講習会テキスト第2部 Windows 版

目次

1.	1	まじめに
1	•	OpenCV とは2
2	2.	作成する RT コンポーネント2
2.	C	vFlip 関数の RT コンポーネント化2
1	•	cvFlip 関数について2
2	2.	コンポーネントの仕様2
e.	3.	Flip コンポーネントの雛型の生成
4	l.	ヘッダ、ソースの編集15
5	5.	CMake によるビルドに必要なファイルの生成17
6	3.	VC++によるビルド18
7	7.	Flip コンポーネントの動作確認18
8	3.	コンポーネントの接続19
3.	F	TC-Library-FUKUSHIMA
1	•	RTC-Library-FUKUSHIMA について
2	2.	コンポーネントをアップロード
4.	F	'lip コンポーネントの全ソース
1	•	Flip コンポーネントソースファイル (Flip.cpp)
2	2.	Flip コンポーネントのヘッダファイル (Flip.h)
ę	3.	Flip コンポーネントの全ソースコード

この講習会テキストは下記ページを参考にしています。 ・チュートリアル(画像処理コンポーネントの作成 Windows 編)

http://www.openrtm.org/openrtm/ja/node/5022 (2016/1/8 アクセス)

※ 文中の「x.y」や「x.y.z」の表記は使用環境の OpenRTM-aist のバージョンに読み替えてください。

1. はじめに

1. OpenCV とは

画像処理・画像解析および機械学習等の機能を持つ C/C++、Java、Python、MATLAB 用ライ ブラリです。

2. 作成する RT コンポーネント

Flip コンポーネント: OpenCV ライブラリが提供する様々な画像処理関数のうち、cvFlip() 関数を用いて画像の反転を行う RT コンポーネント

2. cvFlip 関数の RT コンポーネント化

OpenCVの cvFlip 関数を使用して、入力された画像を左右または上下に反転して出力するコンポー ネントを作成します。 作成手順としては 1)コンポーネントの仕様を決定 2)RTCBuilder を用いたソースコードのひな形の作成 3)アクティビティ処理の実装 4)コンポーネントの動作確認 になります。

1. cvFlip 関数について

cvFlip 関数は、OpenCV で標準的に用いられている関数です。入力された画像データを反転さ せて出力する機能があります。反転させる軸は垂直軸、水平軸、両軸と三種類あり引数で設定す ることが出来ます。

2. コンポーネントの仕様

これから作成するコンポーネントを Flip コンポーネントという名称にします。

このコンポーネントの動作としては画像データを入力ポート(InPort)から受け取り反転処理した画像データを出力ポート(OutPort)へ出力します。

それぞれのポートの名前を入力ポート(InPort)名:originalImage,出力ポート(OutPort) 名:flippedImage とします。

これらのコンポーネントのデータポートは画像の入出力に Camera Image 型を使用しています。

また、画像を反転させる方向は、左右反転、上下反転、上下左右反転の3通りが有ります。これ を実行時に指定できるように、RT コンポーネントのコンフィギュレーション機能を使用して指 定できるようにします。パラメータ名は flipMode という名前にします。

flipMode は cvFlip 関数の仕様に合わせて、型は int 型とし上下反転、左右反転、上下左右反転

それぞれに 0,1,-1 を割り当てることにします。



以上から Flip コンポーネントの仕様をまとめると下記の様になります。

コンポーネント名称	Flip
InPort	
ポート名	originalImage
型	CameraImage
意味	入力画像
OutPort	
ポート名	flippedImage
型	CameraImage
意味	反転された画像
Configuration	
パラメータ名	flipMode
型	int
意味	反転モード
	上下反転:0
	左右反転:1
	上下左右反転:-1

3. Flip コンポーネントの雛型の生成

Flip コンポーネントの雛型の生成方法を説明します。

1. RTCBuilder の起動

OpenRTP を起動させると作成物を保存するディレクトリを指定します。ここでは C 直下

の下記ディレクトリに保存します。

[C:¥rtcws]

最初に起動したとき下記画面がでます。この画面は使用しないので左上の×ボタンを押します。



🖨 Java - Eclipse SDK	
ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィン	ンドウ(W) ヘルプ(H)
【 → □ ◎ ◎ 棽 - ◎ - ◎ - ▼ ▼ ◎ - ◎ - ◎ - ◎ - ◎ - ◎ - ◎ - ◎ - ◎ - ◎	🔡 👪 Java
	□ □ BE ア 表示。
【問題 ☆ @ Javadoc Q 宣言	~
0項目	
۲. III. III.	•

×ボタンを押すと下記画面が表示されます。右上の[その他]をクリックしてください。

下記ウィンドウが出ますので[RTC Builder]を選択します。

パースペクティブを開く	
Good CVS リポジトリー・エクスプローラー Sylava (デフォルト) Java の型階層 Java 参照 Art RT System Editor RTC Builder F・ム同期化 なっていたり、問題	
◆ ブラグイン開発 ¹ → リソース	
ОК ‡	ャンセル

