

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e implementación de capacidades automáticas de  
navegación para un Robot Explorador Modular**

Trabajo de graduación presentado por Javier Ernesto Archila Murillo  
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022







UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e implementación de capacidades automáticas de  
navegación para un Robot Explorador Modular**

Trabajo de graduación presentado por Javier Ernesto Archila Murillo  
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022



Vo.Bo.:

(f)   
\_\_\_\_\_  
Ing. Jonathan de Los Santos

Tribunal Examinador:

(f)   
\_\_\_\_\_  
Ing. Jonathan de Los Santos

(f)   
\_\_\_\_\_  
Ing. Carlos Alberto Esquit Hernandez

(f)   
\_\_\_\_\_  
Ing. Jose Eduardo Morales Espinoza

Fecha de aprobación: Guatemala, 08 de enero de 2022 .



Los rovers exploratorios han logrado hazañas increíbles a lo largo de la historia tecnológica de la humanidad, siendo los más famosos los rovers lunares. Estos, como su nombre lo indica, exploran diversos terrenos de difícil alcance para los humanos. En este trabajo de graduación, el rover exploratorio tiene como función principal explorar terrenos de difícil acceso ubicados en Guatemala, como fincas, terrenos volcánicos y lugares con riesgos a deslaves, entre otros.

Para que un vehículo pueda moverse de forma autónoma y realizar tareas durante horas, debe tener un auto piloto capaz de procesar todas las señales de entrada y salida de los diversos componentes que estos llevan. En el caso de este proyecto de graduación se utilizó un controlador Pixhawk ya que, por trabajos anteriores de fotogrametría aérea en fincas del sur de Guatemala, se contaba con dicho auto piloto, sensores y el conocimiento de configuración, pruebas y funcionamiento completo de este. El trabajo mencionado anteriormente constaba en construir, programar y pilotear aviones a control remoto con autonomía de 45 minutos de vuelo y cámaras profesionales por dentro para tomar fotos de alto detalle a una altura promedio de 100 metros, las cuales pasaban por un post-procesamiento para realizar un mapa de la finca utilizado para delimitar y estudiar cultivos, principalmente. Durante años, este trabajo representaba retos continuos de programación de nuevas funciones, tuneo de controladores PID en las aeronaves e incluso la fabricación de vehículos de despegue y aterrizaje vertical (VTOL, por sus siglas en inglés), donde cada proyecto y cada prueba exitosa representaba una satisfacción personal que se convirtió en lo que hoy es mi pasión por la creación de vehículos autónomos y, desde entonces, la experimentación con diferentes tipos de vehículos (incluyendo aerodeslizadores y cuadricópteros con los rotores en línea) ha sido constante. Otro trabajo que ha alimentado la pasión por vehículos a control remoto es el de grabación profesional con drones FPV (*First Person View*, por sus siglas en inglés), los cuales son drones de carrera adaptados para grabación profesional dirigida a producción de publicidad. La afición por este tipo de drones surgió a través de vídeos en internet y el conocimiento adquirido previamente sobre trabajos con controladores de vuelo, y hoy en día es un pasatiempo que ha abierto muchas puertas y ha permitido conocer gran parte de Guatemala con el objetivo de grabar lugares turísticos desde otro punto de vista.

Con este conocimiento se emprendió en el trabajo de graduación que no solo cumplirá con los objetivos del mismo, sino que su aporte a la comunidad de la Universidad del Valle

de Guatemala y a la sociedad representa arriesgar en menor cantidad vidas humanas para el estudio de terrenos adversos encontrados en diversas partes del territorio nacional a través de los diversos sistemas implementados en el Rover Exploratorio Modular

El sistema generado para el control físico y procesamiento de señales del Rover tiene como base una estructura pre-existente de un proyecto anterior, la cual se arregló y modificó para cambio de motores, colocación de componentes eléctricos y reducir peso. Respecto al procesamiento, utiliza la computadora de compañía Raspberry Pi 3 conectada serialmente al Pixhawk, y para movilizarse utiliza dos motores DC que, acoplados por medio de engranajes y cadenas, mueve sus orugas a través de terreno complicado ya que las orugas, como los tanques, son utilizadas para terrenos de todo tipo. Para controlar a distancia el Rover, se utiliza el auto piloto Pixhawk capaz de generar las señales de salida necesarias para la computadora de compañía y el *driver* de los motores. La computadora de compañía, la cual recibe su nombre de la acción de acompañar al auto piloto para procesamiento de datos y envío de telemetría, recibe información de la cámara y el auto piloto para luego transmitirla, a través de la red, a la computadora del usuario, logrando así un control remoto dependiente de la señal celular. Asimismo, se programó un control RF para pruebas a distancia corta y media, con latencia mínima.

Este sistema utiliza un módem de internet conectado directamente a la Raspberry Pi 3 para lograr la conectividad en cualquier lugar, con la única desventaja de que depende de la señal celular y genera una latencia mayor a un sistema de transmisión de RF. Una gran ventaja de este sistema es que permite la redundancia de control remoto, dándole prioridad al control por medio de RF cuando ambas señales estén en línea y contando con la capacidad de realizar pruebas de campo simultáneas, así como tener siempre un sistema de control remoto de emergencia (o *backup*).

Todo esto ha sido posible gracias a mis padres, quienes me han apoyado en todo momento para realizar mis proyectos. Sin ellos, definitivamente no hubiera sido posible llegar a donde estoy hoy, tener lo que tengo hoy y saber lo que sé hoy. Estaré eternamente agradecido.

<b>Prefacio</b>	<b>VI</b>
<b>Lista de figuras</b>	<b>IX</b>
<b>Resumen</b>	<b>XII</b>
<b>Abstract</b>	<b>XIV</b>
<b>I Introducción</b>	<b>1</b>
<b>II Antecedentes</b>	<b>3</b>
<b>III Justificación</b>	<b>5</b>
<b>IV Objetivos</b>	<b>7</b>
<b>V Alcance</b>	<b>9</b>
<b>VI Marco teórico</b>	<b>11</b>
<b>VII Configuración inicial de Mission Planner</b>	<b>17</b>
A Instalación y configuración inicial en Mission Planner . . . . .	17
B Conexión de auto piloto a la computadora y cambio de firmware . . . . .	18
C Parámetros para señal de motores . . . . .	19
D Parámetros para control RF y a través de red celular . . . . .	21
E Parámetros para comunicación serial . . . . .	22
F GPS . . . . .	23
G Control de Rover por medio de Internet . . . . .	24
<b>VIII Conexión de Pixhawk a Raspberry Pi</b>	<b>27</b>
A Conexión eléctrica . . . . .	27
B Transmisión de datos . . . . .	28
1 Configuración para comunicación serial . . . . .	29

<b>IX Resultados</b>	<b>31</b>
A Pruebas iniciales de estructura y motores . . . . .	31
B Integración de computadora de compañía a sistema de control remoto . . . . .	34
C Manejo de Rover a través de la red utilizando Joystick y control por RF . . . . .	37
<b>X Conclusiones</b>	<b>39</b>
<b>XI Recomendaciones</b>	<b>41</b>
<b>XII Bibliografía</b>	<b>43</b>
<b>XIII Anexos</b>	<b>45</b>
A Guías externas . . . . .	45
B Pruebas realizadas en video . . . . .	45

---

## Lista de figuras

---

1 Autopiloto Pixhawk 4 . . . . .	12
2 Driver DC de 10A de doble canal . . . . .	13
3 Sistema general de video análogo . . . . .	14
4 Sistema general de video digital (sin antenas) (DJI Air Unit) . . . . .	14
5 Sensor ultrasónico . . . . .	15
6 Raspberry Pi 3 . . . . .	15
7 Pantalla de inicio de Mission Planner . . . . .	18
8 Lista de puertos y opciones de baudaje . . . . .	18
9 Selección de firmware en Mission Planner . . . . .	19
10 Parámetro para cambiar el tipo de drivers de motor . . . . .	20
11 Pantalla de cambio de funciones a salidas PWM . . . . .	20
12 Ejemplo de cambio de parámetro para salida digital tipo <i>relay</i> . . . . .	20
13 Receptor satélite DSMX conectado a Pixhawk con puerto dedicado . . . . .	21
14 Pestaña de prueba de salidas y enlazado de control RF . . . . .	22
15 Parámetros de comunicación serial . . . . .	23
16 Visualización de conectores y puertos de entrada de GPS . . . . .	23
17 Control de Joystick para canal de Throttle (acelerador) . . . . .	24
18 Control de Joystick para canal de Roll (giro) . . . . .	25
19 Control de Joystick para canal de cambio de modo . . . . .	25
20 Diagrama de conexión entre Pixhawk y Raspberry, y conexión física . . . . .	28
21 Driver accionando un motor DC a través de señales realizadas en el control RF . . . . .	32
22 El Rover subiendo una grada de ingreso a una casa . . . . .	32
23 El Rover subiendo una banqueta para probar la estabilidad . . . . .	33
24 Motor actualizado . . . . .	33
25 Ventana emergente de GStreamer con el video proveniente de la Raspberry . . . . .	36
26 Electrónicos completos en Rover antes de realizar pruebas de campo . . . . .	36
27 Control de Rover con Joystick . . . . .	37
28 Control de Rover con Joystick . . . . .	37
29 Prueba de adaptación de terreno . . . . .	38



La implementación de un controlador auto piloto con capacidades de navegación semi-inteligentes en una plataforma Rover existente parte de la aplicación del conocimiento mecánico y electrónico en un ámbito físico para aplicaciones directas. El trabajo de graduación siguiente explica el proceso de reutilización y adaptación de una plataforma de un Rover existente utilizado años atrás en distintos proyectos para la implementación de capacidades autónomas, manuales y el envío de información a través de Internet gracias al auto piloto, la computadora de compañía y los sensores presentes, como el de inclinación, temperatura, altura y posición, así como la capacidad de transmisión de vídeo a través de Internet o radio frecuencia (dependiendo del rango necesario y latencia mínima).

