

Revised record

Revised date	version number	revision note
2018/7/18	V0.1	Initial test version
2018/7/19	V0.2	Increase in revised records, etc.
2018/7/26	V0.3	Supplementary android interface
2018/7/26	V0.4	Collate document format
2018/8/14	V0.5	Add synchronization and alignment interface instructions
2018/9/07	V0.6	Increase Java layer synchronization, alignment, and set resolution instructions
2018/9/18	V0.7	Add a new synchronization pattern description
2018/11/16	V0.8	Add DSP interface settings, mirror images, flip configuration instructions
2018/11/22	V0.9	Modify DSP interface settings
2018/11/22	V1.0	Add image alignment mode description
2018/12/12	V1.1	Added description of VideoMode and VideoOpRef
2018/12/13	V1.2	Increase the second way to open device permissions

1. Applicable scene

This guide to the purchase of the company's products for secondary application development technology and maintenance personnel.

2. SDK install

SDK currently supports the Windows, Linux, Android, Arm platform, and the recommended configuration is as follows

A. operating system (OS)

Windows:

Windows 7, 8, 10 on x86 (32/64 bit)

Ubuntu:

Ubuntu 12.04 (32/64/arm) And above

Android:

Android4.0 above

B. processor

Pentium 4, 1.4GHz And above

AMD Athlon 64/FX 1GHz And above

Arm Cortex A8 And above

C. internal storage

greater than 512MB

D. interface

USB2.0

E. exploitation environment

VS2010, VS2015, Eclipse, AndroidStudio

F. video card

Some sample programs need to be higher than ATI RADEON x1300 or NVIDIA GeForce 7300

