

RTミドルウェア講習会

日時:2014年5月25日

場所:富山国際会議場



RTミドルウェア講習会(予定)

| | |
|-----------------|---|
| 10:00- 10:50 | 第1部(その1):インターネットを利用したロボットサービスとRSiの取り組み(最新動向) |
| | 担当:成田 雅彦(産業技術大学院大学) |
| | 概要:プラットフォーム化したロボットの各種機能モジュールのオープン化が進む一方、ネット企業によるロボット企業の買収などの気になる変化も続いています。本講演では、インターネットやクラウドとロボットとの連携、RSi(ロボットサービスイニシアティブ)の実証や試作などの取り組みを紹介し、今後を展望したいと思います。 |
| 11:00- 11:50 | 第1部(その2):OpenRTM-aistおよびRTコンポーネントプログラミングの概要 |
| | 担当:原 功(産業技術総合研究所) |
| | 概要:RTミドルウェアはロボットシステムをコンポーネント指向で構築するソフトウェアプラットフォームです。RTミドルウェアを利用することで、既存のコンポーネントを再利用し、モジュール指向の柔軟なロボットシステムを構築することができます。RTミドルウェアの産総研による実装であるOpenRTM-aistについてその概要およびRTコンポーネントの機能やプログラミングの流れについて説明します。 |
| 11:50- 12:00 | 質疑応答・意見交換 |
| 12:00- 13:00 | 昼食 |
| 13:00- 15:00 | 第2部:RTコンポーネントの作成入門 |
| | 担当:坂本 武志(株式会社グローバルアシスト) |
| | 概要:RTシステムを設計するツールRTSystemEditorおよびRTコンポーネントを作成するツールRTCBuilderの使用方法について解説するとともに、RTCBuilderを使用したRTコンポーネントの作成方法を実習形式で体験していただきます。 |
| 15:00- 16:45 | 第3部:プログラミング実習 |
| | 担当:原功, Geoffrey Biggs(産業技術総合研究所), 坂本武志(株式会社グローバルアシスト) |
| | 概要:OpenRTM-aistを利用して画像処理コンポーネントを作成します。 また、USBメモリ起動のコンポーネントを利用した利用例について体験して頂きます。 |

