

Aelos Smart 高校人工智能大赛培训

目录：

- 一、机器人简介
- 二、Aelos_edu 桌面软件
- 三、网络配置及远程连接
- 四、执行动作
- 五、代码下载
- 六、运行 ROS 节点
- 七、比赛说明
- 八、HSV 颜色获取、修改及踢球程序验证步骤（参照视频）
- 九、比赛程序运行
- 十、常见 FAQ

一、机器人简介

- 1、Aelos Smart 是一款小型双足仿人形机器人。
- 2、尺寸 346mm*224mm*118mm，重量 1.73KG。
- 3、支持两种步态算法，慢走 5 厘米/秒，快走 15 厘米/秒。
- 4、控制器采用高性能 STM32 核心，板载储存空间 128M。
- 5、开发平台为 Raspberry Pi CM4，支持搭载摄像头。
- 6、编程平台：兼容 PC 端 Aelos edu (for mac & PC)，Linux，支持 ROS 和 Python 编程。

7、自由度：共 17 个自由度，头部 1 个关节，肩部 1 个关节（共两只），手臂 2 个关节（共两只），腿部 4 个关节（共两只），脚步 1 个关节（共两只）。

8、舵机：17 个强扭矩伺服舵机；运动范围：180°；精度：1°。

9、电池：7.4V，容量 3200mAH。

10、传感器：内置 3 个传感器，包含地磁传感器、头部摄像头和胸部摄像头，机体前胸自带 2 个磁吸传感器扩展口，2 个传感器扩展口均可实现传感器数据模拟输入和执行器数据输出；

11、配套传感器：机器人功能拓展，10 个外置传感器相互配合完成不同的场景任务。

输入模块

火焰传感器：识别火焰。

光敏传感器：识别环境光源强度。

温度传感器：探测环境温度。

湿度传感器：探测环境湿度。

气敏传感器：检测特定气体。

触摸传感器：感应人体触摸。

人体红外传感器：感应人的远近。

碰撞开关：感应碰撞。

输出模块

LED 灯：可实现常亮、闪烁等多种编程。

风扇：可实现编程控制转动。

12、摄像头：镜头 60 度，500 万像素

摄像头型号：USB+OV5640-160D

教育版本		
Aelos lite机器人PC端教育版使用说明书 Aelos lite机器人PC端教育版使用说明书	01 DEC 2019	下载支持
Aelos Pro机器人PC端教育版使用说明书 Aelos Pro机器人PC端教育版使用说明书	01 DEC 2019	下载支持
Aelos 机器人PC端教育版安装程序 Windows 1.10.0版本 (Windows OS)	01 October 2019	下载支持
Aelos 机器人PC端教育版安装程序 MAC 1.10.0版本 (Mac OS)	01 JUNE 2019	下载支持
Aelos手柄使用指导 Aelos手柄使用指导	01 JUNE 2019	下载支持

芯片：HT67F489

注意：在使用时不要用提机器人的头部或用力拉扯舵机。

机器人可以边充电边使用。

二、Aelos_edu 桌面软件

1、软件下载与安装

(1) 浏览器地址栏中输入乐聚官方网站的地址

www.lejurobot.cn

(2) 在网站的导航栏中，点击“服务与支持”>“下载支持”>

“Aelos 机器人 PC 端教育版安装程序（注意选择相和电脑系统对应的系统版本 Windows 或 MAC）”>“软件下载”即可进行下载。

注意：电脑最好 Windows10 或 Windows11 系统，安装时选择仅为我安装，安装过程中关闭杀毒软件、防火墙等。安装成功后电脑联网，将软件更新到最新版本。

2、软件界面简介

Aelos 桌面软件可分为几个部分，分别是菜单栏、指令栏、



编辑区、动作视图、机值视图区。



- 菜单栏提供多种类的软件功能，可与机器人进行联动使用，如“动作下载”、“U 盘模式”等。
- 指令栏中包括控制指令和动作指令两种指令，都是程序员已经编写封装好的内容，可以供同学们选择使用，方便同学们更快捷地进行程序编写。
- 编辑区是编写程序的主要阵地，指令的添加、删除，程序的整体设计都在编辑区中进行，我们可以在这里直观地看到当前程序的整体情况
- 动作视图可以显示每个动作的详细信息，例如各舵机角度值、速度、刚度以及搭配的音乐等。这些信息以条状记录进行显示，可以显示单一动作，也可以显示一个动作指令里的一组动作。动作视图中也可以对所显示的动作进行预览、修改、

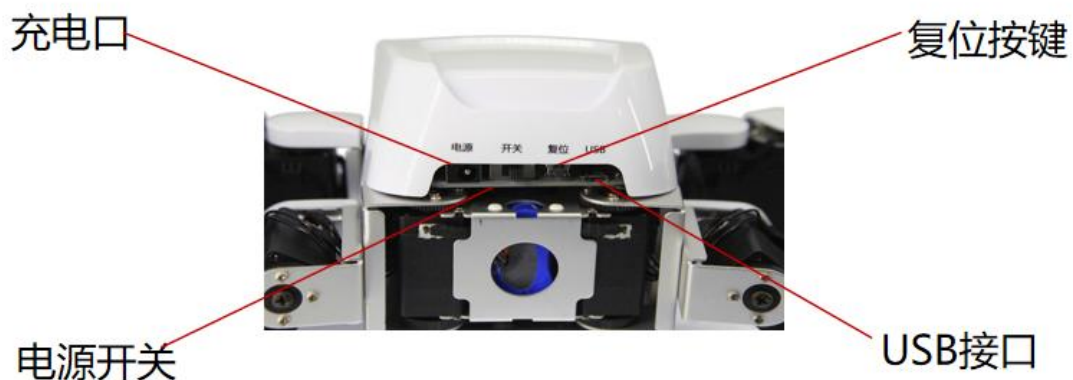
删除或者将整组动作打包成一个新的模块。

- 机值视图区是显示当前机器人身上各个舵机的旋转数值的区域。在机值视图中我们可以看到机器人身体的各个关节处都标有舵机的编号，每个标号下方所显示的就是该舵机的数值。我们可以在机值视图对这些舵机值进行调整，直到 Aelos 达到我们所预想的动作状态。

3、连接串口

打开 Aelos_edu 软件，新建或打开一个 Aelos Pro 的工程，机器人处于开机状态，用数据线连接好机器人与计算机，在 Aelos_edu 软件中连接串口。





4、编辑动作

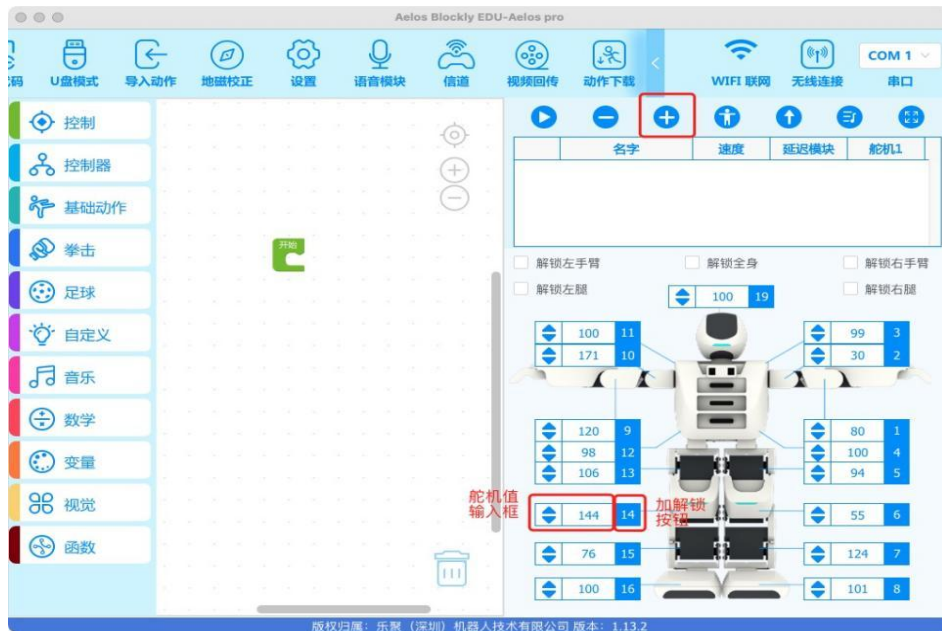
编辑动作分为两部分：调整单个动作帧里面的舵机值，以及将多个动作帧串联成完整的动作。



修改舵机值的方式有两种：

- 点击舵机值视图区中舵机值输入框边上的蓝色按钮可以解锁对应舵机，手动将舵机扭到想要的位置，然后再点击按钮加锁舵机即可。

- 直接在输入框中输入想要调到的舵机值，或者通过点击箭头进行修改，调整完毕后点击增加动作按钮即可将动作帧插入列表。



列表中可以为动作设置下列属性：

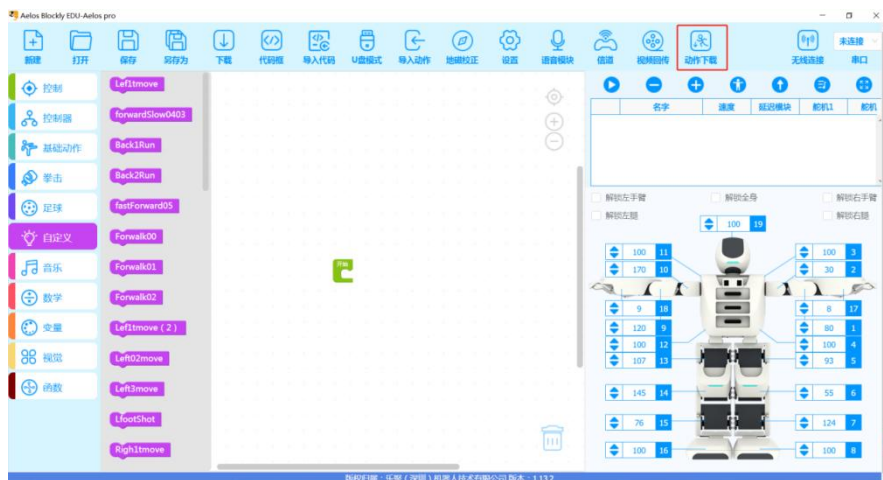
- 速度：机器人执行该动作帧时的运动速度，越大则运动越快。
- 延迟：动作帧间的停顿时间。
- 刚度：改变舵机的柔韧性。例如当执行动作时发现舵机有点软无法很好的支撑，则可以适当加大刚度。

所有动作帧都调整好后点击生成按钮可以生成对应的动作模块，如下图所示：



5、动作下载

该功能会将当前工程内所有的内置动作、自定义动作、音乐及该工程下编辑的动作下载到机器人内。



注意：动作下载后，需重启机器人，才进行 SSH 连接。

三、网络配置及远程连接

镜像基于 Ubuntu MATE 20 系统制作

程序基于 ROS 和 Python3 实现，镜像中已经内置基本环境

1、下载、安装 ssh 连接软件

MAC 用户可以使用系统自带的“终端”软件。

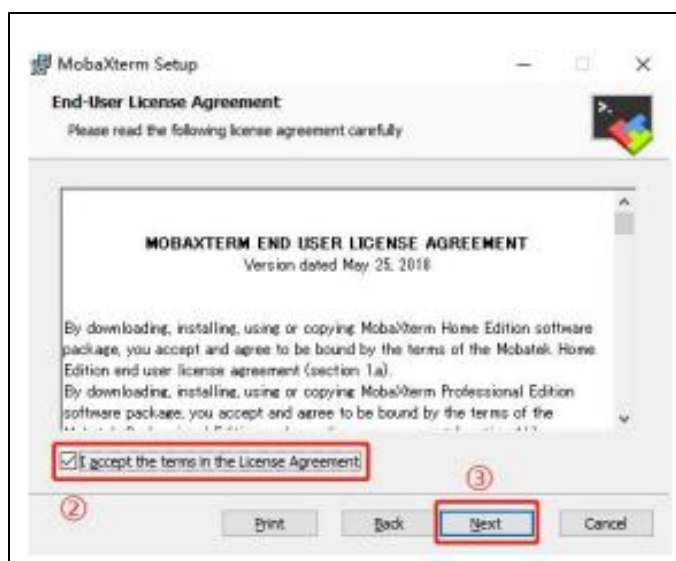
Windows 用户可以下载 MobaXterm，用该软件进行连接。

MobaXterm 是一款专业的仿真终端软件，支持 SSH、Telnet、VNC、RDP、Serial、XDMCP 等协议，同时还集成了网络抓包、端口扫描等运维实用工具。使用此软件可以通过 SSH 协议链接到树莓派进行终端操作。