「RTC Builder」を選択することで、RTCBuilder が起動します。メニューバーに「カナヅチ とRT」のRTCBuilderのアイコンが現れれば完了です。

2. 新規プロジェクトの作成

画面上部のメニューから[ファイル]-[新規]-[プロジェクト]を選択

€ F	TC Builder - Eclipse SDK							
77	マイル(F) 編集(E) ソース(S)	リファクタリング(T)	ナ	ビゲート(N) 検索(A)	プロジェクト(P) 実行	テ(R) ウィンドウ(W) ヘルプ(H)		
	新規(N)	Alt+Shift+N ►	2	プロジェクト(R)	-			😭 🐊 RTC Builder 🐉 »
	ファイルを開く(.)		1	サンプル(X)				
	閉じる(C)	Ctrl+W	F9	その他(0)	Ctrl+N			
	すべて閉じる(L)	Ctrl+Shift+W	Г					
	保管(S)	Ctrl+S						
<u>.</u>	別名保管(A)							
Q	すべて保管(E)	Ctrl+Shift+S						
	前回保管した状態に戻す(T)							
	移動(V)							
	名前変更(M)	F2						
8	リフレッシュ(F)	F5						
	行区切り文子の変換(V)	,						
合	印刷(P)	Ctrl+P						
	ワークスペースの切り替え(W)	•						
	再始動							
2	インポート(I)							
4	エクスポート(0)							
RT	Open New Builder Editor							
	プロバティー(R)	Alt+Enter	F					- 0
	終了(X)		Г					
-								
i e	\$	II			1			
1			_		1		_	

ウィザードを選択					
ウィザー <mark>ド</mark> (W):					
フィルター入力					
は での で で で で で で で で で で で で で	ロジェクト ドファイルからの Java フ ng Framework	ロジェクト			E
٢		< 豆ろ(円)	次へ(N) >	完了(F)	±+7 ⁵ /71

「新規プロジェクト」画面において、「その他」 – 「RTC Builder」を選択し、「次へ」をクリッ

「プロジェクト名」欄に作成するプロジェクト名 (ここでは Flip) を入力して「完了」をクリ ックします。

RT-Component Builder Project				
				6
プロジェクト名(P): Flip				
ロケーション(L): C:¥rtcws¥Flip				参照(R)
?	< 戻る(B)	次へ(N) >	完了(F)	キャンセル

下記画面の様にパッケージエクスプローラ内にプロジェクトが追加されれば完了です。



3. RTC プロファイルエディタの起動 基本的に RTC vml が生成された時点で このプロジョ

基本的に RTC.xml が生成された時点で、このプロジェクトに関連付けられているワー クスペースとして RTCBuilder のエディタが開くはずです。 もし開かない場合は、「カナヅチと RT」の RTCBuilder のアイコンを押下するか、メ

ニューバーの [ファイル]-[Open New Builder Editor] を選択します。



 プロファイル情報入力とコードの生成 一番左の「基本」タブを選択し、基本情報を設定します。コンポーネントの名前や概要 などを記入します。ラベルが赤文字の項目は必須項目です。その他はデフォルトのまま で大丈夫です。

モジュール名:Flip モジュール概要:Flip component バージョン:1.0.0 ベンダ名:Aizu 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:DataFlowComponent 最大インスタンス数:1 実行型:PeriodicExecutionContext 実行周期:1000.0

RTC Builder - Flip/RTC.xml - Eclipse SDK			- O X
ファイル(E) 編集(E) ナビゲート(N) 検索(A) プロ	Jジェクト(₽) 実行(R) ウィンドウ(⊻) ヘルプ(出)		
📑 🕶 🔛 🔛 🏄 💁 🖌 🛷 🕶	堕・凹・骨 キャー	T RT RT	C Builder 🐉 🔌
■ パッケ 🙁 🖳 🗖 🗲 *Flip 🛛			- 0
● ◆ ▼ 基本			<u> </u>
RTC.xml	asic Profile		
このセクションではR	「コンポーネントの基本情報を指定します。	モジュール名:	RTコンポー E
*モジュール名:	Flip		この名称は: 使用できる:
モジュール概要:	Flip component	モジュール概要:	RTコンポー
*バージョン:	1.0.0	11-33-24	ASCII文字だ RTコンポー
*ベンダ名:	Aizu	// /=/.	x.y.z(x,y,z)
*モジュールカデゴリ	: Category 🗸	ベンダ名:	RTコンポー
コンポーネント型:	STATIC	モジュールカテゴリ:	RTコンポー
アクティビティ型:	PERIODIC		選択肢にない 使用できる?
コンポーネント種類:	DataFlow FSM MultiMode	コンポーネント型:	RTコンポー ・STATIC:
		DIC	•
	ーダハート リービスハート コノフィキュレーション トキュメント生成 言語・環境	RTC.XMI	
Buildview SS			
		_	
	Flip	<u></u>	

