## STEP 1 : Install libfreenect2

Note: Ubuntu 12.04 is too old to support. Debian jessie may also be too old, and Debian stretch is implied in the following.

* Download libfreenect2 source

git clone https://github.com/OpenKinect/libfreenect2.git  
 cd libfreenect2

* (Ubuntu 14.04 only) Download upgrade deb files

cd depends; ./download\_debs\_trusty.sh

* Install build tools

sudo apt-get install build-essential cmake pkg-config

* Install libusb. The version must be >= 1.0.20.
  1. (Ubuntu 14.04 only) sudo dpkg -i debs/libusb\*deb
  2. (Other) sudo apt-get install libusb-1.0-0-dev
* Install TurboJPEG
  1. (Ubuntu 14.04 and newer) sudo apt-get install libturbojpeg libjpeg-turbo8-dev
  2. (Debian) sudo apt-get install libturbojpeg0-dev
* Install OpenGL
  1. (Ubuntu 14.04 only) sudo dpkg -i debs/libglfw3\*deb; sudo apt-get install -f
  2. (Other) sudo apt-get install libglfw3-dev
* Install OpenCL (optional)
  1. Intel GPU
     1. (Ubuntu 14.04 only) sudo apt-add-repository ppa:floe/beignet; sudo apt-get update; sudo apt-get install beignet-dev; sudo dpkg -i debs/ocl-icd\*deb
     2. (Other) sudo apt-get install beignet-dev
     3. For older kernels, # echo 0 >/sys/module/i915/parameters/enable\_cmd\_parser is needed. See more known issues at<https://www.freedesktop.org/wiki/Software/Beignet/>.
  2. AMD GPU: Install the latest version of the AMD Catalyst drivers from<https://support.amd.com> and apt-get install opencl-headers.
  3. Mali GPU (e.g. Odroid XU4): (with root) mkdir -p /etc/OpenCL/vendors; echo /usr/lib/arm-linux-gnueabihf/mali-egl/libmali.so >/etc/OpenCL/vendors/mali.icd; apt-get install opencl-headers.
  4. Verify: You can install clinfo to verify if you have correctly set up the OpenCL stack.
* Install CUDA (optional, Nvidia only):
  1. (Ubuntu 14.04 only) Download cuda-repo-ubuntu1404...\*.deb ("deb (network)") from Nvidia website, follow their installation instructions, including apt-get install cuda which installs Nvidia graphics driver.
  2. (Jetson TK1) It is preloaded.
  3. (Nvidia/Intel dual GPUs) After apt-get install cuda, use sudo prime-select intel to use Intel GPU for desktop.
  4. (Other) Follow Nvidia website's instructions.
* Install VAAPI (Intel only)
  1. (Ubuntu 14.04 only) sudo dpkg -i debs/{libva,i965}\*deb; sudo apt-get install -f
  2. (Other) sudo apt-get install libva-dev libjpeg-dev
  3. Linux kernels 4.1 to 4.3 have performance regression. Use 4.0 and earlier or 4.4 and later (Though Ubuntu kernel 4.2.0-28.33~14.04.1 has backported the fix).
* Install OpenNI2

1. (Ubuntu 14.04 only) sudo apt-add-repository ppa:deb-rob/ros-trusty && sudo apt-get update (You don't need this if you have ROS repos), then sudo apt-get install libopenni2-dev
2. (Other) sudo apt-get install libopenni2-dev

* Build

cd ..  
 mkdir build && cd build  
 cmake .. -DCMAKE\_INSTALL\_PREFIX=$HOME/freenect2  
 make  
 make install

* You need to specify cmake -Dfreenect2\_DIR=$HOME/freenect2/lib/cmake/freenect2 for CMake based third-party application to find libfreenect2.
* Set up udev rules for device access: sudo cp ../platform/linux/udev/90-kinect2.rules /etc/udev/rules.d/, then replug the Kinect.
* Run the test program: ./bin/Protonect
* Run OpenNI2 test (optional): sudo apt-get install openni2-utils && sudo make install-openni2 && NiViewer2. Environment variable LIBFREENECT2\_PIPELINE can be set to cl, cuda, etc to specify the pipeline.

