

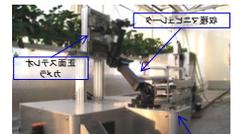
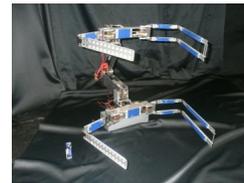
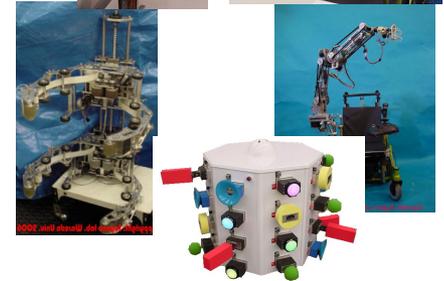
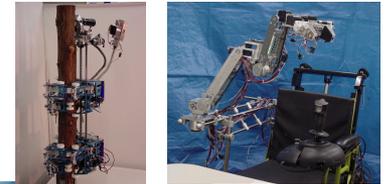
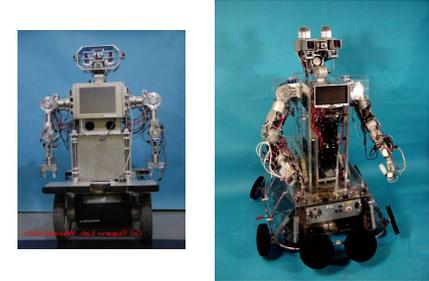
RTMコンテスト作品自慢

フリーランス

菅 佑樹

自己紹介

- 菅 佑樹 (すが ゆうき)
 - 1979年7月31日生まれ (33歳)
 - 早大機械工学科卒 (博士 (工学))
 - 専門: ロボット
 - エンターテイメント, コミュニケーション
 - GA, NNなどの機械学習
 - 林業支援
 - 福祉



ステレオカメラ



パンチルトヘッド



その他

12/08/03



探索用ロボット



小型人工知能ロボット



ロボット用アーム

YUKI SUGA (REVAST CO., LTD)

RTMコンテスト作品の自慢(笑)

- 2008年度
 - RTミドルウェアを用いた名刺受け機能付きマスコットロボットの開発
- 2010年度
 - RTC-scilab -RTミドルウェアとscilabを繋ぐtoolbox-
- 2011年度
 - Arduinoを使ってRTCを開発 RTno

RTミドルウェアを用いた 名刺受け機能付きマスコット ロボットの開発

○菅 佑樹(早大)

坂本義弘(ナレッジサービス)

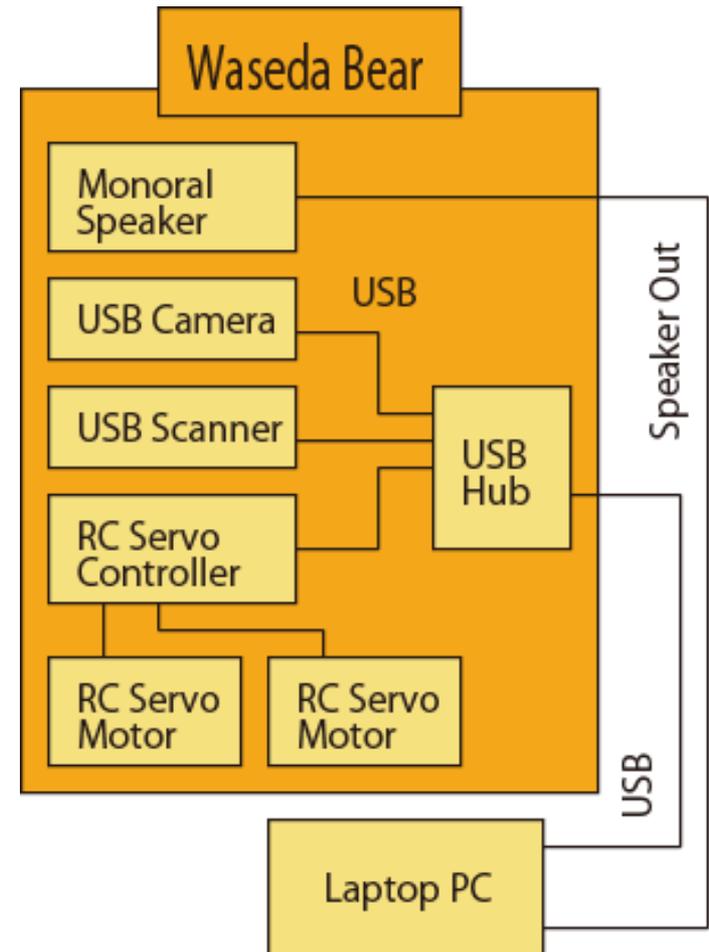
開発概要

- 名刺受け機能付きマスコットロボット
 - 各種展示会などで使用
 - 頭部カメラで首動作
 - 口蓋内スキャナで名刺認識
 - 名刺画像からお客様の芳名認識
 - お客様の芳名読み上げ & 挨拶
 - RTミドルウェア使用
 - ソフトウェアの再利用性向上
 - 顧客の要望に応じてシステムを増設

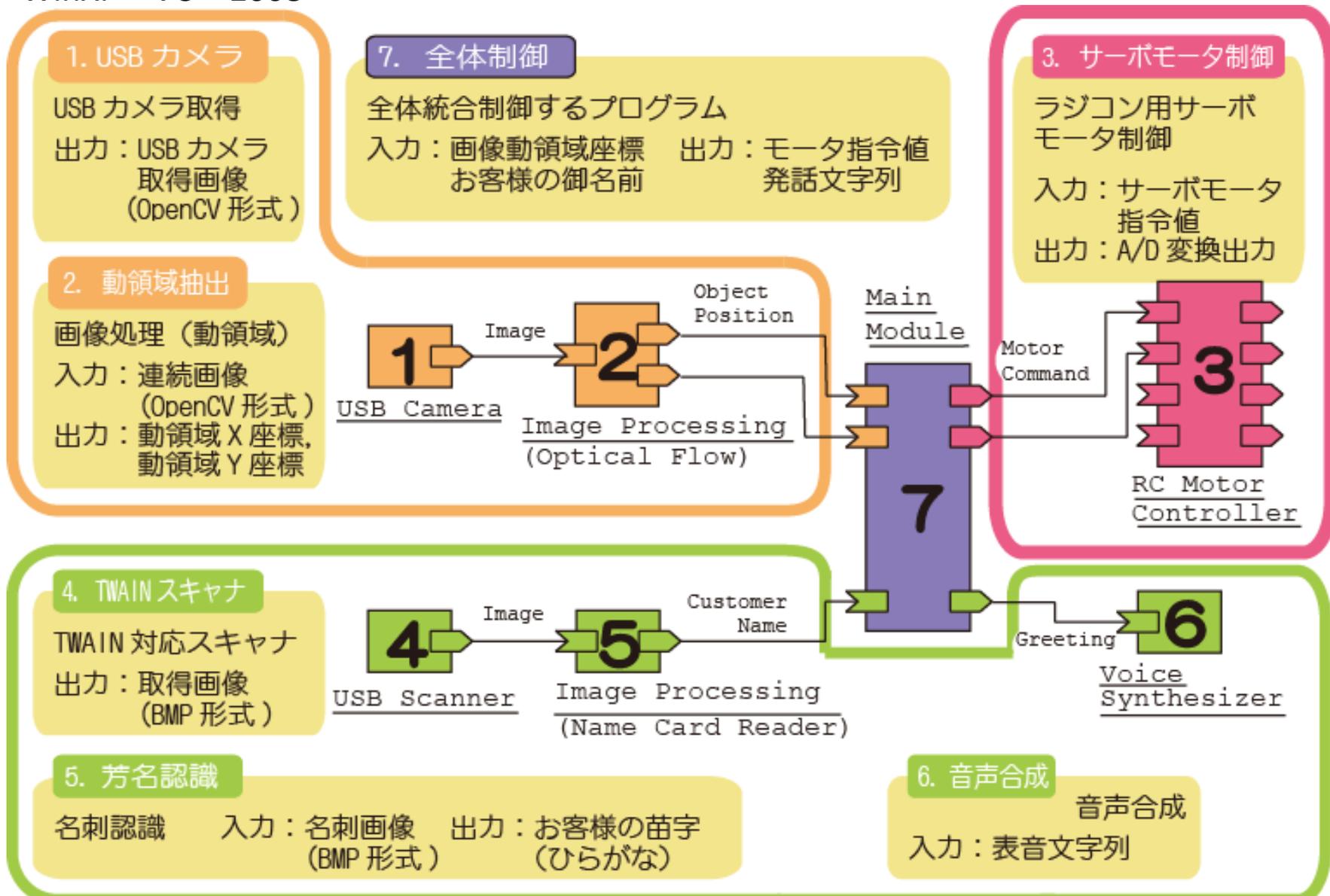


ハードウェア

- Laptop PCに各種I/OボードをUSB接続
 - 小型カメラ
 - RCサーボモータ制御
 - 首フリ動作
 - 小型スキャナ
 - 名刺受け



* WinXP + VC++2008



1. USB カメラ

USB カメラ取得
出力：USB カメラ
取得画像
(OpenCV 形式)

2. 動領域抽出

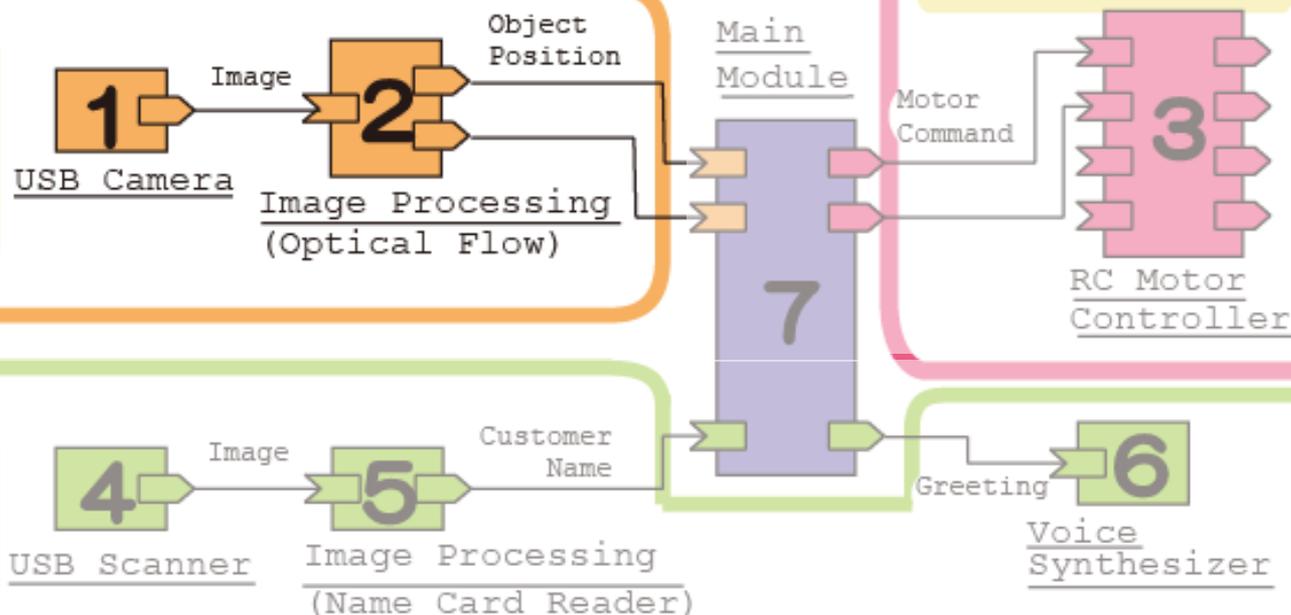
画像処理（動領域）
入力：連続画像
(OpenCV 形式)
出力：動領域 X 座標,
動領域 Y 座標

