

次世代ロボット知能化技術開発プロジェクト
ロボット知能ソフトウェア再利用性向上技術の開発

来訪者受付システム
(RS003)

RH 取扱説明書

0.5 版

2011 年 7 月 13 日

RTC 再利用技術研究センター

目次

| | |
|---|----|
| 1. はじめに..... | 0 |
| 1.1. 目的..... | 0 |
| 1.2. 本書での書式..... | 0 |
| 1.3. 用語の定義、略語..... | 0 |
| 1.4. 参考資料..... | 0 |
| 2. システム概要..... | 1 |
| 2.1. ハードウェア構成..... | 1 |
| 3. 準備..... | 2 |
| 3.1. 本章の構成..... | 2 |
| 3.2. 1.H/W 環境構築..... | 2 |
| 3.2.1. 1-1.RH 制御 PC の内蔵方法..... | 2 |
| 3.3. 2.S/W 環境構築..... | 3 |
| 3.3.1. 2-1.動作環境..... | 3 |
| 3.3.2. 2-2.インストール手順..... | 5 |
| 3.3.3. 2-3.OpenRTM-aist-1.0.0-RELEASE (C++)..... | 5 |
| 3.3.4. 2-4.OpenRTM-aist-Python-1.0.0-RELEASE..... | 6 |
| 3.3.5. 2-5.RTSystemEditor・RTCBuilder-1.0.0-RELEASE..... | 7 |
| 3.3.6. 2-6.rtcree・rtshell..... | 7 |
| 3.3.7. 2-7.OpenCV2.0..... | 7 |
| 3.3.8. 2-8.MOXA Uport 1130 ドライバ..... | 7 |
| 3.3.9. 2-9.python visual..... | 8 |
| 3.3.10. 2-10.OpenHRI..... | 8 |
| 3.3.11. 2-11.ARTToolKit2.27.1..... | 8 |
| 3.3.12. 2-12.URG ライブラリ 0.8.2..... | 8 |
| 3.4. 3.ダウンロード・コンパイル..... | 9 |
| 3.4.1. 3-1.ダウンロード..... | 9 |
| 3.4.2. 3-2.コンパイル..... | 9 |
| 3.5. 4.各種設定..... | 10 |
| 3.5.1. 4-1.環境設定ファイル..... | 10 |
| 3.5.2. 4-2.作業に関する動作設定..... | 11 |
| 3.5.3. 4-3.移動に関する動作設定..... | 11 |
| 3.5.3.1. 目的地の座標を変更と追加..... | 11 |
| 3.5.4. 4-4.音声発話に関する動作設定..... | 12 |
| 3.5.5. 4-5.天井地図画像の作り方..... | 13 |
| 4. 使用方法..... | 14 |
| 4.1. 本章の構成..... | 14 |
| 4.2. 1.事前動作確認..... | 14 |
| 4.2.1. 1-1.RH の実機動作確認..... | 14 |
| 4.2.1.1. デバイスの認識状況確認..... | 14 |
| 4.2.1.2. 環境設定..... | 14 |
| 4.2.1.3. 動作確認..... | 15 |
| 4.2.2. 1-2.天井画像による推定自己位置の確認..... | 15 |
| 4.3. 2.RH 操作方法..... | 16 |
| 4.3.1. 2-1.RH 起動..... | 16 |
| 4.3.1.1. RH 制御 PC の起動..... | 16 |
| 4.3.1.2. アーム制御 PC、台車制御 PC の起動..... | 17 |
| 4.3.1.3. 遠隔操作 PC の起動..... | 17 |
| 4.3.1.4. 遠隔操作 PC 設定..... | 17 |
| 4.3.1.4.1. ネームサーバの削除と起動..... | 17 |

| | | |
|------------|---------------------------------|----|
| 4.3.1.4.2. | 遠隔操作 PC の動作準備 | 17 |
| 4.3.1.4.3. | 遠隔操作 PC のモジュール起動 | 18 |
| 4.3.1.4.4. | eclipse 起動 | 18 |
| 4.3.1.5. | RH 制御 PC のモジュール起動 | 19 |
| 4.3.1.5.1. | ネームサーバの削除と起動 | 19 |
| 4.3.1.5.2. | 天井カメラ(NM33-N-UVC)設定 | 19 |
| 4.3.1.5.3. | 天井ナビゲーションモジュール起動 | 19 |
| 4.3.1.5.4. | AR マーカ認識モジュール起動 | 19 |
| 4.3.1.6. | アーム制御 PC のモジュール起動 | 19 |
| 4.3.1.6.1. | ネームサーバの削除と起動 | 19 |
| 4.3.1.6.2. | アーム制御モジュール起動 | 20 |
| 4.3.1.7. | 台車制御 PC のモジュール起動 | 20 |
| 4.3.1.7.1. | ネームサーバの削除と起動 | 20 |
| 4.3.1.7.2. | アーム制御モジュール起動 | 20 |
| 4.3.1.7.3. | RT System Editor 確認 | 21 |
| 4.3.1.7.4. | 制御端末との接続 | 22 |
| 4.3.2. | 2-2.RH 操作 | 22 |
| 4.3.2.1. | 各端末からの操作 | 22 |
| 4.3.2.2. | 各種スクリプトによる操作 | 22 |
| 4.3.2.3. | ロボットの推定自己位置確認 | 23 |
| 4.3.2.4. | ゲームパッド操作 | 24 |
| 4.3.2.5. | バッテリー残量のチェック | 24 |
| 4.3.3. | 2-3.RH 休止・再開 | 25 |
| 4.3.4. | 2-4.RH 終了 | 25 |
| 4.3.4.1. | コンポーネントの終了 | 25 |
| 4.3.4.2. | 再起動又はシャットダウン | 25 |
| 5. | メンテナンス | 26 |
| 5.1. | 本章の構成 | 26 |
| 5.2. | 1.充電方法 | 26 |
| 5.2.1. | 充電の手順 | 26 |
| 5.2.2. | 注意事項 | 28 |
| 5.3. | 2.SSD/HDD の交換 | 29 |
| 5.3.1. | 交換の手順 | 29 |
| 6. | トラブルシューティング | 30 |
| 6.1. | 動作指令を出しても RH が動き出さない | 30 |
| 6.2. | RH の挙動や言動がおかしい | 30 |
| 6.3. | 推定位置が真値から大きく外れてしまったときの対処法 | 30 |
| 7. | その他 | 31 |
| 7.1. | 延期要求 | 31 |
| 7.2. | その他の要件 | 31 |
| 7.3. | 特記事項 | 31 |

1. はじめに

1.1. 目的

本書は、リファレンスハードウェア(以降、RH と略す)の取扱方法を記載した文書である。

1.2. 本書での書式

本文書で使用している記号・書式の目的を下表に示す。

表.書式一覧

| No. | 記号・書式 | 目的 |
|-----|-------|------|
| 1 | ※ | 注意書き |
| 2 | 赤色の文字 | 注記 |

1.3. 用語の定義、略語

表.用語の定義、略語一覧

| No. | 表記 | 意味 |
|-----|----------|-----------------------|
| 1 | 本システム | 来訪者受付システム |
| 2 | プロジェクト | 次世代ロボット 知能化技術開発プロジェクト |
| 3 | センター | RTC 再利用技術研究センター |
| 4 | 現時点 | 本書作成時点(2010/07/02) |
| 5 | 在籍者 | センター内勤務者 |
| 6 | OS | 動作対象プラットフォーム |
| 7 | RTミドルウェア | OpenRTM-Aist |
| 8 | RTM | RTミドルウェア |
| 9 | OSS | オープンソースソフトウェア |
| 10 | 障害物 | 人及び、人が一人で運ぶ事の出来る物体 |

