



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

**ESCUELA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**Trabajo de Grado Previo a la Obtención del Título de Ingeniero en Electrónica y
Telecomunicaciones**

Título del proyecto:

**DESARROLLO DE UN ROBOT PARA LA EXPLORACIÓN Y RECOLECCIÓN
DE INFORMACIÓN DE LA BIODIVERSIDAD SUBMARINA A TRAVÉS DE
VIDEO EN LA LAGUNA DE ATILLO**

AUTORES:

Jorge Andres Collaguazo Enriquez
Cristian Alexis Armijo Tapia

TUTOR:

Ing. Juan Carlos Cepeda

Riobamba - Ecuador

AÑO 2017

REVISIÓN DEL DE LOS MIEMBROS DEL TRIBUNAL

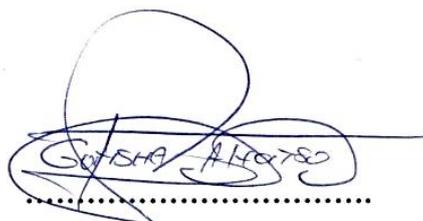
Los miembros del Tribunal de Graduación del Proyecto de investigación de título **DESARROLLO DE UN ROBOT PARA LA EXPLORACIÓN Y RECOLECCIÓN DE INFORMACIÓN DE LA BIODIVERSIDAD SUBMARINA A TRAVÉS DE VIDEO EN LA LAGUNA DE ATILLO**, presentado por: **Jorge Andres Collaguazo Enriquez** y **Cristian Alexis Armijo Tapia**, con asesoría del **Ing. Juan Carlos Cepeda**.

Una vez escuchada la defensa oral y revidado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia de la biblioteca de la facultada de ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Alfonso Gunsha

Miembro del Tribunal

A handwritten signature in blue ink, appearing to read 'ALFONSO GUNSHA', written over a horizontal dotted line.

Firma

Ing. Giovanni Cuzco

Miembro del Tribunal

A handwritten signature in blue ink, appearing to read 'GIOVANNI CUZCO', written over a horizontal dotted line.

Firma

Ing. Juan Carlos Cepeda

Director del Proyecto

A handwritten signature in blue ink, appearing to read 'JUAN CARLOS CEPEDA', written over a horizontal dotted line.

Firma

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de investigación corresponde exclusivamente a Jorge Andres Collaguazo Enriquez, Cristian Alexis Armijo Tapia y al Ingeniero Juan Carlos Cepeda, director del proyecto. Algunas ideas, expresiones, pensamientos, conceptos se han tomado de varios autores como también del internet, ubicando la respectiva autoría como es noma de la investigación científica para enriquecer el estado del arte o marco Teórico. Con respecto a los resultados obtenidos son propiedad exclusiva de los autores, del director del proyecto y patrimonio intelectual de la Universidad Nacional de Chimborazo.



Jorge Andres Collaguazo Enriquez

C.I. 060341513-4



Cristian Alexis Armijo Tapia

C.I. 060394858-9

AGRADECIMIENTO

Este presente trabajo es la muestra de haber llegado al final de nuestra vida universitaria.

Agradecemos a Dios y a todas aquellas personas quienes hemos llegado a conocer en esta etapa tan importante de nuestras vidas.

A nuestras familias y en especial a nuestros padres y hermanos, por el apoyo en cada decisión que se ha tomado.

A nuestros amigos que de una u otra manera correlacionaron gratos momentos que encaminaron nuestra vida.

A los docentes quienes encaminaron nuestras vidas con la certeza de alcanzar un porvenir alentador.

Jorge Collaguazo

Cristian Armijo

DEDICATORIA

Quiero dedicar este trabajo a Dios por ayudarme en todos los momentos de mi vida, por darme la fuerza necesaria para salir adelante.

A mi familia que ha estado a mi lado en todo este tiempo brindándome su apoyo.

Y a todas aquellas personas que siempre me han incentivado, a través de acciones y palabras alentadoras.

Jorge Andres Collaguazo Enriquez

Dedico con todo amor a Dios padre por darme fuerzas, iluminar mi mente.

A mis padres y hermano por ser mi motivación y aliento. A mis docentes de mi querida universidad por compartir sus enseñanzas.

A mis compañeros, amigos que compartieron conocimientos, alegrías y tristezas que durante este periodo brindaron apoyo para que este sueño se haga realidad.

Cristian Alexis Armijo Tapia

CONTENIDO

Índice de Figuras	x
Índice de Tablas.....	xii
RESUMEN	xiii
ABSTRACT	xiv
1. INTRODUCCIÓN	1
2. OBJETIVOS.....	3
2.1. OBJETIVO GENERAL	3
2.2. OBJETIVOS ESPECÍFICOS	3
3. ESTADO DEL ARTE.....	4
3.1. ANTECEDENTES DEL TEMA	4
3.2. ENFOQUE TEÓRICO	6
3.2.1. ROV (VEHÍCULO OPERADO REMOTAMENTE)	6
3.2.2. CONVENCIONES PARA LA NAVEGACIÓN.....	6
3.2.3. ESTRUCTURA DEL ROBOT SUBMARINO.....	7
3.2.4. AISLAMIENTO	7
3.2.5. PRESIÓN HIDROSTÁTICA	8
3.2.6. TARJETA DE DESARROLLO	8
3.2.7. CÁMARA.....	9
3.2.8. ILUMINACIÓN SUBMARINA	9
3.2.9. SISTEMAS DE PROPULSIÓN	10
3.2.10. BATERÍAS.....	12
3.2.11. COMUNICACIÓN I2C.....	12
3.2.12. TELEOPERACIÓN	13
3.3. DEFINICIÓN DE TÉRMINOS BÁSICOS.....	13
4. METODOLOGÍA.....	14

4.1.	TIPO DE ESTUDIO	14
4.1.1.	EXPLORATORIO.....	14
4.2.	MÉTODOS, TÉCNICAS E INSTRUMENTOS	14
4.2.1.	MÉTODOS	14
4.3.	TÉCNICAS.....	15
4.3.1.	OBSERVACIÓN.....	15
4.3.2.	EXPERIMENTACIÓN	15
4.3.3.	INSTRUMENTOS	15
4.3.4.	POBLACIÓN Y MUESTRA	15
4.3.5.	HIPÓTESIS	15
4.3.6.	OPERACIONALIZACIÓN DE LAS VARIABLES	16
5.	RESULTADOS	17
5.1.	ELABORACIÓN DE LA ESTRUCTURA SUBMARINA.....	17
5.1.1.	IDENTIFICACIÓN DEL MATERIAL ADECUADO PARA SOPORTAR LAS CONDICIONES A LAS QUE EL ROV VA A SER SOMETIDO.....	17
5.1.2.	DETERMINAR CARACTERÍSTICAS PARA LA IMPERMEABILIDAD DEL SISTEMA ELÉCTRICO Y ELECTRÓNICO.....	18
5.1.3.	SISTEMA DE DESPLAZAMIENTO Y CONTROL	18
5.2.	PROGRAMACIÓN CONFIGURACIÓN DEL SOFTWARE	21
5.3.	EXPERIMENTACIÓN	24
5.3.1.	TEST 1.....	24
5.3.2.	PREPARACIÓN PARA LA EXPERIMENTACIÓN DE CAMPO	24
5.3.3.	TEST 2.....	24
5.3.4.	TEST 3.....	24
6.	DISCUSIÓN	27
7.	CONCLUSIONES Y RECOMENDACIONES.....	28

7.1.	CONCLUSIONES.....	28
7.2.	RECOMENDACIONES	29
8.	BIBLIOGRAFÍA.....	30
	ANEXOS	32
	ANEXO 1.....	32
	FOTOGRAFÍAS TOMADAS A DIFERENTES ÁNGULOS DE LA CÁMARA.....	32
	ANEXO 2.....	33
	PRUEBAS DE FUNCIONAMIENTO A BAJA PROFUNDIDAD.....	33
	ANEXO 3.....	36
	RECONOCIMIENTO DEL LUGAR DE INVESTIGACIÓN.....	36
	ANEXO 4.....	38
	EXPERIMENTACIÓN EN CAMPO A BAJA PROFUNDIDAD.....	38
	ANEXO 5.....	40
	EXPERIMENTACIÓN DE CAMPO A MAYOR PROFUNDIDAD.....	40
	ANEXO 6.....	42
	PRUEBA DE FUNCIONAMIENTO A UNA ALTA PROFUNDIDAD.....	42
	ANEXO 7.....	46
	EVIDENCIAS DE LA CODIFICACIÓN.....	46
	ACTIVAR LOS ADC EN AVR.....	46
	CONFIGURACIÓN DEL MASTER.....	50
	CONFIGURACIÓN MOVIMIENTO SERVO.....	51
	VELOCIDAD DE MOTORES PROPULSORES.....	55
	CONFIGURA PARA GRABAR EN PROXI, PUERTO DE 8090	57
	CAMBIAR IMÁGENES Y COMADOS DEL TECLADO.....	57
	CAMBIAR LA DIRECCIÓN ETHERNET DE LOCALIZACIÓN.....	60

Índice de Figuras

Figura 1. SNOOPY hidráulico de la Marina de los Estados Unidos.....	4
Figura 2. VICTOR 6000 explorador de fondos marinos.....	4
Figura 3. ROV propulsado por aire comprimido.....	5
Figura 4. Notación en robótica submarina.....	6
Figura 5. Ubicación de los componentes esenciales de la tarjeta.....	9
Figura 6. Comunicación I2C entre Mater y Esclavo.....	13
Figura 7. Carcasa Terminada.....	17
Figura 8. Mecanismos de desplazamiento.....	17
Figura 9. Mecanismo de impermeabilización y alivio de presión.....	18
Figura 10. Espesor del casco de compresión.....	18
Figura 11. Actuadores (Motores Brushless Turnigy).....	19
Figura 12. Cámara Genius Wide Cam F100.....	20
Figura 13. Placa de iluminación.....	20
Figura 14. Micro servo motor de 2.6Kg.....	20
Figura 15. Kit de transmisión y recepción de datos.....	21
Figura 16. Cabina de control del ROV.....	23
Figura 17. Representación de datos en diagrama de barra.....	26
Figura 18. Fotografía con servo a 0° con respecto al horizonte.....	32
Figura 19. Fotografía con servo a 60° grados con respecto al horizonte.....	32
Figura 20. Fotografía con serbo a -60° grados con respecto al horizonte.....	32
Figura 21. Robot submarino y carrete de Cable de conexión.....	34
Figura 22. Fotografía tomada a 1.7m de profundidad con dirigida hacia arriba.....	34
Figura 23. Fotografía tomada a 1.7m de profundidad con posición normal.....	34
Figura 24. Encendidos del mecanismo de iluminación de ambiente.....	35

Figura 25. Fotografías de la piscina de la UNACH, capturada por el robot Submarino.	35
Figura 26. Fotografías del medio de experimentación día 1.	36
Figura 27. Visitando el lugar para la investigación de Campo.....	36
Figura 28. Identificación del lugar donde se realizará la primera prueba del día 2.....	36
Figura 29. Mapa de coordenadas de las primeras pruebas.	37
Figura 30. Mapa de coordenadas de las pruebas finales.....	37
Figura 31. Robot explorando por la superficie.	38
Figura 32. Fotografía tomada por el robot de exploración de la superficie.....	38
Figura 33. Fotografía tomada por el robot de exploración a una profundidad de 2 metros con la cámara visualizando hacia la parte inferior.....	39
Figura 34. Fotografía tomada por el robot de exploración a una profundidad de 2 metros con la cámara visualizando al frente.....	39
Figura 35. Fotografía llevando el vote hasta la orilla de la laguna.....	40
Figura 36. Llevando los materiales para realizar el Test 3.....	40
Figura 37. Fotografía tomada por el robot a una profundidad de 2.5 metros.....	40
Figura 38. Fotografía tomada por el robot a una profundidad de 5 metros.	41
Figura 39. Fotografía tomada por el robot a una profundidad de 7.5 metros.....	41
Figura 40. Fotografía tomada por el robot a una profundidad de 9 metros.	41
Figura 41. Imágenes tomadas en los niveles de iluminación en estado no sumergido.....	43
Figura 42. Imágenes tomadas en los niveles de iluminación con el robot sumergido.	43
Figura 43. Fotografía del interfaz gráfico con la profundidad a nivel del agua.	44
Figura 44. Fotografía del interfaz gráfico con la profundidad a 2 metros.....	44
Figura 45. Fotografía del interfaz gráfico con la profundidad a 7 metros.....	45
Figura 46. Fotografía del interfaz gráfico con la profundidad a 9 metros.....	45

Índice de Tablas

Tabla 1. Materiales propuestos para la construcción del ROV.	7
Tabla 2. Presión hidrostática según la profundidad.....	8
Tabla 3. Comparación entre los 3 tipos más comunes de LED SMD.	10
Tabla 4. Principales características de los motores BrushlessDST-700 y características del controlador (Afro ESC 12amp BEC).	11
Tabla 5. Características de las bateras.	12
Tabla 6. Operacionalización de las variables.	16
Tabla 7. Caracteres de control.	22
Tabla 8. Definiciones de los colores como indicadores dentro de la cabina.	23
Tabla 9. Calidad de imagen con respecto a la iluminación en estado no sumergido.	25
Tabla 10. Calidad de imagen con respecto a la iluminación y profundidad a 2 metros.	25
Tabla 11. Calidad de imagen con respecto a la iluminación y profundidad a 7 metros.	25
Tabla 12. Calidad de imagen con respecto a la iluminación y profundidad a 9 metros.	26
Tabla 14. Funcionamiento de los diferentes sistemas del robot en ambiente y a 1 metro de profundidad.	33
Tabla 15. Funcionamiento de los diferentes sistemas del robot a 5 metros y a 10 metro de profundidad.	42

RESUMEN

El presente proyecto describe el desarrollo de un robot teleoperado que pueda desplazarse por un entorno submarino y recoger información visual a través de una videocámara. Nuestro sistema consta de un chasis en la cual se colocan tres motores para el desplazamiento del robot, un mecanismo impermeabilización de circuitos y otro para las baterías mediante empaques y pegamento epóxico que soporta dos toneladas de presión, un control de movimiento de la cámara por medio de un servo y ejes que facilitan los setenta grados de giro vertical, el sistema de transmisión y recepción mediante IP.

La característica principal del proyecto es recoger información de carácter visual, es decir en forma de fotografías y video con una profundidad máxima de nueve metros, para lo cual se implementa un sistema operativo Linux que maneja al hardware principal, un Bootloader para el gestor de arranque del dispositivo, obteniendo una plataforma base donde concatenar los diferentes archivos de control de motores, serbo, cámara, designación de la dirección IP, iluminación, configuración del protocolo I2C, parámetros para la grabación de video e interface de usuario. Todos estos programas con sus respectivas librerías y archivos ejecutables se inician al prender el Robot.

Como conclusión se extrae que el robot puede recolectar información visual a pocas profundidades y se garantiza la hermeticidad.

Palabras Clave: impermeabilización, transmisión, recepción, Bootloader.

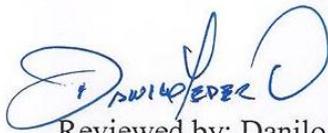
ABSTRACT

This project describes the development of a teleoperated robot that can navigate through an underwater environment and collect visual information through a camcorder. Our system consists of a chassis in which are placed three motors for the displacement of the robot, a waterproofing mechanism of circuits and another for the batteries by means of gaskets and epoxy glue that supports two tons of pressure, a control of movement of the camera through Of a servo and axes that facilitate the seventy degrees of vertical rotation, the system of transmission and reception through IP.

The main feature of the project is to collect visual information, ie in the form of photographs and video with a maximum depth of nine meters, for which is implemented a Linux operating system that handles the main hardware, a Bootloader for the boot loader Of the device, obtaining a base platform to concatenate the different engine control files, serbo, camera, IP address designation, lighting, I2C protocol configuration, parameters for video recording and user interface. All these programs with their respective libraries and executable files are started when the Robot is started.

