

Máster Universitario en Electrónica y Telecomunicación Aplicadas (META)



Trabajo Fin de Máster

Sistema de Control y Comunicaciones de un Perfilador Marino

Autor: Gemma Berciano Rodríguez
Tutor(es): Dr. Aurelio Vega Martínez
Dr. Jorge Cabrera Gámez

Fecha: septiembre 2021

Máster Universitario en Electrónica y Telecomunicación Aplicadas (META)



Trabajo Fin de Máster

Sistema de Control y Comunicaciones de un Perfilador Marino

HOJA DE FIRMAS

Alumna: Gemma Berciano Rodríguez Fdo.:

Tutor: Dr. Aurelio Vega Martínez Fdo.:

Tutor: Dr. Jorge Cabrera Gámez Fdo.:

Fecha: septiembre 2021



ULPGC
Universidad de
Las Palmas de
Gran Canaria

Instituto Universitario de
Microelectrónica Aplicada



Máster Universitario en Electrónica y Telecomunicación Aplicadas (META)



Trabajo Fin de Máster

Sistema de Control y Comunicaciones de un Perfilador Marino

HOJA DE EVALUACIÓN

Calificación:

Presidente

Fdo.:

Secretario

Fdo.:

Vocal

Fdo.:

Fecha: septiembre 2021



t +34 928 451 150
+34 928 451 086
f +34 928 451 083

e: iuma@iuma.ulpgc.es
w: www.iuma.ulpgc.es

Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria

AGRADECIMIENTOS

En estas líneas quiero transmitir mi más sincero agradecimiento a todos aquellos que me han ayudado y apoyado a lo largo de esta etapa de mis estudios.

En primer lugar, me gustaría darle las gracias a mis tutores, Aurelio y Jorge por toda su dedicación y ayuda durante la realización de este trabajo para que pueda haber sido finalizado con éxito. También quiero darle las gracias a Raúl por sus ingeniosas ideas y sus aportaciones, con quien he trabajado codo con codo realizando las pruebas de este proyecto.

En segundo lugar, quiero agradecer a mi familia todo el apoyo recibido y por guiarme a lo largo de toda mi trayectoria académica.

Por último, pero no menos importante, quiero mencionar a mis amigas que siempre han estado apoyándome y a mis compañeros de clase, en especial a Selenia, que me ha acompañado desde que empezamos el grado y se ha convertido para mí en una gran amiga.

RESUMEN

Los gastos y las dificultades logísticas del despliegue de equipos de monitoreo científico en ambientes acuáticos son las mayores limitaciones que existen para realizar dichos registros de datos. Es por ello por lo que se comenzó a crear alternativas menos costosas utilizando placas de microcontroladores de código abierto como la desarrollada en este trabajo.

Este TFM forma parte de un proyecto de investigación llamado “METEROSECO: Hacia una Nueva Metodología para el Estudio de Funciones y Servicios Ecosistémicos de Hábitats Submarinos”, en el que trabajan tres institutos de la Universidad de Las Palmas de Gran Canaria: el Instituto Universitario de Acuicultura Sostenible y Ecosistemas Marinos (IU-ECOQUA), el Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) y el Instituto Universitario de Microelectrónica aplicada (IUMA). El objetivo principal de este proyecto es el estudio de la influencia de la configuración espacio-temporal de dos hábitats marinos en la provisión de funciones y servicios ecosistémicos. Para cumplir este objetivo se ideó un sistema de bajo consumo equipado con sensores y encargado de recoger datos a lo largo de una columna de agua llamado perfilador marino.

Este TFM se encargó de la realización del sistema de control y comunicaciones del perfilador marino. Para ello se contó con la colaboración del Laboratorio de Fabricación de Prototipos y Sistemas Electrónicos del IUMA y el Laboratorio de Robótica del SIANI puesto que disponían de los recursos necesarios para la elaboración del proyecto.

Para comenzar se realizó un estudio sobre los sistemas de monitoreo acuático de bajo consumo existentes y sobre los hábitats de estudio de este proyecto: las praderas de *Cymodocea nodosa* y los fondos de rodolitos. A continuación, se realizó el diseño de una estructura en 3D que soportara la electrónica a utilizar, buscando que fuera sencillamente modificable, de fácil montaje y acceso a los componentes. A la par se estudió y se verificó el funcionamiento de todos los componentes que integran el sistema de manera individual.

Finalmente, se realizó un programa en Arduino IDE en el que el sistema realiza un recorrido vertical, simulando el recorrido que se haría a lo largo de una columna de agua con una parada a mitad de camino para la lectura de sensores y otra al llegar al punto más alto para el guardado de datos y la comunicación vía GMS. Se comprobó su funcionamiento realizando una prueba en el laboratorio.

Abstract

The expenses and logistical difficulties of deploying scientific monitoring equipment in aquatic environments are the major limitations to such data recording. That is why less expensive alternatives were created using open source microcontroller boards such as the one developed in this project.

This Master's project is part of a research project called "METEROSECO: Hacia una Nueva Metodología para el Estudio de Funciones y Servicios Ecosistémicos de Hábitats Submarinos", on which three institutes of the University of Las Palmas de Gran Canaria are working: the University Institute of Sustainable Aquaculture and Marine Ecosystems (IU-ECOQUA), the University Institute of Intelligent Systems and Numerical Applications in Engineering (IUSIANI) and the University Institute of Applied Microelectronics (IUMA). The main objective of this project is to study the influence of the spatial-temporal configuration of two marine habitats on the provision of ecosystem functions and services. To meet this objective, a low-power system, called marine profiler, equipped with sensors and in charge of collecting data along a water column was designed.

This Master's project was in charge of the execution of the control and communications system of the marine profiler. The Manufacturing Laboratory of IUMA Electronic Systems and Prototypes, and the SIANI Robotics Laboratory collaborated in this project since they had the necessary resources for its development.

To begin with, a study on existing low power consumption aquatic monitoring systems and the habitats to be studied in this project was carried out: the Cymodocea nodosa meadows and rhodolith beds. Next, a 3D structure was designed to support the associated electronics, making it simple to modify, and easy to assemble and access the components. At the same time, the functioning of each component that makes up the system was studied and verified individually.

Finally, a program was developed in Arduino IDE. In it the system carries out a vertical path, simulating the one that would be made along a column of water with a halfway stop for sensor reading and another stop when the highest point for data storage and communication via GSM is reached. Its functioning was verified by performing a test in the laboratory.

ÍNDICE

1	INTRODUCCIÓN	1
1.1	ANTECEDENTES	1
1.2	OBJETO DEL TRABAJO	2
1.3	PETICIONARIO.....	2
2	ESTADO DEL ARTE	4
2.1	CYMODOCEA NODOSA	4
2.2	FONDOS DE RODOLITOS	6
2.3	SISTEMAS DE MONITORIZACIÓN EXISTENTES	7
2.4	PERFILADOR MARINO	9
2.5	MICROCONTROLADOR.....	10
2.6	SENSORES Y ACTUADORES	11
2.6.1	<i>Acelerómetro</i>	12
2.6.2	<i>Sensor de micropresión</i>	12
2.6.3	<i>Sensor magnético de efecto Hall</i>	13
2.6.4	<i>Sensor de luz ambiental</i>	13
2.6.5	<i>Sensor de corriente</i>	13
2.6.6	<i>Contador de corriente o Coulomb Counter</i>	14
2.6.7	<i>Sensor de temperatura y de presión exteriores</i>	14
2.6.8	<i>GNSS</i>	15
2.6.9	<i>Pantalla OLED</i>	15
2.6.10	<i>LED RGB</i>	16
2.6.11	<i>Motor</i>	16
2.6.12	<i>Minipush button power Switch</i>	16
2.6.13	<i>Power Switch</i>	17
2.6.14	<i>Baterías</i>	17
2.7	PROTOCOLO DE COMUNICACIÓN FTP	18
2.8	ENTORNO DE PROGRAMACIÓN ARDUINO IDE.....	20
2.9	ENTORNO DE DISEÑO AUTODESK INVENTOR PROFESSIONAL 2022	23
2.10	PRUSASLICER	24
3	SOLUCIÓN HARDWARE ADOPTADA	27
3.1	PARTE SUPERIOR.....	28
3.2	PARTE INFERIOR	32
4	SOLUCIÓN SOFTWARE ADOPTADA.....	37
4.1	FUNCIONES:.....	37
4.1.1	<i>Acelerómetro:</i>	37
4.1.2	<i>Sensor de micropresión:</i>	38
4.1.3	<i>Sensor magnético de efecto Hall</i>	39
4.1.4	<i>Sensor de luz ambiental:</i>	39
4.1.5	<i>Sensor de corriente:</i>	40
4.1.6	<i>Sensor de temperatura:</i>	41
4.1.7	<i>Sensor de presión:</i>	42
4.1.8	<i>GNSS:</i>	42
4.1.9	<i>Pantalla OLED:</i>	43
4.1.10	<i>LED RGB:</i>	43
4.1.11	<i>Motor:</i>	44
4.1.12	<i>Minipush button power switch:</i>	46

Sistema de Control y Comunicaciones de un Perfilador Marino

4.1.13	<i>Coulomb Counter</i> :	46
4.1.14	<i>Power switch</i> :	46
4.1.15	<i>Tarjeta SD</i> :	47
4.2	COMUNICACIONES	47
4.3	PROGRAMA PRINCIPAL	48
5	PRUEBAS REALIZADAS	55
5.1	PRIMERA PRUEBA DE INMERSIÓN	55
5.2	PRUEBAS DE FUNCIONAMIENTO INDIVIDUALES	56
5.2.1	<i>Consumo de corriente del motor</i>	56
5.2.2	<i>Sensores de efecto Hall</i>	57
5.2.3	<i>Funcionamiento del GNSS</i>	58
5.2.4	<i>Consumo de la batería de la electrónica</i>	58
5.3	PRUEBA DEL SISTEMA DE COMUNICACIONES	60
5.4	PRIMERA PRUEBA DE FUNCIONAMIENTO COMPLETA	61
6	PRESUPUESTO	62
7	CONCLUSIONES Y TRABAJOS FUTUROS	64
7.1	CONCLUSIONES	64
7.2	TRABAJOS FUTUROS	64
8	BIBLIOGRAFÍA	66

ÍNDICE DE FIGURAS

Figura 1	Diseño del primer prototipo de perfilador marino	2
Figura 2	Pradera de Cymodocea nodosa [12]	4
Figura 3	Distribución de praderas de Cymodocea nodosa en las islas Canaria. [12].....	5
Figura 4	Banco de peces asociado a una pradera de Cymodocea nodosa [12]	5
Figura 5	Fondos de rodolitos.....	6
Figura 6	Rodolitos de la playa Caleta del Barco en Fuerteventura [16].....	7
Figura 7	Diseño del prototipo del proyecto Cave Pearl (1) [18]	8
Figura 8	Diseño del prototipo del proyecto Cave Pearl (2) [18]	8
Figura 9	Perfilador CTD anclado y de elevación libre [6]	8
Figura 10	Arduino MKR GSM 1400 [19].....	10
Figura 11	Módulos principales del Arduino MKR GSM 1400	10
Figura 12	Cara trasera del Arduino MKR GSM 1400	10
Figura 13	Antena.....	11
Figura 14	Arduino MKR MEM Shield [23].....	11
Figura 15	Cable y conector qwiic	12
Figura 16	Acelerómetro MMA8452Q [24]	12
Figura 17	Sensor de micropresión [25]	12
Figura 18	Sensor magnético de efecto Hall [26]	13
Figura 19	Sensor de luz ambiental [27].....	13
Figura 20	Sensor de corriente [28]	13
Figura 21	Coulomb Counter [29]	14
Figura 22	Sensor de temperatura TSY501 [31]	14
Figura 23	Sensor de presión MS5837 [32].....	14
Figura 24	I2C Bus Splitter [33].....	15
Figura 25	Placa GNSS [34].....	15
Figura 26	Pantalla OLED [35]	15
Figura 27	LED RGB.	16
Figura 28	Motor BLDC.....	16
Figura 29	Minipush button Power Switch [36]	16
Figura 30	Power Switch [37]	17
Figura 31	Batería del motor	17
Figura 32	Batería de la electrónica.....	18
Figura 33	Protocolo FTP modo activo [39].....	19
Figura 34	Protocolo FTP modo pasivo [39]	19
Figura 35	Icono y ventana de inicialización del Arduino IDE	20
Figura 36	Vista inicial del entorno	20
Figura 37	Añadir una librería con Arduino IDE	21
Figura 38	Gestor de librerías del Arduino IDE	22
Figura 39	Acceso a los ejemplos del Arduino IDE	23
Figura 40	Icono y ventana de inicialización del Autodesk Inventor Professional 2022	24
Figura 41	Vista inicial del entorno Autodesk Inventor Professional 2022	24
Figura 42	Icono y Ventana de inicialización del PrusaSlicer	25
Figura 43	Vista inicial del entorno PrusaSlicer.....	25
Figura 44	Vista de laminado del PrusaSlicer (1).....	26
Figura 45	Vista de laminado del PrusaSlicer (2).....	26
Figura 46	Perfilador marino montado	27

Sistema de Control y Comunicaciones de un Perfilador Marino

Figura 47	Espacio entre la parte superior y la parte inferior del perfilador marino	27
Figura 48	Tapa de extremo de domo.....	28
Figura 49	Brida con sellos de junta tórica.....	28
Figura 50	Tubo acrílico de 15 cm	28
Figura 51	Junta tórica.....	28
Figura 52	Tapa de aluminio de 5 orificios	29
Figura 53	Pieza base de la estructura interna de la parte superior del perfilador	29
Figura 54	Pieza intermedia de la estructura interna de la parte superior del perfilador	29
Figura 55	Piezas complementarias de la estructura interna de la parte superior del perfilador ..	30
Figura 56	Pieza superior de la estructura interna de la parte superior del perfilador	30
Figura 57	Estructura interna de la parte superior del perfilador montada	31
Figura 58	Vista lateral de la estructura de la parte superior del perfilador.	31
Figura 59	Placa del LED RGB.....	31
Figura 60	Placa intermedia del motor	32
Figura 61	Conexionado (1) bus I2C	32
Figura 62	Conexionado (2)	32
Figura 63	Pieza resultante del módulo inferior del perfilador marino	33
Figura 64	Diseño de la pieza del módulo inferior de perfilador marino	33
Figura 65	Diseño del compartimento del motor.....	34
Figura 66	Diseño de la tapa del compartimento del motor	34
Figura 67	Anilla superior del perfilador marino	34
Figura 68	Anilla intermedia del perfilador marino	35
Figura 69	Cono de la parte inferior del perfilador marino	35
Figura 70	Anclaje de peso muerto.....	35
Figura 71	Diseño final de la estructura del perfilador marino.....	36
Figura 72	Función del acelerómetro.....	38
Figura 73	Función del sensor de micropresión	38
Figura 74	Función del sensor de efecto Hall.....	39
Figura 75	Función del sensor de luz ambiental.....	39
Figura 76	Función del sensor de corriente	40
Figura 77	Función BubbleSortAsc.....	41
Figura 78	Función MedianCalculation.....	41
Figura 79	Función del sensor de temperatura	42
Figura 80	Función del sensor de presión exterior	42
Figura 81	Función del GNSS	43
Figura 82	Función de la pantalla OLED	43
Figura 83	Funciones del LED RGB	44
Figura 84	Función del motor (1)	45
Figura 85	Función del motor (2)	45
Figura 86	Función del Minipush button Power Switch.....	46
Figura 87	Función del coulomb counter	46
Figura 88	Ejemplo de uso de las funciones de la tarjeta SD	47
Figura 89	Función para las comunicaciones FTP	47
Figura 90	Función test de las comunicaciones FTP	48
Figura 91	Diagrama de flujo del setup	48
Figura 92	Diagrama de flujo del loop	49
Figura 93	Diagrama de flujo del ciclo de trabajo.....	50

Sistema de Control y Comunicaciones de un Perfilador Marino

Figura 94 Diagrama de flujo de la función que se realiza cuando la variable hallEffectSensor toma el valor true.....	50
Figura 95 Función loop (1).....	51
Figura 96 Función loop (2).....	52
Figura 97 Función loop (3).....	52
Figura 98 Función loop (4).....	53
Figura 99 Función loop (5).....	53
Figura 100 Función loop (6).....	54
Figura 101 Perfilador marino completo para la primera prueba de inmersión.....	55
Figura 102 Pieza que guarda el motor.....	55
Figura 103 Primera prueba de inmersión	56
Figura 104 Prueba del motor.....	57
Figura 105 Prueba de sensores de efecto Hall.....	57
Figura 106 Prueba de la placa GNSS	58
Figura 107 Diagrama de flujo del ciclo de trabajo de la prueba del consumo de batería.....	59
Figura 108 Prueba del sistema de comunicaciones utilizando FileZilla	60
Figura 109 Mensajes del puerto serie del Arduino - prueba FTP.....	60
Figura 110 Comprobación de ficheros mediante FileZilla - prueba FTP.....	60
Figura 111 Prueba de funcionamiento completa del perfilador marino	61

ÍNDICE DE TABLAS

Tabla I Datos de consumo de las pruebas	59
Tabla II Presupuesto de recursos hardware	63
Tabla III Presupuesto de recursos software.....	63
Tabla IV Presupuesto recursos humanos	63
Tabla V Presupuesto total	63

Lista de acrónimos

2D	2 Dimensions
3D	3 Dimensions
BLDC	Brushless Direct Current
CAD	Computer Aided Design
CTD	Conductivity Temperature Depth
DIY	Do It Yourself
FTP	File Transfer Protocol
GSM	Global System for Mobile
GNSS	Global Navigation Satellite System
HSPA	High Speed Packet Access
I ² C	Inter Integrated Circuits.
IoT	Internet Of Things
IDE	Entorno de Desarrollo Integrado.
IU- ECOAQUA	Instituto Universitario de Acuicultura Sostenible y Ecosistemas Marinos
IUMA	Instituto Universitario de Microelectrónica Aplicada.
IUSIANI	Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería
LED	Light Emitting Diode
META	Máster en Electrónica y Telecomunicación Aplicadas
OLED	Organic light-Emitting Diode
PCB	Printed Circuit Board
PETG	Polyethylene Terephthalate (PET) with Glicol
RGB	Red Green Blue
RTC	Real Time Clock
SCP	Secure Copy Protocol
SD	Secure Digital
SFTP	SSH File Transfer Protocol
SSH	Secure SHell
TCP	Transmission Control Protocol

Sistema de Control y Comunicaciones de un Perfilador Marino

TFM	Trabajo Fin de Máster
ULPGC	Universidad de Las Palmas de Gran Canaria.
USB	Universal Serial Bus
UMTS	Universal Mobile Telecommunications System

1 INTRODUCCIÓN

1.1 ANTECEDENTES

Una buena gestión de un ecosistema implica la comprensión de su funcionamiento para identificar los procesos naturales y humanos que causan cambios en las comunidades biológicas. En los ecosistemas marinos quedan por evaluar numerosos sistemas ecosistémicos para gran diversidad de ambientes y contextos biogeográficos a nivel de todos los océanos del planeta. Es por ello por lo que se comenzó un proyecto de investigación llamado “*Hacia una Nueva Metodología para el Estudio de Funciones y Servicios Ecosistémicos de Hábitats Submarinos*”, cuyo acrónimo es *METOSERECO*. Este proyecto se basa en la cooperación de tres institutos de la Universidad de Las Palmas de Gran Canaria (ULPGC): el Instituto Universitario de Acuicultura Sostenible y Ecosistemas Marinos (IU- ECOAQUA), el Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) y el Instituto Universitario de Microelectrónica aplicada (IUMA). Su objetivo es estudiar la influencia de la configuración espacio-temporal de dos hábitats marinos en la provisión de funciones y servicios ecosistémicos.

Con la estimación de estos servicios se podría definir y cuantificar las funciones ecosistémicas, las cuales podrían ser convertidas en beneficios a través de valor económico. De esta forma se podría conectar la naturaleza y el bienestar humano para poder realizar una mejor gestión ambiental de los hábitats y para que la sociedad pueda llegar a entender la necesidad de gestionar adecuadamente la naturaleza y sus recursos. Por otro lado, conociendo esa configuración espacio-temporal de los hábitats marinos se podría mejorar su conservación reduciendo las perturbaciones de origen antropogénico que amenazan su estructura y su función [1].

Para el estudio de la distribución espacio-temporal se va a desarrollar un perfilador marino que se desplazará verticalmente desde el fondo hasta la superficie, equipado con sensores que proporcionarán el muestreo temporal y espacial de los hábitats a estudiar. Este perfilador marino será un sistema abierto utilizado para el monitoreo ambiental acuático de bajo consumo y bajo costo que sustituirá a los sensores comerciales de alto costo [2]–[5]. Este TFM se centrará en la tarea del diseño e implementación del sistema de control y comunicaciones del perfilador marino con capacidad de obtener datos en toda una columna de agua [6]. Se partirá de un primer prototipo mostrado en la figura 1, el cual se evaluará y se rediseñará. Los hábitats a estudiar son las praderas de *Cymodocea nodosa* y los fondos de rodolitos los cuales varían estructural y funcionalmente en el espacio, tanto en la horizontal como en la vertical.

El trabajo experimental de este proyecto se realizará en la isla de Gran Canaria, concretamente en la Bahía de Gando, en donde hay praderas de la fanerógama marina *Cymodocea nodosa* entre los 8 y 20 metros de profundidad y fondos de rodolitos entre los 20 y 80 metros de profundidad.

El sistema de comunicaciones dispondrá de un canal bidireccional el cual tendrá un servidor propio con IP fija conectado a internet con el que se comunicará durante las salidas a superficie del perfilador. Las comunicaciones se gestionarán a través de un módem del fabricante U-blox integrado en la placa que se utilizará para controlar el sistema de control [7]. Para una mayor seguridad el sistema de comunicaciones podría disponer de varios servidores alternativos en caso de que el que se fuera a usar fallara.

Sistema de Control y Comunicaciones de un Perfilador Marino

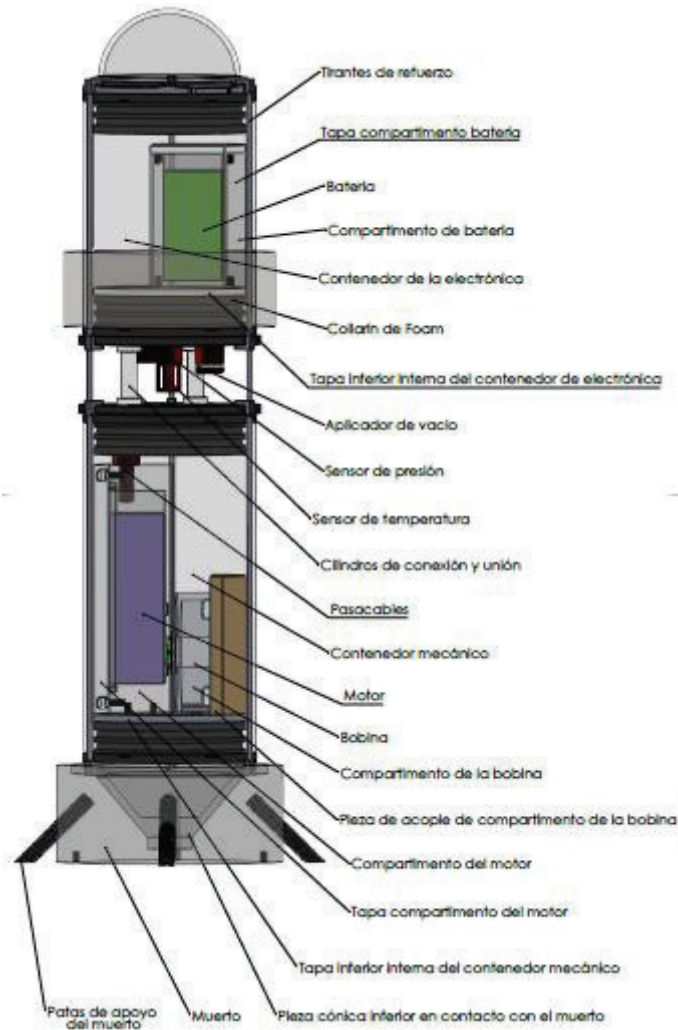


Figura 1 Diseño del primer prototipo de perfilador marino

1.2 OBJETO DEL TRABAJO

El objetivo general de este proyecto es desarrollar el sistema de control y comunicaciones de un perfilador marino que sea capaz de obtener datos en toda una columna de agua.