1.4. 参考資料

本書を作成するにあたり参照した文書・資料を下表に示す。

表.参考資料一覧

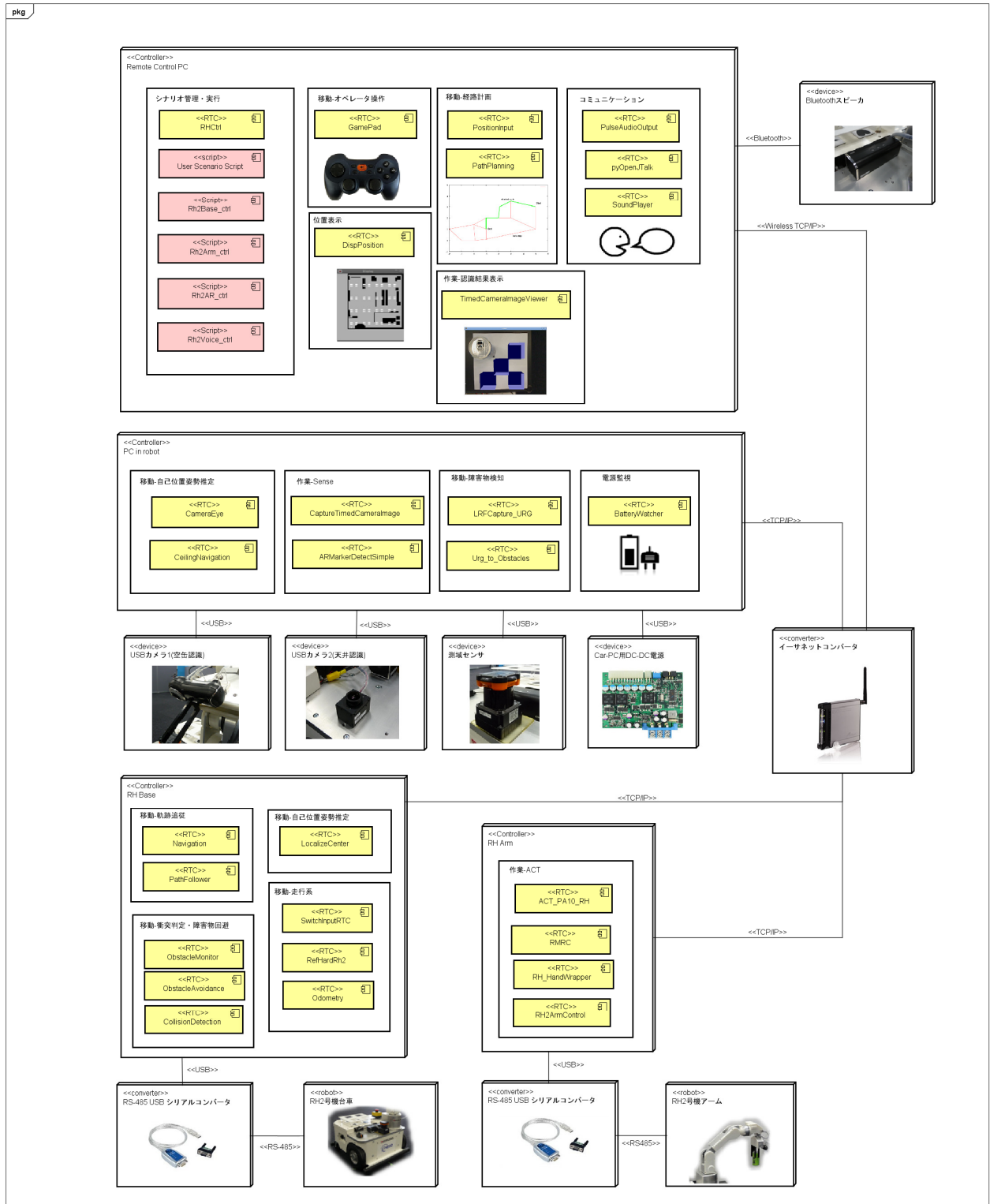
| No. | 文書名 | 備考 / URL |
|-----|-------------------------------|---|
| 1 | OpenRTM-aist Official Website | http://www.is.aist.go.jp/rt/OpenRTM-aist/ |

2. システム概要

2.1. ハードウェア構成

本システムのハードウェア構成について以下に記す。

図.ハードウェア構成図



3. 準備

3.1. 本章の構成

| | | |
|---|---|---|
| 1 | H/W 環境構築今版で用いた機材の制作方法を説明します。 | |
| | 1-1 | RH 制御 PC の内蔵方法 |
| 2 | S/W 環境構築動作に必須である S/W のインストールの手順を説明します。 | |
| | 2-1 | 動作環境 |
| | 2-2 | インストール手順 |
| | 2-3 | OpenRTM-aist-1.0.0-RELEASE |
| | 2-4 | OpenRTM-aist-Python-1.0.0-RELEASE |
| | 2-5 | RTSystemEditor・RTCBuilder-1.0.0-RELEASE |
| | 2-6 | rtctree・rtcshell |
| | 2-7 | OpenCV2.0 |
| | 2-8 | MOXA Uport 1450I ドライバ |
| | 2-9 | python visual |
| | 2-10 | OpenHRI |
| | 2-11 | その他(gtk+-2.0、libglade-2.0、gmodule-2.0) |
| 3 | ダウンロード・コンパイル RH 関連モジュールのダウンロード・コンパイル手順を説明します。 | |
| | 3-1 | ダウンロード |
| | 3-2 | コンパイル |
| 4 | 各種設定ご使用の環境に合わせた各種設定方法を説明します。 | |
| | 4-1 | 環境設定ファイル |
| | 4-2 | 作業に関する動作設定 |
| | 4-3 | 移動に関する動作設定 |
| | 4-4 | 音声発話に関する動作設定 |
| | 4-5 | 天井地図画像の作り方 |

3.2. 1.H/W 環境構築

3.2.1. RH 制御 PC の内蔵方法

RH 制御 PC の内蔵方法については下記マニュアルを参照ください。

- RH 制御 PC の内蔵方法

3.3. S/W 環境構築

3.3.1. 動作環境

1. RH 制御 PC

本 PC は RH に内蔵し、RH に搭載された各種センサーの制御を行う。

| | |
|------|--|
| メーカー | 自作 PC |
| 仕様 | CPU:AMD Phenom II X4 905e 2.5GHz Memory:PC2-6400 4GB SSD:OCZ OCZ SSD2-1SUM60G M/B: GeForce 8200 ITX WIFI GF8200-C-E 電源:Mini-BOX M4-ATX その他:ATX 電源スイッチ E-PC-S その他:技あり! 楽ラック! 2.5/ホワイト(CERS25-WT) |
| OS | Ubuntu 10.04LTS |
| SW | OpenRTM-aist-1.0.0-RELEASE OpenRTM-aist-Python-1.0.0-RELEASE OmniORB4 4.1.2-1 gcc 4.4.3 OpenCV-2.0.0 MOXA Uport 1450I ドライバ |

