# 交叉项目综合训练 A
# 课程报告

姓名： 杜澍浛

学号： 2017011010

班级： 自 71

# 目录

# 目标

（1） 熟悉 PX4 源码框架结构

（2） 熟悉 PX4 基础模块结构

（3） 熟悉 PX4 高级功能的实现原理和配置要求

# 方法

（1） 熟悉 PX4 官网提供的源码（https://github.com/PX4/Firmware）

（2） 熟悉 PX4 源码开发人员文档

（3） 熟悉 PX4 官网提供的文档资料（https://docs.px4.io/master/en/index.html）

# 结果

## PX4 源码——创建后台程序&姿态控制

PX4 自动驾驶仪软件可分为三大部分：实时操作系统、中间件和飞行控制栈，其中 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 中间件运行于操作系统之上，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；而飞行控制栈又可分为决策导航部分、位置姿态估计部分、位置姿态控制部分和控制器输出部分。

在熟悉 PX4 源码的过程中，我主要负责了创建后台程序和位置姿态控制这两部分。

创建后台程序可以避免标准应用中由于忘在行末输入"&"造成 shell 阻塞的问题，其主要步骤包括：用后台进程管理函数替代原主函数→启动新任务并传递非后台程序的具体指令选项到后台主函数→添加停止/状态指令以及安全保护。在一并提交的"创建后台程序.pptx"文件中，我通过一个具体的例程详细说明了各

步骤的实现方法，可供后续参考。

位置姿态控制分为内外环控制，内环控制角速度，外环控制角度。运行时，先根据目标姿态（target）和当前姿态（current）求出偏差角，然后通过角速度来修正这个偏差角，最终达到目标姿态，主要步骤包括：对变量进行内存分配和初始化→源码订阅→参数初始化→Nuttx 任务使能→获取当前姿态→飞行模式判断→发布控制量进行姿态控制。在一并提交的“姿态解算和姿态控制.pptx”文件中，我详细说明了各步骤源码的含义，可供后续参考。

# PX4 文档——基础模块结构

### Basic Concepts

介绍了一些与无人机相关的基本概念，涉及到了 PX4 Autopilot、QGroundControl、Vehicle/Flight Controller Board、Sensors、Outputs、Battery/Power、Radio Control、GCS Joystick Controller、Safety Switch、Data/Telemetry Radios、Offboard/Companion Computer、SD Cards、Arming and Disarming、Flight Modes、Safety Settings 和 Heading and Directions 这些内容的简介。

### Flight Controllers

可供 PX4 运行的飞行器控制板非常多，这里给出一些例子：

（1） Pixhawk Series

Pixhawk 系列开放式硬件飞行控制器在 NuttX OS 上运行 PX4，针对不同的用例和市场需求有各种不同的版本。其中 Holybro Pixhawk 4、Holybro Pixhawk 4 mini、Drotek Pixhawk 3 Pro、mRo Pixracer、Hex Cube Black、CUAV Pixhack v3、mRo Pixhawk 1 都是受 PX4 维护和测试团队支持的。

（2） Raspberry Pi 2/3 Navio2

可以搭载关联计算机，用于实现计算机视觉相关功能以及完成其他计算密集型任务。

（3） Commercial UAVs

PX4 可用于许多流行的商用无人机产品。

**Vehicles/Frames**

选择无人机时首先要考虑的即需求。如果需要进行精确悬停并且不介意缩短飞行时间，可以选择 Multicopter；如果要求更长的飞行时间和更大的覆盖范围，可以选择 Fixed Wing Airplane；甚至还有混合型飞机，称为 VTOL-垂直起降飞机，它可以像 Multicopter 这样在垂直模式下起飞，然后像 Fixed Wing Airplane 这样在向前飞行中过渡。

**Sensors**

基于 PX4 的系统使用传感器确定无人机状态（稳定和实现自主控制所需）。无人机状态包括：位置/高度，方向，速度，空速，姿态，不同方向的旋转速度，电池电量等。Pixhawk 系列飞行控制器中已经包含了最少的传感器集合，此外附加或部的传感器也可以连接到控制器。下面介绍几种典型传感器：

（1） GPS & Compass

PX4 支持许多全球导航卫星系统（GNSS）接收器和指南针（磁力计）。它还支持实时运动（RTK）GPS 接收器，将 GPS 系统的精度提升到厘米级。

（2） Airspeed

强烈建议将空速传感器用于固定翼和 VTOL 机架，它们非常重要，因为自动驾驶仪没有其他检测失速的方法。

（3） Tachometer

极力推荐将转速表（转速计数器传感器）用于旋翼机翼框架，它们能够保证自动驾驶仪检测到失速或其他旋翼故障。

（4） Distance

距离传感器用于精确着陆，避障和地形跟踪。

（5） Optical Flow

PX4Flow 是一款可跟踪运动的光流智能相机，并具集成了声纳传感器。PX4 将传感器输出与其他位置来源（例如 GPS）的信息混合在一起，以提供更准确的位置锁定。当 GPS 不可用时，此传感器可在室内使用。

