**Digital Signage OpenNI2 Driver Installation Instructions**

**V0.1.0**

**GIGA-BYTE TECHNOLOGY CO. LTD. Company:**

**Sign: Sign:**

**Date: Date:**

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rev** | **Modification Descriptions** | **Date** | **Editor** | **Approver** |
| 0.1.0 | Initial version  Release x64 Windows version only | April 11, 2019 | KKPoon |  |

# Table of Contents

[Revision History 2](#_Toc5181555)

[Table of Contents 3](#_Toc5181556)

[1. Voxel SDK 4](#_Toc5181557)

[1.1. Install Voxel SDK for x86 architecture Windows 4](#_Toc5181558)

[1.2. Install Voxel SDK for x64 architecture Windows 9](#_Toc5181559)

[1.3. Install Voxel SDK for i386 architecture Ubuntu 16.04 14](#_Toc5181560)

[1.4. Install Voxel SDK for amd64 architecture Ubuntu 16.04 14](#_Toc5181561)

[2. OpenNI2 Driver 15](#_Toc5181562)

[2.1. Run OpenNI2 Driver for x86 architecture Windows 15](#_Toc5181563)

[2.2. Run OpenNI2 Driver for x64 architecture Windows 16](#_Toc5181564)

[2.3. Run OpenNI2 Driver for i386 architecture Ubuntu 16.04 17](#_Toc5181565)

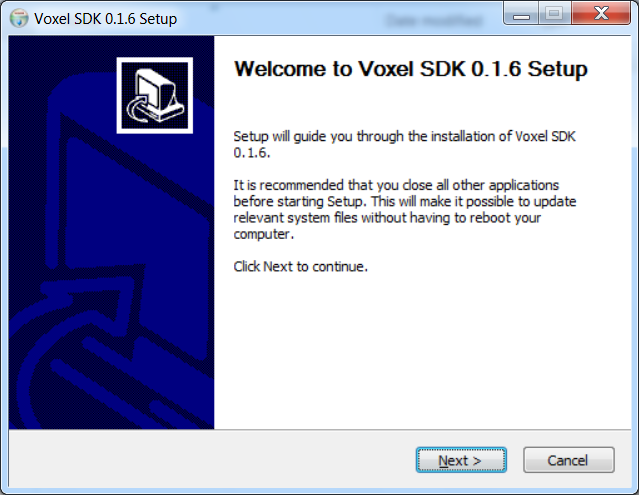
[2.4. Run OpenNI2 Driver for amd64 architecture Ubuntu 16.04 18](#_Toc5181566)

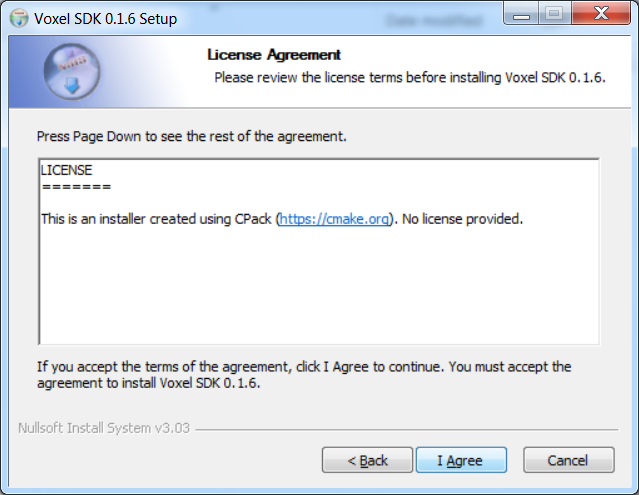
[2.5. Develop your own OpenNI2 application with GIGABYTE camera 19](#_Toc5181567)

