

【0.はじめに】

本 LiveCD は、「独立行政法人新エネルギー・産業技術統合開発機構 (NEDO)」の「次世代ロボット知能化技術開発プロジェクト」における研究開発の成果物である「ロボット知能ソフトウェアプラットフォーム (OpenRTP)」を、推奨動作環境である Linux 系 OS の「Ubuntu」にプレインストールした、カスタム版の Ubuntu ライブ CD です。

これにより、OpenRTP をご自身の PC にインストールすることなしに、Ubuntu 環境上での OpenRTP をお試し頂くことが可能です。また、本 CD から直接インストールすることも可能です。

===== 【1.LiveCD(iso イメージファイル)の利用について】

LiveCD(iso イメージファイル)をどのように使うかが不明なビギナー向けの利用手引きです。既に把握されている方は読み飛ばしてください。

(1)お手元にライブ CD が無い場合

iso イメージファイルをダウンロードし、ご自身で DVD-R(CD-R にはサイズが大のため入らない)などに焼く必要があります。

VMware などの仮想マシンをご利用の場合は、そのまま取得した iso イメージファイルからインストールが可能です。(通常の Ubuntu インストール手順)

iso イメージを CD に焼くには、専用のソフトが必要です。

(参照「<https://wiki.ubuntulinux.jp/UbuntuTips/Install/BurningISO>」)

(2)お手元にライブ CD がある場合

まず、PC 起動をして OS 起動前に BIOS 設定画面に入ります。DVD ドライブから先に起動出来るように、BootSequence を変更します。

そのまま PC を起動させ、ライブ CD をドライブに入れたままでシャットダウンします。再度 PC を起動させると Ubuntu のメニュー画面が出てくるはずですが。

(2-1)PC にインストールせずに利用する場合

「コンピュータに変更を加えないで試してみる」を選択します。

ログイン無しのまま、Ubuntu のデスクトップ画面(Gnome)が表示されます。

この状態で以下の 3 章のサンプルなどを動かすことが既に可能となります。

(2-2)PC にインストールする場合

「Ubuntu をインストール」を選択し、以下を参考にインストールを行ってください。

OpenRTP も同封された状態でインストールされます。

(参照「<https://wiki.ubuntulinux.jp/UbuntuTips/Install/InstallUbuntu>」)

【2.構成について】

動作環境と中身を構成するパッケージは以下を使用しています。

(1)OS:Ubuntu10.04

Linux系OSであり、OpenRTPを使用する上での推奨環境。

(2)OpenRTP:

以下のパッケージ群から構成されています。

[OpenRTM-aist-C++]: ver1.0.0

「pkg_install100_ubuntu.sh」を使用。

対話形式でパッケージ版をデフォルトの「/usr」以下にインストール済み。

[OpenRTM-aist-Java]: ver1.0.0

「OpenRTM-aist-Java-1.0.0-jar.zip」を使用。

「/usr/lib/OpenRTM-aist/1.0」にjar版を展開済み。

[OpenRTM-aist-Python]: ver1.0.0

aptitudeでパッケージ取得。

デフォルトで「/usr/lib/python2.6/dist-packages/OpenRTM_aist」以下にインストール済み。

[RtSystemEditor]: ver1.0.0

eclipse内にplugin化済み。

[RtcBuilder]: ver1.0.0

eclipse内にplugin化済み。

[OpenHRP3]: ver3.1.0.beta4

「OpenHRP-3.1.0.beta4.zip」を使用。

「/usr」以下にパッケージ版をインストール済み。

[eclipse]: ver3.4.2

「eclipse342_hrpdependencies_linux_ja_20100315.tar.gz」(OpenHRP3 Webページより)を使用。

OpenHRP3の必要パッケージとRtSystemEditor/RtcBuilderもインストール済み。

eclipseバージョンは"3.4.2"、「/usr/local」以下に展開済み。

[OpenINVENT]: ver4.0.1

「OpenINVENT-4.0.1.tar.gz」を使用。

「/usr/local」以下に展開しコンパイル済み。

(3)環境変数

使用している環境変数は「/etc/bash.bashrc」に集約しています。

(4)その他

「sun-java-jdk6」「python2.6」を使用しています。その他、OpenRTPに関連するパッケージは、(2)のインストール時のパッケージ取得スクリプトによりインストールしています。

