

# Mavros培训课 – 预备知识

**戚煜华**

**Tel: 18611457441**

**WeChat: qyp0210**

# 目录



1

• 简介

2

• ROS基础知识

3

• Mavlink基础知识

4

• PX4基础知识

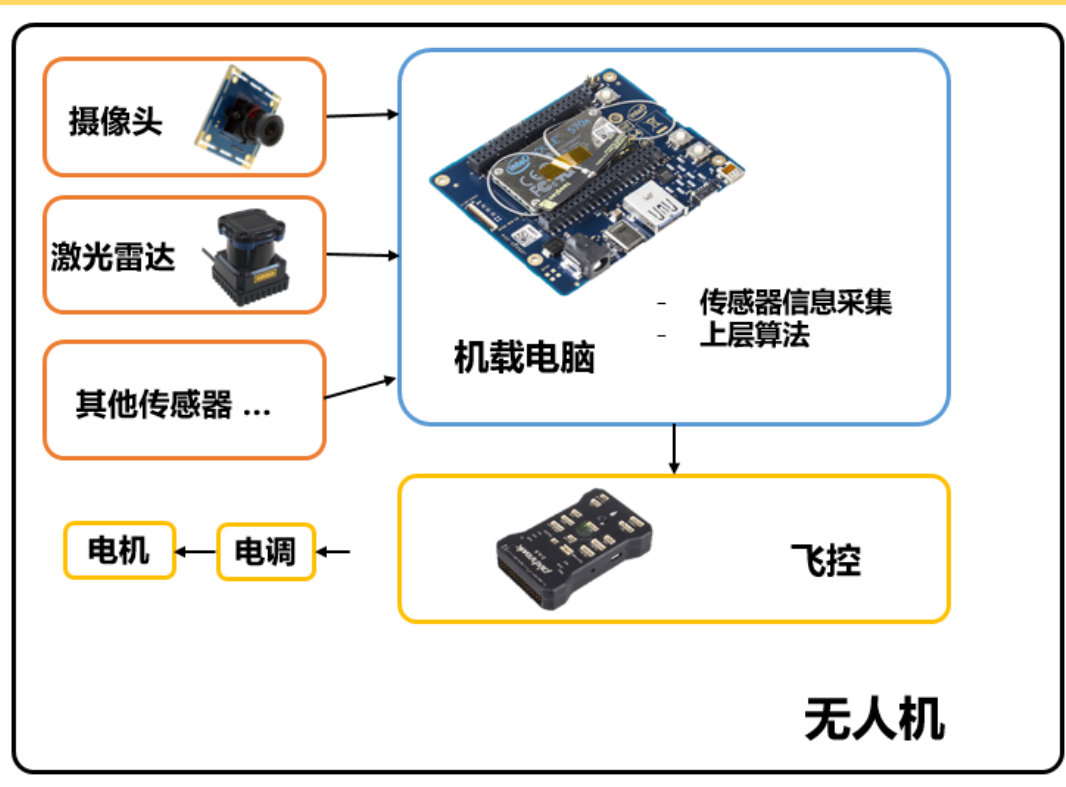
5

• 其他补充知识

# 简介 – 基础概念介绍

## 基础概念

- Pixhawk (PX4与APM)
- Mavlink
- Mavros 和 ROS
- Offboard
- 自主无人机 – Autonomous UAV



# ROS基础知识 – ROS是什么?

## ROS

Robot Operating System (ROS) 是一种得到广泛使用的机器人操作与控制系统软件框架。其提供了一个标准的操作系统环境，包括硬件抽象、底层设备控制、通用功能的实现、进程间消息转发和功能包管理等。

## 个人理解

ROS是一个平台，不仅提供各种开源代码，还提供开源代码间互相通信的机制。比如要搭建一个机器人，电机、力传感器、摄像头这些设备的底层驱动ROS都可以提供；还提供给你PID控制代码，SLAM算法等等；更牛逼的是他还写了一整套通讯协议，让你实现各个程序间的通信，你只需要下载这些功能包，写一个简单的启动脚本，就能运行起来一个机器人系统。

# ROS基础知识 – 如何自学ROS?

## 网友评论

“简单的说，ros就是一个分布式操作系统，通过节点node，主题topic，服务service等进行通信和控制。”

“所以学好ROS就是不断的了解这种通信和交流的机制，然后应用在自己的项目上就好了。”

## 个人建议

- **官网教程**
- **ROS工具书**
- **有问题先百度，不行谷歌**

# ROS基础知识 – 基本概念

## 基本概念

- **功能包** (Package) -- 一个大的文件包 例如 mavros
- **节点** (Node) -- 一个cpp文件
- **话题** (Topic) -- 一个变量
- **服务** (Service) -- 一个api函数

