モジュール仕様書

自律と操作の融合モジュール

Ｖｅｒ．１.０.２

２０１１年6月２７日

セグウェイコンソーシアム

京都大学

改版履歴

Ver.	改版日	改版者	改版内容
1.0.0	2010/01/28	後藤清宏	新規作成
1.0.1	2011/06/24	根和幸	オープンソース対応
1.0.2	2011/06/27	根和幸	環境構築について説明を追記
1.0.3	2011/11/24	根和幸	表 5 を修正

目次

１．　はじめに

１．１．　本書適用範囲

移動ロボットを移動させる手段として，オペレータの『操作』による移動とロボット自身の『自律』走行による移動がある．我々のコンソーシアムではその操作と自律を滑らかに切り替えるモジュールを開発した．本書では，開発したモジュールの構成説明及びモジュールの使用手順を記述している．

１．２．　関連モジュール

本書で述べるモジュールは下表に示すモジュール(一例)と関連している．

表１　関連モジュール　一例

⑥ 移動知能(社会・生産分野)「自律と操縦が融合した電動立ち乗りモビリティシステム」	
モジュール名	備考
Beego 制御モジュール	移動ロボット“Beego”を制御するモジュール
RMP 制御 RTC	移動ロボット“Segway-RMP200”を制御するモジュール
ゲームパッド RTC	オペレータの操縦により速度指令値を出力する RTC
自律移動 RTC 群	目的地まで自律移動する速度指令値を出力する RTC 群
緊急停止デバイス管理モジュール	緊急停止ボタンが押された場合停止の指令値を出すモジュール

１．３．　本書の対象者

本書は RT ミドルウェア(以下 RTM と呼ぶ)，RT コンポーネント(以下 RTC と呼ぶ)を用いたロボットシステム開発者を対象に記述されており，RTM，RTC や関連ツールに関する一般的な知識を持つことを前提とする．

OpenRTM-aist Official Website：
http://www.openrtm.org/openrtm/ja/content/openrtm-aist-official-website

２．　RTC 詳細

２．１．　概要

本知能モジュールは，オペレータの『操作』とロボット自身の『自律』走行とを融合し，ロボットの移動をスムーズに行わせるモジュールである．

本モジュールを使用することで，モジュールのつなぎ換え動作なしで容易に自律と遠隔の切り替えが可能となる．また，緊急停止モジュールより指令があると最優先で止まるような速度を出力する．システム構成図を図１に示す．

