

**IMPLEMENTACIÓN DE NUEVAS TECNOLOGÍAS PARA EL DESARROLLO DE
SISTEMAS INTELIGENTES APLICADOS A LA AGRICULTURA**

Pasantía de investigación

Moisés David Bautista Robles

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Programa Ingeniería Mecatrónica

Pamplona , 2017

Agradecimientos

Agradezco a Dios por todo. Le agradezco a mi madre y mi padre por su apoyo incondicional. Gracias a la profesora Yara Oviedo por su asesoría y tutoría para ayudarme a terminar este proceso, al profesor Jorge Sequeda que hizo esta pasantía posible, bajo su tutoría en el ISER. Gracias también a los libros, que rompen cualquier barrera.

Tabla de contenido

Agradecimientos	2
Tabla de contenido.....	3
Lista de figuras	7
Lista de tablas.....	10
Resumen.....	11
Capítulo 1	12
Introducción	12
Justificación	13
Objetivos	14
Objetivo general.....	14
Objetivos específicos.....	14
Marco teórico.....	15
Agricultura de precisión	15
Clasificación de los drones	18
Costo de los drones.....	19
Fuente de energía	20
Tiempo de vuelo y rango.....	20
Facilidad de uso.....	21
¿Cómo vuela el dron?.....	21
Ascenso	22
Marco o chasis de un dron	23
Motor eléctrico	23
Motor de corriente directa.....	23
Motor <i>brushless</i> o sin escobillas.....	23
Motores Air Gear 350.....	24
Hélice.....	24
Hélices en drones	24
Controlador de vuelo o unidad de control	26
Acelerómetro	26
Giroscopio	26
Barómetro	26
Brújula	26
Magnetómetro	26
GPS	26
Controladora de vuelo Navio2	27
Joystick	28
Raspberry Pi	28
El <i>hardware</i> de la Raspberry Pi2.....	30
Memoria de almacenamiento SD.....	31
Redes de área local	33

WLAN: redes locales inalámbricas	33
Routers	34
Dirección IP	34
Protocolo UDP	34
Protocolo SSH.....	35
Estación terrestre de control o GCS	35
Ardupilot	36
Sistema operativo de robot (ROS).....	36
Microcontrolador	38
Microcontrolador Arduino	39
Arduino Mega.....	40
Sensores	40
Sensor de temperatura y humedad DHT22.....	40
Sensor de humedad del suelo	41
Sensor de gotas de lluvia.....	42
Sensor de luminosidad BH1750.....	43
Sensor de calidad del aire	44
PRIMER PROTOTIPO: DRON GRANJERO	45
Capítulo 2	46
Construyendo el dron cuadricóptero	46
Motores y batería.....	47
Diseño de una carcasa protectora para las tarjetas Navio2 y Raspberry Pi 2	50
Configuración de hardware de la tarjeta de navegación Navio2	53
Sujetar la tarjeta Navio2 en una Raspberry Pi.....	54
Alimentación de la tarjeta Navio2	55
Antena GNSS	56
Los ESC	56
Módem para telemetría	57
Protección UV para el barómetro	57
Colores del led de Navio2.....	58
Diseño de una base elevada para el módulo de GPS	59
Capítulo 3	62
Configuración de la Raspberry Pi	62
Montando la imagen del SO a la tarjeta SD.....	62
Configurando el acceso a Wi-Fi	63
Editar la configuración en la tarjeta SD	66
Usando un monitor y teclado.....	66
Usando Ethernet (Recomendado).....	66
Actualizando.....	69
Instalación y funcionamiento	69
Systemd.....	71
Escogiendo el vehículo, la versión y el tipo de tarjeta	71
Especificando las opciones de ejecución.....	72

Recargar la configuración	73
Activando el ArduPilot.....	73
Conectando la Estación de Control Terrestre GCS	74
Mission Planner (Recomendado).	74
QGroundControl.....	75
APM Planner.....	75
MAVProxy.....	75
Calibración del hardware	76
Calibración de brújulas.....	76
Calibración Onboard.....	76
Calibración en vivo.	77
Calibración del acelerómetro	78
Calibración de los controladores electrónicos de velocidad (ESC)	79
Probando los motores	80
Medición de voltaje y corriente	82
Modos de vuelo.....	83
Configurando el ROS (sistema operativo del robot).....	84
El comando <i>tmux</i>	85
Ejecutando el roscore.....	87
Corriendo el Ardupilot.....	87
Enlazando el ROS a la estación terrestre.....	88
Capítulo 4	90
Configuración del mando manual	90
Probando los controles antes del vuelo	92
Capítulo 5	94
Evaluación de vuelo.....	94
Delimitación	95
Vuelo manual	96
Piloto automático.....	98
SEGUNDO PROTOTIPO: ESTACIÓN METEOROLÓGICA	101
Capítulo 6	102
Diseño electrónico del prototipo	102
Configuración de <i>hardware</i> de los componentes	103
Programación del código.....	112
Capítulo 7	124
Construcción de la estación meteorológica móvil	124
Capítulo 8	131
Evaluando la estación meteorológica	131
Conclusiones	135
Referencias.....	137
Apéndice A: Código completo de la estación meteorológica.....	144

Apéndice B: Medidas de los diseños	155
Apéndice C: Importar datos desde un archivo CSV a Excel	166

Lista de figuras

Figura 1. Delta drone del fabricante Kaiser Baas.....	17
Figura 2. Batería Li Po.....	21
Figura 3. Paquete Air Gear 350: incluye 4 motores brushless, 4 reguladores de velocidad y 4 hélices.....	25
Figura 4. Tarjeta controladora de vuelo Navio2.	28
Figura 5. Raspberry Pi 2 modelo B.....	30
Figura 6. Esquema general de ROS en el SO distribuido por Emlid.....	37
Figura 7. Interacción del nodo mavros con el GCS.....	38
Figura 8. Arduino Mega 2560.....	40
Figura 9. Sensor DHT22.....	41
Figura 10. Sensor de humedad del suelo.....	42
Figura 11. Sensor de gotas de lluvia.....	43
Figura 12. Sensor de luminosidad BH1750.....	44
Figura 13. Sensor de gases MQ-5.....	44
Figura 14. Marco para dron Tarot Iron Man 659 TL65B01.....	46
Figura 15. De izquierda a derecha: ensamble del soporte del dron; fijación de un motor en una de las extensiones del dron; el marco ensamblado al soporte y dos extensiones donde van los motores. Los cables de energía conectados y distribuidos por el marco.....	48
Figura 16. De izquierda a derecha: base para la tarjeta de navegación; tarjeta de navegación ensamblada en el marco; conexión del regulador de velocidad con los motores.....	50
Figura 17. Batería LiPo en el marco del dron.....	51
Figura 18. Diseño de la carcasa para las tarjetas Raspberry Pi y Navio2. Parte inferior (izquierda) y parte superior (derecha).....	52
Figura 19. Carcasa impresa y ensamblada con las tarjetas.....	52
Figura 20. Ensamble de la tarjeta Navio2 con Raspberry Pi.....	54
Figura 21. Carcasa fija en la parte superior del marco del dron.....	55
Figura 22. Módulo de energía que se conecta a la batería y provee 5V a la tarjeta Navio2.....	55
Figura 23. Antena GNSS (GPS) conectada a la tarjeta Navio2.....	56
Figura 24. Esquema de conexión de los motores a los pines PWM de la tarjeta Navio2.....	56
Figura 25. Los dos modos de conectar un adaptador de red inalámbrico a Navio2.....	57
Figura 26. Adaptador de red inalámbrica conectada a la tarjeta Raspberry Pi 2 en el dron.....	58
Figura 27. Diseño de la base de la torrecilla de la antena GPS.....	60
Figura 28. Base del GOS que va en la parte superior de la torrecilla.....	60
Figura 29. Ensamble de la torre de la antena GNSS (GPS) en el marco del dron.....	61
Figura 30. Instalando el SO Emlid Raspbian en una micro SD, utilizando Etcher.....	62
Figura 31. Dirección IP de la tarjeta Navio2 en la aplicación Fing.....	67
Figura 32. Accediendo a la tarjeta Raspberry Pi mediante conexión SSH.....	68
Figura 33. Pantalla de inicio después de acceder al SO mediante SSH.....	70
Figura 34. Ejecución del menú de selección del vehículo.....	71

Figura 35. Seleccionando la opción 15 para cuadricóptero y Navio2.....	71
Figura 36. Asignando la IP de la estación terrestre para el Ardupilot.....	72
Figura 37. Entorno de la aplicación para GCS Mission Planner.....	74
Figura 38. Configuración del tipo de conexión y velocidad de la comunicación.....	75
Figura 39. Completando la conexión entre Mission Planner y el dron.....	75
Figura 40. Rotaciones para calibrar el dron. De izquierda derecha, a nivel, lado izquierdo, lado derecho, nariz abajo, nariz arriba, reverso.....	76
Figura 41. Calibración Onboard de las brújulas finalizada.....	77
Figura 42. Calibración en vivo de las brújulas.....	78
Figura 43. Calibración del acelerómetro.....	79
Figura 44. Calibración de los reguladores de velocidad.....	80
Figura 45. Prueba de movimiento de los motores en Mission Planner.....	81
Figura 46. Distribución de los motores, dirección de giro de cada uno (CW: sentido de las manecillas del reloj, CCW: sentido contrario a las manecillas del reloj). El orden de los motores es 1=A, 4=B, 2=C, 3=D.....	82
Figura 47. Configuración para medir voltaje y corriente de la batería.....	83
Figura 48. Modos de vuelo en Mission Planner.....	84
Figura 49. División de la pantalla del SSH en 4.....	86
Figura 50. Ejecutando roscore en el primer panel.....	86
Figura 51. Modificación temporal de la dirección IP del GCS para el Ardupilot.....	87
Figura 52. Estableciendo el enlace mavros entre el ROS y la estación terrestre.....	88
Figura 53. Muestra de datos medidos en el dron luego de establecer el enlace del ROS.....	89
Figura 54. Control de mando USB.....	90
Figura 55. Configuración de botones y palancas del joystick.....	91
Figura 56. Calibración del mando USB como la radio control del dron.....	92
Figura 57. Revisión de parámetros del dron antes del vuelo.....	94
Figura 58. El dron está calibrado y listo para volar.....	95
Figura 59. Delimitación de la altura del dron.....	96
Figura 60. Prueba de vuelo manual.....	97
Figura 61. Creando rutas de vuelo en Mission Planner.....	98
Figura 62. Creando rutas de vuelo más complejas.....	99
Figura 63. Vuelo del piloto automático. De izquierda a derecha: despegue; maniobras en el aire; preparándose para aterrizar; aterrizaje.....	99
Figura 64. Plan de ruta en forma de malla o cuadrícula, para sobrevolar terrenos de cultivo.....	100
Figura 65. Esquema general de la electrónica, los sensores conectados al microcontrolador Arduino Mega; las mediciones se visualizan en la pantalla LCD y se alimenta la Arduino con baterías.....	102
Figura 66. Esquema de conexión del sensor DHT22 al microcontrolador Arduino.....	105
Figura 67. Esquema de conexión del sensor de humedad del suelo.....	106
Figura 68. Esquema de conexión del sensor de gotas de lluvia al microcontrolador Arduino.....	107
Figura 69. Esquema de conexión del sensor de luminosidad BH1750.....	108

Figura 70. Esquema de conexión del sensor de gases MQ-5 al microcontrolador Arduino.....	109
Figura 71. Esquema de conexión del módulo GPS al microcontrolador Arduino.....	110
Figura 72. Esquema de conexión de la pantalla LCD al microcontrolador mediante bus I2C....	111
Figura 73. Esquema de conexión del módulo para micro SD al microcontrolador Arduino.....	111
Figura 74. Pines de conexión ISP (en el ICSP de Arduino.....)	112
Figura 75. De izquierda a derecha: parte superior de la caja de sensores (orificios para sensores); la parte media (Arduino y sensores); la parte inferior (esta contiene la baquelita).....	124
Figura 76. Impresión de una pieza en 3D en pausa.....	125
Figura 77. Piezas impresas en 3D para la estructura de la caja.....	126
Figura 78. Diseño de la tapa frontal de la estructura de la caja.....	127
Figura 79. Diseño de las piezas para la torre de la caja para el módulo GPS.....	127
Figura 80. Baquelita de la estación meteorológica.....	128
Figura 81. Montaje de la baquelita y los sensores.....	128
Figura 82. Panel solar sobre soporte de acero junto a la caja que contiene los sensores.....	129
Figura 83. Estación meteorológica construida.....	130
Figura 84. Inicio exitoso de la estación meteorológica. Significa que el módulo SD/micro SD está conectado y funciona, así como los puertos seriales y la LCD.....	131
Figura 85. Muestra de dos mediciones de todos los sensores.....	132
Figura 86. Muestra en pantalla de las medidas de luminosidad, gases y ubicación del GPS.....	133
Figura 87. Los archivos guardados en la SD/micro SD donde se almacenan las mediciones.....	133
Figura 88. Mediciones almacenadas en el archivo de texto (txt).....	134
Figura 89. Datos del archivo CSV importados a Excel.....	134
Figura 90. Parte de inferior de la carcasa para Raspberry Pi 2 y Navio2.....	155
Figura 91. Parte superior de la carcasa para Raspberry Pi 2 y Navio2.....	156
Figura 92. Base de la torrecilla del GPS en el dron.....	157
Figura 93. Parte superior de la torrecilla GPS del dron.....	158
Figura 94. Parte superior de la caja de la estación meteorológica.....	159
Figura 95. Parte intermedia de la caja de la estación meteorológica.....	160
Figura 96. Parte inferior de la caja de la estación meteorológica.....	161
Figura 97. Torre del módulo GPS de la estación meteorológica.....	162
Figura 98. Tapa del módulo GPS.....	163
Figura 99. Parte superior de la tapa de la caja previa.....	164
Figura 100. Parte inferior de la tapa de la caja previa.....	165
Figura 101. En la pestaña de datos, seleccionar desde el texto.....	166
Figura 102. En la memoria SD/micro SD se busca el archivo CSV.....	166
Figura 103. Seleccionar “Delimitados”.....	167
Figura 104. Seleccionar solo “Miliares” (en otras versiones “Coma”).....	167
Figura 105. Se deja en “General”.....	168
Figura 106. Datos importados.....	168

Lista de tablas

Tabla 1. Características del hardware de las Raspberry Pi B+ y Pi2 B.....	32
Tabla 2. Especificaciones de los motores Air Gear 350 de T-Motor.....	49
Tabla 3. Características de la tarjeta de control de vuelo Navio2 de Emlid.....	53
Tabla 4. Adaptadores de red inalámbrica soportados por Raspberry Pi.....	63
Tabla 5. Características de hardware de la tarjeta Arduino Mega.....	104
Tabla 6. Especificaciones del sensor DHT22 de temperatura y humedad ambiente.....	105
Tabla 7. Especificaciones del sensor de humedad del suelo.....	106
Tabla 8. Especificaciones del sensor de gotas de lluvia.....	107
Tabla 9. Especificaciones del sensor de luminosidad BH1750.....	108
Tabla 10. Especificaciones del sensor de gases MQ-5.....	109
Tabla 11. Especificaciones técnicas del módulo GPS.....	110

Resumen

El GIESD (Grupo de Investigación En Socio-economía y Desarrollo regional) hace parte del Instituto Superior De Educación Rural (ISER), que cuenta con el laboratorio de redes para sus proyectos, con varias herramientas electrónicas disponibles. Este desarrolla proyectos de electrónica, domótica y redes de conexión. Diferentes proyectos se llevan a cabo, por lo que se organiza en subdivisiones que llevan a cabo estos proyectos simultáneamente; también integra a estudiantes de la escuela básica y media, quienes aprenden de electrónica y microcontroladores para realizar asimismo proyectos dentro del grupo de investigación.

En esta pasantía se realizan dos proyectos, dos prototipos para la agricultura de precisión, con lo que busca integrar herramientas electrónicas a la agricultura.

Seleccionando los componentes, haciendo ingeniería de diseño, usando *Solidworks* como programa CAD, construyendo el dron; calibrando y configurando el *software* del piloto automático en la tarjeta de vuelo Navio2 para crear rutas de vuelo precisas, también permitiendo la conducción manual a control remoto a través de redes inalámbricas.

Diseñando también una estación meteorológica que contiene distintos sensores y GPS. Ensamblando las piezas impresas en 3D, el microcontrolador, los sensores y la pantalla en un solo dispositivo capaz de medir y almacenar los datos obtenidos de forma indefinida, que se visualizan en una pantalla LCD; que al ser móvil consigue ampliar su utilidad, porque se puede usar en cualquier momento y lugar.

Capítulo 1

Introducción

La agricultura de hoy es sobre alta velocidad y alta precisión. En los últimos 10 años, la agricultura de precisión ha experimentado un crecimiento sin precedentes en todo el mundo: Alrededor del 80 por ciento de los nuevos equipos agrícolas vendidos hoy en día tiene algún tipo de componente de agricultura de precisión.

La agricultura de precisión se trata de hacer lo correcto en el lugar correcto en el momento adecuado con la cantidad correcta. Esto conduce a una mayor rentabilidad, una mayor sostenibilidad y un aumento de la productividad. Además, le ahorra tiempo a los agricultores y mejora su bienestar. La aplicación de los productos fitosanitarios con equipos de precisión, incluyendo la comunicación electrónica integrada y la tecnología de tasa variable, demuestra el poder de la tecnología digital en la granja.

Los drones se consideran juguetes caros. Un área que ha recibido poca atención de todos, quizás en detrimento, es el sector agrícola. Es el propósito de este trabajo formar parte de los esfuerzos para utilizar vehículos aéreos no tripulados en un entorno agrícola, explorar su funcionamiento para encontrar sus posibles usos futuros en lo que respecta a la agricultura.

Las estaciones meteorológicas son poco populares en el país. Recopilan datos que son muy útiles para la preparación de terrenos de cultivo, en este documento se muestra cómo se puede conocer más del ambiente circundante y monitorear aspectos del terreno que una persona por sí sola no podría percibir. Esta información puede permitir a un agricultor ajustar sus procesos para abordar los problemas antes de que empeoren. Esto implica mayor rendimiento de cultivos, ya que en muchos de los cultivos comunes los problemas agrícolas pueden identificarse y tratarse antes que con los medios convencionales de detección.

Justificación

Los agricultores no pueden controlar todas las actividades agrícolas, lo que lleva a algunas partes del terreno a ser descuidadas y esto puede llevar a gastos evitables para repararlo. Por un lado, esto significa que algunas áreas sin importancia no ocupan el tiempo de los agricultores, pero por otro lado significa que las áreas que puede llegar a ser importantes se dejan decaer. Los drones pueden sobrevolar los campos de una forma rápida y captar información diversa gracias a sus componentes. Esto permite que aquellos que gestionan los cultivos tengan a su disposición una herramienta para incrementar la productividad.

Por otra parte, la agricultura es extremadamente vulnerable al cambio climático, el aumento de la temperatura global termina por reducir la producción de los cultivos deseados, desplaza las épocas de los cultivos, a la vez que provoca la proliferación de malas hierbas y pestes. Los cambios en los regímenes de lluvias aumentan las probabilidades de fracaso de las cosechas a corto plazo y de reducción de la producción a largo plazo.

Estos proyectos buscan ser parte del principio de la modernización agrícola, en Colombia los campesinos y empresas deben soportar de más los fenómenos naturales con pocas opciones, en la agricultura y la ganadería se usan todavía tecnologías del siglo XIX y XX, no hay razón para depender de estas.

Introducir la tecnología actual al campo es un reto que no se puede omitir, este es un proceso que se construye de proyecto en proyecto. En muchos lugares no hay un apoyo sólido a la agricultura, es un deber de las academias hacer parte del avance científico y cultural, desde donde se desarrollan todas las áreas de la sociedad. La intención de este trabajo es el de aportar un grano de arena a la aceleración del progreso de la base de la economía.

Objetivos

Objetivo general

- Implementar nuevas tecnologías para el desarrollo de sistemas inteligentes aplicados a la agricultura.

Objetivos específicos

- Construir un vehículo aéreo no tripulado (dron) de tipo cuadricóptero.
- Calibrar y configurar el microcontrolador Raspberry Pi 2 y la tarjeta de navegación Navio2.
- Implementar el control remoto mediante conexión Wi-Fi.
- Evaluar el dron en vuelo.
- Diseñar una estación meteorológica, integrando sensores mediante el microcontrolador Arduino.
- Construir una estación meteorológica móvil.
- Evaluar la estación meteorológica.

Marco teórico

Agricultura de precisión

El concepto sobre el que se basa la agricultura de precisión es aplicar la cantidad correcta de insumos, en el momento adecuado y en el lugar exacto. Es el uso de la tecnología de la información para adecuar el manejo de suelos y cultivos a la variabilidad presente dentro de un lote. La agricultura de precisión (AP) involucra el uso de sistemas de posicionamiento global (GPS) y de otros medios electrónicos para obtener datos del cultivo. Las tecnologías de la agricultura de precisión permiten satisfacer una de las exigencias de la agricultura moderna: el manejo óptimo de grandes extensiones. Se presenta como principal ventaja que el análisis de resultados de los ensayos se puede realizar por sectores diferentes dentro de un mismo lote, y de esta manera ajustar el manejo diferencial dentro de los mismos. Por ejemplo, los rendimientos de dos cultivos pueden ser idénticos si se usan los promedios, pero diametralmente opuestos en una situación de loma y en una de bajo en un determinado lote. Este dato sólo podrá obtenerse mediante la realización de un mapa de rendimiento. Del mismo modo, podrán analizarse, el tipo y la dosis de fertilizante a aplicar, la densidad de semilla, la fecha de siembra, el espaciamiento entre hileras, etc. El uso de las tecnologías de la agricultura de precisión puede ayudar a mejorar los márgenes, a través de un aumento del valor del rendimiento (cantidad o calidad), de una reducción en la cantidad de insumos, o de ambos simultáneamente.

Vehículo aéreo no tripulado o dron

El desarrollo tecnológico alrededor de los UAV (Unmanned Aerial Vehicle) o, en español, vehículos aéreos no tripulados, es uno de los de mayor evolución en los últimos años. Un UAV es una aeronave que no cuenta con tripulación humana y puede ser totalmente autónomo u operado por control remoto.

A pesar de sus posibles similitudes estructurales, un UAV no es un aeromodelo, puesto que su uso no está relacionado con dicha actividad deportiva. Mientras que el aeromodelo es un objeto de diversión, el UAV fue concebido con fines militares y posteriormente ha tenido aplicaciones comerciales. Según William Pinilla, docente de la Escuela de Sub-oficiales de la Fuerza Aérea Colombiana: «Esta tecnología fue pensada inicialmente como una aeronave robot, capaz de volar en forma autónoma y cumplir una misión con unos objetivos específicos y de nidos». Estos vehículos cuentan con sistemas avanzados de comunicación, control y navegación, entre ellos un sensor para vigilar desde el espacio aéreo ciertos blancos o terrenos de difícil acceso.

Si se tienen en cuenta los últimos desarrollos, el término UAV podría ya no ser el más acertado para referirse a los vehículos más recientes, debido a su evolución en materia de sistemas complejos llamados UAS (Unmanned Aircraft Systems) o sistemas de aeronaves no tripuladas. Los UAS están conformados por aeronave, estación de control, sub-sistemas de lanzamiento, recuperación, soporte, comunicación y sistema de transporte. A estos últimos sistemas no tripulados se les conoce comúnmente como drones, una adaptación del sustantivo inglés dron, que literalmente traduce zángano.

Otra expresión empleada para referirse a este tipo de aparatos es VANT, o vehículo aéreo no tripulado. Sin embargo, parece no haber consenso respecto a cuál es el vocablo más preciso. Otros términos relacionados son: RPAS (Remotely Piloted Aircraft Systems), que se refiere a sistemas de aeronaves pilotadas remotamente (conformados por aeronave, piloto remoto, enlace comunicacional y estación terrenal) y RPA (Remotely Piloted Aircraft), que alude a una aeronave pilotada a distancia. A fin de cuentas, el término RPAS definiría de forma más precisa el modo de funcionamiento de estos sistemas y no conduciría a errores o malinterpretaciones.



Figura 1. Delta drone del fabricante Kaiser Baas. *Fuente: Kaiser Baas.*

En la aviación militar este tipo de aparatos fue utilizado inicialmente para el reconocimiento aéreo y, posteriormente, como sistemas de bloqueo electrónico. Históricamente, se emplearon en la Guerra del Golfo Pérsico (1990) y en la Guerra de Bosnia (1992). Después entraron en escena los UCAV (*Unmanned Combat Aerial Vehicle*), equipados con armamento aéreo para el combate por aire y labores de bombardeo.

El Departamento de Defensa de los Estados Unidos elaboró un «Mapa de ruta de sistemas de aeronaves no tripuladas (2005–2030)» (*Unmanned Aircraft Systems Roadmap*) en el cual delimitan este tipo de misiones. Se busca que en ellas participen «aeronaves que establezcan puentes aéreos, a manera de plataformas, que faciliten ejecutar ataques de larga distancia y combate aire-aire; este tipo de UAS estaría conformado por UAV del tipo: HALE (High Altitude

Long Endurance), larga duración y extensa autonomía y por UAV Hypersonic (5 Mach) que exceden la velocidad del sonido 5 veces.

Es importante recordar que cuando se hace referencia a un UAV no se está hablando solo del vehículo aéreo, sino también de los sensores que ayudan a controlar su vuelo, así como de sistemas que determinan tanto su rumbo como su trayectoria, y equipos que permiten recolectar información para ser transmitida a los centros de control.

Las principales características para elegir un dron son: autonomía, carga útil transportable, dimensiones, capacidad de despliegue y tipo de misión que debe cumplir. A partir de esto los UAV se clasifican en corto, mediano y largo alcance. Asimismo, dependiendo de su misión principal, se dividen en seis tipos: de blanco, reconocimiento, combate, logística, investigación y desarrollo, y comerciales y civiles.

Clasificación de los drones

Los drones dependiendo su misión principal suelen ser clasificados en 6 tipos:

- De blanco - sirven para simular aviones o ataques enemigos en los sistemas de defensa de tierra o aire.
- Reconocimiento - enviando información militar. Entre estos destacan los MUAVs (Micro Unmanned Aerial Vehicle).
- Combate (UCAV) - Combatiendo y llevando a cabo misiones que suelen ser muy peligrosas.
- Logística - Diseñados para llevar carga.
- Investigación y desarrollo - En ellos se prueban e investigan los sistemas en desarrollo.
- Comerciales y civiles - Son diseñados para propósitos civiles.

También pueden ser categorizados dependiendo de su techo y alcance máximo:

- Handheld: unos 2000 pies de altitud, unos 2 km de alcance.
- Close: unos 5000 pies de altitud, hasta 10 km de alcance.
- NATO: unos 10.000 pies de altitud, hasta 50 km de alcance.
- Tactical: unos 18000 pies de altitud, hasta 160 km de alcance.
- MALE (medium altitude, long endurance) hasta 30000 pies de altitud y un alcance de unos 200 km.
- HALE (high altitude, long endurance) sobre 30.000 pies de techo y alcance indeterminado.
- HYPERSONIC alta velocidad, supersónico (Mach 1-5) o hipersónico (Mach 5+) unos 50000 pies de altitud o altitud suborbital, alcance de 200km.
- ORBITAL en orbitas bajas terrestres (Mach 25+).
- CIS Lunar viaja entre la Luna y la Tierra.

Costo de los drones

Debido a los diversos tipos de drones disponibles, caen en diferentes rangos de precios. Los drones de juguete pequeños sin cámara y con una funcionalidad limitada pueden costar tan solo \$ 40'000 pesos, los drones de rango medio con más potencia y una cámara están disponibles por alrededor de \$ 360'000 pesos. Los drones de gama alta con baterías extendidas, palas de rotor de fibra de vidrio muy potentes y un soporte de cámara estabilizado con cardán solo para uso profesional, como la grabación de eventos deportivos, pueden costar más de \$ 27'000'000. Sin embargo, estos son todos los drones comerciales, mientras que los entornos agrícolas tienen diferentes requisitos.

Fuente de energía

Hasta ahora, todos los drones han dependido de las baterías para alimentar sus sistemas de propulsión (un motor eléctrico que hace girar una hélice). En algunos prototipos de drones, paneles solares a través del dron se han utilizado para recargar las baterías en vuelo, prolongando su uso mientras el sol esté brillando. También hay un dron prototipo llamado *Yeair* que usa un pequeño motor de combustión interna de dos tiempos para alimentar sus cuatro rotores, junto con una batería pequeña para encender el dron y para compensar las pérdidas de potencia durante el ascenso y el descenso. La batería también ayuda al avión no tripulado a hacer un aterrizaje de emergencia si el combustible se agota mientras está en el aire.

Tiempo de vuelo y rango

La mayoría de los drones de consumo, independientemente de su tamaño, tendrán dificultades para volar durante más de 15 minutos. Los drones profesionales como la serie DJI Phantom tienen baterías más grandes y pueden administrar alrededor de 25 minutos de tiempo de vuelo. Los drones alimentados con energía solar mencionados anteriormente también tienen baterías relativamente pequeñas, pero pueden recargarse durante el vuelo, siempre que el sol esté brillando. En lo que concierne al rango, hay dos dimensiones a considerar: distancia y altitud. Los drones no autónomos generalmente se pueden controlar a una distancia de 2 km del operador.

Los zánganos autónomos, específicamente los destinados a usos agrícolas, pueden volar tan lejos como su fuente de poder los lleve porque su software de navegación está a bordo. El alcance vertical es más complicado, ya que existen varias leyes sobre el uso de drones para uso personal. En los Estados Unidos, por ejemplo, estos drones no pueden volar a más de 400 pies (~ 122 m) del suelo. En general, los drones no deben volar más allá de la línea de visión del operador por cuestiones de seguridad y si un dron se va más allá de su rango máximo (vertical u horizontal) podría estrellarse contra el suelo y posiblemente lesionar a los transeúntes.

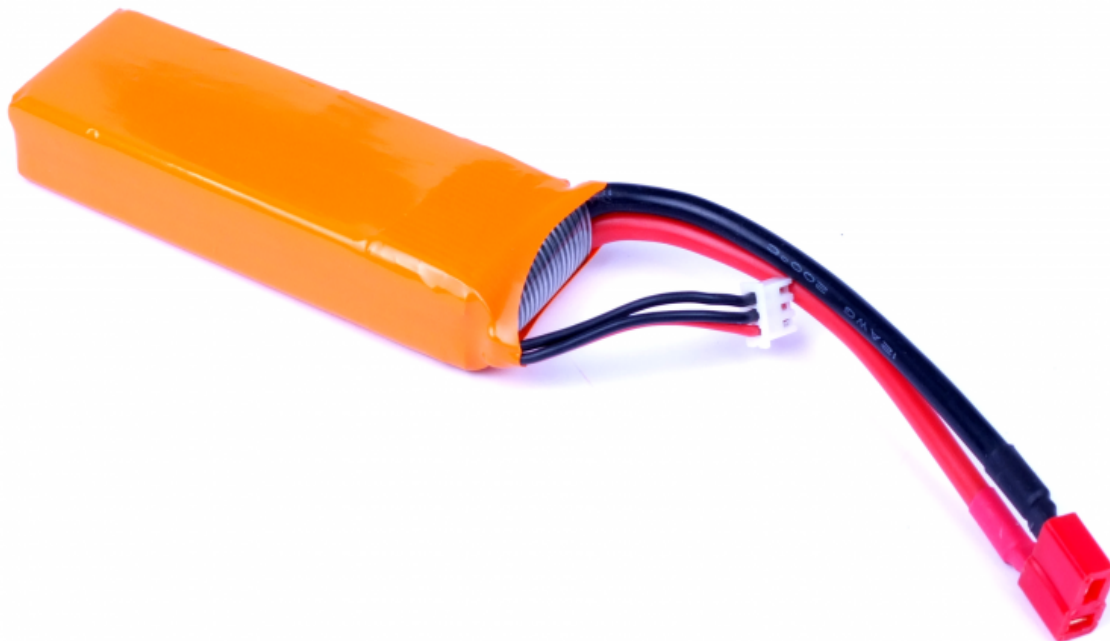


Figura 2. Batería LiPo. Fuente: DroneVal.

Facilidad de uso

En general, los diseños de helicóptero cuádruple son más difíciles de dominar, ya que se mueven rápidamente en las 3 dimensiones (arriba / abajo, adelante / atrás e izquierda / derecha), lo que puede desorientar a los pilotos principiantes. La mayoría de los drones de ala fija vienen con un software que permite el vuelo autónomo, que no requiere la intervención del usuario, aparte de las instrucciones de aterrizaje. Algunas compañías también ofrecen sus drones de helicóptero cuádruple con capacidades de estacionario en su lugar, manteniendo el dron en una posición fija si va más allá del alcance del controlador hasta que el propietario se acerca lo suficiente como para recuperarlo.

¿Cómo vuela el dron?

Para el caso de aeronaves del tipo multirrotor, sea el caso de helicópteros la explicación es más sencilla, puesto que se trata de contrarrestar la fuerza de la gravedad con una fuerza de empuje

contraria proporcionada por hélices conectadas a rotores que según la velocidad y el tamaño ejercerán mas o menos fuerza de empuje.

Se podría considerar a las palas de las hélices de un multirroto, como un ala de avión en movimiento giratorio y aplicar los principios aerodinámicos de las alas de un avión.

Los helicópteros para contrarrestar la fuerza de inercia de su rotor y no girar descontroladamente utilizan unos rotores de cola o estabilizadores.

En los multirrotores como el cuadricóptero el problema vendría si todos los rotores giraran a la vez en el mismo sentido ocasionando que la aeronave girara de forma descontrolada

Para solucionar esto en el cuadricóptero se colocan los rotores con las hélices que giran en sentido opuesto alternadamente y colocados de forma diametral, dependiendo de las velocidades de los motores es lo que se puede controlar el movimiento de la aeronave.

Ascenso

El ascenso es la fuerza que actúa contra el peso de la nave, elevándola en el aire. Un cuadricóptero o dron de cuatro hélices es esencialmente un helicóptero que tiene 4 motores que se encuentran equitativamente distribuidos, dispuestos en las esquinas de un cuerpo cuadrado formando una cruz entre ellos.

A diferencia de los helicópteros, los multirrotores sin asistencia electrónica son inestables, porque es extremadamente difícil controlar múltiples rotores manualmente.

Las hélices del multirroto generan una fuerza de elevación usando principios similares (empujando el aire hacia abajo y la diferencia en la presión del aire). Para que el multirroto salga del suelo y pueda volar y volar, esta fuerza debe ser mayor que el peso de la nave.

Marco o chasis de un dron

Es el esqueleto del multirroto, es la estructura que le da forma y en donde deben ir alojadas todos los componentes del mismo para su correcto funcionamiento: Motores, electrónica, baterías, etc.

Los materiales utilizados para la construcción del chasis, están muy relacionados con las prestaciones a las que se destine el dron (aluminio, fibra de carbono, fibra de vidrio, Kevlar, etc.) y por supuesto muy relacionadas también con el precio.

Motor eléctrico

Los motores eléctricos se han desarrollado en un frenético proceso evolutivo que comienza a mediados del siglo XIX y que aún continúa en la actualidad. Inventores como Edison y Tesla entre muchos otros, realizaron contribuciones significativas que lograron eficacia y eficiencia en la conversión electromecánica de energía.

Motor de corriente directa

La gran variedad de la velocidad junto con su fácil control y la gran flexibilidad de las características par-velocidad del motor de corriente continua, han hecho que se empleen con maquinas de velocidad variable en las que se necesite amplio margen de velocidad y control fino de las mismas. Por ello se utiliza en un creciente número de procesos industriales que requieren una exactitud en su control o una gama de velocidades que no se puede conseguir con motores de corriente alterna, pese al precio de un motor de corriente continua que es considerablemente mayor al de un motor de inducción de igual potencia.

Motor *brushless* o sin escobillas

En este motor el inductor, está formado por un juego de bobinas fijas, y un circuito electrónico que cambia el sentido de la corriente a cada una de las bobinas para adecuarse al giro

del rotor. Este tipo de motores proporciona un buen par de arranque y un eficiente control de la velocidad.

La vida útil del motor brushless es larga: al no tener escobillas, no se producen rozamientos que generen desgaste ni ruido, y no es necesario un mantenimiento continuo. El motor brushless ofrece una muy buena relación entre potencia de salida y tamaño y, por tanto, gran rendimiento. Además, el motor *brushless* también posee una buena relación entre velocidad y par motor.

Motores Air Gear 350

Es un conjunto de cuatro motores sin escobillas de la marca T-Motor, con sus respectivos reguladores de velocidad (ESC). Dos de los motores están marcados en blanco que deben seguir la dirección de las manecillas del reloj, encargados de la propulsión del dron, mientras los otros dos se encargan de las maniobras y la estabilidad del dron en el aire, con dirección contraria a las manecillas del reloj. Soportan pesos entre 1200 y 1500 gramos.

Hélice. Conjunto de aletas que giran alrededor de un eje y empujan el fluido ambiente produciendo en él una fuerza de reacción que se utiliza principalmente para la propulsión de barcos y aeronaves.

Hélices en drones

En la mayoría de los drones existen dos sentidos de giro para las hélices. Si se habla de un cuadricóptero, este necesita 2 hélices que giran en sentido horario y otras 2 en sentido anti horario. Dependiendo del tipo de dron (Cuadri, Hexa, Octa..) cada hélice ira colocada en una posición concreta, la cual, es calve para que el dron pueda levantarse del suelo.

El tamaño y grado de inclinación de las hélices viene determinados por dos números. El tamaño de la hélice se mide de punta a punta, conocidos en algunos casos como diámetro por la circunferencia que genera al girar.

Un aumento en el grado de inclinación o del tamaño supondrá un mayor empuje, asumiendo que no hay variación en las vueltas RPM del motor.



Figura 3. Paquete *Air Gear 350*: incluye 4 motores *brushless*, 4 reguladores de velocidad y 4 hélices. Fuente:

T-Motor.

Controlador de vuelo o unidad de control

Es la parte más importante de un dron. Es la encargada de regular la potencia de cada motor y de estabilizar el dron en función de la información que recogen sus sensores. Si todos los motores reciben la misma potencia, el dron se mantendrá estable en el eje vertical. Si los motores traseros reciben más potencia, el dron se desplazará hacia delante, y viceversa. Los componentes más comunes en una controladora de vuelo son: acelerómetro, giroscopio, barómetro, magnetómetro o brújula, GPS, procesador.

Acelerómetro. Un acelerómetro es un dispositivo que mide la vibración o la aceleración del movimiento de una estructura. La fuerza generada por la vibración o el cambio en el movimiento (aceleración) hace que la masa comprima el material piezoeléctrico, generando una carga eléctrica que es proporcional a la fuerza ejercida sobre él.

El hecho de que la carga sea proporcional a la fuerza y que la masa sea constante hace que la carga también sea proporcional a la aceleración.

Giroscopio. Disco que, en movimiento de rotación, conserva su eje invariable aunque cambie la dirección de su soporte.

Barómetro. Instrumento que sirve para determinar la presión atmosférica. Basándose en la medida de presión atmosférica se puede determinar la altura.

Brújula. Instrumento consistente en una caja en cuyo interior una aguja imantada gira sobre un eje y señala el norte magnético, que sirve para determinar las direcciones de la superficie terrestre.

Magnetómetro. Aparato que mide la inducción de un campo magnético en una dirección determinada.

GPS. El Sistema de Posicionamiento Global (GPS) es un servicio propiedad de los EE.UU. que proporciona a los usuarios información sobre posicionamiento, navegación y cronometría.

Este sistema está constituido por tres segmentos: el segmento espacial, el segmento de control y el segmento del usuario. La Fuerza Aérea de los Estados Unidos desarrolla, mantiene y opera los segmentos espacial y de control.

La índole gratuita, ininterrumpida y fiable del Sistema de Posicionamiento Global (GPS) ha permitido a los usuarios de todo el mundo desarrollar cientos de aplicaciones que afectan casi todas las facetas de la vida moderna.

Controladora de vuelo Navio2

Es una controladora de vuelo fabricada por la empresa *Emlid*, la cual cuenta con un receptor GNSS que capta señales de satélites GPS, GLONASS, Beidou, Galileo y SBAS; un IMU Dual que se compone de acelerómetros, giroscopios y magnetómetros para la orientación y movimiento del dron; un barómetro de alta precisión; puertos ADC, I2C y UART así como 14 salidas PWM, entre otras cosas.

Muchas controladoras de vuelo vienen con un software que permite calibrar el dron de forma precisa, configurar la sensibilidad de los controles de la emisora de radio, la configuración de los botones, la velocidad máxima del dron, vuelta a casa automática, etc. La calibración y configuración de la controladora de vuelo es una parte esencial para que el dron vuele correctamente y evitar accidentes y sorpresas. Otras controladoras de vuelo permiten además la configuración de misiones basadas en *waypoints* (puntos GPS). Estas misiones no envían comandos de velocidad al autopiloto, si no que envían puntos GPS de destino o comandos de posición y el autopiloto se encarga de alcanzarlos.