2.1. windows platform

2.1.1. Installation package type

32-bit installation package

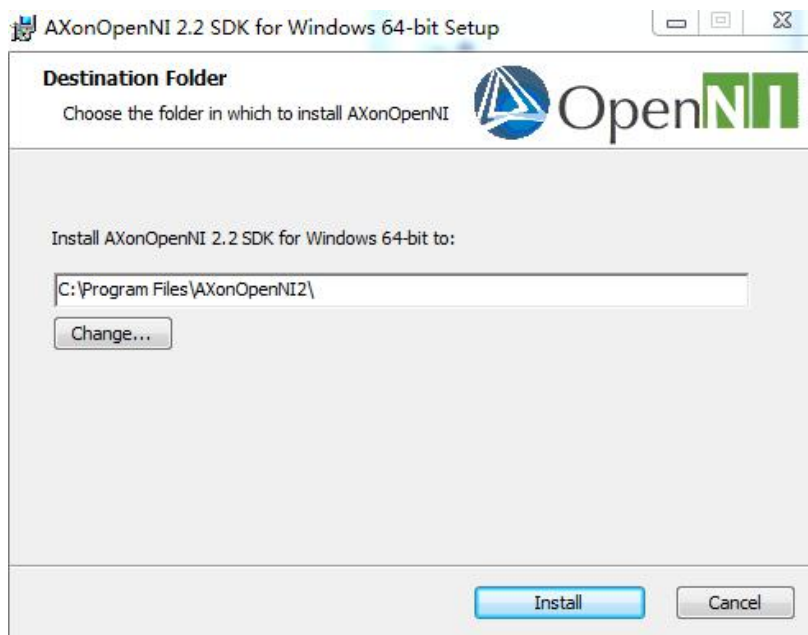
AXonLink-OpenNI-Windows-x86-2.2.msi

64-bit installation package

AXonLink-OpenNI-Windows-x64-2.2.msi

2.1.2. Installation step

- Open the installation package and select the installation path

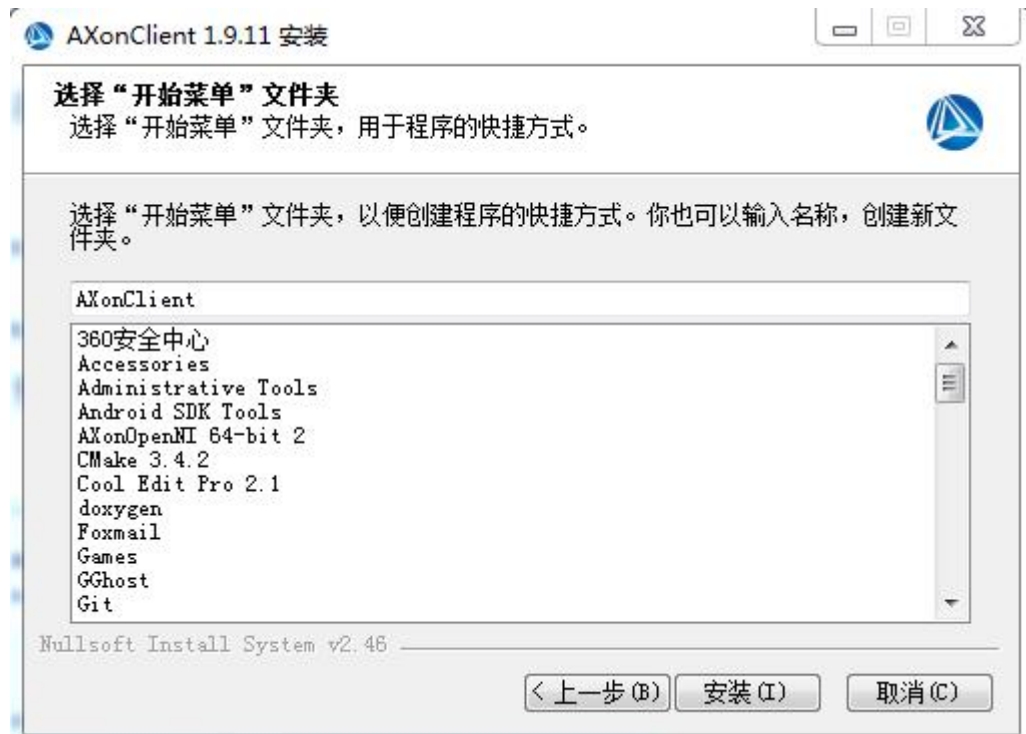


- Installation tips for the client pop up during installation



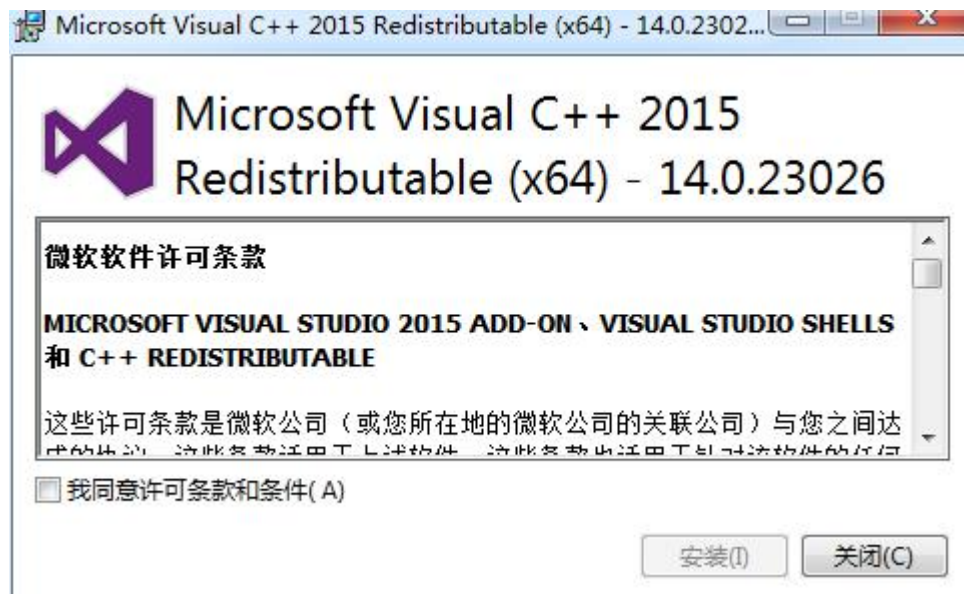
- Click next and configure the installation path.



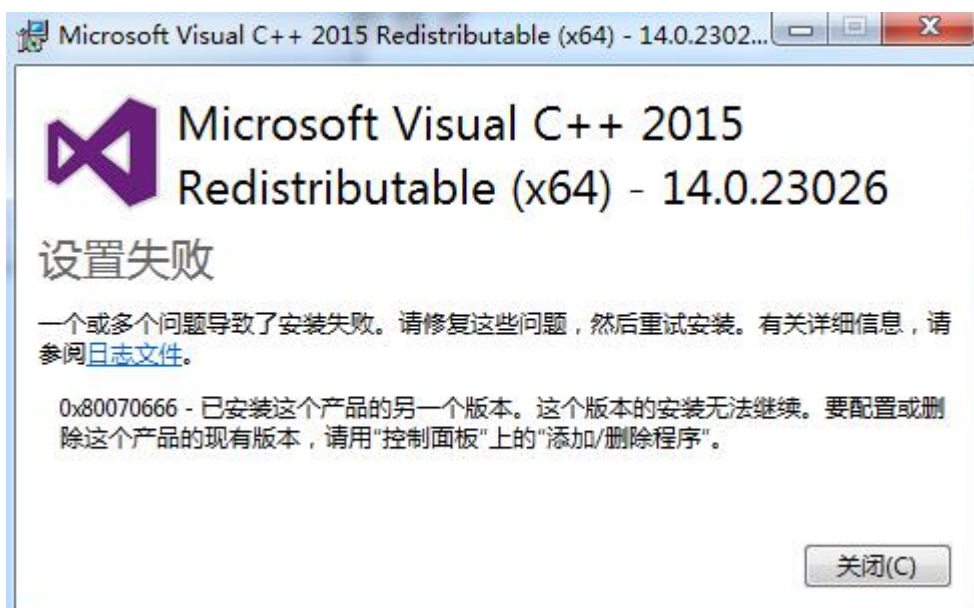


- The VC 2015 installation appears during installation, and click Yes





If the installation will pop up and fail, click close to continue the installation; otherwise, the normal installation



- Continue to install device drivers



- After the installation is complete to connect the device, you can check my computer-> device Manager to see if the device is correctly identified and correctly identified to the device as shown in the following figure



2.2. Linux platform

Take the Hunting 16.04 as an example

2.2.1. Installation package name format

安装包名称: AXonOpenNI-Linux-x64-2.2.tar.bz2

2.2.2. Installation process

1. Install the required drive

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

2. Install graphical window

```
sudo apt-get install build-essential freeglut3 freeglut3-dev
```

3. Installation package decompression

```
tar -jxvf AXonOpenNI-Linux-x64-2.2.tar.bz2
```

4. Execute installation script

```
sudo chmod +x install.sh
```

```
sudo ./install.sh
```

5. Set environment variable

```
source OpenNIDevEnvironment
```

6. Compile sample files(eg: AXon)

```
cd Samples/AXon
```

```
make
```

7. Run the executable program.

```
cd Bin/x64-Release
```

```
sudo ./AXon
```

2.3. Android platform

2.3.1. Installation package name

Installation package name: OpenNI-Android-2.3.tar

2.3.2. Installation process

Unzip the installation package and install the sample apk in the installation package to the android device to run.

Note:

The normal operation of the program requires the Android device to gain root permissions and turn off SELinux.

2.4. Arm Platform

Customize according to demand.

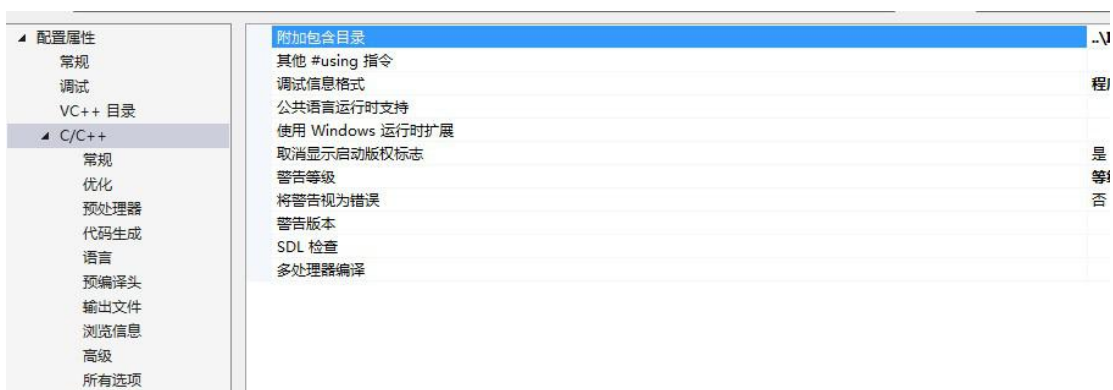
3. SDK Instructions for use

3.1. windows Platform

Windows Directory structure under the platform

AXonClient	2018/7/9 18:53	文件夹	
Documentation	2018/7/9 18:53	文件夹	
Driver	2018/7/9 18:53	文件夹	
Include	2018/7/9 18:53	文件夹	
Lib	2018/7/9 18:53	文件夹	
Redist	2018/7/9 18:53	文件夹	
Samples	2018/7/9 18:53	文件夹	
Tools	2018/7/9 18:53	文件夹	
LICENSE	2018/7/9 15:56	文件	12 KB
NOTICE	2018/7/9 15:56	文件	1 KB