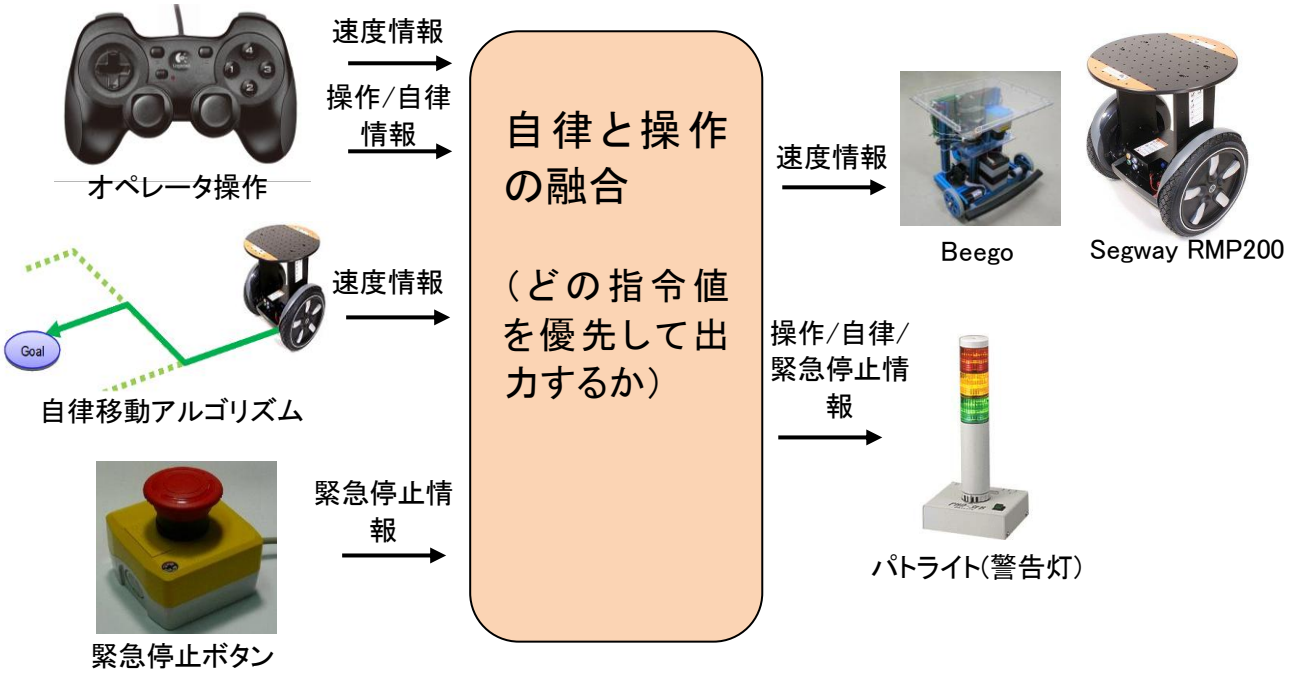

図１　自律と操作の融合モジュールシステム構成図

２．２．　使用環境設定

本書で述べる RTM システムは以下に述べる環境でその動作確認を行っている．ただし，オープンソースで公開しているため，再コンパイルすれば他環境でも動作可能であると考えられる．

表２　動作確認した環境設定

動作 OS	Windows 7 32bit / Ubuntu 10.04 32bit
開発言語	C++
コンパイラ	Visual Studio 2008 / gcc
RT ミドルウェア/バージョン	OpenRTM-aist-C/C++-1.0.0 RELEASE
依存パッケージ	特になし(但し，入出力に用いるデバイスの中には必要なものあり)
モジュール操作	RT System Editor
本書内で使用した PC	Vaio-Z(VGN-Z90NS)

２．３．　CommandSelectorComp

本節では，図１の自律と操作の融合の部分に対応するモジュールである CommandSelector モジュールについての機能概要，モジュール詳細について述べる．

２．３．１．　機能概要

このモジュールは，ゲームパッドからの速度指令値もしくは，自律移動モジュール群の出力結果の速度指令値を入力し，それらの値を融合し最適な速度指令値をロボットに送る役割をしているモジュールである．Ver.1.0.0 では，融合ではなく自律の入力と遠隔の入力の切り替えとなっている．ゲームパッドからは，速度指令値とは別にもう一つポートがあり，自律と操作を切り替えるデータを送信する出力ポートがある．CommandSelector モジュールにその情報の入力ポートがあり，ゲームパッドのボタンを使用することで，コンポーネントのつなぎ換えの動作なく容易に操作指令と自律移動指令を交換できる．また，緊急停止デバイス管理モジュールとつなぐことで，緊急停止ボタンが押された時，ロボットに停止動作の指令を送信すると共に，自律/操作/停止の表示を行っているパトランプの表示灯の変更も行う．詳細は緊急停止デバイス管理モジュール説明書を参照されたい．

２．３．２．　ポート情報

図 2 に示す CommandSelector モジュールの諸元を以下に示す．

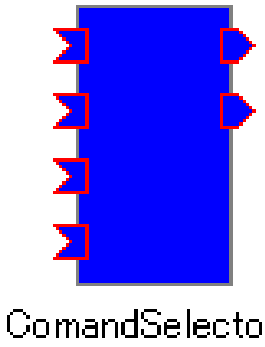

図２　CommandSelector モジュールの入出力ポート

○データポート(InPort)

表３　InPort 一覧

ポート名	型	説明
velocity_manual (操作)	IIS::TimedVelocity2D ※1	操作用モジュール群の出力した速度指令値を入力 v：並進速度[m/s]，w：回転速度[rad/s]
velocity_auto (自律)	IIS::TimedVelocity2D ※1	自律走行用モジュールの出力した速度指令値を入力 v：並進速度[m/s]，w：回転速度[rad/s]
select_auto (切り替え)	TimedBoolean	自律/操作を切り替える． （true: 自律，　false: 操作）

