

Alternativas open-source para vehículos aéreos no tripulados

[Alternativas Dronecode](#)

[Hardware](#)

[Software](#)

[Estimación](#)

[Control](#)

[Estación base](#)

[Capacidades de cámara y video](#)

[Capacidades de ADS-B](#)

[Referencias](#)

[Contacto](#)

Aclaración:

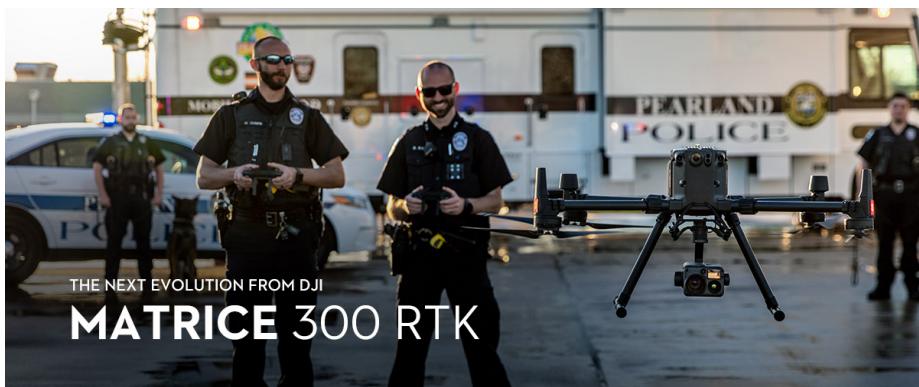
El siguiente es un artículo escrito con fines de divulgación y hecho con el mayor cuidado. Sin embargo no puedo garantizar que todo lo expuesto sea 100% correcto. Si encuentra algun error por favor háganmelo saber (ver sección de Contacto al final del documento).

Historial de actualizaciones

- 06-07-2021
- 07-07-2021

Alternativas Dronecode

Las compañías de Drones dominantes como DJI, Parrot, SenseFly, Freefly, etc, ofrecen al mercado soluciones cerradas de hardware y software. Sus vehículos son de muy buena calidad y su fama es merecida.





Pero si nuestra intención es aprender, experimentar, modificar o innovar con esta tecnología, entonces existen alternativas mejores que nos permiten hacer todo esto y más. Y lo que es más importante, manteniendo un nivel de fiabilidad y prestaciones alto. Es justamente este segmento el que será aquí detallado.

Existe un conjunto de proyectos open-source tanto de hardware como de software apoyados oficialmente por la fundación Dronecode (<https://www.dronecode.org/projects/>). Esta es a su vez parte de la fundación Linux (<https://linuxfoundation.org/>), la cual se dedica a fomentar el desarrollo y mantenimiento de proyectos de software libre.

The Linux Foundation provides a neutral, trusted hub for developers to code, manage, and scale open technology projects

The Dronecode Foundation fosters communities and innovation through **open-standards** using **open-source**. Dronecode is a **vendor-neutral foundation** for open source drone projects. We are a US-based non-profit under the Linux Foundation and provide open source **governance, infrastructure, and services** to software & hardware projects. We work with developers, end-users, and adopting vendors from around the world.

Hardware

Pixhawk (<https://pixhawk.org/>) es uno de esos proyectos apoyados por Dronecode. Es una estándar que define el tipo de sensores y su interconexión con computadores de vuelo, baterías, cámaras, cargas útiles, etc. Es el plano en que diversos fabricantes se basan para comercializar sus productos (<https://pixhawk.org/products/>).

