Neuron Data Reader API 指导手册

By 何元会
唐宇峰

27.01.2016

NeuronDataReader b15

北京诺亦腾科技有限公司

本文档为开发人员阐述了如何使用 NeuronDataReader 库以及如何应用通过该库获得的骨骼数据。

Contents

1. 简介

北京诺亦腾科技有限公司研发的 Axis Neuron 软件可以通过 TCP/IP 或者 UDP 协议输出 BVH 动作数据和中间数据。 NeuronDataReader SDK 则是为简化大家从 Axis 获取数据而提供的编程接口。

1.1. NeuronDataReader 框架

NeuronDataReader 库的结构如下所示。

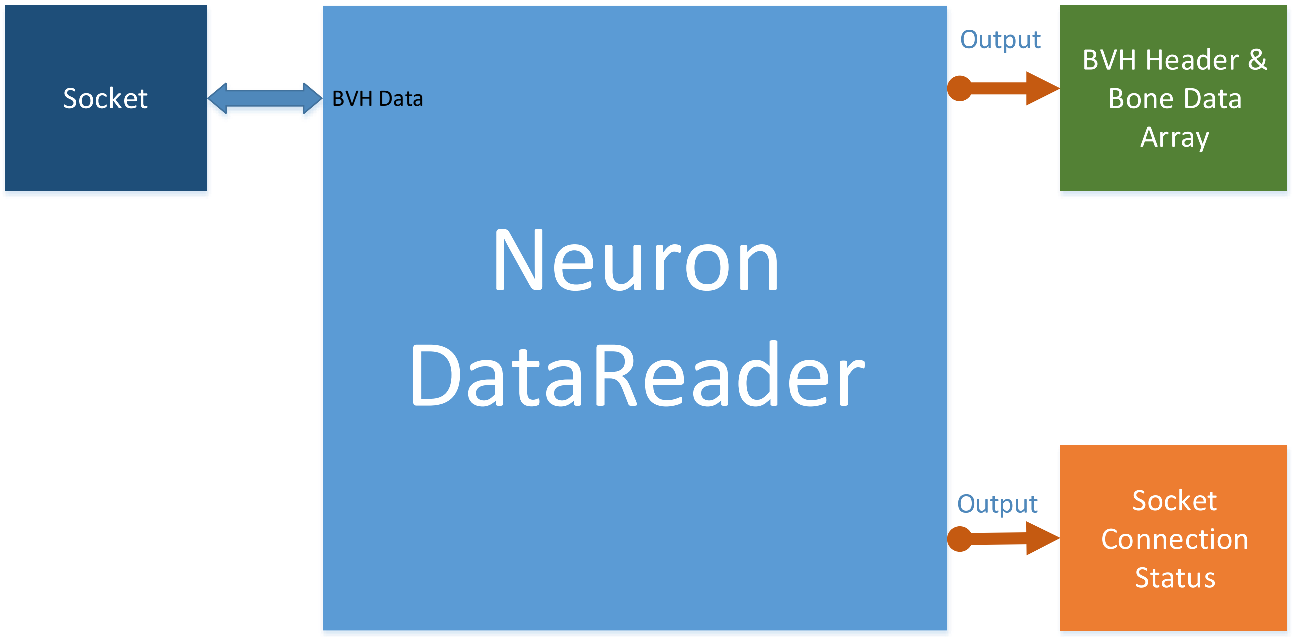


图. 1-1 NeuronDataReader 结构综述

为了兼容多种语言调用，NeuronDataReader SDK 为纯 C 函数接口。

1.2. 骨骼数据格式

在此版本中，骨骼数据包括了 BVH 数据和中间数据，均以回调的方式输出。

1.2.1. BVH 数据

NeuronDataReader 以回调方式将 BVH 动作数据传递到客户端代码。参数包含骨骼数据信息头及骨骼数据，在 float 数据阵列中骨骼数据的序列如 Appendix A 所示。Appendix B 展示了 BVH 数据头的样例，在实时数据流以供参考。

NeuronDataReader 通过网络从 Axis Neuron 获取 BVH 帧数据，每一帧中的 BVH 数据包含了 59 根骨骼的全部的动作数据。

对于带有位移（displacement）的 BVH 数据，每一根骨骼包含了 6 个 float 型数据：位移（X Y Z）和旋转（默认的旋转顺序为 Y X Z）。

对于不带位移（displacement）的 BVH 数据，只有根节点（Hip）包含位移和旋转数据。其余的每一根骨骼只包含了 3 个旋转数据（默认的旋转顺序为 Y X Z）。

因此，如果用户想获取指定骨骼的信息（位置或姿态），用户可以基于下面的公式来计算相关的编号索引。

1) 带有位移（displacement）的 BVH 数据

Displacement_X = bone index * 6 + 0
Displacement_Y = bone index * 6 + 1
Displacement_Z = bone index * 6 + 2

Rotation_Y = bone index * 6 + 3
Rotation_X = bone index * 6 + 4
Rotation_Z = bone index * 6 + 5

2) 不带位移（displacement）的 BVH 数据
除了旋转数据，只有根节点包含位移数据。
Root_Displacement_X = 0
Root_Displacement_Y = 1
Root_Displacement_Z = 2

Rotation_Y = 3 + bone index * 3 + 0
Rotation_X = 3 + bone index * 3 + 1
Rotation_Z = 3 + bone index * 3 + 2

3) 带有前缀（Reference）的 BVH 数据
为了转换或旋转全部的骨骼，但同时不改变骨骼数据，设计者在 BVH 数据结构中加入了一个新的节点，作为根节点的父节点。因此，只要改变这个新节点的位移，整个骨骼模型也会随之改变；只要改变这个新节点的姿态，就可以旋转整个骨骼模型。这个新节点被称之为前缀（Reference）。
前缀（Reference）中有位移和旋转数据，共计 6 个值。
所以，对于上述的 BVH 数据，如果包含了前缀，获取骨骼数据的索引号时，需要在计算时加 6（作为偏移量）。
BVH 数据结构参见 Appendix A。
注意：BVH 数据的位移量通常是常数，它体现了与 TPose 的父节点的初始位置相关的骨骼的初始位置的偏移量。因此它是一个常数。

1.2.2. 中间数据

NeuronDataReader 通过网络从 Axis Neuron 获取中间数据帧，每一帧中的中间数据包含了 21 根骨骼的全部的传感器数据和动作数据以及双脚的接触状态。
对于中间数据，每一根骨骼包含了 16 个 float 型数据，它们分别是：3 个位置数据（世界坐标系中的 X Y Z 轴）、3 个速度数据（世界坐标系中的 X Y Z 轴）、4 个四元数数据（世界坐标系中的 W X Y Z 轴）、3 个加速度数据（模块坐标系中的 X Y Z 轴）和陀螺仪的 3 个数据（模块坐标系中的 X Y Z 轴）。
中间数据格式如下所示：
Position_X = bone index * 16 + 0
Position_Y = bone index * 16 + 1
Position_Z = bone index * 16 + 2

Velocity_X = bone index * 16 + 3
Velocity_Y = bone index * 16 + 4
Velocity_Z = bone index * 16 + 5

Quaternion_W = bone index * 16 + 6
Quaternion_X = bone index * 16 + 7
Quaternion_Y = bone index * 16 + 8
Quaternion_Z = bone index * 16 + 9

Accelerated velocity_X = bone index * 16 + 10
Accelerated velocity_Y = bone index * 16 + 11
Accelerated velocity_Z = bone index * 16 + 12

Gyro_X = bone index * 16 + 13
Gyro_Y = bone index * 16 + 14
Gyro_Z = bone index * 16 + 15

在每一帧数据的最后，包含了左右脚的接触判断。“1”代表与地面接触，“0”代表不接触，该数据是 float 型的，4 字节。

所以，每一帧的中间数据包含了 21 * 16 + 2 = 338 (float)，共计 338 * 4 = 1352 字节。

每一帧的输出序列是从 No.1 到 No.21。骨骼列表和中间数据的输出序列参见 Appendix C 和 Appendix D。

1.3. 数据频率

从 NeuronDataReader 输出的数据的频率取决于当前用户穿戴的传感器的数量。我们规定，当所穿戴的设备节点小于 18 时，对应的采集频率为 120Hz；当所穿戴的设备节点大于等于 18 时，对应的采集频率为 60Hz。相应地，调用回调函数的频率与数据采集频率一致。

需要注意的是，在通过网络进行数据传输的过程中，存在丢帧的较小可能。因此，尽管频率是确定的，但是通过 NeuronDataReader 获得的数据编号可能会发生改变。

1.4. 命令与参数同步

NeuronDataReader 库主要用于读取和解析从服务器获得的数据。考虑到在客户端有大量的参数必须与服务器同步，我们加入了一些相关的 API 接口。

1.5. 用户协议

NeuronDataReader 使用回调的方式来输出数据。因此，在连接到服务器之前，必须注册一个函数用来接收数据。由于用于注册的函数通常为静态函数或全局函数，当注册回调函数时，可以传入一个自定义对象（第一个参数），以便函数被回调时可以获取注册该回调函数的对象。

NeuronDataReader 中的数据处理的线程从 UI 分离出的工作线程。因此用户注册的数据接收函数不能直接访问 UI 元素。但是回调函数的数据或者状态可以被保存至一个本地数组或者缓存，因此 UI 线程可以在其他任何地方访问本地缓存的数据。

NeuronDataReader 库中有一些用来与服务器同步参数或数据的命令。

由于 C#或者 Unity 无法直接调用 C++动态的库和 API，NeuronDataReader 采用了纯 C 的接口。

如果 NeuronDataReader 库用于 Mac 环境下的 C/C++项目，在预定义的配置项中，需要包含一个预定义的标识“__OS_XUN__”。

2. 使用指导

NeuronDataReader 库的数据类型、句柄和程序接口如下所列。

2.1. 数据类型定义

2.1.1. 跨平台数据类型

如果 NeuronDataReader 库用于 Mac 环境下的 C/C++项目，在预定义的配置项中，需要包含一个预定义的标识“__OS_XUN__”。

```
#ifdef __OS_XUN__           // Mac OS X, Linux or Unix like OS data type definition
#define __stdcall           // Empty
#endif

#ifndef NBOOL
#define NBOOL int
#endif

#ifndef TRUE
#define TRUE  1
#endif

#ifndef FALSE
#define FALSE 0
#endif
```

2.1.2.Socket 连接状态

socket 连接状态的枚举类型如下所示：Connected, Connecting, Disconnected.

```
// Socket status
typedef enum _SocketStatus
{
    CS_Running,              // Socket is working correctly
    CS_Starting,             // Is trying to start service
    CS_OffWork,              // Not working
}SocketStatus;
```

2.1.3.数据流的版本

对于不同版本的 NeuronDataReader，用于通讯的数据结构在定义和结构上会有改变。数据版本用来与旧版本的 NeuronDataReader 生成的数据相匹配。

```
// Data version
typedef union DataVersion
{
    uint32_t _VersionMask;
    struct
    {
        uint8_t BuildNumb;      // Build number
        uint8_t Revision;       // Revision number
        uint8_t Minor;          // Subversion number
        uint8_t Major;          // Major version number
    };
}DATA_VER;
```

