智元素格斗机器人体感应用说明书

文档版本号	更新时间	修订人	备注
v1.0	2019-04-03	邱增顺	初版
V1.1	2019-04-05	邱增顺	排版
V1.2	2019-04-10	邱增顺	增加内容
V1.3	2019-04-24	郭磊、邱增顺	新体感软件说明
V1.4	2019-04-25	邱增顺	增加内容
V1.5	2019-07-10	郭磊、邱增顺	控制软件更新
V1.6	2019-07-19	邱增顺	增加内容

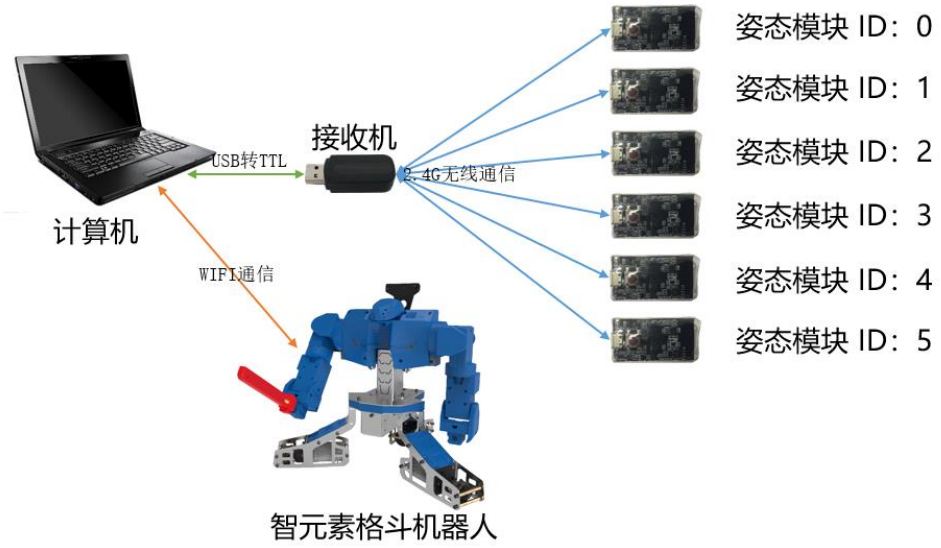
目录

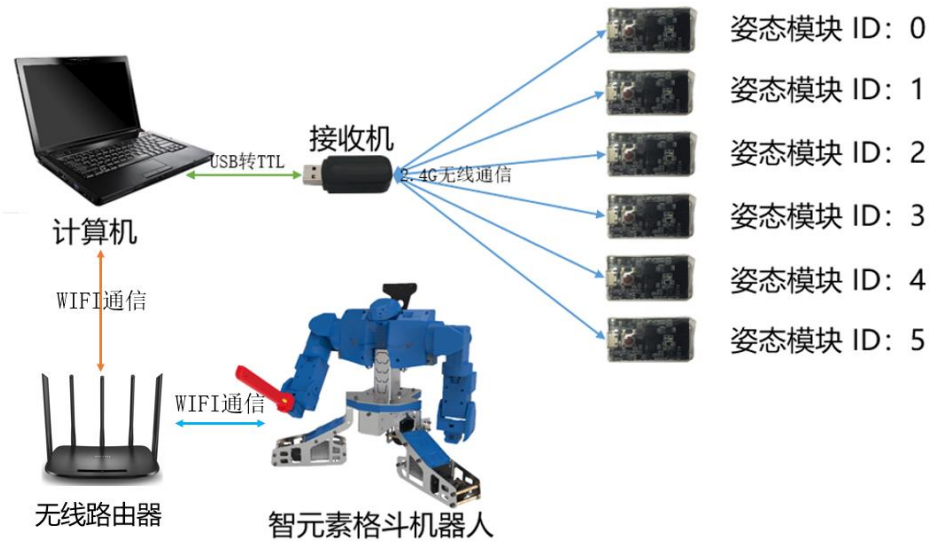
1. 体感应用基本原理 3
2. 无线体感套装 3
2.1. 无线体感套装介绍 3
2.2. 无线体感套装使用 4
2.2.1. 安装驱动 4
2.2.2. 打开上位机软件 4
2.2.3. 配置接收机和姿态模块 5
2.2.4. 穿戴姿态模块 8
3. 智元素机器人控制方法 9
3.1. 计算机与机器人连接 9
3.2. 智元素控制系统与体感控制应用 Demo 的使用 14
4. 体感应用开源软件介绍 22
4.1. 安装 Visual studio 2017 23
4.2. 运行上位机软件 25
4.3. 上位机软件介绍 27
4.3.1. 算法介绍 27
4.3.2. 底盘控制参数介绍 32
5.注意 33

1. 体感应用基本原理

体感应用原理如下：

姿态模块通过 2.4G 无线通信方式将解算好的姿态数据实时发送给接收机，然后接收机将 6 个姿态模块的姿态数据通过 USB 转 TTL 发送给计算机，接下来计算机再通过特定的算法将得到的姿态数据计算成智元素格斗机器人相应的舵机参数、底盘控制参数、还有技能参数并通过 wifi 发送给智元素格斗机器人。这里计算机与机器人之间的连接除了可以直连之外，还可以通过无线路由器进行连接，后面的章节会详细介绍。





2. 无线体感套装

2.1. 无线体感套装介绍

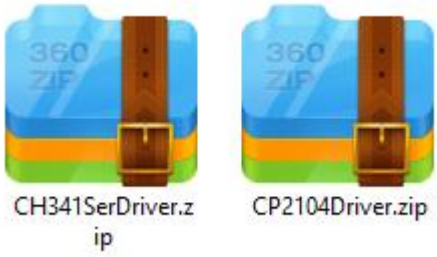
无线姿态模块是一款 9 轴姿态传感器，它集成了先进的算法以及校准技术，直接输出全姿姿态数据，不仅可以进行星型组网还可以多组设备同时工作。

接收机专门用于介绍一个或者多个无线姿态模块的数据，它可以配置数据输出协议、输出速率、模块 ID 等信息，方便与上位机通信。

2.2. 无线体感套装使用

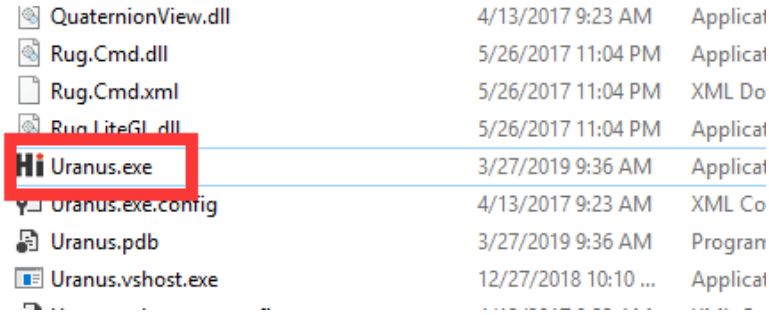
2.2.1. 安装驱动

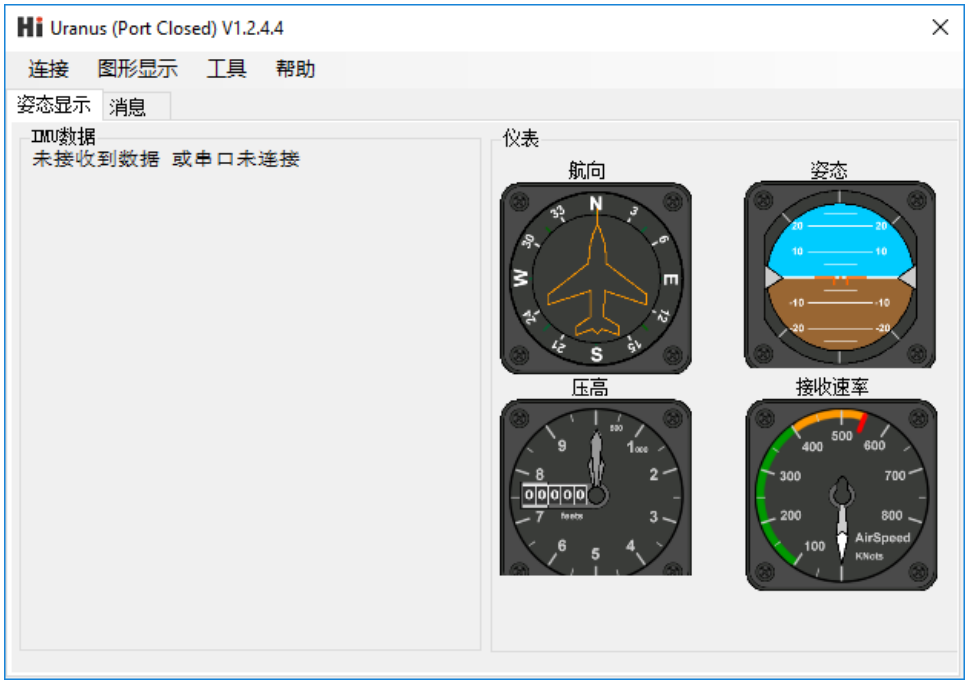
打开“串口驱动程序”文件夹，在该文件夹下找到“CH341SerDriver.zip”和“CP2104Driver.zip”压缩包，解压并分别安装。