- Se evaluará el diseño actual del prototipo y se rediseñará.
- Se realizará el diseño e implementación del sistema de control y de comunicaciones del perfilador, realizando las pruebas necesarias para verificar su correcto funcionamiento.

Se pretende conseguir un diseño robusto, de bajo coste, abierto, es decir, con capacidad para integrar diferentes baterías de sensores, y con una autonomía energética mínima de varios días.

1.3 PETICIONARIO

El peticionario de este TFM es el Instituto Universitario de Microelectrónica Aplicada (IUMA) de la UPLGC con el fin de evaluar a asignatura Trabajo de Fin de Máster incluida en el plan de estudios del Máster en Electrónica y Telecomunicación Aplicadas (META).

Sistema de Control y Comunicaciones de un Perfilador Marino

Además, este TFM contará con el Laboratorio de Fabricación de Prototipos y Sistemas Electrónicos del IUMA y el Laboratorio de Robótica del SIANI puesto que disponen de los recursos necesarios para la realización de este trabajo.

2 ESTADO DEL ARTE

En este apartado se describirán los hábitats de estudio, los sistemas de monitorización existentes, y el sistema a desarrollar en este proyecto. Además, se mostrará y detallará información sobre los componentes hardware y el software utilizados.

2.1 CYMODOCEA NODOSA

La *Cymodocea nodosa* es la segunda planta fanerógama marina más importante en el Mediterráneo por su envergadura y por la extensión que ocupan sus formaciones. Esta planta, que forma praderas submarinas en fondos arenosos y bien iluminados del litoral canario, es conocida como “sebadales” o “seba” y está recogida en la Ley 4/2010, de 4 de junio, del Catálogo Canario de Especies Protegidas como especies de “interés para los ecosistemas canarios” [8], [9]. Las especies recogidas en esta sección son aquellas que son merecedoras de atención particular por su importancia ecológica en espacios de la Red Canaria de Espacios Naturales Protegidos o de la Red Natura 2000, a pesar de no considerarse una especie amenazada [10].



Figura 2 Pradera de Cymodocea nodosa [12]

Las praderas de *Cymodocea* (figura 2) crecen en Canarias sobre sustratos arenosos o arenoso-fangosos y en ambientes relativamente abrigados de los vientos y las corrientes dominantes. Los sebadales canarios son ecosistemas marinos bentónicos bastante frágiles, ya que no se dispone de grandes extensiones de esta pradera si se compara con las existentes de otras zonas del planeta. Estas praderas se localizan, principalmente, en las bahías protegidas de la zona este, sureste, sur y suroeste de casi todas las islas, aunque el hecho de crecer sobre fondos blandos hace que estas plantas sean más abundantes en las islas orientales, donde existen plataformas marinas más amplias y una mayor disposición de sustratos arenosos, mientras que son más escasas en las occidentales, salvo en Tenerife donde existen buenas representaciones de praderas (figura 3) [11].

Sistema de Control y Comunicaciones de un Perfilador Marino



Figura 3 Distribución de praderas de *Cymodocea nodosa* en las islas Canaria. [12].

El crecimiento de *Cymodocea nodosa* en los fondos arenosos permite el desarrollo de un ecosistema con características peculiares que cumple diversas funciones: contribuye a la fijación y estabilización de sedimentos arenosos; interviene activamente en el ciclo de los elementos, fijando el carbono y el nitrógeno de los sedimentos; y permite el desarrollo de comunidades asociadas, tanto de algas epífitas como acompañantes [11]. Esas plantas también sirven de hábitat albergando una gran cantidad de especies de animales invertebrados y formando un lugar importante para la cría de peces, algunas de cuyas especies tienen gran importancia comercial, dando lugar a comunidades mucho más ricas que los fondos arenosos desnudos (figura 4) [11].



Figura 4 Banco de peces asociado a una pradera de *Cymodocea nodosa* [12]

Las praderas de *Cymodocea nodosa* tiene un gran interés ecológico tanto por el incremento de diversidad animal que conlleva su presencia, como porque se instalan sobre fondos blandos, estabilizándolos [13].

Como ya se ha mencionado, estas praderas son ecosistemas muy importantes desde el punto de vista ecológico y económico, ya que sirven de refugio a un gran número de especies de interés comercial. Sin embargo, la fuerte presión antrópica a la que se ven sometidas ha hecho disminuir notablemente la presencia de estos ecosistemas en nuestras costas [8]. En general, los impactos antrópicos, directos o indirectos, son los que ocurren con más frecuencia, siendo responsables de la mayoría de los casos de desaparición de las fanerógamas marinas, como la construcción de nuevas obras costeras, la contaminación o vertidos y el enterramiento por sedimentación. Hay que destacar que la principal causa de la pérdida de praderas marinas a nivel mundial es la reducción de la transparencia del agua, sobre todo como consecuencia del incremento de la carga de nutrientes y la turbidez [12].

2.2 FONDOS DE RODOLITOS

Los fondos de rodolitos, también llamados fondos de maërl, están compuestos mayoritariamente por algas rodófitas calcáreas no geniculadas de vida libre, que forman nódulos de aspecto rocoso sobre el sustrato blando. La composición de especies que conforman el nódulo calcáreo es diversa y está determinada por la profundidad, que condiciona la disponibilidad lumínica, la sedimentación y la hidrodinámica; el tamaño, la morfología y fisiología de dichos nódulos está condicionada por dichos elementos abióticos. Se conocen, al menos, un total de 8 géneros de algas que forman rodolitos: Hydrolithon, Lithophyllum, Lithothamnion, Mesophyllum, Neogonolithon, Phymatolithon, Spongites y Sporolithon, distribuyéndose desde el Ártico hasta los trópicos cubriendo normalmente grandes extensiones, y desde la zona intermareal hasta los 200 m de profundidad [1]. En nuestro Archipiélago se desarrollan principalmente entre los 15 y los 100 metros de profundidad.

Los fondos de rodolitos (figura 5) han sido catalogados como hábitats de gran biodiversidad, zona de cría y refugio para especies comerciales de peces e invertebrados [1]. Se encuentran incluidos en el Anexo I de la Directiva Hábitat como hábitat prioritario a conservar en el territorio de la Unión Europea [14].

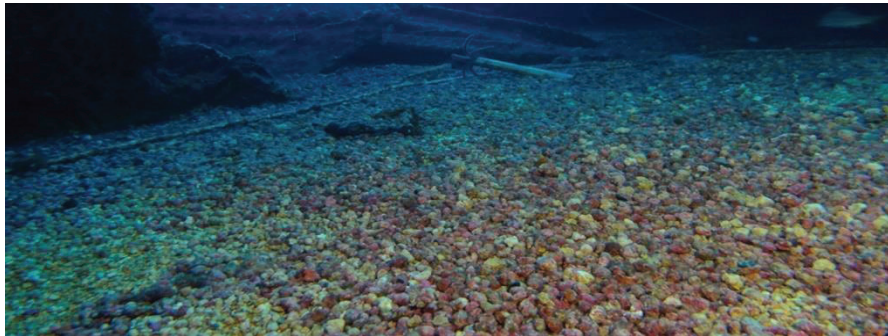


Figura 5 Fondos de rodolitos

Los rodolitos forman un entramado tridimensional que favorece el asentamiento de gran variedad de organismos, por lo que juegan un papel esencial en los ecosistemas marinos y constituyen fondos de pesca altamente productivos. No obstante, el lento crecimiento de este ecosistema (1-2 mm/año) y las condiciones peculiares de su desarrollo le confieren una gran vulnerabilidad. A pesar del esencial papel ecológico que desempeña este ecosistema, en la actualidad esta comunidad se ve amenazada por diferentes circunstancias como la contaminación, acidificación de los océanos o la extracción del maërl, principalmente para uso como fertilizante agrícola [14].

La presencia de estas algas es frecuente en los fondos que rodean todas las islas Canarias y en muchas de sus playas, aunque suele ser más notable en las islas orientales, que son las que cuentan con mayor plataforma insular y atesoran más edad geológica. Cuando están vivos, su color es rosado o violeta y una vez finalizado su ciclo de vida pierden esta coloración y se vuelven blancos. Además, por medio de la acción de las olas y las corrientes estas algas se dividen en fragmentos y terminan en algunas playas, como puede ser la playa Caleta del Barco de Fuerteventura (figura 6). Se identifican fácilmente debido a su similitud con las “roscas o palomitas de maíz”, lo cual ha fomentado el turismo en la zona [15].

Sistema de Control y Comunicaciones de un Perfilador Marino



Figura 6 Rodolitos de la playa Caleta del Barco en Fuerteventura [16]

Por otro lado, los fondos de rodolitos ayudan a la lucha contra el cambio climático, ya que absorben carbono de la atmósfera mediante la fotosíntesis. Gracias a esto, los fondos de rodolitos constituyen una fuente importante de carbono azul, que es como se llama al carbono orgánico capturado por los ecosistemas marinos [15].

2.3 SISTEMAS DE MONITORIZACIÓN EXISTENTES

DIY (*Do It Yourself*) es un movimiento en alza que fomenta el concepto de elaborar las ideas uno mismo rechazando la compra de los sistemas ya existentes.

Enfocados en este concepto los dispositivos como Arduino son extremadamente útiles, ya que aportan un sistema modular y flexible que brinda enormes oportunidades para la automatización, la creación de redes y la recopilación y análisis de datos. Arduino es un microcontrolador fácil de manejar con mucha información sobre su uso y programación. Los foros de usuarios son altamente populares y proporcionan un recurso muy útil para los principiantes en programación ya que en muchas ocasiones puede que ya se hayan resuelto los problemas típicos que pueden surgir. Además, hay una gran variedad de modelos distintos de Arduino, con diferentes características, costes y tamaños, lo que permite elegir el que más se adapte al proyecto que se desee realizar y, gracias a su capacidad modular, se pueden añadir módulos que aporten nuevas funcionalidades al sistema [17].

Los gastos y las dificultades logísticas del despliegue de equipos de monitoreo científico en ambientes acuáticos son las mayores limitaciones que existen para realizar dichas operaciones. Los instrumentos para el registro de datos en estos ambientes son los sensores comerciales más costosos del mercado. Es por ello por lo que se comenzó a crear alternativas menos costosas utilizando placas de microcontroladores de código abierto como Arduino. [3] Existen artículos que detallan sistemas similares al que se quiere llevar a cabo en este proyecto, en los que utilizan un Arduino y otros componentes para la supervisión en entornos acuáticos, formando un sistema de bajo costo, como por ejemplo el del proyecto *Cave Pearl*.

En este proyecto, desarrollado por el Departamento de Ciencias Planetarias y de la Tierra de la universidad de *Northwestern*, se creó una plataforma de registro flexible basada en Arduino para el monitoreo a largo plazo en entornos hostiles. Desarrollaron un sistema de prototipos generalizado de bajo costo y con flexibilidad de diseño al construir también las carcasas submarinas que contendrán los registradores de datos (figura 7 y 8), fomentando el DIY para la reducción de costes, lo que podrá multiplicar el número de prototipos y los puntos de muestreo.[18]

Sistema de Control y Comunicaciones de un Perfilador Marino

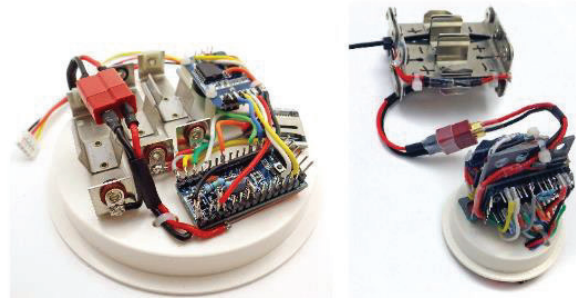


Figura 7 Diseño del prototipo del proyecto Cave Pearl (1) [18]

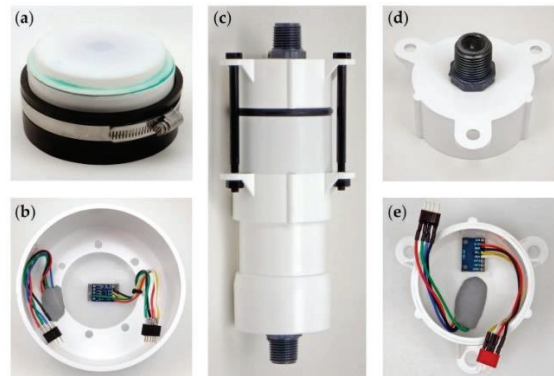


Figura 8 Diseño del prototipo del proyecto Cave Pearl (2) [18]

No es la primera vez que se habla de perfiladores utilizados en ambientes acuáticos. En Baja California, un estado de México, se llevó a cabo un proyecto en el que se desarrolló una plataforma CTD (*Conductivity Temperature Depth*) de elevación libre que realizaba perfiles a lo largo de una columna de agua (figura 9). Como su nombre indica, se trataba de un dispositivo que medía la conductividad eléctrica del agua de mar, su temperatura y la profundidad a la que se encontraba al realizar las dos medidas anteriores. A diferencia del perfilador marino desarrollado en este TFM, el dispositivo de este proyecto evitaba el uso de la electrónica para el ascenso y descenso del perfilador. En su caso, la profundidad a la que se iniciaba el perfilado dependía de la longitud de una correa fijada a un peso muerto y el ascenso estaba controlado por el equilibrio de la flotabilidad y el arrastre de un paracaídas incorporado. Como consecuencia, este perfilador debía tener una velocidad de ascenso lo suficientemente baja para realizar las lecturas con una buena resolución de la profundidad [6].

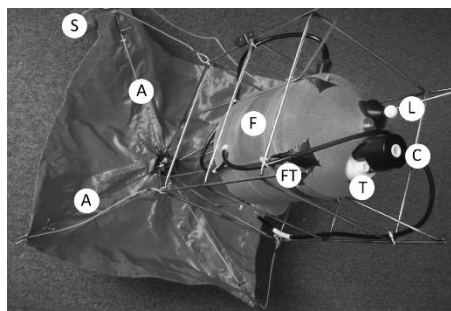


Figura 9 Perfilador CTD anclado y de elevación libre [6]

Sistema de Control y Comunicaciones de un Perfilador Marino

2.4 PERFILADOR MARINO

Como ya se ha mencionado, el alto coste de los equipos de monitoreo científico empleados en los estudios de servicios ecosistémicos impide su aplicación en los estudios de sistemas extensos limitándolos a unos pocos puntos geográficos. Además de este inconveniente, la metodología actual solo permite registrar parámetros de interés a una profundidad fija y solo es posible acceder a dichos datos tras recuperar los equipos de medida. El diseño que se describe en este TFM trata de superar estas tres limitaciones básicas dando una alternativa a la metodología estándar basada en costosos sensores oceanográficos.

Se ha planteado un diseño que consta de dos volúmenes unidos por tres varillas metálicas que proporcionan estabilidad y robustez al perfilador marino. El volumen superior será estanco y contendrá toda la electrónica a utilizar. Por el contrario, el volumen inferior irá inundado y en él se encontrará el motor dentro de un recipiente que lo aisle del agua. Esta diferencia de debe a que el volumen inferior manejará el cabo que permitirá que el perfilador marino suba y baje a lo largo de una columna de agua, teniendo el extremo de dicho cabo atado a un peso muerto.

En este proyecto, las variables de interés básicas para su estudio son el pH, el oxígeno disuelto y la radiación fotosintéticamente activa. El ciclo de operación típico que tendría este dispositivo está condicionado por los momentos en los que se generarán cambios en las variables de interés a lo largo de la columna de agua, por lo que se requiere que se realicen cuatro perfiles diarios sincronizados con las cuatro siguientes fases del día: el amanecer, el mediodía solar, el crepúsculo y la noche.

En este primer prototipo, desarrollado en este TFM, no se integrarán los sensores para realizar las lecturas de estas variables de interés, puesto que este trabajo se centra en el desarrollo del perfilador. En cambio, se han añadido otros elementos que facilitan su desarrollo experimental y que no estarán incluidos en las siguientes versiones de este.

Para dicho desarrollo experimental se necesitan varios componentes electrónicos. Se requieren sensores que permitan controlar la posición del perfilador marino, tanto la colocación que tiene, es decir, si se encuentra en posición vertical o no, como la profundidad a la que se encuentra. Además de la ubicación global del mismo. Por otro lado, se precisan sensores para la supervisión del sistema como el control del estado de la batería encargada de la alimentación de la electrónica, el control de la corriente que consume el motor y la presión interna de la sección del perfilador que contiene la electrónica. Por último, hacen falta sensores para el registro de las condiciones del entorno como puede ser la luz ambiental o la temperatura.

Para conseguir un sistema eficiente energéticamente se ha considerado, además de la elección de un microcontrolador de bajo consumo, la incorporación de dos módulos que trabajen como un interruptor para cortar la alimentación del motor y de los demás elementos en los momentos que no se vayan a utilizar. Únicamente para este modelo experimental se añadirán componentes que permitan informar del estado en el que se encuentra el sistema de forma visual como pueden ser una pantalla OLED y un LED RGB y también se agregarán dos sensores de efecto Hall que servirán para controlar el motor desde el exterior, pudiendo pararlo en caso de que fuera necesario.

Otro de los requerimientos indispensables para este prototipo incluye un motor con cierta capacidad de retención mecánica para poder realizar de forma eficaz los desplazamientos del perfilador. Además, para alimentar todo el sistema se necesitan dos baterías con diferentes capacidades, una que se utilice para alimentar al motor y otra que se encargue de la alimentación de la electrónica.

A continuación, en los apartados 2.5 y 2.6 de este capítulo se describirán los elementos escogidos para este proyecto.

2.5 MICROCONTROLADOR

La placa elegida para el control de todos los sensores y actuadores que componen este proyecto es el Arduino MKR GSM 1400 ilustrada en la figura 10 [19]. La familia de Arduino MKR son una serie de placas más pequeñas pensadas principalmente para su uso en sistemas de IoT (*Internet Of Things*).

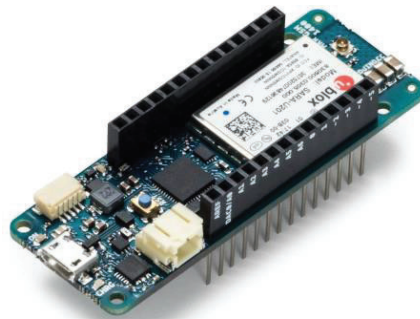


Figura 10 Arduino MKR GSM 1400 [19]

Esta placa está basada en el microcontrolador ARM Cortex-M0 de la familia SAM D21 de *Microchip* e incluye el módulo SARA-U201 de la serie SARA-U2 de u-blox. Además, dispone de un conector para una batería externa y un conector UFL para una antena. En la figura 11 se ilustran estos componentes señalados en la placa Arduino MKR GSM 1400.

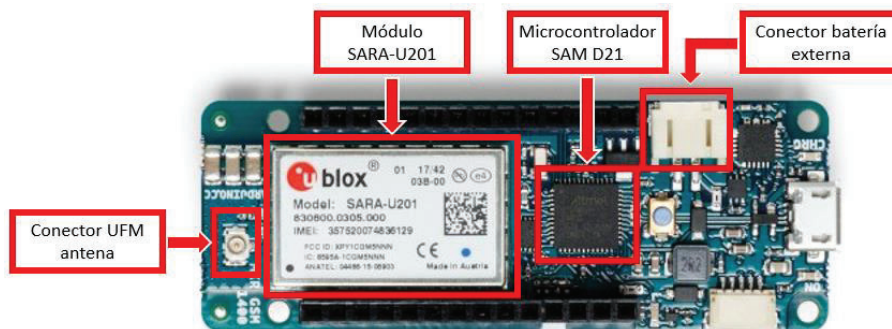


Figura 11 Módulos principales del Arduino MKR GSM 1400

El módulo SARA-U201 es un módulo *High-Speed Packet Access (HSPA)/Global System for Mobile (GSM)* el cual admite comunicaciones 2G y 3G con cobertura global. Se trata de un chipset de baja potencia que opera en las diferentes bandas del rango celular (GSM 850 MHz, E-GSM 1900 MHz, DCS 1800 MHz, PCS 1900 MHz) [20]. Para que esta comunicación sea posible se utilizará una tarjeta SIM conectada en la cara trasera del Arduino y una antena de hilo que se conectará al Arduino por medio del conector UFL (véase en las figuras 12 y 13) [21].

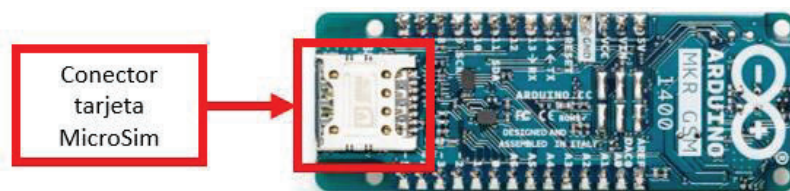


Figura 12 Cara trasera del Arduino MKR GSM 1400

Sistema de Control y Comunicaciones de un Perfilador Marino



Figura 13 Antena

El GSM es un sistema estándar de comunicación 2G desarrollado por el *European Telecommunications Standards Institute* (ETSI). El sistema estándar de comunicación 3G es el UMTS (Universal Mobile Telecommunications System) y, junto con el GSM, conforman la red que cubre el mayor porcentaje de la superficie mundial.

Además de eso, está garantizada la comunicación segura ya que se hace uso del chip de cifrado ATECC508A de *Microchip Technology* [22].

Por otro lado, se utilizará la placa Arduino MKR MEM Shield mostrada en la figura 14. Esta placa aporta una memoria Flash de 2MB y la posibilidad de agregar almacenamiento con una tarjeta microSD gracias a un conector que tiene incorporado. Esta ampliación de memoria permitirá el almacenamiento de los datos recogidos para su posterior análisis [23].

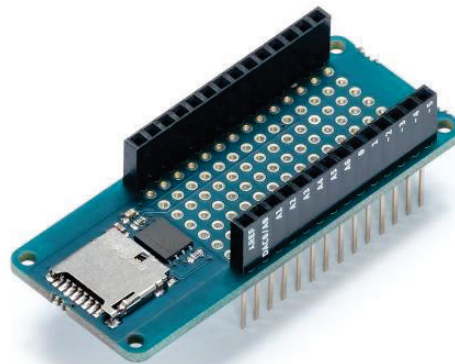


Figura 14 Arduino MKR MEM Shield [23]

Además, gracias a su compatibilidad con el Arduino MKR GSM 1400 se pueden conectar una encima de la otra permitiendo utilizar las ranuras que se encuentran en la parte central de la placa para el conexionado de cables por medio de la soldadura.

2.6 sensores y actuadores

A continuación, se mostrarán los sensores y actuadores escogidos para el primer prototipo del perfilador marino, la mayoría de los cuales pertenecen al sistema *SparkFun Qwiic Connect*.

El sistema *SparkFun Qwiic Connect* es un ecosistema de sensores, actuadores y cables I2C diseñado por *SparkFun* los cuales ayudan a que la creación de prototipos sea más rápida y menos propensa a errores ya que todas las placas habilitadas para *Qwiic* utilizan uno o más conectores JST de 4 pines y paso de 1 mm (figura 15), evitando el uso de la soldadura. Esto permite conectar varios dispositivos entre sí de manera muy sencilla sin tener fallos de conexionado gracias a los conectores polarizados y también reduce la cantidad de espacio requerido. Además, la mayoría de esos dispositivos disponen de librerías para el Arduino IDE facilitando su programación y uso.