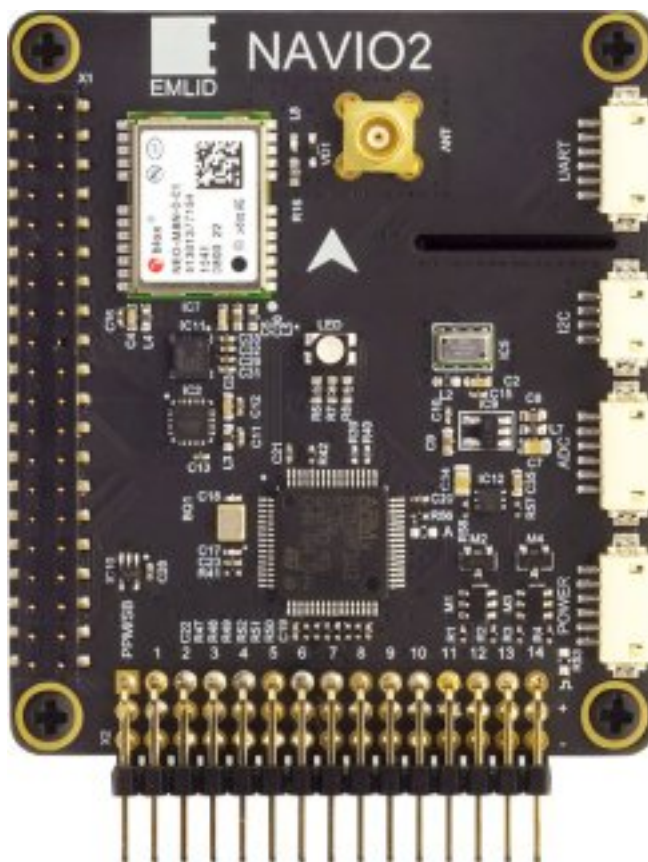


Figura 4. Tarjeta controladora de vuelo Navio2. Fuente: Emlid.

Joystick. Dispositivo que se conecta con un ordenador o videoconsola para controlar de forma manual un *software*, especialmente juegos o programas de simulación. En español llamado también palanca de mando. El nombre proviene del inglés joy que significa alegría y stick que es palo.

Raspberry Pi

La Raspberry Pi es una placa de pequeño tamaño (8,5 x 5,4 cm) que incluye todo un ordenador completo. A estos dispositivos se los conoce por las siglas SBC (Single Board Computer). En su diseño se utiliza un SoC (System on a Chip), que incluye en un solo chip el procesador o CPU (ARMv6/ARMv7), la memoria RAM (512 MB/1 GB) y la tarjeta gráfica o GPU (VideoCore IV).

Los primeros modelos que salieron al mercado consistían en dos tipos de placas: el modelo A, que estaba más bien destinado a desarrolladores y poseía menos prestaciones, y el modelo B, que es el más utilizado por los usuarios domésticos. Este modelo incluye 512 MB o 1 GB de RAM, dos o cuatro puertos USB, un conector de red Ethernet 10/100, salida de vídeo HDMI y otra salida analógica de audio y vídeo. Como unidad de almacenamiento utiliza una tarjeta SD y se alimenta a través de un conector mini USB, similar al de los cargadores para teléfonos móviles. Como sistema operativo puede usar una amplia variedad de distribuciones de Linux (entre ellas, Debian, Ubuntu, Fedora o Arch Linux) u otros sistemas como FreeBSD o RISC OS.

Los primeros diseños se realizaron en el Reino Unido en el año 2006, pero no se empezó a comercializar hasta febrero de 2012 por parte de la Fundación Raspberry Pi, que es la que se encarga de su desarrollo.

En julio de 2014 se presenta una nueva placa, bajo el nombre de modelo B+, que es una actualización del anterior modelo B. Aparte del rediseño de algunos componentes electrónicos de la placa, las diferencias fundamentales entre ambas son la inclusión de dos puertos USB más (sumando cuatro), la sustitución de la tarjeta SD por una micro SD y la fusión de los dos conectores analógicos (audio y vídeo) en un solo conector. Pero la parte fundamental, que es el el SoC (CPU, GPU y RAM), es idéntica en los dos modelos de la clase B.

El 2 de febrero de 2015 salió al mercado una nueva placa: la Raspberry Pi2 modelo B. Visualmente idéntica al modelo B+ anterior, pero añade dos importantes novedades, una CPU de cuatro núcleos ARMv7 y 1 GB de RAM. Todo lo demás se mantiene exactamente igual y, por ello, es totalmente con el modelo B+.

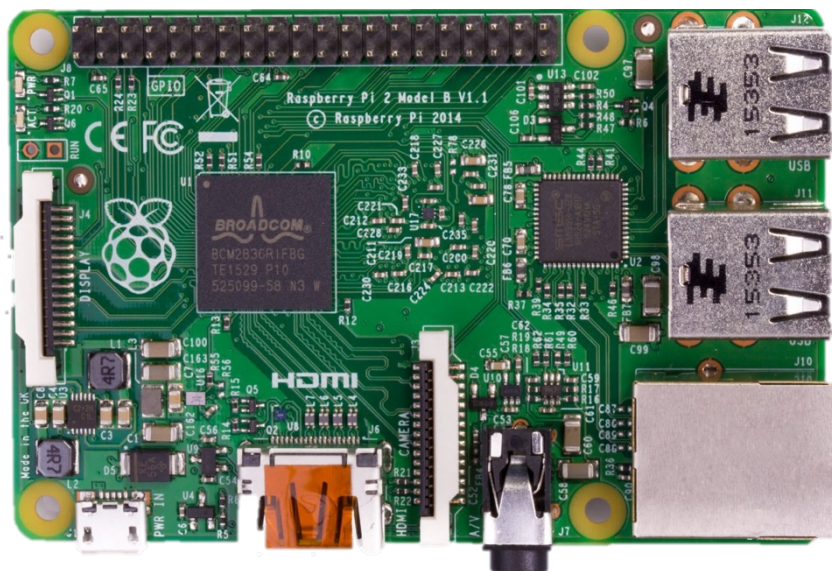


Figura 5. Raspberry Pi 2 modelo B. Fuente: Raspberry Pi Foundation.

Esta placa está destinada a desarrolladores y a personas a las que les gusta aprender, experimentar y trastear con electrónica. Con ella pueden hacerse proyectos de todo tipo (robótica, demótica, etc.), pero la mayoría de los usuarios la utilizan para cosas sencilla y prácticas.

Debido a su bajo consumo (2,5 W), puede conectada las 24 horas del día, y como la CPU no se calienta apenas, no necesita ventiladores, por lo que es totalmente silenciosa.

El hardware de la Raspberry Pi2

La Raspberry Pi2 llega en un momento donde otras placas de desarrollo se estaban aprovechando del tirón de la Raspberry Pi, ofreciendo al algunos casos placas más interesantes. Sinceramente estaban adelantando a la Raspberry Pi por la derecha, ofreciendo más potencia y funciones, y lo único que ha permitido a la Raspberry Pi continuar en el reinado ha sido la ingente comunidad de seguidores que tiene, en el sector educativo y en el gran conjunto de personas que han buscado un ordenador barato capaz de reproducir vídeo 1080p sin despeinarse.

La nueva Raspberry es aproximadamente 6 veces más rápida que los modelos anteriores en las tareas más frecuentes. Atrás quedan también los modelos con 256 y 512 MB, ya que ahora

la memoria RAM es de 1 GB. Como detalle curioso cabe apuntar que es la caja más grande de todas las Raspberry Pi, ya que ahora se cuenta con un manual de inicio rápido y uso seguro en muchos idiomas. Sus características más relevantes en comparación con su modelo anterior se muestran en la tabla 1.

Memoria de almacenamiento SD

Una tarjeta SD (Secure Digital) es una pequeña tarjeta que permite guardar información en dispositivos portátiles como teléfonos móviles, cámaras digitales o tabletas.

Las tarjetas SD se diferencian por sus medidas, su capacidad para almacenar contenidos y la velocidad a la que transmiten y copian los datos.

Medidas:

- Tarjeta SD: 32 mm de alto x 24 mm de ancho x 21 mm de grosor. Es el formato que suelen usar las cámaras digitales u ordenadores.
- Tarjeta mini SD: 21,5 mm de alto x 20 mm de ancho x 14 mm de grosor.
- Tarjeta micro SD: tan sólo 15 mm de alto x 11 mm de ancho x 10 mm de grosor. Se trata del formato más común en teléfonos móviles o tablets.

Almacenamiento:

- SD SC (Standard Capacity): puede guardar hasta 2GB de contenidos.
- SD HC (High Capacity): capaces de almacenar hasta 32GB.
- SD XC (Extended Capacity): llegan hasta 2TB (2.000GB)

Tabla 1*Características del hardware de las Raspberry Pi B+ y Pi2 B*

	Raspberry Pi modelo B+	Raspberry Pi 2 modelo B
CPU	ARM11 ARMv6 700 MHz	ARM11 ARMv7 ARM Cortex-A7 4 núcleos @ 900 MHz
SoC	Broadcom BCM2835	Broadcom BCM2836
Overcolcking	Sí, hasta 1000 MHz	Sí, hasta 1000 MHz
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
RAM	512 MB LPDDR SDRAM 400 MHz	1 GB LPDDR2 SDRAM 450 MHz
USB	4	4
Salidas de vídeo	HDMI 1.4 @ 1920 x 1200 píxeles	HDMI 1.4 @ 1920 x 1200 píxeles
Almacenamiento	microSD	microSD
Ethernet	Sí, 10/100 Mbps	Sí, 10/100 Mbps
Consumo	5 V, 600 mA	5 V, 900 mA, aunque depende de la carga de trabajo de los 4 núcleos
Precio		
estimativo	\$ 120000 COP	\$ 140000 COP

Fuente: Raspberry Pi2 para electrónicos.

Clase:

- Clase 2: graba 2MB por segundo, equivalente al tamaño de una foto común.
- Clase 4: capaz de guardar 4MB por segundo, que corresponde aproximadamente a un archivo con una canción en MP3.
- Clase 6: graba 6MB por segundo.
- Clase 10: capaz de grabar 10MB por segundo o más.

Redes de área local

A las redes de datos que unen ordenadores se les suele llamar LAN (*Local Area Network*). Hay varios sistemas de LAN, pero los más conocidos son: el del tipo Ethernet, que sobre un cable (bus) va uniendo todos los ordenadores y terminales, que es la red más usada para interconectar ordenadores. El otro sistema (*ring*) adopta una estructura en anillo, en el que cada ordenador se conecta a otro y este a otro y así sucesivamente, que es bastante menos utilizado.

Las redes de área local son redes de propiedad privada, dentro de un solo edificio o recinto, que varían desde unos 10 metros hasta unos pocos kilómetros de extensión. Se utilizan para conectar ordenadores personales y estaciones de trabajo en oficinas, organizaciones y fábricas, con el objeto de compartir recursos y comunicar usuarios. Operan a una velocidad entre 10 Mbit/s y 10 Gbit/s, tienen bajo retardo y muy pocos errores.

WLAN: redes locales inalámbricas

Una WLAN (*Wireless LAN*) es un sistema de comunicaciones de datos que transmite y recibe datos utilizando ondas electromagnéticas, en lugar del par trenzado, coaxial o fibra óptica utilizado en las LAN convencionales, y que proporciona conectividad inalámbrica de igual a igual o P2P (*Peer to peer*), dentro de un edificio, de una pequeña área residencial/urbana o de un campus universitario.

Routers

Los *routers* (la denominación inglesa del dispositivo es la más usual) o encaminadores operan en una manera similar a los puentes (que enlazan redes) con la particularidad de que lo hacen en un nivel superior, en el nivel 3 del modelo OSI. Manejan por lo tanto direcciones de red y son dependientes del protocolo.

Los *routers* leen la información de cada paquete, utilizan procedimientos complejos para determinar el destino apropiado, y reempaquetan y retransmiten los datos.

Dirección IP

Dirección IP significa Dirección de Protocolo Internet, y cada dispositivo que se conecta a una red (así como a Internet) tiene una.

Una dirección IP se parece a su número de teléfono. Su número de teléfono es un conjunto único de números que identifican a su teléfono de forma que otra persona pueda llamarle. Igualmente, una dirección IP es un conjunto único de números que identifican a su equipo de forma que pueda enviar y recibir datos hacia y desde otros equipos, respectivamente.

Actualmente, la mayoría de las direcciones IP constan de cuatro conjuntos de números separados por un punto. 192.168.1.42 es un ejemplo de dirección IP.

Protocolo UDP

UDP son las siglas de Protocolo de Datagrama de Usuario (en inglés User Datagram Protocol) un protocolo sin conexión que, como TCP, funciona en redes IP.

UDP/IP proporciona muy pocos servicios de recuperación de errores, ofreciendo en su lugar una manera directa de enviar y recibir datagramas a través una red IP. Se utiliza sobre todo cuando la velocidad es un factor importante en la transmisión de la información, por ejemplo, RealAudio utiliza el UDP.

El FTP utiliza TCP/IP, mientras que TFTP utiliza UDP. TFTP son las siglas de Protocolo de Transferencia de Archivos Triviales (en inglés Trivial File Transfer Protocol), y puesto que es trivial, perder algo de información en la transferencia no es crucial.

Protocolo SSH

SSH™ (o Secure SHell) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas.

SSH está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la shell de comando, tales como telnet o rsh. Un programa relacionado, el scp, reemplaza otros programas diseñados para copiar archivos entre hosts como rcp. Ya que estas aplicaciones antiguas no encriptan contraseñas entre el cliente y el servidor, evite usarlas mientras le sea posible. El uso de métodos seguros para registrarse remotamente a otros sistemas reduce los riesgos de seguridad tanto para el sistema cliente como para el sistema remoto.

Estación terrestre de control o GCS

Una estación terrestre suele ser una aplicación de software que se ejecuta en una computadora en tierra y que se comunica con un dron mediante telemetría inalámbrica. Muestra datos en tiempo real sobre el rendimiento y la posición del dron y puede servir como una cabina virtual, mostrando muchos de los mismos instrumentos que tendrías si estuvieras volando en un avión real. Un GCS también se puede usar para controlar un dron en vuelo, cargando nuevos

comandos de misión y parámetros de configuración. A menudo también se utiliza para controlar las transmisiones de video en vivo de las cámaras de un dron.

Ardupilot

Ardupilot es el software de piloto automático de código abierto más avanzado, completo y confiable disponible. Ha sido desarrollado durante más de 5 años por un equipo de diversos ingenieros profesionales e informáticos. Es el único software de piloto automático capaz de controlar cualquier sistema de vehículo imaginable, desde aviones convencionales, multirrotores y helicópteros hasta embarcaciones e incluso submarinos. Y ahora se está ampliando para ofrecer compatibilidad con nuevos tipos de vehículos emergentes, como aviones cuádruples y helicópteros compuestos.

Instalado en más de 1,000,000 de vehículos en todo el mundo, y con sus herramientas avanzadas de registro de datos, análisis y simulación, Ardupilot es el software de piloto automático más probado y comprobado. La base de código de código abierto significa que está evolucionando rápidamente, siempre a la vanguardia del desarrollo de tecnología. Con muchos proveedores periféricos creando interfaces, los usuarios se benefician de un amplio ecosistema de sensores, computadoras complementarias y sistemas de comunicación.

Sistema operativo de robot (ROS)

El ROS (Robot Operating System) es un esfuerzo de miles de roboticistas de todo el mundo para facilitar el desarrollo de nuevos robots. ROS es un proyecto de código abierto que incluye un montón de herramientas útiles y que hace que el proceso de desarrollo sea más eficiente. La idea es que no es necesario rediseñar la rueda cada vez. Alguien más ya ha hecho eso, y probablemente lo hayan hecho mejor, por lo que se puede enfocar más energía en la parte específica a construir.

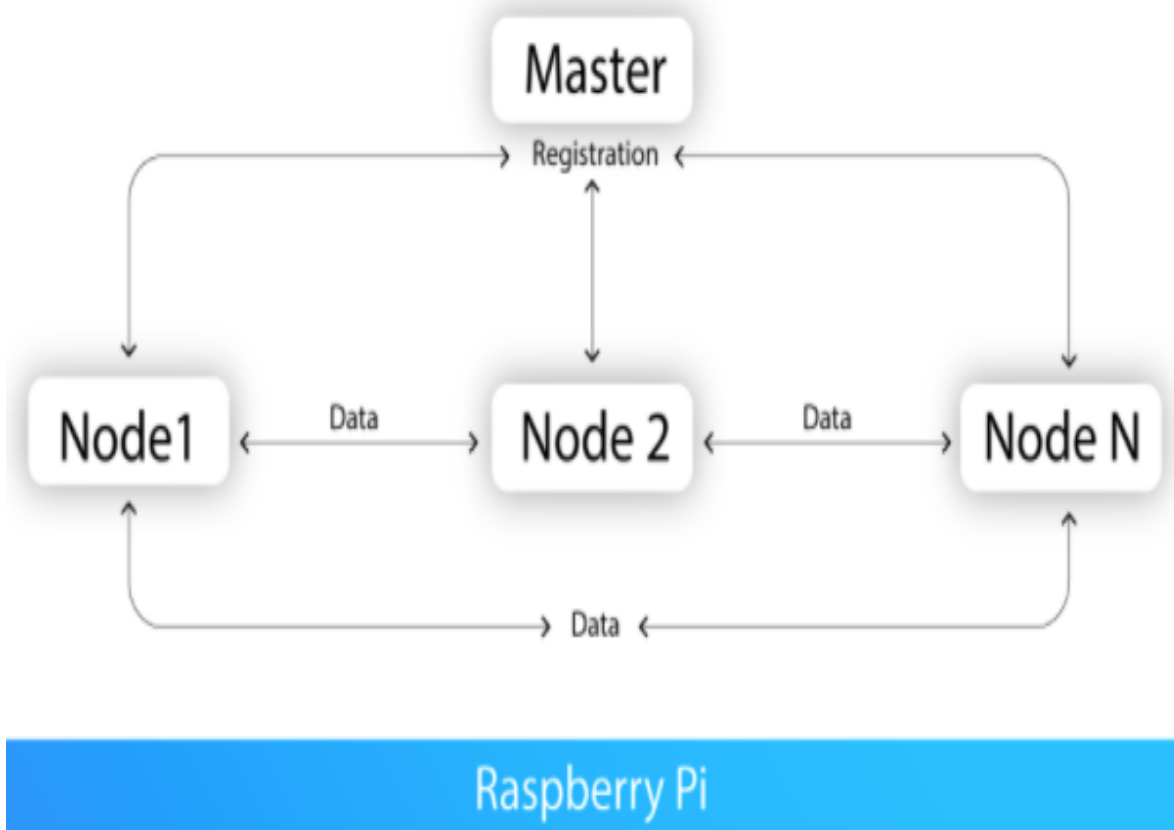


Figura 6. Esquema general de ROS en el SO distribuido por Emlid. *Fuente: Emlid.*

La imagen Emlid incluye ROS preinstalado, por lo que todo lo que hay que hacer es iniciarlo después de una pequeña configuración. Después de ejecutar ROS, está ROS Master, el lugar que contiene todos los servicios, una especie de punto de encuentro para los nodos. Desde este lugar, ahora se encuentran nodos y se hace que se comuniquen entre ellos en una Raspberry Pi. Puede haber diferentes conjuntos de controladores dentro de un nodo.

Los nodos pueden encontrarse y compartir datos. Esta información compartida entre nodos se llama mensajes. Los nodos pueden publicar mensajes a los temas y pueden suscribirse a temas para recibir mensajes.

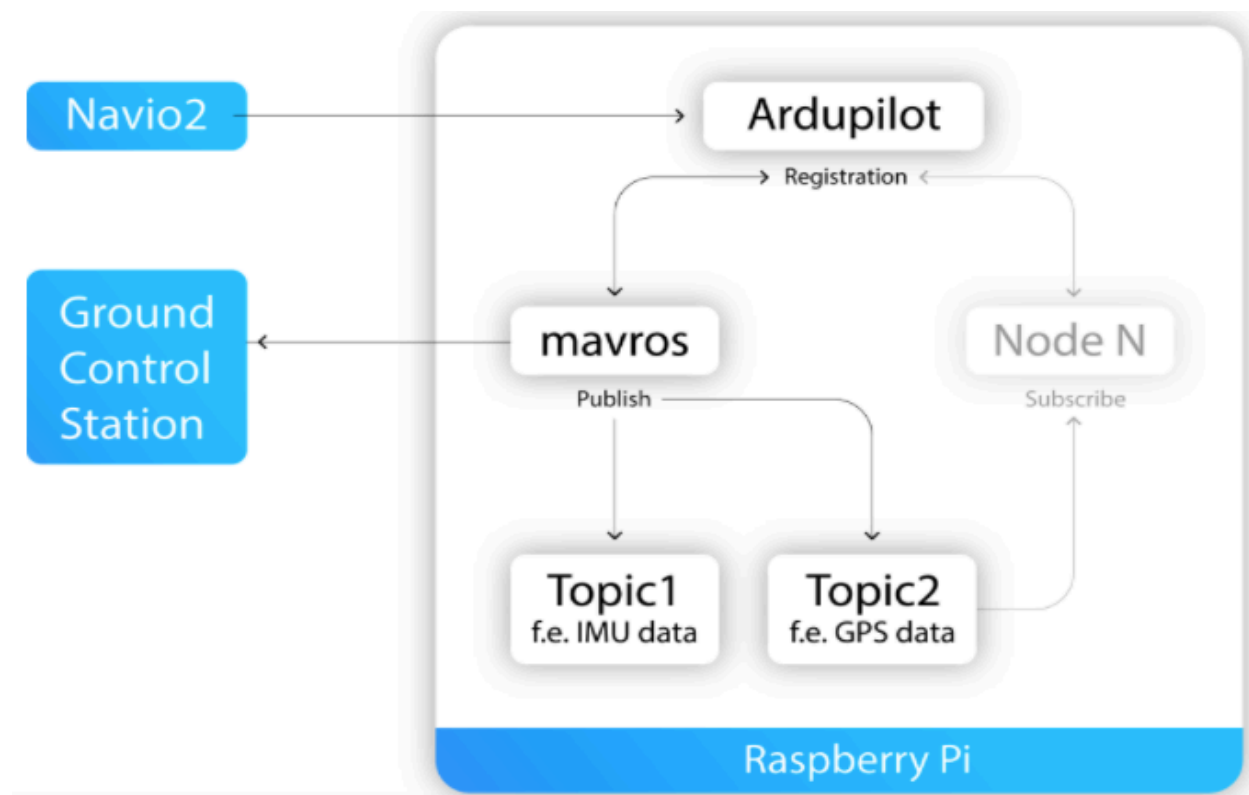


Figura 7. Interacción del nodo *mavros* con el GCS. Fuente: Emlid.

Suponiendo que se ejecuta ROS junto con Ardupilot. La imagen Emlid contiene un nodo *mavros* preinstalado. Este nodo proporciona una gran cantidad de controladores de sensores y comunicación para Ardupilot y proxy para la estación terrestre de control.

Microcontrolador

Los microprocesadores son circuitos integrados que contienen millones de transistores en su interior, los cuales crean circuitos complejos encargados de realizar diferentes tareas. También se los denomina unidad de procesamiento central o CPU, ya que muchos de ellos pueden actuar como el cerebro de un sistema computacional, administrando todas las tareas que este realice y llevando a cabo las operaciones con los datos.

Los microprocesadores están diseñados para interpretar y ejecutar las instrucciones que se le indiquen siempre y que suelen ser operaciones simples, como sumar, restar, multiplicar y dividir.

Pero también existen instrucciones lógicas, como AND, OR, NOT, entre otras. El listado de instrucciones recibe el nombre de programa, que las ejecuta una por una por medio del microprocesador.

Microcontrolador Arduino

Arduino es el ecosistema de hardware y software de código abierto líder en el mundo. La compañía ofrece una gama de herramientas de software, plataformas de hardware y documentación que permiten a casi cualquier persona ser creativo con la tecnología.

Arduino es una herramienta popular para el desarrollo de productos IoT, así como una de las herramientas más exitosas para la educación STEM / STEAM. Cientos de miles de diseñadores, ingenieros, estudiantes, desarrolladores y fabricantes de todo el mundo están utilizando Arduino para innovar en música, juegos, juguetes, hogares inteligentes, agricultura, vehículos autónomos y más.

La primera placa de Arduino se presentó en 2005 para ayudar a los estudiantes de diseño, que no tenían experiencia previa en electrónica o programación de microcontroladores, a crear prototipos de trabajo que conectarán el mundo físico con el mundo digital. Desde entonces, se ha convertido en la herramienta de creación de prototipos de electrónica más popular utilizada por ingenieros e incluso grandes empresas.

Arduino es el primer proyecto de hardware de código abierto generalizado y se creó para construir una comunidad que podría ayudar a difundir el uso de la herramienta y beneficiarse de las contribuciones de cientos de personas que ayudaron a depurar el código, escribir ejemplos, crear tutoriales y otros usuarios en los foros y construir miles de grupos en todo el mundo.

Arduino Mega

El Arduino Mega 2560 es un tablero de microcontroladores basado en el ATmega2560. Tiene 54 pines digitales de entrada / salida (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a una computadora con un cable USB o con un adaptador de CA a CC o batería para comenzar. La placa Mega 2560 es compatible con la mayoría de los escudos diseñados para el Uno y los tableros anteriores Duemilanove o Diecimila.



Figura 8. Arduino Mega 2560. Fuente: Arduino.

Sensores

Sensor de temperatura y humedad DHT22. El DHT22 Sensor digital de temperatura y humedad. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica). Es bastante simple de usar, pero requiere sincronización cuidadosa para tomar datos. El único

inconveniente de este sensor es que sólo se puede obtener nuevos datos una vez cada 2 segundos, así que las lecturas que se pueden realizar serán mínimo cada 2 segundos.



Figura 9. Sensor DHT22. Fuente: MaxElectrónica.

Cada sensor de este modelo se compensa con la temperatura y se calibra en la cámara de calibración precisa y el coeficiente de calibración se guarda en tipo de programa en la memoria OTP, cuando el sensor está detectando, se buscará el coeficiente de la memoria.

El tamaño pequeño y bajo consumo y la larga distancia de transmisión (20m) permiten que el sensor DHT22 se adapte a todo tipo de aplicaciones exigentes.

Su sola fila empaquetada con cuatro pines, hace que la conexión sea muy conveniente.

Sensor de humedad del suelo. Este sensor de humedad puede leer la cantidad de humedad presente en el suelo que lo rodea. Es un sensor de baja tecnología, pero es ideal para el seguimiento de un jardín urbano.

Este sensor utiliza las dos sondas para pasar corriente a través del suelo, y luego lee la resistencia para obtener el nivel de humedad. Más agua hace que el suelo conduzca la electricidad con mayor facilidad (menos resistencia), mientras que el suelo seco es un mal conductor de la electricidad (más resistencia).

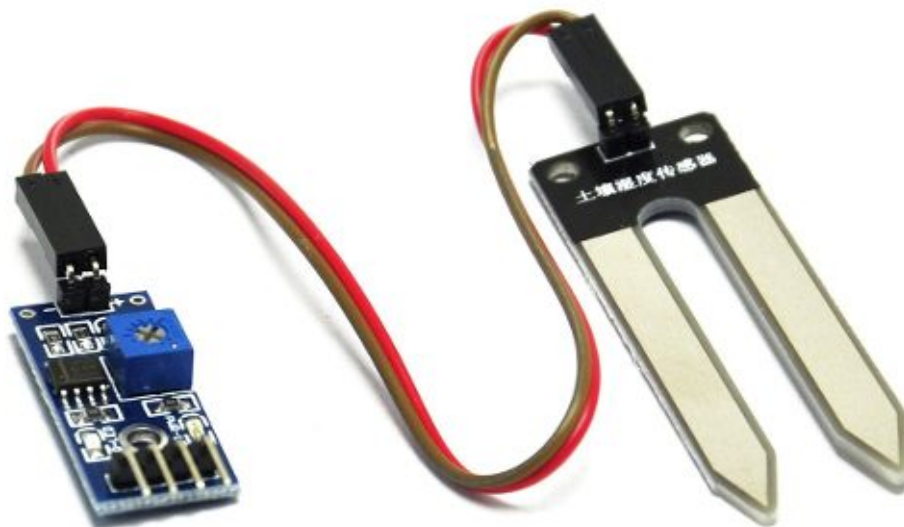


Figura 10. Sensor de humedad del suelo. *Fuente: Mercado Libre Uruguay.*

Sensor de gotas de lluvia. Este Sensor permite detectar gotas de lluvia, como un sensor de lluvia, y seguimiento de humedad y se puede utilizar para una variedad de condiciones climáticas. Convierte en números la señal de referencia de salida output AO. La salida analógica puede ser conectada al puerto AD de un microcontrolador para detectar la intensidad de la humedad y la precipitación.

Este módulo consiste en una serie de pistas conductoras impresas sobre una placa de baquelita. La separación entre las pistas es muy pequeña. Lo que este módulo hace es crear un corto circuito cada vez que las pistas se mojan. El agua hace que se cree un camino de baja resistencia entre las pistas con polaridad positiva y las pistas conectadas al GND.

Posee un amplificador operacional, específicamente el circuito integrado LM393. Este es el encargado de amplificar el pequeño diferencial de voltaje que se genera cuando una gota de agua cae sobre las pistas del módulo.



Figura 11. Sensor de gotas de lluvia. *Fuente: Tdrobótica.*

Sensor de luminosidad BH1750. Esta es una versión mejorada del típico sensor de luz a base de un LDR, el cual simplemente entrega un valor analógico, nos entrega automáticamente el valor En Lux (desde 1 lx hasta 65535 lx), y se comunica por I2C. Este sensor nos permite medir la cantidad de luz por metro cuadrado que hay en alguna habitación y poder modificar automáticamente las luces de los alrededores.

Ejemplos de Lux:

- Noche: 0.001—0.02
- Luz Lunar: 0.02—0.3
- Nublado Interior: 5—50
- Nublado Exterior: 50—500
- Soleado Interior: 100-1000
- Luz mínima para la lectura: 50—60
- Intensidad estándar sistema de video hogareño: 1400

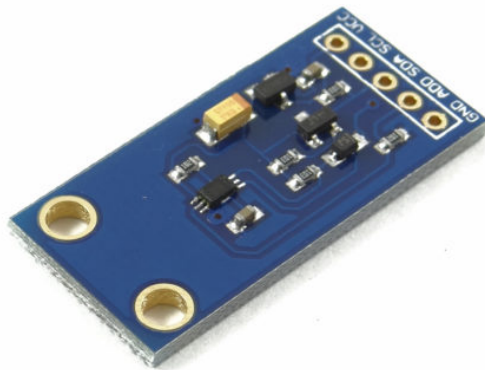


Figura 12. Sensor de luminosidad BH1750. *Fuente: Moviltronics.*

Sensor de calidad del aire. Este sensor de control de calidad de aire es usado para la detección de contaminación en el medio ambiente, por lo general es implementado en circuitos de control como alarmas en las casas, sitios donde se desea prevenir altos niveles de contaminación a nivel aeróbico como industrias que manejan compuestos químicos que pueden ser nocivos también para la salud, especialmente en equipos controladores de calidad de aire en edificios/oficinas.

Este sensor se encarga de la detección de concentración de gas en diversos porcentajes, tal y como los son sus análogos MQ-3/4/5. Este producto incorpora una tarjeta que adapta el sensor para su fácil conexión en Arduino. Cuenta con un amplificador LM393 con un potenciómetro para modificar la ganancia del sensor.

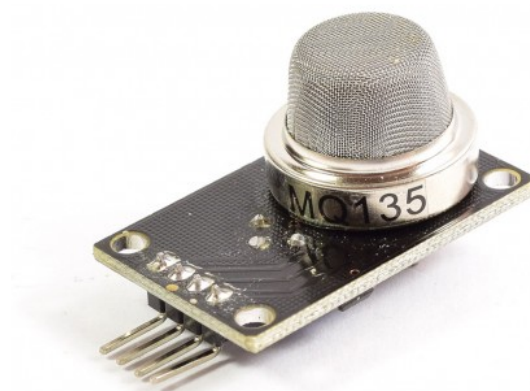


Figura 13. Sensor de gases MQ-135. *Fuente: Naylamp Mechatronics.*

PRIMER PROTOTIPO: DRON GRANJERO

Capítulo 2

Construyendo el dron cuadricóptero

Los drones cuadricópteros son muy comunes actualmente por su buen equilibrio, asimismo se puede maniobrar de muchas maneras, ya que el soporte que ofrece el usar cuatro motores permite realizar diferentes movimientos en el espacio de navegación de manera estable.

El marco para el un dron de cuatro ejes se puede imprimir en 3D o comprar un kit de piezas para armar. En aplicaciones agrícolas, los drones necesitan hacer recorridos a veces de gran distancia o seguir rutas de diferentes duraciones; el marco debe ser elegido de acuerdo a los pesos y distancias que el dron debería soportar. Si el dron ha de ser ágil, se escogen marcos pequeños y livianos (alrededor de 350 gramos) con una diagonal de motor a motor de 400-500mm. Si necesita potencia para durar más en vuelo o llevar más peso se usan marcos con diámetros más grandes entre motores. En esta pasantía se escogió el marco Tarot por su versatilidad, es liviano y se puede adecuar de varias formas.



Figura 14. Marco para dron Tarot Iron Man 650 TL65B01. Fuente: Tarot-rc.

El marco de fibra de fibra Tarot IRON MAN 650 promete ser resistente para llevar los componentes del dron agrícola a la vez que sus componentes son livianos (476 gramos). Con tableros huecos de fibra de vidrio, mecanizados en CNC, y tubos de fibra de vidrio que se fijan con tornillos. Diseñado con gran portabilidad. Permite diferentes configuraciones y distintas maneras de ubicar los componentes, como baterías y motores, para ajustar el centro de gravedad. Especialmente adecuado para hacer vigilancia, teledetección, mapeo, reconocimiento aéreo, observación de incendios, exploración de patrullaje de línea de vida útil, supervisión de granjas y otros requisitos de rendimiento móvil, carga de equipos y necesidad de más áreas de aplicación que se pueden desarrollar.

Se recomiendan las siguientes configuraciones de motores, hélices y baterías:

- Configuración de gopro (recomendado)

180-250W Motor 20 A ESC la batería 3 S,4 S 5000 hélice de 10-12 pulgadas

- Configuración micro sencilla (generalmente recomendada)

300-400W Motor 30 A ESC la batería 4 S 5000-10000 hélice de 12-14 pulgadas

- Configuración SLR (no recomendada)

Motor 400W 40 A ESC 6 S batería 5000-10000 hélice de 14-17 pulgadas

La mayoría de las piezas del marco Tarot de unen con tornillos. Se recomienda cambiar las varas de fibra de vidrio del soporte (patas) por unas de aluminio para aterrizajes fuertes.

Motores y batería

Los motores *Air Gear 350* funcionan en pares; dos de los motores van en sentido de las manecillas del reloj y los otros dos van en sentido contrario a las manecillas del reloj. Tanto el par de motores como el de hélices que vienen con un punto o marca blanca son los que giran en sentido

de las manecillas del reloj, los otros dos pares no tienen marcas. Se decidió usar estos por su popularidad en el desarrollo de drones.



Figura 15. De izquierda a derecha: ensamble del soporte del dron; fijación de un motor en una de las extensiones del dron; el marco ensamblado al soporte y dos extensiones donde van los motores. Los cables de energía conectados y distribuidos por el marco. *Fuente: Autor.*

Se deben tener en cuenta las características de los motores y los reguladores de velocidad primeramente, para seleccionar la batería. A pesar de las recomendaciones que se hagan en cuanto baterías eficientes para drones, si se observan con cuidado las especificaciones de los motores las baterías que se pueden utilizar son más; el número de celdas de la batería LiPo, las corrientes y voltajes que no son de un valor fijo, la temperatura.

Tabla 2*Especificaciones de los motores Air Gear 350 de T-Motor*

Especificaciones	
Temperatura de operación	-5°C~40°C
ESC	
Corriente	20A
Frecuencia	600Hz
Voltaje	11,1V 14,8V
Batería	3S~4S
Motor	
Tamaño del estator	22 x 13 mm
KV	920 rpm/V
Peso	54g
Hélice	
Diámetro/Rosca	9,5 x 4,5 pulgadas

Fuente: T-Motor.

Existen variadas opciones de baterías en el mercado. Las que almacenan más energía se componen de aleaciones con litio; las baterías LiPo (litio y polímero) son las más recomendadas para drones y otros proyectos electrónicos. Se pueden usar de tres celdas (3S) o de cuatro (4S), dependiendo de la potencia que se requiera, así como la corriente por hora depende de la duración de vuelo que convenga (desde 2200 mAh hasta más de 5000 mAh). Para el voltaje de los motores

Air Gear se necesita de 11,1V o 12V mínimo en la batería para elevarse correctamente. En esta pasantía se usó una batería de 2200 mAh, para una duración de vuelo entre 10 y 15 minutos.

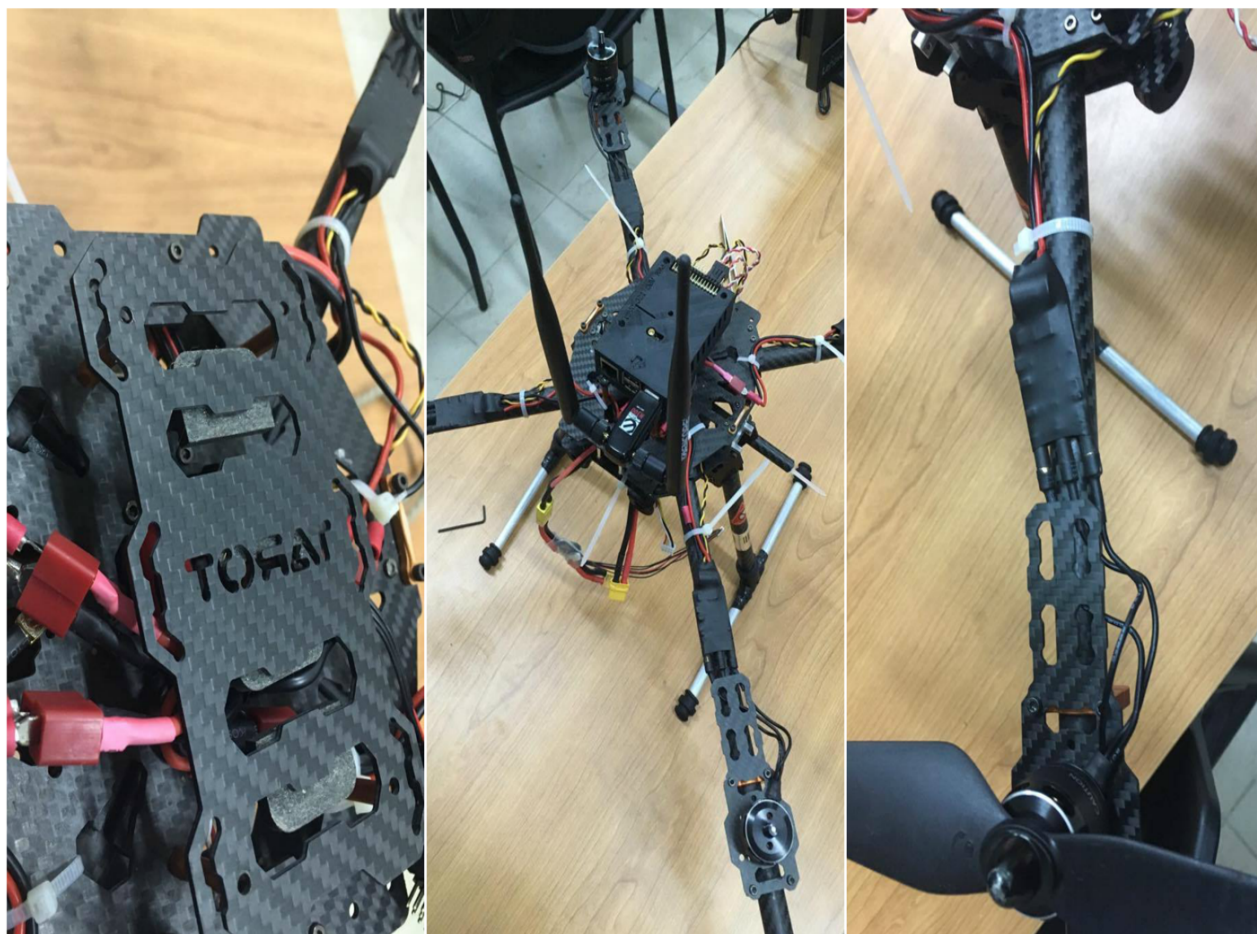


Figura 16. De izquierda a derecha: base para la tarjeta de navegación; tarjeta de navegación ensamblada en el marco; conexión del regulador de velocidad con los motores. *Fuente: Autor.*

Se acopla al marco de fibra de vidrio por debajo o por encima mediante adhesivos, soportes o amarres (figura 17).