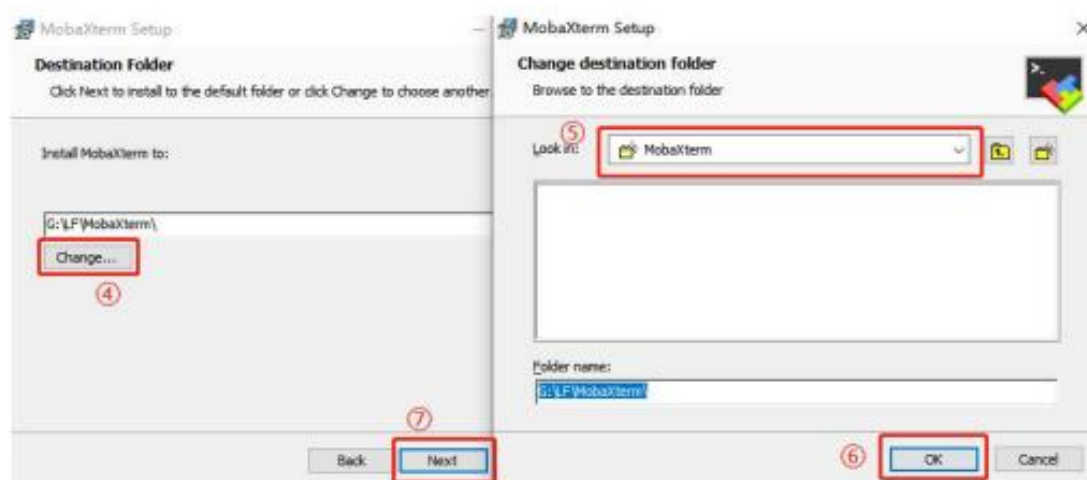
MobaXterm 下载地址：

<https://mobaxterm.mobatek.net/download-home-edition.html>

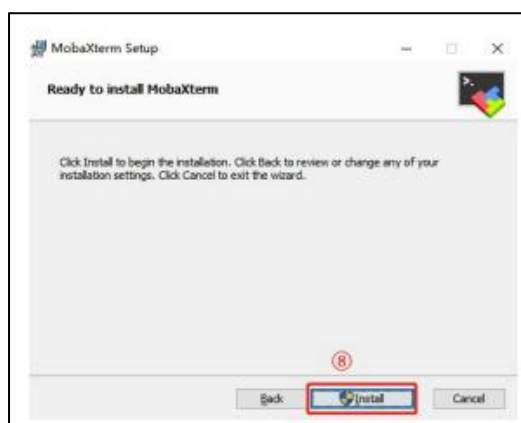
- 下载两个版本都可以，打开下载完毕的安装包：



- 选择安装路径：



- 点击“安装”：

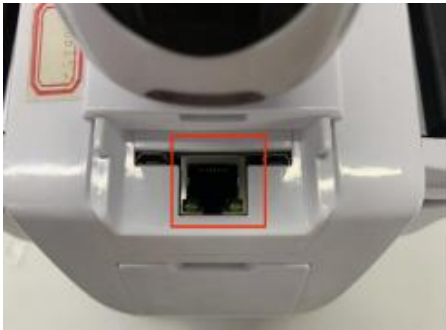


- 安装完成：



2、远程连接步骤

(1) 启动机器人，使用网线连接树莓派网口。将此网线与电脑连接至同一个网络。



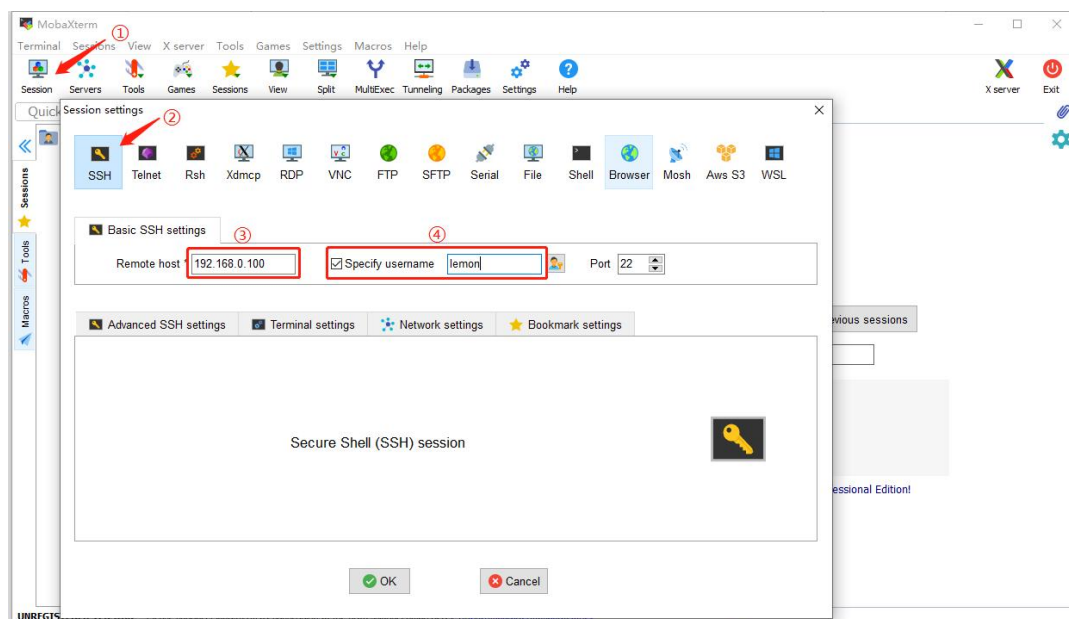
注意：

- 1、在网络配置过程中，应全程保持机器人开机状态。为避免机器人断电关机，请提前为机器人充满电。
- 2、请使用一个路由器，同时连接电脑及机器人。
- 3、树莓派网口位于机器人下巴下方，用力下压滑盖同时向外抽出即可打开。

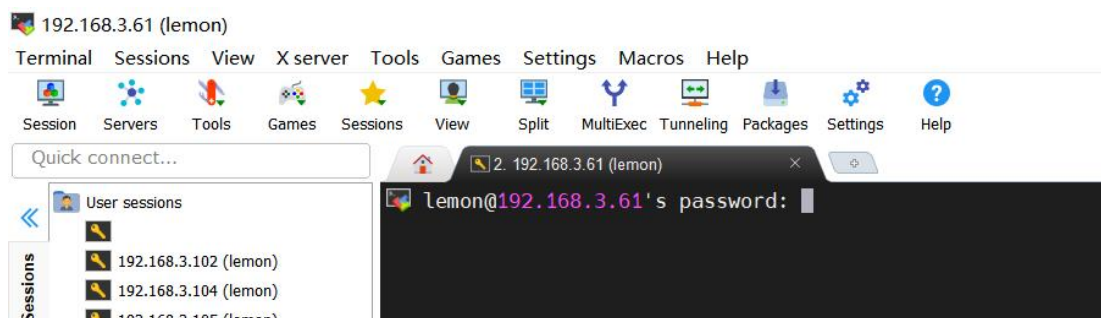
(2) 使用 PC 登录路由器后台管理页面，找到名称为 lemon 的设备，该设备就是机器人。请记住机器人 IP 地址，之后进行 ssh 连接需要使用。



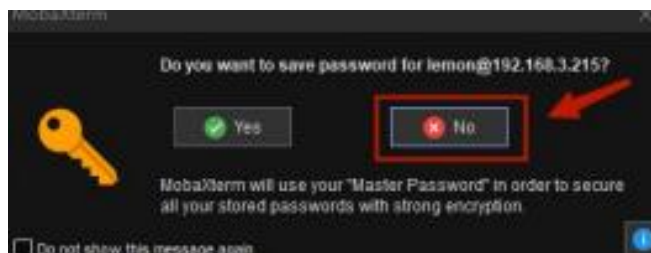
(3) 通打开之前下载好的 MobaXterm，以 lemon 为账号，通过上一步查看到的 IP 地址连接到机器人的终端。



(4) 输入密码：**leju123**。（注意：输入过程中窗口不会显示密码，请仔细输入）



(5) 密码输入成功后，弹出是否储存密码的窗口，推荐选择“**No**”，之后每次登录的时候需输入密码：**leju123**。若想记住密码，可选择“**Yes**”，再按流程操作。



(6) 在终端中输入 `nmcli device wifi list`，来查看机器人可以识别到的 WIFI 信号。在终端中输入 `sudo nmcli dev wifi connect 无线网络名字 password 密码`，来连接到指定的 wifi。

```
lemon@lemon:~$ nmcli device wifi list
IN-USE BSSID SSID MODE CHAN RATE SIGNAL BARS SECURITY
FC:94:CE:7A:10:E0 ChinaNet-aAe4 Infra 4 130 Mbit/s 100 WPA1 WPA2
18:4F:32:07:33:C6 HP-Print-C6-LaserJet Pro MFP Infra 6 65 Mbit/s 87 WPA1 WPA2
78:44:FD:F1:FF:1B jinsejingtian Infra 1 270 Mbit/s 69 WPA1 WPA2
24:58:6E:71:30:80 ChinaNet-piKw Infra 11 130 Mbit/s 60 WPA1 WPA2
9C:A6:15:99:31:45 LeJuRobot Infra 1 270 Mbit/s 44 WPA1 WPA2
FA:D0:27:56:18:5C DIRECT-2756985C Infra 1 65 Mbit/s 39 WPA2
F8:9A:78:FD:F0:E0 Hebfgs1 Infra 11 270 Mbit/s 35 WPA2
F8:9A:78:FD:F0:E5 -- Infra 11 270 Mbit/s 34 WPA2
C8:E7:D8:04:81:52 MERCURY_8152 Infra 6 135 Mbit/s 25 WPA1 WPA2
78:44:FD:FF:2B:43 金色景田 Infra 11 270 Mbit/s 20 WPA1 WPA2
lines 1-11/11 (END)
```

```
lemon@lemon:~$ sudo nmcli dev wifi connect LeJuRobot password leju15000000
[sudo] password for lemon:
Device 'wlan0' successfully activated with '4635f922-ed53-4601-a6dc-b0efe8a1bdbd'.
```

(7) 连接成功后，在终端输入 `ifconfig`，请记住此地址，远程登陆时，需要使用此 IP 地址。

```
lemon@lemon:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.100 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::328e:103b:891b:4873 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:62:ca:be txqueuelen 1000 (Ethernet)
    RX packets 431 bytes 49970 (49.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 350 bytes 47201 (47.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8347 bytes 25791762 (25.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8347 bytes 25791762 (25.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

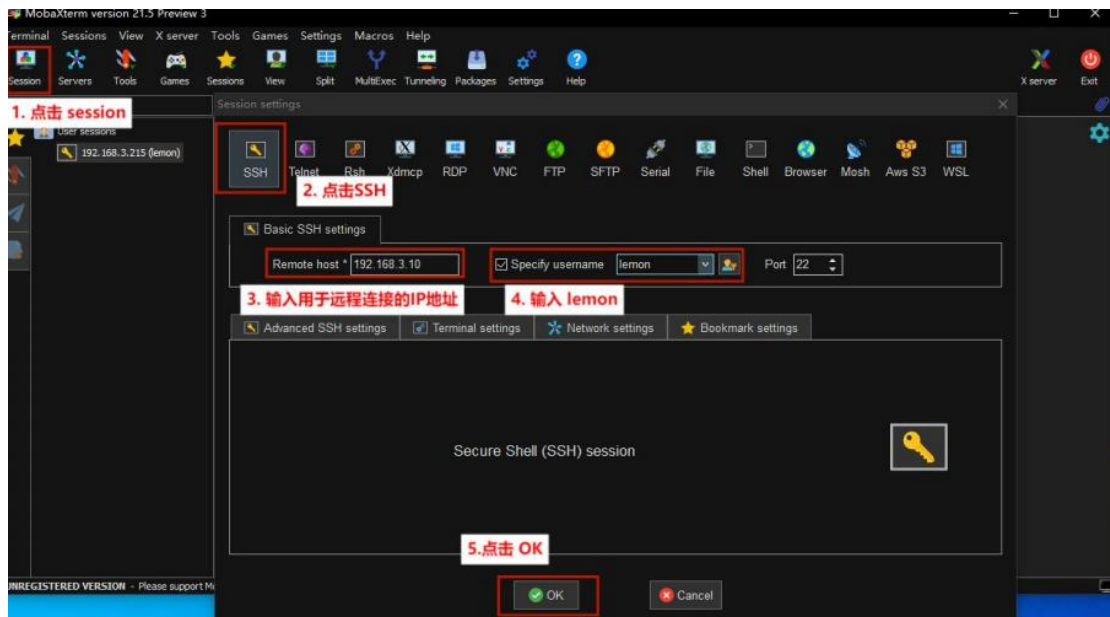
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.122 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::e507:917:b2aa:7da7 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:62:ca:c0 txqueuelen 1000 (Ethernet)
    RX packets 81261 bytes 103641341 (103.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29913 bytes 4837322 (4.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lemon@lemon:~$
```