7. 全体制御

全体統合制御するプログラム
入力：画像動領域座標 出力：モータ指令値
お客様の御名前 発話文字列

3. サーボモータ制御

ラジコン用サーボ
モータ制御
入力：サーボモータ
指令値
出力：A/D 変換出力



4. TWAIN スキャナ

TWAIN 対応スキャナ
出力：取得画像
(BMP 形式)

5. 芳名認識

名刺認識 入力：名刺画像 出力：お客様の苗字
(BMP 形式) (ひらがな)

6. 音声合成

音声合成
入力：表音文字列

USBカメラ & 画像処理コンポーネント

• USBカメラコンポーネント

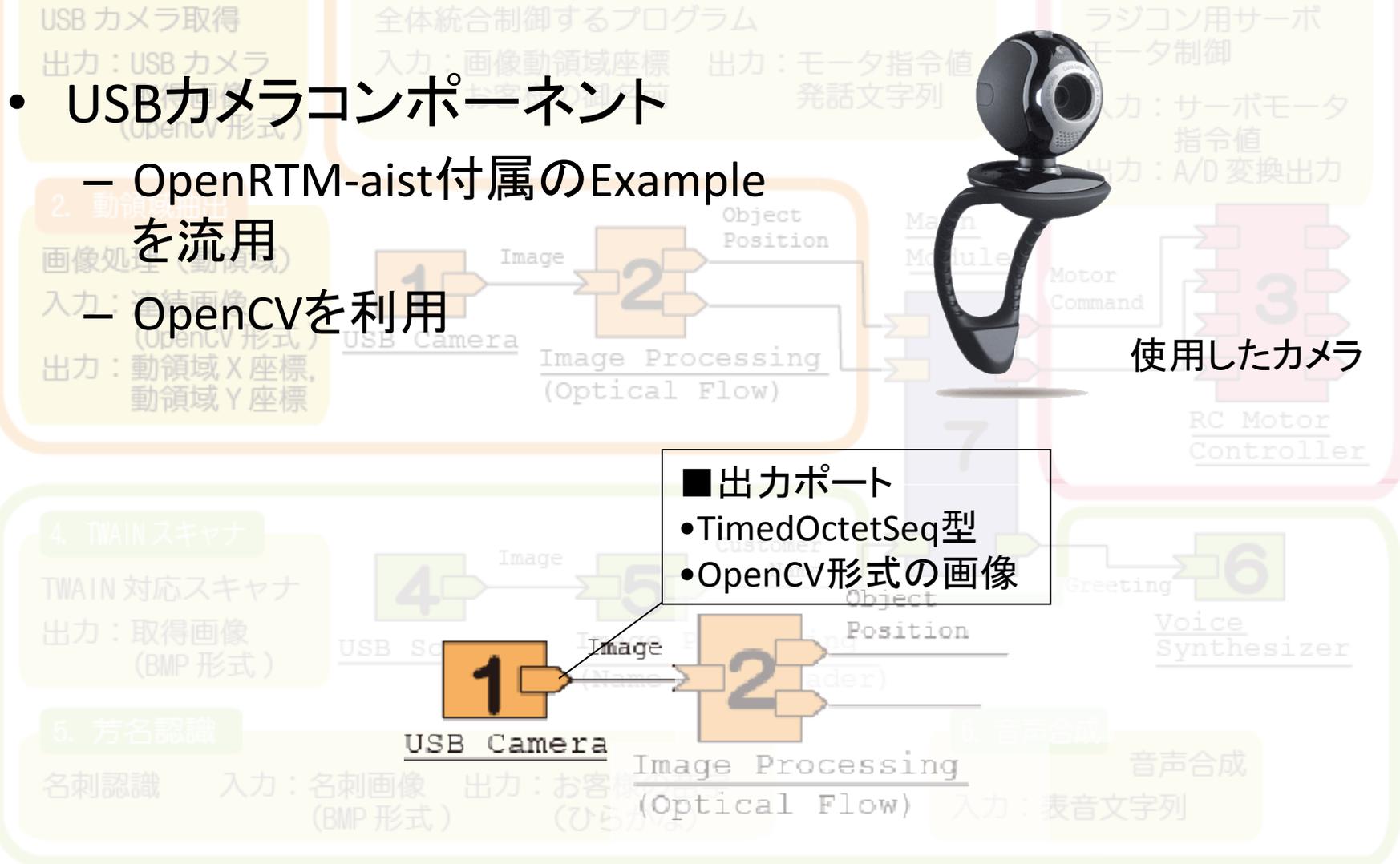
– OpenRTM-aist付属のExample
を流用

– OpenCVを利用



使用したカメラ

■出力ポート
 •TimedOctetSeq型
 •OpenCV形式の画像



USBカメラ & 画像処理コンポーネント

- 画像処理コンポーネント

- OpenCVを利用した画像処理コンポーネント

- 大阪大学 田窪氏のを改変

■入力ポート

- TimedOctetSeq型
- OpenCV形式画像

■出力ポート

- TimedDouble型
- 画像内の動領域重心座標(X)

■出力ポート

- TimedDouble型
- 画像内の動領域重心座標(Y)

1. USB カメラ

USB カメラ取得
出力：USB カメラ
取得画像
(OpenCV 形式)

2. 動領域抽出

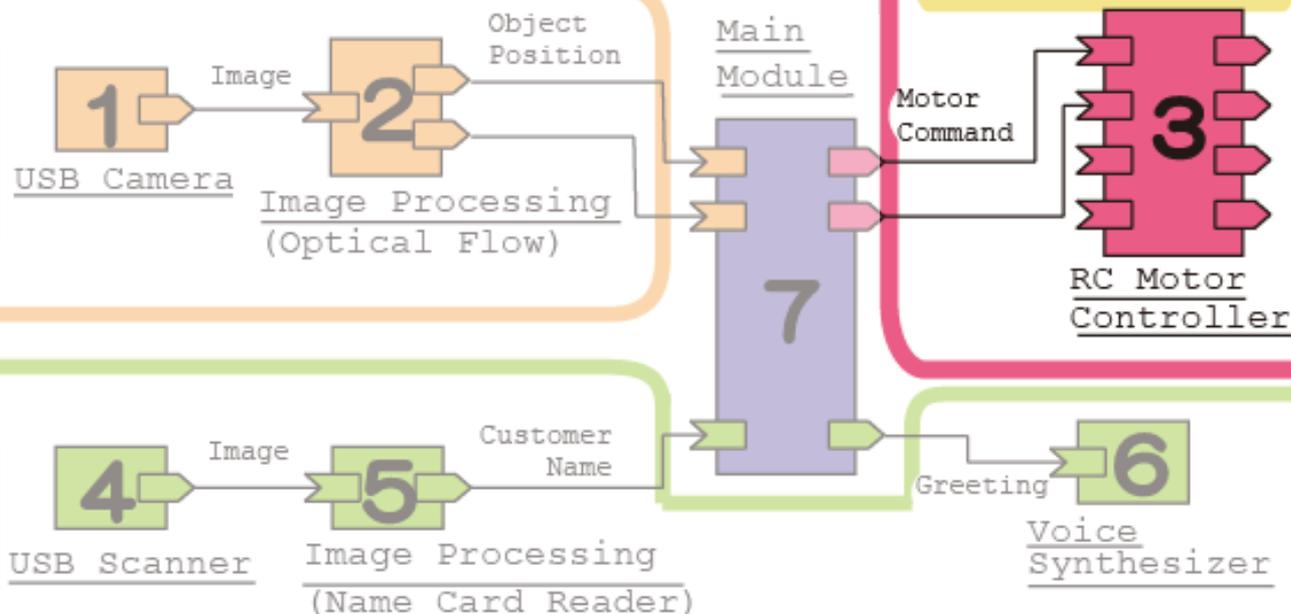
画像処理（動領域）
入力：連続画像
(OpenCV 形式)
出力：動領域 X 座標,
動領域 Y 座標

7. 全体制御

全体統合制御するプログラム
入力：画像動領域座標
お客様の御名前
出力：モータ指令値
発話文字列

3. サーボモータ制御

ラジコン用サーボ
モータ制御
入力：サーボモータ
指令値
出力：A/D 変換出力



4. TWAIN スキャナ

TWAIN 対応スキャナ
出力：取得画像
(BMP 形式)

5. 芳名認識

名刺認識
入力：名刺画像
(BMP 形式)
出力：お客様の苗字
(ひらがな)

6. 音声合成

音声合成
入力：表音文字列

サーボモータ制御コンポーネント

- RCサーボドライバ「iMCS04」
 - iXs Research製
 - 超小型 (45[mm] x 30[mm])
 - USB接続
 - 4chのRCサーボモータを制御
 - 4chのA/Dコンバータ



iMCS04

RC Motor Controller

■ 入力ポート

- TimedInt型
- RCサーボモータへの指令値 (パルス幅)



RC Motor Controller

■ 出力ポート

- TimedFloat型
- A/D変換の出力

1. USB カメラ

USB カメラ取得
出力：USB カメラ
取得画像
(OpenCV 形式)

2. 動領域抽出

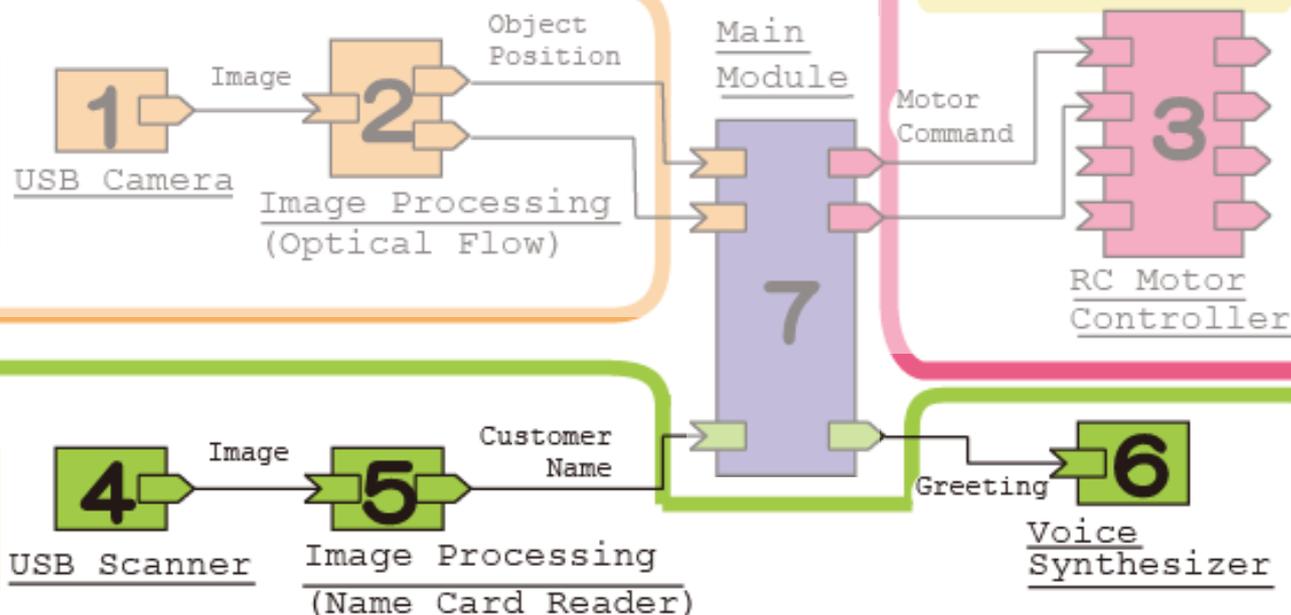
画像処理 (動領域)
入力：連続画像
(OpenCV 形式)
出力：動領域 X 座標,
動領域 Y 座標