El Rover exploratorio puede utilizarse en aplicaciones de alto riesgo como análisis de terrenos peligrosos (ceranos a volcanes, animales peligrosos o lugares con riesgo humano general), exploración de ubicaciones desconocidas y manipulación limitada de un objeto lejano por medio de la implementación de un brazo robótico añadido el cual sería un módulo aparte acoplado a la estructura. Dicho Rover también permite la experimentación con sistemas de largo alcance a través de redes celulares con el objetivo de recopilar información pertinente a la ubicación.

El auto piloto utilizado se conoce como Pixhawk, y es el cerebro de la operación. Permite controlar todo tipo de entradas, salidas y sensores necesarios para casi cualquier aplicación, y se configuró por medio de un programa llamado Mission Planner. Adicionalmente, se utilizó una computadora de compañía Raspberry Pi 3 para el manejo y envío de datos a través de internet. El Rover es alimentado por baterías de polímero de litio (*LiPo*, por sus siglas en inglés) de 24v nominales para una autonomía relativamente alta, y dichas baterías fueron monitoreadas para evitar una sobre descarga.

La computadora de compañía es una computadora portátil sin pantalla ni teclado, solo la tarjeta, y permite tener la versatilidad de una computadora en un empaque pequeño, ligero y eficiente comparado a laptops. La Raspberry se encargó de unificar la señal de vídeo proveniente de la cámara conectada a ella misma y los paquetes de telemetría provenientes del PixHawk, enviándola por medio de internet a un servidor. Se le conectó un módem 3G a uno de sus puertos USB y de esa manera se obtuvo conexión a internet rápida, simple y segura.

La forma de movimiento del rover se basa en dos motores DC con una caja reductora, la cual reduce la velocidad pero aumenta la eficiencia y torque de los mismos, por lo que, acoplados a un sistema de 24 voltios, los motores son bastante eficientes. Estos motores se controlan por medio de un driver que utiliza señales PWM generadas por el auto piloto Pixhawk, así como señales de relay digitales para controlar la dirección de giro. La implementación de los módulos mencionados anteriormente se realiza por medio de comunicación serial, señales PWM y señales digitales, las cuales permiten la comunicación entre componentes que, a su vez, permite controlar el Rover a distancia.

Antes de implementar lo mencionado anteriormente, se necesitó realizar algunas modificaciones a la estructura existente para adaptarla a nuestra necesidad, y dichas modificaciones incluyen modernizar el diseño, reducir la altura del chasis, colocar bases para los diferentes módulos y realizar ajustes generales a las orugas.

Otro resultado significativo fue una prueba realizada con el Pixhawk conectado a una Raspberry con sistema operativo APISync. En esta prueba se logró conectar inalámbricamente la Raspberry, que ya recibía telemetría MAVlink y señal de vídeo, a la computadora por medio de la señal Wifi generada por el módulo de la Raspberry, comprobando que es posible conectar y transmitir inalámbricamente la telemetría y vídeo a través de frecuencias y ancho de banda de Wifi.

The implementation of an autopilot controller with semi-intelligent navigation capabilities on an existing Rover platform starts from the application of mechanical and electronic knowledge in a physical domain for direct applications. The following thesis explains the process of reusing and adapting an existing Rover platform (used years ago on another project) for the implementation of autonomous and manual capabilities, as well as sending telemetry information through the Internet thanks to the on board sensors such as inclination, temperature, height and position, all this along with the capability of video transmission through Internet or radio frequency (depending on the necessary range and minimum latency).

The exploratory rover can be used in high-risk applications such as analysis of dangerous terrain (close to volcanoes, dangerous animals or places with general human risk), exploration of unknown locations and limited manipulation of a distant object through the implementation of an added robotic arm, an external module coupled to the structure. This Rover also allows cellular-linked long range systems experimentation with the objective of gathering pertinent information about the terrain it's on.

The autopilot to be used is known as the Pixhawk, and it's the brain of the operation. It allows controlling all types of inputs, outputs and sensors necessary for almost any application, and was configured through a program called Mission Planner. Additionally, a Raspberry Pi 3 companion computer was used to manage and send data over the internet. The rover is powered by a 24v nominal lithium polymer (*LiPo*) battery for relatively high autonomy, and these batteries are monitored to prevent over discharge.

The companion computer is basically a laptop without a screen or keyboard, just the board, and allows you to have the versatility of a computer in a small, light and efficient package compared to laptops. The Raspberry is in charge of unifying the video signal coming from the camera connected to itself and the telemetry packets coming from the PixHawk, to send it through the internet to a server. A 3G modem was connected to one of its USB ports to obtain a fast and secure internet connection.

The rover's form of movement is based on two DC motors with a gearbox, which reduces the speed but increases their efficiency and torque, so that, coupled to a 24-volt system, the motors are quite efficient. These motors are controlled by a driver that uses PWM signals

generated by the Pixhawk autopilot, as well as digital relay signals to control the direction of turn. The implementation of the modules was made through serial communication, PWM signals aswell as digital signals, all which allow the components to communicate with each other and control the Rover from long distances.

Before implementing the aforementioned, some modifications were needed for the existing structure to adapt it to our needs, and these modifications include modernizing the design, reducing the height of the chassis, placing mounting points for the different modules and making general adjustments to the tracks.

La necesidad de un Rover exploratorio se vuelve evidente cuando la salud humana se ve en riesgo de llegar a un ambiente hostil, y es por esto que los rovers son comúnmente utilizados en aplicaciones como desactivar bombas, vigilancia, exploración en terrenos peligrosos o incluso en explorar otros planetas y satélites naturales.

En este trabajo de graduación se presenta el proceso de creación de un rover autónomo tipo tanque el cual tiene capacidad de transmitir vídeo con latencia mínima a cualquier parte del mundo por medio de internet y transitar por terrenos de difícil acceso gracias a sus orugas tipo tanque.

Dicho rover trabaja por medio de un auto piloto Pixhawk con firmware ArduRover y configurado en Mission Planner para el procesamiento de señales de los sensores de entrada, y cuenta también con una computadora de compañía Raspberry Pi 3, la cual recibe información de sensores en telemetría MAVlink proveniente del Pixhawk por medio de comunicación serial, y dicha telemetría consta de información completa de posición y variables necesarias para el correcto funcionamiento del rover.



Durante años, los vehículos no tripulados han sido de gran ayuda en aplicaciones como seguridad internacional (drones de reconocimiento aéreo), estudios de cultivos (drones de fotogrametría), fotografía aérea estructural (drones de fotografía y vídeo) y manejo de explosivos (robots anti-explosivos terrestres). Estos vehículos permiten la recolección de información de manera segura y remota, así como la manipulación de objetos dentro o fuera del mismo rover.

El Rover más famoso actualmente es el Perseverance, el cual aterrizó en el planeta Marte en febrero del presente año. Este Rover diseñado específicamente para la aplicación de exploración planetaria realiza misiones puramente exploratorias y revelan datos importantes del planeta marciano para su estudio posterior, el cual revela características vitales de la historia del planeta.

De acuerdo a lo mencionado anteriormente, se adquiere que la función principal de un Rover exploratorio es, efectivamente, explorar el terreno en el cual se encuentra. Pueden existir distintas categorías de Rovers, como anti explosivos, búsqueda y rescate, o incluso pilotado por una persona, y cuentan con módulos diversos para aplicaciones específicas, como recolección y análisis de muestras, toma de fotografías espectrales y reconocimiento de lugar.

En vehículos no tripulados semi-automatizados se utilizan diversos módulos que trabajan en sincronía para realizar las misiones, y en estas se necesita un auto piloto o controlador programable que sea capaz de generar las señales y realizar el procesamiento necesario para guiar automáticamente el vehículo de acuerdo al tipo de mismo, los sensores utilizados y la información programada. En la programación deben entrar variables de telemetría, las cuales incluyen factores de velocidad, tiempo, altura, inclinación, dirección y toman datos de las baterías, motores y otros sensores que pueden acoplarse como tubos de Pitot, sensores de corriente, barómetros, entre otros.

Los rovers con aplicaciones directas, como anti-explosivos, han demostrado ser de gran ayuda en minimizar riesgos a la salud humana, asegurar perímetros y explorar áreas peli-

grosas y todo, remotamente.

El proyecto del Rover Exploratorio Modular surge de la reutilización de un proyecto realizado en la Universidad años atrás, donde se creó un robot con el objetivo de limpiar playas. Luego que el proyecto anteriormente mencionado concluyera, el Rover, ya desarmado e incapaz de cumplir su objetivo inicial, se mantuvo guardado en las instalaciones de la Universidad para posteriormente ser utilizado en un proyecto moderno, modificándolo, rearmándolo y reutilizándolo. La idea inicial para el proyecto de graduación fue de crear un robot anti-explosivos para la unidad anti-bombas de la Policía Nacional Civil, pero luego de problemas de presupuesto y comunicación con dicha institución, se decidió colocar dicho proyecto por un lado y empezar uno propio utilizando la plataforma existente. El medio de movimiento de la plataforma existente de Rover son dos orugas paralelas, asimilando el sistema de un tanque moderno. Este sistema cuenta con diversas ventajas como, significativamente, más tracción comparado a llantas todo terreno en terrenos complicados y simplicidad en sistema de giro, sin embargo no puede ser tan ágil ni rápido como un vehículo tradicional [1].

