libsynexens4 SDK Instructions for using multiple machine v1.6

Revise the historical version				
Date	SDK edition	Document version	Description	Author
202212114	v4.0.0.0	v1.0	initial version	YSY
20230424	v4.0.1.0	v1.1	Release version	YSY
20230713	v4.0.2.0	v1.2	Add new supported devices	YSY
20230815	v4.0.3.0	v1.3	New interface	YSY
20230906	v4.1.0.0	v1.4	Update the calling process, and filter the instructions	YSY

20240206	v4.1.2.0	v1.5	New interface	YSY
20240229	v4.1.3.0	v1.6	Update filtering	YSY

catalogue

1. summary 7

2. Environmental configuration 7

2.1. Ubuntu Environment configuration (take Cmake for example) ... 7

2.1.1. Install dependencies 7

2.1.2. Writing a CmakeLists.txt requires a familiarity with CMake8

2.1.3. Create a project compilation file 9

2.1.4. make compile 9

2.1.5. Execute the executable test effect 10

2.2. Windows Environment configuration (take vs2022, for example)11

2.2.1. Create VS engineering 11

2.2.2. Select the solution and the platform corresponding to the SDK 12

2.2.3. Configure the header file path, library path for sdk in the project properties 12
2.2.4. After completing the configuration, you can enter the project for development. If you need to run the demo, just copy the demo code to run it 14
2.3. The SDK must call the process 15
3. API summary 15
3.1. Global interface 15
3.1.1. GetSDKVersion 15
3.1.2. InitSDK 16
3.1.3. UnInitSDK 16
3.1.4. RegisterErrorObserver 16
3.1.5. RegisterEventObserver 17
3.1.6. RegisterFrameObserver 17
3.1.7. UnRegisterErrorObserver 18
3.1.8. UnRegisterEventObserver 18
3.1.9. UnRegisterFrameObserver 19
3.1.10. FindDevice 19
3.1.11. OpenDevice 20
3.1.12. CloseDevice 20
3.1.13. QueryDeviceSupportFrameType 21
3.1.14. QueryDeviceSupportResolution 22

3.1.15. GetCurrentStreamType 22

3.1.16. StartStreaming 23

3.1.17. StopStreaming 23

3.1.18. ChangeStreaming 24

3.1.19. SetFrameResolution 24

3.1.20. GetFrameResolution 25

3.1.21. GetFilter 25

3.1.22. SetFilter 26

3.1.23. GetFilterList 26

3.1.24. SetDefaultFilter 27

3.1.25. AddFilter 27

3.1.26. DeleteFilter 28

3.1.27. ClearFilter 28

3.1.28. SetFilterParam 29

3.1.29. GetFilterParam 29

3.1.30. GetMirror 30

3.1.31. SetMirror 30

3.1.32. GetFlip 31

3.1.33. SetFlip 32

3.1.34. GetIntegralTime 32

3.1.35. SetIntegralTime 33

3.1.36. GetIntegralTimeRange 33

3.1.37. GetDistanceMeasureRange 34

3.1.38. GetDistanceUserRange 34

3.1.39. SetDistanceUserRange 35

3.1.40. GetDeviceSN 35

3.1.41. SetDeviceSN 36

3.1.42. GetDeviceHWVersion 36

3.1.43. GetDepthColor 37

3.1.44. GetDepthPointCloud 38

3.1.45. GetRGBD 39

3.1.46. GetLastFrameData 40

3.1.47. Undistort 40

3.1.48. GetIntric 41

3.1.49. GetTrailFilter 42

3.1.50. SetTrailFilter 42

3.2. Return to the Parameter description 43

4. Filter setting instructions 43

4.1. Filter Parameter setting instructions 43

4.2. Description of the filtered Parameter range 45

4.3. The filter call sequence is indicated 45

5. Description of the data structure definition 46

5.1. Error code 46

5.2. Equipment type 48

5.3. Data flow type 49

5.4. 分辨率枚举 50

5.5. Data frame type 51

5.6. Support type 51

5.7. Event type 52

5.8. Filter type 52

5.9. Equipment information 54

5.10. 事件信息 54

5.11. Data frame information 54

5.12. Data frame； data frames 55

5.13. Point cloud data structure 56

5.14. Camera Parameter construct 56

6. FQA 57

f： DLL not found while running under win 57

f： The Linux runtime prompts uvc _ open: -3 57

f： A select() timeout. error has occurred 57

f： The noise point is relatively large 58

f： The xxx library was not found 58

f： The cs40 cs20-p could not find the Equipment 58

f: cs40 cs20p When connecting to multiple Equipment's, you can only find one or find none 58

7. About the Equipment connection 59

Disclaimer 60

1. summary

Support equipment: cs20 single-frequency cs20 dual-frequency cs30 single-frequency cs30 dual-frequency cs20-p cs40