Sistema de Control y Comunicaciones de un Perfilador Marino



Figura 15 Cable y conector qwiic

Los sensores y actuadores que se utilizan en este primer prototipo del perfilador marino son un acelerómetro, un sensor de micropresión, dos sensores magnéticos de efecto Hall, un sensor de luz ambiental, un sensor de corriente, un contador de corriente, un sensor de presión y un sensor de temperatura exteriores, un GNSS, una pantalla OLED, un LED RGB, un motor, un *Minipush button power Switch* y un *power Switch*. Todos los sensores y actuadores que trabajan con I2C están conectados al mismo bus, esto es posible porque todos tienen distintas direcciones.

2.6.1 Acelerómetro

El acelerómetro a utilizar es el MMA8452Q y la placa que lo contiene pertenece al sistema *SparkFun Qwiic Connect* (ver figura 16). Es un acelerómetro de tres ejes con escalas seleccionables de $\pm 2g$, $\pm 4g$ y $\pm 8g$ con datos tanto filtrados como no filtrados. Este dispositivo puede ser configurado para generar señales de interrupción a partir de cualquier función integrada configurable, lo que permite que este dispositivo supervise eventos y permanezca en modo de bajo consumo durante períodos de inactividad [24].

La dirección I2C de este dispositivo puede ser 0x1C o 0x1D siendo 0x1D su dirección por defecto.



Figura 16 Acelerómetro MMA8452Q [24]

2.6.2 Sensor de micropresión

Se utilizará el sensor de micropresión SEN-16476 integrado en una placa forma parte del sistema *SparkFun Qwiic Connect* mostrada en la figura 17. Este sensor tiene un consumo de energía muy bajo y ofrece un rango de detección de presión calibrado y compensado de 60 mbar a 2,5 bar [25].

La dirección I2C de este dispositivo es 0x18.



Figura 17 Sensor de micropresión [25]

Sistema de Control y Comunicaciones de un Perfilador Marino

2.6.3 Sensor magnético de efecto Hall

El sensor magnético de efecto hall que se utiliza es el modelo A1102 integrado en el módulo VMA313. Este módulo, mostrado en la figura 18, lleva incorporado un generador de tensión Hall para la detección de un campo magnético, un amplificador que amplía la tensión Hall y un disparador *Schmitt* que permite el rechazo del ruido. Además, el módulo dispone de un LED indicador para visualizar si se ha activado el sensor o no. Este se activa con la detección de 30 Gauss y se desactiva con medidas de 10 Gauss [26].



Figura 18 Sensor magnético de efecto Hall [26]

2.6.4 Sensor de luz ambiental

El sensor de luz ambiental APDS-9301 a utilizar está integrado en una placa compatible con las comunicaciones I2C mostrada en la figura 19 que convierte la intensidad de la luz en una señal de salida digital. Se trata de una placa de *SparkFun* bastante sencilla que ha sido diseñada con solo unos pocos componentes pasivos auxiliares además del propio APDS-9301. Todas las lecturas se devuelven al microprocesador elegido en lux, lo que proporciona una medición de iluminancia precisa en diversas condiciones de iluminación [27].

La dirección I2C de este dispositivo es 0x8A.



Figura 19 Sensor de luz ambiental [27]

2.6.5 Sensor de corriente

Sensor corriente y voltaje Zio – INA219, ilustrado en la figura 20, es una placa que pertenece al sistema *SparkFun Qwiic Connect*, el cual puede medir la corriente continua con hasta un 1% de precisión. Este sensor puede medir voltaje hasta 26 V en corriente continua y corriente de hasta 3.2 A [28].

La dirección I2C de este dispositivo puede ser 0x40, 0x41, 0x44 o 0x45, siendo 0x40 su dirección por defecto.



Figura 20 Sensor de corriente [28]

Sistema de Control y Comunicaciones de un Perfilador Marino

2.6.6 Contador de corriente o Coulomb Counter

El contador de corriente o *coulomb counter* es un dispositivo de *SparkFun* que supervisa el consumo de corriente de una batería y lo va almacenando para poder determinar en cualquier momento la cantidad de energía que le queda a la batería (ver en la figura 21) [29].

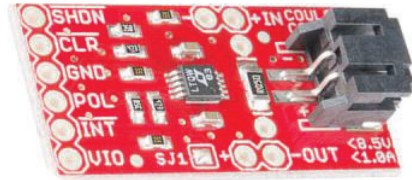


Figura 21 Coulomb Counter [29]

2.6.7 Sensor de temperatura y de presión exteriores

El sensor de temperatura y el sensor de presión exteriores son de la empresa *BlueRobotics*. Esta empresa se dedica a la fabricación de componentes asequibles y robustos para utilizarlos en la exploración oceánica [30].

El sensor de temperatura exterior a utilizar es el TSYS01, mostrado en la figura 22. Este sensor es de respuesta rápida y está sellado del agua, protegido por una jaula de aluminio y preparado para instalarlo en un recinto hermético haciendo uso de una junta tórica y una tuerca. Además, tiene una precisión de $\pm 0.1^{\circ}\text{C}$, utiliza un conector DF13 de 4 pines y se comunica por I2C [31].

La dirección I2C de este dispositivo puede ser 0x70 o 0x77, siendo 0x77 su dirección por defecto.



Figura 22 Sensor de temperatura TSYS01 [31]

El sensor de presión exterior que se va a utilizar es el MS5837, mostrado en la figura 23. Este sensor puede medir hasta 30 bar (300 metros de profundidad) con una resolución de 0.2 mbar (2 mm). Al igual que el sensor de temperatura, este sensor tiene un conector DF13 de 4 pines, se comunica por I2C y está sellado al agua y preparado para instalarlo en un recinto hermético. Además, este componente incluye un sensor de temperatura con una precisión de $\pm 0.1^{\circ}\text{C}$ [32].

La dirección I2C de este dispositivo es 0x76.

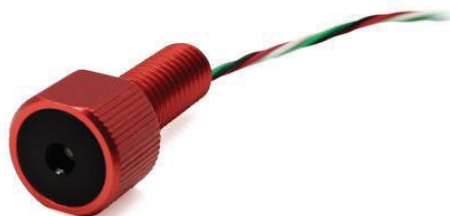


Figura 23 Sensor de presión MS5837 [32]

Para poder incorporar estos dos sensores al bus I2C del resto de los sensores del sistema *SparkFun Qwiic Connect* se utilizará el *I2C Bus Splitter*, mostrado en la figura 24, que contiene orificios

Sistema de Control y Comunicaciones de un Perfilador Marino

para clavijas de encabezado de conectores DF13. Con esto se pueden agrupar los dos sensores en un mismo bus I2C y conectarlos con un solo cable al bus I2C del resto del sistema [33].

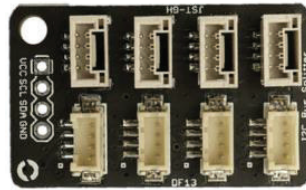


Figura 24 I2C Bus Splitter [33]

2.6.8 GNSS

La placa GNSS que se va a utilizar es la mostrada en la figura 25 y pertenece al sistema *SparkFun Qwiic Connect*. Esta placa contiene el módulo NEO-M9N que es un receptor GNSS de u-blox. Esta placa contiene su propia antena y está equipado con una batería recargable que proporciona energía al RTC (*Real Time Clock*) del módulo GNSS. Esto reduce el tiempo del primer arranque, de un arranque en frío (~ 24 s) a un arranque en caliente (~ 2 s). La batería mantendrá los datos de órbita RTC y GNSS sin estar conectada a la alimentación durante mucho tiempo [34].

La dirección I2C de este dispositivo es 0x42.

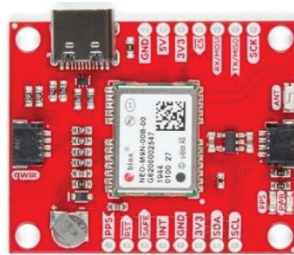


Figura 25 Placa GNSS [34]

2.6.9 Pantalla OLED

El módulo de pantalla *OLED Zio Qwiic* es el mostrado en la figura 26 y es el que se utiliza en este proyecto. A igual que otros componentes, pertenece al sistema *SparkFun Qwiic Connect* por lo que utiliza comunicaciones I2C y sus conectores *qwiic* se encuentran en la parte trasera de la placa. La pantalla es de 1.5 pulgadas y 128x128 pixeles permitiendo mostrar un máximo de 16 líneas de contenido de texto [35].

La dirección I2C de este dispositivo puede ser 0x78 o 0x7A, siendo 0x78 su dirección por defecto.



Figura 26 Pantalla OLED [35]

Sistema de Control y Comunicaciones de un Perfilador Marino

2.6.10 LED RGB

En este proyecto también se hará uso de un LED RGB (*Light Emitting Diode Red Green Blue*) (figura 27) de ánodo común utilizado para visualizar estados del sistema en función de su color y la frecuencia de parpadeo.



Figura 27 LED RGB.

2.6.11 Motor

El motor que se va a utilizar en este proyecto es un motor de corriente continua sin escobillas (BLDC - *Brushless Direct Current*) del fabricante *BringSmart* (figura 28).

Las ventajas de un motor sin escobillas sobre los motores con escobillas son una alta relación potencia-peso, alta velocidad, control casi instantáneo de velocidad (rpm) y torque, alta eficiencia y bajo mantenimiento.



Figura 28 Motor BLDC

2.6.12 Minipush button power Switch

Este dispositivo es esencialmente un interruptor de encendido implementado como un par de MOSFET de canal P (figura 29). La placa dispone de varios pines: alimentación de entrada y salida y pines de control (ON y OFF) con los que se encenderá o se apagará el interruptor, dejando pasar la corriente o no desde la entrada hasta los pines de salida [36].

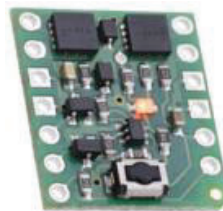


Figura 29 Minipush button Power Switch [36]

Sistema de Control y Comunicaciones de un Perfilador Marino

2.6.13 Power Switch

Esta placa, al igual que la explicada en el apartado anterior, funciona principalmente como un interruptor para cortar la alimentación de los dispositivos deseados y ahorrar la energía que consumen. Este dispositivo mostrado en la figura 30, al igual que otros de los explicados en este capítulo pertenecen al sistema *SparkFun Qwiic Connect*, por lo que utiliza comunicaciones I2C y los conectores *Qwiic* [37].

La dirección I2C de este dispositivo es 0x41.

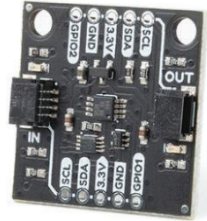


Figura 30 Power Switch [37]

2.6.14 Baterías

Para este proyecto se utilizarán dos baterías recargables. Una batería de 12V y 6000mAh para la alimentación del motor (figura 31) y una batería de 3,7V y 6600 mAh para la alimentación de la electrónica (figura 32).

Ambas baterías tienen incorporado un circuito de protección de carga y descarga llamado Sistema de Gestión de la batería (BMS, PCB o PCM). Estos circuitos se encargan de controlar la tensión de entrada y salida de la batería cortando la tensión de entrada o de salida cuando la batería se haya cargado o descargado hasta el límite recomendable, evitando que alcance valores peligrosos para las vidas de las celdas. Si las celdas se descargan o se sobrecargan fuera de los límites recomendados pueden dañarse [38].

La batería del motor consta de seis celdas e incluye dos conectores, uno hembra y otro macho, con los que se carga o se descarga.



Figura 31 Batería del motor

Por otro lado, la batería de la electrónica consta de tres celdas conectadas en paralelo e incluye dos cables con un conector de dos pines JST.



Figura 32 Bateria de la electrónica

2.7 PROTOCOLO DE COMUNICACIÓN FTP

En este apartado se va a describir el protocolo de comunicación que se utilizará para el envío de las lecturas realizadas por el perfilador.

Como bien se especifica en el título del apartado, se va a utilizar el protocolo de transferencia de archivos o FTP (*File Transfer Protocol*). Este es un protocolo de red basado en la arquitectura cliente-servidor para el transporte de archivos entre sistemas conectados a una red TCP (*Transmission Control Protocol*) [39]. Una gran ventaja de este protocolo es que permite la transmisión de archivos desde el cliente al servidor y viceversa, lo que posibilita el envío de las lecturas realizadas durante los perfiles al servidor y la descarga de nuevas actualizaciones de los parámetros del ciclo de trabajo del perfilador, como, por ejemplo, el tiempo de inactividad o el número de perfiles al día, de manera no síncrona, es decir, sin que sea necesario que el perfilador se encuentre conectado al servidor.

El FTP admite dos modos de conexión del cliente: activo y pasivo. En ambos modos el canal de control del servidor se sitúa en el puerto 21 y gracias a él se establecerá la conexión cliente-servidor. En el modo activo el servidor siempre crea el canal de datos en su puerto 20, mientras que el canal de datos del cliente se asocia a un puerto aleatorio mayor que el 1024. Haciendo uso del canal de control, el cliente le manda un comando PORT indicando el número del puerto escogido para que el servidor pueda abrirle una conexión de datos por la que se transferirán los archivos (figura 33) [39].

Sistema de Control y Comunicaciones de un Perfilador Marino

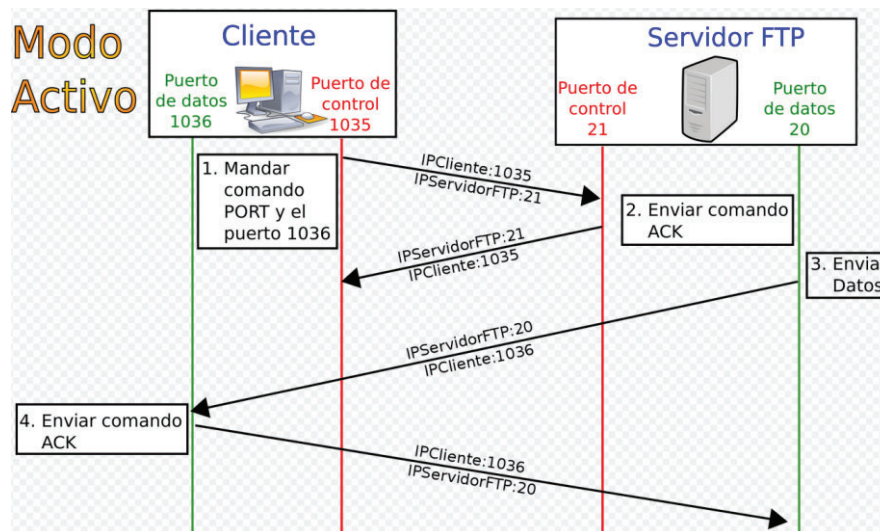


Figura 33 Protocolo FTP modo activo [39]

Por el contrario, en el modo pasivo, el cliente envía un comando PASV al servidor por el canal de control y este le indica el puerto aleatorio, mayor de 1024, al que debe conectarse el cliente para la transferencia de archivos (figura 34) [39].

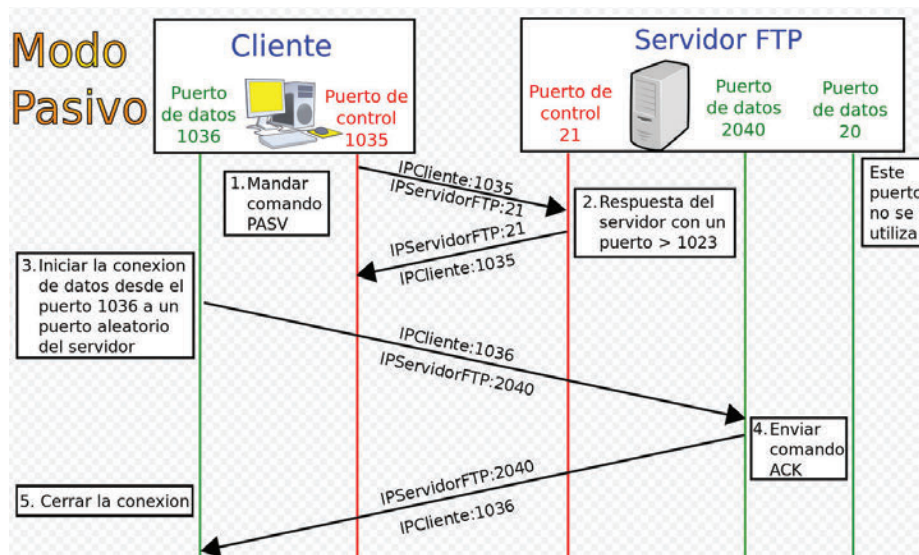


Figura 34 Protocolo FTP modo pasivo [39]

Un problema importante de este protocolo es la falta de seguridad, ya que todo el intercambio de información, desde el *login* y *password* del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor y/o apropiarse de los archivos transferidos [39]. Para solucionar este problema se pueden utilizar protocolos como el SCP (*Secure Copy Protocol*) y el SFTP (*SSH File Transfer Protocol*) para la transferencia segura de los archivos cifrando todo el tráfico de datos gracias al paquete SSH (*Secure Shell*).

Para este primer prototipo se creó un servidor FTP y utilizando el Arduino como cliente se realizó la transferencia del archivo .txt guardado en la SD en modo activo.

2.8 ENTORNO DE PROGRAMACIÓN ARDUINO IDE

Para el desarrollo software del Arduino se ha utilizado el software libre específico de Arduino, el Arduino IDE. Este entorno utiliza el lenguaje de programación de Arduino basado en el lenguaje de programación C++.

Este lenguaje de programación propio de Arduino se puede dividir en tres partes principales: funciones, variables y estructura. Las funciones, como su nombre indica, son las funciones propias de Arduino que sirven para controlar la placa y realizar cálculos. Las variables son los tipos de datos que se pueden declarar con este lenguaje. Por último, la estructura hace referencia a sintaxis, operadores y estructuras similares a los utilizados en el lenguaje C++ [40].

En las siguientes figuras 35 y 36 se muestra el icono, la ventana de inicialización y la vista inicial del entorno del Arduino IDE señalando algunas de las partes principales del mismo.



Figura 35 Icono y ventana de inicialización del Arduino IDE

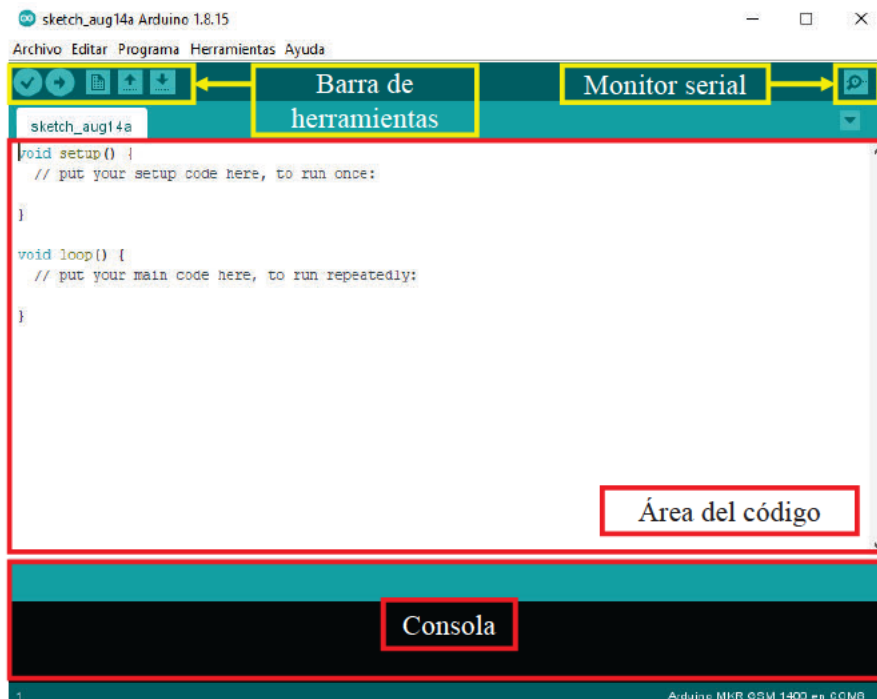


Figura 36 Vista inicial del entorno

Sistema de Control y Comunicaciones de un Perfilador Marino

Como se puede observar en la figura 36 la estructura de programación del Arduino IDE es muy simple, contando con una función *setup()* la cual solo se ejecutará una vez al inicio del programa y otra función *loop()* que se ejecutará continuamente después del *setup()*.

A parte de estas funciones esenciales para la ejecución del programa se pueden crear tantas funciones adicionales como se requieran y llamarlas desde cualquiera de las dos primeras funciones que se han mencionado para tener un código ordenado.

Un aspecto positivo de este entorno es la facilidad con la que se pueden añadir librerías y abrir los ejemplos que contienen. El procedimiento para añadir una librería es el mostrado en la figura 37, se puede añadir una carpeta .zip con la librería o se puede buscar a librería deseada en el gestor de librerías propio del Arduino IDE (figura 38).

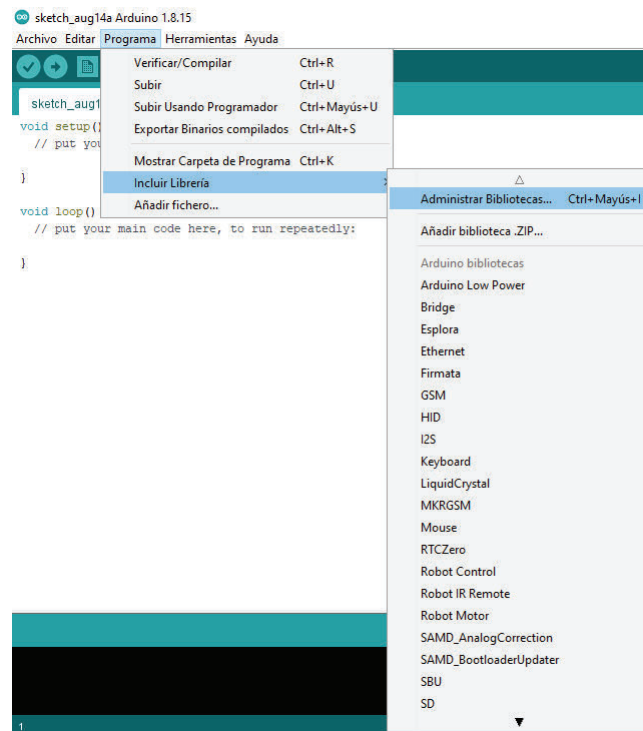


Figura 37 Añadir una librería con Arduino IDE

Sistema de Control y Comunicaciones de un Perfilador Marino

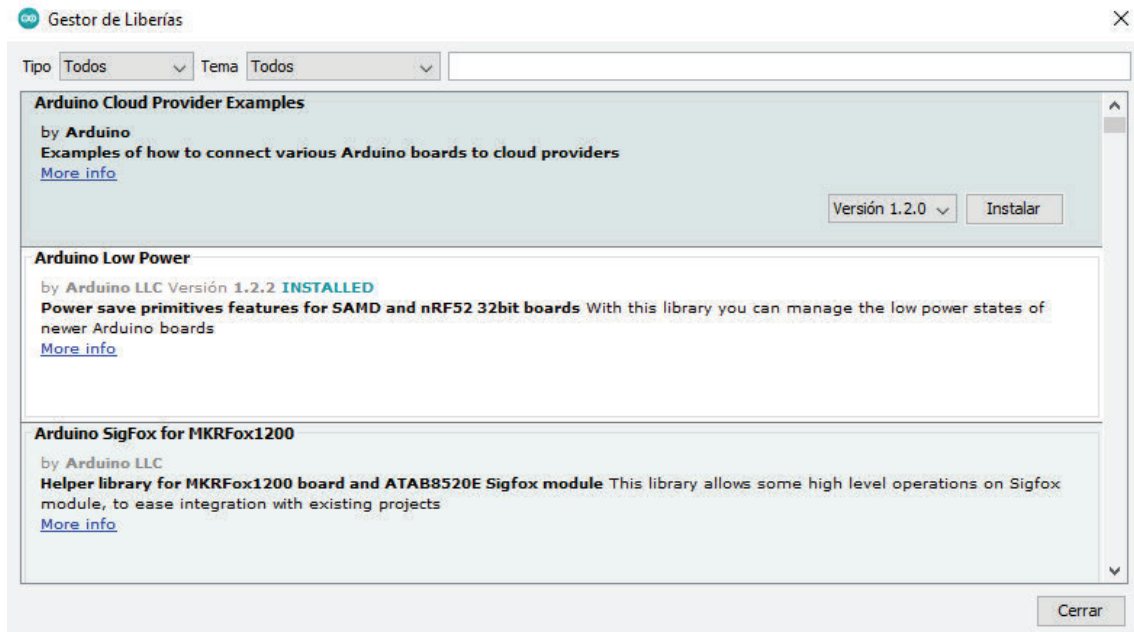


Figura 38 Gestor de librerías del Arduino IDE

Por otro lado, para acceder a los ejemplos de las librerías ya incluidas en el entorno como los de las librerías que se vayan añadiendo, solo hay que realizar los pasos de la figura 39 que como se puede observar son muy sencillos.

