



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño, desarrollo y puesta en marcha de un sistema
de toma de demoras electrónicas*

Grado en Ingeniería Mecánica

ALUMNO: Alejandro De Meer Méndez

DIRECTOR: José María Núñez Ortuño

CURSO ACADÉMICO: 2014-2015

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño, desarrollo y puesta en marcha de un sistema
de toma de demoras electrónicas*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

El presente trabajo surge con el objetivo de diseñar y desarrollar un sistema de posicionamiento para un buque durante la denominada Navegación Costera. El sistema está basado en la triangulación de líneas de posicionamiento tomadas a través de la información que pueden proporcionar los sensores inerciales, con un mayor grado de automatización de los procedimientos actuales.

Su desarrollo consiste en la combinación de dichos sensores inerciales con las tradicionales alidadas azimutales utilizadas en los buques. Los sensores inerciales permiten, mediante la integración de sus componentes, la medición precisa de ángulos relativos al movimiento al que se ven sometidos. Dichos ángulos relativos son referenciados a la giroscópica de bordo o a un compás magnético, lo que permite obtener referencias geográficas de la posición del buque. La información es procesada en una plataforma de desarrollo Arduino y enviada a un ordenador, en el que se ejecuta un programa que resuelve el problema de situación basado en trigonometría esférica.

El programa precisa de la interacción con las denominadas cartas náuticas electrónicas (ECDIS), como fuente de las variables geográficas de la costa, necesarias para la resolución del problema.

Los resultados del estudio y puesta en marcha del sistema pretenden dejar constancia de la posibilidad de automatizar la situación del buque en navegación costera con un sistema independiente del GPS.

PALABRAS CLAVE

Sistema de posicionamiento, Navegación Costera, Arduino, Sensor Inercial, Trigonometría Esférica

AGRADECIMIENTOS

- Al CC (CG-EOF) Gerardo González-Cela Echevarría por concebir la idea del trabajo y por su implicación en el desarrollo del mismo, ofreciendo distintas soluciones a los problemas planteados durante el desarrollo y la facilitación de su conocimiento en el campo de la Navegación.
- Al tutor del trabajo D. José María Núñez Ortuño, profesor en el Centro Universitario de la Defensa, Ingeniero de Telecomunicaciones e Ingeniero Técnico Industrial, por su tutorización, conocimiento, capacidad de resolución e implicación en el desarrollo del trabajo.
- Al Departamento de Navegación, en especial a la dotación del Simulador de Navegación por su predisposición y ayuda en el desarrollo del trabajo y periodo de prácticas.
- A la Escuela Naval Militar y al Centro Universitario de la Defensa, por su formación y capacitación para la realización del trabajo y la autorización del empleo de distintas instalaciones y laboratorios.
- A mi familia, por su educación y su continuo apoyo, en especial a mi padre.

CONTENIDO

Contenido	1
Índice de Figuras	5
Índice de Tablas.....	6
1 Introducción y objetivos	7
1.1 Contextualización.....	7
1.2 Motivación	8
1.3 Objetivos	9
1.4 Metodología de desarrollo	9
1.5 Entorno de Desarrollo	10
1.5.1 Hardware (HW)	10
1.5.2 Software (SW)	11
1.6 Especificación de requisitos.....	11
1.6.1 Requisitos funcionales:	11
1.6.2 Requisitos no funcionales	11
1.7 Riesgos	11
1.8 Organización de la presente memoria	12
2 Estado del arte	14
2.1 Navegación.....	14
2.1.1 Cartas Náuticas	15
2.1.1 Cartografía Digital	16
2.1.2 Cartas en Blanco	17
2.2 Conceptos generales.....	17
2.2.1 Agujas giroscópicas	19
2.2.2 Alidada.....	19
2.2.3 El taxímetro	20
2.3 Posicionamiento en Navegación Costera.....	21
2.3.1 Procedimiento de toma de demoras	21
2.3.2 Procedimientos para resolver el problema de situación	21
2.4 Posicionamiento por GPS.	23
2.4.1 Análisis del posicionamiento por GPS	23
3 Desarrollo del TFG.....	24
3.1 Alcance del TFG	24
3.2 Fases del Desarrollo	24

3.3 Análisis.....	25
3.4 Planificación y supervisión del proyecto	25
3.4.1 Planificación inicial	25
3.4.2 Seguimiento	26
3.4.3 Caminos críticos	26
3.5 Herramientas y entornos de trabajo	26
3.6 Principales elementos del HW	27
3.6.1 Placa Arduino	27
3.6.2 Sensor inercial	28
3.7 Diseño	31
3.7.1 Diseño preliminar	31
3.7.2 Diseño sistema	31
3.7.3 Diseño Arduino.....	32
3.7.4 Diseño mecánico.....	33
3.7.5 Diseño electrónico	34
3.7.6 Comunicación	34
3.8 Sistema de toma de demoras.....	35
3.8.1 Método de obtención del azimut.....	35
3.8.1 Calibración datos	39
3.8.2 Código Arduino	40
3.8.3 Importación de librerías	41
3.9 Sistema resolución problema de situación	41
3.9.1 Código Processing	41
3.9.2 Ortodrómica, solución analítica.....	42
3.10 Interfaz Gráfico	43
3.11 Montaje	45
3.12 Integración	45
3.13 Toma de decisiones	45
4 Pruebas y validación del Sistema	46
4.1 Plan de pruebas	46
4.1.1 Pruebas analíticas.....	46
4.1.2 Simulador de Navegación.....	52
5 Conclusiones y líneas futuras	53
5.1 Conclusiones	53
5.1.1 Fallos.....	54
5.2 Líneas futuras	54

5.2.1 Mejoras	54
6 Bibliografía.....	56
Anexo I: Conceptos	59
Anexo II: Diagrama de Gantt	60
Anexo III: Planos.....	63
Anexo IV: Código Arduino	66
Anexo V: Código Processing	69

ÍNDICE DE FIGURAS

Figura 1-1 Método de desarrollo iterativo e incremental (1)	10
Figura 2-1 Entorno Open CPN	17
Figura 2-2 Ejemplo de Demora (3)	18
Figura 2-3 Proyección Mercatoriana (7)	18
Figura 2-4 Alidada Azimutal TOKIMEC	19
Figura 2-5 Ejemplo de Taxímetro o repetidor de giroscópica.....	20
Figura 2-6 Toma de demoras con la alidada (8).....	21
Figura 2-7 Situación por tres demoras (9).....	22
Figura 3-1 El Sistema Inercial (14)	29
Figura 3-2 Campo Magnético Terrestre (15)	30
Figura 3-3 Diseños preliminares	31
Figura 3-4 Diseño del Proyecto	32
Figura 3-5 Diseño Arduino.....	33
Figura 3-6 Diseño electrónico Arduino	34
Figura 3-7 Inclinação de un cuerpo.....	36
Figura 3-8 Declinación Magnética (20)	38
Figura 3-9 Calibración de datos (22).....	39
Figura 3-10 Entorno Arduino	40
Figura 3-11 Entorno Processing	41
Figura 3-12 Trayectoria Ortodómica (24)	42
Figura 3-13 Programa Interfaz	44
Figura 4-1 Triangulación.....	47
Figura 4-2 Cuadrantes	48
Figura 4-3 Curvatura	49
Figura 4-4 Loxodrómica.....	50
Figura 4-5 Precisión	51
Figura 4-6 Sistema de toma de demoras.....	52
Figura 4-7 Prueba en el Simulador de Navegación	52
Figura 5-1 Patente cálculo de yaw (29)	55

ÍNDICE DE TABLAS

Tabla 3-1 División de tareas del proyecto26

Tabla 3-2 Herramientas y entornos de desarrollo27

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contextualización.

La navegación es el arte que se pone en práctica desde que se inicia la partida de la embarcación hasta su llegada al puerto de arribo. El arte requiere de destreza y de conocimientos técnicos en física, matemáticas, oceanografía, cartografía o astronomía. El objetivo es el de guiar al buque a través de las inclemencias del tiempo y los peligros de la navegación, de acuerdo con su finalidad comercial, de recreo, de vigilancia, militar o de rescate.

Uno de los principales planteamientos, comunes a todas las navegaciones, con el que todo navegante tiene que lidiar, e independientemente a la época a la que queramos referirnos, es el conocimiento de la situación o posición geográfica del buque durante la navegación. Descubrimientos como la información que ofrece la Estrella Polar, el desarrollo de la brújula, el sextante o el astrolabio, se han convertido en objetos en distintas épocas imprescindibles a la hora de salir a la mar.

El proyecto surge de la idea de automatizar el proceso de posicionamiento de un buque durante la Navegación Costera, aquella que se hace tomando referencias a los puntos conspicuos y objetos cercanos a la costa para obtener dicha situación. Coexisten diversos tipos de navegación en la actualidad, diversos métodos que proporcionan una información precisa y segura.

En la Escuela Naval Militar se enseñan, durante el primer curso en la asignatura de Navegación I, este método de posicionamiento tomando líneas de referencia geométricas entre la plataforma que navega y los puntos de la costa. El método se pone en práctica en las Lanchas de Instrucción, mediante el uso de la giroscópica de abordó y la carta náutica de papel tradicional, lo que lo hace un proceso lento comparado con otros sistemas, y en muchas ocasiones simples fallos de transmisión de datos generan grandes problemas, que resultan en la imposibilidad de obtener la situación. Esta práctica es extensible al resto de unidades de la Armada Española, por ser un ejercicio común de adiestramiento del equipo de navegación cuando se está siguiendo la navegación a través de una canal dragada, que son las vías limpias y de profundidad adecuada establecidas normalmente en las entradas y salidas de puerto.

Con el avance que posibilitan las cartas electrónicas y los sensores inerciales integrados en pequeños chips surge la idea de aprovechar ambos para automatizar el proceso de posicionamiento. La idea requiere, por tanto, desarrollar un programa que permita integrar la información que proporcionan los sensores inerciales y las cartas electrónicas, para conseguir como resultado la posición en la que se encuentra el buque.

El potencial de poder integrar en pequeños volúmenes este tipo de sensores aumentan las aplicaciones del mismo. Para el caso del presente trabajo, se pretende hacer uso de dichos sensores para integrarlos con las alidadas azimutales de los buques, a través de las cuales se toman las líneas de referencia a la costa, colocadas encima de los repetidores de giroscópica.

El sistema automatizaría el proceso de posicionamiento en la mar cercana a la costa, evitando la transmisión de información por medios humanos acústicos, sujetos a mayores errores y también a un mayor descontrol en el puente de gobierno. El motivo de este descontrol es el constante flujo de información que no es relevante para el mando en cuya responsabilidad recae el control de la navegación del buque.

La posibilidad de corregir y realizar recomendaciones a través de un sistema de toma de demoras automatizado aumenta de forma considerable. La información que se puede proporcionar es más fluida y, dependiendo del desarrollo del programa, se pueden incluir un mayor número de aplicaciones de cálculo, útiles para facilitar el conocimiento de precisión y fiabilidad de la situación obtenida.

1.2 Motivación

A nivel personal y profesional, como Oficial Alumno de la Escuela Naval Militar y estudiante de Ingeniería Mecánica, es muy atractiva la idea de afrontar un trabajo que permite plantear necesidades técnicas en el ámbito de navegación o de la carrera militar y poder profundizar en ellas para dar solución a tales necesidades a través de la ingeniería.

Este trabajo es un buen ejemplo de ello, y del enfoque didáctico y práctico que contiene. La posibilidad de poder desarrollar proyectos cuya utilidad se puede aplicar de forma directa, es un motivo de gran satisfacción personal. El hecho de que se pueda ver, y comprobar experimentalmente el mismo, supone un gran apremio y un continuo motor de perfeccionamiento del trabajo. Aprender a ofrecer soluciones de cálculo y lógica a través de la programación, conocer e investigar sobre los avances en tecnología, experimentar de forma práctica y siempre intentar perfeccionar los pequeños problemas que se plantean diariamente son motivo de estímulo para la realización del trabajo.

El tener la oportunidad de realizar pruebas de dichos proyectos a bordo de las Lanchas de Instrucción, el Simulador de Navegación, mejoran la variedad de perspectivas a la hora de enfocar distintos problemas, ofrecen muchas posibilidades de análisis del trabajo y demuestran la implicación de la institución.

Analizando el resto de posibilidades, el proyecto permite la materialización de una idea innovadora, pues es la primera vez que se une la tecnología al método tradicional de situación cerca de la costa. El proyecto supone un punto de inflexión en la integración de la tecnología propia de los sensores inerciales en el método posicionamiento geográfico y abre una puerta para la implementación de dichos dispositivos en un mayor número de aplicaciones tanto del campo propiamente náutico como del militar.

El trabajo mantiene su aspecto de complejidad que se puede ir desmenuzando con la respectiva investigación y tutorización, pero también se mantiene en unos marcos estudiados de tiempo y empleo de conocimiento.

Es un aliciente para trabajo en el proyecto realizar un trabajo que abre indudablemente las puertas a una futura capacidad de desarrollo y mejora de la solución obtenida.

1.3 Objetivos

La meta principal del proyecto se engloba en el propio título del trabajo: Diseño, desarrollo y puesta en marcha de un sistema de toma de demoras electrónica.

Partiendo del propio título podemos definir los objetivos secundarios, siempre dependientes de este objetivo principal.

- Conocer la plataforma de desarrollo y el lenguaje de programación propio de Arduino.
- Investigar y comprender el uso de una unidad inercial de movimiento o IMU (*Inertial Movement Unit*), y resto de sensores con los que puede trabajar Arduino.
- El estudio y establecimiento de la comunicación entre el PC, el sistema de toma de demoras y la cartografía electrónica digital o ECDIS (*Electronic Chart Display and Information System*).
- La materialización de un sistema práctico que complemente las alidadas azimutales de los buques con la automatización de toma de líneas de posición.
- Realizar un programa que permita el uso práctico de la toma de demoras electrónicamente para resolver la posición del buque.
- La puesta en marcha y análisis de fallos y errores del sistema generado, y su mejora o propuesta de mejora.
- Realizar un sistema de bajo coste cuyo desarrollo pueda llevar al estudio de su aplicación práctica en las unidades navales.

Es necesario obtener un resultado de bajo coste y suficientemente escalable y portable, para que su rango de aplicación sea más extenso.

Para la resolución se utilizarán las tecnologías referentes a las placas de Arduino y los sensores inerciales, la programación en Processing, y las cartas digitales del programa OpenCPN. Cada una de las tecnologías empleadas resuelve una necesidad concreta.

Las posiciones obtenidas se utilizarán como un sistema de apoyo a la navegación, que podrá compararse con la situación obtenida a bordo por otros medios como el GPS.

El control del sistema de toma de demoras se llevará a cabo mediante un interfaz de usuario, un sistema de control como pueden ser interruptores o botones, que transmitirán las órdenes de actuación, correspondientes al almacenamiento de la información del propio sensor inercial.

1.4 Metodología de desarrollo

El proyecto ha ido evolucionando de acuerdo con el método de desarrollo iterativo e incremental. El motivo del empleo de este método, es desde un principio, la diferenciación de las partes principales del mismo. El desarrollo del hardware y del software se ha encaminado de un modo independiente que también ha permitido dividir estos mismos bloques en pequeñas tareas de las que se van obteniendo resultados en cada campo de forma incremental.

En el caso del hardware, la comparación de sensores, placas, cableado, etc. permiten establecer prioridades y avanzar en distintas etapas de forma paralela. Esta partición del trabajo se puede llegar a ver mayor en el caso del software a desarrollar, pues, conforme se van completando cada una de las etapas que podemos considerar independientes, se acoplan al código principal, que a su vez va adquiriendo cuerpo de manera aditiva y creciendo y mejorando en consecuencia ante los problemas que van surgiendo. La división del trabajo en pequeños hitos a conseguir o tareas es básica. La utilidad de fijar objetivos secundarios día a día mejora la evolución del proyecto.

Al tiempo que la tecnología evoluciona también lo hace la forma de navegar. Mientras aumentan el número de sistemas alternativos, también crece la preocupación de la fiabilidad de cada uno de ellos, y el uso conjunto de todos ellos. La compatibilidad con la plataforma náutica de todos estos sistemas ayudan a establecer prioridades de fiabilidad de la información que se gestiona en un puente de gobierno.

Debido a la naturaleza del proyecto, se ha utilizado una investigación aplicada, basada en la resolución de un problema práctico. Una vez conocido el estado del arte en este campo, se pasa a la investigación bibliográfica del funcionamiento. Hasta este punto el proyecto se mantiene meramente teórico y de investigación. Pero es entonces cuando una vez investigado y acotado el campo de trabajo, se puede dividir este en los principales campos que dan cuerpo al trabajo.

Cada pequeño objetivo del proyecto tiene su propio método de enfoque y trabajo y el avance en cada una de las pequeñas tarea sirve para aumentar y poder medir el avance respecto de la totalidad del trabajo.

Finalmente se procede a la aplicación del conocimiento teórico adquirido en la creación de una solución a los problemas planteados, mediante programación de software, utilizando distintas técnicas de desarrollo.

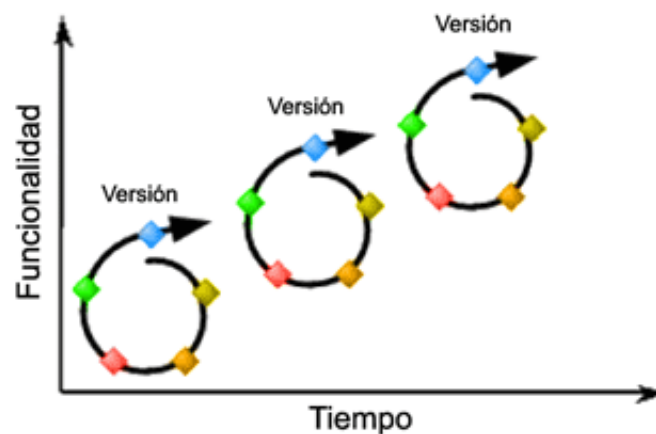


Figura 1-1 Método de desarrollo iterativo e incremental (1)

1.5 Entorno de Desarrollo

El marco que define el campo de actuación del proyecto tiene lugar en torno a los repetidores de giroscópica de los buques, que normalmente se encuentran situados en las bandas del puente de gobierno. El objetivo es desarrollar sobre dichos repetidores la instalación de los sensores inerciales.

Para conseguir dicho propósito es necesario dejar claros ciertos aspectos acerca de los dos grandes apartados en que se divide el proyecto, por un lado el hardware, y por el otro el software, para aclarar conceptos y poder decidir cómo enfocar el problema, de que medios disponemos y cual el mejor modo de emplearlos.

1.5.1 Hardware (HW)

Respecto a la parte tangible del sistema desarrollado, los principales componentes son los componentes básicos que permitirán desarrollar el proyecto debido a sus aplicaciones, está compuesto principalmente por la placa de desarrollo de Arduino, una IMU y elementos auxiliares.

1.5.2 Software (SW)

En cuanto a los programas de los que se ha hecho uso para realizar el proyecto en el presente proyecto se pretende especificar los mismos.

- Entorno de desarrollo Arduino: consiste en un editor de texto para escribir código y se conecta al hardware Arduino para cargar programas y comunicarse con ellos.
- Processing: un entorno de desarrollo bajado en Java con el que se pueden producir proyectos multimedia de forma digital.
- OpenCPN: programa para trabajar con cartas náuticas digitales como herramienta de planificación de navegación.

1.6 Especificación de requisitos

En este apartado se pretenden establecer los requisitos del sistema a desarrollar. Se pretende con ello establecer las condiciones que el sistema debe plasmar para que se pueda cumplir el objetivo principal del trabajo. En nuestro caso para obtener la intersección de las demoras electrónicas.

Diferenciaremos dentro de estas condiciones entre los requisitos *funcionales*, aquellos que definen las características referentes a qué debe hacer el sistema, y *no funcionales*, cuyo propósito es delimitar el marco de accesibilidad, mantenimiento, calidad, etc. así como el modo en que se deberían realizar determinadas acciones.

1.6.1 Requisitos funcionales:

- El sistema debe ser capaz de obtener una lectura angular relativa, obtenida del movimiento de la IMU.
- La información obtenida en dicho sistema debe ser procesable y es necesario que pueda trabajarse con ella desde otro programa.
- Es necesario recibir la información de las referencias de la costa a través de un programa de cartografía electrónica.
- El sistema deberá presentar la validez de la situación obtenida con una referencia de precisión y fiabilidad de la misma.
- El programa deberá tener en cuenta los posibles fallos tanto de lógica como del propio sistema en el que se ejecute.
- El sistema debe ser portátil y con un carácter polivalente que le permita poder ser instalado en diferentes unidades

1.6.2 Requisitos no funcionales

- Dada su funcionalidad se persigue evitar las complicaciones y buscar un prototipo que se adecúe a la sencillez, para mejorar el aspecto e interés económico de la creación de un prototipo con dicha aplicación.
- Se debe pensar en el trabajo con el enfoque a su aplicación en unidades navales, teniendo en cuenta las condiciones de la propia plataforma y del medio para el que está destinado.
- El tiempo es un factor limitante en el desarrollo de dicho proyecto que se debe optimizar de acuerdo a los resultados que se deseen obtener.

1.7 Riesgos

Podemos mencionar la palabra riesgo cuando nos referimos a todos aquellos factores o situaciones que puedan llegar a alterar el desarrollo o funcionamiento del sistema de forma crítica. Este apartado recoge un análisis que se hace inicialmente con la evaluación de los posibles riesgos y aquellos que van apareciendo según avanza el proyecto. En nuestro caso la mayoría estará centrada en la fase de

puesta en marcha y pruebas. Se pueden encontrar riesgos dentro de todos los puntos de estudio y desarrollo:

- El sistema está expuesto al medio exterior y sus complicaciones, por encontrarse los taxímetros sobre los que va el sistema en los alerones exteriores de las bandas.
- La fragilidad de los materiales y estructuras seleccionadas deberá estar a prueba de cualquier posible golpe que pudiera dejar inoperativo el sistema, en mayor medida por la fragilidad de las conexiones realizadas.
- La importancia del tiempo: al utilizar elementos electrónicos como pequeños procesadores y sensores es importante controlar los tiempos de las distintas mediciones y reducir al máximo las pérdidas de tiempo en el empleo del sistema.
- La importancia de la calibración de los sensores empleados es notable, la brusquedad con que realicemos los movimientos afectan directamente a la precisión del sistema.
- La dependencia de un único sistema de navegación supone un riesgo para la seguridad y precisión de la navegación. Es necesario conocer los distintos métodos de posicionamiento y los posibles fallos de los mismos para minimizar los efectos de fallo de los sistemas.

1.8 Organización de la presente memoria

Como finalización de contextualizar el trabajo y definir los principales puntos de actuación u objetivos del mismo, en el presente punto se describe la organización de la memoria y su desglose en sus diferentes apartados.

En el apartado 2, describe el estado del arte de la navegación costera. En el apartado 2.1 se introducen las diferentes metodologías de navegación de la actualidad, profundizando en la cartografía empleada. En el apartado 2.2 se recogen una serie de conceptos generales, relacionados con los sistemas de abordaje empleados durante la navegación costera y el conocimiento de ciertos conceptos de trayectorias náuticas. En el apartado 2.3 se explica el procedimiento actual para tomar demoras y resolver el problema de situación. El apartado 2.4 se puntualiza la navegación por GPS, para conocer el método de navegación más común de la actualidad y realizar un análisis de su coexistencia con el resto de sistemas.

En el apartado 3 se detalla el desarrollo del proyecto. Inicialmente, en el apartado 3.1 se define el alcance del mismo. Seguidamente en el apartado 3.2 se enumeran las fases de desarrollo. Posteriormente se realiza un análisis del desarrollo en el apartado 3.3. En el apartado 3.4 y 3.5 se despliega la planificación y seguimiento que se han llevado acabo y se exponen las diferentes herramientas y programas que han ayudado a la elaboración del trabajo. A continuación, en el apartado 3.6 se definen los principales electos del hardware como puntos de aclaración, el Arduino y el sensor inercial. En el apartado 3.7 se presenta el diseño del sistema de toma de demoras, incluyendo los esquemas de desarrollo preliminar de los bloques en que se divide, definitivo, planos electrónicos y mecánicos y se realiza una explicación del modo en que se comunican los distintos bloques. En el apartado 3.8 se despliega un estudio sobre la obtención del ángulo azimutal y su expresión a analítica a partir del sensor inercial, y también se detalla la calibración de datos y el programa que se usa para obtener dichos resultados. El apartado 3.9 se centra en el método de obtención analítico de la posición del buque y su desarrollo en el entorno de programación elegido. En el apartado 3.10 se explica el interfaz gráfico. Finalmente, en los apartados 3.11, 3.12, 3.13, se hacen pequeños apuntes respecto al montaje, la integración de los sistemas y la toma de decisiones del trabajo.

En el apartado 4 figuran los resultados obtenidos durante el desarrollo del trabajo, puntualizando las pruebas realizadas. Se expone el plan de pruebas con los estudios realizados y, posteriormente se analizan los resultados.

En el apartado 4.1.2 se exponen las conclusiones del proyecto, así como las líneas de trabajo futuras. En el apartado 5.1 se valoran los resultados finales de acuerdo con los requisitos que se habían establecido. Por otra parte, en el apartado 5.2 se tienen en cuenta las posibles mejoras y líneas de actuación con objeto del desarrollo de nuevos métodos de posicionamiento.

En el apartado 6 se cierra el proyecto presentando la bibliografía consultada y empleada tanto en la investigación como en el desarrollo del proyecto.

Para finalizar se pueden consultar los anexos del proyecto referentes al sistema de toma de demoras. El Anexo I: Conceptos, referido a un conjunto de términos y siglas o acrónimos empleados a lo largo del trabajo. El Anexo II: Diagrama de Gantt, en el que se incluye el diagrama de organización del proyecto y su explicación. El Anexo III: Planos, contiene los planos de diseño del sistema de toma de demoras, como una posible solución al sistema. El Anexo IV: Código de Arduino, el Anexo V: Código de Processing, esconden el lenguaje empleado en cada caso para generar el código de cada uno de los sistemas.

2 ESTADO DEL ARTE

2.1 Navegación

La Navegación puede definirse como la ciencia que tiene por finalidad obtener la situación de un móvil en cualquier momento, así como enseñar la conducción del mismo entre dos puntos del globo terrestre con las mayores prestaciones de seguridad y rapidez.

