

第2部:RTコンポーネント作成入門

名城大学 理工学部メカトロニクス工学科 大原賢一







第2部での目標

- RT System Editorを用いたRTCベースのシステム構 築方法の習得(RTC運用時に必要な知識)
- RTC Builderを用いたRTコンポーネントのひな形作成 方法の習得(RTC開発時に必要な知識)



■ 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

など

■ 構成

名

• RTCビルダ

城 大 学

- RTCデバッガ
- 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

PLATFORM An Open Software Platform for Robotic Techn Home Software	OpenRT Platform オフィシャルサイト
トップ	OpenRT Platform オフィシャルサイト
ダウンロード Downloads Open Source Software	ロボット知能ソフトウェアプラットフォーム
知能化PJ Project members only 産総研コンソ Consortium members only	、次世代ロボットシステムの効率的かつ効果的な研究開発環境を構築するために、ロボット知能化技術をRTコンボーネントとしてモジュール化し、これらを統合し て次世代ロボットシステムのシステム設計、シミュレーション、動作生成、シナリオ生成を行うことができるロボット知能ソフトウェアブラットフォームの研究開発を 行っています。
再利用WG WG members only	新着情報
問い合わせ先	1 2 Next
Construction Co	Mar 2012 JDKのパッケージライセンス変更 OracleによりJDKのライセンスが変更になっています。そのため、Ubuntuなどのディストリビューションでsun-java6などのパッケージ配布が中止になりました。 この変更に伴い、OpenHRP31では、そのままではパッケージインストールが不可能になっております。 もし、OpenHRP31をスクラッチから導入されたい場合には、開発チーム(opentp@m.aist.go.jp)までご連絡をお願いいたします。 Ubuntu10.04LTSに対する暫定的が対応
Login	Mar 2012 OpenRTM-aist-1.000合語実装 OpenRTM-aist-1.0をC言語で建築中です。まだ、o版ですが、ドキュメントとソースコードを公開します。 http://openrtp.jp/OpenRTM-C/index.html
	Oct 2011 国際ロボット展 セミナー詳細のついか 2011年11月11日に国際ロボット展で開催されるセミナーの時間割止名セミナーの概要をアップしました。 また、Choreonoidに関するセミナーの配布資料もアップ致しましたので、参照して下さい。 セミナー概要:IRex2011 セミナー2::Rex2011Seminar2





■ 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



■ ダウンロードし, 解凍するだけ

名

城 大 学

※Javaの実行環境については、別途インストールが必要

		× =	🔹 📧 🎦 🛛 Hello My Account Help Contact Sign Out
	PTM-aiet		
The power to cor			
MIDDLEWARE			
ホーム タリ	フロート トキュメノト コミュニナイ 研究・開発	70727 N-FULP	
Google Translate	ホーム >> ダウンロード >> ツール >> Eclipse tools 1.	1.0-RC2	
言語を選択	OpenRTM Eclipse tools 1.1.0-RC2 投稿者:s-kurihara 投稿日時:火, 2011-10-11 18:36		
Powered by Google 翻訳 ナビケーション	これまで、OpenRTM-aistのジ する OpenRT Platform に組。 す。	マールとして開発されてきた RTCBUilder (I日RtcTemplate) および RTSystemEditor (み込まれることになりました。こちらでは、RTSystemEditor 及び RTCBuilder のみを	(日 RtcLink) は、OpenHRP3やその他のツールと統合開発環境を構成 記布していますが、将来的コは様々なツールを一括で提供する予定で
。 ホーム	現在の RTSystemEditor 及乙	、RTCBuilderの最新バージョンは 1.1.0 です。	
▶ Python版	Table of contrasts		
▶ Java版	able of contents		
⇒ ツ≒ル	 ・ 至部入りハックーシ ・ バイナリ 		
 Eclipse tools 1.1.0- 	• Eclipse/JDK/JRE等		
 Eclipse tools 1.1.0- RC1 	・ 過去のバージョン		
• Eclipse tools 1.0- RELEASE	全部入りバッケージ		
 Eclipse tools 1.0.0- 		Eclipse-3.4.2 [Ganymede SR2]	
RC1	Eclipse3.4.2+RTSE+RTCB	eclipse342_rtmtools110-rc2_win32_ja.zip	2044 07 22
 ccupse tools 0.4.2 rtshell(CUPンール) 	Windows用全部入り	MD5:2e6f9fa3e370b6e7ac1f9340d36c7abf	2011.07.22
 ▶ Pythonライブラリ (rtctree/rtsprofile) 	 Ubuntu8.04, Ubuntu9.10, Ubuntu10.04でLint Ubuntu8.04では、apt-get install xulrunne 	x用Ectipse3.4.2が動作しない不具合が報告されています。 sr-1.9 として xulrunninerをアップデートしてください。	
 RtcLink• RtcTemplate 	• Ubuntu9.10,Ubuntu10.04では、以下の方法	透利用するか、Eclipse3.3もしくは3.5をご利用ください。	
 ▶ コンボーネント ■ RTC/RTS仕様記述方式 	¢		
▶ ドキュメント	<pre>\$ su # vi /etc/apt/source.lis</pre>	t	
▶ ವ≋ಎದರ್ಗ	1行追加 - deb http://jp.	archive.ubuntu.com/ubuntu/ jaunty main restricted	
▶ 研究·開発	<pre># apt-get update # apt get install]</pre>	non 1 0	
▶ プロジェクト - いードウェア	# apt-get install Xulrun # dpkg -1 grep xulrunne	r-1.9	
0 /1-I.OT).	ii xulrunner-1 9	1 9 0 &+nobinonly-Oubuntu?	XIII + XPCOM application rupper
0 // T. V. T.	ii xulrunner-1 9	1 9 0 8+nohinon]v=0uhuntu2	XIII + XPCOM application runner

システム構築支援ツール RTSystemEditorについて



名城大学



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





● パースペクティブを開く	<u>- 🗆 ×</u>
 ■ C/C++ ■ CVS リボジドリー・エクスプローラー ■ Java (デフォルト) ● Java の型階層 ● Java 参照 ● Planning ● Pydev ■ TR System Editor ● R TO Builder ● 「-ム同期化 ☆ デバッグ ◆ フラグイン開発 ● リソース 	
<u>OK</u> +	ゃンセル

※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

RIDDLEWARE

RTSystemEditorとは?