次に、「アクティビティ」タブを選択し、使用するアクションコールバックを指定しま す。

Flip コンポーネントでは、onActivated(),onDeactivated(),onExecute()コールバックを 使用します。下図のように赤枠の onAtivated をクリック後に赤枠のラジオボタンにて "on"にチェックを入れます。onDeactivated,onExecute についても同様の手順を行いま す。

RTC Builder - Flip/RTC.x	kml - Eclipse SDK					- 0 X
ファイル(E) 編集(E) ナビ	<u>ビゲート(N)</u> 検索(<u>A</u>)フ	/ロジェクト(<u>P</u>) 実行(<u>R</u>) ウィ	ンドウ(型) ヘルプ(日)			
📑 - 🛛 🕲 🖄 🏄	Q 🛷 -	8 · 9 · • • • • •	*		T RT	C Builder 🐉 👋
第 パッケ 🖾 🗖 🗖	≯ *Flip ⊗					- 0
□ 🙀 ▽		alive状態でのコン	ンポーネントアクション		onAborting	ERROR状態。
🔺 🗁 Flip	onActivated	onDeactivate	ed or	Aborting	onError	ERROR状態
RTC.xml	onError	onReset			onReset	ERROR状態:
		Dataflow型コン	ポーネントのアクション		onExecute	アクティフキ
	onExecute	onStateUpda	ate or	nRateChanged	onStateUpdate	onExecute0
		FSM型コンポ	-ネントのマカミョン		onAction	EXecution C
	onAction	1 JULE - JAN	「イノロックション		onModeChanged	モードが変動
	Charles of the second	Mode型コンポ	ーネントのアクション			=
	antita do Changod		1210000000		動作概要:	アクティビラ
	onmodechangea				事前条件:	アクティビラ
	Documentation				事後条件:	アクティビラ
	このセクションでは 上段のアクションを	各アクションの概要を説明する 選択すると、それぞれのドキュ:	ドキユメントを記述します メントを記述できます。	Γ.		
	アクティビティ名:	onActivated		ON OFF		-
	1					*
	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 RT	C.xml	
	BuildView 🛙					- 0
						*
						E
			Flip			
	•					•
□ ◆						

最終的に下図の様になります。

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					
a	ive状態でのコンポーネントアクショ	>				
onActivated	onDeactivated	onAborting				
onError	onReset					
D	ataflow型コンポーネントのアクショ	>				
onExecute	onStateUpdate	onRateChanged	1			
	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: onExecute			I ON OFF			

「データポート」タブを選択し、データポートの情報を入力します。 先ほど決めた仕様を元に以下のように入力します。[Add]ボタンを押して新しいデータポートを追加します。

・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

RTC Builder - Flip/RTC.xml - Eclipse SDK			- • ×
ファイル(E) 編集(E) ナビゲート(N) 検索(A) プロ:	ジェクト(<u>P</u>) 実行(<u>R</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)		
📑 🕶 📾 🗠 🔒 💁 🗣 🔹 💡	$\mathbb{I} \bullet \mathbb{I} \bullet \Phi \bullet \bullet \bullet \bullet$	11 R	RTC Builder 🐉 »
は パッケ ☆ □ ♪ *Flip ☆			- 0
□ 🔄 🍹 🗸 DataPortプロファイ.	ıL	▼ ヒント	^
▲ Frip スのセクションではRT:	コンポーネントのDataPort(ディタポート)の情報を設定します。	データポート	: RTコンポーネ:
*ボート名 (InPort)	Add *ボート名 (OutPort) Add		データを出力す InPortとOutPc
originalImage	Delete Delete	InPort :	RTコンポーネ:= 他のRTコンポー
- Detail		OutPort :	RTコンポーネ) 他のRTコンポー
このセクションではデー 上のデータポートを選択	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。 ですると、それぞれのドキュメントが記述できます。	ポート名:	データポートを ポート名は、同 ポート名に対し ASCII文字が使
	age (InPort)	データ型:	データポート間
*データ型 RTC::Can	neraImage 🗸 🗸		データ型はOpe 使用することカ
変数名 originalIm	nage	変数名:	データポートに
表示位置 LEFT	•		変数の名称は言・
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 RTG	C.xml	
BuildView 😒			- 0
	Σ orioinalimane		
	Flip		
□*			

「コンフィギュレーション」タブを選択し、先ほど決めた仕様を元に、Configuration の情報を入力します。制約条件および Widget とは、RTSystemEditor でコンポーネン トのコンフィギュレーションパラメータを表示する際に GUI で値の変更を行うための 形式を表すものです。

ここでは、flipModeの値は先ほど仕様を決めたときに、-1,0,1の3つの値のみ取ることにしたので、ラジオボタンを使用することにします。[Add]ボタンを押して新しいコンフィギュレーションを追加します。