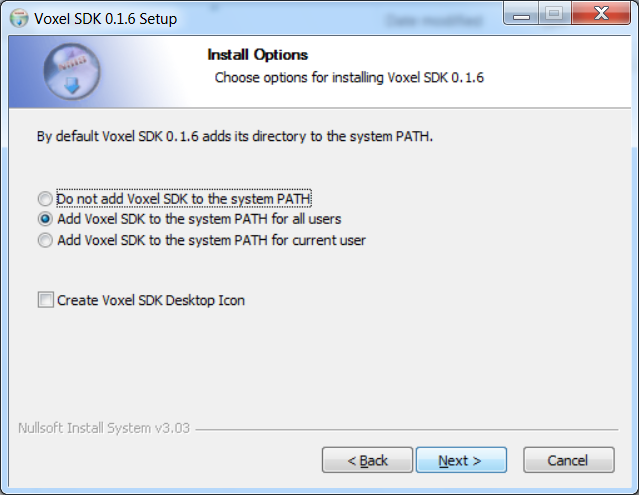
# Voxel SDK

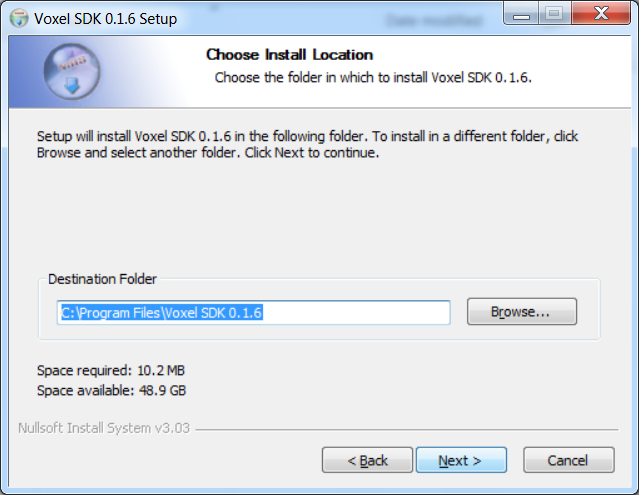
## Install Voxel SDK for x86 architecture Windows

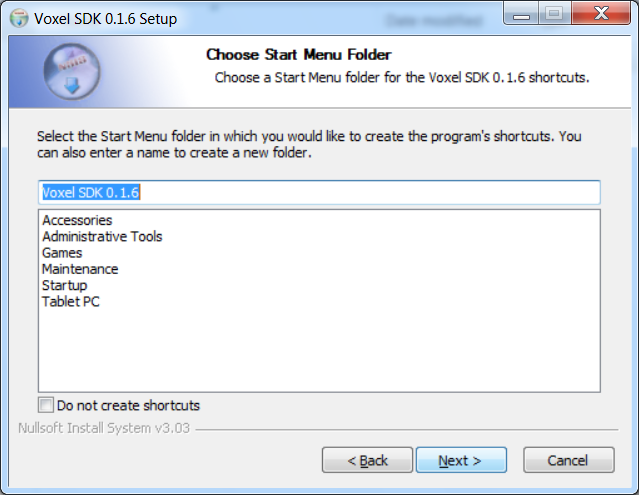
1. Run Windows/VoxelSDK/x86/SDK-x.x.x-win32.exe(where the “x.x.x” stands for version number)

