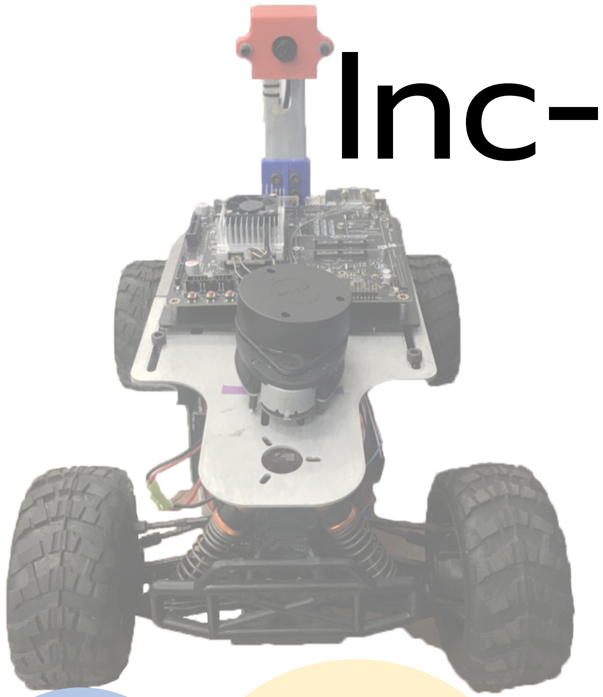
Inc-Racing 2020 Lecture2



ANDAMAN

KMUTT

KMUTT INC

we make computers do more.

Agenda

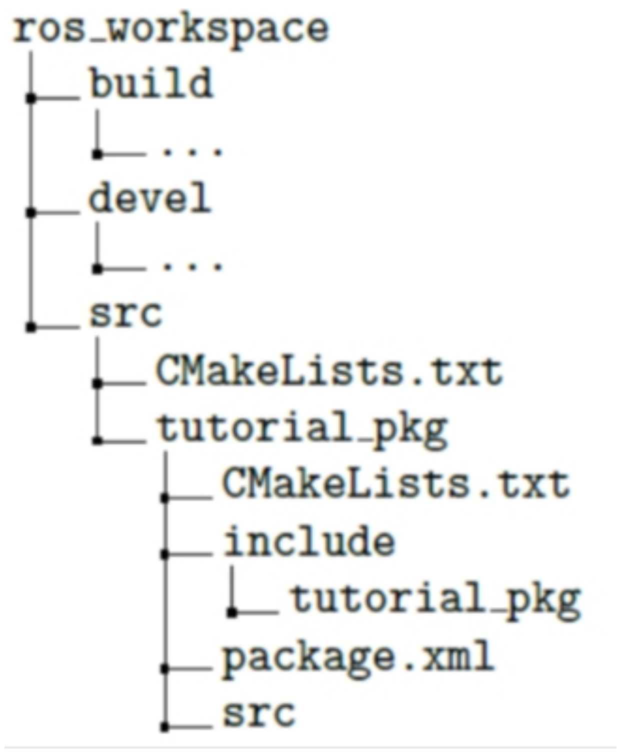
ANDAMAN KMUTT KMUTT INC we make computers do more.

ROS Workspace

Programing

- ROS Program's workspace
- Workspace can keep various of package
- code to use = catkin_init_workspace

Workspace structure



```
ros_workspace
├── build
│   └── ...
├── devel
│   └── ...
└── src
    ├── CMakeLists.txt
    └── tutorial_pkg
        ├── CMakeLists.txt
        ├── include
        │   └── tutorial_pkg
        ├── package.xml
        └── src
```

How to create workspace

```
$ mkdir -p~lcatkin_ws/src
$ cd~lcatkin_ws/src
$ catkin_init_workspace
```