---

## Justificación

---

Un Rover es una herramienta muy útil para aplicaciones que no requieran velocidad ni altura, como reconocimiento de terrenos en tierra, toma de muestras del suelo para análisis, exploración en lugares peligrosos (cerca de lava volcánica, riscos e incluso amenazas de bomba) o algo tan simple como una cortadora de grama automatizada. Las aplicaciones de este tipo de vehículos son bastante amplias y por medio de módulos adaptables a este, pueden ser utilizados para aún más funciones como recuperación de objetos, exploración nocturna, movilización de materiales, entre otros. A través de este proyecto se busca construir un Rover capaz de ser controlado de forma remota a distancia cercana y también por Internet, con capacidad de transmisión de vídeo a través de la web, enviar telemetría del terreno donde se encuentra al usuario y autonomía suficiente para una misión considerable de exploración.

La importancia del proyecto nace de la creación y documentación de un sistema confiable de transmisión de datos y manejo de un vehículo no tripulado a través de largas distancias utilizando sistema de transmisión de datos accesible. La creación del Rover tiene como objetivo aplicar dicho sistema de transmisión de datos y tomarlo como prueba de concepto, así como generar una plataforma física confiable capaz de realizar las actividades mencionadas anteriormente. En pocas palabras, el Rover representa la capacidad de un sistema con auto piloto conectado a la red que se pueden crear sistemas de largo alcance para todo tipo de vehículos, ya que por medio del auto piloto Pixhawk, se puede instalar un firmware de controlador auto piloto en cualquier tipo de vehículo como avión, bote, Rover, carro, entre otros, con aplicaciones diversas desde simple observación hasta mapeo de terrenos o cuerpos de agua, con una inversión económica menor a realizar el trabajo con vehículos tripulados, y con mucho menor riesgo.

Este proyecto se enfoca principalmente en la implementación y configuración de un *controlador de vuelo* o *auto piloto*, configuración e implementación de una computadora de compañía Raspberry, implementación de cámaras de transmisión de vídeo, sensores de proximidad y de telemetría general, y posiblemente programar una inteligencia semi-artificial que le permita al Rover detectar obstáculos imposibles de pasar, alertar al usuario y tomar decisiones por su cuenta [2]. Todo lo anterior con el objetivo de documentar el proceso y

resultado de la creación de un sistema confiable de manejo de vehículos no tripulados a través de la red celular encontrada en el territorio nacional de Guatemala.

### 4.1. Objetivo general

- Crear un rover exploratorio con capacidades automáticas de navegación utilizando una plataforma existente disponible en la Universidad del Valle de Guatemala, con la capacidad de mapeo terrestre, reconocimiento remoto de terrenos de alto riesgo, enviar a través de Internet telemetría importante como temperatura, humedad, imagenes y videos.

### 4.2. Objetivos específicos

- Reutilizar la plataforma del rover proporcionado por la Universidad para convertirlo en un rover exploratorio, añadiéndole capacidades autónomas, de control remoto por RF e Internet y transmisión de telemetría a través de red celular.
- Utilizar nuevos componentes electrónicos relacionados a la etapa de potencia como controladores de motor y motores para mayor autonomía, mayor eficiencia y mayor velocidad, reguladores de voltaje eficientes y confiables, así como utilizar baterías de mayor voltaje y capacidad que permitan reducir la corriente total del sistema de transmisión de potencia.
- Utilizar el autopiloto Pixhawk para programar las funciones totales del rover en conjunto con una Raspberry Pi para manejo de telemetría a través de Internet y manejo remoto y automático.
- Documentar el proceso completo por medio de imágenes, videos, archivos en 3D, códigos utilizados, documentación del autopiloto, planos de la estructura, entre otros.
- Integrar una plataforma confiable de transmisión de datos por Internet utilizando la Raspberry Pi y Pixhawk para enviar y recibir imágenes, telemetría y poder controlar estos de manera autónoma, así como cambiar la configuración de los mismos.

- Implementar distintas formas de manejo remoto como lo es a través de Internet, así como manejo en corta distancia utilizando un control dedicado.
- Generar manuales de usuario referentes a la utilización de los programas y módulos utilizados junto a la integración de los mismos.

La creación de un sistema confiable utilizando tecnología al alcance de todos representa la facilidad de construir sistemas de control de largo alcance los cuales pueden ser aplicados a cualquier vehículo. En este caso se utiliza la plataforma de un Rover con orugas, sin embargo, la transmisión de información a través de internet se mantiene igual para cualquier vehículo utilizando los componentes principales utilizados en este proyecto. El resultado de este trabajo de graduación conviene a la comunidad científica no solo de la Universidad del Valle de Guatemala sino a cualquier individuo interesado en la creación de sistemas de control de largo alcance con el conocimiento previo básico sobre manejo de microcontroladores y trabajo con controladores de vuelo, vehículos a radio control y electrónicos en general.

El trabajo se realizó a lo largo de un año en las instalaciones de la Universidad, teniendo accesibilidad a una plataforma física preexistente de un proyecto anterior sobre la cual trabajar, que requería arreglos mínimos, y los componentes electrónicos utilizados fueron proporcionados por los integrantes del grupo de trabajo. La compra de componentes electrónicos nuevos fue mínima y se mantuvo el presupuesto del proyecto generalmente bajo, ya que se contaba con la mayoría de piezas con anterioridad.

Una de las mayores limitaciones encontradas durante el proyecto fue la utilización de softwares desconocidos para la implementación de la comunicación entre la computadora de compañía y el auto piloto, la cual conllevó varias pruebas y diversos sistemas operativos para llegar a un sistema final. Esto, junto a la pandemia actual de COVID-19 que generó problemas de trabajo presencial y pruebas de campo del Rover, limitó considerablemente los avances durante algunos meses del año.



La mayor ventaja de los robots exploradores es la capacidad de acercarse, manipular y reconocer terrenos, materiales y objetos ubicados en lugares que atentan contra la seguridad de individuales, son de difícil acceso o simplemente es más conveniente enviar un rover. Esto lo realiza por medio de diversos sistemas embebidos dentro del robot, como módulos de movimiento (motores eléctricos con sus respectivos drivers y etapas de potencia), módulos de transmisión de información en tiempo real (cámaras y transmisores de video de baja latencia para controlar el robot por medio de una cámara ubicada dentro del armazón, así como telemetría de datos importantes de voltajes, corrientes o un indicador de intensidad de la señal recibida [o RSSI por sus siglas en inglés]) y módulos de potencia (baterías, reguladores de voltaje step-up o step-down y filtros de ruido electrónico), entre otros.

La capacidad de movimiento generalmente es obtenida por medio de dos orugas controladas independientemente por medio de motores eléctricos sin escobillas. Estos motores cuentan con diversas ventajas respecto a los motores comunes DC con escobillas, siendo la principal eficiencia y potencia.

Las orugas, como medio de movimiento, cuentan con ventajas y desventajas al momento de utilizarlas en un rover. Los primeros usos de las orugas fueron en carretas agrícolas, y contaban con la ventaja de tener más área superficial que generaba más tracción en superficies resbaladizas, como prados llenos de lodo [3]. Por ejemplo, la mayoría de los rovers enviados a otros planetas, si no es que todos, utilizan llantas en vez de orugas. Esto se debe a que el peso es una característica crítica al momento de viajes espaciales; una libra de peso cuesta alrededor de \$10,000 para colocar en órbita, no digamos para realizar un viaje de meses y aterrizar en otro planeta [4].

Existen diversos módulos de autopiloto que permiten realizar las funciones necesarias para este proyecto, sin embargo, el que se utilizará es el Pixhawk.

## Pixhawk

El Pixhawk 1 es una placa autopiloto avanzada que permite volver autónomo casi cualquier tipo de vehículo por medio de diferentes firmwares, y cuenta con un procesador F7 que es de los más potentes de su categoría. Esta placa cuenta con varias entradas y salidas con capacidades de comunicación SPI, PWM, I2C, MavLink y cuentan con diversos GPIO (General Purpose Input Output) para salidas y entradas digitales o análogas y un compás interno.

A este módulo se le conectarán sensores necesarios para el rover, como el GPS, sensor de voltaje y módulo de telemetría general. También irán conectadas dos salidas PWM y dos salidas digitales que se conectan al driver DC de 10A para el control de los motores por medio de un protocolo llamado "brushed with relay". Este protocolo básicamente toma la señal PWM en conjunto con una entrada digital para controlar el motor: si la entrada digital es cero, el motor gira a un lado de acuerdo al pulso variable del PWM (de 1000us a 2000us) y si la entrada digital es uno, el motor gira en sentido contrario utilizando el mismo rango PWM de 1000us a 2000us para obtener, de esta manera, más resolución de salida hacia el motor en vez de dividir el periodo del PWM de 1000us a 1500us y de 1500us a 2000us [2].



Figura 1: Autopiloto Pixhawk 4

La programación del Pixhawk se realiza mediante un programa llamado Mission Planner, dentro del cual se descarga y sube el firmware a utilizar, se le configuran diversos parámetros para el funcionamiento correcto, se escoge el tipo de protocolo de motores a utilizar, se declaran las entradas y salidas y se programan modos de manejo a través de la conexión por control remoto.

## Driver DC de 10 A

El driver es un componente esencial para el correcto funcionamiento de los motores Dc a utilizar en en rover. Como su nombre lo indica, el driver es un controlador capaz de generar un voltaje DC invertible a partir de una entrada PWM y otra digital por motor (con capacidad de dos motores, 4 entradas en total). Este voltaje DC invertible es el que hace girar el motor de escobillas en un sentido o en otro, y el rango del mismo es de cero voltios hasta el voltaje de alimentación, que puede ser hasta 30v DC [5].

Las entradas necesarias de este driver para su funcionamiento serán salidas del Pixhawk, las cuales se configuran en Mission Planner para obtener las salidas digitales y la frecuencia PWM adecuada.



