



SDK 接口文档

XINGYING SDK 接口文档



目录

1. 概述.....	1
1.1 工作流程.....	1
2. 接口.....	2
2.1 初始化	2
2.2 逆初始化.....	2
2.3 获取版本信息.....	2
2.4 设置日志输出级别.....	3
2.5 初始化测力台.....	3
2.6 设置测力台数据回调.....	3
2.7 设置数据回调.....	4
2.8 设置日志输出回调.....	4
2.9 获取包的描述信息.....	5
2.10 获取数据描述信息.....	5
2.11 获取数据描述信息（弃用）.....	6
2.12 释放描述信息内存.....	6
2.13 时间转换.....	6
2.14 时间码转字符串.....	7
2.15 帧拷贝.....	7
2.16 释放帧.....	8
2.17 获取一帧数据.....	8
2.18 设置通知回调.....	8
2.19 根据名称获取标准 Tpose 描述信息.....	9
3. 数据结构.....	10
3.1 测力台数据 sForcePlates.....	10
3.2 测力台 sForcePlateData.....	10
3.3 动捕帧数据 sFrameOfMocapData.....	10
3.4 描述信息 sServerDescription.....	11
3.5 描述信息 sDataDescriptions.....	11
3.6 标记信息 sMarkerSetData.....	12
3.7 标记点坐标 MarkerData.....	12
3.8 刚体数据 sRigidBodyData.....	12
3.9 骨骼数据 sSkeletonData.....	12
3.10 打标签的 Marker 数据 sMarker.....	12
3.11 描述信息 sMarkerSetDescription.....	13
3.12 描述信息 sRigidBodyDescription.....	13
3.13 描述信息 sSkeletonDescription.....	13
3.14 描述信息 sForcePlateDescription.....	13
3.15 错误码.....	14
3.16 通知 sNotifyMsg.....	15
4. 数据解析用例.....	16

本文件版权为北京度量科技有限公司所有，未经本公司许可不得转让或复制给第三者。

The copyright of this document is the property of Beijing NOKOV Science & Technology Co., Ltd.
 Unauthorized disclosure or duplication to the third party is not permitted.

1. 概述

XINGYING SDK 是以 C++ 语言编写的通信插件，是动态库 dll 的插件形式，通过将 SDK 包含到 C++ 项目工程中，通过调用 SDK 的公有接口，即可获取 Marker 点和刚体信息，以及测力台包括其他模拟通道信息，使用参考 SampleClient。

1.1 工作流程

实例化客户端 -> 初始化 -> 获取 sdk 版本信息 -> 设置日志输出输出级别 -> 获取服务描述信息 -> 获取数据描述信息 -> 释放数据描述信息 -> 初始化测力台（如果需要） -> 设置测力台回调（如果需要） -> 设置流数据回调 -> 正常就可以接收数据了

2. 接口

2.1初始化

接口描述：

建立连接与通信

接口名称:

```
int Initialize(char* szServerAddress);
```

参数&返回值：

参数	参数类型	备注
szServerAddress	char*	服务器地址
返回值	int	见 3.15 错误码

2.2 逆初始化

接口描述：

关闭连接，销毁资源

接口名称:

```
int Uninitialize();
```

参数&返回值：

参数	参数类型	备注
无		
返回值	int	0 成功 其它失败

2.3 获取版本信息

接口描述：

获取 SDK 版本信息

接口名称:

```
void NokovSDKVersion(unsigned char Version[4]);
```

参数&返回值：

参数	参数类型	备注
Version	unsigned char 数组	版本信息

返回值	void	
-----	------	--

2.4 设置日志输出级别

接口描述：

设置日志输出级别

接口名称：

```
void SetVerbosityLevel(int level);
```

参数&返回值：

参数	参数类型	备注
level	int	0-None, 1-Info, 2-Warning, 3-Error, 4-Debug
返回值	void	

2.5 初始化测力台

接口描述：

初始化测力台

接口名称：

```
int WaitForForcePlateInit(long time = 0);
```

参数&返回值：

参数	参数类型	备注
time	long	超时时间，单位ms，0表示阻塞，等待初始化成功
返回值	int	0 成功，其它失败

2.6 设置测力台数据回调

接口描述：

设置测力台数据回调

接口名称：

```
int SetForcePlateCallback(void (*CallbackFunction)(sForcePlates* pForcePlate, void* pUserData), void* pUserData = 0);
```