- AXonClient : installation package for client under windows;
- Documentation : OpenNI's standard interface guide, which operates the device via the OpenNI interface. ;
- Driver : Equipment Drive Installation Procedure. ;
- Include : OpenNI related header files, AXonLink.h for some custom data structures and enumerations;
- Lib : OpenNI2.lib;
- Redist : Dynamic link library, the application can run only if the files under Redist are copied to the directory where the application is located. When developing an application, you need to include OpenNI2.lib and phase key files; take VS2015 as an example, include Include in the C / C++ additional inclusion directory

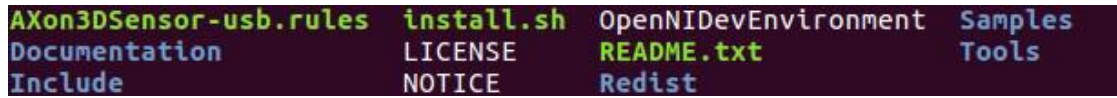


- Add OpenNI2.lib; to the linker-> General-> attach the path where the OpenNI2.lib is located in the inclusion directory; add OpenNI2.lib; to the linker-> input-> add OpenNI2.lib; to the program to add # include < AXonLink.h > and # include < OpenNI.h >.

- **Samples** : A number of sample projects have been included in the directory, and the developer can make reference;
- **Tools** : Run NiViewer.exe in the directory to preview the video stream obtained by the camera;

3.2. Linux Platform

directory structure



- Documentation : Help documentation for Openni;
- Samples : Example project, make can run Makefile that builds a new project using SDK to write with reference to Makefile in the sample example. Linux can also run the Niviewer program, sudo. / NiViewer under Tools

3.3. Android platform

directory structure

	Assets	2018/7/17 20:22	文件夹	
	Documentation	2018/7/17 20:22	文件夹	
	Native	2018/7/17 20:22	文件夹	
	OpenNIAndroidLibrary	2018/7/17 20:25	文件夹	
	Samples	2018/7/17 20:22	文件夹	
	Tools	2018/7/17 20:22	文件夹	
	CHANGES.txt	2018/7/17 20:22	文本文档	11 KB
	LICENSE	2018/7/17 20:22	文件	12 KB
	NOTICE	2018/7/17 20:22	文件	31 KB
	ReleaseNotes.txt	2018/7/17 20:22	文本文档	2 KB

- Assets : Related configuration file;
- Documentation : C and java use documentation, for the development of android applications, please refer to the documentation under java, where getting-started.html guides how to debug the sample project in a development kit;
- Native : Header and library files;
- OpenNIAndroidLibrary : The project of generating jar package with library file;
- Sample : Three application examples of the project, the

prebuild subdirectory contains the project generated. Apk file, can be installed directly on the android device;

- Tools : Niviewer engineering source code and installable apk files, the above projects are available

eclipse Open directly.

3.3.1. Project introduction

Eclipse contains SimpleRead, SimpleWrite, NiViewer projects that exist in Sample- > eclipseProject centre;

Android contains only NiViewer projects and exists in Sample- > androidStudioProject.

4. SDK Interface introduction

The SDK supports most of the interfaces for OpenNI2, and some of the custom parameters can be passed through the property interface of OpenNI2 and

The invoke interface is obtained, and Android does not support the invoke command for the time being.

4.1. Open the device to create the stream and get the stream data information

The device opens to create a stream. The method of obtaining the flow can be operated according to the OpenNI standard process. The OpenNI help document is in the Document folder of the installation directory. .

The usage is as follows:

```
#include <stdio.h>

#include <OpenNI.h>

#include "OniSampleUtilities.h"

#define SAMPLE_READ_WAIT_TIMEOUT 2000 //2000ms

using namespace openni;

int main(int argc, char* argv[])

{

    Status rc = OpenNI::initialize(); /*初始化 OpenNI*/
```



```
if (rc != STATUS_OK)
{
    printf("Initialize failed\n%s\n", OpenNI::getExtendedError());
}
```

```
        return 1;
    }

    Device device;

    if (argc < 2)
        rc = device.open(ANY_DEVICE);    /*打开任意可识别*/
    else
        rc = device.open(argv[1]);

    if (rc != STATUS_OK)
    {
        printf("Couldn't open device\n%s\n", OpenNI::getExtendedError());
        return 2;
    }

    OniVersion driver;
    int nsize;
    nsize = sizeof(driver);

    device.getProperty(ONI_DEVICE_PROPERTY_DRIVER_VERSION, &driver, &nsize); /*获取驱动版本号*/

    printf("AXon  driver  version  V%d.%d.%d.%d\n",  driver.major,  driver.minor,
driver.maintenance, driver.build);

    VideoStream depth;

    if (device.getSensorInfo(SENSOR_DEPTH) != NULL)
    {
        rc = depth.create(device, SENSOR_DEPTH); /*有深度 Sensor 则创建深度流*/

        if (rc != STATUS_OK)
```

```
{
    printf("Couldn't create depth stream\n%s\n", OpenNI::getExtendedError());
    return 3;
}
}

rc = depth.start(); /*Turn on depth flow*/
if (rc != STATUS_OK)
{
    printf("Couldn't start the depth stream\n%s\n", OpenNI::getExtendedError());
    return 4;
}

VideoFrameRef frame;

while (!wasKeyboardHit())
{
    int changedStreamDummy;

    VideoStream* pStream = &depth;

    rc = OpenNI::waitForAnyStream(&pStream, 1, &changedStreamDummy,
SAMPLE_READ_WAIT_TIMEOUT); /*Waiting flow*/

    if (rc != STATUS_OK)
    {
        printf("Wait failed! (timeout is %d ms)\n%s\n", SAMPLE_READ_WAIT_TIMEOUT,
OpenNI::getExtendedError());

        continue;
    }
}
```

```
rc = depth.readFrame(&frame); /*Read deep stream frame data*/

if (rc != STATUS_OK)

{
    printf("Read failed!\n%s\n", OpenNI::getExtendedError());
    continue;
}

if (frame.getVideoMode().getPixelFormat() != PIXEL_FORMAT_DEPTH_1_MM &&
    frame.getVideoMode().getPixelFormat() != PIXEL_FORMAT_DEPTH_100_UM &&
    frame.getVideoMode().getPixelFormat() != PIXEL_FORMAT_DEPTH_1_3_MM)
{
    printf("Unexpected frame format\n");
    continue;
}

DepthPixel* pDepth = (DepthPixel*)frame.getData(); /*Depth data of the current
frame*/

int middleIndex = (frame.getHeight()+1)*frame.getWidth()/2;

/*Print the timestamp and center deep value of the current frame*/
printf("[%08llu]%8d\n", (long long)frame.getTimestamp(), pDepth[middleIndex]);
}

depth.stop(); /*Stop deep flow*/
depth.destroy(); /*Destroy deep flow*/
device.close(); /*closing device*/
OpenNI::shutdown(); /*turn off openni*/

return 0;
```

```
}
```

