

# WISEGLOVE19E/19E+数据手套 (USB 版本) 用户手册



用户手册 Ver1.8.2 Beta

北京鑫天视景科技有限公司

2024 年 6 月

# WISEGLOVE (USB) USER'S MANUAL

## 目 录

1.	产品介绍 .....	3
2.	产品安装 .....	4
2.1	包装明细 .....	4
2.2	数据手套的组件 .....	5
2.3	软件明细及安装方法 .....	5
3	产品使用 .....	6
3.1	使用前的准备工作 .....	6
3.2	穿戴需要注意的问题 .....	8
3.3	运行范例程序 .....	8
3.4	手套的手指标定方法 .....	10
3.5	手套的手臂标定方法 .....	10
3.6	手套的握力传感器标定方法 .....	11
4	WISEGLOVE 参数指标 .....	12
4.1	WISEGLOVE 传感器分布图(含手指力反馈和手臂空间位置捕捉传感器) .....	12
4.2	典型参数指标 .....	12
5	手指传感器算法及手势定义 .....	13
5.1	归一化传感器值 .....	13
5.2	各传感器角度定义 .....	13
5.3	手势定义 .....	14
6	WISEGLOVE SDK 手套开发包 .....	14
6.1	在 VC 中调用手套 SDK 的方法 .....	14
6.2	开发库函数介绍 .....	15
7	WISEHAND SDK 虚拟手开发包 .....	22
7.1	在 VC/VS 中调用 WISEHAND SDK 的方法 .....	22
7.2	开发库函数介绍 .....	23
8	可能的问题 .....	24
8.1	手指弯曲角度和返回角度相差太多 .....	24
8.2	某个手指传感器分辨率不如从前好 .....	25
8.3	某个传感器值无变化 .....	25
8.4	软件找不到手套 .....	25
8.5	其他故障 .....	25

# WISEGLOVE (USB) USER'S MANUAL

## 1. 产品介绍

感谢您购买我公司的WISEGLOVE数据手套产品！您在使用中有任何疑问，请联系我们；如需了解更多最新产品和资讯，请访问我公司的官网。

电话：010-84673664    网址：<http://www.wiseglove.com>

WISEGLOVE 全系列产品都采用了光纤传感器。光纤传感器具有精度高，恢复性好，抗老化，使用寿命长的特点，因此光纤传感器被广泛应用于专业的科技领域。WISEGLOVE 采用了新型的手套模板设计，外观精美，佩戴舒适，结构紧凑，功能接口齐全，开发资料丰富，是 VR 开发应用中不可多得的上肢运动与力反馈整体解决方案。

本产品的应用定位于虚拟现实交互、上肢的运动捕捉、医学人手功能评定、机器人遥操作、康复医疗研究等应用领域。本产品可精确测量人手指动作和手臂动作，如挥手、抓握等手势动作；可软件设置手指的触觉反馈强度；可 WISEGLOVE 数据手套拥有新颖的外观设计，革新的传感器技术以及完善的用户开发接口，可提供完美的人机交互体验。配套的开发资料和示范工程代码，指引用户快速开始开发之旅。

### 一般功能

实时捕捉人手运动，实时驱动3D动画人手的运动；可以编程模拟3D鼠标对物体选取、移动、旋转等操作；通过力反馈手套组件可以把与虚拟物体的接触信息反馈给操作者，使操作者以更加直接，更高效的方式与虚拟世界进行交互，在一定程度上增强了VR交互性和沉浸感。

### 特别功能

手臂跟踪功能：可实时反馈手臂各关节在场景里的移动、旋转 *(适合带空间位置跟踪的型号)*

手指握力反馈：可实时测量人手手指握力 *(适合带手指握力反馈型号)*

### 适用平台

适合 WINDOWS 操作系统平台支持，可以适用 C++，C#，PYTHON 程序语言开发。

### 传感器配置

# WISEGLOVE (USB) USER'S MANUAL

WISEGLOVE扩展了传统传感器数据手套，标准配置有若干手指角度传感器（5/7/14/15/18/19个，参考WISEGLOVE数据手套对应的型号）；力反馈手套具有5个手指力反馈功能、手臂空间位置捕捉功能；增强版手套配置若干手指握力传感器（5/15/19个，参考WISEGLOVE数据手套对应的型号）。

## WISEGLOVE19E/E+数据手套的典型特性

- 统一版型,采用高弹性布料,适合大多数人手,佩带舒适自然,手感好。
- 使用了革新的第3代光纤传感技术,稳定性好,使用寿命长,回复精度高。
- 使用了12位精度AD转换器,能准确测量各传感器的角度变化。
- 5个手指指尖力反馈功能
- 手臂位置跟踪功能(仅WISEGLOVE19E+具备此功能)