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

RTC Builder - Eclipse SD		
ファイル(E) 編集(E) ソー	ス(<u>S</u>) リファクタリング(I) ナビゲート(<u>N</u>) 検索(<u>A</u>) プロジェクト(<u>P</u>) 実行(<u>R</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)	
	物 • Q • Q • Ø · Ø • 图 • 日 • 中 中 • · · · · · · · · · · · · · · · ·	RTC Builder
# /tッケ 23 □ □ □ 気 マ > ┣ Flip	*Flip 23 このセクションではRIコジバーネントのコジノイキュレーション・ハラメーダを指定します。 *名称 ② flipMode Add Delete	ー 日 Lonng. Þaram. 、) ノ(ラメータ名:
		データ型: 👘
	▼ 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。	デフォルト値:
	パラメータを: flipMode ③	変数名:
	*データ型 int	単位: 制約条件:
	変数名: flipMode	
	単位:	
	Widget: radio	Widget :
	▼ 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	- F
	BuildView 🔉	- 0)
< Þ	ZorininalImane Flip	
D*		

「言語・環境」タブを選択し、プログラミング言語を選択します。 ここでは、C++(言語)を選択します。言語・環境はデフォルトでは設定されていないので、指定し忘れる とコード生成時にエラーになりますので、必ず言語の指定を行うようにしてください。

RTC Builder - Flip/RTC.xi	ml - Eclipse SDK	
ファイル(E) 編集(E) ナビ	ゲート(<u>N</u>) 検索(A) プロジェクト(<u>P</u>) 実行(<u>R</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)	
📫 - 🖬 🕼 🖾 🔒	Q₄ • < ∅ • ∅ • ∅ • ♥ • • • •	🗈 💦 RTC Builder 🖏 »
パッケ 🛛 🗖 🗖	א Flip צ	- 0
	言語・環境	<u>^</u>
RTC.xml		
	このセクションでは使用する言語を指定します	言語: RTコンポーネントを作成
		環境: 言語ごとのライブラリの
		許柯明報で設定した内容
	© Python	
	Ruby Use old build environment.	
	▼ 物境 このセクトミュンでは依在するライブラリや使用するOSth どを指定します。	
	Version OS	
	Add	
	Delete	
		÷
	•	,
	基本 アクティビティ データボート 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
	BuildView 🛙	
		Â
		E
	Flip	-
L	· · · · · · · · · · · · · · · · · · ·	

全ての設定が完了しましたら、「基本」タブに戻りコード生成ボタンをクリックします。 問題がなければコンポーネントの雛型が生成されます。

RTC Builder - Flip/RTC.x	ml - Eclipse SDK		
ファイル(E) 編集(E) ナビ	ゲート(<u>N</u>) 検索(<u>A</u>) プロジェクト(<u>P</u>) 実行(<u>R</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)		
	६ • <i>∦</i> • <u>1</u> • २ • २ • • •	🖹 👔 RT	°C Builder 🐉 »
■パッケ 🕱 🛛 🗖	★*Flip \$3		
E 😓 ⊽ ▲ 🕞 Flip → RTC.xml	RTC Type :	- コンパー ロン 「注天、	DataFlow FiniteSta MultiMod
	▼ コード生成とバッケージ化	最大インスタンス数:	生成可能な・
	コードの生成おければいケーミンとを行います	実行型:	実行型を指決
	コード生成 ノバッケージ化	実行周期:	コンポーネ: この設定値(
	▼ プロファイル持報のインポート・エクスポート	概要:	RTコンポー
	プロファイル情報のインボートおよびエクスボートを行います。 インボート エクスボート	RTC Type :	特定機能を 値が省略され
		コード生成:	設定した情報
		パッケージ化:	RTCのソーン
		インポート:	RtcProfile形
		パッケージング:	設定した情報
			-
	*	RTC.xml	•
	BuildView 🕄		
	∑ originalTmage flippedTmage	>	
	Flip		
D°			10

5. 仮ビルド

ここまでの作業で Flip コンポーネントの雛型が生成されました。 次の作業として CMake を利用してビルド環境の Configure を行います。 スタートメニューなどから CMake (cmake-gui)を起動します。

A CMake 3.2.1 - C:/rtcv	vs/Flip/build			
File Tools Options	Help			
Where is the source code:	C:/rtcws/Flip			Browse Source
Where to build the binaries:	C:/rtews/Flip/build			▼ Browse Build
Search:		Gr	ouped 📃 Advanced	Add Entry Remove Entry
Name		Value		
3				
	Press Configure to update and display new values in	n red, then press Generate to e	generate selected build	l files.
Configure Generate	Current Generator: None			

画面上部に以下のようなテキストボックスがあります。

- $\boldsymbol{\cdot}$ Where is the soruce code
- Where to build the binaries

「Where is the soruce code」に CMakeList.txt が有る場所、「Where to build the binaries」 にビルドディレクトリを指定します。

CMakeList.txt はデフォルトでは<ワークス ペースディレクトリ>/Flip になります。

ビルドディレクトリとは、ビルドするためのプロジェクトファイル やオブジェクトファイル、バイナリを格納する場所のことです。場所は任意ですが、この場合 <ワークスペース ディレクトリ>/Flip/build のように分かりやすい名前をつけた Flip のサブディレクトリを 指定することをお勧めします。