参数&返回值：

参数		参数类型	备注
CallbackFunction		函数指针	接收数据的函数指针
	pForcePlate	sForcePlates	测力台数据
	pUserData	void *	用户附加数据（接收）
	返回值	void	
pUserData		void *	用户附加数据（设置）
返回值		int	0 成功

2.7 设置数据回调

接口描述：

设置动捕数据回调

接口名称：

```
int SetDataCallback(void (*CallbackFunction)(sFrameOfMocapData* pFrameOfData, void* pUserData), void* pUserData=0);
```

参数&返回值：

参数		参数类型	备注
CallbackFunction		函数指针	接收数据流的函数指针
	pFrameOfData	sFrameOfMocapData*	数据流，见 3.3
	pUserData	void *	用户附加数据（接收）
	返回值	void	
pUserData		void *	用户附加数据（设置）
返回值		int	0 成功

2.8 设置日志输出回调

接口描述：

设置日志数据回调

接口名称：

```
int SetMessageCallback(void (*CallbackFunction)(int id, char* szTraceMessage));
```

参数&返回值:

参数		参数类型	备注
CallbackFunction		函数指针	接收日志的函数指针
	id	int	输出级别
	szTraceMessage	char	日志信息
	返回值	void	
返回值		int	0 成功

2.9 获取包的描述信息

接口描述:

设置服务端相关信息

接口名称:

```
int GetServerDescription(sServerDescription *pServerDescription);
```

参数&返回值:

参数	参数类型	备注
pServerDescription	sServerDescription *	服务器相关信息, 见 3.4
返回值	int	0 成功, 其它失败

2.10 获取数据描述信息

接口描述:

获取初始信息

接口名称:

```
int GetDataDescriptions(sDataDescriptions** pDataDescriptions);
```

参数&返回值:

参数	参数类型	备注
pDataDescriptions	sDataDescriptions* *	数据描述信息指针
返回值	int	0 成功, 其它失败

2.11 获取数据描述信息（弃用）

接口描述：

获取初始信息

接口名称：

```
int GetDataDescriptionsEx(sDataDescriptions** pDataDescriptions);
```

参数&返回值：

参数	参数类型	备注
pDataDescriptions	sDataDescriptions**	数据描述信息指针
返回值	int	0 成功，其它失败

2.12 释放描述信息内存

接口描述：

释放描述信息内存

接口名称：

```
int FreeDataDescriptions(sDataDescriptions* pDataDescriptions);
```

参数&返回值：

参数	参数类型	备注
pDataDescriptions	sDataDescriptions*	数据描述信息指针
返回值	int	0 成功，其它失败

2.13 时间转换

接口描述：

时间码转换为时、分、秒、帧号

接口名称：

```
bool DecodeTimecode(unsigned int inTimecode, unsigned int inTimecodeSubframe,
int* hour, int* minute, int* second, int* frame, int* subframe);
```

参数&返回值：

参数	参数类型	备注
inTimecode	unsigned int	帧时间码
inTimecodeSubframe	unsigned int	帧数偏差

hour	int*	小时
minute	int*	分钟
second	int*	秒
frame	int*	帧号
subframe	int*	帧数偏差
返回值	bool	true 成功 false 失败

2.14 时间码转字符串

接口描述：

时间码转换为时、分、秒，字符串显示

接口名称：

```
bool TimecodeStringify(unsigned int inTimecode, unsigned int inTimecodeSubframe, char
*Buffer, int BufferSize);
```

参数&返回值：

参数	参数类型	备注
inTimecode	unsigned int	帧时间码
inTimecodeSubframe	unsigned int	帧数偏差
Buffer	char *	字符串输出 buffer
BufferSize	int	buffer 长度
返回值	bool	true 成功 false 失败

2.15 帧拷贝

接口描述：

单帧拷贝

接口名称：

```
int NokovCopyFrame(const sFrameOfMocapData* pSrc, sFrameOfMocapData* pDst);
```

参数&返回值：

参数	参数类型	备注
pSrc	const sFrameOfMocapData*	源数据
pDst	sFrameOfMocapData*	目标数据

返回值	int	0 成功，其它失败
-----	-----	-----------

2.16 释放帧

接口描述：

释放帧数据空间

接口名称：

```
int NokovFreeFrame(sFrameOfMocapData* pDst);
```

参数&返回值：

参数	参数类型	备注
pDst	sFrameOfMocapData*	帧数据指针
返回值	int	0 成功，其它失败

2.17 获取一帧数据

接口描述：

获取上一帧数据

接口名称：

```
sFrameOfMocapData* GetLastFrameOfMocapData();
```

参数&返回值：

参数	参数类型	备注
无		
返回值	sFrameOfMocapData*	帧数据，见 3.3 节

2.18 设置通知回调

接口描述：

设置通知回调

接口名称：

```
int SetNotifyMsgCallback(void (*CallbackFunc)(sNotifyMsg* pNotify,  
void* pUserData), void* pUserData=0);
```