Figura 2: Driver DC de 10A de doble canal

## Sistema de transmisión de video

La transmisión de video con mínima latencia requiere de sistemas de alta potencia y velocidad, normalmente con conexión directa a través de radio frecuencia desde el emisor hasta el receptor. Existen dos tipos de video, análogo y digital.

La transmisión de video análogo es de muy baja latencia, de aproximadamente 30ms en total, sin embargo una gran limitación es el rango y la calidad, ya que dicho rango puede extenderse hasta 10 kilómetros en LOS (Line Of Sight o Línea De Visión, por sus siglas en inglés) que requiere que no se encuentre ningún obstáculo entre el emisor y el receptor, y la calidad de video llega a 360p o 480p, dependiendo de la calidad de la cámara utilizada y las antenas. Esta transmisión ocurre generalmente a una frecuencia de 5.8GHz y se utilizan antenas circularmente polarizadas para maximizar rango y cobertura radial.

El otro sistema de transmisión de video es digital, el cual utiliza dos bandas anchas de frecuencia para transmitir y recibir datos con una latencia de alrededor de 50ms y frecuencias

de 2.4GHz y 5.8GHz en conjunto. Este sistema tiene un rango un poco mayor al análogo debido a la menor frecuencia utilizada, y generalmente tiene más poder de transmisión (transmisores de grado hobby hasta 1200mW). La calidad del video llega hasta 820p [6].



Figura 3: Sistema general de video análogo



Figura 4: Sistema general de video digital (sin antenas) (DJI Air Unit)

## Sensores de proximidad

Existen diversos sensores de proximidad con diferentes protocolos de comunicación, y los más comunes son los sensores ultrasónicos que funcionan enviando una onda de sonido inaudible para los humanos y detectando el rebote de dicha onda después de un tiempo, y de acuerdo a este tiempo que le toma a la onda rebotar y regresar se calcula la distancia hacia un obstáculo u objeto que se encuentre frente a él. Este sensor funciona por medio de comunicación serial a 9600 baudios (bits por segundo). Este sensor puede ser conectado a virtualmente cualquier microcontrolador y su programación es básica y simple [7].



Figura 5: Sensor ultrasónico

## Raspberry Pi 3

La Raspberry Pi es una computadora miniatura con una capacidad de procesamiento alta en comparación al tamaño de la misma. Es uno de los microcontroladores más famosos del mundo y cuenta con diversas entradas y salidas, así como con un sistema operativo, diferentes memorias e incluso entradas de tipo USB para conectar teclados, mouses, módems, etc. Esta computadora es versátil ya que, por el pequeño tamaño y poco peso, es móvil y de fácil uso. Utiliza el sistema operativo Linux y funciona como una computadora normal.

Esta computadora irá a bordo del rover y se encargará, a grandes rasgos, de controlar remotamente el Pixhawk y enviar la telemetría recolectada de sensores conectados a ella misma a través de Internet, así como permitir un control remoto a distancia (por Internet también) [8].



Figura 6: Raspberry Pi 3

## Mission Planner

Como mencionado brevemente, Mission Planner es una aplicación que permite la conexión y programación de microcontroladores autopiloto como el Pixhawk y sus diferentes versiones, así como otros controladores de vuelo.

Este programa facilita grandemente la configuración del rover ya que permite declarar las entradas y salidas que se tendrán presentes, el tipo de conexión a utilizar para controlar los motores, la entrada de sensores hacia él mismo y la configuración de modos de manejo (automático o manual, principalmente), creación de misiones autónomas y procedimiento a seguir en caso de pérdida de señal (FailSafe, en inglés) [9].

---

### Configuración inicial de Mission Planner

---

Mission Planner es una herramienta útil cuando se necesita de una GUI en la cual se pueda controlar un auto piloto. En este caso, a través de Mission Planner es que se carga el firmware de ArduRover, se programan parámetros esenciales para el funcionamiento del mismo (únicos para el Rover), se maneja remotamente y se recibe la telemetría del Rover.

#### **A. Instalación y configuración inicial en Mission Planner**

Como mencionado anteriormente, Mission Planner es una aplicación de computadora que permite la conexión y programación de microcontroladores auto pilotos dedicados a la generación de capacidades autónomas para diferentes tipos de vehículos.

Esta aplicación se puede descargar en la página oficial de [www.ardupilot.org](http://www.ardupilot.org). En el capítulo de anexos se encuentra una guía externa, ya que en este trabajo no es un tema a explicar a detalle.

Luego de la instalación del programa, se cuenta con una pantalla de inicio que muestra pantallas de telemetría, sensores, entradas y salidas digitales y un mapa mundial que se actualiza al recibir una señal válida de GPS por medio del Pixhawk conectado. Desde aquí se aprecia la complejidad del programa en cuanto a variables posibles que se le pueden asignar al vehículo de interés.



Figura 7: Pantalla de inicio de Mission Planner

## B. Conexión de auto piloto a la computadora y cambio de firmware

Conectando el Pixhawk a la computadora por medio de USB por primera vez, se debe cargar y actualizar el firmware de ArduRover, el cual es un firmware dedicado a vehículos terrestres que puedan utilizar auto piloto y cuenten con las mismas características generales. Este firmware limita las capacidades del Pixhawk a las funciones utilizadas solo por vehículos terrestres, e inhibe capacidades de vuelo (en el caso de utilizar el firmware dedicado a aviones) para ahorrar poder de procesamiento principalmente.

Con el Pixhawk conectado, la computadora reconocerá el dispositivo y le asignará un puerto COM, el cual debe seleccionarse en la lista de puertos ubicada en la esquina superior derecha. Cambiar el baudaje a 115200 es necesario. No se debe oprimir el botón de conectar aún.



Figura 8: Lista de puertos y opciones de baudaje

Luego, en la pestaña de SETUP, en la columna de opciones de lado izquierdo, se encuentra una opción de *Install firmware*. Al entrar a esta, aparecerán distintas gráficas mostrando los diversos vehículos a los cuales es aplicable un software de Ardupilot. En este caso, se selecciona la imagen/opción de ArduRover, y se oprime "Yes".

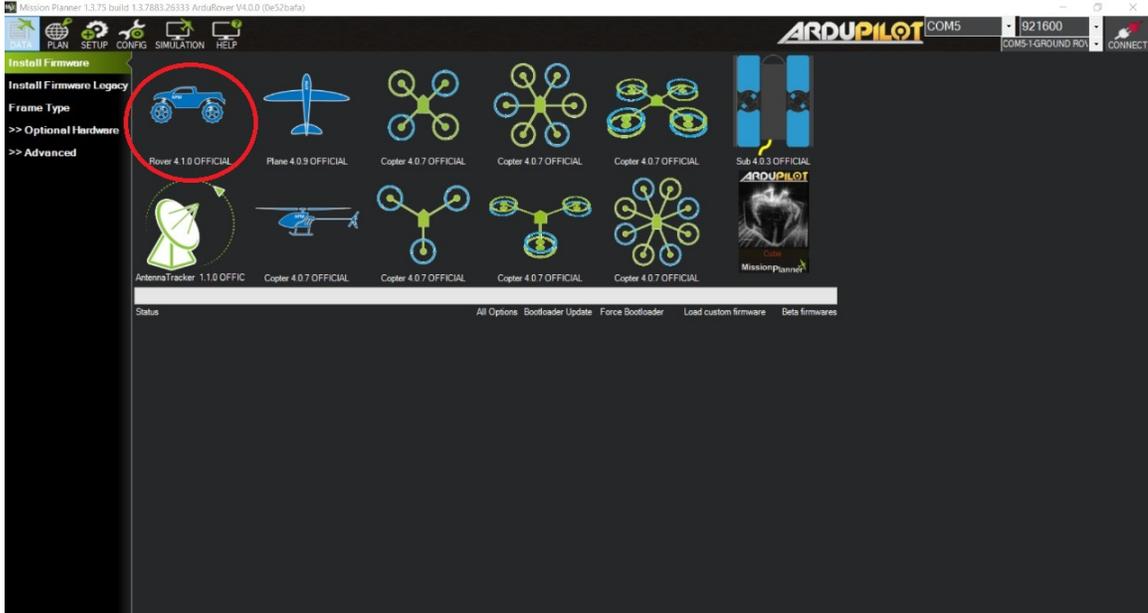


Figura 9: Selección de firmware en Mission Planner

Es probable que Mission Planner requiera que se desconecte y conecte el Pixhawk durante el proceso, y si este fuera el caso, se debe desconectar, presionar OK y conectarlo de nuevo.

Luego de esto, el firmware de ArduRover debería estar instalado correctamente en el Pixhawk.

### C. Parámetros para señal de motores

Dentro del firmware de ArduRover se encuentran distintas configuraciones para vehículos con diferentes sistemas de movilización. Estos pueden ser de orugas, llantas y timón (como un carro normal) o llantas omnidireccionales, por lo que se deben asignar los canales de salida de acuerdo al tipo de sistema que lleva el Rover.

El primer paso para configurar el tipo de Rover que se estará utilizando es definir el tipo de drivers que se usará. En este caso, se utilizó un Driver DC con entradas PWM y de *relay* (entradas digitales), y a este tipo de Drivers se les llama "*Brushed With Relay*", haciendo énfasis en que los motores DC se les conoce como *brushed* por su definición en inglés. Para definir este tipo de driver se modifica el parámetro llamado "MOT\_PWM\_TYPE" se le da un valor de 3.

