

PFCore(RTミドルウェア)トレーニング 初級編

日時:2013年1月21日(月) 13:00~17:00

場所:大阪大学 吹田キャンパス 産学連携本部B棟1F会議室



PFCore(RTミドルウェア)トレーニング 初級編

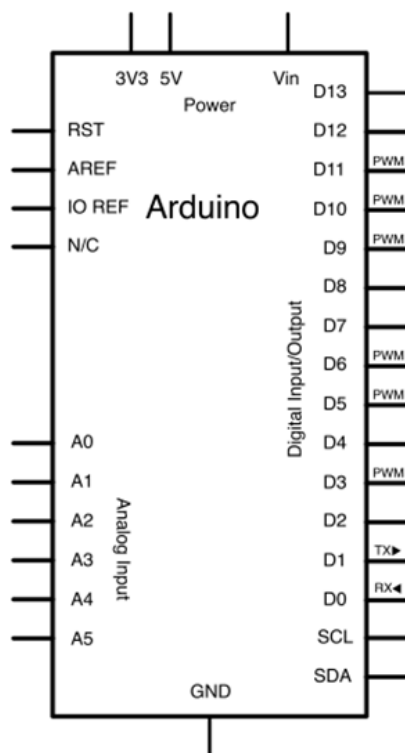


| | |
|-----------------|---|
| 13:00- 13:20 | 第1部:先端計測分野におけるRTミドルウェア(PFCore) |
| | 担当:市田 秀樹(大阪大学 e-square) |
| | 概要:RTミドルウェア(PFCore)およびRTコンポーネントの概要について解説します。 |
| 13:20- 14:10 | 第2部:RTミドルウェア(PFCore)の概略紹介 |
| | 担当:神徳 徹雄(産業技術総合研究所) |
| | 概要:RTミドルウェア(PFCore)およびRTコンポーネントの概要について解説します。 |
| 14:10- 15:05 | 第3部:RTミドルウェア(PFcore)サンプルコンポーネントの紹介とその利用方法 |
| | 担当:原 功(産業技術総合研究所) |
| | 概要:RTコンポーネントを使用した開発事例を紹介し、RTコンポーネントの便利さ、面白さを体感して頂きます。 |
| 15:05- 15:20 | 休憩 |
| 15:20- 17:00 | 第4部:RTミドルウェア(PFCore)開発支援ツールとシステム構築方法 |
| | 担当:坂本 武志(株式会社 グローバルアシスト) |
| | 概要:OpenRTM-aistのインストール方法を紹介するとともに、標準ツールRTSystemEditorを使用して、実際のシステム構築方法などを紹介させていただきます。 |

■ Digital IN/OUT, Analog INが容易にできるマイコン



Microcontroller: ATmega328
Operating Voltage: 5V
Input Voltage (recommended): 7-12V
Input Voltage (limits): 6-20V
Digital I/O Pins: 14
(of which 6 provide PWM output)
Analog Input Pins: 6
DC Current per I/O Pin: 40 mA
DC Current for 3.3V Pin: 50 mA
Flash Memory: 32 KB (ATmega328) of which
0.5 KB used by bootloader
SRAM: 2 KB (ATmega328)
EEPROM: 1 KB (ATmega328)
Clock Speed: 16 MHz

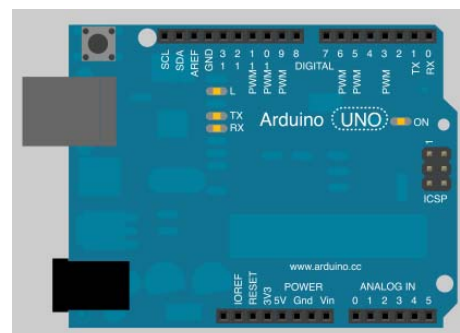


■ Digital IN/OUT : D0~D13

■ Analog IN : A0~A5

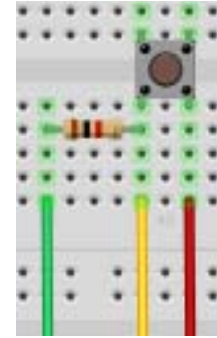
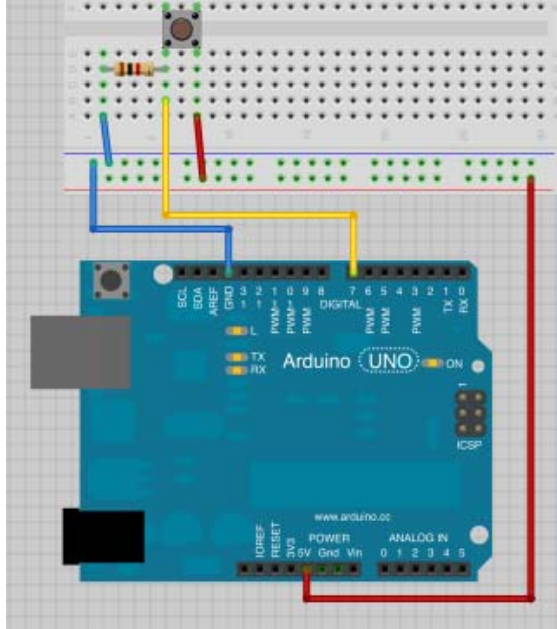
入力範囲 0~5 V

分解能 10bit



■ Digital Out

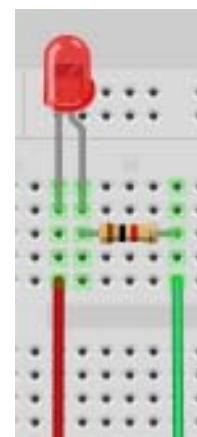
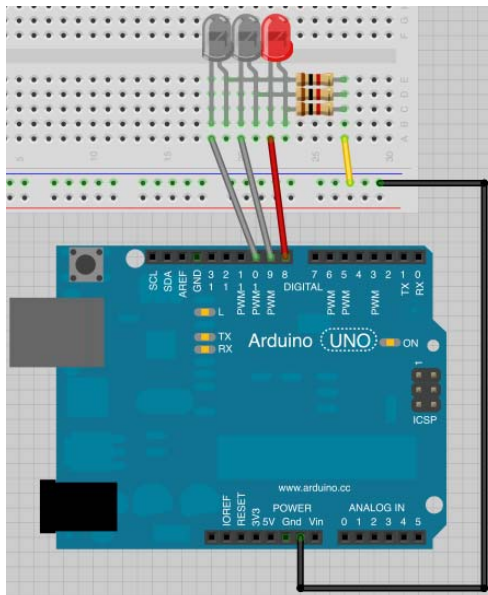
- ブレッドボード上のタクトスイッチのON/OFFをデータポートから出力するコンポーネント



GND D7 5V

■ Digital In

- データポートに入力されたデータを基に、ブレッドボード上のLEDを点灯するコンポーネント



D8 GND

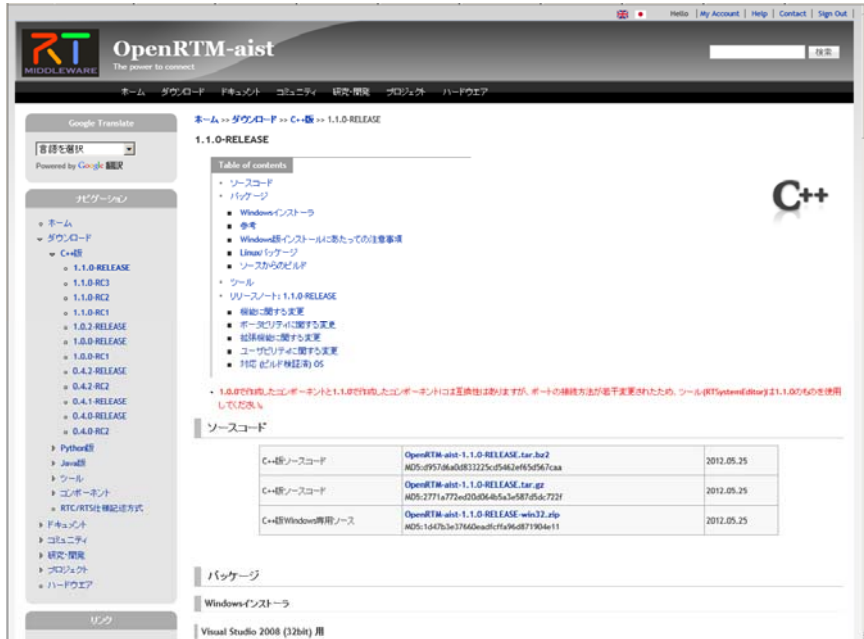
第4部 RTミドルウェア(PFCore)開発支援ツールと システム構築方法



RTミドルウェアのインストール



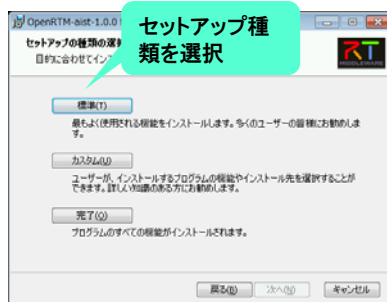
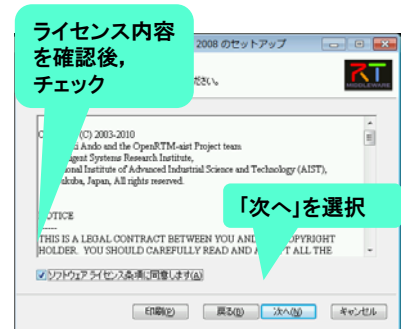
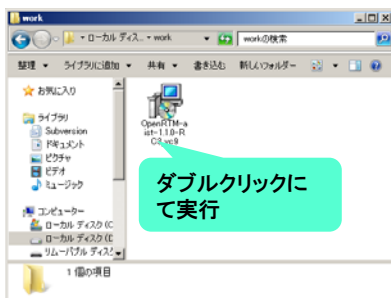
- OpenRTM-aist公式ページの「ダウンロード」ページ
 - <http://www.openrtm.org/openrtm/ja/node/5012>