2.2.2. 打开上位机软件

打开“Uranus 上位机”文件夹，在该文件夹下双击打开“Uranus.exe”。





2.2.3. 配置接收机和姿态模块

当用户初次拿到姿态模块需要对模块的 ID（0-7）和信道号（0-4）进行配置，而接收器只需要配置信道号。

示例：将一个接收器的信道号设置为 1， 将 6 个模块的自身 ID 设置为 0,1,2,3,4,5 并连接到这个接收器上：

接收器配置：AT+GWCH=1

模块 0 配置： AT+GWCH=1 AT+ID=0

模块 1 配置： AT+GWCH=1 AT+ID=1

模块 2 配置： AT+GWCH=1 AT+ID=2

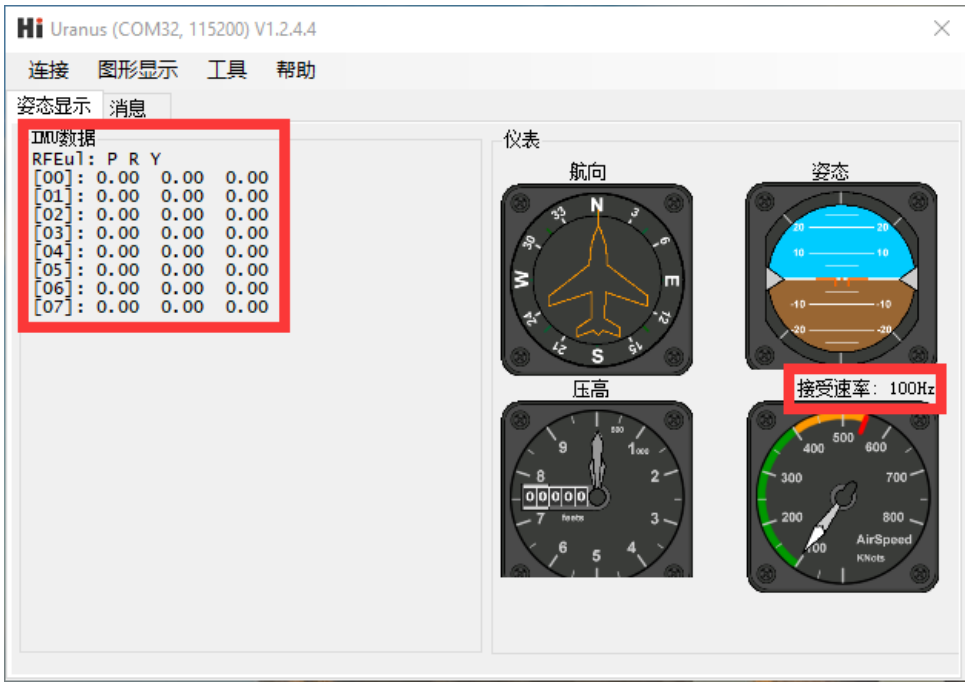
模块 3 配置： AT+GWCH=1 AT+ID=3

模块 4 配置： AT+GWCH=1 AT+ID=4

模块 5 配置： AT+GWCH=1 AT+ID=5

具体配置方法如下：

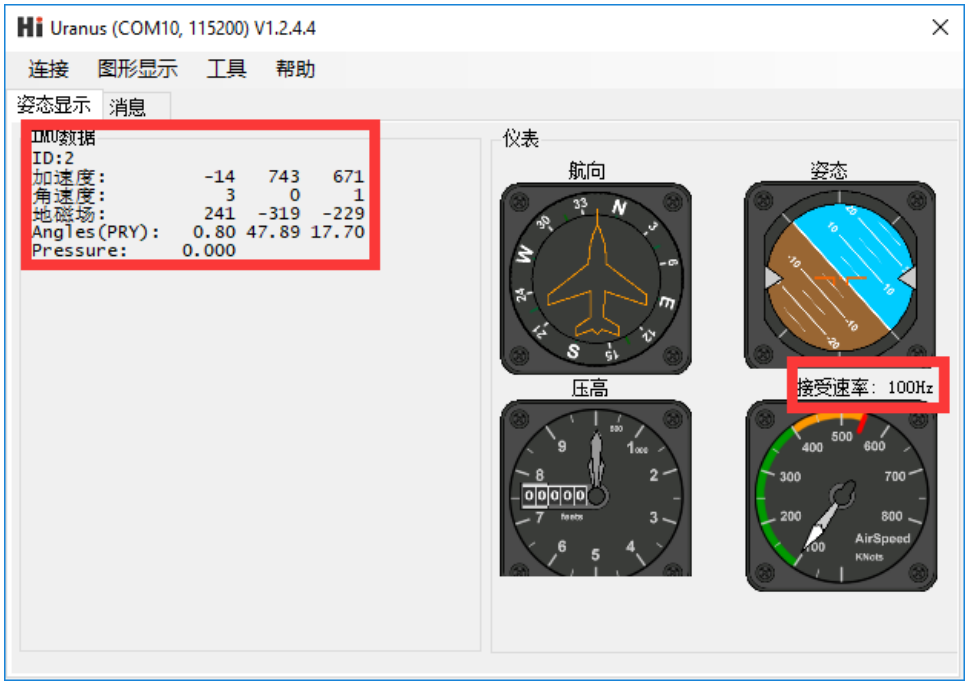
1) 配置接收机，将接收机插在计算机的 USB 接口上，点击“连接”-“串口”-选择对应的 COM 号，如果连接成功会出现 IMU 数据，并且接收速率为 100Hz。



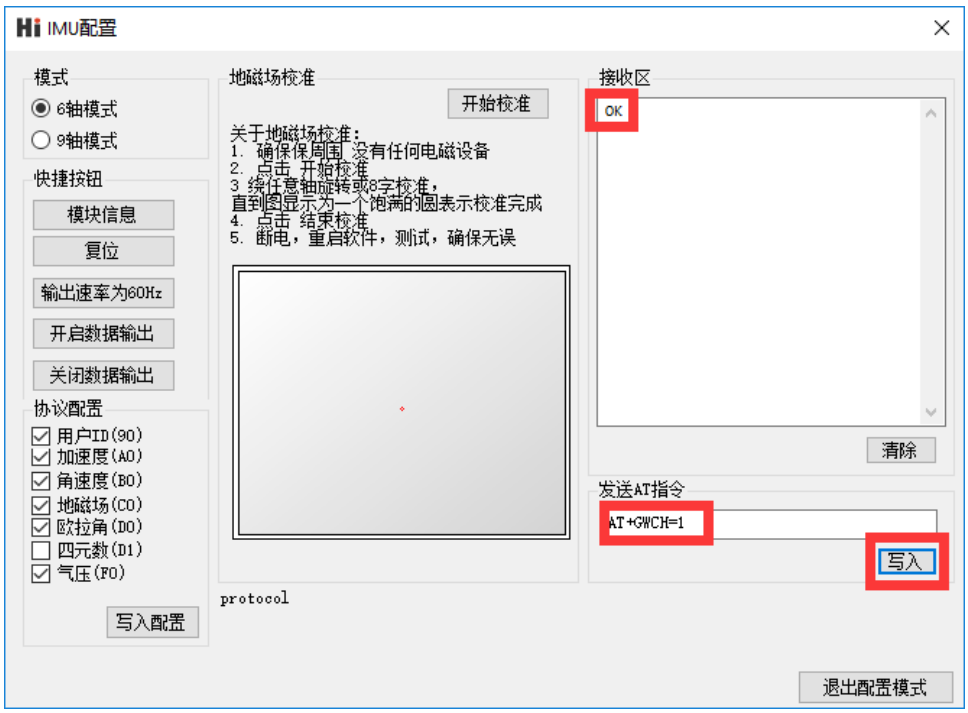
2) b.配置接收机，点击“工具”-“配置模块”，在发送 AT 指令处输入 AT+GWCH=1，配置接收机信道为 1，点击“写入”,如果在接收区出现下图字样，代表配置成功。然后在发送 AT 指令处输入 AT+SETPTL=71，配置接收机输出四元数集合，点击“写入”，如果在接收区出现下图字样，代表配置成功。最后重新上电模块使配置生效。

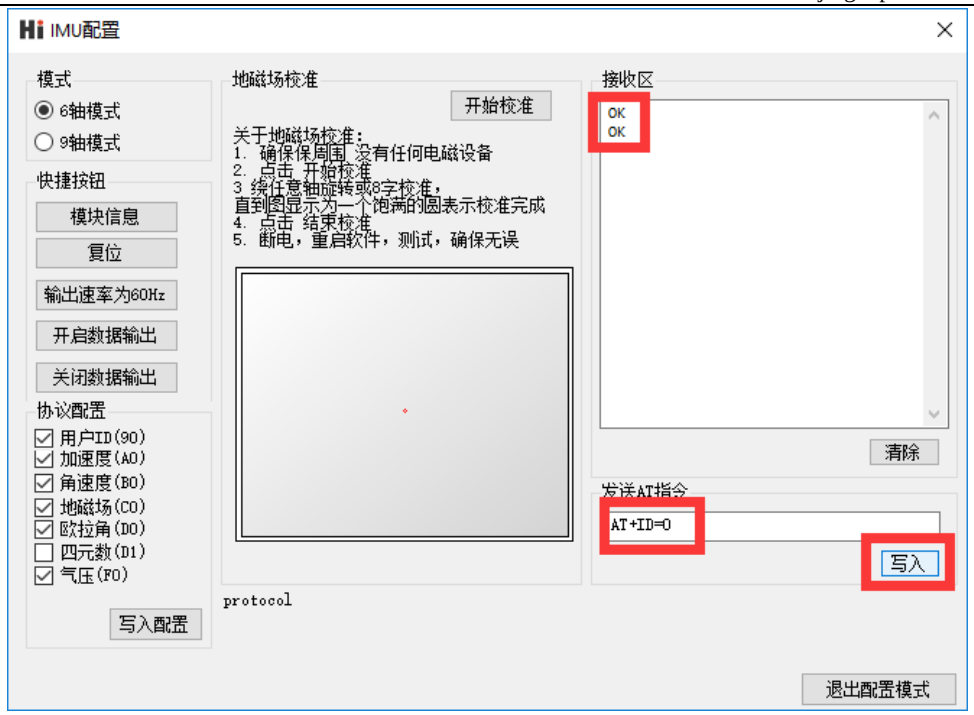
3) c.配置姿态模块，使用 micro 数据线（注意不是一拖三充电线）连接姿态模块和计算机，重新打开 Uranus 上位机软件，长按姿态按键 2S 开启电源（再长按 2S 可以关闭），点击“连接”-“串口”-选择对应的

COM 号，如果连接成功会出现 IMU 数据并且接收速率为 100Hz。



4) d.配置姿态模块，点击“工具”-“配置模块”，在发送 AT 指令处输入 AT+GWCH=1，点击“写入”，再次输入 AT+ID=0，再点击“写入”，最后接受去出现两次“OK”，则证明配置成功，随后重新上电模块，此时重新上电模块就能使配置生效。

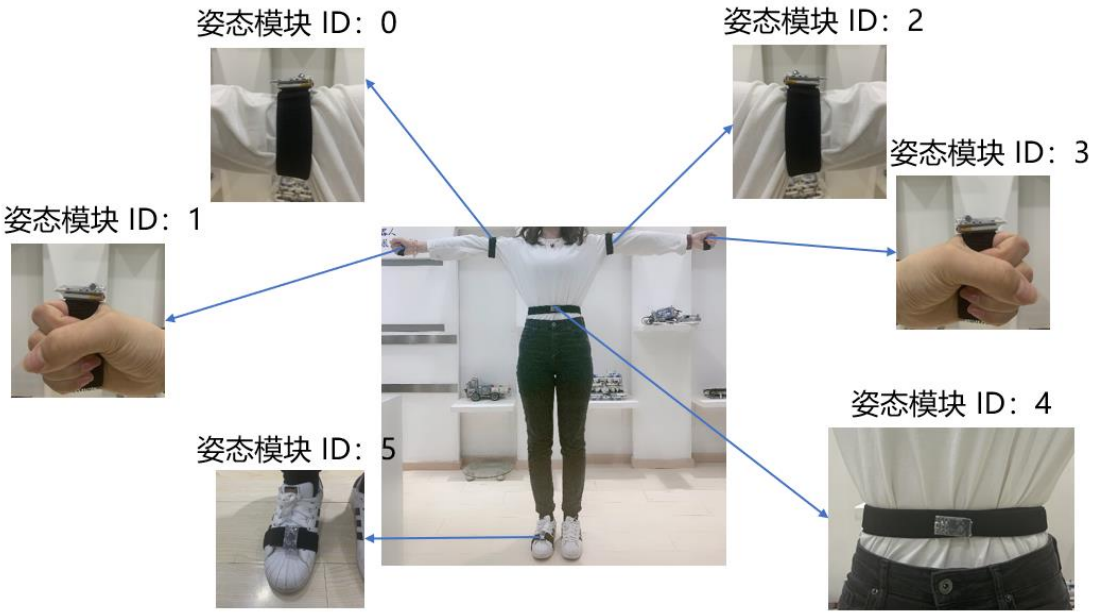




5) e.重复步骤 c、d，改变 ID 号，直到配置完所有模块。
6) f.如果此时接收机正连接着上位机软件，并且所有无线姿态模块处于开机状态，上位机则能显示出 6 个姿态模块的姿态数据。

2.2.4. 穿戴姿态模块

如下图所示，将松紧绑带捆绑在身体相应位置，身体摆成“T”字形，打开所有姿态模块，然后将姿态模块粘在绑带上面，这里需要注意的是，姿态模块对应 ID 顺序，右臂上是 0、1 号姿态模块，左臂上是 2、3 号姿态模块，另外 0 号和 1 号姿态模块初始状态尽量保持在同一水平线上，2 号和 3 号也要保持在同一水平线上。（注意：模块上电之前，请先将模块置于水平桌面上，然后再给姿态模块上电，这样得到的初始数据更加准确。）