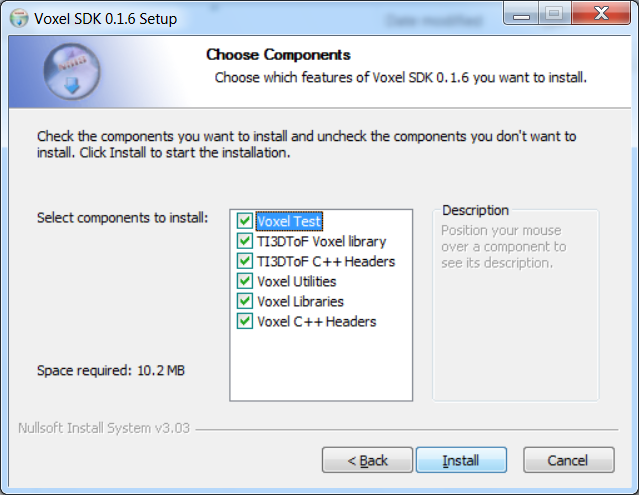




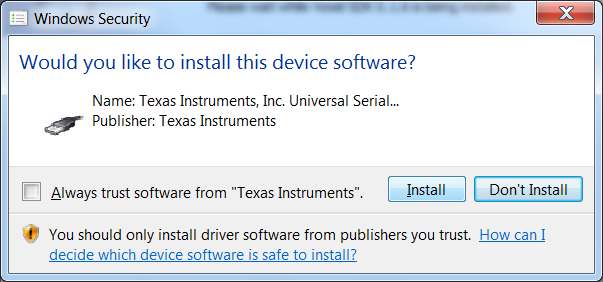


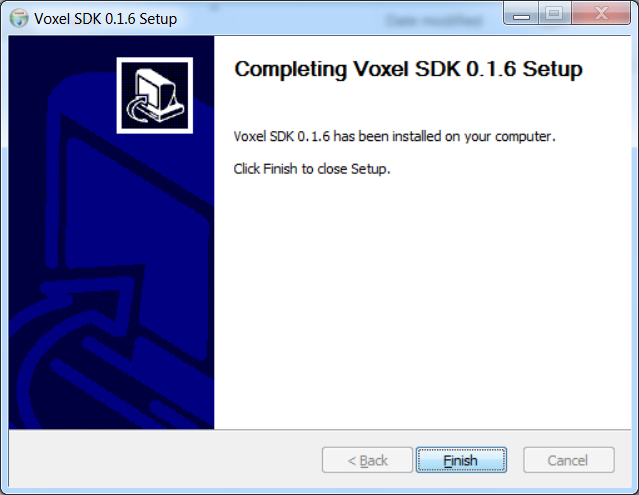








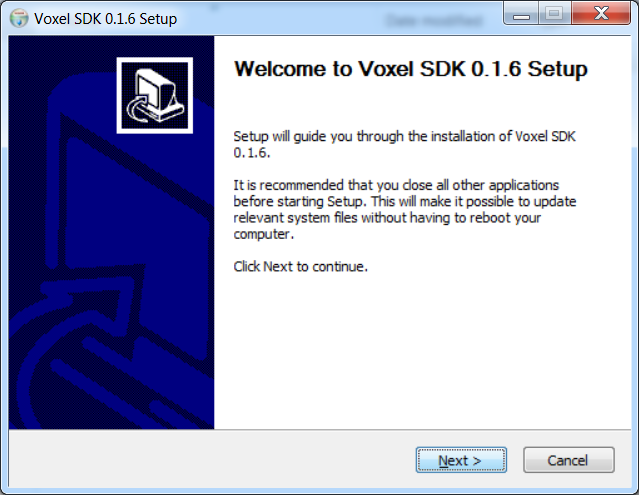


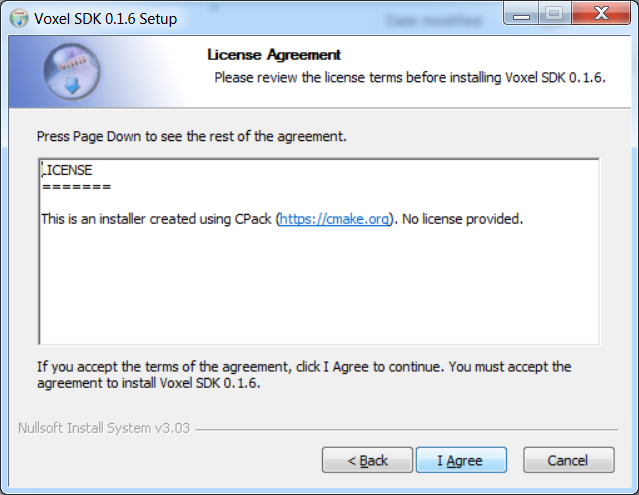


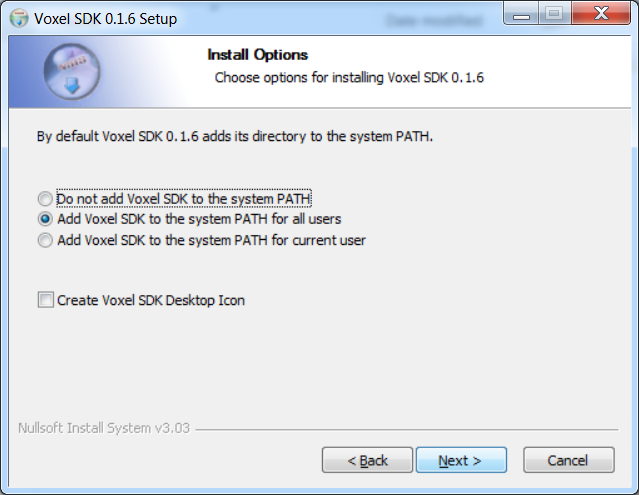
1. Reboot PC

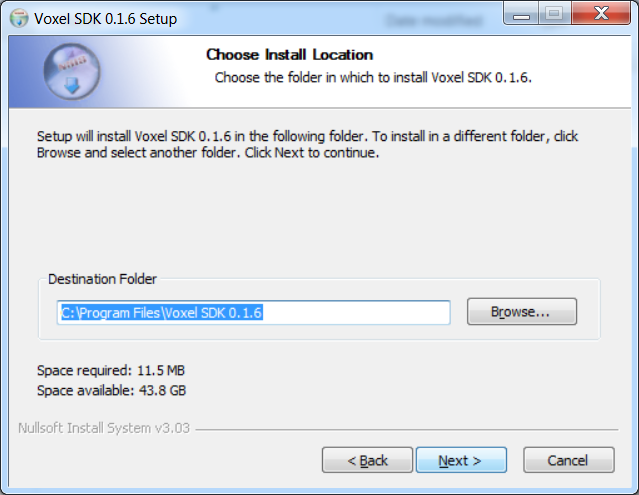
## Install Voxel SDK for x64 architecture Windows

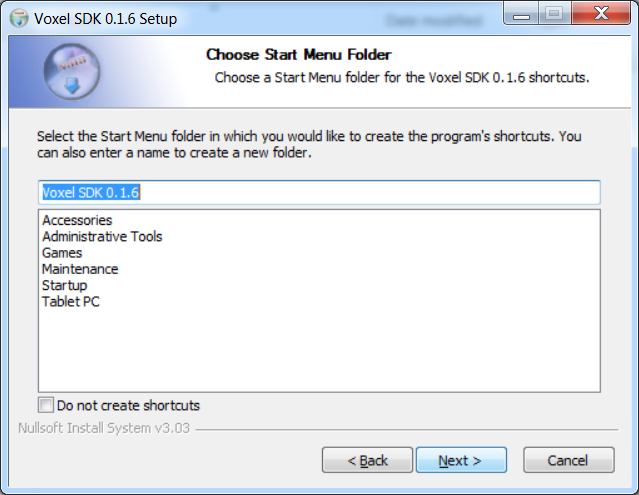
1. Run Windows/VoxelSDK/x64/SDK-x.x.x-win64.exe(where the “x.x.x” stands for version number)

