



# GraspPlugin Manual

[ホーム](#) > [HIRO](#)

## HiroNXGUI について

火, 11/01/2011 - 17:45 — asahi

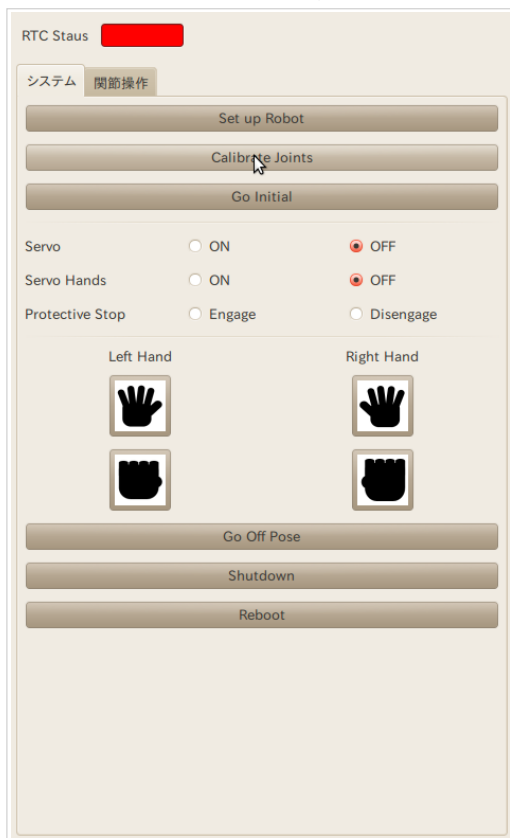
HiroNXGUI は、HiroNXProvider のフロントエンドとして、双腕ロボット HIROの機能呼び出すことのできる RTコンポーネントです。