Diseño de una carcasa protectora para las tarjetas Navio2 y Raspberry Pi 2

Se toman las medidas del acoplamiento entre las tarjetas: el área total que ocupan, la distancia entre los tornillos, y el área que ocupan todos los puertos de conexión(periféricos y alimentación). (Las medidas se encuentran en el Apéndice B).

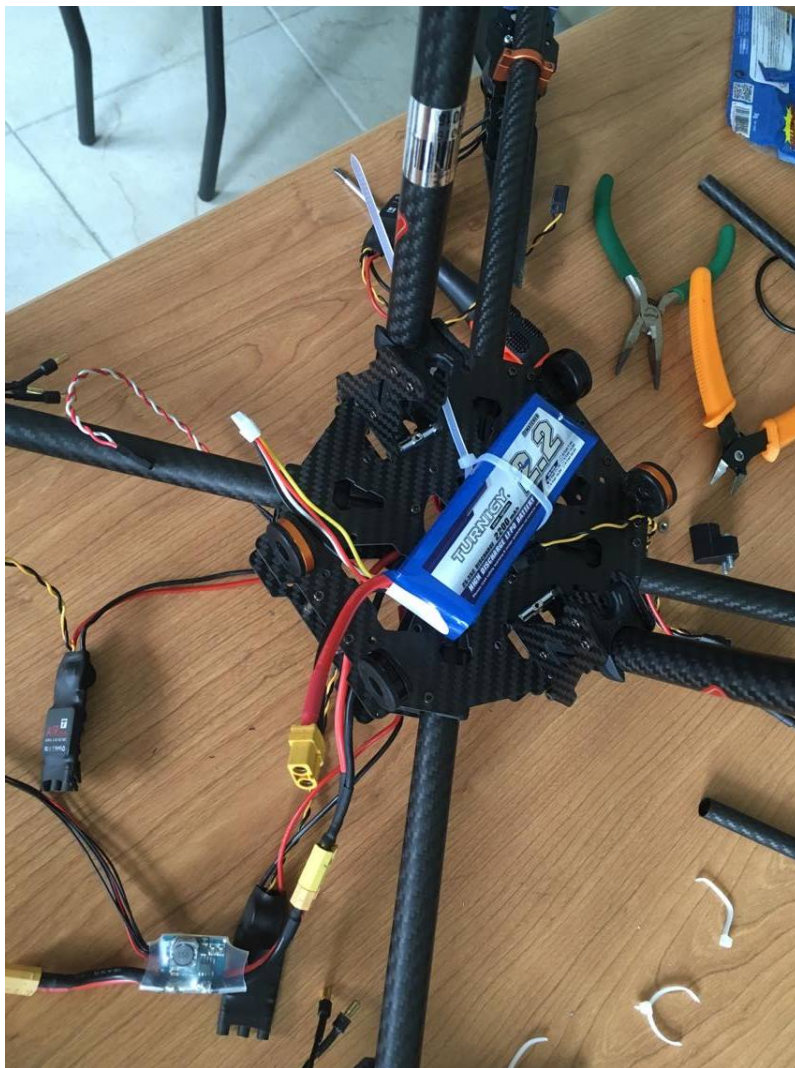


Figura 17. Batería LiPo en el marco del dron. *Fuente: Autor.*

También se pueden agregar detalles como nombrar cada puerto de la tarjeta Navio2 y una señal que indica la orientación de navegación de la misma. Estas piezas se imprimen en 3D para que sean más resistentes que unir piezas con adhesivos.

La tarjeta Navio2 no es muy popular en Colombia, así que el GIESD decidió usarla para aprender su funcionamiento y comprobar su estabilidad y múltiples herramientas. Esta tarjeta de navegación se acopla con una tarjeta Raspberry Pi 2 o 3, desde la cual se ejecuta el *software* y el controlador de la tarjeta Navio2.

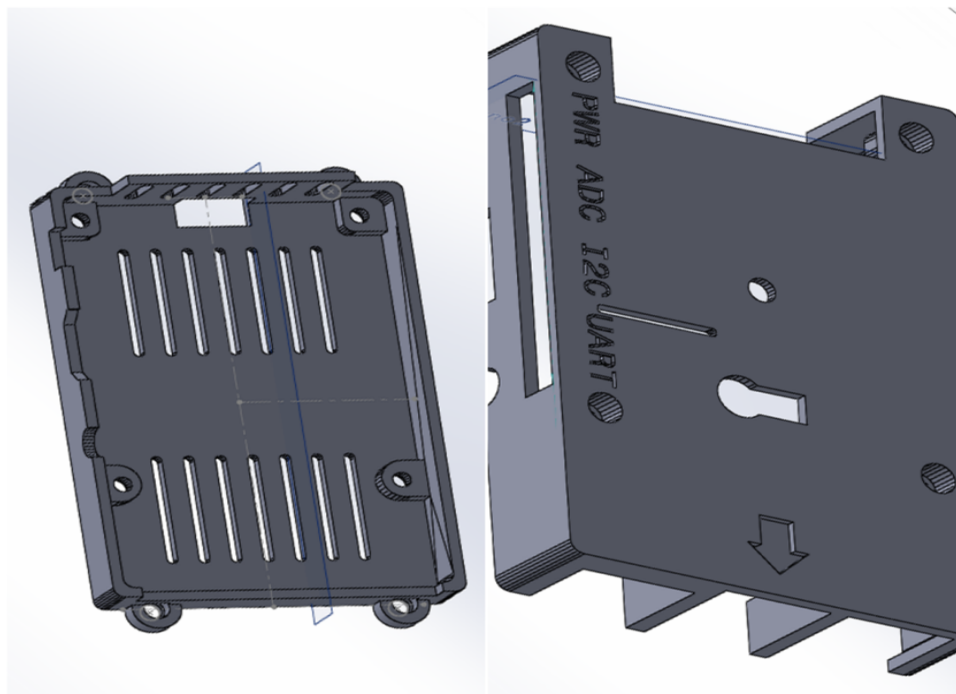


Figura 18. Diseño de la carcasa para las tarjetas Raspberry Pi 2 y Navio2. Parte inferior (izquierda) y parte superior (derecha). *Fuente: Autor.*

Aunque la carcasa proteja las tarjetas, si se desea se colocan espumillas sobre el barómetro y la brújula para más seguridad.



Figura 19. Carcasa impresa y ensamblada con las tarjetas. *Fuente: Autor.*

Configuración de hardware de la tarjeta de navegación Navio2

La configuración de la tarjeta Navio2 soporta dos tarjetas: Raspberry Pi 3 modelo B y Raspberry Pi 2 modelo B. En este prototipo se usa la Raspberry Pi 2 modelo B (Figura 5).

Tabla 3

Características de la tarjeta de control de vuelo Navio2 de Emlid

Especificaciones			
Mecánicas		Eléctricas	
Tamaño	55 x 65 mm	Voltaje de operación	4,75-5,25V
Peso	23 g	Fuente de alimentación	Módulo de energía, riel de servos, USB
Temperatura de operación	-40...+85°C	Consumo de corriente promedio	<150mA
Puertos		Integrados	
Modulo de energía		MPU9250 9DOF IMU	
UART		LSM9DS1 9DOF IMU	
I2C		MS5611 barómetro	
ADC		U-blox M8N GLONASS/GPS/Beidou	
12 salidas PWM		Co-procesador Cortex-M3 RC I/O	
Entrada PPM/S.Bus		Led RGB	

Fuente: Emlid.

Sujetar la tarjeta Navio2 en una Raspberry Pi

Se instalan espaciadores en la parte superior de la Raspberry Pi y se fijan con tornillos por la parte inferior. Se conecta la extensión para el puerto de entrada/salida gpio de 40 pines y luego se conecta la Navio2 con Raspberry Pi, fijándola con tornillos por encima.

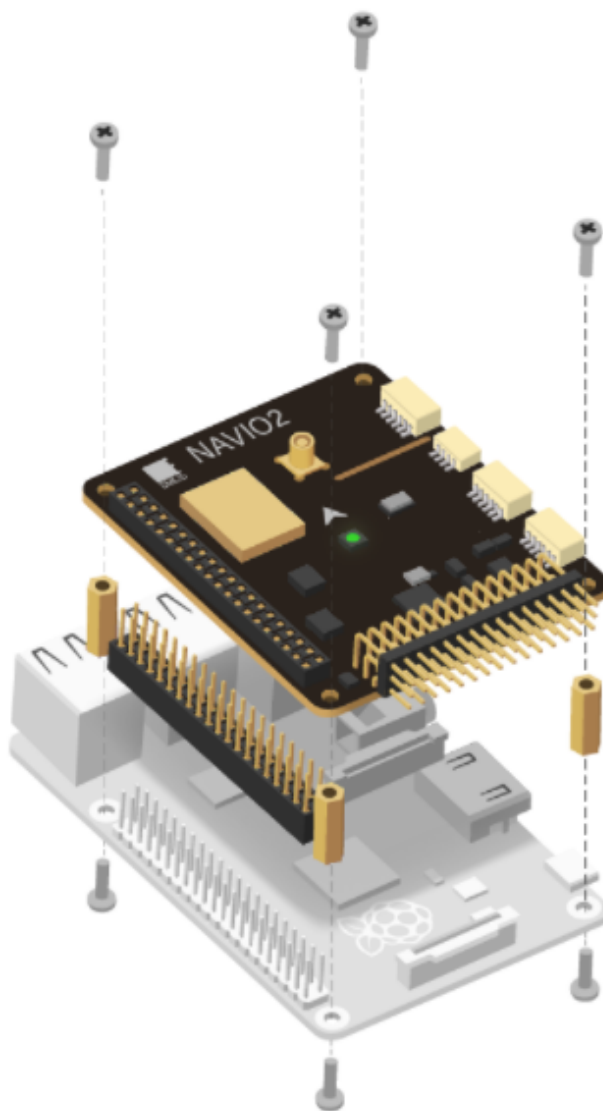


Figura 20. Ensamblaje de la tarjeta Navio2 con Raspberry Pi. *Fuente: Emlid.*

La carcasa se atornilla al tablero de fibra de vidrio en la parte superior del marco.



Figura 21. Carcasa fijada en la parte superior del marco del dron. *Fuente: Autor.*

Alimentación de la tarjeta Navio2

Por precaución, todas las fuentes de alimentación deben proveer voltajes en un rango de 4,8 a 5,3V; de otra forma podría dañarse la tarjeta Navio2 y la Raspberry Pi. Para las pruebas iniciales se puede conectar la Raspberry Pi 2 a una computadora mediante conexión USB, y esta provee energía a la Navio2.

Ya en el dron, la Navio2 debe ser alimentada por un módulo de energía que va conectado a la batería. La tarjeta Navio2 provee de energía a la Raspberry Pi.

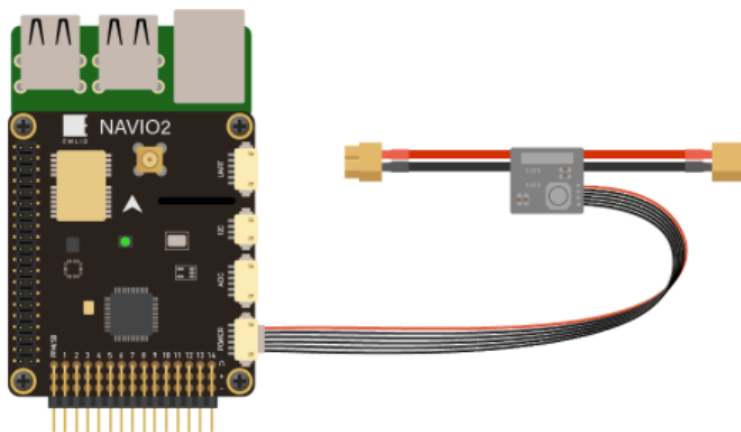


Figura 22. Módulo de energía que se conecta a la batería y provee 5V a la tarjeta Navio2. *Fuente: Emlid.*

Antena GNSS

La antena GNSS, usada para la geolocalización por GPS, va conectada al puerto MCX en la parte superior de la tarjeta Navio2.

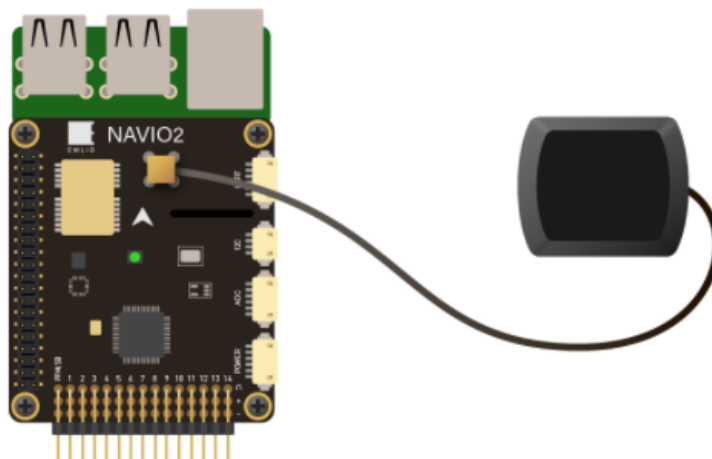


Figura 23. Antena GNSS (GPS) conectada a la tarjeta Navio2. *Fuente: Emlid.*

Los ESC

Los ESC o controladores electrónicos de velocidad van conectados a los pines de salida de Navio2, que van desde 1 a 14.

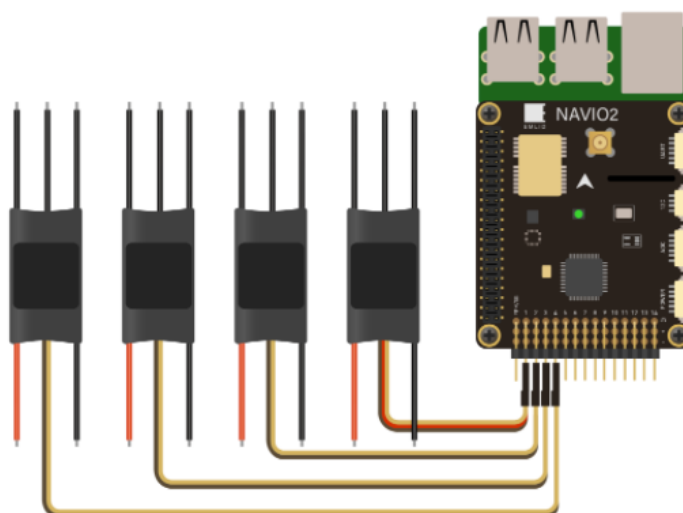


Figura 24. Esquema de conexión de los motores a los pines PWM de la tarjeta Navio2. *Fuente: Emlid.*

Módem para telemetría

Los adaptadores de red pueden ser conectadas ya sea en el puerto UART de Navio2 (figura 25, superior) o en al puerto USB de la Raspberry Pi (figura 25, inferior).

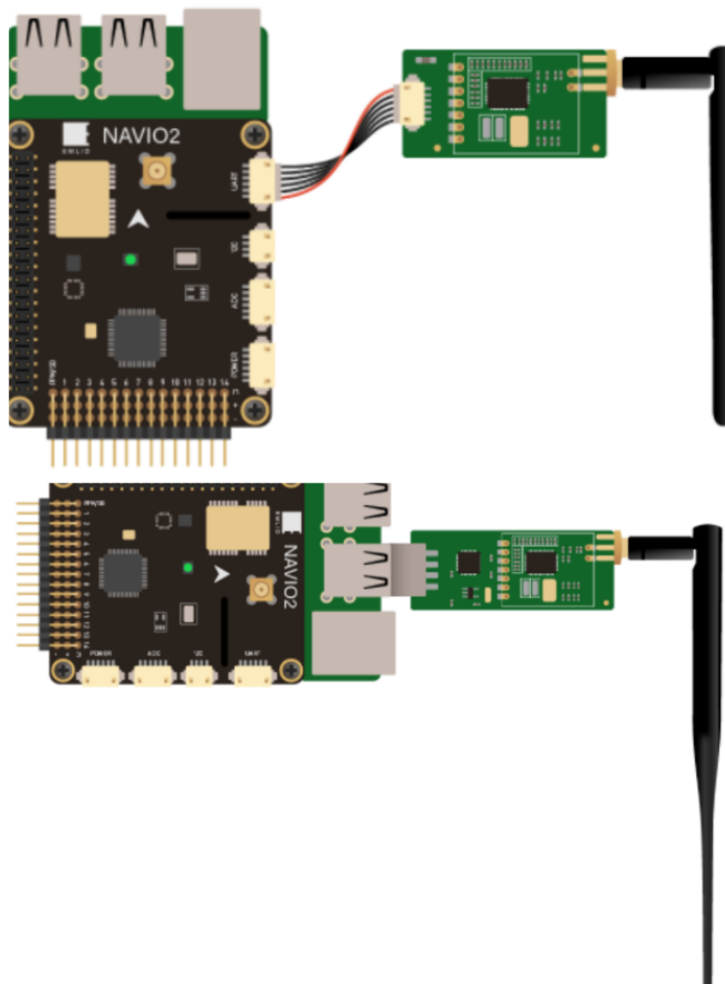


Figura 25. Los dos modos de conectar un adaptador de red inalámbrico a Navio2. Fuente: Emlid.

Protección UV para el barómetro

El barómetro MS5611, con cubierta metálica en la parte superior de la tarjeta Navio2, es sensible a La Luz ultravioleta y puede presentar cambios repentinos en la medición de la altitud bajo luz solar. Es importante cubrirlo con un pedazo de espuma, como la espuma de protección de micrófonos, o colocando las tarjetas en una carcasa protectora para protegerlas también del viento.



Figura 26. Adaptador de red inalámbrica conectada a la tarjeta Raspberry Pi 2 en el dron. *Fuente: Autor.*

Colores del led de Navio2

- Intermitente rojo y azul: Inicializando giroscopios. Mantenga el vehículo quieto y nivelado mientras inicializa los sensores.
- Azul intermitente: Desarmado, no se encontró bloqueo de GPS. Los modos de piloto automático, merodeador y retorno al lanzamiento requieren bloqueo de GPS.
- Azul sólido: armado sin bloqueo de GPS
- Verde intermitente: Desarmado (listo para armar), se ha adquirido el bloqueo GPS. Doble tono rápido cuando se desarma del estado armado.
- Parpadeo rápido verde: igual que el anterior, pero el GPS está utilizando SBAS (por lo que debería tener una mejor estimación de posición).

- Verde fijo: con un solo tono largo en el momento del armado: Armado, bloqueo del GPS adquirido. ¡Listo para volar!
- Doble parpadeo amarillo: fallas en las comprobaciones previas al armado (el sistema se niega a armarse).
- Amarillo parpadeante simple: activación a prueba de fallas de radio
- Amarillo parpadeante - con tono de pitido rápido: batería a prueba de fallas activada
- Intermitente amarillo y azul: con alto-alto-alto-bajo-tono (dah-dah-dah-doh): error de GPS o falla de GPS activada
- Intermitente rojo y amarillo: con tono ascendente: EKF o fallo de navegación inercial
- Intermitente morado y amarillo: error del barómetro
- Rojo sólido: error
- Rojo sólido con la secuencia de tono `SOS`
<[http://firmware.ardupilot.org/downloads/wiki/pixhawk_sound_files/NoSDCard_short.w](http://firmware.ardupilot.org/downloads/wiki/pixhawk_sound_files/NoSDCard_short.wav)
av>` __: Tarjeta SD faltante (u otro error SD como formato incorrecto, etc.)

Diseño de una base elevada para el módulo de GPS

La ubicación del módulo GPS es importante, porque si se encuentra muy cerca de los componentes electrónicos y el cableado del dron este puede recibir interferencias magnéticas o demoras ubicando las coordenadas. Por lo tanto, se requiere construir una base elevada como una torrecilla sobre el dron para colocar el módulo GPS aún más arriba, un poco aislado de los demás componentes obteniendo más claridad de la ubicación.

Se mide la distancia entre dos orificios de la base del dron para sujetar la torrecilla del módulo GPS con tornillos de 3 mm de diámetro.

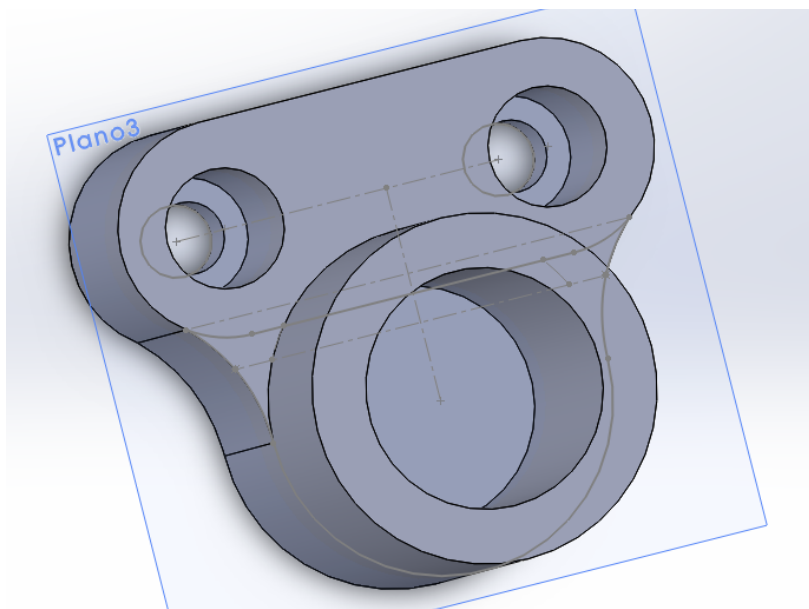


Figura 27. Diseño de la base de la torrecilla de la antena GPS. *Fuente: Autor.*

La columna única de la torrecilla puede encontrarse entre los componentes del armazón del dron. Tomando el diámetro de este y de los tornillos se diseña la base de la torrecilla. En el extremo superior va una base acorde al tamaño del módulo GPS.

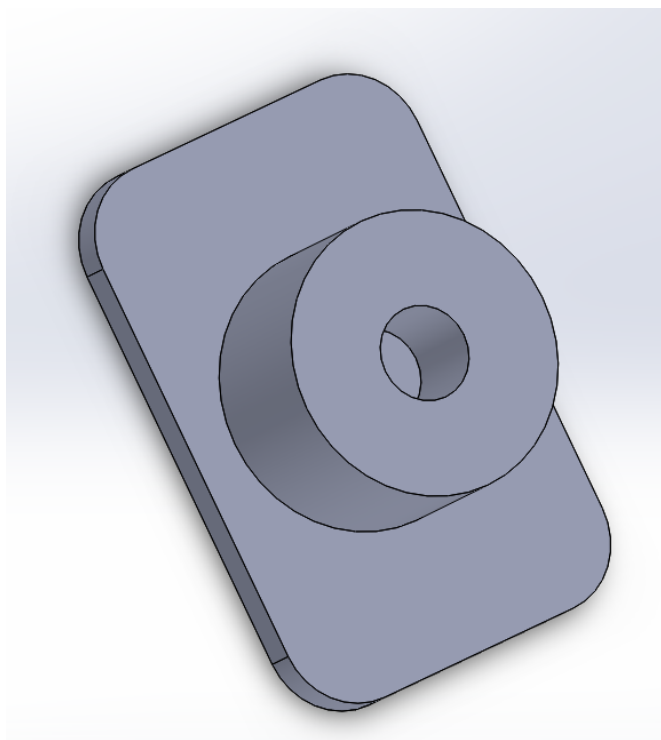


Figura 28. Base del GPS que va en la parte superior de la torrecilla. *Fuente: Autor.*

El módulo GPS se une al extremo superior de la torrecilla con cinta adhesiva de espuma. Luego se fija toda la antena en la base del dron.



Figura 29. Ensamble de la torre de la antena GNSS (GPS) en el marco del dron. *Fuente: Autor.*

Capítulo 3

Configuración de la Raspberry Pi

Las tarjetas Navio requieren del sistema operativo Raspbian pre-configurado por el fabricante para correr eficientemente. Esta funciona para las Raspberry Pi 2 y 3. El sistema operativo carece de interfaz gráfica, ya que no es primordial para aplicaciones de drones. Este se descarga desde la página oficial de *Emlid*.

Montando la imagen del SO a la tarjeta SD

Para instalar el SO *Raspbian* en la micro SD se usa la aplicación *Etcher*, que puede descargarse gratis.

Se selecciona el archivo de la imagen y la ubicación de la tarjeta SD. Se da clic en el botón *Flash!* Para iniciar la instalación y verificación de la imagen del SO. Demora unos pocos minutos.

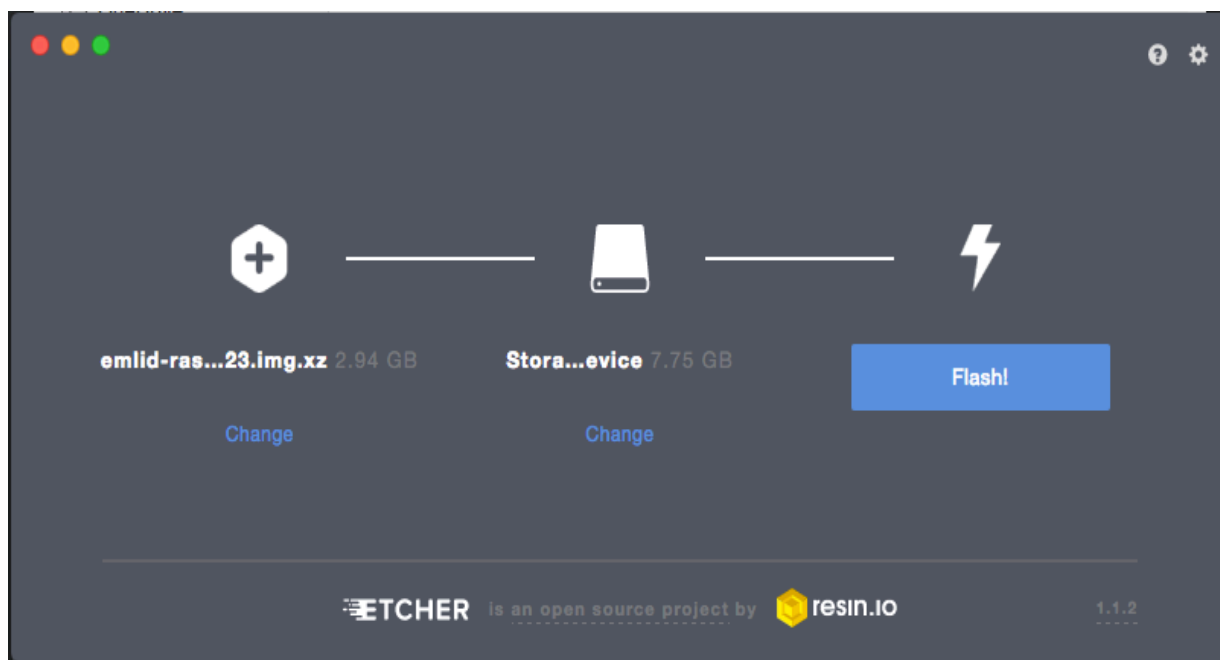


Figura 30. Instalando el SO Emlid *Raspbian* en una micro SD, utilizando *Etcher*. Fuente: Autor.

Configurando el acceso a Wi-Fi

La tarjeta Raspberry Pi 3 tiene un módulo Wi-Fi interno, mientras que la tarjeta Raspberry Pi 2 requiere de un adaptador de red USB. Los adaptadores de red soportados se muestran en la tabla 4:

Tabla 4

Adaptadores de red inalámbrica soportados por Raspberry Pi

Fabricante	Nombre	Descripción de hardware
3COM	3CRUSB10075	ZyDAS zd1211rw
7DayShop	W-3S01BLK	Ralink RT5370
	W-3S01BLKTWIN	Ralink RT5370
Adafruit	Modulo WiFi miniatura, Rpi	Realtek RTL8192CU
ALFA Network Inc.	AWUS036NEH	Ralink RT3070
ALFA Network Inc.	AWUS036NH	Ralink RT3070
ALFA Network Inc.	AWUS036H (500mW)	Ralink RT3070
ALFA Network Inc.	AWUS036H (1W)	Realtek 8187L
ALFA Network Inc.	AWUS036NHA	Atheros AR9271
ALFA Network Inc.	AWUS036NHR	Realtek RTL8188RU
AirLink101	AWLL5088 N150	Realtek RTL8188CUS
AirLink101	AWLL5099 N150	Realtek RTL8188CUS

AirLink101	AWLL6075 N	Realtek RTL8191SU
Asus	USB-N10	Realtek RTL8188SU
Asus	USB-N13	Realtek RTL8192CU
Asus	WL-167G v1	Ralink RT2571
Asus	WL-167G v3	Realtek RTL8188SU
Asus	USB-AC51	USB
AusPi Technologies	AusPi 802.11n	Ralink
Belkin	F5D7050 G	Ralink RT2571W
Belkin	F5D8053 v.6001 N	Realtek RTL8188SU
Belkin	F6D4050 v1/v2	Realtek RT3070
Belkin	F7D1101 v1	Realtek RTL8188SU
Belkin	F7D22102 N300	Realtek RTL8192CU
Belkin	F9L1001 v1 N150	Realtek RTL8188SU
Belkin	F9L1002 v1 N300	
Belkin	Surf Micro WLAN	?
Blitzz	BWU613B	Atmel AT76C505
Broadcom	Adaptador WiFi	Broadcom BCM43143
BlueProton	BT3	Realtek RTL8187L
Buffalo	AirStation N150	?
Buffalo	AirStation N300	?

Buffalo	HWU54G rev. Z2	USB
D-LINK	Airplus G DWL-G122 rev. B1	Ralink RT2571
D-LINK	Airplus G DWL-G122 rev. C	Ralink RT2571
D-LINK	Airplus G DWL-G122 rev. E	Ralink RT2870
D-LINK	DWA-131 rev E1	Realtek RTL8192EU
	PAU05	
	PAU06	
Panda Wireless	PAU07	?
	PAU09	
	TL-WN722N	
TP-LINK	TL-WN722NC	Atheros ath9k_htc
	TL-WN822N v1.1	
TP-LINK	TL-WN822N v2	Atheros AR9170
TP-LINK	TL-WN822N v3	Realtek RTL8192CU

Fuente: eLinux.

Las redes Wi-Fi pueden ser configuradas editando el archivo `wpa_supplicant.conf` localizado en la tarjeta SD. Para añadir la red simplemente se le modifican las siguientes líneas:

```
network={  
    ssid="nombre de la red"  
    psk="contraseña de la red"  
}
```

Para acceder al archivo se puede usar uno de los siguientes métodos:

Editar la configuración en la tarjeta SD

Simplemente conectando la micro SD a la computadora. Después de acceder al contenido de la tarjeta SD, se abre el archivo `wpa_supplicant.conf` ubicado en la partición `/boot` (con privilegios de administrador) y se edita como se muestra previamente.

Usando un monitor y teclado

Se conecta un monitor por HDMI y un teclado por USB a la Raspberry, se energiza y se consigue el acceso a la consola, donde se pueden modificar archivos por líneas de texto.

Usando Ethernet (Recomendado)

Se puede conectar la Raspberry con un cable Ethernet. Para encontrar la dirección IP de la Raspberry pi 2 se usa alguna aplicación o utilidad, como la aplicación móvil *Fing*, buscando el nombre navio.

Se instala en Windows la aplicación Putty, que utiliza el protocolo de seguridad SSH para iniciar la comunicación y acceder a la consola de Raspberry. Se coloca la dirección IP en Putty y se selecciona SSH, luego se da clic en *open* para empezar el enlace.



Figura 31. Dirección IP de la tarjeta Navio2 en la aplicación Fing. *Fuente: Autor.*

Se accede con el usuario “pi” y contraseña “raspberrypi”, por defecto. El archivo se encuentra en la siguiente dirección:

```
pi@navio: ~ $ sudo nano /boot/wpa_supplicant.conf
```

Luego de indicar la red que se va a usar, se debe autorizar la conexión del adaptador de red USB. La dirección para configurar el adaptador es:

```
pi@navio: ~ $ sudo nano /etc/network/interfaces
```

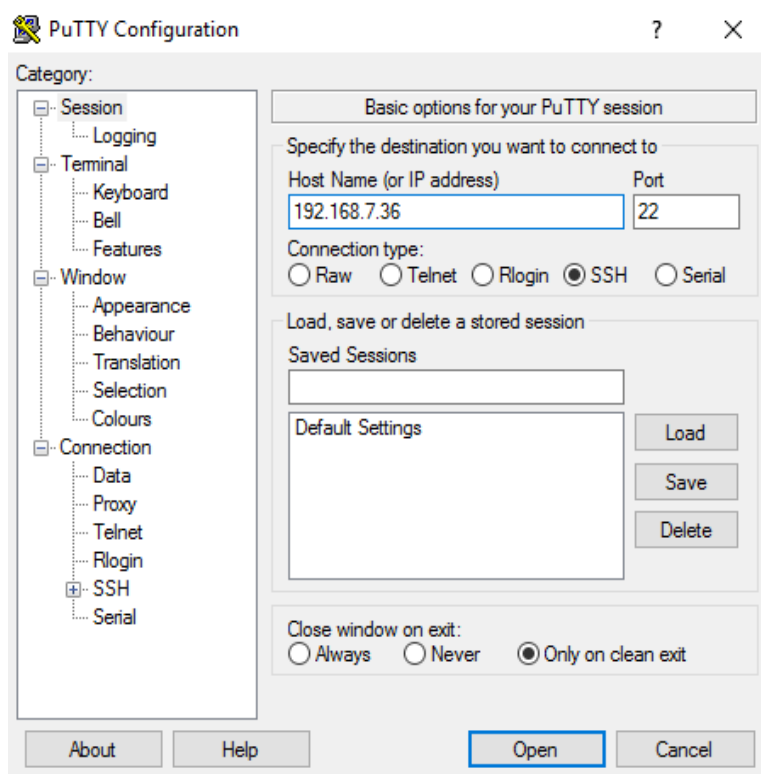


Figura 32. Accediendo a la tarjeta Raspberry Pi mediante conexión SSH. *Fuente: Autor.*

Donde se modifica el archivo de la siguiente manera:

```

auto lo

iface lo inet loopback

iface eth0 inet dhcp

allow-hotplug wlan0

auto wlan0

iface wlan0 inet dhcp

wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

```

Actualizando

Adicionalmente se pueden mantener actualizados los controladores de la tarjeta Navio2 ejecutando la línea:

```
Sudo apt-get update && sudo apt-get distracted-upgrade
```

Después de actualizar se debe reiniciar ejecutando la línea:

```
Sudo reboot -f (también: sudo shutdown -r now)
```

Instalación y funcionamiento

Se puede correr Ardupilot en las Raspberry Pi 3 o 2 con Navio. El código del piloto automático funciona directamente en Raspberry Pi. Para que Ardupilot funcione correctamente se usa la distribución Raspbian que provee Emlid, fabricante de Navio.

El sistema operativo Raspbian Emlid tiene preinstalado el Ardupilot. Incluye todos los vehículos y se basa en la versión más estable: ArduPlane para aeroplanos; ArduRover para vehículos terrestres; ArduCopter para cuadricópteros, octocópteros, etc.

Haciendo uso de la aplicación Putty, se accede por comunicación SSH a la terminal de la Raspberry Pi. Esta vez el enlace se crea por medio del adaptador Wi-Fi, buscando la nueva dirección IP asignada a la tarjeta Raspberry Pi 2 (como en la figura 31 y 32).

Una vez se consigue la comunicación SSH con la Raspberry Pi, aparece un mensaje como este:

```

📧 pi@navio: ~
login as: pi
pi@192.168.7.36's password:

# # # # # ### #####
## # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # ##### # # # #
# ## # # # # # # #
# # # # # # # ### #####

To enable ArduPilot on boot:
STEP 1:
Choose your vehicle, board and ArduPilot version using update-alternatives
(Please, read carefully all options and select appropriate one for either Navio
2 or Navio+)
- sudo update-alternatives --config arducopter
- sudo update-alternatives --config arduplane
- sudo update-alternatives --config ardurover

STEP 2:
Set your GCS IP
- sudo nano /etc/default/arducopter
- sudo nano /etc/default/arduplane
- sudo nano /etc/default/ardurover

STEP 3:
Reload configuration by issuing these commands
- sudo systemctl daemon-reload

To start ardupilot (this won't load ArduPilot on boot):
- sudo systemctl start arducopter
- sudo systemctl start arduplane
- sudo systemctl start ardurover

STEP 4:
Enable service on boot:
- sudo systemctl enable arducopter
- sudo systemctl enable arduplane
- sudo systemctl enable ardurover

To disable ArduPilot on boot:
- sudo systemctl disable arducopter
- sudo systemctl disable arduplane
- sudo systemctl disable ardurover

IMPORTANT:

To show this message one more time type "emlidtool ardupilot"

* Documentation: https://docs.emlid.com/
Last login: Thu Nov 23 21:21:04 2017 from 192.168.7.29

```

Figura 33. Pantalla de inicio después de acceder al SO mediante SSH. Fuente: Autor.

Systemd

Para lanzar el Ardupilot se usa el inicializador del sistema systemd el cual permite administrar todos los servicios y procesos. El comando principal para manejar el systemd es `systemctl`. Algunos de sus usos son: examinar el estado del sistema, administrar el sistema y los servicios, etc. Se pueden consultar más detalles en “`man systemctl`”.

Escogiendo el vehículo, la versión y el tipo de tarjeta

Para seleccionar el vehículo que será usado por defecto se debe configurar con la siguiente línea:

```
pi@navio:~ $ sudo update-alternatives --config arducopter
```

Figura 34. Ejecución del menú de selección del vehículo. *Fuente: Autor.*

Aparece una lista con los tipos de vehículo y de tarjeta Navio. En este punto se escribe el número que corresponde al cuadricóptero (*quadcopter*) y a Navio2.

Selection	Path	Priority	Status
0	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-quad	50	auto mode
1	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-coax	50	manual mode
2	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-heli	50	manual mode
3	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-hexa	50	manual mode
4	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-octa	50	manual mode
5	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-octa-quad	50	manual mode
6	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-quad	50	manual mode
7	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-single	50	manual mode
8	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-tri	50	manual mode
9	/opt/ardupilot/navio/arducopter-3.4/bin/arducopter-y6	50	manual mode
10	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-coax	50	manual mode
11	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-heli	50	manual mode
12	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-hexa	50	manual mode
13	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-octa	50	manual mode
14	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-octa-quad	50	manual mode
* 15	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-quad	50	manual mode
16	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-single	50	manual mode
17	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-tri	50	manual mode
18	/opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-y6	50	manual mode

Press enter to keep the current choice[*], or type selection number:

Figura 35. Seleccionando la opción 15 para cuadricóptero y Navio2. *Fuente: Autor.*

También se puede configurar con la herramienta emlidtool:

```
sudo emlidtool ardupilot
```

Especificando las opciones de ejecución

Se abre el archivo:

```
pi@navio: ~ $ sudo nano /etc/default/arducopter
```

Aquí se especifica la IP de la estación de tierra (computadora desde donde se manejará el dron).

```
TELEM1="-A udp:127.0.0.1:14550"  
#TELEM2="-C /dev/ttyAMA0"  
  
# Options to pass to ArduPilot  
ARDUPILOT_OPTS="$TELEM1 $TELEM2"  
  
# -A is a console switch (usually this is a Wi-Fi link)  
  
# -C is a telemetry switch  
# Usually this is either /dev/ttyAMA0 - UART connector on your Navio  
# or /dev/ttyUSB0 if you're using a serial to USB convertor  
  
# -B or -E is used to specify non default GPS
```

Figura 36. Asignando la IP de la estación terrestre para el Ardupilot. *Fuente: Emlid.*

Todas las líneas marcadas con # son comentarios y no tienen efecto. Se modifica TELEM1 para apuntar a la IP de la computadora. Y se quita el # delante de TELEM2 para que no sea un comentario.

Donde 127.0.0.1 (se modifica) es la dirección IP (figura 36) del dispositivo con la Estación de Control en Tierra - la computadora, teléfono inteligente, etc. Guardar con Ctrl+O y salir con Ctrl+X.

Recargar la configuración

Cada vez que se modifica algo como en los pasos anteriores se necesita recargar la configuración para systemd y que funcione adecuadamente.

```
pi@navio: ~ $ sudo systemctl daemon-reload
```

Activando el ArduPilot

Para iniciar el Ardupilot se escribe la línea:

```
pi@navio: ~ $ sudo systemctl start arducopter
```

Luego de iniciar el led de la tarjeta Navio 2 debe empezar a parpadear. Para detenerlo:

```
pi@navio: ~ $ sudo systemctl stop arducopter
```

Para que Ardupilot inicie con la Raspberry Pi se debe habilitar este servicio:

```
pi@navio: ~ $ sudo systemctl enable arducopter
```

Para deshabilitarlo:

```
pi@navio: ~ $ sudo systemctl disable arducopter
```

Se puede verificar si Ardupilot está habilitado al inicio o no:

```
pi@navio: ~ $ sudo systemctl is-enabled arducopter
```

Conectando la Estación de Control Terrestre GCS

Existen distintas aplicaciones de computadora para funcionar como GCS (*Ground Control Station*).