## 2. 产品安装

### 2.1 包装明细

当您第一次打开手套包装，请确认所有附件是否齐全。如果缺少任何附件，请联系我公司重新配发附件产品。

WISEGLOVE数据手套包装里包含以下组件：

1. WISEGLOVE数据手套 1只
2. USB版本：USB数据线3米 1条  
蓝牙版本：蓝牙接收器1个，锂电池套件 1套
3. 软件U盘（内含说明书、软件开发包以及范例代码）1个
4. 合格证及保修卡 1张

# WISEGLOVE (USB) USER'S MANUAL

## 2.2 数据手套的组件



图1: WISEGLOVE数据手套 (USB版本) 配件



图2: WISEGLOVE数据手套 (蓝牙版本) 配件

**数据手套及配件清单请参考包装内装箱单文件。**

## 2.3 软件明细及安装方法

包装内光盘包含目录及文件：

- |                   |                                 |
|-------------------|---------------------------------|
| \DRIVER           | -- WINDOWS 驱动程序，适合 32 位和 64 位系统 |
| \EXE              | -- WISEGLOVE 数据手套演示执行文件         |
| \ OPENGLLIB       | -- OPENGL 开发库                   |
| \TESTWAVE         | -- WISEGLOVE 数据手套传感器波形显示测试例子    |
| \TESTWISEGLOVECMD | -- WISEGLOVE 数据手套传感器数据和角度的例子    |
| \TESTWISEHAND     | -- 骨骼动画手开发包测试例子                 |

# WISEGLOVE (USB) USER'S MANUAL

\WISEGLOVESDK            -- 数据手套 C++开发包

\WISEHANDSDK            -- 虚拟手开发包(只支持 32 位编译环境)

\ WISEGLOVE (USB) 说明书. PDF    -- 本用户手册

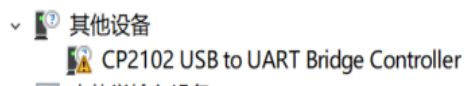
光盘的示例文件不需要安装, 只需要在本地建立新文件夹, 将上述文件拷贝到目录即可使用。

## 3 产品使用

### 3.1 使用前的准备工作

3.1.1 1) 对于 USB 版本数据手套, 将专用 USB 数据线连接手套到电脑主机 USB 口。2) 对于蓝牙版本的数据手套, 需将蓝牙接收器连接到主机 USB 口。系统提示有新硬件。

3.1.2 开始安装数据线的驱动程序, 驱动程序在资料盘根目录 DRIVER 目录下。首先进入 WINDOWS 设备管理器界面, 发现在其他设备栏里有:



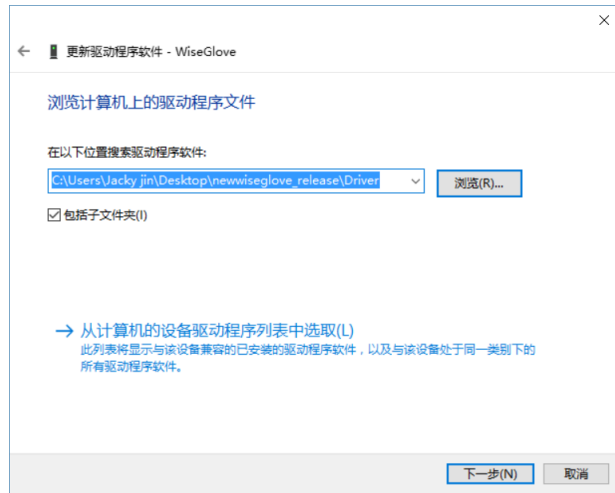
右键单击黄色惊叹号, 出现弹出对话框,

框, 并选择更新设备驱动程序后, 出现更新驱动程序对话框:

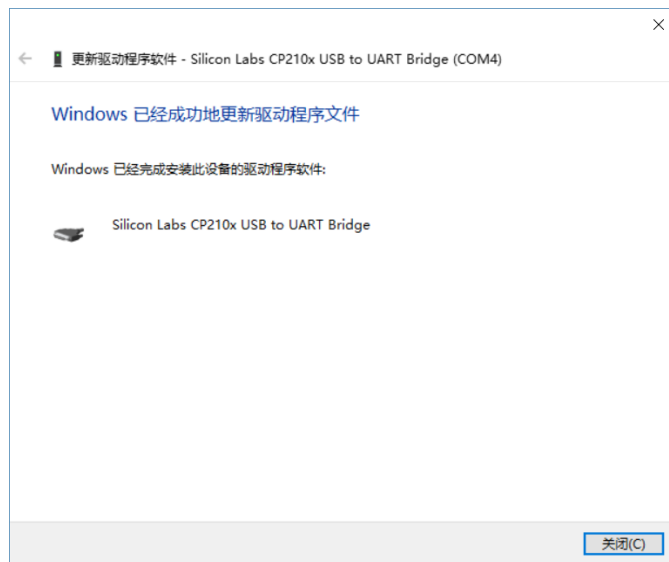


选择“浏览计算机以查找驱动程序软件”

