



## 实训软件配置说明

## 目录

|                                    |    |
|------------------------------------|----|
| 第一章 基本环境软件下载与安装 .....              | 2  |
| 一、下载和安装 Orbbec 相机驱动 .....          | 3  |
| 二、Visual Studio 2019 下载安装 .....    | 4  |
| 1、下载 .....                         | 4  |
| 2、安装 .....                         | 4  |
| 3、安装完成 .....                       | 6  |
| 4、启动 Visual Studio 2019 .....      | 6  |
| 三、Cmake 下载安装 .....                 | 6  |
| 1、下载 .....                         | 6  |
| 2、运行 .....                         | 7  |
| 第二章 3D 传感器工具及 SDK 调(OpenNI2) ..... | 8  |
| 一、Orbbec Viewer 看图工具介绍 .....       | 8  |
| 1、Orbbec Viewer 看图工具 .....         | 8  |
| 2、Orbbec Viewer 功能 .....           | 8  |
| 二、OpenNI2 SDK 介绍 .....             | 9  |
| 1、下载 OpenNI2 SDK V2.3.0.81 .....   | 9  |
| 2、OpenNI2 SDK 使用流程 .....           | 9  |
| 3、示例清单 .....                       | 13 |
| 三、Astra SDK 介绍 .....               | 13 |
| 1、下载 Astra SDK 2.1.3 .....         | 13 |
| 2、Astra SDK 的使用流程 .....            | 13 |
| 3、示例清单 .....                       | 17 |
| 第三章 3D 传感器工具及 SDK 调试(UVC) .....    | 18 |
| 一、Orbbec Viewer 看图工具介绍 .....       | 18 |
| 1、下载 OrbbecViewer 看图工具 .....       | 18 |
| 2、Orbbec Viewer 功能 .....           | 18 |
| 3.设备信息 .....                       | 19 |
| 二、Orbbec SDK 介绍 .....              | 19 |
| 1、下载 Orbbec SDK 工程 .....           | 19 |
| 2、设备连接 .....                       | 19 |
| 3、环境配置 .....                       | 20 |
| 4、Orbbec SDK 编译与使用 .....           | 21 |



# 第一章 基本环境软件下载与安装

## 一、下载和安装 Orbbec 相机驱动

<https://dl.orbbec3d.com/dist/drivers/win32/astra-win32-driver-4.3.0.22.zip>



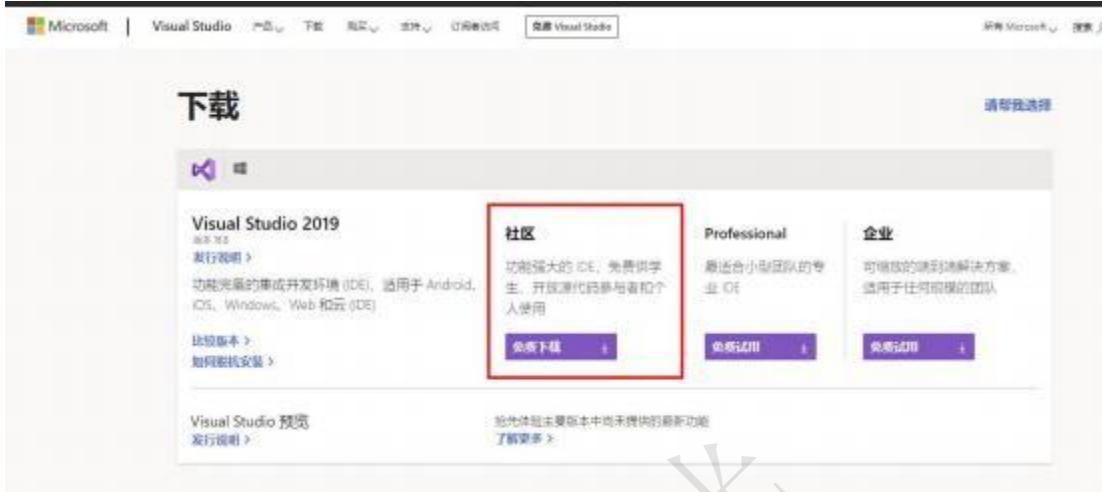
## OpenNI2驱动安装及设备诊断指南

<https://kdocs.cn/l/cuAUDUUJO67E>

## 二、Visual Studio 2019 下载安装（务必 VS2019）

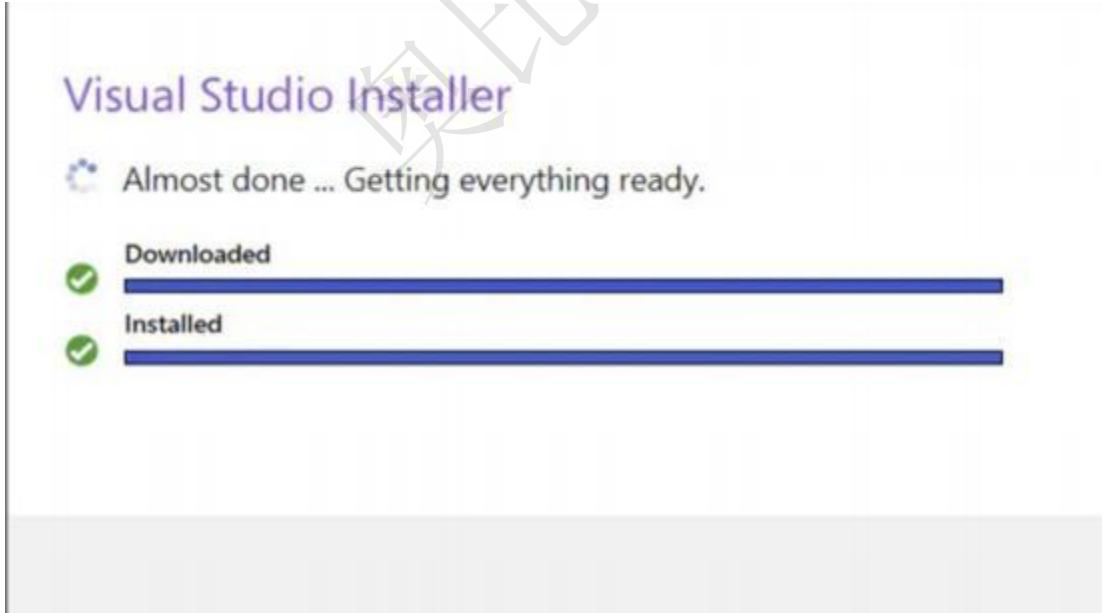
### 1、下载

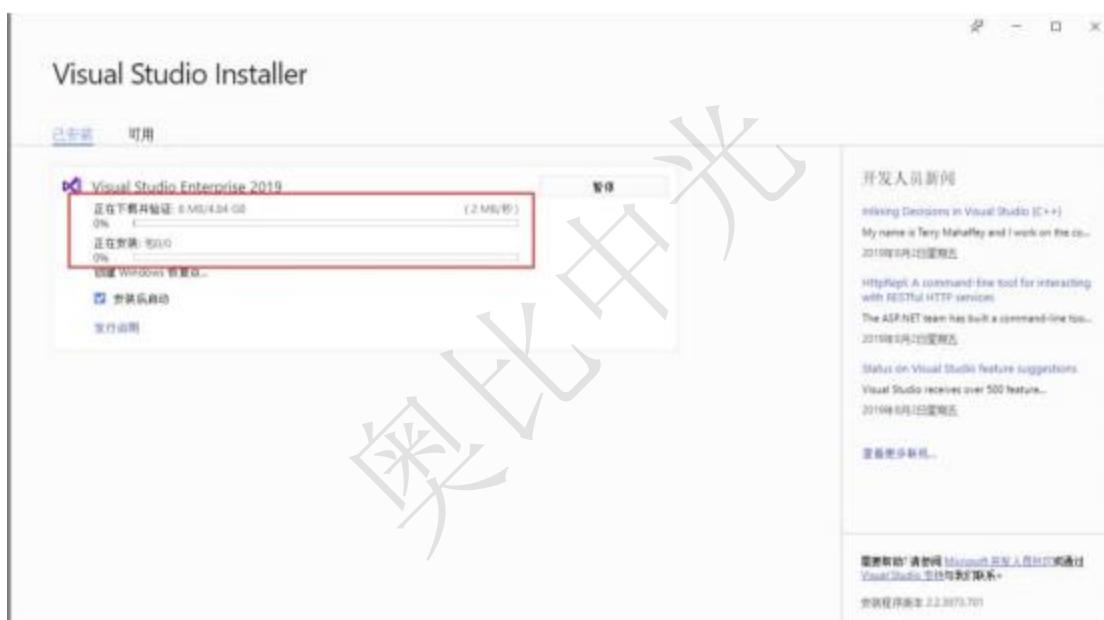
<https://visualstudio.microsoft.com/zh-hans/downloads/>



### 2、安装

下载 Visual Studio 2019 安装程序后，运行安装程序。安装程序将下载所需文件并进行安装。





### 三、Cmake 下载安装

#### 1、下载

<https://cmake.org/download/>

The screenshot shows the CMake download page. At the top, there are links for 'About', 'Resources', 'Developer Resources', and 'Downloads'. Below that, there are sections for 'Source distributions:' and 'Binary distributions:'.