2.1.4.BVH 数据流的头

```
// Header format of BVH data
typedef struct _BvhDataHeader
{
    uint16_t   Token1;           // Package start token: 0xDDFF
    DATA_VER   DataVersion;      // Version of community data format. e.g.: 1.0.0.2
    uint16_t   DataCount;        // Values count
    uint8_t    WithDisp;         // With/out displacement
    uint8_t    WithReference;    // With/out reference bone data at first
    uint32_t   AvatarIndex;      // Avatar index
    uint8_t    AvatarName[32];   // Avatar name
    uint32_t   FrameIndex;       // Frame data index
    uint32_t   Reserved;         // Reserved, only enable this package has 64bytes length
    uint32_t   Reserved1;        // Reserved, only enable this package has 64bytes length
    uint32_t   Reserved2;        // Reserved, only enable this package has 64bytes length
    uint16_t   Token2;           // Package end token: 0xEEFF
}BvhDataHeader;
```

2.1.5.Calc Data 数据流头

```
// Header format of BVH data
typedef struct _CalcDataHeader
{
    uint16_t   Token1;          // Package start token: 0x88FF
    DATA_VER   DataVersion;     // Version of community data format. e.g.: 1.0.0.3
    uint32_t   DataCount;       // Values count
    uint32_t   AvatarIndex;     // Avatar index
    uint8_t    AvatarName[32];  // Avatar name
    uint32_t   FrameIndex;      // Frame data index
    uint32_t   Reserved1;       // Reserved, only enable this package has 64bytes length
    uint32_t   Reserved2;       // Reserved, only enable this package has 64bytes length
    uint32_t   Reserved3;       // Reserved, only enable this package has 64bytes length
    uint16_t   Token2;          // Package end token: 0x99FF

}CalcDataHeader;
```

NeuronDataReader 库主要用于读取和解析从服务器获得的数据。每一帧的数据流包含了 BVH 头和 float 型的 BVH 动作数据。

BVH 头是一个 64 字节的头，包含了 BVH 数据的基本信息：是否含有位移或前缀。

BVH 动作数据是一个 float 型的数组。如果数据包含了前缀，那么前 6 个数据是前缀的位移和旋转数据，通常这些值为“0”。

如果数据包含了位移，每一根骨骼包含 6 个 float 型数据：3 个位移数据和 3 个旋转数据。对于不带位移（displacement）的 BVH 数据，只有根节点（Hip）包含位移和旋转数据。其余的每一根骨骼只包含了 3 个旋转数据（默认的旋转顺序为 Y X Z）。骨骼序列请参考 Appendix A。

BVH 数据属于树形结构，详细的 BVH 数据结构请参考 Appendix B。

2.2. 回调函数和注册回调

通过回调函数，NeuronDataReader 库输出骨骼数据或者 socket 状态。因此，当接收这些数据时，应该先注册 NeuronDataReader 库中相关的回调句柄。

2.2.1.回调 BVH 数据

```
    typedef void (CALLBACK *FrameDataReceived)(void* customedObj, SOCKET_REF sender, 
BvhDataHeader* header, float* data);
```

Parameters

customedObj

User defined object.

sender

Connector reference of TCP/IP client as identity.

header

BvhDataHeader type pointer, to output the BVH data format information.

data

Float type array pointer, to output binary data.

Remarks

The related information of the data stream can be obtained from BvhDataHeader.

2.2.2.回调中间数据

```
    typedef void (CALLBACK * CalculationDataReceived)(void* customedObj, SOCKET_REF sender, 
CalcDataHeader * header, float* data);
```

Parameters

customedObj

User defined object.

sender

Connector reference of TCP/IP client as identity.

Pack

CalcDataHeader type pointer, to output the calculation data format information.

data

Float type array pointer, to output binary data.

Remarks

The related information of the data stream can be obtained from CalcDataHeader.

2.2.3.Socket 状态回调

```
    typedef void (CALLBACK *SocketStatusChanged)(void* customedObj, SOCKET_REF sender,
SocketStatus status, char* message);
```

Parameters

customedObj

User defined object.

sender

Connector reference of TCP/IP client as identity.

status

Indicate the status changes of current socket.

message

Status description.

注意：由于 NeuronDataReader 中的数据处理是多线程异步的，数据接收回调函数不能直接访问 UI 元素。如果需要将数据用于 UI 线程，建议将从回调函数得到的数据保存至一个本地的数组。

2.3. API 使用指导

2.3.1.BRRegisterFrameDataCallback

注册 BVH 数据接收回调句柄。

```
// Register data-receiving callback handle.
BDR_API void BRRegisterFrameDataCallback(void* customedObj, FrameDataReceived handle);
```

Parameters

customedObj

User defined object.

handle

A function pointer of FrameDataReceived type.

Remarks

The handle of FrameDataReceived type points to the function address of the client.

2.3.2.BRRegisterCalculationDataCallback

Register the calculation data receiving callback handle:

```
// Register data-receiving callback handle.
BDR_API void BRRegisterCalculationDataCallback (void* customedObj, CalculationDataReceived
handle);
```

Parameters

customedObj

User defined object.

handle

A function pointer of CalculationDataReceived type.

Remarks

The handle of CalculationDataReceived type points to the function address of the client.

2.3.3.BRRegisterSocketStatusCallback

注册 socket 状态回调句柄。

```
// Register socket status callback
BDR_API void BRRegisterSocketStatusCallback (void* customedObj, SocketStatusChanged handle);
```

Parameters

customedObj

User defined object.

handle

A function pointer.

Remarks

The handle of SocketStatusChanged type points to the function address of the client.

2.3.4.BRConnectTo

由给定的 IP 地址和端口连接至服务器。

```
// Connect to server
BDR_API SOCKET_REFBRConnectTo(char* serverIP, int nPort);
```

Parameters

serverIP

Server’s IP address.

nPort

Server’s port.

Return Values

If connected successfully, return a handle of socket as its identity; otherwise NULL is returned.

2.3.5.BRStartUDPServiceAt

由于 Axis Neuron 可以通过 TCP/IP 或者 UDP 输出数据，NeuronDataReader 也可以读取和解析两种 socket 数据类型。函数 BRStartUDPServiceAt 用来开始一个监听和接收由服务器发出的数据的服务。

```
// Start a UDP service to receive data at 'nPort'
BDR_API SOCKET_REF BRStartUDPServiceAt(int nPort);
```

2.3.6.BRCloseSocket

停止数据接收服务。值得注意的是，需要在程序退出前调用这个函数来断开/停止服务器的服务，否则程序将由于数据接收线程的阻塞而不能退出。

```
// Stop service
BDR_API void BRCloseSocket (SOCKET_REF sockRef);
```

2.3.7.BRGetSocketStatus

检查 socket 状态。实际上通过 socket 回调句柄，这个函数有相同的输出状态。如果已经注册 socket 状态回调句柄，这个函数就不需要使用了。

```
// Check connect status
BDR_API SocketStatus BRGetSocketStatus (SOCKET_REF sockRef);
```

Return Values

Return the status of referred socket.

2.3.8.BRGetLastErrorMessage

一旦出现错误，通过调用'BRGetLastErrorMessage'函数来获取错误信息。

```
BDR_API char* BRGetLastErrorMessage();
```

Return Values

Return the last error message.

Remarks

The error information can be acquired by calling 'BRGetLastErrorMessage' once error occurred during function callback.

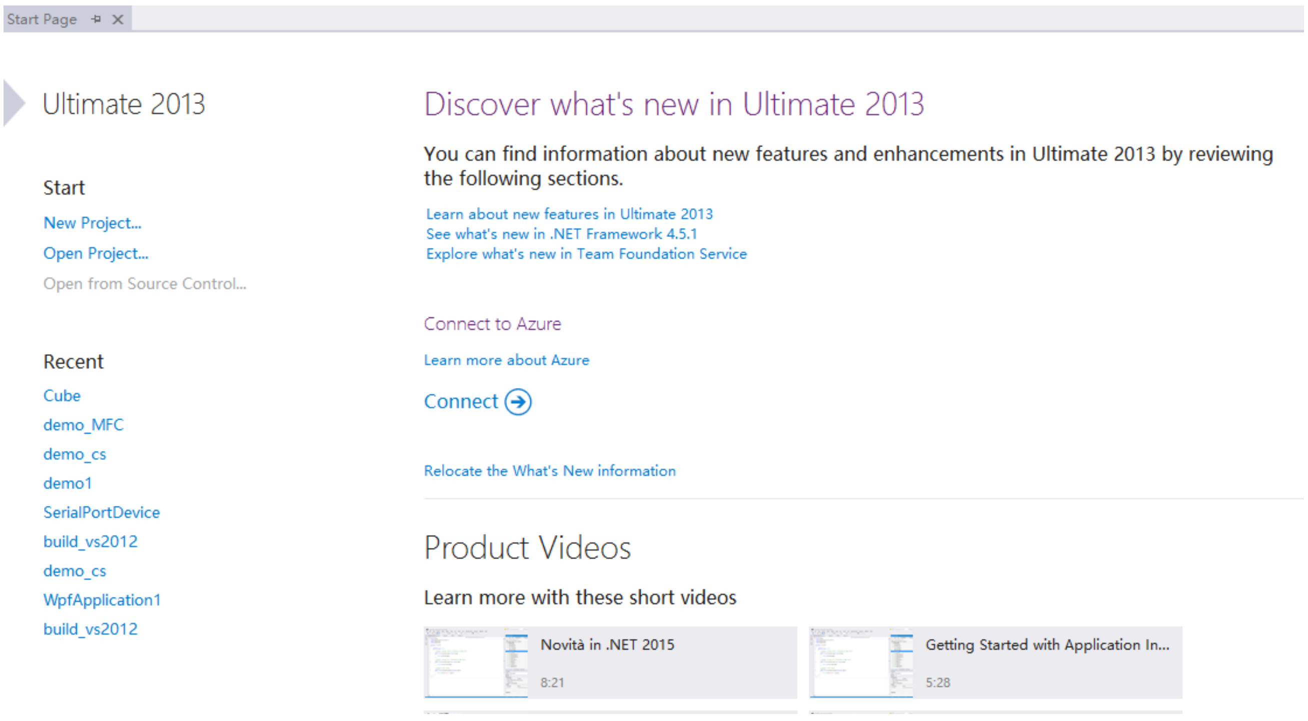
3. Demo 示例

NeuronDataReader 库支持绝大多数常见的开发环境，例如 C/C++/MFC、WPF/C#、Mac Cocoa 和 Unity、Unigine 等游戏引擎。

3.1. C++ demo

以下将介绍如何在 C++环境中使用 NeuronDataReader 库。

1. 在 Windows 平台下，进入 Visual Studio 2013，选择起始页面里的“New Project...”。



Start Page

Ultimate 2013

Start
New Project...
Open Project...
Open from Source Control...

Recent
Cube
demo_MFC
demo_cs
demo1
SerialPortDevice
build_vs2012
demo_cs
WpfApplication1
build_vs2012

Discover what's new in Ultimate 2013

You can find information about new features and enhancements in Ultimate 2013 by reviewing the following sections.