7. 全体制御

全体統合制御するプログラム
入力：画像動領域座標
お客様の御名前
出力：モータ指令値
発話文字列

3. サーボモータ制御

ラジコン用サーボ
モータ制御
入力：サーボモータ
指令値
出力：A/D 変換出力



4. TWAIN スキャナ

TWAIN 対応スキャナ
出力：取得画像
(BMP 形式)

5. 芳名認識

名刺認識
入力：名刺画像
(BMP 形式)
出力：お客様の苗字
(ひらがな)

6. 音声合成

音声合成
入力：表音文字列

スキャナ画像取得コンポーネント

- TWAIN対応スキャナ用画像取得コンポーネント

— 今回は名刺読み取り用小型スキャナを使用

- 102(L)×48(W)×26(H)
- 70[g]
- 600dpi



■出力ポート

- TimedOctetSeq型
- BitMap形式画像データ

名刺読み取りコンポーネント

● 芳名認識コンポーネント

- 名刺からフォントの大きな文字領域を抽出
- データベースと照合し、漢字認識
- 氏名漢字読み仮名データベースと照合し、読み仮名をひらがなで出力

■ 入力ポート

- TimedOctetSeq型
- Bitmap形式画像

■ 出力ポート

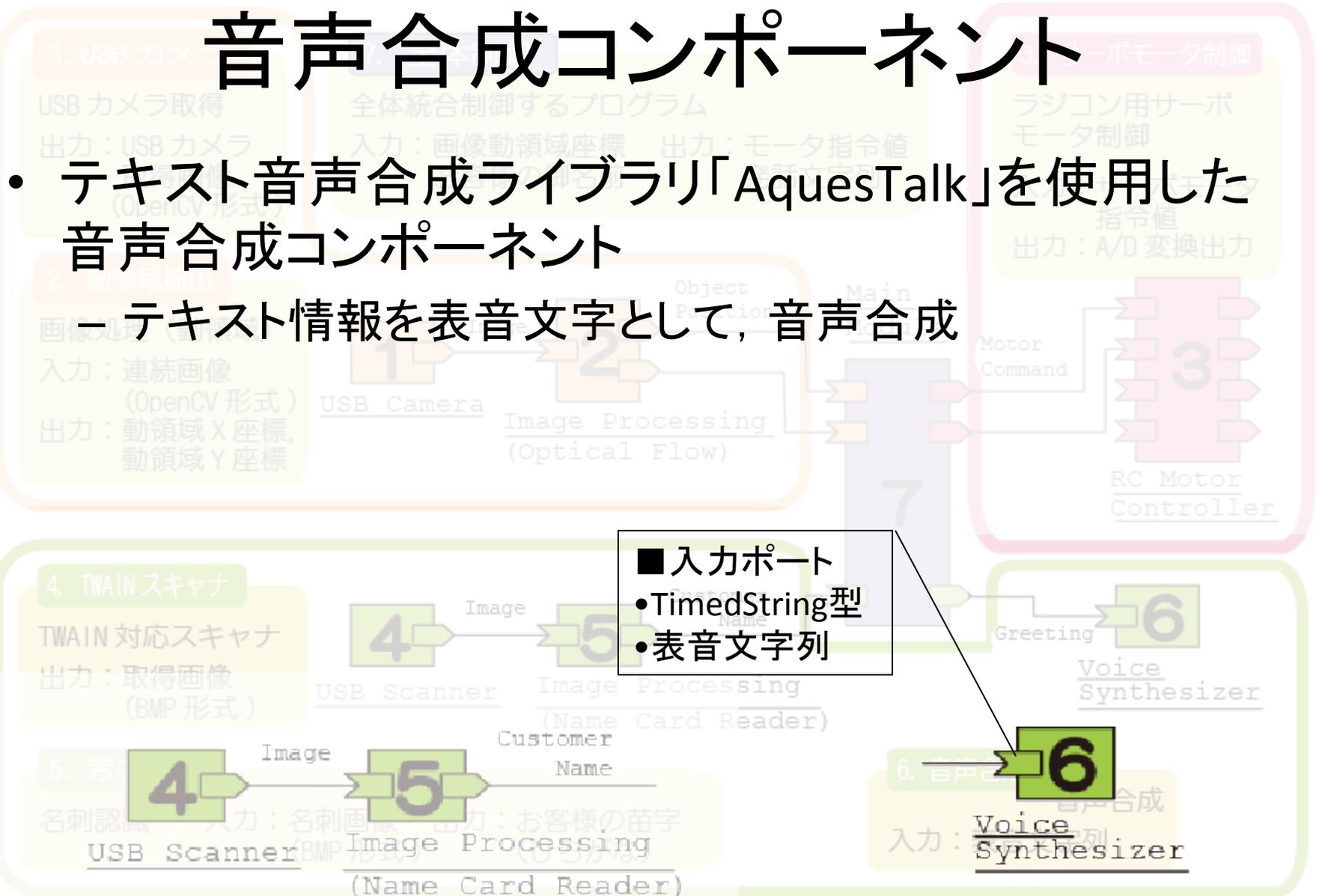
- TimedString型
- 名刺画像から読み込んだお客様のお名前 (ひらがな)

音声合成コンポーネント

- テキスト音声合成ライブラリ「AquesTalk」を使用した音声合成コンポーネント

– テキスト情報を表音文字として、音声合成

■入力ポート
•TimedString型
•表音文字列



1. USB カメラ

USB カメラ取得
出力：USB カメラ
取得画像
(OpenCV 形式)

2. 動領域抽出

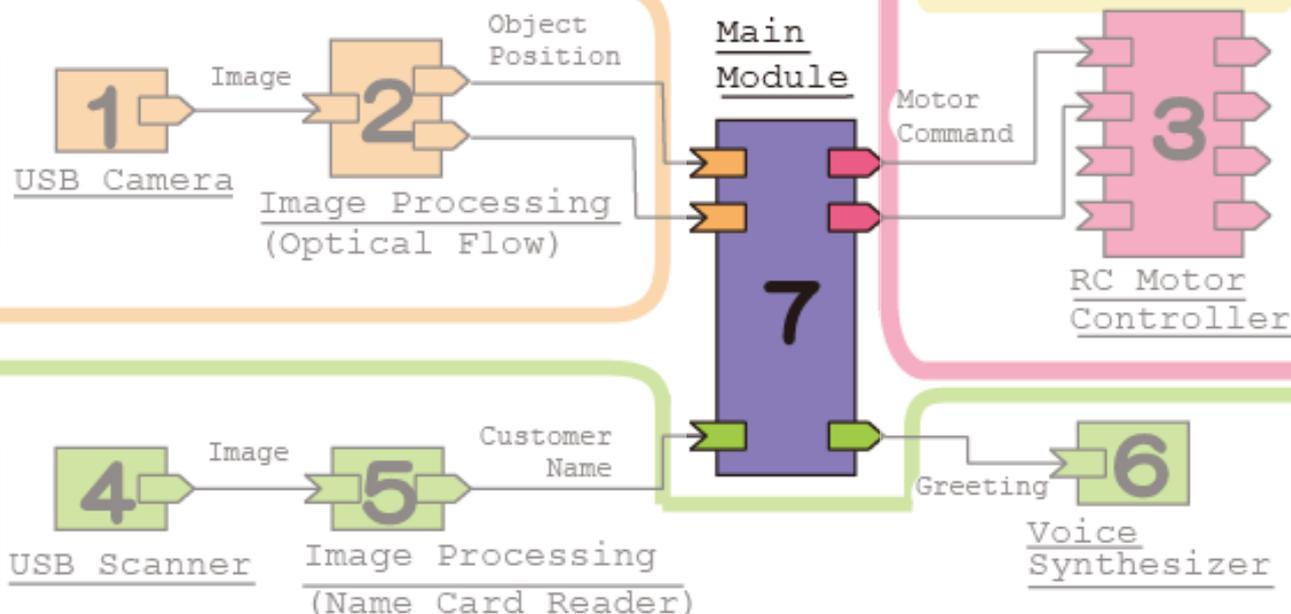
画像処理 (動領域)
入力：連続画像
(OpenCV 形式)
出力：動領域 X 座標,
動領域 Y 座標

7. 全体制御

全体統合制御するプログラム
入力：画像動領域座標
お客様の御名前
出力：モータ指令値
発話文字列

3. サーボモータ制御

ラジコン用サーボ
モータ制御
入力：サーボモータ
指令値
出力：A/D 変換出力



4. TWAIN スキャナ

TWAIN 対応スキャナ
出力：取得画像
(BMP 形式)

5. 芳名認識

名刺認識
入力：名刺画像
(BMP 形式)
出力：お客様の苗字
(ひらがな)

6. 音声合成

音声合成
入力：表音文字列

全体統合コンポーネント

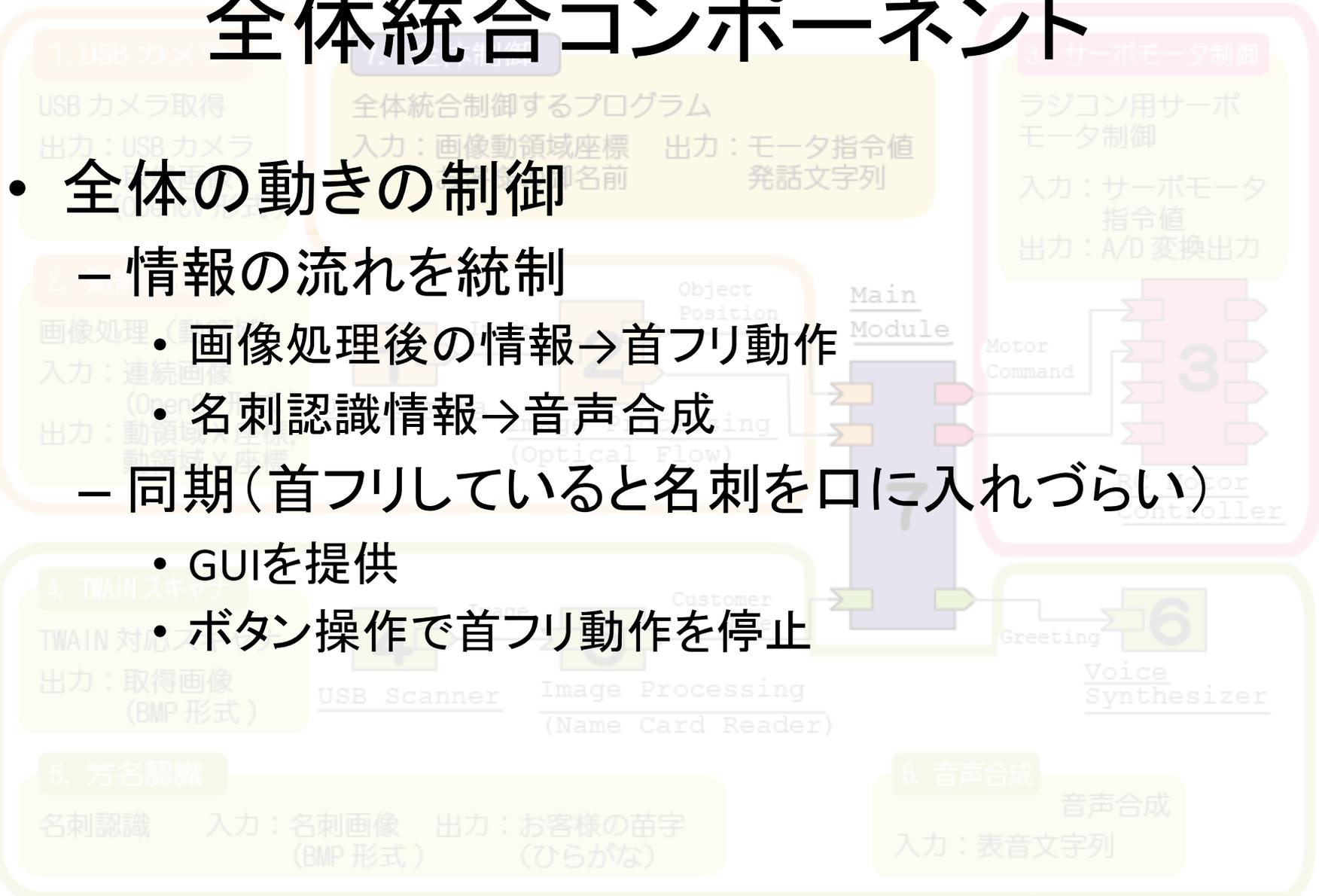
- 全体の動きの制御

- 情報の流れを統制

- 画像処理後の情報 → 首フリ動作
 - 名刺認識情報 → 音声合成

- 同期 (首フリしていると名刺を口に入れづらい)

