MOONSEER VR 智能体感服 V4.0

C/C++ 驱动开发指南

(适配 C#/Unity3D 和 Android)

--- 2022-10-12 ---

Revision Record 文档修订记录

Date 日期	Revision Version 修订版本	Change Description 修改描述	Author 作者
V4.01.a	2022-10-12	初稿发布	MOONSEEER VR

目录

1、简介

该 SDK 采用 C-Style 风格的 C++语言实现， MOONSEER VR 智能体感服 V4.0）（蓝牙版）驱动 SDK（如果没有特别说明，以下均简称：SDK），SDK 我们将以 DLL 的形式提供，开发者可以通过 MOONSEER VR 开发者中心进行最新版的 SDK 下载。

2、SDK 说明

2.1 支持版本

本 SDK 仅支持用于开发 MOONSEER VR 智能体感服 4.0 版本驱动

如果您是接入 Unreal Engine，我们推荐您下载 C++版本的 SDK 驱动

2.2 功能清单

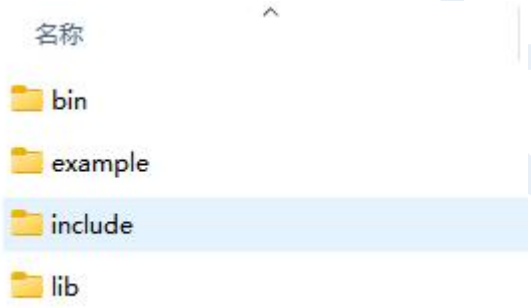
使用该 SDK，第三方开发者能够获得以下 MOONSEER VR 智能体感服的开放能力：

- 智能体感服模块和腿部模块的电量变化通知
- 获取智能体感服和腿部模块的固件信息
- 智能体感服和腿部模块的连接状态及状态变化通知
- 智能体感服和腿部模块的充电情况和充电/未充电状态变化通知
- 智能体感服和腿部模块的标定和标定完成通知
- 智能体感服左侧/右侧胸部按钮按下和弹起通知（支持 Level Trigger 和 Edge Trigger）

- 智能体感服左侧/右侧红外发射模块遮挡和移除遮挡通知（支持 Level Trigger 和 Edge Trigger）
- 智能体感服前后移动触发 （支持 AWSD 模式和线性移动模式）
- 智能体感服左右移动触发（支持 AWSD 模式和线性移动模式）
- 腿部模块前后移动触发（仅支持线性移动模式）
- 自定义标定精度
- 支持 12 路智能体感服力反馈设置，支持多路同时力反馈和多路不同时长反馈

2.2 SDK 目录结构

MOONSEER VR 智能体感服驱动 SDK 的目录结构如下图所示：



SDK 仅有一个 ArmorDeviceDriver.h 头文件，放在 include 文件夹中，如果在 C#/Unity3D/Android 系统中使用，建议采用 DLL 动态加载，其中 bin 和 lib 分别存放 dll 文件和 lib 文件。

- example 目录是智能体感服 Demo，目前提供一个基于 Visual Studio 2019 C#的简单工程，该工程演示了如何在 C#中使用该驱动（Unity3D 类似）

2.3 SDK 使用说明

SDK 根据不同的语言/引擎环境，使用不同的加载方式，如果您是使用 Unreal Engine，建议下载 C++版 SDK。

2.3.1、在 Win32 中直接使用

(未完待续!)

2.3.2、在 C# / Unity3D 引擎中使用

参考 example 中 AromrSDK_Demo(cs) 例子

2.3.3、在 Android 中使用

(未完待续!)

