(DRAFT) - ROS MAINTAINANCE ON RASPBERRY PI

ROS **Kinetic** has been installed from source on the Raspbian distribution.

The **ROS-Comm** variant has been chosen, i.e. only the barebones communication system is available to implement topics, nodes, services and parameters, along with the needed dependencies.

As for the Ubuntu installation the workspace is located in: `~/ros_catkin_ws` and also the installation command has been modified to choose the same installation directory of the Ubuntu version: `/opt/ros/kinetic` **(check also below)**

The `source` command has already been added to `.bashrc` so it should not be needed before operations.

UPDATING AND REBUILDING THE WORKSPACE

Firstly we make sure to be in the workspace directory:

```
$ cd ~/ros_catkin_ws
```

Also we make sure to have an updated system:

```
$ sudo apt-get update && sudo apt-get upgrade
```

Then we move our existing rosinstall file in order to not get it overwritten:

```
$ mv -i kinetic-ros_comm—wet.rosinstall kinetic-ros-comm—full-
wet.rosinstall.old
```

Then we generate the new rosinstall file from the updated source:

```
$ rosinstall_generator ros_comm --rosdistro kinetic --deps —wet-
only --tar > kinetic-ros_comm-wet.rosinstall
```

Then we compare the two files to check which packages have changed:

```
$ diff -u kinetic-ros_comm-wet.rosinstall kinetic-ros_comm-
wet.rosinstall.old
```

If we are satisfied with the changes we can incorporate the new file in the workspace and update it:

```
$ wstool merge -t src kinetic-ros_comm-wet.rosinstall
$ wstool update -t src
```

Finally we can rebuild it from last sources:

```
$ sudo ./src/catkin/bin/catkin_make_isolated --install -
DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/kinetic -j2
```

Two options here are **very** important:

`--install-space /opt/ros/kinetic` to keep the Ubuntu installation folder

`-j2` to only start two compilation jobs and not four otherwise the Pi may (already happened) freeze for lack of memory

It's not a bad decision to update the environment variables on the fly too:

```
$ source /opt/ros/kinetic/setup.bash
```

ADDING ADDITIONAL PACKAGES (not tested yet)

To add new packages to the ROS workspace we have to add to the rosinstall file (which for now only includes ros_comm) all the other packages we may need from the ros ecosystem and then rebuild the whole worskpace (similiarly to the procedure described in the previous section). So for example if the ros_foobar package is needed:

We first create our custom rosinstall file:

```
$ cd ~/ros_catkin_ws
```

```
$ rosinstall_generator ros_comm ros_foobar --rosdistro kinetic --
deps --wet-only --tar > kinetic-custom_ros.rosinstall
```

Then we update the workspace:

```
$ wstool merge -t src kinetic-custom_ros.rosinstall
$ wstool update -t src
```

Then we install any missing new dependencies through `rosdep`:

```
$ rosdep install --from-paths src --ignore-src --rosdistro kinetic
-y -r --os=debian:jessie
```

Then we rebuild the whole workspace:

```
$ sudo ./src/catkin/bin/catkin_make_isolated --install -
DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/kinetic -j2
```

(**NOTE**): As you can see the rosinstall file name depends on which variant of ROS you want to install. By adding a package to ros_comm we are creating a new rosinstall file (in this example called `kinetic-costum_ros.rosinstall)` so we must use it for any additional command. Note also that if you want to add other packages **all the previous ones must be reintegrated** (as in this example we reintegrated ros_comm) in the rosinstall file otherwise they will not be merged when rebuilding the workspace.