- GUIを提供
 - ボタン操作で首フリ動作を停止



1. USB カメラ

USB カメラ取得
出力：USB カメラ
取得画像
(OpenCV 形式)

2. 動領域抽出

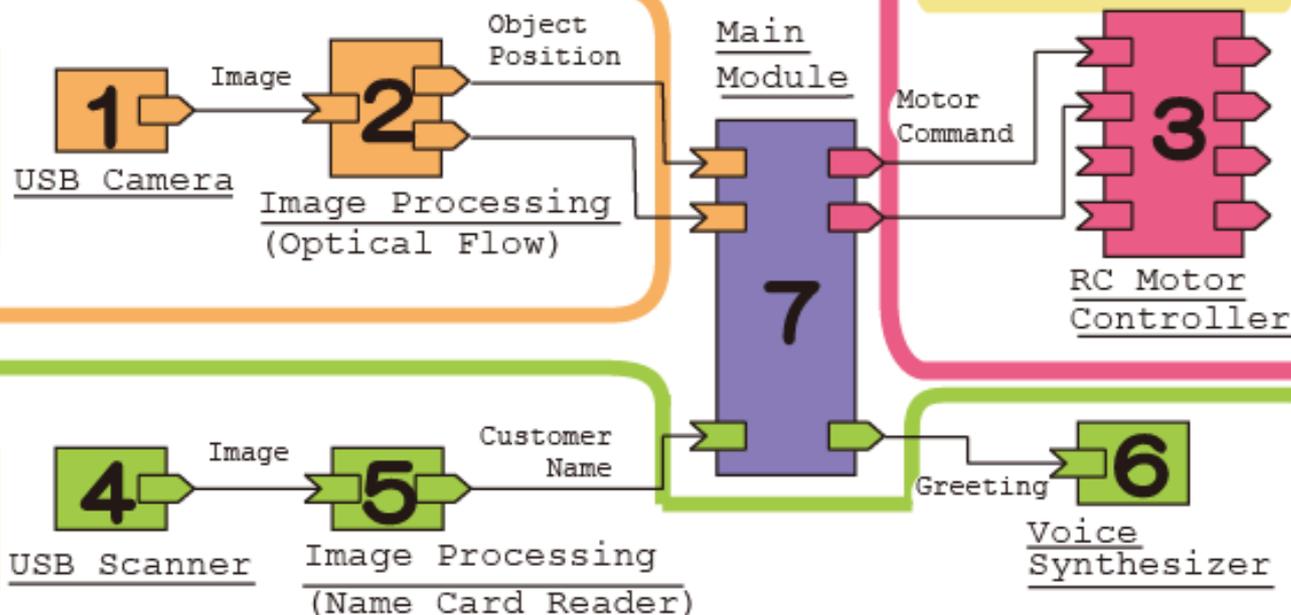
画像処理（動領域）
入力：連続画像
(OpenCV 形式)
出力：動領域 X 座標,
動領域 Y 座標

7. 全体制御

全体統合制御するプログラム
入力：画像動領域座標
お客様の御名前
出力：モータ指令値
発話文字列

3. サーボモータ制御

ラジコン用サーボ
モータ制御
入力：サーボモータ
指令値
出力：A/D 変換出力



4. TWAIN スキャナ

TWAIN 対応スキャナ
出力：取得画像
(BMP 形式)

5. 芳名認識

名刺認識
入力：名刺画像
(BMP 形式)
出力：お客様の苗字
(ひらがな)

6. 音声合成

音声合成
入力：表音文字列

今後の課題等

- 名刺認識ライブラリの改善
 - 認識時間の短縮
 - 認識精度の改善(多種フォントへの対応)
- コンポーネントの再利用性向上
 - 画像形式の統一
 - OpenCV形式が有力か
 - RCサーボコントローラの完成度向上
 - 指令値の形式(パルス幅 or 角度)
- 出力ポートの同期出力
 - 出力ポートに接続されている入力ポートにreadが呼ばれたときに, 同期して出力ポートから画像を転送
 - 名刺スキャナのGUIが必要なくなる



RTC-scilab

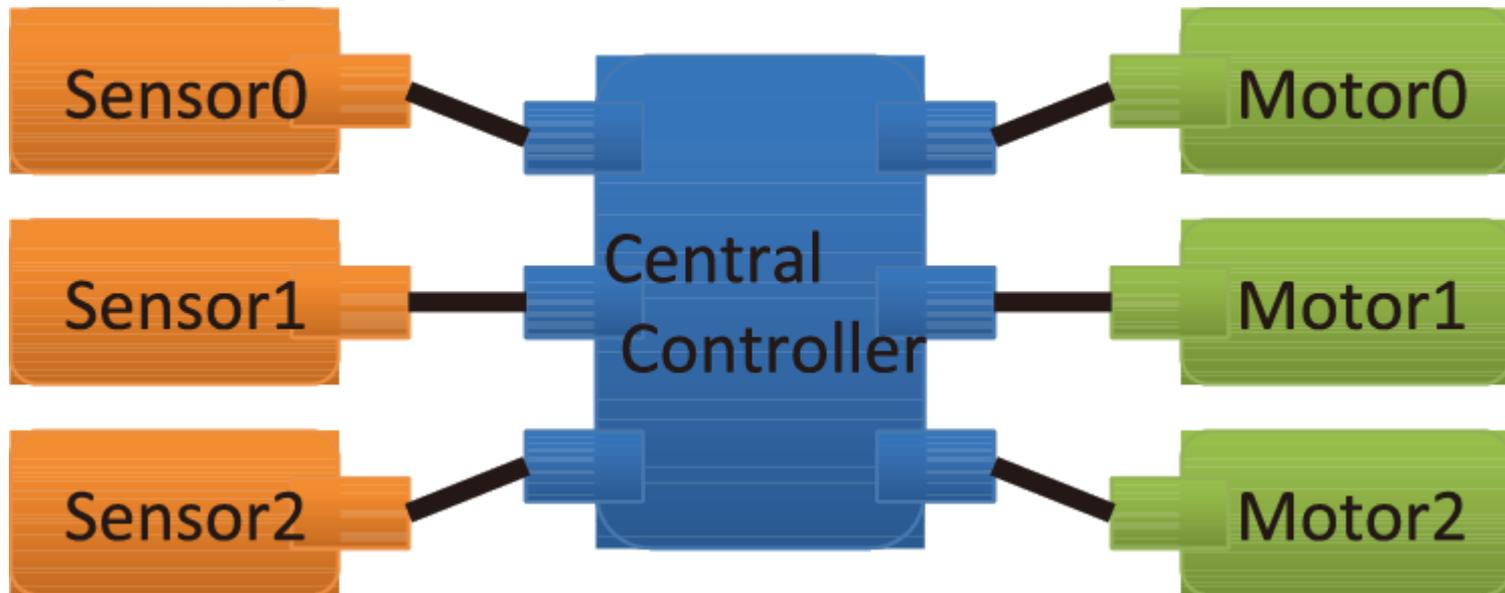
(RTミドルウェアとscilabを繋ぐtool box)

RTC-scilabの概要

- RTC-scilabとは，RTコンポーネントを，scilabおよびxcosを用いて開発するためのツールボックス
 - OpenRTM-aist 1.0
 - Scilab5.2以降，xcos対応
 - RTCの作成・破棄
 - 「ポート」の実装
 - 状態の管理，ポートの接続など
 - OpenHRP3との同期実行
 - Win, Linux対応（Linuxでは一部機能が未対応）

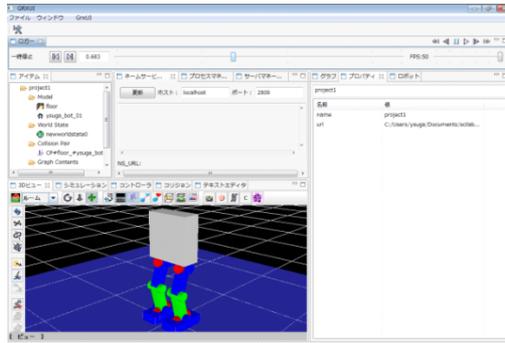
RTC-scilabのコンセプト

- ハードはコンポーネントを使って...
- 中央部はできるだけ簡単に造れないか...

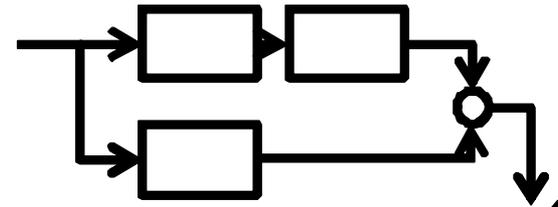


RTC-scilabのコンセプト

RTミドルウェアを用いたメカトロニクス・システム
や動力学シミュレータ



Scilab / xcos による開発
およびシミュレーション

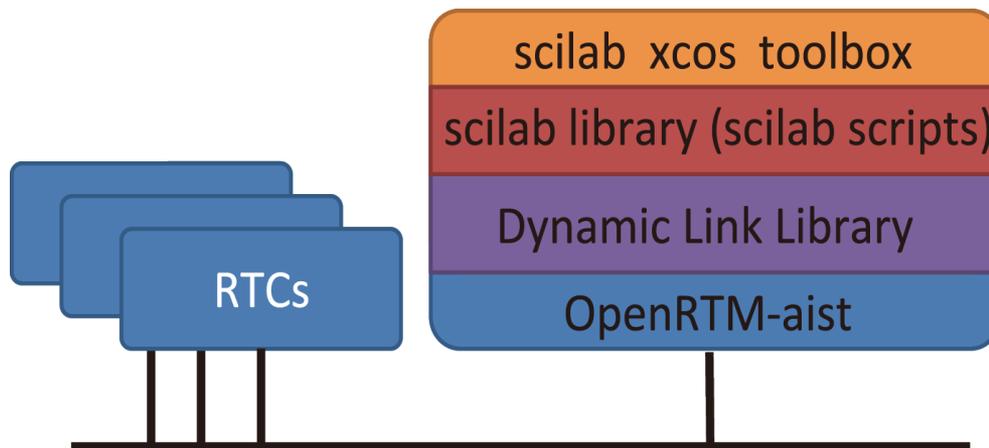


開発者 研究者,
アルゴリズム技術者等)

