

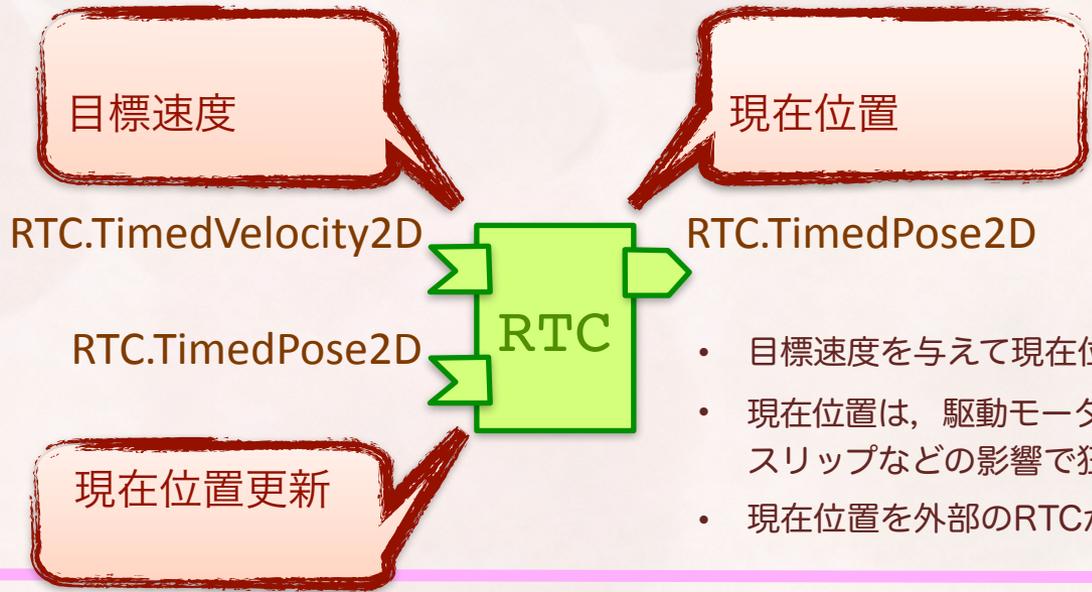
有用なRTCの紹介

菅 佑樹 (早大, SSR)

移動ロボット

サマーキャンプで使える 移動ロボットハードウェア

- Kabuki
- Pioneer 3DX
- OROCHI台車
- RaspberryPiマウス
- Roomba



- 目標速度を与えて現在位置を返す。
- 現在位置は、駆動モータに着いたエンコーダの値の積算なので、スリップなどの影響で狂いやすい
- 現在位置を外部のRTCから更新できる