远程连接的IP地址

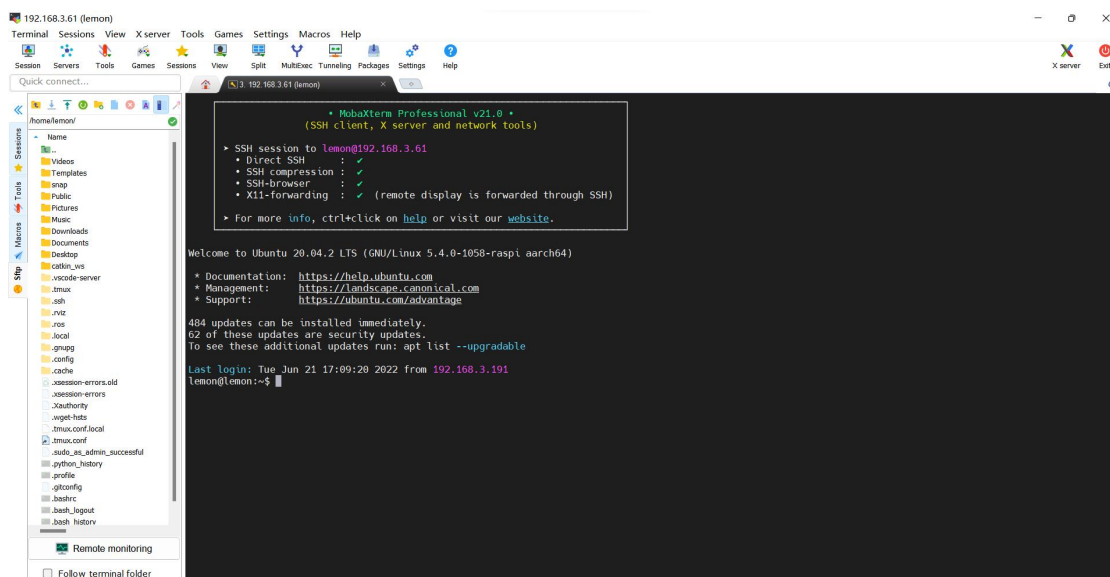
注：记住这个远程连接 IP 地址后，就可以拔掉机器人连接的网线。之后可以使用这个 IP 地址来远程连接机器人。

(8) 按照下图步骤，点击菜单栏“session”，重新建立一个用于远程连接的 session。

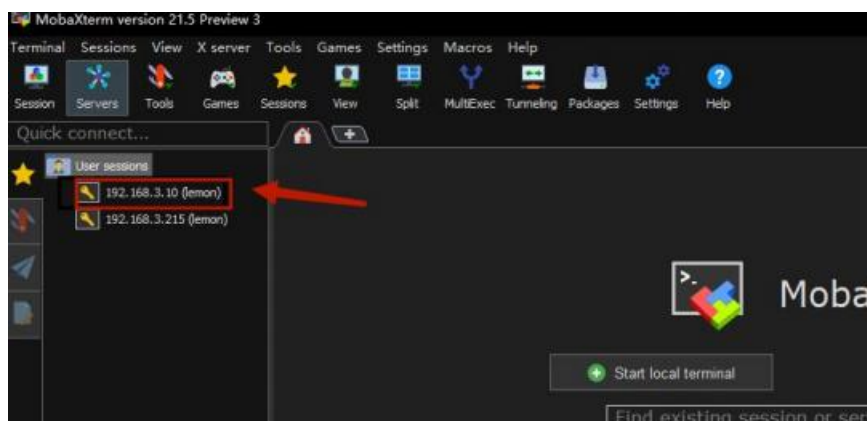


连接成功后，输入密码：**leju123**。密码输入成功后，弹出是否储存密码的窗口，推荐选择“No”，之后每次登录的时候需输入密码：**leju123**。若想记住密码，可选择“Yes”，再按流程操作。

■ 输入密码，成功登录。至此，为机器人作网络配置的任务就完成了。



之后想远程连接机器人时，只需要在机器人开机的状态下，打开“MobaXterm”软件，双击用于远程连接的 session 就可以了。



（备注：若忘记 IP 地址，可以重新登录路由器获取）

注意事项

电量过低可能影响机器人的正常使用，例如执行动作不正常或无法连接机器人，请尽量在电量充足的情况下进行调试。

在关机时请先通过以下指令关闭树莓派，然后再切断机器人电源：
`sudo shutdown -P now`

使用关闭机器人电源的方式关闭树莓派系统可能会导致文件受损，出现 read-only file system 的问题。当出现此问题时可以通过执行下列指令解决。

```
? MobaXterm 20.3 ?
(SSH client, X-server and networking tools)

> SSH session to lemon@192.168.3.221
? SSH compression : ✓
? SSH-browser      : ✓
? X11-forwarding   : ✓ (remote display is forwarded through SSH)
? DISPLAY          : ✓ (automatically set on remote server)

> For more info, ctrl+click on help or visit our website

Last login: Mon Jun  7 19:24:48 2021 from 192.168.3.199
/usr/bin/xauth: error in locking authority file /home/lemon/.Xauthority
mktemp: failed to create file via template '/tmp/setup.sh.XXXXXXXXXX': Read-only file system
Could not create temporary file:
mktemp: failed to create file via template '/tmp/setup.sh.XXXXXXXXXX': Read-only file system
Could not create temporary file:
lemon@lemon:~$
```

```
sudo fsck /dev/mmcblk0p1
sudo fsck /dev/mmcblk0p2
sudo reboot
```

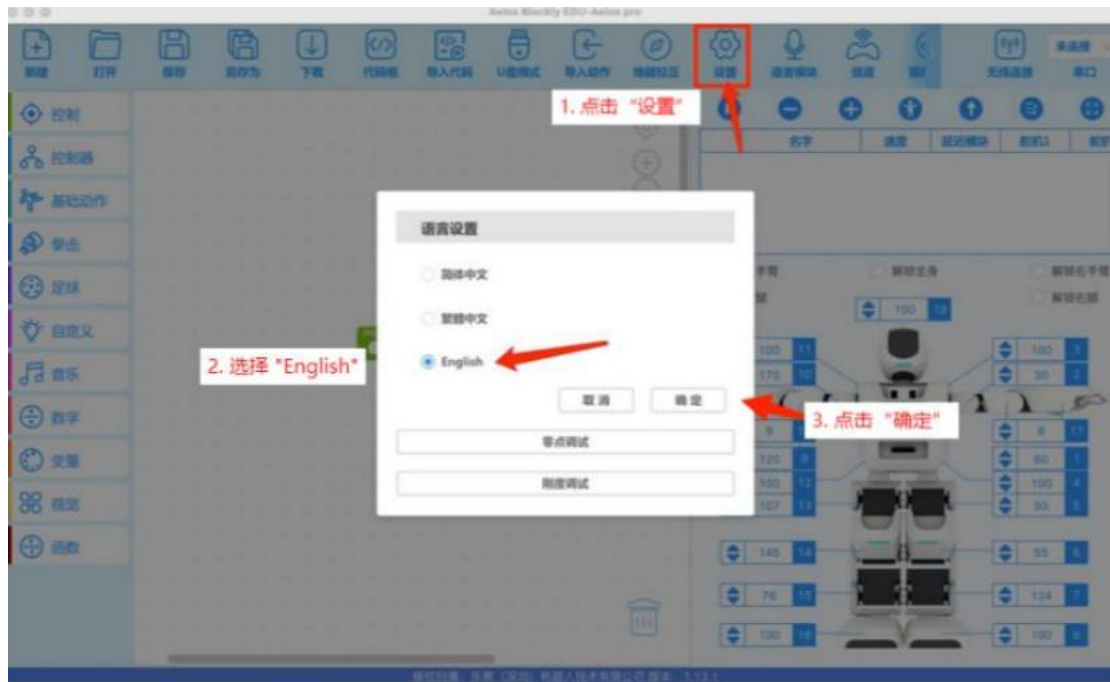
四、执行动作

- 打开“MobaXterm”软件，远程连接 ssh。
- 进入

"/home/lemon/catkin_ws/src/robot_demo/scripts/CMDcontrol.py" 此路径，再输入对应的动作名，即可执行。需要注意的是，在 Aelos 机器人中，动作可被分为三类：第一类是机器人系统内置的动作，第二类是用户自定义编辑的动作，第三类是音乐。对于每一类动作，在执行时会有稍许不一样的地方，需要仔细区分，下面将会对每一类进行介绍。

1、软件系统内置动作

- 对于软件内置的动作，需输入动作的英文名。因此需要先在设置中切换语言为英语，然后在对应分类下找到动作的英文名。例如下蹲对应 Squat。

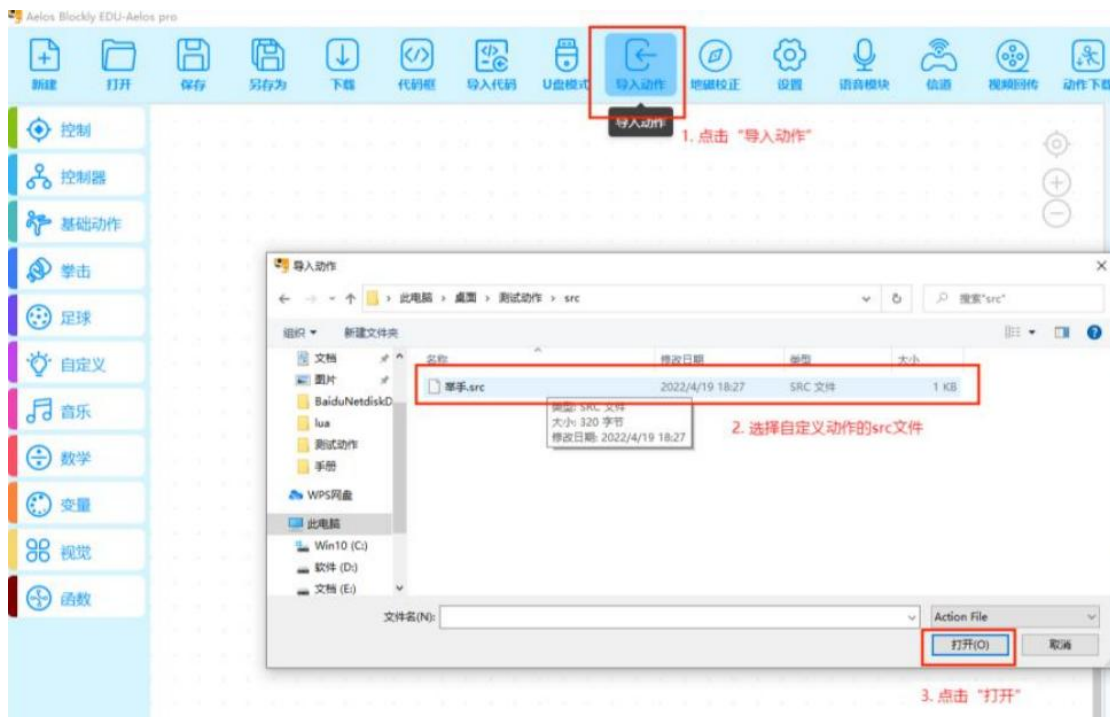


得到软件内置动作的英文名后，再进入路径，输入动作的英文名
字，机器人开始执行动作。

```
lemon@lemon:~$ "/home/lemon/catkin_ws/src/robot_demo/scripts/CMDcontrol.py"  
this is MDS_action  
please act_name123:提示...  
Squat  
haha: Squat  
act_name  
leju_f0b381df91e3f7af9b88235fb80e2f99  
action_wait done  
please act_name123:█
```