Scilab / xcos から直接
RTCを作成することにより,
シームレスな開発 & 実装
を実現

RTC-scilabの構造

- DLLからOpenRTM-aist(C++版)にアクセスし, 他のRTCと接続する
- DLLとの接続を簡便にするためのscilabスクリプト
- XcosでRTC-scilabを使うためのブロック



Scilabのスク립トによるRTC開発

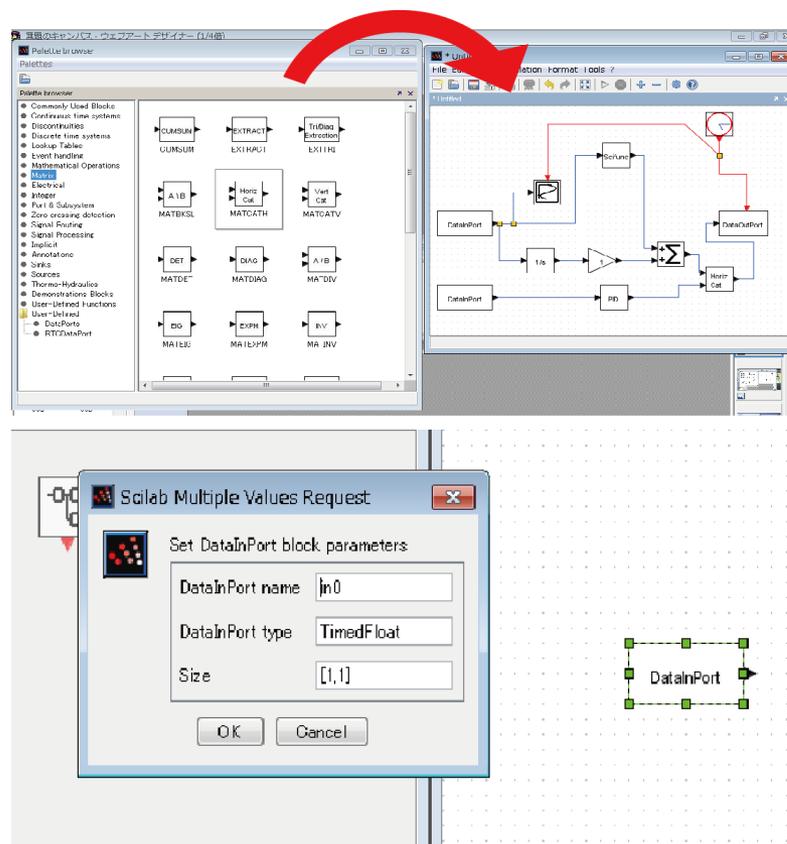
• ソースコードの例

```
001 sciRTC=ScilabComp_create();
002
003 sciRTC = sciRTC.registInPort(sciRTC, "TimedLong", "in");
004 sciRTC = sciRTC.registOutPort(sciRTC, "TimedLong", "out");
005
006 inInPorts = sciRTC.m_InPorts("in");
007 outOutPorts = sciRTC.m_OutPorts("out");
008
009 while %T,
010   if(inInPorts.isNew(inInPorts)) then,
011     len = inInPorts.read(inInPorts);
012     data = inInPorts.get(inInPorts, len);
013     disp("Received Data is =" + string(data));
014     outdata = example_function(data);
015     outOutPorts.write(outOutPorts, outdata);
016   end;
017 end;
```

- Scilab用RTCの生成
- TimedLong型, "in"という名前のInPortの追加
- TimedLong型, "out"という名前のOutPortの追加
- InPortのポーリング
- データの受信. lenはデータ長
- データのscilab側への受けとり
- 出力ポートへのデータの書き込み

Xcosを用いたグラフィカルなRTC作成

- ポートを意味するブロックをxcosダイアグラムに追加
- 専用のグラフィカルな設定ダイアログ
 - ポート名
 - ポートのタイプ
 - ポートのサイズ
- RTC用クロックブロックでRTCと同期して実行



デモ1(動画?)

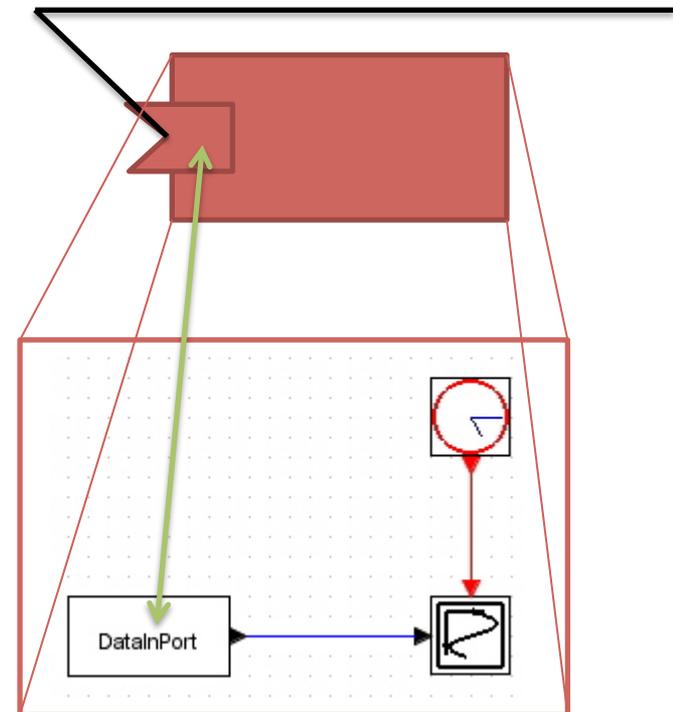
- 3DConnexion社の3次元マウスコンポーネントのデータ取得&視覚化



3次元マウスの各軸のデータ
(double型, 6次元)

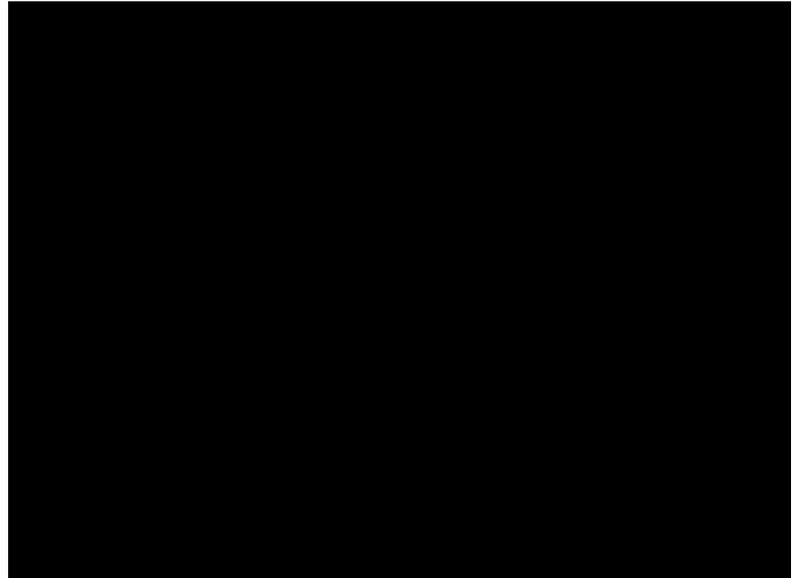


Xcosのscopeブロックへの入力



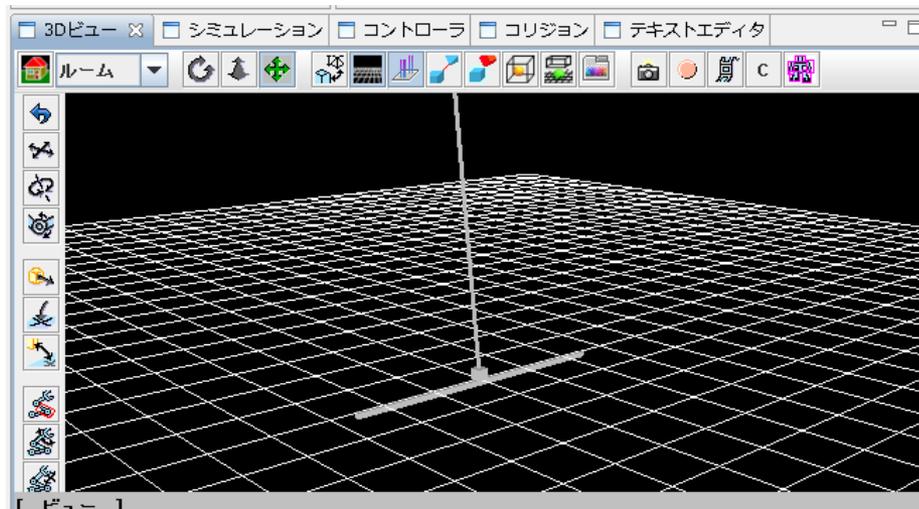
デモ2(動画?)

- OpenHRP3との接続
 - OpenHRP3とは, OpenRTM-aistを使って開発されている
動力学シミュレータ
 - 産総研で開発中の2足歩行ロボットや, 車輪移動型ロボ
ットのシミュレーションに使用されている.
 - フリーかつオープンソース

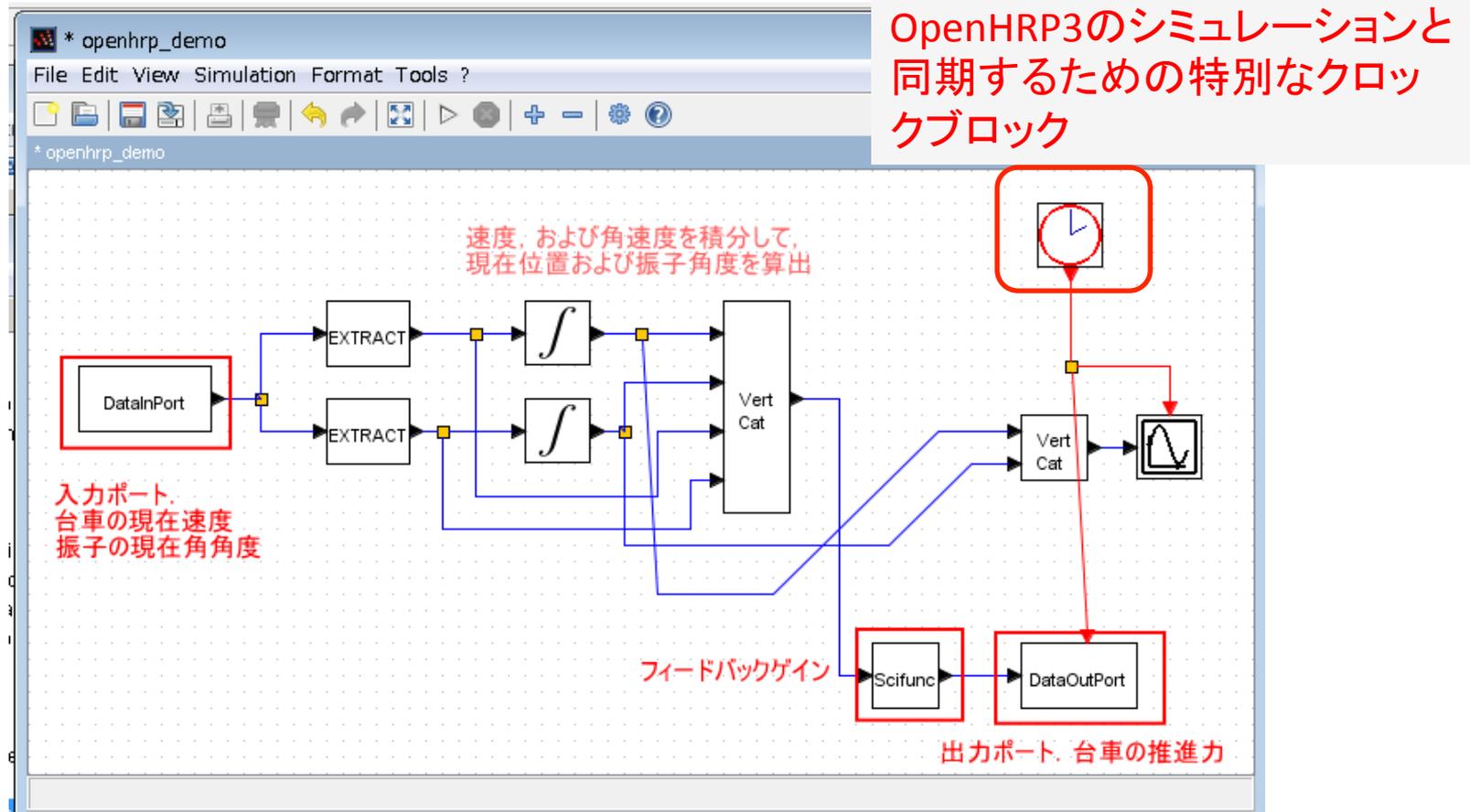


デモ(倒立振子の制御)