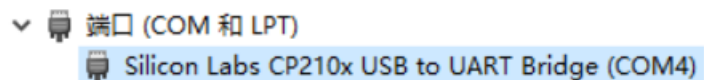
# WISEGLOVE (USB) USER'S MANUAL



点击“浏览”按钮，选择驱动程序所在的目录 Driver，点“下一步”等待驱动程序安装完毕。



这时在“设备管理器”里可以看到端口，说明驱动程序已经安装完毕。如果是有线手套则进入第四步，如果是蓝牙手套则进入第三步。



# WISEGLOVE (USB) USER'S MANUAL

## 3.2 穿戴需要注意的问题

- 3.2.1 左右手的手套不能互换穿戴。穿戴前注意手套上有传感器和电路单元的面和人手手背是同一方向。
- 3.2.2 不能对传感器折弯死弯。传感器的最小弯曲半径是 10 毫米左右，折死弯操作容易导致传感器断裂或者永久损坏。
- 3.2.3 穿手套时，要先松开手套的手腕处和手掌处的粘扣，并轻拉手心的布料缓慢穿戴好手套，再观察传感器是否在每个手指的背面，最好在每个手指的脊背位置比较合适。
- 3.2.4 脱手套时，应先松开手套的手腕处和手掌处的粘扣，且不能拉拽传感器靠近指尖的部分。在不好脱下的情况下，建议拉拽每个手指位于手掌心指中关节处的布料，在手指基本脱下的情况下，再拉食指与中指之间手掌心的布料，**切记不能拉拽传感器，或者布料下层的传感器数据线。**

## 3.3 运行范例程序

**请注意程序开始运行后会独占手套数据端口，因此同时只能有一个演示程序在运行。**

- 3.3.1 运行 testwiseglovecmd.exe。本程序是控制台的演示程序，程序运行后，显示手套型号(MODEL), 序列号(SN), 厂家(MANU)。最下方显示传感器原始数值，手指平直时，屏幕显示的是传感器最小值；手指握拳时，屏幕显示的是传感器最大值；每个传感器各自的最大值与最小值的差值应大于 300，否则该传感器可以认为已经损坏。

```
MODEL:WISEGLOVE5
SN:DG05S709L
MANU:XINTIAN
Sensor:0001 0002 0003 0004 0005
Sdata: 4095 4093 4095 4095 4095
```

图 3 TESTWISEGLOVECMD 运行界面

- 3.3.2 运行 testwave.exe。该程序动态显示传感器的运动数据曲线。手放平的情况下，执行重新标定后，再选择要显示数据曲线的传感器，例如 WISEGLOVE5 是 5 个传感器配置的手套，传感器可选范围是 sensor0~sensor4。WISEGLOVE14 是 14 个传感器配置的手套，传感器可选范围是 sensor0~sensor13。以此类推。



# WISEGLOVE (USB) USER'S MANUAL



图 4 TESTWAVE 运行界面

3.3.3 运行 testwisehand.exe。该程序运行后如果窗口没有最大化，请先最大化窗口。屏幕中央显示一个骨骼动画手模型，屏幕左边显示各个传感器的角度值。在当前程序是活动窗口的情况下，按下键盘 R 键（请先切换到英文输入状态下）表示重新标定手套，Y 键旋转手模型，Z 键手指握力传感器归零，1, 2, 3, 4, 5 键测试手指力反馈器，P 键停止反馈，Q 键退出程序。按照本手册 3.4 节的标定方法执行一遍标定后，即可开始体验数据手套。

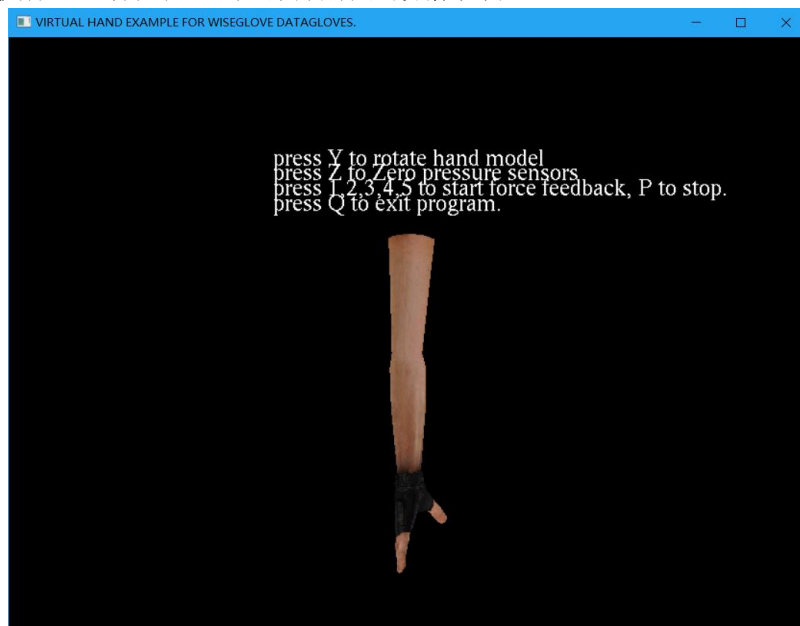


图 5 TESTWISEHAND 运行界面

## 3.4 手套的手指标定方法

手套的标定规定了 5 种标准手势，不同型号的数据手套的传感器分布不同，标定手势略有不同。建议用户穿戴好数据手套后，使用下述标定方法标定手套之后再开始使用。参考下述标定方法对手套标定之后再开始使用。每次穿戴好手套都需要重新执行一次标定过程。



