

Neuron Data Reader API

指导手册

2020.08.28

NeuronDataReader b19
北京诺亦腾科技有限公司

Contents

1. 简介.....	3
1.1. NeuronDataReader 框架.....	3
1.2. 骨骼数据格式.....	3
1.3. 数据频率.....	4
1.4. 注意事项.....	4
2. 使用指导.....	5
2.1. 数据类型定义.....	5
2.1.1. Socket 连接状态.....	5
2.1.2. 数据流的版本.....	5
2.1.3. BVH 数据流的头.....	5
2.1.4. BVH 旋转顺序.....	5
2.2. 回调函数和注册回调.....	6
2.2.1. 回调骨骼数据.....	6
2.2.2. Socket 状态回调.....	6
2.3. API 使用说明.....	6
2.3.1. BRRegisterFrameDataCallback.....	6
2.3.2. BRRegisterSocketStatusCallback.....	6
2.3.3. BRConnectTo.....	7
2.3.4. BRStartUDPServiceAt.....	7
2.3.5. BRCloseSocket.....	7
2.3.6. BRGetSocketStatus.....	7
2.3.7. BRGetLastErrorMessage.....	7
Appendix A: 骨骼数据序列列表.....	8
Appendix B: BVH 数据头模板.....	9
Appendix C: Bone 序列列表.....	18

1. 简介

北京诺亦腾科技有限公司研发的 Axis Studio 软件可以通过 TCP/IP 或者 UDP 协议输出 BVH 动作数据。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

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（XYZ）和旋转（默认的旋转顺序为 YXZ）。

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 YXZ）。

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

$$\text{Displacement_X} = \text{bone index} * 6 + 0$$

$$\text{Displacement_Y} = \text{bone index} * 6 + 1$$

$$\text{Displacement_Z} = \text{bone index} * 6 + 2$$

$$\text{Rotation_Y} = \text{bone index} * 6 + 3$$

$$\text{Rotation_X} = \text{bone index} * 6 + 4$$

$$\text{Rotation_Z} = \text{bone index} * 6 + 5$$

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。

$$\text{Root_Displacement_X} = 0$$

$$\text{Root_Displacement_Y} = 1$$

$$\text{Root_Displacement_Z} = 2$$

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

3) BVH 坐标系

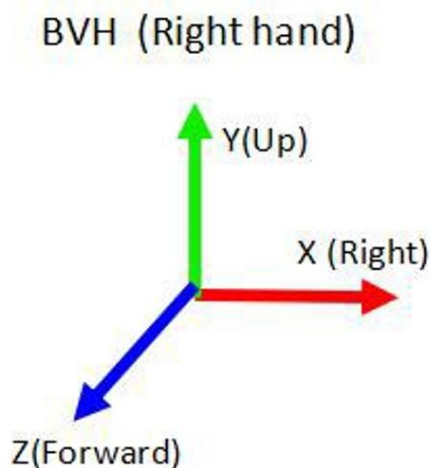


Fig. 1-2 BVH coordinate

1.3. 数据频率

从 NeuronDataReader 输出的数据的频率取决 Axis Studio 软件动捕数据的采集频率，目前支持 30Hz、60Hz、90Hz、120Hz 以及 240Hz，输出数据的频率与该数据应用的采集频率一致。

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。具体 Axis studio 中采集频率设置可参考 Axis Studio 使用手册。

1.4. 注意事项

NeuronDataReader 使用回调的方式来输出数据。因此，在连接到服务器之前，必须注册回调函数用来接收相应的数据。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. 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;
```

2.1.2. 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;
```

2.1.3. 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;
```

详细说明请参考 1.2.

2.1.4. BVH 旋转顺序

```
// BVH rotate orders
typedef enum _RotateOrders
{
    RO_XZY,
    RO_YXZ,
    RO_XYZ,
    RO_YZX,
    RO_ZXY,
    RO_ZYX,
```

```
RO_Unknown, // Unknown type
}RotateOrders;
```

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. 回调骨骼数据

```
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. 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 使用说明

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. BRRegisterSocketStatusCallback

注册 socket 状态回调句柄。

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

Parameters

customObj
User defined object.

handle
A function pointer.