4.2. VideoMode explain

VideoMode represents the format of the data stream, including image width, height, pixel format and frame rate. All VideoMode supported by a device sensor (Color/IR/Depth) can be obtained through `SensorInfo::getSupportedVideoModes()`:

C++

```
const SensorInfo* colorSensorInfoP =  
    device.getSensorInfo(SENSOR_COLOR);  
  
if(colorSensorInfoP)  
{  
    const Array<VideoMode>& allVMs =  
        colorSensorInfoP-  
            >getSupportedVideoModes();  
    .....  
}
```

Java

```
if (device.getSensorInfo(SensorType.COLOR) != null) {  
    List<VideoMode> supportedModes =  
        mVideoStream.getSensorInfo().getSupportedVideoModes();  
}
```

You can modify VideoMode. with `VideoStream::setVideoMode()` For example, the following code sets the Color image to jpeg format:

C++

```
VideoMode vm = ColorStream.getVideoMode();  
vm.setPixelFormat(PIXEL_FORMAT_JPEG);  
ColorStream.setVideoMode(vm);
```

Java

```
VideoMode vm = ColorStream.getVideoMode();

vm.setPixelFormat(PixelFormat.JPEG);

ColorStream.setVideoMode(vm);
```

4.3. VideoFrameRef explain

User programs usually get a VideoFrameRef through VideoStream::readFrame. A VideoFrameRef object represents a data frame, which can be several common methods of Color, IR or Depth. VideoFrameRef:

getWidth	Gets the width of the image (pixel points)
getHeight	Get the height of the image (pixel points)
getData	Get image data
getDataSize	Gets the size of the image data (bytes)

The format of the different image types is different from the data obtained by getData:

- **Depth:** The data type is uint16_t, and the data is arranged in the order of the first row and the last column. A uint16_t represents the depth value of a point. For the unit of depth value, see the unit for obtaining depth data. .
- **IR:** The data type is uint8_t, and the data is arranged in the order of the first and last columns. A uint8_t represents the IR brightness of a point, and a larger value indicates a brighter.
- **Color:**
 - If the pixelFormat in the VideoMode of ColorStream is PIXEL_FORMAT_RGB888, then the data type is RGB888Pixel. A RGB888Pixel_t represents the RGB value of a point. For example, in opencv, you can convert ColorFrame to cv::Mat in the following ways:

C++

```
cv::Mat im_RGB(Colorframe.getHeight(), Colorframe.getWidth(), CV_8UC3,
(void*)Colorframe.getData());
```

Java

In the Java version, `Colorframe.getData ()` is in `bytebuffer` format, and the rest is similar to C, and users can use it according to

pay attention to that the pixel data order in Mat is RGB at this time.

- If the pixelFormat in ColorStream's VideoMode is PIXEL_FORMAT_JPEG, then the data type is jpeg picture data. For example, in opencv, you can convert ColorFrame to cv::Mat in the following ways:

C++

```
cv::Mat im_RGB =
cv::imdecode(std::vector<uchar>((uchar*)Colorframe.getData(),
(uchar*)Colorframe.getData()+Colorframe.getDataSize()),
CV_LOAD_IMAGE_COLOR);
```

Java

Java 版本中 Colorframe.getData()为 bytebuffer 格式，其余和 C++相似，用户可根据条件自行适配至 opencv。

pay attention to,at this time, the pixel data order in Mat is BGR.

4.4. A unit of depth data.

As shown in opening the device to create a stream, get the code in the stream data information,

frame.getVideoMode().getPixelFormat()The resulting value represents the unit of depth data; see the code comments at the variable definition for specific instructions.

You can also get the unit of depth value through
openni::OpenNI::getDepthValueUnit_mm.

Some of the supported values and units represented are as follows:

C++	Java	VALUE
PIXEL_FORMAT_DEPTH_1_MM ONI_PIXEL_FORMAT_DEPTH_1_MM	DEPTH_1_MM	1 毫米
PIXEL_FORMAT_DEPTH_100_UM ONI_PIXEL_FORMAT_DEPTH_100_UM	DEPTH_100_UM	100 微米(0.1 毫米)
PIXEL_FORMAT_DEPTH_1_3_MM ONI_PIXEL_FORMAT_DEPTH_1_3_MM	DEPTH_1_3_MM	1/3 毫米(0.33 毫米)
PIXEL_FORMAT_DEPTH_1_2_MM ONI_PIXEL_FORMAT_DEPTH_1_2_MM	DEPTH_1_2_MM	1/2 毫米(0.50 毫米)

4.5. Gets the firmware version number of the device

C++

```
openni::Status status = STATUS_OK;

AXonLinkFWVersion version;

int dataSize = (int)sizeof(AXonLinkFWVersion);

status=device->getProperty(ONI_DEVICE_PROPERTY_FIRMWARE_VERSION,&version,
&dataSize);
```

