|  |
| --- |
|  |
|  |
|  |
| 无人船软件说明书 |
|  |
|  |
|  |
|  |
|  |
| 单位：杰芯科技 |
| 2017年7月17日 |

目 录

[1 引言 1](#_Toc18330)

[2 启动4G 1](#_Toc15117)

[3 GPS和姿态传感器 2](#_Toc10192)

[4 摄像头 2](#_Toc14433)

[5 执行器 3](#_Toc15765)

[6 无人船三点竞速和无人船定点锚定 4](#_Toc5875)

[7 与上位机通信 4](#_Toc26333)

[8 开机启动 4](#_Toc24127)

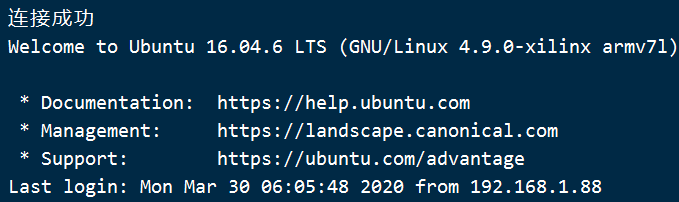
[9 无人船文件夹说明 6](#_Toc4390)

# 引言

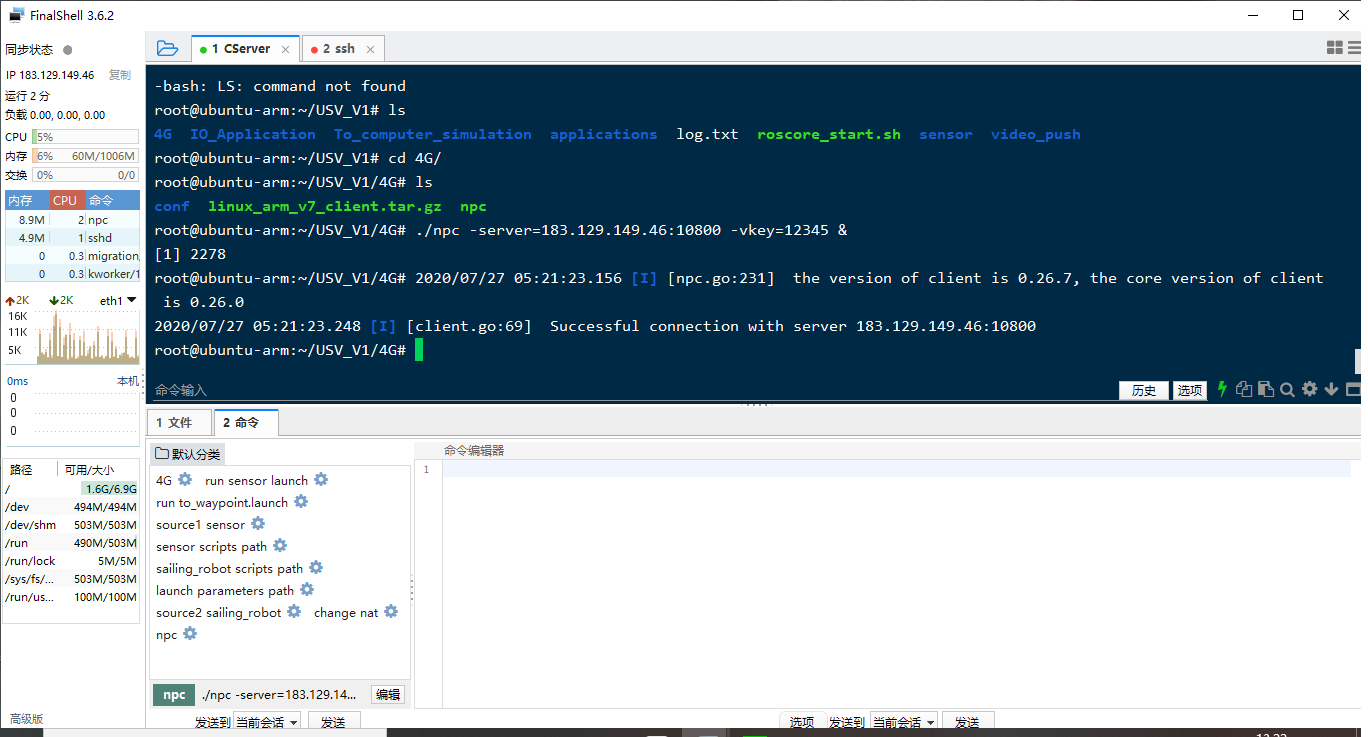
在进行无人船软件操作前，请确认已按照《无人船套件安装说明书》正确安装各类模块,本产品推荐安装Ubuntu16.04的linux系统和ROS（Kinetic版本，机器人操作系统）。整个无人船的控制程序都是基于ROS开发。本教程将对无人船的软件操进行说明。

