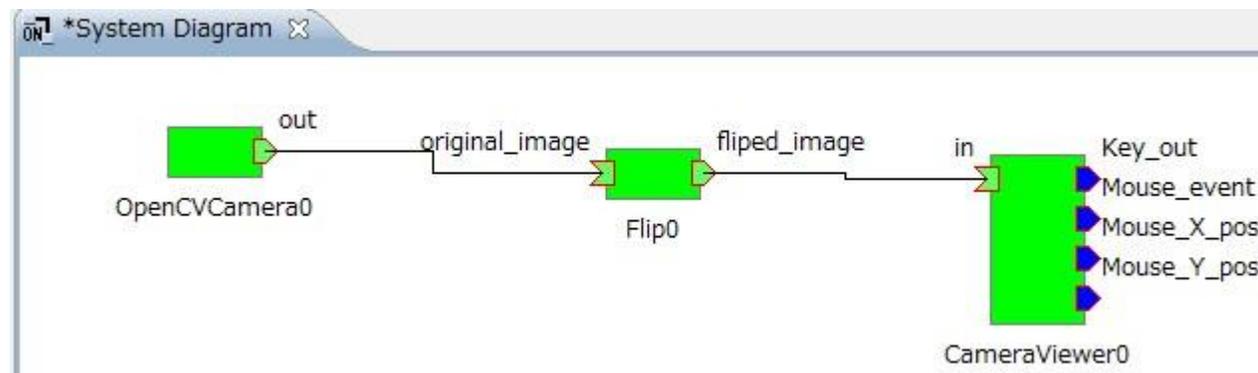


# Flipコンポーネント接続

- Flipコンポーネント接続

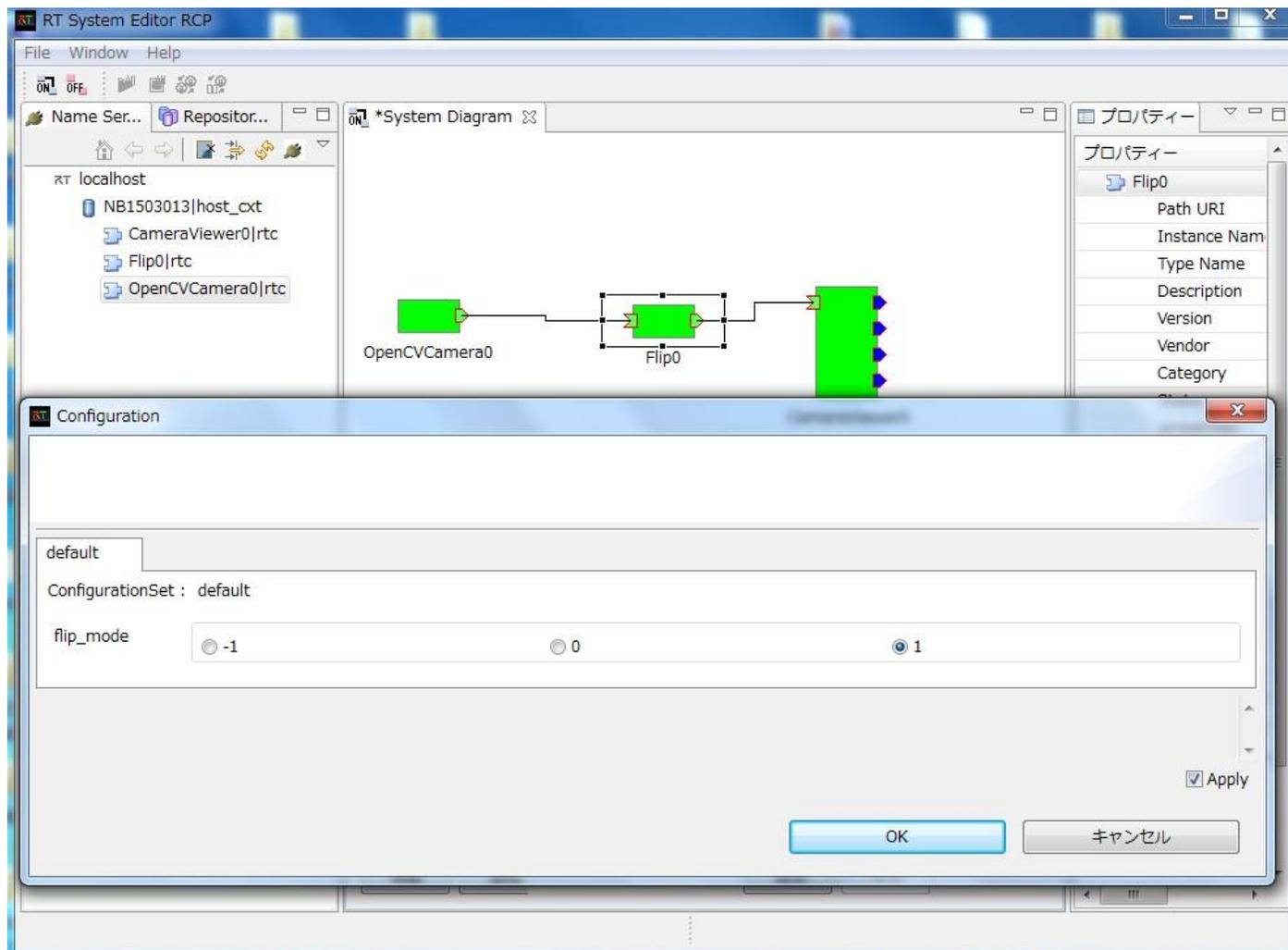


図の様にFlipコンポーネントを接続してアクティブにしてください。



# コンポーネント動作確認

- 確認方法  
 flipModeの値を変更して画像が反転することを確認してください。



# ネットワーク上の他のRTCとの接続

- IPアドレスの確認

- 1)スタートメニュー中の「全てのプログラム」-「アクセサリ」-「コマンドプロンプト」

- 2)コマンド「ipconfig」を実行

- 他PC上で動作するRTCとの接続

- 1)隣の方のIPアドレスを聞く

- 2)RTSystemEditorの「ネームサーバを追加」をクリックして, 上記のIPアドレスを入力する

# 引用元

このスライドは下記文献を参考に作成しています。

- RTコンポーネントの作成入門

<http://www.openrtm.org/openrtm/sites/default/files/5820/150624-02.pdf>

(2016/01/07アクセス)