



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO SISTEMA DE CONTROL DE CUADRICÓTEROS PARA VUELO EN EXTERIORES

Autor: Diego Gil de Antuñano Durán
Director: Juan Luis Zamora Macho

Madrid
Julio de 2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Diego Gil de Antuñano Durán DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Sistema de control de cuadricópteros para vuelo en exteriores, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

6°. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 10 de Julio de 2019

ACCEPTA

Pied

Fdo.....

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

[illegible]

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Sistema de control de cuadricópteros para vuelo en exteriores
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 4º es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro,
ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Fdo.: Diego Gil de Antuñano Durán

Fecha: 10/ 07/ 2019



Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO

Fdo.: Juan Luis Zamora Macho

Fecha: 10 / 07 / 2019





GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO SISTEMA DE CONTROL DE CUADRICÓTEROS PARA VUELO EN EXTERIORES

Autor: Diego Gil de Antuñano Durán
Director: Juan Luis Zamora Macho

Madrid
Julio de 2019

Agradecimientos:

A Juan Luis por su dedicación y orientación,

A Carmen por su ayuda

A José y Antonio

SISTEMA DE CONTROL DE CUADRICÓPTEROS PARA VUELO EN EXTERIORES

Autor: Gil de Antuñano Durán, Diego

Director: Zamora Macho, Juan Luis

Entidad Colaboradora: ICAI-Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

I. INTRODUCCIÓN

Los drones o vehículos aéreos no tripulados (UAV) son dispositivos que están aumentando su presencia en el día a día. Estos vehículos tienen dos formas de funcionamiento. Sistemas controlados y sistemas de vuelo autónomos.

Un dron es un vehículo capaz de llevar a cabo un amplio abanico de tareas. Su principal ventaja es, por lo tanto, la versatilidad que presentan. Hoy en día los drones están presentes en muchos sectores. Los principales sectores en los que están presentes son: infraestructura, agricultura, transporte, seguridad y sector audiovisual.

II. OBJETIVOS

Los objetivos de este proyecto son los siguientes:

1. Integración de los sensores en el vehículo, verificación de su correcto funcionamiento y estimación de las variables de estado
2. Diseño e implantación del control de estabilización y navegación del dron

3. Planificación de misiones usando técnicas de navegación por waypoints y vuelo en exteriores

III. METODOLOGIA

La intención de este proyecto es el vuelo de un dron mediante la creación de una ruta marcada con waypoints. El dron llevará un módulo GPS para guiarle a lo largo de la trayectoria y un microprocesador raspberry pi zero W que servirá como ordenador de a bordo. Además, dispondrá de los sensores necesarios para el vuelo.

Para este proyecto se empezará configurando los distintos sensores y elementos hardware que irán en el UAV. Se harán ensayo en para comprobar el correcto funcionamiento de los mismo. Una vez el hardware este operativo se harán pruebas de comunicaciones usando el protocolo Mavlink. En estas pruebas se comprobará la correcta comunicación entre UAV y base con el objetivo de poder monitorizar el estado del dron desde la base terrestre.

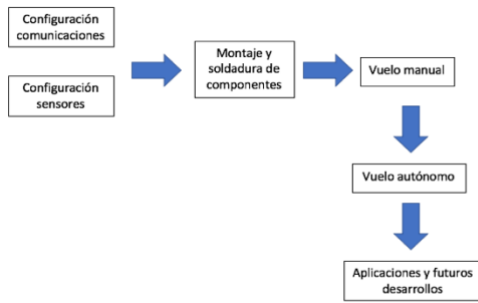


Figura 1. Metodología de trabajo

IV. RESULTADOS

Los resultados del proyecto se pueden dividir en los siguientes tres grupos: componentes hardware, sensores, comunicaciones.

A. Componentes hardware

Tras programar los ESC se realizan ensayos en el banco de pruebas para comprobar el correcto funcionamiento de los pares ESC y motor. En este ensayo se descartó el uso de uno de los motores y dos ESC que mostraban problemas. Se mide el peso que es capaz de levantar cada motor en función del PWM y se comprueba que los cuatro motores estén equilibrados.

PWM	FR (g)	BR (g)	FL (g)	BL (g)
1000	0	0	0	0
1100	19	8	26	19
1200	43	32	45	40
1300	69	65	72	66
1400	106	99	106	98
1500	145	132	132	133
1600	180	165	170	167
1700	227	204	200	202
1800	259	233	226	240
1900	299	271	256	275
2000	328	314	298	309

Tabla 1. Ensayo motores y ESC

A continuación, se vuelven a probar los motores, pero para general el PWM se usa el componente PCA9685 el cual se comunica con la raspberry por I2C. Se mandan instrucciones desde la raspberry la

cual actúa como master a la PCA9685 la cual actúa como esclava.

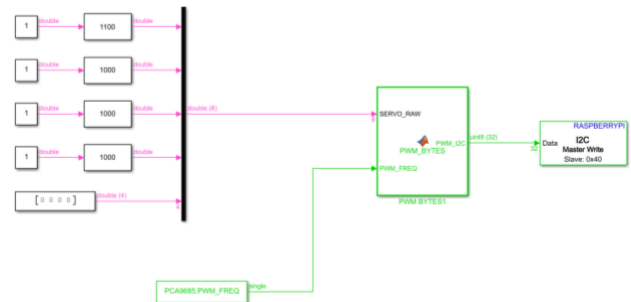


Figura 2. Ensayo PCA9685

B. Sensores

Para los sensores se realizan ensayo con el acelerómetro y con el giroscopo. Se somete a la IMU a distintas aceleraciones y velocidades angulares.

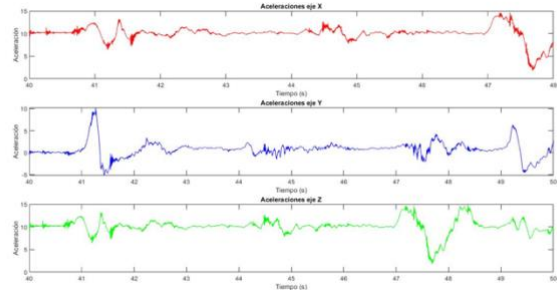


Figura 3. Ensayo acelerómetro

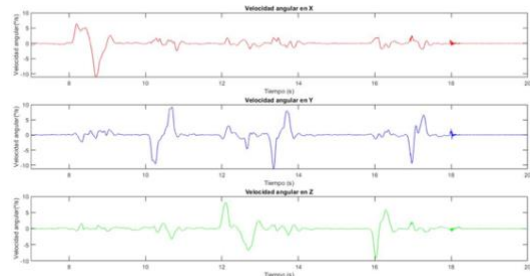


Figura 4. Ensayo giroscopo

Para el ensayo con el barómetro se subió desde la planta 0 del edificio de ICAI de la Universidad Pontificia de Comillas hasta la planta 5 y se observó una variación de presión.

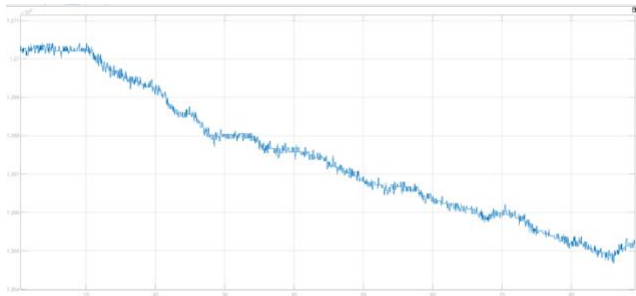


Figura 5. Ensayo barómetro

Por último, se comprueba que la información del GPS es correcta. El GPS se comunica por UART a la raspberry. Se crea un buffer de recepción de mensajes y a continuación se procesa la información la cual llega siguiendo el protocolo UBX. Se diferencia entre los dos mensajes que se mandan y se extrae la información del Payload.

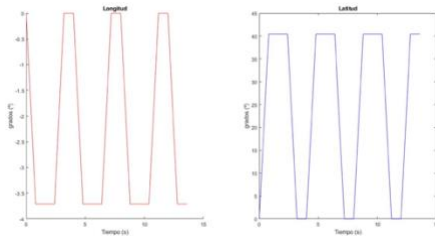


Figura 6. Ensayo GPS sistema WGC 84

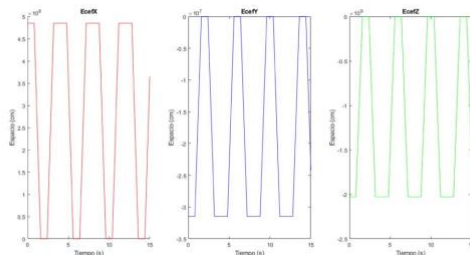


Figura 8. Ensayo GPS sistema ECEF

C. Comunicaciones

Se manda la información del hardware a través de mensajes usando el protocolo Mavlink. El sentido de envío de mensajes es único siendo los mensajes enviados por el dron y recibido por la base terrestre.



Figura 9. Ensayo Mavlink

V. APORTACIONES

Este proyecto se ha centrado en la configuración e implantación de los diferentes elementos hardware tanto sensores como componentes para un sistema de vuelo para cuadricópteros en exteriores.

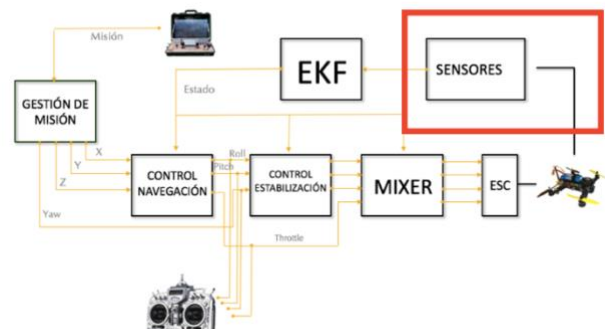


Figura 7. Aportaciones

SYSTEM OF CONTROL OF QUADRICOPTERS FOR FLIGHT IN EXTERIORS

Author: Gil de Antuñano Durán, Diego

Director: Zamora Macho, Juan Luis

Collaborating entity: ICAI-Universidad Pontificia Comillas

ABSTRACT

I. INTRODUCTION

Drones are devices that are increasing their presence on a day-to-day basis. These vehicles have two forms of operation. Controlled systems and autonomous flight systems.

A drone is a vehicle capable of carrying out a wide range of tasks. Its main advantage is, therefore, the versatility that they present. Today, drones are present in many sectors. The main sectors in which they are present are: infrastructure, agriculture, transport, security and audiovisual sector.

II. OBJECTIVES

The objectives of this project are the following:

1. Integration of the sensors in the vehicle, verification of its correct operation and estimation of the state variables.
2. Design and implementation of drone stabilization and navigation control.
3. Mission planning using waypoint navigation techniques and outdoor flight.

III. METHODOLOGY

The intention of this project is the flight of a drone by creating a route marked with waypoints. The drone will carry a GPS module to guide it along the trajectory and a raspberry pi zero W microprocessor

that will serve as an on-board computer. In addition, you will have the necessary sensors for the flight.

For this project we will start by configuring the different sensors and hardware elements that will go in the UAV. They will be tested in order to verify the correct operation of the same. Once the hardware is operational, communications tests will be done using the Mavlink protocol. In these tests the correct communication between UAV and base will be checked with the objective of being able to monitor the drone's status from the ground base.

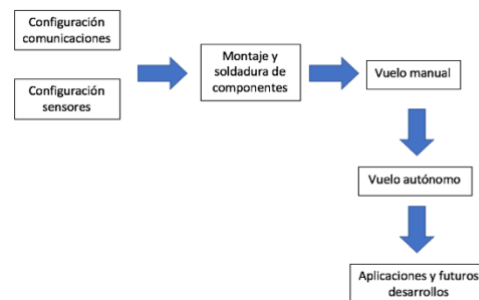


Figura 1. Metodología de trabajo

IV. RESULTS

The results of the project can be divided into the following three groups: hardware components, sensors, communications.

1. Hardware components

After programming the ESC tests are performed on the test bench to check the correct functioning of the ESC and motor pairs. In this test we realized one of the motors and two ESCs showed problems and where ruled out. The weight that each motor is capable of lifting is measured according to the PWM and it is verified that the four motors are balanced.

PWM	FR (g)	BR (g)	FL (g)	BL (g)
1000	0	0	0	0
1100	19	8	26	19
1200	43	32	45	40
1300	69	65	72	66
1400	106	99	106	98
1500	145	132	132	133
1600	180	165	170	167
1700	227	204	200	202
1800	259	233	226	240
1900	299	271	256	275
2000	328	314	298	309

Tabla 1. Ensayo motores y ESC

Next, the motors are re-tested, but using the component PCA9685 to generate the PWM signals. This component communicates with the raspberry by I2C. Instructions are sent from the raspberry which acts as master to the PCA9685 which acts as a slave.

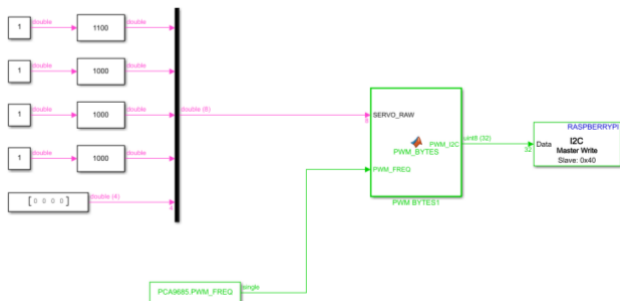


Figura 2. Ensayo PCA9685

We test the accelerometer and the gyroscope. To do that the IMU undergoes different accelerations and angular speeds.

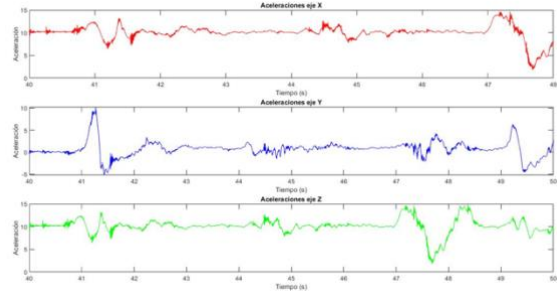


Figura 3. Ensayo acelerómetro

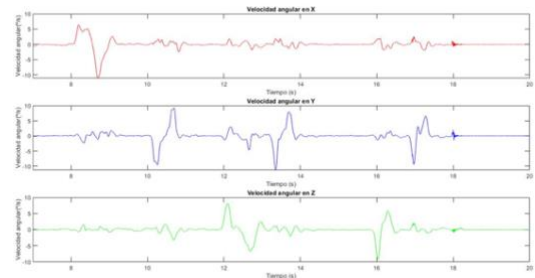


Figura 4. Ensayo giróscopo

For the test with the barometer, it was raised from floor 0 of the ICAI building of the Universidad Pontificia de Comillas to the 5th floor and a pressure variation was observed.

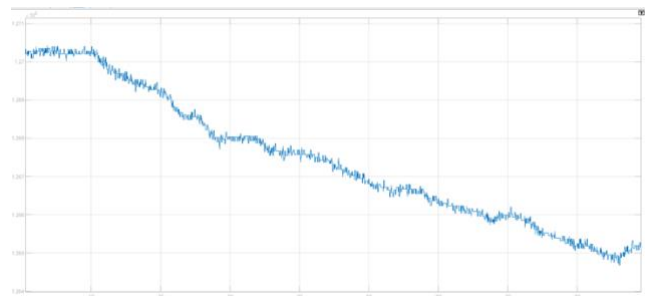


Figura 5. Ensayo barómetro

2. Sensors

Finally, it is verified that the GPS information is correct. The GPS communicates by UART to the raspberry. A message reception buffer is created and then the information which arrives following the UBX protocol is processed. We receive two different.

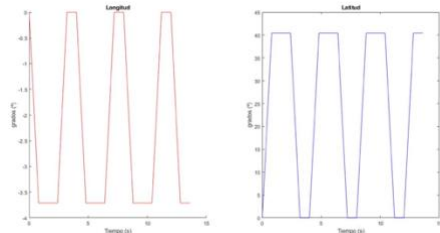


Figura 6. Ensayo GPS sistema WGC 84

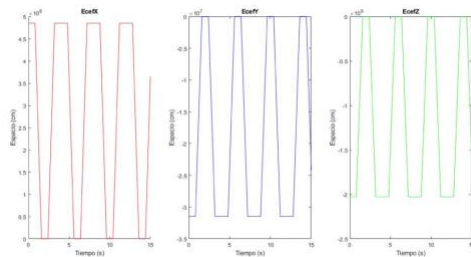


Figura 7. Ensayo GPS sistema ECEF

3. Comunicaciones

The hardware information is sent through messages using the Mavlink protocol. The sending direction of messages is unique being the messages sent by the dron and received by the ground control station.



Figura 8. Ensayo Mavlink

V. APORTACIONES

This project has focused on the configuration and implementation of the different hardware elements, both sensors and components for a flight system for quadcopters outdoors.