Todo este conocimiento ha sido transmitido a lo largo de distintas generaciones y pueblos de la antigüedad, desde los grandes mercaderes fenicios, constructores egipcios o el poderío naval del Imperio Romano. La importancia del carácter marítimo de muchos pueblos es determinante en muchos aspectos económicos, sociales y culturales. En el caso de España, son muchos los acontecimientos acaecidos en la mar los que han determinado históricamente su destino, como Lepanto o Trafalgar. Como cualquier otra ciencia, la navegación ha sufrido un desarrollo y evolución considerables. Y su estudio varía mucho de acuerdo con la época que se analice. Sin embargo, a pesar de esta diversidad, existen unos planteamientos básicos de navegación, comunes a todas las épocas.

Son diversos los métodos que conocemos en nuestros días, que nos proporcionan una navegación precisa y por tanto segura. Se nombran a continuación un conjunto de distintos métodos de navegación:

- La Navegación Costera, se define como la que se efectúa a la vista de la propia costa, cuyos puntos destacados se pueden utilizar como referencias para situarse.
- La Navegación por Estima, se define como aquella que se efectúa, sobre la carta o a través del correspondiente cálculo matemático, suponiendo el conocimiento de la velocidad y rumbos de avance del buque. Siempre complementa al resto de métodos de posicionamiento y sirve de ayuda para el cálculo de rumbos efectivos y, por tanto conocer el abatimiento y la deriva.
- La situación por Métodos Analíticos; obtenida matemáticamente.
- La Navegación Astronómica, basada en la observación de los cuerpos celestes, como las estrellas el Sol, los planetas y la Luna.
- La Navegación por radio o radionavegación, utilizando sistemas de radio emplazados en ubicaciones conocidas que emiten mediante determinadas señales cierta información. (GNSS, VOR, DME, ILS, etc.)
- La Navegación Inercial, por medio de la integración de los datos ofrecidos por sensores inerciales, conociendo posición, velocidad y altitud inicial, y midiendo variación de altitud y aceleración.

Cabe destacar, que la navegación en la que se centra el estudio del presente proyecto es la Navegación Costera. Este tipo de navegación consiste, como se ha enunciado anteriormente, en una técnica de posicionamiento a la vista de la costa, ya sea por medios visuales (alidades azimutales, sextantes, etc.) o por medios electrónicos (transpondedores, radares, etc.). La situación se obtiene por referencias o líneas de posición, a puntos conspicuos, u objetos cuya situación es conocida y están representados en la carta náutica. Aunque este tipo de navegación resulta sencilla, requiere grandes precauciones y vigilancia, cualquier descuido puede causar la pérdida del barco, precisamente por ser la navegación cerca de la costa que es donde los bajos y otros accidentes son más numerosos. (2)

Podemos decir que la Navegación es una ciencia eminentemente práctica, pero el estudio y dominio de la teoría es un requisito imprescindible para toda persona bajo cuya responsabilidad pretenda actuar, con conocimiento de causa, en el puente de gobierno de un buque.

Como se ha introducido anteriormente, la Navegación requiere de otras disciplinas para establecer sus principios y cumplir sus finalidades. Ciencias como las matemáticas, la astronomía, la física, la meteorología o la electrónica, se consideran como pilares básicos sin los cuales seguiríamos pensando en la navegación como en una época de leyenda. Es una ciencia que sigue abierta a los descubrimientos y mejoras científicas, con fundamentos que se incorporan en un ámbito tanto práctico como teórico.

A pesar de toda la literatura o fantasía desarrollada en torno al concepto de navegación y de la mar, no se puede dudar del eminente carácter dinámico de la Navegación. Su estudio obliga a una continua actualización tanto práctica como de conocimientos teóricos. (3)

Pequeños errores cerca de la navegación cerca de la costa pueden tener graves consecuencias para la seguridad de la plataforma náutica. Por esta razón y otras parecidas, ninguna preocupación será exagerada, aun cuando la derrota a seguir sea perfectamente conocida. Por ello a bordo de los buques las metodologías para obtener la situación geográfica son numerosas y siempre se pueden y deben contrastar.

Uno de los principales requisitos para realizar navegaciones y verificar su seguridad es el empleo de la cartografía náutica. El empleo del material náutico que se debe emplear está regulado de acuerdo con la zona de navegación que le haya sido asignada. (4)

2.1.1 Cartas Náuticas

La carta náutica es un documento o publicación gráfica a través de la cual se representan zonas de extensión variable de los mares y costas. Cualquier tipo de carta como queda definido anteriormente es una ley de transformación, que permite el traslado de los puntos de la Tierra a un plano donde las coordenadas angulares pasan a ser lineales. Las cartas son de diversos tipos a fin de cubrir las distintas necesidades del navegante. Lo ideal de todas ellas es que tuvieran propiedades como la conservación de la forma, de los ángulos, que tuviese la misma escala, que se guarde la proporcionalidad de superficies representadas... Sin embargo, tener todas estas propiedades no es posible, cada tipo de carta cubre algunos de los anteriores requisitos, pero no todos. (5)

La solución a la imposibilidad de convertir el esferoide terrestre no desarrollable en un plano, se resuelve por los siguientes métodos o la combinación de ellos:

- Proyección de una parte de la superficie.
- Desarrollo de la parte sobre la que se ha proyectado previamente el esferoide.
- Fórmula matemática, para conseguir un sistema de proyección o alguna propiedad.

La compilación de Cartas Náuticas facilita una navegación de precisión y segura. Las distintas naciones publican dichas cartas, en las que se representan las líneas costeras incluyendo faros,

balizamiento marítimo, profundidades y resto de información de interés para la zona de navegación; esta información que se complementa como los almanaques náuticos.

Si la Tierra tuviera forma de cubo, por ejemplo, la representación plana de la misma sería muy simple. Desarrollaríamos las seis caras de un cubo sobre un plano, con dimensiones proporcionales a las caras de la Tierra, según la escala deseada para su representación. En cada cara representaríamos la correspondiente cara del cubo de la Tierra y obtendríamos un mapa exacto y sin deformaciones.

Pero la Tierra es esférica, y para ser más exactos diremos que realmente se trata de un esferoide al ser ligeramente achatada en los polos y abultada en el ecuador. Las proyecciones que se realizan para representar la superficie terrestre sobre un plano se pueden clasificar bajo diferentes criterios

El Instituto Hidrográfico de la Armada Española, se encarga de mantener actualizadas sus cartas náuticas a través de avisos, correcciones y anuncios de cambios. La importancia de actualización de estas cartas es de especial importancia, el usuario debe mantener al día efectuando las correspondientes correcciones. Es de vital importancia para un buque el estudio de la ruta seleccionada, Y teniendo en cuenta todos los posibles peligros a la navegación.

Las zonas más importantes corresponden con profundidades someras, las rocas sumergidas, los naufragios, etc. a través de símbolos internacionalmente aceptados se representan dichos peligros o ayudas a la navegación que permiten al navegante reconocer la zona de navegación en que se encuentra y tomar las medidas necesarias para asegurar y respetar las reglas de navegación marítima. En cada caso las cartas se adecúan a la profundidad del agua, la cantidad de tráfico y la complejidad del fondo marino.

Una de las características más importantes y diferenciadoras de los distintos tipos de proyección es la deformación o distorsión que sufre la superficie de la Tierra al ser representada. La única manera de representar la superficie terrestre sin deformación o distorsión alguna sería realizando su proyección sobre una superficie esférica, pero mapas o cartas náuticas esféricas no serían en absoluto prácticos. A esta deformación que sufre cualquier representación plana de la superficie terrestre se le llama anamorfosis. (6)

El Catálogo de Cartas Náuticas es una publicación que contiene una relación de cartas náuticas en vigor junto con su número, título, zona que abarca, escala, fecha de publicación, edición. Nos permite seleccionar las cartas y tener preparadas las zonas por las que está previsto navegar.

2.1.1 Cartografía Digital

El programa OpenCPN es un programa libre, que consiste en un Plotter de Navegación. Está diseñado para un uso continuo en el puente de mando de un buque, y permite al operador seguir fácilmente su posición y programar sus navegaciones a través de precisas cartas náuticas electrónicas (ECDIS).

El programa tiene funciones adicionales para predecir mareas y corrientes, así como información importante para un buque en la zona de navegación. Su uso se ha extendido enormemente por su carácter abierto, ya que tanto su descarga como su posterior uso son gratuitos. Es un programa conciso, pensado para la navegación diaria y con una interfaz intuitiva de cara al usuario.

Open CPN admite diversos formatos de cartografía que se pueden comprar y descargar. Un ejemplo de descarga gratuita es a través de la Administración Nacional Oceánica y Atmosférica o NOAA (National Oceanic and Atmospheric Administration), el problema es que no recoge la cartografía náutica de toda la Tierra.

El programa permite recibir señales de otros sistemas como el GPS o el AIS para mostrar las posiciones de los buques. La precisión de estas situaciones depende de la precisión del propio sistema del que se reciba la información. En nuestro caso dependerá de la precisión de la solución obtenida.

Las cartas que utiliza el programa son cartas vectoriales, no *raster*, de las que se obtiene mayor información de los posibles objetos y también más precisión a la hora de disminuir la escala.

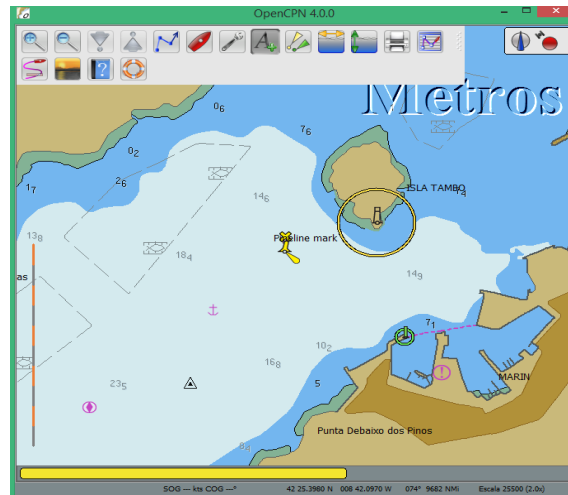


Figura 2-1 Entorno Open CPN

2.1.2 Cartas en Blanco

En el ámbito de la navegación se utilizan diferentes representaciones de la esfera terrestre en el plano según la zona de navegación. Es de uso extendido también la creación de **cartas en blanco**, se engloban bajo este nombre aquellas cartas que no tienen representación del contorno de la costa o del relieve submarino. Se utilizan para ciertas aplicaciones en las que únicamente es necesaria la resolución gráfica de algún problema náutico.

Pueden abarcar una muy amplia extensión y en ocasiones están emitidas por los propios organismos oficiales de Servicios Hidrográficos. También pueden referirse a una pequeña zona, en este último caso se considera plana la superficie abarcada y se puede improvisar una carta en blanco.

2.2 Conceptos generales

La importancia del modo en que se representa la superficie terrestre es un factor crucial para conocer la forma de posicionamiento de un buque en la mar. La elección de la proyección con la que se quiere trabajar determina el modo de trabajo y de obtención de la información.

La información que se representa en dichas cartas para obtener la situación se conocen como **líneas de posición**, que son el lugar geométrico sobre el cual está el barco en un momento dado. En la navegación costera, dicha información se deducen de las observaciones efectuadas a los puntos fijos en tierra. De la intersección de dos o más de estas líneas se obtiene la posición del barco.

Conviene destacar que la observación queda perfectamente definida cuando tres o más líneas de posición se cortan en un punto. La información que se obtiene del corte de dos líneas de posición puede ser inexacta, siempre se podrá obtener una intersección incluso si esta es errónea. A través de las tres líneas de posición se verifican cada uno de los cortes dos a dos generando normalmente el triángulo de posición, que en el mejor de los casos se aproxima a un punto al coincidir estos cortes dos a dos todos en una misma posición.

El conjunto de líneas que se pueden emplear en navegación costera son variados: de demora, de arco capaz, de distancia, isobáticas, etc. Se pueden combinar aun siendo de distinta especie de acuerdo con los elementos de observación de los que se disponga.

Centraremos el estudio de las líneas de posición en la definición y forma de obtener la situación a través de las demoras.

Para entender esta metodología es básico tener claros ciertos conceptos. El **círculo máximo**, en trigonometría esférica, es la intersección de una esfera con un plano que contenga su centro, una circunferencia máxima en su superficie. La circunferencia divide la esfera en dos hemisferios iguales y la distancia entre dos puntos unidos por el menor de los arcos de círculo máximo que los contienen es la menor distancia entre dos puntos, que se denomina **distancia ortodrómica**.

La posición de un punto en el horizonte queda definida por la dirección en la que se encuentra. El arco de círculo máximo que los une precisa de esa dirección y se la conoce por línea de **demora**, o simplemente demora. La **marcación** es el ángulo que forma el meridiano que pasa por el observador y la línea de demora antes citada, el valor de dicho ángulo identifica a la demora correspondiente.

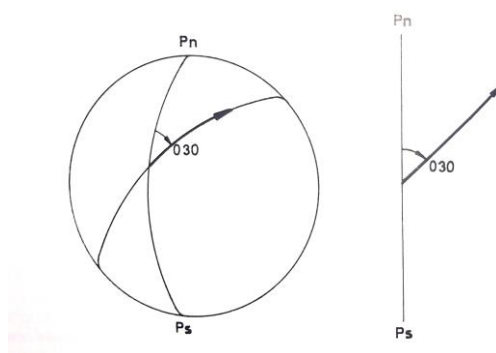


Figura 2-2 Ejemplo de Demora (3)

Las cartas de papel tradicionales, con las que se trabaja en la mayor parte de los casos, se representan normalmente con la **proyección mercatoriana**, que es una proyección cilíndrica circunscrita en el Ecuador y ortomórfica. Esta última característica se debe a la modificación de la proyección cilíndrica mediante un sistema analítico, la modificación permite que la separación entre los meridianos y paralelos sea tal que guarde la misma proporcionalidad en el sentido Norte Sur y en el sentido Este y Oeste, como se puede observar en la Figura 2-3. Los paralelos quedan en toda su extensión a igual distancia del Ecuador, dando lugar a un sistema de coordenadas rectangulares, con lo que los ángulos medidos en la superficie tienen la misma representación en la carta

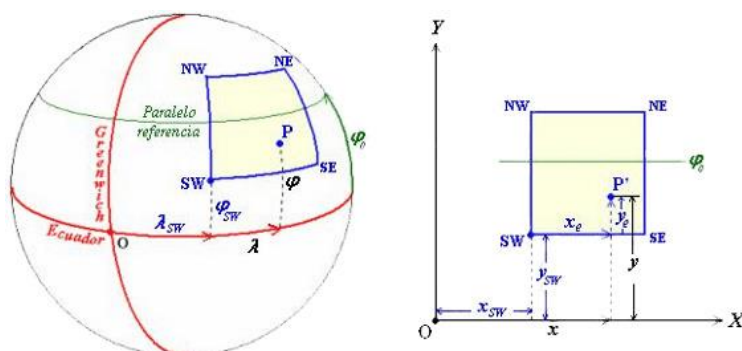


Figura 2-3 Proyección Mercatoriana (7)

Esto supone que la representación de una **trayectoria loxodrómica**, aquella que se realiza siguiendo la línea que une los puntos cortando a todos los meridianos con el mismo ángulo, sobre este tipo de proyección, es una recta, lo que facilita la planificación de la navegación y la obtención de información en la carta. El que la representación de la loxodrómica sea una recta facilita la gestión de

la navegación, ya que es más sencillo trazar trayectorias y asignar las respectivas órdenes de rumbos al timonel.

Así mismo, en el caso de los círculos máximos (como ha quedado definido, la demora es un círculo máximo) que rodean la esfera terrestre, las denominadas trayectorias ortodrómicas, su representación en las cartas mercatorianas es una curva. Normalmente se puede aproximar a una recta como en el caso de las anteriores únicamente cuando las distancias son pequeñas.

El motivo de esta aproximación es que el error es casi inapreciable si nos encontramos realizando navegación costera. Cuanto mayor sea la distancia y mayor la lejanía respecto del Ecuador más notable es la diferencia entre ambas trayectorias geométricas.

La toma de demoras precisan a bordo de un medio visual y de la posibilidad de realizar una medición angular respectivamente, el conjunto está compuesto por agujas giroscópicas, alidadas y taxímetros.

2.2.1 Agujas giroscópicas

La giroscópica es un tipo de aguja a través de la cual se determina el rumbo verdadero sin estar sometida a la influencia de los elementos magnéticos del buque. La rigidez y precisión son los dos parámetros que caracterizan a los giróscopos.

Se denomina **rigidez** o **inercia giroscópica** a la propiedad de mantener su eje apuntando a una dirección fija en el espacio con independencia de los movimientos de la plataforma que los soporta. El parámetro está condicionado por diferentes circunstancias que sirven para analizar también el sensor inercial que se empleará en el sistema desarrollado en el proyecto:

- La velocidad de giro
- La libertad de movimiento en los diferentes ejes
- Que el centro de gravedad esté interactuando en todos los ejes
- Las fricciones y rozamientos del sistema
- Posibles fuerzas perturbadoras.

La **precisión** se define como la reacción del giróscopo, de acuerdo con las leyes definidoras de las fuerzas que intentan alterar su estado de rigidez.

2.2.2 Alidada

Otra entidad cuya definición es necesario conocer es el de la **Alidada**. Es el aparato utilizado para tomar marcaciones, materializando la visual dirigida a un punto o astro. La marcación se lee en la graduación del soporte sobre la cual se coloca, que normalmente es el taxímetro del buque.

La marcación es por tanto un azimuth terrestre que se define como astronómico y se cuenta de igual forma que éstos y los rumbos.



Figura 2-4 Alidada Azimutal TOKIMEC

Está esencialmente compuesta por una regla mecánica horizontal en cuyos extremos hay dos pínulas y en el otro un lado ocular, desde donde el observador enfocarí. En la prolongación de la línea determinada por ambas pínulas, y en el lado opuesto al del ocular hay un espejo que permite reflejar la lectura del taxímetro. Existen diversos sistemas diseñados según el soporte para el cual estén destinados. Entre las numerosas variantes se pueden encontrar incluso aquellas que tienen un antejo acoplado al sistema óptico o modelos con diferentes mejoras para balances anormales o reflejos. Las alidades no poseen dimensiones estandarizadas, las dimensiones son propias del fabricante.

En el caso de la Armada Española, aquella con la que se trabaja más a menudo es la Alidada Azimutal, la marcación que se obtiene está referida al azimut y por tanto es equivalente a la demora. Este tipo de alidada logra por medio de un dispositivo óptico, compuesto por un prisma y un espejo, la posibilidad de visualizar simultáneamente la imagen del objeto y la de la graduación, en el plano determinado por las pínulas

Las alidades están construidas para adaptarlas a los repetidores de las giroscópicas producidas por las respectivas firmas comerciales. Son muy variados los distintos aparatos que se utilizan para medir este ángulo, y la mayoría de ellos cuentan con un aro giratorio en el que están marcados sus extremos y a través del cual se define la visual. El sistema está situado sobre otro aro o disco graduado, en este caso el disco es fijo y en él se refleja el desplazamiento angular respecto a una línea de referencia.

2.2.3 El taxímetro

El taxímetro se compone de un pedestal o basada en cuya parte superior hay un mortero sobre el cual gira la alidada. El mortero dispone de una graduación móvil que se puede fijar en la posición que se quiera según la demora que se desee obtener.

Los taxímetros con los que se suele trabajar son repetidores de la aguja giroscópica del buque, esto significa que el dato obtenido es la demora verdadera.

Los taxímetros suelen estar situados en los alerones del puente de gobierno, para cubrir ambas bandas con su línea 0°-180° paralela a la crujía del buque, de este modo es más sencillo ofrecer como dato al equipo de navegación la marcación. Este paralelismo se puede comprobar, avistándose simultáneamente un objeto desde la lejanía, las lecturas que se obtienen de dicho objeto desde la aguja magistral y el taxímetro deben ser la misma.



Figura 2-5 Ejemplo de Taxímetro o repetidor de giroscópica

2.3 Posicionamiento en Navegación Costera

Una vez analizados los conceptos y sistemas de uso, se procede a la explicación de los procedimientos por los que se resuelve el problema de situación.

Se divide dicho procedimiento en dos bloques principales, el primero el de toma de demoras, por el que se obtiene la información geográfica del entorno, y posteriormente, el método de cálculo de la intersección de las mismas.

2.3.1 Procedimiento de toma de demoras

La toma de marcaciones y demoras se puede realizar con el repetidor de giroscópica, del que obtendríamos la demora verdadera, es el procedimiento más rápido y seguro. En cualquiera de los casos también podemos utilizar la aguja magnética para medir el ángulo respecto del norte magnético, lo que incluye la necesidad de corregir las demoras por el desvío magnético. Por último, se puede realizar a través del taxímetro directamente, con menor precisión. Los operadores que realizan esta tarea reciben la denominación de **marcadores de alerones**.

Es imprescindible tener en cuenta que el intento de tomar una demora en el momento en el que el buque se encuentra haciendo una caída no es fiable. A medida que vamos cayendo, indistintamente la banda que sea, la graduación móvil del mortero varía con el cambio de rumbo, y por el continuo cambio no es fiable la toma de demoras en esta situación.

Respecto al magnetismo del buque es importante hacer un apunte que afectará al resto el proyecto. En la construcción de las distintas unidades navales se utilizan gran cantidad de materiales susceptibles de ser imantados bajo la acción del campo magnético terrestre. El magnetismo que adquieren estos materiales, dependiente de los hierros y aceros empleados, se clasifica como permanente o inducido según se adquiera y pierda, lenta o rápidamente, de forma respectiva. Por este motivo, cualquier elemento magnético que se quiera utilizar a bordo, estará sometido a este fenómeno.

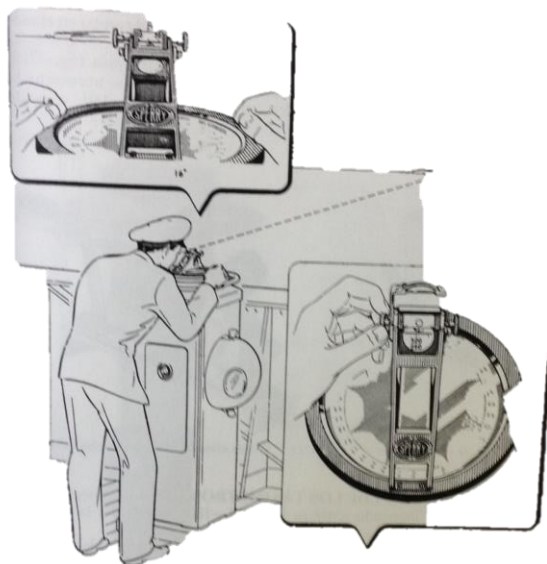


Figura 2-6 Toma de demoras con la alidada (8)

2.3.2 Procedimientos para resolver el problema de situación

Una vez obtenidas las demoras por los marcadores de alerones, la información llega al **sirviente carta**, este operador es el responsable de obtener la situación y realizar la recomendación al puente de

gobierno. Dependiendo del tipo de líneas de posición que reciba deberá actuar de distinto modo. Aquí es donde se materializa la importancia de la cartografía náutica, para obtener la solución se utiliza un método gráfico que consiste en representar en la carta las líneas de posición. Dependiendo de cuáles sean estas líneas de posición y de la proyección que se utilice en la carta varía considerablemente la complejidad de la representación.

Como se ha mencionado anteriormente, a través de la proyección mercatoriana las trayectorias loxodrómicas se representan como una recta, en el caso de las demoras su representación sería una curva ortodrómica, por definición son arcos del círculo máximo entre la posición del buque y la del punto fijo en tierra.

Cuando la distancia entre tierra y el buque es pequeña concluimos que podemos trazar las demoras como rectas. Por la misma razón se considera que las demoras recíprocas se confunden, fundamentando el sistema de trazado.

2.3.2.1 Situación por tres demoras simultáneas

Siempre que sea posible se elegirán puntos que determinen líneas de posición que se corten con ángulos comprendidos entre 60° y 120° , con ángulos superiores a 150° o inferiores a 30° no es admisible el corte, y si alguna vez se utilizan será por causas de fuerza mayor. El problema estriba en que el error en ángulos agudos es mayor como podemos observar en la Figura 2-7 Situación por tres demoras.

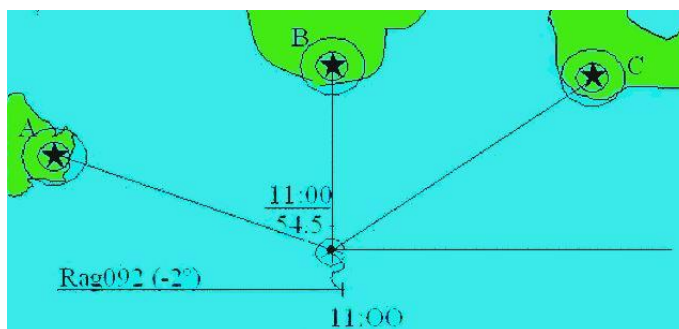


Figura 2-7 Situación por tres demoras (9)

La demora del punto observado determina una línea de posición del buque, que, por su fácil obtención se utilizan con preferencia respecto de otros métodos. Siempre se deben trazar demoras verdaderas.

La intersección de estas líneas delimita la posición del buque en el momento de la observación de las demoras. Cuando el corte de las tres líneas determine un triángulo pequeño se elegirá como situación del buque el centro del mismo.

Para el trazado de las líneas en la carta se emplean distintos métodos, existen trazadores de demoras, transportadores o cartas que ya tienen incorporados círculos graduados para trazarlas. Es necesario remarcar que mientras se hace la observación al tomar varias demoras para trazarlas como simultáneas se debe empezar por la que varíe más despacio o la más difícil de visar con el objetivo de aumentar la precisión.

Para el trazado de una situación por tres demoras simultáneas se sigue con el siguiente procedimiento, desde cada uno de los puntos de tierra se traza la demora inversa correspondiente y si la intersección determina un punto, éste se escogerá este como la situación. Si se obtiene un triángulo se elegirá, como se ha mencionado su centro. En el caso de que el triángulo obtenido sea muy grande se intentará eliminar aquella que ofrezca menos garantías de acuerdo con el rumbo, situación estimada y precisión en su obtención. Repitiendo las marcaciones para comprobar la situación.