# 启动4G

确保无线模块的网口和HCU-311模块的网口1（LAN1）相连，然后通过电池对HCU-311无人船模块进行上电，等HCU模块初始化完毕即RUN开始灯闪烁后，PC机连接名称为USR-\*\*\*\*的WIFI,初始密码为www.usr.cn，然后SSH进控制器（IP：192.168.8.100 Port:22），成功进入控制器后的界面如图所示：



然后进入/root/USV\_V1/4G目录,通过./npc -server=183.129.149.46:10800 -vkey=12345，启动启动4G，运行后界面如图所示。这样之后，就可以通过4G访问控制器了。



这里简单说明下原理，首先我们搭建了具有公网IP的中控服务器，通过NPS(一款轻量级，高性能，功能强大的内网穿透代理服务器，更多关于NPS的详细介绍可查看https://github.com/ehang-io/nps)将控制器的端口和IP映射到具有公网IP的中控服务器上，这样我们通过访问中控服务器的指定的端口就可以进入控制器或者进行其他操作了，关于端口含义请查看我们提供的《NPS内网穿透对照表》，如：

表1NPS内网穿透对照表

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 外网端口 | 设备（HCU）穿透端口 | 功能 |
| 2 | 10800 |  | NPS握手 |
| 3 | 10701 | 22 | 4G SSH |
| 4 | 10702 | 702 | TCP（HCU\_server） |
| 5 | 10703 | 703 | TCP(视频) |

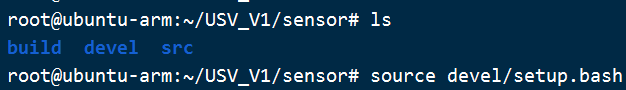
# GPS和姿态传感器

GPS和姿态传感器的数据解算程序，写成了node（节点），通过launch文件同时启动，此launch文件的名称为Alinx.launch,所在目录如图所示:

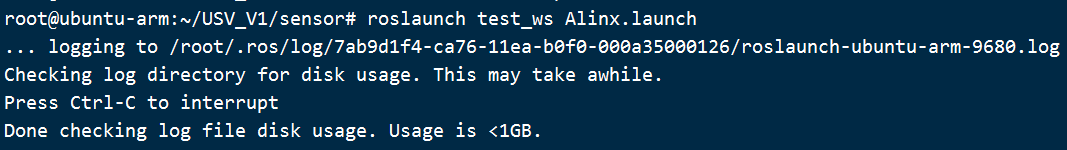


执行此launch文件的过程如下

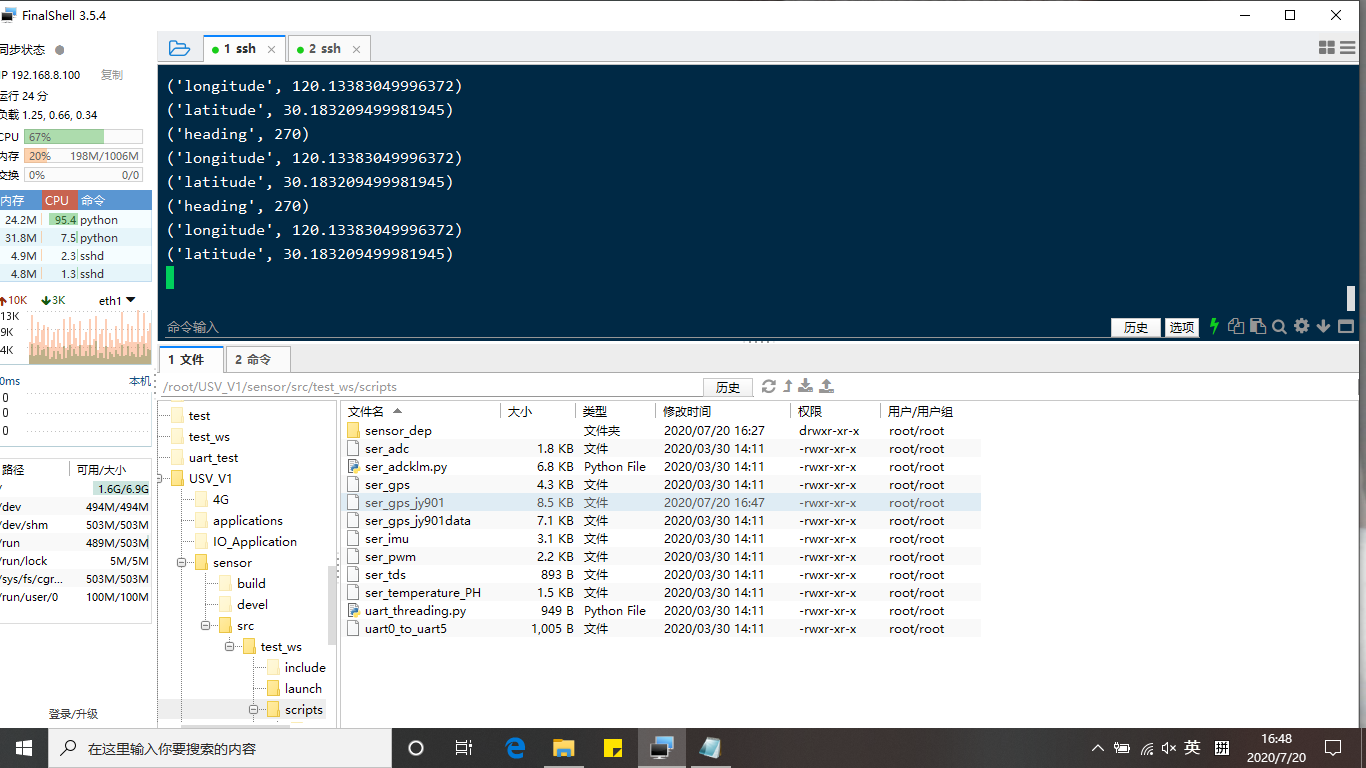
1 进入/USV\_V1/sensor目录source工作空间，如图所示



2 通过 roslaunch test\_ws(包名) Alinx.launch(文件名)执行此文件，如图所示

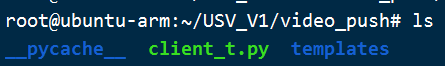


3 执行结果如图所示，终端打印出经纬度和偏航角数据



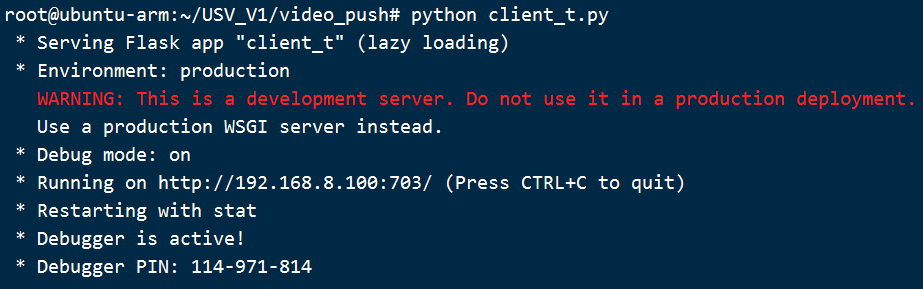
# 摄像头

摄像头为USB驱动设备，通过USB接口插入摄像头，进入/USV\_V1/video\_push目录，与摄像头和视频传输有关的程序，放在client\_t.py中，如图所示:

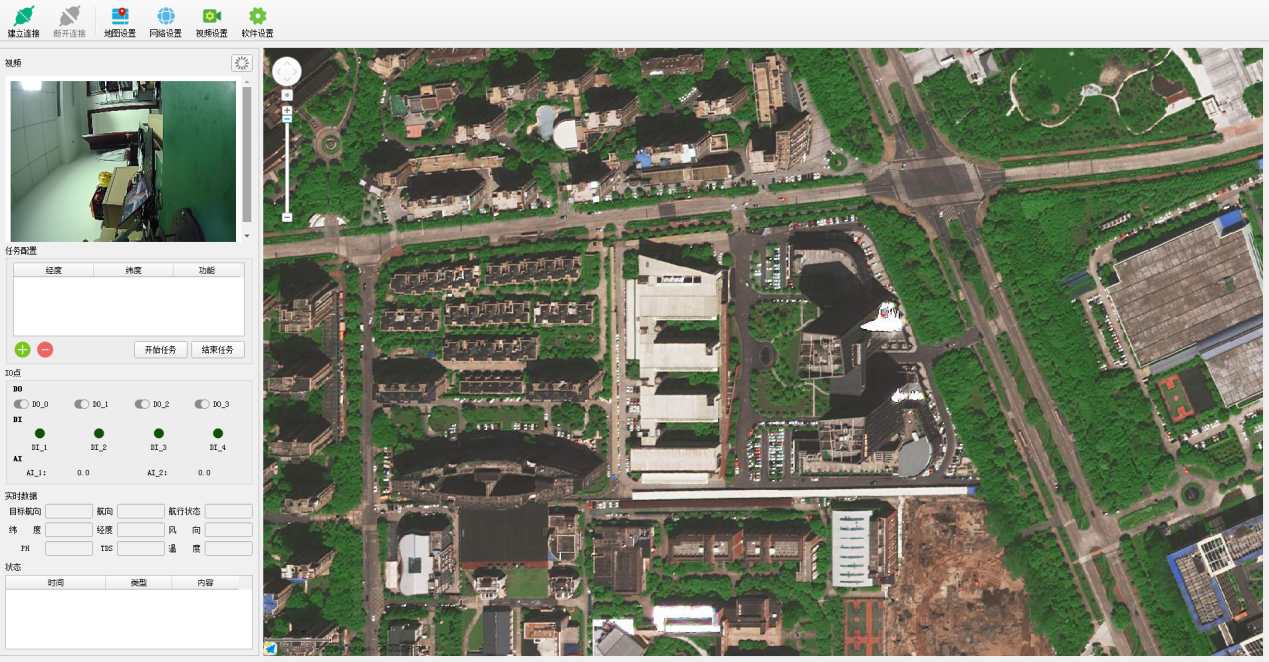


摄像头视频传输操作如下：

1 通过 python client\_t.py运行此程序，如图所示



2 打开上位机，按要求配置好IP(183.129.149.46)和Port(10703)并连接，结果如图所示

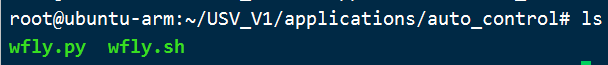


# 执行器

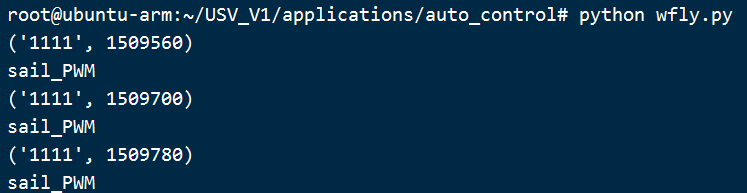
无人船的执行器为控制方向的舵机和提供前进动力的两个电机，我们通过遥控器来控制两者的输出。原理为，HCU控制器通过PI端口读取控制器的PWM数据，然后通过PO端口输出。

执行器的操作过程如下：

1 进入:USV\_V1/applications/auto\_control目录，如图所示：



2 通过python wfly.py运行此程序，结果如图所示：

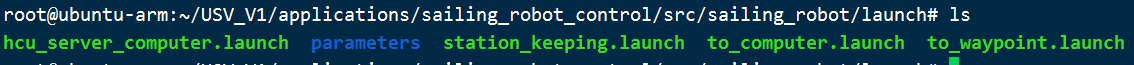


3拨动遥控器手柄观察执行器运行情况

# 无人船三点竞速和无人船定点锚定

与无人船的三点竞速和定点锚定功能有关的节点，已写成了launch文件的形式，分别是to\_waypoint.launch和station\_keeping.launch，此程序放在

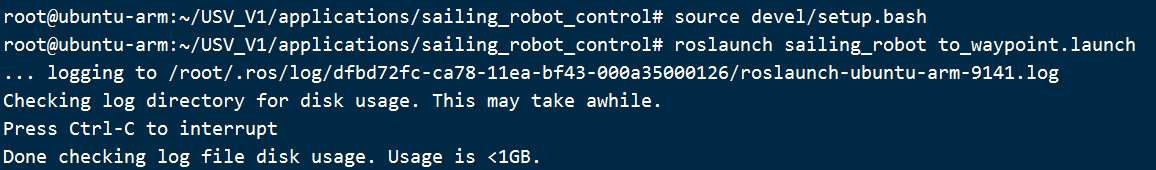
/USV\_V1/applications/sailing\_robol，如图所示：



实现无人船三点竞速或者虚拟锚定功能操作

1 执行传感器Alinx.launch,过程和上文第3部分相同

2 进入USV\_V1/application/sailing\_robot\_control执行to\_waypoint.launch 或者station\_keeping.launch，如图所示：