Java

```
FWVersion fwversion = null;

fwversion = device.getFWVersion();
```

4.6. Gets the version number of the firmware mirror file.

C++

```
// Assume that the path saved by the firmware under windows is

char filePath[256] = "E:\\resource";

filePath .Replace(_T("\\"), _T("\\\\"));

int filepathSize = sizeof(filePath );

AxonLinkFirmWarePacketVersion packetVersion;

memcpy(packetVersion.filename, filePath ,filepathSize);

device->invoke(AXONLINK_DEVICE_INVOKE_GET_FWVERSION,&packetVersion,sizeof(packet
Version);
```

Java

```
/// Android system, put the upgrade file update.img in the Download folder, and pay
attention to adding file read and write permission extension when developing

request
```

```
File sdCard = Environment.getExternalStorageDirectory();
```

```
File directory_download = new File(sdCard, "Download");

String filePath = directory_download.getAbsolutePath()+"/update.img";

FirmWarePacketVersion          fwpacketVersion          =
device.getFirmWarePacketVersion(filePath);
```

4.7. Firmware upgrade

The firmware version number of the device can be compared with the currently mirrored version number to determine whether an upgrade is required.

➤ Set up upgrade files

C++

```
openni::Status rc = STATUS_OK;

rc = device->invoke(AXONLINK_DEVICE_INVOKE_SET_UPLOADFILE, ( char*)FileName,
filepathSize);
```

Java

```
device.SetUpgradeFile(FileName);
```

➤ Start upgrading after device file success

C++

```
uint8_t enable =1;

rc=device->invoke(AXONLINK_DEVICE_INVOKE_SET_UPGRADE_ENABLE,&enable,sizeof(uint8
_t));
```

Java

```
device.UpgradeEnable(true);
```

➤ Gets the current upgrade status

Set a timer, it is recommended to query the current upgrade status every 500ms.

C++

```
AXonLinkSendFileStatus m_upgradeStatus = AXON_LINK_SENDFILE_STATUS_STOP;
```

```
device->invoke(AXONLINK_DEVICE_INVOKE_GET_UPGRADE_STATUS,&m_upgradeStatus,sizeof
(m_upgradeStatus));
```

Java

```
UpgradeStatus status = device.getUpgradeStatus();
```

```
Log.i(TAG,status.toString());
```

➤ End upgrade

If the state is AXON_LINK_SENDFILE_STATUS_SUCCESS or AXON_LINK_SENDFILE_STATUS_FAILED, you need to

Close the upgrade or cause memory leaks.

C++

```
uint8_t enable =0;
```

```
rc=device->invoke(AXONLINK_DEVICE_INVOKE_SET_UPGRADE_ENABLE,&enable,sizeof(uint8
_t));
```

Java

```
device.UpgradeEnable(false);
```

4.8. Restart equipment

C++

```
device->invoke(AXONLINK_DEVICE_INVOKE_SET_REBOOT, 0, 0);
```

Java

```
device.reboot();
```

4.9. Get the camera inside and outside the parameter

C++

```
AXonLinkCamParam camParam;
```

```
int dataSize = sizeof(AXonLinkCamParam);

rc=device->getProperty(AXONLINK_DEVICE_PROPERTY_GET_CAMERA_PARAMETERS,
&camParam,&dataSize);
```

Java

```
CamParam camparam = null;

camparam = mDevice.getCamParam();
```

4.10. Get device serial number

C++

```
char serial[AXON_LINK_SERIALNUMBER_SIZE];

memset(serial, 0, AXON_LINK_SERIALNUMBER_SIZE);

int nsize = AXON_LINK_SERIALNUMBER_SIZE;

rc = device->getProperty(ONI_DEVICE_PROPERTY_SERIAL_NUMBER, serial, &nsize);
```

Java

```
String serialnumber = device.getDeviceSerialNumber();
```

4.11. Set resolution

C++

```
//Set the resolution of Color to the same as depth

VideoStream depthStream;

VideoStream colorStream;

...

const VideoMode depthVideoMode = depthStream.getVideoMode();

VideoMode colorVideoMode = colorStream.getVideoMode();
```

```
colorVideoMode.setResolution(depthVideoMode.getResolutionX(),
depthVideoMode.getResolutionY());

colorStream.setVideoMode(colorVideoMode);
```

Java

```
//Set the resolution of Color to the same as depth

VideoStream depthStream;

VideoStream colorStream;

...

const VideoMode depthVideoMode = depthStream.getVideoMode();

VideoMode colorVideoMode = colorStream.getVideoMode();

colorVideoMode.setResolution(depthVideoMode.getResolutionX(),
depthVideoMode.getResolutionY());

colorStream.setVideoMode(colorVideoMode);
```

4.12. Set up RGB and Depth synchronization

USB transmission and data processing may cause the received RGB and Depth data to be out of sync in time. You can call the synchronization interface to achieve synchronization.

C++

```
rc = device.setDepthColorSyncEnabled(true); //True indicates synchronization,
false indicates non-synchronization.
```

Java

```
device.setDepthColorSyncEnabled(true)
```

The synchronization algorithm has two parameters to set in the same directory as AXonLink. dll or AXonLink. so. :

SyncThresholdTime: If the time difference between the two frame images of the different streams is greater than this value, then the two frames are considered to be non-synchronized and are buffered to wait for a new frame.

Otherwise, it is considered to be synchronous and will be sent to the upper layer.

SyncMaxFrameNum: The number of frames in the cache is equivalent to the maximum length of time in the cache. For example, if the frame rate is 30FPS, **SyncMaxFrameNum=5**, Then the sync window is $1000/30*5=165\text{ms}$.

4.13. Set RGB and Depth alignment

First, set the same resolution for RGB and Depth. See setting Resolution. Then call the alignment interface. Note

Meaning: Setting the alignment increases the CPU resource consumption at the host. The currently supported synchronization pattern is shown in the table below. Note that partial mode requires that RGB and Depth synchronization be set first.

Code example

C++

```
device.setImageRegistrationMode(IMAGE_REGISTRATION_DEPTH_TO_COLOR);//对齐  
device.setImageRegistrationMode(IMAGE_REGISTRATION_OFF);//不对齐
```

Java

```
device.setImageRegistrationMode(ImageRegistrationMode.DEPTH_TO_COLOR)

device.setImageRegistrationMode(ImageRegistrationMode.OFF)
```

4.14. DSP interface settings

DSP interface settings generally do not need to be modified.

The example configuration "off-flying pixel", "de-flying pixel 2", "3D noise reduction", can be set as needed. Other parameters are not recommended for modification.

