



Conocimiento y transferencia de tecnología sobre vehículos aéreos y acuáticos para el desarrollo transfronterizo de ciencias marinas y pesqueras

(POCTEP 0622-KTTSEADRONES-5-E)

Informe técnico

Actividad 2. Desarrollo de sensores y herramientas
informáticas para vehículos aéreos y marinos

Informe sobre el desarrollo de sensores en sistemas fijos
(boyas) y ROVs para su uso en estuarios y explotaciones
piscícolas

Producto final – Acción 2.3A1 | 30/06/2020

Nombre del Proyecto: KTTSeaDrones - Conocimiento y transferencia de tecnología sobre vehículos aéreos y acuáticos para el desarrollo transfronterizo de ciencias marinas y pesqueras

Código del proyecto: POCTEP 0622-KTTSEADRONES-5-E

Datos del proyecto

Localización	España y Portugal		
Fecha de inicio	01/10/2017		
Fecha de finalización	31/12/2021	Prórroga hasta	05/04/2022
Financiación total	€ 1.120.864,43		
Contribución UE	€ 840.648,32		
Cofinanciación UE (%)	75%		

Datos del beneficiario principal (BP)

Beneficiario principal	Universidad de Huelva
Persona de contacto	Vicerrector de Investigación y Transferencia
Dirección	C/ Cantero Cuadrado 6, 21004 Huelva
Teléfono	+34 959 218013
E-mail	kttseadrones@gmail.com
Página WEB (proyecto)	https://kttseadrones.wixsite.com/kttseadrones

Datos del documento

Nombre	Informe sobre el desarrollo de sensores en sistemas fijos (boyas) y ROVs para su uso en estuarios y explotaciones piscícolas
Acción	Acción 2.3A1. Desarrollo de un sistema fijo (boya) y un vehículo operado remotamente (ROV) con la capacidad de recoger, transmitir y almacenar información sobre las condiciones físico-químicas en estuarios y en balsas de engorde en instalaciones de producción acuícola
Autores	Alejandro Garrocho Cruz (UHU) Fernando Gómez Bravo (UHU) Juan Carlos Gutiérrez Estrada (UHU) Inmaculada Pulido Calvo (UHU) Jairo Castro Gutiérrez (UHU) Samuel López Domínguez (UHU) Antonio Peregrín Rubio (UHU)
Equipo técnico	Juan Antonio Gómez Galán (UHU) Raúl Jiménez Naharro (UHU) Rafael López de Ahumada (UHU) Manuel Sánchez Raya (UHU) Manuel Pedro Carrasco (UHU)

Javier Aroba Páez (UHU)	
Francisco José Moreno Velo (UHU)	
José Manuel Martín Ramos (UHU)	
Ana María Roldán Ruiz (UHU)	
Francisco Alfredo Márquez Hernández (UHU)	
Antonio Ángel Márquez Hernández (UHU)	
Ignacio de la Rosa Lucas (UHU)	
Fecha	30/06/2020
Versión (nº)	1.0

Las opiniones presentadas en este informe comprometen únicamente a los beneficiarios del proyecto y, en consecuencia, no representan en ningún caso la opinión oficial de los órganos de gestión del POCTEP.

ÍNDICE

RESUMEN	5
1. INTRODUCCIÓN	5
2. ACCESO REMOTO A UNA RASPBERRY PI CON SSH	7
2.1. RASPBERRY PI: DIRECCIÓN IP FIJA	7
2.2. SECURE SHELL (SSH)	8
2.2.1. Cliente Putty con protocolo SSH	8
3. VEHÍCULO OPERADO REMOTAMENTE (ROV)	9
3.1. INTRODUCCIÓN	9
3.2. HARDWARE	11
3.2.1. Sonar Ping360	12
3.2.2. Ping sonar ecosonda	18
3.2.3. Sonar de barrido lateral	19
3.2.4. WaterLinked	19
3.2.5. Oculus M750d	20
3.2.6. Sonda multiparamétrica	21
3.3. SOFTWARES	22
3.3.1. ArduSub	22
3.3.2. Companion web interface	23
3.3.3. QGroundControl	24
3.3.4. PingViewer	26
3.3.5. DeepView	27
3.3.6. Interfaz WaterLinked	28
3.3.7. Oculus ViewPoint	28
3.3.8. Calsens	29
3.4. PUESTA EN MARCHA	30
4. COMUNICACIÓN CON EL ROV	32
4.1. COMPONENTES ELECTRÓNICOS	32
4.2. ARQUITECTURA	33
4.3. OBTENCIÓN DE DATOS DE SENsoRES	35
4.3.1. Sonares BlueRobotics: PingProtocol	35
4.3.2. Pixhawk	37
4.4. DIRECCIONES Y PUERTOS	39
5. ALGORITMOS EN MATLAB	40
5.1. DATOS SONARES	40

5.1.1. Representación sonar Ping 360	40
5.1.2. Representación Ping sonar ecosonda	41
5.2. CONTROL MANUAL	42
5.2.1. Obtención del puerto del ROV	42
5.2.2. Creación GCS y suscripción al ROV	42
5.2.3. Almacenamiento de datos	42
5.2.4. Creación heartbeat	43
5.2.5. Control	44
5.3. POSICIONAMIENTO	44
5.3.1. Transformación de sistemas de coordenadas	45
5.3.2. Representación datos filtrados Ping360	47
5.3.3. Control con trayectoria	49
	50
5.4. CONTROL AUTOMÁTICO	51
5.4.1. Heading	51
5.4.2. Avanzar distancia especificada	52
6. BIBLIOGRAFÍA	54

Acción 2.3 Desarrollo de un sistema fijo (boya) y un vehículo operado remotamente (ROV) con la capacidad de recoger, transmitir y almacenar información sobre las condiciones físico-químicas en estuarios y en balsas de engorde en instalaciones de producción acuícola

RESUMEN

En este informe se ha desarrollado una descripción cronológica de los procesos realizados a lo largo del proyecto para el avance de la acción 2.3 (“Desarrollo de un vehículo operado remotamente (ROV) con la capacidad de recoger, transmitir y almacenar información sobre las condiciones fisicoquímicas en estuarios y en balsas de engorde en instalaciones de producción acuícola”), correspondiente a la UHU. Permitiendo hacer un seguimiento del progreso y conocer cuáles han sido nuestras tareas, pruebas de campo, dificultades y objetivos conseguidos.

De esta forma, queda constancia del trabajo realizado durante este periodo el cual ha sido desarrollado de forma ostensible, a pesar de las difíciles circunstancias y la cantidad de imprevistos debido a la pandemia.

1. INTRODUCCIÓN

Durante el principio del año 2020, se llevó a cabo las contrataciones del personal técnico para el desarrollo de las acciones propuestas en este proyecto. Inicialmente, se realizaron varias reuniones y conversaciones donde se trataban las acciones a realizar, objetivos que cumplir y los conceptos necesarios para poder desarrollar el proyecto, así como información sobre la instrumentación que teníamos y el equipamiento que recibiríamos a lo largo del año. También se nombraron las salidas exteriores que necesitaríamos hacer para poder comprobar el funcionamiento las tareas realizadas.

Durante este periodo, fue imprescindible realizar un periodo de aprendizaje sobre toda la información proporcionada, y así ser capaces de profundizar en los detalles del proyecto. Aunque no sabíamos cuáles eran las características específicas del equipamiento que estaba por llegar, sabíamos que el ROV tendría una estructura protagonizada por una raspberry, por lo que empezamos a realizar varias pruebas para poder acceder a los datos del microcontrolador para que en el futuro fuera más sencillo llevar a cabo este asunto. Este tema se detallará en el apartado dos.

Una vez se recibió el ROV Sibiu Pro a finales de julio de 2020, pudimos conocer cómo era el conexionado del ROV y todos los componentes que formaban el sistema completo. El primer objetivo sería su puesta en marcha para conocer las características del ROV, sus sensores y actuadores, además del software que utiliza cada dispositivo. Este proceso se ha detallado en el tercer apartado.

Realizada esta primera prueba, la siguiente finalidad consistiría en obtener los datos en crudo de los sensores y la pixhawk (apartado 4) y realizar un algoritmo propio para poder realizar misiones en el futuro (apartado 5), las cuales estarían formadas por un conjunto de acciones concretas. Se realizaría de forma autónoma y, de esta manera, no sería necesario el uso del control manual con el joystick.

En mayo de 2021 recibimos el nuevo ROV BlueROV2, que dispone de una estructura y componentes diferentes. Al igual que con el anterior ROV, el primer objetivo fue su puesta en marcha y comprobación de todos los componentes.

2. ACCESO REMOTO A UNA RASPBERRY PI CON SSH

Como hemos comentado anteriormente, sabíamos que el núcleo de los ROVs que recibiríamos tendría una raspberry. Por lo tanto, debíamos conocer cómo conectarnos a ella para poder desarrollar nuestra investigación. En la página web de NidoRobotics se encontraba una guía sobre la puesta en marcha, donde se pudo conocer que, para conseguir la comunicación con el robot, era necesario mantener una dirección fija en el panel de control de superficie y en la raspberry.

2.1. RASPBERRY PI: DIRECCIÓN IP FIJA

Cuando un dispositivo se conecta a un router, ya sea por cable o wifi, es probable que se le otorgue una dirección IP privada o local distinta. Normalmente, la dirección IP se asigna de forma dinámica proporcionada por el DHCP (Dynamic Host Configuration Protocol).

Para saber la dirección IP que el router le ha asignado a cada dispositivo, podemos usar el terminal, donde introduciremos el comando ‘ifconfig’ y nos aparecerá todas las conexiones de red disponibles en nuestra Raspberry Pi con la IP asignada a cada una de ellas.

Si nos fijamos, al ejecutar el comando ifconfig aparecen 3 conexiones:

- eth0: conexión de red ethernet a través de cable.
- wlan0: conexión de red a través de WiFi.
- lo: conexión para referirse a la propia interfaz de red de la Raspberry Pi.

De toda la información que aparece, nos tenemos que fijar en la línea `inet`. Como hemos comentado, dicha dirección IP puede cambiar cada vez que se vuelva a conectar o se reinicie la Raspberry Pi.

Una forma simple de comprobar la existencia de una dirección IP es utilizar el comando `ping`. Así podemos conocer la respuesta de si existe o no dicha IP.

Nuestro objetivo para el proyecto será acceder a la propia Raspberry Pi de forma remota, pero para ello debemos establecer una dirección IP fija, que nunca cambie, aunque reiniciemos el sistema.

Para configurar una IP fija debemos asegurarnos de que esté fuera del rango DHCP (configuración del router) y que ningún otro dispositivo tenga asignada la misma dirección IP.

La forma más sencilla de establecer una dirección IP desde Raspbian es desde el entorno gráfico. Para ello, haciendo doble clic sobre el ícono de configuración de red, seleccionamos la interfaz `eth0` si estamos conectados por cable de red o `wifi` en caso de estar mediante wifi.

Añadimos la dirección IP que tengamos libre y permitida en nuestro router. Añadimos también la máscara de red o router.

- IPv4: Dirección IP fija que queremos (dejar el /24 al final).
- Router: Dirección IP del router.
- DNS Servers: Dirección del servidor DNS (la del router).

A continuación, guardamos los cambios y reiniciamos la Raspberry Pi para ver que los cambios son correctos. Al reiniciar, volvemos a acceder a la configuración de red y veremos que la IP está guardada [1].

2.2. SECURE SHELL (SSH)

Una Raspberry Pi es muy útil como base para operar servidores gracias a su gran potencial y mínimo consumo. Para el acceso remoto es usual escoger Secure Shell (SSH) para realizar cambios en las configuraciones.

SSH, es un protocolo y el nombre del programa que lo implementa y es ampliamente conocido por ser un protocolo seguro para la administración remota de servidores, routers, switches,... El protocolo SSH permite manejar por completo el servidor o dispositivo de red mediante un intérprete de órdenes, además, también podemos redirigir el tráfico para ejecutar programas gráficos a través de la propia sesión SSH.

Otras características fundamentales de SSH son que nos va a permitir copiar datos de manera segura, tanto archivos como carpetas, a través del protocolo SFTP (SSH FTP), un protocolo hecho desde cero y que no tiene nada que ver con FTPS o FTPES (FTP sobre SSL/TLS). El protocolo SSH es fundamental en el ámbito de las redes y sistemas, además, podremos configurarlo en detalle para dotar a nuestro sistema de la máxima seguridad posible.

El protocolo SSH proporciona confidencialidad (los datos van cifrados punto a punto), autenticación (podremos autenticarnos frente al servidor SSH de múltiples maneras, con usuario/clave, criptografía de clave pública e incluso podremos configurar un segundo factor de autenticación), integridad (si los datos se modifican o los modifica un usuario malintencionado se podrá detectar, ya que usa HMAC para comprobar la integridad de todos y cada uno de los datos) [2].

2.2.1.Cliente Putty con protocolo SSH

OpenSSH es una de las herramientas más utilizadas en la red para establecer conexiones seguras a través de Internet. Este conjunto de herramientas y aplicaciones nació como una alternativa libre a Secure Shell para llevar las conexiones de manera sencilla y fiable a todos los usuarios que se conectarán a Internet, especialmente a través del protocolo SSH. Para conectarnos a través de este protocolo, es necesario disponer de un cliente que, en el caso de Windows, hay que recurrir a aplicaciones de terceros, como Putty [3].

Una vez fue posible conectarse, ya se podría modificar la configuración y obtener información de la raspberry del ROV si fuera necesario.

3. VEHÍCULO OPERADO REMOTAMENTE (ROV)

3.1. INTRODUCCIÓN

Los ROVs que vamos a tratar son vehículos sumergibles que están controlados por un ser humano que se encuentra fuera del propio ROV y se utilizan para observar y estudiar las profundidades marinas. La comunicación se realiza a través de un cable umbilical, permitiendo obtener los datos de la cámara, sensores y sonares situados en el vehículo.

En el proyecto disponemos de dos ROVs diferentes, el SibiuPro recibido en julio de 2020 y el BlueROV2 obtenido en mayo de 2021. Cada uno de ellos cuenta con una estructura electrónica similar, pero con una estructura física diferente, así como ciertos sensores extra que se le han añadido según los criterios para los que se van a utilizar.

En nuestros ROVs, la electrónica está formada por una Raspberry (ordenador reducido en una pequeña placa) y una PixHawk (controlador de vuelo de los drones aéreos).

La PixHawk es la encargada de controlar los 8 motores de propulsión, las luces y el motor que gira la cámara con un movimiento de cabeceo. Además, la PixHawk tiene integrado un acelerómetro, giróscopo o magnetómetro entre otros sensores para poder realizar el control a partir de los datos proporcionados por estos sensores.

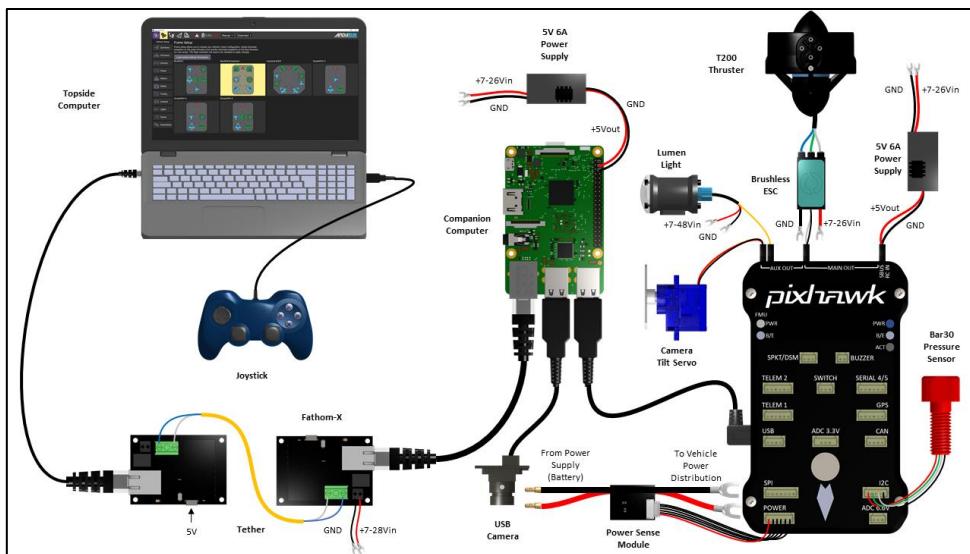


Figura 1. Componentes electrónicos [4].