Calling the command

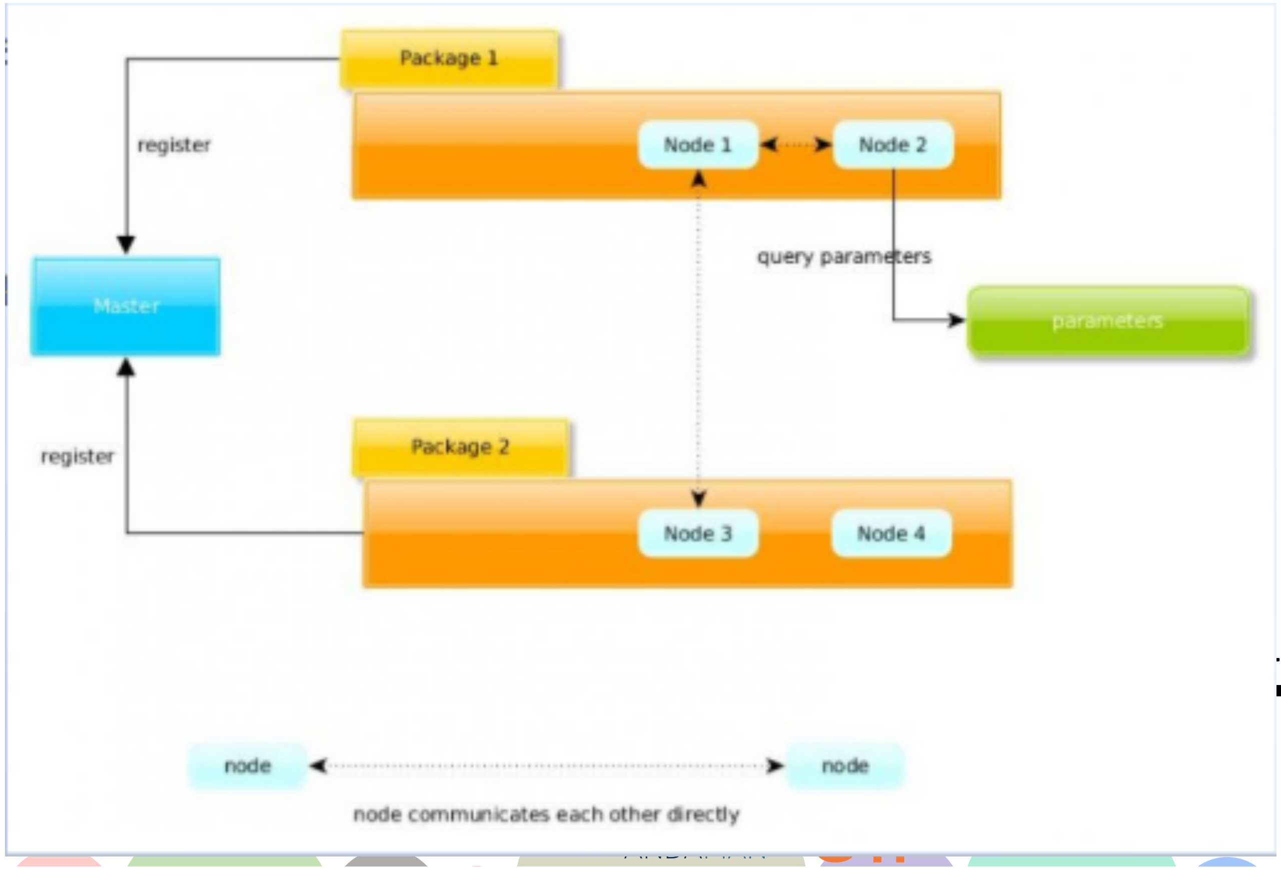
```
$ source ~/catkin_ws/devel/setup.bash
```

Or adding it to file .bashrc in Home Directory

Exercise

- Designing your own robot car
- try to build Workspace named “catkin_ws”
- Build it with catkin_make
- Use the command ls to see the file