3. 智元素机器人控制方法

3.1. 计算机与机器人连接

计算机与机器人连接有两种方法，一种方法是使用计算机无线直连机器人，另外一种方法是将机器人与计算机都连接到相同的局域网内。

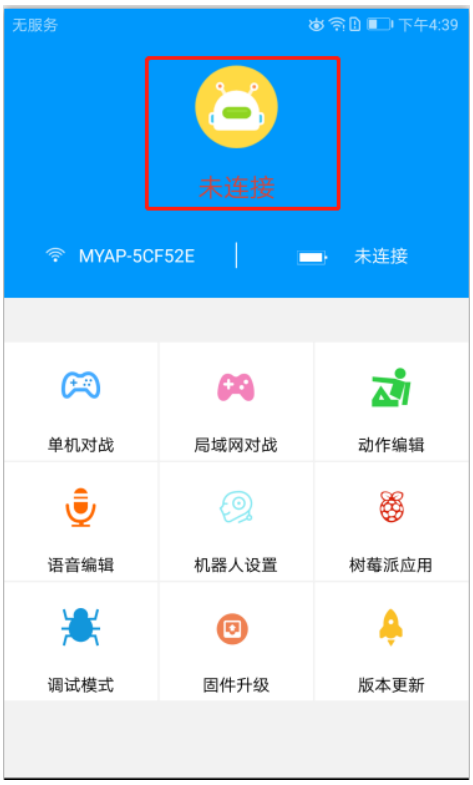
方法一：计算机直连机器人，查看机器人底盘上的热点名称→连接机器人的热点，选择上述对应的机器人热点名称，输入密码（默认 88888888）。

方法二：将机器人与计算机都连接到相同的局域网内。

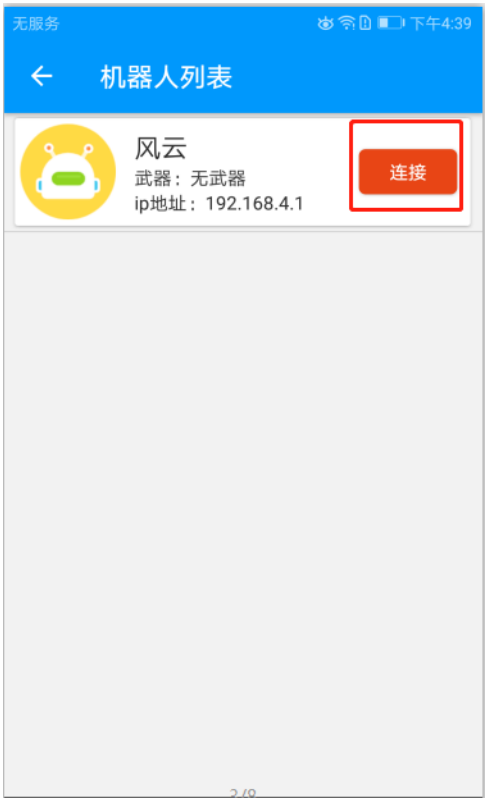
1) 查看机器人底盘上的热点名称。

2) 将手机连接到上图中热点名称的 wifi 网络中去，输入密码（默认 88888888）。

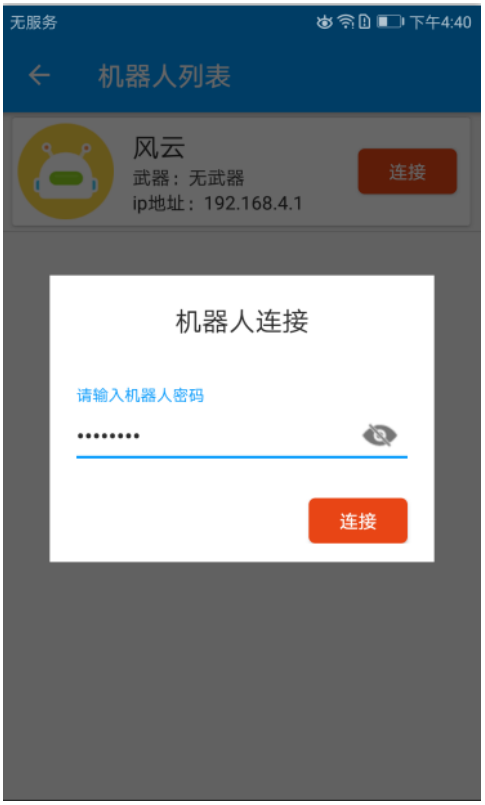
3) 打开智元素手机 app(下载地址见包装清单上的二维码)，点击头像进入机器人列表。



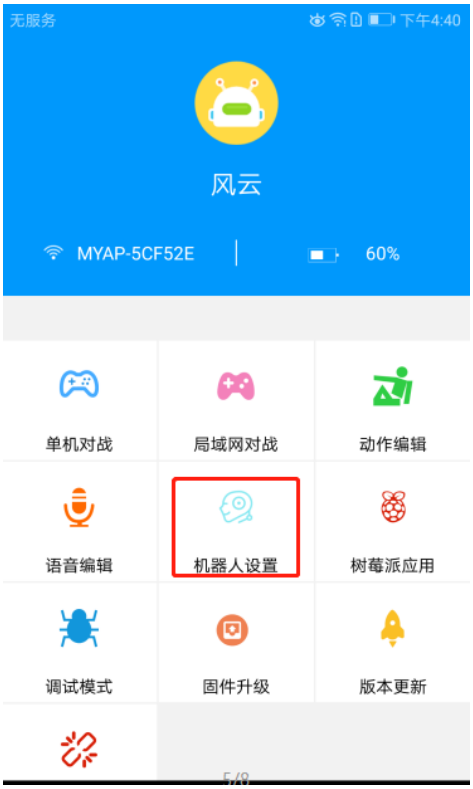
4) 在机器人列表即可看到当前网络中的机器人，如未搜到可以进行下拉刷新，重新搜索。



5) 点击列表中的“连接”按钮输入密码（默认 88888888），点击“连接”。



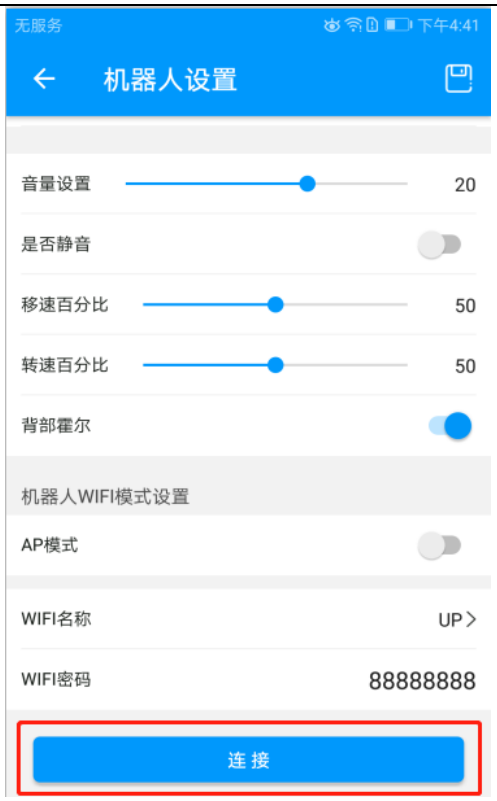
6) 连接成功后回退到主界面，点击“机器人设置”进入机器人设置页面。



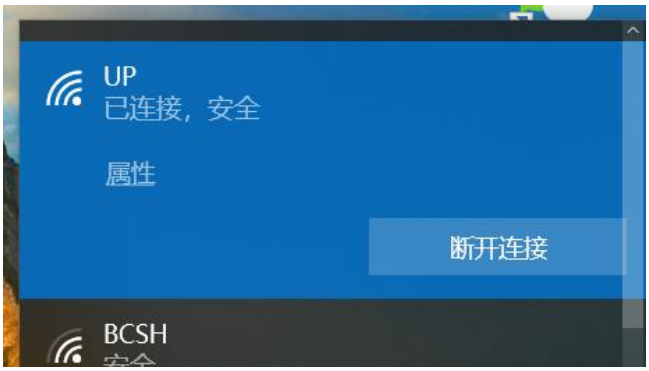
7) 在最下面可以看到机器人 wifi 设置，把 ap 模块的开关关掉之后出现 wifi 名称和 wifi 密码 2 个框。

8) 点击 wifi 名称去选择附近的热点，点击 wifi 列表中我们要连入的无线 wifi 热点。

9) 在 wifi 密码框中输入此 wifi 密码，点击“连接”，机器人即可连接到所选热点当中（当机器人自动连接到热点之后，就会***关闭本身的热点***，如果想再次开启热点就要手动按下机器人肩膀上的按钮，当提示“两秒内松开按键，开启热点”后松开按键，即可重新开启热点）。



10) 把计算机连接到与机器人相同的网络里。

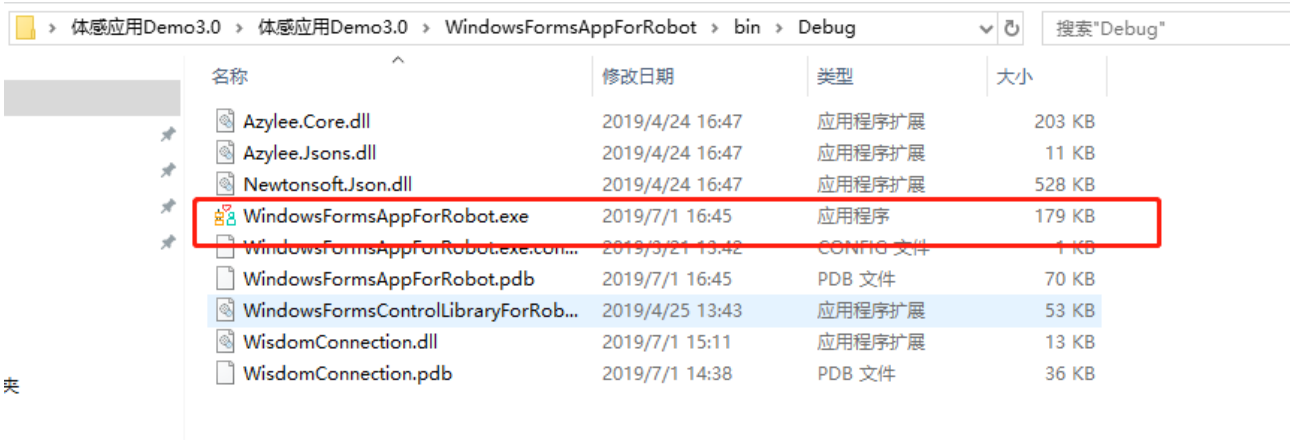


3.2. 智元素控制系统与体感控制应用 Demo 的使用

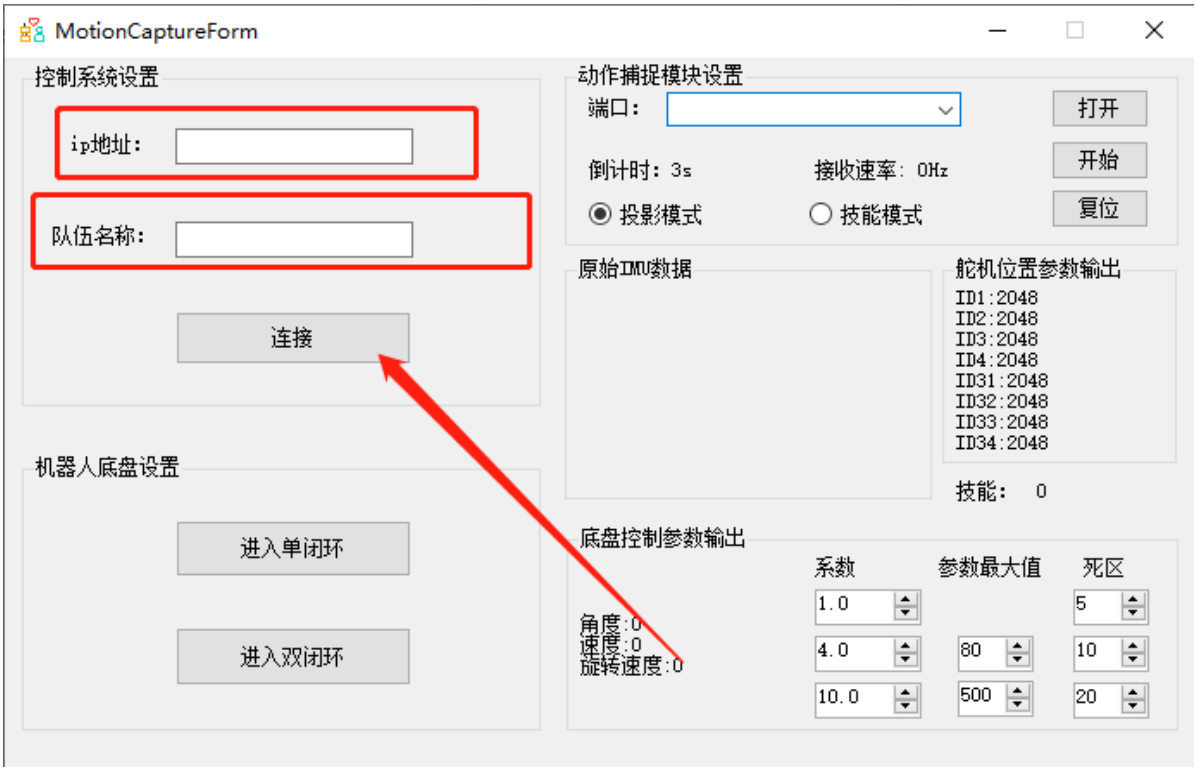
1) 解压并打开“智元素控制系统 1.0.1”文件夹，点击文件夹下的“WisdomElement.exe”即可运行智元素控制系统（***jre 文件夹不能删除，否则会影响使用***）。