尊

城大

● 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







学

统大

名城大学



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]

■ 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] →[ConsoleInComp.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] →[ComsoleOutComp.exe]





城 大 学





システムエディタの起動

Editor	E	clip	se SI)K
集(<u>E</u>) ナ	モゲート	•(<u>N</u>)	検索	(<u>A</u>)
a 🗈	<u>N</u>	DFE.	0	•]
ice 🛛	Ô	Repo	sitory	Vie
IA Z				6

RT System Editor Eclipse SI ファイル(E) 編集(E) ナビゲート(N) 検索	DK X(<u>A</u>) プロジ	「ェクト(P) 実行(B) ウ	ロンドウ心 ヘルプ田)				_0×
📸 • 📰 📥 🚠 🚾 💑 💁	•] 👝 ,	🛷 •] 🖢 • 🖓 •	÷	f 89 89		📑 🕱 I	RTC-CANope , »
鯶 Name S 🙁 🍈 Reposit 🖳 🗖	on Syste	m Diagram 🛛			- 0	□ プロパティー S3	~
Mome S S (Pepoit) □ T below the second	Config Compon active	uration Vie S ₹ entName config	Manager Control	omposite Compo	T Log View) 日 「編集」 」 ①明	D 20/57 ⊠ ⊐D/57 B RT System Diagram System ID Kind Create Date Update Date Update Date Composite	ONLINE None
					++>>tz11		







※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

- 🗆 × 編集(E) ナビゲート(N) 検索(A) プロジェクト(P) 実行(B) ウィンドウ(W) ヘルプ(H) 🖴 🔚 🐻 💑 🔍 • 🔗 • 🗇 • 전 • 전 • 😓 • - > • 🏼 🖉 📽 🏦 📑 📧 RT System E... 🔚 🍺 Name S 🙁 🌎 Reposit) 🖓 🗖 🚮 *System Diagram 🕺 プロパティー ㅋ ㅈ⋷ System Diagram 😑 🔿 🛛 📑 🎲 🤣 😹 值 B…⊼⊤ localhos ≜- 📋 saka-PC|host_cxt System ID ONLINE Kind ConsoleIn0|rtc Create Date Update Date Composite ConsoleInf Configuration 🛛 🕺 💦 Manager Contr 🕅 🔭 Composite Co 🗍 🥂 Execution Con 🕅 🔭 RT Log View) 🖓 🗖 ComponentNam ConfigurationSet 偏集 active config 複製 追加 追加 削除
- 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



※ポートのプロパティが異なる場合など,



Long

奋

stem Editor Eclipse SI		ninim(m) da o indulatio	- 11 MAR A			_ D ×
MARCED JEJ-FOD ACK	B JUSISHO	美田田 ワインドス田	~/F/@			
ii 🗅 🗟] 🗟 🕅 🏊	•] 🔗 •] 🖄 •	9 · · • • · •	-] 🖻 👹 🍪 🔐			RT RT System E 🗈 🍽
S 😫 🌔 Reposit 🔍 🗆	6 System Diagram	8		•	I = 70/(7- %)	~ - 0
👔 📚 🤌 🖉 🔼					プロパティー	値
cahost saka-POhost_cxt ConsoleBill/tc					BRT System Diagram System ID Kind	ONLINE
5 ConsoleOut0 rtc		out	in		Lipdate Date	
	ConsoleIn0		ConsoleCut0		Composite	None
	Configuration	2 KT Manager Co	fr RT Composite Co RT Execut	ion Con 🛛 💦 RT Los View 🗋 📟		
	ComponentName	Configuration	Sat			
	active Coortie	00000	Vakus	NEX.		
	active comis		value			
				キャンセル	1	
				1		
	77.52	3臣九日	近初	AUNE		



コンポーネントの起動







RT System Editor – – Eclipse SE	DK					
マイル(E) 編集(E) ナビゲート(N) 検索	(<u>A</u>) プロジェクト(<u>P</u>) 実行(<u>R</u>)	ウィンドウ(W) ヘルプ(日)				
📬 - 🔛 🚊 📾 🕷 👘 💊	•] 🔗 •] 🖢 • 🕅 • *	> 🗢 • → •] 🖻 🞽 💸	i #			🗈 🏹 RT System E 🗉 🛪
¥ Name S 🛿 🍈 Reposit 🗖 🗖	🔂 *System Diagram 🛛			- 0	プロパティー 23	~ - 8
	out				プロパティー B ネT System Diagram System ID Kind Create Date Update Date Composite	I (18 ONLINE None
	Consolerio					
		Manager Contr KI Composite	Co KI Execution Con KI RI L			
	ComponentName: active config	ConfigurationSet name V:	alue	編集 通用 キャンセル		
•						

※停止はDeactivateを実行

※RTC間の接続を切る場合には接続線をDelete もしくは、右クリックメニューから「Delete」を選択 名城大学





14



■ ネットワークインターフェースが2つある場合

学

统大



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:



RTコンポーネントの動作

学

统

アクション名	説明
Activate	対象RTCを活性化する
Deactivate	対象RTCを非活性化する
Reset	対象RTCをエラー状態からリセットする
Exit	対象RTCの実行主体(ExecutionContext)を停止し,終了する
Start	実行主体(ExecutionContext)の動作を開始する
Stop	実行主体(ExecutionContext)の動作を停止する

■各コンポーネント単位での動作変更



■全コンポーネントの動作を一括変更



※ポップアップメニュー中でのキーバインドを追加 ※単独RTCのActivate/Deactivateについては、グローバルはショートカットキー定義を追加

項目	設定内容
Name	接続の名称
DataType	ポート間で送受信するデータの型.ex)TimedOctet,TimedShortなど
InterfaceType	データを送受信するポートの型.ex)corba_cdrなど
DataFlowType	データの送信方法. ex)push, pullなど
SubscriptionType	データ送信タイミング.送信方法がPushの場合有効.New, Periodic, Flushから選択
Push Rate	データ送信周期(単位:Hz). SubscriptionTypeがPeriodicの場合のみ有効
Push Policy	データ送信ポリシー. SubscriptionTypeがNew, Periodicの場合のみ有効. all, fifo, skip, newから選択
Skip Count	送信データスキップ数. Push PolicyがSkipの場合のみ有効

- SubscriptionType
 - New:バッファ内に新規データが格納されたタイミングで送信
 - Periodic:一定周期で定期的にデータを送信
 - Flush:バッファを介さず即座に同期的に送信
- Push Policy
 - all:バッファ内のデータを一括送信
 - fifo:バッファ内のデータをFIFOで1個ずつ送信
 - skip:バッファ内のデータを間引いて送信
 - new:バッファ内のデータの最新値を送信(古い値は捨てられる)



項目	設定内容
Buffer length	バッファの大きさ
Buffer full policy	データ書き込み時に, バッファフルだった場合の処理. overwrite, do_nothing, blockから選択
Buffer write timeout	データ書き込み時に,タイムアウトイベントを発生させるまでの時間(単位:秒)
Buffer empty policy	データ読み出し時に, バッファが空だった場合の処理. readback, do_nothing, blockから選択
Buffer read timeout	データ読み出し時に,タイムアウトイベントを発生させるまでの時間(単位:秒)

EConnector Prof	ile				×
ConnectorProfileを入	力してください。				
Name :	ConsoleIn0.out_Con:	soleOut0.i	n		
Data Type :	IDL:RTC/TimedLone	g: 1.0			•
Interface Type :	corba_cdr				-
Dataflow Type :	push				•
Subscription Type :	flush				•
Push Rate(Hz):					
Push Policy :	all				7
Skip Count :	[
▼ 詳細					
Buffer (Outport) -			-Buffer (Inport)		
Buffer length :	8		Buffer length :	8	
Buffer full policy :	overwrite	_	Buffer full policy :	overwrite	
Buffer write timeo	iut: 1.0	_	Buffer write timeout :	1.0	
Buffer empty polic	sy: readback	_	Buffer empty policy :	readback	
Buffer read timeo	ut: 1.0		Buffer read timeout :	1.0	
Name		Value			ishn (
				1	ĭ⊔R≆
					HUPP
?				OK	キャンセル