3、SDK 指南

3.1、主要使用步骤

SDK 启动主要使用函数 HS_InitArmor 和 HS_InitLeg 初始化设备驱动，然后通过 HS_Set 系列函数设置事件响应回调函数，所有需要监听的事件回调设置完成之后，调用 HS_OpenDeviceDriver 运行。

程序结束后调用 HS_ReleaseDeviceDriver 释放资源。

所有的回调注册均需要在 HS_OpenDeviceDriver 之前设置

3.2、函数声明说明

1、void HS_Version();

说明：获取版本号,可用于 DLL 测试，在控制台程序中成功调用该方法会在当前控制台输出版本信息

2、int HS_InitArmor();

说明：初始化盔甲，必须调用才能使用盔甲相关的函数

返回值：无错误返回 0，设备初始化失败返回 1

3、int HS_InitLeg();

说明：初始化腿部模块，必须调用才能使用腿部相关的函数

返回值：无错误返回 0，设备初始化失败返回 1

4、int HS_EnablePassThroughMode(char deviceIdentifier[32], char IP[128], int Port, int RepeatNumber, int protocol);

参数：char deviceIdentifier[32] 设备标识符,用于标识透传客户端的唯一 ID

char IP[128] 透传服务器 IP 地址

Int Port 透传服务器端口

int RepeatNumber 重试次数

int Protocol 透传协议，目前暂只支持 UDP

说明：开启 UDP 透传,开启 UDP 透传之后，数据将使用 UDP 方式传输，透传客户端例子参考 example 中 PassThrough_Demo（客户端使用 Cocos Creator,透传服务器使用 Go），该 SDK 默认采用 Native（驱动）模式。

5、void HS_OpenDeviceDriver(int deviceIdentifier = 0, void (*fpResult)(bool bResult, char* errMsg) = 0);

参数： int deviceIdentifier 支持的设备描述符

取值： 1：盔甲，2：左腿模块，4：右腿模块

说明：开启设备

6、void HS_ReleaseDeviceDriver(void(*fpResult)());

说明：关闭设备并释放资源

7、int HS_SetArmorDeviceMoveMode(int mode);

参数 int mode 设置盔甲移动模式

取值：1、虚拟键盘模式

2、线性位移模式

说明：用于设置当前盔甲的回调反馈模式，如果是虚拟键盘模式，则盔甲的移动直接映射成键盘的事件（VK），如果是线性位移，则出发位移回调函数。

8、int HS_SetLegDeviceMoveMode(int deviceIdentifier , int mode);

参数 int deviceIdentifier 设备标识符

取值：2、左腿设备

4、右腿设备

int mode 设置盔甲移动模式

取值：1、虚拟键盘模式

2、线性位移模式

说明：用于设置当前腿部的回调反馈模式，如果是虚拟键盘模式，则盔甲的移动直接映射成键盘的事件（VK），如果是线性位移，则出发位移回调函数。

9、int HS_SetSamplingPrecision(int nSamplingPrecision);

参数： nSamplingPrecision 采样精度，如果精度越高，标定采集数据越长

说明：设置系统标定精度

10、int HS_SetArmorTriggerMode(int TriggerMode);

参数：int TriggerMode 触发模式

取值：1、水平触发

2、边沿触发

说明：设置盔甲的触发模式， 盔甲支持水平触发和上升沿触发（边沿触发）模式，默认为边沿触发。

11、int HS_GetArmorStatus();

说明：返回盔甲的状态，1 是正常，0 是掉线

12、int HS_SetArmorPowerChangedResult(void (*lpFuncPowerChangedResult) (b8 power), b8 changedMode = 1);

说明：设置电量改变回调函数，函数接受 2 个参数，除回调函数外，还需要传入 b8 类型的通知信号模式，0 是每帧上报，1 是改变上报， 回调函数包括 1 个参数，其中参数为 b8(#define unsigned char b8) 类型的电量值

13、int HS_ArmorCalibration();

说明：标定，该函数调用后，系统进入标定模式，标定采用去掉零点偏移量和标度因素进

行标定，标定时，设备需要处于静置状态，并持续一定时间，标定时长根据标定精度而定，如果标定精度越高，耗时越长。

标定成功后， 将根据当前设备的位置状态值做零偏值。该函数可以在任何地方调用

14、int HS_SetArmorCalibrationComplete(void (*lpFuncArmorCalibrationResult)());

说明：盔甲标定成功后回调函数

15、int HS_SetArmorForwardOrBackMove(void
(*lpArmorForwardOrBackMove)(double sine, double angle));