**Source distributions:**

| Platform                              | Files               |
|---------------------------------------|---------------------|
| Unix/Linux Source (has \n line feeds) | cmake-3.20.2.tar.gz |
| Windows Source (has \r\n line feeds)  | cmake-3.20.2.zip    |

**Binary distributions:**

| Platform  | Files   |
|---|---|
| Windows x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!  | cmake-3.20.2-windows-x86_64.msi   |
| Windows x64 ZIP   | cmake-3.20.2-windows-x86_64.zip   |
| Windows i386 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first! | cmake-3.20.2-windows-i386.msi   |
| Windows i386 ZIP  | cmake-3.20.2-windows-i386.zip   |
| macOS 10.13 or later  | cmake-3.20.2-macos-universal.dmg  |
| macOS 10.10 or later  | cmake-3.20.2-macos10.10-universal.dmg<br>cmake-3.20.2-macos10.10-universal.tar.gz |
| Linux x86_64  | cmake-3.20.2-linux-x86_64.sh  |

#### 2、运行

下载 cmake-3.20.2-windows-x86\_64.zip，解压并运行 cmake-gui.exe 文件。



## 第二章 3D 传感器工具及 SDK 调(OpenNI2)

### 一、Orbbec Viewer 看图工具介绍

#### 1、Orbbec Viewer 看图工具

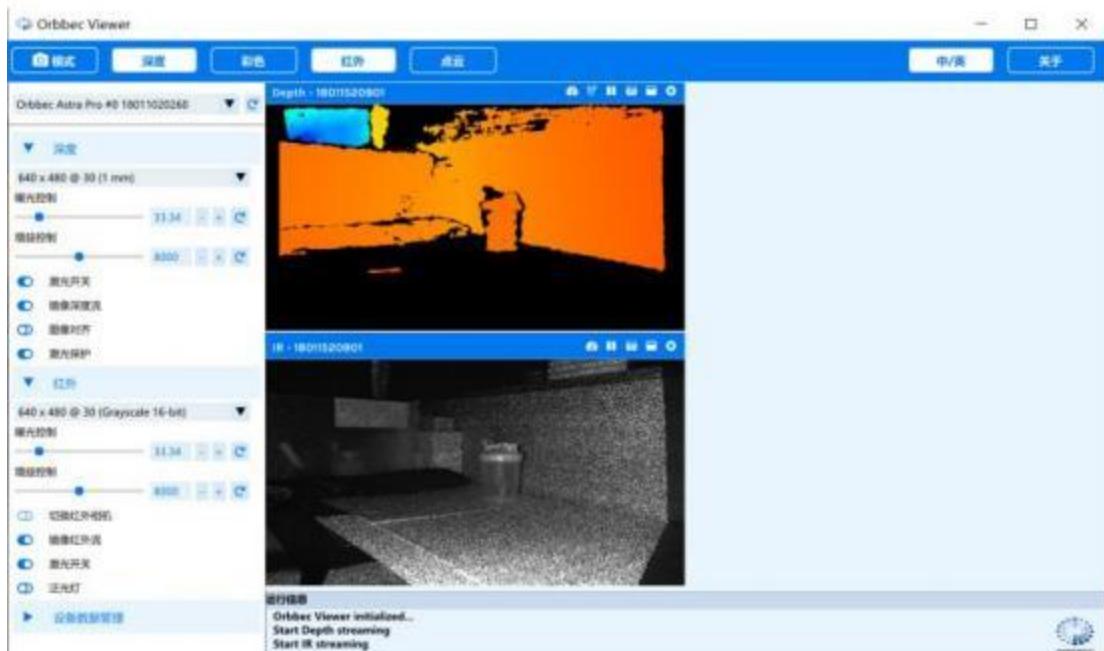
OrbbecViewer+for+OpenNI2\_v1.1.13\_20220722\_windows\_x64

<https://kdocs.cn/l/cfj9GZNl2b7u>

#### 2、Orbbec Viewer 功能

OrbbecViewer for OpenNI2 使用手册

<https://kdocs.cn/l/cazhWWIUDI5z>



## 二、OpenNI2 SDK 介绍

### 1、下载 OpenNI2 SDK V2.3.0.81

[https://dl.orbbec3d.com/dist/openni2/v2.3.0.86-beta6/Orbbec\\_OpenNI\\_v2.3.0.86-beta6\\_windows\\_release.zip](https://dl.orbbec3d.com/dist/openni2/v2.3.0.86-beta6/Orbbec_OpenNI_v2.3.0.86-beta6_windows_release.zip)

### 2、OpenNI2 SDK 使用流程

1) 首先需要安装 cmake 和 vs2019。

(参考上文)