※OutPort側のバッファ, InPort側のバッファそれぞれに設定可能 ※timeoutとして「0.0」を設定した場合は、タイムアウトしない

Buffer Policy

- overwrite:上書き
- readback: 最後の要素を再読み出し
- block : ブロック
- do_nothing:なにもしない

※Buffer Policy = Block+timeout時間の指定で、一定時間後 読み出し/書き込み不可能な場合にタイムアウトを発生させる 処理となる





接続プロファイル(ServicePort)について

項目	設定内容
Name	接続の名称
インターフェース情報	接続するインターフェースを設定. 接続対象のServicePortに複数のServiceInterfaceが定義されていた場合,どのイン ターフェースを実際に接続するかを指定

🖨 Port Profile				×
ポートプロファイルを入力してください。				
Name : MyServiceConsumer0.MyServ	ice_MyServiceProvi	der0.MyService		
▼ 詳細				
Consumer		Provider		追加
				削除
Name	Value			追加
0			ОК	キャンセル





画像処理関連コンポーネントの起動

- カメラコンポーネントの起動
 [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
 →[opencv-rtcs]→ [DirectShowCamComp.exe]
- ビューワコンポーネントの起動 [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components] →[opencv-rtcs]→ [CameraViewerComp.exe]
- 個像処理用コンポーネントの起動
 ■Flipコンポーネントの起動

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

 [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
 - \rightarrow [opencv-rtcs] \rightarrow [EdgeComp.exe]



■ Flip側との接続

城 大 学

■ 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

フィイルED 編集ED ナビゲーNM 検索ED オロジェクNED 東内RD ウィンドウMM ヘルプHD ***	
** * ::::::::::::::::::::::::::::::::	
🍺 Name S 🖄 👘 Reposit 🖓 🗖 📷 PSystem Diagram 🛛 💦 💎 👘	
	- 8)
AT location At locati	









■ 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



名城大学



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











ネットワーク上の別のPCで動作するRTCの利用(1)

アクセス可能なネットワーク上に存在する別のPCで動作するネーミングサー ビスにアクセス

RT System Editor RCP	\$21-999				
File Window Help					
ōn2 ōr⊾ ₩ # 69 69					
🝺 Name 🕅 Rer 🔜 💷	a *System Diagram ≈			- 0	■プロパティー ▽□□
🟠 (P) 🖓 📴 🖗 🖉 🎽					プロパティー 値 🔺
⊿ ™ localhost					⇒ Flip0
Isdlab-vm-01 host		20000			Path URI loca
CameraViewer0 rt			the second se		Instance N Flip
DirectShowCam0					Type Nam Flip
Flip0 rtc		A	10. 57		Descriptio Flip
		Sect	Adda V		Version 1.0
		1.44			Vendor Ker
			and the second sec		Category Ima
		*C.	Contraction of the second s		State INA
					properties
					implem: Flip
					type_na Flip
					descript Flip
					version 1.0
		コンセン	トのボタンを押す		vendor Ker
					categor Ima
					activity_ PEF
	Configuration View Manager C	ontrol View R Com	nposite Component View 🚮 Execution Context View 🕅 RT Log View		max_in: 1
	ComponentName: Flip0	ConfigurationSe	et: default	編集	languag C+
	ac config	name	Value	海田	lang_ty cor
		flipMode	0	週用	conf.def 0
		Cherry and	-	キンセノ	conf\ rad
					conf((-1
					sdo.sen
	複製 追加 削除		追加削除		d III cdo con
					×





ネットワーク上の別のPCで動作するRTCの利用(2)

参照したいネーミングサービスが起動しているPCのIPアドレスとポートを入力 する.

▲ ネームサーバへ接続	X
ネームサーバのアドレスを入力してく	ださい。
localhost	✓ (Address:Port)
OK	キャンセル

IPアドレスの確認方法

コマンドプロンプトにおいて、「ipconfig」と入力する.

他のPCで起動しているコンポーネントの閲覧およびRTCの遠隔利用ができる! (ファイヤーウォールがある場合は見えません(利用できない))

コンポーネント開発ツール RTCBuilderについて





第3部での演習の内容

■ 入力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







RTCBuilder概要

■ 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











RT

- Windowsの場合
 - Eclipse.exeをダブルクリック
- Unix系の場合
 - ターミナルを利用してコマンドラインから起動
 - > Ex) \$ /usr/local/Eclipse/eclipse

■ リークスペースの選択(初回起動版



※ワークスペース Eclipseで開発を行う際の作業領域 Eclipse上でプロジェクトやファイルを作成すると ワークスペースとして指定したディレクトリ以下に 実際のディレクトリ,ファイルを作成する

■ ワークスペースの切替(通常時)

틎 Java – Eclipse SD	к
ファイル(E) 編集(E) ソ	ース(S) リファクタリング
新規(<u>N</u>)	Alt+Shift+N
ファイルを開く()	
開じる(<u>C</u>)	Ctrl+₩
すべて閉じる(<u>L</u>)	Ctrl+Shift+W
保管(<u>S</u>)	Ctrl+S
■ すべて保管(E)	Ctrl+Shift+S
前回保管した状態に戻	!す(<u>T</u>)
移動(⊻)	
名前変更(M)	F2
更新(<u>F</u>)	F5
行区切り文字の変換(<u>v</u>
上 印刷(<u>P</u>)	Ctrl+P
ワークスペースの切り替	ż‱
🚵 インポートΦ	NC .
🛃 エクスポート(<u>0</u>)	





■ 初期画面のクローズ

■ 初回起動時のみ



※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

■ 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





プロジェクト作成/エディタ起動



停

统大

※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」

	RT-Co	mponent Builder Project			_ 🗆 ×
	プロジェク	·名(P): Flip			
	▼ デフォ	ルト・ロケーションの使用(型)			
		y(L): C:¥GlobalAssist¥EclipseTes	t2¥Flip	参照((<u>B</u>)
	3		元了	E)	
	6				
	F	🖁 パッケージ・エクスフ	0 8 1]
				< ▽	7
				₽>	
	[🗉 🥟 Flip			
		> RTC.xm			
Builder - Flip/RTC.xml - E	clipse SDK				_DX
D 編集(D) ソース(D) リファクタ 同 二 み 物・O・	リング① ナビゲート(M Q4・ 00 - ペ・) 検索(A) ブロジェクト(D) 実行(B) ウインドウ(A)	0 NH74D		TY TY RTC-CANepe *
7-9-19270 8 -0	Flip 🕄				8
Fie	基本				-
RTC::ml	■ RT-Component このセクション/ではRT	Basic Profile エンボーネントの基本情報を指定します。		▼ ビント モジュール名:	RTコンボーネントを課題する名前を指定します。
	*モジュール名: モジュール概要:	ModuleName ModuleDescription		モジュール概要:	使用できる文字はアルファベット、数字、ハイフン、アンダースゴ RTエンボーネントが提供する複雑の概要を入力します。
	・バージョン: ・ペンダ名:	1.00 VenderName		バージョン・	ASCII文字が使用できます。 RTコンボーネントのパージョンを指定します。
	・モジュールカテゴリ:	Category	-	心.收名:	×メジスタスは数字の形式とCAJしていたし。 RTコンボーネントを作成した作者名、ペンダ名を指定します。
	コンボーネント型:	STATIC	-	モジュールカテゴリ:	NSULX + FUICH(CHEF)。 RTTンボーキントのカナリを入力します。
	コンボーネント種類:	P Data Chur, D COM D MultiMada		コンボーネント型:	使用できる文字は、アルファベット、数字、ハイフン、アンタース RTLンボーネントの型を推定します。
	最大インスタンス数:	1 Latariow [] FSM [] MultiMode			*STATIC 網想に生成/伸歩れないRTC *UNQUE 動的に生成/伸歩れるユニーカなRTC *COMMUTATIVE 動的に生成可能なRTC
	実行型: 実行周期:	PeriodicExecutionContext 1000.0	×	アクティビティ型:	RTコンボーネントのアウティビティ型を指定します。 ・PERIODIC 第11支系がALEの活用率付至RTO ・SPORADIC 第11支系がALEの活用第15型RTC
	概要:		*	コンボーネント種類:	・EVENTDRIVEN イベンド記動型RTG RTコンボーネントの種類を指定します。 ・DataFlowComponent: 見測実行型RTG
	(<u>×</u>		・FiniteStateMachine: 相談状態遺科型RTC ・MultiMode マルギギード型RTC
	金本」アクティビティーデー BuildView 23	9第一下 サービスボート コンフィギュレーション ドキュメ)	/ト生成 言語・環境 RT(2xml RTC-CANoper	
ľ					
			ModuleName		-
	•				