2) 双击打开软件，点击“查询 ip 地址”即可查询本机所有的 ip 地址，如下图所示：

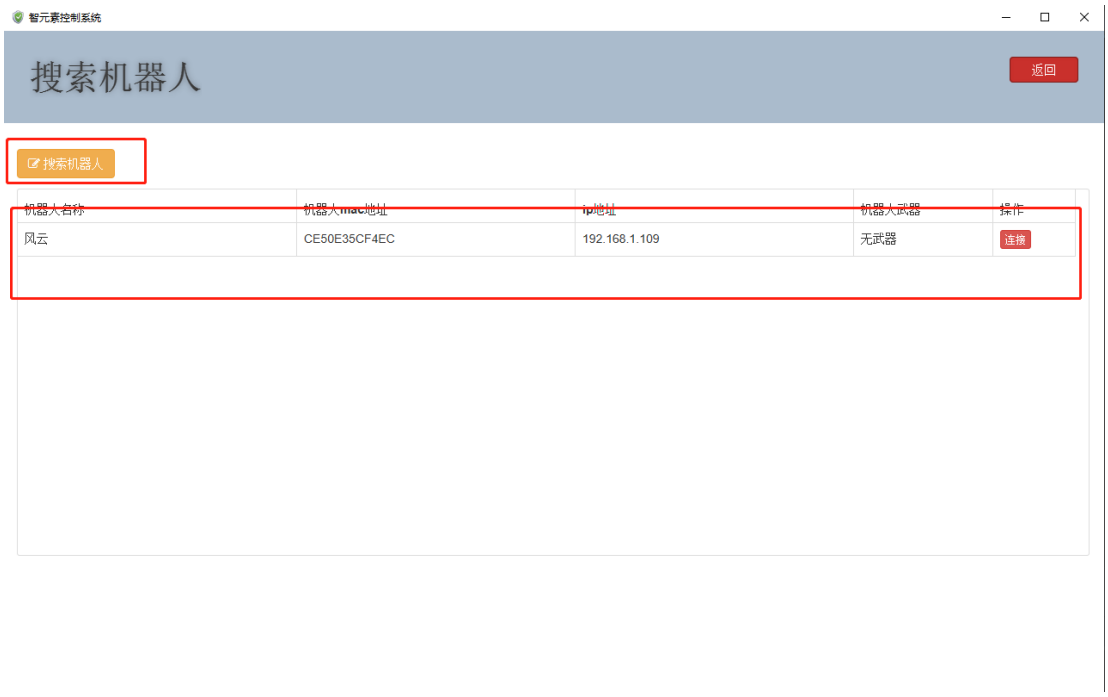
3) 将体感应用 demo 连接到智元素控制系统软件。解压并打开“体感应用 demo3.0”文件夹，依次打开“WindowsFormsAppForRobot”→“ bin”→ “Debug”→“WindowsFormsAppForRobot.exe”。



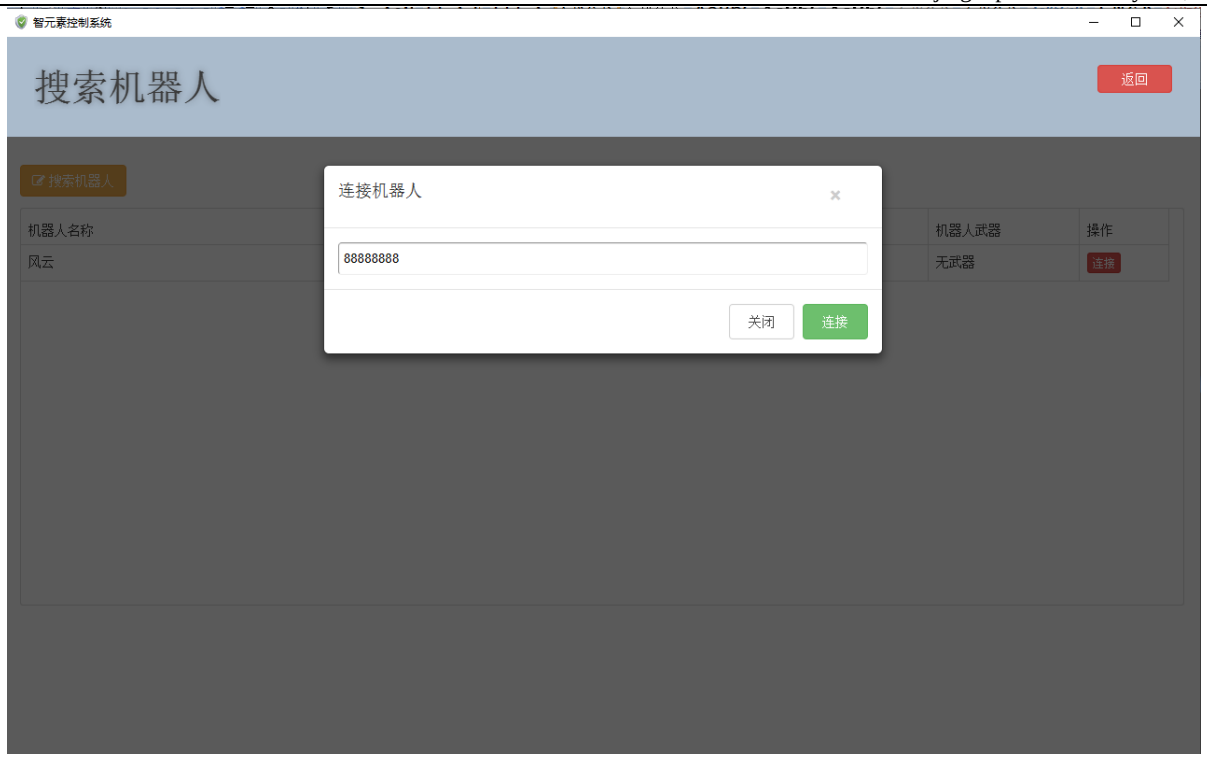
4) 将体感应用软件连接到智元素控制系统。输入 ip 地址和队伍名称之后点击链接按钮即可连接到智元素控制系统。（ip 地址就是上述 b 操作中查看的 ip 地址，***如遇到多个 ip，请依次尝试***。当按钮显示为“断开连接”的时候表示连接成功）

