Centro Universitario de la Defensa
Escuela Naval Militar

TRABAJO FIN DE GRADO

Módulo de control GPS de un USV empleando microcontroladores Arduino

Grado en Ingeniería Mecánica

ALUMNO: José Rico López

DIRECTORA: Paula Gómez Pérez

CURSO ACADÉMICO: 2015-2016

Universidade Vigo

Centro Universitario de la Defensa
en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Módulo de control GPS de un USV empleando microcontroladores Arduino

Grado en Ingeniería Mecánica

Intensificación en Tecnología Naval

Cuerpo General

Universidade Vigo

RESUMEN

El entorno marítimo está lleno de riquezas y oportunidades: no en vano la gran mayoría del comercio mundial se produce a través del mar. No obstante, el mar es un medio cambiante y en ocasiones hostil, y no está exento de riesgos para la vida humana. Existen tareas que se desarrollan en el mar que ponen en peligro al navegante y que son monótonas, como el remolque de blancos de tiro, la comprobación del desminado de un área o la vigilancia marítima en áreas de conflicto.

El presente trabajo busca desarrollar un módulo de control para un vehículo de superficie no tripulado (*Unmanned Surface Vehicle* o *USV*) empleando un microcontrolador Arduino que pueda ser instalado de forma sencilla en diversas plataformas, como las embarcaciones semirrígidas presentes en los buques de la Armada Española. La misión del módulo es hacer que estas unidades puedan seguir de forma autónoma una ruta previamente introducida mediante *waypoints* GPS, y corrigiendo, de forma simultánea, las desviaciones producidas por viento, corriente y marejada.

PALABRAS CLAVE

USV, Arduino, GPS, módulo de control, multiplataforma

AGRADECIMIENTOS

A mis padres, Miguel y Cristina, por su educación y apoyo incondicional.

A la tutora del trabajo, Doña Paula Gómez Pérez, por su gran implicación y ayuda.

CONTENIDO

1 Introducción y Objetivos 5
1.1 Contextualización 5
1.1.1 Buques civiles no tripulados 6
1.1.2 Buques militares no tripulados 9
1.1.3 Proyectos existentes con Arduino 11
1.2 Justificación del proyecto 12
1.3 Objetivos 12
1.4 Organización de la memoria 13
2 Desarrollo del TFG 15
2.1 La plataforma Arduino 15
2.1.1 Características de la plataforma 15
2.1.2 Partes de la placa Arduino 16
2.1.3 Librerías 17
2.1.4 Requisitos del proyecto 17
2.1.5 Arduino UNO 18
2.1.6 Arduino MEGA 2560 19
2.1.7 Sensores y actuadores 19
2.2 Sistema de Posicionamiento Global (GPS) 22
2.3 Fases de implementación del proyecto 24
2.3.1 Primera fase: robot Arduino 24
2.3.2 Segunda fase: coche de radiocontrol. 25
2.3.3 Tercera fase: lancha 34
3 Resultados, pruebas y validación 39
3.1 Pruebas en tierra 39
3.1.1 Pruebas iniciales y depuración del programa 39
3.1.2 Pruebas finales 42
3.2 Pruebas en medio acuático 43
3.2.1 Adaptación del SW a las características de la lancha de radiocontrol 43
3.2.2 Validación y pruebas 43
4 Conclusiones y líneas futuras 45
4.1 Conclusiones 45
4.2 Líneas futuras 46
5 Referencias 49
Apéndice I: Diagramas de flujo del coche 53

I) Diagrama del programa de navegación 53
II) Diagrama de la dirección 54
Apéndice II: diagrama de flujo de la lancha.....55

Índice de figuras

Figura 1-1 Visión del proyecto MUNIN (fuente: [5])....6
Figura 1-2 Imagen del *Waveglider* (fuente: [6])....8
Figura 1-3 Imagen del *C-Enduro* (fuente: [9])....8
Figura 1-4 *Protector USV* (fuente: [10])....9
Figura 1-5 *Common Unmanned Surface Vessel* (fuente:[11])....10
Figura 1-6 Sistema de rastras de influencia (fuente:[12])....10
Figura 1-7 Misiones posibles del *USV* (fuente:[12])....11
Figura 1-8 Proyecto *Whirlgig* (fuente:[19])....12
Figura 2-1 Partes de la placa (fuente: [22])....16
Figura 2-2 Placa Arduino UNO....18
Figura 2-3 Ejemplos de conexión de periféricos....20
Figura 2-4 *Shield* GPS (fuente: [37])....20
Figura 2-5 Magnetómetro....21
Figura 2-6 Sensor de distancia....21
Figura 2-7 Placa de relés....21
Figura 2-8 Constelación de satélites GPS (fuente: [39])....22
Figura 2-9 Arduino Robot (fuente: [42])....24
Figura 2-10 Coche radiocontrol original (Fuente: [43])....25
Figura 2-11 Coche radiocontrol adaptado....26
Figura 2-12 Placa original....26
Figura 2-13 Sustitución por placa de madera....27
Figura 2-14 Montaje final sobre la placa de madera....27
Figura 2-15 Control motor con relés (fuente: [46])....28
Figura 2-16 placa de relés....28
Figura 2-17 Esquema eléctrico del control de propulsión....28
Figura 2-18 Puente H....29
Figura 2-19 Esquema del control de dirección....30
Figura 2-20 Esquema de montaje de un sensor de distancia....30
Figura 2-21 Montaje inicial con un sensor delante....31
Figura 2-22 Montaje mejorado, con dos sensores delante y uno detrás....31
Figura 2-23 Esquema del *shield* GPS....33
Figura 2-24 Esquema del magnetómetro....33
Figura 2-25 Esquema del montaje completo....34
Figura 2-26 Lancha radiocontrol....35

Figura 2-27 Detalle de los motores 35
Figura 2-28 Tapa donde se encuentra la unidad de control.....36
Figura 2-29 Vista de la batería y cableado de los motores.....36
Figura 2-30 Esquema del control de propulsión 37
Figura 2-31 Instalación del microcontrolador en la lancha.....37
Figura 2-32 Esquema completo de la lancha.....38
Figura 3-1 Ruta seguida en el Patio de Aulas 42
Figura 3-2 Ruta seguida en el lago Castiñeiras 44
Figura 3-3 Extracto del vídeo de la prueba de la lancha 44

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contextualización

"Unmanned systems have the potential, and in some cases the demonstrated ability, to reduce risk to manned forces, to provide the force multiplication necessary to accomplish our missions, to perform tasks which manned vehicles cannot, and to do so in a way that is affordable to the nation."

- *Navy Unmanned Surface Vehicle (USV) Master Plan,* Concurred by RDML V. G. Guillory and MAJGEN T.A. Benes [1].

Los vehículos no tripulados permiten al ser humano acceder a sitios que antes le estaban vetados o donde su vida corría un gran riesgo. Las fuerzas armadas de numerosos países, conscientes de estas capacidades, han invertido en el desarrollo de estos vehículos en todos los ámbitos: aéreo, marítimo y terrestre. En la actualidad, la gran mayoría de los ejércitos cuenta con una serie de vehículos aéreos no tripulados en mayor o menor medida, entre los que se encuentra la Armada Española, que cuenta con la 11ª Escuadrilla de aeronaves formada exclusivamente por drones de reconocimiento y está formando a unidades en el manejo de Sistemas Pilotado a Control Remoto (RPAS) [2]. En el ámbito civil los vehículos aéreos no tripulados gozan de gran éxito y se emplean ampliamente para diversas tareas, que abarcan desde la grabación de imágenes hasta el transporte de cargas ligeras [3]. También se han desarrollado coches capaces de conducir sin ayuda que ya han sido probados en carreteras convencionales.

Sin embargo, el desarrollo de los vehículos marítimos de superficie no tripulados no está tan avanzado como el de sus homólogos. El margen presente para el desarrollo de estos vehículos es por consiguiente mayor, y la oportunidad de destacar en este sector no debería desaprovecharse. Este proyecto busca proporcionar una herramienta sencilla y de bajo coste para dotar de autonomía a diversas plataformas navales.

Existen en la actualidad númerosos proyectos de vehículos de superficie no tripulados, con una gran variedad de plataformas y misiones. Para su estudio se realiza una división de los mismos según su misión, civil o militar.

1.1.1 Buques civiles no tripulados

Un buque no tripulado cuenta con una serie de ventajas que lo hacen de gran utilidad para una gran variedad de actividades económicas:

- Gran ahorro en costes y espacios de habitabilidad.
- Ausencia de riesgo para la vida humana.
- Reducción de accidentes debidos al error humano.
- Gestión eficiente de la navegación, gracias a algoritmos de cálculo que tienen en cuenta factores climatológicos, de eficiencia de la maquinaria o de la gestión del tiempo, entre otros.

Evidentemente, existen desventajas asociadas, como la dificultad de solucionar una avería in situ o el riesgo de suponer una presa fácil para la piratería. Si bien el alcanzar un grado de autonomía completo es difícilmente alcanzable en la actualidad, sí sería posible aumentar notablemente el grado de automatización. Por ejemplo, se podría navegar en convoyes de barcos autónomos protegidos por uno convencional con la dotación suficiente para protegerlos y solucionar incidencias.

Entre los buques estudiados asociados al presente TFG destacan los siguientes:

- Comercio marítimo a gran escala: Proyecto MUNIN.
- Buques de investigación oceanográfica.

EL proyecto MUNIN [4], siglas en inglés de navegación marítima no tripulada mediante inteligencia en redes, se trata de un proyecto de colaboración financiado por la Comisión Europea. El objetivo del proyecto es desarrollar y verificar un proyecto de buque autónomo guiado principalmente por un sistema inteligente de toma de decisiones instalado a bordo y controlado por un operador remoto en una estación costera de vigilancia. Los análisis llevados a cabo dentro del proyecto han mostrado que la implantación de un sistema de navegación autónomo. El Proyecto busca estudiar y analizar los aspectos económicos, técnicos y legales de la implantación de un buque no tripulado. Asimismo, también busca explorar nuevas tecnologías aplicables al comercio marítimo convencional.

Figura 1-1 Visión del proyecto MUNIN (fuente: [5])

Se busca mejorar la situación de la industria del transporte marítimo europeo en las siguientes áreas:

- Reducir costes operativos.
- Reducir el impacto ambiental.
- Atraer personal cualificado.

El proyecto toma como base de desarrollo un buque granelero operando en régimen de fletamiento internacionalmente. Este perfil es idóneo para un buque no tripulado debido a que las necesidades de

carga extra son reducidas y los viajes de este tipo suelen ser largos e ininterrumpidos, a diferencia de los buques portacontenedores, lo que permite embarcar una tripulación para las maniobras de atraque y desatraque y efectuar el tránsito de forma autónoma.

El proyecto engloba una serie de elementos necesarios para permitir una navegación autónoma pero controlada de forma remota por un operador en tierra:

- **Módulo de sensores avanzado:** recopila y procesa datos de los sistemas de navegación existentes (RADAR o AIS) y cámaras de luz visible e infrarroja.
- **Sistema autónomo de navegación:** sigue una ruta predeterminada con un cierto grado de libertad que le permite ajustar la ruta de acuerdo con la legislación y con la situación que encuentre, como un riesgo de abordaje o condiciones meteorológicas adversas.
- **Sistema de monitorización y control autónomo del motor:** permite prevenir fallos en el mismo y obtener la máxima eficiencia energética.
- **Centro de control en tierra:** monitoriza y controla constantemente al buque en tránsito. Incluye puestos como un operador encargado de supervisar varios buques a la vez; un ingeniero que asista al operador en cuestiones técnicas y se encargue de los planes de mantenimiento programado; y un equipo de navegación que se encuentre preparado para tomar el control del buque en circunstancias que así lo requieran a través de un sistema remoto que replique el puente de mando del buque.

Los estudios económicos muestran, que en las circunstancias de estudio propuestas, el ahorro económico derivado del empleo de un buque autónomo durante un período de 25 años asciende a 7 millones de dólares.

En el aspecto de la seguridad, los estudios muestran que prácticamente la mitad de las pérdidas en el comercio marítimo son debidas a abordajes y hundimientos. Los análisis llevados a cabo para el proyecto muestran que un buque autónomo permite reducir a la décima parte el riesgo de colisión y abordaje, dado que permite minimizar el error humano debido a la fatiga. Asimismo, el riesgo de fallos del sistema también se ve reducido gracias a la monitorización constante. En cuanto a los incendios se reducen notablemente debido a la ausencia de actividad humana que los pueda causar, y para su extinción se pueden emplear métodos de extinción que son eficaces pero peligrosos para la vida humana, como el uso de instalaciones fijas de anhídrido carbónico.

El proyecto MUNIN se subdivide en diversas subactividades complementarias, etiquetadas con un código. Estas actividades incluyen el desarrollo de la infraestructura, los sistemas y procedimientos empleados en la propulsión y maquinaria, el análisis de costes y beneficios o los aspectos legales, entre otras.

También se encuentra muy extendido el uso de vehículos de superficie no tripulados para recopilar datos oceanográficos, dado que es una tarea repetitiva y que no requiere de mucha supervisión. Existen varios modelos de *USV* en el mercado orientados a este fin.

Uno de los modelos encontrados es el *Wave Glider* [6], producido por la empresa *Liquid Robotics* [7]. Este vehículo está formado por dos partes: una parte superficial que contiene sensores y transmisores y unas placas solares, y una parte sumergida formada por aletas que emplea el movimiento de las olas para propulsar el buque hacia delante. Los sensores situados en la parte flotante permiten medir datos oceanográficos tales como salinidad, temperatura del agua, condiciones meteorológicas, fluorescencia del agua y oxígeno disuelto, además de un compás y un sensor GPS para guiar la navegación.

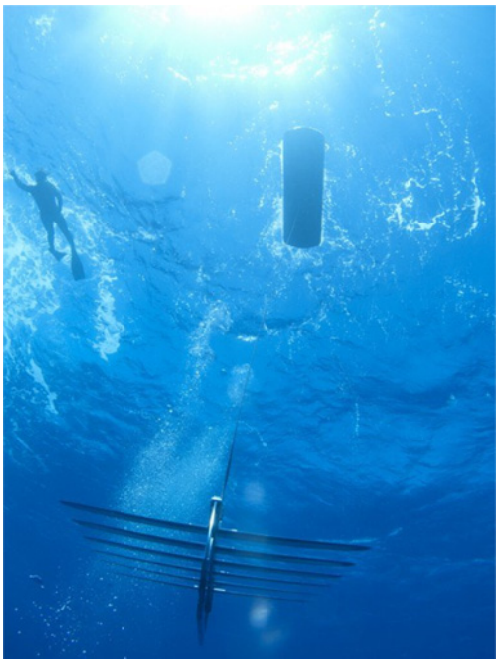


Figura 1-2 Imagen del *Waveglider* (fuente: [6])

Estos *USV* batieron en noviembre de 2012 el récord de distancia recorrida por un vehículo no tripulado [8], completando las 9000 millas náuticas que separan la ciudad estadounidense de San Francisco de la australiana Bundaberg tras superar las duras condiciones meteorológicas, recopilando una gran variedad de datos oceanográficos durante el recorrido.

Otro ejemplo de buque oceanográfico no tripulado es el *C-Enduro* [9], un catamarán con propulsión solar, eólica y diésel, que le confieren una permanencia de hasta tres meses. Está fabricado por la empresa *ASV*, que cuenta en su catálogo con diversos vehículos no tripulados orientados a diversos fines, como blancos de tiro de superficie, buques de rastreo contra minas u otros buques destinados a la recopilación de datos oceanográficos.

Figura 1-3 Imagen del *C-Enduro* (fuente: [9])

1.1.2 Buques militares no tripulados

Los *USV* presentan innegables virtudes para su uso en el ámbito militar, y cada vez más ejércitos cuentan con vehículos de superficie en su haber. Entre los vehículos estudiados se encuentran los drones de ataque y reconocimiento de los ejércitos israelí y estadounidense respectivamente así como los proyectos en los que la Armada Española ha mostrado su interés.

- **Drones militares de ataque:** *Protector USV* [10].
- **Drones militares de reconocimiento:** CUSV [11].
- **Drones en la Armada Española:** proyecto SIRAMICOR[12].

El *Protector USV* [10] es un vehículo de combate naval desarrollado y producido por la empresa israelí *Rafael Advanced Defense Systems*. Se trata de un vehícuo controlado remotamente con un alto grado de independencia capaz de llevar a cabo diversas misiones que incluyen la seguridad marítima, la defensa contra amenaza asimétrica (*Force protection*), misiones antiterrorista o medidas contra minas, entre otras. En la actualidad se encuentra en operación en marinas de diversos países además de Israel.

La versión más reciente del *Protector*, presentada en 2012, dispone de armamento letal y no letal, que pueden incluir ametralladoras, lanzagranadas o un lanzador de misiles *Spyke* en el primer caso y un cañón de agua en el segundo.

El *Protector* cuenta asimismo con diversos sensores, como un Sistema de imagen electro-óptico, RADAR o un medidor de distancia LASER. Cuenta asimismo con varios sistemas de comunicaciones y de mando y control que permiten al operario controlar tanto el gobierno del USV como el de los diversos sistemas de armas que monta.

Figura 1-4 *Protector USV* (fuente: [10])

El *CUSV* [11] es un *USV* producido por la empresa americana *Textron Systems* diseñado para ser desplegado desde los buques de combate litoral de la Marina de los Estados Unidos para cumplir una serie de misiones que incluyen guerra contra minas, guerra antisubmarina, relé de comunicaciones, inteligencia, vigilancia y reconocimiento, guerra de superficie y lanzamiento y recuperación de vehículos aéreos no tripulados.

