UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

Grado en Ingeniería Electrónica Industrial y Automática

VEHÍCULO MARINO DE SUPERFICIE PARA EL APOYO A EMBARCACIONES DE PESCA CON NASAS

Memoria y Anexos

Autores: Antonio Gonzalez Cifuentes

Álvaro Martín Velasco

Director: Spartacus Gomáriz Castro

Convocatoria: Octubre 2019

Agradecer, en primer lugar, a todos los profesores que, a lo largo de estos meses, han podido aconsejarnos y darnos herramientas para poder llevar a cabo este proyecto y, en especial, a nuestro tutor Spartacus, que nos ha guiado en la ideación y corrección de este trabajo.

Por último, agradecer a familiares, parejas, amigos, y compañeros, que nos han ayudado y apoyado incondicionalmente a lo largo de estos años y especialmente en los meses de la elaboración del proyecto.

ÍNDICE

RESUMEN **3**

RESUM **4**

ABSTRACT **5**

1. MARCO TEÓRICO **7**

1.1. INTRODUCCIÓN 7
1.2. MOTIVACIÓN 7
1.3. OBJETIVO 8
1.4. ESTADO DEL ARTE 8
1.5. ALCANCE 9
1.6. SITUACIÓN ACTUAL 12

2. VEHÍCULO DE APOYO **15**

2.1. DISEÑO DEL VEHÍCULO 15
2.2. LEGISLACIÓN VIGENTE 19

3. DISEÑO MECÁNICO **21**

3.1. CASCOS 22
3.1.1. Modelaje *25*
3.2. CAJA ESTANCA 28
3.3. MONTAJE Y ENCAPSULADO DE MOTORES Y EJE MOTOR 31
3.4. ENSAMBLAJE DEFINITIVO 34

4. HARDWARE **38**

4.1. PLACA ARDUINO® MEGA 2560 39
4.2. REGULADORES DE TENSIÓN LM7812, LM7809 Y LM7805 40
4.3. SENSOR DE TEMPERATURA Y HUMEDAD DHT11 41
4.4. ACELERÓMETRO ADXL335 41
4.5. BRÚJULA HMC5883L 42
4.6. SENSOR DE CORRIENTE ACS712 42
4.7. GPS NEO 6M 43
4.8. DRIVER MDD10A 43
4.9. MÓDULO DE COMUNICACIÓN INALÁMBRICA XBEE™ 44
4.10. MOTORES 44
4.11. BALIZA Y BOCINA 45
4.12. MÓDULOS SOLDADOS 45

5. SOFTWARE **50**

5.1. COMUNICACIÓN ARDUINO® - LABVIEW™ 51
5.2. ENSAYOS 56
5.2.1. Control de Relés *56*
5.2.2. Consumo de Motores *58*
5.2.3. Temperatura y Humedad *60*
5.2.4. Acelerómetro *63*
5.2.5. Brújula *67*
5.2.6. Sensor de Corriente *69*
5.2.7. GPS *70*

5.3. COMUNICACIÓN LABVIEW™ - ARDUINO® 72
5.3.1. Control de Motores 76
5.3.2. Comunicación Inalámbrica: XBee™ 77
5.4. CONTROL DE TRAYECTORIA 79
5.5. APLICACIÓN FINAL 82
5.5.1. Control 82
5.5.1.1. Cajas de escritura, tratamiento y lectura 83
5.5.1.2. Adaptación y configuración 86
5.5.2. Registro 91
5.5.3. Interfaz 93

6. ESTUDIO ENERGÉTICO Y PRUEBAS EXPERIMENTALES 94

6.1. ENSAYO DE REGULADORES 95
6.1.1. Regulador LM7812 95
6.1.2. Regulador LM7809 96
6.1.3. Regulador LM7805 97
6.2. ENSAYO DE MOTORES 99
6.2.1. Ensayo a 5 V 99
6.2.2. Ensayo a 7 V 103
6.2.3. Ensayo a 9 V 106
6.3. BATERÍA 109

7. PRUEBAS DE CAMPO 111

7.1. TEST 1 111
7.1.1. Flotabilidad 111
7.1.2. Aplicación 113
7.2. TEST 2 114

8. IMPACTO MEDIOAMBIENTAL DE MATERIALES 117

8.1. BATERÍA PLOMO-ÁCIDO 117
8.2. FIBRA DE VIDRIO Y RESINA DE POLIÉSTER 117

9. PRESUPUESTO 119

9.1. INVERSIÓN REALIZADA 119
9.2. COSTE REAL 122

10. POSIBLES MEJORAS 124

11. CONCLUSIÓN 127

12. BIBLIOGRAFÍA 129

13. ANEXOS 131

RESUMEN

La pesca con nasas se realiza a nivel mundial en diferentes regiones en época invernal y es considerada una de las profesiones más arriesgadas y peligrosas del mundo.

Este tipo de pesca se realiza completamente a ciegas, ya que se tiran las nasas y se recogen al cabo de 12-15 horas sin saber si se habrá pescado algún cangrejo o no, con lo que supone los gastos de hacer salir la embarcación para el reconocimiento de las nasas, manteniendo así al barco y a la tripulación sujetos a los riesgos de la pesca en alta mar.

El presente documento técnico recoge el método de desarrollo de un prototipo de vehículo de apoyo funcional basado en la plataforma ARDUINO®, ideado para ayudar en este tipo de pesca. En éste, se recoge el diseño mecánico, eléctrico y electrónico junto a los ensayos y pruebas experimentales realizados, así como el resultado de la implementación de la aplicación desarrollada en el prototipo final.

La principal función del vehículo es la de facilitar la búsqueda y simplificar la función de los trabajadores de la embarcación, dándoles información en tiempo real de la situación de cada nasa y de la necesidad o no de la recogida de éstas mediante una aplicación desarrollada en LabVIEW™ comunicada ininterrumpidamente con el sistema de ARDUINO® del vehículo, aportando a su vez un manual de instrucciones de la aplicación para el usuario.

RESUM

La pesca amb nanses es realitza a nivell mundial en diferents regions en època hivernal i és considerada una de les professions més arriscades i perilloses del món.

Aquest tipus de pesca es realitza completament a cegues, ja que es tiren les nanses i es recullen al cap de 12-15 hores sense saber si s'haurà pescat algun cranc o no, amb el que suposa les despeses de fer sortir l'embarcació per al reconeixement de les nanses, mantenint així al vaixell i a la tripulació subjectes als riscos de la pesca a alta mar.

El present document tècnic recull el mètode de desenvolupament d'un prototip de vehicle de suport funcional basat en la plataforma ARDUINO®, ideat per ajudar en aquest tipus de pesca. En aquest, es recull el disseny mecànic, elèctric i electrònic al junt als assajos i proves experimentals realitzats, així com el resultat de la implementació de l'aplicació desenvolupada en el prototip final.

La principal funció del vehicle és la de facilitar la recerca i simplificar la funció dels treballadors de l'embarcació, donant-los informació en temps real de la situació de cada nansa i de la necessitat o no de la recollida d'aquestes mitjançant una aplicació desenvolupada en LabVIEW™ comunicada ininterrompudament amb el sistema de ARDUINO® del vehicle, afegint a la vegada un manual d'instruccions de l'aplicació per a l'usuari.

ABSTRACT

Pot fishing is performed worldwide in different regions during the winter season and is considered one of the riskiest and most dangerous professions in the world.

This type of fishing is carried out completely blindly, since the pots are thrown and collected after 12-15 hours without knowing if any crab will have been caught or not, which means the costs of getting the boat out for pot recognition, thus keeping the boat and crew subject to the risks of deep-sea fishing.

This technical document includes the method of developing a functional support vehicle prototype based on the ARDUINO® platform, designed to help in this type of fishing. In this, the mechanical, electrical and electronic design is collected together with the experimental tests performed, as well as the result of the implementation of the application developed in the final prototype.

The main function of the vehicle is to facilitate the search and simplify the function of the ship's workers, giving them real-time information on the situation of each pot and the need or not for the collection of these through an application developed in LabVIEW™ communicated continuously with the ARDUINO® system of the vehicle, also providing an instruction manual of the application for the user.

1. MARCO TEÓRICO

1.1. Introducción

El presente documento recoge la memoria técnica sobre el Trabajo de Fin de Grado en Ingeniería Electrónica Industrial y Automática, realizado por Antonio Gonzalez Cifuentes y Álvaro Martín Velasco a lo largo del último año de carrera y dirigido durante el transcurso de éste por Spartacus Gomáriz Castro.

Como su propio título indica, este trabajo se centra en el diseño e implementación de un vehículo marino de superficie para el apoyo a embarcaciones de pesca con nasas basado en la plataforma ARDUINO®.

1.2. Motivación

Desde nuestro punto de vista todos los trabajos de fin de grado a los cuales hemos tenido acceso o sobre los cuales nos hemos informado respecto a su funcionalidad nos parecen completamente aptos y suficientemente originales como para poderse defender ante una rigurosa evaluación, pero nosotros desde un principio hemos tenido claro que el concepto de "implementación" era totalmente necesario para poder sentirnos motivados frente al desarrollo del trabajo. Con una forma tangible sobre la cual pudiésemos ver el resultado del conocimiento adquirido a lo largo de nuestros estudios podríamos afrontar de una manera más apasionada la laboriosa realización del proyecto.

En un primer lugar debatimos sobre cuál debía ser la funcionalidad del proyecto teniendo por supuesto que el colofón de la implementación reside en un vehículo móvil. Tanteando entre tierra, agua y aire como posibles terrenos de desarrollo, acabamos optando por el mar, puesto que, según nuestro criterio en el momento de la elección, el agua como elemento de desarrollo no está tan explotado a nivel de funcionalidad como pueden ser la tierra o el aire[1],además de ser el elemento que nos permite unificar en uno solo el concepto del desplazamiento en dos direcciones propio de los vehículos teledirigidos terrestres, así como el concepto de vacuidad del espacio aéreo, el cual le permite a los drones la libertad de movimiento.

Escogido el medio, solo resta la funcionalidad. Algunas de las opciones sopesadas han sido las del apoyo para el rescate de náufragos, la recogida de productos tóxicos o el control de magnitudes tales como la salinidad o temperatura de piscifactorías

[1] Desde hace décadas, los vehículos por radio control terrestres han sido utilizados en diversos campos, desde el ocio hasta la guerra. Del mismo modo, el auge por el uso de drones aéreos para diferentes sectores ha llevado una tendencia alcista durante los últimos años.

naturales. Todas ellas fueron descartadas frente al apoyo para embarcaciones de pesca con nasas.

La propuesta nos pareció atractiva debido al nicho concreto sobre el cual nos estábamos centrando que, además de aunar conocimientos más que asimilados, como el diseño eléctrico o electrónico, con otros sobre los cuales se ha ahondado menos como pueden ser el campo de las comunicaciones o el diseño de interfaces de control.

1.3. Objetivo

Dentro del marco teórico del proyecto, se pretende introducir al lector en la situación actual de la pesca en alta mar en general, concretamente la pesca que requiere del uso de nasas con tal de acompañarle en reparar en la necesidad del desarrollo de una aplicación real y funcional que apoye el ejercicio de esta. Del mismo modo, se pretende establecer y delimitar tanto el conocimiento necesario para realizar el diseño e implementación del prototipo como la jurisdicción que rige el uso de vehículos de superficie marítima.

Una vez expuesto el problema, evaluar las posibles soluciones y desarrollar un prototipo final el cual cumpla con las necesidades impuestas dentro del marco teórico son el segundo objetivo principal del proyecto. Estas necesidades condicionan al vehículo a ser autónomo y funcional, del mismo modo debe encargarse de recibir datos de pesca y enviarlos a un receptor situado en la embarcación tripulada con tal de poder llevar el control y registro de toda la información y, desde la cual, se podrá gobernar el propio vehículo de apoyo desarrollado.

1.4. Estado del Arte

La pesca con nasas se realiza a nivel mundial en diferentes regiones en época invernal y es considerada una de las profesiones más arriesgadas y peligrosas del mundo.

La pesca con nasas, se realiza completamente a ciegas, ya que se tiran las nasas y se recogen al cabo de 12-15 horas sin saber si se habrá pescado algún cangrejo o no, con lo que supone los gastos de hacer salir la embarcación para el reconocimiento de las nasas manteniendo así al barco y la tripulación sujetos a los riesgos de la pesca en alta mar. Los gastos tanto en tripulación como en combustible, así como poner en riesgo la embarcación, sin tener unos beneficios asegurados, hace de esta práctica una tarea muy complicada para los pescadores.

Otro punto muy interesante es que los cangrejos no pueden estar más de un cierto tiempo en la nasa, ya que acaban enfermando y pueden envenenar a todos los demás, haciéndolos no aptos para el consumo humano. Un único cangrejo muerto puede

ocasionar la pérdida de toda la captura debido a las toxinas emanadas. El agua muy fría o caliente, o la escasa circulación de la misma en el interior de los contenedores de las embarcaciones, representan un riesgo potencial de muerte para todos los cangrejos. De hecho, el agua estancada puede matarlos más rápidamente como el hecho de abandonarlos fuera del agua.

La propuesta de nuestro proyecto es intentar poner solución a esta problemática.

En cuanto a la situación actual de los drones para pesca, encontramos una gran variedad de drones utilizados en pesca deportiva. Este tipo de drones, se centran en tareas como la ubicación de cebos, o el movimiento de los mismos. Encontramos drones tanto aéreos (en más cantidad) como de superficie capaces de realizar dichas tareas.

Por otro lado, también encontramos drones tanto submarinos como de superficie, que se utilizan a nivel científico, haciendo análisis en diferentes campos tanto en mar como en ríos. Este tipo de drones también son utilizados con fines de exploración o de inspección de diferentes elementos, como cascos de barcos o zonas subacuáticas poco accesibles.

Centrándonos en nuestra propuesta un vehículo de superficie con estructura de catamarán, encontramos varios en el mercado, tanto a nivel de recreo, como a nivel más profesional con fines más específicos.

En cuanto a drones de recreo [1], se pueden hallar pequeños prototipos sin capacidades especificas más que la navegación a precios relativamente bajos (alrededor de 100€).

A nivel profesional, encontramos drones capaces de navegar durante horas y con funciones específicas, como el reconocimiento de bancos de peces, la inspección subacuática o misiones de vigilancia. Estos vehículos tienen costes mucho más elevados, partiendo de precios de alrededor de los 1000€, como por ejemplo el dron Superbathy[2].

1.5. Alcance

El alcance del proyecto es la implementación total del vehículo de apoyo diseñado junto con el software necesario para el correcto funcionamiento de este.

El vehículo es el encargado de recibir mediante un receptor los datos enviados desde la nasa a través de la boya. La nasa, dotada de un sensor láser que registra la entrada de cangrejos mediante pulsos, envía la información a la boya situada en su misma vertical y fijada mediante una cadena metálica para evitar desplazamientos inoportunos sobre

[2] Dron catamarán de superficie que se encuentra actualmente en el mercado de la empresa Heliceo [2].

la superficie marítima debido al oleaje. La comunicación nasa-boya se llevará a cabo mediante un sistema emisor/receptor acústico[3]. Desde la boya, la información será enviada hasta el vehículo de apoyo cuando éste se encuentre en el rango de recepción. Finalmente, el vehículo, que podrá ser pilotado (o no) desde la embarcación tripulada, deberá enviar la información de los pulsos recibida de las nasas a la embarcación tripulada desde la cual se tomaran las medidas pertinentes en función de la información. La siguiente figura refleja el funcionamiento general del complejo:

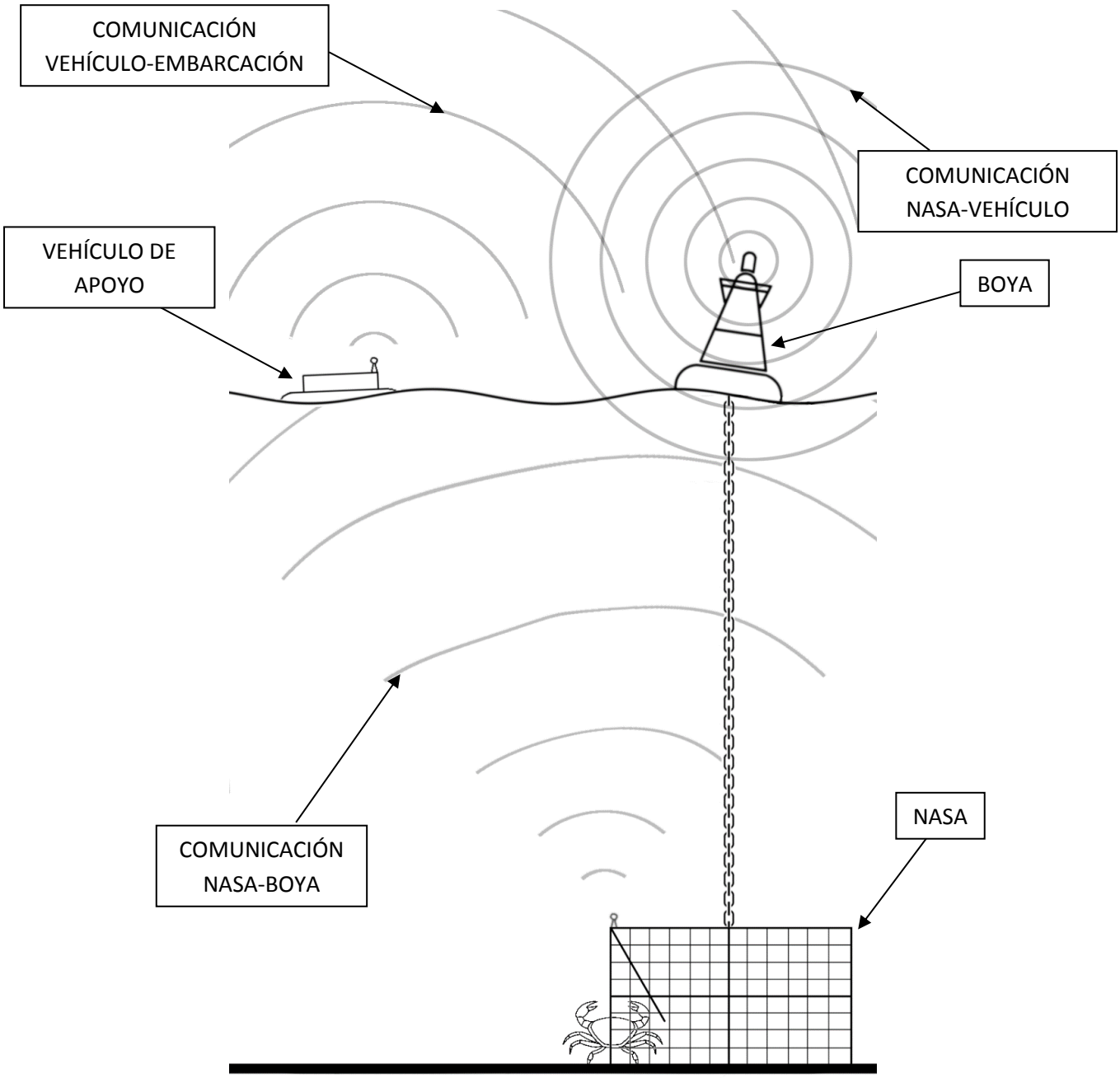


Figura 1.- Esquema general del proyecto

[3]Las comunicaciones subacuáticas pueden basarse tanto en señales ópticas como en ondas electromagnéticas o acústicas. Éstas últimas son las más versátiles.

El principal adversario a combatir en la implementación del prototipo es el presupuesto. Nuestro diseño consta de partes mecánicas que requieren de ciertos materiales para su funcionamiento óptimo, un posible ejemplo son los cascos, los cuales preferiblemente deben estar hechos de fibra de carbono o en su defecto de fibra de vidrio. Del mismo modo, algunos elementos externos tales como balizas, sirenas, etc. requieren de una inversión suficientemente considerable como para suponer un problema en el caso de requerir más de una, igualmente el software necesario para la realización de este nos es limitante, puesto que usar nuevas versiones del software proporcionado por la universidad o utilizar otro tipo de software de pago suele suponer un coste elevado.

Por lo tanto, sin intención de ser austeros, pero teniendo como realidad el adversario de un presupuesto modesto, nos es necesario prescindir de la comunicación nasa-boya-vehículo[4], por lo tanto, ni el diseño ni la implementación de la nasa ni la boya serán un pilar en los cuales se fundamente este proyecto. Así pues, el trabajo se centrará única y exclusivamente en el diseño e implementación del vehículo de apoyo y del sistema de comunicación que deberá tener con la embarcación tripulada desde la cual se gestionará la información. Esta restricción nos sirve de contexto para darle profundidad a los aspectos anteriormente enumerados, es decir, la incapacidad de poder realizar el sistema nasa-boya, nos sirve de precedente para diseñar un vehículo completamente funcional, y equipado con las medidas de sensado y de transmisión de información más complejos y una interfaz más completa a los que se podrían haber desarrollado en el hipotético de haber podido diseñar y/o implementar la totalidad del sistema.

A modo de resumen, existe un abanico de ventajas las cuales aportaría el uso del prototipo implementado. Entre ellas, las más relevantes son:

- La tripulación en cubierta queda reducida: como se ha comentado en los primeros puntos del documento, la pesca del cangrejo real en el Mar de Bering resulta ser uno de los oficios más peligrosos del planeta. El peligro no queda en sí reducido con la implementación de un vehículo de apoyo, pero si se puede reducir la tripulación que se encuentre expuesta a este riesgo.

  En este punto ya no será necesaria una tripulación con el conocimiento de los tiempos los cuales se deben mantener las nasas en el fondo marino ni con la necesidad de estar retirándolas y devolviéndolas continuamente.

[4] Los precios de emisores y receptores acústicos (necesarios para la comunicación bajo el mar) rondan los 400 €, además de requerir de un mantenimiento que prevenga la aparición de flora y fauna del fondo marino, así como elementos con IPs muy elevadas.

- Conocimiento de la situación en todo momento: el principal problema de este tipo de pesca es que esta se realiza completamente a ciegas, ya que se tiran las nasas y se recogen al cabo de 12-15 horas sin saber si se habrá pescado algún cangrejo o no. Llevando el registro de la situación, se puede erradicar este problema.
- Ratio beneficio/coste aumentado: al recoger las nasas sabiendo la cantidad de cangrejos los cuales se encuentran dentro de ellas, se deja de malgastar recursos en hacer salir la embarcación para el reconocimiento de las mismas. Por lo tanto, las nasas pueden ser recogidas en el momento en el que el capitán sea conocedor del beneficio que puede aportar esa pesca.
- Temporadas más cortas: incrementando el rendimiento, se podrían realizar temporadas más cortas llegando antes a los beneficios estándar por temporada. Reduciendo así el tiempo el cual la tripulación se pueda encontrar en situaciones de riesgo.

1.6. Situación Actual

La pesca del cangrejo real en los meses invernales en el Mar de Bering, es considerada una de las profesiones más peligrosas sino la que más del mundo, debido principalmente a las condiciones extremas de frío y oleaje en que se desarrolla el trabajo. Los barcos pesqueros generalmente utilizados para esta actividad miden entre 17,70 m y 70 m de eslora, la tripulación está conformada por un capitán y entre tres y nueve marineros. Los principales tipos de cangrejos que son pescados en esta región son dos:

- Cangrejo rojo real (*Paralithodes camtschaticus):* también conocido como cangrejo rojo gigante, es el principal objetivo de la flota cangrejera de Alaska. Cabe decir que dicha pesca está controlada bajo estrictos parámetros. Por este motivo se trata de una pesca sostenible. Tanto las hembras como los cangrejos que no cumplan la talla mínima son devueltos de manera inmediata al mar. Las cuotas de pesca son muy restrictivas. El caparazón del cangrejo rojo gigante llega a medir hasta 28 cm, y sus largas patas pueden estar separadas 1,80 cm de una punta a la otra. Esta especie de cangrejo es la más valorada de todas. Su color natural es de un tono borgoña.
- *Cangrejo azul real (Paralithodes platypus):* es otra especie que se pesca en las Islas San Mateo y las Islas Pribilof. Su color es marrón con reflejos azules (característica que otorga su nombre). Sus pinzas son de mayor tamaño que las otras especies y se han llegado a pescar ejemplares de hasta 8 Kg de peso. El cangrejo azul es más delicado que su hermano rojo, se desconoce por completo el motivo, pero se cree que puede ser debido al hábitat más frío de las aguas que puebla.

Estos tipos de cangrejos se pescan a una profundidad de entre 20 y 50 m, ya que en invierno emigra de las altas profundidades (120 – 180 m) hasta superficies menos profundas para desovar. Por este motivo, la pesca suele comenzar en otoño (generalmente sobre el 15 de octubre) para acabar en primavera. A pesar de la autorización legal para pescar durante octubre, la mayoría de las tripulaciones esperan hasta enero para llenar sus cupos por cuestiones biológicas y requerimientos del mercado. Debido a que los cangrejos no pueden ser detectados por los radares y no emigran siguiendo un mismo patrón todos los años, los capitanes deben guiarse por su experiencia y por su intuición a la hora de buscar los mejores lugares para pescar.

Las artes de pesca utilizadas para la pesca de los cangrejos son las nasas. Las nasas son una especie de grandes jaulas metálicas dentro de las cuales se coloca el cebo (bacalao y otras variedades), para posteriormente lanzarlas al agua y recogerlas 12 o 24 horas después.

El Mar de Bering es uno de los mares más peligrosos del mundo, no tan solo por las temperaturas extremas (rozando los 30 grados bajo cero), sino también por los elevados vientos de hasta 80 Km/h. Trabajar en esas condiciones es muy peligroso, no solo por el frío, sino porque el viento hace desplazar las nasas de 300 Kg por cubierta a gran velocidad, suponiendo un riesgo para la tripulación.

Figura 2.- Tripulación de un barco pesquero lanzando una nasa al mar [3]

Generalmente, los pescadores ganan entre 45 y 77 dólares por cada cangrejo real que capturan. En 2018 se registraron unos 7 millones de Kg de cangrejo rojo real pescado en las aguas del Mar de Bering, lo que corresponde a unos beneficios de aproximadamente entre 40 y 65 millones de dólares por temporada. Por lo tanto, la

pesca del cangrejo real en el Mar de Bering es una profesión que recompensa su peligrosidad con altos beneficios económicos para el gremio [4].

Por otro lado, cabe mencionar que el proyecto realizado es de gran versatilidad respecto al tipo de cangrejo de mar sobre el cual realizará el apoyo. Algunos de los ejemplos sobre los cuales se podría realizar apoyo serian el cangrejo Moruno o Velludo, el Centollo, el cangrejo Bay e incluso la gran mayoría de langostas[5].

Con tal de realizar el prototipo para un tipo de cangrejo cuya pesca sea de profundidad media, hemos escogido centrar el proyecto en el cangrejo Real. El único inconveniente resulta ser la situación geográfica y climática necesaria para realizar la pesca del mismo, puesto que lugar donde comúnmente se realiza esta actividad es el Mar de Bering (Alaska) durante el invierno.

[5] Todos estos crustáceos son invertebrados de fondo marino. Entre los enumerados, las profundidades medias donde se suelen encontrar oscilan entre los 15 m (Cangrejo Moruno o Velludo) hasta los 100 m (Centollo).

2. VEHÍCULO DE APOYO

Vista la situación actual de la pesca del cangrejo real en el Mar de Bering, la propuesta de un vehículo de apoyo cobra más sentido aún del que podría haber tenido hasta ahora.

2.1. Diseño del vehículo

A nivel mecánico, la embarcación que se pretende realizar es del tipo catamarán. La elección de este modelo de vehículo marítimo ha sido debida a su característica relación entre la altura y la superficie además de las zonas que se encuentran en contacto con el agua. Al no requerir vela puesto que la embarcación será impulsada por motores, la altura total no deberá ser superior a la de una caja de zapatos, es decir, con una caja estanca donde situar el diseño electrónico es más que suficiente. Por otro lado, la superficie sobre la cual colocar la caja estanca puede tener unas dimensiones mayores, por lo tanto, estaríamos hablando de un vehículo cuya relación superficie-altura es bastante grande.

Esta característica es fundamental para asegurar la estabilidad del catamarán, puesto que el centro de gravedad de un cuerpo con poca altura suele ser bajo y, cuanto más bajo es el centro de gravedad de un cuerpo, mayor es su estabilidad. De igual manera, al tener una gran superficie, el riesgo de volcar se reduce notoriamente.

También hay que tener en cuenta la superficie que se encuentra en contacto con el agua. Por lo general un catamarán es una embarcación bi-casco, es decir, cuenta con dos cascos (a babor y a estribor) encargados de ser la estructura de apoyo sobre la cual recae la totalidad del peso del vehículo. Al tener una superficie de contacto con el agua mínima, las pérdidas por el rozamiento con la misma también disminuyen notablemente. Por lo tanto, y a modo de resumen, un catamarán es perfecto para la aplicación que se pretende desarrollar, puesto que consta de una gran estabilidad por su relación superficie-altura además de tener unas pérdidas por rozamiento considerablemente reducidas.

Dejando de lado la estructura física del proyecto, la aplicación que se lleva a cabo consta también de características propias. Mediante el uso de sensores se registra a tiempo real una serie de datos de interés para el usuario, por ejemplo:

- Situación geográfica del vehículo: el uso de actual de boyas limita en gran medida la localización de las mismas para el marinero y su tripulación, es decir, actualmente el usuario encargado de pilotar la nave es el encargado de saber dónde se han situado las boyas y como llegar hasta ellas. Para facilitarle trabajo al operario de esta labor, el vehículo de apoyo consta de un GPS que muestra su situación a tiempo real, además de poder seguir una ruta a través de los puntos marcados por el usuario mientras la embarcación tripulada se

encuentra estacionada. Además, este elemento permite la fácil localización del vehículo en caso de fallo o avería.

A modo de indicador visual, también se puede añadir una brújula que indique si el vehículo se está desplazando en las direcciones las cuales han sido asignadas.

- Sensado de temperatura y humedad: las condiciones climáticas son adversas en el Mar de Bering, pero son bastante regulares, estas suelen situarse entre -30ºC y -40ºC, por lo tanto, llevar un registro de la temperatura resulta considerablemente fácil. El registro de temperatura es un método útil para saber si la situación del vehículo es la correcta. Un fallo en la electrónica del vehículo puede resultar la pérdida total del mismo, y una de las maneras en las que presentan este tipo de incidencias es el sobrecalentamiento de los elementos dañados. Es por este motivo que llevar un registro a tiempo real puede informar rápidamente de un fallo electrónico y, con suerte, prevenir daños mayores.

- Estabilidad de la embarcación: si bien es cierto que las temperaturas son regulares, los vientos también lo son, y estos pueden alcanzar velocidades de hasta 80 Km/h, los cuales pueden suponer un riesgo en la estabilidad del vehículo en sí mismos tanto como en las dimensiones de las olas levantadas. Si por este o cualquier otro motivo se da el caso de vuelque, esta situación puede ser rápidamente informada al usuario el cual deberá tomar las medidas pertinentes.

- Receptores de información de las nasas: pese a no desarrollar en este proyecto la comunicación como tal entre el vehículo de apoyo y las nasas para la pesca de los cangrejos, de algún modo se deberá realizar un simulacro de recepción de datos a tiempo real con tal de comprobar un correcto funcionamiento en el caso de una aplicación comercial final.

  El método de detección de los cangrejos tampoco será un punto el cual afecte en el desarrollo del proyecto, pero el hecho de no poder ser detectados por radares convencionales complica en gran medida el recuento de la cantidad de cangrejos en las nasas. Un posible método para llevar a cabo esta acción puede ser la de situar un sensor láser en las puertas de entrada de las nasas[6].

Al margen del sensado de información también se existen actuadores programables en función de los datos obtenidos. Este es el caso de los motores, la baliza o la bocina. Los motores tienes dos modos de actuación los cuales hemos denominado manuales y autónomos. Cuando los motores se controlan manualmente, el usuario de la aplicación es el encargado de controlar la dirección, sentido y velocidad de los mismos, mientras

[6] En primer lugar, se planteó la posibilidad de proponer un sistema basado en un final de carrera en vez de uno basado en sensores láser. Finalmente fue declinado puesto que es necesario tener en cuenta la flora y fauna del fondo marino las cuales podrían llegar a dificultar el correcto funcionamiento del mismo.

que cuando se encuentran en modo autónomo, el usuario solo deberá establecer un conjunto de coordenadas, las cuales serán seguidas por el vehículo.

Por otro lado, como se ha comentado anteriormente, la ley vigente advierte del uso obligatorio de una baliza de señalización blanca en proa y bocina para situaciones de poca visibilidad (2 bocinazos cada 2 minutos), estos actuadores nunca son controlados por el usuario. La baliza se encuentra en funcionamiento en todo momento, mientras que, en función de las condiciones climáticas (y la elección del usuario) la bocina se encontrará en funcionamiento o no.

Por lo tanto, los motores son los únicos elementos del vehículo sobre los cuales podrá actuar el usuario de la aplicación.

El control de la aplicación se podrá llevar a cabo desde un PC mediante una aplicación en LabVIEW™[7], el propósito de la aplicación es tener una interfaz accesible para cualquier usuario la cual sea entendible y no requiera de conocimientos avanzados para su correcto funcionamiento. La estructura básica que deberá seguir la misma será; en primer lugar, la representación en tiempo real de los datos obtenidos mediante el sensado de las magnitudes físicas anteriormente comentadas y, por otro lado, el control de los actuadores que se requieran (motores, baliza y bocina).

Por otro lado, el registro de datos se llevará a cabo desde una segunda aplicación desarrollada también en LabVIEW™, desde esta segunda aplicación se podrá guardar la información recogida de la sensorica del vehículo y recurrir a ella cuando sea necesario a partir de un documento de cálculo Microsoft Excel™ que se guardará por defecto en el escritorio del PC utilizado[8], pero no se podrá llevar el control sobre el vehículo de apoyo desde la misma.

Ambas aplicaciones pueden utilizarse simultáneamente y no interfieren entre ellas de ningún modo, así pues, el usuario puede llevar el control del vehículo y ver los datos inmediatos del mismo mientras guarda el histórico de datos o comprueba el correcto funcionamiento durante el periodo de tiempo deseado.

Para facilitar la interpretación del sistema, el diagrama de bloques de la aplicación corresponde al siguiente:

[7] Se ha establecido LabVIEW™ como marco de la aplicación ya que es un software que proporciona un potente entorno de desarrollo para el diseño de aplicaciones de Ingeniería de adquisición de datos, análisis de medidas y presentación de datos gracias a un lenguaje de programación carente de la complejidad de otras herramientas de desarrollo.

[8] Para más información sobre el funcionamiento de ambas aplicaciones el Manual de Instrucciones se encuentra situado en el apartado “13. Anexos”.

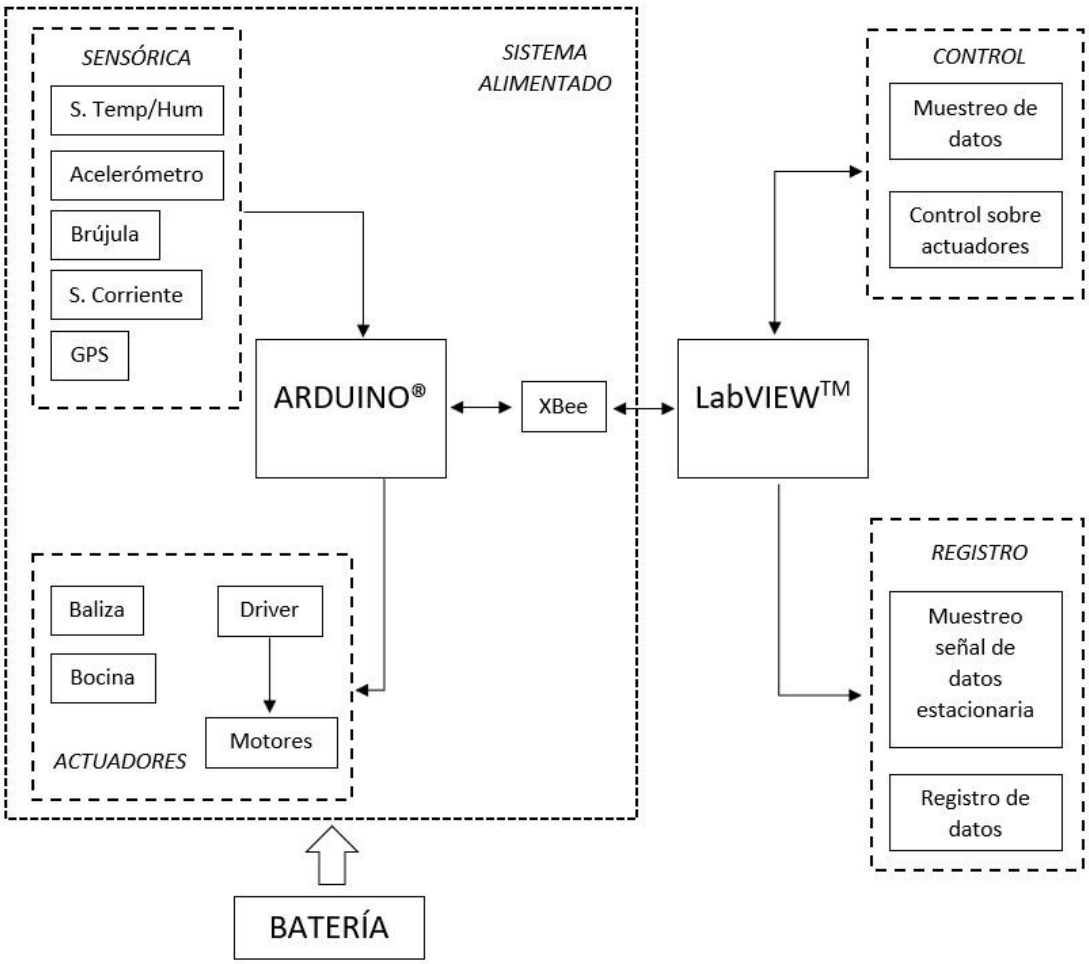


Figura 3.- Diagrama de bloques de la aplicación

ARDUINO® y LabVIEW™ mantienen una comunicación ininterrumpida, ejerciendo de receptores y emisores de datos continuamente.

En primer lugar, ARDUINO® es se encarga de obtener los datos sensados de los elementos de sensórica. Estos datos son interpretados y enviados a LabVIEW™ simultáneamente a través del Xbee™.

LabVIEW™, por su parte, recibe los datos y los muestra en la aplicación o aplicaciones que se encuentren abiertas y en funcionamiento en ese instante. Si se quiere realizar un control sobre el vehículo, entre las opciones a las cuales puede recurrir el usuario (control manual, control automático, encendido de baliza…) escoge la que desee. LabVIEW™ le comunica la información a ARDUINO® y este actúa sobre los actuadores en consecuencia mediante el software IDE de la aplicación.

El complejo del sistema se puede dividir en cinco bloques principales:

- ARDUINO® y LabVIEW™: Componen el cerebro del sistema. ARDUINO® obtiene los datos de los sensores y los envía a LabVIEW™ a través del Xbee™, desde LabVIEW™ el usuario controla la placa de ARDUINO® y decide lo que quiere que suceda en cada momento, teniendo siempre como referencia los datos que ARDUINO® ha registrado previamente, siendo esta información enviada de vuelta la placa utilizando también el Xbee™.
- Alimentación: El principal método de alimentación del conjunto es una batería de Plomo-Ácido de 12V y 8Ah, esta es la encargada de alimentar el driver para el control de motores y, mediante reguladores, alimentar el resto de componentes del circuito (ARDUINO®, sensórica, Xbee™ y el resto de actuadores).
- Sensórica: Los elementos que pertenecen a este apartado son los encargados de transformar las magnitudes físicas a las cuales se exponga el vehículo a señales que pueda interpretar ARDUINO®.
- Actuadores: Los actuadores presentes en el conjunto son tres: la baliza, la bocina y los motores. En función de lo establecido por el usuario de la aplicación, pueden encenderse la baliza o la bocina o, gracias al driver implementado, regular la velocidad de giro de los motores, lo cual proporciona mayor o menor propulsión del vehículo en uno u otro sentido.
- Aplicación: La aplicación desarrollada es la que gobierna el sistema completo. Se trata de una aplicación en LabVIEW™ mediante la cual el usuario puede conocer, registrar o modificar los valores que desee.

2.2. Legislación Vigente

Como vehículo no tripulado, el prototipo diseñado debe ser regulado por el conjunto de normativas que recoge la ley respecto a los drones marítimos, también conocidos como USV.

Por sus siglas en inglés, un USV (*Unmanned Surface Vehicle*) es un vehículo autónomo de superficie. Con superficie se refiere a la lámina de agua (no submarino) y no hay que confundirlo con la superficie terrestre. Los drones terrestres se llaman UGV (*Unmanned Ground Vehicle*). Por tanto, un USV no es cualquier dron marino, es un barco autónomo o barco no tripulado, ya que va por la superficie del agua y no inmerso en la misma.

Un USV es un barco no tripulado y como indica su nombre, consiste en una embarcación que se ha configurado o adaptado para poder ser utilizada en remoto o que funcione de forma autónoma. El concepto es sencillo, al final como un dron aéreo,

podemos controlar esta embarcación a distancia o preprogramar las rutas que debe hacer.

Algunas de las características más importantes de un USV es la cantidad de nuevas posibilidades y ventajas que ofrece respecto a un dron aéreo. Las más relevantes serían:

- Una autonomía muy superior a cualquier dron aéreo.
- Una tecnología más fiable, con componentes más testados, aunque de mayor tamaño.
- Una resistencia de componentes muy superior.

A diferencia de los drones aéreos y su ley específica, los drones marítimos en España no están regulados por una ley concreta. De hecho, el término USV o embarcación no tripulada no entra dentro de las leyes marítimas.

Lo primero que debemos hacer es entender que, a diferencia del aire, donde el espacio aéreo tiene un órgano regulador independientemente ya sea el cielo sobre la propia vivienda o sobre la ciudad de Madrid, en el agua no es lo mismo hablar de una terminal portuaria, de una playa, del medio del océano, de un embalse o de la piscina particular.

Así pues, en lugares no regulados como pueda ser una piscina para acuicultura, un embalse privado o lugares donde no va a haber tráfico marítimo, nuestro USV no debe estar matriculado, ni llevar las medidas de seguridad obligatorias, ni regirse por ningunos patrones preestablecidos y homologados. En cambio, cuando hablamos de áreas reguladas a nivel marítimo, como puede ser un puerto, una playa o el océano, un barco no tripulado está bajo el paraguas legal de cualquier embarcación.

Los dos elementos principales obligatorios a tener en cuenta para la navegación de nuestra embarcación, dado que no tiene tripulación, son los siguientes: Baliza de señalización blanca en proa y bocina para situaciones de poca visibilidad [5] (2 bocinazos cada 2 minutos).[9]

[9] Se adjunta en la bibliografía el enlace a la totalidad de la ley en cuanto a la navegación de embarcaciones. Dicha legislación del ministerio de fomento se extrapola a todas las aguas internacionales ya que se comparte a nivel mundial.

3. DISEÑO MECÁNICO

El diseño llevado a cabo para desarrollar el proyecto de vehículo de apoyo es el de un catamarán. Un catamarán es una embarcación o buque multicasco con dos cascos paralelos de igual tamaño. Es una embarcación de geometría estabilizada, derivando su estabilidad de sus dos quillas, en lugar de una quilla con bulbo de los veleros monocasco.

Los principales aspectos por los cuales se ha escogido este diseño han sido:

- Velocidad: Se requiere de un vehículo que navega por la superficie marina en alta mar, es decir, cuanta mayor velocidad obtenga el vehículo, mayor alcance en distancia obtendremos lo cual equivale en un rango de operación mucho más amplio.

  Los dos cascos del catamarán suelen tener una resistencia hidrodinámica más pequeña que las embarcaciones monocasco, hecho que deriva en una velocidad más elevada con un consumo similar.

- Estabilidad: La estructura principal de un catamarán es la de una gran superficie de poca altura apoyada sobre dos puntos; los cascos. Estos elementos favorecen la estabilidad por dos aspectos: la baja altura de la embarcación hace bajar el centro de gravedad del conjunto y cuanto más bajo se encuentra el centro de gravedad de un cuerpo más estable es, por otro lado, la estabilidad aumenta considerablemente al contar con dos puntos de apoyo situados en los extremos de la embarcación.

  La estabilidad del vehículo es la que marca el riesgo real de volcado, por lo tanto, hacer aumentar este elemento es crucial para evitar el riesgo de fallo general.

- Habilidad: Se entiende por habilidad la relación entre la superficie en centímetros cuadrados (cm^2) aprovechables respecto a la superficie en contacto con el agua.

  El montaje de una estructura electrónica accesible en el trampolín[10] del vehículo requiere de una superficie lo suficientemente grande como para que no resulte un inconveniente el recambio de ningún elemento o la carga de la batería que proporciona potencia al conjunto. Por lo tanto, la habilidad del vehículo es la que marca el tamaño del mismo, y este viene condicionado por la circuitería necesaria para el correcto funcionamiento de la aplicación desarrollada.

[10] Se denomina comúnmente como trampolín la superficie aprovechable entre los dos buques del catamarán. Recibe este nombre por el uso generalizado de telas para su elaboración. En nuestro caso se trata de una plancha de plástico.

Con tal de asegurar la estanqueidad del montaje eléctrico y electrónico se ha optado por el uso de cajas herméticas interconectadas encargadas de mantener resguardados los elementos susceptibles a fallar por el contacto con alta humedad.

El prototipo mecánico se basa en un trampolín de metacrilato y una caja estanca situada sobre éste en la cual se encontrarán los componentes eléctricos y electrónicos. A babor y estribor de este trampolín se encuentran los cascos o flotadores de fibra de vidrio, y bajo todo el sistema, los motores encargados de impulsar el vehículo en la dirección deseada.

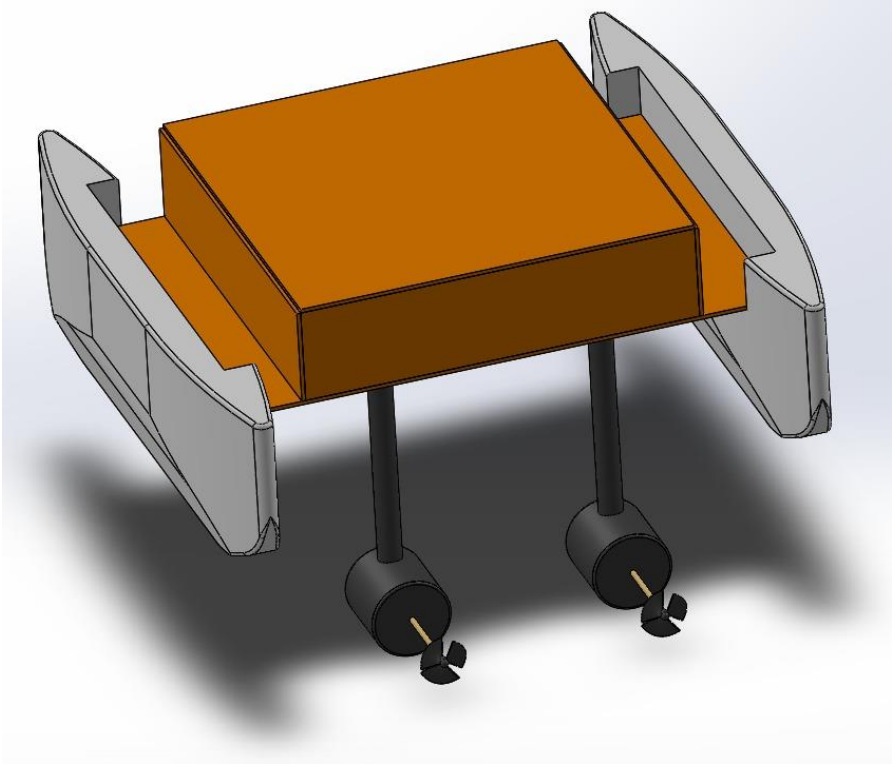


Figura 4.- Diseño en SolidWorks® de la embarcación

3.1. Cascos

El diseño de los cascos del vehículo se llevó a cabo tomando como referencia vehículos ASV[11] (por sus siglas en ingles *Autonomous Sailing Vessel*), vehículos capaces de alcanzar velocidades de hasta 50 nudos[12] (92,60 km/h).

[11] Desarrollados por *Artemis Techologies*, se trata de un tipo de velero no tripulado capaz de navegar a través del agua utilizando sistemas avanzados de control computarizado y fuentes de energía sostenibles.

[12] El nudo es una unidad métrica de velocidad, utilizada tanto para navegación marítima como aérea, equivalente a una milla náutica por hora (1 nudo = 1NM/h = 1,852 km/h = 0,514m/s).

Teniendo en cuenta las especificaciones definidas para las medidas finales del prototipo, se realizó un diseño en SolidWorks® correspondiente a la siguiente figura (los planos acotados de cada elemento se encuentran en el apartado "13. Anexos"):

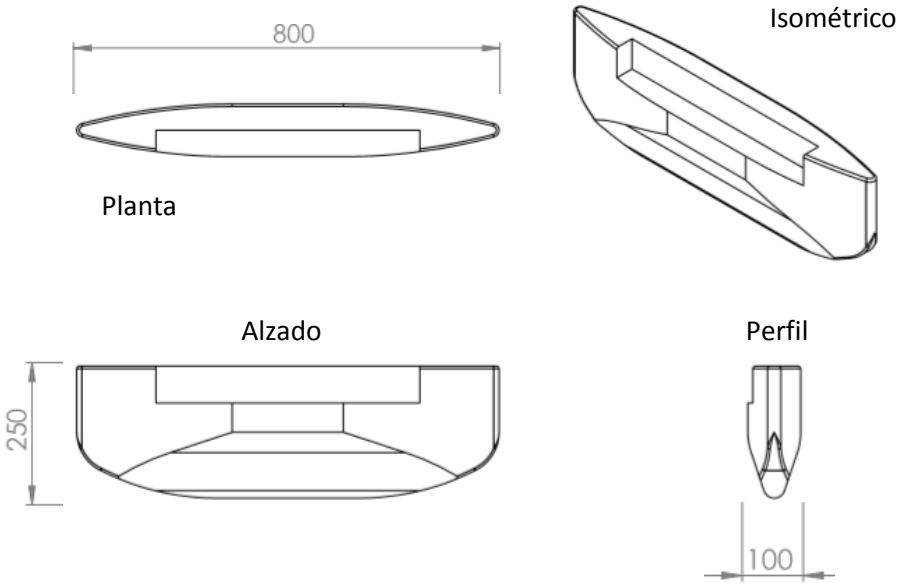


Figura 5.- Vista de planta, alzado, perfil e isométrica del casco diseñado acotadas en milímetros

Una vez realizado el diseño de los cascos, mediante la herramienta FloXpress™ de SolidWorks® se realizó el estudio de rozamiento con valores de hasta 50 km/h y temperatura del agua de 0 ºC (condiciones estándar similares a las previstas en el Mar de Bering). En función de estos elementos, el estudio resultó en las siguientes imágenes:

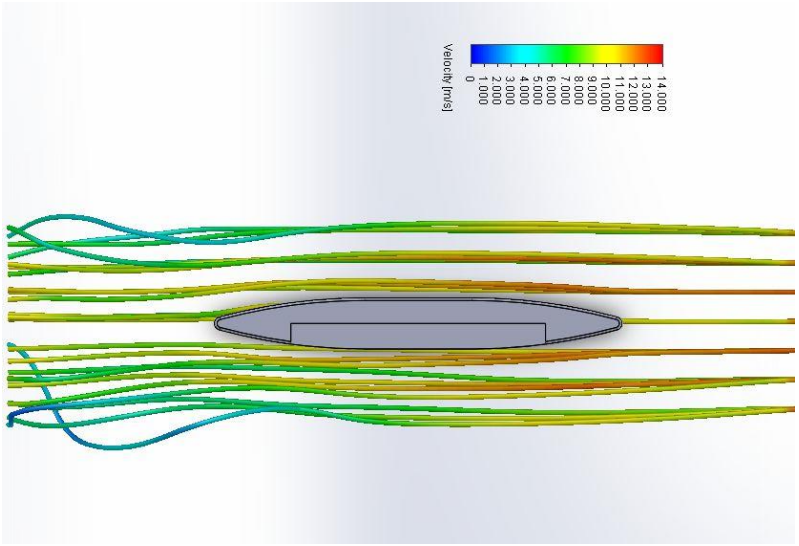


Figura 6.- Vista de planta del ensayo realizado en FloXpress™

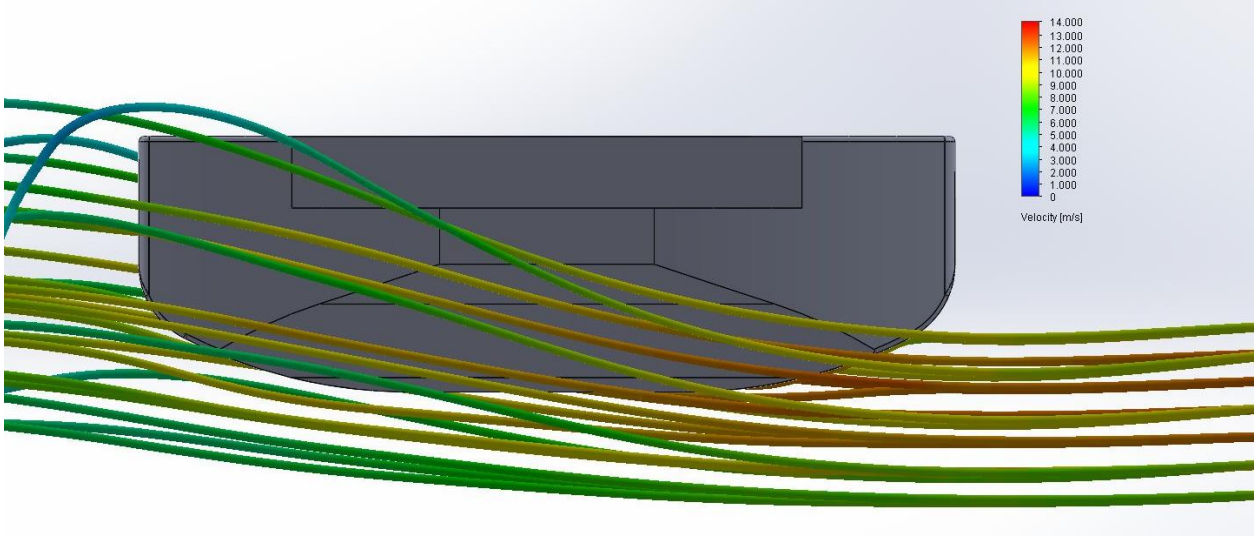


Figura 7.- Vista de alzado del ensayo realizado en FloXpress™

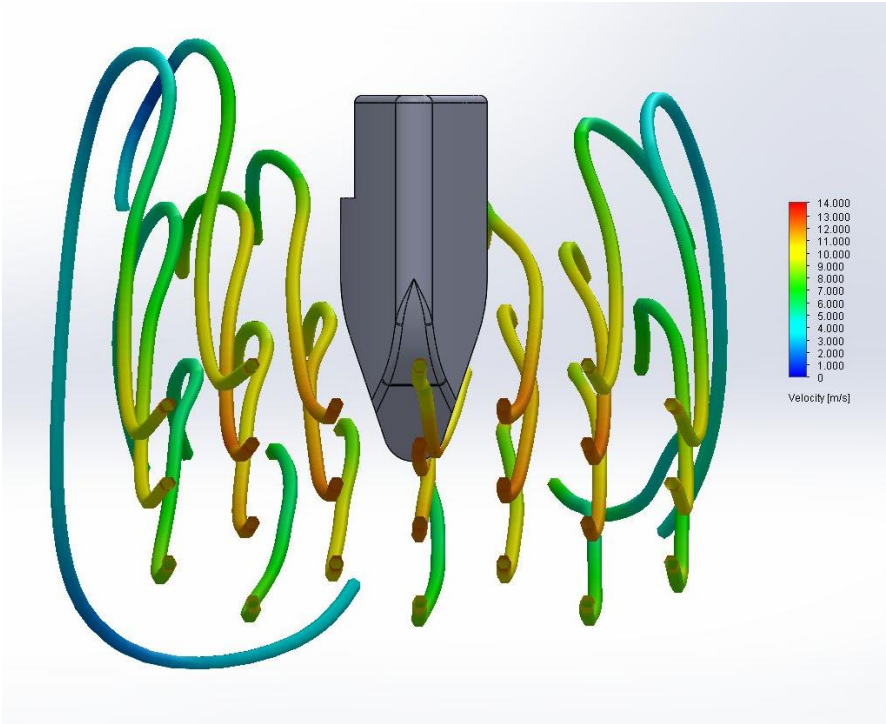


Figura 8.- Vista de perfil del ensayo realizado en FloXpress™

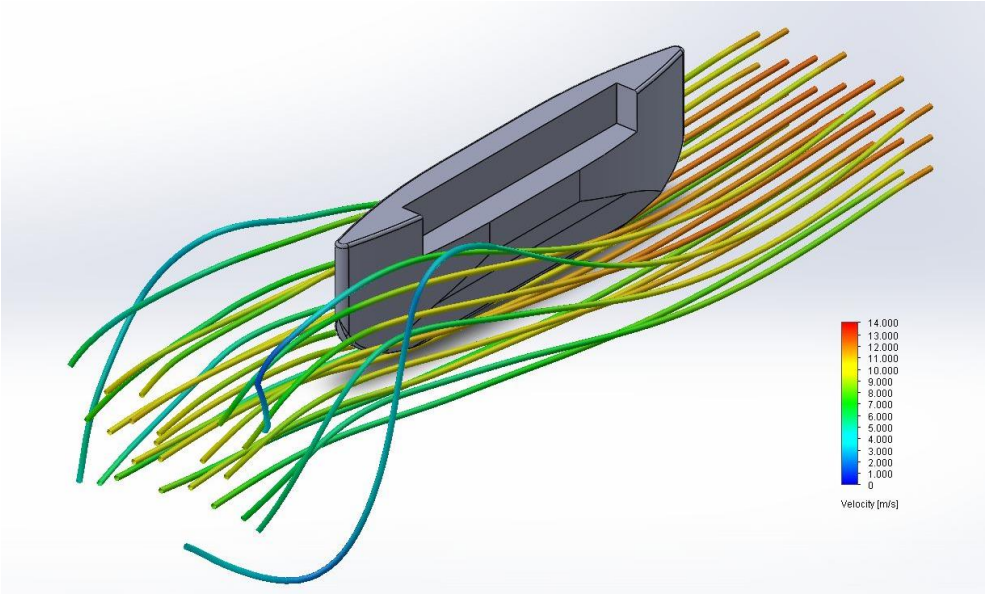


Figura 9.- Vista isométrica del ensayo realizado en FloXpress™

En estas figuras se puede observar la poca caída de velocidad del agua que se encuentra en contacto con la superficie del casco, por lo tanto, tras haber aprobado el diseño teórico, se dio inició al modelado de los mismos.

3.1.1. Modelaje

Una vez finalizado el proceso de diseño y simulación de los flotadores del vehículo, se ha procedido al modelaje de los mismos.

Tras barajar diversas opciones junto a nuestro tutor y después de una búsqueda de empresas que se dedican al modelaje de cascos a pequeña escala, se descartó el encargo de los mismos a casusa de el alto coste económico.

La opción escogida para nuestro prototipo fue la del modelaje de los cascos a partir de poliestireno expandido, con un revestimiento de fibra de vidrio y resina.

El proceso se puede dividir en 5 etapas:

I) MODELAJE: El modelaje de ambos cascos se ha realizado a partir del diseño expuesto en el apartado “3.1. Cascos”. Partiendo de un bloque de poliestireno expandido, se ha ido realizando el corte con una cuchilla caliente y retirando pequeñas capas hasta llegar a la forma deseada.

Figura 10.- Modelaje de casco en poliestireno expandido

II) CAPA PROTECTORA: Tras algunas pruebas con el compuesto químico (resina) y el poliestireno expandido, éste se resentía al químico, por lo que ha sido necesario implantar una capa protectora de un material plástico (cinta adhesiva) para amortiguar la corrosión que provoca la resina.

Figura 11.- Capa protectora de cinta adhesiva

III) CAPA DE FIBRA DE VIDRIO Y RESINA: Una vez puesta la capa protectora, se ha procedido al revestimiento con la fibra de vidrio y la resina.

La fibra de vidrio que se ha utilizado tiene una densidad de 300g/m2 y es utilizada para reparaciones de embarcaciones, pero dadas las pequeñas dimensiones de los flotadores diseñados, esta fibra se adapta totalmente a nuestras necesidades.

La resina utilizada es resina de poliéster tixotropada y preacelerada. Esta resina se conserva en estado líquido, pero al ser mezclada con un catalizador (2% para 500gr de resina), la resina se endurece en unos minutos y se seca. La temperatura de secado ideal es entre 16 ºC y 25 ºC. A menos temperatura el secado se retarda.

Una vez realizada la mezcla de resina y catalizador, se ha ido aplicando a lo largo del casco utilizando una brocha, posteriormente se ha aplicado una capa de fibra de vidrio y más resina.

Este proceso se ha repetido hasta conseguir un grosor adecuado para el casco.

Figura 12.- Revestimiento de fibra de vidrio y resina

IV) PULIDO DE CASCOS: En este punto, se ha procedido a pulir los cascos mediante una pulidora para regular imperfecciones, así como las burbujas de aire anteriormente comentadas. Éstas se han pulido y rellenado de resina con tal de disminuir las imperfecciones.

Figura 13.- Pulido de los cascos

V) PINTURA: Finalmente se ha procedido a aplicar una pintura especial a modo de spray para cascos de embarcaciones.

Dado el limitado presupuesto e inexperiencia en este ámbito, no se ha llegado a una calidad profesional como la de las embarcaciones convencionales, pero para nuestro prototipo, cumple totalmente las funciones esperadas.

3.2. Caja Estanca

Para poder mantener estáticos y seguros todos los componentes eléctricos y electrónicos del prototipo, ha sido necesario, a parte de las cajas herméticas individuales donde se encontrarán dispuestos los componentes de forma individual o en pequeños grupos, una caja estanca donde recoger los el conjunto de estas cajas.

Tras de valorar diversas opciones, nos fue imposible una caja estanca que se adecuara a las necesidades del proyecto en cuanto a dimensiones, al menos no a un precio asequible, así pues, se tomó la decisión de elaborar manualmente la caja hermética acorde a nuestras necesidades [6].

La caja estanca de nuestro prototipo se ha elaborado a partir de planchas de metacrilato de 5mm de grosor. Las medidas de esta caja son 50 cm de ancho y largo y 12 cm de alto, unida a la base del catamarán de 50 cm de ancho y 75cm de largo.

Los 4 laterales de la caja han sido unidos mediante escuadras. Los agujeros de los tornillos se encuentran sellados con goma a presión entre metacrilato y tornillo y finalmente reforzado con silicona. Todas las juntas entre las paredes y la base también han sido selladas con silicona (especial para elementos estancos e impermeables).

Por último, la tapa debía ser retirable para poder acceder a los componentes en caso de avería. Por eso, se ha colocado una junta de goma entre la tapa y los laterales, y se han utilizado tornillos expandibles para las diferentes uniones. Estos tornillos expandibles son especiales para cavidades huecas, como es nuestro caso, y permiten su retirada para poder acceder al interior de la caja.

Una vez finalizada, se han realizado pruebas de estanqueidad llenando el interior de la caja de agua y comprobando posibles fugas, obteniendo un resultado satisfactorio que suma un plus de seguridad respecto al hermetismo de los componentes.

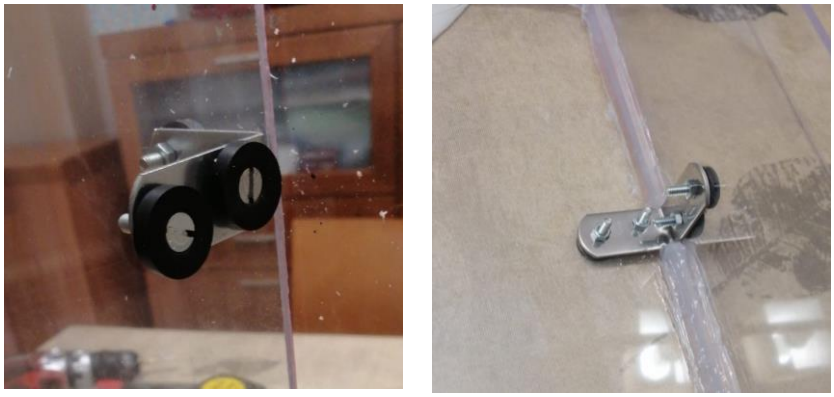


Figura 14.- Métodos de sellado

Con el fin de asegurar la impermeabilidad de los componentes situados en el interior de la caja estanca, se añadió una capa de material absorbente en el fondo de la misma previniendo así la posible situación de fuga por pequeña que sea. Sobre este material absorbente, una plancha de plástico sirve una vez más de protección contra el contacto directo del material absorbente (que podría mojarse) y las cajas estancas donde se sitúan los componentes.

El desarrollo y confección de esta caja estanca, como hemos comentado anteriormente, cumple con nuestras necesidades en cuanto a prototipo. Sabemos que los materiales y uniones utilizados, no garantizan una estanqueidad a largo plazo, pero sí solventan nuestra problemática, ya que se trata de un prototipo. Encontramos cajas

estancas en el mercado, de dimensiones mucho más pequeñas, alrededor de los 50€[13]. No hemos encontrado ninguna de las medidas adecuadas para nuestro proyecto, así que supondría realizarla por encargo, con un elevado coste.

Figura 15.- A la izquierda, la base de material absorbente, a la derecha, el plástico de protección encima de ésta

Figura 16.- Montaje final de la caja estanca con la baliza y la bocina implementadas

[13] En la bibliografía se encuentran los enlaces con varias cajas estancas del mercado.

3.3. Montaje y Encapsulado de Motores y Eje motor

El único punto del prototipo en el cual el agua se encuentra cerca de un componente eléctrico sucede en la el final de las piezas de transmisión, es decir, en el encapsulado de motores. Una fuga de agua en este punto podría llegar a suponer un cortocircuito general y, por lo tanto, la avería del vehículo en sí. Por lo tanto, la necesidad de aislar totalmente los motores del agua y evitar a toda costa la entrada de agua en el encapsulado de los mismos es el primer punto de peligro a tener bajo control.

Para realizar la transmisión del giro del motor a las hélices, han sido barajadas diferentes opciones en cuanto a donde emplazar los componentes.

En primer lugar, se valoró la idea de incluirlos en la caja estanca, lejos del peligro del agua, y mediante una relación de transmisión 1:1 formada por dos engranajes cónicos, trasladar el giro hasta las hélices, estas sí sumergidas en el agua.

Debido a la gran pérdida de rendimiento por fricciones y la dificultad de montaje en cuanto a transmisión, se descartó esta primera opción. El sistema de engranajes podría haberse impreso con una impresora 3D, pero la imposibilidad de realizar el tipo de impresión con los tiempos y el material adecuado tan solo nos hubiese llevado a resultados desfavorables. Por otro lado, la adquisición de un sistema de más calidad, hubiese supuesto un precio excesivamente elevado.

Barajando la necesidad desde otro ángulo, se valoró la idea de situar los motores en el interior de la caja estanca, pero con una orientación ligeramente diagonal, de este modo el eje del motor y la hélice llegarían a entrar en contacto con el agua e impulsarían al vehículo con la seguridad de mantener el motor seco y un rendimiento aceptable. Acabamos considerándola como la peor opción, tanto por la distancia hasta el agua, como por la orientación de las hélices, ya que el vehículo sería impulsado hacia adelante y hacia arriba simultáneamente, transformando el hecho de mantener un buen rendimiento mecánico pero un desplazamiento lento e ineficiente. A pesar de ser sencilla y óptima en cuanto a entradas de agua, hemos considerado que no era oportuna para nuestro proyecto.

Por último, la solución escogida para este problema, ha sido el de ubicar los motores por debajo de la línea de flotación y unir el eje de rotación del motor con la varilla y la hélice directamente, encapsulando los motores y aislándolos para que puedan estar bajo el agua sin que se mojen.

Para el montaje se ha utilizado material de PVC del ámbito de la fontanería, para crear una pequeña capsula hermética. Esta capsula contiene el motor, anclado a una pequeña plataforma de plástico hecha a medida, y se han sellado las dos tapas de la cápsula con un pegamento específico para sellado de tuberías. El motor se mantiene en alto dentro de la capsula, de manera que, si entrase algo de agua a través del eje de

rotación, ésta iría hacia el fondo por gravedad, donde hay colocado un material absorbente.

