

空地协同实验注意事项

实验一、无人机与ep机器人的SDK控制

1.无人机控制

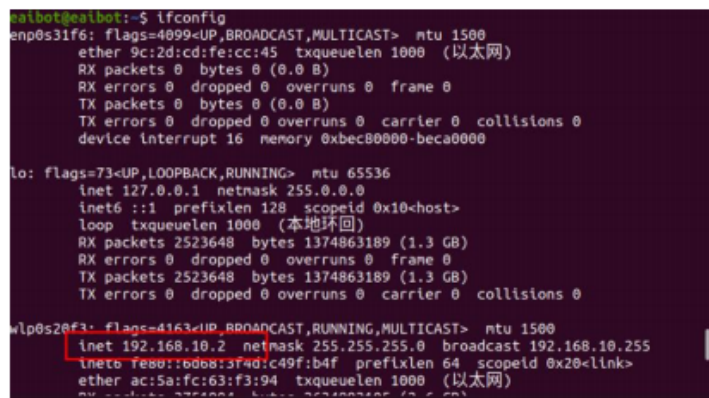
1) 连接无人机WIFI之后，运行测试连接的py程序如果出现OS ERROR，基本都是ip地址的问题，根据下图中的方法，检查自己LOCAL_IP_STR中的字符串是否无连接的无人机的ip地址相符合

机的连接。在代码编辑界面编写以下代码，

```
1. from robomaster import robot
2. import robomaster
3. if __name__ == '__main__':
4.     robomaster.config.LOCAL_IP_STR = "192.168.10.2"
5.     tl_drone = robot.Drone()
6.     tl_drone.initialize()
7.     version = tl_drone.get_sdk_version()
8.     print("SDK version:{0}".format(version))
9.     tl_drone.close()
```

其中第四行处本地 ip 地址需要手动获取，打开终端，输入 ifconfig 命令即

可查看本机 ip，如图 3 所示。



```
raibot@raibot:~$ ifconfig
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 9c:2d:cd:fe:cc:45 txqueuelen 1000 (以太网)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xbec00000-beca0000

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 (本地环回)
    RX packets 2523648 bytes 1374863189 (1.3 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2523648 bytes 1374863189 (1.3 GB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.2 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::b0b8:3f4d:c49f:b4f prefixlen 64 scopeid 0x20<link>
    ether ac:5a:fc:63:f3:94 txqueuelen 1000 (以太网)
    RX packets 2523648 bytes 1374863189 (1.3 GB)
```

图 3 查看本机 ip 地址

代码编写完成后保存并退出，在终端中执行该代码，输出飞机 SDK 版本的信息则代表已经连接上无人机。

EP 机器人的连接方式与无人机类似，它的电源开关位于电池上，如图 3 所

2) 无人机的坐标系是右手系，遵照x正向朝前，y正向朝右，z正向朝上的方向来确定，四个控制杆量abcd分别对应绕x轴旋转的roll（滚转角）绕y轴旋转的pitch（俯仰角），上升量（油门），以及绕z轴旋转的yaw（偏航角）。因此a为正数时，无人机绕x轴逆时针旋转，表现为无人机左飞，b为正，绕y轴逆时针旋转，表现为向前飞，c为正，表现为无人机上升，d为正，表现为无人机逆时针旋转，反之可同理类推。

2.ep机器人控制

1) 机器人控制使用的是robot中的Robot类的实例化对象ep_robot,具体位置控制使用的是Robot类的chassis成员变量,实例化为ep_chassis,需要完成的开火控制任务同样需要实例化一个blaster,即

```
ep_blaster = ep_robot.blaster
```

之后才能使用这个blaster里面的fire方法。

2) 不能在连接无人机WiFi的同时让机器人和电脑组网,如果要连接ep机器人,需要先连接vicon的wifi,而不是无人机的,所以完成空地协同任务在不使用特殊方法的前提下至少需要两台笔记本电脑。

3) 初始化记得填SN号以唯一确定控制的机器人,因为扫码的意思是让机器人连进你的局域网(应该是vicon),所以同时有几台机器人在连接,需要通过序列号唯一确定。

```
ep_robot.initialize(conn_type="sta", sn="XXXXXXXXXXXX")
```

实验二 动作捕捉系统与协同灭火

1.Vicon系统

1) vicon_bridge的安装

实验指导书里面已经有详细说明,在这里简要介绍一下原理。vicon_bridge是一个主要由cpp编写的ros功能包,需要通过catkin_make进行编译安装为可执行文件,通过vicon_bridge中的launch文件来执行,即

```
catkin_make
source ~/vicon_ws/devel/setup.bash
roslaunch vicon_bridge vicon.launch
```

其中source一行的意思是,catkin编译出的可执行文件在devel文件夹下,setup.bash则是一个配置文件,告诉终端有什么可执行文件,只有终端知道有什么可执行文件之后,才能通过roslaunch方法来执行,否则就会报类似Unknown xxx vicon_bridge之类的错。在每个新终端要运行这个vicon_bridge之前,都要source一次。记得改launch里面的ip地址,不然也不能正常运行。