## 起動方法

端末を開いて HiroNXGUI をインストールしたディレクトリに移り、

```
% ./WxHiroNXGUI.py
```

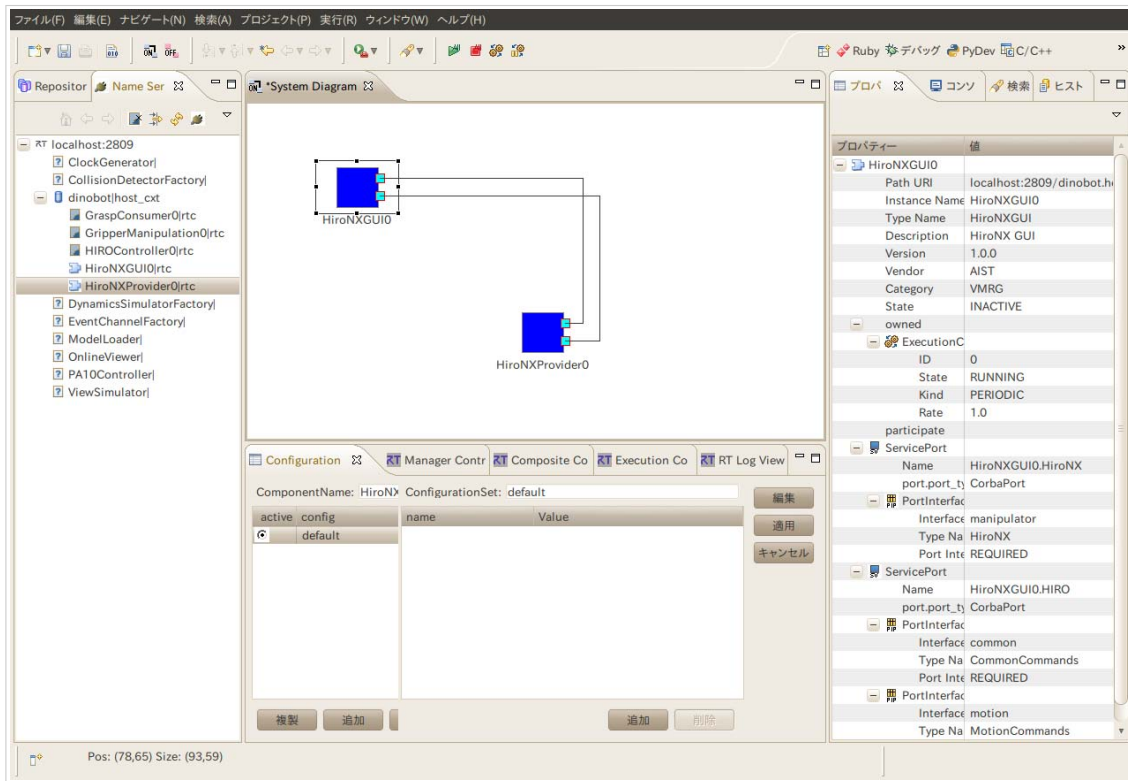
を実行します。下図のようなGUIが表示されます。



## RTCの接続

HiroNXGUI の持つ HiroNXポートとHIROポートを、EclipseのRTシステムエディタや rtcon コマンド等を使って HiroNXProvider と接続します。

RTシステムエディタで接続する場合は、上のポート同士、下のポート同士を接続します。(上図では、左上が HiroNXGUI、右下が HiroNXProvider)



rtcon で接続する場合は、以下のようなコマンドラインを使用します。(localhost/vision.host\_cxt は、実行環境によって変わります)

```
% rtcon localhost/vision.host_cxt/HiroNXGUI0.rtc:HIRO localhost/vision.host_cxt/HiroNXProvider0.rtc:HIRO
% rtcon localhost/vision.host_cxt/HiroNXGUI0.rtc:HiroNX localhost/vision.host_cxt/HiroNXProvider0.rtc:HiroNX
% rtact localhost/vision.host_cxt/HiroNXGUI0.rtc localhost/vision.host_cxt/HiroNXProvider0.rtc
```

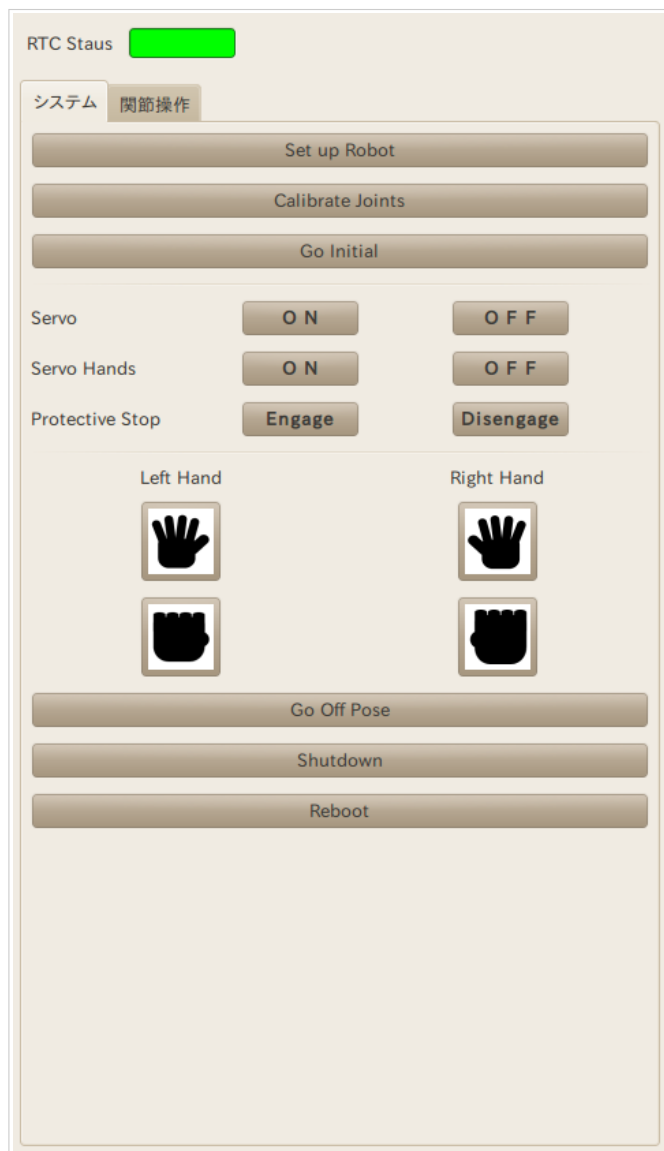
## 使い方

HiroNXGUIはふたつのタブを持ち、それぞれのボタンやスライダー等を操作して、HIROをコントロールします。

上のRTC Statusが赤から緑になったら、アクティベートされたことになります。

(HiroNXProviderと接続されているとは限らない)

