



XVSDK-Viewer 用户指南

Rev1.0

上海詮视传感技术有限公司

Xvisio Technology (Shanghai) Co., Ltd.

历史版本

版本	描述	作者
1.0	初版	Xvisio

目录

1. 概况	4
2. Ubuntu 平台	4
2.1. 配置 Ubuntu 系统环境 DFU	4
2.2. 安装 Ubuntu SDK 工具	6
3. Windows 平台	7
3.1. 安装 Windows SDK 工具	8
4. Android 平台	8
5. 运行 XVSDK-Viewer	10
6. ROS 环境搭建	16
6.1. 搭建流程	17
6.1.1. 基础文件准备	17
6.1.2. Some Libs (基础组件安装)	17
6.1.3. 安装必要的 ROS 包	17
6.1.4. Init Catkin Workspace	18
6.1.5. Build xv_sdk	18
6.2. 启动 Demo	19
6.3. Q&A	20
6.4. python_wrapper 环境搭建	22

1. 概况

本文档主要介绍了如何安装和使用 XVSDK-Viewer 工具。目前该工具支持三个 OS 平台：Ubuntu, Windows 和 Android。

2. Ubuntu 平台

2.1. 配置 Ubuntu 系统环境 DFU

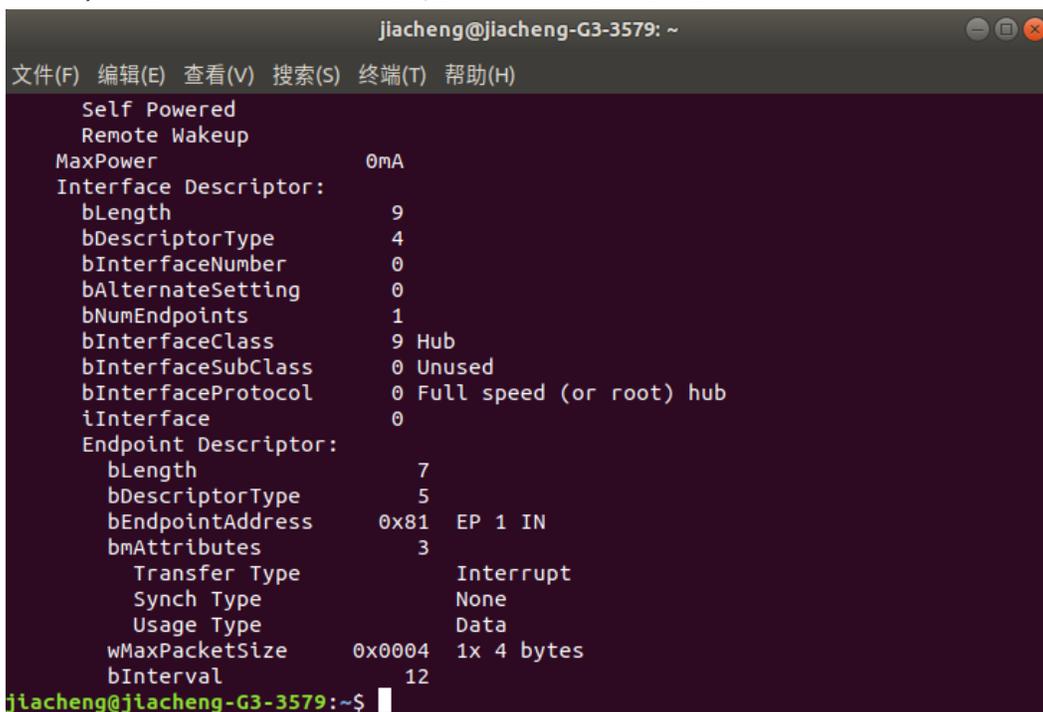
1) 打开终端，输入 “lsusb -vvv”：



```
jiacheng@jiacheng-G3-3579: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
jiacheng@jiacheng-G3-3579:~$ lsusb -vvv
```

图 2-1 输入 “lsusb -vvv”

2) 按下 “enter” 键后将会出现如下一系列编码：



```
jiacheng@jiacheng-G3-3579: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
Self Powered  
Remote Wakeup  
MaxPower          0mA  
Interface Descriptor:  
  bLength          9  
  bDescriptorType  4  
  bInterfaceNumber 0  
  bAlternateSetting 0  
  bNumEndpoints   1  
  bInterfaceClass  9 Hub  
  bInterfaceSubClass 0 Unused  
  bInterfaceProtocol 0 Full speed (or root) hub  
  iInterface       0  
Endpoint Descriptor:  
  bLength          7  
  bDescriptorType  5  
  bEndpointAddress 0x81 EP 1 IN  
  bmAttributes     3  
    Transfer Type  Interrupt  
    Synch Type    None  
    Usage Type    Data  
  wMaxPacketSize  0x0004 1x 4 bytes  
  bInterval       12  
jiacheng@jiacheng-G3-3579:~$
```

图 2-2 编码

3) 打开一个新的终端（不要关闭步骤 2 的终端），输入 “sudo gedit

/etc/udev/rules.d/50-myusb.rules”。

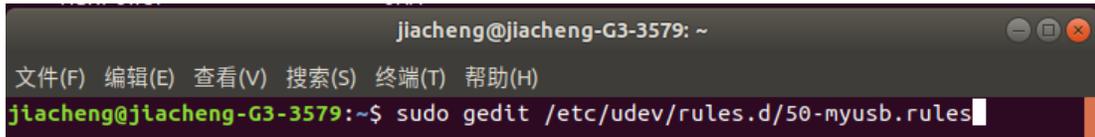


图 2-3 打开新终端

4) 按下 “enter” 键后将弹出一个记事本，在记事本中复制进：
SUBSYSTEMS=="usb",ATTRS{idVendor}=="1a86",ATTRS{idProduct}=="7523", GROUP="users", MODE="0666"，需要多复制几行。



图 2-4 弹出记事本

5) 打开步骤 1 中的终端，从上到下找出

idVendor"和"idProduct"后面的两个编码，并将记事本中的编码改成对应的。

比如“ idVendor 0xAAAA Sunplus Innovation Technology Inc. idProduct 0xBBBB ”，记事本中的指令需要改为 SUBSYSTEMS=="usb", ATTRS{idVendor}=="AAAA", ATTRS{idProduct}=="BBBB", GROUP="users", MODE="0666"

