

サマー・サイエンスキャンプ SUMMER SCIENCE CAMP 2010

「RTで自分のアイデアを実現しよう」



独立行政法人 産業技術総合研究所 知能システム研究部門
2010年8月25日(水)～27日(金)

指導者名簿

- **オーガナイザ**
産業技術総合研究所 知能システム研究部門
統合知能研究グループ・グループ長
神徳徹雄（こうとく・てつお）

- **講師**
産業技術総合研究所 知能システム研究部門
統合知能研究グループ
安藤慶昭（あんどう・のりあき）

株式会社デジタルクラフト
金泰成（きむ・てそん）

- **支援スタッフ**
産業技術総合研究所 知能システム研究部門
統合知能研究グループ
李賢徳（い・ひよんどく）
片見剛人（かたみ・つよと）
栗原眞二（くりはら・しんじ）
宮本晴美（みやもと・はるみ）

株式会社デジタルクラフト
姜榮煥（かん・よんはん）
金湘宰（きむ・さんぜ）
韓相勲（はん・さんふん）

- **産総研事務局**
産業技術総合研究所 広報部 展示業務室
齊藤賢一（さいとう・けんいち）

連絡先

〒305-8568 つくば市梅園 1-1-1 中央第2
独立行政法人産業技術総合研究所 知能システム研究部門
統合知能研究グループ 安藤慶昭
e-mail: n-ando@aist.go.jp, web: <http://staff.aist.go.jp/n-ando>
OpenRTM-aist: <http://www.openrtm.org>



RTで自分のアイデアを実現しよう ～プログラム概要～

独立行政法人産業技術総合研究所
知能システム研究部門
安藤 慶昭

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

コースの狙い

1. 産総研の研究開発の取り組みを知ろう！
2. 数IIで習う三角関数や微分がロボット制御にどのように使われるか学習しよう！
3. 参加者同士で友達になり、交流を深めよう！

使用したレゴはお土産として持ち帰れます！

コースの課題



LEGO Mindstorm NXTを使って、2輪で自立するNXTwayをつくり、それを制御するプログラムを作る。作ったプログラムを友達のNXTwayロボットと互いに接続して一斉制御をする等。

コースのプログラム(1日目)

14:15-15:00	コース概要・自己紹介
15:00-15:15	休憩
15:15-16:00	ロボット工学概論
16:00-16:15	休憩
16:15-17:15	ロボット制御の基礎・宿題の説明
17:30-	サイエンススクエア見学

宿題が出ます！(^^;

コースのプログラム(2日目)

09:00-09:45	ロボットプログラミングの基礎
09:45-10:00	休憩
10:00-12:00	インストール作業、動作確認
12:00-13:00	昼食
13:00-13:45	自己紹介2
13:45-14:00	休憩
14:00-17:15	移動ロボット課題

実際にプログラミングしてロボットを動かします！

コースのプログラム(3日目)

09:00-09:50	グループ発表準備(1)
09:45-10:00	休憩
10:00-11:15	グループ発表準備(2)
11:15-11:30	休憩
11:30-12:00	感想とコメント
12:00-13:00	昼食
13:00-15:00	成果発表会・閉講式

成果発表会があります

自己紹介2って？

- 「私は〇〇さんと同じで、▽▽です。」× 数回、という形式で、改めて自己紹介してもらいます。
- 今晚、みんなでお互いのことをよく話しておかないと、5分持ちません(笑)
- 出身地のこと、学校のこと、家族のことなどいろいろ話してみてください。

昨年の参加者の発表例

ロボットアーム作製

- 二箇所アームを動かせる





RTで自分のアイデアを実現しよう ～ロボット工学概論～

独立行政法人産業技術総合研究所
知能システム研究部門
安藤 慶昭

産総研とは？

産総研について

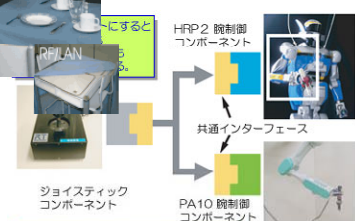
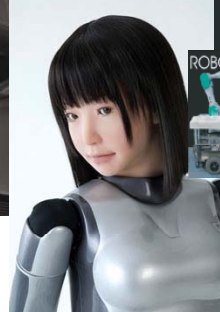
- 独立行政法人産業技術総合研究所
 - 「ライフサイエンス」「情報・通信」「環境・エネルギー」「ナノテク・材料・製造」「地質・海洋」「標準・計測」の6分野
 - 研究者：約2000名



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

産総研について

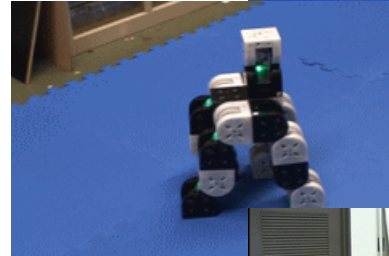
- 知能システム研究部門
 - 「統合知能」「タスク・ビジョン」「ヒューマノイド」「インタラクティブモデリング」「ディペンダブルシステム」「サービスロボティクス」「フィールドロボティクス」の7グループ、「ソフトウェアプラットフォーム研究班」、「AIST-CNRSロボット工学連携研究体」
 - 研究者：約60名(全体約150名)



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

合体変形ロボット:M-TRAN

- 自己再構成型モジュールロボット
 - いろいろな形に変形
 - 様々な環境に適応
 - 探査・救助



http://unit.aist.go.jp/is/frrg/dsysd/mtran3/J_index.htm

メンタルコミットメントロボット:パロ

- メンタルコミットメントロボット
 - 癒し効果(アニマルセラピー)
 - 清潔・手間がかからない
 - ギネス認定「世界でもっとも癒し効果の高いロボット」



http://unit.aist.go.jp/is/frrg/dsysd/mtran3/J_index.htm

ヒューマノイドロボット: HRP

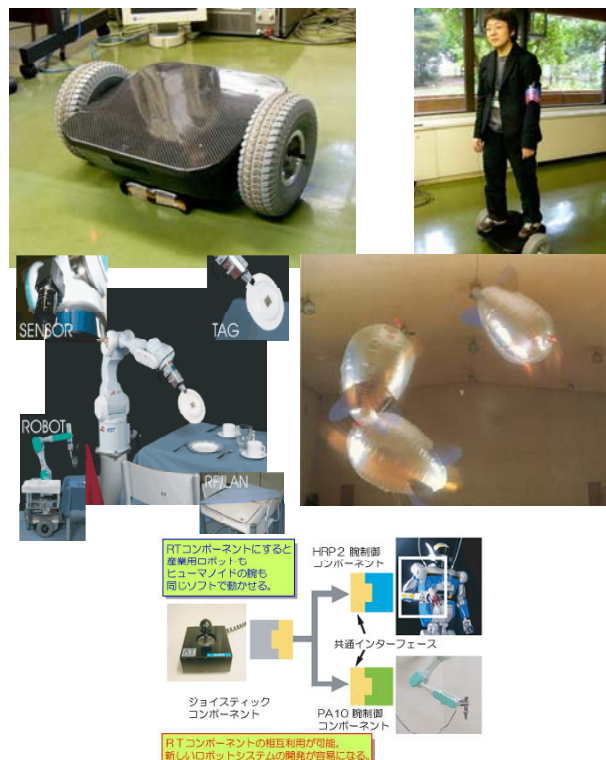
- ヒューマノイドロボット
 - 人型ロボット
 - 人間が生活する環境で活動
 - 二足歩行
 - 研究プラットフォーム



<http://www.aist.go.jp>

その他の研究

- 乗車型移動プラットフォーム
- 飛行船ロボット
- 知能化空間
- RTミドルウェア



RTコンポーネントにすると産業用ロボットもヒューマノイドの動きも同じソフトで動かせる。

HRP2 制御部コンポーネント

ジョイスティックコンポーネント

PA10 制御部コンポーネント

RTコンポーネントの相互利用が可能。新しいロボットシステムの開発が容易になる。

ロボットの实用化

实用化？

- 工場にはたくさんいるよ？
- 日本
 - 稼働台数世界1位
 - 35.8% (2007年)
 - 販売台数世界1位
 - シェア約70%



<http://fanuc.co.jp>



<http://gigazine.net>

<http://www.jara.jp/data/dl/kado.pdf>

2種類のロボット

- 産業用ロボット



<http://fanuc.co.jp>

- サービスロボット



<http://www.nec.co.jp>

サービスロボット？

- 人に対してサービスを提供するロボット
 - 産業用:ものを作る(人間は関係ない)
- 人間がいる場所で動く
 - 産業用:危険なので人間は立ち入らない
- 非定形(いろいろな)の仕事をする
 - 産業用:同じ作業の繰り返し
- 様々な場所(家庭・屋外・公共施設)で使う
 - 産業用:工場内

なぜ必要？

少子高齢化社会

- 労働力不足
- 要介護者増加
- 安全・安心への要求

各種サービスを提供する

ロボットの開発と実用化
への期待の高まり

現実 病院, 福祉施設, 家庭等



製造現場以外で求められる機能が
たくさんある

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

サービスロボットの課題

- 人に対してサービスを提供するロボット
 - いろいろな機能・形態のロボットが必要
- 人間がいる場所で動かなければならない
 - 安全に誤動作なく動く必要がある
- 非定形(いろいろな)の仕事をする
 - 賢くなければいけない
- 様々な場所(家庭・屋外・公共施設)で使う
 - 安くなければいけない

未解決の問題がたくさん

RTミドルウェア

RTミドルウェアプロジェクト

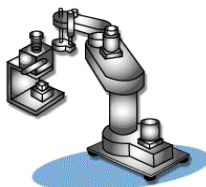
産業界の変革 (大量生産から多品種少量生産への構造変革)

ロボット産業の現状

大量生産向きロボット構築法

- 各社独自仕様によるハードウェアからソフトウェアまでの一括システム開発
- 設計に時間とコストがかかる
 - 高い技術力のある少数メーカーのみ参入
 - 研究成果の移転が困難

コスト競争の体質



21世紀のロボット産業

多品種少量生産向きロボット構築法

- ロボット要素のモジュール化とその統合によるニーズ対応システム開発
- ロボット要素機器の新市場
 - 各専門分野での分業体制
 - インテグレータ産業の創出

付加価値競争の体質

- ロボット開発の活性化
- 産業競争力の強化

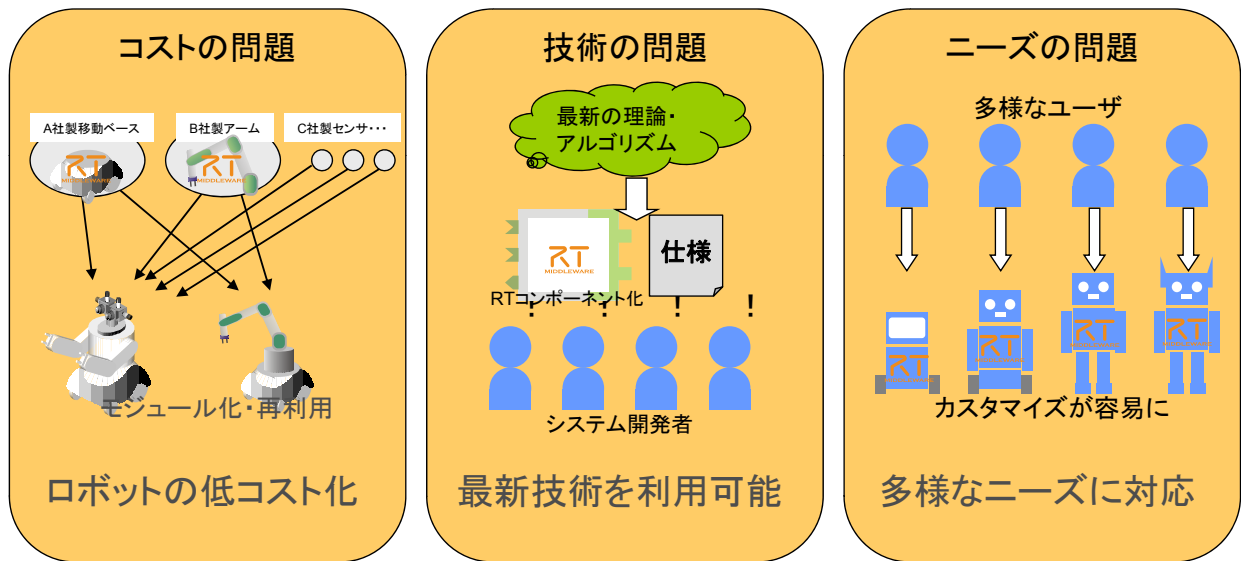
ブレークスルー技術

技術共有を促進する

ロボット共通基盤技術の確立

ロボット産業の産業構造変換

モジュール化による問題解決



ロボットシステムインテグレーションのイノベーション

RTとは?

- RT = Robot Technology cf. IT
 - ≠Real-time
 - 単体のロボットだけでなく、さまざまなロボット技術に基づく機能要素をも含む (センサ、アクチュエータ、制御スキーム、アルゴリズム、etc....)

産総研版RTミドルウェア

OpenRTM-aist

- RT-Middleware (RTM)
 - RT要素のインテグレーションのためのミドルウェア
- RT-Component (RTC)
 - RT-Middlewareにおけるソフトウェアの基本単位

まとめ

- 産総研の紹介
 - 産総研のロボット
- ロボットの実用化
 - 産業用ロボットとサービスロボット
 - ロボット実用化への課題
- RTミドルウェア
 - モジュール化

RTミドルウェアの詳しい情報は...