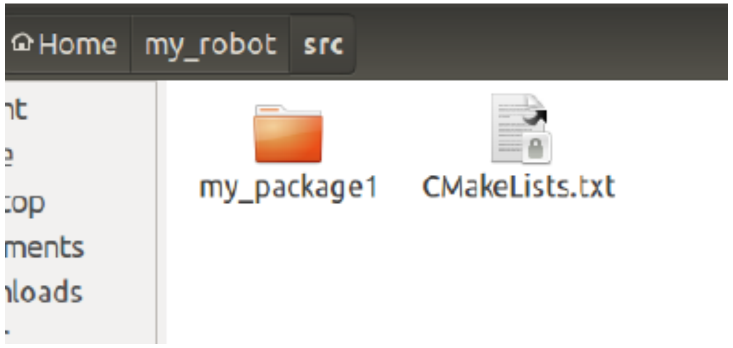
- 1 workspace contain various Package
- 1 Package contain various Node
- Package is using when we get into work

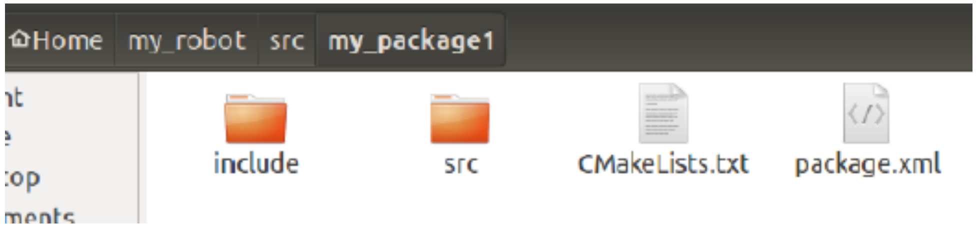


How to create Package

- cd catkin_ws
- cd src

```
$ catkin_create_pkg my_package1 std_msgs roscpp
```





example of file and directory that been built when type the command catkin_create_pkg

Important file

1. CmakeLists.txt

-File that collect the data for building the program (convert to program that can be a file to collect the data for build program or executable file)
-Shouldn’t be deleted and the developer should understand the structure of this file. So, we can be able to create the program

2. package.xml

- Text file keeps in form of XML (extensible Markup Language) that can describe the data connect to the package such as name, version, owner or any library connected to the package

Node

Process that perform the program

catkin_ws

```
$cd ~/catkin_ws/src/my_package 1/src
Stouch mynode 1.cpp
```

Starting the program and writing code

```
#include <ros/ros.h>
int main(int argc, char **argv) {
        ROS_INFO ( "Hello ROS");
}
```

Writing code in “CMakeLists.txt”

```
## Declare a C++ executable
## With catkin_make all packages are built wj
##The recommended prefix ensures that targe1
add _executable (mynodel src/mynodell.cpp)

##Rename C++ executable without prefix
##The above recommended prefix causes long 1
## target back to the shorter version for eas
target_link_libraries(mynodel ${catkin_LIBRARIES})
```

File name

Node name

Building

ยชน์

building the file

1) go to “~/my_robot” as following

```
$ cd
$ cd my_robot
or $ cd ~/my_robot
```

2) calling the command “catkin_ make”

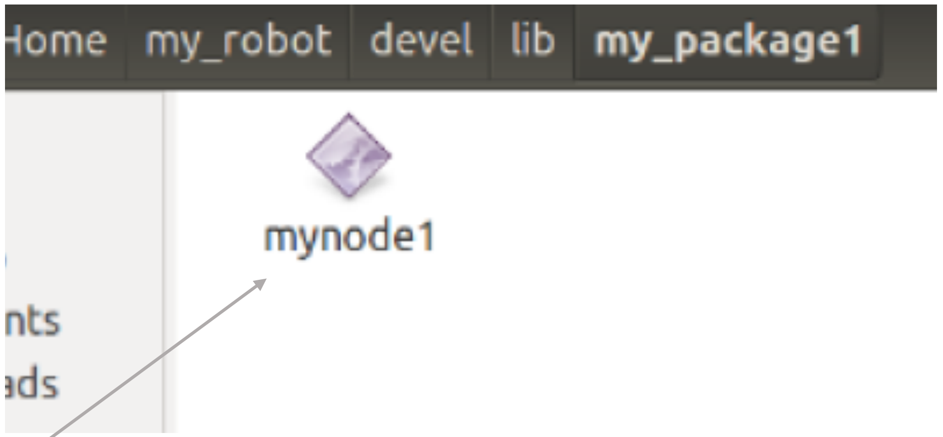
```
$ catkin_make
```

catkin_ws

Lecture2

ANDAMAN KMUTT INC we make computers do more.

