在介绍框架之前，首先要说明一点，px4 的代码十分庞大，但是本章介绍的侧重点仅仅是功能模块代码，这些代码位于源代码跟路径：

/src/modules/

在 px4 源码中对二次开发比较重要的代码模块有：

commander

ekf2

mavlink

mc_att_control

mc_pos_control

navigator

1.commander

commander 模块主要处理一些 px4 的常用命令，如传感器效准、模式切换、加锁解锁、起飞降落、紧急断电等等的命令。

commander 模块的输入 topic 是 vehicle_command。模块会根据不同的命令 id 将命令分解成其他的 topic 并发布出去供其他模块响应。如果我们想使用 commander 模块带来的便利，我们可以在自己的程序代码中直接发布主题：vehicle_command

2.ekf2

ekf2 模块主要处理飞控上的传感器数据融合，如 gps、加速度计、指南针等等。

ekf2 的输入参数都是又 px4 的硬件驱动模块提供的，我们在进行二次开发的过程中不需要关心具体的 topic。

ekf2 的输出 topic 有：

```
vehicle_attitude
vehicle_local_position
vehicle_global_position
```

vehicle_attitude：当前飞控的姿态
vehicle_local_position：当前飞控的本地位置，坐标系为 ned，以飞控加点第一次得到的点为原点
vehicle_global_position：当前飞控的全球位置

如果我们在自己的代码中有需求，可以直接订阅这三个主题，ekf2 保证这些数据的实时更新，我们不需要去关系具体硬件。

3.mavlink

mavlink 模块主要处理 mavlink 通讯，mavlink 的输入总是由飞控的数传串口，输出总是将 mavlink 封包分解后的各个 topic。这也是我们分析飞控的重点模块，是在 px4 中处于最上层的模块之一。

4.mc_att_control

mc_att_control 模块是处理姿态控制的模块，所有的上层模块的飞行决策最终都将化为 topic vehicle_attitude_setpoint 作为 mc_att_control 模块的输入参数。但是我们自己的代码中请不要直

接发布此主题，因为 px4 中对此主题有完整的响应链，如果我们直接发布期望姿态，那么很可能引起混乱，一定要仔细分析，直到分析道最上层的决策 topic。

mc_att_control 模块的输出 topic 有：

actuator_controls_0
当然其他还有另外 7 个控制 topic：
actuator_controls_1
actuator_controls_2
actuator_controls_3
actuator_controls_4
actuator_controls_5
actuator_controls_6
actuator_controls_7
分别代表 8 个混控器 control group，px4 中混控器的作用主要是将期望姿态信息 roll pitch yaw 映射到实际的电机转速或者舵机角度，当然这又是另外一个话题了，这里我们只需要知道 actuator_controls_0 就是

mc_att_control 模块的最终输出，他的输出在通过混控器之后会实际的改变电机转速或者舵机角度。

5.mc_pos_control

mc_pos_control 模块是处理位置控制的模块，位置模块的输出参数永远都是 topic vehicle_attitude_setpoint。可以看出，位置控制模块是处于姿态控制模块的上层，他的输出作为了姿态控制模块的输入，到这里，我们就可以清楚，为什么我们不能直接控制发布姿态主题了，如果我们直接发布了姿态 topic，那么就很有可能与位置控制发布的姿态 topic 冲突进而产生飞行器的不稳定。

位置控制模块的输入参数比较特殊，分别是：

manual_control_setpoint
position_setpoint_triplet

manual_control_setpoint 是遥控器的当前输入值，比如你的油门打到了 100%，那么 topic 中对应的参数可能就是 100
position_setpoint_triplet 是期望位置信息（本地坐标系 ned），主题来自于其他很多地方，如：
mavlink 模块（地面站的控制）
navigator 模块（auto 模式发起）

到这里我们就可以看出，如果我们想控制飞行器飞行到一个地方，那么我们就应该去发布 position_setpoint_triplet 主题，mavlink 作为主题的发起者，它可以发布这个主题，那么我们也可以发布，我们的功能模块应该喝 mavlink 站在同一层上。

在看另外一个主题 manual_control_setpoint 是遥控器的输入，我们可以想象，遥控器已经是最上层的决策了，并且有一点很重要，遥控器是直接控制的飞行器的姿态，而并不是位置信息，所以，在你的程序中如果你想要控制飞行器的姿态，那么应该发布遥控器主题来命令飞行器，而不应该直接发布期望姿态主题。

6. navigator

navigator 模块主要掌管飞行器的各种自动飞行模式，比如：

数传丢失的自动处理

跟随模式

进入禁飞区的自动处理

gps 故障的自动处理

自动降落

留待模式

自动任务

遥控器丢失自动处理

自动返航

自动起飞

所以 navigator 的输入参数有非常多的主题，其中最重要的是传感器主题，用于判断当前是否需要紧急机制，第二重要的是来自自动任务的数据，此数据被 px4 以 romfs 文件系统保存在一个文件中，记录这从地面站发送的航点任务。

但是不管任务有多么复杂，所有的输入参数，最终都会转化为 position_setpoint_triplet 为位置控制模块提供输入参数。

注意：navigator 发布 position_setpoint_triplet 时是处于 px4 的 auto 模式的，此模式是 px4 自己的模式我们无权干涉它的正常执行，我们发布 position_setpoint_triplet 主题时，一定要让 px4 处于 offboard 模式，这种模式才是给用户使用的。