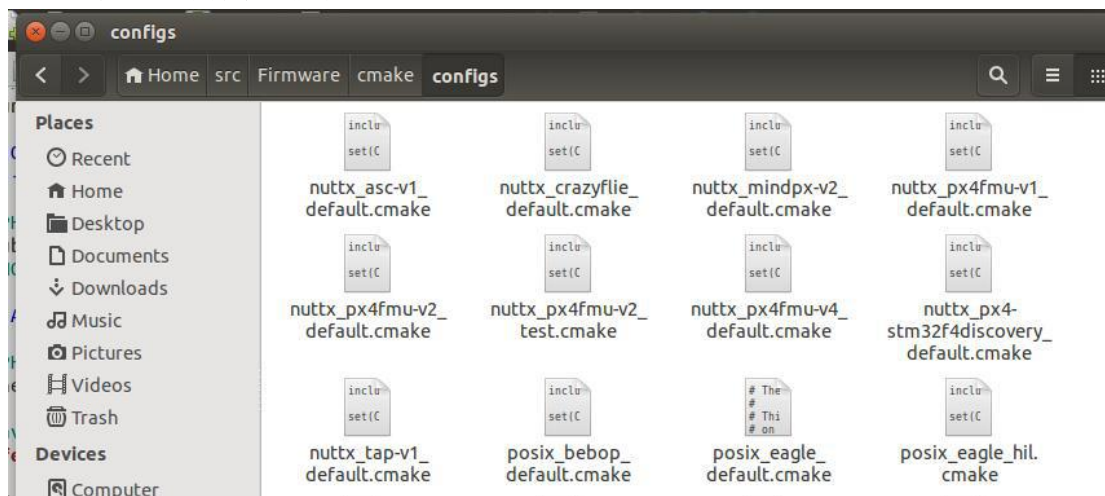


## 编译脚本和启动脚本的修改

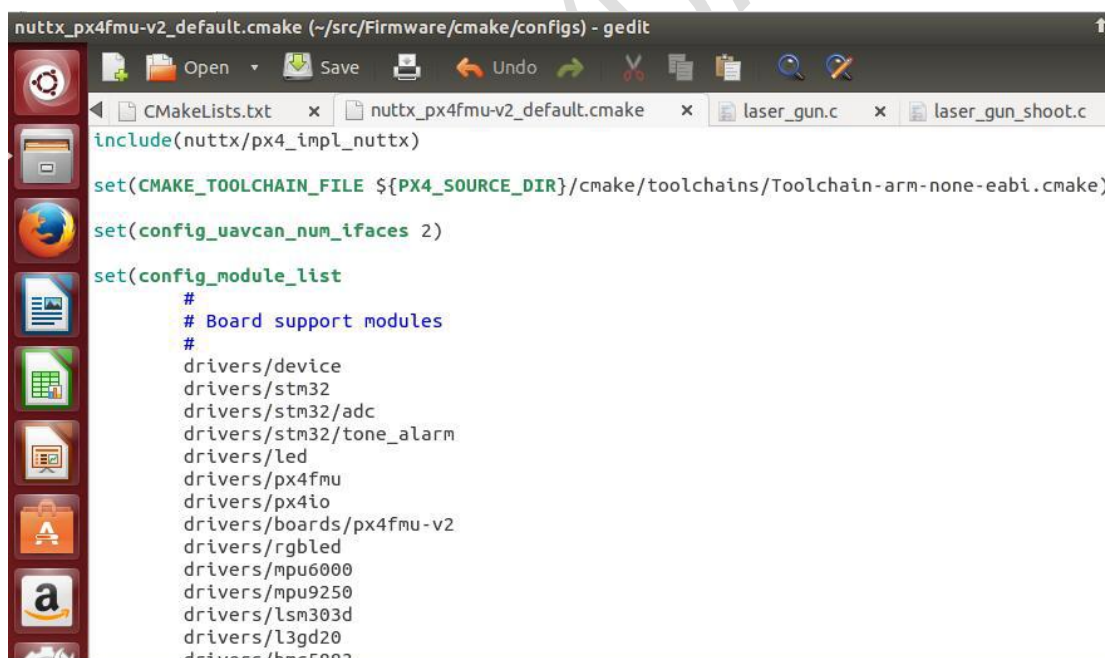
这个 PIXHAWK 原生固件采用的 Cmake 编译脚本的书写方式，降低 makefile 的书写难度。我们只要掌握几个修改脚本就可以掌握系统的编译。

