

1.1.1 PicoNeo2 Native 版 手柄服务 接口

libCVControllerClient.pxr.so

So 中有以下是对应的接口：

API	Description
<code>float* getPicoControllerSensorState(int controllerSerialNum);</code>	<p>功能： 获取手柄当前 6Dof 数据</p> <p>参数： controllerSerialNum 手柄 ID ,0 左手, 1 右手</p> <p>返回值： 手柄 6Dof 数据，包括 rotation 和 position。</p> <p>rotation 包括: ,qx,qy,qz,q0</p> <p>position 包括: x,y,z</p>
<code>int getPicoControllerConnectionState(int controllerSerialNum);</code>	<p>功能： 获取手柄连接状态</p> <p>参数: controllerSerialNum 手柄 ID 0/1</p> <p>返回值: 0 代表未连接, 1 代表已连接</p>
<code>float* getPicoControllerAngularVelocity(int controllerSerialNum);</code>	<p>功能： 获取手柄的角速度</p> <p>参数: controllerSerialNum 手柄 ID 0/1</p> <p>返回值: 角速度, 单位 rad/s</p>
<code>float* getPicoControllerAcceleration(int controllerSerialNum);</code>	<p>功能: 获取手柄的加速度</p> <p>参数: controllerSerialNum 手柄 ID 0/1</p> <p>返回值: 加速度, 单位 mm/s^2</p>
<code>void setPicoMainControllerSerialNum(int controllerSerialNum);</code>	<p>功能: 设置当前的主控手柄</p> <p>参数: controllerSerialNum 手柄 ID 0/1</p> <p>返回值: 无</p>

int getPicoMainControllerSerialNum();	<p>功能：获取当前的主控手柄</p> <p>参数：无</p> <p>返回值：0 代表左手手柄，1 代表右手手柄</p>
int GetPicoControllerDofAbilityState(int controller);	<p>功能：获取当前手柄支持 3Dof 还是 6Dof</p> <p>参数：controller 手柄 ID 0/1</p> <p>返回值：-1 为调用失败，0, 2 为 6Dof, 1 为 3Dof</p>
int getPicoControllerHandness();	<p>功能：获取当前的惯用手柄</p> <p>参数：无</p> <p>返回值：0 代表左手手柄，1 代表右手手柄</p>
void startPicoCVControllerEvent();	<p>功能：开启手柄按键回调线程</p> <p>参数：无</p> <p>返回值：无</p>
void stopPicoCVControllerEvent();	<p>功能：关闭手柄按键回调线程</p> <p>参数：无</p> <p>返回值：无</p>
void registerPicoCVControllerEventVRScene(pVRSceneType* type);	<p>功能：注册手柄按键回调</p> <p>参数：type 按键回调函数集合</p> <p>返回值：无</p>
void setIsPicoEnableHomeKey(bool isEnabled);	<p>功能：是否打开手柄 home 键监听</p> <p>参数：isEnabled true/false</p> <p>返回值：无</p>
float* getPicoHeadSensorData();	<p>功能：获取头戴 6Dof 数据</p> <p>参数：无</p> <p>返回值：头戴 6Dof 数据</p>

<pre>float* getPicoControllerSensorDataPredict(int controllerSerialNum, float headSensorData[], float predictTime);</pre>	<p>功能：预测手柄 predictTime 时间后的 6Dof 数据</p> <p>参数：controllerSerialNum 手柄 ID headSensorData 头戴 6Dof 数据 predictTime 预测时间</p> <p>返回值：预测 手柄 predictTime 时间后的 6Dof 数据 包括 rotation 和 position。</p> <p>rotation 包括：qx,qy,qz,q0 position 包括：x,y,z</p>
<pre>void vibratePicoCV2ControllerStrength(float strength ,int time,int controllerSerialNum);</pre>	<p>功能：手柄震动设置</p> <p>参数：strength 手柄震动强度 (0-255) time 手柄震动时间 (单位：ms) controllerSerialNum 手柄 ID</p> <p>返回值：NULL</p>
<pre>float* getPicoControllerLinearVelocity(int controllerSerialNum)</pre>	<p>功能：获取手柄线速度</p> <p>参数：controllerSerialNum 手柄 ID</p> <p>返回值：手柄线速度(单位：mm/s)</p>
<pre>float* getPicoControllerSensorData(int controllerSerialNum, float headSensorData[]);</pre>	<p>功能：获取手柄根据传入头数据计算后的 6Dof 数据</p> <p>参数：controllerSerialNum 手柄 ID headSensorData 头戴 6Dof 数据 (position 单位为米)</p> <p>返回值：根据传入头数据计算后的 6Dof 数据 包括 rotation 和 position。</p> <p>rotation 包括：qx,qy,qz,q0 position 包括：x,y,z</p>
<pre>void startPicoCVControllerThread(int headSensorState, int handSensorState)</pre>	<p>功能：以 3DOF 或者 6DOF 重启手柄服务线程</p> <p>参数：headSensorState 头戴 sensor 类型 (0/1) 参数：handSensorState 手柄 sensor 类型 (0/1)</p> <p>参数说明：0 代表 3DOF, 1 代表 6DOF</p>

	返回值: 无
float* getPicoControllerFixedSensorState(int controllerSerialNum)	<p>功能: 获取手柄固定坐标系数据</p> <p>参数: controllerSerialNum 手柄 ID 包括 rotation 和 position。</p> <p>rotation 包括: qx,qy,qz,q0 position 包括: x,y,z</p> <p>返回值: 手柄固定坐标系数据</p>
void setPicoUnityVersion(const char* version);	<p>功能: 手柄兼容性设置</p> <p>参数: version 2.9.0.0-->新应用 2.7.0.0-->旧应用</p> <p>返回值: 无</p>
int* getPicoCVControllerKeyEvent(int controllerSerialNum);	<p>功能: 获取手柄按键值</p> <p>参数: 0 左手柄 1 右手柄</p> <p>返回值: 长度为 8 的数组</p> <p>0: touchpad x 轴数据 1: touchpad y 轴数据 2: home 键值 0 抬起 1 按下 3: 返回键 4: touchpad 键 7: trigger 键值 0-255 其他按键无意义</p>
int* getPicoCV2ControllerKeyEvent(int controllerSerialNum);	<p>功能: 获取 cv2 手柄专用键值</p> <p>参数: 0 左手柄 1 右手柄</p> <p>返回值: 长度为 4 的数组</p> <p>0: A 或者 x 键 1: B 或者 Y 键 2: 侧键</p>