Support system: windows ubuntu20.04 armv7 armv8

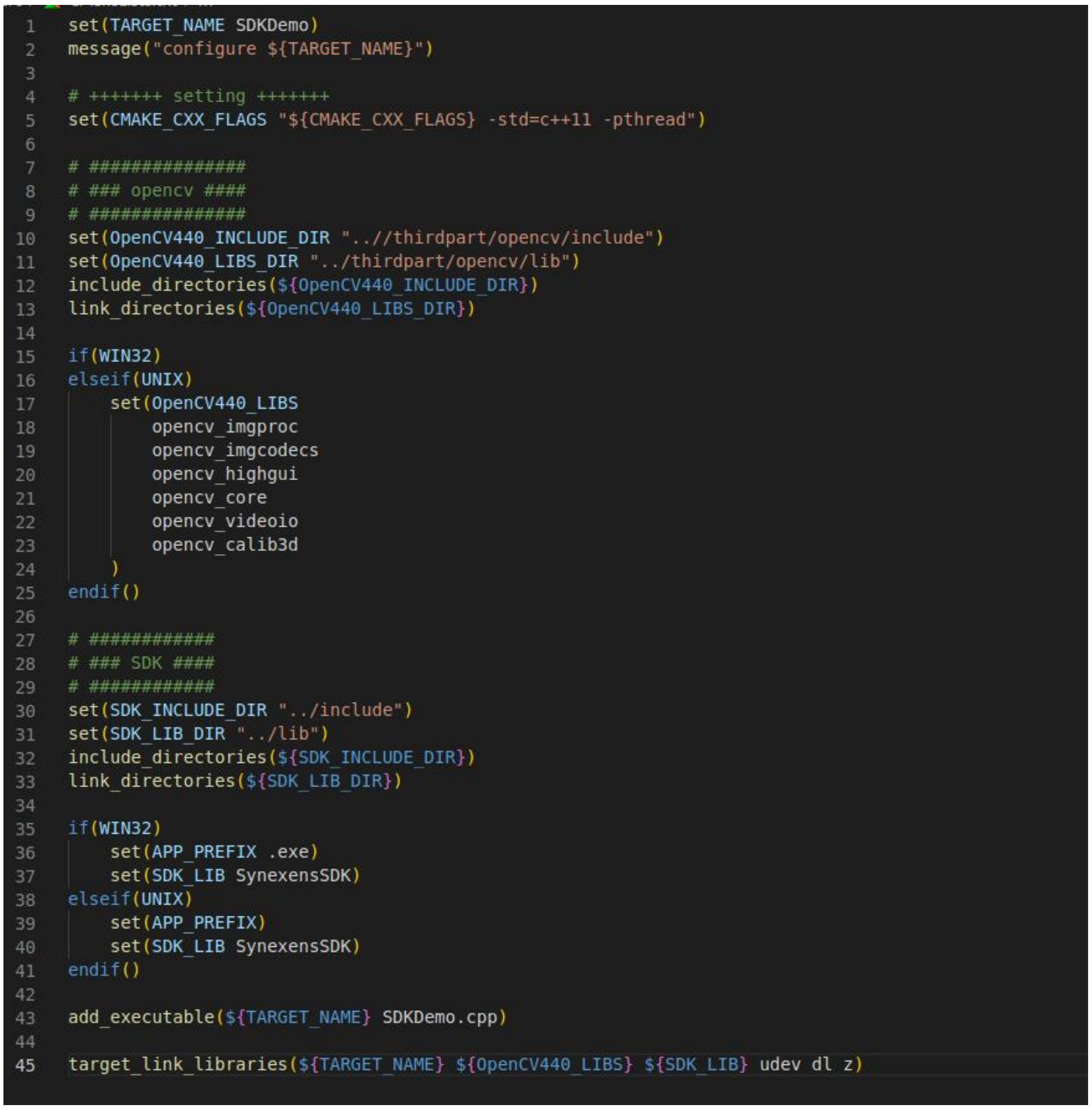
2. Environmental configuration

2.1. Ubuntu Environment configuration (take Cmake for example)

2.1.1. Install dependencies

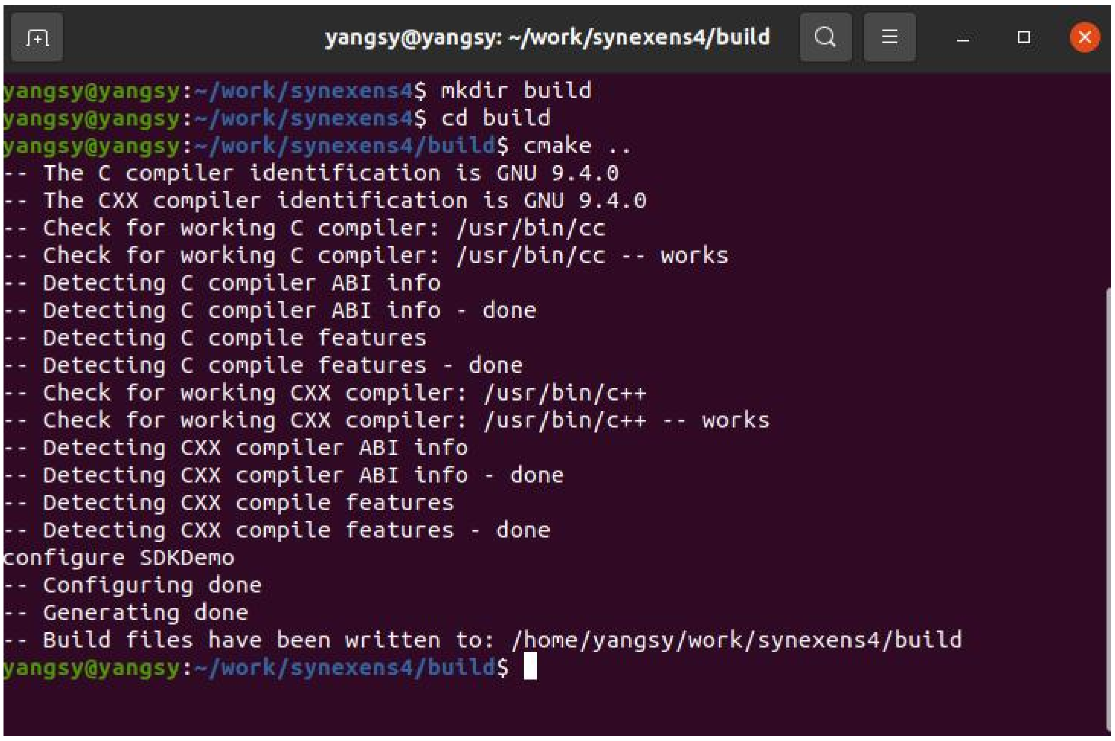
```
sudo apt install libudev-dev

sudo apt install zlib1g-dev
```

2.1.2. Writing a CmakeLists.txt requires a familiarity with CMake



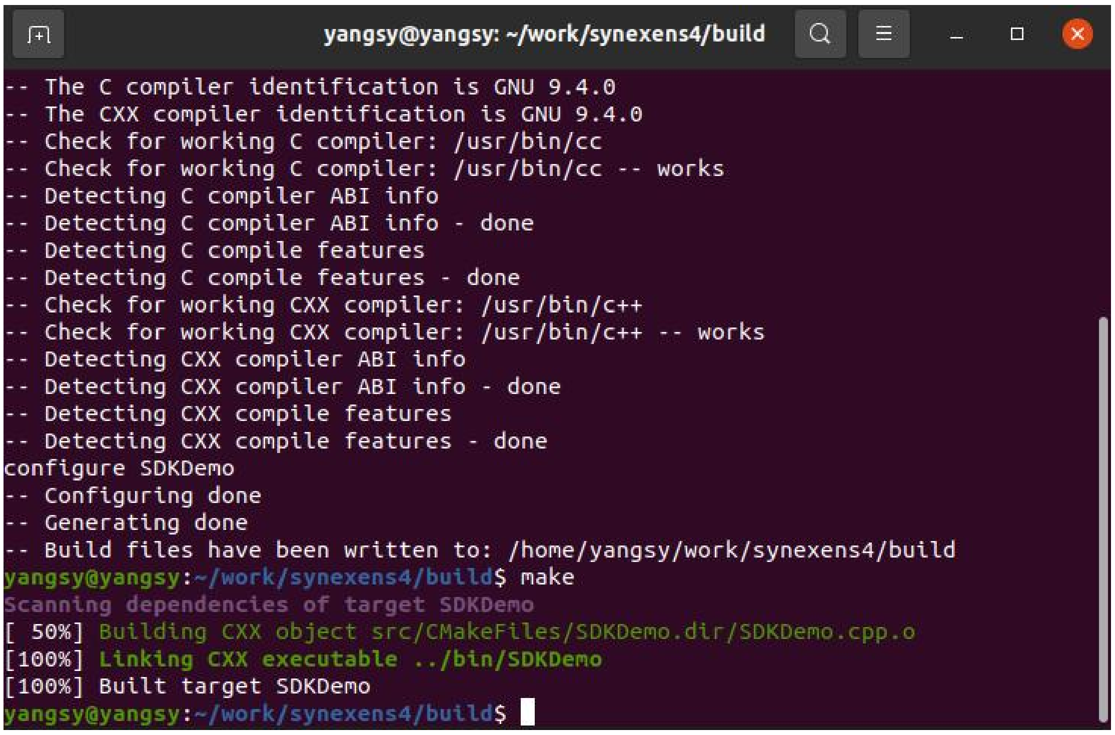
```
set(TARGET_NAME SDKDemo)
message("configure ${TARGET_NAME}")

# +++++++ setting +++++++
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11 -pthread")

# ###############
# ### opencv ####
# ###############
set(OpenCV440_INCLUDE_DIR "..//thirdpart/opencv/include")
set(OpenCV440_LIBS_DIR "../thirdpart/opencv/lib")
include_directories(${OpenCV440_INCLUDE_DIR})
link_directories(${OpenCV440_LIBS_DIR})

if(WIN32)
elseif(UNIX)
    set(OpenCV440_LIBS
        opencv_imgproc
        opencv_imgcodecs
        opencv_highgui
        opencv_core
        opencv_videoio
        opencv_calib3d
    )
endif()

# ############
# ### SDK ####
# ############
set(SDK_INCLUDE_DIR "../include")
set(SDK_LIB_DIR "../lib")
include_directories(${SDK_INCLUDE_DIR})
link_directories(${SDK_LIB_DIR})

if(WIN32)
    set(APP_PREFIX .exe)
    set(SDK_LIB SynexensSDK)
elseif(UNIX)
    set(APP_PREFIX)
    set(SDK_LIB SynexensSDK)
endif()

add_executable(${TARGET_NAME} SDKDemo.cpp)

target_link_libraries(${TARGET_NAME} ${OpenCV440_LIBS} ${SDK_LIB} udev dl z)
```

2.1.3. Create a project compilation file


yangsy@yangsy: ~/work/synexens4/build

```
yangsy@yangsy:~/work/synexens4$ mkdir build
yangsy@yangsy:~/work/synexens4$ cd build
yangsy@yangsy:~/work/synexens4/build$ 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
configure SDKDemo
-- Configuring done
-- Generating done
-- Build files have been written to: /home/yangsy/work/synexens4/build
yangsy@yangsy:~/work/synexens4/build$
```