### 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。

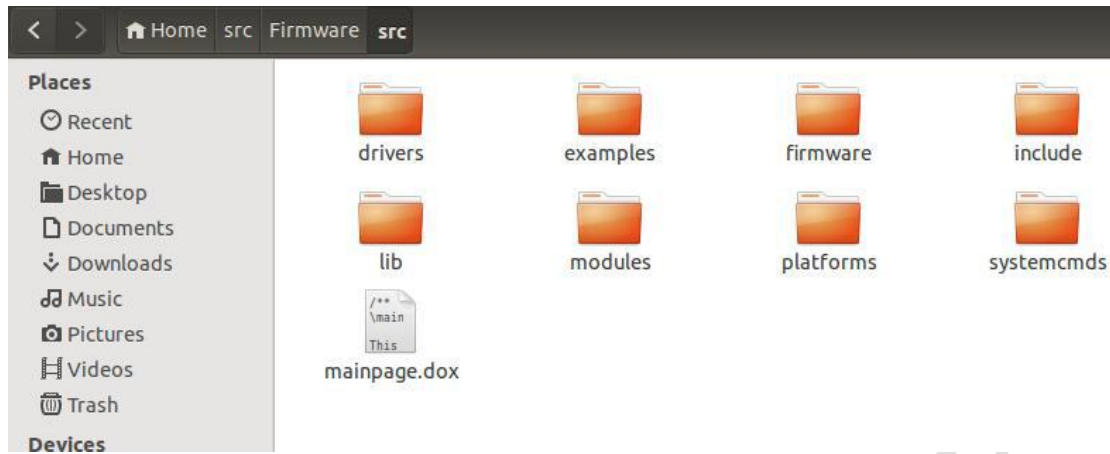
比如下图：



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



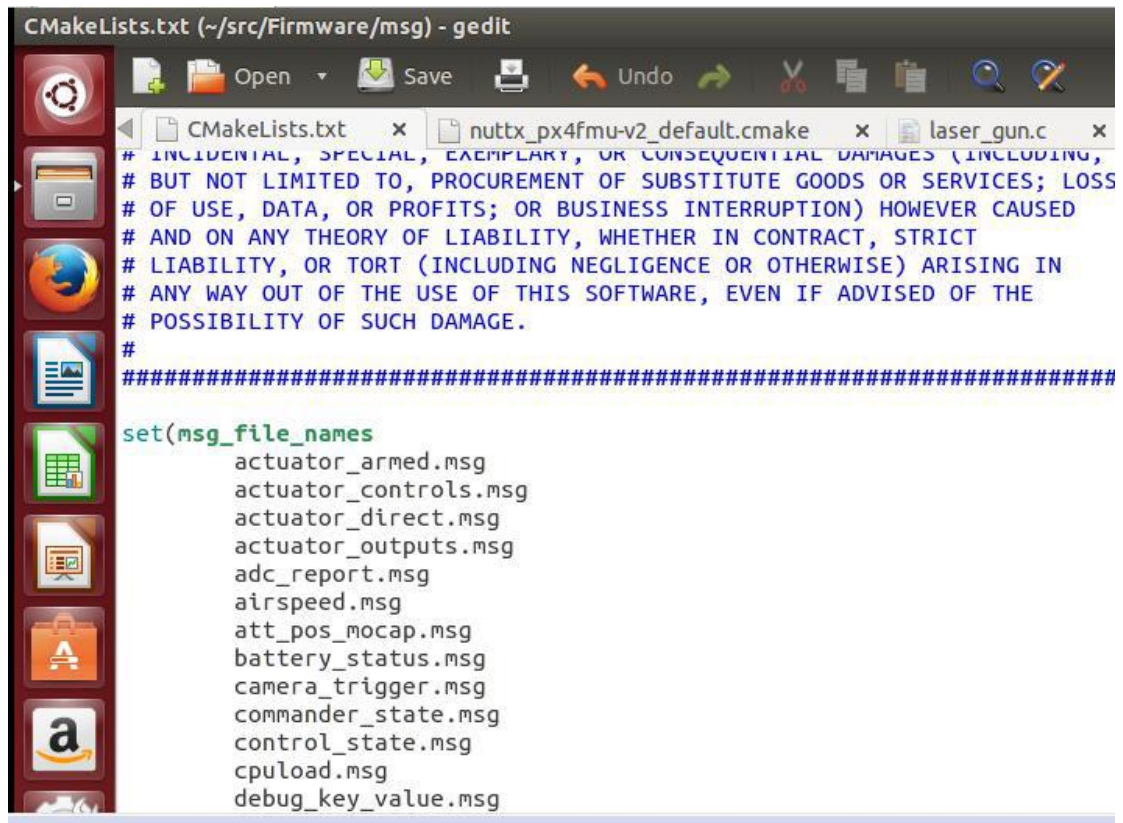
这个 src 目录就是源码，里面有所有的子目录都有 CmakeList.txt 编译脚本文件，我们一般在 modules 里面添加我们自定义的代码，drivers 是传感器驱动目录，lib 是库目录，systemcmds 是系统命令。

```
#####
px4_add_module(
    MODULE modules__mc_pos_control
    MAIN mc_pos_control
    COMPILE_FLAGS
    STACK_MAIN 1200
    SRCS
        mc_pos_control_main.cpp
    DEPENDS
        platforms__common
)
# vim: set noet ft=cmake fenc=utf-8 ff=unix :|
```

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 build 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

```
amov01@ubuntu: ~/src/Firmware/ROMFS/px4fmu_common/init.d
13005_vtol_AAERT_quad      4050_generic_250
13006_vtol_standard_delta  4060_dji_matrice_100
13007_vtol_AAVVT_quad     4900_crazyflie
13008_QuadRanger          50001_axialracing_ax10
13009_vtol_spt_ranger     5001_quad_+
13010_claire              6001_hexa_x
14001_tri_y_yaw+          7001_hexa_+
14002_tri_y_yaw-          8001_octo_x
15001_coax_heli           9001_octo_+
16001_helicopter          rc.axialracing_ax10_apps
20000_snapdragon_rc_pwm   rc.axialracing_ax10_defaults
2100_mpx_easystar         rc.fw_apps
2101_fw_AERT              rc.fw_defaults
2102_3dr_skywalker        rc.interface
2103_skyhunter_1800       rc.io
2104_fw_AETR              rc.mc_apps
2105_maja                 rc.mc_defaults
2106_albatross            rcS
3030_io_camflyer          rc.sensors
3031_phantom              rc.vtol_apps
3032_skywalker_x5         rc.vtol_defaults
3033_wingwing
amov01@ubuntu:~/src/Firmware/ROMFS/px4fmu_common/init.d$ gedit rcS
```

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。