=====

【3.サンプルの動作手順について】

(注意事項)事前に「/etc/hosts」を開き、5行目の以下の部分を「#」でコメントアウトしてください。

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

Ubuntu9.04以降とomniORBとの相性の悪さでNameServiceが正常に起動しない場合があります、以下のサンプルの正常起動が出来ない場合があるという理由からです。

OpenRTPをご利用頂くには、サンプルなどを実際にご利用してからイメージして頂く方が、理解しやすいため、ここでは、OpenRTM/OpenHRP/OpenINVENTそれぞれのサンプルをいくつかピックアップして手順を説明します。この他にもサンプルはありますので、4章のリンクにあるそれぞれのWebページを参照頂き、お試し頂けることを願っています。

なお、サンプル動作において、統合開発環境である「Eclipse(エクリプス)」を使用しています。OpenRTPのRtSystemEditor、OpenHRP、OpenINVENTでは、このeclipseのplugin(プラグイン)という機能を使って、GUIを用いた視覚的な操作を行うことが出来ます。以降はeclipseを使う方法での説明をします。

(3-1)OpenRTM-aist 付属サンプル

ここでは、「SimpleIO」というサンプルの説明を行います。2つのRTC(RTコンポーネント)で構成され、片方のRTCでは、端末で入力された数字を、データ送信だけの処理を行います。もう片方のRTCでは、データを受信し、そのデータを端末に表示するというものです。2つのRTCを接続し、入力した数値が送信され表示される一連の処理を期待したものです。

(a)まずは、端末を起動します。Ubuntu デスクトップ画面左上にある「アプリケーション」-「アクセサリ」-「端末」を選択します。

(b)端末内にて、以下の入力を行います。

```
「cd /usr/local/eclipse」
```

```
「sh eclipse.sh」
```

eclipse 画面が起動し始めます。「ワークスペース・ランチャ」というメニューが出てきます。

(c)ここで、作業ディレクトリを指定し、「OK」を押します。

(例:「/home/ubuntu/workspace」 ホームディレクトリに「workspace/」ディレクトリが作成されます。)これにより、eclipse のメイン画面が起動されます。「ようこそ」という画面が表示されたら「X」を押して閉じてください。

(d)画面上部にあるメニューから、

「ウィンドウ」-「パースペクティブを開く」-「その他」を選択します。

"パースペクティブを開く"という小ウィンドウが表示されますので、ここで「RT System Editor」を選択します。これにて、RtSystemEditor が起動されます。

(e)RtSystemEditor の準備を行います。画面左上にある「ON」という水色のボタンを押します。中央に、「System Diagram」というタブが作成されます。

(f)画面左側にある「Name Service View」というタブの中に"RT 127.0.0.1"という表示があり、左側に「△」ボタンがあります。これをクリックすると、ツリーが展開された表示(中身の表示)が行われます。

(g)(a)の手順により、端末をさらに二つ起動します。各々で「cd /usr/share/OpenRTM-aist/examples」とコマンド入力します。

(h)片方の端末で「./ConsoleInComp」、もう片方の端末で「./ConsoleOutComp」と入力します。それぞれの端末内に文字列が表示されます。また、(f)で開いた中に「ConsoleIn0|rtc」「ConsoleOut0|rtc」が表示されます。(表示されない場合、「xxxxxx|host_cxt」の「△」をクリックするとその中に表示

されています。)

