

# 培训方案

本文档为2023年河北工业大学航模协会程序与电控培养计划

作者：赵桂龙

QQ：2990041462

2023-10-27

## 说在最前头

我们每个人都被教导用最小的投入获得最大的收益，但这件事在团队合作上却有些失效了，它只会导致团队任务的全盘崩溃。

“我是搞结构的为什么要了解电控啊？”

“我是写视觉的为什么要了解ROS是啥啊？”

“我程序设计的没有问题，结构出错了都是ta的问题。”

“这比赛连个评级都没有，对我保研一点用处都没有，摆了摆了。”

学长送给你们一个宝贵经验，想保研就把时间放在卷绩点上，技术类比赛加分没有创新创业类高（没错是Idea和PPT大赛）

一个人不可能把所有事情都做的尽善尽美

选择把精力投入在技术上，必然会失去一些其他东西（比如快乐游戏时光，不一定存在的甜甜的恋爱，还有你内卷绩点的时间）

抱着虚假的努力天天刷手机可不会进步，  
大学也不再适用于高中那套“唯成绩论”，  
功利的想法可以理解，想抱大腿人之常情，  
但我还是相信要先做好人，才能做好人。

理想的团队环境：

为了意见分歧吵架，因为项目进展欢呼、  
大家都是顶梁柱，“都行都行”做不得数、  
遇到倔驴要牵住，及时沟通互帮互助

世界的中心不再是“我自己”，没人会再拽着你学啦，

下面只是一些方向性的分享，学习过程中可能必然会遇到各种各样的问题，请在学习上拒绝社恐，阳光

开朗，如同社交恐怖分子一般，寻求帮助。

最后，请珍惜那些乐于助人的学姐学姐

# 综述

本方案面向CUADC的对地、多旋翼、火箭项目。

涵盖Linux使用、视觉、程序控制飞行、飞行控制器、嵌入式开发等内容。

## 分项目介绍

### 固定翼侦查与定点打击（对地）

本项目的任务为：飞机自动起飞，自动巡航侦查地面靶标，识别靶标上的数字，打击中位数所在靶标，自动降落。

涉及到的技术有：

1. 飞行控制器（Ardupilot固件）的参数调整（有关如何启用各种传感器，配置云台、配置飞控与上位机通信、自动起飞、自动降落、设置巡航速度与半径、遥控器通道的设置、飞行模式配置等）
2. 飞行航点（Ardupilot固件）的绘制（有关起飞，普通航点导航，接受半径，改速，降落点等）
3. OpenCV处理图像（有关拍摄图像的筛选、预处理等）
4. 基于机器学习的目标检测（用于检测靶标在图像中位置，有关YOLO的训练、部署，基于不同平台的加速等）
5. 基于机器学习的数字识别（使用CNN等方式，去年采用了LeNet5，效果一般，有待探索）
6. 基于ROS控制飞行控制器（包含从飞控获取飞行实时数据、控制飞控解锁、切换飞行模式、上传或切换航点等操作）
7. 基于我也不晓得改用啥的图像内靶标的坐标到经纬度转换
8. 基于我同样不晓得用啥的载荷弹道解算
9. Linux平台的使用（机载电脑使用Linux作为操作系统，开发程序环境也是Linux）

### 多旋翼无人机任务飞行

本项目的任务为：飞机自动起飞，自动飞向指定投放区域，将搭载的水瓶投放到指定的圆筒内（桶白，白的有点灰，跟水泥地似的）再前往侦查区域，侦查危险标识，最后返回并自动降落在指定的降落标志上。

涉及到的技术有：

1. 飞行控制器参数调整同上（添加RTK配置）
2. 飞行航点的绘制同上
3. OpenCV处理图像同上
4. 基于机器学习的目标检测同上（用于查找桶的位置）
5. 基于ROS控制飞行控制器（从飞控获取位置信息和朝向、切换模式与解锁、offboard模式微调无人机位置）
6. 基于PID（可能）的offboard位置微调控制
7. Linux平台的使用（这个逃不掉的）

