

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Grado

**EVALUACIÓN DE LAS COMUNICACIONES
INALÁMBRICAS LORA PARA ENLACES DE
TELEMETRÍA EN DRONES**
**(Evaluation of LoRa Wireless communications
for drone telemetry links)**

Para acceder al Título de

*Graduado en
Ingeniería de Tecnologías de Telecomunicación*

Autor: María Josefa Mendivil Abascal

Septiembre - 2018

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: María Josefa Mendivil Abascal

Director del TFG: Adolfo Cobo García

Título: "Evaluación de las comunicaciones inalámbricas LoRa para enlaces de telemetría en drones"

Title: "Evaluation of LoRa wireless communications for drone telemetry links"

Presentado a examen el día: 19/09/2018

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): Adolfo Cobo García

Secretario (Apellidos, Nombre): M^a Angeles Quintela Incera

Vocal (Apellidos, Nombre): Mario Mañana Canteli

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

ÍNDICE

RESUMEN	1
1. INTRODUCCIÓN	2
2. COMUNICACIONES CON DRONES	4
2.1 Qué es un drone	4
2.2 Partes en las que se compone un drone	4
2.3 Métodos de vuelo	6
2.4 Marco normativo para el uso de drones	7
2.5 FPV	9
2.6 Parámetros de vuelo a tiempo real: Definiciones	12
2.7 Entornos de desarrollo más utilizados	12
2.7.1 CARMEN	12
2.7.2 ROS	13
2.7.3 Entornos personalizados	13
2.8 Software empleado para la configuración y manejo de drones	13
2.9 Protocolo MAVLINK	14
2.9.1 MavLink para transmisión de imagen en streaming	15
2.9.2 MavLink para transmisión de vídeo en streaming	17
2.10 Arducopter	18
2.11 Sense and avoid	19
2.12 LRS o LORAN	21
2.13 ELORAN	22
2.14 XLRS	23
3. ESTÁNDAR DE TRANSMISIÓN LORA	24
3.1 LoRa	24

3.2	Arquitectura del sistema LORAWAN	25
3.3	Clases de dispositivos	26
3.4	Tasa de datos.....	27
3.5	Seguridad	27
4.1	Qué es IDE Arduino	28
4.2	Características de las placas.....	29
4.3	Programas utilizados.....	29
4.3.1	Transmisor.....	29
4.3.2	Receptor.....	32
4.4	Resultados obtenidos	36
5.	CONCLUSIONES Y LÍNEAS FUTURAS	39
6.	BIBLIOGRAFÍA	40

ÍNDICE DE FIGURAS

Figura 1.	Partes en las que se compone un drone	6
Figura 2.	Esquema del marco normativo.....	9
Figura 3.	Estructura de paquete MAVLink	15
Figura 4.	MAVLink para transmisión de imágenes en Streaming	16
Figura 5.	MAVLink para transmisión de video en streaming	17
Figura 6.	Diferencias entre el Modelo OSI y el sistema LoRaWAN	25
Figura 7.	Arquitectura del sistema LoRaWAN	25
Figura 8.	Clases de dispositivos	26
Figura 9.	Código transmisor primera parte.....	30
Figura 10.	Código transmisor segunda parte	30
Figura 11.	Código transmisor tercera parte	31

Figura 12. Inicialización y envío de paquetes desde la placa transmisora	32
Figura 13. Código receptor primera parte	32
Figura 14. Código receptor segunda parte.....	33
Figura 15. Código receptor tercera parte	34
Figura 16. Código receptor cuarta parte	35
Figura 17. Inicialización y recepción de paquetes de la placa receptora.....	36
Figura 18. Mapa de pruebas	36
Figura 19. Recogida de datos	38
Figura 20. Paquetes recibidos en función de la distancia	38

RESUMEN

Actualmente las comunicaciones se encuentran en un periodo de constante evolución, por tanto la investigación de nuevas formas de comunicación se convierte en algo imprescindible. Concretamente las comunicaciones inalámbricas se encuentran cada vez más limitadas por las frecuencias de uso y las distancias de alcance entre transmisor y receptor, en este trabajo se estudiará la comunicación entre el mando de control y el drone mediante la utilización de un estándar de transmisión radio de largo alcance denominado LoRa.

1. INTRODUCCIÓN

En el mundo de los drones existen dos causas principales que dificultan sus comunicaciones, la elección de frecuencia y ancho de banda, y el alcance.

La elección de frecuencia se convierte en una tarea compleja pero necesaria para la comunicación entre transmisor y receptor. La frecuencia que se estableció usar en un principio fue la de 2.4GHz compartiéndola con las comunicaciones móviles, debido a la saturación en esta frecuencia existen grandes interferencias que pueden fácilmente llegar a echar abajo las comunicaciones entre el drone y el mando de control. Por este motivo se ha ido estableciendo el uso de nuevas frecuencias y ahora el usuario tiene la opción de escoger entre una pequeña variedad. Cada frecuencia tiene un número limitado de canales, esto permite el uso de la misma frecuencia a varias comunicaciones simultáneamente sin que se solapen las señales. Las frecuencias mayores ofrecen mayor calidad de imagen y mayor ancho de banda.

Hay que tener en cuenta que el uso de una frecuencia en concreto influye considerablemente en la penetración de la señal y su alcance, este último además se ve afectado por muchas otras causas, como son: el tipo de antena utilizada (a menor frecuencia se necesitará una antena de mayor dimensión), el lugar donde se realice el vuelo (ya que hay una diferencia notoria realizando el vuelo en espacio abierto o en ciudad), las condiciones meteorológicas, y su autonomía (tiempo de batería).

En este trabajo se va a evaluar el uso de LoRa para la telemetría, en concreto para medir el alcance a través del uso de dos placas LoRa ESP32 con chip SX1278, una transmisora y una receptora.

LoRa es un estándar de transmisión radio y algunas de las ventajas más llamativas que proporciona a las comunicaciones son: la alta tolerancia a interferencias y su largo alcance, es decir, con su uso podrían solventarse algunos de los problemas mencionados anteriormente.

Este trabajo se divide en tres grandes partes, la primera es una introducción al mundo de los drones, una visión general de qué es un drone y las partes en que se compone, el software y los protocolos que se utilizan, la telemetría y el marco normativo sobre el que

se exponen. En la segunda se define el estándar LoRa, en el que se basan las placas que se han utilizado para el desarrollo. Por último una tercera parte que se enfoca en la experimentación, la transmisión y recepción de paquetes a través de dos placas ESP 32, la documentación de los programas que se están utilizando para el desarrollo y los resultados obtenidos. Para finalizar el trabajo se desarrollan las conclusiones y líneas futuras.

2. COMUNICACIONES CON DRONES

2.1 Qué es un drone

Un vehículo aéreo no tripulado (UAV) o drone es una aeronave que vuela sin tripulación. Pueden distinguirse tres tipos según su uso; para aplicaciones civiles, militares (vehículo aéreo de combate no tripulado UCAV) o no profesional. Una definición más exacta de UAV es un vehículo sin tripulación reutilizable, capaz de mantener un nivel de vuelo de forma autónoma gracias a un motor de expulsión o reacción. Se distinguen de los de uso militar en que estos últimos no se pueden reutilizar puesto que se utilizan como arma en forma de misil.

Existe una gran variedad de formas, tamaños, configuraciones y características de diseño. Pueden ser controlados desde una ubicación remota (más correctamente llamadas Aeronaves Radiocontroladas o Aeronaves R/C), o por el contrario, volar de forma autónoma con un plan vuelo pre-programado (serán capaces de despegar, volar y aterrizar automáticamente).

Las tres empresas más conocidas en este sector son Syma, DJI y Phantom.

En cuanto a los tipos de aeronaves podemos encontrar de ala fija, parecidos a una avioneta tradicional pero sin cabina, pueden tener uno o más motores de hélice, propulsión o multirotor, inspirados en helicópteros, constan de al menos al menos tres motores.

2.2 Partes en las que se compone un drone

- Chasis, estructura o frame: soporta al resto de los elementos, debe ser ligero y resistente.
- Placa controladora: recoge los datos de todo el sistema y se encarga de mantener la estabilidad de la aeronave, debe adaptarse a cada situación y tiene la capacidad de realizar vuelos programados sin necesidad de intervención humana. Se encarga de tomar el control en caso de pérdida del enlace de radio o bajo nivel de batería.

- GPS: es fundamental no solo para las grabaciones sino para la estabilidad de navegación y seguridad, esencial para el modo “vuelta a casa” que permite que en el caso de que surja algún problema la aeronave vuelva al punto del que partió registrando además en todo momento el punto exacto donde se encuentra la aeronave, además da la opción a rutas programadas por el piloto (desde la emisora de control se indica los puntos por los que debe pasar el drone) y a una posición de bloqueo (permite establecer una altura en la que la aeronave se mantendrá constante durante el vuelo). El sistema de posicionamiento puede servirse de más de una red lo que permite mejorar la precisión aunque siempre será de una tarea compleja reducir el margen de error a menos de unos 2 metros, la única opción en la que se podría reducir a unos 4cm sería utilizando tecnología RTK pero supone un aumento considerable en el precio.
- Gimbal: es el elemento móvil que une la cámara del drone y la estabiliza, la mantiene nivelada y la dirige por medio de control en tierra.
- Motores: son los que se encargan de transformar la energía eléctrica en movimiento circular para transmitirlo a la hélice y producir el empuje que permita la elevación y el desplazamiento de la aeronave, todo perfectamente equilibrado para conseguir autonomía en el vuelo.
- Emisora o control en tierra y receptor embarcado: Son los elementos que se encargan del control por parte del piloto, la emisora transmite las órdenes del piloto al receptor y éste a la placa controladora (433MHz, 2.4GHz, 5.8GHz).

Podemos observar las partes en las que se compone el drone en la Figura 1.