El CUSV ha sido probado con éxito y en la actualidad acumula cerca de 2000 horas de operación. Este vehículo cuenta con una arquitectura modular y es autoadrizante, y dispone de un sistema de lanzamiento y recuperación autónoma desde los buques de combate litoral (*LCS*).

Dispone de una gran capacidad de carga, que le permite albergar módulos de inteligencia, vigilancia y reconocimiento, neutralización de minas, sonar de rastreo lateral o armas no letales. También incluye comunicaciones por satélite y un sistema de navegación autónomo. Un operador se encarga de controlar al buque a través de un sistema común de mando y control, que cumple con estándares OTAN. La velocidad máxima del *CUSV* es de 28 nudos, y tiene una autonomía que puede alcanzar las 24 horas.

Figura 1-5 ***Common Unmanned Surface Vessel*** **(fuente:[11])**

En el caso de España, el Programa SIRAMICOR (SIstema de Rastreo de MInas de Influencia por COntrol Remoto) es el principal campo de aplicación de un *USV* [12]. El programa busca desarrollar un conjunto de vehículos no tripulados capaces de arrastrar rastras de influencia capaces de hacer explotar minas a su paso sin suponer riesgo para la vida humana. Se busca complementar las capacidades de los cazaminas de la Armada clase *"Segura"* en aquellas circunstancias en las que no se pueda realizar caza de minas. El SIRAMICOR engloba tres subsistemas: Rastras, Drone y Sistemas de Control, Navegación y Comunicaciones (CNC).

Las rastras consisten en un sistema de multi-influencia, que emulan un determinado campo eléctrico, magnético y acústico capaz de activar las minas.

Drone: ha de ser lo suficientemente grande y potente para remolcar las rastras eficazmente y debe tener una autonomía que le permita llevar a cabo las misiones de desminado con garantías. Entre las diversas plataformas estudiadas se acabó optando por un buque monocasco

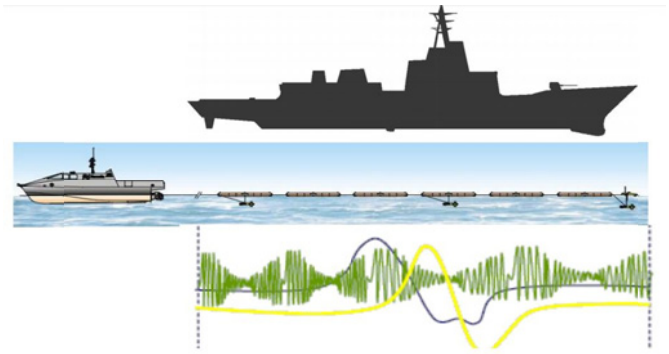


Figura 1-6 Sistema de rastras de influencia (fuente:[12])

Sistemas CNC: estos sistemas han de cumplir varios requisitos. En primer lugar, los drones han de ser capaces de funcionar de forma autónoma en todas las fases, que incluyen tránsito, rastreo y retorno, además de dirigirse a una zona de recuperación preestablecida en caso de fallo o emergencia. Asimismo, será posible controlarlo de forma remota y local. Debe contar con una serie de algoritmos que le permitan detectar obstáculos y evitarlos, gestionar los recursos propios y efectuar una navegación que asegure la efectividad del rastreo, teniendo en cuenta que al maniobrar con la rastra es posible que ésta pierda tensión y no funcione adecuadamente o se enrede.

Figura 1-7 Misiones posibles del *USV* (fuente:[12])

El SIRAMICOR es un proyecto ambicioso y que, llevado a la práctica, dotaría a la Armada de nuevas capacidades y le permitiría llevar a cabo las existentes con mayor eficacia y autonomía. Sin embargo, el déficit presupuestario de Defensa ha llevado a su paralización [13].

1.1.3 Proyectos existentes con Arduino

Existen varios proyectos que emplean la plataforma Arduino asociada a un sensor GPS, muchos de ellos disponibles en la red. Estos han sido de gran ayuda para la realización de este trabajo, tanto en la fase terrestre como en la fase acuática. Buena parte de los proyectos encontrados se basan, al igual que se ha hecho en el presente, en adaptar un modelo de coche radiocontrol para controlarlo mediante Arduino. Dos de los proyectos que se han tomado como inspiración para este TFG se pueden encontrar en [14] y [15].

En cuanto a los proyectos de lanchas controladas por Arduino se han tomado como inspiración otros dos de entre los muchos disponibles. Por un lado se encuentra el proyecto *Whirlgig* [16], financiado por la Universidad de Oslo y construido para levantar cartas batimétricas de lagos en Noruega (Figura 1-8). La misión del programa consiste en navegar siguiendo una ruta predeterminada tomando datos de profundidad cada aproximadamente 10 segundos. La segunda versión de este proyecto [17] se probó con éxito en un lago, recorriendo la considerable distancia de 18 Km.

Por otro lado se encuentra el proyecto mostrado en [18], que consiste en la adaptación de una lancha radiocontrol para seguir una ruta GPS empleando Arduino.

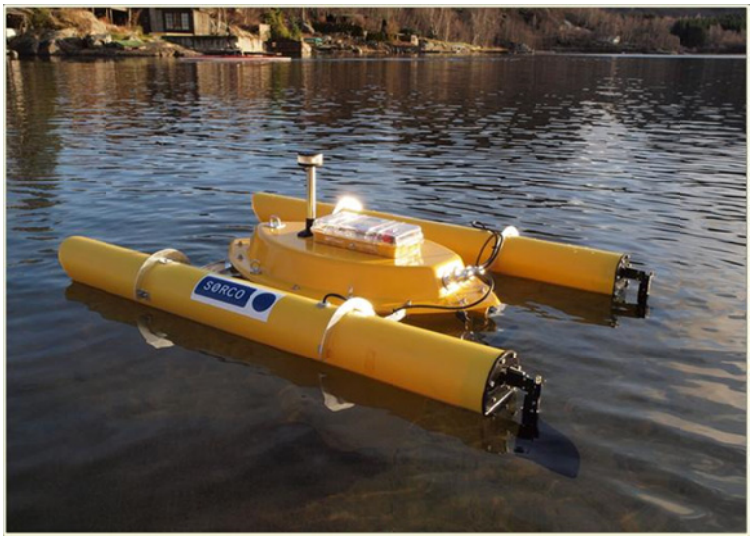


Figura 1-8 Proyecto *Whirlgig* (fuente:[19])

1.2 Justificación del proyecto

Por un lado, existen vehículos militares no tripulados y con grandes capacidades como los expuestos anteriormente, con un coste que en ocasiones excede el millón de dólares. Por otro, se pueden encontrar proyectos de bajo coste empleando plataformas como Arduino o similares destinados a fines lúdicos o civiles. La novedad de este proyecto reside en el enfoque militar del mismo, además de su polivalencia y bajo coste. El módulo desarrollado en este proyecto busca poder ser instalado de forma sencilla en las diversas plataformas con las que cuenta la Armada Española, como las embarcaciones semirrígidas que se encuentran a bordo de todos los buques. Contar con una plataforma autónoma de estas características permitiría llevar a cabo parte de las misiones definidas para el proyecto SIRAMICOR con unas necesidades presupuestarias notablemente inferiores.

1.3 Objetivos

El primer paso para empezar el proyecto es definir claramente los objetivos, que son los siguientes:

- Seleccionar adecuadamente el *hardware* de control que se utilizará en este proyecto.
- Adaptar el *hardware* del módulo de control para ser instalado de forma sencilla en una plataforma de superficie (terrestre y marítima).
- Programar el módulo de control para seguir una ruta previamente introducida.
- Documentar detalladamente el proceso para permitir mejoras y nuevas funcionalidades.

Para lograr alcanzar los objetivos deseados, el trabajo se ha desgranado en una serie de tareas:

- Familiarización con el entorno Arduino. Programación Arduino Robot.
- Documentación sobre proyectos de Arduino con módulo GPS.
- Documentación sobre proyectos de Arduino en barcos.
- Adaptación de un coche radiocontrolado para pruebas en tierra del módulo.
- Construcción del módulo de control.
- Instalación del módulo de control en el coche radiocontrolado.
- Desarrollo de software para evitar obstáculos.

- Desarrollo de software para seguir una ruta preintroducida.
- Integración software.
- Instalación del módulo de control en un barco.
- Prueba final del módulo de control a bordo.
- Documentación gráfica del proceso.
- Redacción de la memoria.

1.4 Organización de la memoria

El desarrollo de la memoria (Capítulo 2) comienza introduciendo al lector el funcionamiento de los microcontroladores Arduino (Sección 2.1) y el sistema GPS (Sección 2.2). Posteriormente se detallan las fases de implementación del proyecto (Sección 2.3): el uso del robot Arduino, del coche radiocontrol y de la lancha.

Las pruebas realizadas en el proyecto se desarrollan en el Capítulo 3, tanto las llevadas a cabo en tierra como en medio acuático.

Finalmente, en el Capítulo 4 se recogen las conclusiones extraídas del proyecto y sus posibilidades de expansión.

Finalmente, la Sección 0 agrupa la bibliografía empleada, y en los Apéndices I y II se encuentran los diagramas de flujo correspondientes al código del coche radiocontrol y de la lancha, respectivamente.

2 DESARROLLO DEL TFG

2.1 La plataforma Arduino

Para la realización de este proyecto se ha seleccionado la plataforma Arduino como módulo de control. Este apartado detallará las características más básicas de dicho hardware.

2.1.1 Características de la plataforma

El sistema Arduino es una plataforma de código abierto para realizar prototipos que engloba tres elementos principales [20]:

- ***Hardware:*** las placas electrónicas Arduino incorporan un microcontrolador y puertos de entrada y salida. Se trata de hardware libre, lo que permite realizar infinidad de proyectos diferentes.
- **Entorno de programación:** simplifica enormemente la creación de código en comparación con anteriores lenguajes. Permite además programar desde diferentes sistemas operativos, como Windows, Mac o Linux.
- **Comunidad:** la comunidad Arduino pone al alcance del público sus proyectos y programas.

El primer Arduino fue introducido en 2005 y tenía como objetivo facilitar la creación de dispositivos capaces de interactuar con su entorno a través de actuadores y sensores de forma sencilla y económica. En la actualidad existen una gran cantidad de proyectos basados en Arduino, que abarcan desde aplicaciones de domótica a impresoras 3D.

El Proyecto Arduino se basa en placas que incluyen diversos microcontroladores. Estas placas cuentan con una serie de entradas y salidas analógicas y digitales que permiten su conexión a otros dispositivos. En la actualidad las placas Arduino se programan mediante USB, por lo que son accesibles al público general. El entorno de programación se conoce como *IDE,* siglas en inglés de entorno de desarrollo integrado, y es gratuito y accesible. Está basado en el proyecto de *software* libre *Processing* [21], que soporta los lenguajes de programación C y C++.

Las carácerísticas de la plataforma Arduino la hacen idónea para el presente proyecto:

- **Software sencillo:** la interfaz es accesible para programadores inexpertos y es lo suficientemente potente para un uso más avanzado.

- **Bajo coste:** los microcontroladores y sensores tienen un precio comparativamente menor al de otros dispositivos similares, y existe un amplio mercado de dispositivos compatibles a un precio notablemente inferior al de los dispositivos de la propia marca.
- **Programas de código abierto:** la comunidad Arduino es muy activa y existen una gran cantidad de proyectos y librerías al alcance del público.
- **Variedad de periféricos:** el Arduino es compatible con actuadores y sensores de muy diversos tipos, y se pueden encontrar en el mercado con facilidad y a precios accesibles. Es posible incluso crear una placa en una *protoboard* a partir del microcontrolador y componentes sueltos y programarla igual que una placa comercial.
- **Tamaño reducido:** la placa UNO cabe en la palma de una mano, por lo que resulta sencillo instalarla tanto en el coche como en la lancha.

Existen en el mercado otros microprocesadores, como la *Raspberry PI*, que también tienen un precio asequible y son adecuados para el proyecto, y cuentan con mayores capacidades de almacenamiento y procesamiento. No obstante, se ha elegido el Arduino por su mayor compatibilidad con periféricos, por su sencillez de programación y porque cuenta con una capacidad más que suficiente para el presente trabajo.

2.1.2 Partes de la placa Arduino

En el mercado se pueden encontrar diversos formatos de placas Arduino, cada una con unas capacidades y conexinones diferentes. Sin embargo, muchas de ellas comparten los siguientes componentes, mostrados en la siguiente figura:

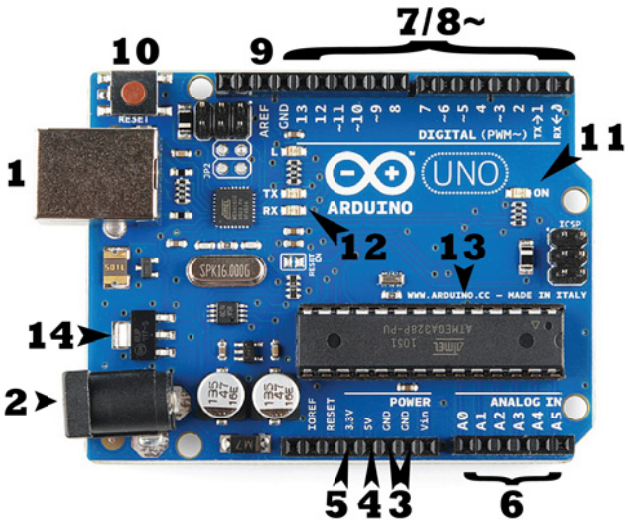


Figura 2-1 Partes de la placa (fuente: [22])

1. **Entrada USB:** permite la conexión con el ordenador para cargar programas y también proporciona la tensión necesaria para el funcionamiento de la placa.
2. **Entrada de alimentación:** Sirve para alimentar la placa, generalmente con una tensión de entre 6 y 12 voltios.
3. **Toma de tierra.**
4. **Salida de 5 voltios:** proporciona la tensión a los diversos sensores y actuadores que se conecten a la placa.

5. **Salida de 3,3 voltios:** funciona igual que la anterior, para los periféricos que funcionen con este voltaje.
6. **Entradas analógicas:** son capaces de reconocer diferentes voltajes y devolver un valor digital mediante el convertidor analógico-digital integrado en la placa (explicado en el apartado 2.3.2.2). Pueden ser usados también como entradas o salidas digitales.
7. **Pines digitales:** pueden funcionar como entradas o salidas, reconociendo o devolviendo un valor 0 o 1, correspondientes a una tensión de 0 o 5 voltios respectivamente.
8. **Pines *PWM*** [23]: pines digitales que permiten regular el voltaje de salida mediante pulsos, como se explica más adelante.
9. ***AREF*** [24]: Permite introducir el voltaje usado como referencia para el convertidor analógico-digital de la placa.
10. **Botón de reset:** permite reiniciar el programa cargado en la placa.
11. **Led de encendido:** se activa al alimentar la placa.
12. **Leds *TX* y *RX:*** se encienden cuando la placa recibe o transmite datos por comunicación serie.
13. **Microcontrolador:** circuito integrado principal, es el "cerebro" de la placa. Varía según la placa de la que se trate.
14. **Regulador de tensión:** reduce el voltaje de entrada hasta los 5 voltios con los que trabaja la placa. Su entrada está limitada a 20 voltios, pero es aconsejable no exceder los 12 voltios.

2.1.3 Librerías

Uno de los componentes clave del entorno Arduino son las librerías [25], que incluyen código de uso común, por lo que simplifican notablemente los programas. Estas presentan diversas funcionalidades: hay librerías que facilitan cálculos, como las librerías estadísticas, hay librerías que permiten la comunicación con otros dispositivos, y hay librerías específicas para diferentes periféricos, entre otras. El uso de una librería para un programa es sencillo: basta con declarar la librería al inicio del programa y emplear las funciones específicas que tiene en el mismo.

El programa de Arduino tiene una serie de librerías preinstaladas, pero el verdadero activo lo forma la comunidad virtual, que sube de forma desinteresada librerías a la red de todo tipo.

En el presente proyecto se han empleado las siguientes librerías:

- ***Adafruit GPS:*** librería propia del módulo GPS elegido.
- ***Software Serial:*** permite la transmisión de datos entre placa y ordenador.
- ***Math:*** librería con funciones matemáticas.
- ***Wire:*** permite comunicación *I2C*, que emplea únicamente dos cables para transmitir datos.
- ***Adafruit Sensor:*** librería común para comunicación con los sensores de la casa *Adafruit.*
- ***Adafruit HMC5883:*** librería del sensor magnético.
- ***NewPing:*** librería del sensor ultrasónico de distancia.
- ***QuickStats:*** librería con funciones estadísticas.

2.1.4 Requisitos del proyecto

Para elegir la placa más adecuada para el proyecto se establecen los siguientes requisitos:

- Posibilidad de conectar un sensor GPS.
- Disponer de 4 pines digitales para tracción y dirección.
- Disponer de 2 pines *I2C* para el magnetómetro.
- Disponer de 2 pines *soft serial* para el sensor GPS.

- Disponer de 1 pin analógico para el potenciómetro de dirección.
- Disponer de Pines extra para futuras mejoras.
- Disponer de memoria suficiente para almacenar el programa.

2.1.5 Arduino UNO

La placa Arduino UNO [26] engloba un microcontrolador ATmega328P [27] y diversas entradas y salidas. Tiene una memoria *flash* de 32 KB donde almacena los programas; una memoria *SRAM* de 2 KB que almacena las variables durante la ejecución del programa; y una EEPROM de 1 KB, empleada para almacenar información a largo plazo [28]. Tiene una velocidad de funcionamiento de 16 MHz.

Cuenta asimismo con 14 entradas o salidas digitales, de las cuales 6 se pueden utilizar como salidas *PWM* [23] que regulan el voltaje de salida mediante pulsos. Tiene también 6 entradas analógicas, entrada USB para comunicación con el ordenador, conector ICSP para programarlo una vez instalado y un botón para reiniciar.

