

## Lab 1: Introduction to ROS

Instructor: Yuhas Michael John

Name: Le Quang Anh, StudentID: U1922940J



*This lab and all related course material on [F1TENTH Autonomous Racing](#) has been developed by the Safe Autonomous Systems Lab at the University of Pennsylvania (Dr. Rahul Mangharam). It is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#). You may download, use, and modify the material, but must give attribution appropriately. Best practices can be found [here](#).*

**Course Policy:** Read all the instructions below carefully before you start working on the assignment, and before you make a submission. All sources of material must be cited. The University Academic Code of Conduct will be strictly enforced.

## 1 Workspaces and Packages

### 1.1 Written Questions

1. CMakeList file contains a set of directives and instructions describing the project's source files and targets. It is the input to the CMake build system for building software packages. Any CMake-compliant package contains one or more CMakeLists.txt file that describe how to build the code and where to install it to.

Makefile is a simple way to organise code compilation for C/C++ objects.

CMakeList is related to makefile. Make (or rather a Makefile) is a buildsystem. It drives the compiler and other build tools to build your code. CMake is a generator of buildsystems. It can produce Makefiles, it can produce Ninja build files, ect (Reference: [stackoverflow](#)).

2. Yes, we also use CMakeLists.txt for Python in ROS

There are no executable object created for python. Python is an interpreted language.

3. We run `catkin_make` in the workspace folder to build all the packages
4. Sourcing `setup.bash` file is so that we can setup the environment, because ROS relies on the notion of combining spaces using the shell environment (Reference: [ROS Wiki](#)).

## 2 Publishers and Subscribers

### 2.1 Written Questions

1. Answer here
2. There is no nodehandle object in Python. `rospy.init_node()` initializes ROS node with a specified name for the rospy process.

3. We use `rospy.spin()` for Python. The command stays in an infinite loop until receiving a shutdown signal, and processes any events that occur (Reference: [ROS Forum](#)). The node with the subscriber will do the callback whenever it receives new data from the topic, until it receives a shutdown signal.
4. Answer here
5. Answer here

## 3 Implementing Custom Messages

### 3.1 Written Questions

1. Answer here
2. `Header` is a special type in ROS, which contains a timestamp and coordinate frame information that are commonly used in ROS (Reference: [ROS Wiki](#)). I can also extract that from the topic and include in the message file.

## 4 Recording and Publishing Bag Files

### 4.1 Written Questions

1. The bag file gets saved in the current directory. To change where it is saved, we need to add the path to the `-o` argument when calling `rosbag record`.
2. By default, it will be in the `/.ros` folder. We can change where it is saved by editing the `args` for the node in the launch file (Reference: [ROS Forum](#)).