



**AMOVLAB**

— 阿木实验室 —

产品使用手册

Z410-4B

Ver. 20220425

## 目录

1. 资料下载链接 .....	4
2. Z410 机型介绍 .....	4
3. 如何快速进行第一次解锁 .....	6
4. 如何进行第一次试飞(新手初次请勿在室内进行飞行) .....	7
4.1 Z410 飞行前的准备: .....	7
4.2 遥控器介绍及自稳模式飞行: .....	7
4.3 定高, 悬停, 返航模式飞行: .....	7
4.4 降落, 绕圈, 有头无头及结束飞行操作: .....	7
5. 使用这台无人机需要具备的一些基础知识: .....	7
5.1 锂电池的正确使用方法和充电操作 .....	7
5.2 BB 响低电压报警器的使用 .....	8
5.3 数传模块的使用(基础版机型配数传) .....	8
5.4 Pixhawk 飞控的基本调试、地面站的使用参考教程 .....	8
5.5 飞行模式的含义和正确使用 .....	9
5.6 Pixhawk 飞控不能解锁常见问题汇总 .....	9
5.7 Ubuntu 系统的基本操作 .....	9
6. 二次开发入门教程 .....	9
7. Z410-4B 单目版教程 .....	9
7.1 建立网络连接 .....	10
7.2 使用树莓派遥测转发功能, 将无人机与地面站进行连接 .....	10
7.3 使用树莓派的图传功能, 可实时回传视频到地面站 .....	11
7.4 PC 访问树莓派的 3 种方式: .....	11
7.5 树莓派的关机操作 .....	12
7.6 使用 Dronekit 测试连接飞控并读取状态数据 .....	13
7.6.1 运行连接示例 .....	13
7.7 启动 SITL 仿真环境并运行 python 示例 .....	16
7.8 SITL 仿真结合 MP 地面站运行 python 示例 .....	19

7.9 使用 Python 脚本控制无人机飞行 .....	21
8. 二维码识别降落例程 .....	23
9. Z410 视觉版教程 .....	25
9.1 一些常用的 T265 命令 .....	25
9.2 启用 T265 定位的飞控参数设置 .....	26
9.3 使用 T265 定位导航的 2 种方法 .....	28
9.3.1 使用 pyrealsense2 启动双目相机 .....	28
9.3.2 ROS 系统下启用 T265 .....	31
9.3.3 开机自启 T265 双目相机教程 .....	34
9.3.4 GPS 定位与 T265 定位的快速切换设置教程 .....	34
10. Z410-4B 无人机扩展 4G 模块实现 4G 图数传功能 .....	35
10.1 Windows 系统下安装 Tailscale .....	36
10.2 无人机(树莓派)加入 Tailscale 虚拟局域网 .....	40
10.3 重新分配 UDP 端口, 获取遥测数据和高清视频 .....	43
11. 手机端 QGC 地面站如何通过 4G 网络连接无人机 .....	43
12. 免责声明与安全飞行 .....	44
12.1 简介 .....	44
12.2 免责声明 .....	44
12.3 产品使用注意事项 .....	45
12.4 安全须知 .....	46
12.5 售后服务条款 .....	47



## 1. 资料下载链接

Missionplanner1.3.77 地面站软件下载链接:

[https://pan.baidu.com/s/140DpacnRj8bXi0EHN\\_SiDA](https://pan.baidu.com/s/140DpacnRj8bXi0EHN_SiDA)

提取码: 1qdg

注意! 电脑最好是使用 Win10 系统, Win10 以前的系统可能由于缺少必要的插件会导致启动失败。

Ardupilot (4.1.0) 固件下载链接:

<https://pan.baidu.com/s/1P2wxKxNYVlnWzeKraLoVBg>

提取码: a7cl

注意! 无人机发货前已经组装调试好, 不需要再下载固件, 此固件可留着备用。

注意! 推荐使用无线连接树莓派 wifi 来连接地面站软件, 尽量避免在有电池供电的情况下有线连接飞控, 长期使用可能导致 LED 损毁!

PC 端安装 Ubuntu 系统:

PC 端安装 Ubuntu 系统的目的是搭建 Ardupilot 开发环境, 从而使用 SITL 仿真。可参考以下教程安装虚拟机。

<https://mp.weixin.qq.com/s/Z46JL7gTzF73DnIJeKLDsA>

如果想自己熟悉搭建 Ardupilot 开发环境的流程, 可以安装纯净版 Ubuntu 系统, 以下是 ubuntu18.04 镜像下载链接

<https://pan.baidu.com/s/1vmkmHmBmPOP9FlinSW-nmg>

提取码: cnlw

以下是搭建 Ardupilot 开发环境的教程, 提供给大家作为参考:

<https://mp.weixin.qq.com/s/KhjxlFBz-kyexx03kgIFXQ>

注意! 国内由于网络原因, 搭建 Ardupilot 开发环境会遇到各种各样的不同问题, 使用时建议先更换 ubuntu 源和 PIP 源。遇到问题更多的需要到官网, 论坛或者利用搜索引擎来解决问题。

## 2. Z410 机型介绍

Z410 无人机是一款专为无人机二次开发入门设计的一款实验机型。

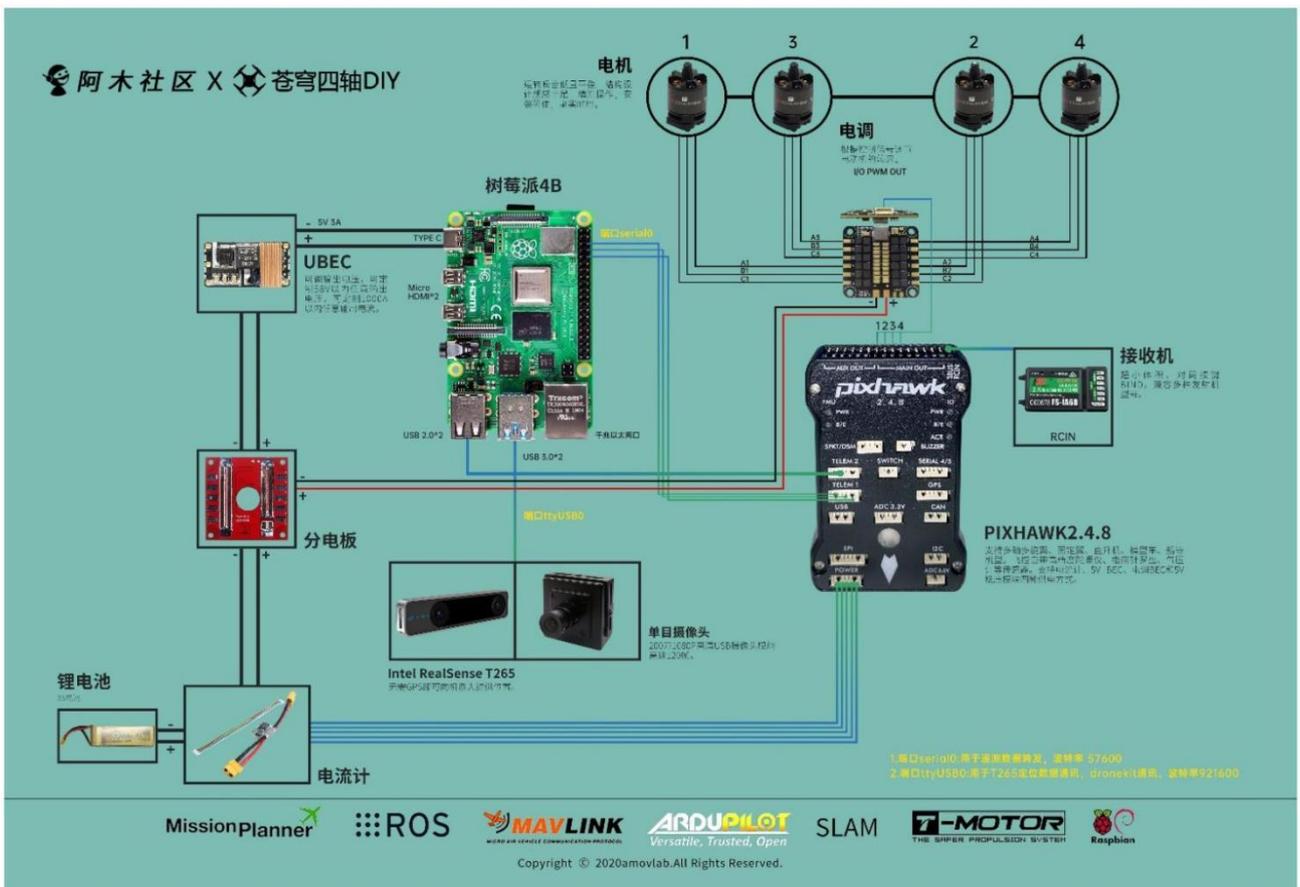
基础版采用 pixhawk2.4.8 飞控和 M8N GPS 模块；

单目版在基础板机型上安装了树莓派 4B 机载电脑，树莓派可通过 Dronekit-python、ROS、Opencv 编程实现对无人机的外部控制；

视觉版在单目版基础上安装了 intel T265 双目相机，可实现在无 GPS 环境下的定位导航。

树莓派系统基于 ubuntu20.04 打造，集成了多种功能模块：ROS、mavros、Dronekit-python、Opencv、Rpanion-server 等。大家不用再花费时间安装编译这些功能包，参考例程可直接上手学习。

## 2.1 Z410 硬件接线图



## 2.2 主要硬件设备及用途介绍

**飞控:** pixhawk2.4.8-- 无人机正常飞行的必要硬件，控制无人机姿态；

**树莓派:** raspberry pi 4B-- 运行外部控制程序及其他系统集成，给飞控发

送外部控制命令，或者网络信号；（单目版和视觉版配置）

**电调：**定制四合一电调—接收飞控的输出信号，处理后驱动电机旋转；

**电机：**T-motor2216—电机旋转带动桨叶，从而给无人机提供上升的动力；

**电池：**3S 5200mah—无人机的动力来源；

**电流计：**给飞控提供稳压电源；检测实时电压，当电池电压过低采取预设值的动作，自主降落或返航；

**UBEC：**给树莓派提供稳定的电源；（单目版和视觉版配置）

**T265 双目相机：**用于视觉定位；（单目版和视觉版配置）

**单目摄像头：**用于实时视频回传；（单目版和视觉版配置）

**接收机：**与遥控器配套，接收遥控器的控制信号，从而控制无人机飞行；

### 3. 如何快速进行第一次解锁

(1) 将无人机平放在地面上；

(2) 电池插上低压报警器（BB 响）测试电压（连接示意图如下），单片电池电压不低于 3.6V（**注意！低于 3.5V 容易造成过放，高于 4.2V 容易造成过充**），贴上魔术贴，绑扎在飞机底板上；



(3) 打开遥控器，将油门摇杆拉倒最低，SWA 和 SWD 开关均拨到 1 挡，解除遥控器报警；

(4) 将无人机接上电池，等待飞控自检。自检时间大概 30 秒左右。此时飞控会闪黄灯，电机会发出“滴~滴~”声；

(5) 长按安全开关，直到安全开关 LED 指示灯常亮。此时飞控 LED 会蓝灯闪烁，电机“滴~滴~”声停止；

(6) 遥控器左边油门摇杆往右下角一直按住大概 2 秒左右，蜂鸣器会长鸣一声，表示飞控已解锁。此时飞控蓝灯长亮，电机会怠速旋转；

(7) 推动油门超过 50%，电机会加速；

(8) 油门摇杆往左下角一直按住大概 2 秒左右，飞控会上锁。此时飞控蓝色指示灯闪烁；

基本的解锁上锁操作完成。

[https://v.youku.com/v\\_show/id\\_XNTE2NzQwODAxMg==.html](https://v.youku.com/v_show/id_XNTE2NzQwODAxMg==.html)

## 4. 如何进行第一次试飞（新手初次请勿在室内进行飞行）

新手请先参考 Z410 基础版室外飞行基本操作视频。了解遥控器的基本使用后再进行飞行。

**盲目的飞行会导致飞行器损坏甚至人身伤害，请务必注意安全！**

### 4.1 Z410 飞行前的准备：

[http://v.youku.com/v\\_show/id\\_XNDY3MjgxNDExNg==.html?x&sharefrom=android&sharekey=137370a19bac8fe70206f0d0ea3f26087](http://v.youku.com/v_show/id_XNDY3MjgxNDExNg==.html?x&sharefrom=android&sharekey=137370a19bac8fe70206f0d0ea3f26087)

### 4.2 遥控器介绍及自稳模式飞行：

[http://v.youku.com/v\\_show/id\\_XNDY3MjgwMjkONjA==.html?x&sharefrom=android&sharekey=9598e833b88a29b0cdbe40474ce8035a4](http://v.youku.com/v_show/id_XNDY3MjgwMjkONjA==.html?x&sharefrom=android&sharekey=9598e833b88a29b0cdbe40474ce8035a4)