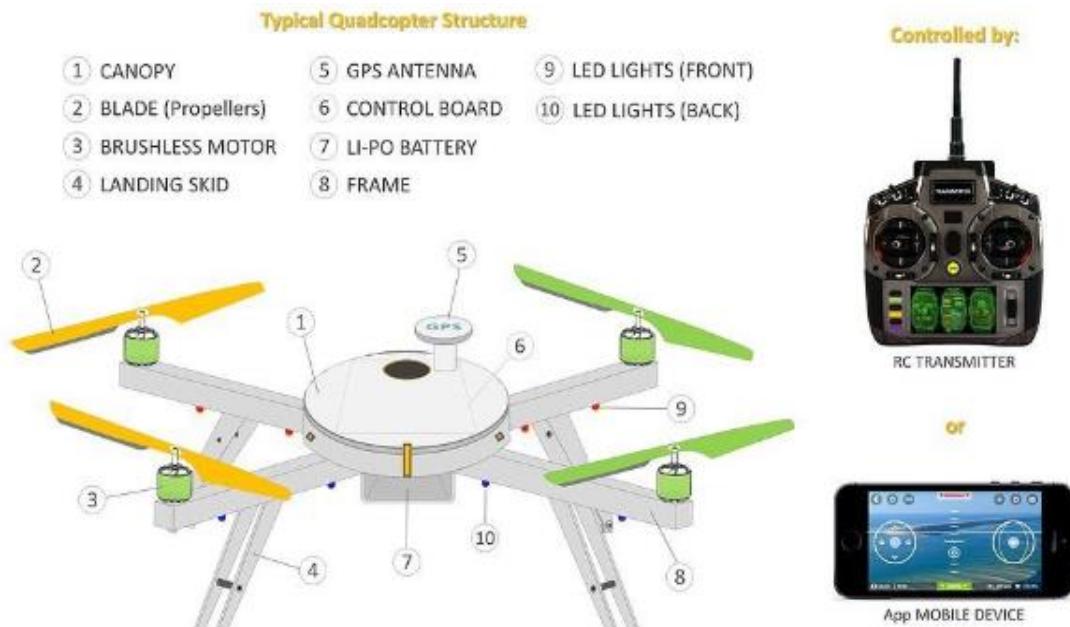


Figura 1. Partes en las que se compone un drone

2.3 Métodos de vuelo

Según su control:

- Control mediante una emisora: un mando con radio control manual que permite manejar el aparato visionándolo directamente desde la tierra.
- Control por tablet/smartphone: mediante el uso de aplicaciones.

Según su autonomía:

- Autónomo: el vehículo debe realizar su misión dentro del ámbito programado, con sólo un monitoreo desde tierra, incluye la operación automática completa con funciones autónomas y operación inteligente.
- Semiautónomo: el piloto realiza cambios y conduce la misión a través de una interfaz de administración del vuelo. Sin esta información el UAV realizará operaciones automáticas pre-programadas.
- No autónomo: La totalidad del control del vehículo se realiza remotamente.

2.4 Marco normativo para el uso de drones

Nos vamos a centrar en el uso de aeronaves de forma no profesional, debemos tener en cuenta una serie de factores para poder hacer uso de nuestro UAV de forma legal. Aunque en los últimos años el uso de drones se ha disparado, la legislación ha ido adaptándose paulatinamente, el último marco normativo se publicó el 29 de Diciembre de 2017.

Debemos volar el drone independientemente de su tamaño siempre teniendo en cuenta las condiciones meteorológicas y de día (excepto si la aeronave pesa menos de 2 kg ya que en este caso se permiten los vuelos nocturnos siempre que no se superen los 50 metros de altura además se requiere la autorización de AESA a través de la presentación de un estudio de seguridad específico), pero sobretodo hacer uso de zonas adecuadas donde se pueda volar con seguridad sin que exista riesgo para las personas, en zonas urbanas están prohibidos los vuelos excepto para aeronaves de menos de 250 gramos mientras no se superen los 20 m de altura. Se deben evitar vuelos cercanos a aeropuertos o zonas de espacio aéreo reservado (a una distancia mínima de 8 km). Los daños causados por el cuadricóptero siempre serán responsabilidad del piloto que lo maneja.

La normativa actual de drones solamente obliga a disponer de licencia de piloto a las personas que vayan a desempeñar una actividad profesional con drones. Estas tendrán que acreditar unos conocimientos teóricos y prácticos, además de contar con un certificado médico de clase LAPL (para drones de < 25KG) o de Clase II (para drones >25KG).

Cualquier aeronave pilotada por control remoto deberá de llevar una placa de identificación de la aeronave, en la que constará el nombre del fabricante, el tipo de drone, el modelo, el número de serie y por último el nombre del operador y sus datos de contacto.

Se distinguen tres tipos de drones: inferiores a 2 kg, de hasta 25 kg, y superiores a 25 kg. Los drones inferiores a 2 kg son los únicos que pueden volar sin restricciones más allá del alcance visual del piloto y alcanzar una altura máxima de 120 m (400 pies), aunque deben estar sujetos al alcance de la emisión por radio de la estación de control. Para

aeronaves que superen los 2 kg, los vuelos estarán permitidos siempre y cuando este cuente con sistemas que permitan detectar y evitar a otros usuarios del espacio aéreo y sean aprobados por AESA. Además, la aeronave deberá disponer de un dispositivo de visión orientado hacia delante y deberá contar con la autorización por parte de AESA basada en un estudio aeronáutico de seguridad.

No es necesario un curso o certificado por parte de la AESA, pero sí se debe emitir a los Servicios de Información Aeronáutica un aviso a los distintos usuarios que estén haciendo uso del espacio aéreo en el momento en el que se vaya a producir el vuelo.

Las naves de hasta 25 kg (también catalogadas como Clase LAPL) no necesitan un permiso para el pilotaje, pero el piloto tiene que hacer una declaración responsable a la AESA de su utilización además de una notificación cinco días antes de su vuelo. El drone debe estar dentro del alcance visual del piloto, a una distancia máxima de 500 m y a una altura sobre el terreno de 120 m. Para la obtención de fotografías o vídeos se debe cumplir la Ley de Protección de datos cuando haya terceras personas involucradas directa o indirectamente.

Los drones con un peso mayor de 25 kg (Clase 2) son los que necesitan unos permisos más exigentes. El piloto necesita un certificado básico o avanzado emitido por una organización de formación aprobada tras superar un curso, sumado a una autorización de la AESA para poder iniciar su actividad. Finalmente, necesita de un registro de Matrícula de Aeronaves de AESA y disponer de certificado de aeronavegabilidad.

Los pilotos deben estar siempre autorizados para poder volar estos aparatos. En recintos completamente cerrados (incluyendo el techo) no están sujetos a la jurisdicción de la AESA, por lo que los titulares de los recintos son los responsables de que los drones estén autorizados. En esta última publicación del marco normativo sobre drones, como se comentó anteriormente, se permiten los sobrevuelos en zonas urbanas y sobre aglomeración de personas cuando la masa máxima al despegue de la aeronave no exceda los 10 kg, la operación se debe de realizar dentro del alcance visual del piloto (VLOS), la zona a volar deberá de estar acordonada por la autoridad competente, o en otro caso, mantener una distancia horizontal mínima de 50 metros con edificios u otro tipo de estructuras además de mantener una distancia de 50 metros con personas ajenas a la operación, además la aeronave deberá de tener instalado un sistema de limitación de

energía del impacto (paracaídas, airbag...), y, por último, se deberá de contar con la autorización por parte de AESA a través de la realización de un estudio de seguridad aeronáutica.

Para habilitarse como operador de aeronaves deben cumplirse una serie de requisitos; ser mayor de edad, acreditar una licencia de piloto y un certificado médico de clase LPAL.

El marco normativo se recoge esquemáticamente en la Figura 2.



Figura 2. Esquema del marco normativo

2.5 FPV

First Person View o vista en primera persona que consiste en el vuelo con cámara abordo y transmisión en tiempo real de la imagen que la videocámara capta desde tierra. Se puede dotar de movimiento a la cámara, tanto en el eje vertical (Tilt) como en el horizontal (Pan) y puede controlarse mediante potenciómetros en la emisora o con dispositivos de detección de movimiento que moverán la cámara de acuerdo a los movimientos de nuestra cabeza.

El equipo FPV está compuesto por una cámara, un transmisor y un receptor de video con una antena y una pantalla o gafas para visualizar la imagen.

En cuanto al uso de frecuencias, tanto para el FPV como para la telemetría del drone en general, la situación actual es la siguiente:

- 35 MHz: rango excelente para el control con pocos fallos, no afecta a la señal de video. Usando la banda de 2.4 GHz para FPV consiguiendo una transmisión de video de alta calidad dentro de la legalidad.
- 900MHz: Ilegal en España, es utilizada para la red 3G de teléfonos móviles
- 1.2 GHz: Provee de una buena penetración frente a los obstáculos pero no es legal, está reservada para otros usos.
- 1.3GHz: En USA se utiliza para transmitir audio y video si se dispone de una licencia de canal de TV local o de radio. En España por el momento se usa para transmisión aérea así que tampoco es legal.
- 2.4GHz: Es la más apropiada para señales de radio aunque la habilidad de penetración no es especialmente buena, también es utilizada para muchos otros equipos como WiFi, Bluetooth y transmisores y receptores RC por lo que podemos encontrarnos con multitud de interferencias.
- 5.8GHz: Cada vez más popular, para la utilización de esta frecuencia se puede contar con dispositivos baratos y asequibles. El uso de esta banda es aconsejable ya que no interfiere con la de 2.4 GHz que es la más recurrida y en consecuencia donde se encuentran más interferencias. Otro aspecto positivo es que se pueden utilizar antenas circulares polarizadas muy pequeñas debido a la alta frecuencia, y se pueden resolver problemas de recepción debidos a la polarización y multipath. Una de las desventajas es que no tiene buen poder de penetración, por lo que los vuelos se limitan a donde alcance la vista (entre unos 500m y 1 Km). El transmisor de vídeo puede estar cerca de otros dispositivos sin que haya efectos o interferencias negativas. En esta banda contamos con 32 canales lo que supone una mayor capacidad y una mayor dificultad para que se

solapan las señales. Son las que más se acercan a la imagen real en cuanto al movimiento.