Mission Planner (Recomendado). Estación en tierra solo para Windows. Sin embargo, es el de características más completas.

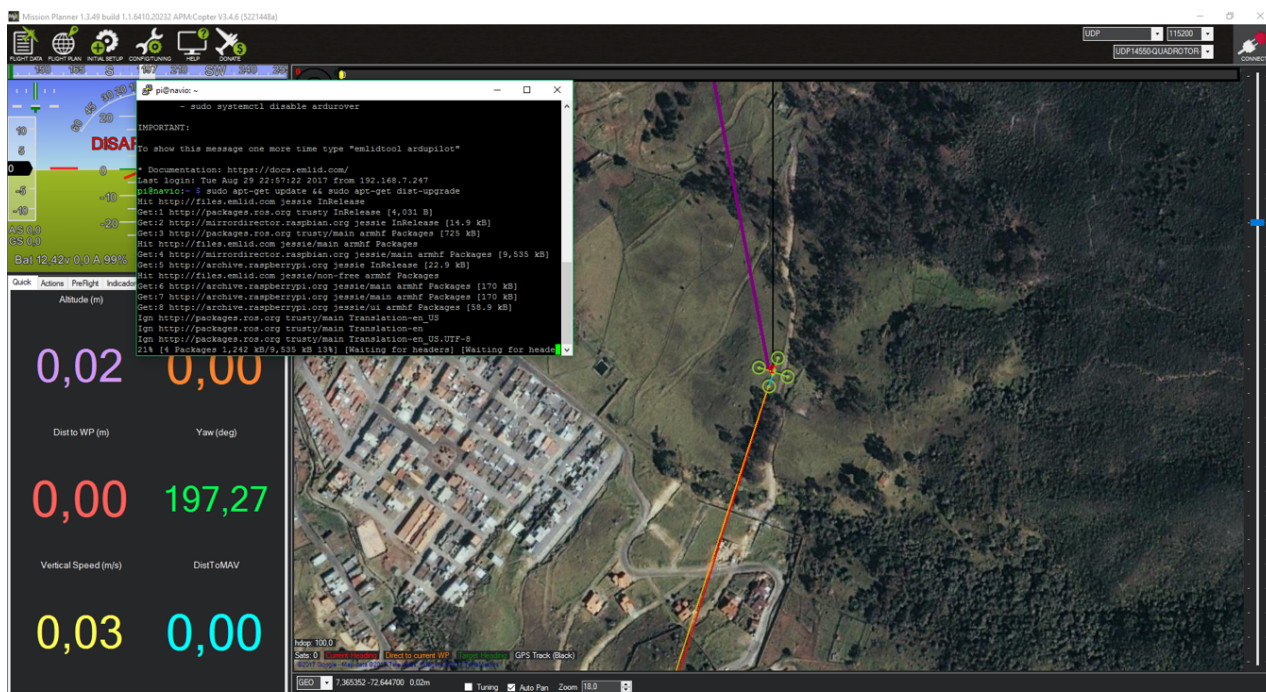


Figura 37. Entorno de la aplicación para GCS *Mission Planner*. Fuente: Autor.

QGroundControl. Es multiplataforma y está basado en pilas de vuelo Mavlink (como Ardupilot).

APM Planner. Es otra estación de tierra que funciona con ArduPilot. Siempre está atento a los puertos de conexión de telemetría por lo que puede conectarse automáticamente.

MAVProxy. Es una estación que usa una consola de códigos y está escrito Python. Es más útil para usuarios avanzados y desarrolladores.

En *Mission Planner*, el tipo conexión (UDP) y la velocidad de transmisión de los datos (115200) se configura usando los desplegables en la esquina superior derecha de la ventana.

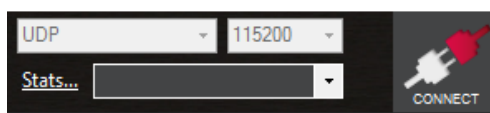


Figura 38. Configuración del tipo de conexión y velocidad de la comunicación. *Fuente: Autor.*

Una vez el adaptador de red esté conectado a la tarjeta de Raspberry y con el led parpadeando se puede dar clic en el botón de conectar; una ventana pedirá confirmar el puerto de conexión.

Después de unos segundos la conexión será establecida. Se interrumpe la conexión dándole al botón de desconectar que es el mismo para conectar.

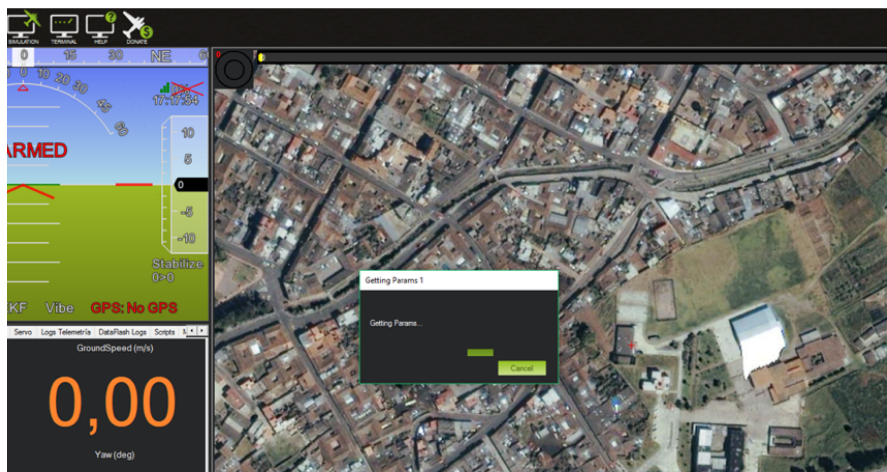


Figura 39. Completando la conexión entre *Mission Planner* y el dron. *Fuente: Autor.*

Si *Mission Planner* no establece la conexión, se revisa que el puerto sea el correcto, si el adaptador de red está conectado y encendido, o que el tipo de conexión sea el correcto (UDP).

Calibración del hardware

Como parte de la configuración inicial se necesita calibrar ciertos componentes. La orientación, el mando de control manual, brújula, los ESC, motores, y acelerómetro se configuran usando *Mission Planner*.

Además, se pueden hacer configuraciones adicionales como el monitor de la batería, sonar, sensor de velocidad del viento, flujo óptico, OSD, cámara, etc.

Calibración de brújulas

Calibración Onboard. En la misma sección de *Inicial setup*, en la opción *Compass* (brújula), aparecen las opciones de calibración.

La tarjeta de navegación Navio2 usa dos brújulas (o magnetómetros) para orientarse y estabilizarse. Por lo que la opción *Use this compass* debe estar marcada en el magnetómetro #1 y #2. Asimismo en el desplegable *Primary Compass* se escoge *Compass2* para que el magnetómetro #2 sea el primario.



Figura 40. Rotaciones para calibrar el dron. De izquierda derecha, a nivel, lado izquierdo, lado derecho, nariz abajo, nariz arriba, reverso. *Fuente: Ardupilot.*

La calibración *Onboard* (ocurre en la misma tarjeta Navio2) se encuentra en la parte inferior. Se da clic en el botón *Start* (iniciar), se rota el dron en todas las direcciones (figura 40), y se espera que la calibración se complete.



Figura 41. Calibración *Onboard* de las brújulas finalizada. Fuente: Autor.

Al final de la calibración, los valores en X,Y y Z deben aparecer en verde; se le da aceptar en la ventana emergente y se reinicia la Raspberry Pi en el dron.

Calibración en vivo. Es un método que no se usa tanto como antes, porque algunas tarjetas de navegación dependen de la estación terrestre para calcular las compensaciones de la brújula.

Esta opción se encuentra al lado de la calibración *Onboard*. La calibración se lleva a cabo igual que el método anterior.

¡Típico error de novato! intentar solventar el problema de interferencias usando papel de aluminio o cualquier otro metal a modo aislante de interferencias electromagnéticas. Un consejo, revisar bien los conocimientos sobre magnetismo y electromagnetismo. No es lo mismo corriente continua o de muy baja frecuencia que corriente alterna. Sólo se podría evitar la interferencia generada por nuestros cables mediante un Mu-metal (carísimo), a costa de perder también el

magnetismo de la tierra, por lo que tampoco es una solución. Una sugerencia puede ser el usar una brújula externa.



Figura 42. Calibración en vivo de las brújulas. Fuente: Autor.

Calibración del acelerómetro

En la sección *Inicial setup* se encuentra la opción *Accel Calibration* (Calibración del acelerómetro).

Se inicia la calibración dando clic en *Calibrate Accel* (Calibrar acelerómetro). *Mission Planner* va dando indicaciones de colocar el vehículo en diferentes posiciones de calibración. Se presiona cualquier tecla para avisar que el dron está en posición y seguir con las demás posiciones.

Las posiciones de calibración son: a nivel, del lado derecho, del lado izquierdo, nariz abajo, nariz arriba y sobre su espalda (parte superior). Como en la figura 40.

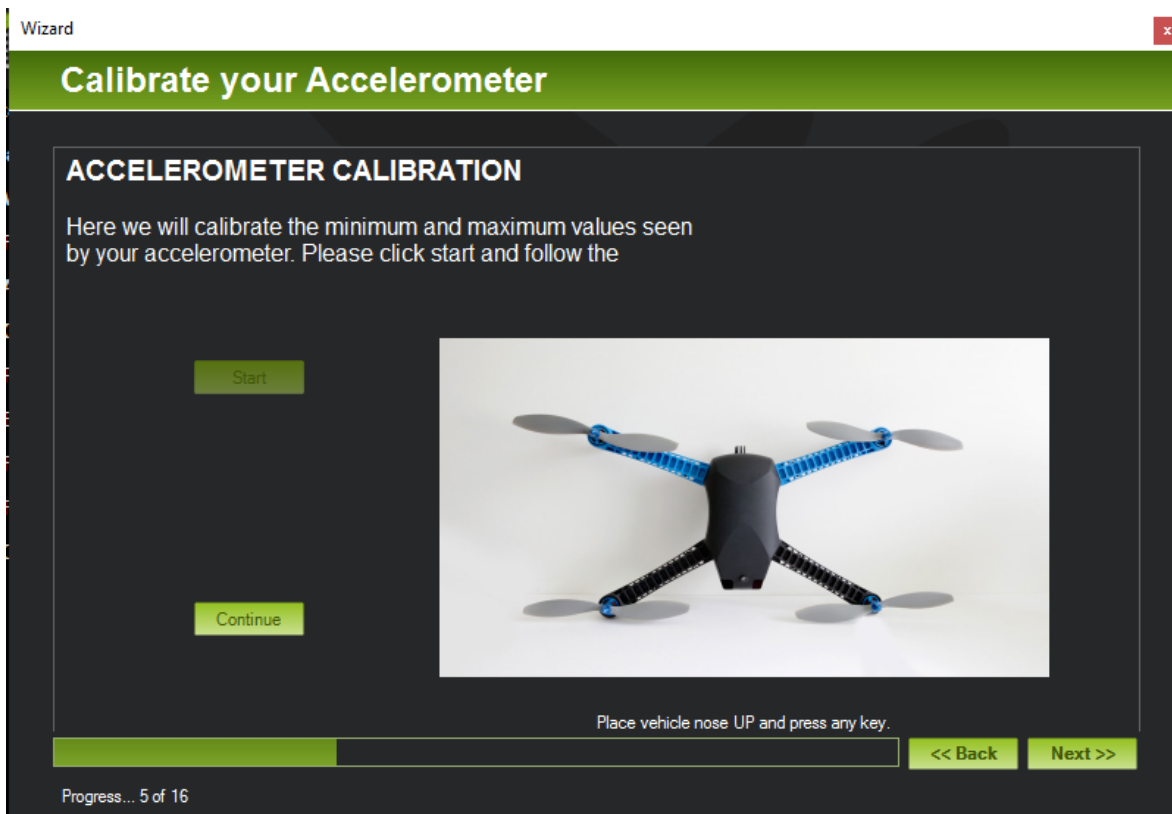


Figura 43. Calibración del acelerómetro. *Fuente: Autor.*

Al final, aparecerá un mensaje de calibración exitosa *Calibration Succesful!* o *Calibration done.*

Calibración de los controladores electrónicos de velocidad (ESC)

Los controladores electrónicos de velocidad son los responsables del giro de los motores a la rapidez pedida por la tarjeta de navegación. La mayoría de los ESC necesitan calibrarse para definir el máximo y el mínimo de valores pwm que el controlador de vuelo enviará.

Por seguridad se deben retirar las hélices de los motores e interrumpir la conexión USB si existe antes de iniciar la calibración.

Siguiendo en la sección *Inicial setup, Mandatory Hardware*, seleccionando la opción *ESC Calibration* se encuentran la calibración de los controladores electrónicos de velocidad.

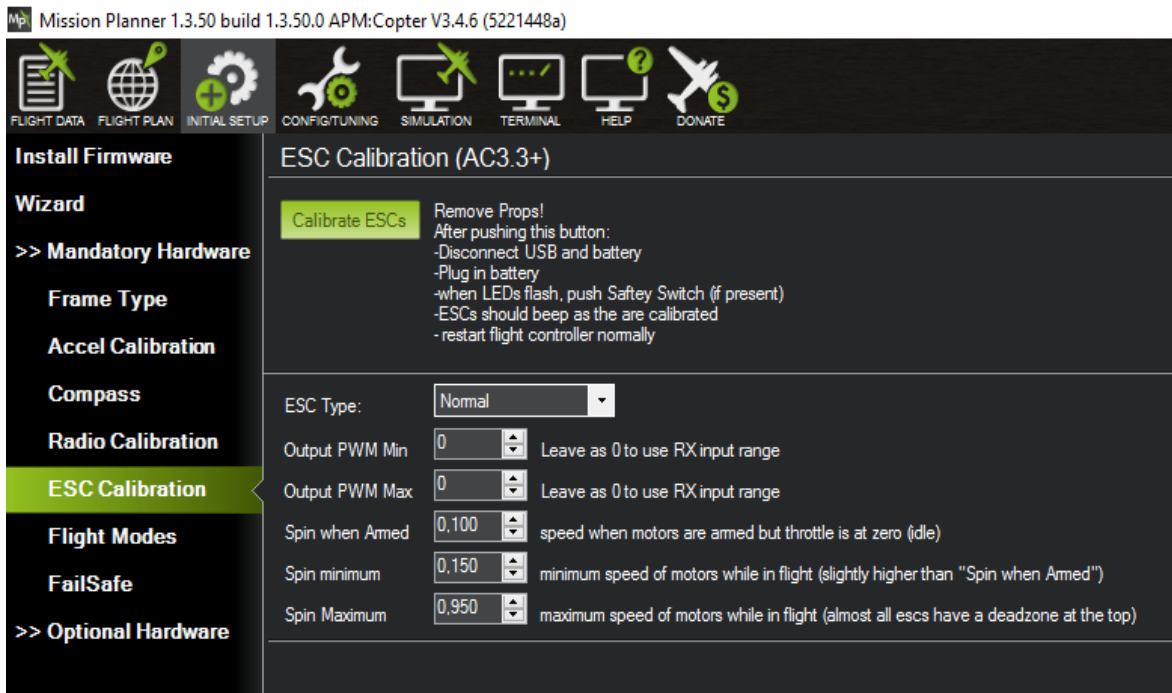


Figura 44. Calibración de los reguladores de velocidad. *Fuente: Autor.*

No se debe olvidar el retirar las hélices. Se da clic en el botón *Calibrate ESCs* y después se desconecta el cable USB (si está conectado) y la batería, se espera algunos segundos para conectar la batería de nuevo, los cuatro ESC empezarán a sonar unos pitidos un rato. Cuando deje de sonar la calibración habrá terminado y se debe reiniciar la Raspberry Pi por el SSH.

Probando los motores

En *Initial setup, Optional Hardware* (en vez de *Mandatory Hardware*), se encuentra la opción *Motor Test* (Prueba de motor). Aquí se puede configurar la máxima aceleración en porcentaje, y probar cada motor por separado o todos en simultáneo.

Se verifica que cada motor gire, seleccionando el porcentaje de aceleración y la duración de la rotación, después presionando el botón que acciona cada uno de los cuatro motores como se ve en la figura 45.



Figura 45. Prueba de movimiento de los motores en *Mission Planner*. Fuente: Autor.

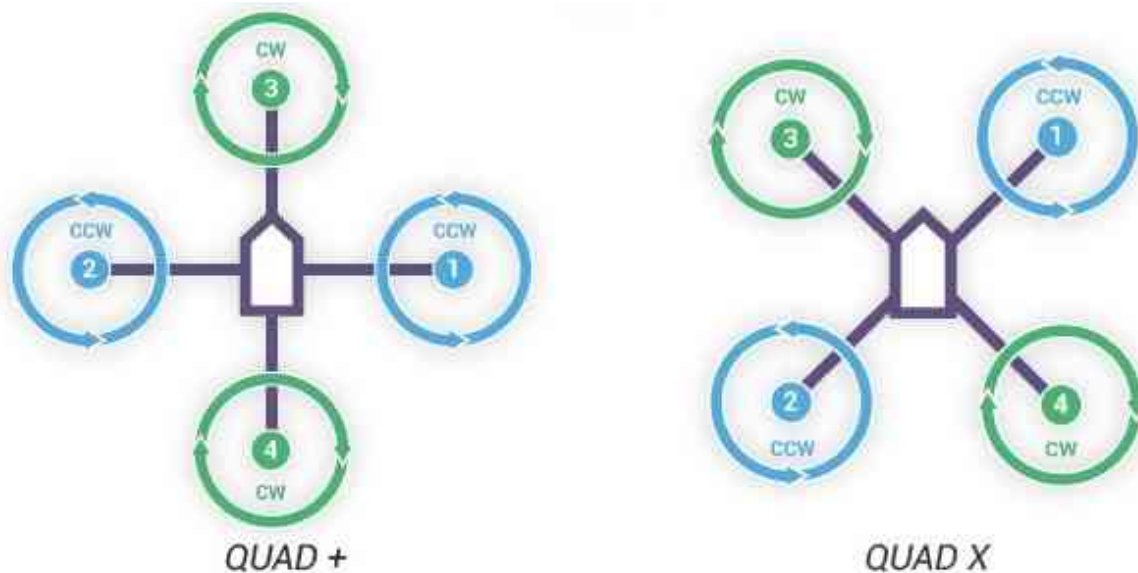


Figura 46. Distribución de los motores, dirección de giro de cada uno (CW: sentido de las manecillas del reloj, CCW: sentido contrario a las manecillas del reloj). El orden de los motores es 1=A, 4=B, 2=C, 3=D. Fuente: *Ardupilot*.

Medición de voltaje y corriente

Si se usa un módulo de energía original conectado a Navio2, se puede obtener una medición del voltaje y la corriente de la batería.

Se configura en *Initial setup, Optional Hardware, Battery Monitor* (Monitor de batería). En el primer desplegable (Monitor) se selecciona la opción 4: *Voltage and Current* (voltaje y corriente); en el siguiente desplegable (Sensor) la opción 0: *Other* (otro); en *APM Ver* se escoge la opción 4: *Pixhawk*. Se escribe 3300 mAh en la opción *Battery Capacity* ubicado a la derecha de los desplegables.

Ahora se necesita reiniciar el dron y también la conexión con *Mission Planner*. Al revisar de nuevo el menú *Battery Monitor* aparecen los valores sentados de voltaje y corriente en la batería.

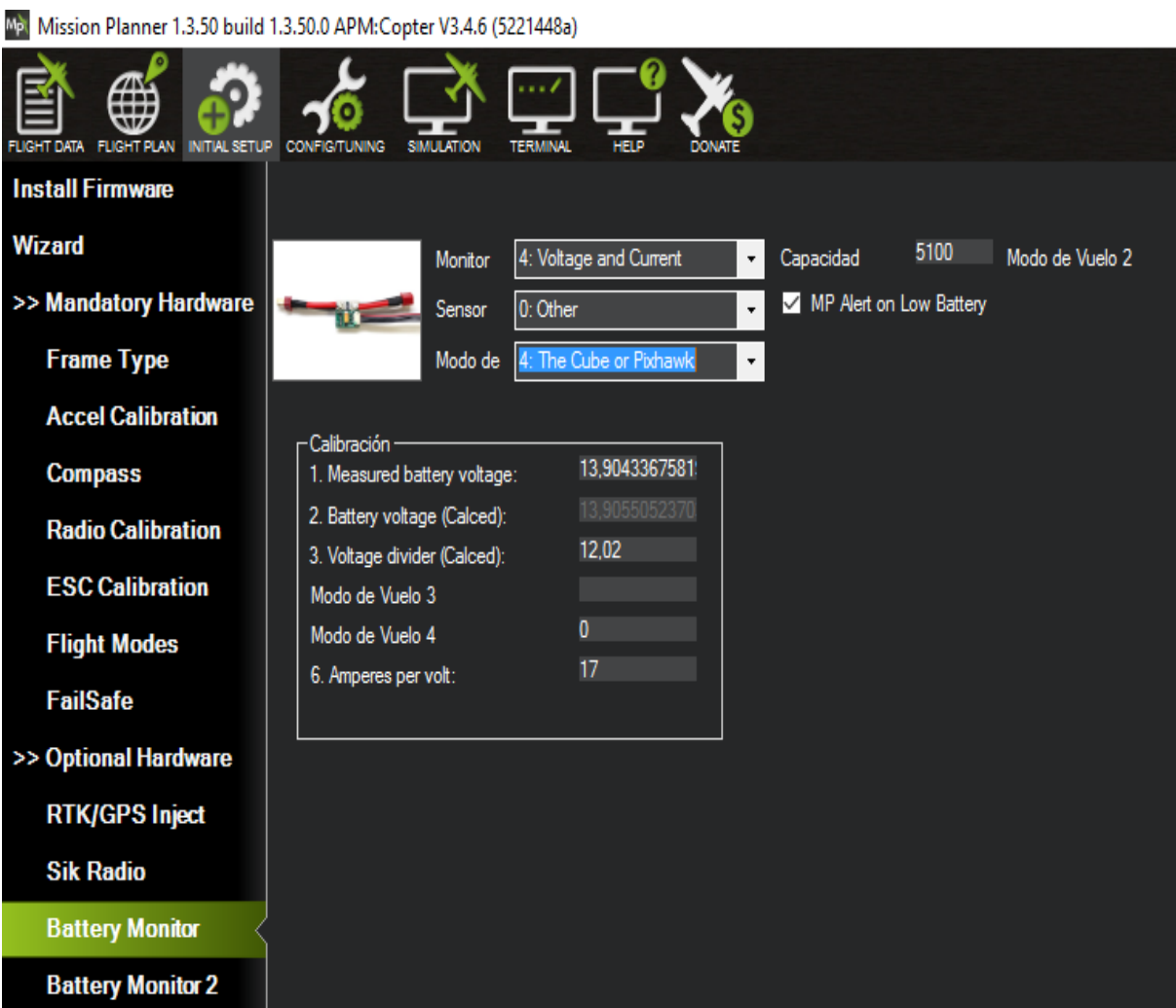


Figura 47. Configuración para medir voltaje y corriente de la batería. Fuente: Autor.

Modos de vuelo

Adicionalmente existe la opción de arreglar modos de vuelo para cambiar. En la sección *Inicial setup*, *Flight Modes*, aparecen distintos desplegables con el modo *Stabilize* por defecto, este modo es para el control manual o también para el modo automático pero que permite tomar el mando en caso de que se requiera. Es opcional cambiar los modos de vuelo, como *Stabilize*, *Loiter*, *AltHold* o *RTL*.

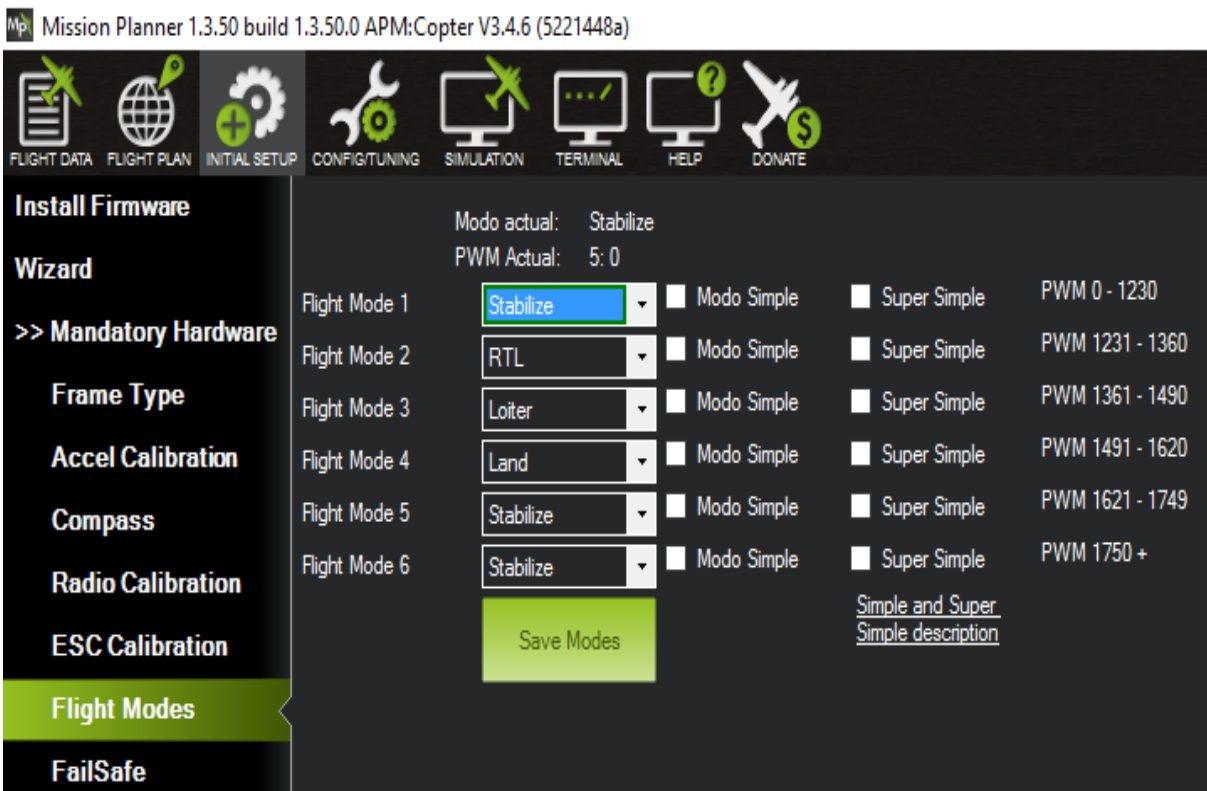


Figura 48. Modos de vuelo en *Mission Planner*. Fuente: Autor.

Configurando el ROS (sistema operativo del robot)

La imagen Raspbian que provee el fabricante Emlid viene con el ROS pre-instalado. Este ayuda a que todo el sistema de navegación y procesamiento del dron se mantenga estable. Es la inteligencia que se encarga de balancear los movimientos en vuelo; para esto se necesita acceder al terminal de la Raspberry Pi por medio de la comunicación SSH.

El ROS necesita una pequeña configuración antes de correr. Se reduce a buscar un script especial provisto en /opt/:

```
pi@navio: ~ $ echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
```

Este comando hará que la configuración del ROS se ejecute en cada inicio de sesión.

El comando *tmux*

En la configuración del ROS se necesita tener varias terminales abiertas simultáneamente.

Es por eso que se usa un terminal multiplexor con el comando *tmux*.

Antes de dividir la pantalla se debe crear una nueva sesión de multiplexor. La sesión debe llamarse “ros”.

```
pi@navio: ~ $ tmux new -s ros
```

Dentro de esta sesión se necesita operar y navegar de alguna manera entre las divisiones.

Para esto existen unos atajos del teclado universales para el *tmux*.

Atajos útiles:

- Ctrl+b + ?: muestra los atajos
- Ctrl+b + \$: renombre la sesión actual
- Ctrl+b + %: divide la pantalla horizontalmente
- Ctrl+b + “: divide la pantalla verticalmente
- Ctrl+b + o: cambia entre divisiones
- Ctrl+b + x: elimina la división

Se divide la pantalla en 4 paneles como en la figura 49. Se recomienda tomarse el tiempo necesario para practicar el dividir la pantalla y navegar entre paneles con los atajos del teclado.

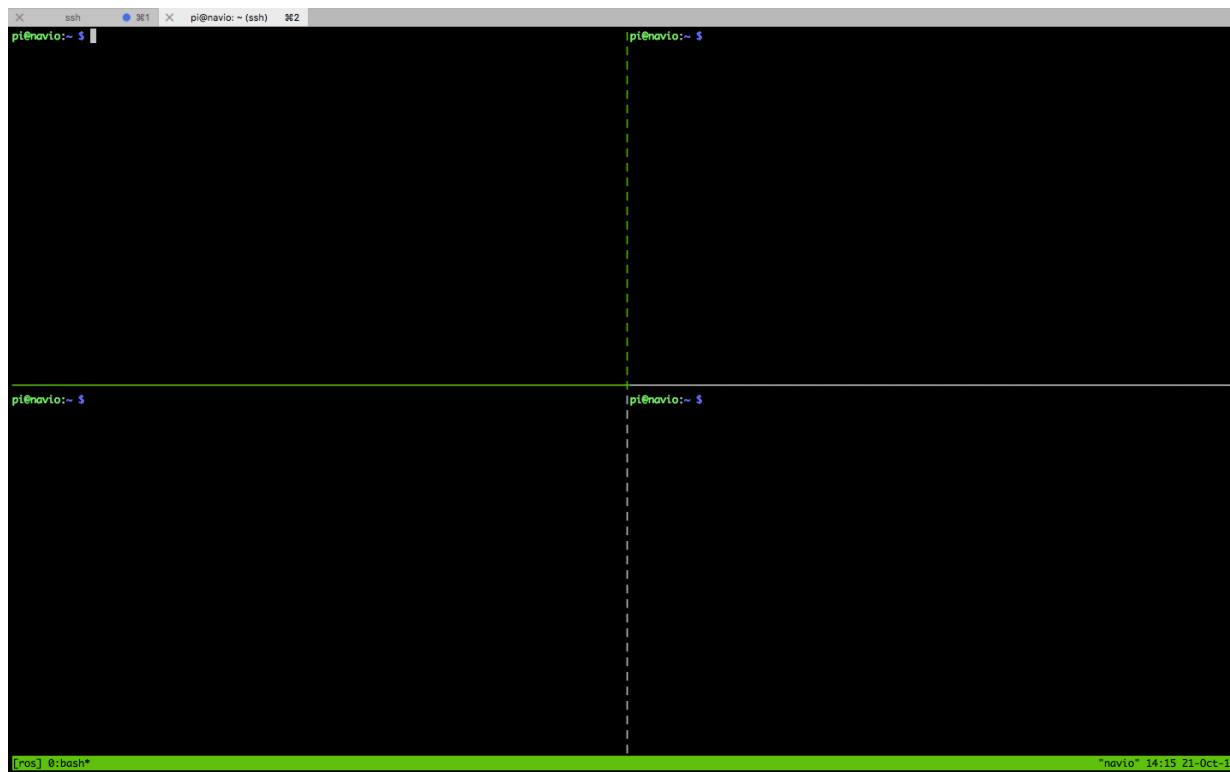


Figura 49. División de la pantalla del SSH en 4. *Fuente: Autor.*

```

pi@navio:~ $ roscore
... logging to /home/pi/.ros/log/42fae934-d063-11e7-acd9-fc8fc40c252d/roslaunch-navio-1013.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://navio:46411/
ros_comm version 1.11.21

SUMMARY
=====

PARAMETERS
 * /rostdistro: indigo
 * /rosversion: 1.11.21

NODES

auto-starting new master
process[rosmaster]: started with pid [1024]
ROS_MASTER_URI=http://navio:11311/

setting /run_id to 42fae934-d063-11e7-acd9-fc8fc40c252d
process[rosout-1]: started with pid [1037]
started core service [/rosout]

```

Figura 50. Ejecutando *roscore* en el primer panel. *Fuente: Autor.*

Ejecutando el roscore

Primeramente se inicia el ROS maestro. En el panel superior izquierdo se ejecuta el comando `roscore`.

El `roscore` es la columna vertebral del ROS. Es primordial correrlo antes de usar el ROS para la ejecución de los nodos y hacer que el enlace entre el dron y el GCS funcione.

Corriendo el Ardupilot

Se modifica temporalmente los parámetros del ArduCopter para ejecutar el ROS. Entrando al archivo del ArduCopter en `/etc/default/ardupilot`:



```

6                                     File: /etc/default/ardupilot                                     Modified
#
# Settings for ArduPilot for Linux.
# Sourced by systemd from ardupilot.service
#
# Access to ArduPilot
#
# "-A udp:127.0.0.1:14650 -C /dev/ttyAMA0"
#
# # ##### # #####
# # # # # # # #
##### ##### # # #
# # # # #####
# # # # #
# # ##### ##### #
#
# The switch (usually this is a Wi-Fi link)
#
# entry switch
# is either /dev/ttyAMA0 - UART connector on your Navio
# or /dev/ttyUSB0 if you're using a serial to USB convertor
#
# You can also specify non default GPS
#
# Run "ardupilot-config --help" for further help

```

Figura 51. Modificación temporal de la dirección IP del GCS para el Ardupilot. *Fuente: Autor.*

Se cambia la dirección IP y el puerto de telemetría a 127.0.0.1:14650.

Después se reinicia el ArduPilot. Como ya se ha iniciado en la página 73, el comando a ejecutar es el siguiente:

```
pi@navio: ~ $ sudo systemctl restart ardupilot
```

Enlazando el ROS a la estación terrestre

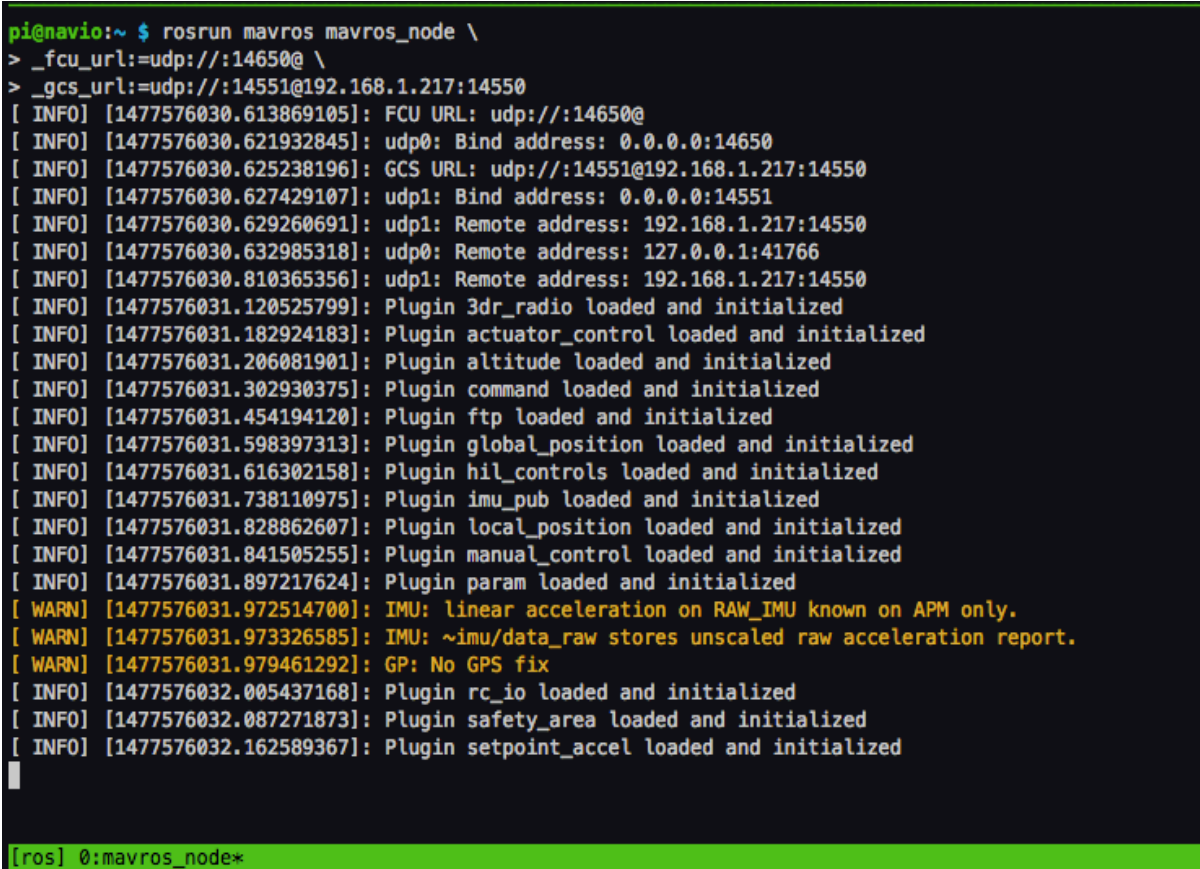
Para este paso se debe tener abierto el GCS *Mission Planner*. Este enlace se hace para que los nodos del ROS se comuniquen entre sí y con la estación de control terrestre (la computadora), facilitando la visualización de datos de los sensores, es decir, es la formación del puente UDP con el GCS.

Se corre el siguiente código en un tercer panel:

```

pi@navio:~ $ rosrund mavros mavros_node \
  _fcu_url:=udp://:14650@ \
  _gcs_url:=udp://:14551@192.168.1.217:14550

```



```

pi@navio:~ $ rosrund mavros mavros_node \
> _fcu_url:=udp://:14650@ \
> _gcs_url:=udp://:14551@192.168.1.217:14550
[ INFO] [1477576030.613869105]: FCU URL: udp://:14650@
[ INFO] [1477576030.621932845]: udp0: Bind address: 0.0.0.0:14650
[ INFO] [1477576030.625238196]: GCS URL: udp://:14551@192.168.1.217:14550
[ INFO] [1477576030.627429107]: udp1: Bind address: 0.0.0.0:14551
[ INFO] [1477576030.629260691]: udp1: Remote address: 192.168.1.217:14550
[ INFO] [1477576030.632985318]: udp0: Remote address: 127.0.0.1:41766
[ INFO] [1477576030.810365356]: udp1: Remote address: 192.168.1.217:14550
[ INFO] [1477576031.120525799]: Plugin 3dr_radio loaded and initialized
[ INFO] [1477576031.182924183]: Plugin actuator_control loaded and initialized
[ INFO] [1477576031.206081901]: Plugin altitude loaded and initialized
[ INFO] [1477576031.302930375]: Plugin command loaded and initialized
[ INFO] [1477576031.454194120]: Plugin ftp loaded and initialized
[ INFO] [1477576031.598397313]: Plugin global_position loaded and initialized
[ INFO] [1477576031.616302158]: Plugin hil_controls loaded and initialized
[ INFO] [1477576031.738110975]: Plugin imu_pub loaded and initialized
[ INFO] [1477576031.828862607]: Plugin local_position loaded and initialized
[ INFO] [1477576031.841505255]: Plugin manual_control loaded and initialized
[ INFO] [1477576031.897217624]: Plugin param loaded and initialized
[ WARN] [1477576031.972514700]: IMU: linear acceleration on RAW_IMU known on APM only.
[ WARN] [1477576031.973326585]: IMU: ~imu/data_raw stores unscaled raw acceleration report.
[ WARN] [1477576031.979461292]: GP: No GPS fix
[ INFO] [1477576032.005437168]: Plugin rc_io loaded and initialized
[ INFO] [1477576032.087271873]: Plugin safety_area loaded and initialized
[ INFO] [1477576032.162589367]: Plugin setpoint_accel loaded and initialized

```

Figura 52. Estableciendo el enlace *mavros* entre el ROS y la estación terrestre. Fuente: Autor.

En la segunda línea el puerto es 14650 como el que se acaba de modificar en el ArduCopter. En la tercera línea, después del símbolo @, 192.168.1.217:14550 es la dirección IP y el puerto con el que se puede conectar al GCS.

El cuarto panel se puede usar para revisar los parámetros procesándose en el ROS. Con la herramienta *rostopic*. Para el enlace *mavros* se corre el siguiente comando:

```
pi@navio: ~ $ rostopic echo /mavros/imu/data
```

Se muestran los datos de los sensores conectados al ROS.

```
[linear_acceleration:
| x: -0.9022118
| y: -0.00980665
| z: 9.91452315
|linear_acceleration_covariance: [8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999999999999e-08]
|---
|header:
| seq: 4608
| stamp:
|   secs: 1477062999
|   nsecs: 603571922
| frame_id: base_link
|orientation:
| x: 0.0459338450794
| y: 0.00703837202794
| z: -0.989672641308
| w: 0.135604595209
|orientation_covariance: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
|angular_velocity:
| x: 0.00016954522889
| y: 0.000584182154853
| z: -0.000778252084274
|angular_velocity_covariance: [1.2184696791468346e-07, 0.0, 0.0, 0.0, 1.2184696791468346e-07, 0.0, 0.0, 0.0, 1.2184696791468346e-07]
|linear_acceleration:
| x: -0.9022118
| y: -0.00980665
| z: 9.91452315
|linear_acceleration_covariance: [8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999999999999e-08]
|---
```

Figura 53. Muestra de datos medidos en el dron luego de establecer en enlace del ROS. *Fuente: Autor.*

Después de escribir *rostopic echo/mavros/* se presiona tabulador para ver la lista de los tópicos disponibles para revisar. Se reinicia el dron. *Fuente: Autor.*

Capítulo 4

Configuración del mando manual

Los controles de mando comúnmente usados para conducir los drones son los RC o que usan transmisores de radio. El cuadricóptero tiene un mando simplificado con acelerador, cabeceo, balanceo y timón.