2.1.4. make compile

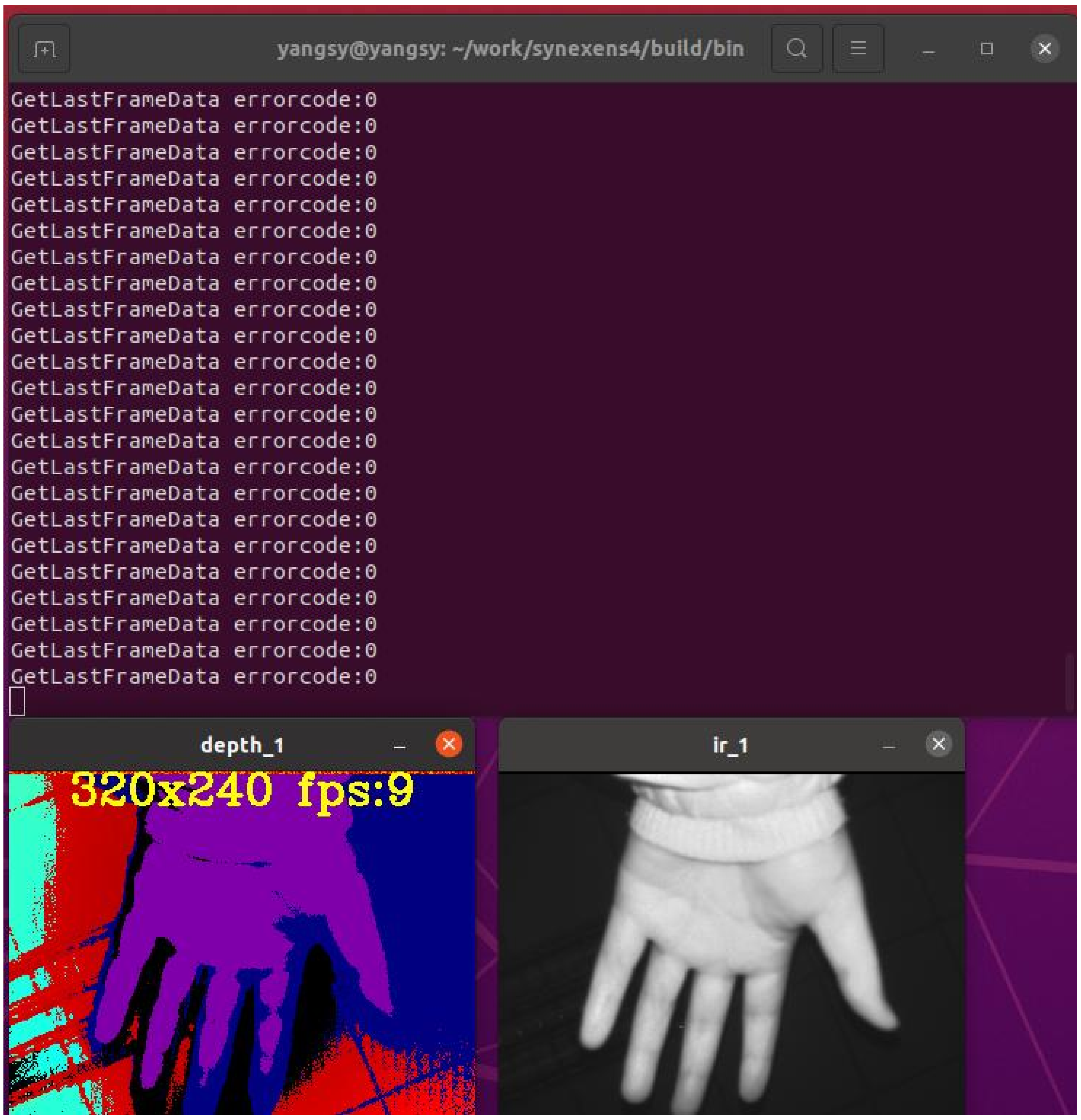

yangsy@yangsy: ~/work/synexens4/build

```
-- 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
configure SDKDemo
-- Configuring done
-- Generating done
-- Build files have been written to: /home/yangsy/work/synexens4/build
yangsy@yangsy:~/work/synexens4/build$ make
Scanning dependencies of target SDKDemo
[ 50%] Building CXX object src/CMakeFiles/SDKDemo.dir/SDKDemo.cpp.o
[100%] Linking CXX executable ../bin/SDKDemo
[100%] Built target SDKDemo
yangsy@yangsy:~/work/synexens4/build$
```

2.1.5. Execute the executable test effect

LD _ LIBRARY _ PATH should be configured before executing the program to find the library file that the program depends on. Example run.sh script is written to facilitate the execution of the program.

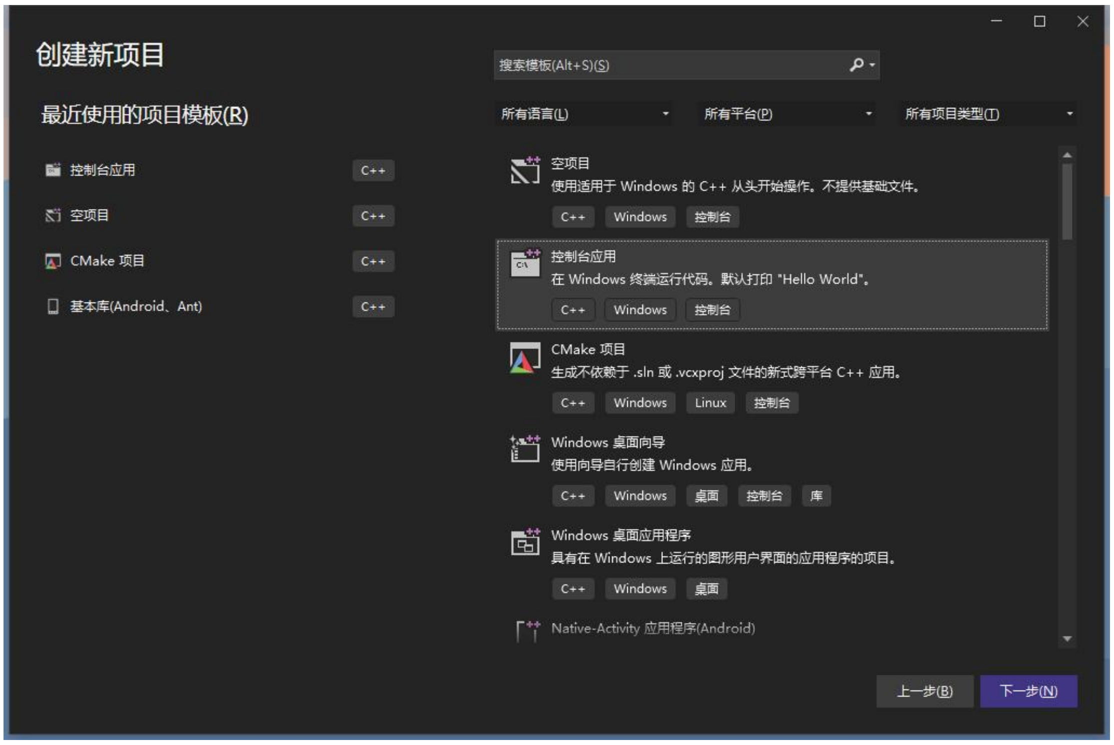
```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:`pwd`
```



```
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
GetLastFrameData errorcode:0
```

2.2. Windows Environment configuration (take vs2022, for example)

2.2.1. Create VS engineering



空项目
使用适用于 Windows 的 C++ 从头开始操作。不提供基础文件。
C++ Windows 控制台

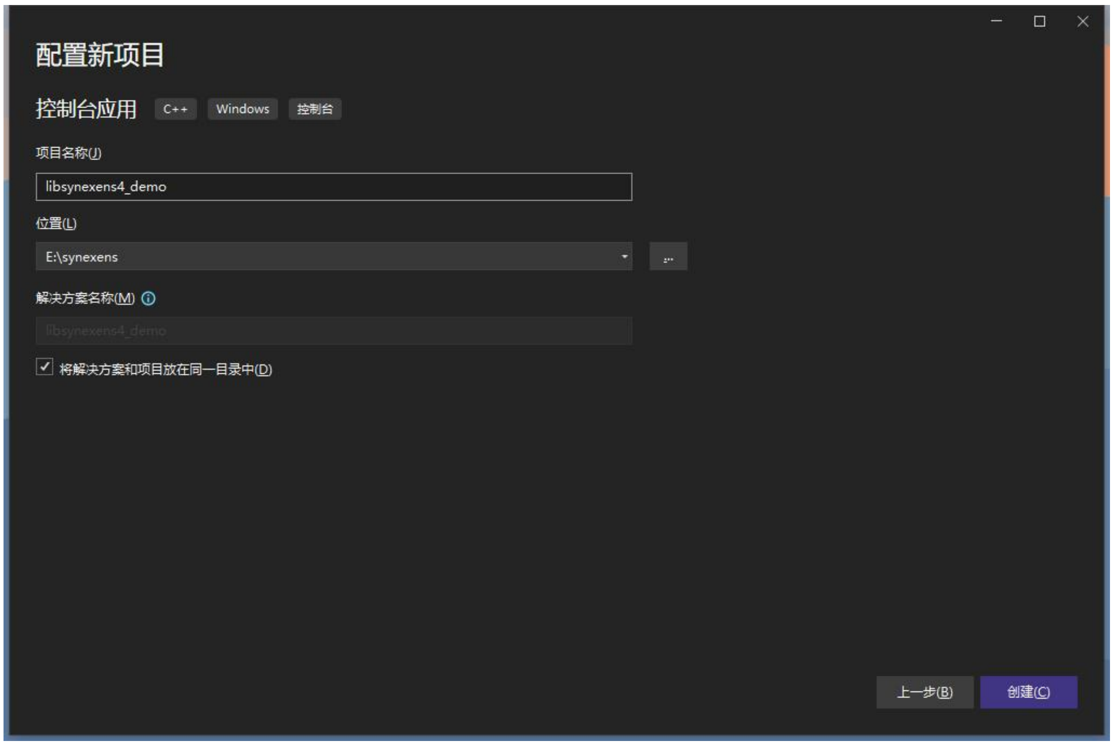
控制台应用
在 Windows 终端运行代码。默认打印 "Hello World"。
C++ Windows 控制台

CMake 项目
生成不依赖于 .sln 或 .vcxproj 文件的新式跨平台 C++ 应用。
C++ Windows Linux 控制台

Windows 桌面向导
使用向导自行创建 Windows 应用。
C++ Windows 桌面 控制台 库

Windows 桌面应用程序
具有在 Windows 上运行的图形用户界面的应用程序的项目。
C++ Windows 桌面

Native-Activity 应用程序(Android)



