ROS <=> Android tutorial

Introduction

This tutorial is supposed to show you how to connect your Duckiebot to your Smartphone via Bluetooth. Tutorial will focus on connection and robot side programming to bake Bluetooth available, not on Android Programming using Java (since this might be too complicated for those who never programmed for Android before). Instead program for Android will be created using MIT App Inventor web application to demonstrate proof of concept. Using Bluetooth on your robot is the core part of this exercise, not programming for your Smartphone.

----

1. Setting your Bluetooth device

First, install all required software for Bluetooth to work. All commands are executed on your Duckiebot, so connect to it now.

Install Bluetooth support:

```
duckiebot:~$ sudo apt-get update
duckiebot:~$ sudo apt-get install bluez bluetooth bluez-tools
```
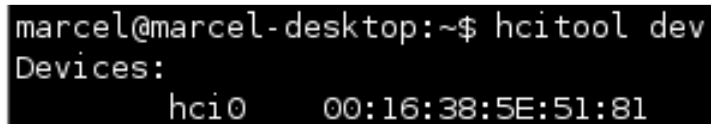
Plug in your Bluetooth adapter into your Duckiebot. Bring up your Bluetooth adapter:

```
duckiebot:~$ sudo hciconfig hci0 up
```

Check if adapter is seen in the system:

```
duckiebot:~$ hcitool dev
```

You should see something like this:

Img 1: hcitool output.

If you don't see any devices, you probably need to install the driver for your Bluetooth adapter. This might happen if your Bluetooth adapter is rare or too new.

Install python support for the Bluetooth now.

```
duckiebot:~$ sudo apt-get install python-pip python-dev ipython
duckiebot:~$ sudo apt-get install bluetooth libbluetooth-dev
duckiebot:~$ sudo pip install pybluez
```

OK, setup is done.

[EXTREMELY IMPORTANT] Changing your Bluetooth adapter's address if possible.

Bluetooth device BD addresses should be generally unique. This is an extremely important rule, and with good Bluetooth adapters it is being followed. However, cheap adapters (like CSR 4.0) usually have <u>THE SAME</u> BD address coming from manufacturer. This is a terrible practice, which will lead to awful consequences in case you're using several bluetooth adapters in a small area, but this is the world we live in, deal with it.

It is impossible to change BD address for many adapters, and Bluez stack does not support changing of BD addresses as well.

For SOME (not all, just some) of the Bluetooth chips, it is possible to change BD address by using an additional program called "bdaddr". However, you need to download it and assemble manually:

```
duckiebot:~$ cd ~/
duckiebot:~$ wget -U "Mozilla" http://www.petrilopia.net/wordpress/wp-
duckiebot:~$ content/uploads/bdaddrtar.bz2
duckiebot:~$ mv bdaddrtar.bz2 bdaddr.tar.bz2
duckiebot:~$ tar xvjf bdaddr.tar.bz2
duckiebot:~$ cd bdaddr
duckiebot:~$ sudo apt-get install libbluetooth-dev
duckiebot:~$ make
```

Now, use "bdaddr" to change the BD Address of your device.

```
duckiebot:~$ sudo ./bdaddr -i hci0 00:10:20:30:40:50
```

!!! PICK SOME UNIQUE BD ADDRESS FOR THIS COMMAND, DO NOT USE THE EXAMPLE 10:20:30:40:50. MAKE SURE BD ADDRESS IS UNIQUE AMONG ALL YOUR BLUETOOTH DEVICES !!!

Unplug your Bluetooth adapter and Plug it again.

Check if BD address changed with:

```
duckiebot:~$ hcitool dev
```

Not all Bluetooth adapters support this feature of changing BD address. Only SOME (not all) of the chips from following manufactures are known to have this possibility: Ericsson, CSR, Texas Instruments, Broadcom, Zeevo and ST.