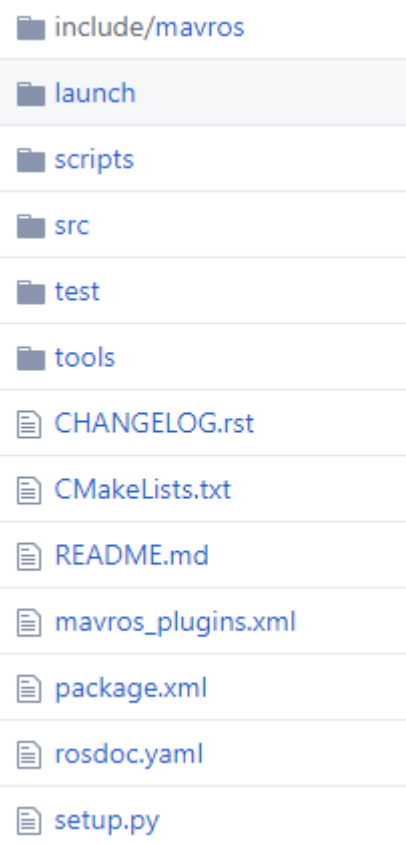
## 运行指令

- **roscore** 运行主节点
- **roslaunch mavros px4.launch**
- **roslaunch mavros offboard\_example**

- **ros中 节点之间（每个cpp之间）传递消息是通过话题（topic）的订阅和发布进行的。**

# ROS基础知识 – 基本概念

## 功能包组成



- **include/package\_name/** 这个目录包含了我们所需要的库的头文件。
- **scripts/** 其中包括Bash, Python或任何其他脚本的可执行脚本文件。
- **msg/** 如果我们需要开发非标准信息, 需要把文件放在这里。
- **src/** 这是存储程序源文件的地方。
- **srv/** 这是存储服务类型的文件。
- **CMakeLists.txt** 这是负责编译的文件
- **package.xml** 这是功能包清单文件。

# ROS基础知识 – 话题发布

## Test\_pub.cpp

```
void main()
{
// 定义发布的主题
ros::Publisher position_pub = nh.advertise< geometry_msgs::Point>("/drone/pos", 10);
geometry_msgs:: Point position_drone;
position_drone.x = 1;
position_drone.y = 0;
position_drone.z = 0;
// 发布
position_pub.publish(position_drone);
}
```

- **position\_pub 是 句柄**
- **geometry\_msgs::Point 是 话题的类型**
- **/drone/pos 是 话题的名字**



# ROS基础知识 – 话题订阅

## Test\_sub.cpp

```
//全局变量
geometry_msgs:: Point pos_drone ;
void pos_cb(const geometry_msgs:: Point ::ConstPtr &msg)
{
    pos_drone = *msg;
}
void main()
{
    // 定义发布的主题
    ros::Subscriber position_sub = nh.subscribe<geometry_msgs:: Point>("/drone/pos", 10, pos_cb);
    // 执行回调函数
    ros::spinOnce();
}
```

- 此处 话题的类型 和 名字 必须匹配，才能正确的订阅。

# ROS基础知识 – 基本命令

## 话题相关命令

- rostopic list [显示所有话题]
- rostopic info + 话题名字 [查看话题的消息类型 被哪些节点使用]
- rostopic echo + 话题名字 [查看话题具体消息内容]
- rostopic bw + 话题名字 [带宽]
- rostopic hz + 话题名字 [频率]
- ....

## 节点相关指令

- rosnode list [显示所有节点]
- rosnode info + 节点名字
- rosnode ping + 节点名字
- rosnode kill + 节点名字
- ...



## 如何入手一个新的ros功能包?

- 阅读github主页上的参考部分, 阅读ros\_wiki上的参考部分
- 打开功能包, 查看每个对应的文件夹下的文件
- 一般是运行launch文件后, rostopic list看一下所有的话题
- 针对自己感兴趣的话题, rostopic echo/info/hz 查看话题的信息
- 尝试自行在该功能包中编写节点文件



## 相关网站

- ROS官网: <http://www.ros.org/>
- ROS安装: <http://www.ros.org/install/>
- ROS官网教程: <http://wiki.ros.org/cn/ROS/Tutorials> (初学者只需掌握初级教程即可)
- 创客智造: <https://www.ncnynl.com/> (ROS实用教程)
- 易科机器人: <http://blog.exbot.net/> (ROS实用教程)
- 其他: CSDN上也有很多博主介绍某一个功能包的教程