この状態で、端末で起動した RTC(RT コンポーネント)が2つ起動されて、NameService に登録されていることとなります。

- (i)"Name Service View"に表示されている「ConsoleIn0|rtc」「ConsoleOut0|rtc」を、(e)で表示した "System Diagram"の方にマウスでドラッグします。青色の四角い Box が表示されます。各々が RTC を表しており、INACTIVE(非活性状態)であることが分かります。
- (j)二つの Box にある"くちばし"のようなものをマウスで接続します。くちばしは「DataPort」を表しており、データの送受信を行う部分です。線を繋ぐことで、データの送り先と受け先を指定した状態になります。(線は DataPort の箇所でもマウスをクリックし、そのまま別の DataPort へドラッグすることで繋がります。)
"Connector Profile"という小ウィンドウが出るので、「OK」を押します。
- (k)RtSystemEditor の中央上部にある「ALL」という緑色のボタンを押します。
"全てのコンポーネントを Activate しても良いですか?"と表示されますので、「OK」を押します。
2つの青色 Box が緑色に変化します。同時に、(h)で入力した端末にさらに文字列が追加表示され、片方に"Please input number"が表示されます。これにて、データ送受信の準備が整いました。
各 RTC は ACTIVE (活性状態)にあります。
- (l)"Please input number"が表示されている端末にて、適当な数字を入力し、Enter を押し、もう片方の端末に、その数字とその他のデータ受信情報が表示されます。これで成功です。
- (m)終了処理を行います。RtSystemEditor の中央上部にある「ALL」という赤色のボタンを押します。
"全てのコンポーネントを Deactivate しても良いですか?"と表示されますので、「OK」を押します。
片方の RTC が青色の四角に戻ります。この状態で、端末にて「Ctrl+d」「Ctrl+c」と入力して RTC を終了させます。"Name Service View"から「ConsoleIn0|rtc」「ConsoleOut0|rtc」の表示が消えます。
これにて、RTC が終了され、NameService からの登録も抹消されたこととなります。
- (n)端末を終了するには、「exit」と入力します。eclipse を終了するには、「ファイル」-「終了」を行います。
- (*)その他、OpenRTM の各種機能に特化したサンプルや、GUI 操作を含むサンプルも用意されていますので、OpenRTM の Web ページのサンプル説明を参照して、ご利用ください。

(3-2)OpenHRP3 付属サンプル

ここでは、「SamplePD」というサンプルの説明を行います。人型ロボットのモデル(VRML 形式)を用意し、決められた動作をするようにプログラムされた RTC と OpenHRP3 のシミュレーション機能をリンクさせ、ロボットが前方に歩く動作を期待したものです。

- (a)(3-1)の(a)-(c)同様にして、eclipse を起動します。(3-1)の(d)と同じ手順で、今度は「GrxUI」というパースペクティブを選択します。これにて、「GrxUI」という OpenHRP3 のシミュレーション用ビューが起動されます。同時に裏では OpenHRP3 を構成する各モジュールが起動されています。適宜ウィンドウ、内部のサブウィンドウなどのサイズは調整ください。
- (b)画面上部のメニューから「GrxUI」-「プロジェクトの読み込み」を選択します。サンプルが格納された一覧が小ウィンドウに表示されます。ここで、「SamplePD.xml」を選択し「OK」を押します。画面左下部の"3D ビュー"タブ内に、青い床面上に直立した緑色のロボットが表示されます。また、画面左側の"アイテム"タブには、「SamplePD」を構成するモデルの情報やコントローラなど各種プロジェクトを構成するものが表示されます。つまり、プロジェクトがロードされたという状態を表しています。
- (c)"3D ビュー"では、マウスの左/中/右ボタン操作により、ズームや回転やスライドなどを行え、見やすいように位置や方向を変更できます。

(d)画面左上部にある"人が走っているような図"のボタンがあると思います、これが、シミュレーションの START ボタンになります。START ボタンを押すと、ロボットが動きだします。

(e)シミュレーション時間にして、約 13 秒後に自動的にシミュレーションは停止します。途中で停止させる場合には、シミュレーション中は、START ボタンが STOP ボタンに切り替わっているので、STOP ボタンを押します。eclipse の終了手順は同様に行います。