2. 遠隔操作 PC

本 PC は遠隔操作・監視用として使用する。

| | |
|------|--|
| メーカー | Lenovo |
| 製品名 | ThinkPad SL300 |
| 仕様 | CPU:Core™2 Duo プロセッサー P8400 2.26GHz Memory:PC2-5300 DDR2 SDRAM 2GBHDD:250GB (Serial ATA) |
| OS | Ubuntu 10.04LTS |
| SW | OpenRTM-aist-1.0.0-RELEASE OpenRTM-aist-Python-1.0.0-RELEASE OmniORB4 4.1.2-1 gcc 4.4.3 OpenCV-2.0.0 |

3.アーム制御 PC

本 PC は負荷分散用として RH のアーム制御に使用する。

| | |
|------|---|
| メーカー | VIA |
| 製品名 | ARTiGO-PC-15 |
| 仕様 | CPU:C7 1.5GHz Memory:DDR2 533 SODIMM 1GB HDD:80GB (Serial ATA) |
| OS | Ubuntu 10.04LTS |
| SW | OpenRTM-aist-1.0.0-RELEASE OpenRTM-aist-Python-1.0.0-RELEASE OmniORB4 4.1.2-1 gcc 4.4.3 libsvm(RH2 号機モータドライバ) |

4.台車制御 PC

本 PC は負荷分散用として RH の台車制御に使用する。

| | |
|------|---|
| メーカー | VIA |
| 製品名 | ARTiGO-PC-15 |
| 仕様 | CPU:C7 1.5GHz Memory:DDR2 533 SODIMM 1GB HDD:80GB (Serial ATA) |
| OS | Ubuntu 10.04LTS |
| SW | OpenRTM-aist-1.0.0-RELEASE OpenRTM-aist-Python-1.0.0-RELEASE OmniORB4 4.1.2-1 gcc 4.4.3 libsvm(RH2 号機モータドライバ) |

3.3.2. インストール手順

今回使用する PC 毎について、以下の手順通りインストールを実施する。

- RH 制御 PC
 - 1) OpenRTM-aist-1.0.0-RELEASE (C++) :2-3
 - 2) OpenRTM-aist-Python-1.0.0-RELEASE :2-4
 - 3) rtctree・rtshell :2-6
 - 4) OpenCV2.0 :2-7
 - 5) ARToolKit2.27.1 :2-11
 - 6) URG ライブラリ 0.8.12 :2-12
- 遠隔操作 PC
 - 1) OpenRTM-aist-1.0.0-RELEASE (C++) :2-3
 - 2) OpenRTM-aist-Python-1.0.0-RELEASE :2-4
 - 3) RTSystemEditor・RTCBuilder-1.0.0-RELEASE :2-5
 - 4) rtctree・rtshell :2-6
 - 5) OpenCV2.0 :2-7
 - 6) python visual :2-9
 - 7) OpenHRI :2-10
- アーム制御 PC
 - 1) OpenRTM-aist-1.0.0-RELEASE (C++) :2-3
 - 2) OpenRTM-aist-Python-1.0.0-RELEASE :2-4
 - 4) rtctree・rtshell :2-6
 - 5) MOXA Uport 1130 ドライバ :2-8
- 台車制御 PC
 - 1) OpenRTM-aist-1.0.0-RELEASE (C++) :2-3
 - 2) OpenRTM-aist-Python-1.0.0-RELEASE :2-4
 - 4) rtctree・rtshell :2-6
 - 5) OpenCV2.0 :2-7
 - 5) MOXA Uport 1130 ドライバ :2-8

3.3.3. OpenRTM-aist-1.0.0-RELEASE (C++)

C++言語で実装された RTC を実行するための RT ミドルウェアをインストールする。

対象となるハードウェア

- RH 制御 PC
- 遠隔操作 PC
- アーム制御 PC
- 台車制御 PC

対応する問題

- InPort::read()でブロックされる問題 [openrtm-users 01308]参照
※3～5 で問題に対応
- Manager の shutdown に関連したバグ [openrtm-users 01149]参照
※3～5 で問題に対応
- ipv6 エントリのコメントアウト [openrtm-users 01270]参照
※6 で問題に対応

本書では、まず一括インストールスクリプトを使用し OpenRTM-aist-1.0.0-RELEASE とその依存パッケージの一括インストールを行う。

その後バグ対応を行い、OpenRTM-aist-1.0.0-RELEASE を再インストールする。

・設定方法

1. 一括インストールスクリプト pkg_install100_ubuntu.sh を下記より適当なディレクトリ(ここでは~/)にダウンロードし、実行する。

※一括インストールスクリプトで「コンパイラや OmniORB4」をインストールする。

pkg_install100_ubuntu.sh ダウンロード

```
$su#sh pkg_install100_ubuntu.sh#exit
```

2. 下記よりソースファイルを(ここでは~/に)ダウンロードし展開する。

OpenRTM-aist-1.0.0-RELEASE.tar.gz ダウンロード

```
tar -xvzf OpenRTM-aist-1.0.0-RELEASE.tar.gz
```

3. 環境構築 1.で導入した「RH/patch」フォルダ内にあるパッチファイルを~/OpenRTM-aist-1.0.0/src/lib/rtm内にコピーする。

```
cp ~/RH/patch/patch_RingBuffer.h ~/OpenRTM-aist-1.0.0/src/lib/rtmcp ~/RH/patch/Manager.cpp.diff  
~/OpenRTM-aist-1.0.0/src/lib/rtm
```

4. パッチを当てる。

```
cd OpenRTM-aist-1.0.0/src/lib/rtmpatch < RingBuffer.h.diffpatch < Manager.cpp.diff
```

5. make する。

```
/OpenRTM-aist-1.0.0$./configure --prefix=/usr/OpenRTM-aist-1.0.0$make clean/OpenRTM-aist-  
1.0.0$make/OpenRTM-aist-1.0.0$sudo make install
```

6. RTSystemEditor からネームサーバは見えるのに、ネーミングコンテキストから先が見えない問題
/etc/hosts の localhost の ipv6 エントリをコメントアウトする。

```
# ::1      localhost ip6-localhost ip6-loopback
```

3.3.4. OpenRTM-aist-Python-1.0.0-RELEASE

Python 言語で実装された RTC を実行するための RT ミドルウェアをインストールする。

対象となるハードウェア

- RH 制御 PC
- 遠隔操作 PC
- アーム制御 RTC
- 台車制御 RTC

・設定方法

1. パッケージインストール

一括インストールスクリプト pkg_install_python_ubuntu.sh を下記から適当なディレクトリ(ここでは~/)にダウンロードし、実行する。

pkg_install_python_ubuntu.sh ダウンロード

```
$su#sh pkg_install_python_ubuntu.sh
```

途中、いくつかの質問をたずねられるので、y あるいは Y を入力しながら完了させる。

3.3.5. RTSystemEditor・RTCBuilder-1.0.0-RELEASE

RTC を接続したり、状態を監視するためのツール RTSystemEditor をインストールする。

RTSystemEditor は Eclipse のプラグインとして提供されるものであるため Eclipse もインストールする必要がある。

以下の URL から RTSystemEditor をプラグインした Eclipse をダウンロードしインストールする。

対象となるハードウェア

- 遠隔操作 PC

URL: <http://www.openrtm.org/openrtm/ja/node/941>

3.3.6. rtctree・rtshell

シェルで RTC の管理をするためのスクリプト rtctree 及び rtshell をインストールする。

以下の URL にてマニュアルを参考の上、インストールする。

対象となるハードウェア

- RH 制御 PC
- 遠隔操作 PC

URL: <http://www.openrtm.org/openrtm/ja/node/869>

<http://www.openrtm.org/openrtm/ja/node/1323>

3.3.7. OpenCV2.0

カメラ画像の取得や天井画像による自己位置推定を行うため、画像処理、解析を行う。

対象となるハードウェア

- RH 制御 PC
- 遠隔操作 PC

・設定方法

1. パッケージインストール

```
$sudo aptitude install libcv4 libhighgui4 libcvaux4 libcv-dev libhighgui-dev libcvaux-devpython-opencv opencv-doc
```

