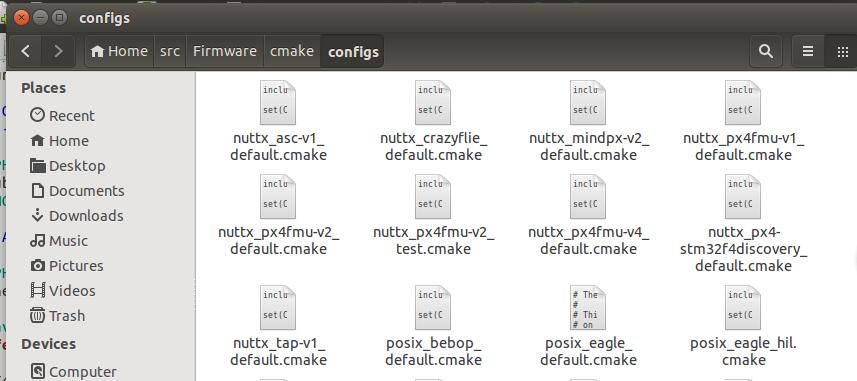
**编译脚本和启动脚本的修改**

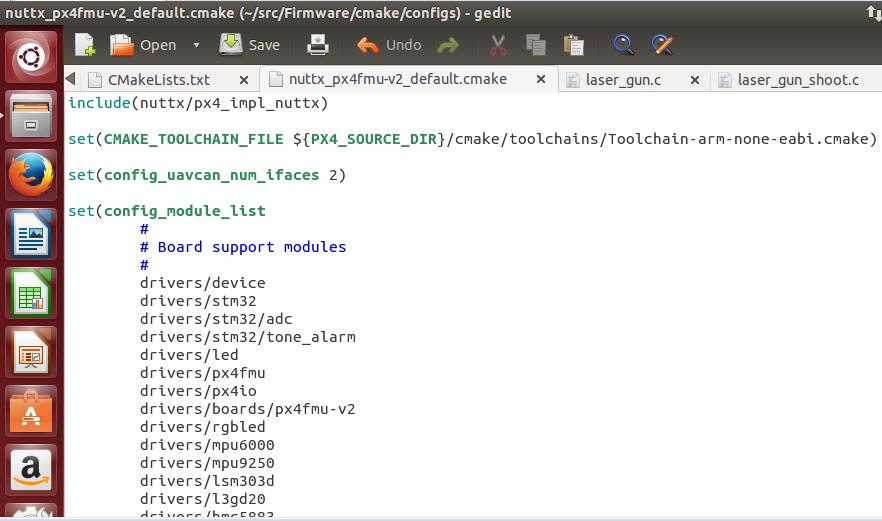
这个PIXHAWK原生固件采用的Cmake编译脚本的书写方式，降低makefile的书写难度。我们只要掌握几个修改脚本就可以掌握系统的编译。

**1 总的模块编译脚本的修改**



其中src/Firmware/cmake/configs是飞控硬件选择文件夹，比如有px4fmu-v2,px4fmu-v1的不同硬件可以选择。这个文件就是配置相应的飞控硬件，要编译什么模块，填写好正确的模块路径，系统就会编译里面的源码。

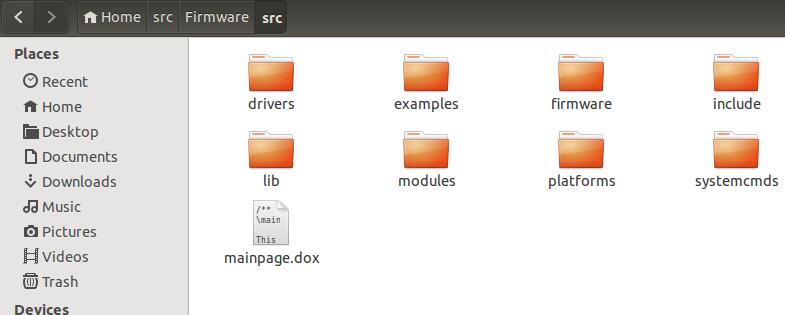
比如下图:



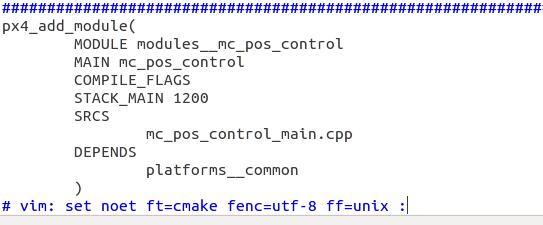
在set(config\_module\_list 中添加要编译的子模块。

**2子模块的CmakeList.txt的修改**

前面配置好的总的编译模块添加了，编译路径之后，实际上编译器是去寻找子模块下面的CmakeList.txt去进一步编译



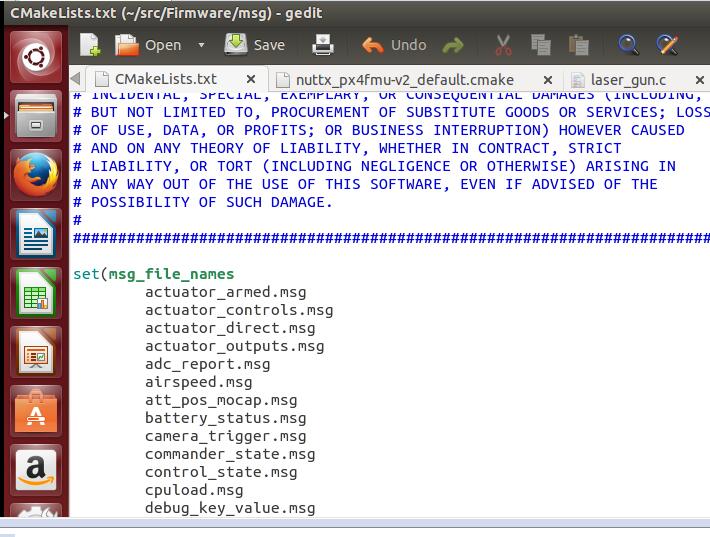
这个src目录就是源码，里面有所有的子目录都有CmakeList.txt编译脚本文件，我们一般在modules里面添加我们自定义的代码，drivers是传感器驱动目录，lib是库目录，systemcmds是系统命令。



按照如上添加修改CmakeList.txt即可。

**3 msg消息主题的添加**

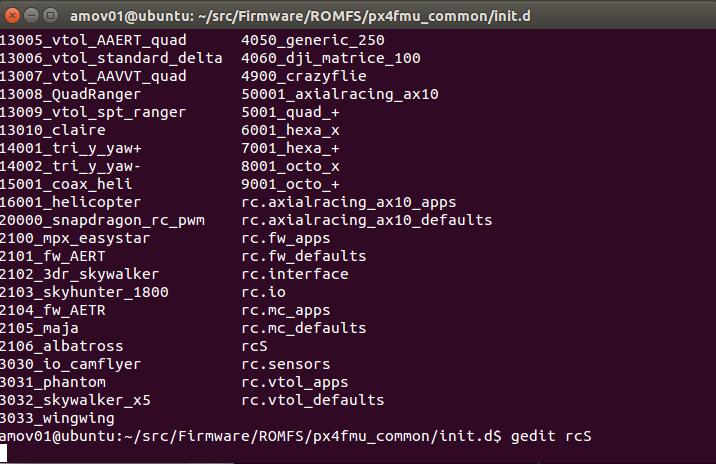
在源码目录msg目录添加，自定义的msg消息成员变量。在msg同目录的修改CmakeList.txt



把你自定义的msg消息添加进去即可。这时候编译make px4fmu-v2\_default后，在

/src/Firmware/build\_px4fmu-v2\_default/src/modules/uORB/topics 这个cmake的bulid目录自动产生相应的自定义msg消息.h头文件。这是后在自己的应用代码中可以调用这个头文件，使用自定义的消息了。

**4 启动脚本修改**



以rc为开头的都是飞控的启动脚本的文件，通过这些文件决定那些应用程序应该执行，那些不执行，pixhawk支持多种类型的飞行器，四轴，固定翼，直升机等等。就是通过这些rc启动脚本指定。比如如果是rc.fw\_defaults是固定翼飞机的脚本，如果飞机是多旋翼这个rc.fw\_defaults是不会执行的，而是执行rc.mc\_defaults这个多旋翼脚本。如果我们的飞机是多旋翼，我们一般在修改rc.mc\_apps了。把相应的应用程序在这个脚本中 XXX start即可。