# Mavlink基础知识 – 基本概念

## 基本概念

- Mavlink是一个用于无人机的通讯协议，在这个通讯协议下有很多很多消息类型。
- 外部与飞控建立连接，绝大部分是利用Mavlink协议，传递的是Mavlink消息。比如：地面站、mavros





## 常用网站

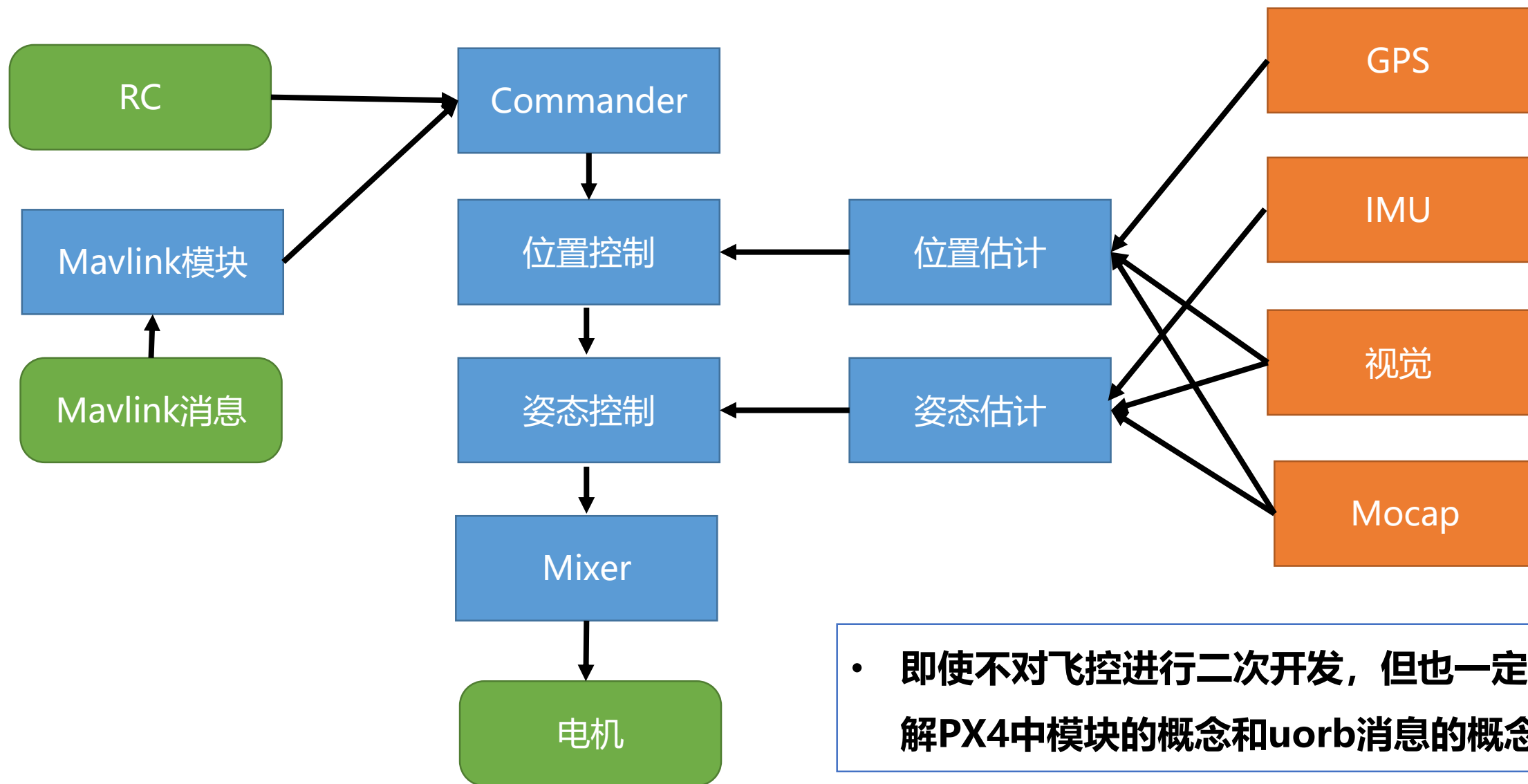
- Mavlink官网: <http://qgroundcontrol.org/mavlink/start>
- Mavlink消息查询: <http://mavlink.org/messages/common>
- 查询LOCAL\_POSITION\_NED ( #32 ) [这条指令包含了无人机ned位置信息]
- 查询 ( #84 ) [机载电脑给无人机发送的控制指令]