OpenRTM-aist学習用台車シミュレータ

<http://ysuga.net/?p=133>

- Loader.batを実行

- シミュレータ
- 仮想ジョイスティック
- コントローラ

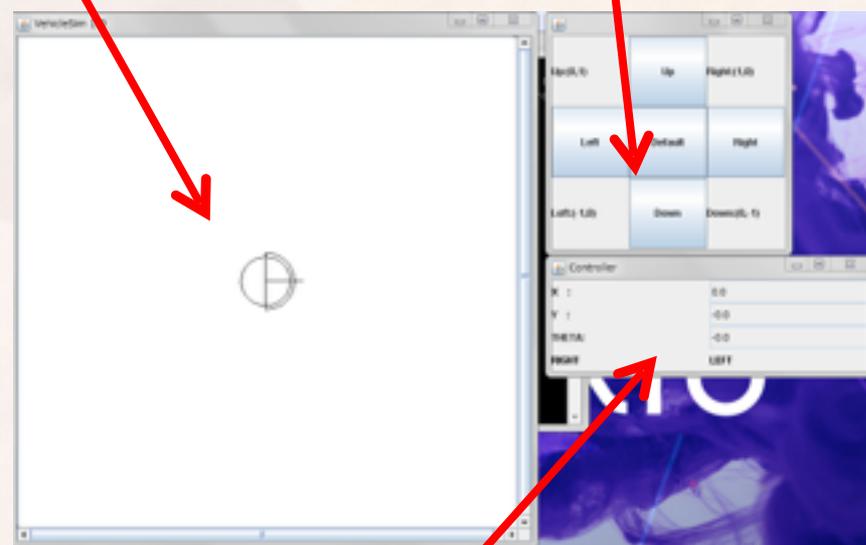
- すべて接続してACTIVATE

- ジョイスティックで操作

- コントローラGUIで位置や接触センサを確認

シミュレータ

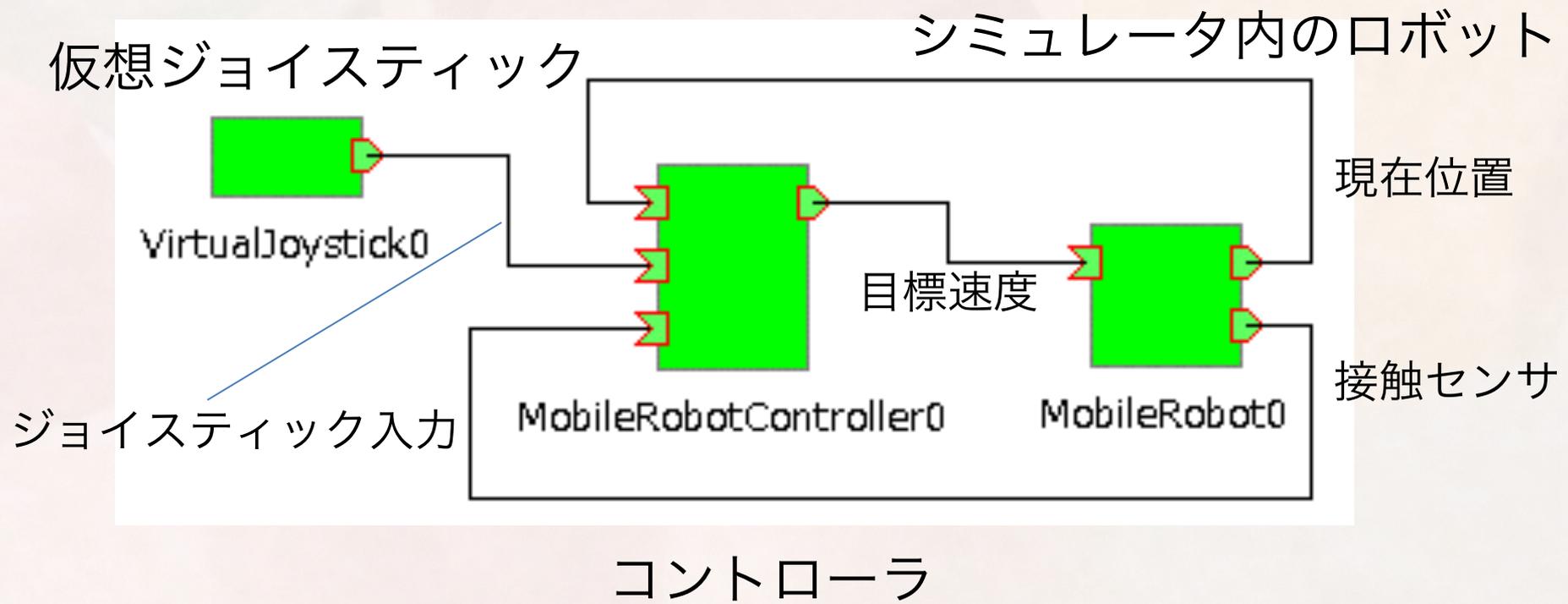
仮想ジョイスティック



コントローラGUI

台車シミュレータの中身