2.3.2.2 Traslado de líneas de posición

Las demoras y el resto de las líneas de posición se pueden trasladar en el tiempo conociendo el rumbo y distancia navegada entre el tiempo que ha transcurrido entre cada una de las tomas.

2.4 Posicionamiento por GPS.

Existen otros métodos de posicionamiento a través de diversos métodos electrónicos, como es con el empleo del RADAR, también con la necesidad de referencias externas o por la emisión de equipos instalados en tierra que desarrollan la radionavegación, sin embargo, el sistema de posicionamiento actual por excelencia es el de la navegación por satélite.

Existen varios Sistemas Globales de Navegación por Satélite o GNSS (*Global Navigation Satellite System*) entre ellos el ruso, denominado GLONASS, otros que están aún por desarrollarse como el GALILEO, de origen europeo, pero el más conocido y extendido mundialmente es el GPS.

Actualmente el acceso a los satélites GPS es libre. El sistema está compuesto por una constelación de 24 satélites y su cobertura es mundial. La distribución de los satélites es a través del planeta permite la triangulación de posiciones que proporcionan la posición en la que se encuentran los receptores de esta señal.

Su uso en la navegación está muy extendido y supone actualmente el medio de posicionamiento común de la mayoría de buques estando regulada la obligatoriedad de su empleo a bordo.

Aun así también es necesario conocer el sistema e investigar los posibles fallos del mismo con el objetivo de estar preparados para cuando cualquier acontecimiento adverso ocurra.

2.4.1 Análisis del posicionamiento por GPS

Aparte de las prestaciones que proporciona el GPS es incluso más necesario conocer los fallos del mismo. La disponibilidad de este sistema depende directamente de EEUU, organismos de este estado repartidos por todo el mundo y con su estación de control principal en la Tierra, situada en Colorado Springs (EEUU).

EL GPS dispone de una monofrecuencia de 12 canales distintos, pero diferencia la frecuencia portadora de las aplicaciones civiles y militares, en el caso de este último se utiliza un código de precisión cifrado. Debido al carácter militar de su desarrollo, el Departamento de Defensa de los EEUU se reserva la opción de poder incluir cierto grado de error aleatorio. Esto implica que un sistema de navegación no puede depender única y exclusivamente, sin ninguna alternativa, de este sistema de navegación. A pesar de la comodidad de tener una posición actualizada constantemente y demás prestaciones del sistema, se debe tener en cuenta que no es una información a la que se pueda acudir ciegamente, sin contrastar con ningún otro método.

Durante la fase de investigación del proyecto, gracias al contacto del CEMEDEN se ha podido tener acceso a un estudio de perturbación del sistema GPS de septiembre de 2014. El ejercicio consistió en intentar inhabilitar la señal del GPS por medio de pequeños perturbadores portátiles. El resultado expone como se comienzan a obtener fallos en las señales, tanto civil como militar, así como errores en el AIS y el NAVTEX. A pesar de ello el sistema de combate, mantiene el cálculo de posicionamiento a través del navegador inercial. (10)

Estos motivos confirman la necesidad que se ha expuesto anteriormente de la multiplicidad de sistemas de posición para mantener el control de la situación en el caso de fallo de distintos equipos y no generar autodependencia de ciertos sistemas por muy prácticos que resulten.

3 DESARROLLO DEL TFG

3.1 Alcance del TFG

La primera caracterización del trabajo es conseguir realizar un prototipo que se pueda emplear en el simulador de Navegación. Lejos de lo que se debería suponer como una plataforma móvil sometida a diferentes movimientos y mayor número de interferencias de distintas magnitudes.

En el proyecto se requiere que el sensor inercial transmita la información del movimiento de ángulos relativos, y dicha información se pueda referenciar al norte verdadero. Se requiere acceder a los códigos de los sensores inerciales para poder tomar los datos con precisión y con una respuesta adecuada a su aplicación.

La resolución de la situación debe adecuarse a las necesidades de precisión de la navegación costera, es importante que analíticamente se suprima al máximo el mayor grado de error de cálculo y aproximación para obtener un programa fiable y rápido.

En el caso del aparato que se integre con la alidada es importante que el modo de actuación y dar información al operador que debe tomar las demoras de lo que se encuentra realizando por medio de actuadores acústicos y lumínicos.

Además, el interfaz con el usuario interpreta la información recibida puede tener desarrollada una parte gráfica que permita analizar rápidamente la situación. Dicha herramienta puede tener múltiples funciones auxiliares en función de las necesidades que se presenten en el desarrollo.

La carencia de la información de que se dispone en las cartas respecto de la posición geográfica de las marcas en tierra debe de ser suplida por la información que tienen las cartas electrónicas, trabajando con programas que permitan gestionar esa información.

El conocimiento y preparación de la zona de navegación sigue siendo necesaria con este sistema, el programa no pretende auto reconocer las marcas sino medir automáticamente el ángulo de referencia que determina la línea de posición.

El prototipo que se diseñe responderá a estas necesidades básicas, dando pie a desarrollar un sistema mejorado a partir de las conclusiones que se deriven del estudio y trabajo realizados en este proyecto.

3.2 Fases del Desarrollo

Para el desarrollo se ha elegido y llevado acabo un modelo de gestión perceptiva y un desarrollo en cascada de las fases que vienen a continuación:

- Análisis
- Planificación y supervisión del proyecto
- Herramientas y entorno para el desarrollo
- Diseño
- Sistema de toma de demoras
- Sistema resolución problema de situación
- Interfaz
- Montaje
- Integración
- Toma de decisiones

A continuación se detallan estas fases incluyendo las actividades realizadas durante el desarrollo. No se pretende que estas actividades sean realizadas en el orden estricto en que aparecen en este documento. Algunas se realizarán sin interrupción, otras requerirán realizarse en varias fases del proyecto y algunas de las actividades iniciadas en las primeras etapas podrían requerir para su finalización resultados de actividades que serán realizadas más adelante.

3.3 Análisis

Se adoptará en el área de desarrollo con el fin de facilitar la división de tareas en el proyecto. La responsabilidad de llevar a cabo todas ellas depende de diversos factores como el material o tiempo disponible que son necesarios tener en cuenta, pero todo este conjunto de condicionantes deben quedar bien definidos.

Esta división de principales campos del proyecto permite distribuir las tareas con un mayor orden. Los recursos humanos se reducen al tutor y al alumno, este último es el principal responsable que el proyecto salga adelante. Para ello deberá dar constancia, a través de reuniones con el tutor, del modo en que evoluciona el progreso de trabajo.

3.4 Planificación y supervisión del proyecto

La planificación es crucial en el desarrollo del trabajo, es importante saber gestionar el tiempo y ser capaces de cumplir los hitos establecidos a corto plazo que permiten avanzar día a día en el proyecto. A través de la revisión se observa y registra regularmente las actividades que se llevan a cabo durante el proyecto. Es un proceso rutinario que pretende incluir todos los aspectos del proyecto.

La supervisión también implica informar sobre el progreso al tutor. Estos informes permiten que la información que se ha recogido y conclusiones a las que se ha llegado se utilicen en la toma de decisiones para mejorar el desarrollo del proyecto.

3.4.1 Planificación inicial

Para llevar al día el proyecto y definir los principales hitos y tareas se ha diseñado un Diagrama de Gantt, el diagrama se encuentra desarrollado en el Anexo II de la memoria, marcando los tiempos del trabajo.

Tarea	Fecha de inicio	Fecha de fin
-------	-----------------	--------------

Asignación TFG	12/12/14	12/12/14
Primeros contactos con tutor	12/12/14	18/12/14
Periodo de investigación	16/12/14	08/01/15
Pruebas en Arduino	23/12/14	07/01/15
Documentación	31/12/14	23/01/15
Programación Arduino	05/01/15	30/01/15
Programación Processing	19/01/15	11/02/15
Mejoras códigos	12/02/15	20/02/15
Diseño Sistema	26/01/15	30/01/15
Desarrollo Sistema	02/02/15	17/02/15
Pruebas Sistema	18/02/15	27/02/15
Redacción memoria	26/01/15	02/03/15
Período margen	02/03/15	06/03/15
Entrega memoria	06/03/15	06/03/15
Exposición TFG	24/03/15	24/03/15

Tabla 3-1 División de tareas del proyecto

3.4.2 Seguimiento

Esta planificación se actualizará semanalmente con objeto de conocer el estado de avance del proyecto y poder corregir las desviaciones que se observen con la mayor anticipación posible.

3.4.3 Caminos críticos

Cada semana se analizará el camino crítico de las actividades para garantizar una temprana detección de desviaciones globales del proyecto. En cada caso el camino crítico lo determinará el hito o conjunto de hitos que sea necesario obtener en el periodo establecido.

3.5 Herramientas y entornos de trabajo

En el siguiente apartado se indican las herramientas que se han seleccionado para facilitar el desarrollo del SW y HW del sistema. A través de la siguiente tabla se presenta dicha información:

Herramienta	Logotipo	Actividades
Entorno de Desarrollo de Arduino		Fase de aprendizaje de programación Desarrollo código para IMU y conjunto

		de elementos instalados en Arduino
Processing		Desarrollo código Programa Principal
Open CPN		Obtención y procesamiento de información de las marcas de referencia
Virtual Serial Port		Enlaces virtuales de puertos COM para comunicación entre programas
Fritzing		Obtención de prototipo virtual de Arduino y diseño de conexiones de circuito
PuTTY (Port unique Terminal TYpe)		Visualización en un hiperterminal de la información que se envía a través de los puertos de comunicación
Gantt Project		Planificación Proyecto y definición de tareas
Mathcad		Verificaciones de cálculo para desarrollar en el programa principal

Tabla 3-2 Herramientas y entornos de desarrollo

Son un ejemplo de la utilidad que ofrecen actualmente la multitud programas que se pueden encontrar de uso libre para el desarrollo de estudios y trabajos. Facilitan la investigación, la resolución de problemas y la exposición de conclusiones de acuerdo con los resultados obtenidos.

3.6 Principales elementos del HW

3.6.1 Placa Arduino

Arduino es la plataforma de electrónica libre, basada en software y hardware flexibles y fáciles de usar, que ofrece multitud de aplicaciones. Se basa simplemente en una placa con micro controlador y una serie de pines de entrada/salida, junto con un entorno de desarrollo sobre los que se montan numerosos circuitos en función de las necesidades de cada proyecto.

El éxito de la placa se debe tanto a su sencillez y a su carácter open-hardware y su precio. Esto permite la posibilidad de crear nuestra propia versión de Arduino, con pequeños cambios se consiguen placas denominadas “no oficiales”, creadas por personas ajenas a la empresa oficial y que son igualmente válidas. Es posible realizar un proyecto sin disponer de ningún tipo de licencia.

En el proyecto se emplea la placa más básica, denominada Arduino UNO, pero el empleo de cualquier otra placa con otro tipo de prestaciones es un apartado reservado para las posibles mejoras del proyecto, ya que el primer objetivo es estudiar la viabilidad del propio proyecto.

La placa puede ser alimentada a través de la conexión USB o, a través de una fuente externa de alimentación, de entre 6 y 20 voltios. Dispone de una memoria de 32 KB, para cargar los programas.

En la primera conexión con el PC si al conectar la placa no aparecen los drivers automáticamente, se pueden seleccionar manualmente dentro del directorio en el entorno de programación de Arduino.

Por su sencillez y bajo coste permiten el desarrollo de múltiples diseños, el entorno de desarrollo integrado libre se puede descargar gratuitamente. (11)

3.6.2 Sensor inercial

Un sensor inercial es un dispositivo capaz de detectar variaciones en aceleración y velocidad angular. Su uso está extendido a aplicaciones de captura y análisis de movimiento. El dispositivo electrónico recibe el nombre de IMU, siglas que provienen del inglés *Inertial Measurement Unit*. Una IMU está compuesta por tres dispositivos diferentes acelerómetros, giróscopos y magnetómetros. Con todos ellos es posible el estudio completo de los distintos planos y el espacio, dependiendo de los ejes que poseen los sensores. Estos dispositivos se utilizan para estudiar, o controlar el movimiento en general de vehículos, como sensores de orientación, teléfonos móviles, análisis de movimiento de sistemas y también son aplicados en el campo de estudio del movimiento antropológico con fines incluso médicos.

- Acelerómetro.

En un solo chip de silicio podemos encontrar estos sensores, su uso está destinado a medir aceleraciones. Con los acelerómetros podemos medir tanto la aceleración estática (la gravedad) como la dinámica (vibraciones) en distintos ejes coordenados. Se pueden encontrar compuestos de pequeñas barras móviles suspendidas sobre resortes con cierta resistencia contra la aceleración, de esta forma cualquier flexión de la barra se podrá utilizar como una medida de tal magnitud. En nuestros días podemos encontrar otras variaciones como acelerómetros piezoeléctricos, capacitivos o de tipo MEMS.

Su principio de operación está basado en el traspaso térmico por medio convencional, la denominada tecnología MEMS (Sistemas micro-electromecánico) como se conocen nanotecnología mecánica. El inclinómetro es un acelerómetro que calcula la dirección del campo gravitatorio, cuando el blanco sobre el que se sitúa se encuentra en movimiento el vector queda enmascarado por el resto de aceleraciones, y el resultado de la medición tiene un error mayor. (12)

Al estar sujeto a la fuerza gravitacional de la Tierra y no dentro de un marco de referencia aislado, siempre se contará con un componente de aceleración fija en el eje -Z. Es importante que esta aceleración se tenga en cuenta para compensarse adecuadamente cuando se quiere trabajar con resultados absolutos y especialmente cuando el objeto de estudio está en rotación. Es usual que los resultados estén expresados en medidas de g ($1g = 9,8 \text{ m/s}^2$).

Atendiendo a la teoría podemos decir que los acelerómetros pueden ser utilizados para hacer una medición o estimación del movimiento, ya que la posición, por definición, puede obtenerse de la doble integración de la aceleración respecto del tiempo:

$$X = \int \int_{t_1}^{t_2} \left[\frac{(a^2)X}{dt} \right] dt dt \quad [3-1]$$

Esta ecuación sin embargo solamente resultará de utilidad para periodos cortos de tiempo puesto que el error de integración se acumula con el cuadrado del tiempo, por lo que momentáneamente, cualquier desviación o ruido durante la medición resultará en grandes errores en función del tiempo que pase.

La utilidad que se le puede dar a este tipo de sensor en el proyecto es a través de la aceleración de la gravedad, por la que se permite determinar la orientación de un cuerpo. Los acelerómetros responden a diferentes movimientos que generan aceleraciones centrífugas, de Coriolis o de la gravedad entre otras. Y es fácil obtener el ángulo cuando el sistema está parado, pues se puede proyectar a la gravedad en cada uno de los ejes para conocer la inclinación de cada uno y obtener la orientación del sistema. (13)

- Giróscopo.

Por otro lado, los giróscopos, se utilizan para medir la orientación de un objeto, basándose en los principios de movimiento angular. En el inicio de su creación, los giróscopos consistían en una rueda giratoria montada sobre un soporte de tal manera que su eje gira libremente en cualquier dirección. También este sistema ha sido reemplazado por dispositivos MEMS, compuestos de masas diminutas que se utilizan las para medir la magnitud de los giros. (12)

Son numerosas las aplicaciones de los giróscopos estabilización de plataformas, navegación inercial, sistemas de guiado armamentístico.

Las medidas de los giróscopos normalmente están establecidas en unidades de grado por segundo. Las medidas se toman respecto de los denominados ángulos de Euler, estos ángulos permiten definir la orientación de un cuerpo rígido a través de un conjunto de tres coordenadas angulares. La notación que reciben estos ángulos es la siguiente: guiñada o *yaw*, cabeceo o *pitch* y balance o *roll*. Son de uso común en la navegación, y especialmente en aeroplanos. Los ángulos definidos por el matemático Euler, representan la orientación en el espacio de un cuerpo rígido. Con los movimientos en estos tres ejes se puede representar cada movimiento en el espacio en tres dimensiones. (13)

En la figura que se muestra a continuación podemos observar dichos ángulos:

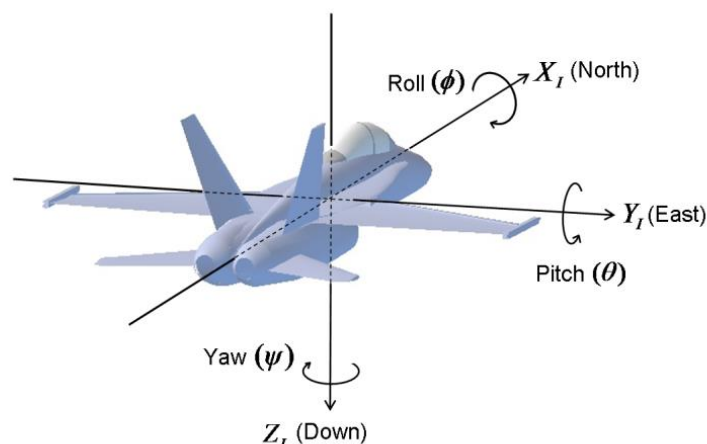


Figura 3-1 El Sistema Inercial (14)

- Magnetómetro.

Los magnetómetros son dispositivos de cuantifican la dirección de la señal magnética. Permiten medir la dirección de los campos magnéticos, recibiendo el apellido de magnetómetros vectoriales, miden la componente del campo magnético en una dirección en particular.

Están sujetos a la variación de temperatura e inestabilidad dimensional de sus núcleos. Para comprender mejor el funcionamiento del dispositivo son de utilidad unas nociones básicas de cómo actúa el campo magnético terrestre.

La Tierra actúa como un dipolo, sin embargo, éstos están desplazados de sus ejes de rotación. La declinación magnética hace referencia a la separación entre el norte magnético y el

norte geográfico. Consideramos como norte magnético la región de la Tierra donde las líneas de campo son perpendiculares a la superficie terrestre.

Gracias a las medidas en los tres ejes del campo magnético podemos calcular la orientación relativa de un objeto. Los compases magnéticos hacen uso de los inclinómetros como referencia para orientar el objeto.

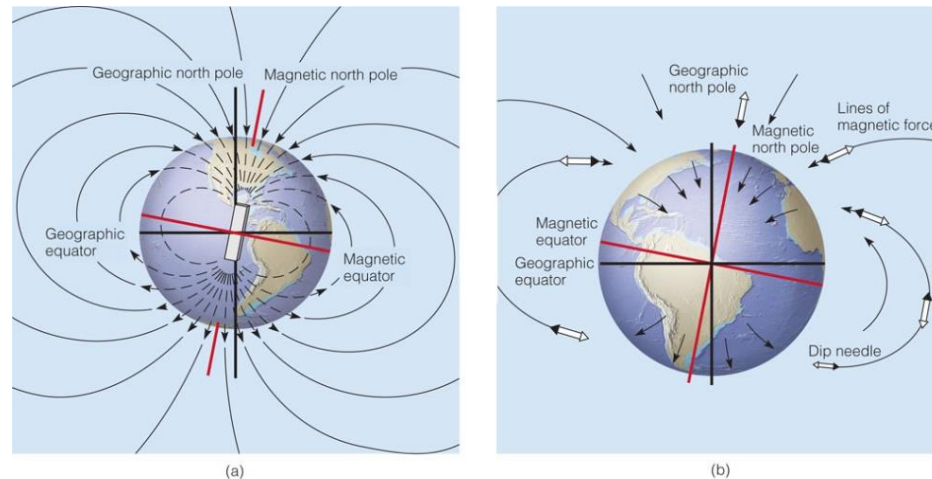


Figura 3-2 Campo Magnético Terrestre (15)

Los acelerómetros y giróscopos suelen combinarse dentro de las Unidades de Medición Inercial (IMU), son útiles para medir la orientación, la velocidad y las fuerzas gravitacionales que experimenta un objeto. Se suelen instalar en satélites, aviones, o transbordadores espaciales. Por medidas de precisión, se suele complementar a los IMU con magnetómetros que permiten contrarrestar los problemas intrínsecos de los IMU. Problemas relacionados con errores de integración acumulados, como los mencionados anteriormente, o falta de precisión para determinar la posición absoluta de un objeto.

La integración de los magnetómetros en los IMU conforman los denominados Sistemas de Referencia de Altitud y Dirección o AHRS (Altitude-Heading Reference System). El sistema realiza un proceso de ‘fusión’ de datos utilizando un tipo de filtro para obtener los tres ángulos de giro respecto al sistema inercial.

Una de las desventajas de usar los sensores inerciales es el error acumulativo con el que se ven afectados. El sistema se encuentra continuamente acumulando errores respecto a posiciones previamente calculadas, es por este motivo que cualquier pequeño error en cada una de estas medidas se va acumulando incrementalmente. Esto ofrece una deriva en las mediciones entre la posición real del sistema y aquella en la que éste cree que se encuentra.

El uso de los sensores inerciales y el desarrollo de la microelectrónica, hacen que estos dispositivos tomen fuerza y se multiplique su uso para distintos fines, ya que el principal parámetro que se había mantenido como desventaja era el precio de estos sistemas. Pero una vez superada esta barrera sus aplicaciones aumentan de forma exponencial.

Con el avance tecnológico de los últimos años, sobre todo en el campo de los teléfonos móviles, así como consolas o videojuegos que integran sensores inerciales de bajo coste y de suficiente precisión, surge la idea de utilizar tales dispositivos en mediciones de ángulos con el movimiento de los mismos sobre el norte verdadero, tal y como se plantea en el presente proyecto.

3.7 Diseño

3.7.1 Diseño preliminar

Para crear el diseño, en primer lugar se esbozan divisiones de bloques atendiendo a como se quiere enfocar el problema, y una vez analizados la función de cada uno de ellos, sus necesidades y posibilidades que ofrecen se obtiene el diseño del sistema. Algunos de estos diseños preliminares se incluyen en la Figura 3-3, en la que se puede observar en el esquema (1) como se pensó en incluir el programa de obtención de la situación geográfica en el propio Arduino. Uno de los motivos por los que se descarta el empleo de Arduino para realizar la situación es la memoria limitada que deben tener los programas que se deseen cargar.

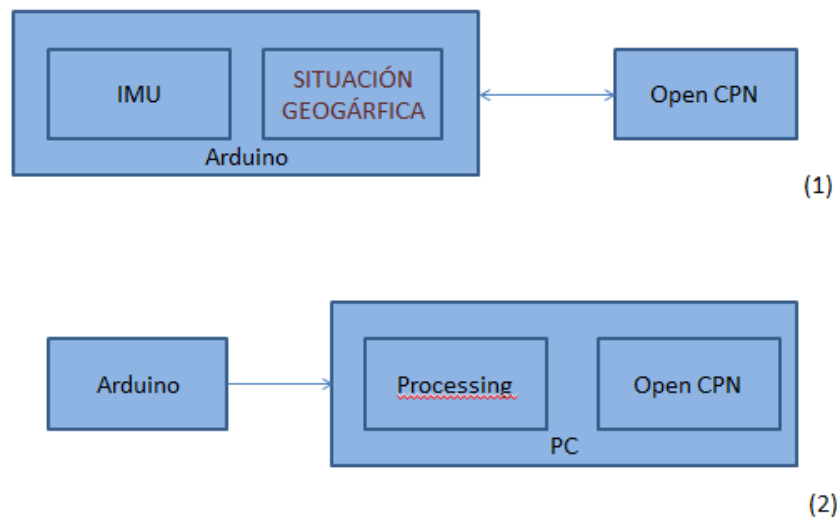


Figura 3-3 Diseños preliminares

- (1) Obtención situación en Arduino
(2) Obtención situación en Processing

3.7.2 Diseño sistema

Durante el diseño preliminar se realiza el análisis completo de la documentación de referencia y tiene como resultado la especificación para diseño en el que se incluyen los elementos y sistemas de mayor nivel que compondrán el sistema de toma de demoras y, asimismo, se establecerá la arquitectura general. De esta forma el diseño quedará dividido en elementos más pequeños que podrán ser desarrollados por separado para finalmente ser integrados y probados. En dicho documento se reflejará el diseño de alto nivel del sistema completo, identificando los elementos de configuración hardware (HWCI), elementos de configuración software (SWCI) e interfaces existentes tanto internos de los diferentes elementos como externos con otros elementos o sistemas.

El diseño detallado comienza una vez finalizado el diseño preliminar de alto nivel. Durante esta etapa se completa el diseño de todos los elementos que fueron identificados durante el diseño preliminar. En esta etapa pueden surgir pequeños cambios o identificarse nuevos elementos que deben ser también diseñados para cumplir con los requisitos del sistema.

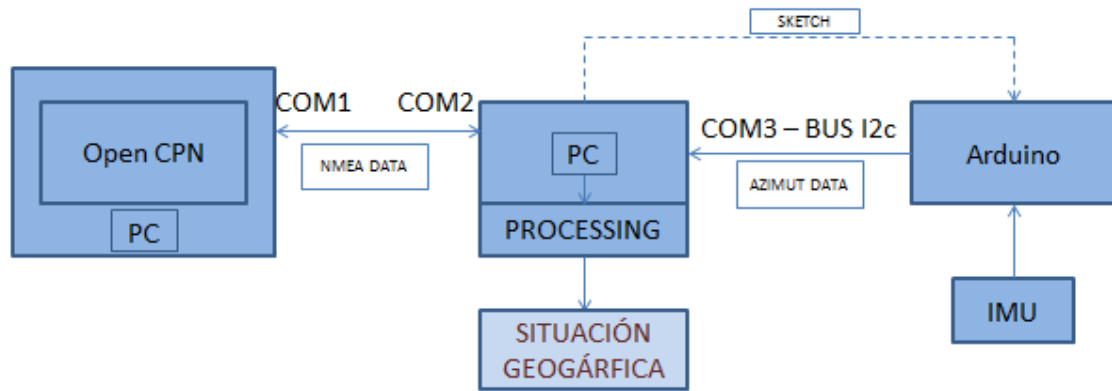


Figura 3-4 Diseño del Proyecto

La Figura 3-4 muestra el esquema de bloques en que está compuesto el conjunto del sistema, los trabajos en cada una de las áreas se pueden en mucha medida trabajar independientemente, agilizando el proceso de trabajo, sin embargo es necesario que en la integración de todos los componentes no haya ningún problema.