- OpenRTM-aist公式Webページ
 - Googleで”OpenRTM”で検索
 - <http://www.is.aist.go.jp/rt/OpenRTM-aist/>
- ダウンロード
 - ソースコード、インストーラ等ダウンロード可能
 - C++版、Python版、Java版(予定)
 - Windows用インストーラ
 - Linux各種ディストリ用パッケージ
- メーリングリスト
 - ユーザコミュニティの情報交換
- 講習会
 - 不定期に開催
 - 実習形式
- マニュアル
 - 随時更新中





RTで自分のアイデアを実現しよう ～ロボット工学基礎論～

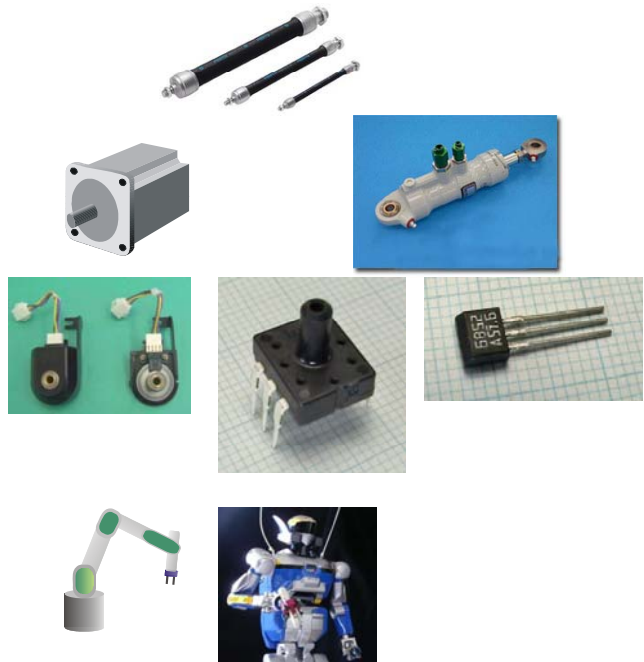
独立行政法人産業技術総合研究所
知能システム研究部門
安藤 慶昭

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

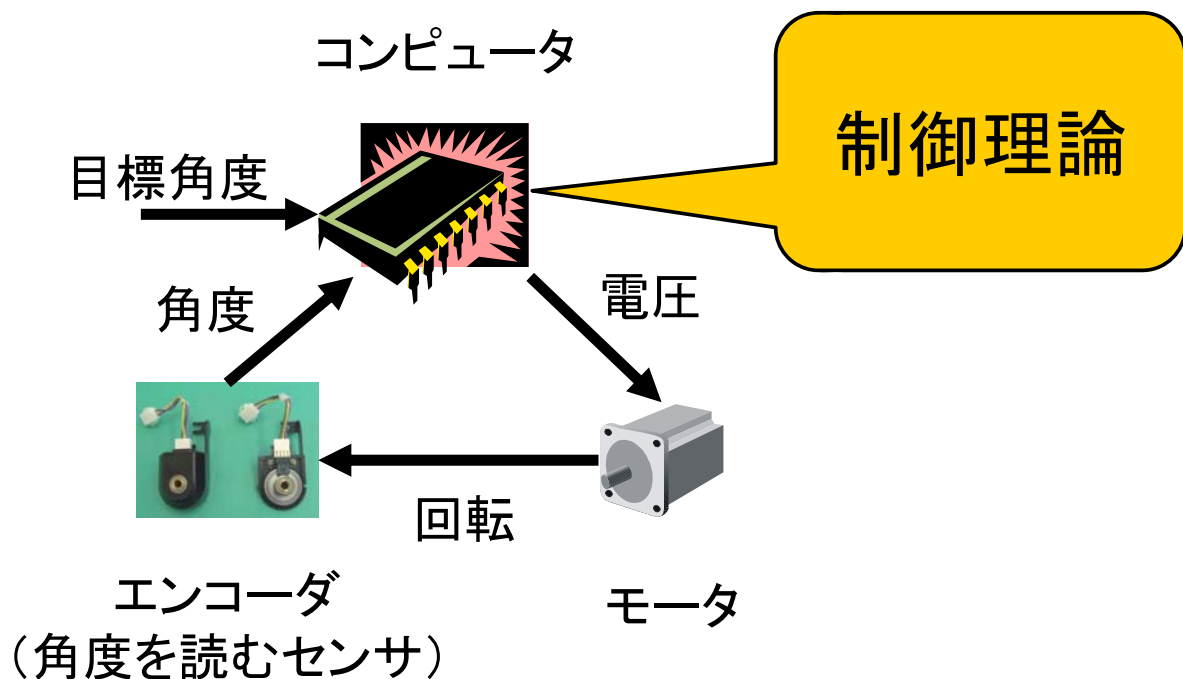
ロボットはどうやって 動いているのか？

ロボットを構成する要素

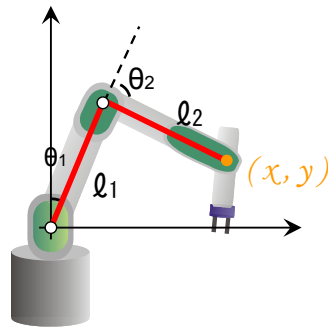
- センサ
 - エンコーダ
 - 力・磁気・加速度など
- コンピュータ
 - ソフトウェア
- アクチュエータ
 - モータなど
- 機構(メカ)



フィードバック制御

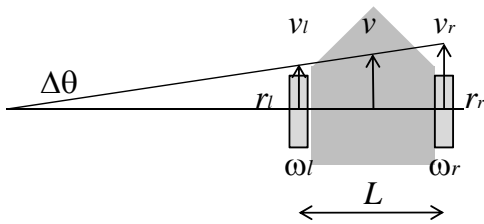


運動学



ロボットアームの運動学

運動学(うんどうがく、英:kinematics)とは運動に影響する要素、原因を考慮しないで物の運動と記述を関係させる古典的な力学の部門であり、基本的に位置の移動とその時間変化を対応させて考える学問である。数理的な手法であり、物理学の原理を基礎としていない。運動学の形式には動作の記述(移動運動学)と回転の記述(回転運動学)がある。実際の物体の場合、運動学はとても複雑であり一般的に厳密な物体の状態は、移動と回転の運動学の両方を兼ね合わせて記述する。
Wikipediaより

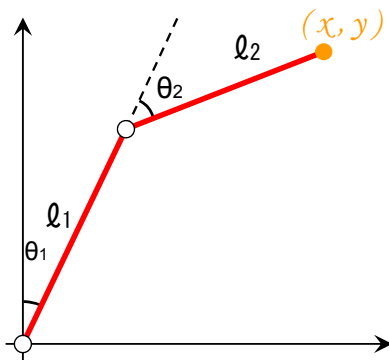


移動ロボットの運動学

ロボットの機構(メカ)を数式で記述する

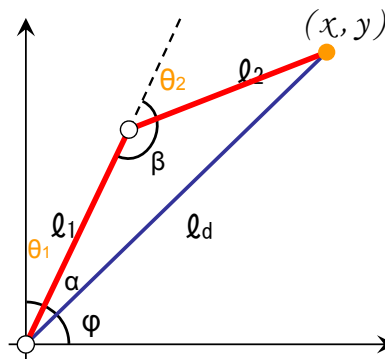
順運動学・逆運動学(2自由度の場合)

- 順運動学



2つの関節の角度がわかっているとき、XY座標を求めることができる

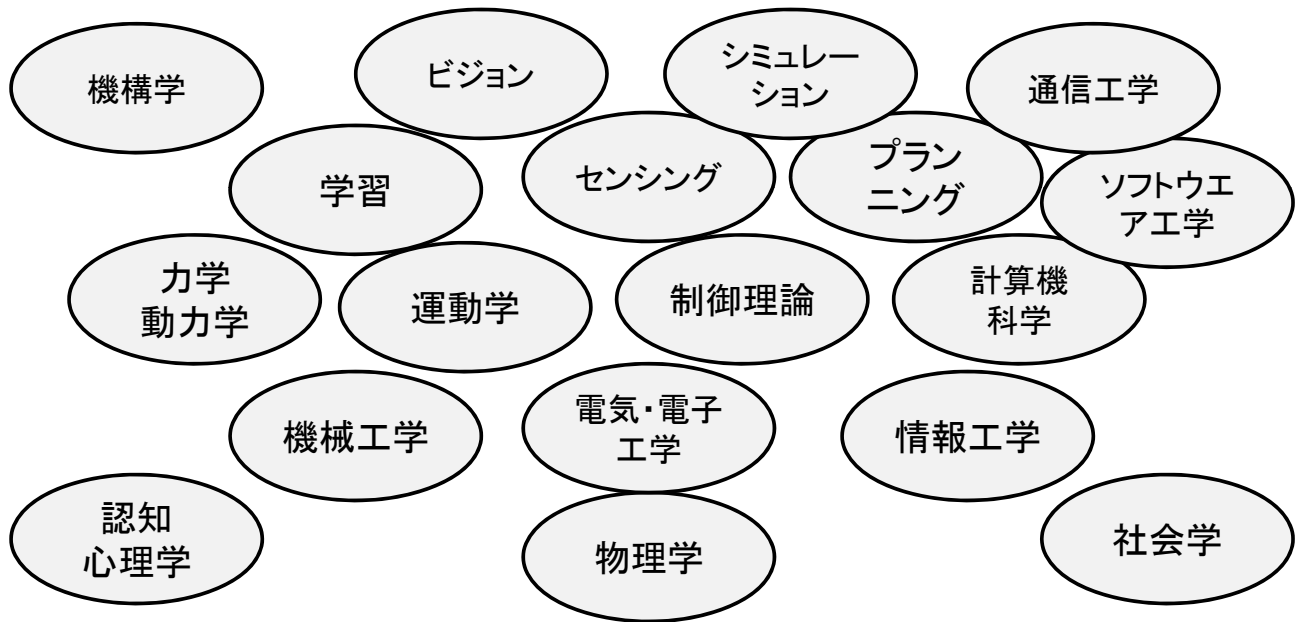
- 逆運動学



XY座標がわかっているとき、関節の角度を求めることができる

注:どの場合も腕の長さははじめから与えられている

そのほかに...

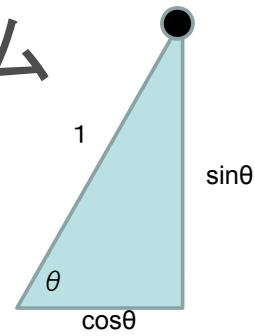
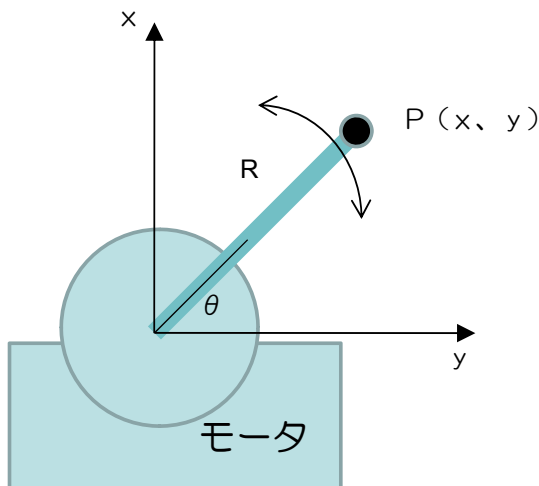


ロボットはいろいろな分野の総合学問
そもそも学問なのか？

自由度？

- 簡単にいえば、関節もしくは動く部分（モータ）の数
- 機構の略してDOF (Degree of Freedom)と書くことも多い。
- 平面上の点を指定するには2自由度で姿勢も考えると3自由度、3次元空間上の点を指定するには3自由度で姿勢も考えると6自由度がそれぞれ必要となる

1自由度のアーム

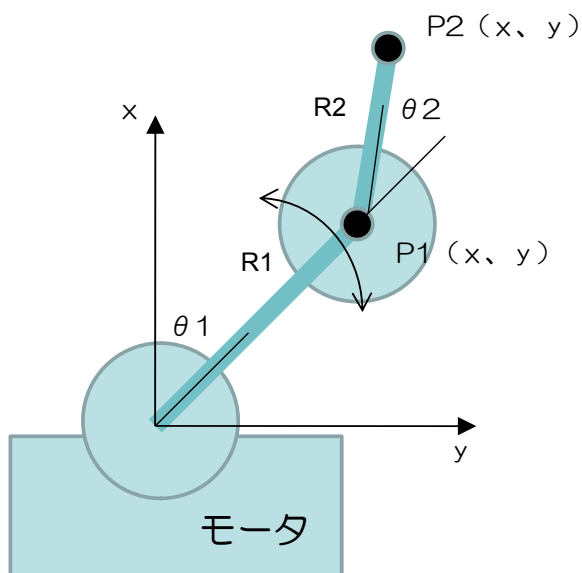


長辺が長さ1の三角形の
それぞれの辺の長さ

点Pの位置は以下のように表せる
 $x = R \times \cos\theta$
 $y = R \times \sin\theta$

2自由度のアーム

1自由度アームの先端にもうひとつモータを
繋いで2自由度アームを作る



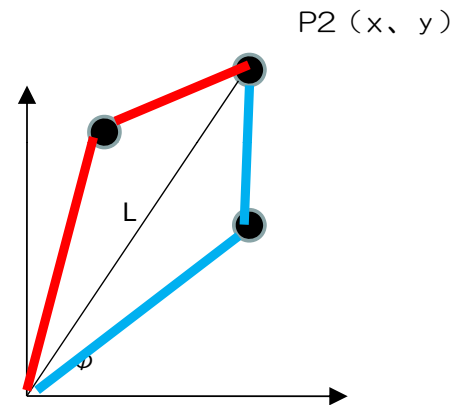
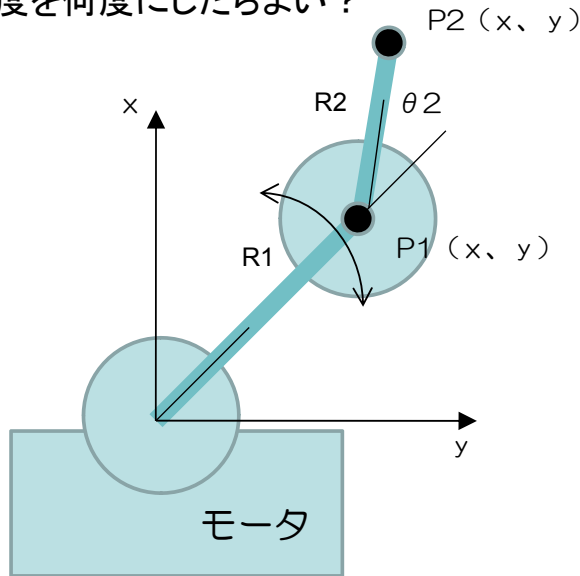
点P1の位置は
 $x = R1 \times \cos\theta1$
 $y = R1 \times \sin\theta1$

点P1から点P2の位置は
 $x = R2 \times \cos(\theta1+\theta2)$
 $y = R2 \times \sin(\theta1+\theta2)$

点P2の位置は以下のように表せる
 $x = R1 \times \cos\theta1 + R2 \times \cos(\theta1+\theta2)$
 $y = R1 \times \sin\theta1 + R2 \times \sin(\theta1+\theta2)$

2自由度のアーム

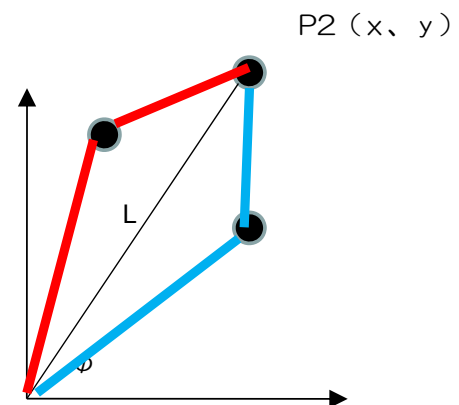
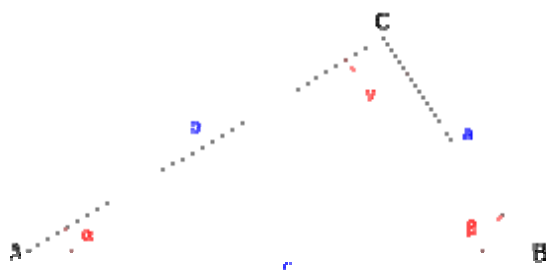
それでは逆に点P2(x, y)の位置にロボットの
手先を持って行くにはそれぞれのモータの角
度を何度にしたらよい？