使用 cmake 进行编译生成示例工程文件：

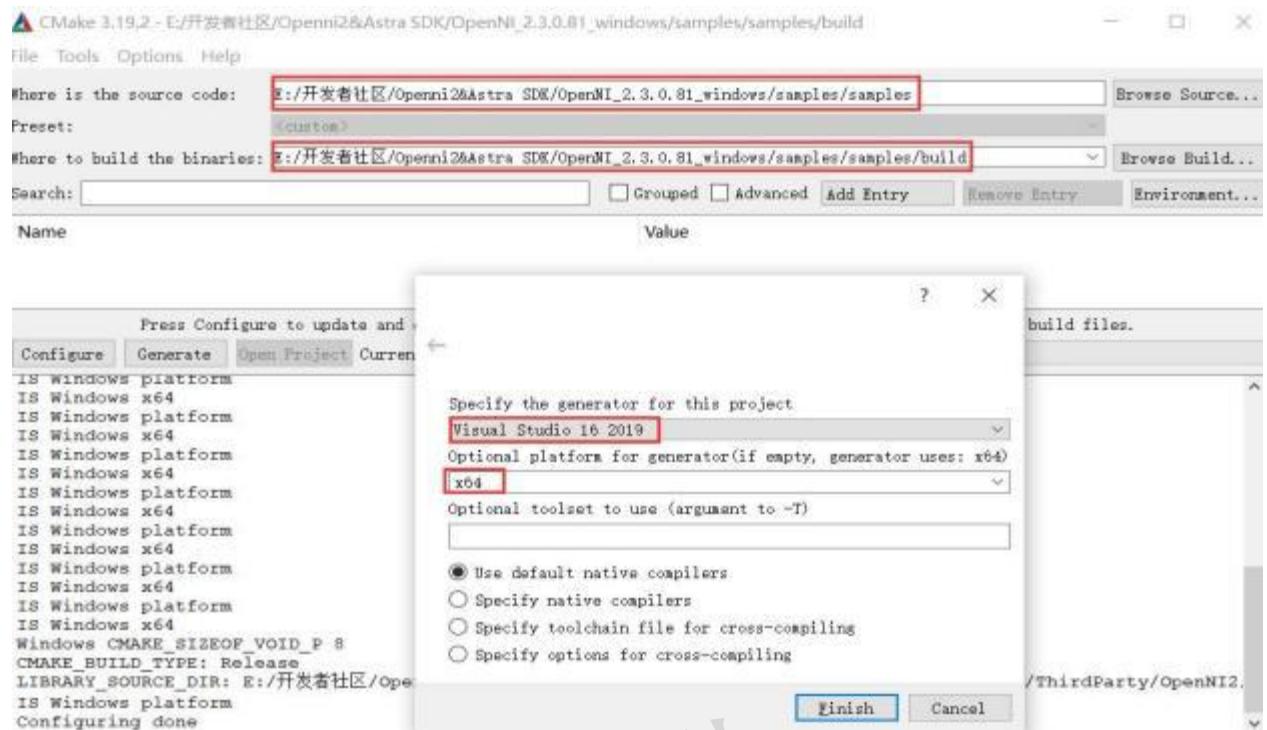
2) 在 SDK 提供的 Samples 目录下创建 build 目录



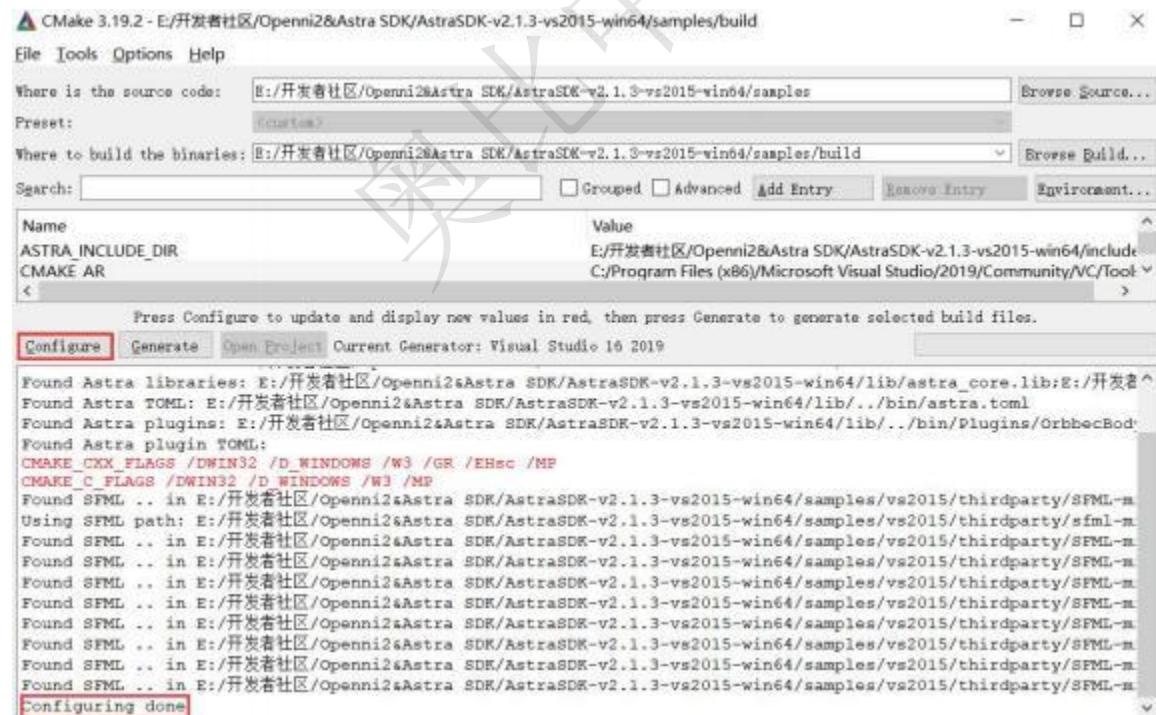
| 名称                      | 修改日期             | 类型         | 大小     |
|-------------------------|------------------|------------|--------|
| build                   | 2022/2/9 15:11   | 文件夹        |        |
| ColorReaderEvent        | 2021/10/28 11:57 | 文件夹        |        |
| ColorReaderPoll         | 2021/10/28 11:57 | 文件夹        |        |
| ColorReaderUVC          | 2021/10/28 11:57 | 文件夹        |        |
| DepthReaderEvent        | 2021/10/28 11:57 | 文件夹        |        |
| DepthReaderPoll         | 2021/10/28 11:57 | 文件夹        |        |
| ExtendedAPI             | 2021/10/28 11:57 | 文件夹        |        |
| GeneratePointCloud      | 2021/10/28 11:57 | 文件夹        |        |
| InfraredOniFileReader   | 2021/10/28 11:57 | 文件夹        |        |
| InfraredOniFileRecorder | 2021/10/28 11:57 | 文件夹        |        |
| InfraredReaderEvent     | 2021/10/28 11:57 | 文件夹        |        |
| InfraredReaderPoll      | 2021/10/28 11:57 | 文件夹        |        |
| MultiDepthViewer        | 2021/10/28 11:57 | 文件夹        |        |
| SimpleViewer            | 2021/10/28 18:30 | 文件夹        |        |
| ThirdParty              | 2021/10/28 11:57 | 文件夹        |        |
| CMakeLists.txt          | 2021/10/28 11:57 | 文本文档       | 13 KB  |
| Readme.pdf              | 2021/10/28 18:30 | WPS PDF 文档 | 948 KB |

