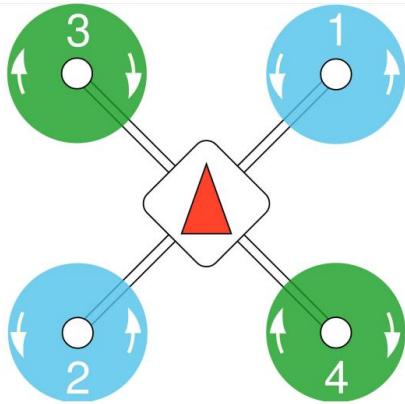


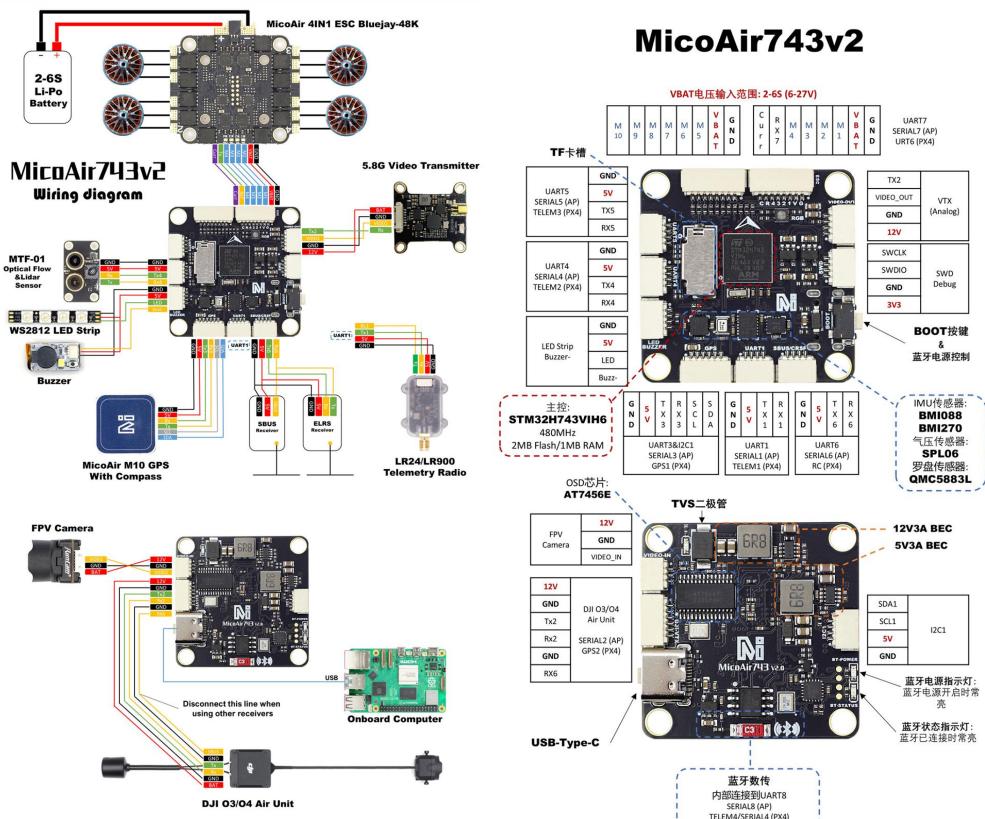
一、飞控安装与接线

电机顺序：



飞控箭头与机头同向为正向，图示为电机序号

将每个电机的三根线与微空飞塔电调部分的电机焊盘连接，注意适当裁切线材长度，不得过长或者过短，接线如下图所示。



图中有几个点我们需要注意一下的就是，这个 MTF 光流模块，我们需要按照一定方向去摆放，MTf - 02 光流模块的正面有一个 PMW3901 的传感器，我们让传感器的缺口朝机尾，然后四合一电调的电源接口我们需要接上附带的 XT60 接口和线，注意也要焊接上附带的滤波电容（一定要接上）然后图上的树莓派接线我们使用 typec 接上我们的香橙派，其他传感器忽略掉。