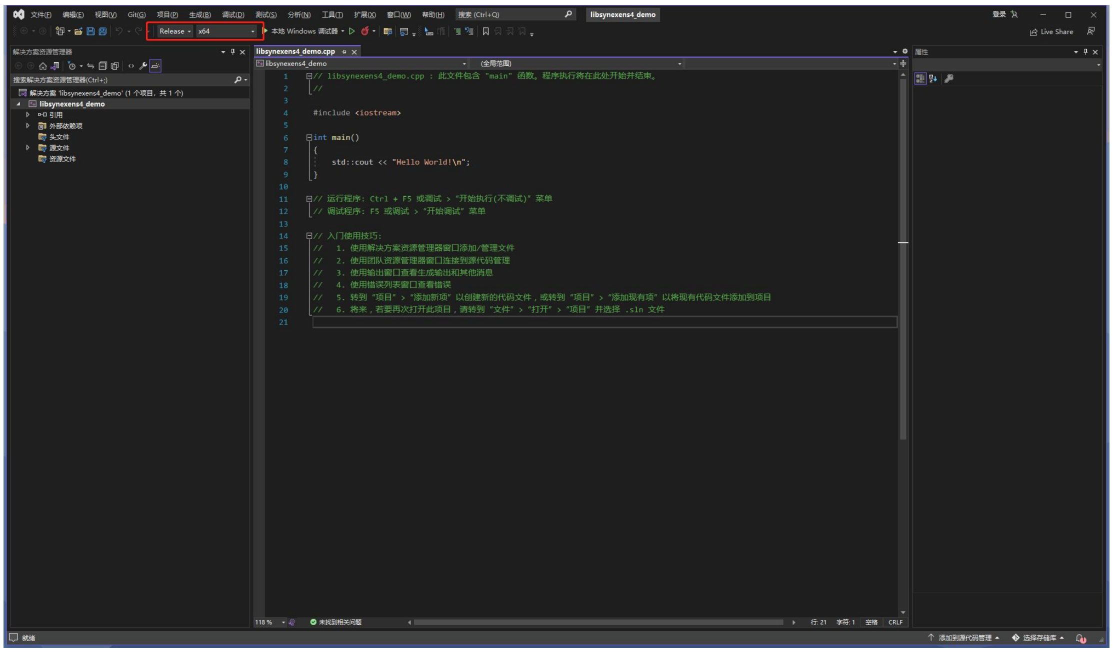
项目名称(J)
libsynexens4_demo

位置(L)
E:\synexens

解决方案名称(M)
libsynexens4_demo

[✓] 将解决方案和项目放在同一目录中(D)

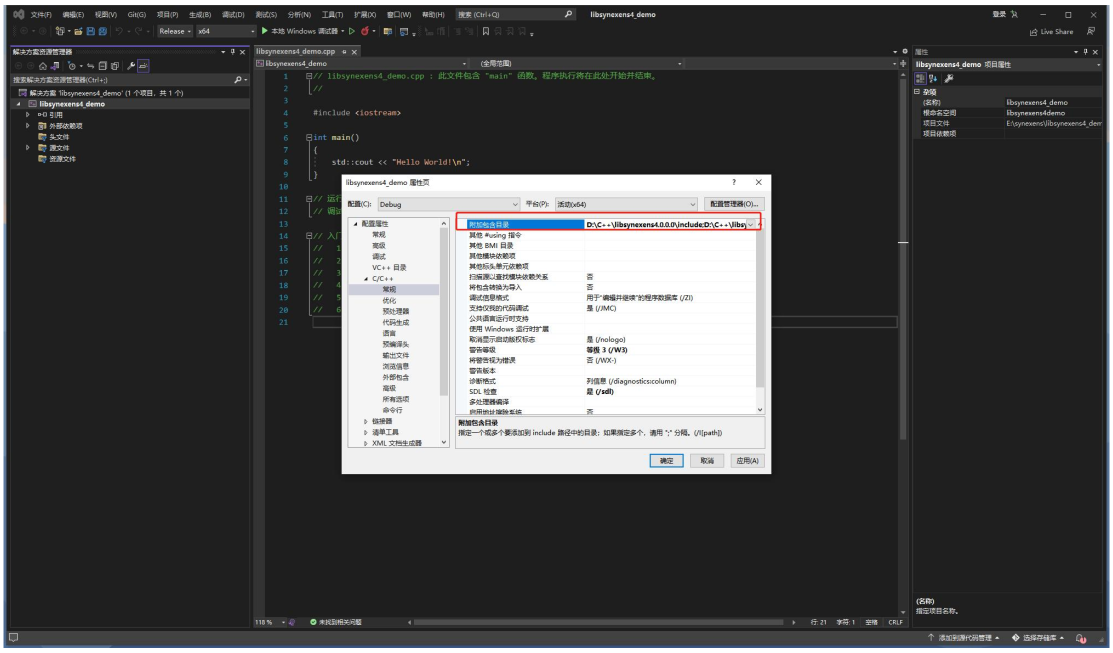
2.2.2. Select the solution and the platform corresponding to the SDK



Release x64

```
// libsynexens4_demo.cpp : 此文件包含 "main" 函数。程序执行将在此处开始并结束。
//

#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
}

// 运行程序: Ctrl + F5 或调试 > “开始执行(不调试)” 菜单
// 调试程序: F5 或调试 > “开始调试” 菜单

// 入门使用技巧:
//   1. 使用解决方案资源管理器窗口添加/管理文件
//   2. 使用团队资源管理器窗口连接到源代码管理
//   3. 使用输出窗口查看生成输出和其他消息
//   4. 使用错误列表窗口查看错误
//   5. 转到 “项目” > “添加新项” 以创建新的代码文件，或转到 “项目” > “添加现有项” 以将现有代码文件添加到项目
//   6. 将来，若要再次打开此项目，请转到 “文件” > “打开” > “项目” 并选择 .sln 文件
```

2.2.3. Configure the header file path, library path for sdk in the project properties

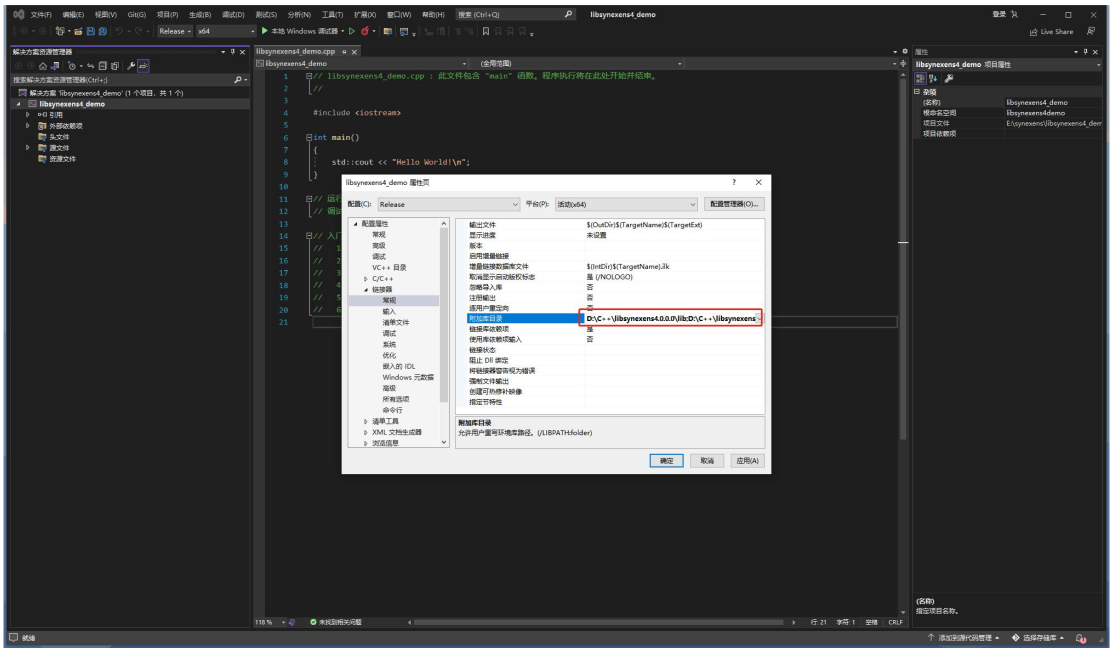


libsynexens4_demo 属性页

配置(C): Debug　平台(P): 活动(x64)

附加包含目录	D:\C++\libsynexens4.0.0.0\include;D:\C++\libsy
其他 #using 指令	
其他 BMI 目录	
其他模块依赖项	
其他标头单元依赖项	
扫描源以查找模块依赖关系	否
将包含转换为导入	否
调试信息格式	用于“编辑并继续”的程序数据库 (/ZI)
支持仅我的代码调试	是 (/JMC)
公共语言运行时支持	
使用 Windows 运行时扩展	
取消显示启动版权标志	是 (/nologo)
警告等级	等级 3 (/W3)
将警告视为错误	否 (/WX-)
警告版本	
诊断格式	列信息 (/diagnostics:column)
SDL 检查	是 (/sdl)
多处理器编译	
启用地址擦除系统	否

附加包含目录
指定一个或多个要添加到 include 路径中的目录；如果指定多个，请用 ";" 分隔。(/I[path])