Checking the file

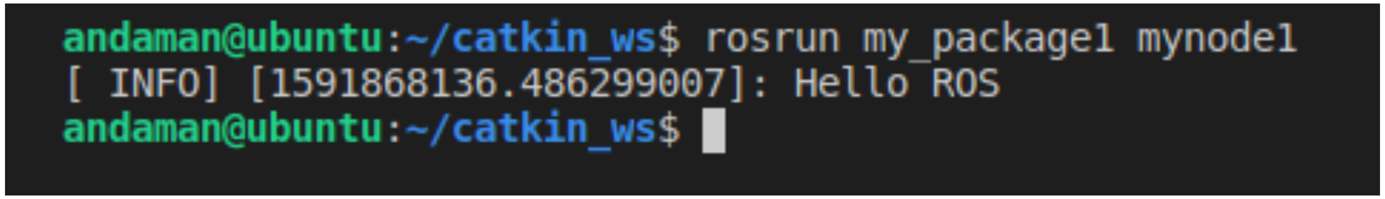


catkin_ws

Run program

Setting the environment of ROS so it can call and use Node

when we build the file called “mynode1.cpp” ROS file cannot call it to work yet because we cannot direct to the position of the file “mynode1” but we can test by the command “rosrun”



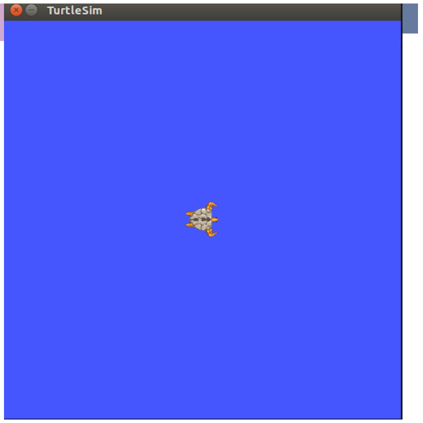
```
andaman@ubuntu:~/catkin_ws$ rosrun my_package1 mynode1
[ INFO] [1591868136.486299007]: Hello ROS
andaman@ubuntu:~/catkin_ws$ 
```

Trying some tools in ROS

- rospack profile
- rospack find [package name]
- rospack list
- rospack depends-on [package name]
- rospack depends [package name]
- rosrun

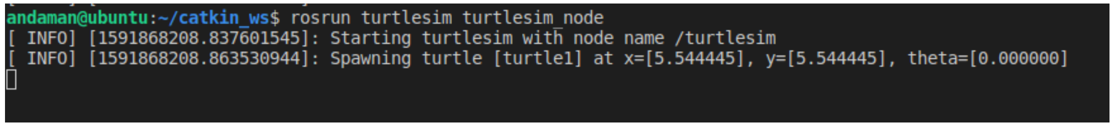
- rosnode list
- rosnode ping [node name]
- rosnode info [node name]
- rosnode machine [PC_NAME or IP]
- rosnode kill [node name]
- rosnode cleanup

Leaning from "turtlesim"



- Easy simulation process of robot
- by use the turtle as the simulator after installing ROS

- Calling the command "rosrun turtlesim turtlesim_node"



```
andaman@ubuntu:~/catkin_ws$ rosrun turtlesim turtlesim_node
[ INFO] [1591868208.837601545]: Starting turtlesim with node name /turtlesim
[ INFO] [1591868208.863530944]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```

- Test the turtlesim

Thank