第3部 プログラミング実習

(独)産業技術総合研究所
知能システム研究部門
ディペンダブルシステム研究グループ
原 功

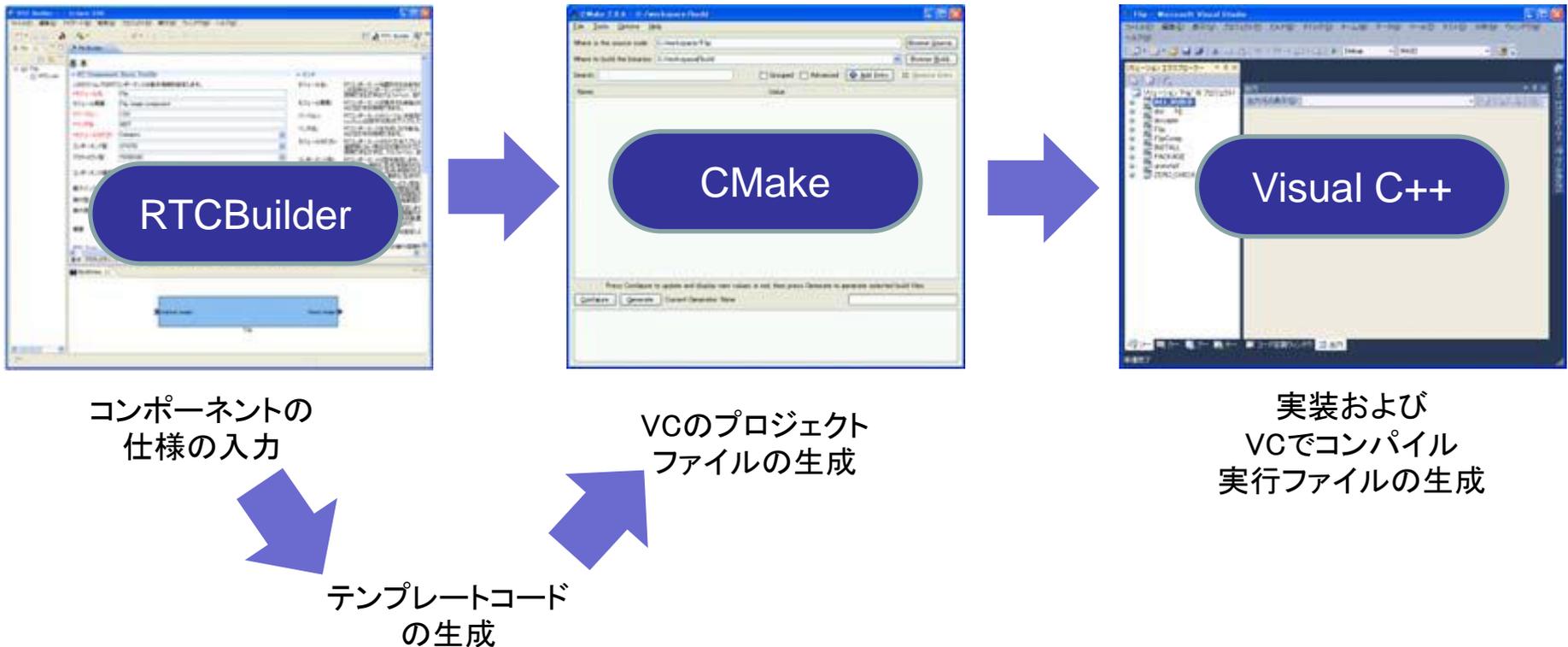


概要

- Filpコンポーネントの作成
- RT Middleware 体験用チュートリアル用
USBを使ってRTCの操作と開発体験

Flipコンポーネントの作成

- 第2部で説明したRTCBuilderを使ってFlipコンポーネントを実装する



Flipコンポーネント

- OpenCVが提供する機能(**cvFlip関数**)を使った画像変換フィルタ
 - 入力ポート: OpenCVCameraやDirectShowCamなどのカメラコンポーネントから送られてくる画像データ
 - 出力ポート: CameraViewコンポーネントや他のフィルタコンポーネントへ送る画像データ
 - コンフィグレーション: Flipの方向を選択できるようにする

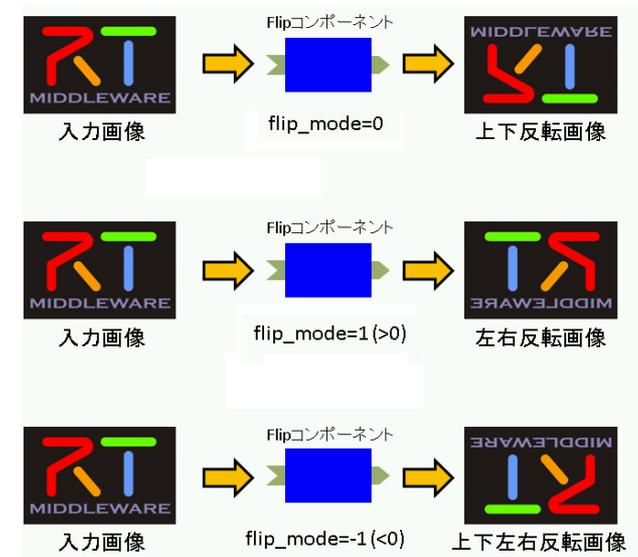
Flipコンポーネントの概要

• cvFlip関数

– cvFlip 関数は、OpenCVで標準的に用いられているIplImage型の画像データを垂直軸 (左右反転)、水平軸 (上下反転)、または両軸 (上下左右反転) に対して反転させます。

```
void cvFlip(IplImage* src, IplImage* dst=NULL, int flipMode=0);  
#define cvMirror cvFlip
```

src 入力配列
dst 出力配列。もしdst=NULLであれば、srcが上書きされます。
flipMode 配列の反転方法の指定内容:
flipMode = 0: X軸周りでの反転(上下反転)
flipMode > 0: Y軸周りでの反転(左右反転)
flipMode < 0: 両軸周りでの反転(上下左右反転)

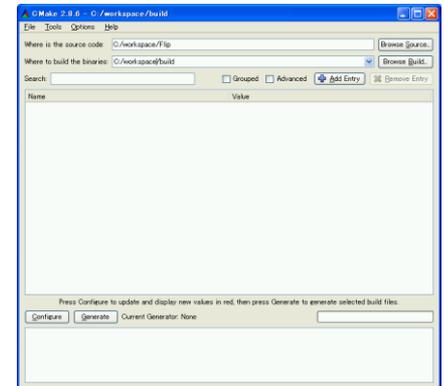


Flipコンポーネントひな形の作成

- RTCBuilderを起動してコンポーネントの基本データを入力
- データポートの情報を入力
- コンフィギュレーションパラメータの設定
- 実装言語を選択
- ひな形を作成

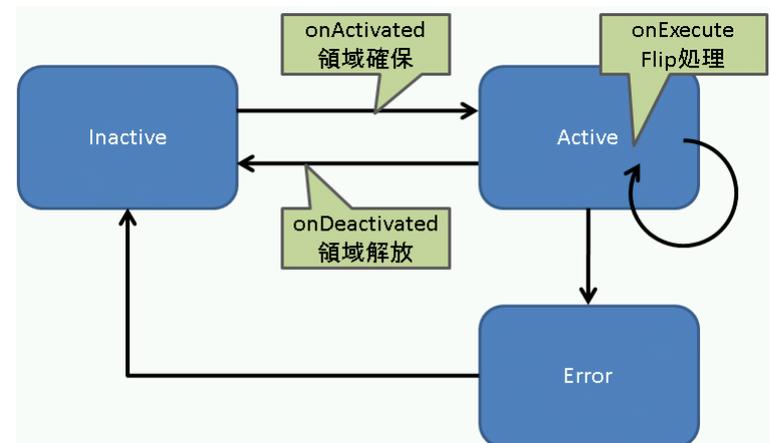
Flipコンポーネントのソリューション生成

- RTCビルダで作成したひな形からVisual C++のソリューションファイルを生成する
- Cmake (gui)を起動し、生成したひな形のフォルダ内のCMakeList.txtを選択
- 実装フォルダ名を入力
- Configuration → Generate の順でソリューションを生成