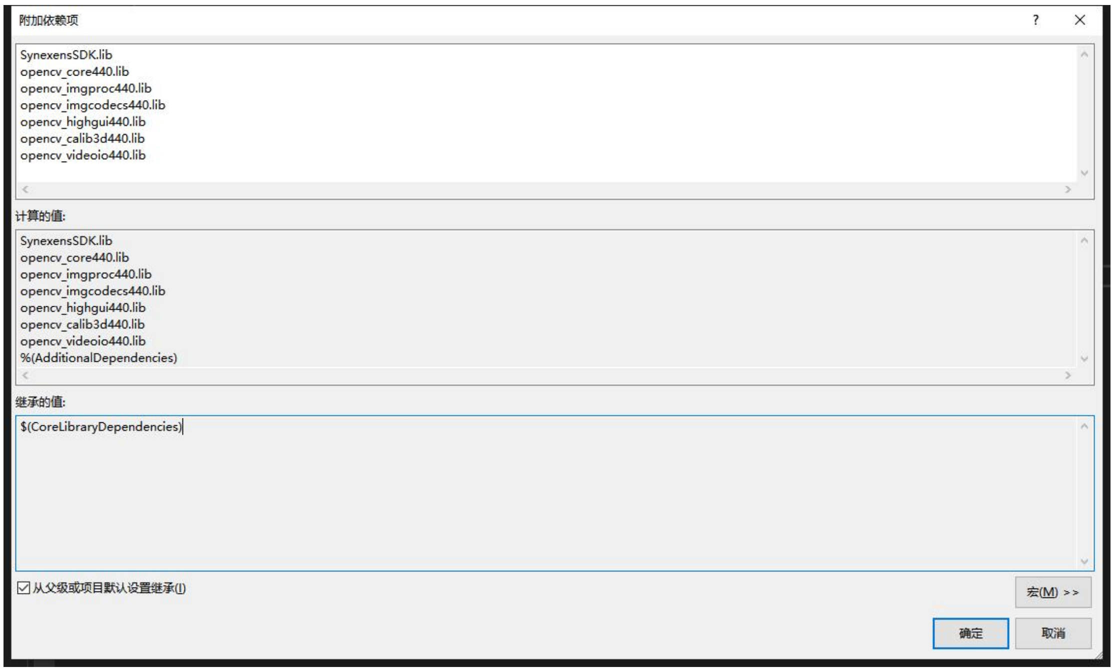


libsynexens4_demo 属性页

配置(C): Release　平台(P): 活动(x64)

输出文件	$(OutDir)$(TargetName)$(TargetExt)
显示进度	未设置
版本	
启用增量链接	
增量链接数据库文件	$(IntDir)$(TargetName).ilk
取消显示启动版权标志	是 (/NOLOGO)
忽略导入库	否
注册输出	否
逐用户重定向	否
附加库目录	D:\C++\libsynexens4.0.0.0\lib;D:\C++\libsynexens
链接库依赖项	是
使用库依赖项输入	否
链接状态	
阻止 Dll 绑定	
将链接器警告视为错误	
强制文件输出	
创建可热修补映像	
指定节特性	

附加库目录
允许用户重写环境库路径。(/LIBPATH:folder)



附加依赖项

```
SynexensSDK.lib
opencv_core440.lib
opencv_imgproc440.lib
opencv_imgcodecs440.lib
opencv_highgui440.lib
opencv_calib3d440.lib
opencv_videoio440.lib
```

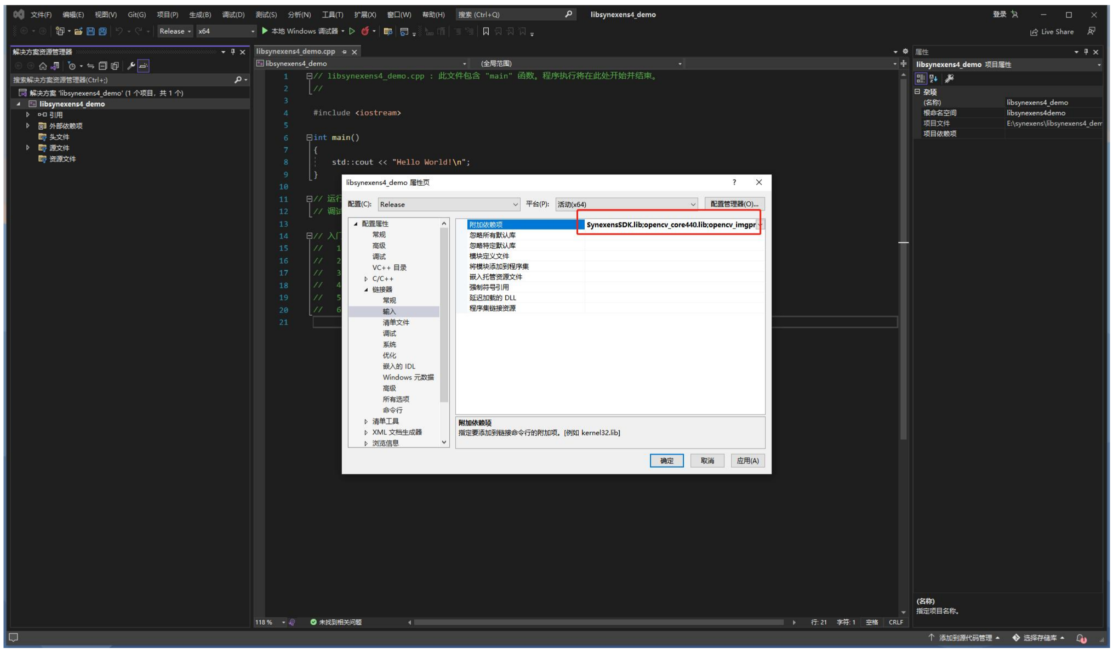
计算的值:

```
SynexensSDK.lib
opencv_core440.lib
opencv_imgproc440.lib
opencv_imgcodecs440.lib
opencv_highgui440.lib
opencv_calib3d440.lib
opencv_videoio440.lib
%(AdditionalDependencies)
```

继承的值:

```
$(CoreLibraryDependencies)
```

[✓] 从父级或项目默认设置继承(I)


libsynexens4_demo 属性页

配置(C): Release　平台(P): 活动(x64)

附加依赖项: SynexensSDK.lib;opencv_core440.lib;opencv_imgpr

附加依赖项
指定要添加到链接命令行的附加项。[例如 kernel32.lib]

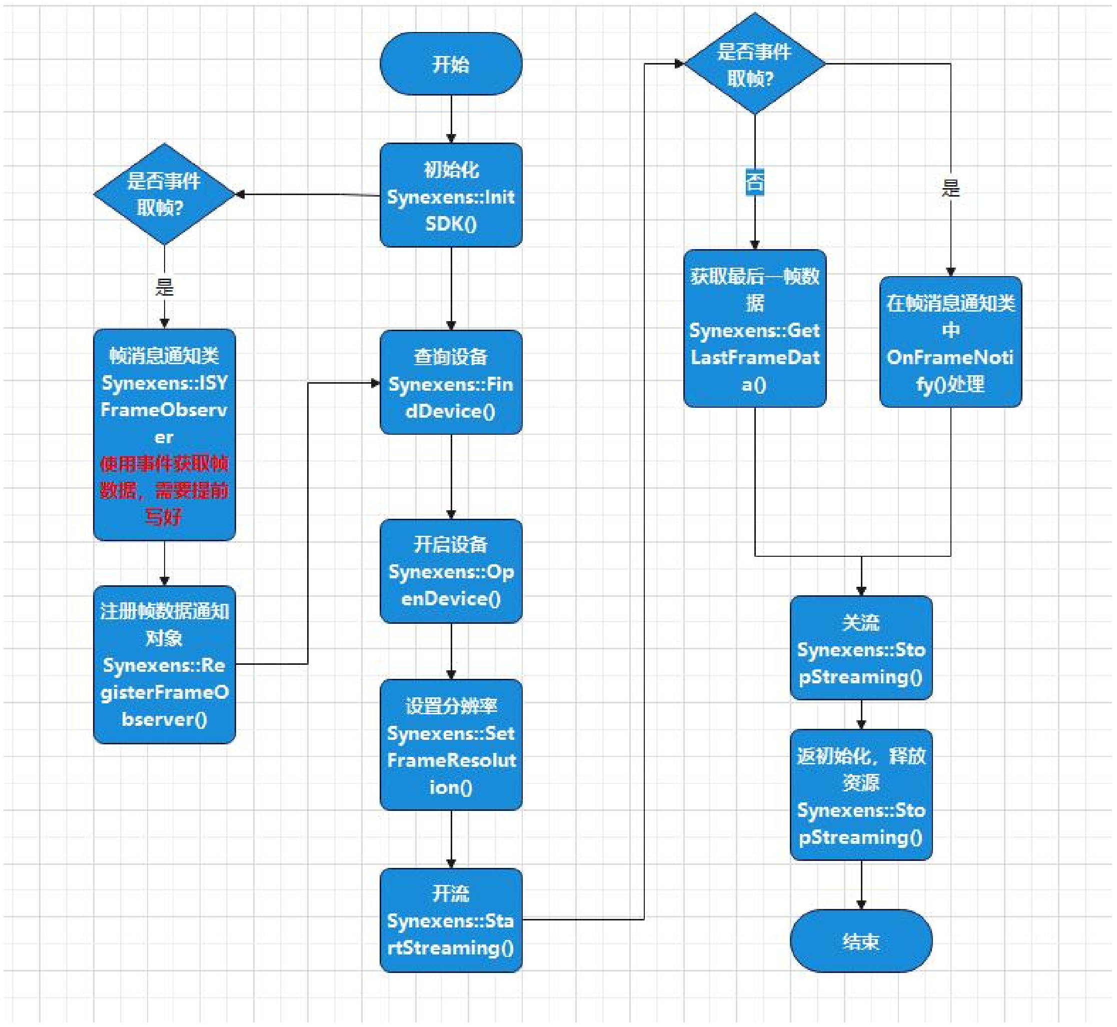
```
// libsynexens4_demo.cpp : 此文件包含 "main" 函数。程序执行将在此处开始并结束。
//

#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
}
```