**Radio Control Systems**

如果要通过手持发射器手动控制无人机，则需要无线电控制系统，它有一个基于地面的远程控制单元，用户可以使用它来指挥无人机。远程控制单元有用于指示无人机活动的物理控件，可以控制其运动（例如调整速度，方向，油门，偏航角，俯仰角，侧倾角等），还可以启用自动驾驶飞行模式（例如自动起飞，着陆，返回地面，执行任务等）。在启用遥测功能的无线电控制系统上，远程控制单元还可以接收和显示来自无人机的信息（例如电池电量，飞行模式）。

**Flight Modes**

飞行模式定义了自动驾驶仪如何响应远程控制输入以及如何在全自动飞行期间控制无人机的运动，不同的飞行模式会向用户提供不同类型/级别的自动驾驶辅助。

一般而言，飞行模式不是手动就是自动。手动模式下，用户通过 RC 控制杆（或操纵杆）控制无人机的运动；自动模式下则由自动驾驶仪完全控制无人机，不需要远程控制输入。

在手动/自动的大分类下，不同的飞行器（Fixed Wing Airplane/Multicopter）又有很多种不同的模式，其分类标准由手动/自动、位置固定/不固定以及需要高度信息/不需要高度信息这几个指标组合而成，具体可以参见文档，在此后的高级功能介绍中也会对部分模式做详细说明。

**Flight Reporting**

PX4 会详细记录飞行器的状态和传感器数据，用于后续的性能问题分析。用户可以参照文档指示下载日志并分析，还可以与开发团队共享日志以供团队协作。

# PX4 文档——高级功能

这一部分的相关文件包括“Advanced Features.docx”和“PX4 Advanced Features.pptx”。

**RTK GPS**

RTK(Real Time Kinematic)可以将 GNSS 或 GPS 系统的精度提升到毫米级别，保证了 PX4 可以在一些高精度要求的场景下使用。

**Precision Landing**

PX4 支持使用 IR-LOCK 传感器、IR 信标（例如 IR-LOCK MarkOne）以及朝下的距离传感器进行精确着陆，这样精度可以达到 10cm，而 GPS 可能达到几米量级。

（1） 配置要求

硬件方面，按照官方指南安装 IR-LOCK 传感器，确保传感器的 x 轴和 y 轴分别与车辆的 y 轴和 x 轴对齐。此外还需要安装一个距离传感器（如 LidarLite v3），要注意在存在 IR-LOCK 信标的情况下，许多基于红外的距离传感器性能均不佳，可以查看 IR-LOCK Guide 获取兼容性信息。

软件方面，Precision landing 需要 irlock 和 landing_target_estimator 两个模块，可以通过在相关的 configuration 文件中加上下面两行来添加模块：

```
drivers/irlock
modules/landing_target_estimator
```

几个重要的参数：

**LTEST_MODE**：决定信标是固定的还是移动的，如果将 LTEST_MODE 设置为移动则信标测量仅用于在精确着陆控制器中生成位置设定值；如果将 LTEST_MODE 设置为平稳，则信标测量值还将由无人机位置估计器（EKF2 或 LPE）使用。

**LTEST_SCALE_X & LTEST_SCALE_Y**：用于缩放信标测量和估计信标相对于无人机的位置和速度。

标定参数的方法：

将 LTEST_MODE 设置为移动，让无人机在信标上方前后左右飞行并记录 landing_target_pose 和 vehicle_local_position。然后将 anding_target_pose.vx_rel 和 landing_target_pose.vx_rel 分别与 vehicle_local_position.vx 和

vehicle_local_position.vy 相比较。如果估计的信标速度始终小于或大于实际速度则调整比例参数以进行补偿。

（2） 模式选择

可以将 Precision Land Modes 配置为“required”或者“opportunistic”，模式的选择决定了着陆的方式。

Required Mode: 设置为此模式后，如果无人机在着陆开始时看不到信标就将搜索信标。 如果找到信标则执行精确着陆。搜索时先爬升到搜索高度（PLD_SRCH_ALT），如果信标在搜索高度仍然不可见，等到超时（PLD_SRCH_TOUT）之后，无人机将在当前位置启动正常着陆。

Opportunistic Mode: 设置为此模式后，无人机当且仅当信标可见才会执行精确着陆，如果看不见则立即在当前位置执行正常着陆。

（3） 重要模块

Landing Target Estimator: 此模块用于估算信标相对于无人机的位置和速度，每当将新的 irlock_report 合并到估算中时，landing_target_estimator 就会发布估算出的相对位置和速度，如果看不到信标或拒绝信标测量，则不会发布任何内容，估算值发布在 landing_target_pose uORB 消息中。