# 与上位机通信

与上位机通信的程序，写在了一个名叫hcu\_to\_computer\_server的节点中，如果你想通过上位机无人船进行操作，如下达任务，结束任务等，则此节点比需要运行起来。为了方便于操作，我们将所有通过上位机操作无人船有关的程序都放到hcu\_server\_computer.launch文件，即只要启动hcu\_server\_computer.launch就可以通过上位机控制无人船了，执行此launch文件操作同上。

1 执行传感器Alinx.launch

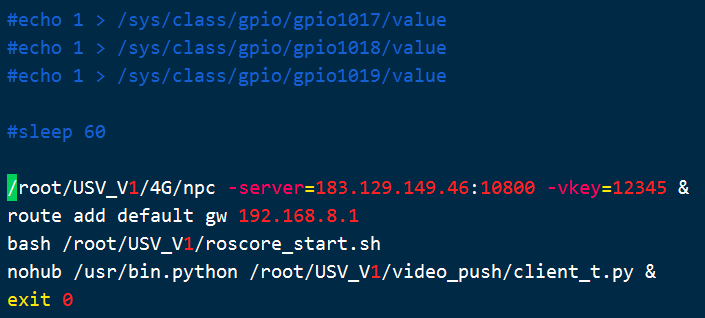
2 执行hcu\_server\_computer.launch

# 开机启动

有些每次开机都需要执行的程序，可以放到开机启动之中，如启动4G，启动roscore等具体操作如下，

1 通过 vi /etc/rc.local打开此文件，

2 将需要开机执行的指令写入其中，如4G，设置默认路由，roscore，视频传输等，如图所示：

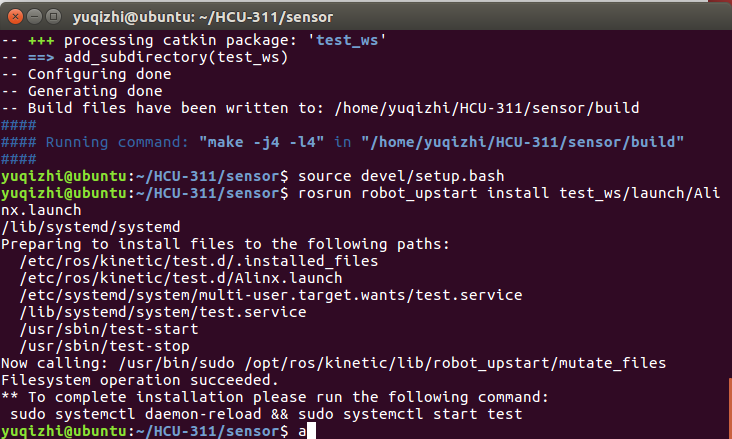


最后可以将传感器的Alinx.launch文件让其开机启动:

1 安装一个功能包 sudo apt-get install ros-kinetic-robot-upstart

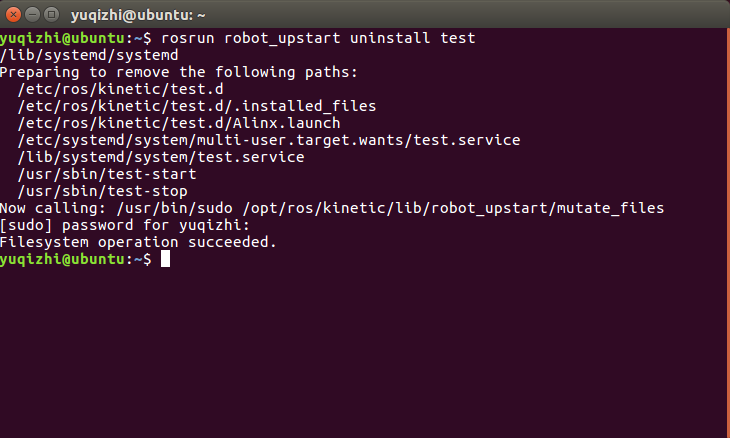
2 设置某个Package的launch文件为开机启动

rosrun robot\_upstart install+launch，如图所示



3.停止launch开机启动

rosrun robot\_upstart uninstall \*\*\*(启动成功后的名称)，如图所示



# 无人船文件夹说明

HCU-311控制器中的USV\_V1文件夹明细如图所示：



USV\_V1主文件夹下文件作用如下：

4G 为NPS的客服端程序，即内网穿透脚本

Applications 为无人应用案例程序，包括三点竞速和定点锚定

IO\_Application为DI,DO,AI,AO,PI,PO等端口使用示例

sensor为各种传感器数据解算程序

To\_computer\_simulation 为无人船仿真程序

video\_push为视频传输程序

roscore\_start.sh为roscore自启动脚本

无人船软件github网址：https://github.com/supcon-nzic/USV