ポート名	型	説明
emo_state (緊急停止)	TimedBoolean	緊急停止ボタンより USB 接続デジタル入出力ボードを介し指令(true)が来ると manvel，autovel より優先的に v=0，w=0 を出力

○データポート(OutPort)

表４　OutPort 一覧

ポート名	型	説明
velocity_selected (出力速度)	IIS::TimedVelocity2D ※1	自律もしくは操作による速度指令値の出力値 v：並進速度[m/s]，w：回転速度[rad/s]
mode (選択されたモード)	TimedLong	選択されたモードを出力. 1 :非常停止，　2：操作，　3: 自律走行

○Configuration

表５　Configuration 一覧

ConfigurationSet	Name	Default Value	説明
default	timeout	1000 [msec]	timeout[msec]以上速度の更新がないと停止する．[msec]単位で値を入力する

※１　IIS::TimedVelocity2D の構造体については，モジュールと同梱している idl を参照のこと.

３．　モジュール使用方法

３．１．　環境の整備

○OpenRTM-aist のインストール

モジュールを使用するには OpenRTM-aist 1.0.0 が動作する環境を整える必要がある。以下の手順で使用する環境に応じてインストールを行う。環境変数については，下記マニュアルに従い設定を行えばよく，追加で設定すべき項目はない.

・C＋＋

【ダウンロード】http://www.openrtm.org/openrtm/ja/node/849

【マニュアル】　http://www.openrtm.org/openrtm/ja/node/999

○RT SystemEditor をインストール

【ダウンロード】http://www.openrtm.org/openrtm/ja/node/941

【マニュアル】　http://www.openrtm.org/openrtm/ja/node/676

３．２．　コンパイル方法

・Windows 環境(VC++2008)でのコンパイル方法

1. コンパイルしたいモジュール名のフォルダに移動する.
2. copyprops.bat を実行する
3. (モジュール名)_vc9.sln をダブルクリックして，VC++のソリューションを開く.
4. Ctrl+F5 を押してビルドする.
5. /(モジュール名)Comp フォルダに(モジュール名)Comp.exe が生成されているので，適当な場所にコピーする.

以上.

・Ubuntu 10.04 でのコンパイル方法

1. コンパイルしたいモジュール名のフォルダにある src フォルダに移動する.
2. Terminal で make を実行する.
3. (モジュール名)Comp が生成されているので，適当な場所にコピーする.

３．３．　起動手順・使用方法

この節では，具体的な手順を説明する．なお，必要な CommandSelectorComp を含め，Comp ファイルと rtc.conf は同一フォルダ上にあるものとする.

３．３．１．　ハードウェア準備

今回は例として，遠隔操作はゲームパッド(Logicool 社)を使用する．緊急停止スイッチとパトランプ

(パトライト社)は，DACS2600(DACS 技研)の USB 接続デジタル入出力ボードと接続する．それぞれの詳細は説明書に記述のため，ここでは省略する．

３．３．２． 起動手順

(1) ネームサーバの起動

・コマンドプロンプト画面で「rtm-naming」と入力し，ネームサーバを立ち上げる．

注)Ubuntu であれば，デフォルト立ち上がっているネームサーバを利用してもよい．もし，独自の設定で利用したい場合には，まず「sudo killall omniNames」と打ち込み，ネームサーバを落としたのち，再度ネームサーバを立ち上げる必要がある．

(2) rtc.conf の調整

・同じ PC でネームサーバを立ち上げているため，corba.nameservers: 127.0.0.1:2809 のままでよい．もし，別 PC なら IP アドレスを変更する．

(3) RT System Editor を起動する

(4) Comp ファイルの実行

・CommandSelectorComp を実行．また，今回は例として同コンソーシアムが提供している GamePad モジュール，緊急停止デバイス管理モジュールを起動する．ただし，それぞれの説明書を読み，環境を事前に整えておくこと．環境が整ってないままモジュールを起動するとエラーが起きる．