- OpenHRP3のシミュレータ空間内の倒立振子を立てる
 - コントローラの入力(振子の角速度, 台車の移動速度)
 - コントローラの出力(台車の駆動力)



状態フィードバックによる倒立振り子制御



まとめ

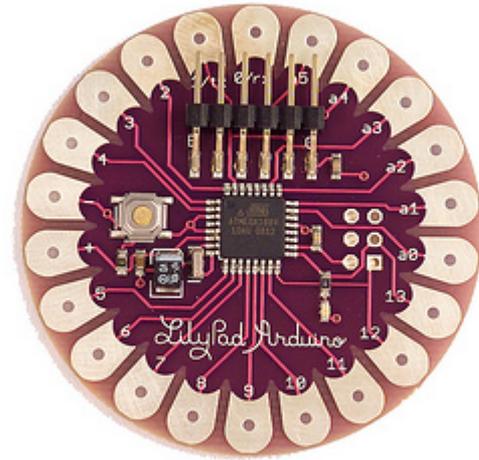
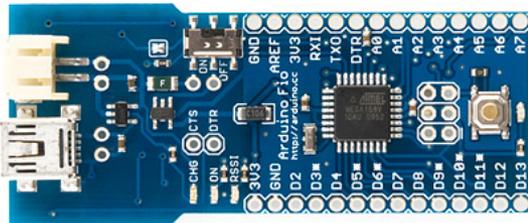
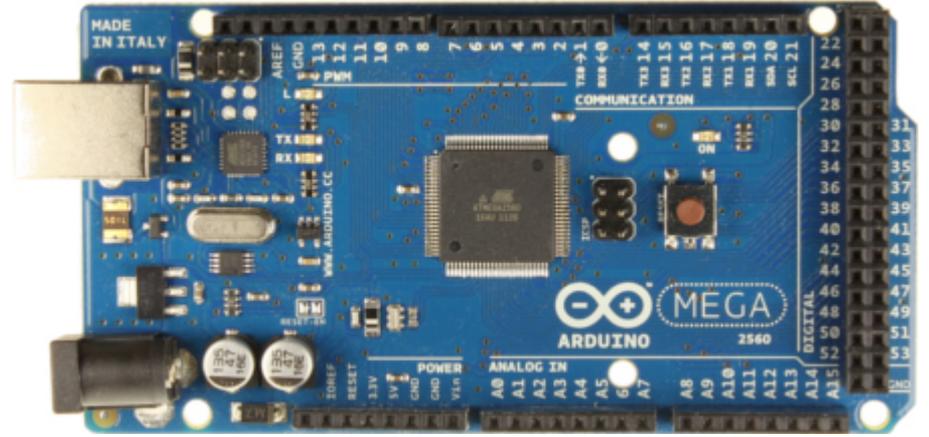
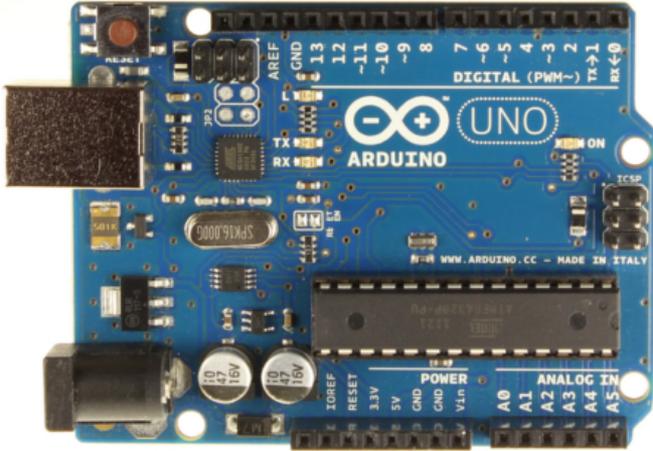
- RTC-scilab
 - ロボットの基盤技術標準化に基づいたソフトウェア要素である「RTコンポーネント」をscilab/xcosを使って開発するためのツールボックス
 - ロボット用のセンサやアクチュエータなどが簡単に使え、独自のアルゴリズムを実ロボット上で実験できる
 - OpenHRP3と同期して、ロボットのシミュレーションが可能

RTコンポーネント対応デバイスを
開発するためのマイコン用
ライブラリ&ツール「RTno」の開発

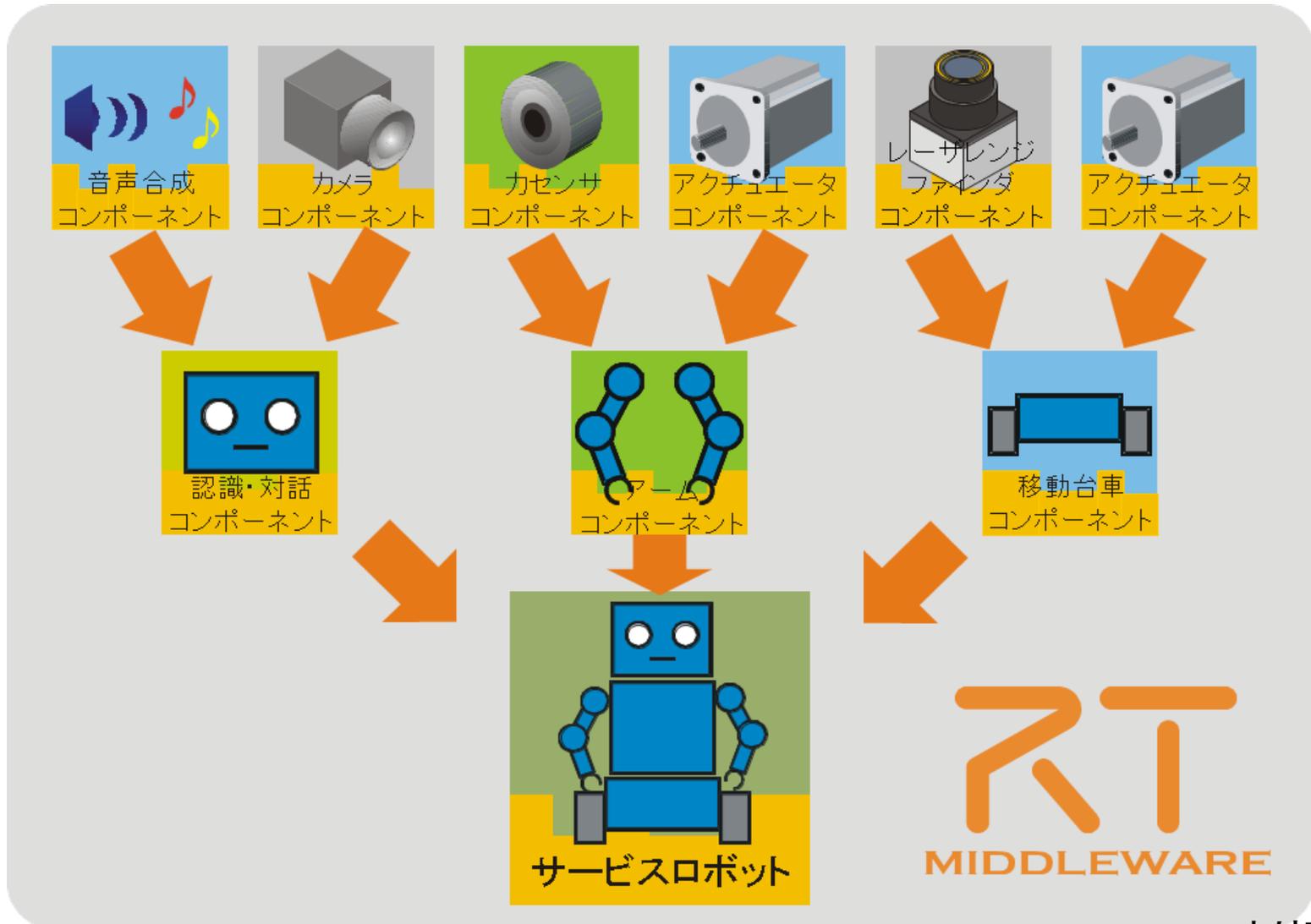
RTno?