URL: <http://opencv.jp/opencv2-x-tips/install-linuxubuntu>

3.3.8. MOXA Uport 1130 ドライバ

RH の台車とアームを接続に使用するシリアルコンバータのドライバをインストールする。

対象となるハードウェア

- アーム制御 PC
- 台車制御 PC

1. Moxa 社のサポートページよりドライバをダウンロード

URL: http://www.moxa.com/support/support_home.aspx

※確認済みバージョンは 1.2.3 であるが、現時点(2011/7/7)で上記 URL から入手可能な最新版は 1.2.0 である。コンパイルが通らない等の問題がある場合は MOXA 社へ問い合わせの上、入手を行うこと。

2. ドライバをインストール

任意の場所にドライバをダウンロードし解凍する。

インストール方法については、付属の readme.tx を参照のこと。

3.3.9. python visual

RH アームのシミュレータ環境として、python visual をインストールする。対象となるハードウェア

- 遠隔操作 PC

1. Synaptic マネージャを起動する。
「システム」→「システム管理」→「Synaptic パッケージ・マネージャ」と選択する。
2. python visual をインストールする。
「python visual」で検索し、チェックを入れ、適用する。
3. 「libgtkglext」で検索し、libgtkglext1 と libgtkglext1-dev にチェックを入れ、適用する。

3.3.10. OpenHRI

音声発話の為にライブラリ及び RTC をインストールする。

対象となるハードウェア

- 遠隔操作 PC

※ Bluetooth スピーカを使用の為に、遠隔操作 PC にしているが、USB スピーカを使用の場合は RH 制御 PC へインストールする。

インストール手順は OpenHRI ホームページ(<http://openhri.net/>)を参照のこと。

3.3.11. ARToolKit2.27.1

AR マーカを認識するために、ARToolkit2.27.1 をインストールする。

対象となるハードウェア

- RH 制御 PC

本パッケージは RH のソースの以下のディレクトリに格納している。

PCinRH/src/lib/ARToolkit インストール及び設定方法については「ARToolKit マーカ検出・位置姿勢推定コンポーネントの使い方」を参照のこと。

3.3.12. URG ライブラリ 0.8.2

北陽電機製 LRF UTM-30LX を制御するため URG ライブラリ 0.8.2 をインストールする。

対象となるハードウェア

- RH 制御 PC

本パッケージは RH のソースの以下のディレクトリに格納している。

PCinRH/src/comp/lrf_urg_module/lib/urg-0.8.12

インストール方法については以下の URL を参照のこと。

URL: http://www.hokuyo-aut.co.jp/02sensor/07scanner/download/urg_programs/devel_lin_page.html

3.4. ダウンロード・コンパイル

3.4.1. ダウンロード

RH ソース一式をダウンロードする。

ダウンロードした RH.zip を展開し、遠隔制御 PC は RemotePC フォルダ下、RH 内蔵 PC は PCinRH 下、アーム制御 PC は RHArm、台車制御 PC は RHBase を各々コピーする。
尚、本書ではインストールフォルダを各 PC の「/opt/RH」とする。

3.4.2. コンパイル

- 遠隔操作 PC

```
$cd /opt/rh/RS003/src/comp  
$mclean; mk; minst
```

次に、python で RTC を制御するために IDL コンパイルを行う。

※インストール済の ExtendedDataTypes.idl を書き換えるため、必ずバックアップ後に実行すること。なお RtcHandle を使用しない場合、本手順は不要である。

```
$cd /usr/lib/python2.6/dist-packages/OpenRTM_aist/RTM_IDL/  
$cp -p ExtendedDataTypes.idl ExtendedDataTypes.idl.org  
$cd /opt/rh/RS003/idl2  
$sudo cp -p /usr/lib/python2.6/dist-packages/OpenRTM_aist/RTM_IDL/  
$sudo cp -p intellirobot.idl /usr/lib/python2.6/dist-packages/OpenRTM_aist/RTM_IDL  
$sudo cp -p ExtendedDataTypes.idl /usr/lib/python2.6/dist-  
packages/OpenRTM_aist/RTM_IDL  
$cd /usr/lib/python2.6/dist-packages/OpenRTM_aist/RTM_IDL  
$sudo omnidl -bpython *.idl
```

- RH 制御
PC

```
$cd /opt/rh/RS003/src/comp  
$mclean; mk; minst
```

- アーム制御 PC

```
$cd /opt/rh/RS003/src/comp  
$mclean; mk; minst
```

- 台車制御 RTC

```
$cd /opt/rh/RS003/src/comp  
$mclean; mk; minst
```

3.5. 各種設定

3.5.1. 環境設定ファイル

本システムは python スクリプトにて RH の各機能の制御を行っているが、python スクリプトを使うために予め PATH の設定が必要である。

- 設定ファイルのパス設定

環境変数 PYTHONPATH へ設定ファイル set_env.py の格納パスを環境に合わせて設定する。

```
[bash]
> vi ~/.bashrc
export PYTHONPATH=/opt/RH/etc/env
> source ~/.bashrc

[csh]
> vi ~/.cshrc
setenv PYTHONPATH /opt/RH/etc/env
> source ~/.cshrc
```

- 設定項目の説明

遠隔制御 PC の set_env.py の内容を環境に合わせて書き換えること。

| 設定項目 | デフォルト値 | 説明 |
|------------------|--|--------------------------------|
| Rh2IdlDir | /opt/rh/RS003/idl | IDL ファイルの格納場所 |
| Rh2BinDir | /opt/rh/RS003/bin | 実行ファイル(バイナリ)の格納場所 |
| Rh2ConfDir | /opt/rh/RS003/etc/conf | 設定ファイル(rtc.conf)の格納場所 |
| Rh2LibDir | /opt/rh/RS003/lib | ライブラリ(バイナリ)の格納場所 |
| RtmToolsDir | Rh2BinDir + "/script/control/rtmttools" | RtcHandle 等の格納場所 |
| Rh2BaseToolsDir | Rh2BinDir + "/script/control/Rh2Base" | 移動系制御スクリプトの格納場所 |
| Rh2ARToolsDir | Rh2BinDir + "/script/control/Rh2AR" | 作業系(AR マーカ認識)制御スクリプトの格納場所 |
| Rh2ArmToolsDir | Rh2BinDir + "/script/control/Rh2Arm" | 作業系(アーム)制御スクリプトの格納場所 |
| Rh2VoiceToolsDir | Rh2BinDir + "/script/control/Rh2Voice" | 音声発話制御スクリプトの格納場所 |
| Rh2RtcDir | Rh2BinDir + "/comp" | RTC(バイナリ)の格納場所 |
| CtrlTermNS | 192.168.30.125:9876 | 制御端末のネームサーバの IP、ポート番号 |
| Rh2BaseNS1 | [192.168.30.138:2809 192.168.30.198:2809] | 移動系(台車制御 PC)のネームサーバの IP、ポート番号 |
| Rh2BaseNS2 | [192.168.30.138:2809 192.168.30.106:2809] | 移動系(RH 内蔵 PC)のネームサーバの IP、ポート番号 |
| Rh2BaseNS3 | localhost:2809 | 移動系(遠隔操作 PC)のネームサーバの IP、ポート番号 |

| 設定項目 | デフォルト値 | 説明 |
|------------|--------------------------------------|-----------------------------------|
| Rh2ArmNS | 192.168.30.191:9876 | 作業系(アーム)のネームサーバの IP、ポート番号 |
| Rh2ARNS | 192.168.30.106:2809 | 作業系 (AR マーカ認識) 制御スクリプトの格納場所 |
| Rh2VoiceNS | localhost:5005 | コミュニケーション系のネームサーバの IP、ポート番号 |
| Rh2CtrlNS1 | 192.168.30.106:2809 | RH 管理系(RH 内蔵 PC)のネームサーバの IP、ポート番号 |
| Rh2CtrlNS2 | [192.168.30.138:2809 localhost:2809] | RH 管理系(遠隔操作 PC)のネームサーバの IP、ポート番号 |