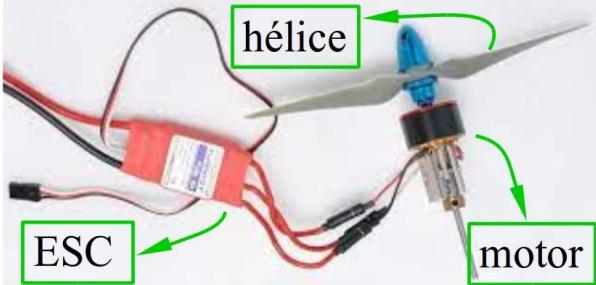
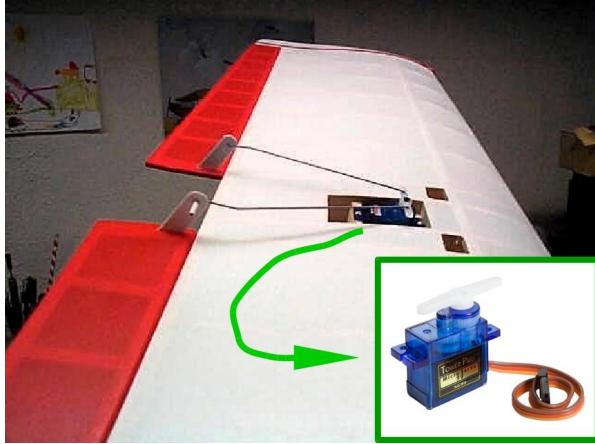
Uno de esos fabricantes es HolyBro que comercializa el producto Pixhawk 4 (<http://www.holybro.com/product/pixhawk-4/>). Otra compañía es Auterion con el producto Skynode (<https://auterion.com/drone-manufacturers/skynode/>). Una tercera opción es HEX/Proficnc y el producto The Cube Orange Standard Set (<https://www.cubepilot.org/#/cube>).

		
Pixhawk 4	Cube Orange	Skynode

Existe un excelente resumen de plataformas de hardware similares en la página web de PX4 (https://docs.px4.io/master/en/flight_controller/). De Px4 hablaremos luego, por ahora nos basta saber que existen más alternativas.

Además de ya mencionado computador de vuelo es necesario contar con los siguientes dispositivos

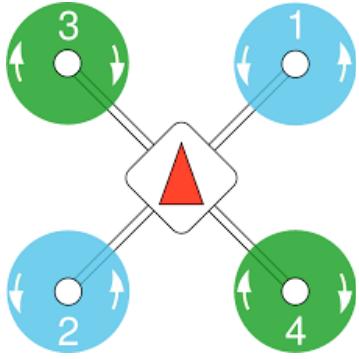
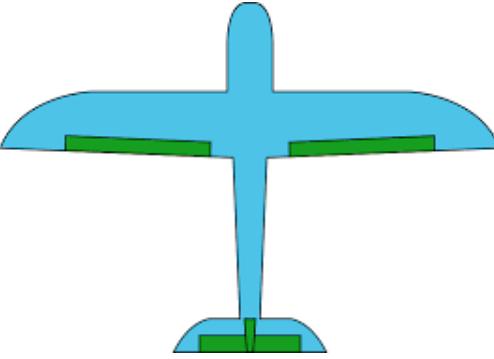
	
Control remoto (en manos del piloto) y radio receptor (en el vehículo) compatibles.	Radio de telemetría (en el vehículo) y otra en tierra (conectada a la estación terrena)

 <p>Actuador para multirotor: controlador de velocidad (i.e. ESC), motor y hélices</p>	 <p>Actuador para avión: Servomotor y superficie de mando (en rojo)</p>
---	---

Software

Px4 (<https://docs.px4.io/master/>) es un software de vuelo que permite estimar el estado del vehículo y ejecutar algoritmos de control para estabilizar el vehículo aéreo, bucles de control PID u otros similares (e.g. model Following control, adaptive control, dynamic inversion y un largo etc). También forma parte de los proyectos apoyados por la fundación Dronecode.