Remarks

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

2.3.3. BRConnectTo

由给定的 IP 地址和端口连接至服务器。

```
// Connect to server
BDR_API SOCKET_REF BRConnectTo(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.

2.3.4. 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);
```

2.3.5. 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);
```

2.3.6. 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.

2.3.7. 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.

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 0.000 84.102 0.000
  CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
  JOINT RightUpLeg
  {
    OFFSET -9.500 -0.000 0.000
    CHANNELS 3 Yrotation Xrotation Zrotation
    JOINT RightLeg
    {
      OFFSET 0.000 -37.051 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation JOINT
      RightFoot
      {
        OFFSET 0.000 -37.051 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation
        End Site
        {
          OFFSET 0.000 -10.000 15.750
        }
      }
    }
  }
  JOINT LeftUpLeg
  {
    OFFSET 9.500 -0.000 0.000
    CHANNELS 3 Yrotation Xrotation Zrotation
    JOINT LeftLeg
    {
      OFFSET 0.000 -37.051 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation JOINT
      LeftFoot
      {
        OFFSET 0.000 -37.051 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation
        End Site
        {
          OFFSET 0.000 -10.000 15.750
        }
      }
    }
  }
}
```

```

}
JOINT Spine
{
  OFFSET 0.000 7.140 0.000
  CHANNELS 3 Yrotation Xrotation Zrotation
  JOINT Spine1
  {
    OFFSET 0.000 15.810 0.000
    CHANNELS 3 Yrotation Xrotation Zrotation JOINT
    Spine2
    {
      OFFSET 0.000 11.220 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation
      JOINT Neck
      {
        OFFSET 0.000 16.830 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation JOINT
        Neck1
        {
          OFFSET 0.000 4.500 0.000
          CHANNELS 3 Yrotation Xrotation Zrotation JOINT
          Head
          {
            OFFSET 0.000 4.500 0.000
            CHANNELS 3 Yrotation Xrotation Zrotation
            End Site
            {
              OFFSET 0.000 17.000 0.000
            }
          }
        }
      }
    }
  }
  JOINT RightShoulder
  {
    OFFSET -2.550 11.730 0.000
    CHANNELS 3 Yrotation Xrotation Zrotation JOINT
    RightArm
    {
      OFFSET -11.450 0.000 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation JOINT
      RightForeArm
      {
        OFFSET -25.000 0.000 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation

```

```

JOINT RightHand
{
  OFFSET -25.000 0.000 0.000
  CHANNELS 3 Yrotation Xrotation Zrotation JOINT
  RightHandThumb1
  {
    OFFSET -2.418 0.185 3.031
    CHANNELS 3 Yrotation Xrotation Zrotation
    JOINT RightHandThumb2
    {
      OFFSET -3.578 0.000 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation JOINT
      RightHandThumb3
      {
        OFFSET -2.485 0.000 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation
        End Site
        {
          OFFSET -2.131 0.000 0.000
        }
      }
    }
  }
}
JOINT RightInHandIndex
{
  OFFSET -3.132 0.494 1.922
  CHANNELS 3 Yrotation Xrotation Zrotation
  JOINT RightHandIndex1
  {
    OFFSET -5.068 -0.089 0.971
    CHANNELS 3 Yrotation Xrotation Zrotation JOINT
    RightHandIndex2
    {
      OFFSET -3.516 0.000 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation
      JOINT RightHandIndex3
      {
        OFFSET -1.993 0.000 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation End
        Site
        {
          OFFSET -1.754 0.000 0.000
        }
      }
    }
  }
}

```

```

    }
  }
}
JOINT RightInHandMiddle
{
  OFFSET -3.285 0.502 0.735
  CHANNELS 3 Yrotation Xrotation Zrotation
  JOINT RightHandMiddle1
  {
    OFFSET -5.026 -0.082 0.305
    CHANNELS 3 Yrotation Xrotation Zrotation JOINT
    RightHandMiddle2
    {
      OFFSET -3.837 0.000 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation
      JOINT RightHandMiddle3
      {
        OFFSET -2.405 0.000 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation End
        Site
        {
          OFFSET -1.918 0.000 0.000
        }
      }
    }
  }
}
JOINT RightInHandRing
{
  OFFSET -3.269 0.523 -0.125
  CHANNELS 3 Yrotation Xrotation Zrotation
  JOINT RightHandRing1
  {
    OFFSET -4.502 -0.021 -0.465
    CHANNELS 3 Yrotation Xrotation Zrotation JOINT
    RightHandRing2
    {
      OFFSET -3.344 0.000 0.000
      CHANNELS 3 Yrotation Xrotation Zrotation
      JOINT RightHandRing3
      {
        OFFSET -2.320 0.000 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation End
        Site

