## \_SDK接口文档

### 1 地址

文档存放在以下地址：

http://192.168.0.230/svn/PMP\_3D/Platform/output

分为Windows版本和Linux版本，目前只给出Release版，Linux版的安装说明在统计目录下的readme中，Windows请复制黏贴。

### 2 SDK接口

所有接口和数据结构都可在“emController.h”和”**emCommon\_type**.h”查到。

#### 2.1.1扫描设备接口 emScanDevice

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief Scan online devices

\* param quick Connect(default)

\* return Scan Device Number

\*attention: Host and device must be on the same network segment, such as 192.168.0.××, otherwise it is invalid, scanned device IP and other information will be put in \*emGuiDevScanInfo[]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int emScanDevice(bool quickCon)**

#### 2.1.2设备连接IP确认/修改接口 emDevChangeIp

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief to sure the scanned IP device to the host

\* param IP of the connected device,index of device in IP list

\* return if ok return =0 else <0

\* attention if the IP is changed ,device IP will be modified and device will be reboot, default ip\_index from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int emDevChangeIp(char\* ip, int ip\_index)**

#### 2.1.3打开设备接口 emOpenDevice

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief Open device mainly binding device serial number and device handle

\* param device handle address,usually define as void \*m\_Device\_1 = NULL;

device index, first number is 0

QMSG\_KEY\_1, msg ID

QuickRun, quick open maily init network and run param

isPointCloud, weather output piontcloud directly,not working

\* return if ok return =0 else <0

\* attention 1.DO NOT EDIT QMSG\_KEY\_1

2.nDeviceIndex start from 0.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emOpenDevice(EM\_DEV\_HANDLE& hDevice, int nDeviceIndex, int msg\_id, bool quickRun = false, bool isPointCloud = false)

#### 2.1.4关闭设备接口 emCloseDevice

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief Close device and release handle sources

\* param device index

\* return none

\* attention 1.nDeviceIndex must Less than the number of physical devices

2.device index start from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** emColseDevice(int nDeviceIndex = 0);

#### 2.1.5网络初始化及连接接口 emInitLibPara

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief init & build up UDP/TCP sockets between host and devices

\* param device index

\* return if ok return =0 else <0

\* attention 1.nDeviceIndex from 0

2.if open device in quick mode,do not use again

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emInitLibPara(**int** nDeviceIndex)

#### 2.1.6视差数据回调注册接口 emRegisterImageCallback()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief callback function to get the parallax data

\* param nDeviceIndex, device index

pUserParam, user data, if ues it,must define RICB struct,can passed udp packs lost rate in first param of RICB,define in emCommon\_type.h

callBackFuncDefinedByUser, a function pointer

\* return pUserParam

\* attention nDeviceIndex from 0, no need delete register

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void\*** emRegisterImageCallback(const int nDeviceIndex, void \*pUserParam, CallBackFuncDefineByAP callBackFuncDefinedByUser)

#### 2.2.1运行设备接口 emDevStart()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief start run device

\* param nDeviceIndex, device index

\* return if ok return =0 else <0

\* attention nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emDevStart(**int** nDeviceIndex)

#### 2.2.2停止设备接口 emDevStop()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief stop device

\* param nDeviceIndex, device index

\* return if ok return =0 else <0

\* attention nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emDevStop(**int** nDeviceIndex)

#### 2.2.3 2D/3D切换接口 emSwitchRunMode()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief switch between 2d or 3d

\* param nDeviceIndex, device index

sw\_mode, 0:2d 1:3d

\* return if ok return 0 else -1

\* attention nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emSwitchRunMode(**int** nDeviceIndex, **int** sw\_mode);

#### 2.2.4设置输出模式接口 emSetOutputOnceOrMulti()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief set output mode: once or continue

\* param nDeviceIndex, device index

onceOrmulti, 0-once; 1-continue

\* return if ok return 0 else -1

\* attention nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emSetOutputOnceOrMulti(**int** nDeviceIndex, **int** onecOrmulti)

#### 2.2.5设置曝光时间接口 emSetExposureTime()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief set exposure time

\* param nDeviceIndex, device index

extime, value of exposure time

typeDev,0:camera 1:projector

\* return if ok return 0 else -1 or exposuretime value

\* attention 1. nDeviceIndex from 0

1. extime must between 1000us and 20000us

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emSetExposureTime(**int** nDeviceIndex, u**int32\_t** extime, int typeDev);

#### 2.2.6获取曝光时间接口 emGetExposureTime()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief get exposure time

\* param nDeviceIndex, device index

extime, value of exposure time

\* return if ok return 0 else -1 or exposuretime value

\* attention 1. nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emGetExposureTime(**int** nDeviceIndex, u**int32\_t** &extime);