(*)その他、「SampleLF」ではロボットが床面の箱を持ち上げるサンプル、「SampleSV」では、車輪型ロボットがクネクネ走行するようなサンプルなど、同様の手順でお試し頂けますので、ご利用ください。

(3-3)OpenINVENT 付属サンプル

OpenINVENT は、OpenRTM-C++/java/python 版、RtSystemEditor、OpenHRP3、RtcHandle ツールなどを利用しております。各々の個別サンプルに加え、全体としての統合サンプルのイメージとしてご利用頂けます。

(*)スペックが高くない PC の場合、全体としての挙動が重くなりますのでご了承ください。

今回は、各機能別のモジュールに分けた 13 個の RTC(C++版 12 個、Java 版 1 個)と OpenHRP3 を用いて、移動台車ロボットが走行するシミュレーションのサンプルを説明します。

このロボットは介助犬をイメージしたものであり、部屋の中である START 位置から GOAL 位置まで、障害物となるものを避けながら走行する一連のシミュレーションです。詳細は OpwnINVENT の Web ページをご覧ください。

ロボットと部屋の環境モデル(VRML 形式)を用意し、RTC として、部屋環境を把握するもの、走行経路を生成するもの、自己位置を算出するもの、走行用の速度指令を送出するもの、障害物を検知するもの、GUI から指令を送出するもの、OpenHRP3 とデータのやりとりをするもの、障害物を検知するための距離センサー機能を有するものなどから構成されています。

以下では、ある部屋環境にて、初期位置からある GOAL 位置まで走行させるが、走行経路の途中に、把握していない障害物(緑の壁)が検知され、それらを避けながら走行し続けるサンプルの操作手順を示していきます。

(*) 既に(3-1)、(3-2)のサンプルにより、細かい操作手順や文言は把握しているものとして、簡略化して説明していきます。

(a)端末を2つ起動します。各々の端末から eclipse を起動します。(2 つ目の eclipse を起動すると、workspace を別に指定する必要があります。「/home/ubuntu/workspace2」など)

(b)片方の eclipse では、パースペクティブを「RtSysmtemEditor」にします。さらに、画面上部のメニューより、「ウインドウ」-「新規ウインドウ」にてもう一つ eclipse を起動します。合計3つになります。新たな eclipse では、パースペクティブを「GrxUI」として開きます。

(c)「GrxUI」にて、(3-2)(b)と同様手順で、OpenINVENT 用のプロジェクトファイルをロードします。「/usr/local/OpenINVENT-4.0.0/Project/GrxUI/GrxUI_Sample.xml」を選択してください。壁で囲われた部屋の中に移動台車ロボットが存在するモデル群がロードされるはずですが。

(d)もう片方の eclipse では、パースペクティブを「InventGUI」として開きます。これは、OpenINVENT の Java 版 RTC(eclipse の plugin 化されたもの)で、GUI 画面の指令やパラメータセットに利用します。

(*)以降の操作は 2 種類を説明します。

RtSystemEditor を使った GUI 操作手順と、RtcHandle というツールを使い、RtSystemEditor を必要としない、CUI 操作手順についてです。お好きな方法でご利用ください。

