

# 第二部: Flipコンポーネント作成 <sub>会津大学</sub>



University of Aizu



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



University of Aizu



# 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

ファイル(F) 編集(E) ソース(S	5) リファクタリング(T) ナヒ	ゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ	(H)	
📬 🕶 🖩 🖪 🛤 🕴 🕈	\$ • O • & • 😕 🔗	• 2 • 7 • • • • • •	RTC Buik	der 🐉
🛚 パッケージ・ 🛛 🦳	🗖 🅕 test 🛛			- 6
E 🕏	▲ 基本			- -
RTC.xml		sic 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:	RTE
	*モジュール名:	ModuleName		この
	モジュール概要:	ModuleDescription	モジュール概要:	RTI
	*バージョン:	1.0.0		ASC
	*ベンダ名:	VenderName	パージョン:	RT=
	*モジュールカテゴリ:	Category	▼ ベンダ名:	RTE
	コンポーネント型:	STATIC	•	ASC
	マクティビティ型・		モジュールカテゴリ:	: RTE 選択
	プラジュビジュ主:	PERIODIC		使用
	コンポーネント種類:	DataFlow     FSM     MultiMode	コンポーネント型:	RTI
	4 基本 アクティビティ デー		・環境 RTC.xml	+
	Real BuildView S?			- 6
	Bandvici a			
		ModuleName		
<b>=</b> \$				



# RTCBuilder起動

• Windows

スタートメニューから「OpenRTM-aist x.y」→「tools」の下の[OpenRTP 1.1.0]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[rtcws]と記入します。





・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。

🔚 CVS リポジトリー・エクスプローラー	
裁Java (デフォルト)	
<sup>と」</sup> Java の型階層	
🔊 Java 参照	
মা RT System Editor	
RTC Builder	
≝ <sup>0</sup> チーム同期化	
参 デバッグ	
◆プラグイン開発	
<u>ト</u> リソース	



## 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

RT-Component Builder Project

?

プロジェクト名:Flip	
	プロジェクト名(P):
	図 デフォルト・ロケーションの使用(D)
	ロケーション(L): C:¥rtcws

参照(R)..

キャンセル

完了(F)



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

*モジュール名:	ModuleName	
モジュール概要:	ModuleDescription	
*バージョン:	1.0.0	
*ベンダ名:	VenderName	
*モジュールカテゴリ:	Category	a R
コンポーネント型:	STATIC	
アクティビティ型:	PERIODIC	7
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:	1000.0	
既要:		
RTC Type :		<u></u>
コード生成とパッケー	-ジ化	
コードの牛成およびパッ	ッケージ化を行います。	
コード生成 パッケー	-ジ化	

RI-Component Basic Profile



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(	こ関するアクション
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	マクション
onActivated onError	onDeactivated onReset	onAborting
	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	7クション
onAction		
	Mode型コンポーネントのフ	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: onActivated	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。



### 選択アクティビティ

- onActivated
- onDeactivated
- onExecute

![](_page_9_Picture_0.jpeg)

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

*	D	et	а	il	

このセクションではデータボート毎の概要を説明するドキュメントを記述します。 上のデータボートを選択すると、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	
Documentation	
概要説明:	
データ型:	12
データ数:	
意味:	
単位:	6
発生頻度,周期:	
如理插度 周期 ·	
KETINGKI PHTU ·	

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)	Add
originalImage	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	1
変数名	originalImage	1
表示位置	LEFT	

・InPort ポート名: originalImage データ型: RTC::Cameralmage 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

![](_page_10_Picture_0.jpeg)

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		RT-Component Service Port Profile	
sv_name	Add Port	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。	
	Add Interface	*ポート名: sv_name	
	Delete	表示位置: LEFT	
		• Documentation	
		概要説明:	
			*
			*
		I/F概要說明:	_

### • 今回は使用しません。

![](_page_11_Picture_0.jpeg)

コンフィギレーション・プロファイル

RTコンポーネントのコンフィギレーションを設定

#### 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:		
Documentatio	on	
データ名:		

Documentation	
データ名:	
デフォルト値:	
ere	*
GA3237.97 ·	-
₩位:	
データ範囲:	
制约点升。	*
1717×171	

+ 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 データボート サービスボート 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

#### 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。

*名称	Add
flipMode	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	ーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー
変数名:	flipMode	デーン <u>デ</u> ー (mt)
単位:		
制約条件:	(-1, 0, 1)	
Widget:	radio	Widget: radio
Step:		

![](_page_12_Picture_0.jpeg)