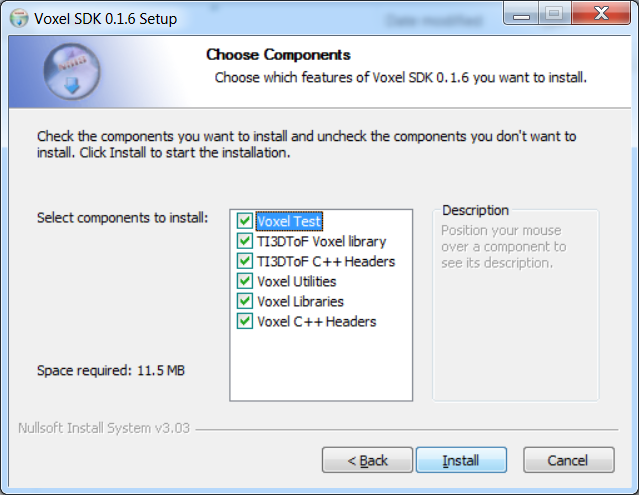




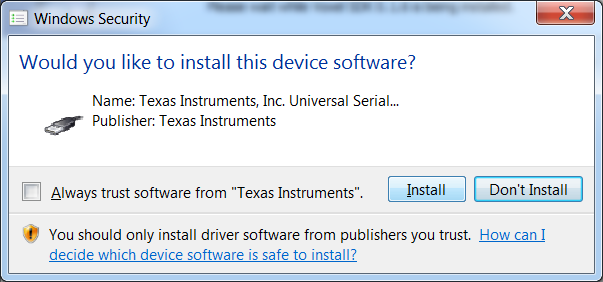


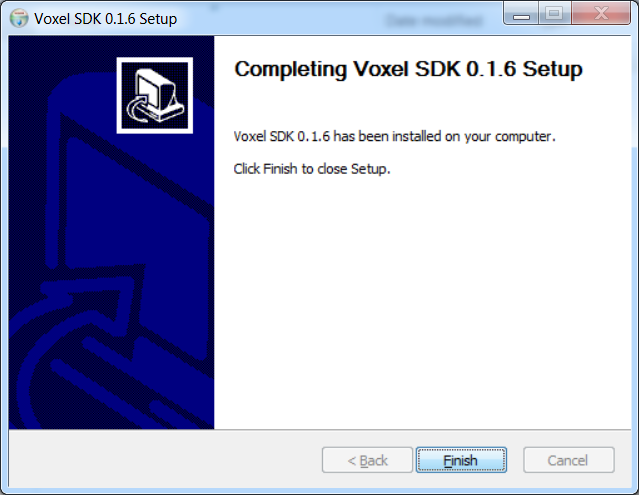












1. Reboot PC

## Install Voxel SDK for i386 architecture Ubuntu 16.04

1. Install the Debian packages(where the “x.x.x” stands for version number)
   * + - libti3dtof-x.x.x-i386-xenial.deb
       - libti3dtof-dev-x.x.x- i386-xenial.deb
       - libvoxel-x.x.x- i386-xenial.deb
       - libvoxel-dev-x.x.x- i386-xenial.deb
       - libvoxel-test-x.x.x- i386-xenial.deb