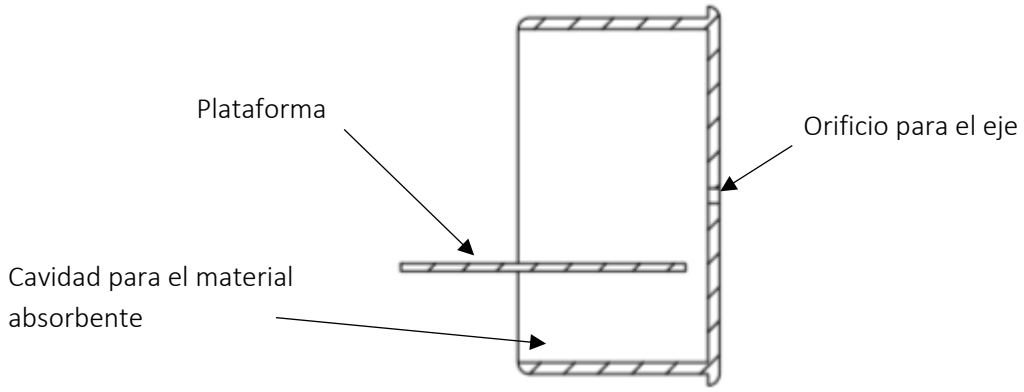


Figura 17.- Vista de perfil seccionada del esquema de la tapa del motor

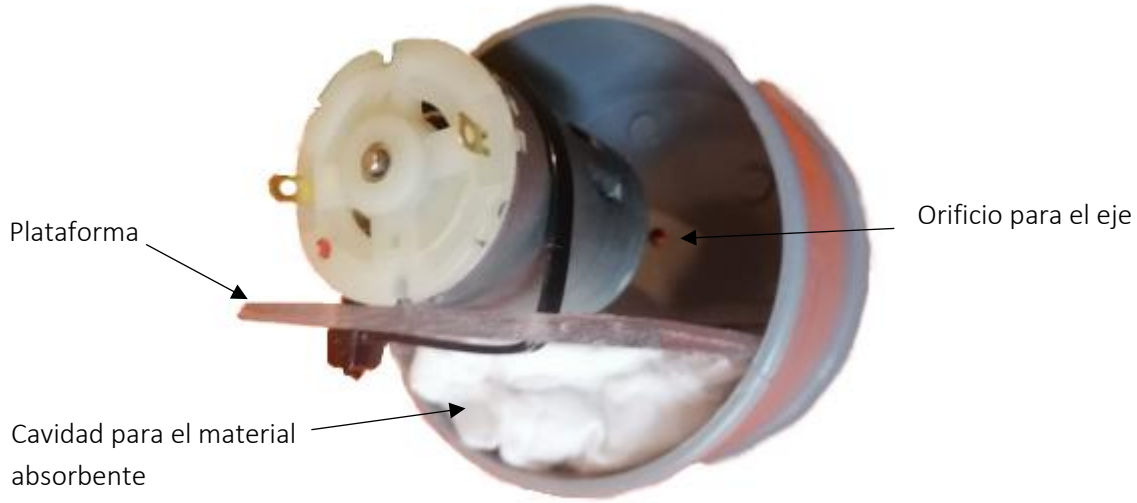


Figura 18.- Vista de tapa del motor desarrollada

Tras consultar varias tiendas de modelaje y radio control de barcos y submarinos, decidimos contar con la colaboración de una de ellas, donde se han comprado los elementos necesarios para crear una transmisión estanca, preparada para submarinos de radio control, mediante la cual resulta muy difícil que circule el agua a través del orificio por donde sale el eje. La varilla consta de un retén que impide la entrada del

agua, así como una capa de una grasa especial que permite el giro del eje e impide la entrada de agua.

Por último, se han acoplado unos tubos de PVC a la capsula por donde bajan los cables de alimentación desde la batería situada en la caja estanca hasta cada motor. Estos tubos se han combinado y sellado tanto a la capsula como a la caja estanca mediante el pegamento anteriormente comentado, especial para sellados herméticos. Como se verá en el apartado “7.2. Test 2”, éste no ha resultado el método definitivo de sellado de juntas.

Figura 19.- Vista del acoplamiento de transmisión junto al cableado de motores

Durante y después del montaje, se ha ido testeando la estanqueidad de las juntas, sumergiendo las mismas durante una hora aproximadamente y verificando que no penetra el agua por ningún punto.

3.4. Ensamblaje definitivo

Como se ha comentado anteriormente, los principales aspectos por los cuales se ha escogido este diseño han sido la velocidad, la estabilidad y la habilidad.

Para el primer punto contamos con unos motores genéricos que no aportarían al conjunto una velocidad notoria, pero el estudio realizado para establecer el diseño de los flotadores reduce las perdidas por fricción al mínimo, lo cual representa una velocidad más elevada respecto al consumo (ver figuras 6-9). De nuevo, mediante la herramienta FloXpress™ de SolidWorks® se realizó el estudio de rozamiento con valores de hasta 50 km/h y temperatura del agua de 0 ºC, valores a los que no esperamos enfrentar el prototipo pero que demuestran su correcto funcionamiento sobre los valores estándar. El estudio realizado se muestra en las siguientes imágenes:

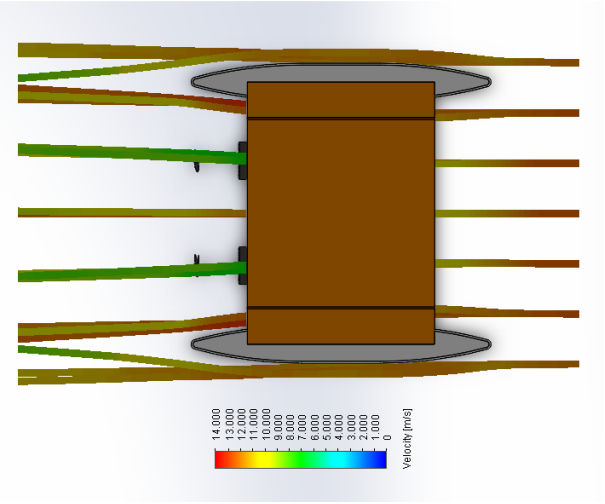


Figura 20.- Vista de planta del ensayo realizado en FloXpress™

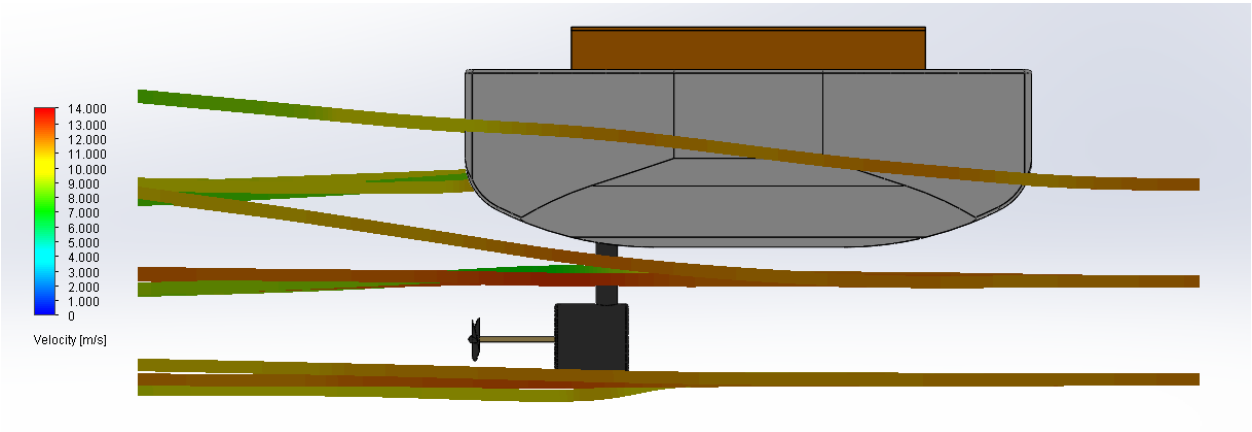


Figura 21.- Vista de perfil del ensayo realizado en FloXpress™

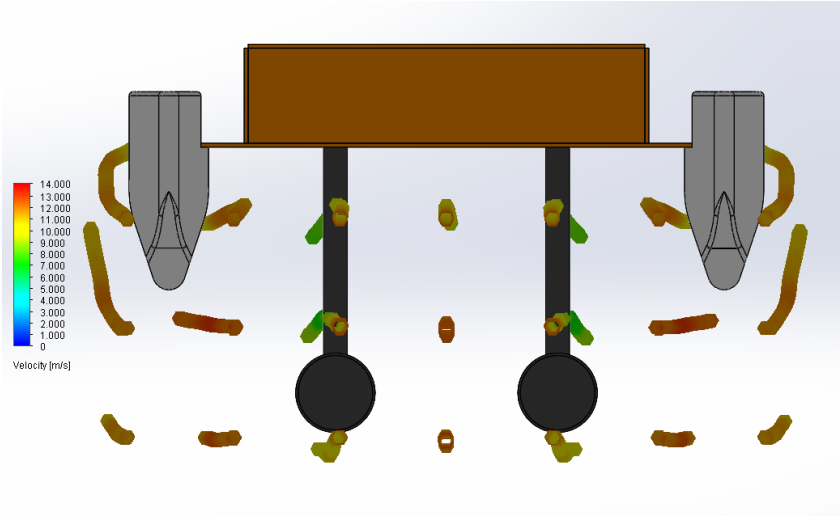


Figura 22.- Vista de alzado del ensayo realizado en FloXpress™

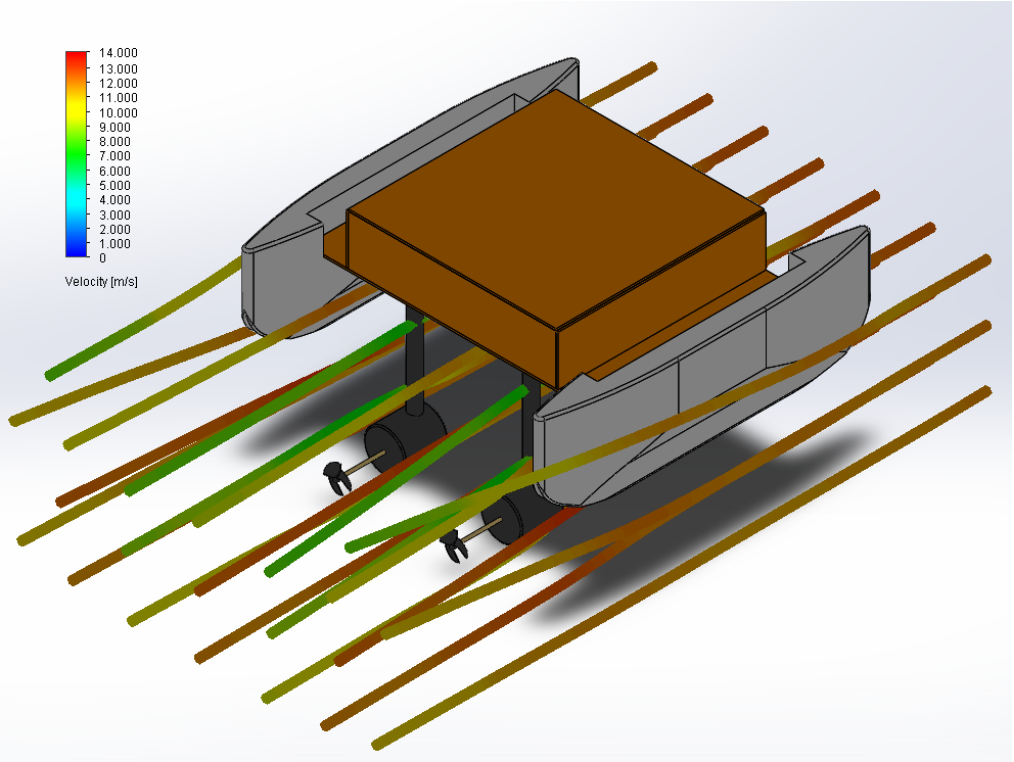


Figura 23.- Vista isométrica del ensayo realizado en FloXpress™

Como se puede apreciar en estas figuras, y de la misma manera que sucedió al realizar el estudio de los flotadores individualmente, el resultado simulado es más que satisfactorio. Se pueden distinguir caídas máximas a 8 o 9 m/s en las zonas de cruce con la pieza de transmisión, pero por lo general muestra un comportamiento de rozamiento mínimo entre el agua y la estructura realizada.

El segundo aspecto a resaltar es la estabilidad del prototipo. Para evaluar esta característica se recurrió a la herramienta del cálculo de centro de masas de SolidWorks®. Como ya se ha comentado, mantener un centro de gravedad bajo es clave para asegurar la estabilidad de cualquier cuerpo, por lo tanto, mantener bajo el centro de gravedad del prototipo ha sido fundamental en el desarrollo del diseño mecánico y se ha realizado de forma satisfactoria como muestran las siguientes imágenes:

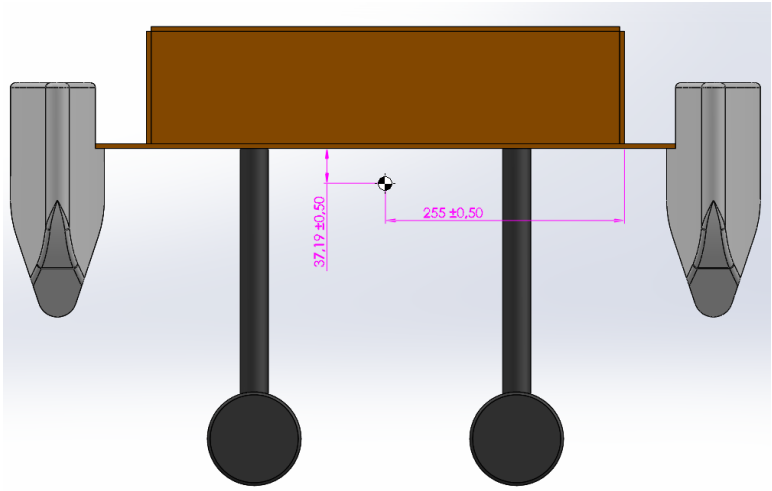


Figura 24.- Vista de alzado del ensayo realizado en sobre el centro de gravedad

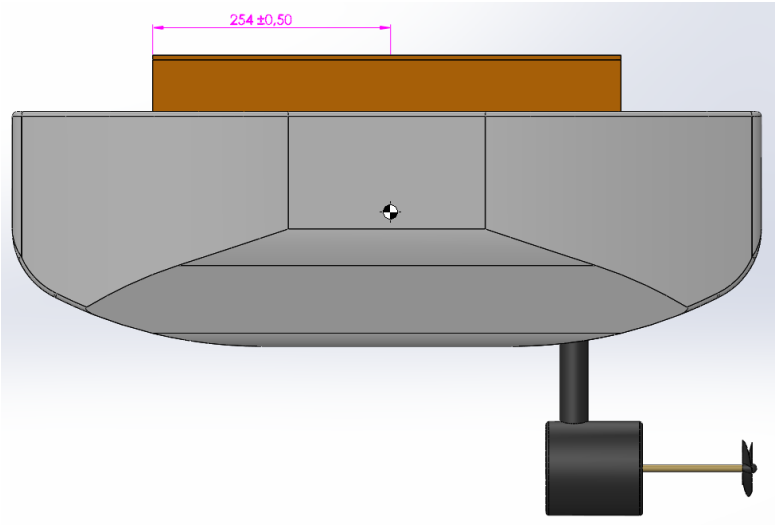


Figura 25.- Vista de perfil del ensayo realizado en sobre el centro de gravedad

Nos encontramos con un centro de masas situado ligeramente por debajo de la estructura principal de la caja hermética diseñada y a media altura de los flotadores. La

situación geométrica de este punto favorece la estabilidad del prototipo y dificulta la situación de vuelque.

Finalmente, la habilidad del prototipo, es decir, la relación entre la superficie aprovechable respecto a la superficie en contacto con el agua es favorable. La superficie de los flotadores en contacto con el agua, al margen de ser superficie no aprovechable, ya que se encuentra alejada del trampolín, es de aproximadamente 2600 cm^2 por flotador (2603,3 cm^2 valor obtenido mediante SolidWorks®) mientras que la superficie aprovechable del trampolín es de 6000 cm^2 (3500 cm^2 de la base y 2500 cm^2 de la tapa). En dicha superficie aprovechable es donde se encuentra situada la circuitería con la cual ha sido distribuida en tres cajas estancas con el suficiente margen como para no interferir entre ellas ni alejarlas en exceso.

Figura 26.- Organización de la caja hermética

Dentro de estas cajas es donde se encuentran situados los módulos electrónicos explicado posteriormente en el apartado “4.12. Módulos Soldados”.

Figura 27.- Organización de los módulos dentro de las cajas interiores

4. HARDWARE

Con tal de facilitar el diseño eléctrico y electrónico del prototipo, y siguiendo el diagrama de bloques expuesto en la figura 3, el hardware de nuestro vehículo consta de varios módulos implementados y algunos circuitos supletorios divididos principalmente en: alimentación, sensórica y actuadores, controlado por una placa de ARDUINO® Mega 2560, a la cual se encuentran conectados todos los periféricos.

En la siguiente figura podemos observar el esquemático general[14] de nuestro hardware:

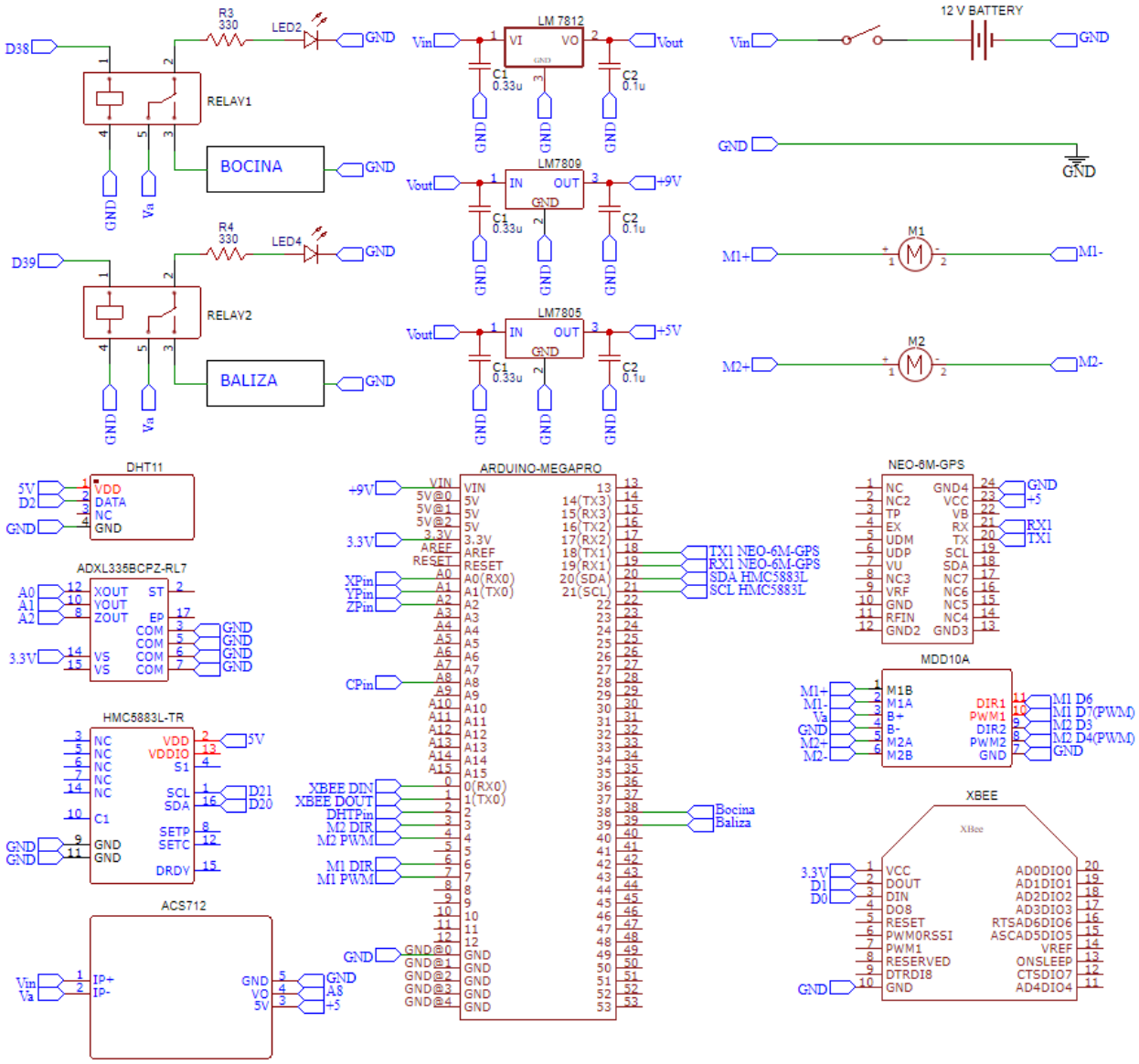


Figura 28.- Esquemático general del sistema

[14] Las hojas de datos de todos los componentes utilizados [7] se encuentran disponibles en el apartado "13. Anexos"

4.1. Placa ARDUINO® MEGA 2560

La elección de la placa ARDUINO® como µcontrolador enfrente a otras propuestas de mercador ha sido principalmente la fiabilidad de la marca y la facilidad de adaptación al lenguaje con este µprocesador, además de que existen infinidad de módulos e información que facilitan la implementación y programación del vehículo. ARDUINO® Mega 2560 es una placa de desarrollo basada en el µcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16Mhz, conexión USB, *jack* para alimentación DC, conector ICSP, y un botón de reseteo[8]. Este µcontrolador va a ser el encargado de gestionar todas las señales tanto de entrada como de salida del resto de módulos y circuitos conectados. También va a ser el encargado de comunicarse con el software basado en LabVIEW™ y procesar la comunicación a distancia.

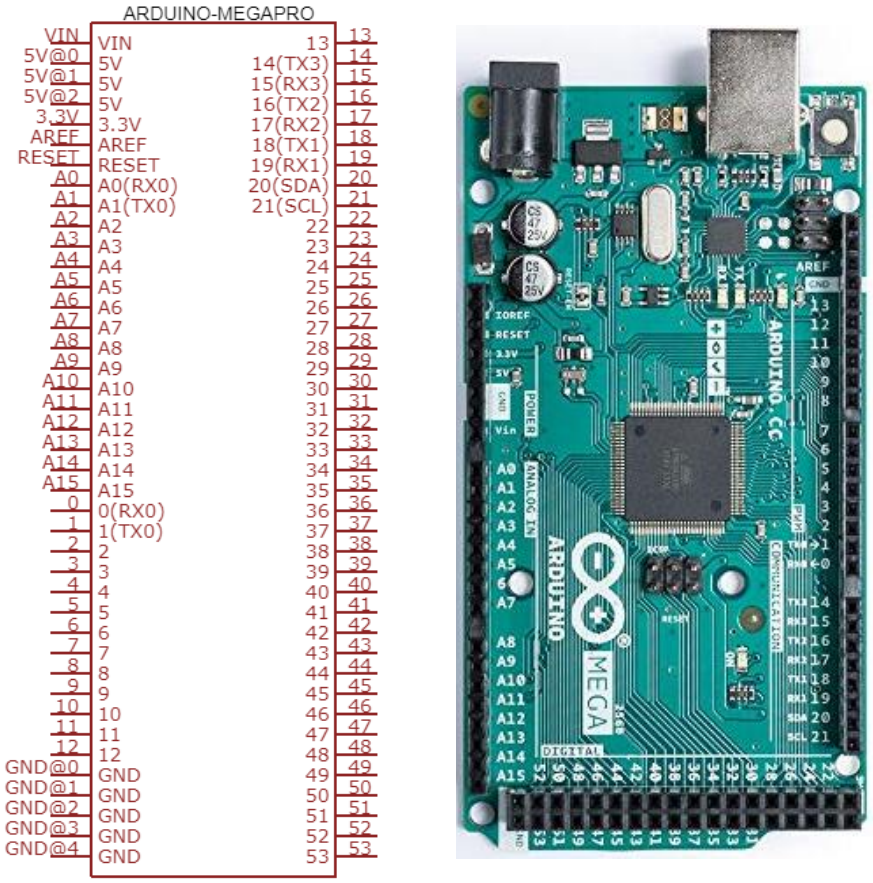


Figura 29.-Placa y simbología del ARDUINO® Mega con µprocesador ATmega2560

4.2. Reguladores de Tensión LM7812, LM7809 y LM7805

Los siguientes circuitos constan de un regulador de tensión LM7812, LM7809 y LM7805.

Estos circuitos sirven para mantener una tensión controlada de 12V, 9V o 5V respectivamente para alimentar diferentes módulos del circuito, así como la misma placa de ARDUINO®. Pese a que el vehículo está alimentado por una batería que proporciona 12 V teóricos, tras varios ensayos, se observa que es bastante irregular según las circunstancias. Por eso se ha propuesto la implementación del regulador de 12V para controlar la tensión y fijarla en 12V reales.

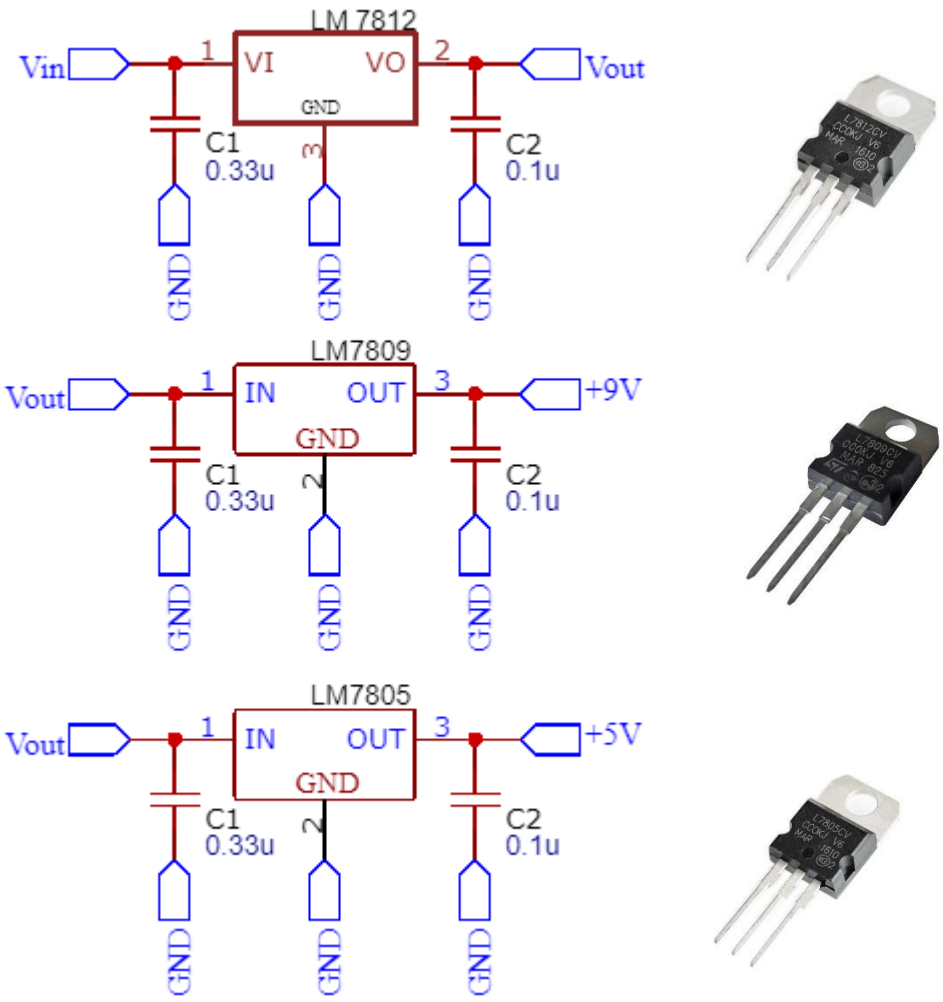


Figura 30.-Esquemático y reguladores de tensión de 12V, 9V y 5V

4.3. Sensor de Temperatura y Humedad DHT11

El DHT11 permite realizar la medición simultánea de temperatura y humedad. Este sensor dispone de un procesador interno que realiza el proceso de medición, proporcionando la medición mediante una señal digital, por lo que resulta muy sencillo obtener la medición desde un µprocesador como ARDUINO®.

Dadas las adversas circunstancias a las que se va a someter el vehículo (ya que puede llegar a usarse en temperaturas realmente bajas) es necesario llevar un control de temperatura del dispositivo por si entrara en valores que hicieran peligrar el correcto funcionamiento del mismo. Por otro lado, si el sensor de humedad (situado en la base de la caja hermética, por debajo del resto del circuito) registrase valores drásticamente superiores, es señal de que posiblemente haya entrado algo de agua y sería necesaria su inmediata retirada del mar.

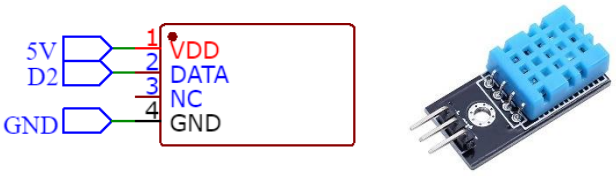


Figura 31.-Esquemático y módulo DHT11

4.4. Acelerómetro ADXL335

El ADXL345 es un acelerómetro µmecanizado capacitivo de 3 ejes independientes (3 DOF), que puede ser fácilmente conectado a un autómata o µprocesador como ARDUINO®. Se trata de un sensor µmecanizado (MEMS) capacitivo que detecta la aceleración en los ejes X, Y y Z. También es posible determinar la orientación del sensor, gracias a la acción de la fuerza de la gravedad. El µcontrolador de ARDUINO®, registra los valores de los diferentes ejes a través de las diferentes salidas del módulo.

Este módulo nos va a permitir determinar la orientación de nuestro vehículo, pudiendo también determinar si el vehículo ha volcado a causa de condiciones externas. Si fuese éste el caso, el usuario sería informado a través de una señal en la interfaz.

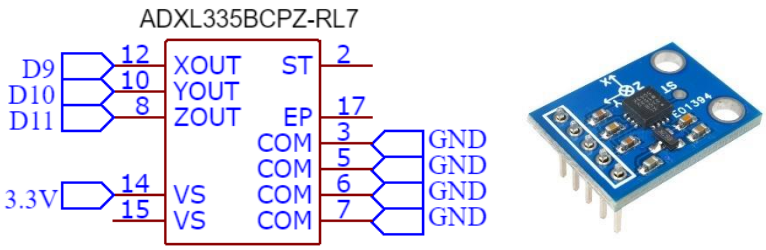


Figura 32.-Esquemático y módulo ADXL335

4.5. Brújula HMC5883L

EL HMC5883L es un magnetómetro de 3 ejes. Con éste se pueden leer las componentes del campo magnético presente, de esta forma conociendo la dirección del campo magnético terrestre es posible calcular la orientación con respecto al norte magnético de la tierra, esto siempre y cuando el sensor no esté expuesto a algún campo magnético externo u algún objeto metálico que altere el campo magnético terrestre.

Este módulo permite la obtención de la orientación respecto a Polo Norte terrestre, para así poder seguir trayectorias y poder realizar desplazamientos con éxito.

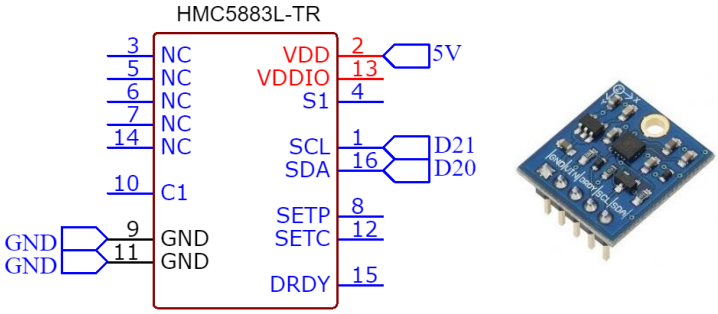


Figura 33.-Esquemático y módulo HMC5883L

4.6. Sensor de Corriente ACS712

Este módulo basado en el circuito integrado ACS712 de Allegro MicroSystems® permite medir la cantidad de corriente que fluye a través de un circuito de corriente alterna (AC) o corriente directa (DC).

El método es a través de un sensor de efecto hall que proporciona un voltaje de salida proporcional a la corriente que fluye en el circuito. El trayecto para la medida de corriente es por el interior del circuito integrado y se encuentra aislado del circuito de procesamiento.

Este módulo permite medir la cantidad de corriente que proporciona la batería, y así poder determinar la autonomía restante del vehículo.

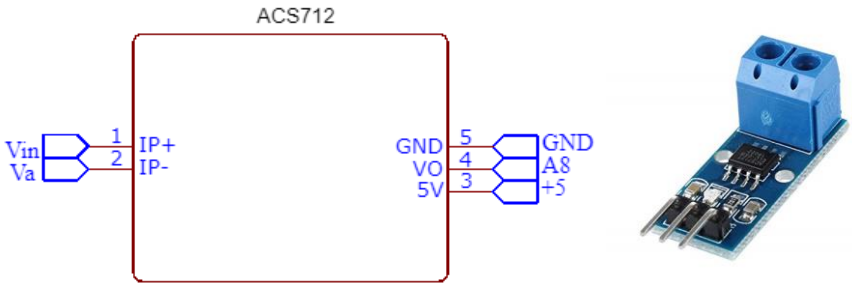


Figura 34.-Esquemático y módulo AC712

4.7. GPS NEO 6M

Los dispositivos NEO-6 son una familia de receptores fabricados por U-Blox®, que pueden ser conectados con facilidad a un procesador como ARDUINO®. Son localizadores GPS con una precisión en posición de 2.5 m, en velocidad 0,1 m/s y en orientación 0,5º, valores más que aceptables para un sistema de posicionamiento GPS.

Este módulo permite la obtención de la localización a tiempo real del vehículo.

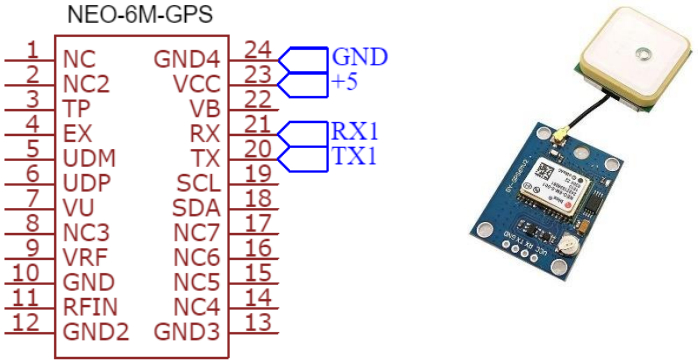


Figura 35.-Esquemático y módulo NEO-6M-GPS

4.8. Driver MDD10A

MDD10A es la versión de doble canal de MD10C que está diseñada para controlar 2 motores de corriente continua con escobillas con alta corriente de hasta 10A. Al igual que MD10C, MDD10A también admite señal PWM. Utiliza componentes que ofrecen un tiempo de respuesta más rápido y elimina el desgaste del relé mecánico. Puede soportar voltajes de 5VDC a 30VDC [9].

Después de valorar diversas opciones para el control de motores, se ha considerado el driver MDD10A como la mejor opción. Se ha descartado el control por relés debido a su desgaste mecánico. Por otro lado, este driver permite grandes cantidades de corriente, ideal en caso de picos de consumo, permite variar el sentido de los motores, y permite el control simultáneo de ambos motores.

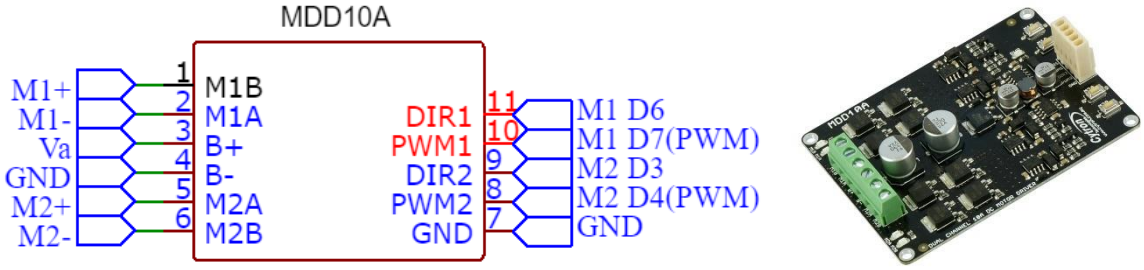


Figura 36.-Esquemático y módulo MDD10A

4.9. Módulo de Comunicación Inalámbrica XBee™

Los módulos XBee™ son módulos integrados que proporcionan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan el protocolo de red llamado IEEE 802.15.4 para crear redes *FAST POINT-TO-MULTIPOINT* (punto a multipunto); o para redes *PEER-TO-PEER* (punto a punto). Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible.

La elección de XBee™ como integrado para la comunicación ha sido a causa del conocimiento adquirido respecto a éste anteriormente, y su relativa facilidad tanto en conexión como en programación en la transmisión de datos.

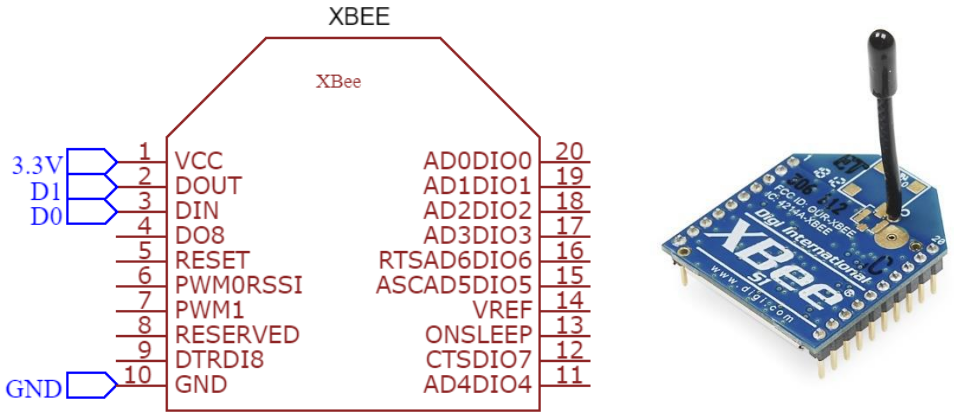


Figura 37.-Esquemático y módulo XBee™

4.10. Motores

Los motores utilizados para nuestro vehículo son el modelo C-6044.

Es un motor DC con gran potencia (47W) que permite una gran revolución de los ejes en caso de que fuera necesario. Por otro lado, no tiene un gran consumo a media potencia, proporcionando unas 19.500 revoluciones por minuto (rpm) en su mayor eficiencia energética [10].

Figura 38.-Motor C-6044

4.11. Baliza y Bocina

Los siguientes circuitos constan de un relé controlado por una salida digital de ARDUINO® para activar y desactivar la baliza o bocina de posición obligatorias que lleva el vehículo. Consta de la salida digital de 5V el control del relé y la alimentación de 12V para activar el actuador (Relé normalmente abierto).

Por otro lado, en el lado del interruptor normalmente cerrado consta de un LED con su respetiva resistencia que verifica el correcto funcionamiento del relé.

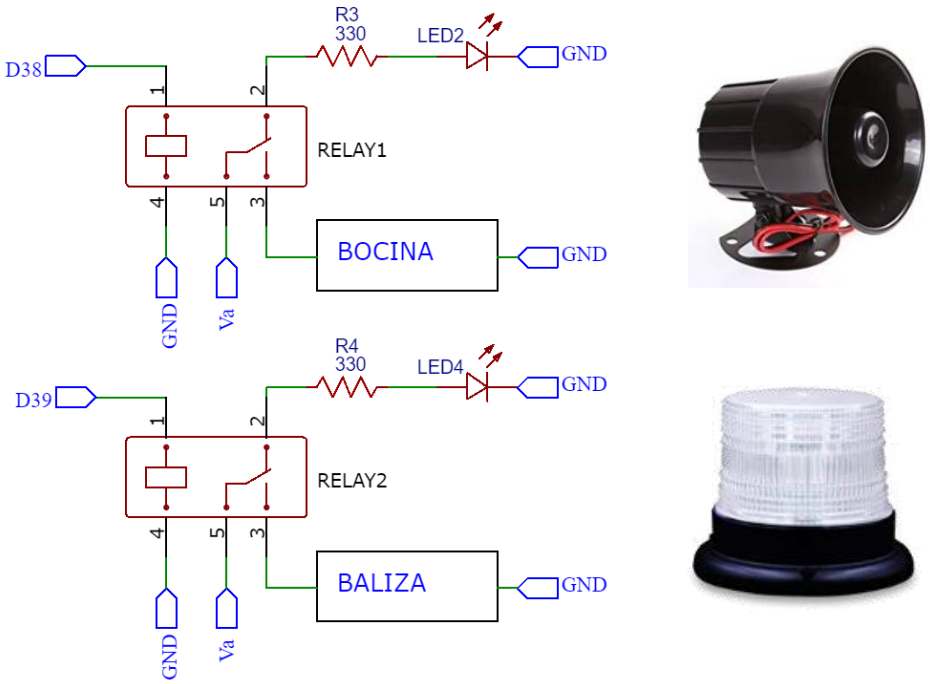


Figura 39.-Esquemático y actuadores (baliza y bocina) controladas por relé

4.12. Módulos Soldados

Con tal de facilitar el montaje final, se desarrolló un método de soldadura el cual permite unir componentes por módulos, los cuales finalmente derivarán al módulo de control o al de alimentación. Los componentes de todos los módulos (excepto el de alimentación) son reemplazables, ofreciendo así una fácil reparación en el caso de que alguno de estos fallara. Los módulos desarrollados son los siguientes:

I) Módulo 1: Control

Este módulo es el encargado de recibir todas las salidas de información de los sensores, así como enviar las ordenes necesarias para cada situación. Es el módulo cerebro, el cual contiene la placa de ARDUINO® y las salidas y entradas necesarias para los sensores y actuadores.

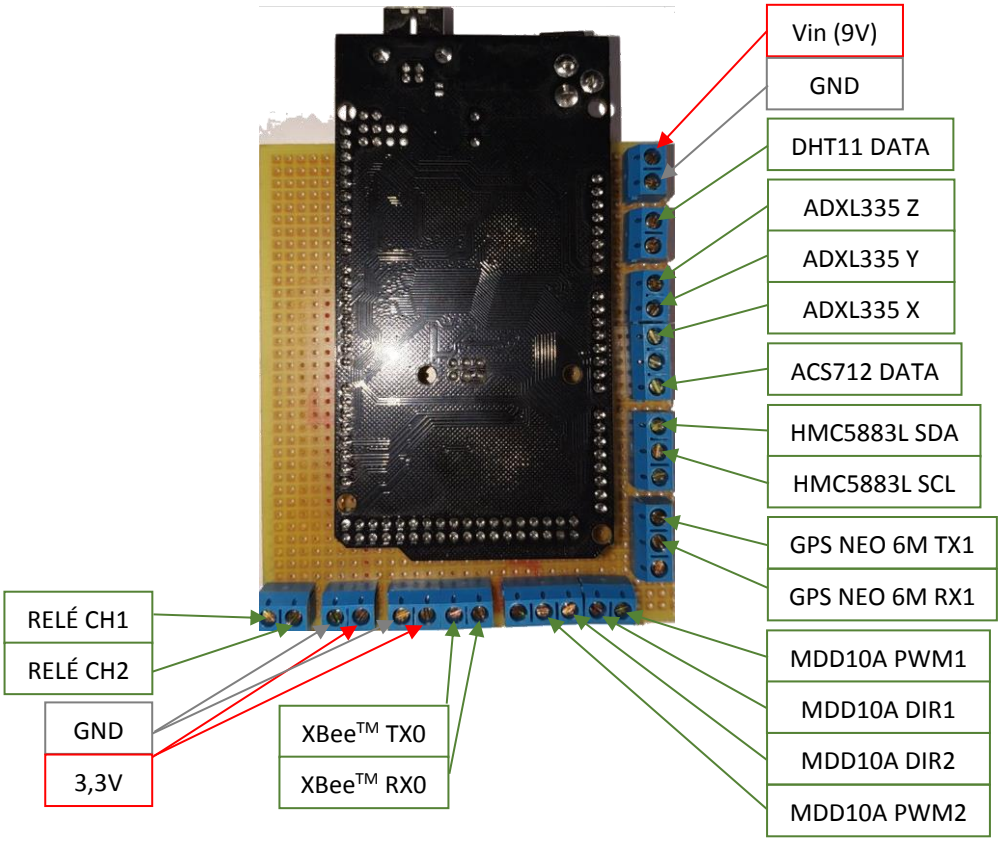


Figura 40.-Módulo de control con sus respectivos pines

II) Módulo 2: Alimentación

La fuente de alimentación principal del circuito es de 12V y amperaje de 8Ah, pero no todos los componentes del prototipo requieren de la misma alimentación, por tanto, se ha unido en un solo módulo la alimentación necesaria para la mayoría de componentes, que son 12V, 9V y 5V.

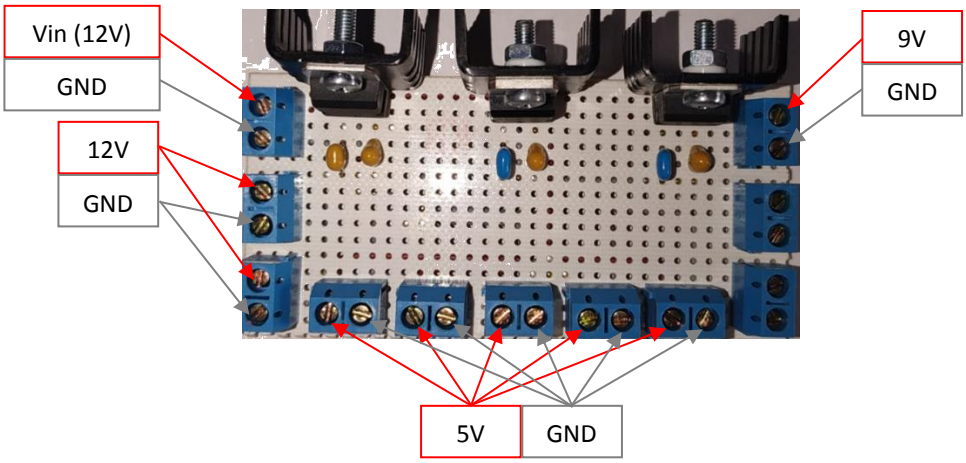


Figura 41.-Módulo de alimentación con sus respectivos pines

III) Módulo 3: Posición

La intención de este módulo es agrupar en uno todos los sensores que hacer referencia a la posición del vehículo puesto que algunos de ellos requieren cierto posicionamiento para devolver el valor correcto. En este módulo nos encontramos el acelerómetro, la brújula y el GPS.

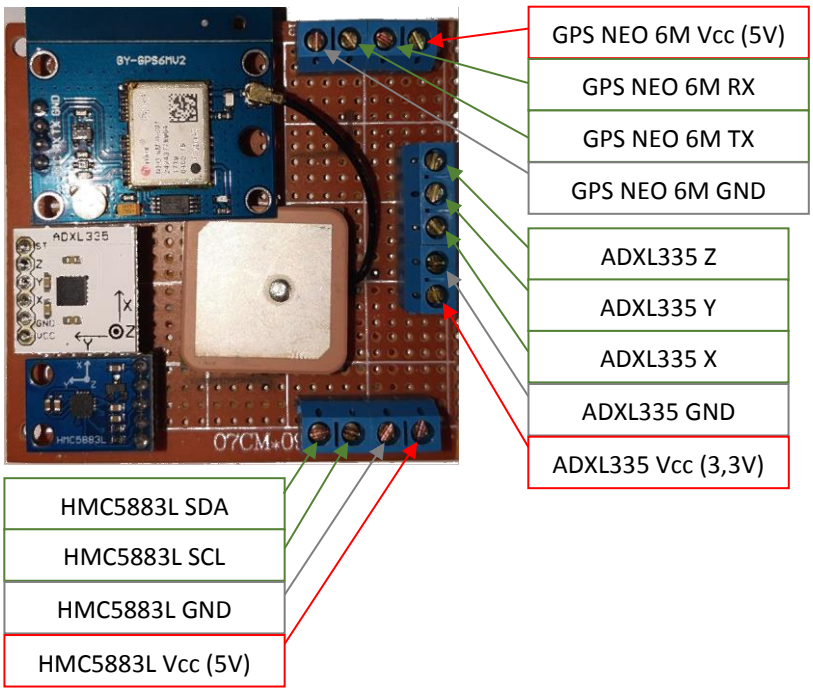


Figura 42.-Módulo de posición con sus respectivos pines

IV) Módulo 4: Inverso

Se trata del módulo que contiene el sensor de corriente y el conjunto de relés. Estos módulos estaban diseñados para disponerse con orientación supina, pero la soldadura complicaba la operación de mantenerlos en el mismo estado. Por lo tanto, se optó por soldar la placa de unión de forma inversa y así facilitar la colocación dentro del diseño final.

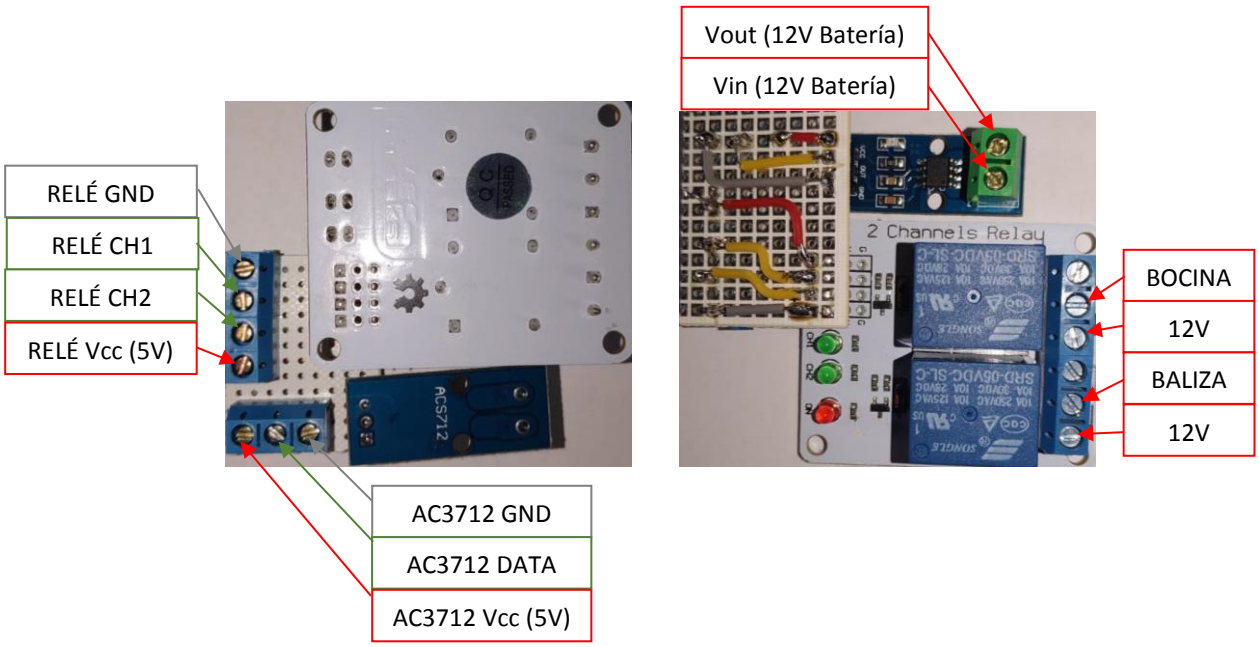


Figura 43.-Módulo inverso con sus respectivos pines

V) Módulo 5: Vertical

Los componentes de este módulo son los únicos que requieren de una orientación vertical. Estos son el sensor de temperatura y el XBee™, por lo tanto, se implantó una pequeña placa la cual estaba destinada a mantener unidos estos dos componentes.

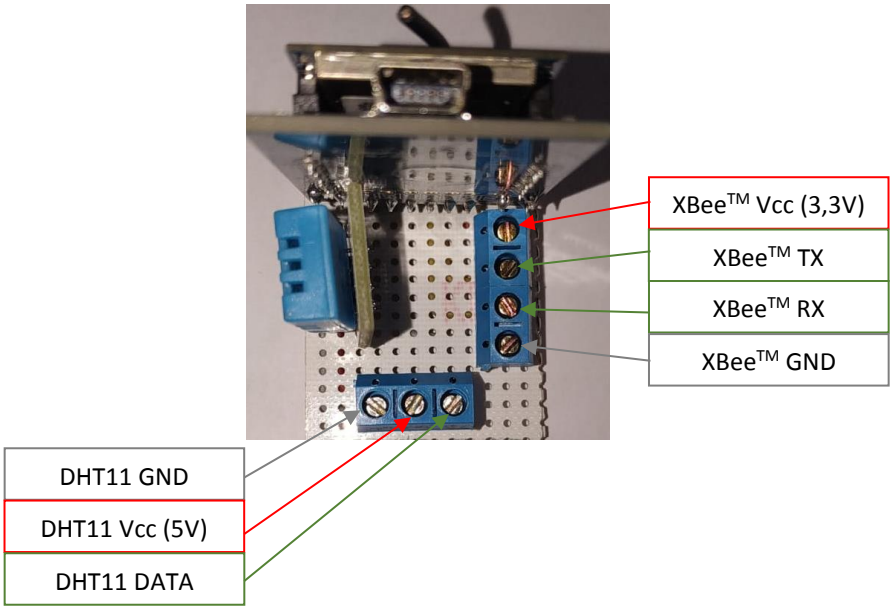


Figura 44.-Módulo vertical con sus respectivos pines

Finalmente, y retomando el punto anterior “3.4. Ensamblaje Definitivo”, la implementación de todos los módulos expuestos en el interior de sus respectivas cajas estancas se puede apreciar con más detalle en la siguiente figura:

Figura 45.- Módulos soldados en el interior de las cajas estancas

5. SOFTWARE

Como ya se ha comentado anteriormente, el desarrollo del proyecto se basa en la plataforma ARDUINO®, el motivo de la elección principalmente es su código abierto y su programación IDE basada en el lenguaje C++, sobre el cual consideramos haber adquirido cierto conocimiento a lo largo del transcurso del Grado.

La programación realizada mediante la plataforma ARDUINO® deberá poder ser gobernada a tiempo real mediante realimentaciones negativas, lazos cerrados o incluso a través del usuario del vehículo de apoyo desde una embarcación tripulada. Del mismo modo se pretende recibir datos constantemente de la situación en la que se encuentra el vehículo de apoyo para poder llevar un registro de datos y así poder tomar decisiones consecuentes a cada necesidad. Pese a que todo lo anteriormente comentado se podría realizar únicamente mediante la plataforma ARDUINO® no resultaría intuitivo para un usuario que no tuviera conocimientos de programación. Por lo tanto, y con intención de hacer la aplicación accesible a cualquier tipo de usuario, se ha desarrollado una interfaz basada en LabVIEW™ de la compañía National Instruments®. El principal motor impulsor de esta elección es el conocimiento establecido recientemente por parte de los dos integrantes del grupo, hecho que nos ha permitido poder retener y ampliar horizontes respecto al método de programación de LabVIEW™ sin haber sido necesario remontarnos a otros métodos de desarrollo y programación de interfaces aprendidos anteriormente, como es el caso de la programación con lenguaje *Csharp* (C#), o haber tenido que afianzar conocimientos sobre otros métodos desde cero.

Así como el entorno IDE de ARDUINO®, LabVIEW™ se caracteriza por su facilidad de uso, su versatilidad y la relativa rapidez de programación una vez establecidos los conocimientos básicos del funcionamiento de ambos.

Por otra parte, la principal característica de LabVIEW™ es el ser una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando así su comprensión. Al tener ya prediseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto, con lo cual en vez de gastar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada programa en LabVIEW™ (VI) consta de dos partes diferenciadas:

- Panel frontal: es la interfaz con el usuario, se utiliza para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios pueden observar los datos del programa actualizados en tiempo real. En esta interfaz se definen los controles e indicadores.

- Diagrama de Bloques: es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan [4].

En el panel frontal se encuentran todo tipo de controles o indicadores, donde cada uno de estos elementos tiene asignado en el diagrama de bloques una terminal, es decir, el desarrollador puede diseñar un proyecto en el panel frontal con controles e indicadores, donde estos elementos serán las entradas y salidas que interactuarán con la terminal del VI. Finalmente será el usuario de la aplicación quien podrá gobernar cómo y cuando quiera la misma.

5.1. Comunicación ARDUINO® - LabVIEW™

La comunicación entre el software desarrollado en la plataforma ARDUINO® con la interfaz del panel frontal de LabVIEW™ se puede realizar de diversas maneras. Con tal de facilitarnos el trabajo a la hora de llevar a cabo pequeños programas y para familiarizarnos con uno de los métodos de comunicación entre ambas plataformas, optamos por el uso la herramienta LIFA (LabVIEW™ *Interface for* ARDUINO®) *Toolkit* [11].

La interfaz de LabView™ para ARDUINO® (LIFA) *Toolkit* es una herramienta gratuita que se puede descargar desde el servidor de NI (National Instruments®) y que permite a los usuarios de ARDUINO® adquirir datos del µcontrolador ARDUINO® y procesarlos en el entorno de programación gráfica de LabVIEW™. Este kit de herramientas de código abierto está hecho para la personalización, permitiendo a los usuarios crear controladores personalizados para sus sensores [12].

Al instalar LIFA *Toolkit* en nuestro LabView™, automáticamente se añadirán las librerías de ARDUINO®[15], de las cuales, las que nos interesan se podrán dividir en cuatro grupos:

- Inicio: inicializa una conexión a un ARDUINO® ejecutando la LIFA. Es aquí donde se configurará principalmente el puerto de entrada y salida de datos, así como el tipo de placa que se está utilizando el tipo de conexión. Adicionalmente se pueden configurar elementos como el error de entrada o la velocidad de transmisión.

[15]Cabe destacar que, para el correcto funcionamiento de la comunicación entre ARDUINO® y LabView™ mediante LIFA *Toolkit*, previamente se debe cargar el programa LIFA_*base* en la placa a utilizar.

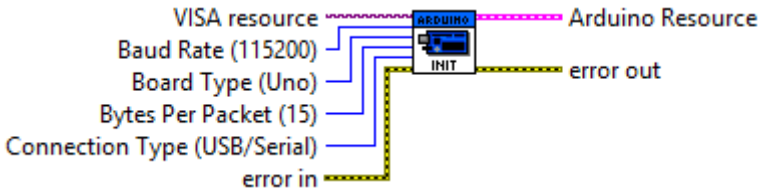


Figura 46.-Herramienta para el inicio de la comunicación LabVIEW™-ARDUINO® y sus pines de conexión

- Configuración de pines: mediante esta librería podemos seleccionar los pines a utilizar en nuestro programa; inicializarlos, seleccionar si son analógicos o digitales, si son entradas o salidas y gobernarlos.

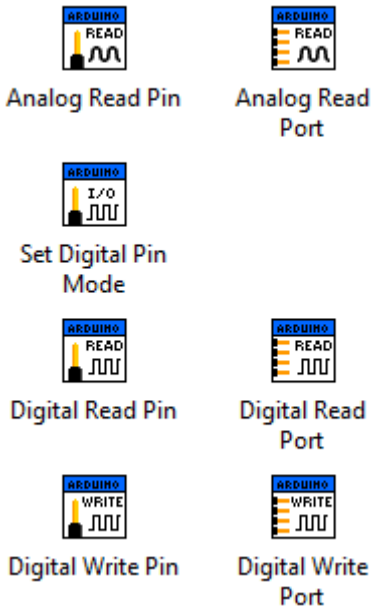


Figura 47.- Herramientas para la configuración de pines

- Sensores: este *Toolkit* contiene un seguido de sensores previamente configurados con tal de facilitar al usuario el uso de diversos tipos de sensores, así como *displays*, LCDs, e incluso motores paso a paso y servomotores.

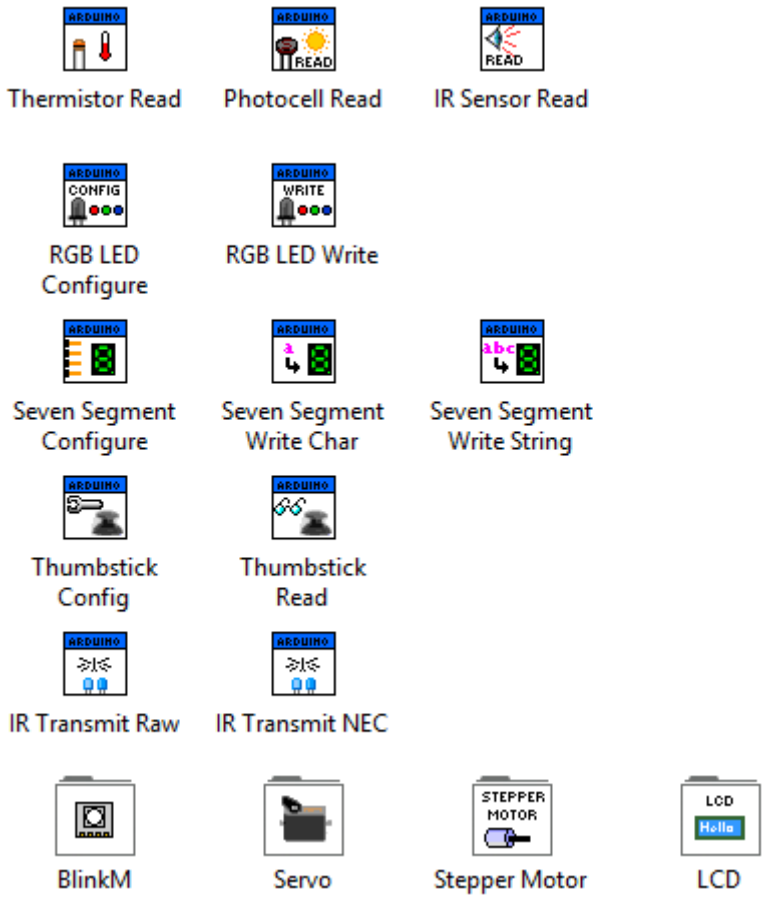


Figura 48.- Herramientas para la configuración de sensores y actuadores

- Cierre: es el encargado de cerrar la comunicación activa con un ARDUINO®.

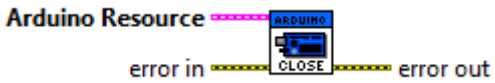


Figura 49.- Herramienta para el cierre de la comunicación LabVIEW™-ARDUINO® y sus pines de conexión

De todos modos, LIFA *Toolkit* no es el único método de comunicación entre la plataforma ARDUINO® y la interfaz del panel frontal de LabVIEW™, algunos de estos, como el explicado posteriormente, no son tan intuitivos, pero se compensan con la versatilidad de aplicaciones que se pueden llegar a realizar y la complejidad de las mismas.

LabVIEW™ ofrece por defecto la sub-librería *SERIAL*, incluida dentro de la librería *Instruments I/O*. Esta librería es de gran utilidad debido a los elementos que contiene, los cuales sirven principalmente para conectar el puerto serie de entrada o salida de

ARDUINO® con el procesamiento de información a nivel visual del panel frontal de LabVIEW™.

Figura 50.- Elementos destacados de la sub-librería "SERIAL"

Estas "cajas" (modo coloquial de denominar a las funciones en el contexto de LabVIEW™) se pueden agrupar principalmente en cuatro grupos:

- Inicio/Configuración: principalmente se utiliza para inicializar el puerto serie especificado por el *VISA resource name* a la configuración específica.

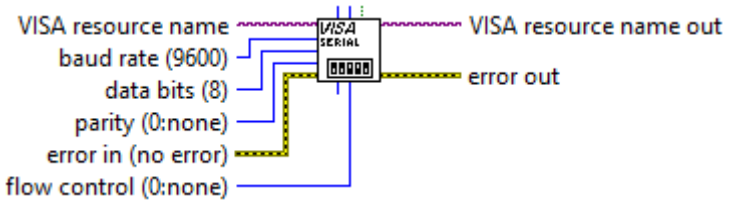


Figura 51.- Herramienta para el inicio de la comunicación LabVIEW™-ARDUINO® y sus pines de conexión

- Bytes at Port: se trata de una caja auxiliar encargada de leer la cantidad de bytes que se encuentran en el puerto serie en ese momento y devolver el valor. Esta caja se suele utilizar a menudo para la lectura y la escritura sobre el puerto en cuestión.

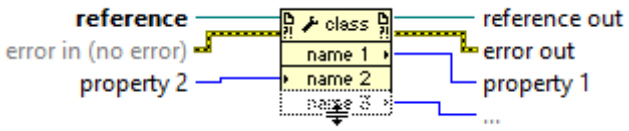


Figura 52.- Herramienta auxiliar y sus pines de conexión

- Read/Write: lee o escribe el numero especificado de bytes a través del dispositivo o de la interfaz especificada por el *VISA resource name*.

Figura 53.- Herramienta "read" y sus pines de conexión

Figura 54.-Herramienta "write" y sus pines de conexión

- Cierre: cierra la sesión con el dispositivo especificado.

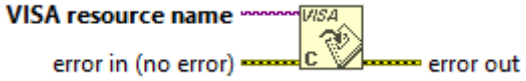


Figura 55.- Herramienta para el cierre de la comunicación LabVIEW™-ARDUINO® y sus pines de conexión

La principal diferencia entre este método de comunicación sin el uso de LIFA *Toolkit* con el explicado anteriormente es la programación en sí. Mientras que utilizando la herramienta LIFA, la placa de ARDUINO® se comportaba como un DAQ (adquiría la señal, pero no la procesaba) dónde todo el peso de la programación recaía sobre el diagrama de bloques de LabVIEW™, utilizando el método de la sub-librería *SERIAL* la programación se puede realizar tanto desde ARDUINO® como desde LabVIEW™, permitiendo así realizar aplicaciones más complejas puesto que podemos, por ejemplo, recoger los datos a tiempo real con ARDUINO®, mientras los procesamos y mostramos con LabVIEW™. Por otro lado, podríamos recogerlos y procesarlos con ARDUINO® mientras que utilizamos LabVIEW™ tan solo para realizar el muestreo final.

En este último caso la versatilidad se convierte en una ventaja.

Con tal de familiarizarnos con ambos métodos de comunicación, estos han sido probados a lo largo del desarrollo del proyecto en aras de escoger cual es el óptimo y descartar los métodos innecesarios. Con este fin (y con el de comprobar el correcto funcionamiento del diseño individualmente) se han realizado diversos ensayos de comunicación.

5.2. Ensayos

Para realizar ensayos de diversos componentes y así poder comprobar el correcto funcionamiento de estos, se han realizado pequeños programas que, además de comunicar LabVIEW™ con ARDUINO® han sido configurados para llevar a cabo una función concreta.

Algunos de estos programas han sido realizados con el LIFA *Toolkit* (debido a la facilidad para realizar pequeños programas no extremadamente complejos con una programación simple basada principalmente en LabVIEW™) así como otros programas no han requerido el uso de esta[16].

5.2.1. Control de Relés

El sistema para el gobierno de algunos de los actuadores del vehículo (baliza, bocina, etc.) se realiza mediante el uso de relés.

Estos relés son los encargados de cerrar el circuito y dejar circular la corriente a través de cada elemento. Con el fin de poder controlar en la totalidad el encendido y apagado de cada uno de los relés y a su vez comprobar el correcto funcionamiento de ambos, se ha realizado un programa para gobernar dos relés simultáneamente en LabVIEW™, cuyo diagrama de bloques es el siguiente:

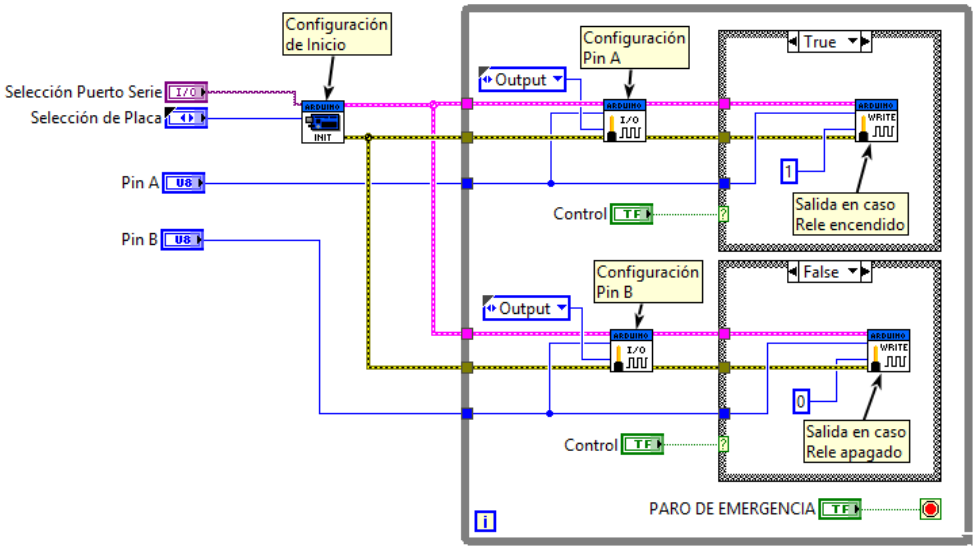


Figura 56.- Diagrama de bloques para el control de relés

[16]LIFA *Toolkit* es una herramienta de programación extremadamente útil cuando se trata de programas cortos o que no requieren de una comunicación externa con sensores o dispositivos sobre los cuales no tiene su propia librería. En estos últimos casos, es preferible centrar la programación en la plataforma ARDUINO® con tal de delegar en LabVIEW tan solo la interfaz de la aplicación.