运行 demo 需要 opencv 依赖库，自行开发不需要依赖 opencv

2.2.4. After completing the configuration, you can enter the project for development. If you need to run the demo, just copy the demo code to run it

Note: Running a demo needs to configure the include path, and missing dll files need to be copied to the program running directory

2.3. The SDK must call the process


开始
初始化 Synexens::InitSDK()
是否事件取帧?
是
帧消息通知类 Synexens::ISYFrameObserver
使用事件获取帧数据，需要提前写好
注册帧数据通知对象 Synexens::RegisterFrameObserver()
查询设备 Synexens::FindDevice()
开启设备 Synexens::OpenDevice()
设置分辨率 Synexens::SetFrameResolution()
开流 Synexens::StartStreaming()
是否事件取帧?
否
是
获取最后一帧数据 Synexens::GetLastFrameData()
在帧消息通知类中 OnFrameNotify()处理
关流 Synexens::StopStreaming()
返初始化，释放资源 Synexens::StopStreaming()
结束

3. API summary

3.1. Global interface

3.1.1. GetSDKVersion

Description: Get the SDK version number

Grammar：

GetSDKVersion(int& nLength, char* pstrSDKVersion = nullptr);

Parameter:

Parameter Name	Description	in/out
nLenght	character size	in/out
pstrSDKVersion	The SDK version number string pointer	in/out

3.1.2. InitSDK

Description: Initializing the SDK

Grammar:

InitSDK();

3.1.3. UnInitSDK

Description: In-initialize the SDK to release resources

Grammar:

UnInitSDK();

3.1.4. RegisterErrorObserver

Description: Register an error message notification object pointer

Grammar:

RegisterErrorObserver(ISYErrorObserver* pObserver);

Parameter:

Parameter Name	Description	in/out
pObserver	Error message notifies the object pointer	in

3.1.5. RegisterEventObserver

Description: Register the event notification object pointer

Grammar:

RegisterEventObserver(ISYEventObserver* pObserver);

Parameter:

Parameter Name	Description	in/out
pObserver	Event notification object pointer	in

3.1.6. RegisterFrameObserver

Description: Register the data frame notification object pointer

Grammar:

RegisterFrameObserver(ISYFrameObserver* pObserver);

Parameter:

Parameter Name	Description	in/out
pObserver	Data frame notification object pointer	in

3.1.7. UnRegisterErrorObserver

Description: Logthe error message notification object pointer

Grammar:

UnRegisterErrorObserver(ISYErrorObserver* pObserver);

Parameter:

Parameter Name	Description	in/out
pObserver	Error message notifies the object pointer	in

3.1.8. UnRegisterEventObserver

Description: Logout event notification object pointer

Grammar:

UnRegisterEventObserver(ISYEventObserver* pObserver);

Parameter:

Parameter Name	Description	in/out

pObserver	Event notification object pointer	in

3.1.9. UnRegisterFrameObserver

Description: Logout the data frame notification object pointer

Grammar：

UnRegisterFrameObserver(ISYFrameObserver* pObserver);

Parameter：

Parameter Name	Description	in/out
pObserver	Data frame notification object pointer	in

3.1.10. FindDevice

Description: Find the device

Grammar：

FindDevice(int& nCount, SYDeviceInfo* pDevice = nullptr);

Parameter：

Parameter Name	Description	in/out
nCount	Number of devices	in/out
pDevice	Device information, memory	in/out

	allocated externally, only nCount when pDevice passes into nullptr	

3.1.11. OpenDevice

Description: Turn on the device

Grammar：

OpenDevice(const SYDeviceInfo& deviceInfo);

Parameter：

Parameter Name	Description	in/out
deviceInfo	Facility information	in

3.1.12. CloseDevice

Description: Turn off the device

Grammar：

CloseDevice(unsigned int nDeviceID);

Parameter：

Parameter Name	Description	in/out
nDeviceID	equipment id	in

3.1.13. QueryDeviceSupportFrameType

Description: The query device supports the data frame type

Grammar：

QueryDeviceSupportFrameType(unsigned int nDeviceID, int& nCount, SYSupportType * pSupportType = nullptr);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nCount	The number of data frame types supported is only used when pSupportType is empty, otherwise used to verify whether the pSupportType memory allocation quantity matches	in/out
pSupportType	Supported data frame types, memory allocated externally, only nCount when pFrameType is passed to nullptr	in/out

3.1.14. QueryDeviceSupportResolution

Description: Query the frame resolution supported by the device

Grammar:

QueryDeviceSupportResolution(unsigned int nDeviceID, SYSupportType supportType, int& nCount, SYResolution* pResolution = nullptr);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
supportType	Frame type	in
nCount	Supported number of resolution, pResolution is empty is only used as the return number, otherwise used to verify whether the pResolution memory allocation quantity matches	in/out
pResolution	Supported resolution type, memory allocated externally, only nCount when pResolution passes to nullptr	in/out

3.1.15. GetCurrentStreamType

Description: Get the current stream type

Grammar:

GetCurrentStreamType(unsigned int nDeviceID);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment ID	in

3.1.16. StartStreaming

Description: Start the data stream

Grammar:

StartStreaming(unsigned int nDeviceID, SYStreamType streamType);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
streamType	Data flow type	in

3.1.17. StopStreaming

Description: Close the data flow

Grammar:

StopStreaming(unsigned int nDeviceID);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in

3.1.18. ChangeStreaming

Description: Switching over the data flow

Grammar:

ChangeStreaming(unsigned int nDeviceID, SYStreamType streamType);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
streamType	Data flow type	in

3.1.19. SetFrameResolution

Description: Set resolution (If the data stream is started, the flow off-> Set resolution-> Open on is performed internally)

Grammar:

SetFrameResolution(unsigned int nDeviceID, SYFrameType frameType, SYResolution resolution);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
frameType	Frame type	in
resolution	Frame resolution	in

3.1.20. GetFrameResolution

Description: Obtain the device frame resolution

Grammar：

GetFrameResolution(unsigned int nDeviceID, SYFrameType frameType, SYResolution& resolution);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
frameType	Frame type	in
resolution	Frame resolution	in

3.1.21. GetFilter

Description: Filtered-on state

Grammar：

GetFilter(unsigned int nDeviceID, bool& bFilter);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bFilter	Filter on state, true-on filtering, false-not on filtering	out

3.1.22. SetFilter

Description: Filter on / off

Grammar:

SetFilter(unsigned int nDeviceID, bool bFilter);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bFilter	Filter on state, true-on filtering, false-not on filtering	in

3.1.23. GetFilterList

Description: Get the filter list

Grammar:

GetFilterList(unsigned int nDeviceID, int& nCount, SYFilterType* pFilterType = nullptr);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nCount	Filter list length	in/out
pFilterType	Filter list	in/out

3.1.24. SetDefaultFilter

Description: Set the default filter

Grammar:

SetDefaultFilter(unsigned int nDeviceID);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in

3.1.25. AddFilter

Description: Increase the filtering

Grammar:

AddFilter(unsigned int nDeviceID, SYFilterType filterType);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
filterType	Filter type	in

3.1.26. DeleteFilter

Description: Remove the filter

Grammar：

DeleteFilter(unsigned int nDeviceID, int nIndex);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nIndex	Index in the filter list	in

3.1.27. ClearFilter

Description: Clear the filter

Grammar：

ClearFilter(unsigned int nDeviceID);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in

3.1.28. SetFilterParam

Description: Set the filter parameters

Grammar：

SetFilterParam(unsigned int nDeviceID, SYFilterType filterType, int nParamCount, float* pFilterParam);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
filterType	Filter type	in
nParamCount	Number of filtering parameters	in/out
pFilterParam	Filter parameters	in/out

3.1.29. GetFilterParam

Description: Obtain the filter parameters

Grammar：

GetFilterParam(unsigned int nDeviceID, SYFilterType filterType, int& nParamCount, float* pFilterParam = nullptr);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
filterType	Filter type	in
nParamCount	Number of filtering parameters	in/out
pFilterParam	Filter parameters	in/out

3.1.30. GetMirror

Description: Obtain the horizontal mirror status

Grammar：

GetMirror(unsigned int nDeviceID, bool& bMirror);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bMirror	Horizontal mirror state, true-horizontal mirror on, false-horizontal mirror not on	out