图 6 WISEGLOVE 手套标定方法

## 3.5 手套的手臂标定方法

WISEGLOVE19FE+数据手套具有手臂捕捉功能，使用前需要标定手臂的初始姿态。身体面对显示屏，手臂竖直向下，且确保手臂传感器正确粘贴于手臂外侧，手臂标定姿态如下图。当手套开始通电启动时，需要等待 2-3 分钟传感器预热后，手在孔重做∞字标定。



图 7 手臂在空中画 8 字标定磁场

# WISEGLOVE (USB) USER'S MANUAL

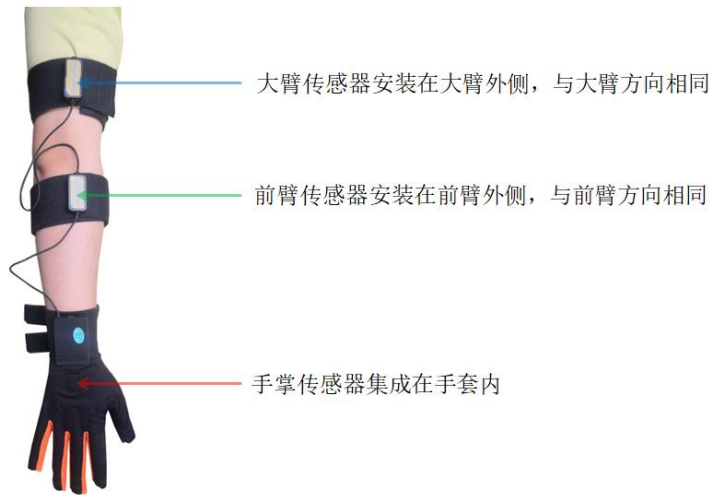


图 8 手臂传感器安装位置与手臂标定姿态

## 3.6 手套的握力传感器标定方法

WISEGLOVE19FE+数据手套具有手指握力传感器，使用前通电需要预热 2-3 分钟，等待传感器温度达到平衡后，执行传感器归零功能。在例子程序 TESTWISEHAND 程序中，键盘 Z 键调用传感器归零功能，如下图：