----[以下、sh スクリプトと RtSystemEditor による起動手順]-----

(e)さらにもう一つ端末を起動し、
「cd /usr/local/OpenINVENT-4.0.1/Script」

と入力します。

- (f) 端末から、「sh Sample.sh」と入力します。利用する RTC が複数起動され、端末がタブとしてまとめられた1つの新たな端末が起動されます。この時、RtSystemEditor の"Name Service View"には多数の RTC が表示されているはずですが。
- (g) 「RtSystemEditor」にて、「System Diagram」を開き、そこでマウス右クリックより、「Open and Restore」を選択します。次のファイル選択画面にて、「/usr/local/OpenINVENT-4.0.1/Project/RtSystemEditor/RTSE_Sample.xml」を選択してください。RtSystemEditor のバグにより、エラー画面が出ますが、無視して進んでください。複数の RTC が各々のポートを接続したものが表示されるはずですが。これが OpenINVENT 全体の RTC 群の構成となります。
- (h) 「RtSystemEditor」にて、緑色の「All」ボタンを押して全 RTC を Activate します。画面上の RTC が緑色に変化するはずですが。(負荷により青から緑に遷移する時間が遅くなる場合があります。)ただし、青色のまま緑に遷移しない RTC (今回は 3 つ) が存在しますが、これらは OpenHRP3 に処理が依存しているため、このままで OK です。
- (i) 「InventGUI」にて、各種指令やパラメータのセットが行えますが、今回はデフォルトのまま利用します。この状態から、画面右上の"2DMapBuilder"タブにある「SET」ボタンを押してください。しばらくすると、InventGUI 画面下部にグリッド上の表示が現れます。これは、「GrxUI」で読み込んだ部屋モデルの内、グレーの壁を既知の障害物とみなした場合に、OpenINVENT 内部で保持する 2D 地図情報を表示したものです。
- (j) 「InventGUI」にて、次は、画面左上に6つ並んだボタンの内の「SET」ボタンを押してください。画面下部に赤い 5 角形のもの (移動ロボットを表す) と青い枠 (GOAL を表す)、そして、青いグリッド列 (経路) が表示されるはずですが。この状態で、ロボット初期位置からと GOAL までの走行すべき計画経路が指定されたこととなります。
- (k) 「GrxUI」にて、(3-2)(d)同様に、シミュレーション START ボタンを押す。シミュレーションは開始されるので、あとは、ロボット走行開始指令を送るのみの状態です。この時、「InventGUI」ではロボットに搭載された距離センサーがセンシングしている領域を表す、紫色の扇形が表示されるはずですが。
- (l) 「InventGUI」にて、画面左上に6つ並んだボタンの内の「START」、「GO」ボタンを連続で押します。走行が開始され、「GrxUI」でのシミュレーション走行が見られると思います。また、「InventGUI」では、OpenHRP3 から取得している位置情報から、走行した軌跡が逐次表示されているはずですが。緑色の壁を未知の障害物として、回避しながら GOAL へ向かうシミュレーションが得られると思います。
- (m) 終了する場合は、シミュレーション STOP をした後に、全 RTC を Deactivate してください。起動している OpenINVENT の RTC 群も端末で「ctrl+c」で停止させてください。

-----[以下、CUI ツール(RtcHandle)による起動手順]-----

- (n) (a)の手順から開始します。「RtSystemEditor」が不要なため、端末を 2 つ起動した後に、eclipse から、「GrxUI」「InventGUI」を開きます。
- (o) さらにもう一つ端末を起動し、
「cd /usr/local/OpenINVENT-4.0.1/Script/RtcHandle」
と入力します。さらに、以下のように入力しツールを起動します。
「python Sample.py」
RTC 群が起動され(f)同様に端末が起動されます。また、自動的に RTC 同士の Port 接続やパラメータセットを行い、準備が来ると、コマンド入力を促された状態になるはずですが。
- (p) 端末にて「a」と入力します。これで、全 RTC が Activate されます。(h)の操作と同じことを行っています。以降は(i)から(l)までの手順は同じです。シミュレーションの START/STOP まで行えます。

(q) 終了処理は、シミュレーション STOP 後に、端末にて「d」と入力し Deactivate させた後に、「f」でスクリプトを終了させます。RTC 群の端末も「ctrl+c」などで停止させてください。

【4.その他】

OpenRTP に関するその他の詳細は、以下のリンクより Web ページをご参照ください。
また、コピーライトやライセンスについても、上記から参照ください。

- (1) OpenRTM-aist Web ページ : 「<http://www.openrtm.org/OpenRTM-aist/html/>」
- (2) OpenHRP Web ページ : 「<http://www.openrtp.jp/openhrp3/jp/>」
- (3) OpenINVENT Web ページ : 「<http://www.openrtp.jp/INVENT/>」