En cuanto a la potencia de emisión máxima legal en la banda de los 2.4 GHz es de 10 mW y a la de 5.8GHz es de 25mW, hay que tener en cuenta que cuanto mayor sea la potencia más lejos podremos volar nuestro UAV.

Es muy importante tener en cuenta los armónicos de frecuencia ya que pueden introducir nuevas interferencias (por ejemplo, la banda de 4.8GHz para 2.4GHz)

La frecuencia de radio debería ser siempre inferior a la de video, de esta forma si se crean interferencias afectarán al video y no a la radio, por lo que no perderemos la conexión con nuestro UAV, de lo contrario, usando una frecuencia de radio superior a la de video necesitaremos filtros en el transmisor para eliminar los armónicos.

A grandes rasgos podemos concluir deduciendo que cuanto menor sea la frecuencia empleada mayor será el alcance con la misma potencia de emisión, de la misma forma que habrá mayor penetración en los obstáculos, a su vez las antenas necesarias serán de mayor tamaño y el ancho de banda será menor igual que la calidad de imagen.

El receptor de vídeo se encarga de recibir inalámbricamente la señal de vídeo que se envía desde el transmisor. Normalmente en una misma banda se utilizan entre 8 y 32 canales, tanto mayor sea el número de canales más usuarios podrán utilizar la misma frecuencia, aunque siempre hay que dejar margen entre canales para evitar interferencias entre los pilotos.

Existe una clase de receptores de vídeo, “diversity”, que mejoran mucho la calidad de la señal de vídeo mediante el uso de dos antenas minimizando el efecto de cancelación de la propagación multirayecto y extendiendo el rango operativo de los sistemas inalámbricos. El mayor inconveniente es el precio.

Las antenas utilizadas están hechas específicamente para cada frecuencia, las que vienen de serie (antenas omnidireccionales de polarización lineal) no dan buenos resultados para la banda de 5.8GHz ya que crean interferencias de vídeo debido a la ya mencionada propagación multicamino y empeoran la calidad.

Hay cuatro grupos de antenas que mejoran las desventajas provocadas por las comentadas anteriormente, antenas de serie en la frecuencia de 5.8GHz, se trata de antenas de polarización lineal omnidireccional, antenas de polarización lineal direccional, antenas de polarización circular omnidireccional y antenas de polarización circular direccional.

Las más adecuadas para un uso no profesional son las antenas de polarización circular omnidireccional ya que tienen una gran ventaja sobre las demás, cuenta con la opción de moverse en todas direcciones sin perder la señal y sin la necesidad de un receptor diversity.

2.6 Parámetros de vuelo a tiempo real: Definiciones

- **Ground speed:** indicador que informa acerca de la velocidad del avión con respecto al suelo.
- **Air speed:** velocidad verdadera, o TAS, es la velocidad relativa de un avión respecto a la masa de aire en la que vuela. Unidad de medida, millas por hora (m.p.h.) o en nudos (kts).
- **Climb rate:** tasa de ascenso, es la velocidad a la que una aeronave incrementa su altitud.
- **Altitude:** altitud, es la distancia vertical de un punto de la superficie terrestre respecto al nivel del mar.

2.7 Entornos de desarrollo más utilizados

2.7.1 CARMEN

El Carnegie Mellon Robot Navigation Toolkit, es una colección de software código abierto para el control de robots móviles. Proporciona funcionalidades de navegación básica, que incluye: control de base y sensores, registro, evasión de obstáculos, localización, planificación de ruta y cartografía.

2.7.2 *ROS*

A partir de este nuevo entorno operativo el anterior ha quedado completamente obsoleto, ofrece las funcionalidades de un sistema operativo aunque debe ser instalado sobre la base de otro.

El objetivo de este sistema operativo es simplificar la tarea de crear cualquier robot imaginable, entregar facilidades para darles funcionalidades y hacerlos robustos y maleables. En ROS cada módulo posee autonomía, de forma similar a lo que sería la comunicación de varios computadores en una red LAN, o en internet en una topología punto a punto, cada módulo interactúa con los otros; esto quiere decir, que todo se hace mediante mensajes enviados de acuerdo a un protocolo, en este caso el XML-RCP, que es un sistema basado en la publicación y registro.

2.7.3 *Entornos personalizados*

Esta es la vertiente alternativa, usada principalmente por empresas privadas especializadas en el sector, donde el más representativo es PARROT, se usa un ordenador de abordo basado en ARM, equipado con todos los sensores necesarios, así como una serie de co-procesadores encargados de realizar muchos de los cálculos por “hardware”. Sobre dicho ordenador, se ejecuta un sistema GNU/Linux que tiene instalado todo el software privativo encargado de realizar todas las tareas necesarias.

2.8 Software empleado para la configuración y manejo de drones

En la aeronave:

- Ardupilot: es el software que se ejecuta en nuestra controladora de vuelo, se encarga de gestionar la información recibida del exterior (sensores físicos o radio)
- ArduCopter: cuenta con un sistema de vuelo totalmente autónomo basado en way-point, es decir, coordenadas utilizadas para ubicar puntos de referencia tridimensionales utilizados en la navegación basada en GPS, que permite una planificación de la misión y telemetría en tiempo real a través de una estación

de control en tierra. Es una herramienta que permite controlar la aeronave con cierta facilidad.

- Megapirate-NG: es el sucesor del original Ardupilot, emplea un código prácticamente idéntico en cuanto a funcionalidades con placas OpenHardware. Actualmente están comenzando a trabajar en placas con procesadores de 32 bits para portar los códigos empleados por Ardupilot.
- Multiwii: es la más sencilla de configurar pero la más limitada en cuanto a funciones. Sólo se puede instalar en controladoras clónicas con componentes de baja calidad.

En la estación base (Ground Station):

- Mission Planner: Es un software únicamente disponible para Windows, es el más depurado y conocido de los encargados de establecer una comunicación mediante MavLink con el drone y permitir así conseguir telemetrías de vuelo y configuraciones remotas de la aeronave. Además es posible obtener datos de telemetría en tiempo real (mediante radioenlace) o actualizar los parámetros de vuelo lo que permite analizar posteriormente la información obtenida durante el vuelo mediante su analizador.
- APM Planner: es un software casi exacto en funcionalidades y estética al Mission Planner que permite ejecutarse en Windows, Linux y MacOS.
- Otras alternativas menos utilizadas como Qgroundcontrol que es una plataforma de escritorio para control de aeronaves mediante Mavlink o DroidPlanner de control remoto para Android que nos permite enlazar con el UAV mediante Bluetooth o una radio de telemetría usando un cable OTG.

2.9 Protocolo MAVLINK

MavLink es un protocolo de comunicación para MAV (Micro Aerial Vehicles) que hoy en día se ha extendido a todo tipo de aviones no tripulados (tanto aéreos como terrestres).

El paquete de MavLink es básicamente una secuencia de bytes codificados y enviados a través de un transductor (a través de serie USB, radio frecuencia, WiFi, GPRS, etc.). Mediante la codificación se ordena la información en una estructura de datos de manera inteligente añadiendo sumas de control o checksums y números de secuencia para que finalmente se envíe a través del canal byte a byte.

Cada paquete MavLink está estructurado como en la Figura 3.

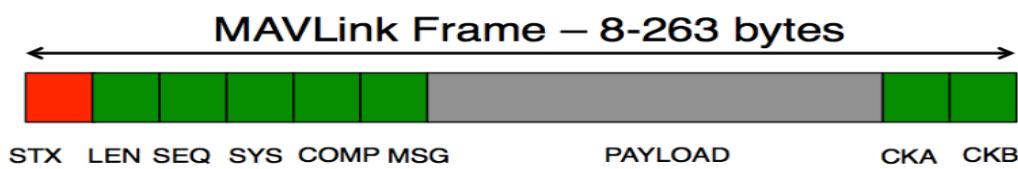


Figura 3. Estructura de paquete MAVLink

2.9.1 MavLink para transmisión de imagen en streaming

Se utilizan dos mensajes MAVLink: Uno de handshake (bienvenida), DATA_TRANSMISSION_HANDSHAKE, para iniciar, controlar y detener la transmisión de imágenes; y otro que contiene los datos, ENCAPSULATED_IMAGE.

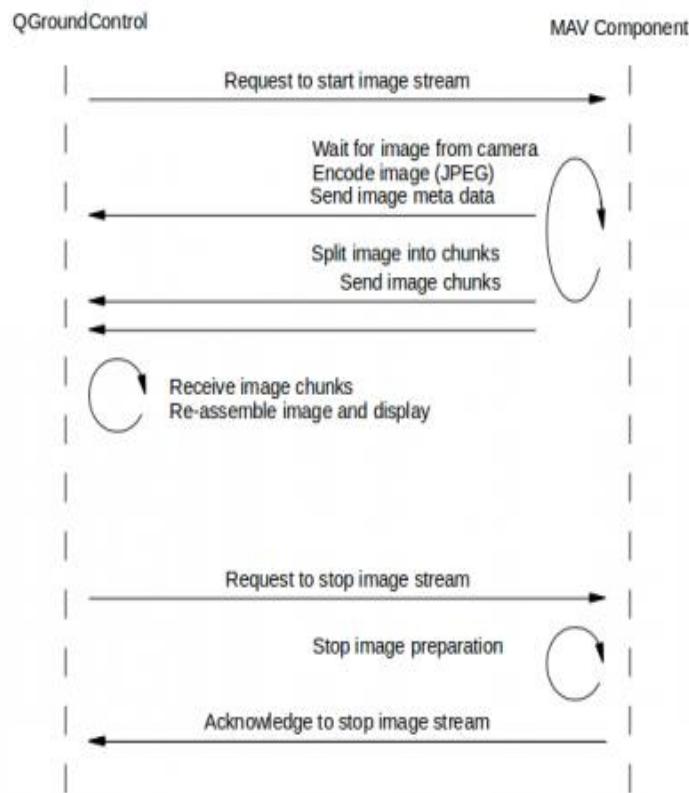


Figura 4. MAVLink para transmisión de imágenes en Streaming

Como se puede ver en la Figura 4 QGroundControl inicia la comunicación mediante una solicitud para el inicio de flujo, cuando el MAV recibe la petición de bienvenida o handshake request envía de vuelta un ACK que confirma la respuesta y se inicia el flujo de imágenes. Ahora, los datos de la imagen se dividen en trozos para poder encajarlos en mensajes MavLink comunes, a continuación se empaquetan y se transmiten. Cada paquete contiene un número de secuencia, así como el ID del flujo de imágenes al que pertenece. Se envían nuevas imágenes de forma periódica y cada una de estas viene con un nuevo paquete DATA_TRANSMISSION_HANDSHAKE ACK (que representa la confirmación de la transmisión de datos) con información actualizada de la imagen en los campos tamaño: size, paquetes: packets y carga útil: payload. Después de este paquete ACK, la nueva imagen llega como una serie de paquetes del tipo ENCAPSULATED_DATA que se refiere a datos encapsulados. Para detener un flujo de imágenes debe enviar un nuevo paquete DATA_TRANSMISSION_HANDSHAKE de solicitud con la frecuencia ajustada a 0. El MAV enviará confirmación enviando de vuelta un paquete ACK con los mismos datos que la solicitud.