2、自定义编辑的动作

- 对于自定义编辑动作，需先将动作导入。具体的操作步骤如下图所示：



动作导入后，再点击动作下载。下载完毕后，重启机器人。

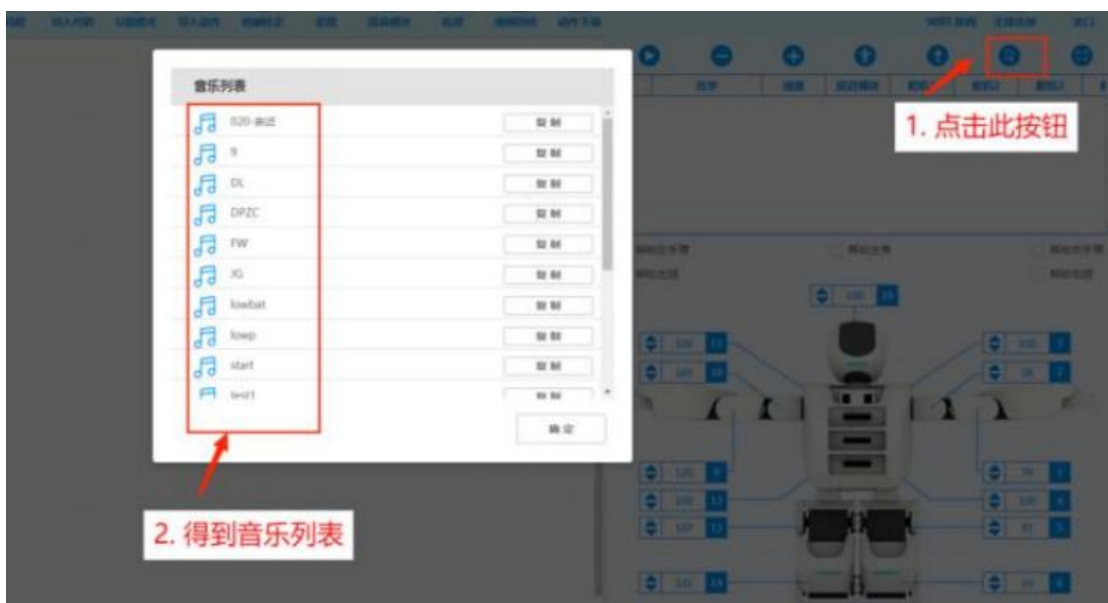


按照下图步骤，进入路径，输入自定义的动作名字，机器人开始执行动作

```
lemon@lemon:~$ "/home/Lemon/catkin_ws/src/robot_demo/scripts/CMDcontrol.py"
this is MD5_action
please act_name123:提示...
举手
haha: 举手
act_name
leju_c5ab749f1113956d563578e41f3c8936
action_wait done
please act_name123:
```

3、机器人音乐

- 对于机器人内置的音乐，可以在软件中通过音乐列表按键获取当前可用的音乐

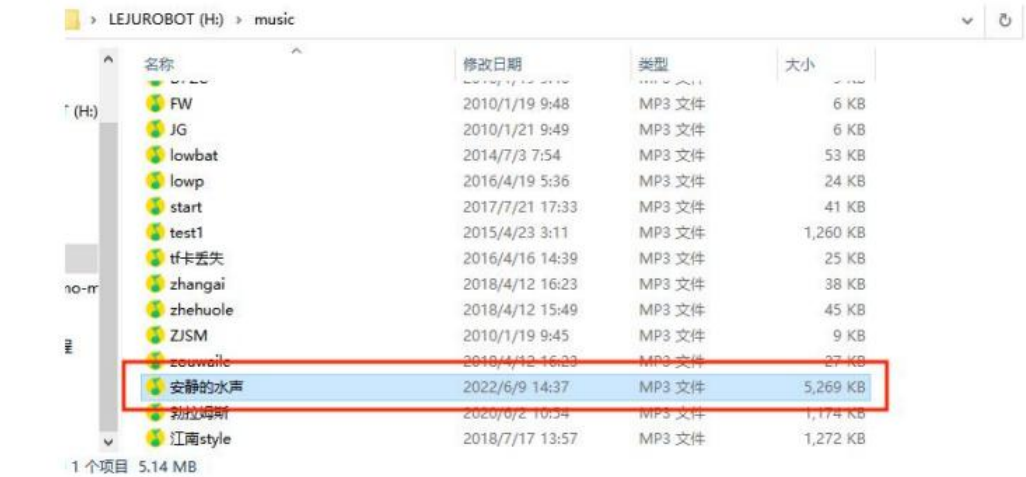


按照下图步骤，进入路径，输入动作名为 ‘play_’ 前缀加上音乐名，即可执行。

```
lemon@lemon:~$ "/home/lemon/catkin_ws/src/robot_demo/scripts/CMDcontrol.py"
this is MD5 action
please act_name123:提示...
play_tf卡丢失
haha: play_tf卡丢失
act_name
leju_02a613b34a84742ba9159e9d2c6bff6e
action_wait done
please act_name123:
```

若想要机器人播放其他音乐，可以用 U 盘模式添加音乐。具体步骤为：连接串口后，点击菜单栏的 U 盘模式。之后进入机器人 U 盘，将准备好的音乐文件拖入至 u 盘 MUSIC 文件夹中。例如，下图中“安静的水声”为新导入的音乐。

（注意：音乐文件需为 MP3 格式；音乐文件的名字中不能有特殊符号，只能为中英文文字和数字。）



导入音乐后，请重启机器人。连接串口，在音乐列表中，能找到我们之前添加的音乐。再点击“动作下载”，将音乐下载至机器

人。下载完毕后。重启机器人,再进入路径,输入动作名为‘play_’前缀加上音乐名,即可执行。

```
lemon@lemon:~$ "/home/lemon/catkin_ws/src/robot_demo/scripts/CMDcontrol.py"
this is MD5_action
please act_name123:提示...
play_安静的水声
haha: play_安静的水声
act_name
leju_1907bb9a3bb63b833ebc6e108da8c14a
action_wait done
please act_name123:█
```

五、代码下载

注意：运行脚本文件的话，会将原本镜像中的 robot_demo 文件夹删除，防止运行我们的代码时冲突，如修改过 robot_demo 文件夹，请先自行进行保存备份，以防代码丢失

```
$ sh -c "$(wget --no-cache http://aelosedu.lejurobot.com/college_caai_demo_downolader.sh -O -)"
```

```
lemon@lemon:~/catkin_ws/src$ sh -c "$(wget --no-cache http://aelosedu.lejurobot.com/college_caai_demo_downolader.sh -O -)"
--2022-06-22 15:12:16-- http://aelosedu.lejurobot.com/college_caai_demo_downolader.sh
Resolving aelosedu.lejurobot.com (aelosedu.lejurobot.com)... 119.96.52.41, 240e:c3:2c00:303::71db:8e29, 240e:95d:801:1::6fb1:829
Connecting to aelosedu.lejurobot.com (aelosedu.lejurobot.com)[119.96.52.41]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1064 (1.0K) [text/x-sh]
Saving to: 'STDOUT'

-
100K[=====] 1.04K --KB/s in 0s

2022-06-22 15:12:16 (20.4 MB/s) - written to stdout [1064/1064]

开始下载压缩包...
--2022-06-22 15:12:16-- http://aelosedu.lejurobot.com/college_caai.zip
Resolving aelosedu.lejurobot.com (aelosedu.lejurobot.com)... 119.96.52.41, 240e:c3:2c00:303::71db:8e29, 240e:95d:801:1::6fb1:829
Connecting to aelosedu.lejurobot.com (aelosedu.lejurobot.com)[119.96.52.41]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 133601 (130K) [application/zip]
Saving to: 'college_caai.zip'

college_caai.zip
100K[=====] 130.47K --KB/s in 0.07s

2022-06-22 15:12:16 (1.70 MB/s) - 'college_caai.zip' saved [133601/133601]

解压...
Archive: college_caai.zip
creating college_caai/
```

比赛中会用到的程序、文件等均可在 college_caai 文件夹中找到，可以复制一份到电脑中。

六、运行 ROS 节点

ROS (Robot Operating System, 下文简称“ROS”) 是一个适用于机器人的开源的元操作系统。它提供了操作系统应有的服务，包括硬件抽象，底层设备控制，常用函数的实现，进程间消息传递，以及包管理。它也提供用于获取、编译、编写、和跨计算机运行代码所需的工具和库函数。

1、运行节点

```

lemon@lemon:~/catkin_ws/src/college_caai/launch$ roslaunch AI_competition.launch
... logging to /home/lemon/.ros/log/576d5a78-f1fb-11ec-8352-499c30e23340/roslaunch-lemon-44228.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://lemon:43867/

SUMMARY
=====

PARAMETERS
* /ar_track_alvar/marker_size: 4.4
* /ar_track_alvar/max_new_marker_error: 0.05
* /ar_track_alvar/max_track_error: 0.05
* /ar_track_alvar/output_frame: /map
* /roscpp: noetic
* /rosversion: 1.15.9
* /usb_cam_chest/camera_frame_id: usb_cam
* /usb_cam_chest/image_height: 480
* /usb_cam_chest/image_width: 640
* /usb_cam_chest/io_method: mmap
* /usb_cam_chest/pixel_format: yuyv
* /usb_cam_chest/video_device: /dev/video2
* /usb_cam_head/camera_frame_id: usb_cam
* /usb_cam_head/image_height: 480
* /usb_cam_head/image_width: 640
* /usb_cam_head/io_method: mmap
* /usb_cam_head/pixel_format: yuyv
* /usb_cam_head/video_device: /dev/video2

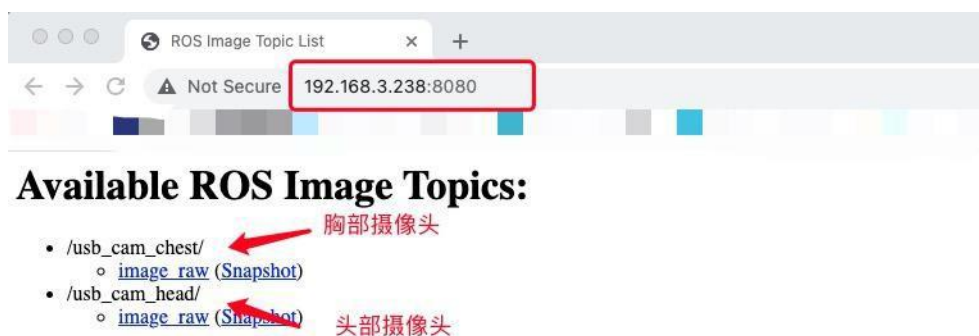
```

输入：cd /home/lemon/catkin_ws/src/college_caai/launch
回车

执行 roslaunch AI_competition.launch，启动 ROS 节点。

2、测试摄像头节点

将电脑和机器人接在相同局域网中，打开浏览器，连接机器人 8080 端口，将会显示如下页面：



此时，点击对应链接即可显示摄像头画面。

3、识别 ARtag

(1) 在 AI_competition.launch 中已经包含启动 ARtag 识别的节点，在执行 roslaunch 命令时会直接启动。

```

<node pkg="tf" type="static_transform_publisher" name="world_to_cam" args="0 0 0 0 0 1 /map /usb_cam 100"/>
  <arg name="marker_size" default="4.4" />
  <arg name="max_new_marker_error" default="0.05" />
  <arg name="max_track_error" default="0.05" />
  <arg name="cam_image_topic" default="/usb_cam_chest/image_raw" />
  <arg name="cam_info_topic" default="/usb_cam_chest/camera_info" />
  <arg name="output_frame" default="/map" />
  <node name="ar_track_alvar" pkg="ar_track_alvar" type="individualMarkersNoKinect" respawn="false" output="screen">
    <param name="marker_size" type="double" value="$(arg marker_size)" />
    <param name="max_new_marker_error" type="double" value="$(arg max_new_marker_error)" />
    <param name="max_track_error" type="double" value="$(arg max_track_error)" />
    <param name="output_frame" type="string" value="$(arg output_frame)" />
    <remap from="camera_image" to="$(arg cam_image_topic)" />
    <remap from="camera_info" to="$(arg cam_info_topic)" />
  </node>

```

(2) 在 college_tag_traker_fast_right.py 文件中包含一个用来处理 tag 信息的类 TagConverter, 这个类在初始化时会启动对识别节点消息的监听。

```

17
18 class TagConverter():
19     def __init__(self):
20         self.sub = rospy.Subscriber(
21             '/ar_pose_marker', AlvarMarkers, self.sub_cb)
22         self.markers = []
23
24     def sub_cb(self, msg):