Learn about new features in Ultimate 2013
See what's new in .NET Framework 4.5.1
Explore what's new in Team Foundation Service

Connect to Azure
Learn more about Azure
Connect

Relocate the What's New information

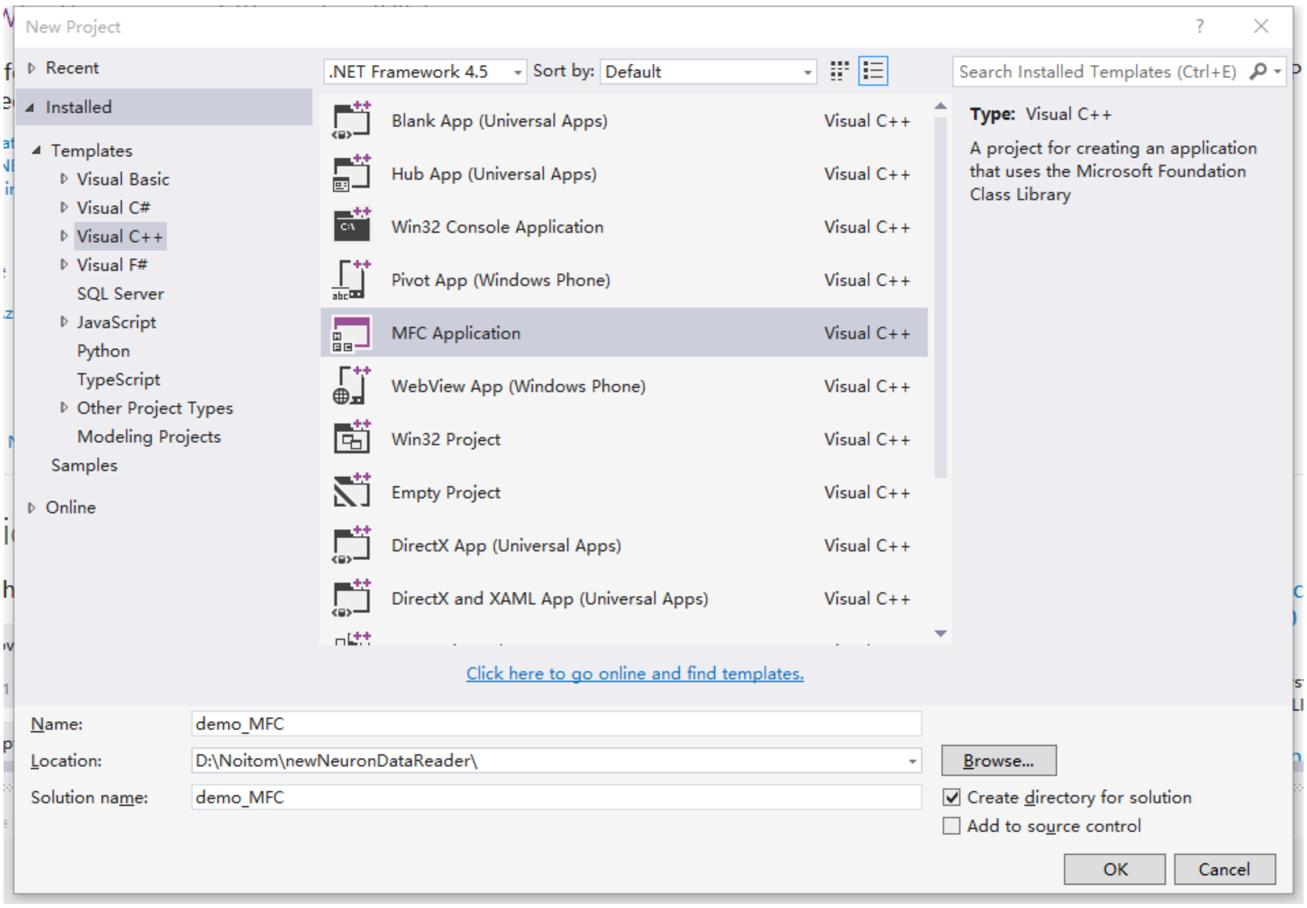
Product Videos

Learn more with these short videos

Novità in .NET 2015 8:21
Getting Started with Application In... 5:28

图. 3-1

2. 选择模板列表里的 Visual C++--> MFC Application，将项目命名为“demo_MFC”。



New Project

- Recent
- Installed
  - Templates
    - Visual Basic
    - Visual C#
    - Visual C++
    - Visual F#
    - SQL Server
    - JavaScript
    - Python
    - TypeScript
    - Other Project Types
    - Modeling Projects
  - Samples
- Online

.NET Framework 4.5 · Sort by: Default · Search Installed Templates (Ctrl+E)

Blank App (Universal Apps)	Visual C++
Hub App (Universal Apps)	Visual C++
Win32 Console Application	Visual C++
Pivot App (Windows Phone)	Visual C++
MFC Application	Visual C++
WebView App (Windows Phone)	Visual C++
Win32 Project	Visual C++
Empty Project	Visual C++
DirectX App (Universal Apps)	Visual C++
DirectX and XAML App (Universal Apps)	Visual C++

Type: Visual C++

A project for creating an application that uses the Microsoft Foundation Class Library

Click here to go online and find templates.

Name: demo_MFC

Location: D:\Noitom\newNeuronDataReader\

Solution name: demo_MFC

Browse...

[✓] Create directory for solution

[ ] Add to source control

OK Cancel

图. 3-2

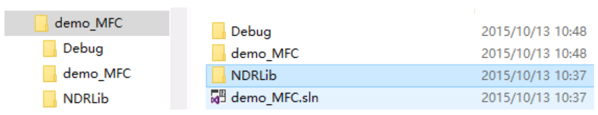
3.点击 OK，将会建立一个空的 MFC 项目。

4.选择 dialog application type.

5.完成建立的步骤。成功建立一个基于对话框架构应用程序。

为了使用 NeuronDataReader SDK，用户还需要包含 NeuronDataReader 的头文件，并在编译的时候链接 NeuronDataReader 的库。

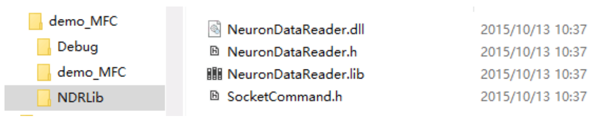
NeuronDataReader SDK 文件需要放在 demo 的根目录下。



- demo_MFC
  - Debug
  - demo_MFC
  - NDRLib

Debug	2015/10/13 10:48
demo_MFC	2015/10/13 10:48
NDRLib	2015/10/13 10:37
demo_MFC.sln	2015/10/13 10:37

NeuronDataReader 文件：



- demo_MFC
  - Debug
  - demo_MFC
  - NDRLib

NeuronDataReader.dll	2015/10/13 10:37
NeuronDataReader.h	2015/10/13 10:37
NeuronDataReader.lib	2015/10/13 10:37
SocketCommand.h	2015/10/13 10:37

相关的代码如下所示。

demo_MFCDlg.h 文件：

```
// demo_MFCDlg.h : header file
//

#pragma once

// Include the NeuronDataReader head file
#include "../NDRLib/NeuronDataReader.h"

#include "afxwin.h"

#define WM_UPDATE_MESSAGE (WM_USER+200)

// Cdemo_MFCDlg dialog
class Cdemo_MFCDlg : public CDialogEx
{
// Construction
public:
	Cdemo_MFCDlg(CWnd* pParent = NULL);	// standard constructor

	enum
	{
		BVHBoneCount = 59,
		CalcBoneCount = 21,
	};
// Dialog Data
	enum { IDD = IDD_DEMO_MFC_DIALOG };

	protected:
	virtual void DoDataExchange(CDataExchange* pDX);	// DDX/DDV support

	static void __stdcall bvhFrameDataReceived(void* customedObj, SOCKET_REF sender, BvhDataHeaderEx* header,
float* data);

	// Receive calculation data
	static void __stdcall CalcFrameDataReceive( void* customObje, SOCKET_REF sender, CalcDataHeader* header,
float* data );
```

```
// Implementation
protected:
	HICON m_hIcon;

	SOCKET_REF sockTCPRef;
	SOCKET_REF sockUDPRef;

	void showBvhBoneInfo(SOCKET_REF sender, BvhDataHeaderEx* header, float* data);

	void showCalcBoneInfo(SOCKET_REF sender, CalcDataHeader* header, float* data );

	// Generated message map functions
	virtual BOOL OnInitDialog();
	afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
	afx_msg void OnPaint();
	afx_msg HCURSOR OnQueryDragIcon();

	afx_msg LRESULT OnUpdateMessage( WPARAM wParam, LPARAM lParam );
	DECLARE_MESSAGE_MAP()
public:
	CComboBox m_wndComBoxBone;

	CString m_strIPAddress;
	CString m_strPort;
	CString m_strUDPPort;

	afx_msg void OnBnClickedOk();
	afx_msg void OnBnClickedCancel();
	afx_msg void OnBnClickedButtonTcpConnection();
	afx_msg void OnBnClickedButtonUdpConnection();
	afx_msg void OnCbnSelchangeComboSelectionBoneIndex();
	CButton m_radioBvh;
	afx_msg void OnBnClickedRadio();

	void UpdateBvhDataShowUI();
	void UpdateCalcDataShowUI();
	CString m_clacAx;
	CString m_calcAy;
	CString m_calcAz;
	CString m_calcGx;
	CString m_calcGy;
	CString m_calcGz;
	CString m_calcPx;
	CString m_calcPy;
	CString m_calcQs;
	CString m_calcQx;
	CString m_calcQy;
	CString m_calcQz;
	CString m_calcVx;
	CString m_calcVy;
	CString m_calcVz;
	CString m_calcPz;
	CString m_bvhAngleZ;
};
```