(5) RTSystemEditor 上でのモジュール接続

・次に示す図３のようにモジュールを接続する

３．３．３． モジュール構成例

モジュールの構成例を図３に示す．ここでは 3.2.2(4)で述べたモジュールを起動した例である．全て接続する必要はなく単体だけでもエラーは起こらず起動できる．なお，図 3 では簡単化のため CommnadSelectorComp の上から 2,3 番目の入力ポートに GamePadComp からの速度指令値を接続している．本来は 3 番目のポートには自律走行させるための速度指令値を入力する．そのように設定し，移動ロボットに 1 番目の出力ポートを接続すると操作と自律を切り替えた速度指令値が送信できるようになる．

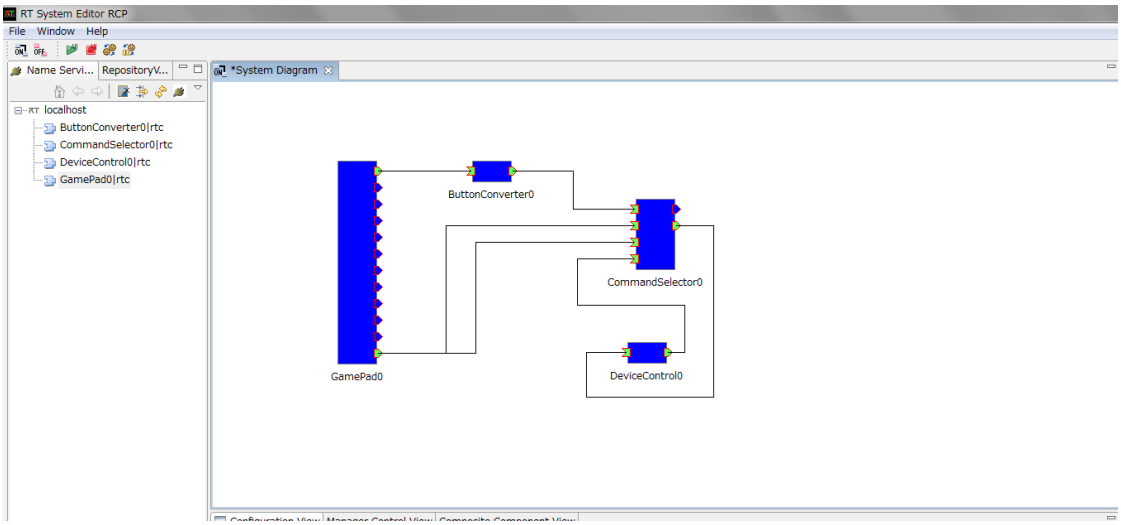

図３　モジュール構成例

３．３．４．　使用方法

(1) デバイス系の電源をつけ，USB を接続する．
　(ア) USB 接続デジタル入出力ボードに電源が入っていることを確認し，USB を接続する
　(イ) GamePad の USB も接続する

(2) DeviceControlComp はシリアル通信を行うため，デバイスマネージャや dmesg の結果を参考に portname のシリアルポート名があっているか確認する
(3) 全てのモジュールをアクティブにする．
　※エラーの場合はモジュールが赤くなる．
(4) 成功の場合は，全てがアクティブの黄緑色になり表示灯が点灯する．ゲームパッドの“1”のボタンを押すと遠隔操作となり表示灯は緑色が点灯し，ジョイスティックを傾けない限りロボットは停止する．“2”のボタンを押すと自律に切り替わるため，表示灯は黄色になり自律で走行を始める．緊急停止ボタンを押すと赤色ランプが点灯し，停止する．緊急停止ボタンのロックを解除しない限り，ジョイスティックを傾けても走行することはない．

３．３．５．　注意事項

○CommandSelectorComp 単体でも起動可能．しかし，動作確認方法がない．
○アクティブの際，一つでもエラーが起こり，モジュールが赤くなるとモジュールは全て起動しなおす必要がある．
○その他のデバイス依存のライブラリはインストールしておくこと．
○ロボットの最大速度を調整する機能はないため，ロボットの最大速度，ゲームパッドの出力の最大速度，自律移動モジュールの最大速度を確認しておくこと．
○ゲームパッドは「PlayStation のコントローラと USB 変換」とのセットの場合は Logicool, Logitech のゲームパッドの機種とボタン配置が異なるため使用できない可能性がある．