## STEP 2 : Set up iai\_kinect2

1. Install the ROS. [Instructions for Ubuntu 14.04](http://wiki.ros.org/indigo/Installation/Ubuntu)
2. [Setup your ROS environment](http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment)
3. Install [libfreenect2](https://github.com/OpenKinect/libfreenect2):
4. Follow [the instructions](https://github.com/OpenKinect/libfreenect2#debianubuntu-1404) and enable C++11 by using cmake .. -DENABLE\_CXX11=ON instead of cmake ..
5. If something is not working, check out the latest stable release, for example git checkout v0.2.0.
6. Clone this repository into your catkin workspace, install the dependencies and build it:

cd ~/catkin\_ws/src/  
git clone https://github.com/code-iai/iai\_kinect2.git  
cd iai\_kinect2  
rosdep install -r --from-paths .  
cd ~/catkin\_ws  
catkin\_make -DCMAKE\_BUILD\_TYPE="Release"

*Note: rosdep will output errors on not being able to locate [kinect2\_bridge] and [depth\_registration]. That is fine because they are all part of the iai\_kinect2 package and rosdep does not know these packages.*

*Note: If you installed libfreenect2 somewhere else than in $HOME/freenect2 or a standard location like /usr/local you have to specify the path to it by adding -Dfreenect2\_DIR=path\_to\_freenect2/lib/cmake/freenect2 to catkin\_make.*

1. Connect your sensor and run kinect2\_bridge:

roslaunch kinect2\_bridge kinect2\_bridge.launch

1. Calibrate your sensor using the kinect2\_calibration. [Further details](https://github.com/code-iai/iai_kinect2/blob/master/kinect2_calibration#calibrating-the-kinect-one)
2. Add the calibration files to the kinect2\_bridge/data/<serialnumber> folder. [Further details](https://github.com/code-iai/iai_kinect2/blob/master/kinect2_bridge#first-steps)
3. Restart kinect2\_bridge and view the results using rosrun kinect2\_viewer kinect2\_viewer kinect2 sd cloud.

***[ERROR 1]in running catkin\_make command on Ubuntu 16.04:*** <https://github.com/code-iai/iai_kinect2/pull/450/commits/2cbd31c19e820ec266cae48b600e1d944bd512b9>

## STEP 3 : Setup rviz

1. Start roscore
2. Run

rosrun rviz rviz

1. Select kinect2\_bridge/link
2. To create Point Cloud
   1. Select Add from bottom tab and create Point Cloud 2
   2. Select whatever you you need to plot in Topic (kinect2\_bridge must be running)

***[ERROR 2]not getting plane to view in Rviz :***

~/catkin\_ws/src/iai\_kinect2/kinect\_bridge/launch/kinect\_bridge.launch

Line number 5 : default = “true”

### MY TERMINAL HISTORY WITH KINECT RUNNING AND DISPLAYING IN THE END:

**[new window]**

cd

git clone https://github.com/OpenKinect/libfreenect2.git  
cd libfreenect2

sudo apt-get install build-essential cmake pkg-config

sudo apt-get install libusb-1.0-0-dev

sudo apt-get install libturbojpeg libjpeg-turbo8-dev

sudo apt-get install libglfw3-dev

sudo apt-get install beignet-dev

sudo apt-get install libva-dev libjpeg-dev

sudo apt-get install libopenni2-dev

sudo apt-get install openni2-utils

cd ~/libfreenect2/

mkdir build && cd build

cmake .. -DCMAKE\_INSTALL\_PREFIX=$HOME/freenect2  
make  
make install

sudo cp ~/libfreenect2/platform/linux/udev/90-kinect2.rules /etc/udev/rules.d/

./libfreenect2/build/bin/Protonect

cd ~/catkin\_ws/src/  
git clone https://github.com/code-iai/iai\_kinect2.git  
cd iai\_kinect2

rosdep install -r --from-paths kinect2\_registration  
rosdep install -r --from-paths kinect2\_calibration

rosdep install -r --from-paths kinect2\_bridge

rosdep install -r --from-paths kinect2\_viewer

sudo subl ~/catkin\_ws/src/iai\_kinect2/kinect2\_registration/CMakeLists.txt [[Error 1]](#f0a6vhqovee1)  
cd ~/catkin\_ws  
catkin\_make -DCMAKE\_BUILD\_TYPE="Release"

cd

cd catkin\_ws/

source devel/setup.bash

roslaunch kinect2\_bridge kinect2\_bridge.launch

**[new tab]**

source devel/setup.bash

rosrun kinect2\_viewer kienct2\_viewer sd both

**[new tab]**

rosrun kinect2\_viewer kinect2\_viewer qhd image

**[all terminal windows closed]**

**[new window]**

subl ~/catkin\_ws/src/iai\_kinect2/kinect\_bridge/launch/kinect\_bridge.launch [[Error 2]](#bykwtj2pk9mp)

FINAL RUNNING THE KINECT

**[new window]**

roscore

**[new tab]**

cd ~/catkin\_ws

source devel/setup.bash

roslaunch kinect2\_bridge kinect2\_bridge.launch

**[new tab]**

source devel/setup.bash

rosrun kinect2\_viewer kinect2\_viewer sd both

**[new tab]**

rosrun rviz rviz