- RT System Editorで表示

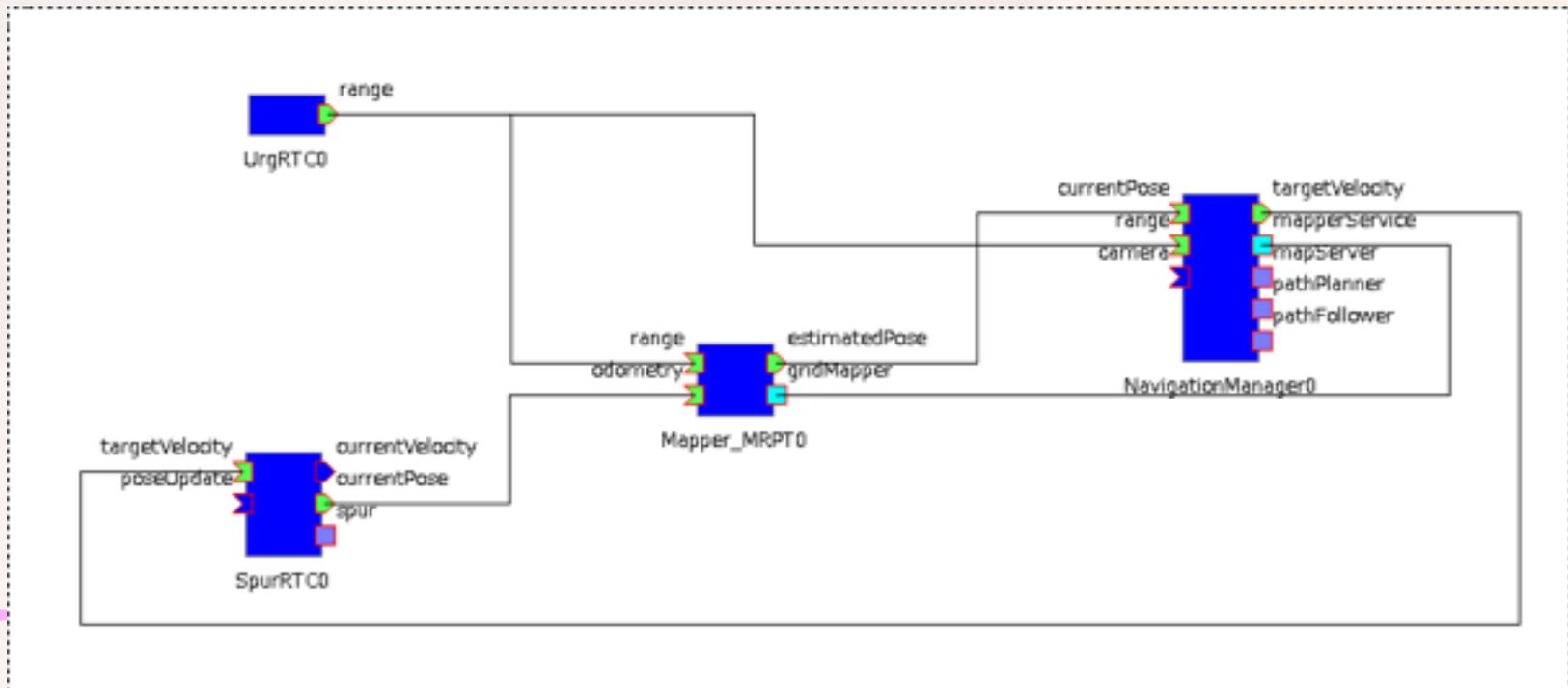


移動ロボットナビゲーション

- 台車型移動ロボットのナビゲーションが目的
 - すべてオープンソース
- マップ作成
 - オンラインSLAMのみ
- 地図を用いたナビゲーション
 - レーザーレンジセンサーを用いた位置推定
 - 軌道計画
 - 軌道追従
- 機能ごとにモジュール化しており、インターフェースを共通化して、入替えが可能
 - 例：独自の移動ロボットを適用
 - 例：独自のプランニング・パスフォロワーを適用
- 基本操作が可能なGUIを準備