## システムタブ



上段のボタンは、HIROの起動に使用します。

Set up Robot → Calibrate Joints → Go Initial と、上から順にボタンを押していき、初期姿勢に移行したら、Servo Hands の ON を押します。

Set up Robot

HIRO を起動する

Calibrate Joints

関節を調整する

Go Initial

初期姿勢に移行する

Servo ON/OFF

HIRO 本体のサーボのスイッチ

Servo Hands ON/OFF

ハンド部分のサーボのスイッチ

Protective Stop Engage/Disengage

中段のボタンは、HIROの操作に使用します。

Left Hand(開き手アイコン)

左ハンドを開く

Left Hand(握り手アイコン)

左ハンドを閉じる

Right Hand(開き手アイコン)

右ハンドを開く

Right Hand(握り手アイコン)

右ハンドを閉じる

下段のボタンはHIROの終了に使います。

まず Go Off Pose で待機姿勢にしたあと、上段ボタンの Servo Hands OFF, Servo OFF を押してサーボを停止します。

その後、Shutdown を押すとHIROの電源が落ち、HiroNXGUIが終了し、vision のネットワークが切れます。

Go Off Pose

待機姿勢をとる

Shutdown

HIRO の電源を落とし、HiroNXGUI を終了する

Reboot

HIRO を再起動する

## 関節操作タブ

HIROの関節角を直接操作するためのスライダーやテキストボックスを用意しています。

RTC Staus ●

システム 関節操作

Go

左肩ヨー	15	右肩ヨー	-15
15		-15	
左肩ピッチ	0	右肩ピッチ	0
0		0	
左肘ピッチ	-143	右肘ピッチ	-143
-143		-143	
左手首ヨー	0	右手首ヨー	0
0		0	
左手首ピッチ	0	右手首ピッチ	0
0		0	
左手首ロール	0	右手首ロール	0
0		0	
左ハンドJ0	0	右ハンドJ0	0
0		0	
左ハンドJ1	0	右ハンドJ1	0
0		0	
左ハンドJ2	0	右ハンドJ2	0
0		0	
左ハンドJ3	0	右ハンドJ3	0
0		0	

首ヨー	0
0	
首ピッチ	60
60	
胴体	0
0	
土台	0
0	

リセット

Go ボタン

下のスライダーで設定した関節角を、HIROに送信します。

左腕パネル・右腕パネル

肩のヨー・ピッチ、肘のピッチ、手首のヨー・ピッチ・ロール、ハンドの4つの関節の角度を、スライダーやテキストボックスを使って入力できます。

首・胴体パネル

首のヨー・ピッチ、胴体・土台の関節角度を入力できます。

リセットボタン

すべての関節の角度を初期姿勢に戻します。(HIROへの送信は行わないので、改めてGoボタンを押してください。)

## 終了方法

以下のどれかの方法で、HiroNXGUIを終了できます。

1. RTシステムエディタで HiroNXGUI0 を右クリックして、コンテキストメニューで exit を選ぶ
2. `rtexit localhost/(PC名).host_cxt/HiroNXGUI0.rtc` を実行する
3. GUI 画面でタイトルバーの×ボタンをクリックする
4. shutdown ボタンを押して HIRO をシャットダウンすると、同時に終了する

## 環境設定

Ubuntu 10.04 の場合、以下のパッケージをインストールしてください。

```
python(2.6)
python-wxgtk2.8
OpenRTM-aist-Python(1.0)
```