### 4.3 定高，悬停，返航模式飞行：

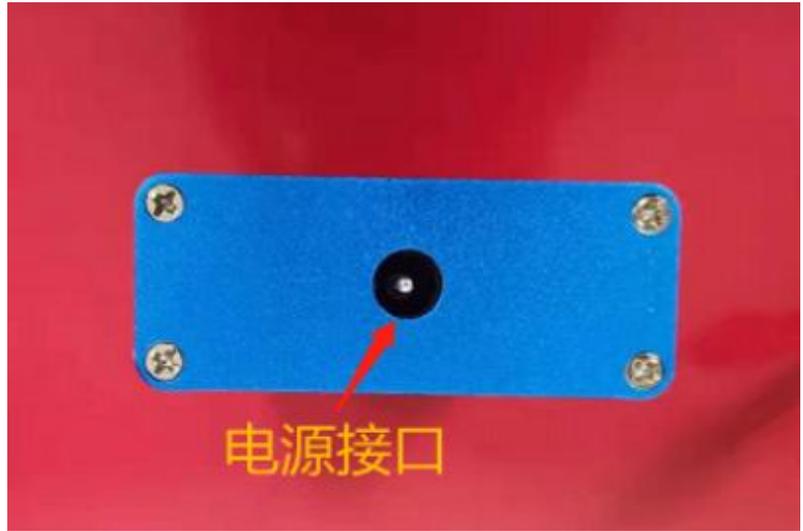
[http://v.youku.com/v\\_show/id\\_XNDY3Mjg1MDUwMA==.html?x&sharefrom=android&sharekey=ea17a33d73174f3fc17280e6d1e8d69e8](http://v.youku.com/v_show/id_XNDY3Mjg1MDUwMA==.html?x&sharefrom=android&sharekey=ea17a33d73174f3fc17280e6d1e8d69e8)

### 4.4 降落，绕圈，有头无头及结束飞行操作：

[http://v.youku.com/v\\_show/id\\_XNDY3Mjg2NjIwNA==.html?x&sharefrom=android&sharekey=52f3a12a8376b91b9f0292fe93cb48b58](http://v.youku.com/v_show/id_XNDY3Mjg2NjIwNA==.html?x&sharefrom=android&sharekey=52f3a12a8376b91b9f0292fe93cb48b58)

## 5. 使用这台无人机需要具备的一些基础知识：

### 5.1 锂电池的正确使用方法和充电操作



[https://v.youku.com/v\\_show/id\\_XMzE5NTM1ODk4MA==.html](https://v.youku.com/v_show/id_XMzE5NTM1ODk4MA==.html)

注意！电量充满后请及时拔出充电器！不要满电后长时间充电，以防电池过充！锂电池是易燃易爆品，短路、剧烈碰撞、针刺，高温等情况都有可能导导致锂电池燃烧，使用时请务必注意安全。请置于阴凉干燥处存放。

## 5.2 低电压报警器（BB 响）的使用

[https://v.youku.com/v\\_show/id\\_XNTE2NjkwMDc2OA==.html](https://v.youku.com/v_show/id_XNTE2NjkwMDc2OA==.html)

## 5.3 数传模块的使用（基础版机型配数传）

[https://v.youku.com/v\\_show/id\\_XNTE2NjkwMDc2NA==.html](https://v.youku.com/v_show/id_XNTE2NjkwMDc2NA==.html)

## 5.4 Pixhawk 飞控的基本调试、地面站的使用参考教程

第一课: <https://mp.weixin.qq.com/s/OC10e1PEhG5Ut0i44SAemw>



第二课: <https://mp.weixin.qq.com/s/ldYP71GftgAF9tEnrcgi4w>

第三课: <https://mp.weixin.qq.com/s/0vN6VtMtSChqy3H2gM1C3w>

注意! 无人机发货前已经进行了调试和试飞。所以没有必要按照教程再做一遍。教程提供给大家学习。

## 5.5 飞行模式的含义和正确使用

[https://mp.weixin.qq.com/s/IeAHBmnx\\_ea3Qd8Jd1NVOA](https://mp.weixin.qq.com/s/IeAHBmnx_ea3Qd8Jd1NVOA)

## 5.6 Pixhawk 飞控不能解锁常见问题汇总

<https://mp.weixin.qq.com/s/xsVch7U80sGDE8jq-FVJWA>

## 5.7 Ubuntu 系统的基本操作

<https://mp.weixin.qq.com/s/wOzOCACHgsz8meFa7QHRJg>

提示: 如若遇见飞控中间灯不亮, 并且地面站显示无法识别到飞控板的情况, 将参数表内的“BRD-TYPE”修改为 2 “Pixhawk”。因为飞控中间的灯大概率烧坏, 地面站无法自动识别正确的飞控板和飞控板型号, 所以手动设定为 Pixhawk。

# 6. 二次开发入门教程

## 6.1 Ardupilot 开源无人机二次开发入门系列教程

[https://mp.weixin.qq.com/mp/appmsgalbum?\\_\\_biz=MzkyNzI1MDUyNw==&action=getalbum&album\\_id=2014808821755379715&scene=173&from\\_msgid=2247485616&from\\_itemidx=1&count=3&nolastread=1#wechat\\_redirect](https://mp.weixin.qq.com/mp/appmsgalbum?__biz=MzkyNzI1MDUyNw==&action=getalbum&album_id=2014808821755379715&scene=173&from_msgid=2247485616&from_itemidx=1&count=3&nolastread=1#wechat_redirect)

# 7. Z410-4B 单目版教程

Z410 单目版在基础版配置上, 扩展了树莓派 4B 机载电脑。目前该系统集成了数据遥测转发, 高清视频回传, ROS, mavros, Dronekit-python 等。我们需要先配置好网络, 通过 PC 远程连接到树莓派, 进而实现对无人机的操作。

树莓派能做什么, 为什么要扩展树莓派?

树莓派是一款基于 ARM 的微型电脑主板, 英文名: Raspberry Pi。是一款入门级的机载电脑。它能够通过串口连接使用 MAVLink 协议与飞控进行通信。

这可用于执行额外的任务，例如：通过 python、ROS 编程实现对无人机的飞行控制；通过安装单目相机实现图像识别，由于存储图像的内存要求，单靠飞控根本无法完成这些任务。

如何对无人机的树莓派进行操作？

## 7.1 建立网络连接

将电池插上 BB 响报警器，绑扎在底板上。无人机连接电池插口，即可启动。启动后先不要挪动无人机，等待飞控和树莓派启动系统，大概 2 分钟左右。系统启动后自动生成 WIFI 热点。

WIFI 名称：ACopter

密码：12345678

将笔记本 WIFI 连接到树莓派热点。

**（注意！因 WIFI 信号范围最大 30 米左右）**

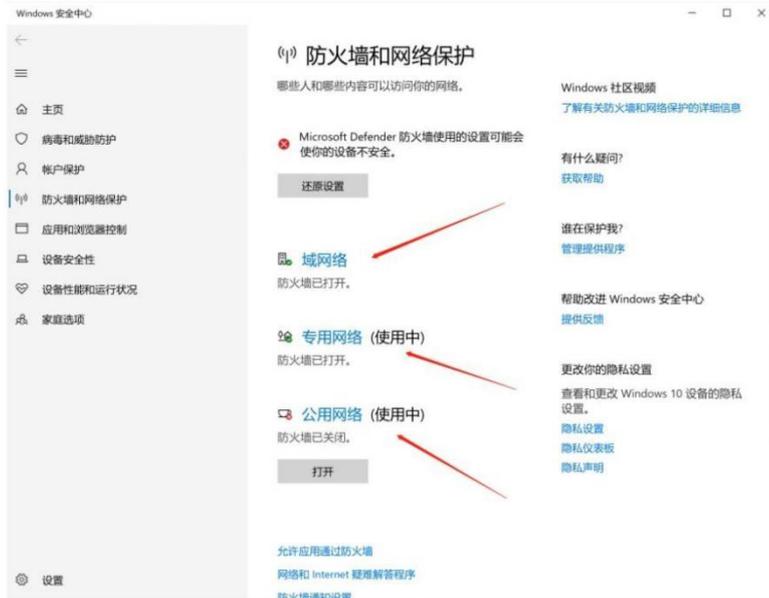
## 7.2 使用树莓派遥测转发功能，将无人机与地面站进行连接

树莓派集成了飞控数据遥测转发的功能，相当于一个路由器。这样我们可以通过 WIFI 网络无线连接飞控和 PC 端地面站软件。

具体操作参考视频：

[https://v.youku.com/v\\_show/id\\_XNTE2MzQ4MTE1Ng==.html](https://v.youku.com/v_show/id_XNTE2MzQ4MTE1Ng==.html)

**注意！如果不能通过 UDP 连接，请先关闭 windows 防火墙。**



### 7.3 使用树莓派的图传功能，可实时回传视频到地面站

树莓派集成了高清图传回传功能，可以将 USB 摄像头拍摄的实时高清信号图像回传到 PC 端地面站软件或者图像播放软件，如：VLC mediaplayer。（下载链接：<https://www.videolan.org/>）

MP 地面站软件显示高清视频需要先安装 GStreamer，下载链接及教程：

<https://pan.baidu.com/s/1AyvQNj2m4E4fMHS6JYnpJA>

提取码：f7c3

安装好 GStreamer 以后，视频回传操作参考以下视频：

[https://v.youku.com/v\\_show/id\\_XNTE2MzQ4MTE20A==.html](https://v.youku.com/v_show/id_XNTE2MzQ4MTE20A==.html)

### 7.4 PC 访问树莓派的 3 种方式：

方法 1：windows 系统下通过远程桌面访问树莓派：

- ① 打开 windows 远程桌面；
- ② 填入树莓派 IP：10.42.0.10，点 连接；
- ③ 在终端输入用户名：ubuntu，密码：123456abc；
- ④ 确认后即可连接到树莓派桌面注意：远程桌面连接占用带宽比较大，速度略有卡顿是正常的。

参考视频:

[https://v.youku.com/v\\_show/id\\_XNTE2MzQ4MTEyMA==.html](https://v.youku.com/v_show/id_XNTE2MzQ4MTEyMA==.html)

方法 1 优点: 有桌面系统, 直观可看。对初学者来说比较容易接受;

方法 1 缺点: 远程桌面会占用很多资源, 导致卡顿;

方法 2: windows 系统下通过 PUTTY 访问树莓派:

① 在 PC 端安装 PuTTY, 下载链接:

[Download PuTTY: latest release \(0.78\) \(greenend.org.uk\)](http://www.greenend.org.uk/gnu/puTTY/download.html)

② 启动 PuTTY, 填写树莓派 IP: 10.42.0.10, 密码: 123456abc

参考视频:

[https://v.youku.com/v\\_show/id\\_XNTE2MzQ4MTIyMA==.html](https://v.youku.com/v_show/id_XNTE2MzQ4MTIyMA==.html)

方法 2 优点: 访问执行速度快, 需要同时使用 MP 地面站和终端命令行比较方便;

方法 2 缺点: 命令行显示, 对初学者来说不直观; 需要远程调用 ubuntu 系统下的窗口不能用这个方法。

方法 3: ubuntu 系统下远程访问树莓派:

① 笔记本打开一个终端, 使用下面的命令修改笔记本上的 hosts 文件:

```
sudo vim /etc/hosts
```

② 将树莓派 4b 的 IP 地址 (10.42.0.10) 加入到 hosts 文件中, 并保存退出;

参考视频:

[https://v.youku.com/v\\_show/id\\_XNTE2NjYxMzEyNA==.html](https://v.youku.com/v_show/id_XNTE2NjYxMzEyNA==.html)

关于 vim 编辑器的使用, 可以参考以下教程的部分内容:

<https://mp.weixin.qq.com/s/BEIz9fa09WYrImbKcRT7NA>

方法 3 优点: 执行速度快, ubuntu 系统下可以做更多的开发

方法 3 缺点: 不能同时使用 MP 地面站

## 7.5 树莓派的关机操作

树莓派是机载电脑, 突然断电有可能造成系统损坏无法开机。建议大家使用以下 2 种方式进行正常的关机操作。

方法 1: 在命令行终端, 输入关机命令

```
sudo halt
```

方法 2: 通过 web 管理界面关机

参考视频:

[https://v.youku.com/v\\_show/id\\_XNTE2NjkxMjM5Ng==.html](https://v.youku.com/v_show/id_XNTE2NjkxMjM5Ng==.html)

树莓派关机以后, 等待大概 1 分钟左右, 树莓派电源指示灯熄灭, 再断开无人机电池。

## 7.6 使用 Dronekit 测试连接飞控并读取状态数据

Dronekit 介绍:

Dronekit 也叫 DroneKit-Python, 是一个用于控制无人机的 Python 库。有了它你就可以在机载电脑上通过 Python 编程实现对无人机的控制。它可以显著增强自动驾驶性能, 为飞行器增加更多智能, 以及执行计算密集或时间敏感的任务 (例如, 计算机视觉, 路径规划或 3D 建模)。除了 DroneKit-Python 以外, 还有 DroneKit-Android 以及 DroneKit-Cloud 的 API 供不同的开发者使用。API 通过 MAVLink 与飞控通信。它提供对连接飞控的遥测, 状态和参数信息的编程访问, 并实现任务管理和对飞行器运动和操作的直接控制。相比之下, Dronekit 比搭建 ros 来控制无人机更容易上手一些。对于 Dronekit, PX4 (原生固件) 被支持的较少, 目前不可以进行模式切换。而对 Ardupilot (APM 固件) 支持的比较多, 可调用的函数也比较多。单目版机型我们已经在树莓派上安装好 Dronekit, 并且在根目录下有一个 test 文件夹, 存放了几个演示示例, 大家可以根据下面的教程操作。

\*如果想了解树莓派与 pixhawk 飞控具体如何连接, 以及如何安装 dronekit 的过程, 可参考以下链接有详细介绍:

第一节: <https://mp.weixin.qq.com/s/1NzoFQqN9q4wPaaZBy5fYg>

第二节: <https://mp.weixin.qq.com/s/QXWT8NOLXTo06n0zwFZoRA>

### 7.6.1 运行连接示例

使用 Dronekit 代码读取飞控当前状态, 测试树莓派与飞控之间通讯是否正常:

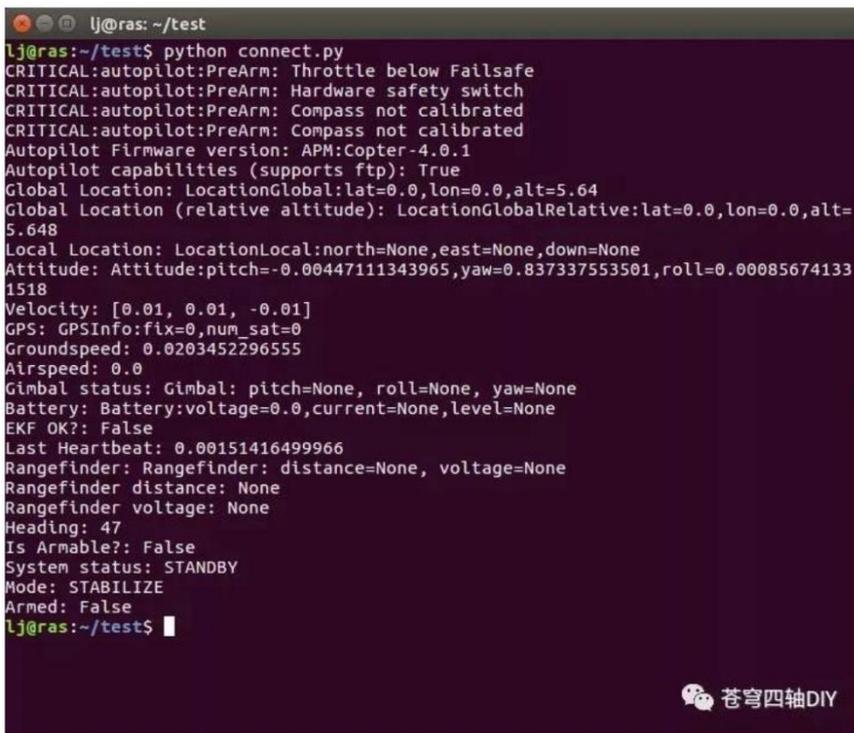
- ① 无人机连接电池，笔记本开启一个终端窗口；
- ② 通过 ssh 命令，连接无人机上的树莓派；
- ③ 将路径切换到 test 文件夹：

```
cd test
```

- ④ 运行 connect.py 脚本：

```
python2 connect.py
```

成功运行后，会显示如下信息：



```
lj@ras: ~/test
lj@ras:~/test$ python connect.py
CRITICAL:autopilot:PreArm: Throttle below Failsafe
CRITICAL:autopilot:PreArm: Hardware safety switch
CRITICAL:autopilot:PreArm: Compass not calibrated
CRITICAL:autopilot:PreArm: Compass not calibrated
Autopilot Firmware version: APM:Copter-4.0.1
Autopilot capabilities (supports ftp): True
Global Location: LocationGlobal:lat=0.0,lon=0.0,alt=5.64
Global Location (relative altitude): LocationGlobalRelative:lat=0.0,lon=0.0,alt=5.648
Local Location: LocationLocal:north=None,east=None,down=None
Attitude: Attitude:pitch=-0.00447111343965,yaw=0.837337553501,roll=0.000856741331518
Velocity: [0.01, 0.01, -0.01]
GPS: GPSInfo:fix=0,num_sat=0
Groundspeed: 0.0203452296555
Airspeed: 0.0
Gimbal status: Gimbal: pitch=None, roll=None, yaw=None
Battery: Battery:voltage=0.0,current=None,level=None
EKF OK?: False
Last Heartbeat: 0.00151416499966
RangeFinder: RangeFinder: distance=None, voltage=None
RangeFinder distance: None
RangeFinder voltage: None
Heading: 47
Is Armable?: False
System status: STANDBY
Mode: STABILIZE
Armed: False
lj@ras:~/test$
```

说明树莓派通过 Dronekit 读取到了目前飞控的数据：系统警报、固件版本、姿态数据、电池电量、解锁状态、当前飞行模式等等。这样就成功运行了第一个 DroneKit-Python 脚本。

**注意！** 如若出现连接异常超时警报，请检查物理连接（usb 转 TTL 模块）是否连接好；或者打开 connect.py 程序查看连接端口是否为 ttyUSB0；无人机通电后如果长时间没有使用串口，串口有可能关闭，无人机重新启动，再执行以上程序。

- ⑤ 如果想查看 connect.py 脚本，可通过 vim 编辑器查看，执行如下命令：

```
sudo vim connect.py
```

其中脚本注释如下：

```
# 飞控软件版本
print "Autopilot Firmware version: %s" % vehicle.version

# 全球定位信息（经纬度，高度相对于平均海平面）
print "Global Location: %s" % vehicle.location.global_frame

# 全球定位信息（经纬度，高度相对于 home 点）
print "Global Location (relative
altitude): %s" % vehicle.location.global_relative_frame

# 相对 home 点的位置信息（向北、向东、向下）；解锁之前返回 None
print "Local Location: %s" % vehicle.location.local_frame

# 无人机朝向（欧拉角：roll, pitch, yaw, 单位为 rad, 范围  $-\pi$  到  $+\pi$ ）
print "Attitude: %s" % vehicle.attitude

# 三维速度 (m/s)
print "Velocity: %s" % vehicle.velocity

# GPS 信息
print "GPS: %s" % vehicle.gps_0

# 地速 (m/s)
print "Groundspeed: %s" % vehicle.groundspeed

# 空速 (m/s)
print "Airspeed: %s" % vehicle.airspeed

# 云台信息（得到的为当前目标的 roll, pitch, yaw, 而非测量值。单位为度）
print "Gimbal status: %s" % vehicle.gimbal

# 电池信息
print "Battery: %s" % vehicle.battery

# EKF（拓展卡曼滤波器）状态
print "EKF OK?: %s" % vehicle.ekf_ok

# 超声波或激光雷达传感器状态
```

```
print "Rangefinder: %s" % vehicle.rangefinder
# 无人机朝向（度）
print "Heading: %s" % vehicle.heading
# 是否可以解锁
print "Is Armable?: %s" % vehicle.is_armable
# 系统状态
print "System status: %s" % vehicle.system_status.state
# 当前飞行模式
print "Mode: %s" % vehicle.mode.name
# 解锁状态
print "Armed: %s" % vehicle.armed
```

## 7.7 启动 SITL 仿真环境并运行 python 示例

在 test 文件夹下我们有几个演示脚本，如果单纯的运行脚本新手没有直观的感受。我们可以结合无人机仿真环境，来学习脚本具体的执行过程。

前面笔记本上安装好 ubuntu 系统镜像，里面已经搭建好了 Ardupilot 开发环境。SITL 仿真也就包含其中。

需要注意：DroneKit-SITL 目前仅为 Mac, Linux 和 Windows 提供 x86 二进制文件。不能在树莓派这样的 ARM 平台上运行它。

我们执行的 python 脚本是在树莓派系统上，而 SITL 仿真环境是搭建在笔记本系统上的。

### ① 树莓派端

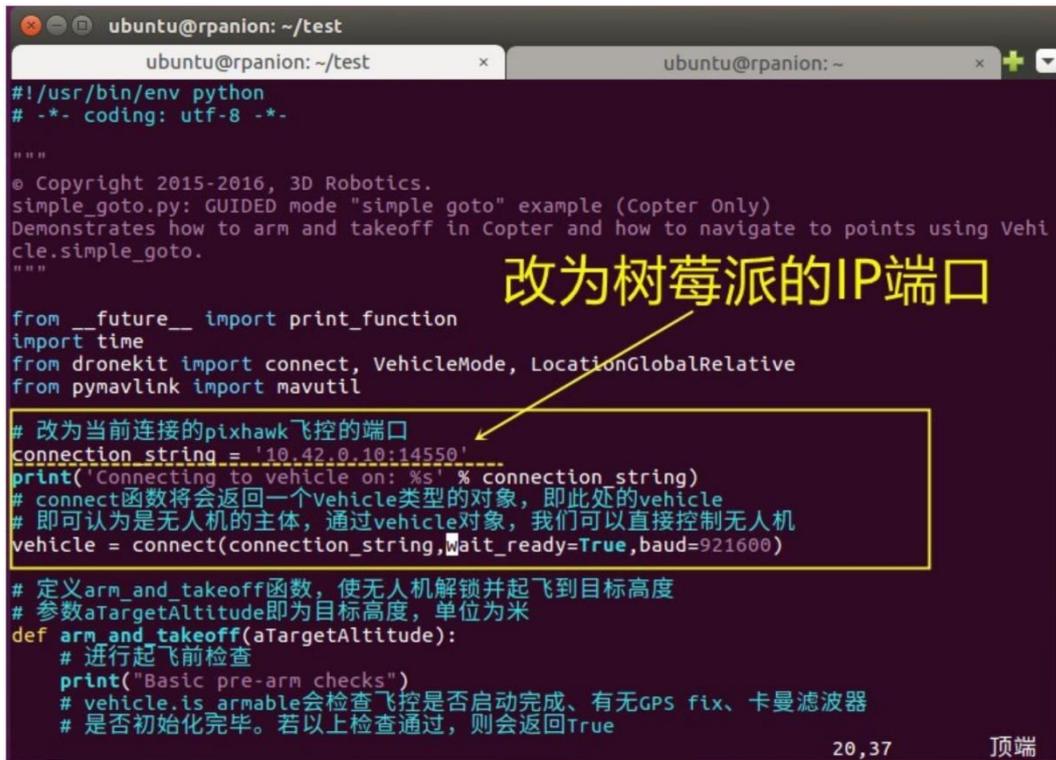
```
进入 test 目录
cd test
```

### ② 编辑 example2.py

```
sudo vim example2.py
```

将其中连接端口改为 `connection_string = '10.42.0.10:14550'` 为树莓派 IP 端口。

脚本里面都配有详细注解，大家可根据注解来理解 dronekit 的基本使用。



```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""
© Copyright 2015-2016, 3D Robotics.
simple_goto.py: GUIDED mode "simple goto" example (Copter Only)
Demonstrates how to arm and takeoff in Copter and how to navigate to points using Vehicle.simple_goto.
"""

from __future__ import print_function
import time
from dronekit import connect, VehicleMode, LocationGlobalRelative
from pymavlink import mavutil

# 改为当前连接的pixhawk飞控的端口
connection_string = '10.42.0.10:14550'
print('Connecting to vehicle on: %s' % connection_string)
# connect函数将会返回一个Vehicle类型的对象，即此处的vehicle
# 即可认为是无人机的主体，通过vehicle对象，我们可以直接控制无人机
vehicle = connect(connection_string, wait_ready=True, baud=921600)

# 定义arm_and_takeoff函数，使无人机解锁并起飞到目标高度
# 参数aTargetAltitude即为目标高度，单位为米
def arm_and_takeoff(aTargetAltitude):
    # 进行起飞前检查
    print("Basic pre-arm checks")
    # vehicle.is_arming会检查飞控是否启动完成、有无GPS fix、卡曼滤波器
    # 是否初始化完毕。若以上检查通过，则会返回True

20,37 顶端
```

### ③ 笔记本端

首先要进入需要仿真的多旋翼无人机的目录下：

```
cd ~/Workspace/ardupilot/ArduCopter
```

第一次运行，需要执行下面的命令对仿真环境进行初始化

```
../Tools/autotest/sim_vehicle.py -w
```

或者是仿真参数被改得乱七八糟的时候，也可用这个命令恢复初始参数。

启动完毕，使用 Ctrl+C 终止正在运行的 sim\_vehicle.py -w

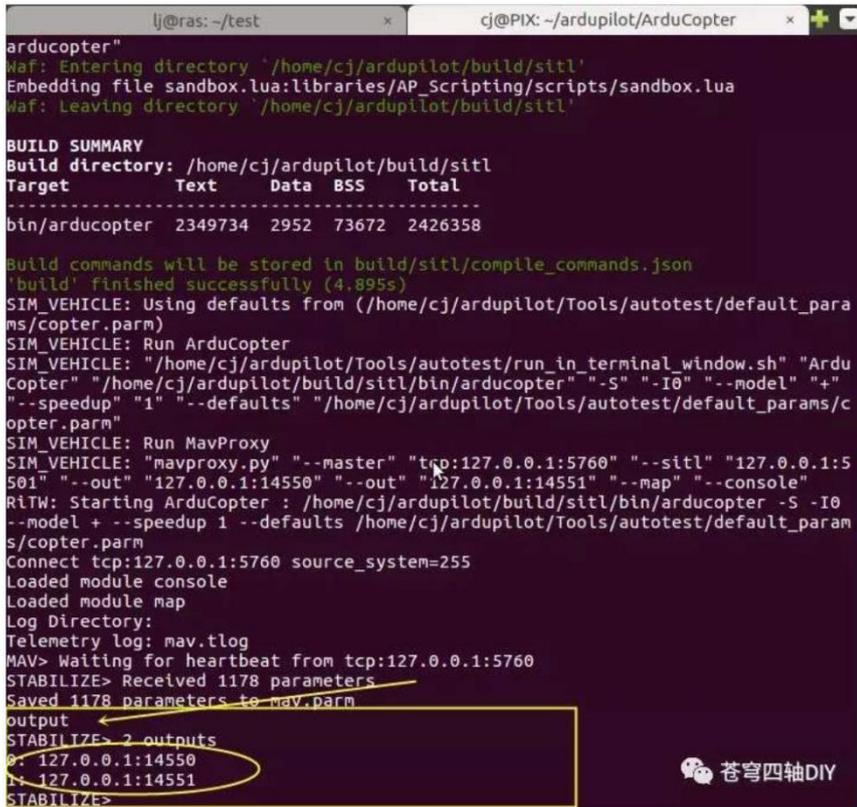
### ④ 接下来就可以启动模拟器了

```
../Tools/autotest/sim_vehicle.py --console --map
```