Por otro lado, a la Raspberry se le conecta tanto la cámara como estos sensores que vemos a continuación, los cuales nos ayudarán a conocer el entorno del ROV:

- SibiuPro: este ROV dispone de un sonar de barrido lateral nos permite observar el fondo marino, con el ecosonda conoceremos la distancia hasta el fondo y con el Sonar360 tendremos información del entorno que nos rodea.

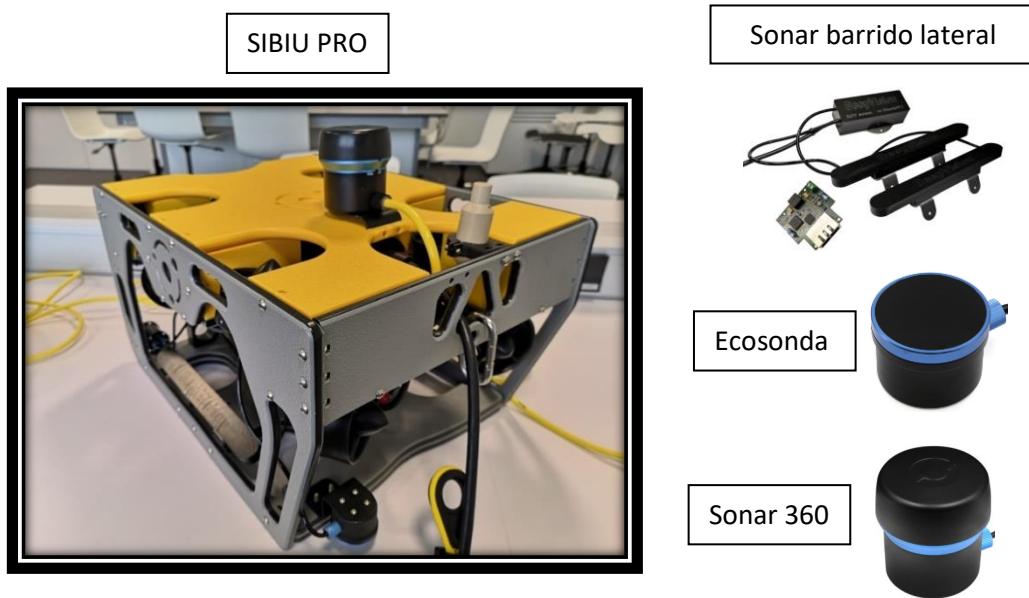


Figura 2. ROV Sibiu Pro.

Por último, con el Water Linked tenemos un sistema de posicionamiento que permite localizar el vehículo respecto a un sistema de referencia local haciendo uso de las sondas receptoras con las que viene equipado que interactúan con la sonda locator-S1 que el ROV tiene incorporado en la parte superior. Además, ofrece la posibilidad de geolocalizar el ROV globalmente mediante la fusión de la pose local y la posición global proporcionada por un receptor GPS.



Figura 3. WaterLinked.

- BlueROV2: en este ROV, además del ecosonda que es común a ambos, hemos añadido tanto el sonar Oculus750d que, a diferencia de otros, no tiene parte mecánica como una sonda multiparamétrica que nos permitirá conocer las propiedades del agua.

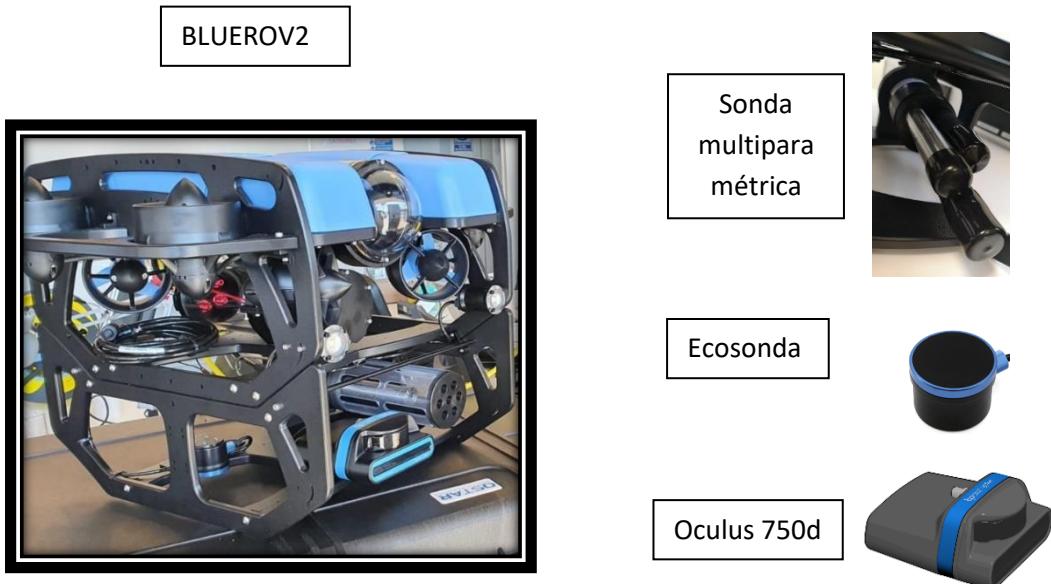


Figura 4. ROV BlueROV2.

Debido a las condiciones de mínima visibilidad que se dan en los estuarios donde los ROVs se van a desarrollar, la información de los sonares acústicos será imprescindible, puesto que la función de la cámara será mínima.

Como explicamos anteriormente, el primer objetivo con la llegada del ROV SibiuPro consistiría en su puesta en marcha para así comprobar el funcionamiento de todos los componentes, sus características, detalles de la información que proporcionan y software que utilizan. Con la llegada del BlueROV2, el procedimiento fue el mismo, confirmando el funcionamiento de sus dispositivos.

En este apartado se va a llevar a cabo la explicación del hardware, es decir, cómo funcionan los sensores que forman parte tanto del SibiuPro como del BlueROV2, además de mostrar el interfaz del software que utilizan y, concluyendo con los procedimientos necesarios para llevar a cabo la puesta en marcha de los ROVs. Para ello, ha sido necesario seguir la información proporcionada por las empresas que han suministrados los productos y estar en contacto telefónico con ellos para resolver las dudas que nos iban surgiendo.

3.2. HARDWARE

El ROV es un dron submarino con el que se podrán desarrollar trabajos de investigación, inspección y mantenimiento de instalaciones sumergida de manera eficiente y segura.

Su cámara de 1080p, específicamente optimizada para el entorno submarino, junto con sus 2 o 4 luces de 1500 lúmenes permiten obtener una imagen nítida en ambientes de baja luminosidad. Incorpora ocho propulsores, lo que le concede suavidad y estabilidad en la navegación. Además, la ingeniería con la que ha sido construido y su innovación tecnológica le permiten alcanzar profundidades de hasta 300m.

Las características principales y especificaciones técnicas de los ROVs son (medidas referidas al SibiuPro):

Especificaciones técnicas			
FÍSICO		CABLE UMBILICAL	
Altura	28,8 cm	Conductores	4 Pares trenzados, 26 AWG
Ancho	34,3 cm	Diámetro	7,6 mm
Largo	52,4 cm	Largo	25-300 m
Peso sin lastre	11,05 kg	Fuerza de trabajo	45 Kg
Peso con lastre	12,45 kg	Tensión de rotura	160 Kg
Agujeros/pasamuros	10 mm	Flotación (Agua salada)	Ligeramente positivo
Tubo principal	4" series	Flotación (Agua dulce)	Neutro
Tubo de la batería	3" series	Refuerzo	Reforzado con kevlar
Espuma de la flotabilidad	R-3318 uretano	Impermeabilizante	Poliacrilato de sodio
Peso del lastre	200 g		
Conector de batería	XT90		
Batería (14,8V, 18Ah)	4S		
RENDIMIENTO		SENSORES	
Profundidad máxima	300 m	Giroscopio 3-DOF	
Propulsores	T-200	Acelerómetro 3-DOF	
ESC	Astro 30A	Magnétómetro 3-DOF	
Corriente máxima de trabajo	3 Nudos	Barómetro interno	
Configuración de propulsores		Sensor de presión Bar30 Blue Robotics	
» 4 Verticales		Sensor de corriente y voltaje	
» 4 Horizontales		Sensor de humedad	
		Sensor de temperatura	
REQUISITOS MÍNIMOS PARA PILOTAR		CÁMARA	
Un computador con:		Cámara	1080 Digital
Procesador a 2,7 GHz		Campo de visión	110° horizontalmente
8 GB de RAM		Ángulo de inclinación de la cámara	+/- 90° (180° rango total)
Gráfica compatible con Direct 12		EMAX E508MD 13g Mini Metal	
Compatible con Windows, Mac y Linux		Digital Servo For RC Model	

Figura 5. Características ROV.

Como ya se ha comentado, además de los sensores que vemos en la tabla anterior, cada uno de ellos dispone de una sensorística añadida que detallamos en los siguientes apartados.

3.2.1.Sonar Ping360

Los sensores SONAR (SOund Navigation And Ranging) son un tipo de sonar activo que funciona transmitiendo pulsos de sonido al agua y registrando los ecos que se devuelven a medida que cada pulso de sonido se refleja en los objetos frente a él.

Los sonares de barrido se definen por tener un haz acústico en forma de “abanico” con un haz vertical ancho y un haz horizontal estrecho para obtener secciones transversales acústicas del entorno, para luego mostrar las secciones transversales como una sola imagen.

Las imágenes acústicas y los retornos de los sonares de exploración pueden ser difíciles de entender. Aunque en las ilustraciones se utiliza Ping360, los conceptos son lo suficientemente generales como para ser aplicados a cualquier sonda de exploración mecánica [5].



Figura 6. Sonar Ping360.

Ese transductor está montado en un motor que lo gira en incrementos de 0.9 grados y, mientras lo hace, genera una imagen circular de los alrededores de la sonda con un alcance máximo de 50 metros. El resultado es similar a lo que podría ver en un radar meteorológico en las noticias locales o en un escáner láser en un robot autónomo.

Al combinar la velocidad conocida del sonido en el agua con el tiempo en que se recibieron los ecos, el sonar puede calcular la distancia que ha viajado el sonido. La ecuación para determinar la distancia acústicamente es:

$$dist = \text{speed sound in water} \cdot \frac{\text{time to return}}{2}$$

Por lo general, la velocidad del sonido en agua salada es de aproximadamente 1500 m/s, pero esto puede variar según la temperatura del agua, la salinidad y la profundidad de funcionamiento del sonar.

Los objetivos con densidades de material muy diferentes a las del agua (como gas, roca, hormigón o metal) serán muy reflectantes y tendrán fuertes ecos. Los ecos de materiales como barro, limo, arena y plantas tendrán ecos más débiles, ya que tienen una densidad similar a la del agua o absorben energía acústica. Del sonar, se obtiene 1200 valores de la fuerza del eco en cada ángulo, cada uno de ellos hacen referencia a una distancia concreta y consisten en un valor entre 0 y 255 (sin rebote e intensidad máxima, respectivamente). Estos se muestran en el interfaz a través de una paleta de colores, teniendo un color diferente dependiendo de la fuerza del rebote.

Cuando se monta en un ROV, la sonda de exploración le brinda una vista de arriba hacia abajo de los alrededores del ROV. Puede ver ecos de objetos como cuerdas, paredes, pilotes de muelles, rocas, naufragios, barcos, peces y cualquier otra estructura u objeto que refleje ondas sonoras. Con eso, tiene puntos de referencia para navegar, independientemente de la visibilidad del agua, y puede ubicar características importantes en el agua rápidamente.

PATRONES DE HAZ ACÚSTICO:

Cuando se imagina cómo funcionan acústicamente los sonares de exploración, a menudo se los compara con encender una linterna en un área oscura. Solo el área iluminada por el haz de la linterna es visible para el usuario, el resto permanece oscuro.

En comparación con un haz de luz, los sonares de exploración tienen un haz acústico con un ancho y una altura fijos llamados "patrones de haz horizontal y vertical". Será este haz acústico el que "ilumine" el área submarina circundante con energía acústica en lugar de energía de luz visible.

Los sonares de barrido suelen tener un haz vertical ancho y un haz horizontal estrecho para producir un "corte" estrecho delante del transductor.

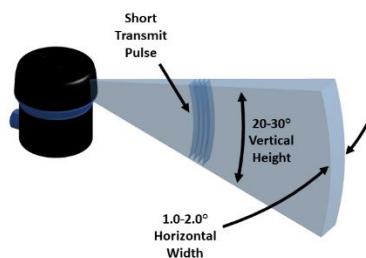


Figura 7. Haz acústico del sonar.

Para "dibujar" una imagen o localizar objetivos, el cabezal del transductor dentro del sonar de exploración gira mecánicamente a través de un motor paso a paso y mueve el cabezal en un arco para construir los "cortes" en una imagen en el software de visualización.

El usuario no podrá ver todo a la vez con el haz estrecho de la linterna, pero la creación de imágenes de la cámara permitirá al usuario ver el área completa como una sola imagen, como lo haría una imagen panorámica.

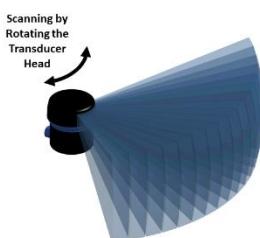


Figura 8. Múltiples haces acústicos.

VISIBILIDAD DE OBJETIVOS:

Los objetivos dentro del patrón de haz horizontal y vertical del sonar serán "iluminados" acústicamente y sus ecos se reflejarán de regreso al sonar para ser recibidos e interpretados como una imagen.

Los objetivos fuera del patrón de haz no se mostrarán en la pantalla del visor.

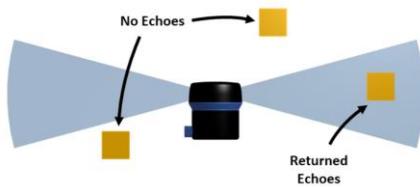


Figura 9. Ecos de objetos.

Las sondas de escaneo no pueden diferenciar entre objetos con el mismo ángulo de llegada vertical, también conocido como "rango inclinado".



Figura 10. Visión objetos verticales.

SOMBRA ACÚSTICAS:

La analogía de la linterna puede seguir utilizándose cuando el sonar localiza un objetivo en el fondo del mar para ayudar a determinar la altura, la forma y la orientación. Una "sombra acústica" se mostrará detrás de un objetivo iluminado de la misma manera que una "sombra de luz visible" será visible detrás de un objeto iluminado con una linterna.

Si el sonar tiene una altitud relativamente alta y un ángulo pronunciado, las sombras acústicas serán cortas. Las sombras acústicas cortas a veces pueden ser difíciles de ver y dificultan la evaluación del objetivo.

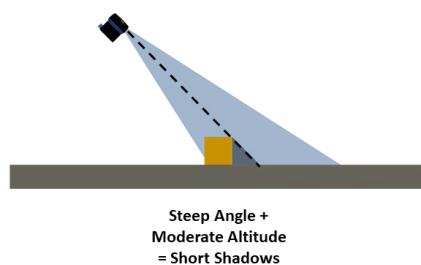


Figura 11. Inclinación alta.