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

rsion	OS		Add
			Dele
細情報			
OS Version		Add	Add
		Delete	Delete
その他OS情報			
の他CPU情報			
ライブラリ情報			
Namo	Version	Info.	Add
Nume			
Nume			Delete
Nume			Delete
Kunte			Delete

![](_page_13_Picture_0.jpeg)

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)	<ul> <li>・選択アクティビティ</li> <li>・ onActivated</li> <li>・ onDeactivated</li> <li>・ onExecute</li> </ul>	・InPort ポート名: originalImage データ型: RTC::CameraImage 変数名: originalImage 表示位置: left	•言語 C++
コンポーネント型: STATIC	名称: flipMode	・OutPort	
アクティビティ型: PERIODIC	データ型: int	ポート名: flippedImage	
コンポーネントの種類: DataFlow	デフォルト値: 0	データ型:	
最大インスタンス数: 1	変数名: flipMode	RTC::CameraImage	
実行型: PeriodicExecutionContext	制約条件: (-1, 0, 1)	変数名: flippedImage	
実行周期: 1000.0	Widget: radio	表示位置: right	

# 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 ①コード生成のボタンを押します。

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

![](_page_14_Picture_0.jpeg)

コンパイルの仕方

![](_page_14_Picture_2.jpeg)

![](_page_15_Picture_0.jpeg)

## 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 トリを指定。

ile Tools Options I	Help			
here is the source code:	C:/test/Flip			Browse Source
here to build the binaries:	C:/test/Flip/build			✓ Browse Buil
earch:			Grouped 🔲 Advanced	🔂 Add Entry
lame		Value		
	Press Configure to update and display	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display i	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display r	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display r Current Generator: None	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display i Current Generator: None	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display i	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display n	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display i	new values in red, then press G	enerate to generate selected buil	d files.
Configure Generate	Press Configure to update and display i	new values in red, then press G	enerate to generate selected buil	d files.

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。

![](_page_15_Picture_6.jpeg)

![](_page_16_Picture_0.jpeg)

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

CMake 3.2.1 - C:/test/Flip/build	
File Tools Options Help	
Where is the source code: C:/test/Flip	Browse Source
Where to build the binaries: C:/test/Flip/build	▼ Browse Build
Search:	Grouped Advanced 🔂 Add Entry
Name	Value
BUILD_DOCUMENTATION	
BUILD_IDL	
BUILD_SOURCES	en e
CMAKE_CONFIGURATION_TYPES	Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_INSTALL_PREFIX	C:/Program Files/Flip
OpenRTM_DIR	C:/Program Files/OpenRTM-aist/1.1/cmake
PROJECT_VERSION	1.0.0
STATIC_LIBS	
Press Configure to update and display n	ew values in red, then press Generate to generate selected build files.
Configure Generate Current Generator: Visual Studio 12 2013	
OPENRIM_VERSION_MINOR=1	
OPENRIM_VERSION_PATCH=1	WERGTON A. D. OPT. OPTIME NO. DEDPENDER. DEVICTION
OPENRIM_CFLAGS=-D_WIN32_;-D_X66_;-D_N1_;-D_03 OPENRIM_INCLUDE_DIRS=C:/Program_Files/OpenRIM-aist/1	<pre>.version_=4;-D_CRI_SECORE_NO_DEFRECALE;-DINCLODE_SEUD_IN_RE_GII;-DREC_CORE .1:C:/Program_Files/OpenRTM-aist/1.1/rtm/idl</pre>
OPENRIM LDFLAGS=	, o,, 110g1um 1110, opennin u100, 111, 100, 101
OPENRTM_LIBRARY_DIRS=C:/Program_Files/OpenRTM-aist/1 OPENRTM_LIBRARIES=optimized;RTC111_vc12;optimized;co OPENRTM_DILC=ompiid1	1/bin;C:/Program Files/OpenRTM-aist/1.1/omniORB/4.1.7_vc12/lib/x86_win32 ill11_vc12;optimized;omniORB417_rt;optimized;omniDynamic417_rt;optimized;c
OPENRTM IDLFLAGS=-bcxx;-Wba;-nf;-IC:/Program Files/O	DpenRTM-aist/1.1/rtm/idl
OPENRIM_IDL_WRAPPER=rtm-skelwrapper.py	
OPENRTM_IDL_WRAPPER_FLAGS=include-dir="";skel-su	(ffix=Skel;stub-suffix=Stub
OpenKIM configuration Found Found Devugent C: (Dregner Files (devugen (bis (devugen e	E
Configuring done	(c (round Actaron 1.0.2.1 )
Generating done	*
	· · · · · · · · · · · · · · · · · · ·

### 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てたら完了

![](_page_16_Picture_4.jpeg)

![](_page_16_Picture_5.jpeg)

![](_page_16_Picture_6.jpeg)

- Linux版 下記コマンドを打ち込む
- \$ cd rtcws/Flip
- \$ mkdir build
- \$ cd build
- \$ cmake ..