Como se puede apreciar en la anterior figura, los relés son controlados mediante un interruptor (Control), el cual da una salida "*True*" o "*False*", en función de esta salida y mediante la librería *Digital Write* de LIFA, le enviamos un valor de "*1*" o "*0*" en el caso de que queramos el relé encendido o apagado. El control de este programa se encuentra en el panel frontal y tiene la disposición siguiente:

Figura 57.- Interfaz del panel frontal para el control de relés

Es importante que la interfaz contenga los elementos justos y necesarios para la comprensión de la aplicación. En este caso tan solo hace falta seleccionar el tipo de placa ARDUINO® o compatible con la que se va a comunicar LabVIEW™, el puerto por el cual se va a realizar la comunicación y los pines a los cuales se van a conectar los relés. Una vez rellenados todos estos campos el control se realiza activando los interruptores de "Control" situados bajo el pin de cada relé. Como se puede apreciar en el diagrama de bloques, el programa se realiza en bucle al estar encerrado dentro de una estructura *WhileLoop*, por lo tanto, el único método de parar el programa es mediante el botón de paro de emergencia, situado en la esquina inferior derecha de la interfaz.

Cabe remarcar que este programa no tendrá una aplicación real en el prototipo final, en cambio nos servirá como toma de contacto con los métodos de comunicación de LabVIEW™ y ARDUINO® así como para mostrar al lector los requisitos básicos de la misma. De todos modos, y aunque no vaya a tener una aplicación como tal, si que nos servirá de apoyo en el momento en el que necesitemos actuar sobre las salidas digitales de la placa.

5.2.2. Consumo de Motores

Con tal de conocer la potencia consumida por los motores, realizamos un nuevo programa para realizar un ensayo de consumo sobre estos. Esta vez el programa no tendrá ningún tipo de aplicación práctica sobre el prototipo final, pero la información de sobre-picos o alteraciones en el funcionamiento de estos elementos es crucial para poder establecer medidas preventivas en el caso de perturbaciones. Es por eso que, mediante un circuito tan simple como un divisor de tensión desarrollamos un programa que nos grafica la tensión, corriente y potencia del motor bajo ensayo.

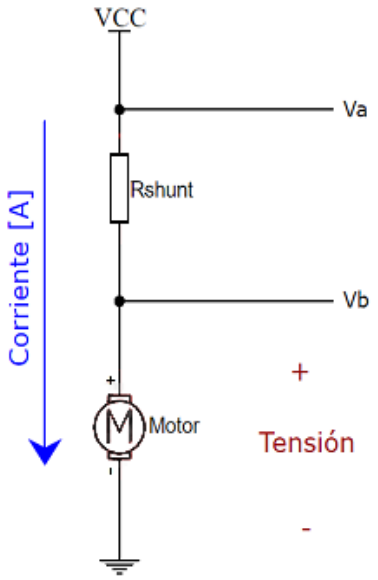


Figura 58.- Esquemático para el ensayo de consumo de motores

El diagrama de bloques que corresponde a esta toma de datos se corresponde al siguiente:

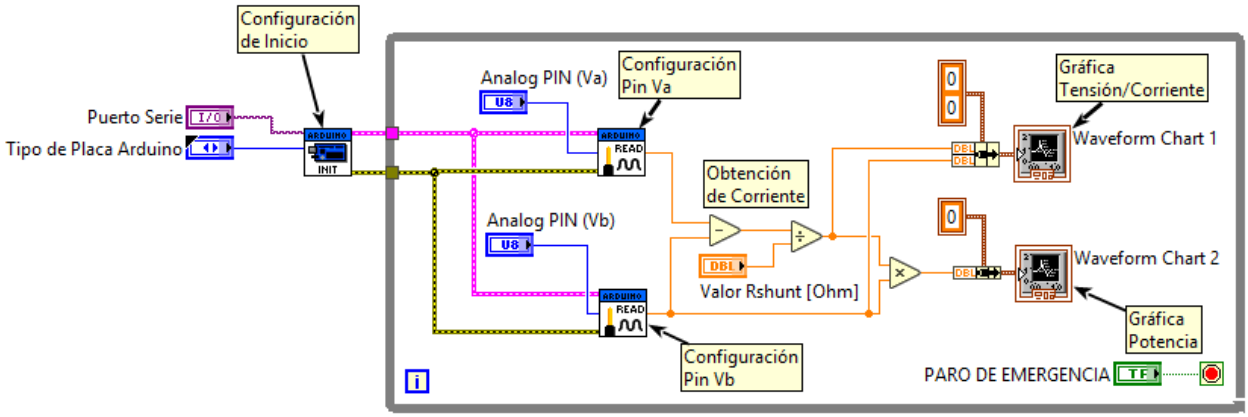


Figura 59.- Diagrama de bloques para el ensayo de consumo de motores

Como se puede apreciar, para realizar el cálculo de la corriente se toman los valores de la tensión en el punto "Va" y en el punto "Vb", la diferencia de estas dos tensiones nos dará como resultado la caída de tensión en la resistencia *shunt*. Sabiendo el valor de esta resistencia la ley de Ohm nos dice:

$$I = \frac{V}{R} \quad \rightarrow \quad I = \frac{V_{CC}}{R_{Shunt}} \qquad (1)$$

Por lo tanto, ya podemos obtener el valor de la corriente que circula por el motor. Por otro lado, la tensión del motor es directamente "Vb", es decir, no se necesita ningún tipo de cálculo para obtenerla. Así pues, para graficar ambos valores, tan solo hace falta introducirlos en la función *Waveform Chart* de LabVIEW™. En el caso de la gráfica de potencia, y sabiendo la ecuación de la potencia eléctrica:

$$P = V \cdot I \quad \rightarrow \quad P = V_b \cdot I \qquad (2)$$

Tan solo será necesario graficar el producto de los dos valores que hemos obtenido anteriormente. El control de esta aplicación se establece según la siguiente interfaz:

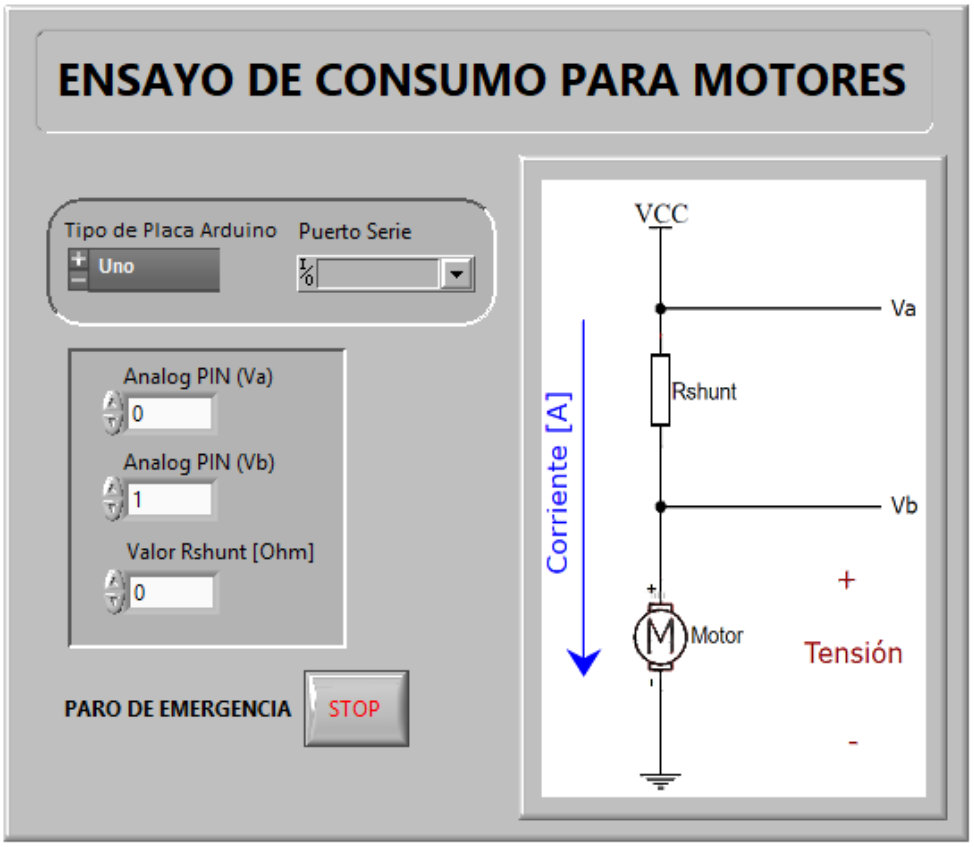


Figura 60.-Interfaz del panel frontal para el ensayo de consumo de motores

De nuevo, tan solo es necesario colocar en la interfaz los elementos que son completamente necesarios para la configuración y el control de la aplicación. En este caso, con tal de servir de ayuda a un posible usuario externo que necesite la aplicación, se ha optado por introducir el esquemático del circuito a realizar con toda la información necesaria para su montaje. Del mismo modo se nombra cada Pin para simplificar y para facilitar la comprensión de la aplicación para el usuario.

Volvemos a ver la selección del tipo de placa y del puerto serie para la comunicación. El único elemento que podría no ser necesario en la interfaz podría ser el valor de la resistencia de *shunt*, pero el hecho de haber probado varias y de que la aplicación pueda servir para más de un usuario, simplifica la variación de la misma.

Con el fin de poder visualizar el registro de datos que se está llevando a cabo a tiempo real, se han añadido dos gráficos, uno de ellos muestra los valores de tensión del motor y de la corriente que circula a través de él, el otro muestra el consumo de potencia del motor. Ambos gráficos se han realizado durante un corto periodo de tiempo (alrededor de 20 segundos) con tal de obtener los valores principales de arranque, estabilización y frenada. Todos estos datos son de nuestro interés puesto que; podemos apreciar la curva de tensión al iniciar el movimiento, el pico de corriente consumida en el momento de arranque (que siempre suele ser más elevado) y la potencia que consume el motor dadas unas entradas específicas.

Los resultados de este ensayo forman parte del apartado “6.1. Ensayos Experimentales”.

5.2.3. Temperatura y Humedad

Para la adquisición de los datos de temperatura y humedad se ha utilizado el sensor DHT11 como ya se ha comentado anteriormente. Se trata de un sensor ampliamente utilizado en la programación mediante la plataforma de ARDUINO® y, en este caso, no disponemos de ninguna librería de LabVIEW™ que nos permita la fácil implementación dentro de la programación por bloques que caracteriza este entorno. Es por esto que el método de comprobación del correcto funcionamiento de este elemento se ha llevado a cabo utilizando la sub-librería *SERIAL* que se ha explicado en el punto “4.1. Comunicación ARDUINO®-LabVIEW™”.

Al ser un componente tan utilizado se han llevado a cabo librerías en la plataforma IDE, por lo tanto, la programación en ARDUINO® resulta muy sencilla, en menos de 20 líneas de código obtenemos los valores de Temperatura (ºC) y Humedad relativa (%) y, lo que es más importante, son escritos en el Monitor Serie de ARDUINO®.

```
/**********************************/
/*      TEMPERATURA Y HUMEDAD      */
/**********************************/

//Librería del sensor//
#include<DHT.h>

//Pin donde irá conectada la señal del sensor//
#define DHTPIN 2

//Definición del tipo de sensor//
#define DHTTYPE DHT11

//Instrucción propia de la librería//
DHT dht(DHTPIN, DHTTYPE);

voidsetup() {
  Serial.begin(9600);
  dht.begin();
}

voidloop() {
  float h =dht.readHumidity();
  float t =dht.readTemperature();

  //Muestreo de datos//
  Serial.print("H");
  Serial.print(h);
  Serial.print("\t");
  Serial.print("T");
  Serial.print(t);
  Serial.println();
}
```

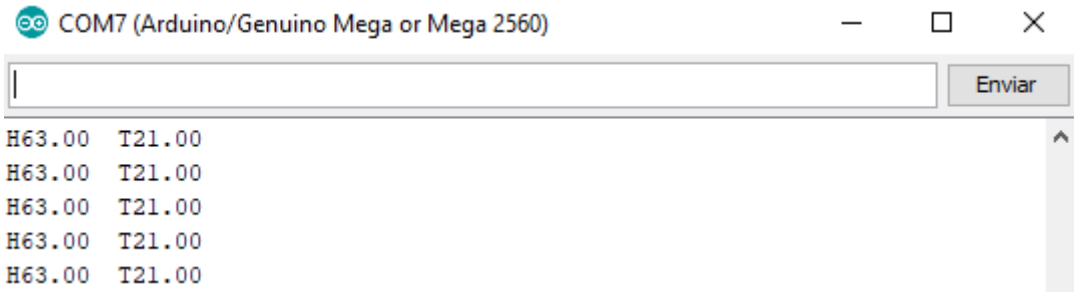


Figura 61.- Datos escritos en el Monitor Serie de ARDUINO®

Una vez obtenidos los valores de Temperatura y Humedad en el monitor serie, es necesario transferir esta información a LabVIEW™ y fraccionar el *string* de datos con tal de diferenciar cada una de las variables que queremos mostrar. Para poder llevar a cabo este fraccionamiento es necesario conocer la longitud del *string* y donde se encuentran los valores los cuales queremos obtener:

Figura 62.-*String* mostrado en el Monitor Serie (en rojo los valores de inicio de cada variable)

Sabiendo los valores de inicio de cada variable como “1” y “8”, podemos realizar el diagrama de bloques del programa de LabVIEW™:

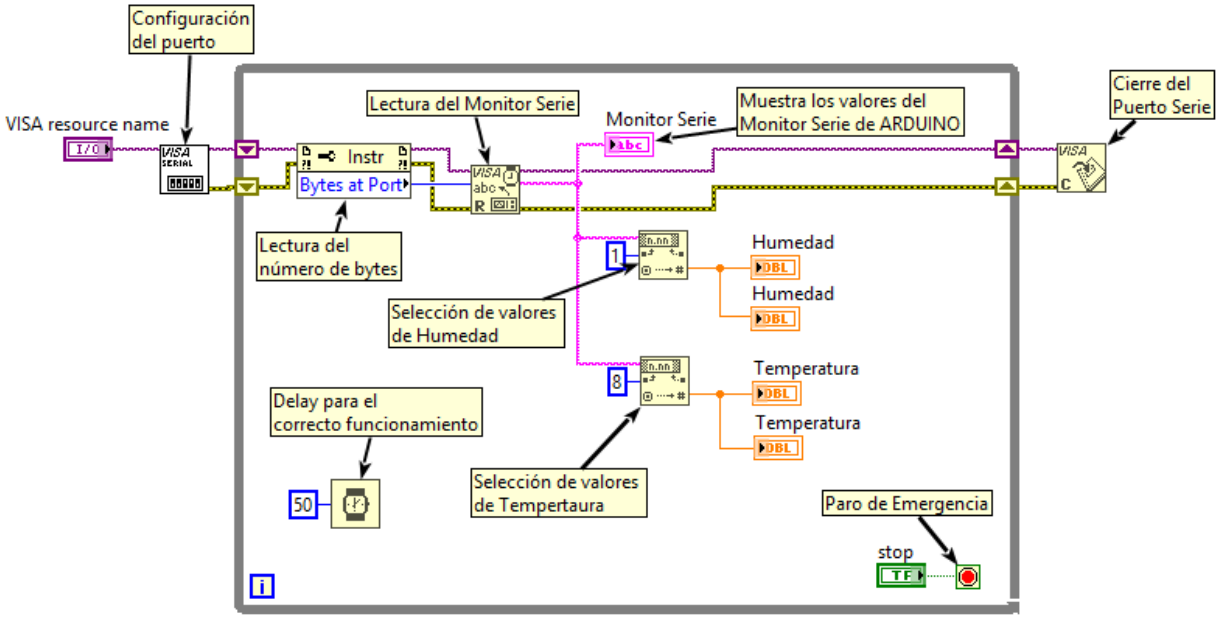


Figura 63.- Diagrama de bloques para el ensayo de obtención de Temperatura y Humedad

Como se puede apreciar, es en el momento de seleccionar los valores de Temperatura y Humedad cuando los valores anteriormente establecidos se utilizan para elegir el punto en el cual se empiezan a tomar los datos deseados.

Finalmente, en la interfaz se muestra, como es de costumbre, un panel de configuración del puerto, y junto a él, el *string* de datos mostrados en el Monitor Serie de ARDUINO®, a su derecha, en dos *Slides* diferenciados, los valores de Temperatura y Humedad.

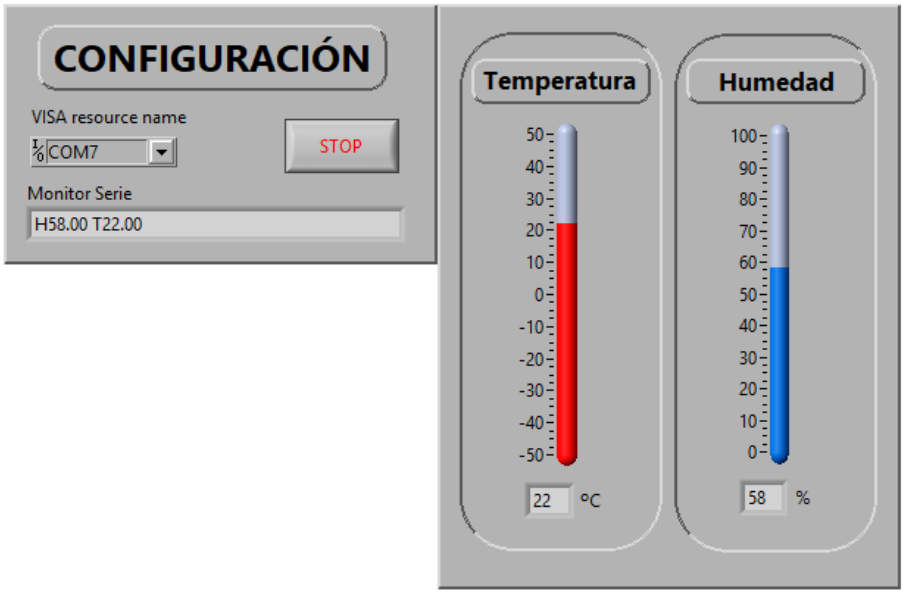


Figura 64.- Interfaz para el ensayo de obtención de Temperatura y Humedad y resultados experimentales

5.2.4. Acelerómetro

El propósito del acelerómetro en nuestro proyecto es el de devolver la inclinación sobre los tres ejes del espacio (X, Y y Z) con tal de poder llevar el control en todo momento del posible peligro por vuelque que podría sufrir el vehículo de apoyo.

Al tratar los datos utilizando el método expuesto en el apartado "5.2.3. Temperatura y Humedad" reconocimos un problema importante: el *delay*.

Al tratar los datos mediante cadenas de *strings* en el monitor serie se pierde la velocidad de transmisión, es decir, LabVIEW™ es el encargado de procesar información por caracteres, hecho el cual le hace perder velocidad de comunicación entre ambas plataformas y, a medida que aumenta el tiempo de ensayo, también lo hace el *delay*. Solucionar este problema complica en cierto modo la programación tanto en LabVIEW™ como en ARDUINO®, pero a cambio nos permite un control completo sobre la entrada y salida de datos a nivel digital. Por lo tanto, la información dejará de transmitirse mediante *strings* sino que lo hará a través de bits.

Para llevar a cabo este método de comunicación es necesario tener en cuenta que en las placas basadas en el µcontrolador ATmega2560 un numero entero (*int*) almacena un valor de 16 bits (2 bytes). Esto produce un rango de -32768 a 32767. Por lo tanto, serán necesarios dos bytes para transmitir cada variable leída. Con esta premisa, el código de ARDUINO® para esta aplicación es el siguiente:

```
/**********************************/
/*          ACELERÓMETRO          */
/**********************************/


//Pines donde irán conectadas las señales del sensor//
constintxPin=A0;
constintyPin=A1;
constintzPin=A2;

//Variables para mantener los valores obtenidos//
intacelx;
intacely;
intacelz;

//Buffer para almacenar los bytes leidos//
bytebuff[6];

voidsetup(){
  Serial.begin(9600);
}

voidloop(){

  //Leer los datos de los pines
  intxRead=analogRead(xPin);
  intyRead=analogRead(yPin);
  intzRead=analogRead(zPin);

  //Convertir a grados el resultado
  acelx=2.5714*xRead-925.71;
  acely=2.5714*yRead-925.71;
  acelz=2.5714*zRead-925.71;

  //Función para formar la trama con todos los valores obtenidos//
  formar_trama(acelx,acely,acelz);

  //Enviar los valores obtenidos//
  Serial.write(buff,6);

}

//Función para formar la trama con todos los valores obtenidos//
voidformar_trama(intacelx,intacely,intacelz){
  buff[0]=(acelx>>8)&255;
  buff[1]=  acelx&255;
  buff[2]=(acely>>8)&255;
  buff[3]=  acely&255;
  buff[4]=(acelz>>8)&255;
  buff[5]=  acelz&255;
}
```

Como se puede apreciar, para poder solventar la imposición de utilizar dos bytes por cada variable leída, hemos creado una función que almacena en un *buffer* los datos en tamaños de dos bytes. Este buffer será enviado por el puerto serie a LabVIEW™ en

código ASCII[17] el cual descifrará y manipulará como un entero. Mediante el diagrama de boques de LabVIEW™ se puede ver como se realiza este proceso además del de transformar los valores obtenidos en rotaciones de una figura la cual simula ser el vehículo de apoyo.

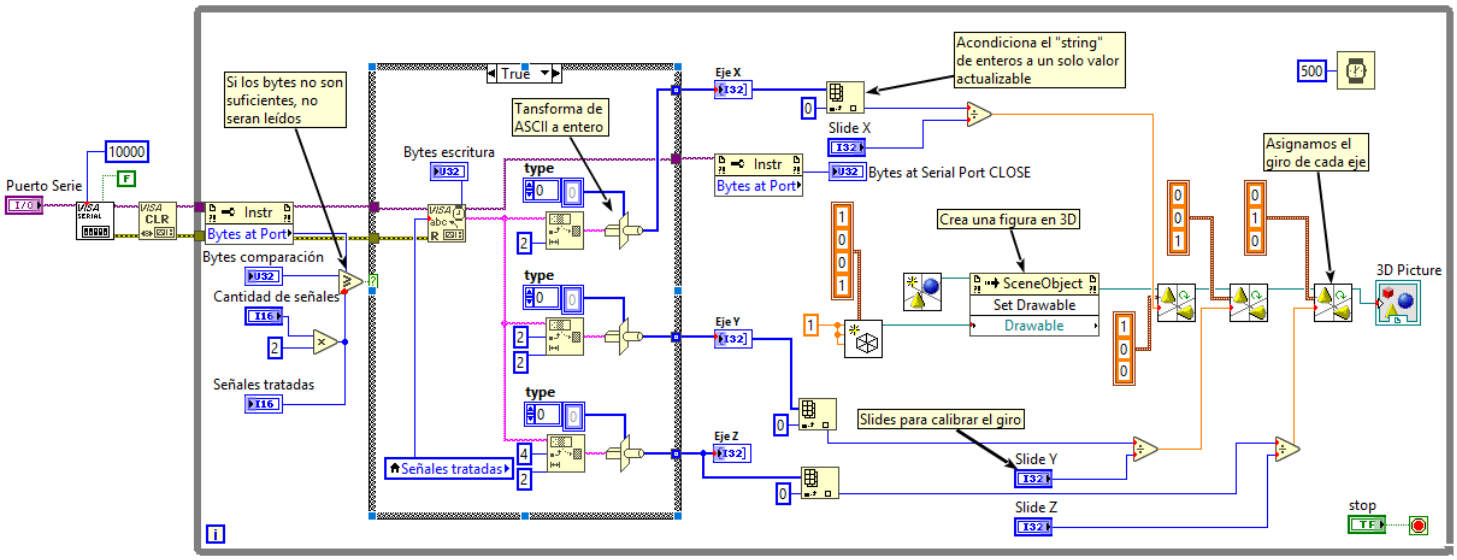


Figura 65.- Diagrama de bloques para el ensayo de obtención de la inclinación del vehículo

Para finalizar el ensayo, en la interfaz de control se puede apreciar la cantidad de bytes con las que se está trabajando, y el condicionante para la lectura.

$$Bytes\ comparación - Bytes\ escritura = Bytes\ at\ Serial\ Port\ CLOSE \qquad (5)$$

Con tal de minimizar las variaciones bruscas de movimiento de la figura 3D, se ha incluido en el programa unos *slides* los cuales sirven para la calibración manual.

[17]El código ASCII es un código de caracteres basado en el alfabeto latino que, como otros códigos de formato de representación de caracteres, es un método para una correspondencia entre cadenas de bits y una serie de símbolos (alfanuméricos y otros), permitiendo de esta forma la comunicación entre dispositivos digitales, así como su procesado y almacenamiento.

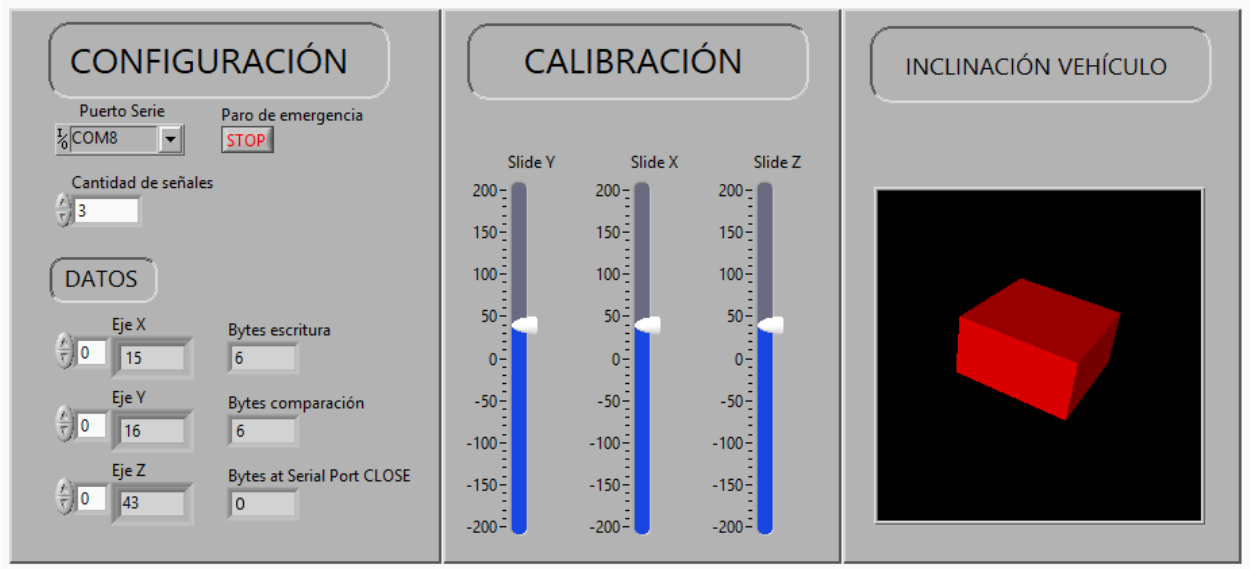


Figura 66.- Interfaz para el ensayo de obtención de la inclinación del vehículo y resultados experimentales

Pero la calibración no se hace únicamente en LabVIEW™, de hecho, en este punto solamente se establecen pequeñas variaciones para que el resultado final sea óptimo. Al tratarse de un acelerómetro raramente utilizado para la navegación o el control de la estabilidad de un vehículo (se trata de una familia de acelerómetros generalmente de tan solo dos ejes), tras varios ensayos llegamos a la siguiente relación entre los datos obtenidos y el ángulo real:

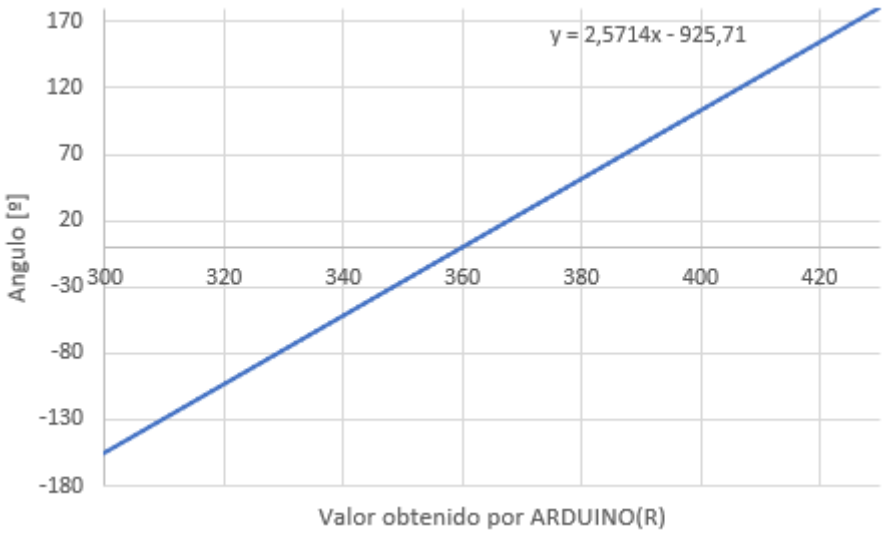


Figura 67.- Relación entre el valor obtenido por ARDUINO® y el valor real necesario a mostrar

De la cual obtenemos la fórmula vista en la gráfica y utilizada en el código (ver línea 31, 32 y 33 del código en este mismo punto) para la calibración del acelerómetro.

Pese a la dificultad de la programación tanto en LabVIEW™ como en ARDUINO®, vista la versatilidad de este método de comunicación, ha sido el escogido para llevar a cabo los ensayos posteriores e incluso para desarrollar la aplicación final.

5.2.5. Brújula

El método de comunicación de la brújula entre ARDUINO® y LabVIEW™ en este caso es completamente igual al anterior. Mediante un *buffer* enviaremos dos bites por cada señal leída (en este caso tan solo una, el ángulo). La programación en ARDUINO® se corresponde con el siguiente código:

```
/**********************************/
/*            BRÚJULA             */
/**********************************/

//Librerías del sensor//
#include <Wire.h>
#include <DFRobot_QMC5883.h>

//Instrucciones propias del sensor//
DFRobot_QMC5883 compass;

//Declinacion[18] en grados en Barcelona//
Const float declinacionAngle= 1.3;

//Buffer para almacenar los bytes leidos//
Byte buff[6];

voidsetup()
{
  compass.setRange(QMC5883_RANGE_2GA);
  compass.setMeasurementMode(QMC5883_CONTINOUS);
  compass.setDataRate(QMC5883_DATARATE_50HZ);
  compass.setSamples(QMC5883_SAMPLES_8);
}

voidloop() {
  Vector norm = compass.readNormalize();

  // Calcular ángulo (en radianes)
  float heading = atan2(norm.YAxis, norm.XAxis);
```

[18]La declinación magnética en un punto de la Tierra es el ángulo comprendido entre el norte magnético local y el norte verdadero (o norte geográfico). En otras palabras, es la diferencia entre el norte geográfico y el indicado por una brújula (el denominado también norte magnético). Para esta aplicación se ha utilizado la declinación propia de la zona del Besós en la cual está situada la universidad (1,1 º).

```
  // Corrección
  if (heading < 0){
    heading += 2 * PI;
  }

  if (heading > 2 * PI){
    heading -= 2 * PI;
  }

  int headingDegrees = heading * 180/M_PI + 100 + declinationAngle;

  //Función para formar la trama con todos los valores obtenidos//
  formar_trama(headingDegrees);

  //Enviar los valores obtenidos//
  Serial.write(buff,2);
}

//Función para formar la trama con todos los valores obtenidos//
voidformar_trama(int headingDegrees) {
  buff[0] = (headingDegrees >> 8) & 255;
  buff[1] = headingDegrees & 255;
}
```

En este caso el diagrama de bloques no difiere con el expuesto anteriormente, tan solo que, la salida de los datos enteros se ha representado en un manómetro. Para el correcto muestreo del ángulo de la brújula se han realizado varios experimentos con tal de llegar a la conclusión de la necesidad de un calibrado en el código. Esto responde a la línea 41 del código anteriormente mostrado:

$$headingDegrees = heading \cdot \frac{180}{M_PI} + \mathbf{100} + declinationAngle \qquad (6)$$

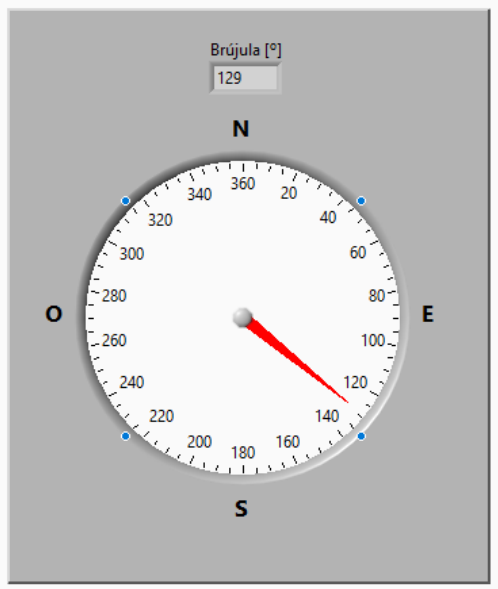


Figura 68.- Interfaz para el ensayo de obtención del ángulo de la brújula y resultados experimentales

Este manómetro ha sido modificado para devolver valores situados entre 0 y 360, los cuales se corresponden a los valores de ángulo recibidos por el puerto serie de ARDUINO®. Finalmente se ha rectificado el muestreo de datos con tal de situar el valor 360 superpuesto con el valor 0, lo cual crea una falsa ilusión de la continuación del giro de la brújula en este impasse.

5.2.6. Sensor de Corriente

El sensor de corrientes mantiene el método de funcionamiento de los anteriores dispositivos. En este caso el código de la aplicación es el siguiente:

```
/**********************************/
/*          CORRIENTE           */
/**********************************/

//Sensibilidad en Voltios/Amperio para sensor de 20A//

float Sensibilidad=0.066;

voidsetup() {

  Serial.begin(9600);
}

voidloop() {
  //Lectura del sensor
  floatvoltajeSensor=analogRead(A0)*(5.0 / 1023.0);

  //Ecuación  para obtener la corriente
  float Corriente=(voltajeSensor-2.5)/Sensibilidad;

  //Función para formar la trama con todos los valores obtenidos//
  formar_trama(Corriente);

  //Enviar los valores obtenidos//
  Serial.write(buff,2);
}

//Función para formar la trama con todos los valores obtenidos//
voidformar_trama(int Corriente) {
  buff[0] = (Corriente >> 8) & 255;
  buff[1] = Corriente & 255;
}
```

Este componente ha sido utilizado para realizar el cálculo de la batería en todo momento en función de la corriente consumida por el sistema.

5.2.7. GPS

Siguiendo el mismo método de funcionamiento que en los apartados anteriores, el código en ARDUINO® para la recogida de la latitud y la longitud del vehículo es el siguiente:

```
/**********************************/
/*               GPS              */
/**********************************/

//Librerías del sensor//
#include<TinyGPS.h>

Float lat,lon;
TinyGPSgps;

voidsetup(){
Serial.begin(9600);
Serial1.begin(9600);
}

voidloop(){

  if(gps.encode(Serial1.read())){
    gps.f_get_position(&lat,&lon);
    }

  int latpart1 = lat;
  int latpart2 = (lat - latpart1)*100;
  int latpart3 = (lat*100 - latpart1*100 - latpart2)*180;

  int lonpart1 = lon;
  int lonpart2 = (lon - lonpart1)*100;
  int lonpart3 = (lon*100 - lonpart1*100 - lonpart2)*180;

  //Función para formar la trama con todos los valores obtenidos//
  formar_trama(latpart1, latpart2, latpart3, lonpart1, lonpart2,
lonpart3);

  //Enviar los valores obtenidos//
  Serial.write(buff,12);
}

//Función para formar la trama con todos los valores obtenidos//
voidformar_trama(floatlat,floatlon) {
  buff[0] = (latpart1 >> 8) & 255;
  buff[1] =  latpart1 & 255;
  buff[2] = (latpart2 >> 8) & 255;
  buff[3] =  latpart2 & 255;
  buff[4] = (latpart3 >> 8) & 255;
  buff[5] =  latpart3 & 255;
  buff[6] = (lonpart1 >> 8) & 255;
  buff[7] =  lonpart1 & 255;
  buff[8] = (lonpart2 >> 8) & 255;
  buff[9] =  lonpart2 & 255;
  buff[10] = (lonpart3 >> 8) & 255;
  buff[11] =  lonpart3 & 255;}
```

Con tal de poder mostrar un GPS actualizable en la interfaz principal de LabVIEW™ se debe crear una URL que se auto-actualice en función de las coordenadas o de la información que queremos registrar [13]. Los datos de latitud y longitud se envían en bloques de tres enteros; unidades, centésimas y diezmilésimas. El motivo de esta división se debe al hándicap de utilizar como variables valores enteros con un rango de 0 a 255 bits (este punto se encuentra explicado en el apartado "5.3. Comunicación LabVIEW™ – ARDUINO®").

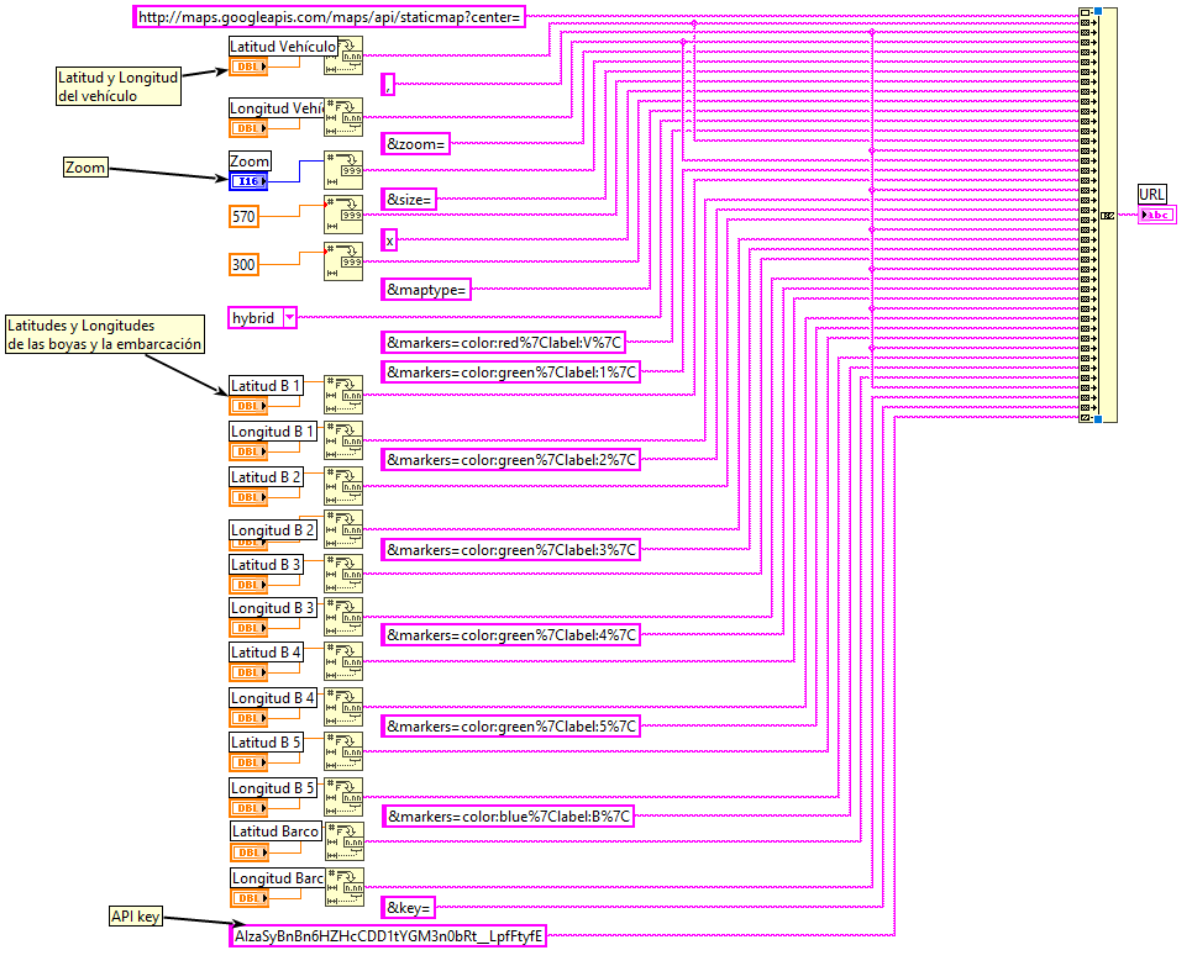


Figura 69.- Diagrama de bloques para la implementación de Google Maps©

El GPS utilizado es el ofrecido por Google Maps©. Entendiendo la disposición de los elementos dentro de la URL actualizable se puede ser capaz de incluir puntos marcados dentro del mapa o incrementar y disminuir el zoom. Para poder utilizar este mapa se ha debido de reclamar un código API[19] para el cual era necesario el depósito

[19]Una API es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. Las siglas API vienen del inglés *Application Programming Interface* (Interfaz de Programación de Aplicaciones).

simbólico de 1€ hacia Google©. El resultado en la interfaz de LabVIEW™ es el siguiente:

Figura 70.- Implementación de Google Maps© en la interfaz de LabVIEW™

Como se puede apreciar en la imagen anterior, el dentro del mapa siempre corresponderá al Vehículo de apoyo (V), el cual está marcado en un claro color rojo. La Embarcación principal (B) desde la cual se llevará el control de la aplicación aparece marcado en azul y el conjunto de Nasas reciben en verde un número del 1 al 5 (configurable desde la embarcación) que corresponderán al orden en el cual el vehículo realice la búsqueda de las mismas.

5.3. Comunicación LabVIEW™ - ARDUINO®

Hasta ahora tan solo se ha repasado la comunicación unidireccional de ARDUINO® a LABVIEW™ debido a que la sensórica no requiere de interacción con el usuario que dirija la aplicación más que tomar decisiones en función de los valores obtenidos.

Pero este no es siempre el caso, de hecho, hay elementos que el propio usuario debe introducir en LabVIEW™ y ARDUINO® debe recoger, interpretar y tomar decisiones al respecto.

Algunos de estos elementos pueden ser el encendido o apagado de la baliza o la bocina, seleccionar el modo manual o el modo automático, introducir las latitudes de las nasas para que el vehículo las busque, etc.

ARDUINO® tan solo es capaz de comprender *strings* en el momento de la recepción de la información, por lo tanto, todos los datos que se quieran transmitir deben ser transformados a *string* en primer lugar. Una vez ya han sido transformados, estos *strings* se pueden concatenar para enviarlos de una sola vez.

El orden en el cual se envían los *strings* debe de ser el mismo con el cual ARDUINO® los identifique, esta recepción se realiza mediante el comando *Serial.read()* de ARDUINO® pero contamos con el hándicap de tan solo poder obtener valores de 0 a 255, por lo tanto, si se quieren obtener valores más grandes, negativos o decimales se debe jugar con los datos tanto desde LabVIEW™ como desde ARDUINO®.

Las variables que deben ser enviadas desde LabVIEW™ a ARDUINO® son 18, las cuales se pueden dividir en booleanas y no booleanas. Dentro de las booleanas nos encontramos:

- Autonomía: Esta variable envía un 0 al estar desactivada y un 1 al estar activada. Al activarse inicia la búsqueda autónoma de las nasas por parte del vehículo.
- Manual: Por defecto esta activada enviando un 0 (se comporta de manera inversa a la variable “Autonomía”. Al estar activa el control manual se inicia.
- Baliza: Esta variable activa el relé correspondiente a la baliza, el cual permite el funcionamiento de la misma. Por defecto se encuentra encendida.
- Bocina: Activa el relé correspondiente a la bocina y mediante el código de ARDUINO® emite un bocinazo cada un cierto tiempo. Por defecto se encuentra apagada.
- Velocidad Motor Derecho: Esta variable devuelve dos componentes. Está diseñada para el control manual y, cuando es diferente de 0 envía el valor correspondiente a ARDUINO® el cual se encarga de hacer girar el motor a la intensidad correspondiente, por lo tanto, una de las booleanas corresponde al “encendido/apagado” del motor. La otra booleana entra en acción al recibir un comando negativo, es decir, cuando el usuario quiere hacer girar el motor de forma inversa. En este momento se envía la segunda booleana para hacer saber a ARDUINO® que el valor recibido debe ser un negativo (ya que como se ha comentado anteriormente, tan solo se podían enviar valores del 0 al 255).
- Velocidad Motor Izquierdo: El funcionamiento es completamente igual al de la variable Velocidad Motor Derecho.

Respecto a las variables no booleanas que nos podemos encontrar, estas son:

- Selección de Región: Las coordenadas se dividen en dos ejes, el de latitud y el de longitud, el valor 0 de los cuales son el Ecuador y el meridiano de *Greenwich* respectivamente. Estos ejes dividen las latitudes y longitudes en positivas y negativas, y, como es imposible enviar números negativos de LabVIEW™ a ARDUINO®, se ha tenido que añadir esta variable de configuración, la cual se encarga de hacerse saber a ARDUINO® el signo de los valores de latitud y longitud que el usuario está introduciendo manualmente en la interfaz la letra de la región en la cual se va a desarrollar la actividad de pesca.

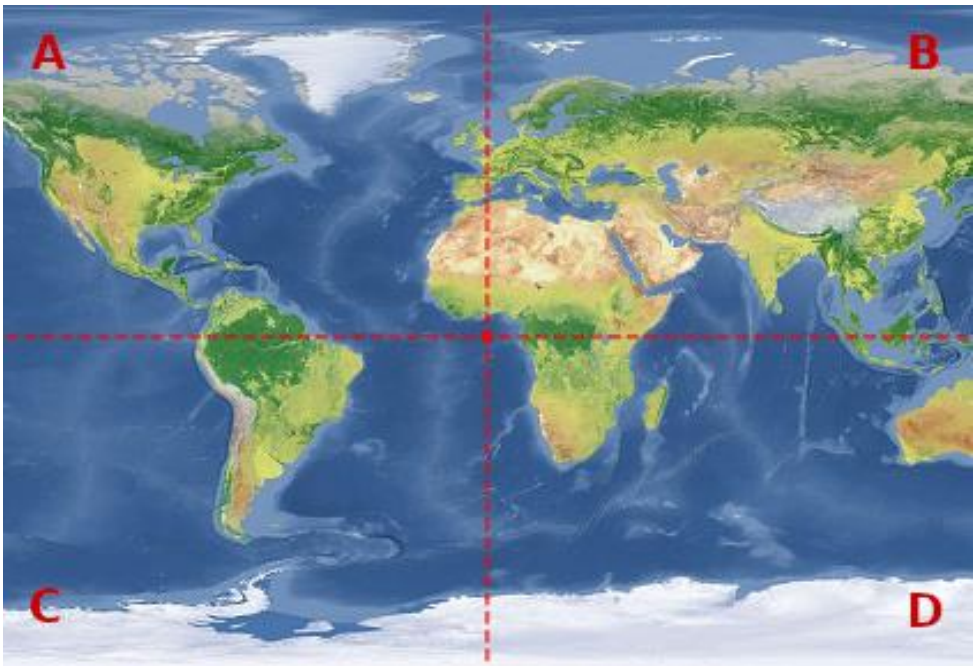


Figura 71.-División por regiones del planeta

- Latitudes y Longitudes: Al usuario le interesa que el vehículo de apoyo realice una ruta por todas las boyas para detectar la cantidad de cangrejos en sus nasas y, que en último lugar vuelva a la posición de la embarcación para que pueda ser recogido, pero las coordenadas se escriben generalmente en grados, lo cual crea la necesidad de un número con decimales los cuales, a mayor cantidad, mayor precisión permiten transmitir, es decir, de la coordenada de longitud 0º 00’ 00” a la coordenada 0º 01’ 00” hay más de 1km de distancia, por lo tanto, es necesario contar con la mayor cantidad de decimales posibles para asegurar la precisión de la aplicación.

  Con tal de comprobar la cantidad de decimales mínimos necesarios para un correcto funcionamiento realizamos un pequeño experimento; sabiendo la distancia entre dos puntos aleatorios del planeta, comprobar cuanta distancia corresponde a los decimales de las coordenadas.

Tomando dos coordenadas aleatorias como pueden ser: 41º 31’ 7010”N, 2º 02’ 6506”E y 41º 46’1158”N, 2º 23’ 9080”E podemos observar en la siguiente imagen obtenida mediante Google Maps© la distancia en kilómetros entre los dos puntos:

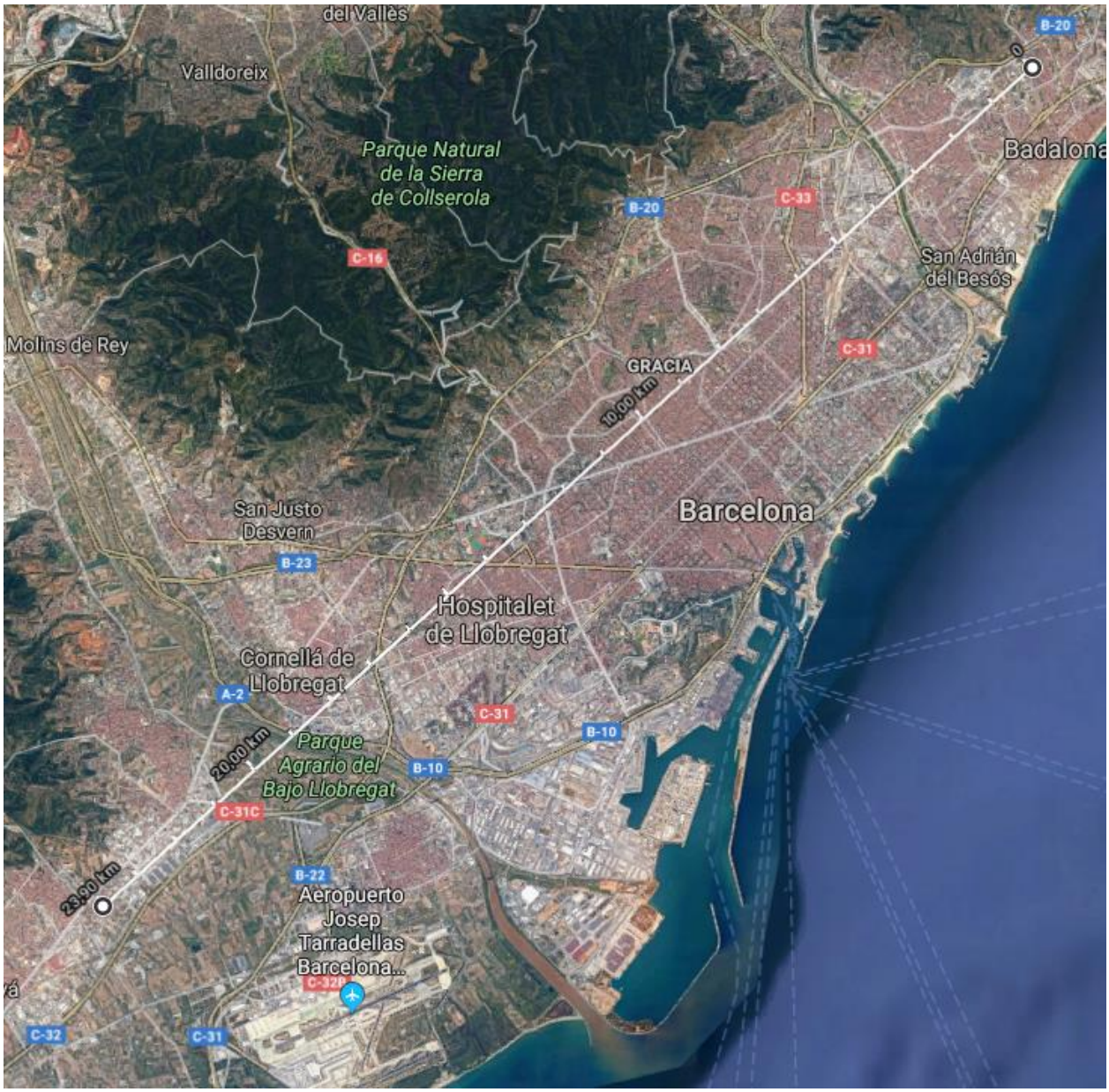


Figura 72.-Distancia entre dos puntos aleatorios cercanos a Barcelona

- La cual corresponde a 23,90km y, calculando la hipotenusa entre las dos parejas de coordenadas escogidas obtenemos una distancia en grados de:

$$a^2 = b^2 + c^2 \qquad (7)$$

$$a^2 = (|41.317010 - 41.461158|)^2 + (|2.026506 - 2.239080|)^2$$

$$a = \sqrt{0.027786459 + 0.4518770547}$$

$$a = 0.6925774424$$

Por lo tanto, conociendo la distancia en kilómetros (23.90km) y la distancia en grados (0º 69” 2577’) podemos saber los grados por kilómetro correspondientes a la zona de Barcelona los cuales son de 0.028978º/km, por lo tanto, realizando una sencilla regla de tres, podemos saber que con una precisión de 0º 00’ 01” obtendríamos una respuesta en precisión de 3,45m aproximadamente, distancia que debería ser más que suficiente para poder recibir correctamente la señal enviada por la boya.

Para poder enviar un valor de 4 decimales con variables limitadas entre el 0 y el 255 se ha optado por dividir la coordenada en tres partes los cuales corresponderían a las decenas y unidades, a las décimas y centésimas y a las milésimas y diezmilésimas, es decir:

- Decenas y unidades: **XX**, 00 00º
- Decimas y centésimas: 00, **XX** 00º
- Milésimas y diezmilésimas: 00, 00 **XX**º

LabVIEW™ es el encargado de dividir estos valores para poder enviarlos como *strings* y, por su parte, ARDUINO® se encarga de recogerlos, y unirlos de tal manera que vuelvan a formar el valor añadido por el usuario.

5.3.1. Control de Motores

Para realizar el control de los motores propulsores del vehículo se ha utilizado el Driver MDD10A de Cytron®. Se trata de un driver que controla la velocidad de hasta dos motores simultáneamente en función de un valor comprendido entre 0 y 255.

Tras numerosos ensayos se llegó a la conclusión que la recta de funcionamiento corresponde a la siguiente gráfica:

Figura 73.- Curva de funcionamiento del driver MDD10A

Cuando la entrada de datos se acerca al valor medio entre el valor máximo y mínimo que se pueden introducir (0 y 255), el motor tiende a frenar, mientras que en estos puntos se encuentra en su mayor velocidad de rotación.

El código mediante el cual se realiza la introducción de estos datos y el control de un solo motor es el siguiente:

```
/**********************************/
/*            MDD10A              */
/**********************************/

//Declaración de pines//
#define MotorDirection 7
#define MotorSpeed 6

void setup() {
  pinMode(MotorDirection, OUTPUT);
  pinMode(MotorSpeed, OUTPUT);
}

void loop() {
  //Activación del Motor
  digitalWrite(MotorDirection, HIGH);
  //Declaración de la velocidad del motor (en este caso 255)
  analogWrite(MotorSpeed,255);
}
```

La declaración de la velocidad del motor se puede introducir como un número negativo o uno positivo, alternando en función de este carácter el sentido de giro del mismo.

5.3.2. Comunicación Inalámbrica: XBee™

Puesto que la aplicación debe ser controlada desde la embarcación tripulada y el vehículo de apoyo debe encontrarse en alta mar tomando los registros pertinentes, es necesario implementar un método de comunicación inalámbrica que no requiera de cableado.

La comunicación inalámbrica se ha llevado a cabo mediante XBee™[20], escogido por la simplicidad de su configuración y por su alcance de comunicación (de hasta 1500m) suficientemente elevado para desarrollar el prototipo diseñado.

Para poder realizar el montaje final, se llevaron a cabo algunas pruebas de comunicación entre dos XBee™ (ya que uno deberá situarse en la embarcación y otro en el vehículo de apoyo) mediante el software XCTU™ proporcionado por DIGI®, una

[20]Diseñados por DIGI®, los módulos XBee™ son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan el protocolo de red llamado IEEE 802.15.4 para crear redes *FAST POINT-TO-MULTIPOINT* (punto a multipunto); o para redes *PEER-TO-PEER* (punto a punto).

vez comprobado el correcto funcionamiento de la comunicación inalámbrica entre dispositivos, se comprobó el correcto procesamiento de la información por parte de ARDUINO®.

Para este ensayo se estableció el siguiente código, el cual se encarga de encender un diodo LED 1, 2 o 3 veces en función del número enviado a través de la consola de XCTU™ a ARDUINO®:

```
/**********************************/
/*              XBEE              */
/**********************************/

void setup() {
  Serial1.begin(9600);
  pinMode(9,OUTPUT);
}

void loop() {

  //Declaración de variables
  int dato;
  int t = 200;


  if (Serial1.available() > 0) {

    //Lectura del valor
    dato = Serial1.read();

    //Encender 1 vez
    if (dato == '1'){
      digitalWrite(9,HIGH);
      delay(t);
      digitalWrite(9,LOW);
      delay(t);
    }

    //Encender 2 veces
    else if (dato == '2'){
      digitalWrite(9,HIGH);
      delay(t);
      digitalWrite(9,LOW);
      delay(t);

      digitalWrite(9,HIGH);
      delay(t);
      digitalWrite(9,LOW);
      delay(t);
    }

    //Encender 3 veces
    else if (dato == '3'){
      digitalWrite(9,HIGH);
      delay(t);
      digitalWrite(9,LOW);
      delay(t);

      digitalWrite(9,HIGH);
      delay(t);
```

```
        digitalWrite(9,LOW);
        delay(t);

        digitalWrite(9,HIGH);
        delay(t);
        digitalWrite(9,LOW);
        delay(t);

      }
    }
}
```

Tras verificar el comportamiento de la comunicación y comprobar que esta se puede realizar en ambos sentidos modificando el código anteriormente anexado (tanto de la embarcación al vehículo como del vehículo a la embarcación). El siguiente paso es sustituir el software de XCTU™ por LabVIEW™ y comprobar que los datos enviados por el puerto serie están utilizando los XBee™ como método de comunicación con ARDUINO® y que los datos enviados desde ARDUINO® no estás sufriendo alteraciones.

En este punto se modificó una versión alfa de la aplicación realizada para comprobar el envío y recepción de datos en tiempo real. Utilizando el código diseñado para el funcionamiento mediante (alterado para utilizar los pines 0 y 1 como puerto serie) XCTU™ y la versión modificada de la aplicación en LabVIEW™, se desconectó la placa de ARDUINO® del PC para conectarlo directamente a la red eléctrica, obteniendo así un solo puerto ocupado por la electrónica de unos de los XBee™. Se comprobó la obtención de un funcionamiento similar debido a que mediante XCTU™ tan solo se enviaba un dato cuando el usuario deseara, mientras que en LabVIEW™ la aplicación recoge y envía constantemente información con ARDUINO®, por lo tanto, en vez de encender un LED 1, 2 o 3 veces, enciende el mismo LED en packs de 1, 2 o 3 veces ininterrumpidamente mientras no se le dé una orden distinta.

5.4. Control de Trayectoria

Mientras que algunos drones de control autónomo trazan una trayectoria basándose en la cercanía a las orillas [14] (sobre todo en el caso de lagos y embalses), otros suelen utilizar sensores de posicionamiento y radiocontrol mucho más avanzados a los que disponemos actualmente [15], en nuestro caso contamos con dos sensores mediante los cuales se realizará un algoritmo de control de trayectoria válido para trazar un camino a través de las boyas en un orden específico.

Para desarrollar el modo autónomo de seguimiento de balizas se ha utilizado la posición geográfica de las balizas y el vehículo de apoyo, así como la orientación ofrecida por el magnetómetro con el fin de realizar un algoritmo de detección y seguimiento de las mismas.

En primer lugar, el magnetómetro o brújula nos indica la orientación respecto al eje de las ordenadas a la cual está encarando el vehículo, por lo tanto, tenemos un vehículo con una dirección fija. Al ángulo que forma la dirección del vehículo con la vertical lo denominaremos beta (β):

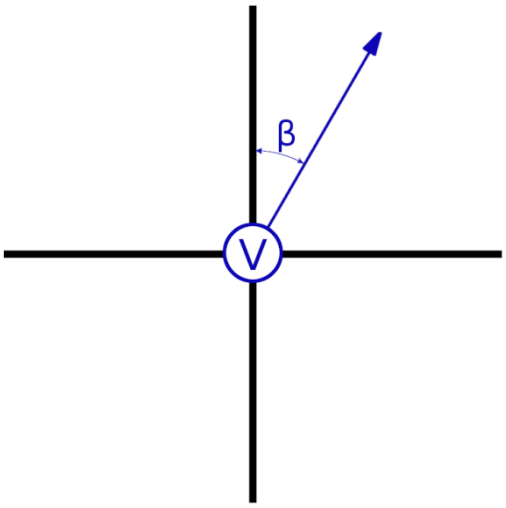


Figura 74.-Dirección y ángulo por defecto encarado por el vehículo

Supongamos que la nasa de la cual el vehículo quiere recoger información se encuentra en el segundo cuadrante:

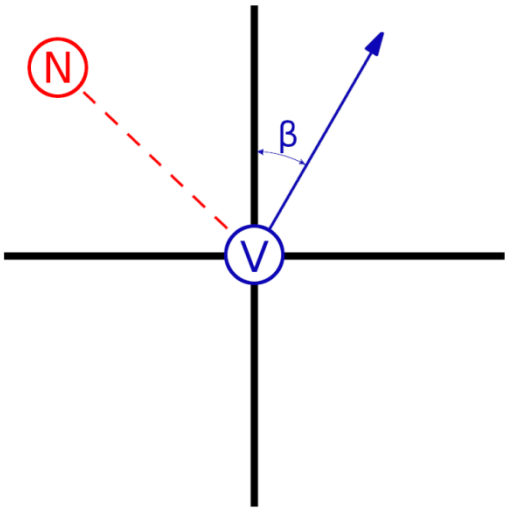


Figura 75.-Dirección la cual se quiere encarar

El GPS nos indica las coordenadas del vehículo, mientras que las coordenadas de la nasa son introducidas manualmente por el usuario de la aplicación, por lo tanto, podemos calcular el ángulo respecto a la horizontal al cual se encuentra la nasa. A este ángulo lo denominamos alfa (α):

$$\alpha = \arctan\left(\frac{LatitudBoya - LatitudVehículo}{LongitudBoya - LongitudVehículo}\right) \qquad (8)$$

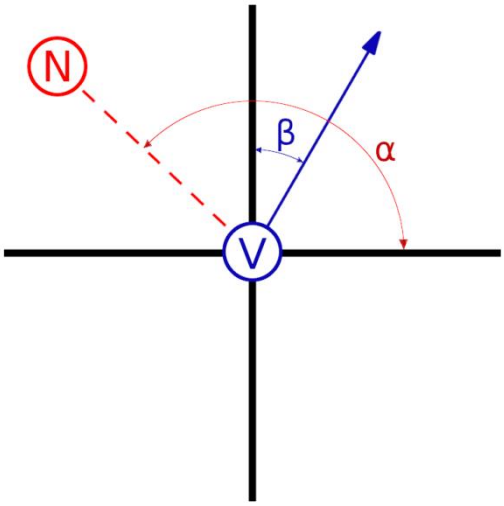


Figura 76.-Ángulo a encarar por el vehículo respecto a la horizontal

Finalmente, queremos obtener el ángulo entre la dirección que encara el vehículo (β) y el ángulo de la nueva dirección que queremos que encare, es decir, la dirección hacia la nasa (α).

El algoritmo establecido para cumplir con este requisito sea cual sea la orientación del vehículo y la posición de la nasa se muestra en la siguiente tabla:

Tabla 1.- Algoritmo para la obtención del ángulo de giro

90 º > α > 0º	$\beta' = 90 - \alpha$
180º > α > 90º	$\beta' = 360º - (\alpha - 90º)$
270º > α > 180º	$\beta' = 360º - (\alpha - 90º)$
360º > α > 270º	$\beta' = 360º - (\alpha - 90º)$

Con este algoritmo se puede establecer el ángulo de giro necesario (β') para encarar la nasa a seguir.

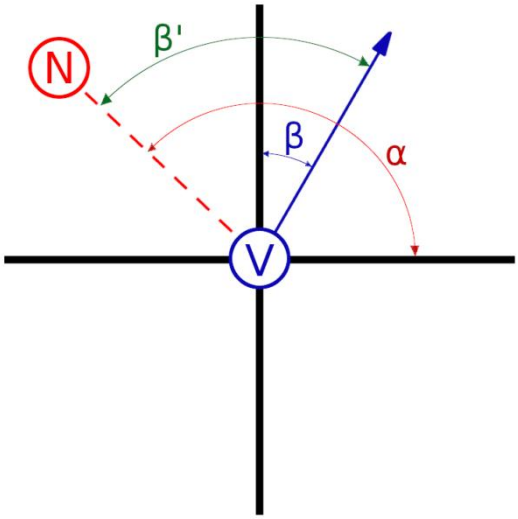


Figura 77.-Ángulo de giro obtenido

Una vez establecido este ángulo y con tal de optimizar recursos, se establece el giro en sentido horario o antihorario en función de la distancia más corta a recorrer. El algoritmo encargado de este punto es tan simple como la diferencia entre β' y β la cual, si es más grande de 180º, fuerza el giro en sentido horario, y si es menor, en sentido antihorario.

5.5. Aplicación Final

5.5.1. Control

LabVIEW™ permite la opción al usuario de crear y diseñar "cajas" con tal de poder utilizarlas allá donde sean necesarias. Esta característica recuerda a las funciones creadas en lenguajes de programación similares a C, las cuales se establecen fuera de los bucles de programa y le permiten al usuario hacer uso de ellas en cualquier momento y tantas veces como sea necesario.

Las "cajas" en LabVIEW™ se comportan como una solución extremadamente visual para realizar este tipo de funciones.

Con tal de simplificar el bucle de programa han sido realizadas un conjunto de "cajas" sin el conocimiento de las cuales el programa se vuelve algo menos intuitivo. Por lo tanto, y agrupándolas por tipos.

5.5.1.1. Cajas de escritura, tratamiento y lectura

Como ya se ha visto anteriormente, una vez establecido el método de comunicación de ARDUINO® a LabVIEW™, este se vuelve repetitivo y monótono, además de requerir un conjunto complejo de instrucciones para el correcto funcionamiento. Con tal de simplificar esta operación se han creado un conjunto de "cajas" auxiliares con diferentes salidas (1, 2 y 3) las cuales se encargan de tratar con la información recibida de una caja de escritura principal.

Figura 78.-Cajas de escritura y tratamiento

La caja "ESCRIBIR" se corresponde a la caja principal. La cual se encarga cribar la cantidad de datos que se van a tratar y las características de los mismos, además de ser la encargada de procesar los valores de temperatura y humedad. El diagrama de bloques y la información sobre las entradas y salidas de esta caja corresponden a la siguiente figura:

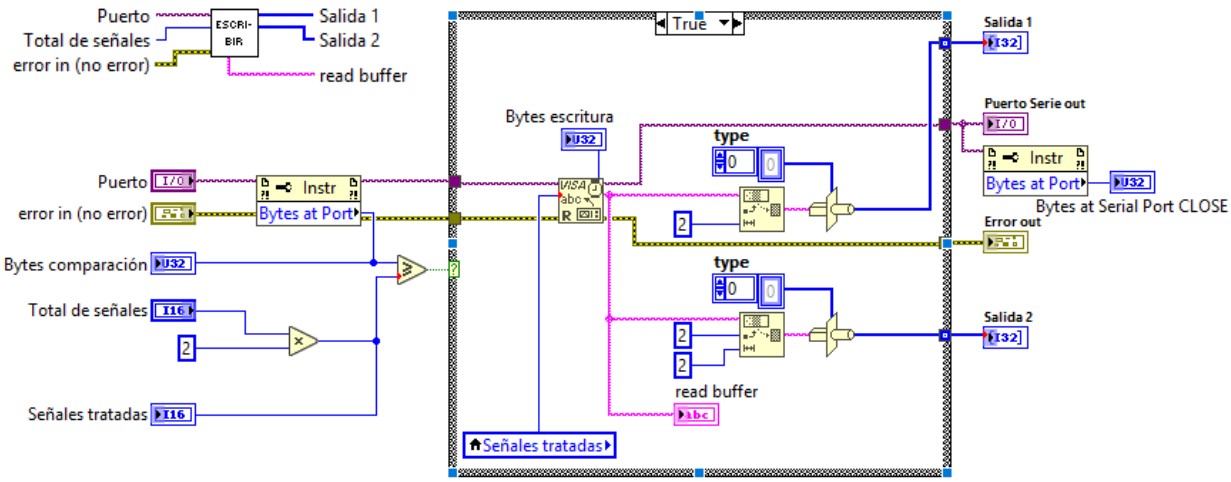


Figura 79.-Caja de escritura principal junto a su diagrama de bloques

Las ajas auxiliares de escritura "1 ESCRITURA", "2 ESCRITURAS" y "3 ESCRITURAS" son similares entre sí, y se han utilizado para elementos los cuales requieran de una, dos o tres salidas respectivamente. Cada una de estas cajas tiene de una a tres entradas de offset para indicar la posición de la trama sobre la cual se encuentran los datos que se quieren tratar. Cada una de estas cajas se conectan entre ellas hasta derivar en la caja de escritura principal.