Si el sonar tiene una altitud más cercana al fondo y un ángulo hacia abajo menos pronunciado, las sombras acústicas proyectadas por los objetivos serán más largas, lo que permitirá una interpretación más fácil.

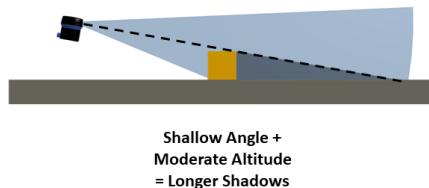


Figura 12. Inclinación baja.

Las sombras acústicas proyectadas por objetivos que están lejos serán relativamente estrechas según la geometría angular del haz del sonar. El ancho de la sombra acústica aumentará a medida que la sonda se acerque al objetivo. Con sombras más amplias, es posible que no se vean objetivos adicionales dentro de la sombra acústica, ya que no hay ninguna señal acústica que los ilumine.

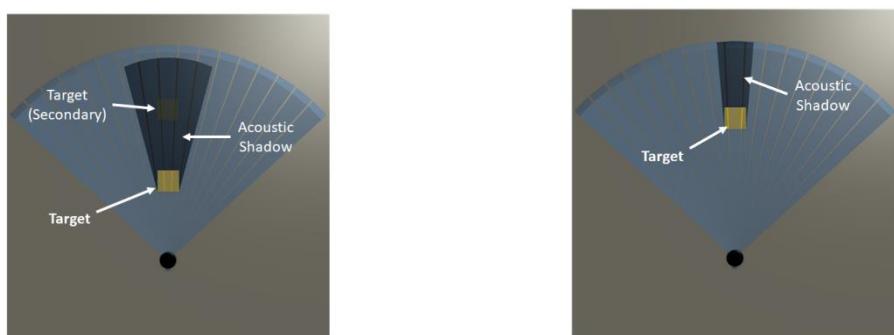


Figura 13. Sombras acústicas.

LOCALIZACIÓN DE OBJETIVOS:

La localización de objetivos en el lecho marino o en la columna de agua es otro uso de un sonar de exploración que no sea para fines de navegación.

Aprender a utilizar un sonar de exploración para encontrar objetivos requerirá algo de práctica, ya que los objetivos más pequeños son mucho más difíciles de localizar e identificar que los más grandes.

Para la localización de objetos, la clave es maniobrar y girar lentamente el ROV, permitiendo el tiempo necesario para generar nuevas imágenes. El proceso se realizaría así:

1º) Colocar el ROV en una posición estable.



Figura 14. Localización de objetos: paso 1.

2º) Permitir la generación de un escaneo polar de 360 grados.

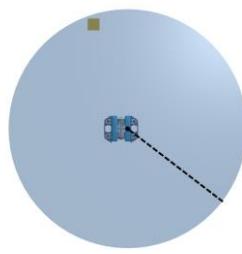


Figura 15. Localización de objetos: paso 2.

3º) Identificar el rumbo relativo del objetivo.

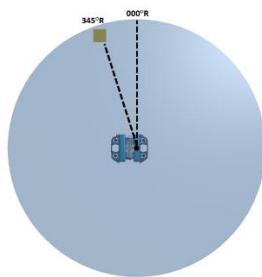


Figura 16. Localización de objetos: paso 3.

4º) Girar el ROV para alinear el objetivo en un rumbo relativo cero.

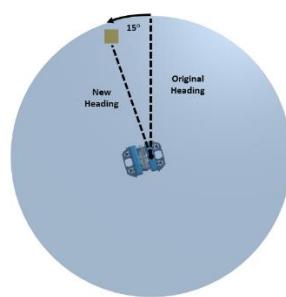


Figura 17. Localización de objetos: paso 4.

5º) Reducir el escaneo de sector de la sonda a aproximadamente 60-90 grados para aprovechar la frecuencia de actualización más rápida.

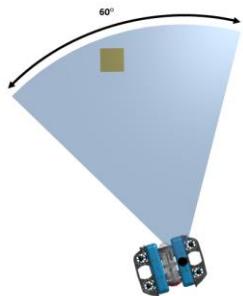


Figura 18. Localización de objetos: paso 5.

6º) Mantener el contacto con el objetivo en la sonda mientras el vehículo avanza.

3.2.2. Ping sonar ecosonda

La sonda Ping es un ecosonda multipropósito de un solo haz. Se puede utilizar como altímetro para ROV, para trabajos de batimetría a bordo de un USV, como sonar para evitar obstáculos y para otras aplicaciones de medición de distancias submarinas.



Figura 19. PingSonar ecosonda.

Una ecosonda, como la Ping, es una de las formas más simples de sonar submarino. Funciona mediante el uso de un transductor piezoelectrico para enviar un pulso acústico ultrasónico al agua y luego escucha el retorno de los ecos. Con esa información, puede determinar la distancia al eco más fuerte, que suele ser el fondo del océano o un objeto grande. También puede proporcionar la respuesta de eco completa (intensidad del eco en función del tiempo) que se puede trazar como la pantalla de una sonda de sonda.

El Ping utiliza una frecuencia de transductor de 115 kHz, diferente de las que se utilizan en la mayoría de las ecosondas de barcos para evitar interferencias. Tiene un rango de medición de 30 metros y un ancho de haz de medición de 30 grados, perfecto para aplicaciones en un bote oscilante o para evitar obstáculos. Un algoritmo avanzado de seguimiento del fondo se ejecuta en el dispositivo para determinar la distancia al fondo marino, incluso en situaciones complicadas con múltiples ecos [6].

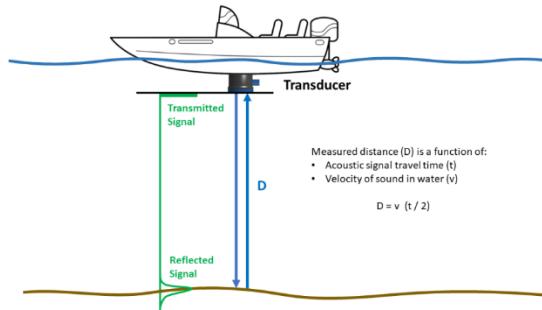


Figura 20. Funcionamiento ecosonda.

3.2.3. Sonar de barrido lateral

La sonar de barrido lateral (modelo BR-ROV) consta de dos partes, la caja de la sonda y los transductores. Ambas están montados en el exterior del ROV. Cuando el ROV esté encendido, el sonar será detectado automáticamente por el software DeepView y el sistema estará preparado para su uso [7].



Figura 21. Sonar de barrido lateral.

El sonar de barrido lateral tiene un funcionamiento común con el resto de sonares que ya hemos visto.

3.2.4. WaterLinked

El sistema WaterLinked, está formado por la maleta de superficie Master D1, receptores colocados en el entorno y el localizador que se encuentra en la parte superior del ROV. Esta maleta de superficie, a su vez, está compuesta por varios elementos, como son un GPS y una unidad de medición inercial (IMU).

El sistema GPS submarino se basa en el posicionamiento acústico Short Baseline (SBL). El localizador se coloca en el ROV y funciona como una baliza que envía un pulso acústico. Cerca de la superficie, hay cuatro receptores sumergidos en el agua. Los receptores escuchan el

pulso del localizador. El tiempo de llegada a cada receptor se utiliza para calcular la posición del localizador. Los sistemas SBL, en comparación con los sistemas USBL, tienen la ventaja de funcionar bien en aguas poco profundas y en entornos acústicos reflectantes, como en jaulas para peces, cerca de instalaciones portuarias, cerca de cascos de barcos, dentro de tanques de agua, etc.

Una vez que se conoce la posición relativa a los receptores, la posición global se puede encontrar añadiéndola a la posición obtenida por un receptor GPS. El sistema de GPS subacuático hace esa parte internamente para que pueda proporcionar la posición global real del ROV como salida [8].

El master D-1 calcula la posición de cualquier localizador (locator-S1) en relación con la posición de los receptores (receivers-D1). Combinando GPS e IMU, que se encuentran integrados en el interior del master, proporcionará la posición GPS absoluta del localizador. Con una pequeña introducción de medidas, el kit se integra fácilmente y se estará ejecutando en un servidor web interno que expone una GUI y un API.

Gracias a este sistema, el ROV puede proporcionar el dato sobre su posición en el sistema global (latitud y longitud).

3.2.5. Oculus M750d

Los sonares tradicionalmente han sido dispositivos mecánicos que funcionaban girando sus transductores (transmisores y receptores) para escanear el sector frente a ellos. En cada posición del transductor, un pulso de sonido es transmitido al agua y los ecos recogidos por el receptor son representados en la pantalla. Dependiendo de la resolución requerida, una imagen podría actualizarse en décimas de segundos y cualquier movimiento del sonar durante este período podría manchar y distorsionar las imágenes.

A diferencia de los sonares mecánicos, Oculus es un sonar multihaz que no tiene partes móviles. Por lo tanto, una serie de receptores recopilan ecos de un solo pulso de transmisión y se combinan matemáticamente los datos en una imagen utilizando un proceso conocido como "beam-forming". Esto permite que las imágenes se produzcan muchas veces por segundo y se vean en tiempo real como la salida de una cámara de video.



Figura 22. Oculus M750d.

El sonar que dispone el BlueROV2 es de doble frecuencia, es decir, tienen baja y alta frecuencia, lo que permite al operador elegir entre rangos acústicos más largos o imágenes de alta resolución, respectivamente.

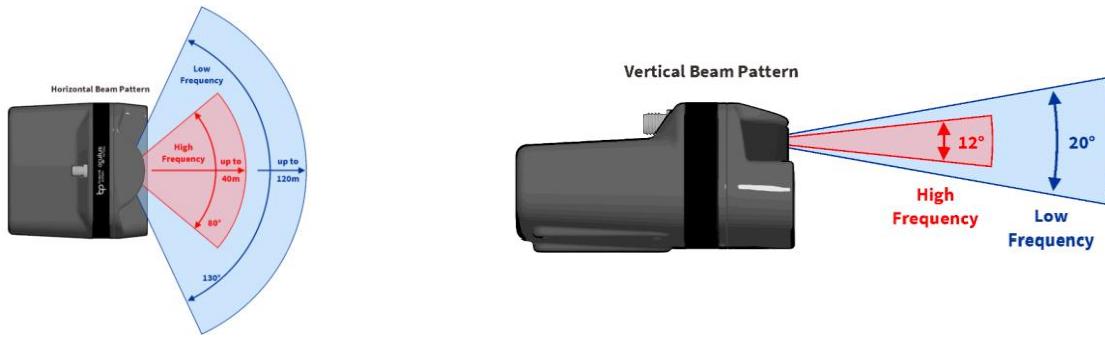


Figura 23. Rango Oculus.

Las imágenes de la sonda pueden requerir algo de entrenamiento y práctica para interpretarlas, ya que la pantalla le aparece al operador como una vista de arriba hacia abajo que cubre el área frente a ellos, iluminando acústicamente un sector hasta 130 ° horizontalmente y 20° verticalmente dependiendo del modo de frecuencia seleccionado [9].

3.2.6. Sonda multiparamétrica

La sonda multiparamétrica esta compuesta por 3 sensores difetentes:

- **C4E sensor:** el electrodo trabaja con una tecnología en 4 electrodos: una corriente alterna de voltaje constante es establecido entre un par de electrodos primarios en grafito. Los electrodos del secundario en platino permiten regular el voltaje impuesto a los electrodos primarios para reflejar el ensuciamiento. El voltaje medido entre los electrodos del primario está en función de la resistencia del lugar y, por lo tanto, de la conductividad. El valor medido de la conductividad se compensa automáticamente con el temperatura y se transfiere sin interferencias a la unidad de visualización conectada y al controlador a través de un interfaz digital [10].
- **OPTOD sensor:** el sensor de oxígeno disuelto OPTOD aplica la medición óptica basada en luminiscencia tecnología y mide de forma fiable y precisa sin necesidad de calibración. Sin consumibles ni mantenimiento requerido, el sensor OPTOD ofrece un retorno inmediato al inversión. La única intervención necesaria es reemplazar el disco DO cada dos años. Dado que no consume oxígeno, el sensor OPTOD se puede utilizar en todos los medios; incluso cuando hay un flujo de agua muy débil. El valor medido de oxígeno disuelto se compensa automáticamente con la temperatura, presión del aire y salinidad (contenido de sal), y se transfieren sin interferencia a la unidad de visualización y controlador conectados a través de una interfaz digital. La tapa de la membrana es fácil de reemplazar lo que significa que el sensor es muy fácil de mantener. Los datos de calibración actuales se guardan directamente en la electrónica del sensor. Como resultado, la función Plug and Play del sistema se habilita sin la necesidad para recalibración [11].
- **NTU sensor:** la medición de la turbidez según DIN EN ISO 7027 es un método de control probado y comprobado aguas con turbidez de baja a fuerte. El principio de medición del

sensor de turbidez NTU se basa en una medición de luz infrarroja según el método de luz dispersa a 90 °. Gracias a la medición de la luz en una longitud de onda de 880 nm y el amplio rango de medición de 0 a 4.000 NTU, el sensor se puede utilizar en una serie de Aplicaciones de tratamiento de agua y aguas residuales, p. ej. para controlar la calidad del agua, filtro de detección fugas, y para inspección final de aguas residuales. Aparte del valor de turbidez, el sensor también mide la temperatura del medio y Suspendedo Sólido en mg / L (0-4500 mg / L) [12].



Figura 24. Sonda multiparamétrica.

3.3. SOFTWARES

3.3.1. Ardusub

El sistema de control ArduSub se compone de tres componentes de software principales:

➤ Firmware del piloto automático ArduSub.

El firmware ArduSub es el software del piloto automático responsable de procesar la entrada del piloto y controlar el ROV. ArduSub es el "cerebro" del ROV.

➤ Software de computadora complementario.

El software de computadora complementario transmite las comunicaciones entre el piloto automático y QGroundControl a través de comunicaciones Ethernet (Raspberry). El software Companion también transmite video HD a QGroundControl.

➤ Software QGroundControl.

QGroundControl es la interfaz de usuario para operar el ROV.



Figura 25. ArduSub.

Aquí hay un diagrama sobre qué software se carga en cada componente hardware y sus conexiones básicas [13]:

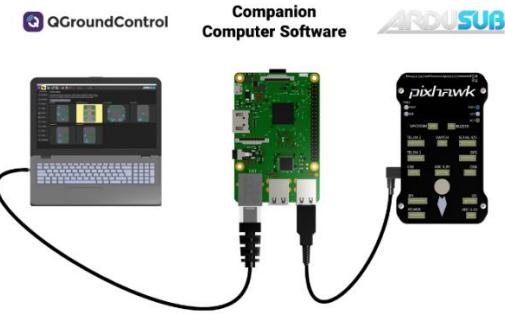


Figura 26. Diagrama componentes principales.

ArduSub viene instalado de fábrica en el controlador de vuelo del ROV como software de a bordo. Este controlador de vuelo consiste en una PixHawk, que es un popular controlador de vuelo de propósito general basado en el diseño de hardware abierto. Este, controla la dinámica del ROV, así como las funciones de auto-estabilización y telemetría.

3.3.2. Companion web interface

Para acceder a esta interfaz hay que dirigirse a la dirección: 192.168.2.2:2770 y, entonces, aparece la ventana de Network, donde tenemos disponibles diferentes pestañas [14]:

➤ **Network:**

- **Wifi Setup:** Opciones para conectar el ROV a una red Wifi disponible (necesario solo para actualizaciones).
- **Wifi Status:** Informa de la red a la que está conectado y si hay o no conectividad a internet.
- **Bandwidth and Latency Testing:** Herramienta para hacer un test de ancho de banda y latencia desde el ordenador al ROV.
 - **Upload:** Realiza un test de subida
 - **Download:** Realiza un test de bajada.