Sistema de Control y Comunicaciones de un Perfilador Marino

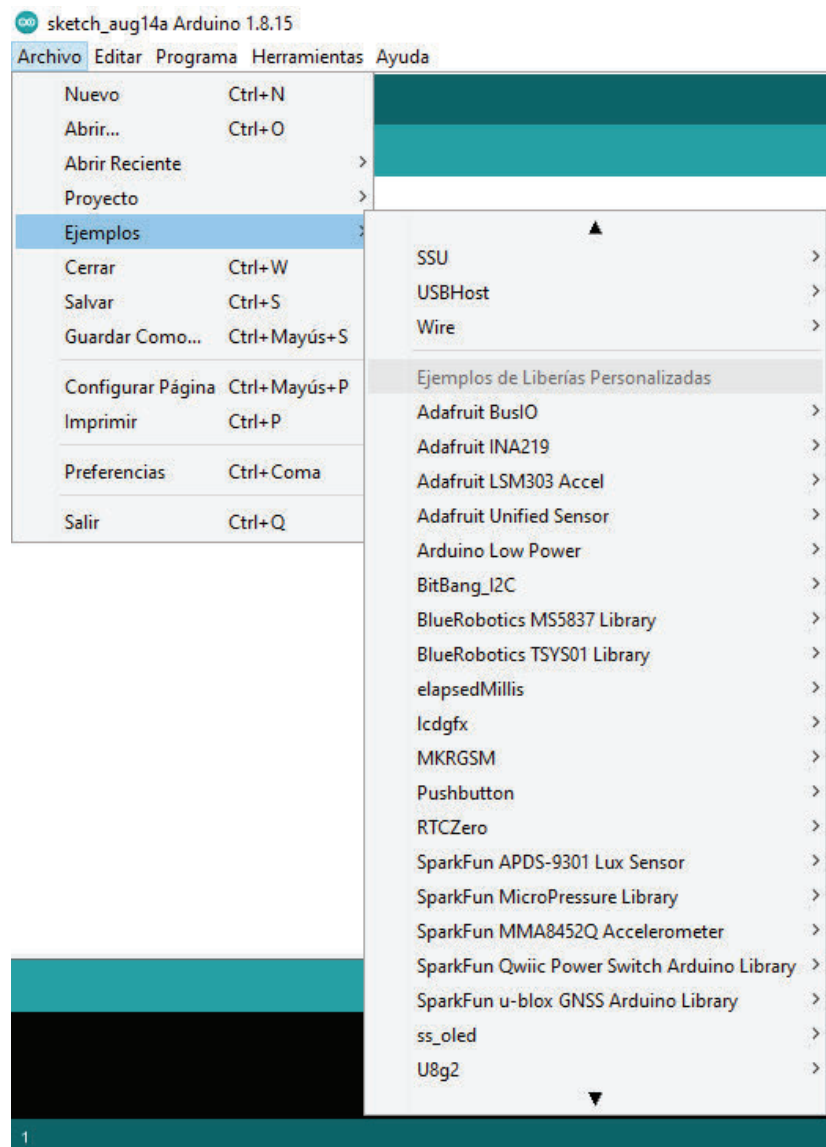


Figura 39 Acceso a los ejemplos del Arduino IDE

2.9 ENTORNO DE DISEÑO AUTODESK INVENTOR PROFESSIONAL 2022

Para el diseño de las piezas en 3D se utilizó el software *Autodesk Inventor Professional 2022*, un software de CAD (*Computer Aided Design*) que ofrece herramientas profesionales para el diseño mecánico 3D [41].

En las figuras 40 y 41 se muestran el icono, la ventana de inicialización y la vista inicial del entorno señalando algunas de las partes principales del mismo.

Sistema de Control y Comunicaciones de un Perfilador Marino



Figura 40 Icono y ventana de inicialización del Autodesk Inventor Professional 2022

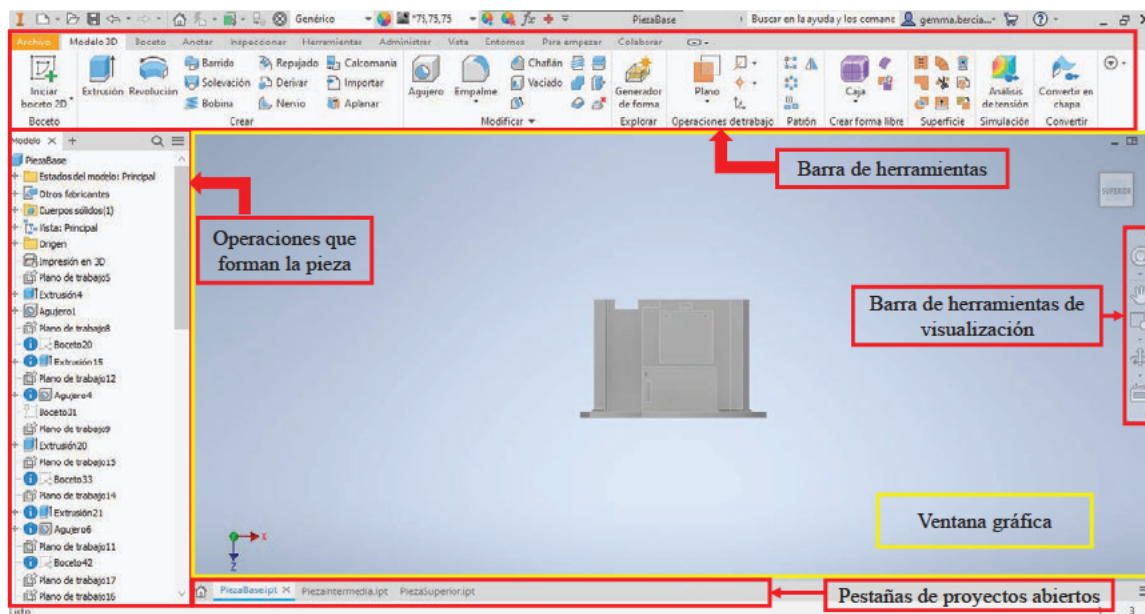


Figura 41 Vista inicial del entorno Autodesk Inventor Professional 2022

2.10 PRUSASLICER

Por último, para convertir el diseño resultante del *Autodesk Inventor Professional 2022* a código G, que es el lenguaje que necesita la impresora 3D, utilizamos el *PrusaSlicer*, un software de laminado para modelos de piezas en 3D [42].

En las figuras 42 y 43 se muestran el icono, la ventana de inicialización y la vista inicial del entorno señalando algunas de las partes principales del mismo.

Sistema de Control y Comunicaciones de un Perfilador Marino



Figura 42 Icono y Ventana de inicialización del PrusaSlicer

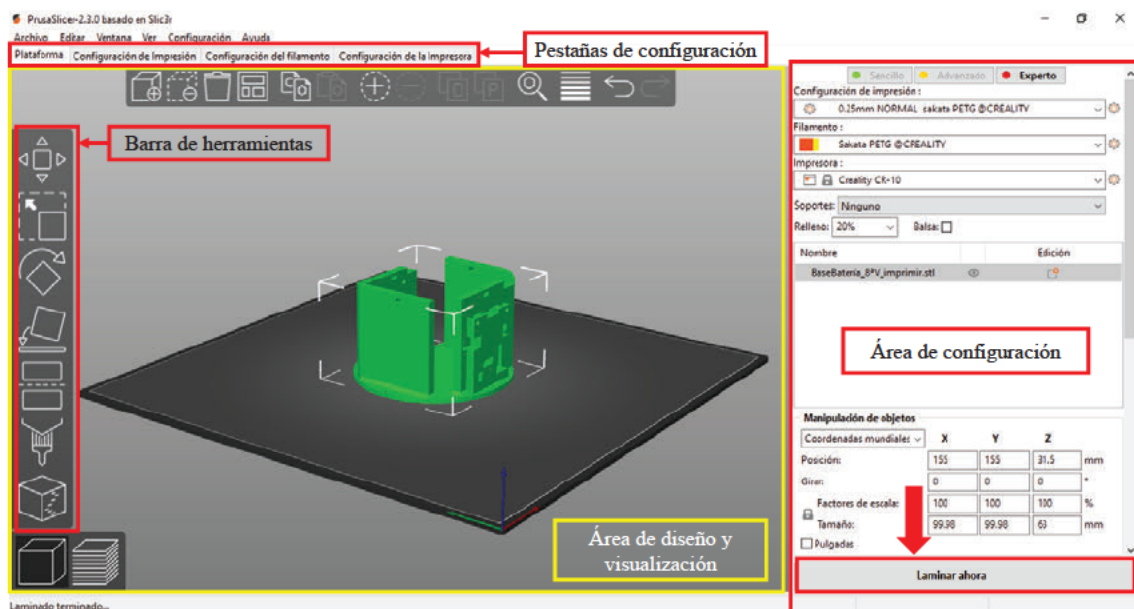


Figura 43 Vista inicial del entorno PrusaSlicer

En el área de configuración se especifica la impresora 3D a utilizar y el material con el que se va a imprimir entre otros factores. Una vez ya se han configurado todos los parámetros de impresión se pulsa el botón *Laminar ahora* señalado en la Figura 43.

Después de que se realice el laminado aparecerá una vista como la de la figura 44, en la que se detallarán las funciones y el tiempo que requerirán en una ventana informativa para saber el tiempo total estimado de impresión. Además, haciendo uso de la barra señalada en rojo se puede observar el laminado a distintas alturas de la pieza (figura 45).

Finalmente, cuando se está conforme con el resultado que se estima obtener se pulsa el botón *Exportar código G* señalado en la figura 45, para obtener el fichero que se le envía a la impresora 3D.

Sistema de Control y Comunicaciones de un Perfilador Marino

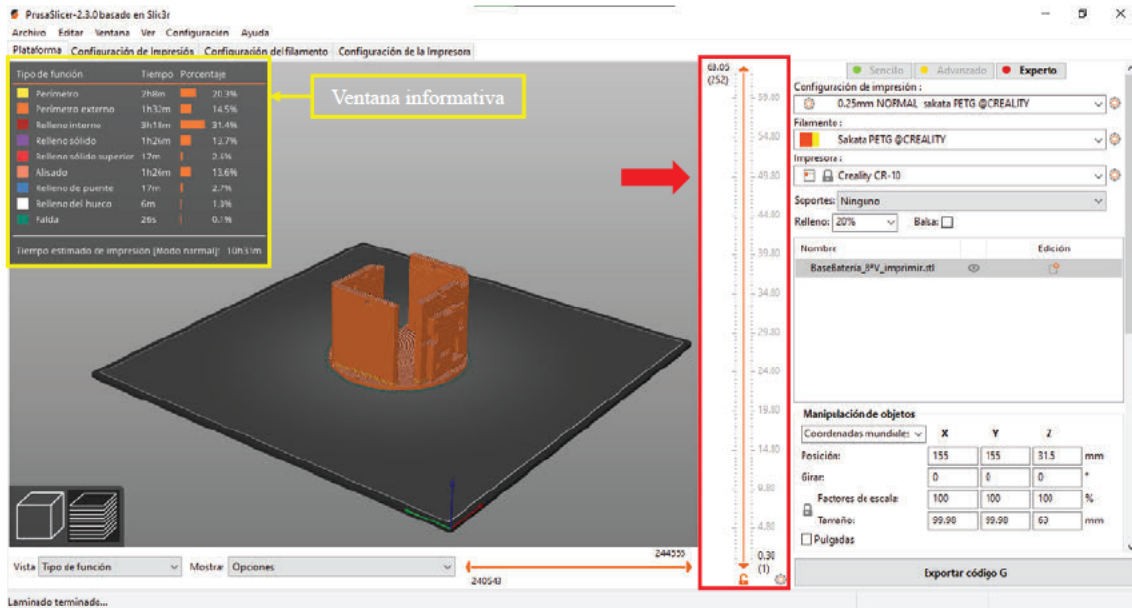


Figura 44 Vista de laminado del PrusaSlicer (1)

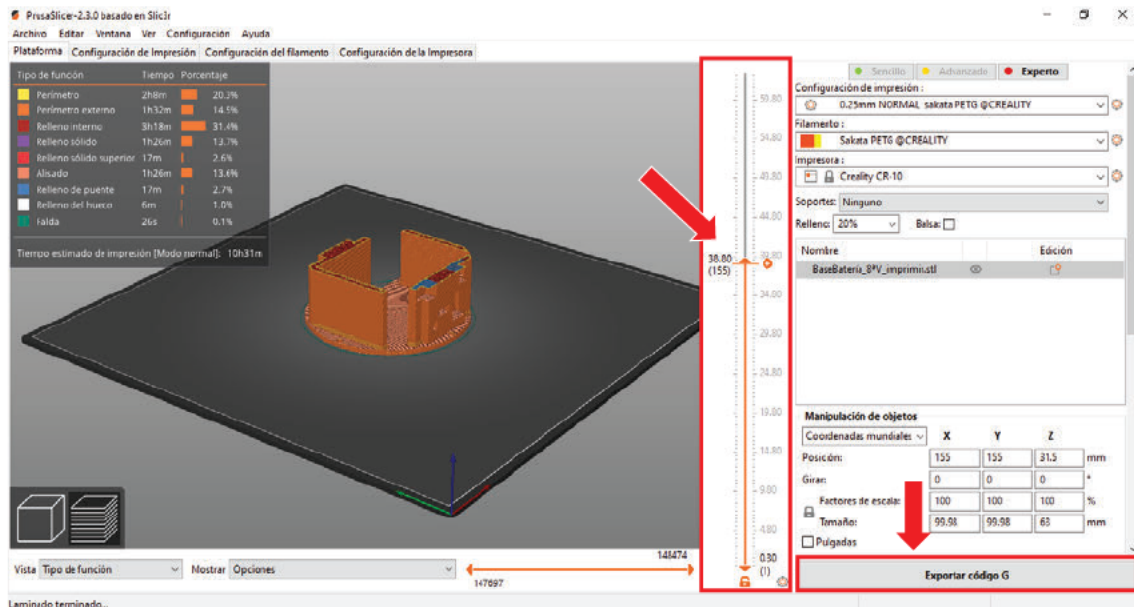


Figura 45 Vista de laminado del PrusaSlicer (2)

3 SOLUCIÓN HARDWARE ADOPTADA

En este capítulo se describirá la solución hardware adoptada.

Para crear una estructura robusta, estanca y preparada para trabajar en las condiciones oceánicas se adquirieron materiales de la serie 4'' de la empresa *BlueRobotics* [43].

La estructura del perfilador marino puede dividirse en dos partes (figura 46), una parte superior y una parte inferior, las cuales se describirán a continuación. Entre esas dos secciones del perfilador marino se encuentran los sensores de presión y temperatura de *BlueRobotics*, además de una válvula de alivio para liberar la presión del módulo durante el cierre y un pasacables para subir los cables del motor al volumen superior (figura 47). El pasacables irá sellado del agua utilizando un sellador sintético de alta resistencia con el que se rellenará el hueco que no ocupe el cable.

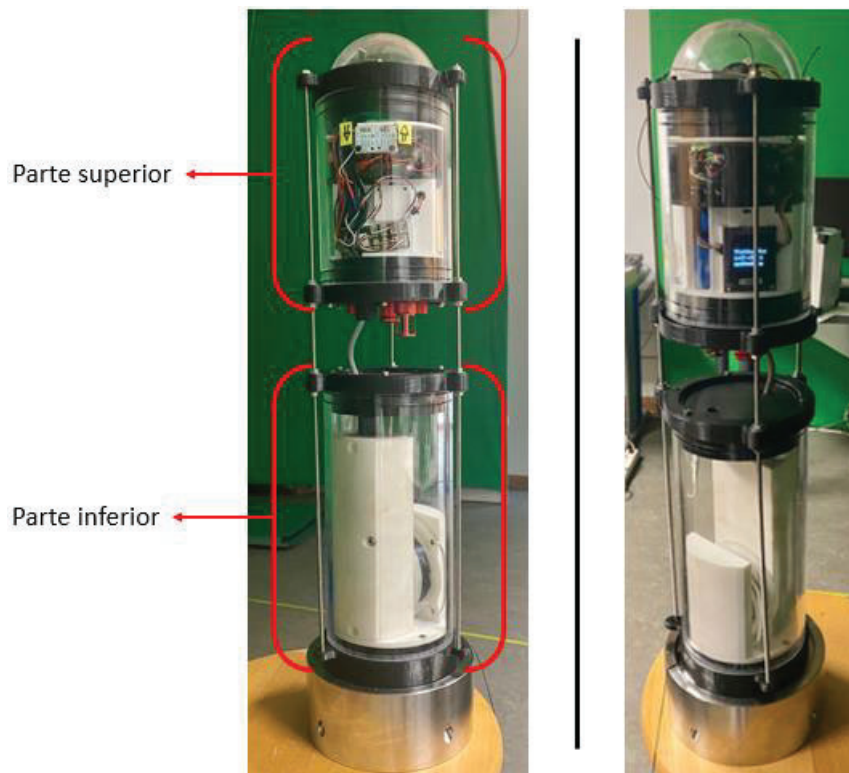


Figura 46 Perfilador marino montado

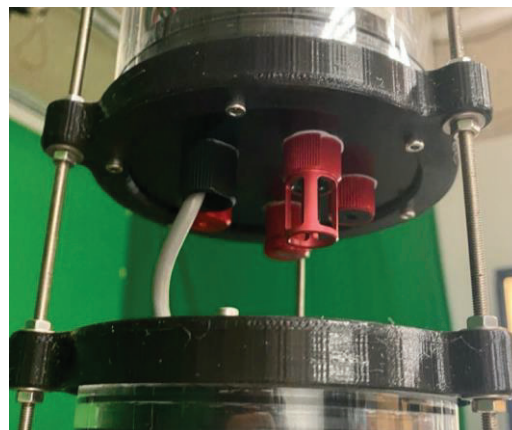


Figura 47 Espacio entre la parte superior y la parte inferior del perfilador marino

Sistema de Control y Comunicaciones de un Perfilador Marino

3.1 PARTE SUPERIOR

En esta sección de la estructura se encuentra toda la electrónica distribuida y anclada con una posición fija gracias a tres piezas diseñadas e impresas con una impresora 3D. Esta sección debe ser estanca y estar sellada del agua.

Para la estructura externa de la parte superior del perfilador marino se compraron las siguientes piezas de la serie 4'' de *BlueRobotics*: una tapa de extremo de domo (figura 48), dos bridas con sellos de junta tórica (figura 49), un tubo de acrílico de 15 cm (figura 50), varias juntas tóricas (figura 51) y una tapa de aluminio con 5 orificios (figura 52).



Figura 48 Tapa de extremo de domo



Figura 49 Brida con sellos de junta tórica



Figura 50 Tubo acrílico de 15 cm



Figura 51 Junta tórica

Sistema de Control y Comunicaciones de un Perfilador Marino

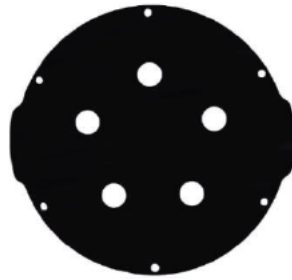


Figura 52 Tapa de aluminio de 5 orificios

Como se mencionó al principio de este capítulo, la estructura interna está formada por tres piezas impresas con la impresora 3D. En este caso, para la impresión se utilizó la impresora *Creativity CR10* del Laboratorio de Fabricación de Prototipos y Sistemas Electrónicos del IUMA y se utilizó el material PETG (*Polyethylene Terephthalate* (PET) mejorado con Glicol) debido a su facilidad de impresión y a que es un material duradero, con una buena resistencia a la temperatura y resistencia mecánica gracias a su buena adhesión entre capas [44].

En este TFM se diseñó la estructura interna, la cual está dividida en: la pieza base (figura 53), la pieza intermedia (figura 54) y la pieza superior (figura 56). A continuación, se mostrarán los diseños 3D realizados de cada una de estas piezas desde distintos ángulos y desde la planta para tener una visualización completa de las mismas.

➤ Pieza base:

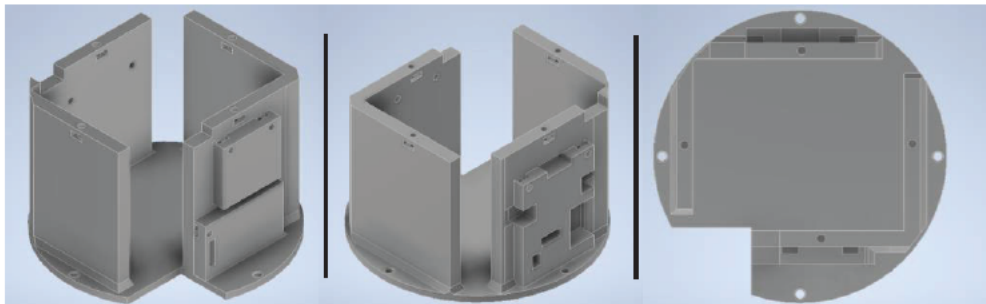


Figura 53 Pieza base de la estructura interna de la parte superior del perfilador

En esta pieza irán las dos baterías en el hueco central y, utilizando tuercas y tornillos, irán atornillados el *power switch*, la pantalla OLED y el *I2C Bus Splitter* en los soportes diseñados para ellos. También en la parte superior de la pieza hay huecos para tuercas y tornillos que servirán para atornillar la pieza intermedia encima de la pieza base. Además, cuenta con un hueco en la base de la pieza para el paso de cables desde la parte inferior.

➤ Pieza intermedia:

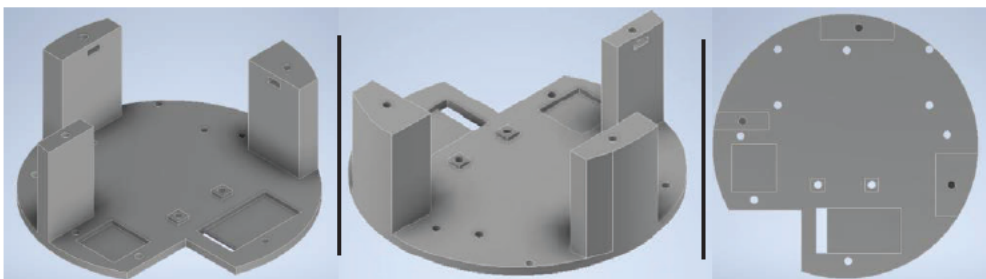


Figura 54 Pieza intermedia de la estructura interna de la parte superior del perfilador

Sistema de Control y Comunicaciones de un Perfilador Marino

En esta pieza se fijarán los siguientes componentes: el Arduino MKR GSM 1400 con el Arduino MKR MEM Shield; el sensor de micropresión, el acelerómetro y el sensor de corriente en la parte central atornillados unos encima de otros; el *Mini Push Button Power Switch* y el *Coulomb Counter*, fijados gracias a unas piezas complementarias mostradas en la figura 55.

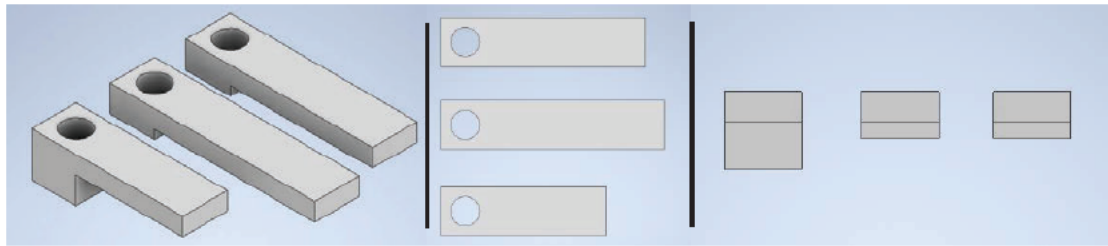


Figura 55 Piezas complementarias de la estructura interna de la parte superior del perfilador

Este diseño se dispuso de tal forma que el acelerómetro estuviera situado en el centro de la pieza para que sus medidas reflejaran más adecuadamente el movimiento que pueda tener el perfilador por las corrientes de agua.