NODES	0		0:Manual 1:Acro 3:Steering 4:Hold 5:Loiter 6:Follow 7:Simple 10:Auto 11:RTL 12:SmartRTL 15:Guided	Dr
MOT_PWM_FREQ	16	kHz	120	Mo
MOT_PWM_TYPE	3		0:Normal 1:OneShot 2:OneShot125 3:BrushedWithRelay 4:BrushedBiPolar 5:DShot150 6:DShot300 7:DShot500 8:DShot1200	Th use sep
MOT_SAFE_DISARM	0		0:PWM enabled while disarmed 1:PWM disabled while disarmed	Dis Th

Figura 10: Parámetro para cambiar el tipo de drivers de motor

En la página de Initial Setup, en la pestaña de Servo Output, aparecen 8 canales de salidas PWM que puede generar el Pixhawk. En el caso del Rover, solo se utilizarán dos señales PWM de salida ya que solo cuenta con dos motores que deben funcionar a velocidades distintas. Estos dos canales pueden ser escogidos aleatoriamente, pero por conveniencia se tendrá el canal 1 como la salida del motor izquierdo y el canal 2 como la salida del motor derecho.



Figura 11: Pantalla de cambio de funciones a salidas PWM

De acuerdo al tipo de Driver DC que se está utilizando se necesitan otras dos señales de salida, las cuales son digitales y le permiten saber a dicho driver la dirección de los motores (adelante o atrás). Esta señal digital, referida como *relay*, trabaja en conjunto a la señal PWM. Entiéndase, la señal PWM es un indicador de la velocidad y la señal digital de *relay* es un indicador de dirección. En este caso, se utilizan dos salidas digitales, y estas serán el parámetro RELAY\_PIN y RELAY\_PIN2 con los valores de 50 y 51, respectivamente. Esto activa las señales digitales en los pines de salida físicos AUXOUT1 y AUXOUT2

RELAY			
RELAY_DEFAULT	0	0:Off 1:On 2:NoChange	The state of the relay on boot.
RELAY_PIN	12	-1:Disabled 13:APM2 A9 pin 47-APM1 relay 50:Pixhawk AUXOUT1 51:Pixhaw...	Digital pin number for first relay control. This is the pin used for camera control.
RELAY_PIN2	-1	-1:Disabled 13:APM2 A9 pin 47-APM1 relay 50:Pixhawk AUXOUT1 51:Pixhaw...	Digital pin number for 2nd relay control.
RELAY_PIN3	-1	-1:Disabled 13:APM2 A9 pin 47-APM1 relay 50:Pixhawk AUXOUT1 51:Pixhaw...	Digital pin number for 3rd relay control.
RELAY_PIN4	-1	-1:Disabled 13:APM2 A9 pin 47-APM1 relay 50:Pixhawk AUXOUT1 51:Pixhaw...	Digital pin number for 4th relay control.

Figura 12: Ejemplo de cambio de parámetro para salida digital tipo *relay*

## D. Parámetros para control RF y a través de red celular

El hardware y firmware del Pixhawk junto a Ardupilot cuentan con la facilidad de conexión de ciertos receptores de RF haciendo referencia principalmente al sistema de receptores satélite de la marca **spektrum**. Esta marca cuenta con un protocolo de comunicación propietario llamado DSMX, y por ser de aplicación común, el Pixhawk cuenta con una entrada dedicada a este tipo de receptores, por lo que la conexión desde el control remoto (de la misma marca) hacia el receptor satélite se simplifica grandemente teniendo solo que realizar la conexión eléctrica del receptor en el puerto dedicado y enlazar el control con el receptor por medio del botón en la pantalla de SETUP y pestaña de *Radio Calibration*.

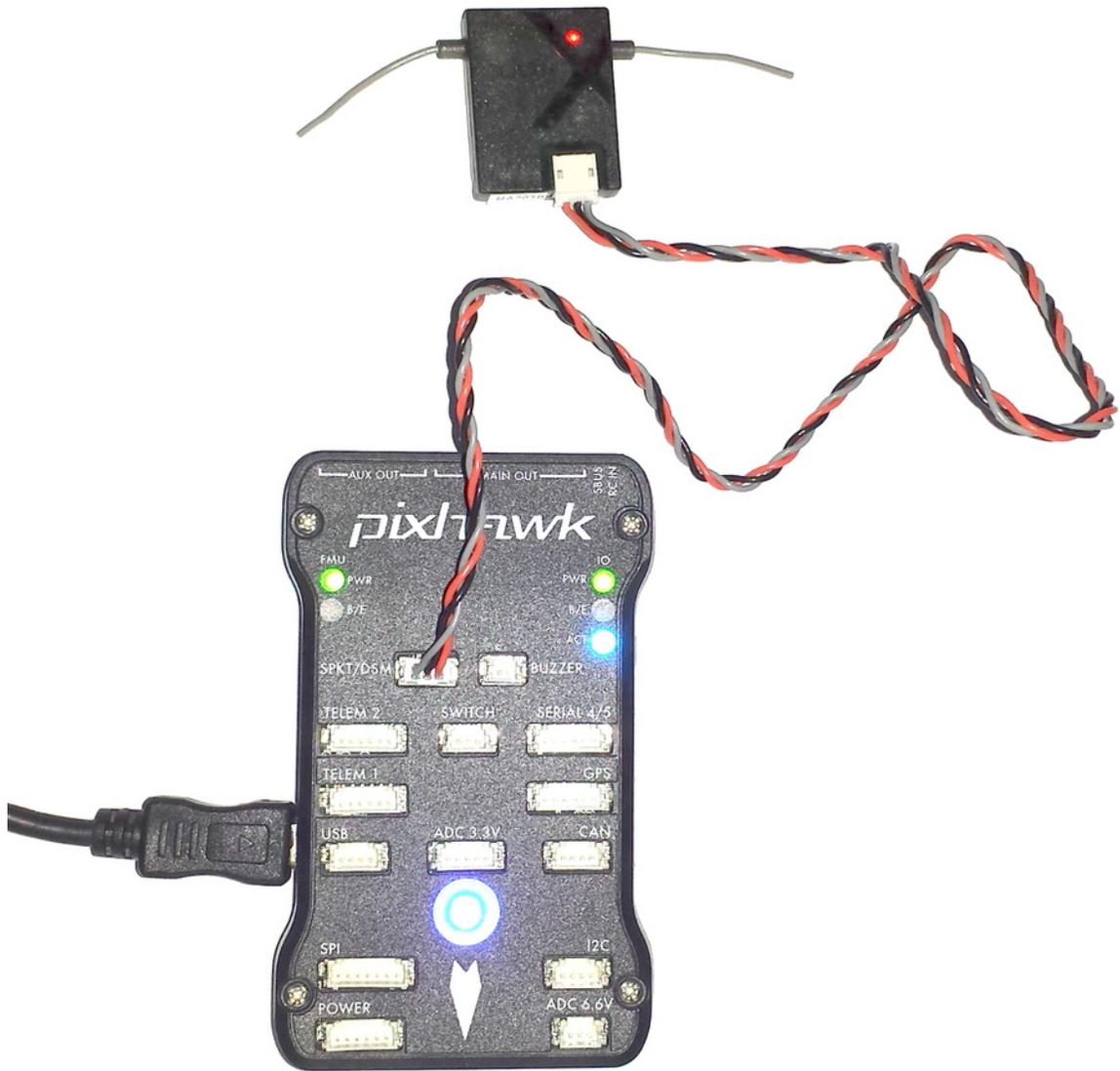


Figura 13: Receptor satélite DSMX conectado a Pixhawk con puerto dedicado



Figura 14: Pestaña de prueba de salidas y enlazado de control RF

## E. Parámetros para comunicación serial

El Pixhawk, por defecto, no cuenta con comunicación serial activa. Para que este pueda transmitir y recibir información a través del puerto serial 2 (se utiliza el dos por conveniencia) se deben ajustar tres parámetros dentro del firmware:

$$\text{SERIAL2\_PROTOCOL} = 2$$

Este parámetro permite habilitar la comunicación con protocolo MAVlink a través del puerto serial 2.

$$\text{SERIAL2\_BAUD} = 921$$

Este parámetro ajusta el baudaje, o velocidad de comunicación, entre el Pixhawk y Raspberry. El valor generalmente es de 921600 baudios.

$$\text{LOG\_BACKEND\_TYPE} = 1$$

Este parámetro genera una codificación para transmitir la información. Es un bitmap para ajustar el tipo de *loggers* a utilizar.

Command	Value	Unit	Range	Description
RSSI_TYPE	0		0:Disabled 1:An	Radio Receiver RSSI type. If your radio receiver supports RSSI of some kind, set it here, then set its associated RSSI_XXXXX parameters, if any.
RST_SWITCH_CH	0			RC channel to use to reset to last flight mode after go/no-go takeover.
RTL_SPEED	0	m/s	0 100	Return-to-Launch speed default. If zero use WP_SPEED or CRUISE_SPEED.
SAIL_ENABLE	0		0:Disabled 1:Ena.	This enables Sailboat functionality
■ SCHED				
■ SERIAL				
■ SERIAL0				
■ SERIAL1				
■ SERIAL2				
SERIAL2_BAUD	921		1:1200 2:2400.	The baud rate of the Telem2 port. Most stm32-based boards can support rates of up to 1500. If you setup a rate you cannot support and then can't connect to your board you should load a firmware from a different vehicle type. That will reset all your parameters to defaults.
SERIAL2_OPTIONS	0			Control over UART options. The InvertRX option controls invert of the receive pin. The InvertTX option controls invert of the transmit pin. The HalfDuplex option controls half-duplex (onewire) mode, where both transmit and receive is done on the transmit wire.
SERIAL2_PROTOCOL	1		-1:None 1:MAV.	Control what protocol to use on the Telem2 port. Note that the Frsky options require external converter hardware. See the wiki for details.
■ SERIAL3				

Figura 15: Parámetros de comunicación serial

## F. GPS

Comúnmente, los GPS utilizados con auto pilotos Pixhawk traen un compás interno que debe alinearse con la parte frontal del vehículo, y este se encuentra en un loop redundante con el compás interno del Pixhawk. Por lo mismo que la unidad de GPS cuenta con el chip de posicionamiento global y un compás, y ambas se comunican por distintos protocolos, el GPS debe conectarse por medio de dos puertos: uno serial (el puerto dedicado llamado GPS) y uno de comunicación I2C (el único de la placa).