2.9.2 *MavLink para transmisión de vídeo en streaming*

El protocolo de comunicación de transmisión de vídeo es mucho más simple que el de imagen. Consiste en un solo mensaje MavLink, VIDEO_STREAM que se utiliza para iniciar y detener el flujo de vídeo.

El flujo queda plasmado en la Figura 5, es generado por FFmpeg en el lado MAV. Un pequeño envoltorio MavLink agarra la imagen de la cámara, añade canales (Y) UV en formato YUV420 y alimenta la imagen. La salida se envía entonces a la estación de control o Groundstation. En la recepción, QGroundControl abre una ventana VLC para redistribuir el flujo de vídeo: La corriente MAV se ofrece como corriente de RTP (en una dirección de multidifusión) y como secuencia HTTP (para la transmisión de unidifusión directa) a la red. Esto se hace sin modificar el flujo original, para mantener en rendimiento un impacto lo más bajo posible.

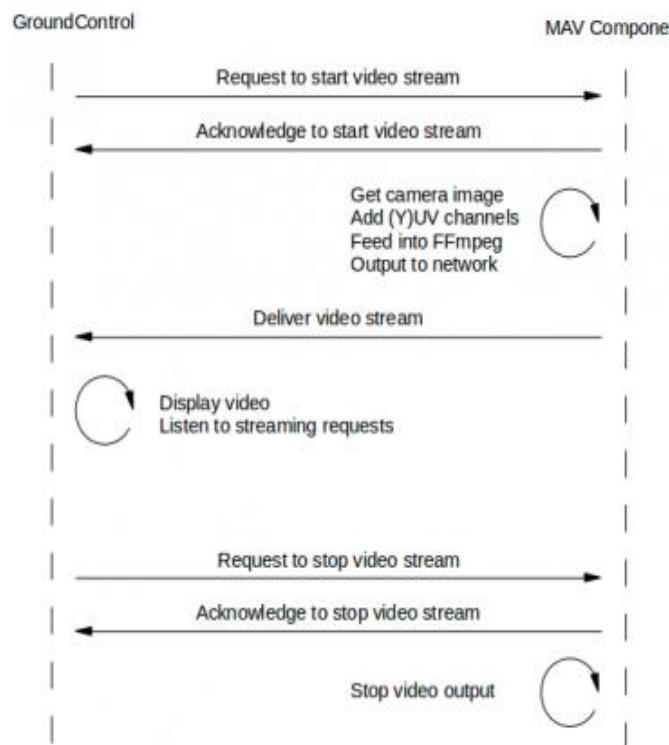


Figura 5. MAVLink para transmisión de video en streaming

Algunos de los pilotos automáticos que utilizan MavLink son ArduPilotMega, pxIMU Autopilot, SLUGS Autopilot, FLEXIPILOT y SmartAP Autopilot.

2.10 Arducopter

Como se describió anteriormente ArduCopter es una herramienta que permite controlar la aeronave con cierta facilidad, se va a realizar una descripción más detallada sobre la plataforma. Es un código abierto para aeronaves multirotor, es de las más utilizadas y fue creada por la comunidad DIY Drones. Este sistema cuenta con capacidad de vuelo totalmente autónomo basado en way-point, planificación de la misión y telemetría en tiempo real a través de una estación de control en tierra. Se ejecuta en el piloto automático ArduPilot o Pixhawk .

Principales Características:

- Control automático de gran calidad del nivel y la altitud y vuelo en línea recta. Dispone de un modo de vuelo simple. El sistema es capaz de responder a las órdenes que el usuario le da mediante la interpretación de lo que este desea hacer, es decir, el usuario se limita a indicar al drone hacia donde desea que se desplace y ArduCopter se encargará de modificar los valores necesarios para lograr ese desplazamiento manteniendo la estabilidad.
- No se requiere programación, basta con la instalación del software y la configuración de indicadores visuales rápidos, planificador de la misión y estación terrestre completa.
- Puntos GPS ilimitados. Al apuntar y hacer clic en los puntos del mapa en el planificador de la misión, el ArduCopter volará por sí mismo hacia ellos. No hay límites de distancia.
- Con simplemente mover un interruptor, el drone mantendrá su posición utilizando sus sensores GPS y de altitud.

- Es posible indicarle las coordenadas de la base, de modo que se le puede ordenar en cualquier momento volver al punto de despegue mediante el método vuelta a casa.
- Comunicación completa durante el vuelo, incluso es posible cambiar los puntos de la planificación de vuelo mientras el drone está volando.
- Despegue y aterrizaje automático.
- Multi-plataforma. Compatible con Windows, Mac y Linux. Se puede usar la utilidad de configuración Misión Planner gráfica en Windows (funciona bajo Parallels en Mac) o utilizar una interfaz de línea de comandos en cualquier otro sistema operativo. Una vez que ArduCopter está configurado, se puede utilizar con una selección de tres estaciones de tierra, incluyendo QGroundcontrol, que se ejecuta de forma nativa en Windows, Mac y Linux.
- Compatibilidad con estándares de la industria robótica líder, tales como el sistema ROS, de Willow Garages, y el protocolo de comunicaciones

2.11 Sense and avoid

Sentir y evitar, el objetivo de esta tecnología se centra en evitar colisiones en el aire, es decir, evitar obstáculos. De esta forma el UAV es capaz de detectar en vuelo a otras aeronaves dentro de un mismo entorno, lo que supone llevar a cabo maniobras de evasión automáticamente.

Se pueden diferenciar dos tipos de fuentes de información que integran estos tipos de sistemas, fuentes cooperativas y fuentes no cooperativas, las que conciernen al primer tipo son UAVs que facilitan datos sobre todos sus parámetros (rutas, dirección, posición, velocidad...) con lo que es posible prever posibles colisiones y actuar en consecuencia, por otro lado las fuentes no cooperativas son aquellos elementos que forman parte del entorno y permiten su identificación (árboles, edificios, montañas e incluso otros UAVs) pero no comparten información. En este segundo tipo de fuentes es donde actualmente se centra el estudio para la mejora del denominado Sense and Avoid.

Opciones de mercado donde podemos encontrarlo:

- DJI fue uno de los primeros en lanzar un drone que pudiera detectar y evitar obstáculos. Anunciaron Guidance, una combinación de sensores ultrasónicos y cámaras estéreo que permiten al drone detectar objetos a una distancia de hasta 65 pies (20 metros) y alejarse de objetos a una distancia pre-configurada. Posteriormente incorporaron esa tecnología al Phantom 4, este cuenta con sensores frontales de obstáculos combinados con visión avanzada por computadora y procesamiento que le permiten reaccionar y evitar obstáculos en su trayectoria sin señal GPS, utiliza un conjunto de algoritmos de software especializados para la computación espacial y detección de profundidad 3D acoplados con una unidad de procesamiento VPU, además los sistemas de detección de obstáculos de este dispositivo permiten el vuelo en una trayectoria automática alrededor de objetos. Se han encontrado pruebas de que no se trata de un sistema perfecto.
- Intel logra la tecnología sense and avoid mediante sensores activos. Gracias a su tecnología Intel RealSense integrada en un drone AscTec mostraron la forma seguir un obstáculo evitando otros a su paso. Recientemente anunciaron el Aero Ready-to-Fly Drone, un cuadricóptero funcional con Intel Aero Compute Board, equipado con capacidades de profundidad y visión Intel RealSense, ejecutando un sistema operativo Linux de código abierto. Más adelante se anunció el añadido de una cámara Intel RealSense R200 instalada en fábrica y un módulo de procesador Intel Atom para el Typhoon H de Yuneec donde el módulo mapeará el entorno del Tifón H en 3D que posteriormente utiliza para navegar de forma autónoma en su entorno, reorientándose alrededor de obstáculos. Intel finalmente adquirió el proveedor de DJI de Movidius, VPU, lo que supuso un gran poder para la compañía.
- Parrot S.L.A.M. Dunk integra aplicaciones de software avanzadas basadas en la construcción de mapeo robótico denominado localización y mapeo simultáneo, o SLAM, un programa computacional que se emplea para construir o actualizar un mapa de un entorno desconocido mientras que al mismo tiempo realiza un seguimiento de la ubicación de un agente dentro de él. El uso de

SLAM por Parrot permite al UAV entender y mapear su entorno en 3D y localizarse en entornos con múltiples barreras y donde no hay señales GPS disponibles, es decir, evita obstáculos mediante sensores activos.