飞机结构中，按照我们附赠的无人机结构打印件的 STL 文件自行使用 3D 打印机打印模型，然后安装即可（按照我们示范无人机）



按照图示布局安装。

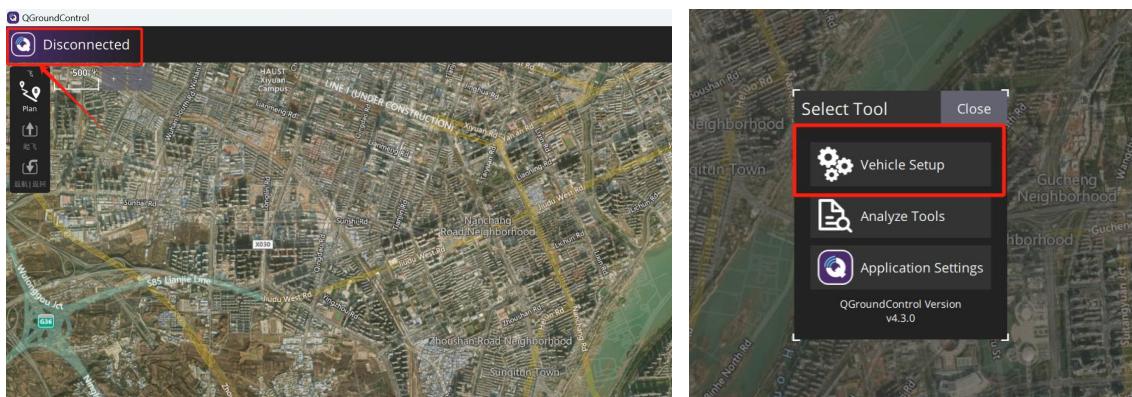
安装事项：不要把 5V UBEC 模块的电源线靠近飞控布局，因为很可能会影响飞控的磁罗盘和陀螺仪等传感器正常工作。电调的电源总线一定要加滤波电容，防止上电瞬间波峰烧毁芯片。

二、飞控基本设置（Groundcontrol 地面站）

1. 固件烧录：

我们统一使用 PX4 固件，版本使用 v1.14.3 的固件，这个版本的参数设置会方便一些且稳定，我们的资料包里面提供了固件文件。

我们打开 QGroundcontrol（简称 QGC），然后这个时候飞控和电脑先别连接，先点开左上角的 Q 图标，然后点击 Vehicle Setup，然后我们选择固件一栏，然后我们这个时



候将飞控接入电脑，然后这个时候 QGC 会自动识别飞控，选择高级设置，自定义固件，然后点击确认，然后这个时候选择我们资料的固件烧录进去。

2. 基本参数设置：

逐个设置遥控器，电机，电源，还有机架设置这些，将飞机传感器设定一下。这一部分详细过程参照 <https://docs.px4.io/main/zh/index.html>，按照这里把遥控器，电机，电源，还有机架等等先设置好。

特定参数设置： EKF2_AID_MASK 请设置成 24 或者 26（如果加了光流），
EKF2_HGT_MODE 请设置成 vision 这个选项。遥控器的话，请把五通道设置成模式开关，
把通道设置成紧急开关。

修改 dshot_config 为 dshot600

修改 CBRK_SUPPLY_CHK 为 894281 执行这步跳过了电源检查，因此左侧栏的电
池设置部分就算是红
的也没关系

修改 CBRK_USB_CHK 为 197848

修改 CBRK_IO_SAFETY 为 22027

修改 SER_TEL1_BAUD 为 921600

修改 SYS_USE_IO 为 0（搜索不到则不用管）

上电前请先用万用表通断档检测电源正负焊点是否短接，强烈建议第一次上电前先接一
个短路保护器

检测电机转向前确保没有安装螺旋桨！！！

修改电机转向：进入 mavlink 控制台

dshot reverse -m 1

dshot save -m 1

修改 1 为需要反向的电机序号

第一次试飞请务必找有自稳模式下飞行经验的飞手协助，只飞过大疆无人机的飞手 99%无
法飞好！

三、机载电脑环境配置

1. 镜像下载

点击 [Ubuntu_免费高速下载|百度网盘-分享无限制](#) 下载香橙派 5MAX 官方镜像。

使用 balancer 软件烧录进内存卡，然后将内存卡插入香橙派电脑。

笔记本上也需要安装 ubuntu，推荐装 20.04 版本。虚拟机或双系统都可以，如果有长期学习打算推荐双系统

2. ROS 环境配置

ros1 安装：

感谢鱼哥提供的一键 ros 安装工具：在 ubuntu 终端输入

```
wget http://fishros.com/install -O fishros && . fishros
```

然后选择安装 ROS 桌面端，选择 ROS1-noetic 版本，中途如果提示要换源的时候，选择还原并清除第三方源。

PIP 安装（选装）

终端输入：

```
python3 --version
```

```
sudo apt install python3-pip python3-dev build-essential libssl-dev libffi-dev
```

```
python3-setuptools
```

```
pip3 --version
```