Enhanced Vehicle Position Estimation: 如果使用参数 LTEST_MODE 将信标指定为固定的，则可以借助信标测量来改善位置/速度估计。

（4） 完整流程

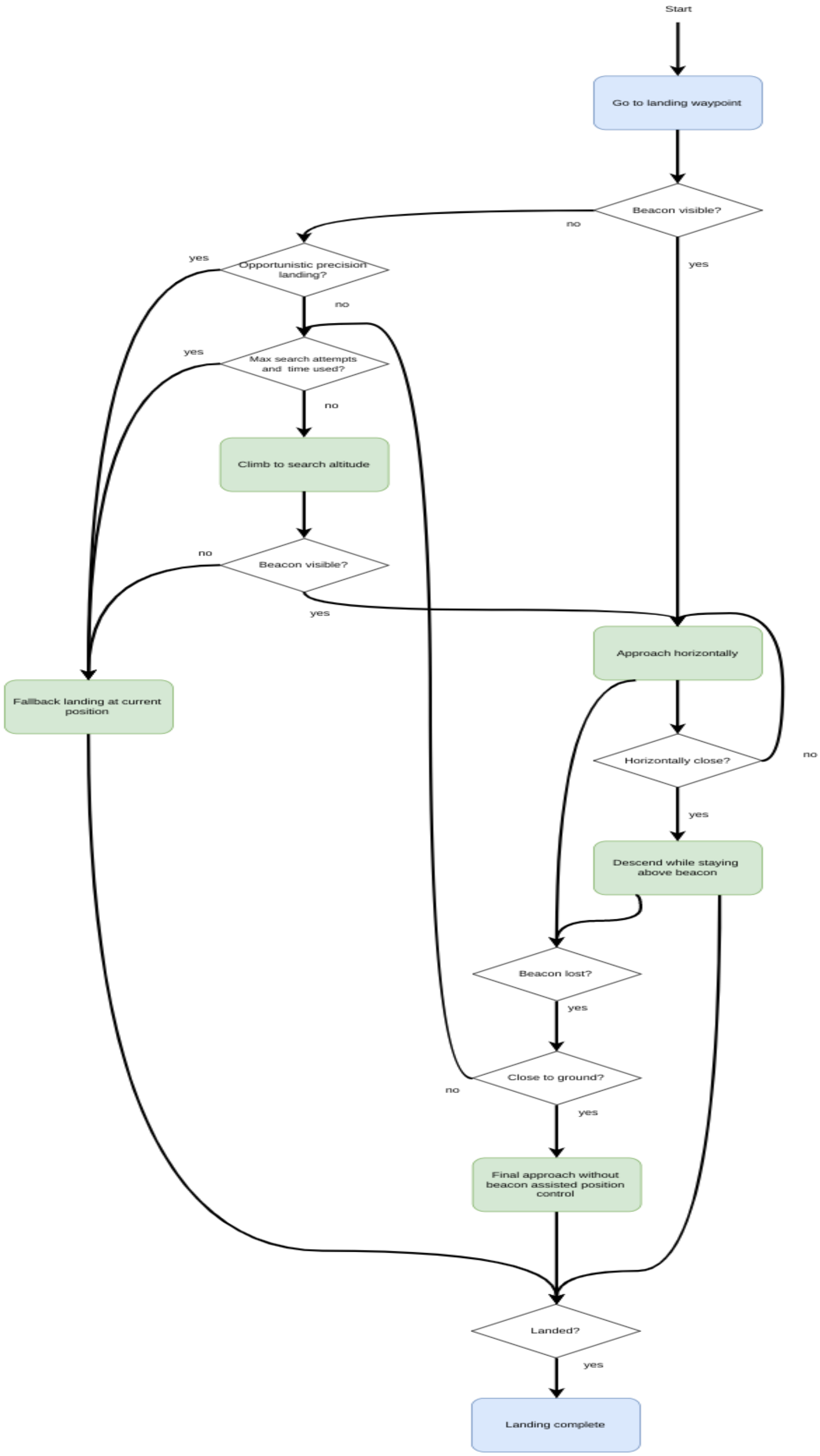
分为以下几个阶段：

1、Horizontal approach：无人机在保持其当前高度的同时水平接近信标。 一旦信标相对于无人机的位置低于阈值就进入下一阶段。如果在此阶段丢失信标，无人机在 Required Mode 下将启动搜索过程，而在 Opportunistic Mode 下进行常规着陆。

2、Descent over beacon：无人机下降，同时保持在信标上方居中。如果在此阶段丢失信标，无人机在 Required Mode 下将启动搜索过程，而在 Opportunistic Mode 下进行常规着陆。