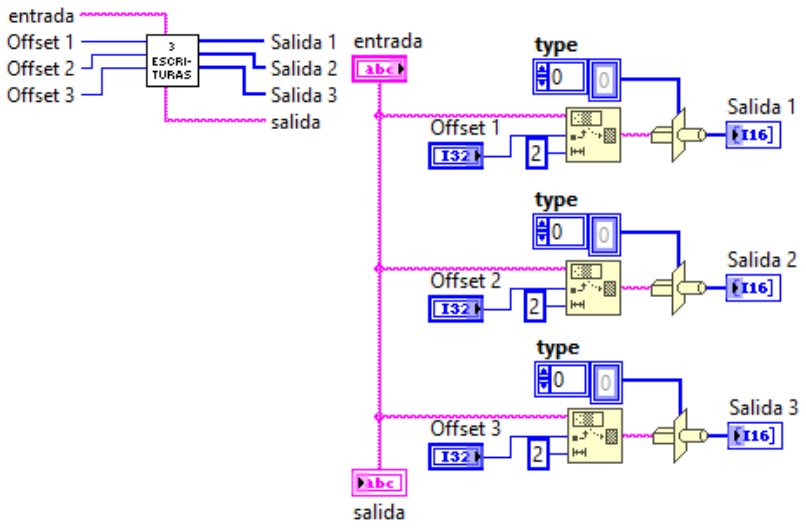


Figura 80.-Caja auxiliar de 3 escrituras junto a su diagrama de bloques

Todas y cada una las cajas de escritura están preparadas para devolver un *array* de datos. Existiendo la posibilidad de no requerir del *array* entero sino tan solo del último valor obtenido mediante el sensado, se ha diseñado la caja de "TRATAMIENTO", la cual transforma el *array* en un solo elemento actualizable con el cual trabajaremos. Esta caja ha sido utilizada sobre todo para ahorrar espacio en el diagrama de bloques final más que tener un uso complejo como es el caso de las anteriores.

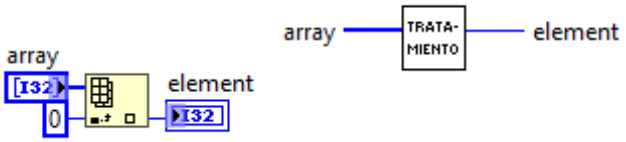


Figura 81.-Caja auxiliar de tratamiento junto a su diagrama de bloques

Respecto a la caja de escritura, principalmente es una, el problema llega cuando una gran cantidad de variables se deben transformar a *string* para poder ser concatenadas y enviarse por el puerto serie a ARDUINO®.

Figura 82.-Cajas de lectura y transformación

La caja "LEER" se corresponde a la caja principal la cual se encarga de transferir los datos obtenidos en LabVIEW™ por el puerto serie para que ARDUINO® los reciba. El diagrama de bloques de este elemento se corresponde al siguiente:

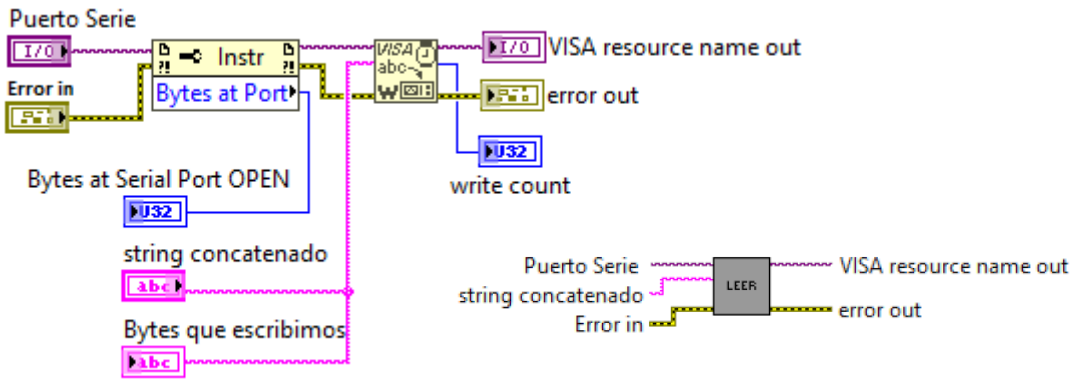


Figura 83.-Caja de lectura principal junto a su diagrama de bloques

De esta caja cabe destacar la entrada de *string* concatenado, ya que los elementos que se deberán enviar deben de ser en código ASCII, y LabVIEW™ es capaz de transmitir este código transformando los valores que se desean transferir a hexadecimal y de hexadecimal a ASCII.

Esta es la función de las cajas de transformación cuya función se basa siempre en el mismo algoritmo que se puede apreciar en la siguiente figura, correspondiente a la caja de transformación para los relés:

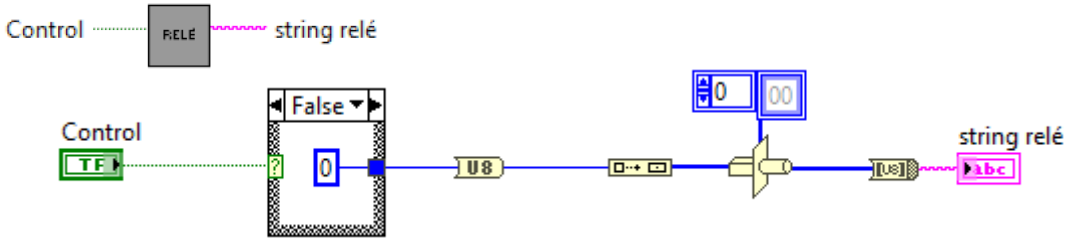


Figura 84.-Caja de transformación de relé junto a su diagrama de bloques

Mediante esta misma función se rigen las cajas de transformación de "REGIÓN", "AUTO/MANUAL" y parte de "MOTOR".

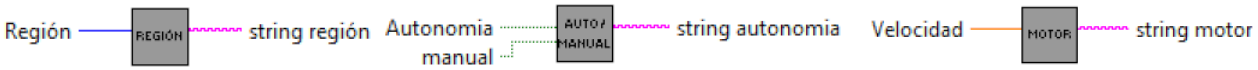


Figura 85.-Cajas de transformación de "REGIÓN", "AUTO/MANUAL" y "MOTOR"

La caja que más difiere del resto es la de "STRING DEC" la cual se encarga de (como se ha explicado anteriormente) dividir los valores de coordenadas en tres parejas con tal de que puedan ser enviados. Estos tres valores, tras haber sido divididos, se concatenan y forman un *string* el tripe de largo que el resto, teniendo que tener en cuenta así que ARDUINO® recibirá tres valores en vez de uno cada vez que reciba uno de estos *strings*.

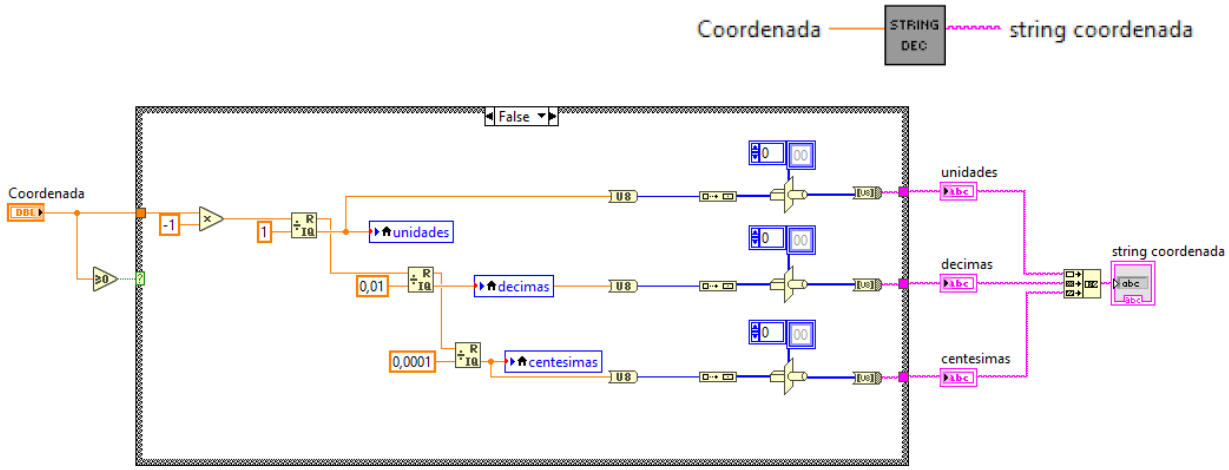


Figura 86.-Caja de transformación de coordenadas junto a su diagrama de bloques

5.5.1.2. Adaptación y configuración

Tras obtener ciertas señales, estas deben ser tratadas con tal de crear restricciones, requisitos o condiciones. Para estos casos se han diseñado cajas de configuración. Por otro lado, si las señales no son del todo comprensibles para el usuario en primera instancia, han sido diseñadas cajas de adaptación las cuales se encargan de solventar esta problemática.

En primer lugar, podemos analizar la señal de temperatura. Como se ha visto en apartados anteriores, la temperatura en el mar de Bering es extrema pero también notablemente constante, es por esto que se ha diseñado una configuración la cual permite al usuario establecer valores de temperatura máxima y mínima mediante los cuales pueda saber si el vehículo se encuentra en condiciones desfavorables. Esta caja se encarga de encender un LED de emergencia en pantalla si se ha sobrepasado cualquiera de las temperaturas límite establecidas por el usuario, además de mantener otro LED encendido en pantalla en el caso de que el vehículo se encuentre en condiciones estables.

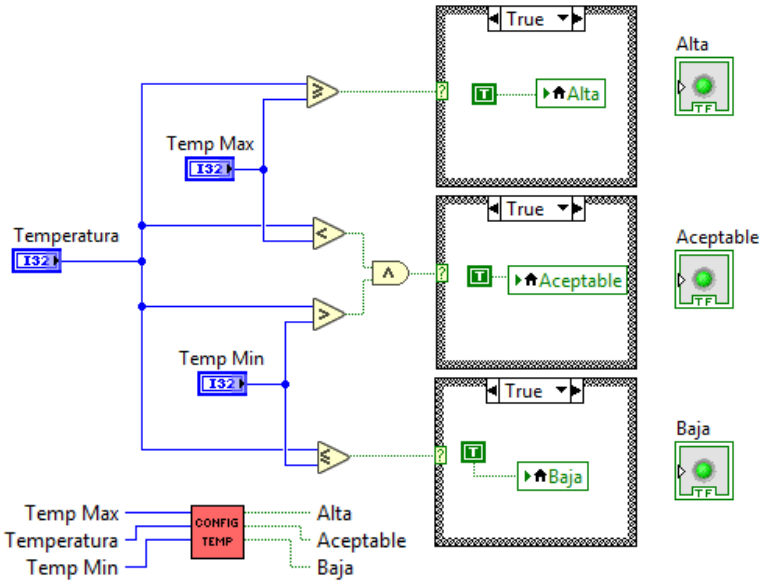


Figura 87.-Caja para la configuración de temperatura junto a su diagrama de bloques

Otra caja creada, en este caso como adaptación de señal es la encargada de transformar la señal obtenida del acelerómetro para transformarla en ángulos de rotación de los tres ejes del espació del vehículo de apoyo vistos sobre una figura en tres dimensiones.

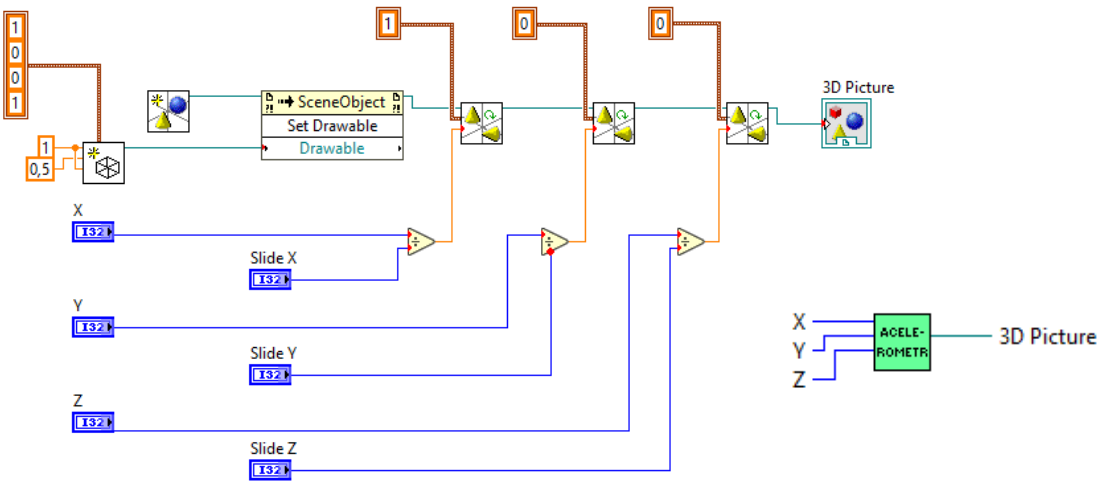


Figura 88.-Caja para la adaptación del acelerómetro junto a su diagrama de bloques

Como último ejemplo de caja de configuración podemos ver la caja para el conteo de la batería restante. Se trata de una caja simple encargada de (siempre y cuando se inicie la aplicación con la batería completamente cargada), mostrar en tiempo real el tanto por ciento de batería restante, en función de la corriente que puede llegar entregar nuestra batería.

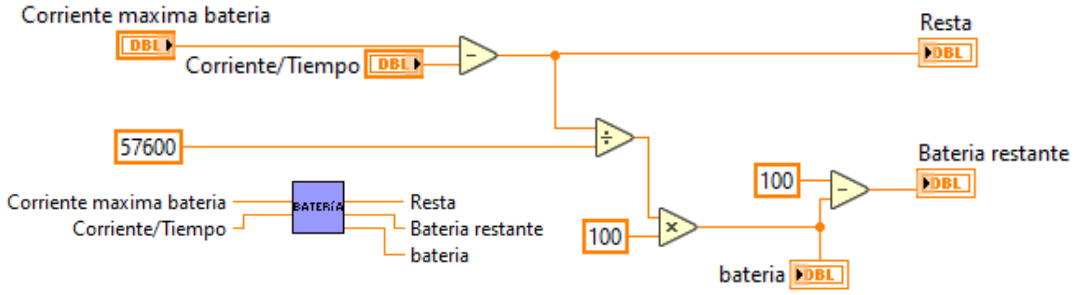


Figura 89.-Caja para la configuración de la batería junto a su diagrama de bloques

La última caja encargada de adaptar las variables obtenidas mediante ARDUINO® es la caja de GPS, que como se ha visto anteriormente se encarga de, mediante los valores recibidos por parte del usuario de la aplicación en LabVIEW™ modificar constantemente la URL con tal de hacer visibles los nuevos datos en el mapa de la aplicación y hacer visual el recorrido y la situación geográfica de todos los elementos a tener en cuenta. Visto en la figura 69 el diagrama de bloques principal de esta aplicación, la caja de aplicación es la siguiente:

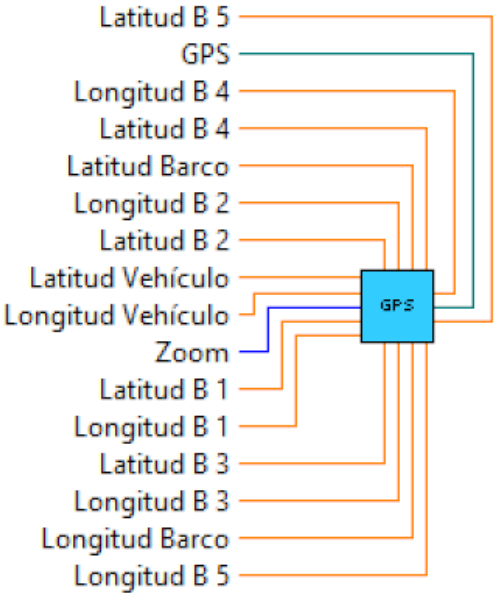


Figura 90.-Caja para de adaptación y configuración del GPS

Existen otros elementos decorativos a los y que hacen más intuitiva la acción de programar el vehículo, pero de todos ellos tan solo nos centraremos en el encargado de informar visualmente (mediante el encendido constante de LEDs) el momento de la recogida de las nasas.

El usuario se encarga de indicar cuál es la cantidad mínima de cangrejos con la cual le resultaría rentable recoger la nasa, con este valor introducido en la interfaz de LabVIEW™, el mismo programa es quien recibe la información por parte de ARDUINO® cuando se encuentre a una distancia de menos de 0º 00' 01" e informa al usuario del momento en el cual sería conveniente recoger las nasas. El diagrama de bloques de esta aplicación corresponde al siguiente:

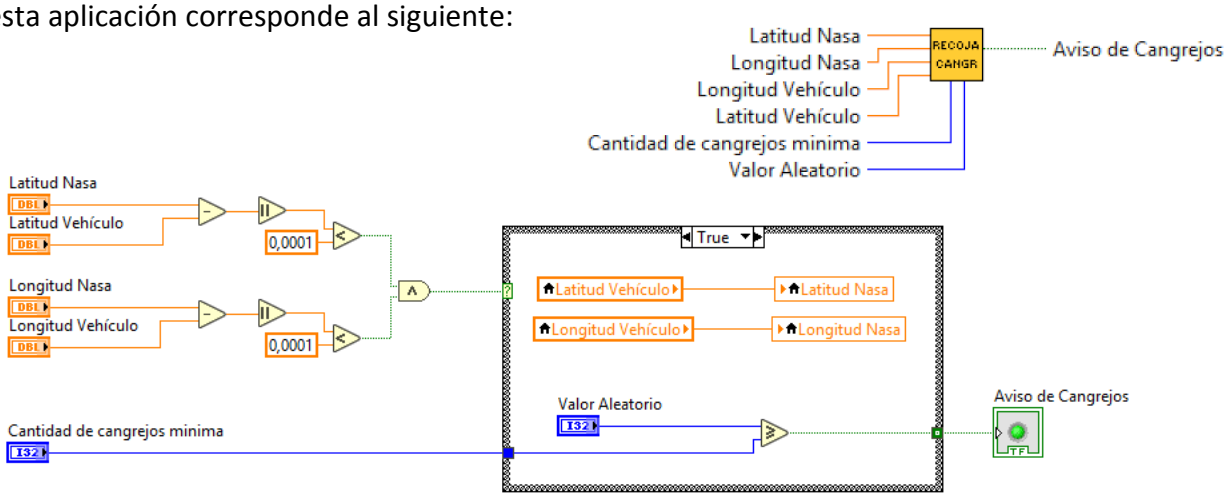


Figura 91.-Caja para la contabilización de cangrejos junto a su diagrama de bloques

En este caso, al no disponer del sensado real de la cantidad de cangrejos, ARDUINO® genera un número aleatorio y es introducido en LabVIEW™ como si se tratara de una variable más.

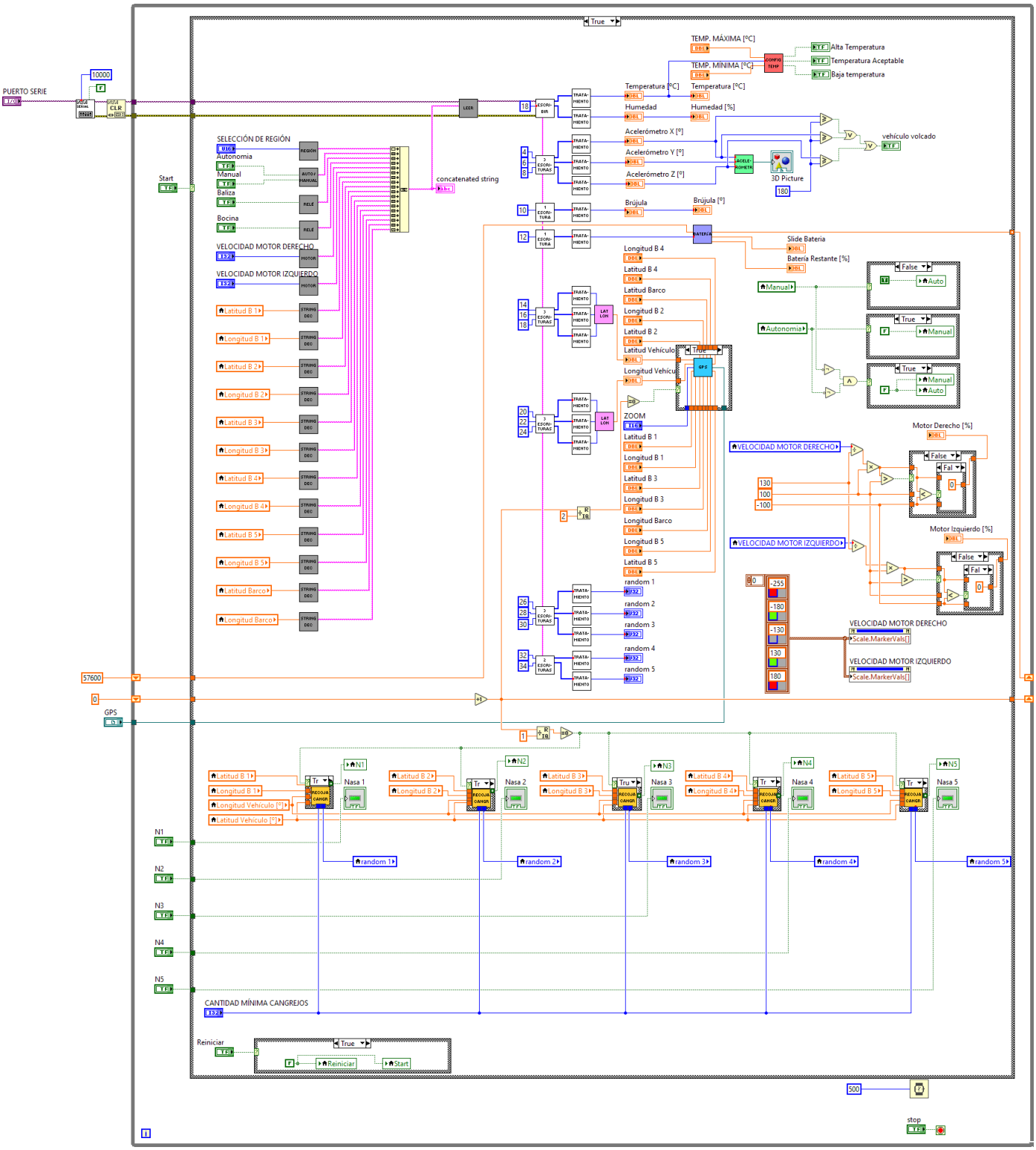


Figura 92.-Diagrama de bloques de la aplicación de control

5.5.2. Registro

Anteriormente no se ha comentado en ningún momento como se realizaría el registro de los datos obtenidos mediante la aplicación y, esto se debe a que el registro de todos los datos se realizará desde una aplicación independiente a la de control que, pese a compartir el esqueleto en el diagrama de bloques, se convierte en una aplicación más fácil de programar que no requiere de las cajas de adaptación, configuración ni lectura, tan solo se centra en la muestra constante mediante gráficas de todos los elementos sensados.

Al margen de mostrar las gráficas, utilizando la herramienta de LabVIEW™ *WriteToMeasurement* se ha podido desarrollar la capacidad de, pulsando un el botón indicado para cada gráfica se pueden registrar todos los datos en un libro de Microsoft Excel® en una carpeta en el escritorio del PC utilizado para la aplicación. Además, existe la opción de registrar todos los sensados que se quieran al unísono, incluso se ha añadido la capacidad de pulsar un botón para el registro simultaneo de todos los sensados.

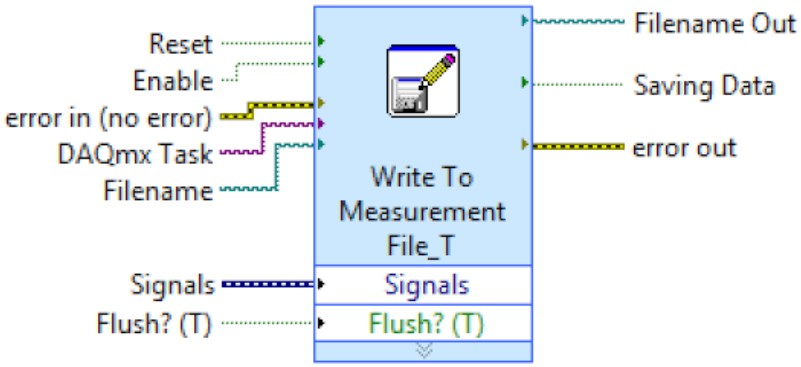


Figura 93.-Caja de la herramienta *WriteToMeasurement*

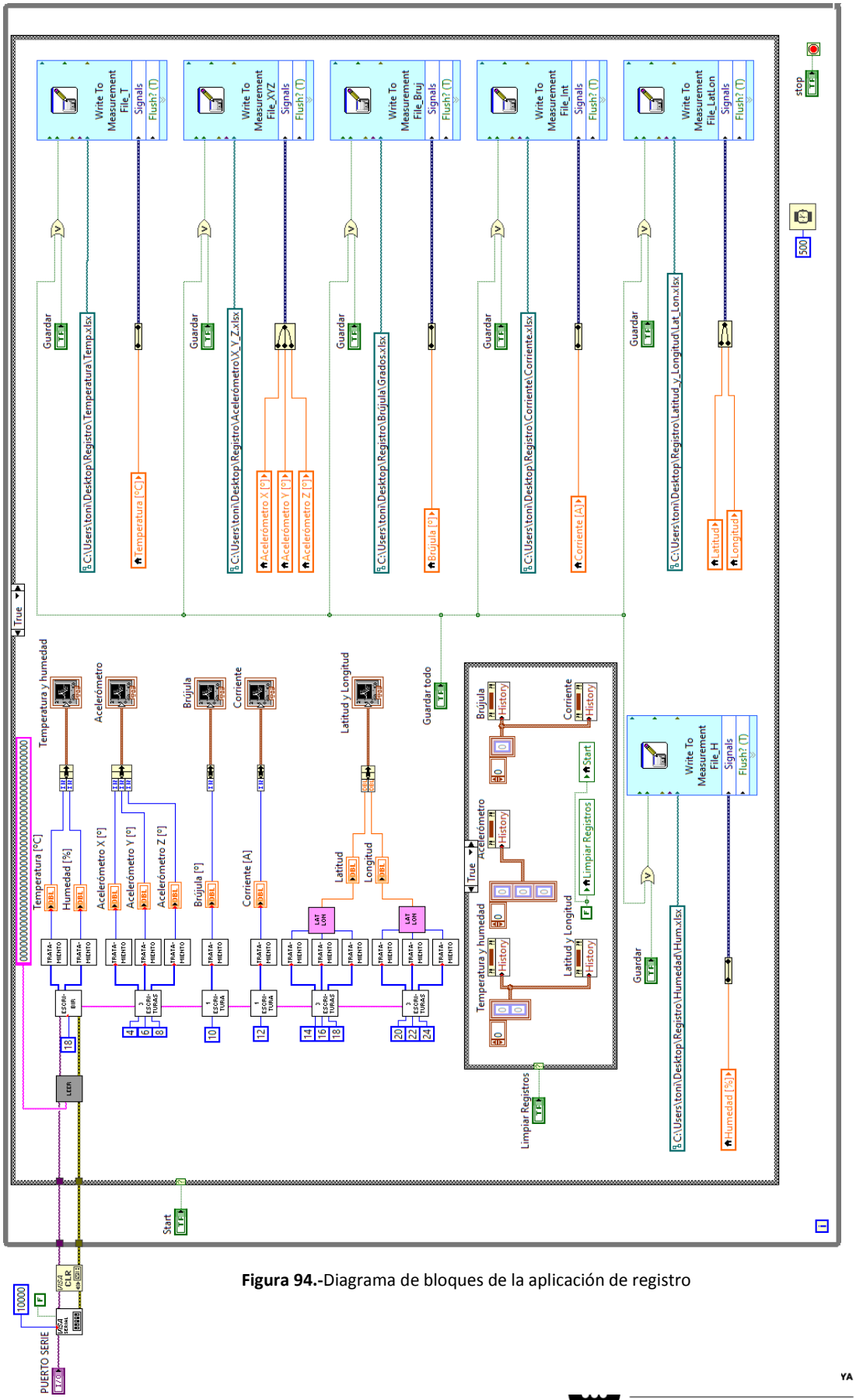


Figura 94.-Diagrama de bloques de la aplicación de registro

5.5.3. *Interfaz*

Se ha hecho hincapié a lo largo del proyecto en la necesidad de realizar una interfaz sencilla e intuitiva para el usuario que deba controlar o registrar datos del vehículo de apoyo, es por eso que se ha tratado de hacer muy visual este elemento con el fin de que cualquier individuo con un conocimiento mayor o menor de radiocontrol o de registro de información pueda, sin ningún impedimento, realizar las operaciones que desee[21].

Figura 95.- Interfaz de la aplicación de control

Figura 96.-Interfaz de la aplicación de registro

[21] El apartado "13. Anexos" contiene el Manual de Instrucciones de la aplicación en el caso de querer saber en profundidad el control y registro de datos del vehículo.

6. ESTUDIO ENERGÉTICO Y PRUEBAS EXPERIMENTALES

Un estudio energético permite evaluar y mejorar el modo de explotación y el funcionamiento de las prestaciones de un prototipo. Del mismo modo que comprobar si es factible o no le vehículo diseñado.

Centrando este punto en pruebas como vehículo para la validación de las prestaciones desarrolladas evaluaremos el consumo y funcionamiento de los elementos que constituyen el prototipo.

Para poder hacer el estudio de nuestro sistema, se han estudiado los consumos de los diferentes subsistemas en el laboratorio. De esta manera, hemos podido hacer un dimensionado de la batería para poder poner nuestro prototipo en funcionamiento el tiempo necesario para realizar las pruebas experimentales.

En la siguiente tabla encontramos los consumos de los diferentes componentes del sistema, testeados en el laboratorio:

Tabla 2.- Consumos de los elementos del prototipo

	Tensión	Corriente	Potencia
ARDUINO® Mega 2560	8,89V	8mA	71,12mW
S. Temp/Hum – DHT11	4,99V	1mA	4,99mW
Acelerómetro – ADXL335	3,29V	1mA	4,99mW
Brújula – HMC5883L	4,99V	1mA	4,99mW
GPS – GPSNEO6M	4,99V	6,4mA	31,94mW
Xbee – Xbee Pro S1	3,29V	5,9mA	19,41mW
Motor (Duty Cicle = 50%)	12,53V	4,7A	58,89W
Baliza	11,99V	0,625A	7,5W
Bocina	11,99V	1,25A	15W

Una vez realizado este estudio, concluimos, que el consumo total de nuestro sistema, en condiciones normales, donde la bocina no va a estar en uso, es de 10,231 A aproximadamente.

Por otro lado, en caso de necesitar el uso de la bocina, como indica el reglamento, se deberá activar 2 segundos cada 2 minutos, esto supondría un consumo añadido de 1,25 A en una suma total de 1 minuto cada hora. Por lo tanto, mientras esté en funcionamiento la bocina, el consumo total asciende a 11,481 A.

Los tres elementos que se han tratado en profundidad durante la realización de este estudio han sido los reguladores, los motores y finalmente, la batería.

6.1. Ensayo de Reguladores

Los reguladores utilizados son reguladores lineales, los cuales suelen tener algo de rizado, que se amortigua con los condensadores que lleva en su montaje. Aun así, no suelen presentar un gran rendimiento gran rendimiento.

6.1.1. Regulador LM7812

Este regulador alimenta la baliza y la bocina, así como al regulador LM7809. Para realizar el cálculo del rendimiento de todos los reguladores, hemos utilizado la siguiente fórmula:

$$\eta = \frac{P_{out}}{P_{\mathbf{in}}} = \frac{V_{out} \cdot I_{out}}{V_{\mathbf{in}} \cdot I_{\mathbf{in}}} \qquad (9)$$

Una vez testeados las tensiones y corrientes de entrada y salida en el montaje final, comprobamos que la intensidad de entrada y de salida de los reguladores, son exactamente iguales, por lo tanto, el rendimiento de éstos, se basara en la división de las tensiones *out*/*in* de los reguladores.

$V_{in} = V_{batería}$= 12,53V $\qquad$ $I_{in}= I_{out}$= 0,831A

V_{out} = 11,99 V

$$\eta = \frac{P_{out}}{P_{\mathbf{in}}} = \frac{V_{out} \cdot I_{out}}{V_{\mathbf{in}} \cdot I_{\mathbf{in}}} = \frac{V_{out}}{V_{\mathbf{in}}} = \frac{11.99\,V}{21.53\,V} \rightarrow 95,69\,\%$$

El estudio realizado sobre el rizado de este regulador, concluye en las siguientes imágenes:

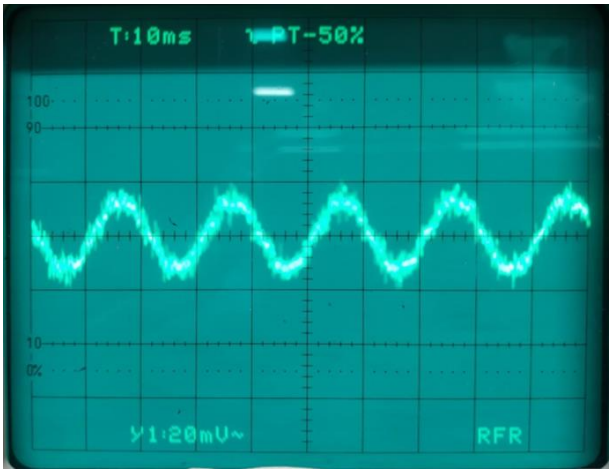


Figura 97.- Señal de salida del regulador LM7812 alimentado a 12V sin aplicación de condensadores

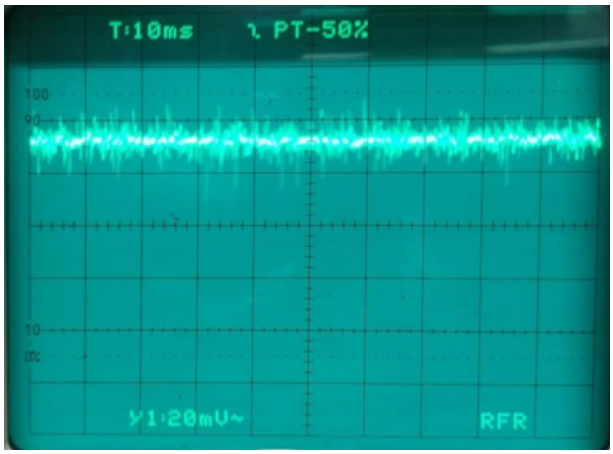


Figura 98.- Señal de salida del regulador LM7812 alimentado a 12V aplicando condensadores

6.1.2. Regulador LM7809

Siguiendo el procedimiento del regulador anterior, encontramos los siguientes valores:

$V_{in} = V_{batería} = 12{,}53V$ $I_{in} = I_{out} = 0{,}206A$

$V_{out} = 11{,}99$ V

$$\eta = \frac{P_{out}}{P_{\mathbf{in}}} = \frac{V_{out} \cdot I_{out}}{V_{\mathbf{in}} \cdot I_{\mathbf{in}}} = \frac{V_{out}}{V_{\mathbf{in}}} = \frac{8.89\,V}{11.99\,V} \rightarrow 74.14\,\%$$

Un punto a tener en cuenta es que ARDUINO® se alimenta a partir de este regulador, hecho que implica que el regulador de 3,3V que incorpora ARDUINO® también. Esto conlleva una mayor entrada de intensidad en este regulador debido a los componentes que son alimentados a 3,3V.

El estudio realizado sobre el rizado de este regulador, concluye en las siguientes imágenes:

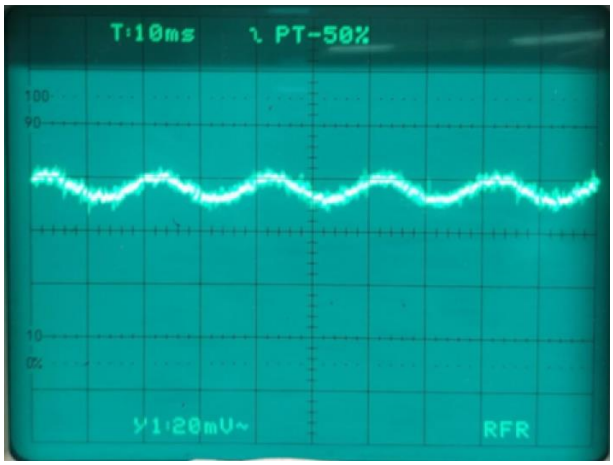


Figura 99.- Señal de salida del regulador LM7809 alimentado a 12V sin aplicación de condensadores

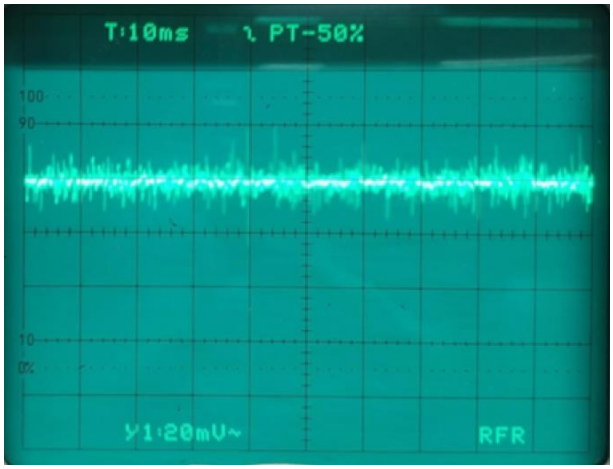


Figura 100.- Señal de salida del regulador LM7809 alimentado a 12V aplicando condensadores

6.1.3. Regulador LM7805

Por último, realizamos el estudio del regulador a 5V:

V_{in} = = 8,89V I_{in}= I_{out} = 0,066A

V_{out} = 4,99V

$$\eta = \frac{P_{out}}{P_{\mathbf{in}}} = \frac{V_{out} \cdot I_{out}}{V_{\mathbf{in}} \cdot I_{\mathbf{in}}} = \frac{V_{out}}{V_{\mathbf{in}}} = \frac{4.99\ V}{8.89\ V} \rightarrow 56.13\,\%$$

El estudio realizado sobre el rizado de este regulador, concluye en las siguientes imágenes:

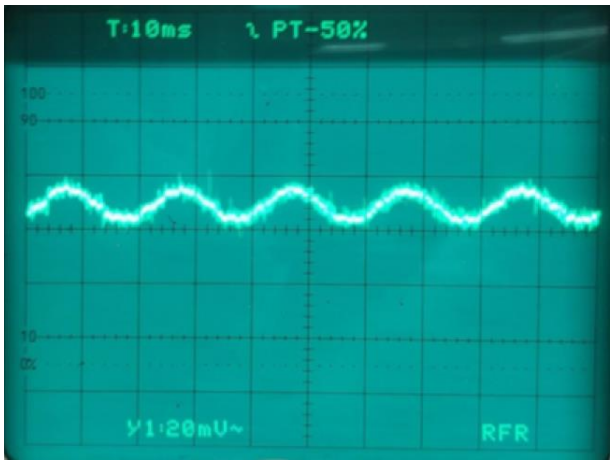


Figura 101.- Señal de salida del regulador LM7805 alimentado a 9V sin aplicación de condensadores

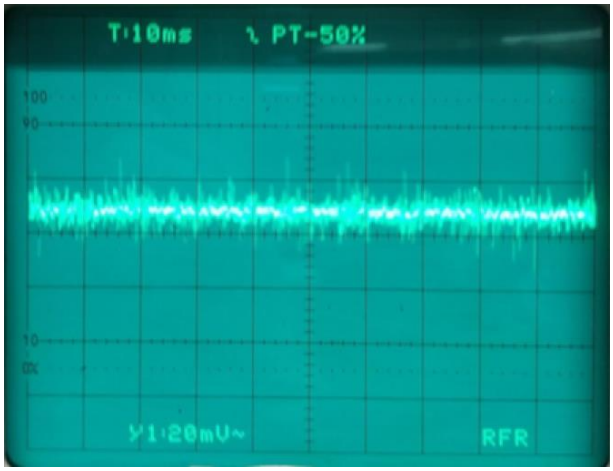


Figura 102.- Señal de salida del regulador LM7805 alimentado a 9V aplicando condensadores

Tras visualizar las señales anteriores en el osciloscopio, podemos observar que los condensadores del montaje de cada regulador, amortiguan mucho el rizado de las señales.

En los montajes sin condensadores, el rizado de la señal es notoriamente sinusoidal, señal mediante la cual se obtendría un resultado más irregular.

Finalizando el montaje aplicando los condensadores, podemos observar que la señal mantiene estacionariamente un valor estable a lo largo del tiempo.

Encontramos, en el caso del montaje sin condensadores, un rizado de unos 20mV aproximadamente, mientras que, en el montaje con condensadores, obtenemos una señal continua, aunque se puede apreciar cierto ruido, posiblemente debido al uso del osciloscopio. En el mercado podemos encontrar reguladores más estables y más

eficientes, por ejemplo, los conmutables, descartados a favor de éstos puesto que suplen totalmente las necesidades del prototipo.

6.2. Ensayo de Motores

Como se comentó en el punto "5.2.2. Consumo de Motores" el ensayo realizado sobre los motores con tal de tener una primera toma de contacto con los métodos de comunicación entre ARDUINO® y LabVIEW™ nos ofreció a su vez información sobre las características de los motores en cuanto a sobrepicos de arranque, corrientes inversas, etc.

Para poder llevar a cabo el ensayo se nos cedió una de las fuentes de la universidad, el inconveniente de esta fuente es el hecho de que limita la corriente consumida a un máximo de 3 A, por lo tanto, decidimos efectuar tres ensayos a diferentes tensiones con tal de ver el comportamiento de los motores y tratar de no limitar su funcionamiento debido a la corriente consumida. Así pues, las tensiones a las cuales se han realizado las pruebas han sido 5 V, 7 V y 9 V. Todos estos ensayos pueden estar sujetos a ruidos o perturbaciones al haber utilizado *Protoboards* y cableado de la universidad para su *Setup*, que ha sido utilizado a lo largo de varios años y no nos ofrece un funcionamiento tan bueno como debería.

6.2.1. Ensayo a 5 V

Los valores obtenidos de las gráficas del panel frontal de LabVIEW™ son:

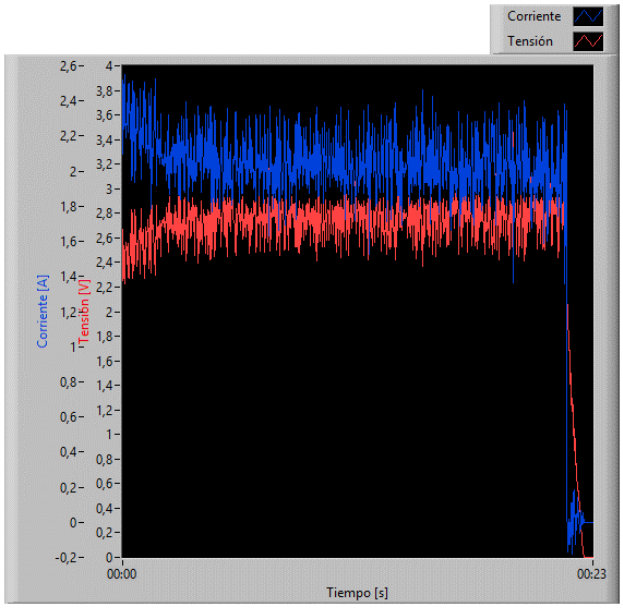


Figura 103.- Gráfica en LabVIEW™ para los valores de tension y corriente a 5 V

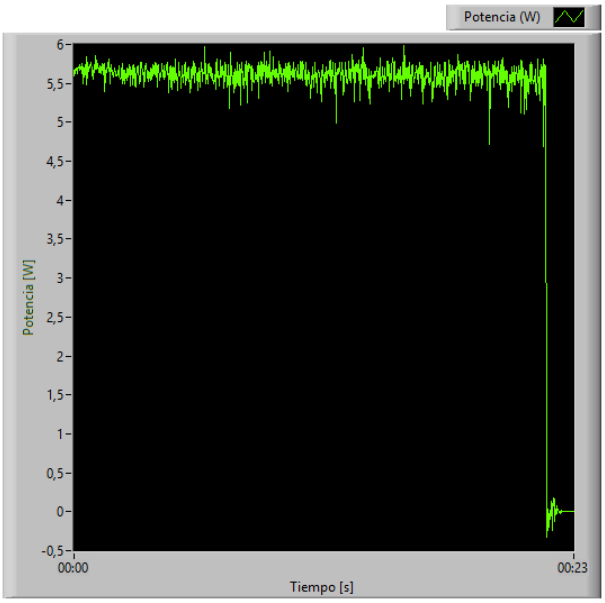


Figura 104.- Gráfica en LabVIEW™ para el valor de potencia a 5 V

En estas gráficas se puede apreciar comportamientos como el de la tensión y la corriente tanto al encendido como en el apagado de la fuente, pero nos es imposible dar valores a ojo a los picos de corriente o al tiempo de caída de tensión al apagado. Para poder tomar estos datos con precisión, LabVIEW™ nos permite exportar los datos registrados a una hoja de cálculo, sobre la cual podremos trabajar.

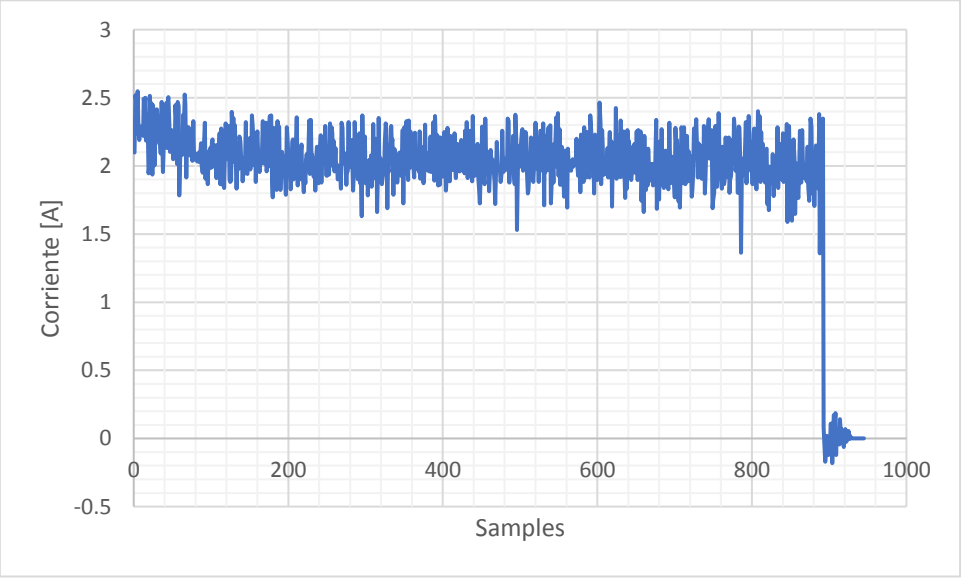


Figura 105.- Gráfica en Microsoft Excel® para los valores de corriente a 5 V

A través del uso de las hojas de cálculo podemos obtener los valores exactos de, por ejemplo, el pico de corriente al arranque del motor, que en este caso es de 2,548 A, o de como la corriente se estabiliza alrededor de los 2 A pasado cierto tiempo. Incluso se puede apreciar una corriente negativa en el momento de apagar el motor.

Los motores de DC realizar el giro transformado energía eléctrica en mecánica mediante un sistema de bobinas e imanes en el rotor o estator del motor[22], al dejar de recibir energía eléctrica, pero manteniendo parte de su energía mecánica por la inercia de giro, realizan el paso inverso, es decir, se transforman en un generador durante un pequeño periodo de tiempo, en el cual ofrecen corriente en vez de consumirla. En este caso en concreto, la corriente que ha llegado a proporcionar el motor ha sido de 181,3 mA.

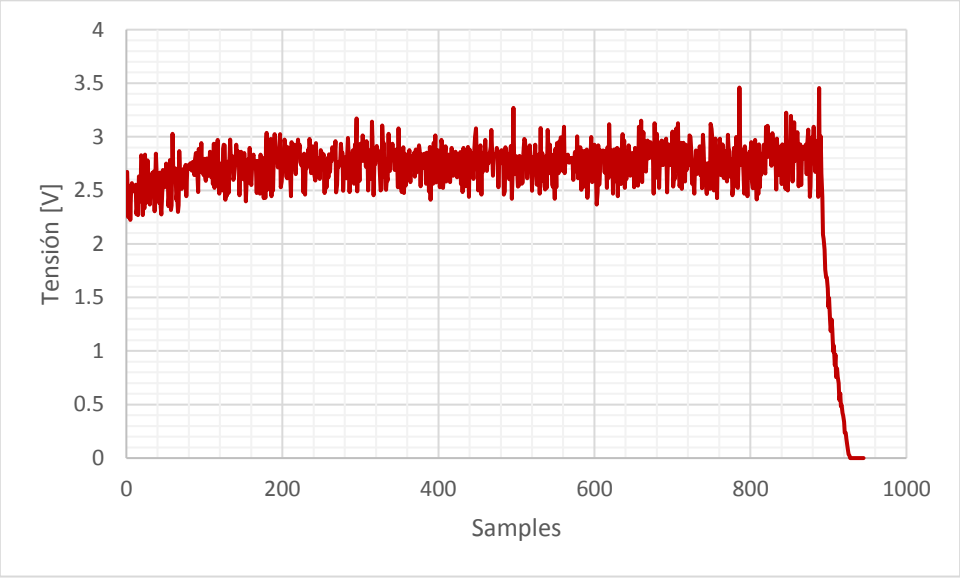


Figura 106.- Gráfica en Microsoft Excel® para los valores de tensión a 5 V

Del mismo modo en el caso de la corriente podemos ver como el motor tarda alrededor de 200 *samples* (5 segundos) en estabilizarse a 2,8 V y como hace una pendiente al apagar la fuente de alimentación en un periodo de algo menos de 1 segundo.

[22]En este motor y en la amplia mayoría, es en el estator (parte exterior fija del motor) donde se suele encontrar el complejo de bobinas y, en el rotor (parte móvil situada en el interior del estator) se encuentran situados los imanes permanentes. De todos modos, no siempre se cumple esta disposición y existen motores en los cuales los imanes han intercambiado su posición con las bobinas y viceversa.

Del mismo modo, se pude apreciar en la siguiente gráfica como la potencia se mantiene constante desde el inicio a aproximadamente 5,6 W y llega a proporcionar hasta 333,6 mW.

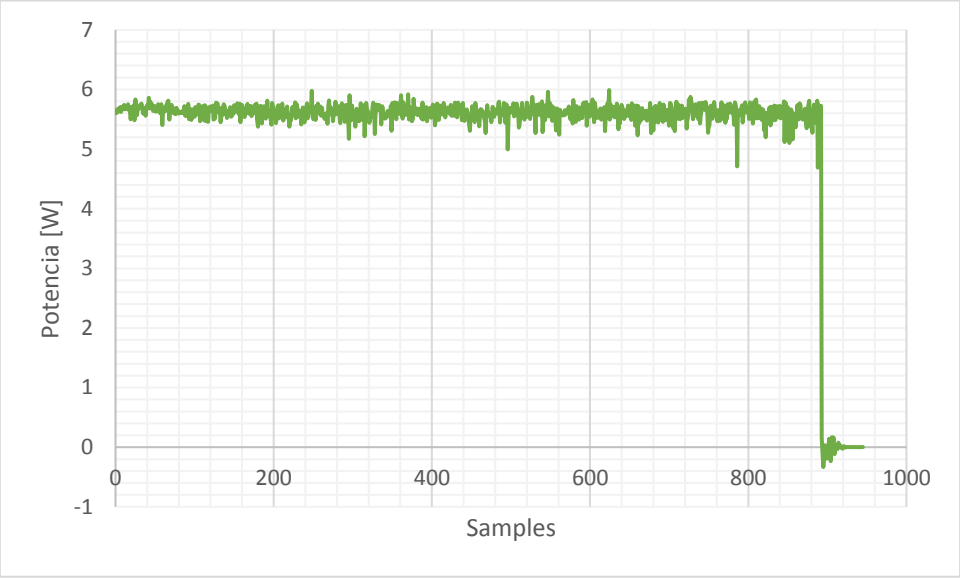


Figura 107.- Gráfica en Microsoft Excel® para los valores de potencia a 5 V

Los datos obtenidos de este ensayo quedan registrados en la siguiente tabla:

Tabla 3.- Resultados del ensayo de motores a 5 V

	Corriente
Valor medio	2,0041 A
Valor máximo	2,5480 A
Valor máximo proporcionado	181,30 mA

	Tensión
Valor medio	2,7979 V
Tiempo de arranque	4,9735 s
Tiempo de apagado	0,9947 s

	Potencia
Valor medio	5,6250 W
Valor máximo proporcionado	333,62 mW

6.2.2. Ensayo a 7 V

En este caso, los valores obtenidos de las gráficas del panel frontal de LabVIEW™ han sido:

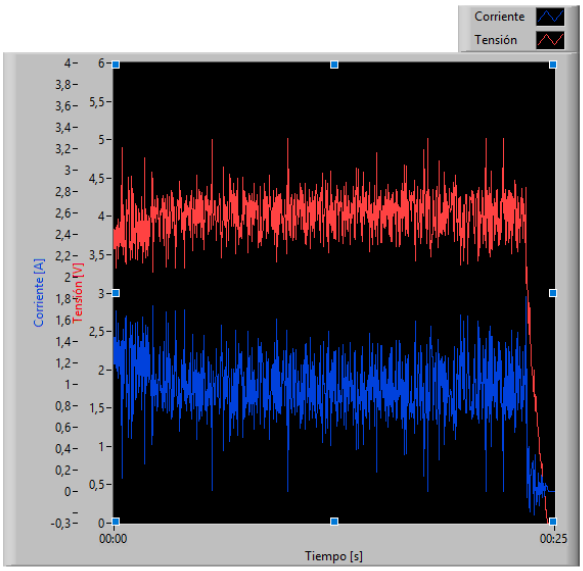


Figura 108.- Gráfica en LabVIEW™ para los valores de tensión y corriente a 7 V

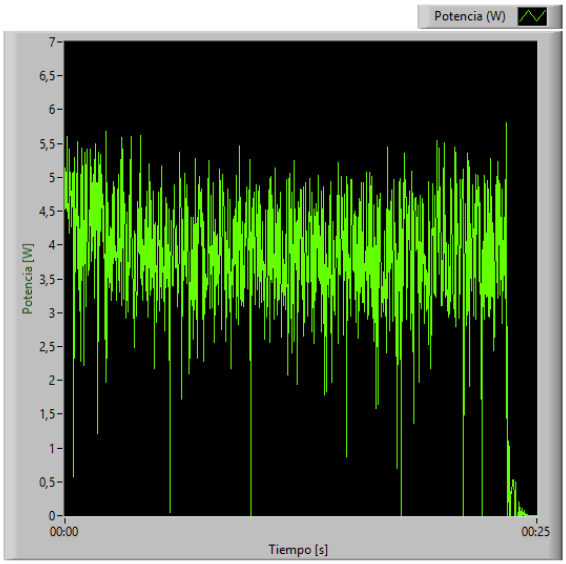


Figura 109.- Gráfica en LabVIEW™ para los valores de potencia a 7 V

En este caso, los valores no se aprecian tan bien como en las anteriores gráficas obtenidas en LabVIEW™, por lo tanto, es mejor recurrir directamente a los valores exportados a la hoja de cálculo.

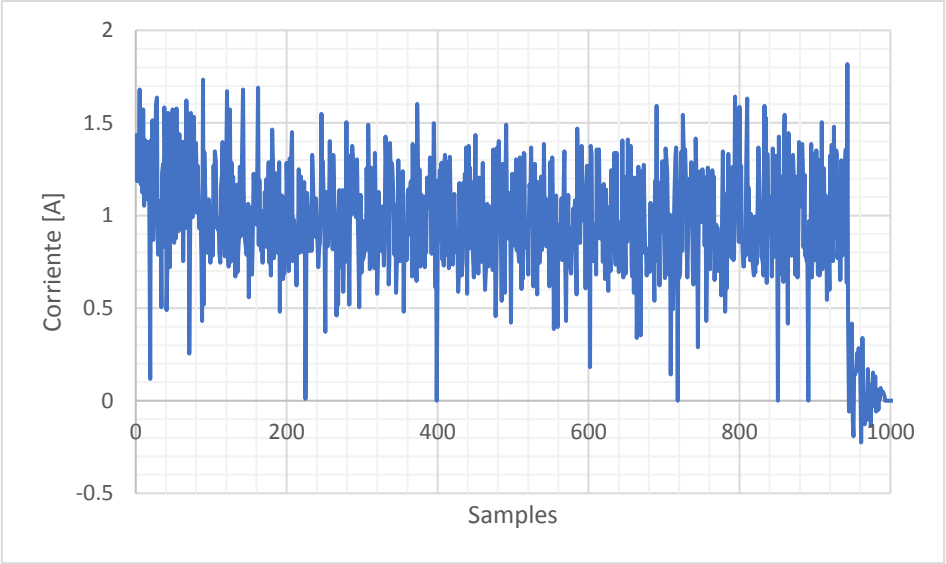


Figura 110.- Gráfica en Microsoft Excel® para los valores de corriente a 7 V

En este caso, debido probablemente a las permutaciones por el ruido de los elementos utilizados en el ensayo, no se aprecia de forma tan nítida el pico de arranque, de todas formas, lo que sí se puede distinguir es la recta que forma la corriente al inicio del ensayo. Esta recta, pese a no ser muy llamativa, queda retratada en la siguiente figura:

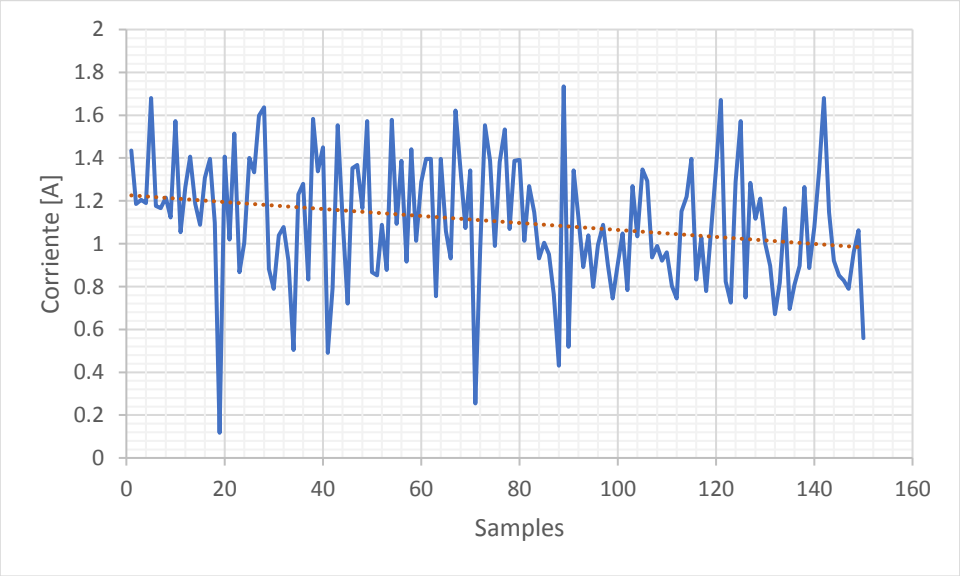


Figura 111.- Detalle de los valores de corriente durante el arranque a 7 V

En el caso de este ensayo, el pico de corriente llega a ser de hasta 1,8 A, posteriormente se estabiliza en aproximadamente 0,9 A y finalmente, en el momento del apagado es capaz de proporcionar 225,4 mA.

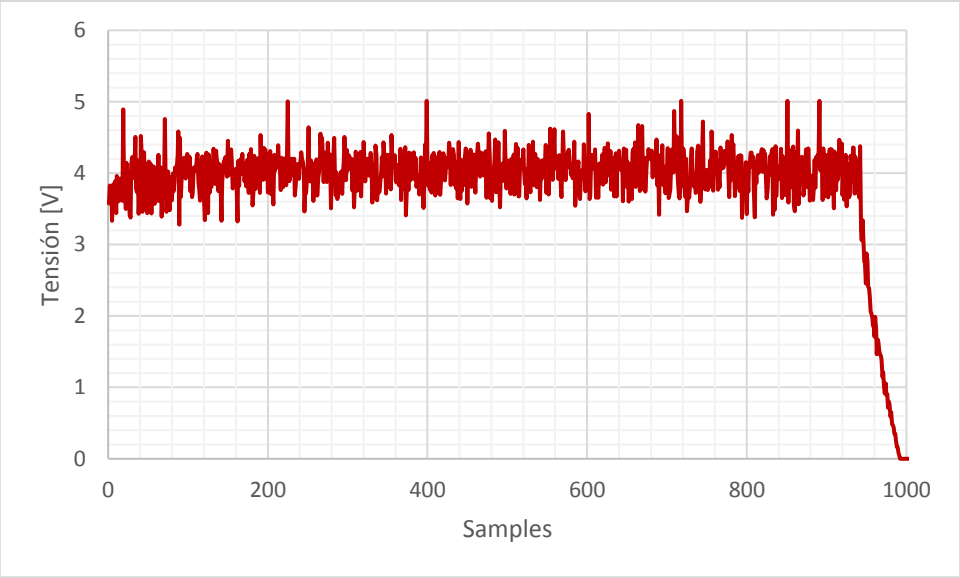


Figura 112.- Gráfica en Microsoft Excel® para los valores de tensión a 7 V

En el caso de la tensión, tampoco se aprecia la misma recta de carga, este fenómeno puede darse a que al motor le "cuesta" menos llegar hasta la tensión de estabilización, ya que se le está alimentando a una tensión más elevada con la restricción de los 3 A de la fuente de alimentación. En este caso, de igual manera se puede ver un valor ligeramente superior en el tiempo de apagado que en caso anterior, que es de algo más de 1 segundo.

Finalmente, la potencia consumida por el motor bajo estas condiciones ha sido de aproximadamente 3,8 W y se aprecia una pequeña rampa en el encendido. Esta rampa se debe a que el pico de corriente no es lo suficientemente elevado como para contrarrestar la rampa de arranque de la tensión en el motor. En este caso, la potencia que ha llegado a generar el motor ha sido de 529 mW.

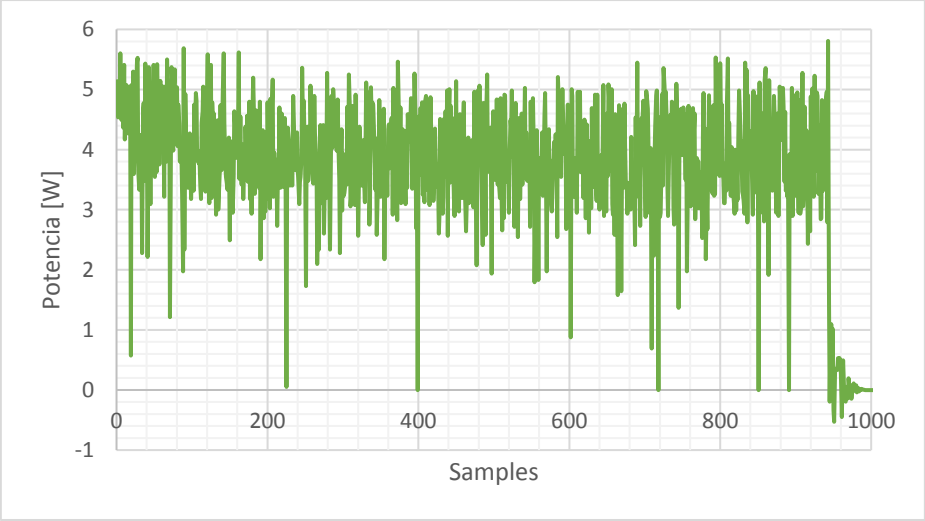


Figura 113.- Gráfica en Microsoft Excel® para los valores de potencia a 7 V

Para este ensayo, los valores obtenidos se corresponden a:

Tabla 4.- Resultados del ensayo de motores a 7 V

	Corriente
Valor medio	0,9310 A
Valor máximo	1,8179 A
Valor máximo proporcionado	225,40 mA

	Tensión
Valor medio	4,0768 V
Tiempo de arranque	1,9450 s
Tiempo de apagado	1,2622 s

	Potencia
Valor medio	3,8001 W
Valor máximo proporcionado	529,06 mW

6.2.3. Ensayo a 9 V

Para el último ensayo realizado sobre los motores, los valores obtenidos de las gráficas del panel frontal de LabVIEW™ han sido:

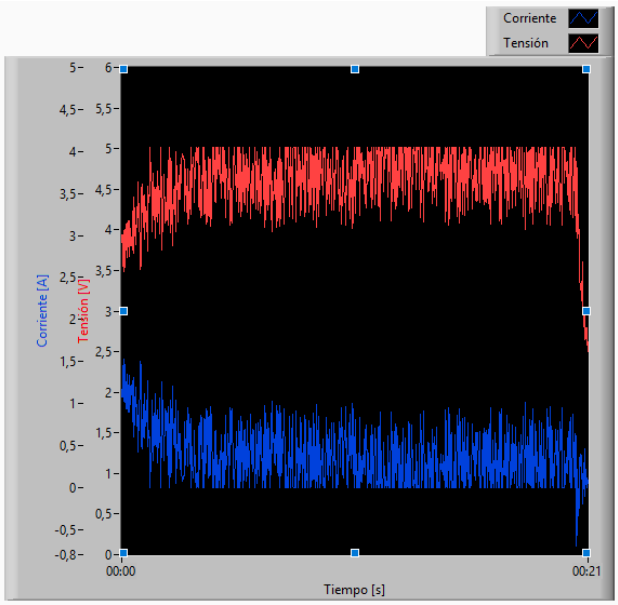


Figura 114.- Gráfica en LabVIEW™ para los valores de tension y corriente a 9 V

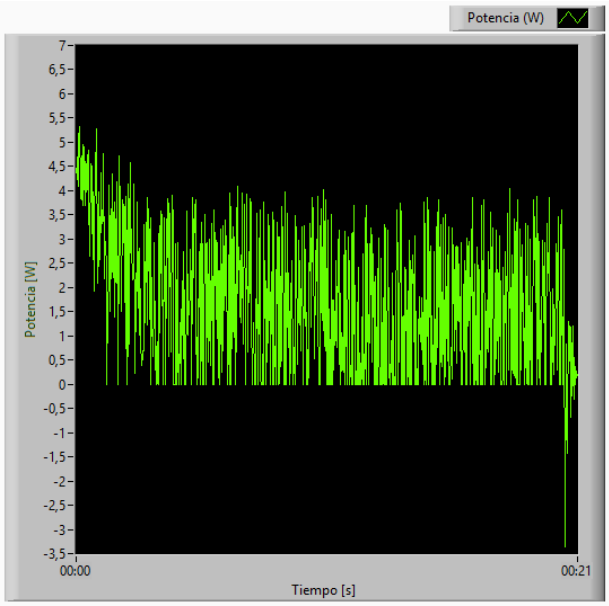


Figura 115.- Gráfica en LabVIEW™ para los valores de potencia a 9 V

En este caso, el ruido ha aumentado drásticamente, lo que nos dificulta la medición de algunos valores.

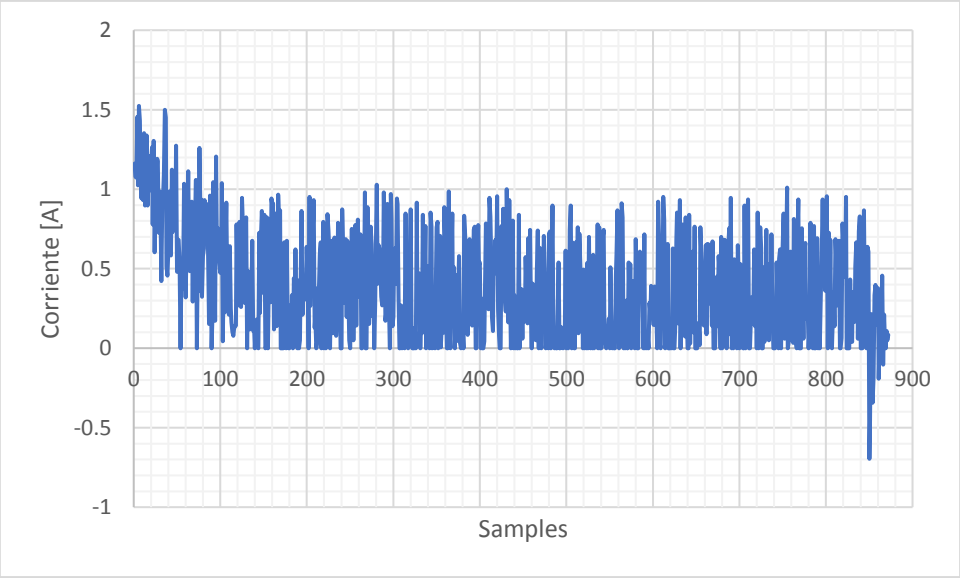


Figura 116.- Gráfica en Microsoft Excel® para los valores de corriente a 9 V

De nuevo volvemos a preciar un consumo elevado en el momento del arranque. En concreto el pico máximo llega hasta los 1,5 A. Tras el arranque, la señal se estabiliza en aproximadamente 0,3 A, este comportamiento se puede deber a la restricción de corriente por parte de la fuente de alimentación.

Finalmente, en el momento de paro, el motor es capaz de suministrar hasta 695 mA.

