KOBUKI

RTコンポーネント利用マニュアル

# RTミドルウエアの概要

## RTミドルウエアとは

RTミドルウエアとは，独立行政法人産業技術総合研究所（以下，産総研）が主体となって策定を続けているロボット技術用の基盤ソフトウエア用通信仕様である．この通信仕様は，UMLやCORBAの策定で有名な「Object Management Group, OMG」によって策定されている国際規格であり，今後，この規格に則った形でのソフトウエアが各国の様々な企業からリリースされることが予定されている．

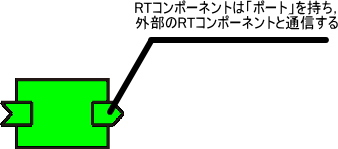
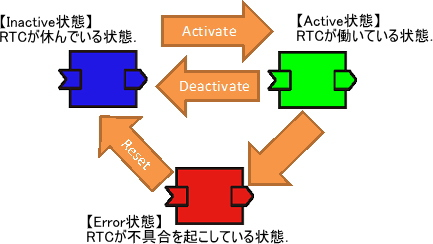
## OpenRTM-aistとは

OpenRTM-aistとは，産総研が主体となって開発している，RTミドルウエア規格に則ったソフトウエアである．RTミドルウエア規格を独自に拡張した機能を搭載しており，分散ネットワーク技術であるCORBAをベースに作られている．現状で，C++，Java，Pythonでの利用が可能であり，Windowsおよび各種Linuxでの使用，そして相互の接続性が確認されている．

## RTコンポーネントとは

RTコンポーネント（以下，RTC）とは，RTミドルウエア規格における概念で，アクチュエータやセンサなどのロボット要素をカプセル化するソフトウエアの単位を言う．RTCは，「ポート」と呼ばれるインターフェースを使って，他のRTCと接続およびデータの受け渡しを行う．ポートはタイプを持ち，同じタイプのポートは接続性が確認されている．

　また，RTCは「CREATED」，「ACTIVE]，「INACTIVE」，「ERROR」の4つの状態を持っており，ツール等で状態変化を行うことにより，RTCの制御開始，終了，エラー処理などが可能である．

# RTミドルウエアを使うメリットについて

## 制御プログラム開発の簡単化

RTミドルウエアを使うことにより，従来は製品仕様書やAPIのリファレンスマニュアルを熟読し，プログラムを作成する必要がありましたが，RTミドルウエアによってインターフェースが共通化されることにより，面倒な初期化作業などの多くを自動化することが出来るようになります．

　ユーザは，CD-ROMに同梱されるRTCをインストールし，環境に合わせて，COMポート番号など若干のコンフィグレーションを変更するだけでロボット製品を試すことが出来ます．また，テストに使用したRTCを，そのままユーザ自身のシステムに組み込むことが出来ます．

## ソフトウエアのインターフェースの共通化

RTミドルウエアを使うことにより，異なるインターフェースを持ったロボット製品を組み合わせて用いる場合に，従来はそれぞれに仕様書を熟読し，異なるライブラリを併用してプログラムを作成する必要がありましたが，RTミドルウエアによってインターフェースが共通化されることにより，複数製品の組合せ動作が簡単化されます．

# OpenRTM-aistのインストール (Windows版)

## OpenRTM-aistのダウンロード

産総研の公式サイト（http://www.openrtm.org/）から，OpenRTM-aistをダウンロードしてインストールします．

KobukiRTCはソースコードで提供されているので，C++コンパイラであるVisual C++ 2010，CMake 2.8，Doxygenが必要です．

# KOBUKI用RTコンポーネントのインストール

## libkobukiセットアップ

Kobuki用のライブラリをまずセットアップします．CD-ROMから，「libkobuki.msi」をダブルクリックしてインストールします．

Libkobukiをセットアップすると，スタートメニューからDemoソフトウエアを使ってKobukiの動作を確認する事が出来ます．ただし，デフォルトで，COM1を使ってKOBUKIに接続しますので，COM2以降を使う場合は，引数を与える必要があります．