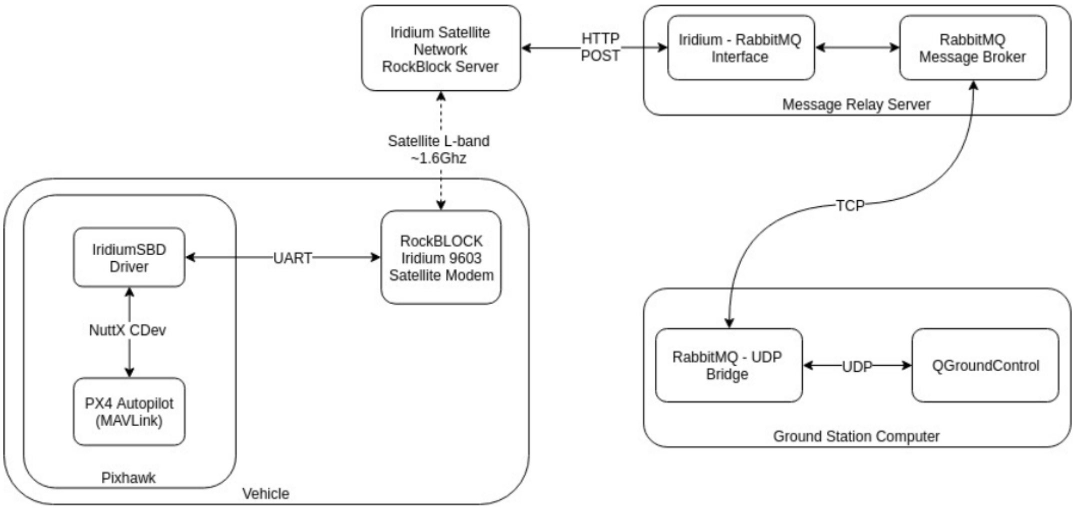
3、Final approach：当无人靠近地面（距离低于阈值）时，它会下降，同时保持在信标上方。 如果在此阶段丢失信标，不管在哪种模式下下降都将继

续进行。


Start
Go to landing waypoint
Beacon visible?
no
yes
Opportunistic precision landing?
yes
no
Max search attempts and time used?
yes
no
Climb to search altitude
Beacon visible?
no
yes
Approach horizontally
Fallback landing at current position
Horizontally close?
no
yes
Descend while staying above beacon
Beacon lost?
yes
Close to ground?
no
yes
Final approach without beacon assisted position control
Landed?
yes
Landing complete

**Iridium/RockBlock Satellite Communication System**

卫星通信系统可用于在地面站和无人机之间提供远程高延迟链路。其基本框架如下（文档内容基本上是配置教程，在此不赘述）：




**Air Traffic Avoidance: ADS-B/FLARM/UTM**

PX4 可以使用 ADS-B 或 FLARM 应答器一定程度地避免空中交通碰撞。 如果检测到潜在碰撞，PX4 可以发出警告、立即着陆或返回（取决于 NAV_TRAFF_AVOID 的值）。文档中的硬件配置部分在此不赘述。此外，PX4 还可以使用 MAVLink UTM GLOBAL POSITION 来实现相同的功能。

关于软件配置，Flarm / PingRX 的配置方式与其他 MAVLink 外围设备相同, 唯一特别的是必须将端口波特率设置为 57600，并设置一个低带宽配置文件（MAV_X_MODE）。

使用此功能时，如果在飞行过程中收到了有效的应答器报告，PX4 首先根据应答器位置和航向信息来估计两架无人机在相遇前是否处在相同高度。如果高度相近，PX4 会估计无人机到下一个路标的路径与其他无人机预测路径间的最短距离。 如果此距离小于阈值则启动回避操作，无人机会发出警告、着陆或返回。用户可以对人工和无人操作分别配置检测距离。相关参数及其含义如下表所示：

| Parameter | Description |
|---|---|
| NAV_TRAFF_AVOID | =0：禁用；=1：仅警告；=2：返回模式；=3：着陆模式 |
| NAV_TRAFF_A_RADM | 设置人工操作的回避距离 |
| NAV_TRAFF_A_RADU | 设置无人操作的回避距离 |

**Obstacle Avoidance**

这一部分介绍如何使无人机在遵循预定路径的同时绕过障碍物。实现该功能需要在无人机飞行的同时运行另一台关联计算机上的视觉软件。该软件可以为给定的轨迹规划实际路线，在障碍物周围进行地图绘制和导航，帮助找到最佳路径。避障适用于自动模式，避障模式下的最大速度约为3m/s（计算避障路径的成本）。目前 Mission Mode 和 Offboard Mode 都支持避障。

使用 Offboard Mode 时，在关联计算机上运行 ROS 节点，输出初始版本的路径，再将此输出传入避障模块（另一个 ROS 节点），避障软件将改进版的路径规划以 SET_POSITION_TARGET_LOCAL_NED 消息流的形式发送到飞行堆栈中，无人机由此获取加入避障内容后的路线。其中 PX4 端仅需要将自身置于 Offboard Mode 下即可，关联计算机端的硬件设置参见文档。