2) 刚体的创建和跟踪

在创建自己的刚体之前,需要先取消掉左上角(如果存在)的其他刚体的勾选,否则将无法看到自己的小球的点,按住ctrl再用鼠标点击自己车或无人机的三个点就可以在左下角创建自己的刚体并进行追踪,建议统一命名为"car_编号"和"drone_编号",如"car_1"和"drone_D4A4"等。创建完自己的刚体之后再把别的同学的刚体重新勾选。

3) 刚体的消失

三个小球中只要有一个的位置发生了变化,就会被vicon试做刚体失效,无法发布其位置,需要重新执行(2)中的创建过程,后面会提到这个问题所导致的一种抽象bug。

ros系统

1) 一些简单的操作已经在文档中有所提及，这里再次强调几个比较重要或方便的操作

- `ctrl + shift + c` 和 `ctrl + shift + v` 是终端中的复制和粘贴操作，主要是因为 `ctrl + c` 是终端的普通暂停操作，所以要加 `shift`，顺带一提终端里没有 `ctrl+v` 的操作，在终端外的复制粘贴和终端内不同。
- `TAB` 键是终端的补全键，敲出 `sou` 后再按 `tab` 键就会自动补全为 `source,rosl` 加 `tab` 补全为 `roslaunch` 等，熟悉起来可以大大加快终端操作的速度。

2) 简单介绍本次实验中设计的一些 `ros` 操作:

- `rospy.init_node('listener', anonymous=True)` # 创建了一个匿名的名为“listener”的 `ros` 节点，其中 `anonymous` 允许在同一个局域网内多个同名节点不串台，否则大家都叫 `listener` 的话会出现消息乱飞的情况。
- `rospy.Subscriber('/vicon/tt/tt', TransformStamped, callback_tt)` 创建了一个 `ros` 的订阅者，这个订阅者订阅的话题名称为“/vicon/tt/tt”，话题的消息类型为“`TransformStamped`”，回调函数名称为“`callback_tt`”。在运行过程中，如果这个订阅者（`subscription`）收到了话题名为“/vicon/tt/tt”的消息，就会执行回调函数。

空地协同中可能遇到的问题及解决方法

1) 坐标系问题：

正面对讲台时，`vicon` 系的 `x` 轴正方向向右，`y` 轴正方向朝前。对于 `ep` 机器人，它的前方是贴着序号的那个方向，`x` 正方向朝前，`y` 正方向朝右（说明机器人是左手系，`vicon` 是右手系），先搞清楚坐标系方向，想清楚自己的 `x_val` 和 `y_val` 应该如何反映两者之间的位置变换关系才能让机器人正确位移。一般情况下完成验收可以通过固定车的初始方向来实现。（如果想挑战自己，也可以通过获取车的姿态来实现任意初始方向下的空地协同，需要查清楚 `TransformStamped` 这个消息类型的结构体的具体组成方式，以及使用 `numpy` 或者 `Eigen` 库进行旋转变换，或者手写四元数旋转矩阵之类的）

2) `undefine` 问题：

在通过回调函数获取坐标时，有时候会出现某个在回调函数中声明为 `global` 的变量未定义的问题，例如 `undefined variable x_tt` 之类的。

`undefine` 问题出现的原因是：`main` 函数中使用的全局变量是在回调函数中声明的，如果话题没有被发出或者订阅者的话题名称不对，回调函数未被触发，就会导致变量未定义的问题发生。

- 首先检查自己是否连接了 `vicon` 的 `WiFi`
- 接着检查自己是否已经在终端中运行了 `vicon_brigde`
- 然后检查自己订阅者中的话题名称是否和通过 `rostopic list` 看到的话题名称完全一致
- 再通过 `rostopic echo 你的话题名` 来观察无人机和车的两个 `vicon` 话题是否被正常发出

如果以上都检查完毕，程序应该可以正常运行。

前面提到了小球位置变化会，导致刚体被销毁，`vicon` 无法正常跟踪，因此无法发出话题。双面胶的粘性不足，队员或者自己的小手不干净，车或者无人机被拿出了 `vicon` 的相机视野范围都有可能造成刚体被摧毁，需要重新创建。

3) 可能存在的其他问题

- 无法控制车：请检查自己的 `wifi` 是否是 `vicon`，初始化中是否使用序列号，然后再重新扫码连接。

- 无法控制无人机：如果出现OS ERROR，请检查自己的ip地址，不同时间同一个无人机WiFi的ip也可能不同，请使用**ifconfig**检查，一般ip应该是192.168.10.x（x是个位数，如果出现255那就是失联了，超过10也不太对劲）；如果是正常启动了，显示takeoff了，但是出现什么**resp error**之类的问题，目前未查明原因，但是拆掉无人机的电池再换块装回去强制重启可以解决，车遇到一些看不懂的问题也可以试一下重启车。

祝大家实验顺利

2025 Summer © String All rights reserved.