![](_page_17_Picture_0.jpeg)

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 ②ソリューションを開き、[Flip.h]と[Flip.cpp]を開き、編集 ションファイルを開きます。 をしてください。

![](_page_17_Picture_3.jpeg)

Microsoft Visual Studio Solution Version: Visual Studio 2013

	PACKAGE.vcxproj
of	VC++ Project
	10.9 KB

![](_page_17_Picture_6.jpeg)

10.5 KB PACKAGE.vcxproj.filters VC++ Project Filters File

INSTALL.vcxproj

VC++ Project

509 バイト

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

![](_page_17_Picture_9.jpeg)

![](_page_18_Picture_0.jpeg)

## 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。

増 パッケージ・エクスプロ 🛿 🔷 🗖 🗖	Flip 📄 CMakeLists.txt 🕱		- 6
□ 🕏 ▽	<pre>set(comp_srcs Flip.cpp ) set(ctandalone spcs Flip(own cpp)</pre>		
<ul> <li>Flip</li> <li>build</li> <li>cmake</li> <li>doc</li> <li>doc</li> <li>include</li> <li>src</li> <li>CMakeLists.txt</li> <li>Flip.cpp</li> <li>FlipComp.cpp</li> <li>CMakeLists.txt</li> <li>COPYING</li> <li>COPYING</li> <li>COPYING.LESSER</li> <li>Flip.conf</li> <li>README.Flip</li> <li>rtc.conf</li> <li>RTC.xml</li> <li>RTC.xml20151209151939</li> </ul>	<pre>set(standalone_srts FlipComp.cpp)  if (DEFINED OPENRTM_INCLUDE_DIRS) string(REGEX REPLACE "-I" ";"     OPENRTM_INCLUDE_DIRS "\${OPENRTM_INCLUDE_DIRS     string(REGEX REPLACE "; ";"     OPENRTM_INCLUDE_DIRS "\${OPENRTM_INCLUDE_DIRS}  if (DEFINED OPENRTM_LIBRARY_DIRS) string(REGEX REPLACE "-L" ";"     OPENRTM_LIBRARY_DIRS "\${OPENRTM_LIBRARY_DIRS     string(REGEX REPLACE "; ";"     OPENRTM_LIBRARY_DIRS "\${OPENRTM_LIBRARY_DIRS}  if (DEFINED OPENRTM_LIBRARY_DIRS)  if (DEFINED OPENRTM_LIBRARY_DIRS)  if (DEFINED OPENRTM_LIBRARY_DIRS)  if (DEFINED OPENRTM_LIBRARIES)  string(REGEX REPLACE "; ";"     OPENRTM_LIBRARIES "\${OPENRTM_LIBRARIES}")  string(REGEX REPLACE "; ";"     OPENRTM_LIBRARIES "\${OPENRTM_LIBRARIES}")  endif (DEFINED OPENRTM_LIBRARIES)  include_directories(\${PROJECT_SOURCE_DIR}/inclu include_directories(\${PROJECT_BINARY_DIR})  include_directories(\${PROJECT_BINARY_DIR}/idl) include_directories(\${PROJECT_BINARY_DIR}/idl) </pre>	RS}") RS}") RS}") RS}") ude) ude/\${PROJECT_NAME})	
	· 说明	リソース	パス
	4		

![](_page_19_Picture_0.jpeg)

# ビルドの実行

![](_page_19_Picture_2.jpeg)

### 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。

![](_page_19_Picture_4.jpeg)

Linux版
 下記コマンドを打ち込む

\$ cd rtcws/Flip/build
\$ make

![](_page_20_Picture_0.jpeg)

## RTSystemEditorについて

![](_page_20_Picture_2.jpeg)

![](_page_21_Picture_0.jpeg)

# 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

![](_page_21_Picture_3.jpeg)

![](_page_22_Picture_0.jpeg)

# RTSystemEditor起動

## パースペクティブの切り替え

![](_page_22_Picture_3.jpeg)

