roswifibot

ROS Driver for Wifibot lab mobile robot. More information on the official webpage

It is based on low-level "`libwifibot`" driver, available here and here "`libwifibot`" is wrapped within this ROS package.

This package is a fork of the original "`roswifibot`" package, available on SourceForge.

The fork has been ported to catkin and maintained with recent versions of ROS. Retro-compatibility has been kept with rosmake (older versions of ROS, for instance `fuerte`).

Features

- "`wifibot_node`": low-level robot control
- "`hokuyo_node`": low-level Hokuyo laser driver
- "`camera1394`": low-level Firewire camera driver
- "`turtlebot_teleop`": keyboard-based teleoperation
- "`rviz`": vizualisation

Licence

BSD

Authors

Original authors according to the "`manifest`": claire, jean-charles

Fork maintainer: Arnaud Ramey (arnaud.a.ramey@gmail.com)

Compile and install

ROS Fuerte + rosmake

Dependencies with ROS Fuerte:

```
$ sudo apt-get install  ros-fuerte-robot-model  ros-fuerte-navigation  ros-fuert
e-laser-drivers  ros-fuerte-viz ros-fuerte-perception ros-fuerte-camera1394  ros
-fuerte-turtlebot-apps
```

Compile with rosmake (older versions of ROS, for instance fuerte):

```
$ cd cmake ; bash package2rosmake.bash
$ rosmake roswifibot
```

Note: to revert the package back to catkin-compliant:

```
$ cd cmake ; bash package2catkin.bash
```

ROS Indigo + catkin

Compile with catkin:

```
$ catkin_make --only-pkg-with-deps roswifibot
```

Run

1) Run the robot driver:

```
$ roslaunch roswifibot robot_launch.launch
```

2) If there is a **Hokuyo** laser, run the driver in a new terminal:

```
$ roslaunch roswifibot robot_launch.launch
```

3) If there is a Firewire camera, run the driver in a new terminal:

```
$ roslaunch roswifibot firewire.launch
```

4) If there is a **Kinect** camera, run the driver in a new terminal:

```
$ roslaunch roswifibot robot_launch.launch
```

Note: steps 1) to 4) can be replaced with a one-liner:

```
$ roslaunch roswifibot robot_kinect_joy_launch.launch
```

5) Run **joypad-based teleoperation** in a new terminal:

```
$ roslaunch roswifibot joy_teleop.launch
```

6) Run **keyboard-based teleoperation** in a new terminal:

```
$ roslaunch roswifibot keyboard_teleop.launch
```

7) Run **"`rviz`"**, the vizualisation tool, in a new terminal:

```
$ roslaunch roswifibot rviz.launch
```

Publications

- `/camera/depth/image` Kinect depth image
- `/camera/rgb/image_color` Kinect RGB image
- `/image_raw` Firewire camera

Troubleshooting

Problemm: compilation error

When you launch

```
$ catkin-make
```

you obtain a message as:

```
"roswifibot-master/msg/Status.msg: [roswifibot-master/Status] is not a legal type name"
```

or

```
"ERROR: package name 'roswifibot-master' is illegal and cannot be used in message generation."
```

Solution: You must rename you package folder to "roswifibot", not "roswifibot-master".

Problem: roscore error.

When you launch

```
$ roscore
```

Error on screen:

```
Param xml is <param command="rosversion ros" name="rosversion"/>
Invalid <param> tag: Cannot load command parameter [rosversion]: command [rosversion ros] returned with code [1].
```

Solution: Based on this link

```
$ sudo apt-get install python-rospkg
```

Problem: Acces denied to devices (robot or Hokuyo).

Solution:

```
$ sudo chmod a+rwx /dev/ttyS* /dev/ttyACM*
```

Problem: Connexion with the robot is slow (several seconds).

Few TF messages:

```
  $ rostopic hz /tf
```

returns less than 100Hz. Topic "`/odom`" is not published.

Cause: Another executable is already connected with the robot.

Solution: kill all processes "`robot_server*`". Run:

```
  $ ps aux | grep robot_server
```

and kill all all associated PIDs. Then stop and relaunch the launch file.

Problem: Firewire camera is not recognized.

Solution: Run "`coriander`" and check camera is recognized.

Problem: Hokuyo laser range finder dies.

When launching "`wifibot_node`", the Hokyuo device is suddenly turned off, (or any other plugged on the robot); the device "`/dev/ttyACM0`" disappears

Solution: Based on ros.org The problem is in fact linked with the electrical relays of the Wifibot. The orders sent to the robot can cut off the relays (electrical supplies) for the devices plugged on the robot. By default, the orders shut down these relays and so the Hokyuo device is shut down.

The "`wifibot_node`" ROS driver has been modified to enable such configuration. For instance, set as command-line argument "`_relay1:=true`" to activate the first relay.

Note: at the driver level, you could call "`setRelays(true, true, true)`" to enable them.