使用 Mission Mode 时，某些任务行较常规情况为会发生变化：

1、常规情况下，无人机要“到达”一个预定位置，不仅要求位置吻合（处于预设的半径范围内），还需要处于一定的航向。启用避障之后只要求位置吻合。

2、启用避障后，无人机一到达前一个预定位置（进入设定的半径范围内），PX4 就立刻发射下一个预定位置。

3、如果某个预定位置处于障碍物范围内，无人机可能无法到达此位置，此时任务被卡住。

4、在 QGroundControl / PX4 中设置的原始任务速度将被忽略，无人机速度完全由避障软件决定。

如果 PX4 在 1.5 秒内没有收到预定位置的更新信息，它会转入 Hold Mode。

**Safe Landing**

此功能可确保车辆仅在平坦地形上着陆，要求在无人机飞行的同时运行另一台关联计算机上的视觉软件，可在 Land Mode 和 Mission Mode 下使用。如果收到着陆的指令，无人机首先下降到可以测量地面情况的高度（依靠关联计算机给出的 loiter_height 参数）。 如果着陆区域不够平坦，无人机会以正方形螺旋状向外移动，并定期停下来重新检查地形，以寻找合适的着陆点。需要注意的是，在道路上着陆是允许的，如果检测到汽车，汽车驶过之后会被“遗忘”；如果使用雷达或超声波传感器则降落在水上是允许的，但如果使用立体摄像机或激光雷达则不能降落在水上，这是因为立体摄像机这类设备而言水面会被判定为不够平坦。

**Collision Prevention**

防撞功能用于控制无人机在撞到障碍物之前自动减速和停止，在 Position Mode 下启用，可以使用来自外部关联计算机、MAVLink 上的外部测距仪、连接到飞行控制器的测距仪以及以上任意组合的传感器的数据（关于障碍物的数据）。但是如果传感器范围不够大，此功能可能会限制最高速度，因为它会禁止无人机在没有可用传感器数据的方向运动。

启用后，无人机在靠近障碍物时会减速，并在达到最小允许间隔时停止运动。为了远离障碍物（或者平行于障碍物飞行），用户必须保证无人机朝不会朝靠近障碍物的设定点移动，下面介绍相关算法和软件设置，关联计算机和传感器设置参见文档。

其算法原理总结如下：

来自所有传感器的数据被整合到环绕无人机的 36 个分区中，每个分区中除非无可用数据，否则一定包含着对应的传感器数据以及上一次观察时间的记录。当无人机接到朝某个特定方向运动的指令后，该方向对应的所有分区都要被检测，如果检测结果是目前的移动指令会使得无人机更靠近某个障碍物，其速度就会受到限制。相关参数如下表所示：

| Parameter | Description |
|---|---|
| CP_DIST | 设置最小允许距离（无人机和障碍物间的最小距离） |
| CP_DELAY | 设置传感器传输延迟和对速度设定值的跟踪延迟 |
| CP_GUIDE_ANG | 设置无人机如果在该方向上发现较少障碍物时可能偏离的角度 |
| CP_GO_NO_DATA | =1：允许无人机在没有传感器覆盖的方向飞行（默认为 0，不允许） |

下面详细介绍一下不太好理解的几个参数：

1、**CP_DELAY**：功能执行中的延迟主要有两个来源：传感器传输延迟和对速度设定值的跟踪延迟。直接连接到飞行控制器的距离传感器的延迟可以假定为 0，但对于基于外部视觉的系统，传感器延迟可能高达 0.2s。而要测量对速度设定值的跟踪延迟，可以在 Position Mode 下先令无人机以全速飞行然后再使其停止，然后从记录中测量实际速度和速度设定值之间的延迟，它通常在 0.1 到 0.5s 之间，具体取决于无人机尺寸和调整情况。

2、**CP_GUIDE_ANG**：如果与当前指令中的飞行方向邻近的某个方向更好，算法可以轻微调整航向，这有助于避免无人机卡在障碍物上的情况，CP_GUIDE_ANG 设定的就是能够调整的最大角度。但是也不能将此值设得过大，否则无人机可能会大幅度偏离指示方向，实际操作中常常将其设为 30°。

3、**CP_GO_NO_DATA**：如果无人机超过 0.5s 没有接收到任何传感器数据，它将发出警告：未收到数据，不允许移动。 这会速度设定值强制降到零。 5 秒钟未收到任何数据后，无人机将切换到 Hold Mode。 如果希望无人机能够再次移动，则需要将 CP_DIST 设置为负值或切换到 Position Mode 以外的其他模式。但是如果将 CP_GO_NO_DATA 设为 1，当某个传感器失去连接（对应方向失去传感器覆盖后），无人机尽管收不到数据，仍然可以向该方向运动。