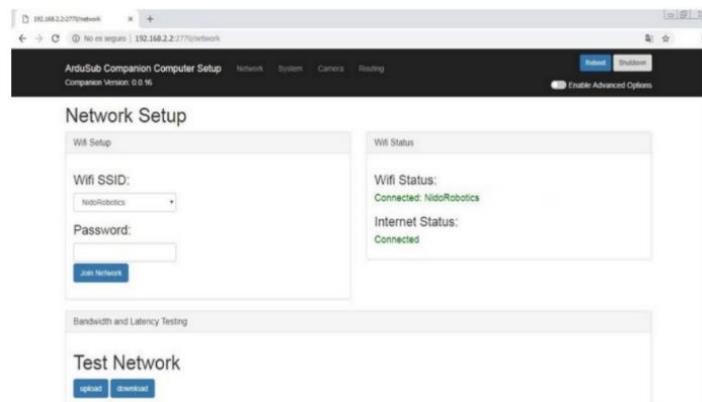


Figura 27. Companion Web Interface.

Pulsando arriba a la derecha en "mostrar opciones avanzadas". Con la opción "Manual" (por defecto): El ROV funcionará con la dirección IP 192.168.2.2 (El PC deberá estar configurado en la dirección 192.168.2.1 conforme a las indicaciones de los manuales).

➤ **System:**

En esta pestaña tenemos datos relacionados con la Raspberry, uso de CPU y RAM, servicios en ejecución, hardware conectado y versión, actualizaciones y datos de ArduSub.

➤ **Camera:**

En la pestaña camera podemos ver y modificar los parámetros de la cámara del ROV. A la derecha está la configuración de la retransmisión en directo de la imagen del ROV y a la izquierda los parámetros propios de la cámara. Estos cambios afectarán a la grabación del vídeo.

Es posible guardar los cambios en los ajustes de video en diferentes perfiles. Por ejemplo, puede guardarse un perfil para condiciones de mala visibilidad, otro para condiciones de contraluz, etc. Todos los perfiles se guardan en la memoria del ROV.

➤ **Routing:**

Se puede utilizar para enrutar comunicaciones serie entre dispositivos y puertos de red.

3.3.3. QGroundControl

Este software permite al usuario pilotar el ROV mientras visualiza el vídeo y los datos de navegación en tiempo real.

Una vez abramos el programa, cuando QGroundControl reconozca el ROV, veremos el logo de ArduSub en la esquina superior derecha, señal de que QGC recibe datos desde el ROV.



Figura 28. QGroundControl.

En el ícono del engranaje, tenemos las configuraciones del ROV. Éstas se quedan guardadas en el ROV y se conecte donde se conecte permanecerán así.

En la pantalla de "Summary" tenemos un resumen del estado y las calibraciones del ROV:

- Sensors: los sensores, si está en verde, están todos correctamente.
- Power: tipo de batería y capacidad.
- Safety: opciones configuradas de seguridad.
- Camera: opciones de los canales del servo de la cámara.
- Ligths: configuración de luces.
- Frame: la configuración de motores y la versión del software.

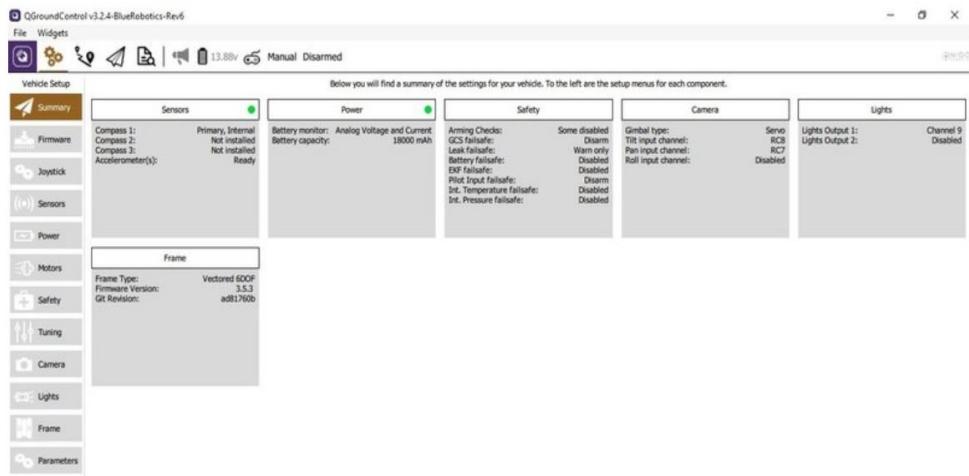


Figura 29. Interfaz QGroundControl.

Después podemos observar información detallada de cada uno de estos apartados en sus secciones concretas, permitiéndose realizar calibraciones de los sensores y el mando, modificaciones en sus botones, visualización de los parámetros, etc.

También tenemos el apartado "Sensors", que es la pestaña para poder calibrar los sensores del ROV, tal como el acelerómetro y el compass. También se puede calibrar el nivel de horizonte. Si las opciones están en verde, no hace falta calibrar, si están en amarillo sería recomendable realizar las calibraciones otra vez. Si están en rojo debería volver a hacerse la calibración. Normalmente solo pedirá que se recalibre si se han reseteados los parámetros del ROV, o si hay un cambio notable en la situación geográfica desde la última calibración.

Modos de navegación

Actualmente están soportados tres modos de navegación:

- "Manual": Permite mover el ROV sin que éste realice ningún tipo de corrección. Suele utilizarse en superficie, para testear motores.
- "Stabilize": El ROV realiza correcciones automáticas de Pitch, Roll y Yaw (cabeceo, rumbo y guiñada).
- "Depth Hold": Incluye un control automático de profundidad.

Grabación de video

Para grabar el vídeo, desplegamos el menú que tenemos debajo de la brújula, y pulsamos en "Video Stream".



Figura 30. Visión cámara ROV.

Durante la inspección, volvemos a la pestaña de "Values" donde podemos visualizar los datos de telemetría. Todos los datos que tengamos activados se guardarán junto con el video en un archivo de subtítulos, que podremos visualizar con reproductores de video [15].

3.3.4. PingViewer

PingViewer es una interfaz gráfica de usuario que te permite conectar, configurar, ver y registrar datos de los dispositivos Ping (Sonar 360 y Ping sonar):

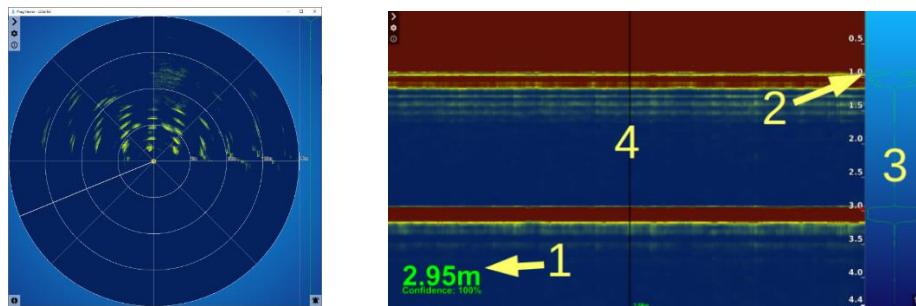


Figura 31. Representación en PingViewer (Ping360 y PingSonar ecosonda).

La interfaz para el ping sonar se compone de cuatro apartados:

1. Distancia de lectura
2. Distancia del eje
3. Gráfico de retorno
4. Cascada

La configuración nos permite modificar algunos parámetros para optimizar la recepción de los datos en función de las condiciones del entorno donde nos encontramos y del objetivo. Esta

configuración consiste en un menú principal y una configuración avanzada, difiriendo entre ambos dispositivos [16]:

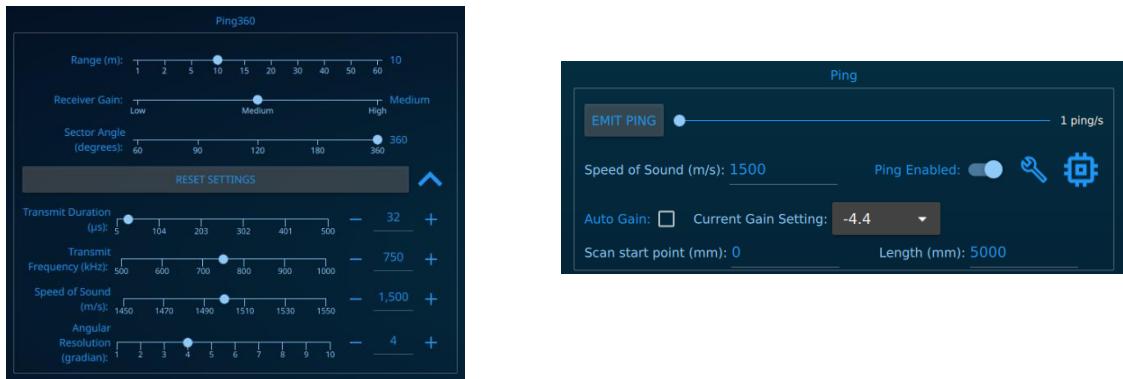


Figura 32. Ajustes PingViewer (Ping360 y PingSonar ecosonda).

3.3.5. DeepView

Este programa va a ser el encargado de permitirnos observar y grabar la información captada por los sonares de barrido lateral. Cuando el sonar conecte, veremos que se activa la barra de herramientas. El mismo programa puede utilizarse para visualizar a posteriori los datos tomados.

Configuración recomendada: 1.5 nudos (kn) con la ganancia del ROV entre el 50% y el 75%, dependiendo de la corriente.

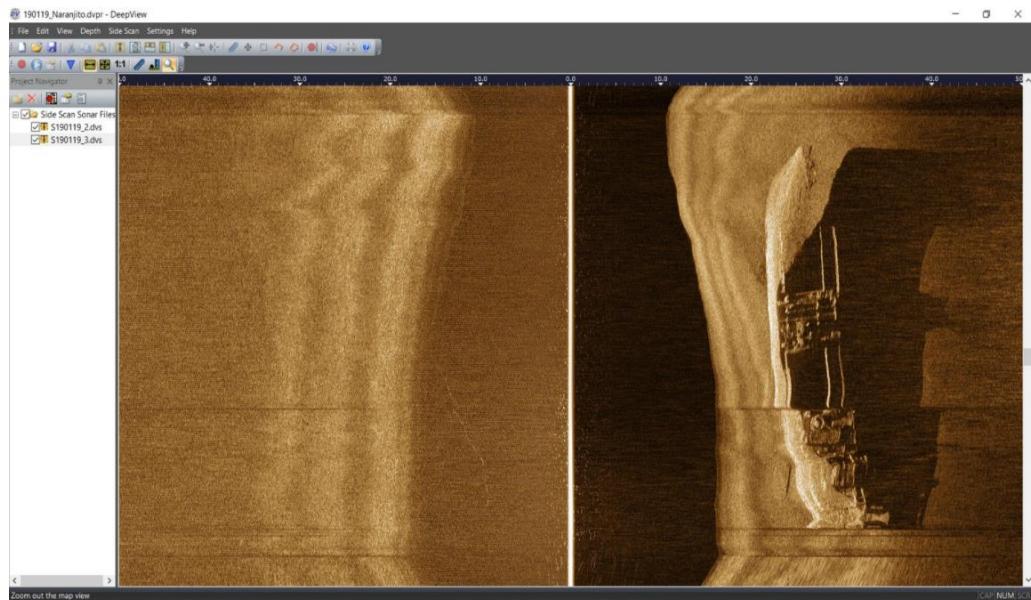


Figura 33. Visión de datos en DeepView.

3.3.6. Interfaz WaterLinked

En esta web podemos observar la información que nos proporciona la maleta de superficie (“WaterLinked”). Para llevar a cabo este apartado, primero hablaremos de cómo conectar la maleta, luego comentaremos toda la información que podemos extraer de este dispositivo y, finalmente, se realizarán algunas recomendaciones a la hora de colocar las sondas.

Desde el navegador, ponemos la IP: 192.168.2.94 y accedemos a la interfaz del sistema WaterLinked. Obtenremos la pantalla de inicio, donde se muestra el mapa. En la barra lateral tenemos varias opciones, que nos llevan a distintas pantallas:

- En Position se muestra el mapa del satélite. En Receivers se muestra la situación del ROV con respecto a las sondas (siguiente apartado). En Settings se muestra la pantalla de configuración. En Diagnostic se puede ver los datos de los receptores, la IMU y el GPS. En About tenemos la información del sistema, como la versión y también el número de serie.
- En la barra superior, tenemos el estado de calibración de la orientación del sistema (IMU) y el estado de la señal GPS, mostrándose con una brújula.
- En el apartado de Settings podemos configurar el tipo de localizador (locator S1 en nuestro caso), la señal GPS con la que se va a trabajar, orientación y unidades.
- En la pestaña de "Diagnostic" podemos ver si las sondas están trabajando correctamente o no. Con el ROV fuera del agua, en la gráfica se mostrará solo ruido.

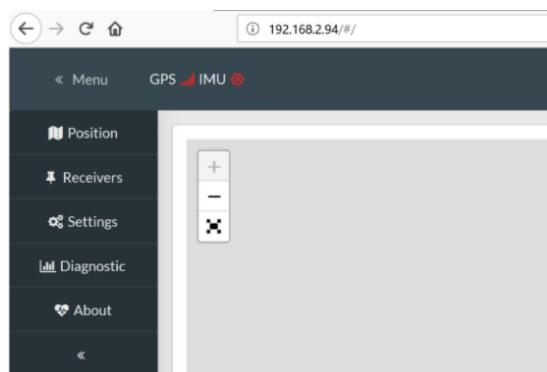


Figura 34. Interfaz WaterLinked.

Para configurar las sondas, se debe medir la distancia y profundidad que éstas tienen con respecto a la unidad superior. Además hay que tener en cuenta el número que se le asigna, para conectarlo en la maleta a dicho número.

3.3.7. Oculus ViewPoint

ViewPoint es una aplicación de software que se utiliza para ver, grabar y reproducir imágenes de la sonda Oculus, en además de proporcionar herramientas que ayudan a configurar el entorno y la red.

Cuando se establece una conexión con Oculus, ViewPoint mostrará la imagen de la sonda en vivo en el centro de la pantalla con controles e información de estado dispuestos en las esquinas de la ventana:

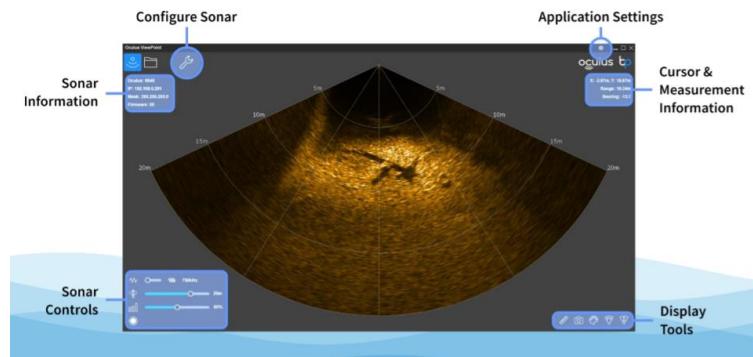


Figura 35. Interfaz WaterLinked.

El brillo de los objetivos y las texturas en la pantalla representan cuánto sonido se ha reflejado en ellos (con materiales gaseosos o densos reflejan más que con barro, arena o limo), mientras que las sombras acústicas se pueden proyectar en función de la geometría de los objetos o del lecho marino [9].