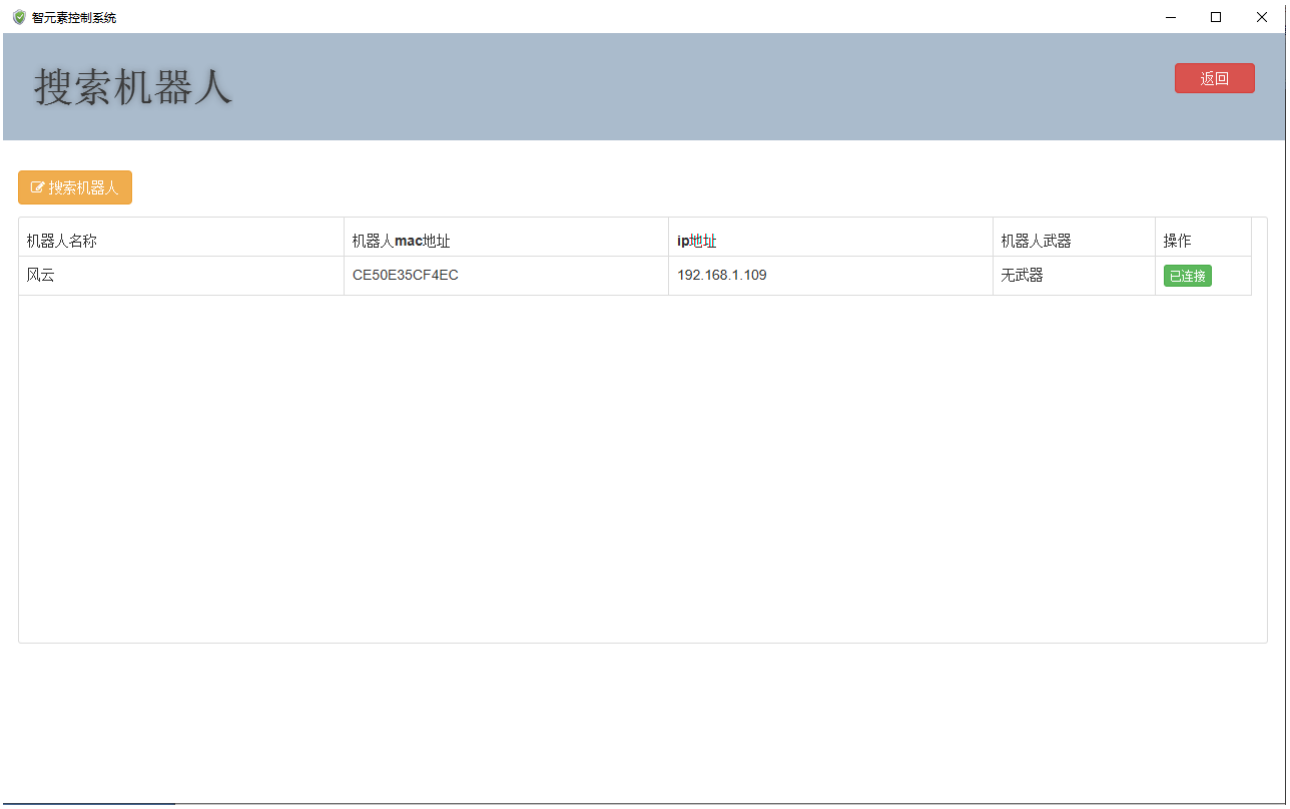


5) 智元素控制系统软件连接机器人。点击智元素控制系统软件中的“机器人列表”，点击“搜索机器人”即可看到当前网络中的机器人。



6) 点击列表后面的“连接”按钮输入密码即可连接到机器人（密码默认 88888888）。

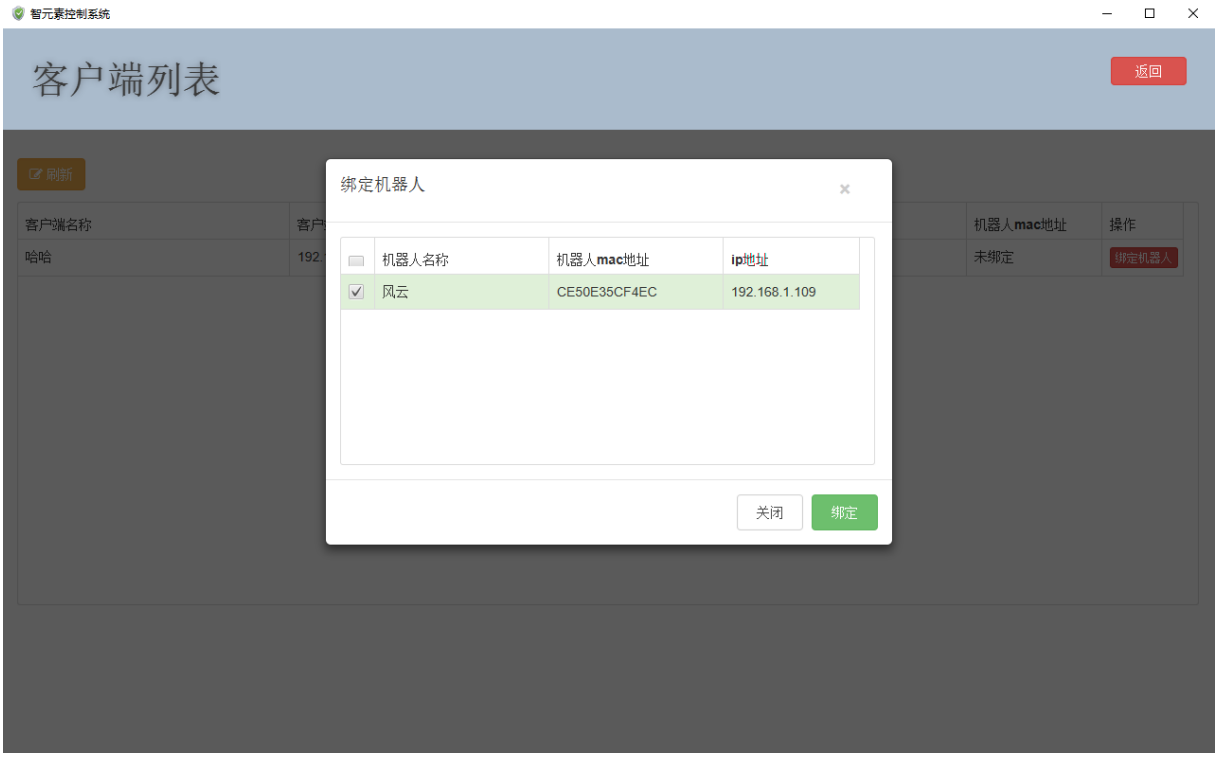




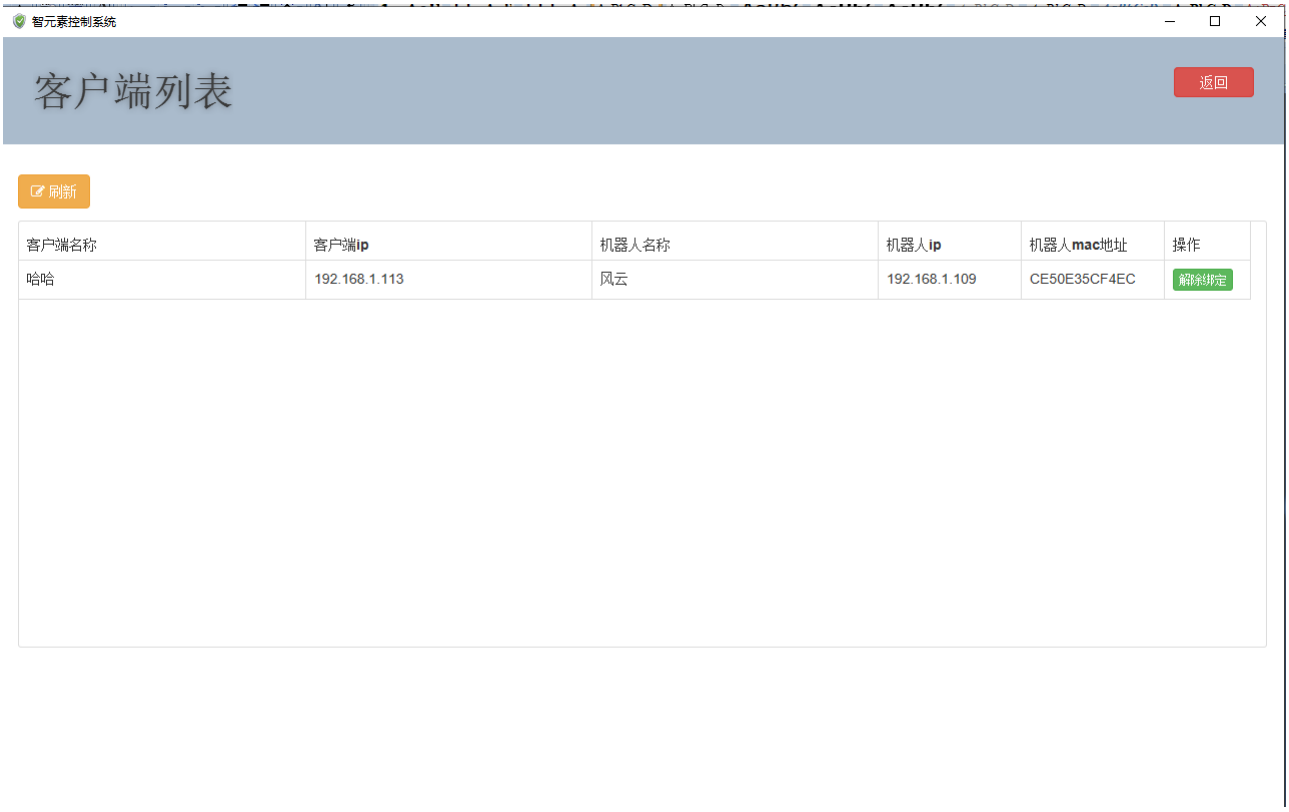
7) 点击右侧“返回”按钮返回到首页，再点击“客户端列表”，即可看到当前连入的体感上位机软件。点击列表后的“绑定机器人”按钮在弹出的机器人列表中选择一个机器人即可绑定该机器人。

智元素控制系统
智元素控制系统
点击查看ip地址
app v1.0.1
机器人列表
客户端列表
单机对战
局域网对战

智元素控制系统
客户端列表
返回
刷新
客户端名称
客户端ip
机器人名称
机器人ip
机器人mac地址
操作
哈哈
192.168.1.113
未绑定
未绑定
未绑定
绑定机器人



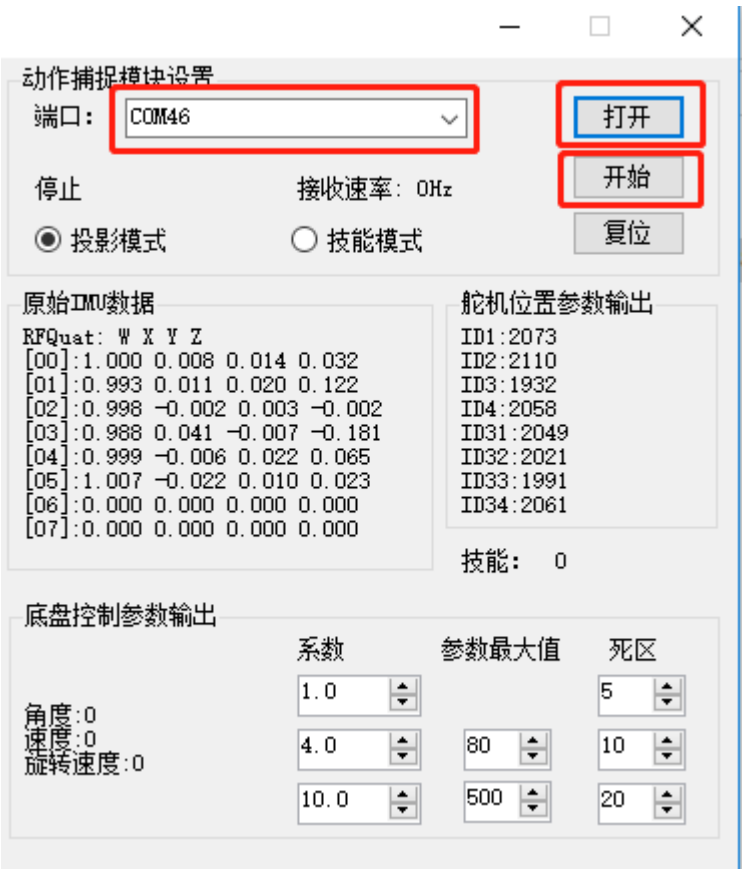
8) 此时绑定成功页面显示解除绑定，点击“解除绑定”即可解除体感上位机软件与机器人的连接。



9) 点击右侧“返回”按钮返回到首页，点击“单机对战”会出现当前机器人的列表，点击列表后的“查看单机对战属性”，即可进入单机对战模式，点击“进入战斗模式按钮”即可进行战斗。

10) 点击右侧“返回”按钮返回到首页，当有 2 个上位机软件连接到系统，并绑定机器人，点击“局域网对战”即可进入局域网战斗。

11) 将无线姿态模块接收机插入到计算机 USB，并将姿态模块节点穿戴在身上。打开“WindowsFormsAppForRobot.exe”软件，选择对应的端口，点击“打开”-“开始”，3s 后即可通过身体来操作机器人。