Flipコンポーネントの実装

- CMakeで生成したソリューションファイルをVisual C++で開く
- Flip.cppの実装
 - onInitialize: コンポーネントの初期化
 - onActivate: データ保存領域の確保
 - onDeactivate: データ保存領域の解放
 - onExecute: Flip処理



Flipコンポーネント: onInitialize

- ヘッダファイルの編集(OpenCV用ヘッダの追加)
- 画像保存用メンバー変数の追加
- その他は、基本的にコンポーネントクラスの初期化を行うので特に追加記述なし

Flipコンポーネント: onActivate

```
RTC::ReturnCode_t Flip::onActivated(RTC::Uniqueld 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;
}
```

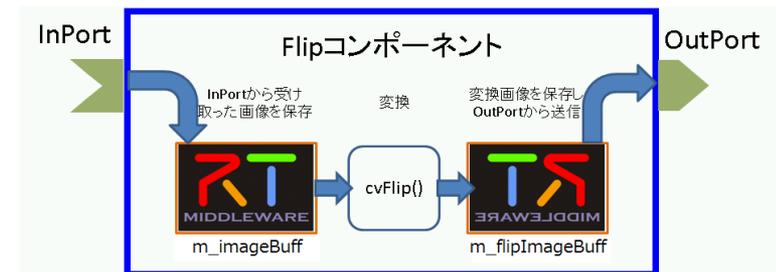
Flipコンポーネント: onDeactivate

```
RTC::ReturnCode_t Flip::onDeactivated(RTC::Uniqued 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;
}
```

Flipコンポーネント: onExecute

- 新しい画像データのチェック
 - InPortデータの読み込み
 - 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
 - InPortのイメージサイズが変更された場合の処理
 - Flip処理用イメージ用メモリの確保
 - InPortの画像データをIpImageのimageDataにコピー
 - 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
 - 画像データのサイズ取得
 - 反転した画像データをOutPortにコピー
 - 反転した画像データをOutPortから出力する。



Filpコンポーネントの実行

The screenshot shows the RT System Editor interface. The main window displays a system diagram with three components: USBCameraAcquire0, Filp0, and USBCameraMonitor0. The Filp0 component is highlighted with a dashed box. The Properties window on the right shows the configuration for Filp0, including its path, instance name, type, and execution state. The Configuration window at the bottom shows the configuration for the Filp0 component, with the flipMode set to 1.

| active | config | name | Value |
|-------------------------------------|---------|----------|-------|
| <input checked="" type="checkbox"/> | default | flipMode | 1 |

RT Middleware 体験用チュートリアル

- 2013年度NEDO講座の成果の一つ
- USB等に保存し、OpenRTMをインストールせずに既存のRTCを使った体験システム
- GitHub.comに公開
 - 誰でもダウンロードして利用可能
 - Git機能を使った自己アップデート

チュートリアルの内容

• デモの体験

- 画像処理 (OpenRTM-aist, OpenCV)
- 音声命令による動画再生 (OpenHRI)
- ロボットモーションの新規作成 (Choreonoid)
- 音声命令によるモーションの再生 (OpenHRI, Choreonoid)
- 状態遷移モデル変更による, 新たなモーション実行 (SEATSAT)

• RTC 開発の体験

- 音声認識されたキーワードのログを取得し, 時刻とともにファイルに保存する ロガー RTC の開発
- キーボードからの入力によって Choreonoid のモーションを呼び出すことが出来る RTC の開発

チュートリアルの実行

- RTM体験用チュートリアルを入れたUSBを接続
- USBメモリのトップフォルダにある “index.htm” をInternet Explorer 開く

openrtm_tutorial 0.4 documentation »
索引

目次

RT Middleware 体験用チュートリアル用 USB

- デモの体験
- RTC 開発の体験

このページ

ソースコードを表示

RT Middleware 体験用チュートリアル用 USB

デモの体験

- (デモ 1) 画像処理 (OpenRTM-aist, OpenCV)
- (デモ 2) 音声命令による動画再生 (OpenHRI)
- (デモ 3) ロボットモーションの新規作成 (Choreonoid)
- (デモ 4) 音声命令によるモーションの再生 (OpenHRI, Choreonoid)
- (デモ 5) 状態遷移モデル変更による, 新たなモーション実行 (SEATSAT)

RTC 開発の体験

- (開発 1) 音声認識されたキーワードのログを取得し, 時刻とともにファイルに保存する ロガー RTC の開発
- (開発 2) キーボードからの入力によって Choreonoid のモーションを呼び出すことができる RTC の開発

openrtm_tutorial 0.4 documentation »
索引

© Copyright 2014, NEDO, TORC. このドキュメントは Sphinx 1.1.3 で生成しました。