demo_MFCDlg.cpp 文件：

```
// demo_MFCDlg.cpp : implementation file

#include "stdafx.h"
#include "demo_MFC.h"
#include "demo_MFCDlg.h"
#include "afxdialogex.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// Load SDK library
#pragma comment(lib, "../NDRLib/NeuronDataReader.lib")//Add Lib


// CAboutDlg dialog used for App About

class CAboutDlg : public CDialogEx
{
public:
	CAboutDlg();

// Dialog Data
	enum { IDD = IDD_ABOUTBOX };

	protected:
	virtual void DoDataExchange(CDataExchange* pDX);	// DDX/DDV support

// Implementation
protected:
	DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
```

```
END_MESSAGE_MAP()


// Cdemo_MFCDlg dialog

Cdemo_MFCDlg::Cdemo_MFCDlg(CWnd* pParent /*=NULL*/)
	: CDialogEx(Cdemo_MFCDlg::IDD, pParent)
	, m_strIPAddress(_T(""))
	, m_strPort(_T(""))
	, m_strUDPPort(_T(""))

	// Calc数据显示控件
	, m_calcPx( _T( "" ) )
	, m_calcPy( _T( "" ) )
	, m_calcPz( _T( "" ) )

	, m_calcVx( _T( "" ) )
	, m_calcVy( _T( "" ) )
	, m_calcVz( _T( "" ) )

	, m_calcQs( _T( "" ) )
	, m_calcQx( _T( "" ) )
	, m_calcQy( _T( "" ) )
	, m_calcQz( _T( "" ) )

	, m_clacAx( _T( "" ) )
	, m_calcAy( _T( "" ) )
	, m_calcAz( _T( "" ) )

	, m_calcGx( _T( "" ) )
	, m_calcGy( _T( "" ) )
	, m_calcGz( _T( "" ) )

	, m_bvhAngleZ( _T( "" ) )
{
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);

	sockTCPRef = NULL;
	sockUDPRef = NULL;

}

void Cdemo_MFCDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialogEx::DoDataExchange( pDX );
	DDX_Text( pDX, IDC_TEXT1, m_strIPAddress );
	DDX_Text( pDX, IDC_TEXT2, m_strPort );
	DDX_Text( pDX, IDC_UDP, m_strUDPPort );
	DDX_Control( pDX, IDC_RADIO_BVH, m_radioBvh );
	DDX_Text( pDX, IDC_STATIC_AX, m_clacAx );
	DDX_Text( pDX, IDC_STATIC_AY, m_calcAy );
	DDX_Text( pDX, IDC_STATIC_AZ, m_calcAz );
	DDX_Text( pDX, IDC_STATIC_GX, m_calcGx );
	DDX_Text( pDX, IDC_STATIC_GY, m_calcGy );
	DDX_Text( pDX, IDC_STATIC_GZ, m_calcGz );
	DDX_Text( pDX, IDC_STATIC_PX, m_calcPx );
	DDX_Text( pDX, IDC_STATIC_PY, m_calcPy );
	DDX_Text( pDX, IDC_STATIC_QS, m_calcQs );
	DDX_Text( pDX, IDC_STATIC_QX, m_calcQx );
	DDX_Text( pDX, IDC_STATIC_QY, m_calcQy );
	DDX_Text( pDX, IDC_STATIC_QZ, m_calcQz );
	DDX_Text( pDX, IDC_STATIC_VX, m_calcVx );
	DDX_Text( pDX, IDC_STATIC_VY, m_calcVy );
	DDX_Text( pDX, IDC_STATIC_VZ, m_calcVz );
	DDX_Text( pDX, IDC_STATIC_PZ, m_calcPz );
	DDX_Control( pDX, IDC_COMBO_SELECTION_BONE_INDEX, m_wndComBoxBone );
	DDX_Text( pDX, IDC_STATIC_ANGLE_Z, m_bvhAngleZ );
}

BEGIN_MESSAGE_MAP(Cdemo_MFCDlg, CDialogEx)
	ON_WM_SYSCOMMAND()
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_BN_CLICKED(IDOK, &Cdemo_MFCDlg::OnBnClickedOk)
	ON_BN_CLICKED(IDCANCEL, &Cdemo_MFCDlg::OnBnClickedCancel)
	ON_BN_CLICKED(IDC_BUTTON_TCP_CONNECTION, &Cdemo_MFCDlg::OnBnClickedButtonTcpConnection)
	ON_BN_CLICKED(IDC_BUTTON_UDP_CONNECTION, &Cdemo_MFCDlg::OnBnClickedButtonUdpConnection)
	ON_CBN_SELCHANGE(IDC_COMBO_SELECTION_BONE_INDEX,
&Cdemo_MFCDlg::OnCbnSelchangeComboSelectionBoneIndex)
	ON_BN_CLICKED( IDC_RADIO_BVH, &Cdemo_MFCDlg::OnBnClickedRadio )
	ON_BN_CLICKED( IDC_RADIO_CALC, &Cdemo_MFCDlg::OnBnClickedRadio )
	ON_MESSAGE( WM_UPDATE_MESSAGE, OnUpdateMessage )
END_MESSAGE_MAP()


// Cdemo_MFCDlg message handlers

BOOL Cdemo_MFCDlg::OnInitDialog()
```

```
{
	CDialogEx::OnInitDialog();

	// Add "About..." menu item to system menu.

	// IDM_ABOUTBOX must be in the system command range.
	ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
	ASSERT(IDM_ABOUTBOX < 0xF000);

	CMenu* pSysMenu = GetSystemMenu(FALSE);
	if (pSysMenu != NULL)
	{
		BOOL bNameValid;
		CString strAboutMenu;
		bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
		ASSERT(bNameValid);
		if (!strAboutMenu.IsEmpty())
		{
			pSysMenu->AppendMenu(MF_SEPARATOR);
			pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
		}
	}

	// Set the icon for this dialog.   The framework does this automatically
	//   when the application's main window is not a dialog
	SetIcon(m_hIcon, TRUE);			// Set big icon
	SetIcon(m_hIcon, FALSE);		// Set small icon

	// TODO: Add extra initialization here
	BRRegisterFrameDataCallback(this, bvhFrameDataReceived);

	BRRegisterCalculationDataCallback(this, CalcFrameDataReceive);

	m_strIPAddress = L"127.0.0.1";//Default IP Address
	m_strPort = L"7001";//Default Port

	m_strUDPPort = L"7001"; // Default UDP Port

	m_radioBvh.SetCheck( BST_CHECKED );
	UpdateBvhDataShowUI();
	UpdateData(FALSE);

	return TRUE;  // return TRUE   unless you set the focus to a control
}
```

```
void __stdcall Cdemo_MFCDlg::bvhFrameDataReceived(void* customedObj, SOCKET_REF sender, BvhDataHeaderEx*
header, float* data)
{
	Cdemo_MFCDlg* pthis = (Cdemo_MFCDlg*)customedObj;
	pthis->showBvhBoneInfo(sender, header, data);
}

void __stdcall Cdemo_MFCDlg::CalcFrameDataReceive( void* customObje, SOCKET_REF sender, CalcDataHeader*
header, float* data )
{
	Cdemo_MFCDlg* pthis = (Cdemo_MFCDlg*)customObje;
	pthis->showCalcBoneInfo( sender, header, data );

}

void Cdemo_MFCDlg::showBvhBoneInfo(SOCKET_REF sender, BvhDataHeaderEx* header, float* data)
{
	USES_CONVERSION;

	// show frame index
	char strFrameIndex[60];
	itoa(header->FrameIndex, strFrameIndex, 10);
	GetDlgItem(IDC_STATIC_FRAME_INDEX)->SetWindowText(A2W(strFrameIndex));

	// calculate data index for selected bone
	int dataIndex = 0;
	int curSel = m_wndComBoxBone.GetCurSel();
	if ( curSel == CB_ERR ) return;

	if (header->WithDisp)
	{
		dataIndex = curSel * 6;
		if (header->WithReference)
		{
			dataIndex += 6;
		}

		float dispX = data[dataIndex + 0];
		float dispY = data[dataIndex + 1];
		float dispZ = data[dataIndex + 2];

		char strBuff[32];
		sprintf_s(strBuff, sizeof(strBuff), "%0.3f", dispX);
```

```
        GetDlgItem(IDC_STATIC_DISP_X)->SetWindowText(A2W(strBuff));

        sprintf_s(strBuff, sizeof(strBuff), "%0.3f", dispY);
        GetDlgItem(IDC_STATIC_DISP_Y)->SetWindowText(A2W(strBuff));

        sprintf_s(strBuff, sizeof(strBuff), "%0.3f", dispZ);
        GetDlgItem(IDC_STATIC_DISP_Z)->SetWindowText(A2W(strBuff));

        float angX = data[dataIndex + 4];
        float angY = data[dataIndex + 3];
        float angZ = data[dataIndex + 5];

        sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angX);
        GetDlgItem(IDC_STATIC_ANGLE_X)->SetWindowText(A2W(strBuff));

        sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angY);
        GetDlgItem(IDC_STATIC_ANGLE_Y)->SetWindowText(A2W(strBuff));

        sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angZ);
        GetDlgItem(IDC_STATIC_ANGLE_Z)->SetWindowText(A2W(strBuff));

    }
    else
    {
        if (curSel == 0)
        {
            dataIndex = 0;
            if (header->WithReference)
            {
                dataIndex += 6;
            }

            // show hip's displacement
            float dispX = data[dataIndex + 0];
            float dispY = data[dataIndex + 1];
            float dispZ = data[dataIndex + 2];

            char strBuff[32];
            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", dispX);
            GetDlgItem(IDC_STATIC_DISP_X)->SetWindowText(A2W(strBuff));

            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", dispY);
            GetDlgItem(IDC_STATIC_DISP_Y)->SetWindowText(A2W(strBuff));
            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", dispZ);
            GetDlgItem(IDC_STATIC_DISP_Z)->SetWindowText(A2W(strBuff));

            // show hip's angle
            float angX = data[dataIndex + 4];
            float angY = data[dataIndex + 3];
            float angZ = data[dataIndex + 5];

            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angX);
            GetDlgItem(IDC_STATIC_ANGLE_X)->SetWindowText(A2W(strBuff));

            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angY);
            GetDlgItem(IDC_STATIC_ANGLE_Y)->SetWindowText(A2W(strBuff));

            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angZ);
            GetDlgItem(IDC_STATIC_ANGLE_Z)->SetWindowText(A2W(strBuff));
        }
        else
        {
            dataIndex = 3 + curSel * 3;
            if (header->WithReference)
            {
                dataIndex += 6;
            }

            // show displacement
            char strBuff[32];
            sprintf_s(strBuff, sizeof(strBuff), "NaN");
            GetDlgItem(IDC_STATIC_DISP_X)->SetWindowText(A2W(strBuff));
            GetDlgItem(IDC_STATIC_DISP_Y)->SetWindowText(A2W(strBuff));
            GetDlgItem(IDC_STATIC_DISP_Z)->SetWindowText(A2W(strBuff));

            // show angle
            float angX = data[dataIndex + 1];
            float angY = data[dataIndex + 0];
            float angZ = data[dataIndex + 2];

            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angX);
            GetDlgItem(IDC_STATIC_ANGLE_X)->SetWindowText(A2W(strBuff));

            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angY);
            GetDlgItem(IDC_STATIC_ANGLE_Y)->SetWindowText(A2W(strBuff));

            sprintf_s(strBuff, sizeof(strBuff), "%0.3f", angZ);
```

```
                        GetDlgItem(IDC_STATIC_ANGLE_Z)->SetWindowText(A2W(strBuff));
                }
        }
}

LRESULT Cdemo_MFCDlg::OnUpdateMessage( WPARAM wParam, LPARAM lParam )
{
        UpdateData( FALSE );
        return 0;
}

void Cdemo_MFCDlg::showCalcBoneInfo( SOCKET_REF sender, CalcDataHeader* header, float* data )
{
        USES_CONVERSION;
        // show frame index
        char strFrameIndex[60];
        itoa( header->FrameIndex, strFrameIndex, 10 );
        GetDlgItem( IDC_STATIC_FRAME_INDEX )->SetWindowText( A2W( strFrameIndex ) );

        int dataIndex = 0;
        int curSel = m_wndComBoxBone.GetCurSel();
        if ( curSel == CB_ERR ) return;

        if ( curSel > CalcBoneCount ) return;

        dataIndex = 16 * curSel;
        //CString tmptmp;
        //tmptmp.Format( L"%d\n", dataIndex );
        //OutputDebugString( tmptmp );

        CString tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 0] );
        m_calcPx = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 1] );
        m_calcPy = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 2] );
        m_calcPz = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 3] );
        m_calcVx = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 4] );
        m_calcVy = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 5] );
        m_calcVz = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 6] );
        m_calcQs = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 7] );
        m_calcQx = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 8] );
        m_calcQy = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 9] );
        m_calcQz = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 10] );
        m_clacAx = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 11] );
        m_calcAy = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 12] );
        m_calcAz = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 13] );
        m_calcGx = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 14] );
        m_calcGy = tmpData;
        tmpData.Format( L"%0.3f", data[dataIndex + 15] );
        m_calcGz = tmpData;

        PostMessage(WM_UPDATE_MESSAGE,0,0);//OK  —  UpdateDate

}

void Cdemo_MFCDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        if ((nID & 0xFFF0) == IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout.DoModal();
        }
        else
        {
                CDialogEx::OnSysCommand(nID, lParam);
        }
}

// If you add a minimize button to your dialog, you will need the code below
//    to draw the icon.    For MFC applications using the document/view model,
//    this is automatically done for you by the framework.

void Cdemo_MFCDlg::OnPaint()
{
        if (IsIconic())
        {
                CPaintDC dc(this); // device context for painting

                SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics(SM_CXICON);
                int cyIcon = GetSystemMetrics(SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width() - cxIcon + 1) / 2;
                int y = (rect.Height() - cyIcon + 1) / 2;

                // Draw the icon
                dc.DrawIcon(x, y, m_hIcon);
        }
        else
        {
                CDialogEx::OnPaint();
        }
}

// The system calls this function to obtain the cursor to display while the user drags
//    the minimized window.
HCURSOR Cdemo_MFCDlg::OnQueryDragIcon()
{
        return static_cast<HCURSOR>(m_hIcon);
}

void Cdemo_MFCDlg::OnBnClickedOk()
{
        //CDialogEx::OnOK();

}

void Cdemo_MFCDlg::OnBnClickedCancel()
{
        CDialogEx::OnCancel();
}

void Cdemo_MFCDlg::OnBnClickedButtonTcpConnection()
{
        UpdateData();
```

```
	if (sockTCPRef)
	{
		// close socket
		BRCloseSocket(sockTCPRef);

		// reconnect
		sockTCPRef = NULL;

		// change the title of button
		GetDlgItem(IDC_BUTTON_TCP_CONNECTION)->SetWindowText(L"Connect");
	}
	else
	{
		USES_CONVERSION;

		// get port number
		char* port = W2A(m_strPort);
		long nPort = 0;
		try
		{
			nPort = atoi(port);
		}
		catch (CException* e)
		{
			AfxMessageBox(L"Port number error", MB_OK);
			return;
		}

		// connect to remote server
		sockTCPRef = BRConnectTo(W2A(m_strIPAddress), nPort);

		// if success, change the title of button
		if (sockTCPRef)
		{
			GetDlgItem(IDC_BUTTON_TCP_CONNECTION)->SetWindowText(L"Disconnect");
		}
		else
		{
			// if failed, show message
			AfxMessageBox(A2W(BRGetLastErrorMessage()), MB_OK);
		}
	}
}
```