En este caso se usa un mando *joystick*, para que la computadora tenga el control tanto del piloto automático como del control manual. Se configura en el mismo GCS *Mission Planner* y transmite las órdenes a través de este, por lo que usa la conexión Wi-Fi también.



Figura 54. Control de mando USB. Fuente: MundoCdWeb.

Se conecta el *joystick* por USB. En la pantalla inicial del *Mission Planner* (sección *Flight data*), en la pestaña *Actions* (acciones) se presiona el botón *Joystick*.

1. El nombre del control de mando debe aparecer en el desplegable de la parte superior.
2. El botón de habilitar, una vez presionado, le dirá al *Mission Planner* empezar a enviar comandos desde el mando.
3. En la fila *Roll*, clic en *Auto Detect* (autodetectar) y luego se menea la palanca izquierda, para el movimiento derecha-izquierda.

4. Se hace lo mismo para el *Pitch* (el cabeceo, la palanca izquierda, para el movimiento adelante-atrás) el cual se revierte seleccionando la caja de la opción *Reverse*, *Throttle* (acelerador, palanca derecha, moviéndola adelante y atrás) y *Rudder/Yaw* (timón, palanca derecha, moviéndola a la derecha e izquierda).
5. Se configuran también los botones para arme y el desarme del dron (*Arm, Desarm*), asimismo el botón para despegar (*Takeoff*).
6. Cuando se termine de configurar los botones que se deseen, se presiona el botón de guardar o *Save*.

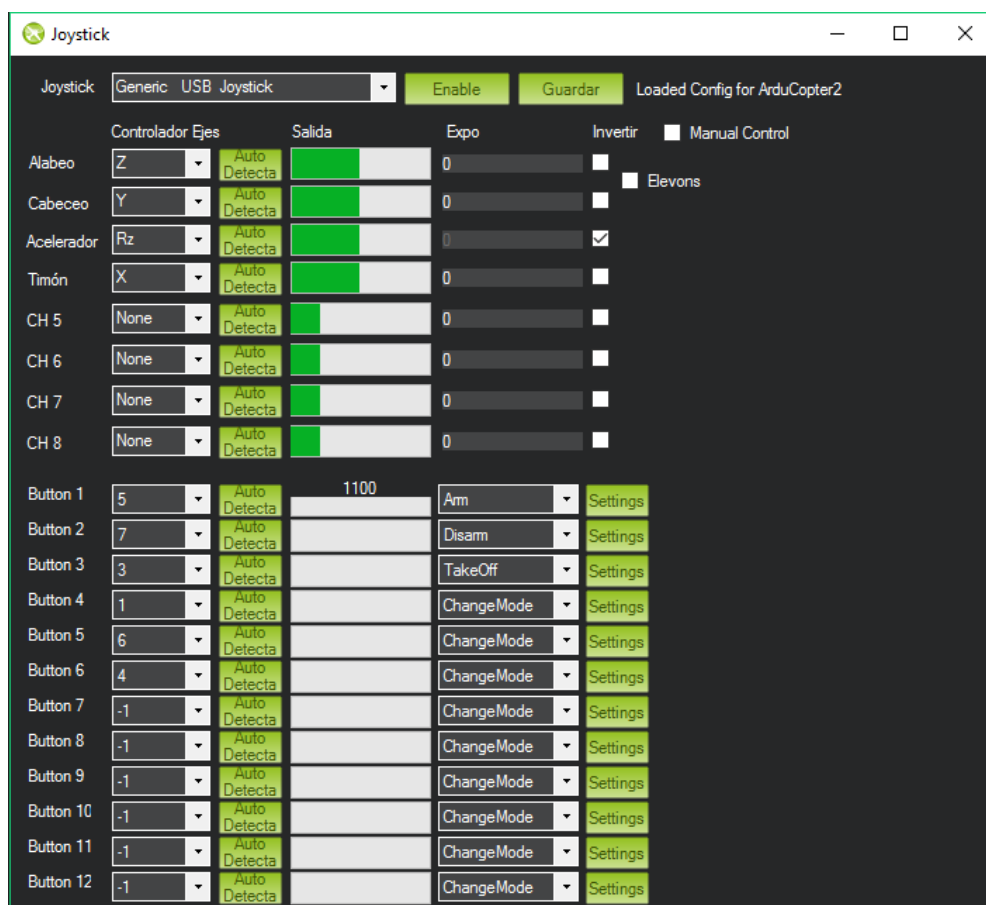


Figura 55. Configuración de botones y palancas del joystick. Fuente: Autor.

Para ajustar los parámetros del control para que funcione mejor, se va a *Config/Tuning* en el *Mission Planner*, la opción *Full Parameter List*. El parámetro *SYSID_MYGCS* debe ser 255,

aunque viene por defecto. El parámetro RC1_MIN se cambia a 1101 y RC1_MAX a 1901 (asimismo con los parámetros MIN/MAX para RC2_, RC3_ y RC4_).

Probando los controles antes del vuelo

Antes de volar la primera vez, el *joystick* se usa en vez del radio-control en la calibración del RC (Radio-control).



Figura 56. Calibración del mando USB como la radio control del dron. *Fuente: Autor.*

1. Se conecta el mando por USB.
2. En la configuración del *joystick* se da clic en el botón de habilitar o *enable*.
3. El dron debe estar conectado a *Mission Planner*.
4. Se va a *Initial setup*, *Mandatory Hardware*, *Radio Calibration*.
5. Hay un botón que se llama calibrar radio. Se acepta todo y cuando lo pida el GCS se mueven las palancas en todas las direcciones.
6. Se finaliza la calibración luego de ver que todas las barras hayan aparecido en verde y el mando USB ya queda calibrado como RC.

Así, el mando USB *joystick* queda configurado para el modo manual a través del enlace *Wi-Fi* del GCS con el dron, permitiendo el control remoto del vuelo totalmente desde la computadora en una conexión *Wi-Fi*.

Capítulo 5

Evaluación de vuelo

Cuando las calibraciones sean completadas, el dron está listo para realizar su primer vuelo.

1. El dron se conecta al GCS (*Mission Planner*).
2. Se habilita el uso del *joystick*.
3. Se verifica que no haya ningún aviso de error, la señal del dron cerca del 100%, el GPS debe mostrar la ubicación actual y la carga de la batería es óptima.
4. En la pestaña *Actions*, se configura el modo en *Stabilize* en uno de los desplegables y se presiona el botón de al lado (Configurar modo).
5. Se presiona el botón *Arm/Desarm* para verificar que el sistema de vuelo del dron se arma y los motores se preparan.



Figura 57. Revisión en parámetros del dron antes del vuelo. Fuente: Autor.

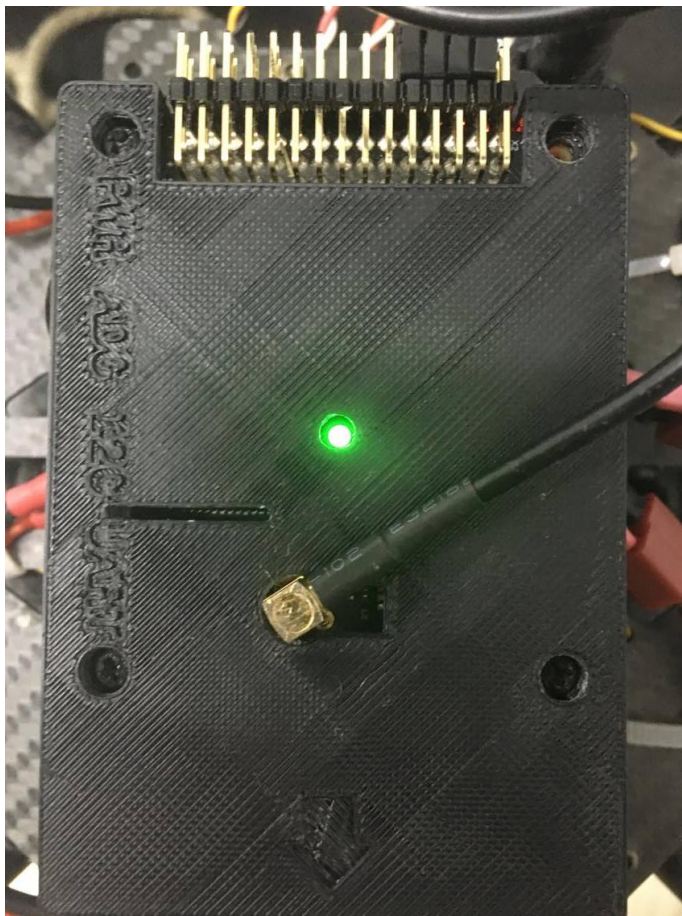


Figura 58. El dron está calibrado y listo para volar. *Fuente: Autor.*

Delimitación

Yendo a *Config/Tuning, GeoFence*, se configuran límites para prevenir que el dron se pierda si pierde la señal *Wi-Fi* gracias al GPS, está volando muy alto o se sale de los círculos de los puntos de paso.

1. El tipo de delimitación tiene diferentes opciones, aunque se recomienda limitar la altura.
2. Las acciones posibles son parecidas a las de los modos de vuelo, donde se recomienda usar RTL (regresar al punto inicial), aterrizar o ambos en caso de querer tener las dos opciones.
3. Se escribe la altura máxima que alcanza el dron o menos.

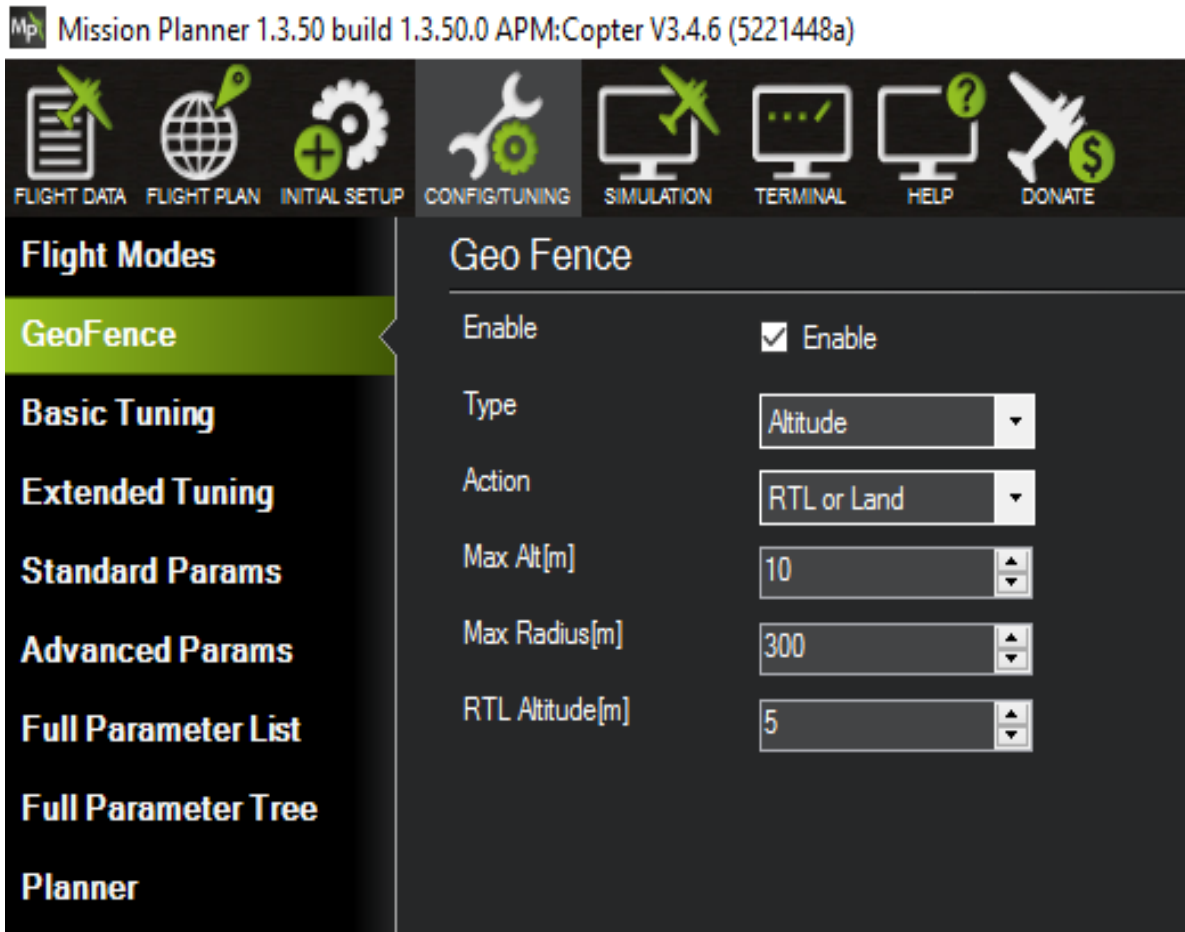


Figura 59. Delimitación de la altura del dron. *Fuente: Autor.*

Vuelo manual

1. Con el *joystick* habilitado, con el acelerador en bajo se presiona el botón asignado previamente para la acción para armar el dron (*Arm*).
2. Si el dron se arma correctamente se puede presionar el botón para despegar.
3. Cuando el dron ha despegado ya se pueden mover las palancas para conducir el dron por el aire.



Figura 60. Prueba de vuelo manual. *Fuente: Autor.*

Los motores responden muy bien, manteniendo al dron estable en el aire. El dron se mueve de acuerdo al movimiento configurado en las palancas del mando. Hay que tener buena percepción espacial para conducir el dron, ya que pueden ocurrir choques al no saber exactamente hacia donde se mueve el dron en el aire. Se debe practicar teniendo precaución. Cuando el *router* tiene varios dispositivos conectados el dron tiene retrasos en recibir los comandos del *joystick*.

Piloto automático

En la sección *Flight mode* del *Mission Planner* aparece el mapa satelital para dibujar rutas que el dron recorrerá en piloto automático. Se sitúa el punto *home* del dron, que será el lugar inicial o de despegue, si no ha sido situado automáticamente. La altitud máxima se configura en el panel de *waypoints* en la parte inferior.

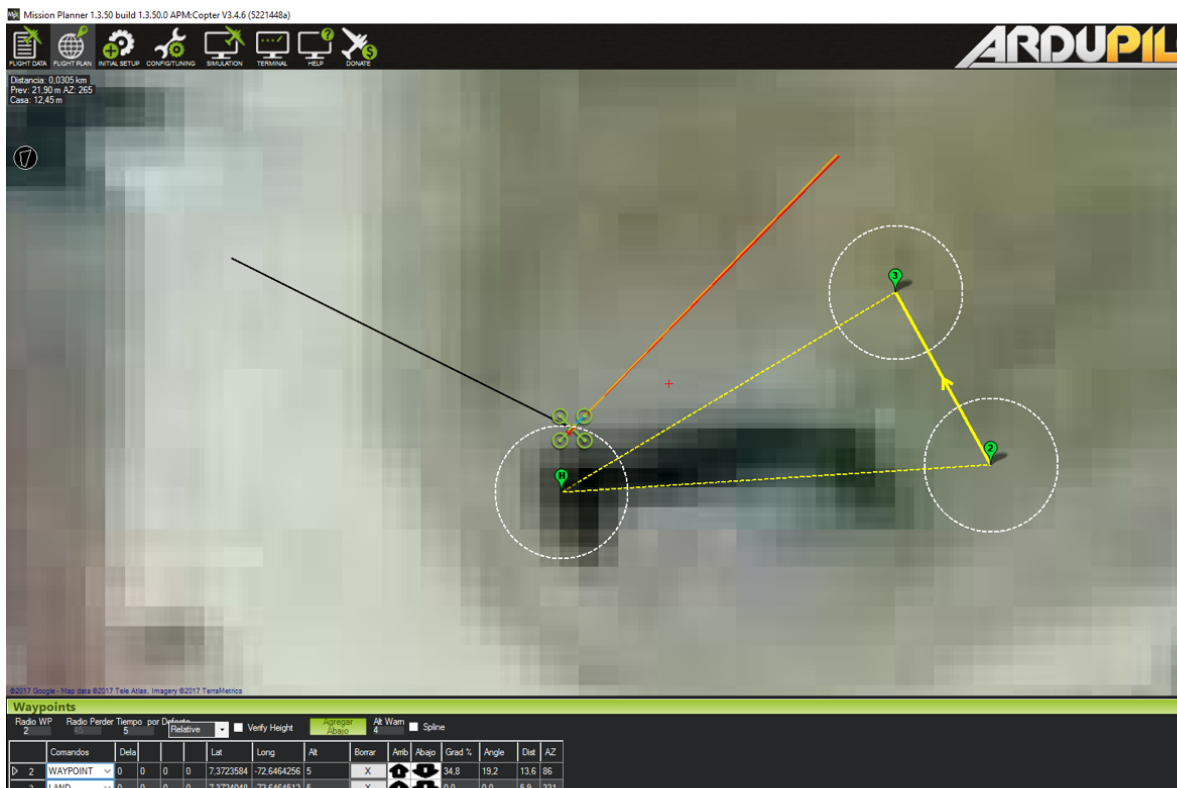


Figura 61. Creando las rutas de vuelo en *Mission Planner*. Fuente: Autor.

1. Se da clic en un lugar del mapa para crear un punto de paso (*waypoint*) de la ruta, pero este se cambia en el desplegable a despegue (*Takeoff*).
2. Luego se crea ahora sí el primer punto de paso, y los puntos que se requieran.
3. Al crear el último punto de paso, se cambia en el desplegable por aterrizar (*Land*).
4. En la parte izquierda de la pantalla se ubica el botón *Write WPs* que escribe la ruta en el Ardupilot del dron. Estos también se pueden guardar como un archivo que se puede cargar de nuevo en el *Mission Planner*.



Figura 62. Creando rutas de vuelo más complejas. *Fuente: Autor.*

En la sección de *Flight data*, en la pestaña de acciones, se determina el modo de vuelo en “Auto”, se presiona el botón *Resume Mission*. Se acepta todo, y el dron activará al piloto automático e iniciará el despegue y completará la ruta cargada.



Figura 63. Vuelo del piloto automático. De izquierda a derecha: despegue; maniobras en el aire; preparándose para aterrizar; aterrizaje. *Fuente: Autor.*

El vuelo en modo automático es más estable que el modo manual, ya que como fue aclarado anteriormente no tener una buena percepción espacial o no conocer suficiente los movimientos del dron pueden ocurrir accidentes. Tampoco está recibiendo órdenes remotas, por lo que puede volar incluso si pierde la señal de conexión, porque los puntos se cargan en la tarjeta y usando el GPS alcanzará todos los objetivos. Se puede limitar también para que no vuele fuera del rango de la señal del *router*.

De esta manera se pueden diseñar rutas para sobrevolar cosechas, por ejemplo, creando una ruta en forma de parrilla o malla para que el dron vuele sobre las líneas de cultivo en un área determinada.

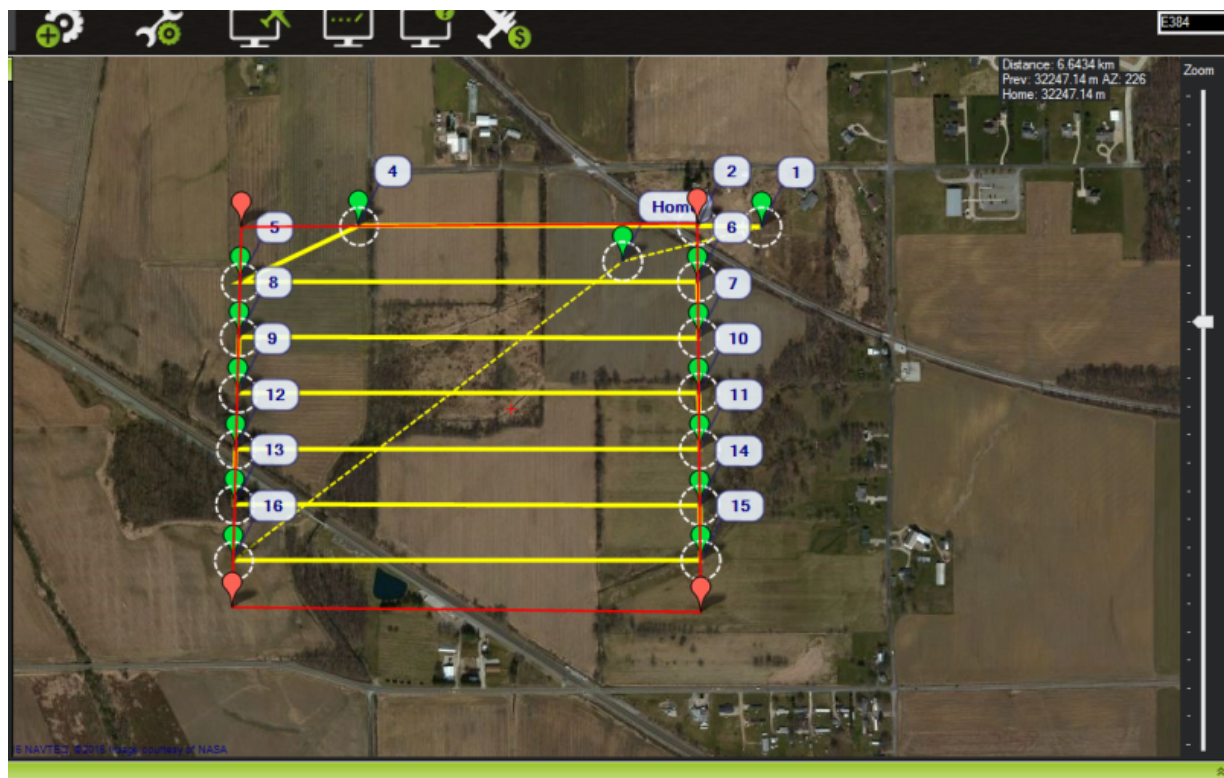


Figura 64. Plan de ruta en forma de malla o cuadrícula, para sobrevolar terrenos de cultivo. Fuente: Ardupilot.

SEGUNDO PROTOTIPO: ESTACIÓN METEOROLÓGICA

El sensor DHT22 funciona muy bien con Arduino y es compacto, además de que realiza mediciones tanto de temperatura como de humedad en el ambiente, lo que resume dos sensores en uno con una sola salida de datos conectada a Arduino.

Para el sensor de humedad del suelo se seleccionó el YL-69 de fabricación china (no hay muchas versiones de este), es barato y funciona perfecto con Arduino.

Del sensor de gotas de lluvias existen varios modelos muy similares, como el FC-37 usado por el GIESD, es fácil de conseguir y funciona con Arduino de igual manera que el sensor de humedad de suelo.

El sensor de luminosidad BH1750 fue seleccionado por su popularidad en proyectos con Arduino, que siendo complejo es compacto. Los cambios de voltaje en este sensor no definen porcentajes de luminosidad como las fotorresistencias, sino que ofrece medidas de luz por área.

Los sensores de gases son distintos, pero cada uno mide ciertos gases, se seleccionó el MQ-135 porque mide los gases más relevantes en la contaminación del aire, como el humo.

El módulo GPS de *Ublox* es de los más recomendados para usar con Arduino para dispositivos no muy grandes. El módulo para SD o micro SD para Arduino es estándar.

En el laboratorio de redes del GIESD se cuenta con todos estos sensores para proyectos y estudios, también se cuenta con varias pantallas LCD, de 16x2 y 20x4; se seleccionó de 20x4 por ser la más grande disponible.

Configuración de *hardware* de los componentes

Cada sensor tiene especificaciones eléctricas y una o más formas de conectarse al microcontrolador. Asimismo, el microcontrolador Arduino Mega tiene especificaciones eléctricas de uso.

Las características del microcontrolador Arduino Mega se muestran en la tabla 5:

Tabla 5

Características de hardware de la tarjeta Arduino Mega

Microcontrolador	ATmega2560
Voltaje de operación	5V
Voltaje de alimentación (recomendado)	7-12V
Voltaje de alimentación (límite)	6-20V
Pines digitales I/O	54 (15 PWM)
Pines de entrada análoga	16
Corriente directa en pines I/O	20 mA
Corriente directa en el pin de 3,3 V	50 mA
Memoria flash	256 KB (8 KB para bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Largo	101,52 mm
Ancho	53,3 mm
Peso	37 g

Fuente: Arduino.

Los pines sensor DHT22 de temperatura y humedad son 3: voltaje de entrada, tierra y un pin de salida digital, este va conectado a un pin digital de Arduino a la vez que se conecta una resistencia desde este pin al voltaje de entrada.

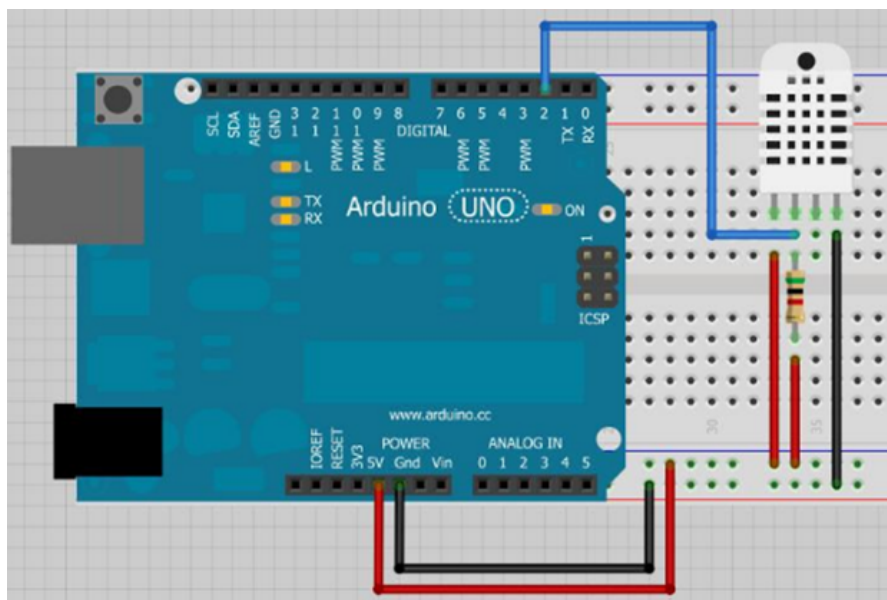


Figura 66. Esquema de conexión del sensor DHT22 al microcontrolador Arduino. Fuente: Iser.

Tabla 6

Especificaciones del sensor DHT22 de temperatura y humedad ambiente

Voltaje de operación	3,3-6V
Rango de medición temperatura	-40°C a 80°C
Precisión de medición de temperatura	< ±0,5°C
Resolución temperatura	0,1 °C
Rango de medición de humedad	0-100% HR
Precisión de medición de humedad	2% HR
Resolución humedad	0,1% HR
Tiempo de sensado	2s

Fuente: Electronilab.

En el sensor de humedad del suelo, tiene los pines para alimentarse, voltaje de entrada y tierra, y un pin de salida digital y uno de salida analógica. El pin de salida digital se puede omitir, ya que el sensor analógico es quien entrega los datos de variación en la humedad.

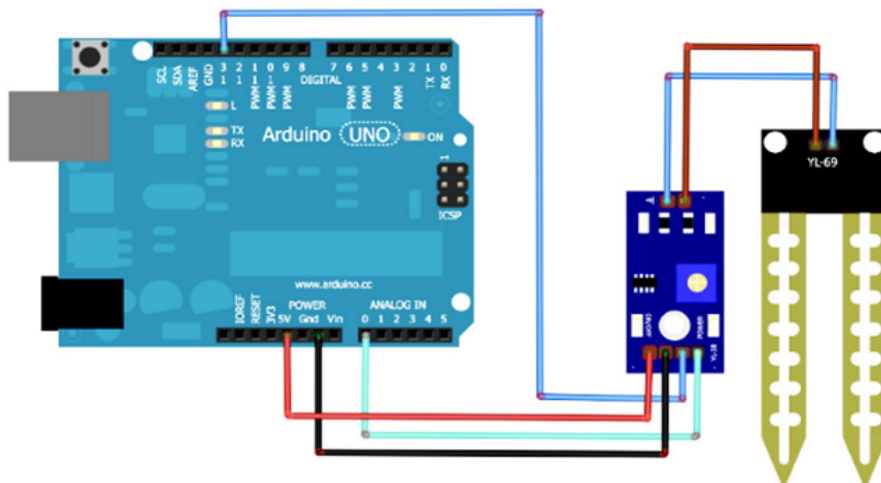


Figura 67. Esquema de conexión del sensor de humedad del suelo. *Fuente: Iser.*

Sus especificaciones técnicas se muestran en la tabla 7:

Tabla 7

Especificaciones del sensor de humedad del suelo

Voltaje de operación	3,3-5V
Dimensiones PCB	30 x 16 mm
Dimensiones de sonda	60 x 30 mm
Salidas	Digital y analógica
Chip integrado	LM393

Fuente: Electronilab.

El sensor de gotas de lluvia se configura de manera similar al anterior.

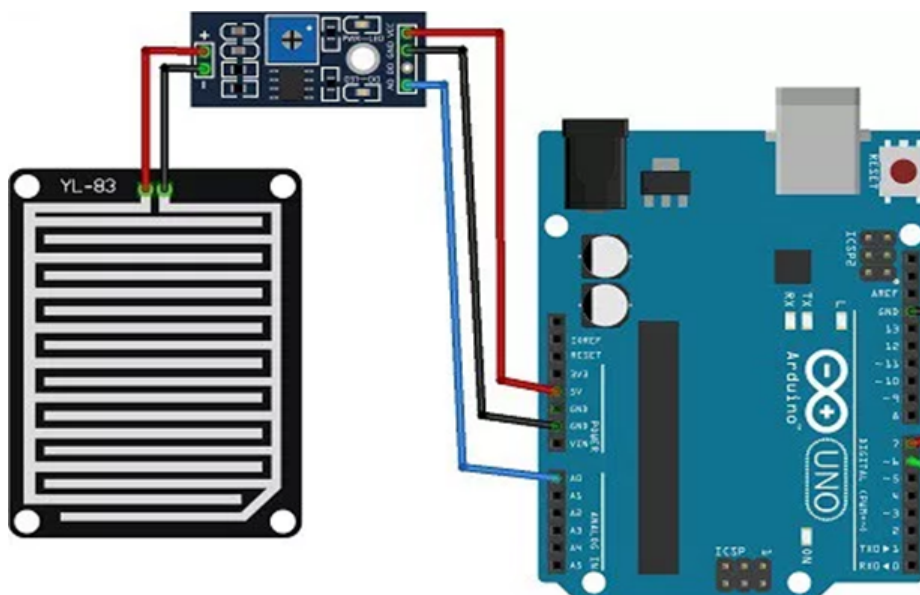


Figura 68. Esquema de conexión del sensor de gotas de lluvia al microcontrolador Arduino. *Fuente: Iser.*

Tabla 8

Especificaciones del sensor de gotas de lluvia

Voltaje de operación	3,3-5V
Tamaño PCB	32 x 14 mm
Tamaño celda	50 x 40 mm

Chip integrado LM393

Fuente: Electronilab.

El sensor de luminosidad BH1750, viene para conectarse mediante el bus de datos I2C, por el cual transfiere los datos ya transformados en *lux* al microcontrolador. Por lo tanto, las salidas SCL y SDA del sensor van a sus respectivos pines SCL y SDA en Arduino.

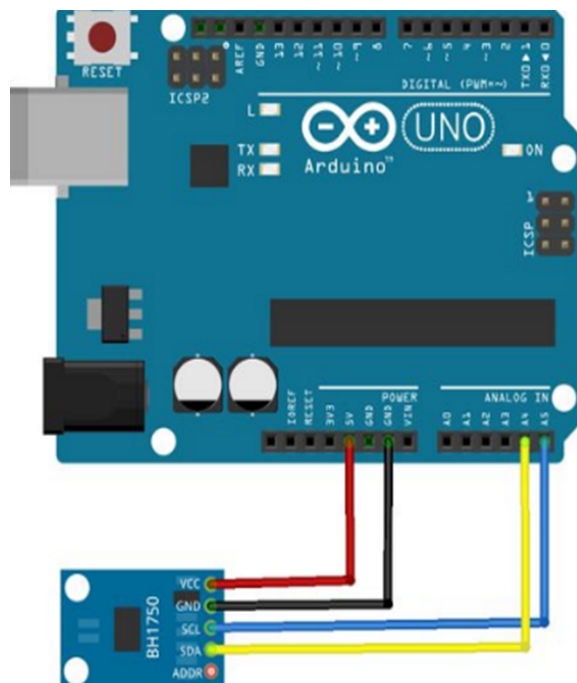


Figura 69. Esquema de conexión del sensor de luminosidad BH1750. Fuente: Iser.

Sus características son:

Tabla 9

Especificaciones del sensor de luminosidad BH1750

Voltaje de operación	3,3-5V
Comunicación	Bus I2C
Rango de medición	1-65535 lux
Consumo de energía	Muy bajo
Rechazo al ruido	50/60 Hz
<u>Espectro infrarrojo</u>	<u>baja influencia</u>

Fuente: Moviltronic.

El sensor de gases tiene una salida digital y una analógica, pero como en sensores anteriores, se puede omitir el pin digital y usar solo la salida analógica.

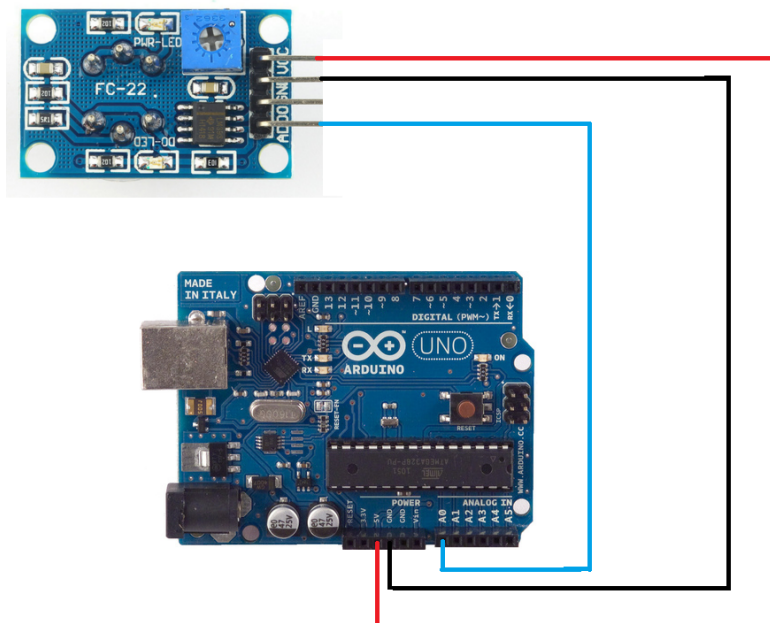


Figura 70. Esquema de conexión del sensor de gases MQ-135 al microcontrolador Arduino. *Fuente: Maker Electronics.*

Sus características son:

Tabla 10

Especificaciones del sensor de gases MQ-135

Voltaje de operación	5V DC o AC
Gases	Amoniaco, sulfuro, benceno, humo
Rango de medición	10-1000 ppm
Temperatura de funcionamiento	-20°C a 70°C
Consumo de potencia	800 mW

Fuente: Electronilab.

El módulo de GPS se conecta a uno de los puertos seriales de la tarjeta Arduino. Tiene un pin de salida (TX) y uno de entrada (RX), los cuales van a los pines RX y TX del puerto serial de Arduino, respectivamente. El puerto serial viene pre-configurado en Arduino, aunque se pueden crear otros mediante librerías.

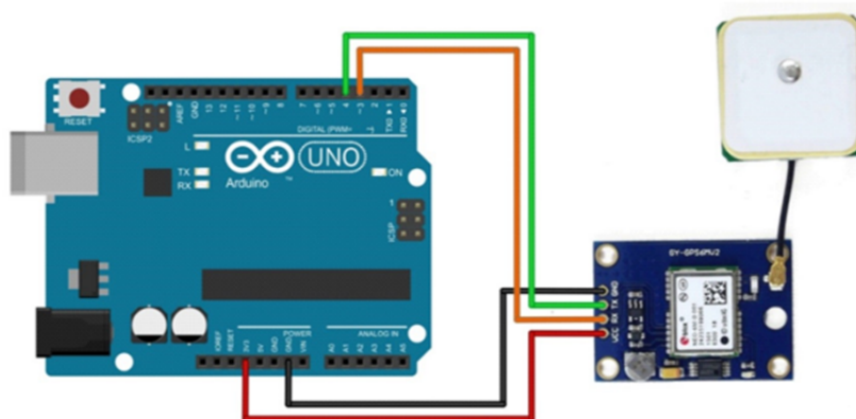


Figura 71. Esquema de conexión del módulo GPS al microcontrolador Arduino. *Fuente: Iser.*

Basado en el receptor de la marca *Ublox* modelo NEO-7M-000, el módulo incluye su antena cerámica para colocarse directamente sobre el PCB, por lo que ya viene listo para operar sin requerir más accesorios. Especificaciones técnicas:

Tabla 11

Especificaciones técnicas del módulo GPS

Voltaje de operación	3,3-5V
Módulo	NEO-7M
Comunicación	Serial
Tamaño de antena	25 x 25 mm
Tamaño de módulo	25 x 35 mm
BAUDRATE	9600

Fuente: Electronilab.

La pantalla LCD se conecta mediante bus I2C (el cual está acoplado previamente a la LCD), por lo que comparte los pines SDA y SCL de la Arduino con el sensor de luminosidad. Se alimenta con 5V. Se puede usar una LCD de cualquier tamaño, 16 x 2, 20 x 4, etc.

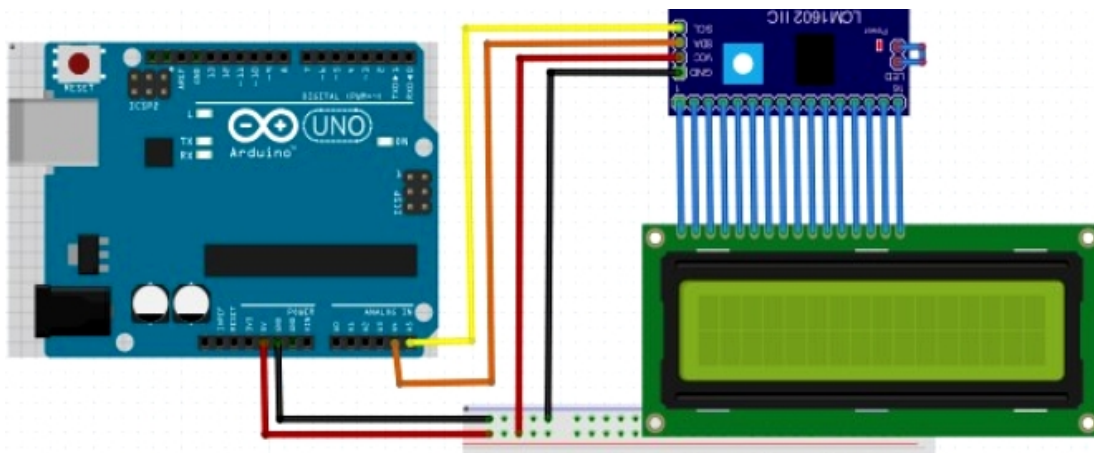


Figura 72. Esquema de conexión de la pantalla LCD mediante bus I2C. Fuente: Iser.

Ordenando los datos medidos, se coloca un módulo SD o micro SD para almacenar estos en una memoria, para revisarlos y observar las estadísticas. El módulo funciona con 3,3V o 5V de alimentación.