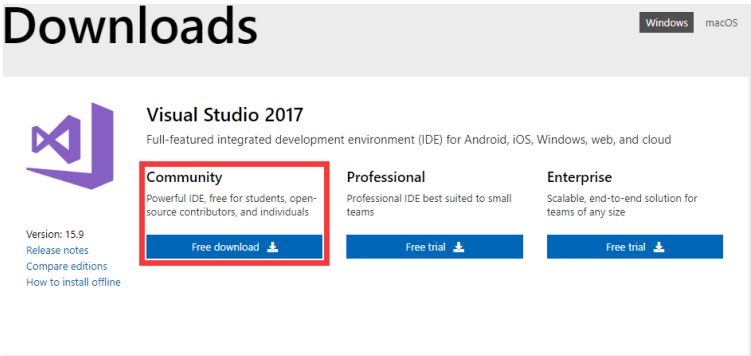


4. 体感应用开源软件介绍

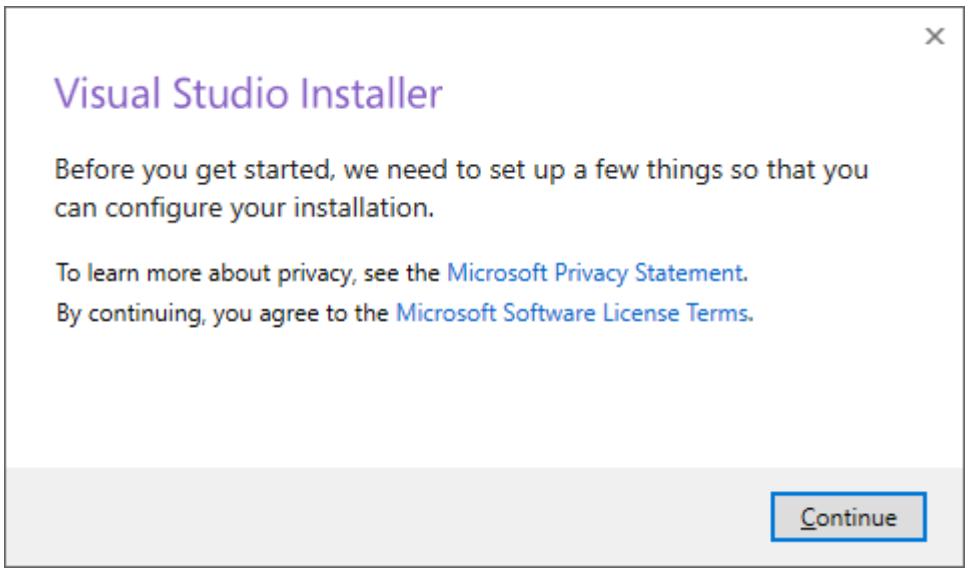
针对体感应用，我司提供了配套的开源软件，用户可以任意修改使用该开源软件。

4.1. 安装 Visual studio 2017

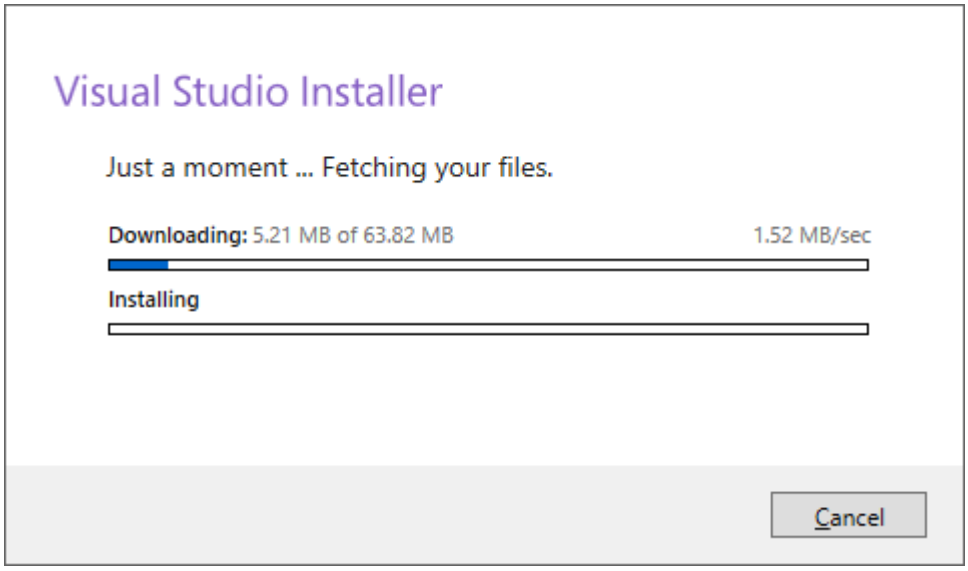
1) 首先到官方网站下载 Visual Studio Community 2017 安装器只有几 M，它只是一个在线安装器。网址：https://visualstudio.microsoft.com/downloads/



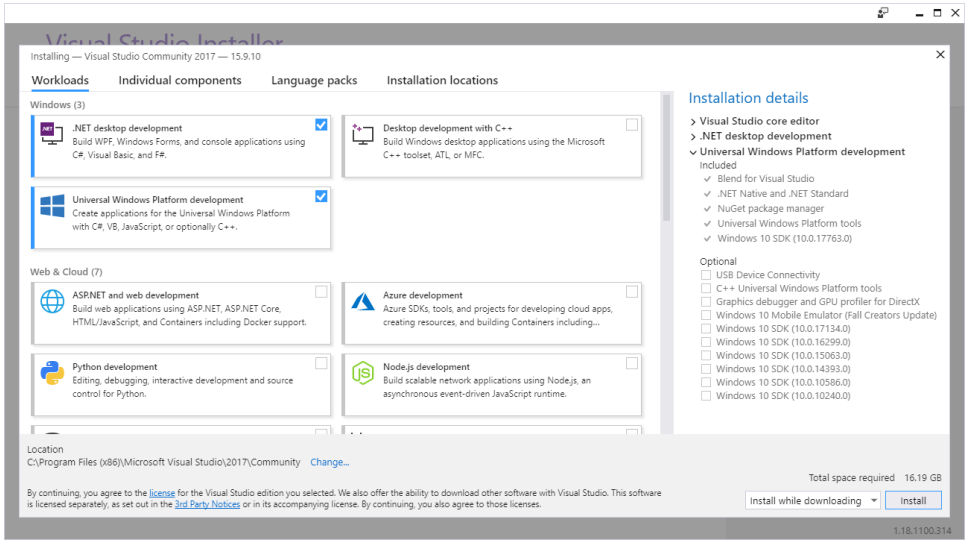
2) 我们运行这个安装器，点击“继续”。



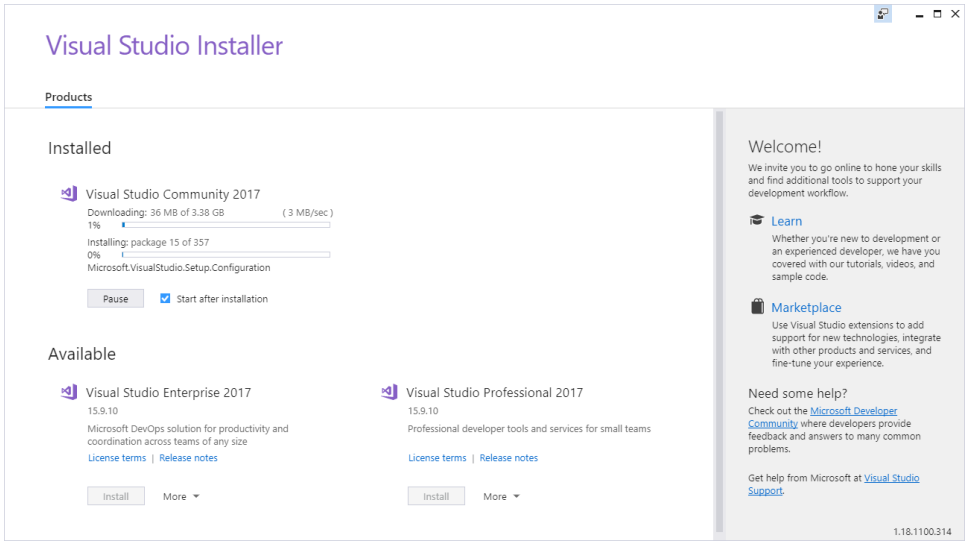
3) 软件会先下载一些前置的文件，用于安装使用的文件体积很小下载完成，并已安装后，会给出提示，并自动进入下一步。