- Neurala es una solución de software que analiza las imágenes de las cámaras para una mejora en la navegación del UAV, se trata de una tecnología pasiva, utiliza hardware basado en una GPU que ejecuta software de inteligencia artificial, puede ayudar a la aeronave a identificar áreas seguras para el vuelo y aterrizaje. Neurala anunció el lanzamiento de Bots Software Development Kit, que permitirá a los fabricantes instalar software 'neuronal' de inteligencia artificial directamente en sus aplicaciones sin necesidad de un hardware adicional. A pesar de estos avances, evitar la colisión total todavía está en desarrollo.
- LeddarTech anunció su modular Vu8, ideal para el uso autónomo del drone. Es un sensor activo y compacto de estado sólido LiDAR que detecta objetivos en un rango de hasta 705 pies (o 215 metros) y pesa 75 gramos, podría ser utilizado para evitar colisiones, la navegación y como un altímetro para drones. LiDAR Vu8 es inmune a la luz ambiental y fue diseñado para proporcionar una detección multi-objetivo altamente precisa en ocho segmentos independientes. Está en proceso de desarrollo y aún no existe una demostración real.

2.12 LRS o LORAN

El sistema LRS, Long Range System o sistema de largo alcance se caracteriza por su gran alcance de radio control, hasta 50Kms según el modelo de equipo y su alta fiabilidad demostrada por la gran cantidad de horas de vuelo y más de un año de pruebas exitosas en campo. En la actualidad ha sido probado con éxito y total fiabilidad hasta 26.5Km. Trabaja en la Banda ICM de 869Mhz para Europa y 902-928Mhz para América.

Fue desarrollado en Estados Unidos durante la Segunda Guerra Mundial. Operaba a frecuencias bajas con el fin de proporcionar un alcance mejorado de hasta 2.400 km con precisión de decenas de millas.

Era un sistema costoso en cuanto a implementación, se limitó su uso en el ámbito militar y comercial. La Marina de Estados Unidos comenzó el desarrollo de Loran-B, que ofreció exactitud de unas pocas decenas de pies. La Fuerza Aérea de los EE.UU. había trabajado en un concepto diferente, conocido como Cyclan, que la Marina asumió como Loran-C, que ofrecía una gama más larga que LORAN y precisión de cientos de pies.

Loran-A utilizaba las mismas frecuencias que la banda de aficionados de radio de 160 metros, y los operadores de radio se sometían a estrictas reglas para operar a niveles de potencia reducidos, dependiendo de su ubicación y distancia a la costa, los operadores de EE.UU. se limitaron a máximos de 200 a 500 vatios durante el día y de 50 a 200 vatios por la noche.

2.13 ELORAN

O LORAN mejorado, se añadió una nueva infraestructura y tecnología que ofrece más seguridad en las comunicaciones, se ofrece la posibilidad de realizar operaciones no tripuladas y/o autónomas y nuevas opciones de uso como:

- Control de tiempo de transmisión
- Las correcciones diferenciales (dLoran y / o DGPS)
- Los receptores pueden usar señales de todo en vista
- Canal de datos Loran (LDC)
- Integridad adicional
- Transmisiones sincronizadas a UTC

Se desarrolló en los Estados Unidos a partir de mediados de la década de 1990. En ese momento fue simplemente el comienzo del Programa de Modernización y Recapitalización de Loran que la Guardia Costera de los Estados Unidos (USCG) estaba llevando a cabo para asegurar que el sistema fuera eficiente y efectivo si operara más allá del año 2000. ELoran supuso una importante actualización tecnológica de Loran-C.

2.14 XLRs

XLRs tiene una telemetría con un alcance extraordinario en la banda libre ISM (desde 866 a 960MHz), mediante FHSS se consiguen saltos de frecuencia configurable con una excelente sincronización y recuperación rápida. Ofrece hasta 200Km de distancia, y se puede aumentar el alcance y asegurar buenas coberturas en diferentes condiciones con el uso de repetidores en tiempo real o mediante celdas ISM 5G RF. El sistema de radio frecuencia es una parte fundamental del sistema XLRs.

La estabilidad de frecuencia +-1ppm es un parámetro muy importante, sobre todo si la temperatura del emisor y del receptor son muy diferentes.

Solo algunos pocos equipos de gama baja y baja potencia (25mW), usan XTAL ajustado a +-5ppm, aun así es varias veces mejor que la que necesitan los chips de RF. Esta excelente estabilidad, hace que cuando los equipos estén muy lejos o reciban una señal muy débil sigan funcionando correctamente.

Utiliza 5G, Quinta generación de módulos RF de fabricación propia DMD, que supone más de 1.100 horas de vuelo FPV y FHSS, espectro extendido por salto de frecuencia, con sincronización mejorada que permite que a pesar de la interferencia entre varios canales el sistema siga sincronizado e inmune. Dependiendo del sistema XLRs en ocasiones puede estar preparado para trabajar en red por radio en las bandas ICM, Wifi, Bluetooth y por cable con RCBus (micro red local), por ethernet o Internet IoT.

3. ESTÁNDAR DE TRANSMISIÓN LORA

3.1 LoRa

Es una técnica de modulación también llamada CSS (técnica de espectro ensanchado que utiliza pulsos de chirp¹ modulados en frecuencia lineal de banda ancha para codificar información) que permite comunicaciones de largo alcance, unas decenas de kilómetros en campo abierto, con bajas velocidades de bits y bajo consumo de energía, mediante el uso de un transceptor y un microcontrolador compatible. LoRa es un protocolo de capa física, disponible en varias bandas de frecuencia mientras que LoRaWAN es un protocolo de comunicación construido en LoRa.

Los protocolos de comunicaciones son muy complejos y se dividen en capas que facilitan su descripción, además los hacen más flexibles, es decir, si ocurre algún problema en una de sus capas esta puede ser sustituida por una variante sin impactar en las demás. El estándar Lora se basa en una modulación de capa física, define como codificar los bits mediante una modulación CSS y como enviar estos bits usando una de las bandas de frecuencia ya disponible. Lora puede ser utilizado únicamente para construir su propio protocolo sobre él, pero hay que tener en cuenta las capas de la 2 a la 7.

LoraWan es un stack de protocolos estandarizados construidos sobre Lora (capa 1) y la capa final de aplicación (capa 7). El modelo OSI, en cambio, gestiona enlaces punto a punto a través de un enrutamiento directo. Podemos observar las diferencias entre el modelo OSI y el sistema LoRaWan en la Figura 6.

¹ chirp: un chirrido es una señal en la que la frecuencia aumenta (hasta chirrido) o disminuye (chirrido hacia abajo) con el tiempo.

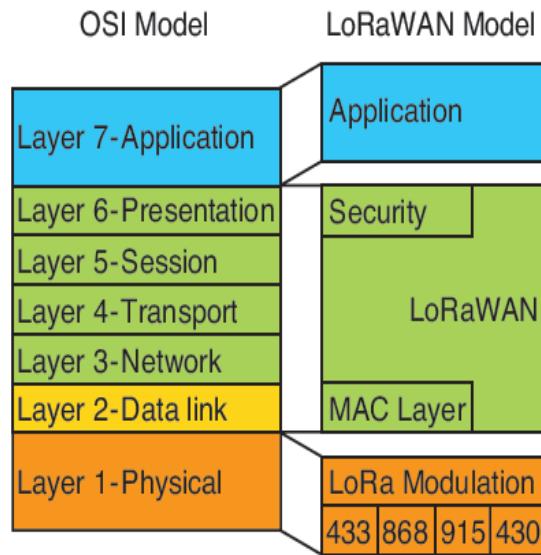


Figura 6. Diferencias entre el Modelo OSI y el sistema LoRaWAN

3.2 Arquitectura del sistema LORAWAN

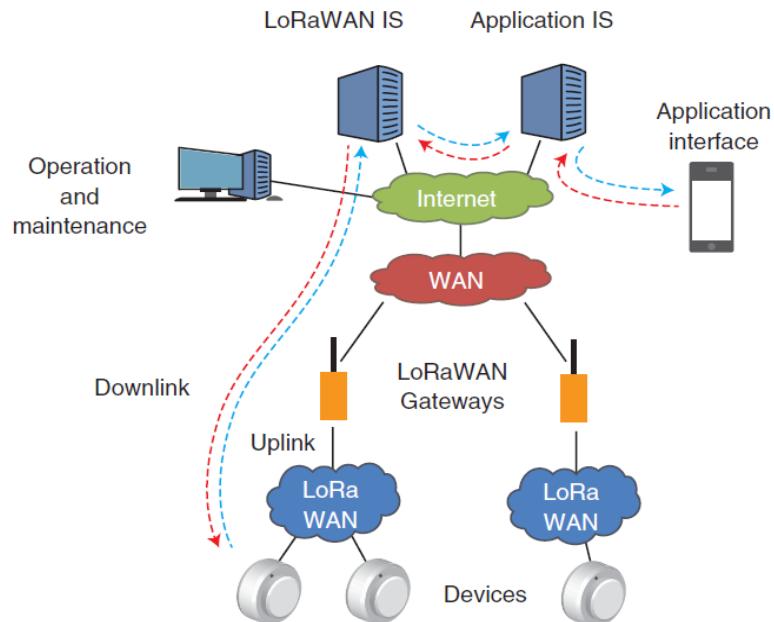


Figura 7. Arquitectura del sistema LoRaWAN

En la Figura 7 se representa la arquitectura general de una red LoraWan, que incluye tres subsistemas distintos, dispositivos, puntos de acceso y un sistema de información. Los dispositivos LoraWan se construyen utilizando chips RF de bajo coste compatibles con Lora, estos dispositivos envían mensajes utilizando un mecanismo de difusión, lo

que significa que no tienen un destinatario específico, y estos mensajes son siempre recibidos por puntos de acceso, nunca por otros dispositivos.

Los dispositivos son simples y de bajo coste pero los puntos de acceso utilizan arquitecturas complejas, se construyen a través de chips con arquitecturas RX ya que un punto de acceso escucha simultáneamente varios canales de RF y varios modos de modulación, mientras que un dispositivo cualquiera solo necesita administrar un canal y una modulación.

El sistema LoraWan requiere un sistema de información que incluya un servidor de red y uno o más servidores de aplicación, los puntos de acceso son simplemente dispositivos de paso que reciben marcos de datos RF de los dispositivos y los envían al servidor de red, quien comprueba el contenido de la trama, elimina duplicados y lo envía al servidor de aplicaciones correspondiente, además comprueba si es necesario respuesta y en caso afirmativo le pide a los puntos de acceso que la devuelvan, es una parte esencial de la red LoraWan.

