Installation Guide

Installation of ROS Server

Currently, we use an Ubuntu 12.04 with Low Latency kernel and Ros Hydro. The server PC should have the IP

```
192.168.1.10
```

and the KUKA arm controllers should be reachable from the server.

Preparing KUKA controller

There are scripts in <kukie_home>/iis_scripts/krl. Copy the files <left/right>IISKukie.dat and <left/right>IISKukie.src to the KUKA controller. To run the software on the KRL side, execute the script. It will block at the line

```
wait for ($FRI_FRM_INT[1] == 101); Wait for a command
```

To use iis_kukie, you have to execute this script **before** you start the ROS side of iis_kukie.

Download from Git Repository

Authenticate at the IIS server and download the framework from the git repository (see git documentation). E.g:

For members using the iis infrastructure in our lab:

```
git clone --recursive git@git.uibk.ac.at:iis-software/robot_sw.git
```

For external users:

```
git clone --recursive https://git.uibk.ac.at/iis-software/robot_sw.git
```

Install dependencies

You can use the synaptic manager for this by

```
  sudo apt-get install synaptic
  sudo synaptic
```

Install the following packages (examples are provided below for ROS Groovy and ROS Hydro) (this list may not be complete, if you find some package that needs to be installed, please add it to the list):

ROS Groovy

```
  sudo apt-get install ros-groovy-cob-common ros-groovy-pr2 ros-groovy-ros-full ros-groovy-orocos-toolchain ros-groovy-rtt-ros-comm ros-groovy-ros-comm ros-groovy-geometry ros-groovy-common-msgs ros-groovy-common ros-groovy-rtt-geometry ros-groovy-moveit-full ros-groovy-moveit-full-pr2 libgsl0ldbl libgsl0-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev ros-groovy-qt-build libqwt6 libqwt-dev libsdl1.2-dev ros-groovy-moveit-full ros-groovy-cmake-modules ros-groovy-map-msgs ros-groovy-controller-manager python-catkin-tools
```

ROS Hydro

```
  sudo apt-get install ros-hydro-cob-common ros-hydro-ros-full ros-hydro-orocos-toolchain ros-hydro-rtt-ros-comm ros-hydro-ros-comm ros-hydro-geometry ros-hydro-common-msgs ros-hydro-rtt-geometry ros-hydro-moveit-full ros-hydro-moveit-full-pr2 ros-hydro-geometry-experimental libgsl0ldbl libgsl0-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev ros-hydro-qt-build libqwt6 libqwt-dev libsdl1.2-dev ros-hydro-moveit-full ros-hydro-cmake-modules ros-hydro-map-msgs ros-hydro-controller-manager python-catkin-tools
```

ROS Indigo

```
  sudo apt-get install ros-indigo-desktop-full ros-indigo-cob-common ros-indigo-ros-comm ros-indigo-geometry ros-indigo-common-msgs ros-indigo-control-msgs ros-indigo-geometry-experimental libgsl0ldbl libgsl0-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev ros-indigo-moveit-core ros-indigo-moveit-ros-planning ros-indigo-moveit-ros-planning-interface libghc-zlib-dev zlibc zlib1g-dbg zlib-bin ros-indigo-qt-build libqwt6 libqwt-dev libsdl1.2-dev ros-indigo-moveit-full ros-indigo-cmake-modules ros-indigo-map-msgs ros-indigo-controller-manager python-catkin-tools
```

ROS Kinetic

```
  sudo apt-get install ros-kinetic-desktop-full ros-kinetic-ros-comm ros-kinetic-geometry ros-kinetic-common-msgs ros-kinetic-control-msgs libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev ros-kinetic-moveit-core ros-kinetic-moveit-ros-planning ros-kinetic-moveit-ros-planning-interface libghc-zlib-dev zlibc zlib1g-dbg ros-kinetic-qt-build libqwt-dev libsdl1.2-dev ros-kinetic-cmake-modules ros-kinetic-map-msgs ros-kinetic-controller-manager cmake libgsl-dev libgsl2 libopencv-dev python-catkin-tools ros-kinetic-openni2-launch ros-kinetic-openni-launch
```

if you want to install the workspace under ros kinetic, you have to install moveit from source (http://moveit.ros.org/install/source/)

