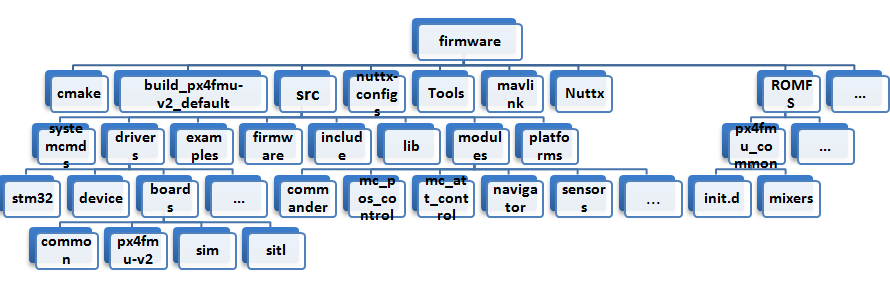
1.1Px4自动驾驶仪使用方法

PX4自动驾驶仪作为当今最为成熟的开源飞控，可以在GitHub中搜索PX4/Firmware获得。作为完整的开源飞控解决方案，其中不仅包含了自驾仪主体的代码，还包含仿真测试模块，与各种工具模块。其配套地面站为qgroundcontrol，同样在GitHub中搜索mavlink/qgroundcontrol可以获取。PX4的底层代码由C语言所写，主体代码为C++所写，仿真模块由java所写，各种工具模块为python所写，编译需安装CMake、Java sdk和PX4 tool chain。很多大型工程为追求更好的开发和维护都用到了Makefile文件，PX4开发所使用的Cmake是比makefile高一级的编译配置工具，便于多语言的混合编译。编译过程从根目录下的Makefile开始，在cmake/configs文件夹下包含着各种型号的飞控板与自驾仪版本所对应的cmake文件，在这里我们选着对应的编译文件为nuttx\_px4fmu-v2\_default.cmake，其中包含着cmake编译所需工具、硬件驱动、nuttx实时操作系统、中间件与自驾仪的各个模块，在此文件中包含着各个模块所对应的文件夹路径，每个文件夹中都会有各自的cmake文件，包含着各自文件夹中需要编译的文件。CMake是一个跨平台的编译工具，它并不编译出最终的软件，而是根据不同的平台（如LINUX、POSIX、Mac OS 、Windows等）产生标准的构建档，用来描述工程中各个文件的编译顺序与链接方式，与PX4 ToolChain配合，调用PX4 Toolchain中的GNU编译器对C、C++、Java等语言的文件进行交叉编译。



PX4自动驾驶软件可分为三大部分：实时操作系统、中间件和飞行控制栈。Nuttx实时操作系统提供POSIX-style的用户操作环境，进行底层的任务调度。PX4中间件运行与操作系统之上，提供设备驱动和一个微对象请求代理（micro object request broker,uORB）用于驾驶仪上运行的单个任务之间的异步通信。Px4的飞行控制栈运行于Px4中间件之上，经过层层封装此部分专注于飞行所需的各个线层，是自驾仪主要的逻辑所在。



PX4代码的文件结构：CMake为混合编译时的组态档，控制整个工程的编译内容、顺序与链接方式。Tools包含PX4的各种工具，比如编译所需的下载工具等。Mavlink是同一实验室专为无人机无人车等设备发布的通讯协议，在此项目中负责无人机与地面站的通讯。Nuttx是nuttx实时操作系统的源代码。ROMFS为将烧录进飞控ROM的文件，包括软件启动文件与电机作动器混频方式的配置文件。Src为Px4的源码主要放置文件夹，其中systemcmds主要放置系统工具，能够通过存放在ROMFS中的启动文件启动，也能在飞控连接计算机时，在nsh串口控制台中用命令行进行调用，功能包括控制I2C、查看修改参数、查看软件版本、校准电调、查看系统性能、bootloader升级等工具。Platforms是在nuttx系统之上编写的中间件与编程环境，封装了nuttx系统，定义了PX4平台自己的接口，与nuttx操作系统分离，便于移植到其它系统。Drivers中包含各种硬件的驱动，其中boards中定义了各种型号板子的接口配置以及相应配置接口（LED、PWM、USB、定时器等），device中定义了各种外设的基类，如I2C、SPI、ringbuffer等，stm32中包括adc的基类、高精度定时器、伺服控制的驱动程序。modules中包含了最上层飞行栈中各个不同的模块，包括commder状态机、navigator导航模块、mc\_pos\_control位置控制模块、mc\_att\_control姿态控制模块等。Lib中包含了px4平台使用的各种库文件，如数学工具库、地理坐标转换工具库、RC信号解码库等。

Px4飞行栈主要可以分为四部分，决策导航部分，包含模块主要有commander和navigator，根据飞行器自身安全状态和接收到的命令，决定工作于什么模式，下一步的目标位置速度与姿态应该是怎么样的；位置姿态估计部分,主要有sensors、position\_control、attitude\_estimator，根据各种传感器融合得到自身的位置和姿态信息；位置姿态控制部分，主要有mc\_pos\_control、mc\_att\_control，根据期望位置和姿态按照设定好的控制律，尽可能快且稳定的到达期望位置和姿态。控制器输出部分，包括mixer、执行器与pwm限幅。