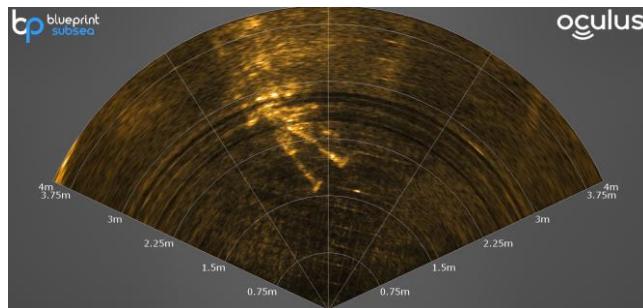


Figura 36. Visión de datos del sonar Oculus.

Las características y peculiaridades de su visión, en cuanto a la visibilidad del objetivo, sombras acústicas, localización de objetivos y patrones de haz acústicos, son similares que la del sonar360.

3.3.8. Calsens

El software CALSENS es un soporte para configurar los sensores, el menú de calibración, para seguir en tiempo real la medición de los parámetros seleccionados y para registrar los parámetros medidos.

El menú de medición de grabación da la posibilidad de seleccionar los parámetros a registrar, el valor promedio de la medición, la frecuencia de adquisición (a partir de 1 segundo) y también para tener en cuenta algunas compensaciones configuraciones (temperatura, presión, salinidad) [17].

3.4. PUESTA EN MARCHA

Para realizar la puesta en marcha, vamos a llevar a cabo una serie de pasos encontrados en la web del proveedor para que sea sencillo entender todo el conexionado y funcionamiento.

El conexionado consiste en conectar el usb y el cable de red de la caja de superficie al ordenador, así como el usb del joystick y la batería quitando manualmente la tapa del cilindro inferior. Conectar el cable amarillo (umbilical) desde la caja de superficie al lateral del soporte (bobina), tal y como vemos en la siguiente imagen:

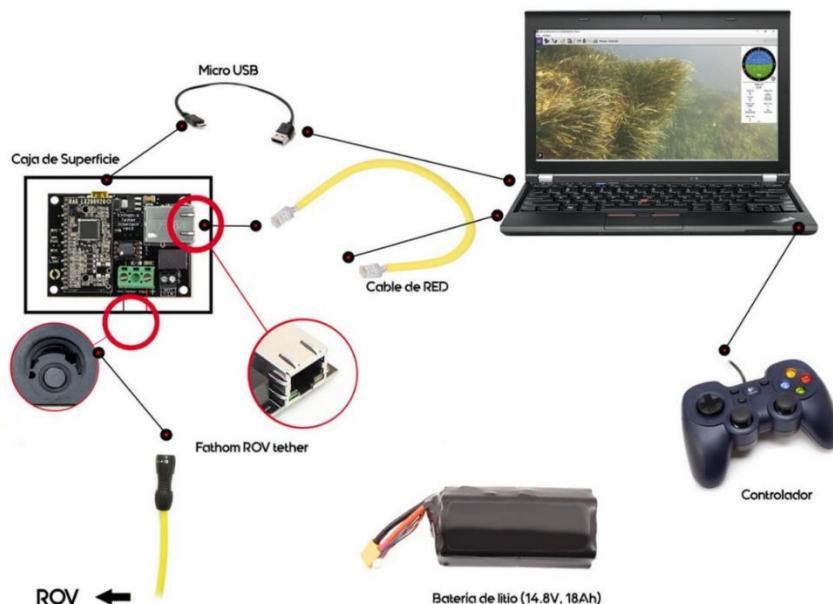


Figura 37. Conexionado para la puesta en marcha.

El diagrama de hardware del ROV se puede ver en la figura siguiente. Muestra como los diferentes sensores y actuadores se comunican entre sí y cómo se envían los datos a una estación de control del lado superior. Las flechas indican en qué dirección se dirige el flujo de datos y energía.

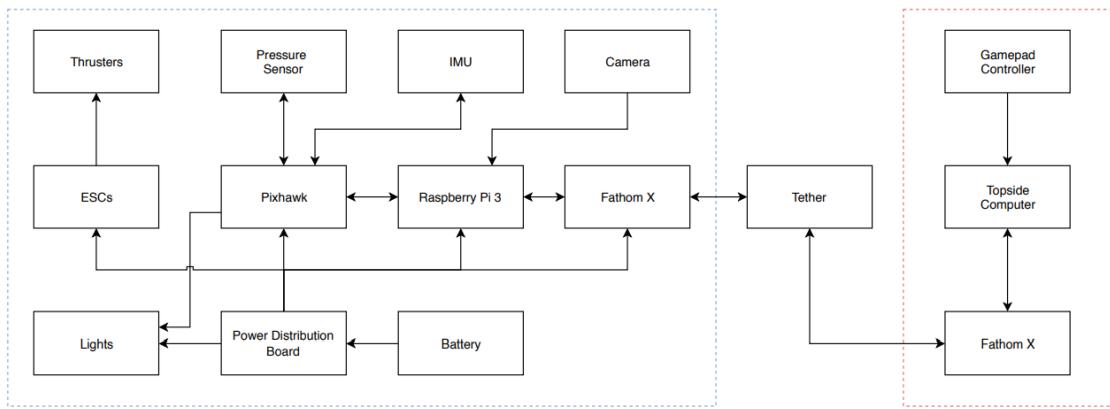


Figura 38. Diagrama de hardware.

Es necesario configurar la dirección IP fija de nuestro PC que realizará la función de panel de control: 192.168.2.1.

4. COMUNICACIÓN CON EL ROV

Como dijimos en la introducción, el nuevo objetivo tras la puesta en marcha sería obtener los datos en crudo de los sensores del Sibiu (Sonar 360 y Ecosonda, principalmente), la telemetría de la PixHawk y la localización proporcionada por el sistema WaterLink. Para ello será necesario saber qué protocolo de comunicación usan, cuál es su dirección IP, su puerto y el formato del mensaje que utilizan.

Una vez obtenida toda esta información y sepamos cómo funcionan los dispositivos, podremos crear un algoritmo a través de Matlab que se comunique con los elementos sin necesidad de usar su software propio. De esta forma, obtendremos sus datos y, con el propio Matlab, debemos representar y/o interpretar la información, con el fin de automatizar los movimientos del ROV en relación con una premisa decidida por el usuario.

4.1. COMPONENTES ELECTRÓNICOS

En primer lugar, vamos a detallar cuáles son los componentes electrónicos más importantes que se encuentran en el sistema ROV-Panel de control, comenzando por las placas de comunicación, siguiendo con la raspberry y concluyendo con la pixhawk.

La comunicación, como hemos comentado anteriormente, se realiza a través del cable umbilical, el cual está conectado desde la caja de superficie al ROV. Tanto en el interior de la caja de superficie como del ROV se encuentran las placas Fathom-X que proporcionan una robusta conexión Ethernet, de alta velocidad y larga distancia a través de un solo par de cables. Permite la transmisión de videos HD y datos de gran ancho de banda a más de 300 m de longitud de conexión.

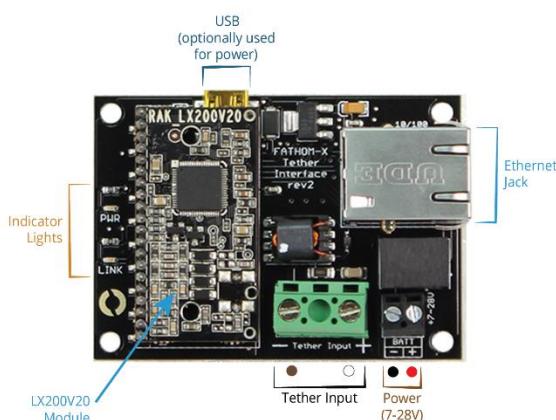


Figura 39. Fathom-X.

Una vez encendido, actúa como un puente Ethernet transparente para extender cualquier conexión Ethernet a través del tether.

La placa Fathom-X que se encuentra en el interior del ROV se conecta a la raspberry a través de ethernet. A su vez, la raspberry tiene conectado en sus puertos USB la cámara, la pixhawk y los sonares (sonar360 y ecosonda).

Por último, tenemos la pixhawk, que dispone de varios sensores integrados como pueden ser un acelerómetro, giróscopo, magnetómetro, barómetro,... Además, tiene conectado los actuadores del ROV: servomotor de la cámara, luces y los 8 propulsores.

Ante la necesidad de realizar pruebas donde nos conectaremos remotamente con la raspberry, decidimos comprar el paquete de electrónica avanzada del ROV para ArduSub proporcionado por BlueRobotics. De esta forma, podemos realizar las pruebas pertinentes con este sistema sin necesidad de usar el ROV. El paquete fue colocado sobre una base para facilitar el conexionado y tener una visión clara del conexiónado, obteniendo un sistema como el que vemos en la siguiente imagen:

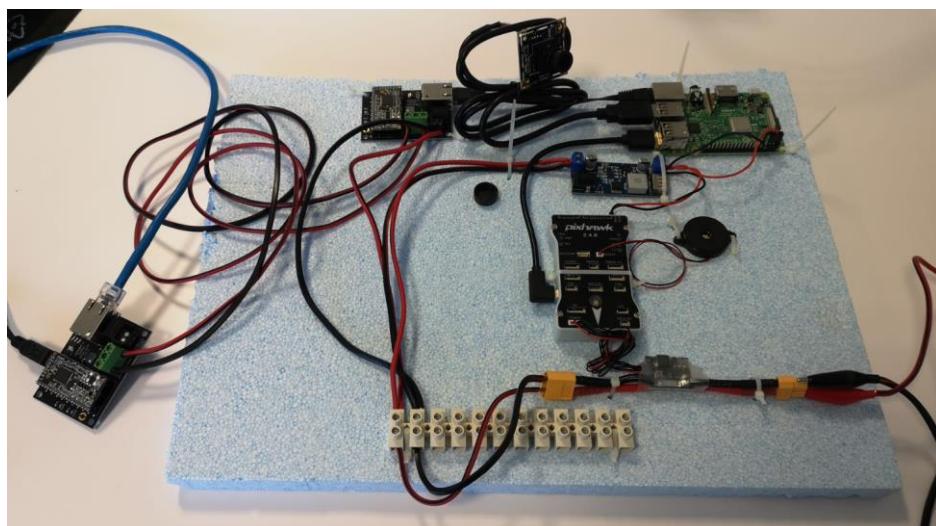


Figura 40. Sistema electrónico básico de un ROV.

4.2. ARQUITECTURA

A continuación, se va a comentar la arquitectura de comunicación entre nuestro ROV y el panel de control. El proceso comienza con el envío, por parte del joystick, de las acciones que queremos realizar, recibidas por el panel de control que, a través del protocolo correspondiente llega a enviar la información a la caja de superficie a través del cable ethernet.

A esta se le conecta el water linked y, a través del cable propio del ROV, se le conecta al Sibiu Pro. Esta información es recibida por la Raspberry que controla los circuitos electrónicos de los dispositivos, gracias a ArduSub, y envía la información al dispositivo correspondiente, ya sean la cámara, los sonares o la PixHawk, todo conectado por USB.

En la siguiente imagen podemos observar detalladamente cómo es la comunicación:

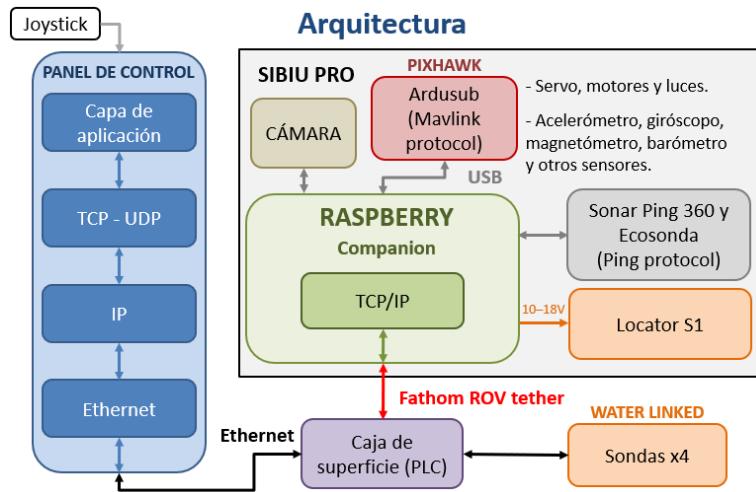


Figura 41. Diagrama comunicación.

Nuestro objetivo primordial consiste en realizar una serie de modificaciones a esta arquitectura para sustituir el mando por un algoritmo que desarrollemos para realizar automatismos que dependan de la misión que queramos realizar.

El usuario, mientras va recibiendo los datos de los sensores y del control de operación, pedirá realizar alguna misión, por ejemplo, seguir un banco de peces. Esta misión está formada por una serie de operaciones que se enviarán al controlador de operación para que, automáticamente, se realicen las tareas necesarias sobre los actuadores. Todo esto será posible gracias a la información que recibimos de los sensores iniciales y del sistema de referencia.

Podemos observar de forma gráfica cómo quedaría la arquitectura modificada:

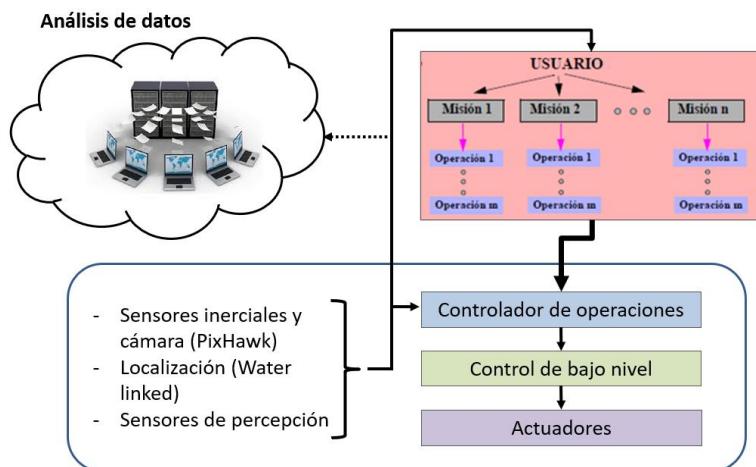


Figura 42. Arquitectura modificada.

4.3. OBTENCIÓN DE DATOS DE SENSORES

4.3.1. Sonares BlueRobotics: PingProtocol

El protocolo está diseñado para comunicaciones síncrona con formato maestro-esclavo. El esclavo tan solo enviará el dato cuando el maestro se lo solicite. Esto permite que el protocolo sea usado como un bus half-duplex como es el RS485. Algunos dispositivos también otorgan facilidades para proporcionar comunicación asíncrona.

PingProtocol utiliza un formato de mensaje fijo, a través del cual, podemos hacer una solicitud al ROV de la información del sonar360 y el ecosonda con el ID del mensaje correspondiente enviando el mensaje a la IP de la raspberry del ROV y al puerto adecuado. El puerto pertinente al ecosonda es el 9090, mientras que el sonar360 usa el puerto 9092.

Cada mensaje está formado por un encabezado, una carga de datos y la verificación por suma. El formato de mensaje es el siguiente [18]:

Byte	Type	Name	Description
0	u8	start1	Start frame identifier, ASCII 'B'
1	u8	start2	Start frame identifier, ASCII 'R'
2-3	u16	payload_length	Number of bytes in payload.
4-5	u16	message_id	The message id.
6	u8	src_device_id	The device ID of the device sending the message.
7	u8	dst_device_id	The device ID of the intended recipient of the message.
8-n	u8[]	payload	The message payload.
(n+1)-(n+2)	u16	checksum	The message checksum. The checksum is calculated as the sum of all the non-checksum bytes in the message.

Figura 43. Formato mensaje PingProtocol.