Es frecuentemente usado para volar tanto multirrotores, aviones y diseños experimentales

 <ul style="list-style-type: none"> • u_1 motor 1 • u_2 motor 2 • u_3 motor 3 • u_4 motor 4 	 <ul style="list-style-type: none"> • u_1 aileron • u_2 elevator • u_3 throttle • u_4 rudder
<p>Fuente: https://docs.px4.io/master/en/airframes/airframe_reference.html#quadrotor-x</p>	<p>Fuente: https://docs.px4.io/master/en/airframes/airframe_reference.html#standard-plane</p>

Estimación

La primera parte de su utilidad está basada en el procesamiento de sensores inerciales (acelerómetros, giroscopios, magnetómetro y barómetro) más sensores de posición y velocidad como los GNSS (e.g. GPS, GLONASS, BDS y Galileo). Estos sensores son combinados junto a un modelo dinámico del vehículo para obtener una estimación de las variables

$$x = [x, y, z, u, v, w, \varphi, \theta, \psi, P, Q, R]^T$$

donde x es el vector de estado, $[x, y, z]^T$ la posición, $[u, v, w]^T$ la velocidad, $[\varphi, \theta, \psi]^T$ la posición angular (i.e. [ángulos de Euler](#)) y finalmente $[P, Q, R]^T$ la velocidad angular.

Control

Luego de la estimación de las variables de estado del vehículo, se hace necesario el manipularlas según escojamos a través del uso de actuadores. Los actuadores son todos aquellos dispositivos que ejercen una fuerza y/o un torque sobre el vehículo. Rotores, propulsores, alerones, elevadores, y un largo etcétera.

Existen generalmente dos problemas típicos en la teoría de control. El primero es el problema de regulación, dónde queremos mantener al vehículo en un estado (e.g. vuelo estático a 5 m sobre el suelo, o vuelo a 5 m/s en dirección norte) determinado a pesar de las perturbaciones (e.g. viento, inestabilidades del motor, ruido en los sensores). El segundo es el problema de seguimiento, en donde queremos mantener el estado del vehículo (e.g. posición y/o velocidad) lo más cercano posible a una señal de referencia (e.g. comandos del piloto, ruta predefinida). En cualquier caso, el sistema dinámico del vehículo es expresado como

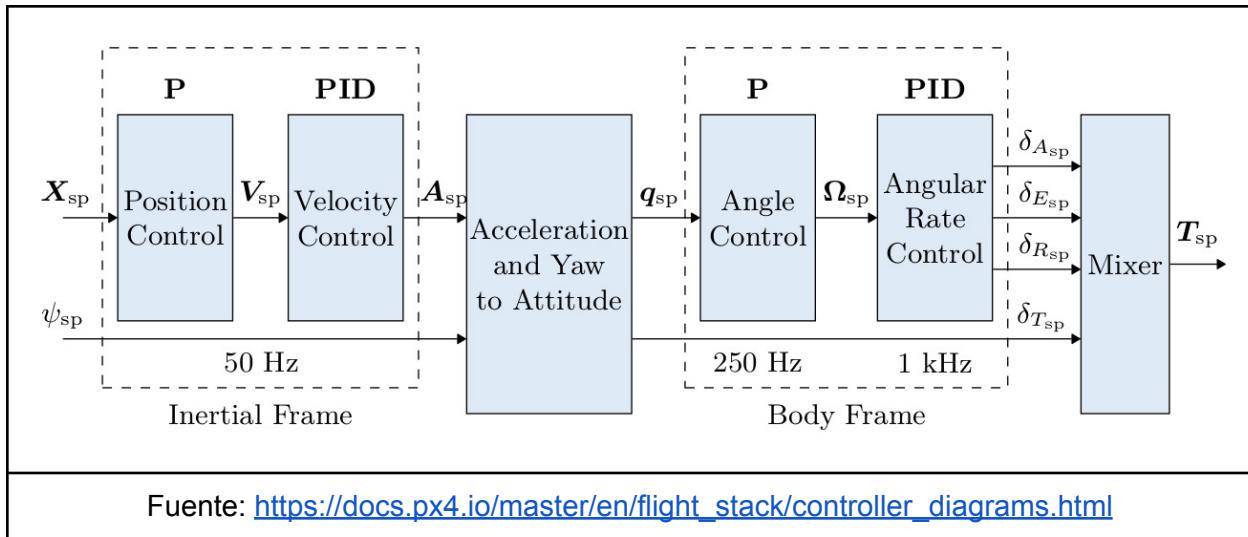
$$\dot{x} = f(x, u)$$

donde u es el vector de variables de control (e.g. velocidad del motor, ángulo de deflexión del alerón).