## インストール方法

### subversion レポジトリからソースを入手する場合

(あらかじめ、パッケージから subversion を導入してください)

リポジトリからチェックアウトします。

```
% svn co https://hironx-interface.googlecode.com/svn/HiroNXInterface/HiroNXGUI/
```

実行したディレクトリに HiroNXGUI が掘られ、その下にソースファイルが展開されます。  
展開したら、即実行できます。

## ソースプログラムについて

プログラム本体の WxHiroNXGUI.py は、python で書かれたプログラムです。

python ソースプログラム

WxHiroNXGUI.py: メインGUIプログラム(名前にWxがつくの注意)  
HiroNXGUI.py: コンシューマRTCプログラム(主に onInitialize, onExecute などのアクションメソッドを記述)  
Joint.py: ロボットの関節情報管理クラス  
WxHelper.py: GUI制御ライブラリ  
bodyinfo.py: HIRO姿勢データ

実行時に読み込む設定ファイル

rtc.conf: RTコンポーネントの設定ファイル  
joints.txt: 関節操作タブでの関節の並び順と表示名を定義。関節名の頭の一文字が、LかRかそれ以外かで別々のパネルに振り分けられる。---は上の項目との間に隙間を開ける。  
main.wrl: 関節操作タブでの関節の可動域を、このファイルから読み取る。  
HiroNXGUI.xrc: GUI リソースファイル

アイコン画像ファイル(左右のハンドの開閉ボタンに貼られる)

lclose.png  
lopen.png  
rclose.png  
ropen.png

l/rで左右、open/closeで開閉を示す。

以下のファイルは、開発者が WxFormBuilder で GUI画面 を作成するためのファイル。

HiroNXGUI.fbp: HiroNXGUI.xrc の元となる。

以下のファイルは、開発者がRTコンポーネントのスケルトンを生成するために利用する。

HIROController.idl: HIROController の IDL定義ファイル

HiroNX.idl: HiroNX の IDL定義ファイル

RTC.xml: RT ビルダーのソース XML

idlcompile.sh: IDLコンパイルのためのバッチプログラム

以下すべて、IDL から生成された py ファイル。

HIROController\_idl.py

HiroNX\_idl.py

./\_GlobalIDL:

\_\_init\_\_.py

./\_GlobalIDL\_\_POA:

\_\_init\_\_.py

## 機能追加について

HiroNXGUI に機能を追加する場合、ほとんどは HiroNX.idl に新しい命令を追加することになるはずです。

ここでは例として、newCommand メソッドを追加してみます。

なお、開発用のPC(Ubuntu 10.04)には、以下のパッケージが必要です。

wxFormBuilder

omniidl4

## IDL ファイルを更新する

HiroNXGUI ディレクトリにある HiroNX.idl を直接修正してもかまわないですが、対応するプロバイダRTC(今回は HiroNXProvider )ですでにIDLに命令を追加しているなら、それをコピーする方が早くて間違いがないでしょう。(参照:[HiroNXProvider について](#))