**Path Planning Interface**

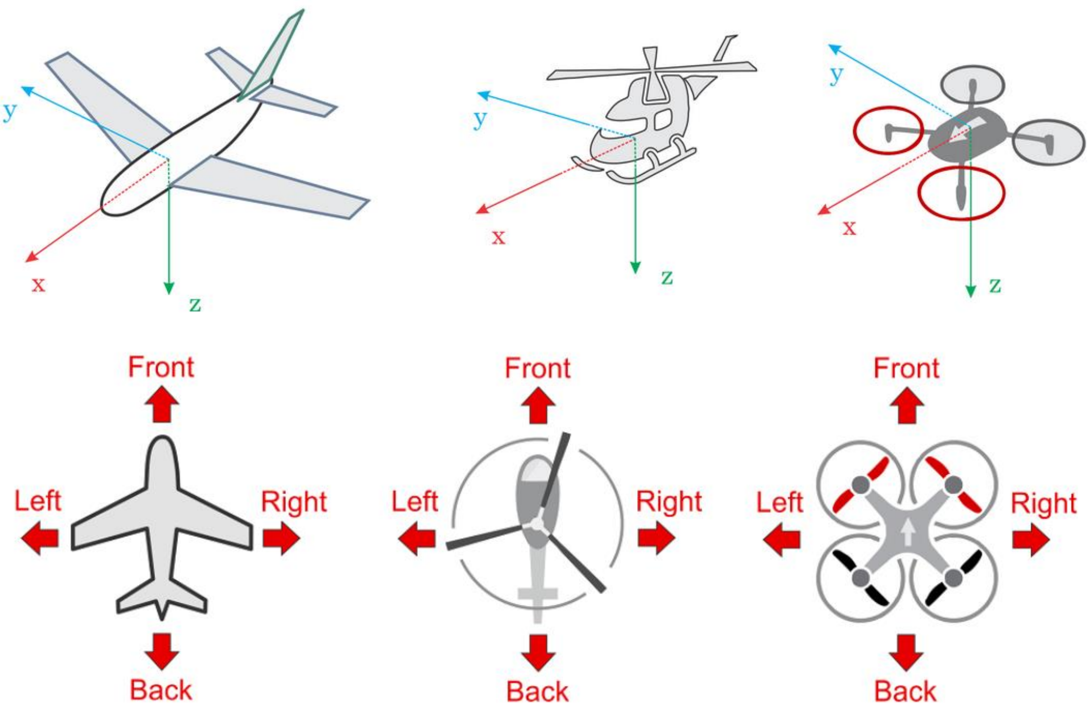
PX4 使用许多 MAVLink 接口来集成关联计算机上的路径规划服务（前面介绍过的避障、安全着陆等功能中关联计算机提供路径时都要依靠这些接口），这一部分的详细信息参见文档中的说明。

# 英文总结

（1） Basic Concepts

A drone is an unmanned "robotic" vehicle that can be remotely or autonomously controlled. The "brain" of the drone is called an autopilot. It consists of *flight stack* software running on *vehicle controller* ("flight controller") hardware.

All the vehicles, boats and aircraft have a heading direction or an orientation based on their forward motion. It is important to know the vehicle heading direction in order to align the autopilot with the vehicle vector of movement, multicopters have a heading even when they are symmetrical from all sides. Usually manufacturers use a colored props or colored arms to indicate the heading.




PX4 is a core part of a broader drone platform that includes the QGroundControl ground station, Pixhawk hardware and MAVSDK for integration with companion computers, cameras and other hardware using the MAVLink protocol, which is supported by the Dronecode Project. It can run on many flight controller boards. For instance, Pixhawk

Series open-hardware flight controllers run PX4 on NuttX OS.

The Dronecode ground control station is called QGroundControl. Users can use *QGroundControl* to load (flash) PX4 onto the vehicle control hardware, setup the vehicle, change different parameters, get real-time flight information and create and execute fully autonomous missions. *QGroundControl* runs on Windows, Android, MacOS or Linux.

PX4 was initially designed to run on Pixhawk Series controllers, but can now run on Linux computers and other hardware. Users should select a board that suits the physical constraints of the vehicle, the activities they wish to perform, and of course the cost.

When planning using PX4 for flying aircraft, the most important question users should consider is, what will be the application, is it for fun or for work, what is the planned flight times and coverage. Generally, if precision hovering is needed and shorter flight times can be acceptable then users should focus on **multicopters**. On the other hand, if longer flights and larger areas of coverage are needed then fixed wing aircrafts will be the right choice. There is a mixed type of aircraft called **VTOL** - Vertical Takeoff and Landing aircraft. It can take off in vertical mode like multicopters and then transition in forward flight like fixed wing aircrafts.

PX4-based systems use sensors to determine vehicle states including position/altitude, heading, speed, airspeed, orientation (attitude), rates of rotation in different directions, battery level, etc. The flight system *minimally requires* a gyroscope, an accelerometer, a magnetometer (compass) and a barometer. Moreover, GPS or other positioning system is needed to enable all automatic modes and some assisted modes, fixed wing and VTOL-vehicles should additionally include an airspeed sensor. Pixhawk Series flight controllers have already incorporated the minimal set of sensors, additional/external sensors can also be attached to the controller.