ディレクトリは自動で作成されるので指定前に作成する必要はありません。

今回は以下の様になるはずです。

Where is the soruce code	C:¥rtcws¥Flip
Where to build the binaries	C:¥rtcws¥Flip¥build

指定したら、下の Configure ボタンを押します。すると下図のようなダイアログが表示さ れますので、生成したいプロジェクトの種類を指定します。



Visual Studio バージョン	32/64 bit	生成したいプロジェクトの種類
Visual Studio 2013	32 bit	Visual Studio 12 2013
	64 bit	Visual Studio 12 2013 Win 64
Visual Studio 2015	32 bit	Visual Studio 14 2015
	64 bit	Visual Studio 14 2015 Win 64

ダイアログで Finish を押すと Configure が始まります。問題がなければ下部のログウイン ドウに Configuring done と出力されますので、続けて Generate ボタンを押します。 Generating done と出ればプロジェクトファイル・ソリューションファイル等の出力が完 了します。

次に先ほど指定した build ディレクトリの中の Flip.sln をダブルクリックして Visual Studio 2013 を起動します。

起動後、ソリューションエクスプローラーの ALL_BUILD を右クリックしビルドを選 択してビルドします。特に問題がなければ正常にビルドが終了します。



4. ヘッダ、ソースの編集

アクティビティ処理の実装
Flip コンポーネントでは、InPort から受け取った画像を画像保存用バッファに保存し、その保存した画像を OpenCV の cvFlip0関数にて変換します。その後、変換された画像を
OutPort から送信します。

onActivated0,onExecute0,onDeactivated0での処理内容は下記図になります。



Flipコンポーネントの処理の流れは以下の図になります。

	Inportからの 画像を保存	Flipコンポーネント	OutPortへ画 像を出力	
Inport		変換		Outport
	UNIVERSITY OF AIZU	cvFlip関数	UNIVERSITY OF AIZU	
	m_imageBuff		m_flipImageBuff	

2. ヘッダファイル (Flip.h) の編集

OpenCV のライブラリを使用するため、OpenCV のインクルードファイルをインクルード します。下記内容をインクルードしている所に追加してください。

```
//OpenCV 用インクルードファイルのインクルード
#include<cv.h>
#include<cxcore.h>
#include<highgui.h>
```

画像の保存用にメンバー変数を追加します。下記内容を class の private:の中(// <rtctemplate block="private_attribute">の下)に追加してください。

IplImage* m_imageBuff; IplImage* m_flipImageBuff;

3. ソースファイル (Flip.cpp) の編集

下記のように、onActivated(),onDeactivated(),onExecute()を実装します。

onActivated()

```
RTC::ReturnCode_t Flip::onActivated(RTC::UniqueId ec_id)
```

```
// イメージ用メモリの初期化
m_imageBuff = NULL;
m_flipImageBuff = NULL;
// OutPortの画面サイズの初期化
m_flippedImage.width = 0;
m_flippedImage.height = 0;
return RTC::RTC_OK;
```

onDeactivated0

}

RTC::ReturnCode_t Flip::onDeactivated(RTC::UniqueId ec_id)
{
 if(m_imageBuff != NULL)
 {
 // イメージ用メモリの解放
 cvReleaseImage(&m_imageBuff);
 cvReleaseImage(&m_flipImageBuff);
 }
 return RTC::RTC_OK;
}

onExecute()

```
RTC::ReturnCode_t Flip::onExecute(RTC::UniqueId ec_id)
 // 新しいデータのチェック
 if (m_originalImageIn.isNew()) {
   // InPort データの読み込み
   m_originalImageIn.read();
   // InPort と OutPort の画面サイズ処理およびイメージ用メモリの確保
  if(m_originalImage.width != m_flippedImage.width || m_originalImage.height != m_flippedImage.height)
    3
  m_flippedImage.width = m_originalImage.width;
  m_flippedImage.height = m_originalImage.height;
   // InPort のイメージサイズが変更された場合
   if(m_imageBuff != NULL)
     {
       cvReleaseImage(&m_imageBuff);
       cvReleaseImage(&m_flipImageBuff);
   // イメージ用メモリの確保
   m_imageBuff = cvCreateImage(cvSize(m_originalImage.width, m_originalImage.height), IPL_DEPTH_8U, 3);
   m_flipImageBuff = cvCreateImage(cvSize(m_originalImage.width, m_originalImage.height), IPL_DEPTH_8U, 3);
   // InPort の画像データを IplImage の imageData にコピー
   memcpy(m_imageBuff->imageData,(void *)&(m_originalImage.pixels[0]),m_originalImage.pixels.length();
   // InPort からの画像データを反転する。 m_flipMode 0: X 軸周り, 1: Y 軸周り, -1: 両方の軸周り
   cvFlip(m_imageBuff, m_flipImageBuff, m_flipMode);
   // 画像データのサイズ取得
   int len = m_flipImageBuff > nChannels * m_flipImageBuff > width * m_flipImageBuff > height;
   m_flippedImage.pixels.length(len);
   // 反転した画像データを OutPort にコピー
   memcpy((void *)&(m_flippedImage.pixels[0]),m_flipImageBuff->imageData,len);
   // 反転した画像データを OutPort から出力する。
   m_flippedImageOut.write();
 }
 return RTC::RTC OK;
```