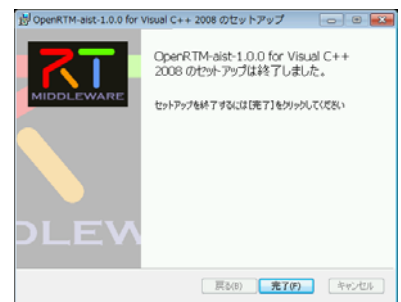
※使用環境に対応したパッケージをダウンロード

RTミドルウェアのインストール

- ダウンロードしたインストーラを実行
 - OpenRTM-aist-1.1.0-RC3_vc9.msiなど



- 標準:全ての機能をインストール
- カスタム:インストールする機能を選択
- 完了:全ての機能をインストール (標準と同様)



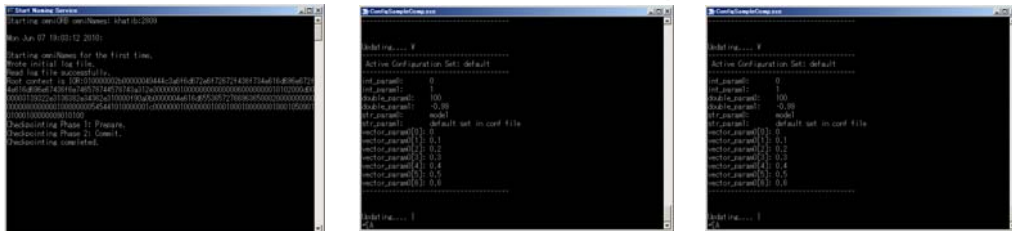
サンプルコンポーネントの起動

- Naming Serviceの起動
 - [スタート]メニューから
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[tools]→[Start Naming Service]

- CosoleInCompの起動
 - [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[examples]→ [ConsoleInComp.exe]

CosoleOutCompの起動

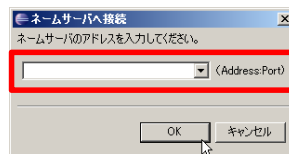
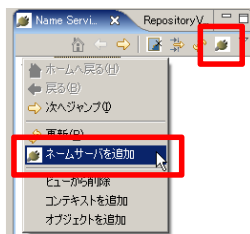
- [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[examples]→ [ConsoleOutComp.exe]



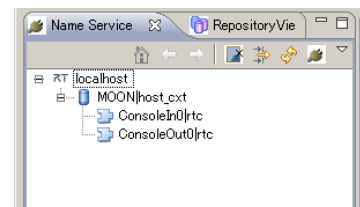
サンプルコンポーネントの動作確認

- ツールの起動
 - [スタート]メニューから
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[tools]→[RTSystemEditor]

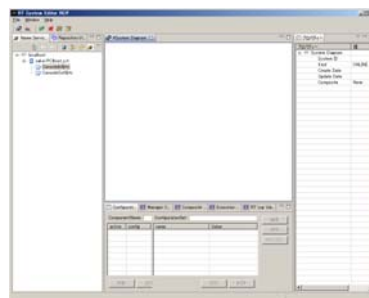
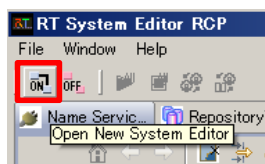
- ネームサービスへ接続



※対象ネームサーバのアドレス、ポートを指定
→ポート省略時のポート番号は
設定画面にて設定可能

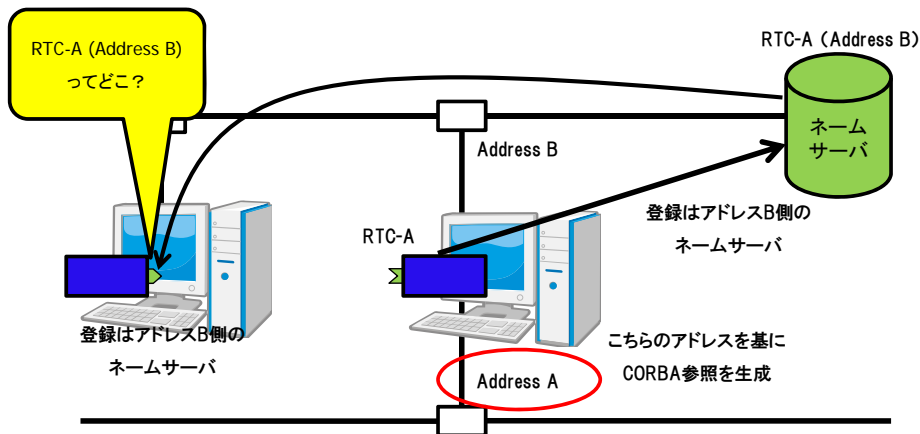


- システムエディタの起動



ネームサービスに接続できない場合

■ ネットワークインターフェースが2つある場合



■ RTC.confについて

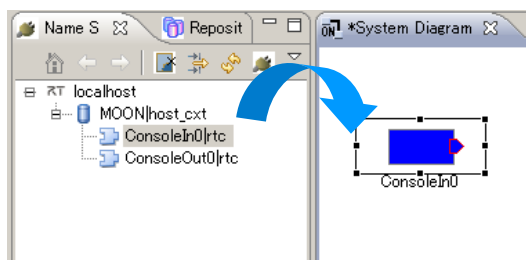
- RTC起動時の登録先NamingServiceや、登録情報などについて記述

■ 記述例:

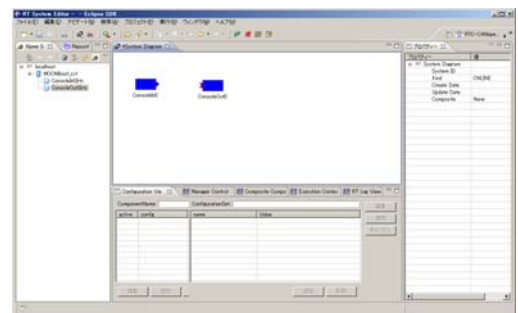
- **corba.nameservers:** localhost:9876
- **naming.formats:** SimpleComponent/%n.rtc
- **corba.endpoints:**192.168.0.12:

サンプルコンポーネントの動作確認

■ RTコンポーネントの配置

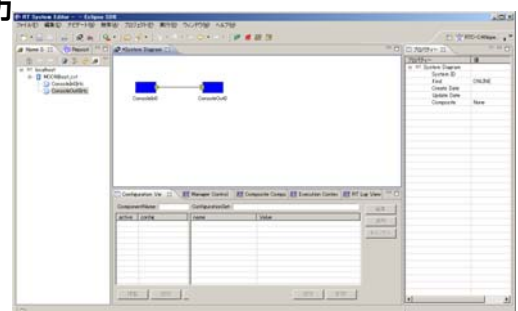
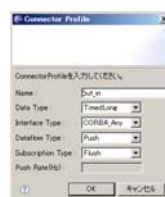
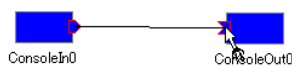


※ネームサービスビューから対象コンポーネントをドラッグアンドドロップ



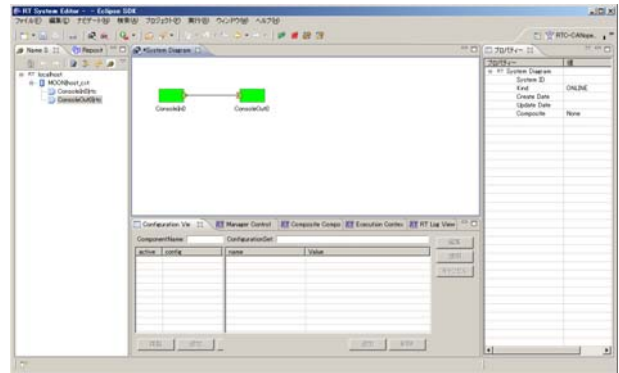
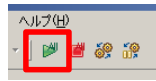
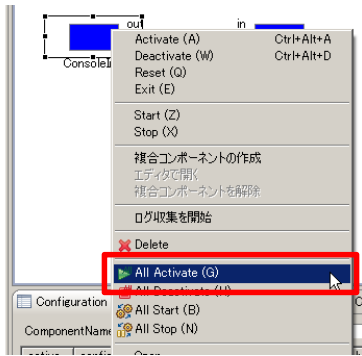
■ ポートの接続