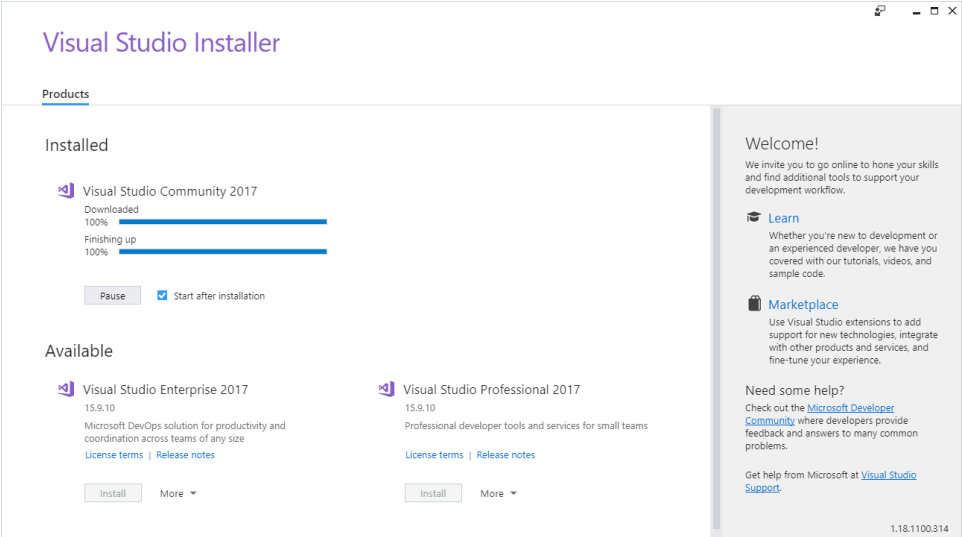


4) 接下来会弹出 visual studio 的安装配置菜单，我们选择如下两项：

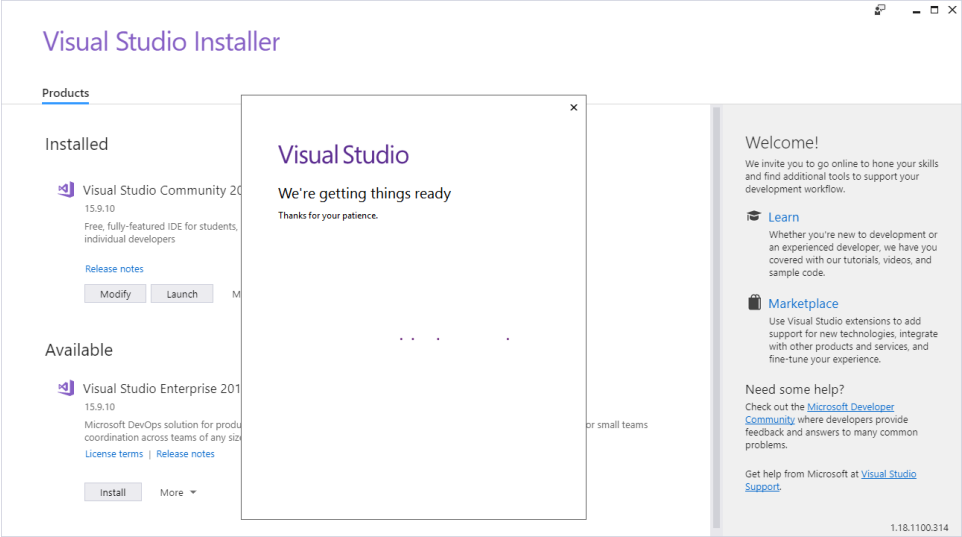


5) 接下来就是耐心等待了。

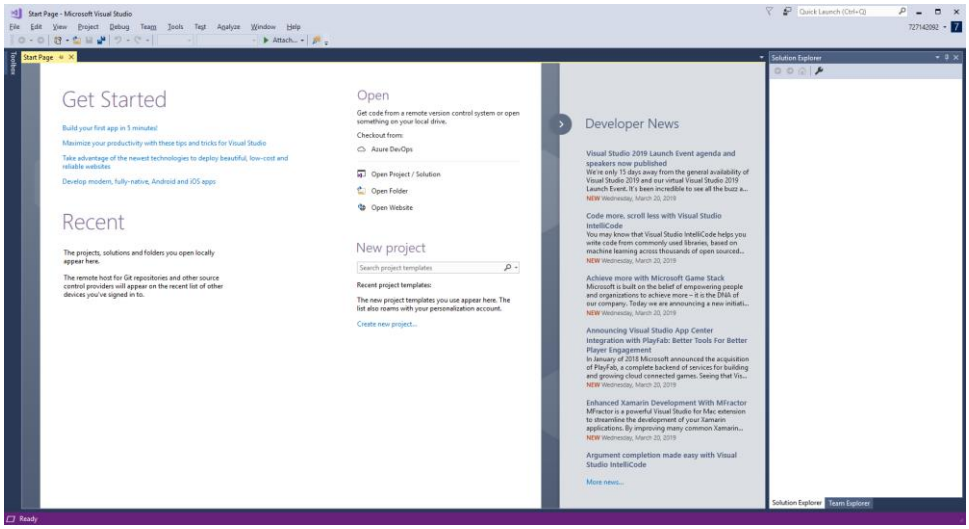




6) 待安装完成软件，软件会自动启动，首次启动时间较慢，耐心等待。

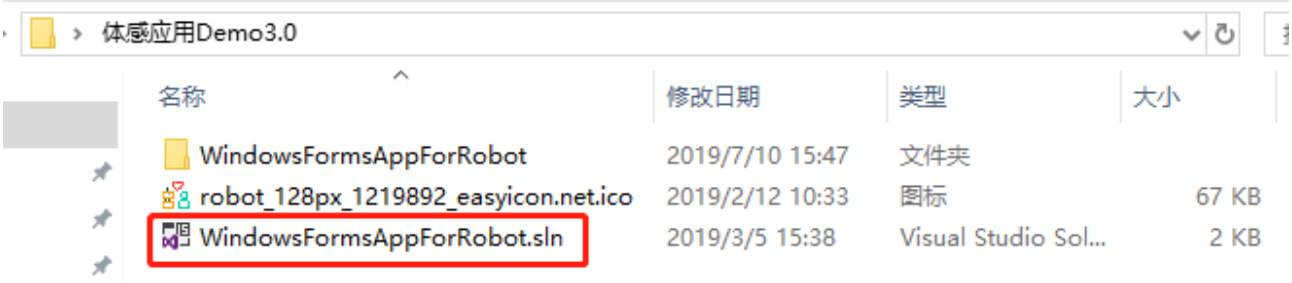


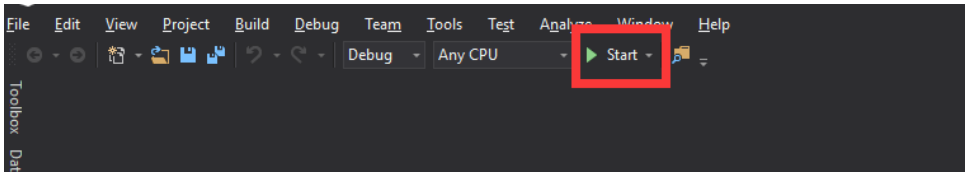
7) 软件可以正式使用了。



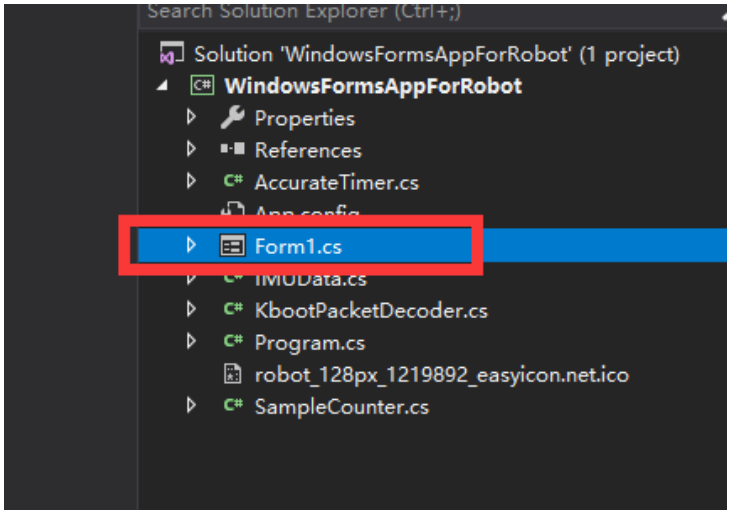
4.2. 运行上位机软件

打开“体感应用 Demo3.0”文件夹，双击“WindowsFormsAppForRobot.sln”即可打开工程文件。点击“Start”即可运行软件。

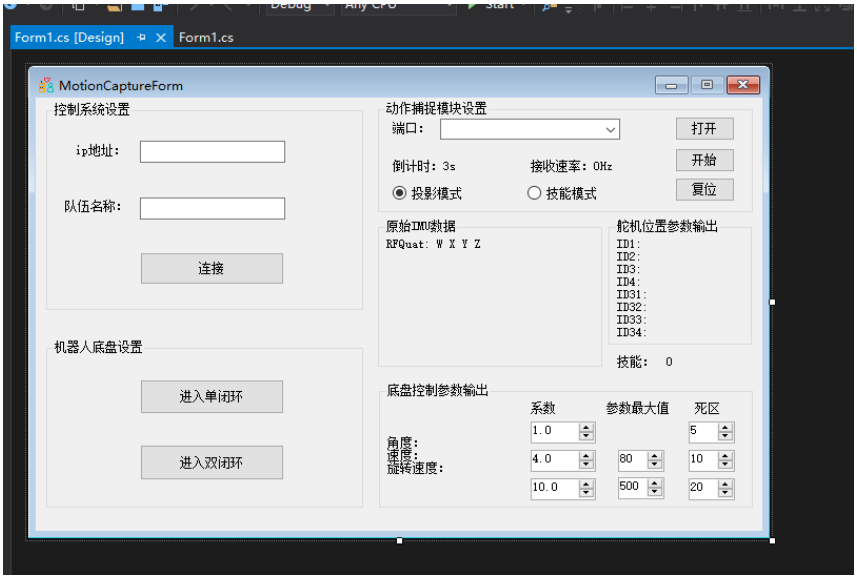




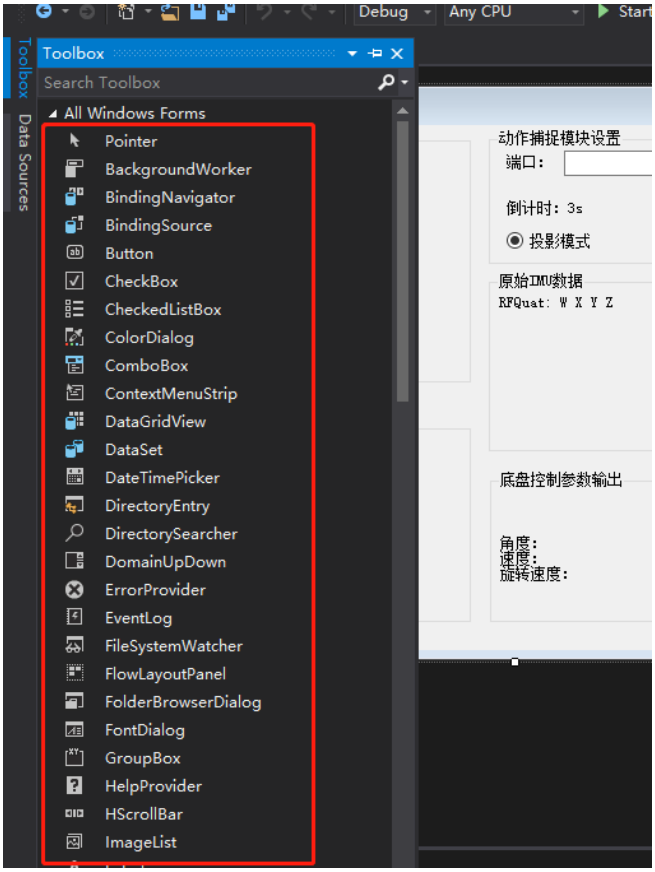
停止运行，在窗口右侧找到“Form1.cs”，单击，并按键盘的 F7 按键，就能查看相关代码。



另外我们也可以双击“Form1.cs”，就可以弹出窗体。窗体上的控件可以任意拖动、改变位置。

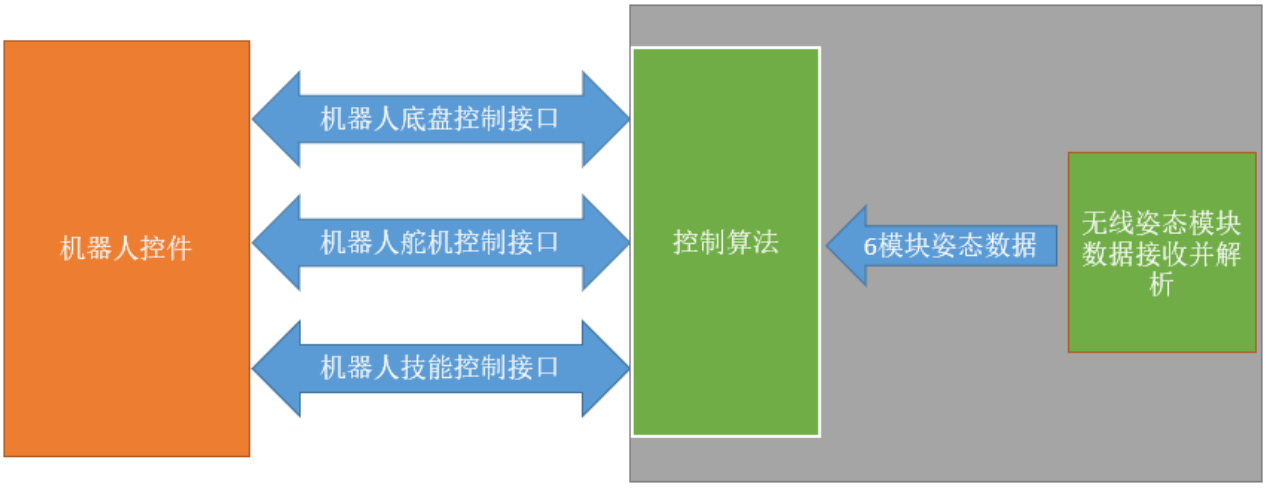


另外在软件的左侧有“工具箱”，我们可以任意添加里面的控件。



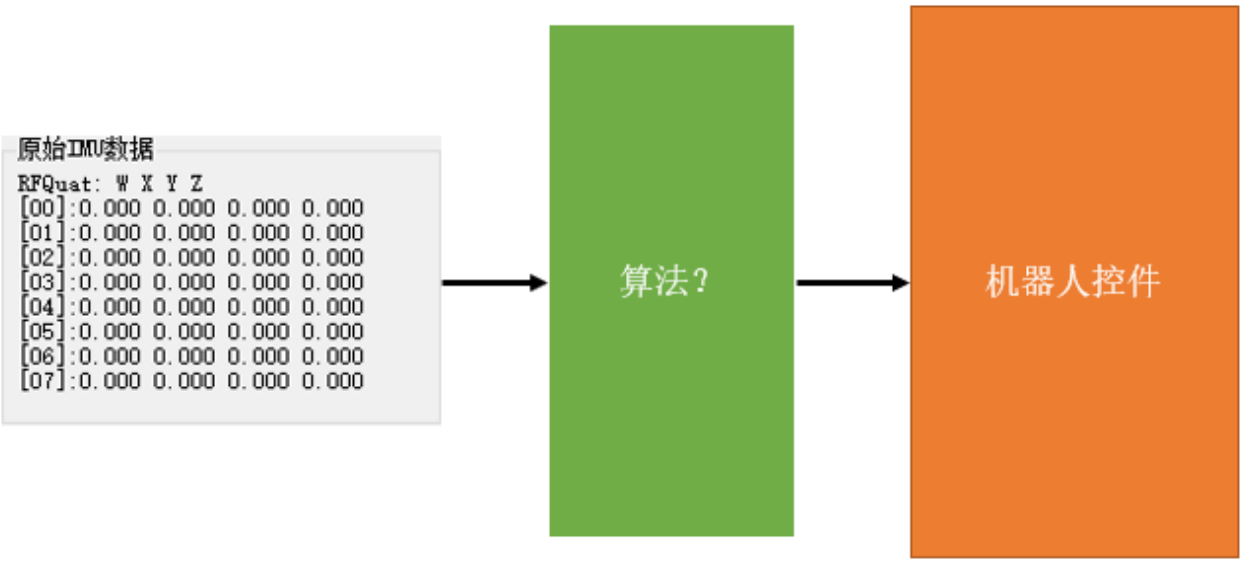
4.3. 上位机软件介绍

程序整体框架如下：首先接收无线姿态模块的数据并解析，然后将 6 模块的姿态数据（四元数）传给机器人控制算法，经过控制算法的解算后就能输出机器人的相关控制数据，最后将相关控制数据传给机器人控件就能达到控制机器人的目的。