3) 使用 cmake 生成 vs2019 x64 项目：打开 cmake，设置源代码路径和生成 build 路径。

点击 configure，弹出配置窗口。选择 visual studio 16 2019，x64，如下图



4) 点击“Configure”配置 Visual Studio 16 2019，再点击 Finish 完成配置。



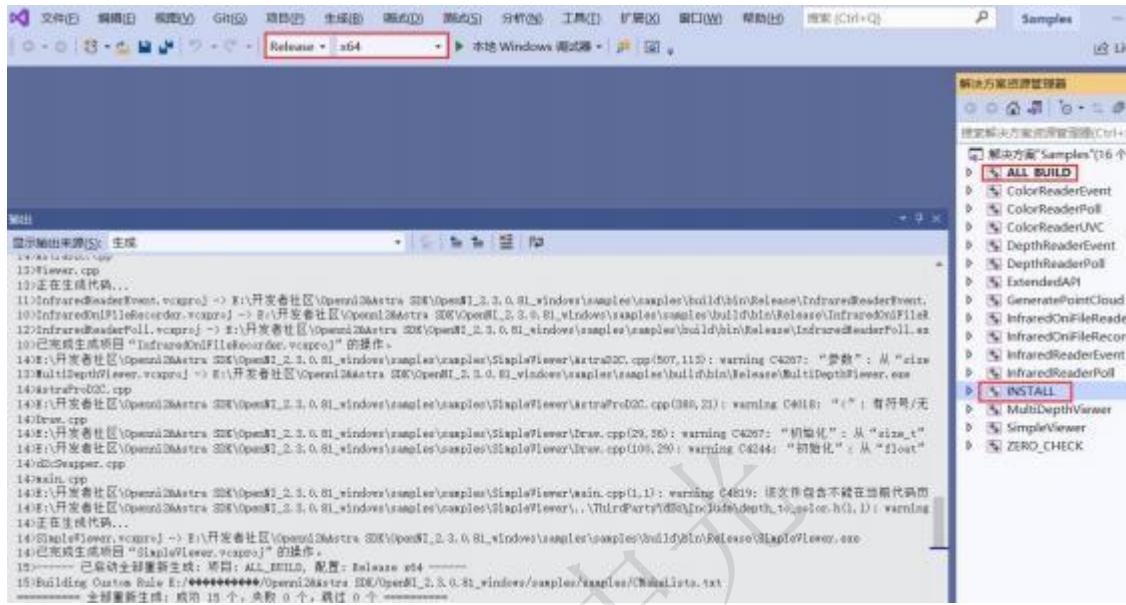
5) 编译完成后点击 generate，完成后如下图：



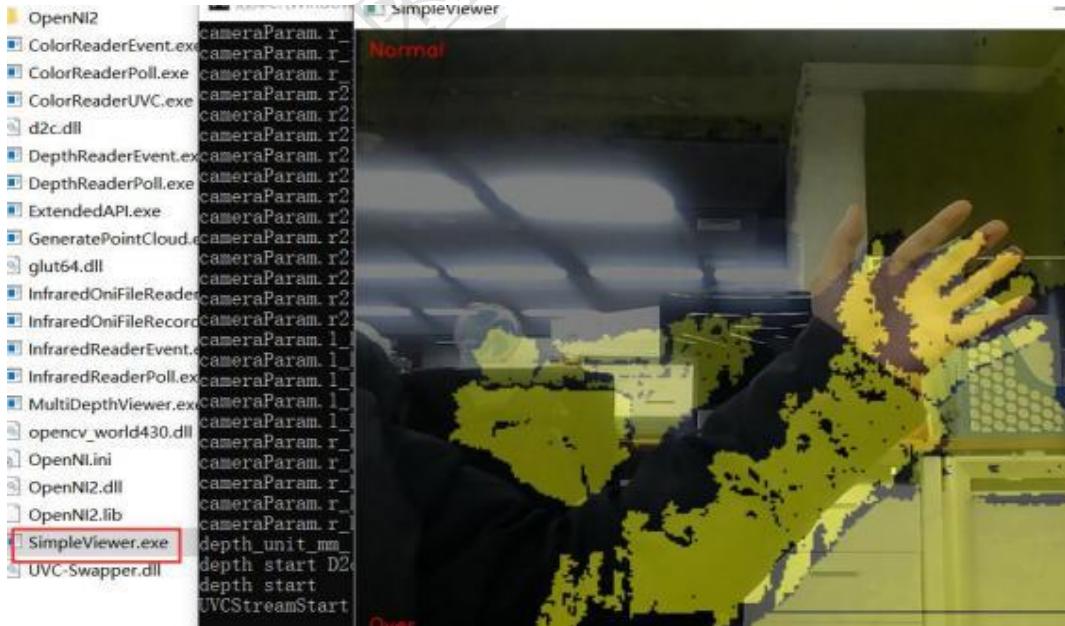
6) 进入编译后的 build 文件夹中，用 vs2019 打开 Samples.sln，如下图：

|                              |                |                            |
|------------------------------|----------------|----------------------------|
| 📁 CMakeFiles                 | 2022/2/9 15:25 | 文件夹                        |
| 📁 ColorReaderEvent           | 2022/2/9 15:25 | 文件夹                        |
| 📁 ColorReaderPoll            | 2022/2/9 15:25 | 文件夹                        |
| 📁 ColorReaderUVC             | 2022/2/9 15:25 | 文件夹                        |
| 📁 DepthReaderEvent           | 2022/2/9 15:25 | 文件夹                        |
| 📁 DepthReaderPoll            | 2022/2/9 15:25 | 文件夹                        |
| 📁 ExtendedAPI                | 2022/2/9 15:25 | 文件夹                        |
| 📁 GeneratePointCloud         | 2022/2/9 15:25 | 文件夹                        |
| 📁 InfraredOniFileReader      | 2022/2/9 15:25 | 文件夹                        |
| 📁 InfraredOniFileRecorder    | 2022/2/9 15:25 | 文件夹                        |
| 📁 InfraredReaderEvent        | 2022/2/9 15:25 | 文件夹                        |
| 📁 InfraredReaderPoll         | 2022/2/9 15:25 | 文件夹                        |
| 📁 MultiDepthViewer           | 2022/2/9 15:25 | 文件夹                        |
| 📁 SimpleViewer               | 2022/2/9 15:25 | 文件夹                        |
| 📄 ALL_BUILD.vcxproj          | 2022/2/9 15:25 | VC++ Project 27 KB         |
| 📄 ALL_BUILD.vcxproj.filters  | 2022/2/9 15:25 | VC++ Project Fil... 1 KB   |
| 📄 cmake_install.cmake        | 2022/2/9 15:25 | CMAKE 文件 8 KB              |
| 📄 CMakeCache.txt             | 2022/2/9 15:25 | 文本文档 19 KB                 |
| 📄 INSTALL.vcxproj            | 2022/2/9 15:25 | VC++ Project 13 KB         |
| 📄 INSTALL.vcxproj.filters    | 2022/2/9 15:25 | VC++ Project Fil... 1 KB   |
| 📄 Samples.sln                | 2022/2/9 15:25 | Microsoft Visual ... 17 KB |
| 📄 ZERO_CHECK.vcxproj         | 2022/2/9 15:25 | VC++ Project 36 KB         |
| 📄 ZERO_CHECK.vcxproj.filters | 2022/2/9 15:25 | VC++ Project Fil... 1 KB   |

7) 点击“Samples.sln”打开工程，点击“ALL\_BUILD”编译所有 Samples，点击“INSTALL”安装依赖库：



8) 运行执行文件 打开 build\bin\Release 文件夹，点击“exe”文件，例如执行 SimpleViewer.exe 可以显示如图运行的结果。



### 三、Astra SDK 介绍

#### 1、下载 Astra SDK 2.1.3

<https://dl.orbbec3d.com/dist/astra/v2.1.3/AstraSDK-v2.1.3-94bca0f52e-20210608T034051Z-vs2015-win64.zip>

#### 2、Astra SDK 的使用流程

- 1) 首先需要安装 cmake 和 vs2019。  
(参考上文)