RTno = arduino + RT-middleware

arduinoとは



RTミドルウェアとは



RTnoの目的

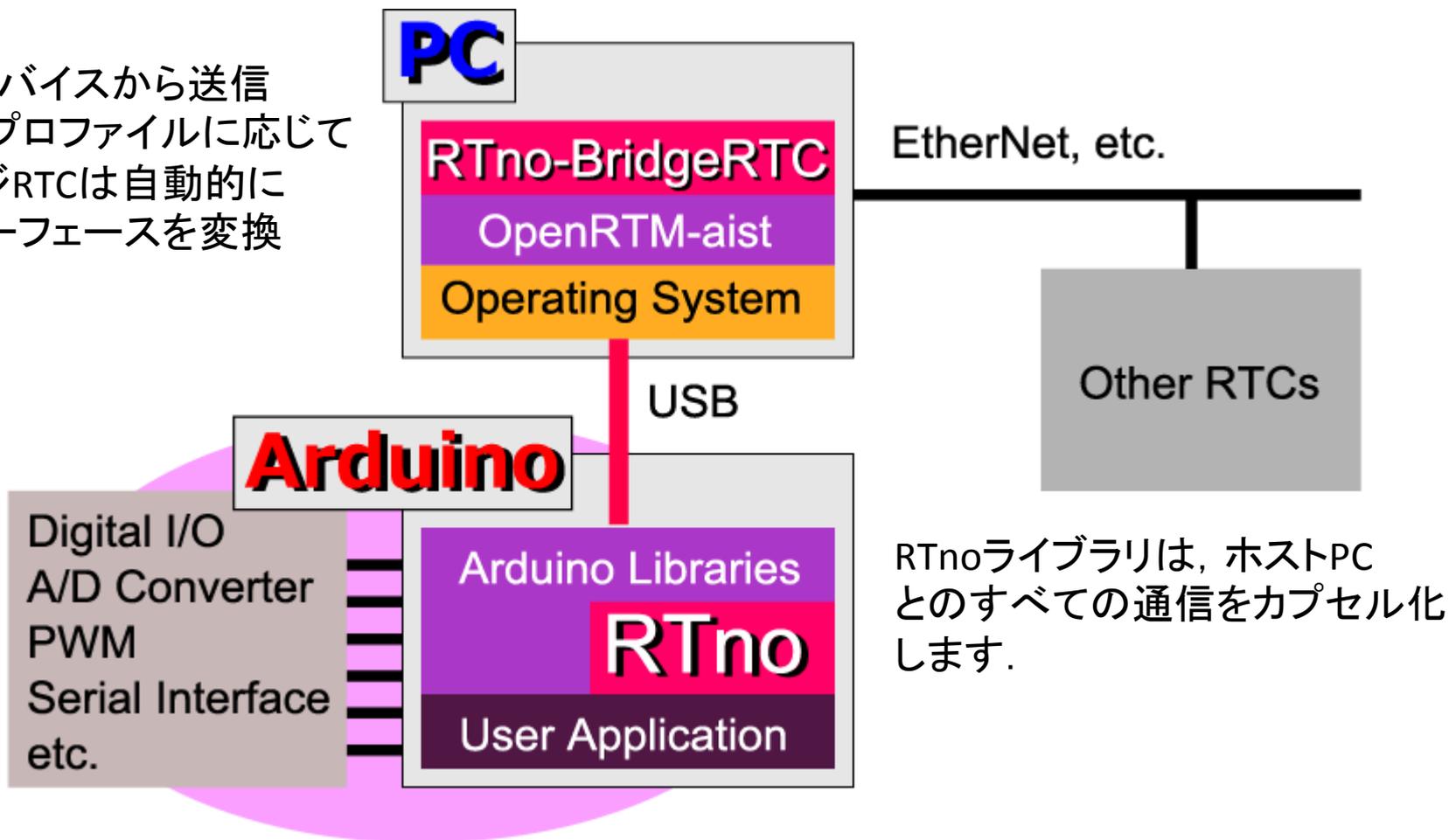
- OpenRTM-aist対応デバイスを簡単に作る
- ウケを取る

従来手法

- OpenRTM-aistの組み込みデバイス対応
 - 組み込みLinuxへの対応
 - TOPPERS対応
- RTC-CANopen
 - CANopen対応デバイスならば対応可能
- RTC-Lite (miniRTC, microRTC)
 - CANバスとZigBee無線ネットワークに対応しており、ARM系の軽量なマイコンでの動作報告がある
- RTno
 - 安価なarduino対応マイコンボード対応
 - Float型, シーケンス型に対応

RTno のアーキテクチャ

Rtnoデバイスから送信されるプロファイルに応じてブリッジRTCは自動的にインターフェースを変換する



RTnoライブラリは、ホストPCとのすべての通信をカプセル化します。

RTnoを使ったRTC対応デバイス開発

- rtccconf関数でセッティングを行う
 - シリアルポートの選択, 設定
 - 実行コンテキスト(タイマもしくはブリッジ同期)

```
#include <RTno.h>

void rtccconf(void) {
    conf._default.connection_type = ConnectionTypeSerial1;
    conf._default.baudrate = 57600;
    exec_cxt.periodic.type = Timer1ExecutionContext;
    exec_cxt.periodic.rate = 1000; // [Hz]
}
```

RTnoを使ったRTC対応デバイス開発

- グローバルスコープでポートおよびバッファを宣言
 - OpenRTM-aistのポートの作成とほぼ同じ

```
TimedLong in0;  
InPort in0In("in0", in0);  
TimedLongSeq in0;  
InPort in0In("in1", in0);  
  
TimedLong out0;  
OutPort out0Out("out0", out0);  
TimedLongSeq out0;  
OutPort out0Out("out1", out0);
```

RTnoを使ったRTC対応デバイス開発

- onInitializeで初期化
 - 起動直後に呼ばれる.

```
int RTno::onInitialize() {
    addInPort(in0In);
    addInPort(in1In);
    addOutPort(out0Out);
    addOutPort(out1Out);

    int LED = 13;
    pinMode(LED, OUTPUT);
    return RTC_OK;
}
```

RTnoを使ったRTC対応デバイス開発

- onActivated, onDeactivatedを記述

```
int RTno::onActivated()
{
    digitalWrite(LED, LOW);
    return RTC_OK;
}
```

```
int RTno::onDeactivated()
{
    digitalWrite(LED, LOW);
    return RTC_OK;
}
```

RTnoを使ったRTC対応デバイス開発

- onExecute内で、データの授受を行う

```
int RTno::onExecute() {
    if(in0In.isNew()) {
        in0In.read();
        long data = in0.data;
        digitalWrite(LED, data);
    }

    if(in0In.isNew(&in1In)) {
        in0In.read();
        for(int i = 0; i < in0.data.length; i++) {
            long data_buffer = in0.data[i];
        }
    }
}
```

```
    out0.data = 3.14159;
    out0Out.write();

    out0.data.length(3);
    out0.data[0] = 1.1;
    out0.data[1] = 2.2;
    out0.data[2] = 3.3;
    out0Out.write();

    return RTC_OK;
}
```

まとめ

- 早い・安い・巧いRTCを提供
 - 学習用に
 - マイコンがわかる人のRTMへの切り口
 - RTMがわかる人のマイコンへの切り口
 - ロボットにシンプルなI/Oを追加
- 今後の展望
 - EtherNetシールドに対応
 - デバイスVSデバイス通信に対応
 - 他のマイコンに移植

あとづけですが・・・

- 毎年, いろいろ考えて出していた・・・かも。
 - 2008年度 マスコットロボット
 - RTM自体が本格スタート直後
 - RTMのメリットの具現化
 - 2010年度 RTC-scilab
 - 知能化PRJでRTCが増えて来た？
 - RTCを使いこなすための中・上位ツール
 - 2011年度 RTno
 - 知能化PRJで作られるRTCは高価で数が少ない
 - 安価な対応デバイス提供

名刺受け機能付きマスコットロボット用コンポーネント群

菅佑樹(早大) 坂本義弘(ナレッジサービス(株))

概要:

- ◆ RTミドルウェア(以下RTM)を用い、名刺受け機能搭載型マスコット・ロボット・システムを開発いたしました。
- ◆ 本ロボットは頸部に2つの自由度を持ち、カメラから取得した画像を用いて首振り・うなずき動作が可能のほか、口蓋内に備えた名刺スキャナと、独自に開発した芳名認識ソフトウェアによって、お客様の御氏名を抽出・認識し、音声合成によって読み上げを行うことができます。
- ◆ RTミドルウェアにより、顧客の要望に応じたシステムの追加等を迅速に行うことができます。

特徴:

- ◆ 頭部カメラ画像内の動領域に追従する首動作
- ◆ 口蓋内の名刺スキャナを使用した名刺受け機能
- ◆ 受け取った名刺からお客様の御名前を読み取り機能
- ◆ 音声合成エンジンを使った御名前読み上げ機能

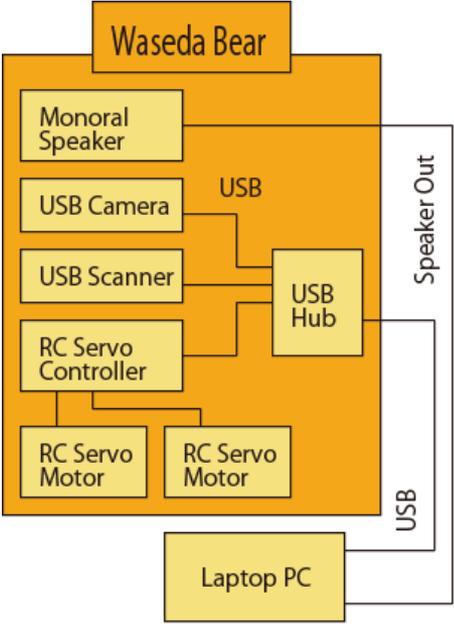
開発したコンポーネント群:

(RTM ver.0.4.2 Windows版)

1. USBカメラコンポーネント (RTM付属サンプルを使用)
2. 動領域抽出コンポーネント
3. iMCs04(サーボモータ制御ボード)コンポーネント
4. TWAIN対応スキャナ制御コンポーネント
5. 芳名認識コンポーネント
6. AquesTalkを用いた音声合成コンポーネント
7. 全体制御コンポーネント



マスコットロボット外観



マスコットロボット システム図

連絡先:

作成者代表: 菅 佑樹 (早稲田大学)

URL: <http://www.ysuga.net/robot/>

E-mail: [ysuga\[at\]ysuga.net](mailto:ysuga[at]ysuga.net)

著作権等問い合わせ: ナレッジサービス株式会社

URL: <http://kserv.jp/>

E-mail: [info\[at\]kserv.jp](mailto:info[at]kserv.jp)

1. USB カメラ

USB カメラ取得
出力：USB カメラ
取得画像
(OpenCV 形式)

2. 動領域抽出

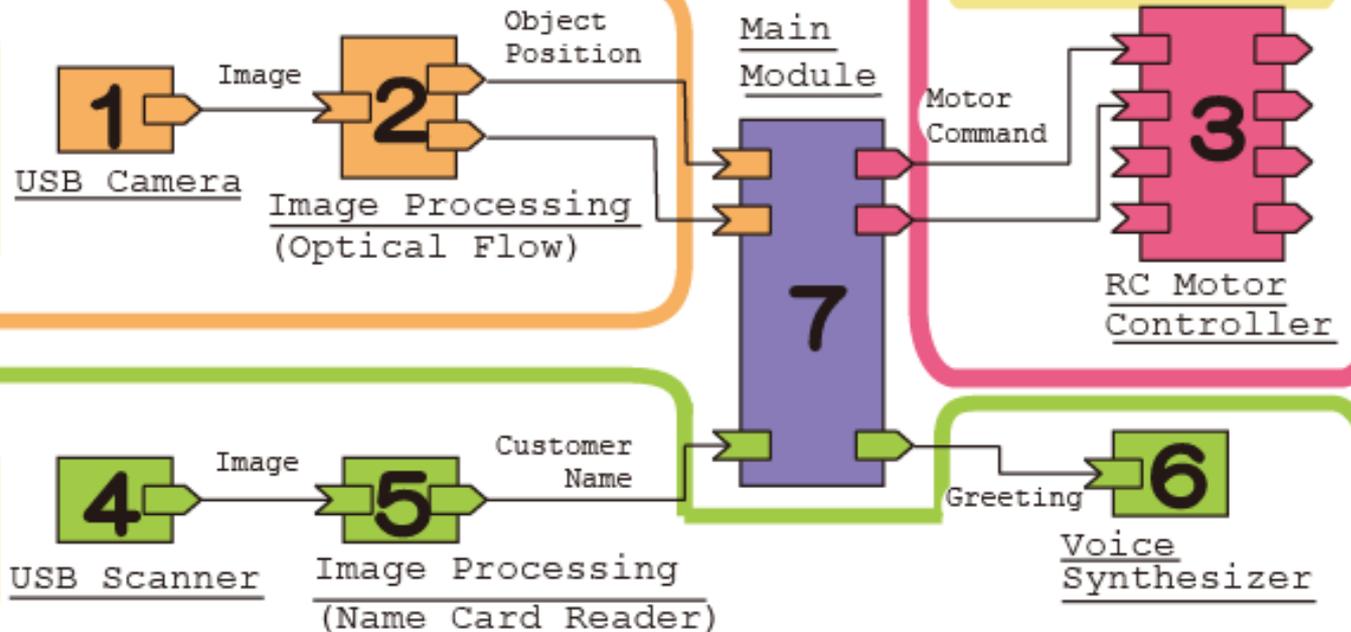
画像処理 (動領域)
入力：連続画像
(OpenCV 形式)
出力：動領域 X 座標,
動領域 Y 座標