- ①接続元のポートから接続先の②接続プロファイルを入力ポートまでドラッグ



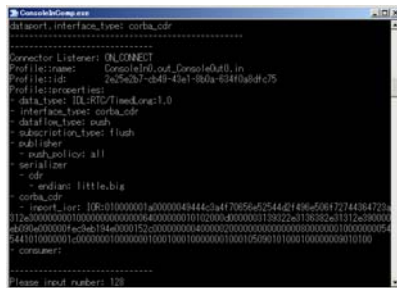
サンプルコンポーネントの動作確認

■ コンポーネントの起動

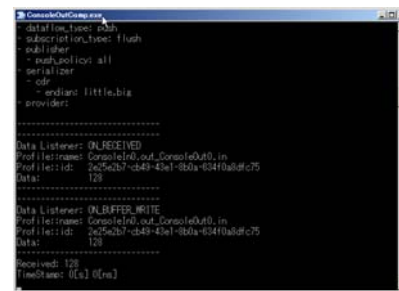


■ 動作確認

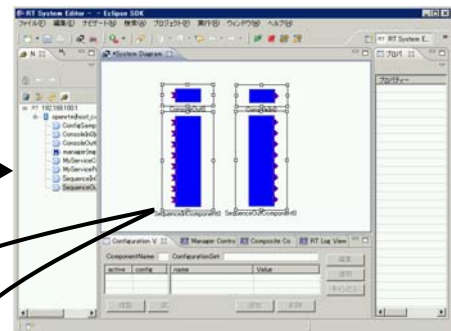
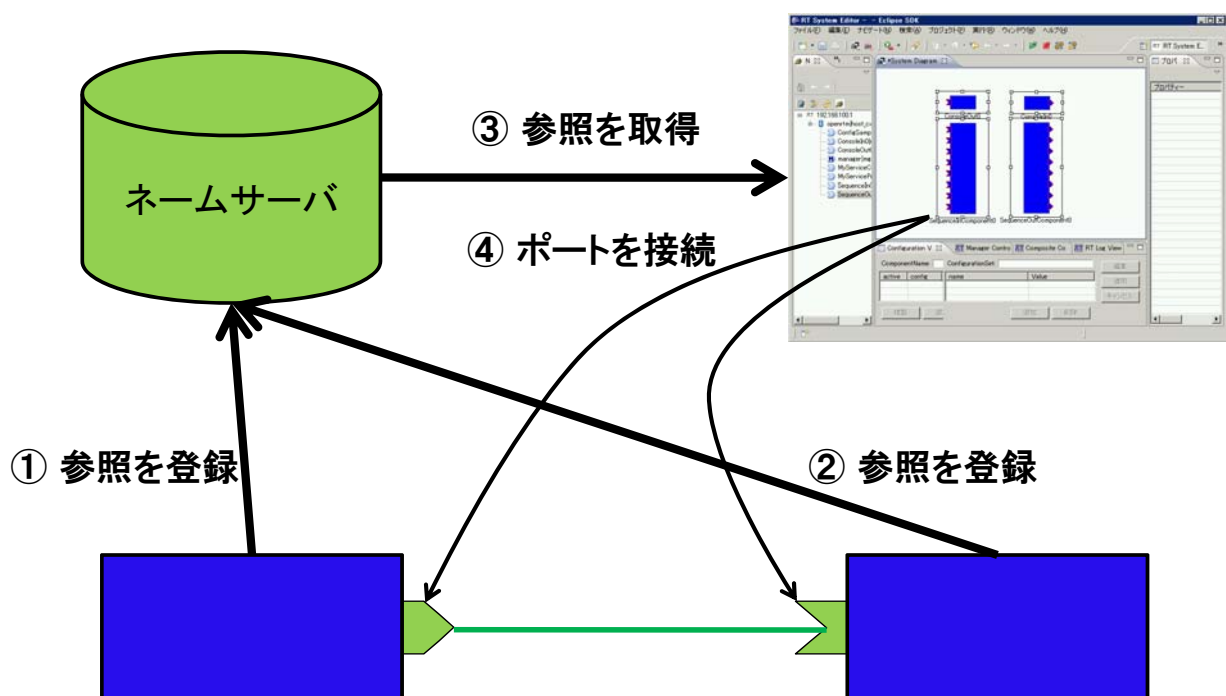
① ConsoleIn側で数字を入力



② ConsoleOut側が表示



RTコンポーネントの動作シーケンス



OpenRTM-aistの開発支援ツール



OpenRT Platform

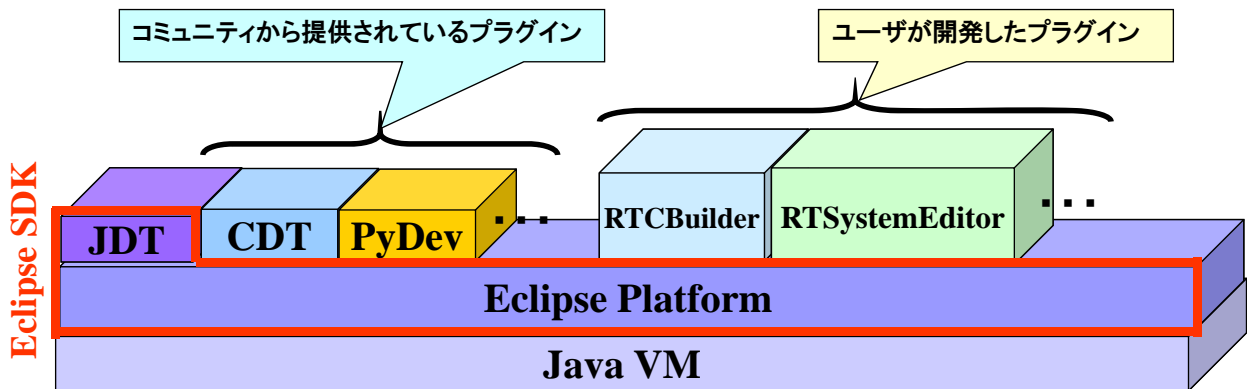


- **ロボット知能ソフトウェアプラットフォーム**
 - <http://www.openrtp.jp/wiki/>
 - システム設計, シミュレーション, 動作生成, シナリオ生成などをサポート
 - **OpenRT Platformツール群**
 - コンポーネント開発, システム開発における各開発フェーズの作業支援
 - 開発プラットフォームにEclipseを採用
 - **構成**
 - **RTCビルダ**
 - **RTCデバッグ**
 - **RTシステムエディタ**
 - **ロボット設計支援ツール**
 - **シミュレータ**
 - **動作設計ツール**
 - **シナリオ作成ツール**
- など



■ オープンソース・コミュニティで開発されている統合開発環境

- マルチプラットフォーム対応. WindowsやLinuxなど複数OS上で利用可能
- 「Plug-in」形式を採用しており, 新たなツールの追加, 機能のカスタマイズが可能
- RCP(Rich Client Platform)を利用することで, 簡単に単独アプリ化が可能



RTSystemEditorのインストール

■ ダウンロードし, 解凍するだけ

- <http://openrtm.org/pub/OpenRTM-aist/tools/1.1.0/>

※Javaの実行環境については, 別途インストールが必要

OpenRTM-aist
The power to connect

ホーム > ダウンロード > ツール > Eclipse tools 1.1.0-RC2

OpenRTM Eclipse tools 1.1.0-RC2
最終更新: kunitada 2011-07-22 18:36

これまで, OpenRTM-aistのツールとして開発されてきた RTCBuilder (旧RtcTemplate) および RTSystemEditor (旧 RtcLink) は, OpenRTP3やその他のツールと統合開発環境を構成する OpenRT Platform に組み込まれることになりました。こちらでは, RTSystemEditor および RTCBuilderのみを配布していますが, 将来的には様々なツールを一緒に提供する予定です。

現在の RTSystemEditor および RTCBuilderの最新バージョンは 1.1.0 です。

Table of contents

- 全部入りパッケージ
- バイナリ
- Eclipse SDKのインストール
- 過去のバージョン

全部入りパッケージ

| Eclipse-3.4.2 [Ganymede SR2] | | |
|------------------------------|--|------------|
| Eclipse3.4.2-RTSE-RTCB | eclipse342_rtmtools110-rc2_wb32_ja.zip | 2011.07.22 |
| Eclipse3.4.2-RTSE-RTCB | MD5:2e6f9fa3e370d6a7ac1f9340d36c7abf | |