输入法设置（选装）

```
sudo apt-get install ibus ibus-clutter ibus-gtk ibus-gtk3 ibus-qt4
```

```
im-config -s ibus
```

```
sudo apt-get install ibus-pinyin
```

```
ibus-setup
```

swap 交换储存设置（很重要）

1.新增 swapfile 文件大小自定义 (12G)

```
sudo fallocate -l 12G /var/swapfile
```

2.配置该文件的权限

```
sudo chmod 600 /var/swapfile
```

3.建立交换分区

```
sudo mkswap /var/swapfile
```

4.启用交换分区

```
sudo swapon /var/swapfile
```

5.自启动启用

```
sudo bash -c 'echo "/var/swapfile swap swap defaults 0 0" >> /etc/fstab'
```

mavros 安装 (PX4 飞控通信包)

MAVROS 是一层 MAVLink 与 ROS 通信的封装，旨在方便无人机与机载电脑通信。

```
sudo apt-get install ros-noetic-mavros*
```

```
sudo apt-get install ros-noetic-serial
```

```
cd /opt/ros/noetic/lib/mavros
```

```
sudo ./install_geographiclib_datasets.sh
```

接上飞控后，查看 mavros 对应的串口号：

```
ls /dev/tty*
```

给予相应的权限：

```
sudo chmod 777 /dev/ttyACM0
```

修改 launch 文件：

```
sudo chmod 777 /opt/ros/noetic/share/mavros/launch/px4.launch
```

```
sudo gedit /opt/ros/noetic/share/mavros/launch/px4.launch
```

将其中的

```
<arg name="fcu_url" default="/dev/ttyACM0:57600" />
```

修改为自己的端口及波特率

```
<arg name="fcu_url" default="/dev/ttyACM0:921600" />
```

启动 Mavros

```
roslaunch mavros px4.launch
```

Realsense SDK 安装（双目相机驱动包）

1. 确保未连接任何 RealSense 设备

2. 打开终端，运行：

```
wget
```

```
https://github.com/IntelRealSense/librealsense/raw/master/scripts/libuvc_installati
```

```
on.sh
```

```
chmod +x ./libuvc_installation.sh
```

```
./libuvc_installation.sh
```

```
realsense-viewer
```

这个时候如果成功弹出：realsense 的相机界面说明安装成功

导航程序包下载

下载 ego-planner 源码并编译

```
git clone https://github.com/ZJU-FAST-Lab/Fast-Drone-250
```

```
cd Fast-Drone-250
```

安装 ceres 与 glog 与 dynamic-reconfigure

解压 3rd_party.zip 压缩包

```
unzip 3rd_party.zip
```

进入 glog 文件夹打开终端

```
./autogen.sh && ./configure && make && sudo make install
```

```
sudo apt-get install liblapack-dev libsuitesparse-dev libcxsparse3 libgflags-dev
```

```
libgoogle-glog-dev libgtest-dev
```

(这里我强调一下，官方是 libcxsparse3.1.2，我们要安装 libcxsparse3 才可以)

进入 ceres 文件夹打开终端

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
sudo make -j4
```

```
sudo make install
```

更改 IMU 话题频率（非常重要，如果没更改会导致 VINS 发散）

进入到工作空间下 shfiles 文件夹，打开 rpx4.sh，在 roslaunch mavros px4.launch 之后的& sleep 改成 10（等待 10s，给启动节点预留足够时间）然后后面添加两行：

```
rosrun mavros mavcmd long 511 105 4000 0 0 0 0 0 & sleep 3;
```

```
rosrun mavros mavcmd long 511 31 4000 0 0 0 0 0 & sleep 3;
```

在 Fast-Drone-250 文件夹打开终端

```
sudo apt-get install ros-noetic-ddynamic-reconfigure
```

```
catkin_make
```

```
source devel/setup.bash
```

```
roslaunch ego_planner single_run_in_sim.launch
```

在 Rviz 内按下键盘 G 键，再单击鼠标左键以点选无人机目标点（这个是仿真测试）

如正常运行，说明导航程序包已经成功安装

ros 分布式通讯（必须要，因为 RK3588 对 rviz 兼容不太好）

在 orangepi 终端上

ifconfig (查看 IP)

hostname (查看计算机名称)

在 orangepi 上： ping 192.168.**.**

sudo gedit /etc/hosts

如果没有 gedit 请在文件管理器找到这个文件然后 sudo chmod 777 hosts，不允许 sudo chmod 777 *不然会导致 sudoer 出问题。

加上一行： 192.168.**.** 虚拟机用户名

ping 虚拟机用户名

在虚拟机上： ping 192.168.**.** (ping 一下香橙派)

sudo vim /etc/hosts

加上一行： 192.168.**.** orangepi

ping orangepi

配置主机的 IP 地址：

orangepi 端输入以下命令配置.bashrc 文件：

sudo vim ~/.bashrc

在最后一行加入以下代码：