The conclusion is that the robot can collect visual information in a few depths and ensure the tightness.

Key words: waterproofing, transmission, reception, Bootloader.



Reviewed by: Danilo Yopez O.
English professor



1. INTRODUCCIÓN

La exploración de ecosistemas subacuáticos es muy limitada debido a las condiciones climáticas, el tiempo de sondeo, difícil localización y profundidad, desafíos tecnológicos que conlleva la investigación de estos medios y el peligro de la exploración humana en lugares llenos de vegetación, sitios angostos, que pueden impedir el ascenso o descenso del buzo.

Cuando se hace referencia a una estructura que pueda soportar una presión de dos atmosferas, se toma en cuenta su peso, las líneas de grosor, material anticorrosivo, además en el diseño es importante que se oponga lo menos posible al paso del agua, es decir que permita un desplazamiento fácil dentro del líquido.

Se ha conseguido por medio de procedimientos de ordenado la creación de un interfaz capaz de mostrar video en tiempo real, estado de la batería, profundidad de Robot, estado de conexión, porcentaje de lúmenes emitida por el robot para poder visualizar el entorno, el porcentaje de velocidad con el que los motores funcionan en el momento, de esta manera se puede interactuar con todo el sistema en una sola pantalla, lo cual facilita el control del desplazamiento y grabación del entorno marino en el que se realiza la investigación.

Para la comunicación entre dispositivos se utiliza I2C, sistema que permite trabajar con un dispositivo como maestro y varios como esclavos, de esa manera utiliza dos líneas de comunicación. Es importante recalcar la parte electrónica, de control y potencia, se encuentran distribuidas de la manera que no tengan fluctuaciones entre sí y adecuados en el menor espacio físico posible, de esta forma el entorno posea menos presión atmosférica, lo cual ayuda a que el robot ejerza menos potencia al momento de sumergirlo.

Este trabajo se encuentra distribuido en varias partes, entre las cuales consta:

- Análisis de baterías, estabilidad, actuadores y peso, para el robot pueda sumergirse con la mejor efectividad posible
- Diseño estructural el cual menciona las dimensiones físicas del robot
- Propuesta para amplificar la señal mayor a 10 metros
- Acoplamiento del sistema de visualización (cámara) y el mecanismo de control (Motores)
- Mecanismo de impermeabilización se realiza para proteger la parte electrónica y de potencia.

En cada una de estas partes se evidencia la investigación a posibles alternativas que se usaría para la elaboración del proyecto, se toma en cuenta varios factores como el tamaño de los materiales, calidad, compatibilidad entre sí, precisión y precio.

Para el desarrollo de la programación necesaria para el funcionamiento, se necesita conocimiento sobre varios sistemas de programación como: Raspberry pi3 para cargar la imagen del sistema en el microcontrolador, programación en C para desarrollar el código

base y cargar librerías, Linux, Arduino, programación en HTML para la creación del interfaz de usuario con direccionamiento IP.

Uno de los parámetros importantes que se tomó en cuenta el tiempo de exploración ya que el mecanismo de transmisión se alimenta de forma independiente desde la conexión de 5 Voltios y el sistema electrónico desde las baterías que se encuentra ubicadas en la parte inferior del robot, esto permite ampliar el reconocimiento del lugar por lo menos una hora de trabajo continuo y en modo de solo visualización sin que los actuadores funcionen hasta dos horas y media lo cual brinda una ventaja.

La tecnología actual es un instrumento fundamental para la investigación científica, ya sea en laboratorios o en trabajos de campo. Los robots teledirigidos son ampliamente utilizados en entornos de difícil acceso o riesgosos para el investigador.

Este proyecto es importante ya que permite realizar una exploración en ambientes subacuáticos, de esta manera es posible la recolección de información por medio de imágenes y video en tiempo real, obtener datos característicos del agua que ayudan a tomar decisiones con respecto a la contaminación que pueda existir en el medio.

La posibilidad de tener el conocimiento de la profundidad que se encuentra el robot, brinda una ayuda excepcional al control y desplazamiento del robot ya que entrega conocimiento preciso para tomar decisiones, junto con la cámara, para tener una mejor estabilidad.

Teniendo en cuenta que el océano cubre dos terceras parte del planeta, con esta propuesta pueden surgir descubrimientos, que se desarrollarían minuciosamente, respondiendo a grandes preguntas como que se pueda encontrar en el interior de una fosa tectónica inexplorada o dentro de una corriente marina como en las galápagos, estas investigaciones tendrían como material principal un robot submarino tele operado con el cual se demostraría las curiosidades humanas con material científico comprobado.

La investigación del proyecto brinda una base para desarrollos a futuro en la indagación tanto en la rama de comunicaciones, como los mecanismos que se puedan añadir al proyecto, para el futuro se podría aumentar un brazo robótico con el cual se pueda obtener muestras solidas de los ambientes explorados, lo cual abre un mundo de exploración de vestigios marinos, por otra parte también se puede desarrollar un mecanismo de succión de agua con la cual se obtendría muestras que faciliten al investigador la obtención de datos primordiales como la pureza o contaminantes en aguas profundas.

2. OBJETIVOS

2.1.OBJETIVO GENERAL

Desarrollar un robot para la exploración y recolección de información de la biodiversidad submarina a través de video en la Laguna de Atillo.

2.2.OBJETIVOS ESPECÍFICOS

- Realizar el sistema de comunicación entre submarino y operador.
- Implementar el sistema de control del robot tele-operado.
- Desarrollar la estructura para que soporte nueve metros de Profundidad.
- Diseñar un método que permita emersión y descenso del Robot.
- Transmitir video en tiempo real.
- Inclusión de sensores de profundidad.

3. ESTADO DEL ARTE

3.1. ANTECEDENTES DEL TEMA

La marina de los Estados Unidos desarrollo uno de los primeros ROVs de observación de pequeño tamaño. Lo llamaron SNOOPY, el cual era operado hidráulicamente desde la superficie, por su tamaño fue uno de los primeros vehículos portátiles. La versión que le sigue fue el SNOOPY eléctrico, que su alcance se amplió, al cual se le añadió sonares y otros sensores. Se puede ver en la Figura 1. (Robert D. Christ, 2011)



Figura 1. SNOOPY hidráulico de la Marina de los Estados Unidos.

Fuente: (Robert D. Christ, 2011)

VICTOR 6000 este robot diseñado para explorar los fondos marinos a distancia, puede realizar imágenes de alta calidad, puede transportar y operar diversos equipos, herramientas científicas, alcanzando una profundidad de 6000 metros. (Marie, 2010) Se muestra en la Figura 2.



Figura 2. VICTOR 6000 explorador de fondos marinos.

Fuente: (Marie, 2010)

El Ingeniero Carlos Quesada en su proyecto Estudio y diseño de un ROV propulsado por aire comprimido. Presenta el proyecto donde hace el estudio y diseño de un sumergible no tripulado propulsado mediante un motor de aire comprimido. Las partes que se van a diseñar va a ser el chasis, los ejes de transmisión, los conductos por donde transcurra el aire y el anclaje de todos los componentes. A demás de ser diseñado en apoyo a buceadores, que proporcionará diferentes sistemas de ayuda, como la posibilidad de

compartir aire en caso de emergencia, por lo cual la profundidad máxima será de 40-50 metros, con una autonomía mayor a 1 hora. (Carlos, 2013-2014). Se muestra en la Figura 3.

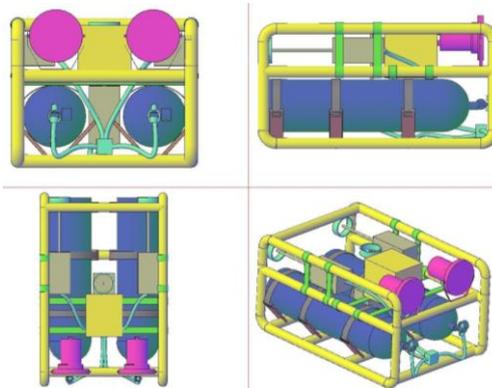


Figura 3. ROV propulsado por aire comprimido.
Fuente: (Carlos, 2013-2014)

El Dr. Rafael García graduado en la universidad Autónoma de Barcelona en ciencias de la computación, Muestra un punto de vista enfocado hacia la “Robótica Submarina como una Herramienta clave para la Exploración del fondo del Mar”, la cual habla sobre la importancia de la toma de imágenes submarinas para caracterizar la naturaleza, distribución de características Geológicas y comunidades Biológicas, extraer indicadores ecológicos y para proporcionar registros permanentes de la condición del fondo del mar. (Rafael, 2012)

Los ingenieros Héctor Moreno, Roque Saltaren, Lisandro Puglisi, Isela Carrera en su trabajo titulado Robótica submarina: Conceptos. Elementos, Modelado y Control, hablan acerca de “El conocimiento científico de los mares profundos está creciendo rápidamente mediante el uso de una variedad de tecnologías. Las primeras exploraciones científicas se llevaron a cabo mediante vehículos submarinos ocupados por humanos. Recientemente, los robots submarinos han comenzado a revolucionar la exploración del fondo marino, ofreciendo mejor información a un costo más reducido. Estos robots han permitido realizar operaciones en aguas profundas, y también, intervenir en desastres como lo son las fugas en instalaciones petrolíferas”. (Hector A Moreno, 2014)

Con respecto a los sensores se mencionan como una parte esencial para la localización, velocidad y aceleración del Robot, además del estado de funcionamiento del mismo. Se pueden clasificar en tres grupos los de posicionamiento, los de estado interno y los de medición del ambiente.

Un sensor interesante se denomina IMU (Unidad de Medición Inercial) el cual está compuesto de tres giroscopios que entregan información de cabeceo, alabeo y guiñado, informando la aceleración lineal del vehículo, así como la velocidad angular. (Hector A Moreno, 2014)

3.2.ENFOQUE TEÓRICO

3.2.1. ROV (VEHÍCULO OPERADO REMOTAMENTE)

Los robots submarinos operados remotamente (ROVs, por las siglas en inglés de Remotely Operated Vehicle) están conectados a la superficie mediante un cordón umbilical (un conjunto de cables unidos), que permite el intercambio de datos y la alimentación de energía al robot. A través de una interfaz gráfica del ordenador situado en la superficie, el usuario define los comandos que el robot deberá ejecutar. A su vez el ROV envía las señales de sus sensores (presión, temperatura, imágenes, video, etc.) al ordenador en superficie, para que el usuario conozca el estado del robot y el ambiente que lo rodea. (Hector A Moreno, 2014) Se menciona dos tipos de ROVs:

3.2.1.1.Tele-operados

El operador va observando por las pantallas lo que el ROV ve, y va dándole unas pautas ya programadas de lo que debe hacer, como por ejemplo subir a tantos metros, u otras operaciones ya programadas, por lo que el ROV ya tiene predefinidas unas operaciones y el operador sólo se encarga de introducir qué operación debe realizar en ese momento. (Francy, 2014)

3.2.1.2.Operados con mando a distancia

El operador dirige completamente al ROV con un joystick y ve en cada momento lo que está realizando el ROV por lo que el ROV no realiza ninguna operación de forma automática o programada. (Francy, 2014)

3.2.2. CONVENCIONES PARA LA NAVEGACIÓN

En navegación marítima y robótica submarina existe una convención (SNAME) para expresar la posición, orientación y fuerzas que se ejercen sobre el vehículo. Representación de la posición del vehículo con respecto al marco inercial N. Se muestra en la Figura 4.

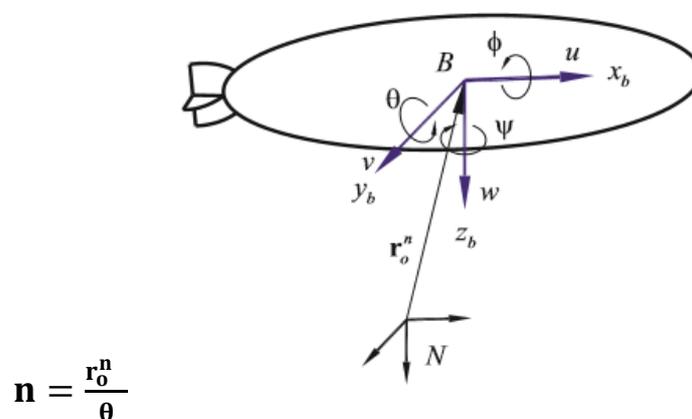


Figura 4. Notación en robótica submarina.

Fuente: (Hector, 2013)

3.2.3. ESTRUCTURA DEL ROBOT SUBMARINO

La cabina o casco consiste de la estructura y material que reviste al robot submarino. Los factores que determinan el diseño son: la profundidad a la que va a operar, los niveles de temperatura que va a resistir, la resistencia a la corrosión, el volumen de los elementos que contendrá, la facilidad para ensamblar y desensamblar, factibilidad de su construcción y el costo. (Hector, 2013) Se muestra en la Tabla 1.

Material	Densidad (kg/dm³)	Resistencia a la fluencia (MPa)	Módulo de Tensión (GPa)	Resistencia Especifica (kNm/kg)
Acero alta Resistencia (HY80)	7.86	550	207	70
Aleación de Aluminio (7075)	2.9	503	70	173
Acrílico	1.2	103	3.1	86
PVC	1.4	48	35	34

Tabla 1. Materiales propuestos para la construcción del ROV.

Fuente: (Hector, 2013)

3.2.4. AISLAMIENTO

Un O-Ring es el elemento más accesible y eficaz para prevenir la fuga de fluidos bajo distintas condiciones de servicio. (Parker). Sus principales ventajas son:

- Puede ser utilizado como sello estático, cuando las piezas no tienen movimiento y también en uso dinámico de vaivén, oscilante o rotativo.
- Permite el sellado en todas las direcciones, radial, axial u oblicua.
- El fluido puede ser líquido o gaseoso.
- Disponible en medidas y tolerancias normalizadas internacionalmente.
- Se dispone de una amplia variedad de dimensiones y compuestos.
- Son de bajo precio relativo.
- En muchos casos permiten su desarme y reutilización.
- Cierran en un amplio rango de presiones, temperaturas y tolerancias.
- Una vez instalados no requieren de ajustes durante su tiempo de vida útil.
- Son de peso reducido y requieren muy poco espacio, permitiendo un diseño muy compacto.

3.2.5. PRESIÓN HIDROSTÁTICA

Se define presión como la fuerza dividida por la superficie ($P=F/S$). Como unidad de presión se utiliza la atmósfera, la que ejerce el aire que rodea a nivel del mar (el peso de un cilindro de mercurio de 760mm, de altura y cuya base tenga una superficie de 1 cm²). El valor de 1 atmósfera es la presión resultante de efectuar una fuerza de 1 Kg. en una superficie de 1 cm² y aproximadamente equivale a un bar (1,03 bares = 1 atm). Se muestra en la Tabla 2.

Hay que distinguir en buceo dos tipos de presión: la atmosférica (El peso del aire sobre la superficie del agua) y la hidrostática (el peso del agua sobre el submarinista), La suma de las dos presiones parciales da como resultado la presión absoluta, que es la que vendría afectar. (Donosti)

$$\text{Presión absoluta} = \text{presión hidrostática} + \text{presión atmosférica}$$

Y sustituyendo los términos por sus valores, se halla la relación entre profundidad y presión.

$$\text{Presión absoluta} = (\text{profundidad} / 10) + 1$$

Aplicando la fórmula, se puede ver como varia la presión a medida que aumenta la profundidad.

Profundidad	Ubicación	Presión Absoluta
0 metros	(en superficie y a nivel del mar)	1 atm.
10 metros	(bajo el agua)	2 atm.
20 metros	(bajo el agua)	3 atm.

Tabla 2. Presión hidrostática según la profundidad.

