

## Getting started Manual InMoov ROS Project

# Content

Introduction.....	3
Kubuntu 16.04 LTS.....	3
Installation.....	3
ROS Kinetic Kame.....	3
Installation .....	3
Getting Started .....	3
MoveIt.....	3
Installation.....	3
Getting Started .....	3
InMoov.....	4
Installation of some packages needed for InMoov.....	4
Setup .....	4
MoveIt .....	4
Arduino.....	5
Msg Srv.....	5
IDE Roboware Studio.....	5
Very Good Guide to the Packages.....	6
inmoov_bringup .....	6
inmoov_firmware_old? .....	6
inmoov_meshes .....	6
inmoov_tools .....	6
inmoov_controls .....	6
This package contains the following:.....	6
inmoov_msgs .....	7
inmoov_description .....	7
inmoov_servobrain .....	7
.....	7

## **Introduction:**

This Project includes three major pieces of software. The intended operating system is Ubuntu 16.04 LTS newer versions probably work just fine but 16.04 is the official version ROS Kinetic Kame is supported on. We chose a slightly different version Kubuntu 16.04 LTS . . Also we use the version of MoveIt matching to ROS Kinetic Kame which needs to be installed for the simulation and motion planning involved.

## **Kubuntu 16.04 LTS**

### ***Installation***

The current setup runs on Kubuntu 16.04 LTS as it is the primary target for ROS Kinetic. Kubuntu needs to be installed first here are some manuals to do this.

-Installation Manual (en):

<https://userbase.kde.org/index.php?title=Kubuntu/Installation&oldid=356826>

-Installation Manual (de):

[https://wiki.ubuntuusers.de/Kubuntu\\_Installation/](https://wiki.ubuntuusers.de/Kubuntu_Installation/)

## **ROS Kinetic Kame**

### ***Installation***

The InMoov ROS Project currently runs on ROS Kinetic Kame which comes with very good documentation and an installation manual. <http://wiki.ros.org/kinetic/Installation/Ubuntu>

### ***Getting Started***

ROS is a elegant system for controlling robots but it has various mechanisms and ideas which need to be understood in order to efficiently use it. Therefore it is a good idea to play around with some or all of the tutorials at this point. <http://wiki.ros.org/ROS/Tutorials>

## **MoveIt**

### ***Installation***

Motion Planning, Collision checking in Simulation is done using MoveIt we need to install this after ROS. Make sure you chose the correct version corresponding to ROS Kinetic Kame <http://moveit.ros.org/install/>

### ***Getting Started***

MoveIt also Provides some good Tutorials which can be found here: [http://docs.ros.org/kinetic/api/moveit\\_tutorials/html/index.html](http://docs.ros.org/kinetic/api/moveit_tutorials/html/index.html)

# InMoov

## Installation of some packages needed for InMoov

After Installing Kubuntu and ROS it is necessary to install a few extra packages which might not be included in the main installation. Here is a command to do just this.

```
sudo apt-get install arduino dkms qt5-default qtcreator clang-format-3.8 ros-kinetic-desktop-full ros-kinetic-rqt ros-kinetic-rqt-common-plugins ros-kinetic-moveit ros-kinetic-rosserial-arduino ros-kinetic-rosserial ros-kinetic-xacro ros-kinetic-opencv3 ros-kinetic-cv-bridge ros-kinetic-compressed-depth-image-transport ros-kinetic-compressed-image-transport ros-kinetic-usb-cam ros-kinetic-pyros ros-kinetic-trac-ik liburdfdom-tools python-wstool python-catkin-tools python-pip
```

Next Step we need to install a Python Package:

```
sudo pip install qdarkstyle
```

## Setup

The first step is to setup a new catkin workspace.

<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

After completing this copy the xenial\_kinetic branch of InMoov ROS packages into the src folder you just created. These packages can be found here:

[https://github.com/TheOnlyRoxion/inmoov\\_ros/tree/xenial\\_kinetic](https://github.com/TheOnlyRoxion/inmoov_ros/tree/xenial_kinetic)

The Next step is to navigate in a Terminal to your catkin workspace and run the Commands „catkin\_make“ to build all packages and „. devel/setup.bash“ to let ROS know about all of your new ROS packages.