ロボットアームを折り曲げてLの長
さにしたものをφ回転すると考える
と2つの姿勢の可能性がある

2自由度のアーム

それでは逆に点P2(x, y)の位置にロボットの
手先を持って行くにはそれぞれのモータの角
度を何度にしたらよい？



余弦定理:
三角形の角と辺の関係式

$$\cos \alpha = \frac{b^2 + c^2 - a^2}{2bc}$$

$$L^2 = x^2 + y^2$$

$$L^2 = R1^2 + R2^2 - 2R1R2\cos(180-\theta2)$$

宿題1

2自由度アームの制御に必要な計算式を考えてみよう

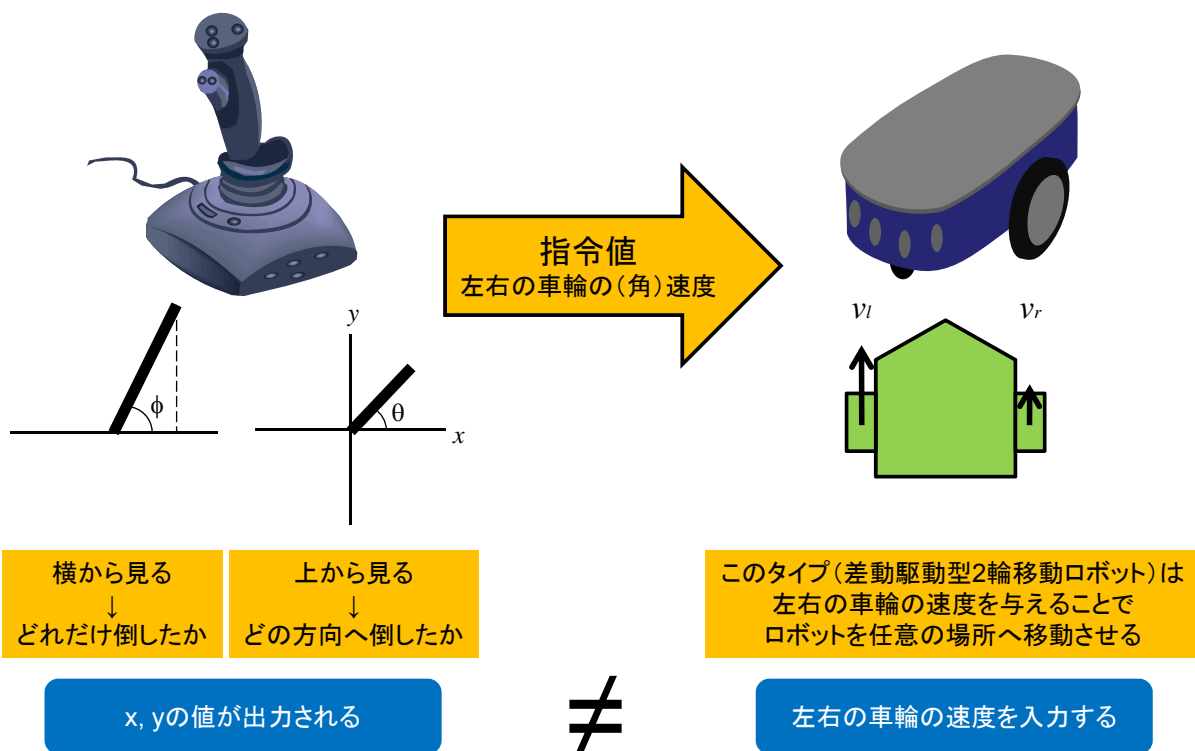
最終的には...

$\theta_1 =$

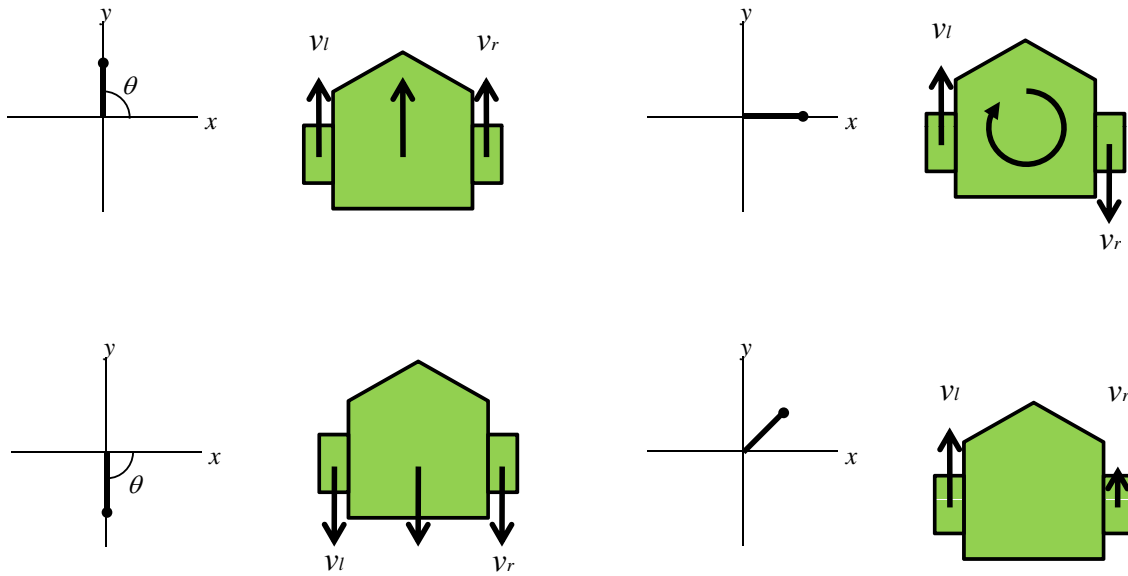
$\theta_2 =$

の形にしたい。

ジョイスティックによる操作

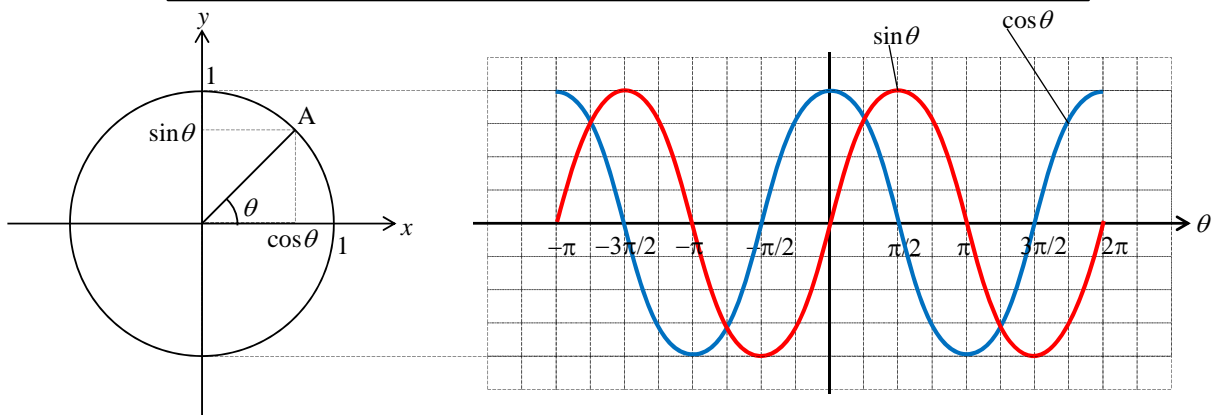


移動ロボットの制御



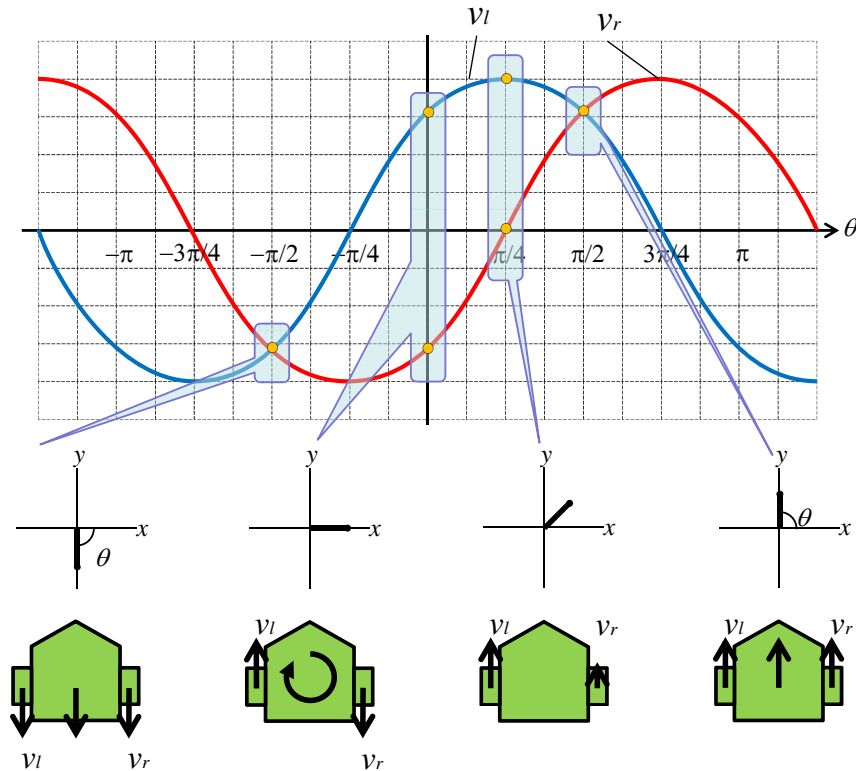
三角関数

関数: $f(x) \rightarrow x$ が有る値のとき特定の値となる式のこと
 三角関数: $\sin \theta, \cos \theta, \tan \theta$



半径1の円周上を点 (x, y) が動くとき、 x 軸と線分OAがなす角: θ
 角度 θ のときの x の値 = $\cos \theta$
 角度 θ のときの y の値 = $\sin \theta$
 θ を変化させてゆくと右のグラフのようになる (サインカーブ)

ジョイスティックと車輪の速度



宿題2

ジョイスティックから車輪の速度を導き出す計算式を考えてみよう

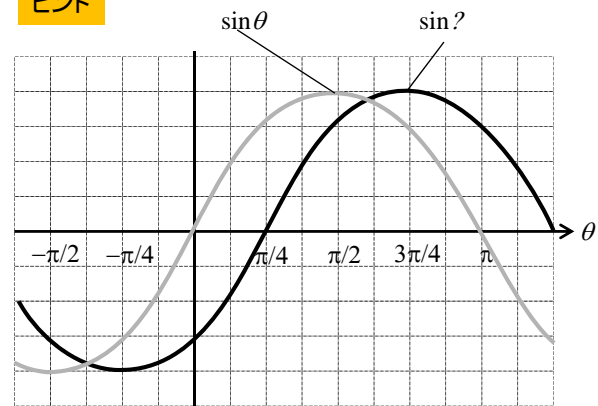
最終的には...

$v_l = x, y$ の式

$v_r = x, y$ の式

の形にしたい。

ヒント



$\sin \theta$ を x 軸方向に $\pi/4$ だけずらす

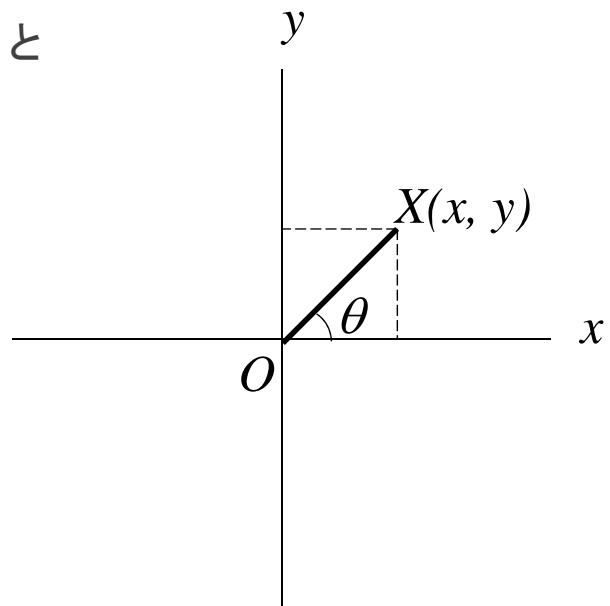
逆関数

OX が x 軸となす角度を θ とすると

$$\tan \theta = \frac{y}{x}$$

なので、逆関数 \arctan を使うと、

$$\theta = \arctan \frac{y}{x}$$



$$\sin \theta = x \text{ のとき } \arcsin x = \theta$$

$$\cos \theta = x \text{ のとき } \arccos x = \theta$$

$$\cos \theta = x \text{ のとき } \arctan x = \theta$$

コースのプログラム(2日目)

09:00-09:45	ロボットプログラミングの基礎・宿題
09:45-10:00	休憩
10:00-12:00	インストール作業、動作確認
12:00-13:00	昼食
13:00-13:45	自己紹介2
13:45-14:00	休憩
14:00-17:00	ロボットアーム課題