Fuente: (Donosti)

3.2.6. TARJETA DE DESARROLLO

3.2.6.1. Beagle Bone Black

Es una tarjeta de bajo costo, con un sistema operativo (Debian Linux) precargado permitiendo a desarrolladores y aficionados un fácil acceso y un entorno de desarrollo amigable, además es compatible con otros software basados en Linux como lo es Android, Ubuntu, Cloud9 IDE en Node.js w/BoneScript library, etc.

3.2.6.2. Ubicación de los componentes claves

La tarjeta BeagleBone Black conformada por un procesador AM335x 1GHz ARM Cortex-A8, Memoria RAM de 512MB DDR3, Memoria de almacenamiento de 4GB de 8bits eMMC, chip de administración de energía IC TPS65217, Transceptor Ethernet PHY, Chip HDMI Framer TDA19988. Se muestra en la Figura 5.

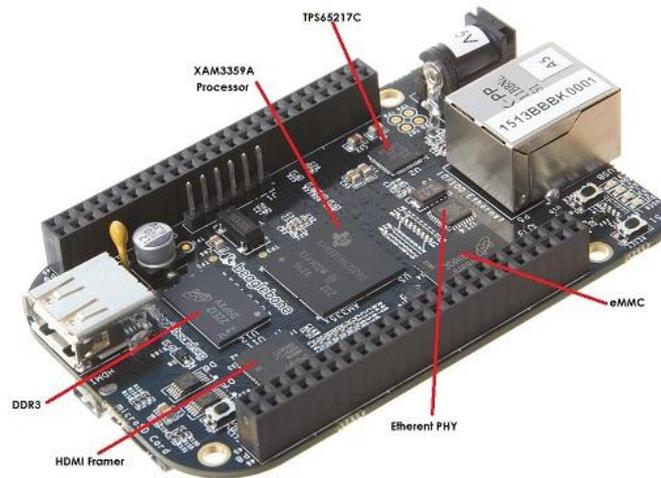


Figura 5. Ubicación de los componentes esenciales de la tarjeta.

Fuente: (BeagleBoard, 2017)

3.2.7. CÁMARA

El ROV dispondrá de una cámara. La función principal es mandar una señal de vídeo que permita al operador poder visionar por dónde va el vehículo para no tener ningún impacto. Una función secundaria es permitir grabar vídeos y capturar imágenes para un posterior análisis o para almacenarlas. Para grandes profundidades se usan cámaras infrarrojas o en blanco y negro. Para poca profundidad se pueden usar perfectamente cámaras a color. En este caso se usará una cámara de color, ya que la profundidad no será muy grande. Esto proporcionará mayor detalle, poder diferenciar perfectamente texturas, imperfecciones o materiales de lo que se capte con las imágenes. (Carlos, 2013-2014)

Se selecciona una cámara modelo WIDECAM F100 de la marca GENIUS la cual tiene un lente ultra angular de 120 grados que permite una amplia visualización, además permite grabar video en Full HD 1080p con hasta 30 cuadros por segundo. (GENIUS, 2017)

3.2.8. ILUMINACIÓN SUBMARINA

Se dispondrán de LEDs de tipo SMD orientados hacia el frente, para una visualización más clara bajo del agua. De esta forma poder iluminar la zona donde se encuentre además de ayudar a buzos a poder ver bajo poca iluminación. (Carlos, 2013-2014)

LED SMD

Es un LED encapsulado en una resina semirrígida, ensamblada de manera superficial, permitiendo de esta forma tener una gran superficie semiconductor ofreciendo calidad y funcionamiento óptimo. Por lo general su vida útil es de 50000 horas o de 6 años funcionando las 24 horas del día, los más comunes son los siguientes, además sus características se muestran en la Tabla 3.

- **3528:** Son pequeños y de poca potencia. Se puede encontrarlos en tiras de LEDs o en diólicas agrupados en gran cantidad.
- **5050:** Este tipo encapsulan tres LEDs lo que equivale al 3528. Son de uso más común y se encuentra en muchas bombillas LED.
- **5630:** Este tipo de LED SMD es algo más actual y más potente que el 5050, además tiene un tamaño inferior. Estas características podrían dar a entender que es la mejor elección, pero por lo visto la gente está bastante descontenta con el resultado en cuanto a durabilidad.

	SMD 3528	SMD 5050	SMD 5630
Potencia	0.08W	0.24W	0.5W
Luminosidad	6lm	20lm	50lm
Eficiencia	90lm/w	90lm/w	110lm/w

Tabla 3. Comparación entre los 3 tipos más comunes de LED SMD.

3.2.9. SISTEMAS DE PROPULSIÓN

Este sistema permite que los robots submarinos avancen en el medio acuático. Se considera parte del sistema de propulsión, tanto a los elementos que generan las fuerzas, como aquellos que controlan la dirección de estas fuerzas. La mayoría de estos sistemas consisten de impulsores de hélices, aletas móviles para el guiado, y sistemas de lastre para realizar las operaciones de viraje. (Hector, 2013)

3.2.9.1. Impulsores de Hélice

El sistema de empuje con hélice y motor es el más utilizado en la mayoría de robots submarinos. Consiste generalmente en un motor eléctrico, al cual se le acopla una hélice. Al girar la hélice produce un efecto de empuje al desplazar el fluido de adelante hacia atrás, debido a la diferencia de presión producida. (Hector, 2013)

Se utiliza un modelo matemático sencillo del impulsor con el cual la fuerza del impulsor es proporcional al cuadrado de la velocidad angular de la hélice y a su vez esta velocidad

es proporcional al voltaje del motor tomando en cuenta el supuesto de que la dinámica de los impulsores tiene unas constantes de tiempo menores a la dinámica del vehículo. (Hector A Moreno, 2014)

La hélice es un dispositivo constituido por un número variable de aspas (2, 3, 4...) que al girar alrededor de un eje producen una fuerza propulsora. Cada pala está formada por un conjunto de perfiles hidrodinámicos que van cambiando progresivamente su ángulo de incidencia desde la raíz hasta el extremo. (Francy, 2014)

3.2.9.2. Motores y Controlador de Motores Brushless

Los motores elegidos son de tipo Brushless Out-runner motor 700kv (Turnigy Aerodrive DST-700) cuenta con las siguientes características que se pueden apreciar en la Tabla 4.

EL controlador Afro ESC 12amp BEC UltraLite Multirotor ESC V3 (SimonK firmware). Con una alta eficiencia, todo el diseño MOSFET de canal N, entrada PWM de hardware sin fluctuaciones, y la respuesta de potencia uniforme. El buque de la serie CES Afro con un software específico SimonK multi-rotor precargado, por lo cual no necesita de reflash. (Simonk, 2017). Cuenta con las siguientes características planteadas en la Tabla 4.

Motores		Controlador Brushless	
Kv	700 RPV	Consumo de corriente	12 A continuo
Voltaje	7,4~14,4V	Rango de voltaje	2-4s Lipo
Corriente máxima	11 A	BEC	5.0V 0.5 Linear
Potencia máxima	140W	Frecuencia de entrada	1KHz
Resistencia	0.27ohms	Firmware	afro_nfet.hex
Tamaño del eje	M3	Alambre de descarga/tapones	20 AWG/2mm macho
Peso	65g	Cable de motor/tapones	20AWG/2mm hembra
Diámetro del motor	30mm	Peso	10g
Sugerido ESC	15 A	Tamaño	25x20x6 mm

Tabla 4. Principales características de los motores BrushlessDST-700 y características del controlador (Afro ESC 12amp BEC).

Fuente: (Turning, 2017), (Simonk, 2017)

3.2.10. BATERÍAS

Están compuestas por una o más celdas electro-químicas que convierten la energía química almacenada en energía eléctrica. Las mismas se clasifican en primarias y secundarias. Las baterías primarias no se pueden re-cargar, mientras las baterías secundarias si son recargables.

Las baterías primarias tienen una mayor densidad de energía que las secundarias (el caso del Li primario), sin embargo, su empleo es en general más costoso. Las baterías primarias más comunes son las Alcalinas, las cuales son más económicas y fácil de emplear. (Hector, 2013)

Se selecciona baterías del tipo LiFePO₄ 26650 pila recargable: 3.2V 3300 mAh (10 Wh, 10A Rate) - IEC / UN38.3. Con las siguientes características presentadas en la Tabla 5.

Voltaje nominal	Promedio 3.2 a 3.3 V
Capacidad nominal	3300 mAh
Densidad de energía	117,33 Wh / kg
Corriente de carga	3.3 A Máx
Corriente de descarga	9.9A (3,0 C) continuo
Temperatura de operación	La carga 0 - 45 o C Descarga: - 20 - 60 o C
Rendimiento ciclo	> 2000 (80% de la capacidad inicial en la tasa de 0.2C, la norma IEC) 2 veces más de NiMH y 10 tiempo más de SLA
Dimensiones (DxH) (max con pestaña)	26.21mm (1.03 ") x 65,66 mm (2,58") Aquí Dimensión (LxWxH): 10 mm (0,4 ") x 6,0 mm (0,2") x 0,2 mm (0,007")
Peso	3.2 Oz (90 gramos)

Tabla 5. Características de las baterías.

Fuente: (Battery, 2000-2016)

3.2.11. COMUNICACIÓN I2C

Circuito inter-integrado es un sistema para el intercambio de datos serial entre los microcontroladores y los circuitos integrados especializados de generación. Se utiliza cuando la distancia entre ellos es corta (el receptor y el transmisor están normalmente en la misma placa de circuito impreso). (Verle, 2009)

El bus I2C solo necesita 2 líneas para transmitir y recibir datos, los cuales son: para datos (SDA) y para la señal de reloj (SCL), esta forma de comunicación utiliza una sincronía con un tren de pulsos que viaja en la línea SCL, de tal manera que en los flancos negativos se revisan los datos RX o TX. La velocidad de transmisión puede ser de 100Kbits/seg en modo standard, 400Kbits/seg en modo rápido, 3.4Mbit/seg en alta velocidad. Cada dispositivo conectado al bus tiene un código de dirección seleccionable mediante software, por lo que existe una relación permanente Master/ Slave. El Master es el dispositivo que inicia la transferencia en el bus y genera la señal de reloj (SCL), y el Slave es el dispositivo direccionado, sin embargo, cada dispositivo reconocido por su código (dirección), puede operar como transmisor o receptor de datos ya que la línea (SDA) es bidireccional. (Reyes, 2006) Se muestra en la Figura 6.

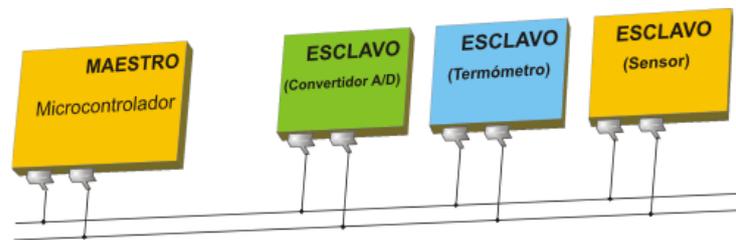


Figura 6. Comunicación I2C entre Mater y Esclavo.

Fuente: (Verle, 2009)

3.2.12. TELEOPERACIÓN

Es la acción que realiza un ser humano de manipular a distancia un dispositivo, consta de los siguientes elementos:

- **Operador:** Ser Humano que realiza a distancia el control de operación.
- **Dispositivo teleoperado:** Es la máquina que trabaja en la zona remota.
- **Interfaz:** Conjunto de dispositivos que permiten la interacción del operador con el sistema de teleoperación.
- **Control y canales de comunicación:** Conjunto de dispositivos que modulan, transmiten y adaptan el conjunto de señales que se transmiten entre la zona remota y la local.
- **Sensores:** Conjunto de dispositivos que recogen la información, de la zona local y remota para ser utilizada por la interfaz y el control. (Emmanuel Nuño Ortega, 2004)

3.3.DEFINICIÓN DE TÉRMINOS BÁSICOS

- **PH.** - Coeficiente que indica el grado de acides de un líquido.
- **ROV.** - Vehículo Operado remotamente.
- **SNAME.** - Sociedad de Arquitectos Navales e Ingenieros Marinos.
- **IMU.** - Sensor de medición Inercial.

4. METODOLOGÍA.

4.1.TIPO DE ESTUDIO

4.1.1. EXPLORATORIO

Este proyecto recurre al tipo de estudio ya que existen pocos antecedentes frente al desarrollo de robot para exploración Submarina, de esta manera se busca una solución efectiva para convergencia sistemas software y hardware, que a su vez cumplan con los objetivos trazados para la investigación y el proyecto sirva como base para futuras investigaciones.

4.2.MÉTODOS, TÉCNICAS E INSTRUMENTOS

4.2.1. MÉTODOS

Los métodos que serán utilizados para la investigación:

4.2.1.1.Método descriptivo

Consiste en realizar una exposición narrativa, numérica y/o gráfica, lo más detallada y exhaustiva posible de la realidad que se investiga.

- Análisis de las variables que intervienen en el funcionamiento del robot.
- Análisis del diseño estructural para efectuar los movimientos necesarios para realizar su objetivo.

4.2.1.2.Método investigativo experimental

- Se van a realizar una serie de experimentos a varias profundidades para determinar el correcto funcionamiento de los sistemas de control.
- Realizar las pruebas a diferentes velocidades, evidenciando la calidad de la imagen.

4.2.1.3.Método analítico

- Comprobación de las hipótesis a través de las variables y mediciones establecidas.
- Análisis del rendimiento del robot.

4.3.TÉCNICAS

4.3.1. OBSERVACIÓN

Esta técnica consiste en la recolección de información que sea de apoyo para el desarrollo del proyecto y la observación del correcto trabajo de los diferentes sistemas incorporados en el proyecto de investigación.

4.3.2. EXPERIMENTACIÓN

Dentro de la investigación es necesaria también la utilización esta técnica ya que se realizan pruebas de funcionalidad a las posibles soluciones que se presentan a los objetivos propuestos.

4.3.3. INSTRUMENTOS

Los instrumentos necesarios son libros, folletos, archivos, páginas web, blogs, cursos, DATASHEET, y conocimientos previos dentro de la carrera ya que son útiles para el diseño e implementación de los diferentes sistemas que utiliza el robot submarino.

4.3.4. POBLACIÓN Y MUESTRA

Para este tipo de investigación se toma como población al número de pruebas realizadas.

4.3.5. HIPÓTESIS

El desarrollo de un robot para la exploración y recolección de información de la Biodiversidad Submarina mediante video, nos permitirá conocerla.

4.3.6. OPERACIONALIZACIÓN DE LAS VARIABLES

En la Tabla 6 se detalla operacionalización de la variable dependiente y variable independiente.

VARIABLES	CONCEPTO	INDICADORES	INSTRUMENTOS
Desarrollo de un robot	Parámetros que intervienen en la creación de una carcasa y sistema de control apto para ser sumergido	Diseño del casco (estructura sumergible)	Tipo de algoritmo Tipo de comunicación entre sistemas Diseño de interfaz de usuario Característica de los materiales DATASHEET
		Diseño del sistema de control	
Para la exploración y recolección de información de la biodiversidad submarina a través de video en la laguna de Atillo.	Se pretende una exploración para tomar decisiones acertadas con respecto al medio indagado	Calidad de la imagen	Sensor de profundidad Cámara de fotos Indicadores ambientales Estándares ambientales
		Profundidad	
		Riesgo medioambiental	
		Visualización de fauna marina	

Tabla 6. Operacionalización de las variables.

Fuente: Autores

5. RESULTADOS

5.1. ELABORACIÓN DE LA ESTRUCTURA SUBMARINA

5.1.1. IDENTIFICACIÓN DEL MATERIAL ADECUADO PARA SOPORTAR LAS CONDICIONES A LAS QUE EL ROV VA A SER SOMETIDO.

Se buscó varias alternativas para la carcasa del submarino entre las cuales se tiene aluminio, el cual tiene varias ventajas como ser antioxidante, tener un peso relativamente bajo, pero la falta de manipulación y el alto costo se buscó otra opción, con el plástico PVC se solucionaba el costo pero aun la manipulación no dejaba de ser un problema, al seguir investigando se propuso la idea de utilizar Acrílico por su fácil adquisición, manipulación y resistencia frente a la presión hidrostática de dos atmosferas que debe resistir para dar como cumplido uno de los objetivos.