El proyecto se divide en distintos bloques de trabajo:

- El primer bloque es el de Arduino, este bloque incluye el diseño mecánico de integración del sensor inercial (IMU) en la alidada. También incluye el desarrollo del código de Arduino que permita enviar esta información a un ordenador para poder ser procesada y tomar acciones en
- El siguiente bloque está encuadrado en el programa OpenCPN, del programa se obtienen la posición de las marcas en tierra necesarias para la resolución del problema geográfico. El programa, aunque esté incluido en el ordenador se considera como un bloque aparte y se comunica a través de puertos COM con el programa principal.
- En el bloque del PC se engloba el programa diseñado en Processing. El programa supone la mayor parte del trabajo del proyecto, las posibles mejoras, nuevas ideas y complementos al código de este proyecto son numerosas y de diferente utilidad.
- Todos los bloques siguen un objetivo común la obtención de la situación geográfica del buque, y no deben apartarse de este objetivo o suponer un impedimento para el mismo.

3.7.3 Diseño Arduino

El diseño preliminar en Arduino consiste en el montaje del circuito en un *protoboard*. El montaje permite diseñar y cargar el código con la posibilidad de hacer pequeños cambios de manera sencilla según sean necesarios. En la Figura 3-1, se puede observar el circuito de toma de demoras con el conjunto de los distintos elementos que se han ido añadiendo al conjunto, cada uno con su respectiva aplicación.

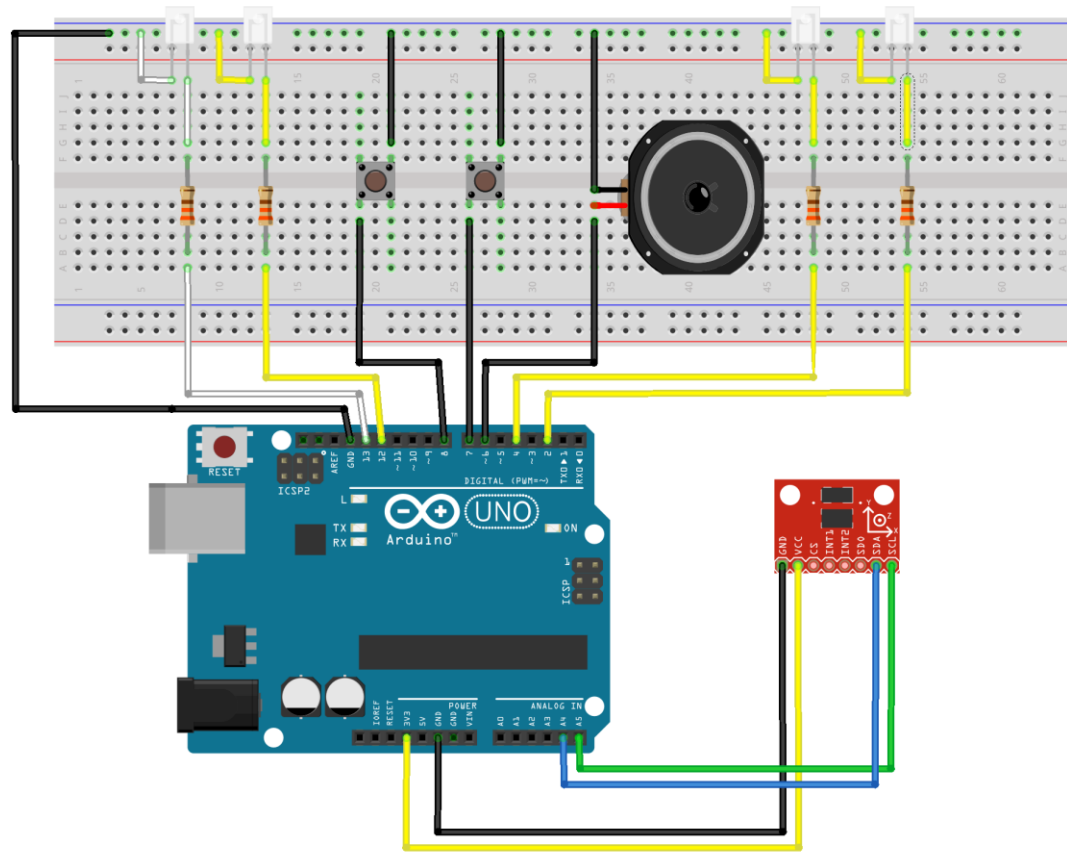


Figura 3-5 Diseño Arduino

El diseño está compuesto por:

- Una placa de Arduino UNO.
- 4 LEDs, uno blanco y tres amarillos que se identifican cada uno con una acción. El blanco se encenderá cuando se inicialice a 0 el sensor inercial referenciando el norte con la información del taxímetro. El uso de los LEDs es una referencia visual que indica cual es el número de demora que están registradas en el programa principal.
- Como complemento a la referencia visual también se ha decidido incorporar un zumbador piezoeléctrico al conjunto, la función de dicho elemento es realizar un tono al pulsar cada botón. Supone una referencia acústica para el operador.
- El principal componente es el sensor inercial MPU 9250, la elección de dicho sensor está en función de sus características, recogidas en el Manual de especificaciones de productos del autor. Es un sensor compuesto por tres dispositivos, el giróscopo, acelerómetro y magnetómetro. El análisis de su empleo y aplicaciones es un estudio importante y crucial para el desarrollo del proyecto que se incluye más adelante. (16) (17)

3.7.4 Diseño mecánico

El diseño mecánico ha ido evolucionando a lo largo del proyecto y es uno de los principales puntos de conclusión. En función de cómo se enfoque la obtención de datos del sensor y el movimiento que se precise realizar surgen diversas posibilidades de diseños mecánicos. Su puesta en práctica está condicionada por el tiempo que se dispone de trabajo, por lo que es necesario optar por

una de ellas y estudiar las posibles mejoras o distintos enfoques de otro tipo de esquemas. En el Anexo III, se incluye unos planos de diseño con las principales partes desarrolladas.

3.7.5 Diseño electrónico

El circuito eléctrico consta de dos bloques diferenciados, el primero es el esquema electrónico de Arduino. En la Figura 3-3 se puede observar el circuito electrónico de Arduino, en el que se incluyen los elementos de la *protoboard* de la fase del diseño preliminar.

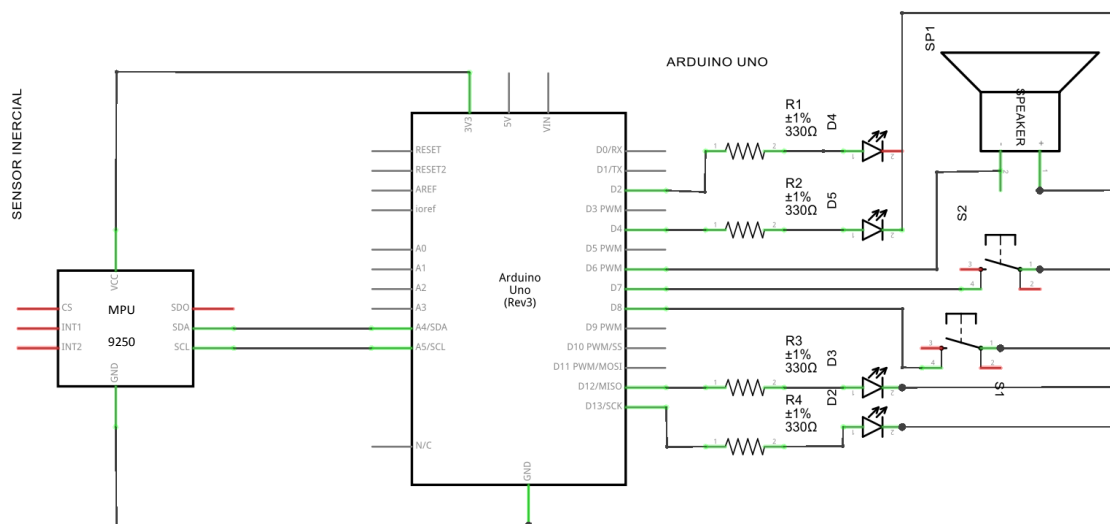


Figura 3-6 Diseño electrónico Arduino

El circuito está compuesto por tres bloques principales, en el centro del mismo se encuentra la placa de Arduino, sobre la que se establecen las conexiones de todo el resto de elementos. Los otros dos bloques componen esta serie de elementos, en la parte izquierda el sensor inercial, y en la derecha el circuito de interacción con el operador que incluye los botones, el zumbador piezoeléctrico y los LEDs.

3.7.6 Comunicación

Esta funcionalidad permite la comunicación entre los distintos programas. El traslado de información se realiza por medio de puertos serie COM. El esquema de puertos se recoge en el diseño del proyecto en la Figura 3-4.

En el caso de Arduino UNO los programas se cargan los programas por medio de un cable USB. En el caso del programa cargado se utiliza este cable también para enviar la información del sensor desde Arduino al PC. El modo de enviar esta información es a través del bus I2c, el protocolo es un estándar que facilita la comunicación entre dispositivos.

El sensor inercial trabaja con las señales de las siguientes conexiones:

- SCL (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema.
- SDA (System Data) es la línea por la que se mueven los datos entre los dispositivos.

El protocolo permite escribir la información necesaria del sensor inercial y enviarla por el puerto de comunicación. El Arduino está conectado al puerto COM3, y por medio de Processing podemos leer dicha información separar cada cadena de caracteres que devuelve Arduino o componente en más pequeños fragmentos que dividen la información recibida para trabajar con ella de forma independiente.

En el caso de OpenCPN se crea un enlace de puertos virtual con el programa *Virtual Serial Port*, OpenCPN es un programa libre de cartografía electrónica que está pensado para establecer conexiones con equipos para recibir la señal GPS y poder situar la posición del buque. En el caso del proyecto se aprovechan dichas conexiones del programa al exterior, pero con la creación de puertos virtuales se dirige dicha información a un puerto al que se puede acceder con Processing,

Al igual que la cadena de Arduino, la información recibida de OpenCPN es recibida y procesada a través del protocolo de la Asociación Nacional de Electrónica Marina o NMEA (*National Marine Electronics Association*). Esta organización es la precursora de la comunicación entre equipos marítimos que, tradicionalmente se han utilizado sin poder integrar toda su información en un mismo sistema.

El protocolo NMEA es un medio a través del cual los instrumentos marítimos como pueden ser los receptores de GPS pueden comunicarse entre ellos. Con este protocolo se suelen cifrar posiciones GPS con pequeñas sentencias que fácilmente se pueden descomponer y seleccionar la información deseada para trabajar con ella. (18)

3.8 Sistema de toma de demoras

El sistema creado en definitiva tiene el objetivo principal de tomar demoras, estas demoras se identifican con los ángulos azimutales, con este motivo se generan los siguientes métodos:

3.8.1 Método de obtención del azimuth

El método de obtención de los ángulos de Euler es por medio del código que se carga en Arduino, el MPU9250 tiene 3 sensores distintos que se han explicado anteriormente. El código modificado desde RTIMU permite la selección en la función de inicialización (*setup*) de los sensores que se deseen emplear. Esta selección se debe hacer como consecuencia de analizar las necesidades del sistema y prestaciones de cada uno de los sensores.

En el código se incluye una función de fusión que según la selección de sensores habilitados ofrece distintos resultados. Es importante hacer un estudio de cada sensor, su aplicación y verdadera utilidad en el proyecto.

Cuando se empiezan a realizar pruebas con el proyecto se observa que el código que se está empleando no ofrece los resultados esperados, el sensor responde al movimiento pero este tiene que ser algo brusco y tiene la tendencia de volver al mismo punto de partida al estabilizarse del movimiento. Al encontrarse estos primeros problemas que pueden afectar al trabajo es importante analizar el origen y ofrecer una posible solución.

Los problemas que afectan o podrían afectar se analizan a continuación:

- Aplicación del acelerómetro

Sobre un objeto estático el acelerómetro de tres dimensiones 3D proporciona el valor de la gravedad en los tres ejes. El vector de la aceleración de la gravedad siempre apunta al eje Z, el centro de la Tierra, de la proyección de este vector se pueden obtener la inclinación del objeto sobre cada eje

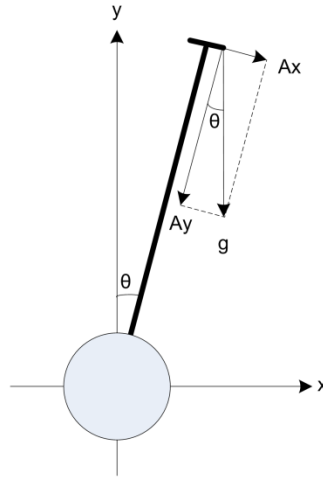


Figura 3-7 Inclinación de un cuerpo

Atendiendo a la Figura 3-7 la inclinación del sistema puede calcularse según la siguiente fórmula:

$$\theta = \arctan\left(\frac{a_x}{a_y}\right) = \arcsin\left(\frac{a_x}{\sqrt{a_x^2 + a_y^2}}\right) = \arcsin\left(\frac{a_x}{g}\right) \quad [3-2]$$

Siendo a_x , a_y , a_z las lecturas del acelerómetro en los ejes X, Y y Z respectivamente.

Las medidas que devuelve el acelerómetro se ha indicado que están referidas a $1g$ ($9,8 \text{ m/s}^2$), obteniéndose que:

$$\theta = \arcsin(a_x) \quad [3-3]$$

Y en el caso de que estos ángulos sean pequeños se puede aproximar el arco por su seno, por lo que:

$$\theta \approx \sin(\theta) = a_x \quad [3-4]$$

En el caso de que el objeto se encuentre en movimiento este puede estar sometido a distintas aceleraciones, los acelerómetros medirán la aceleración neta compuesta por la aceleración de la gravedad y la propia aceleración del objeto. Por tanto, en este caso no puede obtenerse directamente la aceleración de cada eje a partir de la medida de la aceleración

Los acelerómetros además son sensibles a las vibraciones, que más que un problema de las mismas, es una consecuencia de las fuerzas creadas por las vibraciones del objeto, que se generan con el movimiento del propio sensor.

Los ángulos de Euler se pueden obtener en los acelerómetros y por tanto pueden ayudar a la ecuación de fusión de los sensores de acuerdo con las siguientes ecuaciones para calcular la variación de los ángulos relativos *roll*, *pitch* y *yaw*.

El ángulo de *roll* es:

$$\phi = \arctan\left(\frac{a_y}{a_z}\right) \quad [3-5]$$

El ángulo de *pitch* es:

$$\theta = \text{atan}\left(\frac{-ax}{ay \cdot \sin(\emptyset) + az \cdot \cos(\emptyset)}\right) \quad [3-6]$$

En el caso de la guiñada (*yaw*), teniendo en cuenta que el acelerómetro se instala sobre una superficie plana fijando el plano XY, un movimiento de ‘guiñada puro’ no tendrá como resultado ninguna medida en el sensor en el eje Z porque es paralelo a la gravedad y fijo. En los ejes X e Y sí que sería posible medir aceleraciones pero únicamente debidas a vibraciones y pequeños movimientos a corto plazo que pueden confundirse con el propio ‘ruido de fondo’ del acelerómetro. Es posible realizar cálculos del ángulo *yaw* por otra serie de métodos pero por cuestiones operativas no se profundizarán sobre ellas en el presente trabajo.

- Aplicación del magnetómetro

El principal problema de los datos que se obtienen de sensores magnéticos es la escasa precisión que tienen por el momento. La variación de los datos obtenidos simplemente activando el magnetómetro varían manteniendo el sensor en posición estática algún grado, incluso llegando a cambiar completamente el ángulo sin atender a ningún movimiento del propio sensor, con el código de fusión de autor de la librerías.

La aplicación al estar orientada a su uso en un buque, en este caso en los buques tiene mayores repercusiones al estar sometidos a dos tipos de magnetismos, el **permanente**, que se adquiere en su tiempo de construcción al someterse a un campo manteniéndose inmóvil, y el **inducido**, dependiente del campo magnético terrestre del momento. Las correcciones para su uso a bordo de un barco difieren del enfoque que se le pueda dar a la hora de generar un prototipo y realizar prácticas experimentales en un simulador de navegación, donde no encontramos los mismos problemas que en un buque. (19)

Esto se entiende como una primera aproximación que valida el objetivo, la creación del sistema de posicionamiento, pero que también es necesario procesar en la mayor medida posible estos datos, y poder juzgar según sean obtenidos los resultados.

Situando el magnetómetro 3D sobre el plano XY, la siguiente operación nos daría el la orientación del sensor (*yaw*) respecto al norte magnético:

$$\theta = \text{atan}\left(\frac{m_y}{m_x}\right) \quad [3-7]$$

Donde m_x , m_y , m_z son las lecturas del magnetómetro en los ejes X, Y y Z respectivamente.

Si la plataforma no está estabilizada, como es el caso de un buque, los componentes vectoriales del campo magnético en cada eje estarán “alterados” puesto que no están referidos al sistema de coordenadas terrestre. Necesitamos conocer el campo magnético respecto al sistema de coordenadas terrestre pero sin embargo lo que tenemos es el campo magnético respecto al objeto, que estará rotado respecto a la tierra, debido, en nuestro caso, a los movimientos propios del buque.

Podemos deshacer la rotación girando el vector campo magnético respecto al sistema de coordenadas terrestre. La forma más sencilla de hacerlo es encontrar los ángulos de roll y pitch utilizando el acelerómetro. Una vez calculados, construimos una matriz de rotación que rote el vector campo magnético en el sistema de coordenadas terrestre.

El ángulo de *yaw* es:

$$\Psi = \text{atan} \left(\frac{mz \cdot \sin(\phi) - my \cdot \cos(\phi)}{mx \cdot \cos(\theta) + my \cdot \sin(\theta) \cdot \sin(\phi) + mz \cdot \sin(\theta) \cdot \cos(\phi)} \right) \quad [3-8]$$

El ángulo de guiñada calculado está referido al norte magnético que no coincide con el norte verdadero, la diferencia es la llamada declinación magnética. Para obtener el ángulo de guiñada verdadero debemos **restar la declinación** que depende de la ubicación geográfica del punto en el que esté ubicado el magnetómetro. Por este motivo de desajuste respecto a la lectura del sensor y la de la giroscópica de abordó se ha diseñado una función en Processing de reasignación de valores para transformar el norte del sensor en el norte verdadero.

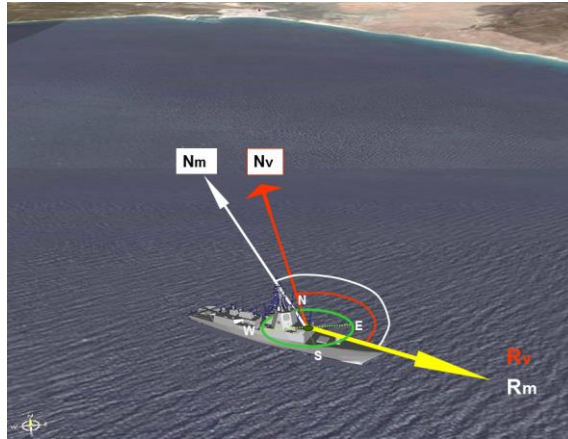


Figura 3-8 Declinación Magnética (20)

Al ser su fin la instalación en los buques, y conociendo que las brújulas magnéticas sufren desviaciones debidas a los campos magnéticos de sus alrededores así como a su cercanía a dispositivos electrónicos. En un principio se debe descartar el uso del sensor en su empleo a bordo de los buques, aun así, por temas de investigación podremos utilizar la información del magnetómetro para obtener datos experimentales en las pruebas realizadas en el Simulador de Navegación de la Escuela Naval Militar.

- Aplicación del giróscopo

A través del giróscopo se obtiene la velocidad angular de giro respecto a los tres ejes del espacio. El ángulo de giro puede determinarse por integración de esta velocidad angular.

$$\theta(t) = \int_{t_1}^{t_2} G(t) dt \quad [3-9]$$

Donde $G(t)$ es la lectura del giróscopo respecto a la dirección de rotación. Cuando el intervalo de tiempo es pequeño, la lectura del giróscopo puede considerarse constante y la ecuación anterior se puede escribir como:

$$\theta(t) \approx \theta(t_1) + G(t) \cdot (t_2 - t_1) = \theta(t_1) + G(t) \cdot \Delta t \quad [3-10]$$

Las medidas de un giróscopo son inmunes a movimientos que no sean angulares y por tanto menos susceptibles a vibraciones, sin embargo, puesto que la medida angular es acumulativa, un error de medida en el giróscopo se traducirá en una estima errónea del ángulo. Los giróscopos tienen un error de deriva debido al cual, cuando están estáticos, dan una velocidad angular distinta de cero. Esta

velocidad angular, cuando se integra respecto al tiempo para conseguir el ángulo, provoca un error angular que **aumenta linealmente con el tiempo**. Sin embargo, esta deriva puede substraerse de la medida puesto que es relativamente constante para cada temperatura cabe la posibilidad de poder generar un ecuación que sustraiga dicho error teniendo en cuenta la temperatura.

Además de lo anterior, el giróscopo tiene otro inconveniente, mide la velocidad angular, que no es una medida absoluta, por lo que el ángulo obtenido es también relativo, y dicha medida es necesario poder referenciarla al norte verdadero.

- Señal de fusión de sensores

Las lecturas que proporcionan cada uno de estos sensores anteriores se pueden combinar para obtener una estima más fiable del ángulo de guiñada (*yaw*).

En este trabajo se utilizará únicamente el acelerómetro para combinarlo a las lecturas del magnetómetro, compensando los movimientos de *roll* y *pitch* de la plataforma.

En el cálculo del ángulo de guiñada utilizaremos la medida del giróscopo y el magnetómetro. La medida angular relativa del giróscopo se obtiene de la siguiente expresión:

$$\theta(t) \approx \theta(t_1) + G(t) \cdot \Delta t \quad [3-11]$$

3.8.1 Calibración datos

Si se va a utilizar el magnetómetro se debe hacer la calibración magnética del mismo, esto se realiza de manera sencilla a través de uno de los códigos de RTIMU, con el nombre 'ArduinoMagCal.ino'. (21)

Durante la realización de las distintas pruebas realizadas, se concluye que se puede estudiar la posibilidad de aplicar un filtro a la señal obtenida por el sensor inercial para adecuarla más a las necesidades del sistema y recibir datos más precisos con la información recibida. Estos sensores tienen diversas aplicaciones, pero para su instalación en las alidadas se deberá optimizar la precisión de los movimientos que se realizan sobre ellas que normalmente son lentos y necesitan de gran precisión y estabilidad. Siempre habrá un compromiso entre la rapidez y la precisión. (19) (22)

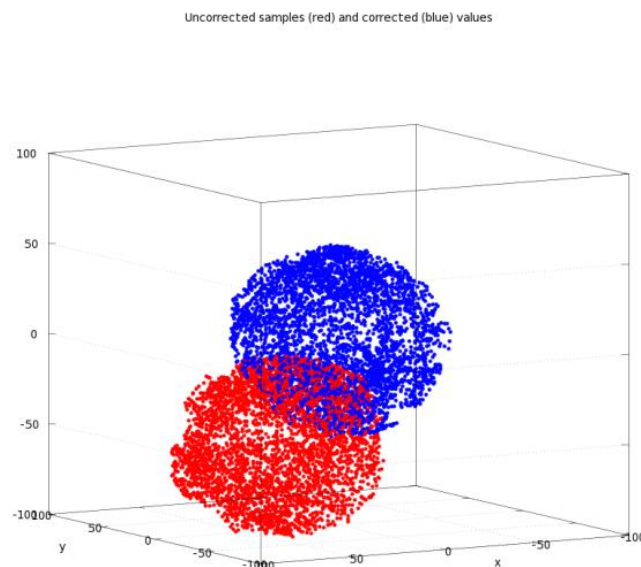


Figura 3-9 Calibración de datos (22)

El conjunto de marcas azules son los datos correctos centrados en el origen, tras la calibración si se hace uso del magnetómetro el sistema funcionará con una mejor respuesta.

3.8.2 Código Arduino

El entorno de desarrollo de Arduino se conecta con la placa de Arduino a la que actualiza los programas y se comunica con ellos. Los programas que se generan reciben el nombre de bocetos o *sketches*. Existe una interfaz con el programa para visualizar la información que pasa a través del puerto serie. Con órdenes sencillas se pueden ‘imprimir’ todo tipo de variables para transmitir las entre el PC y Arduino. Dispone de distintas aplicaciones, entre otras la de verificación y compilación del código, y la posibilidad de actualizar programas. (23)

Trabajando con Arduino es importante seleccionar el puerto serie al que está conectado la placa, y también modificar las preferencias dependiendo del tipo de placa con el que estemos trabajando y ajustar correctamente la velocidad de comunicación.

Gracias al carácter libre con el que surge la idea de Arduino, existen repartidos por todo el mundo cibernético numerosos ejemplos de códigos de programación y de montajes o proyectos al alcance de todo el mundo. Esta posibilidad hace que se pueda mejorar casi continuamente todo tipo de programas, y también hace que sea más fácil el entorno de aprendizaje en el que multitud de usuarios suben sus proyectos o creaciones a la web y a través de vídeos tutorizados enseñan a materializar sus propias creaciones.

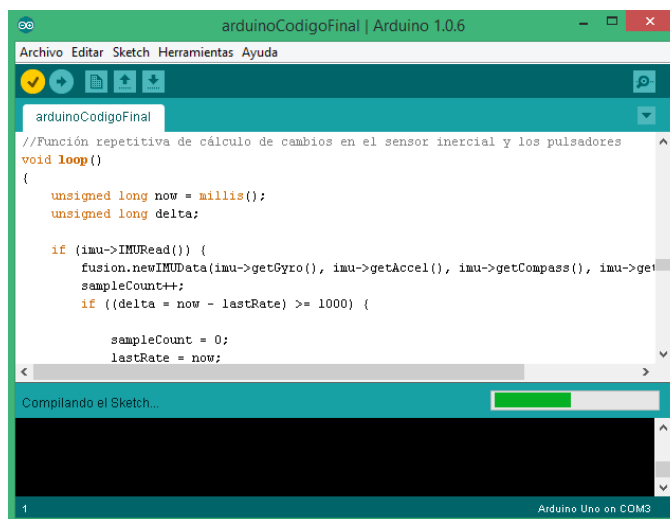


Figura 3-10 Entorno Arduino

El programa definitivo de Arduino recibe el nombre de ‘arduinoCodigoFinal’, está recogido en el Anexo IV. En la primera parte del código están definidos las diferentes variables y elementos lógicos, se definen los distintos pines que se utilizarán en el proyecto y su naturaleza.