参数&返回值：

参数	参数类型	备注
----	------	----

CallbackFunction		函数指针	接收数据的函数指针
	pNotify	sNotifyMsg	通知数据, 见 3.16
	pUserData	void *	用户附加数据 (接收)
	返回值	void	
pUserData		void *	用户附加数据 (设置)
返回值		int	0 成功

2.19 根据名称获取标准 Tpose 描述信息

接口描述:

根据名称获取标准 Tpose 数据

接口名称:

```
int GetTposeDataDescriptions(char* name, sDataDescriptions** pDataDescriptions);
```

参数&返回值:

参数	参数类型	备注
name	char*	名称
pDataDescriptions	sDataDescriptions **	描述信息, 见 3.5
返回值	int	0 成功, 其它失败

北京度量科技有限公司 010-64922321 Info@nokov.com www.nokov.com

3. 数据结构

3.1 测力台数据 sForcePlates

名称	类型	说明
iFrame	int	帧号
nForcePlates	int	数据数量
ForcePlates	sForcePlateData [8]	数据, 见 3.2 节

3.2 测力台 sForcePlateData

名称	类型	说明
Fxyz	float [3]	
xyz	float [3]	
Mfree	float	

3.3 动捕帧数据 sFrameOfMocapData

名称	类型	说明
iFrame	int	帧号
nMarkerSets	int	标记数量
MocapData	sMarkerSetData []	标记信息 3.6
nOtherMarkers	int	未定义的标记数量
OtherMarkers	MarkerData*	未定义的标记点坐标, 见 3.7
nRigidBody	int	刚体数量
RigidBody	sRigidBodyData []	刚体数据, 见 3.8
nSkeletons	int	骨骼数量
Skeletons	sSkeletonData []	骨骼数据, 见 3.9
nLabeledMarkers	int	打标签的 marker 数量
LabeledMarkers	sMarker []	打标签的 marker 数据, 见 3.10
nAnalogdatas	int	模拟通道数

Analogdata	float []	模拟通道数据
fLatency	float	时间差
Timecode	unsigned int	SMPTE 时间码
TimecodeSubframe	unsigned int	帧数偏差
iTimeStamp	long long	时间戳
params	short	参数

3.4 描述信息 sServerDescription

名称	类型	说明
HostPresent	bool	true 正常运行, false 不可用
szHostComputerName	char []	主机名称
HostComputerAddress	unsigned char [4]	主机地址
szHostApp	char []	应用程序名称
HostAppVersion	unsigned char [4]	应用程序版本
NokovSDKVersion	unsigned char [4]	SDK 版本

3.5 描述信息 sDataDescriptions

名称	类型	说明
type	int	0-标记点, 1-刚体, 2-骨骼, 3-测力台
Data	union	描述信息
MarkerSetDescription	sMarkerSetDescription*	见 3.11
RigidBodyDescription	sRigidBodyDescription*	见 3.12
SkeletonDescription	sSkeletonDescription*	见 3.13
ForcePlateDescription	sForcePlateDescription*	见 3.14

	MarkerSetData	sMarkerSetData*	见 3.6
--	---------------	-----------------	-------

3.6 标记信息 sMarkerSetData

名称	类型	说明
szName	char [256]	名称
nMarkers	int	数量
Markers	MarkerData *	数据, 见 3.7

3.7 标记点坐标 MarkerData

```
typedef float MarkerData[3]; // posX, posY, posZ
```

3.8 刚体数据 sRigidBodyData

名称	类型	说明
ID	int	唯一号
x, y, z	float	坐标
qx, qy, qz, qw	float	方位
nMarkers	int	marker 数量
Markers	MarkerData*	marker 数据
MarkerIDs	int *	marker ID
MarkerSizes	float*	标记大小
MeanError	float	偏差
params	short	自定义跟踪标记

3.9 骨骼数据 sSkeletonData

名称	类型	说明
skeletonID	int	唯一标识
nRigidBodyData	int	刚体数量
RigidBodyData	sRigidBodyData *	刚体数据, 见 3.8