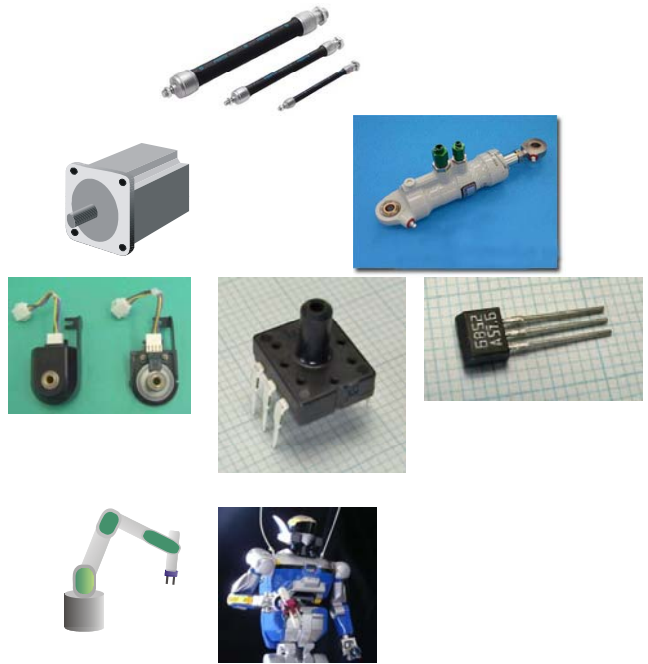
実際にプログラミングしてロボットを動かします！

ロボットはどのように動いているのか

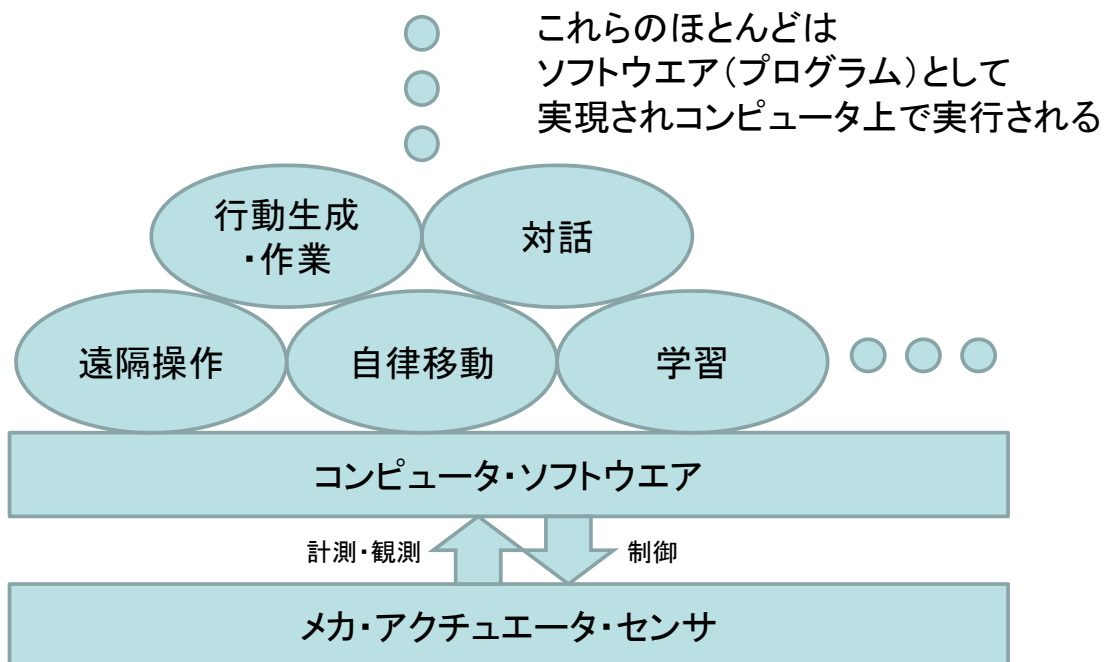
独立行政法人産業技術総合研究所
知能システム研究部門
タスクインテリジェンス研究グループ
安藤 慶昭

ロボットを構成する要素

- アクチュエータ
 - モータなど
- センサ
 - エンコーダ
 - 力・磁気・加速度など
- 機構(メカ)
- コンピュータ
 - ソフトウェア



ロボットを構成する要素



コンピュータとソフトウェア

パソコンとロボット

画面がないのでウィンドウは出てこないが、コンピュータ上でちゃんと動いている。

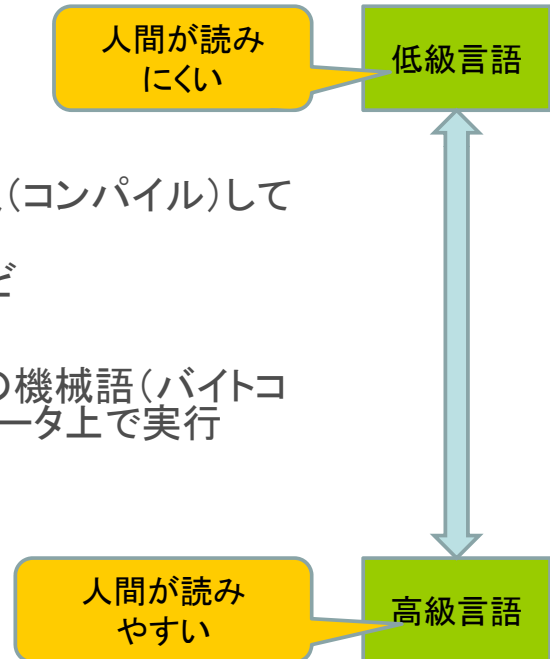


プログラム

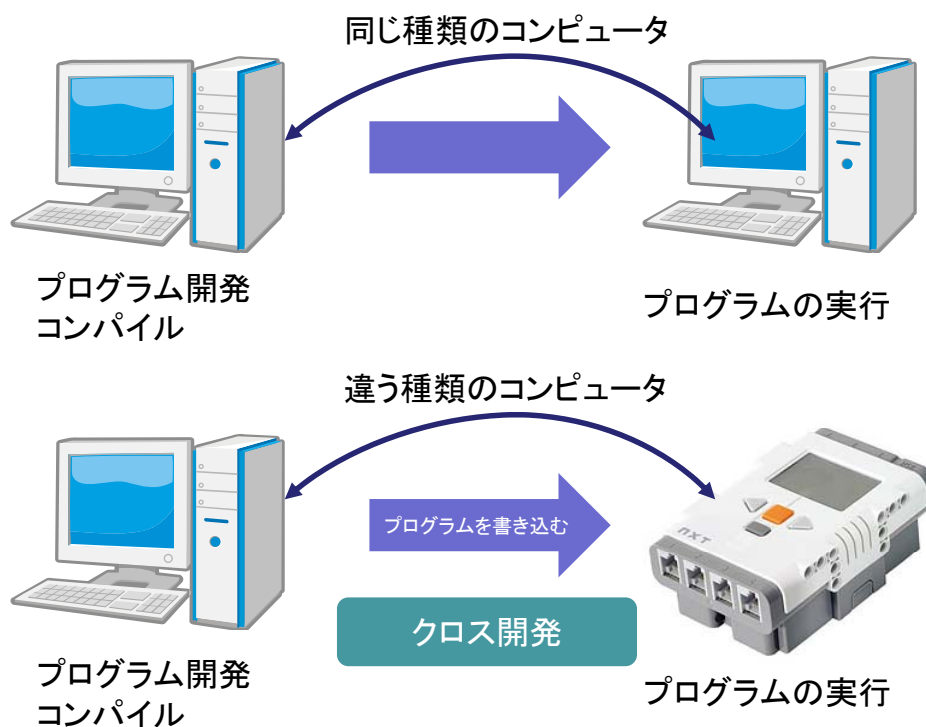
- プログラムとは？
 - コンピュータにやらせたいことを書いた手順書
 - アルゴリズム
 - コンピュータは機械語(マシン語)しか理解できない
- プログラミング言語
 - プログラムはプログラミング言語で書く
 - プログラミング言語にはいろいろな種類がある
 - それぞれ長所・短所がある(適材適所)

プログラミング言語

- 機械語
- アセンブリ言語
- コンパイル型言語
 - プログラムをすべて機械語に翻訳(コンパイル)してから実行
 - C、C++、FORTRAN、Pascal など
- 中間言語型
 - プログラムを仮想コンピュータ用の機械語(バイトコード)に翻訳してから仮想コンピュータ上で実行
 - Java、.NET(C#など)
- インタプリタ型
 - プログラムを一行ずつ実行
 - Ruby、Python、Perl、BASIC

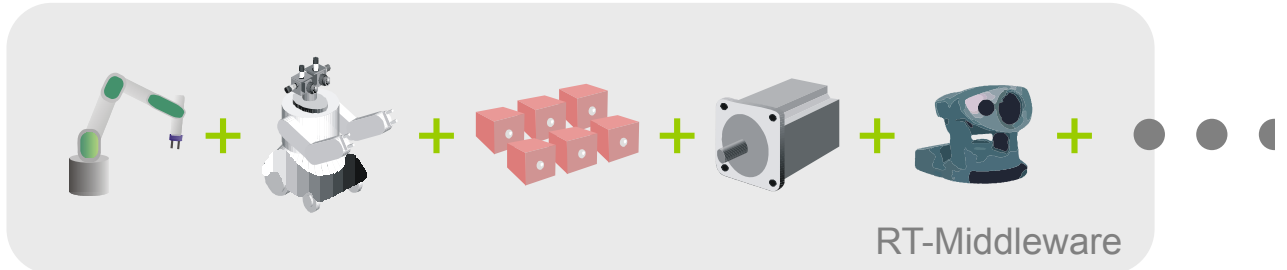


プログラム開発



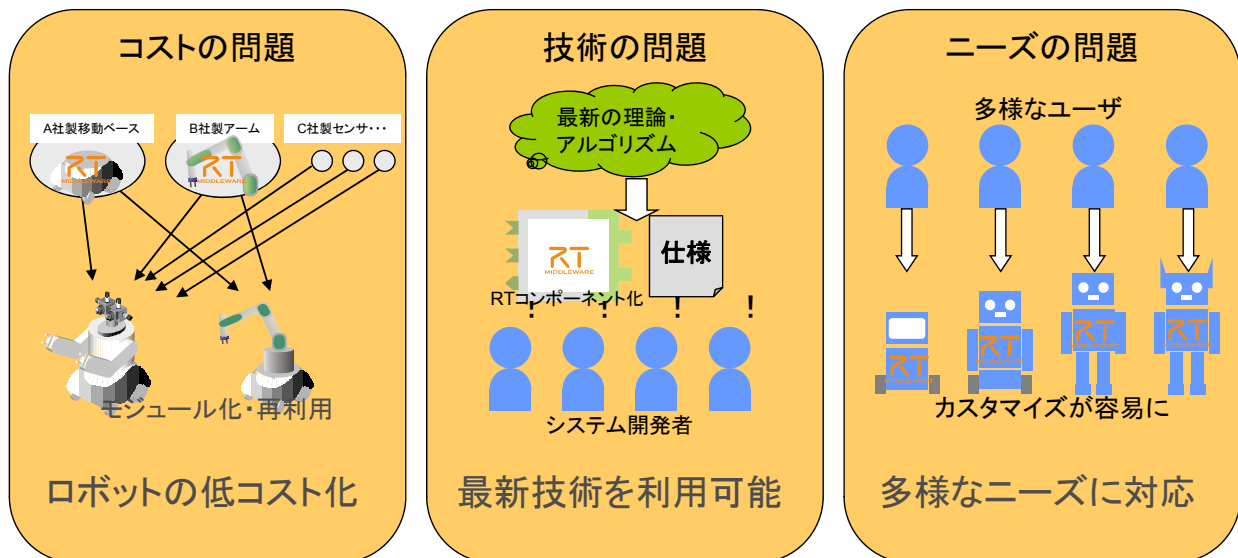
RTミドルウェアとは？

- RT = Robot Technology cf. IT
 - ≠Real-time
 - 単体のロボットだけでなく、さまざまなロボット技術に基づく機能要素をも含む (センサ、アクチュエータ, 制御スキーム、アルゴリズム、etc....)



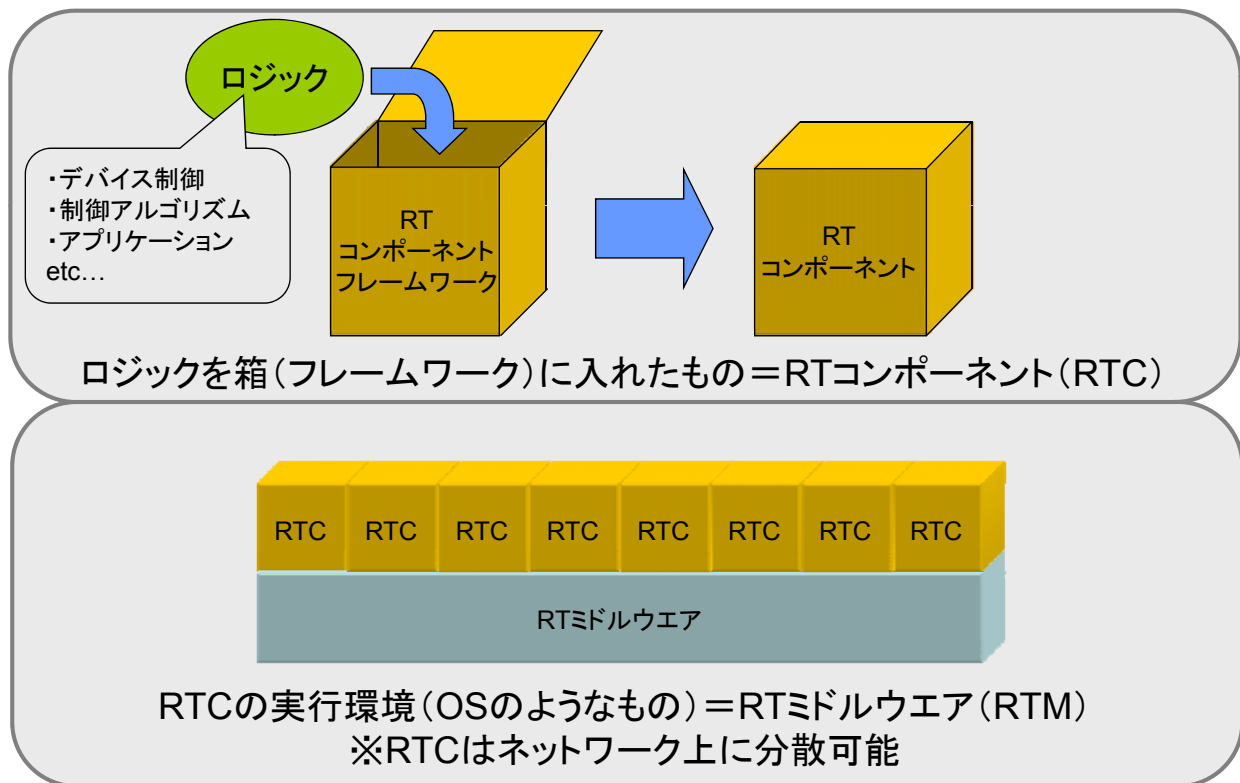
- RT-Middleware
 - RT要素のインテグレーションのためのミドルウェア
- RT-Component
 - RT-Middlewareにおけるソフトウェアの基本単位

RTミドルウェアの目的 モジュール化による問題解決



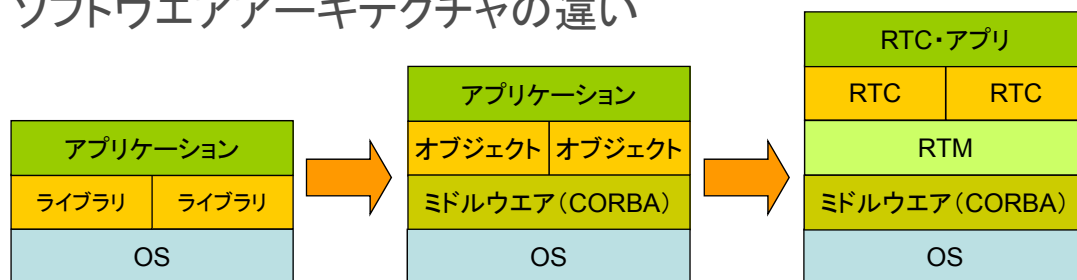
ロボットシステムインテグレーションのイノベーション

RTミドルウェアとRTコンポーネント



RTコンポーネントとは？

ソフトウェアアーキテクチャの違い



従来ソフトウェアから分散オブジェクトへ

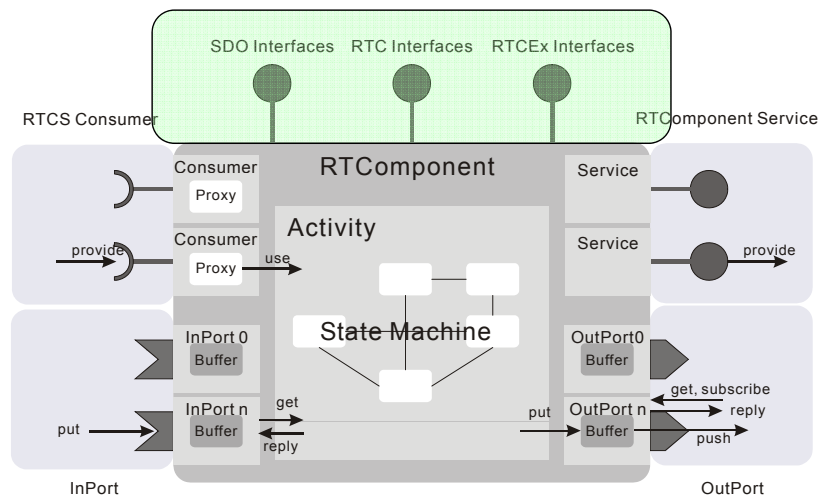
- ・ オブジェクト指向開発
- ・ 言語・OSの壁を越えて利用できる
 - インターフェースをIDLで定義
 - 各言語へ自動変換
 - OS、アーキテクチャの違いを吸収
- ・ ネットワーク透過に利用できる
 - 分散システムを容易に構築可能