说明：设置盔甲前后移动的回调函数，当模式为线性移动时生效，回调函数接受两个 double 类型的参数，sine 为当前移动位移量的正弦值，angle 为位移量值，其中，向前为负数，向后为正数，0 为静止状态。

16、int HS_SetArmorLeftOrRightMove(void (*lpArmorLeftOrRightMove)
(double sine, double angle));

说明：设置盔甲左右移动的回调函数，当模式为线性移动时生效，回调函数接受两个 double 类型的参数，sine 为当前移动位移量的正弦值，angle 为位移量值，其中，向左为负数，向右为正数，0 为静止状态。

17、int HS_SetArmorLeftButtonStateChanged(void
(*lpArmorLeftButtonStateChanged)(bool state));

说明：设置盔甲左侧胸甲点击事件回调，如果设置为水平触发，则按下后持续触发，如果设置为上升沿触发，则按下触发一次，弹起触发一次，回调函数参数 state 为 true 时为按下，false 为弹起

18、int HS_SetArmorRightButtonStateChanged(void
(*lpArmorRightButtonStateChanged)(bool state));

说明：设置盔甲右侧胸甲点击事件回调，如果设置为水平触发，则按下后持续触发，如果设置为上升沿触发，则按下触发一次，弹起触发一次，回调函数参数 state 为 true 时为按下，false 为弹起

19、Int HS_SetArmorLeftInfraredStateChanged(void
(*lpArmorLeftInfraredStateChanged) (bool state));

说明：设置盔甲左侧红外事件触发，如果设置为水平触发，则进入红外探测区域后持续触发，离开后停止触发，如果设置为上升沿触发，则进入红外探测区域触发一次，离开触发一次，回调函数参数 state 为 true 时为进入探测区域，false 为离开探测区域

20、int HS_SetArmorRightInfraredStateChanged(void
(*lpArmorRightInfraredStateChanged)(bool state));

说明：设置盔甲右侧红外事件触发，如果设置为水平触发，则进入红外探测区域后持续触发，离开后停止触发，如果设置为上升沿触发，则进入红外探测区域触发一次，离开触发一次，回调函数参数 state 为 true 时为进入探测区域，false 为离开探测区域

21、int HS_SetArmorVoltageStateChanged(void (*lpArmorVoltageStateChanged)
(int state));

说明：设置盔甲电压改变回调函数，回调函数接受一个 int 参数 state，0 为低电压，1 为正常工作电压

22、int HS_SetArmorRechargingStateChanged(void
(*lpArmorRechargingStateChanged)(int state));

说明：设置盔甲充电状态改变回调函数，回调函数接受一个 int 参数 state，1 为正常充电，2 为未充电

23、void HS_BindMoveVirtualKey(int deviceIdentifier,b8 w, b8 a, b8 s, b8 d);

说明：绑定移动虚拟键位，如果设置的是虚拟键位移动模式，则这里可以设置四个移动方向的键盘映射值，默认是对应的 WASD 四个键，分别对应 Ascii 为：65,87,83,68

24、bool HS_ArmorDeviceShake(unsigned short MonitorIndexs, b8 t1, b8 t2, b8 t3, b8 t4, b8 t5, b8 t6, b8 t7, b8 t8, b8 t9 ,b8 t10, b8 t11, b8 t12, b8 byteScale = 100);

说明：盔甲震动函数,该函数可以在任何地方调用。

参数: MonitorIndex 电机索引，取值如下：

* 1 号马达: 1
* 2 号马达: 2
* 3 号马达: 4
* 4 号马达: 8
* 5 号马达: 16
* 6 号马达: 32
* 7 号马达: 64
* 8 号马达: 128
* 9 号马达: 256
* 10 号马达: 512
* 11 号马达: 1024
* 12 号马达: 2048

参数：t1-t12 对应 1-12 个电机不同的震动时间，如果 MonitorIndex 未设置当前电机序号，则对应的时间将被忽略。

参数 byteScale 是时间基数，默认为 100

说明： 目前系统精度设置为 100ms，当倍数为 1 的时候,震动时间为

T（max） = 100ms * byteScale(1) * t(F) t ∈ { N | [0,F] }

T（min） = 100ms * byteScale(1) * t(1)