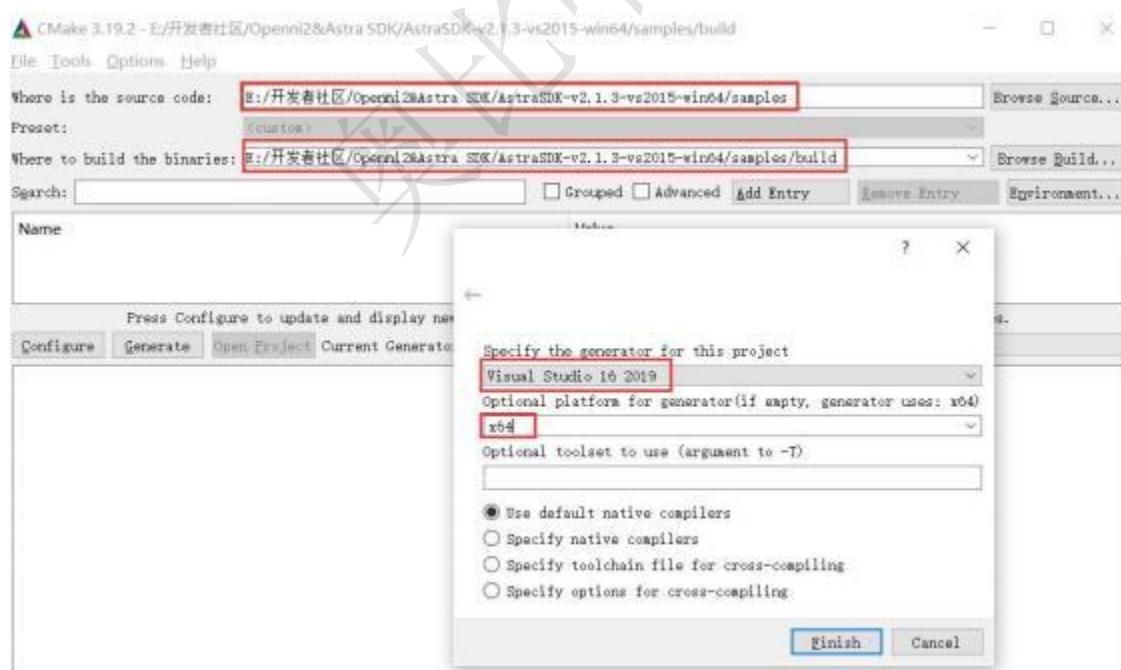
使用 cmake 进行编译生成示例工程文件：

- 2) 在 SDK 提供的 Samples 目录下创建 build 目录

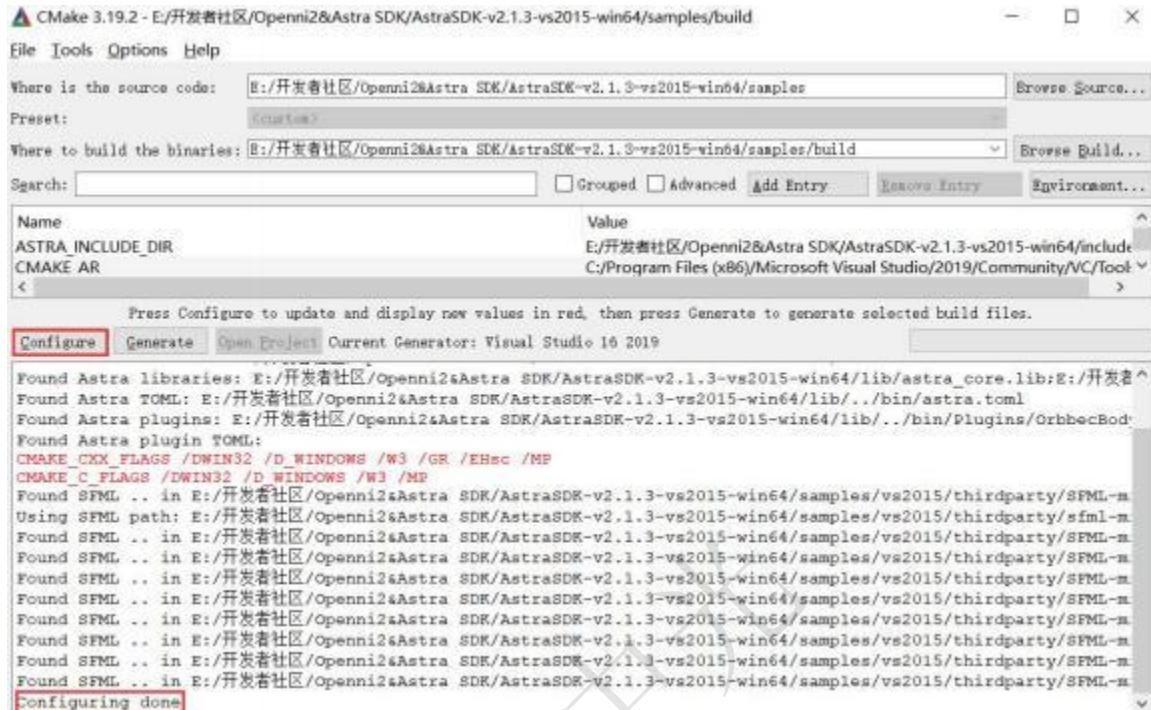
---

|                |                 |      |
|----------------|-----------------|------|
| build          | 2022/2/9 17:29  | 文件夹  |
| c-api          | 2021/6/8 11:42  | 文件夹  |
| cmake          | 2021/6/8 11:42  | 文件夹  |
| common         | 2021/6/8 11:42  | 文件夹  |
| cpp-api        | 2021/6/8 11:42  | 文件夹  |
| plugins        | 2021/6/8 11:42  | 文件夹  |
| sfml           | 2021/6/8 11:42  | 文件夹  |
| vendor         | 2021/6/8 11:42  | 文件夹  |
| vs2015         | 2021/7/5 16:44  | 文件夹  |
| CMakeLists.txt | 2021/7/7 10:00  | 文本文档 |
| readme.txt     | 2021/2/19 17:22 | 文本文档 |

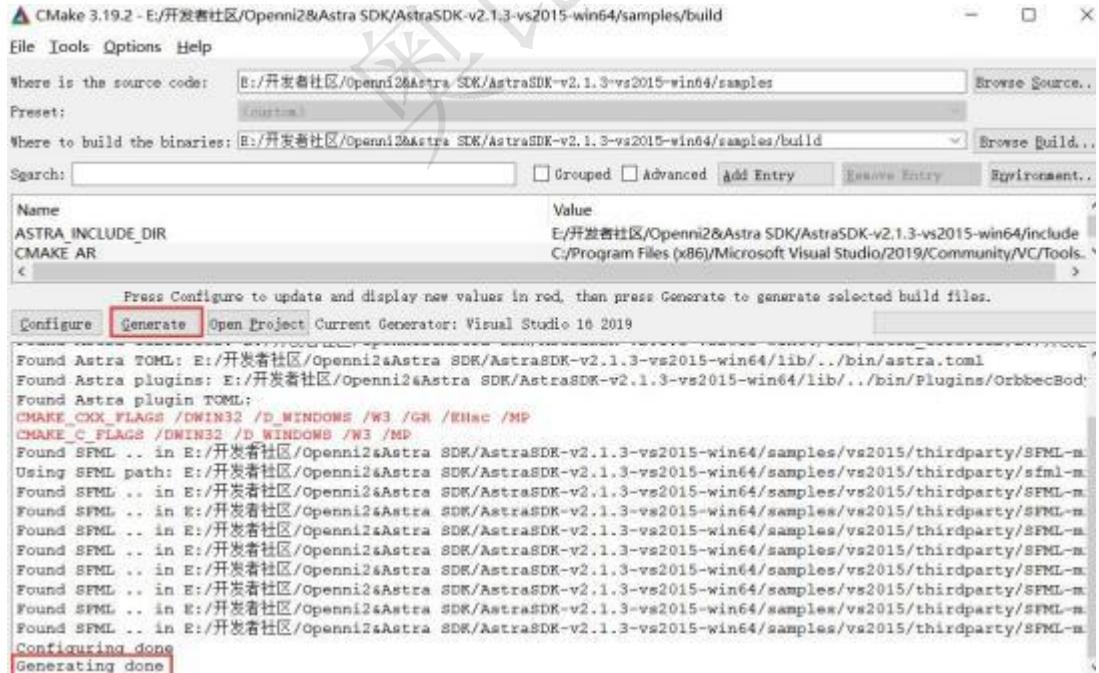
3) 使用 cmake 生成 vs2019 x64 项目：打开 cmake，设置源代码路径和生成 build 路径。配置 samples CMakeLists.txt 所在目录，配置编译 build 目录，点击 configure，弹出配置窗口，选择 visual studio 16 2019，x64，如下图



