

从零开始 DIY 小四轴

首先，DIY 最重要的是自己动手的精神和永远好奇的心态。这样才能不断进步。

其次，小四轴虽小，确是一个综合学科实力的体现。也正因为如此，它带来的挑战才能带给我们无穷的乐趣。

网友 hitor 说：

这里我要声明一下 DIY 开源四轴的意义。虽然我们利用开源的程序毕竟不如自己去写代码、研究算法有意义，但我认为对于新手来说，我们所掌握的知识 and 技能有限，不是每个人都对捷联惯导、多传感器数据融合算法（互补滤波、卡尔曼滤波）、PID 控制算法、神经网络控制算法等感兴趣和熟知。如果从一开始就要独立完成一套四轴系统，并让其稳定的飞起来，这是很有难度的而且会耗费我们大量的时间却不一定有很多收获。所以我们可以先利用别人的开源代码，让自己的四轴飞起来，我们转过头来再去研究他们的代码，把他们的代码研究透了，我们再写出属于自己的代码。其实这是一个学习的过程，这样你研发的周期缩短了，不易走弯路，不易走到死胡同里转不出来，从而丧失了继续研究的信心。

折腾就是 DIYer 的美德，浪漫就是浪费时间慢慢磨。

想起纳尔逊·曼德拉的《漫漫自由路》。

再次，让我们谈谈独立完成一个小四轴可能需要的一些知识：

一些基本的数学知识，程序员都需要的。

一些基本的物理知识，至少听见角速度，加速度不太头疼。

如果能看英文资料，你的视野将大大开拓。

一些基本的电子线路知识，不知道 MOS 管，驱动，负载什么的，肯定是混不下去的。

自己动手焊接是 DIY 的标志，尽管一些特别小的，或者某些封装，确实要拜托机器的，还是让机器来吧；

如果能画 PCB Layout，那太好了；不过自制 PCB，去腐蚀啥的，我就不碰了。

基本的控制原理，负反馈，PID 什么的等等。

单片机原理和编程，至少学过 51 吧，其实还是推荐 AVR，STM32 等。

常见开发板听过吗？Arduino，Raspery Pi，其他 AVR 和 STM32 开发板，比如战舰，好的开发板能让学习 MCU 的过程事半功倍。

简单的机械原理；电机的基本原理；一些传感器的基本原理；这些其实中学物理都教过了。

如果要自己写上位机软件：那么 C#，QT，Java 什么的得会一样吧，不过我不准备自己写了，就用网上各位大神的吧。

任何事情，都是坚持才能有收获。只要有滴水穿石，日积月累的毅力，每个人都可以实现自己的创意，尽管有时候只是小小的创意或者你只是感觉把这件事做成了，重要的是过程，如此而已。

我就是从零开始做小四轴，立此存照。先学 STM32，“Hello，World”。

行知创客 <http://weibo.com/u/1038426340>