Ubuntu8.04, Ubuntu9.10, Ubuntu10.04でLinux用Eclipse3.4.2が動作しない。平井君が報告されています。

- Ubuntu8.04では, apt-get install xulrunner-1.9 として xulrunnerをアップグレードしてください。
- Ubuntu9.10, Ubuntu10.04では, 以下の方法を利用するか, Eclipse3.3.18の23.5をご利用ください。

```
$ su
# vi /etc/apt/source.list
1行追加 - deb http://jp.archive.ubuntu.com/ubuntu/ jaunty main restricted
# apt-get update
# apt-get install xulrunner-1.9
# dpkg -i |grep xulrunner-1.9
<< xulrunner-1.9
```

システム構築支援ツール RTSystemEditorについて

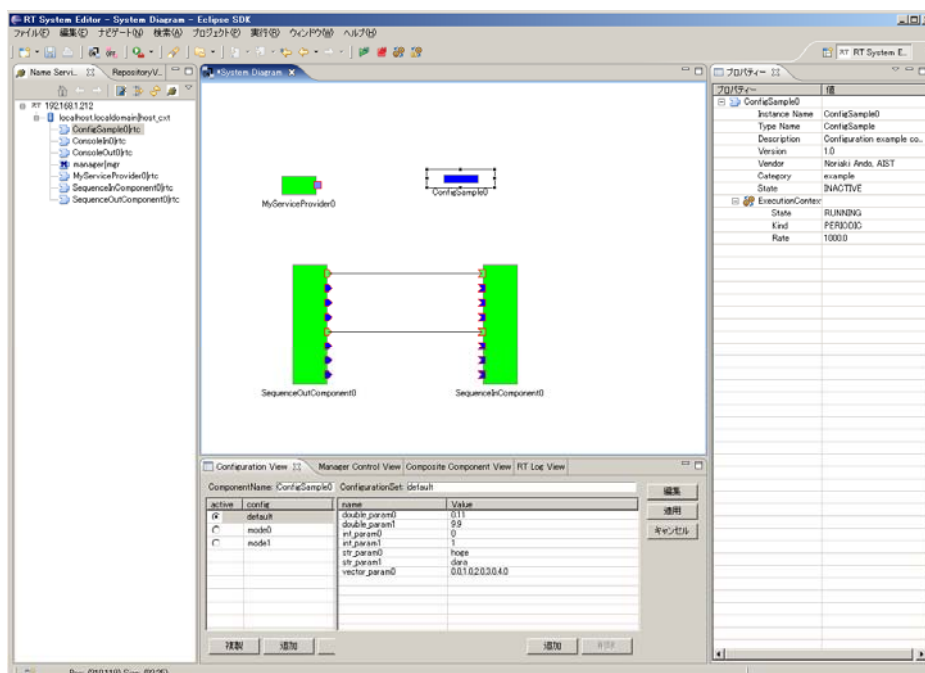


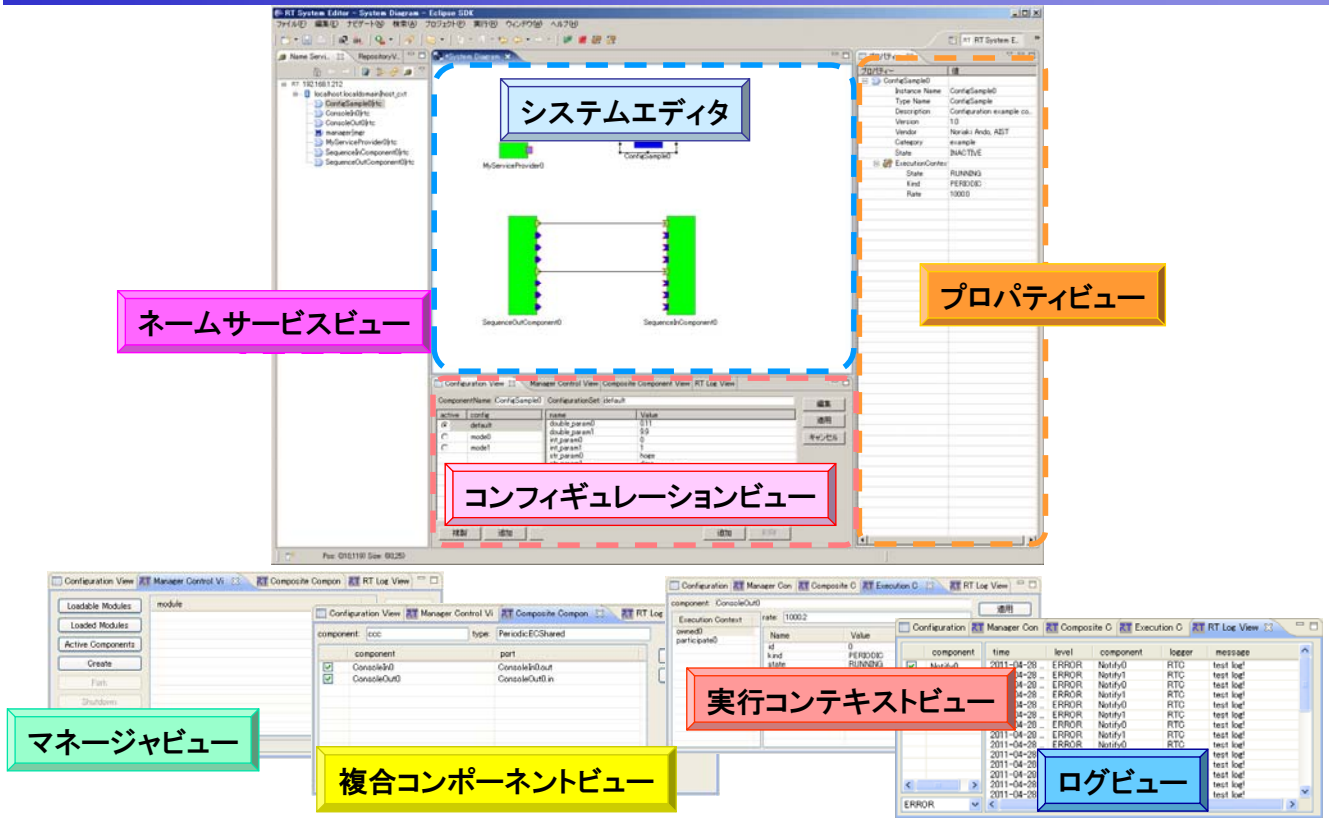
RTSystemEditor概要



■ RTSystemEditorとは？

- RTコンポーネントを組み合わせて、RTシステムを構築するためのツール



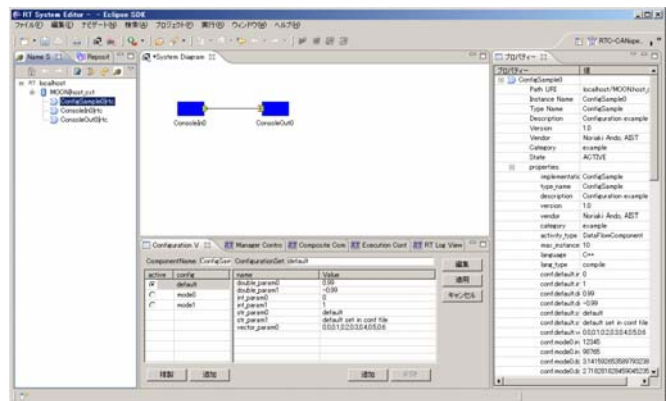


コンフィギュレーションコンポーネントの起動

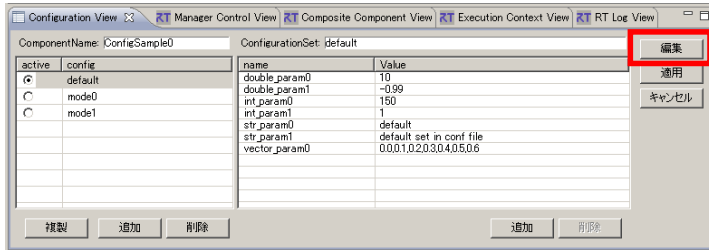
- **ConfigSampleCompの起動**
 - [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[examples]→ **[ConfigSampleComp.exe]**

```

ConfigSampleComp.exe
-----
Updating... ¥
-----
Active Configuration Set: default
-----
int_param0: 0
int_param1: 1
double_param0: 100
double_param1: -0.99
str_param0: model1
str_param1: default set in conf file
vector_param0[0]: 0
vector_param0[1]: 0.1
vector_param0[2]: 0.2
vector_param0[3]: 0.3
vector_param0[4]: 0.4
vector_param0[5]: 0.5
vector_param0[6]: 0.6
-----
Updating... |
e[LA
    
```



