Introduction to ROS: ROS beginner level tutorials

The objective of this lab is getting to know fundamental concepts of ROS.

SETTING UP YOUR COMPUTER

Open terminal window. Clone your git repository to your home folder

```
$ git clone URL_for_<yourname>-rtech_repository
```

Use **ls** to confirm that ***<yourname>-rtech*** has been downloaded to your home folder**.**

Let's set up *.gitignore* to avoid uploading **build**, **devel**, and **install** directories. Create a new file with the following content:

```
build/
devel/
install/
```

Save this file as ***.gitignore*** in ***<yourname>-rtech***. Push the file to your git cloud storage. Do you remember how to do that?

BEFORE YOU START WITH ROS TUTORIALS

In this lab you will be following a selection of ROS beginner tutorials available online. Throughout the course we are using **Kinetic** distribution of **ROS**. Every now and then, the tutorials use a placeholder for your ROS distribution - use your better judgement to make adjustments as needed.

NB! When completing the following tutorials, always remember that *<yourname>-rtech* will be your catkin workspace.
That means, whenever you need to type *catkin_ws*, type *<yourname>-rtech*, instead.

Next, let's create a catkin workspace and remember that **your git repository is your catkin workspace**.

```
$ cd ~/<yourname>-rtech
$ mkdir src
$ cd ~/<yourname>-rtech/src
```

Even though the workspace is empty (i.e., there are no packages in the **src** folder) you can already *build* the workspace:

```
$ cd ~/<yourname>-rtech
$ catkin_make
```

The `catkin_make` command is a convenience tool for working with catkin workspaces. If you list your current folder you should now see additional **build** and **devel** folders. Inside the **devel** folder you can see that there are now several **setup.*sh** files. Sourcing any of these files will overlay this workspace on top of your environment. **Before continuing source your new setup.*sh file by typing**:

```
$ source devel/setup.bash
```

In order to verify the last command. check that `ROS_PACKAGE_PATH` environment variable includes the your personal catkin workspace. Type:

```
$ echo $ROS_PACKAGE_PATH
/home/<youruser>/<yourname>-rtech/src:/opt/ros/kinetic/share
```

CORE ROS TUTORIALS

Now that your work environment is all setup, continue with the ROS file system tutorial. And remember to substitute *catkin_ws* with ***<yourname>-rtech*** and that you are working on **ROS Kinetic**.

1. Navigating the ROS Filesystem

   This tutorial introduces ROS filesystem concepts, and covers using the `roscd`, `rosls`, and `rospack` command line tools.
2. Creating a ROS Package

   This tutorial covers using roscreate-pkg or catkin to create a new package, and rospack to list package dependencies.
3. Building a ROS Package

   This tutorial covers the toolchain to build a package.
4. Understanding ROS Nodes

   This tutorial introduces ROS graph concepts and discusses the use of `roscore`, `rosnode`, and `rosrun` command line tools.
5. Understanding ROS Topics

   This tutorial introduces ROS topics as well as using the `rostopic` and `rqt_plot` command line tools.
6. Understanding ROS Services and Parameters

   This tutorial introduces ROS services, and parameters as well as using the `rosservice` and `rosparam` command line tools.
7. Using rqt_console and roslaunch

   This tutorial introduces ROS using `rqt_console` and `rqt_logger_level` for debugging and `roslaunch` for starting many nodes at once.
8. Creating a ROS msg and srv

   This tutorial covers how to create and build msg and srv files as well as the `rosmsg`, `rossrv` and `roscp` command line tools.
9. Writing a Simple Publisher and Subscriber (C++)

   This tutorial covers how to write a publisher and subscriber node in C++.
10. Examining the Simple Publisher and Subscriber

    This tutorial examines running the simple publisher and subscriber.
11. Writing a Simple Service and Client (C++)

    This tutorial covers how to write a service and client node in C++.
12. Examining the Simple Service and Client

    This tutorial examines running the simple service and client.
13. Recording and playing back data

    This tutorial will teach you how to record data from a running ROS system into a .bag file, and then to play back the data to produce similar behavior in a running system.
14. Getting started with roswtf

    Basic introduction to the `roswtf` tool.

CLEAN UP YOUR WORKSPACE

NB! Before you leave the lab, make sure you have pushed all the files in your catkin workspace to your git cloud service.

In terminal, `cd` to **<yourname>-rtech**

Type

```
git config user.email "youremail@example.com"
```

Type

```
git status
```

You should now see all the new and modified files in red.

Prepare the relevant files for the commit.

```
git add file_name_in_red1 file_name_in_red2
```

When you now type

```
git status
```

you should see all the added files in green. You are now ready to commit changes. Type

```
git commit –m "Insert a brief explanation"
```

Your changes have now been committed but not yet uploaded to the cloud. To upload your files, type

```
git push
```

In your web browser, **verify that all the files** have been uploaded to the **<yourname>-rtech** repository.

Delete the **<yourname>-rtech** folder and any other files you created from the lab's computer.