En la Figura 7 se muestra el modelo de la carcasa termina con las dimensiones de 30cm de largo, 20cm de ancho y 16cm de alto, con 3mm de espesor del acrílico.



Figura 7. Carcasa Terminada.

Fuente: Autores

La Figura 8 muestra el mecanismo de propulsión, desplazamiento e inmersión con la respectiva colocación de los motores y cableado, teniendo en cuenta que las fuerzas de arrastre generadas sean adecuadas para el correcto movimiento del Robot.

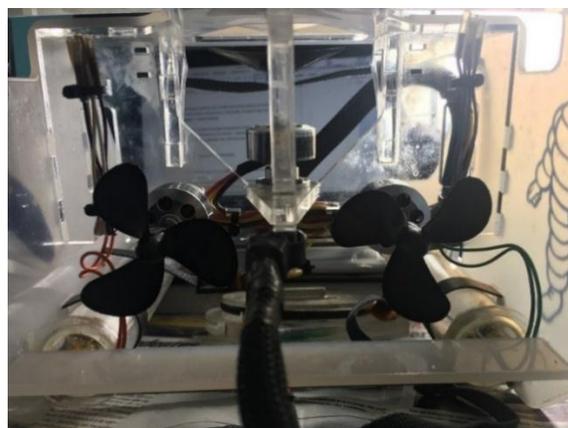


Figura 8. Mecanismos de desplazamiento.

Fuente: Autores

5.1.2. DETERMINAR CARACTERÍSTICAS PARA LA IMPERMEABILIDAD DEL SISTEMA ELÉCTRICO Y ELECTRÓNICO.

En este proceso se utiliza un mecanismo con tapas y empaques de esta manera los sistemas se mantienen en un ambiente cerrado y para el alivio de presión de esta cámara se realiza al usar un embolo colocado en una de las tapas, esta funciona dentro de la cámara en el momento inicial de la inmersión y conformes esta avanza, la parte movable sale de la cámara hasta un tope, generando su descompresión, se puede observar en la Figura 9.



Figura 9. Mecanismo de impermeabilización y alivio de presión.

Fuente: Autores

En esta parte también se tuvo en cuenta las líneas del grosor del acrílico ya que este parámetro es esencial para que pueda soportar la presión, para lo cual el espesor a utilizar es de 5mm, como se muestra en la Figura 10.



Figura 10. Espesor del casco de compresión.

Fuente: Autores

5.1.3. SISTEMA DE DESPLAZAMIENTO Y CONTROL

Selección de los actuadores (motor DC), propelas, cámara, servo para el direccionamiento visual e iluminación para la misma.

Los actuadores que se eligieron para el desarrollo del proyecto son motores BRUSHLESS TURNIGY DST-700 con protección IP 68, estos motores se les agrega antes y después de su uso un lubricante siliconado para prolongar la vida útil, se muestra en la Figura 11.

Las propelas en un inicio se pensó en desarrollarlas en aluminio pero por el grosor, manipulación y costo, se desistió de la idea, para lo cual se buscó una alternativa eficiente y de bajo costo, se compraron tres propelas de diferentes especificaciones técnicas, la primera con un diámetro de 6cm, tres paletas de propulsión y dirección de giro hacia la derecha, la segunda con el mismo diámetro, numero de paletas pero con la dirección de giro hacia la izquierda, para la tercera propela se escogió una con un diámetro de 6cm, 2 paletas de propulsión ya que ofrece mayor fuerza de inmersión del robot y ángulo de giro hacia la izquierda.



Figura 11. Actuadores (Motores Brushless Turnigy).

Fuente: Autores

Para encontrar el Driver controlador, inversor de giro se toma en cuenta el amperaje del motor y el voltaje aceptado, estableciendo este criterio se tomó la decisión de comprar tres controladores de 12 Amperios, las especificaciones se encuentran en la Tabla 4.

Para encontrar una cámara acorde a las necesidades se tomaron varias opciones, de forma que estas puedan ser acopladas al procesador, se tenía dos plataformas principales RaspBerry Pi, BeagleBone Black, las características mínimas de la cámara tenían que ser, resolución mayor a 8 Mpx y grabación en HD, se encontró la Genius WideCam F100, se muestra en la Figura 12, la cual brinda compatibilidad entre entornos y características superior a lo requerido, entre las cuales consta:

- Foto de interpolación de 12 Mpx
- Micrófono estéreo de alta sensibilidad incorporado
- Grabación Full HD de 1920 x 1080p hasta 30 fps
- Lente gran angular de 120 grados



Figura 12. Cámara Genius Wide Cam F100.

Fuente: Autores

La iluminación se adquirió una placa que consta de 4 leds de tipo SMD 5630 de alto brillo y 2 controladores de intensidad de luminosidad, por lo que permite establecer diferentes niveles de iluminación para un mejor enfoque de la cámara, se muestra en la Figura 13.

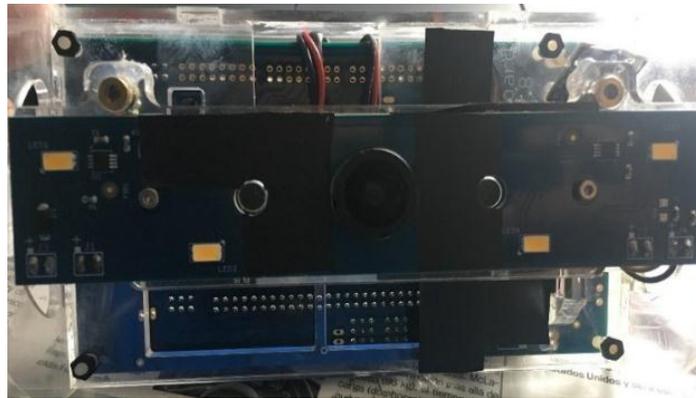


Figura 13. Placa de iluminación.

Fuente: Autores

El movimiento de la cámara se realiza mediante un micro servo motor de 2.6kg de torque, al ser de tamaño pequeño es perfecto para el mecanismo, se puede ver en la Figura 14.



Figura 14. Micro servo motor de 2.6Kg.

Fuente: Autores

Para la Transmisión de datos mayor a 10 metros, se utilizó un kit que permite transferir información, desarmando el contenido de los dispositivos y adaptándolo de forma que los datos enviados desde el codificador ubicado en el robot, lleguen sin pérdidas al decodificador que está conectado al computador, este procedimiento se evidencia en la Figura 15.



Figura 15. Kit de transmisión y recepción de datos.

Fuente: Autores

5.2.PROGRAMACIÓN CONFIGURACIÓN DEL SOFTWARE

Ya que en este procedimiento se tiene un extenso número de comandos, para su entendimiento se necesita conocimientos básicos sobre instalación de sistemas operativos en dispositivos micro-controladores, programación en C, HTML, AVR, acoplamiento entre sistemas LAN. El proceso de programación a grandes rasgos se inicia con el cargado de la imagen básica Debian para BeagleBone Black, configuración de puertos, añadir librerías para utilizar el lenguaje de programación Arduino, añadir librerías de control para utilizar comandos necesarios para desarrollar el código, para cada uno de los parámetros se necesitó programas individuales como para la activación de los ADC, las vinculación de la cámara, movimiento del servo, las respectivas condiciones para la velocidad y control de los actuadores, la iluminación de la cabina, sensor MS5803 se utilizó la librería wire la cual permite utilizar 8bits en comunicación I2C, estableciendo correlación de comandos en envío y recepción de datos, la vinculación de variables para interfaz de los datos y HTML, la configuración de puertos I2C para el sensor e inicializa el I2C del protocolo, configuración de la interfaz gráfica para que muestre los diferentes parámetros en una sola pantalla, configuración de los puertos 8090 para realizar la grabación de la exploración por medio de VLC media Player, cada uno de estos programas tiene sus librerías vinculadas, y además estos son correlacionados entre sí, de esta manera se establece un lenguaje de codificación ordenado y con fácil interpretación de errores.

Una de las partes más importantes es la vinculación de los comandos emitidos a través del teclado, correspondan a la acción que reconozca el microcontrolador, en este programa también se define las imágenes estáticas que se van a encontrar en la cabina de mando, su ubicación, referencia, tamaño, tipo y nombre.

Para la programación de los controladores se utiliza la plataforma llamada Cloud9 la cual permite editar los subprogramas, librería, que tienen diferentes extensiones como: .ccp,.sh,.js.ejs, ya que el programa es compatible con BBB, ayuda al programador a no tener programas indeseados que consumirían recursos de la tarjeta, y concentrarse en los demás archivos de configuración.

Los comandos que se utilizan para el desplazamiento del robot se detallan a continuación en la tabla 7:

N°	Carácter	Acción
1	Flecha arriba	El robot se encamina hacia adelante
2	Flecha abajo	El robot se encamina hacia atrás
3	Flecha derecha	El robot se encamina hacia la derecha
4	Flecha Izquierda	El robot se encamina hacia izquierda
5	Tecla control	El robot realiza la acción de inmersión
6	Tecla shift	El robot realiza la acción de emersión
7	Tecla 1,2,3,4,5	Con estas teclas se define la velocidad de propulsión de los actuadores siendo 1 la más débil y la 5 velocidad máxima
8	Tecla Q	Inclina el sistema de la cámara hacia arriba a través del envío de pulsos al servo motor
9	Tecla A	Inclina el sistema de la cámara hacia el centro a través del envío de pulsos al servo motor
10	Tecla Z	Inclina el sistema de la cámara hacia abajo a través del envío de pulsos al servo motor
11	Tecla O	Se realiza el incremento de la luminosidad otorgada por los leds
12	Tecla P	Se realiza el decremento de la luminosidad otorgada por los leds
13	Tecla I	Encender y apagar la iluminación a la máxima potencia otorgada por los leds
14	Tecla N	Para mantener la profundidad de robot

Tabla 7. Caracteres de control.

Fuente: Autores

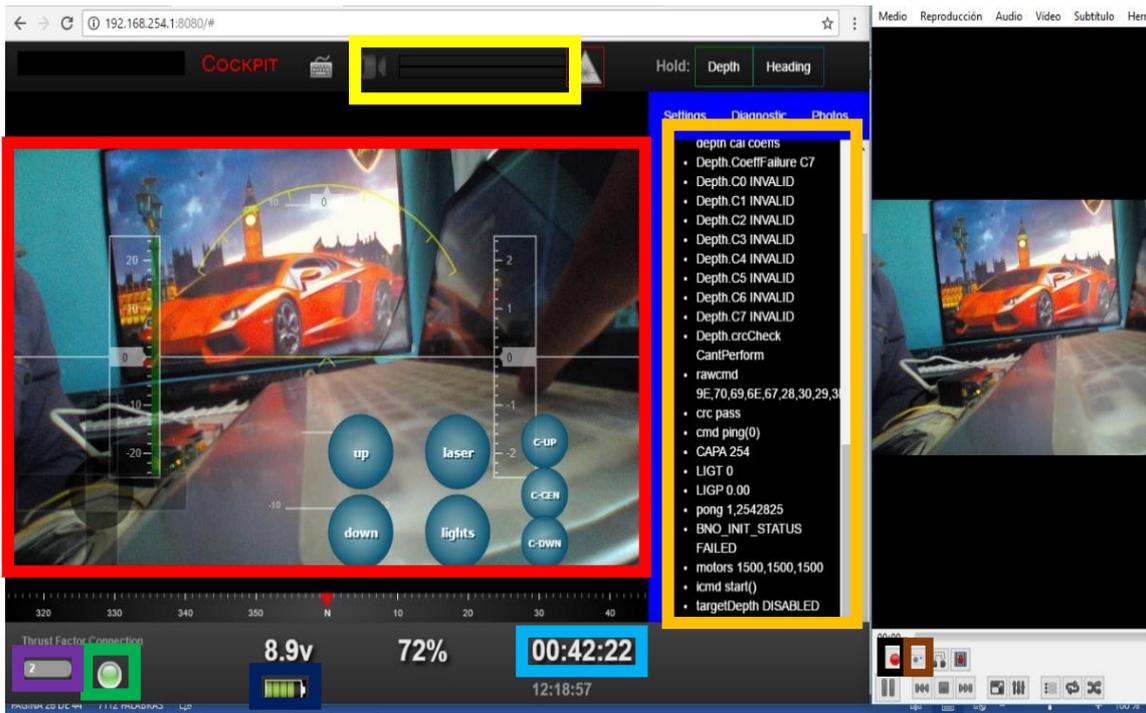


Figura 16. Cabina de control del ROV.

Fuente: Autores

La Figura 16 muestra la posición con colores de los diferentes indicadores que ayudan al correcto control del robot, que se relacionan con las definiciones de la Tabla 8.

Nº	Color	Definición
1	Rojo	Espacio en donde se visualiza la imagen
2	Verde	Se observa la conexión o desconexión de Robot
3	Azul	Estado del batería
4	Amarillo	Estado de luminosidad
5	Naranja	Datos de control de transmisión
6	Purpura	Velocidad de motores
7	Celeste	Tiempo de transmisión
8	Negro	Grabación de video
9	Marrón	Toma de fotografías

Tabla 8. Definiciones de los colores como indicadores dentro de la cabina.

Fuente: Autores

5.3. EXPERIMENTACIÓN

5.3.1. TEST 1

Con este experimento se pretende demostrar el correcto funcionamiento del mecanismo de control del robot, para lo cual se realiza pruebas primero de control de la cámara, fotografía las cuales se muestran en el Anexo 1 y de desplazamiento en la piscina de la universidad Nacional de Chimborazo, con una calificación de niveles que tienen una relación de 5 al correcto funcionamiento y 1 al mal funcionamiento, los resultados serán mostrados en la tabla 13, las evidencias del Test 1 se muestran en el Anexo 2.

5.3.2. PREPARACIÓN PARA LA EXPERIMENTACIÓN DE CAMPO

En el primer día de experimentación se realizó el reconocimiento del lugar, de esta manera se ubicó el lugar donde se realizan las pruebas, como se puede observar en el Anexo 3, este día no se pudo experimentar con el robot ya que las condiciones de accesibilidad al medio y el clima, no prestaban la seguridad necesaria. Para el segundo día de fue necesaria la preparación de un continente frente a los aspectos climáticos y de accesibilidad al medio, las evidencias pueden ser observadas en el Anexo 4. Para empezar el último día de experimentación en la Laguna de Atillo se prepara una excursión con una canoa de remos la cual fue trasladada desde la ciudad de Riobamba, los respectivos chalecos salvavidas, remos, una cuerda de protección sujeta desde la orilla a la canoa. La evidencia se observa en el Anexo 5.

5.3.3. TEST 2

Con este experimento se pretende demostrar el correcto funcionamiento del mecanismo de control del robot, para lo cual se realiza pruebas de desplazamiento en la laguna de Atillo ya que esta permite una profundidad de 9 metros, las evidencias del Test 2 se observa en el Anexo 6.

5.3.4. TEST 3

En este experimento se realizan pruebas de la calidad de imagen que puede proporcionar el Robot frente a las condiciones adversas como la claridad del agua, la turbidez, algas o impurezas del ambiente submarino, y de cómo afecta la utilización de diferentes niveles de iluminación emitidos al ambiente. Los resultados se muestran en las tablas 9 a la tabla 12 y las fotografías desde la Figura 34 a la Figura 37 (Anexo 5). Con respecto a la calificación de la estabilidad se utiliza una escala de malo, regula, bueno, muy bueno, excelente se muestra en la Tabla 14.