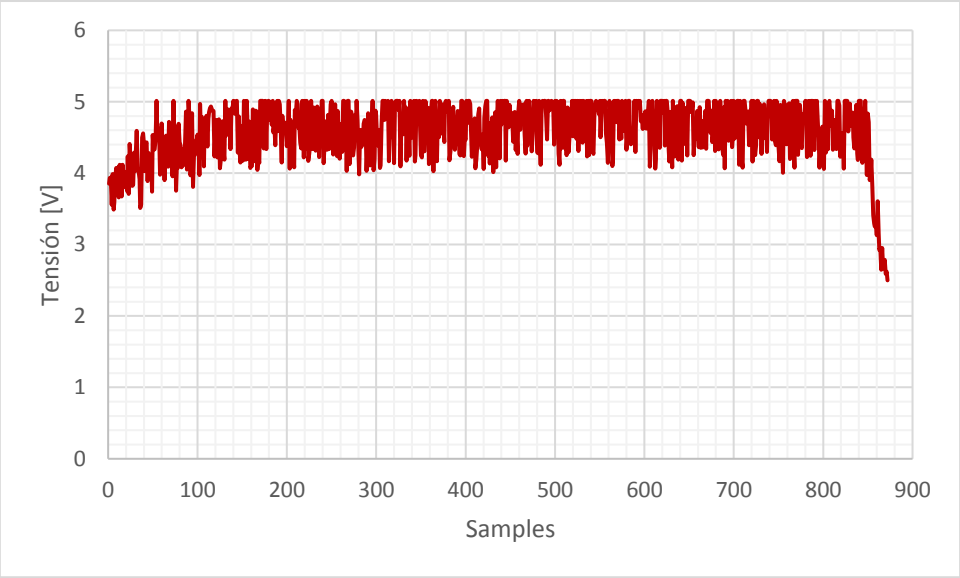


Figura 117.- Gráfica en Microsoft Excel® para los valores de tensión a 9 V

Para el caso de la tensión, se aprecia la recta de carga tras el arranque junto con lo que parece ser una saturación de tensión a 5 V, esto es debido en primer lugar a que la fuente no suministra los 9 V establecidos de manera constante y uniforme, sino que reduce algo este valor en función de lo que consume el elemento conectado a ella, por otra parte, la resistencia *shunt* utilizada para los ensayos es una resistencia de potencia de 1 Ω con de 50 W, es decir, al tener la fuente de alimentación una restricción de 3 A, estamos forzando a la resistencia *shunt* a tener una caída de tensión de 3 V, por lo tanto, la caída de tensión en el motor será de algo menos de 9 V (proporcionados por la fuente) menos los 3 V de caída de la *shunt*.

$$V_{Shunt} = R_{Shunt} \cdot I_{FA} \qquad (3)$$

$$V_{Motor} = V_{FA} - V_{Shunt} \qquad (4)$$

Por ejemplo:

$R_{Shunt} = 1\ \Omega$
$I_{FA} = 3\ A$
$V_{FA} = 7,5\ V$

$$V_{Shunt} = R_{Shunt} \cdot I_{FA} = 1\ \Omega \cdot 3\ A = 3\ V$$

$$V_{Motor} = V_{FA} - V_{Shunt} = 7,5\ V - 3\ V = 4,5\ V$$

Finalmente, la potencia consumida se ve alterada por todas las restricciones que se han tenido a lo largo de este ensayo. La forma final de la onda estudiada es la siguiente:

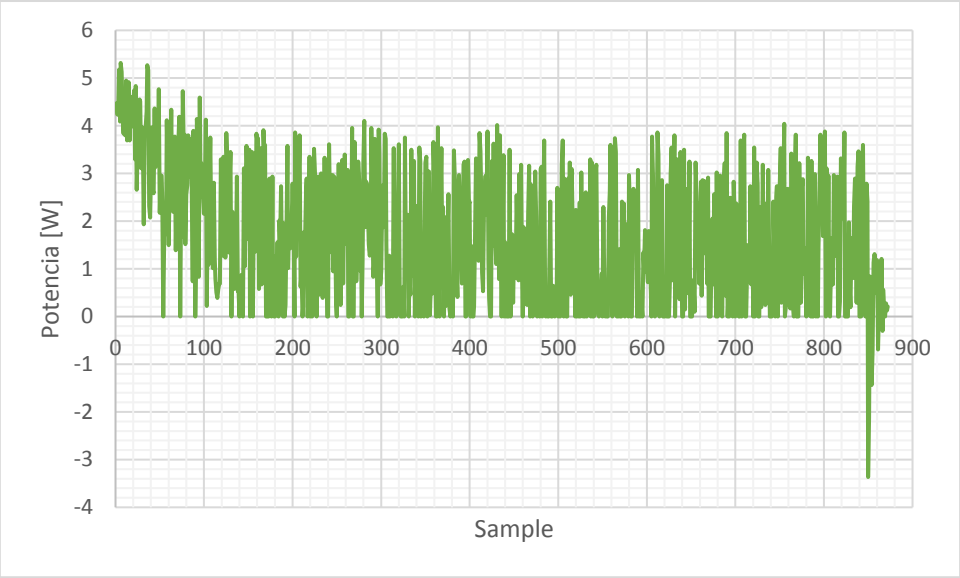


Figura 118.- Gráfica en Microsoft Excel® para los valores de potencia a 9 V

Los resultados para este ensayo son los siguientes (estos datos pueden no ser idóneos debido a las limitaciones con las que hemos contado a la hora de realizarlo):

Tabla 5.- Resultados del ensayo de motores a 5 V

	Corriente
Valor medio	0,2744 A
Valor máximo	1,5239 A
Valor máximo proporcionado	695,88 mA

	Tensión
Valor medio	4,7334 V
Tiempo de arranque	2,566 s
Tiempo de apagado	0,5251 s

	Potencia
Valor medio	1,3438 W
Valor máximo proporcionado	3,365 W

6.3. Batería

La elección de la batería ha sido muy condicionada por el presupuesto disponible para el proyecto, optando así por la más económicamente viable del mercado.

Una vez se ha finalizado el estudio de consumos de nuestro sistema al completo, se ha determinado que la batería que suple nuestras necesidades energéticas y se ajusta a nuestro presupuesto es una batería de plomo-ácido de 12V y 8Ah.

Esta batería, según el estudio de consumos anterior, nos va a permitir un funcionamiento de visto en el apartado "6. Estudio Energético y Pruebas

Experimentales" unos 30 minutos; más que suficiente para las pruebas de campo de nuestro prototipo (sin tener en cuenta los picos de arranque que pueden llegar a ser excesivos).

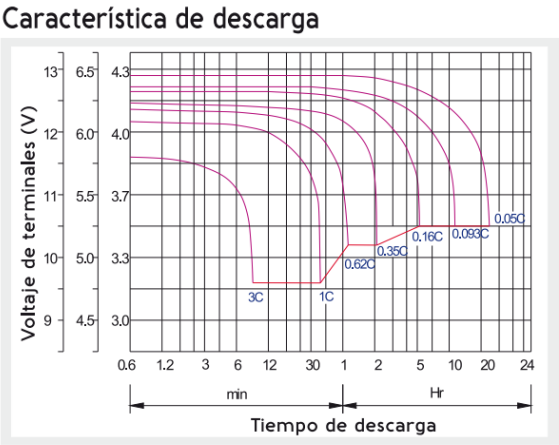


Figura 119.- Característica de descarga de la batería seleccionada extraída del *datasheet*

En el mercado encontramos baterías con mejor rendimiento, como pueden ser las de litio, pero para nuestras necesidades, hemos escogido el modelo SOLARX-8 C120 del fabricante Xunzel debido a que se trata de una batería fácil de encontrar y accesible al gran público.

A modo de información adicional, presentamos una tabla con el recopilatorio de las mejoras en las prestaciones entre las baterías de litio y las de plomo-ácido:

Tabla 6.- Comparativa entre baterías de litio y de plomo-ácido [16]

	LÍTIO	PLOMO-ÁCIDO
EFICIENCIA DE CARGA	97%	85%
AUTODESCARGA	Prácticamente nula	5% mensual
CAPACIDAD DE DESCARGA PROFUNDA	90%	60%
GARANTIA	10 años	2 años
RAPIDEZ DE CARGA	Gran capacidad de carga	Se deben cargar con entre un 10% y un 20% de su capacidad nominal, y el último 15% de carga (etapa absorción) se realiza de manera lenta e ineficiente. Para cargarlas completamente se necesitan entre 9 y 12 horas.
EMISIÓN DE GASES	No emite gases	Pueden emitir gases corrosivos.

7. PRUEBAS DE CAMPO

Bien es cierto que las pruebas experimentales en tierra han sido un éxito, pero el prototipo no está ideado con el fin de trabajar en estas condiciones. Es por este motivo que se realizaron pruebas en el mar con tal de comprobar la respuesta de las diferentes partes del proyecto del mismo modo que se hizo en tierra.

El lugar donde se realizaron las pruebas fue en las zonas de baño del fórum en el puerto de Sant Adrià del Besos, Barcelona.

La intención de la prueba era testear tanto el funcionamiento de la aplicación como la flotabilidad del vehículo de apoyo. Pero surgieron varios inconvenientes:

Pese al correcto funcionamiento en tierra, el magnetómetro HMC5883L comenzó a presentar un comportamiento inesperado bloqueando el tránsito de información entre LabVIEW™ y ARDUINO®, fue por este motivo que se desestimó el uso de la brújula para este ensayo en pos de solucionar el problema por separado.

El hecho de no utilizar la brújula impide el correcto funcionamiento del algoritmo de control de trayectoria, el cual depende únicamente de la brújula y el GPS.

Por otro lado, el transporte hasta el puerto forzó el sellado de uno de los motores, suponiendo un posible riesgo en la estanqueidad del del conjunto encapsulado - eje de motor, por lo tanto, se decidió retirar los motores con el fin de centrar la prueba en el correcto funcionamiento de la aplicación y el ensayo de flotabilidad pertinente del vehículo.

Tras el nuevo sellado del encapsulado se realizaron nuevas pruebas, en este caso en la playa de Sant Adrià del Besos, Barcelona, las cuales se centraron en el correcto funcionamiento y el sellado de posibles fugas.

7.1. Test 1

El primer test, realizado en el fórum, se realizó con la intención de ensayar la flotabilidad y la aplicación en general. Los resultados se encuentran expuestos en los siguientes puntos.

7.1.1. Flotabilidad

El estudio de flotabilidad resultó ser un éxito rotundo. Tras las simulaciones realizadas en el apartado “3.4. Ensamblaje Definitivo” se concluyó un centro de gravedad simulado (ver figuras 24 y 25) que ha resultado encontrarse sobre la línea de flotabilidad del prototipo hallada tras haber realizado esta prueba,

En las siguientes imágenes se puede apreciar esta línea de flotabilidad encontrada aproximadamente unos 5 cm por debajo de la parte inferior de la caja estanca diseñada.

Figura 120.- Pruebas de campo del prototipo

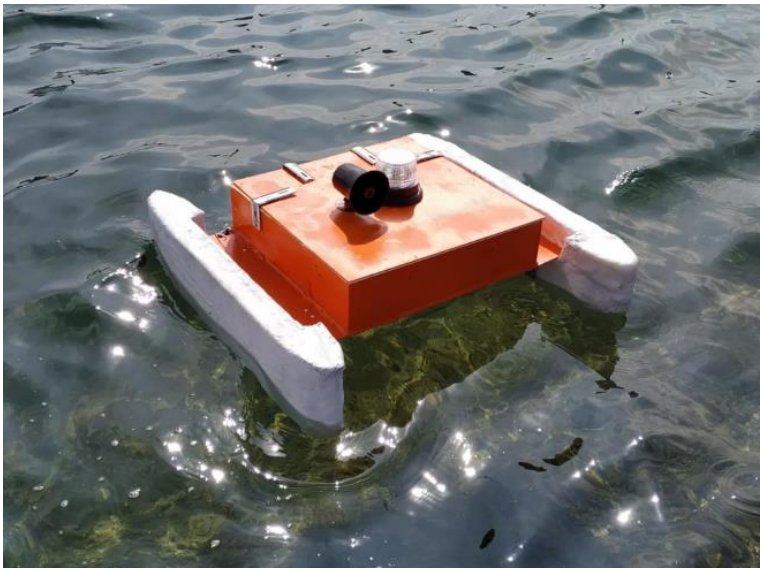


Figura 121.- Pruebas de campo del prototipo

El resultado obtenido se ha dado por válido y el diseño mecánico implementado en el conjunto caja hermética-flotadores un éxito.

7.1.2. Aplicación

Del mismo modo que el diseño mecánico, también se testearon el software y el hardware mediante las aplicaciones de Control y Registro. De hecho, el correcto funcionamiento de cualquiera de las dos es más que suficiente para verificar el comportamiento de la aplicación, puesto que son aplicaciones hermanadas que reciben y envían datos simultáneamente.

Al margen de los resultados de la brújula, que no han sido visibles (se mantienen en 101º que es el valor de calibración), el resto de sensores y actuadores funcionaron acorde a lo esperado.

La siguiente captura, que corresponde a la aplicación de Control, muestra la señal de temperatura, humedad e inclinación del vehículo mientras se encontraba en el agua.

Figura 122.- Interfaz de control durante las pruebas de campo

Como se puede apreciar, la restricción del GPS a las diezmilésimas hace menguar ligeramente la posición, mostrando que se encuentra en tierra mientras que realmente se encuentra en la orilla del mar. La temperatura y la humedad mantienen valores constantes, así como el acelerómetro indicando la inclinación que, pese al *delay*, responde notablemente bien. La baliza y la bocina también respondieron como era de esperar, del mismo modo que en tierra.

Del mismo modo, el registro de datos fue un éxito, obteniendo los datos en hojas de cálculo Microsoft Excel® entre las cuales, por ejemplo, encontramos la de temperatura y humedad, correspondiente a la siguiente tabla:

Tabla 7.- Datos registrados de temperatura y humedad durante las pruebas de campo

Time	Temperatura [ºC]	Humedad [%]
28/09/2019 11:49:59,631	21	87
29/09/2019 11:50:00,132	21	87
30/09/2019 11:50:00,947	21	87
01/10/2019 11:50:01,977	21	87
02/10/2019 11:50:02,763	22	87
03/10/2019 11:50:03,627	22	87

Los resultados obtenidos son satisfactorios para este ensayo, además gracias a estas últimas hojas de cálculo obtenidas mediante Microsoft Excel®, percibimos la irregularidad (dentro de unos márgenes) de la obtención de datos, puesto que los periodos entre registro fluctúan entre el medio segundo y el segundo completo.

La respuesta para este incidente es que, al mantener abiertas ambas aplicaciones simultáneamente, los bits de entrada de datos se acumulan durante un instante (concretamente 38 bits), es decir, al no producirse el muestreo de los datos en ambas aplicaciones en el mismo momento, los datos que deberían mostrarse en ambas se acumulan momentáneamente para posteriormente liberarse cada una en su respectiva interfaz, pero al producirse una retención en los datos de una u otro aplicación, el resultado es un *delay* de hasta 1 segundo en la aplicación que se ha puesto en funcionamiento en último lugar.

De todos modos, las fluctuaciones en los tiempos de registro se consideran ínfimas en comparación a los tiempos que deberá mantenerse en funcionamiento el prototipo, por lo tanto, se establece como una variación en la precisión que no afecta al comportamiento de ninguna de las dos aplicaciones de manera preocupante o que altere los resultados.

7.2. Test 2

Tras estos resultados, y vista la imposibilidad de la prueba de motores, se sellaron de nuevo las junturas del encapsulado (esta vez con resina epoxy), como se puede apreciar en la siguiente imagen, en la cual se encuentran dos tipos de encapsulados con diferente color, diferenciando así el antiguo diseño del nuevo diseño.

Figura 123.- Prototipo post-reparación del encapsulado

Pese a la imposibilidad de las pruebas de control de autónomo debido al magnetómetro, esta vez sí que se pudieron testear los motores en un funcionamiento manual controlado.

La zona que se estableció como campo de pruebas en esta ocasión fue la playa de Sant Adrià del Besos, concretamente en el delta del rio Besos debido a la tranquilidad de sus aguas.

La localización se puede ver, de nuevo, en el GPS de la aplicación.

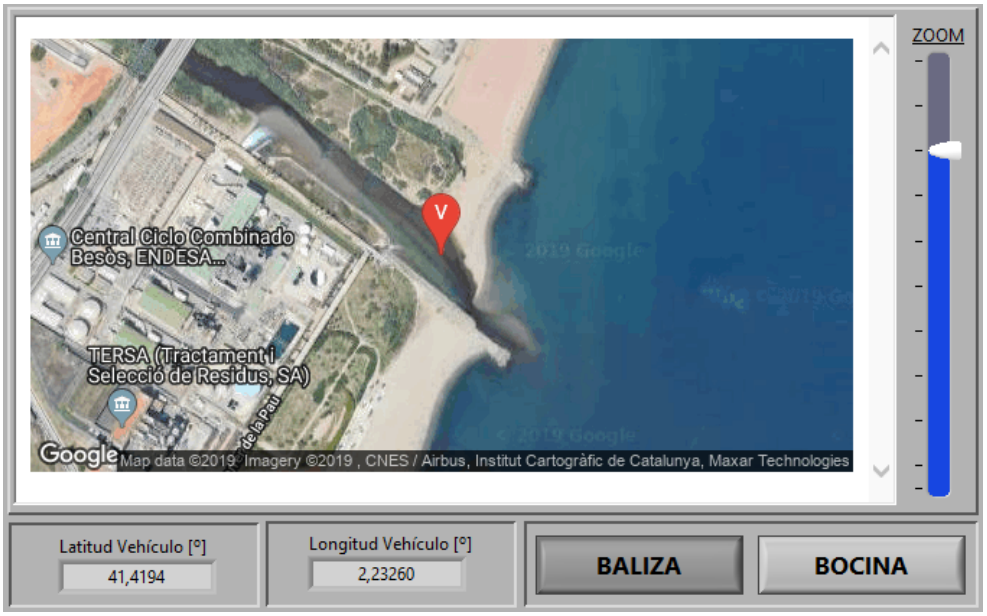


Figura 124.- Indicador GPS de posición en la desembocadura del rio Besós

La respuesta de todos los actuadores en estas condiciones fue la esperada y la sumersión de los motores en el agua no resultó en ninguna fuga, por lo tanto, los resultados obtenidos de este ensayo se consideraron exitosos.

Figura 125.- Encapsulado totalmente sumergidos para la segunda prueba

Tras realizar varias pruebas de control manual por el delta del rio Besós se pudo apreciar como los motores eran incapaces de impulsar el vehículo con la velocidad suficiente como para poder hablar de unas pruebas en alta mar, culpa de esto son tanto los motores (por no poder desplazar a gran velocidad el vehículo), como la batería que, al proporcionar una alimentación de 12V y 8Ah, tras un corto periodo de funcionamiento, el pico de arranque de ambos motores superaba los límites de consumo en corriente proporcionados.

Al margen de estos inconvenientes, a nivel de prototipo ha superado todos los test y se trata de un vehículo funcional con una aplicación de control y registro igualmente funcionales.

8. IMPACTO MEDIOAMBIENTAL DE MATERIALES

Una vez finalizado el prototipo, hemos querido tener en cuenta un pequeño estudio sobre el impacto medioambiental que tienen algunos de los materiales utilizados en nuestro proyecto.

Tanto a nivel de mecánica como en hardware, hemos valorado, que los dos elementos principales de mayor impacto medioambiental, al margen de los evidentes, como podrían ser pinturas o pastas adhesivas, son la batería de plomo-ácido y el uso de fibra de vidrio junto a la resina de poliéster utilizada.

8.1. Batería Plomo-Ácido

Este tipo de baterías poseen dos sustancias peligrosas: el electrolito ácido y el plomo. El primero es corrosivo, tiene alto contenido de plomo disuelto y en forma de partículas, y puede causar quemaduras en la piel y en los ojos. En tanto, el plomo es altamente tóxico para la salud humana, ingresa al organismo por ingestión o inhalación y se transporta por la corriente sanguínea acumulándose en todos los órganos, especialmente en los huesos [17].

El plomo de las baterías usadas suele ser reinsertado en el mercado por los trabajadores informales, quienes lo revenden; no obstante, en el proceso de extracción de este tóxico eliminan a los drenajes o a la tierra el ácido, lo que deriva en contaminación grave para el medio ambiente.

La disposición inadecuada de las escorias que se generan en la fundición es otra potencial fuente de contaminación de suelo y agua.

La exposición prolongada al plomo puede provocar anemia y problemas en el sistema nervioso central, cuyas consecuencias van desde sutiles cambios psicológicos y de comportamiento hasta graves efectos neurológicos, también genera intoxicación a partir de la inhalación.

Se trata de un elemento potencialmente contaminante que, en entrar en contacto con la biosfera marina, podría suponer un peligro para el ecosistema dañado.

8.2. Fibra de Vidrio y Resina de Poliéster

La fibra de vidrio junto a la resina de poliéster es conocido que presentan numerables problemas en su manipulación, tanto para la salud como para el medioambiente.

Son una causa importante de incapacidad y dermatitis. También pueden provocar irritación de las vías respiratorias, incluso asma por la inhalación de los vapores que se generan durante el proceso de curado.

Encontramos los siguientes puntos más relevantes en cuanto a contaminación:

- Volátiles o emanaciones que se producen durante el mezclado.

- Vapores desprendidos por disolventes orgánicos.
- Polvo procedente de labores de mecanizado.
- Polvo procedente de cargas añadidas a la resina, tales como sílice, caolín, etc.

Estas resinas son altamente inflamables, sus vapores pueden ser explosivos y pueden viajar desde el sitio de almacenamiento o de uso hasta encontrar una fuente de ignición. Sus vapores son más pesados que el aire, por este motivo tienden a acumularse en las partes bajas.

Sus recipientes deben protegerse de las fuentes de calor y fuego, igualmente de las chispas. Las descargas de la energía estática pueden ser suficientes para iniciar un incendio o una explosión [18].

9. PRESUPUESTO

Materializar un prototipo como el que se ha llevado a cabo para la realización de este proyecto es una empresa laboriosa si solo se consta de dos desarrolladores, ha implicado una inmensa cantidad de horas a lo largo de un año desde los primeros esbozos. Pero este no ha sido el único campo sobre el cual se ha debido de realizar esfuerzo, sino que el coste total del sistema en si también supone una dificultad añadida para el modesto presupuesto de dos estudiantes.

Indiferentemente del resultado final, un prototipo completamente funcional y que realizara el trabajo para el cual se desarrolla la aplicación (la pesca real en mar abierto) supondría una mayor inversión económica, de la cual nos hubiese sido imposible hacernos cargo. Es por esto que este punto lo queremos dividir en dos partes separadas e independientes.

En una de ellas se expondrán los gastos reales que han supuesto el desarrollo del vehículo y la aplicación presentada (teniendo en cuenta ensayos y errores de selección de componentes). Por otro lado, se tratará de hacer lo más exhaustivamente posible la representación real del coste económico que supondría realizar el mismo proyecto con un presupuesto ilimitado.

9.1. Inversión Realizada

Las siguientes tablas muestran el desembolso realizado para el desarrollo e implementación del vehículo presentado. En ellas se desglosan todos los componentes comprados y finalmente se puede obtener el resultado final del presupuesto invertido.

Tabla 8.- Inversión realizada en electrónica y software

ELECTRÓNICA Y SOFTWARE			
Componente	***Cantidad***	***Precio por unidad (IVA incluido)***	***Precio total***
Placa de Desarrollo ATmega2560 MEGA	1	29,90 €	29,90 €
Sensor de Temperatura y Humedad digital DHT11	1	4,86 €	4,86 €
Módulo 2 Relés 12V	1	8,35 €	8,35 €
Acelerómetro de 3 ejes ADXL335	1	15,72 €	15,72 €
Sensor de Corriente 30A ACS712	1	5,99 €	5,99 €
Motor Driver Speed Controller MDD10A	1	18,48 €	18,48 €
Brújula HMC5883L Magnetómetro	1	11,90 €	11,90 €
GPS GY-GPS6MV2	1	18,87 €	18,87 €
Baliza 12V	1	23,70 €	23,70 €
Bocina 12V	1	9,59 €	9,59 €
Motor 6-12V 47W 4.7A 20400rpm	2	10,83 €	21,66 €
Batería SOLARX 8Ah - 12V	1	20,95 €	20,95 €
Placa de Topos	1	6,99 €	6,99 €
Regulador LM7812	3	0,34 €	1,02 €
Regulador LM7805	3	0,33 €	0,99 €

Regulador LM7809	1	0,49 €	0,49 €
Disipador	3	1,99 €	5,98 €
Condensador MKT 63V R.5	5	0,30 €	1,50 €
Condensador MKT 100K 100V	5	0,12 €	0,60 €
Tira 40 pines macho	2	0,83 €	1,67 €
Tira 40 pines acodados macho	1	1,73 €	1,73 €
Tira 40 pines hembra	2	0,99 €	1,98 €
60 pines de Tornillo	1	8,99 €	8,99 €
Funda cableado	8	0,19 €	1,55 €
Interruptor Basculante	1	0,83 €	0,83 €
Cable Rojo (2x0,5mm)	1	0,21 €	0,21 €
Cable Negro (2x0,5mm)	1	0,21 €	0,21 €
Cable Bicolor	1	1,70 €	1,70 €
Cableado Rígido	1	8,98 €	8,98 €
Código API Google Maps	1	1,00 €	1,00 €

TOTAL	**236,39 €**

Tabla 9.- Inversión realizada en el diseño mecánico

DISEÑO MECÁNICO			
Componente	***Cantidad***	***Precio por unidad (IVA incluido)***	***Precio total***
Placa Poliestireno Transparente	1	21,95 €	21,95 €
Resina de Poliéster 1kg (+ catalizador)	2	16,71 €	33,42 €
Fibra de Vidrio MAT-300 5m2	1	12,90 €	12,90 €
Poliestireno extruido (125x60x3)	1	2,49 €	2,49 €
Caja Estanca (153x110)	4	3,89 €	15,56 €
Caja Estanca (220x170x85)	1	4,25 €	4,25 €
Poliestireno Expandido	1	17,15 €	17,15 €
Discos de unión 18mm	9	1,62 €	14,58 €
Discos de unión 16mm	25	0,32 €	8,10 €
Pintura Blanca 750mL	1	12,50 €	12,50 €
Spray Pintura Naranja 400mL	1	10,95 €	10,95 €
Spray Pintura Blanca 400mL	1	10,95 €	10,95 €
Brocha plana	1	1,99 €	1,99 €
Funda termorretractil	1	2,10 €	2,10 €
Punteras	1	2,00 €	2,00 €
Conector Estanco	1	10,95 €	10,95 €
Escuadra 40x40	8	0,50 €	4,00 €
T Métrica 4x15	2	1,79 €	3,58 €
T Métrica 4x20	2	1,79 €	3,58 €
T Métrica 4x30	2	1,99 €	3,98 €

Escuadra 100	16	1,05 €	16,80 €
Arandelas	20	0,11 €	2,19 €
Tacos Metal Expansible 8x16mm	1	3,19 €	3,19 €
Silicona Neutra	2	3,59 €	7,18 €
Burlete 5.5Mx2-3MM	1	3,95 €	3,95 €
Cinta Reparación 25MMx50MM	1	2,99 €	2,99 €
Metacrilato precortado 50x100x4	1	15,99 €	15,99 €
Varillas	2	18,95 €	37,90 €
Acopladores	4	3,95 €	15,80 €
Adhesivo tuberías	1	2,29 €	2,29 €
Tapón Ciego	4	1,15 €	4,60 €
Manguitos PVC	2	0,90 €	1,80 €
Tubo Corrugado	1	4,95 €	4,95 €
Tubo PVC	1	2,19 €	2,19 €
Plancha PVC	1	5,99 €	5,99 €
Bridas	50	0,06 €	3,09 €
Cardán	2	8,85 €	17,70 €
Resina Epoxy	1	27,50 €	27,50 €
Hélices	2	3,70 €	7,40 €

TOTAL	**352,29 €**

Tabla 10.- Inversión realizada en pruebas y ensayos

PRUEBAS Y ENSAYOS			
Componente	***Cantidad***	***Precio por unidad (IVA incluido)***	***Precio total***
UART TTL Serial Cable PL2303TA	1	6,10 €	6,10 €
Arduino Mega 2560 R3	1	33,60 €	33,60 €
ELEGOO Mega2560 R3	1	13,99 €	13,99 €
Poliestireno Expandido 100mm DINA4	1	3,90 €	3,90 €
GPS GY-GPS6MV2	1	5,30 €	5,30 €
Brújula HMC5883L Magnetómetro	1	11,90 €	11,90 €
Giroscopio de 3 ejes	1	14,52 €	14,52 €
Acelerómetro de 3 ejes ADXL335	1	15,72 €	15,72 €
Sensor de Corriente 30A ACS712	4	3,83 €	15,32 €
Conector Alimentación 2.1 x 5.5 x 9.5mm	1	0,50 €	0,50 €
Resistencia Metálica 1 Ohm 50 W	1	3,66 €	3,66 €
Modulo 4 Relés 12V	1	14,12 €	14,12 €
Malla Desoldante	1	1,24 €	1,24 €

TOTAL	**139,87 €**

El presupuesto total invertido en la realización del proyecto se muestra en la siguiente tabla:

Tabla 11.- Inversión total realizada

PRESUPUESTO FINAL	**728,55 €**

9.2. Coste Real

Para la realización de este cálculo se ha optado por realizar pequeñas variaciones en los gastos (todos ellos contrastados) con tal de profesionalizar el resultado tratando siempre de mantener el espíritu de prototipo. Se ha tenido en cuenta el sistema total que no se ha llegado a realizar y que se puede ver en la figura 1.

Tabla 12.- Inversión hipotética necesaria en pruebas y ensayos

ELECTRÓNICA Y SOFTWARE			
Componente	***Cantidad***	***Precio por unidad (IVA incluido)***	***Precio total***
Placa de Desarrollo ATmega2560 MEGA	2	29,90 €	59,80 €
Sensor de Temperatura y Humedad digital RHT05	1	15,50 €	15,50 €
Módulo 2 Relés 12V	1	8,35 €	8,35 €
Acelerómetro y giroscopio de 3 ejes SEN-11028	1	26,96 €	26,96 €
Sensor de Corriente 30A SCT 013	1	5,80 €	5,80 €
Motor Driver Speed Controller	1	18,48 €	18,48 €
Brújula HMC5883L Magnetómetro	1	11,90 €	11,90 €
GPS GY-GPS6MV2	1	18,87 €	18,87 €
Baliza 12V	1	23,70 €	23,70 €
Bocina 12V	1	9,59 €	9,59 €
Motor 6-12V 47W 4.7A 20400rpm	2	10,83 €	21,66 €
Batería de Litio 18Ah - 12V	1	98,95 €	98,95 €
PCB 15x15cm	1	163,45 €	163,45 €
Regulador LM7812	3	0,34 €	1,02 €
Regulador LM7805	3	0,33 €	0,99 €
Regulador LM7809	1	0,49 €	0,49 €
Disipador	3	1,99 €	5,98 €
Condensador MKT 63V R.5	5	0,30 €	1,50 €
Condensador MKT 100K 100V	5	0,12 €	0,60 €
Interruptor Basculante	1	0,83 €	0,83 €
Cable Bicolor	1	1,70 €	1,70 €
Emisor Acústico V7DT [19]	1	1.500,00 €	1.500,00 €
Receptor Acústico VR2W [19]	1	250,00 €	250,00 €
Xbee	4	21,68 €	86,72 €

Código API Google Maps	1	1,00 €	1,00 €
LabVIEW Completo	1	3.451,00 €	3.451,00 €

TOTAL	**4.034,84 €**

Como se puede apreciar en la anterior tabla, el presupuesto invertido en electrónica y software varía notablemente debido al hecho de la inclusión del emisor y receptor acústico (imprescindible para la comunicación Nasa-Boya) y la licencia de compra de LabVIEW™ Completo, el cual es necesario para el control y registro de datos del vehículo.

También es cierto que no se ha tenido en cuenta la mano de obra de realización del proyecto completo, pero se podría estimar un año de trabajo de al menos un equipo de 3 personas con tal de realizar la totalidad del proyecto. Contando que el sueldo medio de un ingeniero en España es de 35.000 €, se le debería incrementar al valor anterior un presupuesto en mano de obra de 105.000 €.

Tabla 13.- Inversión hipotética necesaria en diseño mecánico

DISEÑO MECÁNICO			
Componente	***Cantidad***	***Precio por unidad (IVA incluido)***	***Precio total***
Flotadores en fibra de vidrio	2	85,00 €	170,00 €
Caja principal de fibra de vidrio	1	100,00 €	100,00 €
Varillas	2	18,95 €	37,90 €
Caja Estanca (153x110)	3	3,89 €	11,67 €
Acopladores	4	3,95 €	15,80 €
Pintura Blanca 750mL	1	12,50 €	12,50 €
Spray Pintura Naranja 400mL	1	10,95 €	10,95 €
Spray Pintura Blanca 400mL	1	10,95 €	10,95 €

TOTAL	**369,77 €**

El presupuesto hipotético total invertido en la realización del proyecto se muestra en la siguiente tabla:

Tabla 14.- Inversión hipotética total realizada

PRESUPUESTO FINAL	**6.103,70 €**

10. POSIBLES MEJORAS

Si bien es cierto que el resultado obtenido es más que satisfactorio a nuestro parecer, se trata de un prototipo el desarrollo del cual ha estado repleto de hándicaps. En primer lugar, y como se ha comentado numerosas veces a lo largo del proyecto, el presupuesto con el cual se ha contado ha sido realmente modesto, esto se traduce en la elección de componentes tanto mecánicos como eléctricos y electrónicos suficientes como para llevar a cabo el prototipo propuesto, pero que bajo las condiciones a las cuales se debería enfrentar realmente el vehículo no son los más indicados. Algunos de los componentes que se pueden agrupar dentro de este problema son:

- Batería: La batería escogida plomo-ácido es de 12V 8Ah, se trata de unas características suficientes para las pruebas de corto recorrido, pero no está preparada como para mantener en funcionamiento dos motores a media/alta velocidad durante largos periodos. Con un presupuesto ilimitado podríamos haber optado a baterías con mayor amperaje hora.
- Sensor de Temperatura: El sensor DHT11 tiene un rango de medición de 0ºC a 50ºC, lo cual indica que a temperaturas inferiores a 0 ºC (las cuales se podrían dar en el trampolín del catamarán en el Mar de Bering) dejaría de muestrear los datos de forma correcta. Una forma de solventar este problema sería con sensores del tipo RHT05, el cual consta con un rango operacional de -40ºC a 120ºC o incluso una sonda de temperatura tipo PT100.
- Flotadores: Como bien se explica en el apartado “3.1. Cascos”, los flotadores se han diseñado en SolidWorks® para posteriormente realizarlos en un núcleo de poliestireno expandido recubierto por fibra de vidrio, pero con un presupuesto holgado se podría haber realizado un pedido a medida de estos componentes para que se adecuaran a nuestras necesidades, lo cual hubiese supuesto un ahorro de tiempo y esfuerzo.
- Hélices: Las hélices que impulsan el vehículo son típicas en el modelaje y radiocontrol, pero no tienen una vida útil larga. Unas hélices de latón podrían haber solventado este inconveniente.

La inexperiencia en ciertos ámbitos tratados a lo largo del desarrollo del prototipo también nos ha jugado una mala pasada, lo cual se podría transcribir en la optimización del tiempo de un modo más eficiente, una selección más exhaustiva de los componentes utilizados, un código completamente depurado o una comunicación LabVIEW™ – ARDUINO® / ARDUINO® – LabVIEW™ más eficaz. De todos modos, este aspecto ha sido más notorio en unos puntos que en otros, y son dos los principales elementos los cuales han supuesto un reto por este motivo:

- Brújula: La brújula utilizada des de un inicio fue la HMC5883L, una brújula (en principio) de fácil y rápida programación gracias las librerías de esta familia de brújulas. Probablemente por desconocimiento del método de programación

adecuado o de configuración necesaria, la programación de la misma ha supuesto más un hándicap que una ventaja.

Con tal de solventar este problema se incluyeron las librerías de la familia más cercada de brújulas, las QMC588, mediante las cuales se solucionó parcialmente el problema, puesto que la simultaneidad de operación entre LabVIEW™ y ARDUINO® era interrumpida tras periodos de tiempo aleatorios y solventada al retirar la brújula.

- Flotadores: Los flotadores han sido modelados y recubiertos en fibra de vidrio a mano; el modelaje en polietileno expandido fue tedioso pero satisfactorio, obteniendo los resultados deseados en un principio, pero el desconocimiento respecto a la aplicación y el modelaje con fibra de vidrio ha dado como resultado unos flotadores asimétricos y con imperfecciones, que cumplen con las necesidades pero que no resultan completamente eficientes.
- Caja hermética: La caja hermética se ha realizado a conciencia y comprobando punto por punto la estanquidad de la misma, pero eso no quita que se ha tratado de un proceso nuevo y en su mayor medida experimental. El hecho de ser la primera ver en realizar un elemento que requiera de unas características sales como la hermeticidad, ha resultado en un continuo comprobar y arreglar las fugas y los imperfectos de la misma. La caja definitiva es completamente estanca, pero el proceso de este éxito ha sido largo y arduo.

Finalmente, y tratándose de un trabajo de fin de grado, hay una fecha límite sobre la cual entregar los resultados obtenidos mediante el uso de los conocimientos adquiridos a lo largo del periodo de educación en la facultad, es decir, el tiempo ha resultado ser el último gran obstáculo.

Desde el inicio de la realización del proyecto se han ido marcando objetivos plausibles en periodos de tiempo más o menos largos pero la incorrecta selección de componentes, el desconocimiento sobre ciertos puntos, el trabajo o motivos personales pueden suponer el aplazamiento de alguno de estos objetivos, por lo tanto, a lo largo del trabajo ha habido periodos en los cuales se ha trabajado contrarreloj para poder cumplir con las obligaciones necesarias respecto al mismo.

Trabajar bajo la presión del tiempo puede aumentar la motivación y las ganas de obtener resultados positivos de manera rápida, aumentar la concentración y reducir la sensación de cansancio, pero también puede actuar de forma completamente contraria. Es por esto que ha resultado ser un enemigo mental, al cual nos hemos debido de enfrentar constantemente y el cual en ciertas ocasiones nos ha ganado alguna batalla. Por lo tanto, con un lapso de tiempo más holgado, el proyecto podría haber resultado en algo de una mayor envergadura, más complejo y óptimo.

Al margen de lo anteriormente comentado en este apartado, y como se puede apreciar en la figura 1, el proyecto realizado no abarca la totalidad del conjunto, puesto que la comunicación Nasa-Boya no se ha realizado. Por lo tanto, la mayor de las mejoras sobre el proyecto habría sido el poder abarcar la totalidad de la complejidad del proyecto.

11.CONCLUSIÓN

La principal conclusión extraída de este proyecto es que se ha desarrollado el prototipo de un vehículo marino de superficie para el apoyo a embarcaciones de pesca con nasas cumpliendo en la medida de lo posible con la normativa actual y los requisitos del usuario medio de la aplicación realizada.

Al margen de ésta, las conclusiones extraídas a lo largo del desarrollo son la siguientes:

De todos los métodos de comunicación entre LabVIEW™ y ARDUINO® sobre los cuales se ha trabajado, la comunicación por bits es la más beneficiosa. Pese a ser un método rápido y fiable, el método de comunicación por LIFA *Toolkit* restringe su uso siendo prácticamente imposible añadir nuevas librerías, por otro lado, el método de comunicación por *strings* es más fácil de programar, pero añade el hándicap del *delay* o el retraso en la comunicación. Pese a tener un método de programación más complejo, al enviar la información por bits encriptados y desencriptarlos mediante LabVIEW™ la comunicación por bits se convierte en una manera robusta de enviar información acabando con los *delays* y siendo completamente versátil en cuanto a la integración de nuevos dispositivos.

La selección de los componentes apropiados en innegablemente importante para el funcionamiento idóneo del complejo. Durante la realización del proyecto se ha barajado continuamente la posibilidad de cambiar componentes tanto electrónicos como mecánicos, muchos de estos cambios no han sido posibles debido al modesto presupuesto disponible para el desarrollo. Componentes mejores tales como sensores o baterías e incluso una estructura diferente hubiesen hecho menos sinuoso el avance hacia resultados satisfactorios.

La concienciación medioambiental es muy importante para el futuro. Estamos viviendo una crisis medioambiental como no ha habido otra, por lo tanto, es de gran importancia el uso concienzudo de los materiales para la realización de cualquier proyecto, y más aún, tratándose de un vehículo marino. Una biosfera marina tan grande como son los océanos y mares los cuales albergan innumerable cantidad de seres vivos deben ser respetados, y desde nuestro punto de vista consideramos importante el uso responsable de los elementos escogidos en este o cualquier otro proyecto.

La primera lluvia de ideas sobre este proyecto se realizó un año antes de la entrega. Se ha tratado de un proyecto ambicioso, teniendo en cuenta que tan solo se contaba con dos desarrolladores, y complejo hasta el punto de haberse convertido en una rutina dedicar una gran cantidad de horas al día o contar con los fines de semana para poder seguir con el desarrollo non-stop. Pero para el cumplimiento de los tiempos y las necesidades establecidas creemos importantes algunos aspectos. El primero de todo es la dedicación, un trabajo no se realiza solo y no por ser una obligación debe ser un

suplicio, por lo tanto, la positividad y la proactividad han sido indispensables para la realización del proyecto. Por otro lado, la exigencia, si el trabajo puede hacerse mejor y en menos tiempo el resultado beneficia doblemente, y esta actitud se ha visto recompensada constantemente. Finalmente, y no menos importante, el trabajo en equipo. Afortunadamente (por temas de proximidad), el desarrollo del proyecto se ha podido dividir de tal manera que los desarrolladores no han necesitado constantemente la ayuda del otro para poder avanzar en su labor, sino que cada uno ha podido ir mejorando sus resultados individualmente, pero no siempre son buenas noticias, errar es humano, y a lo largo de este trabajo se han cometido errores, el trabajo en equipo, según nuestro punto de vista, implica el apoyo a cada una de las partes, la entrega de la ayuda para cada situación, y la exigencia propia y del grupo, con tal de motivar mutuamente y mejorar resultados.

Para acabar, nos gustaría volver a agradecer a los profesores, compañeros, amigos, familia y parejas, la formación, el apoyo y la ayuda incondicional que nos han ofrecido a lo largo de todos estos años y especialmente durante la realización de este proyecto, ya que, sin ellos no hubiese sido posible.

12. BIBLIOGRAFÍA

[1] Ley de la náutica de recreo. (2019). Retrieved from https://www.fomento.gob.es/marina-mercante/nautica-de-recreo

[2] Superbathy, dron catamarán batimétrico. (2019). Retrieved from http://www.heliceo.com/es/produits-pour-geometres/superbathy-el-dron-catamaran-batimetrico/

[3] Coast Guard Compass (2019). Retrieved from https://coastguard.dodlive.mil/2011/10/coast-guard-prepares-for-red-king-crab-season/

[4] Wikipedia. (2019). Retrieved from https://www.wikipedia.org

[5] Normativa luces de navegación. (2019). Retireved from http://como.nauticpedia.com/luces-de-navegacion-obligatorias-segun-embarcacion/

[6] Cajas estancas en mercado. (2019). Retrieved from https://www.automation24.es/caja-de-bornes-rittal-kl-1504-510-400-x-200-x120?previewPriceListId=1&refID=adwords_shopping_ES&gclid=Cj0KCQjwoKzsBRC5ARIsAITcwXFuG5SZcLL5jjgpKZsZvk3tmf1MIkOmvMqstvrnY5ITPtQWz9H5nKkaAvHpEALw_wcB

[7] Datasheets de componentes electrónicos. (2019). Retrieved from https://www.alldatasheet.es

[8] Manual y guía de Arduino. (2019). Retieved from https://arduino.cl

[9] Digital Marketplace. (2019) Retrieved from https://www.cytron.io/

[10] Electronic distributor. (2019) Retrieved from https://fadisel.com/

[11] National Instruments. (2019). Retrieved from http://www.ni.com/es-es.html

[12] Ruiz Gutiérrez, J. (2012). Utilización de Labview para la Visualización y Control de la Plataforma Open Hardware Arduino.

[13] National Instruments. (2019). Guía de elementos básicos para iniciar proyectos con NI y myRIO.

[14] Dunbanin, M. Grinham, A. Udy, J. (2009) An Autonomous Surface Vehicle for Water Quality Monitoring. Sydney.

[15] Samaniego, F. Sanchis, J. García-Nieto, S. Simarro, R. (2019). Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): A Dynamic Path Planning Algorithm for UAVs

[16] Comparativa baterías. (2019). Retrieved from https://www.cambioenergetico.com/blog/comparacion-baterias-de-plomo-acido-y-bateria-de-litio-para-energia-solar/

[17] Benavides, M. Gélvez, R. (2015). Análisis ambiental del ciclo de vida de las baterías plomo-ácido. Bogotá.

[18] Pietropaoli, A. Basti, F. Veiga-Álvarez, A. Maqueda-Blasco, J. Manejo de la fibra de vidrio en entorno laboral, potenciales efectos sobre la salud y medidas de control. (2015). Madrid.

[19]Study behavior and migrations of animals. (2019). Retrieved from https://www.vemco.com/

[20]Simulador y diseño de circuitos EasyEda. (2019). Retrieved from https://easyeda.com

[21]Biblioteca digital de la UPC: Bibliotécnica. (2019). Retrieved from https://bibliotecnica.upc.edu/

[22]Aguilar, W. Costa-Castellón, R. Angulo, C. Molina, L. (2014). Control Autónomo de Cuadricopteros para Seguimiento de Trayectorias. Barcelona.

13.ANEXOS

En este punto se encuentran los elementos necesarios para complementar la información global del proyecto, en orden de aparición encontramos:

- Manual de instrucciones: PDF con el manual de instrucciones en tres idiomas (castellano, catalán e inglés) de las dos aplicaciones de LabVIEW™ desarrolladas para el funcionamiento del proyecto.
- Esquemático de la aplicación: PDF con el esquema eléctrico de la aplicación.
- Planos de piezas: PDF con los planos de todas las piezas desarrolladas por nosotros totalmente acotadas.
- Código de ARDUINO®: El código cargado en la placa de ARDUINO® desarrollado íntegramente por nosotros (el diagrama de bloques de LabVIEW™ que recoge este código se puede ver en las figuras 105 y 107).
- Hojas de datos: PDF con los *datasheets* requeridos de todos los elementos y módulos utilizados en la realización del proyecto.
    - ARDUINO® MEGA 2560.
    - Batería SOLARX® 12V 8Ah.
    - Reguladores LM7805, LM7809 y LM7812
    - Sensor de temperatura DHT11.
    - Acelerómetro ADXL335.
    - Brújula HMC5883L.
    - Sensor de corriente ACS712.
    - GPS NEO 6M.
    - Driver MDD10A.
    - Motores C-6044.

2019

MANUAL DE USUARIO

APLICACIONES PARA CONTROL Y REGISTRO DE VEHÍCULO
PARA APOYO A LA PESCA DE FONDO MARINO

ANTONIO GONZALEZ
ÁLVARO MARTÍN

Contenido

1. INTRODUCCIÓN 2

2. APLICACIÓN PARA EL CONTROL DEL VEHÍCULO DE APOYO 3

2.1. Configuración inicial 4

2.2. Indicadores de temperatura 5

2.3. Indicadores de posición 6

2.4. Modos de funcionamiento 8

2.4.1. MODO AUTÓNOMO 8

2.4.2. MODO MANUAL 9

2.5. Recoger Nasas 10

3. APLICACIÓN PARA EL REGISTRO DE DATOS DEL VEHÍCULO DE APOYO 11

3.1. Panel de Control 12

3.2. Guardar Registros 12

3.3. Gráficas 14

4. INICIALIZACIÓN 15

4.1. *Secuencia de Inicio* 15

1. INTRODUCCIÓN

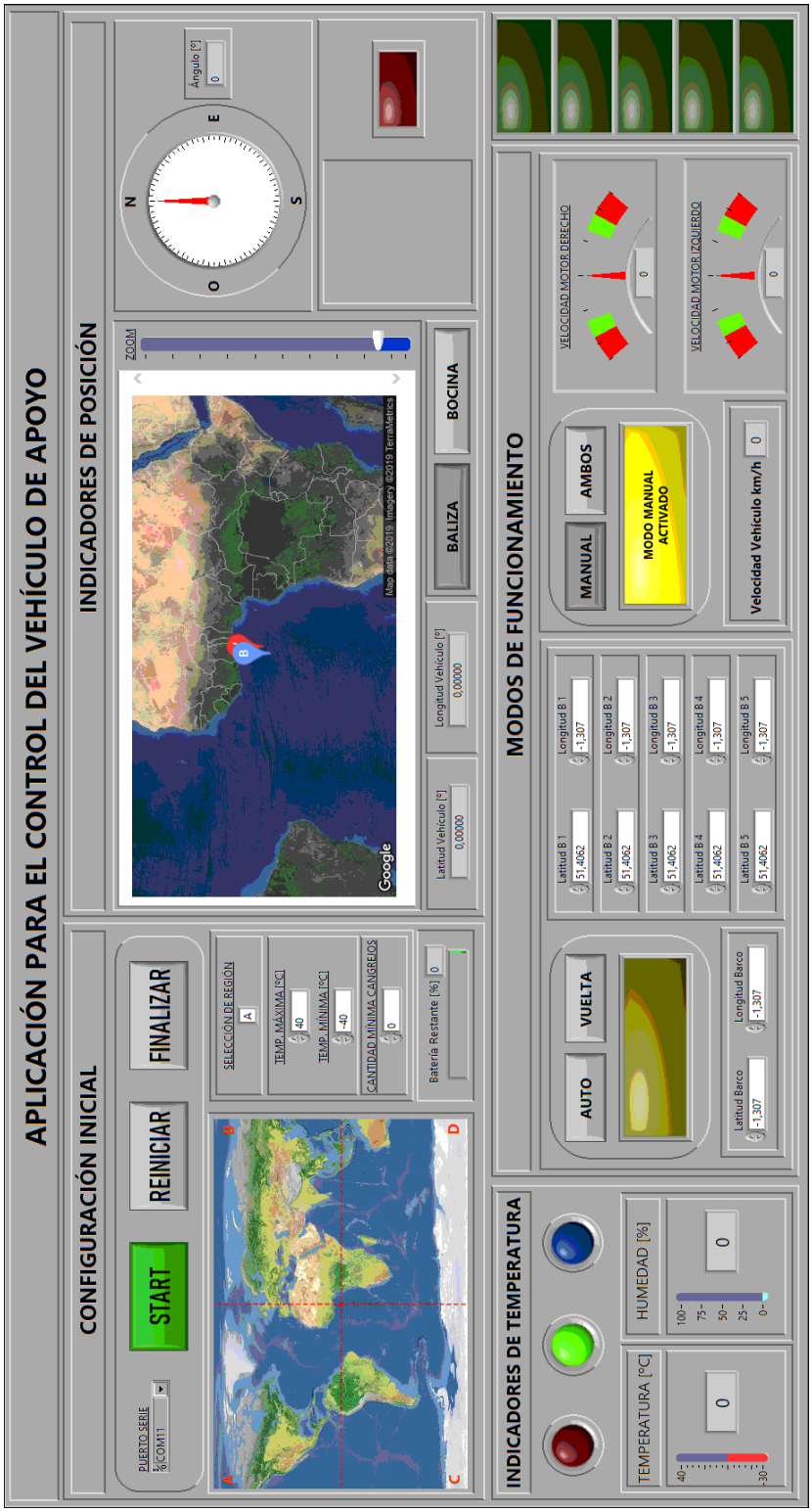
El propósito de esta aplicación es el control y registro de datos del vehículo de apoyo para la pesca de cangrejos y está diseñada para cumplir únicamente con esta misión.

A continuación, se dará explicación a cada punto de la aplicación y se darán los conocimientos necesarios para poder llevar a cabo un correcto funcionamiento del conjunto vehículo-aplicación.

La aplicación se divide en dos partes, una para el control del vehículo de apoyo y otra para el registro de los datos que está tomando. El usuario puede escoger entre tener abierta una, la otra o las dos a la vez.

2. APLICACIÓN PARA EL CONTROL DEL VEHÍCULO DE APOYO

Esta es la aplicación encargada de controlar el vehículo y la cual le otorga al usuario información visual altamente intuitiva con tal de simplificar la comprensión y el manejo de la misma.



2.1. *Configuración inicial*

En este punto el usuario indica las condiciones las cuales debe seguir la aplicación en función del conocimiento propio

1. **PUERTO SERIE**: Antes de iniciar la aplicación el usuario debe seleccionar el puerto serie en el cual está conectado el dispositivo de comunicación inalámbrica. Se debe buscar una opción del tipo "**COM X**" siendo "X" un número entero.
2. **START, REINICIAR Y FINALIZAR**: Estos tres botones controlan el funcionamiento general del programa. Con el botón **START** se inicia el programa y con el **FINALIZAR** se finaliza. Si en cualquier momento el usuario añade nuevos datos o quiere reiniciar por cualquier motivo, debe pulsar el botón **REINICIAR**.

3. **SELECCIÓN DE REGIÓN**: En función de la zona del mapa mostrado a la izquierda de esta opción donde se halle el vehículo y las nasas, el usuario debe introducir la letra A, B, C o D. Se recomienda no utilizar este vehículo en los cruces de regiones ya que puede llegar a no comprender bien las instrucciones.
4. **TEMP. MÁXIMA y TEMP. MÍNIMA**: El vehículo cuenta con un sensor de temperatura y humedad. La temperatura puede variar considerablemente en función de la zona o la estación del año en la cual se realice la pesca. Por lo tanto, se le otorga al usuario la selección de temperatura máxima y mínima.
5. **CANTIDAD MÍNIMA DE CANGREJOS**: El usuario indica cual es la cantidad mínima de cangrejos por nasa a la cual quiere que se le avise para ir a recogerlos.
6. **BATERÍA RESTANTE**: Indicador de autonomía restante en tanto por ciento (%).

2.2. Indicadores de temperatura

1. **INDICADORES DE TEMPERATURA**: Estos indicadores se encienden siguiente las ordenes de:

   LED ROJO – Se ha sobrepasado la temperatura máxima

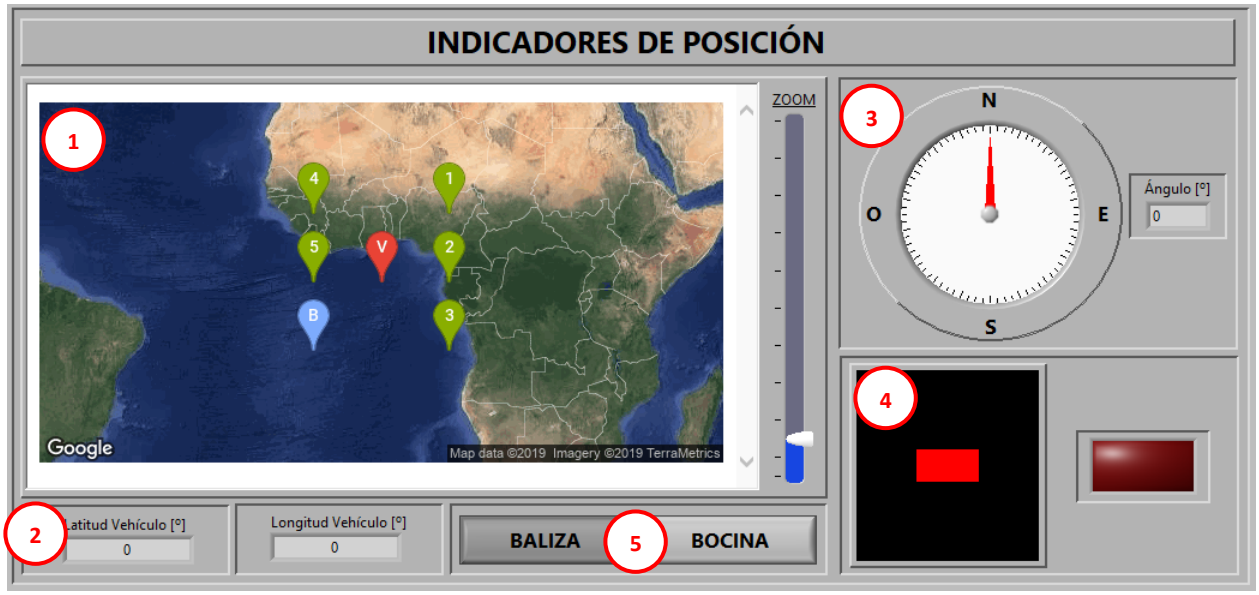
   LED VERDE – La situación se encuentra estable (temperatura entre máxima y mínima)

   LED AZUL – Se ha sobrepasado la temperatura mínima

2. **TEMPERATURA**: Indicador numérico y en barra de la temperatura en grados centígrados (ºC).

3. **HUMEDAD**: Indicador numérico y en barra de la humedad en tanto por ciento (%).

2.3. Indicadores de posición

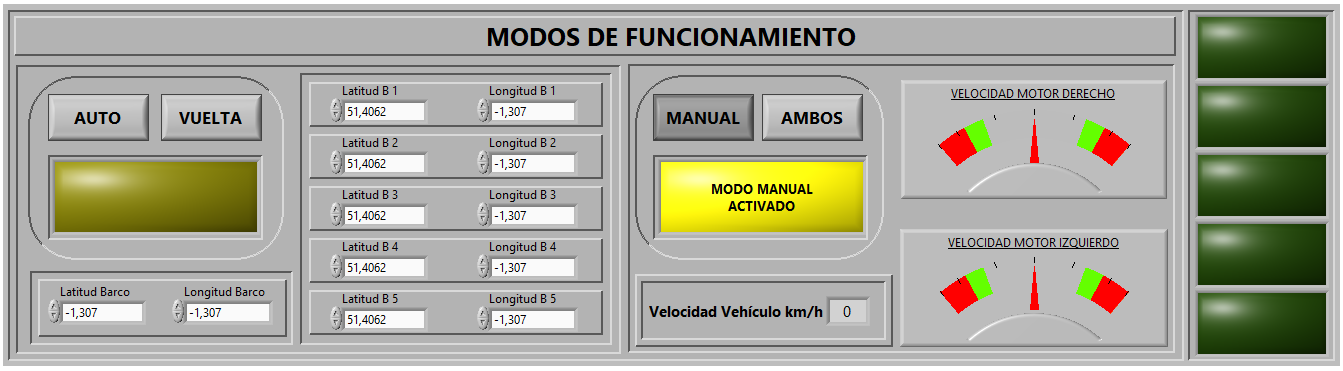


1. **MAPA**: En este mapa se muestra en tiempo real y con un retraso de 900ms la posición del vehículo de apoyo, de las nasas y de la embarcación tripulada desde la cual se está controlando el vehículo. El código de lectura de las banderas es el siguiente:

Símbolo	Significado
	Vehículo de apoyo
	Nasa 1
	Nasa 2
	Nasa 3
	Nasa 4
	Nasa 5
	Embarcación tripulada

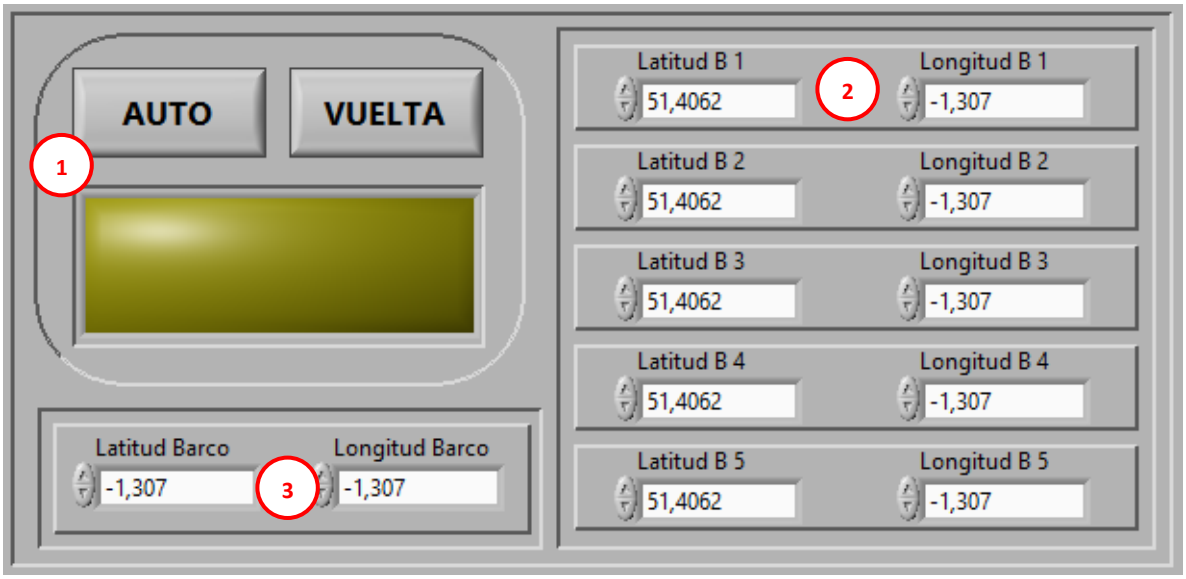
2. **LATITUD Y LONGITUD DEL VEHÍCULO**: En este apartado se pueden ver las coordenadas de posición del vehículo en tiempo real.
3. **BRÚJULA**: Indicador de orientación visual y en grados (º) del vehículo de apoyo.
4. **ACELERÓMETRO**: Este indicador muestra la inclinación del vehículo en tiempo real. En caso de volcado, el piloto de la derecha en encenderá mostrando el texto de "**VEHÍCULO VOLCADO**".
5. **BALIZA Y BOCINA**: En el caso de estar activados, la baliza iluminará constantemente el vehículo de apoyo y la bocina soltará un bocinazo cada 2 minutos.

2.4. *Modos de funcionamiento*



Se divide en dos modos de funcionamiento: Autónomo y Manual. Para activar uno es necesario previamente haber desactivado el otro. En caso de error por parte del usuario y mantener los dos modos activados simultáneamente, el vehículo por defecto detectará el modo Manual.

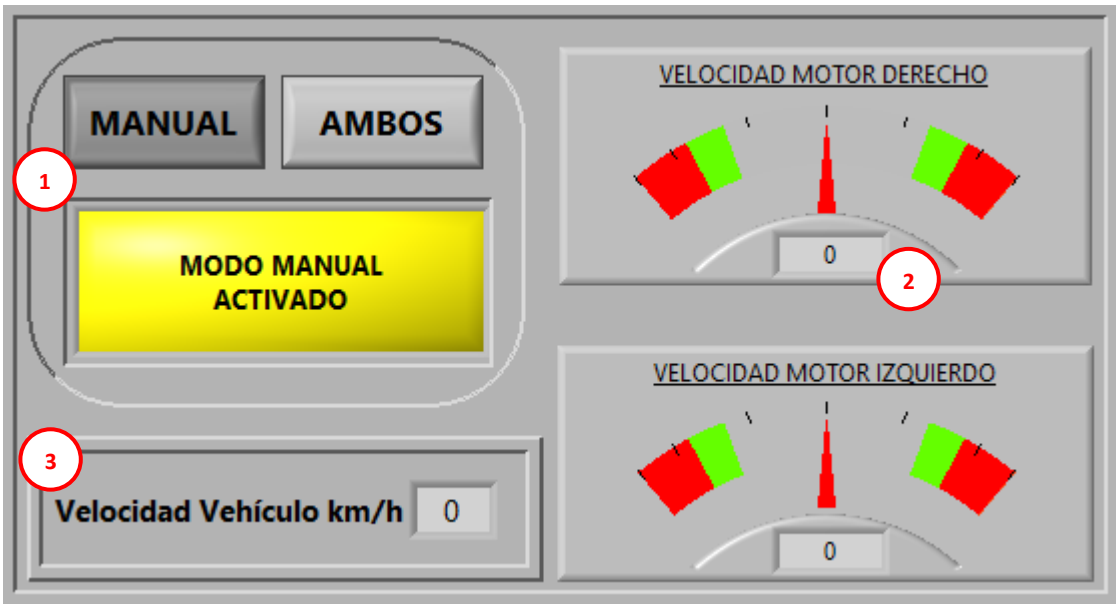
2.4.1. MODO AUTÓNOMO



1. **AUTÓNOMO Y VUELTA**: Para activar el modo autónomo es necesario pulsar el botón "AUTÓNOMO". Mientras este se encuentre pulsado, el indicador amarillo mostrará el texto "**MODO AUTÓNOMO ACTIVADO**". En el caso de que se quiera interrumpir el modo automático con tal de hacer volver el vehículo hasta la embarcación tripulada, en cualquier momento se puede pulsar el botón "VUELTA"

2. **LATITUD Y LONGITUD NASAS**: Es necesario introducir las coordenadas de las nasas para que el vehículo pueda dirigirse hacia ellas con una precisión de hasta 0,0001º. El recorrido del vehículo siempre será en orden de la 1 a la 5.
3. **LATITUD Y LONGITUD BARCO**: Cuando el vehículo acabe la ruta por todas las nasas, volverá a la embarcación principal, para ello es necesario situar geográficamente la embarcación.

2.4.2. MODO MANUAL



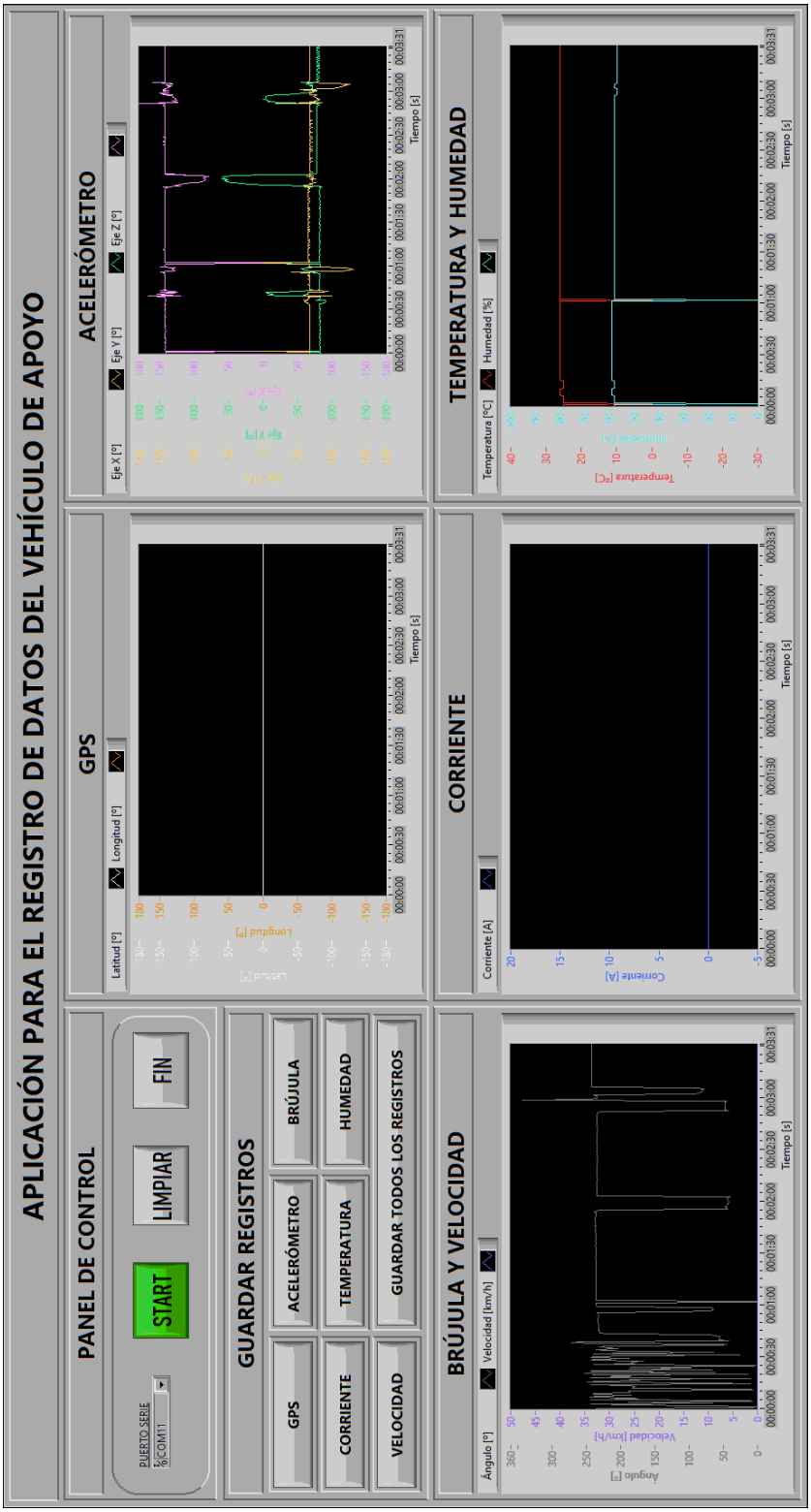
1. **MANUAL**: Para activar el modo manual es necesario pulsar el botón "MANUAL". Mientras este se encuentre pulsado, el indicador amarillo mostrará el texto "**MODO MANUAL ACTIVADO**".
2. **VELOCIDAD MOTOR DERECHO Y VELOCIDAD MOTOR IZQUIERDO**: Estas palancas sirven para controlar la velocidad de los motores, siendo la derecha avanzar y la izquierda retroceder. Bajo cada una de éstas, se muestra en tanto por ciento (%) la velocidad a la que se está haciendo girar el motor.
3. **VELOCIDAD VEHÍCULO**: Indicador de la velocidad del vehículo.

2.5. Recoger Nasas

Los indicadores verdes de la esquina inferior derecha de la interfaz se encenderán mostrando el texto "**RECOGER NASA X**" (siendo X un entero del 1 al 5) en función de la nasa que haya superado la capacidad mínima de cangrejos introducida por el usuario.