Los mensajes PingProtocol están divididos en 3 categorías: comunes, del PingSonar y del Sonar360. Hay una gran variedad de mensajes para cada dispositivo, pero para la obtención de los datos solo vamos a necesitar los que vemos en este apartado.

A su vez, los mensajes están divididos en 4 categorías:

- General: de propósito general.
- Read/get: para responder a una solicitud de mensaje general del maestro. Con estos mensajes leeremos datos del dispositivo.
- Write/set: para configurar algunos parámetros del dispositivo. Con estos mensajes escribiremos datos para el dispositivo.

- Control: para realizar alguna acción más compleja.

Para solicitar los datos, en primer lugar, debemos colocar en los bytes 4-5 del formato de mensaje el número 6, que hace referencia a una solicitud general:

Type	Name	Description	Units
u16	requested_id	Message ID to be requested.	

Figura 44. Formato mensaje solicitud general.

A continuación, debemos poner en la carga de datos (payload) el número del mensaje que corresponda a la información que queremos obtener, esto variará en función del dispositivo.

- Ecosonda: cargaremos en el payload el mensaje 1211, que nos permitirá obtener la distancia con respecto al suelo y la seguridad de ese dato dependiendo de la irregularidad del suelo.

Type	Name	Description	Units
u32	distance	Distance to the target.	mm
u8	confidence	Confidence in the distance measurement.	%

Figura 45. Formato mensaje ecosonda.

- Sonar 360: en el sonar360 no se realiza una solicitud de información con el mensaje común 6, sino que, al tener que decidir el ángulo de la cabeza del rotor, usaremos un mensaje de tipo write/set. Por ello, tenemos que enviarle el mensaje 2601 con la posición concreta para el ángulo deseado, así como otros variables que nos permitirán modificar la distancia y la ganancia de las señales recibidas.

Type	Name	Description	Units
u8	mode	Operating mode (1 for Ping360)	
u8	gain_setting	Analog gain setting (0 = low, 1 = normal, 2 = high)	
u16	angle	Head angle	gradian
u16	transmit_duration	Acoustic transmission duration (1~1000 microseconds)	microsecond
u16	sample_period	Time interval between individual signal intensity samples in 25nsec increments (80 to 40000 == 2 microseconds to 1000 microseconds)	
u16	transmit_frequency	Acoustic operating frequency. Frequency range is 500kHz to 1000kHz, however it is only practical to use say 650kHz to 850kHz due to the narrow bandwidth of the acoustic receiver.	kHz
u16	number_of_samples	Number of samples per reflected signal	
u8	transmit	0 = do not transmit; 1 = transmit after the transducer has reached the specified angle	
u8	reserved	reserved	

Figura 46. Formato mensaje Sonar Ping360.

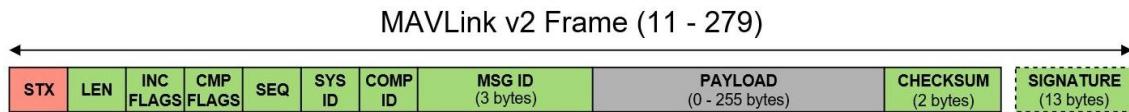
4.3.2. Pixhawk

Para comunicarnos con este dispositivo utilizaremos Mavlink, que se trata de un protocolo de mensajería muy ligero para comunicarse con drones (y entre componentes de drones a bordo).

MAVLink sigue un moderno patrón híbrido de publicación-suscripción y diseño punto a punto: los flujos de datos se envían/publican como topics, mientras que los subprotocolos de configuración, como el protocolo de misión o el protocolo de parámetros, son punto a punto con retransmisión.

Los mensajes se definen dentro de archivos XML. Cada archivo XML define el conjunto de mensajes admitido por un sistema MAVLink en particular, también denominado "dialecto". El conjunto de mensajes de referencia que implementan la mayoría de las estaciones de control en tierra y los pilotos automáticos se define en common.xml (la mayoría de los dialectos se basan en esta definición).

A continuación, se muestra el formato del paquete de mensajería del protocolo MAVLink v2:



Byte Index	C version	Content	Value	Explanation
0	uint8_t magic	Packet start marker	0xFD	Protocol-specific start-of-text (STX) marker used to indicate the beginning of a new packet. Any system that does not understand protocol version will skip the packet.
1	uint8_t len	Payload length	0 - 255	Indicates length of the following payload section. This may be affected by payload truncation.
2	uint8_t incompat_flags	Incompatibility Flags		Flags that must be understood for MAVLink compatibility (implementation discards packet if it does not understand flag).
3	uint8_t compat_flags	Compatibility Flags		Flags that can be ignored if not understood (implementation can still handle packet even if it does not understand flag).
4	uint8_t seq	Packet sequence number	0 - 255	Used to detect packet loss. Components increment value for each message sent.
5	uint8_t sysid	System ID (sender)	1 - 255	ID of system (vehicle) sending the message. Used to differentiate systems on network. Note that the broadcast address 0 may not be used in this field as it is an invalid source address.
6	uint8_t compid	Component ID (sender)	1 - 255	ID of component sending the message. Used to differentiate components in a system (e.g. autopilot and a camera). Use appropriate values in MAV_COMPONENT. Note that the broadcast address MAV_COMP_ID_ALL may not be used in this field as it is an invalid source address.
7 to 9	uint32_t msgid:24	Message ID (low, middle, high bytes)	0 - 16777215	ID of message type in payload. Used to decode data back into message object.
For n-byte payload: n=0 : NA, n=1 : 10, n>2 : 10 to (9+n)	uint8_t payload[max 255]	Payload		Message data. Depends on message type (i.e. Message ID) and contents.
(n+10) to (n+11)	uint16_t checksum	Checksum (low byte, high byte)		CRC-16/MCRF4XX for message (excluding magic byte). Includes CRC_EXTRA byte.

Figura 47. Formato mensaje MAVLink.

Los mensajes incluidos en el dialecto common.xml se pueden clasificar en 3 categorías:

- **Comandos:** en ellos se definen los valores de 7 parámetros, los cuales hacen referencia a diferentes magnitudes dependiendo del ID del comando. Para ArduSub, estos mensajes no están reconocidos aún y, por tanto, no están habilitados.
- **Mensajes:** el estado general del sistema. Si el sistema sigue el estándar MAVLink, el estado del sistema se define principalmente por tres modos: MANUAL (sistema bajo

control RC), HOLD DEPTH (modo para mantener la profundidad) y STABILIZE (permite movimientos más suaves), el resto de los modos automáticos no están disponibles para drones submarinos.

- Enumeraciones: son valores que definen la configuración del ROV.

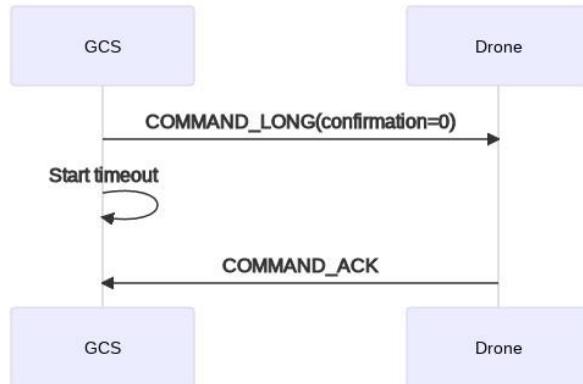


Figura 48. Diagrama de comunicación MavLink.

Dentro de los mensajes de MAVLink, cabe destacar el mensaje Heartbeat (o latido), el protocolo de Heartbeat se utiliza para anunciar la existencia de un sistema en la red MAVLink, junto con su identificación de sistema y componente, tipo de vehículo, stack de vuelo, tipo de componente y modo de vuelo.

El Heartbeat permite que otros componentes puedan descubrir sistemas que están conectados a la red e inferir cuándo se han desconectado. Se considera que un componente está conectado a la red si su mensaje HEARTBEAT se recibe con regularidad y se desconecta si no se reciben varios mensajes esperados.

Los componentes deben transmitir regularmente su HEARTBEAT y monitorear los HEARTBEAT de otros componentes/sistemas.

La velocidad a la que se debe transmitir el mensaje HEARTBEAT y cuántos mensajes pueden "perderse" antes de que se considere que un sistema ha agotado el tiempo de espera/desconectado de la red, depende del canal. Los componentes suelen publicar su latido a 1 Hz y consideran que otro sistema se ha desconectado si no se reciben cuatro o cinco mensajes.

Un componente puede optar por no enviar o transmitir información en un canal (que no sea HEARTBEAT) si no detecta otro sistema, y continuará enviando mensajes a un sistema mientras recibe latidos. Por tanto, es importante que los sistemas:

- Transmite un latido incluso cuando no esté al mando del sistema remoto.
- No difunda un latido cuando estén en un estado de falla (es decir, no publique un latido de un hilo separado que desconozca el estado del resto del componente) [19].

4.4. DIRECCIONES Y PUERTOS

Como ya sabemos, tanto la dirección IP de la raspberry como del panel de control han de estar fijadas para establecer la comunicación. A continuación, se muestra una imagen con las direcciones y puertos de cada dispositivo, así como la información que transmiten, protocolo que usan, programas que corren y softwares donde se observan los datos.

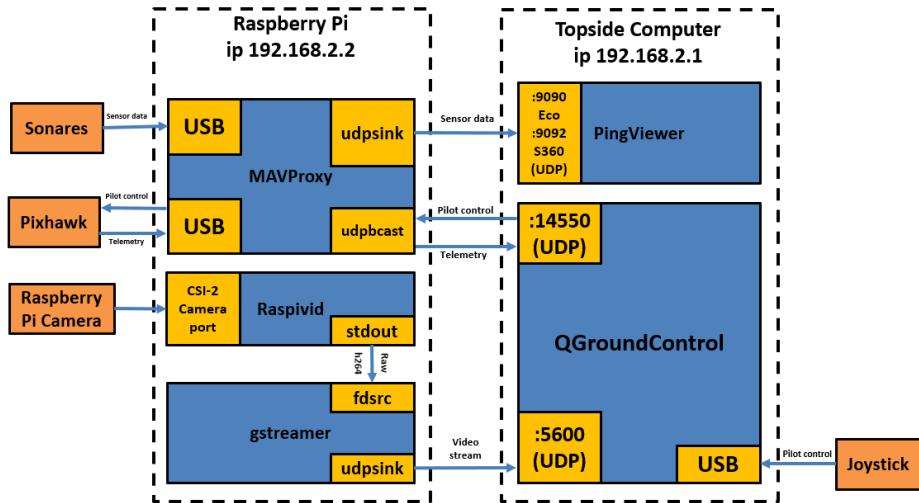


Figura 49. Direcciones IP y puertos.

5. ALGORITMOS EN MATLAB

5.1. DATOS SONARES

Para comunicarnos con los sonares a través de Matlab, es necesario abrir un puerto UDP por dónde enviaremos y recibiremos los mensajes del sonar deseado. Para ello, formaremos un vector con los valores correspondientes a los valores que vimos en el formato del mensaje PingProtocol que queramos enviar, transmitiéndolos a la IP de la raspberry y al puerto correspondiente de acuerdo con la figura siguiente. Tras recibir los datos, debemos almacenarlos y se representarán de un modo u otro dependiendo del dispositivo del que se trata.

El diagrama de flujo que representa los algoritmos de ambos sonares se corresponde con el siguiente:

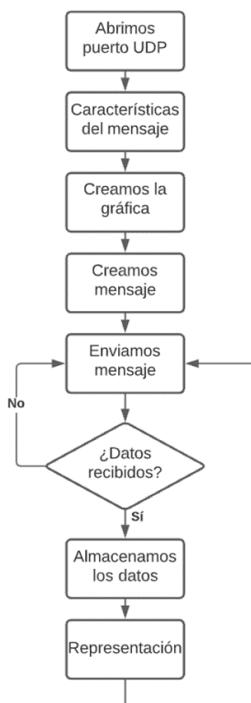


Figura 50. Diagrama de flujo para obtener datos de los sonares.

5.1.1. Representación sonar Ping 360

Esta representación se ha realizado buscando algo similar a lo que hace su propio software PingViewer, buscando una representación polar de los datos. En la primera vuelta se crean todos los puntos necesarios (1200 puntos para cada ángulo y 400 ángulos diferentes) y, a partir de aquí, tan solo es necesario cambiar al color correspondiente dependiendo de la intensidad del rebote, evitando la saturación al no crear más puntos.

Es posible aprovechar espectros de colores ya creados. Pero para aprovechar al máximo la diferencia de color entre intensidades, se ha escogido un tramo de un espectro similar al de PingViewer.

Además de este proceso, se han realizado algoritmos que permiten observar un sector deseado, pudiendo parar y reanudar la toma de datos de forma interactiva, obteniendo los siguientes resultados.

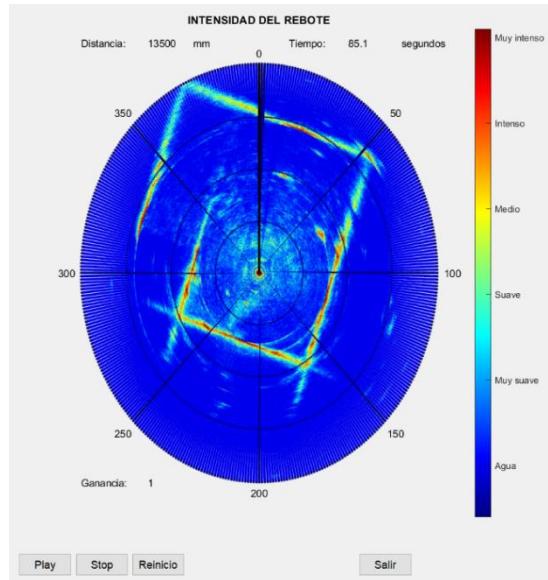


Figura 51. Representación datos Ping360.

5.1.2. Representación Ping sonar ecosonda

La representación de los datos del ecosonda es más sencilla, pues solo tenemos los datos de la profundidad medida y de la confianza del mismo. Así que dibujaremos dos gráficas de la siguiente manera:

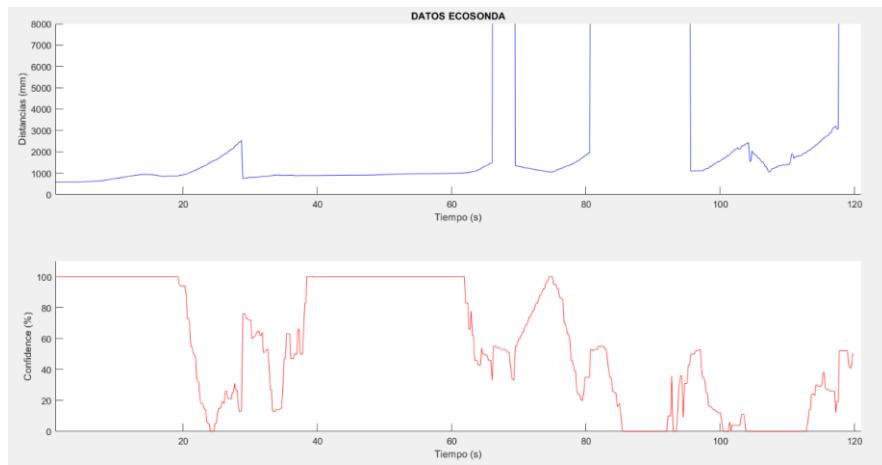


Figura 52. Representación datos PingSonar ecosonda.

5.2. CONTROL MANUAL

Los clientes MAVLink intercambian información dentro de la red utilizando estructuras de datos comúnmente definidas como mensajes. El protocolo de parámetros MAVLink se utiliza para intercambiar los ajustes de configuración entre ROV y la estación de control de tierra (GCS). Por ejemplo, GCS inicia una solicitud en forma de mensajes y el UAV responde con datos. A continuación, se va a explicar cómo se ha logrado realizar el control manual a través de Matlab, al igual que se realiza con QGroundControl, a través del joystick.