注意：只有出现在连续的两行的 idVendor 和 idProduct 才需要改，单独出现的 idVendor 或 idProduct 不需要：

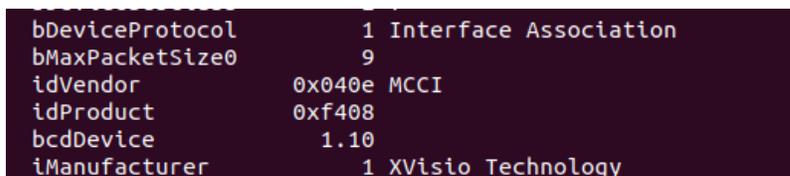


图 2-5 更改指令

6) 修改完成后如下图：



图 2-6 修改完成

7) 删掉多余的指令，保存并退出。

(注意：如果步骤 2 弹出的不是记事本而是命令窗口，需要输入“wq!”才能做到保存退出。)

8) 在终端中输入“sudo udevadm control --reload”重新加载 udev 规则，然后输入“sudo apt-get install dfu-util”安装驱动。

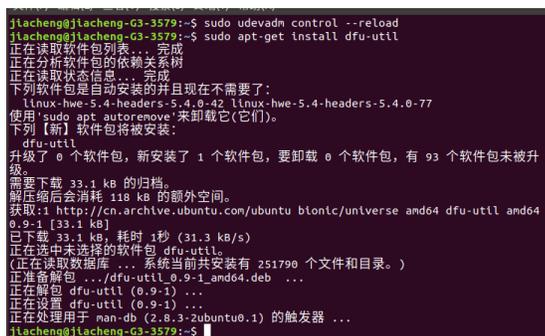


图 2-7 安装驱动

2.2. 安装 Ubuntu SDK 工具

1) 获取最新安装包;

安装包 1: xv sdk_3.2.0-2022xxxx_amd64.snap

安装包 2: xv sdk_3.2.0-2022xxxx_bionic_amd64 (for Ubuntu 18.04)

xv sdk_3.2.0-2022xxxx_focal_amd64 (for Ubuntu 20.04)

2) Copy 至对应系统的桌面文件夹下;

3) 如果以前有使用 snap 安装过 xv sdk-viewer，请先卸载 snap 目录下的 xv sdk-viewer。

- 查看已经安装的所有 snap 包，其中是否已经存在 xv sdk-viewer。

命令：`sudo snap list`

- 删除 xv sdk-viewer。

命令：`sudo snap remove xv sdk-viewer`

- 重新安装命令见第四步

4) 安装 Snap 包

命令：`sudo snap install --devmode xv sdk_3.2.0-2022xxxx_ amd64.snap`

5) 安装 Deb

命令：`sudo dpkg -i xv sdk_3.2.0-2022xxxx_ bionic_ amd64(或者 xv sdk_3.2.0-2022xxxx_ focal_ amd64)`

注意：因为部分 PC 环境不完整，可能双击 Deb 安装会出现异常，可以通过命令完成安装。

6) 新开终端输入命令：`xv sdk-viewer` 打开 viewer 界面。

7) 新开终端输入命令：`all_ stream` 打开所有 camera 界面。

3. Windows 平台

在进行下记的工具安装之前，先行配置 Windows 系统环境 DFU，具体步骤参考如下文档：



图 3-1 参考文档

DFU 驱动文件包资源如下：



图 3-2 DFU 驱动文件包

3.1. 安装 Windows SDK 工具

1) 获取 xv sdk-viewer 的安装包和 SDK 安装包。

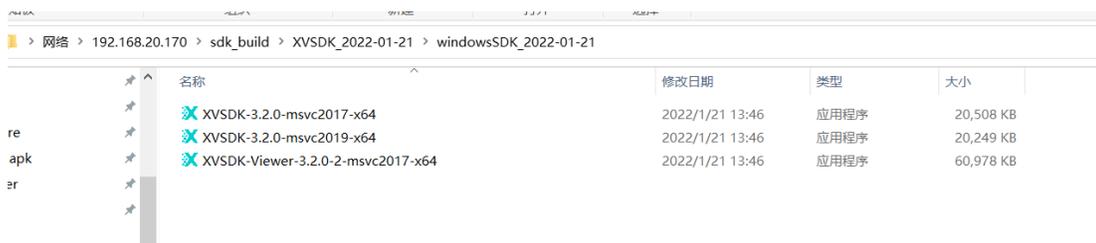


图 3-3 SDK 安装包

2) windows 键打开程序列表后找到 viewer 程序，点击打开 viewer 显示。

3) 找到 SDK 安装目录，一般默认为 C:\Program Files\xv sdk\bin

直接双击打开 all_stream，即可打开所有 camera。

4. Android 平台

1) 同 Daily build 目录下，获取安卓 APK。如下图：

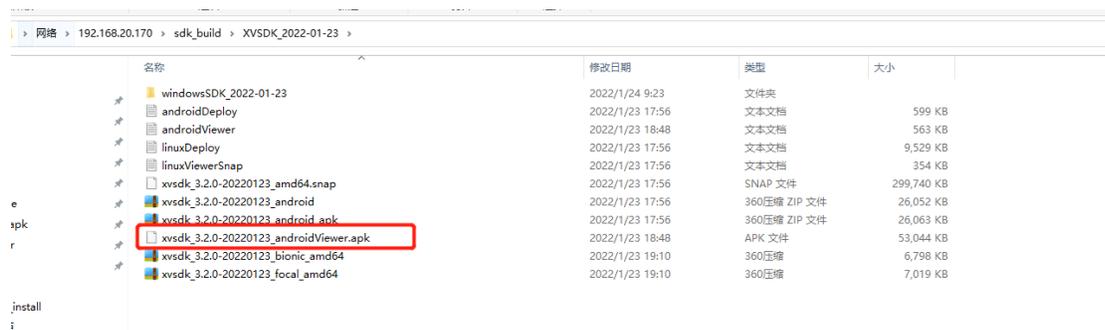


图 4-1 获取安卓 APK

2) 安装至移动安卓设备端：

➤ 方法一：命令安装方法

Step1: 手机直连 PC 后，在弹出的调试界面上点击确认：



图 4-2 点击“确认”

Step2: adb devices 确认手机连接状态正常；

如下红框内的状态为正常连接状态，如出现绿框未认证的状态，返回第一步确认是否授权 USB 调试权限。

```
C:\Users\DELL\Desktop\adb>adb devices
List of devices attached
* daemon not running; starting now at tcp:5037
* daemon started successfully
320595792127    device
```

图 4-3 正常连接状态

```
F:\ADB 1.0.31>adb devices
List of devices attached
0123456789ABCDEF    unauthorized
```

图 4-4 未认证状态

Step3:

设备连接正常后输入：`adb install "C:\Users\DELL\Desktop\xxxx.apk"` 进行安装。

"C:\Users\DELL\Desktop\" 这一段为 demo apk 所在路径，一般简易操作可以直接找到 apk 拖拽进命令框内，路径地址会自动生成，回车安装即可。出现如下图的 success 字样代表已安装成功。

```
C:\Users\DELL\Desktop\adb>adb install "C:\Users\DELL\Desktop\CES Demo\手势演示Demo\Gesture_YX.apk"  
Success
```

图 4-5 安装成功

➤ 方法二：安装包直装

将 apk 复制到手机 sdcard 文件夹内，在手机文件管理器中找到 apk 后直接点击安装。

重要提示 首次安装 apk 后，必须给予 apk 所有权限，可以在不连接 Glass 的状态下，先行打开 Demo apk，并且根据弹出的提示全部点击“是”或者“允许”等选项，给与 apk 权限。

5. 运行 XVSDK-Viewer

1) 连接 glass 与 pc，打开新的终端，输入“xv sdk-viewer”即可打开 xv sdk-viewer，如下图所示：

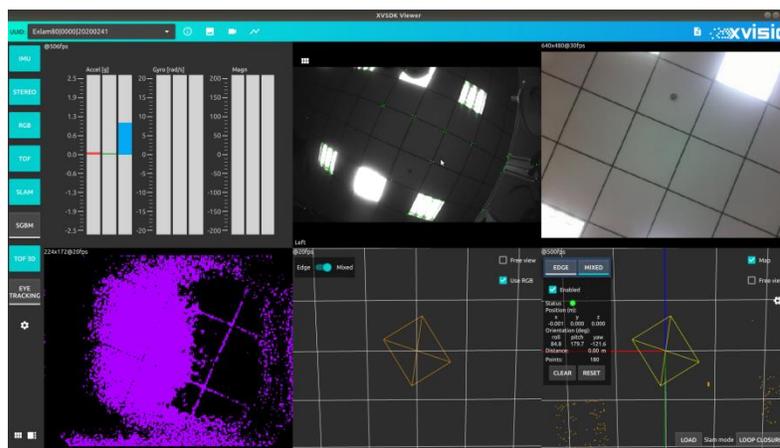


图 5-1 打开 xv sdk-viewer

2) 下图 UUID 后为 glass 的 SN 号，可用于确认 glass 是否注册过手势，未注册手势的 glass 无 SN 号。点击红框中的 info 可以查看版本信息：

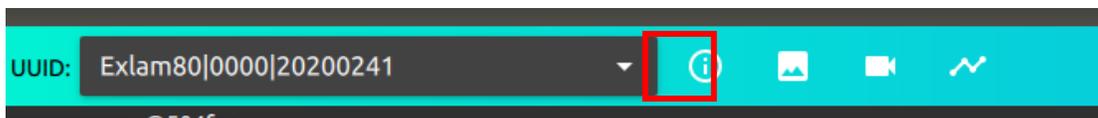


图 5-2 查看 SN

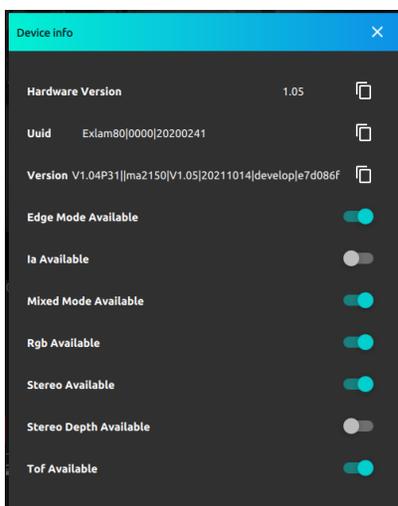


图 5-3 版本信息

3) 下图三组数据分别为轴序 (Accel)；角速度 (Gyro)；地磁 (Magn)

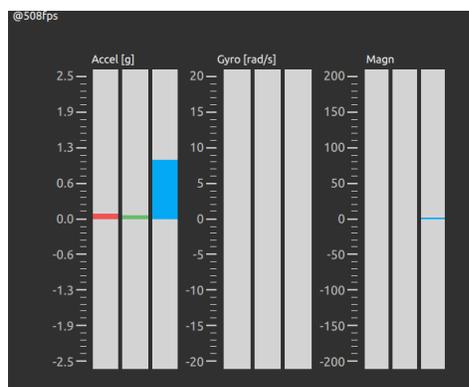


图 5-4 轴序 (Accel)；角速度 (Gyro)；地磁 (Magn)

Accel: glass 水平向前时 Y 轴数据向下, glass 竖直向前时 X 轴数据向下, glass 水平向下时 Z 轴数据向下。

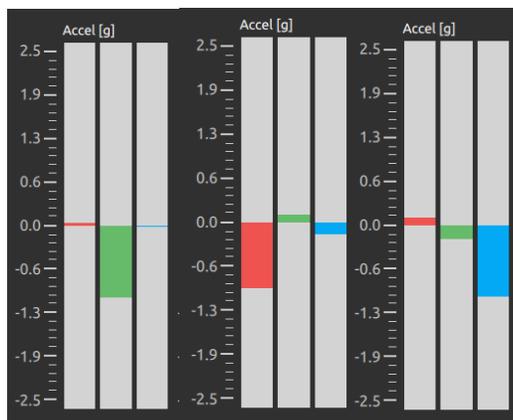


图 5-5 轴序 (Accel)

Gyro: 晃动 glass 时会有数据跳动, 静止时无数据跳动。

Magn: 代表地磁数据。

4) 下图为 fisheye 的出图, 一般情况下默认显示左侧 fisheye 的图片。

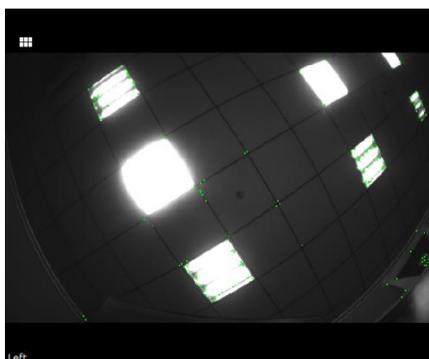


图 5-6 左侧 fisheye

5) 点击左上方的图标, 可令界面同时显示两侧 fisheye 的图像, 点击左侧或右侧的出图可以放大图像。

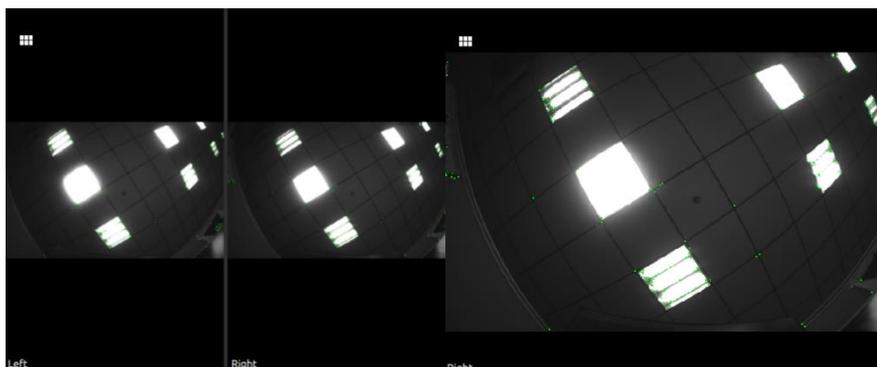


图 5-7 放大图像

6) 点击大图会在右侧出现设置按钮，点击红框上方两个按钮可以对图像进行左右或上下翻转；红框中的按钮可以隐藏和显示特征点；红框下方的按钮可以关闭自动曝光，进行手动调节。