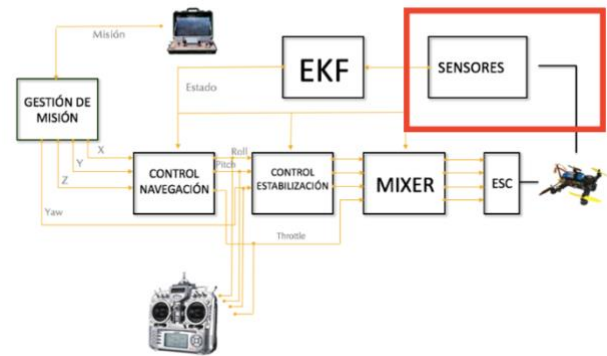


Figura 9. Aportaciones

INDICE:

PARTE 1 – MEMORIA.....	1
Capítulo 1. Introducción	3
1.1 Introducción.....	3
1.2 Motivación del proyecto.....	4
1.3 Objetivos del proyecto.....	5
1.4 Metodología de trabajo.....	5
1.5 Recursos empleados	6
1.5.1 Hardware.....	6
1.5.2 Software.....	6
1.6 Estructura de la memoria.....	6
Capítulo 2. Estado del Arte.....	7
2.1 Aplicaciones	7
2.1.1 Infraestructura.....	7
2.1.2 Agricultura.....	8
2.1.3 Transporte.....	9
2.1.4 Seguridad.....	9
2.1.5 Entretenimiento visual.....	10
2.2 Estrategias de vuelo en exteriores	10
2.2.1 Sistemas de navegación con reconocimiento visual.....	11
2.2.2 Sistemas de navegación guiados por LIDAR.....	11
2.2.3 Sistemas de navegación por waypoints (navegación guiada por GPS).....	11
2.2.4 Sistemas de navegación con sistemas inteligentes	12
Capítulo 3. Hardware.....	13
3.1 Descripción general el UAV.....	13
3.2 Motores y ESC's.....	13
3.2.1 Programación de ESC.....	14
3.2.1.1 Colocación de los motores en la estructura del dron.....	14
3.2.2 Emparejamiento ESC-Motor	15
3.3 Raspberry pi zero W	16
3.4 Controlador de PWM PCA9685.....	17
3.5 HAT para la expansión de puertos serie.....	17
3.6 IMU	18
3.7 Barómetro	20
3.8 Magnetómetro.....	20
3.9 GPS.....	21
3.9.1 Sistema de coordenadas ECEF	22
3.9.2 Sistema de coordenadas WGS 84	22
3.9.3 GPS Ublox neo 6M.....	23
3.9.3.1 U-center.....	23
3.9.3.2 Protocolo UBX.....	25
3.9.3.3 Mensaje NAV-POSLLH	26
3.9.3.4 Mensaje NAV-SOL.....	26
Capítulo 4. Software	29
4.1 Estructura general	29
4.2 GPS UBlox Neo 6M.....	30
4.3 Barómetro	34
4.4 IMU	37
4.5 Magnetómetro.....	38
4.6 Mavlink.....	39
Capítulo 5. Conclusiones y resultados	43
5.1 Ensayo con banco de pruebas para motores y ESCs	43
5.2 Ensayo movimiento de motores con PCA9685.....	43

5.3	Ensayo IMU.....	44
5.4	Ensayo barómetro.....	45
5.5	Ensayo GPS Ublox 6M	46
		47
5.6	Ensayo Mavlink.....	47
Capítulo 6.	Conclusiones y futuros desarrollos.....	49
6.1	Conclusiones.....	49
6.2	Futuros desarrollos	49
Bibliografía		51
PARTE 2 – PRESUPUESTO.....		53
Capítulo 1. Mediciones		55
1.1	Componentes de hardware.....	55
1.2	Software.....	55
1.3	Herramientas y equipos	55
1.4	Mano de obra directa	56
Capítulo 2. Precios unitarios.....		57
2.1	Hardware	57
2.2	Software.....	57
2.3	Herramientas y equipo.....	57
2.4	Mano de obra	58
Capítulo 3. Sumas parciales.....		59
3.1	Hardware	59
3.2	Software.....	59
3.3	Herramientas y equipo.....	60
3.4	Mano de obra	60
Capítulo 4. Presupuesto general.....		61
Anexo A: Información componentes hardware		63
Anexo B: Filtro de kalman extendido (EKF)		67
Anexo C: Montaje.....		71

Índice de Figuras:

Figura 1. Mercado UAVs en 2015 según estudio PwC [2]	3
Figura 2. Metodología de trabajo	5
Figura 3. Gráfico circular mercado drones	7
Figura 4. Imagen infraestructura de carreteras [7]	8
Figura 5. Mapeado del crecimiento de cultivos [8]	9
Figura 6. UAV haciendo reconocimiento visual con LIDAR [10].....	11
Figura 7. Descripción general del hardware	13
Figura 8. Programación ESCs	14
Figura 9. Sentido giro motores de un dron [19]	15
Figura 10. Raspberry pi zero W.....	16
Figura 11. PCA9685 Servotreiber	17
Figura 12. Waveshare serial expansion HAT	18
Figura 13. Orientación de los ejes del acelerómetro y del giróscopo.....	19
Figura 14. Ejes del UAV del proyecto	20
Figura 15. Triangulación señales GPS por 4 satélites	21
Figura 16. Esquema coordenadas ECEF [21].....	22
Figura 17. Sistema de coordenadas WGS 84 [22].....	23
Figura 18. Recepción de mensajes desde el Packet Console.....	24
Figura 19. Recepción de mensajes desde el Bynary Console.....	24
Figura 20. Estructura mensaje del protocolo UBX.....	25
Figura 21. Esquema general del software.....	29
Figura 22. Esquema del bloque hardware	29
Figura 23. Recepción de mensajes del GPS desde Simulink	30
Figura 24. Inicialización mensajes Matlab function del GPS	31
Figura 25. Bucle para la identificación del tipo de mensaje.....	32
Figura 26. Procesado del mensaje NAV-POSLLH	33
Figura 27. Procesado del mensaje NAV-SOL.....	33
Figura 28. Lectura datos barómetro.....	35
Figura 29. Calibración datos de presión	35
Figura 30. Esquema para la obtención de medidas	36
Figura 31. Lectura datos IMU	37
Figura 32. Inicialización IMU	38
Figura 33. Ensayo PCA9685	44
Figura 34. Medidas acelerómetro	44
Figura 35. Medidas giróscopo	45
Figura 36. Ensayo barómetro.....	45
Figura 37. Longitud y latitud GPS.....	46
Figura 38. Coordenadas sistema ECEF GPS.....	46
Figura 39. Velocidades sistemas ECEF GPS	47
Figura 40. Comunicaciones Mavlink.....	47
Figura 41. Esquema EKF.....	69
Figura 42. Pinout raspberry pi zero	74

Índice de Tablas:

Tabla 1: Fuerza motores en función de su PWM	15
Tabla 2: Estructura del mensaje NAV-POSLLH	26
Tabla 3: Estructura del Payload del mensaje NAV-POSLLH.....	26
Tabla 4: Estructura mensaje NAV-SOL	26
Tabla 5: Estructura del Payload del NAV-SOL	27
Tabla 6. Conexión barómetro	34
Tabla 7. Registros magnetómetro	39
Tabla 8. Mavlink #29 SCALED_PRESSURE [12].....	39
Tabla 9. Mavlink #30 ATTITUDE [12]	40
Tabla 10. Mavlink #24 GPS_RAW_INT [12].....	41
Tabla 11. Ensayo motores y ESC	43
Tabla 12. Componentes	55
Tabla 13. Software.....	55
Tabla 14. Herramientas y equipo.....	55
Tabla 15. Mano de obra	56
Tabla 16. Precio unitario componentes	57
Tabla 17. Precio unitario software.....	57
Tabla 18. Precio unitario herramientas y equipo	57
Tabla 19. Precio unitario mano de obra.....	58
Tabla 20. Precio total componentes.....	59
Tabla 21. Precio total software	59
Tabla 22. Precio total herramientas y equipo	60
Tabla 23. Precio total mano de obra	60
Tabla 24. Presupuesto general	61
Tabla 25. Recuento de pines periféricos.....	73

PARTE 1 – MEMORIA

Capítulo 1. Introducción

Este capítulo es una introducción al proyecto “Sistema de control de cuadricópteros para vuelo en exteriores”. En el apartado 1.1 se hablará de los drones, algunos de los usos que se le da actualmente, la importancia de estos y su potencial. En el apartado 1.2 explicaremos la motivación del proyecto, en el 1.3 los objetivos que nos hemos marcado, en el 1.4 la metodología de trabajo que vamos a seguir. A continuación, en el apartado 1.5 analizaremos brevemente los recursos que vamos a emplear tanto de hardware como de software. Por último, en el apartado 1.6 explicaremos la estructura de la memoria.

1.1 Introducción

Los drones o vehículos aéreos no tripulados (UAV) son dispositivos que están aumentando su presencia en el día a día. Estos vehículos tienen dos formas de funcionamiento. Por un lado, están los sistemas RPAS (Remotely Piloted Aircraft System) en los cuales el dispositivo se controla a distancia. Por otro lado, están los sistemas AAS (Autonomous Aerial Systems) que funcionan de forma autónoma gracias a distintas estrategias de vuelo. Independientemente del modo de navegación es necesario tener una Ground Control Station (GCS) desde la cual se opera y se monitorizan distintos parámetros del dron. [1]

Un dron es un vehículo capaz de llevar a cabo un amplio abanico de tareas. Su principal ventaja es, por lo tanto, la versatilidad que presentan. Hoy en día los drones están presentes en muchos sectores. Los principales sectores en los que están presentes son: infraestructura, agricultura, transporte, seguridad y sector audiovisual. [2] A continuación, en la Figura 1 se muestra como está repartido el mercado de los drones según un informe de PWC con datos correspondientes a 2015.

2015	
Infrastructure	45.2
Transport	13.0
Insurance	6.8
Media & Ent.	8.8
Telecommunication	6.3
Agriculture	32.4
Security	10.5
Mining	4.3
Total	127.3

Figura 1. Mercado UAVs en 2015 según estudio PwC [2]

Otra ventaja que presentan los drones es la eliminación del riesgo de vida humana ya que se trata de vehículos no tripulados. Esto hace que sean extremadamente útiles en labores de rescate o búsqueda de heridos en entornos peligrosos. Por ejemplo, los drones pueden ser usado para ayudar a combatir los incendios ya sea ayudando a extinguir el fuego con agua o con cámaras que faciliten mas información al cuerpo de bomberos y ayuden al monitoreo del incendio.

En muchas ocasiones se usan sistemas de navegación autónomos para manejar los drones lo que los convierte en extremadamente eficientes e independientes. Además, esto hace que la coordinación entre distintos sistemas UAV sea muy fácil pudiendo usar “enjambres de drones”. Este termino normalmente se usa en el ámbito militar para referirse a escuadrones de drones, pero se puede extrapolar a otros ámbitos. Un ejemplo de la coordinación de los drones son los espectáculos visuales con drones de Intel.

1.2 Motivación del proyecto

Desde su invención los drones han ayudado al ser humano en multitud de tareas, su uso supone un gran avance para la sociedad. Como ya hemos visto anteriormente los drones son extremadamente versátiles y eficientes. Pueden llevar a cabo de forma exitosa y rápida tareas complejas en cantidad de ámbitos. No obstante, a mi parecer, sigue habiendo muchos ámbitos en los que el uso de drones no esta implantado. Veo por tanto un gran potencial en el uso de estos dispositivos y mi intención es buscar nuevas aplicaciones para ellos o mejorar aquellas que ya existen. De esta manera busco dar soluciones a problemas reales a través del diseño de un control para cuadricópteros. El llevar a cabo un diseño de forma exitosa y encontrar aplicaciones que puedan ayudar a crear del mundo un sitio mejor supone un reto y una gran motivación para realizar este proyecto.

Según un artículo de Bank of America Merrill Lynch el mercado de los drones dedicados a agricultura movía 2bn de US dólares en 2015 y se estima que el 80% del mercado de los drones acabe dedicándose a la agricultura. [2] Este es uno de los sectores en los que me gustaría buscar aplicaciones para los drones y como acabamos de comprobar esta en pleno crecimiento suponiendo esto otra motivación.

Por último, existe un interés personal en el desarrollo de este tipo de dispositivos debido a mi fascinación por los drones. Además, a lo largo de la carrera hemos diseñado e implementado físicamente controles de velocidad para vehículos, controles para estabilizar vehículos equilibristas, controles para trazadores cartesianos... Este proyecto supone una oportunidad para aplicar estos conocimientos.

1.3 Objetivos del proyecto

A continuación, se muestran los objetivos fijados para el proyecto:

1. Integración de los sensores en el vehículo, verificación de su correcto funcionamiento y estimación de las variables de estado
2. Diseño e implantación del control de estabilización y navegación del dron
3. Planificación de misiones usando técnicas de navegación por waypoints y vuelo en exteriores

1.4 Metodología de trabajo

La intención de este proyecto es el vuelo de un dron mediante la creación de una ruta marcada con waypoints. El dron llevará un módulo GPS para guiarle a lo largo de la trayectoria y un microprocesador raspberry pi zero W que servirá como ordenador de a bordo. Además, dispondrá de los sensores necesarios para el vuelo.

En primer lugar, se comprobarán las comunicaciones entre el dron y el ordenador. Del mismo modo comprobaremos el correcto funcionamiento de los sensores. Una vez se comprenda el funcionamiento de los sensores y comunicaciones pasaremos al vuelo real del dron. Antes de volar de forma autónoma volaremos con un mando como si se tratase de un sistema RPAS. En el vuelo manual comprobaremos la estabilidad del dron y se monitorizará toda la información de los sensores para ver si es posible llevar a cabo el vuelo autónomo. El último paso será pasar a modo autónomo, es decir crear un sistema AAS.

Cuando el dron vuele correctamente tanto en modo autónomo como pilotado podremos pasar a buscarle aplicaciones y desarrollar accesorios para convertirlo en un dron más versátil y útil para la sociedad.

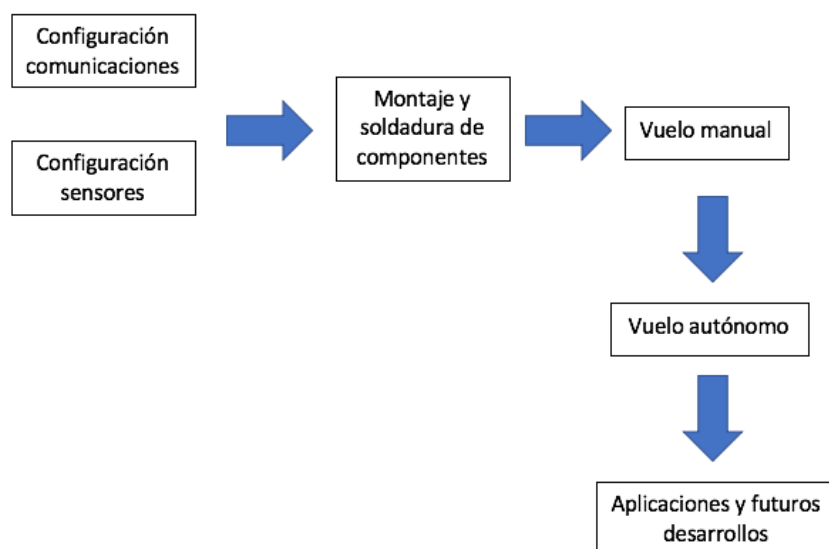


Figura 2. Metodología de trabajo

1.5 Recursos empleados

En esta sección se hará una lista del hardware y software usado a lo largo del proyecto sin entrar a hablar sobre el funcionamiento de cada uno de los componentes.

1.5.1 Hardware

- Raspberry pi zero W
- Waveshare serial expansión hat
- Modulo GPS Ublox neo m8n
- Bateria lipo de 3 celdas
- 4 ESC BLHeli
- 4 Motores Emax MT2204-2300Kw
- Estructura fibra 250 Quadcopter

1.5.2 Software

- Matlap/Simulink 2018b
- Putty
- Software u-center_v19.05

1.6 Estructura de la memoria

La memoria estará dividida en siete capítulos y tres anexos. En el capítulo 2 trataremos el estado del arte. En el hablaremos de las principales aplicaciones que se le están dando a los drones actualmente apoyándonos en el estudio de PwC “*Clarity from above*” sobre el mercado de drones. En este capítulo también se hablará sobre las principales soluciones tecnológicas que se usan para el vuelo de drones en exteriores.

En el capítulo 3 trataremos el hardware usado en este proyecto. Hablaremos sobre sus funciones, sus principales especificaciones técnicas y como se comunican con el microcontrolador. En el capítulo 4 entraremos a hablar sobre el software. Primero se explicará la estructura general del sistema de control del cuadricóptero y a continuación se hablará de la configuración del hardware usado en este proyecto. En este capítulo también explicamos brevemente las comunicaciones Mavlink entre el UAV y la base terrestre.

En el capítulo 5 tenemos los resultados de los principales ensayos realizados a lo largo del proyecto. Se mostrarán gráficas con datos obtenidas de distintos sensores. En el capítulo 6 se explicarán posibles futuros desarrollos para el proyecto.

Por último, tenemos los anexos. En el anexo A hay enlaces a la información de los componentes hardware usados. En el anexo B hablaremos del filtro de Kalman que es indispensable para la navegación. En el anexo C explicaremos el montaje del dron desde el punto de vista de los pines usados en la raspberry.

Capítulo 2. Estado del Arte

En este capítulo revisaremos las principales aplicaciones de los drones según el análisis de mercado realizado por PwC y que se ve reflejado en Figura 1. A continuación, haremos una revisión de las distintas soluciones tecnológicas que existen actualmente para el vuelo autónomo de cuadricópteros en exteriores.

2.1 Aplicaciones

Para estudiar las distintas aplicaciones nos basaremos en el estudio de PwC. [2] En la Figura 3 se ven los mismos datos de dicho estudio pero que se han dispuesto en formato de gráfico circular. Como se puede observar, solo entre el sector de infraestructura y de agricultura se cubre más del 50% del mercado de drones. Hablaremos por lo tanto de estos dos sectores y además del de transporte, seguridad y entretenimiento visual que son los siguientes sectores que más mercado abarcan.

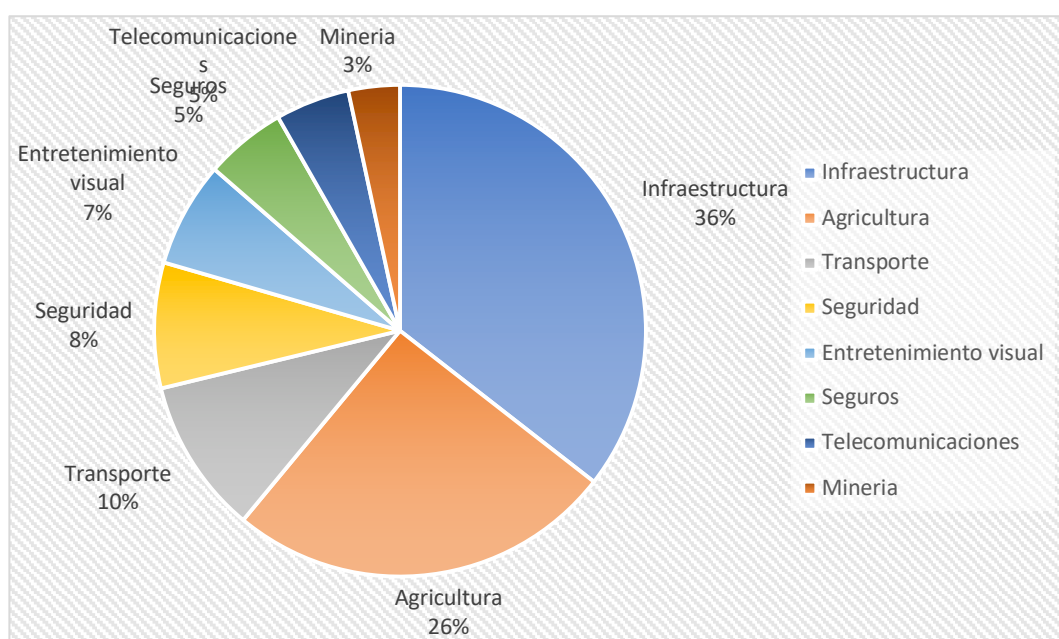


Figura 3. Gráfico circular mercado drones

2.1.1 Infraestructura

Como muestra la Figura 3 este es el sector donde más drones se venden. Los drones son capaces de llevar a cabo trabajos peligrosos sin riesgo de vidas humanas. También ayudan a obtener información variada de forma precisa y efectiva en relación con el coste de las operaciones. Las principales industrias de infraestructura que se benefician del uso de drones son: la construcción de carreteras, construcción de vías de tren, producción de energía, petróleo y gas.