sudo apt-get install libusb-1.0-0-dev libudev-dev linux-libc-dev

cd Linux/VoxelSDK/x86

sudo dpkg -i \*.deb

* + - * libvoxel-util-x.x.x- i386-xenial.deb

## Install Voxel SDK for amd64 architecture Ubuntu 16.04

1. Install the Debian packages(where the “x.x.x” stands for version number)
   * + - libti3dtof-x.x.x-amd64-xenial.deb
       - libti3dtof-dev-x.x.x-amd64-xenial.deb
       - libvoxel-x.x.x-amd64-xenial.deb
       - libvoxel-dev-x.x.x-amd64-xenial.deb
       - libvoxel-test-x.x.x-amd64-xenial.deb

sudo apt-get install libusb-1.0-0-dev libudev-dev linux-libc-dev

cd Linux/VoxelSDK/x64

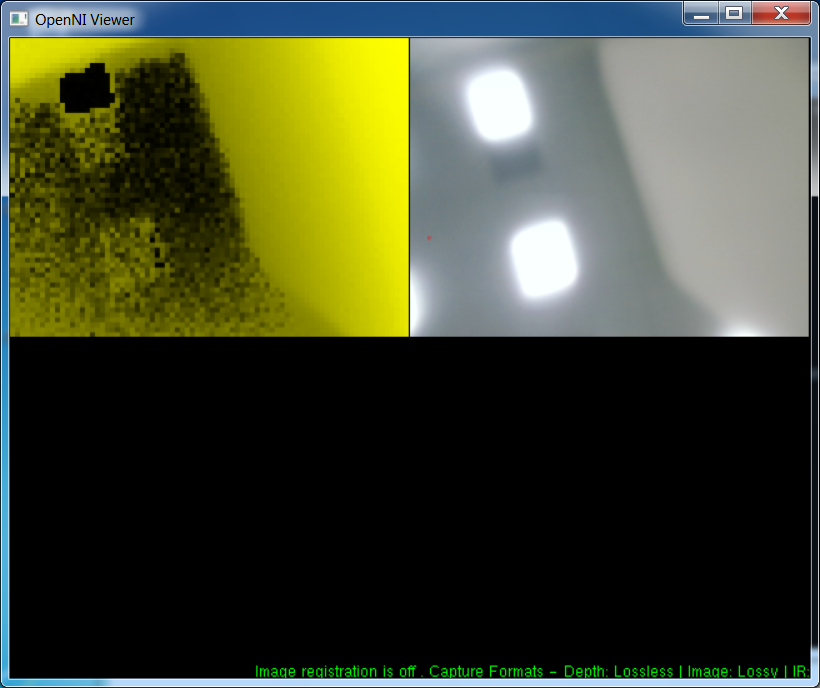
sudo dpkg -i \*.deb

* + - * libvoxel-util-x.x.x-amd64-xenial.deb

# OpenNI2 Driver

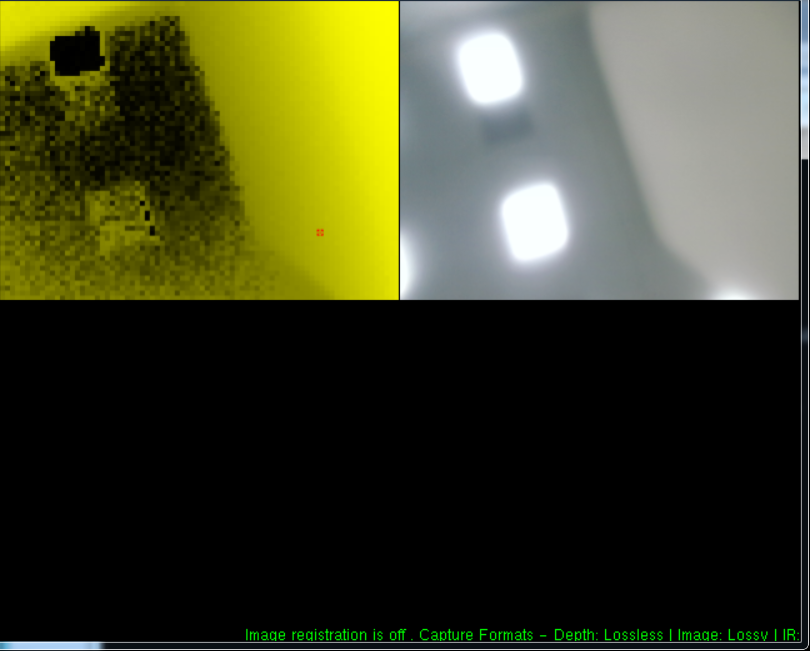
## Run OpenNI2 Driver for x86 architecture Windows

1. Run Windows/OpenNI2/x86/NiViewer.exe
2. If everything is correct, you can see the image from camera



## Run OpenNI2 Driver for x64 architecture Windows

1. Run Windows/OpenNI2/x64/NiViewer.exe