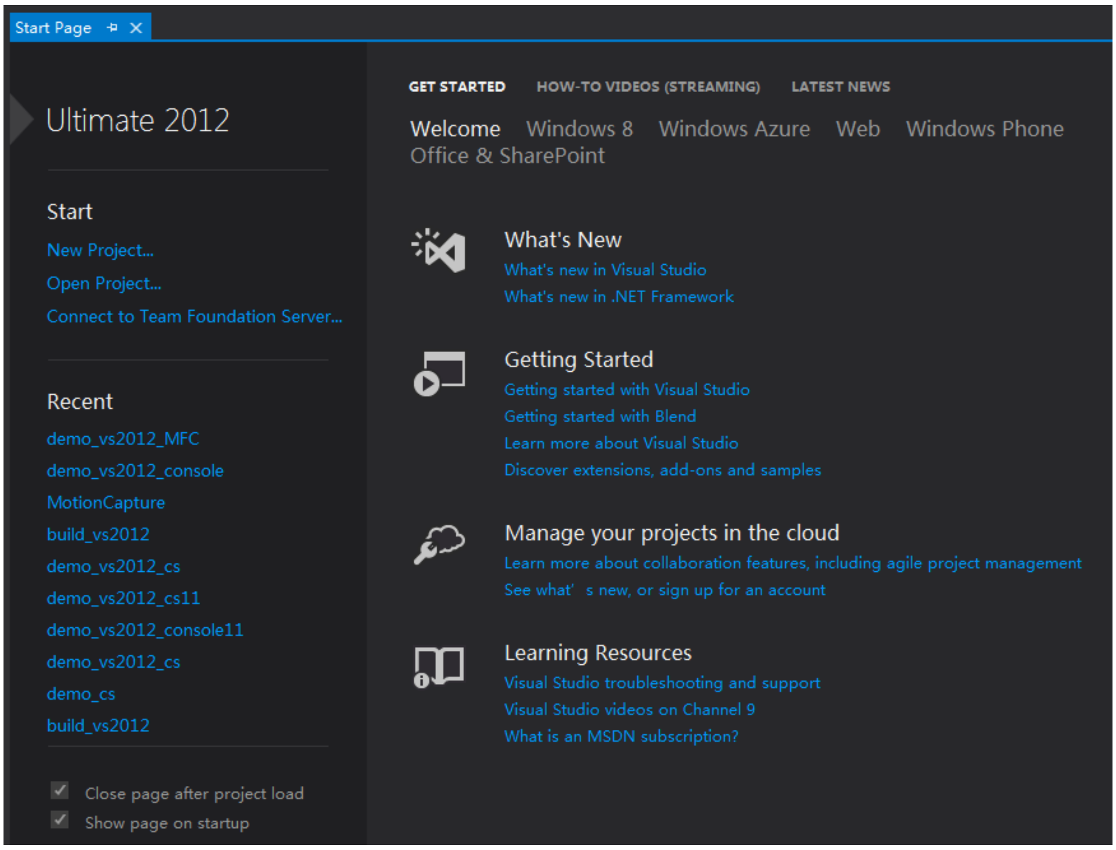
```
void Cdemo_MFCDlg::OnBnClickedButtonUdpConnection()
{
	UpdateData();

	if (sockUDPRef)
	{
		// close socket
		BRCloseSocket(sockUDPRef);

		// reconnect
		sockUDPRef = NULL;

		// change the title of button
		GetDlgItem(IDC_BUTTON_UDP_CONNECTION)->SetWindowText(L"Connect");
	}
	else
	{
		USES_CONVERSION;

		// get port number
		char* UDPport = W2A(m_strUDPPort);
		long nUDPPort = 0;
		try
		{
			nUDPPort = atoi(UDPport);
		}
		catch (CException* e)
		{
			AfxMessageBox(L"UDPPort number error", MB_OK);
			return;
		}

		// connect to remote server
		sockUDPRef = BRStartUDPServiceAt(nUDPPort);

		// if success, change the title of button
		if (sockUDPRef)
		{
			GetDlgItem(IDC_BUTTON_UDP_CONNECTION)->SetWindowText(L"Disconnect");
		}
		else
		{
			// if failed, show message
```

```
                AfxMessageBox(A2W(BRGetLastErrorMessage()), MB_OK);
            }
        }
}


void Cdemo_MFCDlg::OnCbnSelchangeComboSelectionBoneIndex()
{
        UpdateData();
}


void Cdemo_MFCDlg::OnBnClickedRadio()
{
        if (m_radioBvh.GetCheck() == BST_CHECKED)
        {
                UpdateBvhDataShowUI();
        }
        else
        {
                UpdateCalcDataShowUI();
        }
}

void Cdemo_MFCDlg::UpdateBvhDataShowUI()
{
        CString BoneID;
        m_wndComBoxBone.ResetContent();
        for ( int i = 0; i < BVHBoneCount; i++ )
        {
                BoneID.Format( L"%s%d", L"Bone", i );
                m_wndComBoxBone.AddString( BoneID );
        }
        m_wndComBoxBone.SetCurSel( 0 );
}

void Cdemo_MFCDlg::UpdateCalcDataShowUI()
{
        CString BoneID;
        m_wndComBoxBone.ResetContent();
        for ( int i = 0; i < CalcBoneCount; i++ )
        {
                BoneID.Format( L"%s%d", L"Bone", i );
                m_wndComBoxBone.AddString( BoneID );
        }
        m_wndComBoxBone.SetCurSel( 0 );
}
```

3.2. C Sharp demo

以下将介绍如何在 C#环境中使用 NeuronDataReader 库。

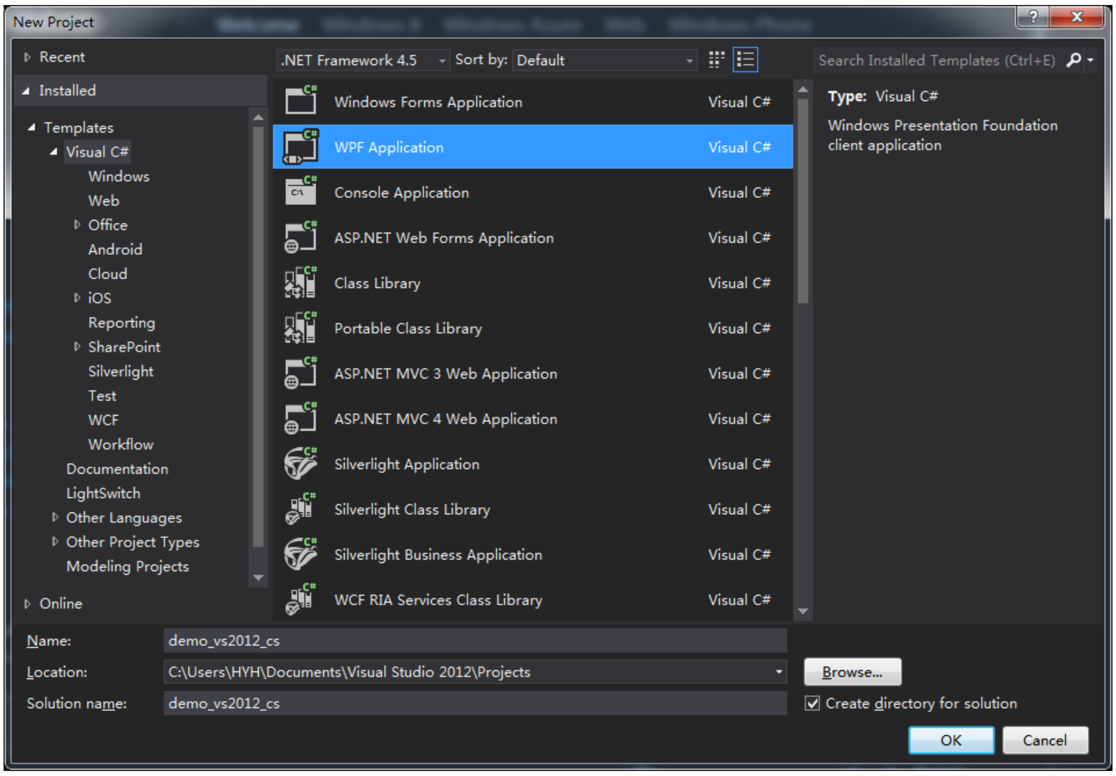
1. 在 Windows 环境下，进入 Visual Studio 2012，选择起始页面里的“New Project…”。



Start Page
Ultimate 2012
Start
New Project...
Open Project...
Connect to Team Foundation Server...
Recent
demo_vs2012_MFC
demo_vs2012_console
MotionCapture
build_vs2012
demo_vs2012_cs
demo_vs2012_cs11
demo_vs2012_console11
demo_vs2012_cs
demo_cs
build_vs2012
Close page after project load
Show page on startup
GET STARTED
HOW-TO VIDEOS (STREAMING)
LATEST NEWS
Welcome
Windows 8
Windows Azure
Web
Windows Phone
Office & SharePoint
What's New
What's new in Visual Studio
What's new in .NET Framework
Getting Started
Getting started with Visual Studio
Getting started with Blend
Learn more about Visual Studio
Discover extensions, add-ons and samples
Manage your projects in the cloud
Learn more about collaboration features, including agile project management
See what' s new, or sign up for an account
Learning Resources
Visual Studio troubleshooting and support
Visual Studio videos on Channel 9
What is an MSDN subscription?