4) 点击“Configure”配置 Visual Studio 16 2019，再点击 Finish 完成配置。



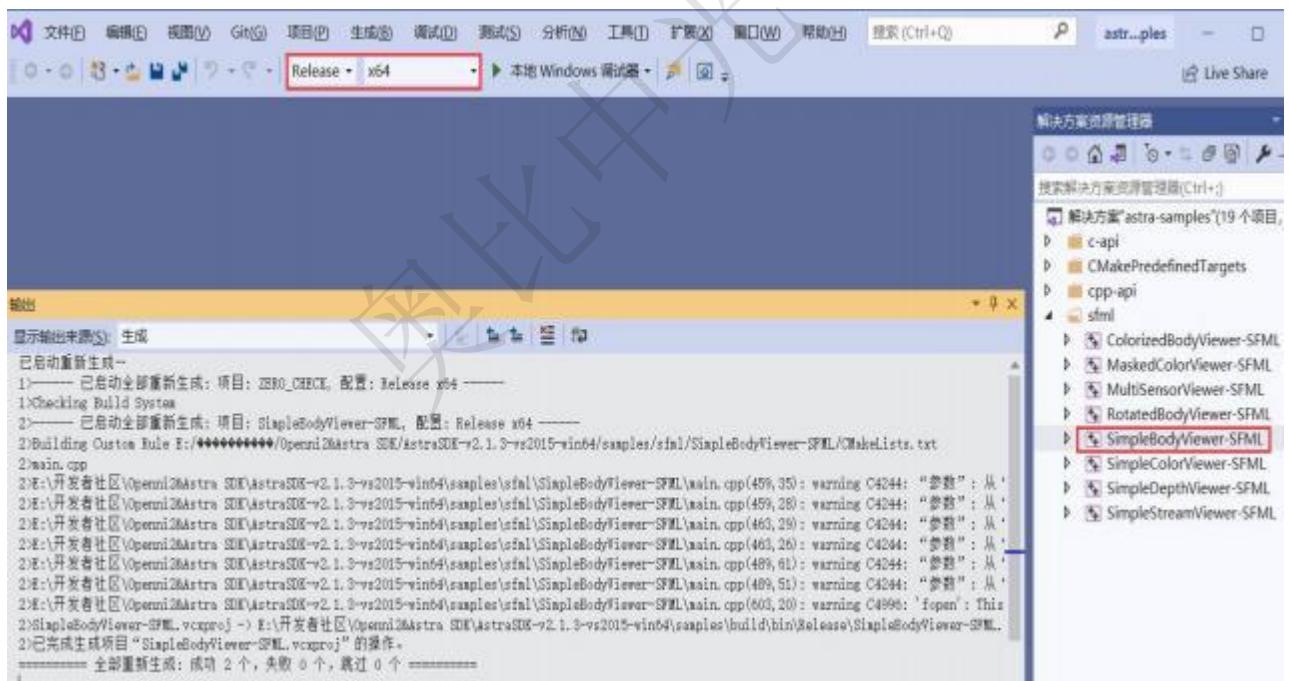
5) 编译完成后点击 generate，完成后如下图：



6) 进入编译后的 build 文件夹中，用 vs2019 打开 astra-samples.sln，如下图：

|                            |                |                            |
|----------------------------|----------------|----------------------------|
| .vs                        | 2022/2/9 17:41 | 文件夹                        |
| bin                        | 2022/2/9 17:42 | 文件夹                        |
| c-api                      | 2022/2/9 17:39 | 文件夹                        |
| CMakeFiles                 | 2022/2/9 17:42 | 文件夹                        |
| cpp-api                    | 2022/2/9 17:39 | 文件夹                        |
| lib                        | 2022/2/9 17:43 | 文件夹                        |
| sfml                       | 2022/2/9 17:39 | 文件夹                        |
| x64                        | 2022/2/9 17:42 | 文件夹                        |
| ALL_BUILD.vcxproj          | 2022/2/9 17:39 | VC++ Project 33 KB         |
| ALL_BUILD.vcxproj.filters  | 2022/2/9 17:39 | VC++ Project Fil... 1 KB   |
| <b>astra-samples.sln</b>   | 2022/2/9 17:39 | Microsoft Visual ... 22 KB |
| cmake_install.cmake        | 2022/2/9 17:39 | CMAKE 文件 2 KB              |
| CMakeCache.txt             | 2022/2/9 17:37 | 文本文档 20 KB                 |
| ZERO_CHECK.vcxproj         | 2022/2/9 17:39 | VC++ Project 48 KB         |
| ZERO_CHECK.vcxproj.filters | 2022/2/9 17:39 | VC++ Project Fil... 1 KB   |

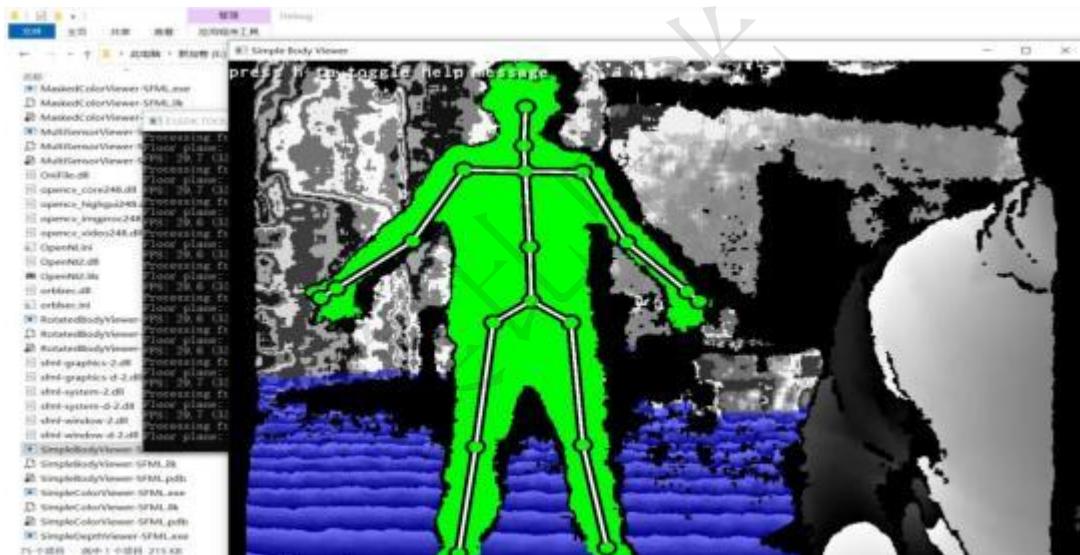
7) 选择想要编译的项目，选中后点击鼠标右键，选择生成项目，如下图：



8) **注意**：执行应用程序时提示缺少库文件，可在 sdk 自带的 bin 文件夹中找到并复制粘贴到应用程序的同级目录下，把“自带 bin 文件夹”下除 exe 外的文件/文件夹都复制到 build/bin/Release 路径中。

| 名称                    | 修改日期           | 类型     | 大小     |
|-----------------------|----------------|--------|--------|
| 应用程序 (17)             |                |        |        |
| smll-window-d-2.dll   | 2021/6/8 11:41 | 应用程序扩展 | 38 KB  |
| smll-window-2.dll     | 2021/6/8 11:41 | 应用程序扩展 | 375 KB |
| smll-system-d-2.dll   | 2021/6/8 11:41 | 应用程序扩展 | 12 KB  |
| smll-system-2.dll     | 2021/6/8 11:41 | 应用程序扩展 | 56 KB  |
| smll-graphics-d-2.dll |                |        |        |
| smll-graphics-2.dll   |                |        |        |
| orbbec.dll            |                |        |        |
| OpenNI2.dll           |                |        |        |
| opencv_video248.dll   |                |        |        |
| opencv_imgproc248.dll |                |        |        |
| opencv_highgui248.dll |                |        |        |
| opencv_core248.dll    |                |        |        |
| OnFile.dll            |                |        |        |
| AstraDotNet.dll       |                |        |        |
| astra_core_api.dll    |                |        |        |
| astra_core.dll        |                |        |        |
| astra.dll             |                |        |        |

9) 运行执行文件 进入 build/bin/Release 文件夹下，选择对应的 exe 文件，双击执行，例如运行 SimpleBodyViewer-SFML.exe 文件如下图：



### 3、示例清单

[https://dl.orbbec3d.com/dist/astra/v2.1.3/Doc\\_SDK\\_AstraSDK\\_v2.1.3.zip](https://dl.orbbec3d.com/dist/astra/v2.1.3/Doc_SDK_AstraSDK_v2.1.3.zip)

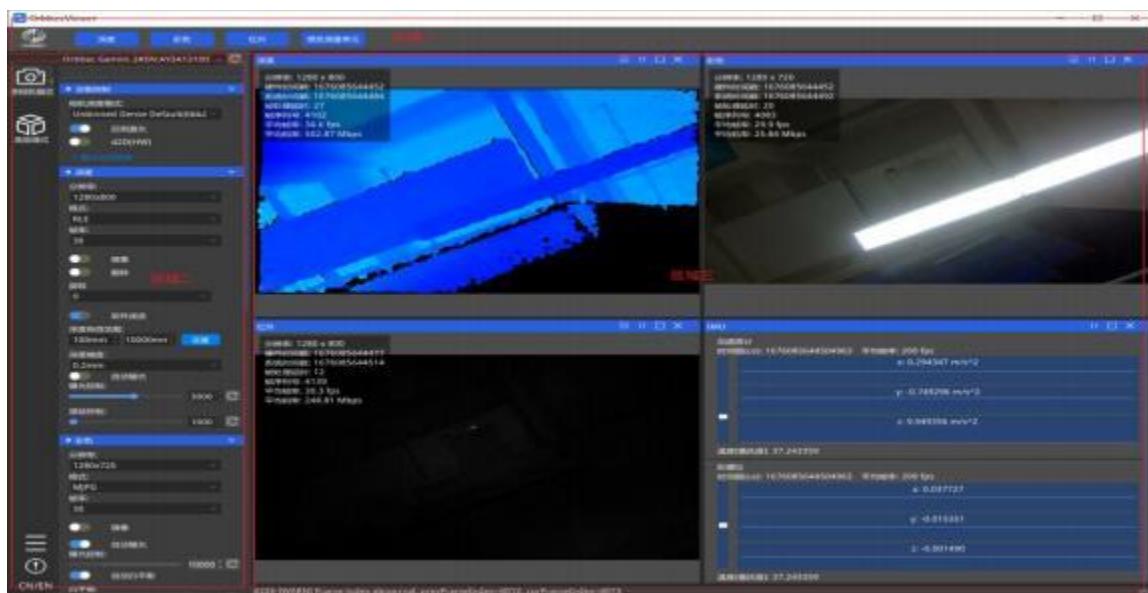
# 第三章 3D 传感器工具及 SDK 调试(UVC)