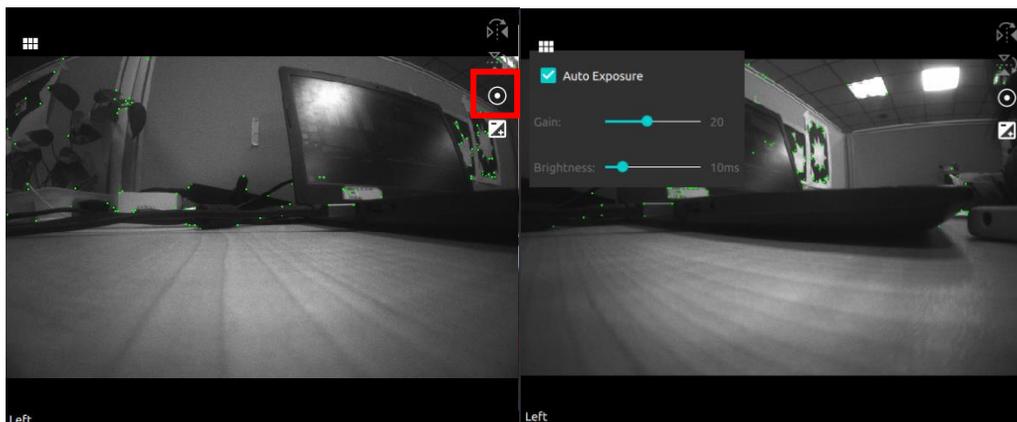


图 5-8 左右或上下翻转图像

7) 下图为 RGB 图像，左上角为 RGB 图像的分辨率与帧率。点击图像时会在右侧出现设置按钮，可通过设置调节 RGB 的分辨率与帧率。

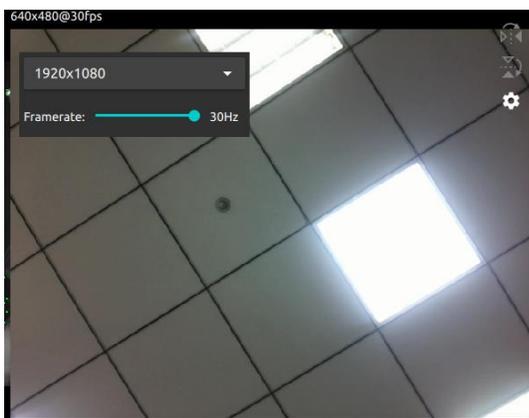


图 5-9 RGB 图像

8) 下图为 TOF 图像，左上角为 TOF 图像的分辨率和帧率。点击图像时会在右侧出现设置按钮，可通过设置调节 TOF 的最大识别距离与帧率。

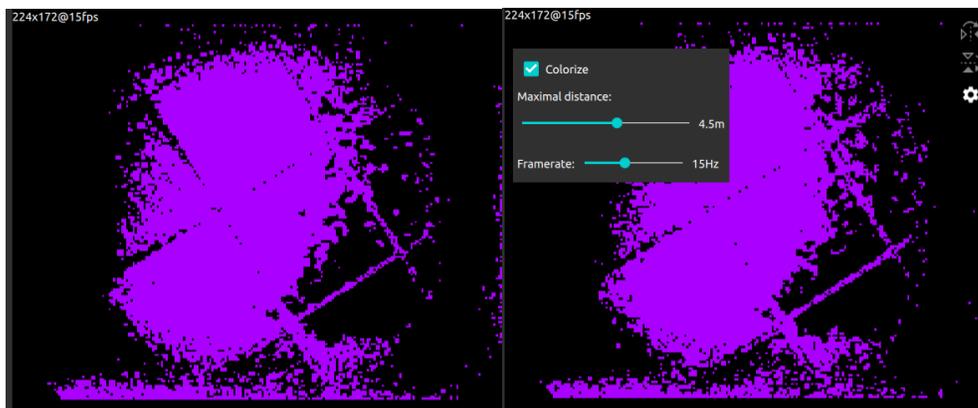


图 5-10 TOF 图像

9) 下图为 SGBM 图像，点开设置后可以调节图片大小、baseline、FOV、Confidence、Mode、Max distance、Min distance 等参数。