分散オブジェクトからRTCへ

- ・ インターフェースがきちんと決まっている
 - IDLで定義された標準インターフェース
 - 呼び出しに対する振る舞いが決まっている(OMG RTC 標準仕様)
 - 同じ部品として扱える
- ・ コンポーネントのメタ情報を取得することができる
 - 動的な接続や構成の変更ができる
- ・ ロボットシステムに特有な機能を提供
 - 後述

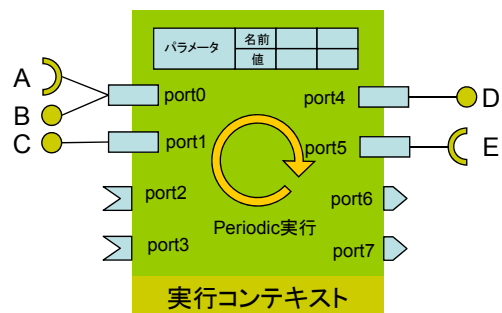
RTコンポーネントアーキテクチャ

- メタ情報取得
 - プロファイル
 - どんなコンポーネントか？



メタ情報取得

- メタ情報
 - コンポーネントのモデル仕様
- RTCのことはRTCに聞けばわかる (イントロスペクション機能)
 - コンポーネントの名前・タイプ
 - ポートの数・種類
 - ポートのインターフェース情報
 - ポート毎のプロパティ
 - パラメータ情報
 - 実行コンテキスト情報
- システムの動的構成に不可欠

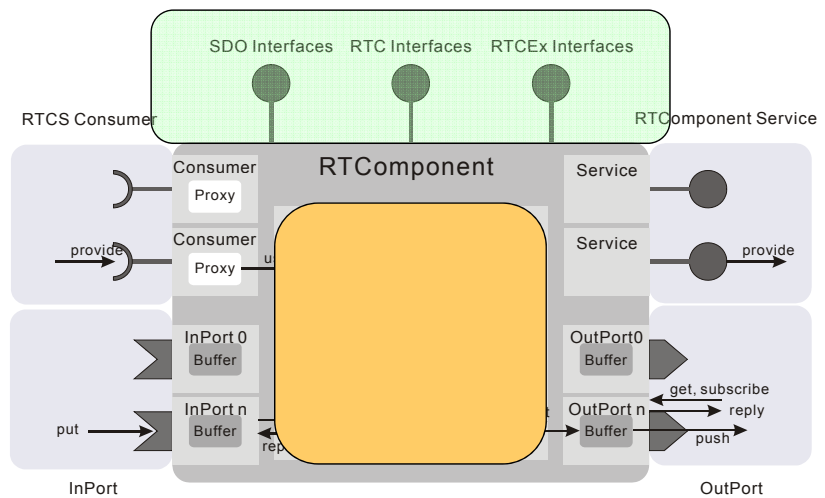


コンポーネントメタ情報の例

コンポーネント名	MyManipulator0
タイプ名	Periodic実行型
port0	Provide: A, Required: B
port1	Provide: C
Port2	DataPort: InPort, velocity, float x6
Port3	DataPort: InPort, position, float x6
Port4	Provide: D
Port5	Required: E
Port6	DataPort: OutPort, status int x1
Port7	DataPort: OutPort, velocity, float x6
実行コンテキスト	周期: 10ms
パラメータ	gain0(float x6), flag(int x1), dev_file(string)

RTコンポーネントアーキテクチャ

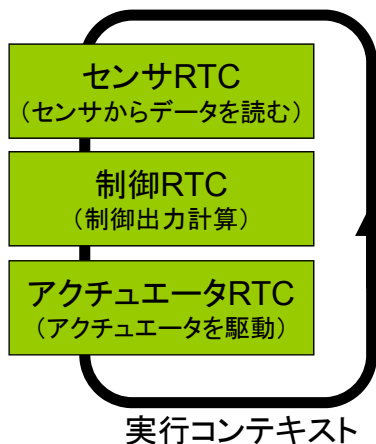
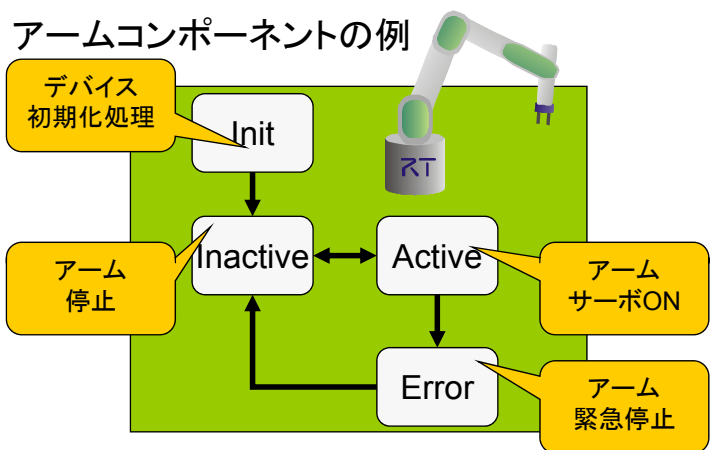
- メタ情報取得
 - プロファイル
 - どんなコンポーネントか？
- アクティビティ
 - ユーザ定義ロジックの実行



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

アクティビティ

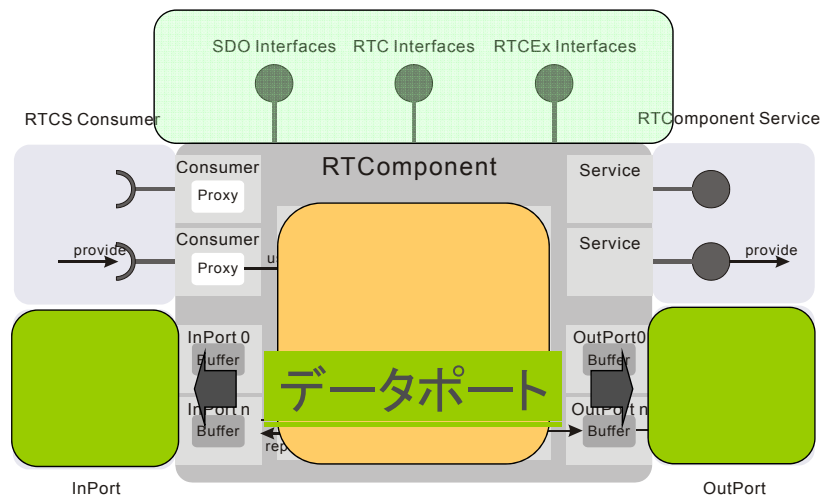
- ロジックを実行する部分
- 共通の状態遷移を持つ
 - 初期化
 - 非アクティブ (OFF状態)
 - アクティブ状態 (ON状態)
 - エラー (エラー状態)



別々に作成された複数のコンポーネントをシーケンシャルにリアルタイム実行し制御等を行うことも可能
→ 複合コンポーネント

RTコンポーネントアーキテクチャ

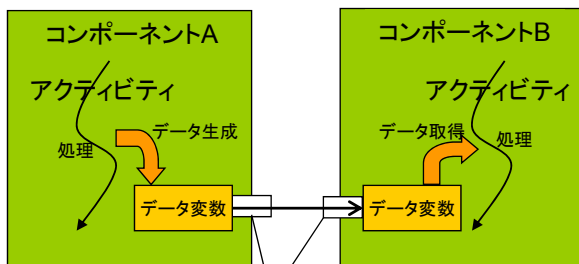
- メタ情報取得
 - プロファイル
 - どんなコンポーネントか？
- アクティビティ
 - ユーザ定義ロジックの実行
- データポート
 - Data Centric な相互作用



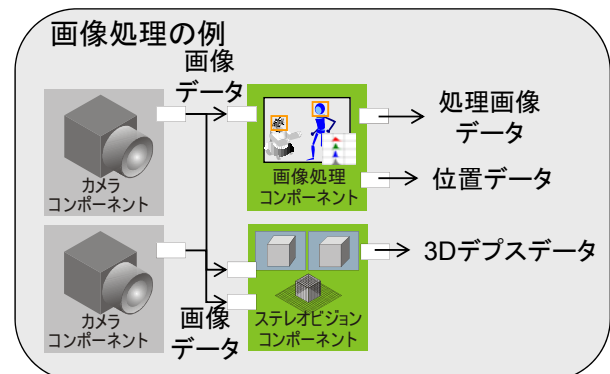
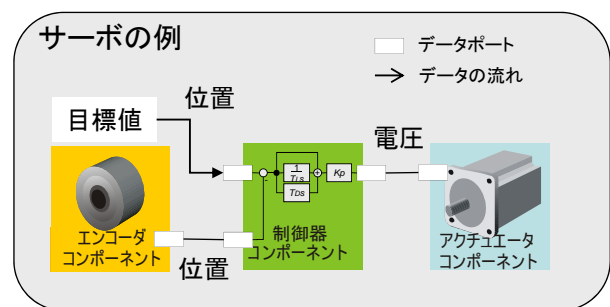
NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

データポート

- データ指向ポート
- 連続的なデータの送受信
 - 位置制御サーボ
 - 位置・電圧値
 - 画像処理
 - 画像データ
 - 処理結果
- 主にロボットの下位レベル処理に利用
- 同じデータ型のポート同士接続可能
- 動的に接続・切断可能

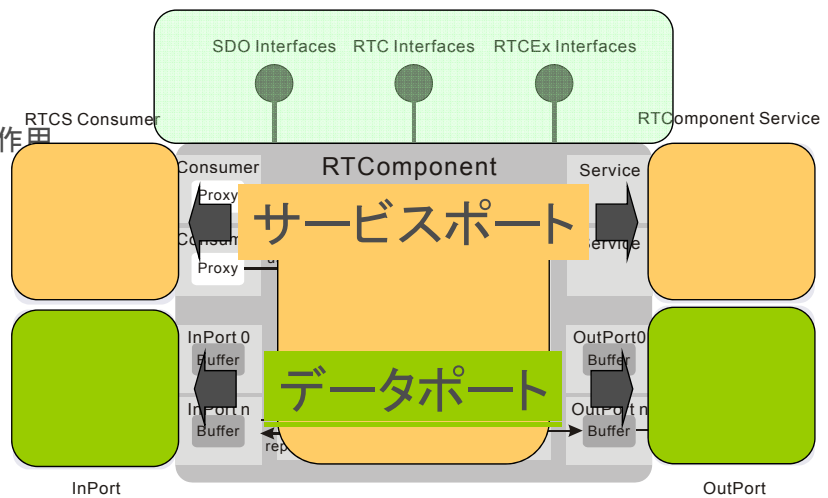


データが自動的に伝送される



RTコンポーネントアーキテクチャ

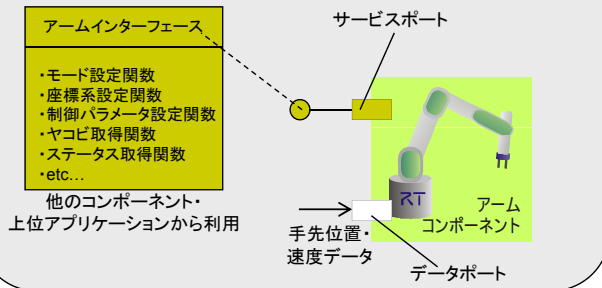
- メタ情報取得
 - プロファイル
 - どんなコンポーネントか？
- アクティビティ
 - ユーザ定義ロジックの実行
- データポート
 - Data Centric な相互作用
- サービスポート
 - request/response型相互作用



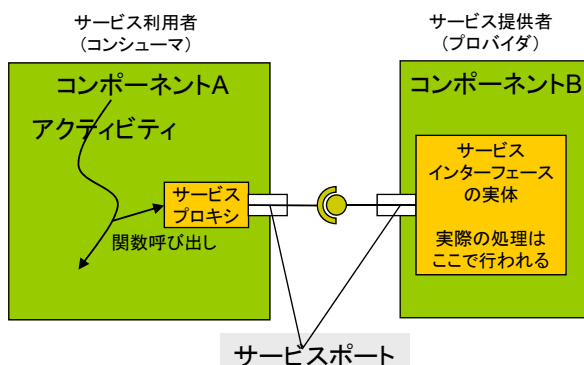
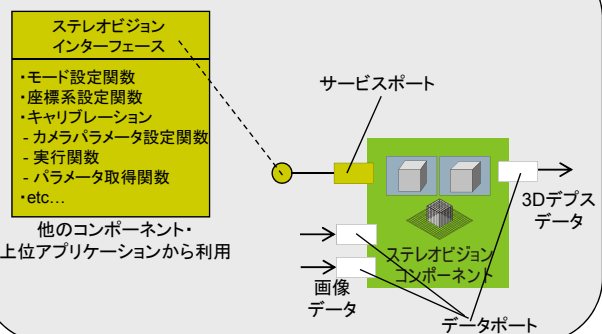
サービスポート

- 任意に定義可能なインターフェースを持つポート
 - コマンド・関数を自由に追加
 - 他のコンポーネントからアクセス可能
 - (本当は標準化したい)
- 内部の詳細な機能にアクセス
 - パラメータ取得・設定
 - モード切替
 - 処理の依頼と結果取得
 - etc...