En estas industrias las empresas trabajan con gran cantidad de activos que suelen estar distribuidos a lo largo de grandes extensiones de tierras. Por este motivo, las aplicaciones de drones usados en infraestructuras suelen ser la monitorización de construcciones, el mantenimiento y la realización de inventario.

Estas aplicaciones se le están dando ya a los drones. No obstante, hay start-up y empresas que investigan formas nuevas en las que los drones pueden aportar a la sociedad. En el futuro se espera combinar la tecnología de impresión 3D con los drones para reparar estructuras. También se podrán llevar a cabo trabajos en altura como la limpieza de ventanas en rascacielos eliminando el riesgo humano. [2]



Figura 4. Imagen infraestructura de carreteras [7]

2.1.2 Agricultura

La automatización de la agricultura está ganando terreno debido a la falta de personal para cultivar las tierras y la apuesta por granjas de grandes extensiones. Además, de la necesidad de un aumento en la producción de alimentos motivado por un aumento de la población mundial. Surge así la “agricultura de precisión” modalidad que se caracteriza por la automatización y el uso de algoritmos de inteligencia artificial y “big data” para mejorar los resultados de las cosechas.

Los drones llegan a cabo una importante labor en esta nueva forma de agricultura. Con ellos se pueden recorrer los campos de cultivos rápidamente y obtener información valiosa sobre como evoluciona la producción con el fin de crear procesos más eficientes. Sus principales tareas son la supervisión, análisis, monitorización y fumigación de los cultivos.

En el ámbito de la supervisión y monitorización existes antecedentes al uso de drones. El más sofisticado era el uso de imágenes vía satélites. El problema es que esta forma de monitorizar cultivos es que las imágenes de los cultivos tenían que ser pedidas con mucha antelación. Además, es una tecnología cara y muy poco precisa que no ofrecía garantías a los agricultores. [2]



Figura 5. Mapeado del crecimiento de cultivos [8]

2.1.3 Transporte

Los vehículos no tripulados están empezando a tener gran importancia en un amplio espectro de actividades relacionadas con el transporte. Desde el reparto de paquetes comprados por internet, transporte de medicinas, reparto de piezas y hasta reparto de comida para el mismo día.

La industria del transporte está optando por el uso de drones debido a la velocidad a la que pueden realizar repartos, la accesibilidad que tienen los drones y el bajo coste en comparación con otras formas de transporte en las que se necesita la presencia de un humano. [2]

2.1.4 Seguridad

Las compañías de seguridad siempre se han apoyado en las nuevas tecnologías para llevar a cabo sus tareas. No obstante, siempre se han tratado de ayudas para los operarios. Mediante el uso de drones se puede reducir el numero de operarios mejorando los resultados de vigilancia. Esto se debe a que los drones pueden recorrer rápidamente perímetros de seguridad y monitorizar grandes extensiones de terreno.

Los drones se pueden equipar con cámaras y sensores de última generación, pero siguen estando limitados por las baterías y el tiempo de vuelo. Para garantizar la calidad de las tareas de seguridad es necesario dar continuidad a la vigilancia. Por este motivo se están buscando soluciones para aumentar el tiempo de vuelo como el uso de drones que van cableados. [2]

2.1.5 Entretenimiento visual

Es uno de los sectores más populares en el uso de drones. Dentro de este sector donde más se usan es para el rodaje de películas y fotografía. Gracias a la miniaturización de la tecnología son capaces de llevar cámaras de alta calidad que antes solo se podían llevar en helicópteros. De esta manera se reducen los costes de rodajes y además se obtienen mejores resultados ya que los drones pueden acercarse más al objetivo para grabar captando ángulo que un helicóptero no podría. Esta tecnología a sido usada ya para grabar películas como “*Harry Potter*” o “*El Lobo de Wall Street*”, documentales en los que se graba la vida animal e incluso competiciones deportivas como los juegos olímpicos de Sochi en 2014. [2]

2.2 Estrategias de vuelo en exteriores

La estrategia básica de navegación sobre las que se apoyan el resto de las estrategias es la navegación inercial. Los sensores inerciales miden rotación y aceleración. Para ello se usan giroscopios y acelerómetros. Estos están integrados en la IMU (Inertial Measurement Unit).

Los errores cometidos en sistemas de navegación inerciales tienden a crecer con el tiempo. El error producido no lo podemos conocer. No obstante, podemos hacer aproximaciones del error mediante datos estadísticos. El CEP (circle of equal probability) es un método usado por U.S. Air Force para medir como crece el error producido de los sensores en el que se establecen diferentes rangos de funcionamiento. [4]

Esto, junto con el alto coste de los sensores, son las principales desventajas de este tipo de sistemas de navegación. A pesar de ello tiene algunas ventajas como que se trata de un sistema autónomo, que no necesita ayuda externa o condiciones de visibilidad para funcionar por lo que puede ser usado a oscuras, bajo el agua y otras situaciones similares.

Una forma de disminuir el error causado por la IMU es mediante el uso de sistemas duales. Esto implica tener dos IMU's, dos barómetros etc... Así conseguimos información redundante que será procesada por el control obteniendo un sistema de vuelo más fiable.

Otra forma, es mediante el uso integrado de sistemas inerciales con otros sistemas como el GPS haciendo que disminuyan notablemente los errores de media y los costes de los sensores. Por este motivo, podemos decir que los sistemas de navegación inercial son la base que usan otros sistemas de navegación de los que hablaremos a continuación.

Las estrategias de vuelo autónomo en exteriores más usadas vienen explicadas a continuación.

2.2.1 Sistemas de navegación con reconocimiento visual

Esta forma de vuelo se caracteriza por el uso de cámaras para facilitar el control del dron. Las cámaras se encargan del reconocimiento del entorno y la detección y seguimiento de posibles obstáculos. Esto junto con los algoritmos adecuados permiten al dron crear rutas alternativas a la programada e incluso misiones en las que el dron interactúe dinámicamente con el entorno. Es por lo tanto una forma de vuelo muy versátil lo que se convierte en su mayor ventaja. No obstante, requiere de una gran carga computacional. Entre los sensores de reconocimiento visual destaca el sensor de flujo óptico. Este sensor combina cámara óptica, sonar y giroscopio para obtener una percepción del entorno. [5]

2.2.2 Sistemas de navegación guiados por LIDAR

El LIDAR es un objeto capaz de detectar obstáculos y medir distancias a objetos a su alrededor. Usando un filtro que integre la información inercial obtenida con la IMU con esta nueva información se puede conseguir un sistema de navegación muy preciso. Las siguientes fases son necesarias para obtener la información del LIDAR: preprocesamiento, extracción de puntos, organización de puntos y visualización de objeto generado. [5]

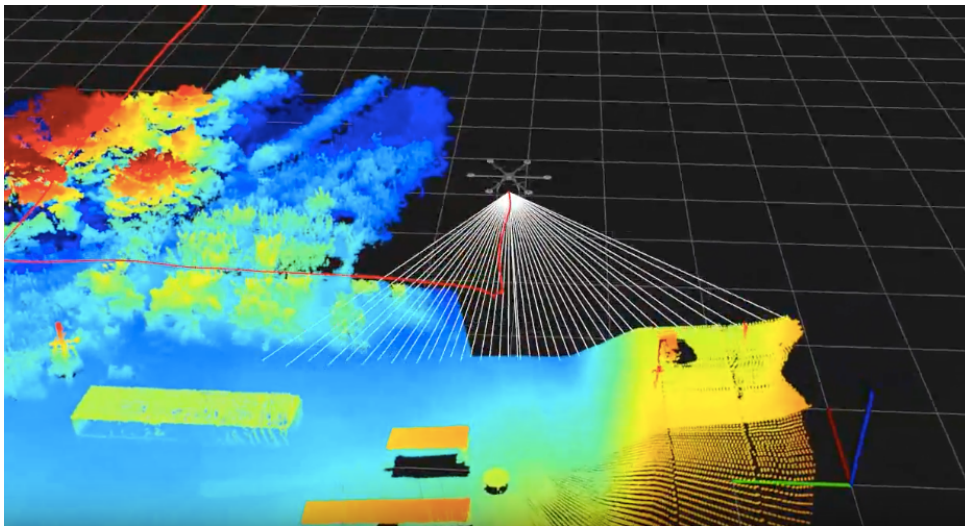


Figura 6. UAV haciendo reconocimiento visual con LIDAR [10]

2.2.3 Sistemas de navegación por waypoints (navegación guiada por GPS)

Sirve para planificar la misión del dron antes de volar. Desde la base terrestre se fijan unos waypoints o puntos de referencia que marcan el camino que debe seguir el dron. Para esta modalidad de vuelo es necesario usar GPS además de los sensores usados en los sistemas de navegación inerciales. Gracias al GPS conseguimos corregir el error que

causan los sensores de la IMU haciendo que esta estrategia de vuelo sea bastante fiable.
[1]

2.2.4 Sistemas de navegación con sistemas inteligentes

Existen otras alternativas al uso del GPS que se basan en el uso de sistemas inteligentes y algoritmos. Un ejemplo es el vuelo de drones a través de la lectura de códigos QR. Se sitúan códigos a lo largo de la zona donde va a volar el dron. A medida que este va leyendo los códigos QR es capaz de elaborar un mapa de la zona de vuelo. Este tipo de sistemas de navegación es más útil para vuelo en interiores.

Capítulo 3. Hardware

A lo largo de este capítulo hablaremos del hardware usado en el proyecto. Se analizarán las especificaciones técnicas de este, las comunicaciones y protocolos que se han usado para implementarlo y los datos que recibimos de cada sensor.

3.1 Descripción general el UAV

En la Figura 7 se muestra el dron construido en este proyecto. En el se pueden ver los distintos componentes hardware que se han usado. Se ha decidido usar el microcontrolador raspberry pi zero W como microcontrolador. La raspberry pi zero W dispone de conexiones SPI, I2C y UART las cuales usaremos para comunicarnos con los sensores y periféricos.

La raspberry se comunicará con el GPS y con la emisora por UART, con el barómetro y con la IMU y con el controlador de PWM por I2C. A lo largo de este capítulo hablaremos de los componentes hardware y sus principales especificaciones técnicas.

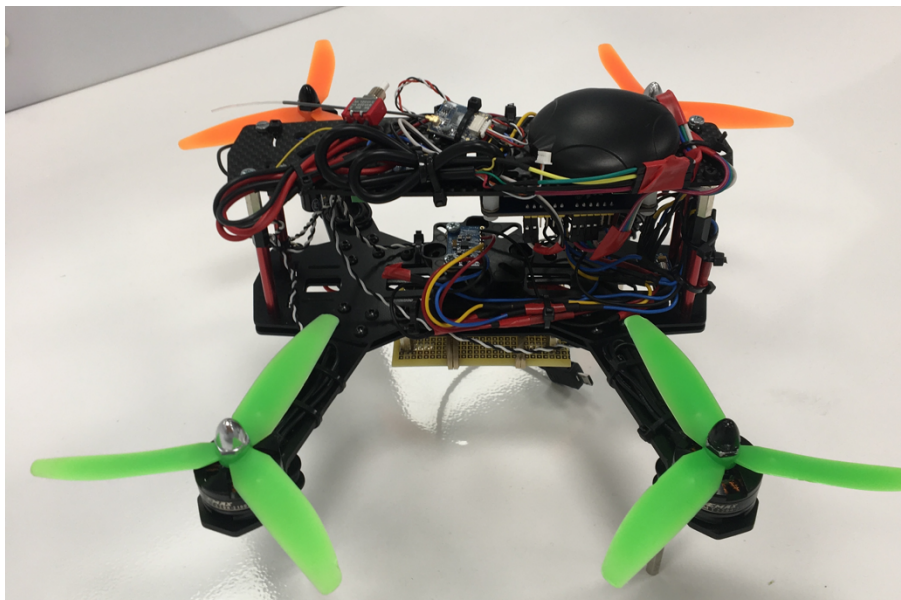


Figura 7. Descripción general del hardware

3.2 Motores y ESC's

Para este proyecto se han usado 4 Motores Emax MT2204-2300Kw y 4 ESC BLHeli. A lo largo de este apartado se explica como se han programado los ESC's, como se colocan los motores en el dron y las pruebas que se han realizado en el banco de pruebas con cada par ESC-motor, así como la forma de emparejarlos.

3.2.1 Programación de ESC

Para la programación de los ESCs se necesita una batería lipo, un distribuidor de potencia, un programador, un par motor-ESC y una aplicación que permita programar los ESCs. En este caso se ha usado la aplicación BLHeliSuite 16.7.14.9.0.0.

Conectamos la batería y el ESC al distribuidor de potencia. Por otro lado, se conecta el cable de PWM del ESC al programador que a su vez va al ordenador. A continuación, se abre la aplicación BLHeliSuite. Se selecciona Puerto COM en el que hemos conectado el programador (en nuestro caso el 5) y se pone el Baud a 19200.

Pulsamos el botón *Connect*, y luego *Read Setup* y se mostrara por pantalla la configuración actual del ESC. Hay que cambiar la configuración para dejarla igual que en la Figura 8. Pulsamos *Write Setup* para guardar la configuración, y *Disconnect* para quitar el ESC de forma segura. Repetimos el mismo proceso para los 4 ESC guardando la misma configuración.

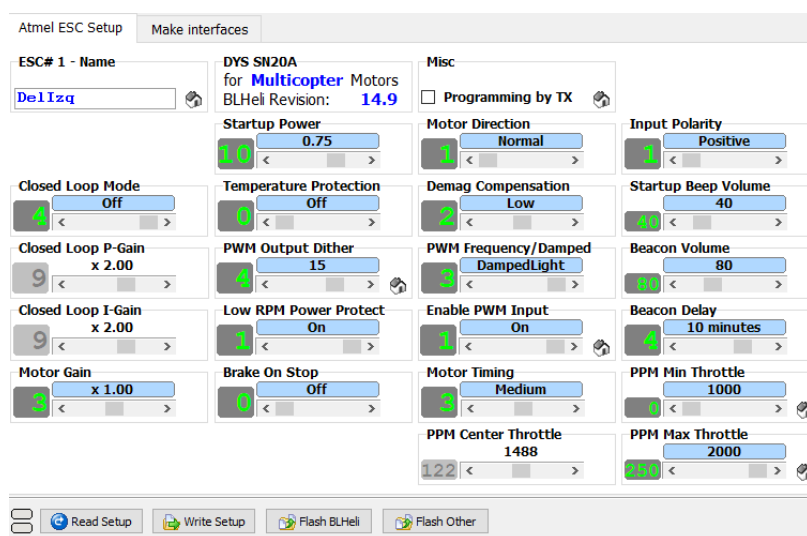


Figura 8. Programación ESCs

3.2.1.1 Colocación de los motores en la estructura del dron

De los cuatro motores existen dos que hacen girar las hélices en sentido horario y dos que lo hacen en sentido antihorario. Para el correcto funcionamiento del dron es muy importante la forma en la que se colocan los motores. La colocación depende del sentido de giro de los motores y es como se muestra en la Figura 9.

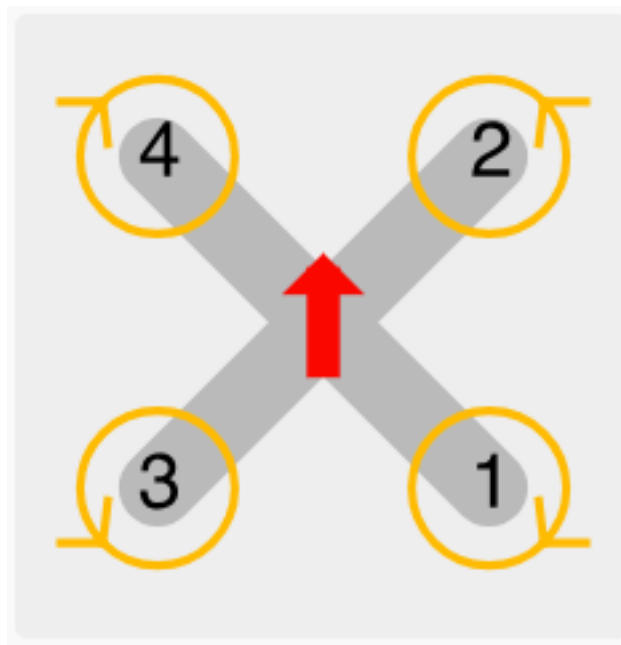


Figura 9. Sentido giro motores de un dron [19]

3.2.2 Emparejamiento ESC-Motor

Cada ESC se tiene que emparejar con un motor y probar en el banco de pruebas. El objetivo de esto comprobar el correcto funcionamiento de los motores/ESC y ver que fuerza da cada par ESC-motor con el objetivo de emparejarlos de la forma mas equilibrada posible.

Se manda a los ESCs señales PWM para controlar los motores que hacen mover las hélices. En el banco de pruebas vemos los gramos que es capaz de levantar cada motor para un PWM determinado y elaboramos una tabla. Elaboramos tablas para todas las combinaciones de motor-ESC y nos quedamos con la que de unos valores más parecidos para un PWM determinado ya que será la configuración más estable.