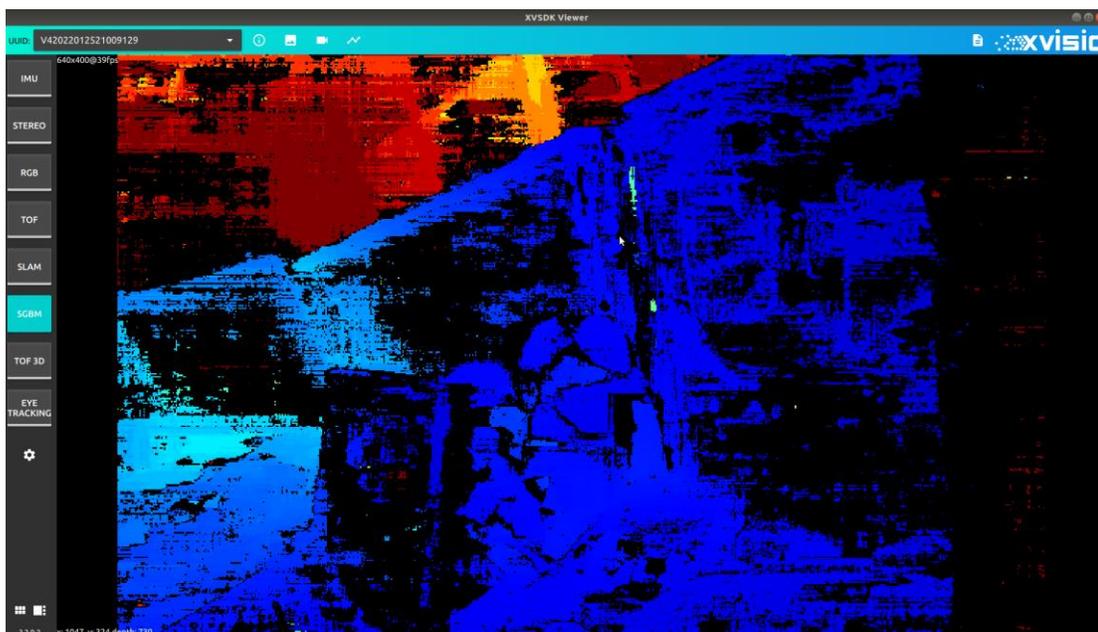


图 5-11 SGBM 图像

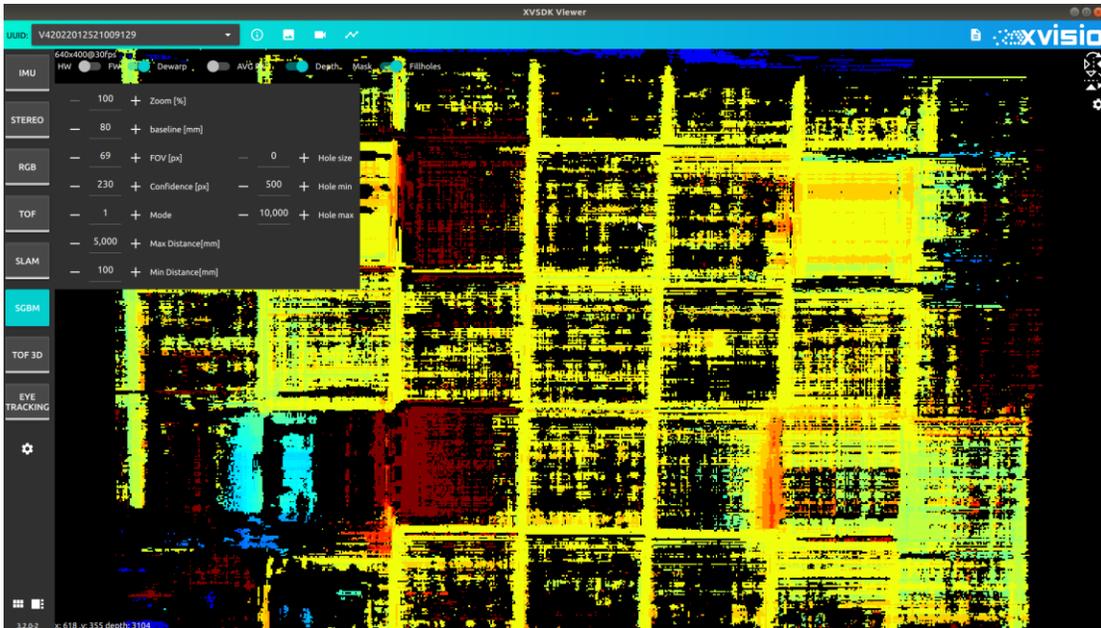


图 5-12 设置参数

10) 下图为 rgbd, 有 edge 和 mixde 两种模式, 可通过左侧红框中的按键切换。取消右侧红框的勾选, 可以从 rgb 切换至 TOF。

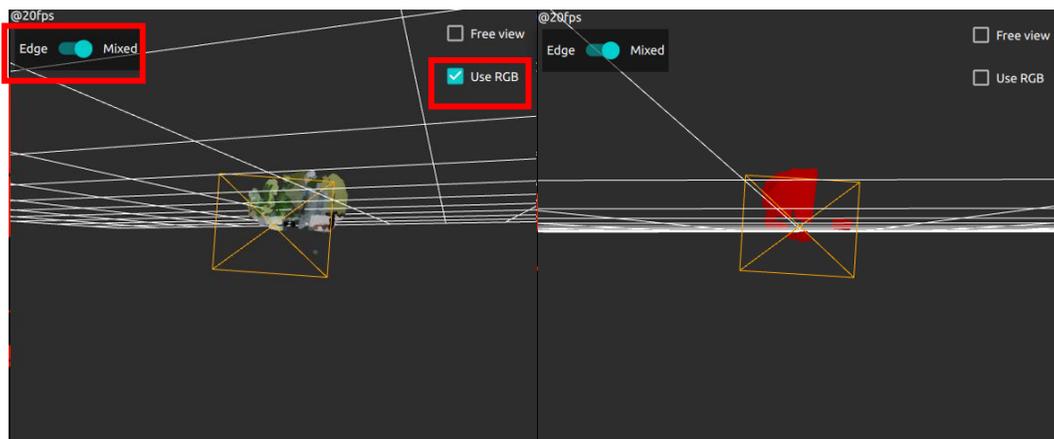


图 5-13 RGBD

11) 下图为 SLAM, 有 edge 和 mixed 两种模式, mixed 模式下可通过右下方红框内的“LOOP CLOSURE”按钮调节至“CSlam mode”, “LOAD”键可以读取地图。