Tanto Px4 como la mayoría de los software populares, utilizan las herramientas clásicas de control lineal y bucles PID para resolver el problema de seguimiento. En este caso se linealiza la dinámica del vehículo en torno a vuelo en estático para el caso de un multirrotores, y vuelo a velocidad constante para un avión. Así se obtiene un sistema

$$\dot{x} = Ax + Bu$$

donde tanto A como B son matrices con coeficientes constantes, pero dependientes del punto de linealización. La derivación matemática queda para otro ocasión, pero para el caso específico de un multirrotores, el diseño es



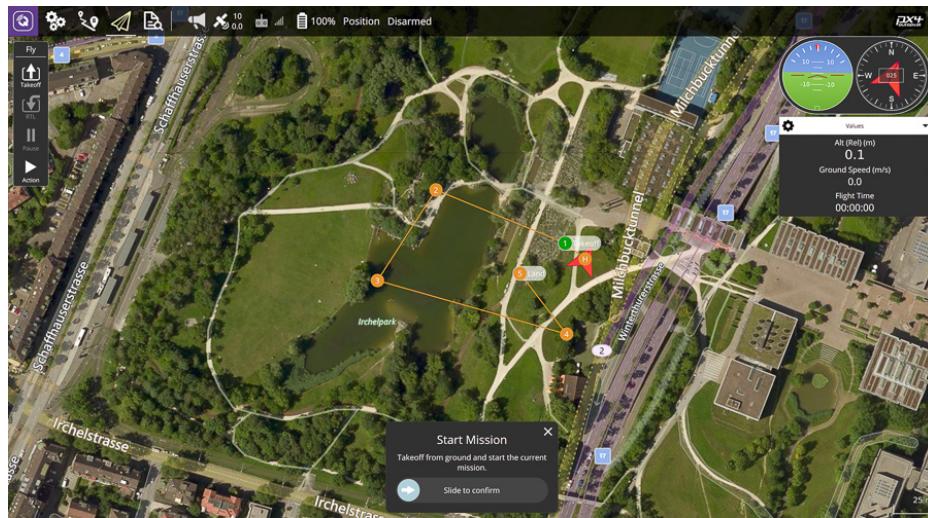
La entrada al diagrama es el conjunto de comandos del piloto, la salida corresponde a los comandos a los motores del vehículo y/o servo motores.

La velocidad a la que se ejecuta este bucle es de vital importancia, tanto en lo teórico como en lo práctico. En lo teórico existe una frecuencia bajo la cual el sistema de control no podrá estabilizar el vehículo. El valor específico depende de la constante de tiempo dada por la dinámica del vehículo y de los actuadores. En la práctica los multirrotores son estables a partir de una frecuencia de control de 50 Hz. Esto es la cantidad de veces por segundo que se envían comandos al motor. Los controladores de motores más modernos permiten llegar a los 400 Hz y hasta los 1000 Hz, permitiendo vuelos más suaves y precisos.

Estación base

QGroundControl (<http://qgroundcontrol.com/>) es también uno de esos proyectos apoyados por Dronecode. Es un software que provee las funcionalidades necesarias de una estación terrena basada en el protocolo de comunicaciones MAVLink (<https://mavlink.io/en/>). Es decir, configuración en tierra del vehículo, telemetría en vuelo del vehículo y gestión de misiones autónomas.

