# Electronics Design Lab Report #1

This week I finished watching the tutorials Programming for Robotics (ROS) Course suggested last lesson and simpler tutorial about using ROS (https://www.youtube.com/watch?v=ehtUb55Rmmg&list=PLk51HrKSBQ8-jTgD0qgRp1vmQeVSJ5SQC).

Last is very simple beginner tutorial that helped me to learn the key commands in ROS and understand how it functions. While doing my first assignment I read a lot of ROS WIKI pages about how to create package etc. Additionally, I analyzed github code that is written by TA which is very good in explanation. My homework could be done primary because of these codes. Unfortunately, my code still had a lot of bugs, that's why I was forced to contact TA for advice that helped me a lot. From there I learned how to debug my program and actually how the bags occur, what causes them and how to prevent making them.

Here I want to summarize the key fact information that I learned this week:

- ROS is operating via the nodes, each part of project that performs its operation
- Nodes operate with each other via messages, each node can post the message to topic which is some environment and each node can subscribe to this topic to receive the message.
- Master is the branch that controls all the nodes, i.e. registers and stores the information
- Rviz is visualization tool that can show sensor information or even perform the simulation
- There are many ros commands to better interact in the shell which are very similar to unix commands
- Several packages i.e. projects are stored in one catkin workspace that builds the environment
- Ros files mostly are written using c++ or python
- We use multiple shells to see the interaction of the nodes in the project
- Another way to launch many files simultaneously is to use .launch files where one can write several launching commands in one file.
- Bach should be sourced every time the command window opens
- Build your catkin workspace catkin_ws_make everytime the code is changed to build these changes

Thank you.