アームの例

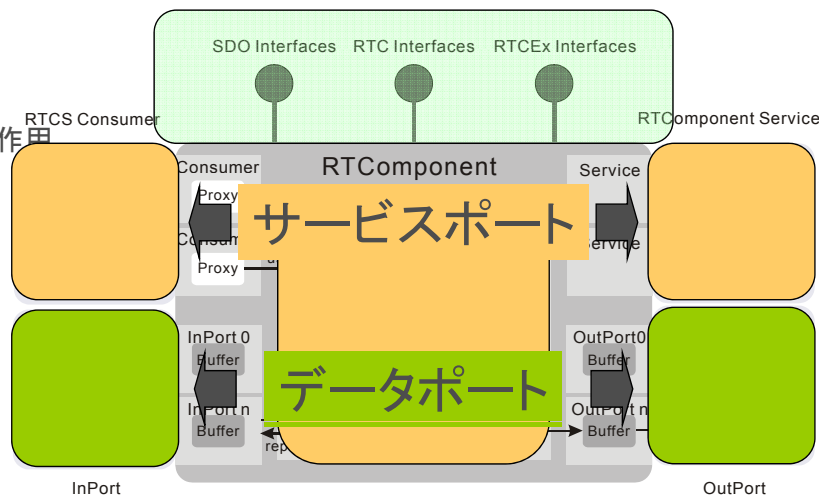


ステレオビジョンの例



RTコンポーネントアーキテクチャ

- メタ情報取得
 - プロファイル
 - どんなコンポーネントか？
- アクティビティ
 - ユーザ定義ロジックの実行
- データポート
 - Data Centric な相互作用
- サービスポート
 - request/response型相互作用
- コンフィギュレーション
 - ユーザ定義の設定



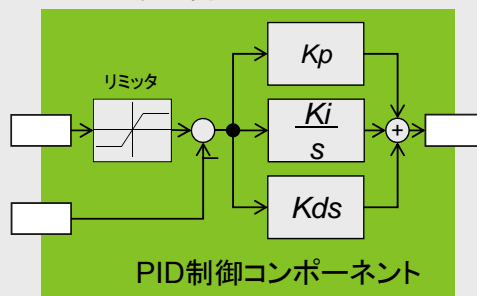
コンフィギュレーション

- コンフィギュレーション
 - パラメータを管理
 - コンフィギュレーションセット
 - セット名、名前:値のリスト
 - 複数のセットを保持
 - セットを切替可能

複数のセットを動作時に切り替えて使用可能

セット名	名前	値				
セット名	名前	値				

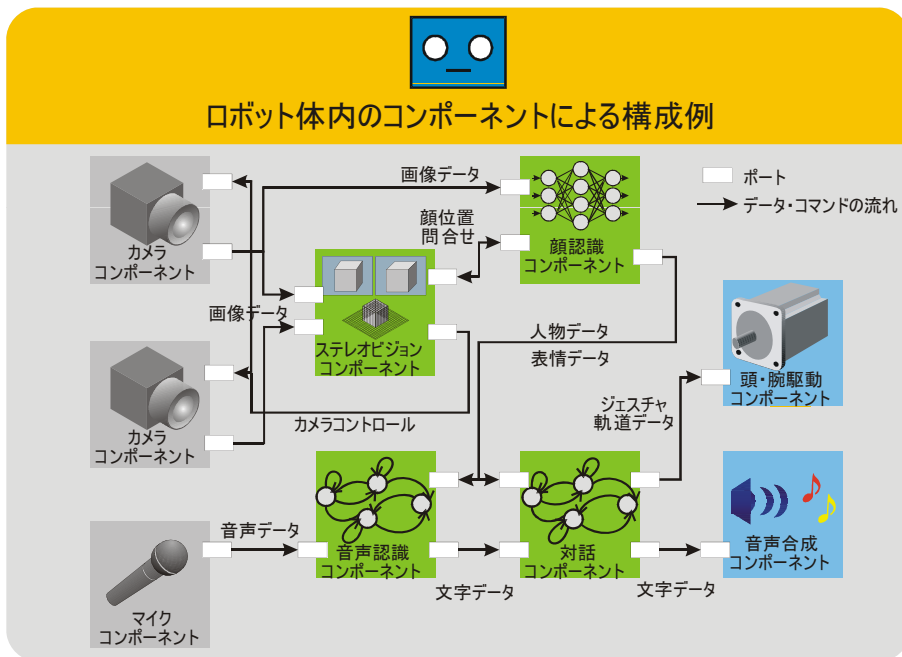
PIDコントローラの例



mode	名前	Kp	Ki	Kd	In _{max}	In _{min}
modeA	名前	0.6	0.01	0.4	5.0	-5.0
modeB	名前	0.8	0.0	0.01	10.0	-10.0
modeC	名前	0.3	0.1	0.31	1.0	-1.0

制御対象やモードに応じて複数のPIDゲインおよび入力リミッタ値を切り替えて使用することができる。動作中の切り替えも可能。

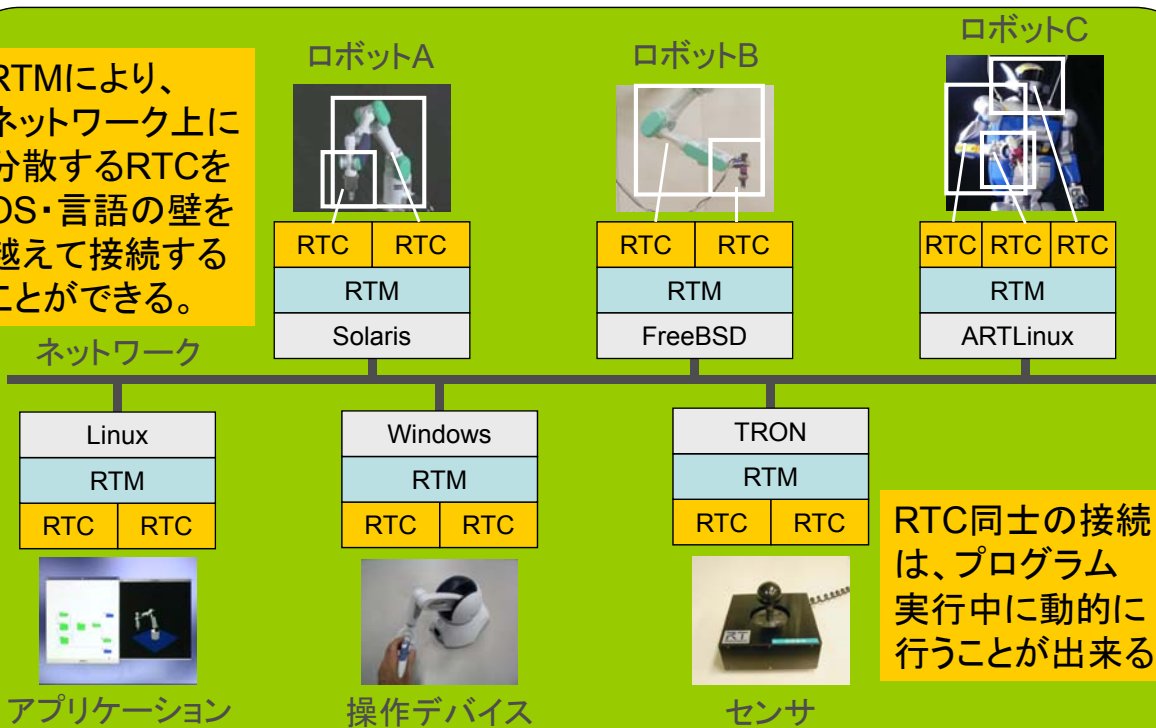
RTCの分割と連携



(モジュール)情報の隠蔽と公開のルールが重要

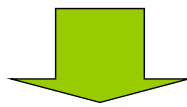
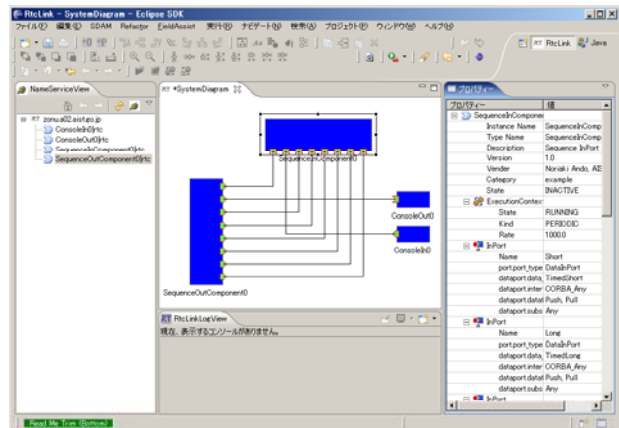
RTミドルウェアによる分散システム

RTMにより、ネットワーク上に分散するRTCをOS・言語の壁を越えて接続することができる。



開発環境

- rtc-template
 - RTコンポーネント・コードジェネレータ
 - コンポーネントモデルを定義することで、雛形プログラムの自動生成を行い、コンポーネント作成者を中身のプログラミングに集中させることができる
- RtcLink
 - システム作成者はこのツールを使って簡単にシステム設計ができる



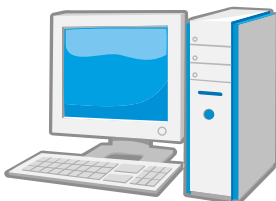
RTC・RTM統合開発環境の整備

RTC設計・実装・デバッグ、RTMによるインテグレーション・デバッグまでを一貫して行うことができる統合開発環境をEclipse上に構築

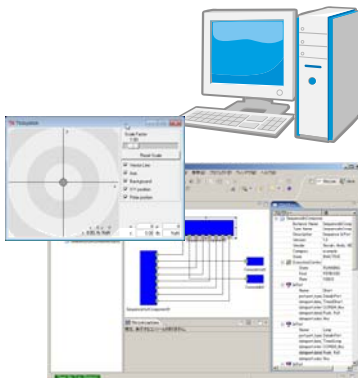
NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

今回のシステム

- ① LEGOで動かすプログラムをコンパイル
- ② コンパイルしたプログラムをLEGOに転送
- ③ LEGOのプログラムを起動



無線 (bluetooth) で通信



- ④ PCでLEGOと通信するRTコンポーネントを起動
- ⑤ ジョイスティックコンポーネントを起動
- ⑥ 2つのコンポーネントを接続して操作する




NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

まとめ

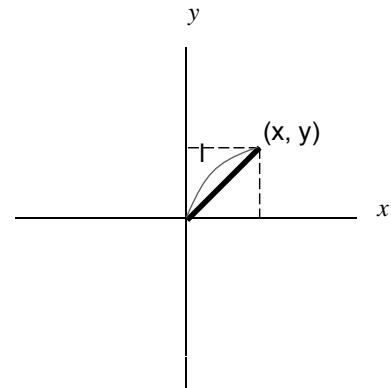
- RTミドルウェア
 - ロボットシステム用ソフトウェアプラットフォーム
- RTコンポーネント
 - ソフトウェア部品
- 利点
 - 構成を自由にかつ動的に決めることができる
 - 部品の再利用性の向上
 - システム構築が容易になる

インストール後のテスト

- ネームサーバの起動
 - 「スタート」→「プログラム」→「OpenRTM-aist」→「C++」→「examples」→「Start Naming Service」
- Eclipseを起動
 - Eclipseフォルダの中の  このアイコンをダブルクリック
- サンプルを実行
 - 「スタート」→「プログラム」→「OpenRTM-aist」→「Python」→「examples」→ 適当に起動

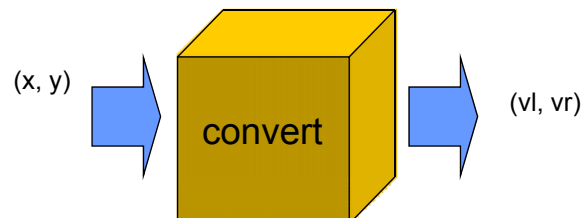
Pythonでのプログラミング

- 代入 (xに1を代入)
 - $x=1$
- 四則演算
 - $\times \rightarrow *$ 、 $\div \rightarrow /$
- 平方根($\sqrt{\quad}$)
 - `math.sqrt()`: 例 `math.sqrt(x*x + y*y)`
- arccos、arctan
 - `math.acos(x)`, `math.atan(x)`
- 円周率
 - `math.pi`
- ユークリッド距離
 - `math.hypot(x, y)` : 右図 $l = \sqrt{x^2+y^2}$



関数とは

ある値 (今回はジョイスティックの出力 x, y) を入れると別の値 (今回は左右の車輪の速度 v_l, v_r) を出してくれるもの



プログラムはたくさんの関数の集まりで構成されている。

サマーサイエンスキャンプ

NXTwayとBluetooth通信コンポーネント

金 泰成



目次

- 概要
- LEGO MINDSTORM NXTway組立て
- 環境説明
- Bluetoothニックネーム入力及びアップロード
- コンポーネント起動及び動作確認

概要

概要

- NXT MINDSTORMとOpenRTMのBluetoothコンポーネントと接続し、NXT MINDSTORMをコントロールする方法を説明する。
- NXT MINDSTORMに対するソフトウェアのインストール方法および基本的な操作、RTコンポーネントとの接続する方法を説明する。



LEGO MINDSTORM NXTway組立て

2010 サマーサイエンスキャンプ

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

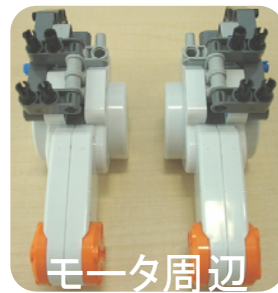
LEGO MINDSTORM NXTway組立て(1)

- 必要な部品
 - LEGO MINDSTORM Education NXT Base Set
 - Product ID : W979797
 - LEGO Education Resource Set
 - Product ID : W979648
 - NXT Gyro Sensor
 - Part No : NGY1044



LEGO MINDSTORM NXTway組立て(2)

- 組立て順番
 - モータ周辺
 - 車輪
 - センサ周辺
 - 本体
 - 接続



- 参照資料

- 提供した資料を参照して下さい
- http://lejos-osek.sourceforge.net/NXTway-GS_Building_Instructions.pdf

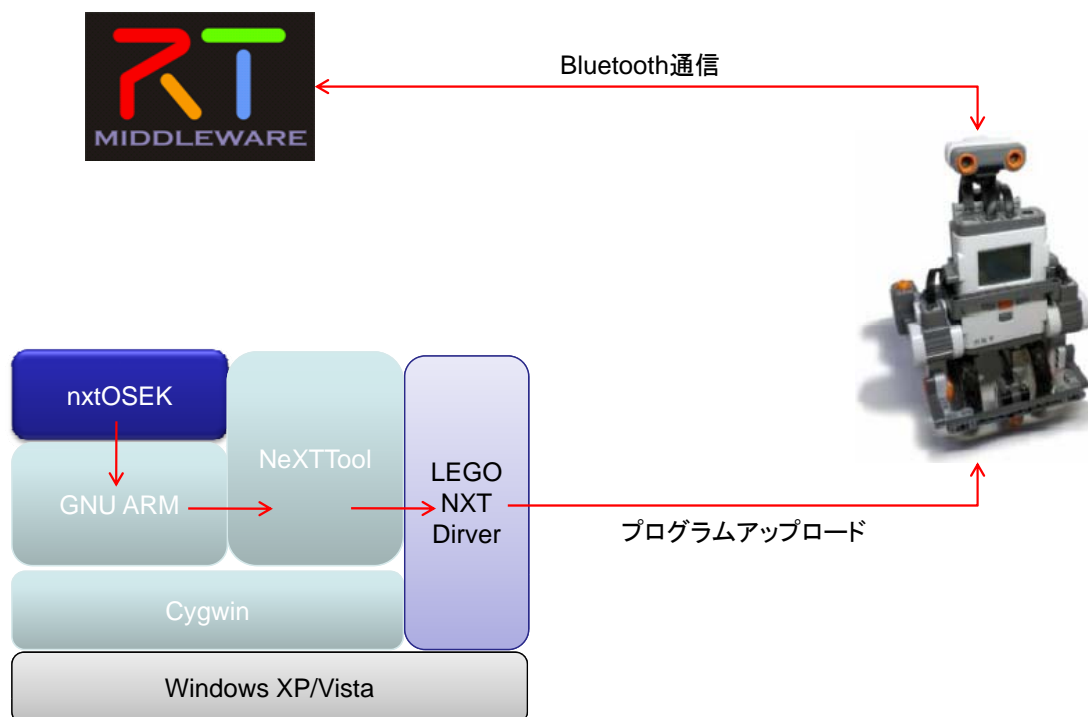


nxtOSEK環境説明

環境説明(1)