图. 3-3

2. 选择模板列表里的 Visual C#--> WPF Application，将项目命名为“demo_cs”。


New Project
Recent
Installed
Templates
Visual C#
Windows
Web
Office
Android
Cloud
iOS
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
Documentation
LightSwitch
Other Languages
Other Project Types
Modeling Projects
Online
Sort by:
Search Installed Templates (Ctrl+E)
Windows Forms Application Visual C#
WPF Application Visual C#
Console Application Visual C#
ASP.NET Web Forms Application Visual C#
Class Library Visual C#
Portable Class Library Visual C#
ASP.NET MVC 3 Web Application Visual C#
ASP.NET MVC 4 Web Application Visual C#
Silverlight Application Visual C#
Silverlight Class Library Visual C#
Silverlight Business Application Visual C#
WCF RIA Services Class Library Visual C#
Browse...
OK
Cancel

.NET Framework 4.5

Sort by: Default

Type: Visual C#

Windows Presentation Foundation client application

Name: demo_vs2012_cs

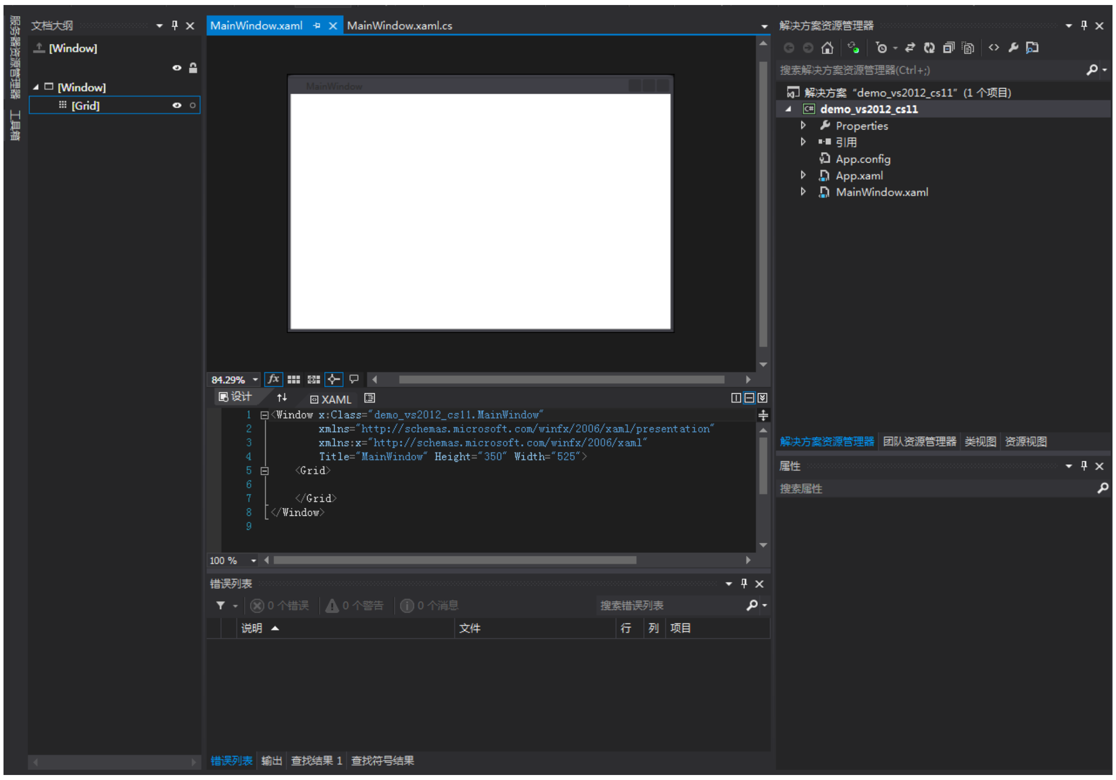
Location: C:\Users\HYH\Documents\Visual Studio 2012\Projects

Solution name: demo_vs2012_cs

☑ Create directory for solution

图. 3-4

3.　点击 OK，将会建立一个空的 WPF 项目。



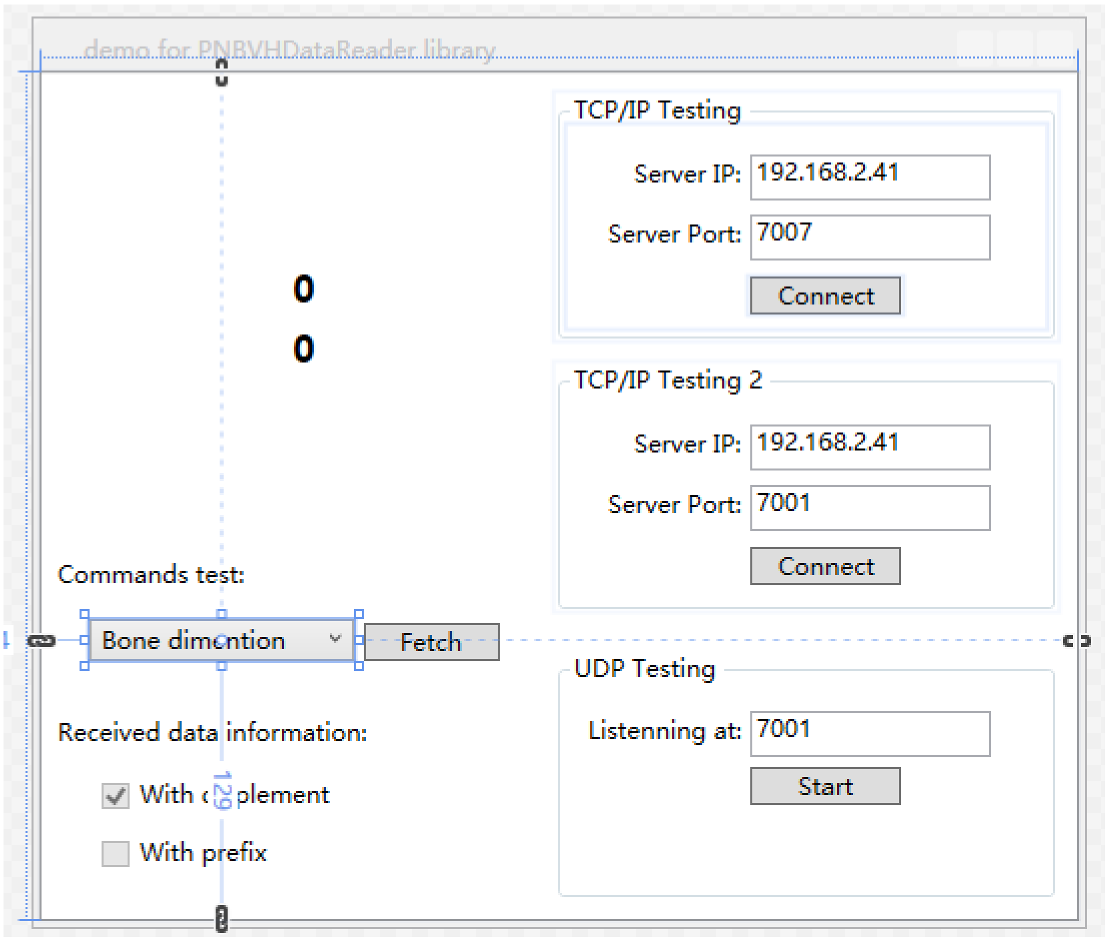
```
<Window x:Class="demo_vs2012_cs11.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid>

    </Grid>
</Window>
```

图. 3-5 WPF 初始项目界面

由于 NeuronDataReader.dll 中的 API 接口不能直接在 C#中被访问，因此，需要建立一个输出类的函数来包装函数库。

```
/* Copyright: Copyright 2014 Beijing Noitom Technology Ltd. All Rights reserved.
* Pending Patents: PCT/CN2014/085659 PCT/CN2014/071006
*
* Licensed under the Neuron SDK License Beta Version (the “License");
* You may only use the Neuron SDK when in compliance with the License,
* which is provided at the time of installation or download, or which
* otherwise accompanies this software in the form of either an electronic or a hard copy.
*
* Unless required by applicable law or agreed to in writing, the Neuron SDK
* distributed under the License is provided on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing conditions and
* limitations under the License.
*/


using System;
using System.Linq;
using System.Text;
using System.Runtime.InteropServices;  // For DllImport()


namespace NeuronDataReaderWraper
{
    #region Basic data types
    /// <summary>
    /// Socket connection status
    /// </summary>
    public enum SocketStatus
    {
        CS_Running,
        CS_Starting,
        CS_OffWork,
    };

    /// <summary>
    /// Data version
    /// </summary>
    public struct DataVersion
    {
        public byte BuildNumb;          // Build number
        public byte Revision;           // Revision number
        public byte Minor;              // Subversion number
        public byte Major;              // Major version number
    };

    /// <summary>
    /// Header format of BVH data
    /// </summary>
    [StructLayout(LayoutKind.Sequential, Pack=1)]
    public struct BvhDataHeader
    {
        public ushort HeaderToken1;    // Package start token: 0xDDFF
        public DataVersion DataVersion;// Version of community data format. e.g.: 1.0.0.2
        public UInt32 DataCount;       // Values count, 180 for without displacement data
        public UInt32 bWithDisp;       // With/out displacement
        public UInt32 bWithReference;  // With/out reference bone data at first
        public UInt32 AvatarIndex;     // Avatar index
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 32)]
        public string AvatarName;      // Avatar name
        public UInt32 Reserved1;       // Reserved, only enable this package has 64bytes
length
        public UInt32 Reserved2;       // Reserved, only enable this package has 64bytes
length
        public ushort HeaderToken2;    // Package end token: 0xEEFF
    };
    #endregion

    #region Command data types
    /// <summary>
    /// Command identities
    /// </summary>
    public enum CmdId
    {
        Cmd_BoneSize,                 // Id used to request bone size from server
        Cmd_AvatarName,               // Id used to request avatar name from server
        Cmd_FaceDirection,            // Id used to request face direction from server
        Cmd_DataFrequency,            // Id used to request data sampling frequency from
server
        Cmd_BvhInheritance,            // Id used to request bvh inheritance from server
        Cmd_AvatarCount,           // Id used to request avatar count from server
        Cmd_CombinationMode,          //
        Cmd_RegisterEvent,            //
        Cmd_SetAvatarName,            //
    };

    // Sensor binding combination mode
    public enum SensorCombinationModes
    {
        SC_ArmOnly,             // Left arm or right arm only
        SC_UpperBody,           // Upper body, include one arm or both arm, must have chest
node
        SC_FullBody,            // Full body mode
    };

    /// <summary>
    /// Header format of Command returned from server
    /// </summary>
    [StructLayout(LayoutKind.Sequential, Pack=1)]
    public struct CommandPack
    {
        public UInt16 Token1;                // Command start token: 0xAAFF
        public UInt32 DataVersion;           // Version of community data format. e.g.:
1.0.0.2
        public UInt32 DataLength;            // Package length of command data, by byte.
        public UInt32 DataCount;             // Count in data array, related to the specific
command.
        public CmdId CommandId;              // Identity of command.
        [MarshalAs(UnmanagedType.ByValArray, SizeConst = 40)]
        public byte[] CmdParaments;          // Command paraments
        public UInt32 Reserved1;             // Reserved, only enable this package has
32bytes length. Maybe used in the future.
        public UInt16 Token2;                // Package end token: 0xBBFF
    };


    /// <summary>
    /// Fetched bone size from server
    /// </summary>
    [StructLayout(LayoutKind.Sequential, Pack=1)]
    public struct CmdResponseBoneSize
    {
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = 60)]
        public string BoneName;       // Bone name
        public float BoneLength;      // Bone length
    };

    #endregion
    #region Callbacks for data output
    /// <summary>
    /// FrameDataReceived CALLBACK
    /// Remarks
    /// The related information of the data stream can be obtained from BvhDataHeader.
    /// </summary>
    /// <param name="customObject">User defined object.</param>
    /// <param name="sockRef">Connector reference of TCP/IP client as identity.</param>
    /// <param name="bvhDataHeader">A BvhDataHeader type pointer, to output the BVH data
format information.</param>
    /// <param name="data">Float type array pointer, to output binary data.</param>
    [UnmanagedFunctionPointer(CallingConvention.StdCall)]
    public delegate void FrameDataReceived(IntPtr customObject, IntPtr sockRef, IntPtr
bvhDataHeader, IntPtr data);

    /// <summary>
    /// Callback for command communication data with TCP/IP server
    /// </summary>
    /// <param name="customedObj">User defined object.</param>
    /// <param name="sockRef">Connector reference of TCP/IP client as identity.</param>
    /// <param name="cmdHeader">A CommandHeader type pointer contains command data
information.</param>
    /// <param name="cmdData">Data pointer of command, related to the specific
command.</param>
    /// <remark>The related information of the command data can be obtained from
CommandHeader. The data content is identified by its command id.</remark>
    [UnmanagedFunctionPointer(CallingConvention.StdCall)]
    public delegate void CommandDataReceived(IntPtr customedObj, IntPtr sockRef, IntPtr
cmdHeader, IntPtr cmdData);

    /// <summary>
    /// SocketStatusChanged CALLBACK
    /// Remarks
    ///   As convenient, use BRGetSocketStatus() to get status manually other than register
this callback
    /// </summary>
    /// <param name="customObject">User defined object.</param>
    /// <param name="sockRef">Socket reference of TCP or UDP service identity.</param>
    /// <param name="bvhDataHeader">Socket connection status</param>
    /// <param name="data">Socket status description.</param>
    [UnmanagedFunctionPointer(CallingConvention.StdCall)]
    public delegate void SocketStatusChanged(IntPtr customObject, IntPtr sockRef,
SocketStatus status, [MarshalAs(UnmanagedType.LPStr)]string msg);
```

```
    #endregion

    // API exportor
    public class NeuronDataReader
    {
        #region Importor definition
#if UNITY_IPHONE && !UNITY_EDITOR
         private const string ReaderImportor = "__Internal";
#elif _WINDOWS
         private const string ReaderImportor = "NeuronDataReader.dll";
#else
        private const string ReaderImportor = "NeuronDataReader";
#endif
        #endregion

        #region Functions API
        /// <summary>
        /// Register receiving and parsed frame data callback
        /// </summary>
        /// <param name="customedObj">Client defined object. Can be null</param>
        /// <param name="handle">Client defined function.</param>
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern void BRRegisterFrameDataCallback(IntPtr customedObj,
FrameDataReceived handle);

        /// <summary>
        ///
        /// </summary>
        /// <returns></returns>
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        //[return: MarshalAs(UnmanagedType.LPStr)]
         private static extern IntPtr BRGetLastErrorMessage();
         /// <summary>
         /// Call this function to get what error occurred in library.
         /// </summary>
         /// <returns></returns>
        public static string strBRGetLastErrorMessage()
        {
            // Get message pointer
            IntPtr ptr = BRGetLastErrorMessage();
            // Construct a string from the pointer.
             return Marshal.PtrToStringAnsi(ptr);
        }

        // Register TCP socket status callback
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern void BRRegisterSocketStatusCallback(IntPtr customedObj,
SocketStatusChanged handle);

        // Connect to server by TCP/IP
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern IntPtr BRConnectTo(string serverIP, int nPort);

        // Check TCP/UDP service status
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern SocketStatus BRGetSocketStatus(IntPtr sockRef);

        // Close a TCP/UDP service
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern void BRCloseSocket(IntPtr sockRef);

        // Start a UDP service to receive data at 'nPort'
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern IntPtr BRStartUDPServiceAt(int nPort);
        #endregion

        #region Commands API
        /// <summary>
        /// Register receiving and parsed Cmd data callback
        /// </summary>
        /// <param name="customedObj">Client defined object. Can be null</param>
        /// <param name="handle">Client defined function.</param>
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern void BRRegisterCommandDataCallback(IntPtr customedObj,
CommandDataReceived handle);

        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern bool BRRegisterAutoSyncParmeter(IntPtr sockRef, CmdId cmdId);
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern bool BRUnregisterAutoSyncParmeter(IntPtr sockRef, CmdId cmdId);

        // Check TCP connect status
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern bool BRCommandFetchAvatarDataFromServer(IntPtr sockRef, int
avatarIndex, CmdId cmdId);

        // Check TCP connect status
        [DllImport(ReaderImportor, CallingConvention = CallingConvention.Cdecl, CharSet =
CharSet.Ansi)]
        public static extern bool BRCommandFetchDataFromServer(IntPtr sockRef, CmdId cmdId);
        #endregion
    }
}
```
然后在主窗口中添加一些控制器，以连接至服务器并显示接收的 BVH 数据的基础信息：