PWM	FR (g)	BR (g)	FL (g)	BL (g)
1000	0	0	0	0
1100	19	8	26	19
1200	43	32	45	40
1300	69	65	72	66
1400	106	99	106	98
1500	145	132	132	133
1600	180	165	170	167
1700	227	204	200	202
1800	259	233	226	240
1900	299	271	256	275
2000	328	314	298	309

Tabla 1: Fuerza motores en función de su PWM

Si se suma el peso que levanta cada motor para un PWM tenemos el peso total que puede levantar nuestro dron. Los valores que más interesa que se parezcan son los que se corresponden con el peso de nuestro dron ya que será el PWM al que funcione normalmente. En este proyecto estimamos que el peso de dron será de unos 550 gramos para lo que cada motor tendría que levantar 137,5 gramos. En la Tabla 1 se corresponde con el PWM de 1500. Estos valores son razonablemente parecidos.

3.3 Raspberry pi zero W

Para este proyecto se decide usar la raspberry pi zero W como controlador. Sus dimensiones son de 65x30x5mm lo que la hace ideal para embarcarla en un dron. Además, tiene un consumo de potencia bajo pudiendo ser alimentada por una batería lipo.

Sus principales características técnicas son:

- Conexión Bluetooth y Wireless LAN
- 512MB RAM y 1GHz single-core CPU
- Conector mini HDMI y conector micro USB
- HAT compatible de 40 pines [7]



Figura 10. Raspberry pi zero W

3.4 Controlador de PWM PCA9685

El PCA9685 es un controlador de servos por PWM. Nos permite controlar los cuatro motores de forma sencilla. El PCA9685 puede generar hasta 16 señales PWM diferente pudiendo controlar hasta 16 servos. Se puede ajustar la frecuencia del PWM hasta 1600Hz y tiene una precisión de 12bits. Las comunicaciones con la raspberry se llevan a cabo con protocolo I2C. [4]

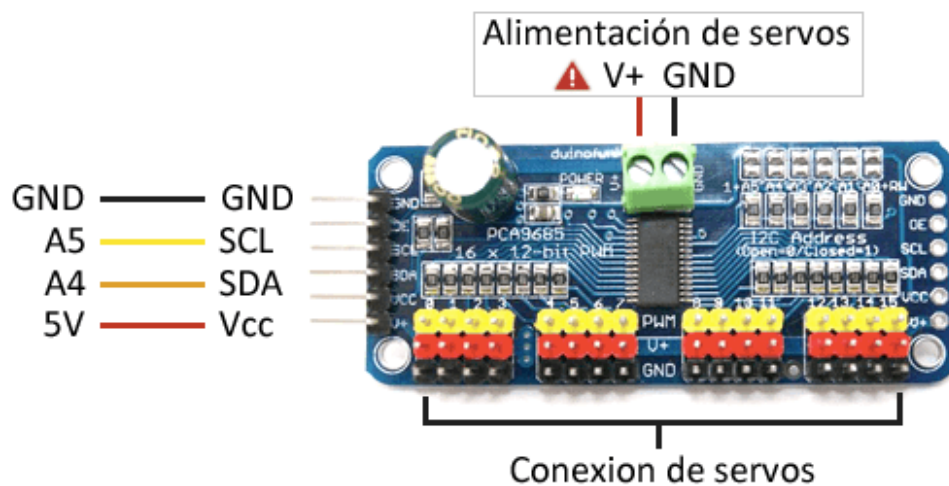


Figura 11. PCA9685 Servotreiber

3.5 HAT para la expansión de puertos serie

La raspberry pi zero dispone de un solo puerto serie tipo UART. Para este proyecto necesitamos por lo menos dos puertos UART, uno para las comunicaciones con el GPS y otro para comunicarnos con la emisora. El waveshare serial expansión hat es un módulo que nos da dos puertos UART extra. Es por lo tanto un elemento esencial que nos va a permitir comunicarnos con los periféricos.

Las principales características del waveshare expansión hat son:

- Compatible con raspberry pi zero/zero W
- Alimentación 3,3V
- Expansión de dos puertos UART y 8 pins GPIO
- Diseño compacto y bajo consumo [4]

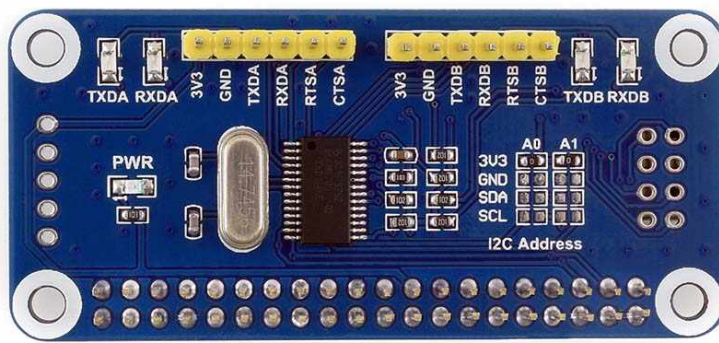


Figura 12. Waveshare serial expansion HAT

Para poder usar el serial expansión Hat hay que habilitarlo en la raspberry. Para ello hay que seguir los siguientes pasos:

1. Sudo raspi-config → Interfacing Options → I2C → Yes
2. Sudo nano /boot/config.txt
3. Añadir al final del sudo nano /boot/config.txt file la siguiente línea de código:
dtoverlay=sc16is752-i2c,int_pin=24,addr=0x48
4. Sudo reboot

3.6 IMU

En este proyecto se usa la IMU MPU-9265 que incorpora un giróscopo un acelerómetro y un magnetómetro. Nosotros solo usaremos el giróscopo y el acelerómetro. Las comunicaciones entre módulo se realizan mediante interfaz SPI a los pines de la raspberry pi zero W.

Sus principales características son:

- Temperatura de funcionamiento: entre -40° y 105°.
- Alimentación: 3.3V
- Máxima aceleración: 10000g
- Velocidad salida datos acelerómetro: 4 – 4000Hz
- Velocidad salida datos giróscopo: 4 – 8000Hz

En la Figura 13 se puede observar como están orientados los ejes para el acelerómetro y el giróscopo. El punto negro que aparece en las figuras se corresponde con el pin 1.

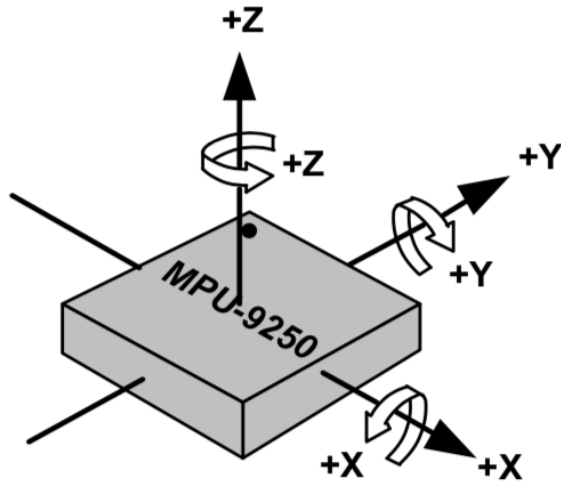


Figura 13. Orientación de los ejes del acelerómetro y del giróscopo

Al implantar la IMU en el UAV los ejes del sensor que se observan en la Figura 13 no coinciden con los ejes del UAV que se observan en la Figura 14. Por lo tanto, hay que realizar un cambio en la orientación de los ejes. Para el cambio de ejes realizamos la siguiente transformación:

Acelerómetro:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{UAV} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{IMU}$$

Giróscopo:

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}_{UAV} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}_{IMU}$$

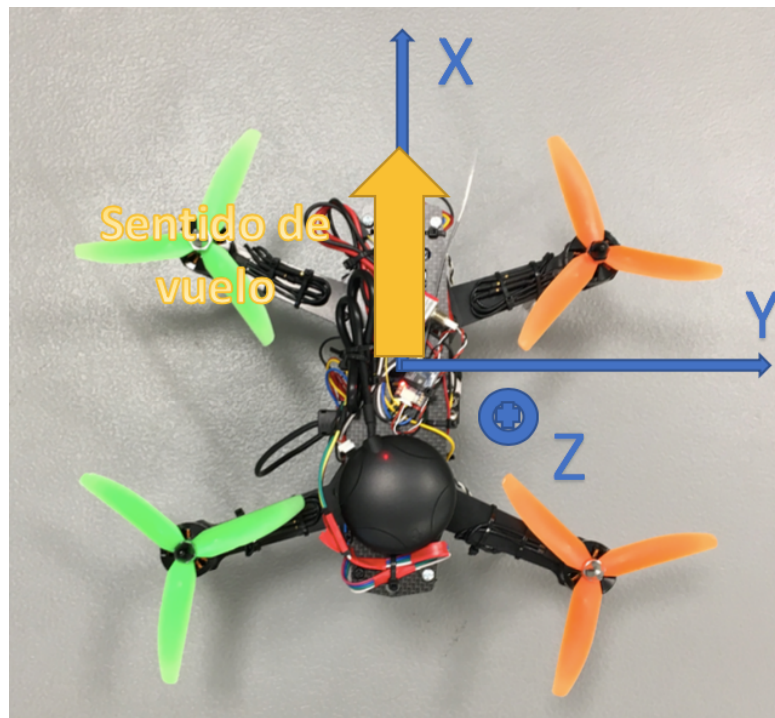


Figura 14. Ejes del UAV del proyecto

3.7 Barómetro

El M5611 es un barómetro. Este aparato mide la presión permitiéndonos calcular datos de la altura con una resolución de hasta 10 cm. Además, incluye un termómetro que nos da la temperatura con alta resolución. El M5611 permite comunicaciones con la mayoría de los microcontroladores usando protocolo I2C o SPI. Se usa un *pin select* (PS) que se conecta a *high* cuando se va a usar I2C y a *low* cuando se va a usar SPI.

Tiene unas dimensiones de 5.0mm x 3.0mm x 1.0mm lo que permite incorporarlo en casi cualquier sitio haciéndolo idóneo para un UAV. Se alimenta a 3V de tensión continua y es capaz de darnos datos de temperatura entre -40° y 125°. En cuanto a la presión se mueve en un rango de valores entre 10mbar y 1200mbar.

3.8 Magnetómetro

El magnetómetro es el elemento encargado de informarnos sobre la orientación en el campo magnético terrestre. Para ello localiza el norte geográfico. El HMC5883L usa tres sensores dispuestos en los ejes X, Y, Z para darnos la orientación respecto al campo magnético terrestre.

En este proyecto se va a usar el magnetómetro HMC5883L. Para comunicarnos con este componente se usan comunicaciones I2C siendo este un esclavo de la raspberry pi zero W. El bus de comunicaciones I2C del magnetómetro funciona a 400Hz. Este magnetómetro se alimenta a 3,3V usando los pines de la raspberry.

Sus principales características son:

- Precisión: 1° - 2°.
- Temperatura de funcionamiento: entre -30 °C y 85°C.
- Dimensiones: 2.85x3.00x3.15 mm
- Peso: 18gr.
- Velocidad salida de datos: 0,75-160Hz
- Convertidor ADC de 12 bits

3.9 GPS

El GPS es uno de los pilares sobre los que se apoya este proyecto. Se usará un módulo receptor de GPS que irá embarcado en el dron y que será fundamental para poder llevar a cabo las misiones.

El GPS (Global Positioning System) es un sistema de posicionamiento que indica la posición de un objeto en el globo terráqueo llegando a conseguir precisión de centímetros. Este sistema fue desarrollado por el departamento de defensa de Estados Unidos con finalidad militar y hoy en día es usado por la mayoría de los dispositivos que necesitan hacer uso de su posición. Existen alternativas de al GPS como el GLONASS desarrollado por la Unión Soviética, Galileo desarrollado por la Unión Europea o el sistema Beidou desarrollado por la República popular de China.

El sistema GPS cuenta con un total de 24 satélites que orbitan la tierra en 6 orbitas distintas. Se encuentran a una altura de 20.000 Km aproximadamente y giran alrededor del globo terráqueo siguiendo trayectorias sincronizadas para cubrir toda la esfera terrestre. Para poder obtener una posición el receptor necesita tener señal de como mínimo 3 satélites. Los receptores usan la triangulación para obtener la posición del receptor. Este método se apoya en la geometría y trigonometría para calcular la posición relativa del receptor respecto de los satélites y conocida la posición real de los satélites se calcula la posición real del receptor.

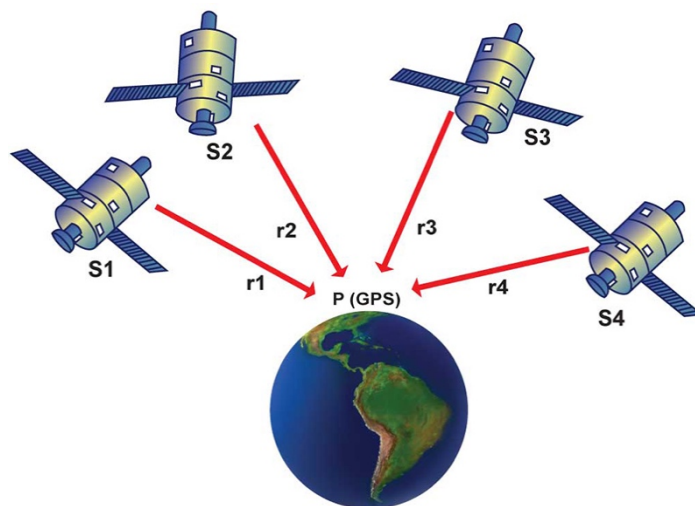


Figura 15. Triangulación señales GPS por 4 satélites

Una vez se tiene la posición del receptor existen distintas formas de obtener las coordenadas. Las más usadas y con las que trabajaremos en este proyecto son: ECEF y WGS 84. A continuación se explicaran brevemente las principales características de estos sistemas de coordenadas.

3.9.1 Sistema de coordenadas ECEF

El sistema de coordenadas ECEF (Earth Centered – Earth Fixed) es un sistema cartesiano que nos indica la posición de un objeto ubicándolo en los ejes X, Y y Z. Para ellos se considera a la tierra como una esfera perfecta y se ubica en su centro el origen de coordenadas de nuestros ejes. Los ejes estarían colocados como se muestra en la Figura 16. El eje Z en línea con el eje de rotación de la tierra, el eje X pasa por el centro de la tierra y el meridiano de Greenwich y el eje Y formando 90 grados con el eje Z y 90 grados con el eje X. Los ejes están fijos a la tierra y rotan con ella.

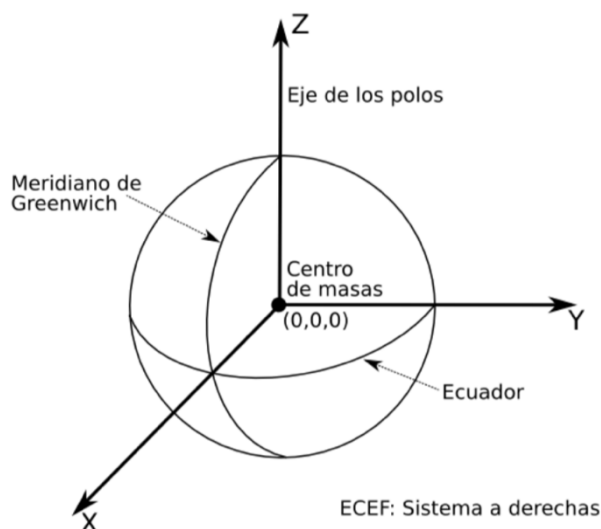


Figura 16. Esquema coordenadas ECEF [21]

3.9.2 Sistema de coordenadas WGS 84

El sistema WGS 84 (World Geodetic System 1984) es un sistema que indica la posición de un objeto considerando la tierra como un elipsoide. Este sistema nos da la posición de un objeto en función de su longitud, latitud y su altura. Para ello se toma el ecuador como origen de latitudes (latitud 0), el meridiano de Greenwich como origen de longitudes (longitud 0) y la altura se considera 0 a nivel del mar.

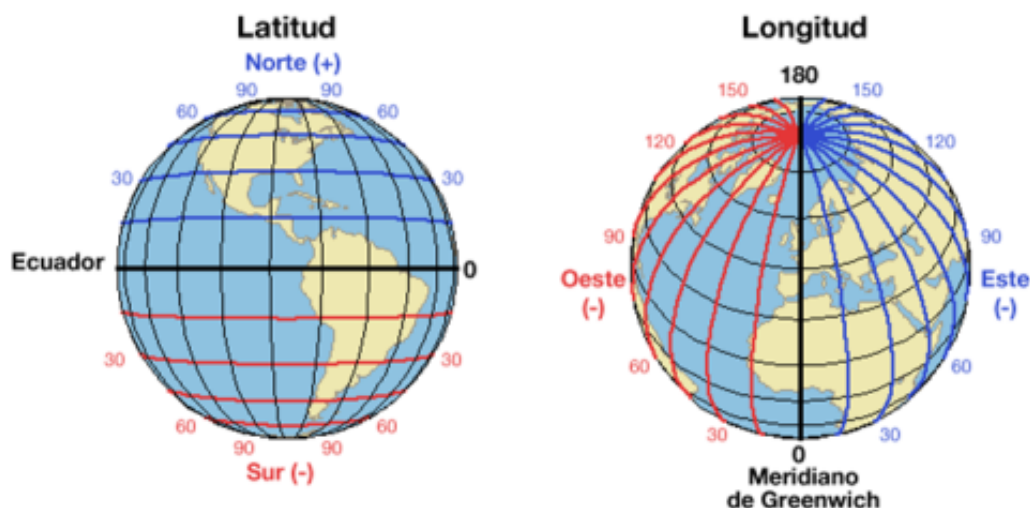


Figura 17. Sistema de coordenadas WGS 84 [22]

3.9.3 GPS Ublox neo 6M

El GPS Ublox NEO-6M es un módulo capaz de recibir y seguir señales GPS.

El módulo ublox NEO-6M dispone de las siguientes comunicaciones: UART, USB, DDC (comunicación compatible con I2C). Para todas estas comunicaciones puede usar dos protocolos distintos para mandar y recibir mensajes. Estos protocolos son: UBX y NMEA. Existe un tercer protocolo llamado RTCM pero solo sirve para mandar mensajes.

En este proyecto se va a trabajar con el protocolo UBX y las comunicaciones se realizarán por UART al waveshare serial expansión HAT que va conectado a la raspberry pi zero W.

Antes de trabajar con Matlab/Simulink se usa la aplicación u-center para configurar el GPS y asegurarse de que funciona correctamente. Esta aplicación pertenece a la compañía ublox.

3.9.3.1 U-center

El u-center es la aplicación propia de ublox. Con ella se puede llevar a cabo la configuración del módulo ublox, comprobar que se reciben los mensajes deseados y que la información recibida es correcta. Para este proyecto se ha descargado la versión u-center_v19.05.