# PX4基础知识 – PX4代码框架

## PX4代码框架

- 开发环境：windows、linux（双系统or虚拟机）、mac 【推荐linux或mac】
- uORB消息 `Firmware / msg /` 这个目录下定义了所有uORB消息的格式
- 模块： `Firmware / src / modules /` 这个目录下的每个文件夹都代表一个模块，负责飞控系统的一个功能。
- 编译脚本 `Firmware / cmake / configs / nuttx_px4fmu-v2_default.cmake`
- 启动脚本 `Firmware / ROMFS / px4fmu_common / init.d / rc.mc_apps`
- 开发：如何新建一个module，msg等等

# PX4基础知识 – 主模块关系



- 即使不对飞控进行二次开发，但也一定要理解PX4中模块的概念和uorb消息的概念



# PX4基础知识 – mavlink模块

## Mavlink模块

- Mavlink模块负责mavlink相关的参数设置，mavlink消息的解压和收取、mavlink消息的打包和发送。这样使得我们如果想使用发过来的消息，直接去订阅相关的topic就行。
- 由于Mavlink消息封装还是较为复杂、解包也需要一定的时间，所以对于一些需要低延时的命令可以不走Mavlink协议，直接通过数传给飞控发送一些消息也可以，就是自己要在飞控端写接收的模块

- 简单来讲，负责mavlink消息的收发。



# PX4基础知识 – mavlink模块

## Mavlink 传输模式

- Normal
- **Onboard**
- OSD
- Magic
- Config

## 与mavros相关的文件

- mavlink\_main.cpp 主cpp
- mavlink\_message.cpp 负责发送mavlink消息
- mavlink\_receiver.cpp 负责接收mavlink消息

# PX4基础知识 – 位置、姿态控制模块

## 位置控制与姿态控制

- 位置控制
- 输入：期望位置 当前位置
- 输出：期望姿态角
- 姿态控制
- 输入：期望姿态角 当前姿态
- 输出：期望力矩和期望推力

## offboard模式

- 控制模块中提供以下接口给offboard模式
- 期望位置
- 期望速度
- 期望加速度
- 期望姿态角
- 期望力矩和期望推力

- 对控制输入输出不理解的同学，可以课后找一些无人机建模控制的论文读读

# PX4基础知识 – 位置、姿态估计模块

## attitude\_estimator\_q.cpp

- 姿态估计模块（多传感器融合）
- 输入：各种传感器的数值
- 输出：无人机姿态（欧拉角 角速度）
- mocap
- vision
- 直接赋值

## position\_estimator\_inav.cpp和lpe.cpp

- 位置估计模块（多传感器融合）
- 输入：各种传感器的数值
- 输出：无人机位置（位置和速度）
- mocap
- vision



## 地面站

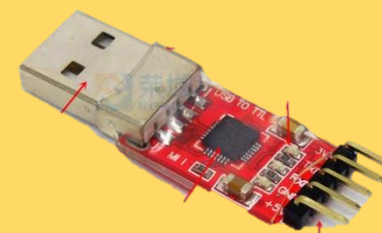
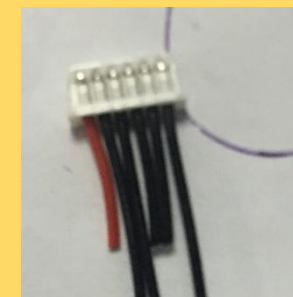
- 会用地面站刷固件
- 会用地面站对一个新的飞控进行一些初始设置（设置机型、校准传感器、校准遥控器）
- 电调校准
- 设置飞控中的参数
- 监控飞控状态

## 地面站参数

- 地面站参数修改：  
SYS\_COMPANION 修改为  
Companion Link(921600,8N1)

## ttl转usb模块线

- 用于连接机载电脑和无人机
- 连接方式：最靠近红色线的黑线连模块的Rx端
- 次靠近的黑线连模块的Tx端
- 地线接GND
- 插在飞控的TELEM2口



全功能5P插针



## 常用网站

- Pixhawk官网: [www.pixhawk.com](http://www.pixhawk.com)
- PX4开发手册: [dev.px4.io](http://dev.px4.io)
- 官方论坛: [discuss.px4.io](http://discuss.px4.io)
- PX4代码: [github.com/PX4/Firmware](https://github.com/PX4/Firmware)
- 各种民间大神的博客: [http://blog.csdn.net/qg\\_21842557/article/details/50884695](http://blog.csdn.net/qg_21842557/article/details/50884695) (summer) 、 <http://blog.csdn.net/oqqenvy12/article/details/70247988> (Fantasy) 、 <http://blog.csdn.net/luoshi006/article/category/6389437> (洛神)

# 提问环节