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





基本プロファイル

■ RTコンポーネントの名称など,基本的な情報を設定

≻*Flip 🛿					
基本					
▼ RT-Componer	it Basic Profile	▼ ピント			
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:	RTコンボーネントを識別する名前を指定しま		
*モジュール名:	Flip		しの名称はコンボーネントのヘースインスタンス 使用できる文字はアルファベット、数字、ハイ:	、名にも使用されます。 フン、アンダースコアのみです。	
モンユール(税要: *バージョン:	Flip image component	モジュール概要:	RTコンボーネントが提供する機能の概要をみ ASCIT文字が使用できます。	し力します。	
*べ)))ダ名:	AIST	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。		
*モジュールカテゴリ:	Category				
コンポーネント型:	STATIC	モジュー	·ル名: Flip		
アクティビティ型:	PERIODIC			Elin imaga aa	mnonont
		モンユー	ル 版 安 ・ 壮 息 (rip image coi	mponent)
コノホーネノト推発す	🗹 DataFlow 🗖 FSM 🗋 MultiMode	バー・ジョ	~ 100		
最大インスタンス数	: 1				
実行型:	PeriodicExecutionContext	ベンダネ	;仟意(AISI)		
実行周期:		—		± (0.1	
把 西 ·	OpenCVライフラリのうち,cvFlipの関数を用いて画像の反転を行うコンホーネント	モンユー	ルカナコリ 仕	息(しategory)	
1941 安 ·		 `,#	「ATワ・「忯J、く大」	TIC	
RTC Type :		コンハー	イン「全・いる		
▼ コード生成とパッ	ケージ化	アクティ	ビティ型: PFRIC)DIC	
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	コンホー	ネントの種類	Dataflow	
		=+ハ	フカ、フ米ト・1		
▼ ノリノア1ル1行報		取入1ノ	インノへ致・		
ノロファイル情報的イ	ンボートおよびエジスボートを行います。	室行刑	Periodic Execut	lionContext	
	/h=P	' – 비조			
		実行周期	Я: 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





■ 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	ィビティ					□設定対象のアクティビティ
▼ ₽974ピティ	6		◆ ピント			
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。		onInitialize	初期化処理です。コンポーネントライフサイクル開始時に一度だけ呼ばれます。常に有効。		を選択
	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	onFinalize	終了処理です。コンボーネントライフサイクルの終了時に1度だけ呼ばれます。 「		
onInitialize	onFinalize		onStartup onShutdown	ExecutionContextが実行を簡好自身のとさい見たけ呼ばれます。 ExecutionContextが実行を停止するとき1度だけ呼ばれます。		
	実行コンテキストの起動と停止に関	はるアクション	onActivated	非アクティブ状態からアクティブ化されるとき1度だけ呼ばれます。		
onStartup	onShutdown		onDeactivated	アクティブ状態から非アクティブ化されるとき1度だけ呼ばれます。		onActivated N
	alive状態でのコンポーネントア	20ション	onAborting	ERROR状態に八る前に1度にいずはれます。 ERROR状態に八る間周期的に呼ばれます。		onError は
onActivated	onDeactivated	onAborting	onReset	ERROR状態からリセットされ非アクティブ状態に移行するときに1度だけ呼ばれます。		
onError	on Keset	-L	onExecute	アクティブ状態時に周期的に呼ばれます。		
-	Dataflow型コンボーネントのア	(0)JED - 01 - 1	onStateUpdate	onExecuteの接角回呼ばれます。 1 ExecutionContextのrateが変更されたとき呼ばれます。		
onExecute	onStateUpdate	onRateChanged	onAction	対応する状態に応じた動作を実行するために呼ばれます。		2)使用/未使用を設定
A 11	FSM空コンボーネントのアク	VEV	onModeChange	d モードが変更された時に呼ばれます。		
onAction	Ma Ja 田山つ いやい うくいし のつわ	12 I N.	動作概要:	アクティビティの概要説明を記述します。		_
onModeChong	Midde エコンホーネントのアク	CEV.	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。		
ormodeorians			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。		
▼ Documenta	ation					
このセクションで 上段のアクション	は各ノクソヨンの破異を説明するトキュメントを記述(ンを選択すると、それぞれのドキュメントを記述できま	します。 す。				° 🖻
アクティビティ名	: onInitialize	• ON • OF	F			
	コンポーネント自身の各種初期化処理		A			以下をチェック:
動作概要:						
	+>1		<u>~</u>			onActivated
本前冬 件	au .		<u>^</u>			
- The second			-			an Departive ted
	コンボーネントの初期化処理が正常に完了して	เาอ				ondeactivated
事後条件:						
			×			onExecute
					-	
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.>	ml Mapping ID USB Pi	P RTC-CANopen		

※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

學

城 大

データポート		①該当種類の欄の「Add」ボタンをクリック
▼ DataPortプロファイル	★ ピント	
このセクションではRTコンボーネントのDataPort(データボート)の情報を設定します。	データポート: RTコンポーネント間で: データを出力するOutP InPortとOutPortを接続	し,ポートを追加後,直接入力で名称設定
ariginalImage FilppedImage Delete Delete	InPort: RTコンボーネントにデー 他のRTコンボーネント(ヌポート)の情報を設定します。
	のUtront: RTコンボーネント(他のRTコンボーネント(ポート名: データボートを識別する	*ボート名(OutPort) Add
▼ Detail	ホート名に対して一意 ポート名に対して一意 ASCII文字が使用でき	
このセクションではテータボート毎の機要を認明するドギ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。	データ型: データボート間でやり取 InPortとOutPortを接続	
ポート名: originalImage (InPort)	データ型はOpenRTM2 使用することができます	
*データ型 RTC::CameraImage	変数名: データポートに関連付(変数の名称は言語に。	
変数名 originalImage	ポートの場所: RTSystemEditorなど このプロパティはオプショ	◎記古ナブ型桂却ナ 、
表示位置 LEFT	ドキュメント: データボートに関する情 全てを記述する必要は	②設定9る空情報を一見から選択
Documentation	レベルの情報を記述す	▼ Detail
キャプチャされた画像データ ・		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。
データ型: CameraImage型OpenRTM-aistのInterfaceDataTypes.idlにて定義されているデータ型		ポート名: originalImage (InPort)
データ数: 任意		
反転処理の対象となる画像データ 二意味:		*デーク型 RTC::CameraImage 変数名 RTC::BumperArrayGeometry RTC::BumperGeometry
ー 単位: ねし		表示位置 Introduce ameralmage RTC::CameraInfo RTC::Carlike
		Documentation