La placa funciona a 5 voltios. La alimentación puede provenir de la entrada USB o de una entrada Jack hembra con un regulador de tensión incorporado.

Figura 2-2 Placa Arduino UNO

La placa cumple los requisitos del proyecto en los siguientes aspectos:

Dispone de 14 pines digitales de entrada o salida. Los pines digitales cuando actúan como entrada devuelven un valor 0 o 1 dependiendo de si reciben o no voltaje. Cuando actúan como salida dan una salida de 5 voltios en caso de activarse o no dan salida en caso contrario. Además, 6 de ellos pueden emplearse para regular la tensión de salida de forma digital mediante control por pulsos (*PWM* [23]). Esto es útil para regular la velocidad de un motor o la intensidad lumínica de un LED, entre otras aplicaciones.

Dispone de 6 entradas analógicas. Las entradas analógicas disponen de un conversor analógico/digital [29] que permite obtener un valor numérico entre 0 y 1023 en función del voltaje de entrada, que ha de estar comprendido entre 0 y 5 voltios. Esto es útil para leer potenciómetros.

Permite comunicación por *I2C* [30], un protocolo de transmisión de datos que permite conectar una gran cantidad de dispositivos (128 en el caso del UNO) empleando únicamente dos cables: *SDA* para datos y *SCL* para el reloj de sincronización. En el proyecto el magnetómetro empleado emplea este protocolo para transmitir sus datos.

Permite comunicación *soft serial* [31], empleada para recibir datos del GPS. Este tipo de comunicación funciona enviando la información bit a bit, y es la que permite también la conexión de la placa con el ordenador.

2.1.6 Arduino MEGA 2560

La placa Arduino MEGA 2560 [32] está basada en el microcontrolador ATmega2560 [33]. Tiene una mayor memoria y número de entradas y salidas que el UNO: dispone de 54 salidas/entradas digitales de las cuales 15 soportan *PWM*, 16 entradas analógicas y 4 *UART* o puertos de comunicación *software serial*. Tiene una memoria *flash* de 256 KB, *SRAM* de 8 KB y *EEPROM* de 4 KB, y una frecuencia de funcionamiento de 16 MHz. Al igual que el Uno, cuenta con entradas *USB*, alimentación, *ICSP* y botón de reset.

El Arduino MEGA 2560 tiene unas capacidades notablemente superiores a las del UNO, pero tiene una característica que motiva descartarlo para el proyecto: no permite comunicación *soft serial* en los pines 7 y 8, que son los empleados por el módulo GPS. Es por ello que se descarta esta placa en favor de la primera.

La siguiente tabla muestra una comparativa entre el Arduino UNO y el MEGA. El MEGA tiene mucha más memoria y entradas que el primero, pero se descarta debido a que es incompatible con el módulo GPS.

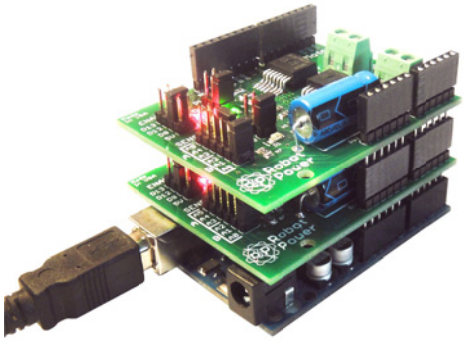
	UNO	**MEGA**
Microprocesador ATmega	328P	2560
Memoria SRAM	2 KB	8 KB
Memoria Flash	32 KB	256 KB
Memoria EEPROM	1 KB	4 KB
Número de pines digitales	14	54
Número de entradas analógicas	6	16
Admite Shield GPS	Sí	No
Compatible sensores *I2C*	Sí	Sí

Tabla 2-1 Comparativa entre tarjetas Arduino

2.1.7 Sensores y actuadores

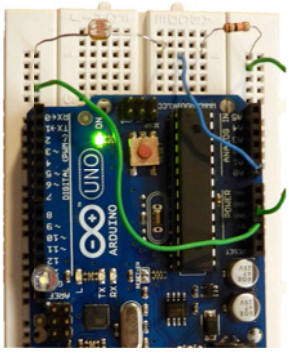
La placa Arduino permite conectar una gran variedad de dispositivos. Existen numerosos sensores producidos por diversas empresas entre los que se incluyen receptores GPS, giróscopos, magnetómetros, acelerómetros, sensores de ultrasonidos, de humedad, temperatura, fotómetros, receptores RFID, LAN y *WiFi*, entre otros. Los periféricos de Arduino se pueden conectar de tres formas:

- ***Shield:*** se conecta directamente encima de la placa. Es fácil de instalar y permite fácil acceso a los pines no empleados por el mismo, por lo que permite apilar más de uno.
- ***Breakout*** [34]*:* incluye el componente deseado y generalmente un circuito integrado para hacer la conexión con el Arduino más sencilla. Tienen un tamaño reducido, son fáciles de conectar y suelen ser más económicos que los shield.
- ***De forma directa:*** también es posible conectar al Arduino periféricos a través de circuitos creados por el usuario en una *protoboard* o similar. El precio suele ser notablemente inferior al de los montajes anteriores, pero es más complejo de montar y ocupa un mayor espacio.



(a) Formato *Shield* (fuente: [35])

(b) Formato *Breakout*



(c) Montaje en protoboard (fuente: [36])

Figura 2-3 Ejemplos de conexión de periféricos

El presente proyecto emplea los siguientes periféricos:

- **Receptor GPS:** Sensor que proporciona la posición GPS del módulo. Estos sensores incluyen un módulo GPS comercial y la circuitería necesaria para conectarlo al Arduino. Se ha escogido un sensor en formato *shield* de la casa Adafruit dado que permite integrarlo en la placa Arduino directamente, sin necesidad de cableado adicional. Concretamente, el modelo seleccionado ha sido el *Adafruit Ultimate GPS Logger Shield* [37], que permite utilizar una serie de librerías existentes, que simplifican los cálculos necesarios para obtener datos GPS. La Figura 2-4 muestra el modelo empleado en este TFG:

Figura 2-4 *Shield* GPS (fuente: [37])

- **Magnetómetro:** Mide la dirección del campo magnético terrestre, de forma análoga a la aguja magnética de un buque. Nos permite conocer la dirección en la que apunta el USV y de esta forma dirigirlo hasta el siguiente waypoint.

Figura 2-5 Magnetómetro

- **Sensor de distancia ultrasónico:** Sensor que emplea pulsos de ultrasonidos para conocer la distancia a un obstáculo, lo que permite esquivarlos.

Figura 2-6 Sensor de distancia

- **Placa de relés:** los relés son interruptores controlados eléctricamente que permite abrir o cerrar un circuito con voltajes o intensidades relativamente elevadas empleando una tensión notablemente inferior. La placa de relés incluye el relé propiamente dicho y el circuito necesario para controlarlo empleando el controlador Arduino.

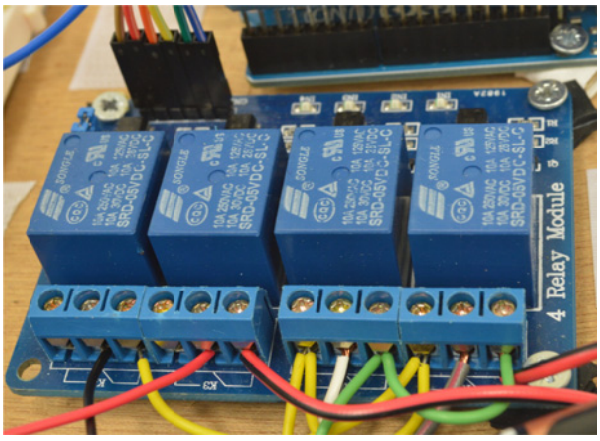


Figura 2-7 Placa de relés

2.2 Sistema de Posicionamiento Global (GPS)

La segunda piedra angular de este trabajo es el control del USV mediante GPS. Es por esto que esta sección describe, de forma sucinta, los aspectos más destacables de este sistema.

El sistema de posicionamiento global, o GPS por sus siglas en inglés, es un sistema de navegación por satélite basado en una constelación de 24 satélites puestos en órbita por el Departamento de Defensa de los Estados Unidos. El GPS fue creado originalmente para aplicaciones militares, aunque en la década de 1980 se hizo disponible para el público general.

El GPS está formado por tres partes o segmentos: Espacial, de Control y Usuario [38]:

- **Segmento Espacial:** formado por una constelación de 24 satélites operativos y 3 más de reserva, que recorren 6 órbitas elípticas inclinadas 55º grados con respecto al Ecuador y 60º entre sí, a una altura media de 20182 kilómetros, de forma que en cualquier lugar y momento sea posible tener al menos 4 satélites a la vista. Los satélites cuentan con un reloj atómico en su interior con una elevada precisión, y emiten constantemente unos códigos pseudoaleatorios que permiten conocer la distancia del satélite al receptor.

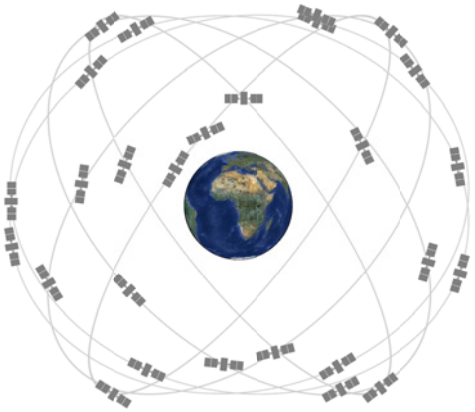


Figura 2-8 Constelación de satélites GPS (fuente: [39])

- **Segmento de Control:** comprende una estación principal de control y una serie de estaciones monitoras que se encargan de realizar el seguimiento de los satélites. Tras procesar los datos de seguimiento en la estación principal se vuelven a enviar los datos a los satélites.
- **Segmento Usuario:** incluye el equipo receptor y la antena. Para obtener una situación es preciso que la antena se encuentre a la vista de los satélites GPS. Aunque es posible obtener una situación con tres satélites, los relojes situados en el receptor son mucho menos precisos que los relojes atómicos que se encuentran en los satélites y por tanto sería una situación con un error notable. Este error se corrige mediante un cuarto satélite, de forma que también es posible conocer la altura. Un mayor número de satélites permite así obtener una situación aún más precisa.

Los factores que influyen en el error del GPS incluyen los siguientes [40]:

- **Errores de propagación de las ondas:** en la ionosfera y la troposfera la señal del satélite se ralentiza. El GPS calcula el retraso medio para corregir parcialmente este error
- **Multitrayecto:** este fénómeno se produce cuando la señan GPS rebota en objetos como edificios o el propio suelo antes de llegar al receptor, lo que aumenta la distancia recorrida por la señal.
- **Errores en el reloj del receptor:** la precisión del mismo es inferior a la de los relojes atómicos situados en los satélites, por lo que hay errores en la medida del tiempo.

- Errores de órbita o de efemérides: la posición transmitida por el satélite presenta ciertas imprecisiones.
- **Número de satélites visibles:** la precisión aumenta al aumentar el número de satélites visibles. La señal de satélite puede verse bloqueada por irregularidades del terreno, interferencias electrónicas o edificios, causando errores de posición o incluso haciendo que no sea posible obtener una posición.
- **Geometría de los satélites:** cuando los satélites disponibles se encuentar alineados o muy próximos entre sí los errores afectan de forma más adversa a la posición. La geometría más idónea se da cuando los ángulos entre satélites están entre 60º y 120º.
- **Disponibilidad selectiva:** la señal satélite se puede degradar de forma intencionada por el Departamento de Defensa de los EEUU para dificultar a su adversario obtener una posición precisa. En el año 2000 se apagó la disponibilidad selectiva, o SA por sus siglas en inglés, lo que supuso una gran mejora para los GPS civiles.

Para obtener una sitiación más precisa es preciso comparar la posición del receptor de forma simultánea con la obtenida por otro situado en una posición conocida. Este procedimiento se conoce como GPS diferencial, o DGPS [38]. El fundamento de este sistema se basa en emplear un receptor de posición conocida como punto de referencia estático, de forma que se pueda conocer la distancia exacta al satélite y compararla con la distancia obtenida, conociendo así el error de los datos satélite. La estación receptora transmite así un mensaje de corrección de errores a otros receptores que se encuentran en su vecindad que les permite corregir las soluciones de sus posiciones.

Esta estación, conocida como “estación monitora” y que es independiente de las estaciones del segmento terrestre del sistema se encarga de calcular el error diferencial, que es la diferencia entre la distancia real al satélite y la recibida, y transmitirlo a los receptores GPS próximos. Con este sistema se puede reducir el error de GPS hasta menos de un metro.

También existen antenas emisoras en tierra que emiten en la misma frecuencia y con el mismo código que los satélites, de forma que actúan como un “pseudosatélite”.

2.3 Fases de implementación del proyecto

El proyecto se ha implementado de forma progresiva en tres plataformas diferentes:

- **Primera fase:** Robot Arduino: En primer lugar se ha empleado el Arduino Robot para familizarizarse con el entorno de programación.
- **Segunda fase:** Coche Radiocontrol: debido a las limitaciones de uso y conexión de accesorios del robot se adapta un coche radiocontrol
- **Tercera fase:** Lancha Radiocontrol: plataforma final de desarrollo del proyecto, y primer paso para implantar el módulo en medios navales.

2.3.1 Primera fase: robot Arduino

En primera instancia se ha empleado el dispositivo *"Arduino Robot"* [41] para familiarizar al usuario con el entorno de Arduino. El Arduino Robot consta de dos placas con un procesador ATmega32u4, que es el mismo que el que tiene la placa Arduino Leonardo, unidas entre sí para formar una estructura. El robot tiene asimismo un portapilas integrado y diversos sensores y actuadores, como dos motores de corriente continua, una pantalla LCD, un lector de tarjetas microSD, un pequeño altavoz, un magnetómetro y varios pulsadores. Dado que ya viene montado y con una serie de programas ya hechos, resulta muy útil para familiarizarse con el entorno de Arduino.

Figura 2-9 Arduino Robot (fuente: [42])

No obstante, esto supone también una desventaja, ya que las opciones de incluir nuevos sensores o actuadores se ven limitadas, puesto que muchas de las entradas de la placa están ocupadas por los dispositivos propios de la placa. Además, las ruedas finas y la reducida altura del robot lo confinan a superficies lisas, lo que imposibilita probarlo en exteriores o incluso en casas con alfombras. Esto, sumado a su elevado coste en comparación con robots de similares características construidos uniendo placas Arduino con sensores y actuadores de forma separada han llevado a la retirada del mercado del robot apenas tres años después de su aparición.

Los programas probados con el robot, la mayoría de ellos modificaciones a la librería del mismo, incluyen:

- prueba de avance, giro y retroceso de forma cíclica.
- Control de la secuencia de movimientos a través de los pulsadores propios del robot, mostrándolo en pantalla.
- Regulación de la velocidad del robot mediante el potenciómetro.
- Uso del altavoz disponible para producir melodías sencillas.
- Muestra de imágenes almacenadas en la memoria extraíble.
- Calibración del magnetómetro interno.

Una vez familiarizado con el entorno Arduino se pasó a desarrollar el módulo de control e implementarlo en un coche radiocontrol que se pueda probar en exteriores y con rutas más largas.

2.3.2 Segunda fase: coche de radiocontrol.

La Figura 2-10 muestra el coche de radiocontrol utilizado en este TFG.

Figura 2-10 Coche radiocontrol original (Fuente: [43])

Se trata del modelo eléctrico de un 4x4 con unas dimensiones de aproximandamente 40 cm de largo por 25 cm de ancho. Dispone de ruedas de tacos y una altura al suelo suficiente para circular por terrenos irregulares.

Para la propulsión cuenta con un motor eléctrico en la parte trasera que transmite el movimiento a las ruedas traseras a través de un tren de engranajes y un diferencial. Para la dirección de las ruedas delanteras se emplea otro motor eléctrico conectado a un tren de engranajes. Este tren de engranajes mueve a través de una rueda excéntrica dos vástagos, cada uno conectado al eje de cada rueda. Un potenciómetro conectado al tren de engranajes nos permite conocer y modificar la posición de la dirección en función de la resistencia eléctrica medida en el mismo.

El control del vehículo originalmente se realizaba a través de un circuito eléctrico, que se encargaba de traducir las órdenes recibidas por radiofrecuencia a través de una antena a los motores de propulsión y dirección. La alimentación original provenía de un conjunto de baterías de 9 voltios alojadas en la parte inferior.

2.3.2.1 Adaptación de la plataforma

El primer paso para adaptar el coche consistió en desmontar la carrocería plástica exterior y acceder a la placa de control. Dado que el módulo de control actúa directamente sobre los motores la placa carece de utilidad para este proyecto, por lo que es preciso retirarla.

Una vez retirada se procede a identificar la función de cada cable. Por un lado hay dos cables conectados al motor trasero que le sirven para avanzar o retroceder en función de la polaridad. De la parte delantera salen cinco cables:

- Tres de ellos están conectados al potenciómetro que permite conocer la posición de la dirección en función del valor de su resistencia.
- Los otros dos actúan sobre el motor delantero, permitiendo girar a la derecha o a la izquierda.

Conocidas las funciones de cada cable, se vuelve a montar el coche. En la Figura 2-11 se muestra el resultado final.

Figura 2-11 Coche radiocontrol adaptado

2.3.2.2 Instalación del módulo de control

El módulo de control está formado por la placa Arduino, una placa de relés y los diversos sensores que se irán incorporando, como el magnetómetro o el sensor GPS.

Estos elementos se agruparon inicialmente en una placa de polipropileno en panal de abeja cortada a la medida del hueco existente en el chasis del coche. Posteriormente, dada la baja solidez de la placa, se optó por sustituirla por una placa de la misma medida de madera contrachapada de 3 mm.