demo for PNBVHDataReader library

0

0

TCP/IP Testing

Server IP: 192.168.2.41

Server Port: 7007

Connect

TCP/IP Testing 2

Server IP: 192.168.2.41

Server Port: 7001

Connect

Commands test:

Bone dimention

Fetch

UDP Testing

Listenning at: 7001

Start

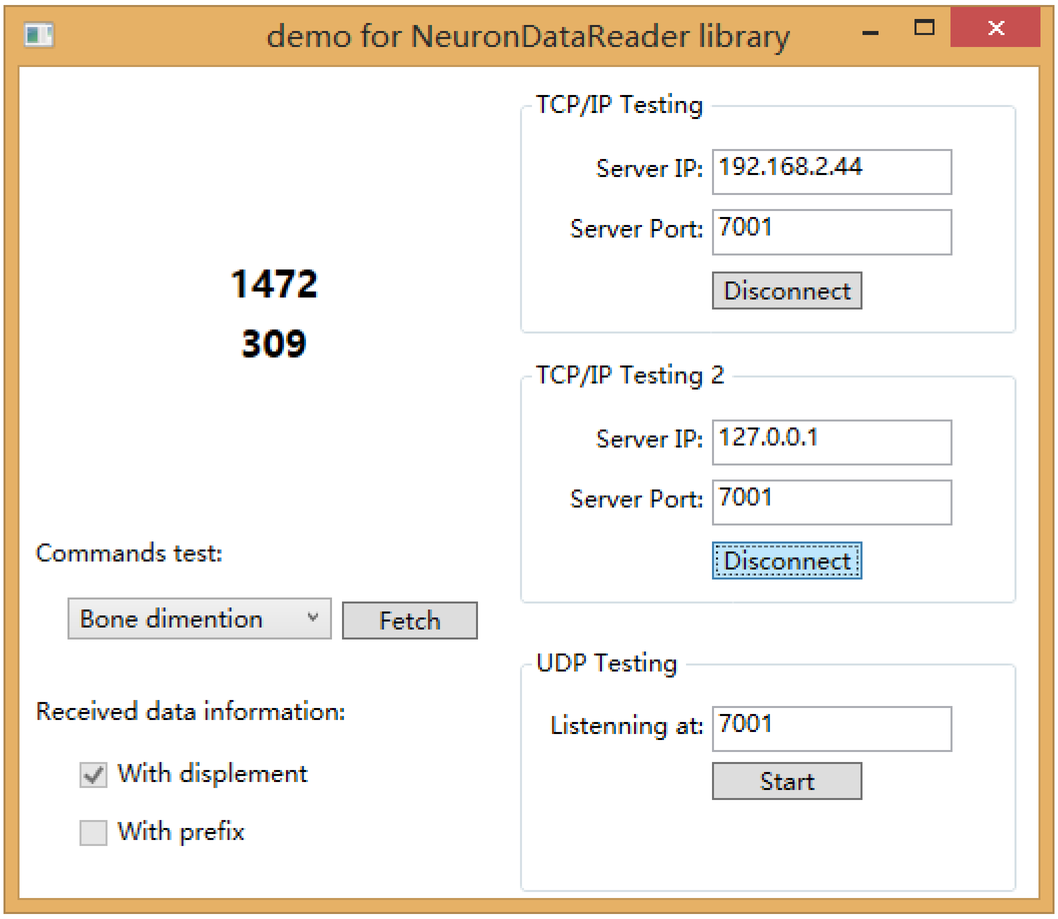
Received data information:

☑ With displement

☐ With prefix

图. 3-6 界面配置

将 NeuronDataReader.dll 拷贝至与“*.exe”文件相同的文件夹目录下，然后运行程序，结果如下所示：


demo for NeuronDataReader library

1472

309

Commands test:

Bone dimention
Fetch

Received data information:

- [✓] With displement
- [ ] With prefix

TCP/IP Testing

Server IP: 192.168.2.44

Server Port: 7001

Disconnect

TCP/IP Testing 2

Server IP: 127.0.0.1

Server Port: 7001

Disconnect

UDP Testing

Listenning at: 7001

Start

图. 3-7 运行结果

4. 问题反馈

如果您在使用中遇到了任何本文档中未提及的 bug 或者问题，请发邮件至 yufeng.tang@noitom.com 或者 yuanhui.he@noitom.com

谢谢！

Appendix A: 骨骼数据序列列表

	Bone Name	Sequence In Data Block
Body	Hips	0
	RightUpLeg	1
	RightLeg	2
	RightFoot	3
	LeftUpLeg	4
	LeftLeg	5
	LeftFoot	6
	Spine	7
	Spine1	8
	Spine2	9
	Spine3	10
	Neck	11
	Head	12
	RightShoulder	13
	RightArm	14
	RightForeArm	15
	RightHand	16
Fingers	RightHandThumb1	17
	RightHandThumb2	18
	RightHandThumb3	19
	RightInHandIndex	20
	RightHandIndex1	21
	RightHandIndex2	22
	RightHandIndex3	23
	RightInHandMiddle	24
	RightHandMiddle1	25
	RightHandMiddle2	26
	RightHandMiddle3	27
	RightInHandRing	28
	RightHandRing1	29
	RightHandRing2	30
	RightHandRing3	31
	RightInHandPinky	32
	RightHandPinky1	33
	RightHandPinky2	34
	RightHandPinky3	35
Body	LeftShoulder	36
	LeftArm	37
	LeftForeArm	38
	LeftHand	39
Fingers	LeftHandThumb1	40
	LeftHandThumb2	41
	LeftHandThumb3	42
	LeftInHandIndex	43
	LeftHandIndex1	44
	LeftHandIndex2	45
	LeftHandIndex3	46
	LeftInHandMiddle	47
	LeftHandMiddle1	48
	LeftHandMiddle2	49
	LeftHandMiddle3	50
	LeftInHandRing	51
	LeftHandRing1	52
	LeftHandRing2	53
	LeftHandRing3	54
	LeftInHandPinky	55
	LeftHandPinky1	56
	LeftHandPinky2	57
	LeftHandPinky3	58