Las funciones incluidas en el código tienen un objetivo específico. La primera de ellas de inicialización o *setup* en la que se definen constantes, se fijan y definen distintos parámetros o selección de los sensores que se desee utilizar. La segunda función es la función bucle o *loop*, de acuerdo con su nombre es un bucle que el programa repite constantemente para realizar el conjunto de acciones que tienen lugar en determinados eventos como el de pulsar los botones o que necesitan constante actualización como la información del sensor inercial.

El código es el resultado de una modificación del código registrado por RTIMU, el código inicialmente crea una función de fusión de los tres sensores integrados en el MPU 9250, el código incluye la lógica para enviar la información tanto del IMU como de los botones por el bus I2c para que pueda ser procesado. (21)

3.8.3 Importación de librerías

El uso del código abierto necesita de la descarga e importación en el entorno de distintas librerías sobre las que trabaja el código principal. Después de analizar el código y estudiar su respuesta, como se indicará en el apartado de análisis y conclusiones, es necesario modificar el mismo y generar uno que se adecúe a las necesidades del proyecto. Aun así inicialmente es el que se utiliza para las principales pruebas con la idea de modificar el código.

3.9 Sistema resolución problema de situación

3.9.1 Código Processing

Processing es un entorno de programación de código abierto. El lenguaje, que recibe el mismo nombre que el programa, está basado en Java, lo que la sitúa como una herramienta poderosa a la hora de realizar proyectos complejos. En el caso de este proyecto, el programa principal está desarrollado en este entorno. Es este programa el que, al ejecutarse, se comunica con las cartas electrónicas y el Arduino y es capaz de realizar los cálculos de trigonometría esférica necesarios.

El programa surgió inicialmente como un programa a modo cuaderno de bocetos, en el que se enseñaba a programar con una ayuda de un contexto visual. De este punto ha evolucionado a una herramienta para uso de profesionales. Cada programa es conocido como un sketch o boceto, y cada uno de estos permite realizar presentaciones en dos y tres dimensiones. Sus capacidades se extienden con el uso de librerías. Las librerías permiten que los sketches realicen funciones más allá del código de procesamiento central. Existen cientos de estas librerías de carácter libre, y desarrolladas por la comunidad de Processing, éstas pueden unirse a nuestro programa para aumentar las funciones del mismo, como permitir trabajar con sonidos, geometría 3D avanzada. Estas herramientas hacen que los códigos necesarios para crear un sketch sean más sencillos.

El entorno de desarrollo de Processing tiene es potencialmente configurable, permiten incluso generar códigos que se puedan cargar sobre sistemas operativos como Android y que aumenta las posibilidades y compatibilidades de sus funcionalidades

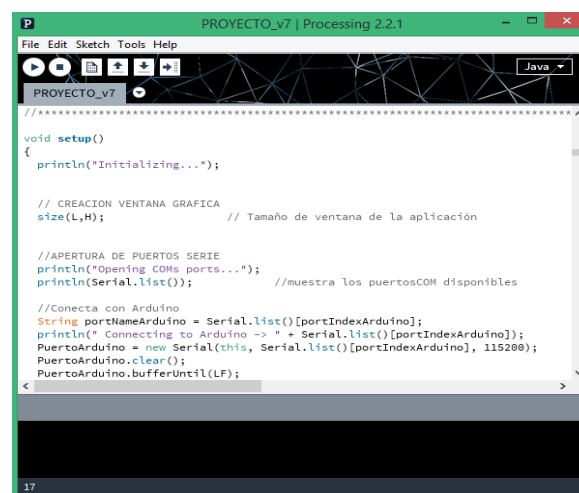


Figura 3-11 Entorno Processing

El código que se ha desarrollado en este entorno es el código principal del proyecto. Su diseño es completamente nuevo y no parte de ninguno previo como en el caso de Arduino, y sobre dicho código se han centrado los esfuerzos del trabajo. El código se ha desarrollado siguiendo el citado modelo

iterativo incremental, aumentando cada versión del mismo con las diferentes mejoras y aplicaciones que si iban desarrollando.

Está compuesto por tres componentes principales que responden a los requisitos básicos. El primero es la comunicación con los puertos y la obtención de la información de los programas auxiliares como son el OpenCPN y la información que proviene de Arduino. El segundo componente principal es la resolución analítica del problema de posición, para ello se realiza el desarrollo matemático de este tipo de geometrías y su forma de proyección en el plano cuyo análisis se expone posteriormente y cuyo estudio es parte del apartado de prácticas del presente proyecto. El último componente es el apartado gráfico del problema, es el que mayores problemas y necesidades de mejora presenta, pero al ser un trabajo cuyo resultado se aprecia de manera gráfica, pequeños cambios e ideas suponen grandes cambios que se pueden visualizar.

El código está distribuido en las siguientes funciones:

Generales:

```
setup (), loop (), serialEvent () .
```

Analíticas:

```
lat_ortodromica (), long_loxodromica (), xgraf (), ygraf (),  
AjustarCuadrícula () .
```

Gráficas:

```
PintarMarca (), TrazarInterseccionOrtodromicas (),  
TrazadoLoxodromicas (), PintarCuadrícula (), TextoLongLatv (),  
TextoLongLatErr (), Giroscopica (), Flecha () .
```

Obtención de datos:

```
ExtraeDemoras (), ExtraerMarcas () .
```

3.9.2 Ortodrómica, solución analítica

La trayectoria ortodrómica es el arco del círculo máximo que une dos puntos de una esfera, se caracteriza por ser la distancia más corta. Una línea de demora está representada por una ortodrómica.

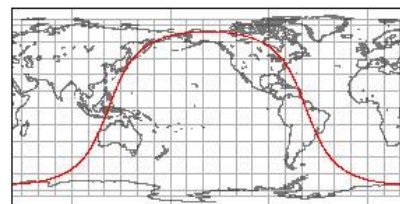
Toda ortodrómica queda definida por sus constantes, su cálculo es el objetivo para definir su descripción analítica. Estas constantes son:

α = longitud del punto de corte con el Ecuador

β = ángulo de inclinación de la misma respecto al Ecuador.



EL MISMO 'GRAN CÍRCULO' EN TRES VISTAS AZIMUTALES ORTOGRÁFICAS DIFERENTES



UN 'GRAN CÍRCULO' EN UNA PROYECCION CILÍNDRICA

Figura 3-12 Trayectoria Ortodrómica (24)

El cálculo se hace basándose en el conocimiento de las coordenadas de un punto, el de la marca seleccionada en tierra y conocido el rumbo inicial. Partiendo del análisis del caso a resolver se deduce que:

$$\cos\beta = \cos(l) * \sin(Ri) \quad [3-12]$$

La obtención de α requiere del previo cálculo de $(L - \alpha)$. Su valor corresponde al nodo determinado por el sentido del rumbo inicial (R_i). La determinación de $(L - \alpha)$ se puede realizar de dos modos distintos:

- Con la ecuación ortodrómica:

$$\text{sen}(L - \alpha) = \text{tg}(l) * \text{tg}(R_i) \quad [3-13]$$

- Con el rumbo inicial:

$$\text{tg}(L - \alpha) = \text{sen}(l) * \text{tg}(R_i) \quad [3-14]$$

Conociendo este término podemos calcular la última constante que queda para definir la trayectoria:

$$\alpha = L - (L - \alpha) \quad [3-15]$$

Una vez están calculadas las constantes el trazado de la ortodrómica es en función de la variación de latitud y longitud, ambas funciones dependientes por estas constantes. La parte analítica de cálculo del código obtiene la latitud a partir de la variación de longitud. Del siguiente modo:

$$\text{lat}(\text{orto}) = \text{arctg}(\text{sen}(L - \alpha) * \text{tg}(\beta)) \quad [3-16]$$

Según se varía la longitud avanzando hacia la demora establecido como rumbo inicial se obtienen las respectivas latitudes de los puntos que conforman el círculo máximo. (25)

3.10 Interfaz Gráfico

En la Figura 3-3 se presenta la pantalla de trabajo del programa sobre la que se muestra gráficamente la obtención de la situación geográfica del buque. El objetivo es hacer sencillo y visual el entorno de trabajo.

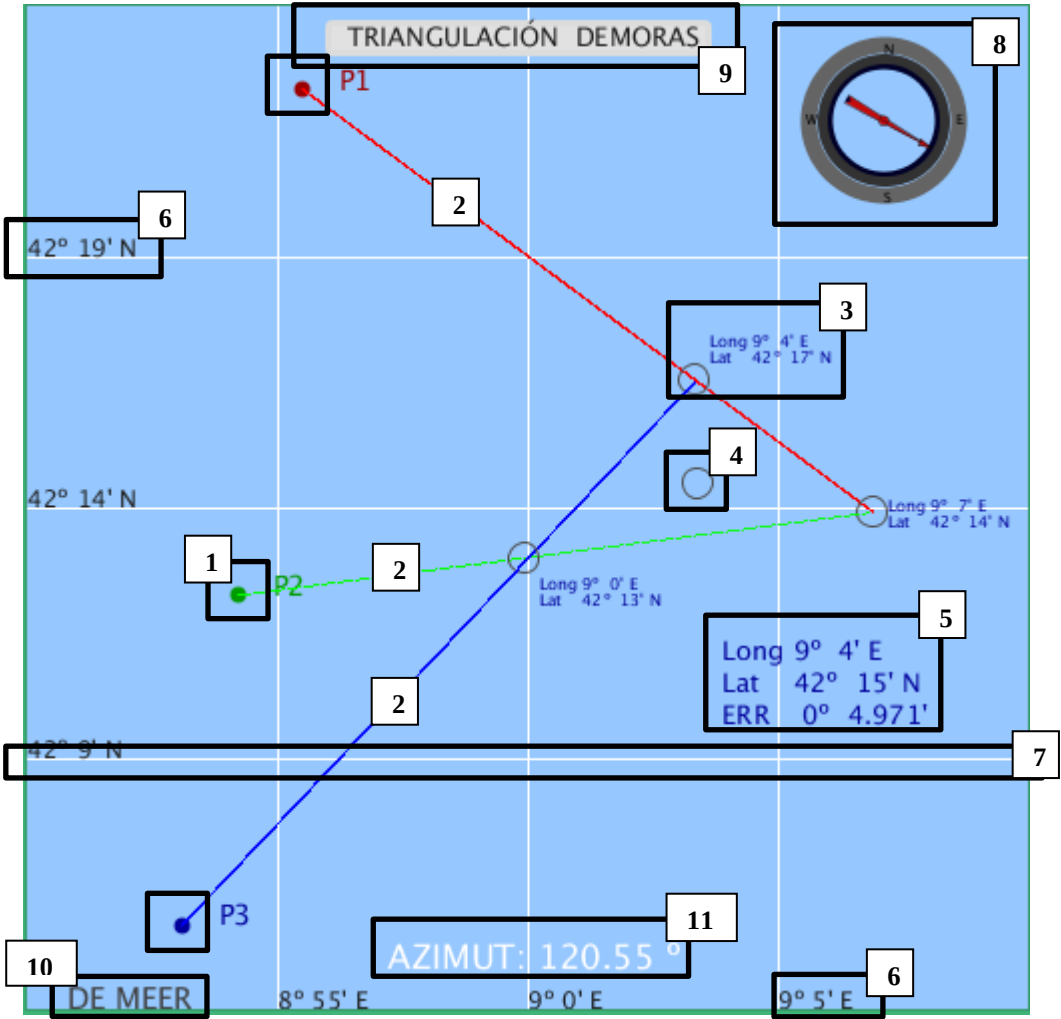


Figura 3-13 Programa Interfaz

Elementos visuales del programa, la presente figura muestra la parte gráfica representada en una ventana al activar el programa, es necesario el análisis de cada uno de los elementos que se visualizan para comprender la información que se está representando. Es el elemento visual de los cálculos que realiza el programa:

1. Referencias seleccionadas, los puntos representados se corresponden con las marcas seleccionadas desde Open CPN, y están situadas de acuerdo con la información de posicionamiento geográfico recibida.
2. Líneas de posición, representan las demoras con las que se registran dichas marcas desde la posición de observación a bordo del buque.
3. Puntos de intersección de demoras, estos puntos constituyen el denominado triángulo de posición en el que, si la información recibida es fiable, debe estar situado el buque.
4. Centro de gravedad del triángulo de posición, resultado presentado en un formato visual.
5. Información del centro de gravedad del triángulo en coordenadas geográficas y error de la situación.
6. Coordenadas geográficas de paralelos y meridianos, su distribución es en tres tercios de la ventana de representación y se aproxima su resultado al número entero de minutos del meridiano/paralelo en cuestión.

7. Red de meridianos y paralelos, suponen una buena referencia para observar la situación de los puntos y poder hacerse una idea visualmente del dimensionamiento del área de observación.
8. Brújula inercial, es la representación en pantalla del movimiento del sensor inercial, la flecha se mueve solidaria al cambio de información transmitido por el sensor inercial.
9. Título.
10. Referencia a la autoría del trabajo
11. Valor que indica de forma numérica el ángulo que está midiendo el sensor inercial y que se corresponde con la demora a la que se está apuntando con la alidada azimutal.

3.11 Montaje

El montaje comienza una vez finalizado el diseño detallado. Durante esta etapa se realiza la adquisición de materiales y se formalizan la adquisición de ciertos elementos y sistemas.

Los trabajos diferenciados en este apartado responderán a los requisitos establecidos anteriormente y seguirán las instrucciones de diseño, si es necesario se pueden proponer cambios al diseño original que mejoren el sistema en cualquier aspecto.

Los circuitos electrónicos, preliminarmente montados en las *protoboards* se disponen sobre una caja en la que se integra la palca de Arduino, el zumbador piezoeléctrico, un circuito de diseño propio en el que se sueldan los LEDs y también tiene integrado el sensor inercial. De la caja anterior se prolonga a través de un cable un mando de control con dos botones que envían las órdenes al sistema para tomar demoras y reinicializar el sistema el sistema está en el tercer plano del Anexo III.

3.12 Integración

La integración de todos los diferentes programas, sensores y sistemas que se emplean en el proyecto hace que la gestión de la integración sea de vital importancia para el funcionamiento del sistema de demoras.

3.13 Toma de decisiones

La toma de decisiones del proyecto será justificada y documentada de forma que queden identificados las alternativas planteadas, el método de análisis seguido, las situaciones de compromiso y los criterios utilizados en la elección.

4 PRUEBAS Y VALIDACIÓN DEL SISTEMA

4.1 Plan de pruebas

Las pruebas que se han llevado a cabo para analizar el sistema se pueden dividir en dos grupos, el primero recoge un conjunto de pruebas analíticas con diversos ejemplos y situaciones del supuesto buque en diferentes posiciones para comprobar las aplicaciones del programa y su funcionamiento. Por otro lado también se recogen los resultados obtenidos en las prácticas realizadas en el Simulador de Navegación.

4.1.1 Pruebas analíticas

Una vez desarrollada la parte analítica del código y desarrollada la parte de presentación gráfica podemos obtener y analizar distintos resultados de acuerdo a los parámetros establecidos que son introducidos en el programa. La facilidad de obtención de datos a partir de la cartografía náutica digital permite plantear infinidad de problemas de posicionamiento de forma sencilla, y avanzar de manera efectiva en el desarrollo del código supliendo el mayor número de fallos del mismo.

En las siguientes figuras se desarrollan las mencionadas pruebas, explicando en cada uno de los casos lo que se pretende analizar, su utilidad y propuestas mejora del programa.

A continuación se relacionan los ejercicios analizados de acuerdo con los posibles casos de estudio y análisis establecidos:

- Ejercicio 1. Triangulación.
- Ejercicio 2. Cuadrantes.
- Ejercicio 3. Curvaturas.
- Ejercicio 4. Intersección Loxodrómica.
- Ejercicio 5. Precisión.

- Ejercicio 1. Triangulación:

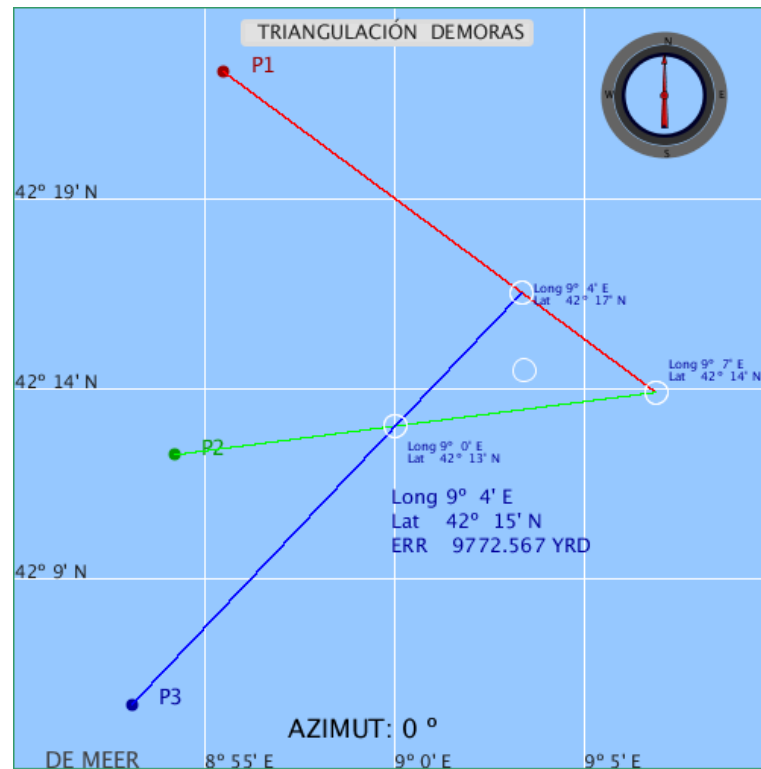


Figura 4-1 Triangulación

En primer lugar, en la Figura 4-1 se muestra uno la comprobación del cálculo de posicionamiento, el cálculo del triángulo de posicionamiento. Las marcas P1, P2 y P3 ocupan la posición de las marcas en tierra a las que se les toman las demoras. El programa autoajusta la ventana gráfica de acuerdo con las coordenadas geográficas de los puntos que se envían y se pintan las demoras medidas a partir de las posiciones en tierra, su delineación tiene lugar hasta el momento en que se obtiene el punto de intersección común de las distintas demoras dos a dos. Los círculos blancos entre cada demora representan estos puntos de intersección, en los que se despliega, además una etiqueta con la posición de los mismos.

En el centro del triángulo de posición aparece otro círculo del mismo color, debajo del triángulo encontramos las coordenadas de dicho punto, que, dependiendo del error, y la posible confianza de las demoras tomadas, se puede considerar como la posición geográfica del buque.

En este caso el error es desorbitado, pero el motivo es el haber realizado una prueba como escenario en la que se visualicen sin problema el conjunto de elementos y los resultados que se pueden obtener.

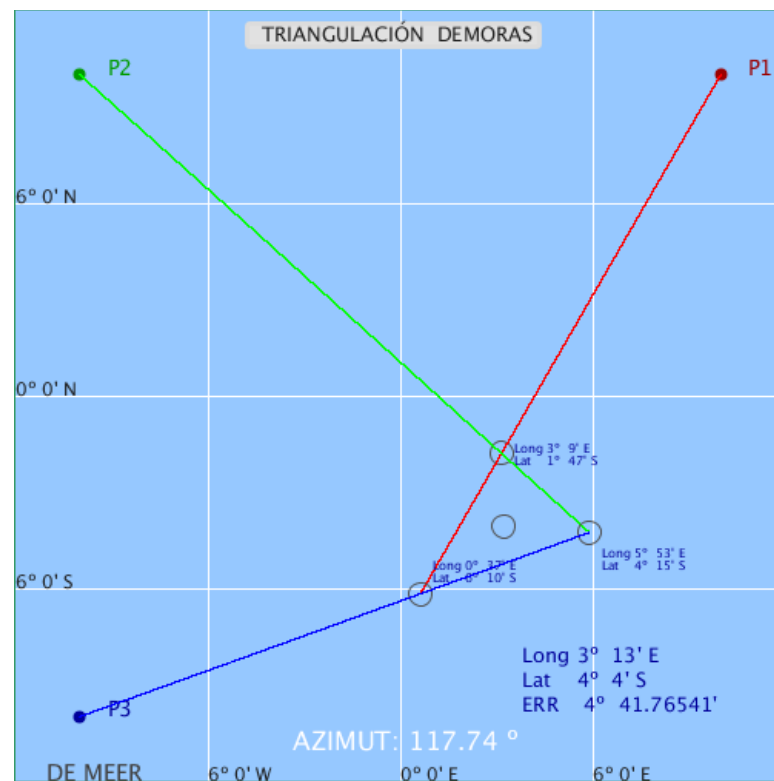


Figura 4-2 Cuadrantes

El objetivo del siguiente ejercicio es comprobar que el programa puede trabajar sin distinción en cualquiera de los cuadrantes, para ello las marcas en tierra se sitúan cada una en una situación irreal, separadas varios grados unas de otras y centrando la ventana gráfica en la intersección entre el meridiano de *Greenwich* y el Ecuador.

Una vez situadas las marcas, se seleccionan las demoras necesarias para que el triángulo de posicionamiento se sitúe en el único cuadrante de la carta en blanco que queda libre tras situar las marcas.

En esta resolución se observa como el error es también desmesurado, en este caso se trata de un problema ficticio de comprobación de los cálculos analíticos.

El resultado es favorable ya que todos los distintos parámetros se sitúan en sus posición, y se obtiene una posición analíticamente válida de acuerdo a los datos introducidos, los elementos quedan representados en todos los cuadrantes admitiendo latitudes sur y longitudes oeste negativas y diferenciándolas del resto, en la cadena de caracteres que envía OpenCPN y el código de cálculo de Processing.

- Ejercicio 3. Curvatura:

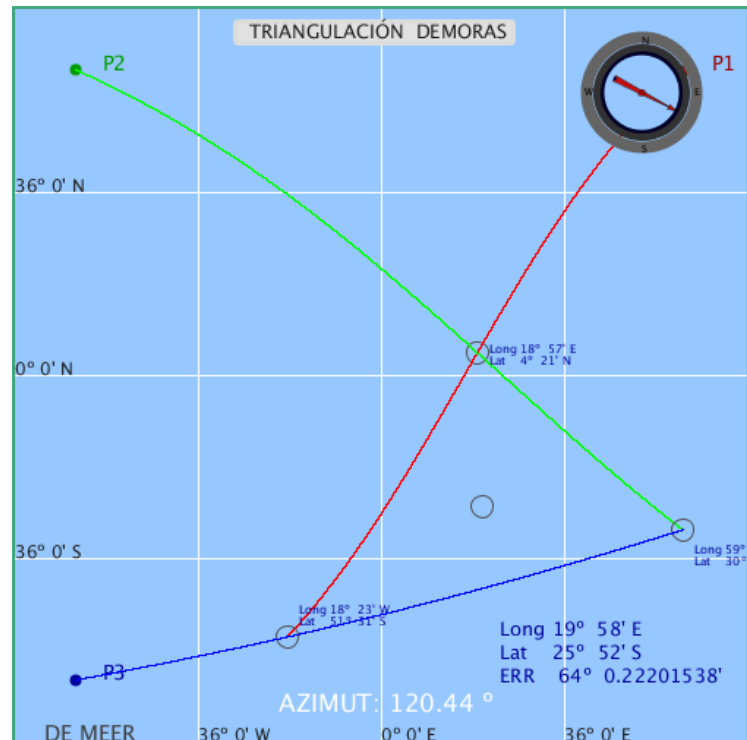


Figura 4-3 Curvatura

Como un mero interés teórico y didáctico el ejemplo sirve para demostrar la forma de visualización y representación de la información que aparece en el programa. Con este ejercicio comprobamos que la grafía se está haciendo correctamente en la ventana gráfica.

La ventana gráfica, por las funciones que la componen, se puede definir por sus propiedades como una carta digital en blanco y autoajustable. Esta carta en blanco no es el resultado de ninguna transformación analítica como la mercatoriana o cualquier otro tipo de proyección, es en definitiva una proyección directa, por eso mismo la malla con las coordenadas que se presenta es simplemente un método directo para situar los puntos conspicuos y los resultados.

La demostración de la Figura 4-3 es que la representación de los círculos máximos son curvos sobre la ventana de representación, y por tanto el triángulo de posición se transforma en esta figura geométrica curva.

Su interés no se centra en la situación, pues, para comenzar a apreciar la curvatura de la recta los puntos deben de estar separados varias decenas de grados, lo que desde un aspecto físico no tiene sentido, teniendo en cuenta que la Navegación Costera por demoras se realiza dentro del horizonte visible. Dependiendo de las condiciones atmosféricas, el alcance no suele superar las 20 millas náuticas (26), sabiendo que es necesario tener en cuenta la altura del punto conspicuo para realizar dicho cálculo.

- Ejercicio 4. Loxodrómica:

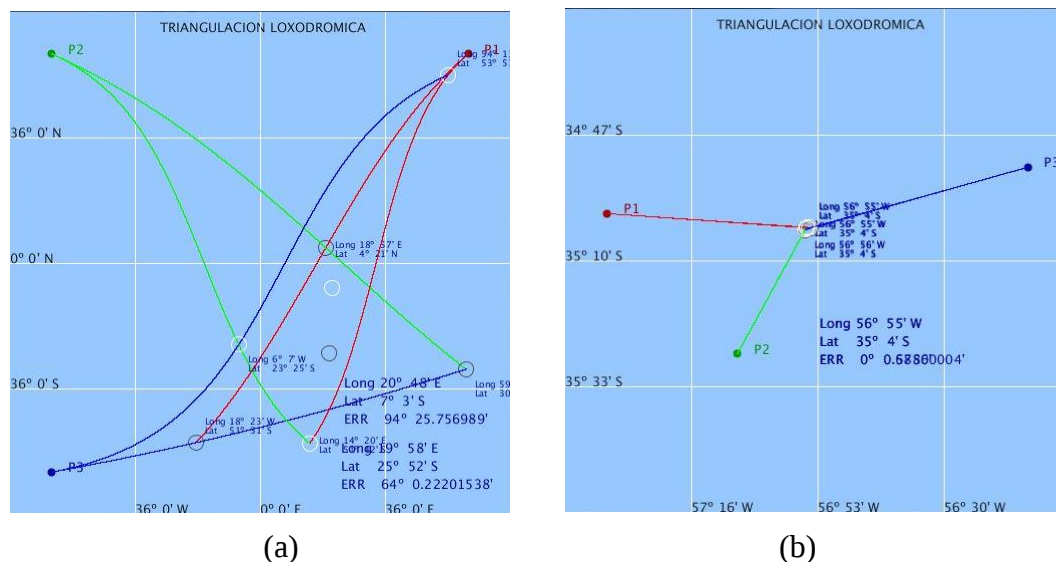


Figura 4-4 Loxodrómica

En este ejercicio, se pretende comprobar el error que se realiza al dibujar las demoras en las cartas mercatorianas de papel. Las demoras sobre las cartas mercatorianas se pintan como una recta cuando, en realidad por definición, son arcos de círculos máximos, luego la verdadera trayectoria entre la situación y el objeto al que se toma una demora es una trayectoria ortodrómica.