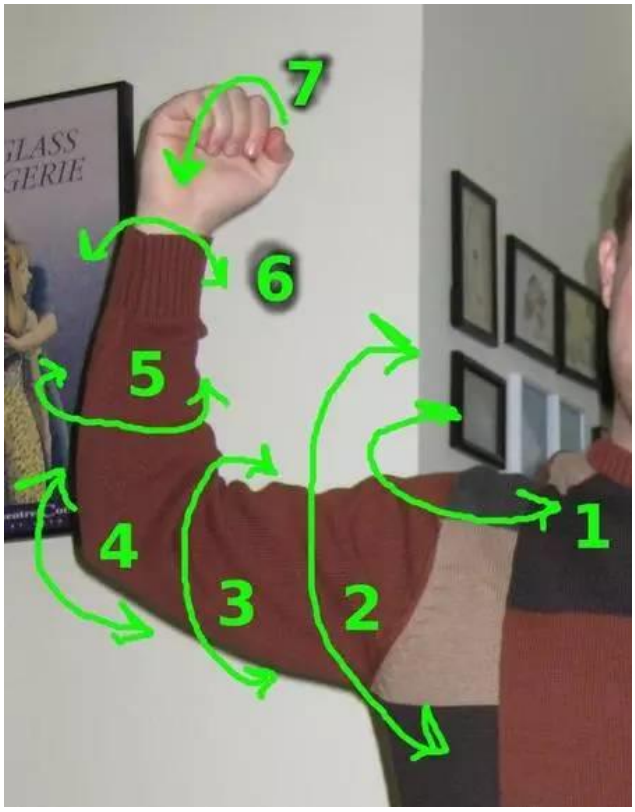


4.3.1. 算法介绍

如下图所示，我们可以轻松的获得姿态模块的原始数据（四元数），但是通过什么样的算法才能获得机器人的控制信息呢？这是需要需要重点关注的，下面将介绍如何实现控制控制机器人的算法。



投影控制算法：投影算法是直接将人的手臂动作投影到机器人身上，首先人的手臂有 7 个自由度，而机器人手臂只有 4 个自由度，因为自由度数量的不同，所以要将图中的 3，4，5 自由度去掉，也就是人做动作的时候肘关节不能动。另外，欧拉角的数值是不能直接使用到舵机参数上的，一方面是舵机的控制是数字量，需要手臂将旋转角度转换为舵机的数字量，另一方面是机器人手臂舵机的旋转是相较于上一个舵机的旋转，而姿态模块的旋转数据是基于当地地理坐标的三维方位数据，他们的坐标系不一致所以不能直接应用。



如果想要通过姿态模块来控制舵机，就得获得相邻模块的相对四元数，在把相对四元数转换为相对欧拉角，就可以利用相对欧拉角来控制舵机了。其中获取相对四元数的代码如下所示：

```
/// <summary>
/// 获取相对四元数
/// </summary>
/// <param name="Qm">Qm</param>
/// <param name="Qn">Qn</param>
/// <returns>相对四元数</returns>
private float[] getQmn(float[] Qm, float[] Qn)
{
    float[] Qmn = new float[4];
    float[] Qm_1 = new float[4];//Qm的逆
    Qm_1[0] = Qm[0];
    Qm_1[1] = -Qm[1];
    Qm_1[2] = -Qm[2];
    Qm_1[3] = -Qm[3];

    Qmn[0] = Qm_1[0] * Qn[0] - Qm_1[1] * Qn[1] - Qm_1[2] * Qn[2] - Qm_1[3] * Qn[3];
    Qmn[1] = Qm_1[0] * Qn[1] + Qm_1[1] * Qn[0] + Qm_1[2] * Qn[3] - Qm_1[3] * Qn[2];
    Qmn[2] = Qm_1[0] * Qn[2] - Qm_1[1] * Qn[3] + Qm_1[2] * Qn[0] + Qm_1[3] * Qn[1];
    Qmn[3] = Qm_1[0] * Qn[3] + Qm_1[1] * Qn[2] - Qm_1[2] * Qn[1] + Qm_1[3] * Qn[0];
    return Qmn;
}
```

总体步骤如下：

1) 在系统初始化阶段， 系统读取右腕关节和右大臂的初始四元数，如下公式：

$$q_1^{'} = q_{10}^{'} + iq_{11}^{'} + jq_{12}^{'} + kq_{13}^{'}$$

$$q_2^{'} = q_{20}^{'} + iq_{21}^{'} + jq_{22}^{'} + kq_{23}^{'}$$

代码如下：

```
for (int i = 0; i < 4; i++)
{
    initQr0[i] = imuData.RFQuat[0, i];
    initQr1[i] = imuData.RFQuat[1, i];
```

2) 计算右腕关节和右大臂的姿态的初始偏置四元数，如下公式：

$$T' = q_{10}'^{2} + q_{11}'^{2} + q_{12}'^{2} + q_{13}'^{2}$$

$$q_1'^{-1} = \frac{q_{10}' - iq_{11}' - jq_{12}' - kq_{13}'}{T'}$$

$$\Delta q' = q_1'^{-1} \times q_2' = \begin{pmatrix} \Delta q_1' \\ \Delta q_2' \\ \Delta q_3' \\ \Delta q_4' \end{pmatrix} = \begin{pmatrix} \dfrac{q_{20}'q_{10}' + q_{21}'q_{11}' + q_{22}'q_{12}' + q_{23}'q_{13}'}{T'} \\ \dfrac{-q_{20}'q_{11}' + q_{21}'q_{10}' + q_{22}'q_{13}' - q_{23}'q_{12}'}{T'} \\ \dfrac{-q_{20}'q_{12}' - q_{21}'q_{13}' + q_{22}'q_{10}' + q_{23}'q_{11}'}{T'} \\ \dfrac{-q_{20}'q_{13}' + q_{21}'q_{12}' - q_{22}'q_{11}' + q_{23}'q_{10}'}{T'} \end{pmatrix}$$

代码如下：

```
initQr01 = getQmn(initQr0, initQr1);
```

3) 在系统工作阶段， 系统读取右腕关节和右大臂的实时四元数，如下公式：

$$q_1 = q_{10} + iq_{11} + jq_{12} + kq_{13}$$

$$q_2 = q_{20} + iq_{21} + jq_{22} + kq_{23}$$

代码如下：

```
for (int i = 0; i < 4; i++)
{
    currentQr0[i] = imuData.RFQuat[0, i];
    currentQr1[i] = imuData.RFQuat[1, i];
```

4) 计算右腕关节和右大臂的姿态的实时绝对偏置四元数，如下公式：

$$T = q_{10}^{\ 2} + q_{11}^{\ 2} + q_{12}^{\ 2} + q_{13}^{\ 2}$$

$$q_1^{\ -1} = \frac{q_{10} - iq_{11} - jq_{12} - kq_{13}}{T}$$

$$\Delta q = q_1^{\ -1} \times q_2 = \begin{pmatrix} \Delta q_0 \\ \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \end{pmatrix} = \begin{pmatrix} \dfrac{q_{20}q_{10} + q_{21}q_{11} + q_{22}q_{12} + q_{23}q_{13}}{T} \\ \dfrac{-q_{20}q_{11} + q_{21}q_{10} + q_{22}q_{13} - q_{23}q_{12}}{T} \\ \dfrac{-q_{20}q_{12} - q_{21}q_{13} + q_{22}q_{10} + q_{23}q_{11}}{T} \\ \dfrac{-q_{20}q_{13} + q_{21}q_{12} - q_{22}q_{11} + q_{23}q_{10}}{T} \end{pmatrix}$$

代码如下：

```
currentQr01 = getQmn(currentQr0, currentQr1);
```

5) 计算右腕关节和右大臂的姿态的实时相对偏置四元数，如下公式：

$$K = \Delta q_0^{\ 2} + \Delta q_1^{\ 2} + \Delta q_2^{\ 2} + \Delta q_3^{\ 2}$$

$$\Delta q^{-1} = \frac{\Delta q_0 - i\Delta q_1 - j\Delta q_2 - k\Delta q_3}{K}$$

$$q = \Delta q^{-1} \times \Delta q' = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \dfrac{\Delta q_0' \Delta q_0 + \Delta q_1' \Delta q_1 + \Delta q_2' \Delta q_2 + \Delta q_3' \Delta q_3}{K} \\ \dfrac{-\Delta q_0' \Delta q_1 + \Delta q_1' \Delta q_0 + \Delta q_2' \Delta q_3 - \Delta q_3' \Delta q_2}{K} \\ \dfrac{-\Delta q_0' \Delta q_2 - \Delta q_1' \Delta q_3 + \Delta q_2' \Delta q_0 + \Delta q_3' \Delta q_1}{K} \\ \dfrac{-\Delta q_0' \Delta q_3 + \Delta q_1' \Delta q_2 - \Delta q_2' \Delta q_1 + \Delta q_3' \Delta q_0}{K} \end{pmatrix}$$

代码如下：

```
Qr01 = getQmn(currentQr01, initQr01);
```

6) 计算右腕关节在三轴方向上的相对运动角度 ，如下公式：

$$\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix}=\begin{bmatrix}atan2(2(q_0q_1+q_2q_3),1-2(q_1^2+q_2^2))\\ \arcsin(2(q_0q_2-q_1q_3))\\ atan2(2(q_0q_3+q_1q_2),1-2(q_2^2+q_3^2))\end{bmatrix}$$

代码如下：

```
correctedEulerAngle_r01 = quaternionToEulerAngle(Qr01);
```

7) 计算右腕关节两个舵机数字量值，代码如下：

```
servo32Position = (UInt16)(correctedEulerAngle_r01[0] * 4096 / 250 * (-1) + 2048);
servo31Position = (UInt16)(correctedEulerAngle_r01[1] * 4096 / 250 * (-1) + 2048);
```

8) 最后对舵机位置进行限位，如果不限位，舵机会过载，从而影响舵机寿命，代码如下：

```
servo31Position = GetServoPositonValue(servo31PositionMin, servo31PositionMax, servo31Position);
```

技能控制算法：该算法是延续的上面的投影控制算法，通过判断得到的舵机位置参数信息是否在某个范围内来控制释放技能的目的。

通过如下步骤即可达到让机器人释放技能的目的：

1) 记录下你的手臂当前的位置。如下图所示：

2) 将记录下来的参数填入到下面的函数当中，其中 servo31Position、servo32Position、servo33Position、servo34Position 为舵机位置参数，500 为范围参数，这是前面参数的误差值。当手臂再次达到一开始的位置就能触发技能。官方提供的代码里面只设置了两个技能，如果需要添加其他的技能，请自行添加相关代码。

```
motion1 = getSkillNum(servo31Position, servo32Position, servo33Position, servo34Position, 2050, 2046, 883, 2049, 500);
```

4.3.2. 底盘控制参数介绍

机器人底盘有 3 个控制参数，分别为角度、速度、和旋转速度。其中角度是用来控制机器人朝着那个方向移动，速度用来设置机器人的行驶速度，旋转速度是用来设置机器人原地旋转的速度，我们穿戴在脚上的传感器就是用来产生这三个控制参数的数值。当我们的脚前倾或者后倾就能产生角度（0°代表向前移动，180°代表向后移动）和速度的控制参数，当我们的脚向左边或者右边摆动就能产生旋转速度控制参数。

另外在软件上我们可以看到“角度”、“速度”、“旋转速度”，这些就是我们实际输出给底盘的控制参数，而“系数”、“参数最大值”、“死区”这些参数是用来修正“角度”，“速度”，和“旋转速度”的，这些参数都是可调的，目的是为了达到更好的控制体验。我们以下图的“旋转速度”为例，它的“系数”是10，代表的是单位角度下的增长量，给我们的控制感受就是数值越高动作越激进。“参数最大值”就是旋转速度所能达到的最大值，在这里我们设置的是 500，如果感觉机器人行驶速度太慢，这个数值可以调高。“死区”的数值是 20，代表的是 0-20 范围内输出的旋转速度都是 0，给我们的控制感受就是数值太小不易让机器人“站住不动”，太大的话影响控制行程。

5.注意

1) 比赛过程中官方会提供笔记本与无线模块，用户只需把“体感应用 demo3.0”文件夹中的“Debug”文件拷贝到比赛计算机就可以进行比赛。
2) 机器人主控板需要使用最新的固件，请使用 APP 查询是否是最新的固件。