正常启动后，就会看到三个窗口：Terminal, Console, Map，这样最基本的软件在环仿真程序就运行起来了。

### ⑤ 启动后通过 output 命令列出 MAVProxy 转发数据的接口，如下

所示:



```
arducopter"
Waf: Entering directory `/home/cj/ardupilot/build/sitl'
Embedding file sandbox.lua:libraries/AP_Scripting/scripts/sandbox.lua
Waf: Leaving directory `/home/cj/ardupilot/build/sitl'

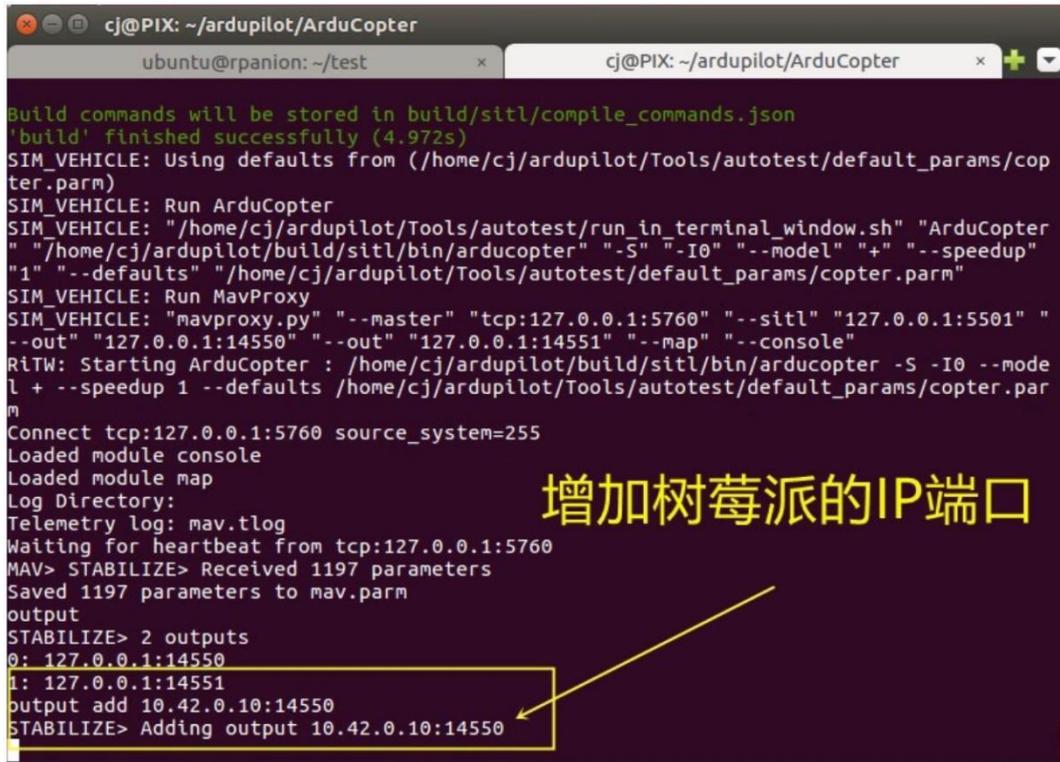
BUILD SUMMARY
Build directory: /home/cj/ardupilot/build/sitl
Target      Text      Data      BSS      Total
-----
bin/arducopter 2349734 2952 73672 2426358

Build commands will be stored in build/sitl/compile_commands.json
'build' finished successfully (4.895s)
SIM_VEHICLE: Using defaults from (/home/cj/ardupilot/Tools/autotest/default_params/copter.parm)
SIM_VEHICLE: Run ArduCopter
SIM_VEHICLE: "/home/cj/ardupilot/Tools/autotest/run_in_terminal_window.sh" "ArduCopter" "/home/cj/ardupilot/build/sitl/bin/arducopter" "-S" "-I0" "--model" "+" "--speedup" "1" "--defaults" "/home/cj/ardupilot/Tools/autotest/default_params/copter.parm"
SIM_VEHICLE: Run MavProxy
SIM_VEHICLE: "mavproxy.py" "--master" "tcp:127.0.0.1:5760" "--sitl" "127.0.0.1:5501" "--out" "127.0.0.1:14550" "--out" "127.0.0.1:14551" "--map" "--console"
RtW: Starting ArduCopter : /home/cj/ardupilot/build/sitl/bin/arducopter -S -I0 --model + --speedup 1 --defaults /home/cj/ardupilot/Tools/autotest/default_params/copter.parm
Connect tcp:127.0.0.1:5760 source_system=255
Loaded module console
Loaded module map
Log Directory:
Telemetry log: mav.tlog
MAV> Waiting for heartbeat from tcp:127.0.0.1:5760
STABILIZE> Received 1178 parameters
Saved 1178 parameters to mav.parm
output
STABILIZE> 2 outputs
0: 127.0.0.1:14550
1: 127.0.0.1:14551
STABILIZE>
```

六 使用 output add 命令增加树莓派 IP 接口:

```
output add 10.42.0.10:14550
```

与 example2.py 代码中端口地址一致。



```
Build commands will be stored in build/sitl/compile_commands.json
'build' finished successfully (4.972s)
SIM_VEHICLE: Using defaults from (/home/cj/ardupilot/Tools/autotest/default_params/copter.parm)
SIM_VEHICLE: Run ArduCopter
SIM_VEHICLE: "/home/cj/ardupilot/Tools/autotest/run_in_terminal_window.sh" "ArduCopter"
"/home/cj/ardupilot/build/sitl/bin/arducopter" "-S" "-I0" "--model" "+" "--speedup"
"1" "--defaults" "/home/cj/ardupilot/Tools/autotest/default_params/copter.parm"
SIM_VEHICLE: Run MavProxy
SIM_VEHICLE: "mavproxy.py" "--master" "tcp:127.0.0.1:5760" "--sitl" "127.0.0.1:5501"
"--out" "127.0.0.1:14550" "--out" "127.0.0.1:14551" "--map" "--console"
RITW: Starting ArduCopter : /home/cj/ardupilot/build/sitl/bin/arducopter -S -I0 --model
+ --speedup 1 --defaults /home/cj/ardupilot/Tools/autotest/default_params/copter.parm
Connect tcp:127.0.0.1:5760 source_system=255
Loaded module console
Loaded module map
Log Directory:
Telemetry log: mav.tlog
Waiting for heartbeat from tcp:127.0.0.1:5760
MAV> STABILIZE> Received 1197 parameters
Saved 1197 parameters to mav.parm
output
STABILIZE> 2 outputs
0: 127.0.0.1:14550
1: 127.0.0.1:14551
output add 10.42.0.10:14550
STABILIZE> Adding output 10.42.0.10:14550
```

⑦ 树莓派端执行

运行 example2.py 代码

python2 example2.py

参考视频: [https://v.youku.com/v\\_show/id\\_XNDY2ODYxMTE20A==.html](https://v.youku.com/v_show/id_XNDY2ODYxMTE20A==.html)

⑧ 我们还可以将航点文件导入 SITL 进行演示

具体参考如下视频

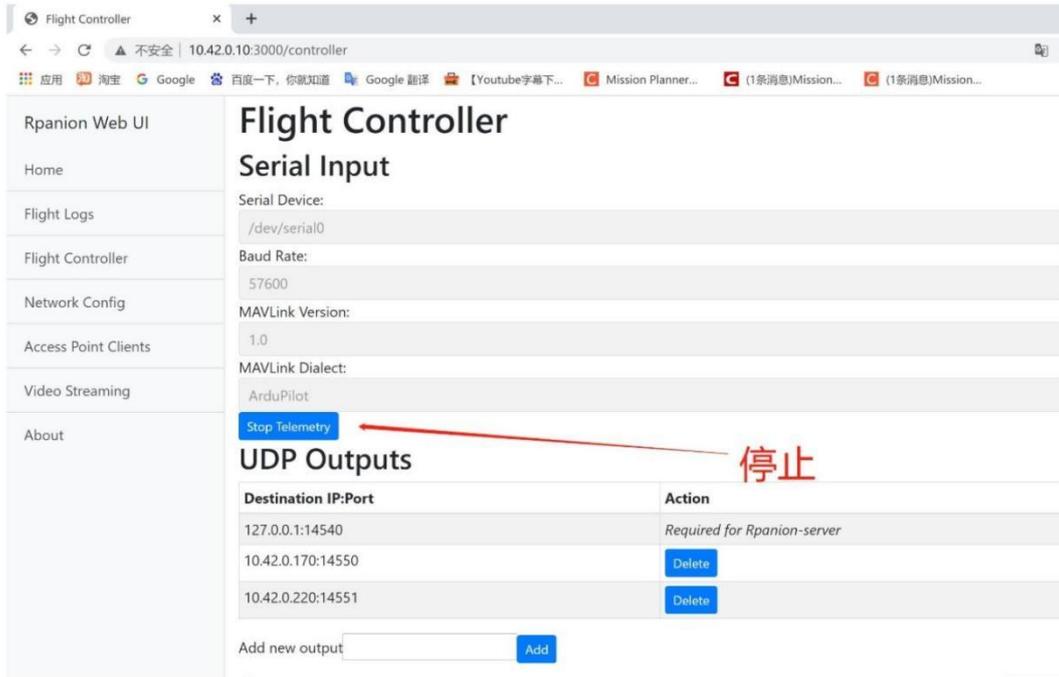
[https://v.youku.com/v\\_show/id\\_XNDY2ODYxMjgzNg==.html](https://v.youku.com/v_show/id_XNDY2ODYxMjgzNg==.html)

## 7.8 SITL 仿真结合 MP 地面站运行 python 示例

首先, 将树莓派无人机接上电池, 确保 ubuntu 电脑, MP 地面站电脑都连接到树莓派 WIFI 热点。

MP 地面站电脑:

要在 MP 上显示仿真, 遥测连接务必断开。因为遥测连接的是实际的飞控, 做仿真需要连接仿真数据。



ubuntu 笔记本端:

启动 SITL 仿真

```
cd ~/Workspace/ardupilot/ArduCopter
```

```
../Tools/autotest/sim_vehicle.py --console --map
```

使用 output add 命令, 增加树莓派的 IP:

```
output add 10.42.0.10:14550
```

再使用 output add 命令, 增加 MP 地面站的 IP, MP 地面站 IP 地址在遥测页面查看:

```
output add 10.42.0.170:14551
```

树莓派端:

运行示例代码:

```
cd test
```

```
python2 example2.py
```

以下视频演示 SITL 仿真如何结合 MP 地面站运行

[https://v.youku.com/v\\_show/id\\_XNTE2NjczMDA5Mg==.html](https://v.youku.com/v_show/id_XNTE2NjczMDA5Mg==.html)

SITL 与 MP 地面站结合运行, 那么我们可以直接在 MP 地面站的全部参数

列表里，查找需要修改的参数进行修改。

有关 SITL 仿真的详细介绍，可查看以下链接：

<https://ardupilot.org/dev/docs/using-sitl-for-ardupilot-testing.html>

## 7.9 使用 Python 脚本控制无人机飞行

根据 dronekit 的官方示例，我们编写了几个简单的程序控制无人机飞行。

编写完以后先在 SITL 仿真环境下运行看看是否和预想的一致。没有问题后，再连接无人机实地飞行。

example5.py 控制无人机起飞到 3 米高度，然后悬停 5 秒，再自动降落；

大家可根据我们提供的示例，结合自己的实际情况，编写控制程序。

Dronekit 室外控制无人机起飞降落演示：

[https://v.youku.com/v\\_show/id\\_XNDY5NzA50TQ20A==.html](https://v.youku.com/v_show/id_XNDY5NzA50TQ20A==.html)

example4.py 在室外空旷处，此脚本控制无人机起飞到 3 米高度，然后飞出一个 2 米\*2 米

的正方形，再自动降落。

Dronekit 室外控制无人机飞出一个正方形演示：

[https://v.youku.com/v\\_show/id\\_XNDY5NzEwMzExNg==.html](https://v.youku.com/v_show/id_XNDY5NzEwMzExNg==.html)

还有值得注意的是，python 脚本中设置属性：

少数的属性变量可以被设置，通过设置这些属性变量，可以控制无人机的运行状态。可设置的属性变量如下：

vehicle.home\_location

vehicle.gimbal

vehicle.airspeed

vehicle.groundspeed

vehicle.mode

vehicle.armed

vehicle.disarmed

设置示例：

① 锁定无人机:

```
vehicle.disarmed = False
```

② 切换到 GUIDED 模式:

```
vehicle.mode = VehicleMode("GUIDED")
```

③ 设置航点模式下，无人机飞行的地速为 3.2m/s（注意：读、写 groundspeed 的意义不同）

```
print "Param: %s" % vehicle.parameters['THR_MIN']
```

```
vehicle.groundspeed = 3.2
```

### 读取参数

参数以字典（dictionary）的形式，存储在 `vehicle.parameters` 变量中。  
具体参数的名称

即为相应的键值（key）。

例如，在屏幕上显示 THR\_MIN 参数（THR\_MIN 代表油门处于最低时的电机怠速，以 PWM 值表示）：

```
print "Param: %s" % vehicle.parameters['THR_MIN']
```

显示全部参数:

```
print "\nPrint all parameters (iterate `vehicle.parameters`):"
```

```
for key, value in vehicle.parameters.iteritems():
```

```
print " Key:%s Value:%s" % (key,value)
```

### 设置参数

使用读取参数类似的方法，即可设置参数:

```
vehicle.parameters['THR_MIN']=100
```

以上只是截取了部分使用较多的知识点进行介绍，完整的教程可以到官网查看。

关于 dronekit 的一些参考资料:

有关请访问 <https://readthedocs.org/projects/dronekit-python/>。这包括指南，示例和 API 参考资料。

示例源代码托管在 Github 上，作为 `/dronekit-python/examples` 的子

文件夹。

DroneKit 论坛是寻求有关如何使用该库的技术支持的最佳场所。也可以查看 Gitterchannel。

说明文件：

<https://dronekit-python.readthedocs.io/en/latest/about/index.html>

指南：

<https://dronekit-python.readthedocs.io/en/latest/guide/index.html>

API 参考：

<https://dronekit-python.readthedocs.io/en/latest/automodule.html>

例子：<https://dronekit-python/examples>

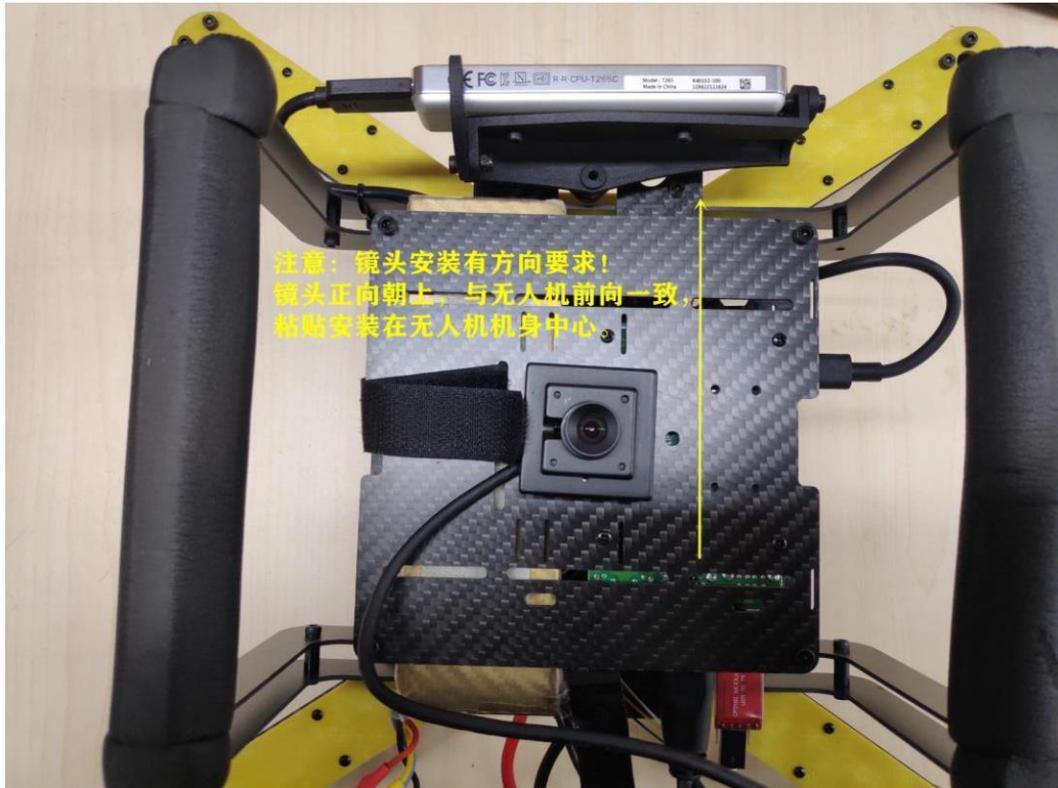
论坛：<https://github.com/dronekit/dronekit-python/issues>

Gitter：<https://gitter.im/dronekit/dronekit-python>

## 8. 二维码识别降落例程

Z410 具备 Arcuo 二维码识别降落的功能。请将教程最后附件 1 的二维码打印粘贴在一张板材上。（注意：板材不能太轻，否则无人机会把二维码标识吹跑了！）

- ① 先将前向的单目摄像头取下，使用 3M 胶粘在飞机机身下方，如图所示：  
**注意！镜头安装有方向要求！镜头正向朝上，与无人机前向一致，安装在无人机机身中心。**



- ② 如前面操作一样，将笔记本通过 ssh 远程连接树莓派，然后按照下面切换路径

```
cd ~/Workspace/aruco-3.1.12/build/utils
```

- ③ 然后启动二维码识别例程

```
python2 vision_landing.py
```

- ④ 将无人机在悬停模式下起飞并飞出一段距离，切换到返航模式，它会自动识别返航点下方的二维码并降落。

二维码精准降落演示：

<https://www.bilibili.com/video/BV1rA4y197Na/>

- ⑤ Z410 树莓派无人机机器视觉入门系列教程

<https://mp.weixin.qq.com/s/g3utWpZIenJYKK71pG5xKQ>

## 9. Z410 视觉版教程

为什么要使用双目相机？

我们在室外飞行的时候，无人机定位是依靠 GPS 全球定位系统。GPS 导航现已十分普及，但 GPS 的精度目前限制在大约 10 米。另外，更新速率比较慢，大约是 10Hz，并且只能支持户外，因为它们需要接收来自 GPS 卫星的电子三角测量信号，同时山脉和建筑物的反射会造成多径干扰。所以在高大建筑周边，建筑物室内和隧道等地方就没有 GPS 信号，我们就需要使用其他方法进行定位导航。

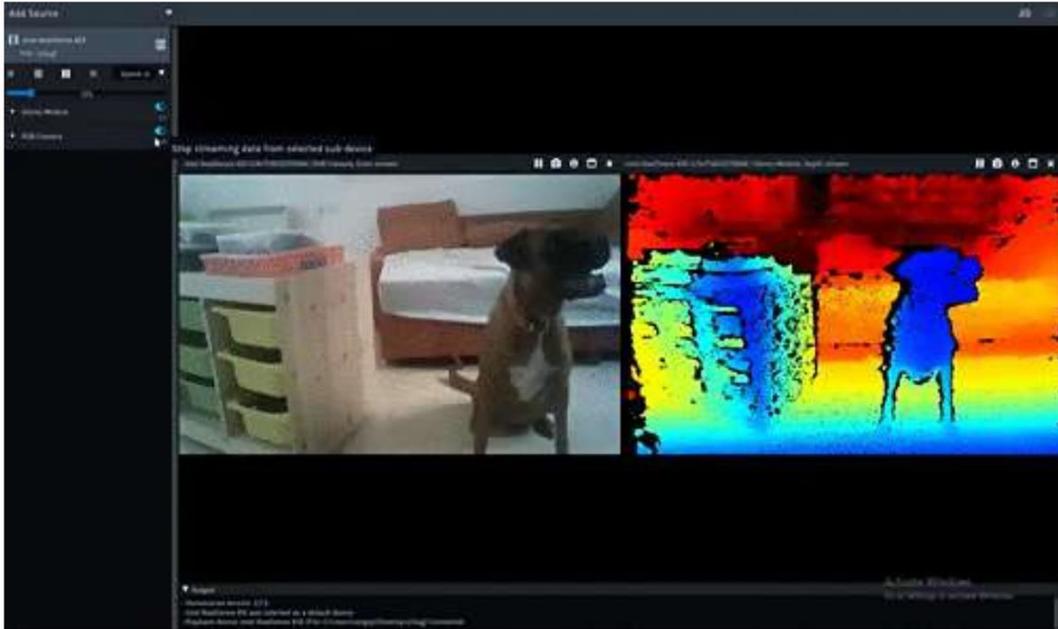
视觉版机型在单目版基础上增加了 intel T265 双目相机，可以通过 v-slam(视觉惯性测距即时定位与地图构建)技术，让无人机进行室内定位导航。英特尔实感追踪摄像头 T265 是一个完整的嵌入式 SLAM 解决方案。它根据 VIO(视觉惯性里程计)并利用最新的算法来输出 6 自由度追踪。通过对 6 自由度姿态数据(位置和方向)的处理，发送给无人机定位。双目定位后，如果室内空间足够，就可以进行悬停飞行，一键返航，航点飞行等。从而为以后室内测试更复杂的避障，导航规划等项目奠定基础。

### 9.1 一些常用的 T265 命令

在安装有 ubuntu 系统的 PC 端，输入以下命令来实时查看 T265 视频：  
演示视频：

[https://v.youku.com/v\\_show/id\\_XNTE2Njg5NDE2NA==.html](https://v.youku.com/v_show/id_XNTE2Njg5NDE2NA==.html)

**realsense-viewer**: RealSense Viewer 是旗舰工具，通过简单的跨平台 UI 提供对大多数相机功能的访问。注意：由于树莓派 WIFI 带宽有限，通过 ssh 访问树莓派开启 realsense-viewer 是有卡顿的。如果要观察 T265 视频画面，建议使用 rs-capture 命令。



<https://github.com/IntelRealSense/librealsense/tree/master/tools/realsense-viewer>

**rs-capture:** 将深度和 RGB 数据流式传输和渲染到屏幕

<https://dev.intelrealsense.com/docs/rs-capture>

**rs-pose:** 应用程序将打开一个窗口，在该窗口中打印设备位置相对于初始位置的当前 x、y、z 值。

<https://dev.intelrealsense.com/docs/rs-pose>

**rs-pose-predict:** 应用程序应打开一个窗口，在该窗口中打印设备位置相对于其初始位置的预测 x、y、z 值。

<https://dev.intelrealsense.com/docs/rs-pose-predict>

**rs-enumerate-devices:** T265 在出厂前就对相机进行了标定，每个相机内部都带有相机的内参和外参。在终端中输入此命令，即可读取到 T265 的配置信息

**rs-enumerate-devices -c:** 如果需要读取到相机内参和外参，以及 IMU 的参数则可以使用此命令（添加 -c 参数）

## 9.2 启用 T265 定位的飞控参数设置

默认情况下，无人机发货前设置 pixhawk 飞控使用 GPS 在室外定位导航。

如果要使用 T265 双目相机进行室内定位，飞控参数需要进行如下设置：

把飞控连接地面站，点击 配置/调试 菜单---全部参数列表：

先检查以下参数：

AHRS\_EKF\_TYPE = 3 (EKF 类型=EKF3)

EK3\_ENABLE = 1 (EKF3 启用)

EK2\_ENABLE = 0 (EKF2 停用)

将第一组定位源参数按照以下设置

EK3\_SRC1\_POSXY = 6 (水平位置使用外部导航)

EK3\_SRC1\_POSZ = 1 (垂直位置使用气压计定高)

EK3\_SRC1\_VELXY = 6 (水平速度使用外部导航)

EK3\_SRC1\_VELZ = 6 (垂直速度使用外部导航)

EK3\_SRC1\_YAW = 1 (偏航使用外置罗盘)

COMPASS\_USE = 1 (使用外置罗盘)

GPS\_TYPE = 0 (禁用 GPS)

VISO\_TYPE = 2 (IntelT265)

禁用所有速度融合

EK3\_SRC\_OPTIONS = 0

RTL\_ALT=0 室内测试请将返航高度设置为 0，让飞行器一当前高度返航。默认是 1.5 米。

(以下是飞控默认数据，要使用 GPS 定位改回来就可以了。)

以下是第一组定位源设置，默认情况下不变，使用 GPS 定位

EK3\_SRC1\_POSXY = 3 (主要水平位置使用 GPS)

EK3\_SRC1\_POSZ = 1 (主要垂直位置使用气压计)

EK3\_SRC1\_VELXY = 3 (主要水平速度使用 GPS)

EK3\_SRC1\_VELZ = 3 (主要垂直速度使用 GPS)

EK3\_SRC1\_YAW = 1 (主要偏航/指南针航向)

GPS\_TYPE = 1 (GPS 类型 自动)

VISO\_TYPE = 0

本机搭载 Intel T265 双目相机，在室内没有 GPS 信号的情况下，代替 GPS 导航定位。双目定位后，如果室内空间足够，就可以进行悬停飞行，一键返航，失控返航，航点飞行。从而为以后室内测试更复杂的避障，导航规划等项目奠定基础。

## 9.3 使用 T265 定位导航的 2 种方法

### 9.3.1 使用 pyrealsense2 启动双目相机

使用 pyrealsense2 定位原理：将从 Realsense T265 获得的 6 自由度姿态数据（位置和方向）和置信度数据交由 python 脚本处理（该脚本位于 vision\_to\_mavros/scripts/t265\_to\_mavlink.py），处理后的结果通过 MAVLink 发送到 ArduPilot 无人机，从而实现定位。

该脚本将执行以下任务：

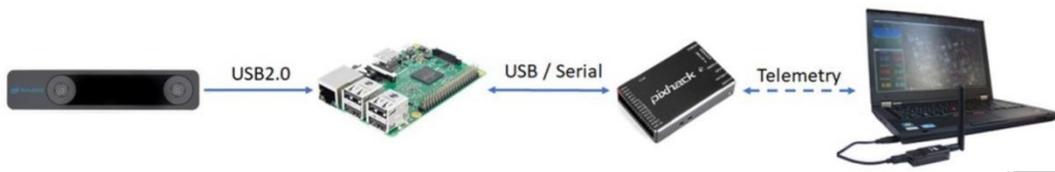
a. 使用 pyrealsense2 的相关 API 获取 6 自由度姿势数据并跟踪置信度数据，

pyrealsense2 是 librealsense 的 Python 包装器；

b. 执行必要的矩阵变换，以匹配 Realsense T265 和 NED 的框架以及其他处理步骤；

c. 将姿态数据打包到 MAVLink 消息 VISION\_POSITION\_ESTIMATE 中，并将置信度数据打包到虚拟消息中，然后将它们以预定频率发送到 ArduPilot，以免输入数据淹没飞控；

d. 自动设置 EKF home 点，让设置和飞行都更为简单；



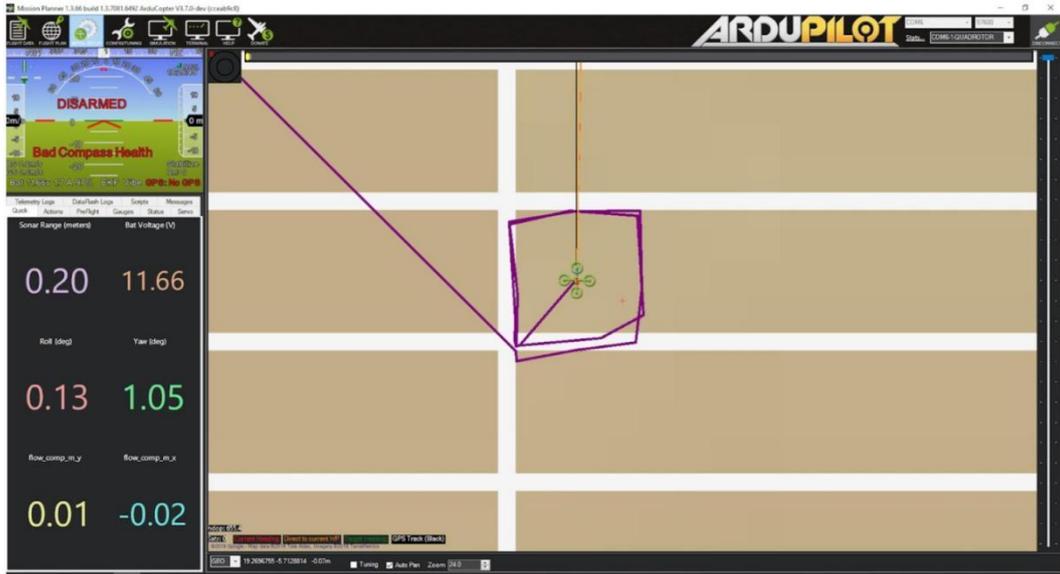
飞行器机载电脑已经安装好 librealsense 和 pyrealsense2，根据以下步骤即可启动 T265