If users want to *manually* control the vehicle from a handheld transmitter then a radio control (RC) system will be required. An *RC system* has a ground-based *remote control unit* which can be used to command the vehicle. The unit contains a radio module which is bound to, and communicates with, a (compatible) radio module on the vehicle. The vehicle-based unit is connected to the flight controller. The flight controller determines how to interpret the commands based on the current autopilot flight mode and vehicle state, and drives the vehicle motors and actuators appropriately. An important quality of an RC system is how many "channels" it supports. The number of channels defines how many different physical controls on the unit can be used to send commands to the vehicle.

Flight modes define how the autopilot responds to remote control input, and how it manages vehicle movement during fully autonomous flight. The modes provide different types/levels of autopilot assistance to the user. Users can transition between flight modes using switches on the remote control or with a ground control station. However, not all flight modes are available on all vehicle types and some modes behave differently on different vehicle types, PX4 will not allow transitions to those modes until the right conditions are met.

Flight Modes are, generally speaking, either *manual* or *autonomous*. Manual modes are those where the user has control over vehicle movement via the RC control sticks (or joystick), while *autonomous* modes are fully controlled by the autopilot, and *require* no pilot/remote control input.

Manual modes may further be divided into "easy" and "acrobatic" modes. In the easy modes, roll and pitch sticks set the vehicle angle, resulting in left-right and forward-back movement *in the horizontal plane* (respectively). Not only does this make movement predictable, but because angles are controlled, the vehicle is impossible to flip. In acrobatic modes RC sticks control the rate of angular rotation (around the respective axis), vehicles can flip, and while more maneuverable, are harder to fly.

In autonomous modes RC stick movement will by default change the vehicle to Position mode when flying as a multicopter (unless handling a critical battery failsafe). Stick movement is ignored for fixed-wing flight.

Last of all, PX4 logs detailed aircraft state and sensor data, which can be used to analyze performance issues. It uses SD memory cards for storing flight logs, which are also required in order to use UAVCAN peripherals and fly missions. By default, if no SD card is present PX4 will play the format failed (2-beep) tune twice during boot (and none of the above features will be available).

（2） Advanced Features

Real Time Kinematic (RTK) increases the accuracy of GNSS/GPS systems to centimeter-level, it allows PX4 to be used in applications like precision surveying, where pinpoint accuracy is essential.PX4 supports a number of RTK GPS devices, it requires QGroundControl running on a laptop/PC and a vehicle with a WiFi or Telemetry radio link to the laptop.

PX4 supports precision landing for multicopters (from PX4 v1.7.4) using the IR-LOCK Sensor, an IR beacon (e.g. IR-LOCK MarkOne) and a downward facing range sensor. This enables landing with a precision of roughly 10 cm (GPS precision, by contrast, may be as large as several meters).A precision landing can be initiated by entering the Precision Land flight mode, or as part of a mission.

A precision landing can be configured to either be "required" or "opportunistic". The choice of mode affects how a precision landing is performed. In Required Mode the vehicle will search for a beacon if none is visible when landing is initiated. The vehicle will perform a precision landing if a beacon is located. By contrast, in Opportunistic Mode the vehicle will use precision landing if (and only if) the beacon is visible when landing is initiated. If it is not visible the vehicle immediately performs a normal landing at the current position. The whole procedure consists of three phases:

1. **Horizontal approach:** The vehicle approaches the beacon horizontally while keeping its current altitude. Once the position of the beacon relative to the vehicle is below a threshold (PLD_HACC_RAD), the next phase is entered. If the beacon is lost during this phase (not visible for longer than PLD_BTOUT), a search procedure is initiated (during a required precision landing) or the vehicle does a normal landing (during an opportunistic precision landing).
2. **Descent over beacon:** The vehicle descends, while remaining centered over the beacon. If the beacon is lost during this phase (not visible for longer than PLD_BTOUT), a search procedure is initiated (during a required precision landing) or the vehicle does a normal landing (during an opportunistic precision landing).
3. **Final approach:** When the vehicle is close to the ground (closer than PLD_FAPPR_ALT), it descends while remaining centered over the beacon. If the beacon is lost during this phase, the descent is continued independent of the kind of precision landing.

A satellite communication system can be used to provide long range high latency link between a ground station and a vehicle. The documentation has thoroughly described how to set up a system that uses RockBlock as the service provider for the Iridium SBD Satellite Communication System. Given good signal quality, users can expect a latency between 10 to 15 seconds.