※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



■ 生成対象RTCに付加するServicePortの情報を設定

サービスポート				
RT-Component Service Ports		RT-Component S	Service Port Interface Profile	_
sv_name if_name	Add Port Add Interface Delete	このセクションではRT *インターフェース名: 方向: インスタンス名: 変数名: *IDLファイル: *インターフェース型: IDLパス: ▼ Documentati	コンポーネントのService Interfaceの情報を設定します。 if_name Provided C:¥work¥MyService.idl MyService MyService2 MyService3 MyService4 MyService5	Browse Browse
		概要説明:		

サービスインターフェースの指定

城大 学

名

● 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

学

城大

	* L7F	①「Add」ボタンをクリックし、追加後、
このセクションではRTコンボ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 flipMode Add	Config. Param.: RTユン ユンフ・ 再利明 パラメ	直接入力で名称設定
Delete	パラメータ名: コンフ・ パラメ	▼ RT-Component Configuration Parameter Definitions
	名前(データ型: コンフ・	このセクションではRTコンボ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。
	ジョー 基本! デフォルト値・コンフ.	conf_name0
	アン RTコン 解釈	Delete
▼ Detail	変数名:コンフ・	
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。	単位: コンフ-	
パラメータ名: flipMode	制約条件: コンフ・ ・指定	
est _bR0 liet	・100< ・範囲	の詳細両面にて 刑虐祝 亦物々たじた恐守
	・列写 ・配列	(2)計神画面にて, 空間報, 炙奴石なとで設た
変数名: flipMode	・フィック Widget: コンフ・	
単位:	Step: 設定)	I 名称: flipMode I
制約条件: (-1.0.1)		
Widget: radio		Iデータ型: inf
Step:		
Documentation		デフォルト値: ()
データ名: flipMode		
		I 変数名: flinMode I
■11年の1月1日 (1997) (19		
		制約冬件' (_1 0 1)
単位: なし		「「」「」「」「」「」「」「」「」「」「」「」「」「」」
データ範囲: -1.0.1		Widget radio
0:上下反転したい場合 1:左右反転したい場合 		
咖啡洗什~ -1・ト下左右反射,たい場合		

※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: (值1, 值2, …)
 - ▶ 例) (val0, val1, val2)
 - 配列型: 值1, 值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



■ 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

城 大 学

名

語・環境			
			★ t21
O セクションでは使用する言語を指え O C++ O Python O Java ● Ruby	Èします	Use old build environment.	言語: RTコンボ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、プロファイル内にのみ保存されます
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	
			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
- 言羊糸田'青幸版			
OS Version	Add CPU	Add Delete	













既存のRTCの設定(RTC Profile)を利用した

い場合

①「基本」タブ下部の「インポート」ボタン をクリック

▼ 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
▼ ブロファイル情報のインボート・エクスボート
プロファイル指載のインボートおよびエクスボートを行います。
インポート



RIG Builder - Flip/RIG.xml -	Eclipse SUK		
ファイル(E) 編集(E) ナビゲート(N) 1	実常(A) プロジェクト(E) 実行(B) ウィンドウ(E) ヘルブ(E)		
] 📬 • 🗟 🚔] 🏯] 💁 •] 😂 🛷 •] 🤤 • 🖄 • 🐆 🔶 • •		🔝 🧝 RTC-CANope 🔓 🐲
😫 パッケージ・エクスプロ 🖾 🔍 🗖	> Fig. 8		- B
🖃 🎭 🔽	コンボーネント種類: DataFlow DFSM MultiMode	コノホ = ホノド主・	・STATIC 動きに生成/ADまたれないRTC ・ UNIDUE 計算に生成/ADまたれないRTC ・ UNIDUE 計算に生成/ADまたれるユークなRTC
RTG.xml	REPL Print Control	アクティビティ型:	*COMMOTATIVE 動産化生死の確認すれて
	実行問明: 1000.0		- PERIODIC、実行周期が固定の周期実行型RTC - SPORADIC:実行周期が引固定の周期実行型RTC - EVENTDRIVEN: イベンド配動型RTC
	概要:		RTコンボーネントの種類を指定します。 • DataFlowComponent、周期更行型RTC • FiniteStateMachine: 有限状態運移型RTC • MultiMade: マルチモーや現在O
	RTC Type :	最大インスダンス数:	生成可能なインスタンス数を指定します。制限がない場合はC
	▼ コード生成とパッケージ化	実行型:	実行型を指定します。
	コードの生成およびパッケ 🏀 Finish		コンボーネントアクションの実行周期を[Hz]単位で指定します。 この設定値はデータフロー型コンボーネントのみで有効です。
	コード生成 パッケージ Profile Importが終了しました		RTコンボーネントの簡単な説明を記述します。
	▼ プロファイル情報のイ:		特定機能を実現するRTCの種類を区別する必要がある場合 値が省略された場合には通常のRTCとして解釈されます。
	プロファイル情報のインボ・		
	インボート エクスボー	OK I	設定した情報を基にRTCのスケルトンコードを生成します。 RTCのソースコード、実行用バイナリコードなどを各種形式で기
		インボート: パッケージング:	RtcProfile形式で保存した情報を読み込みます。 設定した情報をRtcProfile形式で出力します。
			e
	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 RTO-CANopen	USB PhP
	BuildView 😒		
		ModuleName	
	1		

■ 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



おわりに

- 第2部では、RTコンポーネント開発とRTコンポーネントを 用いたシステム構築に必要なツールであるRT System Editorの使い方を体験した.
- RTC BuilderやRT System Editorについては、 産総研原
 氏によりブラウザ上で動作するバージョンが開発が進め
 られている.
- http://openrtp.org/rtcbow/index.html
- http://hara.jpn.com/siwiki/_hara/ja/Software/RTSEoW.h tml
- RT System Editorを用いたシステム構築は初期段階での運用には適しているが、実運用段階では、rtshellなどのRTシステムの自動構築を可能にするツールの利用が好ましい。
 - http://openrtm.org/openrtm/ja/node/869

RTCBuilder補足説明





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



生成されたドキュメントの例

flip 1.0.0

メイン・	e−ジ	クラス	ファイ
椿成	構成素引	構成	XUN

クラス Flip

Flip image component. [詳細]

#include <Flip.h>

すべてのメンバー覧

Public メソッド

	Flip (RTC::Manager "manager) constructor
	~Flip () destructor
virtual RTC::ReturnCode_t	onInitialize ()
virtual RTC::ReturnCode_t	onActivated (RTC::UniqueId ec_id)
virtual RTC::ReturnCode_t	onDeactivated (RTC::UniqueId ec_id)
virtual RTC::ReturnCode_t	onExecute (RTC::UniqueId ec_id)

Protected 変数

int	m_flipMode
CameraImage	m_originalImage
InPort< CameraImage >	m_originalImageIn
CameraImage	m_flippedImage
OutPort < CameraImage >	m_flippedImageOut

説明

Flip image component.