➤ Pieza superior:

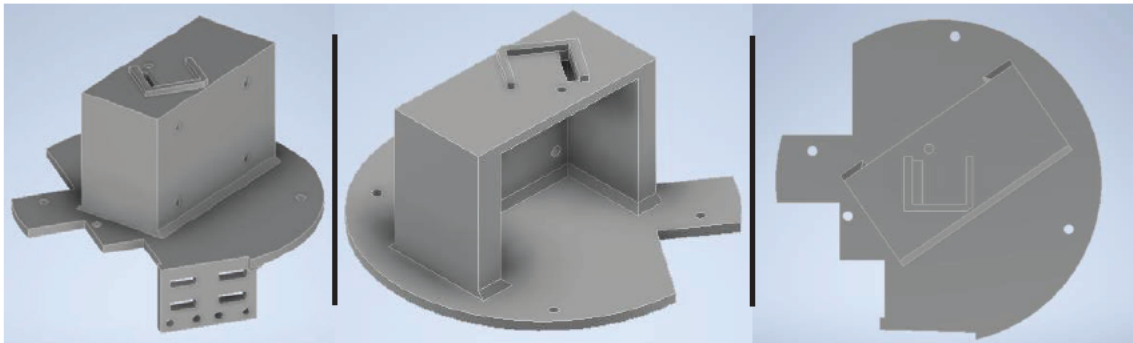


Figura 56 Pieza superior de la estructura interna de la parte superior del perfilador

En esta pieza, mostrada en la figura 56, se fijarán los dos sensores Hall en los soportes que sobresalen de la base de la pieza, al lado irá el LED RGB montado en una pequeña PCB con sus respectivas resistencias y en el bloque central se fijará el GNSS en el lateral y el sensor de luz ambiental en la parte superior.

Esto se ha pensado así para que ese bloque central llegue a la tapa de extremo de domo de tal forma que el sensor de luz pueda coger toda la intensidad lumínica posible y recoger la máxima información. De la misma manera, el GNSS se ha fijado en un lateral de ese bloque para que su antena esté también lo más cerca posible de la tapa superior y pueda conectarse con mayor facilidad una vez la estructura ha llegado a superficie.

Las dimensiones de estas piezas están condicionadas por el tamaño del tubo acrílico y se realizó un diseño de fácil montaje y acceso. En la figura 57 se puede ver la estructura montada con todos los componentes fijados. Como se puede ver en la figura 58 se diseñó para poder acceder al puerto USB y a la tarjeta SD sin tener que desmontar la estructura.

Sistema de Control y Comunicaciones de un Perfilador Marino

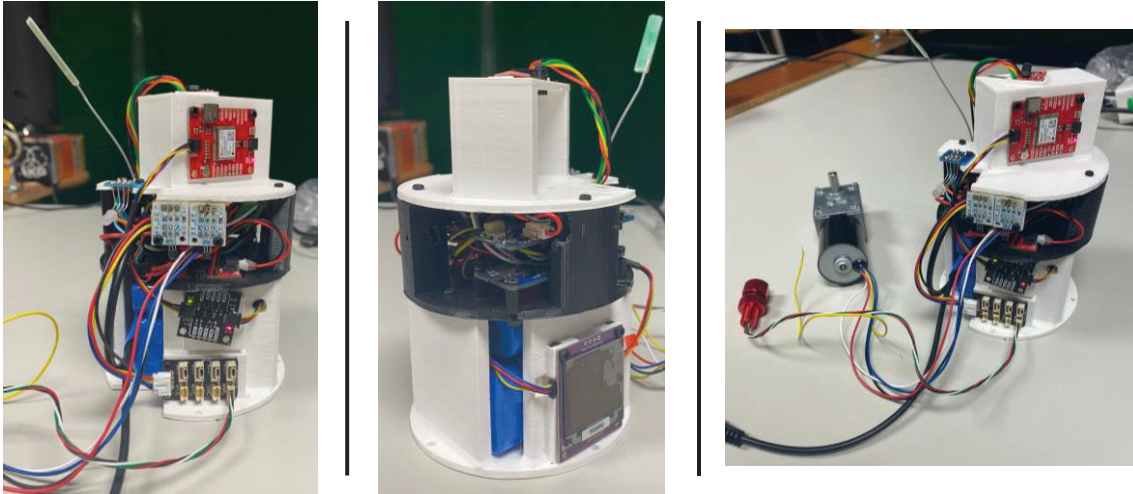


Figura 57 Estructura interna de la parte superior del perfilador montada

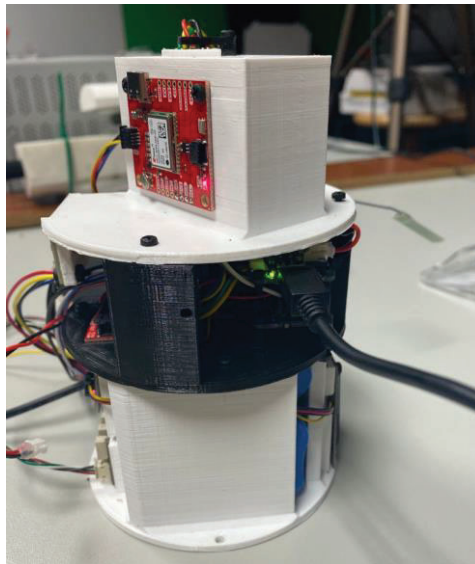


Figura 58 Vista lateral de la estructura de la parte superior del perfilador.

Por otro lado, dado que el LED RGB estaba suelto se soldó junto con tres resistencias a una pequeña PCB la cual está conectada al Arduino (ver en la figura 59).

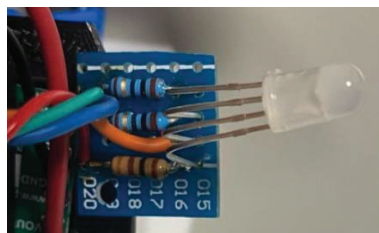


Figura 59 Placa del LED RGB

De la misma manera, se utilizó una PCB, mostrada en la figura 60, como intermediario entre el motor y el Arduino, el *Minipush button power Switch* y el sensor de corriente para poder desconectar el motor con facilidad.

Sistema de Control y Comunicaciones de un Perfilador Marino

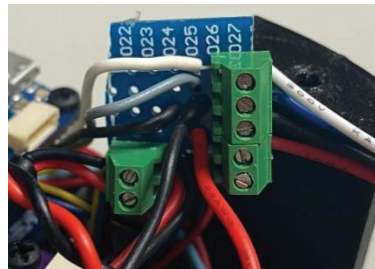


Figura 60 Placa intermedia del motor

A continuación, se muestra en las figuras 61 y 62 el conexionado de todos los componentes.

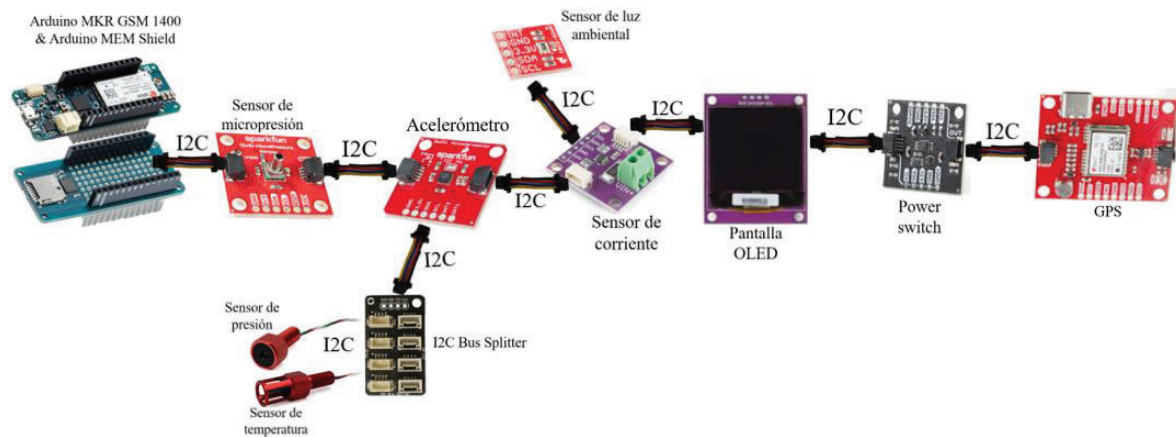


Figura 61 Conexionado (1) bus I2C

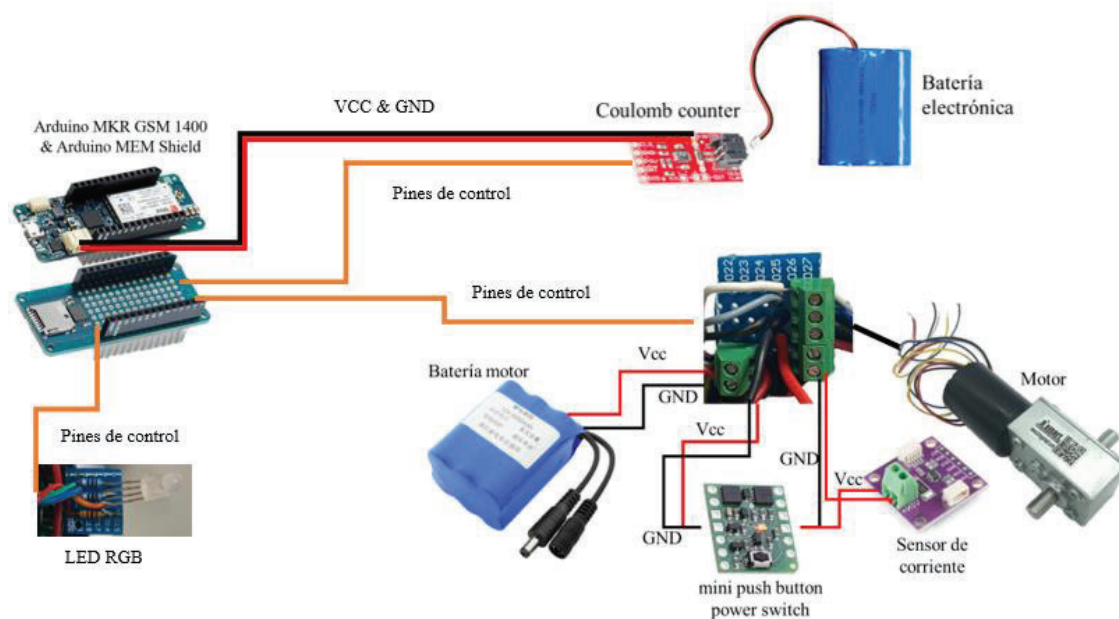


Figura 62 Conexionado (2)

3.2 PARTE INFERIOR

En esta parte de la estructura se encuentra la pieza que almacenará el motor y el cabo diseñada por un colaborador del proyecto (figura 63). A diferencia de las piezas impresas descritas en el apartado anterior, las piezas que se encuentran en esta parte del perfilador fueron fabricadas con polvo de nylon por la empresa Eysler Hidráulica.

Sistema de Control y Comunicaciones de un Perfilador Marino



Figura 63 Pieza resultante del módulo inferior del perfilador marino

Se eligió este material por su gran resistencia y sus cualidades de fricción. Las piezas fabricadas con polvo de nylon tienen una mejor vida útil y por ello son idóneas para piezas que requieren resistencia a lo largo del tiempo [45].

A continuación, en la figura 64 se muestra el diseño final de dicha pieza.

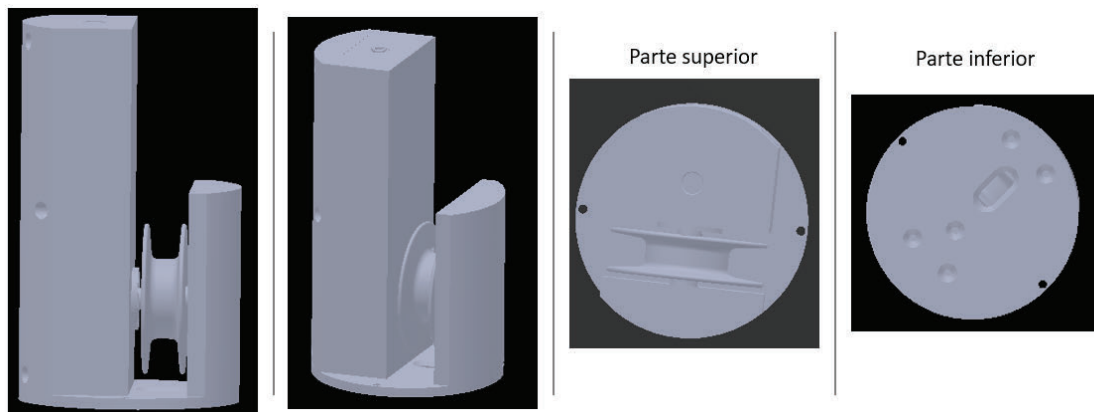


Figura 64 Diseño de la pieza del módulo inferior de perfilador marino

En las figuras 65 y 66 se muestra más detalladamente la pieza en la que va encajada el motor, la cual tiene una ranura destinada a guardar la junta tórica que permite el sellado del compartimento.

Sistema de Control y Comunicaciones de un Perfilador Marino

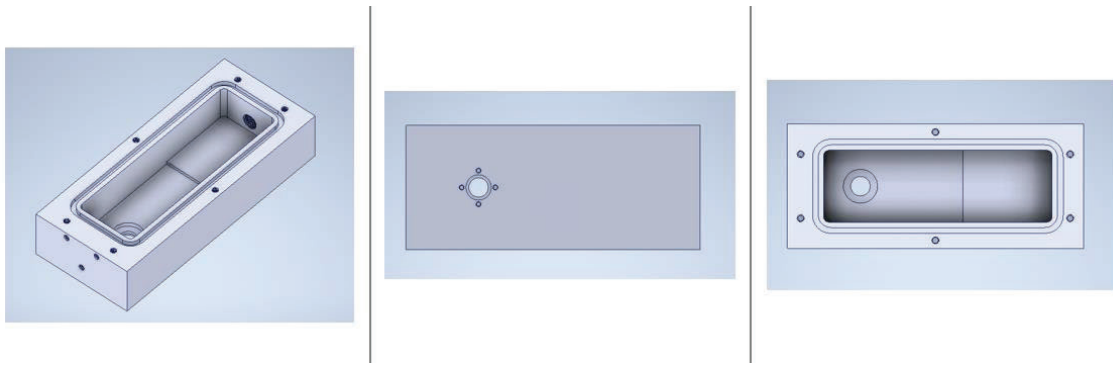


Figura 65 Diseño del compartimento del motor

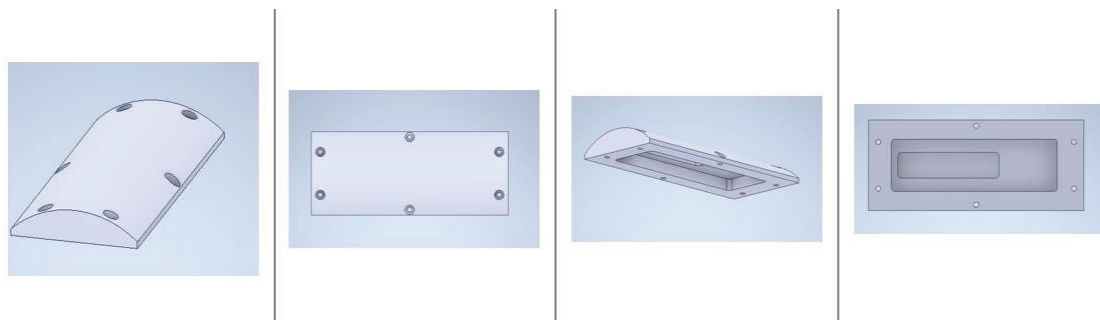


Figura 66 Diseño de la tapa del compartimento del motor

Para darle una mayor estabilidad y robustez a la estructura completa del perfilador marino se diseñaron e imprimieron cuatro piezas con PETG en la impresora 3D, realizadas también por un colaborador del proyecto. Dichas piezas están ubicadas en los extremos de cada una de las partes del perfilador, por las que pasarán tres varillas metálicas.

Se realizaron tres diseños: una anilla superior (figura 67), una anilla intermedia (figura 68) y un cono (figura 69) ubicado en la parte inferior del perfilador el cual se adapta a la forma del anclaje de peso muerto que estará situado debajo del perfilador marino (figura 70).

En la figura 71 podemos ver el diseño final del perfilador marino sin tener en cuenta las piezas del módulo superior.



Figura 67 Anilla superior del perfilador marino

Sistema de Control y Comunicaciones de un Perfilador Marino

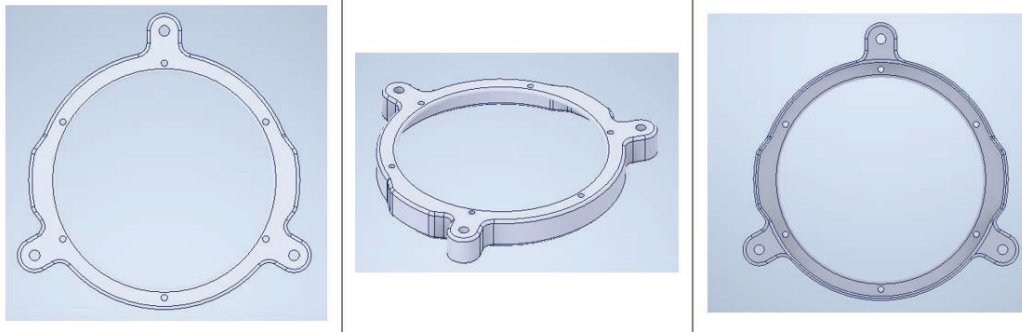


Figura 68 Anilla intermedia del perfilador marino

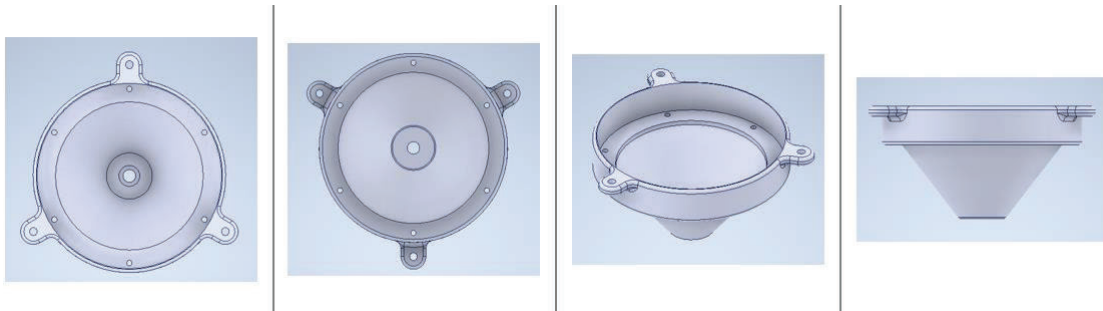


Figura 69 Cono de la parte inferior del perfilador marino

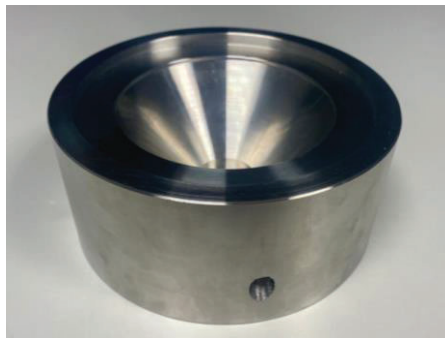


Figura 70 Anclaje de peso muerto

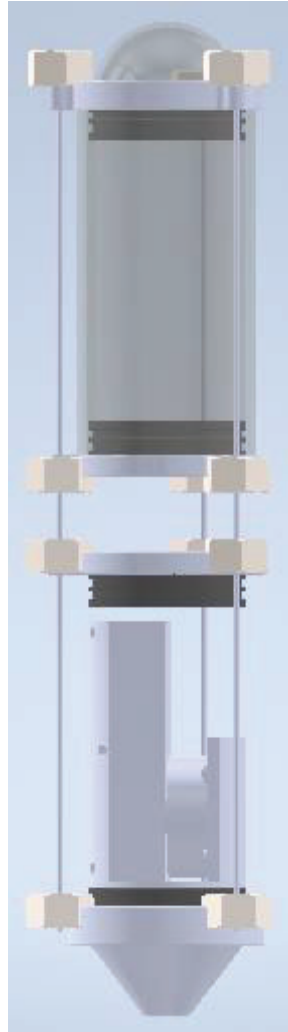


Figura 71 Diseño final de la estructura del perfilador marino

4 SOLUCIÓN SOFTWARE ADOPTADA

En este capítulo se explicará la solución software adoptada.

El desarrollo del software se enfocó teniendo en cuenta las principales dificultades del perfilador. Dichos retos son: la detección del momento en el que debe empezar la ejecución del ciclo de trabajo, la detección de la llegada a la superficie y la detección de la llegada al fondo.

La solución al primer problema fue incorporar dos sensores de efecto Hall y, haciendo uso de un imán, se activará cualquiera de los dos una vez esté el perfilador bien posicionado para indicar que puede comenzar el ciclo de trabajo.

Para el segundo problema se planteó como solución el poder conectar el módulo GNSS o el módulo SARA-U201 del Arduino ya que a cierta profundidad ninguno de estos módulos podrá conectarse. Una vez cualquiera de estos módulos se pueda conectar se comprobaría por medio del acelerómetro si el perfilador se encuentra en posición vertical o si, por el contrario, ha subido demasiado y se debe bajar hasta encontrarse en una posición adecuada.

Por último, la detección de la llegada al fondo se solucionó con la incorporación del sensor de corriente del motor. Una vez el perfilador llegue al fondo e intente seguir recogiendo cabo para seguir bajando la corriente consumida por el motor aumentará. Con ese aumento de la corriente se detectará que se ha llegado al fondo y cuando ocurra se parará el motor completamente y se impondrá una condición que no le deje seguir recogiendo el cabo hasta haberlo soltado, por lo que el perfilador una vez llegue al fondo solo podrá subir y no seguir bajando.

Por otro lado, la mayoría de los componentes disponen de librerías con funciones ya desarrolladas lo que facilita su uso. Para que el código fuera más ordenado se dividió el proyecto en funciones. En la mayoría de las funciones se utilizaron macros de depuración para mostrar por el puerto serie los valores recogidos de los sensores. Además, a todas las funciones que deben devolver datos, como las lecturas de los sensores, se les envía por referencia una variable de tipo *String* utilizada para almacenar dichos datos y poder acceder fácilmente a ellos desde el *loop*. Con esto se conseguirá tener en el *loop* una variable con todas las lecturas de los sensores la cual se servirá para el guardado en la tarjeta SD.

A continuación, se mostrarán las funciones realizadas:

4.1 FUNCIONES:

4.1.1 Acelerómetro:

La función del acelerómetro, mostrada en la figura 72, es bastante simple. Haciendo uso de las funciones de la librería "*SparkFun_MMA8452Q.h*" se comprueba si hay datos nuevos que leer del acelerómetro y se guardan en una variable de tipo *String* que se envía por referencia. Como ya se ha mencionado, esta variable se utilizará para guardar en la tarjeta SD los datos obtenidos del sensor.