- Cygwin
 - Windows環境で各種のLinuxソフトウェアを実行できることである
- GNU ARM
 - NXTのARM7コアプロセッサ(AT91SAM7S256)に対応したGCCコンパイラパッケージである
- LEGO MINDSTORMS NXT Driver
 - LEGO社から提供されるNXTとのUSB通信ドライバーである
- NeXTTool
 - NXTと通信用PCコンソールで、*.rxn(アプリ)と*.rwn(ファームウェア)をNXTにアップできる
- 拡張NXTファームウェア
 - 標準NXTファームウェアをベースに機能拡張したものである
- nxtOSEK
 - C、C++ベースのNXT用ライブラリ及びプログラム

環境説明(2)



Bluetoothニックネーム入力 及びアップロード

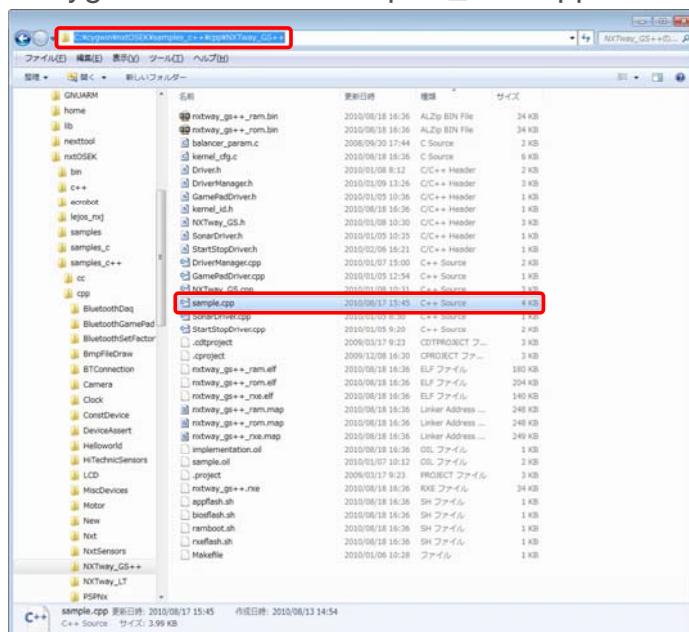
2010 サマーサイエンスキャンプ

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

Bluetoothニックネーム入力及びアップロード(1)

- Bluetoothニックネーム入力
- ファイルオープン

「c:¥cygwin¥nxtOSEK¥samples_c++¥cpp¥NXTway_GS++¥sample.cpp」



2010 サマーサイエンスキャンプ

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

Bluetoothニックネーム入力及びアップロード(2)

- Bluetoothニックネーム入力
 - ファイルオープン
 - メモ帳を利用してオープン



Bluetoothニックネーム入力及びアップロード(3)

- Bluetoothニックネーム入力
 - 関数検索(btConnection.connect)
 - メモ帳でオープンしたファイルに「Ctrl + F」を入力
 - 「検索する文字列」に「btConnection.connect」を入力して「次を検索」ボタンをクリック



Bluetoothニックネーム入力及びアップロード(4)

- Bluetoothニックネーム入力
 - ニックネーム入力
 - Bluetoothニックネームを入力する

```

//=====
// Main Task
TASK(TaskMain)
{
    // establish bluetooth connection with a PC to use a PC HID GamePad controller
    BTConnection btConnection(bt_Lcd, nxt);
    (void)btConnection.connect(BT_PASS_KEY);

    for (U32 i = 5; i <= Lcd::MAX_CURSOR_Y; i++)
    {
        lcd.cursor(0,5);
        lcd.putf("snsns", " ");
        lcd.disp();
        SetRelAlarm(Alarm4msec, 1, 4); // Set 4msec periodical Alarm for the drive event
    }

    while(1)
    {
        sonarDriver.checkObstacles(sonar);
        clock.wait(40); // 40msec wait
    }
}
//=====
// Main Task
TASK(TaskMain)
{
    // establish bluetooth connection with a PC to use a PC HID GamePad controller
    BTConnection btConnection(bt_Lcd, nxt);
    (void)btConnection.connect(BT_PASS_KEY, nickname);

    for (U32 i = 5; i <= Lcd::MAX_CURSOR_Y; i++) lcd.clearRow(i);
    lcd.cursor(0,5);
    lcd.putf("snsns", "TOUCH:START/STOP", "STAND IT UP AND", "WAIT FOR A BEEP.");
    lcd.disp();
    SetRelAlarm(Alarm4msec, 1, 4); // Set 4msec periodical Alarm for the drive event

    while(1)
    {
        sonarDriver.checkObstacles(sonar);
        clock.wait(40); // 40msec wait
    }
}

```

Bluetoothニックネーム入力及びアップロード(5)

- Bluetoothニックネーム入力
 - 上書き保存

```

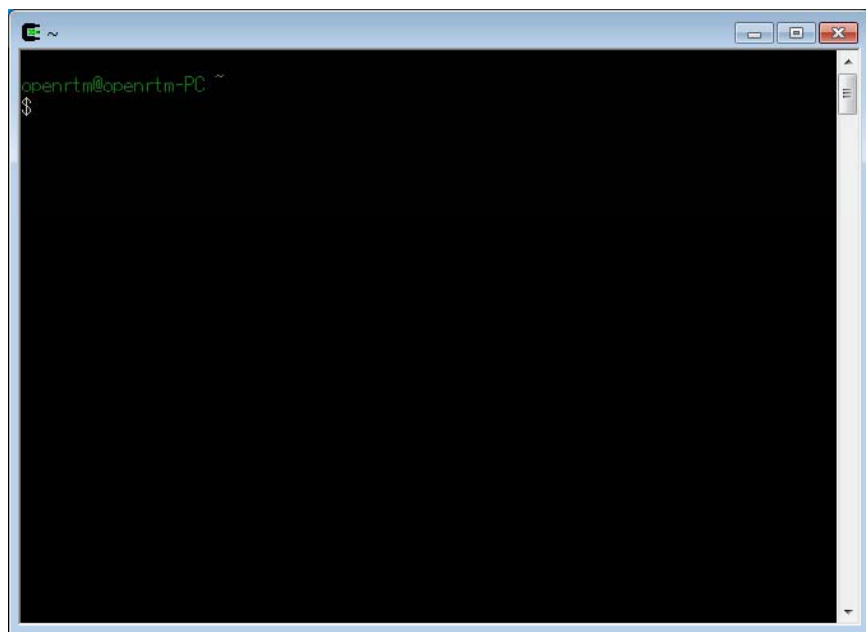
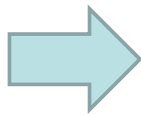
sample.cpp - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
 新規(N) Ctrl+N
 開く(O)... Ctrl+O
 上書き保存(S) Ctrl+S
 名前を付けて保存(A)...
 ページ設定(U)...
 印刷(P)... Ctrl+P
 メモ帳の終了(X)

#include "Nxt.h"
#include "Clock.h"
#include "Lcd.h"
#include "GamePad.h"
#include "Daq.h"
#include "Speaker.h"
#include "Bluetooth.h"

```

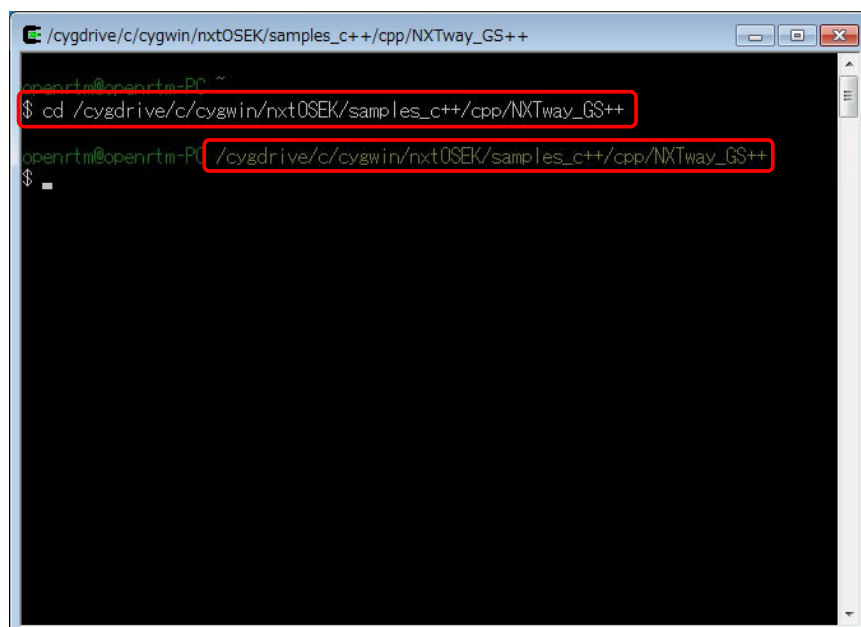
Bluetoothニックネーム入力及びアップロード(6)

- Cygwinを起動



Bluetoothニックネーム入力及びアップロード(7)

- フォルダに移動
「cd /cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++」



Bluetoothニックネーム入力及びアップロード(8)

- コンパイル
「make all」

```

/cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
openrtm@openrtm-PC ~
$ cd /cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
openrtm@openrtm-PC /cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
$ ls
Driver.h          GamePadDriver.h  SonarDriver.h
DriverManager.cpp Makefile          SonarDriver.cpp
DriverManager.h  NXTway_GS.cpp    StartStopDriver.o
GamePadDriver.cpp NXTway_GS.h      StartStopDriver.o

openrtm@openrtm-PC /cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
$ make all
Compiling ../../ecrobot/..toppers_osek/Makefile to Makefile.o
Compiling ../../ecrobot/..toppers_osek/SonarDriver.cpp to SonarDriver.o
Compiling ../../ecrobot/..toppers_osek/StartStopDriver.cpp to StartStopDriver.o
Compiling ../../ecrobot/..toppers_osek/DriverManager.cpp to DriverManager.o
Compiling ../../ecrobot/..toppers_osek/NXTway_GS.cpp to NXTway_GS.o
Compiling ../../ecrobot/..toppers_osek/sample.cpp to sample.o
Assembling ../../ecrobot/..lejos_nxi/src/nxtvm/platform/nxt/vectors.s to vectors.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/debug.S to debug.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/lego_nxt/sys_support.S to sys_support.o
Assembling ../../ecrobot/..ecrobot/c/nxt_binary_header.s to nxt_binary_header.o
Assembling ../../ecrobot/..ecrobot/c/nxt_entry_point.s to nxt_entry_point.o
Assembling ../../ecrobot/..ecrobot/c/ecrobot_init.s to ecrobot_init.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/cpu_support.S to cpu_support.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/cpu_support.oram to cpu_support.oram
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/irq.s to irq.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/irq.oram to irq.oram
Generating binary image file: nxtway_gs++_rom.bin
Generating binary image file: nxtway_gs++_ram.bin
Generating binary image file: nxtway_gs++.rxe

openrtm@openrtm-PC /cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
$

```

Bluetoothニックネーム入力及びアップロード(9)

- アップロード(USB接続後)
「sh ./rxeflash.sh」

```

/cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
tors.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/debug.S to debug.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/lego_nxt/sys_support.S to sys_support.o
Assembling ../../ecrobot/..ecrobot/c/nxt_binary_header.s to nxt_binary_header.o
Assembling ../../ecrobot/..ecrobot/c/nxt_entry_point.s to nxt_entry_point.o
Assembling ../../ecrobot/..ecrobot/c/ecrobot_init.s to ecrobot_init.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/cpu_support.S to cpu_support.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/cpu_support.oram to cpu_support.oram
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/irq.s to irq.o
Assembling ../../ecrobot/..toppers_osek/config/at91sam7s-gnu/irq.oram to irq.oram
Generating binary image file: nxtway_gs++_rom.bin
Generating binary image file: nxtway_gs++_ram.bin
Generating binary image file: nxtway_gs++.rxe

openrtm@openrtm-PC /cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
$ sh ./rxeflash.sh
Executing NeXTTool to upload nxtway_gs++.rxe...
nxtway_gs++.rxe=34240
NeXTTool is terminated.

openrtm@openrtm-PC /cygdrive/c/cygwin/nxtOSEK/samples_c++/cpp/NXTway_GS++
$

```

コンポーネント起動及び動作確認

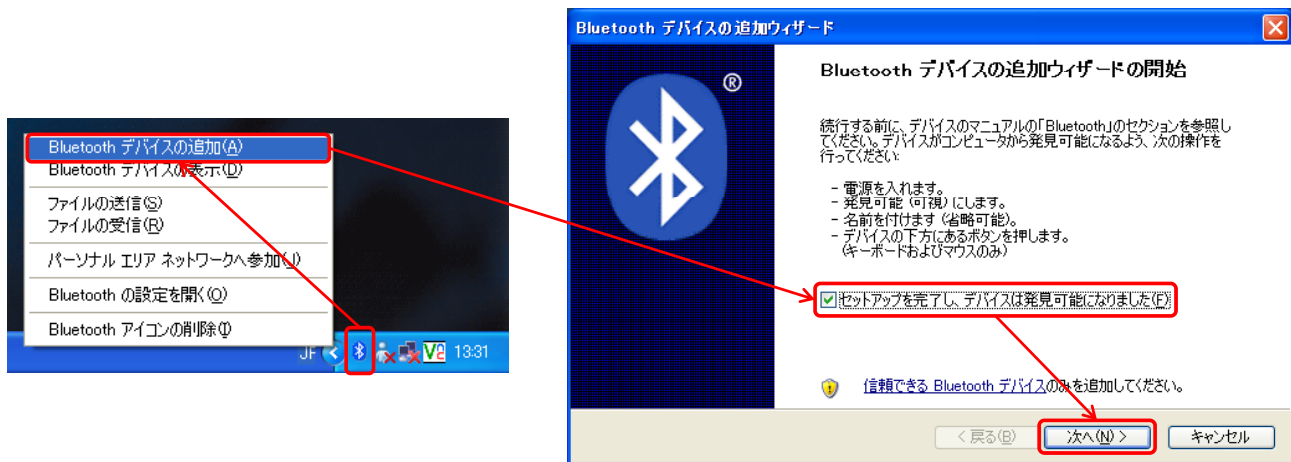
コンポーネント起動及び動作確認(1)