5.2.1. Obtención del puerto del ROV

En primer lugar, es necesario conocer a qué dirección y a qué puerto hay que enviar el mensaje desde la GCS hasta el ROV. La dirección IP de destino es conocida, ya que se sabe que la raspberry trabaja con la dirección fija 192.168.2.2. En cambio, cada vez que alimentamos el ROV, el puerto de la raspberry se genera de forma aleatoria y empieza a enviar mensajes UDP en broadcast (por todas las direcciones de la red). Para conocer ese puerto y cómo es el protocolo de comunicación más detalladamente, ha sido de gran ayuda el uso de Wireshark, que es un analizador de protocolos de red que le permite capturar datos de paquetes de una red en vivo.

Para conocer el puerto de forma automática, se procedió a realizar una captura de un paquete con la misma filosofía que Wireshark, pero esta vez a través de Matlab. Para ello, se hace uso de una función realizada por usuarios en la que puedes detallar el filtro y la cantidad de paquetes que quieras obtener. En nuestro caso, conocemos que se utiliza el protocolo UDP, que es a través de ethernet y que el puerto de destino es el 14550. Con esas condiciones, capturamos un paquete y extraemos el puerto de donde se ha enviado, obteniendo automáticamente el puerto aleatorio que crea la electrónica del ROV.

5.2.2. Creación GCS y suscripción al ROV

Antes de empezar con la comunicación con el ROV, en primer lugar, hay que detallar el dialecto que vamos a usar (common.xml) y crear el objeto del nodo de la estación de tierra (SystemID=255, ComponentID=190 y componente GCS). A continuación, conectamos ese objeto con el puerto adecuado (14550) y creamos un objeto cliente que se suscriba al nodo de la estación de tierra.

5.2.3. Almacenamiento de datos

Una vez tenemos los objetos necesarios creados, se procede a detallar qué información se quiere recibir del ROV. Estos ROVs están continuamente enviando mensajes MAVLink a través del puerto, que pueden ser obtenidos. Dentro de estos mensajes caben destacar los siguientes [19]:

ATTITUDE

ATTITUDE (#30)

[Message] The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right).

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

Figura 53. Mensaje Attitude.

GLOBAL POSITION INT

GLOBAL_POSITION_INT (#33)

[Message] The filtered global position (e.g. fused GPS and accelerometers). The position is in GPS-frame (right-handed, Z-up). It is designed as scaled integer message since the resolution of float is not sufficient.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
lat	int32_t	degE7	Latitude, expressed
lon	int32_t	degE7	Longitude, expressed
alt	int32_t	mm	Altitude (MSL). Note that virtually all GPS modules provide both WGS84 and MSL.
relative_alt	int32_t	mm	Altitude above ground
vx	int16_t	cm/s	Ground X Speed (Latitude, positive north)
vy	int16_t	cm/s	Ground Y Speed (Longitude, positive east)
vz	int16_t	cm/s	Ground Z Speed (Altitude, positive down)
hdg	uint16_t	cdeg	Vehicle heading (yaw angle), 0.0..359.99 degrees. If unknown, set to: UINT16_MAX

Figura 54. Mensaje Global_position_INT.

Para obtener los datos de estos mensajes, se crea un objeto para recibir mensajes y procesarlos mediante mavlinksub. Este suscriptor recibe mensajes sobre el tipo de mensaje deseado y busca específicamente los que coinciden con el sistema y el ID de componente.

5.2.4. Creación heartbeat

Como vimos anteriormente, el Heartbeat ha de enviarse continuamente con una frecuencia específica. Para lograrlo a través de Matlab, en primer lugar, se ha creado el mensaje con el dialecto concreto y se han modificado sus parámetros para que coincidan con los del ROV, gracias a que han podido ser determinados con la ayuda de Wireshark. Una vez teníamos el mensaje creado, el siguiente objetivo consiste en crear un timer que permitiera enviarlo a través de la configuración con frecuencia concreta. Cuando se pone en marcha el timer del Heartbeat, ya estamos totalmente conectados con el ROV y estamos recibiendo sus mensajes en la dirección fija.

Esta estrategia con los timer también ha sido utilizada para obtener los mensajes que queremos almacenar del apartado anterior. Cada cierto tiempo (especificado por el usuario en la creación del timer), se llama a una función que permite que se recibe la información de los mensajes ATTITUDE y GLOBAL_POSITION_INT, almacenándolos en un ‘array struct’ (tipo de datos que agrupa datos relacionados mediante contenedores de datos denominados campos).

5.2.5. Control

Para el control, se ha realizado un algoritmo en el cual se crea el mensaje por defecto de control manual cuando no se está pulsando ningún botón, este mensaje está formado por 6 parámetros (target, X, Y, Z, r y buttons). El parámetro target se mantiene en 1 pues se trata del ID del ROV. Los parámetros X, Y, Z y r hacen referencia al cabeceo, alabeo, empuje y guiñada, respectivamente. El comando buttons consiste en un campo de bits donde cada bit se corresponde con uno de los botones que pueden ser pulsados [19].

MANUAL_CONTROL (#69)

[Message] This message provides an API for manually controlling the vehicle using standard joystick axes nomenclature, along with a joystick-like input device. Unused axes can be disabled and buttons are also transmit as boolean values of their

Field Name	Type	Description
target	uint8_t	The system to be controlled
x	int16_t	X-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to forward(1000)-backward(-1000) movement on a joystick and the pitch of a vehicle.
y	int16_t	Y-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to left(-1000)-right(1000) movement on a joystick and the roll of a vehicle.
z	int16_t	Z-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to a separate slider movement with maximum being 1000 and minimum being -1000 on a joystick and the thrust of a vehicle. Positive values are positive thrust, negative values are negative thrust.
r	int16_t	R-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to a twisting of the joystick, with counter-clockwise being 1000 and clockwise being -1000, and the yaw of a vehicle.
buttons	uint16_t	A bitfield corresponding to the joystick buttons' current state, 1 for pressed, 0 for released. The lowest bit corresponds to Button 1.

Figura 55. Mensaje Manual_control.

En el algoritmo desarrollado, primero creamos el objeto del mando que nos permite conocer los botones que están siendo pulsados y luego entramos en el bucle.

En él, se realiza una lectura de los botones que están siendo pulsados, diferenciando entre los botones (buttons), la cruceta (povs) y los joysticks (axes). Para cada caso, se realizan las acciones necesarias ya sea aumentando el valor de la variable buttons, calculando el movimiento de los propulsores con los joysticks o realizando las acciones determinadas con la cruceta. Una vez se ha concretado el mensaje final con las acciones que, a través del mando, se ha decidido realizar, mandamos el mensaje y volvemos a iniciar el bucle.

5.3. POSICIONAMIENTO

En este apartado, donde es necesario el uso del sistema de localización WaterLinked para conocer la posición del ROV y de la maleta de superficie, vamos a explicar los algoritmos que

se han creado para representar tanto la posición del ROV para realizar trayectorias, como los datos filtrados del Ping360.

5.3.1. Transformación de sistemas de coordenadas

Por un lado, para obtener la matriz de transformación del sistema local (WaterLinked) al global (visión satélite), se utiliza la señal GPS o directamente se usan las coordenadas donde se va a colocar la maleta de superficie (introduciendo manualmente la posición estática en latitud y longitud).

Una vez se conoce la posición de la maleta de superficie, a través del calibrado del sistema de receptores se obtienen las señales emitidas por el localizador que se encuentra en el ROV. Esto, junto con los datos de la orientación del ROV debido a la IMU que incorpora la controladora de vuelo, es posible conseguir una mayor precisión del posicionamiento a través de la combinación de dichos datos. Combinando esta información, es posible obtener la matriz de transformación que nos permite pasar del sistema de referencia del ROV al local.

A través de las matrices anteriores, el sistema de posicionamiento proporciona el dato sobre la posición del ROV en el sistema global (latitud y longitud) con una gran precisión.

$${}^G T_R = {}^G T_L \cdot {}^L T_R$$

Este dato se puede obtener al abrir una comunicación con el ROV a través del protocolo MavLink, a través del cual podemos solicitar toda la información disponible. En la siguiente figura se puede observar el funcionamiento de este sistema de forma gráfica:

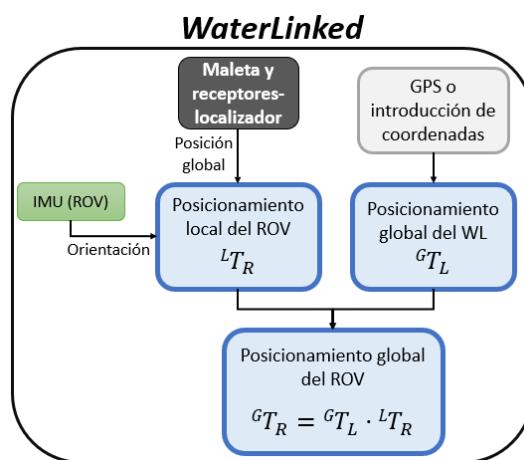


Figura 56. Algoritmo sistema WaterLinked.

Por otro lado, el sonar va a poder ofrecernos la información del entorno del ROV que, tras un procesado de los datos, permitirá obtener la posición de los objetivos que se deseen (ángulo y distancia) con respecto a la posición del ROV. Esto conlleva obtener la matriz de

transformación desde los objetivos al vehículo, lo cual permite, a su vez, conocer la matriz de los objetivos en el sistema global que estamos buscando.

$${}^G T_{Ob} = {}^G T_R \cdot {}^R T_{Ob}$$

Para hacer uso de los datos a través de Matlab, se realizó un tratamiento de los mismos de forma que se almacenaran tan solo los datos más relevantes. Para ello, una vez están almacenados todos los datos, se llama a una función dentro del algoritmo que consiste en:

- Eliminación de datos inferiores a un límite, despreciando así todos los datos que hacen referencia a rebotes con una intensidad menor, como la del agua.
- Eliminación de datos redundantes. Debido a la gran cantidad de datos que proporciona el sensor, cuando estamos detectando un obstáculo hay bastantes datos que se deben al mismo objeto. Para solucionarlo, una vez detectamos un obstáculo, colocamos un punto y obviamos los datos que pertenecen al mismo, hasta que volvamos a ver rebotes inferiores.
- Eliminación de ruido (datos cercanos a la posición del ROV).
- Identificación de objetivos (clustering).

Por último, como es posible saber la posición de la maleta de superficie y, por tanto, se conoce la matriz para pasar del sistema de referencia global al local, se realizará la comprobación de los datos obtenidos representándolos en el sistema de referencia local en coordenadas X,Y.

$${}^L T_{Ob} = {}^L T_R \cdot {}^R T_{Ob}$$

Este posicionamiento de objetivos se resume en la siguiente figura:

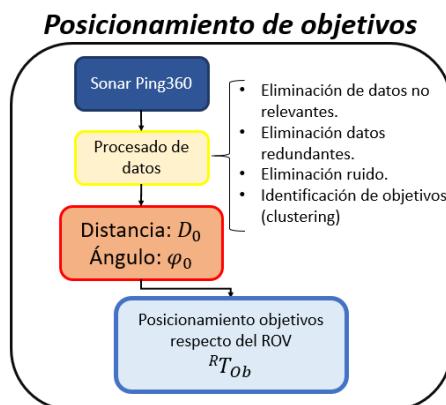


Figura 57. Algoritmo posicionamiento de objetivos.

En este apartado, es esencial la precisión de los datos telemétricos del dispositivo encargado como controlador de vuelo. Los datos a tener en cuenta para detectar el objetivo es el ángulo (heading) en el que se encuentra el obstáculo, el heading actual y la distancia entre el ROV y el objetivo.

Una vez conocida esa información, el ROV podría realizar un movimiento automático de cambio de rumbo y acercarse al objetivo deseado de forma autónoma, a través del cálculo de

la velocidad de avance y velocidad de giro que sea necesario mediante un control proporcional, ver siguiente figura:

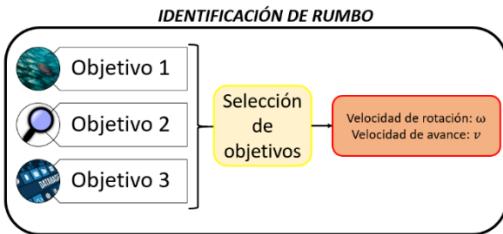


Figura 58. Algoritmo identificación de rumbo.

5.3.2. Representación datos filtrados Ping360

Para realizar la representación una vez tenemos los resultados filtrados y las matrices de transformaciones necesarias, es conveniente crear dos sistemas gráficos: uno donde marquemos la posición del WaterLinked como origen de coordenadas (local) y dibujemos el perímetro de la piscina donde estemos trabajando, y otro donde dibujemos los datos en la visión satélite que proporciona el software (global).

En la prueba que vemos a continuación, llevamos a cabo la localización de objetivos con el ROV y los obstáculos en posiciones estáticas y conocidas, donde comprobaremos las diferencias entre la información del sonar y las medidas realizadas. La disposición del ROV, la maleta de superficie del WaterLinked, sus receptores y los obstáculos en la piscina están distribuidos de acuerdo con la siguiente figura:

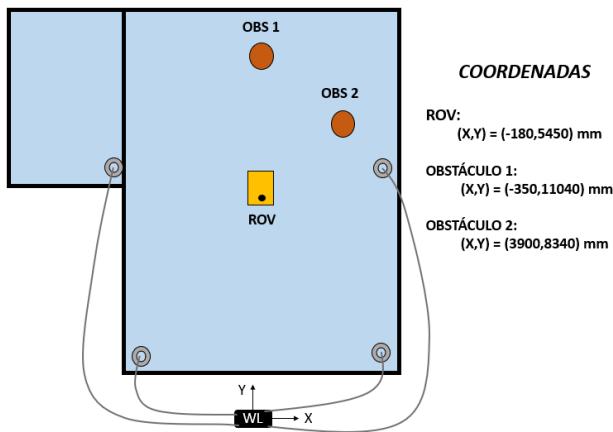


Figura 59. Distribución de objetos en la piscina.

En esa posición y con el ROV parado, comenzamos a almacenar información de una vuelta completa del sonar. En la representación local que vemos a continuación, se han mostrado las coordenadas medidas del perímetro de la piscina (línea roja), los obstáculos y el maletín de superficie (cuadrados negros), mientras que los datos ya tratados del sonar han sido especificados con asteriscos:

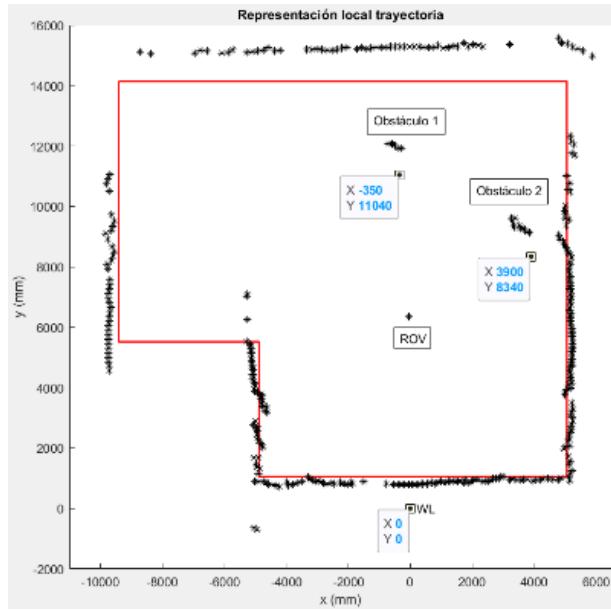


Figura 60. Representación local de datos del Ping360.