■ RTコンポーネントのコンフィギュレーション情報の確認/編集



※「編集」ボタンにより、各種コントロールを用いた一括編集が可能

※「Apply」チェックボックスがONの場合、設定値を変更すると即座にコンポーネントに反映
→テキストボックスからフォーカス外れる、ラジオボタンを選択する、スライダーを操作する、スピナを変更する、などのタイミング

※コンフィギュレーション情報を複数保持している場合、上部のタブで編集対象を切り替え

制約違反

即時反映

コンフィギュレーション情報の設定方法

● rtc.conf内

[カテゴリ名]. [コンポーネント名]. config_file: [コンフィギュレーションファイル名]

※例) example.ConfigSample.config_file: configsample.conf

● コンフィギュレーションファイル内

● コンフィギュレーション情報

conf. [コンフィグセット名]. [コンフィグパラメータ名] : [デフォルト値]

※例) conf.mode0.int_param0: 123

● Widget情報

conf. __widget__. [コンフィグパラメータ名] : [Widget名]

※例) conf.__widget__.str_param0: radio

● 制約情報

conf. __constraints__. [コンフィグパラメータ名] : [制約情報]

※例) conf.__constraints__.str_param0: (bar,foo,foo,dara)

conf. __[コンフィグセット名]. [コンフィグパラメータ名] : [制約情報]

※例) conf._mode1.str_param0: (bar2,foo2,dara2)

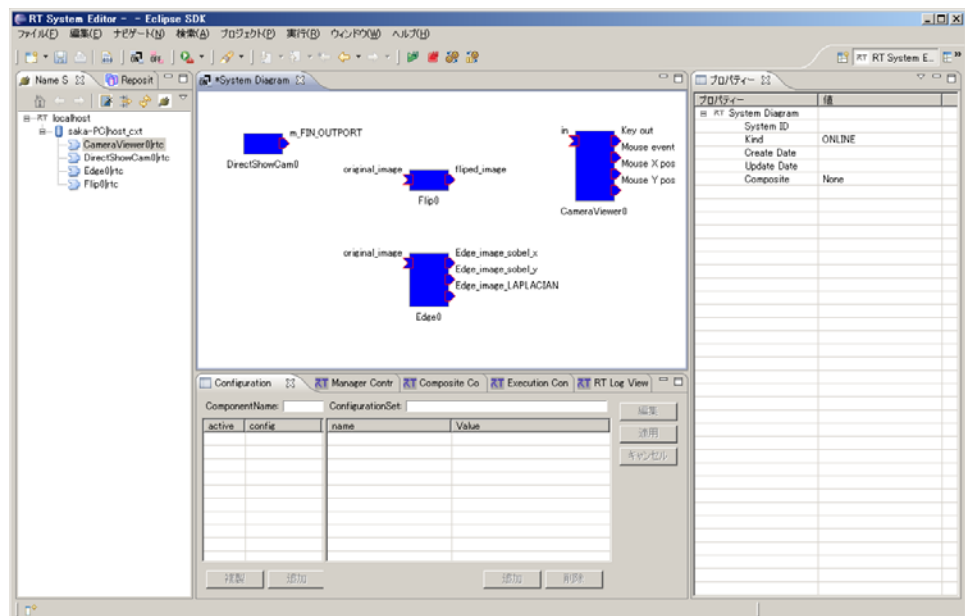
RTCの利用者が設定するのではなく、RTC開発者、RTC管理者が設定することを想定。

RTCBuilderを使用することで設定可能

- CameraViewerCompの起動
 - [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [CameraViewerComp.exe]
- DirectShowCamCompの起動
 - [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [DirectShowCamComp.exe]
- 画像処理用コンポーネントの起動
 - [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [FlipComp.exe]
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [EdgeComp.exe]

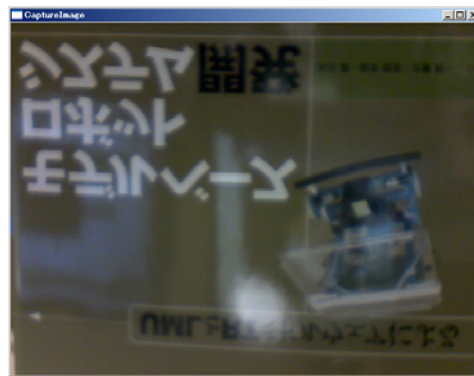
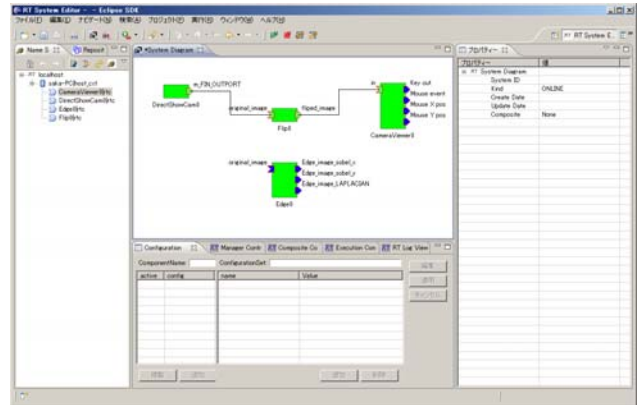
システムの構築

- 以下のコンポーネントをエディタ上に配置
 - DirectShowCam
 - Flip
 - Edge
 - CameraViewer

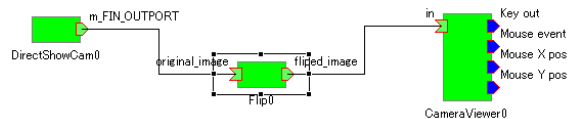


システムの構成

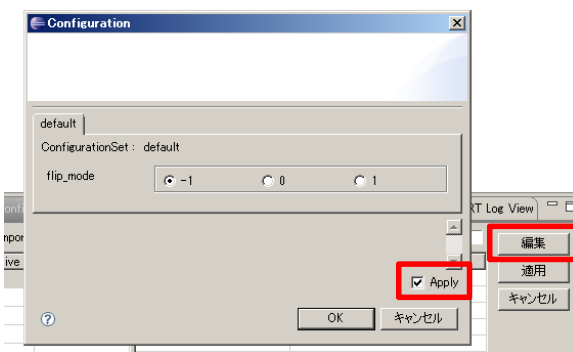
- Flip側との接続
 - DirectShowCam → Flip
 - CameraViewerと接続
(接続プロファイルはデフォルト設定)
 - AllActivateを実行



コンフィギュレーションの変更



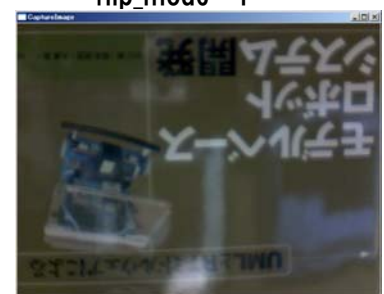
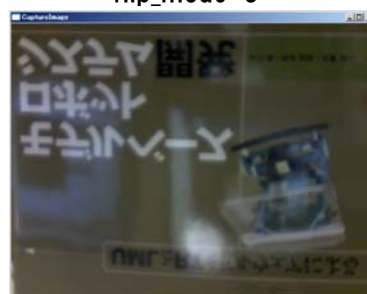
- ConfigurationViewの「編集」
- 表示されたダイアログ内で「flip_mode」の値を変更
- 「Apply」のチェックボックス