3. APLICACIÓN PARA EL REGISTRO DE DATOS DEL VEHÍCULO DE APOYO

Esta es la aplicación encargada de registrar los datos del vehículo y guardar los datos de la misma.



3.1. *Panel de Control*

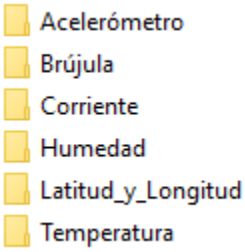
1. **PUERTO SERIE**: Antes de iniciar la aplicación el usuario debe seleccionar el puerto serie en el cual está conectado el dispositivo de comunicación inalámbrica. Se debe buscar una opción del tipo "**COM X**" siendo "X" un número entero.
2. **START, LIMPIAR Y FIN**: Estos tres botones controlan el funcionamiento general del programa. Con el botón **START** se inicia el programa y con el **FIN** se finaliza. Si en cualquier momento el usuario quiere reiniciar el muestreo o limpiar los datos obtenidos por cualquier motivo, debe pulsar el botón **LIMPIAR**.

3.2. *Guardar Registros*

1. **REGISTRO SIMPLE**: Si se quiere guardar el registro de una o varias variables simultáneamente se debe pulsar el botón de la variable que se desea guardar, mientras el botón se encuentre pulsado el se irá tomando y guardando el registro de los datos.
2. **GUARDAR TODOS LOS REGISTROS**: Si se quieren guardar todos los registros simultáneamente se puede pulsar el botón de "**GUARDAR TODOS LOS REGISTROS**" con tal de simplificar la operación.

Al iniciar el registro de datos se creará una carpeta en el escritorio del PC como la siguiente:

Dentro de la cual se generarán sub-carpetas para la o las variables que se hayan deseado guardar:

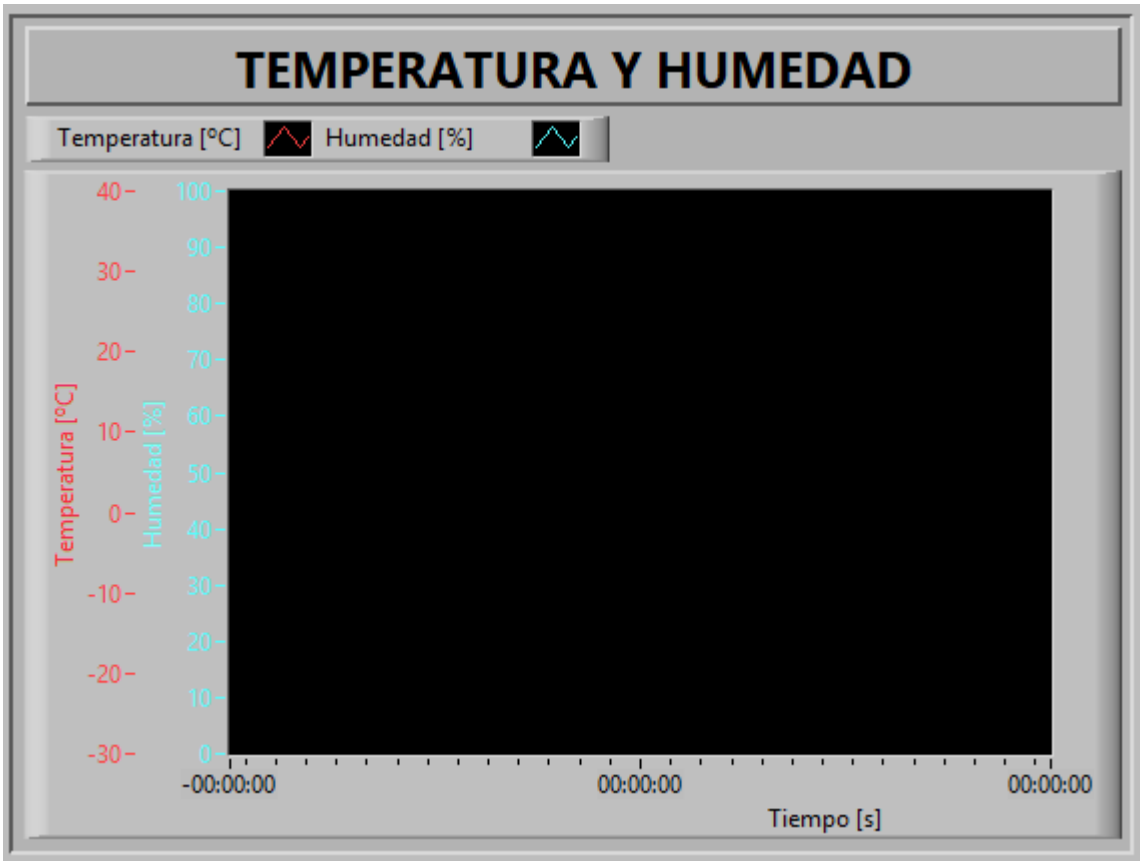


Cada carpeta contiene uno o varias hojas de cálculo Microsoft Excel™ en función de las veces que se haya deseado recoger y guardar datos. Estos documentos Excel™ se guardan según las siguientes especificaciones:

Time	Latitud	Time 1	Longitud
16/08/2019 13:47:16,028	0	16/08/2019 13:47:16,028	0
16/08/2019 13:47:18,750	0	16/08/2019 13:47:18,750	0
16/08/2019 13:47:19,049	0	16/08/2019 13:47:19,049	0
16/08/2019 13:47:19,349	0	16/08/2019 13:47:19,349	0

La aplicación se encarga de registrar la fecha y hora exacta de recogida de datos y la divide en los datos muestreados (en el caso anterior, al tratarse del GPS, es necesario registrar dos variables, la de Latitud y la de Longitud).

3.3. Gráficas



El resto de la interfaz se divide en 5 gráficas las cuales ofrecen un apoyo visual al usuario. Estas son:

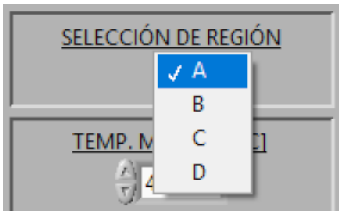
a. **GPS**: Registra las coordenadas de **LATITUD** y **LONGITUD** del vehículo de apoyo.
b. **ACELERÓMETRO**: Registra la **INCLINACIÓN** de los ejes **X**, **Y** y **Z** del vehículo de apoyo.
c. **BRÚJULA Y VELOCIDAD**: Registra la **ORIENTACIÓN** y la **VELOCIDAD** del vehículo de apoyo.
d. **CORRIENTE**: Registra la **CORRIENTE** consumida por el vehículo de apoyo.
e. **TEMPERATURA Y HUMEDAD**: Registra la **TEMPERATURA** y la **HUMEDAD** a la cual se encuentra el vehículo de apoyo.

4. INICIALIZACIÓN

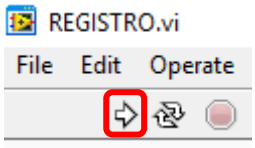
4.1. *Secuencia de Inicio*

Para el correcto funcionamiento de la aplicación es necesario seguir la secuencia siguiente:

1. **ABRIR LA APLICACIÓN**: Para abrir la o las aplicaciones que se deseen es necesario hacer doble clic sobre los iconos (en el caso de querer abrir la aplicación de control) o (para abrir la aplicación de registro).
2. **FINALIZAR EL PROGRAMA ANTERIOR**: Para no llevar a confusión a la aplicación (que arranca por defecto), es necesario finalizar la aplicación pulsando sobre los botones FINALIZAR (vea 2.1.2.) o FIN (vea 3.1.2.).
3. **INTRODUCIR DATOS**: Los tipos de datos que se pueden introducir en esta aplicación son dos:
    a. *Datos Numéricos*: son la mayoría y son los datos que corresponden a valores numéricos tales como Longitud y Latitud (vea 2.4.2.) o temperaturas máximas y mínimas (vea 2.1.4.).
    b. *Datos Alfanuméricos*: son los datos que corresponden al Puertos Serie (vea 2.1.1. o 3.1.1.) y Selección de Región (vea 2.1.3.) los cuales abren desplegables con facilitando el dato a ingresar.



4. **ARRANCAR EL PROGRAMA**: Para iniciar la aplicación siempre que se pulsan los botones de FINALIZAR (vea 2.1.2.) o FIN (vea 3.1.2.) es necesario pulsar la flecha blanca situada en la esquina superior izquierda del programa:



5. **INICIAR LA APLICACIÓN**: Finalmente para iniciar la aplicación es necesario pulsar el botón **START** (vea 2.1.2 o 3.1.2.)

2019

MANUAL D'USUARI

APLICACIONS PER AL CONTROL I REGISTRE DE VEHICLE PER AL SUPORT A LA PESCA DE FONS MARÍ

ANTONIO GONZALEZ
ÁLVARO MARTÍN

Contingut

1. INTRODUCCIÓ 2

2. APLICACIÓ PER AL CONTROL DEL VEHICLE DE SUPORT 3

2.1. Configuració inicial 4

2.2. Indicadors de temperatura 5

2.3. Indicadors de posició 6

2.4. Modes de funcionament 8

2.4.1. MODE AUTÒNOM 8

2.4.2. MODE MANUAL 9

2.5. Recollir Nanses 10

3. APLICACIÓ PER AL EL REGISTRE DE DADES DEL VEHICLE DE SUPORT 11

3.1. Panel de Control 12

3.2. Guardar Registres 12

3.3. Gràfiques 14

4. INICIALIZACIÓ 15

4.1. *Seqüencia d'Inici* 15

1. INTRODUCCIÓ

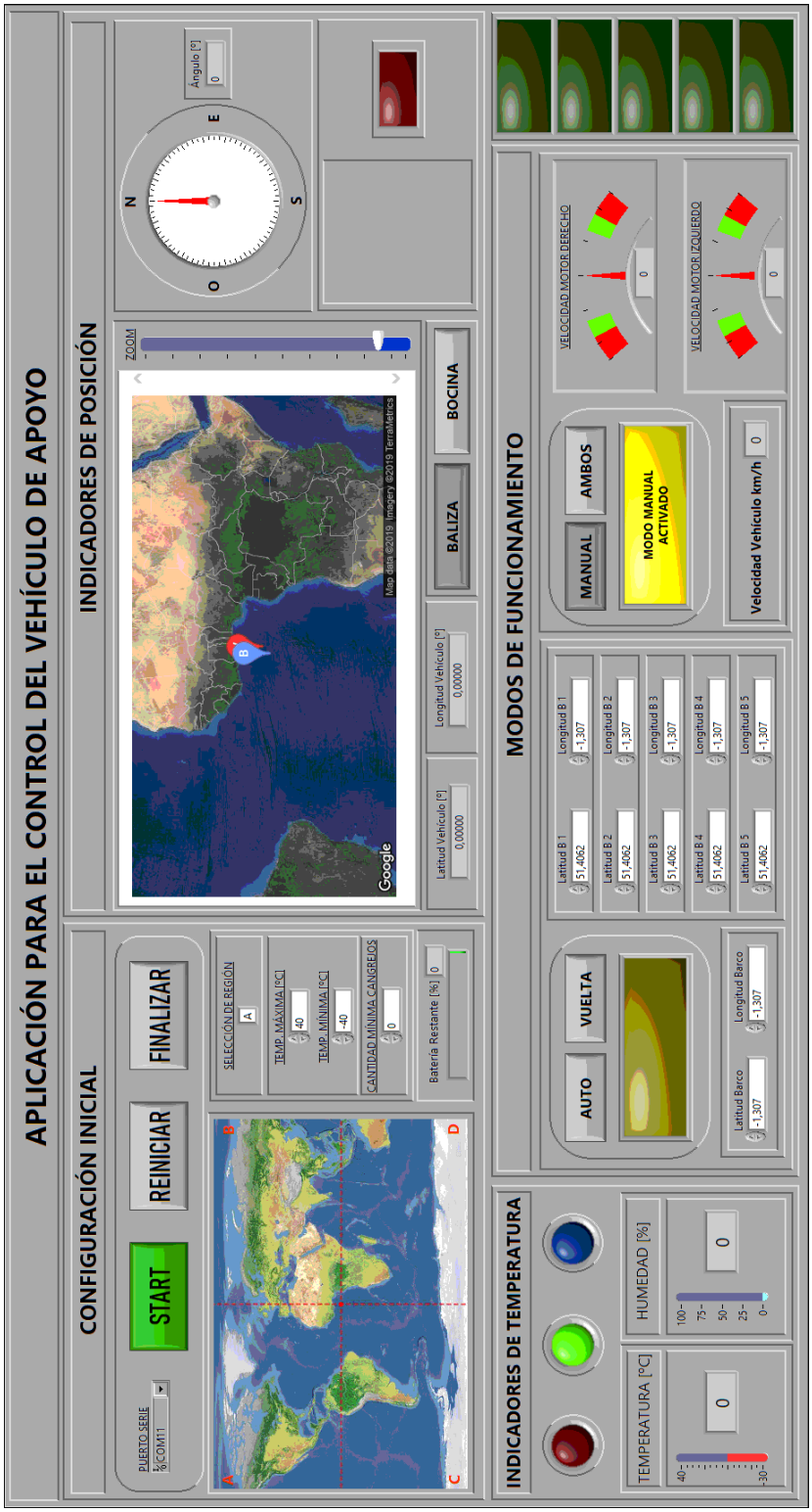
El propòsit d'aquesta aplicació és el control i registre de dades del vehicle de suport per a la pesca de crancs i està dissenyada per complir únicament amb aquesta missió.

A continuació, es donarà explicació a cada punt de l'aplicació i es donaran els coneixements necessaris per poder dur a terme un correcte funcionament del conjunt vehicle-aplicació.

L'aplicació es divideix en dues parts, una per al control del vehicle de suport i una altra per al registre de les dades que està prenent. L'usuari pot escollir entre tenir oberta una, l'altra o les dues alhora.

2. APLICACIÓ PER AL CONTROL DEL VEHICLE DE SUPORT

Aquesta és l'aplicació encarregada de controlar el vehicle i la qual li atorga a l'usuari informació visual altament intuïtiva per tal de simplificar la comprensió i el maneig de la mateixa.



2.1. *Configuració inicial*

En aquest punt l'usuari indica les condicions les quals ha de seguir l'aplicació en funció del coneixement propi

1. **PORT SERIE**: Abans d'iniciar l'aplicació l'usuari ha de seleccionar el port sèrie en el qual està connectat el dispositiu de comunicació sense fils. S'ha de buscar una opció del tipus "**COM X**" sent "X" un nombre sencer.
2. **START, REINICIAR Y FINALITZAR**: Aquests tres botons controlen el funcionament general del programa. Amb el botó **START** s'inicia el programa i amb el **FINALITZAR** es finalitza. Si en qualsevol moment l'usuari afegeix noves dades o vol reiniciar per qualsevol motiu, s'ha de polsar el botó **REINICIAR**.
3. **SELECCIÓ DE REGIÓ**: En funció de la zona del mapa mostrat a l'esquerra d'aquesta opció on es trobi el vehicle i les nanses, l'usuari ha d'introduir la

lletra A, B, C o D. Es recomana no utilitzar aquest vehicle a les cruïlles de regions ja que pot arribar a no comprendre bé les instruccions.

4. **TEMP. MÀXIMA y TEMP. MÍNIMA**: El vehicle compta amb un sensor de temperatura i humitat. La temperatura pot variar considerablement en funció de la zona o l'estació de l'any en la qual es realitzi la pesca. Per tant, se li atorga a l'usuari la selecció de temperatura màxima i mínima.
5. **CUANTITAT MÍNIMA DE CRANCS**: L'usuari indica quina és la quantitat mínima de crancs per nansa a la qual vol que se li avisi per anar a recollir-los.
6. **BATERIA RESTANT**: Indicador d'autonomia restant en tant per cent (%).

2.2. Indicadors de temperatura

1. **INDICADORS DE TEMPERATURA**: Aquests indicadors s'encenen seguint les ordres de:

   LED VERMELL – S'ha sobrepassat la temperatura màxima

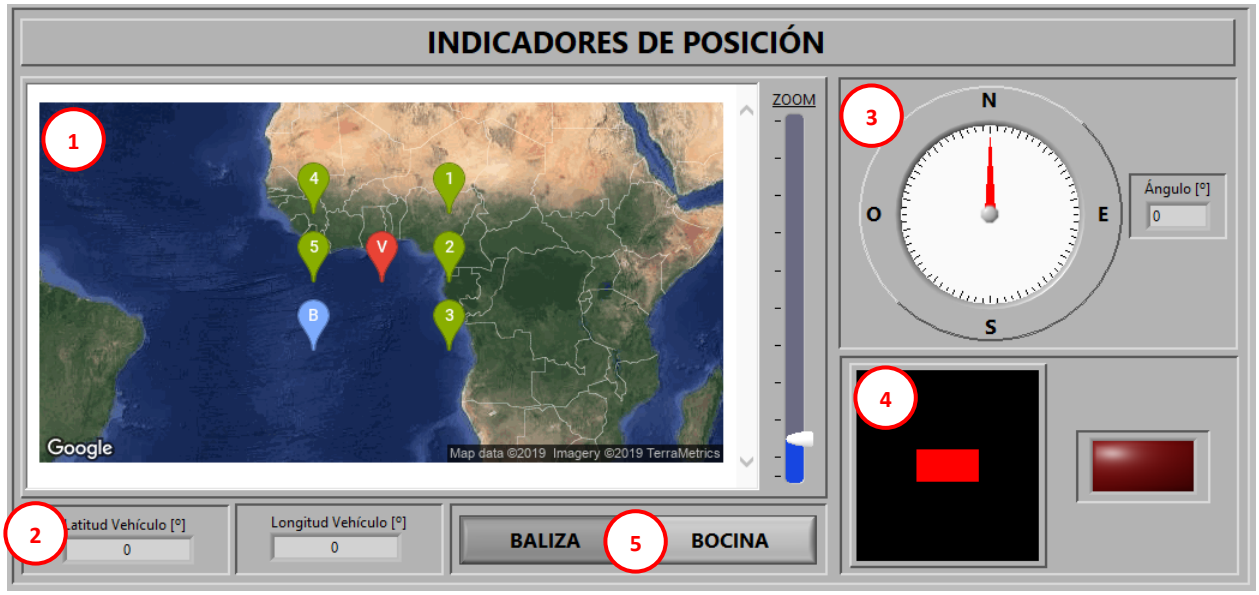
   LED VERD – La situació es troba estable (temperatura entre màxima y mínima)

   LED BLAU – S'ha sobrepassat la temperatura mínima

2. **TEMPERATURA**: Indicador numèric i en barra de la temperatura en graus centígrads (ºC).

3. **HUMITAT**: Indicador numèric y en barra de la humitat en tant per cent (%).

2.3. Indicadors de posició

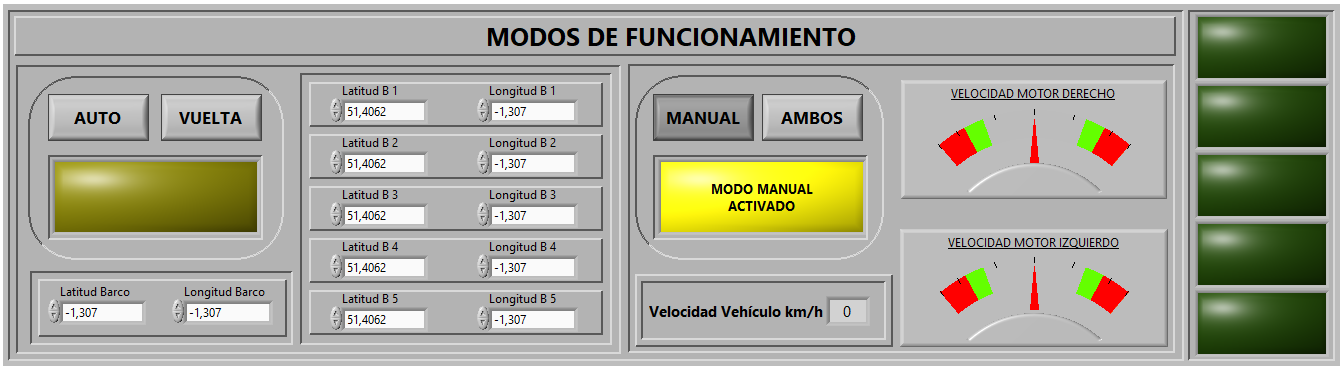


1. **MAPA**: En aquest mapa es mostra en temps real i amb un retard de 1 segon la posició del vehicle de suport, de les nanses i de l'embarcació tripulada des de la qual s'està controlant el vehicle. El codi de lectura de les banderes és el següent:

Símbolo	Significado
	Vehicle de suport
	Nansa 1
	Nansa 2
	Nansa 3
	Nansa 4
	Nansa 5
	Embarcació tripulada

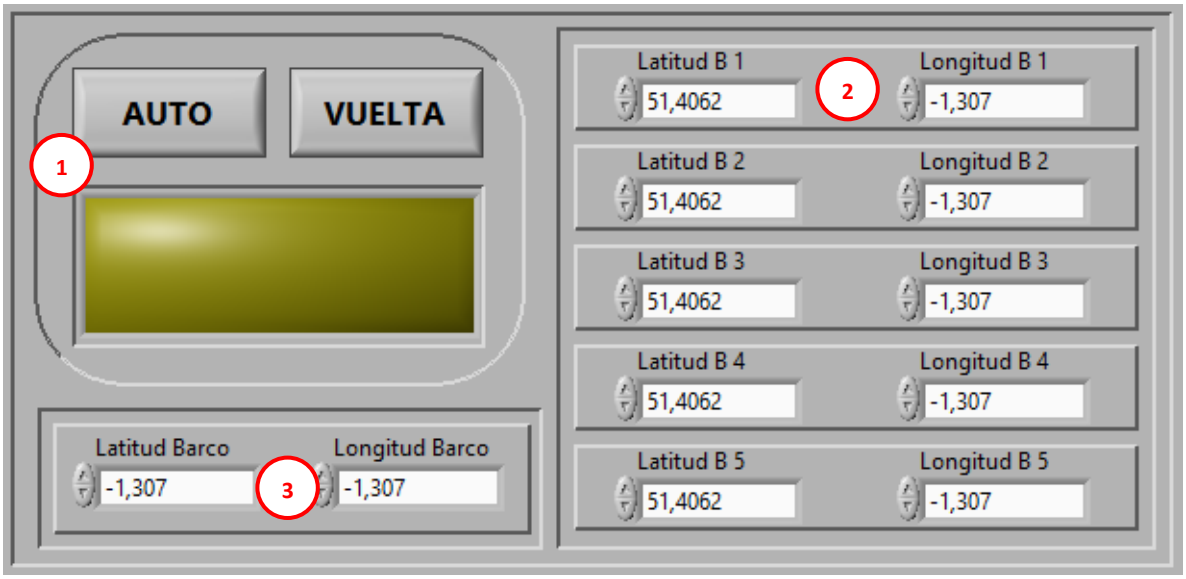
2. **LATITUD Y LONGITUD DEL VEHICLE**: En aquest apartat es poden veure les coordenades de posició del vehicle en temps real.
3. **BRÚIXOLA**: Indicador d'orientació visual i en graus (º) del vehicle de suport.
4. **ACELERÒMETRE**: Aquest indicador mostra la inclinació del vehicle en temps real. En cas de bolcat, el pilot de la dreta en encendrà mostrant el text de "**VEHICLE BOLCAT**".
5. **BALISA I BOTZINA:** En el cas d'estar activats, la balisa il·luminarà constantment el vehicle de suport i la botzina deixarà anar 1 toc de botzina cada 2 minuts.

2.4. *Modes de funcionament*



Es divideix en dos modes de funcionament: Autònom i Manual. Per activar un és necessari prèviament haver desactivat l'altre. En cas d'error per part de l'usuari i mantenir les dues maneres activats simultàniament, el vehicle per defecte detectarà la manera Manual.

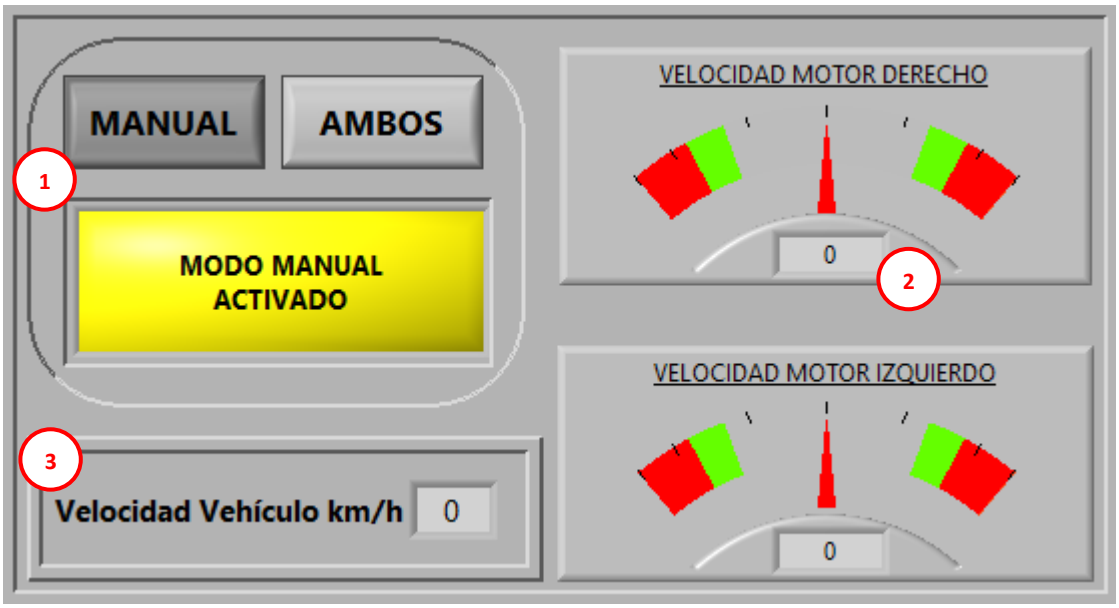
2.4.1. MODE AUTÒNOM



1. **AUTÒNOMO Y TORNADA**: Per activar el mode autònom és necessari prémer el botó "AUTÒNOM". Mentre aquest es trobi premut, l'indicador groc mostrarà el text "**MODE AUTÒNOM ACTIVAT**". En el cas que es vulgui interrompre la manera automàtica per tal de fer tornar el vehicle fins l'embarcació tripulada, en qualsevol moment es pot prémer el botó "**TORNADA**".

2. **LATITUD Y LONGITUD NANSES**: Cal introduir les coordenades de les nanses perquè el vehicle pugui dirigir cap a elles amb una precisió de fins a 0,0001º. El recorregut del vehicle sempre serà en ordre de la 1 a la 5.
3. **LATITUD Y LONGITUD VAIXELL**: Quan el vehicle acabi la ruta per totes les nanses, tornarà a l'embarcació principal, per a això cal situar geogràficament l'embarcació.

2.4.2. MODE MANUAL



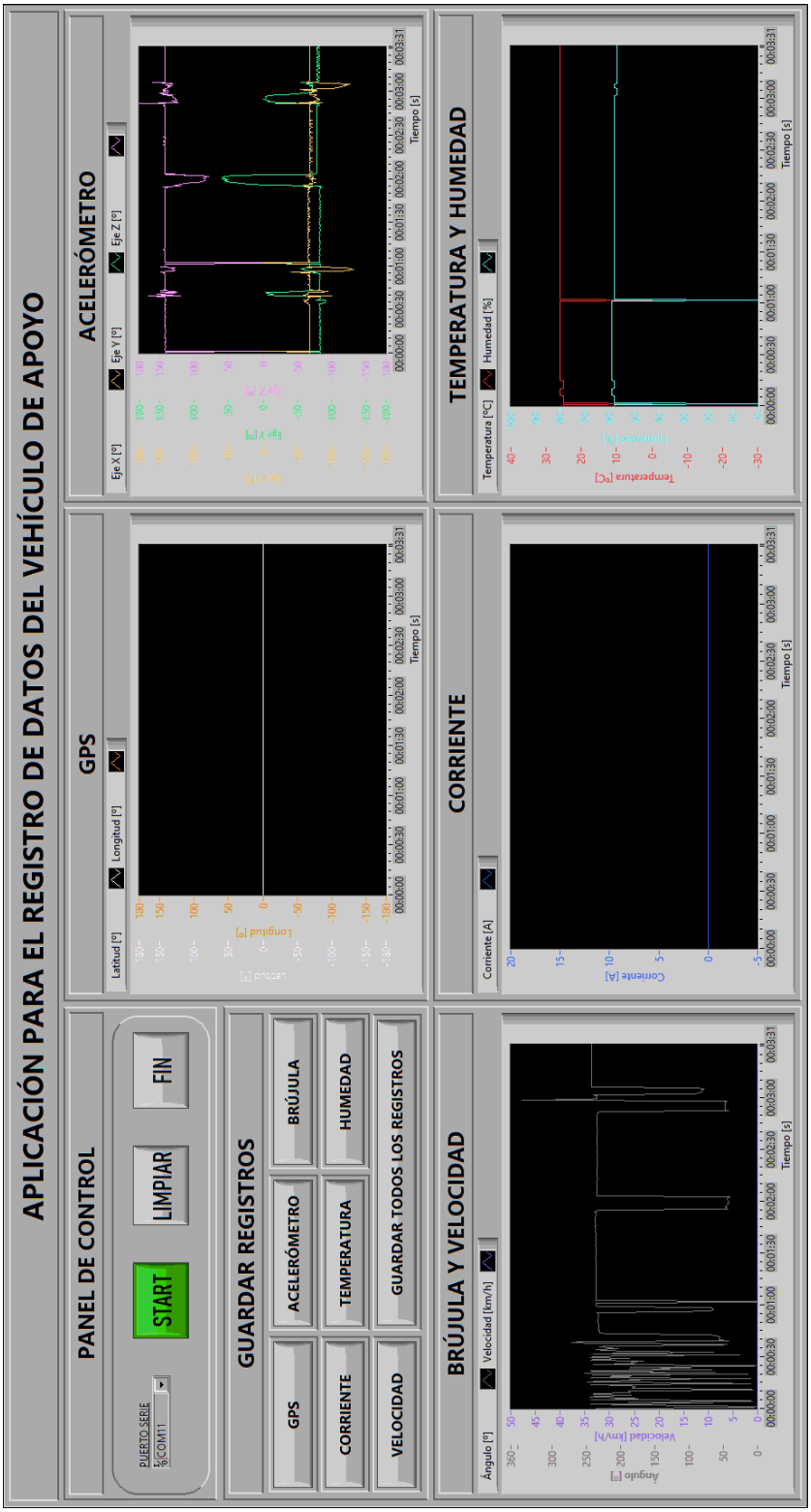
1. **MANUAL**: Per activar el mode manual és necessari prémer el botó "MANUAL". Mentre aquest es trobi premut, l'indicador groc mostrarà el text "**MODE MANUAL ACTIVAT**".
2. **VELOCITAT MOTOR DRET Y VELOCIDAD MOTOR ESQUERRE**: Aquestes palanques serveixen per controlar la velocitat dels motors, sent la dreta avançar i l'esquerra retrocedir. Sota cadascuna d'aquestes, es mostra en tant per cent (%) la velocitat a la qual s'està fent girar el motor.
3. **VELOCITAT VEHICLE**: Indicador de la velocitat del vehicle.

2.5. Recollir Nanses

Els indicadors verds de la cantonada inferior dreta de la interfície s'encendran mostrant el text "**RECOLLIR NASA X**" (sent X un sencer de l'1 al 5) en funció de la nansa que hagi superat la capacitat mínima de crancs introduïda per l'usuari.

3. APLICACIÓ PER AL EL REGISTRE DE DADES DEL VEHICLE DE SUPORT

Aquesta és l'aplicació encarregada de registrar les dades del vehicle i guardar les dades de la mateixa.



3.1. *Panel de Control*

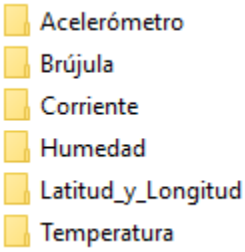
1. **PORT SERIE**: Abans d'iniciar l'aplicació l'usuari ha de seleccionar el port sèrie en el qual està connectat el dispositiu de comunicació sense fils. S'ha de buscar una opció del tipus "**COM X**" sent "X" un nombre sencer.
2. **START, NETEJAR Y FI**: Aquests tres botons controlen el funcionament general del programa. Amb el botó **START** s'inicia el programa i amb el **FI** es finalitza. Si en qualsevol moment l'usuari vol reiniciar el mostreig o netejar les dades obtingudes per qualsevol motiu, s'ha de polsar el botó de **NETEJAR**.

3.2. *Guardar Registres*

1. **REGISTRE SIMPLE**: Si es vol guardar el registre d'una o diverses variables simultàniament s'ha de prémer el botó de la variable que es vol guardar, mentre el botó es trobi premut el s'anirà prenent i guardant el registre de les dades.
2. **GUARDAR TOTS ELS REGISTRES**: Si es volen guardar tots els registres simultàniament es pot prémer el botó de "**GUARDAR TOTS ELS REGISTRES**" per tal de simplificar l'operació.

En iniciar el registre de dades es crearà una carpeta a l'escriptori del PC com la següent:

Dins de la qual es generaran sub-carpetes per a la o les variables que s'hagin desitjat guardar:

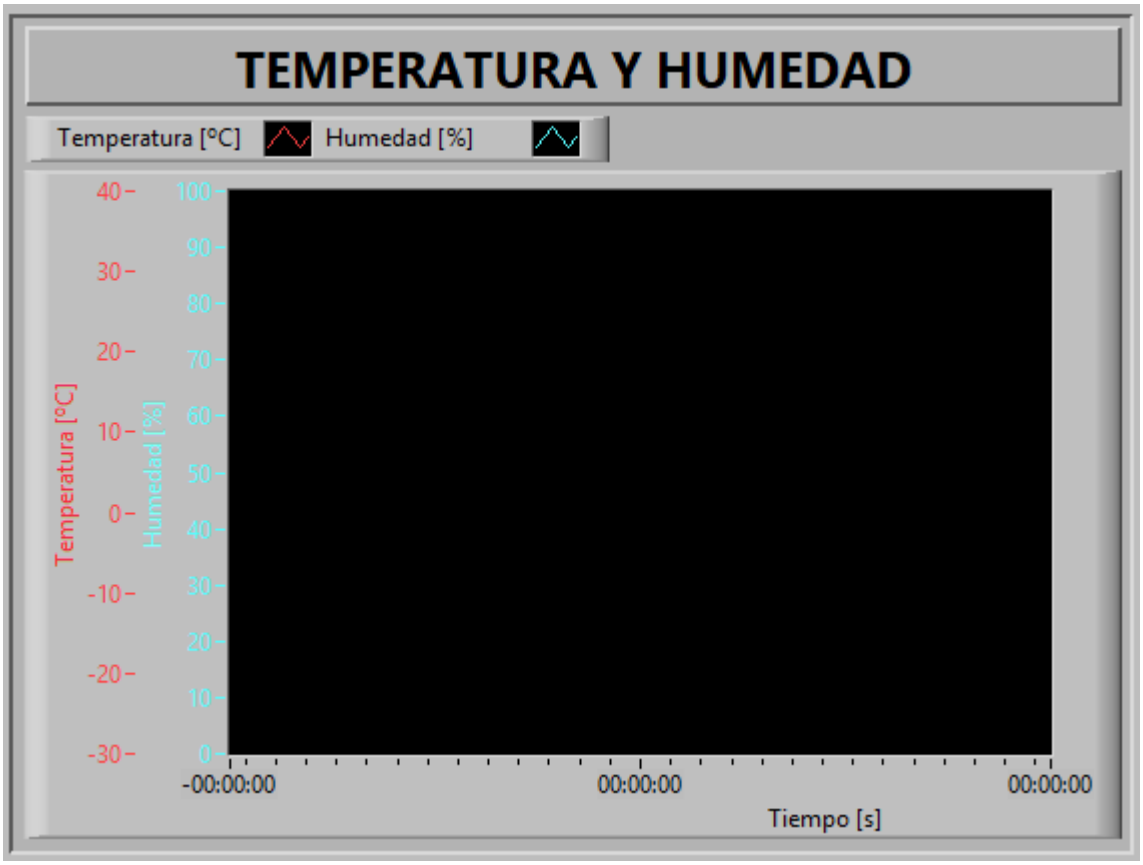


Cada carpeta conté un o diversos fulls de càlcul Microsoft Excel™ en funció de les vegades que s'hagi desitjat recollir i guardar dades. Aquests documents Excel™ es guarden segons les següents especificacions:

Time	Latitud	Time 1	Longitud
16/08/2019 13:47:16,028	0	16/08/2019 13:47:16,028	0
16/08/2019 13:47:18,750	0	16/08/2019 13:47:18,750	0
16/08/2019 13:47:19,049	0	16/08/2019 13:47:19,049	0
16/08/2019 13:47:19,349	0	16/08/2019 13:47:19,349	0

L'aplicació s'encarrega de registrar la data i hora exacta de recollida de dades i la divideix en les dades mostrejats (en el cas anterior, en tractar del GPS, cal registrar dues variables, la de Latitud i la de Longitud).

3.3. Gràfiques



La resta de la interfície es divideix en 5 gràfiques les quals ofereixen un suport visual a l'usuari. Aquestes són:

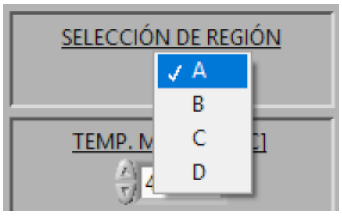
a. **GPS**: Registra les coordenades de **LATITUD** y **LONGITUD** del vehicle de suport.
b. **ACELERÒMETRE**: Registra la **INCLINACIÓ** dels eixos **X**, **Y** y **Z** del vehicle de suport.
c. **BRÚIXOLA Y VELOCITAT**: Registra la **ORIENTACIÓ** i la **VELOCITAT** del vehicle de suport.
d. **CORRENT**: Registra la **CORRENT** consumida pel vehicle de suport.
e. **TEMPERATURA Y HUMITAT:** Registra la **TEMPERATURA** y la **HUMITAT** a la qual es troba el vehicle de suport.

4. INICIALIZACIÓ

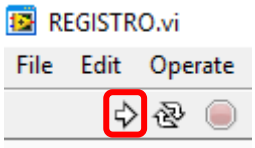
4.1. *Seqüencia d'Inici*

Per al correcte funcionament de l'aplicació és necessari seguir la seqüència següent:

1. **OBRIR LA APLICACIÓ**: Per obrir la o les aplicacions que es desitgin cal fer doble clic sobre les icones (en el cas de voler obrir l'aplicació de control) o (per obrir l'aplicació de registre).
2. **FINALITZAR EL PROGRAMA ANTERIOR**: Per no portar a confusió a l'aplicació (que arrenca per defecte), cal sortir de l'aplicació fent clic sobre els botons FINALITZAR (vegi 2.1.2.) O FI (vegi 3.1.2.).
3. **INTRODUIR DADES**: Els tipus de dades que es poden introduir en aquesta aplicació són dos:
    a. *Dades Numèriques*: són la majoria i són les dades que corresponen a valors numèrics com ara Longitud i Latitud (vegi 2.4.2.) o temperatures màximes i mínimes (vegi 2.1.4.).
    b. *Dades Alfanumèriques*: són les dades que corresponen als Ports Sèrie (vegi 2.1.1. o 3.1.1.) i Selecció de Regió (vegi 2.1.3.) els quals obren desplegables amb facilitant la dada a ingressar.



4. **ARRENCAR EL PROGRAMA**: Per iniciar l'aplicació sempre que es premen els botons de FINALITZAR (vegi 2.1.2.) O FI (vegi 3.1.2.) Cal prémer la fletxa blanca situada a la cantonada superior esquerra del programa:



5. **INICIAR LA APLICACIÓ**: Finalment per iniciar l'aplicació cal prémer el botó START (vegi 2.1.2 o 3.1.2.).

2019

USER MANUAL

APPLICATIONS FOR CONTROL AND REGISTRATION OF VEHICLE TO SUPPORT FISHING OF MARINE FUND

ANTONIO GONZALEZ
ÁLVARO MARTÍN

Content

1. INTRODUCTION 2

2. APPLICATION FOR CONTROL OF SUPPORT VEHICLE 3

2.1. Initial Setup 4

2.2. Temperature Indicators 5

2.3. Position indicators 6

2.4. Operating Modes 8

2.4.1. AUTONOMOUS MODE 8

2.4.2. MANUAL MODE 9

2.5. Pot Pick Up 10

3. APPLICATION FOR DATA RECORDING OF SUPPORT VEHICLE 11

3.1. Control Panel 12

3.2. Save Records 12

3.3. Plots 14

4. INITIALIZATION 15

4.1. *Start Sequence* 15

1. INTRODUCTION

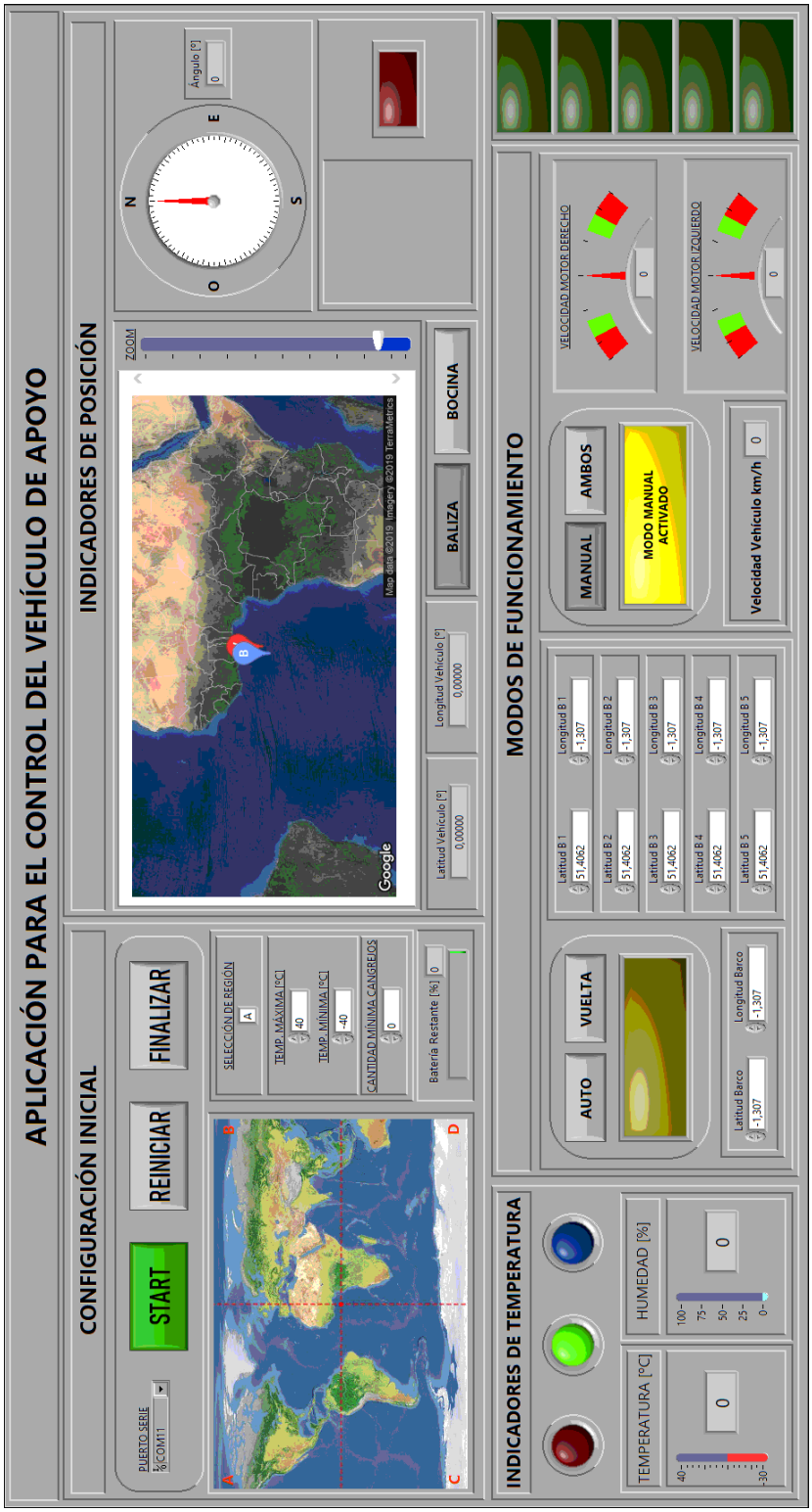
The purpose of this application is the control and data recording of the support vehicle for crab fishing and is designed to fulfil this mission only.

Next, an explanation will be given to each point of the application and the necessary knowledge will be given to be able to carry out a correct operation of the vehicle-application set.

The application is divided into two parts, one for the control of the support vehicle and another for the registration of the data you are taking. The user can choose between having one, the other or both open at the same time.

2. APPLICATION FOR CONTROL OF SUPPORT VEHICLE

This is the application in charge of controlling the vehicle and which gives the user highly intuitive visual information in order to simplify the understanding and handling of the vehicle.



2.1. *Initial Setup*

At this point the user indicates the conditions which the application must follow based on their own knowledge.

1. **SERIAL PORT**: Before starting the application, the user must select the serial port on which the wireless communication device is connected. An option of type "**COM X**" should be searched for, "X" being an integer.
2. **START, RESTART AND FINISH**: These three buttons control the overall operation of the program. With the **START** button the program is started and with the **FINISH** it ends. If at any time the user adds new data or wants to restart for any reason, he must press the **RESET** button.
3. **REGION SELECTION**: Depending on the area of the map shown to the left of this option where the vehicle is located, the user must enter the letter A, B, C

or D. It is recommended not to use this vehicle at the crossroads of regions since You may not understand the instructions well.

4. **MAXIMUM TEMP. and MINIMUM TEMP.**: The vehicle has a temperature and humidity sensor. The temperature can vary considerably depending on the area or season of the year in which the fishing takes place. Therefore, the user is given the maximum and minimum temperature selection.
5. **MINUMUM AMOUNT OF CRABS**: The user indicates the minimum amount of crabs per pot to which he wants to be notified to pick them up.
6. **REMAINING BATTERY**: Indicator of remaining autonomy as a percentage (%).

2.2. Temperature Indicators

1. **TEMPERATURE INDICATORS**: These indicators light following the orders of:

   RED LED – The maximum temperature has been exceeded

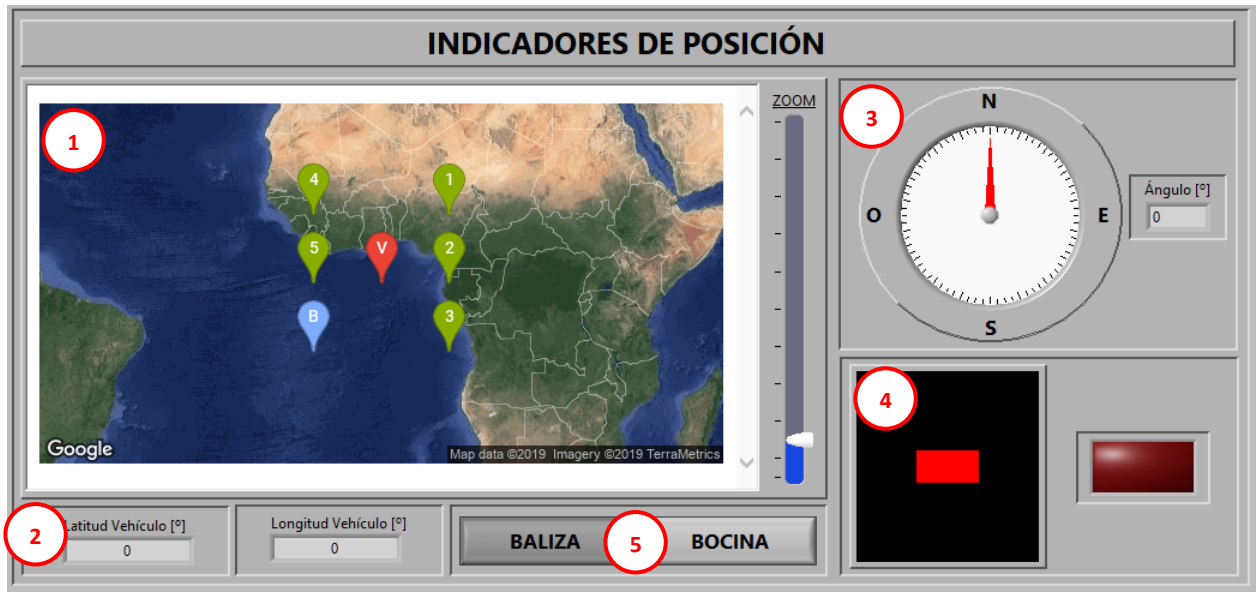
   GREEN LED – The situation is stable (temperature between maximum and minimum)

   BLUE LED – The minimum temperature has been exceeded

2. **TEMPERATURE**: Numerical and bar temperature indicator in degrees Celsius (ºC).

3. **HUMIDITY**: Numerical and bar moisture indicator in percent (%).

2.3. Position indicators

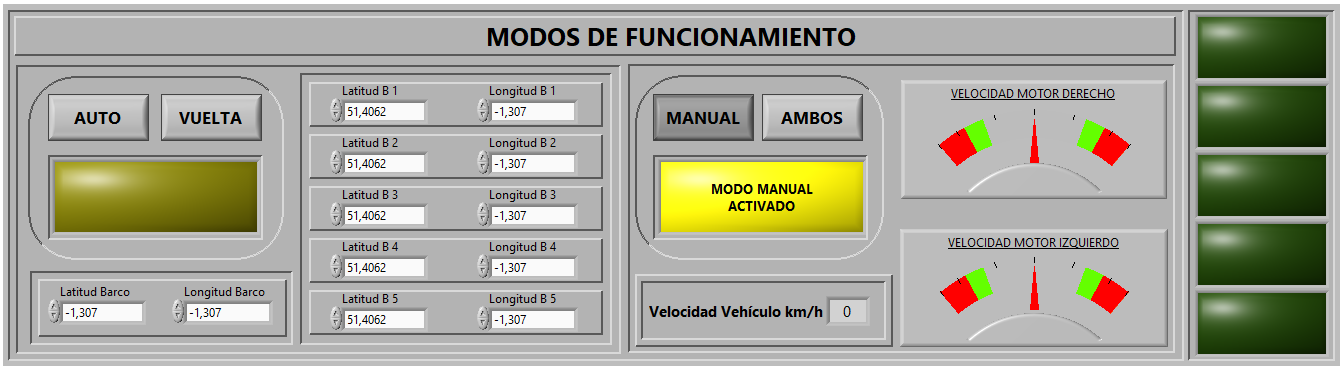


1. **MAP**: This map shows in real time and with a 1 second delay the position of the support vehicle, the pots and the manned vessel from which the vehicle is being controlled. The reading code of the flags is as follows:

Símbolo	Significado
	Support Vehicle
	Pot 1
	Pot 2
	Pot 3
	Pot 4
	Pot 5
	Manned boat

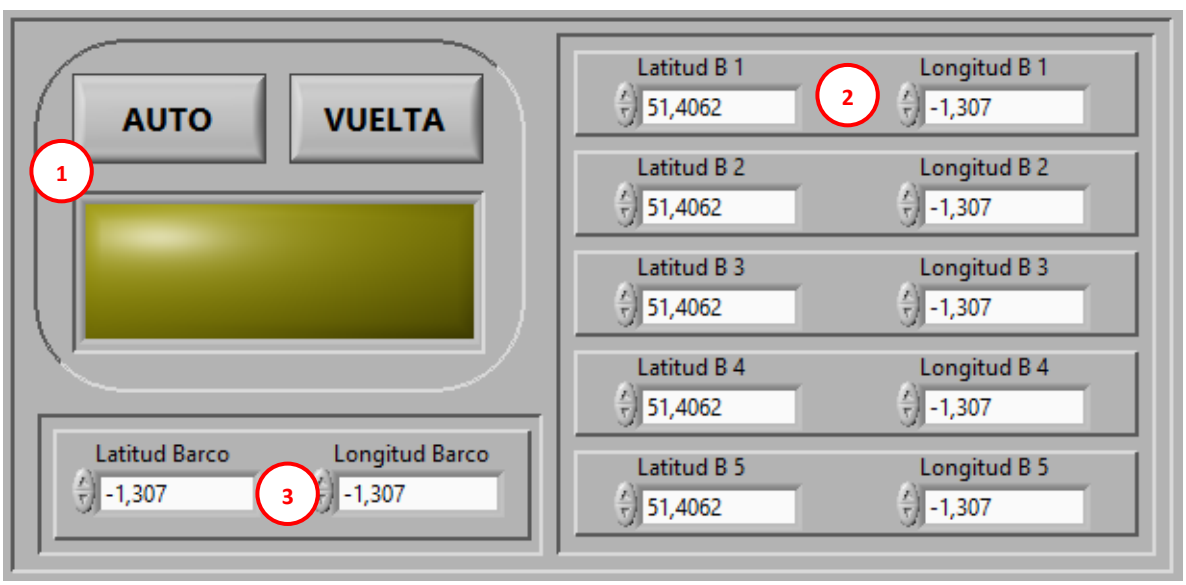
2. **LATITUDE AND LONGITUDE OF THE VEHICLE**: In this section you can see the position coordinates of the vehicle in real time.
3. **COMPASS**: Visual orientation indicator and in degrees (º) of the support vehicle.
4. **ACCELEROMETER**: This indicator shows the inclination of the vehicle in real time. In the event of a dump, the pilot on the right will turn on showing the text of “VEHICLE TURNED OVER”.
5. **BEACON AND HORN**: In case of being activated, the beacon will constantly illuminate the support vehicle and the horn will release a horn every 2 minutes.

2.4. *Operating Modes*



It is divided into two modes of operation: Autonomous and Manual. To activate one, it is necessary to have previously deactivated the other. In case of error by the user and keep both modes activated simultaneously, the default vehicle will detect the Manual mode.

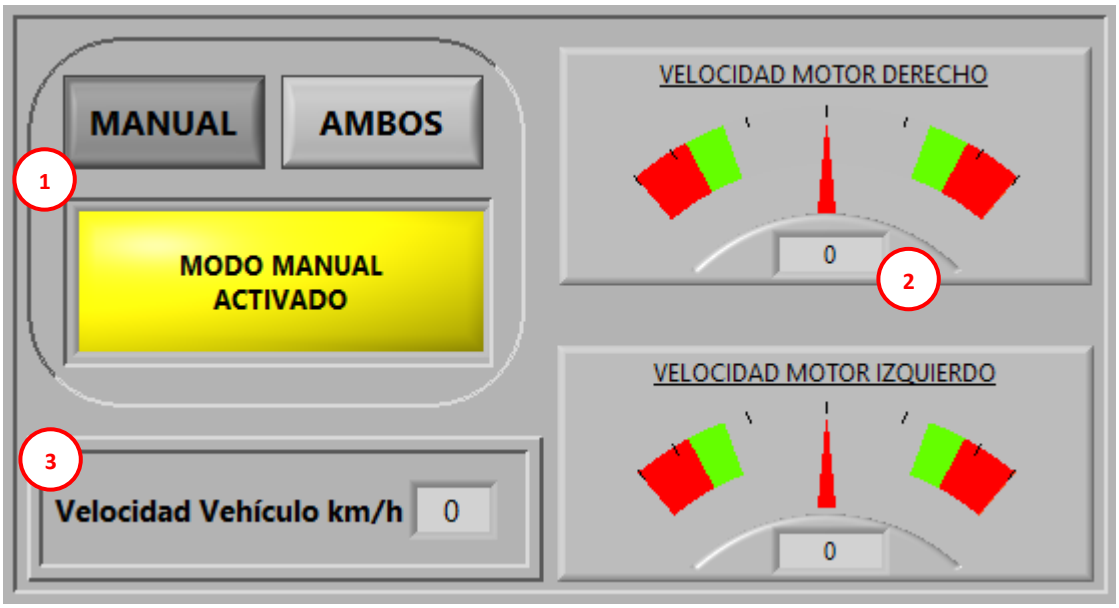
2.4.1. AUTONOMOUS MODE



1. **AUTONOMOUS AND RETURN**: To activate the autonomous mode it is necessary to press the “AUTONOMOUS” button. While it is pressed, the yellow indicator will show the text “**AUTONOMOUS MODE ON**”. In case you want to interrupt the automatic mode in order to return the vehicle to the manned vessel, at any time you can press the "RETURN" button.

2. **LATITUDE Y LONGITUDE POTS**: It is necessary to enter the coordinates of the pots so that the vehicle can go towards them with an accuracy of up to 0.0001º. The route of the vehicle will always be in order from 1 to 5.
3. **LATITUDE Y LONGITUDE SHIP**: When the vehicle finishes the route through all the pots, it will return to the main vessel, for this it is necessary to geographically locate the vessel.

2.4.2. MANUAL MODE



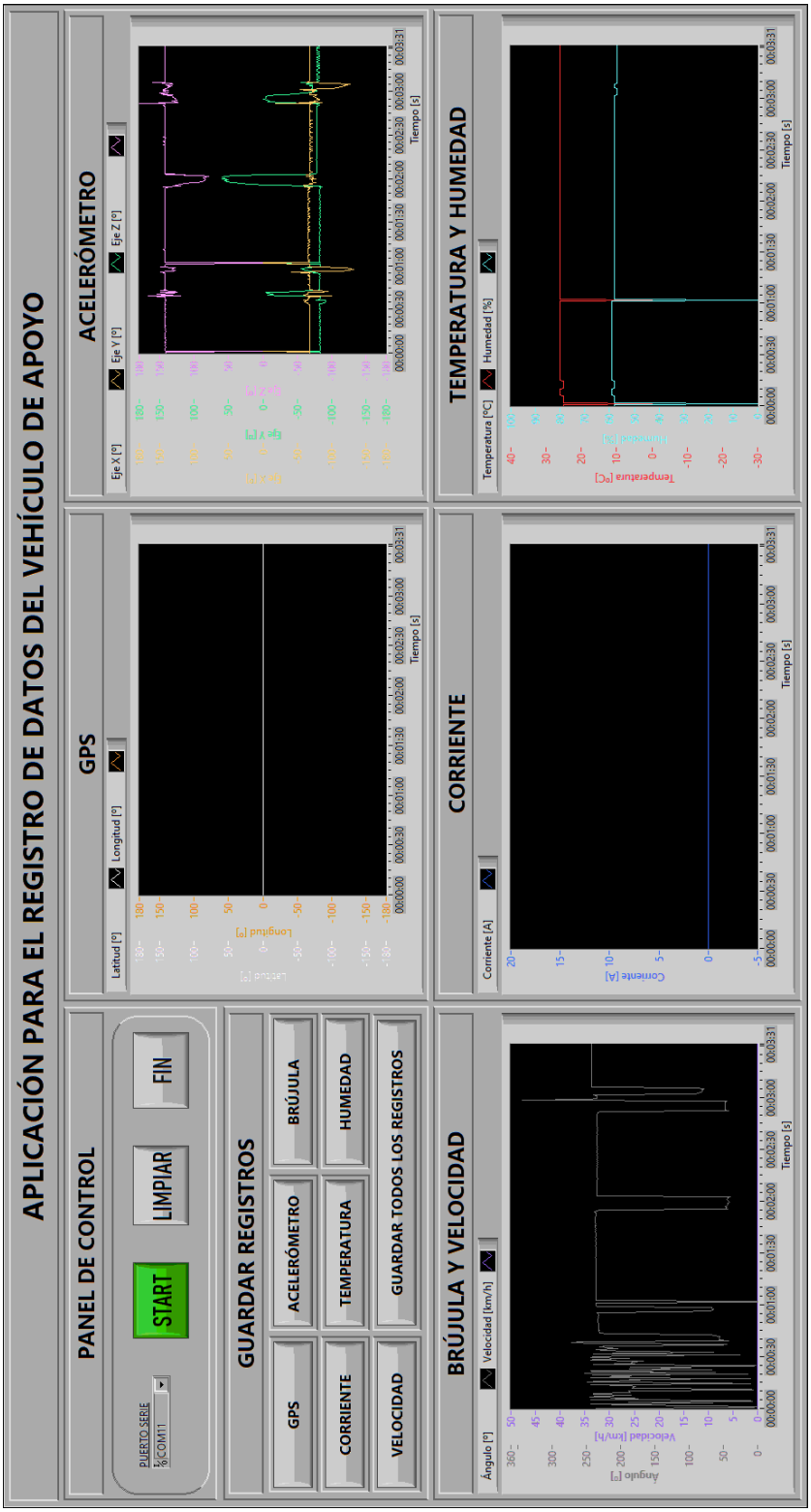
1. **MANUAL**: To activate the manual mode it is necessary to press the "MANUAL" button. While it is pressed, the yellow indicator will show the text "**MANUAL MODE ON**".
2. **RIGHT MOTOR SPEED AND LEFT MOTOR SPEED**: These levers serve to control the speed of the engines, with the right moving forward and the left moving backwards. Under each of these, the speed at which the motor is being rotated is shown in percent (%).
3. **VEHICLE SPEED**: Vehicle speed indicator.

2.5. Pot Pick Up

The green indicators in the lower right corner of the interface will light up showing the text "**PICK UP POT X**" (where X is an integer from 1 to 5) depending on the pot that has exceeded the minimum capacity of crabs introduced by the user.

3. APPLICATION FOR DATA RECORDING OF SUPPORT VEHICLE

This is the application in charge of registering the vehicle data and saving the vehicle data.



3.1. *Control Panel*

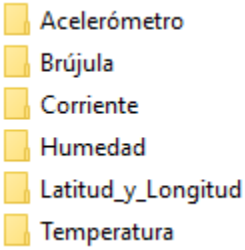
1. **SERIAL PORT**: Before starting the application, the user must select the serial port on which the wireless communication device is connected. An option of type "**COM X**" should be searched for, "X" being an integer.
2. **START, CLEAR AND FINISH**: These three buttons control the overall operation of the program. With the **START** button the program starts and with the **FINISH** it ends. If at any time the user wants to restart the sampling or clean the data obtained for any reason, he must press the **CLEAR** button.

3.2. *Save Records*

1. **SIMPLE RECORD**: If you want to save the record of one or several variables simultaneously, you must press the button of the variable that you want to save, while the button is pressed, the record will be taken and saved.
2. **SAVE ALL RECORDS**: If you want to save all the records simultaneously, you can press the "**SAVE ALL RECORDS**" button to simplify the operation.

When you start data logging, a folder will be created on the PC desktop like the following:

Within which sub-folders will be generated for the variable(s) you want to save:

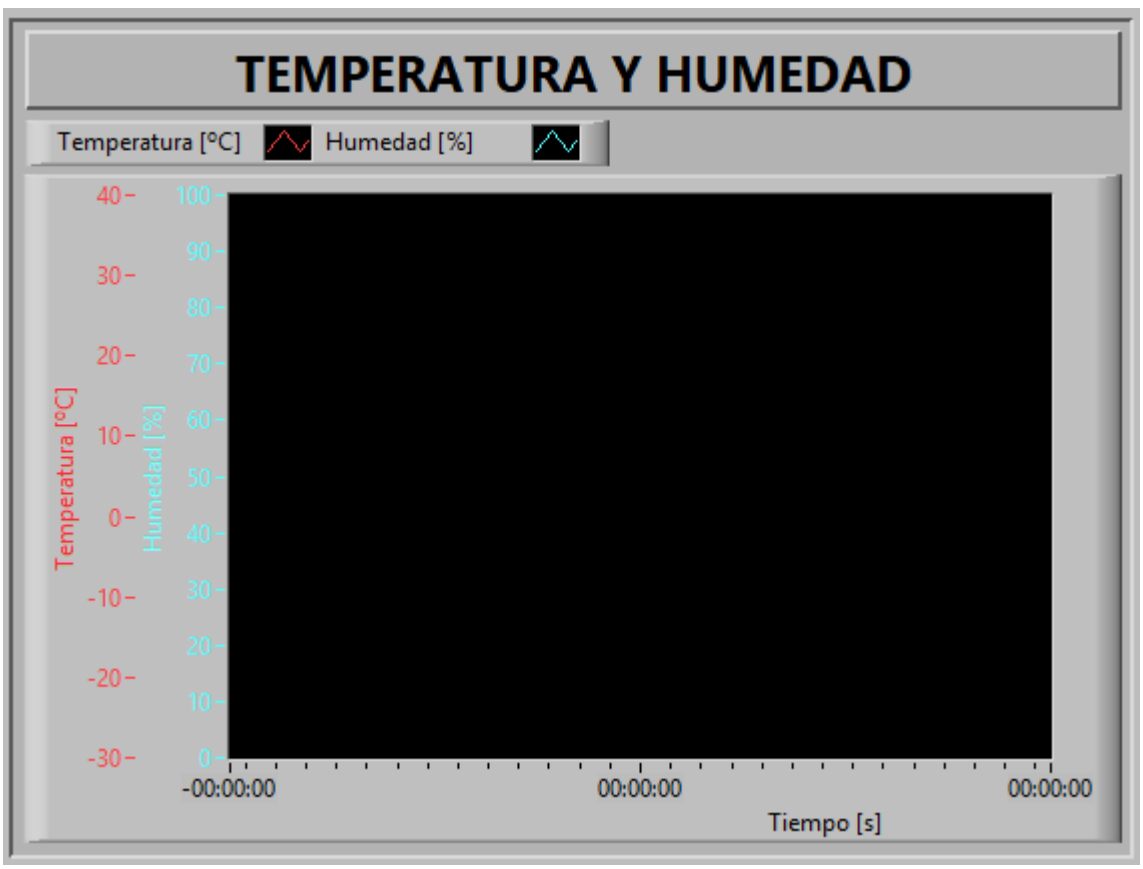


Each folder contains one or more Microsoft Excel™ spreadsheets depending on how many times you have wanted to collect and save data. These Excel™ documents are saved according to the following specifications:

Time	Latitud	Time 1	Longitud
16/08/2019 13:47:16,028	0	16/08/2019 13:47:16,028	0
16/08/2019 13:47:18,750	0	16/08/2019 13:47:18,750	0
16/08/2019 13:47:19,049	0	16/08/2019 13:47:19,049	0
16/08/2019 13:47:19,349	0	16/08/2019 13:47:19,349	0

The application is responsible for recording the exact date and time of data collection and divides it into the sampled data (in the previous case, since it is GPS, it is necessary to register two variables, Latitude and Longitude).

3.3. Plots



The rest of the interface is divided into 5 graphics which offer visual support to the user. These are:

a. **GPS**: Registers the **LATITUDE** and **LONGITUDE** coordinates of the support vehicle.
b. **ACCELEROMETER**: Registers the **INCLINATION** of the **X**, **Y** and **Z** axes of the support vehicle.
c. **COMPASS AND SPEED**: Registers the **ORIENTATION** and **SPEED** of the support vehicle.
d. **CURRENT**: Register the **CURRENT** consumed by the support vehicle.
e. **TEMPERATURE AND HUMIDITY**: Record the **TEMPERATURE** and **HUMIDITY** to which the support vehicle is located.