----
2. Pairing Duckiebot with a Smartphone

UBUNTU 14.04:

First, set your Bluetooth name same as your duckiebots' (you need to distinguish it somehow).

duckiebot:~$ sudo hciconfig hci0 name your_duckiebot_name

Next, make your device discoverable:

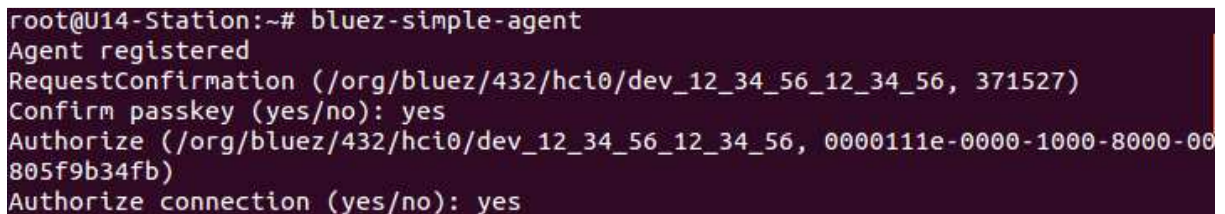duckiebot:~$ sudo hciconfig hci0 piscan

Now, launch Bluetooth agent on your Duckiebot:

duckiebot:~$ sudo bluez-simple-agent

Take your smartphone, find your robot's Bluetooth and pair with it.

On your Duckiebot, say "yes" to PIN confirmation request and to connection request. Wait for a moment (about 1 minute), and then close bluez-simple-agent by pressing Ctrl+C.

Img 2: Confirm pairing in Bluetooth Agent (say "yes" when asked).

----

3. Creating Android application using MIT AppInventor

Go to http://ai2.appinventor.mit.edu

Here you can create a simple application for your android without unspeakable horrors of Android Studio.

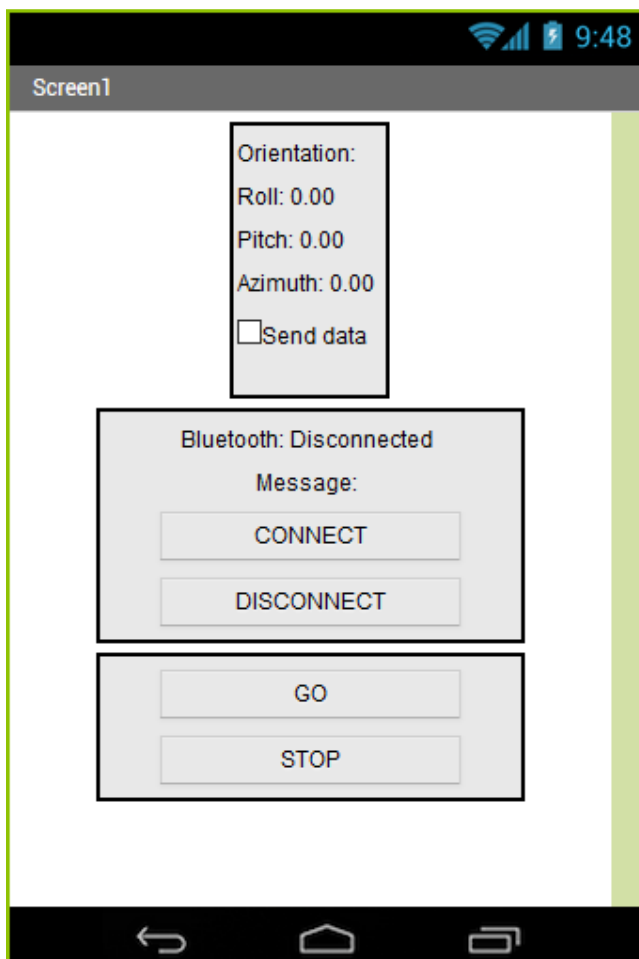For this demo, you may use already provided example.