```

(3) sub_cb 方法对读取到的 tag 的信息进行处理, 最终返回 tag 的 id、距离、角度等参数。

```

24     def sub_cb(self, msg):
25         self.markers = []
26         time_sec = msg.header.stamp.secs
27         for marker in msg.markers:
28             pos = marker.pose.pose.position
29             quat = marker.pose.pose.orientation
30             # print(marker)
31
32             rpy = tf.transformations.euler_from_quaternion(

```

(4) 在本 demo 中是按照数字的顺序粘贴的 ARtag, 因此在类中提供一个 get_nearest_marker 方法用来获取 id 最小的 tag。其他情况下可以通过上面的 dist 等参数进行获取。

```
def get_nearest_marker(self):
    min_id = 999
    min_idx = 0
    markers = []
    for i in range(50):
        time.sleep(0.01)
        markers += self.markers

    for index, m in enumerate(markers):
        if m[0] < min_id:
            min_idx = index
            min_id = m[0]
    if min_id == 999:
        return []
    else:
        return markers[min_idx]
```

获取最小的id

(5) 当 artag 对不正时，可以打开测试标定，并注释掉动作

```
36 for i in range(len(rpy)):
37     rpy_arc[i] = rpy[i] / math.pi * 180
38
39 # print(rpy_arc)
40 # print("poseX:", pos.x, "poseY:", pos.y, "poseZ:", pos.z)
41 # print("poseX--poseY--rpy_y:", pos.x, ",",
42 #       pos.y, ",", rpy_arc[2]) # 测试标点 x不标
43
44 self.markers.append(
```

将这两行注释打开

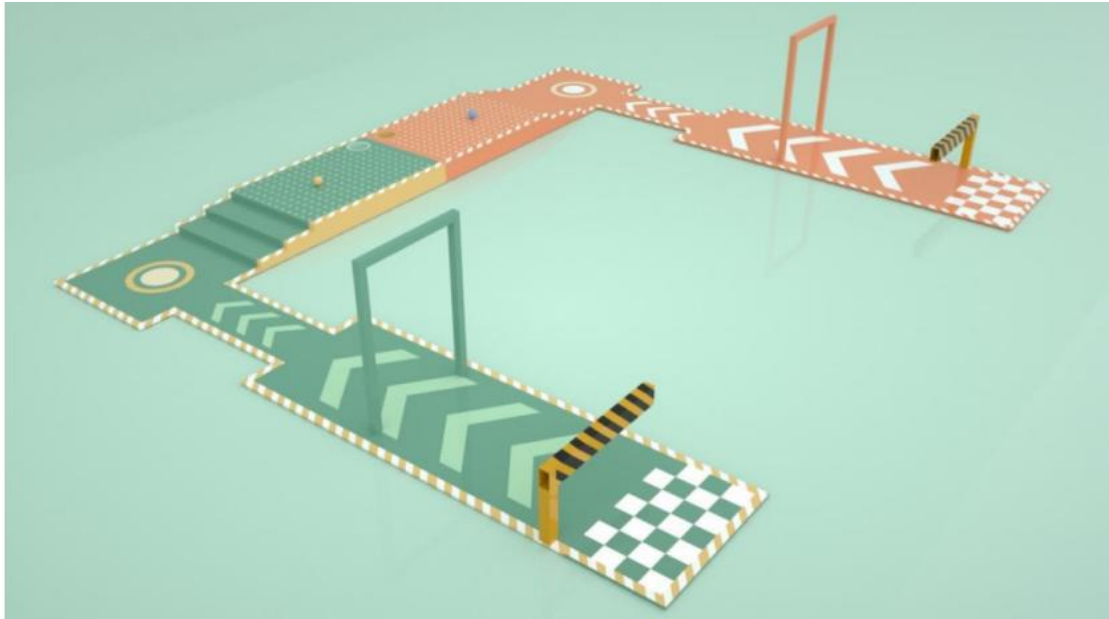
```
74
75
276 # 准备行走到窄桥起点，对齐二维码执行快走
277 elif marker[0] == 3:
278     stage = 'bridge_start'
279     result = turn_to_tag(robot_tag_x, robot_tag_y, tag_yaw,
280                        0.0, 0.10862467691622797, -10.682594986449436)
281     if (result == False):
282         continue
283
284     print("dis_x:", robot_tag_x, "dis_y:",
285           robot_tag_y, "theta:", tag_yaw)
286     print('快走过桥')
287     action('fastForward03')
288
289 # 窄桥中间
290 elif marker[0] == 4:
291     stage = 'bridge_middle'
292
293     result = turn_to_tag(robot_tag_x, robot_tag_y, tag_yaw,
294                        0.0, 0.10862467691622797, -10.682594986449436)
295     if (result == False):
296         continue
297     print('行走至独木桥中间')
298     action('fastForward03')
299
300 # 窄桥终点
```

然后运行程序，可以获取到标点信息

然后替换到对应的 tag 码下的 turn_tu_tag

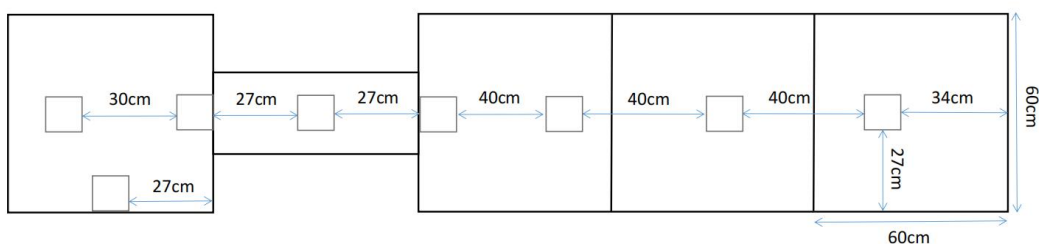
七、比赛说明

1、场地说明



整个赛道由 5 个任务组成：过道闸、过门、过独木桥、上台阶以及踢球，主要应用颜色识别和 ARtag 识别完成任务。给出的示例代码中主要包含了后面 5 个任务的演示案例，通过在场地中央粘贴 ARtag 作为机器人前进的参照物，使机器人能够沿着设置好的路线前进通过路上的障碍。用户可自行调整程序改善整体实现效果。

▪ ARtag 粘贴示意图如下：



2、思路讲解

在程序中根据识别到 ARtag 不同将整个程序分为快走阶段、过桥阶段、转弯阶段、上楼梯阶段以及踢球阶段。

(1) 快走阶段：先执行对齐 tag 操作，然后执行快走。


```

256 # 快走阶段
257 if marker[0] < 3:
258
259     if (marker[0] == 0) and (stage == 'stage_one'):
260         print('看不到栏杆直接快走5555')
261         action('fastForward05')
262     elif (marker[0] == 0) and (stage != 'stage_one'):
263         action('Back2Run')
264     elif marker[0] == 1:
265         result = turn_to_tag(
266             robot_tag_x, robot_tag_y, tag_yaw, 0.0, 0.10413927570163643, -10.90461776956659)
267         if(result == True):
268             print('快走5555')
269             action('fastForward05')
270     else:
271         result = turn_to_tag(robot_tag_x, robot_tag_y, tag_yaw, 0.0, 0.11164769788617308, -10.950821964652746)
272         if(result == True):
273             print('快走5555')
274             action('fastForward05')
275

```

(2) 过桥阶段：在桥的起点、终点及桥中央分别贴有 ARtag，在桥的起点处会执行比较严格的方向校正，保证整体过桥的效果。

```

276 # 准备行走到窄桥起点，对齐二维码执行快走
277 elif marker[0] == 3:
278     stage = 'bridge_start'
279     result = turn_to_tag(robot_tag_x, robot_tag_y, tag_yaw,
280                         0.0, 0.10862467691622797, -10.682594986449436)
281     if (result == False):
282         continue
283
284     print("dis x:", robot_tag_x, "dis y:",
285           robot_tag_y, "theta:", tag_yaw)
286     print('快走过桥')
287     action('fastForward03')
288
289 # 窄桥中间
290 elif marker[0] == 4:
291     stage = 'bridge_middle'
292
293     result = turn_to_tag(robot_tag_x, robot_tag_y, tag_yaw, 0.0, 0.10862467691622797, -10.682594986449436)
294     if (result == False):
295         continue
296     print('行走至独木桥中间')
297     action('fastForward03')
298
299 # 窄桥终点
300 elif marker[0] == 5:
301     stage = 'bridge_end'
302
303     result = turn_to_tag(robot_tag_x, robot_tag_y, tag_yaw, 0.0, 0.10862467691622797, -10.682594986449436)
304     if (result == False):
305         continue
306     print('快走333至独木桥终点')
307
308     action('fastForward03')
309

```

校正过程，调用turn_to_tag函数进行tag的校正

(3) 转向阶段：在转向处以及台阶前端贴有 ARtag，先执行转向，然后走到台阶前准备上台阶。


```

310
311
312     elif marker[0] == 6:
313         stage = 'corner_turn'
314         result = turn_to_tag(robot_tag_x, robot_tag_y, tag_yaw, 0.0 , 0.10970597601956752 , -10.018373577110546)
315         if (result == False):
316             continue
317
318         print('行走至转向位, 然后转向')
319         flag = True
320
321         action('turn010R')
322         action('turn010R')
323         time.sleep(1)
324         action("Right02move")
325
326         print("66666666666")
327         # time.sleep(5)
328
329     elif marker[0] == 7:
330         if (flag == True):
331             stage = 'prepare_up_floor'
332             result = turn_to_tag(
333                 robot_tag_x, robot_tag_y, tag_yaw, 0.0 , 0.10862467691622797 , -10.682594986449436)
334             if (result == False):
335                 continue
336
337             # follow once 到楼梯起始位
338             # 往前挪几步贴住台阶
339             # 执行上台阶动作1、2、3
340             print('准备上台阶')
341             action('Forwalk02')

```

(4) 上楼梯阶段：如果转向阶段能够顺利执行，那么本段主要难点在于机器人动作的调整，逻辑方面连续执行动作即可。

```

219
220
221         if stage == 'prepare_up_floor':
222             print('开始上台阶')
223
224             action("Forwalk00")
225             for i in range(1):
226                 walk_slow()
227             action("turn001R")
228             action("Forwalk00")
229
230             action('UpBridge2')
231             walk_slow00()
232
233             action('UpBridge2')
234             walk_slow00()
235
236             action('UpBridge2')
237             time.sleep(0.5)
238             action('turn001L')
239             stage = 'kick_ball'

```

(5) 踢球阶段：给出的示例中主要使用了颜色识别，分别在图片中找到球和球洞的位置，通过调整机器人的位置使机器人、球和球洞位于一条直线上，然后执行踢球动作。

3、说明

(1) 主程序在

/home/lemon/catkin_ws/src/college_caai/scripts/college_tag_traker_left.py 中 (left 和 right 文件分别对应不同转弯方向)，运行前需要先执行 roslaunch 命令跑起 ros 环境。

(2) 颜色识别较容易受到光照、自然光等因素影响，在调试时需要尽量减少周围环境影响，例如用白板围住场地、拉上窗帘打开室内灯。如果识别效果不佳可以调整颜色参数进行改善。

(3) 练习场地的 ARtag 粘贴方式不一定是最终比赛的粘贴式，仅供参考。

4、修改程序

(1) 打开 college_tag_traker_fast_right.py

(2) 找到 main 函数

```
202
203
204 def main():
205     try:
206         cur_tagid = 0
207         step = 0
208
209         rospy.init_node('ar_tag_tracker')
210         tag = TagConverter()
```

(3) 调整转向幅度和旋转方向修改

```
235
236
237     if stage == 'corner_turn':
238         print('转向对准楼梯Tag')
239         has_big_turn = True
240         walk_slow()
241         walk_slow()
242         action('turn010R')
243         action('turn003R')
```

```

299         if marker[0] == 5:
300             if not has_big_turn:
301                 walk_slow()
302                 walk_slow()
303                 walk_slow()
304                 action('turn010R')
305                 action('turn003R')

```

修改转向幅度：修改 action 中动作

左右转向修改：action 中左右转动作更换

(4) 循环执行向前快走

循环执行向前快走

```

while not rospy.is_shutdown():
    time.sleep(0.1)
    action('fastForward02')

```

每走两步执行一次右转

```

while not rospy.is_shutdown():
    time.sleep(0.1)
    action('fastForward02')
    action('turn003R')

```

转向两次后停止动作

```

while not rospy.is_shutdown():
    time.sleep(0.1)
    action('turn003R')
    action('turn003R')
    exit(0)