## 一、Orbbec Viewer 看图工具介绍

### 1、下载 OrbbecViewer 看图工具

OrbbecViewer是一个基于Orbbec SDK的实用工具

<https://gitee.com/orbbecdeveloper/OrbbecSDK#orbbecviewer>



### 2、Orbbec Viewer 功能

OrbbecViewer 工具手册

<https://gitee.com/orbbecdeveloper/OrbbecSDK/blob/main/doc/OrbbecViewer/Chinese/Orbbe cViewer.md>

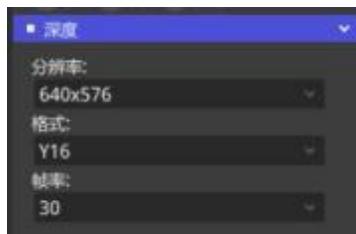
OrbbecViewer 工具的使用

1 软件主界面

| 区域  | 名称    | 功能                          |
|-----|-------|-----------------------------|
| 区域一 | 设备管理区 | 传感器列表、设备固件更新                |
| 区域二 | 控制面板区 | 数据流开关及参数配置、图像采集功能控制、D2C功能控制 |
| 区域三 | 图像预览区 | 传感器图像预览，帧率、时间戳、Log等信息查看     |

## 2 数据流

OrbbecViewer 允许用户选择和配置深度、红外和彩色数据流，以及在工具中配置参数。



## 3. 设备信息

OrbbecViewer 工具包含简单的设备信息，如固件版本、产品识别码、相机参数、温度等。

## 二、Orbbec SDK 介绍

### 1、下载 Orbbec SDK 工程

下载 OrbbecSDK\_C\_C++\_v1.7.5\*\*\*win\_x64\_release

<https://github.com/Orbbec/OrbbecSDK.git>

### 2、设备连接

将设备连接到主机

导航至“控制面板”->“设备管理器”

浏览查找 Orbbec 设备，如下图所示，设备连接成功



### 3、环境配置

配置 OpenCV ( Examples 依赖)

数据渲染依赖第三方库 OpenCV 开源视觉库(<https://opencv.org/releases/>)，

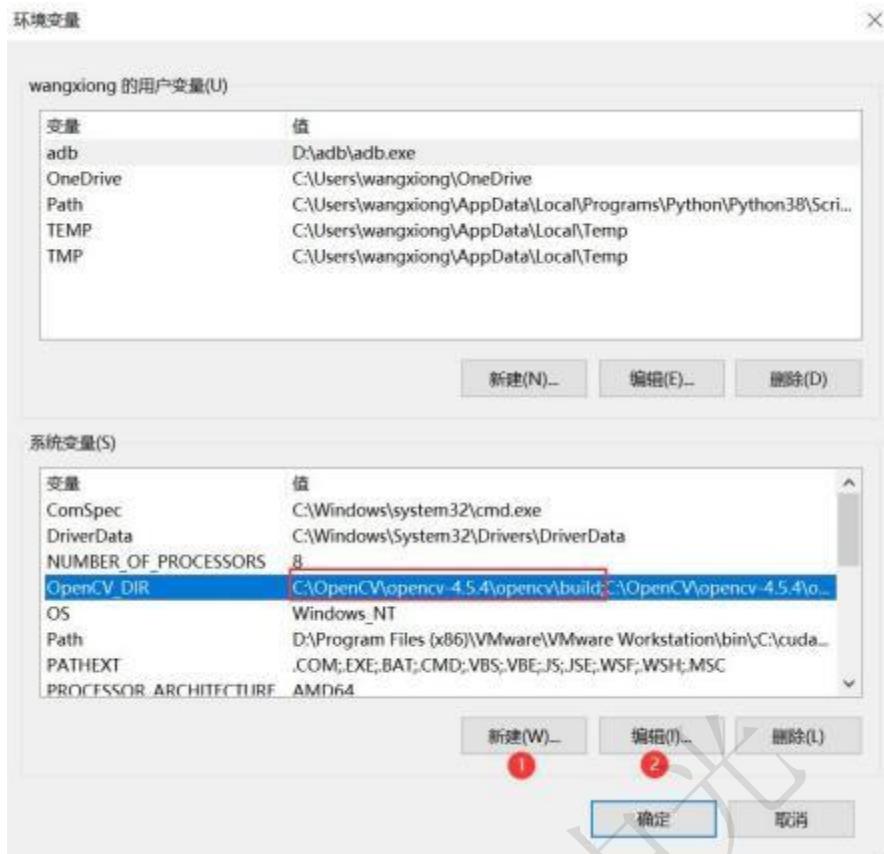
这里以 OpenCV 4.5.4 为例演示安装配置

1) 执行 OpenCV 安装文件，选择 opencv 要安装的目录，点击 extract 执行安装；

地磁盘 (C:) > OpenCV > opencv-4.5.4 > opencv >

| 名称                 | 修改日期             | 类型   |
|--------------------|------------------|------|
| build              | 2022/12/28 17:44 | 文件夹  |
| sources            | 2022/12/28 17:45 | 文件夹  |
| LICENSE.txt        | 2021/10/10 1:32  | 文本文档 |
| LICENSE_FFMPEG.txt | 2021/10/10 1:32  | 文本文档 |
| README.md.txt      | 2021/10/10 1:32  | 文本文档 |

2) 在系统的环境变量中添加 OpenCV 的路径，变量名输入 OpenCV\_DIR, 注意字母的大小写，变量值为 OpenCV 安装目录的 build 文件夹路径；



## 4、Orbbec SDK 编译与使用

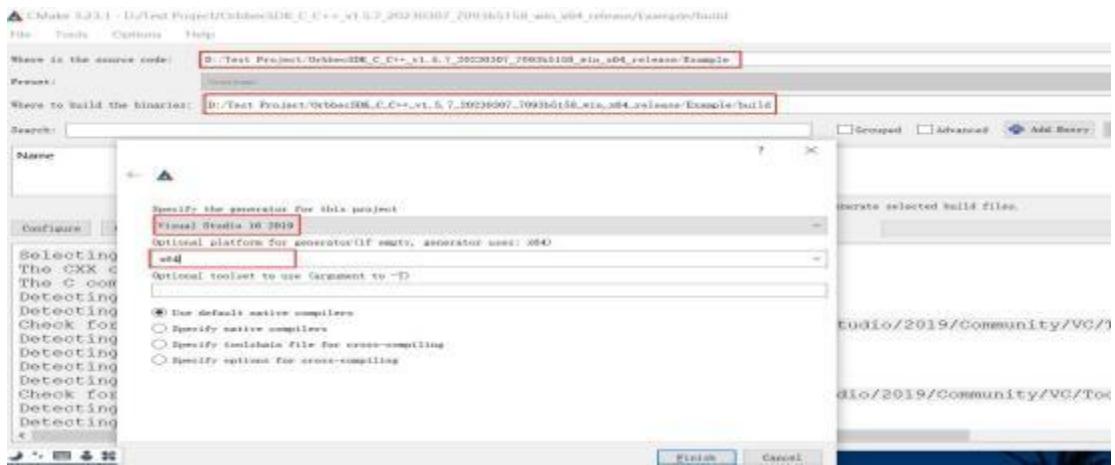
生成你的第一个应用程序

软件依赖：VisualStudio2019、cmake 3.10 及以上版本

1) 下载/获取我们的 SDK 软件包，存放位置假设为 D 盘根目录：“D:\Test Project”，目录结构如下所示：

| 名称                           | 修改日期           | 类型            | 大小 |
|------------------------------|----------------|---------------|----|
| Driver                       | 2023/3/7 16:21 | 文件夹           |    |
| Doc                          | 2023/3/7 16:21 | 文件夹           |    |
| Example → <b>示例工程与代码</b>     | 2023/3/7 16:21 | 文件夹           |    |
| Script                       | 2023/3/7 16:21 | 文件夹           |    |
| SDK → <b>放置具体wrapper的SDK</b> | 2023/3/7 16:21 | 文件夹           |    |
| README.md                    | 2023/3/6 20:00 | Markdown File |    |
| README_EN.md                 | 2023/3/6 20:00 | Markdown File |    |