## kobukirtcセットアップ

KobukiRTCはソースコードで提供されています．ビルドには，CMake2.8とDoxygenが必要です．

ビルドはOpenRTM-aistのC＋＋版の開発方法に従ってビルドします．

## テスト実行

KobukiRTCに付属のKobukiTestRTCを使って，RTCのテストを行います．

### ネームサーバーとRT System Editorの起動

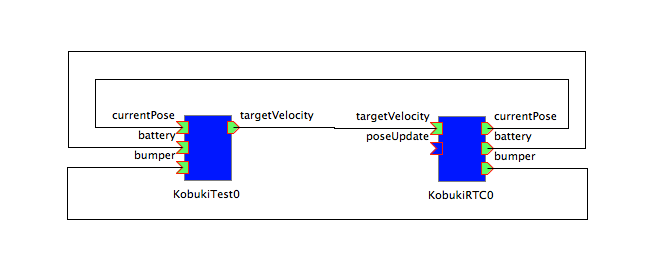
ネームサーバーとRT System Editorを起動します．RT System EditorではName Server Viewに使用するネームサーバー（通常はlocalhost）が表示されているようにします．

### KobukiRTCおよびKobukiTestRTCの起動

「KobukiRTCComp.exe」さらにテスト用の「KobukiTestRTCComp.exe」を実行して起動します．

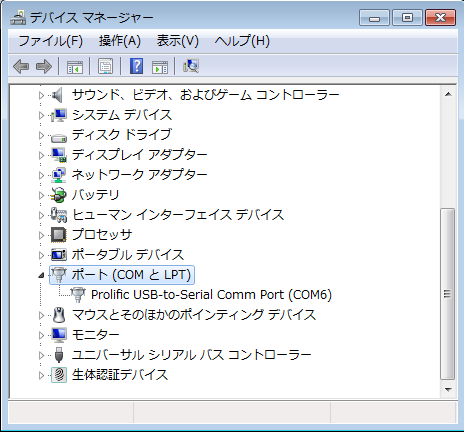
### 接続

同名のポートを接続していきます．すべて「push」になっていることを確認してください．



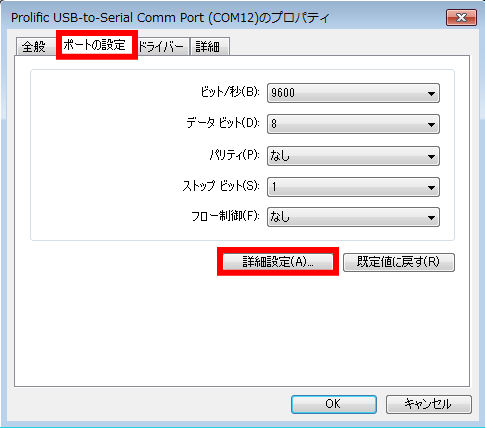
### コンフィグレーションの変更

KobukiをPCに接続し，接続に使用しているCOMポートをデバイスマネージャで確認します．

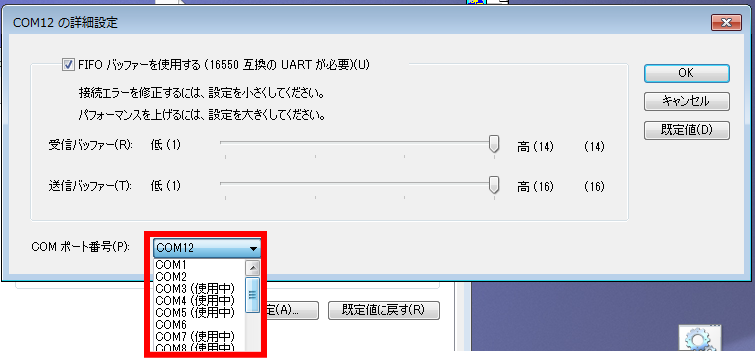


※このとき，COMポート番号がCOM9を超える場合は，COM番号の再設定が必要になります．

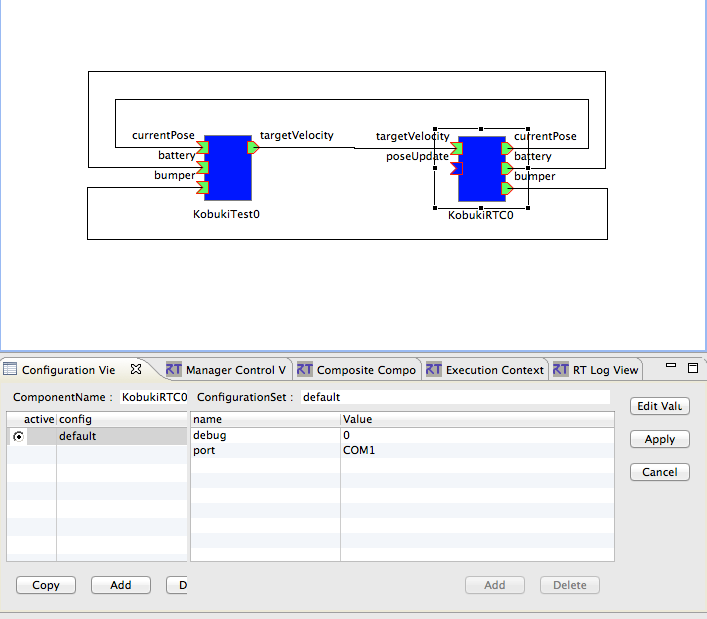
まず，デバイスマネージャ上で，対象となるCOMポートをダブルクリックし，COMポートのプロパティを開きます．次に，「ポートの設定」タブの「詳細設定」ボタンをクリックします．



このとき表示される詳細設定ダイアログ上で，COMポート番号をCOM9よりも若い番号に変更します．多くの番号に「使用中」と書かれていますが，起動しているデバイスマネージャ上で現在使われていないCOMポートであれば問題になりません．



次に，コンフィグレーションを変更します．まず，System Editor上のKobukiRTC0を選択状態にし，次にConfiguration Viewを表示させます．すると下図のような表示が出ますので，portのプロパティを変更します．下図ではCOM1に設定した場合の例を示しています．