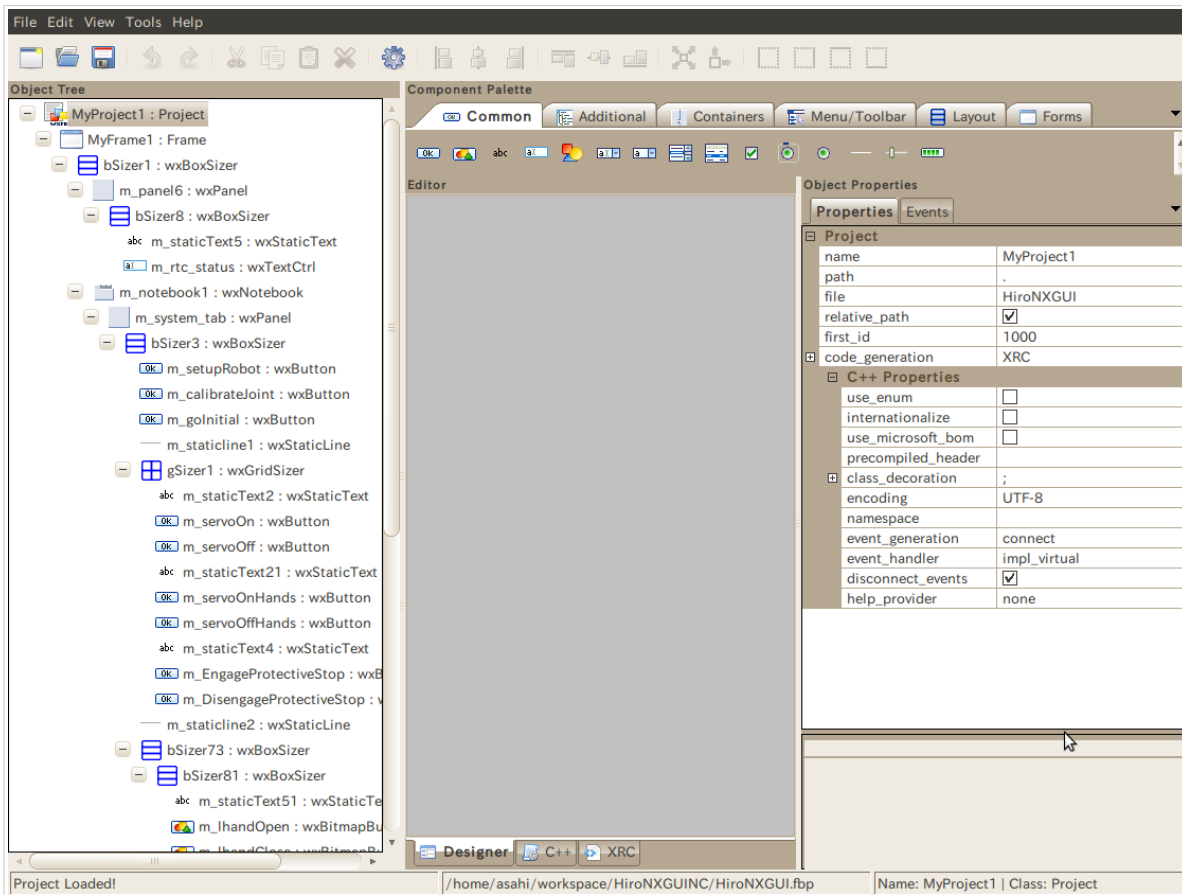
更新したIDL を omniidl でコンパイルします。

```
% omniidl -bpython HiroNX.idl
```