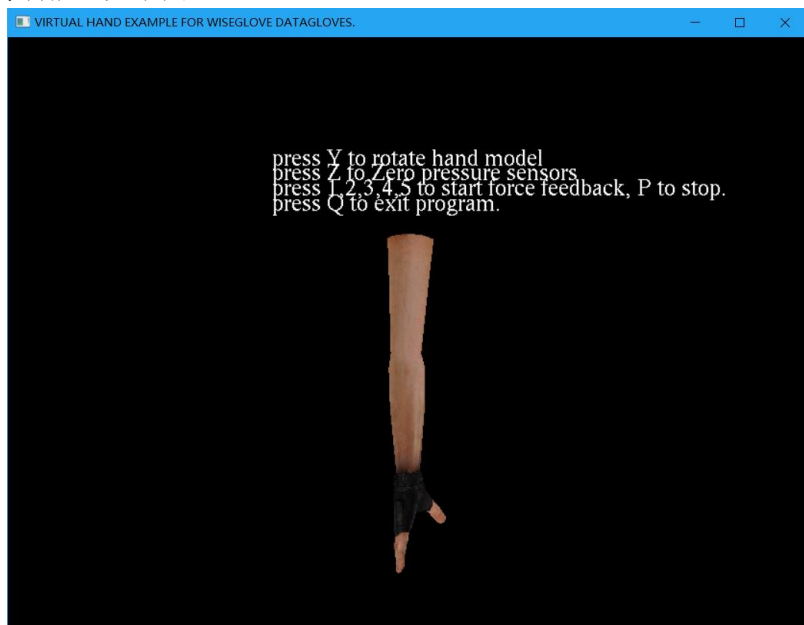


图 9 手指握力传感器初始归零

# WISEGLOVE (USB) USER'S MANUAL

## 4 WISEGLOVE 参数指标

### 4.1 WISEGLOVE 传感器分布图(含手指力反馈和手臂空间位置捕捉传感器)

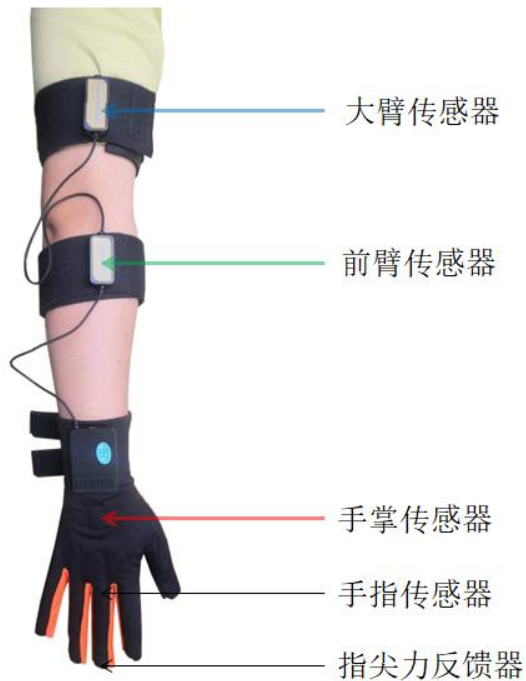


图 10 WISEGLOVE19FE+数据手套的传感器分布

### 4.2 典型参数指标

面料	合成弹力纤维布
手指角度传感器	数量 19 个传感器 采样精度 12bit 数据精度 动态精度 0.2°
手臂位置传感器*	3 个手掌传感器 精度 XY 轴 0.02° Z 轴 1°
握力传感器*	15 个手指握力传感器 24bit 数据精度 量程范围 2500 克 精度 9 克

# WISEGLOVE (USB) USER'S MANUAL

力反馈*	可选 5 个手指力反馈器 软件可编程 128 级反馈强度（0-7200RPM）
数据更新率	USB 版本 80HZ 蓝牙版本 50HZ
计算机接口	USB 或蓝牙 4.0
供电	5V @300mA

## 5 手指传感器算法及手势定义

### 5.1 归一化传感器值

一个传感器值在 A（最小值）和 B（最大值）之间。在标定手套的时候就已经完成了最大值和最小值的标定。例如, 如果当前的手套值是 C, 则满足的关系是  $A \leq C \leq B$ , 伪语言表示为:

if (C>B) B=C

if (C<A) A=C

以上两条保证了当前值 C 永远在 A 和 B 之间。假设某一个手指的活动角度是 E, 则当前的弯曲角度 F 可以通过下面简单算法得到:

$$F = E * (C - A) / (B - A)$$

### 5.2 各传感器角度定义

针对一般人手的生理特性，在开发库中使用了以下传感器的角度定义，用户也可以根据本人手指的运动角度，调用函数库相应的函数设置合适的角度范围参数，见下表 1，表 2:

表 1 余指运动范围:

关节运动范围		食指（度）	中指（度）	无名指（度）	小指（度）
指端	弯曲	83	80	75	78
	外展	12	12	12	12
指中	弯曲	101	103	105	103

## WISEGLOVE (USB) USER'S MANUAL

	外展	12	12	12	12
指根	弯曲	83	90	88	90
	外展	22	22	22	22
总角度		257	273	268	271

表 2 拇指运动范围：

	运动	角度
指端	弯曲	100
指中	弯曲	45
指根	弯曲	42
	外展	53

### 5.3 手势定义

如前面所述，一个手指的归一化活动范围是【0, 1】，我们可以认为当  $(C-A)/(B-A) < 0.5$  时手指是开， $(C-A)/(B-A) > 0.5$  时手指是合，通过这个函数关系，用户可以定义每个手指的开合，组成各种手势，自定义的手势可用来执行程序预定义的各种指令。

## 6 WISEGLOVE SDK 手套开发包

本 SDK 提供了 C++ 形式的静态库和动态库，对于 VC/VS 编译器可以直接静态链接编译，对于其他语言（例如 C#, JAVA 等）需要使用调用 DLL 的方法才能使用。

### 6.1 在 VC 中调用手套 SDK 的方法

- 1) 在工程源文件中加入 `#include "Wiseglove.h"`
- 2) 在连接输入库设置里加入 `Wiseglove.lib`
- 3) 把 `Wiseglove.dll` 复制到项目执行文件目录下，或者其他系统 PATH 路径下。

## 6.2 开发库函数介绍

用户可以参考光盘里的例子程序和 SDK 头文件说明，最简单的调用方法在范例工程 TESTWISEGLOVECMD 里有描述。

**函数名:** `int Scan(void)`

功能: 查询电脑中存在的 WISEGLOVE 数据手套

参数: 无

返回: 存在数据手套的个数 (例如 2 个手套, 返回 2)

