

Seeker_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. 建立连接与通信:

- (1) 创建 SeekerSDKClient对象，输入 Seeker 目标的IP 地址，如果与 SeekerSDKClient 运行在同一台电脑上，输入“10.1.1.198”，C++代码如下：

```
// 第一个参数为XINGYING软件的SDK广播IP，见面板Setting => NetWork => LocalAddress  
// ServerIP: “10.1.1.198”
```

```
SeekerSDKClient* theClient = new SeekerSDKClient();  
theClient->Initialize(“10.1.1.198”);
```

2. 接收数据:

- (1) 设置回调函数用于接收传输过来的数据，C++代码如下：

```
void DataHandler(sFrameOfMocapData* data, void* pUserData);  
theClient->SetDataCallback(DataHandler, theClient);
```

3. 数据解析: (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.nRigidBodies;
```

- (4) 获取第 i 个刚体的姿态信息（位置，旋转）:

```
float point = data.RigidBodies[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
```

4. 测力台使用:

(1) 初始化测力台:

// 等待测力台初始化就绪, 0ms表示无限等待

```
retCode = theClient->WaitForForcePlateInit(0);
```

(2) 设置测力台回调:

```
void ForcePlateHandler(sForcePlates* pForcePlate, void* pUserData);
```

```
theClient->SetForcePlateCallback(ForcePlateHandler, theClient);
```

(3) 解析测力台数据, 第i个测力台X轴上的力(N):

```
pForcePlate->ForcePlates[i].Fxyz[0];
```

5. 关闭连接:

```
theClient->Uninitialize();
```