Luego de conectado el GPS, este ya se comunica sin necesidad de cambiar parámetros ya que el puerto dedicado está habilitado por default para recibir cualquiera de los protocolos de GPS (NMEA, Ublox, Galileo, etc) a un baudaje común. El compás, sin embargo, debe calibrarse si es requerido.



Figura 16: Visualización de conectores y puertos de entrada de GPS

## G. Control de Rover por medio de Internet

Parte de los objetivos es crear un sistema de control remoto por medio de la red aparte del control remoto por RF, y este se obtiene utilizando la función de Joystick en Mission Planner. Esta función permite configurar un control reconocido como Joystick en la interfaz de Windows (cualquier control de consolas, como PlayStation, Xbox, Nintendo o incluso controles genéricos) para cumplir las mismas funciones que el control RF pero transmitiendo la información a través de la telemetría MavLink, que en este caso será por medio de la red de Internet proporcionada por un módem.

El control utilizado es un control de PlayStation 4 conectado por USB a la computadora corriendo Mission Planner. Este control tiene los requerimientos necesarios para las funciones necesarias, como armar, desarmar, cambiar entre modos de manejo (automático y manual) y los movimientos hacia adelante/atrás e izquierda/derecha.



Figura 17: Control de Joystick para canal de Throttle (acelerador)



Figura 18: Control de Joystick para canal de Roll (giro)

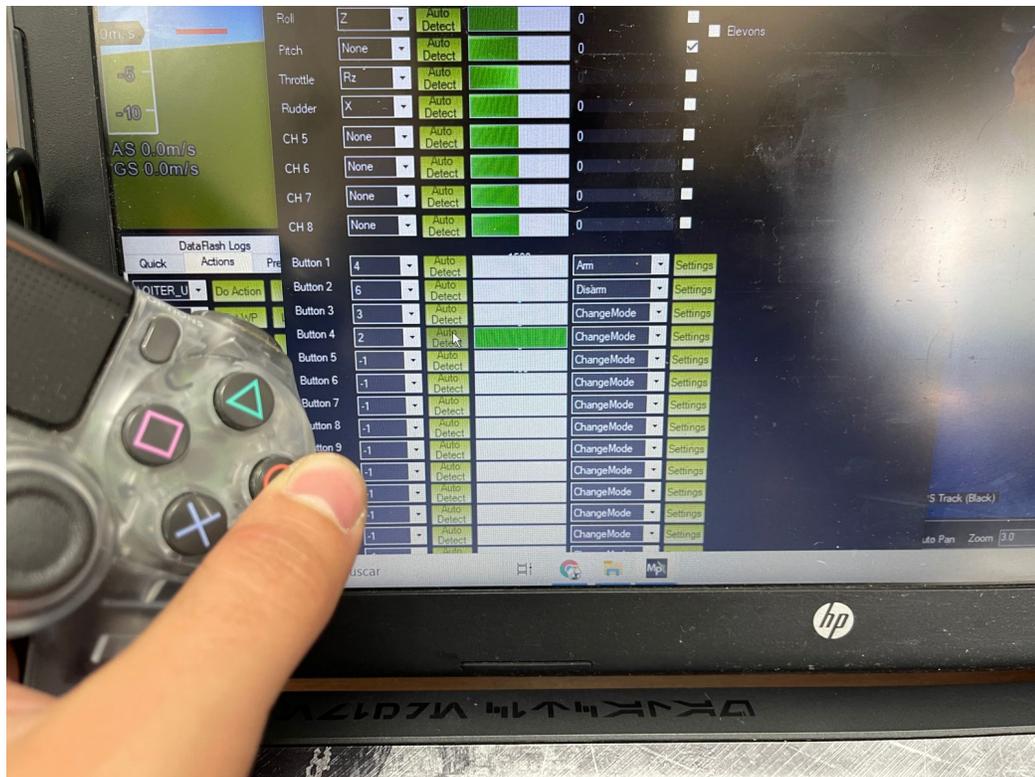


Figura 19: Control de Joystick para canal de cambio de modo



---

### Conexión de Pixhawk a Raspberry Pi

---

Como mencionado anteriormente, se utiliza una computadora de compañía Raspberry Pi 3 la cual se encarga de recibir la telemetría en protocolo MAVlink del Pixhawk y mandarla a través de internet. Esta debe ser configurada y conectada correctamente.

#### A. Conexión eléctrica

Una de las bases de la comunicación serial se encuentra en que necesita, como mínimo, tres cables: Tierra (común), Receptor (Rx) y Transmisor (Tx).

Para que dos dispositivos puedan transmitir y recibir datos por comunicación serial, ambos deben contar con terminales de Rx y Tx, las cuales se conectan inversas a ellas mismas. Entiéndase, el Rx del primer componente se conecta al Tx del segundo componente, y el Tx del primer componente se conecta al Rx del segundo componente. No es diferente el caso del Pixhawk y la Raspberry, las cuales deben conectarse como mencionado anteriormente, y la figura siguiente permite visualizarlo:

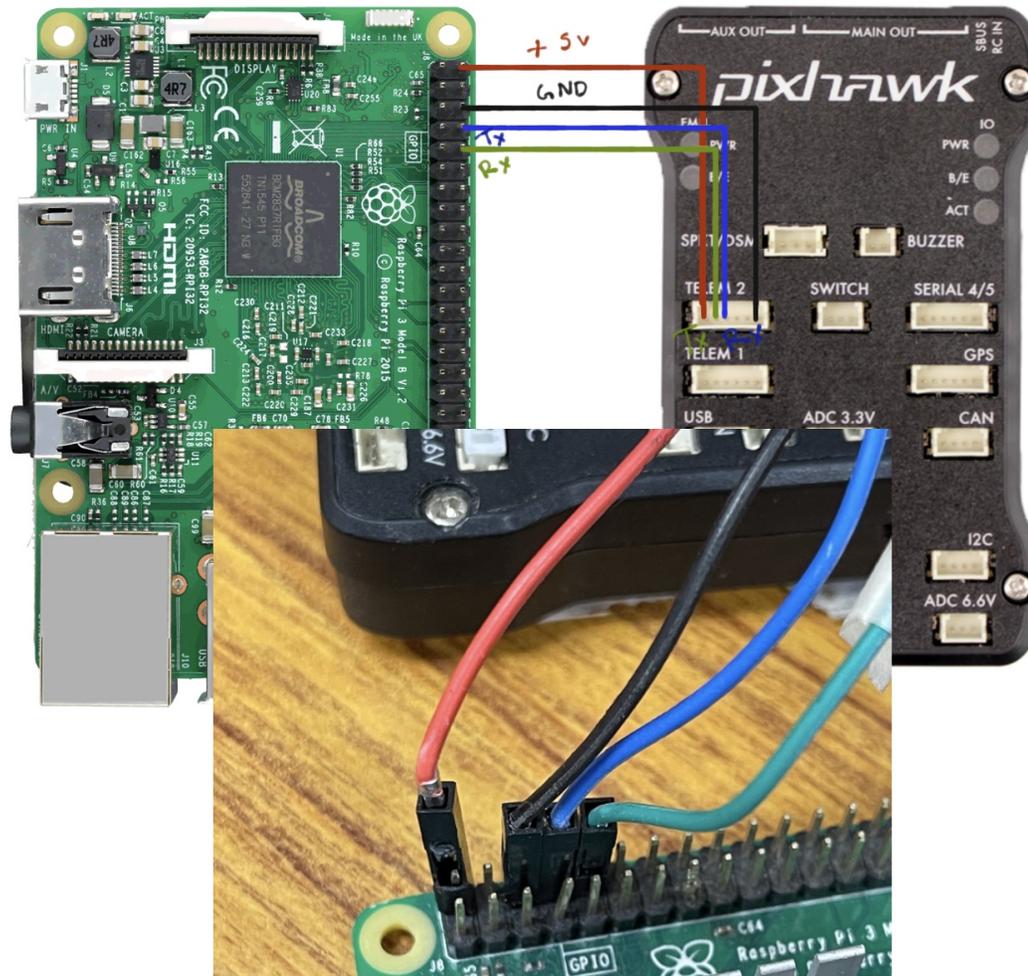


Figura 20: Diagrama de conexión entre Pixhawk y Raspberry, y conexión física

## B. Transmisión de datos

Como mencionado anteriormente, la computadora de compañía Raspberry se comunica por medio de serial con el Pixhawk, y para obtener dicha comunicación es necesario configurar tanto el Pixhawk como la Raspberry. Cabe mencionar que existen diferentes sistemas operativos para Raspberry, siendo el original y más común, Raspbian. En el caso final del proyecto se utilizó este sistema operativo, sin embargo se experimentó con otros sistemas operativos como APSSync, el cual simplifica un poco la configuración de la Raspberry, pero simplifica y limita las capacidades de la Raspberry al no contar con una pantalla gráfica. La configuración del Pixhawk fue realizada en el capítulo VII, y para el caso de la Raspberry utilizando el sistema operativo Raspbian se necesitan cambiar ciertas configuraciones. Para controlar la Raspberry se necesita un monitor, teclado y mouse externo que se conectan a la misma. Se asume que si se está trabajando con Raspberry esto no debe ser explicado debido

a la experiencia con la misma.