4. INITIALIZATION

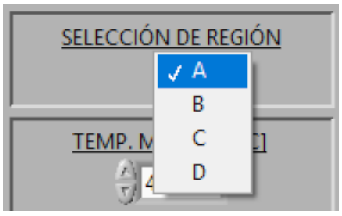
4.1. *Start Sequence*

For the correct operation of the application it is necessary to follow the following sequence:

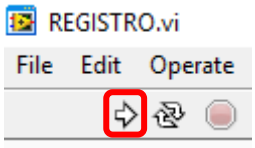
1. **OPEN THE APPLICATION**: To open the desired application (s) it is necessary to double click on the icon

(in case you want to open the control application) or

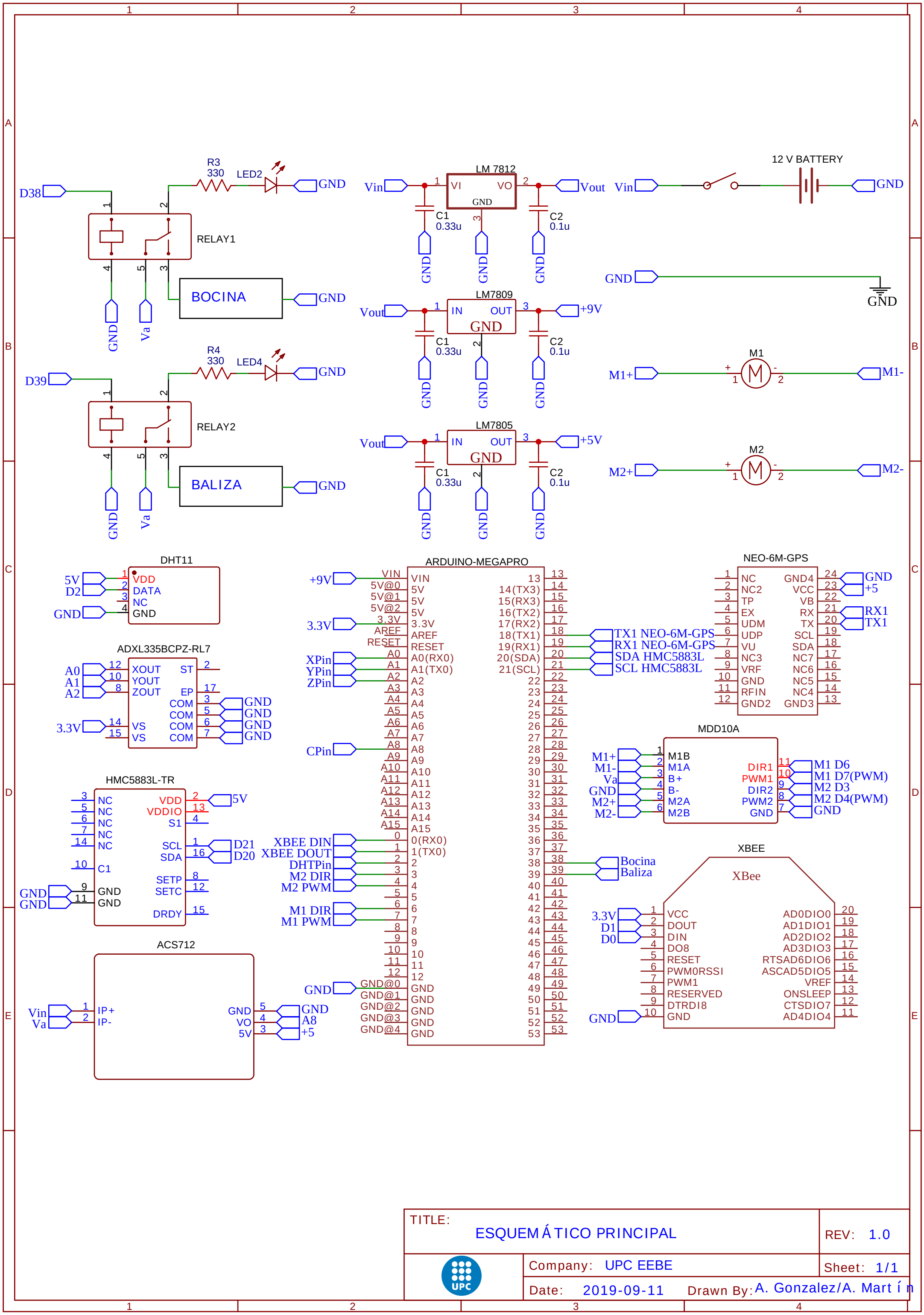
(to open the registration application).
2. **FINISH THE PREVIOUS PROGRAM**: In order not to confuse the application (which starts by default), it is necessary to terminate the application by clicking on the FINISH (see 2.1.2.) or END (see 3.1.2.) buttons.
3. **INTRODUCE DATA**: The types of data that can be entered in this application are two:
    a. *Numeric Data*: they are the majority and are the data that correspond to numerical values such as Longitude and Latitude (see 2.4.2.) or maximum and minimum temperatures (see 2.1.4.).
    b. *Alphanumeric Data*: are the data that correspond to the Serial Ports (see 2.1.1. or 3.1.1.) and Region Selection (see 2.1.3.) which open drop-down with facilitating the data to enter.



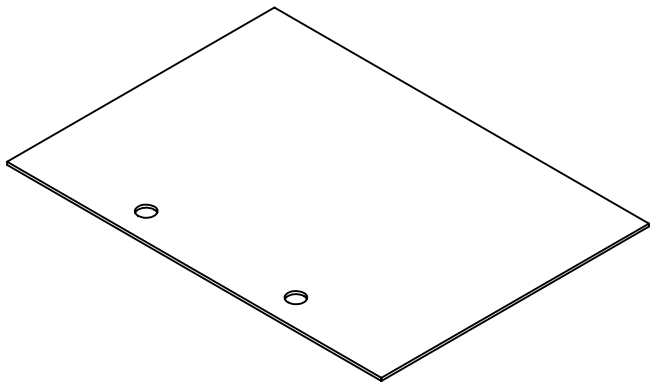
4. **START THE PROGRAM**: To start the application whenever the FINISH (see 2.1.2.) or END (see 3.1.2.) buttons are pressed, it is necessary to press the white arrow in the upper left corner of the program:

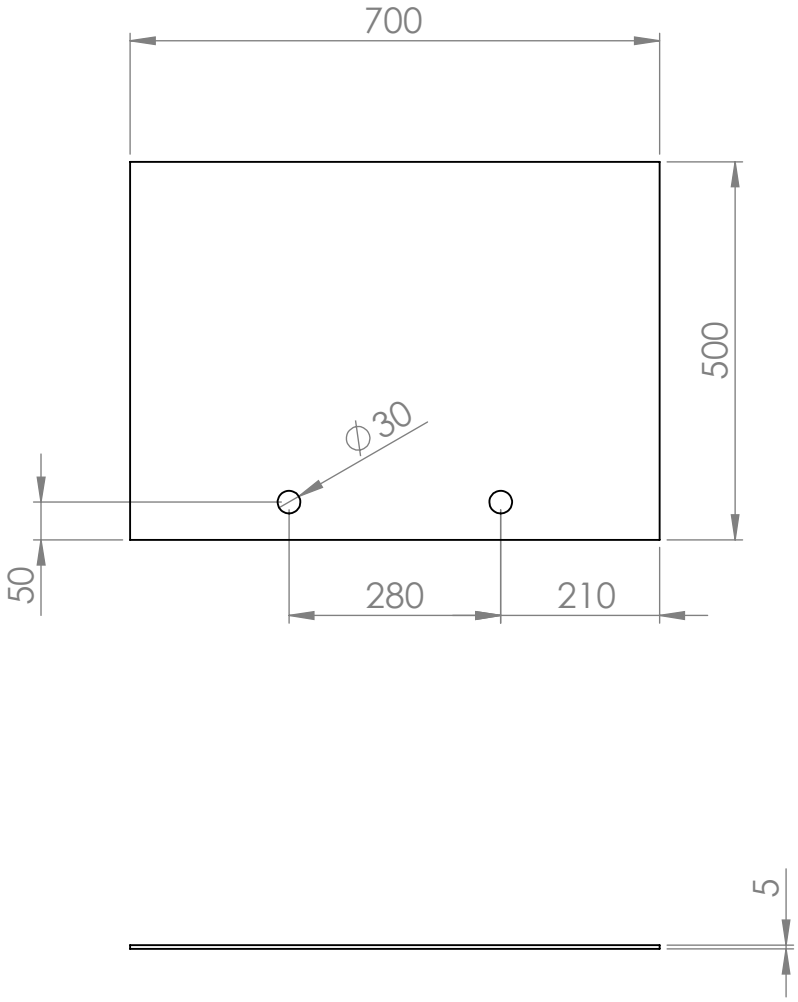


5. **START THE APPLICATION**: Finally, to start the application it is necessary to press the **START** button (see 2.1.2 or 3.1.2.)



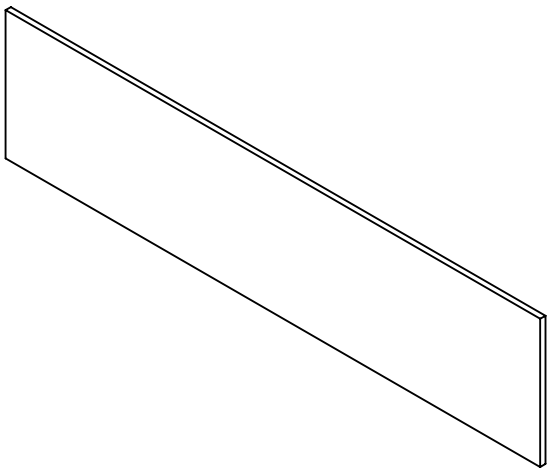
TITLE: ESQUEMÁTICO PRINCIPAL	REV: 1.0
Company: UPC EEBE	Sheet: 1/1
Date: 2019-09-11	Drawn By: A. Gonzalez/A. Martín

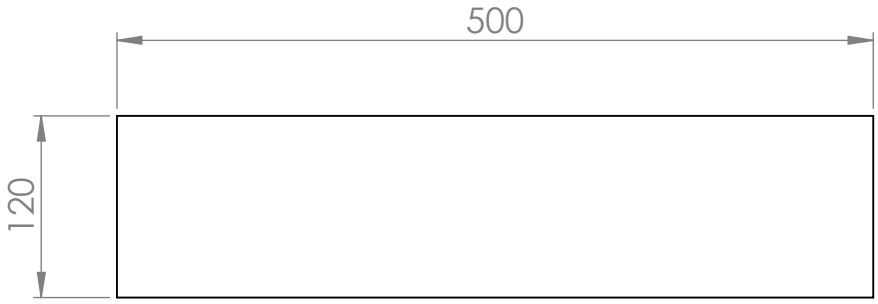




VEHÍCULO DE APOYO PARA LA PESCA DE FONDO MARINO CON NASAS		Material: Metacrilato	
Pieza: **Plancha Base**		Nº Pieza: 1	Cantidad: 1
Antonio Gonzalez Cifuentes	Firma:	Escala: 1:10	Hoja: 1 de 1
Álvaro Martín Velasco	Firma:	UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola d'Enginyeria de Barcelona Est	

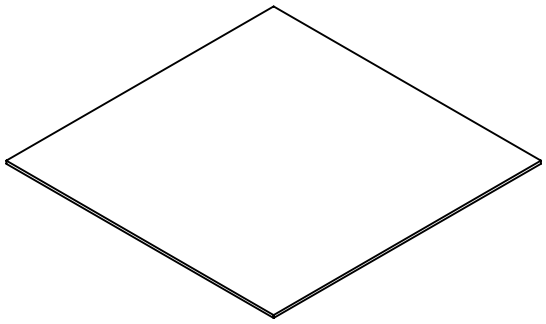
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

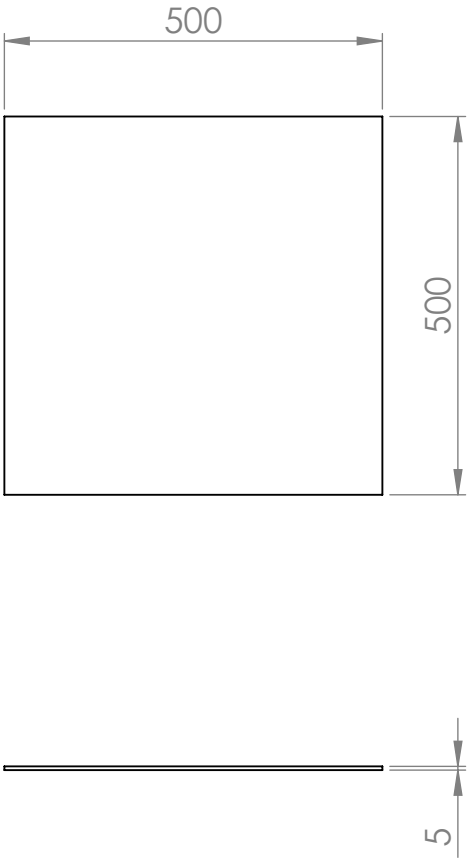




VEHÍCULO DE APOYO PARA LA PESCA DE FONDO MARINO CON NASAS		Material: Metacrilato	
Pieza: **Pared Caja Hermética**		Nº Pieza: 2	Cantidad: 4
Antonio Gonzalez Cifuentes	Firma:	Escala: 1:5	Hoja: 1 de 1
Álvaro Martín Velasco	Firma:	UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola d'Enginyeria de Barcelona Est	

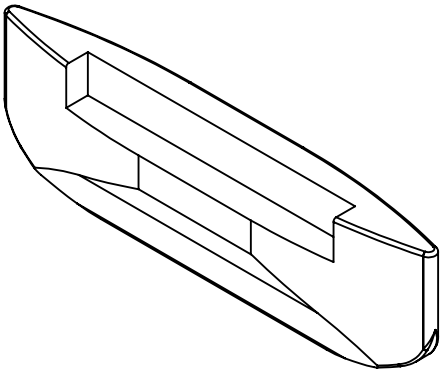
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

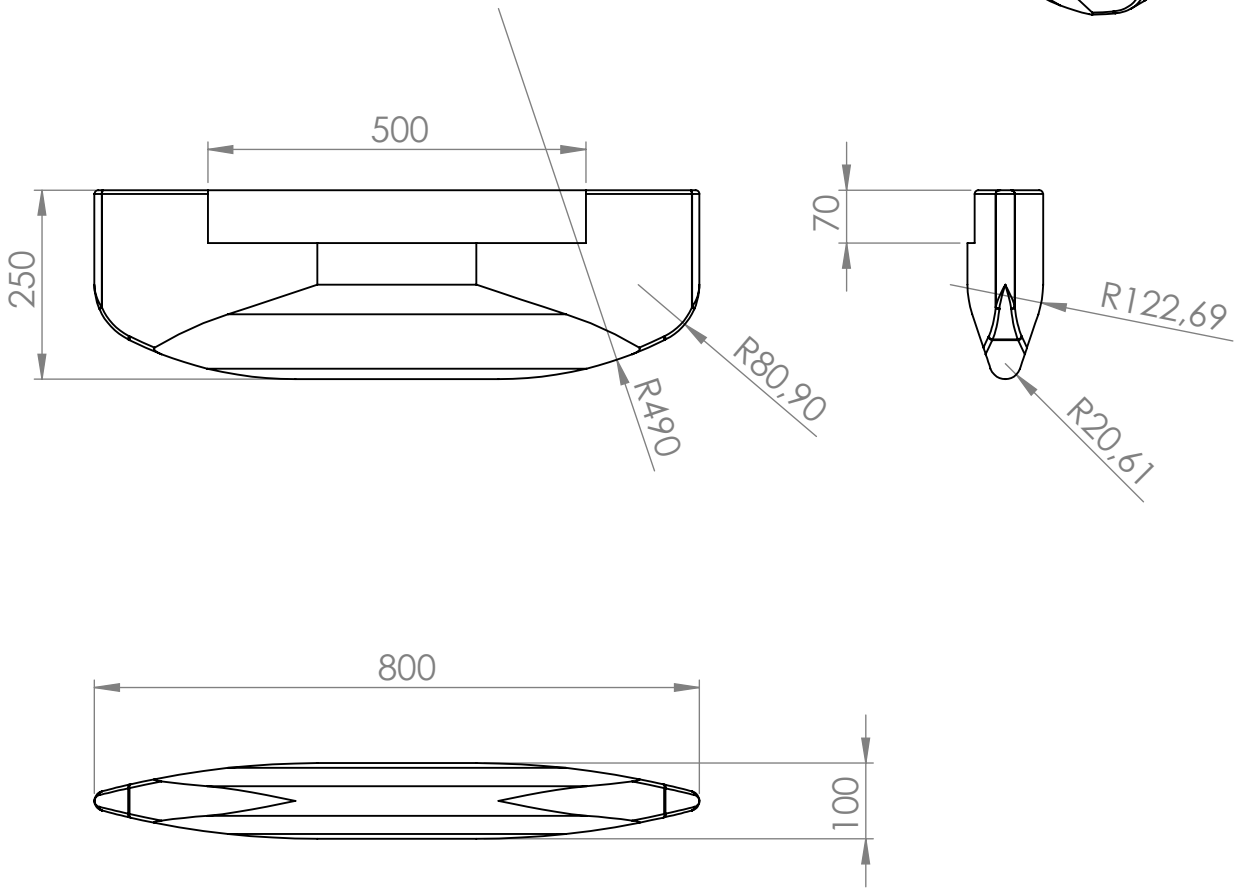




VEHÍCULO DE APOYO PARA LA PESCA DE FONDO MARINO CON NASAS		Material: Metacrilato	
Pieza: **Cubierta Caja Hermética**		Nº Pieza: 3	Cantidad: 1
Antonio Gonzalez Cifuentes	Firma:	Escala: 1:10	Hoja: 1 de 1
Álvaro Martín Velasco	Firma:	UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola d'Enginyeria de Barcelona Est	

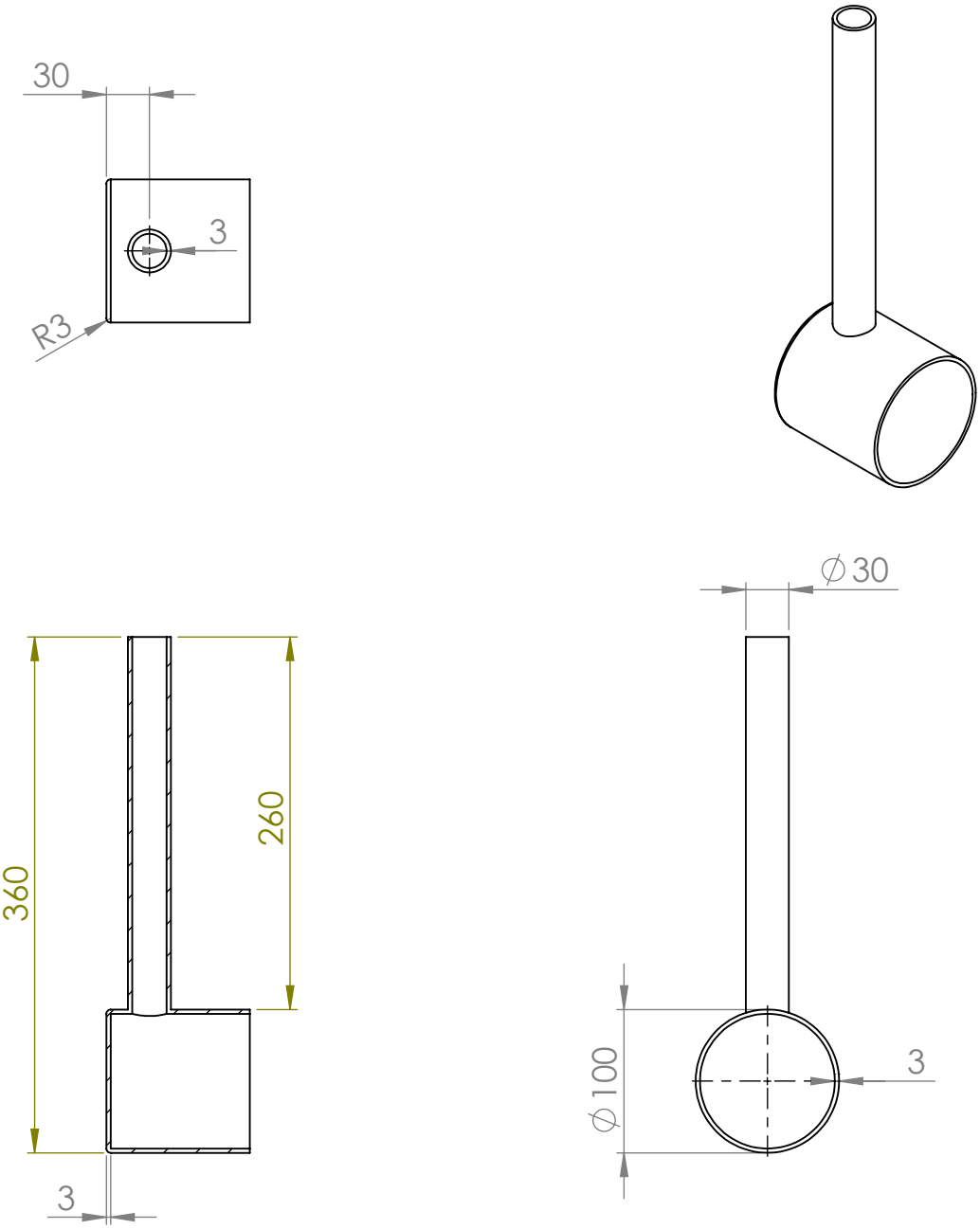
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.





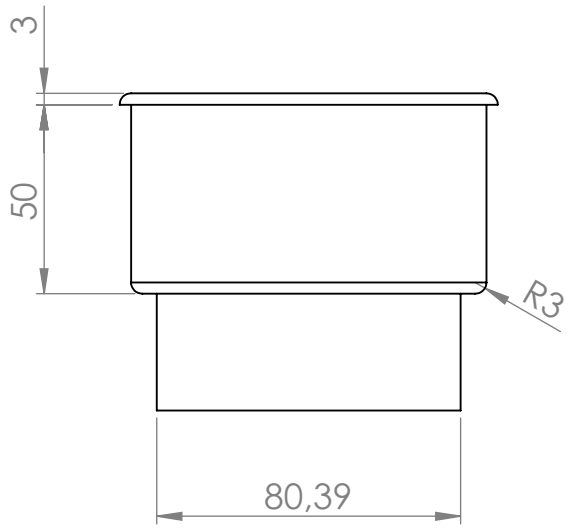
VEHÍCULO DE APOYO PARA LA PESCA DE FONDO MARINO CON NASAS		Material: Poliestireno expandido revestido en fibra de vidrio	
Pieza: **Flotadores**		Nº Pieza: 4	Cantidad: 2
Antonio Gonzalez Cifuentes	Firma:	Escala: 1:10	Hoja: 1 de 1
Álvaro Martín Velasco	Firma:	UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola d'Enginyeria de Barcelona Est	

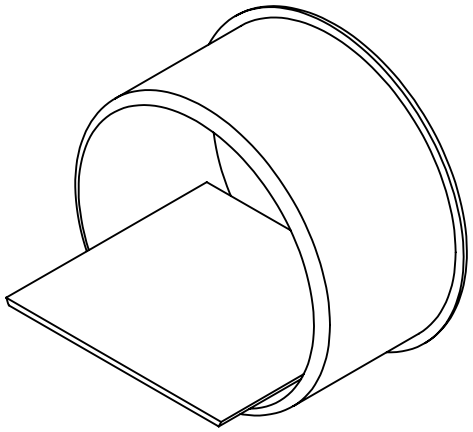
Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

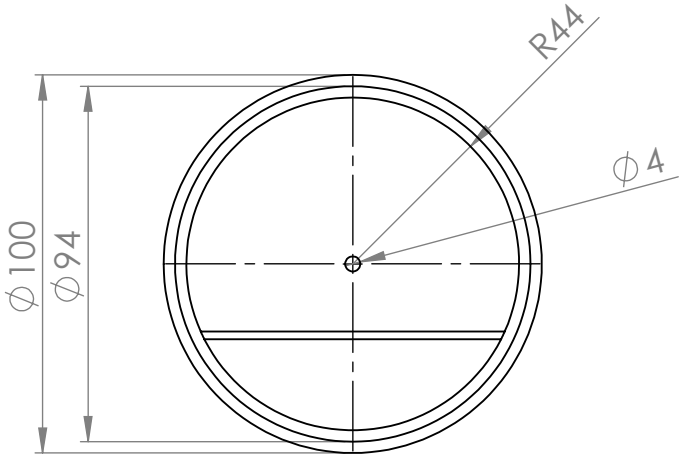


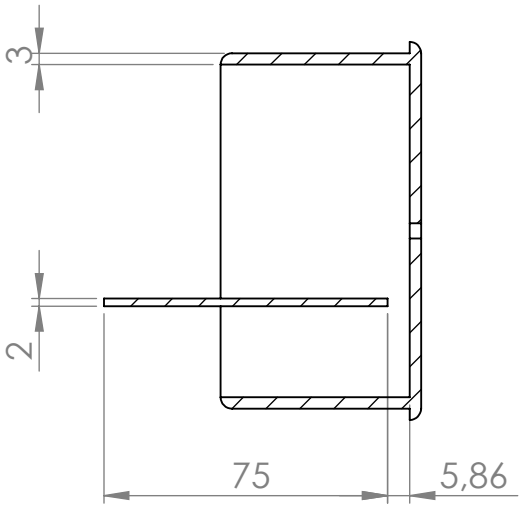
VEHÍCULO DE APOYO PARA LA PESCA DE FONDO MARINO CON NASAS		Material: PVC	
Pieza: **Transmisión**		Nº Pieza: 5	Cantidad: 2
Antonio Gonzalez Cifuentes	Firma:	Escala: 1:5	Hoja: 1 de 1
Álvaro Martín Velasco	Firma:	UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola d'Enginyeria de Barcelona Est	

Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.









VEHÍCULO DE APOYO PARA LA PESCA DE FONDO MARINO CON NASAS		Material: PVC	
Pieza: **Tapa Transmisión**		Nº Pieza: 6	Cantidad: 2
Antonio Gonzalez Cifuentes	Firma:	Escala: 1:5	Hoja: 1 de 1
Álvaro Martín Velasco	Firma:	UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola d'Enginyeria de Barcelona Est	

Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

```
/*********************************************/
/*                                           */
/*   CODIGO PARA LA APLICACION DEL VEHICULO  */
/*    DE APOYO A LA PESCA DE FONDO MARINO    */
/*                                           */
/*                                           */
/*   ANTONIO GONZALEZ                        */
/*   ALVARO MARTIN                           */
/*                                           */
/*   2019                                    */
/*                                           */
/*********************************************/


/************************/
/*      LIBRERIAS       */
/************************/


//Librerias Utilizadas//
#include <DHT.h>
#include <Wire.h>
#include <DFRobot_QMC5883.h>
#include <TinyGPS.h>
#include <math.h>


/************************/
/*         PINES        */
/************************/

//Temperatura y Humedad//
#define DHTPIN 2

//Acelerometro//
#define xPin A0
#define yPin A1
#define zPin A2

//Corriente//
#define CPin A8

//Baliza y Bocina//
#define Bocina 38
#define Baliza 39

//Motores//
#define MotorDDirection 7
#define MotorDSpeed 6
#define MotorIDirection 4
#define MotorISpeed 3


/************************/
/*     CONDICIONES      */
/************************/
```

```
//Condiciones Sensor de Temperatura y Humedad//
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);


//Condiciones Acelerometro//
int acelx;
int acely;
int acelz;


//Condiciones Brujula//
DFRobot_QMC5883 compass;


//Condiciones Sensor Corriente//
float Sensibilidad=0.066;


//Condiciones GPS//
TinyGPS gps;
int constRegLat;
int constRegLon;
int velvehiculo;
float lat, lon;

//Condiciones Bocina//
#define T_apagado 120000
#define T_encendido 2000


//Constante de trama//
byte buff[38];

/************************/
/*       CREACION       */
/************************/

//Creacion de los valores random de cangrejos por boya//
int Randomin1 = 0;
int Randomin2 = 0;
int Randomin3 = 0;
int Randomin4 = 0;
int Randomin5 = 0;

int Randomout1 = 0;
int Randomout2 = 0;
int Randomout3 = 0;
int Randomout4 = 0;
int Randomout5 = 0;

//Creacion del timer//
unsigned long timerbocina=0;

//Creacion Variables Auxiliares Boyas//
int Boya1 = 1;
int Boya2 = 0;
```

```
int Boya3 = 0;
int Boya4 = 0;
int Boya5 = 0;
int Emb = 0;


/************************/
/*          SETUP        */
/************************/

void setup() {
  Serial.begin(9600);

  //Setup Temperatura y Humedad//
  dht.begin();

  //Setup Brujula//
  compass.begin();
  compass.setRange(QMC5883_RANGE_2GA);
  compass.setMeasurementMode(QMC5883_CONTINOUS);
  compass.setDataRate(QMC5883_DATARATE_50HZ);
  compass.setSamples(QMC5883_SAMPLES_8);

  //Setup GPS//
  Serial1.begin(9600);

  //Declaracion Pines Baliza y Bocina//
  pinMode(Baliza, OUTPUT);
  pinMode(Bocina, OUTPUT);

  //Declaracion Pines Motores//
  pinMode(MotorDDirection, OUTPUT);
  pinMode(MotorDSpeed, OUTPUT);
  pinMode(MotorIDirection, OUTPUT);
  pinMode(MotorISpeed, OUTPUT);

  //Temporizador Bocina//
  timerbocina = millis();
}


/************************/
/*         MAIN         */
/************************/

void loop() {
  while (Serial.available()>43){

    /************************/
    /*      OBTENCION       */
    /************************/

    //Obtencion del Comando de Region//
    int cmdReg = Serial.read();

    //Obtencion del Comando de Autonomia//
    int cmdAutoMan = Serial.read();
```

```
//Obtencion de los Comandos de Baliza y Bocina//
int cmdBal = Serial.read();
int cmdBoc = Serial.read();

//Obtencion de los Comandos de Velocidad//
int VelMotDer = Serial.read();
int cmdDer = Serial.read();

int VelMotIzq = Serial.read();
int cmdIzq = Serial.read();

//Obtencion de las Latitudes y Longitudes de las boyas y la embarcacion//
int Lat1u = Serial.read();
int Lat1d = Serial.read();
int Lat1c = Serial.read();
int Lon1u = Serial.read();
int Lon1d = Serial.read();
int Lon1c = Serial.read();

int Lat2u = Serial.read();
int Lat2d = Serial.read();
int Lat2c = Serial.read();
int Lon2u = Serial.read();
int Lon2d = Serial.read();
int Lon2c = Serial.read();

int Lat3u = Serial.read();
int Lat3d = Serial.read();
int Lat3c = Serial.read();
int Lon3u = Serial.read();
int Lon3d = Serial.read();
int Lon3c = Serial.read();

int Lat4u = Serial.read();
int Lat4d = Serial.read();
int Lat4c = Serial.read();
int Lon4u = Serial.read();
int Lon4d = Serial.read();
int Lon4c = Serial.read();

int Lat5u = Serial.read();
int Lat5d = Serial.read();
int Lat5c = Serial.read();
int Lon5u = Serial.read();
int Lon5d = Serial.read();
int Lon5c = Serial.read();

int LatBu = Serial.read();
int LatBd = Serial.read();
int LatBc = Serial.read();
int LonBu = Serial.read();
int LonBd = Serial.read();
int LonBc = Serial.read();

if (cmdReg == 0){
  constRegLat = 1;
```

```
  constRegLon = -1;
}

if (cmdReg == 1){
  constRegLat = 1;
  constRegLon = 1;
}

if (cmdReg == 2){
  constRegLat = -1;
  constRegLon = -1;
}

if (cmdReg == 3){
  constRegLat = -1;
  constRegLon = 1;
}

float lat1 = unirlat(Lat1u, Lat1d, Lat1c);
float lon1 = unirlon(Lon1u, Lon1d, Lon1c);

float lat2 = unirlat(Lat2u, Lat2d, Lat2c);
float lon2 = unirlon(Lon2u, Lon2d, Lon2c);

float lat3 = unirlat(Lat3u, Lat3d, Lat3c);
float lon3 = unirlon(Lon3u, Lon3d, Lon3c);

float lat4 = unirlat(Lat4u, Lat4d, Lat4c);
float lon4 = unirlon(Lon4u, Lon4d, Lon4c);

float lat5 = unirlat(Lat5u, Lat5d, Lat5c);
float lon5 = unirlon(Lon5u, Lon5d, Lon5c);

float latB = unirlat(LatBu, LatBd, LatBc);
float lonB = unirlon(LonBu, LonBd, LonBc);


/************************/
/*    INSTRUCCIONES     */
/************************/

//Instrucciones Temperatura y Humedad//
char h = dht.readHumidity();
char t = dht.readTemperature();


//Instrucciones Acelerometro//
int xRead = analogRead(xPin);
int yRead = analogRead(yPin);
int zRead = analogRead(zPin);

acelx = 2.5714*xRead-925.71;
acely = 2.5714*yRead-925.71;
acelz = 2.5714*zRead-925.71;


//Instrucciones Brujula//
```

```
    Vector norm = compass.readNormalize();
    float heading = atan2(norm.YAxis, norm.XAxis);
    float declinationAngle = 1.3;
    if (heading < 0){
      heading += 2 * PI;
    }
    if (heading > 2 * PI){
      heading -= 2 * PI;
    }
    int headingDegrees = heading * 180/M_PI + 100 + declinationAngle;
    if (headingDegrees>360){
      headingDegrees = headingDegrees - 360;
    }

    //Instrucciones Sensor Corriente//
    float voltajeSensor= analogRead(CPin)*(5.0 / 1023.0);
    float corriente=(voltajeSensor-2.5)/Sensibilidad;


    //Instrucciones GPS//
    if(gps.encode(Serial1.read())){
      gps.f_get_position(&lat, &lon);
      velvehiculo = gps.f_speed_kmph();
      }

    int latpart1 = lat;
    int latpart2 = (lat - latpart1)*100;
    int latpart3 = (lat*100 - latpart1*100 - latpart2)*180;

    int lonpart1 = lon;
    int lonpart2 = (lon - lonpart1)*100;
    int lonpart3 = (lon*100 - lonpart1*100 - lonpart2)*100;


    //Incremento Cangrejos por ciclo//
    Randomin1 = Randomin1 + random(0,2);
    Randomin2 = Randomin2 + random(0,2);
    Randomin3 = Randomin3 + random(0,2);
    Randomin4 = Randomin4 + random(0,2);
    Randomin5 = Randomin5 + random(0,2);

    Randomout1 = Randomin1/200;
    Randomout2 = Randomin2/200;
    Randomout3 = Randomin3/200;
    Randomout4 = Randomin4/200;
    Randomout5 = Randomin5/200;

    //Formar y enviar Trama//
    formar_trama(t, h, acelx, acely, acelz, headingDegrees, corriente,
latpart1, latpart2, latpart3, lonpart1, lonpart2, lonpart3, Randomout1,
Randomout2, Randomout3, Randomout4, Randomout5, velvehiculo);
    Serial.write(buff,38);

    //Instrucciones Baliza//
    if (cmdBal==1){
      Encender(Baliza);
    }
```

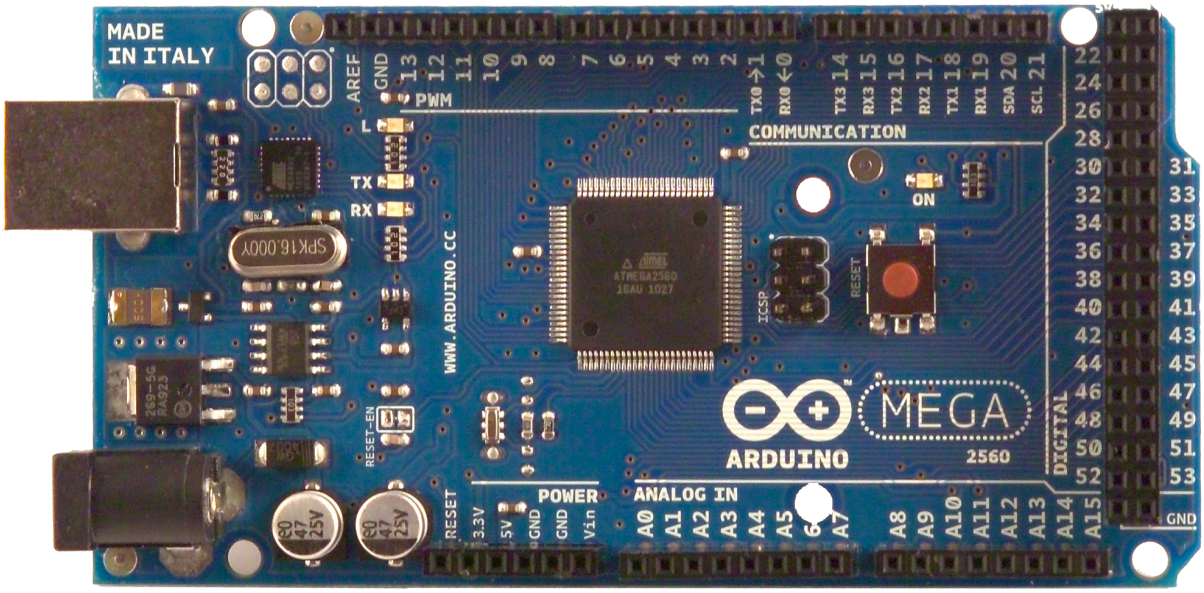
```
    else if (cmdBal==0){
      Apagar(Baliza);
    }

    //Instrucciones Bocina//
    if ((millis() - timerbocina < T_apagado) && (cmdBoc == 1) ){
      Apagar(Bocina);
    }
    else if ((millis() - timerbocina > T_apagado) && (millis() - timerbocina
< (T_apagado + T_encendido)) && (cmdBoc == 1)){
      Encender(Bocina);
    }
    else if ((millis() - timerbocina > (T_apagado + T_encendido)) && (cmdBoc
== 1)){
      Apagar(Bocina);
      timerbocina = millis();
    }

    if (cmdBoc == 0){
      Apagar(Bocina);
    }

    //Instrucciones Control de Motores//
    if (cmdAutoMan == 0){
      if (VelMotDer <= 130){
        digitalWrite(MotorDDirection, LOW);
      }
      else if ((VelMotDer > 130) && (cmdDer == 0)){
        digitalWrite(MotorDDirection, HIGH);
        analogWrite(MotorDSpeed,VelMotDer);
      }
      else if ((VelMotDer > 130) && (cmdDer == 1)){
        digitalWrite(MotorDDirection, HIGH);
        analogWrite(MotorDSpeed,(VelMotDer*(-1)));
      }

      if (VelMotIzq <= 130){
        digitalWrite(MotorIDirection, LOW);
      }
      else if ((VelMotIzq > 130) && (cmdIzq == 0)){
        digitalWrite(MotorIDirection, HIGH);
        analogWrite(MotorISpeed,VelMotIzq);
      }
      else if ((VelMotIzq > 130) && (cmdIzq == 1)){
        digitalWrite(MotorIDirection, HIGH);
        analogWrite(MotorISpeed,(VelMotIzq*(-1)));
      }
    }

    float alpha1 = AnguloCoordenadas(lat,lon,lat1,lon1);
    float alpha2 = AnguloCoordenadas(lat,lon,lat2,lon2);
    float alpha3 = AnguloCoordenadas(lat,lon,lat3,lon3);
    float alpha4 = AnguloCoordenadas(lat,lon,lat4,lon4);
    float alpha5 = AnguloCoordenadas(lat,lon,lat5,lon5);
    float alphaB = AnguloCoordenadas(lat,lon,latB,lonB);
```

```
    if (cmdAutoMan == 1){

      BuscarBoya(alpha1, headingDegrees, lon, lat, lon1, lat1, Boya1, Boya2);

      BuscarBoya(alpha2, headingDegrees, lon, lat, lon2, lat2, Boya2, Boya3);

      BuscarBoya(alpha3, headingDegrees, lon, lat, lon3, lat3, Boya3, Boya4);

      BuscarBoya(alpha4, headingDegrees, lon, lat, lon4, lat4, Boya4, Boya5);

      BuscarBoya(alpha5, headingDegrees, lon, lat, lon5, lat5, Boya5, Emb);

      BuscarBoya(alphaB, headingDegrees, lon, lat, lonB, latB, Emb, Boya1);
    }
  }
}


/************************/
/*      FUNCIONES       */
/************************/

//Creacion de Trama//
void formar_trama(int temp, int hum, int acelx, int acely, int acelz, int
brujula, int corriente, int latpart1, int latpart2, int latpart3, int
lonpart1, int lonpart2, int lonpart3, int random1, int random2, int random3,
int random4, int random5, int velvehiculo) {
  buff[0] = (temp >> 8) & 255;
  buff[1] =  temp & 255;
  buff[2] = (hum >> 8) & 255;
  buff[3] =  hum & 255;
  buff[4] = (acelx >> 8) & 255;
  buff[5] =  acelx & 255;
  buff[6] = (acely >> 8) & 255;
  buff[7] =  acely & 255;
  buff[8] = (acelz >> 8) & 255;
  buff[9] =  acelz & 255;
  buff[10] = (brujula >> 8) & 255;
  buff[11] =  brujula & 255;
  buff[12] = (corriente >> 8) & 255;
  buff[13] =  corriente & 255;
  buff[14] = (latpart1 >> 8) & 255;
  buff[15] =  latpart1 & 255;
  buff[16] = (latpart2 >> 8) & 255;
  buff[17] =  latpart2 & 255;
  buff[18] = (latpart3 >> 8) & 255;
  buff[19] =  latpart3 & 255;
  buff[20] = (lonpart1 >> 8) & 255;
  buff[21] =  lonpart1 & 255;
  buff[22] = (lonpart2 >> 8) & 255;
  buff[23] =  lonpart2 & 255;
  buff[24] = (lonpart3 >> 8) & 255;
  buff[25] =  lonpart3 & 255;
  buff[26] = (random1 >> 8) & 255;
  buff[27] =  random1 & 255;
  buff[28] = (random2 >> 8) & 255;
  buff[29] =  random2 & 255;
```

```
  buff[30] = (random3 >> 8) & 255;
  buff[31] =  random3 & 255;
  buff[32] = (random4 >> 8) & 255;
  buff[33] =  random4 & 255;
  buff[34] = (random5 >> 8) & 255;
  buff[35] =  random5 & 255;
  buff[36] = (velvehiculo >> 8) & 255;
  buff[37] =  velvehiculo & 255;
}

//Obtencion de latitud//
float unirlat (int latu,int latd, int latc){
  float lat = (latu + (latd*0.01) + (latc*0.0001))*constRegLat;
  return lat;
}

//Obtencion de longitud//
float unirlon (int lonu,int lond, int lonc){
  float lon = (lonu + (lond*0.01) + (lonc*0.0001))*constRegLon;
  return lon;
}

//Encender dispositivo (baliza o bocina)//
void Encender (int dispositivo){
  digitalWrite (dispositivo, HIGH);
}

//Apagar dispositivo (baliza o bocina)//
void Apagar (int dispositivo){
  digitalWrite (dispositivo, LOW);
}

//Hallar el angulo entre el vehiculo y la boya//
float AnguloCoordenadas(float lat, float lon, float latBB, float lonBB){
  float y = latBB - lat;
  float x = lonBB - lon;
  float ang = atan2(y,x);
  float alpha = ang *(180/PI);
  return alpha;
}

//Dirigirse hacia la boya//
void BuscarBoya (float alpha, int angulo, float lon, float lat, float lonBB,
float latBB, int Boya, int Boyasig){
  float beta;
  if (((abs(lon - lonBB) > 0.0001) || (abs(lat - latBB) > 0.0001)) && (Boya =
1)){

    if (alpha <= 90){
      beta = 90 - alpha;
    }
    else if ((alpha > 90) && (alpha <= 180)){
      beta = 360 - alpha;
    }
    else if ((alpha > 180) && (alpha <= 270)){
      beta = 360 - alpha;
    }
```

```
    else if ((alpha > 360) && (alpha <= 360)){
      beta = 360 - alpha;
    }

    if ((abs(beta - angulo) > 180) && (abs(beta - angulo) < 350)){
      GirarAntihorario();
    }
    else if ((abs(beta - angulo) < 180) && (abs(beta - angulo) > 10)){
      GirarHorario();
    }
    else {
      Avanza();
    }
  }

  else if (((abs(lon - lonBB) <= 0.0001) && (abs(lat - latBB) <= 0.0001)) &&
(Boya = 1)){
  Boya = 0;
  Boyasig = 1;
  }
}

//Girar Sentido Horario//
void GirarHorario(void){
  digitalWrite(MotorIDirection, HIGH);
  analogWrite(MotorISpeed,110);
  digitalWrite(MotorDDirection, HIGH);
  analogWrite(MotorDSpeed,-110);
}

//Girar Sentido Horario//
void GirarAntihorario(void){
  digitalWrite(MotorIDirection, HIGH);
  analogWrite(MotorISpeed,-110);
  digitalWrite(MotorDDirection, HIGH);
  analogWrite(MotorDSpeed,110);
}

//Avanzar//
void Avanza(void){
  digitalWrite(MotorIDirection, HIGH);
  analogWrite(MotorISpeed,200);
  digitalWrite(MotorDDirection, HIGH);
  analogWrite(MotorDSpeed,200);
}
```

Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications — Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials — Page 6

Terms & Conditions — Page 7

Enviromental Policies
half sqm of green via Impatto Zero® — Page 7

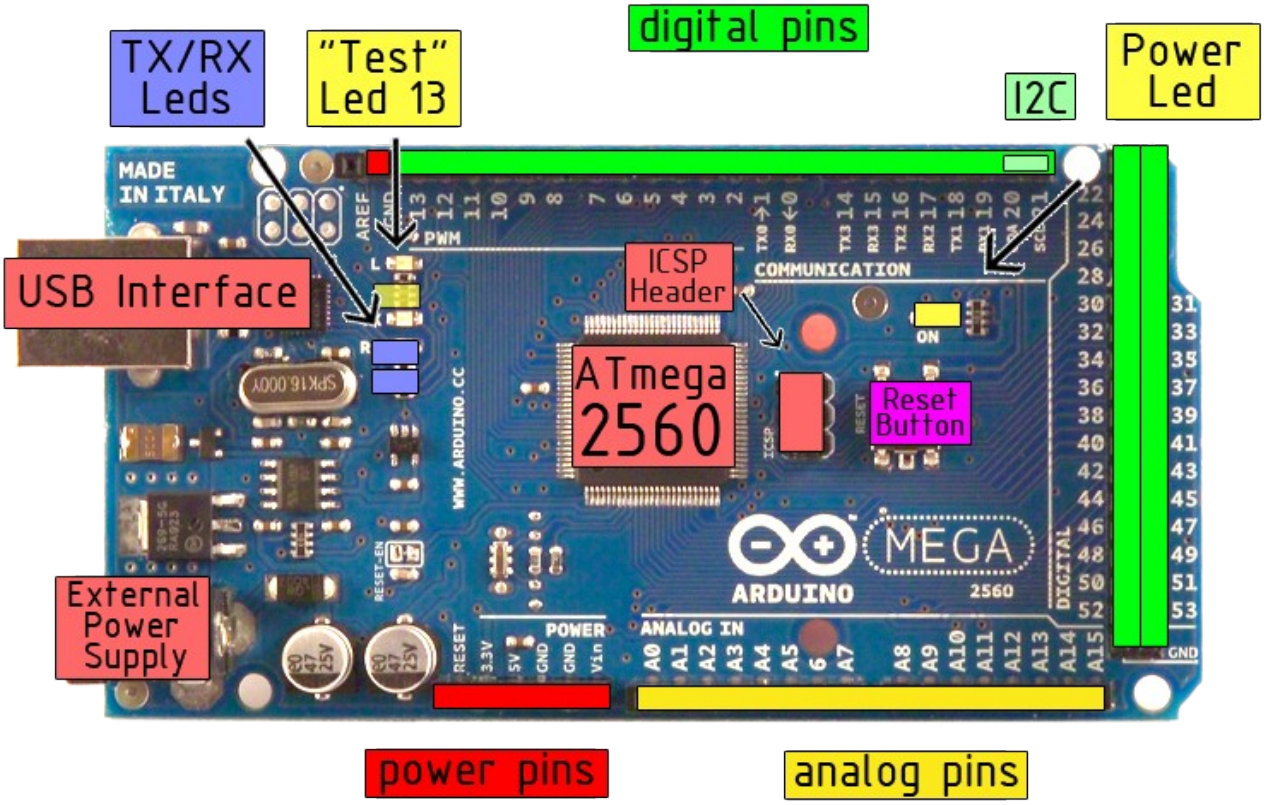
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

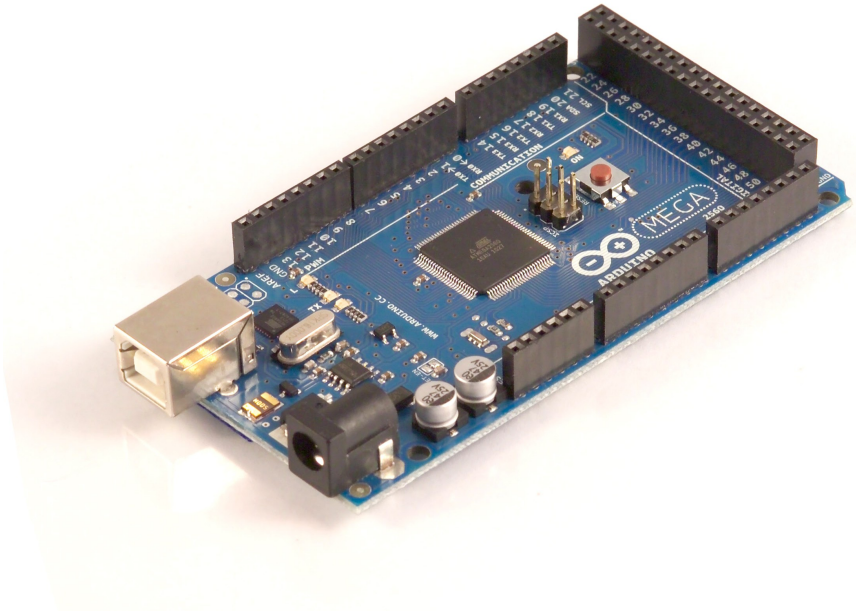
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.



Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I^2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

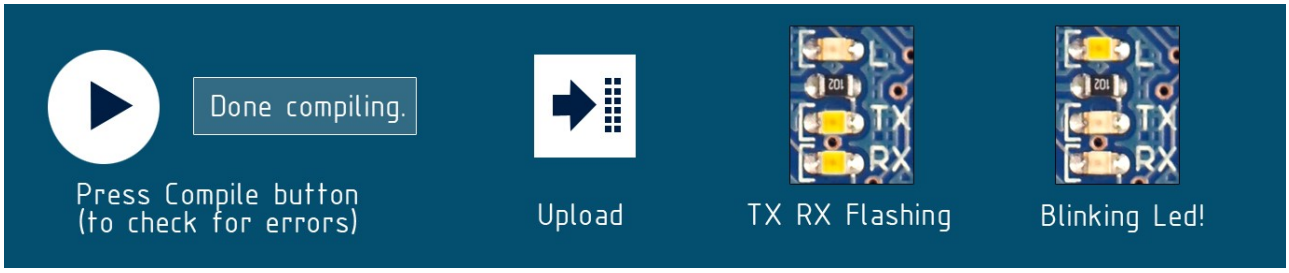
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

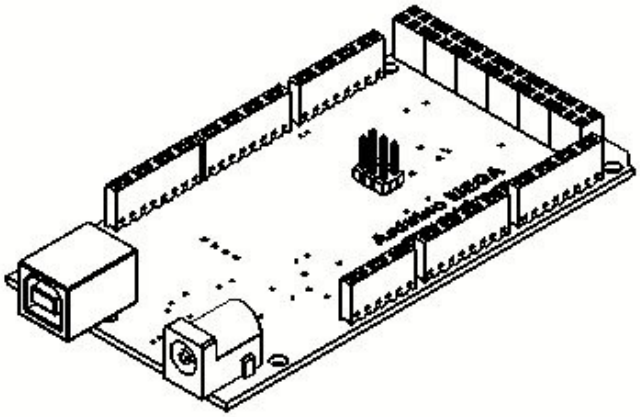
File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

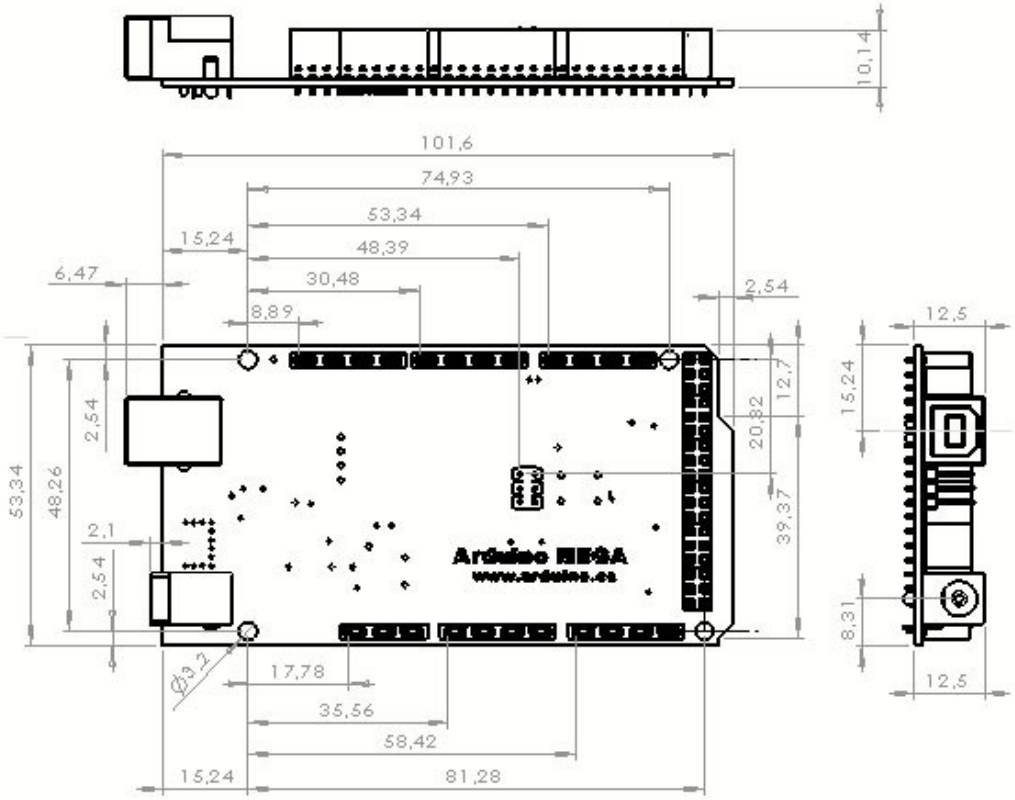
Once you have your skecth you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.






10,14
101,6
74,93
53,34
15,24
48,39
6,47
30,48
2,54
8,89
12,5
15,24
12,7
2,54
20,82
53,34
48,26
39,37
2,1
2,54
Arduino MEGA
www.arduino.cc
8,31
Ø3,2
17,78
12,5
35,56
58,42
15,24
81,28

Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

xunzel™

applied Solar and Wind Energy™

FICHA TÉCNICA

SOLARX™ Series

SOLARX-48
12V - 48Ah - C120
12V - 42Ah - C20

SOLARX-78
12V - 78Ah - C120
12V - 68Ah - C20

SOLARX-120
12V - 120Ah - C120
12V - 105Ah - C20

SOLARX-240
12V - 240Ah - C120
12V - 210Ah - C20

GARDEN LAND CARAVAN MARINE

SOLARX™ Series-XUNZEL

XUNZEL Baterías solares AGM de 12V, de descarga profunda y sin mantenimiento

Características del Producto

- Baterías solares de 12V totalmente **libres de mantenimiento**. No hay que rellenarlas de agua durante toda su vida útil
- **Selladas y seguras**. No derraman
- Baterías de ácido plomo reguladas por válvula (VRLA) con separadores de vidrio absorbente **(AGM)**. Resistentes a **descargas profundas**
- Ideales para aplicaciones fotovoltaicas y eólicas de aislada (Off-Grid) y Off-Shore: SAI (UPS) y backup, telecomunicaciones y CATV, tráfico, aplicaciones agropecuarias, barcos y caravanas, protección catódica, instalaciones profesionales....
- Especialmente diseñadas para usos con frecuentes descargas profundas. Resisten descargas cíclicas. Tienen una vida cíclica 30% mayor que las baterías de Standby por la robustez de su red y la particularidad de la pasta de sus placas
- **Alta durabilidad**, rendimiento excepcional y larga vida útil (vida de diseño en flotación 10 años)
- **Baja auto-descarga**, extraordinaria capacidad de descargas profundas y excelente comportamiento de recarga
- Baja resistencia interna. Por esto, es adecuado para aplicaciones de corrientes de descarga altas: inversores, propulsores, cabrestantes, arranque de motores....
- Se puede colocar en cualquier posición, vertical u horizontal
- Muy seguro, totalmente aislado para garantizar la protección contra contacto, con control de válvulas de cada célula y ultra contenedor ABS (UL94-HB)
- Certificados: ADR/RID (transporte terrestre y ferroviario), IMDG Code (transporte marítimo) y FAA e IATA (transporte aéreo). Estándares: DIN 43539-T5 (descarga profunda), IEC61427-2005 e IEC60896-21/22-2004

Modelos Disponibles

	CAPACIDAD [Ah] –100% DOD (a 1.75V/celda) a 25°C		
Modelo	**C20**	**C100**	**C120**
SOLARX-3	2.45	2.65	3.00
SOLARX-8	7.00	7.87	8.00
SOLARX-14	12.50	13.80	14.00
SOLARX-30	26.00	29.20	30.00
SOLARX-48	42.00	47.20	48.00
SOLARX-78	68.30	76.70	78.00
SOLARX-120	105	118	120
SOLARX-240	210	236	240

XU-29092I5-AH

El diseño y todas las fotos y dibujos de estas fichas están protegidas por Ley y no pueden ser distribuidas, ni reproducidas, ni total ni parcialmente, ni publicadas o usadas para ningún propósito sin el expreso consentimiento por escrito de XUNZEL.
©2015-2016 XUNZEL - Xunzel no se hace responsable de posibles errores tipográficos

Industry Leading Technology for Off-Grid, Off-Shore and Backup Power Applications

SOLAR WIND MARINE BOAT CARAVAN CAMPING UPS-BACKUP ENERGY PROFESSIONAL TELECOM CATV AGRICULTURAL WORKS TRAFFIC

xunzel™

info@xunzel.com
www.xunzel.com

© 2015-2016 Copyright Xunzel. Information contained in this document is subject to change without notice.

xunzel™

applied Solar and Wind Energy™

FICHA TÉCNICA

SOLARX™ Series

Especificaciones

MODELO		SOLARX-3	SOLARX-8	SOLARX-14	SOLARX-30	SOLARX-48	SOLARX-78	SOLARX-120	SOLARX-240
Celdas por unidad		6							
Voltaje nominal		12							
Capacidad (100% DOD, hasta 1.75V/celda @ 25°C) [Ah]									
C20		2.45	7.00	12.50	26.00	42.00	68.30	105	210
C100		2.65	7.87	13.80	29.20	47.20	76.70	118	236
C120		3.00	8.00	14.00	30.00	48.00	78.00	120	240
Peso [kg]		0.97	2.00	3.60	8.10	13.00	21.00	30.00	60.00
Dimensiones (mm)	L	178	151	151	166	198	350	328	522
	W	35	65	98	175	166	167	172	240
	H	67	100	95	125	171	180	222	223
Resistencia interna (Aprox.)		50 mΩ	30 mΩ	16 mΩ	10 mΩ	8 mΩ	6 mΩ	5 mΩ	4 mΩ
Auto-descarga		Las baterías SOLARX™ se pueden almacenar durante más de 6 meses a 25°C. Por favor, recargue la batería antes de usarla							
Rango de temperatura de funcionamiento	Descarga	-20°C~60°C							
	Carga	0°C~50°C							
	Almacenamiento	-20°C~60°C							
Rango de temperatura de funcionamiento normal		25°C~5°C							
Corriente de descarga máxima (5 segundos)		23A	70A	120A	260A	400A	650A	1000A	2000A
Corriente de carga máxima recomendada		0.69A	2.1A	3.6A	7.8A	12A	19.5A	30A	60A
Voltaje de Absorción	@ 25°C	14.4V – 14.7V							
Voltaje de Flotación	@ 25°C	13.7V – 13.9V				13.6V - 13.8V			
Voltaje de Ecualización	@ 25°C	14.6V – 14.8V							
Efecto de la temperatura en el voltaje de carga		-3mV/°K por celda				-4mV/°K por celda			
Terminales		Faston F1	Faston F2		M5	M6		M8	
Material del contenedor		A.B.S. (UL94-HB), nivel de inflamabilidad UL94-V2 disponible bajo pedido				A.B.S. (UL94-HB), nivel de inflamabilidad UL94-V1 disponible bajo pedido			
Cubierta		Polipropileno de alta densidad							
Asas		No disponible				Incorporadas	Plástico moldeado con cuerda (x2)	Correa	Plástico moldeado con cuerda (x2)

Efecto de la temperatura en la capacidad

Coeficientes de capacidad dependiendo de la temperatura

Tipo de bateria		-20°C	-10°C	0°C	5°C	10°C	20°C	25°C	30°C	40°C	45°C
Baterías AGM	12V	46%	66%	76%	83%	90%	98%	100%	104%	107%	109%

XU-290915-AH

El diseño y todas las fotos y dibujos de estas fichas están protegidas por Ley y no pueden ser distribuidas, ni reproducidas, ni total ni parcialmente, ni publicadas o usadas para ningún propósito sin el expreso consentimiento por escrito de XUNZEL.
©2015-2016 XUNZEL - Xunzel no se hace responsable de posibles errores tipográficos

xunzel™
info@xunzel.com
www.xunzel.com

© 2015-2016 Copyright Xunzel. Information contained in this document is subject to change without notice.

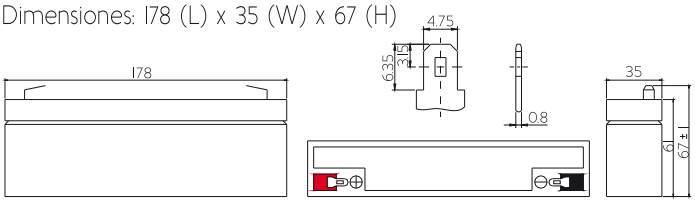
xunzel™

applied Solar and Wind Energy™

FICHA TÉCNICA

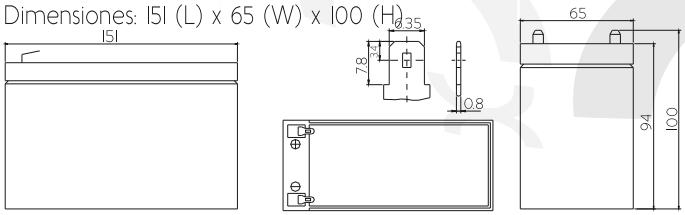
SOLARX-3

Unidades: mm
Dimensiones: 178 (L) x 35 (W) x 67 (H)



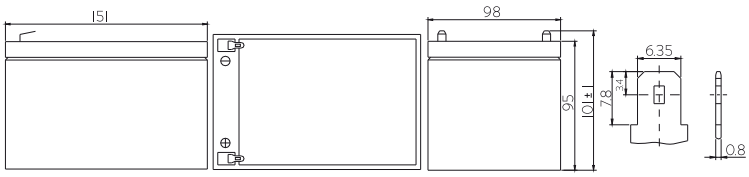
SOLARX-8

Unidades: mm
Dimensiones: 151 (L) x 65 (W) x 100 (H)



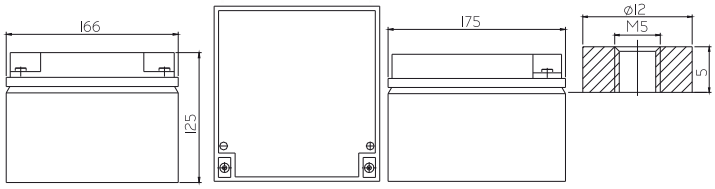
SOLARX-14

Unidades: mm
Dimensiones: 151 (L) x 98 (W) x 95 (H)



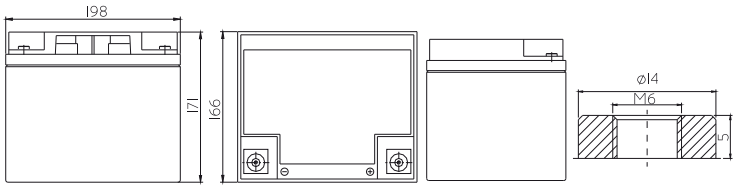
SOLARX-30

Unidades: mm
Dimensiones: 166 (L) x 175 (W) x 125 (H)



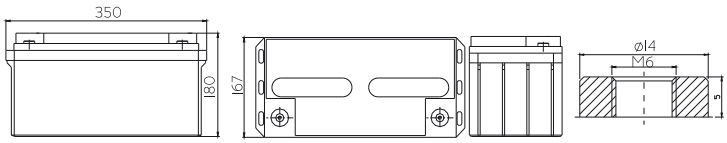
SOLARX-48

Unidades: mm
Dimensiones: 198 (L) x 166 (W) x 171 (H)



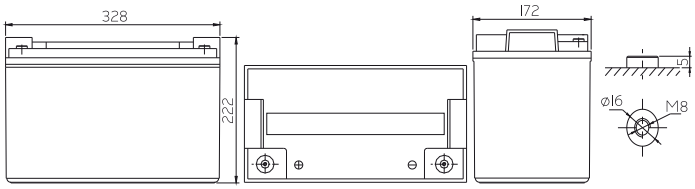
SOLARX-78

Unidades: mm
Dimensiones: 350 (L) x 167 (W) x 180 (H)



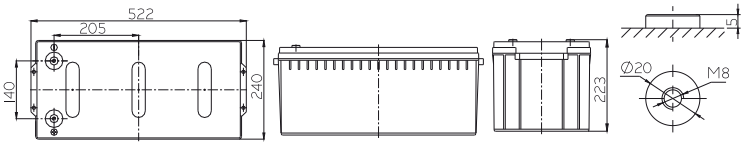
SOLARX-120

Unidades: mm
Dimensiones: 328 (L) x 172 (W) x 222 (H)



SOLARX-240

Unidades: mm
Dimensiones: 522 (L) x 240 (W) x 223 (H)



XU-2909215-AH

El diseño y todas las fotos y dibujos de estas fichas están protegidas por Ley y no pueden ser distribuidas, ni reproducidas, ni total ni parcialmente, ni publicadas o usadas para ningún propósito sin el expreso consentimiento por escrito de XUNZEL.
©2015-2016 XUNZEL - Xunzel no se hace responsable de posibles errores tipográficos

xunzel™
info@xunzel.com
www.xunzel.com

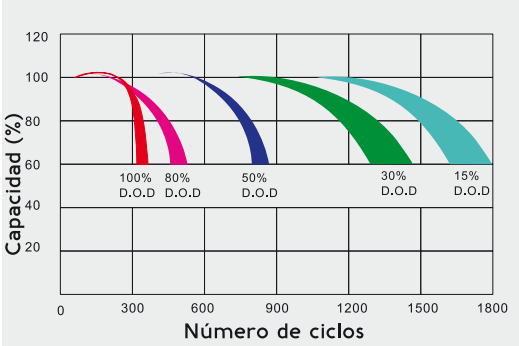
© 2015-2016 Copyright Xunzel. Information contained in this document is subject to change without notice.

xunzel™

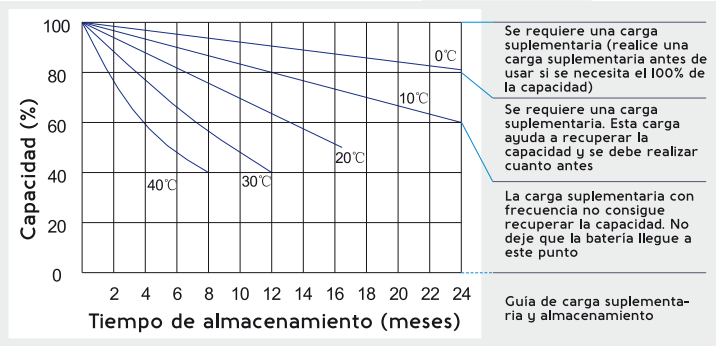
applied Solar and Wind Energy™

FICHA TÉCNICA

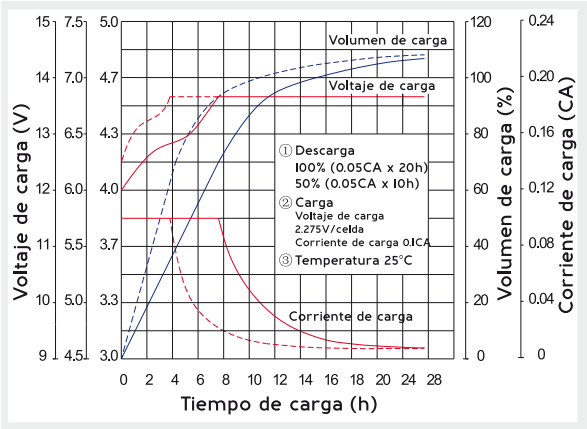
Característica de vidas cíclicas



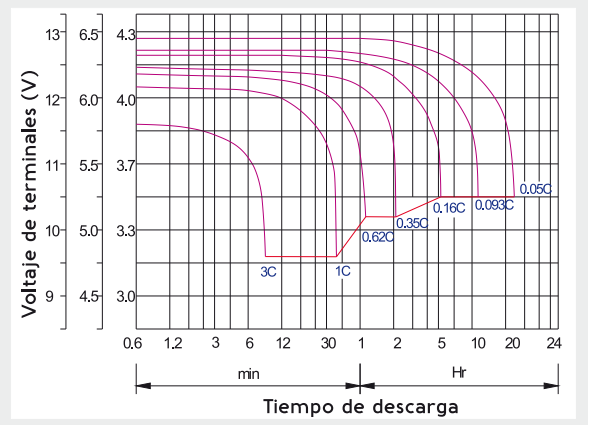
Característica de almacenamiento



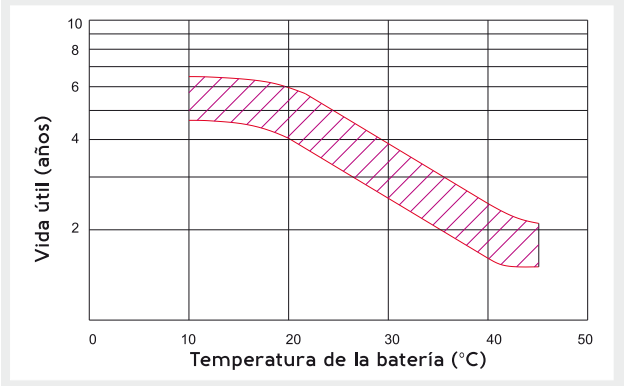
Característica de carga para uso cíclico



Característica de descarga



Efecto de la temperatura en la vida de flotación a largo plazo



Corriente de descarga vs. Voltaje de descarga

Voltaje de descarga final (V/celúla)	1.75 V	1.70 V	1.60 V
Corriente de descarga (A)	(A) ≤ 0.2C	0.2cC < (A) < 1.0C	(A) ≥ 0.2C

Voltaje constante	-0.2C x 2h + 2.4 ~ 2.45V/celda x 24h. Corriente máx. 0.3CA
Corriente constante	-0.2C x 2h + 0.1 CA x 12h
Carga rápida	-0.2C x 2h + 0.3CA x 4.0h

Cargue las baterías por lo menos cada 6 meses si se almacenan a 25°C

Mantenimiento y advertencias

- Evite descargas demasiado profundas, sobre todo en conexiones en serie
- Cargar la batería con voltajes apropiados garantiza que la batería se recargue completamente. En general, la capacidad de carga debe ser 1.1-1.15 veces la capacidad de descarga
- La vida útil de la batería depende de muchos factores: la profundidad de descarga (DOD, la más significante), la temperatura, el ratio de descarga, el método de carga....
- Se recomienda verificar el voltaje, la temperatura y las conexiones de la batería cada 6 meses

CE

XU-290915-AH

El diseño y todas las fotos y dibujos de estas fichas están protegidas por Ley y no pueden ser distribuidas, ni reproducidas, ni total ni parcialmente, ni publicadas o usadas para ningún propósito sin el expreso consentimiento por escrito de XUNZEL.
©2015-2016 XUNZEL - Xunzel no se hace responsable de posibles errores tipográficos

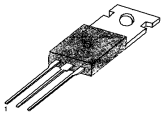
© 2015-2016 Copyright Xunzel. Information contained in this document is subject to change without notice.