## 航天火箭发射与返回

本项目的任务为：火箭携带一枚鸡蛋作为载荷，同时携带赛方规定的气压高度计，在指定高度（40m）进行二级分离，一二级打开降落伞并回收。

在电控上，该项目需要设计一个能实时确定高度，点燃开伞药或驱动其他机构的控制电路与配套程序。

定高可采用气压计、加速度计等传感器的数据融合实现。由于传感器采集数据受到环境因素影响会有波动，还需要采用诸如一阶低通滤波、卡尔曼滤波等算法去对数据进行处理。

涉及到的技术有：

1. 常见元器件的作用与元器件选型
2. 原理图绘制与PCB设计
3. 阅读芯片DataSheet的能力
4. 根据手册编写芯片驱动程序
5. 滤波算法与数据融合算法

## 技术方向总结

### First OF ALL

### 提问

学会提问，拒绝做伸手党，拒绝不劳而获。

在你觉得一件事情很麻烦，想去麻烦你的学长学姐之前，请把这个读了。

[Stop-Ask-Questions-The-Stupid-Ways](#)

**实在不行，把这个读了**

[关于提问这件事](#)

这个选读（推荐读一读）

[How-To-Ask-Questions-The-Smart-Way](#)

自查表：

下面的三个F是什么意思

- RTFM
- STFW
- RTFSC

## 搜索

好了我假设你至少看过一篇文章了，那么接下来，请停止使用充满了莆田系广告的百度，转而拥抱暂且不太作恶的Bing，如果有**科学上网**的手段，也可以使用Google。同时把搜索框里的话换成英文，这能很好的解决诸如“为什么我就搜不到？”的问题。

## 科学上网

搞程序的如果不会科学上网如同自断一臂，毕竟现在连全球最大的同性交友开源代码托管平台都被墙了。等你用上Linux，发现下载某些东西的时候很慢你就懂了。

## Linux

对于对地和多旋翼，大部分内容相通，都包含视觉、飞行控制以及ROS控制程序的开发。

其中最**重要**也最**基本**的是对Linux的使用，不论是程序的开发过程，还是最终将程序部署到机载电脑上，都离不开程序的运行环境配置、操作系统自身的配置等。

这里给出一些需要学习的知识点：

- **不要乱抄CSDN上看不懂的东西**
- Linux系统的安装（实体机或虚拟机）
- Linux Shell入门，推荐[The missing Semester](#)
- 换源（为了加快软件下载速度）
- Linux的常见命令、权限控制(sudo)、文件系统节点等（入门就足够啦）
- Linux环境变量（.bashrc等）

自查表：

- 如何安装/删除一个软件
- 什么是家目录
- 如何创建/删除/重命名一个文件（夹）
- 什么是超级用户和普通用户

Linux的东西很多很杂，更推荐边用边学，记住Linux永远是简单优雅的，如果你发现自己需要做的某件事不够简单优雅，那么一定有更好的办法去实现它。

CSDN上固然有不少好帖子，但他也有很多过时的和错误的内容，这对小白实在是致命的，中文的Linux社区生态远不如英文社区，所以请在装系统的时候把语言设置为英文，不然出错了都搜不到。

机载电脑性能有限，为了开发效率，开发时更推荐在自己的电脑上装一个Linux系统，前期用**虚拟机**就好，推荐Vmware，用的人多，出错好找问题。

灵活运用虚拟机的快照功能，在又双叒叕把系统搞坏的时候（对初学者这是必然事件）可以快速回到搞坏之前，所付出的仅仅是一点磁盘空间。

## 视觉开发

视觉开发大体包含两个内容，**图像处理与机器视觉**。

**图像处理**学习**OpenCV**就可以了，网上大多数OpenCV的课程是用Python讲的，更容易懂一些，但由于后面要用到ROS，它是要用到C++进行开发的，所以还需要花费一点的学习成本去阅读OpenCV的API文档（真的写的稀烂）

**机器视觉**这部分有两个任务，数字识别简单些，目标检测复杂些，推荐先前者再后者，前者的入门课程可以推荐一点，[点我进入Bilibili大学](#)，当然这点还远远不够，可以再学习些pytorch相关的内容。

目标检测可以使用**YOLO**，毕竟这是时下最棒的目标检测模型了，网上大多教程都是讲yoloV5的，然而yoloV8都出来了（汗），不论是准确度还是性能都有很大提升，所以更推荐V8。