3.5.2. 作業に関する動作設定

下記マニュアルを参照のこと。

- 分解運動速度制御モジュール(ACT 共通 I/F 対応版)操作手順書

3.5.3. 移動に関する動作設定

3.5.3.1. 目的地の座標を変更と追加

- ・目的地を変更又は追加する場合は以下のフォルダに格納されている部分を修正する。

遠隔操作 PC:「/opt/rh/RS003/etc/conf/Position.conf」

RH 内蔵 PC:「/opt/rh/RS003/etc/conf/LocalPosition.conf」

- ・座標について

| 名前 | X 座標(m) | Y 座標(m) | 姿勢(deg) |
|---------|---------|---------|---------|
| 待機場所 | 3.4 | 7.7 | -90 |
| 飲み物搭載場所 | 1.44 | 10.3 | -90 |
| ゴミ箱 | 4.24 | 1.1 | 180 |
| 会議室 | 6.08 | 1.06 | 0 |

- ・経路マップの生成

経路マップは経路マップ生成ツールにて生成可能です。

使用方法は「オープンソース移動知能モジュール群経路計画・起動追従モジュール 機能仕様書」を参照のこと。

3.5.4. 音声発話に関する動作設定

メッセージの変更・追加は遠隔操作 PC : /opt/rh/RS003/bin/script/control/Rh2Voice/message.py をメンテナンスして下さい。

```
# message
# RHBase
mstr001 = "移動開始します。"
mstr002 = "前進します。"
mstr003 = "前進します。退避願います。"
mstr004 = "バック、します。"
mstr005 = "停止します。"
mstr006 = "旋回します。"
mstr007 = "到着しました。"
mstr008 = "一度前進して、くるっと旋回して、戻りますので、退避願います。"

# RHArm
mstr010 = "あーむが動きます。ご注意ください。"
mstr011 = "直立姿勢に戻ります。"
mstr012 = "走行姿勢に戻ります。"

# DrinkTransport
mstr020 = "お待たせ致しました。お飲み物を、お持ちしました。受取台に置いておきます。"
mstr021 = "ありがとうございます。またのご利用を、お待ち、しております。失礼致します。"
mstr022 = "どうぞお飲みになって下さい。"

# GarbageDump
mstr030 = "空き缶を、回収に参りました。"
mstr031 = "回収が終わりました。失礼致します。"
mstr033 = "廃棄を開始します。"
mstr034 = "廃棄が終了しました。"

# Battery
mstr040 = "バッテリー残量がありません。緊急停止を、行います。"
mstr041 = "バッテリー残量がありません。充電を、してください。"
mstr042 = "運動機能を停止します。"
mstr043 = "運動機能を再開します。"

# Boot
mstr050 = "起動が完了しました。"
mstr051 = "終了します。本日もお疲れさまでした。"
mstr052 = "起動を開始します。"

# ObstacleMonitor(add RS003)
mstr060 = "どいてください。"
mstr061 = "回避します。"
mstr062 = "いい加減どいてください。"
```


3.5.5. 天井地図画像の作り方

天井画像地図の作り方については下記マニュアルを参照ください.

- 天井画像地図の作り方マニュアル

4. 使用方法

4.1. 本章の構成

| | | |
|---|------------------------------------|------------------|
| 1 | 事前動作確認各機能単体での動作確認方法を説明します。 | |
| | 1-1 | RH の実機動作確認 |
| | 1-2 | 天井画像による推定自己位置の確認 |
| | RH 操作方法本システムにおける RH の各種操作方法を説明します。 | |
| 2 | 2-1 | RH 起動 |
| | 2-2 | RH 操作 |
| | 2-3 | RH 休止・再開 |
| | 2-4 | RH 終了 |

4.2. 1.事前動作確認

4.2.1. RH の実機動作確認

RH とアーム制御 PC 及び台車制御 PC が正しく接続できていることを確認するため、RH2 号機に付属するサンプルプログラムを用いることとし、手順を以下に記す。

以下に当てはまる場合は本確認を必ず行ってからシステム動作を行うこと。

- MOXA Uport 1130 ドライバをインストール後、初めて接続する場合。
- RH 本体を修理後、初めて接続する場合。

4.2.1.1. デバイスの認識状況確認

- ・MOXA Uport 1130 が認識されているかを確認

```
$ls /dev 「tab キー」
```

一覧が表示され、アーム制御 PC 及び台車制御 PC で各々「ttyUSB0」が表示されていることを確認。

- ・ゲームパッドが認識されているかを確認

```
$ls /dev/input/ 「tab キー」
```

一覧が表示され「js0」が表示されていることを確認

4.2.1.2. 環境設定

ご使用の環境に合わせて、一部ソースを修正する必要がある

- ・calib_armV2 (アームのキャリブレーション)

```
$vi calib_armV2.c11 #define RS485_DEVNAME "/dev/ttyUSB0"
```

moxa の接続番号を変更する場合は 11 行目の「ttyUSB0」をご使用の環境に合わせ、変更する。その後 make を行う。

- ・joydriveV2 (アームの操作)

MOXA Uport 1130 の差込位置やゲームパッドの認識番号が変更された時に、以下の箇所を修正する

```
$vi joydriveV2.c18 #define JOYDEVNAME "/dev/input/js0"21 #define RS485_DEVNAME  
"/dev/ttyUSB0"
```

ゲームパッドの認識番号を変更する場合は 18 行目の「js0」を変更する

moxa の接続番号を変更する場合は 21 行目の「ttyUSB0」を変更する

再度 make をおこなう

・台車の操作の場合

moxa の接続とゲームパッドの認識番号を変更した時に、以下の箇所を修正する

```
$vi joybaseV2.c18 #define JOYDEVNAME "/dev/input/js0"21 #define RS485_DEVNAME  
"/dev/ttyUSB0"
```

ゲームパッドの認識番号を変更する場合は 18 行目の「js0」を修正する

moxa の接続番号を変更する場合は 21 行目の「ttyUSB0」を修正する。その後 make を行う。

4.2.1.3. 動作確認

・アームのキャリブレーション

```
$cd joydrive091215$./calib_armV2
```

<注意>

- キャリブレーション時、アームが動作するため、RH の周囲から離れること。
- ごく稀に J5 軸 (屈曲) (「RH 動作仕様書」を参照のこと) が動いた後、キャリブレーションがストップする現象が発生する。この場合は、「非常停止ボタン」押下後、アーム制御 PC を再起動すること。
- 起動後、4 ボタン (速度制御モード) を押さないと、ゲームパッドの操作はできません。