Sistema de Control y Comunicaciones de un Perfilador Marino

```
/* Accelerometer: */
void AccelerometerReadings(String &dataString)
{
    if (accel.available()) // Wait for new data from accelerometer
    {
        dataString += "\nACCELEROMETER READINGS: ";
        dataString += String(accel.getCalculatedX(),3);
        dataString += ", ";
        dataString += String(accel.getCalculatedY(),3);
        dataString += ", ";
        dataString += String(accel.getCalculatedZ(),3);
        dataString += ".\n";

        #if defined( DEBUG_ACCELEROMETER )
        Serial.println("ACCELEROMETER READINGS: ");
        // Acceleration of x, y, and z directions in g units
        Serial.print(accel.getCalculatedX(), 3);
        Serial.print("\t");
        Serial.print(accel.getCalculatedY(), 3);
        Serial.print("\t");
        Serial.print(accel.getCalculatedZ(), 3);
        Serial.println();
        Serial.println(" ");
        delay(1000);
        #endif
    }
}
```

Figura 72 Función del acelerómetro

4.1.2 Sensor de micropresión:

La función del sensor de micropresión, al igual que la función del acelerómetro, es muy simple gracias al uso de la librería *SparkFun_MicroPressure.h*. Solamente se llama a la función *readPressure* indicando en qué unidades se quieren obtener los datos para recoger las lecturas del sensor.

```
/* MicroPreassure sensor */
void MicroPreassureSensorReadings(String &dataString)
{
    dataString += "MICROPRESSURE SENSOR READINGS: ";
    dataString += String(mPressureSensor.readPressure(PA));
    dataString += " Pa.\n ";
    #if defined( DEBUG_MPRESSURE_SENSOR )
    Serial.println("MICROPRESSURE SENSOR READINGS: ");
    Serial.print(mPressureSensor.readPressure(),4);
    Serial.println(" PSI");
    Serial.print(mPressureSensor.readPressure(PA),1);
    Serial.println(" Pa");
    Serial.print(mPressureSensor.readPressure(KPA),4);
    Serial.println(" kPa");
    Serial.print(mPressureSensor.readPressure(TORR),3);
    Serial.println(" torr");
    Serial.print(mPressureSensor.readPressure(INHG),4);
    Serial.println(" inHg");
    Serial.print(mPressureSensor.readPressure(ATM),6);
    Serial.println(" atm");
    Serial.print(mPressureSensor.readPressure(BAR),6);
    Serial.println(" bar");
    Serial.println();
    Serial.println(" ");
    delay(1000);
    #endif
}
```

Figura 73 Función del sensor de micropresión

Sistema de Control y Comunicaciones de un Perfilador Marino

4.1.3 Sensor magnético de efecto Hall

A diferencia de los dos casos anteriores, para el sensor de efecto Hall no hay librerías, se lee directamente el pin al que está conectado (ver figura 74). Se realizan diez medidas y, posteriormente, se calcula la media para filtrar los picos de señal que se puedan dar. Las variables que guardan las lecturas se envían por referencia, para poder acceder a los valores leídos desde el *loop*.

```
/* Hall effect sensor: */
void HallEffectSensorReading(long &measureSensor1, long &measureSensor2)
{
  for(int i = 0; i < 10; i++)
  {
    measureSensor1 += analogRead(pinHall_1);
    measureSensor2 += analogRead(pinHall_2);
  }

  // Calculation of the mean of the 10 measurements:
  measureSensor1 /= 10;
  measureSensor2 /= 10;

  #if defined( DEBUG_HALL_EFFECT_SENSOR )
  Serial.println("HALL EFFECT SENSOR READINGS:");
  Serial.print("Unconverted measurement Right sensor: ");
  Serial.println(measureSensor1);
  Serial.print("Unconverted Measure Left Sensor: ");
  Serial.println(measureSensor2);
  Serial.println("  \n\n");
  #endif
  delay(1000);
}
```

Figura 74 Función del sensor de efecto Hall

4.1.4 Sensor de luz ambiental:

El sensor de luz ambiental también cuenta con su propia librería *Sparkfun_APDS9301_Library.h* la cual se utilizará para obtener los datos en lux del sensor (figura 75).

```
/* Ambient light sensor */
void AmbientLightSensorReadings(String &dataString)
{
  static unsigned long outLoopTimer = 0;

  // This is a once-per-second timer that calculates and prints off
  // the current lux reading.
  if (millis() - outLoopTimer >= 1000)
  {
    outLoopTimer = millis();
    dataString += "\n AMBIENT LIGHT SENSOR READINGS: ";
    dataString += " Luminous flux: ";
    dataString += String(ambLightSensor.readCH0Level());
    dataString += ".\n";
    #if defined( DEBUG_AMB_LIGHT_SENSOR )
    Serial.println("AMBIENT LIGHT SENSOR READINGS: ");
    Serial.print(" Luminous flux: ");
    Serial.println(ambLightSensor.readCH0Level(), 6);
    delay(1000);
    #endif
  }
}
```

Figura 75 Función del sensor de luz ambiental

Sistema de Control y Comunicaciones de un Perfilador Marino

4.1.5 Sensor de corriente:

Este sensor también cuenta con una librería propia *Adafruit_INA219.h* y funciones para obtener las lecturas del sensor (ver figura 76). Al igual que en la función del sensor de efecto Hall en el sensor de corriente se realizan tantas lecturas como se indique en la variable *numReadings*, pero en vez de calcular la media, se guardan todas las lecturas en un vector, se ordenan los datos de menor a mayor utilizando la función *BubbleSortAsc* de la figura 77 y se calcula la mediana utilizando la función *MedianCalculation* de la figura 78. Además, se compara el valor de la mediana y si es mayor o igual a 650 (mA) se cambia a *true* la variable *stopMotor*, la cual indicará en la función del motor que este debe parar.

```
// -----  
// Function of current sensor  
// -----  
void GetCurrentMotor()  
{  
    float shuntvoltage = ina219.getShuntVoltage_mV();  
    float busvoltage = ina219.getBusVoltage_V();  
    float current_mA = ina219.getCurrent_mA();  
    float power_mW = ina219.getPower_mW();  
    float loadvoltage = busvoltage + (shuntvoltage / 1000);  
    int currentReadingsLength = sizeof(currentReadings) / sizeof(float);  
    float sum = 0;  
    float median = 0;  
    currentReadings[readings] = current_mA;  
    readings = readings + 1;  
  
    if(readings == numReadings)  
    {  
        for(int i = 0; i < numReadings; i++)  
        {  
            sum = sum + currentReadings[i];  
        }  
        BubbleSortAsc(currentReadings, currentReadingsLength);  
        median = MedianCalculation(currentReadings, currentReadingsLength);  
        #if defined( DEBUG_CURRENT_SENSOR )  
            Serial.print("\nMedian current:      "); Serial.print(median); Serial.println(" mA");  
            Serial.println("");  
        #endif  
        if(median >= 650)  
        {  
            stopMotor = true;  
        }  
        sum = 0;  
        readings = 0;  
    }  
}
```

Figura 76 Función del sensor de corriente

Sistema de Control y Comunicaciones de un Perfilador Marino

```
// -----  
// Function to sort data  
// -----  
void BubbleSortAsc(float* values, int length)  
{  
    int i, j, flag = 1;  
    int temp;  
    for (i = 1; (i <= length) && flag; i++)  
    {  
        flag = 0;  
        for (j = 0; j < (length - 1); j++)  
        {  
            if (values[j + 1] < values[j])  
            {  
                temp = values[j];  
                values[j] = values[j + 1];  
                values[j + 1] = temp;  
                flag = 1;  
            }  
        }  
    }  
}
```

Figura 77 Función BubbleSortAsc

```
// -----  
// Median Calculation  
// -----  
float MedianCalculation(float* values, int length)  
{  
    float result = 0;  
    if (length % 2 == 0)  
    {  
        result = (values[length/2] + values[(length/2)-1])/2;  
        return result;  
    }  
    else  
    {  
        result = values[(length-1)/2];  
        return result;  
    }  
}
```

Figura 78 Función MedianCalculation

4.1.6 Sensor de temperatura:

Para el sensor de temperatura exterior también existe una librería específica para ese componente la *TSYS01.h* con la que obtendremos las lecturas del sensor haciendo uso de la función *temperature*.

Sistema de Control y Comunicaciones de un Perfilador Marino

```
/* External temperature sensor */
void TemperatureSensorReadings(String &dataString)
{
    temperatureSensor.read();
    dataString += "EXTERNAL TEMPERATURE SENSOR READINGS: ";
    dataString += "Temperature: ";
    dataString += String(temperatureSensor.temperature());
    dataString += " deg C.\n ";
    #if defined( DEBUG_EXT_TEMPERATURE_SENSOR )
        Serial.println("EXTERNAL TEMPERATURE SENSOR READINGS: ");
        Serial.print("Temperature (from temperature sensor): ");
        Serial.print(temperatureSensor.temperature());
        Serial.println(" deg C");
        delay(1000);
    #endif
}
```

Figura 79 Función del sensor de temperatura

4.1.7 Sensor de presión:

Al igual que el sensor de temperatura, el sensor de presión exterior cuenta con la librería *MS5837.h* y las funciones *pressure* y *depth*, entre otras para obtener lecturas del sensor.

```
/* External pressure sensor */
void ExtPressureSensorReadings(String &dataString)
{
    extPressureSensor.read();
    dataString += "EXTERNAL PRESSURE SENSOR READINGS: ";
    dataString += " Pressure: ";
    dataString += String(extPressureSensor.pressure());
    dataString += " mbar, ";
    dataString += "Depth: ";
    dataString += String(extPressureSensor.depth());
    dataString += " m.\n ";
    #if defined( DEBUG_EXT_PRESSURE_SENSOR )
        Serial.println("EXTERNAL PRESSURE SENSOR READINGS: ");
        Serial.print("Pressure: ");
        Serial.print(extPressureSensor.pressure());
        Serial.println(" mbar");
        Serial.print("Depth: ");
        Serial.print(extPressureSensor.depth());
        Serial.println(" m");
        delay(1000);
    #endif
}
```

Figura 80 Función del sensor de presión exterior

4.1.8 GNSS:

El GNSS cuenta con su librería *SparkFun_u-blox_GNSS_Arduino_Library.h* con las funciones que retornan los valores obtenidos por el componente (figura 81).

Sistema de Control y Comunicaciones de un Perfilador Marino

```
/* GPS */
void GetPosicionGPS(String &dataString)
{
  //Query module only every second. Doing it more often will just cause I2C traffic.
  //The module only responds when a new position is available
  if (millis() - lastTime > 1000)
  {
    lastTime = millis(); //Update the timer
    long latitude = myGNSS.getLatitude();
    long longitude = myGNSS.getLongitude();
    long altitude = myGNSS.getAltitude();
    byte SIV = myGNSS.getSIV();
    dataString += "GPS: \n";
    dataString += "Latitude: "; dataString += String(latitude);
    dataString += ", longitude: "; dataString += String(longitude); dataString += " degrees * 10^-7, altitude: ";
    dataString += String(altitude); dataString += " mm, SIV: "; dataString += String(SIV);
    dataString += ".\n ";
    #if defined( DEBUG_GNSS )
      Serial.println("GNSS: ");
      Serial.print(F("Lat: ")); Serial.print(latitude);
      Serial.print(F(" Long: ")); Serial.print(longitude); Serial.print(F(" (degrees * 10^-7)"));
      Serial.print(F(" Alt: ")); Serial.print(altitude); Serial.print(F(" (mm)"));
      Serial.print(F(" SIV: ")); Serial.print(SIV);
      Serial.println();
      delay(500);
    #endif
  }
}
```

Figura 81 Función del GNSS

4.1.9 Pantalla OLED:

La pantalla OLED hace uso de la librería *U8g2lib.h* que configura todos los parámetros de configuración de la pantalla y permite escribir en ella de forma sencilla con las funciones que se pueden ver en la figura 82.

```
/* Oled Display */
void DisplayPrint(String text1, String text2, String text3)
{
  u8g2.setFont(u8g2_font_ncenB14_tr);
  u8g2.firstPage();
  do
  {
    u8g2.setCursor(10, 40);
    u8g2.print(text1);
    u8g2.setCursor(10, 60);
    u8g2.print(text2);
    u8g2.setCursor(10, 80);
    u8g2.print(text3);
  }
  while ( u8g2.nextPage() );
}
```

Figura 82 Función de la pantalla OLED

4.1.10 LED RGB:

Las funciones del LED RGB son bastante simples. Hay cinco funciones: *initialBlink*, *Red*, *Blue*, *Green* y *LedOff*. *InitialBlink* va encendiendo y apagando cada color y al terminar se apaga el LED completamente. Las demás funciones son para encender solo un color del LED y en el caso de la función *LedOff* para apagarlo.

Sistema de Control y Comunicaciones de un Perfilador Marino

```
/* LED RGB: */
void initialBlink()
{
    digitalWrite(pinR, LOW);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinB, HIGH);

    delay(1000);
    digitalWrite(pinR, HIGH);
    digitalWrite(pinG, LOW);
    digitalWrite(pinB, HIGH);

    delay(1000);
    digitalWrite(pinR, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinB, LOW);
    delay(1000);

    // LED off:
    digitalWrite(pinR, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinB, HIGH);
    delay(1000);
}

/* Red light */
void Red()
{
    digitalWrite(pinR, LOW);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinB, HIGH);
}

/* Green light */
void Green()
{
    digitalWrite(pinR, HIGH);
    digitalWrite(pinG, LOW);
    digitalWrite(pinB, HIGH);
}

/* Blue light */
void Blue()
{
    digitalWrite(pinR, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinB, LOW);
}

/* LED off */
void LedOff()
{
    digitalWrite(pinR, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinB, HIGH);
}
```

Figura 83 Funciones del LED RGB

4.1.11 Motor:

La función del motor se puede ver en las figuras 84 y 85. Primero realiza la siguiente comprobación: si la variable *onlyDrop* es *true* y si la velocidad de entrada, *target_speed*, es mayor que cero, el motor no se moverá, en cambio, si *target_speed* es menor que cero se cambiarán los valores de las variables *onlyDrop* y *stopMotor* a *false*. Esto se utiliza para cuando el sensor de corriente ha detectado que hay un mayor consumo y el perfilador no puede seguir bajando.

Si ninguno de esos condicionantes se cumple el motor entra en un *do... while()* que llama a la función *set_speed_inc* para mover el motor de manera progresiva, con rampas de aceleración y de frenado que permiten que el motor vaya aumentando o disminuyendo la velocidad progresivamente. Además, si la variable *stopMotor* cambia su valor a *true* se para el motor de manera progresiva, se cambia el valor de *hallEffectSensor* y de *onlyDrop* a *true* y se sale de la función con un *return*.

Sistema de Control y Comunicaciones de un Perfilador Marino

```
/* Motor */
void MoveMotor(int16_t target_speed)
{
    static int16_t current_speed = 0;
    static elapsedMillis motorCtrlCycle_ms;
    bool speed_updated = true;
    TurnSwitchOn();

    if(onlyDrop == true && target_speed > 0)
    {
        target_speed = 0;
    }
    else if(onlyDrop == true && target_speed < 0)
    {
        stopMotor = false;
        onlyDrop = false;
    }
    else
    {
        do
        {
            if ((motorCtrlCycle_ms >= MOTOR_CTRL_TIMESTEP_MS) || speed_updated )
            {
                Green();
                motorCtrlCycle_ms = 0;
                set_speed_inc(current_speed, target_speed);
                speed_updated = false;
                LedOff();
                GetCurrentMotor();
            }
        }
    }
}
```

Figura 84 Función del motor (1)

```
    if(stopMotor == true)
    {
        target_speed = 0;
        onlyDrop = true;
        speed_updated = true;
        hallEffectSensor = true;
        do
        {
            if ((motorCtrlCycle_ms >= MOTOR_CTRL_TIMESTEP_MS) || speed_updated )
            {
                Red();
                motorCtrlCycle_ms = 0;
                set_speed_inc(current_speed, target_speed);
                speed_updated = false;
                LedOff();
            }
        }
        while(current_speed != target_speed);
        return;
    }
}
else
{
    delay(250);
}
} while(current_speed != target_speed);
}
```

Figura 85 Función del motor (2)

Sistema de Control y Comunicaciones de un Perfilador Marino

4.1.12 Minipush button power switch:

Para el control del *minipush button power switch* se utilizan dos funciones: *TurnSwitchOn* y *TurnSwitchOff*. Cada función envía un pulso al pin correspondiente, *switchOn* para encender el *minipush button power switch* y *switchOff* para apagarlo.

```
/* Minipush button Power Switch */
/* A high pulse (> 1 V) at the ON pin turns the switch on;
   A high pulse (> 1 V) on the OFF pin turns the switch off;
   */
void TurnSwitchOn()
{
    #if defined( DEBUG_MOSFET )
        Serial.println("Minipush button Power Switch connected");
    #endif
    digitalWrite(switchON, HIGH);
    delay(1);
    digitalWrite(switchON, LOW);
}
void TurnSwitchOff()
{
    #if defined( DEBUG_MOSFET )
        Serial.println("Minipush button Power Switch disconnected");
    #endif
    digitalWrite(switchOFF, HIGH);
    delay(1);
    digitalWrite(switchOFF, LOW);
}
```

Figura 86 Función del Minipush button Power Switch

4.1.13 Coulomb Counter:

El coulomb counter va restando o sumando el valor de los miliamperios consumidos o aportados al valor inicial de miliamperio hora cada vez que salte la interrupción *myISR* para llevar la cuenta del porcentaje de carga de la batería (ver figura 87).

```
/* Coulomb Counter */
void CoulombCounter(String &dataString)
{
    if (isrflag)
    {
        // Reset the flag to false so we only do this once per INT
        isrflag = false;
        // Print out current status (variables set by myISR())
        dataString += "COULOMB COUNTER: ";
        dataString += String(battery_mAh); dataString += " mAh, ";
        dataString += " State Of Charge soc: "; dataString += String(battery_percent); dataString += "% time: ";
        dataString += String((time-lasttime)/1000000.0); dataString += "s mA: "; dataString += String(mA);
        dataString += ".\n";
        #if defined( DEBUG_COULOMB_COUNTER )
            Serial.print("mAh: ");
            Serial.print(battery_mAh);
            Serial.print(" soc: ");
            Serial.print(battery_percent);
            Serial.print("% time: ");
            Serial.print((time-lasttime)/1000000.0);
            Serial.print("s mA: ");
            Serial.println(mA);
        #endif
    }
}
```

Figura 87 Función del coulomb counter

4.1.14 Power switch:

Al power switch no se le ha hecho una función porque para encenderlo y apagarlo se hace uso directamente de las funciones *powerOn* y *powerOff* de la librería *SparkFun_Qwiic_Power_Switch_Arduino_Library.h*

Sistema de Control y Comunicaciones de un Perfilador Marino

4.1.15 Tarjeta SD:

Por último, a la tarjeta SD, al igual que al power switch no se le ha hecho una función porque se utilizan directamente las funciones de la librería *SD.h*. Se puede ver un pequeño ejemplo en la figura 88.

```
/* Guardado en la SD */
myFile = SD.open("Test.txt", FILE_WRITE);
if (myFile)
{
  #if defined( DEBUG_SD )
    Serial.print("Writing values in Test.txt...");
    Serial.print(data);
  #endif
  myFile.println("MEASUREMENTS \n");
  myFile.println(data);
  myFile.println(" \n ");
  // close the file:
  myFile.close();
  #if defined( DEBUG_SD )
    Serial.println("done.");
  #endif
} else {
  DisplayPrint((" Error "), ("saving data "), (" to SD "));
  // if the file didn't open, print an error:
  #if defined( DEBUG_SD )
    Serial.println("error opening Test.txt");
  #endif
}
```

Figura 88 Ejemplo de uso de las funciones de la tarjeta SD

4.2 COMUNICACIONES

Para el apartado de las comunicaciones del módulo SARA-U201 del Arduino se utiliza una rama de la librería *MKRGSM* llamada *feature/ftp* la cual implementa la posibilidad de comunicar el Arduino MKR GSM 1400 con un servidor FTP.

Utilizando las funciones de dicha librería se ha realizado una pequeña función que intenta establecer conexión con el servidor y le envía el archivo *.txt* con las lecturas de los sensores (figura 89).

```
void FTPCommunication(const char *myFile)
{
  /* Conexión al servidor */
  test("Connect to FTP server",
  ftp.connect(SECRET_FTP_HOST, SECRET_FTP_USER, SECRET_FTP_PASSWORD, SECRET_FTP_PORT, false));

  /* Subimos al servidor el archivo .txt */
  test("Upload from local file system to FTP server",
  ftp.upload("myFile", "myFileToServer", 10000));

  /* Desconexión del servidor */
  test("Disconnect to FTP server",
  ftp.disconnect());
}
```

Figura 89 Función para las comunicaciones FTP

La función *test* de la figura 90 se utiliza simplemente para tener una mejor visualización de si se ha completado con éxito la función que realizada.

Sistema de Control y Comunicaciones de un Perfilador Marino

```
bool test(const String& msg, bool function)
{
  if (function == true) {
    Serial.print("OK - ");
  }
  else {
    Serial.print("ERROR - ");
  }
  Serial.println(msg);

  return function;
}
```

Figura 90 Función test de las comunicaciones FTP

4.3 PROGRAMA PRINCIPAL

A continuación, se mostrará la estructura completa del programa que se utilizó para la prueba final del perfilador.

Como se mencionó en el apartado 2.8 del capítulo 2 la estructura de programación consta de dos funciones principales, el *setup* y el *loop*. Como se puede ver en el diagrama de flujo de la figura 91 en el *setup* se realiza la configuración de los pines y se inicializan todos los componentes.

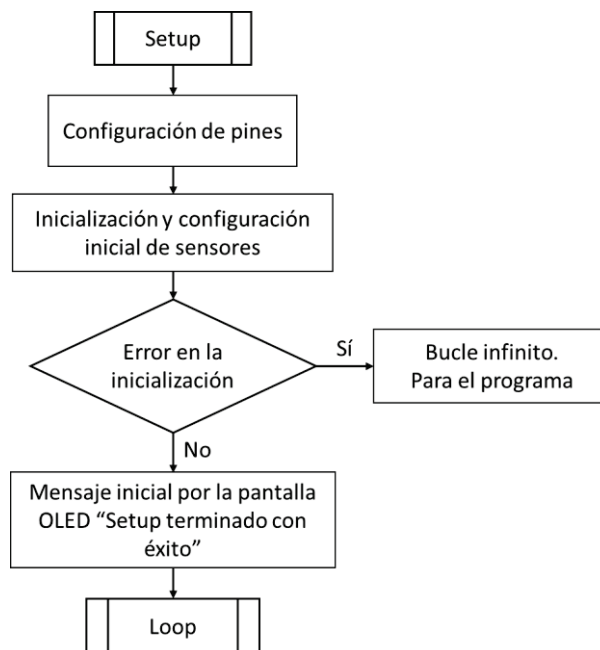


Figura 91 Diagrama de flujo del setup.

El *loop* es la función que se repite una y otra vez, por lo que contiene el ciclo de trabajo que realizará el perfilador (figura 92).

Sistema de Control y Comunicaciones de un Perfilador Marino

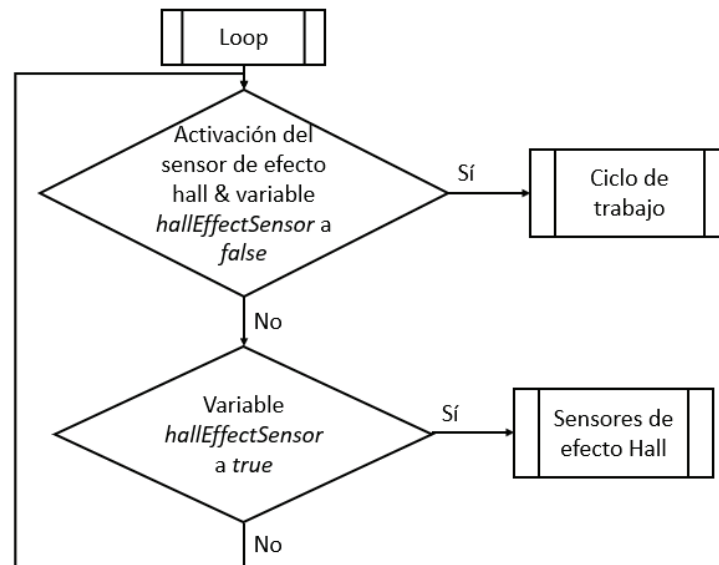


Figura 92 Diagrama de flujo del loop

El ciclo de trabajo, descrito en el diagrama de flujo de la figura 93, se puede dividir en tres fases:

1. Subida del perfilador, parada, lectura de sensores y reposo del sistema en modo de bajo consumo durante tres minutos.
2. Subida del perfilador, parada y lectura del sensor de luz ambiental y del estado de carga de la batería. Se guardan los datos en la SD y se queda el sistema en modo de bajo consumo durante dos minutos.
3. Bajada del perfilador al muerto y estado de modo de bajo consumo durante tres minutos.