## 1. Configuración para comunicación serial

El primer paso es entrar a la configuración de la placa, y esto se realiza por medio de la consola de comandos. Una vez abierta la consola, se debe escribir el comando:

```
sudo raspi-config
```

Luego de ejecutar la línea anterior, se accede a la pantalla de configuraciones, donde se encuentran diversas opciones. En esta pantalla se debe seleccionar la opción de *Interfacing Options*, la cual nos lleva a la página donde se permite generar cambios en la interface de entradas y salidas de la Raspberry.

Luego, ya que se está utilizando la cámara de la Raspberry, se accede a la opción de *Camera*. Dentro de la opción de cámara, se selecciona **Enable**, y luego **Ok**. Luego, se regresa al menú de *Interfacing Options* y se selecciona **Serial**. Inmediatamente después se presentará un texto preguntando si se quiere habilitar un login de shell a través de serial (“*Would you like a login shell to be accessible over serial?*”) y se debe seleccionar **No**. Luego, preguntará si se quiere habilitar el hardware de puerto serial (“*Would you like the serial port hardware to be enabled?*”), y se selecciona **Yes**.

Después de regresar al menú de *Interfacing Options* se debe salir del menú y reiniciar la Raspberry para que los cambios sean validados y ejecutados.

Con esta configuración realizada, ya se puede utilizar la Raspberry conectada por medio de serial al Pixhawk, y la misma Raspberry se encarga automáticamente de unir las señales de telemetría MAVlink provenientes del Pixhawk con la señal de vídeo para transmitirla por internet.



#### A. Pruebas iniciales de estructura y motores

Durante el semestre se realizaron diversas pruebas, sin embargo para llegar a ese punto se requirió hacerle un servicio general a la estructura del Rover. Este servicio incluyó principalmente limpieza, ajuste de tensión de las orugas y lubricación del sistema de sprockets y cadenas. Se modificaron levemente las bases de los motores para colocar unos motores encontrados en la universidad y se trabajó en un diseño para modernizar visiblemente el Rover.

Luego de los trabajos iniciales, se conectó el Pixhawk con el Driver de los motores y, utilizando el control remoto por RF, se controló el Rover para realizar pruebas de campo de motores (aún con los motores viejos) y orugas. En la figura a continuación se observa el control remoto por RF utilizado, junto al driver, que está siendo controlado el Pixhawk (no en la imagen) para probar las salidas de este hacia el driver.



Figura 21: Driver accionando un motor DC a través de señales realizadas en el control RF

Luego de comprobar la integración del driver con el Pixhawk, se colocó este sistema en el Rover para hacer pruebas de campo sin la computadora de compañía. Esto requirió el uso de baterías Lipo de menor voltaje al planeado (de 14.8v nominales) para prueba de concepto. Con esto, el Rover ya fue manejable por control remoto.



Figura 22: El Rover subiendo una grada de ingreso a una casa



Figura 23: El Rover subiendo una banqueta para probar la estabilidad

Existe más documentación de pruebas realizadas en forma de vídeos, por lo que no se colocan en este documento, pero se realizará un video final incluyendo dichas pruebas.

Durante estas pruebas iniciales se concluyó que los motores eran bastante pesados, ineficientes y lentos, por lo que se decidió colocar unos motores más pequeños pero más rápidos y eficientes.



Figura 24: Motor actualizado

El proceso de selección de motores y cálculos de potencia se detalla en el trabajo de graduación titulado "Diseño de estructura Mecánica, Selección de Motores e Implementación de Sensores para un Robot Explorador Modular".

## **B. Integración de computadora de compañía a sistema de control remoto**

La computadora de compañía Raspberry Pi 3 toma un rol importante en el Rover ya que esta es la encargada de transmitir y recibir telemetría a través de la red, y sin esta, se estaría limitado al rango de control remoto por RF, el cual no es ilimitado o tan grande como un control remoto por señal celular.

Para las pruebas de concepto y pruebas generales se utilizó una Raspberry Pi 3 con el sistema operativo de APSync, el cual es un OS dedicado a funcionar como computadora de compañía y facilita la configuración y el funcionamiento del envío de datos por Wifi. Con este sistema operativo, y las conexiones realizadas anteriormente por serial, se obtuvo un envío de datos por wifi exitoso, pero con la desventaja que esta señal wifi es generada por el módulo de la Raspberry, por lo que es de corto alcance. A pesar de la desventaja mencionada, el envío de datos a través de wifi y la capacidad de conectarse al Pixhawk de manera inalámbrica y ya no a través de un cable presentó un avance significativo en el proyecto.

Utilizar APSync con la transmisión de la red Wifi local por medio del propio módulo de la Raspberry generaba una limitación enorme mencionada anteriormente, la cual era el rango de transmisión de datos. Dicho rango no supera los 50 metros y no se contaba con ningún software de transmisión de datos a través de la red celular. Esta limitación se trató de eliminar utilizando la plataforma LogMeIn Hamachi, la cual es una VPN (Virtual Private Network, por sus siglas en inglés) que, configurada correctamente, puede simular una conexión o red local aunque los dispositivos se encuentren conectados en distintas redes, y esto último es clave ya que el sistema operativo de APSync transmite telemetría a través de una red local y Mission Planner recibe telemetría a través de una red local, por lo que lograr simular dicha red local utilizando red celular elimina la limitación de rango y facilita grandemente la conexión.

A la computadora de compañía se le instaló Hamachi y se realizaron las configuraciones necesarias, sin embargo se encontraron problemas con los puertos de transmisión dentro de la red junto a problemas de envío de información a través de varios firewalls de algunas redes y de algunas computadoras, por lo que el sistema no fue práctico ni funcional en su totalidad. La idea de utilizar APSync por su facilidad de transmisión de telemetría se cambió por encontrar una solución que permitiera el envío de telemetría a través de la red celular, y la solución (resumidamente) consta de utilizar una Raspberry con el sistema operativo Raspbian (no dedicado a computadoras de compañía) y manualmente recibir la telemetría y enviarla a través de Hamachi utilizando scripts y otros programas. Esta solución se detalla en el trabajo de graduación titulado "Diseño e implementación de comunicación inalámbrica y creación de backup de datos para un robot explorador modular".

Utilizar otro método para enviar y recibir telemetría deja una incógnita: ¿qué pasa con la

cámara? Ya que el sistema operativo de APSTync se encargaba de unir la telemetría MAVLink con el *Video Feed* automáticamente, se debió encontrar otra solución para el envío de vídeo desde la Raspberry, a través de Hamachi, para una computadora, y esto se resolvió estudiando a profundidad el sistema de envío de vídeo de APSTync, el cual funciona con un programa llamado "GStreamer" que transmite el vídeo a través de *Pipelines* en puertos especificados en el script. Este programa, resumidamente, crea un túnel directo de comunicación entre redes no necesariamente locales, sin embargo, que sean locales facilita el proceso teniendo que definir únicamente el puerto donde se transmitirá la información. En la Raspberry con sistema operativo Raspbian se instala el programa **GStreamer** y se corre el código siguiente, que permite configurar el túnel con codificación de video y puerto definido:

```
raspivid -t 0 -w 1280 -h 720 -fps 30 -b 1700000 -o - | gst-launch-1.0 -v fdsrc
! h264parse config-interval=1 ! rtph264pay ! udpsink host = 192.168.1.1 port=
9000
```

Este proceso es fácilmente replicable utilizando scripts ubicados en el directorio de boot de la Raspberry para que corran automáticamente cada vez que se conecta la Raspberry, evitando tener que conectarla a un monitor y teclado cada vez que se quiere utilizar el Rover.

Para recibir el video en la computadora, se pueden seguir dos rutas: utilizar el programa GStreamer para generar una ventana emergente con el video recibido o, a través de Plug-ins de Mission Planner, colocar el vídeo en la pantalla de inicio de Mission Planner.

Luego de discutirlo con el grupo de trabajo, se decidió utilizar la vía de colocar el vídeo en una ventana emergente debido a la naturaleza del programa de Mission Planner, el cual requiere una cantidad considerable de poder computacional y colocar vídeo sería podría sobrecargar el programa.

Para esto se debe instalar el programa GStreamer en la computadora desde el link siguiente:

```
https://gstreamer.freedesktop.org/download/
```

Y seleccionar la opción de 32 bits, ya que la opción de 64 bits presenta fallas en varias ocasiones según documentación visitada.