```

八、HSV 颜色获取、修改及踢球程序验证步骤（参照视频）

1. OpenCV、python 下载与安装

颜色识别前需安装 OpenCV、python 及依赖库，建议使用下面方法，如果安装过其他版本的 python 需要先卸载再进行安装。

(1) 下载 Anaconda，地址 <https://www.anaconda.com/>，本次

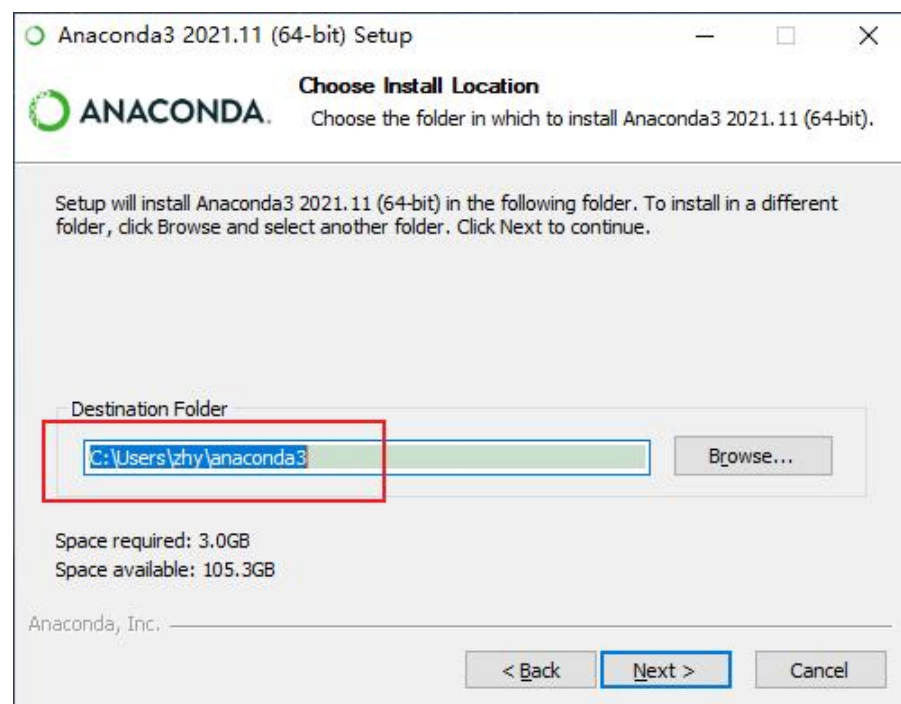
版本 Anaconda3-2021.11-Windows-x86_64.exe

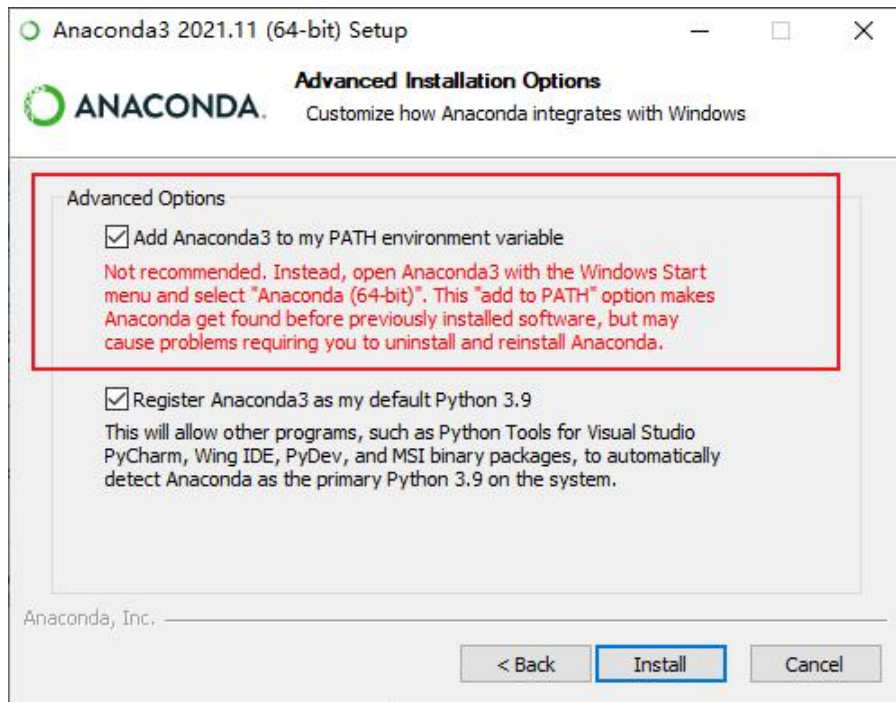
下载 OpenCV 包，地址

<https://pypi.tuna.tsinghua.edu.cn/simple/opencv-contrib-python/>，选择

opencv_contrib_python-4.5.2.52-cp39-cp39-win_amd64.whl

(2) 安装 Anaconda，注意记住安装路径，并选择注册环境变量，如下图，安装完成后可在开始菜单找到





将

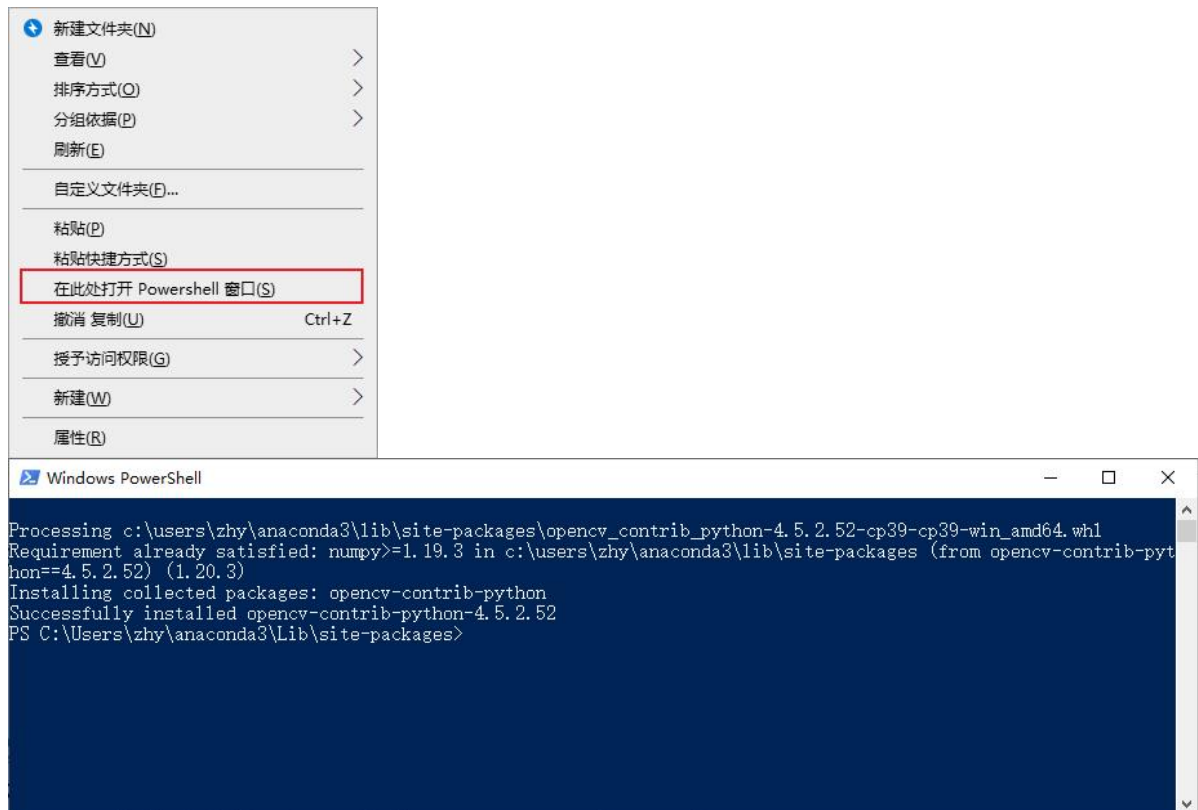
opencv_contrib_python-4.5.2.52-cp39-cp39-win_amd64.whl

文件复制到 Anaconda 安装路径 anaconda3\Lib\site-packages\
下

在这个文件夹中，按住 shift 单击右键打开 powershell 窗口，
输入

```
pip install .\opencv_contrib_python-4.5.2.52-cp39-cp39-win_amd64.whl,
```

完成 OpenCV 安装



2、HSV 颜色获取、修改及踢球程序验证

方式二：

1、运行视觉节点

```
cd /home/lemon/catkin_ws/src/college_caai/launch
```

```
roslaunch AI_competition.launch
```

2、获取图片，将电脑和机器人接在相同局域网中，打开浏览器，连接机器人 8080 端口（如：192.168.3.238:8080），将会显示如下页面：



Available ROS Image Topics:

- /usb_cam_chest/
 - [image_raw \(Snapshot\)](#) 胸部摄像头
- /usb_cam_head/
 - [image_raw \(Snapshot\)](#) 头部摄像头

3、点击胸部摄像头的 Snapshot, 获取一张图片, 将图片另存到与 RGB2HSV_sampling.py 相同的文件夹, 图片格式为 jpg.

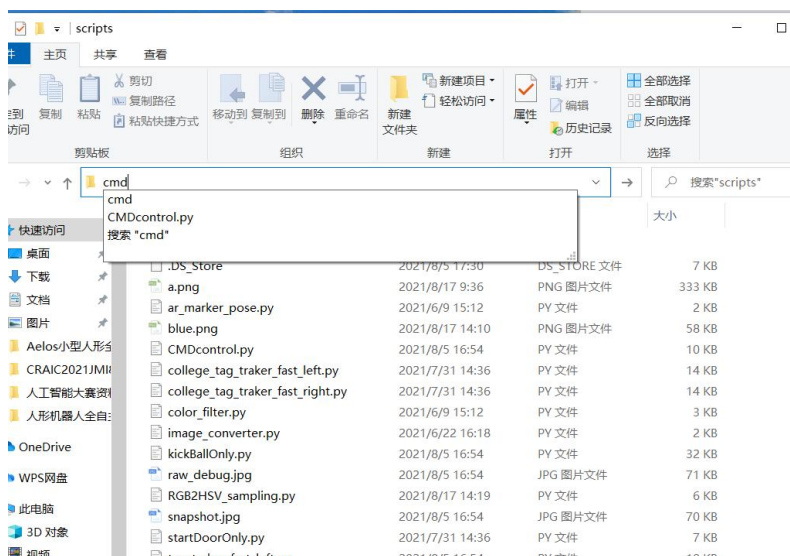
4、使用编译器打开 RGB2HSV_sampling.py, 将第 15 行 “blue” 更改为要识别的图片名字

```
15 photo_path='./blue.png'
```

5、在 RGB2HSV_sampling.py 中增加第 73 行（第一次运行此程序时需要增加）

```
72 # 新建窗口
73 cv2.namedWindow("Camera",cv2.WINDOW_NORMAL)
74 cv2.namedWindow("robotPreviewH",cv2.WINDOW_NORMAL)
75 cv2.namedWindow("robotPreviewH_HSV",cv2.WINDOW_NORMAL)
76 cv2.namedWindow("colorMask",cv2.WINDOW_NORMAL)
77
```

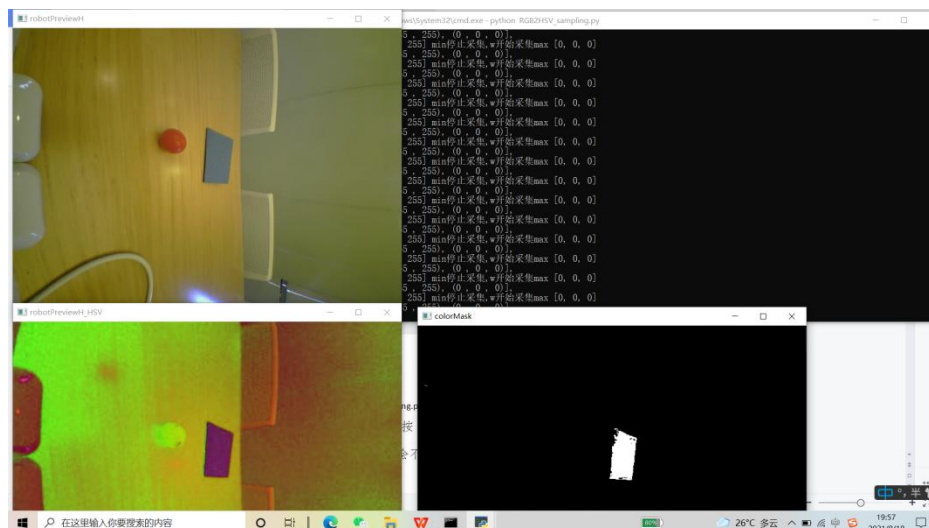
6、打开 RGB2HSV_sampling.py 存储的文件夹, 在地址框输入 cmd