Sistema de Control y Comunicaciones de un Perfilador Marino

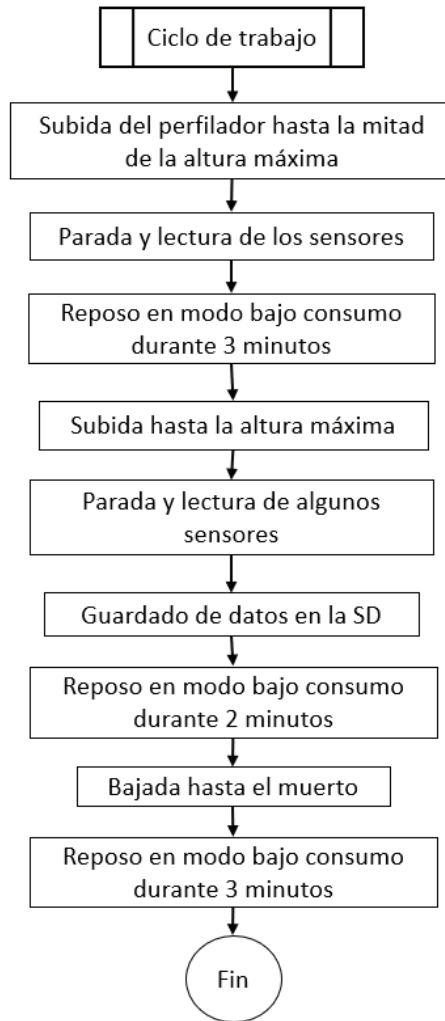


Figura 93 Diagrama de flujo del ciclo de trabajo

El control del movimiento del motor se realizará por tiempo y en esta prueba el ciclo de trabajo está condicionado por la altura del laboratorio, por lo que el motor subirá el perfilador durante cinco segundos en las dos subidas y durante 12 segundos en la bajada, un poco más del doble que la suma de tiempo de la subida porque esos tiempos no contemplan los tiempos de las rampas de aceleración y frenado, por lo que el perfilador se mueve durante más tiempo que lo indicado.

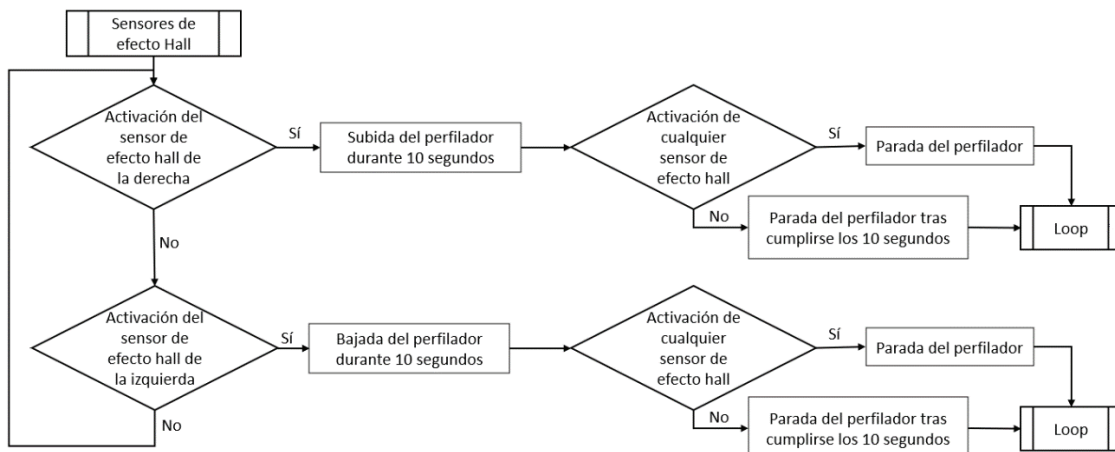


Figura 94 Diagrama de flujo de la función que se realiza cuando la variable hallEffectSensor toma el valor true

Sistema de Control y Comunicaciones de un Perfilador Marino

Como se puede ver en el diagrama de la figura 94 cuando la variable *hallEffectSensor* toma como valor *true* se inicia un bucle esperando a que alguno de los dos sensores de efecto Hall se activen para subir o bajar el perfilador durante diez segundos en función de cuál de los dos sensores se active. Durante la bajada o subida del perfilador se comprobarán continuamente los sensores de efecto Hall, si alguno se activa el motor se parará inmediatamente. Una vez se pare el motor se cambiará el valor de la variable *hallEffectSensor* a *false* y se volverá al *loop* esperando a que alguno de los sensores de efecto Hall se vuelva a activar para así volver a comenzar el ciclo de trabajo.

Por otro lado, en esta versión del programa no se ha incluido el GNSS ni el *power switch* debido a un problema que se originó durante el proceso de pruebas. En caso de haberse añadido estos dos componentes el *power switch* habría estado apagado hasta la segunda subida del perfilador y se habría utilizado el GNSS una vez el perfilador estuviera en la altura máxima. Después de usarlo se habría vuelto a apagar el *power switch* para ahorrar batería.

En las figuras de la 95 a la 100 se puede ver el contenido del *loop*. Como se puede observar en el código, mientras se mantiene activo el motor se comprueba el valor de los sensores Hall para que en caso de activarse alguno de los dos, el motor se pare y se active la variable *hallEffectSensor*.

```
void loop()
{
  static int16_t target_speed = 0;
  bool Stop = false;
  String data = " ";
  /* Esperamos a efecto Hall */
  HallEffectSensorReading(measureSensor1, measureSensor2);
  if((measureSensor1 < 30 && hallEffectSensor == false) || (measureSensor2 < 30 && hallEffectSensor == false))
  {
    /* Inicio, subida a Vmáx */
    target_speed = -100;
    DisplayPrint(("Move up"), (" motor"), (" "));
    MoveMotor(target_speed);
    for(int i=0; i<50; i++)
    {
      HallEffectSensorReading(measureSensor1, measureSensor2);
      if(measureSensor1 < 30 || measureSensor2 < 30)
      {
        DisplayPrint(("Hall effect "), ("Sensor!!"), (" "));
        target_speed = 0;
        MoveMotor(target_speed);
        hallEffectSensor = true;
        return;
      }
      delay(100);
    }
    target_speed = 0;
    DisplayPrint(("Stop"), (" motor"), (" "));
    MoveMotor(target_speed);
    /* Le quitamos la alimentación al motor */
    TurnSwitchOff();
    /* Perfilador a mitad de camino, iniciamos lecturas */
    Blue();
    DisplayPrint((" Start "), (" measurement "), (" "));
  }
}
```

Figura 95 Función loop (1)

Sistema de Control y Comunicaciones de un Perfilador Marino

```
/* MicroPreassure sensor */
MicroPreassureSensorReadings(data);
/* Accelerometer readings: */
AccelerometerReadings(data);
/* Ambient light sensor readings */
AmbientLightSensorReadings(data);
/* External pressure sensor readings */
ExtPressureSensorReadings(data);
/* External temperature sensor readings */
TemperatureSensorReadings(data);
LedOff();

/* Reposo durante 3 minutos */
DisplayPrint((" "),("sleep time"),(" 3 minutes "));
LowPower.deepSleep(180000);
/* Subida "a superficie" a Vmax */
target_speed = -100;
DisplayPrint(("Move up"),("motor"),(" "));
MoveMotor(target_speed);
for(int i=0; i<50; i++)
{
  HallEffectSensorReading(measureSensor1, measureSensor2);
  if(measureSensor1 < 30 || measureSensor2 < 30)
  {
    DisplayPrint(("Hall effect "),("Sensor!!"),(" "));
    target_speed = 0;
    MoveMotor(target_speed);
    hallEffectSensor = true;
    return;
  }
  delay(100);
}
target_speed = 0;
DisplayPrint(("Stop"),("motor"),(" "));
MoveMotor(target_speed);
```

Figura 96 Funcion loop (2)

```
/* Le quitamos la alimentacion al motor */
TurnSwitchOff();
/* Ambient light sensor readings */
data += "\nArrival of the profiler to the surface\n";
AmbientLightSensorReadings(data);
/* Comprueba el estado de carga de la batera de la electronica */
CoulombCounter(data);
/* Guardado en la SD */
DisplayPrint(("saving data"),(" to SD "),(" "));
myFile = SD.open("Test.txt", FILE_WRITE);
// if the file opened okay, write to it:
if (myFile)
{
  #if defined( DEBUG_SD )
  Serial.print("Writing values in Test.txt...");
  Serial.print(data);
  #endif
  myFile.println("MEASUREMENTS \n");
  myFile.println(data);
  myFile.println(" \n ");
  // close the file:
  myFile.close();
  #if defined( DEBUG_SD )
  Serial.println("done.");
  #endif
} else {
  DisplayPrint((" Error "),("saving data "),(" to SD "));
  // if the file didn't open, print an error:
  #if defined( DEBUG_SD )
  Serial.println("error opening Test.txt");
  #endif
}
}
```

Figura 97 Funcion loop (3)

Sistema de Control y Comunicaciones de un Perfilador Marino

```
/* Reposo durante 2 minutos */
DisplayPrint((" "), ("sleep time"), (" 2 minutes "));
LowPower.deepSleep(120000);
/* Bajada del perfilador hasta el muerto */
target_speed = 100;
DisplayPrint(("move down"), ("motor"), (" "));
MoveMotor(target_speed);
for(int i=0; i<120; i++)
{
    HallEffectSensorReading(measureSensor1, measureSensor2);
    if(measureSensor1 < 30 || measureSensor2 < 30)
    {
        DisplayPrint(("Hall effect "), ("Sensor!!"), (" "));
        target_speed = 0;
        MoveMotor(target_speed);
        hallEffectSensor = true;
        return;
    }
    delay(100);
}
target_speed = 0;
DisplayPrint(("Stop"), ("motor"), (" "));
MoveMotor(target_speed);
/* Le quitamos la alimentación al motor */
TurnSwitchOff();
/* Reposo durante 3 minutos */
DisplayPrint((" "), ("sleep time"), (" 3 minutes "));
LowPower.deepSleep(180000);
DisplayPrint(("end of"), ("cycle "), (" "));
}
```

Figura 98 Función loop (4)

```
if(hallEffectSensor == true)
{
    do
    {
        DisplayPrint(("Waiting for"), ("hall effect "), ("sensor "));
        HallEffectSensorReading(measureSensor1, measureSensor2);
        if(measureSensor1 < 30)
        {
            DisplayPrint(("Move up "), (" motor "), (" "));
            target_speed = -100;
            MoveMotor(target_speed);
            for(int i=0; i<100; i++)
            {
                HallEffectSensorReading(measureSensor1, measureSensor2);

                if(measureSensor1 < 30 || measureSensor2 < 30)
                {
                    DisplayPrint(("Hall effect "), ("Sensor!!"), (" "));
                    target_speed = 0;
                    MoveMotor(target_speed);
                }
                delay(100);
            }
            target_speed = 0;
            DisplayPrint(("Stop"), ("motor"), (" "));
            MoveMotor(target_speed);
            /* Le quitamos la alimentación al motor */
            TurnSwitchOff();
            hallEffectSensor = false;
            DisplayPrint(("Waiting for"), ("hall effect"), ("sensor "));
            return;
        }
    }
}
```

Figura 99 Función loop (5)

Sistema de Control y Comunicaciones de un Perfilador Marino

```
else if(measureSensor2 < 30)
{
  DisplayPrint(("Move down "), (" motor "), (" "));
  target_speed = 100;
  MoveMotor(target_speed);
  for(int i=0; i<100; i++)
  {
    HallEffectSensorReading(measureSensor1, measureSensor2);

    if(measureSensor1 < 30 || measureSensor2 < 30)
    {
      DisplayPrint(("Hall effect "), ("Sensor!!"), (" "));
      target_speed = 0;
      MoveMotor(target_speed);
    }
    delay(100);
  }
  target_speed = 0;
  DisplayPrint(("Stop"), ("motor"), (" "));
  MoveMotor(target_speed);
  /* Le quitamos la alimentación al motor */
  TurnSwitchOff();
  hallEffectSensor = false;
  DisplayPrint(("Waiting for"), ("hall effect "), (" sensor "));
  return;
}

}while(measureSensor1 < 30 || measureSensor2 < 30);
}
```

Figura 100 Función loop (6)

5 PRUEBAS REALIZADAS

En este capítulo se describirán las pruebas realizadas con el perfilador marino y los resultados obtenidos.

5.1 PRIMERA PRUEBA DE INMERSIÓN

La primera prueba de inmersión del perfilador se realizó en el Laboratorio de Robótica del SIANI utilizando para ello un depósito de agua dulce.

Para esta primera inmersión se montó la estructura completa del perfilador (ver en la figura 101). En la parte superior se sustituyó la electrónica junto con las piezas impresas en 3D por una bolsa de arena de un peso aproximadamente igual. Por otro lado, en la parte inferior se colocó la pieza que guarda el motor con el motor dentro para comprobar su estanqueidad. Previamente a esta prueba se realizó un pequeño ensayo de la pieza sin el motor en su interior para verificar su sellado al agua. Debido al éxito de este ensayo se prosiguió con las pruebas.



Figura 101 Perfilador marino completo para la primera prueba de inmersión

Para garantizar el sellado de la pieza que guarda el motor se le aplicó una capa de grasa por encima de la junta tórica y se le añadió un sellador sintético de alta resistencia para sellar el hueco sobrante del pasacables. (ver figura 102).



Figura 102 Pieza que guarda el motor

Sistema de Control y Comunicaciones de un Perfilador Marino

Esta prueba consistió en sumergir el perfilador completo para verificar la estanqueidad, tanto del módulo superior como de la pieza que guarda el motor, su flotabilidad y su estabilidad.

Como se puede ver en la figura 103, se comprobó que el perfilador tenía flotabilidad y su posición era estable.



Figura 103 Primera prueba de inmersión

Posteriormente, al sacar el perfilador del agua se comprobó que ni en el módulo superior ni en la pieza que almacena el motor entró agua, verificando así su estanqueidad.

5.2 PRUEBAS DE FUNCIONAMIENTO INDIVIDUALES

En este apartado se van a describir algunas de las pruebas que se les hicieron a algunos de los sensores y actuadores de manera individual.

5.2.1 Consumo de corriente del motor

Una de las pruebas que se hicieron se ve ilustrada en la figura 104 en la que, como se puede observar, el motor, ya dentro de su compartimento, está conectado al Arduino MKR GSM 1400 y a un sensor de corriente el cual recibe la alimentación de una fuente de alimentación. Con este conexionado se podrá medir la corriente que le llega al motor. Además, en la bobina de la pieza que contiene el motor está fijado el cabo el cual se irá enrollando o desenrollando poco a poco a medida que el motor vaya girando en un sentido u otro. El otro extremo del cabo está fijado a un carrete de pesca que ayudará en la prueba. Este carrete dispone de un freno que se ajusta de manera manual y que se utilizará para simular la llegada del perfilador al anclaje de peso muerto.

Sistema de Control y Comunicaciones de un Perfilador Marino

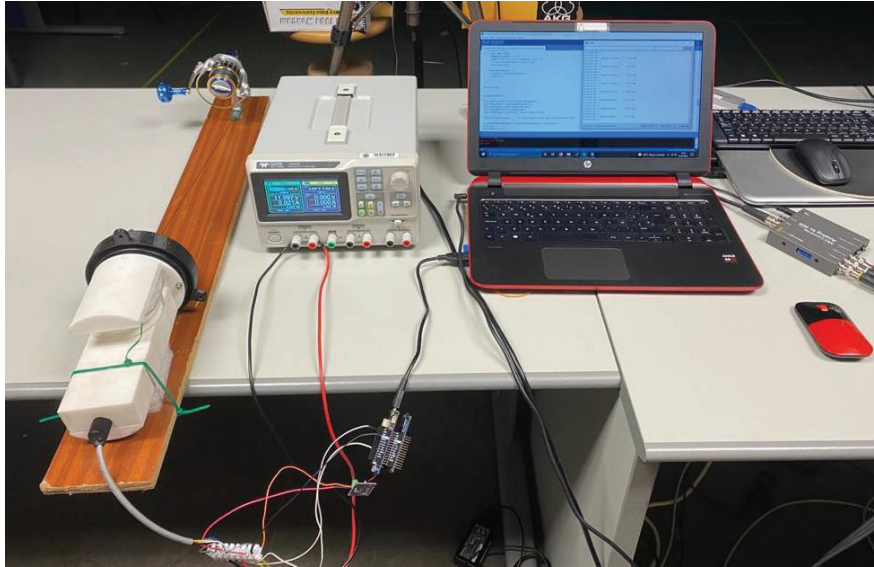


Figura 104 Prueba del motor

La prueba consistió en mover el motor en un sentido de giro para que recogiera el cabo y, en un momento dado, incrementar el freno del carrete. Al incrementar el freno del carrete el motor necesitaba más corriente para seguir recogiendo el cabo, lo cual se registraba con el sensor de corriente. En consecuencia, se añadió al programa que controlaba el movimiento del motor la detección de dicho aumento de consumo de corriente para que cuando este se produjera el motor parara bajando su velocidad moderadamente.

5.2.2 Sensores de efecto Hall

Otras de las pruebas previas al montaje del perfilador fue la verificación de los sensores de efecto Hall. Se quería comprobar que los sensores de efecto Hall se activarían al acercarlos un imán a pesar de tener la pared del tubo de acrílico en medio y que, situándolos uno al lado del otro, solo se activaría el elegido mientras el otro permanecería inactivo.

Esta prueba se realizó antes de imprimir la pieza superior de la electrónica, por lo que se pegaron los sensores de efecto Hall con cinta aislante. Como se puede ver en la figura 105, la prueba dio buenos resultados.

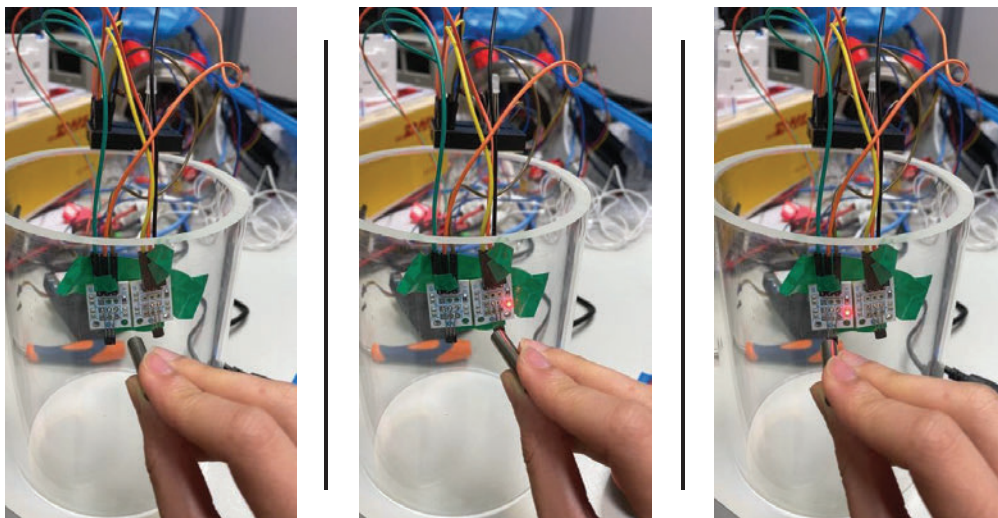


Figura 105 Prueba de sensores de efecto Hall

Sistema de Control y Comunicaciones de un Perfilador Marino

5.2.3 Funcionamiento del GNSS

Por último, se realizó una prueba con todos los componentes conectados. En esta prueba se observó que cuando se encendía la placa GNSS los demás componentes conectados al bus I2C dejaban de funcionar correctamente.

La primera suposición del problema fue pensar que el módulo GNSS requería tanta alimentación que dejaba al resto de los componentes del bus I2C sin la suficiente alimentación como para funcionar correctamente. Por ello se conectó la placa GNSS al bus I2C y se alimentó con una fuente de alimentación externa que le pudiera aportar la potencia necesaria para su funcionamiento sin quitarles alimentación al resto de componentes (figura 106).

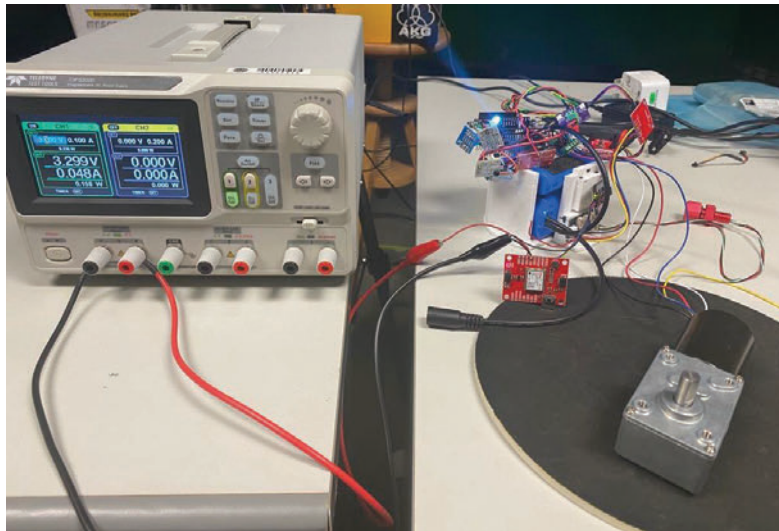


Figura 106 Prueba de la placa GNSS

El resultado de esta primera prueba no dio los resultados esperados y siguió persistiendo el mismo problema, por lo que se optó por desconectar todos los componentes del bus I2C e ir añadiendo de uno en uno componentes al bus conectándolos con el GNSS y el *power switch*. Con este proceso se observó que, con un solo componente en el bus aparte del módulo GNSS y el *power switch*, el sensor adicional funcionaba perfectamente, pero al añadir otro componente más al bus I2C, es decir, dos componentes conectados al bus aparte del módulo GNSS y el *power switch*, los sensores dejaban de funcionar correctamente.

Como consecuencia de este problema y debido a la falta de tiempo se optó por no contar con el GNSS para el resto de las pruebas.

5.2.4 Consumo de la batería de la electrónica

En este apartado se realizaron cuatro pruebas utilizando todos los componentes, a excepción del GNSS y el *power switch*, en las que el perfilador realizaba un ciclo de trabajo, representado en un diagrama de flujo en la figura 107.

El ciclo de trabajo consistía en subir el perfilador hasta la mitad de la altura final a la que se quería llegar, pararse, recoger lecturas de los sensores y quedarse en reposo durante dos minutos. Una vez había expirado el tiempo de reposo el perfilador subía hasta la altura final y volvía al estado de reposo durante tres minutos. Finalmente, una vez cumplidos los tres minutos de reposo, el perfilador bajaba hasta el punto inicial y volvía al estado de reposo durante tres minutos. Para comenzar el ciclo se esperaba a que cualquiera de los dos sensores de efecto Hall se activara.