**函数名:** `unsigned char GetPort(unsigned char num)`

功能: 取得数据手套 COM 口, 例如 COM1 表示为 1, 以此类推

参数: num, 为手套在系统中数据结构中的索引, num = 0 表示 Scan() 函数扫描到的第一个手套 COM 口

返回: COM 口索引

**函数名:** `unsigned int GetPortBaudrate(unsigned int num) = 0;`

功能: 取得串口通讯波特率, 返回数据为 115200 或者 256000 之一

参数: num, 为手套在系统中数据结构中的索引, num = 0 表示 Scan() 函数扫描到的第一个手套 COM 口

返回: 串口波特率值

**函数名:** `bool Open(unsigned char nPort, int baudrate)`

功能: 使用用户指定的串口端口号打开手套端口, 在已知手套 COM 口的情况下, Scan() 和 GetPort() 函数不是必须要调用的

参数: unsigned char nPort, COM 端口, 例如 num=1, 表示 COM1 端口

Int baudrate, 串口波特率参数, 一般手套参数值为 115200, 特俗定制版本的手套参数值为 256000

返回: 成功返回 true(1), 否则返回 false(0)

## WISEGLOVE (USB) USER'S MANUAL

**函数名:** void Close( void )

功能: 调用本函数关闭数据端口

参数: 无

返回: 无

**函数名:** bool LoadCalib(char \*filename)

功能: 在使用手指传感器手动标定时, 使用本函数加载标定数据文件

参数: char \*filename, 标定文件名字串, 例如" c:\\calib.txt"

返回: 成功返回 true(1), 否则返回 false(0)

**函数名:** bool SaveCalib(char \*filename)

功能: 在使用手指传感器手动标定的情况下, 使用本函数保存标定数据

参数: char \*filename, 标定文件名字串, 例如" c:\\calib.txt"

返回: 成功返回 true(1), 否则返回 false(0)

**函数名:** void SetCalibMode(int mode)

功能: 设置手指传感器标定模式

参数: int mode

mode = CALIB\_AUTO, 自动标定

mode = CALIB\_MANU, 手动标定

返回: 无

**函数名:** void SetCalib(int index, unsigned short min, unsigned short max, float angle)

功能: 手指传感器设置标定参数

参数: int index, 传感器索引值, 例如 19 传感器型号的范围为【0, 18】

unsigned short min, 最小值限制

unsigned short max, 最大值限制

float angle, 关节活动角度范围 (度为单位)

返回: 如果成功 返回 TRUE(1), 否则返回 FALSE(0)



## WISEGLOVE (USB) USER'S MANUAL

**函数名:** void SetCalibmin(int index, unsigned short min)

功能: 手指传感器设置最小值参数

参数: int index, 传感器索引值, 范围【0, 18】

unsigned short min, 最小值限制

返回: 如果成功 返回 TRUE(1), 否则返回 FALSE(0)

**函数名:** void SetCalibmax(int index, unsigned short max)

功能: 手指传感器设置最大值参数

参数: int index, 传感器索引值, 范围【0, 18】

unsigned short max, 最大值限制

返回: 如果成功 返回 TRUE(1), 否则返回 FALSE(0)

**函数名:** void SetCalibRange(int index, float angle)

功能: 手指传感器设置关节活动角度范围参数

参数: int index, 传感器索引值, 范围【0, 18】

float angle, 关节活动角度范围

返回: 如果成功 返回 TRUE(1), 否则返回 FALSE(0)

**函数名:** unsigned short GetCalibmin(int index)

功能: 取得手指传感器最小值限制值

参数: int index, 传感器索引值, 范围【0, 18】

返回: 如果成功 返回最小值限制值, 否则返回 0

**函数名:** unsigned short GetCalibmax(int index)

功能: 取得手指传感器最大值限制值

参数: int index, 传感器索引值, 范围【0, 18】

返回: 如果成功 返回最大值限制值, 否则返回 0

**函数名:** float GetCalibRange(int index)

功能: 取得手指传感器关节活动角度范围

## WISEGLOVE (USB) USER'S MANUAL

参数: int index, 传感器索引值, 范围【0, 18】

返回: 如果成功 返回关节活动角度范围, 否则返回 0

**函数名:** void ResetCalib()

功能: 调用本函数可以恢复手指传感器的所有标定值到缺省状态, 标定模式恢复为自动标定模式

参数: 无

返回: 无

**函数名:** bool SetMin\_adj(int index, float min\_adj)

功能: 设置传感器最小角度限制

参数: int index, 传感器索引值, 范围【0, 18】

float min\_adj, 最小角度限制值, 范围【0, 0.3】

返回: 成功返回 true(1), 否则返回 false(0)

**函数名:** float GetMin\_adj(int index)

功能: 读传感器最小角度限制

参数: int index, 传感器索引

返回: 如果成功 返回最小角度限制, 否则返回 0

**函数名:** BOOL SetMax\_adj(int index, float max\_adj)

功能: 设置传感器最大角度限制

参数: int index, 传感器索引值, 范围【0, 18】

float max\_adj, 最大角度限制值, 范围【0, 0.3】

返回: 如果成功 返回 TRUE(1), 否则返回 FALSE(0)