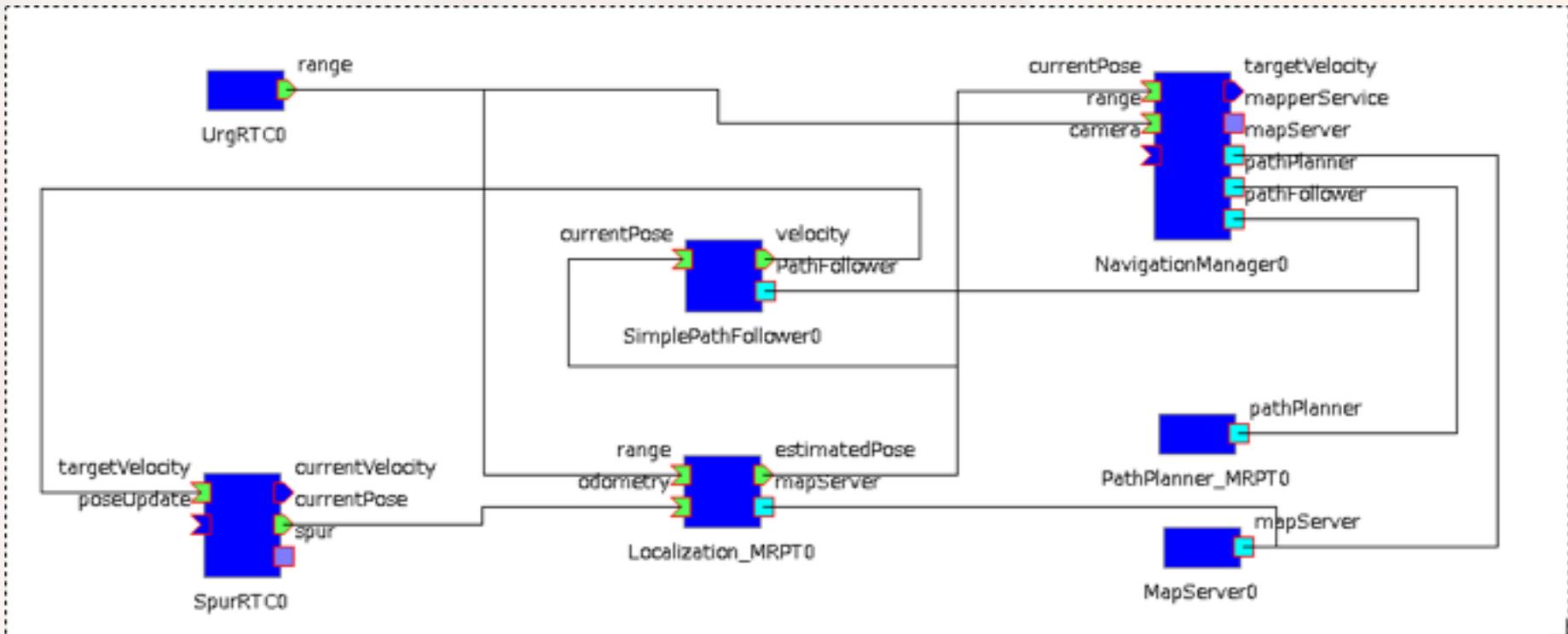
マップ作成

- 北陽電機URGセンサを使ってマップ作成
- MRPTを利用



ナビゲーション

- 出来上がったマップをMapServerで配信
- パーティクルフィルタでの自己位置推定
- Pathプランニングと簡易なパス追従が可能



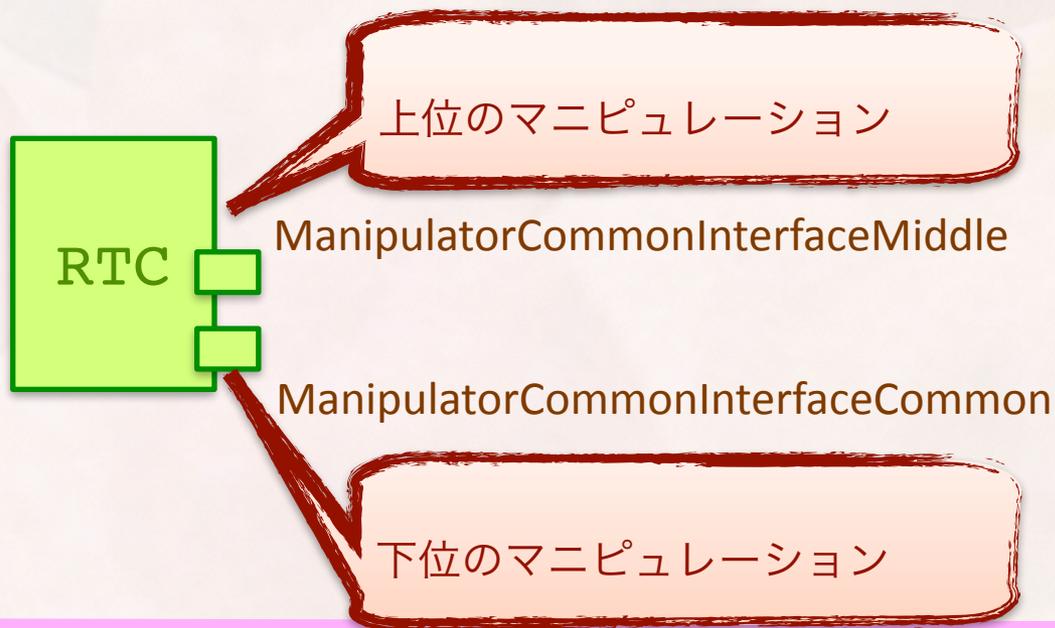
フレームワークの特徴

- ・上位のインターフェースはすべてサービスポート
- ・処理の成功・失敗を返り値で受け取ることができる
- ・サービスロボットの上位処理はデータフロー的ではなく、イベント的
- ・MobileRobot.idlでインターフェースを定義済み
 - ・これをインポートすればコンシューマ側は簡単に作成できる
- ・ロボット側はTimedVelocity2D, TimedPose2Dのデータポート

アーム系

サマーキャンプで使える アーム系ハードウェア

- OROCHIアーム
- Vistonアカデミックスカラロボット



- サービスポートで動作
- 手先の目標位置を指定するとそちらに直線で動く

マニピュレーター 共通インターフェース

- 2013年に改定 (埼玉大学 琴坂研究室による)
- 2つのレベルに分割
 - 共通レベル (ManipulatorCommonInterface_Common)
 - 各関節レベルでの動作, 位置獲得
 - エラー情報取得・復帰
 - 中位レベル (ManipulatorCommonInterface_Middle)
 - 各関節レベルでの動作
 - 手先位置での動作
 - 手先移動範囲の指定や, ツールおよび台座のオフセット
 - ハンドの開閉
 - 共通するデータ型をManipulatorCommonInterface_DataTypesに規定
- http://github.com/sugarsweetrobotics/VS_ASR_RTC/
- <https://github.com/sugarsweetrobotics/ManipulatorCommonTest>

```
/*
Manipulator Common Interface (Data type defenition)
- This IDL is used as service port on RTC
- This command specification is provided by Intelligent RT Software
Project of JARA.
rev. 20140120
*/
#ifndef MANIPULATORCOMMONINTERFACE_DATATYPES_IDL
#define MANIPULATORCOMMONINTERFACE_DATATYPES_IDL
#include "BasicDataType.idl"
module JARA_ARM {
    typedef sequence<double> DoubleSeq;
    typedef sequence<double> JointPos;

    struct LimitValue {
        double upper;