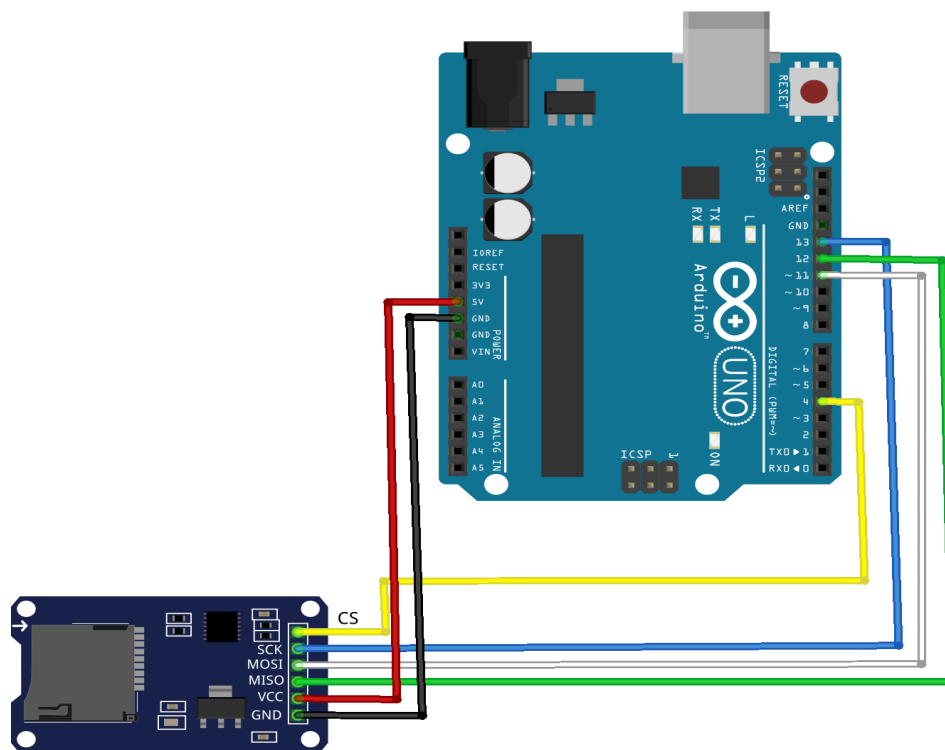


Figura 73. Esquema de conexión del módulo para micro SD al microcontrolador Arduino. Fuente: Tiendatec .

El CS puede ir en cualquier pin digital. Los demás pines del módulo van conectados a unos pines predeterminados para cada tarjeta; otra forma de conectar los pines de este módulo es conectarlos a los pines ICSP de Arduino, los cuales realizan la conexión de pines MOSI, MISO, SCK como los del módulo. Esta misma conexión puede hacerse con el módulo para SD.

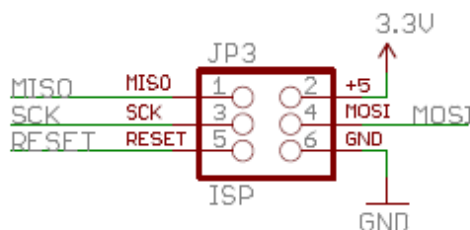


Figura 74. Pines de conexión ISP (en el ICSP de Arduino). Fuente: *Aprendiendo Arduino*.

Programación del código

El código funciona para Arduino Uno y Mega, utilizando el entorno de Arduino (IDE). El siguiente código está adaptado para una LCD 20 x 4. Teniendo en cuenta que se debe descargar cada una de las librerías propuestas o similares para que pueda cargarse a la tarjeta:

```
#include <AltSoftSerial.h> // Librería de puerto serial

#include <SD.h>

#include <TinyGPS.h> // Traductor de datos del GPS

#include <BH1750.h> // Librería del sensor de luminosidad

#include <Adafruit_Sensor.h> // Librería para el DHT22

#include <Wire.h> // Librería I2C

#include <DHT.h> // Librería para el DHT22

#include <DHT_U.h> // Librería para el DHT22

#include <LiquidCrystal_I2C.h> // Librería I2C para pantallas LCD

LiquidCrystal_I2C lcd(0x3F, 20, 4); // 0x3F es la dirección I2C asignada a la LCD
```

Se declaran todas las variables y puertos seriales a usar:

```
#define DHTPIN 2 //Pin digital del Arduino para el DHT22
#define DHTTYPE DHT22 //Definir el sensor como un DHT22
DHT dht(DHTPIN, DHTTYPE);
float temperatura;
float humedad;
TinyGPS gps;//Declarar el objeto gps
AltSoftSerial serialgps; //Se declara el puerto serial, que por defecto son 9, 8 en Arduino
Uno y 46,48 en Arduino Mega. (Tx,Rx)
//Declarar la variables para la obtención de datos
float latitude, longitude;
int year;
byte month, day, hour, minute, second, hundredths;
unsigned long chars;
unsigned short sentences, failed_checksum;
char c;
int BH1750address = 0x23; //Dirección I2C del sensor de luminosidad
byte BH1750buffer[2];
float humedads;
float lluvia;
float aire;
const int PinCs=4;//Pin de conexión al módulo micro SD
File myFile;
```

En la configuración de inicio del programa se inicializan los puertos seriales a 9600 baudios, el pin usado para conectar el CS del módulo micro SD se declara como salida. Se inicializa la LCD, luego se inicializa el módulo micro SD, si este inicia exitosamente se procede a crear el archivo CSV (Excel) y el encabezado de cada columna, por cada tipo:

```
void setup() {  
  Serial.begin(9600);  
  serialgps.begin(9600);//Iniciar el puerto serie del gps  
  pinMode(0,INPUT); // entradas análogas  
  pinMode(1,INPUT);  
  pinMode(2,INPUT);  
  pinMode(5,INPUT);  
  pinMode(4,OUTPUT);  
  Wire.begin();  
  lcd.begin();  
  lcd.backlight();  
  lcd.setCursor(12,0);lcd.print("ESTACION");  
  lcd.setCursor(7,1);lcd.print("AGRICOLA ISER");  
  lcd.setCursor(2,3);lcd.print("Iniciando...");  
  delay(3000);  
  if (!SD.begin(PinCs))  
  {  
    lcd.setCursor(0,3);lcd.print("Error al iniciar SD");  
  }  
}
```



```
else
{
  lcd.setCursor(2,3);lcd.print("Inicio exitoso");
};

//Hoja de excel
if(!SD.exists("datalog.csv"))
{
  myFile = SD.open("datalog.csv", FILE_WRITE);
  if (myFile)
  {
    myFile.println("Fecha,Hora,Latitud,Longitud,Temperatura,Humedad,Humedad del
suelo,Lluvia,Luminosidad,Calidad del aire");
    myFile.close();
  }
  else
  {
    lcd.setCursor(0,3);lcd.print("Error creando csv");
  };
};
}
```

El código repetitivo se divide por cada toma de datos de cada uno de los sensores. Se hacen dos tomas de datos por repetición, con 4 segundos de espera entre cada medición. Primero se miden y muestran en pantalla los datos del DHT22 de temperatura y humedad, del sensor de humedad del suelo y el sensor de gotas de lluvia:

```
void loop() {  
  
  delay(4000); //Tiempo de espera entre una lectura y otra. Para el DHT22 son 2 segundos  
  
  lcd.clear();  
  
  //Lectura del sensor de temperatura y humedad DHT22  
  
  temperatura = dht.readTemperature(); //Lee la temperatura  
  
  humedad = dht.readHumidity(); //Lee la humedad  
  
  lcd.setCursor (0,0) ;lcd.print("Temperatura:");  
  
  lcd.setCursor(13,0) ;lcd.print(temperatura);  
  
  lcd.setCursor(18,0) ;lcd.print((char)223);  
  
  lcd.setCursor(19,0) ;lcd.print("C");  
  
  lcd.setCursor (0,1) ;lcd.print("Humedad:");  
  
  lcd.setCursor(9,1) ;lcd.print(humedad);  
  
  lcd.setCursor(14,1) ;lcd.print("%");  
  
  //Lectura del sensor de humedad del suelo  
  
  humedads=analogRead(0);  
  
  humedads=map(humedads,0,1023,100,0);  
  
  lcd.setCursor (0,2) ;lcd.print("H. suelo:");  
  
  lcd.setCursor(10,2) ;lcd.print(humedads);  
  
  lcd.setCursor(15,2) ;lcd.print("%");  
  
}
```

```
//Lectura del sensor de lluvia;

lluvia=analogRead(1);

lluvia=map(lluvia,0,1023,100,0);

lcd.setCursor (0,3) ;lcd.print("Lluvia:");

lcd.setCursor(8,3) ;lcd.print(lluvia);
```

Luego de 4 segundos de espera, se dejan de mostrar las mediciones anteriores para mostrar ahora los datos del sensor de luminosidad, el sensor de gases y la latitud/longitud dadas por el GPS cuando está disponible el acceso a la señal satelital.

```
delay(4000);

lcd.clear();

//Lectura del sensor de luminosidad

uint16_t BH1750val = 0;

BH1750_Init(BH1750address);

if(2 == BH1750_Read(BH1750address)) {

    BH1750val=((BH1750buffer[0]<<8)|BH1750buffer[1])/1.2;

    lcd.setCursor(0,0); lcd.print("Luz:");

    lcd.setCursor(5,0); lcd.print(BH1750val, DEC);

    if((BH1750val)<100)

    {

        lcd.setCursor(9,0);lcd.print("lux");

    }

    else

    {
```

```
if((BH1750val)<10000)
{
    lcd.setCursor(10,0); lcd.print("lux");
}
else
{
    lcd.setCursor(12,0); lcd.print("lux");
};
};
};

//Lectura del sensor de calidad del aire
aire=analogRead(2);
aire=map(aire,0,1023,100,0);
lcd.setCursor(0,1);lcd.print("Calidad aire:");
lcd.setCursor(14,1);lcd.print(aire);
lcd.setCursor(19,1);lcd.print("%");

//Lectura del gps
while(serialgps.available())
{
    c = serialgps.read();
    if(gps.encode(c))
    {
        gps.f_get_position(&latitude, &longitude);//Se obtiene la latitud y la longitud
```

```

lcd.setCursor(0,2); lcd.print("Lat:");

    lcd.setCursor(5,2); lcd.print(latitude,5);

    lcd.setCursor(0,3); lcd.print("Lon:");

    lcd.setCursor(5,3); lcd.print(longitude,5);

    gps.crack_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths); //Se
obtiene la hora

    Serial.print("Fecha: "); Serial.print(day, DEC); Serial.print("/");
    Serial.print(month, DEC); Serial.print("/"); Serial.print(year);
    Serial.print(" Hora: "); Serial.print(hour, DEC); Serial.print(":");
    Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);
    Serial.print("."); Serial.println(hundredths, DEC);
};
};

```

Para el sensor de luminosidad BH1750 se usan unas funciones adicionales para obtener la dirección y establecer la transmisión de datos mediante el bus I2C, que se agregan al final del código, fuera del código repetitivo:

```

int BH1750_Read(int address)
{
    int i =0;

    Wire.beginTransmission(address);

    Wire.requestFrom(address, 2);

    while(Wire.available()) {

        BH1750buffer[i] = Wire.read();
    }
}

```

```

    i++;
}

Wire.endTransmission();

return i;
}

void BH1750_Init(int address)
{
    Wire.beginTransmission(address);

    Wire.write(0x10);

    Wire.endTransmission();
}

```

Después de hacer todas las mediciones, el código finaliza con el almacenamiento de los datos organizadamente en la memoria SD o micro SD. Guarda un archivo de solo texto, donde cada muestreo se guarda en filas; otro archivo es de extensión .csv que se usa para importar los datos a Excel, con encabezados de columna y los datos organizados en tabla:

```

//Carga de datos a la micro SD

myFile = SD.open("datalog.txt", FILE_WRITE);//abrir el archivo

if (myFile)
{
    myFile.print("Fecha: ");

    myFile.print(day, DEC); myFile.print("/");

    myFile.print(month, DEC); myFile.print("/"); myFile.print(year);

    myFile.print(" Hora: "); myFile.print(hour, DEC); myFile.print(":");
}

```

```
myFile.print(minute, DEC); myFile.print(":"); myFile.print(second, DEC);  
  
myFile.print("."); myFile.print(hundredths, DEC);  
  
myFile.print(", latitud: ");  
myFile.print(latitude);  
  
myFile.print(", longitud: ");  
myFile.print(longitude);  
  
myFile.print(", temperatura: ");  
myFile.print(temperatura);  
  
myFile.print(", humedad: ");  
myFile.print(humedad);  
  
myFile.print(", humedad del suelo: ");  
myFile.print(humedads);  
  
myFile.print(", lluvia: ");  
myFile.print(lluvia);  
  
myFile.print(", luminosidad: ");  
myFile.print(BH1750val,DEC);  
  
myFile.print(", calidad del aire: ");  
myFile.println(aire);  
  
myFile.close(); //cerrar el archivo  
  
}  
  
else  
  
{
```

```
lcd.clear();

lcd.setCursor(3,2);lcd.print("Error al abrir");

lcd.setCursor(3,3);lcd.print("el archivo txt");

};

delay(100);

//Datos en hoja de excel

myFile = SD.open("datalog.csv", FILE_WRITE);//abrir el archivo

if (myFile)

{

    myFile.print(day, DEC); myFile.print("/"); myFile.print(month, DEC); myFile.print("/");

myFile.print(year);

    myFile.print(",");

    myFile.print(hour, DEC); myFile.print(":"); myFile.print(minute, DEC); myFile.print(":");

myFile.print(second, DEC); myFile.print("."); myFile.print(hundredths, DEC);

    myFile.print(",");

    myFile.print(latitude);

    myFile.print(",");

    myFile.print(longitude);

    myFile.print(",");

    myFile.print(temperatura);

    myFile.print(",");

    myFile.print(humedad);

    myFile.print(",");
```



```
myFile.print(humedads);  
  
myFile.print(",");  
  
myFile.print(lluvia);  
  
myFile.print(",");  
  
myFile.print(BH1750val,DEC);  
  
myFile.print(",");  
  
myFile.println(aire);  
  
  
myFile.close(); //cerrar el archivo  
}  
else  
{  
    lcd.clear();  
    lcd.setCursor(3,2);lcd.print("Error al abrir");  
    lcd.setCursor(3,3);lcd.print("el archivo csv");  
};  
} //Fin
```

Si se desea, se pueden limitar las mediciones que se almacenan condicionando el código del módulo para que se ejecute cada cierto tiempo. (Véase el Apéndice A, para el código ininterrumpido).

Capítulo 7

Construcción de la estación meteorológica móvil

En este punto se debe diseñar la estructura (caja) que contiene los sensores, el microcontrolador, LCD, etc. Asimismo como un soporte para este. El diseño puede ser de cualquier manera, tal que soporte y contenga todos los componentes y estos funcionen normalmente acoplados a la estructura. (Véase Apéndice B)

Todas las partes de imprimen en 3D o se cortan en 2D por CNC. Si la impresora 3D tiene un espacio de trabajo reducido, se divide la caja de los sensores en partes acoplables.

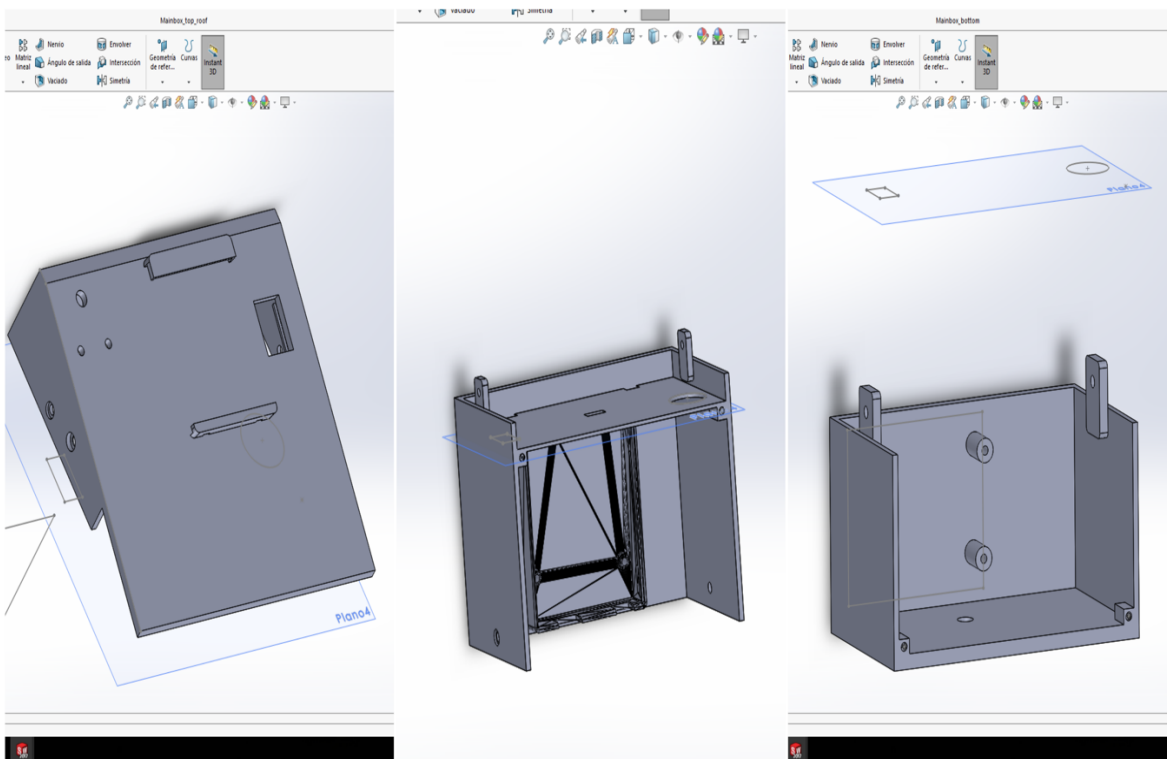


Figura 75. De izquierda a derecha: parte superior de la caja de sensores (orificios para sensores); la parte media (Arduino y sensores); la parte inferior (esta contiene la baquelita). Fuente: Autor.

La ventaja de imprimir en 3D es que la estructura queda más firme, al no tener que usar adhesivo o bastantes tornillos para unir las partes si fueran mecanizadas en CNC. La desventaja es que las piezas demoran más en estar terminadas.



Figura 76. Impresión de una pieza en 3D en pausa. *Fuente: Autor.*

Si la caja se imprime en partes, estas podrían acoplarse usando tornillos o adhesivos.

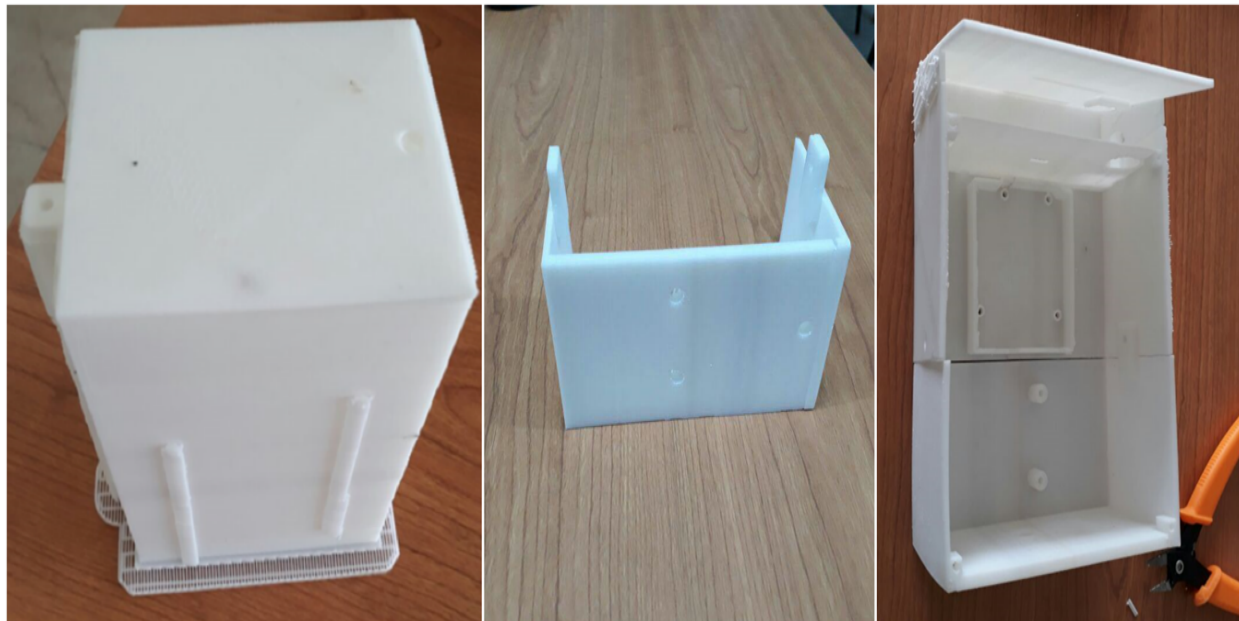


Figura 77. Piezas impresas en 3D para la estructura de la caja. *Fuente: Autor.*

Entonces, se diseña la tapa frontal de la caja, de tal forma que la pantalla de la LCD sea visible, que tenga espacio para el interruptor de encendido e indicador de encendido.

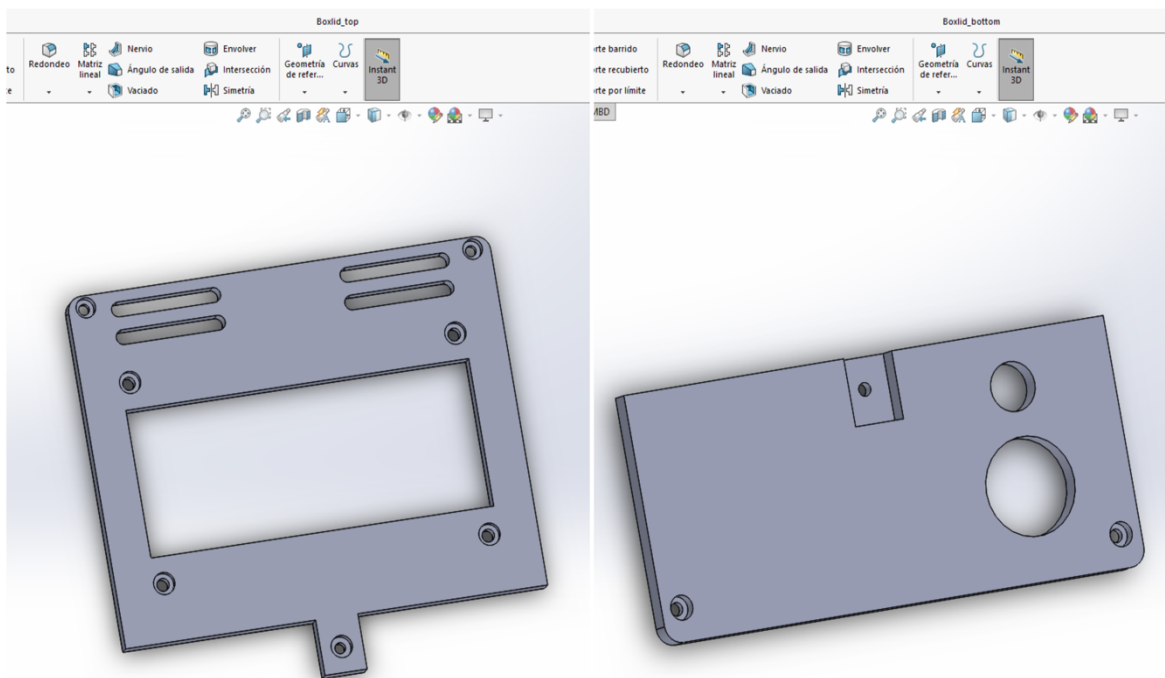


Figura 78. Diseño de la tapa frontal de la estructura de la caja. *Fuente: Autor.*

Se esboza la torre para el módulo GPS, para que de esta manera obtenga la señal del satélite más efectivamente.

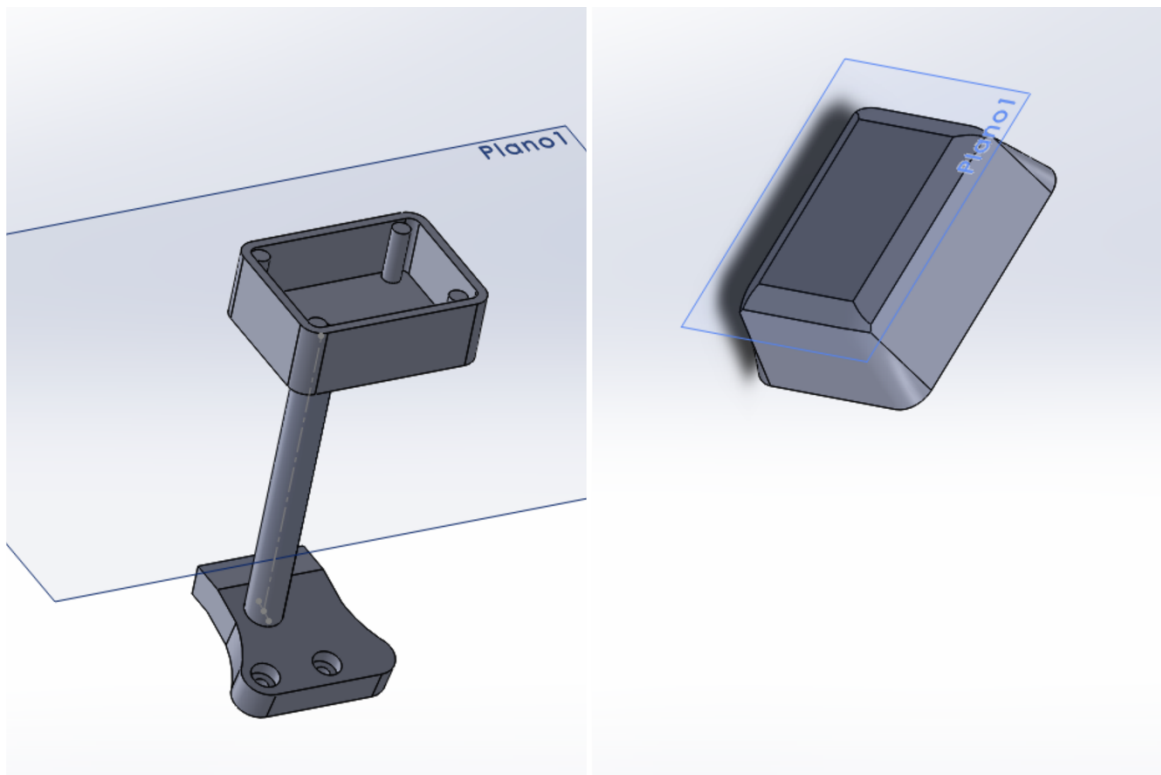


Figura 79. Diseño de las piezas para la torre de la caja para el módulo GPS. *Fuente: Autor.*

La baquelita contiene un circuito para alimentar a todos los sensores desde la Arduino (aunque algunos sensores se conectan directamente a la Arduino cuando se requiere), también espacios para conectar módulos como el de SD/micro SD, del sensor de gotas de lluvia o del sensor de humedad del suelo. Hace de puente para conectar la Arduino con todos los sensores para transmitir datos.

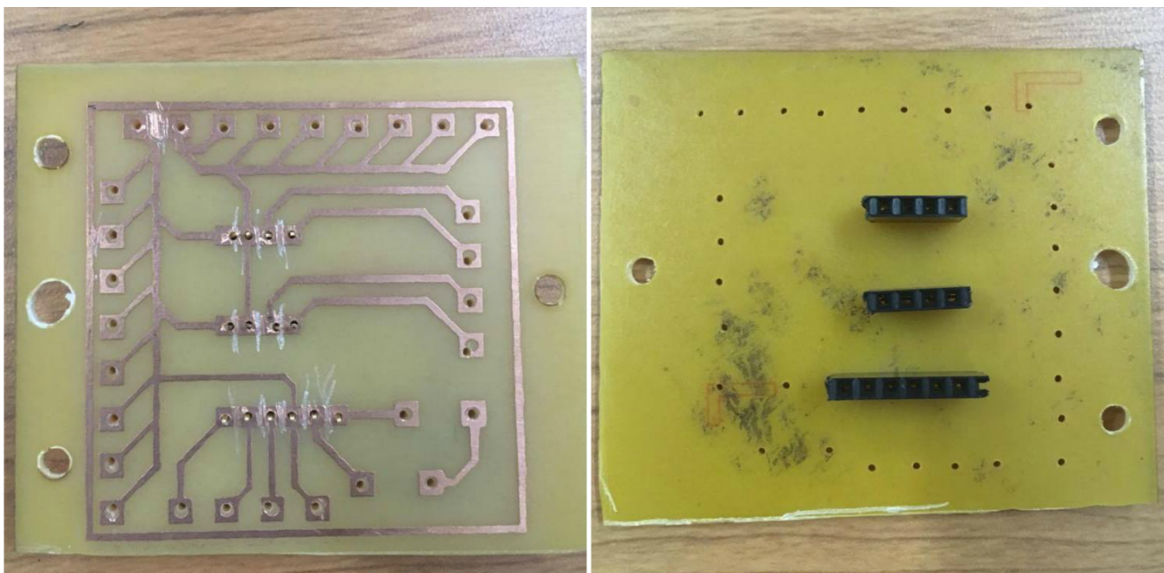


Figura 80. Baquelita de la estación meteorológica. *Fuente: Autor.*

Después de acoplar la baquelita en la caja se procede a instalar todos los sensores en sus respectivas ubicaciones. Se sueldan los cables en la baquelita que luego irán soldados a todos los componentes. La torre del GPS también se ensambla junto con el módulo.

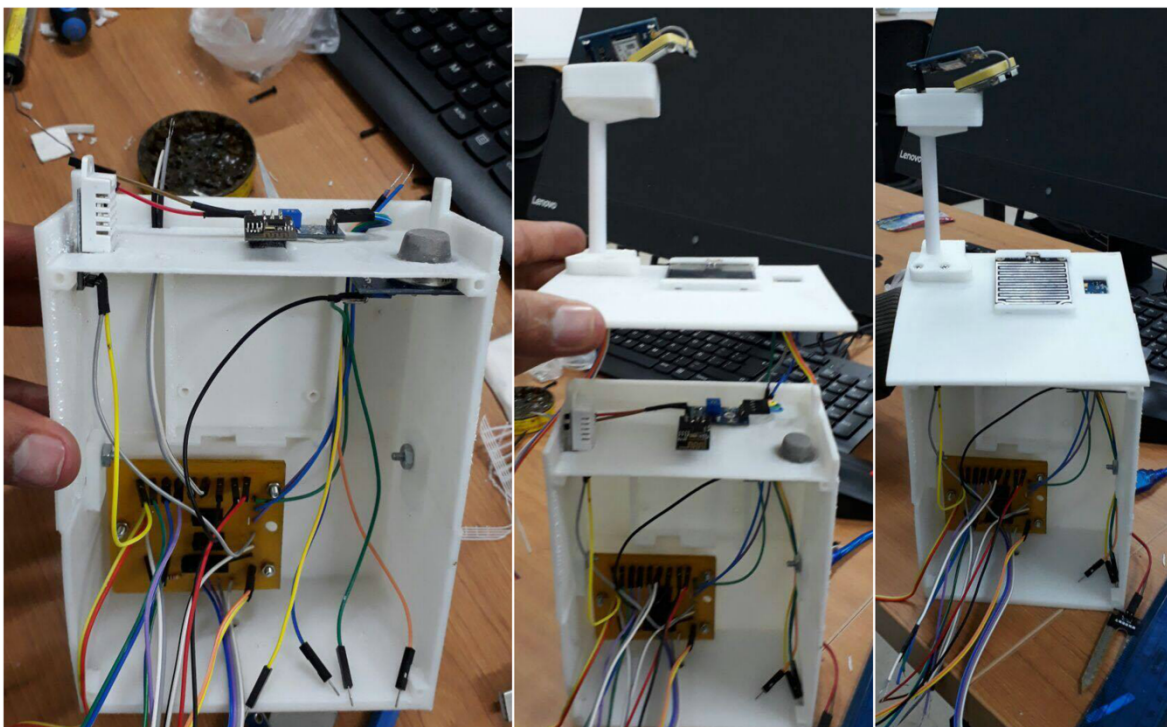


Figura 81. Montaje de la baquelita y los sensores. *Fuente: Autor.*

La alimentación de la Arduino puede darse desde la computadora, baterías LiPo, baterías AA, celdas solares, que provean suficiente corriente; según la especificaciones de la tabla #. Se pueden combinar fuentes de energía, como baterías recargables y celdas solares que las mantengan recargadas mientras reciban luz solar.

El soporte para la caja se recomienda de acero inoxidable. La celda solar se debe adaptar al voltaje de la batería que se utilice, con corriente aproximada a la salida de las baterías.



Figura 82. Panel solar sobre soporte de acero junto con la caja que contiene los sensores. *Fuente: Autor.*

Luego se ensambla la tapa frontal con LCD y el interruptor de encendido.



Figura 83. Estación meteorológica construida. *Fuente: Autor.*

Capítulo 8

Evaluando la estación meteorológica

Al encender la estación meteorológica se aprecia un mensaje con el nombre determinado para la estación meteorológica: ESTACION AGRICOLA ISER.



Figura 84. Inicio exitoso de la estación meteorológica. Significa que el módulo SD/micro SD está conectado y funciona, así como los puertos seriales y la LCD. *Fuente: Autor.*

Después de iniciar exitosamente, los sensores empiezan a tomar medidas y los datos se muestran en la pantalla LCD. Primero aparece la lectura de los sensores DHT22 (temperatura y humedad ambiente), humedad del suelo y detector de gotas de lluvia.

Al cambiar de pantalla, aparece la segunda lectura, la de los sensores de luminosidad BH1750, calidad del aire o gases, y la ubicación, fecha, hora obtenidos por el GPS.

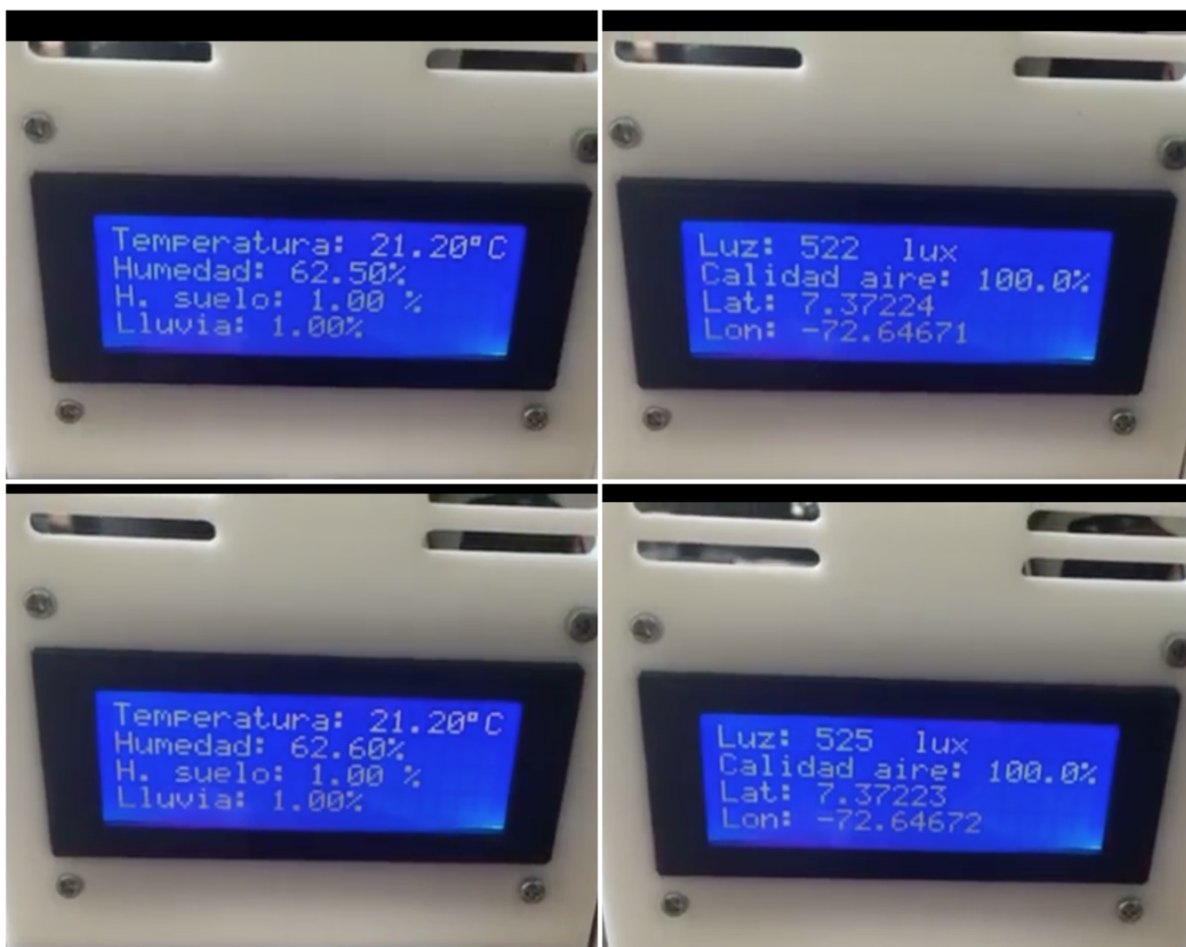


Figura 85. Muestra de dos mediciones de todos los sensores. Después del inicio de mide temperatura, humedades y gotas de lluvia (arriba-izquierda); 4 segundos después se mide luminosidad, gases y ubicación geográfica (arriba-derecha); después de otros 4 segundos repite las mediciones (abajo). *Fuente: Autor.*

Por razones climáticas, el GPS puede demorar en captar la señal satelital. Mientras tanto, los datos del GPS, como no existen aún, no se muestran.

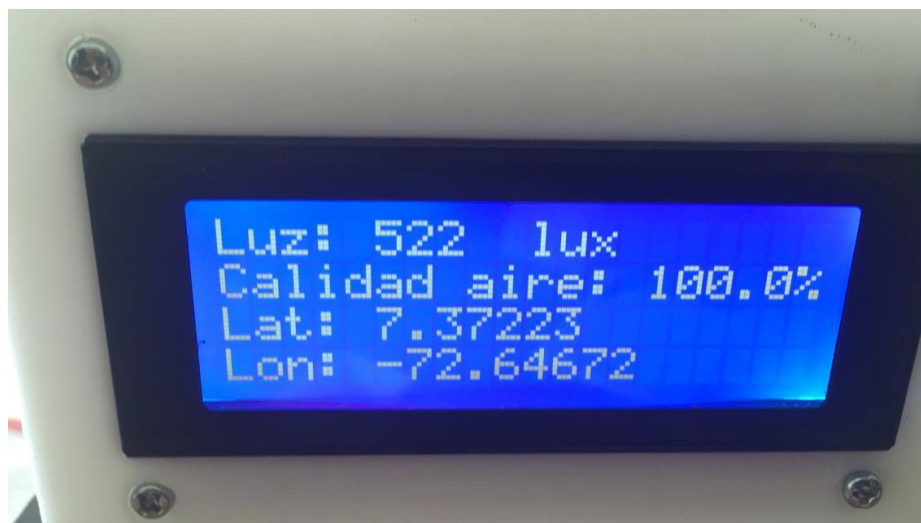


Figura 86. Muestra en pantalla de las medidas de luminosidad, gases y ubicación del GPS. *Fuente: Autor.*

La estación agrícola móvil puede tomar datos en cualquier parte, vélgase la redundancia, tanto en interiores como exteriores: en invernaderos de cualquier tipo y en terrenos de cultivo a campo abierto.

Conectando la SD/micro SD a una computadora se pueden encontrar los dos archivos de mediciones:

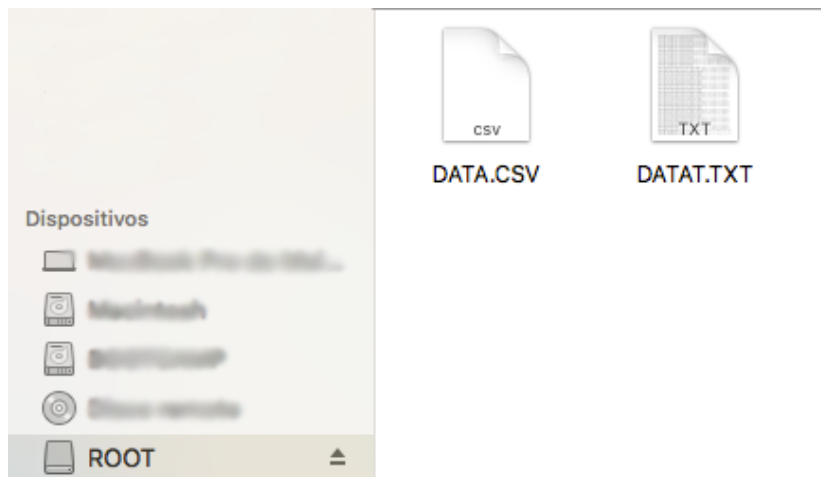


Figura 87. Los dos archivos guardados en la SD/micro SD donde se almacenan las mediciones. *Fuente: Autor.*



Figura 88. Mediciones almacenadas en el archivo de texto (txt). Fuente: Autor.