Al usar u-center nos comunicaremos con el GPS vía USB usando un adaptador UART-USB (esto solo aplica para este apartado, durante el resto del proyecto nos comunicaremos vía UART tal y como se dijo anteriormente).

Al abrir el u-center lo primero que hay que hacer es seleccionar el puerto COM al que estamos conectados y el baudrate al que opera el módulo ublox. Esto se hace desde dos pestañas situadas en la esquina superior izquierda debajo de la pestaña File.

Una vez hecho esto ya deberíamos estar recibiendo datos del módulo ublox. Para ver los mensajes que estamos recibiendo tenemos varias opciones. Si vamos a la pestaña View → Packet Console se abre una ventana como la de la Figura 18 en la que se ve el nombre de los mensajes que estamos recibiendo. Si vamos a View → Binary Console se ven los mensajes que recibimos tal y como se muestra en la Figura 19. También existen otras formas de ver los mensajes que estamos recibiendo.

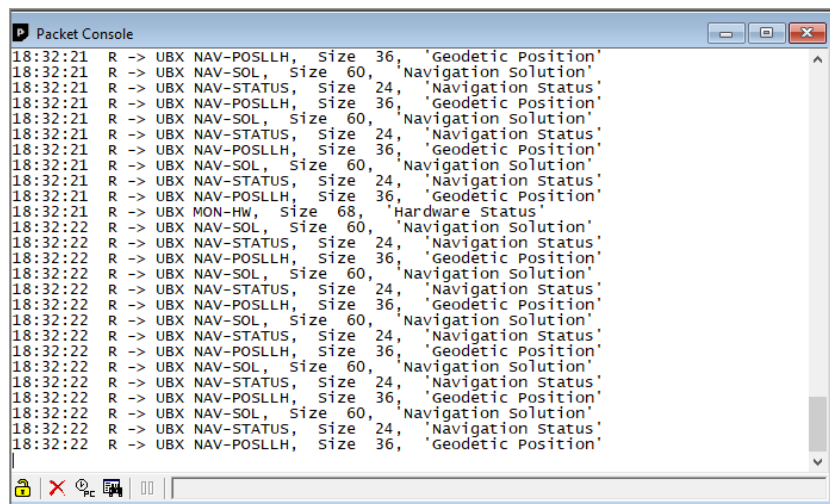


Figura 18. Recepción de mensajes desde el Packet Console

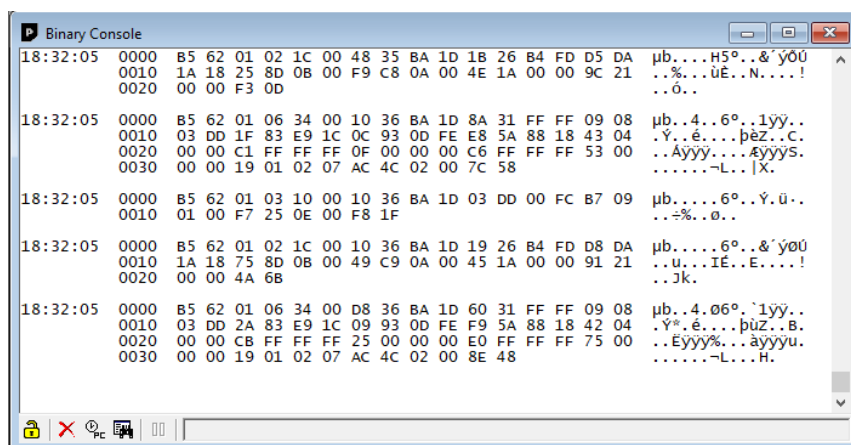


Figura 19. Recepción de mensajes desde el Binary Console

Una vez se haya comprobado que se reciben mensajes se procede a la configuración del GPS. Esto se hace desde *View → Messages View*. En esta pantalla se habilita el protocolo UBX pulsando con el botón derecho y dándole a *enable*. Los otros dos protocolos los dejamos deshabilitados. Para continuar la configuración se permanecerá en la misma pantalla.

En *UBX → CFG → PRT* y se cambia la configuración del puerto que vamos a usar. También se configura el baudrate a 38400 que es el usado comúnmente en drones. Por último, se configura la frecuencia a la que el módulo manda mensajes. Esto se hace desde *UBX → CFG → RATE* y se pone a 200ms y frecuencia 5Hz. Más adelante cuando se programe la recepción de los mensajes del GPS se tendrá que tener en cuenta estos tiempos.

Por último, se selecciona que mensajes se quieren recibir del módulo ublox neo 6M. Para ello hay que pulsar con el botón derecho sobre cada mensaje y darle a *enable*. En este proyecto solo se van a habilitar los mensajes NAV-POSLLH y NAV-SOL.

3.9.3.2 Protocolo UBX

El protocolo UBX es propiedad de u-blox y es una de las formas que tienen los módulos u-blox de comunicarse con su estación base.

Las principales características del protocolo UBX son:

- Formato compacto usando datos de 8 bits
- Usa un checksum para proteger las comunicaciones
- Usa dos identificadores en cada mensaje (class y Message ID)

Independientemente del mensaje enviado todos tienen una misma estructura básica que se puede observar en la Figura 20.



Figura 20. Estructura mensaje del protocolo UBX

El mensaje empieza con la cabecera que tiene una longitud de 2 bytes. El contenido enviado en ella es común a todos los mensajes del protocolo UBX. Se envía 'B5 62' (en hexadecimal). Esto indica al receptor que se trata de un mensaje UBX. A continuación, se manda la Clase y el ID. Cada uno de estos campos tiene una longitud de 1 byte y con ellos se define que mensaje se está mandando. A esto le siguen 2 bytes en los que se indica la longitud del Payload. Luego sigue el Payload que es la información útil del mensaje e.g: latitud, longitud, tiempo... El Payload no tiene una longitud definida. Por último, hay dos bytes de Checksum en los que se indica que se ha finalizado correctamente el envío del mensaje.

3.9.3.3 Mensaje NAV-POSLLH

El mensaje NAV-POSLLH manda información sobre posición geodésica en la que se encuentra el módulo. Este mensaje usa el sistema WGS 84 para enviar la información de posición. Esta información es muy útil para la navegación en exteriores. La estructura del mensaje es la que se muestra en la Tabla 2.

Cabecera	Clase	ID	Longitud del Payload (en bytes)	Payload	Checksum
0xB5 0x62	0x01	0x02	28	-	-

Tabla 2: Estructura del mensaje NAV-POSLLH

La longitud total del mensaje es de 36 bytes y la longitud del Payload es 28. La información del Payload se ha recogido en la Tabla 3. En ella se especifican los distintos campos que tiene el mensaje, con que bytes se corresponden dentro del Payload, el formato de la información, las unidades y la escala que se tiene que aplicar. Para este proyecto interesan especialmente los campos de longitud, latitud y altura.

Byte de offset	Formato	Escala	Nombre	Unidades	Descripción
0	U4	-	iTOW	ms	Hora de la semana del GPS
4	I4	1e-7	lon	deg	Longitud
8	I4	1e-7	lat	deg	Latitud
12	I4	-	height	mm	Altura sobre el elipsoide
16	I4	-	hMSL	mm	Altura sobre el nivel del mar
20	U4	-	hAcc	mm	Exactitud horizontal estimada
24	U4	-	vAcc	mm	Exactitud vertical estimada

Tabla 3: Estructura del Payload del mensaje NAV-POSLLH

3.9.3.4 Mensaje NAV-SOL

Este mensaje combina información sobre la posición, velocidad y tiempo usando uso del sistema de coordenados ECEF. Al igual que en el mensaje NAV-POSLLH esta información es muy útil para la navegación. La estructura del mensaje NAV-SOL se muestra en la Tabla 4.

Cabecera	Clase	ID	Longitud del Payload (en bytes)	Payload	Checksum
0xB5 0x62	0x01	0x06	52	-	-

Tabla 4: Estructura mensaje NAV-SOL

La longitud total del mensaje es de 60 bytes y la longitud del Payload es 52. La información del Payload se ha recogido en la Tabla 5. En ella se especifican los distintos campos que tiene el mensaje, con que bytes se corresponden dentro del payload, el formato de la información, las unidades y la escala que se tiene que aplicar. Para este proyecto interesan especialmente los campos de coordenadas X, Y y Z en sistema ECEF y los de velocidad en X, Y, Z en el sistema ECEF.

Byte de offset	Formato	Escala	Nombre	Unidades	Descripción
0	U4	-	iTOW	ms	-
4	I4	-	fTOW	ns	-
8	I2	-	week	-	-
10	U1	-	gpsFix	-	-
11	X1	-	flags	-	-
12	I4	-	ecefX	cm	Coordenada X en ECEF
16	I4	-	ecefY	cm	Coordenada Y en ECEF
20	I4	-	ecefZ	cm	Coordenada Z en ECEF
24	U4	-	pAcc	cm	Precisión de posición
28	I4	-	ecefVX	cm/s	Velocidad en X en ECEF
32	I4	-	ecefVY	cm/s	Velocidad en Y en ECEF
36	I4	-	ecefVZ	cm/s	Velocidad en Z en ECEF
40	U4	-	sAcc	cm/s	Precisión de la velocidad
44	U2	0,01	pDOP	-	-
46	U1	-	reserved1	-	-
47	U1	-	numSV	-	-
48	U4	-	reserved3	-	-

Tabla 5: Estructura del Payload del NAV-SOL

Capítulo 4. Software

4.1 Estructura general

En la Figura 21 se muestra el Simulink general que se usará en este proyecto. Es especialmente interesante el bloque de *HARDWARE* dentro del cual tenemos los tres bloques que se muestran en la Figura 22. Hay otro bloque llamado *MONITORIZACION* que hemos usado para ver algunas de las variables de nuestro control. Hay otro bloque llamado *SIMULACION* que aparece comentado y por último el bloque de *CONTROL*.

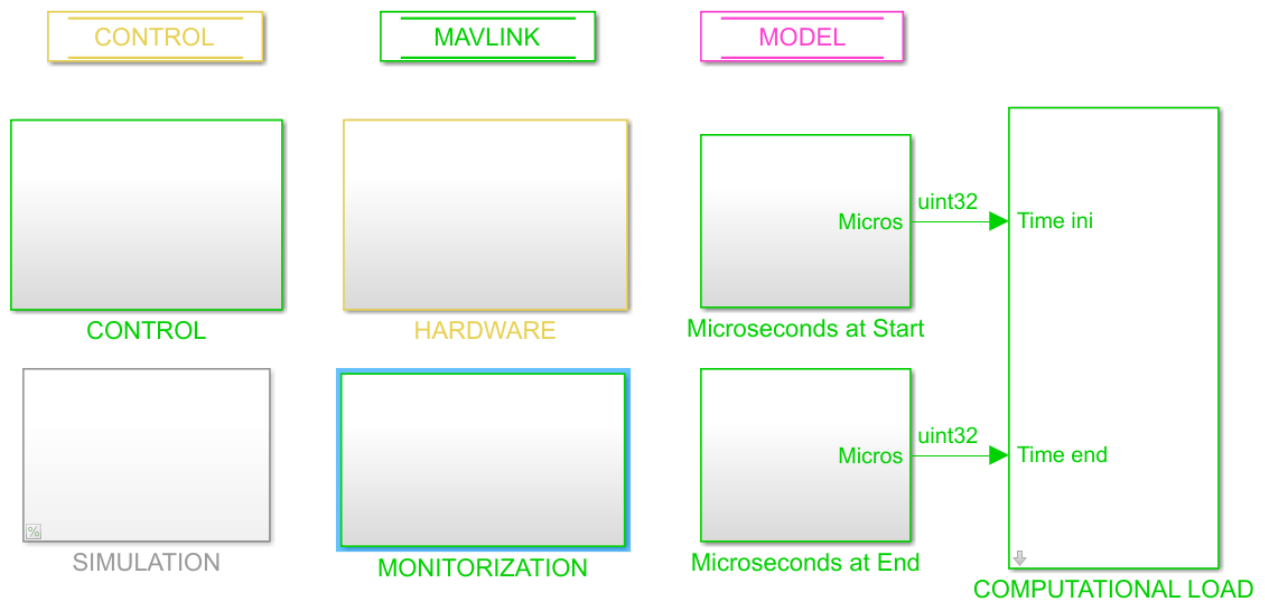


Figura 21. Esquema general del software

Dentro del bloque de *HARDWARE* tenemos otros tres bloques llamados *SENSORS*, *ACTUATORS* y *COMMUNICATIONS*. En *SENSORS* se llevará a cabo la comunicación, configuración y recepción de la información de los sensores. Por lo tanto, dentro tendremos bloques específicos para el GPS, para la IMU, para el barómetro y para el magnetómetro.

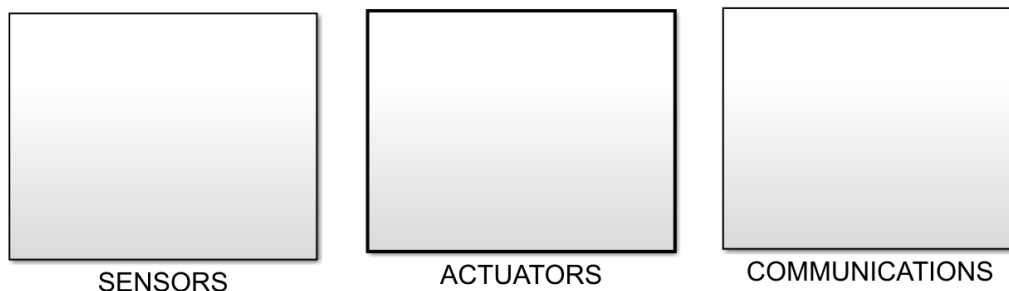


Figura 22. Esquema del bloque hardware

El bloque de *COMMUNICATIONS* tiene dentro tres bloques. Estos se llaman *UART*, *UDP* y *TCP/IP*. Estos se usan durante la primera fase del proyecto para comprobar que las comunicaciones se realizaban correctamente. Por último tenemos el bloque de *ACTUATORS* el cual apenas se usa en este proyecto.

Este proyecto se centra en el bloque de *SENSORS*. Dentro de este bloque hay bloques para el GPS Ublox NEO 6M con el que se establecen comunicaciones mediante uart, para la IMU MPU9250 se establecen comunicaciones mediante SPI, para el barómetro MS5611 se establecen comunicaciones mediante SPI y otro bloque para el magnetómetro se establecen comunicaciones mediante I2C. A lo largo de este capítulo se explicará como se han recibido los mensajes de todos estos componente y como se han configurado.

4.2 GPS UBlox Neo 6M

En la Figura 23 se muestra el simulink usado para recibir, procesar, guardar y monitorizar la información que se obtiene del GPS. La recepción se hace por la *UART* a la raspberry. Se establece un buffer de recepción de información de 144 bytes con un tiempo de muestreo de 0,8 segundos. Es importante que el tiempo de muestreo sea el especificado ya que hemos configurado el GPS para que mande un mensaje cada 0,2 segundos siendo mensaje más corto de 36 bytes por lo que necesitamos 0,8 segundo si queremos llenar 144 bytes de información.

El buffer entra a la *Matlab function* donde los datos son procesados. También entran los buses de *CONTROL* y *GPS* con los campos que se van a rellenar. La salida son los buses de *CONTROL* y *GPS* con la información actualizada. También salen datos con el objetivo de facilitar la monitorización.

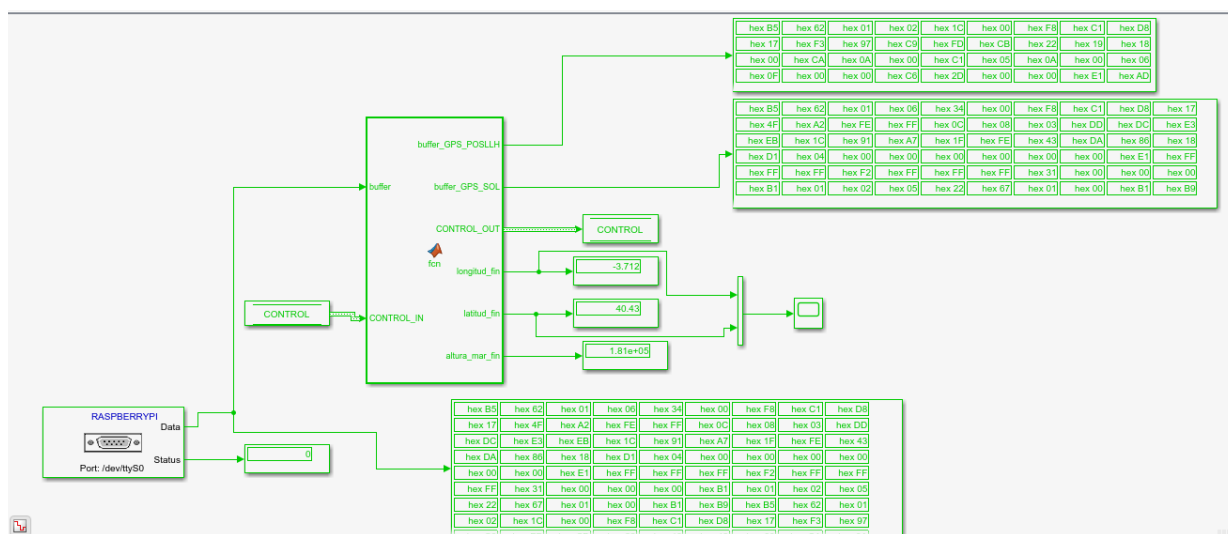


Figura 23. Recepción de mensajes del GPS desde Simulink

En la Figura 24 se muestra la inicialización de las variables que se van a usar en la función.

En la Figura 25 se muestra el código usado en el bloque *Matlab function* para identificar el mensaje recibido. A la función le entra un buffer formado por 72 datos del tipo `int8`. Se usa un bucle *for* para recorrer el buffer y comparar cada dato con los dos bytes de cabecera, el byte de ID y el de clase. Se busca encontrar una secuencia de datos consecutivos que cumpla que los dos primeros bytes sean igual a la cabecera de los mensajes UBX '0xB5 0x62'. El tercer dato tiene que ser igual a la clase del mensaje que en el caso de este proyecto será siempre 0x01 ya que se reciben mensajes de la clase NAV. Por último, se compara el cuarto dato con el ID del mensaje. Si coincide que el ID es 0x02 entonces se trata de un mensaje NAV-POSLLH y se pone a true la variable *enable_posllh* que es la que indicadora de que se esta recibiendo un mensaje NAV-POSLLH y se guarda en *posicion_posllh* la posición en la que empieza el mensaje. Si el ID es 0x06 entonces se trata de un mensaje NAV-SOL, se pone a true la variable *enable_sol* y guardamos en la variable *posición_sol* la posición donde empieza el mensaje.