flip_mode=1

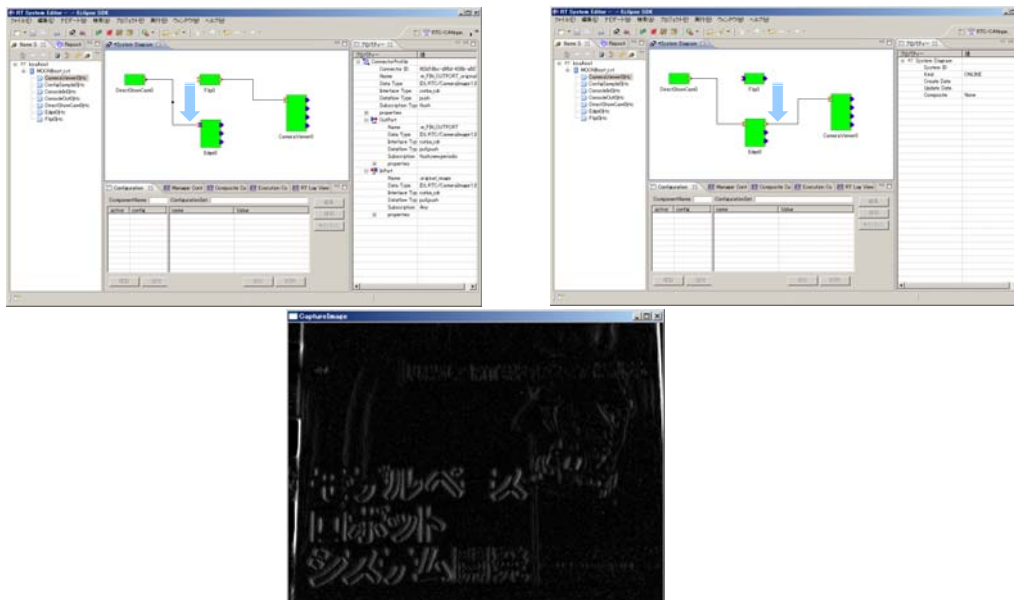
flip_mode=0

flip_mode=-1



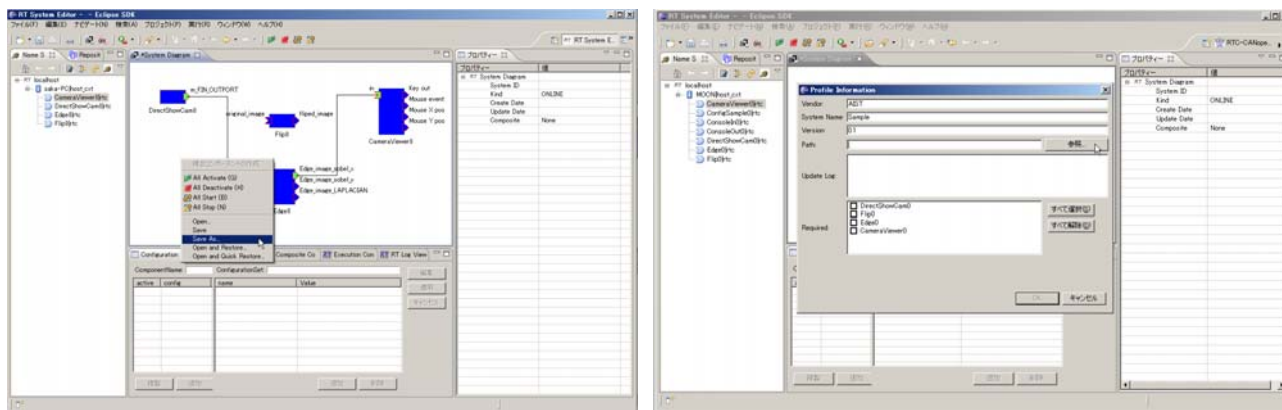
システム構成の変更

- Edge側への差し替え
 - Flipに繋がっている接続線を選択
 - Flip側のPort部分に表示されているハンドルをEdge側のPortに繋ぎ替え
 - 接続プロファイルはデフォルト設定のまま



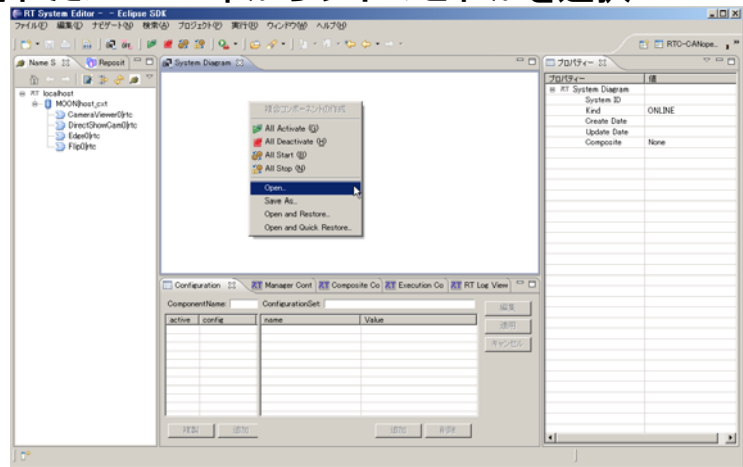
システム構成の保存・復元

- エディタ上で右クリック
- 表示されたメニュー内から「Save As...」を選択
- 【Profile Information】画面にて、ベンダ名、システム名、バージョン番号、保存先ファイル名を指定



※指定したファイルにXML形式(RtsProfile)で保存

- システムエディタを閉じる
- 各コンポーネントを一度終了し、再起動
- エディタ上で右クリックし、表示されたメニュー中から以下のどれかを選択
 - 「Open」
 - 「Open and Restore」
 - 「Open and Quick Restore」
- 【ファイル選択画面】にて、先ほど保存したファイルを指定



※Open: 使用していたRTCのみを読み込み

Open and Restore: 使用していたRTCを読み込むと同時に、接続、コンフィギュレーションセットの内容も復帰

Open and Quick Restore: 読み込み内容はOpen and Restoreと同様。該当RTCを検索する際にIORのみを使用

既存コンポーネントの再利用

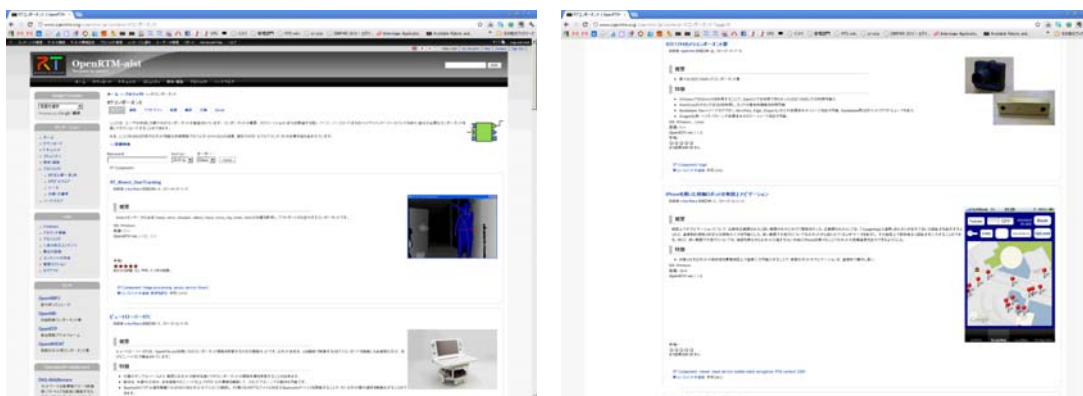
■ プロジェクトとは

- ユーザが作成した様々なコンポーネントやツールの公開場所
- ユーザ登録すれば、誰でも自分の成果物の紹介ページを作成可能
- 他のユーザに自分のコンポーネント等を紹介することができる

■ プロジェクトのカテゴリ

- RTコンポーネント: 1つのコンポーネントまたは複数のコンポーネント群などが登録されています。
- RTミドルウェア: OpenRTM-aistや他のミドルウェア、ミドルウェア拡張モジュール等が登録されています。
- ツール: 各種ツール(RTSystemEditorやrtshellを含む)ツールはこのカテゴリになります。
- 関連ドキュメント: 関連ドキュメントとは、各種インターフェースの仕様書やマニュアル等を含みます。

プロジェクトページ



| タイプ | 登録数 |
|------------|-----|
| RTコンポーネント群 | 287 |
| RTミドルウェア | 14 |
| ツール | 19 |
| 仕様・文書 | 4 |
| ハードウェア | 28 |