5. CMake によるビルドに必要なファイルの生成

C:¥rtcws¥Flip¥src¥CMakeLists.txt を編集します。

このコンポーネントでは OpenCV を利用していますので、OpenCV のヘッダのインクルード パス、ライブラリやライブラリサーチパスを与えてやる必要が有ります。以下の 2 点を追加・ 変更するだけで OpenCV のライブラリがリンクされ使えるようになります。

- ・find_package(OpenCV REQUIRED)を追加
- ・最初の target_link_libraries に \${OpenCV_LIBS} を追加
 - ・target_link_libraries は2ヶ所あります。
 - ・追加するときは\${OpenCV_LIBS}の前に半角スペースを入れてください。

set(comp_srcs Flip.cpp) set(standalone_srcs FlipComp.cpp) find_package(OpenCV REQUIRED) ←この行を追加 : 中略 add_dependencies(\${PROJECT_NAME} ALL_IDL_TGT) target_link_libraries(\${PROJECT_NAME} \${OPENRTM_LIBRARIES} \${OpenCV_LIBS}) ←OepnCV_LIBS を追加 : 中略 add_executable(\${PROJECT_NAME}Comp \${standalone_srcs} \${comp_srcs} \${comp_headers} \${ALL_IDL_SRCS}) target_link_libraries(\${PROJECT_NAME}Comp \${OPENRTM_LIBRARIES} \${OpenCV_LIBS}) ←OepnCV_LIBS z追加

6. VC++によるビルド

CMakeList.txt を編集したので、再度 CMake GUI で Configure および Generate を行います。 CMake の Generate が正常に終了した事を確認し、Flip.sln ファイルをダブルクリックし、 Visual C++ 2013 を起動します。

Visual C++ 2013 の起動後、下図のように右クリックでコンポーネントのビルドを行います。

FIL	E EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP					
Ser				Solution E	xplorer	• ‡ ×
ver E				000	0-20	10 × -
xplo				Search So	lution Explorer ((Ctrl+:) P -
Ę		ä	Build		n 'Flip' (9 proj	ects)
8			Rebuild		BUILD	
box			Clean		_IDL_TGT	
			View	•	ygen_doc	
			Analyze	•	Comp	
			Project Uniy	,	TALL	
			Scope to This		KAGE	
		(D)	New Solution Explorer view		O CHECK	
			Build Dependencies	,		
			Add Class Wissed	, chill chill y		
		₩.	Manage NuGet Packages	CUT+SHITC+X	plorer Team B	Explorer Class View
		ö	Set as StartUp Project			
	Output		Debug		Project Prop	erties -
	Show output from: 🔹 🔹 😫 🛃		Source Control	,		
		ж	Cut	Ctrl+X		
			Paste	Ctrl+V		ALL_BUILD
		×	Remove	Del	ependencies	C:Xrtour/XElinXhuildXALL_BUIT
		X=	Rename		nespace	C.+ICCWS+FIID+DUIG+ALL_DOI
			Unload Project			
			Rescan Solution			
	Error List Output Find Symbol Results	\$	Open Folder in File Explorer		e project nam	ε.
Pars	ing included files (470 of 483) - c:¥Program Files¥Windows Kits¥8.1¥Include¥um¥winioctl.h	۶	Properties	Alt+Enter		

7. Flip コンポーネントの動作確認

1. NameService の起動

コンポーネントの参照を登録するためのネームサービスを起動します。

[スタート]メニューから[すべてのプログラム]→[OpenRTM-aist x.y]→ [tools]→[Start Naming Service]をクリックして下さい。

※[Start Naming Service]をクリックしても omniNames が起動されない場合は、フルコン ピュータ名が 14 文字以内に設定されているかを確認してください。

※OpenRTM-aist C++ 1.1.1 使用の方は[Start C++ Naming Service]クリックしてください。

※Windows8の場合下記パスを参考にあります。

C:¥ProgramData¥Microsoft¥Windows¥Start Menu¥Programs¥OpenRTM-aist x.y¥Tools

- Flip コンポーネントの起動
 C:¥rtcws¥Flip¥build¥src¥Debug の FlipComp.exe をダブルクリックで起動させます。
- カメラコンポーネントとビューアコンポーネントの起動 USB カメラのキャプチャ画像を OutPort から出力する OpenCVCameraComp と InPort で受け取った画像を画面に表示する CameraViewerComp を起動します。 [スタート]メニューから[すべてのプログラム]→[OpenRTM-aist x.y]→ [C++]→ [Components]→[OpenCV-Examples] 内にあるのでダブルクリックで起動してください。

8. コンポーネントの接続

1. RTSystemEditorの起動

最初に RTSystemEditor を起動します。

起動方法は RTCBuilder 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] を選択します。そして「パースペクティブ」の中から[RT System editor]を選択して起動さ せます。