PX4 can use ADS-B or FLARM transponders to support simple air traffic avoidance in missions. If a potential collision is detected, PX4 can warn, immediately land or return (depending on the value of NAV_TRAFF_AVOID). As for the implementation, PX4 listens for valid transponder reports during missions, if a valid transponder report is received, it first uses the transponder position and heading information to estimate whether the vehicles will share a similar altitude before they pass each other. If they may then PX4 will estimate the closest distance between the path to the next waypoint and the other vehicles’ predicted path. If the crossing point is less than the configured distance for altitude and path, the Traffic Avoidance Failsafe action will be started, the vehicle will either warn, land or return. The detection distance can be configured separately for manned and unmanned aviation. PX4 will also forward the transponder data to a GCS if it has been configured for the MAVLink instance.

Besides ADS-B and FLARM, PX4 can use MAVLink UTM_GLOBAL_POSITION messages to support simple air traffic avoidance in missions. If a potential collision is detected, PX4 can warn, immediately land, or return (depending on the value of NAV_TRAFF_AVOID). When it comes to implementation, PX4 listens for UTM_GLOBAL_POSITION MAVLink messages during missions. When a valid message is received, its validity flags, position and heading are mapped into the same transponder_report UORB topic used for ADS-B traffic avoidance.

Obstacle Avoidance enables a vehicle to navigate around obstacles when following a preplanned path. The feature requires a companion computer that is running computer vision software which provides a route for a given desired trajectory, mapping and navigating around obstacles to achieve the best path. Obstacle avoidance is intended for automatic modes and is currently supported for multicopter vehicles in Missions and Offboard mode.

The Safe Landing feature ensures that vehicles only land on flat terrain. It can be enabled in both Land mode and Mission mode on multicopter vehicles that have a companion computer running the appropriate vision software. It can also be used for VTOL vehicles in MC mode. If commanded to land, the vehicle first descends to a height where it can measure the surface, if the landing area is not sufficiently flat, the vehicle moves outwards in a square-spiral pattern, periodically stopping to re-check the terrain for a landing spot that isn't too rough.

Collision Prevention may be used to automatically slow and stop a vehicle before it crashes into an obstacle. It can be enabled for multicopter vehicles in Position mode, using sensor data from an offboard companion computer, offboard rangefinders over MAVLink, a rangefinder attached to the flight controller, or any combination of the above. However, it may restrict vehicle maximum speed if the sensor range isn't large enough. It also prevents motion in directions where no sensor data is available.

PX4 uses a number of MAVLink interfaces for integrating path planning services from a companion computer (including obstacle avoidance in missions, safe landing, and future services). Path planning is enabled on PX4 in automatic modes (landing, takeoff, hold, mission, return) if COM_OBS_AVOID=1. In these modes planning software is expected to supply setpoints to PX4, if the software cannot support a particular flight mode it must mirror back setpoints from the vehicle.

Motion Capture (MoCap) is a computer vision technique for estimating the 3D pose (position and orientation) of a vehicle using a positioning mechanism that is external to the vehicle. It is commonly used to navigate a vehicle in situations where GPS is absent (e.g. indoors) and provide position relative to a local co-ordinate system. In most cases, MoCap systems detect motion using infrared cameras, but other types of cameras, Lidar or Ultra Wideband (UWB) may also be used.

Visual Inertial Odometry (VIO) is a computer vision technique used for estimating the 3D pose (local position and orientation) and velocity of a moving vehicle relative to a local starting position. It is commonly used to navigate a vehicle in situations where GPS is absent or unreliable (e.g. indoors, or when flying under a bridge). It uses Visual Odometry to estimate vehicle pose from camera images, combined with inertial measurements from the vehicle IMU (to correct ferrors associated with rapid vehicle movement resulting in poor image capture).

# 感想和展望

## 感想

说到对本学期课程的感想，我首先想到的词是遗憾。我相信有很多同学和我一样，在选课的时候非常期待能接触无人机这个新鲜的项目，进而参与到国际赛事中，但受疫情影响，这些想法都没能实现。另外，我是飞控组的成员，我们组没有办法实际操作，组长安排的工作内容仅限于熟悉源码和工作文档，这也是遗憾的一大来源。

但除了遗憾，这门课程其实也带给我相当大的收获。我学到了很多无人机控制的理论知识，也在实验中体验了无人机的工作过程，这些都为我将来进一步深入这一领域打下了良好的基础。

## 展望

正如感想部分所述，我还没有机会接触到无人机复杂功能的实现，如果可能的话，我希望在以后的课题或项目中再次选择与无人机有关的内容，弥补这一遗憾。

此外，对于课程内容，个人认为实验环节繁复的配置步骤太多，观察到的实验现象则比较简单。虽然实际操作中必须进行前期的配置工作，但由于大家只进行过一次实验，相关内容很容易被遗忘，最后只留下“花了很多时间看操作手册”的印象。如果以后的实验内容可以更侧重于各种复杂的实验现象，或许可以进一步提升同学们的兴趣，增进大家对相关原理的理解。

最后感谢老师和助教一年来的辛苦付出！