如果看不懂的话，建议从b站上找同济子豪兄讲yoloV1的视频，大体了解下yolo的原理，以及它的输出层结构、后处理的方法等。

## 飞控调参和航线规划

关于飞控的参数，最权威的教程是官方**英文**文档，中英文的内容是不一样的，只看中文有大坑！！！！

根据飞控使用的固件去查看对应的文档

[PX4](#)

这里给出几个主题，根据这几个主题去查阅文档相关部分，等你把这些东西在一架飞机上调通了，这部分你就毕业了。

- 空速计、激光雷达的启用和配置
- 数传的配置
- 机载电脑串口配置
- 航点接受半径和巡航速度
- 飞行模式(takeoff、manual、mission或auto、自稳)
- 任务模式自动起飞与自动降落
- 云台自稳
- 遥控器配置

关于航线设置也有一些坑

航线设置用的软件是QGC或Mission Planner这里建议两个都用，因为各有优缺点  
记得给Mission Planner的地图源换成Bing卫星地图，这个比较新。

以下是要注意的一些点：

- 尽量不要设置锐角转弯，否则飞机可能原地打转
- 速度修改航点是附着在普通航点上的，如果当前飞机无法到达这个普通航点的接受半径（比如设置的一连串航点转弯半径太小）飞机就不会修改速度，所以建议在飞行途中多设置几个，确保飞机一定能经过某个改速航点。
- 飞机转弯时会掉高，转弯越急，掉高越狠，注意留有余量。
- 降落时要提前降低高度，在land点前的最后一个航线之前，一定要将飞机方向领正，否则轻则机毁，重则炸坏电件。

## 飞行控制程序、坐标反推，弹道解算

这些是我们23年第一年参赛尚未解决的问题，在这里我只能给出一些方向性的说明。

这里的飞行控制程序，指的是如何让机载电脑控制飞机的模式、解锁与上锁、航线更新、从飞控获取飞行时数据，根据数据配合视觉结果，解算靶标经纬度，调整飞行姿态，自动规划新航线的程序的统称。

根据飞控文档，官方推荐的做法是使用ROS开发，我没用过ROS，航协甚至都没人用过ROS，这部分需要同学们自行探索去实现。

但毋庸置疑的是，ROS的使用对大家的Linux水平是个考验。

我们23年的方案是使用mavSDK实现的，这东西问题很多，所以还是坚定的拥抱ROS吧！

# 嵌入式开发

这部分我不是行家，只能给大家介绍一点

传统的学习路线大概是51->32，但生活在2023年的我们其实拥有更加新手友好的选择。

为什么不从Arduino开始呢？

这个东西简单到是可以给小孩子玩的，对于嵌入式学习，我更倾向于采用项目驱动型学习，即拿着身边现有的各种模块，上网学习也好，查看示例代码也好，先想办法把他们驱动起来，然后去了解对应的原理，

等你能搜到的一些简单的东西都驱动过了，你就对嵌入式开发的一些基础知识，如GPIO，PWM，I2C，SPI等不再陌生了。

对做东西而言，不必死扣这些协议背后的底层原理，只要知道它“是什么，怎么用”，实在不行再去探索背后的实现原理。

采用Arduino开发，在程序开发难度方面，有着显而易见的优势（各种库函数直接拿来用，等你用上了就懂了）

在硬件上，很容易就能买到十几块钱甚至几块钱的支持Arduino的开发板，如ESP8266，ESP32等（十几块钱还带wifi），比动辄三五百的stm32便宜到不知哪里去了。等入门后，倘若真的需要，在掏钱包也不迟。

等玩过很多模块后，就可以开始学习怎么做自己的PCB了，在B站搜索立创EDA，稍微看上那么几节课，然后开始抄别人的电路，可以在B站找感兴趣的，也可以直接去立创开源广场找

进阶一点的，开始抄芯片DataSheet的示例电路，反正每个月都可以在嘉立创免费打样两次电路板，哪怕从打别人的板子开始也好。

推荐从ams1117这个芯片开始，因为他实在是太简单了。

抄着抄着就会了（确信）