既存コンポーネントの再利用

■ プロジェクトから対象コンポーネントを取得

■ 「顔検出コンポーネント」

<http://www.openrtm.org/openrtm/ja/project/facedetect>

対象コンポーネントをダウンロード

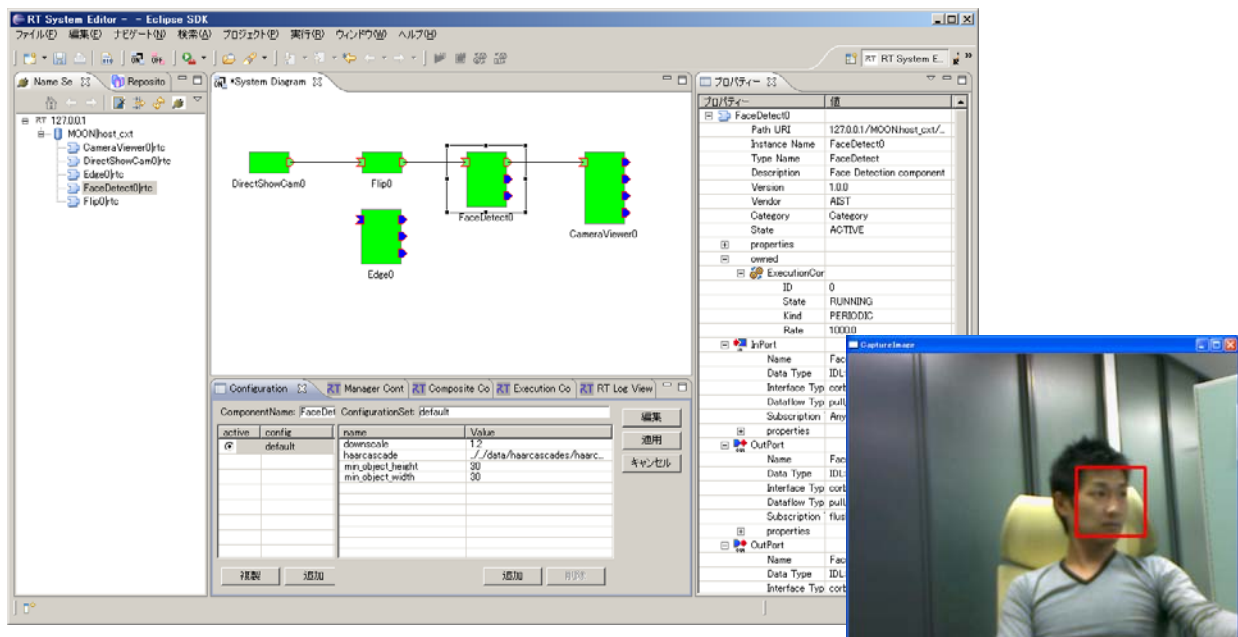


既存コンポーネントの再利用

■ ダウンロードしたファイル(FaceDetect.zip)を解凍

■ 解凍したディレクトリ内の以下のファイルを実行し、システムエディタ上に配置

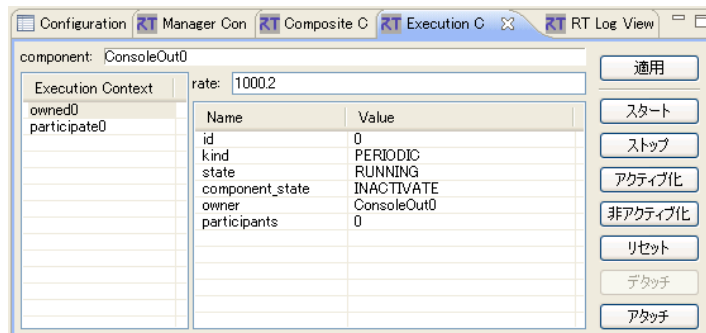
$\$(FaceDetect_Root)/build/Release/FaceDetectComp.exe$



- IPアドレスの確認
 - 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」
 - コマンド「ipconfig」を実行
- 他PC上で動作するRTCとの接続
 - 隣の方のIPアドレスを聞く
 - RTSystemEditorの「ネームサーバを追加(コンセントのアイコン)」をクリックして、上記のIPアドレスを入力する
 - 隣の方のネームサーバ内の階層化にあるDirectShowCamをシステムエディタにDnDする
 - 上記でDnDしたDirectShowCamと自分のPC上で起動したCameraViewerのデータポートを接続する

補足説明

■ RTコンポーネントが属する実行コンテキスト(EC)を一覧表示

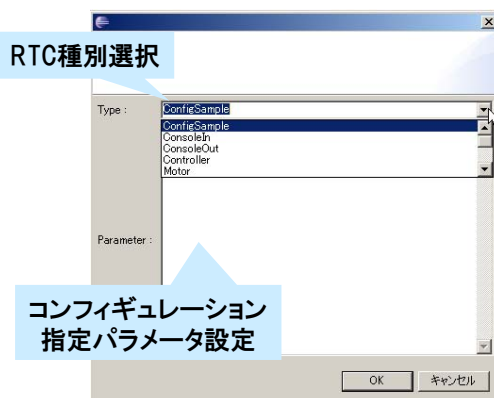
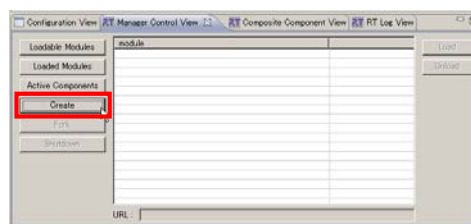


| 属性名 | 説明 |
|-----------------|--------------------------------------|
| id | ECのID. オンラインの場合には, context_handleを表示 |
| kind | ECの種別(PERIODIC/EVENT_DRIVEN/OTHER) |
| state | ECの状態(RUNNING/STOPPING) |
| component state | 対象RTCの状態(ACTIVE/INACTIVE/ERROR) |
| owner | 対象ECを所有しているオーナーRTCのインスタンス名 |
| participants | 対象ECに参加中のRTCの数 |

※対象ECの実行周期の変更, EC自身の動作開始/終了, 新規RTCへのアタッチ, アタッチ済みRTCのデタッチも可能

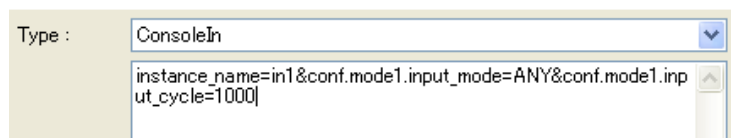
マネージャビュー

■ RTコンポーネントの新規インスタンスの生成

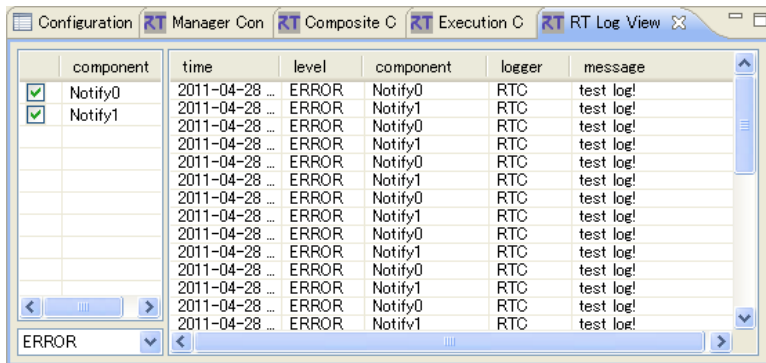


● コンフィギュレーション指定パラメータ

- conf. [ConfigSet名]. [Configパラメータ名]=[設定値]の形式にてConfigurationSetの値も設定可能



■ 選択したRTCから収集したログ情報を一覧表示

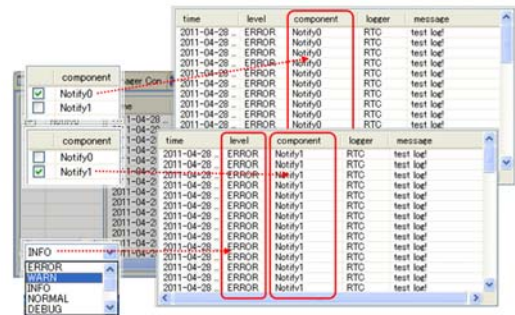


※近日機能追加予定

● ログ収集の開始/停止



● ログ情報のフィルタリング

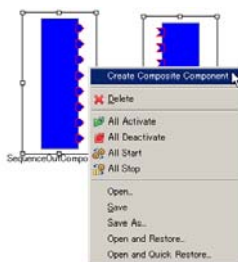


複合コンポーネント

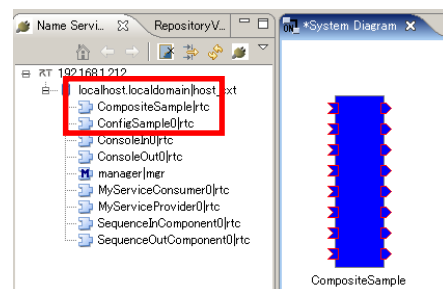
■ 複数のRTCをまとめて、1つのRTCとして扱うための仕組み

● 複合コンポーネントの作成方法

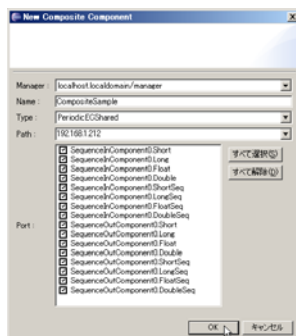
① 複数RTCを選択している状態で右クリック



③ 複合コンポーネントを生成



② 複合コンポーネントのプロパティを設定



| 項目 | 設定内容 |
|---------|------------------------|
| Manager | 複合コンポーネントを制御するマネージャを選択 |
| Name | 複合コンポーネントのインスタンス名を入力 |
| Type | 複合コンポーネントの型を選択 |
| Path | 複合コンポーネントのパスを入力 |
| Port | 外部に公開するポートを選択 |

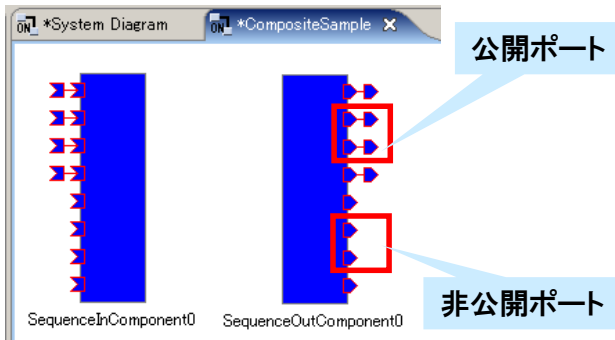
※生成対象複合コンポーネント外部と接続されているPortは強制的に公開されます

■ 複合コンポーネントのタイプについて

| タイプ名 | 説明 |
|---------------------|---|
| PeriodicECShared | 実行主体であるExecutionContextのみを共有. 各子コンポーネントはそれぞれの状態を持つ |
| PeriodicStateShared | 実行主体であるExecutionContextと状態を共有 |
| Grouping | 便宜的にツール上のみでグループ化 |