3.1.31. SetMirror

Description: Horizontal mirror image on / off

Grammar:

SetMirror(unsigned int nDeviceID, bool bMirror);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bMirror	Horizontal mirror switch, true-on horizontal mirror, false-off horizontal mirror	in

3.1.32. GetFlip

Description: Get the vertical flip state

Grammar:

GetFlip(unsigned int nDeviceID, bool& bFlip);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bFlip	Vertical flip state, true-vertical flip on, false-not on	out

3.1.33. SetFlip

Description: Vertical flip, on / off

Grammar:

SetMirror(unsigned int nDeviceID, bool bMirror);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bFlip	Vertical flip switch, true-on vertical flip, false-off vertical flip	in

3.1.34. GetIntegralTime

Description: Get integral time

Grammar:

GetIntegralTime(unsigned int nDeviceID, int& nIntegralTime);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nIntegralTime	Integral time	out

3.1.35. SetIntegralTime

Description: Set the integration time

Grammar:

SetIntegralTime(unsigned int nDeviceID, int nIntegralTime);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nIntegralTime	Integral time	in

3.1.36. GetIntegralTimeRange

Description: Obtain the integral time adjustment range

Grammar:

GetIntegralTimeRange(unsigned int nDeviceID, SYResolution depthResolution, int& nMin, int& nMax);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
depthResolution	depth resolution ratio	in
nMin	Integration time minimum	out
nMax	Integration time max	out

3.1.37. GetDistanceMeasureRange

Description: Get the ranging range

Grammar:

GetDistanceMeasureRange(unsigned int nDeviceID, int& nMin, int& nMax);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nMin	Minimum range	out
nMax	Maximum range	out

3.1.38. GetDistanceUserRange

Description: Get user ranging range

Grammar:

GetDistanceUserRange(unsigned int nDeviceID, int& nMin, int& nMax);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nMin	Minimum range of distance measurement	out
nMax	Maximum range of	out

	measurement	

3.1.39. SetDistanceUserRange

Description: Set the user ranging range

Grammar:

SetDistanceUserRange(unsigned int nDeviceID, int nMin, int nMax);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nMin	Minimum range of distance measurement	in
nMax	Maximum range of measurement	in

3.1.40. GetDeviceSN

Description: Read the Equipment sn number

Grammar:

GetDeviceSN(unsigned int nDeviceID, int& nLength, char* pstrSN = nullptr);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nLength	Character size	in/out
pstrSN	Equipment sn Number string pointer, by the external memory allocation, pstrSN incoming nullptr only get nLength	in/out

3.1.41. SetDeviceSN

Description：Write to the Equipment sn number

Grammar：

SetDeviceSN(unsigned int nDeviceID, int nLength, const char* pstrSN);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nLength	Character size	in
pstrSN	Equipment sn number string pointer	in

3.1.42. GetDeviceHWVersion

Description：Read the Equipment firmware version number

Grammar:

GetDeviceHWVersion(unsigned int nDeviceID, int& nLength, char* pstrHWVersion = nullptr);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nLength	Character size	in/out
pstrHWVersion	Firmware version number string pointer, by the external allocated memory, pstrHWVersion incoming nullptr only get nLength	in/out

3.1.43. GetDepthColor

Description: Obthe depth corresponding pseudo-color

Grammar:

GetDepthColor(unsigned int nDeviceID, int nCount, const unsigned short* pDepth, unsigned char* pColor);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nCount	Data volume (memory space	in

	pDepth nCount * 2 bytes, pColor nCount * 3 bytes)	
pDepth	Deep data	in
pColor	Depth corresponds to pseudo-color (24-bit RGB format)	in/out

3.1.44. GetDepthPointCloud

Description: Obtain the depth of the corresponding point cloud data

usage:

GetDepthPointCloud(unsigned int nDeviceID, int nWidth, int nHeight, const unsigned short* pDepth, SYPointCloudData* pPointCloud, bool bUndistort = false);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nWidth	Width	in
nHeight	Height	in
pDepth	Deep data	in
pPointCloud	Depth corresponds to point cloud data, with is allocated externally	in/out
bUndistort	Crop flag, true-crop false-do not crop	in

3.1.45. GetRGBD

Description：Gain RGBD

Grammar：

GetRGBD(unsigned int nDeviceID, int nSourceDepthWidth, int nSourceDepthHeight, unsigned short* pSourceDepth, int nSourceRGBWidth, int nSourceRGBHeight, unsigned char* pSourceRGB, int nTargetWidth, int nTargetHeight, unsigned short* pTargetDepth, unsigned char* pTargetRGB);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
nSourceDepthWidth	Source depth data width	in
nSourceDepthHeight	Source depth data height	in
pSourceDepth	Source depth data	in
nSourceRGBWidth	Source RGB data width	in
nSourceRGBHeight	Source RGB data height	in
pSourceRGB	Source RGB data	in
nTargetWidth	The RGBD data width	in
nTargetHeight	The RGBD data height	in
pTargetDepth	Depth data in RGBD, with externally allocated memory, and data length consistent	in/out

	with the source RGB length	
pTargetRGB	RGB data in RGBD, memory is allocated externally, and the data length coincides with the source RGB length	in/out

3.1.46. GetLastFrameData

Description: Get the latest frame of data

Grammar:

GetLastFrameData(unsigned int nDeviceID, SYFrameData*& pFrameData);

Parameter:

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
pFrameData	Last frame data	in/out

3.1.47. Undistort

Description: To aberrate

Grammar:

Undistort(unsigned int nDeviceID, const unsigned short* pSource, int nWidth, int nHeight, bool bDepth, unsigned short* pTarget);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
pSource	To distortion data pointer	in
nWidth	Image width	in
nHeight	Image height	in
bDepth	Is it the depth data / RGB data	in
pTarget	Dedistortion result data pointer, by external memory, the data length is consistent with the time to be distorted data pointer length	out

3.1.48. GetIntric

Description：Get the camera parameter

Grammar：

GetIntric(unsigned int nDeviceID, SYResolution resolution, SYIntrinsics& intrinsics);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
resolution	Equipment Resolution	in
intrinsics	Camera parameter	in/out

3.1.49. GetTrailFilter

Description：Get the drag shadow filter on state

Grammar：

GetTrailFilter(unsigned int nDeviceID, bool& bFilter);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bFilter	The drag shadow filter is turned on in the open state	out

3.1.50. SetTrailFilter

Description：Turn the drag shadow filter on / off

Grammar：

SetTrailFilter(unsigned int nDeviceID, bool bFilter);

Parameter：

Parameter Name	Description	in/out
nDeviceID	Equipment id	in
bFilter	Drag filter switch	in

3.2. Return to the Parameter description

All interfaces return the Parameter as error codes. For more details, please see the data structure definition description

4. Filter setting instructions

4.1. Filter Parameter setting instructions

The amplitude filter, AMPLITITUD

Example：

```
float threshold_value{ 0 };

threshold_value[0] = 10;// amplititud_threshold

int num = 1;

SetFilterParam(nDeviceID, filterType, num , threshold_value);
```

Median filter, MEDIAN

Example:

```
float threshold_value{ 0 };

threshold_value[0] = 3;// median_ksize
```

```
threshold_value[1] = 1;// median_iterations

int num = 2;

SetFilterParam(nDeviceID, filterType, num , threshold_value);
```

The boundary filter is EDGE

Example:

```
float threshold_value{ 0 };

threshold_value[0] = 50;//edge_threshold

int num = 1;

SetFilterParam(nDeviceID, filterType, num , threshold_value);
```

Spot-filtered SPECKLE

Example:

```
float threshold_value{ 0 };

threshold_value[0] = 40;// speckle_size

threshold_value[1] = 100;// speckle_max_diff

int num = 2;

SetFilterParam(nDeviceID, filterType, num , threshold_value);
```

4.2. Description of the filtered Parameter range

Filter interface	Parameter1-min	Parameter1-max	Parameter1 Recommended value	Parameter2-min	Parameter2-max	Parameter2 Recommended value
AMPLITITUD	0	100	6			
MEDIAN	3	5	3	0	5	1
EDGE	20	200	50			
SPECKLE	24	200	40	40	200	

4.3. The filter call sequence is indicated

CS20:Median, boundary, spots, median

CS30:The front segment has built-in median, boundary, and median filter can be added to the back end.