InPerthyの入力画像を原規の心Perthyを出力するニンボーネント。 反動の注意軸:RCOコンフィキュレーション増送を用して肌Phodeという名前のパラメータで指定します。 机PModeは、反乱たい、内向に応じて下記のように指定してくたさい。 ・と下反転したい場合、1 ・上下左右反乱したい場合、1

作成するRTCの入出力仕欄は以下のとおりです。

関数

RTC::ReturnCode_t Flip::onActivated (RTC::UniqueId ec_id) [virtual]

データ領域の確保 ・イメージ用メモリの初期化 ・outPortの画面サイズの初期化

RTC::ReturnCode_t Flip::onDeactivated (RTC::UniqueId ec_id) [virtual]

データ領域の解放 ・イメージ用メモリの解放

RTC::ReturnCode_t Flip::onExecute (RTC::UniqueId ec_id) [virtual]

Flip処理 ・新規データのチェック ・InPortの画像データを内部パッファにコピー ・内部パッファの画像データを反転 ・反転した画像データをOutPortにコピー

RTC::ReturnCode_t Flip::onInitialize () [virtual]

コンポーネント自身の各種初期化処理

変数

int Flip::m_flipMode [protected]

画像の反転方法を指定するバラメータ

Name: flipMode flipMode
 DefaultValue: 0

Deraultvalu
 Unit: なし

flip 1.0.0

メインページ クラス ファイル ファイルー覧

C:/GlobalAssist/EclipseTest2/Flip/Flip.h

説明を見る。

00001 // -*- C++ -*-00024 #ifndef FLIP H 00025 #define FLIP_H 00027 #include <rtm/Manager.h> 00028 #include <rtm/DataFlowComponentBase.h> 00029 #include <rtm/CorbaPort.h> 00030 #include <rtm/DataInPort.h> 00031 #include <rtm/DataOutPort.h> 00032 #include <rtm/idl/BasicDataTypeSkel.h> 00033 #include <rtm/idl/ExtendedDataTypesSkel.h> 00034 #include <rtm/idl/InterfaceDataTypesSkel.h> 00035 00036 // Service implementation headers 00037 // <rtc-template block="service_impl_h"> 00038 00039 // </rtc-template> 00040 00041 // Service Consumer stub headers 00042 // <rtc-template block="consumer_stub_h"> 00043 00044 // </rtc-template> 00045 00046 using namespace RTC; 00047 00078 class Flip 00079 : public RTC::DataFlowComponentBase 00080 { 00081 public: 00086 Flip(RTC::Manager* manager); 00087 00091 ~Flip(); 00092 00093 // <rtc-template block="public_attribute"> 00094 00095 // </rtc-template> 00096 and a second second



■ ソリューション中の「PACKAGE」をビルド

城大 学

🚸 Flip – Microsoft	Visual Studio			
ファイル(F) 編集(E)	表示(V) プロジェクト(P) ビルド(B)	デバッグ(D) ツール(T) テスト(S)	ウィンドウ(W) ヘルプ(H)	
🛅 - 🛅 - 🚰 🔒	🖉 👗 🖻 🛍 🤊 - 🖓 - 📮	- 🖳 🕨 Debug 🔹 Win32	- 12	- 💀 🗒
ソリューション エクスプロー	-5 - PACKA + 🕈 🗙			
				4
😡 ソリューション 'Flip' 南一 🐻 ALL BUILD	(8 ブロジェクト)			
E- doxygen_doc				
l⊞⊷ 🚰 flip tincomp				τ, τ <mark>ε</mark>
E In PACKAGE	ี ยั <i>ม</i> ห(U)			U a
E- ZERO_CI	リビルド(E)			
	クリーン(N)			,
	プロジェクトのみ(J)			でした
	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			<u>Ň</u>
	プロジェクトのビルド順序(I)			
	カスタム ビルド規則(B)			
	ツールのビルド順序(し)			
	追加(D)			
	参照(F)			
🔩	Web 参照の追加(E)…			
出力 &	、 クラス ダイアグラムで表示(V)			+ ₽ ×
出刀元の表示(S):	スタートアップ プロジェクト(こ設定(A)			
1>try deleting	デバッグ(G) ▶	BN. SN.		
1>If you insta 💑	切り取り(T)	iling run,	mi/doxygen/ntmi/dir_zaz3818C2D26C3Ce2	16CC8D3104332et_0ep.m
1>try deleting 1>error: probl	1 貼り付(F)	en. CclipseRTMTest/Flip/build/doc/ht	ml/doxygen/html/dir_309d483ef0f84f1cf	e00009dc05bbe0a_dep.m
1>If you insta 🗙 1>try deleting	削除(∨)	iling run, en.		
1>ビルドログは 1>doxygen_doc	名前の変更(M)	st¥Flip¥build¥doc¥doxygen doc.di	<u>r¥Debug¥BuildLog.htm</u> ″ に保存されました	č.
En	プロジェクトのアンロード(L)) スキップ =======		
ا	エクスプローラでフォルダを開く(X)			▼
📴 コード定義ウィン 🗟) プロパティ(R)			
		-		

 ●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>





■ 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」	②FRtcBu	ilder」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
実行(R) ウィンドウ(M) ヘルブ(H) 新規ウィンドウ(M) 新規工ディター(E) パースペクラィブを閉(Q) ・ パースペクラィブを閉(Q) ・ パースペクティブのカスタマイズ(Z) パースペクティブの別と(保管(A) パースペクティブの別と(保管(A) パースペクティブの別じる(保管(A) パースペクティブを閉じる(Q) すべてのパースペクティブを閉じる(L) ナビビモニュウン(G) ・	 Size アメリタースプ ー ト <l< th=""><th>RtcBuilder デー対型: IDL File Directories</th><th>「一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一</th><th>にて場所を指定 「アレクラスス」 ・ ポポ ・ ローボ ・ ロー ・ ローボ ・ ロー ・ ロー ・ ・ ロー ・ ロー ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・</th></l<>	RtcBuilder デー対型: IDL File Directories	「一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	にて場所を指定 「アレクラスス」 ・ ポポ ・ ローボ ・ ロー ・ ローボ ・ ロー ・ ロー ・ ・ ロー ・ ロー ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
うこう シュン (m) ・ (つ ワーキング・セット (W) ・ 設定(P) ・	0	 	/復元①) / 適用(A) / OK _ キャンセル /	フォルボ: マイドキュシント の) 適用(の) 新にんいきゅり多の作気(の) 一の キャンセル () ーの キャンセル

 ※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-aistIこ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/