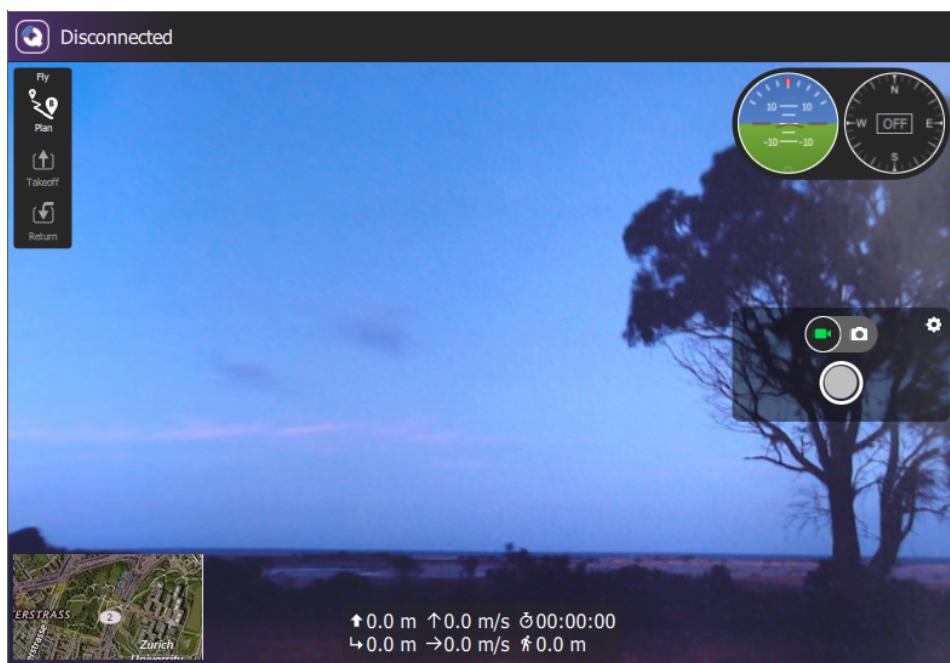
En concreto con este programa se puede saber planificar las zonas en que un vehículo volará, las zonas prohibidas, etc. Y una vez en el aire, conocer en todo en línea la posición, velocidad, carga de baterías y otros valores de vuelo importantes.



Fuente: <https://docs.qgroundcontrol.com/master/en/index.html>

Capacidades de cámara y video

Evidentemente un vehículo aéreo que no pueda transportar cámaras, sensores y carga es de baja utilidad para los usuarios de esta plataforma. Es por ello que cualquier alternativa a las soluciones de las compañías comerciales dominantes debe contemplar esta utilidad. En conjunto, PX4 y QGroundControl permiten hacer streaming de video. Personalmente no he probado esta opción, pero es algo que planeo hacer pronto.



Fuente: https://docs.px4.io/master/en/companion_computer/video_streaming.html

Capacidades de ADS-B

ADS-B (i.e [Automatic Dependent Surveillance-Broadcast](#)) es un protocolo de comunicación usado para alertar a los receptores circundantes acerca de la posición de un vehículo aéreo. Es un sistema frecuentemente usado en aviación comercial y que ha sido propuesto como identificación electrónica para vehículos aéreos no tripulados. A pesar de que varios fabricantes ya lo incorporan en sus productos para drones, no está contemplado como parte de la reciente actualización del Part 107 (equivalente a la DAN 151 para Chile) que será obligatorio en EEUU a partir del 2024 (https://www.faa.gov/uas/commercial_operators/).

Referencias

1. Estimación usada en el PX4
https://docs.px4.io/master/en/advanced_config/tuning_the_ecl_ekf.html
2. Bucles de control en el PX4
https://docs.px4.io/master/en/flight_stack/controller_diagrams.html#multicopter-control-architecture

Contacto

Tomas I. Opazo
toopazo (at / arroba / @) protonmail.com