最大震动时间为 1500ms,最小震动时间为 100ms

注：函数调用成功返回 true，失败返回 false，该函数返回值 true 和 false 仅代表指令是否下发成功。如果本次指令下发成功但下位机未被执行，则会自动尝试重发指令直到设备成功执行位置。

25、int HS_LegCalibration();

说明：腿部标定，该函数调用后，系统进入标定模式，标定采用去掉零点偏移量和标度因素进行标定，标定时，设备需要处于静置状态，并持续一定时间，标定时长根据标定精度而定，如果标定精度越高，耗时越长。

标定成功后， 将根据当前设备的位置状态值做零偏值。该函数可以在任何地方调用

26、int HS_SetLegCalibrationComplete(void (*lpFuncLegCalibrationResult)(),
int deviceIdentifier);

说明：腿部模块标定成功后回调函数，回调函数由 int 类型的 deviceIdentifier 确定是左腿模块还是右腿模块，其中，2 是左腿模块，4 是右腿模块

27、int HS_SetLegPowerChangedResult(void (*lpFuncLegPowerChangedResult)
(b8 power), int identifier, b8 changedMode = 1);

说明：设置电量改变回调函数，函数包括 3 个参数，除回调函数外，还需要传入 int 类型的 identifer 标识设备，其中，2 为左腿模块，4 为右腿模块，b8 类型的通知信号模式，0 是每

帧上报，1 是改变上报，回调函数包括 1 个参数，其中参数为 b8(#define unsigned char b8) 类型的电量值。

28、int HS_SetLegVoltageStateChanged(void (*lpLegVoltageStateChanged)
(int state), int identifier);

说明：设置盔甲电压改变回调函数，回调函数接受一个 int 参数 state，0 为低电压，1 为正常工作电压。 用 int identifer 标识设备，取值为 2 是左腿模块，4 是右腿模块

29、int HS_SetLegRechargingStateChanged(void (*lpLegRechargingStateChanged)
(int state), int identifier);

说明：设置盔甲充电状态改变回调函数，回调函数接受一个 int 参数 state，1 为正常充电，2 为未充电。用 int identifer 标识设备，取值为 2 是左腿模块，4 是右腿模块

30、int HS_SetLegForwardOrBackMove(void (*lpLegForwardOrBackMove)
(double sine, double angle), int deviceIdentifier);

说明：设置腿部模块前后移动的回调函数，回调函数接受两个 double 类型的参数，sine 为当前移动位移量的正弦值，angle 为位移量值，其中，向前为负数，向后为正数，0 为静止状态。

注：腿部模块仅支持线性模式

31、int HS_SetLegLeftOrRightMove(void (*lpLegLeftOrRightMove)
(double sine, double angle), int deviceIdentifier);
}

说明：设置腿部模块左右移动的回调函数，回调函数接受两个 double 类型的参数，sine 为当前移动位移量的正弦值，angle 为位移量值，其中，向左为负数，向右为正数，0 为静止状态。

注：腿部模块仅支持线性模式

4、SDK 故障与售后

4.1、HS_InitArmor 和 HS_InitLeg 返回 0

如果这两个函数返回 0，说明系统无法找到设备，排查包括以下几点但不局限于：

- 1、电池电量是否充足 （可先充一会电再试）
- 2、信号接收器是否正常工作
- 3、势能设备电源开关是否打开

4.2、其他故障

如果遇到其他更多故障或 SDK 接口问题，您可以使用以下方法寻求技术支持：

- 1、Moonseer 开发者中心工单系统
  https://developer.moonseer.com
- 2、Moonseer 售后客服专线

  联系方式：

5、SDK 版本及更新记录

5.1、SDK 版本说明

通过 HS_Version() 获取当前版本号，您还可以在开发者中心获取最新版本 SDK

5.2、SDK 更新记录

时间	版本	更新记录
2021/11/10	4.01.0000	首次发布 Moonseer 全新势能设备 SDK
2022/10/14	4.01.0000	SDK 驱动提供 C-Style 风格用于支持 unity3d
2022/10/20	4.02.alpha	加入透传模式