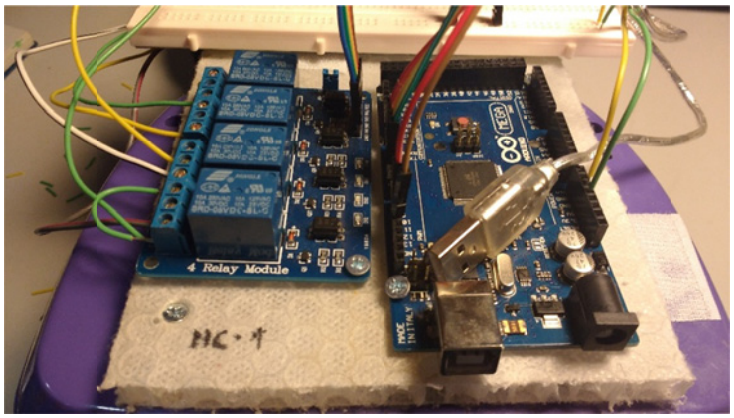


Figura 2-12 Placa original

Figura 2-13 Sustitución por placa de madera

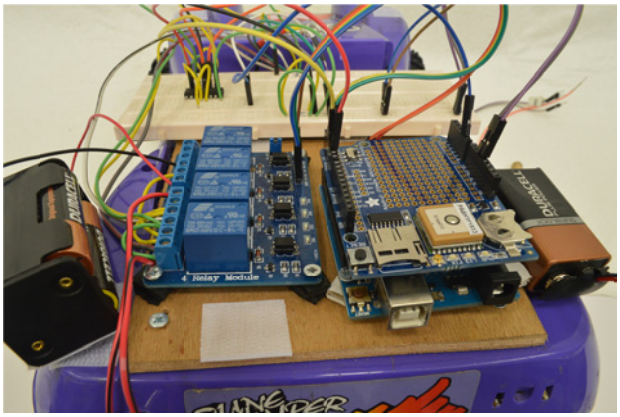


Figura 2-14 Montaje final sobre la placa de madera

Los motores eléctricos del coche son motores de corriente continua, compuestos por dos partes:

- **Estator:** parte fija del motor, cuenta con un imán permanente o un electroimán.
- **Rotor:** está formado por un conjunto de bobinas alimentadas a través de unos contactos eléctricos conocidos como escobillas.

El funcionamiento de dichos motores, que se explica de forma esquemática en [44], es sencillo: al alimentar la bobina con corriente continua se induce un magnetismo en la misma. La fuerza magnética generada entre la bobina y el imán del estator es la que produce el giro del rotor. Las escobillas permiten cambiar las diferentes bobinas que reciben alimentación, de forma que la que se encuentre en mejor posición para generar un par de fuerzas capaz de mover el eje es la que recibe tensión.

Para controlar el motor de propulsión emplearemos una placa de relés. Un relé es un interruptor controlado eléctricamente, en el que habitualmente la potencia empleada para abrir el interruptor es menor que la que circula por este. Para el proyecto se ha empleado una placa que agrupa 4 relés y los circuitos de control necesarios para poder controlarlos a través de las salidas digitales del Arduino [45].

La placa de relés empleada tiene seis pines: Tensión, masa y una entrada digital por cada uno de los cuatro relés. Cada motor requiere dos relés para poder cambiar el sentido de giro. En este caso, se emplean las salidas digitales 2 y 3 del Arduino a la placa de relés, de forma que cuando ambos estén abiertos o cerrados el coche se pare, cuando uno de ellos esté abierto y el otro cerrado se cierra el circuito y el coche avanza y cuando estos se encuentren de la forma opuesta se invierte la polaridad y el coche retrocede. La Figura 2-15 muestra el esquema del circuito empleado para controlar el motor mediante los relés, y la Figura 2-17 muestra la instalación de la placa en el vehículo.

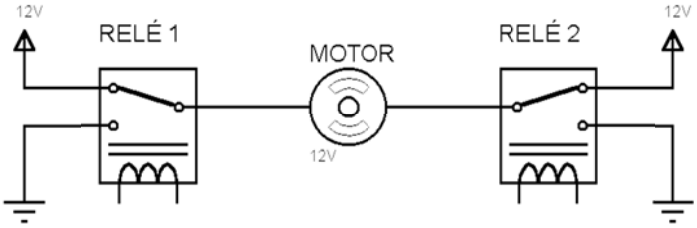


Figura 2-15 Control motor con relés (fuente: [46])

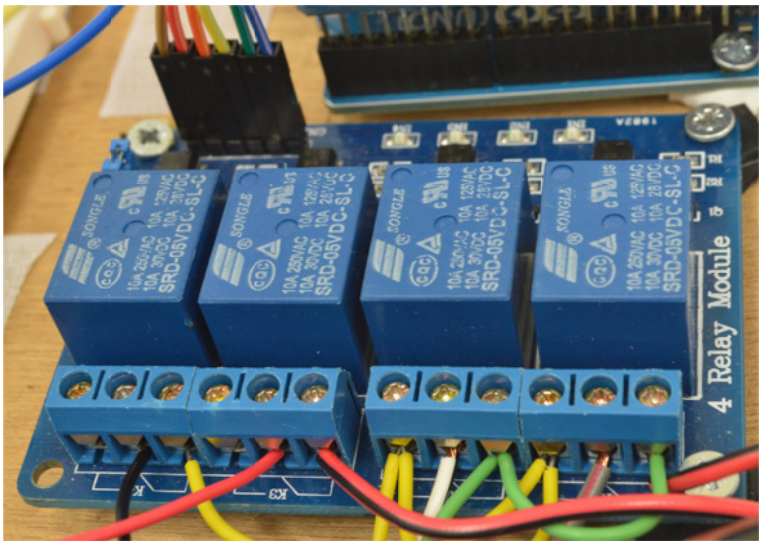


Figura 2-16 placa de relés

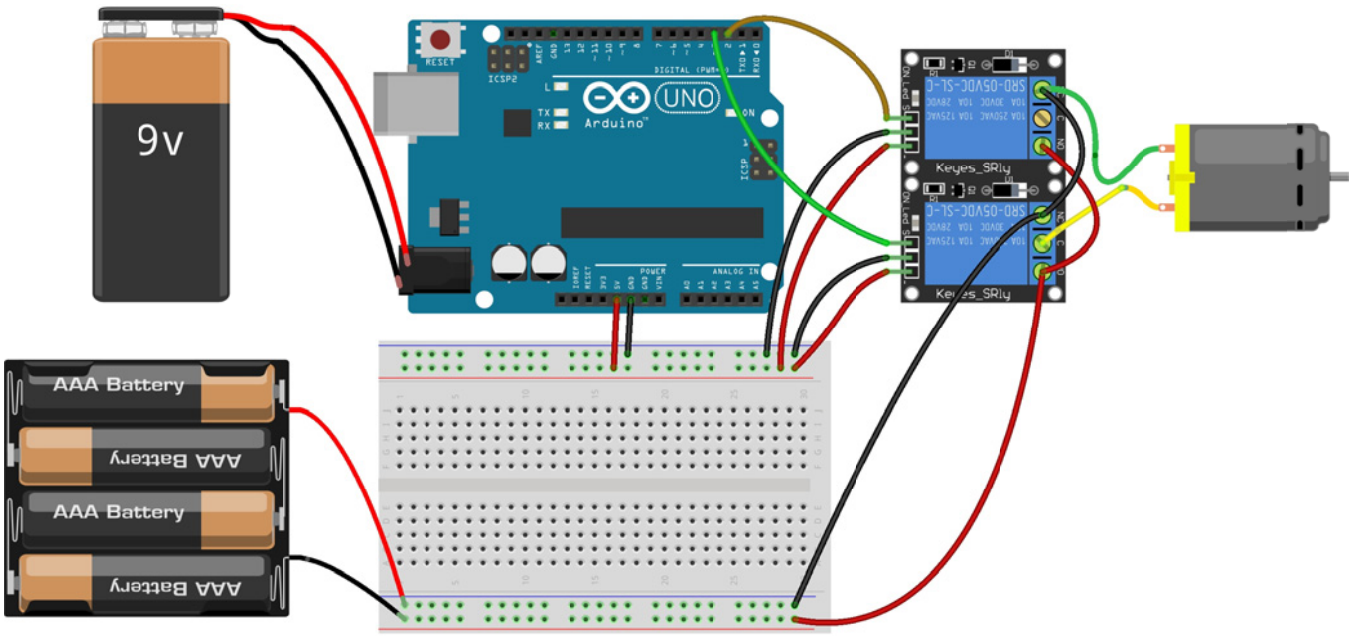
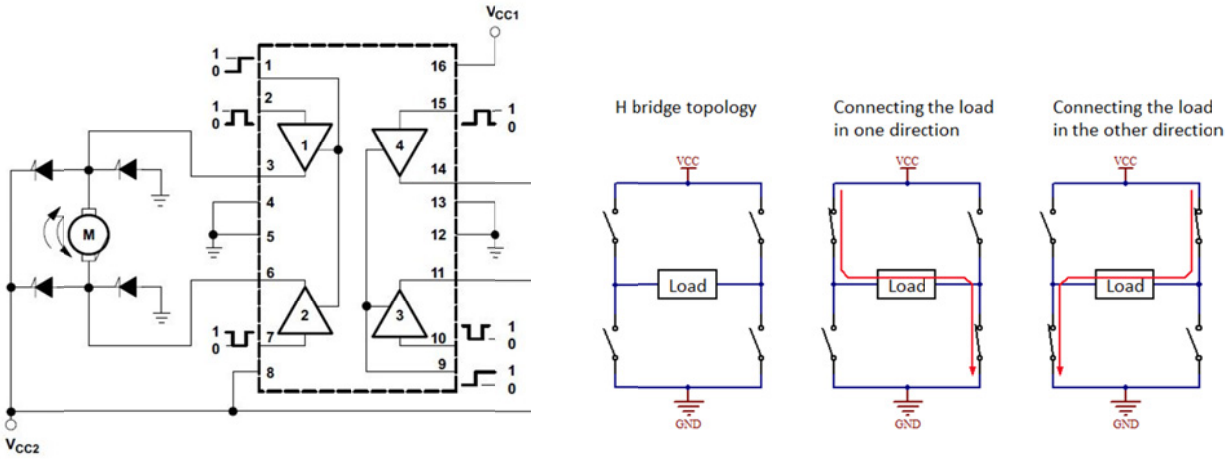


Figura 2-17 Esquema eléctrico del control de propulsión

Para conocer la posición de la dirección se emplea un potenciómetro, o resistencia variable, que gira solidario al tren de engranajes de la misma. El potenciómetro tiene los cables de los extremos conectados a las salidas de 5V y masa del Arduino y el cable central conectado a la entrada analógica A1 del mismo. La tarjeta cuenta con seis entradas analógicas, en las cuales utiliza un conversor analógico/digital de 10-bits [29], capaz de mapear los valores de voltaje de la entrada analógica a valores enteros comprendidos entre 0 y 1023 (2^{10-1}). Este valor numérico será el que emplearemos en el programa para saber en qué posición se encuentra la dirección del coche.

Originalmente el motor encargado de la dirección se controlaba mediante la placa de relés, pero se comprobó que empleando este sistema analógico el control es demasiado impreciso. Por ello se opta por controlar el mismo de forma electrónica mediante un circuito conocido como puente H. El puente H agrupa una serie de transistores que, de forma análoga al circuito de relés empleado para la propulsión, permite cerrar el circuito del motor e invertir la polaridad para regular su giro en ambos sentidos.



(a) Esquema del circuito L293D (fuente: [47])

(b) Funcionamiento del puente H (fuente: [48])

Figura 2-18 Puente H

En el mercado se pueden encontrar varios circuitos integrados con el puente H ya instalado. En este caso se ha optado por el L293D, un circuito integrado muy económico y cuyas características son suficientes para este proyecto [47]. La dirección con el puente H es notablemente más precisa que mediante relés. El conexionado final con la placa Arduino se muestra en la Figura 2-19.

Una vez instalado el módulo básico de control se procede a realizar diversas pruebas para comprobar el funcionamiento. Estas pruebas permiten obtener empíricamente qué relés es necesario activar para que el coche avance, retroceda, gire a la derecha y gire a la izquierda y qué valores del potenciómetro se corresponden con dirección centrada y máximo giro a derecha e izquierda.

Inicialmente se alimentó la placa Arduino con una pila de de 9 voltios y los motores con una batería USB de 5 voltios. No obstante, el consumo del motor de propulsión excedía la intensidad máxima proporcionada por la batería, por lo que la protección por sobretensión de la batería USB se activaba y dejaba de alimentar al motor. Por ello se descartó la batería y se pasó a alimentar con 4 pilas AA conectadas en serie agrupadas en un portapilas.

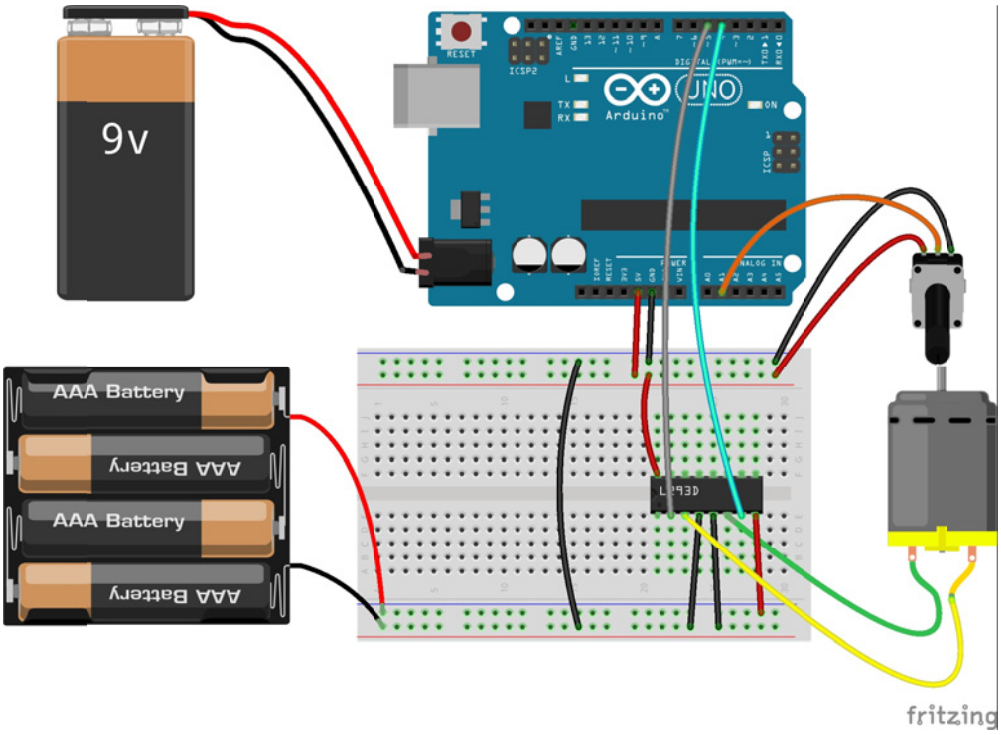


Figura 2-19 Esquema del control de dirección

2.3.2.3 Detección de obstáculos

Para la detección de obstáculos se le añade al vehículo un sensor de ultrasonidos HC-SR04 [49]. Este sensor funciona de forma análoga a un SONAR: emite un pulso de ultrasonidos y de forma simultánea inicia un temporizador. Al encontrar un obstáculo, el pulso rebota en el mismo y es detectado a su vuelta, parando el temporizador. Sabiendo que la velocidad del sonido en el espacio es de 340 m/s podemos conocer la distancia a dicho obstáculo [50].

$$R = \frac{v \cdot t}{2} \tag{1}$$

Cada sensor tiene 4 pines: dos de ellos conectados a tensión y masa y los otros dos, llamados *trigger* y *echo*, encargados de emitir el pulso y detectarlo, respectivamente. Estos dos últimos se conectan en las entradas y salidas digitales de la placa Arduino, como se muestra en la Figura 2-20.

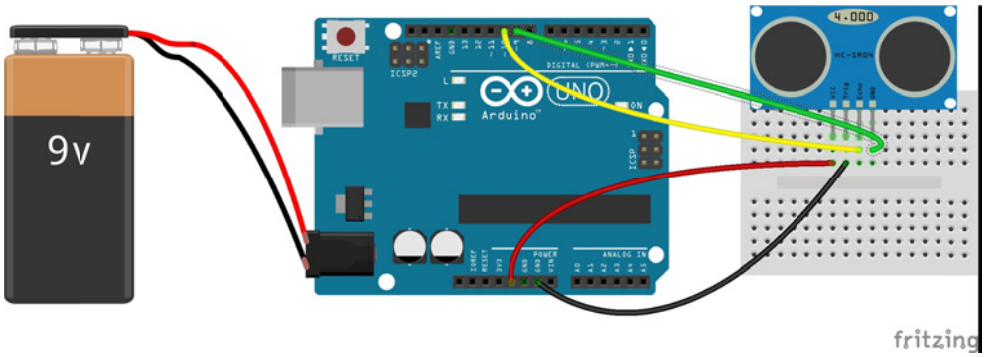


Figura 2-20 Esquema de montaje de un sensor de distancia

Durante las primeras pruebas, realizadas en interiores, con una tensión de alimentación de 6 Voltios, se constata que el coche avanza tan rápido que no es capaz de parar antes de impactar contra el obstáculo. En consecuencia, se sustituye el portapilas de 4 pilas y 6 voltios por uno de 2 pilas y 3 voltios, para reducir la velocidad del vehículo, con lo que se consiguen los resultados deseados. Así se consigue que el coche avance hasta encontrar un obstáculo, se detenga, gire, retroceda y avance en otro sentido.