Description of the "de-flying pixel" and the "the de-flight pixel 2":

1. "fly Pixel 2" works better, but consumes host CPU resources. By default, it is closed.
2. "to fly pixels" is less effective, but does not consume CPU resources. The default is open.

C++

```
//Gets the current state
AXonDSPInterface m_dsp_process_interface;

device.invoke(AXONLINK_DEVICE_INVOKE_GET_DSP_DPINTERFACE,&m_dsp_process_interface,
sizeof(m_dsp_process_interface)) ;

// 0 means closed, 1 means open

m_dsp_process_interface. R2Z = 1;           // Turn on 3D noise reduction

m_dsp_process_interface. FLYING = 1;        // Open the fly pixel

m_dsp_process_interface. FLYING_2 = 1;      // Open the fly pixel 2

device.invoke(AXONLINK_DEVICE_INVOKE_SET_DSP_DPINTERFACE,&m_dsp_process_interface,
sizeof(m_dsp_process_interface)) ;
```

Java

```
//配置 3D 降噪

int status = device.getDSPInterfaceStatus();           //Gets the current
interface status

// setDSPInterfaceStatus 2nd parameter, 1 open 0 close

device.setDSPInterfaceStatus(AXonDSPInterface.DNR3, 1); // Turn on 3D noise
reduction

device.setDSPInterfaceStatus(AXonDSPInterface.FLYPIX, 1); // Open to fly pixels

device.setDSPInterfaceStatus(AXonDSPInterface. FLYING_2, 1); // Open the fly pixel 2
```

4.15. Set up left and right mirror

C++

```
// true Turn on false to close
```

```
mStream.setMirroringEnabled(true);
```

```
// Get status  
  
bool status = mStream. getMirroringEnabled();
```

Java

```
// True turns on false and closes  
  
mStream.setMirroringEnabled(true);  
  
// Get status  
  
boolean status = mStream. getMirroringEnabled();
```

4.16. Set up flip up and down

C++

```
// 1 open 0 close  
  
OniBool wValue = 1;  
  
mStream.setProperty(AXONLINK_STREAM_PROPERTY_FLIP, &wValue, sizeof(OniBool));  
  
// Get status  
  
OniBool wValue;  
  
int size = sizeof(OniBool);  
  
mStream.getProperty(AXONLINK_STREAM_PROPERTY_FLIP, &wValue, &size);
```

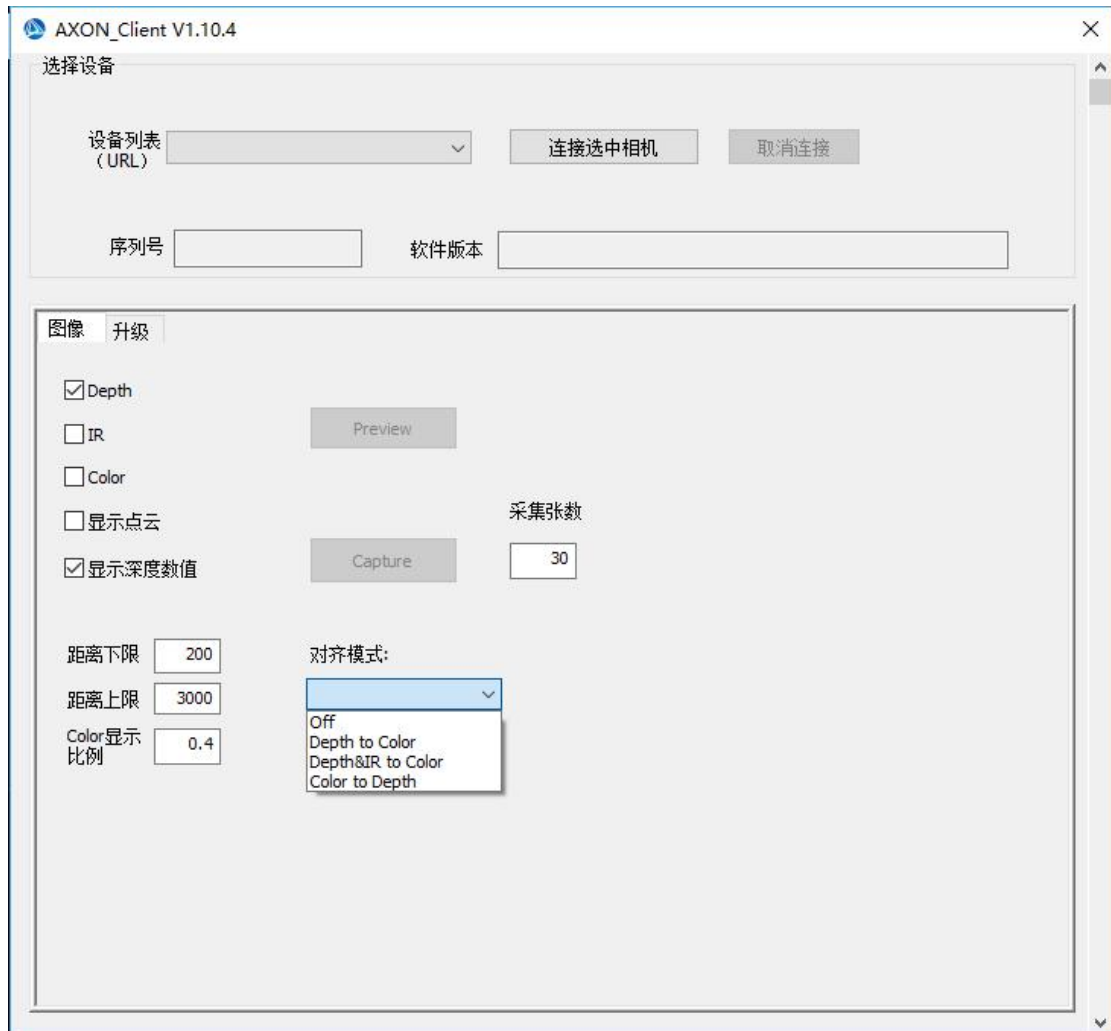
Java

```
// true open false close  
  
mStream. setFlipEnabled(true);  
  
// Get status  
  
boolean status = mStream. getFlipEnabled ();
```

5. Windows Client usage instructions

The client interface under Windows is as follows

5.1. Display page



➤ device attachment

When an identifiable device is inserted, the list of devices is automatically updated, and multiple devices can select the device you want to operate through the drop-down box