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



③FlipComp¥¥Debug内の FlipComp.exeを起動



②VisualStudioを用いたビルド

	G++ 2008 Expri	ess Edition	and the second		
ファイルモン 編集(日) 表示(1) 、	709:010 21	ドロ ティックロ ウインドウ	8 AL78		
3-3-3-4 N N X 3	3四 7 - 四	5 993-9308%F(B) F7	* 🧐 W2<	· Q C 3 X 8 2 · ,	
国家をいう使任う。	9 9 9	シリューションのリビルド(B) Cbi+Alt+F7			
ソリューション エクスプローラ 🔷 📮 🗧	USBC.	ソリューションのグリーン(2)			
0	00-10	annenene en en	~		
💭 UU1-919 (USBCamera_vc8) (10 // ·	通行の通知などのような	501		
B Blacker Filer	7#1	運動範囲のカーン他			
🗉 🔛 Source Files		/አቃ ይሉዮው			
USBCamera.cpp	1 1	積成マネージャ(2).			
Header Files	* \$101	1			
IE Source Files		Transferration and the			
	\$inclus	de "USBCamera.h"			
	// Modu	ale specification			
	static	const char# usbcamers_spec[] =			
	1	enlanentation id" "UtBCasers"			
	11 *t)	prementation_id', 'USBCamera', (pe_name', 'USBCamera',			
	de	escription", "USB Camera c	omponent.",		
		endor", "AIST",			
	Ca ac	tivity type", "PERIODIC",	ing',		
	"ki	ind", "DataFlowComp	onent",		
	11	inguage", "C++",			
	11	Configuration variables			
A STREET OF DESTREET OF DESTREET	2 °cr	onf.default.deviceNumber*, "0".			
dina - latina (dina a					
14/75					
出7) 山力あるまま(5)		a training and the train			•





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

尊

e aettings Help					
- 1. C	and the design of the second				
/work/doxygen	Select				
ep 2: Contigure doxygen using the Wiza	rd and/or expert tab, then switch to the Bun tab to generate the documentation				
Wizard Expert Bup					
fanics					
Project	Provide some information about the project you are documenting				
- Mode Output	Project name: USBCamera				
Diagrams					
	Project synopsis:				
	Project version or id:				
	Project logo: Select				
	Specify the directory to scan for source code				
	Saura and directory C (with (with more 100 (USD) among) Salart				
	builde code directory. C/work/workspace199/d/bbcamera				
	Scan recursively				
	Specify the directory where doxygen should put the generated documentation				
	Destination directory: C:/work/workspace199/USBCamera Select				
	Province				
	Elevine Inde				

③「Output」セクションにて、html出力を指定

ep 1: Specify the working directory from	which doxygen will run	
/work/doxygen	Sele	ct
ep 2: Configure doxygen using the Wizar	d and/or Expert tab, then switch to the Run tab to generate the documentation	_
Wizard Expert Bun		
Topics		
Project	Select the output format(s) to generate	
- Mode	HTML	
Diagrama	O ship UTM	
Cragrams	I pain HTML	
	 with navigation panel 	
	 prepare for compressed HTML (chm) 	
	With search function	
	Unange color	
	LaTeX	2
	 as intermediate format for hyperlinked PDF 	
	 as intermediate format for PDF 	
	 as intermediate format for PostScript 	
	Man pages	
	Rich Text Format (RTF)	
	T XMI	
	Previous Next	

(2)「Mode」セクションにて, 出力内容,使用言語を指定

orop 1. opecny the working directory noninvinent	uoxygen wiii run
C:/work/doxygen	Select.
Step 2: Configure doxygen using the Wizard and/ Wizard Expert Run	or Expert tab, then switch to the Run tab to generate the documentation
Topics	Select the desired extraction mode
Mode	Documented entities only
Diagrams	All Entities
	✓ Include cross-referenced source code in the output
	-Select programming language to optimize the results for
	[Optimize for C++ output]
	O optimize for othogs output
	O Optimize for Java or C≢ output
	O Optimize for C or PHP output
	O Uptimize for Fortran output

③「Run」タブにて、「Run doxygen」を実行



RTSystemEditor補足説明







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ミドルウエア:OpenRTM-aistや他のミドルウエア、ミドルウエ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。





昼

大

タイプ	登録数
RTコンポーネント群	638
RTミドルウエア	29
ツール	39
仕様·文書	4
ハードウエア	30

名城大学



■ 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

				🚟 💻 💿 🗰 Hell	o My Account Help Contact Sign Out
MIDDLEWARE Open	RTM-aist				検索
ホーム ダワ	ウンロード ドキュメント コミュニ	ティ 研究・開発 ブロ	ジェクト		
Google Translate	ホーム >> プロジェクト >> RT:	コンポーネント >> 顔検出=	レポーネント		Issues
	顔検出コンボーネント				To avoid duplicates, please
ナビゲーション	顔検出コンボーネント				issue.
 ホーム ダウンロード 	投稿者:s-kurihara 投稿日8	寺:火, 2011-11-15 18:20			
▶ ドキュメント	同合先(メールアドレス): ope	nrtm@openrtm.org	Capture In see		検索 高度な検索
 * コミュニティ ▶ 研究・開発 	INPORTING/CJICATICE画家が 出結果を描画した画像をOutF	う顔を検出し、ハノ」画家にあ Portから出力するとともに、			All issues O open, O total
▶ プロジェクト	のOutPortでは、顔の位置や ンポーネントです。	険出した顔の数を出力する 。			Bug reports O open, O total
リンク	詳細は、 こちら をご覧ください。	5			Subscribe via e-mail Issue statistics
	Downloads				
OpenHRP3	バージョン D	ownloads		C ET RS	ユーザロクイン
動力学シミュレータ	0.1	Download (17.44 MB)	<u>2011-11月-15</u>	Notes	ユーザー名: *
OpenHRI	° View all releases 🔊		ALE		
対話制御コンポーネント群	Resources 。ホームページ	Development ° View pendir	ng patches		パスワード: *
OpenRTP	Read documentation				
統合開発ブラットフォーム	Look at screenshots				ログイン
	OS: Windows				• アカウントの作成
OpenINVENT	言語: C++				• バスワードの再発行
移動ロボット用コンボーネント群	UpenRIM ver.: 1.1 ☆☆☆☆☆				

名城大学



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





■ 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

Configuration View 🛛 🥂 Manager Control View 🔭 Composite Component View 🤾 Execution Context View 🧮 RT Log View								
Compor	nentName: ConfigSample0	ConfigurationSet: default		編集				
active	config	name	Value					
•	default	double_param0	10	適用				
	we de 0	double_param1	-0.99	And School 1				
	modeu	int_param0	150	キャンセル				
0	mode1	int_param1	1					
		str_param0	default					
		str_param1	default set in conf file					
		vector_param0	0.0,0.1,0.2,0.3,0.4,0.5,0.6					
		-						
,								
21	製 追加 削除		追加					

- ※「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, スライドバーを操作する, 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

Configuration	2
	変更あり
default mode0	mode1
ConfigurationSet	: default
double_param0	
double_param1	-0.99
int_param0	200 ▲ 割約条件[0<=x<=150]を満たしていません。 ↓
int_param1	F
str_param0	
str pr	default set in conf file
削約違反	0.0.1,0.2,0.3,0.4,0.5,0.6
0	「 Apply キャンセル
V	
	即時反映

名城大学



• 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



■ 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

城 大 学

名

Configuration 🕅 Ma	nager Con 🥂 Compos	ite C RT Execution C	🔀 🛛 🕅 RT Log View 🗆 🗖
component: ConsoleOut	0		通用
Execution Context	rate: 1000.2		
owned0 participate0	Name	Value	スタート
	id kind state component_state	0 PERIODIC RUNNING INACTIVATE	ストップ アクティブ化
	owner participants	ConsoleOut0 0	非アクティブ化
			<u> アタッチ</u>