Appendix B: BVH 数据头模板

```
HIERARCHY
ROOT Hips
{
    OFFSET 0.00 104.19 0.00
    CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
    JOINT RightUpLeg
    {
        OFFSET -11.50 0.00 0.00
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT RightLeg
        {
            OFFSET 0.00 -48.00 0.00
            CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
            JOINT RightFoot
            {
                OFFSET 0.00 -48.00 0.00
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                End Site
                {
                    OFFSET 0.00 -1.81 18.06
                }
            }
        }
    }
    JOINT LeftUpLeg
    {
        OFFSET 11.50 0.00 0.00
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT LeftLeg
        {
            OFFSET 0.00 -48.00 0.00
            CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
            JOINT LeftFoot
            {
                OFFSET 0.00 -48.00 0.00
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                End Site
                {
                    OFFSET 0.00 -1.81 18.06
                }
            }
        }
```

```
}
JOINT Spine
{
        OFFSET 0.00 13.88 0.00
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT Spine1
        {
                OFFSET 0.00 11.31 0.00
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                JOINT Spine2
                {
                        OFFSET 0.00 11.78 0.00
                        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                        JOINT Spine3
                        {
                                OFFSET 0.00 11.31 0.00
                                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                  JOINT Neck
                                {
                                        OFFSET 0.00 12.09 0.00
                                        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                        JOINT Head
                                        {
                                                OFFSET 0.00 9.00 0.00
                                                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                                End Site
                                                {
                                                        OFFSET 0.00 18.00 0.00
                                                }
                                        }
                                }
                                JOINT RightShoulder
                                  {
                                        OFFSET -3.50 8.06 0.00
                                        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                        JOINT RightArm
                                        {
                                                OFFSET -17.50 0.00 0.00
                                                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                                JOINT RightForeArm
                                                {
                                                        OFFSET -29.00 0.00 0.00
                                                        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                                        JOINT RightHand
```

```
{
  OFFSET -28.00 0.00 0.00
  CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
  JOINT RightHandThumb1
  {
      OFFSET -2.70 0.21 3.39
      CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
      JOINT RightHandThumb2
      {
          OFFSET -2.75 -0.64 2.83
          CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
          JOINT RightHandThumb3
          {
              OFFSET -2.13 -0.81 1.59
              CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                End Site
              {
                  OFFSET -1.80 -0.90 1.80
              }
          }
      }
  }
  JOINT RightInHandIndex
  {
      OFFSET -3.50 0.55 2.15
      CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
      JOINT RightHandIndex1
      {
          OFFSET -5.67 -0.10 1.09
          CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
          JOINT RightHandIndex2
          {
              OFFSET -3.92 -0.19 0.20
              CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
              JOINT RightHandIndex3
              {
                  OFFSET -2.22 -0.14 -0.08
                  CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                  End Site
                  {
                      OFFSET -2.28 0.00 0.00
                  }
              }
          }
```

```
        }
}
JOINT RightInHandMiddle
{
        OFFSET -3.67 0.56 0.82
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT RightHandMiddle1
        {
                OFFSET -5.62 -0.09 0.34
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                JOINT RightHandMiddle2
                {
                        OFFSET -4.27 -0.29 -0.20
                        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                        JOINT RightHandMiddle3
                        {
                                OFFSET -2.67 -0.21 -0.24
                                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                End Site
                                {
                                        OFFSET -2.28 0.00 0.00
                                }
                        }
                }
        }
}
JOINT RightInHandRing
{
        OFFSET -3.65 0.59 -0.14
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT RightHandRing1
        {
                OFFSET -5.00 -0.02 -0.52
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                JOINT RightHandRing2
                {
                        OFFSET -3.65 -0.29 -0.74
                        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                        JOINT RightHandRing3
                        {
                                OFFSET -2.55 -0.19 -0.44
                                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                                End Site
                                {
                                  OFFSET -2.16 0.00 0.00
                              }
                          }
                      }
                  }
              }
              JOINT RightInHandPinky
              {
                  OFFSET -3.43 0.51 -1.30
                  CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                  JOINT RightHandPinky1
                  {
                      OFFSET -4.49 -0.02 -1.18
                      CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                      JOINT RightHandPinky2
                      {
                          OFFSET -2.85 -0.16 -0.90
                          CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                          JOINT RightHandPinky3
                          {
                              OFFSET -1.77 -0.14 -0.66
                              CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                              End Site
                              {
                                  OFFSET -1.68 0.00 0.00
                              }
                          }
                      }
                  }
              }
            }
        }
    }
}
JOINT LeftShoulder
{
    OFFSET 3.50 8.06 0.00
    CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
    JOINT LeftArm
    {
        OFFSET 17.50 0.00 0.00
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT LeftForeArm
        {
```

```
OFFSET 29.00 0.00 0.00
CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
JOINT LeftHand
{
    OFFSET 28.00 0.00 0.00
    CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
    JOINT LeftHandThumb1
    {
        OFFSET 2.70 0.21 3.39
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT LeftHandThumb2
        {
            OFFSET 2.75 -0.64 2.83
            CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
            JOINT LeftHandThumb3
            {
                OFFSET 2.13 -0.81 1.59
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                End Site
                {
                    OFFSET 1.80 -0.90 1.80
                }
            }
        }
    }
    JOINT LeftInHandIndex
    {
        OFFSET 3.50 0.55 2.15
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT LeftHandIndex1
        {
            OFFSET 5.67 -0.10 1.08
            CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
            JOINT LeftHandIndex2
            {
                OFFSET 3.92 -0.19 0.20
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                JOINT LeftHandIndex3
                {
                    OFFSET 2.22 -0.14 -0.08
                    CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                    End Site
                    {
                        OFFSET 2.28 0.00 0.00
            }
        }
    }
}
JOINT LeftInHandMiddle
{
    OFFSET 3.67 0.56 0.82
    CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
    JOINT LeftHandMiddle1
    {
        OFFSET 5.62 -0.09 0.34
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT LeftHandMiddle2
        {
            OFFSET 4.27 -0.29 -0.20
            CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
            JOINT LeftHandMiddle3
            {
                OFFSET 2.67 -0.21 -0.24
                CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
                End Site
                {
                    OFFSET 2.28 0.00 0.00
                }
            }
        }
    }
}
JOINT LeftInHandRing
{
    OFFSET 3.65 0.59 -0.14
    CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
    JOINT LeftHandRing1
    {
        OFFSET 5.00 -0.02 -0.52
        CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
        JOINT LeftHandRing2
        {
            OFFSET 3.65 -0.29 -0.74
            CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
            JOINT LeftHandRing3
            {
                OFFSET 2.55 -0.19 -0.44
```

```
													CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
													End Site
													{
														OFFSET 2.16 0.00 0.00
													}
												}
											}
										}
									}
									JOINT LeftInHandPinky
									{
										OFFSET 3.43 0.51 -1.30
										CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
										JOINT LeftHandPinky1
										{
											OFFSET 4.49 -0.02 -1.18
											CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
											JOINT LeftHandPinky2
											{
												OFFSET 2.85 -0.16 -0.90
												CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
												JOINT LeftHandPinky3
												{
													OFFSET 1.77 -0.14 -0.66
													CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
													End Site
													{
														OFFSET 1.68 0.00 0.00
													}
												}
											}
										}
									}
								}
							}
						}
					}
				}
			}
		}
	}
}
MOTION
```

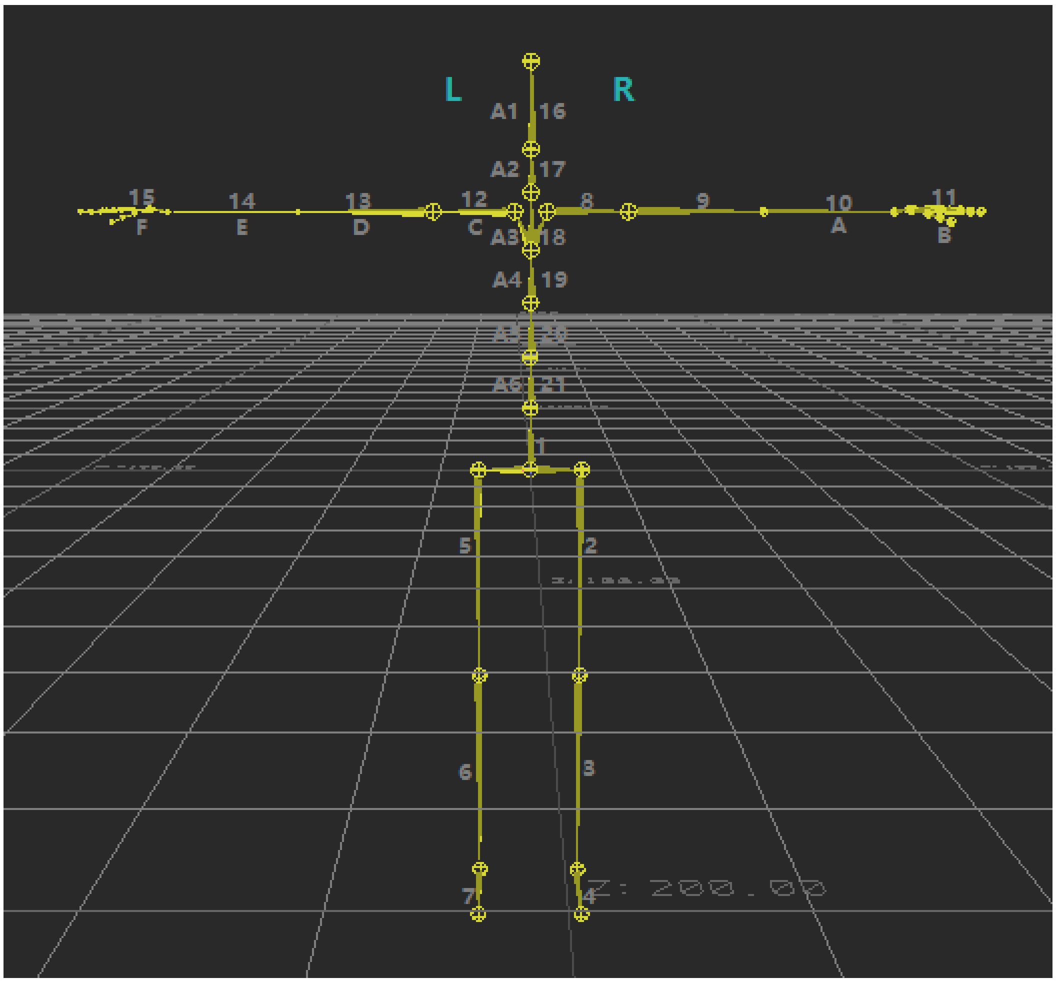
Frames: 0

Frame Time: 0.010

Appendix C: Bone 序列表

0. Hips
1. RightUpLeg
2. RightLeg
3. RightFoot
4. LeftUpLeg
5. LeftLeg
6. LeftFoot
7. RightShoulder
8. RightArm
9. RightForeArm
10. RightHand
11. LeftShoulder
12. LeftArm
13. LeftForeArm
14. LeftHand
15. Head
16. Neck
17. Spine3
18. Spine2
19. Spine1
20. Spine

Appendix D: 骨骼参考图



Revision history

Revision	Author	date	Description/changes
D1	Yuanhui He	12/22/2014	Initial released
D2	Peng Gao	12/22/2014	**Added:** Description of APIs. **Modified:** Format edit.
D3	Siyuan Deng	12/23/2014	**Added:** English translation. **Modified:**
D4	Yuanhui He	12/25/2014	**Added:** **Modified:** Delete string data stream type.
D5	Jinzhou Chen	12/25/2014	**Modified:** Modify the English translation.
D6	Yuanhui He	12/25/2014	**Modified:** Modify the English translation and some API description.
D7	Yuanhui He Siyuan Deng	1/26/2015	**Added:** Appendix, Skeleton Data format **Modified:** Data format description
D8	Yuanhui He	2/3/2015	**Added:** UDP protocol type support.
D9	Tobi	11/3/2015	**Modify:** English review.
D10	Yuanhui He	20/3/2015	**Add:** Multi-client is supported. **Modify:** English review.
D11	Yuanhui He	24/4/2015	**Add:** Some commands or APIs added to be used to sync parameters or data from server. **Delete:** Unity demo
D12	Yuanhui He	5/5/2015	**Modify:** Merged some TCP/UDP functions.
D13	Yufeng Tang	10/10/2015	**Add:** Details of skeleton data and data acquisition frequency description. C++ demo.

D14	Yufeng Tang	23/10/2015	**Add:** Details of calculation data and data acquisition frequency description. Appendix D: Bone index for calculation data.
D15	Yuanhui He	12/11/2015	**Add:** Appendix C: Bone Sequence Table
D16	Yufeng Tang	30/12/2015	**Add:** Chinese translation. **Modify:** Appendix A: Skeleton Data Sequence in Array. **Delete:** 2 command communication API.