またはスタートメニューの「OpenRTM-aist x.y」→「tools」→[RTSystemEditorRCP]から 起動します。

RT System Editor RCP		X-
File Window Help		
THE OFE		
🍺 Name Ser 👘 Repositor 🖵 🗖	-8	□ プロパティー ▽ □ □
🕆 💐 🗞 🗐 🗘 🖒		
אד localhost		
	Configur KI Manager KI Composi KI Executio KI RI Log U	
	ComponentName: ConfigurationSet: 編集	
	active config name value	
	10 H	
	キャンセル	
	複製 追加 追加 削除 □詳細	

2. コンポーネントの接続

Name Service View に何も表示されていない場合は、RTSystemEditor の左側の Name Service View のコンセントアイコンをクリックし、ネームサーバへ接続します。表示され た接続ダイアログに localhost と入力します。

ネームサーバのアドレスをノ	、力してください。
localhost	← (Address:Port)
OK	±475/47/1.

Name Service View に[localhost]のリストが表示されます。

メニューバーの online エディタアイコン(ON と書かれたアイコン)をクリックし、 SystemDiagram を開きます。

RT System Editor RCP			
File Window Help			
न्द्र 📴 👹 🖉 🔐			
🏄 Name Ser 👘 Repositor 🗖 🗖	🖬 System Diagram 🕱 📃 🗖	□ プロパティー	~ - 8
🏠 🗘 🖓 📑 🎲 🤣 🎽 🎽		プロパティー	値
ד localhost		त्रा System Diagram	
		System ID	
		Kind	ONLINE
		Create Date	
		Update Date	
		Composite	None
	Configur T Manager T Composi T Executio T RT Log C		
	ComponentName: ConfigurationSet: 福集		
	active config name value		
	複製 し 追 加 道 加	< <u> </u>	Þ
		L	

次に Name Service View から各コンポーネントをドラックアンドドロップで SystemDiagram 上にコンポーネントを配置してください。

コンポーネントのデータポート同士を接続します。片方のデータポート上でドラッグする と線が伸びるので、接続したいデータポート上まで線を伸ばし接続します。 接続すると接続プロファイルが表示されるので OK をクリックします。