Piscina	Iluminación	Estabilidad	Visibilidad
1metro	Nivel 0	5	5
1metro	Nivel 2	5	5
1metro	Nivel 4	5	5
1metro	Nivel 6	5	5
1metro	Nivel 8	5	5

Tabla 9. Calidad de imagen con respecto a la iluminación en estado no sumergido.

Fuente: Autores

Profundidad	Iluminación	Estabilidad	Visibilidad
2metros	Nivel 0	4	4
2metros	Nivel 2	4	4
2metros	Nivel 4	4	5
2metros	Nivel 6	5	5
2metros	Nivel 8	5	5

Tabla 10. Calidad de imagen con respecto a la iluminación y profundidad a 2 metros.

Fuente: Autores

Profundidad	Iluminación	Estabilidad	Visibilidad
7metros	Nivel 0	3	3.3
7metros	Nivel 2	4	3.5
7metros	Nivel 4	4	4
7metros	Nivel 6	4	4.5
7metros	Nivel 8	5	4.5

Tabla 11. Calidad de imagen con respecto a la iluminación y profundidad a 7 metros.

Fuente: Autores

Profundidad	Iluminación	Estabilidad	Visibilidad
9metros	Nivel 0	3	3
9metros	Nivel 2	3	3.2
9metros	Nivel 4	3	3.5
9metros	Nivel 6	4	4
9metros	Nivel 8	4	4.2

Tabla 12. Calidad de imagen con respecto a la iluminación y profundidad a 9 metros.

Fuente: Autores

Tomando en cuenta estos valores se realiza una tabla resumen, mostrando las fluctuaciones de control visibilidad y estabilidad, en cada uno de los ambientes de experimentación, se puede observar en la Figura 17.

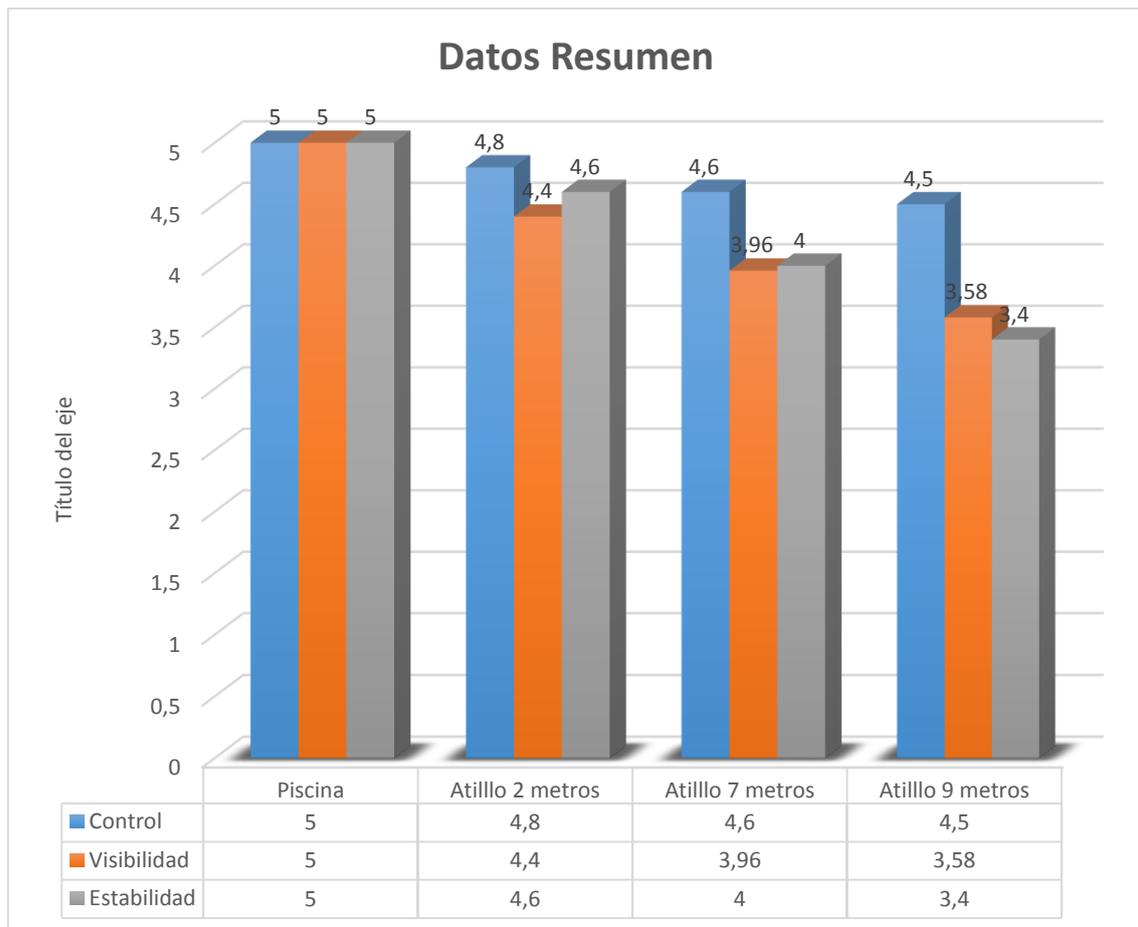


Figura 17. Representación de datos en diagrama de barra.

Fuente: Autores

6. DISCUSIÓN

Esta investigación tuvo como propósito identificar y describir características necesarias para el desarrollo de un Robot para exploración Submarina capaz de transmitir video en tiempo real a una profundidad de diez metros, sobre todo se pretendió buscar una solución efectiva a la exploración subacuática. Se contrasta investigaciones previas, demostrando que el proyecto brinda mejores características de control, transmisión de datos en imagen y video. A continuación, se discutirán los principales hallazgos de este estudio.

En el test uno donde se pretendió realizar un control a todos los mecanismos tanto de actuadores como sensores de navegación, los resultados muestran un correcto e instantáneo funcionamiento, es muy importante para el investigador ir comprobando el trabajo de cada uno de los sistemas, así como la interrelación entre sí.

Las pruebas realizadas en la piscina de la Universidad Nacional de Chimborazo a la profundidad de un metro los mecanismos de navegación, actuadores, sensores, impermeabilidad, cámara y video funcionan de forma correcta y en tiempo real, esta prueba ya en su campo de funcionamiento, muestra que se puede trabajar explorando de una forma continua y ya que el video se guarda en la memoria del computador solucionando el problema de quedarse sin espacio para su guardado.

Se realizaron varias pruebas en las que el tiempo de funcionamiento tiene un rol muy importante, ya que de este periodo se toman los resultados de la investigación, a los cuales se influye la carga de las baterías, y descarga de la mismas, debilitando la velocidad de los motores hasta llegar al punto de solo dejar funcionando la cámara, esto ocurre en el transcurso de una hora y cuarenta minutos de un uso moderado.

De los resultados obtenidos se puede deducir que el sistema de control funciona correctamente, de forma fiable e instantánea lo cual permite controlar el robot con mucha eficacia frente a situaciones de peligro o en un entorno de difícil acceso.

7. CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

Al culminar con la investigación se concluye lo siguiente:

- El prototipo es capaz de desplazarse en diferentes direcciones al frente, atrás, girar en su propio eje tanto a la izquierda, derecha y además de la sumersión, emersión.
- El sistema es capaz de mantener al prototipo estable a una cierta profundidad y permitiendo una completa operación de la cámara.
- El cable umbilical de 50m que conecta el prototipo con el computador permite una inmersión de 9m y la posibilidad de navegar a esta profundidad y desplazarse a mayor distancia.
- La recolección de imágenes y video mediante la cámara Genius Wide Cam F100 a diferentes profundidades presento un desempeño óptimo dando como resultado una calidad de imagen aceptable, de esta forma permitiendo una correcta navegación.
- Los datos obtenidos por el sensor de profundidad representan un porcentaje por debajo del 5% de error.
- El sistema de sellado de la cámara que contiene todo el sistema eléctrico y electrónico garantiza la ausencia de agua en el interior de los mismo, dando fiabilidad y durabilidad al prototipo.
- El marco del prototipo y piezas de soporté para motores se utilizó acrílico transparente y blanco el cual se cortó mediante laser de esta forma se garantiza precisión, simetría y resistencia de las mismas, cada parte se unieron mediante tornillos, super-glue y pegamento Epóxido.

7.2.RECOMENDACIONES

Una vez realizada las correspondientes conclusiones de la investigación se proponen las siguientes recomendaciones:

- El proyecto puede ser mejorado con la implementación de varios extras, como un sistema de recolección de muestras de agua o suelo en profundidad, obteniendo mayores variables para una mejor toma de decisiones con respecto a cualquier ambiente.
- Se recomienda para futuros proyectos cambiar el mecanismo de transmisión de información, de cable bifilar a tecnología inalámbrica, si bien el sistema cumple sus funciones perfectamente sería un valor agregado de investigación cambiarlo.
- Con respecto al control del dispositivo se recomienda implementar un mando más ergonómico que el teclado de la PC, como un emulador de joystick y un sistema de visualización que no sufra deslumbramientos por la luz ambiental.
- El sistema puede ser modificado en hardware y software permitiendo ayudar en la recolección de muestras dentro del ambiente subacuático.
- Con respecto a la experimentación de campo es recomendable la utilización de mecanismo para identificar en que parte del lago se encuentra ubicado el submarino haciendo la investigación.

8. BIBLIOGRAFÍA

- Battery, S. (2000-2016). baterryspace. Obtenido de AA Portable Power Corp: <http://www.batteryspace.com/lifepo4-26650-rechargeable-cell-3-2v-3300-mah-19-8a-rate-10wh---un38-3-passed-ndgr.aspx>
- BeagleBoard. (30 de Abril de 2017). BeagleBoneBlack-Books. Obtenido de Beagleboard - beagleboard.org: <http://elinux.org/Beagleboard:BeagleBoneBlack>
- Carlos, Q. (2013-2014). Estudio y diseño de un ROV Propulsado por aire comprimido. Obtenido de Ingeniería Técnica Naval Especializada en servicio y propulsión del buque: <http://upcommons.upc.edu/bitstream/handle/2099.1/20358/pfc%20Carlos%20Quesada.pdf>
- Chris, G. (Marzo de 2013). Importancia de explorar las profundidades del océano.mexico. Obtenido de Expansión,CNN: <http://expansion.mx/opinion/2013/03/22/opinion-la-importancia-de-explorar-las-profundidades-del-oceano>
- Damian, M. F. (2012). SCRIBD. Obtenido de Diseño de un prototipo ROV (RMOTEL OPERATED VEHICLE) Subacuatico experimental: <https://es.scribd.com/document/346068130/145298-1>
- Donosti, B. (s.f.). ACTUALIDAD/ARTÍCULOS/FÍSICA DEL BUCEO Y SUS LEYES. Obtenido de Física del buceo y sus leyes: <http://www.buceodonosti.com/buceodonosti/de/fisica-del-buceo-y-sus-leyes>
- Emmanuel Nuño Ortega, L. B. (Abril de 2004). Teleoperación. Obtenido de Técnicas, aplicaciones, entorno sensorial y teleoperación inteligente- IOC-DT-P-2004-05: <https://upcommons.upc.edu/bitstream/handle/2117/570/IOC-DT-P-2004-05.pdf?sequence=1>
- Francy, M. R. (2014). Prototipo de robot submarino con la capacidad del seguimiento de trayectorias mediante el procesamiento de imágenes. Obtenido de Universidad Pedagógica Nacional: <http://repositorio.pedagogica.edu.co/xmlui/bitstream/handle/123456789/530/TE-17007.pdf?sequence=1>
- GENIUS, G. (2017). Camara Web. Obtenido de Widedcam F100: <http://us.geniusnet.com/product/widedcam-f100>
- Hector A Moreno, R. S. (Enero - Marzo de 2014). Elsevier. Obtenido de Robótica Submarina: Conceptos, Elementos, Modelado y Control: <http://www.elsevier.es/es->

revista-revista-iberoamericana-automatizada-e-informatica-331-articulo-robotica-submarina-conceptos-elementos-modelado-S1697791213000885?referer=buscador

Hector, A. M. (2013). Modelado, Control y Diseño De Robots Submarino de Estructura Paralela con Impulsores Vectorizados. Obtenido de http://oa.upm.es/16634/1/04_HECTOR_ALONSO_MORENO_AVALOS.pdf

Marie, J. R. (Martes 5 de octubre de 2010). Ifremer. Obtenido de Victor 6000: <http://flotte.ifremer.fr/fleet/Presentation-of-the-fleet/Underwater-systems/VICTOR-6000>

Parker. (s.f.). Catalogo O-Ring. Obtenido de <http://www.elhinel.com.ar/parker/Catalogo%20O-Ring.pdf>

Rafael, G. (Noviembre de 2012). Robótica Submarina como una herramienta clave para la exploración del fondo del Mar, Argentina, Buenos Aires. Obtenido de IEEE: <https://meetings.vtools.ieee.org/m/15295>

Reyes, C. A. (2006). Microcontroladores PIC, Programacion en BASIC. Quito-Ecuador: RISPGRAF.

Robert D. Christ, R. L. (2011). El Manual de ROV. En R. L. Robert D. Christ, A User Guide For Observation Class Remotely Operated Vehicles (págs. 4-5). Butterworth-Heinemann.

Simonk, T. H. (2017). HobbyKing. Obtenido de Afro ESC 12amp BEC UltraLite Multirotor ESC V3 (SimonK firmware).

Turning. (2017). HobbyKing. Obtenido de Turnigy Aerodrive DST-700: https://hobbyking.com/en_us/turnigy-aerodrive-dst-700-brushless-outrunner-motor-700kv.html

Verle, M. (1 de Enero de 2009). MikroEleKtronika. Obtenido de Microcontroladores PIC - Programacion en C con ejemplos: <https://learn.mikroe.com/ebooks/microcontroladorespic/chapter/detalles-importantes/>

ANEXOS

ANEXO 1.

FOTOGRAFÍAS TOMADAS A DIFERENTES ÁNGULOS DE LA CÁMARA.



Figura 18. Fotografía con servo a 0° con respecto al horizonte.

Fuente: Autores



Figura 19. Fotografía con servo a 60° grados con respecto al horizonte

Fuente: Autores



Figura 20. Fotografía con servo a -60° grados con respecto al horizonte.

Fuente: Autores

ANEXO 2.

PRUEBAS DE FUNCIONAMIENTO A BAJA PROFUNDIDAD.

N°	Ítem de Control	Funcionamiento en ambiente	Funcionamiento a 1m de profundidad
1	Adelante	5	5
2	Atrás	5	5
3	Izquierda	5	5
4	Derecha	5	5
5	Inmersión	5	5
6	Emersión	5	5
7	Servo	5	5
8	Cámara	5	5
9	Iluminación	5	5
10	Impermeabilidad	No corresponde	5
11	Cambio de velocidad	5	5
12	Transmisión en Real time	5	5
13	Captura de imágenes	5	5
14	Captura de video	5	5
15	Estabilidad	No corresponde	5

Tabla 13. Funcionamiento de los diferentes sistemas del robot en ambiente y a 1 metro de profundidad.