#### 2.2.7视差转换点云接口 emExchangeParallaxToPointCloud()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief only ROI pointcloud and correct images

\* param parallexBuf, parallex data from callbackfunction

grayImg, Corrected original image

pointCloud, Self-format pointcloud data

\* return if ok return =0 else <0

\* attention nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emExchangeParallaxToPointCloud(uint8\_t\* parallexBuf, uint8\_t \*grayImg, PointCloud\_EM\_Ptr& pointCloud);

#### 2.2.8视差转换点云扩展接口 emExchangeParallaxToPointCloudEx()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief ALL pointcloud and correct images

\* param parallexBuf, parallex data from callbackfunction

grayImg, Corrected original image

pointCloud, Self-format pointcloud data

\* return if ok return =0 else <0

\* attention nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emExchangeParallaxToPointCloudEx(uint8\_t\* parallexBuf, uint8\_t \*grayImg, PointCloud\_EM\_Ptr& pointCloud);

#### 2.2.9设置ROI区域值 emSetRoiValue()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief set ROI

\* param nDeviceIndex:device index

startX: row start value

startY: col start value

width: col value

height: row value

\* return if ok return =0 else <0

\* attention 1. nDeviceIndex from 0

2. ROI area must more than 128\*128

1. Height must less then 1536
2. Width must less then 2048
3. startX,startY start form 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**int** emSetRoiValue(const int nDeviceIndex, **const** **int** startX, **const** **int** startY, **const** **int** width, **const** **int** height);

#### 2.3获取系统状态信息接口

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief

\* param

\* return if ok return 0 else -1

\* attention nDeviceIndex from 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//获取系统运行状态信息

**int emGetSysStatus(int nDeviceIndex, uint64\_t \*allFrameCnt, uint64\_t \*powerOnTime, int32\_t \*frameRate, int32\_t \*inSideTemp, int32\_t \*tempStatu)**

//获取设备网络信息

**int emGetSysSensorInfo(int nDeviceIndex, char\* ip, char \*mac, char \*sn, char \*version, char \*name);**

//获取配置文件

**int emLoadConfigIniFile(int nDeviceIndex, char\* loadputfilePath);**

//恢复配置文件

**int emRecoveryConfigIniFile(const int nDeviceIndex = 0)；**

//同步系统配置文件

**int emSyncLocalConfigIniFile(const int nDeviceIndex, char\* filepath);**

//读取Q矩阵

**int upQmaxData(Qmax\* QData, const char\* qFilePath);**

#### 2.4自定义格式点云数据转化函数 convert2PCLPointCloud()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* brief get pcl-format pointcloud data

\* param pointCloud, self-format pointcloud data

pcl\_cloud, pcl-format pointcloud data

\* return none

\* attention 1. PointCloud\_EM\_Ptr if define in emCommon\_type.h

1. pointcloud format is <xyzrgb>
2. User must support pcl library or not been supported

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** convert2PCLPointCloud(PointCloud\_EM\_Ptr pointCloud, pcl::PointCloud<pcl::PointXYZRGB>::Ptr pcl\_cloud)

**This function defined in file emPointCloudOperation.h, especially functions 2.2.6 & 2.4, If you do not use these two functions, you need to calculate the point cloud yourself,Use the function in 2.3 to obtain the q matrix and use the similar triangle principle to calculate (recommended),thanks!**

### 3函数使用流程

#include "emController.h"

#include "stdio.h"

#include <stdlib.h>

#include <unistd.h>

void \*m\_Device\_1 = NULL;

void OnTestCallBackFun(PNP\_FRAME\_CALLBACK\_PARAM\* pFrame)

{

int recvBufID\_Old = 0;

int recvBufID = pFrame->nFrameID;

if(recvBufID\_Old != recvBufID)

{

recvBufID\_Old = pFrame->nFrameID;

printf("recv ok.\n");

///////////////////////////////////////////

//ADD YOUR TODO CODE

//memcpy(ImgBuffer,(unsigned char\*)pFrame->pImgBuf, pFrame->pBufferSize);

}

}

int main()

{

// test for single capturing

emController \*emDemo = new emController();

if (emDemo->emScanDevice(true) > EM\_STATUS\_SUCCESS)

{

if(EM\_STATUS\_SUCCESS == emDemo->emOpenDevice(m\_Device\_1, 0, MSQ\_KEY, true, false))

{

emDemo->emRegisterImageCallback(0, (void\*)NULL, OnTestCallBackFun);

emDemo->emDevStart(0);

usleep(10 \* 1000 \*1000);

emDemo->emDevStop(0);

}

else

printf("Open failed!\n");

}

else

printf("Scan failed!\n");

return 0;

}