Figura 2-21 Montaje inicial con un sensor delante

Se comprueba que con un solo sensor detecta bien los obstáculos cuando se aproxima de forma perpendicular, pero falla en ocasiones al aproximarse de forma oblicua. Además, al retroceder para cambiar de dirección en sitios angostos puede golpearse por detrás. Para solventarlo, se le incorporan dos sensores en la parte delantera con cierto ángulo con respecto a la línea longitudinal y uno en la parte trasera, y se mejora el programa para que decida qué sensor ha detectado el obstáculo y actúe en consecuencia girando hacia el lado contrario, deteniéndose además si detecta un obstáculo detrás.

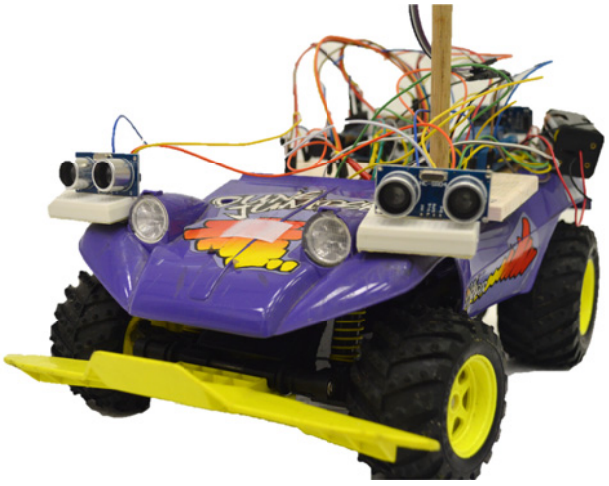


Figura 2-22 Montaje mejorado, con dos sensores delante y uno detrás

Los sensores de ultrasonidos permiten conocer la distancia con precisiones cercanas a 1 cm, pero devuelven ecos falsos con frecuencia, lo que conduce a que el coche se detenga mienras se encuentra en funcionamiento sin tener obstáculos delante. Para subsanar este error es preciso tomar varias mediciones y hallar la mediana de las mismas, de forma que se descarten las distancias que sean

dispares. El programa Arduino no dispone *per se* de funciones estadísticas, pero se le pueden introducir librerías que permiten calcular medianas y otros cálculos matemáticos.

También se constata que en ocasiones el procesador actúa de forma diferente al estar conectado a una pila que al estar conectado al ordenador. Se llega a la conclusión de que el elevado consumo de los sensores y actuadores conectados al Arduino (15 mA por sensor [49] y 20 mA por relé [45]) puede ocasionar un déficit de tensión en el procesador y conducir a estos errores.

En base a los resultados obtenidos, se puede concluir que los sensores de ultrasonidos resultan de utilidad para el proyecto. Sin embargo, presentan dos desventajas principales:

- Su alcance máximo está limitado a 4 metros, lo que dificulta su uso en mar abierto.
- Devuelven ecos falsos con demasiada frecuencia.

Para solventar estos inconvenientes, se abre el camino para explorar alternativas, como el uso de telémetros láser, con mucho mayor alcance y directividad, o radares convencionales.

2.3.2.4 Introducción módulo GPS

El objetivo principal del presente trabajo es el lograr que el módulo de control dirija de forma autónoma por una ruta prefijada a un vehículo. Por ende, la navegación GPS es el punto clave del proyecto.

El módulo GPS elegido es el *FGPMMOPA6H* de la empresa *GlobalTop* [51], capaz de buscar hasta 66 canales y recibir 22 de forma simultánea, con una sensibilidad de -165dBm y un consumo reducido, de unos 20 mA.

El módulo GPS elegido se encuentra en formato *shield*, por la facilidad de montaje y conexión que supone. Además, el shield dispone de una librería asociada que incluye los cálculos necesarios de análisis de las sentencias NMEA obtenidas del GPS, o *parsing*, lo que facilita notablemente la programación.

La comunicación entre el *shield* y la placa se produce por el protocolo de comunicación *software serial* [31] a través de los pines 7 y 8. La comunicación serie [52] es aquella en la que se envía un bit de cada vez. En la placa Arduino se emplea un elemento, conocido como *UART* [53] por sus siglas en inglés, que permite establecer comunicaciones entre el ordenador y la placa incluso mientras hay un programa en ejecución. Para la comunicación entre el *shield* y la placa se que establece una *UART* secundaria en los pines 7 y 8 de forma que la *UART* principal, que ocupa los pines 0 y 1, queda libre para cargar programas en la placa. La Figura 2-23 muestra el esquema de conexionado del *shield* GPS.

Como contrapartida, el *shield* [37] no puede conectarse directamente al Arduino MEGA debido a que éste no soporta comunicación *software serial* en dichos pines. Por ello, se opta por cambiar la placa MEGA por una placa UNO, de capacidad inferior a la MEGA pero suficiente en esta fase del proyecto.

A raíz de este cambio de placa es preciso volver a cablear las entradas y salidas y volver a comprobar los programas anteriores del MEGA con la placa UNO.

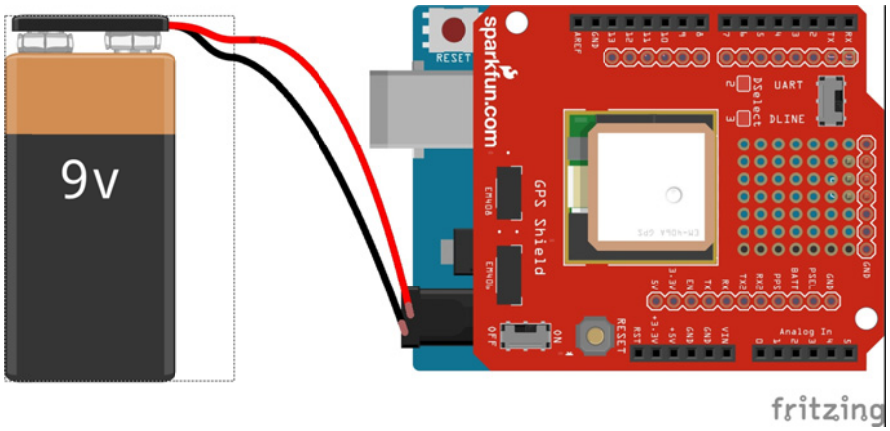


Figura 2-23 Esquema del *shield* GPS

Para el funcionamiento del programa de navegación es preciso conocer también el rumbo del vehículo. El sensor GPS proporciona un valor aproximado basado en posiciones sucesivas, pero para obtener un valor más preciso se emplea un magnetómetro. El magnetómetro mide la dirección del campo magnético terrestre, de forma análoga a una brújula. En primer lugar se hicieron pruebas con el sensor MPU 9250 [54] , que incluye giróscopo y acelerómetro además del citado magnetómetro. La falta de librerías para este sensor capaces de dar una lectura correcta llevó a descartarlo.

El magnetómetro elegido finalmente es el *Honeywell HMC5883L* [55], para el que existen varias librerías comprobadas. EL HMC5883L es un magnetómetro de tres ejes que funciona a 33 voltios, montado en una placa adaptada a Arduino que permite la comunicación con la placa mediante el protocolo *I2C* [30]. El protocolo *I2C* permite la comunicación entre numerosos dispositivos empleando únicamente dos cables: *SDA* para datos y *SCL* para el reloj. En el Arduino UNO estos conectores se encuentran en las entradas analógicas 4 y 5. En la Figura 2-24se muestra el conexionado del sensor.

Cabe destacar la elevada sensibilidad del magnetómetro, ya que los valores variaban de forma apreciable con el campo magnético de la propia placa. Debido al magnetismo inducido y permanente de los motores y de otros elementos, se opta por situar el magnetómetro en una pértiga de madera con el sensor fijado con cinta adhesiva, ya que tras atornillarlo se comprobó que el campo magnético del tornillo invalidaba completamente las lecturas del sensor.

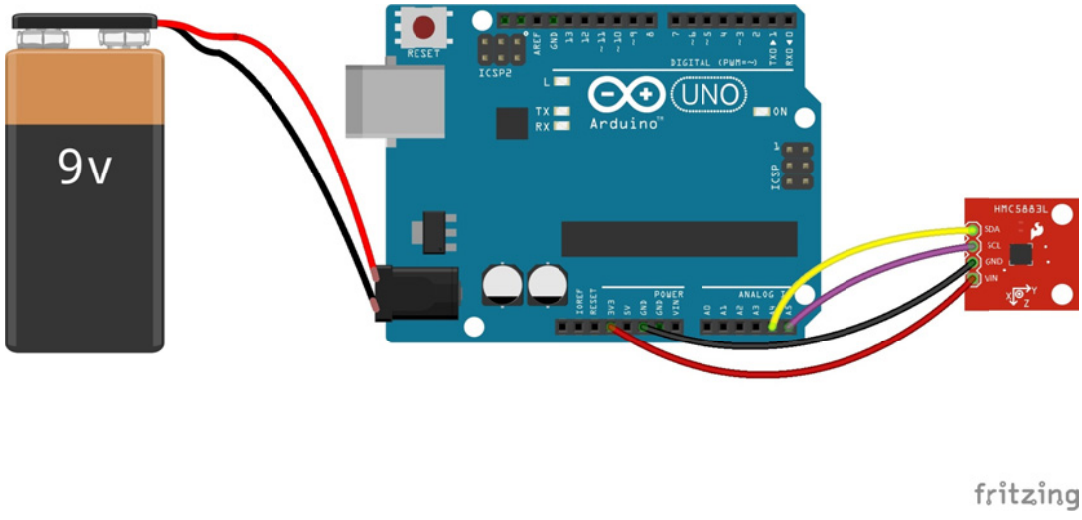


Figura 2-24 Esquema del magnetómetro

La Figura 2-25 muestra el esquema de montaje completo del vehículo, con todos los sensores necesarios para la navegación por GPS.

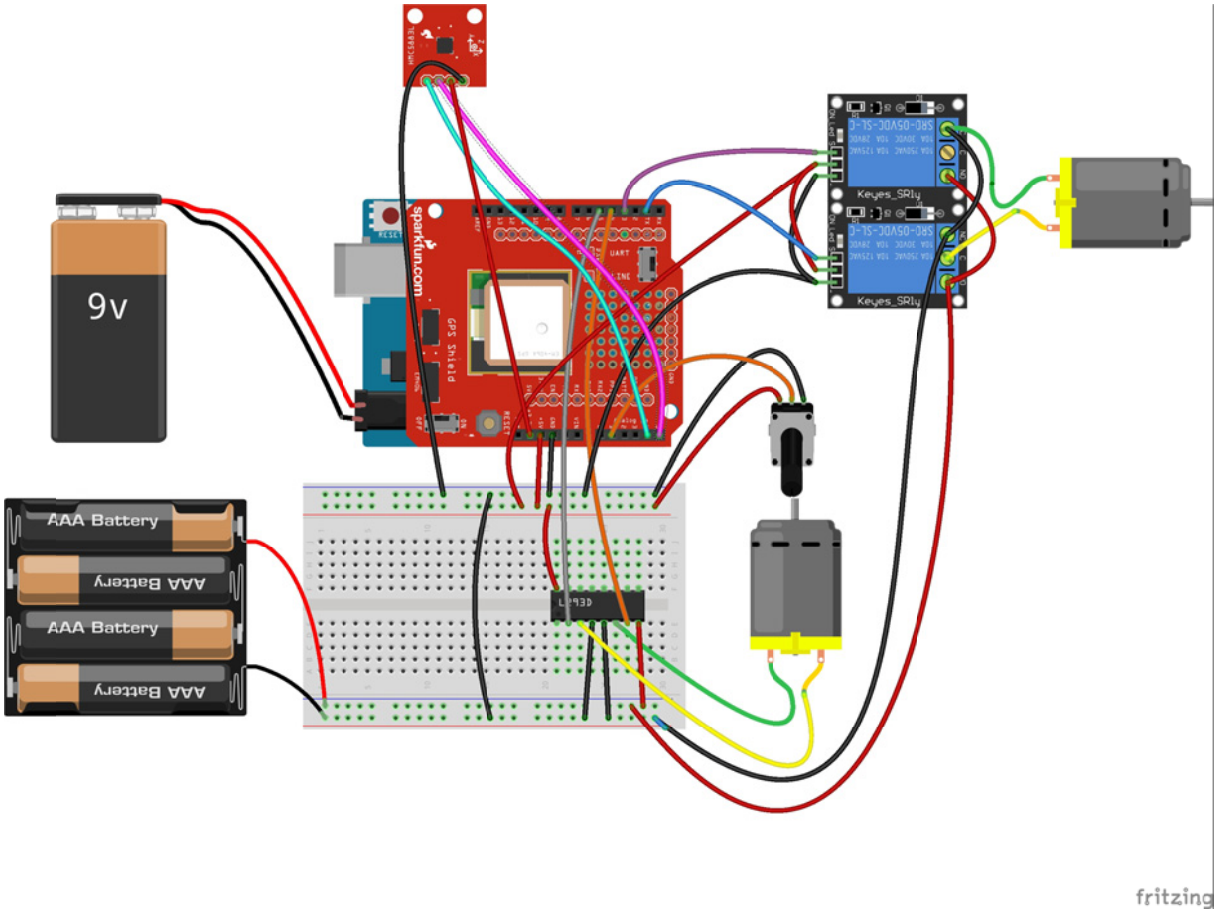


Figura 2-25 Esquema del montaje completo

2.3.3 Tercera fase: lancha

2.3.3.1 Adaptación de la plataforma

Tras realizar las pruebas en medio terrestre con el coche radiocontrol se pasa a las pruebas en medio acuático, que se efectuarán con la lancha radiocontrol que se muestra en la Figura 2-26 [56]. Esta lancha tiene unas dimensiones de 62 centímetros de eslora por 25 de manga y está propulsada por dos motores eléctricos alimentados por una batería recargable que le confieren una autonomía superior a las dos horas y una velocidad máxima de cerca de tres nudos. Dispone además de un sistema de volquete pensado originalmente para usar como dispensador de cebo y de una luz led en la parte superior. El control de los propulsores y del volquete se efectúa a través de un mando a distancia que opera a una frecuencia de 27 MHz. Cabe destacar que la lancha no cuenta con la funcionalidad de marcha atrás.

Figura 2-26 Lancha radiocontrol

La principal diferencia en cuanto al control con el coche consiste en que, mientras que este tiene un motor de propulsión y un servo de dirección, la lancha cuenta con dos motores de propulsión y la caída a una banda u otra se logra deteniendo uno de ellos.

Figura 2-27 Detalle de los motores

Debido a la complicación para recuperar la lancha en caso de fallo y a que ésta será empleada en futuros proyectos la adaptación de la misma tiene dos requisitos:

- La lancha debe poder ser operada con el mando radiocontrol en todo momento.
- No será posible realizar modificaciones permanentes en su estructura eléctrica y unidad de control.

Teniendo esto en cuenta, se procede a desmontar la lancha para analizar la forma en la que controla los motores. Los motores y la batería se encuentran anclados en el casco, que está pegado y atornillado a la cubierta (Figura 2-29). En la cubiera se encuentra también el motor encargado de levantar y bajar el volquete mediante un tornillo sin fin con sistema de inversión automática [57]. En el casco también se encuentran dos luces en las aletas, que parpadean cuando el barco gira y se iluminan de forma continua cuando avanza. Por otra parte, existe una tapa extraíble que se atornilla a la cubierta y está sellada mediante una junta de goma donde se encuentra la unidad de control, que incluye el receptor de radiofrecuencia, el regulador de carga de la batería, el interruptor de funcionamiento y el interruptor magnético que indica cuando el volquete está en posición de reposo (Figura 2-28).

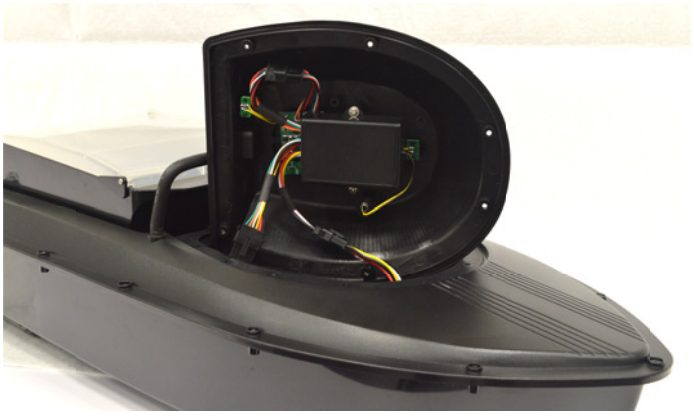


Figura 2-28 Tapa donde se encuentra la unidad de control

Figura 2-29 Vista de la batería y cableado de los motores

La batería se puede desconectar para evitar que se descarge durante el almacenamiento. La unidad de control se conecta al casco a través de dos conectores, uno de seis pines y otro de cuatro. El conector de seis pines se encarga de las luces del casco y el volquete, por lo que no es relevante para el proyecto. El conector de cuatro pines es el que se encarga de alimentar a la unidad de control y actuar sobre los motores. Además, los pines de estos terminales, tanto los macho como los hembra, se pueden conectar directamente con los cables empleados en Arduino. Se comprueba que al unir el terminal rojo con los terminales blanco o amarillo se mueven los motores de estribor y babor respectivamente, por lo que se opta por realizar realizar un puente en una *protoboard* de forma que se mantenga la funcionalidad original del controlador y se pueda cerrar el circuito también mediante dos relés controlados por Arduino. Para mantener la estanqueidad del conjunto se opta por colocar la protoboard, el microcontrolador Arduino y los relés en una placa de madera contrachapada cortada a medida y sujeta a la tapa mediante velcro, de forma que quede cerrada estancamente.