Como se ha enunciado, la representación de la loxodrómica es la que se representa como una recta, no la ortodrómica, debido a la poca distancia se permite dicha aproximación, para facilitar la delineación de las demoras en la carta y porque la diferencia en casos de corta distancia es inapreciable.

En la Figura 4-4 se puede comprobar la diferencia entre los distintos resultados. En el primer caso, Figura 4-4 (a), las distancias entre las marcas mantienen entre ellas distancias desproporcionadas a una situación común, del orden de varias decenas de grados. El primer problema tiene como objetivo reafirmar que las trayectorias que se consideren son distintas, y por tanto el cálculo no es el mismo. Se observa como ambas trayectorias mantienen la curvatura, de acuerdo con las conclusiones del ejercicio anterior. La intersección de las loxodrómicas se representa con círculos negros, y la de las trayectorias ortodrómicas con círculos blancos.

Si bien en la primera imagen se puede observar la diferencia entre ambos resultados, en el caso de la segunda imagen, Figura 4-4 (b), coinciden todos en un mismo punto. Este ejemplo sitúa al buque en la desembocadura de río de la Plata en Sudamérica, y la diferencia entre realizar un cálculo con una ecuación ortodrómica y de realizarla con una loxodrómica, suponen una diferencia de 2 yd o 1,8m. Y gráficamente se puede confirmar que la aproximación a ciertas escalas entre una y otra es prácticamente la misma.

- Ejercicio 5. Precisión:

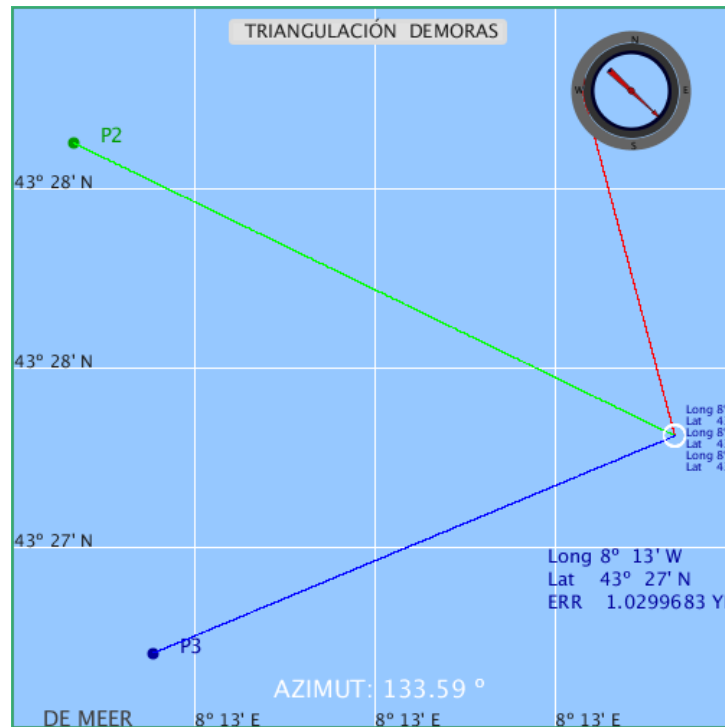


Figura 4-5 Precisión

El último ejercicio analítico tiene como objetivo establecer la precisión del sistema de posicionamiento, en el análisis no se tienen en cuenta los posibles errores accidentales debidos a las condiciones meteorológicas o al error de medición humano. Para ello se introducen los datos directamente en el programa, que previamente se han calculado con OpenCPN, en este caso la posición del buque dentro de la zona portuaria de Ferrol.

Las trayectorias ortodrómicas, puesto que se van obteniendo avanzando en la longitud, dependiendo del paso que se establezca puede variar la precisión del sistema, del mismo modo que se la distancia entre la posición del buque y las marcas aumenta considerablemente, entonces también se multiplica el tiempo de cálculo y la espera de obtener la situación, tiempo en el que el buque puede haber avanzado considerablemente las posición respecto a la de medición y esta se deberá descartar.

Por tanto existe un compromiso entre la rapidez de cálculo y la precisión del mismo que debe adecuarse generando una función que cambie automáticamente el paso en función de la distancia entre el corte de las demoras y la posición de las marcas.

Como se puede observar en la Figura 4-5 la precisión que se ha obtenido en las condiciones establecidas de minimización de errores ajenos a los analíticos es de aproximadamente 1 yd que equivalen a 0,9m. Por este motivo, se puede comprobar la corrección del planteamiento analítico del trabajo.

4.1.2 Simulador de Navegación

También se han realizado pruebas en el Simulador de Navegación de la Escuela Naval Militar, para este caso, y aprovechando los escenarios del simulador, se ha querido cargar un escenario de la ría de Ferrol, y para poder compararlo con el Ejemplo 4, se ha decidido obtener una prueba de precisión.

En el caso del simulador se posiciona el buque en la misma zona y se reconocen las puntas que están marcadas en la carta electrónica, y que se podrían comprobar en la carta de papel. Para las condiciones del simulador se mantiene el barco en estacionario anulando los efectos de viento y corriente que se pueden cargar en el sistema. Se envían desde Open CPN los puntos elegidos y reconocidos al programa, y este los reposicione en su carta en blanco.

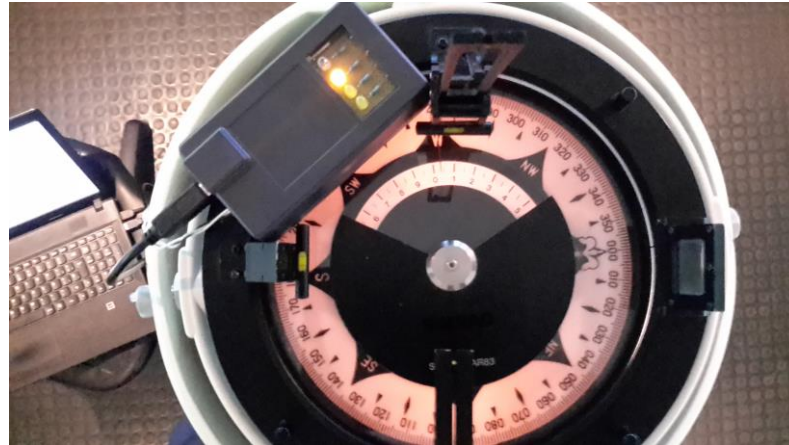


Figura 4-6 Sistema de toma de demoras

Una vez reconocidas las puntas en el escenario del simulador se procede a realizar una situación con el sistema de toma de demoras. En primer lugar se reinicializa el norte, para tomar la misma referencia que el norte del simulador, y poder tomar el resto de demoras verdaderas respecto a la orientación del simulador.

Posteriormente se realiza la toma de demoras de cada una de las marcas, apuntando con el plano de las pínulas de la alidada a cada una de las marcas y pulsando el botón “tomar demoras”. Cuando ya se han recogido las tres demoras el programa resuelve la situación geográfica del buque, obteniéndose un error al calcular el triángulo de posición de 26 yd que equivalen a 23,4 m.



Figura 4-7 Prueba en el Simulador de Navegación

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

De acuerdo con las metas establecidas en el planteamiento del trabajo podemos concluir de exitoso el prototipo que ha resultado del presente trabajo. Paso por paso se van adquiriendo pequeños hitos, y día a día se avanza hacia la resolución de problemas y el estudio de posibles mejoras.

Algunos de estos pequeños objetivos son los siguientes:

Según las especificaciones, se ha conseguido desarrollar un programa en el entorno de desarrollo de Arduino, entendiendo su composición y realizando cambios en las librerías para optimizar la respuesta del sensor inercial de acuerdo con las necesidades y leyes de movimiento característicos de las alidades.

Se ha investigado e incluso adjuntado en el trabajo un estudio de los sensores inerciales o IMUs, sus aplicaciones, las posibilidades que ofrecen y también sus pequeños fallos. Se ha desarrollado también, un interfaz de usuario sencillo, rápido y sin posibilidad de error para el usuario. Y en el propio sistema se integran sensores visuales y acústicos a través de los cuales se puede conocer si la información ha sido enviada.

También se ha creado una red de comunicaciones entre los distintos programas a través puertos serie, recurriendo a la creación de conexiones entre puertos virtuales para la transmisión de la información en sus respectivos protocolos de comunicación.

Respecto al prototipo, es cierto que no se ha creado un sistema todo lo portable que se había planteado y diseñado, aunque sí es un sistema práctico. El resultado es debido a que la atención se ha centrado en la resolución del problema de situación, dejando tiempo para la realización de mejoras en el sistema. En el caso del tiempo de mejora resulta más práctico resolver y mejorar los códigos que emplea el programa para optimizar los sistemas que el hecho de replantear una nueva estructura desde cero.

El desarrollo de un programa para la obtención de la posición del buque cumple con las expectativas, la elección del entorno de programación Processing es un acierto, no por el potencial del programa comparado con otros entornos de programación, más bien, por el carácter didáctico de Processing, al ser posible ‘pintar’ en la ventana gráfica gran parte del código con lo que visualmente se puede comprobar e ir entendiendo las distintas funciones y su aplicación.

Una vez que se han podido realizar pruebas el resultado de las mismas ha sido satisfactorio, sin embargo con algo más de tiempo para programar nuevas pruebas se podrían haber llevado a cabo un

mayor número de ensayos en el mismo Simulador de navegación, e incluso haber embarcado en distintas unidades como las Lanchas de Instrucción u otros buques.

Sin que se haya incluido, el presupuesto se ha tenido en cuenta en todos los aspectos del trabajo, el uso de software libre y el empleo de Arduino garantizan esta peculiaridad del sistema. Aun así es necesario aclarar el elevado coste de todo el material náutico, en el caso de la alidada se ha recibido una del Simulador de Navegación, y en el caso de querer naval izar el sistema para que pueda estar a la intemperie deberá estudiarse otro prototipo con otros materiales y diseño.

5.1.1 Fallos

Los fallos con los que se han encontrado en el sistema son los siguientes:

- Estabilidad del sistema

Los fallos de estabilidad están referidos al sensor inercial, normalmente, aun realizando la calibración justo previamente antes de la puesta en marcha, se observan pequeñas variaciones en torno a un grado que tienen mayores consecuencias en proporción directa a la distancia de la posición. Los sensores tienen además otro problema, cuando tiene que cambiar el sentido al pasar de 180° a -180° dan un pequeño salto de valores, lo que se conoce como ***gimbal lock***. (27)

- Fallos de saturación

Son fallos puntuales que pueden llegar a aparecer según la presión de la situación, en el caso de los programas, se han registrado fallos en el OpenCPN, que en ocasiones tiene que cerrarse y es posible que se pierdan las marcas. También fallos en la ejecución del programa de triangulación, cuando el ‘paso’ para calcular el círculo máximo no está establecido acorde con las distancias, el programa puede entrar en un bucle sin salida que no converja a una solución.

5.2 Líneas futuras

El apartado es de utilidad para marcar nuevos objetivos desde el punto en que se da por finalizado el presente trabajo, para poder estudiar su posible repercusión así como para señalar los futuros pasos en los que se puede investigar para seguir desarrollando el, hasta ahora, prototipo de toma de demoras electrónicas.

5.2.1 Mejoras

Entre las mejoras que se pueden plantear está el diseño y construcción de algún sistema más desarrollado y que incluya componentes como el Arduino NANO, para reducir el espacio. También existe la posibilidad de desarrollar el método para que el sistema de toma de demoras se pueda comunicar de modo inalámbrico y evitar el cableado de por medio del sistema, ello llevaría asociado el estudio de un sistema de alimentación mediante batería y un transmisor radio de muy bajo consumo para garantizar la autonomía.

Se puede estudiar la fusión de este trabajo con otro TFG, el trabajo compuesto de los Alféreces de Fragata (AAFF) Gerardo y Santiago González- Aller Rodríguez. El TFG realizado por estos alumnos consiste en el diseño de una aplicación móvil para Android de posicionamiento astronómico a partir de un móvil. Una integración de ambos sistemas de navegación puede resultar en la creación de un nuevo sistema de navegación.

El cálculo de parámetros de navegación de estima teniendo en cuenta la integración de los diferentes posicionamientos del buque, pueden ayudar a este a estimar corrientes si se realizan las posiciones con la debida precisión.

Otra de las posibles mejoras es el estudio de los distintos filtros de fusión ya que en este trabajo sólo se ha utilizado el Filtro Complementario, que es el más sencillo. Otros filtros de fusión como el de Kalman, Kalman-Extendido, etc. mejorarían la calidad de las medidas y el procesamiento de datos, permitiendo obtener mejores resultados, más precisos y suficientemente rápidos. (28)

Otra de las posibles mejoras es el estudio de los distintos filtros como el de Kalman, que se añaden sobre la información recibida. El propósito es mejorar el procesamiento de datos y permite obtener mejores resultados, más precisos y suficiente mente rápidos.

La última de las aplicaciones propuestas es el empleo de los sensores inerciales de acuerdo a sus propiedades. En el caso del acelerómetro existen varias soluciones, una de ellas es estimar el yaw por medio de dos acelerómetros instalados sobre el mismo objeto giratorio pero con posiciones diametralmente opuestas. (29)

También se puede utilizar una patente como la de '*Method and Apparatus for Vehicle Yaw Rate Estimation*', que surgió para cubrir las demandas de seguridad de los automóviles en el que se emplean distintos sensores. Su esquema se puede visualizar en la Figura 5-1.

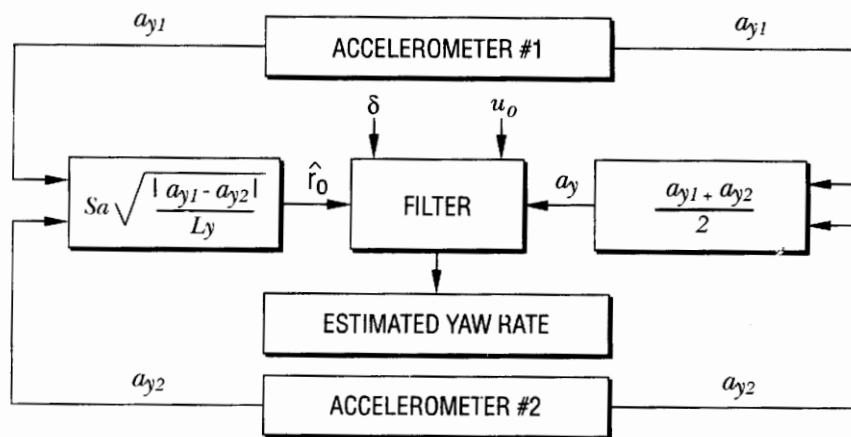


Figura 5-1 Patente cálculo de yaw (29)

Otro método relacionado con los acelerómetros sería tratar de realizar un cambio en la ley de giro, para que el movimiento horizontal de la alidada se traslade al plano vertical para obtener una mayor precisión del ángulo. En este caso a través de un conjunto de motores síncronos se podría automatizar el proceso.

En el caso de querer navalizar el sistema para que pueda estar a la intemperie deberá estudiarse otro prototipo con otros materiales y diseño. Además, habría que hacer un estudio de los diversos modelos de alidada que emplean las distintas unidades de la Armada.

6 BIBLIOGRAFÍA

En esta sección figuran todas las referencias del proyecto:

1. **AudieMan.** La bitácora de AudieMan. [En línea] 2010. <http://audiemangt.blogspot.com.es/2010/05/metodologia-agil-msf-microsoft-solution.html>.
2. **Escuela Naval Militar.** *Introducción a la Navegación Costera*. s.l. : CAE-UVICOA, 2011.
3. **Curbera, C.N. Martínez Jiménez C.N. Moreu.** *Artronomía y Navegación Tomo I*. Vigo : Librería San José, 1970.
4. **BOE orden FOM/1144.** Ministerio de la Presidencia. *Boletín Oficial del Estado*. [En línea] 2003. <http://www.boe.es/buscar/doc.php?id=BOE-A-2003-9581>.
5. **SHOA.** Rincón del Navegante. [En línea] http://www.shoa.cl/Vaul/Vaul/meteonavegacion/Cartas_de_Navegacion.pdf.
6. **Olaya, Víctor.** *Sistemas de Información Geográfica*. [En línea] 2011. <file:///C:/Users/Alejandro/Downloads/852260379.libro-SIG-cap-3.pdf>.
7. **Gamboa, José Manuel Millán.** JM Ediciones. [En línea] Instituto Hidrográfico de la Marina, 2004. <http://www.jmediciones.com/RINCON/esqueleto.html>.
8. **Curbera, C.N. Martínez Jiménez y C.N. Moreu.** *Astronomía y Navegación Tomo: III Segundo Curso de Náutica*. Vigo : Librería San José, 1987.
9. **Náutica.** Náutica. [En línea] 2010. <http://naut.blogcindario.com/2010/03/00031-posicion-del-buque-por-demoras-marcaciones-marcacion-corrida-casos-especiales.html>.
10. **Llanes, T.N. Carlos Pajares.** *Infome NEMO 14 EW Trials*. 2014.
11. **Banzi, Massimo.** *Getting Started with Arduino*. s.l. : Make:Books - O'Reilly Media, Inc, 2011.
12. **Pérez, Luis Eduardo Ardila.** *Unidad de Medición inalámbrica para cohetes*. Colombia : s.n., 2011-2012.
13. **Ortuño, José María Núñez.** *Sistemas de Medición Inercial (IMU) un ejemplo de tecnología de uso dual*. Marín : s.n.
14. **CH Robotics.** CH Robotics. [En línea] <http://www.chrobotics.com/library/understanding-euler-angles>.
15. **Geología y Biología.** [En línea] 2011. <http://carrasconoemibio1.blogspot.com.es/2011/09/el-origen-y-estructura-de-la-tierra.html>.

16. **Inven Sense Inc.** MPU-9250 Product Specification Revision 1.0. [En línea] 2014. www.invensense.com.
17. —. MPU-9250 Register Map and Descriptions Revision 1.4. [En línea] 2013. www.invensense.com.
18. **Kalian, Pavel.** GitHub. [En línea] 2013. <https://github.com/OpenCPN/OpenCPN>.
19. **Phidgets Inc.** Phidgets. [En línea] 2014. http://www.phidgets.com/docs/Compass_Primer.
20. **González-Cela, CC. Gerardo.** Apuntes Navegación Tema VI: Rumbos y Marcaciones. *Navegación I*.
21. **Richards-Tech.** GitHub - RTIMULib (Arduino). [En línea] 2014. <https://github.com/richards-tech/RTIMULib-Arduino/blob/master/libraries/RTIMULib/RTIMU.cpp>.
22. **Richards-Thec.** richards-tech notes. [En línea] RTIMULib, 2014. <https://richardstechnotes.wordpress.com/2014/10/09/this-is-why-9-dof-imu-data-is-useless-without-magnetometer-calibration/>.
23. **Tienda de Robótica y el Equipo de Cosas de Mecatrónica.** Tienda Robotica. [En línea] 2012. www.tiendaderobotica.com.
24. **Weisstein, Eric W.** MathWorld--A Wolfram Web Resource. [En línea] <http://mathworld.wolfram.com/GreatCircle.html>.
25. **Curbera, C.N. Jiménez Martínez C.N. Moreu.** *Astronomía y Navegación Tomo III: Curso de Capitanes*. Vigo : Librería San José, 1987.
26. **Montañasdesur.** Montañasdelsur. [En línea] 2013. <http://www.montanasdelsur.es/2013/12/18/distancia-al-horizonte-cuanto-vemos-desde-las-alturas/>.
27. **Jalan Ayer Keroh Lama.** International Journal of Engeneering and Technology Innovation. [En línea] 2011. <http://sparc.nfu.edu.tw/~ijeti/download/V1-35-52.pdf>.
28. **Kalman, R.E.** University of North Carolina at Chapel Hill. *A new Approach to Linear Filtering and Prediction Problems*. [En línea] <https://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>.
29. **Kinonix.** Kinonix. [En línea] 2008. <http://www.kionix.com/sites/default/files/AN019%20Two%20accelerometers%20for%20rotation.pdf>.
30. Web de La Moncloa. [En línea] [Citado el: 13 de enero de 2015.] <http://www.lamoncloa.gob.es>.
31. **Rodríguez, Juan y Fernández, Vicente.** *Cómo redactar el estado del arte de un trabajo*. s.l. : Editorial Genios, 2010.
32. **Martínez, Pedro y García, Ana.** *Cómo escribir una buena memoria de TFG*. s.l. : Publicaciones del 2000, 2013.
33. **Pérez, Antonio.** *Cómo escribir una bibliografía*. s.l. : Nuevas publicaciones.
34. **AudieMan, La Bitacora de.** [En línea] http://audiemangt.blogspot.com.es/2010_05_01_archive.html.
35. **Minsterio de industria.** RD 239049889 para la calafaction . MADrid : BOE, 1991. ISBN 32138/6546845984561.
36. **NOAA.** National Geophysical Data Center. [En línea] National Oceanic and Atmospheric Administration. <http://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml>.

37. **Echevarría, CC. González Cela.** Declinación magnética. *Apuntes Navegación Tema VI Rumbos y Marcaciones.*

ANEXO I: CONCEPTOS

Conceptos del proyecto

- Navegación costera
- Sensor inercial IMU
- Giroscópica
- Arduino
- Canal dragada
- Alidada azimutal
- Hardware
- Software
- Cartas náuticas
- Carta en blanco
- Círculo máximo
- Distancia ortodrómica
- Demora
- Marcación
- Proyección mercatoriana
- Trayectoria loxodrómica
- Aguja giroscópica
- Rigidez
- Precisión
- Taxímetro
- Marcadores de alerones
- Sirviente carta
- Magnetismo permanente
- Magnetismo inducido
- Declinación magnética
- Ángulos de Euler: *roll pitch* y *yaw*
- Puente de gobierno
- gimbal lock

Abreviaturas y acrónimos

- ECDIS
- GPS
- NOAA
- AIS
- NAVTEX
- CEMEDEN
- NMEA
- MPU 9250

ANEXO II: DIAGRAMA DE GANTT

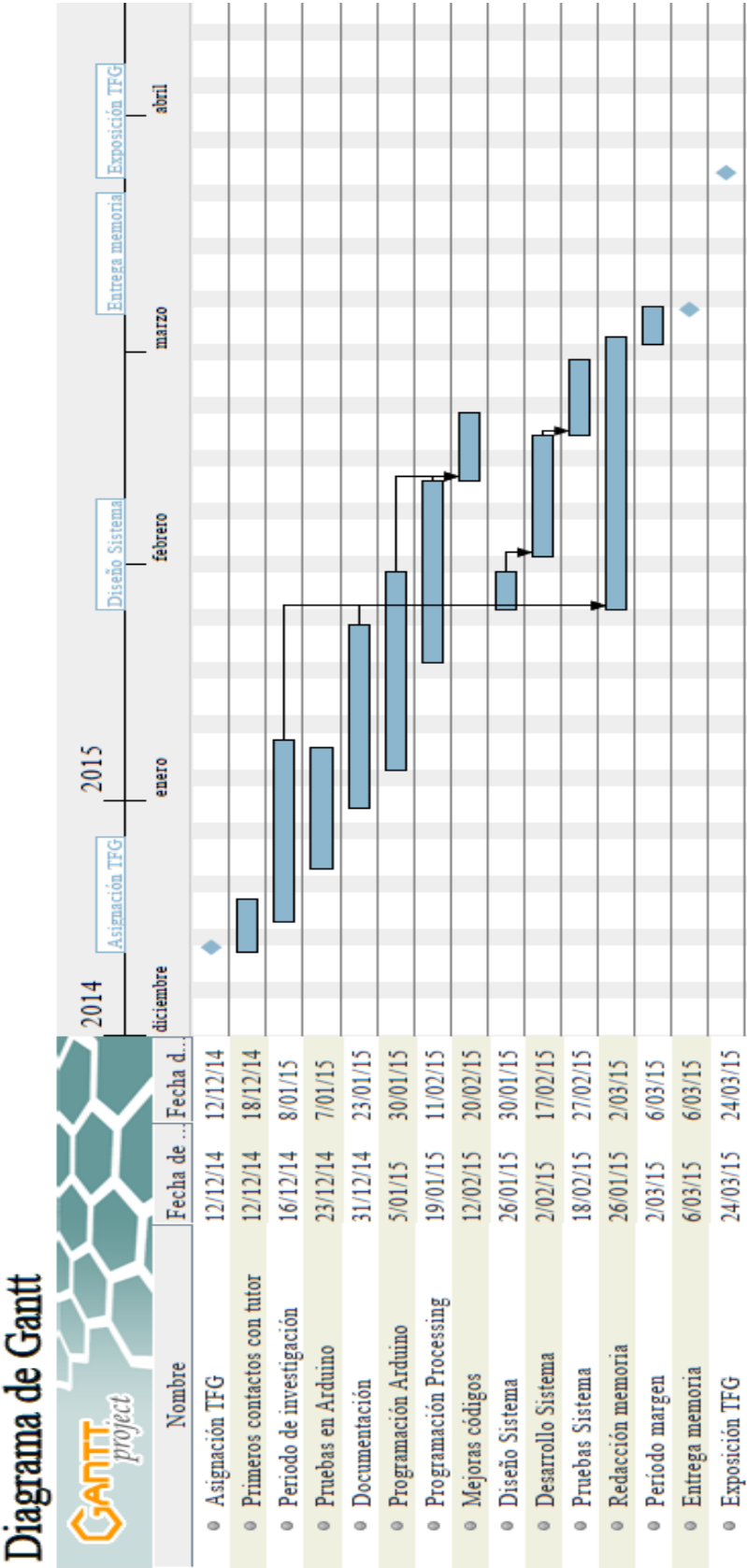
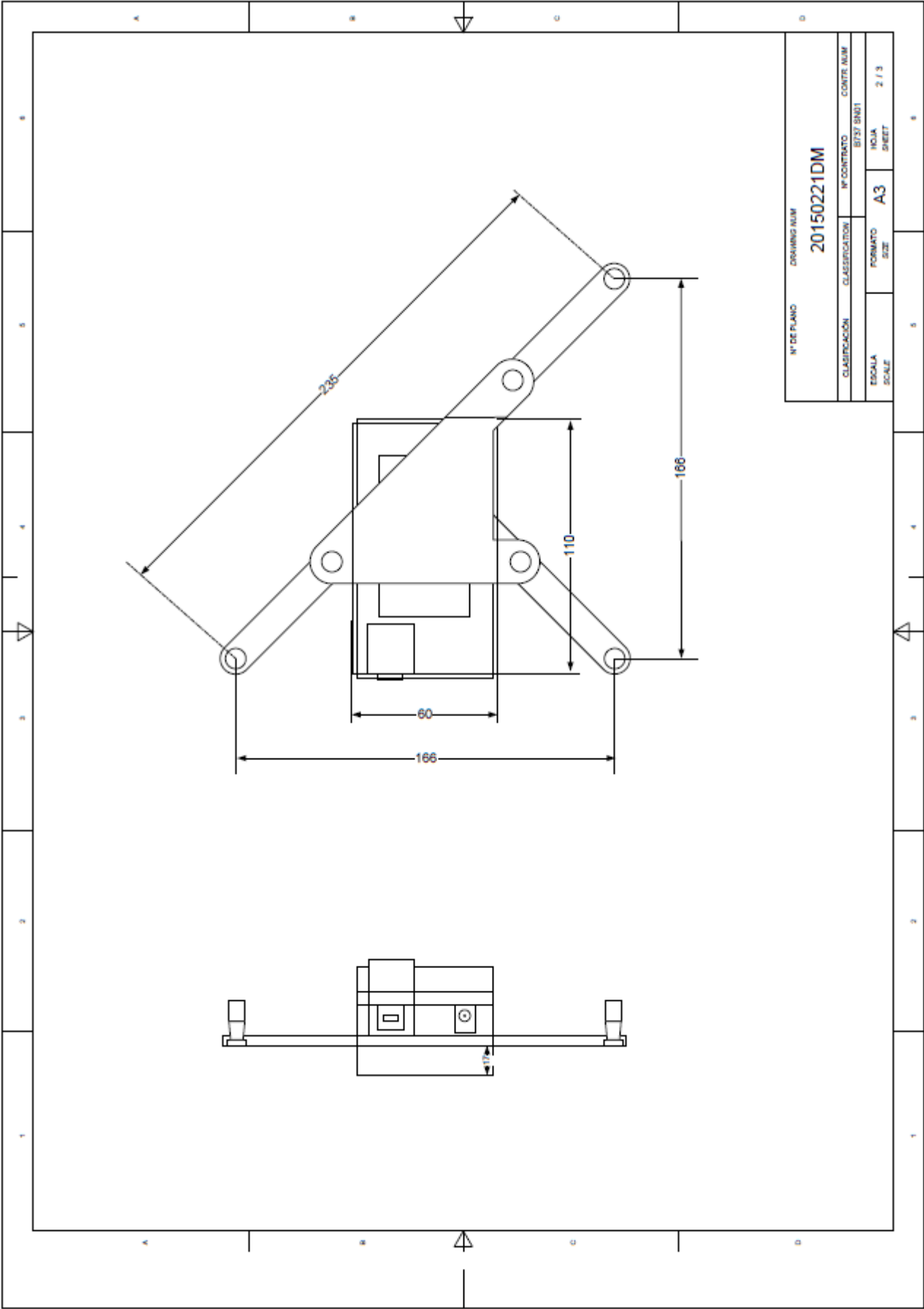