无人机接上电池，打开 missionplanner 地面站，使用数传连接地面站和飞



下是在 2m x 2m 的正方形中行走的示例。

如果外部定位导航数据由于任何原因丢失（跟踪丢失，脚本中断等），重新



运行脚本

也不会成功定位，需要重新启动无人机（重新上电）并执行之前的操作。

飞行测试：（请有无人机操作经验的人操作）

- a. 在自稳或者定高模式下解锁起飞，检查无人机飞行是否平稳；
- b. 使用遥控器操作无人机四处移动，并观察 Mission Planner 上的位置是否稳定；

c. 以上没有问题，保持无人机在 0.8~1 米左右高度，油门 50%左右，切换到 Loiter 模式，但是如果出现任何问题，请随时准备切换回 Stabilize（自稳）/ Alt-Hold（定高）；

d. 正常情况下，无人机应该稳定悬停在空中；

e. 遥控器操作无人机四处缓慢移动，查看无人机稳定状态。测试时注意始终准备切换回“稳定/保持”状态；

f. 如果一切正常，下次就可以在 Loiter 模式下解锁和起飞。

注意：请确保双目相机水平向前安装，测试环境光照充足，双目前方特征明显的视觉范围测试。

数据查看：

- a. 视觉测距信息将出现在 VISO 数据闪存日志消息中。

b. EKF 的视觉测距信息将显示在 XKFD 消息中。

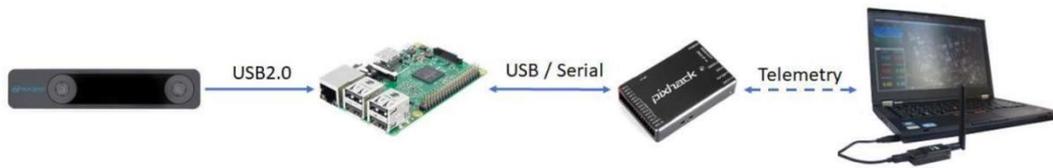
参考视频:

[https://v.youku.com/v\\_show/id\\_XNTE2NjYxMzE1Ng==.html](https://v.youku.com/v_show/id_XNTE2NjYxMzE1Ng==.html)

### 9.3.2 ROS 系统下启用 T265

这是第二种使用双目相机定位的方法。

ROS 下 双目相机 T265 的定位原理:从 `realsense-ros` 节点获得的位置数据由 `vision_to_mavros` 节点处理,并通过主题 `/mavros/vision_pose/pose` 发送到 `mavros` 节点。`mavros` 将负责 ENU-NED 帧转换,并通过 MAVLink 将其发送到 ArduPilot 无人机,从而实现室内定位。



本机已经安装好 `librealsense`、`realsense-ros`、`mavros`、`vision_to_mavros` 等必要的组件,根据以下教程即可启动 T265 定位。

主从机设置完成

无人机接上电池,打开 `missionplanner` 地面站,使用数传连接地面站和飞控;

使用 `ssh` 远程连接树莓派;

开启一个终端,运行 `realsense-ros` 节点

```
roslaunch realsense2_camera rs_t265.launch
```

`/camera/odom/sample/` 和 `/tf` 主题将以 200Hz 频率发布。

开启另一个终端,运行 `mavros` 节点

```
roslaunch mavros apm.launch
```

```
rostopic echo /mavros/vision_pose/pose 未发布。
```

开启第三个终端,运行 `vision_to_mavros` 节点

```
roslaunch vision_to_mavros t265_tf_to_mavros.launch
```

```
rostopic echo /mavros/vision_pose/pose
```

 现在应该显示来自 T265 的姿态数据。

rostopic hz /mavros/vision\_pose/pose 应该显示该主题以 30Hz 的频率发布。

以上 3 个节点运行成功，并且 FCU 开始接收 VISION\_POSITION\_ESTIMATE 消息，就看到地面站会消息框会显示“GPS Glitch”和“GPS Glitch cleared”，确认系统已识别了外部本地化数据。

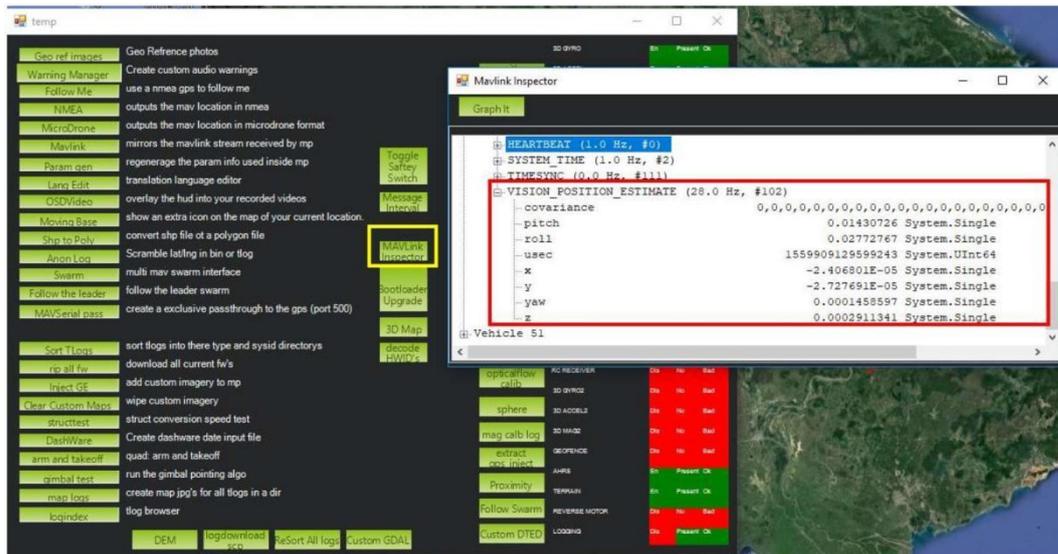
设置 EKF HOME 点

a. 使用 Mission Planner: 右键单击地图上的任意点 > Set Home Here > Set EKF Origin Here.

b. 使用脚本设置 EKF HOME 点，开启第四个终端，运行：

```
roslaunch vision_to_mavros set_origin.py
```

打开地面站软件，快捷键 CTR+F，点击 Mavlink Inspector，能看到数据已经上传到飞控了。室内定位运行成功，切换到 loiter 悬停模式，指示灯是蓝色，表示可以解锁；如果切换到 loiter 变成黄灯，表示室内定位运行失败，请重新检查再启动。



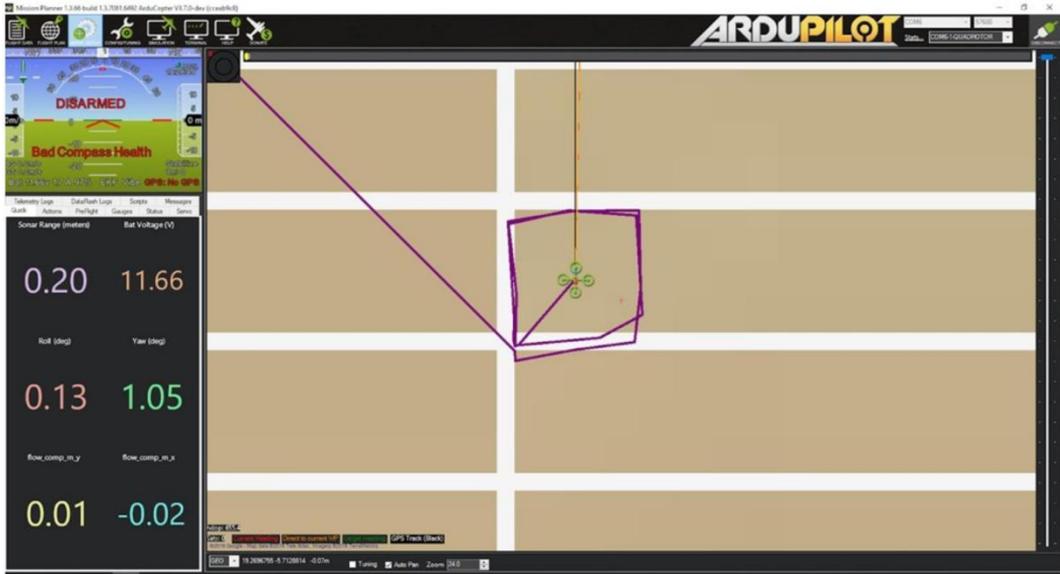
同时 usb 转 ttl 模块上的 rx 和 tx 指示灯会快速闪烁，表示有数据传输；稍等片刻直到四轴飞行器图标出现在 Mission Planner 的地图上；

拿起无人机并四处走动，检查地图上是否显示了无人机的位置运动。地图上显示的无人机的轨迹应该反映真实的运动，而不会产生过多的失真或超调。以

下是在 2m x 2m 的正方形中行走的示例

飞行测试：（请有无人机操作经验的人操作）

a. 在自稳或者定高模式下解锁起飞，检查无人机飞行是否平稳；



b. 使用遥控器操作无人机四处移动，并观察 Mission Planner 和 rviz 上的位置，以查看跟踪是否稳定；

c. 以上没有问题，保持无人机在 0.8~1 米左右高度，油门 50%左右，切换到 Loiter，但是如果出现任何问题，请随时准备切换回 Stabilize / Alt-Hold。

d. 正常情况下，无人机应稳定悬停并能够保持其位置。使用遥控器缓慢移动无人机 2-3 米，验证比例（便于在 rviz 上查看）

e. 如果一切正常，下次就可以在 Loiter 模式下解锁和起飞。

**注意！在切换到 Loiter 模式之前，请始终确认位置反馈运行正常。注意环境中的工作边界，即由于缺少功能，请勿快速移动或旋转运动，可能丢失跟踪定位。**

f. 如果外部定位导航数据由于任何原因丢失（跟踪丢失，脚本中断等），重新运行脚本也不会成功定位，需要重新启动无人机（重新上电）并执行之前的操作。

数据查看：

视觉测距信息将出现在 VISO 数据闪存日志消息中。

演示视频：

[https://v.youku.com/v\\_show/id\\_XNTE2NjYxMzE4MA==.html](https://v.youku.com/v_show/id_XNTE2NjYxMzE4MA==.html)

成功运行以上节点，下次可以只需要开启一个终端，使用下面这个命令一次启动所有节点：

```
roslaunch vision_to_mavros t265_all_nodes.launch
```

此命令一次执行 3 个 launch：

```
rs_t265.launch
```

```
apm.launch
```

```
t265_tf_to_mavros.launch
```