        double lower;
    };
    struct RETURN_ID {
        long id;
        string comment;
    };
    const long OK = 0;
    const long NG = -1;
    const long STATUS_ERR = -2;
    const long VALUE_ERR = -3;
    const long NOT_SV_ON_ERR = -4;
    const long FULL_MOTION_QUEUE_ERR = -5;
    const long NOT_IMPLEMENTED = -6;
    struct TimedJointPos {
        RTC::Time tm;
        JointPos pos;
    };
    typedef unsigned long ULONG;
};
#endif // MANIPULATORCOMMONINTERFACE_DATATYPES_IDL
```

ManipulatorCommonInterface_DataTypes.idl

RETURN_IDでデータを返す

```
/*  
Manipulator Common Interface (Common Commands)  
- This IDL is used as service port on RTC  
- This command specification is provided by Intelligent RT Software  
Project of JARA.  
rev. 20140120  
*/
```

```
*/  
#ifndef MANIPULATORCOMMONINTERFACE_COMMON_IDL  
#define MANIPULATORCOMMONINTERFACE_COMMON_IDL  
#include "ManipulatorCommonInterface_DataTypes.idl"  
module JARA_ARM {  
    enum AlarmType {  
        FAULT,  
        WARNING,  
        UNKNOWN  
    };  
    struct Alarm {  
        unsigned long code;  
        AlarmType type;  
        string description;  
    };  
    typedef sequence<Alarm> AlarmSeq;  
    typedef sequence<LimitValue> LimitSeq;  
    struct ManipInfo {  
        string manufactur;  
        string type;  
        ULONG axisNum;  
        ULONG cmdCycle;  
        boolean isGripper;  
    };  
};
```