・ゲームパッド

1. アームの操作

```
$cd joydrive091215$./joydriveV2
```

起動後、ゲームパッドを動かし、データが流れることを確認する。

4 ボタンを押下し、「-- svm_setmode(): SVM_VELOC (0x01)」と表示されると、ゲームパッドの指示で動作可能となる。

2. 台車の操作

```
$cd joydrive091215$./joybaseV2
```

起動後、ゲームパッドを動かし、データが流れることを確認する。

4 ボタンを押下し、「-- svm_setmode(): SVM_VELOC (0x01)」と表示されると、ゲームパッドの指示で動作可能となる。

4.2.2. 天井画像による推定自己位置の確認

天井画像マッチングが正常に動作しているかどうかの確認、およびトラブルシューティングについては下記マニュアルを参照ください。

- 天井画像マッチングによる推定自己位置の確認法

4.3. 2.RH 操作方法

4.3.1. RH 起動

4.3.1.1. RH 制御 PC の起動

- ・非常停止ボタンを ON
非常停止がボタンが OFF であれば ON にします。
OFF のまま、電源を入れたときにアームの体勢が維持できなくなり、倒れてしまうので注意します。
 - ・アームが直立姿勢(全ての軸が 0 度)であれば問題ないです。
- 図.安全停止ボタン ON



- ・メイン電源を ON
メイン電源を ON にすると電圧計が、現在の電圧を指します。
電圧が 21V 以下であれば充電します。
- 図.メイン電源 ON



- ・電源を入れる前の準備
キーボードとモニタを内蔵 PC に接続する
参考 URL : <https://forums.ubuntu-linux.jp/viewtopic.php?id=4222>
 - ・RH 制御 PC の電源を ON
ATX 電源スイッチを使用しております。
ボタンを押すと、右側の緑ランプが点灯し、URG の起動も始まります。
「ピピ」と音が鳴ると PC が立ち上がった目印です。
- 図.RH 制御 PC 電源



4.3.1.2. アーム制御 PC、台車制御 PC の起動

- ・RH 荷台上の PC2 つの電源(写真赤枠)を ON



4.3.1.3. 遠隔操作 PC の起動

- ・遠隔操作 PC を起動させます。

4.3.1.4. 遠隔操作のモジュール起動

4.3.1.4.1. ネームサーバの削除と起動

端末を開きます。

ネームサーバを 2 つ起動させます。

```
$rtm-naming 2809$rtm-naming 5005
```

<注意> ネームサーバを安定稼働させるために、Linux のブートプロセスとして既に起動しているネームサーバは一旦 kill を行った方が無難です。

```
$sudo killall omniNames
```

4.3.1.4.2. 遠隔操作 PC の動作準備

Bluetooth スピーカとの接続を確認

サウンドが以下の図の様に設定してあることと音が RH 搭載のスピーカから鳴っていることを確認
図.サウンドの設定



4.3.1.4.3. 遠隔操作 PC のモジュール起動

端末を開きます。
start.sh を起動します。

```
$ sh start.sh
```

次に start2.sh を起動
RH から「音声起動しました。」とアナウンスされます。

```
$ sh start2.sh
```

最後に start3.sh を起動

```
$ sh start3.sh
```

4.3.1.4.4. eclipse 起動

端末を開きます。
eclipse を起動させます。

```
$/eclipse -clean ワークスペース名 /home/rtc/workspace
```

eclipse の起動が終了したら以下の設定をおこないます。

ウィンドウ >> パースペクティブを開く >> その他をクリックパースペクティブのメニューから「RT System Editor」を選択

4.3.1.5. RH 制御 PC のモジュール起動

RH のアームと走行系のコンポーネントを起動させます。
指示方法は RH 遠隔操作 PC から SSH を使用します。

4.3.1.5.1. ネームサーバの削除と起動

ネームサーバを起動させます。

```
$rtm-naming 2809
```

<注意>

ネームサーバを安定稼働させるために、Linux のブートプロセスとして既に起動しているネームサーバは一旦 kill を行った方が無難です。

```
$sudo killall omniNames
```

4.3.1.5.2. 天井カメラ(NM33-N-UVC)設定

専用ツールを使用して、カメラの設定を行う。

なお、電源断により設定値が消去されるため、毎回起動時に必ず実施すること。

※本ツール(lucvview)は標準で付属していないため、カメラの購入元へお問い合わせの上、入手して下さい(二次配布禁止のため、当方では配布致しません)。

```
$cd /opt/rh/RS003/bin/tool$sudo lucvview -d /dev/video1
```

設定方法については「オープンソース移動知能モジュール自己位置姿勢推定モジュール機能仕様書」を参照のこと。

4.3.1.5.3. 天井ナビゲーションモジュール起動

```
$ssh start.sh
```

4.3.1.5.4. AR マーカ認識モジュール起動

```
$ssh ar_start.sh
```

4.3.1.6. アーム制御 PC のモジュール起動

4.3.1.6.1. ネームサーバの削除と起動

ネームサーバを起動させます。

```
$rtm-naming 9876
```

<注意>

ネームサーバを安定稼働させるために、Linux のブートプロセスとして既に起動しているネームサーバは一旦 kill を行った方が無難です。

```
$sudo killall omniNames
```

4.3.1.6.2. アーム制御モジュール起動

RH の非常停止ボタンを解除し、start.sh を起動する。

```
$ cd /opt/rh/RS003/bin/script/control  
$ ./start.sh
```

- start.sh を実行することで、アームのキャリブレーションの実行及び必要なコンポーネントの起動を行う。

4.3.1.7. 台車制御 PC のモジュール起動

4.3.1.7.1. ネームサーバの削除と起動

ネームサーバを起動させます。

```
$rtm-naming 2809
```

<注意>

ネームサーバを安定稼働させるために、Linux のブートプロセスとして既に起動しているネームサーバは一旦 kill を行った方が無難です。

```
$sudo killall omniNames
```

4.3.1.7.2. アーム制御モジュール起動

start.sh を起動する。

```
$ cd /opt/rh/RS003/bin/script/control  
$ ./start.sh
```

- start.sh を実行することで、アームのキャリブレーションと走行姿勢に移動を行い必要なコンポーネントの起動を行う。

4.3.1.8. モジュール接続・活性化(遠隔操作 PC)

アーム関連の RTC の接続を行う。

```
$ cd /opt/rh/RS003/bin/script/control/Rh2Arm  
$ python Rh2Arm_run.py
```

次に、アームを走行姿勢へ移行する。

```
$ cd /opt/rh/RS003/bin/script/control  
$ python RHArmReady.py
```

「アームが動きます。ご注意ください。」と発話した後、アームが動き、走行姿勢へと移行する。
走行姿勢は以下の通り。

図.アーム走行姿勢



次に、アームを走行姿勢へ移行する。

```
$ cd /opt/rh/RS003/bin/script/control  
$ python RHArmReady.py
```

次に台車関連の RTC の接続・活性化を行う。

```
$ cd /opt/rh/RS003/bin/script/control/Rh2Base  
$ python Rh2Base_run.py
```