2. If everything is correct, you can see the image from camera



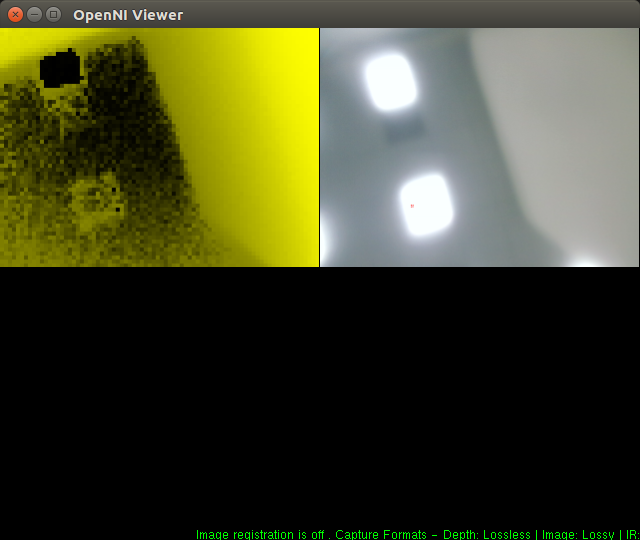
## Run OpenNI2 Driver for i386 architecture Ubuntu 16.04

sudo apt-get install freeglut3

cd Linux/OpenNI2/x86

./NiViewer

1. Run Linux/OpenNI2/x86/NiViewer
2. If everything is correct, you can see the image from camera



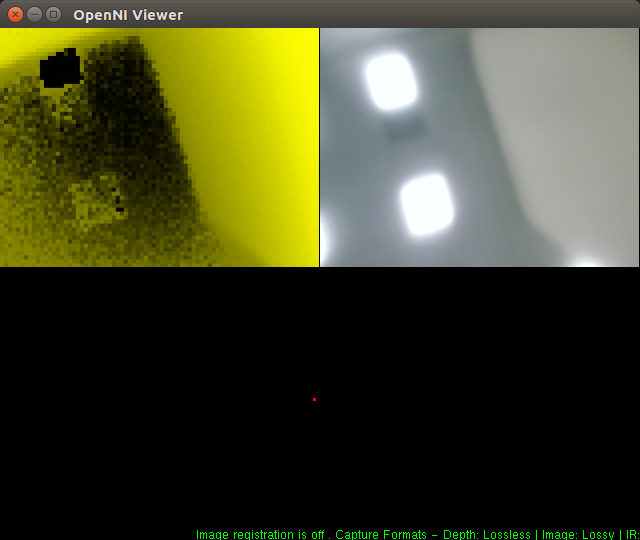
## Run OpenNI2 Driver for amd64 architecture Ubuntu 16.04

sudo apt-get install freeglut3

cd Linux/OpenNI2/x64

./NiViewer

1. Run Linux/OpenNI2/x64/NiViewer
2. If everything is correct, you can see the image from camera



## Develop your own OpenNI2 application with GIGABYTE camera

1. Download and install [OpenNI2 SDK](https://structure.io/openni)
2. Setup OpenNI2 development environment to your application
3. Copy GB\_DepthCamera.dll or libGB\_DepthCamera.so to “/OpenNI2/Drivers/” folder at directory of your application depend on your platform
4. Run your application