Fuente: Autores

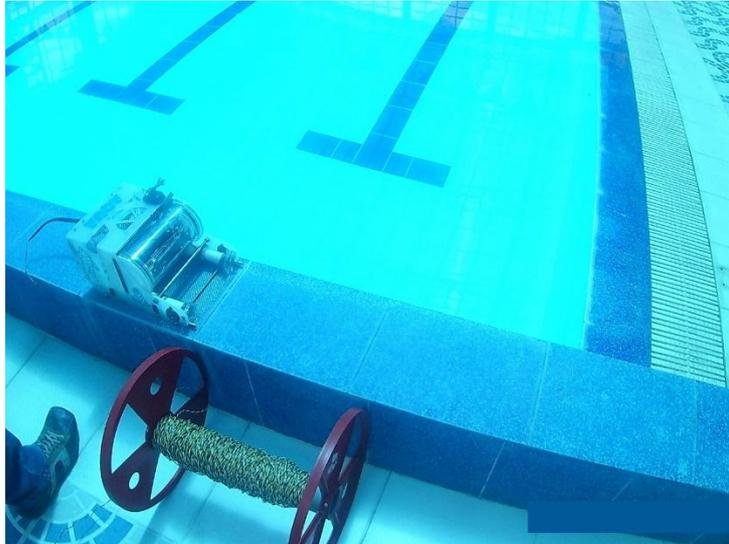


Figura 21. Robot submarino y carrete de Cable de conexión.
Fuente: Autores



Figura 22. Fotografía tomada a 1.7m de profundidad con dirigida hacia arriba.
Fuente: Autores



Figura 23. Fotografía tomada a 1.7m de profundidad con posición normal.
Fuente: Autores

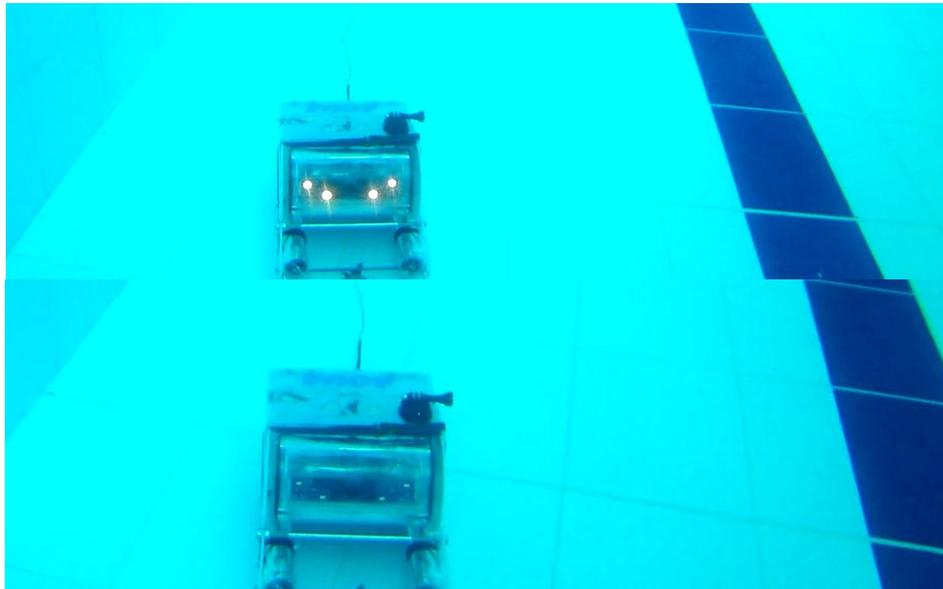


Figura 24. Encendidos del mecanismo de iluminación de ambiente.
Fuente: Autores



Figura 25. Fotografías de la piscina de la UNACH, capturada por el robot Submarino.
Fuente: Autores

ANEXO 3.

RECONOCIMIENTO DEL LUGAR DE INVESTIGACIÓN.



Figura 26. Fotografías del medio de experimentación día 1.
Fuente: Autores



Figura 27. Visitando el lugar para la investigación de Campo.
Fuente: Autores



Figura 28. Identificación del lugar donde se realizará la primera prueba del día 2.
Fuente: Autores



Figura 29. Mapa de coordenadas de las primeras pruebas.
Fuente: Autores



Figura 30. Mapa de coordenadas de las pruebas finales.
Fuente: Autores

ANEXO 4.

EXPERIMENTACIÓN EN CAMPO A BAJA PROFUNDIDAD.



Figura 31. Robot explorando por la superficie.
Fuente: Autores



Figura 32. Fotografía tomada por el robot de exploración de la superficie.
Fuente: Autores

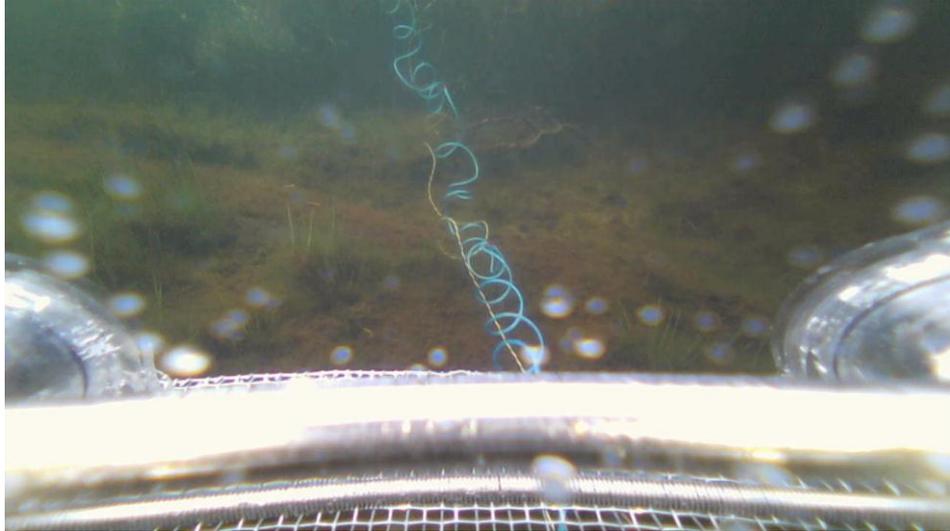


Figura 33. Fotografía tomada por el robot de exploración a una profundidad de 2 metros con la cámara visualizando hacia la parte inferior.
Fuente: Autores

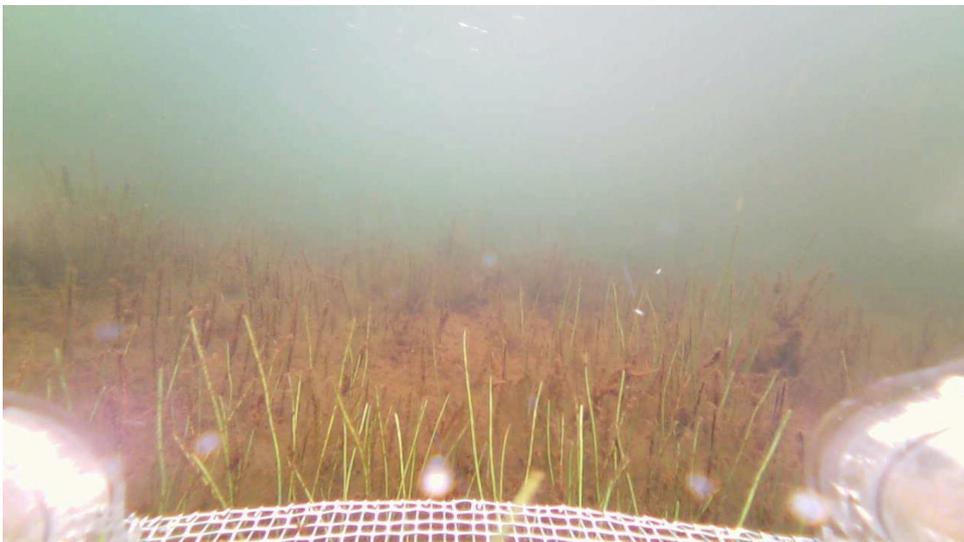


Figura 34. Fotografía tomada por el robot de exploración a una profundidad de 2 metros con la cámara visualizando al frente.
Fuente: Autores

ANEXO 5.

EXPERIMENTACIÓN DE CAMPO A MAYOR PROFUNDIDAD.



Figura 35. Fotografía llevando el vote hasta la orilla de la laguna.

Fuente: Autores



Figura 36. Llevando los materiales para realizar el Test 3.

Fuente: Autores

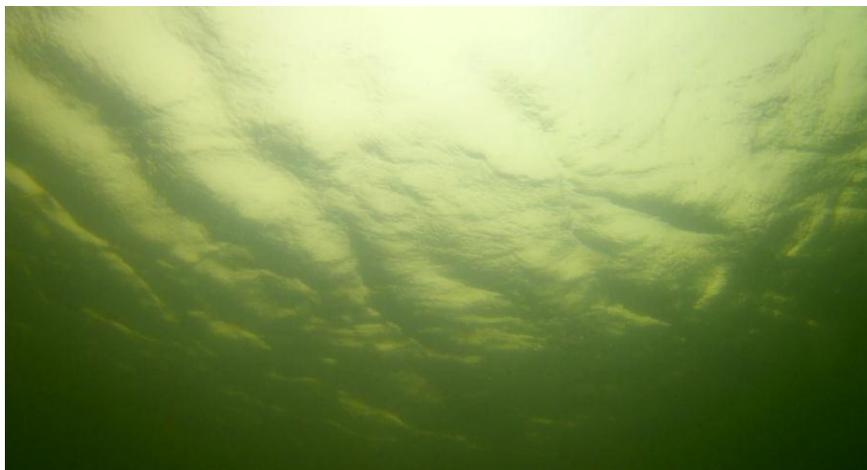


Figura 37. Fotografía tomada por el robot a una profundidad de 2.5 metros.

Fuente: Autores

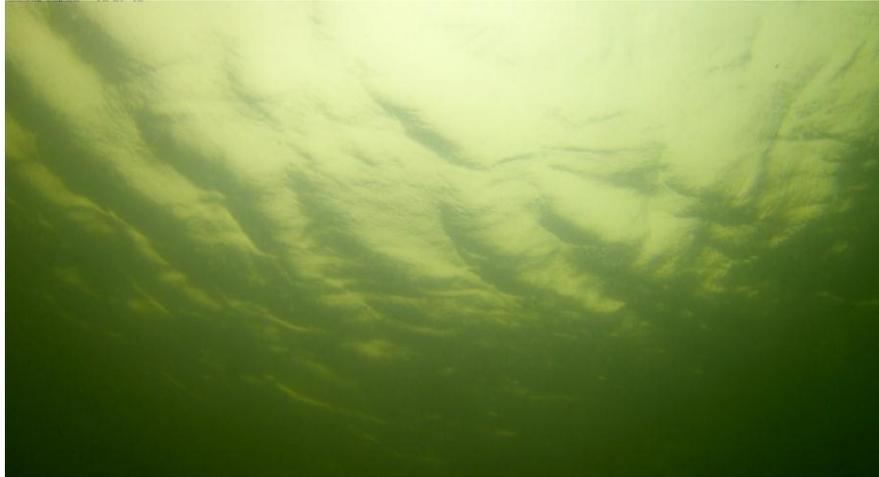


Figura 38. Fotografía tomada por el robot a una profundidad de 5 metros.
Fuente: Autores

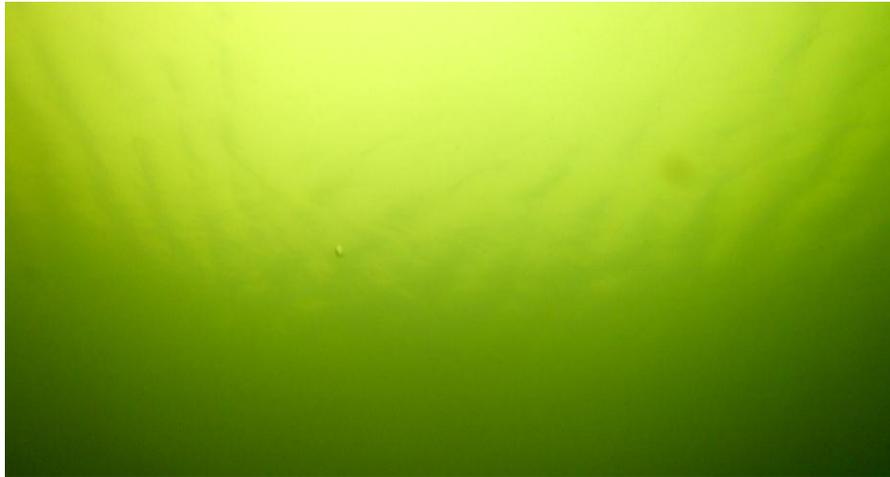


Figura 39. Fotografía tomada por el robot a una profundidad de 7.5 metros.
Fuente: Autores

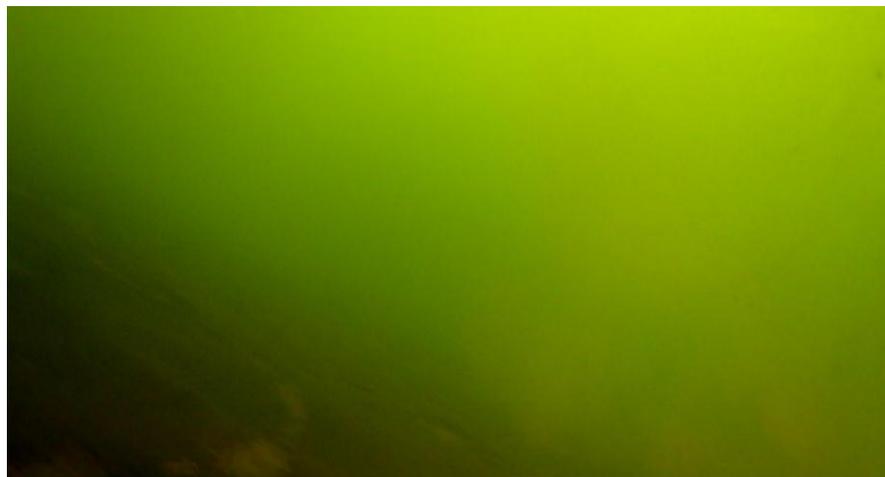


Figura 40. Fotografía tomada por el robot a una profundidad de 9 metros.
Fuente: Autores

ANEXO 6.

PRUEBA DE FUNCIONAMIENTO A UNA ALTA PROFUNDIDAD.

Nº	Ítem de Control	Funcionamiento a 7m	Funcionamiento a 9m
1	Adelante	5	5
2	Atrás	5	5
3	Izquierda	5	5
4	Derecha	5	5
5	Inmersión	5	5
6	Emersión	5	5
7	Servo	5	5
8	Cámara	5	5
9	Iluminación	5	5
10	Impermeabilidad	5	5
11	Cambio de velocidad	5	5
12	Transmisión en Real time	5	5
13	Captura de imágenes	5	5
14	Captura de video	5	5
15	Estabilidad	4	3.7

Tabla 14. Funcionamiento de los diferentes sistemas del robot a 5 metros y a 10 metro de profundidad.

Fuente: Autores



Figura 41. Imágenes tomadas en los niveles de iluminación en estado no sumergido.
Fuente: Autores



Figura 42. Imágenes tomadas en los niveles de iluminación con el robot sumergido.
Fuente: Autores

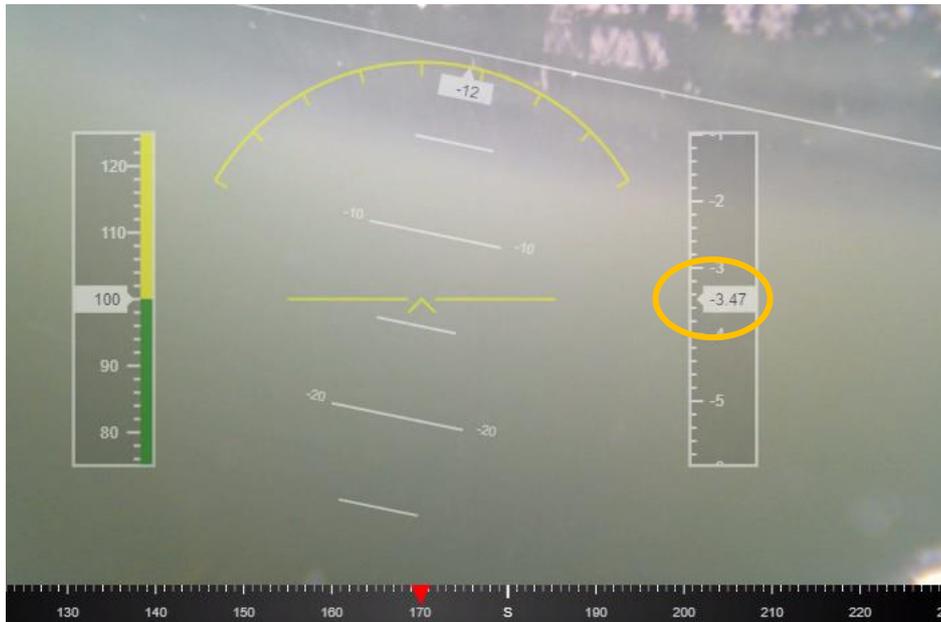


Figura 43. Fotografía del interfaz gráfico con la profundidad a nivel del agua.

