

## 1.Files

- (1).lib----- The directory of library.
- (2).include----- The directory of header files.
- (3).bin----- The directory of binary files.
- (4).SampleClient.cpp-----The source code of client receiving program for reference.
- (5).CMakeLists.txt-----The script of program build.

## 2.Demo test

- (1).Set the PC IP address to 10.1.1.198 and subnet mask to 255.255.255.0. Disable the firewall and network blocking software.
- (2).Run XINGYING motion capture software as an administrator.
- (3).Select the data broadcast interface, select “NIC Address”, and then check “Setting”,“SDK Enabled”.XINGYING sends motion capture data, the local and other computers can receive data
- (4).In live mode, click run;Or, in post-process mode, play back data.
- (5). On Linux, copy dynamic library to library directory:  

```
sudo cp libnokov_sdk.so /usr/lib
```
- (6). Perform:  

```
sudo ./SampleClient
```

  
Enter XINGYING broadcast IP to receive SDK data.

Note: In post-process mode, you need to close the client receiver before switching. If the connection fails, check the destination NIC name in “SampleClient.cpp” and ensure that the destination NIC Address is in the same network segment. You may need to disable Wireless Network Connection on some PC.

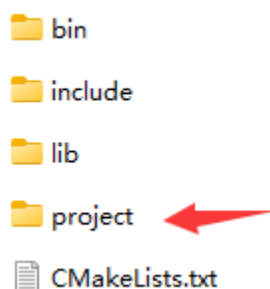
## 3.Version support

- a) XINGYING 1.0.0.2910,2.x,3.x
- b) XING 1.0.0.2909
- c) CONG 1.2.0.2912

## 4.cmake build

Windows:

- 1. Create a new folder in the same level directory as CMakeLists.txt to store project files, as shown in the example: project



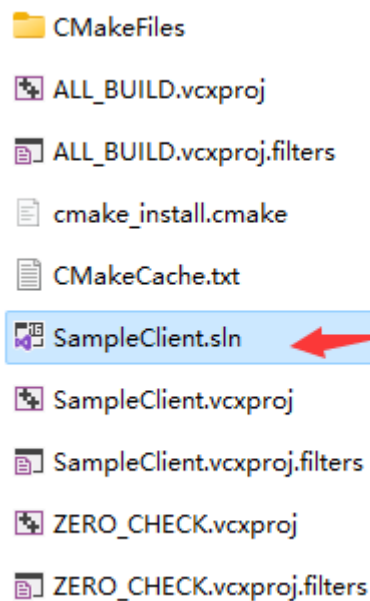
- 2. Open a command line window and enter the ‘project’ directory

```
C:\Windows\System32\cmd.e X + v
E:\temp\【XING_Windows_SDKDemo】2.4.0.5329\project>
```