e Window Help				
र् हेह्				
Name Ser 👘 Repositor 🗖 🗖	ត្រ] *System Diagram 🙁	- 8	□ プロパティー	
🟠 🗘 🔿 📑 🀎 🤣 💆			プロパティー	値
त्र localhost	-		त्र System Diagram	
NB1503013 host_cxt			System ID	
CameraViewer0 rtc			Kind	ONLIN
Flip0 rtc			Create Date	
OpenCVCamera0 rtc			Update Date	
			Composite	None
	OpenCVCamera0	p0		
		CameraViewer0		
	Configur KT Manager KT Composi	T Executio T RT Log 🗖 🗆		
	ComponentName: ConfigurationSet:			
	Componentitiame. Configurationset.	編集		
	active config name	Value 適用		
		キャンセル		
	· 複製 · 追t	追加 削除		
		~~~~~	2 111	

接続が完了すると下記図の様になります。

3. コンポーネントの Activate

RTSystemEditorの上部にあります「ALL」というアイコンをクリックし、 全てのコンポ ーネントをアクティブにします。正常にアクティブになると、下図のように黄緑色でコンポ ーネントが表示されます。

RT System Editor RCP			
File Window Help			
ōR ōf 🔛 🖻 🥙 🔐			
🏄 Name Ser 👘 Repositor 🖓 🗖	💀 *System Diagram 🙁 🗖	□ プロパティー	
🖞 (+) (+) 👔 🐉 📀 🌶 🗸		プロパティー	値
דא localhost		त्र System Diagram	2.000
NB1503013 host_cxt		System ID	
CameraViewer0 rtc		Kind	ONLINE
Flip0 rtc		Create Date	
OpenCVCamera0 rtc		Update Date	
		Composite	None
	OpenCVCamera0		
	FIIpo		
	CameraViewer0		
	Configur KI Manager KI Composi KI Executio KI RI Log		
	ComponentName: ConfigurationSet: 編集		
	active config name Value		
	道用 道用		
	キャンセル		
	複製 追加 削除	< III	•
L			

#### 4. 動作確認

下図のようにコンフィギュレーションビューにてコンフィギュレーションを変更すること ができます。Flip コンポーネントをクリックしてコンフィギュレーションビューの編集を 押すと下記ダイアログが出てきます。「flipMode」を「0」や「-1」などに変更し画像が反転 することを確認してください。

RT System Editor RCP		-		
File Window Help				
祝 萜 🕨 🖉 🦛 🏤				
🏄 Name Ser 🕅 Repositor 🗖 🗖	🚮 *System Diagram 🔀	- 0	□ プロパティー	~
🗄 ⇔ ⇔ 📑 券 🔗 🗯 ▽			プロパティー	值 🔺
त्र localhost			🗊 Flip0	
NB1503013 host_cxt			Path URI	localhc
CameraViewer0 rtc			Instance Name	Flip0
Flip0 rtc			Type Name	Flip 😑
DpenCVCamera0 rtc			Description	Flip im
			Version	1.0.0
	OpenCVCamera0 Flip0		Vendor	AIST
			Category	Catego
default ConfigurationSet : default				
flipMode 💿 -1	© 0	<b>0</b> 1		
		ОК	V +77211	Apply
		4/2	<b>د</b> ۱۱۱	+

## 3. RTC-Library-FUKUSHIMA

1. RTC-Library-FUKUSHIMA について

**RTC-Library-FUKUSHIMA** とはロボット産業振興のために作成された **RTC** ソフトウェアラ イブラリーです。

主にコンポーネントの登録やダウンロードしての再利用などが出来ます。

		♥ <u>ログイン</u>	Google"カスタム検索		検索
UNIVERSITY OF AIZU		ミドルウェア	ライブラリ	ドキュメント	フォーラム
RTC-Library-FUKU OpenRTM-aistを利用した、ロ ・RTCライブラリ	SHIMA Jボット・テクノロジー・コンポーネ OpenRTM-aist	ント (RTC) ライン	プラリ		
お知らせ	イベント 2015年07月24日	<u>RTミドルウェア</u> 習会が行われまし	<u>強化月間2015 i</u> した	<u>n 中央大学・RT</u> ≋	ミドルウェア講
▶ お知らせ一覧を見る	案内 2015年07月09日	<u>Fedora22のOp</u> e しました	enRTM-aist (C	<u>++/Python) /್ರಿ</u>	ッケージを公開

2. コンポーネントをアップロード

RTC-Library-FUKUSHIMA へのコンポーネントのアップロードの仕方を説明します。

1. ログイン

RTC-Library-FUKUSHIMA へは下記 URL でアクセスします。 RTC-Library-FUKUSHIMA: https://rtc-fukushima.jp/

今回の講習会では本番の環境を使わずにローカルの環境を使用します。



サイトにアクセス出来たらサイト上部のログインをクリックしてください。

ログイン画面に移行しユーザー名またはメールアドレスとパスワードを入力する欄があり ますので入力してログインをしてください。

		Google"カスタム検索		検索
UNIVERSITY OF AIZU	ミドルウェア	ライブラリ	ドキュメント	フォーラム
ログイン				
<u> トップページ &gt; ログイン</u>				
ユーザー名またはメールアドレス   パスワード:				

2. コンポーネントのアップロード手順

コンポーネントをアップロードするにはログイン後、トップページから「ライブラリ」を選 択し、「コンポーネント登録/パッケージ登録」のタブを選択します。そして「コンポーネン ト登録」を選択します。

<u>トップページ</u> > ラ1	ブラリ		
ライブラリ検索	コンボーネント登録/パッケージ登録		
コンポーネン	ト登録/パッケージ登録		
RTコンポーネントま 利用規約などに同意	たはパッケージの登録を行います。 の上、ガイドラインに従って登録をお願いし	<b>\たします。</b>	
RTコンポーネントま 未登録の場合は、先	たはパッケージの登録には、会員登録およる に会員登録を行っていただき、ログインをお	ジログインが必要です。 3願いいたします。	
▶ コンポーネント	登録 ▶ パッケージ登録		
▶ 登録ガイドライ	>		

下記登録画面に遷移したことを確認してください。下記画面で登録を行います。

	▲ ようこそ、もうえ様	☞ <u>¤グアウト</u>	Google"カスタム検索		検索
UNIVERSITY OF AIZU		ミドルウェア	ライブラリ	ドキュメント	フォーラム
コンポーネント登録					
<u>トップページ</u> > <u>ライブラリ</u> > コンポ-	ーネント登録				

今回は下記項目を登録します。

RTC.xml ファイル読み込み

Flip コンポーネントで作成された RTC.xml を指定します。指定後、「RTC.xml ファイル読 み込み」のボタンを押してください。

RTCBuilder で設定したコンポーネントの情報が登録されます。

コンポーネント登録情報入力

・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:コンポーネントを zip に圧縮してアップロードします。その
際、build 以下は削除か退避しておいてください。
・同意する:チェックを入れてください。
・私はロボットではありません:チェックを入れてください。※ローカル環境ではなし

入力が終わりましたら、「確認」のボタンを押し登録情報確認ページに遷移してください。

## 4. Flip コンポーネントの全ソース

- Flip コンポーネントソースファイル (Flip.cpp) Flip.cpp のソースコードを以下に記載します。 Flip.cpp : https://rtc-fukushima.jp/wp/wp-content/uploads/2016/02/Flip_cpp.txt
- Flip コンポーネントのヘッダファイル (Flip.h) Flip.h のソースコードを以下に記載します。 Flip.h : https://rtc-fukushima.jp/wp/wp-content/uploads/2016/02/Flip_h.txt
- Flip コンポーネントの全ソースコード
   Flip コンポーネントの全ソースコードを以下に添付します。
   Flip.zip: https://rtc-fukushima.jp/wp/wp-content/uploads/2016/02/Flip.zip