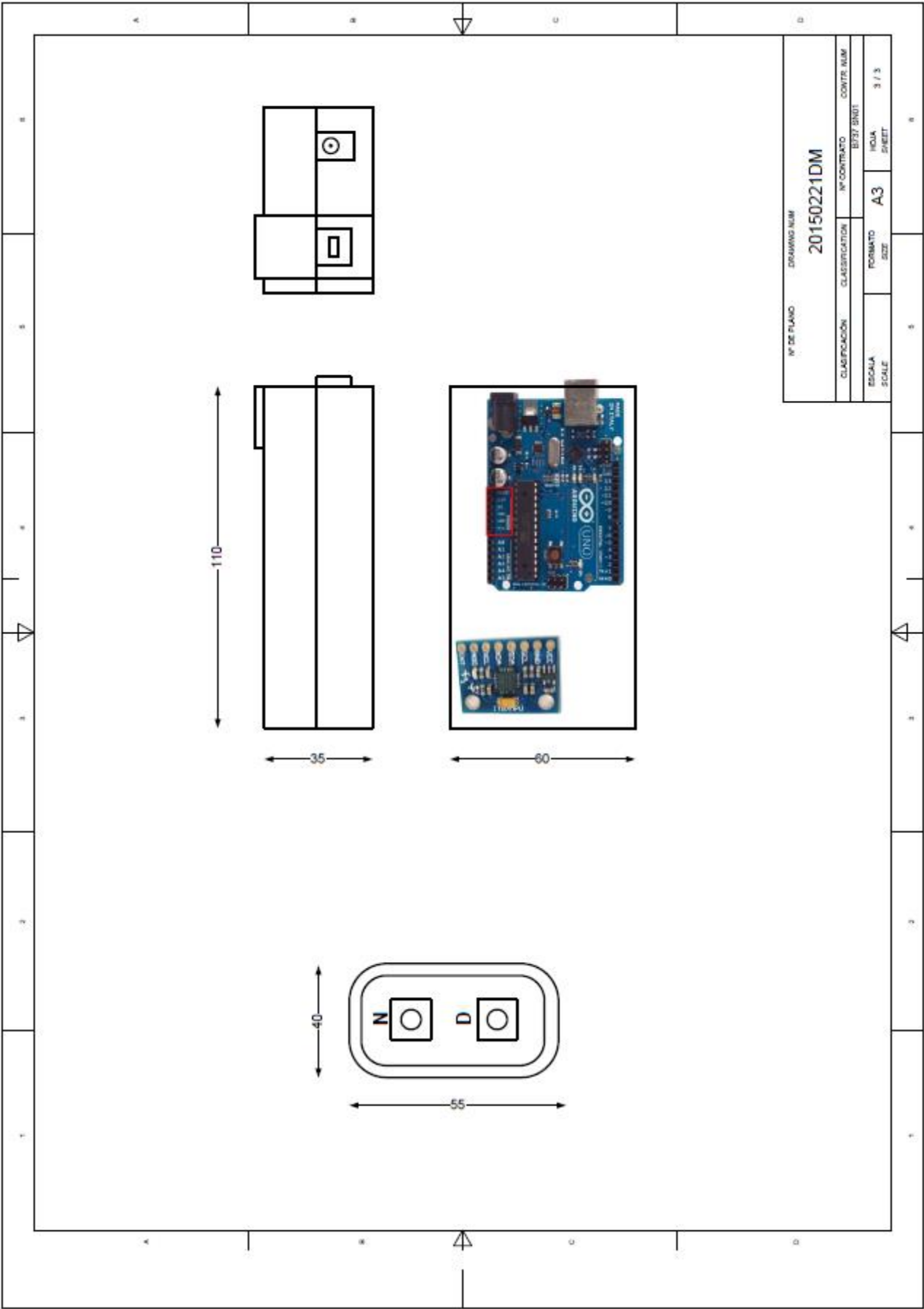
Figura AII- 0- 0-1

Descripción de tareas

- Asignación TFG
- Primeros contactos con tutor
 - En este periodo se plantean los siguientes hitos:
 - Presentación al tutor y establecer contacto.
 - Conocer el área de trabajo.
 - Conocer el enfoque y objetivos generales del trabajo.
 - Comienzo de organización del trabajo
 - Primer enfoque del trabajo en aspectos generales
- Periodo de investigación
 - Tras la asignación del trabajo comienza el periodo de investigación, se pretende profundizar en los campos de trabajo y familiarizarse con las aplicaciones de uso.
- Pruebas en Arduino
 - Periodo de familiarización con el software y los elementos de trabajo de Arduino. Se realizan pequeños proyectos guiados por tutoriales con Arduino.
- Documentación
 - En esta fase se pretende recopilar toda la información posible relacionada con el trabajo y empezar también a filtrar la misma para evitar el exceso de información. Durante el resto del periodo de trabajo se seguirán analizando las distintas áreas de trabajo, pero este en un periodo dedicado exclusivamente a ponerse al día en materia.
- Programación Arduino
 - Una vez están claras las líneas de actuación puede comenzar el trabajo de programación y montaje del circuito electrónico acoplado a Arduino. Este tiempo se subdivide en distintos periodos en los que se persiguen pequeños hitos del código para que vaya funcionando cada parte y luego ir integrando todas ellas.
- Programación Processing
 - Al igual que la programación en Arduino, el programa desarrollado en este software pretende ir resolviendo incrementalmente las distintas funciones en las que se puede ir subdividiendo el código principal. Es importante fijar cuales son las funciones principales y necesarias y cuáles son las posibles mejoras que se pueden desarrollar en el tiempo del que se dispone.
- Mejoras códigos
 - Este periodo está fijado en principio para establecer la fecha en que se pretende tener los códigos listos para dedicar más tiempo al trabajo de la memoria. Establece un margen desde el cual es necesario que funcionen las funciones básicas de los programas.
- Diseño Sistema
 - En este periodo de tiempo se pretende comenzar a diseñar el HW del proyecto, a establecer los materiales que hay que adquirir y proyectar el sistema sobre la alidada azimutal obtenida del simulador de navegación.
- Desarrollo Sistema

- Por desarrollo se pretende definir el montaje y puesta en marcha de cada una de las partes del sistema diseñado y puesto en práctica a través de protoboards. Se pretende dar el paso a la integración de los circuitos soldando las placas y materializando los diseños estudiados adecuándolos a la realidad de cada situación y necesidad.
- Pruebas Sistema
 - Si bien se realizan pruebas de funcionamiento periódicas comprobando el funcionamiento el objetivo es disponer del tiempo necesario para validar y poder analizar los resultados de la verdadera aplicación del sistema, hasta el punto en el que se ha podido desarrollar hasta el momento.
- Redacción memoria
 - Es importante el tiempo reservado para desarrollar la memoria del trabajo. también fijar plazos para ir obteniendo resultados y avanzar paralelamente al proyecto la parte que recoge toda la información del trabajo.
- Período margen
 - Se pretende marcar un tiempo de finalización para asegurar la entrega y poder analizar cómo enfocar y si es posible desarrollar distintos campos dentro del proyecto para mejorar el mismo. Es un buen periodo para recapacitar acerca de las consecuencias y repercusiones que pueda tener el trabajo.
- Entrega memoria
- Exposición TFG





ANEXO IV: CÓDIGO ARDUINO

```
/*
*****
**      Envío de datos al Puerto Serie      **
*****
// El archivo ha sido modificado partiendo del código RTIMULib-Arduino
//Copyright (c) 2014, richards-tech

// Librerías incluídas para la lectura del sensor inercial

#include <Wire.h>
#include "I2Cdev.h"
#include "RTIMUSettings.h"
#include "RTIMU.h"
#include "RTFusionRTQF.h"
#include "CalLib.h"
#include <EEPROM.h>

RTIMU *imu;                      // el objeto del sensor inercial
RTFusionRTQF fusion;             // el objeto de la fusión de los sensores
RTIMUSettings settings;          // el objeto de ajustes

#define DISPLAY_INTERVAL 50 //<-INTERVALO DE PRESENTACIÓN DE LA INFORMACIÓN [ms]

#define SERIAL_PORT_SPEED 115200 //<-----CAMBIAR VELOCIDAD DEL PUERTO SERIE-----

#define TIME_INTERVAL 1500

//Definición de los tonos del zumbador piezoeléctrico
#define TONO_1 1700
#define TONO_2 1432
#define TONO_3 956
#define TONO_ERROR 600

RTVector3 Vector;

//Variables del sensor inercial
unsigned long lastDisplay;
unsigned long lastRate;
int sampleCount;

//Definición de los pulsadores y leds
int norteLD = 12;    // Salida led conectado en pin 12
int norteSW = 8;     // Entrada pulsador conectado en pin 8
int norte = 3;       // Valor de desviación angular norte (test)

int demoraLD1 = 13;  // Salida led conectado en pin 13
int demoraLD2 = 2;   // Salida led conectado en pin 2
int demoraLD3 = 4;   // Salida led conectado en pin 4
int demoraSW = 7;    // Entrada pulsador conectado a pin 7
int demora = 0;      // Angulo de demora
int nd = 0;          // Numero de demoras capturadas

//Definición del zumbador piezoeléctrico
double zumbador = 6;

//Función principal
void setup()
{
    int errcode;
    Serial.begin(SERIAL_PORT_SPEED);

    pinMode(norteLD, OUTPUT);      // Configuración como salida (led)
```

```
pinMode(norteSW, INPUT);      // Configuración como entrada (pulsador)
digitalWrite(norteSW, HIGH);  // Configuración con pull-up interno

pinMode(demoraLD1, OUTPUT);   // Configuración como salida (led)
pinMode(demoraLD2, OUTPUT);   // Configuración como salida (led)
pinMode(demoraLD3, OUTPUT);   // Configuración como salida (led)
pinMode(demoraSW, INPUT);     // Configuración como entrada (pulsador)
digitalWrite(demoraSW, HIGH);  // Configuración con pull-up interno

pinMode(zumbador, OUTPUT);    // El zumbador como salida

Wire.begin();
imu = RTIMU::createIMU(&settings);          //Crea el objeto IMU

//Comprobación de que se recibe información del sensor inercial
Serial.print("ArduinoIMU starting using device "); Serial.println(imu->IMUName());
if ((errcode = imu->IMUInit()) < 0) {
    Serial.print("Failed to init IMU: "); Serial.println(errcode);
}

//Comprobación de la realización de la calibración previa del magnetómetro
if (imu->getCalibrationValid())
    Serial.println("Using compass calibration");
else
    Serial.println("No valid compass calibration data");

lastDisplay = lastRate = millis();
sampleCount = 0;

// Selección de los sensores que se usen en el algoritmo de fusión
// Cambie entre 'true' y 'false' para la selección de los sensores que se utilicen

fusion.setGyroEnable(true);
fusion.setAccelEnable(true);
fusion.setCompassEnable(true);
}

//Variables asociadas a los pulsadores
int  n = 0;
int  d = 0;

// LECTURA PULSADORES. Para evitar lectura multiple
int d0 = 0, d1 = 0;

//Función repetitiva de cálculo de cambios en el sensor inercial y los pulsadores
void loop()
{
    unsigned long now = millis();
    unsigned long delta;
    if (imu->IMURead()) {
        fusion.newIMUData(imu->getGyro(), imu->getAccel(), imu->getCompass(), imu-
>getTimestamp());
        sampleCount++;
        if ((delta = now - lastRate) >= 1000) {

            sampleCount = 0;
            lastRate = now;
        }
        if ((now - lastDisplay) >= DISPLAY_INTERVAL) {
            lastDisplay = now;

            Vector.setZ(imu->getGyro().z());

            Serial.print("y:");
            RTMath::displayYaw((RTVector3&)fusion.getFusionPose());
            Serial.print("#");
```

```
        Serial.print("g:");
        Serial.print (Vector.z(),4);
        Serial.print("#");

        //BOTONES
        Serial.print("n:");    Serial.print(n);                Serial.print("#");
        Serial.print("d:");    Serial.print(d);                Serial.print("#");

        Serial.println();
    }
}

n = 0; // variable que almacena el boton inicio norte
if (digitalRead(norteSW)==LOW) n = 1; // pasa a valor 1 si se ha pulsado el boton
inicit norte

d1 = 0; // variable que almacena el boton demora
if (digitalRead(demoraSW)==LOW) d1 = 1; // pasa a valor 1 si se ha pulsado el boton
demora

//Funciones para asegurar que se envíe una sola pulsación
if      (d1==1 && d0==0) { d0=1; d=1; }
else if (d1==1 && d0==1) { d0=1; d=0; }
else if (d1==0 && d0==1) { d0=0; d=0; }
else if (d1==0 && d0==0) { d0=0; d=0; }

//Acciones que se toman tras pulsar el boton inicio norte
if (n == 1)
{
    digitalWrite(demoraLD1, LOW); // Se apaga el led D1
    digitalWrite(demoraLD2, LOW); // Se apaga el led D2
    digitalWrite(demoraLD3, LOW); // Se apaga el led D3
    digitalWrite(norteLD, HIGH);  // Se enciende el led N
    sonarTono(TONO_ERROR,TIME_INTERVAL*0.1);
    delay(30);
    digitalWrite(norteLD, LOW);    // Se apaga el led N
    sonarTono(TONO_ERROR,TIME_INTERVAL*0.1);
    nd = 0;
}

//Acciones que se toman tras pulsar el boton demora
if (d == 1)
{
    if (nd == 0)
    {
        digitalWrite(demoraLD1, HIGH);
        digitalWrite(demoraLD2, LOW); // Se apaga el led D2
        digitalWrite(demoraLD3, LOW); // Se apaga el led D3
        sonarTono(TONO_1,TIME_INTERVAL*0.3);
        nd++;
    }
    else if (nd==1)
    {
        digitalWrite(demoraLD2, HIGH);
        sonarTono(TONO_2,TIME_INTERVAL*0.3);
        nd++;
    }
    else if (nd==2)
    {
        digitalWrite(demoraLD3, HIGH);
        sonarTono(TONO_3,TIME_INTERVAL*0.3);
        nd = 0;
    }
}
}

// Función que establece el tono asociado a cada pulsación
void sonarTono(int tono, int duracion)
{
    tone(zumbador,tono,duracion);
    delay(duracion);
}

//-----FIN DEL PROGRAMA-----//
```

ANEXO V: CÓDIGO PROCESSING

```

//*****//
//**                                     TRIANGULACIÓN
**//
//*****//

import processing.serial.*;

//DEFINICIÓN DE ELEMENTOS
int LF = 10; //ASCII line feed -> Salto de línea
int CR = 13; //ASCII RETRONO DE CARRO

// Definicion de variables para pantalla grafica
int L=500, H=500; // TAMAÑO VENTANA
int linewidth = 1; // Grosor linea
int textsize = 8; // Tamaño del texto en el grafico

// CALCULO LOXODROMICA. Precision de calculo para intersección entre loxodromicas
float ERROR = 0.00001;
float error = 0.00001;
float STEP = ERROR/10;
float step = ERROR/10;

// LIMITES GEOGRAFICOS PARA VENTANA GRAFICA (Longitud, Latitud).
float LNG_A=0, LAT_A=0;
float LNG_B=10, LAT_B=10;

//PUNTA 1
float P1_lat = 1.0, a1, b1; // la latitud en grados
String northSouth1; // norte o sur
float P1_long = 1.0, c1, d1; // la longitud en grados
String eastWest1; // leste u oeste
float P1_demora = 30 + 180;

//PUNTA 2
float P2_lat = 1.0, a2, b2; // la latitud en grados
String northSouth2; // norte o sur
float P2_long = -1.0, c2, d2; // la longitud en grados
String eastWest2; // leste u oeste
float P2_demora = 40 + 90;

//PUNTA 3
float P3_lat = -1.0, a3, b3; // la latitud en grados
String northSouth3; // norte o sur
float P3_long = -1.0, c3, d3; // la longitud en grados
String eastWest3; // leste u oeste
float P3_demora = 70;

int contmarcas = 0;

// Variables obtencion de puntos de interseccion de loxodromicas.
float P0_long, P0_lat;
float P_long, P_lat;

float P12_long, P12_lat;
float P13_long, P13_lat;
float P23_long, P23_lat;

float ERR;
float ERR_1;
float ERR_2;
float ERR_3;

// VARIABLES AUXILIARES PARA CONTROL DE PROGRAMA
char ch = 'x';
int pintaloxo = 0;

//Variables lectura de Arduino:
Serial PuertoArduino;
```

```
short portIndexArduino = 2; // <-----puertoCOM arduino-----
-----
String StringArduino;
float Azimuth;
float offset = 0;
int ndemoras = 0;

// variables pulsadores
int pulsanorte = 0;
int pulsademora = 0;

//Variables Open CPN:
Serial PuertoOpenCPN; // puerto serie
short portIndexOpenCPN = 1; // <-----puertoCOM openCPN-----
-----
String StringOpenCPN;
int sensorValue = 0; // valor del sensor

// Variables para la Brújula Inercial
float Psi;
float CompassMagnificationFactor=0.10;
float SpanAngle=120;

//-----FUNCIONES PRINCIPALES-----//
//*****//
void setup()
{
    println("Initializing...");

    // CREACION VENTANA GRAFICA
    size(L,H); // Tamaño de ventana de la aplicación

    //APERTURA DE PUERTOS SERIE
    println("Opening COMs ports...");
    println(Serial.list()); //muestra los puertosCOM disponibles

    //Conecta con Arduino
    String portNameArduino = Serial.list()[portIndexArduino];
    println(" Connecting to Arduino -> " + Serial.list()[portIndexArduino]);
    PuertoArduino = new Serial(this, Serial.list()[portIndexArduino], 115200);
    PuertoArduino.clear();
    PuertoArduino.bufferUntil(LF);

    //Conecta con OpenCPN
    String portNameOpenCPN = Serial.list()[portIndexOpenCPN];
    println(" Connecting to OpenCPN -> " + Serial.list()[portIndexOpenCPN]);
    PuertoOpenCPN = new Serial(this, Serial.list()[portIndexOpenCPN], 4800);
    PuertoOpenCPN.clear();
    PuertoOpenCPN.bufferUntil(CR); // read bytes into a buffer until you get a linefeed
    (ASCII 13):

    smooth(); //Suaviza formas geométricas incluidas
}

//*****//
void draw()
{
    clear(); // Borra pantalla gráfica
    //background(0); // pantalla con fondo negro
    background(150,200,255); // pantalla con fondo azul

    if (keyPressed) ch = key;
    else if (ch == '1')
    {
        EjemploMarcas1();
        AjustarCuadricula();
        PintarCuadricula();
        PintarMarca(" P1", P1_long, P1_lat, LNG_A, LAT_A, LNG_B, LAT_B, 155, 0, 0);
        PintarMarca(" P2", P2_long, P2_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 155, 0);
        PintarMarca(" P3", P3_long, P3_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 0, 155);
    }
    // TrazadoLoxodromicas();
    TrazadoOrtodromicas();
}
else if (ch=='2')
{
```

```

        EjemploMarcas2();
        AjustarCuadricula();
        PintarCuadricula();
        PintarMarca(" P1", P1_long, P1_lat, LNG_A, LAT_A, LNG_B, LAT_B, 155, 0, 0);
        PintarMarca(" P2", P2_long, P2_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 155, 0);
        PintarMarca(" P3", P3_long, P3_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 0, 155);
// TrazadoLoxodromicas();
TrazadoOrtodromicas();
    }
    else if (ch=='3')
    {
        EjemploMarcas3();
        AjustarCuadricula();
        PintarCuadricula();
        PintarMarca(" P1", P1_long, P1_lat, LNG_A, LAT_A, LNG_B, LAT_B, 155, 0, 0);
        PintarMarca(" P2", P2_long, P2_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 155, 0);
        PintarMarca(" P3", P3_long, P3_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 0, 155);
// TrazadoLoxodromicas();
TrazadoOrtodromicas();
    }
    else if (ch=='9')
    {
        EjemploMarcas9();
        AjustarCuadricula();
        PintarCuadricula();
        PintarMarca(" P1", P1_long, P1_lat, LNG_A, LAT_A, LNG_B, LAT_B, 155, 0, 0);
        PintarMarca(" P2", P2_long, P2_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 155, 0);
        PintarMarca(" P3", P3_long, P3_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 0, 155);
// TrazadoLoxodromicas();
TrazadoOrtodromicas();
    }
    else if (ch=='0')
    {
        EjemploMarcas0();
        AjustarCuadricula();
        PintarCuadricula();
        PintarMarca(" P1", P1_long, P1_lat, LNG_A, LAT_A, LNG_B, LAT_B, 155, 0, 0);
        PintarMarca(" P2", P2_long, P2_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 155, 0);
        PintarMarca(" P3", P3_long, P3_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 0, 155);
// TrazadoLoxodromicas();
TrazadoOrtodromicas();
    }
    else
    {
        AjustarCuadricula();
        PintarCuadricula();
        PintarMarca(" P1", P1_long, P1_lat, LNG_A, LAT_A, LNG_B, LAT_B, 155, 0, 0);
        PintarMarca(" P2", P2_long, P2_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 155, 0);
        PintarMarca(" P3", P3_long, P3_lat, LNG_A, LAT_A, LNG_B, LAT_B, 0, 0, 155);
        //println(" P1= " + P1_long + " " + P1_lat + " P2= " + P2_long + " " + P2_lat + " P3= " +
P3_long + " " + P3_lat);
        if ((ndemoras==3) & (contmarcas==3)) {pintaloxo = 1; ; ndemoras=0; contmarcas=0;}
        if (pintaloxo == 1)
        {
// TrazadoLoxodromicas();
TrazadoOrtodromicas();
        }
    }

//DIBUJAR BRÚJULA INERCIAL
Giroscopica();
}

//*****//
void serialEvent(Serial p) {
    StringOpenCPN = PuertoOpenCPN.readStringUntil(LF); //LF
    print(StringOpenCPN);

    if (StringOpenCPN != null)
    {
        ExtraerMarcas(); //Si se recibe info de OpenCPN obtener marcas de acuerdo con función
    }

    StringArduino = PuertoArduino.readString(); //Lectura datos de Arduino
    // println(StringArduino); //Comprobación datos obtenidos

