rosxwiimote

Description

A ROS package to control Wiimotes.

The official ROS `wiimote` package, included in `ros-kinetic-wiimote` Ubuntu package, is based on the CWiiD library (official page) and only works with older Wiimotes (`Nintendo RVL-CNT-01`). This package is based on the newer `xwiimote` (official page) and is also compatible with newer Wiimotes (`Nintendo RVL-CNT-01-TR`) and Nunchuks.

Install

Bluetooth configuration and pairing

First check Kernel module `hid-wiimote` is loaded.

```
$ sudo modprobe hid-wiimote
$ lsmod | grep wii
Expected output:
hid_wiimote            XXXX  0
```

To make the loading of this module permanent after boot:

```
$ sudo nano /etc/modules
Add:
hid-wiimote
```

Add current user to group "input" (from here):

```
$ sudo usermod -aG input $USER
```

Pair your Wiimote with your computer using `blueman` (Bluetooth device manager) and connect to HID:

```
$ sudo hcitool dev
Devices:
  hci0  00:19:0E:16:AF:22
$ sudo hcitool scan
Scanning ...
  40:F4:07:C5:B7:BD Nintendo RVL-CNT-01-TR
$ blueman-manager
```

Press the red sync button on the back of the WiiMote, the 4 leds will blink. Then in the Bluetooth device manager:

```
"Search" button
  Nintendo RVL-CNT-01-TR
  Peripheral
  40:F4:07:C5:B7:BD
Right click > Pair
Right click > Connect to HID
```

See image "blueman.png". Check it works:

```
$ sudo evtest
/dev/input/event15: Nintendo Wii Remote Accelerometer
/dev/input/event16: Nintendo Wii Remote IR
/dev/input/event17: Nintendo Wii Remote
/dev/input/event18: Nintendo Wii Remote Nunchuk
/dev/input/event19: Nintendo Wii Remote Motion Plus
```

Troubleshooting:

If you are prompted for PIN input, it is because you pressed 1+2 instead of the red sync button. If it still asks for the PIN code with the red button, you can try "Installation > pair without PIN code".

xwiimote installation

`xwiimote` (version 2-3build1) is included in Ubuntu packages. Unfortunately, you need to compile the latest version from sources if you want the Nunchuk to be recognized. To do so:

```
$ sudo apt purge xwiimote libxwiimote2
$ sudo apt install  libudev-dev  libncurses-dev
$ git clone https://github.com/dvdhrm/xwiimote.git
$ cd xwiimote
$ sh autogen.sh
$ make
$ sudo make install
```

Then to use it:

```
$ xwiishow list
Listing connected Wii Remote devices:
  Found device #1: /sys/devices/pci0000:00/0000:00:1d.0/usb5/5-2/5-2:1.0/bluetoo
th/hci0/hci0:12/0005:057E:0330.0002
End of device list

$ xwiishow 1
See image "xwiishow1.png"
```

You might need to resize the font of the terminal to see the extensions (Nunchuk for instance). For instance, size 10 is enough: see image "xwiishow2.png" You can also use xterm that has a small font:

```
$ xterm -e xwiishow 1
```

Troubleshooting

If you get the following error:

```
$ xwiishow
Xwiishow: error while loading shared library: libxwiimote.so.2: cannot open shar
ed object file: No such file or directory
```

Then your `LD_LIBRARY_PATH` environment variable must be incomplete:

```
$ export | grep LD_LIBRARY_PATH
```

It should contain "/usr/local/lib". Otherwise append at the end of the file and save:

```
$ nano ~/.bashrc
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

Then reload your environment variables:

```
$ source ~/.bashrc
```

Check the README inclued in the `xwiimote` project in case of trouble.

Licence

See LICENCE

Usage

Parameters

- `~device_idx` [int, default:-1] The index of the Wiimote device to use. Starts at 1, so if you want to use the seconde Wiimote, use `_device_idx:=2` Leaves at -1 to skip this parameter, in this case you need to specify the device index with `~device_path`.
- `~device_path` [std::string, default:""] The full path of the Wiimote device to use. It's an absolute sysfs path to the device's root-node. This is normally a path to `/sys/bus/hid/devices/[dev]/`. You can use this path to create a new struct xwii*iface object. Leaves empty (`""`) to skip this parameter, in this case you need to specify the device index with `~device*idx`.

Subscriptions

- `~fb` [sensor_msgs/JoyFeedback] Feedback on the Wiimote: turn on or off rumble and LEDs.
- `~rumble` [std_msgs/Float32, seconds] Turn the rumble on for a given duration, in seconds. Given durations will be clamped in the (10 ms, 10 s) span.

Publications

- `~joy` [sensor_msgs/Joy] Acquired Wiimote state. List of buttons and axes:

4 axes:

```
0. left-right rocker (3 possible values: -1=left 0=released 1=right)
1. up-down rocker (3 possible values: -1=left 0=released 1=right)
2. nunchuk left-right joystick (floating value in the range -1=left .. 1=right)
3. nunchuk down-up joystick (floating value in the range -1=down .. 1=up)
```

9 buttons (O=released, 1=pressed):

```
0. XWII_KEY_A
1. XWII_KEY_B
2. XWII_KEY_PLUS
3. XWII_KEY_MINUS
4. XWII_KEY_HOME
5. XWII_KEY_ONE
6. XWII_KEY_TWO
7. XWII_KEY_C
8. XWII_KEY_Z
```

See `launch/test.launch` for an example.

To test LEDs:

```
LED 3 on:
$ rostopic pub /xwiimote_node/fb sensor_msgs/JoyFeedback '{type: 0, id: 2, intensity: 1}'
LED 3 off:
$ rostopic pub /xwiimote_node/fb sensor_msgs/JoyFeedback '{type: 0, id: 2, intensity: 0}'
```

To test rumble:

```
On:
$ rostopic pub /xwiimote_node/fb sensor_msgs/JoyFeedback '{type: 1, intensity: 1
}'
Off:
$ rostopic pub /xwiimote_node/fb sensor_msgs/JoyFeedback '{type: 1, intensity: 0
}'
Timed:
$ rostopic pub /xwiimote_node/rumble std_msgs/Float32  .7
```