To check if everything went smoothly run the following commands in Terminal:

- start roscore = `roscore`
- open a new Terminal and run = `roslaunch inmoov_description demo.launch`

Now RViz and a Joint\_state\_publisher UI should start.

After everything is started you should be able to use Joint\_state\_publisher to move the robot.

## MoveIt

The Model used for Simulation in the InMoov packages should be properly setup.

Should the need arise to modify InMoov's model inmoov\_description contains those folders. The movit folder contains the setup\_assistant. You can launch this in Terminal using “roslaunch inmoov\_description setup\_assistant.launch” which also comes with a good Tutorial.

[http://docs.ros.org/kinetic/api/moveit\\_tutorials/html/doc/setup\\_assistant/setup\\_assistant\\_tutorial.html](http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/setup_assistant/setup_assistant_tutorial.html)

For some changes e.g. joint limits or kinematics it might be more useful to change those values directly in the corresponding file. You can find those under /inmoov\_description/moveit/config .

## **Arduino**

In order to use Arduino we need to setup a few things Start at 2.2.

[http://wiki.ros.org/roscpp/arduino/Tutorials/Arduino IDE Setup](http://wiki.ros.org/roscpp/arduino/Tutorials/Arduino%20IDE%20Setup)

## **Msg Srv**

If changing or adding new messages or services, add them to Arduino libraries follow this manual.

[http://wiki.ros.org/roscpp/Tutorials/Adding Other Messages](http://wiki.ros.org/roscpp/Tutorials/Adding%20Other%20Messages)

## **IDE Roboware Studio**

This is useful for everything link to installation and manual (read it) here:

<http://www.robware.me/#/home>

# Comprehensive Guide to the Packages

## ***inmoov\_bringup***

This package contains the following:

- config
  - contains Settings concerning Arduino/servo configuration min/max/pin/etc. These should not be used. A more up to date version is located at inmoov\_controls.
- include
  - contains a function to import servo settings into pytho and to export servo settings from param server. Also A Python class definition for servos
- launch
  - contains a basic launch file which should not be used for current InMoove setup (it loads the wrong settings files)
- nodes
  - contains python code to launch a couple of nodes

## ***inmoov\_firmware\_old?***

This package contains the following:

- This package is not used by Henry InMoov
- it is designed for different hardware

## ***inmoov\_meshes***

This package contains the following:

- all .stl files to display InMoov simulation

## ***inmoov\_tools***

This package contains the following:

- head tracker is a set of script that uses tf to point the head to a specific Position
- demo scripts/launch files which alsouse kinect/webcams
- servobus launch connects Arduino to ros (inmoov\_contols contains up to date version)

## ***inmoov\_controls***

This package contains the following:

- Arduino
  - Arduino sketch to upload into InMoove (Note: needs to be adjusted for left/right)
  - InmoovTranslator (not used)
- config

contains InMoov servo limits, simulation rotation limits etc.

-include( not used)

-launch

-control\_bringum.launch launches the ros side of arduino communication

-params.yaml (not used)

-scripts

-inmoov\_topic\_splitter.py subscribes to joint\_states and Publishes (/joint\_states\_r/l) reduced messages that are more suitable for serial speeds

-inmoov\_translator\_server.py (not used)

-srv (not used)

### ***inmoov\_msgs***

-defines data structure for rosmessages and services

### ***inmoov\_description***

This package contains the following:

-launch

-display.launch launches Rviz with a given xarco descriptor file.

-gazebo.launch alansrobotlab: First Pass at Gazebo

-moveit-display.launch launches Rviz with movit with a given xarco file etc.

-moveit

This folder is auto generated when running the .setup\_assistant. It comes with various launch files that show important things.

-robots/urdf

these folders contain urdf/xarco files that describe the robot structure. It is setup in a Tree layout going from Joint to Joint including the proper .stl and relative Pose.

### ***inmoov\_servobrain (not used)***