También se ha hecho una comprobación adicional para cerciorarnos de que vamos a recibir un mensaje completo. Como el buffer tiene una dimensión de 144 bytes si se identifica un mensaje con NAV-POSLLH para que el mensaje este completo tiene que estar en la posición 109 del buffer como muy lejos ya que el mensaje tiene una longitud de 36 bytes. Si el mensaje empezase en la posición 110 del buffer y leyésemos el mensaje estaríamos perdiendo el ultimo byte del mensaje. Ocurre lo mismo para el mensaje NAV-SOL pero con la diferencia de que para leer un mensaje completo tiene que estar en la posición 85 ya que este mensaje tiene una longitud total de 60 bytes.

```
CONTROL_OUT=CONTROL_IN;
GPS_OUT=GPS_IN;
enable_posllh=false;
enable_sol=false;
posicion_posllh=0;
posicion_sol=0;
buffer_GPS_POSLLH=zeros(36,1,'uint8');
buffer_GPS_SOL=zeros(60,1,'uint8');
```

Figura 24. Inicialización mensajes Matlab function del GPS

```

for i=1:1:72
    if buffer(i)== hex2dec('B5') && buffer(i+1)==hex2dec('62')&&
buffer(i+2)== hex2dec('01') && buffer(i+3)==hex2dec('02') && i<=109
        enable_posllh=true;
        posicion_posllh=i;
    elseif buffer(i)== hex2dec('B5') && buffer(i+1)==hex2dec('62')&&
buffer(i+2)== hex2dec('01') && buffer(i+3)==hex2dec('06') && i<=85
        enable_sol=true;
        posicion_sol=i;
    else
    end
end
end

```

Figura 25. Bucle para la identificación del tipo de mensaje

Una vez identificados los tipos de mensajes se guarda cada mensaje en un buffer distinto desde el que se procesa la información. Se creará un buffer_GPS_POSLLH de dimensión 36 bytes tipo uint8 donde se guardarán los mensajes NAV-POSLLH y un buffer_GPS_SOL de dimensión 60 bytes tipo uint8 donde se guardarán los mensajes NAV-SOL.

Para procesar la información se usan en las Tabla 3 y Tabla 5 donde se explica en qué bytes esta guarda la información, la escala que usa y en que formato esta guarda. Las siglas I4 se refieren a int32 y U4 a uint32. Se usa el comando *typecast* para seleccionar la información y pasarla al formato deseado.

Por último, se guarda en el bus de CONTROL y GPS la información que interesa. Todo esto queda reflejado en la Figura 26 para el mensaje NAV-POSLLH y en la Figura 27 para el mensaje NAV-SOL.

```

length_posllh = uint16(0);
latitud = int32(0);
longitud = int32(0);
altura_mar = int32(0);
altura_eli = int32(0);

if enable_posllh
buffer_GPS_POSLLH=buffer(posicion_posllh:posicion_posllh+35);
%datos
length_posllh = typecast(buffer_GPS_POSLLH(5:6), 'uint16');
longitud = (1e-7)*typecast( buffer_GPS_POSLLH(11:14), 'int32');
latitud = (1e-7)*typecast( buffer_GPS_POSLLH(15:18), 'int32');
altura_mar = (1e-3) *typecast( buffer_GPS_POSLLH(22:25), 'int32');
altura_eli = typecast( buffer_GPS_POSLLH(18:21), 'int32');
%asignacion bus local
GPS.LLA=[latitud,longitud, altura_mar];

end

```

Figura 26. Procesado del mensaje NAV-POSLLH

```

length_sol = uint16(0);
ecefX = int32(0);
ecefY = int32(0);
ecefZ = int32(0);
ecefVX = int32(0);
ecefVY = int32(0);
ecefVZ = int32(0);
if enable_sol
    buffer_GPS_SOL=buffer(posicion_sol:posicion_sol+59);
    %datos
    length_sol = typecast(buffer_GPS_SOL(5:6), 'uint16');
    ecefX = typecast(buffer_GPS_SOL(19:22), 'int32'); %coordenada X
    ecefY = typecast(buffer_GPS_SOL(23:26), 'int32'); %coordenada Y
    ecefZ = typecast(buffer_GPS_SOL(26:29), 'int32'); %coordenada
Z
    ecefVX = typecast(buffer_GPS_SOL(33:36), 'int32'); %velocidad
coordenada X
    ecefVY = typecast(buffer_GPS_SOL(37:40), 'int32'); %velocidad
coordenada Y
    ecefVZ = typecast(buffer_GPS_SOL(41:44), 'int32'); %velocidad
coordenada Z

    %actualizacion bus de control
    CONTROL_OUT.INPUT.GPS.EARTH_POS=[ecefX, ecefY, ecefZ];
    CONTROL_OUT.INPUT.GPS.EARTH_VEL=[ecefVX, ecefVY, ecefVZ];
end

```

Figura 27. Procesado del mensaje NAV-SOL

4.3 Barómetro

Como se menciona anteriormente las comunicaciones con el barómetro se van a realizar mediante SPI. Para poder usar SPI es necesario conectar el pin PS (*Pin Select*) a tierra. En este proyecto se conecta el pin CSB (Chip Select) al pin de la raspberry GPIO7 (SPI0_CE1_N). Es importante tener esto en cuenta para configurar correctamente los bloques en Simulink.

Pin MS5611	Pin raspberry
PS	Gnd
CSB	GPIO7 (SPI0_CE1_N)
SDO	GPIO9 (MISO)
SDI/SDA	GPIO9 (MOSI)
SCLK	GPIO11 (SCLK)

Tabla 6. Conexión barómetro

Para llevar a cabo la lectura de la presión y la temperatura se siguen los pasos indicados en la Figura 30. Primero se leen seis registros que se necesitarán usar más adelante para calibrar los valores de temperatura. A continuación, se lee la información de presión y temperatura bruta (la que nos indica el barómetro). Con estos valores y la información de los registros de calibración se calcula la temperatura. Por último, con la temperatura se calcula el valor de la presión. Para este sensor no es necesario llevar a cabo ningún tipo de inicialización.

Con el valor de la presión y temperatura se puede calcular la altura a la que se encuentra el módulo usando la siguiente ecuación. [9]

$$P = P_0 * e^{-\frac{g*M(h-h_0)}{R*T}}$$

Se toman como valores de referencia $P_0 = 101300Pa$ y $h_0 = 0$ y se despeja el valor de h .

Tanto la calibración de la presión como el cálculo de la altura se realizan en una *Matlab function* dentro del bloque *BARO_SCALING* en el cual se actualizan los resultados en el bus de *CONTROL*. Esto se puede ver en la Figura 29. En el bloque *READ_PROM* se lee la información para calibrar, en el bloque *READ_RAW_TEMP* se lee la información sobre la temperatura y en el bloque *READ_RAW_PRESSURE* se lee la información sobre la presión. Estos bloques se pueden ver en la Figura 28.

PRESSURE AND TEMPERATURE CALCULATION

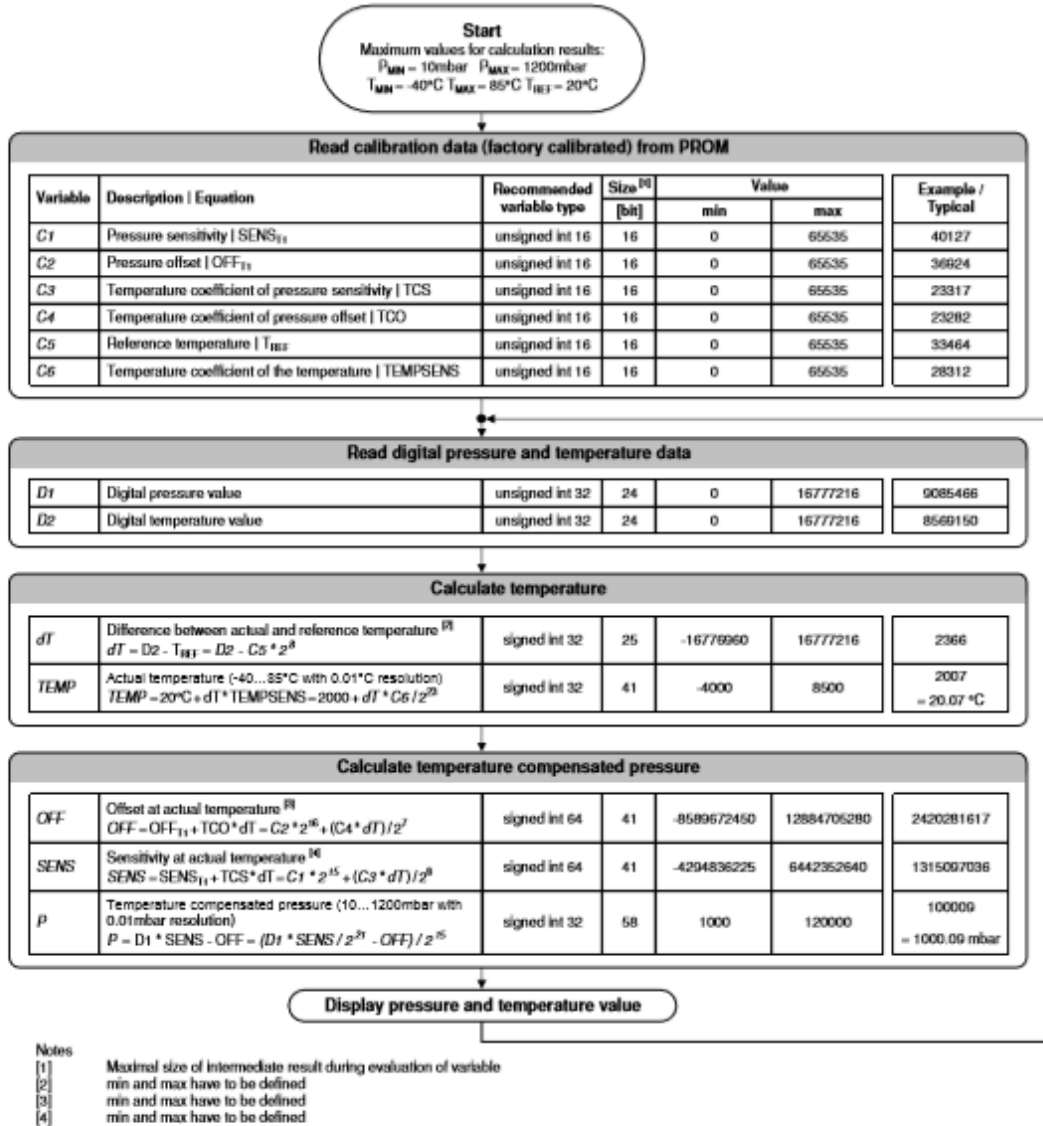


Figura 30. Esquema para la obtención de medidas

4.4 IMU

El sensor MPU-9250 permite comunicaciones por SPI o por I2C. En este proyecto se usa SPI para comunicarse con el MPU-9250. Se conecta el *Chip Select* al pin GPIO 8 (CEO) de la raspberry ya que se ha usado el GPIO 7 (CE1) para el barómetro.

Para comprobar el correcto funcionamiento de las comunicaciones con la IMU se accede al registro *WHOAMI* usando los bloques SPI de la librería raspberry. Es el registro 117 y es un registro solo de lectura del cual tenemos que leer 0x71.

Para la obtención de medidas hay que llevar a cabo una inicialización del sensor que se realiza en el bloque INICIALIZACION. La inicialización se puede ver en la Figura 32. Este bloque solo se ejecuta una vez y una vez realizada se entra en el bloque READ IMU donde se lee la información útil de la IMU.

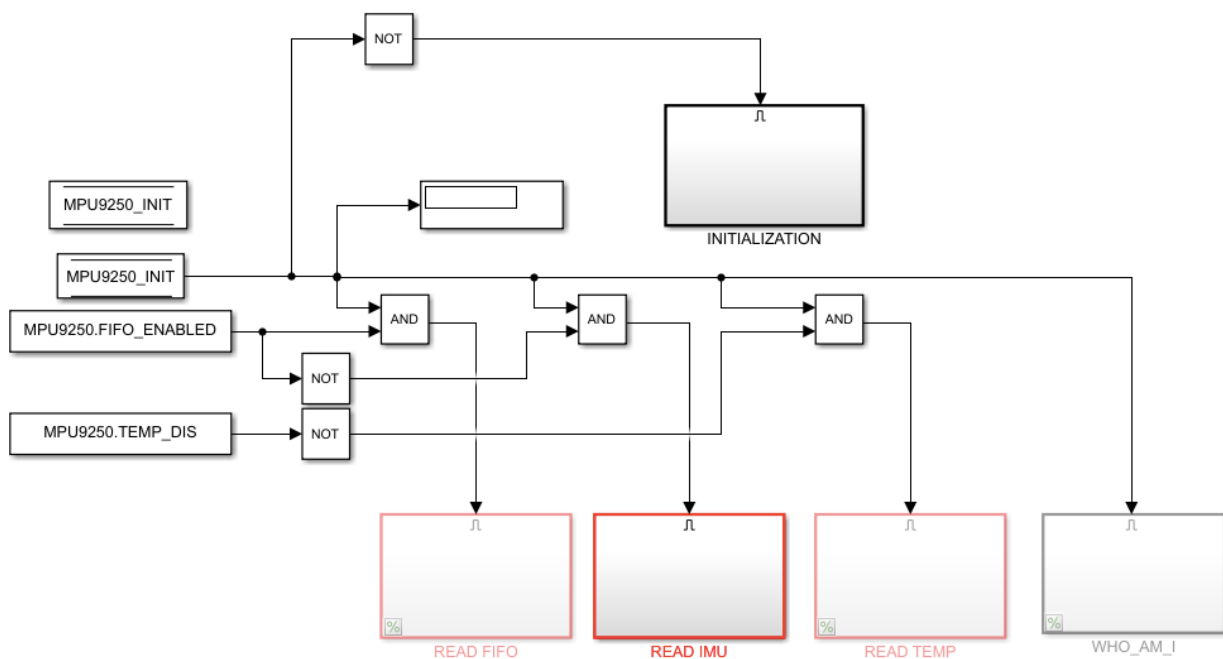


Figura 31. Lectura datos IMU

```

MPU9250.PWR_MGMT_1_BYTE_0 = hex2dec('80');
% Clock Source & Temperature enable
MPU9250.PWR_MGMT_1_BYTE_1 = bin2dec(['0000' num2str(MPU9250.TEMP_DIS)
MPU9250.CLKSEL]);
MPU9250.PWR_MGMT_2_BYTE = bin2dec('00000000');
% CONFIGURATION
MPU9250.CONFIG_BYTE = bin2dec(['00000' MPU9250.DLPF_CFG]);
MPU9250.GYRO_CONFIG_BYTE = bin2dec(['000' MPU9250.FS_SEL '000']);
MPU9250.ACCEL_CONFIG_BYTE = bin2dec(['000' MPU9250.AFS_SEL '000']);
MPU9250.ACCEL_CONFIG_2_BYTE = bin2dec('00000000');
MPU9250.INT_PIN_CFG_BYTE_0 = bin2dec('00010010');
MPU9250.USER_CTRL_BYTE = bin2dec(['0' num2str([MPU9250.FIFO_ENABLED
MPU9250.I2C_MST_EN MPU9250.I2C_IF_DIS]) '0000']);
MPU9250.I2C_MST_CTRL_BYTE = hex2dec('0D');
AK8963_ADDRESS = bin2dec('00001100');

```

Figura 32. Inicialización IMU

4.5 Magnetómetro

El módulo HMC5883 se comunica única y exclusivamente por I2C a frecuencias de 100Hz a 400Hz. Se usan por lo tanto bloques de simulink de la librería de raspberry para la lectura y escritura en I2C.

Este sensor tiene 13 registros y cada registro es de 8 bits. Los tres primeros registros (posiciones de memoria 0x00, 0x01 y 0x02) son registros lectura/escritura y sirven para configurar el magnetómetro. Los siguientes seis registros (posiciones 0x03 – 0x08) son registros de lectura que nos dan la información del campo magnético en los ejes X, Y, Z. Los últimos 4 registros son de lectura no se trabaja con ellos en este proyecto.

Nada más iniciarse el programa se manda la información necesaria a los tres primeros registros para configurar el magnetómetro. En este proyecto se ha configurado para tener una velocidad de salida de datos de 15Hz y una ganancia de 390 en modo de lectura continua de datos. Esto solo se ejecuta una vez.

Una vez configurado se leen los registros (0x03 – 0x08) con la información del campo magnético. Se procesa esta información uniendo los registros 0x03 y 0x04 en un int16 con el dato final del campo magnético en el eje X, los registros 0x05 y 0x06 en otro int16 con el dato final del campo magnético en el eje Z y por último, unir los registros 0x05 y 0x06 en otro int16 con el dato final del campo magnético en el eje Y.

Dirección del registro	Tipo	Función	Valor a mandar
00	lectura/escritura	Configuración velocidad salida de datos	0x70
01	lectura/escritura	Configuración ganancia	0xA0
02	lectura/escritura	Configuración modo	0x00
03	lectura	Eje X MSB	-
04	lectura	Eje X LSB	-
05	lectura	Eje Z MSB	-
06	lectura	Eje Z LSB	-
07	lectura	Eje Y MSB	-
08	lectura	Eje Y LSB	-

Tabla 7. Registros magnetómetro

4.6 Mavlink

Mavlink es un protocolo de comunicaciones para comunicarse con drones. En este proyecto se usan las comunicaciones Mavlink 1. El protocolo Mavlink dispone de distintos mensajes para comunicarse con el UAV aportando cada uno información sobre distintos sensores o información necesaria para la misión. [11]

En este proyecto se van a usar los mensajes propios para IMU, el barómetro y el GPS. Para la IMU se usa el mensaje #30 llamado ATTITUDE. En este mensaje se transmite la información referente a la IMU. La información que envía este mensaje se muestra en la Tabla 9.