这样可避免开启多个终端，简化操作，方便飞行。

再使用脚本设置 EKF HOME 点：

```
roslaunch vision_to_mavros set_origin.py
```

### 9.3.3 开机自启 T265 双目相机教程

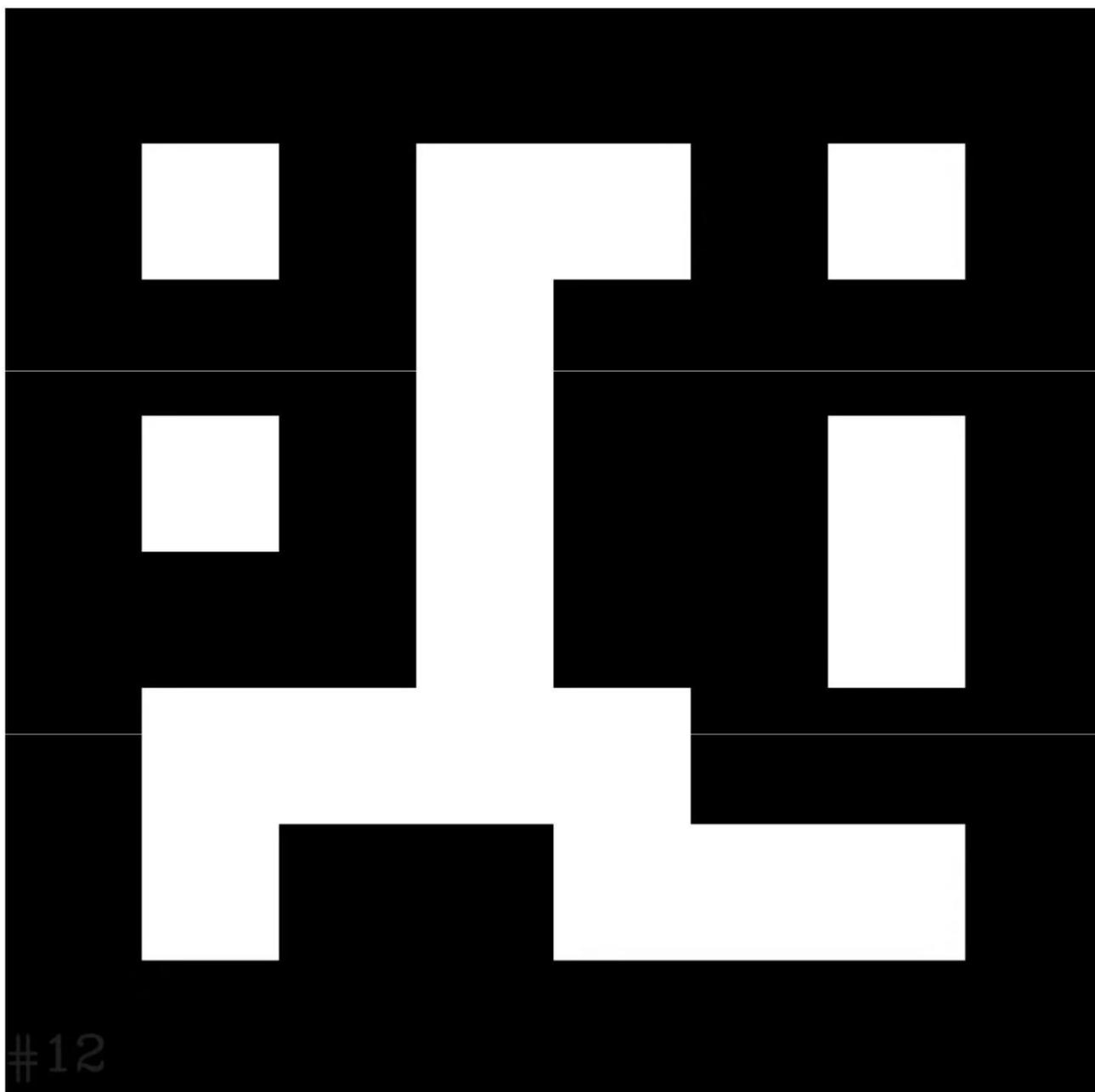
<https://mp.weixin.qq.com/s/P3NB2fLlh-eliHyN8PtUWg>

### 9.3.4 GPS 定位与 T265 定位的快速切换设置教程

<https://mp.weixin.qq.com/s/0ix848mSd-ImWm7v9irMRg>

以上教程是 Z410 的一些基本操作。大家可以利用 ubuntu 系统与 ROS 系统的开源特性，在此基础上进行扩展，比如避障，跟踪，导航等方面。大家也可以加入到 Z410 技术讨论微信群，分享自己在开发方面的心得、方法、示例和对遇到的问题进行讨论。

附件 1: Arcuo 码



## 10. Z410-4B 无人机扩展 4G 模块实现 4G 图数传功能

Z410-4B 无人机单目版与视觉版有树莓派 4B 机载电脑，可扩展 4G 模块，通过网络实现高清图传和数传的功能。从而实现超远程控制。之前 Z410-4B 无人机的控制是通过遥控器的 2.4G 无线电信号，高清图传是通过树莓派 WIFI 网

络回传。无人机扩展 4G 模块后，通过软件的方式与电脑组网通讯。也就是说，在 4G 网络下，无人机的实时数据回传、高清视频回传都依靠 4G 网络进行。

优点是遥控距离不受限制（有 4G 网络的地方都可以）；

缺点是初次设置比较繁琐。之前没有 Pixhawk 和树莓派使用经验的同学可能需要多点耐心。

**注意！因网络有延迟（200~300ms 左右）和中断的可能，不建议新手使用 4G 作为无人机的控制方式。可以应用在远距离遥测数据回传，和高清视频回传。**

要实现 4G 模式下的远距离图数传，需要进行以下 3 步：

第一步 在安装 MP 地面站的 Windows 电脑上安装 Tailscale，获取电脑虚拟 IP，并组建虚拟局域网；

第二步 无人机（树莓派）加入 Tailscale 虚拟局域网，获取树莓派 IP；

第三步 重新分配 UDP 端口，MP 地面站获取遥测数据和高清视频。

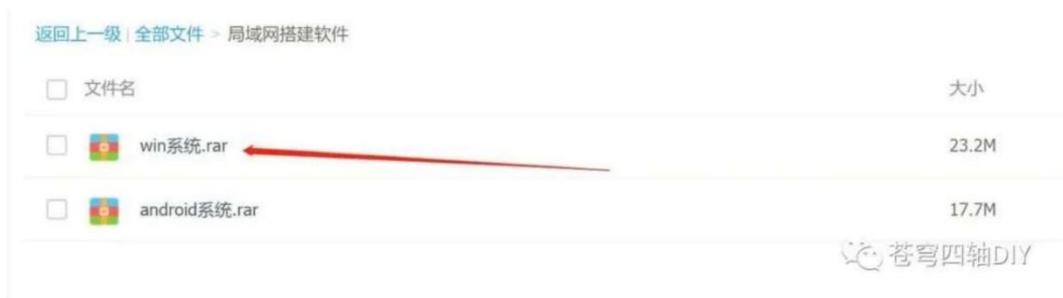
下面分别介绍每一步的具体操作：

## 10.1 Windows 系统下安装 Tailscale

下载链接：

<https://pan.baidu.com/s/11I-QQHW7oE1F1bfFnmGgpg>

提取码：921q



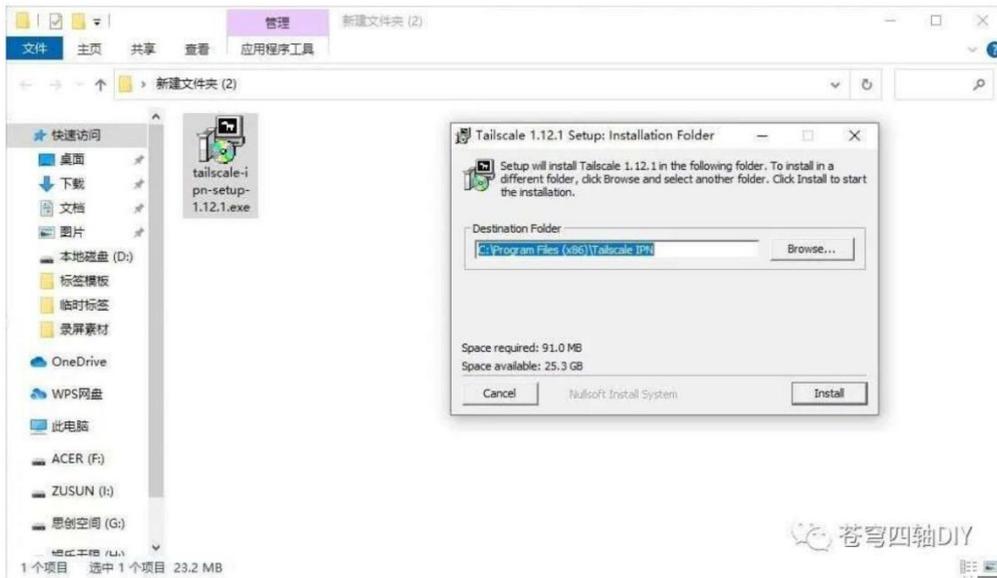
下载 win 系统.rar，解压后得到安装文件。

Tailscale 属于一种虚拟组网工具，基于 WireGuard。他能帮助我们安装 Tailscale 服务的电脑、手机、平板都放到同一个虚拟局域网中。通俗来讲也就是在家里的电脑，还有办公室的电脑，甚至云服务器都能放到同一个局域网内，进行远程访问。当然在这里我们的目的是要把笔记本电脑（操作端）和无人机（树莓派）都放在一个虚拟局域网内，以便相互通讯访问。

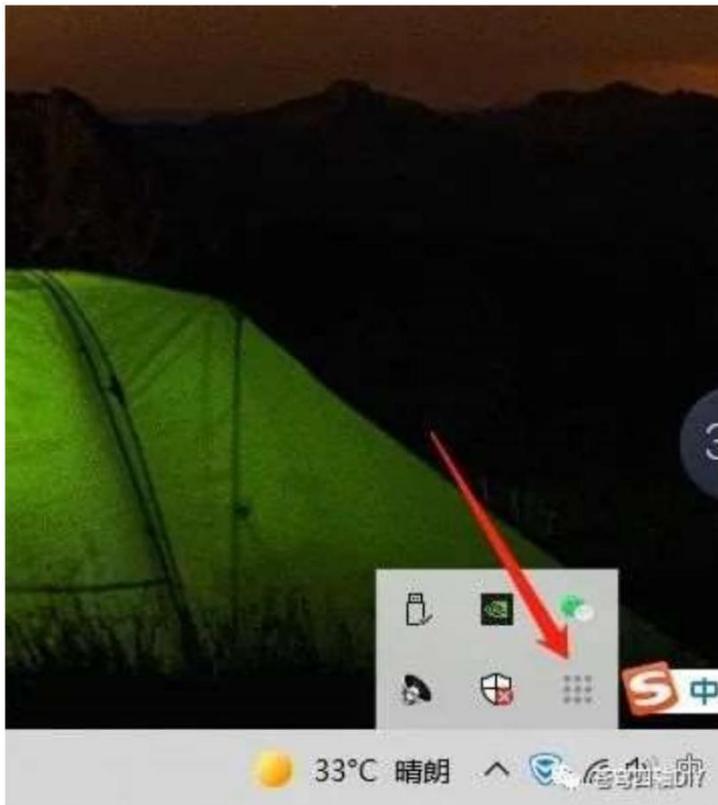
在 Windows 电脑上安装 Tailscale:

先自行登录 github 网站 (<https://github.com/>) 注册一个账号, 后面会用到;

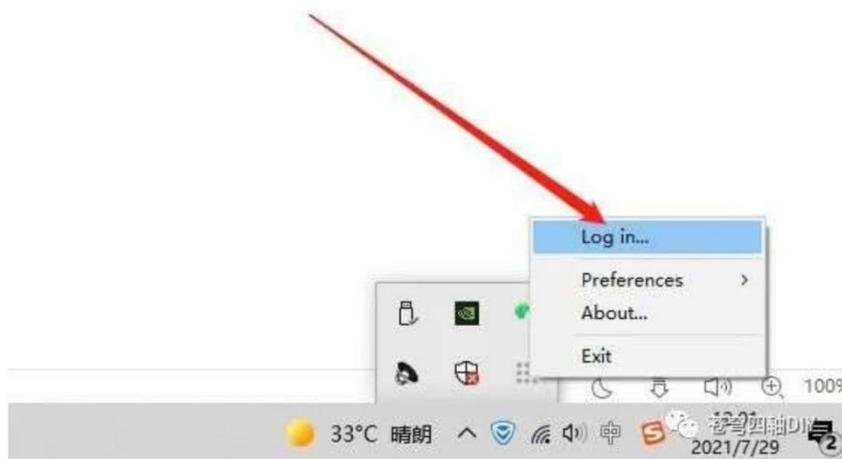
A. 双击解压后的安装文件, 默认路径安装即可。



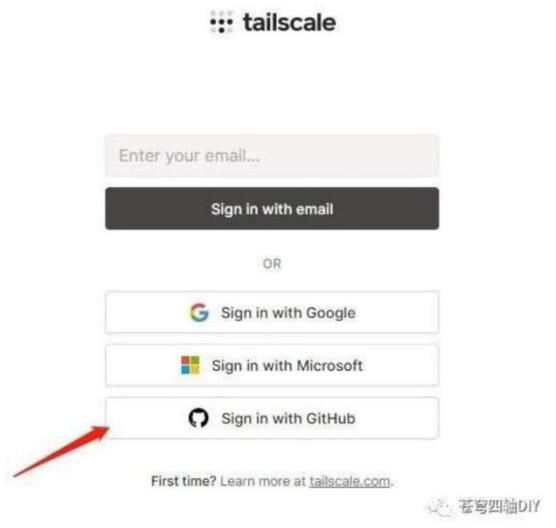
B. 安装以后重启电脑, tailscale 会自动启动, 如下所示:



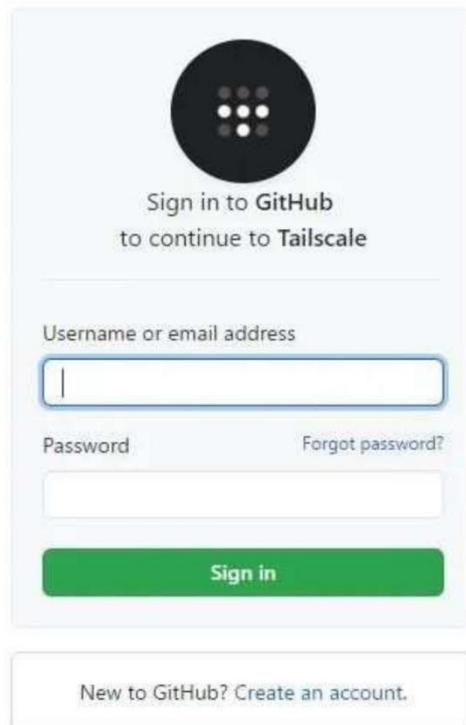
C. 点击鼠标右键，选择 LOG IN... 注册



D. 浏览器会自动进入 tailscale 官网，选择 Sign in with GitHub，使用 github 账号注册。如果有 google 或者微软账号也可以。注意不要使用邮箱注册，那是增值服务！国内由于网络原因，github 有时登录不上，多试几次。