7、回车会出现下图

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation. 保留所有权利。
C:\Users\Lenovo\Desktop\人工智能大赛资料包_V1.2\代码\robot_demo\scripts>
```

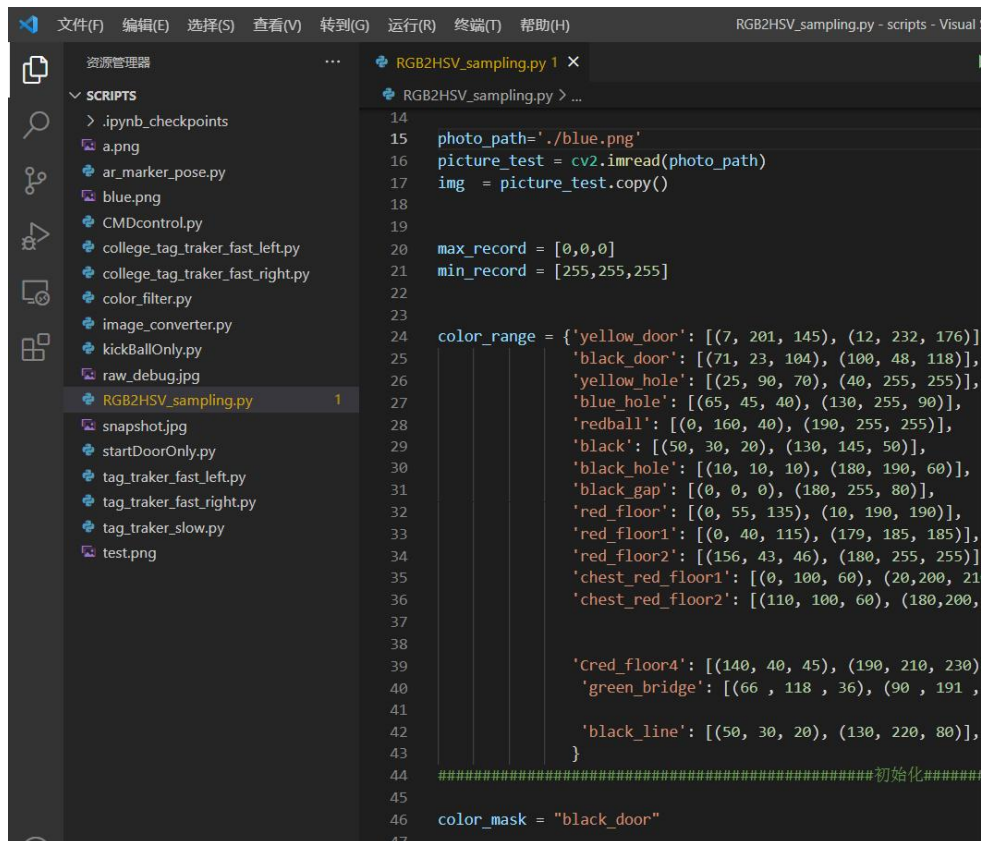
8、输入 python RGB2HSV_sampling.py,回车，出现下图则运行成功，按“w”开始采集颜色，将鼠标放在要采集的图片物体上晃动，程序会不停打印出识别的颜色（英文输入法）



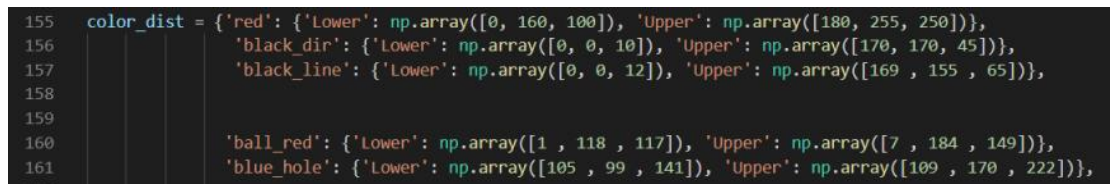
9、按“w”停止采集颜色，代码最后一组数值为采集的 HSV 数值

```
C:\Windows\System32\cmd.exe - python RGB2HSV_sampling.py
259 182 [ 8 220 151] 正在采集, w停止采集
260 182 [ 8 220 153] 正在采集, w停止采集
261 182 [ 8 219 154] 正在采集, w停止采集
262 182 [ 8 217 155] 正在采集, w停止采集
263 182 [ 9 219 156] 正在采集, w停止采集
264 182 [ 9 221 157] 正在采集, w停止采集
265 182 [ 9 223 158] 正在采集, w停止采集
266 183 [ 9 220 158] 正在采集, w停止采集
267 183 [ 10 218 160] 正在采集, w停止采集
268 183 [ 10 217 163] 正在采集, w停止采集
268 183 [ 10 217 163] 正在采集, w停止采集
269 183 [ 10 218 165] 正在采集, w停止采集
270 183 [ 11 218 164] 正在采集, w停止采集
271 183 [ 11 224 163] 正在采集, w停止采集
271 182 [ 11 220 166] 正在采集, w停止采集
271 181 [ 11 218 170] 正在采集, w停止采集
271 181 [ 11 218 170] 正在采集, w停止采集
272 180 [ 10 219 169] 正在采集, w停止采集
272 180 [ 10 219 169] 正在采集, w停止采集
停止采集
C:\Users\Lenovo\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9.qbz5n2kfraSp0\LocalCache\local-packages\Python39\site-packages\seaborn\distributions.py:2557: FutureWarning: distplot is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
[[7, 201, 144] min停止采集, w开始采集max [12, 228, 176]
[[7, 201, 144], [12, 228, 176]],
[[7, 201, 144] min停止采集, w开始采集max [12, 228, 176]
[[7, 201, 144], [12, 228, 176]]
```

10、使用编译器打开 RGB2HSV_sampling.py, 将 24 行的数值改为采集后的数值，将 46 行红色的名字改为 24 行的红色名字，保存后，再次运行程序，黑色 colorMask 图片上会出现物体的白色轮廓，说明取色成功。



11、将 kickBallOnly.py 的 160（球的颜色）、161（球洞颜色）修改为采集 HSV 的颜色值然后保存



17、运行 kickBallOnly 验证踢球

方式一：使用 VScode 软件

1、运行视觉节点

```

cd /home/lemon/catkin_ws/src/college_caai/launch
roslaunch AI_competition.launch

```

2、获取图片，将电脑和机器人接在相同局域网中，打开浏览器，连接机器人 8080 端口（如：192.168.3.238:8080），将会显示如下页面：



- 3、点击胸部摄像头的 Snapshot, 获取一张图片, 将图片另存到与 RGB2HSV_sampling.py 相同的文件夹, 图片格式为 jpg.
- 4、使用 VSCode 软件打开 RGB2HSV_sampling.py, 将第 15 行“blue”更改为要识别的图片名字

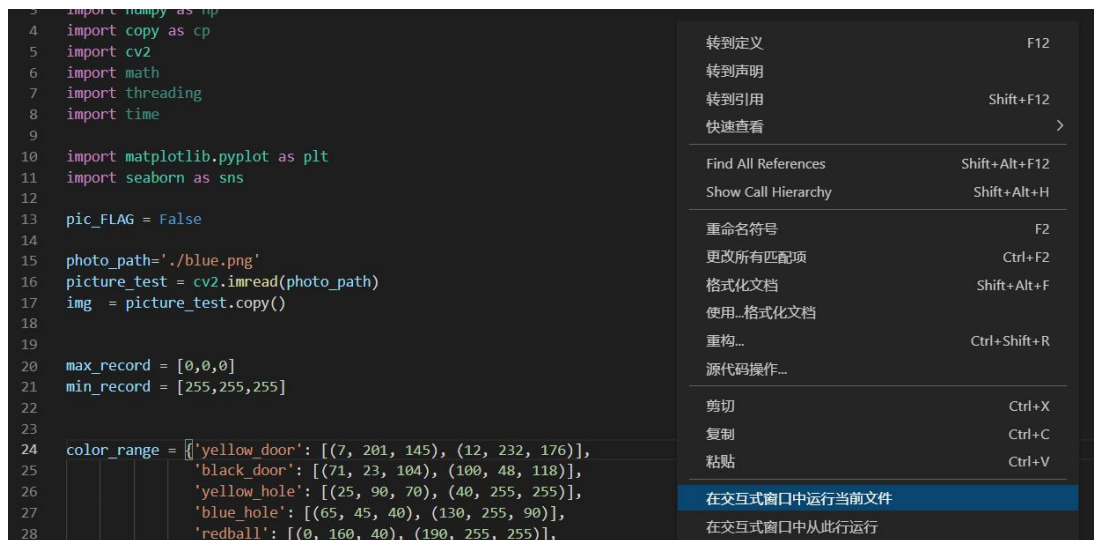
```
15 photo_path='./blue.png'
```

- 5、在 RGB2HSV_sampling.py 中增加第 73 行（第一次运行此程序时需要增加）

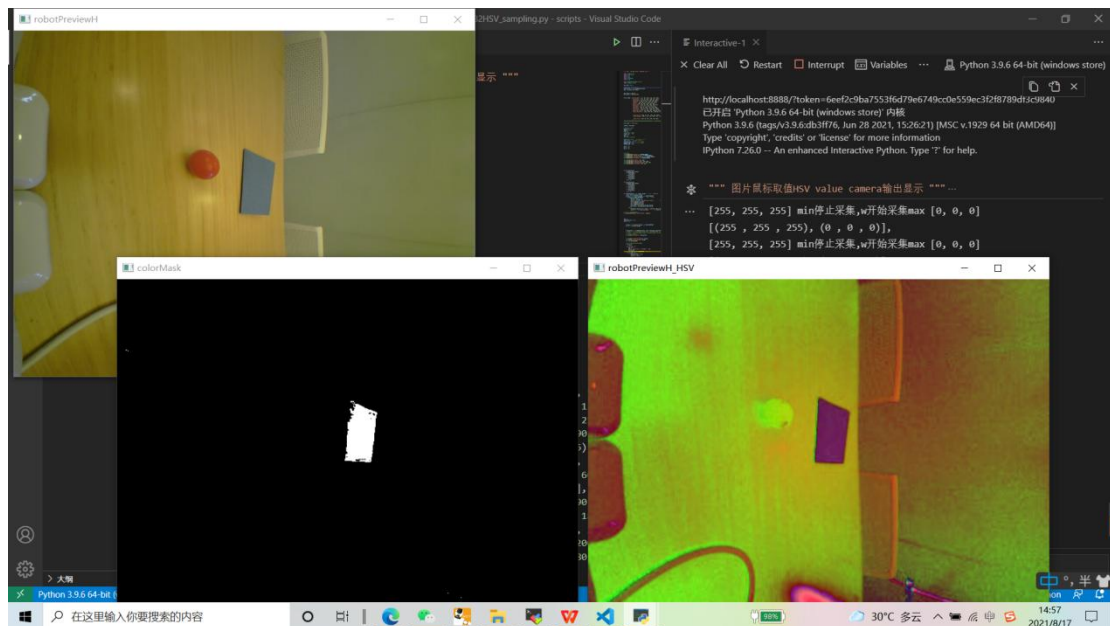
```
72 # 新建窗口
73 cv2.namedWindow("Camera",cv2.WINDOW_NORMAL)
74 cv2.namedWindow("robotPreviewH",cv2.WINDOW_NORMAL)
75 cv2.namedWindow("robotPreviewH_HSV",cv2.WINDOW_NORMAL)
76 cv2.namedWindow("colorMask",cv2.WINDOW_NORMAL)
77
```

- 6、在 VSCode 中右键选择“在交互式窗口运行当前文件”运行程序

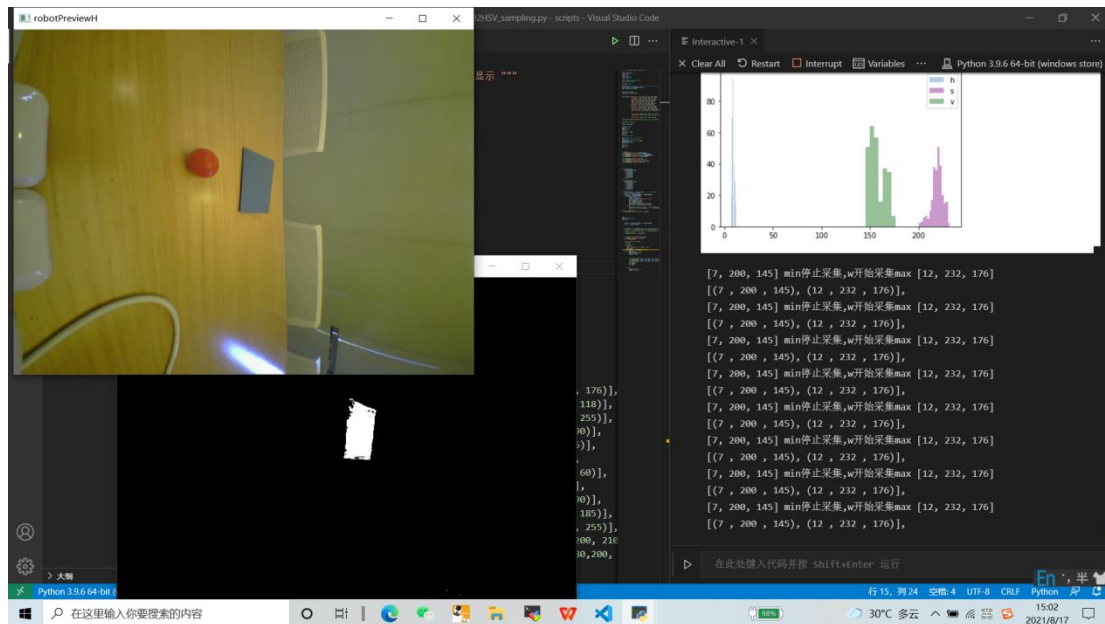
（需要安装好 python 及相关依赖库）



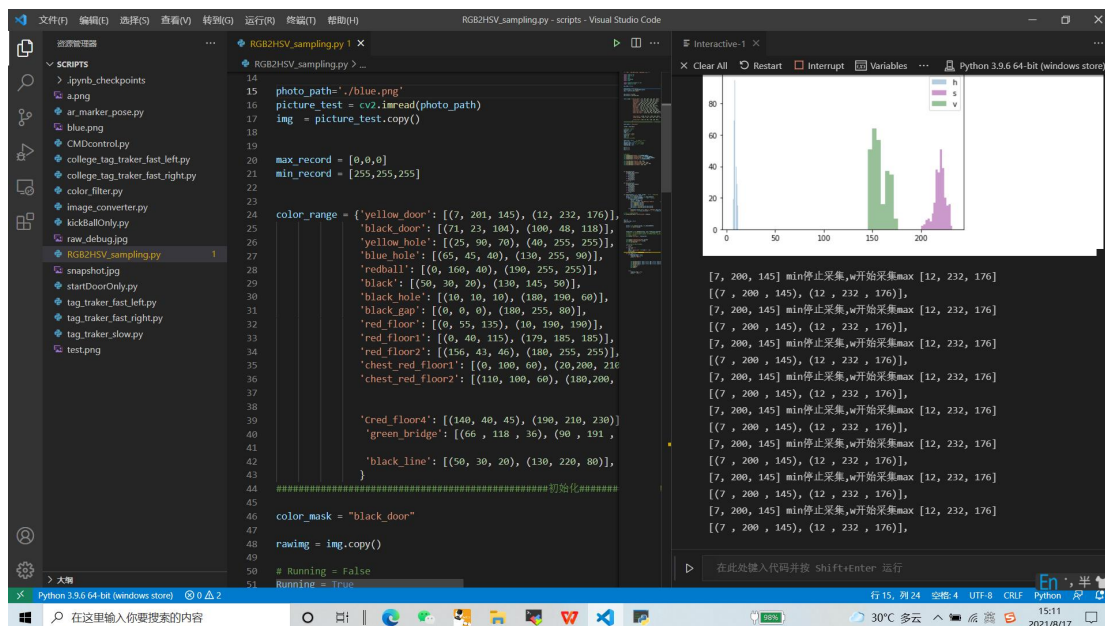
7、出现下图运行成功，按“w”开始采集颜色，将鼠标放在要采集的图片物体上晃动，程序会不停打印出识别的颜色（英文输入法）



8、按“w”停止采集颜色，代码最后一组数值为采集的 HSV 数值



9、验证采集的颜色，将 24 行的数值改为采集后的数值，将 46 行红色的名字改为 24 行的红色名字，再次运行程序黑色图片上会出现白色的轮廓



10、将 kickBallOnly.py 的 160（球的颜色）、161（球洞颜色）修改为采集 HSV 的颜色值然后保存