Para el barómetro se usa el mensaje #29 llamado SCALED_PRESSURE. En el se envía la información del barómetro. Los campos que usa el mensaje se muestran en la Tabla 8.

Por último, para el GPS se usa el mensaje #24. La información con los campos de este mensaje se muestra en la Tabla 10.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
press_abs	float	hPa	Absolute pressure
press_diff	float	hPa	Differential pressure 1
temperature	int16_t	cdegC	Temperature

Tabla 8. Mavlink #29 SCALED_PRESSURE [12]

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
roll	float	rad	Roll angle (-pi..+pi)
pitch	float	rad	Pitch angle (-pi..+pi)
yaw	float	rad	Yaw angle (-pi..+pi)
rollspeed	float	rad/s	Roll angular speed
pitchspeed	float	rad/s	Pitch angular speed
yawspeed	float	rad/s	Yaw angular speed

Tabla 9. Mavlink #30 ATTITUDE [12]

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp
fix_type	uint8_t		GPS_FIX_TYPE	GPS fix type.
lat	int32_t	degE7		Latitude
lon	int32_t	degE7		Longitude
alt	int32_t	mm		Altitude
eph	uint16_t			GPS HDOP
epv	uint16_t			GPS VDOP
vel	uint16_t	cm/s		GPS ground speed. If unknown, set to: UINT16_MAX
cog	uint16_t	cdeg		Course over ground
satellites_visible	uint8_t			Number of satellites
alt_ellipsoid **	int32_t	mm		Altitude

h_acc **	uint32_t	mm		Position uncertainty.
v_acc **	uint32_t	mm		Altitude uncertainty.
vel_acc **	uint32_t	mm		Speed uncertainty.
hdg_acc **	uint32_t	degE5		

Tabla 10. Mavlink #24 GPS_RAW_INT [12]

Capítulo 5. Conclusiones y resultados

A lo largo de este capítulo se expondrán brevemente los resultados de los principales ensayos realizados a lo largo de este proyecto.

5.1 Ensayo con banco de pruebas para motores y ESCs

El objetivo de este ensayo es comprobar el correcto funcionamiento de los motores y los ESCs y obtener información sobre el peso que es capaz de levantar el dron. En este ensayo se descarto el uso de uno de los motores y dos ESC que mostraban problemas.

PWM	FR (g)	BR (g)	FL (g)	BL (g)
1000	0	0	0	0
1100	19	8	26	19
1200	43	32	45	40
1300	69	65	72	66
1400	106	99	106	98
1500	145	132	132	133
1600	180	165	170	167
1700	227	204	200	202
1800	259	233	226	240
1900	299	271	256	275
2000	328	314	298	309

Tabla 11. Ensayo motores y ESC

5.2 Ensayo movimiento de motores con PCA9685

El objetivo de este ensayo es comprobar el correcto funcionamiento de los motores y las comunicaciones con la PCA9685. Para ello se moverán los motores uno a uno igual que en el ensayo anterior pero esta vez no se usa un banco de pruebas. Los motores estarán ya puestos en el dron y la señal PWM se envia a través del módulo PCA9685 que recibirá la instrucción de mandar un PWM determinado de la raspberry por medio de las comunicaciones I2C. Para hacer este ensayo se usa el Simulink de la Figura 33 para mandar las instrucciones.

Hay 4 bloques tipo *Constante* con un valor de 1 y 4 bloques *Slider Gain* que dan una ganancia entre 1000 y 2000. Se dejan todas las *Slider Gain* en 1000 correspondiéndose con un PWM enviado de 1000. Con ese valor los motores no se mueven. Se sube el valor de una de las *Slider Gain* a 1100 y comprobamos que el motor correspondiente empieza a girar. Se repite el proceso para cada uno de los motores y se comprueba que todos funcionan correctamente. Este ensayo, a diferencia del ensayo de motores y ESCs se realiza sin poner las hélices ya que no nos interesa ver que fuerza genera cada motor sino comprobar que las conexiones entre raspberry y PCA9685 es correcta y que los motores se arman correctamente.

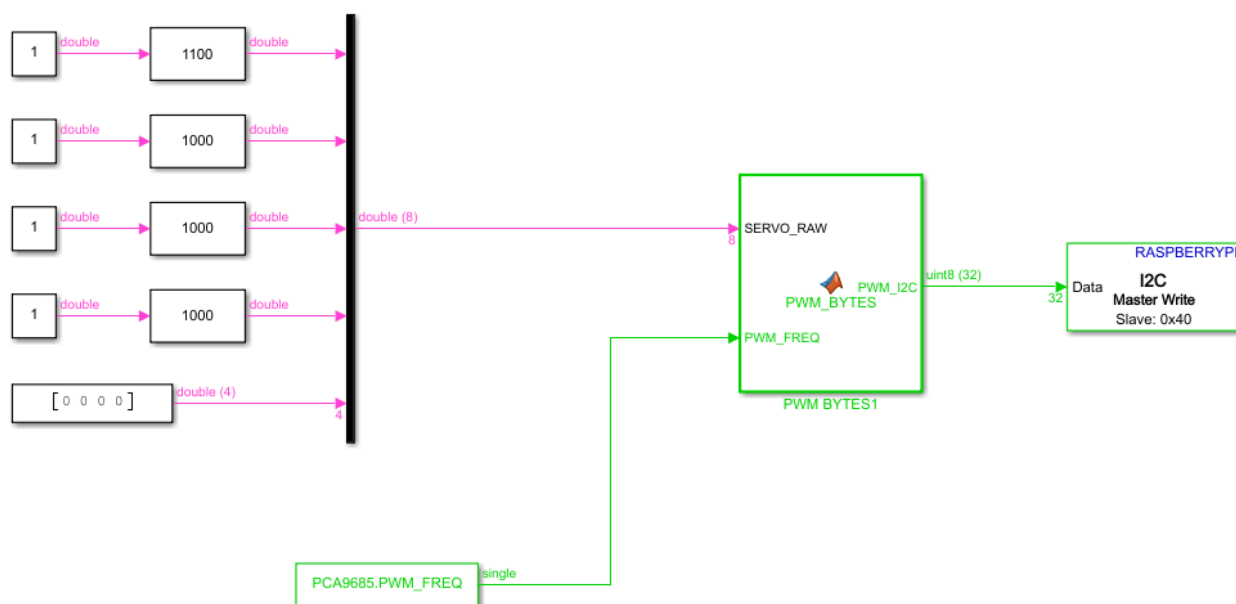


Figura 33. Ensayo PCA9685

5.3 Ensayo IMU

En este ensayo se comprueba el correcto funcionamiento de la IMU. Para ello la IMU sufre aceleraciones en los ejes X, Y y Z obteniéndose las gráficas de la Figura 34. La primera gráfica se corresponde con el eje X, la segunda con el eje Y y la tercera con el eje Z. Cada vez que se observa un pico es porque se le ha dado una aceleración en ese eje. Para mirar el sentido de la aceleración miramos el primer pico. El segundo se debe a la deceleración que sufre la IMU al parar. Esto sirve también para comprobar que los ejes del dron se corresponden con los ejes de la IMU y en caso de no ser así reorientar los ejes de la IMU creando las matrices descritas en el apartado 3.6.

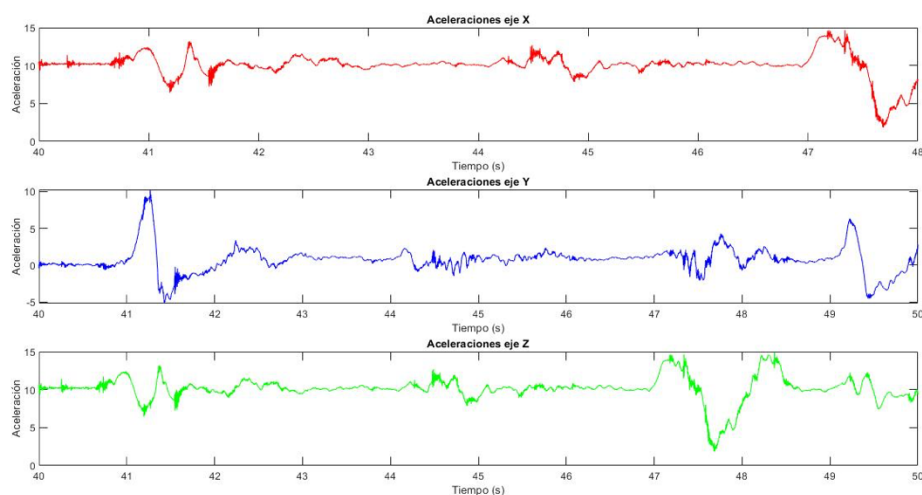


Figura 34. Medidas acelerómetro

Para comprobar el gir6scopo el proceso a seguir es el mismo que con el aceler6metro. En la Figura 35 se muestran los valores del ensayo del gir6scopo. Al igual que con el aceler6metro se aprovecha este ensayo para comprobar que los ejes de la IMU est6n orientados como los del dron.

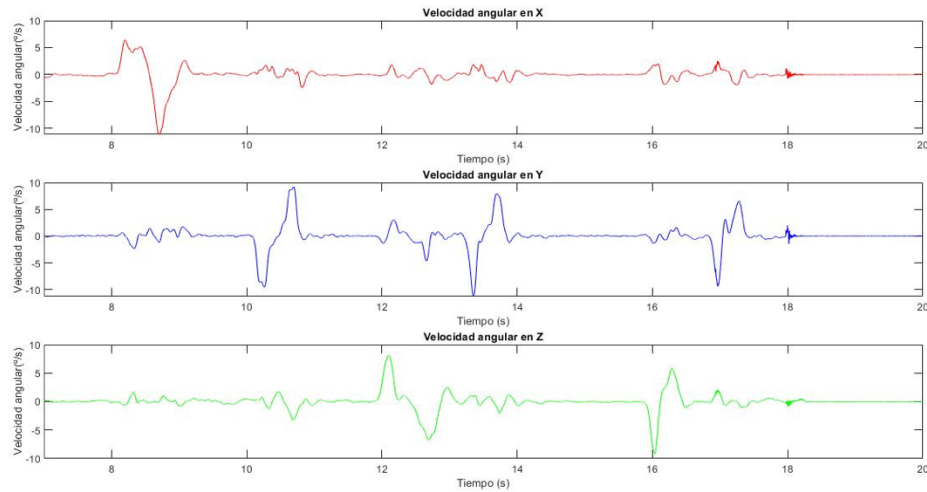


Figura 35. Medidas gir6scopo

5.4 Ensayo bar6metro

Para este ensayo se subi6 desde la planta 0 del edificio de ICAI de la Universidad Pontificia de Comillas hasta la planta 5 y se observ6 una variaci6n de presi6n. Se monitorizaron los resultados y quedan reflejados en la Figura 36.

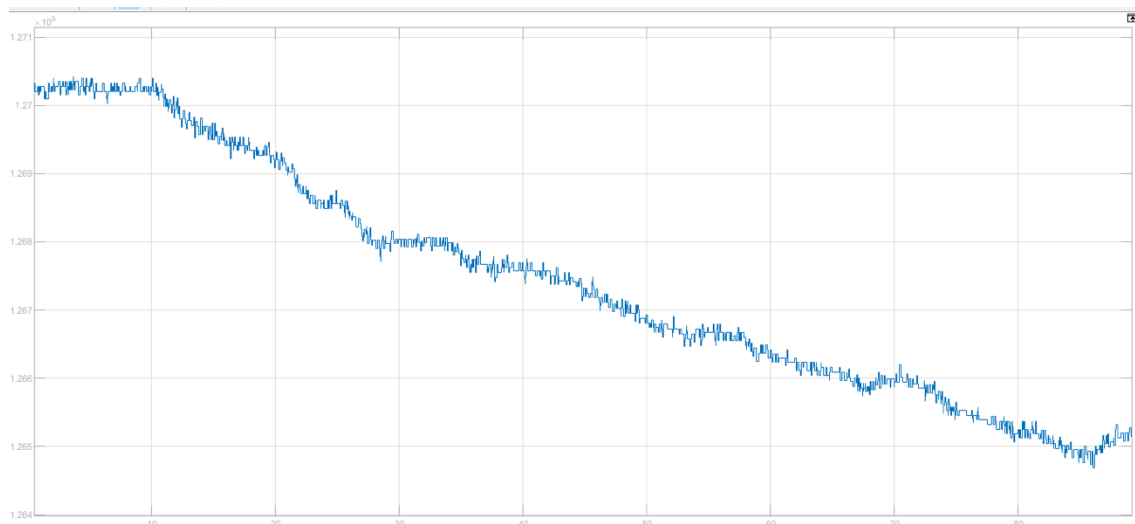


Figura 36. Ensayo bar6metro

5.5 Ensayo GPS Ublox 6M

En este ensayo se comprueba que la configuración, recepción y filtrado de la información del GPS es correcta. En la Figura 37 se muestra información de las coordenadas en el sistema WGS 84. En la Figura 38 se muestra información de las coordenadas en sistemas ECEF. En la Figura 39 se muestran las velocidades en sistema ECEF.

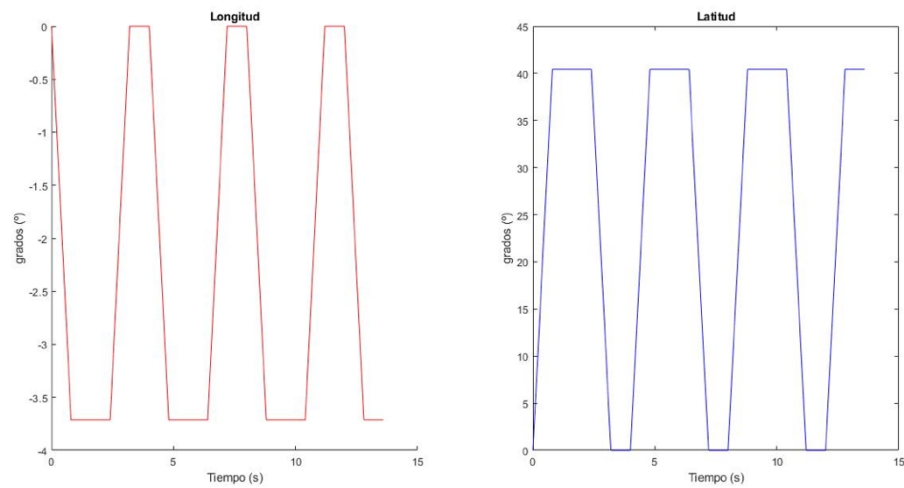


Figura 37. Longitud y latitud GPS

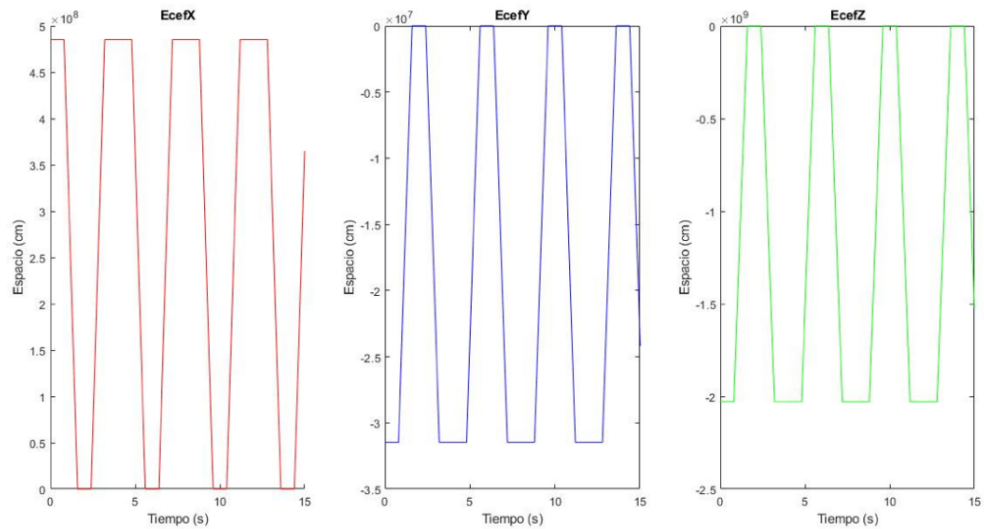


Figura 38. Coordenadas sistema ECEF GPS

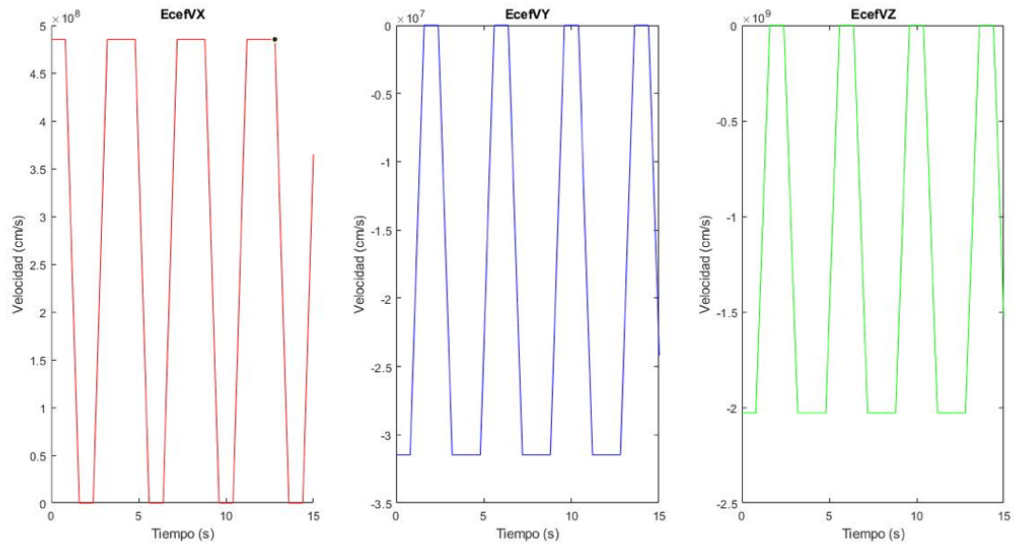


Figura 39. Velocidades sistemas ECEF GPS

5.6 Ensayo Mavlink

El último ensayo que se realiza es el de mandar la información de los mensajes usando el protocolo Mavlink. El primer objetivo de este ensayo es comprobar el correcto funcionamiento de las comunicaciones entre UAV y base terrestre. El segundo objetivo es comprobar que se codifica correctamente el mensaje con la información en el UAV y se decodifica correctamente en la base terrestre. El sentido de envío de los mensajes es unilateral, es decir, los mensajes son transmitidos desde el dron y recibidos por la base terrestre. Con esta información se puede monitorizar los sensores del dron desde la base terrestre.