Download the project file from this link:

https://github.com/YuryXW/csp_bluetooth_practice/blob/master/BluetoothIMU.aia?raw=true

Import this project in AppInventor, then build and download it (use the QR code for this task, it is convenient).

You can edit this program however you want. All blocks have commentaries to tell you what they do.

Direct APK Link.

https://github.com/YuryXW/csp_bluetooth_practice/blob/master/BluetoothIMU.apk?raw=true

Img 4: Smartphone app to be used in this demo.

----
4. Download the demo package for this tutorial

First, add your user ("ubuntu" by default) to the "dialout" group, so you'll have an access to virtual serial ports created for Bluetooth.

```
duckiebot:~$ sudo useradd -G dialout ubuntu
```

Next you need to register the Serial Port service. Do this after each reset.

```
duckiebot:~$ sudo sdptool add –channel=22 SP
```

[**IMPORTANT**] This command should give you an output with successful result "Serial Port service registered". If no output was given, everything is broken. Yes, again. Keep calm, deep breath, it's fixable. To fix this problem do this:

Edit the "`/etc/systemd/system/dbus-org.bluez.service`" file.

```
duckiebot:~$ sudo nano /etc/systemd/system/dbus-org.bluez.service
```

Change this line:

```
ExecStart=/usr/lib/bluetooth/bluetoothd
```

To this:

```
ExecStart=/usr/lib/bluetooth/bluetoothd --compat
```

Run this command as well:

```
duckiebot:~$ sudo chmod 777 /var/run/sdp
```

Reboot your Duckiebot:

```
duckiebot:~$ sudo reboot now
```

After reboot, execute this command again:

```
duckiebot:~$ sudo sdptool add –channel=22 SP
```

Change to your working folder, please. Now, git clone the package for this tutorial:

```
duckiebot:~$ cd your_duckietown_working_folder
duckiebot:~$ git clone
https://github.com/YuryXW/csp_bluetooth_practice.git
duckiebot:~$ cd ~/duckietown
duckiebot:~$ source environment.sh
duckiebot:~$ cd catkin_ws
duckiebot:~$ catkin_make
duckiebot:~$ cd ..
duckiebot:~$ source environment.sh
duckiebot:~$ source set_ros_master.sh
duckiebot:~$ source set_vehicle_name.sh your_robot_name
```

Launch the test node:

```
duckiebot:~$ roslaunch csp_bluetooth_demo csp_bluetooth_demo.launch
veh:=your_robot_name
```

Connect to your robot from the cellphone app you've downloaded.

Press "CONNECT" and select your robot's Bluetooth.

You can do two things now. First is to drive around your robot using the Smartphone. First, make sure you can control your duckiebot with a joystick (demo node will publish to the same topic as joystick does, "/your_robot_name/joy_mapper_node/car_cmd". Press "back" button on the joystick to be sure you've enabled its state in Duckiebot.

Press "GO" to make robot move forward until you press "STOP".

Now, check "Send Data" checkbox, this will enable sending the orientation sensor (accelerometer) data to your Duckiebot. Demonstration node you're using is designed to make your robot steer based on these readings, try to control your Duckiebot this way. Tilt your phone forward and backward to control velocity, and from side to size to control rotation. Tick "Send data" off when you've played enough.

Now, test how the phone will react to robot seeing the red line. Move to the red line SLOWLY, and when your Duckiebot will detect it, your phone will vibrate.

Additional tasks:
For these tasks, you may need to modify the "csp_bluetooth_demo_node.py" file (located in
"your_working_folder/csp_bluetooth_practice/csp_bluetooth_demo/src"

1. Change roll and pitch, so that now Roll controls velocity and Pitch controls omega, so you
can hold your phone in landscape mode.
        HINT: Check "message_handler" function.

2. Make your robot stop after it sees the red line and vibrates. There are two ways doing so, you
either change your node code, OR you can change the Android code so that your Smartphone will
send command for robot to stop after it receives the "VIBRATE" message.