- NXT側
 - NXT起動(⑦番までした後、再起動する必要がある)



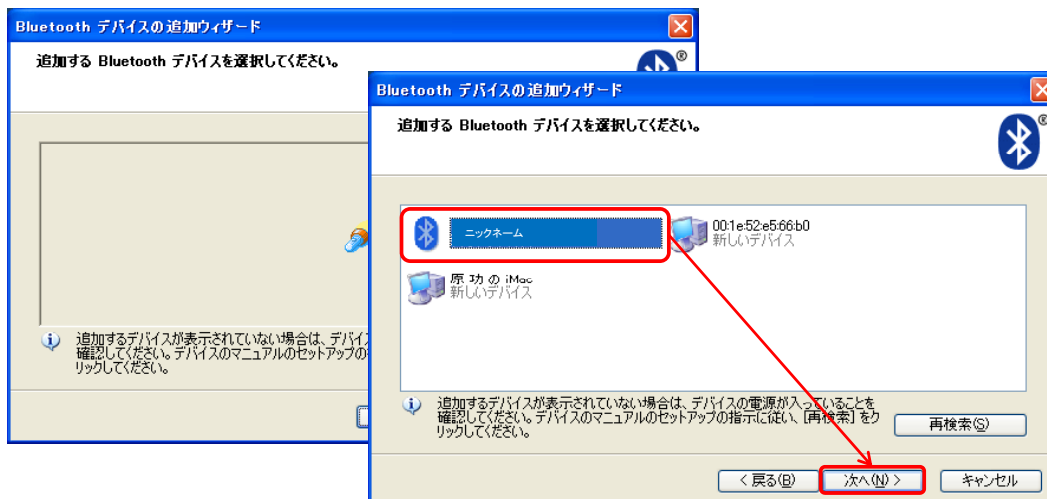
コンポーネント起動及び動作確認(2)

- PC側
 - Bluetooth接続



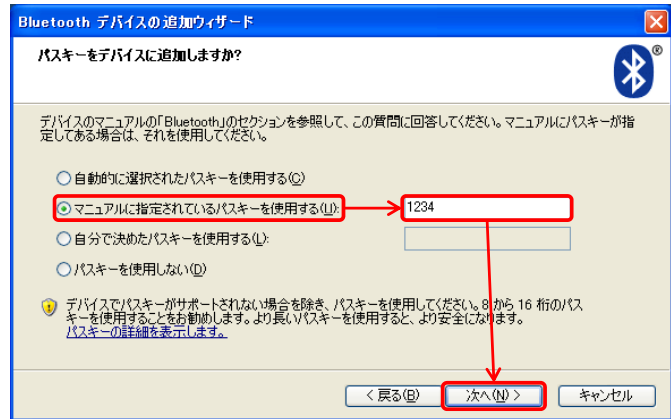
コンポーネント起動及び動作確認(3)

- PC側
 - Bluetooth接続
 - デバイス追加



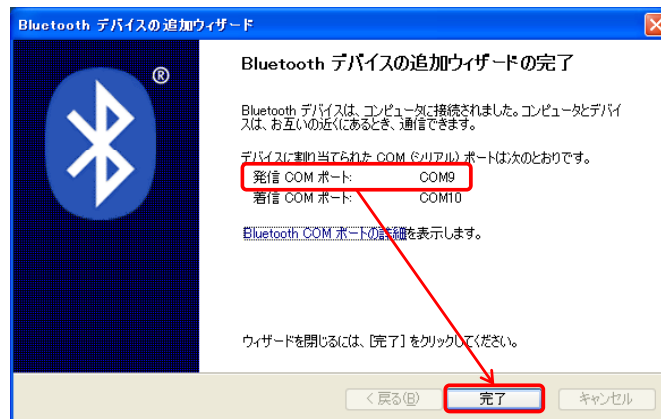
コンポーネント起動及び動作確認(4)

- PC側
 - Bluetooth接続
 - NXT側は⑦番の状態待機
 - コードに「1234」を入力



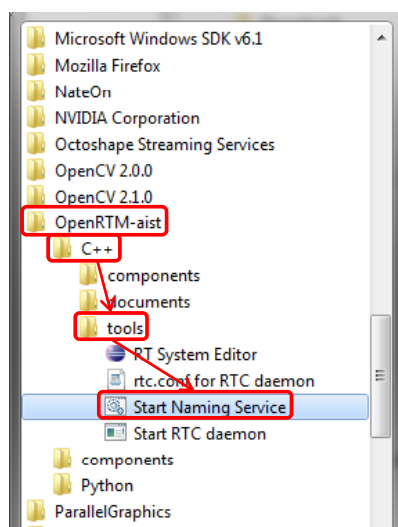
コンポーネント起動及び動作確認(5)

- PC側
 - Bluetooth接続
 - COMポート確認



コンポーネント起動及び動作確認(6)

- PC側
 - Naming Serviceを起動する



```

Start Naming Service
Starting omniORB omniNames: openrtm-PC:2809

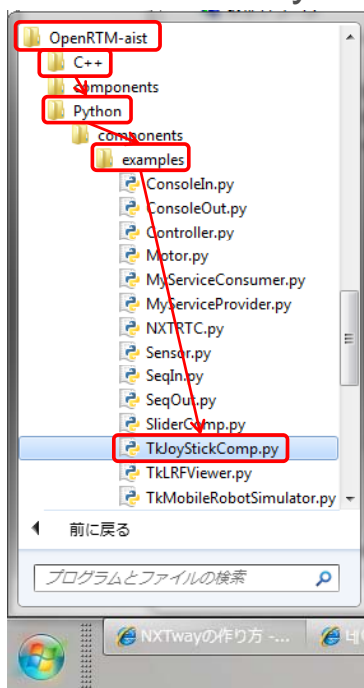
Thu Aug 19 11:37:19 2010:

Starting omniNames for the first time.
Wrote initial log file.
Read log file successfully.
Root context is IOR:010000002b00000049444c3a6f6d672e6f72672f436f734e616d696e672f
4e616d696e67436f6e746578744578743a312e30000001000000000000007000000010102000d00
00003139322e3136382e38332e310000f90a0b0000004e616d6553657276696365000300000000
00000800000010000000054544101000001c000000100000010001000100000010001050901
010001000000090101000354544108000000df986c4c010017fc
Checkpointing Phase 1: Prepare.
Checkpointing Phase 2: Commit.
Checkpointing completed.
  
```

※Naming Service : ネット上でどこでもRTコンポーネントをネームで見つけるようにするためのアプリケーション
詳しくは <http://www.nec.co.jp/middle/Orbix/shoukai/corba.html#4.5> を参照すること

コンポーネント起動及び動作確認(7)

- PC側
 - TkJoyStickCompを起動する

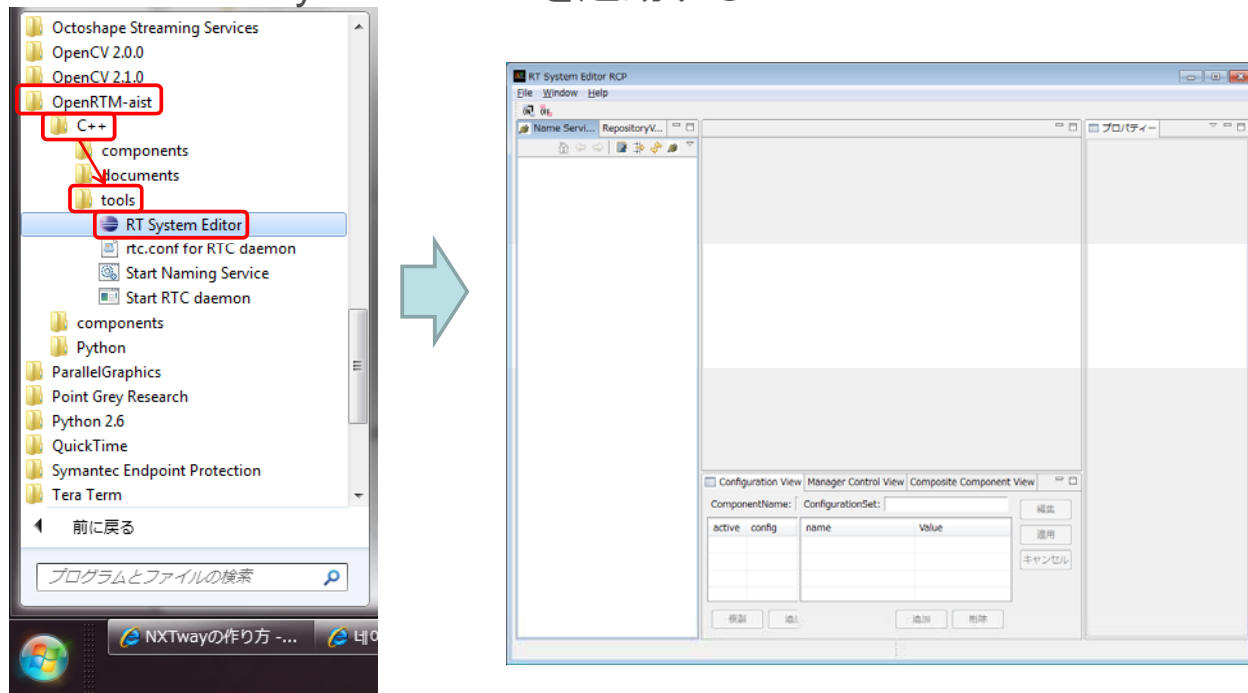


```

TkJoyStickComp.py
comp_args: TkJoyStick
C:\Program Files (x86)\OpenRTM-aist\1.0\examples\Python\TkJoyStick\TkJoyStick.py
:122: DeprecationWarning: integer argument expected, got float
circrange = range(circnum)
  
```

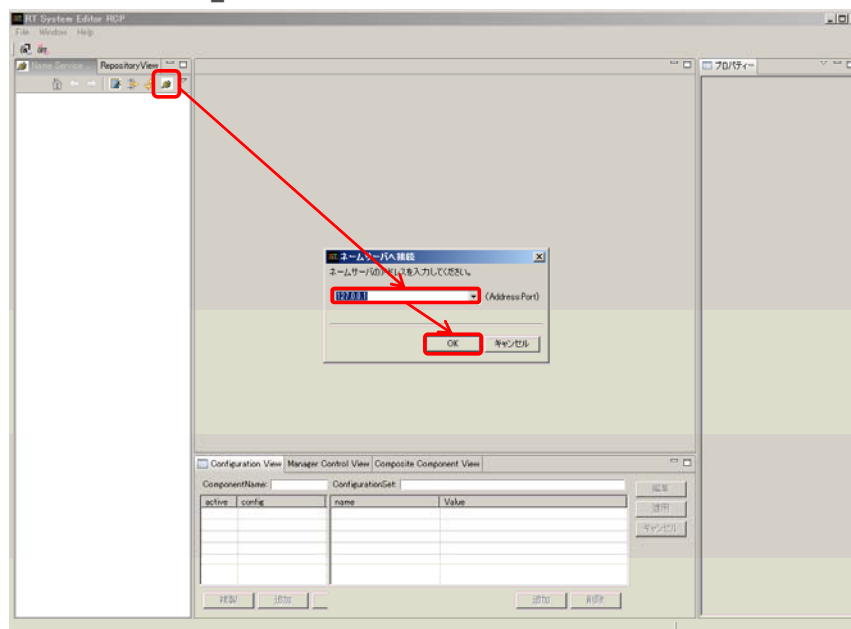

コンポーネント起動及び動作確認(8)

- PC側
 - RT System Editorを起動する



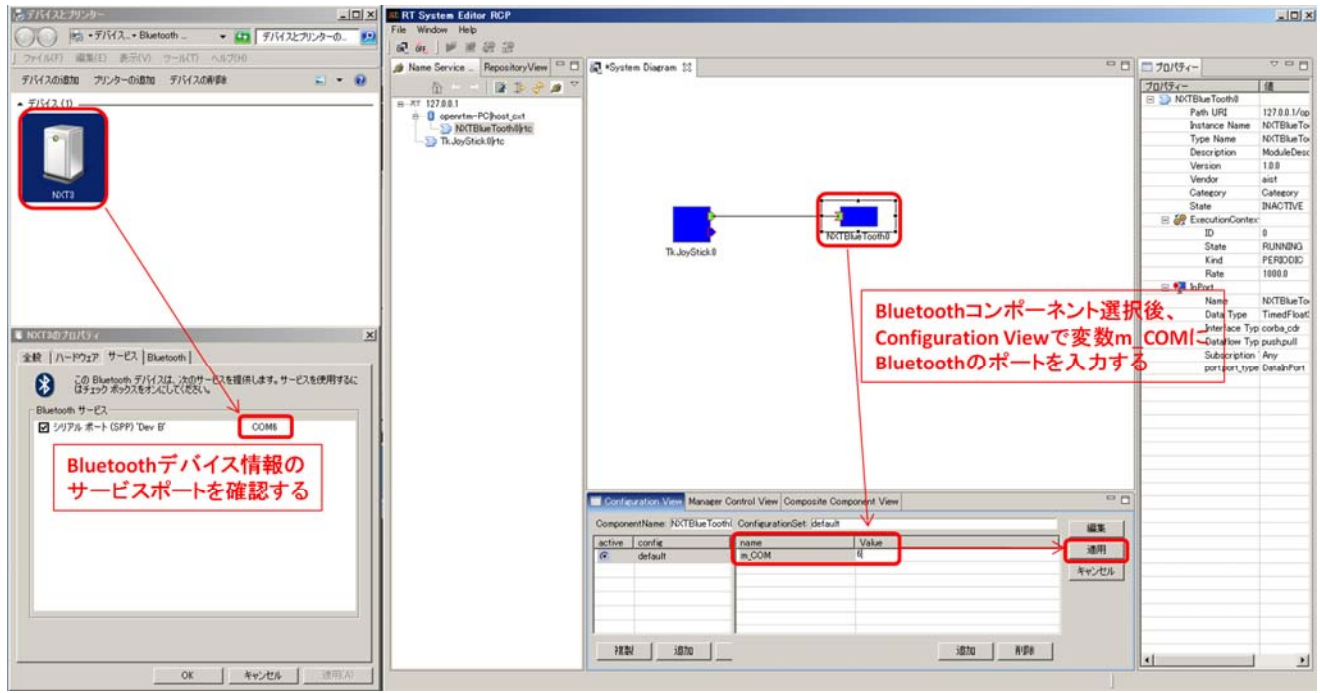
コンポーネント起動及び動作確認(9)

- PC側
 - ネームサーバ追加
「127.0.0.1」



コンポーネント起動及び動作確認(10)

- PC側
- ポートセット



コンポーネント起動及び動作確認(11)

- PC側
- Activateする