```



```

{
  OFFSET 25.000 0.000 0.000
  CHANNELS 3 Yrotation Xrotation Zrotation
  JOINT LeftHand
  {
    OFFSET 25.000 0.000 0.000
    CHANNELS 3 Yrotation Xrotation Zrotation JOINT
    LeftHandThumb1
    {
      OFFSET 2.418 0.185 3.031
      CHANNELS 3 Yrotation Xrotation Zrotation
      JOINT LeftHandThumb2
      {
        OFFSET 3.578 0.000 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation JOINT
        LeftHandThumb3
        {
          OFFSET 2.485 0.000 0.000
          CHANNELS 3 Yrotation Xrotation Zrotation
          End Site
          {
            OFFSET 2.131 0.000 0.000
          }
        }
      }
    }
  }
  JOINT LeftInHandIndex
  {
    OFFSET 3.132 0.494 1.922
    CHANNELS 3 Yrotation Xrotation Zrotation
    JOINT LeftHandIndex1
    {
      OFFSET 5.068 -0.089 0.971
      CHANNELS 3 Yrotation Xrotation Zrotation JOINT
      LeftHandIndex2
      {
        OFFSET 3.516 0.000 0.000
        CHANNELS 3 Yrotation Xrotation Zrotation
        JOINT LeftHandIndex3
        {
          OFFSET 1.993 0.000 0.000
          CHANNELS 3 Yrotation Xrotation Zrotation End
          Site
          {

```

```

        OFFSET 1.754 0.000 0.000
    }
}
}
}
}
JOINT LeftInHandMiddle
{
    OFFSET 3.285 0.502 0.735
    CHANNELS 3 Yrotation Xrotation Zrotation
    JOINT LeftHandMiddle1
    {
        OFFSET 5.026 -0.082 0.305
        CHANNELS 3 Yrotation Xrotation Zrotation JOINT
        LeftHandMiddle2
        {
            OFFSET 3.837 0.000 0.000
            CHANNELS 3 Yrotation Xrotation Zrotation
            JOINT LeftHandMiddle3
            {
                OFFSET 2.405 0.000 0.000
                CHANNELS 3 Yrotation Xrotation Zrotation End
                Site
                {
                    OFFSET 1.918 0.000 0.000
                }
            }
        }
    }
}
JOINT LeftInHandRing
{
    OFFSET 3.269 0.523 -0.125
    CHANNELS 3 Yrotation Xrotation Zrotation
    JOINT LeftHandRing1
    {
        OFFSET 4.502 -0.021 -0.465
        CHANNELS 3 Yrotation Xrotation Zrotation JOINT
        LeftHandRing2
        {
            OFFSET 3.344 0.000 0.000
            CHANNELS 3 Yrotation Xrotation Zrotation
            JOINT LeftHandRing3
            {

```

```

        OFFSET 2.320 0.000 0.000

        CHANNELS 3 Yrotation Xrotation

        Zrotation End Site

        {

            OFFSET 1.804 0.000 0.000

        }

    }

}

JOINT LeftInHandPinky

{

    OFFSET 3.071 0.456 -1.167

    CHANNELS 3 Yrotation Xrotation

    Zrotation JOINT LeftHandPinky1

    {

        OFFSET 4.023 -0.021 -1.059

        CHANNELS 3 Yrotation Xrotation Zrotation

        JOINT LeftHandPinky2

        {

            OFFSET 2.678 0.000 0.000

            CHANNELS 3 Yrotation Xrotation

            Zrotation JOINT LeftHandPinky3

            {

                OFFSET 1.692 0.000 0.000

                CHANNELS 3 Yrotation Xrotation

                Zrotation End Site

                {

                    OFFSET 1.598 0.000 0.000

                }

            }

        }

    }

}

```



```
    }  
  }  
}  
}
```

MOTION

Frames: 0

Frame Time: 0.010000

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. Neck1↵
17. Neck↵
18. Spine2↵
19. Spine1↵
20. Spine↵
21. RightHandThumb1↵
22. RightHandThumb2↵
23. RightHandThumb3↵
24. RightInHandIndex↵
25. RightHandIndex1↵
26. RightHandIndex2↵
27. RightHandIndex3↵
28. RightInHandMiddle↵
29. RightHandMiddle1↵
30. RightHandMiddle2↵
31. RightHandMiddle3↵
32. RightInHandRing↵
33. RightHandRing1↵
34. RightHandRing2↵
35. RightHandRing3↵

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. Neck1↵
17. Neck↵
18. Spine2↵
19. Spine1↵
20. Spine↵
21. RightHandThumb1↵
22. RightHandThumb2↵
23. RightHandThumb3↵
24. RightInHandIndex↵
25. RightHandIndex1↵
26. RightHandIndex2↵
27. RightHandIndex3↵
28. RightInHandMiddle↵
29. RightHandMiddle1↵
30. RightHandMiddle2↵
31. RightHandMiddle3↵
32. RightInHandRing↵
33. RightHandRing1↵
34. RightHandRing2↵
35. RightHandRing3↵