Update your ros database by

```
  sudo rosdep init
```

```
  rosdep update
```

Setup environment variables

Add an adapted version of the following example to your .bashrc file (<distribution> = groovy, hydro or indigo):

```
  source /opt/ros/<distribution>/setup.bash
  export ROS_HOSTNAME= <your hostname>
  export ROS_IP= <your ip>
  export ROS_MASTER_URI='<your ros server ip>:11311'
  export IIS_INCLUDE_PATH='<your path>/iis_robot_sw'
  export VREP_ROOT=$IIS_INCLUDE_PATH/external_libs/vrep
  export
ROS_PACKAGE_PATH=$ROS_PACKAGE_PATH:$IIS_INCLUDE_PATH/external_rosmake_ws:$II
S_INCLUDE_PATH/iis_rosmake_ws
  export
PYTHONPATH=$PYTHONPATH:$IIS_INCLUDE_PATH/external_libs/LeapSDK/lib:$IIS_INCL
UDE_PATH/external_libs/LeapSDK/libx64
  export VREP_PACKAGE_PATH=$IIS_INCLUDE_PATH/external_libs/vrep
```

Setup Environment via bashrc

```

ROSVERSION=melodic
WS_DIR=$HOME
WORKSPACE=iis_ws

#ROSVERSION=kinetic
#WORKSPACE=robot_sw/iis_catkin_ws

function getmyip() {
  MYIP=$(ip addr show wlan0 |  grep "inet " | awk '{print $2}' | sed
's%/.*$%%g')
  if [ -z $MYIP ] ; then
    MYIP=$(ip addr show eth0 |  grep "inet " | awk '{print $2}' | sed
's%/.*$%%g')
  fi
  if [ -z $MYIP ] ; then
    MYIP=127.0.0.1
  fi
  echo "using IP: $MYIP"
}

function ros_local() {
  getmyip
  source /opt/ros/$ROSVERSION/setup.bash
  export ROS_MASTER_URI=http://$MYIP:11311
  echo "set ROS_MASTER_URI=$ROS_MASTER_URI"
```

```
  export ROS_HOSTNAME=$MYIP
  export ROS_IP=$MYIP

  if [ "$1" != "" ]
  then
    WORKSPACE=$1
  fi

  echo "using workspace: $WORKSPACE"
  export IIS_INCLUDE_PATH=$WS_DIR/$WORKSPACE
  if [ -f $IIS_INCLUDE_PATH/devel/setup.bash ] ; then
    echo "sourcing $IIS_INCLUDE_PATH/devel/setup.bash"
    source $IIS_INCLUDE_PATH/devel/setup.bash
  else
    echo "No setup.bash file found on $IIS_INCLUDE_PATH/devel/setup.bash"
  fi

}

function ros_lab() {
  ros_local
  export ROS_MASTER_URI=http://192.168.64.104:11311
  echo "changed ROS_MASTER_URI=$ROS_MASTER_URI"
}

function ros_robotino() {
  ros_local
  export ROS_MASTER_URI=http://192.168.1.111:11311
  echo "changed ROS_MASTER_URI=$ROS_MASTER_URI"
}
```

Adapt `ROS_VERSION`, `WS_DIR`, `WS` to you system. `WS_DIR` can be used when all workspaces are stored in the same directory, e.g. `$HOME/Repos`. The `WS` variable specifies the default workspace to be used when no argument is specified. An argument can be specified for initializing a different workspace, e.g. `ROS_LOCAL robot_sw/catkin_iis_ws`.

Build

Call the make file in the root folder of the workspace by typing

```
  make
```

In some cases, this my fail several times, so just call it 2 to 3 times.

Usage

Arm

Description of the arm control topics can be found here.

Grippers

Schunk SDH2

- Schunk SDH Hand

Schunk PG70

- Schunk PG70

Pisa/IIT SoftHand

- Pisa/IIT Softhand

KIT Head

Description of the KIT Head can be found here

Simulator

deprecated Simulator documentation can be found at the simulator description page . The simulator provides topics according to the topics for the robotics hardware control used in the IIS labs. For information on these topics see the documentation of the hardware components (e.g. Arm documentation).

Error Handling

If you get the error:

```
  ... Unable to handle 'index' format version '2', please update rosdistro
...
```

Do:

```
  sudo apt-get install --only-upgrade python-rosdistro python-rosdep python-
rosinstall python-rosinstall-generator python-bloom
```

```
rosdep update
```