~Common.idl

マニピュレーター情報構造体

```
const ULONG CONST_BINARY_00000001 = 0x01; //isServoOn  
const ULONG CONST_BINARY_00000010 = 0x02; //isMoving  
const ULONG CONST_BINARY_00000100 = 0x04; //isAlarmed  
const ULONG CONST_BINARY_00001000 = 0x08; //isBufferFull
```

```
interface ManipulatorCommonInterface_Common {  
    RETURN_ID clearAlarms();  
    RETURN_ID getActiveAlarm(out AlarmSeq alarms);  
    RETURN_ID getFeedbackPosJoint(out JointPos pos);  
    RETURN_ID getManipInfo(out ManipInfo mInfo);  
    RETURN_ID getSoftLimitJoint(out LimitSeq softLimit);  
    RETURN_ID getState(out ULONG state);  
    RETURN_ID servoOFF();  
    RETURN_ID servoON();  
    RETURN_ID setSoftLimitJoint(in LimitSeq softLimit);  
};
```

```
};  
#endif // MANIPULATORCOMMONINTERFACE_COMMON_IDL
```

関節のリミット設定

サーボON/OFF

```
/*  
Manipulator Common Interface (Middle Level Commands)  
- This IDL is used as service port on RTC  
- This command specification is provided by Intelligent RT Software  
Project of JARA.  
rev. 20140120  
*/
```

```
#ifndef MANIPULATORCOMMONINTERFACE_MIDDLE_IDL  
#define MANIPULATORCOMMONINTERFACE_MIDDLE_IDL
```

```
#include "ManipulatorCommonInterface_DataTypes.idl"
```

```
module JARA_ARM{
```

```
    typedef double HgMatrix [3][4];
```

```
    struct CarPosWithElbow {  
        HgMatrix carPos;  
        double elbow;  
        ULONG structFlag;  
    };
```

```
    struct CartesianSpeed {  
        double translation;  
        double rotation;  
    };
```

```
    interface ManipulatorCommonInterface_Middle {  
        RETURN_ID closeGripper();  
        RETURN_ID getBaseOffset(out HgMatrix offset);  
        RETURN_ID getFeedbackPosCartesian(out CarPosWithElbow pos);  
        RETURN_ID getMaxSpeedCartesian(out CartesianSpeed speed);  
        RETURN_ID getMaxSpeedJoint(out DoubleSeq speed);  
        RETURN_ID getMinAccelTimeCartesian(out double accelTime);  
        RETURN_ID getMinAccelTimeJoint(out double accelTime);  
        RETURN_ID getSoftLimitCartesian(out LimitValue xLimit,  
            out LimitValue yLimit, out LimitValue zLimit);  
        RETURN_ID moveGripper(in ULONG angleRatio);  
        RETURN_ID moveLinearCartesianAbs(in CarPosWithElbow carPoint);  
        RETURN_ID moveLinearCartesianRel(in CarPosWithElbow carPoint);  
        RETURN_ID movePTPCartesianAbs(in CarPosWithElbow carPoint);  
        RETURN_ID movePTPCartesianRel(in CarPosWithElbow carPoint);  
        RETURN_ID movePTPJointAbs(in JointPos jointPoints);  
        RETURN_ID movePTPJointRel(in JointPos jointPoints);
```

Middle.idl

デカルト座標系での位置・姿勢
を表すための構造体 (変換行列)

グリッパ開閉

手先位置取得

movePTPCartesianAbs
(手先の絶対位置座標指定移動)