Figura 40. Comunicaciones Mavlink

Capítulo 6. Conclusiones y futuros desarrollos

En este capítulo se exponen las conclusiones y los futuros desarrollos del proyecto.

6.1 Conclusiones

En los ensayos de motores tanto en el banco de pruebas como usando el controlador de PWM PCA9685 se obtuvieron unos resultados muy satisfactorios. Los motores respondían correctamente a las indicaciones y la fuerza de estos es perfecta para nuestro dron.

En los ensayos usando sensores también se obtuvieron resultados satisfactorios. Cabe mencionar que en las gráficas de coordenadas del GPS se observa que las señales no son continuas. Esto se debe a los tiempos de espera de llegada de información al buffer del GPS. Para ensayos posteriores se filtró la información del GPS obteniendo gráficas continuas logrando una mejor monitorización. Lamentablemente, no fue posible la integración del magnetómetro en el dron. No se logró recibir información válida y se siguen desconociendo los motivos.

Por último, se logró enviar información del UAV a la base terrestre usando el protocolo de comunicaciones Mavlink.

6.2 Futuros desarrollos

A continuación, se exponen los posibles futuros desarrollos de este proyecto. Una vez el funcionamiento del hardware sea óptimo el próximo paso es implantar el sistema de navegación autónoma por GPS usando waypoints. Este sistema se desarrolló en fases anteriores del proyecto por lo que simplemente habría que adaptarlo.

Una vez implantada la navegación autónoma se pueden incorporar nuevos sensores al dron con el objetivo de tener información más precisa y lograr así una navegación más fiable. Entre los sensores que se le pueden incorporar al UAV sería especialmente útil el uso de un lidar. Como ya se explicó en el apartado 2.2.2 el lidar es un sensor que nos permite detectar y medir distancia a objetos. Combinar este sensor con el uso del GPS nos permite crear un sistema de vuelo autónomo mucho más fiable. La navegación se seguiría realizando por waypoints usando el GPS pero gracias a la tecnología del lidar se puede realizar un mapeado del terreno y detectar si hay obstáculos en medio de la ruta evitando así la colisión del UAV con cualquier objeto. Con esta tecnología se pueden generar rutas alternativas a la original rodeando los obstáculos y además dejar constancia de estos obstáculos en un mapa usando las coordenadas que da el GPS.

Bibliografía

- [J. Loring Castillo, «Navegación autónoma de un cuadricóptero en exteriores,» Madrid,
1 Septiembre 2018.
]
- [M. Mazur, A. Wisniewski y J. McMillan, «Clarity from above, PwC global report on
2 the commercial applications of drone technology,» Mayo 2016.
]
- [S. Nahal, B. Ma y F. Tran , «Feed the World: global food security primer,» Marzo
3 2015.
]
- [M. S. Greawal, «Global Positioning Systems, Inertial Navigation, and Integration,»
4 John Wiley & Sons, Inc., 2001.
]
- [N. González García, «Control de un cuadricóptero para navegación en interiores
5 usando un sensor de flujo optico,» Madrid.
]
- [I. Fernández, «ATENEA: Advanced techniques for deeply integrated
6 GNSS/INS/LiDAR navigation,» 2010.
]
- [«Raspberrypi,» [En línea]. Available:
7 <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>. [Último acceso: 1 Julio
] 2019].
- [«Luis Llamas,» 24 Noviembre 2016. [En línea]. Available:
8 [https://www.luisllamas.es/controlar-16-servos-o-16-salidas-pwm-en-arduino-con-](https://www.luisllamas.es/controlar-16-servos-o-16-salidas-pwm-en-arduino-con-pca9685/)
] [pca9685/](https://www.luisllamas.es/controlar-16-servos-o-16-salidas-pwm-en-arduino-con-pca9685/). [Último acceso: 23 Junio 2019].
- [«Waveshare,» [En línea]. Available: [https://www.waveshare.com/serial-expansion-](https://www.waveshare.com/serial-expansion-hat.htm)
9 [hat.htm](https://www.waveshare.com/serial-expansion-hat.htm) . [Último acceso: 1 Julio 2019].
]
- [L. Sanchez-Quiñones Campuzano, «Control de un cuadricóptero para seguimiento
1 automático de personas,» Madrid, Julio 2017.
0
]
- [«Mavlink,» [En línea]. Available: <https://mavlink.io/en/>. [Último acceso: 7 Julio
1 2019].
1
]
- [«QuantDare,» [En línea]. Available: <https://quantdare.com/filtro-kalman/>. [Último
1 acceso: 4 Junio 2019].
2
]
- [«Asidek,» [En línea]. Available: [https://www.asidek.es/bim-autodesk-](https://www.asidek.es/bim-autodesk-infraestructuras/)
1 [infraestructuras/](https://www.asidek.es/bim-autodesk-infraestructuras/) . [Último acceso: 1 Julio 2019].
3
]

[«Work with drone,» [En línea]. Available: <https://workwithdrone.com/es/8-con-nuestro-vehiculos-aereos-no-tripulados-drones-o-rpas-trabajamos-para-optimizar-tu-plantacion-controles-de-plagas-estres-hidrico-y-nutricional-enfermedades-y-plagas.html>. [Último acceso: 1 Julio 2019].

[taroz1461, «YouTube,» [En línea]. Available: <https://www.youtube.com/watch?v=cbczfgH1x0s>. [Último acceso: 1 Julio 2019].

[J. R. Megías, «Desarrollo de una plataforma de control basada en raspberry pi para la implementación y ensayo del control de orientación mediante soporte de pruebas indoor,» 2016.

[K. Yakovlev, «Distributed control and navigation system for quadrotor UAVs in GPS-denied environments».

[J. N. Lastra, «Diseño de un drone programable de bajo coste,» Octubre 2017.

[R. E. Kalman, Topics in Mathematical System Theory, 1969.

[Á. S. Ramírez, «El filtro de Kalman,» Julio 2003.

[«Dronemania,» [En línea]. Available: <https://dronemania.es/10-cosas-que-tienes-que-hacer-despues-de-construir-tu-drone-de-carreras/>. [Último acceso: 1 Julio 2019].

[G. Ferrer Mínguez, «Integración Kalman de sensores inerciales INS con GPS en un UAV,» 2009.

[«Google,» [En línea]. Available: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjItq_IzfjAhWQlxQKHeecDxIQjRx6BAGBEAU&url=https%3A%2F%2Fsiglibleuruguay.wordpress.com%2F2015%2F12%2F23%2Fsistemas-de-referencia-de-coordenadas%2F&psig=AOvVaw2NutN6YylhaIwK0aKrwKF. [Último acceso: 2 Julio 2019].

[«Mavlink,» [En línea]. Available: <https://mavlink.io/en/messages/common.html>. [Último acceso: 4 Julio 2019].

PARTE 2 – PRESUPUESTO

Capítulo 1. Mediciones

1.1 Componentes de hardware

<i>Componente</i>	<i>Cantidad</i>
<i>Raspberry pi Zero W</i>	1
<i>Waveshare serial expansión HAT</i>	1
<i>Ublox Neo 6M</i>	1
<i>Motores Emax MT2204-2300Hw</i>	4
<i>MS5611 (barómetro)</i>	1
<i>MPU-92.65 (IMU)</i>	1
<i>PCA9685 Servotreiber</i>	1
<i>Bateria Lipo</i>	1
<i>Estructura y 4 hélices</i>	1

Tabla 12. Componentes

1.2 Software

<i>Componente</i>	<i>Cantidad</i>
<i>Matlab/Simulink</i>	1
<i>u-center</i>	1
<i>Putty</i>	1

Tabla 13. Software

1.3 Herramientas y equipos

<i>Elementos</i>	<i>Cantidad</i>
<i>Ordenador</i>	2
<i>Cargador batería Lipo</i>	1
<i>Programador</i>	1
<i>Soldador</i>	1

Tabla 14. Herramientas y equipo

1.4 Mano de obra directa

<i>Nombre</i>	<i>Horas</i>
<i>Estado del Arte</i>	20
<i>Configuración Sensores</i>	20
<i>Configuración GPS</i>	30
<i>Montaje y soldadura</i>	20
<i>Diseño vuelo guiado</i>	20
<i>Redacción memoria</i>	10

Tabla 15. Mano de obra

Capítulo 2. Precios unitarios

2.1 Hardware

<i>Componente</i>	<i>€/unidad</i>
<i>Raspberry pi Zero W</i>	10.53
<i>Waveshare serial expansión HAT</i>	12.99
<i>Ublox Neo 6M</i>	17.00
<i>Motores Emax MT2204-2300Hw</i>	13.23
<i>MS5611 (barómetro)</i>	3.52
<i>MPU-92.65 (IMU)</i>	1.24
<i>PCA9685 Servotreiber</i>	6.59
<i>Bateria Lipo</i>	14.39
<i>Estructura y 4 hélices</i>	-

Tabla 16. Precio unitario componentes

2.2 Software

<i>Componente</i>	<i>€/unidad</i>
<i>Matlab/Simulink</i>	700.00
<i>u-center</i>	-
<i>Putty</i>	-

Tabla 17. Precio unitario software

2.3 Herramientas y equipo

<i>Elementos</i>	<i>€/unidad</i>
<i>Ordenador</i>	900.00
<i>Cargador batería Lipo</i>	27.26
<i>Programador</i>	4.51
<i>Soldador</i>	24.99

Tabla 18. Precio unitario herramientas y equipo

2.4 Mano de obra

<i>Nombre</i>	<i>Horas</i>
<i>Estado del Arte</i>	20
<i>Configuración Sensores</i>	20
<i>Configuración GPS</i>	30
<i>Montaje y soldadura</i>	20
<i>Diseño vuelo guiado</i>	20
<i>Redacción memoria</i>	10

Tabla 19. Precio unitario mano de obra

Capítulo 3. Sumas parciales

3.1 Hardware

<i>Componente</i>	<i>unidades</i>	<i>€/unidad</i>	<i>Total (€)</i>
<i>Raspberry pi Zero W</i>	1	10.53	10.53
<i>Waveshare serial expansión HAT</i>	1	12.99	12.99
<i>Ublox Neo 6M</i>	1	17.00	17.00
<i>Motores Emax MT2204-2300Hw</i>	4	13.23	52.92
<i>MS5611 (barómetro)</i>	1	3.52	3.52
<i>MPU-92.50 (IMU)</i>	1	1.24	1.24
<i>PCA9685 Servotreiber</i>	1	6.59	6.59
<i>Bateria Lipo</i>	1	14.39	14.39
<i>Estructura y 4 hélices</i>	1	-	-
		TOTAL	119.19 €

3.2 Software

Tabla 20. Precio total componentes

<i>Componente</i>	<i>unidades</i>	<i>€/unidad</i>	<i>Total (€)</i>
<i>Matlab/Simulink</i>	1	700.00	700.00
<i>u-center</i>	1	-	-
<i>Putty</i>	1	-	-
		TOTAL	700.00 €

Tabla 21. Precio total software

3.3 Herramientas y equipo

<i>Elementos</i>	<i>unidades</i>	<i>€/unidad</i>	<i>Total (€)</i>
<i>Ordenador</i>	2	900.00	1800
<i>Cargador batería Lipo</i>	1	27.26	27.26
<i>Programador</i>	1	4.51	4.51
<i>Soldador</i>	1	24.99	24.99
TOTAL			1856.76 €

Tabla 22. Precio total herramientas y equipo

3.4 Mano de obra

<i>Nombre</i>	<i>Horas</i>	<i>€/hora</i>	<i>Total (€)</i>
<i>Estado del Arte</i>	20	15	300
<i>Configuración Sensores</i>	40	15	600
<i>Configuración GPS</i>	40	15	600
<i>Montaje y soldadura</i>	20	15	300
<i>Diseño vuelo guiado</i>	20	15	300
<i>Redacción memoria</i>	20	15	300
<i>Documentación</i>	20	15	300
TOTAL			2700 €

Tabla 23. Precio total mano de obra

Capítulo 4. Presupuesto general

<i>Nombre</i>	<i>Coste parcial (€)</i>
<i>Componentes hardware</i>	119.19 €
<i>Software</i>	700.00 €
<i>Herramientas y equipos</i>	1856.76 €
<i>Mano de obra directa</i>	2700.00 €
<i>TOTAL</i>	5375.95 €

Tabla 24. Presupuesto general

Anexo A: Información componentes hardware

En este anexo se incluyen los enlaces a los componen que se han usado en el proyecto.

Raspberry pi zero W:

- https://cdn.sparkfun.com/assets/learn_tutorials/6/7/6/PiZero_1.pdf

IMU MPU-9250:

- <https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>

Barómetro MS6211:

- https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5611-01BA03%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5611-01BA03_B3.pdf%7FCAT-BLPS0036

GPS Ublox Neo 6M:

- https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf
- https://www.u-blox.com/sites/default/files/products/documents/u-blox6_ReceiverDescrProtSpec_%28GPS.G6-SW-10018%29_Public.pdf?utm_source=en%2Fimages%2Fdownloads%2FProductDocs%2Fu-blox6_ReceiverDescriptionProtocolSpec_%28GPS.G6-SW-10018%29.pdf
- https://www.u-blox.com/sites/default/files/u-center_UserGuide_%28UBX-13005250%29.pdf

Magnetómetro HMC5883:

- https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf

PCA9685:

- <https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf>

Anexo B: Filtro de kalman extendido (EKF)

El filtro de kalman es un algoritmo imprescindible a la hora de crear sistemas de navegación tanto asistidos como autónomos. Tiene su origen el trabajo de Rudolpf Emil Kalman (1960). El objetivo de este algoritmo es estimar los estados de variables no observables a partir de variables observables del sistema. El algoritmo hace uso de métodos de estimación por mínimos cuadrados recursivos.

El algoritmo consta de dos tipos de ecuaciones. Las ecuaciones principales (relacionan variables de estado y variables observables) y las ecuaciones de estado (definen la forma de las variables de estado). [1] Con estas ecuaciones se estiman las variables de estado, este proceso consta de dos pasos: predicción y corrección. En la etapa de predicción se estiman los estados de las variables a partir de los estados anteriores. En la etapa corrección, se mejora esta estimación haciendo uso de variables observables (como la información que obtenemos de los sensores) del sistema consiguiendo un resultado más fiable.

El filtro de kalman esta aplicado a sistemas lineales¹ y gaussianos². No obstante, la mayor parte de las aplicaciones reales siguen sistemas no lineales y/o no gaussianos. En este tipo de aplicaciones se usa el filtro extendido de kalman (EKF).

Para el vuelo de cuadricopteros se usa el EKF. En la etapa de predicción usaremos los estados anteriores del dron para predecir sus próximas variables de estados. En la etapa de corrección usaremos la información de los sensores para rectificar las variables de estado calculas en la etapa anterior. Usaremos información de la IMU (acelerómetros, giróscopos), del barómetro, del magnetómetro y información de posición del GPS.

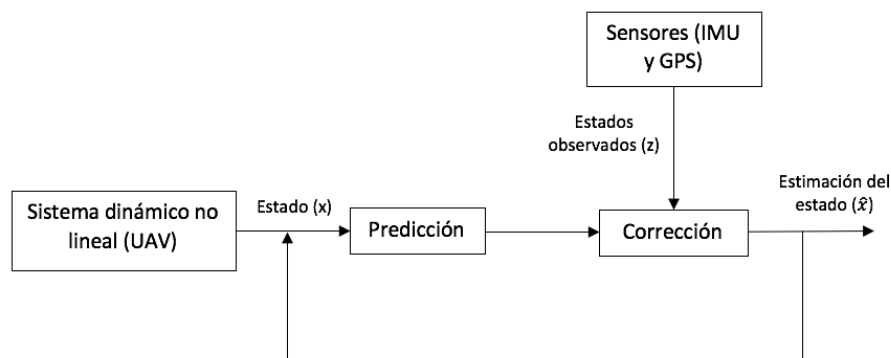


Figura 41. Esquema EKF

¹ Sistema lineal significa que todas sus funciones son lineales en x .

² Sistema gaussiano significa que todo el ruido es gaussiano.

Anexo C: Montaje

Para realizar el montaje del dron primero hay que calcular los pines que se van a necesitar. Para ello se estudian los periféricos (GPS, barómetro, IMU, magnetómetro, emisora y controlador de servos) uno a uno y observándose que tipo de comunicaciones se sabrá que pines van a necesitar. En la Tabla 25 se ha hecho un estudio de los distintos periféricos y los pines que utilizan.

		GPS	Emisora	IMU	Barómetro	Servo	Magnetómetro	Total
Alimentación	3,3V	1	1	1	1	1	1	6
	GND	1	1	1	1	1	1	6
UART	TX	1	1	0	0	0	0	2
	RX	1	1	0	0	0	0	2
I2C	SCL	0	0	0	0	1	1	2
	SDA	0	0	0	0	1	1	2
SPI	MOSI	0	0	1	1	0	1	2
	MISO	0	0	1	1	0	1	2
	SLK	0	0	1	1	0	1	2
	C0/C1	0	0	1	1	0	1	2

Tabla 25. Recuento de pines periféricos

Se soldarán cables tipo hembra a los pines correspondientes de la raspberry y cables tipo macho a los periféricos correspondientes. Se suelda a un conector con el mismo numero de pines y que luego va puesto en la raspberry. Se hace lo mismo para los periféricos. El objetivo de esto es por un lado mayor comodidad a la hora de soldar, que en caso de cometerse algún error en la soldadura no se dañe ningún componente y que luego se puedan reaprovechar los componentes para otras cosas. Los pines disponibles en la raspberry se muestran en la Figura 42.



Figura 42. Pinout raspberry pi zero

En la raspberry solo hay pines para sacar 1 Rx y 1 Tx. Para el otro Rx y Tx se usan los pines adicionales que nos proporciona el waveshare serial expansión HAT. Por otro lado, solo hay un 1 SCL y 1 SDA pero esto no es ningún problema ya que de la raspberry iremos al primer periférico (en nuestro caso el controlador de servos) y de ese mismo periférico iremos al segundo (el barómetro). Ocurre lo mismo para las conexiones SPI, tendremos que hacer cables dobles para el MOSI, MISO y para el SCLK.