
ROS

About ROS

- Software framework for building robots
- Tools
  - Flexible messaging service
  - Pose elimination
  - Localization
  - Navigation
- Libraries
  - Main Client libraries
    - rospy, roscpp
  - Experimental libraries
    - rosjava, rosgo

Building a ROS Workspace

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin_make
```

This is where you create your ros workspace. ROS uses a tool called catkin, to build its directories.

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

Here, this will allow your workspace is properly overlayed.. To avoid writing this line everytime you use catkin add it to computers your setup file.

```
$ echo $ROS_PACKAGE_PATH
```

Check if you setup your ROS directory properly. You should get something along the lines to this:

```
/home/youruser/catkin_ws
```

What is in a ROS workspace?

catkin

- src
  - <package>
    - CMAKELists, msg, package, scripts, src
- build
  - <package>
    - Files from scr will be build and put in respected folder
- devel
  - <package>
    - libraries, executables, or generated code
- install
  - <package>
    - Similar to build, but it contains setup.sh files

Adding the arduino driver to your workspace

```
$ sudo apt-get install ros-kinetic-rosserial-arduino
$ sudo apt-get install ros-kinetic-rosserial
```

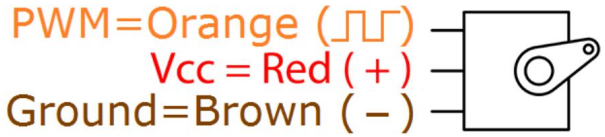
This will install ROS binaries onto your ROS workstation

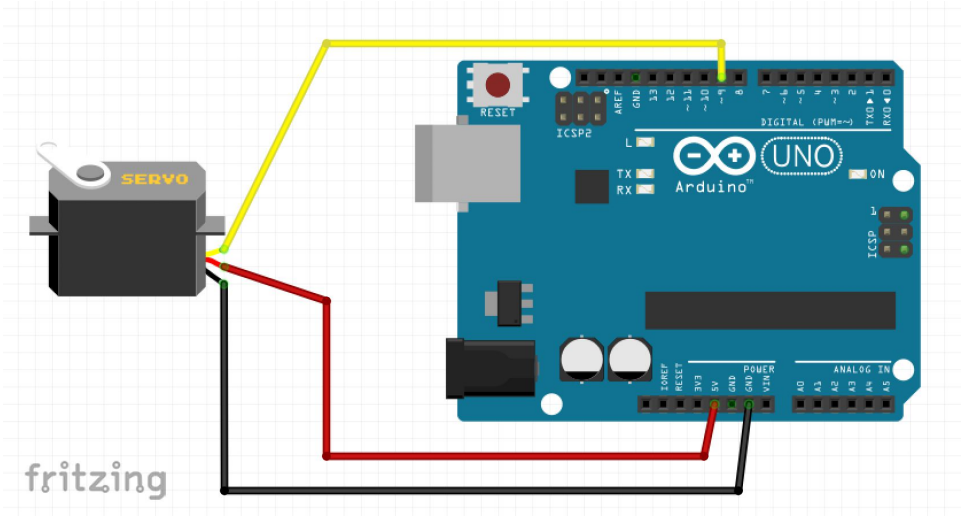
```
$ cd <sketchbook>/libraries ;replace sketchbook with where you normally save your arduino sketches
$ rm -rf ros_lib
$ rosrun rosserial_arduino make_libraries.py .
```

This will install ros_lib onto your Arduino environment, to enable ROS enviromnet to interact with your arduino.

What are we doing with the Arduino?

We are going to communicate with Arduino to tell the servo how many degrees it should turn. First add the code to the sketchbook and set up the arduino and servo as it follows:





Communicating with ROS

`$ roscore`

Run ros and tell it we are going create a node

`$ rosrun rosserial_python serial_node.py /dev/ttyACM0`

Establish connection to arduino so it to communicate with you

`$ rostopic pub servo std_msgs/UInt16 <angle>` ;change <angle> to a num from 0-180

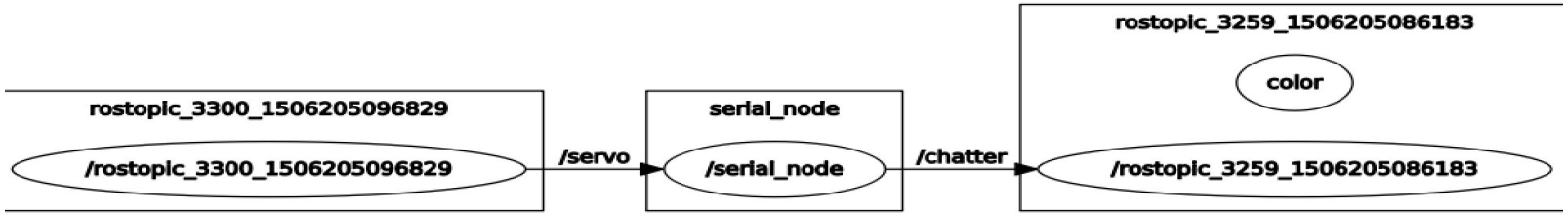
Control servo with rostopic

`$ rostopic echo chatter`

Check out the messages that are sent from the servo

*Note all these commands are done in separate terminals

ROS Nodes


rostopic_3300_1506205096829
/rostopic_3300_1506205096829
/servo
serial_node
/serial_node
/chatter
rostopic_3259_1506205086183
color
/rostopic_3259_1506205086183