**函数名:** float GetMax\_adj(int index)

功能: 读传感器最大角度限制

参数: int index, 传感器索引值, 范围【0, 18】

返回: 如果成功 返回最大角度限制, 否则返回 0

## WISEGLOVE (USB) USER'S MANUAL

**函数名:** `BOOL GetSn(unsigned char *sn)`

功能: 读数据手套序列号字符串

参数: `unsigned char * sn`, 保存序列号字符串, 一般设置为 16 个字节

返回: 如果成功 返回 `TRUE(1)`, 否则返回 `FALSE(0)`

**函数名:** `BOOL GetModel(unsigned char *model)`

功能: 读数据手套型号字符串

参数: `unsigned char * model`, 保存型号字符串, 一般设置为 16 个字节

返回: 如果成功 返回 `TRUE(1)`, 否则返回 `FALSE(0)`

**函数名:** `BOOL GetManu(unsigned char *manu)`

功能: 读数据手套厂家字符串

参数: `unsigned char * manu`, 保存厂家字符串, 一般设置为 16 个字节

返回: 如果成功 返回 `TRUE(1)`, 否则返回 `FALSE(0)`

**函数名:** `int GetNumOfSensor(void)`

功能: 读手指传感器数量

参数: 无

返回: 如果成功 返回传感器数量值, 否则返回 0

**函数名:** `int GetNumOfPressure (void)`

功能: 读握力传感器数量

参数: 无

返回: 如果成功 返回握力传感器数量, 否则返回 0

**函数名:** `int GetNumOfArm (void)`

功能: 读手臂传感器数量

参数: 无

返回: 如果成功 返回握力传感器数量, 否则返回 0

**函数名:** `unsigned int GetData(unsigned short *data)`

功能：读所有传感器值

参数：unsigned short \* data, 传感器数据，最多 19 个 unsigned short 型数据

返回：如果成功 返回当前数据包的时间戳(ms)，否则返回 0

**函数名：** unsigned int GetScaledData(unsigned short \*data)

功能：读所有传感器归一化值，范围【0, 4095】

参数：unsigned short \* data, 传感器数据，最多 19 个 unsigned short 型数据

返回：如果成功 返回当前数据包的时间戳(ms)，否则返回 0

**函数名：** unsigned int GetAngle(float \*angle)

功能：读手指传感器的角度

参数：float \*angle, 传感器角度数据， 5-19 个 float 型数据

返回：如果成功 返回当前数据包的时间戳(ms)，否则返回 0

**函数名：** unsigned int GetPressureRaw (unsigned short \*pressure)

功能：读手指压力传感器值

参数：int \*pressure, 传感器值， 5/15/19 个 int 型数据, 单位克

返回：如果成功 返回当前数据包的时间戳(ms)，否则返回 0

**函数名：** unsigned int GetQuat(float \*quat)

功能：读手臂传感器的角度

参数：float \* quat, 传感器角度数据，最多 12 个 float 型数据，依次为手掌，前臂，上臂关节角度的四元素表示(w0, x0, y0, z0), (w1, x1, y1, z1), (w2, x2, y2, z2)

返回：如果成功 返回当前数据包的时间戳(ms)，否则返回 0

**函数名：** void RestQuat(void)

功能：标定手臂传感器。标定之前请确认：手臂自然伸直且自然下垂，上臂传感器带有线的一个侧面朝地面且与水平面平行。执行此函数后，程序将自动匹配模型手与传感器的对应关系。

参数：无

## WISEGLOVE (USB) USER'S MANUAL

返回：无

**函数名：** void ZeroPressure (void)

功能：握力传感器设置逻辑 0 克，当握力传感器没有负载时，执行此函数，所有传感器的值清零

参数：无

返回：无

**函数名：** void CalibPressure(int index)

功能：标定握力传感器，首先握力无负载时握力值清零，再加 500 克砝码，竖直压在传感器上表面后，执行此函数。所有握力传感器在出厂时均已标定，一般情况下不需要再做标定，除非遇到使用时握力远超过设计量程 2500 克力时，才需要重新标定。

参数：无

返回：无

**函数名：** bool SetFeedBack(unsigned char \*fddata)

功能：设置手指力反馈器的强度参数

参数：unsigned char \*fddata，为 5 个无符号字节型数据，范围为 0-255，当值为 0 时，反馈器停止，值越大时，反馈器输出强度越大

返回：如执行成功返回 true(1)，否则返回 false(0)

**函数名：** WiseGlove\* GetWiseGlove()

功能：用于新建一个手套指针，在对手套端口操作之前，首先必须要调用本函数新建一个手套指针，例如，WiseGlove\* pGlove = GetWiseGlove();