2.3.3.2 Instalación del módulo de control

El esquema de conexión de la propulsión de la lancha es semejante al empleado en el coche. Sin embargo, sólo hace falta un relé por motor ya que el giro de estos no se invierte. En la siguiente imagen se incluye el conexionado, incluyendo la batería de litio de la lancha:

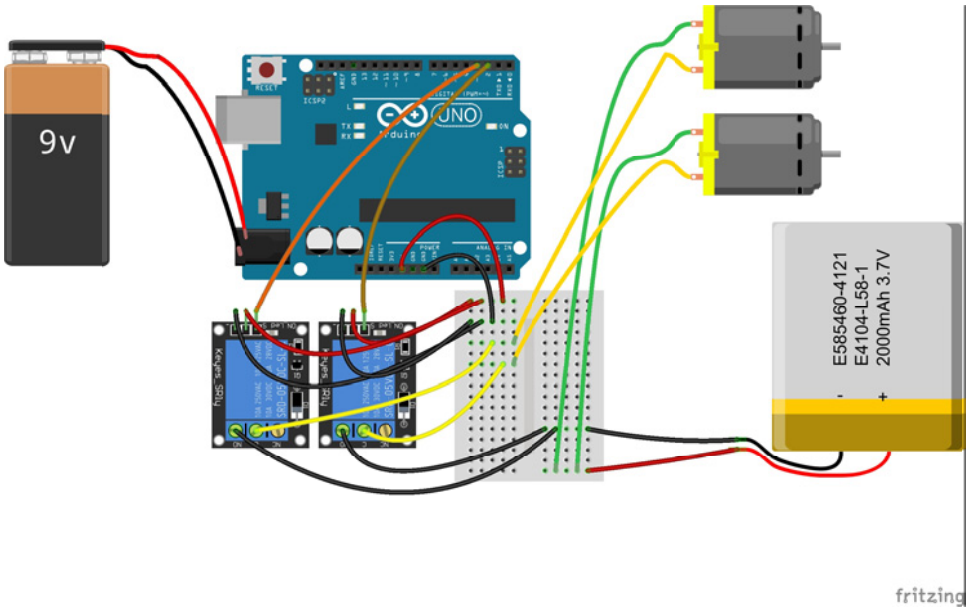


Figura 2-30 Esquema del control de propulsión

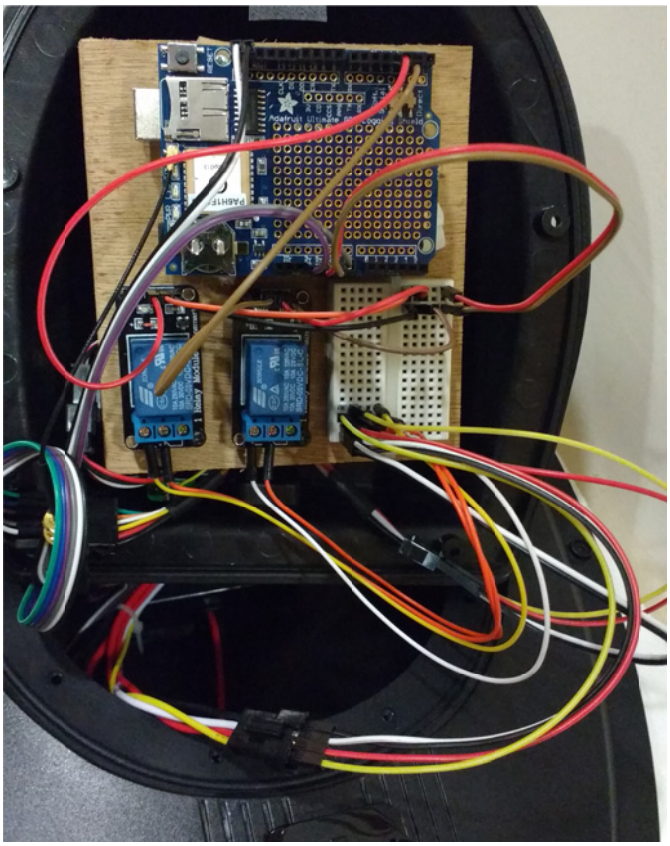


Figura 2-31 Instalación del microcontrolador en la lancha

De forma análoga al coche, también es preciso instalar el magnetómetro. Para la instalación del mismo existe un compromiso entre mantenerlo alejado de los campos magnéticos generados por el cableado, los motores y la placa y protegerlo del agua. Como el sensor quedaría muy expuesto si se sitúa en una pértiga, se opta por situarlo dentro del casco pero lo más alejado posible de los elementos interferentes (especialmente los dos motores y la antena GPS, que está imantada para facilitar su

fijación a superficies metálicas). Así, se sitúa a proa de la lancha sobre un soporte creado para ello, y se comprueba que las lecturas dentro y fuera de la lancha no varían de forma significativa.

El módulo GPS se sitúa asimismo enganchado a la placa Arduino. Como el módulo va situado en el interior del casco y el sensor está invertido y cubierto por la placa y el receptor del mando a distancia se opta por conectarle una antena GPS externa para asegurar que recibe una buena posición.

La antena elegida, recomendada por el fabricante del módulo GPS, es una antena activa que proporciona una ganancia de 28 dB al receptor [58]. Como es activa tiene un consumo de 10 mA, que se ven más que justificados por la mejora de la calidad de la situación. La antena tiene una base magnética y es resistente a la intemperie, y se conecta mediante un cable coaxial rematado en un conector *SMA*, típicamente empleado en estas antenas. El *shield* GPS cuenta con un conector de reducido tamaño de tipo *uFL*, por lo que es necesario usan un adaptador. Con objeto de minimizar las interferencias producidas por la antena, se opta por situarla en la parte posterior de la lancha.

El montaje completo de la lancha queda como sigue:

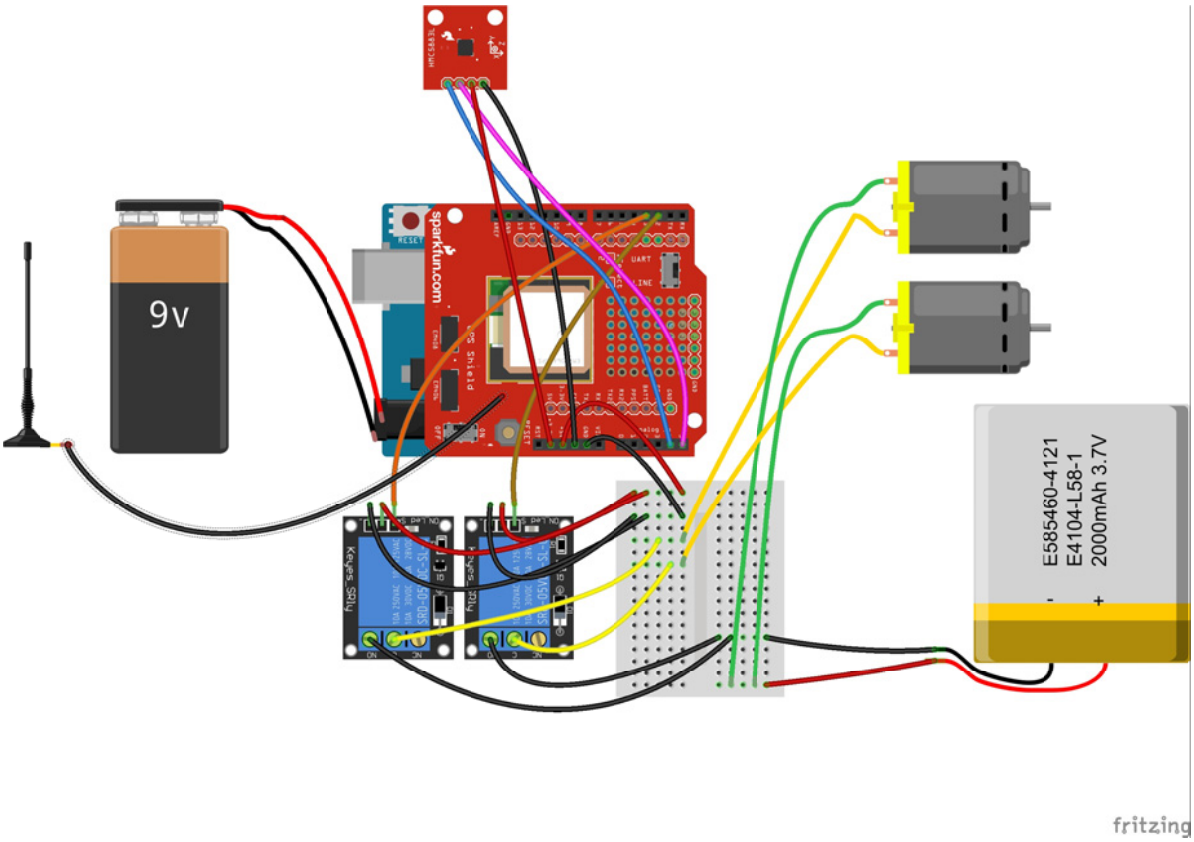


Figura 2-32 Esquema completo de la lancha

3 RESULTADOS, PRUEBAS Y VALIDACIÓN

3.1 Pruebas en tierra

3.1.1 Pruebas iniciales y depuración del programa

Una vez instalado el *shield* se comprueba el funcionamiento del sensor GPS usando el módulo direct connect del mismo, que permite puentear la placa y obtener las sentencias *NMEA* [59] directamente en el monitor serie.

La información transmitida por el GPS se encuentra en formato *NMEA* (National Maritime Electronics Association). Este protocolo se basa en sentencias defindas por la primera palabra, conocida como tipo de datos. Las dos clases de sentencias que se emplean en el proyecto son la $GPRMC y la $GPGGA. Ambas proporcionan fecha y hora, coordenadas, altitud, velocidad estimada y tipo de situación. Este ultimo determina si el GPS tiene comunicación con suficientes satélites como para obtener una posición y si esta es bidimensional (posición) o tridimensional (posición y altura). La interpretación de las sentencias NMEA es la siguiente:

```
$GPGGA,123519,4223.509,N,00842.765,W,1,08,0.9,43.5,M,46.9,M,,,*47
```

Donde:

- GGA: Hace referencia al tipo de sentencia
- 123519 : Posición tomada a 12:35:19 UTC
- 4223.509,N: Latitud 42º 23.509’ N
- 00842.765,W: Longitud 8º 42.765 W
- 1: tipo de situación. Los más frecuentes son 0(sin situación), 1 (situación GPS) y 2 (GPS diferencial)
- 08: Número de satélites recibidos
- 0.9: Error de posición horizontal
- 43.5, M: Altitud 43,5 metros sobre el nivel del mar
- 46.9,M: Altitud del Geoide a nivel del mar sobre el elipsoide WGS84
- (campo vacío) Tiempo desde última situación DGPS
- (campo vacío) Número de identificación de estación DGPS
- *47: Suma de dígitos para comprobación