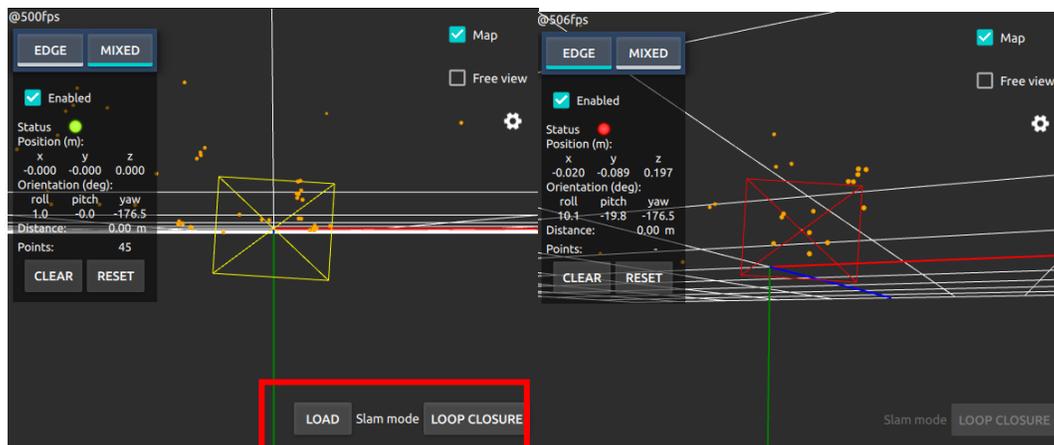


图 5-14 SLAM

移动模组，可以在该区域中看到模组的移动轨迹，左侧方框内的“CLEAR”可以清除移动轨迹，“RESET”可以清除轨迹并令图像回到原点。

12) 点击左侧选项可以隐藏和开启部分界面。

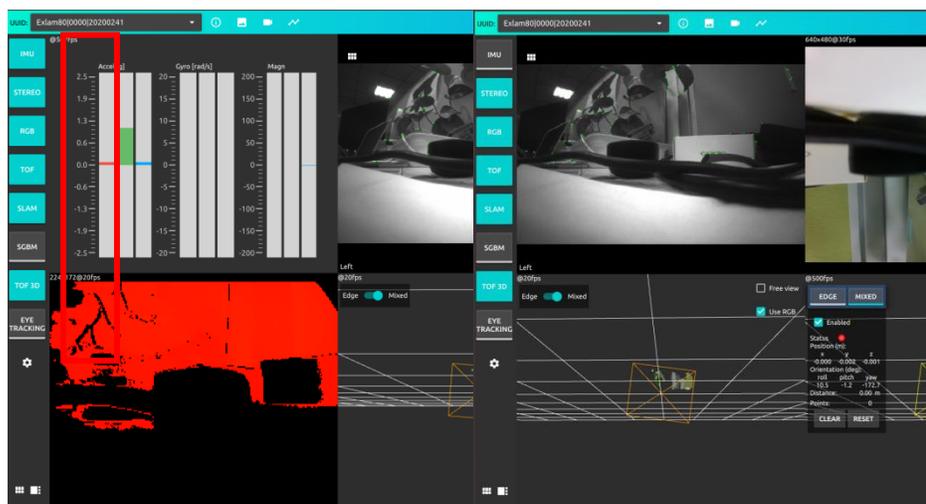


图 5-15 选项

6. ROS 环境搭建

在 ROS 环境搭建之前，需要准备好 ubuntu 版本 sdk 安装包。

6.1. 搭建流程

6.1.1. 基础文件准备

```
sudo apt-get install udev //安装 udev
```

```
cp ~/99-xvisio.rules ~/etc/udev/rules.d/ //若已存在则可跳过，若复制不成功见章节 6.3 解决。
```

```
sudo udevadm control --reload-rules && udevadm trigger //让系统识别usb 设备
```

6.1.2. Some Libs (基础组件安装)

- `sudo apt update`
- `sudo apt install -y lsb-release gnupg git g++ cmake cmake-curses-gui git pkg-config autoconf`
- `sudo apt install -y libtool libudev-dev libjpeg-dev zlib1g-dev libopencv-dev rapidjson-dev`
- `sudo apt install -y libeigen3-dev libboost-thread-dev libboost-filesystem-dev libboost-system-dev`
- `sudo apt install -y libboost-program-options-dev libboost-date-time-dev`

6.1.3. 安装必要的 ROS 包

注意：ubuntu 20.04 需要将 melodic 替换为 noetic。

```
sudo rm /etc/apt/sources.list.d/ros-latest.list //若提示无该文件可删除，跳过继续进行下一步。
```

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key  
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

```
sudo apt update
```

```
sudo apt install -y ros-melodic-desktop-full ros-melodic-ddynamic-reconfigure
```

```
sudo apt install -y python-rosdep python-rosinstall python-rosinstall-generator  
python-wstool build-essential
```

```
sudo rosdep init //如果出现问题见章节 6.3 解决
```

```
rosdep update
```

```
source /opt/ros/melodic/setup.bash
```

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

6.1.4. Init Catkin Workspace

```
mkdir -p ~/catkin_ws/src
```

```
cd ~/catkin_ws/
```

```
catkin_make //在 ros 工作空间环境初始化, 创建几个文件夹
```

```
source ${HOME}/catkin_ws/devel/setup.bash
```

```
echo "source ${HOME}/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

6.1.5. Build xv_sdk

根据 Ubuntu 版本安装 sdk3.2.0 (ubuntu 20.04 需要使用 xvsdk_3.2.0-20220321_bionic_amd64.deb) 。

`xvsdk_3.2.0-20220321_bionic_amd64.deb` // 双击 `xvsdk_3.2.0-20220321_bionic_amd64.deb` 安装或用命令安装, 步骤见章节 6.3.

```
cd ~/catkin_ws/
```

```
cp -r xv_sdk ~/catkin_ws/src/ //将 xv_sdk 文件夹放到 src 文件夹中
```

xv_sdk 文件夹中 include 文件夹, xv_sdk.hpp 文件的 line 38, 选配属性如下:

```
///define NOT_USE_RGB  
define NOT_USE_TOF  
  
///define NOT_USE_SGBM  
  
define NOT_USE_FE  
  
///define USE_MAPPING_ON_HOST
```

```
ifndef NOT_USE_TOF
```

```
 //define TOF_QVGA
```

```
endif*/ifndef NOT_USE_TOF*/
```

```
-----  
rosdep install --from-paths src --ignore-src -r -y //安装工作空间中 ros 包的依赖
```

```
catkin_make -DXVSDK_INCLUDE_DIRS="/usr/include/xv_sdk" -  
DXVSDK_LIBRARIES="/usr/lib/libxv_sdk.so" //安装出错见章节 6.3 解决方案。
```

6.2. 启动 Demo

注意: 后面每次启动 demo 只需运行下面指令即可, 需要开三个终端分别启动。

1) node launch

```
roscore //用于启动 ros master 节点管理器
```

2) Roslaunch (重新打开一个终端)

```
cd ~/catkin_ws/
```

```
roslaunch xv_sdk xv_sdk.launch //启动 ros 关联的所有结点
```

3) run demo (重新打开一个终端)

```
roslaunch rviz rviz -d `rospack find xv_sdk`/rviz/demo.rviz //启动 demo
```

//点击 Color Image 下的 Image Topic 可以选择查看 rgb/tof/fisheye/rgbd 的预览

6.3. Q&A

Q1:在执行 roscore 时可能出现如下错误:

1)IOError:[Errno 13] Permission denied: 'home/[user]/.ros/roscore-11311.pid'

该问题由于该路径下 ros 文件权限造成。

输入: `sudo chmod 777 -R ~/.ros/`

再次启动 ROS: roscore

2)若存在不能正常启动的情况, 可以执行:

```
sudo apt-get install ros-melodic-desktop
```

```
source ~/.bashrc
```

再启动 roscore

Q2: 执行 catkin_make 时若出现一下错误:

1) 通过 CMakeList.txt 找到 No xv_sdk provided, 即 include 和 lib 没识别到。

很可能是先将 xv_sdk 拷贝到 catkin_ws/src 中, 然后再执行 catkin_make, 则会报出上述错误。

解决: 将 xv_sdk 先删掉, 然后执行 catkin_make, 再将 xv_sdk 复制过去即可。

2) 若提示文件不存在，则检测文件是否存在异常。文件操作中操作不当可能会出现文件破损丢失的情况。

Q3: Ubuntu18.04 上安装 `xv sdk_3.2.0-20220321_bionic_amd64.deb`

1) `sudo apt-get update`

2) `sudo apt-get install -y g++ cmake libjpeg-dev zlib1g-dev udev libopencv-core3.2 libopencv-highgui-dev liboctomap1.8 libboost-chrono-dev libboost-thread-dev libboost-filesystem-dev libboost-system-dev libboost-program-options-dev libboost-date-time-dev`

3) `sudo dpkg -i xv sdk_3.2.0-20220321_bionic_amd64.deb`

Q4: 安装 `sudo rosdep init` 时报错: *"The 'rosdep==0.21.0' distribution was not found and is required by the application"*

解决方案如下: 将默认的 `python3` 改成 `python2`

`sudo update-alternatives --config python`

`sudo update-alternatives --list python`

`sudo update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1`

`sudo update-alternatives --list python`

`sudo update-alternatives --config python` 根据提示操作输入 1

`python` //查看默认的是 `python2.7.17`, 接下来继续安装 `sudo rosdep init`

Q5: `99-xvisio.rules` 复制到 `etc/udev/rules.d/` 下失败

解决方法:

1) `sudo gedit /etc/udev/rules.d/99-xvisio.rules` //打开一个文本文件

2) 粘贴以下内容, 保存退出:

```
SUBSYSTEM=="usb", ATTR{idVendor}=="040e", MODE="0666",  
GROUP="plugdev"
```

```
SUBSYSTEM=="usb", ATTR{idVendor}=="0e8d", MODE="0666",  
GROUP="plugdev"
```

```
SUBSYSTEM=="usb", ATTR{idVendor}=="05c6", MODE="0666",  
GROUP="plugdev"
```

```
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", MODE="0666",  
GROUP="plugdev"
```

```
SUBSYSTEM=="usb", ATTR{idVendor}=="22d9", MODE="0666",  
GROUP="plugdev"
```

```
SUBSYSTEM=="usb", ATTR{idVendor}=="19d2", MODE="0666",  
GROUP="plugdev"
```

3) `ls /etc/udev/rules.d/` 查看 99-xvisio.rules 是否存在

4) 查看内容是否写入成功

```
cd /etc/udev/rules.d
```

```
cat 99-xvisio.rules
```

Q6: 卸载 ROS 方法:

```
sudo apt-get purge ros-*
```

6.4. python_wrapper 环境搭建

1) 双击运行 "XVSDK-3.2.0-msvc2019-x64.exe to install xvsdk" , 点击 "下一步" :

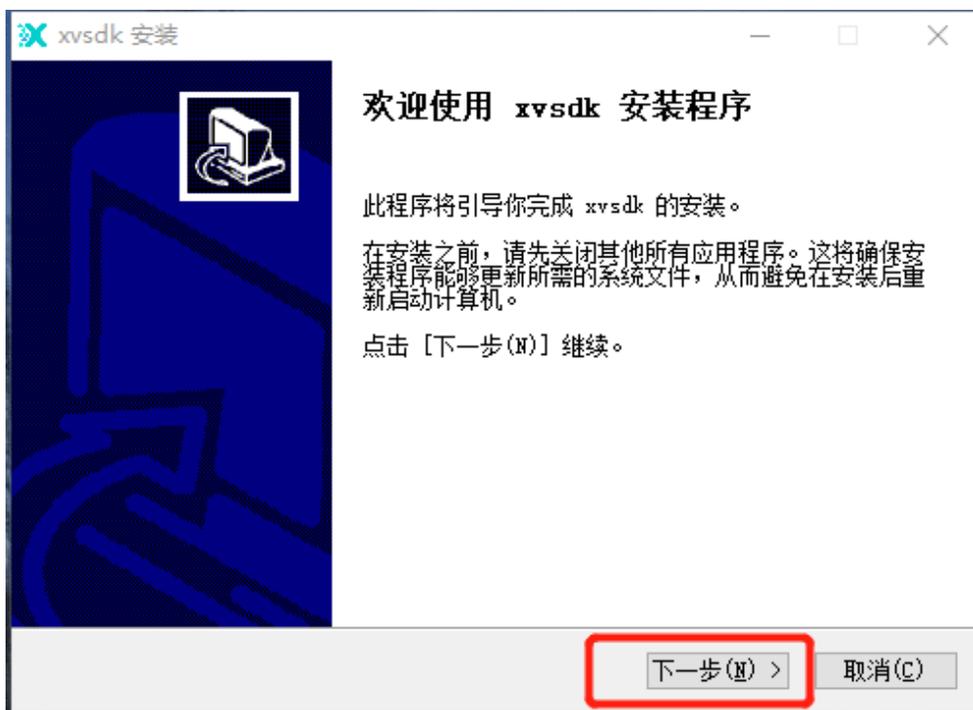


图 6-1 点击“下一步”

2) 点击“我接受”：



图 6-2 点击“我接受”

3) 选择安装路径后点击“下一步”：

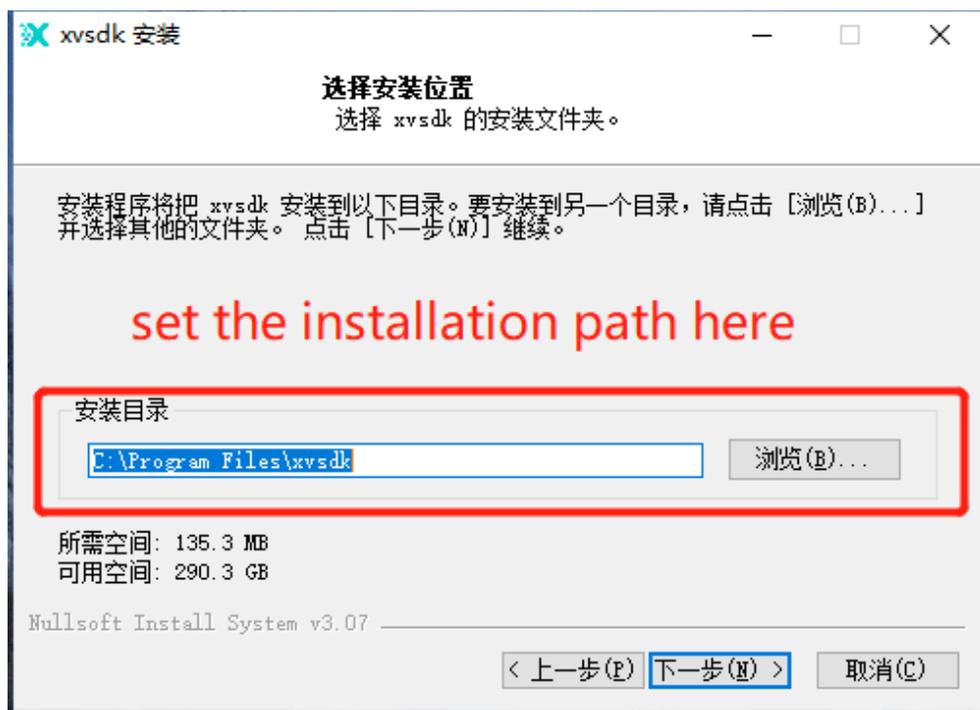


图 6-3 选择安装路径

4) 点击“下一步”：

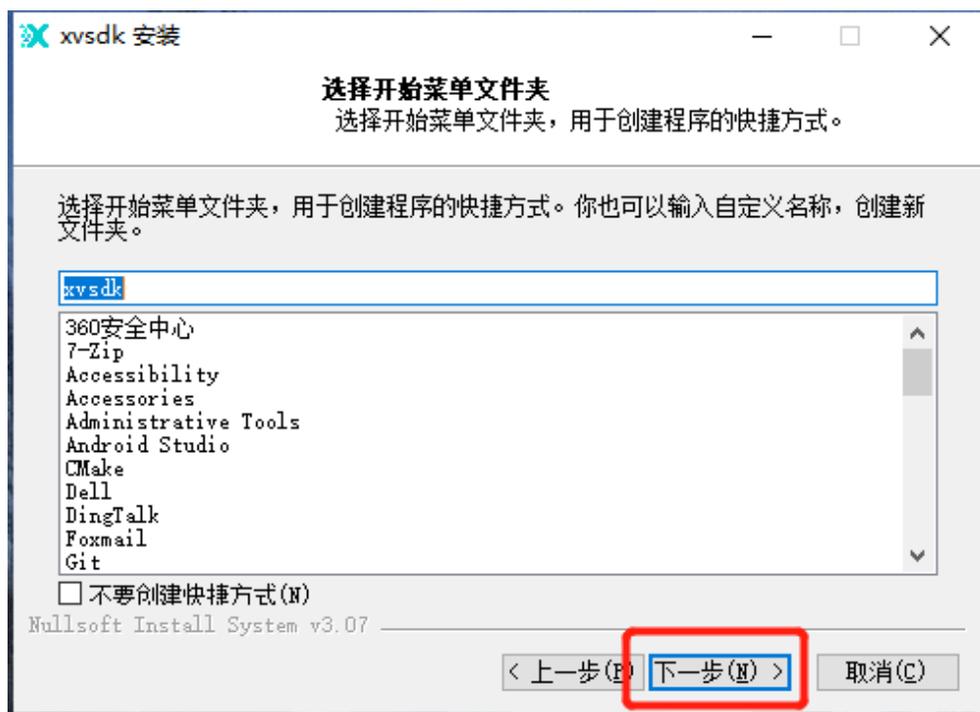


图 6-4 点击 “下一步”

5) 选择 “python_wrapper” ， 点击 “安装” ：

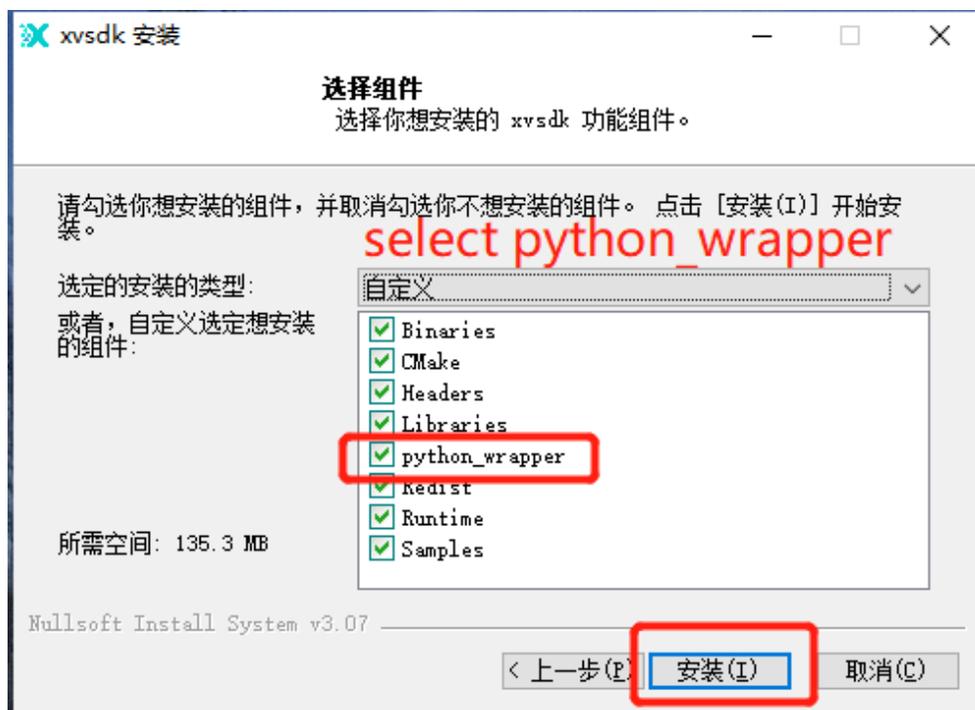


图 6-5 点击 “安装”

6) 安装后，在\bin 的目录下可以找到 “python-wrapper” 文件夹：

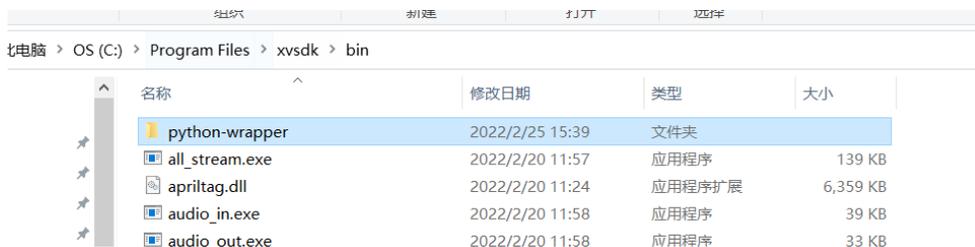


图 6-6 “python-wrapper” 文件夹

7) 双击打开 “python-wrapper” 文件夹， 点击运行 “python PythonDemo.py” 。