```
export ROS_MASTER_URI=http://主机 IP:11311
```

```
export ROS_HOSTNAME=主机 IP
```

```
: wq 保存退出
```

```
source ~/.bashrc
```

配置从机 IP

我这里从机是我的笔记本电脑， 所以我在虚拟机端输入以下命令配置.bashrc 文件：

```
sudo vim ~/.bashrc
```

在最后一行加入以下代码：

```
export ROS_MASTER_URI=http://主机 IP:11311
```

```
export ROS_HOSTNAME=从机 IP
```

```
: wq 保存退出
```

```
source ~/.bashrc
```

四、常用工具下载（选择性下载）

Plotjuggler:

```
sudo apt install ros-noetic-plotjuggler
```

```
sudo apt install ros-noetic-plotjuggler-ros
```

```
roslaunch plotjuggler plotjuggler
```

Net-tools:

```
sudo apt install net-tools
```

```
ifconfig
```

ssh:

```
sudo apt install openssh-server
```

在笔记本上： ping 192.168.**.**

sudo gedit /etc/hosts

加上一行： 192.168.**.** fast-drone

ping fast-drone

ssh fast-drone@fast-drone (ssh 用户名@别名)

五、Ego-Planner 代码框架与参数介绍

src/planner/plan_manage/launch/single_run_in_exp.launch 下的：

map_size : 当你的地图大小较大时需要修改，注意目标点不要超过 map_size/2

fx/fy/cx/cy : 修改为你的深度相机的实际内参

max_vel/max_acc : 修改以调整最大速度、加速度。速度建议先用 0.5 试飞，最大不要超过

2.5，加速度不要超过 6

flight_type : 1 代表 rviz 选点模式，2 代表 waypoints 跟踪模式

src/planner/plan_manage/launch/advanced_param_exp.xml 下的：

resolution : 代表栅格地图格点的分辨率，单位为米。越小则地图越精细，但越占内存。

最

小不要低于 0.1

obstacles_inflation : 代表障碍物膨胀大小，单位为米。建议至少设置为飞机半径（包括螺旋桨、桨保）的 1.5 倍以上，但不要超过 resolution 的 4 倍。如果飞机轴距较大，请相应改

大 resolution

src/realfight_modules/px4ctrl/config/ctrl_param_fpv.yaml 下的：

mass : 修改为无人机的实际重量

hover_percent : 修改为无人机的悬停油门, 可以通过 px4log 查看, 具体可以参考文档 如果

你的无人机是和课程的一模一样的话, 这项保持为 0.3 即可。如果更改了动力配置, 或重量发

生变化, 或轴距发生变化, 都请调整此项, 否则自动起飞时会发生无法起飞或者超调严重的情况。

gain/Kp,Kv : 即 PID 中的 PI 项, 一般不用太大改动。如果发生超调, 请适当调小。如果无人机响应较慢, 请适当调大。

rc_reverse : 这项使用乐迪 AT9S 的不用管。如果在第十一课的自动起飞中, 发现飞机的飞行方向与摇杆方向相反, 说明需要修改此项, 把相反的通道对应的值改为 true。其中 throttle

如果反了, 实际实验中会比较危险, 建议在起飞前就确认好, 步骤为:

```
roslaunch mavros px4.launch
```

```
rostopic echo /mavros/rc/in
```

打开遥控器, 把遥控器油门从最低满满打到最高

看 echo 出来的消息里哪项在缓慢变化 (这项就是油门通道值), 并观察它是不是由小变大

如果是由小变大, 则不需要修改 throttle 的 rc_reverse, 反之改为 true
其他通道同理

六、VINS 的参数设置与外参标定（非常重 要）

检查飞控 mavros 连接正常

ls /dev/tty* , 确认飞控的串口连接正常。一般是 /dev/ttyACM0

sudo chmod 777 /dev/ttyACM0 , 为串口附加权限

roslaunch mavros px4.launch

rostopic hz /mavros/imu/data_raw , 确认飞控传输的 imu 频率在 200hz 左右

检查 realsense 驱动正常

roslaunch realsense2_camera rs_camera.launch

进入 远程桌面 , rqt_image_view 查看 /camera/infra1/image_rect_raw ,
/camera/infra2/image_rect_raw , /camera/dept
h/image_rect_raw 话题正常

VINS 参数设置

进入 realflight_modules/VINS_Fusion/config/

驱动 realsense 后, rostopic echo /camera/infra1/camera_info , 把其中的 K 矩阵中
的

fx,fy,cx,cy 填入 left.yaml 和 right.yaml

在 home 目录创建 vins_output 文件夹(如果你的用户名不是 fast-drone , 需要修改
config 内

的 vins_out_path 为你实际创建的文件夹的绝对路径)

修改 fast-drone-250.yaml 的 body_T_cam0 和 body_T_cam1 的 data 矩阵的第

四列为你的无

人机上的相机相对于飞控的实际外参，单位为米，顺序为 x/y/z，第四项是 1，不用改

VINS 外参精确自标定

```
sh shfiles/rspx4.sh
```

```
rostopic echo /vins_fusion/imu_propagate
```

拿起飞机沿着场地尽量缓慢地行走，场地内光照变化不要太大，灯光不要太暗，不要使用会
频

闪的光源，尽量多放些杂物来增加 VINS 用于匹配的特征点

把 vins_output/extrinsic_parameter.txt 里的内容替换到 fast-drone-250.yaml 的
body_T_cam0 和 body_T_cam1

重复上述操作直到走几圈后 VINS 的里程计数据偏差收敛到满意值（一般在 0.3 米内）

建图模块验证

```
sh shfiles/rspx4.sh
```

```
roslaunch ego_planner single_run_in_exp.launch
```

进入远程桌面 rosrun ego_planner rviz.launch (这一块我不建议在香橙派直接使用
Rviz，因为 RK 系列芯片的可视化界面可能有点兼容性问题，建议还是用分布式通信来查看
Rviz 的界面)

七、导航飞行

自动起飞：

(注意：所有的操作建议都在虚拟机使用分布式通信连接香橙派，稳定一些)

```
sh shfiles/rspx4.sh
```

```
rostopic echo /vins_fusion imu_propagate
```

拿起飞机进行缓慢的小范围晃动，放回原地后确认没有太大误差

遥控器 5 通道拨到内侧，六通道拨到下侧，油门打到中位

```
roslaunch px4ctrl run_ctrl.launch
```

```
sh shfiles/takeoff.sh , 如果飞机螺旋桨开始旋转，但无法起飞，说明 hover_percent
```

参数过小；如果飞机有明显飞过 1 米高，再下降的样子，说明 hover_percent 参数过大

遥控器此时可以以类似大疆飞机的操作逻辑对无人机进行位置控制

降落时把油门打到最低，等无人机降到地上后，把 5 通道拨到中间，左手杆打到左下角上

锁

Ego-Planner 实验

自动起飞

```
roslaunch ego_planner single_run_in_exp.launch
```

```
sh shfiles/record.sh
```

然后打开 Rviz

```
roslaunch ego_planner rviz.launch
```

按下 G 键加鼠标左键点选目标点使无人机飞行

如果实验中遇到意外怎么办！！！

case 1：VINS 定位没有飘，但是规划不及时/建图不准确导致无人机规划出一条可能撞进障碍

物的轨迹。如果飞手在飞机飞行过程中发现无人机可能会撞到障碍物，在撞上前把 6 通道拨回

上侧，此时无人机会退出轨迹跟随模式，进入 VINS 悬停模式，在此时把无人机安全着陆即可

case 2 : VINS 定位飘了，表现为飞机大幅度颤抖/明显没有沿着正常轨迹走/快速上升/快速下

降等等，此时拨 6 通道已经无济于事，必须把 5 通道拨回中位，使无人机完全退出程序控制，

回到遥控器的 stabilized 模式来操控降落

case 3 : 无人机已经撞到障碍物，并且还没掉到地上。此时先拨 6 通道，看看飞机能不能稳

住，稳不住就拨 5 通道手动降落

case 4 : 无人机撞到障碍物并且炸到地上了：拨 5 通道立刻上锁，减少财产损失

case 5 : 绝招 反应不过来哪种 case，或者飞机冲着非常危险的区域飞了，直接拨 7 通道紧急

停桨。这样飞机会直接失去动力摔下来，对飞机机身破坏比较大，一般慢速情况下不建议。