```

155 color_dist = {'red': {'Lower': np.array([0, 160, 100]), 'Upper': np.array([180, 255, 250])},
156               'black_dir': {'Lower': np.array([0, 0, 10]), 'Upper': np.array([170, 170, 45])},
157               'black_line': {'Lower': np.array([0, 0, 12]), 'Upper': np.array([169, 155, 65])},
158
159               'ball_red': {'Lower': np.array([1, 118, 117]), 'Upper': np.array([7, 184, 149])},
160               'blue_hole': {'Lower': np.array([105, 99, 141]), 'Upper': np.array([109, 170, 222])},
161

```

11、运行 kickBallOnly 验证踢球

九、比赛程序运行

在运行程序之前要确认下载过工程：

Aelos_edu 桌面软件打开 abe 格式的“src 动作模块”工程，点“动作下载”。

1、MobaXterm 第一个终端：

```

cd /home/lemon/catkin_ws/src/college_caai/launch
roslaunch AI_competition.launch

```

2、MobaXterm 第二个终端：

```

cd /home/lemon/catkin_ws/src/college_caai/scripts/
①python college_tag_traker_fast_right.py
②python college_tag_traker_fast_left.py

```

十、常见 FAQ

1. 连接网线后无法在路由器管理页面找到设备

(1)可能是路由器配置问题,如果路由器选择“自动选择 WAN 网口”，路由器可能会把机器人误识别为宽带入口，导致在设备列表中不显示 机器人。将配置修改为“固定 WAN 网口”可以解决此问题。



2. 远程 ssh 登录时出现 Access denied 错误

此问题由于登录时输入的账号密码有误导致，请仔细核对路由器中设备的 IP 是否正确以及设备名与资料中是否一致，并确认正确输入了资料中提供的账号和密码。

3. 运行 CMDcontrol 时出错

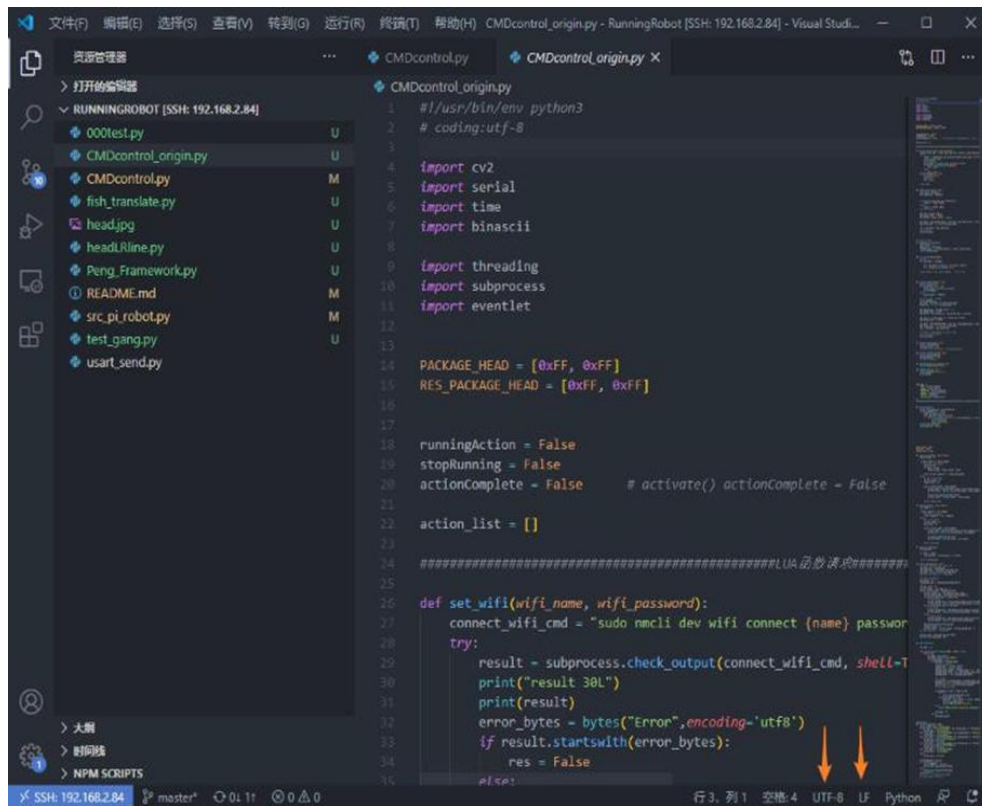
(1) 请确保文件具有执行权限，执行命令 `$ls`，绿色字体文件表示具有执行权限

```
scripts — lemon@lemon: ~/catkin_ws/src/robot_demo/scripts — ssh lemon@192.168.3.102 — 80x24
lemon@lemon:~$ cd catkin_ws/src/robot_demo/scripts/
lemon@lemon:~/catkin_ws/src/robot_demo/scripts$ ls
CMDcontrol.py          debug.jpg
RGB2HSV_sampling.py    image_converter.py
__pycache__            kickBallOnly.py
ar_marker_pose.py      tag_traker_fast_left.py
college_tag_traker_fast_left.py  tag_traker_fast_right.py
college_tag_traker_fast_right.py  tag_traker_slow.py
color_filter.py
lemon@lemon:~/catkin_ws/src/robot_demo/scripts$
```

可以使用 `$ sudo chmod +x [文件名]` 添加执行权限

(2) 如果运行时提示没有那个文件或目录，请先确认文件的路径没有错误。另外文件编码方式不一样也有可能導致此错误，Windows 系统下的换行符为 CRLF，Linux 系统下换行符为 LF。

可以在 vscode 中看到文件的编码方式，修改为对应系统的换行方式即可。



4. 运行 CMDcontrol 时不执行动作

- (1) 请打开 main.lua、lib.lua 和 Actionlib.lua 三个文件查看是否乱码，如果乱码将电脑中的备份文件进行替换。
- (2) 请确认机器人没有处在在线模式或者 U 盘模式，如果无法确定请按 reset 键重启机器人。
- (3) 在 action_append 函数中添加打印，打印队列中待执行的动作函数的名字，并确认 Actionlib 文件中包含对应的动作函数。

```

actionComplete = False
if len(action_list) > 0 :
    print("队列超过一个动作")
    action_list.append(acted_name)
else:
    action_list.append(acted_name)

```

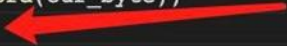


(4) 在 CMD_transfer 方法中添加打印, 如果该打印没有执行, 则串口没有收到数据, 请检查串口的名称是否正确。

```

with serial.Serial('/dev/serial0', 9600) as ser:
    while True:
        cur_byte = ser.read()
        tmp_head.append(ord(cur_byte))
        print(tmp_head)
        if len(tmp_head) == 2:
            if tmp_head == PACKAGE_HEAD:
                # print("receive")
                length_high = ord(ser.read())
                length_low = ord(ser.read())
                package_data_length = (length_high << 8) + length_low
                package_data_byte = ser.read(package_data_length)
                package_data = list(package_data_byte)

```



5. 执行动作不正常或摔倒

(1) 在桌面软件中调整机器人的零点。(调零点时, 建议备份现有零点文件)

(2) 在桌面软件中调整动作, 可以通过修改动作帧的舵机值和刚度 值来调整动作的执行情况。每台机器人都可能有微小的差别, 需要根据实际情况调整零点和舵机值。

(3) 在执行动作过程中尽量保证电量充足。

6. 使用过程中出现系统无法正常使用, 例如提示 read-only file system 等问题

(1) 此问题一般由于直接关闭机器人电源，树莓派在工作状态下直接断电，导致系统文件损坏引起。建议在关闭机器人时先使用 `$sudo shutdown-P now` 指令关闭树莓派，然后再关闭机器人电源，以减少此问题发生的可能。

(2) 如果已经出现问题，建议根据问题查找对应的解决方案，或者重新烧录镜像进行解决。