参数：无

返回：Wiseglove 型指针（备注，任何时候只要调用了该函数一定要在程序退出前执行 DelWiseGlove() 销毁指针，否则会造成内存泄漏）

**函数名：** void DelWiseGlove(WiseGlove\* p)

功能：用于销毁一个手套指针，只要调用过 GetWiseGlove() 创建过手套指针，那么程序退出前，

## WISEGLOVE (USB) USER'S MANUAL

一定要执行本函数释放资源，否则会造成内存泄漏。

参数: WiseGlove\* p, Wiseglove 型指针

返回: 无

### 7 WISEHAND SDK 虚拟手开发包

本 SDK 提供了 C++形式的静态库和动态库，对于 VC/VIS 编译器可以直接使用，对于的其他编译器可能需要转换格式后使用。



图 11 WISEHAND 人手开发库

#### 7.1 在 VC/VIS 中调用 WISEHAND SDK 的方法

- 1) 在工程源文件中加入 `#include "Wisehand.h"`
- 2) 在连接输入库设置里加入 `Wisehand.lib`
- 3) 把 `Wisehand.dll` 复制到项目执行文件目录下，或者其他系统 PATH 路径下

## 7.2 开发库函数介绍

具体用法用户可以参考光盘里的 testwisehand 例子程序和 SDK 头文件说明。

**函数名:** BOOL loadModelData( char \*filename )

功能: 绘制本开发包提供的虚拟手之前, 要调用本函数加载手模型

参数: char \*filename, 文件名字串, 例如 "c:\\\\lefthand.obj"

返回: 如果成功加载 返回 TRUE(1) 否则返回 FALSE(0)

**函数名:** void setVhandtype( bool type )

功能: 根据实际手套的左右手类型, 设置手模型类型

参数: BOOL type, type = VHAND\_LEFT 为左手模型, type = VHAND\_RIGHT 为右手模型

返回: 无

**函数名:** void drawhand(float angle[19], int mode)

功能: 在 OPENGL 环境中绘制虚拟手模型

参数: float angle[19], 19 个自由度角度数据, 单位为度

int mode, mode 绘图模式

mode=0 画纹理贴图

mode=1 为画网格

mode=2 画填充面

返回: 无

**函数名:** void DrawJoints(int type)

功能: 在 OPENGL 环境中绘制虚拟手模型骨骼

参数: int type, type = 1 画关节点间连线, type=0 为画点

返回: 无

**函数名:** WiseHand\* GetWiseHand(void)

## WISEGLOVE (USB) USER'S MANUAL

功能：用来新建一个手模型指针，如果要绘制本 SDK 提供的模型手，首先必须要建立一个指针，例如 `WiseHand * myhand = GetWiseHand()`

参数：无

返回：WiseHand 型指针

**函数名：**`void DelWiseHand(WiseHand* p)`

功能：用于销毁一个模型手指指针，只要调用过 `GetWiseHand()` 创建过手套指针，那么程序退出前，一定要执行本函数释放资源，否则会造成内存泄漏。

参数：`WiseHand * p`, WiseHand 型指针

返回：无

## 8 可能的问题

### 8.1 手指弯曲角度和返回角度相差太多

1. 首先请检查是否做了握拳标定？如果没有请参考本手册 3.4 节做好标定后再测试。否则进行下一步。
2. 请检查该传感器是否在对应的手指关节上。例如 5 传感手套的传感器分布：拇指传感器在指尖关节上，余指传感器指中关节上。14 传感手套的传感器分布：拇指传感器在指尖、指中关节上，余指传感器在掌指、指中关节上。
3. 这个问题属于传感器不同步的情况，在出厂时，都经过调校达到较好的同步状态，使用过程中反复穿戴时可能会引起某个或者某几个传感器前后位置移动，某个传感器与其它的传感器不同步时，需要轻微调整这个传感器的前后（指尖-手掌方向）位置关系，重新标定后再测试，如果仍然不能符合要求，建议返厂调试。



## 8.2 某个手指传感器分辨率不如从前好

使用光盘附带的 TESTWISEGLOVECMD 程序观察传感器的返回值，手指平伸时的值和手指握拳时的值是否很接近？如果非常接近（平摊手掌和握拳时传感器差值小于 300），那么该传感器已经损坏，需要更换新的传感器才可以使用，联系我公司维修或更换。

## 8.3 某个传感器值无变化

若其他手指的传感器值有变化，单独此传感器值无变化。可能情况：1. 传感器在布料里翻转了，在不戴手套的情况下，尝试把传感器往反方向弯曲，是否有角度变化，如有变化则可以用镊子辅助把传感器翻转 180°。2. 该传感器本身损坏或者信号线断开，需要返厂修理，联系我公司维修或者更换。

若所有传感器值都没有变化，USB 版本的手套请重新拔插手套专用数据线，再启动程序观察是否正常。蓝牙版本的手套，请检查蓝牙连接是否正常或手套电池是否需要充电。

## 8.4 软件找不到手套

8.4.1 是否有其他应用程序占用了手套端口？

8.4.2 是否为手套安装好驱动程序？驱动文件在本资料盘中。

8.4.3 没有使用自动初始化函数？如果是，请检查测试程序的串口端口号是否和手套所在端口号一致？

8.4.4 USB 数据线不能联通电脑和手套？建议更换 USB 口或者 USB 线尝试。

## 8.5 其他故障

如果您有其他疑难问题不能解决，请联系获取帮助。