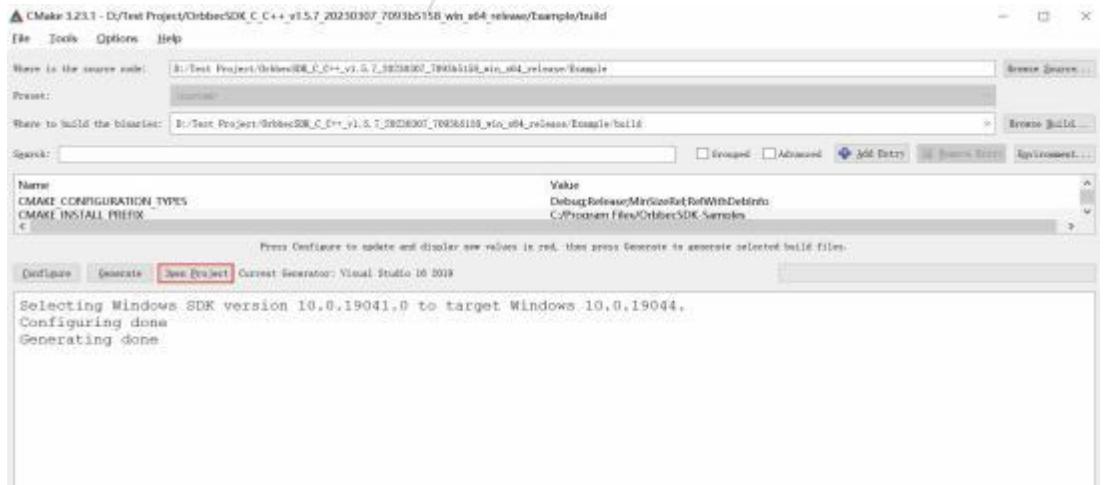
- 2) 打开 Cmake，将“Examples”文件夹设置为代码路径，“Examples”下的“build”文件夹设置为生成二进制文件的路径，如 Examples 下无 build，需要新建该文件夹。
- 3) 点击“Configure”并选择对应的 Visual Studio 版本和平台版本后，点击“Finish”，如下所示：



4) 点击“Generate”，如下所示：



5) 可以通过如下方式打开 Sample 工程，通过 cmake，点击“Open Project”按钮，打开 Visual Studio 工程。



6) 打开 Examples 工程界面如下所示：

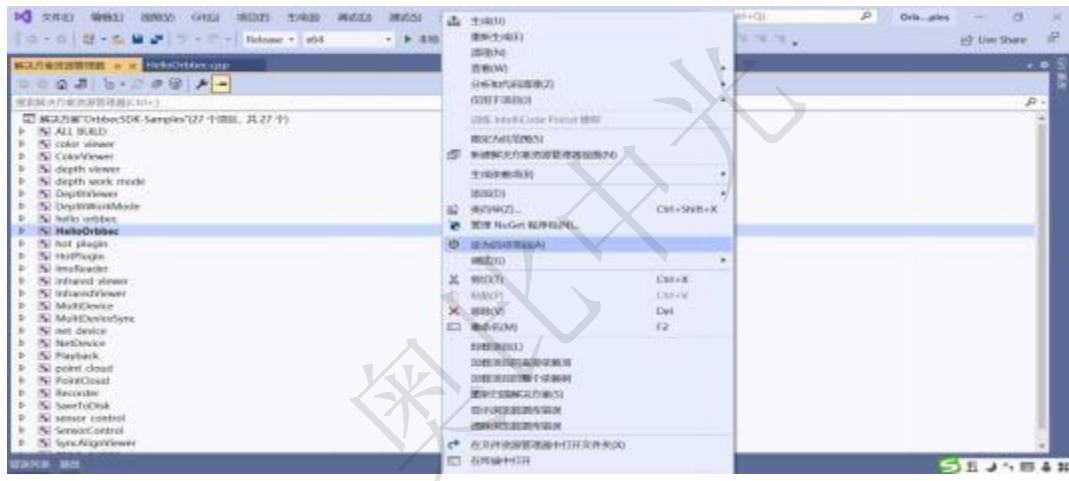
The screenshot shows the Visual Studio IDE with the 'HelloOrbbeccpp' project open. The code editor displays the main source file 'HelloOrbbeccpp.cpp'. The code includes comments explaining the creation of a context and querying device lists. The Solution Explorer on the right shows multiple projects under the 'OrbbeccSamples' solution.

```

1 //include <iostream>
2 #ifndef _WIN32_WCE_
3 #include <Windows.h>
4 #else
5 #include <emmintrin.h>
6 #endif
7 #include "liborbbecc/orbServer.hpp"
8 #include "liborbbecc/orb/Error.hpp"
9
10 #define _CRT_SECURE_NO_WARNINGS
11
12 int main(int argc, char **argv) try {
13     //打印出版本的版本号，30进制版本号分为主版本号，副版本号和修正版版本号
14     std::cout << "SDK version: " << std::Version::getMajor() << "." << std::Version::getMinor() << "." << std::Version::getPatch();
15
16     //创建一个Context，与Pipeline不同，Context是更高级的入口，在开发流程中常用操作上
17     //使用管道会稍微复杂一些，但是这是API可以提供更多灵活的操作，如加载多个设备，读写
18     //设备及相机的属性等
19     orb::Context ctx;
20
21     //查询已连接设备的列表
22     ext::DeviceList* extDeviceList = ctx.queryDeviceList();
23
24     //获取放入设备的数量
25     if(extDeviceList->deviceCount() == 0) {
26         std::cout << "Device not found" << std::endl;
27         return -1;
28     }

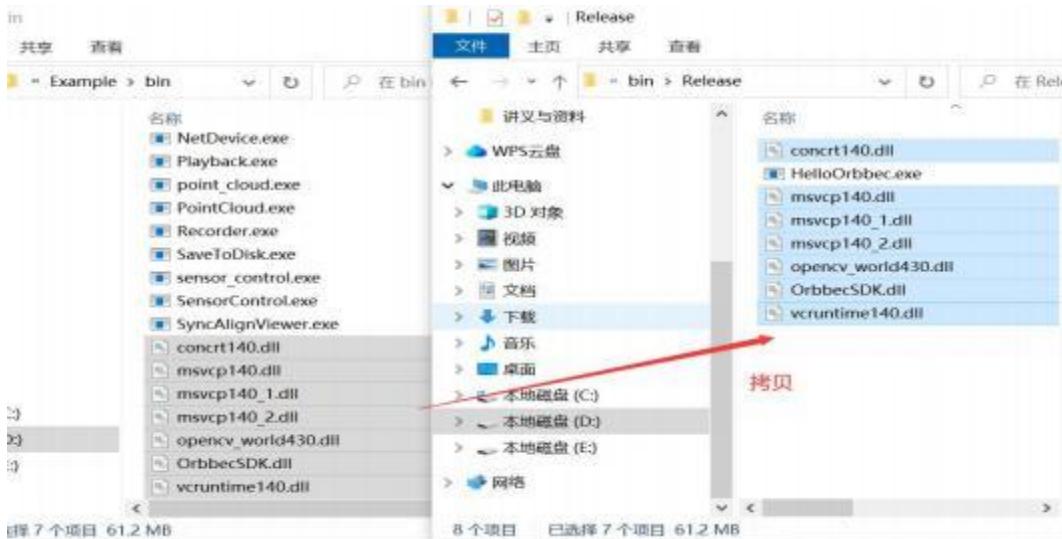
```

7) 选择你想要运行的工程，右键点击并将其“设置为启动项目”，在运行选项处选择 release 和 64 位版本；



8) 将设备连接到主机

9) 将 bin 目录(Example\bin)下的 dll 文件拷贝至执行文件处(build\bin\Release)



10) 运行工程，结果如下所示，至此第一个 Example 就运行成功了。