Fuente: Autores

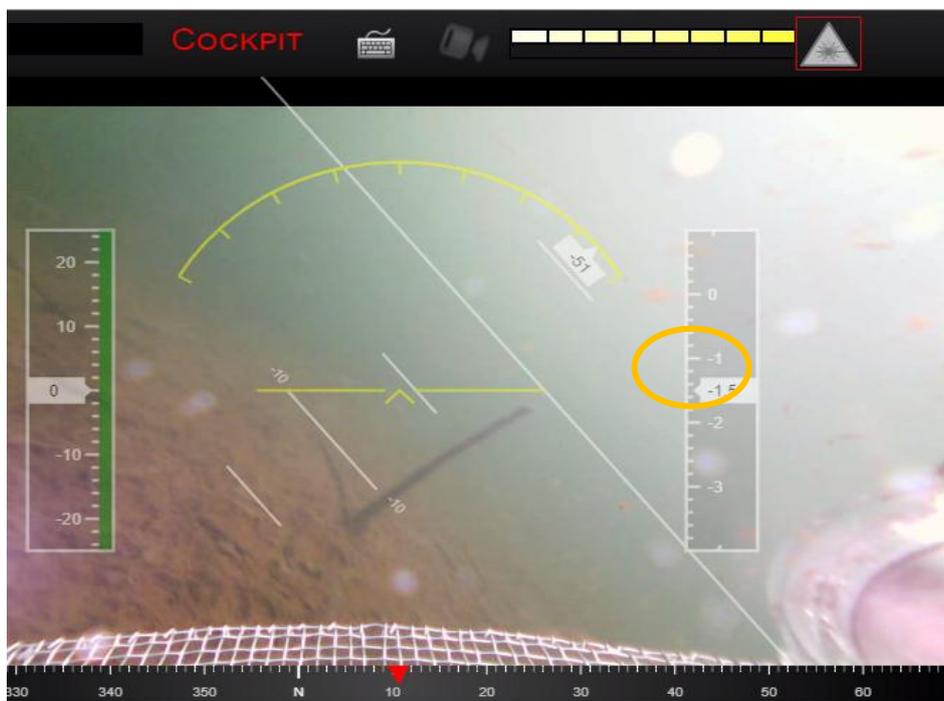


Figura 44. Fotografía del interfaz gráfico con la profundidad a 2 metros.

Fuente: Autores

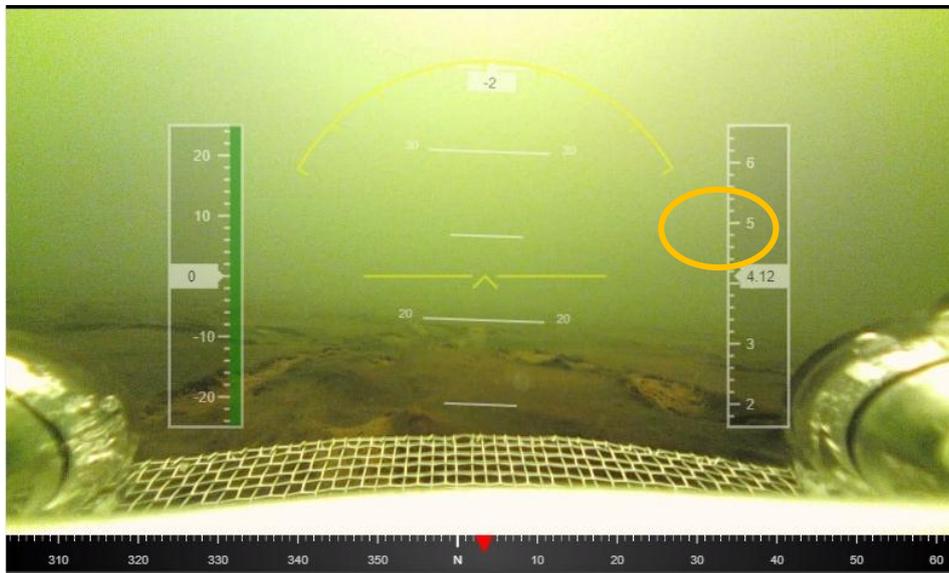


Figura 45. Fotografía del interfaz gráfico con la profundidad a 7 metros.
Fuente: Autores

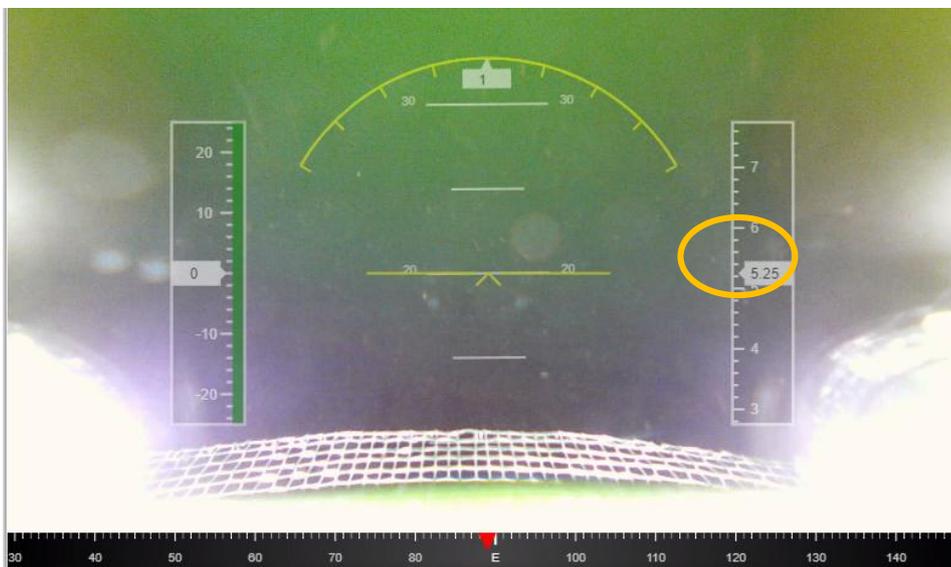


Figura 46. Fotografía del interfaz gráfico con la profundidad a 9 metros.
Fuente: Autores

ANEXO 7.

EVIDENCIAS DE LA CODIFICACIÓN.

ACTIVAR LOS ADC EN AVR

- **openrov\arduino\OpenROV\ CControllerBoard.cpp**

```
#include "AConfig.h"
#if( HAS_OROV_CONTROLLERBOARD_25 )
// Includes
#include <Arduino.h>
#include "NDataManager.h"
#include "CControllerBoard.h"
#include "CTimer.h"
#include "Utility.h"
#include "CPin.h"

// File local variables and methods
namespace
{
    // Define the number of samples to keep track of. The higher the number,
    // the more the readings will be smoothed, but the slower the output will
    // respond to the input. Using a constant rather than a normal variable lets
    // use this value to determine the size of the readings array.
    const int numReadings = 1;
    int readings[numReadings]; // the readings from the analog input

    CTimer time;
    CTimer onesecondtimer;
    CTimer statustime2;

    int index          = 0; // the index of the current reading
    int total          = 0; // the running total
    int average        = 0; // the average
    int tempPin        = A8;

    float celsiusTempRead;

    CPin i2cpower( "i2cpower", I2CPOWER_PIN, CPin::kDigital, CPin::kOutput );

    double GetTemp( void )
    {
        unsigned int wADC;
        double t;

        // The internal temperature has to be used
        // with the internal reference of 1.1V.
        // Channel 8 can not be selected with
        // the analogRead function yet.

        // Set the internal reference and mux.
        ADMUX = ( _BV( REFS1 ) | _BV( REFS0 ) | _BV( MUX3 ) );
        ADCSRA |= _BV( ADEN ); // enable the ADC

        delay( 20 ); // wait for voltages to become stable.
```

```

        ADCSRA |= _BV( ADSC ); // Start the ADC

        // Detect end-of-conversion
        while( bit_is_set( ADCSRA, ADSC ) );

        // Reading register "ADCW" takes care of how to read ADCL and ADCH.
        wADC = ADCW;

        // The offset of 324.31 could be wrong. It is just an indication.
        t = ( wADC - 324.31 ) / 1.22;

        // The returned temperature is in degrees Celcius.
        return ( t );
    }

float mapf( long x, long in_min, long in_max, long out_min, long out_max )
{
    return ( float )( x - in_min ) * ( out_max - out_min ) / ( float )( in_max - in_min ) + out_min;
}

void readTemp()
{
    float analogTempRead = analogRead( tempPin );

    float volt = mapf( analogTempRead * 1.0, 0, 1024, 0, 5.0 ); // change 4: 1024.0, otherwise will calc
integer value!!
    celsiusTempRead = ( volt - .4 ) * 51.28;
}

float readCurrent( int pin )
{
    int voltage = analogRead( pin );

    return mapf( voltage, 0, 1023, 0, 10 );
}

float read20Volts( int pin )
{
    int voltage = analogRead( pin );

    return mapf( voltage, 0, 1023, 0, 20 );
}

float readBrdCurrent( int pin )
{
    int voltage = analogRead( pin );

    return mapf( voltage, 0, 1023, 0, 2 );
}

long readVcc()
{
    // Read 1.1V reference against AVcc
    // set the reference to Vcc and the measurement to the internal 1.1V reference
#ifdef __AVR_ATmega32U4__ || defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
    ADMUX = _BV( REFS0 ) | _BV( MUX4 ) | _BV( MUX3 ) | _BV( MUX2 ) | _BV( MUX1 );
#elif defined( __AVR_ATtiny24__ ) || defined( __AVR_ATtiny44__ ) || defined( __AVR_ATtiny84__ )

```

```

        ADMUX = _BV( MUX5 ) | _BV( MUX0 );
#elif defined( __AVR_ATtiny25__ ) || defined( __AVR_ATtiny45__ ) || defined( __AVR_ATtiny85__ )
        ADMUX = _BV( MUX3 ) | _BV( MUX2 );
#else
        ADMUX = _BV( REFS0 ) | _BV( MUX3 ) | _BV( MUX2 ) | _BV( MUX1 );
#endif

        delay( 2 ); // Wait for Vref to settle
        ADCSRA |= _BV( ADSC ); // Start conversion

        while( bit_is_set( ADCSRA, ADSC ) ); // measuring

        uint8_t low = ADCL; // must read ADCL first - it then locks ADCH
        uint8_t high = ADCH; // unlocks both

        long result = ( high << 8 ) | low;

        result = 1125300L / result; // Calculate Vcc (in mV); 1125300 = 1.1*1023*1000
        return result; // Vcc in millivolts
    }
}

void CControllerBoard::Initialize()
{
    // Reset timers
    time.Reset();
    statustime2.Reset();
    onesecondtimer.Reset();

    i2cpower.Reset();
    i2cpower.Write( 0 );
    delay( 10 );
    i2cpower.Write( 1 );

    // Initialize all the readings to 0:
    for( int thisReading = 0; thisReading < numReadings; ++thisReading )
    {
        readings[ thisReading ] = 0;
    }
}

void CControllerBoard::Update( CCommand& commandIn )
{
    if( time.HasElapsed( 100 ) )
    {
        // subtract the last reading:
        total = total - readings[index];
        // read from the sensor:
        readings[index] = readBrdCurrent( A0 );

        // add the reading to the total:
        total = total + readings[index];
        // advance to the next position in the array:
        index = index + 1;

        // if we're at the end of the array...
        if( index >= numReadings )
            // ...wrap around to the beginning:
    }
}

```

```

        {
            index = 0;
        }

        // calculate the average:
        average = total / numReadings;
    }

    if( onesecondtimer.HasElapsed( 1000 ) )
    {
        readTemp();
        Serial.print( F( "BRDT:" ) );
        Serial.print( celsiusTempRead );
        Serial.print( ';' );
        Serial.print( F( "SC1I:" ) );
        Serial.print( readCurrent( A3 ) );
        Serial.print( ';' );
        Serial.print( F( "SC2I:" ) );
        Serial.print( readCurrent( A2 ) );
        Serial.print( ';' );
        Serial.print( F( "SC3I:" ) );
        Serial.print( readCurrent( A1 ) );
        Serial.print( ';' );
        Serial.print( F( "BRDI:" ) );
        Serial.print( readBrdCurrent( A0 ) );
        Serial.print( ';' );
        Serial.print( F( "BT1I:" ) );
        Serial.print( readCurrent( A6 ) );
        Serial.print( ';' );
        Serial.print( F( "BT2I:" ) );
        Serial.print( readCurrent( A5 ) );
        Serial.print( ';' );
        Serial.print( F( "BRDV:" ) );
        Serial.print( read20Volts( A4 ) );
        Serial.print( ';' );
        Serial.print( F( "AVCC:" ) );
        Serial.print( readVcc() );
        Serial.println( ';' );
    }

    // Update Cape Data voltages and currents
    if( statustime2.HasElapsed( 100 ) )
    {
        NDataManager::m_capeData.VOUT = read20Volts( A4 );

        // #315: deprecated: this is the same thing as BRDI:
        NDataManager::m_capeData.IOOUT = readBrdCurrent( A0 );

        // Total current draw from batteries:
        NDataManager::m_capeData.BTTI = readCurrent( A5 ) + readCurrent( A6 );
        NDataManager::m_capeData.FMEM = util::FreeMemory();
        NDataManager::m_capeData.UTIM = millis();
    }
}

#endif

```

CONFIGURACIÓN DEL MASTER

- `\openrov\arduino\OpenROV\ CI2C.cpp`

```
void CI2C::scan()
{
    uint16_t tempTime = timeOutDelay;
    timeOut( 80 );
    uint8_t totalDevicesFound = 0;
    Serial.println( "log:Scanning for devices...please wait;" );
    Serial.println();

    for( uint8_t s = 0; s <= 0x7F; s++ )
    {
        returnStatus = 0;
        returnStatus = start();

        if( !returnStatus )
        {
            returnStatus = sendAddress( SLA_W( s ) );
        }

        if( returnStatus )
        {
            if( returnStatus == 1 )
            {
                Serial.println( "log:There is a problem with the bus, could not complete scan;" );
                timeOutDelay = tempTime;
                return;
            }
        }
        else
        {
            Serial.print( "log:Found device at address - " );
            Serial.print( " 0x" );
            Serial.print( s, HEX );
            Serial.println( ";" );
            totalDevicesFound++;
        }

        stop();
    }

    if( !totalDevicesFound )
    {
        Serial.println( "log:No devices found;" );
    }

    timeOutDelay = tempTime;
}

uint8_t CI2C::available()
{
    return( bytesAvailable );
}
```

```

uint8_t CI2C::receive()
{
    bufferIndex = totalBytes - bytesAvailable;

    if( !bytesAvailable )
    {
        bufferIndex = 0;
        return( 0 );
    }

    bytesAvailable--;
    return( data[bufferIndex] );
}

```

CONFIGURACIÓN MOVIMIENTO SERVO

- **openrov\arduino\OpenROV\ CServo.cpp**

```

// Interrupt handlers for Arduino
#if defined(_useTimer1)
ISR( TIMER1_COMPA_vect )
{
    handle_interrupts( _timer1, &TCNT1, &OCR1A );
}
#endif

#if defined(_useTimer3)
ISR( TIMER3_COMPA_vect )
{
    handle_interrupts( _timer3, &TCNT3, &OCR3A );
}
#endif

#if defined(_useTimer4)
ISR( TIMER4_COMPA_vect )
{
    handle_interrupts( _timer4, &TCNT4, &OCR4A );
}
#endif

#if defined(_useTimer5)
ISR( TIMER5_COMPA_vect )
{
    handle_interrupts( _timer5, &TCNT5, &OCR5A );
}
#endif

#elif defined WIRING
// Interrupt handlers for Wiring
#if defined(_useTimer1)
void Timer1Service()
{
    handle_interrupts( _timer1, &TCNT1, &OCR1A );
}
#endif
#if defined(_useTimer3)