3.10 打标签的 Marker 数据 sMarker

名称	类型	说明
----	----	----

ID	int	唯一标识
x	float	x
y	float	y
z	float	z
size	float	大小
params	short	自定义参数

3.11 描述信息 sMarkerSetDescription

名称	类型	说明
szName	char	名称
nMarkers	int	marker 数量
szMarkerNames	char**	marker 数据

3.12 描述信息 sRigidBodyDescription

名称	类型	说明
szName	char[256]	名称
ID	int	唯一标识
parentID	int	父刚体 ID
offsetx, offsety, offsetz	float	相对父刚体的偏移
qx, qy, qz, qw	float	相对于父刚体的方向

3.13 描述信息 sSkeletonDescription

名称	类型	说明
szName	char[256]	名称
skeletonID	int	ID,唯一标识
nRigidBodyes	int	刚体数量
RigidBodyes	sRigidBodyDescription[200]	刚体数据, 见 3.12

3.14 描述信息 sForcePlateDescription

名称	类型	说明
----	----	----

ID	int	唯一 ID
scale	int	比例
fWidth	float	板材物理宽度（制造商提供）
fLength	float	板材物理长度
Position	float[3]	位置
Electrical	float [3]	
Orientation	float[3][3]	方向
fCalMat	float[8][8]	力板校准矩阵（仅适用于原始模拟电压通道类型）
fCorners	float[4][3]	板角，在板坐标中，从板+x，+y 顺时针
iPlateType	int	加力板“类型”
iChannelDataType	int	0=校准的力数据，1=原始模拟电压
nChannels	int	通道数
szChannelNames	char[32][256]	通道名

3.15 错误码

名称	值	说明
eErrorNone	0	无错误
eErrorFailed	1	失败
eErrorNoInitializeSDK	2	未初始化 SDK
eErrorHandle	3	句柄错误
eErrorParameter	4	参数错误
eErrorBufferNoEnough	5	缓冲区太小
eErrorMemory	6	内存错误
eErrorSystemFailed	7	操作系统返回错误

eErrorNoldleResource	8	无可用资源
eErrorTimeOut	9	超时错误
eErrorFunctionVersionLow	10	功能版本低于服务器
eErrorFunctionVersionHigh	11	功能版本高于服务器
eErrorTaskNoRun	12	任务未启动
eErrorAlreadyRun	13	重复启动
eErrorConnectFailed	14	连接失败
eErrorNoResponse	15	服务器无响应(未启动服务端或者未开启 SDK 功能)

3.16 通知 sNotifyMsg

名称	类型	说明
nType	int	事件类型
nValue	int	动作类型
nParam1	float	参数 1
nParam2	float	参数 2
nParam3	float[3]	参数 3
nParam4	float [3]	参数 4
nTimeStamp	float[3][3]	时间戳, ms
sMsg	float[8][8]	通知消息描述

4.数据解析用例

读取 Marker 数据和刚体数据,在回调函数中:

(1) 获取所有的捕获的 MarkerSet 数量:

```
int markerSetNumber = data.nMarkerSets;
```

(2) 获取第 i 个 MarkerSet 的第 j 个 marker 位置信息:

```
float point = data.MocapData[i].Markers[j][0]; // 0-x 1-y 2-z
```

(3) 获取所有的刚体数量:

```
int number = data.nRigidBodyes;
```

(4) 获取第 i 个刚体的姿态信息 (位置, 旋转):

```
float point = data.RigidBodyes[i].x; // x y z qx qz qw
```

(5) 获取 NamedMarker 的数量:

```
int number = data.nLabeledMarkers;
```

(6) 获取第 i 个 NamedMarker 的位置信息:

```
float point = data.LabeledMarkers[i].x; // x y z
```

(7) 获取 UnNamedMarker 的数量:

```
int number = data.nOtherMarkers;
```

(8) 获取第 i 个 UnNamedMarker 的位置信息:

```
float point = data.OtherMarkers[i][0]; //0-x 1-y 2-z
```

(9) 获取模拟通道的数量:

```
int number = data.nAnalogdatas;
```

(10) 获取第 i 个模拟通道的数据:

```
float point = data.Analogdata[i];
```

(11) 获取帧发送时刻的绝对时间戳:

```
long long timestamp = data.iTimeStamp //1970-01-01 至今的毫秒数
```

NOKOV 度量动作捕捉系统



度量科技公众号

北京度量科技有限公司 010-64922321
www.nokov.com info@nokov.com
北京总部 北京市朝阳区安慧里四区15号五矿大厦8层820室
上海子公司 上海市长宁区通协路268号尚品都汇B201室
武汉分公司 湖北省武汉市东湖高新区武大航城二区A2栋1010
深圳分公司 深圳市宝安区满京华艺尚大厦4座301-A-035