Click on the connection to select the camera button to connect;

➤ image display

Check the Depth, IR, Color stream, point cloud option, click on Preview to preview the image of the selected stream;

➤ Image preservation

Click the Capture button to save pictures of the selected stream. By default, 30 pieces can be

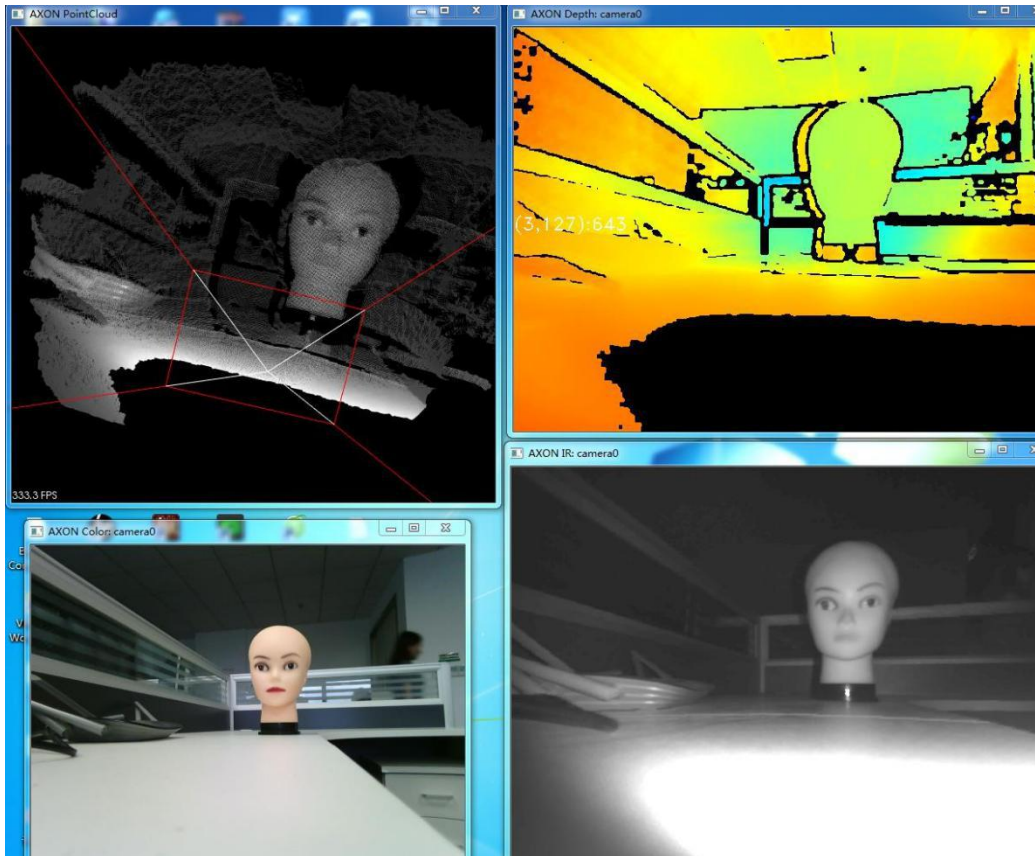
collected. The number of sheets to be saved can be modified manually. The pictures are saved in the image folder under the folder where the program is located, and the point cloud data is saved in depthdata/pointCloud.

Under folder;

➤ **Parameter setting**

The setting of the distance between the upper and lower limits can adjust the effective range of pseudo-color display of depth stream, and the depth value not within the limit range is not displayed. The Color display ratio can adjust the size of color stream display window and take effect when the Preview button is clicked again.

All the display effects are as follows:

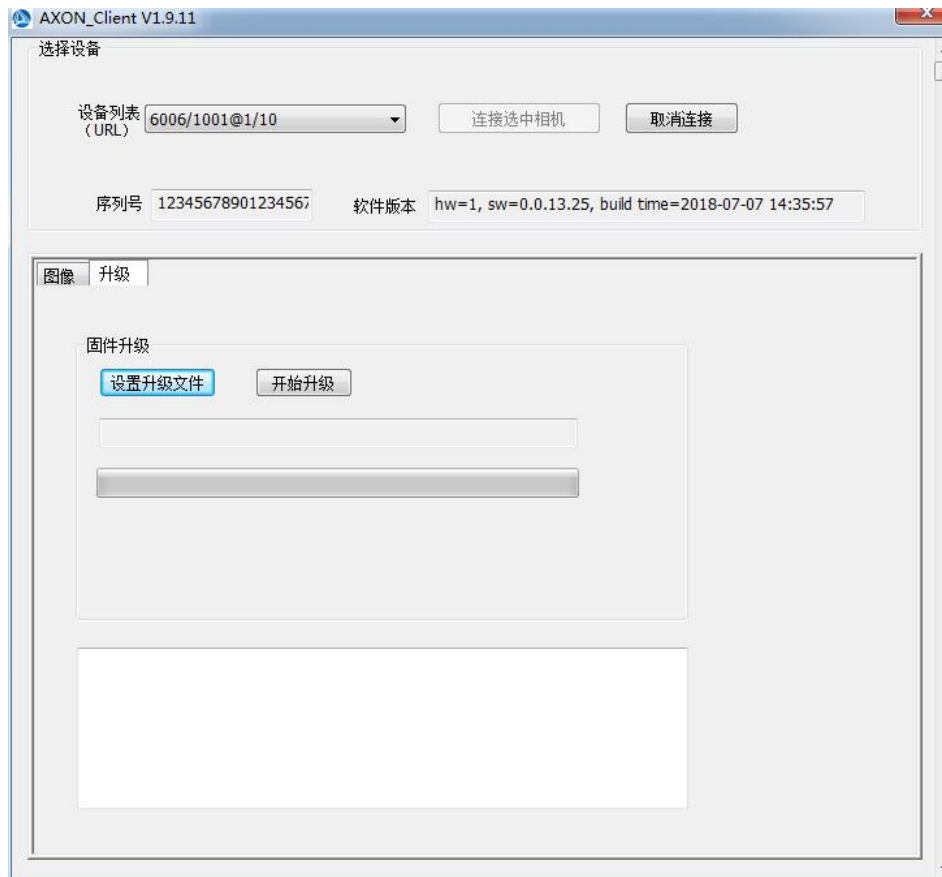


➤ **Alignment mode**

Four modes

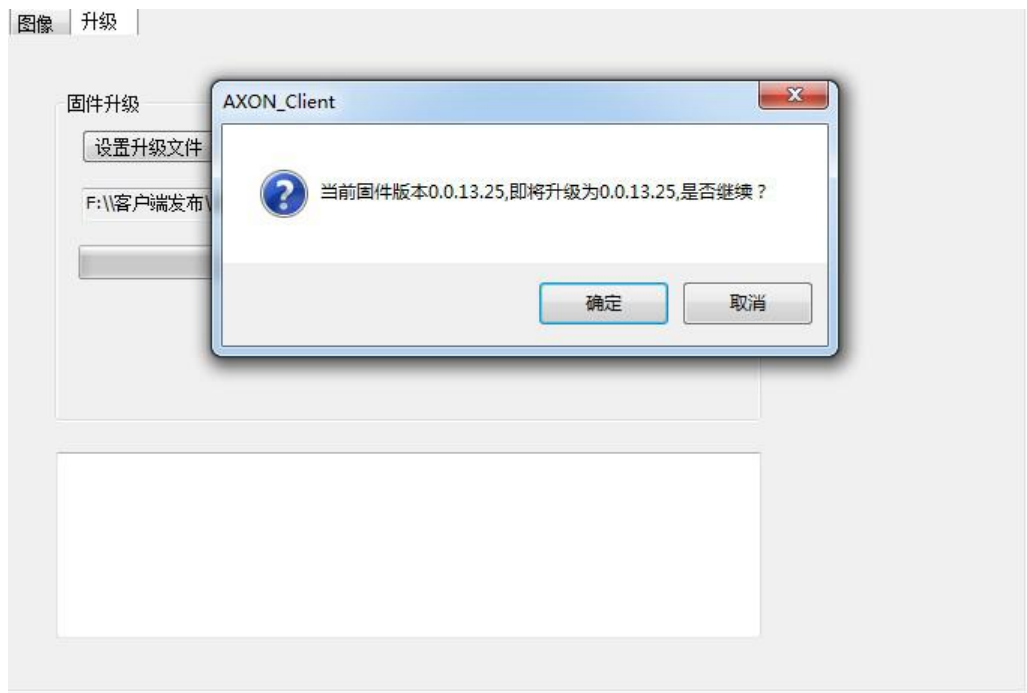
- | | |
|-------------------|---|
| Off | : Close alignment mode |
| Depth to Color | : depth map alignment color map |
| Depth&IR to Color | : Align color map with depth map and infrared map |
| Color to Depth | : Color map alignment depth map |

5.2. Upgrade page

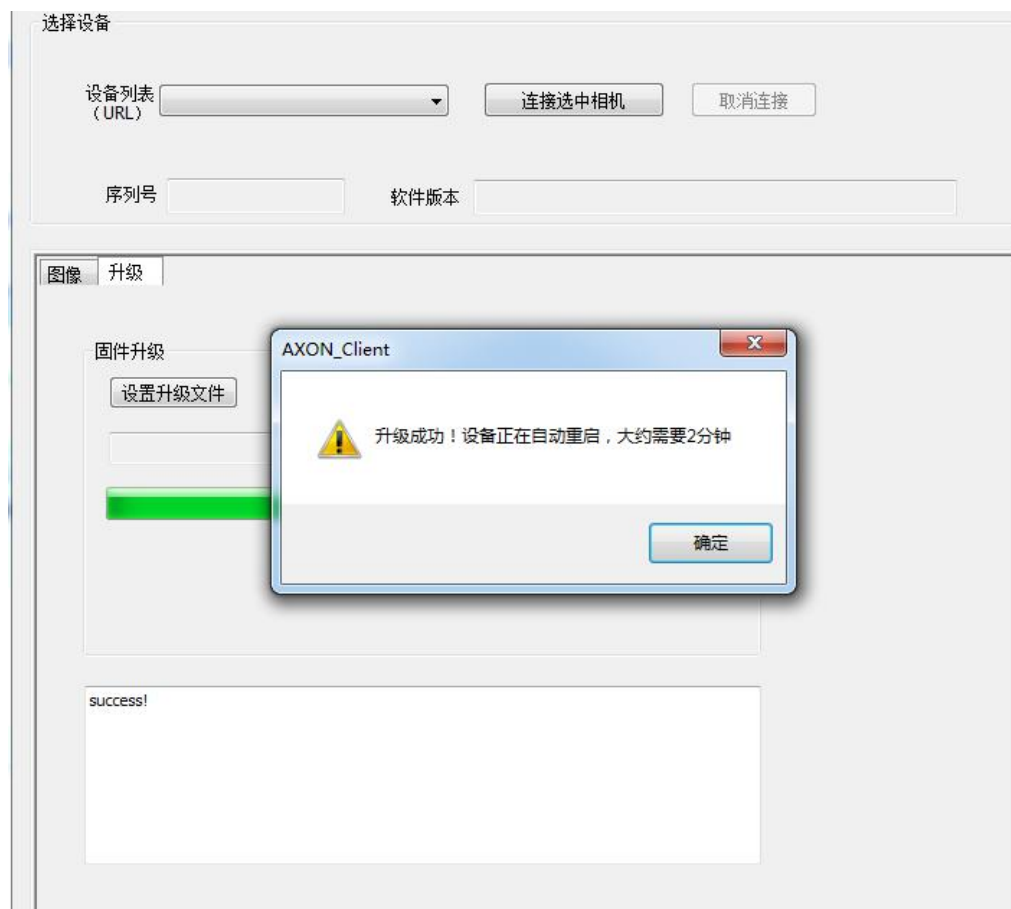


- Click set up upgrade File

Select the mirrored file you want to upgrade (extension ".img"); click "start upgrade button"



- To make sure you want to upgrade, click OK, after the upgrade is successful, there are the



following tips, click OK, wait for the device to restart

6. FAQ

1. The device cannot be identified after the win10 system is updated

In win10 environment, if the system is updated, it may cause the device to be unrecognized, and an attempt is made to restore the system until before the update.

2. Android platform can't get the device

After you get root permissions, close Selinux.

3. The Linux platform cannot open the device

4. Two methods

- (1) The application uses sudo to execute;
- (2) Use sudo to execute the install.sh script in the root directory and replug the device USB.