```

```

void Timer3Service()
{
    handle_interrupts( _timer3, &TCNT3, &OCR3A );
}
#endif
#endif

static void initISR( timer16_Sequence_t timer )
{
    #if defined ( _useTimer1 )

        if( timer == _timer1 )
        {
            TCCR1A = 0;          // normal counting mode
            TCCR1B = _BV( CS11 ); // set prescaler of 8
            TCNT1 = 0;          // clear the timer count
            #if defined( __AVR_ATmega8__ ) || defined( __AVR_ATmega128__ )
                TIFR |= _BV( OCF1A ); // clear any pending interrupts;
                TIMSK |= _BV( OCIE1A ); // enable the output compare interrupt
            #else
                // here if not ATmega8 or ATmega128
                TIFR |= _BV( OCF1A ); // clear any pending interrupts;
                TIMSK1 |= _BV( OCIE1A ); // enable the output compare interrupt
            #endif
        }
        #if defined(WIRING)
            timerAttach( TIMER1OUTCOMPAREA_INT, Timer1Service );
        #endif
    }

    #endif

    #if defined ( _useTimer3 )

        if( timer == _timer3 )
        {
            TCCR3A = 0;          // normal counting mode
            TCCR3B = _BV( CS31 ); // set prescaler of 8
            TCNT3 = 0;          // clear the timer count
            #if defined( __AVR_ATmega128__ )
                TIFR |= _BV( OCF3A ); // clear any pending interrupts;
                ETIMSK |= _BV( OCIE3A ); // enable the output compare interrupt
            #else
                TIFR3 = _BV( OCF3A ); // clear any pending interrupts;
                TIMSK3 = _BV( OCIE3A ); // enable the output compare interrupt
            #endif
        }
        #if defined(WIRING)
            timerAttach( TIMER3OUTCOMPAREA_INT, Timer3Service ); // for Wiring platform only
        #endif
    }

    #endif

    #if defined ( _useTimer4 )

        if( timer == _timer4 )
        {
            TCCR4A = 0;          // normal counting mode

```

```

        TCCR4B = _BV( CS41 ); // set prescaler of 8
        TCNT4 = 0;          // clear the timer count
        TIFR4 = _BV( OCF4A ); // clear any pending interrupts;
        TIMSK4 = _BV( OCIE4A ); // enable the output compare interrupt
    }

#endif

#if defined (_useTimer5)

    if( timer == _timer5 )
    {
        TCCR5A = 0;          // normal counting mode
        TCCR5B = _BV( CS51 ); // set prescaler of 8
        TCNT5 = 0;          // clear the timer count
        TIFR5 = _BV( OCF5A ); // clear any pending interrupts;
        TIMSK5 = _BV( OCIE5A ); // enable the output compare interrupt
    }

#endif

static void finISR( timer16_Sequence_t timer )
{
    //disable use of the given timer
    #if defined WIRING // Wiring
        if( timer == _timer1 )
        {
            #if defined(__AVR_ATmega1281__)||defined(__AVR_ATmega2561__)
                TIMSK1 &= ~_BV( OCIE1A ); // disable timer 1 output compare interrupt
            #else
                TIMSK &= ~_BV( OCIE1A ); // disable timer 1 output compare interrupt
            #endif
            timerDetach( TIMER1OUTCOMPAREA_INT );
        }
        else if( timer == _timer3 )
        {
            #if defined(__AVR_ATmega1281__)||defined(__AVR_ATmega2561__)
                TIMSK3 &= ~_BV( OCIE3A ); // disable the timer3 output compare A interrupt
            #else
                ETIMSK &= ~_BV( OCIE3A ); // disable the timer3 output compare A interrupt
            #endif
            timerDetach( TIMER3OUTCOMPAREA_INT );
        }
    #else
        //For arduino - in future: call here to a currently undefined function to reset the timer
    #endif
}

static boolean isTimerActive( timer16_Sequence_t timer )
{
    // returns true if any servo is active on this timer
    for( uint8_t channel = 0; channel < SERVOS_PER_TIMER; channel++ )
    {
        if( SERVO( timer, channel ).Pin.isActive == true )
        {
            return true;
        }
    }
}

```

```

        }
    }

    return false;
}

/***** end of static functions *****/

CServo::CServo()
{
    if( ServoCount < MAX_SERVOS )
    {
        this->m_index = ServoCount++;          // assign a servo index to this instance
        servos[this->m_index].ticks = usToTicks( DEFAULT_PULSE_WIDTH ); // store default values - 12
Aug 2009
    }
    else
    {
        this->m_index = INVALID_SERVO ; // too many servos
    }
}

uint8_t CServo::Activate( int pin )
{
    return this->Activate( pin, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH );
}

uint8_t CServo::Activate( int pin, int min, int max )
{
    if( this->m_index < MAX_SERVOS )
    {
        // Set servo pin to output
        pinMode( pin, OUTPUT );

        servos[ this->m_index ].Pin.number = pin;

        // todo min/max check: abs(min - MIN_PULSE_WIDTH) / 4 < 128
        this->m_min = ( MIN_PULSE_WIDTH - min ) / 4; //resolution of min/max is 4 uS
        this->m_max = ( MAX_PULSE_WIDTH - max ) / 4;
        // initialize the timer if it has not already been initialized
        timer16_Sequence_t timer = SERVO_INDEX_TO_TIMER( m_index );

        if( isTimerActive( timer ) == false )
        {
            initISR( timer );
        }

        servos[this->m_index].Pin.isActive = true; // this must be set after the check for isTimerActive
    }

    return this->m_index ;
}

void CServo::Detach()
{
    servos[this->m_index].Pin.isActive = false;
    timer16_Sequence_t timer = SERVO_INDEX_TO_TIMER( m_index );
}

```

```

        if( isTimerActive( timer ) == false )
        {
            finISR( timer );
        }
    }

void CServo::Write( int value )
{
    if( value < MIN_PULSE_WIDTH )
    {
        // treat values less than 544 as angles in degrees (valid values in microseconds are handled as
microseconds)
        if( value < 0 )
        {
            value = 0;
        }

        if( value > 180 )
        {
            value = 180;
        }

        value = map( value, 0, 180, SERVO_MIN(), SERVO_MAX() );
    }

    this->WriteMicroseconds( value );
}

```

VELOCIDAD DE MOTORES PROPULSORES

- `\openrov\arduino\OpenROV\ CThrusters_2X1`

```

    if( trg_throttle >= 0 )
    {
        p = 1500 + ( 500.0 / abs( port_motor.m_positiveModifier ) ) * trg_throttle;
        s = p;
    }
    else
    {
        p = 1500 + ( 500.0 / abs( port_motor.m_negativeModifier ) ) * trg_throttle;
        s = p;
    }

    trg_motor_power = s;

    int turn = trg_yaw * 250; //max range due to reverse range

    if( trg_throttle >= 0 )
    {
        int offset = ( abs( turn ) + trg_motor_power ) - 2000;

        if( offset < 0 )
        {
            offset = 0;
        }
    }

```

```

    p = trg_motor_power + turn - offset;
    s = trg_motor_power - turn - offset;
}
else
{
    int offset = 1000 - ( trg_motor_power - abs( turn ) );

    if( offset < 0 )
    {
        offset = 0;
    }

    p = trg_motor_power + turn + offset;
    s = trg_motor_power - turn + offset;
}
}

if( command.Equals( "lift" ) )
{
    if( command.m_arguments[1] >= -100 && command.m_arguments[1] <= 100 )
    {
        trg_lift = command.m_arguments[1] / 100.0;
        v = 1500 + 500 * trg_lift;
    }
}

#ifdef ESCPOWER_PIN
else if( command.Equals( "escp" ) )
{
    escpower.Write( command.m_arguments[1] ); //Turn on the ESCs
    Serial.print( F( "log:escpower=" ) );
    Serial.print( command.m_arguments[1] );
    Serial.println( ';' );
}

#endif
else if( command.Equals( "start" ) )
{
    port_motor.Activate();
    vertical_motor.Activate();
    starboard_motor.Activate();
}
else if( command.Equals( "stop" ) )
{
    p = MOTOR_TARGET_NEUTRAL_US;
    v = MOTOR_TARGET_NEUTRAL_US;
    s = MOTOR_TARGET_NEUTRAL_US;
    // Not sure why the reset does not re-attach the servo.
    //port_motor.stop();
    //vertical_motor.stop();
    //starboard_motor.stop();
}

#ifdef ESCPOWER_PIN
else if( ( command.Equals( "mcal" ) ) && ( canPowerESCs ) )

```

```

{
  Serial.println( F( "log:Motor Calibration Staring;" ) );
  //Experimental. Add calibration code here
  Serial.println( F( "log:Motor Calibration Complete;" ) );
}

#endif

```

CONFIGURA PARA GRABAR EN PROXI, PUERTO DE 8090

- **openrov\image-customization\nginx\ cockpit**

```

server {
  listen 8081;
  server_name _;
  ...

  location /cockpit/camera1 {
    proxy_pass http://127.0.0.1:8090/?action=stream;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_read_timeout 43200000;

    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_set_header X-NginX-Proxy true;
  }

  location / {
    proxy_pass http://127.0.0.1:8080/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_read_timeout 43200000;
    proxy_redirect off;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_set_header X-NginX-Proxy true;
  }
}

```

CAMBIAR IMÁGENES Y COMADOS DEL TECLADO

- **\openrov\cockpit\src\views\ index.ejs**

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

  <!-- The following viewport code is intended to fullscreen the cockpit and prevent scrolling -->

```

```

<meta id="Viewport" name="viewport" width="initial-scale=1, maximum-scale=1, minimum-scale=1, user-
scalable=no">
<script type='text/javascript' src="js/libs/jquery.min.js"></script>
<script type='text/javascript' src="js/libs/jquery-ui.min.js"></script>

<script type="text/javascript">
// this code again helps prevent scrolling and sets viewport for mobile
$(function(){
if( /Android|webOS|iPhone|iPad|Pod|BlackBerry/i.test(navigator.userAgent) ) {
var ww = ( $(window).width() < window.screen.width ) ? $(window).width() : window.screen.width; //get proper
width
var mw = 480; // min width of site
var ratio = ww / mw; //calculate ratio
if( ww < mw){ //smaller than minimum size
$(#Viewport).attr('content', 'initial-scale=' + ratio + ', maximum-scale=' + ratio + ', minimum-scale=' + ratio + ',
user-scalable=yes, width=' + ww);
}else{ //regular size
$(#Viewport).attr('content', 'initial-scale=1.0, maximum-scale=2, minimum-scale=1.0, user-scalable=yes, width=' +
ww);
}
}
});
</script>

<title><%= title %></title>
<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="css/bootstrap-responsive.css" rel="stylesheet">
<link rel="stylesheet" href="themes/OpenROV/css/style.css">
<link href="css/ui-lightness/jquery-ui-1.8.23.custom.css" rel="stylesheet" />
<% for(var i=0; i<styles.length; i++) { %>
<link rel="stylesheet" href="<%= styles[i] %>"><% } %>

</head>

<script type='text/javascript' src="js/libs/eventemitter2.js"></script>
<script type='text/javascript' src="js/libs/bootstrap.min.js"></script>
<script type='text/javascript' src="js/libs/mousetrap.min.js"></script>
<script type='text/javascript' src="js/libs/knockout-2.1.0.js"></script>
<script type='text/javascript' src="js/libs/gamepad.js"></script>
<script type='text/javascript' src="js/utilities.js"></script>
<script type='text/javascript' src="js/frame.js"></script>
<script type='text/javascript' src="js/cockpit.js"></script>
<script type='text/javascript' src="js/libs/db.js"></script>
<script type='text/javascript' src="js/libs/IndexedDBShim.min.js"></script>
<script type='text/javascript' src="/config.js"></script>
<script type='text/javascript' src="/socket.io/socket.io.js"></script>

<script type='text/javascript' src="bower_components/knockout.validation/Dist/knockout.validation.js"></script>
<script type='text/javascript' src="js/knockout-extensions.js"></script>

<% for(var i=0; i<scripts.length; i++) { %>
<script type="text/javascript" src="<%= scripts[i] %>"></script><% } %>

<body>

<div class="navbar navbar-inverse navbar-fixed-top nav-collapse">
<div class="navbar-inner">

```

```

    <a class="brand" href="#"><span class="navbar-
titel">Cockpit</span></a>
    <ul class="nav" id="navtoolbar">
      <li><a href="#"></a></li>
      <li><a href="#" id="servoTilt"></a></li>
    </ul>
    <ul id="menuitems" class="nav pull-right">
      <li><a href="#" id="show-settings">Settings</a></li>
      <li><a href="#" id="show-diagnostic">Diagnostic</a></li>
    </ul>
  </div>
</div>

```

```

<!-- Main ROW -->
<div id="main-row" class="container-fluid full-height">
  <div class="row-fluid full-height">
    <div class="span9 border full-height" id="outter-videocontainer">
      <div id="video-container" class="full-height center pagination-centered" border=1>
        
        <canvas id="video-canvas"/>
      </div>
    </div>
    <div id="rov_status_panel" class="span3 pull-right visible-desktop" style="overflow: auto; height: 100%">
      <div id="buttonPanel" class="controller well well-small"></div>
    </div>
  </div>
</div>
<!-- FOOTER -->
<div id="footer" class="navbar navbar-inverse navbar-fixed-bottom nav-collapse collapse visible-desktop">
  <div class="navbar-inner">
    <div id="footercontent" class="container-fluid" style="height: 100%">
      </div>
    </div>
  </div>
</div>

```

```

<!-- SETTINGS -->
<div class="drop-in-right" id="settings">
  <div class="back-button"></div>
  <h2>Settings</h2>
  <div class="accordion" id="plugin-settings"></div>
</div>
</div>

```

```

<div class="drop-in-right" id="diagnostic">
  <div class="back-button"></div><h3>Diagnostics</h3>
  <div class="settings-block">
    <h4>Pre-Flight Check</h4>
    </div>
    <div class="settings-block">
      <h4>Calibration</h4>
      <a href="#" class="btn" id="callibrate_compass">Calibrate Compass</a>
      <a href="#" class="btn" id="zero_depth">Zero Depth</a>
    </div>
  </div>

```

```

</div>
<div class="hidden" id="keyboardInstructions">

```

```

<p>
  Use the <i>Up</i>, <i>Down</i>, <i>Left</i>, <i>Right</i>, <i>Shift</i> and <i>Ctrl</i> keys to maneuver.
  Use <i>1</i>-<i>5</i> to set power.
</p>
<p>Use <i>Q</i>, <i>A</i>, <i>Z</i> to tilt servo.</p>
<p>Use <i>o</i>, <i>p</i> to increase brightness of lights.</p>
  <p>Use <i>l</i> to toggle callibration lasers.</p>
</div>
</body>
<script type='text/javascript' src='js/view_index.js'></script>
</html>

```

CAMBIAR LA DIRECCIÓN ETHERNET DE LOCALIZACIÓN

- `\openrov\image-customization\customizations\0080-setup-ip-network.sh`

```

#!/bin/bash
set -xe

cat /etc/network/interfaces | grep 'address 192.168.254.1' && exit 0

## fix network

cat >> /etc/network/interfaces << __EOF__

auto eth0:0
iface eth0:0 inet static
name Ethernet alias LAN card
address 192.168.254.1
netmask 255.255.255.0
broadcast 192.168.254.255
network 192.168.254.0

__EOF__

```