Ya instalado GStreamer, se debe abrir la ventana de comandos de Windows y correr la siguiente línea:

```
c:\gstreamer\1.0\msvc_x86\bin\gst-launch-1.0 udpsrc port=9000 !
application/x-rtp,encoding-name=H264,payload=96 ! rtph264depay ! avdec_h264 !
videoconvert ! autovideosink
```

La cual genera la ventana emergente mostrada a continuación, con la transmisión de vídeo en vivo:

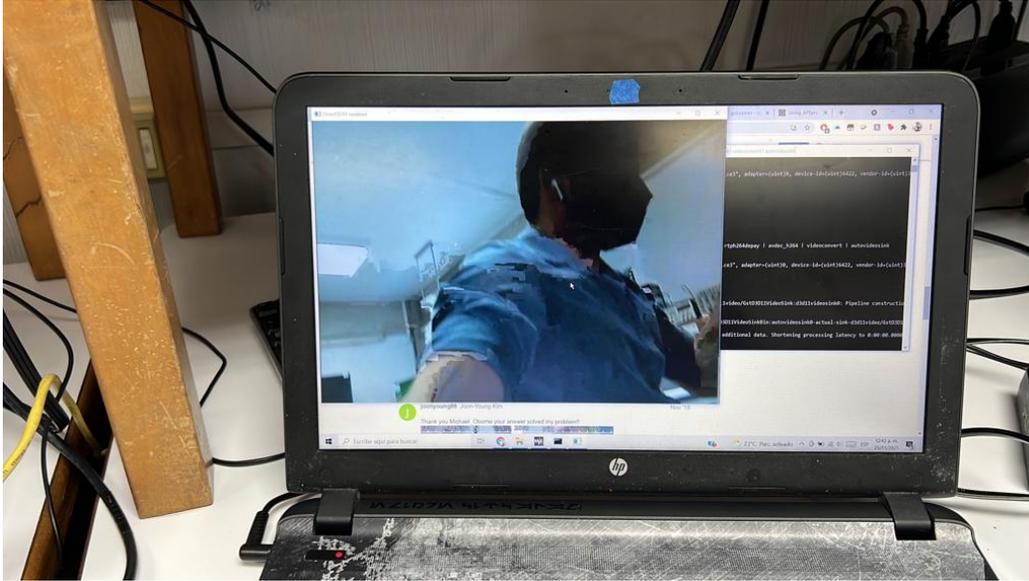


Figura 25: Ventana emergente de GStreamer con el video proveniente de la Raspberry

La integración de todo el sistema con la estructura original del Rover utilizada para pruebas de campo se visualiza de la siguiente manera:

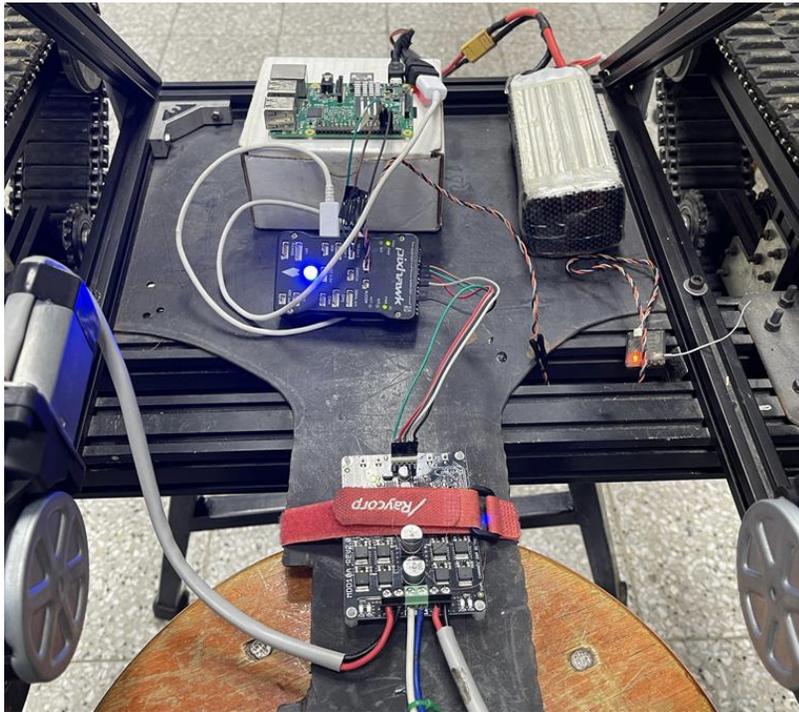


Figura 26: Electrónicos completos en Rover antes de realizar pruebas de campo

## C. Manejo de Rover a través de la red utilizando Joystick y control por RF

El control del Rover a través de un Joystick fue probado con el sistema operativo de APISync en la computadora de compañía, y esta función es propia de Mission Planner dependiendo puramente de la comunicación serial (telemetría MAVlink) entre este y el Pixhawk, por lo que esta funcionalidad no se ve afectada utilizando cualquier otro sistema de transmisión de telemetría mientras efectivamente exista la comunicación y conexión directa.

En estas pruebas de campo se tenía el control RF y el Joystick conectado a la computadora para probar la funcionalidad redundante. Se empezaron las pruebas utilizando el control RF y, en momentos aleatorios, se apagaba dicho control para empezar un control a través de Wifi, el cual es igual al control por RF. La limitante aquí, nuevamente, fue el rango de la señal Wifi, sin embargo, el control por Wifi es total.

En las figuras siguientes se observa una prueba de campo utilizando el Joystick como control principal para el rover. Este se configuró como explicado en la sección G del capítulo VII.



Figura 27: Control de Rover con Joystick

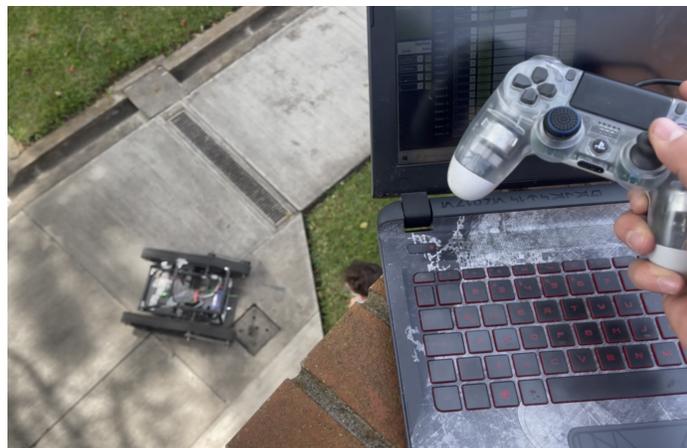


Figura 28: Control de Rover con Joystick



Figura 29: Prueba de adaptación de terreno

- La integración de la plataforma reutilizada facilitó el proceso de fabricación del Rover realizándole solamente modificaciones a esta para obtener reducciones de peso y modernizarlo visualmente.
- La utilización de componentes modernos para el control del rover y manejo de motores permite realizar muchas más funciones, como control remoto a través de internet, sin necesidad de módulos grandes e ineficientes.
- La utilización de baterías de Polímero de Litio (LiPo) reduce significativamente el peso comparado a baterías de Plomo (Pb) y aumenta la autonomía del mismo por su densidad energética mayor.
- La implementación de la Raspberry Pi 3 como computadora de compañía permitió realizar el control remoto por internet así como recibir telemetría importante a distancia.
- La utilización de la plataforma LogMeIn Hamachi para la transmisión de datos a través de internet simplificó el proceso de la comunicación a través de un servidor.



---

### Recomendaciones

---

- La plataforma de Rover existente fue útil, sin embargo, es bastante pesada y cuenta con una reducción de velocidad por el sistema de engranes y cadena muy grande, por lo que una plataforma más ligera y simple sería ideal al momento de realizar un proyecto parecido.
- Aparte del control por internet y RF, existe transmisión de telemetría por medio de módulos especiales a frecuencias de 433MHz. Estos, si se tratara de alguna aeronave y no vehículo terrestre, serían de mayor utilidad ya que el rango es mayor a 10km y son ligeros, por lo que, si en alguna aplicación de Rover, no se perdiera línea de visión, los módulos de telemetría dedicados de 433MHz son una opción viable para reducir latencia y peso.
- Los otros sistemas operativos disponibles para computadoras de compañía dedicadas a Pixhawk incluyen ciertas ventajas para aplicaciones más específicas, como transmisión de variables de inclinación o velocidad en formatos distintos para automatizar totalmente el manejo o vuelo de los vehículos. En caso del Rover, el sistema operativo de Raspbian es suficiente para la aplicación.



- 
- [1] LiteTrax. (). “WHEELS VS TRACKS: ADVANTAGES AND DISADVANTAGES,” dirección: <https://litetrax.com/wheels-vs-tracks-advantages-disadvantages/>. (visitado: 19.05.2021).
- [2] H. Willee. (). “Pixhawk 4,” dirección: [https://docs.px4.io/master/en/flight\\_controller/pixhawk4.html%22](https://docs.px4.io/master/en/flight_controller/pixhawk4.html%22). (visitado: 17.05.2021).
- [3] R. Beamond, “INVENTION OF THE CATERPILLAR TRACK,” 2014. dirección: <http://www.users.waitrose.com/~rogerbeamond/INVENTION.pdf>.
- [4] “Advanced Space Transportation Program: Paving the Highway to Space,” 2008. dirección: <https://www.nasa.gov/centers/marshall/news/background/facts/astp.html#:~:text=Today%5C%2C%5C%20it%5C%20costs%5C%20%5C%2410%5C%2C000%5C%20to,per%5C%20pound%5C%20within%5C%2040%5C%20years..>
- [5] R. inc. (). “Cytron 10A 5-30V Dual Channel DC Motor Driver,” dirección: <https://www.robotshop.com/nl/en/cytron-10a-5-30v-dual-channel-dc-motor-driver.html>. (visitado: 10.05.2021).
- [6] D. Windestal. (). “FPV Starting Guide,” dirección: <https://rcexplorer.se/educational/2009/09/fpv-starting-guide/>. (visitado: 22.06.2021).
- [7] K. CORPORATION. (). “¿Qué es un sensor ultrasónico?” Dirección: <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/>. (visitado: 20.06.2021).
- [8] R. P. FOUNDATION. (). “Raspberry Pi 3 Model B,” dirección: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. (visitado: 15.05.2021).
- [9] A. D. Team. (). “Mission Planner Overview,” dirección: <https://ardupilot.org/planner/docs/mission-planner-overview.html>. (visitado: 19.05.2021).



## A. Guías externas

Guía de conexión de Pixhawk con Raspberry:

<https://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html>

Guía de instalación de Mission Planner en computadoras:

<https://ardupilot.org/planner/docs/mission-planner-installation.html>

Guía de instalación de firmware en Pixhawk utilizando Mission Planner:

<https://ardupilot.org/planner/docs/common-loading-firmware-onto-pixhawk.html>

## B. Pruebas realizadas en video

Video demostrativo del funcionamiento completo del Rover

<https://youtu.be/VG1vKceXbGI>