5. Description of the data structure definition

5.1. Error code

```
enum SYErrorCode

    {

        //Success

        SYERRORCODE_SUCCESS = 0,

        //Fail

        SYERRORCODE_FAILED = 1,

        //Equipment not present

        SYERRORCODE_DEVICENOTEXIST = 2,

        //Equipment did not open

        SYERRORCODE_DEVICENOTOPENED = 3,

        //Unsupported resolution

        SYERRORCODE_UNKOWNRESOLUTION = 4,

        //Equipment the pointer handle is empty

        SYERRORCODE_DEVICEHANDLEEMPTY = 5,

        //Equipment output data formatting has failed

        SYERRORCODE_SETOUTPUTFORMATFAILED = 6,

        //Failed to get the video stream control pointer

        SYERRORCODE_GETSTREAMCTRLFAILED = 7,

        //Failed to start the video streaming
```

```
SYERRORCODE_STARTSTREAMINGFAILED = 8,

//The communication pointer is empty

SYERRORCODE_COMMUNICATEOBJECTEMPTY = 9,

//Invalid SN number

SYERRORCODE_UNKOWNSN = 10,

//String length overflow

SYERRORCODE_STRINGLENGTHOUTRANGE = 11,

//Invalid frame type

SYERRORCODE_UNKOWNFRAMETYPE = 12,

//Invalid Equipment type

SYERRORCODE_UNKOWNDEVICETYPE = 13,

//Equipment the object pointer is empty

SYERRORCODE_DEVICEOBJECTEMPTY = 14,

//The notification object pointer is null

SYERRORCODE_OBSERVEREMPTY = 15,

//The notification object was not found

SYERRORCODE_OBSERVERNOTFOUND = 16,

//Quantity overflow

SYERRORCODE_COUNTOUTRANGE = 17,

//The UVC failed to initialize

SYERRORCODE_UVCINITFAILED = 18,

//The UVC failed to find the equipment
```

```
        SYERRORCODE_UVCFINDDEVICEFAILED = 19,

        //No data frames

        SYERRORCODE_NOFRAME = 20,

        //The program path fetch failed

        SYERRORCODE_GETAPPFOLDERPATHFAILED = 21,

        //The video stream is not started

        SYERRORCODE_NOSTREAMING = 22,

        //The algorithm pointer is empty

        SYERRORCODE_RECONSTRUCTIONEMPTY = 23,

    };
```

5.2. Equipment type

```
enum SYDeviceType

    {

        //Invalid

        SYDEVICETYPE_NULL = 0,

        //CS30 dual frequency

        SYDEVICETYPE_CS30_DUAL,

        //CS30 single frequency

        SYDEVICETYPE_CS30_SINGLE,

        //CS20 dual frequency

        SYDEVICETYPE_CS20_DUAL,
        //CS20 single frequency
        SYDEVICETYPE_CS20_SINGLE,
        //CS20_P
        SYDEVICETYPE_CS20_P,
        //CS40
        SYDEVICETYPE_CS40,
    };
```

5.3. Data flow type

```
enum SYStreamType
    {
        //Invalid
        SYSTREAMTYPE_NULL = 0,
        //RAW
        SYSTREAMTYPE_RAW,
        //Depth
        SYSTREAMTYPE_DEPTH,
        //RGB
        SYSTREAMTYPE_RGB,
        //Depth+IR
        SYSTREAMTYPE_DEPTHIR,
        //Depth+RGB
        SYSTREAMTYPE_DEPTHRGB,
        //Depth+IR+RGB
        SYSTREAMTYPE_DEPTHIRRGB,
        //RGBD (depth of + RGB after mapping)
        SYSTREAMTYPE_RGBD,
        //RAW_RGB
        SYSTREAMTYPE_RAWRGB,
    };
```

5.4. 分辨率枚举

```
enum SYResolution
    {
        //Invalid
        SYRESOLUTION_NULL = 0,
        //320*240
        SYRESOLUTION_320_240,
        //640*480
        SYRESOLUTION_640_480,
        //960*540
        SYRESOLUTION_960_540,
        //1920*1080
        SYRESOLUTION_1920_1080,
    };
```

5.5. Data frame type

```
enum SYFrameType
    {
        //Invalid
        SYFRAMETYPE_NULL = 0,
        //RAW
        SYFRAMETYPE_RAW,
        //Depth
        SYFRAMETYPE_DEPTH,
        //IR
        SYFRAMETYPE_IR,
        //RGB
        SYFRAMETYPE_RGB,
    };
```

5.6. Support type

```
enum SYSupportType
    {
        //Invalid
        SYSUPPORTTYPE_NULL = 0,
		//Depth
		SYSUPPORTTYPE_DEPTH,
		//RGB
		SYSUPPORTTYPE_RGB,
		//RGBD
		SYSUPPORTTYPE_RGBD,
	};
```

5.7. Event type

```
enum SYEventType
	{
		//Invalid
		SYEVENTTYPE_NULL = 0,
		//Equipment linkage
		SYEVENTTYPE_DEVICECONNECT,
		//Equipment disconnect
		SYEVENTTYPE_DEVICEDISCONNECT,
	};
```

5.8. Filter type

```
enum SYFilterType
```

```
{
    //Invalid
    SYFILTERTYPE_NULL = 0,
    //Median
    SYFILTERTYPE_MEDIAN,
    //Amplitude
    SYFILTERTYPE_AMPLITUDE,
    //Boundary
    SYFILTERTYPE_EDGE,
    //Spot
    SYFILTERTYPE_SPECKLE,
    //Big gold threshold
    SYFILTERTYPE_OKADA,
    //Boundary 2
    SYFILTERTYPE_EDGE_MAD,
    //Gauss
    SYFILTERTYPE_GAUSS,
    //Standby
    SYFILTERTYPE_EXTRA,
    //Standby 2
    SYFILTERTYPE_EXTRA2,
};
```

5.9. Equipment information

```
struct SYDeviceInfo

    {

        //Equipment unique ID

        unsigned int m_nDeviceID = 0;

        //Equipment type

        SYDeviceType m_deviceType = SYDEVICETYPE_NULL;

    };
```

5.10. 事件信息

```
struct SYEventInfo

    {

        // Event type

        SYEventType m_eventType = SYEVENTTYPE_NULL;

        //Event information data

        void* m_pEventInfo = nullptr;

        // DL

        int m_nLength = 0;

    };
```

5.11. Data frame information

```
struct SYFrameInfo
```

```
    {
        //Frame type
        SYFrameType m_frameType = SYFRAMETYPE_NULL;
        //Height (pixel)
        int m_nFrameHeight = 0;
        //Width (pixel)
        int m_nFrameWidth = 0;
    };
```

5.12. Data frame; data frames

```
struct SYFrameData
    {
        //The number of frames
        int m_nFrameCount = 0;
        //Frame information
        SYFrameInfo* m_pFrameInfo = nullptr;
        //Frame data
        void* m_pData = nullptr;
        //Data length
        int m_nBuffferLength = 0;
    };
```

5.13. Point cloud data structure

```
struct SYPointCloudData
    {
        //X
        float m_fltX = 0.f;
        //Y
        float m_fltY = 0.f;
        //Z
        float m_fltZ = 0.f;
    };
```

5.14. Camera Parameter construct

```
struct SYIntrinsics
    {
        // Lens perspective
        float m_fltFOV[2];
        // Distortion coefficient
        float m_fltCoeffs[5];
        // Focus length in the x-direction
        float m_fltFocalDistanceX;
        // The focal length in the y-direction
        float m_fltFocalDistanceY;
        // The imaging center point in the x direction, cx
        float m_fltCenterPointX;
        // The imaging center point in the y direction is the cy
        float m_fltCenterPointY;
        // Width
        int m_nWidth;
        // Height
        int m_nHeight;
    };
```

6. FQA

f：DLL not found while running under win

a：Need to copy the prompted dll file to the program run directory

f：The Linux runtime prompts uvc _ open: -3

a：Get the script compressed file and execute the script file inside

f：A select() timeout. error has occurred

a：Equipment Opening timeout may be caused by insufficient power supply and insufficient usb bandwidth. It is recommended to connect external hub for power supply or access different usb interfaces

f：The noise point is relatively large

a：You can set the filter Parameter through the GUI to get the desired effect and add it to the SDK

f：The xxx library was not found

a：Run the program through run.sh to ensure the correct library path imported by run.sh, or install the dependency library under the usr / lib

f：The cs40 cs20-p could not find the Equipment

a：Determine that the Equipment startup and Equipment remain in the same network segment as the industrial controller. Make sure to ping the ip address of the Equipment

f: cs40 cs20p When connecting to multiple Equipment's, you can only find one or find none

a：To ensure that Equipment is connected through the same network port, it is recommended to connect multiple Equipment via a switch .

7. About the Equipment connection

Note: There is no limit to Equipment connection in the SDK, theoretically unlimited Equipment. How many Equipment can be connected on the specific industrial control machine depends on the hardware support of the industrial control machine. At present, there are several points to note after testing：

1. An external hub even with more than one usb connection port can only be connected to one Equipment .
2. An independent usb on the industrial controller can connect up to two Equipment, which needs to be adjusted according to different models. In fact, the independent usb interface (some industrial controllers may have multiple usb ports, but these usb may use the same bandwidth and the same power supply).
3. At present, two CS20s and one CS30 have been successfully connected to the industrial control machine.

Disclaimer

The device application information and other similar content described in this publication facilitate you only and may be replaced by updated information. It is your own responsibility to ensure that the application meets the technical specifications. The Company shall make no representations or guarantees, express or implied, in writing, or oral, statutory or otherwise, including, except for its use, quality, performance, marketable or applicable for a particular purpose. The Company shall not be liable for the consequences of such information and the use of thereof. The product shall not be used as a critical component in the life support system without the written approval of the Company.