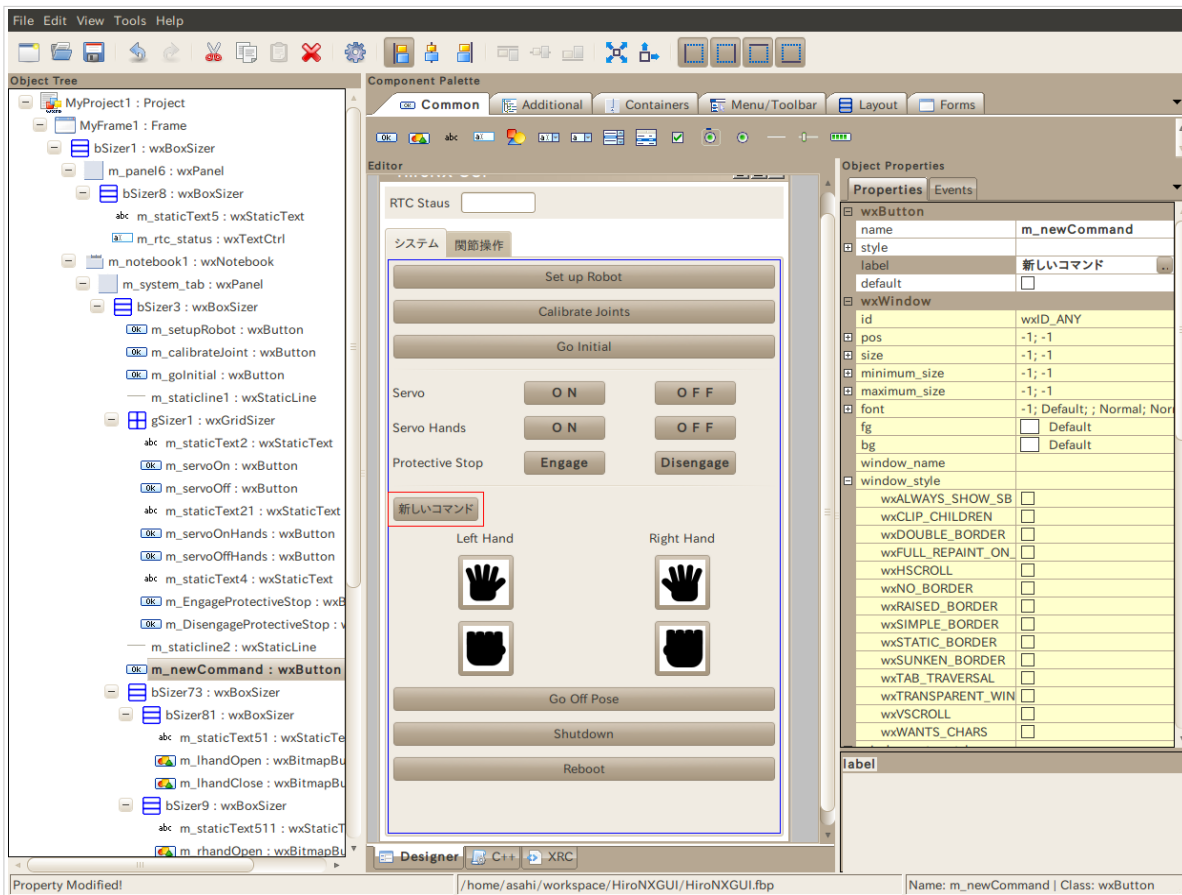
HiroNX\_idl.py が更新されます。

## wxFormBuilder でボタンを追加する

wxFormBuilder を立ち上げて、HiroNXGUI.fbp を開きます。



左側のObject Tree ペインの、MyProject1 - MyFrame1- bSizer1 - m\_notebook1 - m\_system\_tab - bSizer3 以下に、新しいボタンを追加します。



新しいボタンの位置ですが、おおむねHIROの起動時に使用するボタン(Set Up Robot, Calibrate Joints, Go Initial Servo ON/OFFなど)は上に、終了時に使用するボタン(Go Off Pose, Shutdown, Reboot など)は下に、配置しています。  
今回は Protective Stop と Hand Open の間に配置することになります。

追加するコマンド名が newCommand なので、新しいボタンの name プロパティも「m\_newCommand」とします。WxHiroNXGUI.py では、

システムタブのボタンについては、ボタン名から `m_` を取ったものをコマンドとみなしプロバイダRTCを呼び出す仕組みになっているので、新しいボタンには必ず正しい名前を設定しなくてはなりません。

labelプロパティには任意のラベルを設定できるので、ここでは「新しいコマンド」とします。

ボタンを追加したら、これをもとに `HiroNXGUI.xrc` を生成します。

File メニューから `Generate Code` を選んでください。

もちろん、`wxFormBuilder` を終了するときにセーブするのも忘れずに。

作業はこれで完了です。`WxHiroNXGUI.py` を起動して、「新しいボタン」をクリックしてみましょう。

```
% ./WxHiroNXGUI.py
./WxHiroNXGUI.py
.
comp_args: HiroNXGUI
gen HiroNXGUI
CommandEvent
m_newCommand newCommand
RTC not connected.
```

何も接続していませんが、`m_newCommand` のクリックに対して、端末に応答メッセージが出力されます。

---

[◀ HiroNX の操作手順](#)[↑ 上位](#)[HiroNXProvider について ▶](#)

---

[印刷用ページ](#)   [ログイン \(登録\)](#)してコメントを投稿