Para abrir el archivo CSV en Excel se debe seguir cierto procedimiento de importación. (Véase el apéndice C)

	A	B	C	D	E	F	G	H	I	J
1	Fecha	Hora	Latitud	Longitud	Temperatura	Humedad	Humedad del suelo	Lluvia	Luminosidad	Calidad del aire
2	24/11/17	16:57:13.0	7.37	-72.65	20.10	78.80	1.00	1.00	136	100.00
3	24/11/17	16:57:22.0	7.37	-72.65	20.10	78.80	1.00	1.00	136	100.00
4	24/11/17	16:57:22.0	7.37	-72.65	20.10	78.70	1.00	1.00	136	100.00
5	24/11/17	16:57:39.0	7.37	-72.65	20.10	78.70	1.00	1.00	136	100.00
6	24/11/17	16:57:39.0	7.37	-72.65	20.10	78.70	1.00	1.00	136	100.00
7	24/11/17	16:57:56.0	7.37	-72.65	20.10	78.70	1.00	1.00	135	100.00
8	24/11/17	17:1:51.0	7.37	-72.65	20.30	78.30	1.00	1.00	129	100.00
9	24/11/17	17:1:59.0	7.37	-72.65	20.30	78.20	1.00	1.00	129	100.00
10	24/11/17	17:2:8.0	7.37	-72.65	20.30	78.20	1.00	1.00	128	100.00
11	24/11/17	17:2:8.0	7.37	-72.65	20.30	78.20	1.00	1.00	128	100.00
12	24/11/17	17:2:25.0	7.37	-72.65	20.30	78.20	1.00	1.00	128	100.00
13	24/11/17	17:2:34.0	7.37	-72.65	20.30	78.20	1.00	1.00	128	100.00
14	24/11/17	17:2:42.0	7.37	-72.65	20.30	78.10	1.00	1.00	128	100.00
15	24/11/17	17:2:51.0	7.37	-72.65	20.30	78.10	1.00	1.00	128	100.00
16	24/11/17	17:2:51.0	7.37	-72.65	20.30	78.10	1.00	1.00	127	100.00
17	24/11/17	17:3:8.0	7.37	-72.65	20.30	78.10	1.00	1.00	127	100.00
18	24/11/17	17:3:17.0	7.37	-72.65	20.30	78.00	1.00	1.00	127	100.00
19	24/11/17	17:3:17.0	7.37	-72.65	20.30	78.00	1.00	1.00	127	100.00
20	24/11/17	17:3:34.0	7.37	-72.65	20.30	78.00	1.00	1.00	126	100.00
21	24/11/17	17:7:23.0	7.37	-72.65	20.40	77.40	1.00	1.00	122	100.00
22	24/11/17	17:7:40.0	7.37	-72.65	20.40	77.40	1.00	1.00	122	100.00
23	24/11/17	17:7:49.0	7.37	-72.65	20.40	77.30	1.00	1.00	122	100.00
24	24/11/17	17:7:49.0	7.37	-72.65	20.40	77.30	1.00	1.00	123	100.00
25	24/11/17	17:8:6.0	7.37	-72.65	20.40	77.30	1.00	1.00	123	100.00
26	24/11/17	17:8:6.0	7.37	-72.65	20.40	77.20	1.00	1.00	123	100.00
27	24/11/17	17:8:6.0	7.37	-72.65	20.40	77.20	1.00	1.00	122	100.00
28	24/11/17	17:8:32.0	7.37	-72.65	20.40	77.20	1.00	1.00	122	100.00
29	24/11/17	17:8:32.0	7.37	-72.65	20.40	77.20	1.00	1.00	122	100.00
30	24/11/17	17:8:40.0	7.37	-72.65	20.40	77.20	1.00	1.00	122	100.00

Figura 89. Datos del archivo CSV importados a Excel. Fuente: Autor.

Conclusiones

- El objetivo principal fue completado, el vehículo aéreo no tripulado fue construido, calibrado y vuela, la estación meteorológica móvil toma todas las mediciones y las almacena, funciona a cabalidad.
- Construir un dron se reduce a conseguir las partes y ensamblarlas. La fibra de vidrio es una buena, pero los metales u otros materiales resistentes también son opción. Los cables deben organizarse eficazmente para no formar espirales o cruces que provoquen interferencia magnética en el controlador de vuelo.
- En la calibración, las brújulas son de los componentes más sensibles al magnetismo, por lo que recibe interferencia ciertas veces. No se debe intentar aislar con metales, solo es buscar un lugar despejado y mantener apagados los dispositivos electrónicos alrededor en lo posible.
- Las baterías de 4000, 5000 y 6000 mAh son de gran utilidad por su gran capacidad de almacenamiento de energía eléctrica, y por consiguiente más tiempo de vuelo, pero las baterías de 2200 mAh también dan la potencia suficiente para el vuelo de un dron, durante hasta poco más de diez minutos.
- El control manual del dron es importante, aunque conducirlo en vuelo no es tan fácil como parece. Se debería leer más sobre las maniobras que puede hacer en vuelo para conocer mejor el manejo de las palancas del mando. Hay que tener noción suficiente del posicionamiento en el espacio.
- Usar la conexión inalámbrica entre la estación terrestre al dron reduce el tiempo de configuración, también se usa un componente menos en el dron, ya que no usa receptor RC para conectarse a un mando.

- Las misiones automáticas permiten un mejor uso del vehículo, es más preciso que el control manual por un principiante, así que no hay prisa de aprender a conducirlo sino que se pueden crear rutas para la supervisión de cultivos brevemente. Si se sigue desarrollando este proyecto y similares, la optimización del campo por medio de la electrónica le dará un gran empuje a la agricultura.
- La estructura de la estación meteorológica se puede construir de varias maneras. La impresión 3D se demora más de lo que se quisiera, pero los acabados quedan muy bien y las piezas son bastante rígidas, pero si todas las partes se mecanizaran en CNC sería más barato y más rápido se arma.
- Los sensores usados para la estación meteorológica están disponibles en muchas partes y junto con el intuitivo entorno de desarrollo de Arduino, el desarrollo de este dispositivo se puede llevar a cabo en cualquier región.
- Suelen encontrarse librerías para Arduino que están obsoletas, así que se recomienda buscar varias que sean similares para comparar su utilidad.
- El módulo GPS de la estación meteorológica algunas veces no envía datos porque el clima afecta mucho la recepción del satélite. En Arduino Uno cualquier librería de puerto serial funciona, sin embargo, en Arduino Mega no todas, incluso los puertos seriales incluidos pueden fallar, por lo que es recomendable usar la librería AltSoftSerial.
- Aparte del mal clima, no hay nada más que interrumpa las mediciones de la estación meteorológica. Durante su vida útil podrá mostrar datos y almacenarlos sin parar. Con la interpretación correcta de estas mediciones se logrará mejorar la producción de los campos de cultivo.

Referencias

- Adrián González, Gelberth Amarillo, Milton Amarillo, Francisco Sarmiento. (2015). *Drones aplicados a la agricultura de precisión* (Artículo). Universidad de Cundinamarca, Universidad Nacional Abierta y a Distancia, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. (Consultado en agosto de 2017).
- Alegsa, Leandro. (22 de junio de 2016). *Definición de Joystick (Palanca de mando)*. Recuperado en noviembre de 2017 de <http://www.alegsa.com.ar/Dic/joystick.php>
- Aosong Electronics Co. (2017). *DHT22 Datasheet*. Recuperado en noviembre de 2017 de <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- Aprendiendo Arduino. (6 de noviembre de 2016). ICSP. (Figura 74). Recuperado en noviembre de 2017 de <https://aprendiendoarduino.wordpress.com/tag/icsp/>
- Arduino. (2005). *Arduino Mega 2560 rev 3*. (Figura 8). Recuperado en noviembre de 2017 de <https://store.arduino.cc/usa/arduino-mega-2560-rev3>
- Cheyla Vanesa Acevedo Londoño, Luis Carlos Durán Rodríguez. (2017). Desarrollo de un prototipo para monitoreo de variables y georreferenciación de espacios controlados de producción agrícola en la granja la rinconada del ISER de Pamplona (Trabajo de grado). Instituto Superior de Educación Rural ISER, Pamplona, Norte de Santander, Colombia. (Figura 65-69, 71, 72). (Consultado en noviembre de 2017).
- Drone Center. (2015). *Construye tu dron*. Recuperado en noviembre de 2017 de <http://dronecenter.blogspot.com.co/p/construye-tu-drone.html>
- DroneVal. (2017). Batería LiPo de seguridad 3300mAh. (Figura 2). Recuperado en noviembre de 2017 de <https://droneval.com/baterias-lipo/1060-bateria-lipo-de-seguridad-3300mah-2s-74v-25c-50c>

- EcuRed. (2017). *Vehículo aéreo no tripulado*. Recuperado en noviembre de 2017 de http://www.ecured.cu/Veh%C3%ADculo_a%C3%A9reo_no_tripulado
- Electronilab. (2017). *Sensor detector de lluvia o gotas de agua*. Recuperado en noviembre de 2017 de <https://electronilab.co/tienda/sensor-detector-de-lluvia-o-gotas-de-agua-humedad/>
- Electronilab. (2017). *Sensor de humedad de suelo*. Recuperado en noviembre de 2017 de <https://electronilab.co/tienda/sensor-de-humedad-de-suelo-higrometro/>
- Electronilab. (2017). *Sensor de temperatura y humedad DHT22*. Recuperado en noviembre de 2017 de <https://electronilab.co/tienda/sensor-de-temperatura-y-humedad-dht22/>
- Emliano García, Fernando Flego. (2014). *Agricultura de precisión* (Artículo). Universidad de Palermo, Buenos Aires, Argentina. Recuperado en agosto de 2017 de <http://www.palermo.edu/ingenieria/downloads/pdfwebc&T8/8CyT12.pdf>
- Emlid. (2017). Hardware setup. (Figura 20, 22-25). Recuperado en septiembre de 2017 de <https://docs.emlid.com/navio2/ardupilot/hardware-setup/>
- Emlid. (2017). Installation and running. (Figura 36). Recuperado en septiembre de 2017 de <https://docs.emlid.com/navio2/common/ardupilot/installation-and-running/>
- Emlid. (2017). *Navio2 docs*. (Figura 4). Recuperado en septiembre de 2017 de <https://docs.emlid.com/navio2/>
- Emlid. (2017). ROS. (Figura 6,7). Recuperado en septiembre de 2017 de <https://docs.emlid.com/navio2/common/dev/ros/>
- Equipo de desarrollo de Ardupilot. (2016). *Choosing a ground station*. Recuperado en noviembre de 2017 de <http://ardupilot.org/copter/docs/common-choosing-a-ground-station.html#common-choosing-a-ground-station>

Equipo de desarrollo de Ardupilot. (2016). Compass Calibration. (Figura 40). Recuperado en noviembre de 2017 de <http://ardupilot.org/copter/docs/common-compass-calibration-in-mission-planner.html>

Equipo de desarrollo de Ardupilot. (2016). Frame class and type configuration. (Figura 46). Recuperado en noviembre de 2017 de <http://ardupilot.org/copter/docs/frame-type-configuration.html>

Equipo de desarrollo de Ardupilot. (2016). LEDs. Recuperado en noviembre de 2017 de <http://ardupilot.org/copter/docs/common-leds-pixhawk.html?highlight=led>

Equipo de desarrollo de Ardupilot. (2016). *Mission Planning*. (Figura 64). Recuperado en noviembre de 2017 de <http://ardupilot.org/copter/docs/common-mission-planning.html>

Equipo de desarrollo de Ardupilot. (2016). *Software*. Recuperado en noviembre de 2017 de <http://ardupilot.org/about>

FPV Max. (2016). *Hélices para drones: tipos y tamaños*. Recuperado en noviembre de 2017 de <http://fpvmax.com/2017/02/10/helices-drones-tipos-tamanos/>

Gnome Help. (2014). *¿Qué es una dirección IP?* Recuperado en noviembre de 2017 de <https://help.gnome.org/users/gnome-help/stable/net-what-is-ip-address.html.es>

Gobierno de los Estados Unidos. (2017). *El sistema de posicionamiento global*. Recuperado en noviembre de 2017 de <https://www.gps.gov/systems/gps/spanish.php>

Huidobro Moya, José Manuel. (2ª ed.). (2015). *Telecomunicaciones: tecnologías, redes y servicios*. Bogotá, Colombia: Ra-Ma Editorial, Ediciones de la U. (Consultado en noviembre de 2017).

InfoRepuesto. (11 de junio de 2017). *Como funcionan y vuelan los drones*. Recuperado en noviembre de 2017 de <http://inforepuesto.com/como-funcionan-y-vuelan-los-drones/>

- JaviRock. (24 de noviembre de 2016). *¿Qué es un dron?* Recuperado en noviembre de 2017 de <http://mascandobits.es/invitados/que-es-un-dron-conceptos-basicos/>
- Jazmín Carolina, Ortega. (2014) *Desarrollo de un prototipo de adquisición de variables ambientales en cultivos hidropónicos de lechuga, mediante una red de sensores, utilizando un sistema embebido* (Tesis de pregrado). Universidad de San Buenaventura. Facultad de ingeniería. Departamento de ingeniería de electrónica e ingeniería de telecomunicaciones, Bogotá, Colombia. (Consultado en octubre de 2017).
- Kaiser Baas. (2017). Delta drone. (Figura 1). Recuperado en noviembre de 2017 de <https://www.kaiserbaas.com/products/delta-drone>
- Maker Electronics. (2017). MQ2 sensor de gas humo propano. (Figura 70). Recuperado en noviembre de 2017 de <https://www.makerelectronico.com/producto/mq2-sensor-de-gas-humo-arduino-propano/>
- Marthinus Reinecke, Tania Prinsloo. (2017). *The influence of drone monitoring on crop health and harvest size* (Artículo). Universidad de Pretoria, Pretoria, Sudáfrica. Recuperado en noviembre de 2017 de <http://ieeexplore.ieee.org/document/8016168/>
- Masadelante. (2017). *¿Qué es el UDP? – Definición de UDP*. Recuperado en noviembre de 2017 de <http://www.masadelante.com/faqs/udp>
- MaxElectrónica. (2017). Sensor de temperatura y humedad. (Figura 9). Recuperado en noviembre de 2017 de <http://www.maxelectronica.cl/temperatura-y-humedad/165-sensor-de-temperatura-y-humedad-dht22.html>
- Mercado Libre Uruguay. (2017). Sensor de humedad de suelo. (Figura 10). Recuperado en noviembre de 2017 de https://articulo.mercadolibre.com.uy/MLU-443940114-sensor-de-humedad-de-suelo-_JM

Molina Martínez, José Miguel. Cánovas Rodríguez, Francisco Javier. Ruz Vila, Francisco Asís.

(1ª ed.). (2014). *Motores y máquinas eléctricas: fundamentos de electrotecnia para ingenieros*. Ciudad de México, México: Alfa Omega Grupo Editor, Marcombo.

(Consultado en noviembre de 2017).

Moviltronic SAS. (2017). *Sensor luminosidad BH1750*. (Figura 12) Recuperado en noviembre de 2017 de <https://moviltronics.com.co/sensores/138-sensor-luminosidad-bh1750.html>

MundoCdWeb. (2017). Joystick Xbox 360 con cable. (Figura 54). Recuperado en noviembre de 2017 de <https://www.mundocdweb.com/joystick/83-joystick-wired-xbox-360-con-cable-usb-pcxbox-mundocdweb.html>

Naylamp Mechatronics. (2017). Sensor calidad de aire MQ135. (Figura 13). Recuperado en noviembre de 2017 de <http://www.naylampmechatronics.com/sensores-gas/73-sensor-calidad-aire-mq135.html>

Omega. (2017). *Acelerómetro*. Recuperado en noviembre de 2017 de <https://es.omega.com/prodinfo/acelerometro.html>

Omniblug. (24 de julio de 2014). *Sensor de temperatura y humedad DHT11-DHT22*. Recuperado en noviembre de 2017 de <http://www.omniblug.com/sensor-temperatura-humedad-DHT11-DHT22.html>

Programas de estudios de EEB. (13 de abril de 2013). *La tecnología punta de la agricultura*. Recuperado en noviembre de 2017 de <http://www.abc.es/natural-biodiversidad/20130412/abci-agricultura-tecnologia-punta-201304121101.html>

Prohobbys radiocontrol. (24 de enero de 2016). *¿Cómo funciona la emisora de un dron?* Recuperado en noviembre de 2017 de <http://prohobbysradiocontrol.net/wordpress/como-funciona-la-emisora-de-un-dron/>

- Raspberry Pi Foundation. (2015). Raspberry Pi 2 Modelo B. (Figura 5). Recuperado en noviembre de 2017 de <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- Real Academia Española. (2017). *Diccionario de la lengua española: brújula*. España. Recuperado en noviembre de 2017 de <http://dle.rae.es/?id=6B70B7K>
- Real Academia Española. (2017). *Diccionario de la lengua española: giroscopio*. España. Recuperado en noviembre de 2017 de <http://dle.rae.es/?id=JDfCuKx>
- Real Academia Española. (2017). *Diccionario de la lengua española: hélice*. España. Recuperado en noviembre de 2017 de <http://dle.rae.es/?id=K6RQJuP>
- Real Academia Española. (2017). *Diccionario de la lengua española: magnetómetro*. España. Recuperado en noviembre de 2017 de <http://dle.rae.es/?id=NtmaBLd>
- Red Hat, Inc. (2005). *Protocolo SSH*. Recuperado en noviembre de 2017 de <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>
- Superintendencia de Industria y Comercio. (Septiembre de 2015). Drones: *vehículos aéreos no tripulados y sus sistemas de comunicación*. Recuperado en noviembre de 2017 de http://www.sic.gov.co/recursos_user/documentos/publicaciones/Boletines/Drones.pdf
- TarjetasSD. (2017). *¿Qué es una tarjeta SD?* Recuperado en noviembre de 2017 de <http://tarjetasd.com/>
- Tarot. (25 de enero de 2013). *Tarot IRON MAN 650 full-folding four-axis vehicle rack TL65B01*. (Figura 14). Recuperado en noviembre de 2017 de http://www.tarot-rc.com/index.php?main_page=product_info&cPath=65_70&products_id=197
- Tdrobótica. (2017). Sensor de gotas de lluvia. (Figura 11). Recuperado en noviembre de 2017 de <http://tdrobotica.co/sensor-de-gotas-de-lluvia/778.html>

- Tiendatec. (2017). Módulo adaptador microsd para Arduino. (Figura 73) Recuperado en noviembre de 2017 de <https://www.tiendatec.es/arduino/modulos/631-modulo-adaptador-microsd-para-arduino-8406311180004.html>
- T-Motor. (2017). *Air Gear 350*. (Figura 3). Recuperado en noviembre de 2017 de <http://store-en.tmotor.com/goods.php?id=452>
- Tojeiro Calaza, Germán. (1ª ed.). (2016). *Raspberry Pi 2 para electrónicos*. Ciudad de México, México: Alfa Omega Grupo Editor, Marcombo. (Consultado en septiembre de 2017).
- Users. Benchimol Daniel. (1ª ed.). (2011). *Microcontroladores*. Buenos Aires, Argentina: Fox Andina, DALAGA S.A. (Consultado en noviembre de 2017).

Apéndice A: Código completo de la estación meteorológica

```
#include <AltSoftSerial.h>//Librería de puerto serial

#include <SD.h>

#include <TinyGPS.h>//Traductor de datos del GPS

#include <BH1750.h>//Librería del sensor de luminosidad

#include <Adafruit_Sensor.h>//Librería para el DHT22

#include <Wire.h> //Librería I2C

#include <DHT.h>//Librería para el DHT22

#include <DHT_U.h>//Librería para el DHT22

#include <LiquidCrystal_I2C.h>//Librería I2C para pantallas LCD

LiquidCrystal_I2C lcd(0x3F, 20, 4); //0x3F es la dirección I2C asignada a la LCD

#define DHTPIN 2 //Pin digital del Arduino para el DHT22

#define DHTTYPE DHT22 //Definir el sensor como un DHT22

DHT dht(DHTPIN, DHTTYPE);

float temperatura;

float humedad;

TinyGPS gps;//Declarar el objeto gps

AltSoftSerial serialgps; //Se declara el puerto, que por defecto son 9, 8 en Arduino Uno
y 46,48 en Arduino Mega. (Tx,Rx)

//Declarar la variables para la obtención de datos

float latitude, longitude;

int year;
```

```
byte month, day, hour, minute, second, hundredths;

unsigned long chars;

unsigned short sentences, failed_checksum;

char c;

int BH1750address = 0x23; //Dirección I2C del sensor de luminosidad

byte BH1750buffer[2];

float humedads;

float lluvia;

float aire;

const int PinCs=4; //Pin de conexión al módulo micro SD

File MyFile;

void setup() {

  Serial.begin(9600);

  serialgps.begin(9600); //Iniciar el puerto serie del gps

  pinMode(0,INPUT); // entradas análogas

  pinMode(1,INPUT);

  pinMode(2,INPUT);

  pinMode(5,INPUT);

  pinMode(4,OUTPUT);

  Wire.begin();

  lcd.begin();

  lcd.backlight();

  lcd.setCursor(12,0); lcd.print("ESTACION");
```

```
lcd.setCursor(7,1);lcd.print("AGRICOLA ISER");

lcd.setCursor(2,3);lcd.print("Iniciando...");

delay(3000);

if (!SD.begin(PinCs))

{

    lcd.setCursor(0,3);lcd.print("Error al iniciar SD");

}

else

{

    lcd.setCursor(2,3);lcd.print("Inicio exitoso");

};

//Hoja de excel

if(!SD.exists("datalog.csv"))

{

    myFile = SD.open("datalog.csv", FILE_WRITE);

    if (myFile)

    {

        myFile.println("Fecha,Hora,Latitud,Longitud,Temperatura,Humedad,Humedad del
suelo,Lluvia,Luminosidad,Calidad del aire");

        myFile.close();

    }

else

{
```



```
lcd.setCursor(0,3);lcd.print("Error creando csv");

};

};

}

void loop() {

  delay(4000); //Tiempo de espera entre una lectura y otra. Para el DHT22 son 2 segundos

  lcd.clear();

  //Lectura del sensor de temperatura y humedad DHT22

  temperatura = dht.readTemperature(); //Lee la temperatura

  humedad = dht.readHumidity(); //Lee la humedad

  lcd.setCursor (0,0) ;lcd.print("Temperatura:");

  lcd.setCursor(13,0) ;lcd.print(temperatura);

  lcd.setCursor(18,0) ;lcd.print((char)223);

  lcd.setCursor(19,0) ;lcd.print("C");

  lcd.setCursor (0,1) ;lcd.print("Humedad:");

  lcd.setCursor(9,1) ;lcd.print(humedad);

  lcd.setCursor(14,1) ;lcd.print("%");

  //Lectura del sensor de humedad del suelo

  humedads=analogRead(0);

  humedads=map(humedads,0,1023,100,0);

  lcd.setCursor (0,2) ;lcd.print("H. suelo:");

  lcd.setCursor(10,2) ;lcd.print(humedads);

  lcd.setCursor(15,2) ;lcd.print("%");
```

```
//Lectura del sensor de lluvia;

lluvia=analogRead(1);

lluvia=map(lluvia,0,1023,100,0);

lcd.setCursor (0,3) ;lcd.print("Lluvia:");

lcd.setCursor(8,3) ;lcd.print(lluvia);

delay(4000);

lcd.clear();

//Lectura del sensor de luminosidad

uint16_t BH1750val = 0;

BH1750_Init(BH1750address);

if(2 == BH1750_Read(BH1750address)) {

    BH1750val=((BH1750buffer[0]<<8)|BH1750buffer[1])/1.2;

    lcd.setCursor(0,0); lcd.print("Luz:");

    lcd.setCursor(5,0); lcd.print(BH1750val, DEC);

    if((BH1750val)<100)

    {

        lcd.setCursor(9,0);lcd.print("lux");

    }

    else

    {

        if((BH1750val)<10000)

        {
```

```
    lcd.setCursor(10,0); lcd.print("lux");
}
else
{
    lcd.setCursor(12,0); lcd.print("lux");
};
};
};
//Lectura del sensor de calidad del aire
aire=analogRead(2);
aire=map(aire,0,1023,100,0);
lcd.setCursor(0,1);lcd.print("Calidad aire:");
lcd.setCursor(14,1);lcd.print(aire);
lcd.setCursor(19,1);lcd.print("%");
//Lectura del gps
while(serialgps.available())
{
    c = serialgps.read();
    if(gps.encode(c))
    {
        gps.f_get_position(&latitude, &longitude);//Se obtiene la latitud y la longitud
        lcd.setCursor(0,2); lcd.print("Lat:");
        lcd.setCursor(5,2); lcd.print(latitude,5);
```

```

    lcd.setCursor(0,3); lcd.print("Lon:");

    lcd.setCursor(5,3); lcd.print(longitude,5);

    gps.crack_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths); //Se
obtiene la hora

    Serial.print("Fecha: "); Serial.print(day, DEC); Serial.print("/");

    Serial.print(month, DEC); Serial.print("/"); Serial.print(year);

    Serial.print(" Hora: "); Serial.print(hour, DEC); Serial.print(":");

    Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);

    Serial.print("."); Serial.println(hundredths, DEC);

};

};

//Carga de datos a la micro SD

myFile = SD.open("datalog.txt", FILE_WRITE);//abrir el archivo

if (myFile)
{
    myFile.print("Fecha: ");

    myFile.print(day, DEC); myFile.print("/");

    myFile.print(month, DEC); myFile.print("/"); myFile.print(year);

    myFile.print(" Hora: "); myFile.print(hour, DEC); myFile.print(":");

    myFile.print(minute, DEC); myFile.print(":"); myFile.print(second, DEC);

    myFile.print("."); myFile.print(hundredths, DEC);

    myFile.print(", latitud: ");

```

```
myFile.print(latitude);

myFile.print(" , longitud: ");

myFile.print(longitude);

myFile.print(" , temperatura: ");

myFile.print(temperatura);

myFile.print(" , humedad: ");

myFile.print(humedad);

myFile.print(" , humedad del suelo: ");

myFile.print(humedads);

myFile.print(" , lluvia: ");

myFile.print(lluvia);

myFile.print(" , luminosidad: ");

myFile.print(BH1750val,DEC);

myFile.print(" , calidad del aire: ");

myFile.println(aire);

myFile.close(); //cerrar el archivo

}

else

{

  lcd.clear();

  lcd.setCursor(3,2);lcd.print("Error al abrir");

  lcd.setCursor(3,3);lcd.print("el archivo txt");
```

```
};  
  
delay(100);  
  
//Datos en hoja de excel  
myFile = SD.open("datalog.csv", FILE_WRITE);//abrir el archivo  
if (myFile)  
{  
  myFile.print(day, DEC); myFile.print("/"); myFile.print(month, DEC);  
myFile.print("/"); myFile.print(year);  
  myFile.print(",");  
  myFile.print(hour, DEC); myFile.print(":"); myFile.print(minute, DEC);  
myFile.print(":"); myFile.print(second, DEC); myFile.print("."); myFile.print(hundredths,  
DEC);  
  myFile.print(",");  
  myFile.print(latitude);  
  myFile.print(",");  
  myFile.print(longitude);  
  myFile.print(",");  
  myFile.print(temperatura);  
  myFile.print(",");  
  myFile.print(humedad);  
  myFile.print(",");  
  myFile.print(humedads);  
  myFile.print(",");  
  myFile.print(lluvia);
```

```
myFile.print(BH1750val,DEC);

myFile.print(",");

myFile.println(aire);

myFile.close(); //cerrar el archivo
}
else
{
  lcd.clear();
  lcd.setCursor(3,2);lcd.print("Error al abrir");
  lcd.setCursor(3,3);lcd.print("el archivo csv");
};
} //Fin
int BH1750_Read(int address)
{
  int i =0;
  Wire.beginTransmission(address);
  Wire.requestFrom(address, 2);
  while(Wire.available()) {
    BH1750buffer[i] = Wire.read();
    i++;
  }
  Wire.endTransmission();
```

```
return i;
}
void BH1750_Init(int address)
{
  Wire.beginTransmission(address);
  Wire.write(0x10);
  Wire.endTransmission();
}
```


Apéndice B: Medidas de los diseños

Todas las medidas acotadas se miden en milímetros.

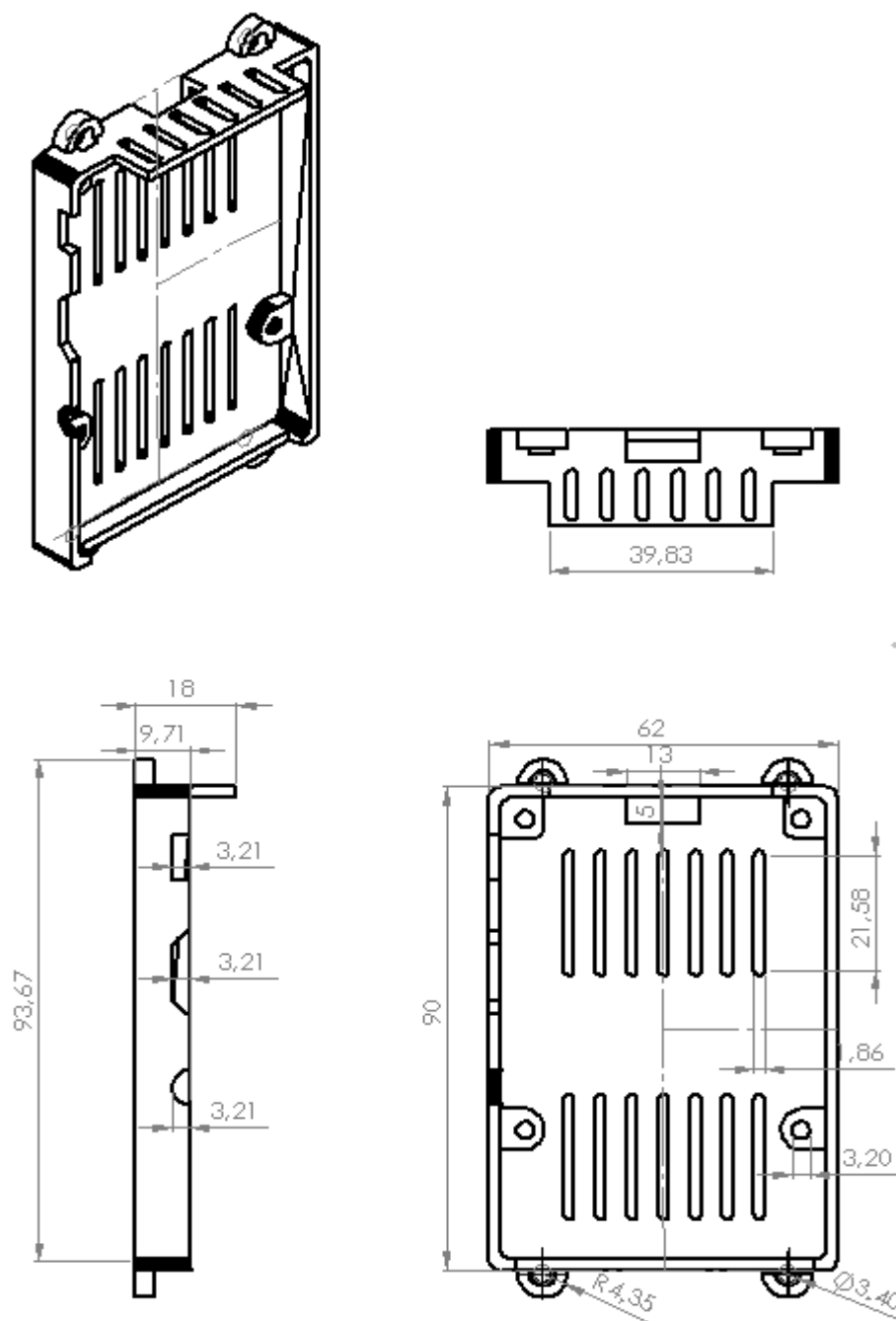


Figura 90. Parte de inferior de la carcasa para Raspberry Pi 2 y Navio2. Fuente: Autor.

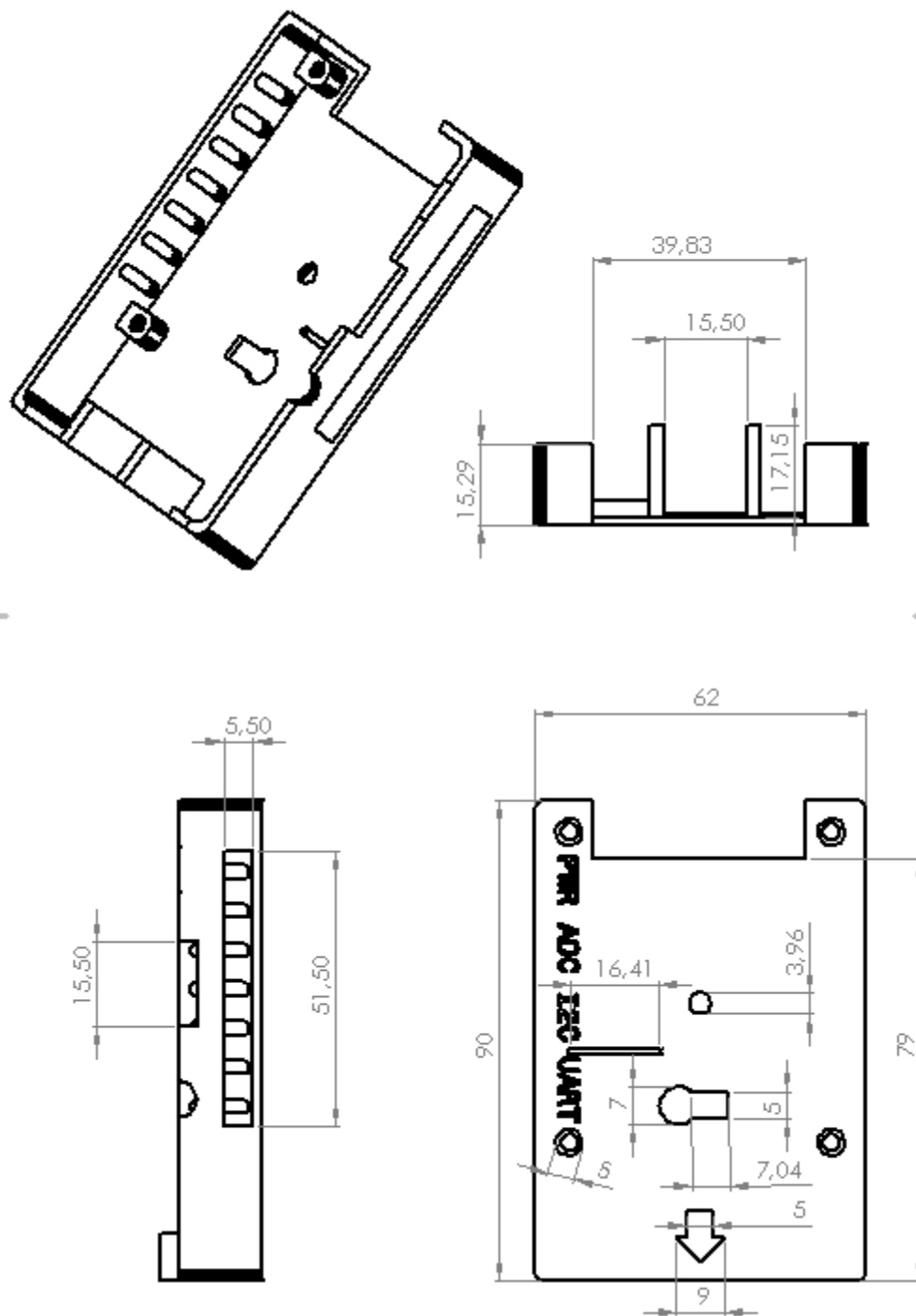


Figura 91. Parte superior de la carcasa para Raspberry Pi 2 y Navio2. Fuente: Autor.

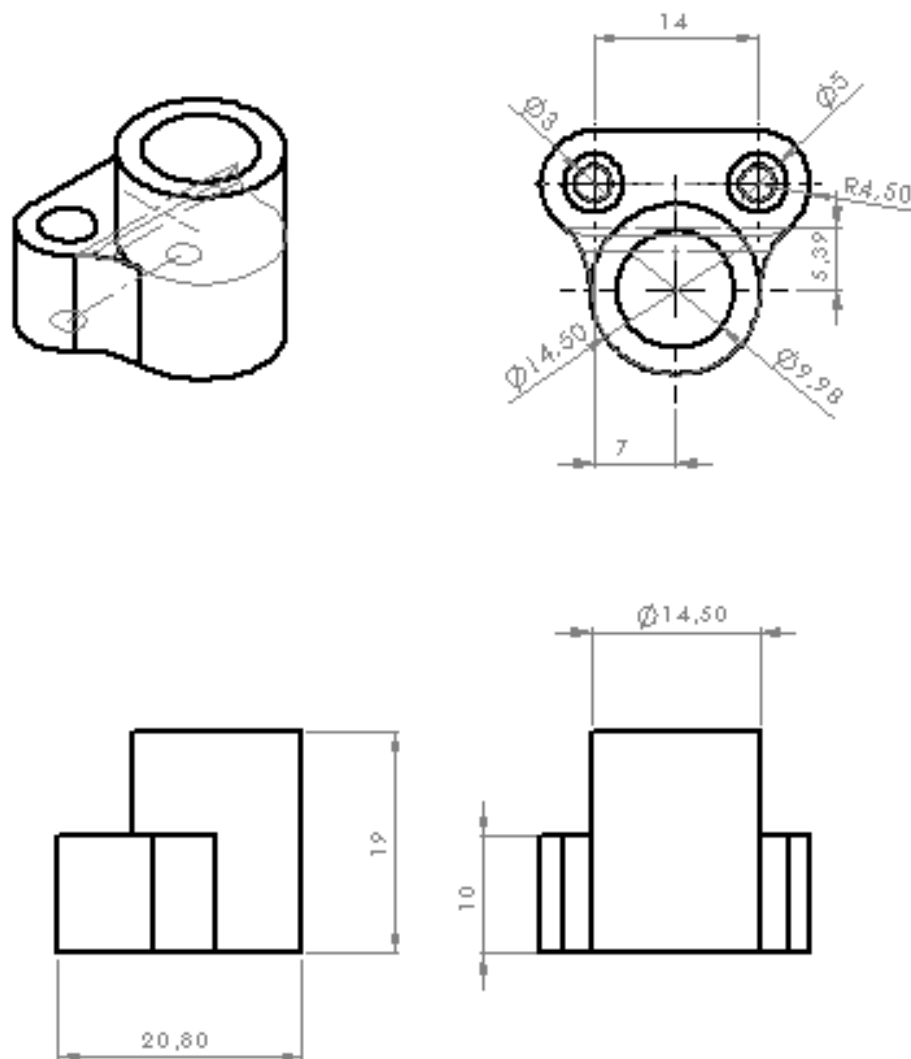


Figura 92. Base de la torrecilla del GPS en el dron. Fuente: Autor.

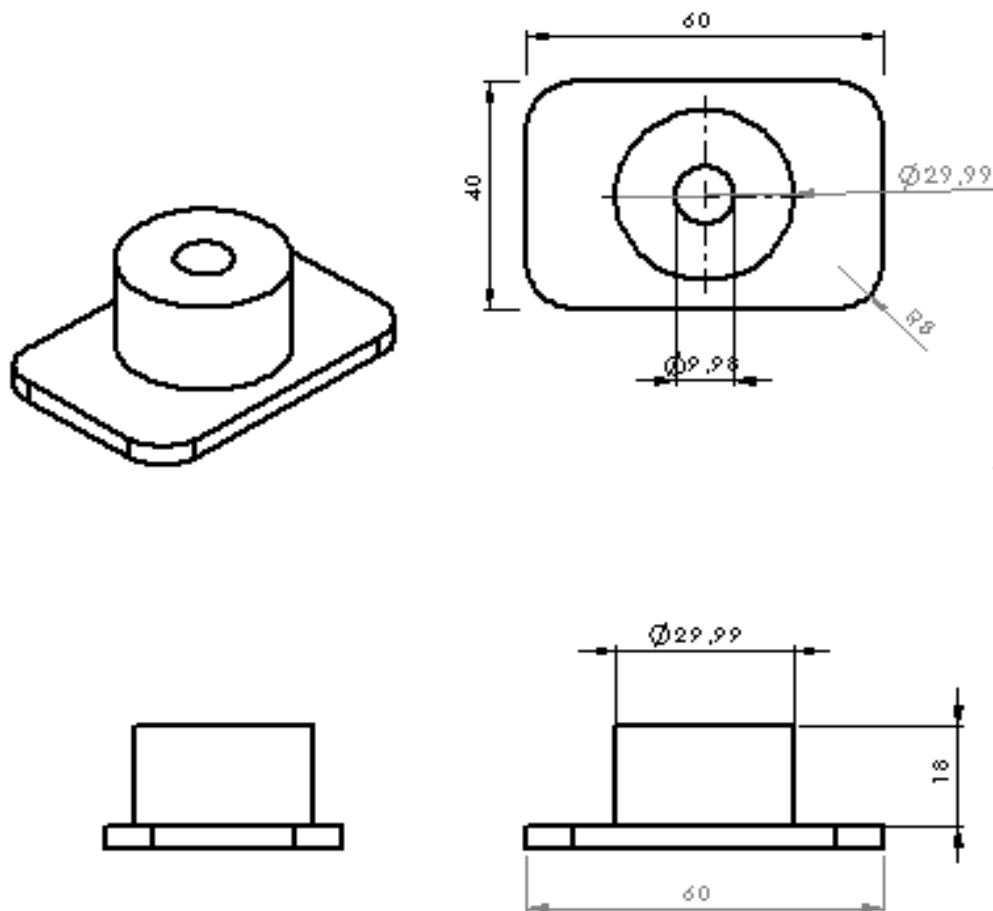


Figura 93. Parte superior de la torrecilla GPS del dron. Fuente: Autor.