③ portを変更

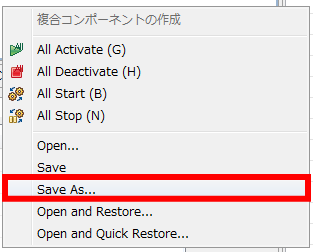
①選択

②Configuration View

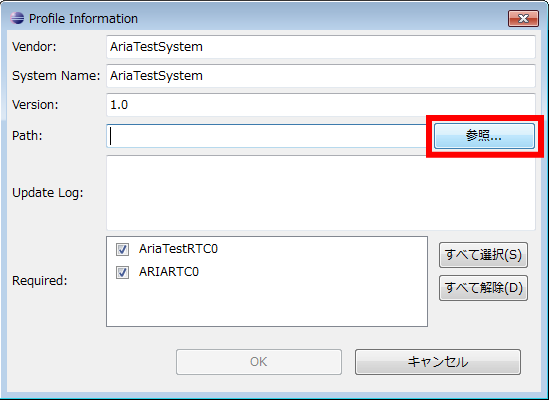
コンフィグレーションを変更したら，必ず「適用」ボタンをクリックします．これにより，ユーザの変更が反映されます．

### 接続とコンフィグレーションを保存

接続図を保存しておくことにより，次回からはRTCを立ち上げた後に，保存したファイルを選択することで，すべてのポートを接続し，コンフィグレーションを復元できます．System Editor上で右クリックし，「Save As...」を選択します．



表示されたダイアログに必要事項を記入します．「参照」ボタンをクリックして，保存するファイルの場所とファイル名を選択し，OKを選びます．



## KobukiTestRTC（テストプログラム）の使い方

KobukiTestRTCのコンソール画面にはロボットの位置，バッテリ電圧およびバンパーの反応が表示されています．矢印キーでロボットを操作し，スペースキーで停止させる事が出来ます．

ロボットの座標系は下図のように定義されています．最初に電源を入れた位置をゼロ点とします．

x

y

**

## KobukiRTCの仕様

KobukiRTCのデータポートの仕様をまとめます．

### 入力ポート

#### targetVelocity (TimedVelocity2D)

目標速度(x, y, θ) [mm/sec]．データ受信後，直ちに速度を変更する．

#### poseUpdated (TimedPose2D)

現在位置(x, y, θ) [mm] を更新する場合に，このポートにデータを送る．

### 出力ポート

#### batteryVoltage (TimedDouble)

バッテリ電圧 [V]

#### currentPose (TimedPose2D)

現在位置(x, y, θ) [mm]

#### bumper (TimedBooleanSeq)

接触センサの値．RIGHT，CENTER，LEFTの順序

### コンフィグレーション

#### port (文字列)

シリアルポート名．COM1や，/dev/ttyS0などを指定．