## \_API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，提供Release版，Linux版的安装说明在同级目录下的readme中，Windows请复制黏贴。

### 2 SDK接口

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

#### 2.1.1扫描设备接口 emScanDevice

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

\* brief 扫描同网段在线设备

\* param quickCon：是否启用快速链接，在单个设备的系统中可以减少其他函数调用快速链接完成扫描设备过程；多设备中第一个设备被选中并完成以上设置过程

\* return 发现设备个数，当小于0时扫描设备失败

\*attention: 第一次使用时控制pc必须与嵌入式设备在相同网段，该网段为：192.168.0.xx,当发现设备后可以使用emDevChangeIp来修改嵌入式设备的IP地址，扫描到的设备信息都存在emGuiDevScanInfo二级数组中，如果是能快速链接则不能修改IP地址

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

**int emScanDevice(bool quickCon)**

#### 2.1.2设备连接IP确认/修改接口 emDevChangeIp

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

\* brief 确认或修改设备IP

\* param ip:要确认的设备IP或者要修改的设备IP

ip\_index:设备序号，如果单个设备则恒为0

\* return 成功返回0，失败返回值小于0

\* attention 如果修改IP地址，系统会重新启动且以后都保持修改后的IP地址 ip\_index序号从0开始，推荐在有GUI的界面中将该IP列表显示出来为最佳，否则需要自己开关相机时明确相机序号

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

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

#### 2.1.3打开设备接口 emOpenDevice

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

\* brief 打开设备，完成控制类初始化及ip\_index序号的绑定

\* param hDevice 设备控制句柄，绑定控制类及设备序号，始终指向当前操作的设备，如果为NULL，后续操作都无效

nDeviceIndex:设备序号，从0开始

msg\_id:初始化控制类消息，可以自定义值，推荐使用 QMSG\_KEY\_1，定义在头文件中，大于10000

QuickRun: 快速打开，包含网络初始化

isPointCloud:是否打开直接在回调中使用点云数据而不是视差数据， 测试中，尚未开放，是能会导致打开函数失败，目前必须为false

\* return 成功返回0，失败返回值小于0

\* attention 1.不要重复定义QMSG\_KEY\_1

2.nDeviceIndex 从0开始，一下函数都包含这一条原则.

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

**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 关闭设备并释放句柄

\* param nDeviceIndex:设备序号

\* return 无返回值

\* attention nDeviceIndex为以打开设备序号，其他无效

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

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

#### 2.1.5网络初始化及连接接口 emInitLibPara

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

\* brief 初始化网络（TCP/UDP）并建立网络处理线程

\* param nDeviceIndex:设备序号

\* return 成功返回0，失败返回值小于0

\* attention 如果在打开设备时使能快速打开，请不要再次调用

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

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

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

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

\* brief 注册获取视差或点云数据的回调函数

\* param nDeviceIndex:设备序号

pUserParam: 用户数据，有固定格式结构体，格式请参照 RICB结构体，定义在emCommon\_type.h头文件中

callBackFuncDefinedByUser:回调函数

\* return 用户定义数据指针

\* attention 参看RICB结构体

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

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

#### 2.2.1运行设备接口 emDevStart()

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

\* brief 开始运行

\* param nDeviceIndex:设备序号

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

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

#### 2.2.2停止设备接口 emDevStop()

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

\* brief 停止运行

\* param nDeviceIndex:设备序号

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

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

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

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

\* brief 视频流和3D数据模式切换

\* param nDeviceIndex:设备序号

sw\_mode: 0:视频流模式 1:3d模式

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

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

#### 2.2.4设置运行模式接口 emSetOutputOnceOrMulti()

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

\* brief 设置相机运行模式

\* param nDeviceIndex:设备序号

onceOrmulti:0-单次运行; 1-连续运行

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

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

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

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

\* brief 设置曝光时间

\* param nDeviceIndex:设备序号

extime:曝光时间值

typeDev:0:设置相机曝光时间 1:设置投影曝光时间

\* return 成功返回0，失败返回值小于0

\* attention 1. 系统默认曝光时间为12ms,如果设置曝光时间小于12ms，则只修改相机曝光时间，投影时间不能修改；如果设置曝光时间大于12ms，则需要先修改投影曝光时间后再修改相机曝光时间，否则无效

1. 推荐使用相机曝光时间等于投影曝光时间且12ms为投影曝光时间的下限，不可再减少
2. 投影曝光时间绝对不能小于相机曝光时间

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

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

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

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

\* brief 获取相机曝光时间

\* param nDeviceIndex:设备序号

extime:相机曝光时间值

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

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

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

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

\* brief 仅包含ROI的视差到点云转化

\* param parallexBuf, 回调函数中返回的视差数据

grayImg, 矫正后的灰度图

pointCloud, 头文件中自定义类型的点云数据

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

**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 全数据的视差到点云转化

\* param parallexBuf, 回调函数中返回的视差数据

grayImg, 矫正后的灰度图

pointCloud, 头文件中自定义类型的点云数据

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

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

#### 2.2.9设置ROI区域值 emSetRoiValue()

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

\* brief 通过起始点及宽高设置ROI区域

\* param nDeviceIndex:设备序号

startX: 行起始点坐标值

startY: 列起始点坐标值

width: 宽

height: 高

\* return 成功返回0，失败返回值小于0

\* attention 1. nDeviceIndex 从0开始

2. ROI 区域必须大于像素值 128\*128

3. 高小于1536，宽小于2048

4. startX,startY 从1开始

5. 起始坐标值与宽高没有关系

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

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

**Note: If you use emSetRoiValue() function, please make sure that the next time you start the program, set the relevant area according to the config.ini file returned by the embedded device, because the library does not contain the reading and setting of the configuration file.**

#### 2.3获取系统状态信息接口

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

\* brief

\* param

\* return 成功返回0，失败返回值小于0

\* attention nDeviceIndex 从0开始

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

//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 转换点云数据格式为PCL格式

\* param pointCloud, 源数据格式，非PCL格式

pcl\_cloud, 目标数据格式，PCL格式

\* return 无

\* attention 1. PointCloud\_EM\_Ptr定义见 emCommon\_type.h

1. 点的格式为 <xyzrgb>
2. 用户环境必须支持pcl，否则无法运行
3. 使用时将emPointCloudOperation.h包含在.cpp中

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

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

**Note: 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;

}