4.3.1.8.1. RT System Editor 確認

起動させたコンポーネントが正常に起動したか確認。

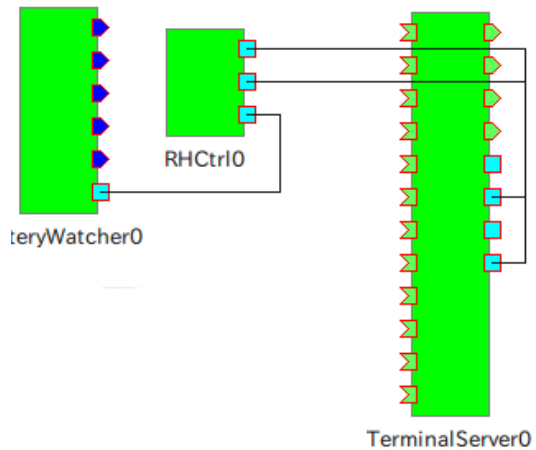
ネームサービスビューを左クリックし、「Add Name Server」を選択し、
環境設定ファイルの「set_env.py」のに設定したネームサーバをそれぞれ開くと、
登録内容がツリー形式で表示されます。

4.3.1.8.2. 制御端末との接続

環境設定ファイルの「set_env.py」の「CtrlTermNS」が、制御端末(TerminalServer)のネームサーバになっており以下の図の通りに接続をおこなう。

※接続後 RHCtrl を Activate にする

図. ターミナルとの接続



4.3.2. 2-2.RH 操作

4.3.2.1. 各端末からの操作

操作について下記のマニュアルを参照

- 端末機能 取扱説明書

4.3.2.2. 各種スクリプトによる操作

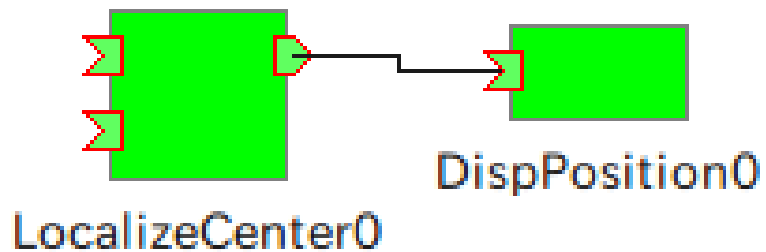
遠隔操作 PC より python スクリプトを使用して各種動作を行います。
各スクリプトの詳細については下記のマニュアルを参照。

- RH 詳細設計書

4.3.2.3. ロボットの推定自己位置確認

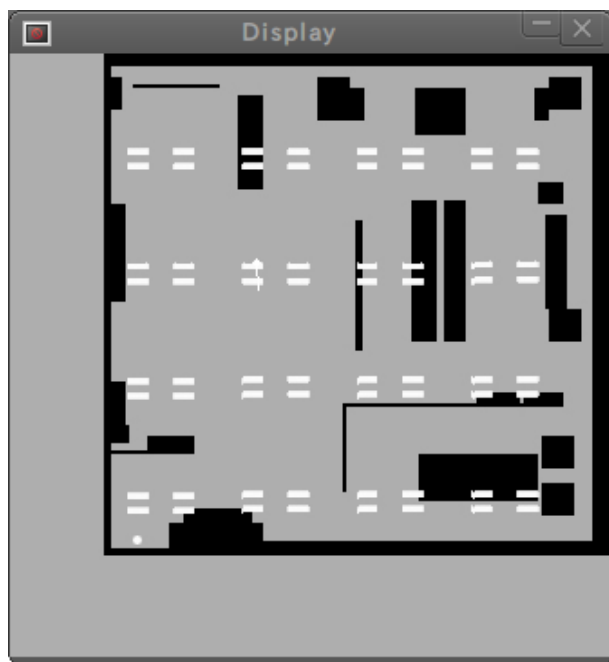
以下の図の様に接続を行い Activate にする

図. LocalizeCenter への接続



Activate 後、以下の図が表示される

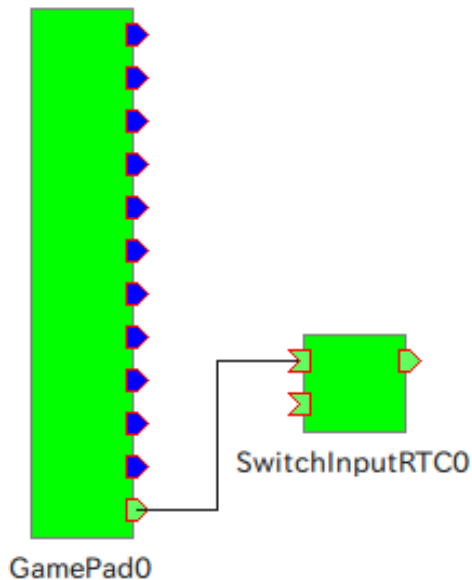
図. 現在座標の表示



4.3.2.4. ゲームパッド操作

ゲームパッドで操作するときにゲームパッドのコンポーネントを接続する必要がある
接続先の内容は以下の図の様に接続を行い Activate にする

図. GamePad 接続

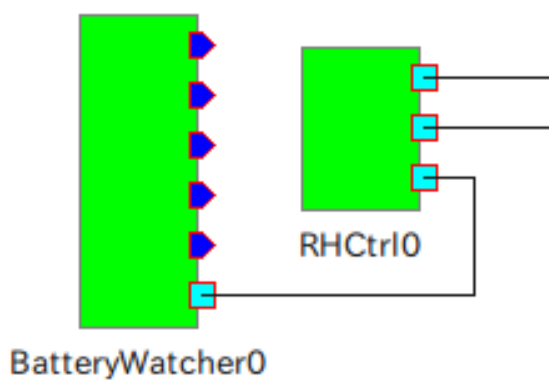


ゲームパッドの 6 番を押すと停止中、走行中に関わらずリモコンモードへの切り替えが可能。

4.3.2.5. バッテリー残量のチェック

BatteryWatcher のコンポーネントを起動し RHCtrl コンポーネントに以下の図の様に接続を行い Activate にする

図. バッテリーチェック接続



バッテリーの残量が 21V になると以下の警告アナウンスをおこなう
「バッテリー残量がありません。充電をしてください。」

4.3.3. 2-3.RH 休止・再開

RH 休止及び再開させる場合は以下の指示をおこないます。

休止

```
$ cd /opt/bin/script/control($ python RHBaseStop.py)$ python RHStop.py
```

<注意>

- 走行中の場合は、走行中の経路のリセットを行うため、RHBaseStop.py を実行して下さい。
実行しない場合、再開後、不正に走り出す危険があります。
- 動作中の中断・作業の再開については考慮していません。全て終了させて、最初からやり直して下さい。

再開

```
$ cd /opt/bin/script/control$ python RHRestart.py
```

- 再開後は「天井画像による推定自己位置の確認」を必ず行ってください。

4.3.4. 2-4.RH 終了

4.3.4.1. コンポーネントの終了

start.sh でコンポーネント起動後、コンポーネントを終了させる場合は以下の指示をおこないます。

```
$ cd /opt/bin/script/control$ python RHArmStandUp.py$ python RHStop.py$ sh kl.sh
```

- ・アームを直立姿勢に移動し、台車のコンポーネントを Deactivate する。
※アームを直立姿勢に移動後、非常停止ボタンを ON にしておくこと。

4.3.4.2. 再起動又はシャットダウン

RH 制御 PC が不調であれば再起動かシャットダウンをおこないます。

再起動

```
$sudo reboot
```

シャットダウン

```
$sudo shutdown -h now
```

5. メンテナンス

5.1. 本章の構成

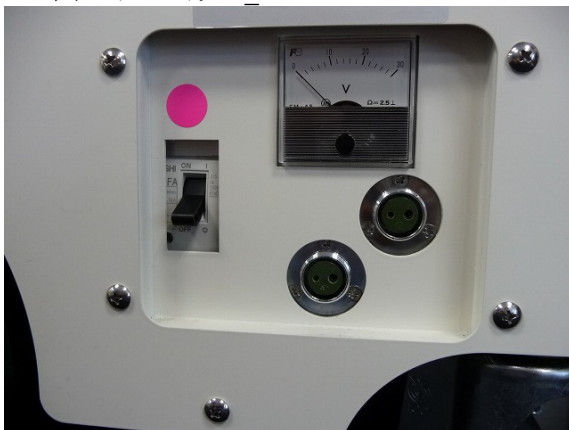
| | |
|---|---|
| 1 | 充電方法各機能単体での動作確認方法を説明します。 |
| 2 | SSD の交換 RH コントロール PC の SSD の交換方法を説明します。 |