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

晶CVS リポジトリー・エクスプローラー	-
🐉 Java (デフォルト)	
<sup>2] Java</sup> の型階層	
影 Java 参照	
RT System Editor	
RTC Builder	
≦ <sup>0</sup> チーム同期化	
参デバッグ	
◆ プラグイン開発	
<b>≧</b> リソース	

![](_page_23_Picture_0.jpeg)

# 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 1.1]→ [tools]→[Start Naming Service]
- CameraViewerCompの起動

[スタート]メニューから起動

• OpenCVCameraCompの起動

[スタート]メニューから起動

![](_page_24_Picture_0.jpeg)

ネームサービスへ接続

• 接続方法

#### Name Service Vie 않 ⑦ RepositoryView □ ☆ ☆ ☆ ■ ▼ ①Name Service View のコンセントのア イコンをクリック

ネームサーバのアドレス	を入力してください。	
[	•	(Address:Port)
ОК	<b>+</b> 72	

## 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

### ③Name Service Viewに記入したものが記 載されれば完了

![](_page_24_Picture_7.jpeg)

![](_page_25_Picture_0.jpeg)

コンポーネント配置

![](_page_25_Figure_2.jpeg)

※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。

 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

![](_page_26_Picture_0.jpeg)

データポート接続

データポートを接続

![](_page_26_Figure_3.jpeg)

### 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

![](_page_26_Figure_5.jpeg)

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

ConnectorProfileを入	力してください。	
Name :	ppenCVCamera0.out_Flip0.original_image	
Data Type :	IDL:RTC/CameraImage:1.0	•
Interface Type :	corba_cdr	Ŧ
Dataflow Type :	push	•
Subscription Type :	flush	•
Push Rate(Hz) :		
Push Policy :	all	v
Skip Count :		
🕅 詳細		
0		

![](_page_27_Picture_0.jpeg)

コンポーネントアクティブ

## コンポーネントの起動方法

![](_page_27_Figure_3.jpeg)

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。

![](_page_27_Figure_5.jpeg)

![](_page_28_Picture_0.jpeg)

# 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。

RT System Editor RCP	×
File Window Help	
🌶 Name Ser 👘 Repositor 🖓 🗖 🗑 🔹 *System Diagram 🕱 👘 🖓	
	プロパティー
ا ته localhost	5 Flip0
0 NB1503013 host_cxt	Path URI
Di CameraViewer0 rtc	Instance Nam
D Flip0 rtc	Type Name
DenCVCamera0 rtc	Description
	Version
OpenCVCamera0 Flip0	Vendor
	outegory
default ConfigurationSet : default	
flip_mode	
	· ·
	Apply
OK	+7761

![](_page_29_Picture_0.jpeg)

# RTSystemEditor補足説明

![](_page_29_Picture_2.jpeg)

University of Aizu

![](_page_30_Picture_0.jpeg)

## RTC-Library-FUKUSHIMA

• RTC-Library-FUKUSHIMA

OpenRTM-aistを利用した、ロボット・テクノロジー・コンポーネント(RTC)ライブラリ サイト

- 機能
  - イベント情報
  - コンポーネントの登録・検索・ダウンロード

![](_page_30_Picture_7.jpeg)

![](_page_31_Picture_0.jpeg)

# RTC-Library-FUKUSHIMAへのアップロードの 仕方

### 1) RTC-Library-FUKUSHIMAへアクセス

http://rtc-fukushima.jp/

2)ログイン

![](_page_31_Picture_5.jpeg)

![](_page_32_Picture_0.jpeg)

## 3)コンポーネント登録画面 画面上部の「ライブラリー」をクリック、 その後ライブラリの画面下部の「コンポーネント登録」クリック

![](_page_32_Picture_2.jpeg)

![](_page_33_Picture_0.jpeg)

登録手順

1) RTC.xmlファイル読み込み 本日作成したFlipコンポーネント内のRTC.xmlを指定します。 RTCBuilderで入力した内容が登録されます。

### 2) コンポーネント登録情報入力

コンポーネントの登録情報を入力します。 RTC.xmlファイルを読み込んだ場合、概要、問い合わせ先には RTCBuilderの内容が挿入されます。

3) 登録内容確認 「確認する」ボタンを押し、登録内容を確認しください。 コンポーネントの登録情報
コンポーネント名:Flip
概要:Flip component
カテゴリ: カメラ
タグ: C++、OpenCV、画像処理
ファイルアップロード: Flipの中のbuild以下を削除または退 避させzipで圧縮。参照でzipを指定。
同意する:チェック
私はロボットではありません:チェック

![](_page_34_Picture_0.jpeg)

# 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

![](_page_35_Picture_0.jpeg)

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)