Lora Alliance administra la propia especificación LoraWan pero también proporciona un stack de software de referencia LoraWan para usar en código abierto (pero la modulación Lora en sí se mantiene de forma privada).

3.3 Clases de dispositivos

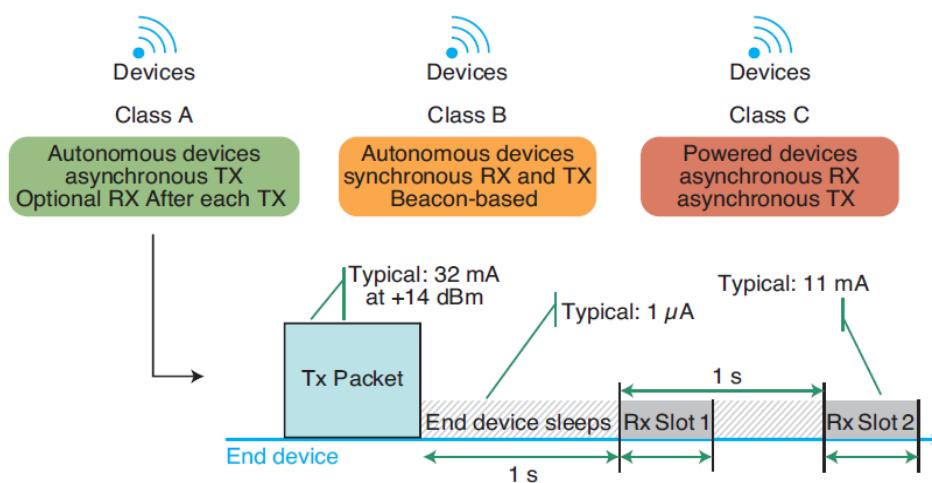


Figura 8. Clases de dispositivos

Existen tres clases de dispositivos representados en la Figura 8, el más común es el de la clase A, este duerme casi todo el tiempo, únicamente se despierta cuando quiere enviar un mensaje LoraWan en cualquier canal RF, a continuación vuelve a dormir y se despierta nuevamente un segundo más tarde, escucha el mismo canal para comprobar si el mensaje fue enviado, en caso de que no haya sido enviado no recibe nada y repite la misma operación un segundo más tarde pero en otro canal y se vuelve a dormir. Suele ser fácilmente alimentado por batería, como el receptor de RF tiene que ser encendido con poca frecuencia, la desventaja es que la red no puede enviar un mensaje a un dispositivo de clase A excepto justo después de la transmisión ascendente.

Los dispositivos de clase B resuelven el problema anterior con una ventana de recepción periódica independiente a que el dispositivo tenga o no un mensaje que enviar, en cambio este modo es mucho más complejo ya que los dispositivos deben sincronizarse con la red, esta clase no es realmente compatible con las implementaciones actuales de LoraWan.

Por último los de clase C tienen alimentación por línea por lo que pueden recibir de forma continua.

3.4 Tasa de datos

Lora soporta varias velocidades de datos, las bajas proporcionan el rango más largo pero utilizan más batería y más espectro RF que las más altas. LoraWan soporta un modo de velocidad de datos adaptable llamado ADR, gestionado principalmente por el servidor de red, el dispositivo comienza con una velocidad baja por defecto e indica al servidor si acepta el ADR a través de un bit de configuración. El servidor devuelve un comando para reducir su potencia de transmisión o aumentar su velocidad binaria.

3.5 Seguridad

LoraWan incluye dos niveles de seguridad en el estándar, en primer lugar una clave de autentificación entre el dispositivo y el servidor de red y en segundo lugar la clave de sesión de aplicación que se utiliza para cifrar el mensaje. Ambos mecanismos se construyen alrededor del algoritmo de cifrado AES.

4. EXPERIMENTACIÓN

Se basa en dos placas, una transmisora y una receptora, entre las que se establece una comunicación en la que la transmisora envía una serie de paquetes que la receptora tiene que captar. Con esta experimentación se plantea cuál es la distancia real que se puede alcanzar entre ambas placas mientras exista una conexión decente entre ellas, es decir, debido a la calidad de las antenas y a las interferencias, se producen pérdidas de paquetes, y estas aumentan con la distancia.

El interés de la experimentación surge al plantearnos qué ocurriría utilizando estas placas para establecer una conexión entre el mando de control y el drone. Asimismo, debido al interés de alcanzar largas distancias entre ambos, se plantea el uso de estas placas ya que utilizan la modulación Lora, caracterizada especialmente por su largo alcance.

Se han realizado dos programas que corresponden a la placa transmisora y receptora respectivamente. La realización de dichos programas ha sido llevada a cabo mediante el uso del software IDE Arduino.

El objetivo de estos programas es contabilizar mediante una serie de pruebas a distintas distancias los paquetes que se pierden en dicha comunicación entre placas.

4.1 Qué es IDE Arduino

Arduino IDE es un entorno de desarrollo en el que se realiza la programación de cada una de las placas de Arduino. Es una plataforma de prototipos electrónicos de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. El microcontrolador de la placa se programa usando el Arduino Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.).

Arduino simplifica el proceso de trabajo con microcontroladores y ofrece algunas ventajas para aficionados:

- Bajo coste.
- Multiplataforma.
- Entorno de programación simple y claro.
- Código abierto y software extensible. (se puede expandir mediante librerías C++ y añadir código AVR-C directamente en los programas)
- Código abierto y hardware extensible. (los diseñadores pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo)

4.2 Características de las placas

- Chip: SX1278
- Procesador: ESP 32
- Temperatura de funcionamiento: Entre -40 Celsius y 90 Celsius
- Voltaje de alimentación: Entre 3.3V y 7V
- Frecuencia: 433MHz
- Potencia de transmisión: :19.5 dBm @ 11b, 16.5 dBm @ 11g, 15.5 dBm @ 11n.
- Alta sensibilidad: -148dBm
- Salida de potencia: + 20 dBm
- Alta fiabilidad
- Distancia de transmisión (distancia de comunicación de área abierta medida 2,6 km).

4.3 Programas utilizados

4.3.1 Transmisor

```

#include <SPI.h>
#include <LoRa.h>
#include "SSD1306.h"
#include<Arduino.h>

//OLED pins to ESP32 GPIOs via this connecthin:
//OLED_SDA -- GPIO4
//OLED_SCL -- GPIO15
//OLED_RST -- GPIO16

SSD1306 display(0x3c, 4, 15);

// WIFI_LoRa_32 ports
// GPIO5 -- SX1278's SCK
// GPIO19 -- SX1278's MISO
// GPIO27 -- SX1278's MOSI
// GPIO18 -- SX1278's CS
// GPIO14 -- SX1278's RESET
// GPIO26 -- SX1278's IRQ(Interrupt Request)

#define SS      18
#define RST     14
#define DIO      26
#define BAND   433E6 //915E6
#define PABOOST true

```

Figura 9. Código transmisor primera parte

En la primera parte del programa, Figura 9, se incluyen las librerías necesarias para el desarrollo, así como las constantes que se irán utilizando a lo largo de todo el código.

```

int counter = 1;

void setup() {
  pinMode(25,OUTPUT); //Send success, LED will bright 1 second
  pinMode(16,OUTPUT);
  digitalWrite(16, LOW); // set GPIO16 low to reset OLED
  delay(50);
  digitalWrite(16, HIGH);

  Serial.begin(115200);
  while (!Serial); //If just the the basic function, must connect to a computer
  // Initialising the UI will init the display too.
  display.init();
  display.flipScreenVertically();
  display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.drawString(5,5,"LoRa Sender");
  display.display();

  SPI.begin(5,19,27,18);
  LoRa.setPins(SS,RST,DIO);
  Serial.println("LoRa Sender");
  if (!LoRa.begin(BAND,PABOOST)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  Serial.println("LoRa Initial OK!");
  display.drawString(5,20,"LoRa Initializing OK!");

```

Figura 10. Código transmisor segunda parte

Ahora como se observa en la Figura 10 se establece la configuración, los modos de los pines, el comienzo de la conexión, el tipo de letra que se utilizará en el display y, por último, el mensaje que aparecerá cuando la placa esté encendida. En caso de que todo vaya bien, se escribirá por pantalla "Inicio de LoRa con éxito". Por el contrario, aparecerá "Comienzo de LoRa fallido".

```

display.display();
delay(2000);
}
void loop() {
  if(counter==100){
    //LoRa.end();
    counter = 1;
  }
  Serial.print("Sending packet: ");
  Serial.println(counter);
  display.clear();
  display.setFont(ArialMT_Plain_10);
  display.drawString(3, 5, "Sending packet ");
  display.drawString(50, 30, String(counter));
  display.display();
  // send packet
  LoRa.beginPacket();
  LoRa.print(counter);
  LoRa.endPacket();
  counter++;
  digitalWrite(25, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(25, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
  delay(1000);
}
  
```

Figura 11. Código transmisor tercera parte

Para terminar, como se muestra en la Figura 11 se hace un bucle mediante el que se irán enviando los paquetes al receptor uno a uno, "Enviando paquete: x" es lo que se verá en pantalla.

Las pruebas van a realizarse sobre 100 paquetes, por eso, cuando el contador llegue a 100, se establecerá a 1 para comenzar una nueva prueba.

La Figura 12 representa la visualización del display de la placa transmisora cuando se inicializa la conexión entre las dos placas y comienza el envío de paquetes.