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6ª
```

Donde:

- RMC: Tipo de sentencia (*Recommended Minimum sentence C*)
- 123519: Situación tomada a las 12:35:19 UTC
- A: Estado de la conexión: A(activo) o V (nulo)
- 4807.038,N: Latitud 48º 07.038' N
- 01131.000,E: Longitud 11 deg 31.000' E
- 022.4: Velocidad en nudos
- 084.4: Dirección desplazamiento en grados
- 230394: Fecha de la posición 23 de marzo de 1994
- 003.1,W: Declinación magnética
- *6A: Suma de dígitos para comprobación

Tras comprobar que las sentencias son obtenidas correctamente, se procede a usar el *shield* en modo *software serial*, en el que la placa se encarga de procesar los datos. Se vuelve a comprobar que se obtienen datos GPS correctos, esta vez empleando los programas que incluye la librería del sensor. Se comprueba que únicamente se obtiene posición cuando el sensor se sitúa cerca de una ventana, por lo que en el módulo final será preciso conectar una antena externa para aumentar la sensibilidad del sensor. Trasladando la posición obtenida al programa cartográfico *Google Earth* el error obtenido es de unos 15 metros en interiores. En exteriores este error se reduce notablemente, llegando a ser inferior a 3 metros (debido, en gran parte, al número de satélites a los que se conecta el sensor en cada caso).

Para un sistema de navegación nos encontramos con los siguientes requisitos:

- **Posición actual:** en este caso, disponemos de un sensor GPS.
- **Posición de destino:** los puntos de destino se obtienen mediante software cartográfico. Google Maps es un recurso sencillo y accesible y permite tanto obtener las coordenadas de un punto como representar en el mapa las coordenadas obtenidas. El programa de navegación se encarga de calcular el rumbo que hay que seguir desde la posición actual para llegar a esta posición.
- **Rumbo del vehículo:** la diferencia entre el rumbo al destino y el rumbo del vehículo es la que determina si se ha de seguir recto o es preciso girar. El sensor GPS proporciona un rumbo basado en la obtención de posiciones sucesivas, pero tiene errores de precisión que hacen recomendable obtenerlo a través de un sensor independiente.

El programa incial empleado para la navegación toma como base dos proyectos encontrados en la red para controlar de forma autónoma coches radiocontrolados, [14] y [15].

En primer lugar se toma como punto de partida uno de los proyectos [14] que emplea el mismo módulo GPS y un magnetómetro de características similares. Con todo, es preciso adaptar la plataforma, ya que tanto el control de avance y retroceso como el de giro se realizan de forma diferente. Tras varias pruebas infructuosas en las que el vehículo actúa de forma extraña y gira en bucles se opta por hacer el programa partiendo de cero, lo que permite identificar errores y subsanarlos más eficazmente.

Dicho programa se realiza en tres fases:

- La primera fase consiste en programar el vehículo para que avance en línea recta hacia un waypoint y se detenga al llegar.
- La segunda se encarga también de realizar las correcciones de rumbo precisas en caso de que el vehículo no se encuentre apuntando hacia el destino.
- La tercera incluye varios waypoints, de forma que el coche siga una ruta.

La clave del éxito de este programa reside en una característica propia del microcontrolador Arduino, la función bucle, que es el núcleo de todos los programas Arduino. Los programas anteriores contenían un bucle *while* dentro de esta función, lo que no permitía actualizar de forma constante los datos de GPS y compás y por tanto conducían a error. Tras un programa fallido empleando un bucle *while*, se vuelve a probar la primera fase del programa sustituyendo el bucle por una condición que se comprueba de forma constante (dentro del propio *loop* del Arduino), esta vez con éxito.

Para la segunda fase del programa se realiza un primer programa introduciendo el cálculo del giro que es necesario hacer dentro del temporizador que analiza los datos GPS, que se realiza cada segundo. En la primera prueba, a pesar de que el programa funciona, el vehículo realiza grandes correcciones de rumbo sin llegar a converger en una trayectoria rectilínea. Tras analizar las causas de este comportamiento, se concluye que esto se debe a que la caída que realiza el vehículo en un segundo es muy grande, por lo que es preciso re-calcularla y actuar sobre la dirección del vehículo con una mayor frecuencia. Esto se logra situando la parte del programa encargada de ajustar la dirección fuera del temporizador, de forma que se actualice cada vez que se ejecute el bucle principal. Las pruebas con este nuevo programa han permitido constatar su eficacia, ya que el coche efectua correcciones cada menos tiempo y los desvíos de la derrota inicial se reducen de forma notable.

En la tercera fase se implementa el uso de *arrays* de dos dimensiones para almacenar las coordenadas de los puntos a seguir. Para ello, se sustituye la declaración de las variables que almacenan la latitud y longitud de destino por un único *array* [60] de dos columnas y un número de filas igual al número de *waypoints* que se quiere que tenga la ruta.

Asimismo, se declara un contador que permita identificar el par de coordenadas de destino para cada tramo de la ruta, de forma que cada vez que se alcance el *waypoint* de destino se actualice el contador y el coche se dirija hacia el siguiente waypoint. Para la condición de parada se comprueba que este contador equivale al número de waypoints introducidos.

En la primera prueba de este programa, el robot actúa de la forma deseada, puesto que se dirige por si sólo hacia los waypoints independientemente de la posición de partida y cambia al siguiente waypoint tras llegar a cada uno de sus destinos intermedios, deteniéndose al final del recorrido. Sin embargo, las correcciones de rumbo del software siguen siendo excesivas, a pesar de actualizarse con mayor frecuencia. Esto es debido a que la dirección no está graduada (ni es posible calibrarla), sino que el vehículo gira de forma brusca independientemente de lo desviado que se encuentre de la dirección deseada.

Para solventar este problema y lograr que la ruta converja, minimizando a cada iteración el error de rumbo, se opta por mejorar el programa con una dirección gradual, que gire en función de lo desviado que se encuentre el coche de la ruta. Con esta mejora, el coche gira de forma brusca cuando se encuentra al rumbo opuesto que desea seguir, mientras que si la corrección es pequeña gira de forma suave.

El programa de dirección proporcional calcula y posiciona la dirección en función del rango de desvío del rumbo deseado. Como se vio anteriormente, es posible conocer la posición en la que se encuentra la dirección conociendo el valor de la resistencia del potenciómetro unido a esta, que se obtiene en forma digital mediante el convertidor analógico digital presente en la placa Arduino. El cálculo de la posición de la dirección correspondiente a cada desvío se realiza mediante la función *map* [61], que traslada un valor de un rango de valores a otro. En este caso, sabemos que el máximo desvío de la ruta es de 180 grados, y hemos obtenido los límites del valor del potenciómetro a izquierda y derecha. La función *map* nos permite, por tanto, pasar un valor que se encuentra entre -180 y 180 a un rango entre los límites del potenciómetro, que en este caso se han establecido entre 350 y 660. Se puede deducir por tanto que cuando el valor sea -180, es decir, cuando el coche se encuentre apartado al máximo hacia la derecha, la dirección girará al máximo a la izquierda hasta que adopte el giro correspondiente al valor de 350, y de igual forma en el sentido contrario. Por consiguiente, el valor 0

de diferencia debería centrar la dirección, aunque en este caso existe una diferencia entre el valor calculado y el correspondiente a la dirección centrada, por lo que se le aplica un factor de corrección.

La fórmula siguente muestra el cálculo realizado, siento *dif* la diferencia entre el rumbo del vehículo y el desado, *inmin* e *inmax* -180 y 180 respectivamente, *finmin* y *finmax* 350 y 660 y *valpot* el valor que ha de tener el potenciómetro de la dirección:

$$Valpot = \frac{(dif - inmin) \cdot (finmax - finmin)}{inmax - inmin} + finmin \quad (2)$$

Para posicionar las ruedas directrices en la posición deseada se sigue el diagrama de flujo mostrado en el apartado II del Apéndice I. Tras obtener el valor deseado del potenciómetro se mide el valor real del mismo y se comparan. Si el deseado es mayor que el real es debido a que la dirección está más a la izquierda de lo deseado, por lo que se gira a la derecha dentro de un bucle hasta que se encuentre en la posición adecuada. De forma pareja, si el valor real es mayor que el deseado la dirección debe posicionarse más a la izquierda. Para evitar que el programa busque corregir el rumbo constantemente con una precisión que el mecanismo no permita alcanzar, se aplica una cierta tolerancia al posicionado de las ruedas, lo que ayuda también a reducir el consumo eléctrico y la fatiga del motor al trabajar con constantes interrupciones y cambios de sentido.

3.1.2 Pruebas finales

La prueba del programa del coche radiocontrol se realiza en el Patio de Aulas de la Escuela Naval Militar, tras haber introducido previamente en el programa las coordenadas obtenidas en la aplicación en línea *Google Maps* [62].

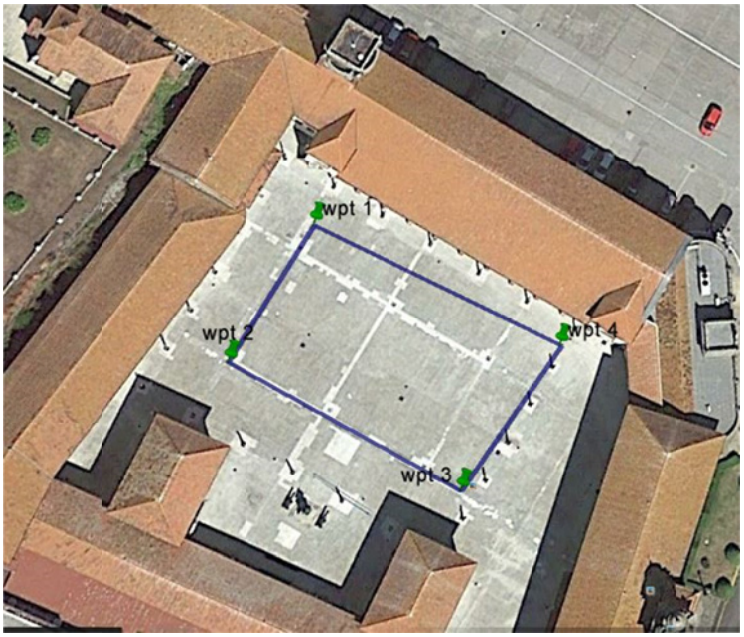


Figura 3-1 Ruta seguida en el Patio de Aulas

El resultado es satisfactorio, puesto que el coche recorre los *waypoints* uno a uno deteniéndose en cada cambio el tiempo programado y parándose en el punto final. Además, la dirección gradual hace que las caídas de rumbo sean grandes cuando el coche se encuentre apartado de la derrota y que

conforme se acerca al waypoint estas sean cada vez menores, de forma que la oscilación en el rumbo se amortigüe conforme avanza el coche. Con todo, se aprecian los siguientes puntos a mejorar:

- Por un lado, el vehículo no es capaz de avanzar perfectamente en línea recta. Esto se debe la dirección presenta una holgura notable y las ruedas no están perfectamente alineadas, por lo que las irregularidades del terreno hacen que ésta se mueva y no le permita mantener una línea recta una vez a rumbo. La solución del problema en este caso no es fácil, ya que requeriría cambiar el mecanismo de dirección de plástico con vástagos de alambre por uno metálico y con menores holguras. Sin embargo, esto no reviste problema a la hora de instalar el módulo en una lancha, ya que en este caso el juego sería notablemente menor.
- Por otro lado, la tolerancia en la corrección de la dirección, que es generosa a consecuencia de lo anterior, reduce la precisión en el guiado, conduce a que el coche pueda avanzar en una dirección hasta 10 grados apartada de la derrota, lo que puede suponer una distancia considerable si éste se encuentra muy alejado del siguiente *waypoint*. Cuando el vehículo se encuentra cerca del punto la variación de la demora al mismo en relación al avance del vehículo es mayor, y por tanto las correcciones son más frecuentes y el desvío es menor. Como este desvío es debido a la tolerancia de la dirección, si reducimos la holgura del sistema y por tanto la tolerancia de la dirección reduciremos también el apartamiento de la derrota deseada.

3.2 Pruebas en medio acuático

3.2.1 Adaptación del SW a las características de la lancha de radiocontrol

El programa empleado en la lancha, aunque en esencia trabaja igual que el del coche, requiere una serie de modificaciones por las siguientes razones:

- En el programa del coche, el control del rumbo se realiza a través del motor que actúa sobre las ruedas directrices, mientras que el motor de propulsión gira ininterrumpidamente hasta que llega al destino. En este caso, basta con comprobar el funcionamiento del motor de propulsión con la misma frecuencia con la que se comprueba la distancia, ya que hacerlo cada vez que se ejecuta el programa carece de sentido.
- El control de la lancha requiere actuar constantemente sobre el funcionamiento de los dos motores, ya que estos controlan tanto propulsión como dirección. En este caso se hace necesario controlar su funcionamiento cada vez que se actualice la corrección de rumbo que es preciso aplicar, y por ello se sitúa la función fuera del temporizador en el que se adquieren coordenadas y se calculan distancia y demora entre el siguiente waypoint y la posición actual.

En el Apéndice II: se muestra un diagrama de flujo del programa empleado en la lancha.

3.2.2 Validación y pruebas

Las pruebas en medio acuático tuvieron lugar en el Lago de Castiñeiras, ya que poseía las características idóneas: Sus dimensiones de 110 m de largo por 50 m de ancho son suficientes para probar la lancha con garantías y sin riesgo de perderla, no hay barcos o canoas con los que pueda colisionar y el agua dulce es menos corrosiva que el agua salada.

Se realizaron tres pruebas:

- La primera consistió en comprobar el manejo de la lancha mediante el mando de control remoto.
- La segunda prueba se realizó con un recorrido cercano a la orilla del lago para recuperar la lancha fácilmente en caso del posible fallo.

- La tercera prueba consistió en una vuelta alrededor del lago para comprobar la efectividad del programa final.

Figura 3-2 Ruta seguida en el lago Castiñeiras

Durante las pruebas se hicieron patentes las diferencias entre el movimiento en medio terrestre y en medio acuático. Así, se comprueba que la lancha cae de una forma mucho más lenta que el buggy y tiene un radio de giro mayor, por lo que las correcciones de la dirección son menores pero también le lleva más tiempo el alcanzar el rumbo deseado.

También se aprecia la inercia con la que cuenta la lancha, de forma que una vez parados los motores avanza unos metros hasta detenerse, a diferencia del coche, que se detiene de forma inmediata. Debido a esto, es aconsejable dejar un margen suficiente con la orilla en el último *waypoint* y realizar la aproximación final mediante el mando de radio control. En el día de la prueba existía una ventolina que hacía que la lancha abatiera cuando estaba parada. Con la lancha en movimiento, la corrección constante del rumbo hacía que el viento no le afectara de forma apreciable.

Las pruebas en el lago son satisfactorias, ya que la lancha ha cumplido lo que se le había programado: seguir una ruta preintroducida deteniéndose en cada *waypoint* y parando al final de la ruta, corrigiendo los desvíos sobre la misma. Este programa puede servir como punto de partida para proyectos más ambiciosos, algunos de los cuales se esbozan en el siguiente capítulo de la presente memoria

Figura 3-3 Extracto del vídeo de la prueba de la lancha

4 CONCLUSIONES Y LÍNEAS FUTURAS

4.1 Conclusiones

El objetivo inicial marcado en la memoria se ha cumplido, por lo que es razonable calificar el proyecto como exitoso. Con todo, a lo largo del mismo se han presentado diversos problemas:

Los sensores de distancia permiten obtener mediciones muy precisas, pero también dan lugar a mediciones erráticas que han de ser promediadas para obtener un valor fiable. Ésta es una de las razones por las que se ha descartado su uso para el proyecto final.