Por otro lado, en la visión satélite se ha marcado de color verde los datos del sonar y de color negro las posiciones medidas de los obstáculos:



Figura 61. Representación global de datos del Ping360.

Se puede comprobar cómo los obstáculos y los límites de la piscina son muy parecidos a los que hemos medido. El error que existe se debe a que la velocidad del agua está fijada en 1500m/s, pero esta puede ser modificada dependiendo de las características de la misma hasta que los datos coincidan con mayor precisión.

5.3.3. Control con trayectoria

Como ya hemos dicho anteriormente, gracias al WaterLinked, podemos conocer la posición del ROV. Por tanto, es posible representar su posición mientras realizamos el control manual del apartado anterior.

Este algoritmo funciona de forma similar al control manual, excepto por el hecho de que cada vez que obtenemos los datos que queremos representar, los mostramos en la gráfica local y global.

En la prueba que vemos a continuación, hemos enumerado las esquinas de la piscina y se ha realizado una conducción por ellas de forma consecutiva para representar los datos:

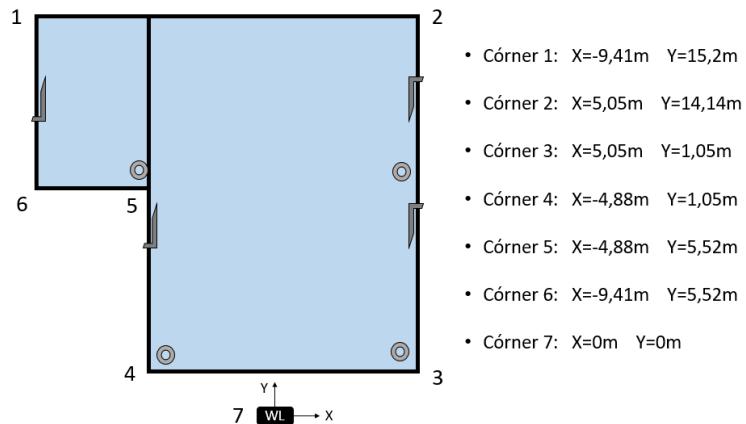


Figura 62. Coordenadas locales de las esquinas de la piscina.

- Prueba 1: Trayectoria 2-3-4.

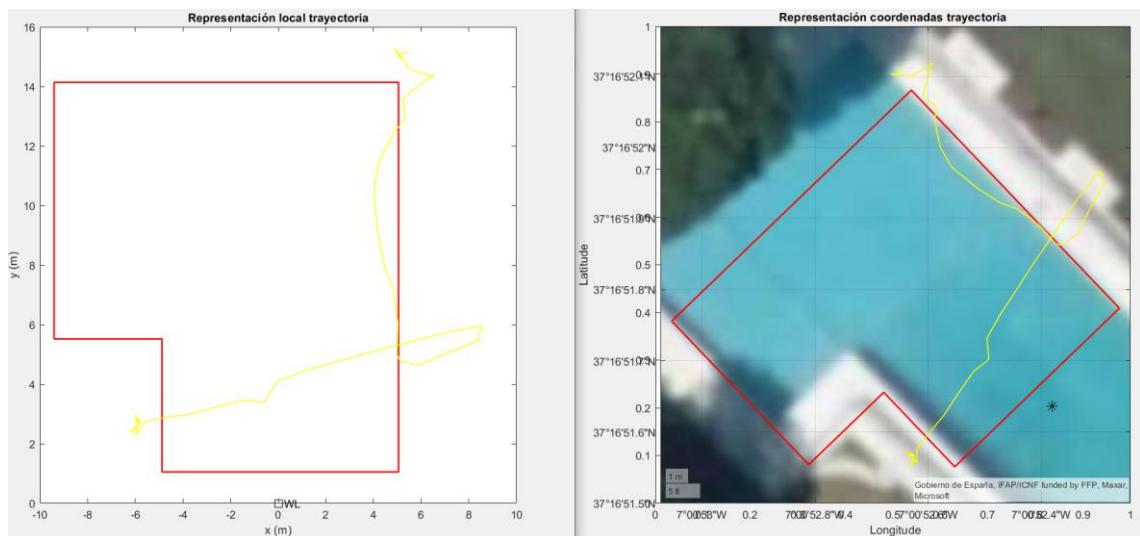


Figura 63. Trayectoria prueba 1.

- Prueba 2: Trayectoria 2-3-4-5-2.

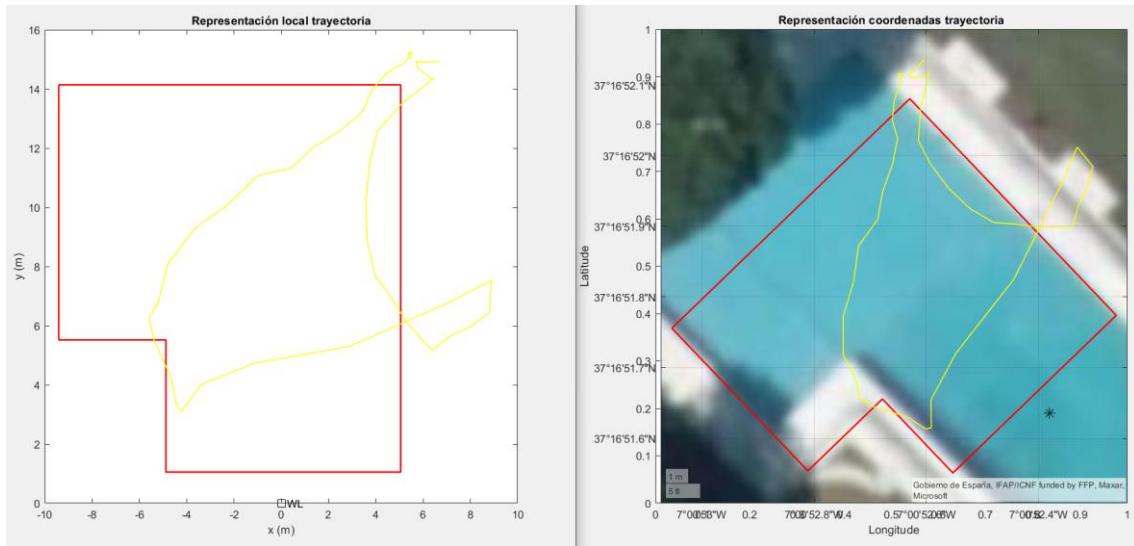


Figura 64. Trayectoria prueba 2.

- Prueba 3: Trayectoria 1-5-4-3-2-1.

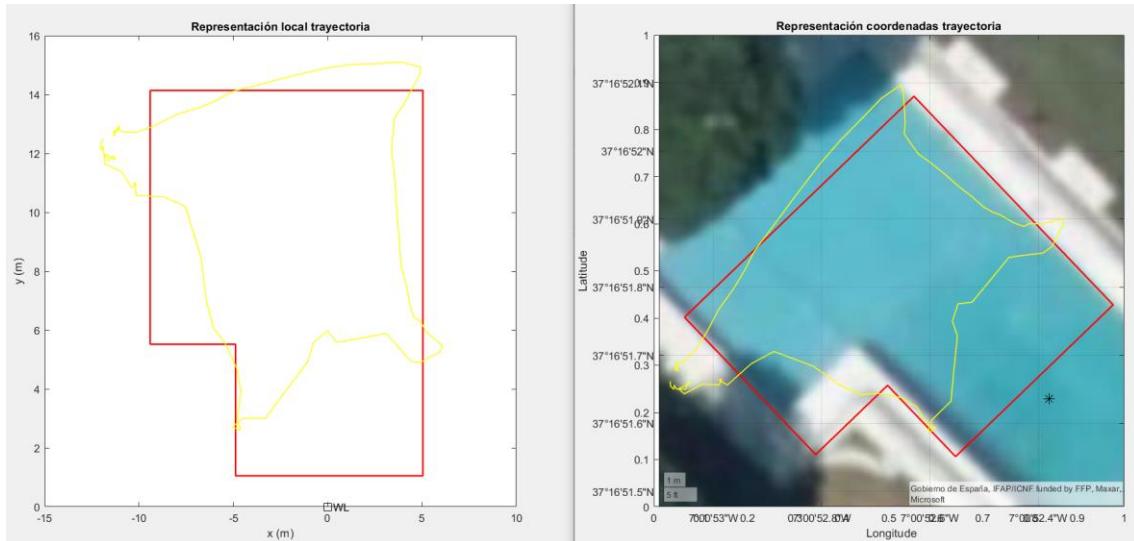


Figura 65. Trayectoria prueba 3.

En estos resultados es posible observar cómo la precisión de la ubicación del ROV no es adecuada para realizar la experimentación que estamos buscando. Esto se debe a que la precisión viene dada por la distancia del localizador a los receptores y puede verse afectada por los reflejos acústicos. En un tanque pequeño, el entorno acústico tiene muchos reflejos que afectan a la precisión. Se puede esperar una precisión de 5 a 10 cm cuando se configura correctamente, pero para ello hay que tener en cuenta las siguientes cuestiones:

- La mejor configuración en un tanque pequeño es extender los receptores en forma de L en la esquina del tanque y establecer el rango de búsqueda un poco más grande que el tamaño del tanque.
- Si desea reducir los reflejos acústicos (trayectos múltiples) en el tanque, se puede experimentar introduciendo algún tipo de material humectante en el agua. La mayoría de los materiales con superficies blandas suelen ser buenos.
- La transmisión acústica puede verse afectada por estos escenarios:
 - o Reflejos de superficies duras cercanas a su área de operación (cascos de barcos, fondo marino, construcciones, etc.)
 - o Amortiguación de señales de objetos en el agua (grandes consentimientos de peces, etc.).

Al trabajar en una piscina de hormigón, las superficies son duras y por tanto los reflejos empeoran mucho la precisión. Este hecho no ocurrirá cuando se trabaje en la piscifactoría, donde se podrá obtener una precisión mayor.

5.4. CONTROL AUTOMÁTICO

En las pruebas que vamos a llevar a cabo en este apartado vamos a crear un algoritmo que permita capturar la información necesaria del ROV para cualquier instante y teniendo como objetivo alcanzar una determinada posición. Mediante el cálculo de la diferencia entre el valor actual y el objetivo, realizaremos un control de la velocidad a la que el ROV va a girar o avanzar.

La velocidad de movimiento del ROV va a depender proporcionalmente de la diferencia calculada. En estos casos, es necesario conocer cuál es el rango de la zona muerta (valor a partir del cual el ROV no supera la resistencia por el rozamiento del agua) y la zona de saturación (valor máximo de la velocidad de movimiento), los cuales dependerán del valor de la diferencia y de la ganancia del ROV.

Cada ROV puede trabajar en varias ganancias de movimiento (SibiuPro desde 0.2 hasta 0.8 con incrementos de 0.1 y el BlueROV2 con 0.25, 0.5 y 0.75).

5.4.1. Heading

En esta prueba queremos introducir manualmente el ángulo (heading) al que queremos que el ROV se coloque automáticamente. El algoritmo irá actualizando el valor del ángulo cada 0.5 segundos y calculando la velocidad a la que tiene que girar y el sentido de giro, de acuerdo con la siguiente gráfica:

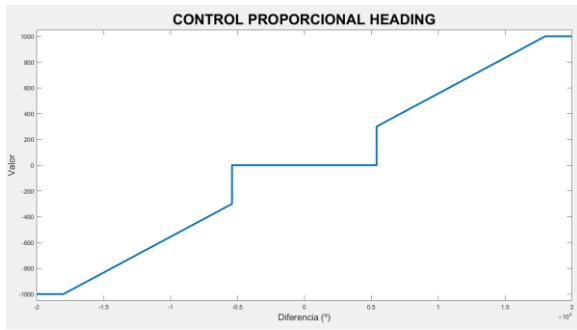


Figura 66. Gráfica control proporcional heading.

La variable de la velocidad de giro del ROV tiene un rango desde -1000 hasta 1000, girando para un sentido u otro dependiendo de si el valor es positivo o negativo, siendo 0 quedarse en estático.

Para las primeras pruebas, se utilizaron las ganancias mínimas para cada ROV provocando una mayor longitud de la zona muerta de trabajo. Esta ganancia se fue incrementando hasta que con 0.5 se pudo comprobar como el ROV alcanzaba la posición definitiva sin demasiadas oscilaciones y de forma rápida, ya que con ganancias superiores el ROV sobre oscilaba.

5.4.2. Avanzar distancia especificada

En este caso, habrá que especificar la distancia que queremos recorrer y la distancia a partir de la cual se empieza a realizar el control proporcional (por encima de ella el ROV avanza a velocidad máxima). Esta vez, se piden los datos sobre su posición que obtenemos gracias al WaterLinked y calculamos la diferencia con la posición de origen, obteniendo los valores de movimiento como se muestra en la siguiente gráfica (distancia límite = 3m):

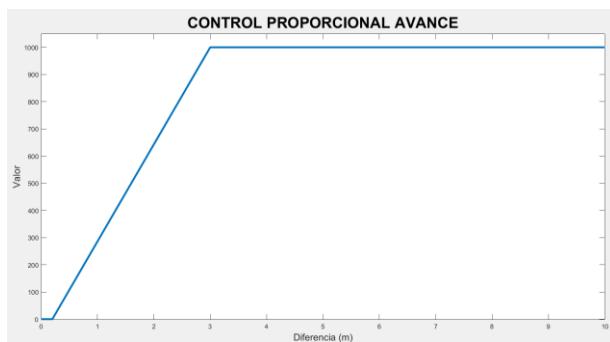


Figura 67. Gráfica control proporcional

La variable de la velocidad de avance del ROV tiene un rango desde 0 hasta 1000, siendo 500 quedarse parado.

Para las primeras pruebas, se utilizaron las ganancias mínimas para cada ROV provocando una mayor longitud de la zona muerta de trabajo. Esta ganancia se fue incrementando hasta que con 0.5 se pudo comprobar como el ROV alcanzaba la posición definitiva sin demasiadas oscilaciones y de forma rápida.

Durante estas pruebas, se pudo comprobar como la ganancia del ROV tan solo provocaba que el error entre la distancia a recorrer especificada y la medida real fuese mayor, pero el ROV siempre lograba pararse en una posición cercana al objetivo.

6. BIBLIOGRAFÍA

- [1] <https://www.programoergosum.es/tutoriales/direcciones-ip-en-raspberry-pi/>
- [2] <https://www.redeszone.net/tutoriales/servidores/servidor-openssh-linux-configuration-maxima-seguridad/>
- [3] <https://www.redeszone.net/2017/12/14/activar-cliente-openssh-oculto-windows-10/>
- [4] <http://www.ardusub.com/introduction/hardware-options/connection-diagrams.html>
- [5] <https://bluerobotics.com/learn/understanding-and-using-scanning-sonars/>
- [6] <https://bluerobotics.com/learn/ping-sonar-technical-guide/>
- [7] Manual BR-ROV Sonar. ROV Side Scan Sonar
- [8] <https://waterlinked.github.io/explorer-kit/introduction/>
- [9] Manual Oculus M750d
- [10] Manual C4E sensor
- [11] Manual OPTOD sensor
- [12] Manual NTU sensor
- [13] <http://www.ardusub.com/introduction/required-software.html>
- [14] <http://www.ardusub.com/reference/companion-web-ui.html>
- [15] <http://www.ardusub.com/reference/qgc-configuration.html>
- [16] <https://docs.bluerobotics.com/ping-viewer/>
- [17] Manual Calsens
- [18] <https://docs.bluerobotics.com/ping-protocol/>
- [19] <https://mavlink.io/en/>



Interreg

España - Portugal

Fondo Europeo de Desarrollo Regional
Fundo Europeu de Desenvolvimento Regional