```

```
try {
    String[] dataStrings = split(StringArduino, '#'); //División de la trama recibida
    Psi = float(dataStrings[2].substring(2));
    if (Psi<0){ Azimuth = Psi + 360;} else {Azimuth = Psi;}
    pulsanorte = int(dataStrings[3].substring(2));
    pulsademora = int(dataStrings[4].substring(2));

    ExtraeDemoras();

    print("DEMORA1 " + P1_demora + " ");
    print("DEMORA2 " + P2_demora + " ");
    print("DEMORA3 " + P3_demora + " ");

    print("azimut: ");
    println(Azimuth-offset);

    println(contmarcas+" "+ ndemoras+ " " + key);

    print("MARCA 1: ");
    println(P1_lat + " " + northSouth1 + ", " + P1_long + " " + eastWest1);
    print("MARCA 2: ");
    println(P2_lat + " " + northSouth2 + ", " + P2_long + " " + eastWest2);
    print("MARCA 3: ");
    println(P3_lat + " " + northSouth3 + ", " + P3_long + " " + eastWest3);
}
catch (Exception e) {
    println("Caught Exception");
}
}

////////////////////////////////////
//                                F U N C I O N E S                                //
//
////////////////////////////////////

//                                FUNCIONES DE CAPTURA DE DATOS                                //
//-----//
//FUNCION////////////////////////////////////
void ExtraeDemoras()
{
    if (pulsanorte == 1)
    {
        offset = Psi;
        pulsanorte = 1;
        print( " offset: " + offset);
    }

    if ((pulsademora == 1)&(ndemoras == 0))
    {
        pintaloxo = 0; // variable auxiliar para que pinte la loxodromica cada
pasada
        P1_demora = Psi - offset;
        ndemoras++;
        pulsademora = 0;
        print( "DEMORA1: " + P1_demora);
    }
    if ((pulsademora == 1)&(ndemoras == 1))
    {
        P2_demora = Psi - offset;
        ndemoras++;
        pulsademora = 0;
        print( " DEMORA2: " + P2_demora);
    }
    if ((pulsademora == 1)&(ndemoras == 2))
    {
        P3_demora = Psi - offset;
        ndemoras++;
        pulsademora = 0;
        print(" DEMORA3: " + P3_demora);
        println( " ndemoras: " + ndemoras);
    }
}

//FUNCIÓN////////////////////////////////////
void ExtraerMarcas()
{
```

```
// OBTENCION MARCAS DEL MAPA OPENCNP

String items[] = (split(StringOpenCPN, ','));

if (items[0].equals("$ECWPL") == true)
{
    if (contmarcas == 0)
    {
        a1 = float(items[1].substring(0, 2));
        b1 = float(items[1].substring(2));
        northSouth1 = items[2];
        if (items[2].equals("N") == true) {P1_lat = (a1+(b1/60)); } else {P1_lat = -(a1+(b1/60));}
        c1 = float(items[3].substring(0, 3));
        d1 = float(items[3].substring(3));
        P1_long = (c1+(d1/60));
        eastWest1 = items[4];
        if (items[4].equals("E") == true) { P1_long = (c1+(d1/60)); } else {P1_long = -(c1+(d1/60));}

        print("MARCA 1: ");
        println(P1_lat + " " + northSouth1 + ", " + P1_long + " " + eastWest1);

// Si no se asigna esto aqui, no se autoajusta bien la cuadrícula.
        P12_long = P1_long;    P12_lat = P1_lat;
        P13_long = P1_long;    P13_lat = P1_lat;
        P23_long = P1_long;    P23_lat = P1_lat;

        contmarcas++;
    }
    else if (contmarcas == 1)
    {
        // move the items from the string into the variables:
        a2 = float(items[1].substring(0, 2));
        b2 = float(items[1].substring(2));
        northSouth2 = items[2];
        if (items[2].equals("N") == true) {P2_lat = (a2+(b2/60)); } else {P2_lat = -(a2+(b2/60));}
        c2 = float(items[3].substring(0, 3));
        d2 = float(items[3].substring(3));
        eastWest2 = items[4];
        if (items[4].equals("E") == true) { P2_long = (c2+(d2/60)); } else {P2_long = -(c2+(d2/60));}

        print("MARCA 2: ");
        println(P2_lat + " " + northSouth2 + ", " + P2_long + " " + eastWest2);

        contmarcas++;
    }
    else if (contmarcas == 2)
    {
        // move the items from the string into the variables:
        a3 = float(items[1].substring(0, 2));
        b3 = float(items[1].substring(2));
        northSouth3 = items[2];
        if (items[2].equals("N") == true) {P3_lat = (a3+(b3/60)); } else {P3_lat = -(a3+(b3/60));}
        c3 = float(items[3].substring(0, 3));
        d3 = float(items[3].substring(3));
        eastWest3 = items[4];
        if (items[4].equals("E") == true) { P3_long = (c3+(d3/60)); } else {P3_long = -
(c3+(d3/60));}
        print("MARCA 3: ");
        println(P3_lat + " " + northSouth3 + ", " + P3_long + " " + eastWest3);

        contmarcas=0;
    }
    }
    else
    {
        println("error: la informacion enviada no esta codificada de acuerdo con el protocolo NMEA");
    }
}

//----- FUNCIONES DE CALCULO -----//
//FUNCION//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
float lat_ortodromica(float Po_long, float Po_lat, float P_long, float Po_demora)
{
    float Po_rumbo=0;
    float Alfa=0, Beta=0;
```

```
if (Po_demora <= 180) Po_rumbo = Po_demora + 180;
if (Po_demora > 180) Po_rumbo = Po_demora - 180;

Alfa = Po_long - 180/PI*atan(sin(Po_lat*PI/180)*tan(Po_rumbo*PI/180));
Beta = 180/PI*atan(tan(Po_lat*PI/180)/sin(Po_long*PI/180-Alfa*PI/180));

return 180/PI*atan(sin(P_long*PI/180-Alfa*PI/180)*tan(Beta*PI/180));
}

//FUNCION////////////////////////////////////
float long_loxodromica(float Po_long, float Po_lat, float P_lat, float Po_demora)
{
    float Po_rumbo=0;

    if (Po_demora <= 180) Po_rumbo = Po_demora + 180;
    if (Po_demora > 180) Po_rumbo = Po_demora - 180;

    return Po_long + 180/PI*log(tan(PI/4 - Po_lat/2*PI/180)/tan(PI/4 -
P_lat/2*PI/180))*tan(Po_rumbo*PI/180);
}

//FUNCION////////////////////////////////////
// Trazado de la interseccion entre ortodromicas con origen en Pa y Pb
void TrazarInterseccionOrtodromicas(int linecolorR, int linecolorG, int linecolorB, int circlecolor,
float Pa_long, float Pa_lat, float Pa_demora, float Pb_long, float Pb_lat , float Pb_demora)
{
    // Trazado de la interseccion entre loxodromicas con origen en P1 y P2
    if ((0 <= Pa_demora)&(Pa_demora < 180)) step = -STEP;
    else if ((180 <= Pa_demora)&(Pa_demora < 360)) step = STEP;

    stroke(linecolorR, linecolorG, linecolorB);          // Color linea
    strokeWeight(lineweight);                          // Grosor de linea en pixels

    for (P_long = Pa_long;
        abs(lat_ortodromica(Pa_long, Pa_lat, P_long, Pa_demora)
        - lat_ortodromica(Pb_long, Pb_lat, P_long, Pb_demora)) > ERROR;
        P_long = P_long + step)
    {
        if(keyPressed) ch = key;
        if (ch == ' ') {ch= 'x'; break;}
        P_lat = lat_ortodromica(Pa_long, Pa_lat, P_long, Pa_demora);
        point((P_long-LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A));
    }

    // Se pinta un circulo indicando la intersección
    stroke(circlecolor);          // Color gris
    //stroke(255, 255, 255);      // Color
    strokeWeight(lineweight);     // Grosor de linea en pixels
    noFill();                    // Sin relleno en los circulos
    ellipse((P_long-LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A), 15, 15);
    //print("P12 LONG = " + P_long + " LAT = " + P_lat + " / ");
}

//FUNCION////////////////////////////////////
// Trazado de la interseccion entre loxodromicas con origen en Pa y Pb
void TrazarInterseccionLoxodromicas(int linecolorR, int linecolorG, int linecolorB, int circlecolor,
float Pa_long, float Pa_lat, float Pa_demora, float Pb_long, float Pb_lat , float Pb_demora)
{
    if ((0<=Pa_demora)&(Pa_demora< 90)) step = -STEP;
    else if ((90<=Pa_demora)&(Pa_demora<270)) step = STEP;
    else if ((270<=Pa_demora)&(Pa_demora<360)) step = -STEP;
    stroke(linecolorR, linecolorG, linecolorB);          // Color linea
    strokeWeight(lineweight);                          // Grosor de linea en pixels

    for (P_lat = Pa_lat;
        abs(long_loxodromica(Pa_long, Pa_lat , P_lat, Pa_demora)
        - long_loxodromica(Pb_long, Pb_lat , P_lat, Pb_demora)) > ERROR;
        P_lat = P_lat + step)
    {
        if(keyPressed) ch = key;
        if (ch == ' ') {ch= 'x'; break;}
        P_long = long_loxodromica(Pa_long, Pa_lat , P_lat, Pa_demora);
        point((P_long-LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A));
    }

    // Se pinta un circulo indicando la intersección
    stroke(circlecolor);          // Color gris
    //stroke(255, 255, 255);      // Color
    strokeWeight(lineweight);     // Grosor de linea en pixels
```



```
    TextoLongLat(255, 0, 0, textsize*0, textsize); // Con la variable de entrada textsize se
desplaza el texto verticalmente en la cantidad de pixels indicada
    P13_long = P_long;
    P13_lat = P_lat;

    TrazarInterseccionLoxodromicas( 0, 255, 0, 80, P2_long, P2_lat, P2_demora, P3_long, P3_lat ,
P3_demora);
    TrazarInterseccionLoxodromicas( 0, 0, 255, 80, P3_long, P3_lat, P3_demora, P2_long, P2_lat ,
P2_demora);
    TextoLongLat(255, 0, 0, textsize*2, textsize*3); // Con la variable de entrada textsize se
desplaza el texto verticalmente en la cantidad de pixels indicada
    P23_long = P_long;
    P23_lat = P_lat;

    // Escribe en el grafico longitud y latitud promedio de las tres intersecciones y el error mayor
    P_long = (P12_long + P13_long + P23_long)/3;
    P_lat = (P12_lat + P13_lat + P23_lat)/3;
    ERR_1 = sqrt(sq(P_long - P12_long))+sqrt(sq(P_lat - P12_lat));
    ERR_2 = sqrt(sq(P_long - P13_long))+sqrt(sq(P_lat - P13_lat));
    ERR_3 = sqrt(sq(P_long - P23_long))+sqrt(sq(P_lat - P23_lat));
    ERR = ERR_1;
    if (ERR < ERR_2) ERR = ERR_2;
    else if (ERR < ERR_3) ERR = ERR_3;

    stroke(80); // Color blanco
    //stroke(255, 255, 255); // Color blanco
    strokeWeight(lineWeight); // Stroke weight to 8 pixels
    noFill(); // Sin relleno en los circulos
    ellipse((P_long-LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A), 15, 15);
    //println("Pmed LONG = " + P_long + " LAT = " + P_lat + " / ");

    TextoLongLatErr( 0, 255, 0, textsize*11, textsize*13, textsize*15);
}

//FUNCION////////////////////////////////////
void AjustarCuadricula()
{
    LNG_A = P1_long;
    LAT_A = P1_lat;

    // LONGITUD MINIMA
    if (LNG_A > P2_long) LNG_A = P2_long;
    if (LNG_A > P3_long) LNG_A = P3_long;
    if (LNG_A > P12_long) LNG_A = P12_long;
    if (LNG_A > P13_long) LNG_A = P13_long;
    if (LNG_A > P23_long) LNG_A = P23_long;

    // LATITUD MINIMA
    if (LAT_A > P2_lat) LAT_A = P2_lat;
    if (LAT_A > P3_lat) LAT_A = P3_lat;
    if (LAT_A > P12_lat) LAT_A = P12_lat;
    if (LAT_A > P13_lat) LAT_A = P13_lat;
    if (LAT_A > P23_lat) LAT_A = P23_lat;

    LNG_B = P1_long;
    LAT_B = P1_lat;

    // LONGITUD MAXIMA
    if (LNG_B < P2_long) LNG_B = P2_long;
    if (LNG_B < P3_long) LNG_B = P3_long;
    if (LNG_B < P12_long) LNG_B = P12_long;
    if (LNG_B < P13_long) LNG_B = P13_long;
    if (LNG_B < P23_long) LNG_B = P23_long;

    // LATITUD MAXIMA
    if (LAT_B < P2_lat) LAT_B = P2_lat;
    if (LAT_B < P3_lat) LAT_B = P3_lat;
    if (LAT_B < P12_lat) LAT_B = P12_lat;
    if (LAT_B < P13_lat) LAT_B = P13_lat;
    if (LAT_B < P23_lat) LAT_B = P23_lat;

    float p = LNG_B-LNG_A;
    float q = LAT_B-LAT_A;

    if (p > q)
    {
        LAT_A = LAT_A - p/10 - (p-q)/2;
        LAT_B = LAT_B + p/10 + (p-q)/2;
        LNG_A = LNG_A - p/10;
```

```

    LNG_B = LNG_B + p/10;
}
else if (p < q)
{
    LNG_A = LNG_A - q/10 - (q-p)/2;
    LNG_B = LNG_B + q/10 + (q-p)/2;
    LAT_A = LAT_A - q/10;
    LAT_B = LAT_B + q/10;
}
else
{
    LNG_A = LNG_A - q/10;
    LNG_B = LNG_B + q/10;
    LAT_A = LAT_A - q/10;
    LAT_B = LAT_B + q/10;
}
}

//-----FUNCIONES DE PINTADO-----//
//FUNCION////////////////////////////////////////////////////////////////////
void PintarMarca(String Name, float P_long, float P_lat, float LNG_A, float LAT_A, float LNG_B,
float LAT_B, int R, int G, int B)
{
    // Trazado punto
    stroke(R, G, B);           // Color punto
    textSize(14);
    strokeWeight(8);           // Grosor trazos en pixels
    float x = xgraf(P_long, LNG_A, LNG_B, L);
    float y = ygraf(P_lat, LAT_A, LAT_B, H);
    point(x, y);

    // Escribe texto
    fill(R, G, B);             // Color letras
    text(" " + Name, x, y);
}

//FUNCION////////////////////////////////////////////////////////////////////
float xgraf(float P_long, float LNG_A, float LNG_B, float L)
{
    return (P_long-LNG_A)*L/(LNG_B-LNG_A);
}

//FUNCION////////////////////////////////////////////////////////////////////
float ygraf(float P_lat, float LAT_A, float LAT_B, float H)
{
    return H - (P_lat-LAT_A)*H/(LAT_B-LAT_A);
}

//FUNCIÓN PINTAR CUADRÍCULA////////////////////////////////////////////////////////////////////
void PintarCuadricula()
{
    String pcardinal;

    //PINTADO DE CUADRICULA
    stroke(255);               // Color blanco
    strokeWeight(1);           // Stroke weight to 8 pixels
    textSize(12);
    fill(30);

    // EXTREMOS PANTALLA GRAFICA
    float xa = xgraf(LNG_A, LNG_A, LNG_B, L);
    float ya = ygraf(LAT_A, LAT_A, LAT_B, H);
    float xb = xgraf(LNG_B, LNG_A, LNG_B, L);
    float yb = ygraf(LAT_B, LAT_A, LAT_B, H);

    // LINEAS HORIZONTALES (LATITUD)
    line(xa, ya*3/4, xb, ya*3/4);
    if ((LAT_A+(LAT_B-LAT_A)*1/4) >= 0) pcardinal = "N"; else pcardinal = "S";
    text(floor(abs((LAT_A+(LAT_B-LAT_A)*1/4))) + "° " + (int)((abs(LAT_A+(LAT_B-LAT_A)*1/4)-
floor(abs(LAT_A+(LAT_B-LAT_A)*1/4)))*60) + "' " + pcardinal, xa, ya*3/4);

    line(xa, ya*2/4, xb, ya*2/4);
    if ((LAT_A+(LAT_B-LAT_A)*2/4) >= 0) pcardinal = "N"; else pcardinal = "S";
    text(floor(abs((LAT_A+(LAT_B-LAT_A)*2/4))) + "° " + (int)((abs(LAT_A+(LAT_B-LAT_A)*2/4)-
floor(abs(LAT_A+(LAT_B-LAT_A)*2/4)))*60) + "' " + pcardinal, xa, ya*2/4);

```

```
line(xa, ya*1/4, xb, ya*1/4);
if ((LAT_A+(LAT_B-LAT_A)*3/4) >= 0) pcardinal = "N"; else pcardinal = "S";
text(floor(abs((LAT_A+(LAT_B-LAT_A)*3/4))) + "° " + (int)((abs(LAT_A+(LAT_B-LAT_A)*3/4)-
floor(abs(LAT_A+(LAT_B-LAT_A)*3/4))*60) + "' " + pcardinal, xa, ya*1/4);

// LINEAS VERTICALES (LONGITUD)
line(xb*1/4, ya, xb*1/4, yb);
if ((LAT_A+(LAT_B-LAT_A)*1/4)>=0) pcardinal = "E"; else pcardinal = "W";
text(floor(abs((LNG_A+(LNG_B-LNG_A)*1/4))) + "° " + (int)((abs(LNG_A+(LNG_B-LNG_A)*1/4)-
floor(abs(LNG_A+(LNG_B-LNG_A)*1/4))*60) + "' " + pcardinal, xb*1/4, ya);

line(xb*2/4, ya, xb*2/4, yb);
if ((LAT_A+(LAT_B-LAT_A)*2/4)>=0) pcardinal = "E"; else pcardinal = "W";
text(floor(abs((LNG_A+(LNG_B-LNG_A)*2/4))) + "° " + (int)((abs(LNG_A+(LNG_B-LNG_A)*2/4)-
floor(abs(LNG_A+(LNG_B-LNG_A)*2/4))*60) + "' " + pcardinal, xb*2/4, ya);

line(xb*3/4, ya, xb*3/4, yb);
if ((LAT_A+(LAT_B-LAT_A)*3/4)>=0) pcardinal = "E"; else pcardinal = "W";
text(floor(abs((LNG_A+(LNG_B-LNG_A)*3/4))) + "° " + (int)((abs(LNG_A+(LNG_B-LNG_A)*3/4)-
floor(abs(LNG_A+(LNG_B-LNG_A)*3/4))*60) + "' " + pcardinal, xb*3/4, ya);

// TITULO PANTALLA GRAFICA
stroke(220,220,220); // Color blanco
strokeWeight(3); // Stroke weight to 8 pixels
fill(225,225,225);
rect(150,8,190,15, 2);

textSize(13);
fill(30);
text("TRIANGULACIÓN DEMORAS", 160, 20);

//AUTOR.
textSize(15);
fill(64,64,64);
text("DE MEER", 20, 500);

//Muestra en pantalla la demora a la que se está apuntando con la alidada.

int Azimuth1=round(Azimuth);
textSize(17);
fill(255);
text("AZIMUT: "+Azimuth+" °", 180, 480);
}

//FUNCION////////////////////////////////////
void Giroscopica()
{
  translate(6*L/7,0.8*H/7); //Traslada la giros a la parte inferior derecha.
  scale(CompassMagnificationFactor); //Escala las dimensiones en que se ha configurado la Giros.

  //Circunferencias de la Giros.
  noFill();
  stroke(100);
  strokeWeight(80);
  ellipse(0, 0, 750, 750);

  strokeWeight(50);
  stroke(50);
  fill(0, 0, 40);
  ellipse(0, 0, 610, 610);

  noStroke();
  fill(150,200,255);
  ellipse(0, 0, 510, 510);

  //Puntos Cardinales.
  fill(0);
  textSize(68);

  text("W", -390, -50, 100, 80);
  text("E", 360, -50, 100, 80);
  text("N", 0, -400, 100, 80);
  text("S", 0, 340, 100, 80);
  rotate(PI/4);
  /* textSize(40);
  fill(255);
  text("NW", -370, 0, 100, 50);
  text("SE", 365, 0, 100, 50);
```

```

    text("NE", 0, -355, 100, 50);
    text("SW", 0, 365, 100, 50); */
    rotate(-PI/4);
    Flecha();
}
void Flecha()
{
    //Rotación Flecha
    rotate((PI)+(Azimuth-offset)*(PI/180));

    //Diseño
    stroke(0);
    strokeWeight(4);
    fill(200, 10, 10);
    triangle(-20, -210, 20, -210, 0, 270);
    triangle(-15, 210, 15, 210, 0, 270);
    ellipse(0, 0, 45, 45);
    fill(0, 0, 50);
    noStroke();
    ellipse(0, 0, 10, 10);
    triangle(-20, -213, 20, -213, 0, -190);
    triangle(-15, -215, 15, -215, 0, -200);
}

//FUNCION//////////////////////////////////////
// Escribe en el grafico longitud y latitud de la intersección entre P1 y P2
void TextoLongLat(int linecolorR, int linecolorG, int linecolorB, int txtpos1, int txtpos2)
{
    String pcardinal;

    // Escribe en el grafico longitud y latitud de la intersección entre P1 y P2
    stroke(linecolorR, linecolorG, linecolorB);          // Color linea
    strokeWeight(2); // Stroke weight to 8 pixels
    textSize(textSize);
    //text(" Long " +str(int(P_long))+"" " +str((P_long-int(P_long))*60)+"'", (P_long-
LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A));
    //text(" Lat " +str(int(P_lat))+"" " +str((P_lat -int(P_lat ))*60)+"'", (P_long-
LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A)+textSize);
    if (P_long>=0) pcardinal = "E"; else pcardinal = "W";
    text(" Long " + floor(abs(P_long))+"" " +(int)((abs(P_long)-floor(abs(P_long)))*60)+"' " +
pcardinal, (P_long-LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A)+txtpos1);
    if (P_lat >= 0) pcardinal = "N"; else pcardinal = "S";
    text(" Lat " +floor(abs(P_lat))+"" " +(int)((abs(P_lat) -floor(abs(P_lat )))*60)+"' " +
pcardinal, (P_long-LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A)+txtpos2);
}

//FUNCION//////////////////////////////////////
// Escribe en el grafico longitud y latitud de la intersección entre P1 y P2
void TextoLongLatErr(int linecolorR, int linecolorG, int linecolorB, int txtpos1, int txtpos2, int
txtpos3)
{
    String pcardinal;
    // Escribe en el grafico longitud y latitud
    stroke(linecolorR, linecolorG, linecolorB);          // Color linea
    strokeWeight(2); // Stroke weight to 8 pixels
    textSize(13);
    //text(" Long " +str(int(P_long))+"" " +str((P_long-int(P_long))*60)+"'", (P_long-
LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A));
    //text(" Lat " +str(int(P_lat))+"" " +str((P_lat -int(P_lat ))*60)+"'", (P_long-
LNG_A)*L/(LNG_B-LNG_A), H - (P_lat-LAT_A)*H/(LAT_B-LAT_A)+textSize);
    if (P_long>=0) pcardinal = "E"; else pcardinal = "W";
    text(" Long " + floor(abs(P_long))+"" " +(int)((abs(P_long)-floor(abs(P_long)))*60)+"' " +
pcardinal, (P_long-LNG_A)*L/(LNG_B-LNG_A) - 100, H - (P_lat-LAT_A)*H/(LAT_B-LAT_A)+txtpos1);
    if (P_lat >= 0) pcardinal = "N"; else pcardinal = "S";
    text(" Lat " +floor(abs(P_lat))+"" " +(int)((abs(P_lat) -floor(abs(P_lat )))*60)+"' " +
pcardinal, (P_long-LNG_A)*L/(LNG_B-LNG_A) - 100, H - (P_lat-LAT_A)*H/(LAT_B-LAT_A)+txtpos2);
    //text(" ERR " +str(int(ERR))+"" " +str((((float)int(ERR*100000))/100000-int(ERR))*60)+"'",
(P_long-LNG_A)*L/(LNG_B-LNG_A) - 100, H - (P_lat-LAT_A)*H/(LAT_B-LAT_A)+txtpos3);
    //text(" ERR " +str(int(ERR*60*2000))+"" " YRD", (P_long-LNG_A)*L/(LNG_B-LNG_A) - 100, H -
(P_lat-LAT_A)*H/(LAT_B-LAT_A)+txtpos3);
    text(" ERR " +str(ERR*60*2000)+" YRD", (P_long-LNG_A)*L/(LNG_B-LNG_A) - 100, H - (P_lat-
LAT_A)*H/(LAT_B-LAT_A)+txtpos3);
}

//
//-----
//FUNCION//////////////////////////////////////

```

```
void EjemploMarcas1() // FERROL
{
    P1_long = -8.224178;    P1_lat = 43.472471;    P1_demora = 349;    // R = 349 - 180
    P2_long = -8.232821;    P2_lat = 43.47142;    P2_demora = 304;    // R = 304 - 180
    P3_long = -8.23146;    P3_lat = 43.462658;    P3_demora = 240;    // R = 240 - 180
}
//FUNCION////////////////////////////////////
void EjemploMarcas2() // Corcega en tierra
{
    P1_long = 8.9369;    P1_lat = 42.3834;    P1_demora = 315;    // R = 135 = 315 - 180;
    P2_long = 8.9150;    P2_lat = 42.2145;    P2_demora = 260;    // R = 80 = 260 - 180;
    P3_long = 8.8965;    P3_lat = 42.1044;    P3_demora = 215;    // R = 35 = 215 - 180;
}

//FUNCION////////////////////////////////////
void EjemploMarcas3() // Argentina rio de la Plata
{
    P1_long = -57.5340;    P1_lat = -35.0268;    P1_demora = 275;    // R = 95 = 275 - 180;
    P2_long = -57.1400;    P2_lat = -35.4525;    P2_demora = 204;    // R = 24 = 204 - 180;
    P3_long = -56.2617;    P3_lat = -34.8861;    P3_demora = 71;    // R = 251 = 71 + 180
}

//FUNCION////////////////////////////////////
void EjemploMarcas9() // Test cuatro cuadrantes
{
    P1_long = 40.0000;    P1_lat = 40.0000;    P1_demora = 30;    // R = 210 = 30 + 180;
    P2_long = -40.0000;    P2_lat = 40.0000;    P2_demora = 312;    // R = 132 = 312 - 180;
    P3_long = -40.0000;    P3_lat = -40.0000;    P3_demora = 250;    // R = 70 = 250 - 180;
}

//FUNCION////////////////////////////////////
void EjemploMarcas0() // Test cuatro cuadrantes
{
    P1_long = 60.0000;    P1_lat = 60.0000;    P1_demora = 30;    // R = 210 = 30 + 180;
    P2_long = -60.0000;    P2_lat = 60.0000;    P2_demora = 312;    // R = 132 = 312 - 180;
    P3_long = -60.0000;    P3_lat = -60.0000;    P3_demora = 250;    // R = 70 = 250 - 180;
}
```