movePTPCartesianRel
(手先の相対位置座標指定移動)

```

RETURN_ID openGripper();
RETURN_ID pause();
RETURN_ID resume();
RETURN_ID stop();
RETURN_ID setAccelTimeCartesian(in double acfTime);
RETURN_ID setAccelTimeJoint(in double acfTime);
RETURN_ID setBaseOffset(in HgMatrix offset);
RETURN_ID setControlPointOffset(in HgMatrix offset);
RETURN_ID setMaxSpeedCartesian(in CartesianSpeed speed);
RETURN_ID setMaxSpeedJoint(in DoubleSeq speed);
RETURN_ID setMinAccelTimeCartesian(in double acfTime);
RETURN_ID setMinAccelTimeJoint(in double acfTime);
RETURN_ID setSoftLimitCartesian(in LimitValue xLimit,
    in LimitValue yLimit, in LimitValue zLimit);
RETURN_ID setSpeedCartesian(in ULONG spdRatio);
RETURN_ID setSpeedJoint(in ULONG spdRatio);
RETURN_ID moveCircularCartesianAbs(in CarPosWithElbow carPointR,
    in CarPosWithElbow carPointT);
RETURN_ID moveCircularCartesianRel(in CarPosWithElbow carPointR,
    in CarPosWithElbow carPointT);
RETURN_ID setHome(in JointPos jointPoint);
RETURN_ID getHome(out JointPos jointPoint);
RETURN_ID goHome();
};
#endif // MANIPULATORCOMMONINTERFACE_MIDDLE_IDL

```

setSpeedCartesian
(直行座標系における手先速度)

組み込み

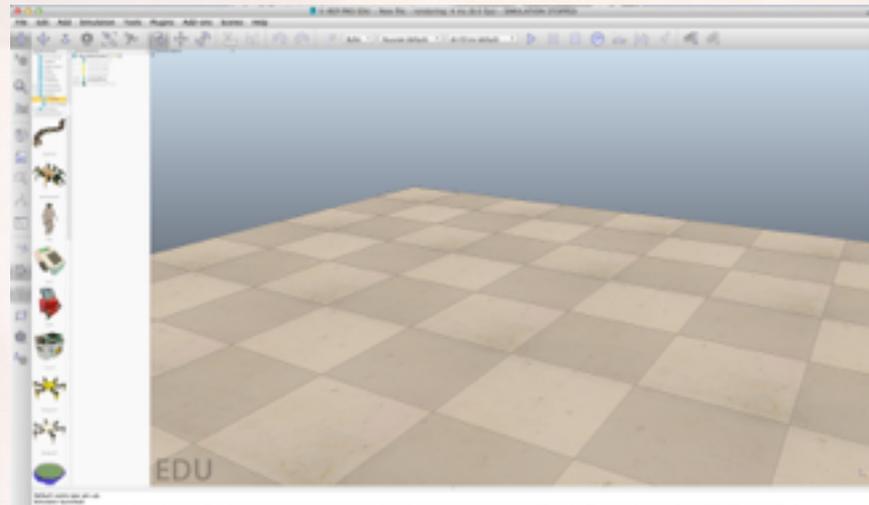
RTno (あーるていーの)

- Arduino + RT = RTno
 - Arduinoを簡単にRTコンポーネントにするライブラリ
 - 自作の回路やデバイスをRTコンポーネントにできる
 - <http://ysuga.net/?p=124>

シミュレーション

V-REPシミュレータ

- RTコンポーネントとの通信に対応した動力学シミュレータ V-REP
 - プラグインをインストールして使う
 - 使い方はこちら
 - <http://ogata-lab.jp/?p=1264>



最近どんなことしてるか

C言語ラッパー

- C言語でRTMを使う
- C言語とのインターフェースを持つ他の言語やツールと繋げる
- C# (Visual C#) に対応
- Unity (Mono) に対応

wasanbon

- RTCのソースコード収集とビルド・設定を補助ツール
- ソースコードの収集
 - <http://wasanbon.org>
- ウェブサービスとの連携

ご清聴ありがとうございました

菅 佑樹 (ysuga@ysuga.net)

株式会社SUGAR SWEET ROBOTICS代表取締役

早稲田大学ヒューマノイド研究所招聘研究員

芝浦工業大学SIT研究所客員研究員

東京大学情報理工学系研究科特任研究員 (JSK-START)プロジェクト

<http://ysuga.net>

<http://sugarsweetrobotics.com>

<http://opensource-robotics.tokyo.jp>

尾形 哲也 (ogata@waseda.jp)

早稲田大学基幹理工学部表現工学科 教授

科学技術振興機構さきがけ研究「情報と人」領域研究員

<http://ogata-lab.jp>

本研究の一部はJSTさきがけ領域「情報環境と人」及び文科省科研費新学術領域研究「構成論的発達科学」(No. 24119003)の助成を受けています。ここに感謝の意を表します。