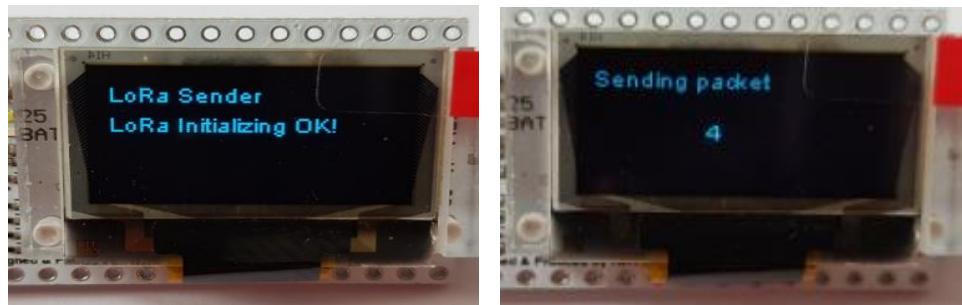


Figura 12. Inicialización y envío de paquetes desde la placa transmisora

4.3.2 Receptor

```
#include <SPI.h>
#include <LoRa.h>
#include "SSD1306.h"

SSD1306 display(0x3c, 4, 15);

//OLED pins to ESP32 GPIOs via this connection:
//OLED_SDA -- GPIO4
//OLED_SCL -- GPIO15
//OLED_RST -- GPIO16
//WIFI_LoRa_32 ports
//GPIO5 -- SX1278's SCK
//GPIO19 -- SX1278's MISO
//GPIO27 -- SX1278's MOSI
//GPIO18 -- SX1278's CS
//GPIO14 -- SX1278's RESET
//GPIO26 -- SX1278's IRQ(Interrupt Request)

#define SS      18
#define RST     14
#define DIO      26
#define BAND    433E6
#define PABOOST true

int counterRecibidos=0;
int counterPerdidos=0;
int variableAnterior=0;
int variableActual=0;
int numPrueba=1;
```

Figura 13. Código receptor primera parte

En la Figura 13, de la misma forma que en el transmisor se incluyen las librerías y se inicializan las constantes.

```
int sum=0;
boolean paqueteDistorsionado = false;
int i = 0;

void setup() {
  pinMode(16,OUTPUT);
  digitalWrite(16, LOW);    // set GPIO16 low to reset OLED
  delay(50);
  digitalWrite(16, HIGH);
  display.init();
  display.flipScreenVertically();
  display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_LEFT);

  Serial.begin(115200);
  while (!Serial); //if just the the basic function, must connect to a computer
  delay(1000);
  Serial.println("LoRa Receiver");
  display.drawString(5,5,"LoRa Receiver");
  display.display();
  SPI.begin(5,19,27,18);
  LoRa.setPins(SS,RST,DIO0);

  if (!LoRa.begin(BAND,PABOOST)) {
    display.drawString(5,25,"Starting LoRa failed!");
    while (1);
  }
  Serial.println("LoRa Initial OK!");
  display.drawString(5,25,"LoRa Initializing OK!");
```

Figura 14. Código receptor segunda parte

De nuevo se establece la configuración del receptor como se muestra en la Figura 14.

```

    display.display();
}

void loop() {
  // try to parse packet
  int packetSize = LoRa.parsePacket();
  paqueteDistorsionado = false;
  if (packetSize>0) { //Entra cuando packetSize es mayor que cero es decir cuando hay
    // read packet
    while (LoRa.available()) {
    // received a packets
    Serial.print("Received packet. ");
    display.clear();
    display.setFont(ArialMT_Plain_10);
    display.drawString(3, 0, "Prueba "+ String(numPrueba));
    display.display();

    String data = LoRa.readString(); //lee caracteres del buffer serial en una cadena.
    display.drawString(80,0, data);
    display.display();
    /*Serial.print(data);
    display.drawString(0,17, "Packet received: ");
    display.drawString(78,17, data);
    display.display();*/

    variableActual= data.toInt();
    for(int i=0; i<data.length(); i++){
      if(!isDigit(data.charAt(i))){ paqueteDistorsionado=true;
      }
      if(!paqueteDistorsionado){
        if(variableActual==(variableAnterior+1)){
          counterRecibidos++;
        }else{
          sum=variableActual-(variableAnterior-1);
          if(sum<40){
            counterPerdidos = sum + counterPerdidos-1;
          }
        }
        variableAnterior=variableActual;
      }
    }
    if((counterRecibidos+counterPerdidos)==100){
      counterPerdidos=0;
      counterRecibidos=0;
      numPrueba++;
    }

    display.drawString(0,17, "Received Packet: ");
    display.drawString(87,17, String(counterRecibidos));
    display.drawString(0,30, "Lost packet: ");
    display.drawString(87,30, String(counterPerdidos));
    display.display();
  }
}

```

Figura 15. Código receptor tercera parte

En el código de la Figura 15 se realiza un bucle que vaya recibiendo y tratando uno a uno los paquetes que va enviando el trasmisor.

En primer lugar se mide el tamaño del paquete recibido a través de la función LoRa.parsePacket(), en caso de que el tamaño sea superior a 0, es decir, hay paquete en espera, mediante LoRa.available() obtenemos el número de bytes disponibles para la

lectura. Se establece la configuración del display y posteriormente a través de LoRa.readString() se leen los datos que están en el buffer, es decir, los datos que han sido enviados por el transmisor.

Pasamos los datos recibidos, que corresponderán siempre con el número del paquete, a un entero para poder realizar ciertas operaciones sobre él.

Se ejecuta un bucle for que trata los datos cuando aún pertenecen a una forma String. Se recoge cada carácter mediante la función charAt(i) y se comprueba con isDigit() si este se corresponde con un número. En caso de que lo haga paqueteDistorsionado se pondrá a verdadero, por el contrario seguirá siendo falso. Este bucle se utiliza para comprobar si el paquete recibido está distorsionado y el flag lo usaremos posteriormente para realizar otras acciones según su valor. Se ha tomado la decisión de que, si el paquete llega distorsionado, se trate como si fuese un paquete perdido, ya que no se puede considerar como recibido debido a que se está perdiendo una parte de la información.

Si el paquete no está distorsionado y la variable actual es igual a la anterior más una unidad, significa que no hemos perdido el paquete, por lo que añadimos uno al contador de paquetes recibidos. Si no es así, obtendremos los paquetes que han sido perdidos entre la recepción actual y la anterior, y el resultado serán los paquetes perdidos. Hacemos una segunda comprobación de la primera opción por si algo falla, en este caso que el resultado obtenido anteriormente no sea mayor de 40 unidades. Si lo es, lo descartaremos como no recibido o erróneo. En caso de que se hubieran perdido más de 40 paquetes, no sería significativo porque estaríamos en un caso de desastre absoluto, ya que se recibiría un total de muy pocos paquetes. Cuando se llega a los 100 paquetes enviados, se ponen a cero los contadores y se suma una unidad al número de prueba.

```
// print RSSI of packet
Serial.print(" with RSSI ");
Serial.println(LoRa.packetRssi());
display.drawString(20, 45, "RSSI: ");
display.drawString(60, 45, (String)LoRa.packetRssi());
display.display();
}
```

Figura 16. Código receptor cuarta parte

Por último como se puede apreciar en la Figura 16 se calcula el RSSI de cada paquete, nivel de potencia de la señal recibida.

La Figura 17 representa la visualización del display de la placa receptora cuando se inicializa la conexión entre las dos placas y comienza la recepción de paquetes.