■ 複合コンポーネントエディタ

- 複合コンポーネントをダブルクリックすることで表示

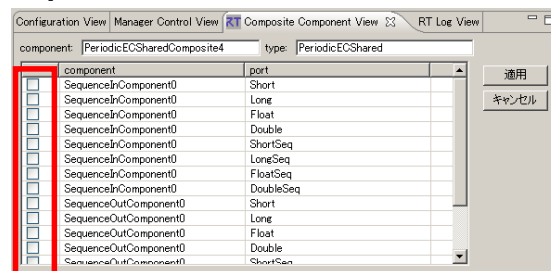


- ※エディタ内に別RTCをDnDすることで、子コンポーネントの追加が可能
→追加したRTCのポートは全て非公開に設定
- ※エディタ内のRTCを削除することで、子コンポーネントの削除が可能
→削除されたRTCは、親エディタに表示

■ 公開ポートの設定

- 複合コンポーネントビュー

ポート公開情報

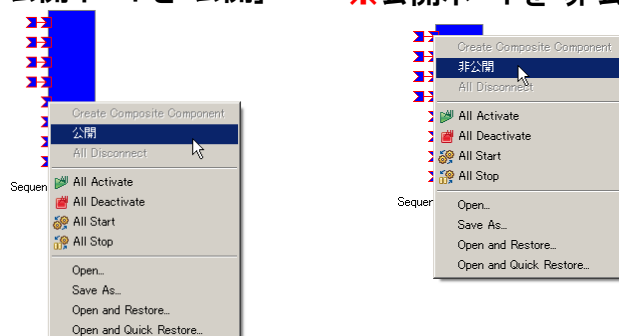


- ※ポート公開情報を変更し、「適用」をクリック

- 複合コンポーネントエディタ

※非公開ポートを「公開」

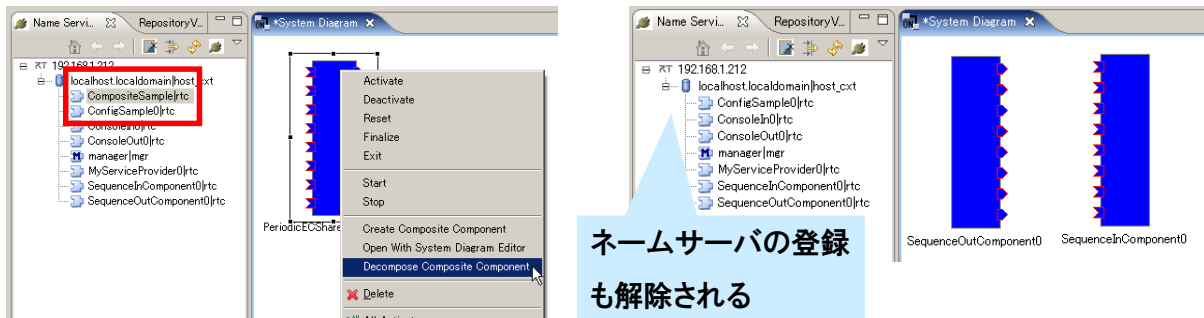
※公開ポートを「非公開」



外部コンポーネントと接続されているポートを「非公開」に設定することはできません

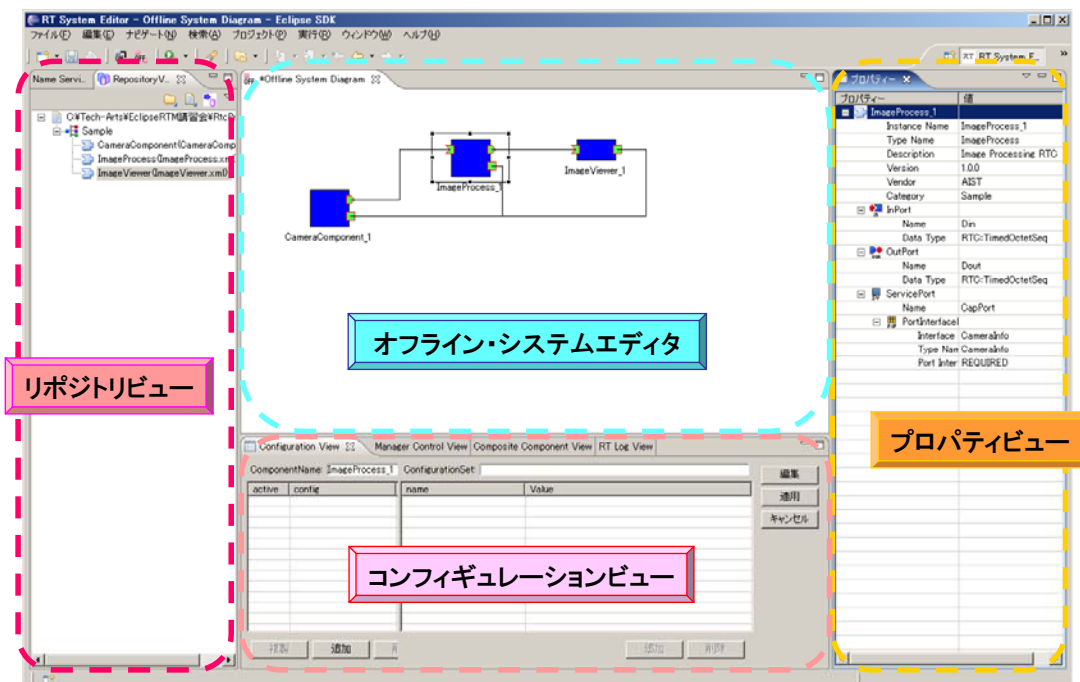
■ 複合コンポーネントの解除

- ① 複合RTCを右クリックし、複合コンポーネントの解除を選択
- ② 複合コンポーネントが分解され、内部のRTCが表示



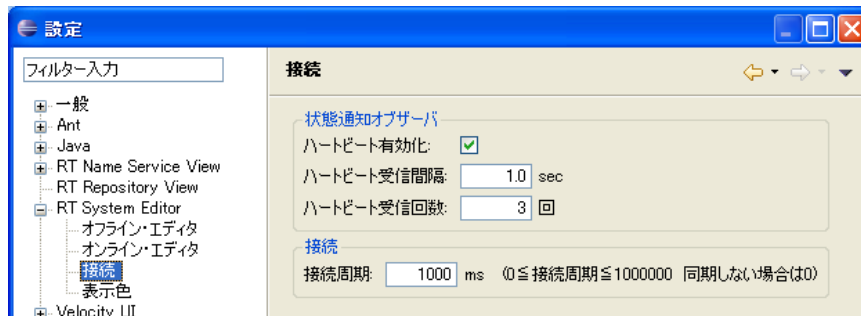
※エディタ上で、(Deleteキーなどで)単純に削除した場合は、エディタから表示が消えるのみ複合コンポーネントは解除されない

- RTコンポーネントの仕様を用いてRTシステムを構築
 - 実際のRTコンポーネントが動作している必要はない



■ 接続一状態通知オブザーバ

- RTCの生存確認用オブザーバに関する設定
 - RTSE側から生存確認を行うのではなく、RTC側から通知(ハートビート)を行う形
 - OpenRTM-aist-1.1以降で対応



- ハートビート有効化: ハートビートによる生存確認機能の有効化
- ハートビート受信間隔: ハートビートの受信間隔。この間隔以内にRTC側からハートビートが送られてこないと生存確認失敗と判断
- ハートビート受信回数: この回数を超えて生存確認に失敗した場合、対象RTCに異常が発生したと判断

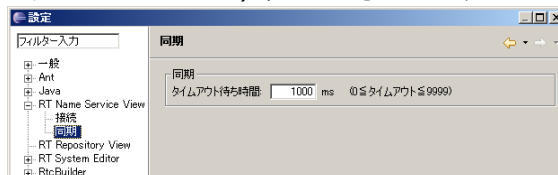
■ 「RT Name Service View」-「接続」【接続周期】

- ネームサービスビューが、ネームサーバに情報を問い合わせる周期



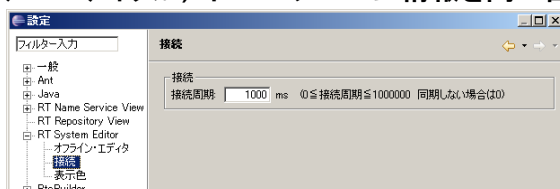
■ 「RT Name Service View」-「同期」【タイムアウト待ち時間】

- ネームサービスビューが、リモートオブジェクトのレスポンスを待つ時間



■ 「RT System Editor」-「接続」【接続周期】

- システムエディタが、ネームサーバに情報を問い合わせる周期



**【接続周期】をゼロに設定すると
ネームサーバとの同期を行わない**