E. 填入 github 账号和密码



Sign in to GitHub  
to continue to Tailscale

Username or email address

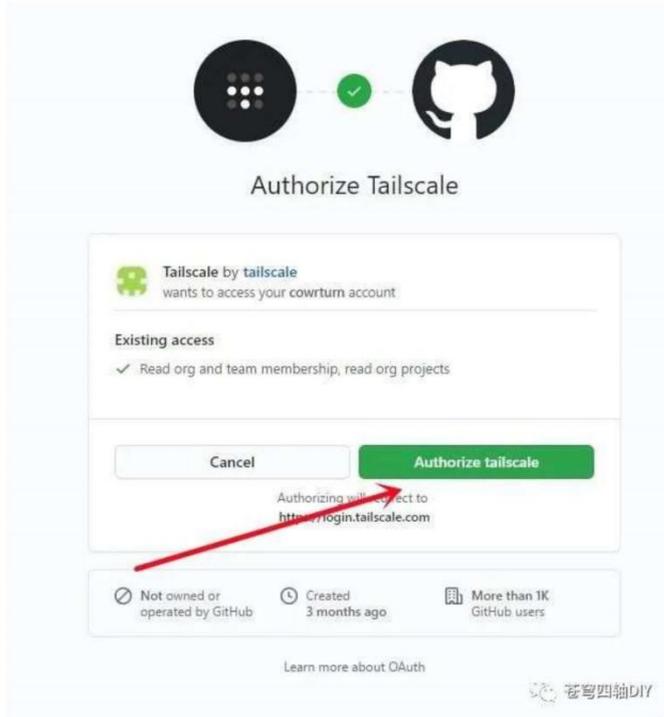
Password [Forgot password?](#)

[Sign in](#)

[New to GitHub? Create an account.](#)

👉 驾考四轴DIY

F. github 注册时留的邮箱里面会收到验证码，填入后验证即可看到以下界面：



G. 点击上图中的 Authorize tailscale，这样就成功注册好了！

H. 再次点击右下角 tailscale 图标，就能看到电脑的名称和分配的虚拟 IP 地址

I. 点击 Admin console...

J. 会进入 IP 管理页面，在这里能看到所有加入这个局域网的设备和对应的 IP 地址

K. 记录下你自己电脑对应的 IP，后面会用到。

## 10.2 无人机（树莓派）加入 Tailscale 虚拟局域网

Z410 无人机树莓派系统里我们已经安装好 Tailscale，在这里所要做的就是将树莓派加入你组建的 Tailscale 虚拟局域网中。

- ① 将无人机接上电池，电池记得插上 BB 响低电压报警器；
- ② 将 4G 模块插到树莓派的任意一个 USB 口（4G 模块需要插入 SIM 卡），树莓派能通过 4G 模块联网（SIM 卡记得取消流量限制）；





没有 4G 模块也可先使用网线插入树莓派的网口，保证树莓派能正常上网；以后再安装 4G 模块。

③ 等待大概 1~2 分钟，树莓派启动完毕，会自动生成一个名为 ACopter 的 WIFI 热点，密码：12345678。将笔记本 WIFI 连接到树莓派热点，这样笔记本与树莓派之间就组成了一个局域网。

④ 右键点击 WIN 图标，打开 Windows PowerShell（管理员）终端；或者之前你熟悉使用的终端均可；

⑤ 使用 ssh 连接树莓派，密码：123456abc

```
ssh ubuntu@rpanion
```

或者 `ssh ubuntu@10.42.0.10`

⑥ 先执行以下命令（退出此局域网也是这个命令）：

```
sudo tailscale up --force-reauth
```

⑦ 再执行：

```
sudo tailscale up
```

会得到一个网址，如下所示：

⑧ 复制得到的网址，打开笔记本浏览器中进入，使用之前相同的 github 账号登录。这样树莓派就与笔记本加入了同一个局域网。这个局域网不受物理网络的限制。

⑨ 获取树莓派的 IP 地址

```
ip addr show tailscale0
```

看树莓派 IP 的方式：

方法 1：终端输入 ifconfig 查看 tailscale 的 IP；

方法 2：进入 Tailscale 管理界面查看 rpanion 设备对应的 IP。记录下树莓派的 IP 地址，后面要反反复用到。

### 10.3 重新分配 UDP 端口，获取遥测数据和高清视频

接下来我们使用 Tailscale 得到的虚拟 IP 重新分配 UDP 端口连接地面站和回传视频：

<https://www.bilibili.com/video/BV1UF411s7Ti/>

系统调试好以后，下次操作就方便了，主要流程如下：

① 打开遥控器；

② 无人机插上 4G 模块；

③ 将电池绑扎好，无人机接上电池，电池插上 BB 响报警器；

④ 等待树莓派系统启动，大约 2 分钟左右（可通过查看笔记本网络里出现 WIFI 名称 ACopter 表示树莓派已经正常启动了）；

⑤ 启动 MP 地面站软件，使用 UDP 连接无人机和地面站，遥测数据回传完成；

⑥ 右键点击地面站飞行数据页面左上窗口设置 Video，稍等片刻等待图像正常显示；

## 11. 手机端 QGC 地面站如何通过 4G 网络连接无人机

我们也可以通过手机 QGC 地面站对无人机进行操作，比如引导模式、航点飞行。

**第一步：安装软件**

手机上需要安装安卓版的 Tailscale 和 QGC，下载链接如下：

[https://pan.baidu.com/s/loxkmpU9\\_pddWcIckc9b49g](https://pan.baidu.com/s/loxkmpU9_pddWcIckc9b49g)

提取码: 3r1r

### 第二步: 手机注册 Tailscale

手机启动 Tailscale, 通过之前的 github 账号注册, 注册完成也会得到一个虚拟 IP, 如下所示, 其中 100.90.218.62 就是我们的手机获取到的虚拟 IP:

注意! 用手机操作 Tailscale 要保持在后台运行, 不能关闭!

### 第三步: 分配 UDP 端口

笔记本连接树莓派 WIFI, 打开浏览器输入 10.42.0.10:3000, 进入 WEB 管理界面, 将手机的虚拟 IP 加入 UDP 分配端口, 端口可以是 14553, 14554 等, 只要和之前分配的不重合就可以:

### 第四步: QGC 地面站的设置

参考视频: (**注意: 此视频中**使用虚拟摇杆控制的是无人车, 无人机**不建议使用虚拟摇杆控制!!!**)

<https://www.bilibili.com/video/BV1LM4y1G7Fk/>

树莓派无人机可扩展的应用

- ① 室外 GPS 定位条件下, 设置航点自动巡航;
- ② 在树莓派上安装 ROS 和 mavros, 使用 ROS 系统控制无人机;
- ③ 在树莓派上安装 Dronekit-python, 使用 python 编程控制无人机;
- ④ 多机组网编队。。。

## 12. 免责声明与安全飞行

### 12.1 简介

非常感谢您购买 Z410-4B。在您阅读本说明书时, 请结合产品及相关配件等设备, 边阅读边操作。您在使用产品时, 如遇到困难请查阅本说明书或致电售后热线 (028-87872048) 及登陆我司官方售后交流论坛 [bbs.amovlab.com](https://bbs.amovlab.com)—《问答专区》寻求帮助, 我们将竭诚为您服务!

### 12.2 免责声明

(1) 为保护用户的合法权益，请您在使用本产品前务必仔细阅读我们随附本产品提供的说明书、免责声明和安全须知。阿木实验室保留对上述文档进行更新的权利。请您务必按照说明书和安全须知操作本产品。

(2) 本产品不适合未满 18 周岁及其他不具备完全民事行为能力的人士使用，请您避免上述人士接触本产品，在有上述人士出现的场合操作时请您格外注意。

(3) 本产品为科研教育性产品，源码开源支持二次开发，并非稳定的商业产品，使用时需有一定的基础，且稳定程度不能和商业产品相提并论。购买产品表示您知晓并理解此产品的特殊性。二次开发有一定门槛，需要使用者具备一定的开发能力、自主学习能力。

(4) 一旦开始使用本产品，即视为您已阅读、理解、认可和接受本产品的说明书、免责声明和安全须知的全部条款和内容。使用者承诺对自己的行为及因此而产生的所有后果负责。使用者承诺仅出于正当目的使用本产品，并且同意本条款及阿木实验室可能制定的任何相关政策或者准则。

(5) 在使用本产品的过程中，请您务必严格遵守并执行包括但不限于说明书和安全须知里的要求。对于违反安全须知所提示的使用行为或不可抗因素导致的一切人身伤害、事故、财产损失、法律纠纷，及其他一切造成利益冲突的不利事件，均由用户自己承担相关责任和损失，阿木实验室将不承担任何责任。

(6) 用户使用本产品直接或间接发生的任何违反法律规定的行为，阿木实验室将不承担任何责任。

(7) 在使用本产品前，请仔细阅读本文说明。一旦您开始使用本产品，表示您对本文全部内容的认可和接受。请严格遵守本文安装和使用说明。因用户不当使用、安装、私自改装等造成的任何结果和损失，阿木实验室将不承担法律责任。

### 12.3 产品使用注意事项

- (1) 产品上电前请务必保证天线完全安装，以免上电造成模块损坏。
- (2) 电源电压切勿接反，以免烧坏设备。
- (3) 图数传模块电源切勿接入串口或网口，以免造成设备损坏。

- (4) 请保持模块干燥，勿靠近水。
- (5) 请小心保存模块，勿摔，勿撞，勿跌落。
- (6) 如果是新手请在户外定点模式下进行飞行，并且定点模式要求 GPS 的卫星数在 15 颗以上。
- (7) 飞机调试时或在传感器校准时（电调）请务必卸下螺旋桨。
- (8) 飞机上电时，传感器会进行初始化，请勿移动飞机。

## 12.4 安全须知

- (1) 本产品不适合未满 18 周岁及其他不具备完全民事行为能力的人士使用。
- (2) 本产品具有较大的机身尺寸、高速旋转的螺旋桨和强大的飞行动力，在运行时具有一定的危险性。未按要求操作和使用本产品可能会发生危险和伤害。
- (3) 使用本产品时，请远离机场、铁路、高速公路、高层建筑、电线等危险环境。
- (4) 使用本产品时，请远离手机基站，大功率发射设备等高电磁干扰的环境。
- (5) 使用本产品时，请远离各种载人飞行器。
- (6) 使用本产品时，请绑定安全绳操作，避免安全事故发生。
- (7) 请勿在下雨、雷电、沙尘、雾气、下雪、大风、低温等恶劣环境使用本产品。
- (8) 在高于 3000 米海拔以上飞行时，环境因素会导致飞行性能下降，请谨慎使用本产品。
- (9) 本产品为非防水设计，请勿尝试在水面降落。
- (10) 操作本产品 in 低空飞行时，请始终保持无人机和人或动物保持 10 米以上的安全距离。
- (11) 在非人烟稀少的地区使用本产品时，请始终保持无人机在操作者目视范围内飞行。
- (12) 不要将本产品悬停或飞越人群上空，请勿以惊吓他人娱乐。
- (13) 当有围观人群靠近时，请小心操控，尽快降落离开该场所，避免发生

意外。

- (14) 请勿在儿童嬉戏的场所附近操作本产品。
- (15) 请勿使用本产品追逐交通工具或影响交通工具的正常运行。
- (16) 非极其必要情况，请勿当本产品在空中飞行时关闭电机。
- (17) 本产品不可在饮酒、疲劳、服用药物、身体不适等情况下使用。
- (18) 请在每次使用前对本产品进行检查，包括但不限于零部件的牢固度、机体和螺旋桨的裂痕和磨损、电池电量、指示灯的有效性等。当发现异常时，请立即停止使用并更换相应配件。
- (19) 工作状态异常的无人机可能会发生意外，切勿启动螺旋桨或者勉强飞行。
- (20) 请勿尝试阻止本产品工作中的任何运动部件。
- (21) 请勿改装本产品，或将本产品用于非原设计用途。
- (22) 本产品不包括载物功能，请勿用于运输用途。
- (23) 请使用原厂配件进行维护和替换。
- (24) 转借给其他人员操作，请确保操作人员理解和遵守本安全须知。
- (25) 请访问 [www.amovlab.com](http://www.amovlab.com) 获取最新的免责声明和安全须知。

## 12.5 售后服务条款

(1) 本条款仅适用于阿木实验室所生产的产品，阿木实验室通过其授权经销商销售的产品亦适用本条款。

(2) 我方提供产品的保修期是指交货后产品初步验收合格之日起 12 个月。非甲方原因产生的故障，由我方负责维修。（耗材不在保修范围内）产品在甲方使用过程中，因为机械碰撞、电路短路、软件重新修改产生的问题，我方可以根据解决问题的难易程度提供微信语音/视频的指导。

(3) 产品自购买之日起，一周内经我司核实为质量问题，由阿木实验室承担返修产品的往返快递费，购买阿木实验室产品超过一周到一年内经我司核实为质量问题，用户和我公司各自承担寄出返修产品的快递费。

(4) 返修时需提供购买凭证和保修卡或交易记录。

(5) 产品自购买之日起七天内，在正常使用情况下出现质量问题，外观无损坏，凭保修卡或购机凭证在经销商处协商可以免费更换同型号产品；经销商在收到更换产品时必须第一时间通知我公司予以备案更换。

(6) 对于自购买之日起人为损坏、改装、拆机及超过一年免费保修期的，用户必须支付往返邮费及维修成本费用。 收费标准：人工费+配件费用

(7) 为确保您的权益受到保护，并能及时有效的为您服务，请在购买阿木实验室产品时完整填写好保修卡及索要购机凭证。

(8) 请访问 [www.amovlab.com](http://www.amovlab.com) 获取完善的售后服务条款

更多信息请关注阿木实验室官网或以下微信公众号



(阿木实验室公众号二维码)



(铂贝学院二维码)



(售后客服)



**AMOVLAB**

**— 阿木实验室 —**

阿木实验室 AMOVLAB

电话：028-87872048

邮箱：[service@amovauto.com](mailto:service@amovauto.com)

地址：成都市郫都区菁蓉镇大禹东路66号