La alimentación de la placa Arduino afecta a su funcionamiento, especialmente cuando el consumo de los sensores y actuadores conectados es considerable. Así, se ha observado como programas que funcionan conforme a lo esperado con la placa conectada al ordenador o a una fuente de alimentación actúan de forma inesperada conectados a una pila que se encuentre a la mitad de su vida útil.

En relación a lo anterior, cabe destacar además la relación existente entre la calidad del firme y la energía necesaria para mover el coche y el sistema de dirección. En interiores y con un suelo liso alimentar los motores a 6 voltios hacía que el coche se moviera demasiado rápido, por lo que se redujo el voltaje a 3 voltios. En exteriores y con un suelo irregular, por el contrario, el motor de dirección era incapaz de mover las ruedas y en consecuencia hacía que el programa se detuviera, ya que el código para girar la dirección contiene un bucle *while*. Este problema se solventó volviendo a aumentar el voltaje de los motores a 6 voltios.

La posición obtenida por GPS presenta diversos errores: en interiores el error puede superar los 20 metros, y en exteriores es imprecisa hasta el momento en el que obtiene una posición fiable, que puede reducir el error a menos de tres metros. En el programa final se ha puesto como condición necesaria para avanzar el disponer de al menos 6 satélites, pero como se ha visto en el apartado 2.2 esto no es suficiente para garantizar la exactitud de la posición.

El control de la dirección de la plataforma tiene que ser muy estable para conseguir dirigir el rumbo del vehículo de forma precisa. La excesiva holgura en la dirección del vehículo, sumada a la baja precisión del potenciómetro, ha sido causa de desvíos en la ruta programada.

El microcontrolador utilizado tiene que actualizar el rumbo de forma rápida, por lo que resulta conveniente tener una capacidad de procesado mayor aunque el coste se incremente ligeramente. De lo contrario, el error en la posición de una iteración a la siguiente puede ser demasiado grande. Esto se puede hacer problemático en función de la velocidad del vehículo empleado.

A la hora de programar en Arduino resulta conveniente respectar el propósito de la función *loop*, que es funcionar de forma cíclica. Los bucles del tipo *while* o *for* situados dentro de esta función le

impiden continuar de forma natural y esto puede llevar a errores. En el caso de este programa, los bucles de este tipo presentes en los programas iniciales impedían actualizar las mediciones de datos GPS y datos de rumbo magnético, y por consiguiente el programa no funcionaba correctamente por no disponer de información actualizada.

Este proyecto ha permitido además obtener valiosas lecciones aprendidas, que permitirán agilizar el trabajo y la implementación de futuras mejoras. Algunas de ellas son:

- El conocimiento previo del entorno de programación es de gran ayuda a la hora de afrontar un proyecto. Muchos de los errores encontrados en la primera fase del proyecto son debidos a la inexperiencia, mientras que conforme se ha ido avanzando en el proyecto las deficiencias se corregían de forma más rápida y eficaz.
- Para identificar un error en el código resulta muy útil desglosarlo en pequeñas partes y comprobarlas una a una. De esta forma se puede confinar el problema en un área concreta, en vez de enfrentarse a un programa completo.
- Es conveniente hacer una lista comprobación en caso de errores comunes antes de descartar un programa. Algunos de ellos incluyen el no modificar el tipo de placa conectada dentro del programa, usar una velocidad de comunicación demasiado elevada, tener mal enchufado el cable USB o usar pilas descargadas, entre otros.
- El hábito de documentar y referenciar los pasos tomados conforme se van haciendo facilita mucho la redacción de la memoria. Una libreta siempre a mano permite realizar esquemas y apuntar las ideas que van surgiendo.
- El material que la comunidad Arduino sube desinteresadamente a la red es de gran ayuda y proporciona ideas y permite tomar muchos trabajos como punto de partida para futuras mejoras. No obstante, es preciso tener un espíritu crítico y emplear dicho material como una ayuda, no como una referencia infalible.
- Las gráficas y los esquemas facilitan la comprensión de los circuitos. Existen programas gratuitos, como el *Fritzing* [63], que permiten desarrollar esquemas eléctricos de conexionado de forma sencilla.

4.2 Líneas futuras

El límite para la expansión del presente proyecto se encuentra únicamente en la imaginación, ya que existen innumerables formas de mejorarlo o ampliarlo. Algunas de ellas son:

- **Detección de obstáculos mediante sensor LASER:** El corto alcance de los sensores de ultrasonidos (menos de 4 metros) hace que sean poco adecuados para ser empleado en el medio marino, en el que es necesaria mayor anticipación. Los sistemas RADAR son eficaces pero tienen un coste elevado y no es sencillo implementarlos, por lo que la opción más adecuada para la detección de obstáculos pasa por emplear telémetros LASER para detectar si hay objetos en la línea directa de avance del *USV*.
- **Implantación del módulo en una RHIB:** El siguiente paso para el desarrollo del módulo es aplicar el programa desarrollado a un vehículo de tamaño real. En este caso el código a emplear sería el del coche radiocontrol, puesto que al igual que éste hay que controlar avance y dirección.
- **Aro salvavidas seguidor de náufrago:** se podría adaptar la lancha para que localizara a un hombre al agua que disponga de un dispositivo GPS y se dirigiera hacia él llevándole un aro salvavidas y una radiobaliza que facilite su localización.
- **Programa para importar rutas GPS:** el método empleado para introducir los waypoints en el código es hacerlo uno a uno. Sería posible realizar un programa que permite importar

rutas en formato .gpx u otro de forma que se facilite la forma de introducir la ruta en el *USV*.

- **Programa de vigilancia con imágenes asociadas a la ruta:** el módulo GPS dispone también de la posibilidad de almacenar en una tarjeta *microSD* la ruta seguida por el vehículo. Si se le instala una cámara de vigilancia al vehículo se podría conocer la hora y posición de las imágenes obtenidas.
- **Implantación de una ruta de retorno:** en caso de fallo de algún o falta de combustible es conveniente que el USV vuelva de forma automática a un punto prefijado para ser recuperado.
- **Comunicaciones en tiempo real:** es posible dotar al módulo de diferentes formas de comunicación con el medio exterior. Estas formas incluyen WiFi, radifrecuencia o red GSM de telefonía móvil, entre otras. Mediante este enlace se podría modificar la ruta seguida por la lancha en tiempo real o darle orden de regresar al punto de partida. También se podría configurar el módulo de forma que emita una señal de socorro en caso de no poder volver por sus propios medios al punto de recogida.
- **Interruptor de parada de emergencia:** en el caso de instalar el módulo en una embarcación de dimensiones considerables es conveniente instalar un interruptor capaz de pararla de forma remota en caso de que exista riesgo de abordaje o se encuentra cerca de un náufrago al que pueda herir con las hélices. Este interruptor puede ser accionado por radiofrecuencia u otro método, y es aconsejable que funcione independientemente del módulo de control, cortando, por ejemplo, la alimentación de combustible.

5 REFERENCIAS

[1] *The Navy Unmanned Surface Vehicle (USV) Master Plan (23 julio 2007*) [online] Enlace: http://www.navy.mil/navydata/technology/usvmppr.pdf

[2] *Infantes de Marina del Tercio de Armada (TEAR) y de la Fuerza de Guerra Naval Especial (FGNE) se forman para operar sistemas pilotados a control remoto (RPAS)* [online] Enlace: http://www.armada.mde.es/ArmadaPortal/page/Portal/ArmadaEspannola/conocenos_noticias/prefLang_es/00_noticias--2016--01--NT-014-DRON-TEAR_es?_selectedNodeID=2446009&_pageAction=selectItem

[3] *Amazon podrá repartir paquetes con drones | EL MUNDO* [online] Enlace: http://www.elmundo.es/tecnologia/2015/03/20/550c3ecfca474124708b4572.html

[4] *MUNIN – Maritime Unmanned Navigation through Intelligence in Networks* [online] Enlace: http://www.unmanned-ship.org/munin/

[5] *MUNIN-final-brochure* [online] Enlace: http://www.unmanned-ship.org/munin/wp-content/uploads/2016/02/MUNIN-final-brochure.pdf

[6] *Liquid Robotics announces next generation of Wave Glider sea-going robot* [online] Enlace: http://www.gizmag.com/liquid-robotics-wave-glider-sv3-aquatic-robot/26977/

[7] *Liquid Robotics* [online] Enlace: http://www.liquid-robotics.com/index.html

[8] *First of four autonomous Wave Glider robots successfully crosses Pacific* ocean [online] Enlace: http://www.gizmag.com/wave-glider-arrives-australia/25316/

[9] *Autonomous Surface Vehicles (ASV) Ltd* [online] Enlace: http://asvglobal.com/

[10] *Rafael PROTECTOR Unmanned Naval Patrol Vehicle* [online] Enlace: http://www.rafael.co.il/Marketing/288-1037-en/Marketing.aspx

[11] *Textron Systems Common Unmanned Surface Vehicle* [online] Enlace: http://www.textronsystems.com/products/unmanned/cusv

[12] *Boletín de Observación Tecnológica en Defensa Nº 24.* [online] Enlace: http://www.defensa.gob.es/Galerias/documentacion/ficheros/DGT-doc-boletin24.pdf

[13] *El SIRAMICOR de Navantia se queda en puerto por falta de presupuesto* [online] Enlace: http://www.infodefensa.com/es/2013/04/26/noticia-el-siramicor-de-navantia-se-queda-en-puerto-por-falta-de-presupuesto.html

[14] *Arduino Powered Autonomous Vehicle* [online] Enlace: http://www.instructables.com/id/Arduino-Powered-Autonomous-Vehicle/?ALLSTEPS

[15] *Mini GPS Autonomous Car for Under $*100 [online] Enlace: http://ttjcrew.com/2009/07/20/mini-gps-autonomous-car-for-under-100/

[16] *Whirlgig* [online] Enlace: http://letsmakerobots.com/files/whirligig.pdf

[17] *SEA RENDERING* [online] Enlace: http://letsmakerobots.com/node/21098

[18] *Boat Autopilot Based on the Arduino* [online] Enlace: http://www.instructables.com/id/Boat-Autopilot/?ALLSTEPS

[19] *Whirligig* [online] Enlace: http://letsmakerobots.com/node/16995

[20] *Lajara, J. and Pelegrí Sebastiá, J. (2014). Sistemas integrados con Arduino. Barcelona: Marcombo.*

[21] *Open processing* [online] Enlace: http://www.openprocessing.org/

[22] *What is an Arduino?* [online] Enlace: https://learn.sparkfun.com/tutorials/what-is-an-arduino

[23] *Arduino – PWM* [online] Enlace: https://www.arduino.cc/en/Tutorial/PWM

[24] *Analog reference* [online] Enlace: https://www.arduino.cc/en/Reference/AnalogReference

[25] *Arduino – Libraries* [online] Enlace: https://www.arduino.cc/en/Reference/Libraries

[26] *Arduino Board UNO* [online] Enlace: https://www.arduino.cc/en/Main/ArduinoBoardUno

[27] *ATmega48A/PA/88A/PA/168A/PA/328/P* [online] Enlace: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

[28] *Arduino – Memory [online] Enlace: https://www.arduino.cc/en/Tutorial/Memory*

[29] *Arduino Analog Read* [online] Enlace: https://www.arduino.cc/en/Reference/AnalogRead

[30] *I2C* [online] Enlace: https://learn.sparkfun.com/tutorials/i2c

[31] *Arduino Software Serial* [online] Enlace: https://www.arduino.cc/en/Reference/SoftwareSerial

[32] *Arduino Board Mega 2560* [online] Enlace: https://www.arduino.cc/en/Main/ArduinoBoardMega2560

[33] *Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf* [online] Enlace: http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

[34] *What is a breakout board for Arduino* [online] Enlace: https://programmingelectronics.com/what-is-a-breakout-board-for-arduino/

[35] *arduino-compatible-mega-motor-shield* [online] Enlace: http://www.robotshop.com/media/files/images/arduino-compatible-mega-motor-shield-1a-5-28v-large.jpg

[36] *Light sensor using Arduino* [online] Enlace: http://rookieelectronics.com/wp-content/uploads/2012/07/DSCN4516.jpg

[37] *Adafruit Ultimate GPS Logger Shield* [online] Enlace: https://learn.adafruit.com/downloads/pdf/adafruit-ultimate-gps-logger-shield.pdf

[38] Moreu Curbera, J. and Martínez Jiménez, C. *(1972).* Astronomía y navegación. *[Madrid. IMPOSA].*

[39] *GPS Space Segment* [online] Enlace: http://www.gps.gov/systems/gps/space/

[40] *What is GPS?* [online] Enlace: http://www8.garmin.com/aboutGPS/

[41] *Arduino Robot* [online] Enlace: https://www.arduino.cc/en/Main/Robot

[42] *Arduino* Robot [online] Enlace: http://www.robotshop.com/media/files/images/arduino-2wd-robot-us-large.jpg

[43] *Buggy toy car* [online] Enlace: http://www.vw-buggy.at/model-*cars/manufacturer/buggy/573*

[44] *Funcionamiento del motor de corriente continua* [online] Enlace: https://www.youtube.com/watch?v=gTCfcGW_mWs

[45] *Sainsmart 4-Channel 5V Relay Module* [online] Enlace: http://www.sainsmart.com/4-channel-5v-relay-module-for-pic-arm-avr-dsp-arduino-msp430-ttl-logic.html

[46] *Reles_motor* [online] Enlace: http://mural.uv.es/palmaro/imagenes/reles_motor.gif

[47] *L293x Quadruple Half-H* Drivers [online] Enlace: http://www.ti.com/lit/ds/symlink/l293.pdf

[48] *Multiple Motor H-Bridge* [online] Enlace: http://electronics.stackexchange.com/questions/207319/multiple-motor-h-bridge

[49] *Ultrasonic Ranging Module HC* - SR04 [online] Enlace: http://www.micropik.com/PDF/HCSR04.pdf

[50] *Tutorial Sensor Ultrasónico HC-SR04 y Arduino* [online] Enlace: http://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-y-arduino/

[51] *GlobalTop-FGPMMOPA6H-Datasheet-V0A-Preliminary* [online] Enlace: https://www.adafruit.com/datasheets/GlobalTop-FGPMMOPA6H-Datasheet-V0A.pdf

[52] *Serial Communicacion* [online] Enlace: https://en.wikipedia.org/wiki/Serial_communication

[53] *Universal asynchronous receiver/transmitter* [online] Enlace: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

[54] *MPU-9250 Product Specification* [online] Enlace: http://store.invensense.com/datasheets/invensense/MPU9250REV1.0.pdf

[55] *3-Axis Digital Compass IC* HMC5883L [online] Enlace: https://www.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf

[56] *AMEWI - RC Modellbau* [online] Enlace: http://www.amewi.com/marken/amewi/boote.html?showall=&start=5

[57] *Self reversing screws* [online] Enlace: http://www.abssac.co.uk/p/Self+reversing+screws/8/#.VsDK8PnhDIU

[58] *GPS ACTIVE 28dB MAGNETIC ANTENNA* [online] Enlace: https://www.adafruit.com/datasheets/GPS-01.pdf

[59] *NMEA data [online]* Enlace: http://www.gpsinformation.org/dale/nmea.htm

[60] *Tipos de datos: arrays en Arduino* [online] Enlace: http://panamahitek.com/tipos-de-datos-arrays-en-arduino/

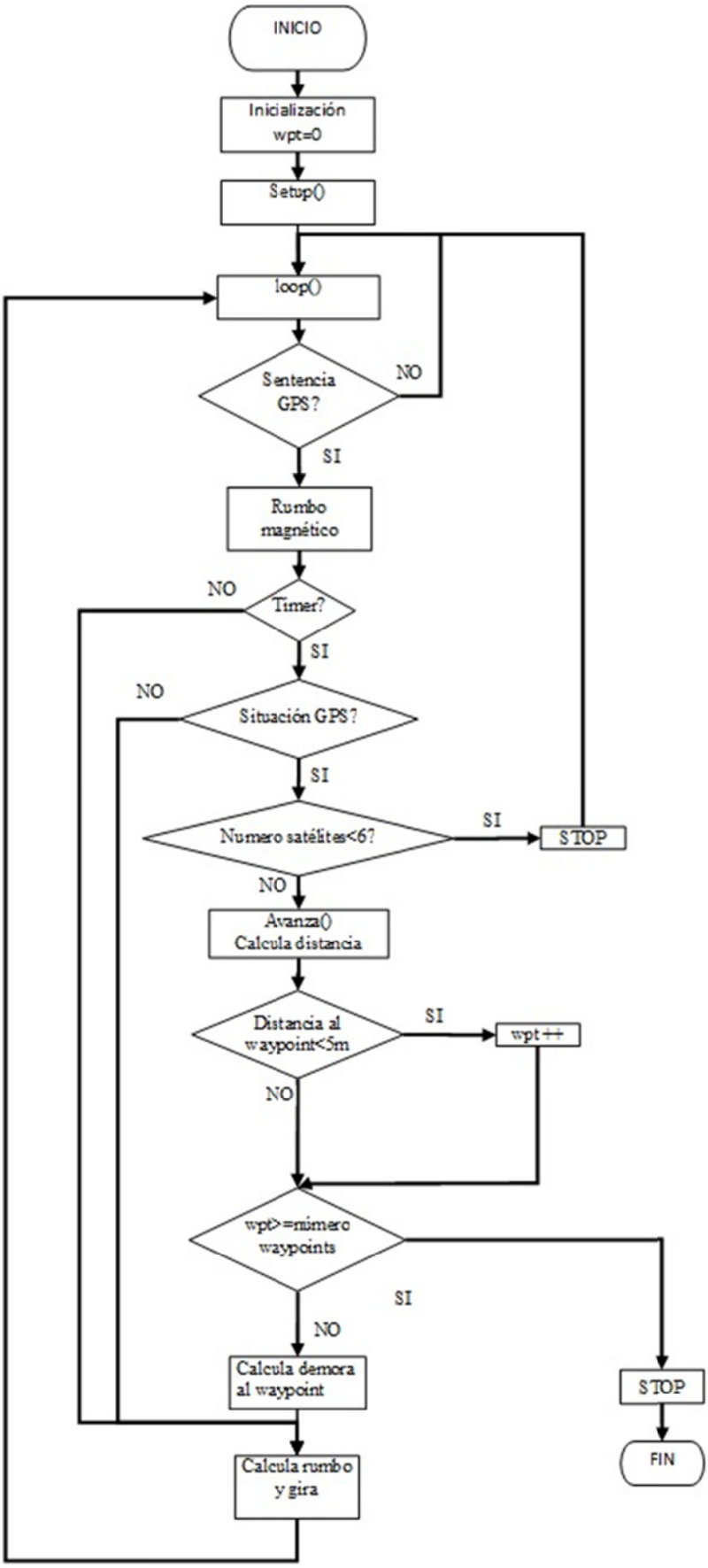
[61] *Arduino – map* [online] Enlace: https://www.arduino.cc/en/Reference/Map

[62] *Google Maps* [online] Enlace: https://www.google.es/maps

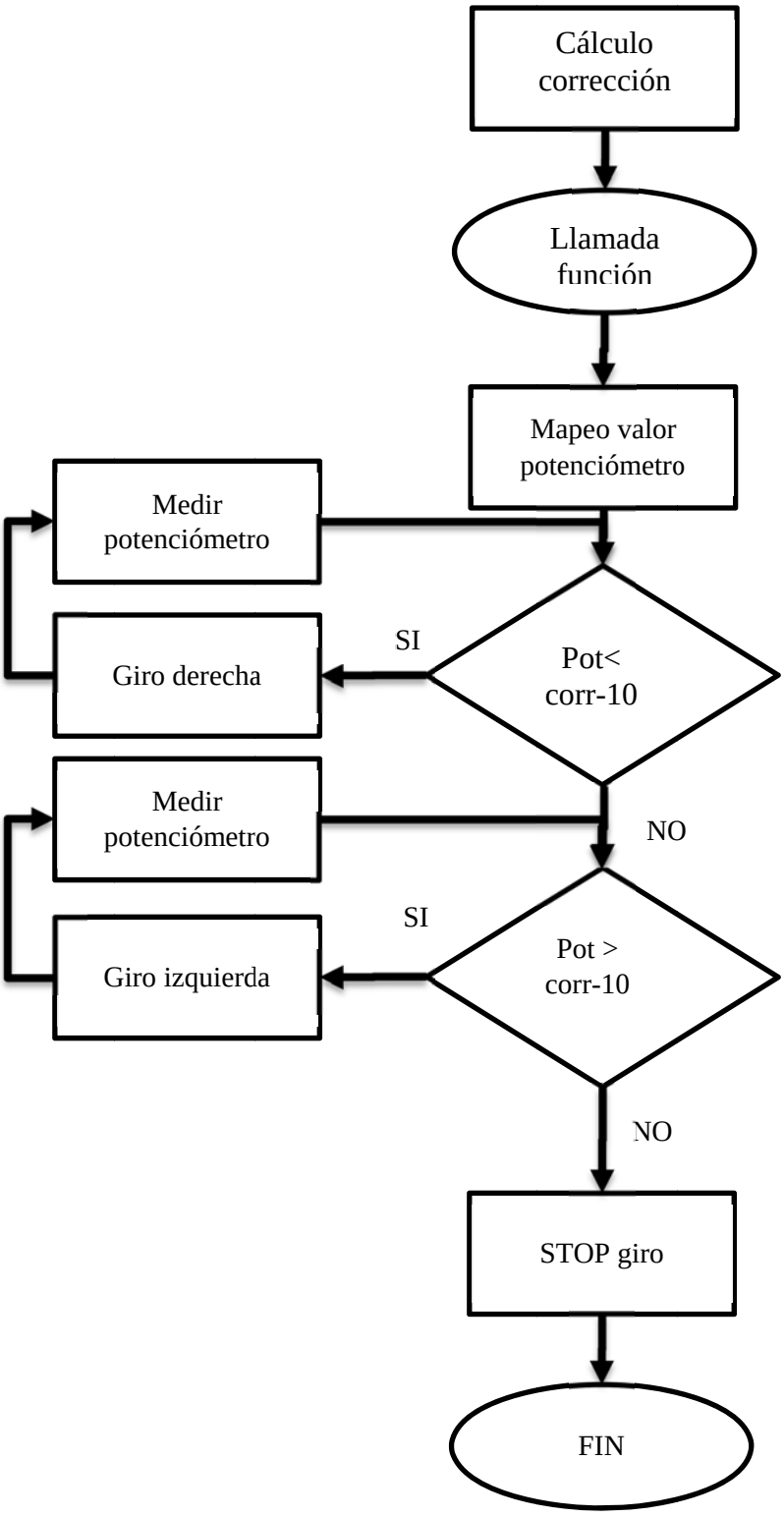
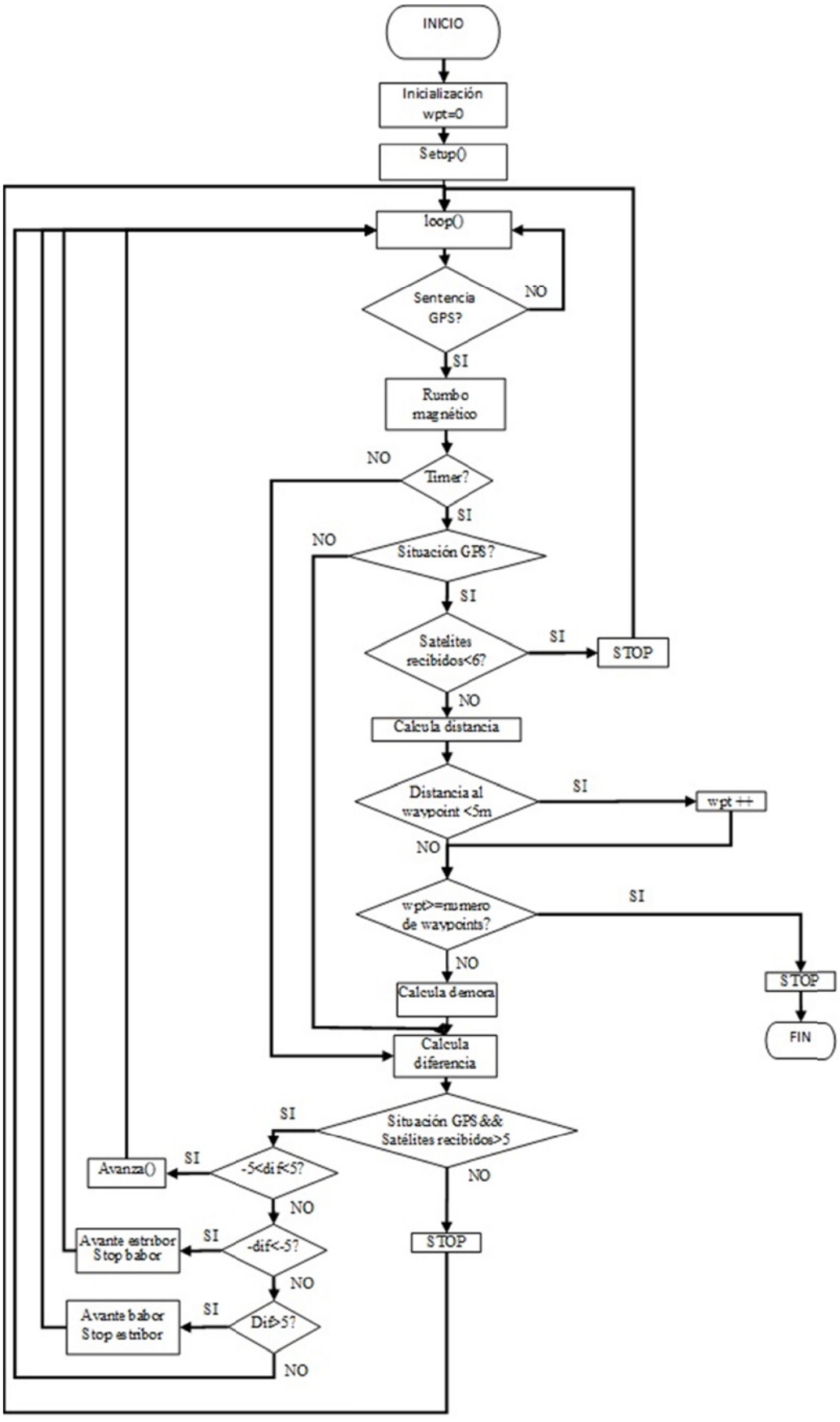
[63] *Fritzing* [online] Enlace: http://fritzing.org/home/

APÉNDICE I: DIAGRAMAS DE FLUJO DEL COCHE

I) Diagrama del programa de navegación



II) **Diagrama de la dirección proporcional**



APÉNDICE II: DIAGRAMA DE FLUJO DE LA LANCHA