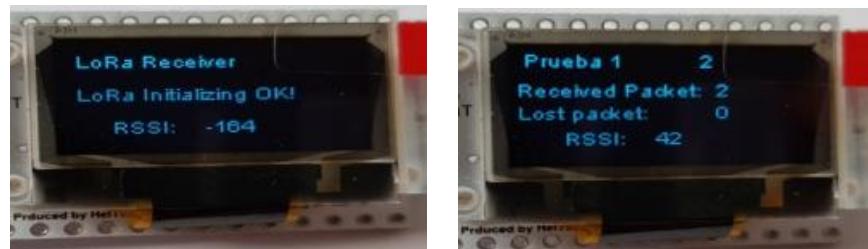


Figura 17. Inicialización y recepción de paquetes de la placa receptora

4.4 Resultados obtenidos

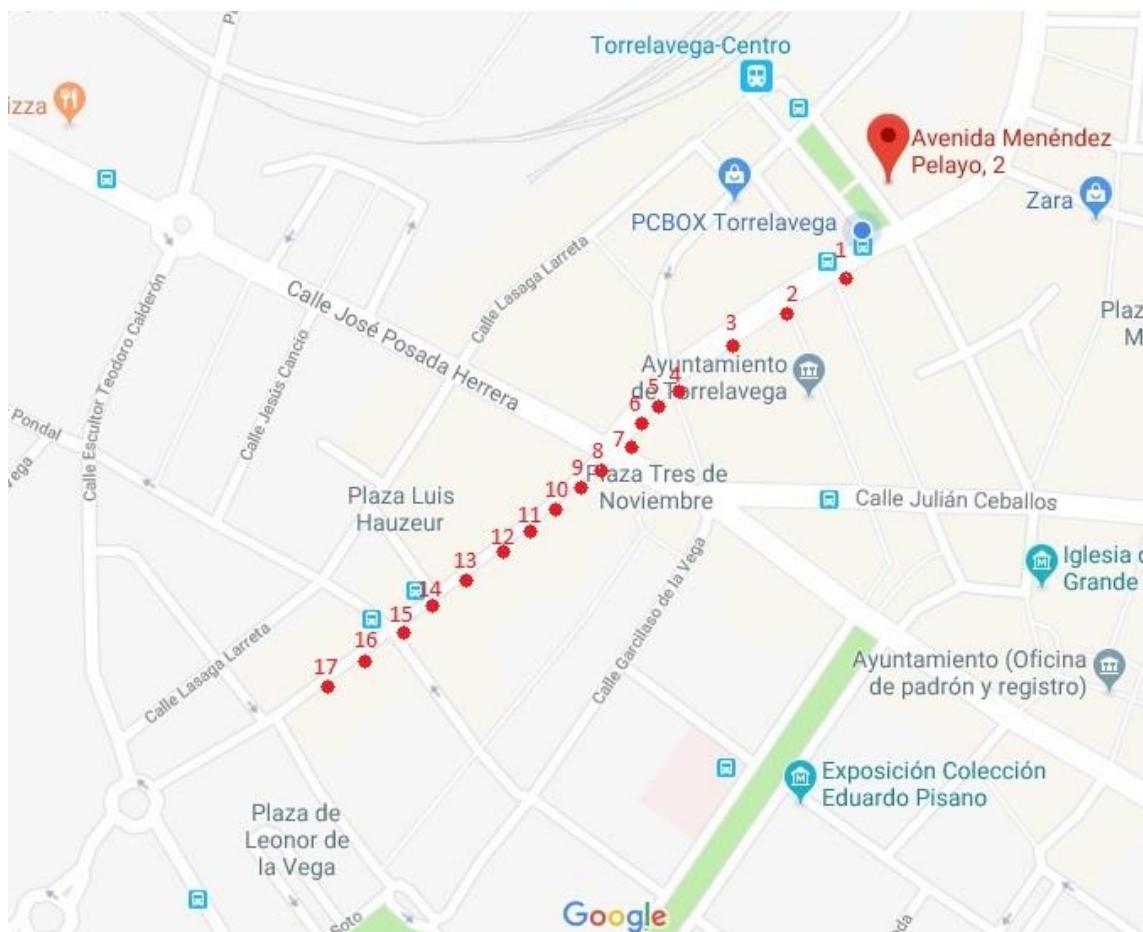


Figura 18. Mapa de pruebas

En la Figura 18 se puede observar el mapa que contiene los puntos exactos donde se han llevado a cabo las diferentes pruebas, la placa trasmisora se ha mantenido en una posición constante, que corresponde al punto de Avenida Menendez Pelayo, 2, Torrelavega y la placa receptora se ha ido desplazando por los puntos del 1 al 18 que aparecen en el mapa.

Punto de partida: Avenida Menendez Pelayo, 2

Nº de Prueba	Distancia (m)	Paquetes Recibidos	Paquetes Perdidos	RSSI máxima	RSSI mínima	RSSI media
Prueba 1	46.83	97	3	-87	-104	-95.5
Prueba 2	82.53	100	0	-102	-117	-109.5
Prueba 3	129.62	100	0	-102	-116	-109
Prueba 4	169.92	93	7	-109	-125	-117
Prueba 5	187.08	87	13	-116	-127	-121.5
Prueba 6	194.32	47	53	-103	-121	-112
Prueba 7	213.33	99	1	-121	-109	-115
Prueba 8	230.85	58	42	-124	-106	-115
Prueba 9	250.04	55	45	-127	-115	-121
Prueba 10	268.84	55	45	-114	-115	-114.5
Prueba 11	277.84	9	91	-108	-115	-111.5
Prueba 12	287.21	100	0	-110	-122	-116
Prueba 13	341.50	4	96	-111	-121	-116
Prueba 14	365.94	46	54	-104	-123	-113.5

Prueba 15	376.91	43	57	-102	-113	-107.5
Prueba 16	409.35	1	99	-108	-108	-108
Prueba 17	476.44	1	99	-108	-108	-108
Prueba 18	500	0	100	No existe recepción		

Figura 19. Recogida de datos

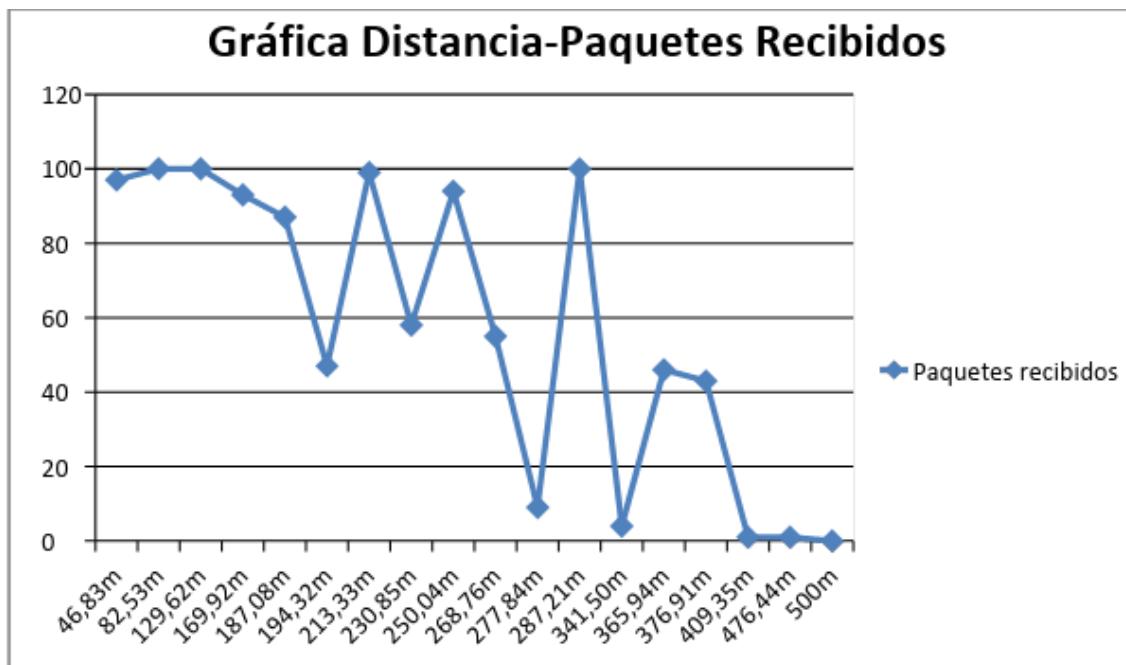


Figura 20. Paquetes recibidos en función de la distancia

5. **CONCLUSIONES Y LÍNEAS FUTURAS**

Estas pruebas han sido realizadas en ciudad, y, debido a interferencias y obstáculos se produce el fenómeno denominado propagación multicamino, es decir, las señales son recibidas por la placa receptora por dos o más caminos y en diferentes tiempos lo que puede causar problemas en la recepción de la señal debido a interacción entre las señales recibidas. Este fenómeno se ve reflejado en los resultados obtenidos, puesto que en puntos de medida consecutivos donde la distancia entre ellos es muy reducida los resultados difieren considerablemente. Como se puede observar en la Figura 20, hasta la prueba 5, que fue realizada a 187.08m de distancia entre las placas, apenas se pierde información y esto permite establecer una buena comunicación entre transmisor y receptor. Entre la 6 y la 15 se observa mucha diferencia entre puntos consecutivos, esto se debe siempre a obstáculos, que en este caso concreto, uno de muchos serían los edificios pertenecientes a la ciudad, y provocan grandes saltos en función de la distancia. A partir del punto 16, es decir, 409.35m, se obtiene un único paquete, es decir, no existe conexión entre las placas.

Se llega a la conclusión final de que, hasta aproximadamente unos 200m de distancia, se puede obtener una comunicación de calidad entre las placas, continua y sin interrupciones. Hasta los 400m, se podría realizar la comunicación, pero como se ha visto en los resultados obtenidos, hay puntos en los que prácticamente no se reciben paquetes, es decir, habría que tener en cuenta que la comunicación podría perderse totalmente en algún punto. A partir de los 400m sería imposible la conexión.

Es importante tener en cuenta que estas pruebas han sido realizadas en ciudad. Los resultados cambiarían considerablemente si se realizaran en un espacio abierto, pudiendo establecerse comunicaciones entre placas con una distancia entre ellas de hasta unos 2km.

En cuanto a las líneas futuras existe un amplio abanico por el que se puede continuar. Algunas ideas son:

1. Añadir un control de errores más sofisticado.
2. Uso de otras modulaciones.
3. Programar los módulos LoRa para conectarlos a la telemetría y realizar de nuevo las pruebas.

6. **BIBLIOGRAFÍA**

Para la elaboración del trabajo se han unido conocimientos e investigación propia junto con el análisis de la información obtenida de ciertas fuentes.

- "Integración de sistemas electrónicos de comunicación, sensado y control remoto en vehículos aéreos no tripulados", Jose Luis Rodriguez Magdaleno
 - <http://qgroundcontrol.org/mavlink/start>
 - <https://wikiversus.com/electronica-y-gadgets/drones/todo-lo-que-debes-saber-antes-de-comprarte-un-dron-fpv-que-es-en-que-consiste/>
 - <http://fpvmax.com/que-es-el-fpv/>
 - <http://www.aerial-insights.co/blog/normativa-drones-espana/>
 - http://www.gisandbeers.com/RRSS/Publicaciones/Decreto_1036-2017_drones_rpas.pdf
 - <https://tecnologia-informatica.com/que-es-drone-para-que-sirve-comprar/>
 - <https://esenziale.com/tecnologia/partes-drone/>
 - <https://dronprofesional.com/blog/cuales-son-las-partes-de-un-dron/>
 - <https://ingeoexpert.com/tipos-de-drones/>
 - <https://www.arduino.cc/>
 - <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
 - <https://es.aliexpress.com/item/LoRa-ESP32-0-96-Inch-Blue-OLED-Display-SX1278-Bluetooth-WIFI-Lora-Kit-32-Module-Internet/32829056144.html?spm=a2g0s.9042311.0.0.XgUCHO>
 - https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series#instructions
 - <http://www.fernandok.com/2018/02/esp32-lora-mudando-as-frecuencias.html>
 - <https://aprendiendoarduino.wordpress.com/2017/01/23/programacion-arduino-5/>
 - <https://www.instructables.com/id/Introducing-LoRa/>
 - <https://www.luisllamas.es/arduino-bytes-puerto-serie/>
 - <https://www.yeeply.com/blog/mejores-aplicaciones-para-drones/>
 - <https://en.wikipedia.org/wiki/MAVLink>

- <http://www.xdrones.es/mavlink/>
- <https://www.frc.ri.cmu.edu/projects/senseavoid/>
- <https://www.arducopter.co.uk/>
- <https://es.wikipedia.org/wiki/LRS>
- <https://es.wikipedia.org/wiki/LORAN>
- <https://d3.xlrs.eu/>
- <https://medium.com/beelan/haciendo-iot-con-lora-cap%C3%ADtulo-1-qu%C3%A9-es-lora-y-lorawan-8c08d44208e8>
- <https://lora-alliance.org/>