5.2. 1.充電方法

RH の充電方法を以下に明記します。
充電の目安は 21V 以下になったら充電します。

5.2.1. 充電の手順

- 1) 非常停止ボタンを ON
アームが倒れない様にするため充電する時には必ず ON にします。
- 2) メインスイッチを OFF
メインスイッチ OFF にすると電圧計は「0」を指している事を確認
メインスイッチが ON になっていると充電系統が動作しない。
図.メインスイッチ_OFF



- 3) 充電器の用意
図.充電器



充電器を2つ用意します。

コネクタは RH 専用のコネクタとなっています。
コネクタは大小の筒状が2つあり、極性を間違えて接続できないようになっています。
図.コネクタ



- 4) コネクタ接続
メインスイッチの右側に充電用のコネクタ差込口が2つあります。
そこに、コネクタを差し込みます。

図.コネクタを差し込む



- 5) 充電開始
充電器のスイッチを入れて充電します。
充電中は赤色のランプが点灯します。

図.充電中

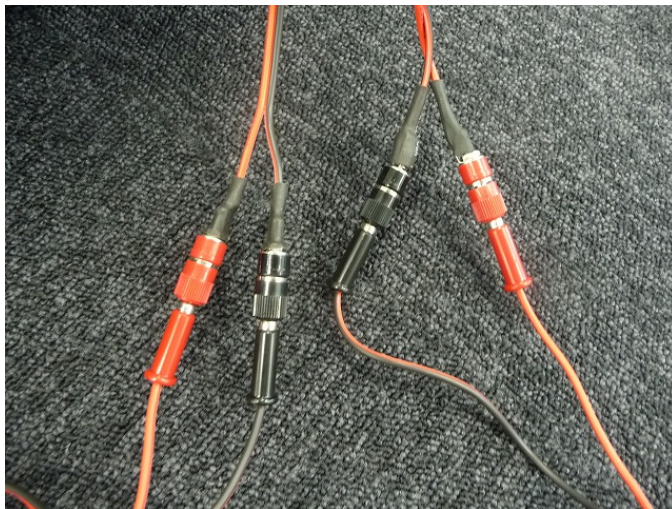


- 6) 充電完了
充電完了は緑色のランプが点灯します。
図.充電完了



5.2.2. 注意事項

- 1) 充電器のケーブルについて
ショートしないよう充電器のケーブルは離しておきます。
図.ケーブル

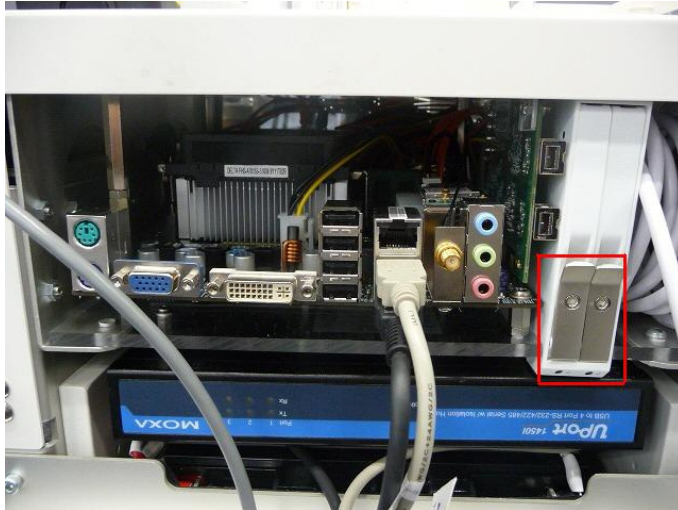


5.3. 2.SSD/HDD の交換

RH コントロール PC は 2.5 インチ SATA SDD または HDD が容易に交換可能です。
その手順を以下に記します。

5.3.1. 交換の手順

- 1) 赤枠の銀色の金具を引きます。
図.SSD/HDD の設置場所



- 2) SSD/HDD を引き出して、入れ替えを行います。
最大 2 台収納可能です。複数台使用の場合はブートディスクを左に入れてください。
図.SSD/HDD 取り出し



6.トラブルシューティング

6.1. 動作指令を出しても RH が動き出さない

【現象】

会議端末から「給仕依頼」や「廃棄依頼」、またスクリプトから直接指示を与えても、RH が動き出さない。

【原因】

以下の原因が考えられます。

- 1.一部の RTC でエラーが発生している／落ちている。
- 2.自律走行モードではなくリモコンモードになっている。
- 3.RH 本体が故障している。

【対応方法】

- 1.現在の RTC の状況を「RT System Editor」でご確認下さい。
RTC が不正に終了している場合は一旦全て終了後、起動からやり直して下さい。
また RTC の設定を変更した場合は設定内容の不正によりエラーが発生することがございます。
「RH 詳細設計書」を参照の上、今一度設定の見直しをお願いします。
- 2.ゲームパッドの 6 番ボタンを押したままになっている可能性があります。
リモコンモードの 6 番ボタンをもう一度押して見て下さい。
- 3.起動からやり直しても、同様の現象が起きる場合は、RH 本体が故障している場合があります。
「RH の実機動作確認」にて動作を確認の上、挙動がおかしい場合は以下にお問い合わせ下さい。

<連絡先>

株式会社前川製作所
技術研究所 基盤技術開発グループ
山下 智輝
email:tomo-3440@mayekawa.co.jp

6.2. RH の挙動や言動がおかしい

【現象】

RH が小刻みに停止したり、「直進します。直進します。直進します。」等と同じ言葉を連呼する。

【原因】

スクリプトによる起動、休止、再開等を行った際、順番を誤ると RTC 間のコネクタが多重に引かれることがあります。

【対応方法】

現在のコネクタの状況を「RT System Editor」でご確認下さい。
コネクタが多重に引かれている場合、一旦全て終了後、起動からやり直して下さい。

6.3. 推定位置が真値から大きく外れてしまったときの対処法

【対応方法】

- ・下記マニュアルを参照ください。
 - 推定位置が真値から大きく外れてしまったときの対処法

7. その他

7.1. 延期要求

特になし。

7.2. その他の要件

特になし。

7.3. 特記事項

本書をご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- 本書は独立行政法人 新エネルギー・産業技術総合開発機構の「次世代ロボット知能化技術開発プロジェクト」内実施者向けに評価を目的として提供するものであり、商用利用など他の目的で使用することを禁じます。
- 本書に情報を掲載する際には万全を期していますが、それらの情報の正確性またはお客様にとっての有用性等については一切保証いたしません。
- 利用者が本書を利用することにより生じたいかなる損害についても一切責任を負いません。
- 本書の変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本書の情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

【連絡先】

RTC 再利用技術研究センター

〒101-0021 東京都千代田区外神田 1-18-13 秋葉原ダイビル 1303 号室

Tel/Fax:03-3256-6353 E-Mail:contact@rtc-center.jp