7. 全体制御

全体統合制御するプログラム
入力：画像動領域座標 出力：モータ指令値
お客様の御名前 発話文字列

3. サーボモータ制御

ラジコン用サーボ
モータ制御
入力：サーボモータ
指令値
出力：A/D 変換出力



4. TWAIN スキャナ

TWAIN 対応スキャナ
出力：取得画像
(BMP 形式)

5. 芳名認識

名刺認識 入力：名刺画像 出力：お客様の苗字
(BMP 形式) (ひらがな)

6. 音声合成

音声合成
入力：表音文字列

RTC-scilab

菅 佑樹 (株式会社リバスト)



概要:

RTC-scilabは、フリーかつオープンソースな科学技術計算支援ソフトウェア「scilab」と、OpenRTM-aistとの連携を簡単にするためのライブラリです。

特徴:

- ◆ scilab言語を使ったRTコンポーネント開発
- ◆ グラフィカルな開発ツール「xcos」を使ったRTC
- ◆ 動力学シミュレータ「OpenHRP3」との連携

インタフェース:

◆ scilab言語によるプログラミングやxcosによるグラフィカルな開発により、自由に入出力データポートを追加出来ます (TimedLong(Seq), TimedFloat(Seq), TimedDouble(Seq)に対応。)

◆ 対応バージョン

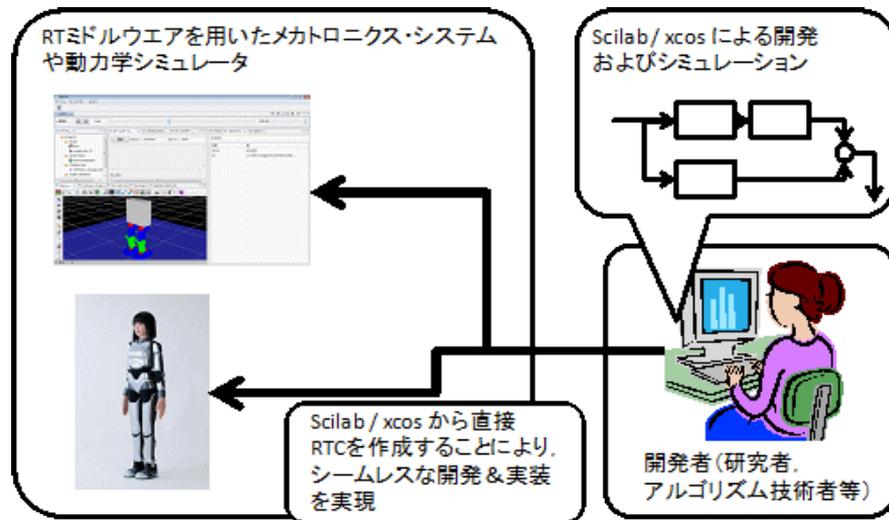
- ◆ OpenRTM-aist C++ 1.0 Release
- ◆ scilab 5.2以降
- ◆ OpenHRP3.1 Release

◆ 対応OS

- ◆ Windows, Linux

ライセンス:

- ◆ GPL v3にて、下記URLで配布中
- ◆ http://www.ysuga.net/robot/rtm/rtc_scilab



RTC-scilabを使うことにより、数式処理やデータの視覚化、グラフィカルな制御システム開発が可能であり、またOpenHRP3と連携することにより、ロボットのダイナミカルな制御システムのプロトタイプングが可能です。RTC-scilabはロボットの制御システム開発を効率化します。

連絡先:

ysuga.net

URL: <http://www.ysuga.net/robot/rtm>

E-mail: ysuga [at] ysuga.net

RTC-scilabの機能



- scilab言語を使ったRTC開発
 - 行列計算や数値解析, データの視覚化など, scilabの様々な技術計算ツールを使うことで, RTC開発効率を高めます.

- グラフィカルなツール「xcos」を使ったRTC開発

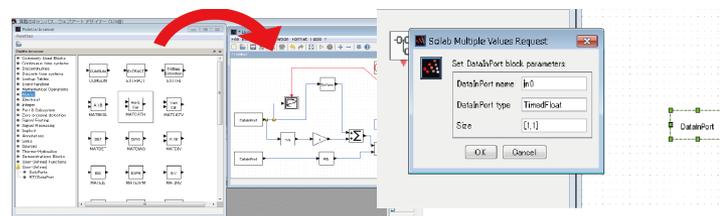
- ブロック線図を描くように開発が出来るので, より直感的なRTC開発が可能になります.

- OpenHRP3との連携

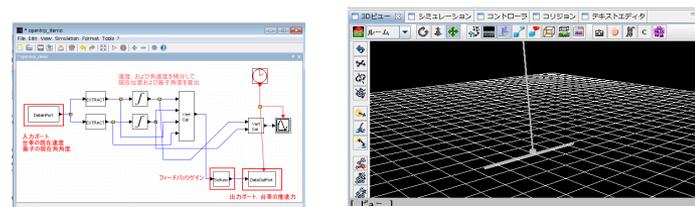
- 動力学シミュレータのOpenHRP3と連携することで, ロボット用コントローラ開発を支援します.

```
001 sciRTC=ScilabComp_create(); _____  
002  
003 sciRTC = sciRTC.registInPort(sciRTC, "TimedLong", "in"); _____  
004 sciRTC = sciRTC.registOutPort(sciRTC, "TimedLong", "out"); _____  
005
```

Scilab言語は数式処理に適した言語であり, 世界中の多くの開発者が数式処理のためのライブラリを無償で提供しています.



Xcosを使えば, パレットから制御ブロックをドラッグ & ドロップで追加して, 制御アルゴリズムとデータポートを追加出来ます. 必要な設定は専用の設定ダイアログに記入するだけです.

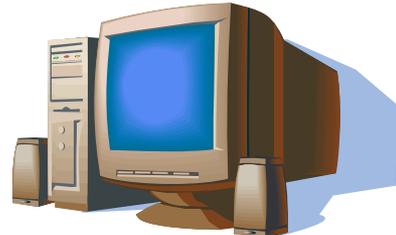


OpenHRP3との連携が可能です. 双方のシミュレーション時間を同期できるので, ロボットのダイナミカルな制御システムのプロトタイピングが出来ます.

arduinoマイコンでRTC対応デバイスを作るためのライブラリ

RTno (アールティノー)

菅 佑樹(株式会社リバスト)



概要:

arduinoマイコンボードは、世界中で使われている安価で使いやすい試作用マイコンボードです。RTnoは、arduinoを使って簡単にRTC対応のデバイスを試作するためのツールとライブラリです。

特徴:

- ◆arduinoマイコンのコードをRTC風にかける
- ◆PCと直接通信し、PC側でOpenRTM-aist対応RTCに変換
- ◆変換用プロキシRTCはボーレート等の設定のみで使える

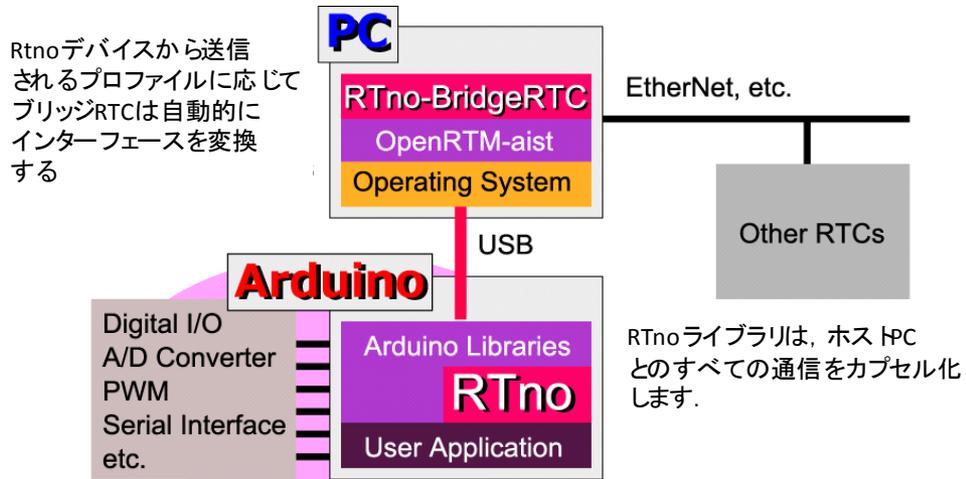
インタフェース(または、使用環境など):

arduinoマイコンボード用コードに自由にデータポートを実装可能(TimedBoolean, TimedLong, TimedFloat, TimedDoubleおよびそれらのシーケンスに対応)

- ◆OpenRTM-aist 1.0 RELEASE対応
- ◆Windows, Linuxで動作確認済み

ライセンス(公開条件):

GPL version3



RTnoライブラリは、ホストPCとのすべての通信をカプセル化します。



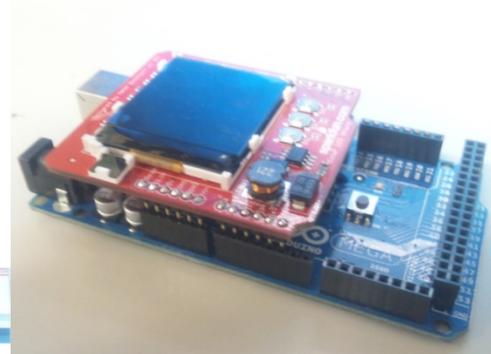
連絡先:

ysuga [あっとまーく] ysuga.net

URL: <http://www.ysuga.net/robot/rtm/rtc/rtno>

RTnoを使ったRTC対応デバイス作成手順(例)

1. arduinoマイコンボードと回路を準備
 - arduino対応の回路(Shield)が多数販売されている
 - 写真はsparkfunのカラーLCDキットを利用
2. PCに接続してプログラム作成
 - OpenRTM-aistと同じく, onInitialized, onExecuteなどの関数をオーバーライドする方式
 - addOutPort, addInPortでポートを追加
 - rtcconf関数でボーレートや実行コンテキストを選択
 - readやwriteでPCと整数, 実数型および, それらのシーケンス型のデータ交換
3. RTnoProxyを起動
 - COMポート等の設定変更のみ
 - arduino側からプロファイルを受信し自動的にRTCのインターフェースを変更



テーマ選びのポイント

- すぐに試せるか否か
 - 対応デバイスは購入可能？安価？高価？
 - 複雑なライブラリのインストールが必要？
- 他に似たモノはあるか？
 - 勿論，独自のモノが良い。
 - 別に他に似たモノがあっても良い！
 - もちろん，論文や発表は参照して欲しい
 - 差別化出来るか？
 - 対応デバイスの数や価格
 - 導入しやすさ

作品開発のポイント

- 出来るだけ自前の方が良い
 - 他のライブラリに依存しすぎると
 - 導入が不便
 - 配布ライセンスに縛りが出来る
- 対応OSを広くする工夫
 - 移植性を考えたコード作成
 - coilを使う
 - Serial portなどはカプセル化する
- テスト用RTCを作る
 - そのままデモ用に出来る

公開のポイント

- ドキュメントを書く
 - PDFで発行するよりは、フリーのウェブサービスでWikiなどにして、適宜更新ができるようにすると良いと思う
- インストラを付ける
 - 是非, tryしてほしいポイント. Visual Studioなら簡単.
- ライセンスをどうするか？
 - 曖昧なことが多い. 是非, すこし勉強して欲しい
 - 依存するライブラリのライセンスにも注意
 - 非商用のみ, というライセンスは怪しいので使えない
 - EPL, GPL, LGPLなどを適宜使うのが良いかと.