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





■ RTコンポーネントの新規インスタンスの生成

Configuration View	Manager Control View 🛛 💦 Composite Component View 💦 RT Log View	- 0
Loadable Modules Loadable Modules Loaded Modules Active Components Create Fork Shutdown	module	Load
	URL:	



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

ype :	ConsoleIn	*
	instance_name=in1&conf.mode1.input_mode=ANY&conf.mode1.inp ut_cycle=1000	-

RIDDLEWARE

■ 選択したRTCから収集したログ情報を一覧表示

Cor	nfiguration 🕅	Manager Con	T Compos	ite C 🕅 Execut	ion C 🔀	RT Log View 🛛	
	component	time	level	component	logger	message	^
	Notify0	2011-04-28	ERROR	Notify0	RTC	test log!	
	Notify1	2011-04-28	ERROR	Notify1	RTC	test log!	
-		2011-04-28	ERROR	Notify0	RTC	test log!	=
		2011-04-28	ERROR	Notify1	RTC	test log!	
		2011-04-28	ERROR	Notify0	RTC	test log!	
		2011-04-28	ERROR	Notify1	RTC	test log!	
		2011-04-28	ERROR	Notify0	RTC	test log!	
		2011-04-28	ERROR	Notify1	RTC	test log!	
		2011-04-28	ERROR	Notify0	RTC	test log!	
		2011-04-28	ERROR	Notify1	RTC	test log!	
		2011-04-28	ERROR	Notify0	RTC	test log!	
		2011-04-28	ERROR	Notify1	RTC	test log!	
<		2011-04-28	ERROR	Notify0	RTC	test log!	
		2011-04-28	ERROR	Notifv1	RTC	test log!	×
ERRO	DR 🔽	<][:	

※近日機能追加予定

● ログ収集の開始/停止

城 大 学

名

Noti	I Activate (<u>A</u>) Deactivate (<u>W</u>) Reset (<u>Q</u>) Exit (<u>E</u>) Start (7)	Ctrl+Alt+A Ctrl+Alt+D	No ¹	ť	Activate (<u>A</u>) Deactivate (<u>W</u>) Reset (<u>Q</u>) Exit (<u>E</u>) Start (<u>7</u>)	Ctrl+Alt+A Ctrl+Alt+D
	Stop 🖄				Stop ⊘	
	複合コンポーネントの作成 エディタで開く 複合コンポーネントを解除				複合コンポーネントの作成 エディタで開く 複合コンポーネントを解除	
	ログ収集を開始 Create Profile Table	k			ログ収集を停止 Create Profile Table	4

ログ情報のフィルタリング

	time	level	component	logger	message		^
	2011-04-28	ERROR	Notify0	RTC	test log!		
	2011-04-28	ERROR	Notify0	RTC	test log!		
	2011-04-28	ERROR	Notify0	RTC	test log!		
	2011-04-28	ERROR	Notify0	RTC	test log!		_
component o	2011-04-28	ERROR	Notify0	RTC	test log!		
L. Component rager Con	2011-04-28	ERROR	Notify0	RTC	test log!		
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!		
Notify1 me	2011-04-28	ERROR	Notify0	RTC	test log!		
1-04-28	2011-04-28	ERROR	Notify0	RTC	test log!		
	2011-04-28	ERROR	NotifyU	RTC	test log!		
component 11-04-2 ti	me lev	vel Co	mponent	logger	message	^	
Notify0 1-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
1000000 1-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		~
Notify1	11-04-28 ER	ROR	tify1	RTC	test log!		
2011 04 2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
2011-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
2011-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
2011-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
2011-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
2011-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
TNEO 11-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
111-04-2 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
ERROR N 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
WARN 20	11-04-28 ER	ROR No	tify1	RTC	test log!		
INFO 20	11-04-28 ER	ROR No	tifv1	RTC	test log!	~	
NORMAL						>	
DEBUG 💌 🦳							8

■ 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

亭

大

SequenceOutCompo Sequen

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

🗲 New Co	mposite Component 🔀
Manager :	localhost.localdomain/manager
Name :	CompositeSample
Type :	PeriodicECShared
Path :	192.168.1.212
Port :	Sequence b/Component() Short ▼ <cwrs)< td=""> Sequence b/Component() Loug ▼<cwrs)< td=""> Sequence b/Component() Louble ▼<cwrs@< td=""> Sequence b/Component() Louble ▼ Sequence b/Component() Louble ■ Sequence b/Component() Louble ■ Sequence/Loupenent() Float ■ Sequence/Locomponent() Float ■</cwrs@<></cwrs@<></cwrs@<></cwrs@<></cwrs@<></cwrs)<></cwrs)<>
	OK ++>tell

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
Туре	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
ᄽᆂᇠᅿᄻ	。 自体合う、ポーク、トリ 如と快結され ているDa

※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





■ 公開ポートの設定

学

● 複合コンポーネントビュー

城大

>onfigur	ation View Manager Control View	Composite Component View 🛛 🥂 RT	Log View	- 0)
compon	ent: PeriodicECSharedComposite4	type: PeriodicECShared		
_	component	port	A	適用
	SequenceInComponent0	Short		200713
	SequenceInComponent0	Long		キャンセル
	SequenceInComponent0	Float		
	SequenceInComponent0	Double		
	SequenceInComponent0	ShortSeq		
	SequenceInComponent0	LongSeq		
	SequenceInComponent0	FloatSeq		
	SequenceInComponent0	DoubleSeq		
	SequenceOutComponent0	Short		
	SequenceOutComponent0	Long		
	SequenceOutComponent0	Float		
	SequenceOutComponent0	Double		
	SequenceOutComponent0	ShortSea	–	
_				

※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

嘭

城大







- 接続ー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

🖨 設定	
フィルター入力	接続 (→
 →般 Ant Java RT Name Service View RT Repository View RT System Editor ATフライン・エディタ オンライン・エディタ 接流 表示色 Velocity UI 	状態通知オブザーバ ハートビート有効化: ▼ ハートビート受信間隔: 1.0 sec ハートビート受信回数: 3 回 接続 接続周期: 1000 ms (0≦接続周期≦1000000 同期しない場合は0)

- 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】

尊

城大

● 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

€ 設定	
フィルター入力	接続 ♀ → →
 → 一般 ➡ Ant ➡ Java ➡ RT Name Service View ➡ (副期) ➡ 同期 	接続 接続周期: 1000 ms (0≦接続周期≦1000000 同期しない場合は0) デフォルトポート番号: 2809 (0≦ポート番号≦65535)
n RT System Editor n RtcBuilder	

- 「RT Name Service View」ー「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

€ 設定		<u>_ 0 ×</u>
フィルター入力	同期	⇔ • ⇒ •
 → 一般 Ant → Java ➡ RT Name Service View ➡ 持続 ➡ RT Repository View ➡ RT System Editor ➡ RtcBuilder 	「同期 タイムアウト待ち時間: 1000 ms (0≦タイムアウト≦9999)	

- 「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







- 「RT System Editor」-「アイコン」【表示アイコン】
 - RTC内に表示するアイコンを指定可能

学

统 大

• カテゴリ単位, RTC名称単位で設定が可能