Sistema de Control y Comunicaciones de un Perfilador Marino

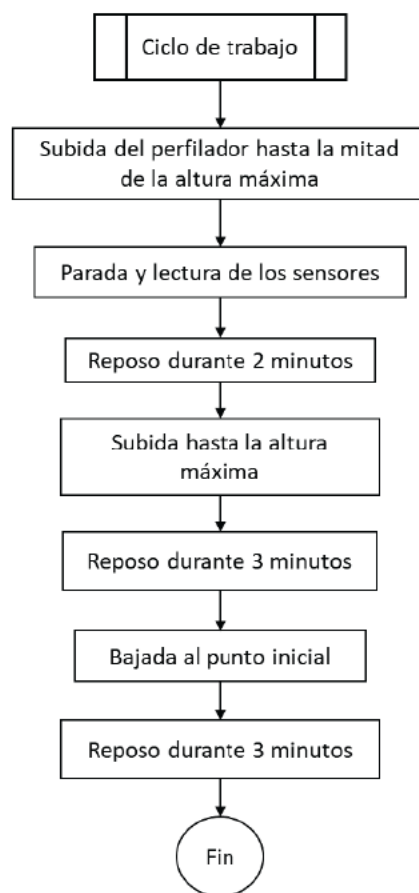


Figura 107 Diagrama de flujo del ciclo de trabajo de la prueba del consumo de batería

En la primera prueba, durante los tiempos de reposo el *Minipush button power Switch* se quedaba activo y simplemente hacíamos que el sistema se quedara en espera utilizando con un *delay*. En la segunda prueba se desactivaba el *Minipush button power switch* durante los tiempos de reposo, es decir, se le cortaba la alimentación al motor. En la tercera prueba, aparte de desactivar el *Minipush button power switch* durante los tiempos de reposo, se sustituyó el *delay* y se activó el modo de bajo consumo o modo *sleep* del microcontrolador. En la prueba 4 se añadió el *power switch* para cortar la alimentación a los componentes del bus I2C durante el tiempo de reposo ya que con el modo de bajo consumo no se podía realizar esta opción. A pesar de poner el microcontrolador en modo de bajo consumo el LED RGB y los sensores de efecto Hall seguían consumiendo alimentación, por lo que se realizó una última prueba conectando la alimentación de estos tres componentes a la alimentación del bus I2C para poder cortarla con el *power switch*.

Tras estas cinco pruebas se comprobó que el consumo en el estado de reposo era mucho menor en la última prueba. En la prueba 2, aunque no se vea reflejado la reducción de consumo de la batería de la electrónica, se está ahorrando un consumo de 23 mA procedente de la batería del motor. A continuación, en la tabla I se puede ver el consumo medido durante el estado de reposo en las cinco pruebas realizadas.

	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
Estado de reposo	0.052 A	0.052 A	0.038 A	0.018 A	0.012 A

Tabla I Datos de consumo de las pruebas

En estas pruebas no se ha incluido el GNSS por los motivos expuestos en el apartado anterior de este capítulo.

5.3 PRUEBA DEL SISTEMA DE COMUNICACIONES

Para el apartado de las comunicaciones, como se indicó en el capítulo 2 de esta memoria, se utiliza el protocolo FTP. Se realizó una prueba utilizando el software *FileZilla* [46] para crear el cliente FTP y se creó un servidor FTP al que conectarse (figura 108).

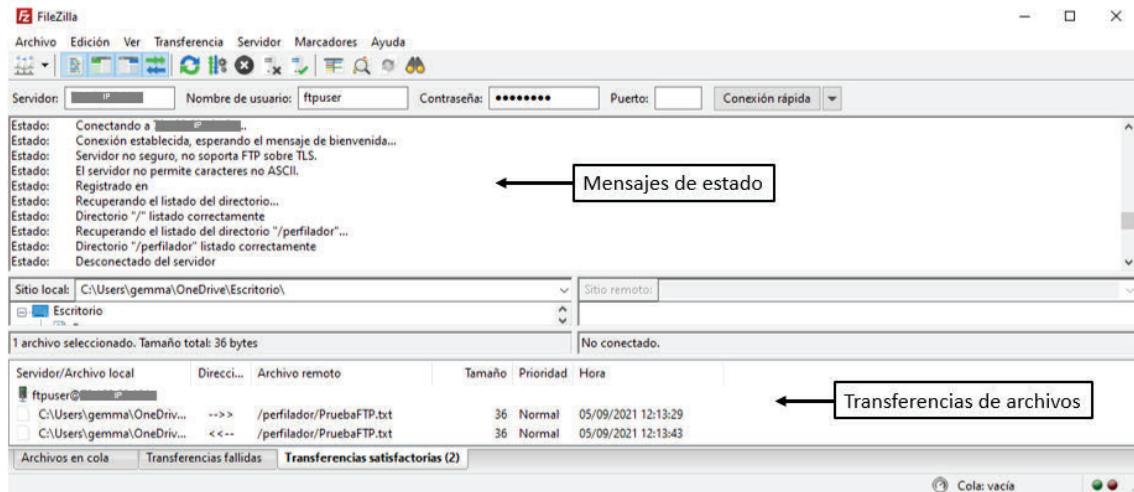


Figura 108 Prueba del sistema de comunicaciones utilizando FileZilla

Una vez se comprobó el funcionamiento del servidor y la conexión, se procedió a realizar la transferencia de archivos utilizando el Arduino. Como se puede ver en las figuras 109 y 110 se completó con éxito la prueba.

```
22:39:43.231 -> Función FTP
22:39:43.231 ->
22:39:44.039 -> OK - Connect to FTP server
22:39:44.161 -> OK - Create remote directory
22:39:44.321 -> OK - Local file system write
22:39:44.921 -> OK - Upload from local file system to FTP server
22:39:45.311 -> OK - Disconnect to FTP server
22:39:45.311 ->
22:39:45.311 -> FIN Función FTP
```

Figura 109 Mensajes del puerto serie del Arduino - prueba FTP



Figura 110 Comprobación de ficheros mediante FileZilla - prueba FTP

5.4 PRIMERA PRUEBA DE FUNCIONAMIENTO COMPLETA

Esta prueba se realizó en el laboratorio y se comprobó el funcionamiento del perfilador marino completo con todos componentes incluidos excepto por el GNSS. Para conseguir el movimiento vertical del perfilador se colgaron con dos cuerdas dos poleas a la estructura auxiliar existente fijada en el techo del laboratorio. Por ambas poleas se encontraba el cabo, en uno de sus extremos se encontraba el perfilador colgado y en el otro iba un contrapeso que le permitía subir y bajar (figura 111).



Figura 111 Prueba de funcionamiento completa del perfilador marino

Para esta prueba se utilizó el programa descrito en el capítulo 4 de esta memoria.

6 PRESUPUESTO

En este apartado se va a describir detalladamente el presupuesto del presente TFM, el cual se divide en recursos hardware, recursos software y recursos humanos.

Recursos hardware:

Para el desarrollo de este proyecto se han necesitado diferentes recursos hardware recogidos en la tabla 1, los cuales tienen un costo asociado.

Recursos hardware	Coste estimado
Arduino MKR GSM 1400	59,90 €
Arduino MKR MEM Shield	16,90 €
Acelerómetro	10,59 €
Sensor de micropresión	12,67 €
Sensores de efecto Hall	5,00 €
Sensor de luz ambiental	4,24 €
Sensor de corriente	6,74 €
Coulomb Counter	12,67 €
Sensor de temperatura exterior	50,85 €
Sensor de presión exterior	61,02 €
I2C Bus splitter	11,86 €
GNSS	59,28 €
Pantalla OLED	16,91 €
LED RGB	10,42 €
Resistencias	3 x 1,53 €
Motor	34,20 €
Minipush button power switch	2,96 €
Power Switch	5,89 €
Batería motor (12V)	34,82 €
Batería electrónica (3,3V)	24,90 €
Tapa de extremo de domo	29,66 €
Tapa de extremo de aluminio de 5 orificios	16,95 €
Brida de junta tórica	2 x 24,58 €
Juego de juntas tóricas	2 x 2,54 €
Tubo de acrílico fundido (220 mm)	0,34 €/ mm * 220 mm = 74,80 €
Tubo de acrílico fundido (150 mm)	0,34 €/ mm * 150 mm = 51,00 €
Impresión de piezas por la empresa Eyser Hidráulica	166,00 €
Cabo	0.02 €/m * 100 m = 2,00 €

Sistema de Control y Comunicaciones de un Perfilador Marino

Bobina de 1 kg de PETG	27,73 €
Total:	868,79 €

Tabla II Presupuesto de recursos hardware

Recursos software:

En la tabla 2 se recogen los recursos software utilizados en este proyecto.

Recursos software	Tipo de licencia	Coste estimado	Amortización
Autodesk Inventor Professional 2022	Universitaria	0,00 €	-
Arduino IDE	Pública	-	-
Office 365	Universitaria	0,00 €	Amortizada
PrusaSlicer	Pública	-	-
Total:			0,00 €

Tabla III Presupuesto de recursos software

Recursos humanos:

En la siguiente tabla está recogido el coste total de un contrato para un técnico titulado de máster, grupo TCP3, con una dedicación semanal de 20 horas. [47] Para sacar el coste por horas se divide la retribución mensual entre el número de horas trabajadas en un mes.

$$20 \frac{h}{semana} * 4 semanas = 80 \frac{horas}{mes}; \quad coste\ por\ horas = \frac{831,43\ €/mes}{80\ h/mes} = 10,393\ €/h$$

Recursos humanos	Coste mensual	Horas
Salario	10.393€/h	300
Total:		3117,90 €

Tabla IV Presupuesto recursos humanos

El coste total del proyecto será la suma de los costes por recursos hardware, software y humanos.

Recursos	Coste
Recursos hardware	868,79 €
Recursos software	0,00 €
Recursos humanos	3117,90 €
Costes totales sin IGIC:	3986,69 €
IGIC (7%):	279,07 €
Total:	4265,76 €

Tabla V Presupuesto total

Dña. Gemma Berciano Rodríguez declara que el presupuesto final de este TFM asciende a cuatro mil doscientos sesenta y cinco con setenta y seis céntimos (4265,76 €).

Las Palmas de Gran Canaria, a 08 de septiembre de 2021.

BERCIANO
RODRIGUEZ
GEMMA -
44748719B



Firmado digitalmente por
BERCIANO RODRIGUEZ
GEMMA - 44748719B
Fecha: 2021.09.07
23:22:43 +01'00'

Fdo. Gemma Berciano Rodríguez

7 CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES

En este TFM se ha diseñado y desarrollado un sistema de control y de comunicación de un perfilador marino con el fin de obtener datos a lo largo de una columna de agua. Se realizaron estudios sobre los sistemas monitoreo acuático de bajo consumo y sobre los hábitats de estudio del proyecto: la *Cymodocea nodosa* y los fondos de rodolitos.

Este ha sido un primer proyecto de colaboración entre los institutos universitarios IU-ECOAGUA, IUSIANI e IUMA, en el cual se ha desarrollado un dispositivo perfilador que permitirá alojar sensores de bajo coste, dotándolos además de la capacidad para obtener medidas en toda la columna de agua. Todo ello como alternativa a la metodología estándar, basada en costosos sensores oceanográficos que se ubican bien sobre el fondo o a una profundidad fija.

Este perfilador marino debe ser eficiente energéticamente ya que se quiere que el sistema sea autónomo durante varios días. Es por ello por lo que se incluyeron componentes para controlar la alimentación de algunos dispositivos y poder apagarlos en los tiempos de inactividad del perfilador marino.

En primer lugar, se realizó el diseño en 3D de una estructura que soporta todos los componentes electrónicos presentes en el perfilador, buscando la facilidad de acceso al microcontrolador y un montaje sencillo de la estructura. Posteriormente, se realizaron las etapas de programación.

Se han realizado pruebas de funcionamiento durante todo el proceso de diseño, primero de todos los módulos electrónicos por separado para verificar su funcionamiento, continuando con otras pruebas que los integraban poco a poco en el sistema comprobando a su vez el consumo del sistema cuando se añadían ciertos componentes.

Finalmente se completó con éxito una prueba de funcionamiento global de todos los elementos electrónicos y mecánicos que constituyen el perfilador. En esta prueba el perfilador marino realizó un ciclo de trabajo completo que ha permitido verificar el movimiento del motor, la lectura de los sensores y guardado de datos en la tarjeta SD.

7.2 TRABAJOS FUTUROS

Dentro del proyecto METEROSECO, el perfilador será probado en los próximos meses en el entorno marino de la bahía de Gando por los investigadores del Instituto Universitario de Acuicultura Sostenible y Ecosistemas Marinos (IU- ECOAQUA), bajo la supervisión del investigador principal del proyecto Fernando Tuya Cortés.

Para trabajos futuros de este proyecto se podrá rediseñar por completo la estructura del perfilador. En busca de un prototipo menos costoso, se dejarán de lado las piezas de la estructura exterior de *BlueRobotics* y se diseñará una carcasa de PVC.

Se considerará el diseño de un sistema electrónico a medida que se ajuste mejor a los requerimientos de bajo consumo y que permita prolongar el periodo de funcionamiento de las baterías. Adicionalmente, el diseño de un circuito impreso específico facilitará la reducción del cableado de los sensores, además de reducir los costes de fabricación y montaje de los distintos elementos del perfilador en la producción de pequeñas series de perfiladores.

Centrándonos en el campo de la programación, se consideraría utilizar un sistema de tiempo real dividiendo el ciclo de trabajo en tareas con distintas prioridades que se realizarían de manera

Sistema de Control y Comunicaciones de un Perfilador Marino

simultánea proporcionando un marco de programación más estructurado y flexible que el aportado con la programación en Arduino.

Por otro lado, para una segunda fase de este proyecto, tras realizar las pruebas en un entorno marino, se integrarán otros sensores que registren los datos de las variables de interés básicas: pH, oxígeno disuelto y radiación fotosintéticamente activa y se prescindirá del uso de la pantalla OLED, el LED RGB y los sensores de efecto Hall. Además, se terminará de implantar el sistema de comunicación para que sea seguro.

8 BIBLIOGRAFÍA

- [1] F. Tuya Cortés, “METOSERECO: Hacia una Nueva Metodología para el Estudio de Funciones y Servicios Ecosistémicos de Hábitats Submarinos,” pp. 1–20, 2020.
- [2] E. Mallon, “Underwater Arduino Data Loggers.” <https://thecavepearlproject.org/> (accessed Feb. 24, 2021).
- [3] Beddows, P.A., Mallon, and E.K., “Cave Pearl Data Logger: A Flexible Arduino-Based Logging Platform for Long-Term Monitoring in Harsh Environments,” *Sensors*, 2018. <https://www.mdpi.com/1424-8220/18/2/530/htm> (accessed Feb. 24, 2021).
- [4] R. Bardaji, A.-M. Sánchez, C. Simon, M. R. Wernand, and J. Piera, “Estimating the Underwater Diffuse Attenuation Coefficient with a Low-Cost Instrument: The KdUIO DIY Buoy,” *Sensors*, 2016. <https://www.mdpi.com/1424-8220/16/3/373/htm> (accessed Feb. 24, 2021).
- [5] J. Trevathan and R. Johnstone, “Smart Environmental Monitoring and Assessment Technologies (SEMAT)—A New Paradigm for Low-Cost, Remote Aquatic Environmental Monitoring,” *Sensors*, 2018. <https://www.mdpi.com/1424-8220/18/7/2248/htm> (accessed Feb. 24, 2021).
- [6] H. Maske, J. Ochoa, C. A. Jáureguí, and A. C. Avendaño, “Free-rising, tethered CTD profiler: increased vertical resolution and near surface profiling.” <https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.4319/lom.2012.10.475> (accessed Feb. 23, 2021).
- [7] U-blox, “Home | u-blox.” <https://www.u-blox.com/en> (accessed Aug. 18, 2021).
- [8] M. Z. Elso, P. G. Jiménez, N. G. Henríquez, and R. R. Romero, “La planta marina Cymodocea nodosa. Biología y Cultivo - Universidad de Las Palmas de Gran Canaria.” https://spdc.ulpgc.es/libro/la-planta-marina-cymodocea-nodosa_83404/ (accessed Aug. 25, 2021).
- [9] G. de C. (BOC), “Catálogo Canario de especies protegidas (Ley 4/2010 de 4 de junio),” pp. 15200–15225, 2010.
- [10] G. de C. (BOC), “Catálogo Canario de Especies Protegidas.” https://www.gobiernodecanarias.org/medioambiente/temas/biodiversidad/conservacion_de_especies/catalogo-y-seguimiento-de-especies-amenazadas/catalogo_de_especies_amenazadas_de_canarias/ (accessed Aug. 25, 2021).
- [11] F. E. Rodríguez, “Las praderas de fanerógamas marinas en Canarias y su diversidad.” <http://www.gobiernodecanarias.org/medioambiente/sostenibilidad/apps/revista/2001/21/265/index.html> (accessed Aug. 25, 2021).
- [12] F. Espino, F. Tuya, and R. J. Haroun, “Los sebadales de Canarias. Oasis de vida en los fondos arenosos,” pp. 1–66, 2008.
- [13] Disputación de Málaga, “Cymodocea (Cymodocea nodosa). Senda Litoral - Costa del Sol Málaga - Diputación de Málaga.” https://www.malaga.es/es/turismo/naturaleza/lis_cd-10247/cymodocea-cymodocea-nodosa-senda-litoral (accessed Aug. 25, 2021).
- [14] Instituto Español de Oceanografía, “Noticias - Instituto Español de Oceanografía.” http://193.146.153.5/es/web/baleares/noticias?p_p_id=ieolistadosestructuramain_WA

Sistema de Control y Comunicaciones de un Perfilador Marino

- R_IEOListadoContenidosPorEstructuraportlet&p_p_lifecycle=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column-7&p_p_col_count=2&_ieolistadosestructuramain_WAR_IEOListadoConteni (accessed Aug. 26, 2021).
- [15] IU-ECOQUA, “La ULPGC estudia el impacto positivo para el medio ambiente de los rodolitos, las famosas ‘palomitas de maíz’ de algunas playas canarias | ECOQUA.” <https://ecoqua.ulpgc.es/es/blog/la-ulpgc-estudia-el-impacto-positivo-para-el-medio-ambiente-de-los-rodolitos-las-famosas> (accessed Aug. 26, 2021).
- [16] C. González, “Fuerteventura - Los rodolitos de las playas de las Palomitas - GUÍA DE FUERTEVENTURA.” <https://fuerteventuraenimagenes.com/los-rodolitos-de-las-playas-de-las-palomitas/> (accessed Aug. 26, 2021).
- [17] D. Cressey, “The DIY electronics transforming research,” *Nature*, vol. 544, no. 7648, pp. 125–126, Apr. 2017, doi: 10.1038/544125a.
- [18] P. A. Beddows and E. K. Mallon, “Cave Pearl Data Logger: A Flexible Arduino-Based Logging Platform for Long-Term Monitoring in Harsh Environments,” *Sensors*, vol. 18, no. 2, p. 530, Feb. 2018, doi: 10.3390/s18020530.
- [19] Arduino, “Arduino MKR GSM 1400.” <https://store.arduino.cc/arduino-mkr-gsm-1400-1415> (accessed Aug. 12, 2021).
- [20] U-blox, “SARA-U2 series | u-blox.” <https://www.u-blox.com/en/product/sara-u2-series> (accessed Aug. 12, 2021).
- [21] Waspote, “Waspote Technical Guide.” <http://docplayer.net/79030139-Waspote-technical-guide.html> (accessed Aug. 02, 2021).
- [22] Microchip Technology, “ATECC508A | Microchip Technology.” <https://www.microchip.com/en-us/product/atecc508a> (accessed Aug. 12, 2021).
- [23] Arduino, “Arduino MKR Mem Shield | Arduino Official Store.” <https://store.arduino.cc/arduino-mkr-mem-shield> (accessed Aug. 12, 2021).
- [24] SparkFun, “SparkFun Triple Axis Accelerometer Breakout - MMA8452Q (Qwiic) - SEN-14587 - SparkFun Electronics.” <https://www.sparkfun.com/products/14587> (accessed Aug. 02, 2021).
- [25] SparkFun, “SparkFun Qwiic MicroPressure Sensor - SEN-16476 - SparkFun Electronics.” <https://www.sparkfun.com/products/16476> (accessed Aug. 02, 2021).
- [26] Velleman, “VMA313: SENSOR MAGNÉTICO DE EFECTO HALL (HOLZER) (2 uds.) – Velleman – Wholesaler and developer of electronics.” <https://www.velleman.eu/products/view?id=435540&country=us&lang=es> (accessed Aug. 02, 2021).
- [27] SparkFun, “SparkFun Ambient Light Sensor Breakout - APDS-9301 - SEN-14350 - SparkFun Electronics.” <https://www.sparkfun.com/products/retired/14350> (accessed Aug. 02, 2021).
- [28] SparkFun, “Zio Current and Voltage Sensor - INA219 (Qwiic) - SEN-15176 - SparkFun Electronics.” <https://www.sparkfun.com/products/15176> (accessed Aug. 02, 2021).
- [29] SparkFun, “SparkFun Coulomb Counter Breakout - LTC4150 - BOB-12052 - SparkFun Electronics.” <https://www.sparkfun.com/products/12052> (accessed Aug. 02, 2021).

Sistema de Control y Comunicaciones de un Perfilador Marino

- [30] Bluerobotics, "Blue Robotics' Story and Team." <https://www.bluerobotics.com/about/> (accessed Aug. 12, 2021).
- [31] Bluerobotics, "Fast-Response, High Accuracy ($\pm 0.1^{\circ}\text{C}$) Temperature Sensor." <https://bluerobotics.com/store/sensors-sonars-cameras/sensors/celsius-sensor-r1/> (accessed Aug. 02, 2021).
- [32] Bluerobotics, "Depth/Pressure Sensor for 300m (1000 ft) Depth with 2mm Resolution." <https://bluerobotics.com/store/sensors-sonars-cameras/sensors/bar30-sensor-r1/> (accessed Aug. 02, 2021).
- [33] Bluerobotics, "I2C Bus Splitter for Pixhawk Autopilot." <https://bluerobotics.com/store/comm-control-power/tether-interface/i2c-bus-splitter/> (accessed Aug. 02, 2021).
- [34] SparkFun, "SparkFun GPS Breakout - NEO-M9N, Chip Antenna (Qwiic) - GPS-15733 - SparkFun Electronics." <https://www.sparkfun.com/products/15733> (accessed Aug. 02, 2021).
- [35] SparkFun, "Zio Qwiic OLED Display (1.5inch, 128x128) - LCD-15890 - SparkFun Electronics." <https://www.sparkfun.com/products/15890> (accessed Aug. 02, 2021).
- [36] Pololu, "Pololu - Mini MOSFET Slide Switch with Reverse Voltage Protection, SV." <https://www.pololu.com/product/2811> (accessed Aug. 12, 2021).
- [37] SparkFun, "Qwiic Power Switch - SPX-16740 - SparkFun Electronics." <https://www.sparkfun.com/products/16740> (accessed Aug. 02, 2021).
- [38] "Protección de las baterías – Blog Baterías de litio." <https://www.bateriasdelitio.net/?p=54> (accessed Aug. 15, 2021).
- [39] Wikipedia la enciclopedia libre, "Protocolo de transferencia de archivos." https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_archivos#Modos_de_conexi3n_del_cliente FTP (accessed Sep. 05, 2021).
- [40] Arduino, "What is Arduino? | Arduino." <https://www.arduino.cc/en/Guide/Introduction> (accessed Aug. 12, 2021).
- [41] AutoDesk, "Software Inventor 2022 | Autodesk." <https://www.autodesk.es/products/inventor/overview#what-is-inventor> (accessed Aug. 13, 2021).
- [42] Prusa3D, "PrusaSlicer - Prusa3D - 3D Printers." <https://www.prusa3d.es/prusaslicer/> (accessed Aug. 13, 2021).
- [43] Bluerobotics, "All 4" Series - Blue Robotics." <https://bluerobotics.com/product-category/watertight-enclosures/4-series/> (accessed Aug. 02, 2021).
- [44] Filament2print, "¿Qué es el PETG?" https://filament2print.com/es/blog/49_petg.html (accessed Aug. 13, 2021).
- [45] 3D Natives, "¿Qué características tiene el nylon en la impresión 3D? - 3Dnatives." <https://www.3dnatives.com/es/caracteristicas-tiene-nylon-en-impresion-3d-120320202/> (accessed Aug. 14, 2021).
- [46] FileZilla, "FileZilla - The free FTP solution." <https://filezilla-project.org/> (accessed Sep. 05, 2021).
- [47] ULPGC, "Tabla de retribuciones del personal contratado en proyectos," p. 1, 2021.