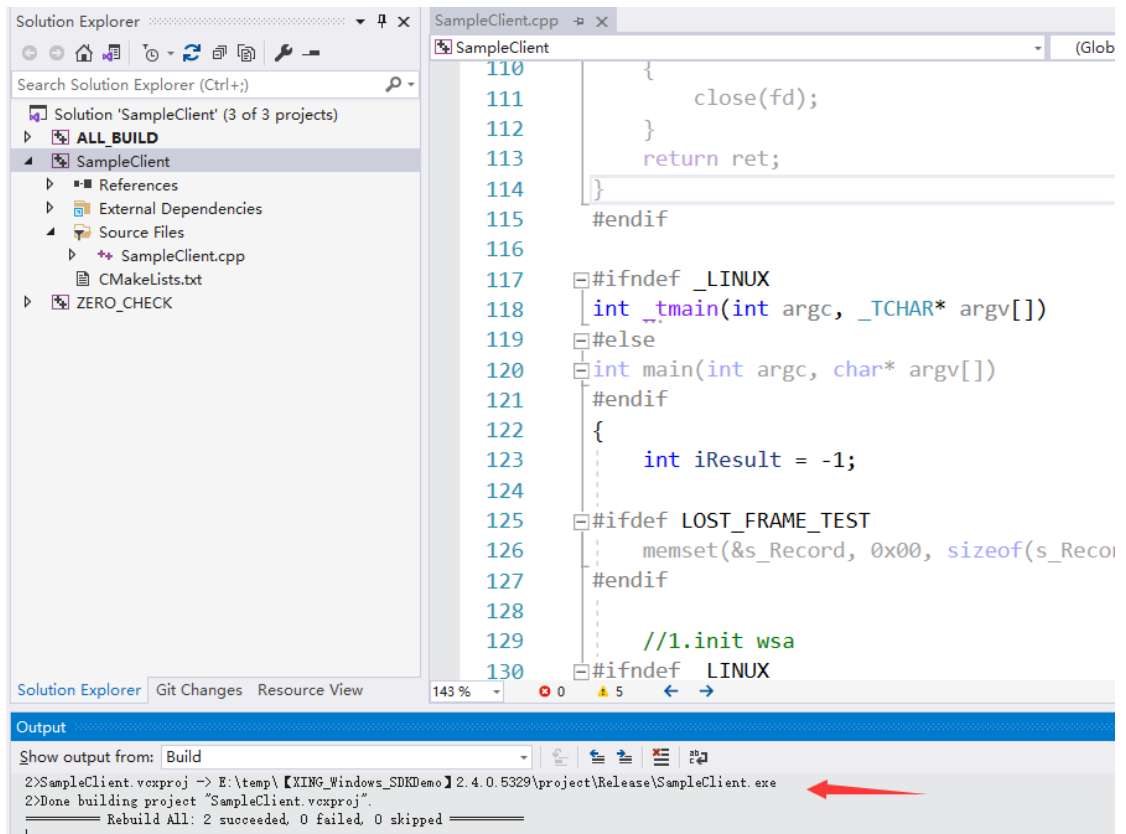
3. Run the command cmake to generate the project file. If there is an error here, please check if there are any illegal characters in the path. Do not use any special symbols, especially symbols, in the path &

```
E:\temp\【XING_Windows_SDKDemo】2.4.0.5329\project>cmake ..
-- Building for: Visual Studio 16 2019
-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.22631.
-- The C compiler identification is MSVC 19.29.30152.0
-- The CXX compiler identification is MSVC 19.29.30152.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.29.30133/bin/Hostx64/x64/cl.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.29.30133/bin/Hostx64/x64/cl.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Build type is
-- Configuring done (2.2s)
-- Generating done (0.0s)
-- Build files have been written to: E:/temp/【XING_Windows_SDKDemo】2.4.0.5329/project
```

4. In File Explorer, double-click to open the project file



5. Directly compile and generate executable files



6. The executable file is located in the Debug or Release directory of the "project" directory

CMakeFiles	2024/1/9 11:03
Debug	2024/1/9 11:02
Release	2024/1/9 11:03
SampleClient.dir	2024/1/9 11:02
x64	2024/1/9 11:02
ALL_BUILD.vcxproj	2024/1/9 10:21
ALL_BUILD.vcxproj.filters	2024/1/9 10:21
cmake_install.cmake	2024/1/9 10:21
CMakeCache.txt	2024/1/9 10:21
SampleClient.sln	2024/1/9 10:21
SampleClient.vcxproj	2024/1/9 10:21
SampleClient.vcxproj.filters	2024/1/9 10:21
ZERO_CHECK.vcxproj	2024/1/9 10:21
ZERO_CHECK.vcxproj.filters	2024/1/9 10:21

Linux:

1. In the same level directory as CMakeLists.txt, create a new folder to store project files. In the example, create:: project

```
root@ubuntu:~/nokov/XING_Linux_SDK_2.4.0.5329# pwd
/root/nokov/XING_Linux_SDK_2.4.0.5329
root@ubuntu:~/nokov/XING_Linux_SDK_2.4.0.5329# mkdir -p project
root@ubuntu:~/nokov/XING_Linux_SDK_2.4.0.5329#
```

2. Run the command cmake, A Makefile will be generated. If there is an error here, please check if there are any illegal characters in the path and do not use special symbols

```
root@ubuntu:~/nokov/XING_Linux_SDK_2.4.0.5329/project# cmake ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Build type is Release
-- Configuring done
-- Generating done
-- Build files have been written to: /root/nokov/XING_Linux_SDK_2.4.0.5329/project
root@ubuntu:~/nokov/XING_Linux_SDK_2.4.0.5329/project# ls
CMakeCache.txt CMakeFiles cmake_install.cmake Makefile
```

3. Run the make command to generate an executable file

```
[100%] Linking CXX executable SampleClient
[100%] Built target SampleClient
root@ubuntu:~/nokov/XING_Linux_SDK_2.4.0.5329/project# ls
CMakeCache.txt CMakeFiles cmake_install.cmake Makefile SampleClient
```