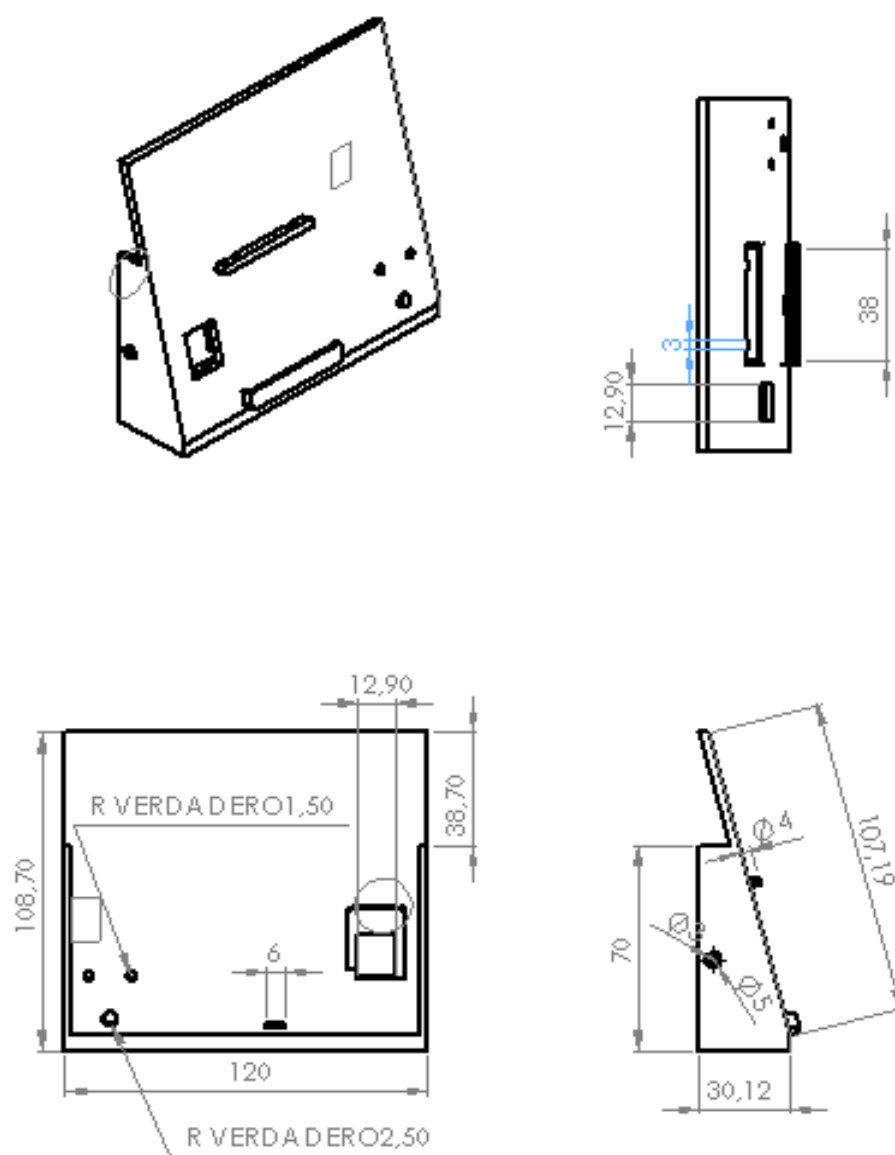


Figura 94. Parte superior de la caja de la estación meteorológica. Fuente: Autor.

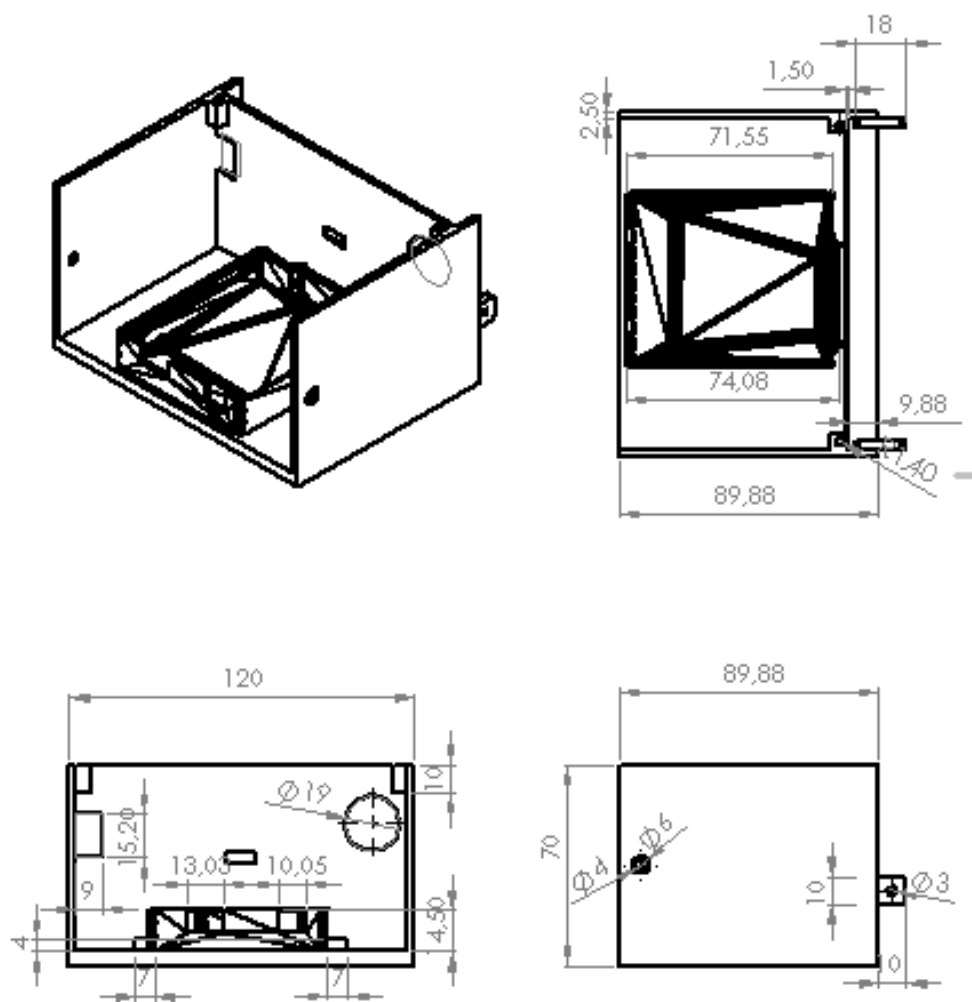


Figura 95. Parte intermedia de la caja de la estación meteorológica. Fuente: Autor.

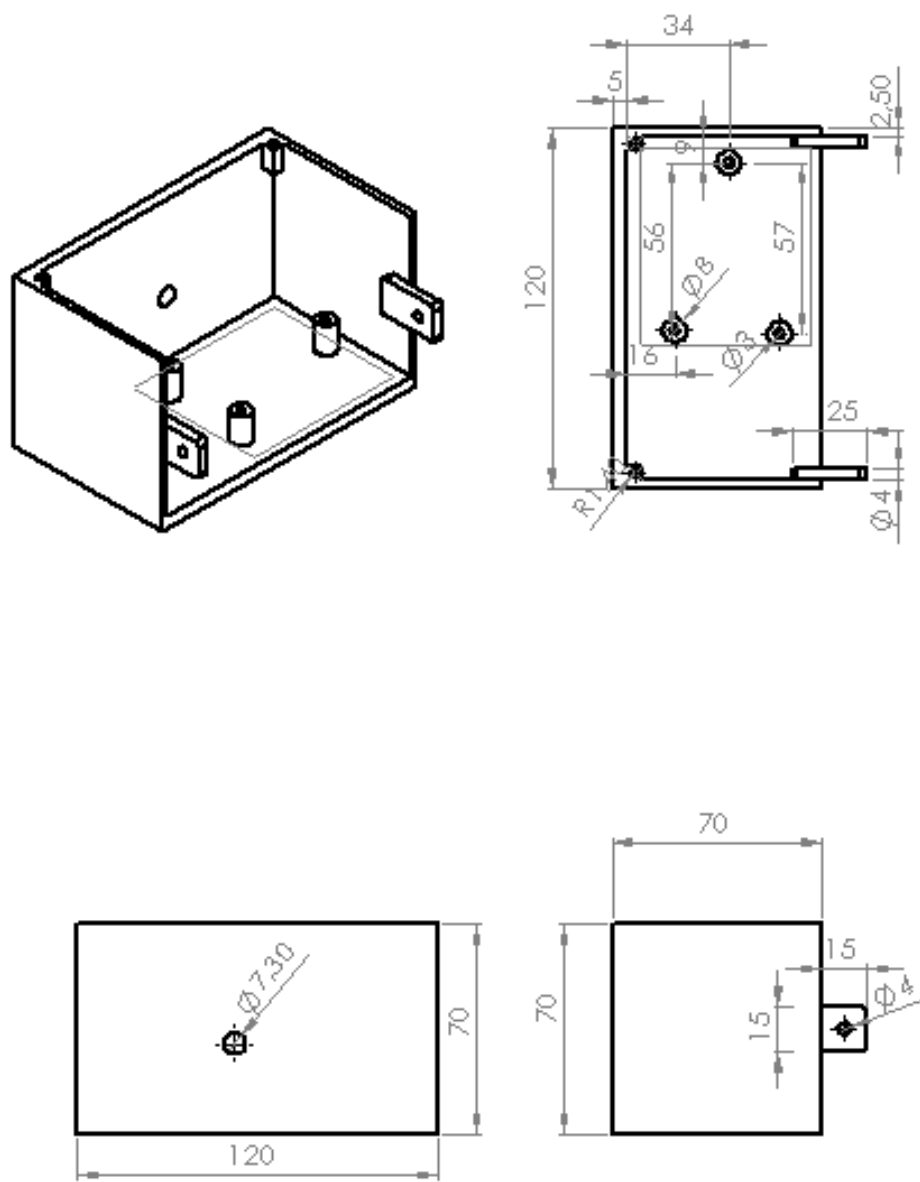


Figura 96. Parte inferior de la caja de la estación meteorológica. Fuente: Autor.

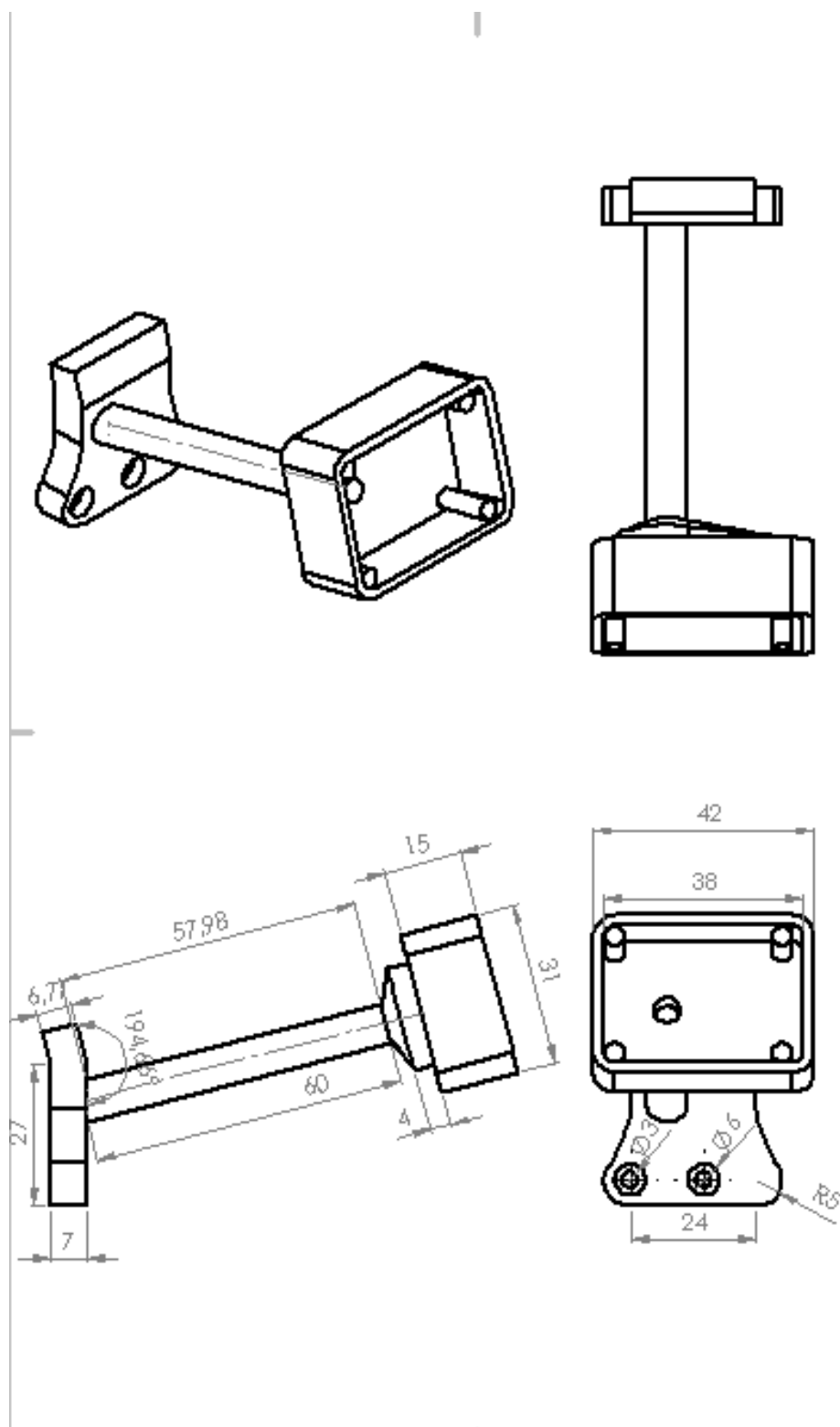


Figura 97. Torre del módulo GPS de la estación meteorológica. Fuente: Autor.

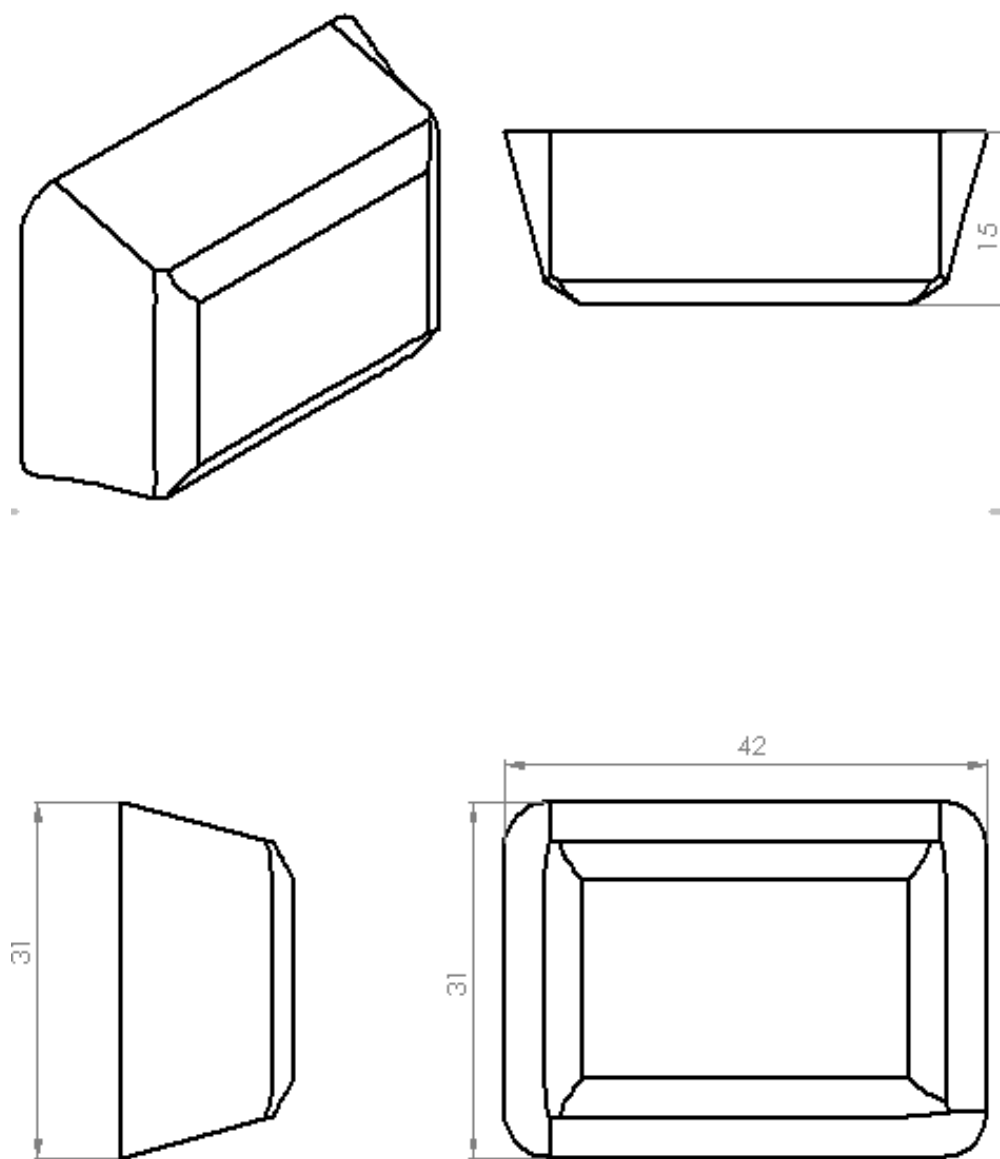


Figura 98. Tapa del módulo GPS. *Fuente: Autor.*

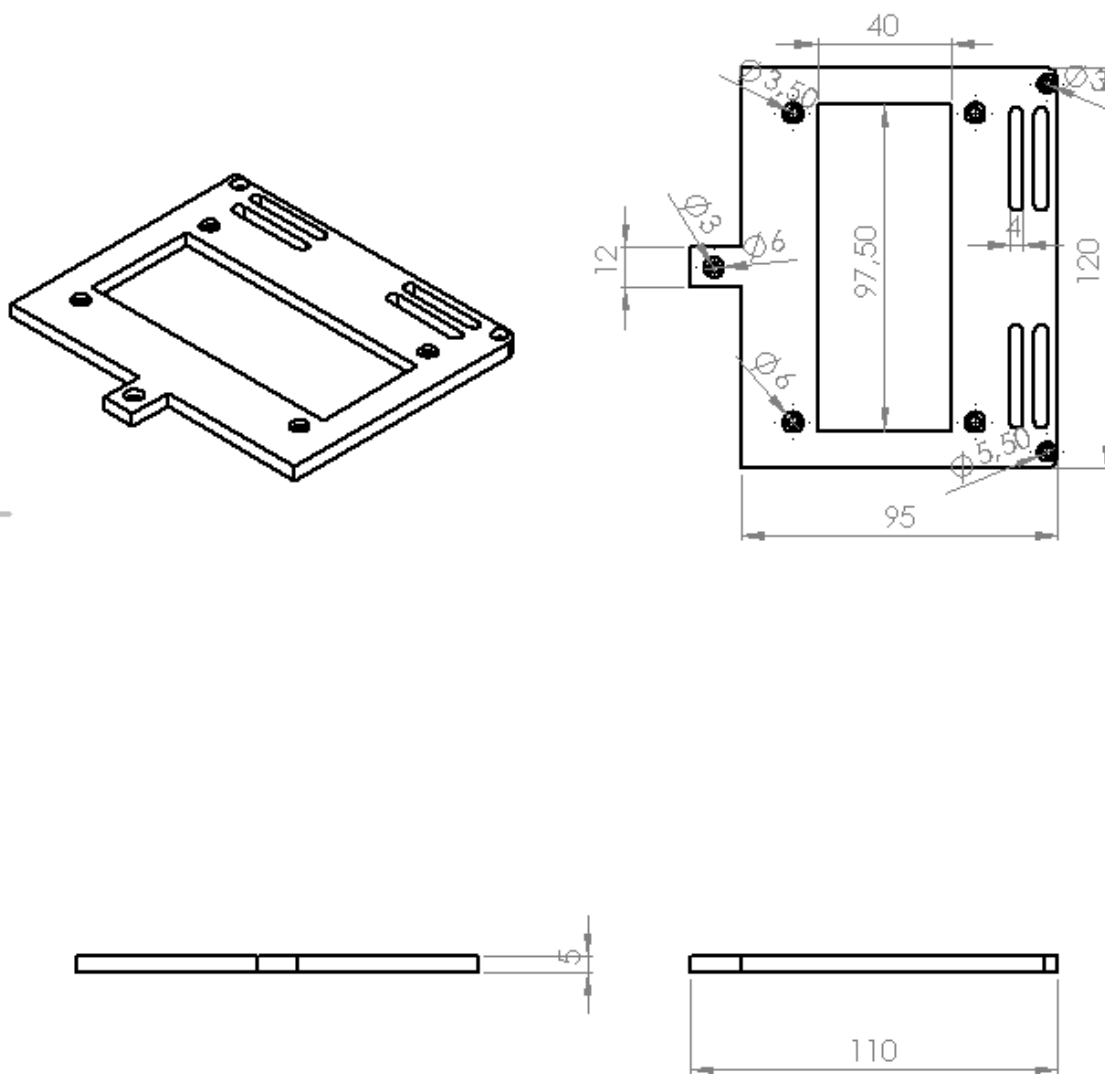


Figura 99. Parte superior de la tapa de la caja previa. Fuente: Autor.

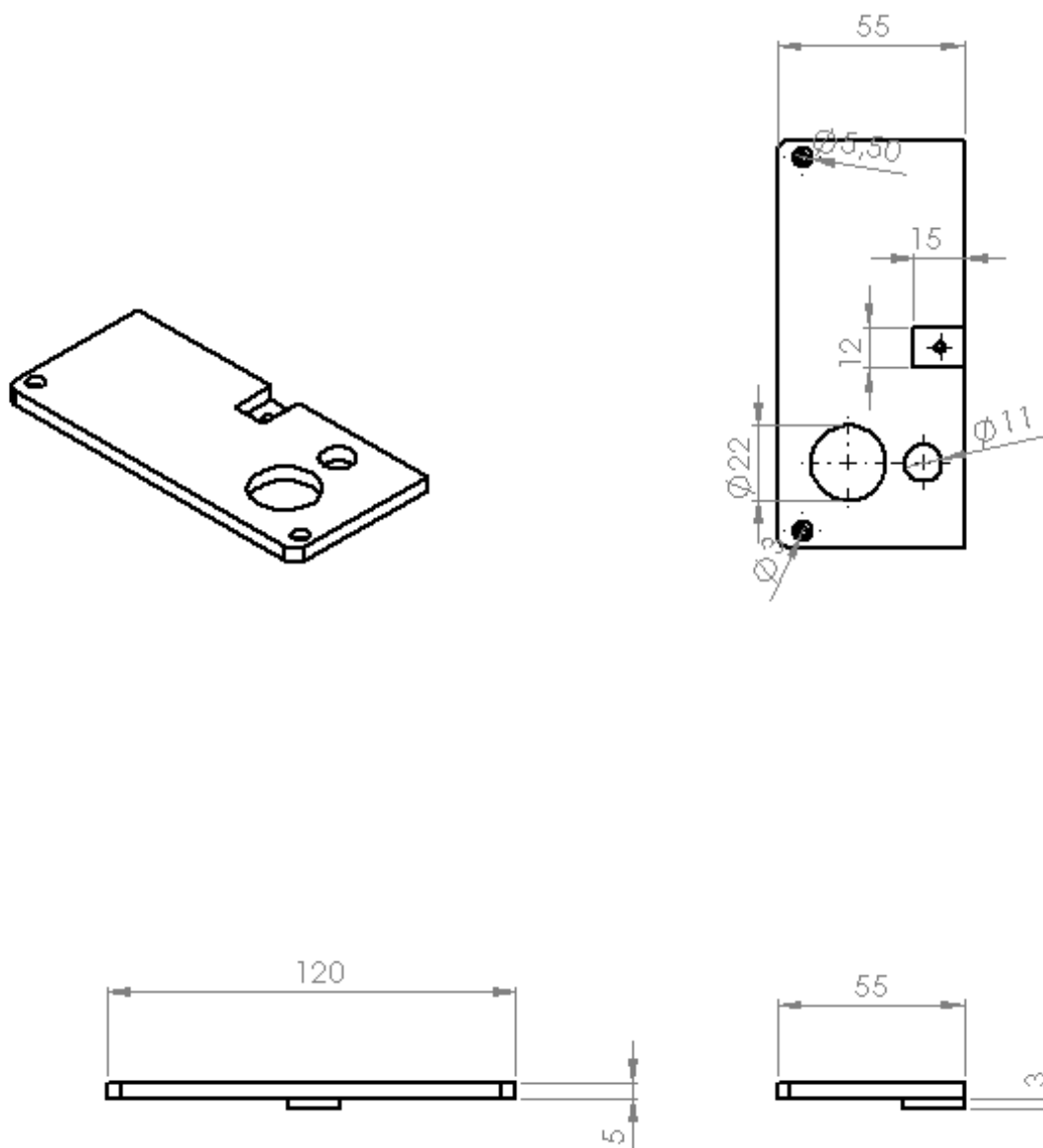


Figura 100. Parte inferior de la tapa de la caja previa. Fuente: Autor.

Apéndice C: Importar datos desde un archivo CSV a Excel

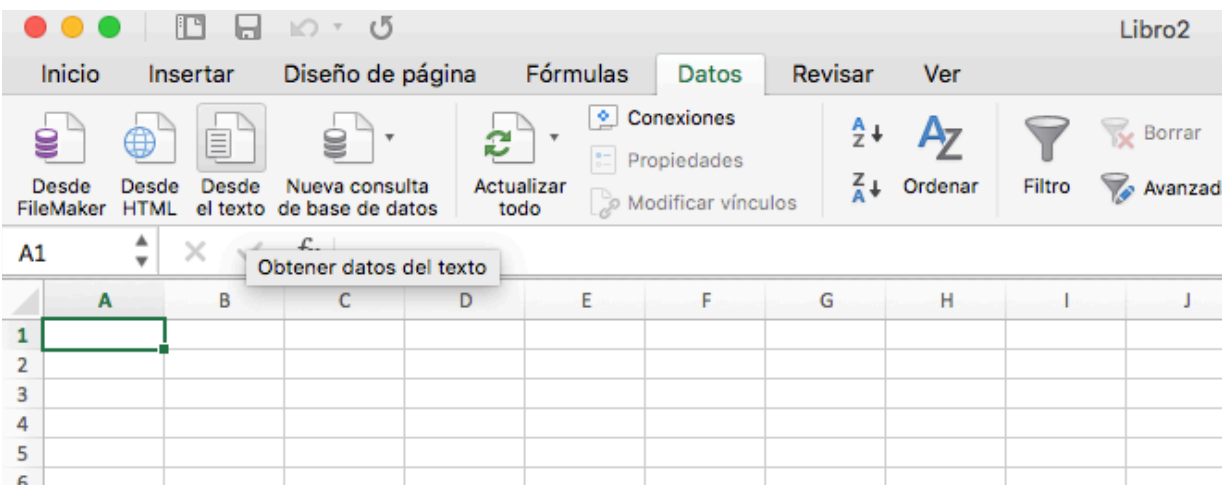


Figura 101. En la pestaña de datos, seleccionar desde el texto. *Fuente: Autor.*

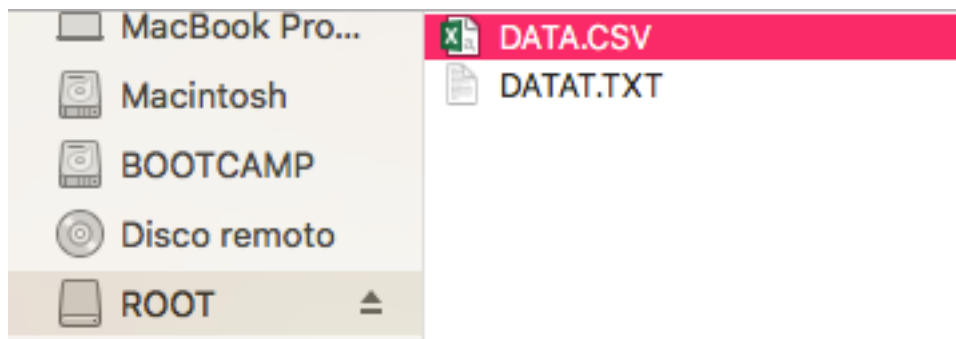


Figura 102. En la memoria SD/micro SD se busca el archivo CSV. *Fuente: Autor.*

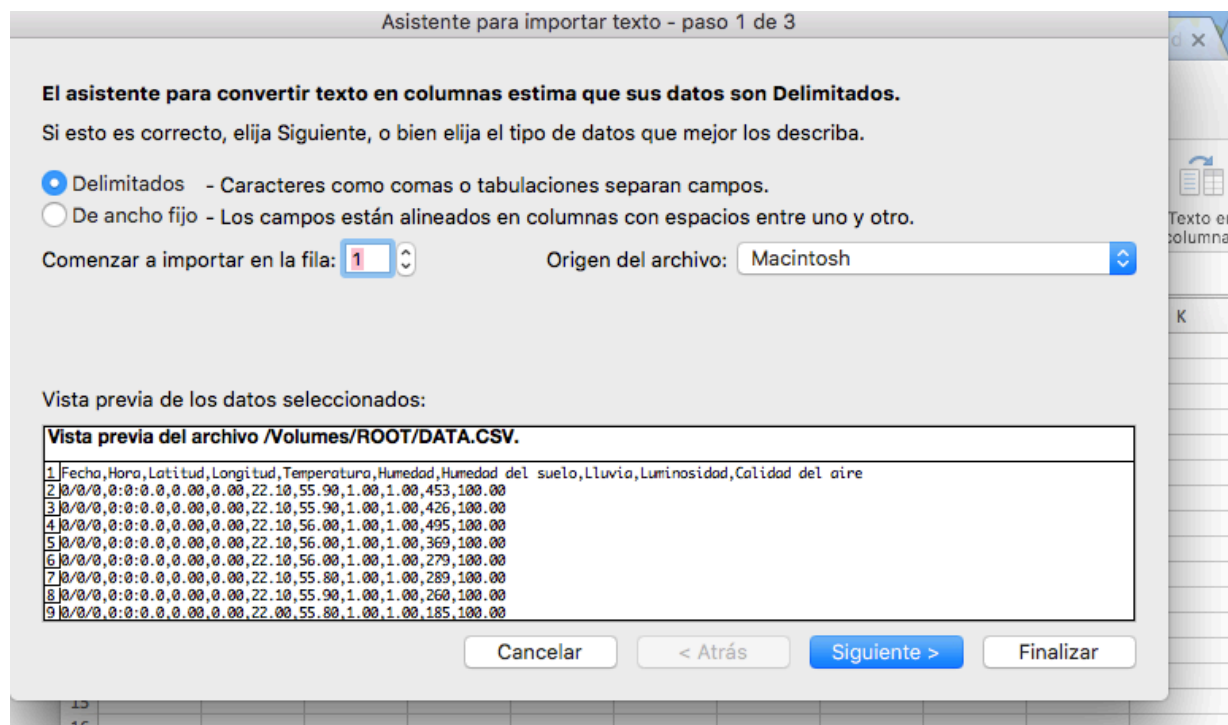


Figura 103. Seleccionar “Delimitados”. Fuente: Autor.

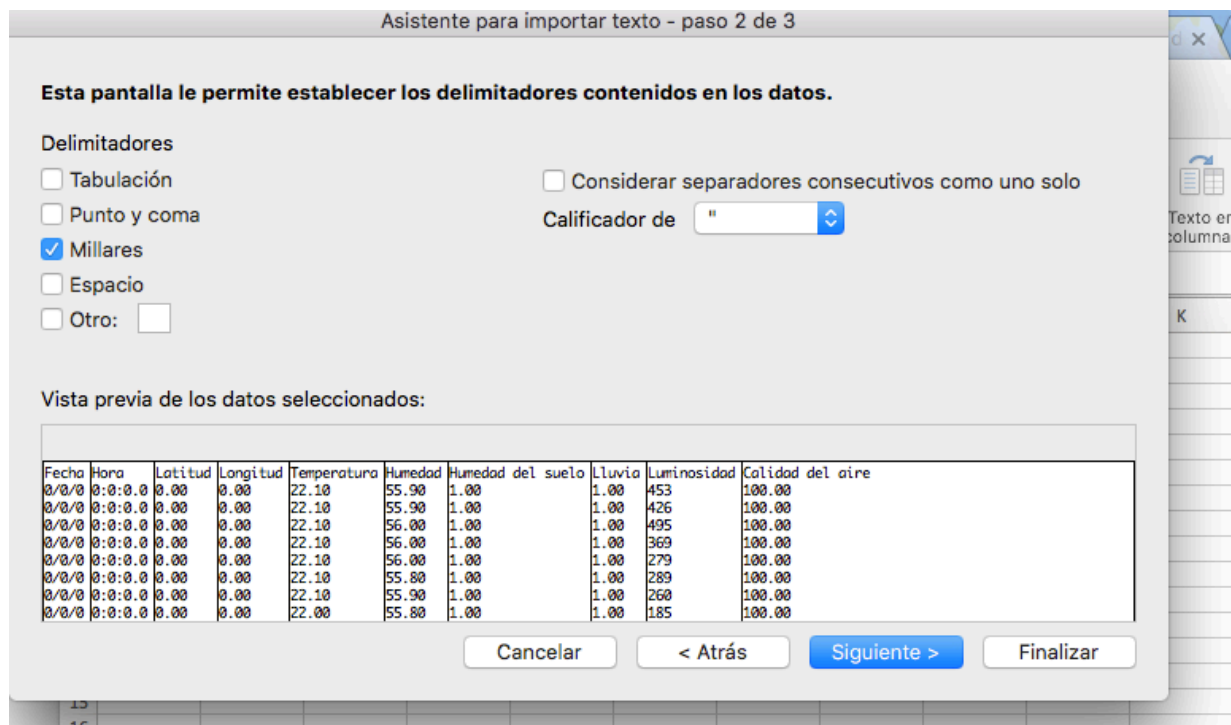


Figura 104. Seleccionar solo “Millares” (en otras versiones “Coma”). Fuente: Autor.

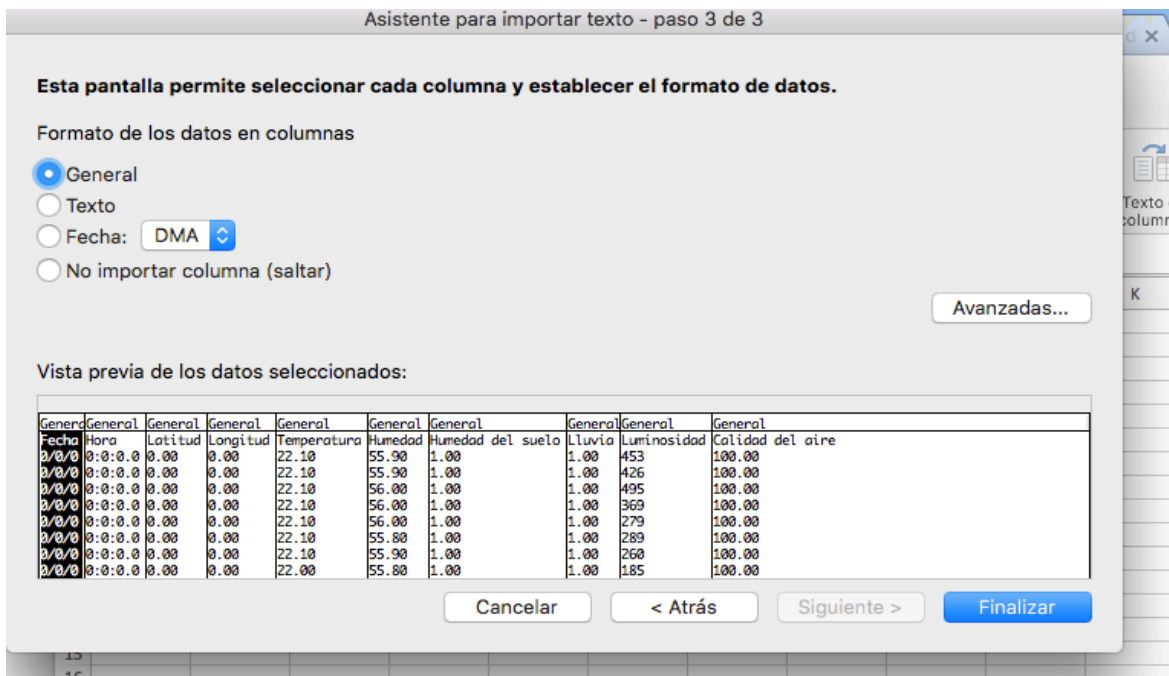


Figura 105. Se deja en “General”. Fuente: Autor.

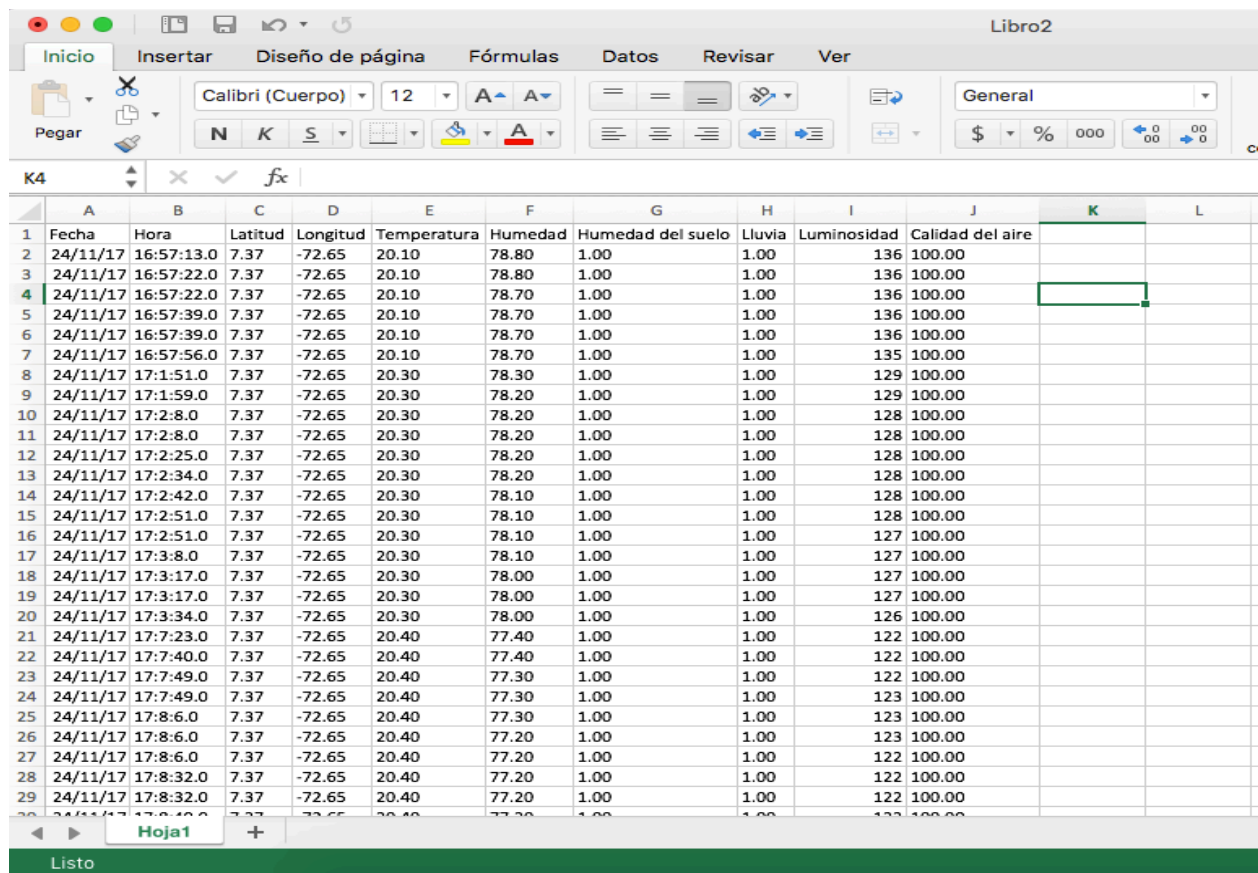


Figura 106. Datos importados. Fuente: Autor.