LM78XX (KA78XX, MC78XX) FIXED VOLTAGE REGULATOR (POSITIVE)

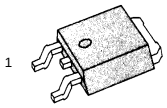
3-TERMINAL 1A POSITIVE VOLTAGE REGULATORS

The LM78XX series of three-terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

TO-220



D-PAK



1: Input 2: GND 3: Output

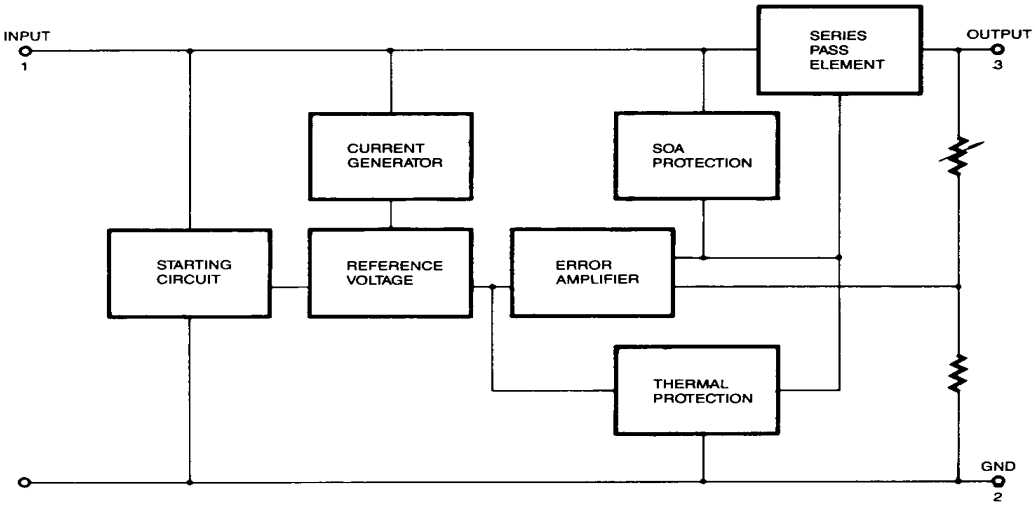
FEATURES

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 11, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor SOA Protection

ORDERING INFORMATION

Device	Output Voltage Tolerance	Packag	Operating Temperature
KA78XXCT	± 4%	TO-220	0 ~ +125 °C
KA78XXAT	± 2%		
KA78XXIT	± 4%		-40 ~ +125 °C
KA78XXR		D-PAK	0 ~ +125 °C
KA78XXAR	± 2%		
KA78XXIR	± 4%		-40 ~ +125 °C

BLOCK DIAGRAM



Rev. B

FAIRCHILD SEMICONDUCTOR™

©1999 Fairchild Semiconductor Corporation

ABSOLUTE MAXIMUM RATINGS (T_A = +25°C, unless otherwise specified)

Characteristic	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air	$R_{\theta JA}$	65	°C/W
Operating Temperature Range KA78XX/A/R/RA KA78XXI/RI	T_{OPR}	0 ~ +125 -40 ~ +125	°C °C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C

LM7805/I/R/RI ELECTRICAL CHARACTERISTICS

(Refer to test circuit, $T_{MIN} < T_J < T_{MAX}$, I_O = 500mA, V_I = 10V, C_I= 0.33µF, C_O= 0.1µF, unless otherwise specified)

Characteristic	Symbol	Test Conditions		LM7805I			LM7805			Unit
				Min	Typ	Max	Min	Typ	Max	
Output Voltage	V_O	T_J =+25 °C		4.8	5.0	5.2	4.8	5.0	5.2	V
		5.0mA ≤ I_O ≤1.0A, P_O ≤ 15W V_I = 7V to 20V V_I = 8V to 20V		4.75	5.0	5.25	4.75	5.0	5.25	
Line Regulation	ΔV_O	T_J=+25°C	V_O = 7V to 25V		4.0	100		4.0	100	mV
			V_I = 8V to 12V		1.6	50		1.6	50	
Load Regulation	ΔV_O	T_J=+25°C	I_O = 5.0mA to1.5A		9	100		9	100	mV
			I_O =250mA to 750mA		4	50		4	50	
Quiescent Current	I_Q	T_J =+25 °C			5.0	8		5.0	8	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A			0.03	0.5		0.03	0.5	mA
		V_I= 7V to 25V						0.3	1.3	
		V_I= 8V to 25V			0.3	1.3				
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O= 5mA			-0.8			-0.8		mV/ °C
Output Noise Voltage	V_N	f = 10Hz to 100Khz, T_A=+25 °C			42			42		µV/Vo
Ripple Rejection	RR	f = 120Hz V_O = 8 to 18V		62	73		62	73		dB
Dropout Voltage	V_O	I_O = 1A, T_J =+25 °C			2			2		V
Output Resistance	R_O	f = 1KHz			15			15		mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A =+25 °C			230			230		mA
Peak Current	I_{PK}	T_J =+25 °C			2.2			2.2		A

* $T_{MIN} < T_J < T_{MAX}$
LM78XXI/RI: T_{MIN}= - 40 °C, T_{MAX} = +125 °C
LM78XX/R: T_{MIN}= 0 °C, T_{MAX}= +125 °C
* Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

LM7809/I/R/RI ELECTRICAL CHARACTERISTICS

(Refer to test circuit. $T_{MIN} < T_J < T_{MAX}$, I_O= 500mA, V_I= 15V, C_I = 0.33µF, C_O = 0.1µF. unless otherwise specified)

Characteristic	Symbol	Test Conditions		LM7809I			LM7809			Unit
				Min	Typ	Max	Min	Typ	Max	
Output Voltage	V_O	T_J =+25 °C		8.65	9	9.35	8.65	9	9.35	V
		5.0mA ≤ I_O ≤1.0A, P_D ≤15W V_I= 11.5V to 24V V_I = 12.5V to 24V		8.6	9	9.4	8.6	9	9.4	
Line Regulation	ΔV_O	T_J=+25 °C	V_I = 11.5V to 25V		6	180		6	180	mV
			V_I = 12V to 25v		2	90		2	90	
Load Regulation	ΔV_O	T_J=+25 °C	I_O = 5mA to 1.5A		12	180		12	180	mV
			I_O = 250mA to 750mA		4	90		4	90	
Quiescent Current	I_Q	T_J=+25 °C			5.0	8		5.0	8	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A				0.5			0.5	mA
		V_I = 11.5V to 26V							1.3	
		V_I = 12.5V to 26V				1.3				
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O = 5mA			-1			-1		mV/ °C
Output Noise Voltage	V_N	f = 10Hz to 100Khz, T_A =+25 °C			58			58		µV/V_O
Ripple Rejection	RR	f = 120Hz V_I = 13V to 23V		56	71		56	71		dB
Dropout Voltage	V_D	I_O = 1A, T_J=+25 °C			2			2		V
Output Resistance	R_O	f = 1KHz			17			17		mΩ
Short Circuit Current	I_{SC}	V_I= 35V, T_A =+25 °C			250			250		mA
Peak Current	I_{PK}	T_J= +25 °C			2.2			2.2		A

* $T_{MIN} < T_J < T_{MAX}$
  LM78XXI/RI: T_{MIN}= - 40 °C, T_{MAX} = +125 °C
  LM78XX/R: T_{MIN}= 0 °C, T_{MAX}= +125 °C
* Load and line regulation are specified at constant, junction temperature. Change in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

LM7812/I/R/RI ELECTRICAL CHARACTERISTICS

(Refer to test circuit, $T_{MIN} < T_J < T_{MAX}$, I_O=500mA, V_I=19V, C_I= 0.33µF, C_O= 0.1.µF, unless otherwise specified)

Characteristic	Symbol	Test Conditions		LM7812I			LM7812			Unit
				Min	Typ	Max	Min	Typ	Max	
Output Voltage	V_O	T_J =+25 °C		11.5	12	12.5	11.5	12	12.5	V
		5.0mA ≤ I_O≤1.0A, P_D≤15W V_I = 14.5V to 27V V_I= 15.5V to 27V		11.4	12	12.6	11.4	12	12.6	
Line Regulation	ΔV_O	T_J =+25°C	V_I = 14.5V to 30V		10	240		10	240	mV
			V_I = 16V to 22V		3.0	120		3.0	120	
Load Regulation	ΔV_O	T_J =+25°C	I_O = 5mA to 1.5A		11	240		11	240	mV
			I_O = 250mA to 750mA		5.0	120		5.0	120	
Quiescent Current	I_Q	T_J =+25 °C			5.1	8		5.1	8	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A			0.1	0.5		0.1	0.5	mA
		V_I = 14.5V to 30V						0.5	1.0	
		V_I = 15V to 30V				1.0				
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O = 5mA		0.5	-1			-1		mV/ °C
Output Noise Voltage	V_N	f = 10Hz to 100Khz, T_A =+25 °C			76			76		mV/V_O
Ripple Rejection	RR	f = 120Hz V_I = 15V to 25V		55	71		55	71		dB
Dropout Voltage	V_D	I_O = 1A, T_J=+25 °C			2			2		V
Output Resistance	R_O	f = 1KHz			18			18		mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A=+25 °C			230			230		mA
Peak Current	I_{PK}	T_J = +25 °C			2.2			2.2		A

$T_{MIN} < T_J < T_{MAX}$
LM78XXI/RI: T_{MIN}= - 40 °C, T_{MAX} = +125 °C
LM78XX/R: T_{MIN}= 0 °C, T_{MAX}= +125 °C
* Load and line regulation are specified at constant, junction temperature. Change in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

TYPICAL PERFORMANCE CHARACTERISTICS

Fig. 1 Quiescent Current

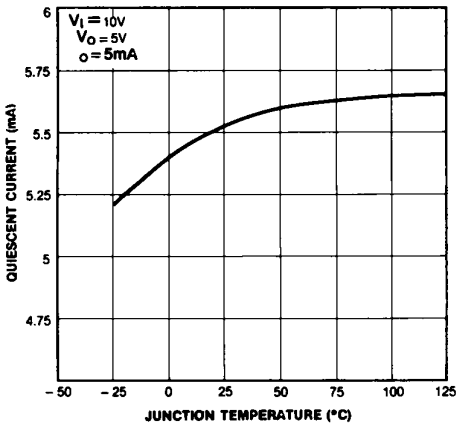


Fig. 2 Peak Output Current

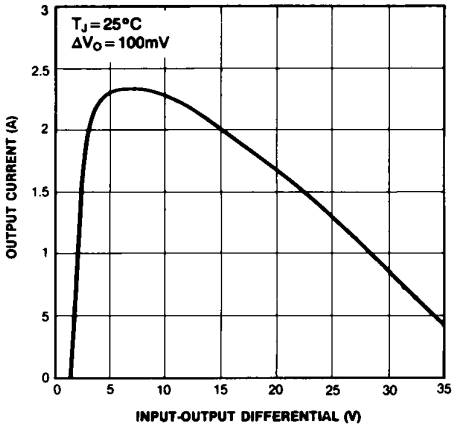


Fig. 3 Output Voltage

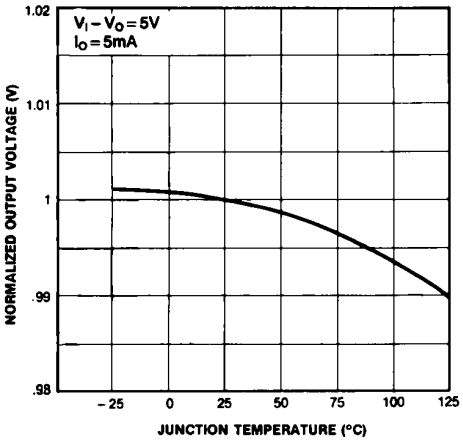
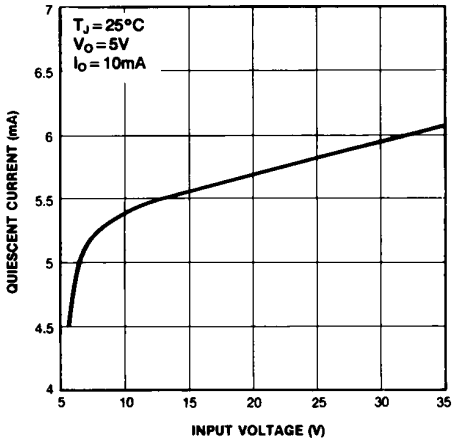


Fig. 4 Quiescent Current



TYPICAL APPLICATIONS

Fig. 5 DC Parameters

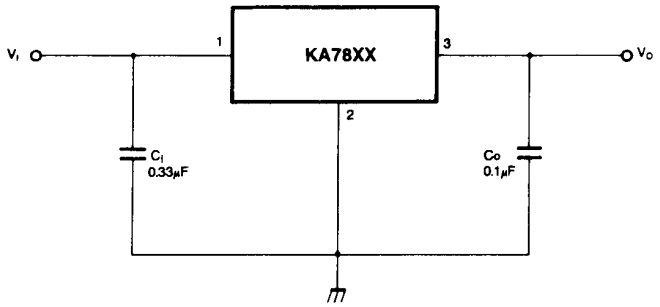


Fig. 6 Load Regulation

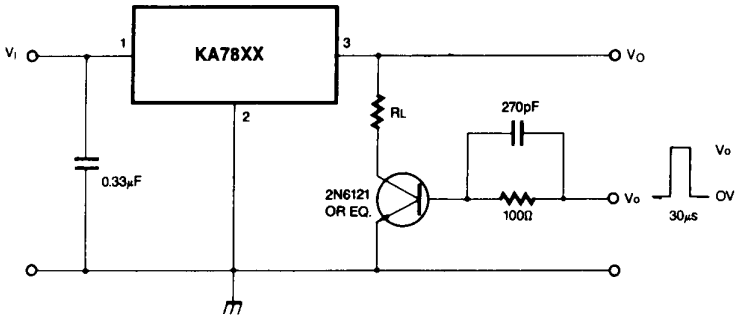
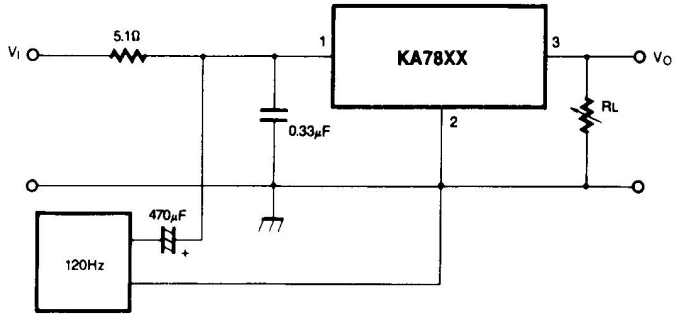


Fig. 7 Ripple Rejection



FAIRCHILD SEMICONDUCTOR™

TYPICAL APPLICATIONS (Continued)

Fig. 8 Fixed Output Regulator

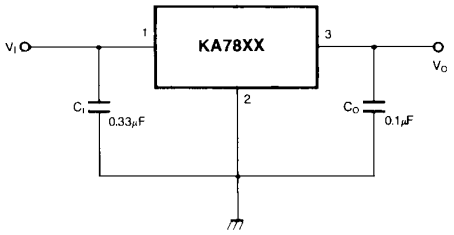
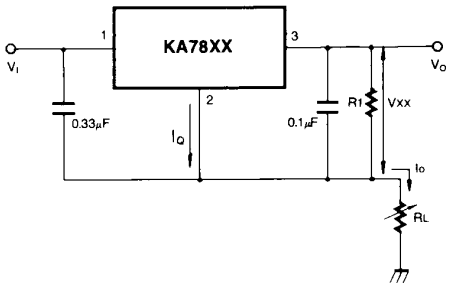


Fig. 9 Constant Current Regulator



$$I_O = \frac{\quad}{R_1} + I_Q$$

Notes:

(1) To specify an output voltage. substitute voltage value for "XX." A common ground is required between the input and the Output voltage. The input voltage must remain typically 2.0V above the output voltage even during the low point on the input ripple voltage.

(2) C_I is required if regulator is located an appreciable distance from power Supply filter.

(3) C_O improves stability and transient response.

Fig. 10 Circuit for Increasing Output Voltage

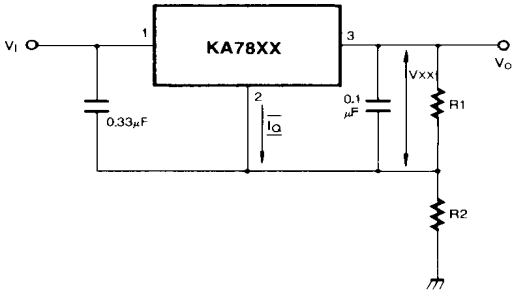
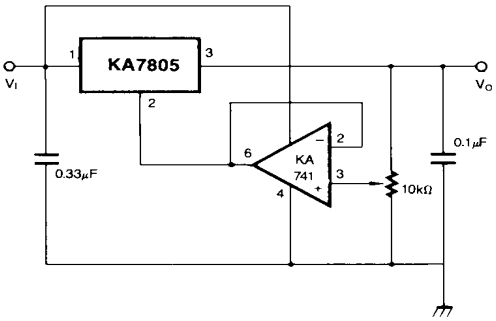


Fig. 11 Adjustable Output Regulator (7 to 30V)



$I_{RI} \geq 5\ I_Q$

$V_O = V_{XX}(1+R_2/R_1)+I_QR_2$

TYPICAL APPLICATIONS (Continued)

Fig. 12 High Current Voltage Regulator

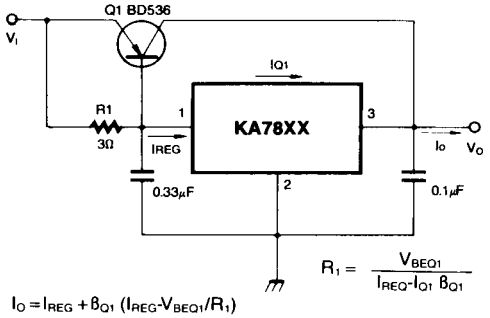


Fig. 13 High Output Current with Short Circuit Protection

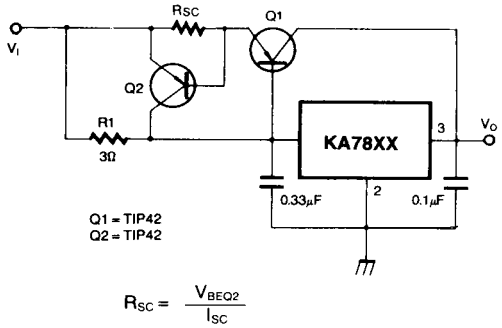


Fig. 14 Tracking Voltage Regulator

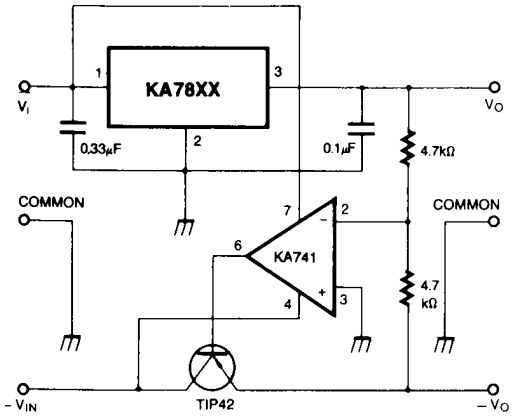
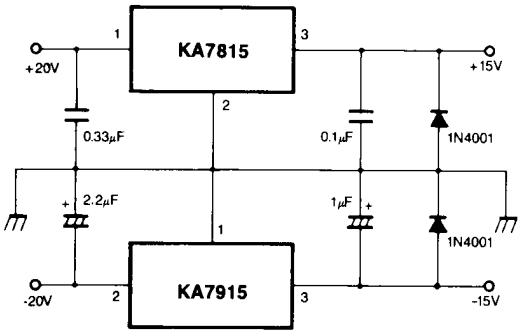


Fig. 15 Split Power Supply (±15V-1A)



TYPICAL APPLICATIONS (Continued)

Fig. 16 Negative Output Voltage Circuit

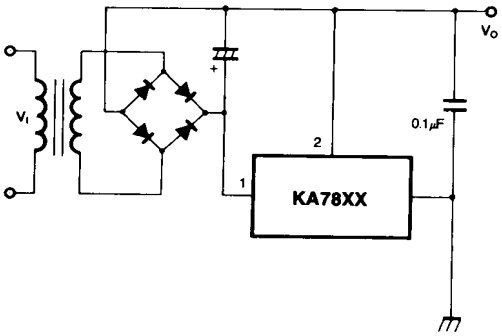
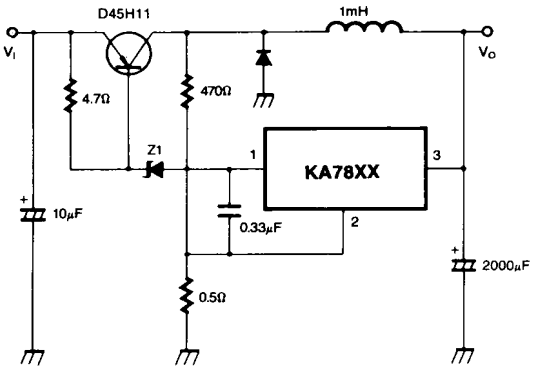


Fig. 17 switching Regulator



TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™
CoolFET™
CROSSVOLT™
E^2CMOS™
FACT™
FACT Quiet Series™
FAST®
FASTr™
GTO™
HiSeC™
ISOPLANAR™
MICROWIRE™
POP™
PowerTrench™
QS™
Quiet Series™
SuperSOT™-3
SuperSOT™-6
SuperSOT™-8
TinyLogic™

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.
As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

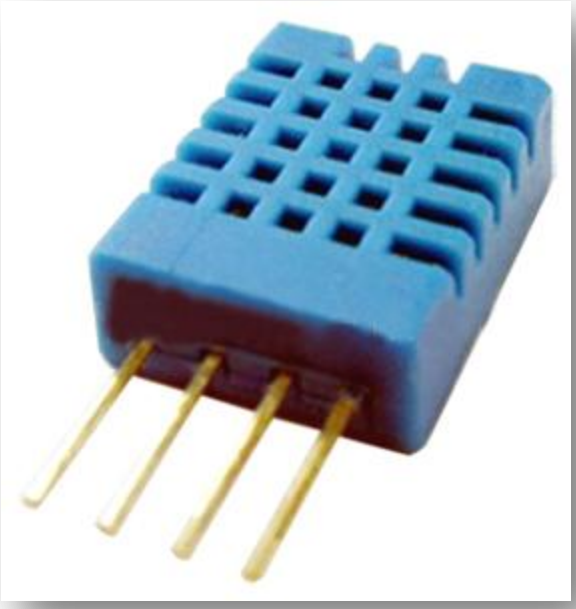
DHT11 Humidity & Temperature Sensor

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output.

DHT 11 Humidity & Temperature Sensor

1. Introduction

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.



Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5％RH	±2℃	1	4 Pin Single Row

Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

3. Typical Application (Figure 1)

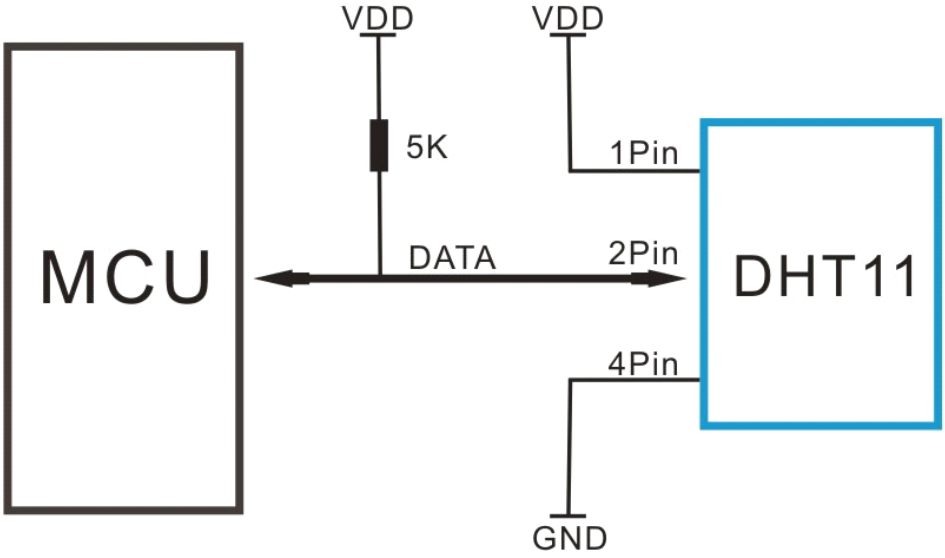


Figure 1 Typical Application

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

When the connecting cable is shorter than 20 metres, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 metres, choose a appropriate pull-up resistor as needed.

4. Power and Pin

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

5. Communication Process: Serial Interface (Single-Wire Two-Way)

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.

Data consists of decimal and integral parts. A complete data transmission is **40bit**, and the sensor sends **higher data bit** first.
Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

5.1 Overall Communication Process (Figure 2, below)

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

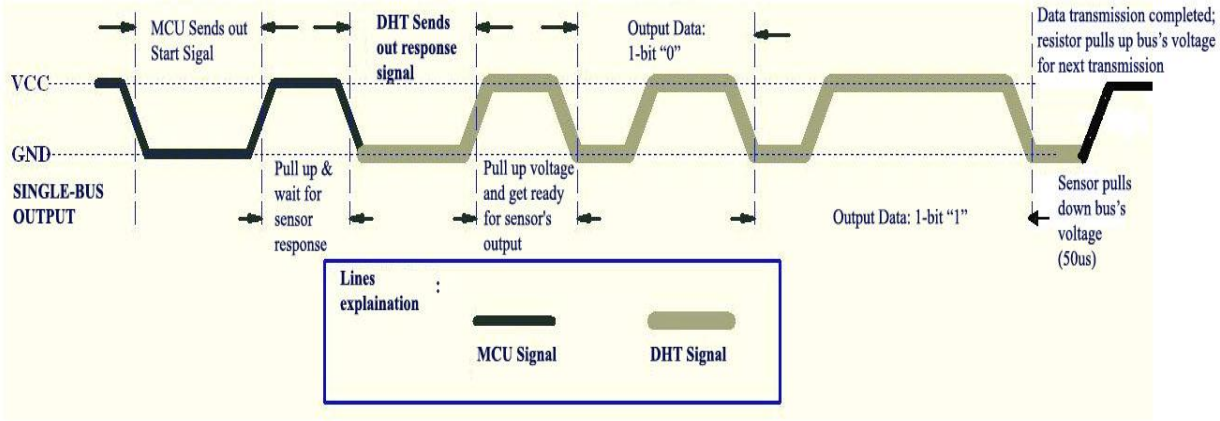


Figure 2 Overall Communication Process

5.2 MCU Sends out Start Signal to DHT (Figure 3, below)

Data Single-bus free status is at high voltage level. When the communication between MCU and DHT11 begins, the programme of MCU will set Data Single-bus voltage level from high to low and this process must take at least 18ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40us for DHT's response.

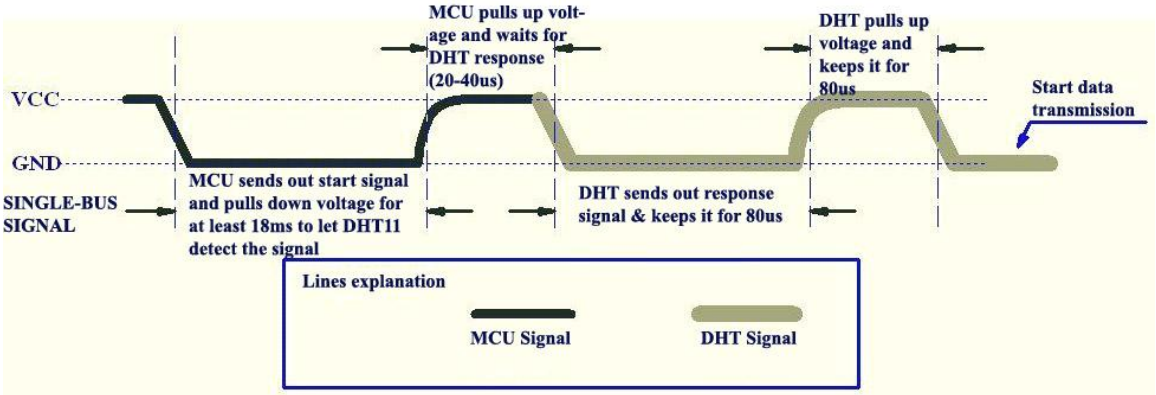


Figure 3 MCU Sends out Start Signal & DHT Responses

5.3 DHT Responses to MCU (Figure 3, above)

Once DHT detects the start signal, it will send out a low-voltage-level response signal, which lasts 80us. Then the programme of DHT sets Data Single-bus voltage level from low to high and keeps it for 80us for DHT's preparation for sending data.

When DATA Single-Bus is at the low voltage level, this means that DHT is sending the response signal. Once DHT sent out the response signal, it pulls up voltage and keeps it for 80us and prepares for data transmission.

When DHT is sending data to MCU, every bit of data begins with the 50us low-voltage-level and the length of the following high-voltage-level signal determines whether data bit is "0" or "1" (see Figures 4 and 5 below).

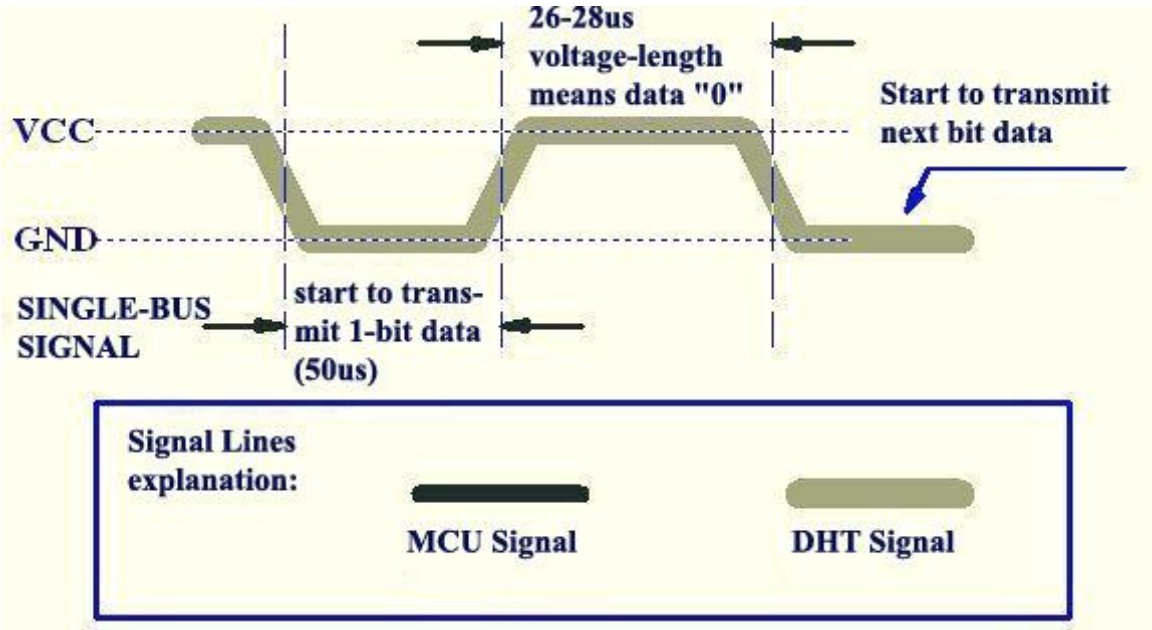


Figure 4 Data "0" Indication

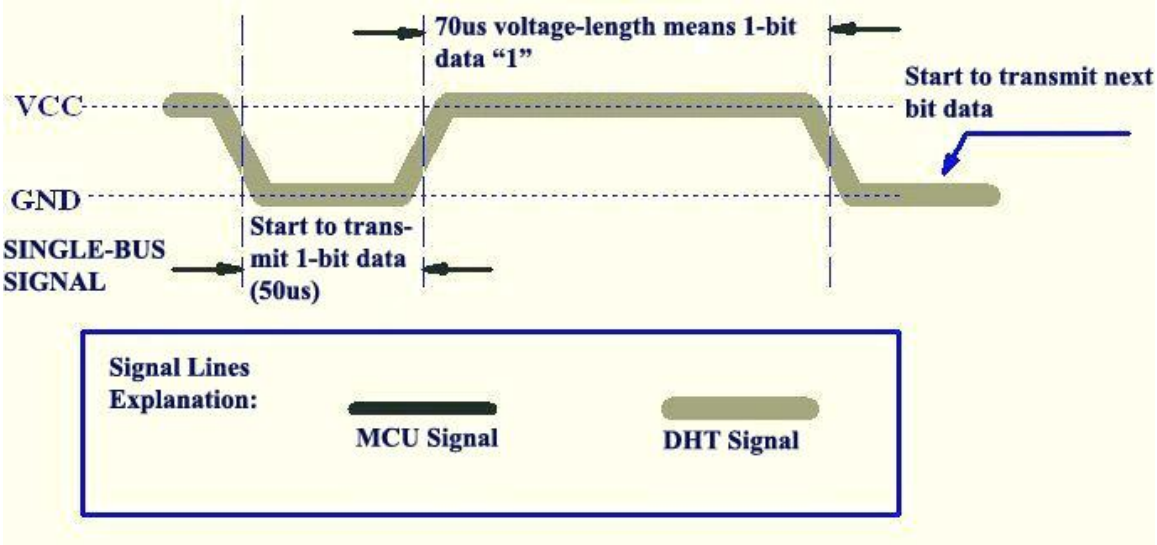


Figure 5 Data "1" Indication

If the response signal from DHT is always at high-voltage-level, it suggests that DHT is not responding properly and please check the connection. When the last bit data is transmitted, DHT11 pulls down the voltage level and keeps it for 50us. Then the Single-Bus voltage will be pulled up by the resistor to set it back to the free status.

6. Electrical Characteristics

VDD=5V, T = 25℃ (unless otherwise stated)

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

Note: Sampling period at intervals should be no less than 1 second.

7. Attentions of application

(1) Operating conditions

Applying the DHT11 sensor beyond its working range stated in this datasheet can result in 3%RH signal shift/discrepancy. The DHT11 sensor can recover to the calibrated status gradually when it gets back to the normal operating condition and works within its range. Please refer to (3) of

this section to accelerate its recovery. Please be aware that operating the DHT11 sensor in the non-normal working conditions will accelerate sensor's aging process.

(2) Attention to chemical materials

Vapor from chemical materials may interfere with DHT's sensitive-elements and debase its sensitivity. A high degree of chemical contamination can permanently damage the sensor.

(3) Restoration process when (1) & (2) happen

Step one: Keep the DHT sensor at the condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;
Step two:K keep the DHT sensor at the condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Temperature Affect

Relative humidity largely depends on temperature. Although temperature compensation technology is used to ensure accurate measurement of RH, it is still strongly advised to keep the humidity and temperature sensors working under the same temperature. DHT11 should be mounted at the place as far as possible from parts that may generate heat.

(5) Ligh t Affect

Long time exposure to strong sunlight and ultraviolet may debase DHT's performance.

(6) Connection wires

The quality of connection wires will affect the quality and distance of communication and high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius and contact should take less than 10 seconds.
* Avoid using the sensor under dew condition.
* Do not use this product in safety or emergency stop devices or any other occasion that failure of DHT11 may cause personal injury.
* Storage: Keep the sensor at temperature 10-40℃, humidity <60%RH.

Disclaimer

This is a translated version of the manufacturer's data sheet. OSEPP is not responsible for the accuracy of the translated information.

Small, Low Power, 3-Axis ±3 *g* Accelerometer

ADXL335

FEATURES

3-axis sensing
Small, low profile package
4 mm × 4 mm × 1.45 mm LFCSP
Low power : 350 µA (typical)
Single-supply operation: 1.8 V to 3.6 V
10,000 *g* shock survival
Excellent temperature stability
BW adjustment with a single capacitor per axis
RoHS/WEEE lead-free compliant

APPLICATIONS

Cost sensitive, low power, motion- and tilt-sensing applications
Mobile devices
Gaming systems
Disk drive protection
Image stabilization
Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 *g*. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_X, C_Y, and C_Z capacitors at the X_{OUT}, Y_{OUT}, and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm × 4 mm × 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

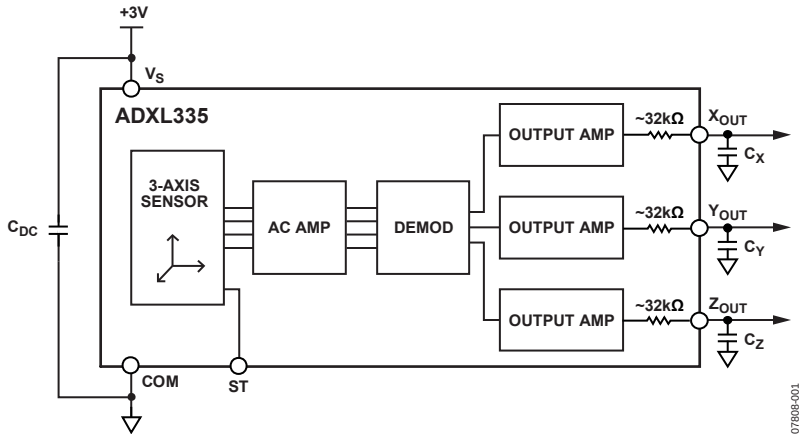


Figure 1.

Rev. B
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 **www.analog.com**
Fax: 781.461.3113 **©2009–2010 Analog Devices, Inc. All rights reserved.**

TABLE OF CONTENTS

Features ... 1
Applications ... 1
General Description ... 1
Functional Block Diagram ... 1
Revision History ... 2
Specifications ... 3
Absolute Maximum Ratings ... 4
ESD Caution ... 4
Pin Configuration and Function Descriptions ... 5
Typical Performance Characteristics ... 6
Theory of Operation ... 10
Mechanical Sensor ... 10
Performance ... 10
Applications Information ... 11
Power Supply Decoupling ... 11
Setting the Bandwidth Using C_X, C_Y, and C_Z ... 11
Self-Test ... 11
Design Trade-Offs for Selecting Filter Characteristics: The Noise/BW Trade-Off ... 11
Use with Operating Voltages Other Than 3 V ... 12
Axes of Acceleration Sensitivity ... 12
Layout and Design Recommendations ... 13
Outline Dimensions ... 14
Ordering Guide ... 14

REVISION HISTORY

1/10—Rev. A to Rev. B

Changes to Figure 21 ... 9

7/09—Rev. 0 to Rev. A

Changes to Figure 22 ... 9
Changes to Outline Dimensions ... 14

1/09—Revision 0: Initial Version

SPECIFICATIONS

T_A = 25°C, V_S = 3 V, C_X = C_Y = C_Z = 0.1 µF, acceleration = 0 *g*, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		*g*
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[1]			±1		%
SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	270	300	330	mV/*g*
Sensitivity Change Due to Temperature[3]	V_S = 3 V		±0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)					
0 *g* Voltage at X_{OUT}, Y_{OUT}	V_S = 3 V	1.35	1.5	1.65	V
0 *g* Voltage at Z_{OUT}	V_S = 3 V	1.2	1.5	1.8	V
0 *g* Offset vs. Temperature			±1		m*g*/°C
NOISE PERFORMANCE					
Noise Density X_{OUT}, Y_{OUT}			150		µ*g*/√Hz rms
Noise Density Z_{OUT}			300		µ*g*/√Hz rms
FREQUENCY RESPONSE[4]					
Bandwidth X_{OUT}, Y_{OUT}[5]	No external filter		1600		Hz
Bandwidth Z_{OUT}[5]	No external filter		550		Hz
R_{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST[6]					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		µA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	−150	−325	−600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V_S = 3 V		350		µA
Turn-On Time[7]	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		−40		+85	°C

[1] Defined as coupling between any two axes.
[2] Sensitivity is essentially ratiometric to V_S.
[3] Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.
[4] Actual frequency response controlled by user-supplied external filter capacitors (C_X, C_Y, C_Z).
[5] Bandwidth with external capacitors = 1/(2 × π × 32 kΩ × C). For C_X, C_Y = 0.003 µF, bandwidth = 1.6 kHz. For C_Z = 0.01 µF, bandwidth = 500 Hz. For C_X, C_Y, C_Z = 10 µF, bandwidth = 0.5 Hz.
[6] Self-test response changes cubically with V_S.
[7] Turn-on time is dependent on C_X, C_Y, C_Z and is approximately 160 × C_X or C_Y or C_Z + 1 ms, where C_X, C_Y, C_Z are in microfarads (µF).

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 *g*
Acceleration (Any Axis, Powered)	10,000 *g*
V_S	−0.3 V to +3.6 V
All Other Pins	(COM − 0.3 V) to (V_S + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	−55°C to +125°C
Temperature Range (Storage)	−65°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

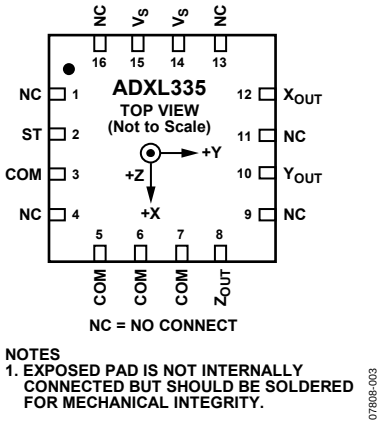


Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	NC	No Connect.[1]
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect.[1]
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z_{OUT}	Z Channel Output.
9	NC	No Connect.[1]
10	Y_{OUT}	Y Channel Output.
11	NC	No Connect. [1]
12	X_{OUT}	X Channel Output.
13	NC	No Connect. [1]
14	V_S	Supply Voltage (1.8 V to 3.6 V).
15	V_S	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect. [1]
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

[1] NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

TYPICAL PERFORMANCE CHARACTERISTICS

N > 1000 for all typical performance plots, unless otherwise noted.

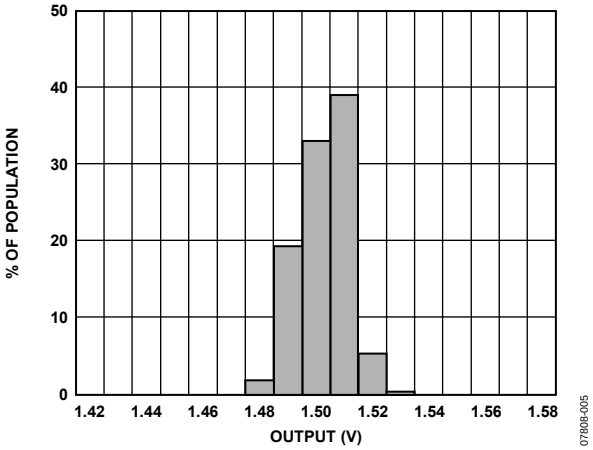


Figure 3. X-Axis Zero g Bias at 25°C, V_S = 3 V

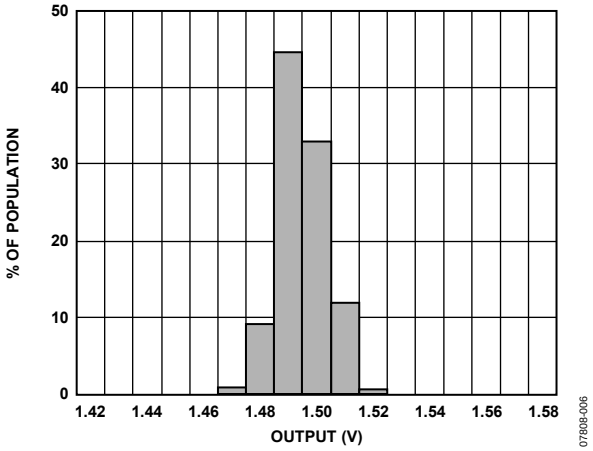


Figure 4. Y-Axis Zero g Bias at 25°C, V_S = 3 V

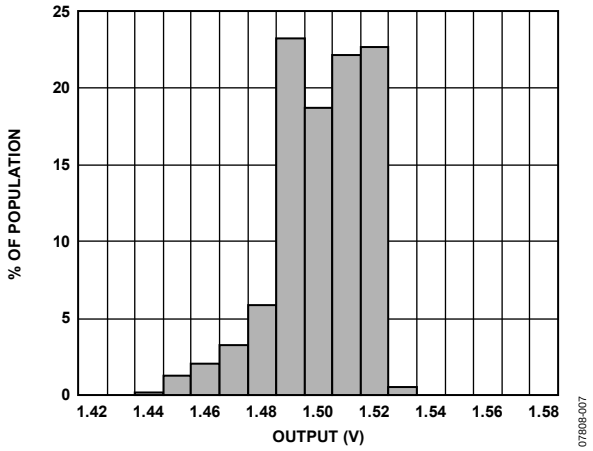


Figure 5. Z-Axis Zero g Bias at 25°C, V_S = 3 V

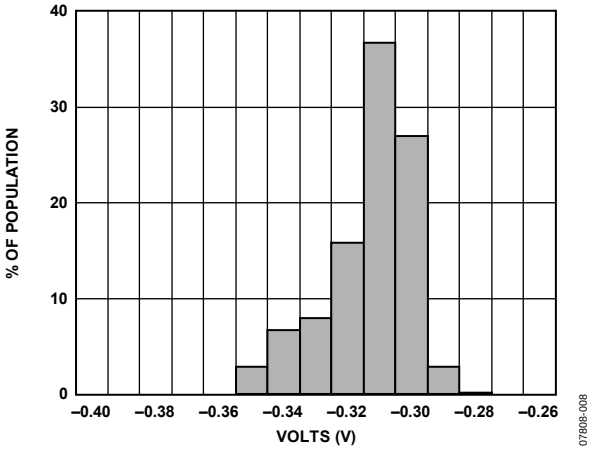


Figure 6. X-Axis Self-Test Response at 25°C, V_S = 3 V

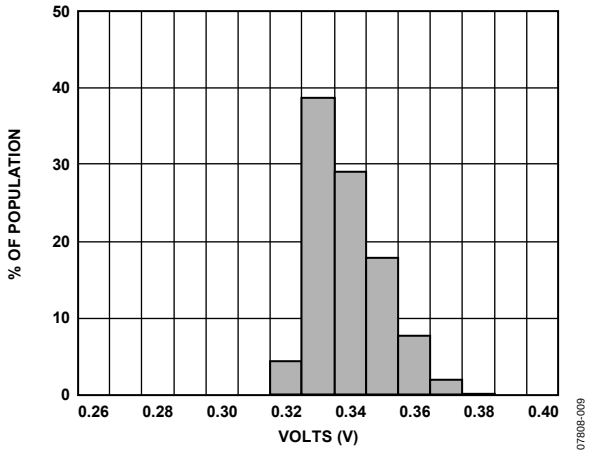


Figure 7. Y-Axis Self-Test Response at 25°C, V_S = 3 V

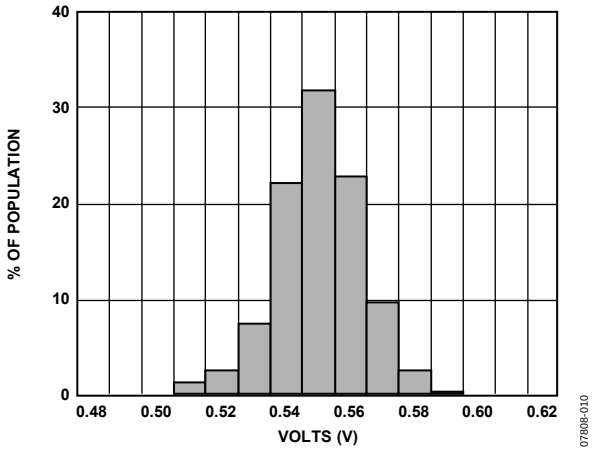


Figure 8. Z-Axis Self-Test Response at 25°C, V_S = 3 V

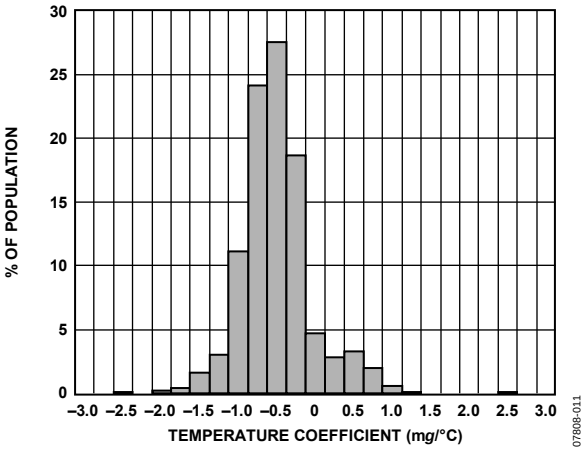


Figure 9. X-Axis Zero g Bias Temperature Coefficient, V_S = 3 V

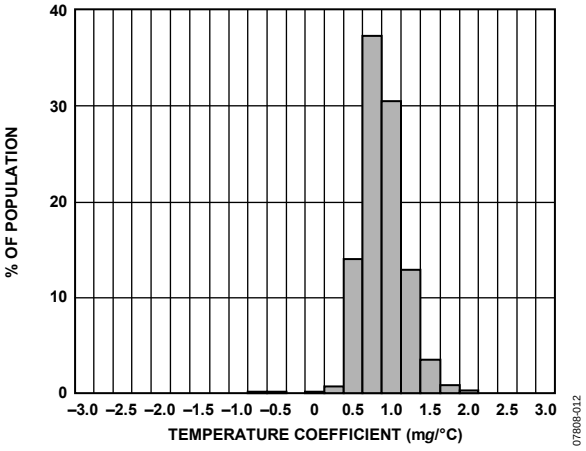


Figure 10. Y-Axis Zero g Bias Temperature Coefficient, V_S = 3 V

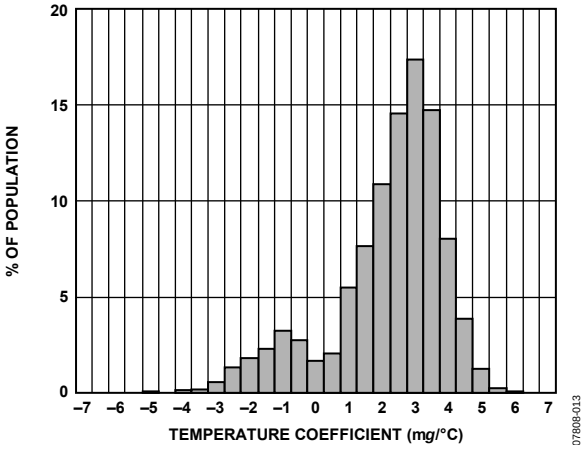


Figure 11. Z-Axis Zero g Bias Temperature Coefficient, V_S = 3 V

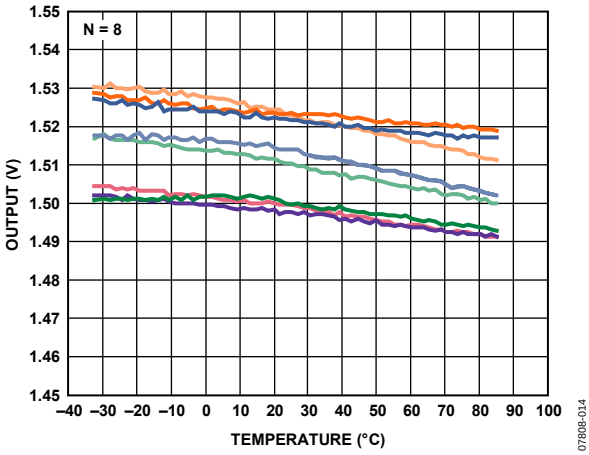


Figure 12. X-Axis Zero g Bias vs. Temperature—Eight Parts Soldered to PCB

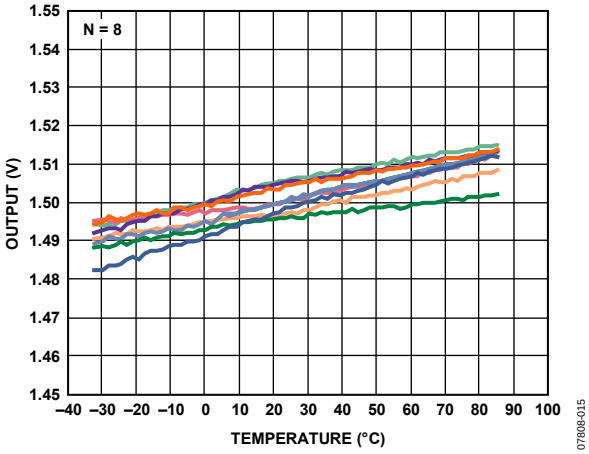


Figure 13. Y-Axis Zero g Bias vs. Temperature—Eight Parts Soldered to PCB

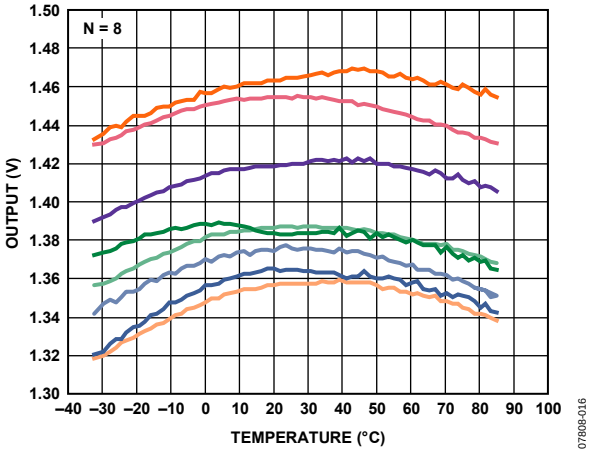


Figure 14. Z-Axis Zero g Bias vs. Temperature—Eight Parts Soldered to PCB

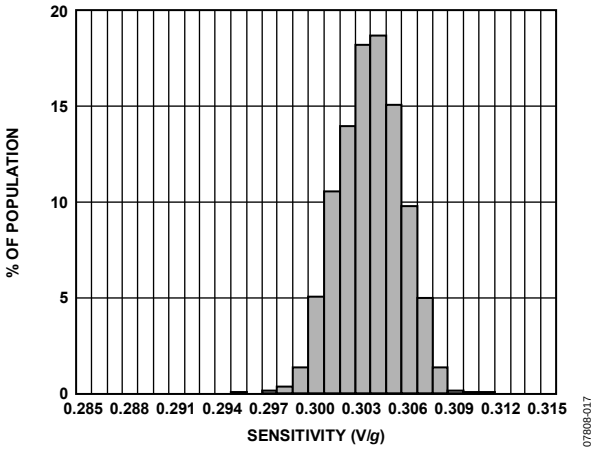


Figure 15. X-Axis Sensitivity at 25°C, V_S = 3 V

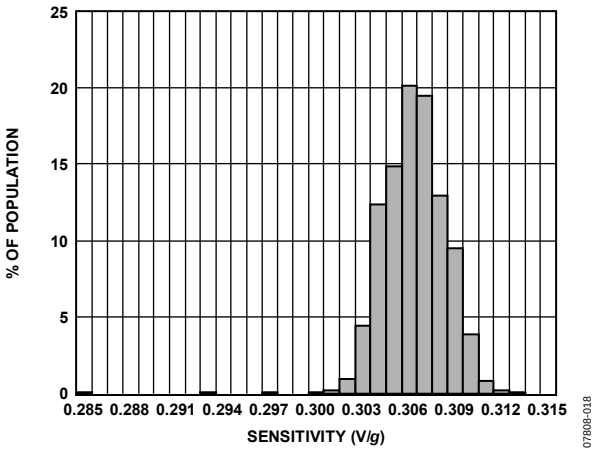


Figure 16. Y-Axis Sensitivity at 25°C, V_S = 3 V

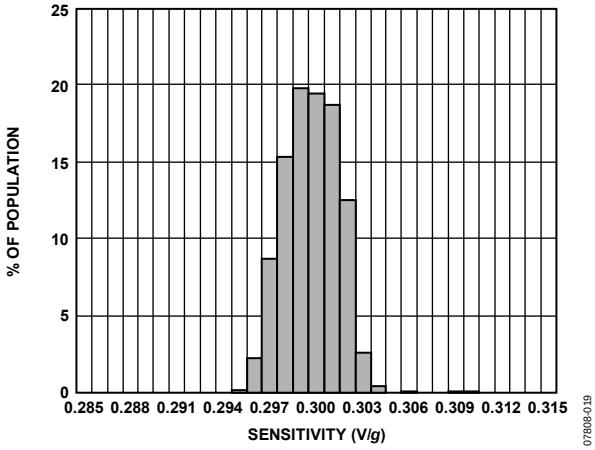


Figure 17. Z-Axis Sensitivity at 25°C, V_S = 3 V

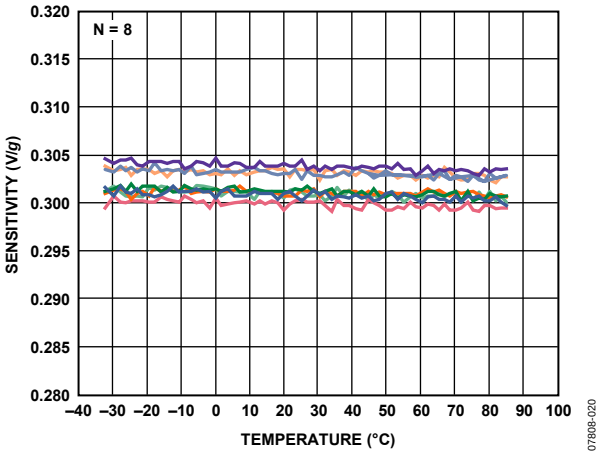


Figure 18. X-Axis Sensitivity vs. Temperature—Eight Parts Soldered to PCB, V_S = 3 V

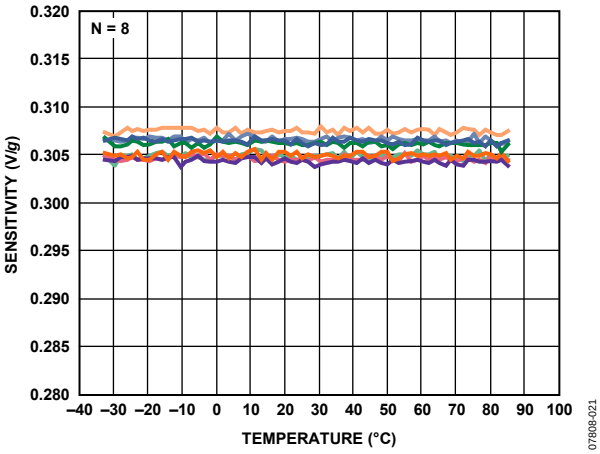


Figure 19. Y-Axis Sensitivity vs. Temperature—Eight Parts Soldered to PCB, V_S = 3 V

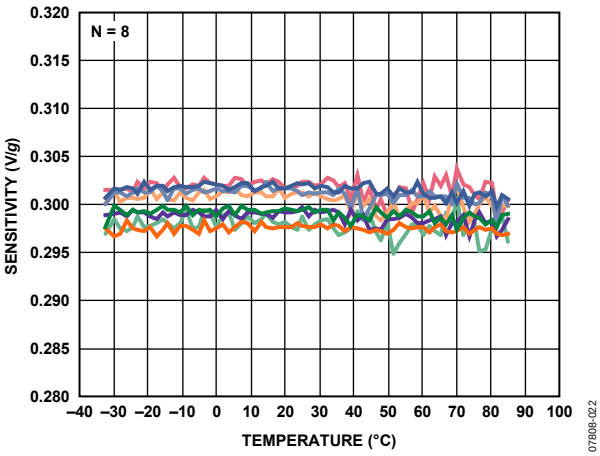


Figure 20. Z-Axis Sensitivity vs. Temperature—Eight Parts Soldered to PCB, V_S = 3 V

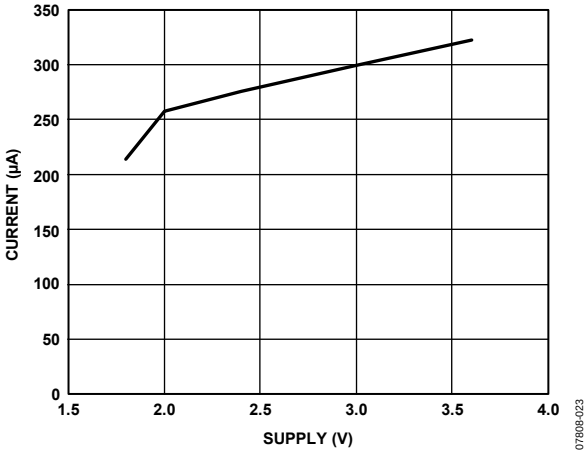


Figure 21. Typical Current Consumption vs. Supply Voltage

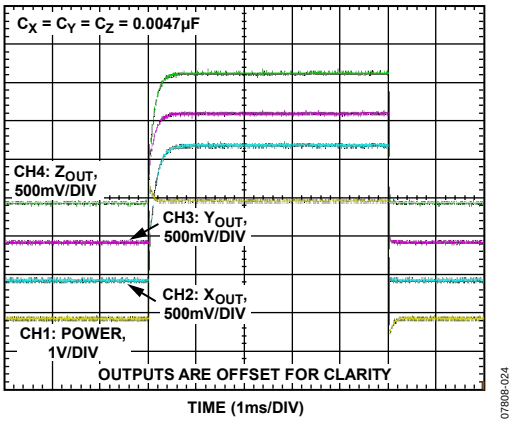


Figure 22. Typical Turn-On Time, V_S = 3 V

THEORY OF OPERATION

The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ±3 *g* minimum. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

The sensor is a polysilicon surface-micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

The demodulator output is amplified and brought off-chip through a 32 kΩ resistor. The user then sets the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

MECHANICAL SENSOR

The ADXL335 uses a single structure for sensing the X, Y, and Z axes. As a result, the three axes' sense directions are highly orthogonal and have little cross-axis sensitivity. Mechanical misalignment of the sensor die to the package is the chief source of cross-axis sensitivity. Mechanical misalignment can, of course, be calibrated out at the system level.

PERFORMANCE

Rather than using additional temperature compensation circuitry, innovative design techniques ensure that high performance is built in to the ADXL335. As a result, there is no quantization error or nonmonotonic behavior, and temperature hysteresis is very low (typically less than 3 m*g* over the −25°C to +70°C temperature range).

APPLICATIONS INFORMATION

POWER SUPPLY DECOUPLING

For most applications, a single 0.1 μF capacitor, C_{DC}, placed close to the ADXL335 supply pins adequately decouples the accelerometer from noise on the power supply. However, in applications where noise is present at the 50 kHz internal clock frequency (or any harmonic thereof), additional care in power supply bypassing is required because this noise can cause errors in acceleration measurement.

If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite bead can be inserted in the supply line. Additionally, a larger bulk bypass capacitor (1 μF or greater) can be added in parallel to C_{DC}. Ensure that the connection from the ADXL335 ground to the power supply ground is low impedance because noise transmitted through ground has a similar effect to noise transmitted through V_S.

SETTING THE BANDWIDTH USING C_X, C_Y, AND C_Z

The ADXL335 has provisions for band limiting the X_{OUT}, Y_{OUT}, and Z_{OUT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is

$$F_{-3\,dB} = 1/(2\pi(32\ k\Omega) \times C_{(X,\,Y,\,Z)})$$

or more simply

$$F_{-3\,dB} = 5\ \mu F/C_{(X,\,Y,\,Z)}$$

The tolerance of the internal resistor (R_{FILT}) typically varies as much as ±15% of its nominal value (32 kΩ), and the bandwidth varies accordingly. A minimum capacitance of 0.0047 μF for C_X, C_Y, and C_Z is recommended in all cases.

Table 4. Filter Capacitor Selection, C_X, C_Y, and C_Z

Bandwidth (Hz)	Capacitor (μF)
1	4.7
10	0.47
50	0.10
100	0.05
200	0.027
500	0.01

SELF-TEST

The ST pin controls the self-test feature. When this pin is set to V_S, an electrostatic force is exerted on the accelerometer beam. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output is −1.08 *g* (corresponding to −325 mV) in the X-axis, +1.08 *g* (or +325 mV) on the Y-axis, and +1.83 *g* (or +550 mV) on the Z-axis. This ST pin can be left open-circuit or connected to common (COM) in normal use.

Never expose the ST pin to voltages greater than V_S + 0.3 V. If this cannot be guaranteed due to the system design (for instance, if there are multiple supply voltages), then a low V_F clamping diode between ST and V_S is recommended.

DESIGN TRADE-OFFS FOR SELECTING FILTER CHARACTERISTICS: THE NOISE/BW TRADE-OFF

The selected accelerometer bandwidth ultimately determines the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor to improve the resolution of the accelerometer. Resolution is dependent on the analog filter bandwidth at X_{OUT}, Y_{OUT}, and Z_{OUT}.

The output of the ADXL335 has a typical bandwidth of greater than 500 Hz. The user must filter the signal at this point to limit aliasing errors. The analog bandwidth must be no more than half the analog-to-digital sampling frequency to minimize aliasing. The analog bandwidth can be further decreased to reduce noise and improve resolution.

The ADXL335 noise has the characteristics of white Gaussian noise, which contributes equally at all frequencies and is described in terms of μ*g*/√Hz (the noise is proportional to the square root of the accelerometer bandwidth). The user should limit bandwidth to the lowest frequency needed by the application to maximize the resolution and dynamic range of the accelerometer.

With the single-pole, roll-off characteristic, the typical noise of the ADXL335 is determined by

$$rms\ Noise = Noise\ Density \times (\sqrt{BW \times 1.6})$$

It is often useful to know the peak value of the noise. Peak-to-peak noise can only be estimated by statistical methods. Table 5 is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table 5. Estimation of Peak-to-Peak Noise

Peak-to-Peak Value	% of Time That Noise Exceeds Nominal Peak-to-Peak Value
2 × rms	32
4 × rms	4.6
6 × rms	0.27
8 × rms	0.006

USE WITH OPERATING VOLTAGES OTHER THAN 3 V

The ADXL335 is tested and specified at V_S = 3 V; however, it can be powered with V_S as low as 1.8 V or as high as 3.6 V. Note that some performance parameters change as the supply voltage is varied.

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At V_S = 3.6 V, the output sensitivity is typically 360 mV/*g*. At V_S = 2 V, the output sensitivity is typically 195 mV/*g*.

The zero *g* bias output is also ratiometric, thus the zero *g* output is nominally equal to $V_S/2$ at all supply voltages.

The output noise is not ratiometric but is absolute in volts; therefore, the noise density decreases as the supply voltage increases. This is because the scale factor (mV/*g*) increases while the noise voltage remains constant. At V_S = 3.6 V, the X-axis and Y-axis noise density is typically 120 μ*g*/√Hz, whereas at V_S = 2 V, the X-axis and Y-axis noise density is typically 270 μ*g*/√Hz.

Self-test response in *g* is roughly proportional to the square of the supply voltage. However, when ratiometricity of sensitivity is factored in with supply voltage, the self-test response in volts is roughly proportional to the cube of the supply voltage. For example, at V_S = 3.6 V, the self-test response for the ADXL335 is approximately −560 mV for the X-axis, +560 mV for the Y-axis, and +950 mV for the Z-axis.

At V_S = 2 V, the self-test response is approximately −96 mV for the X-axis, +96 mV for the Y-axis, and −163 mV for the Z-axis.

The supply current decreases as the supply voltage decreases. Typical current consumption at V_S = 3.6 V is 375 μA, and typical current consumption at V_S = 2 V is 200 μA.

AXES OF ACCELERATION SENSITIVITY

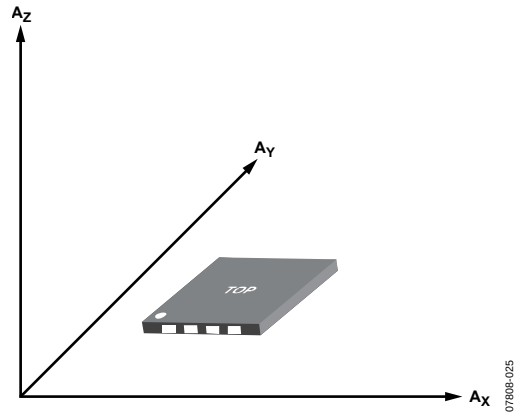


Figure 23. Axes of Acceleration Sensitivity; Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis.

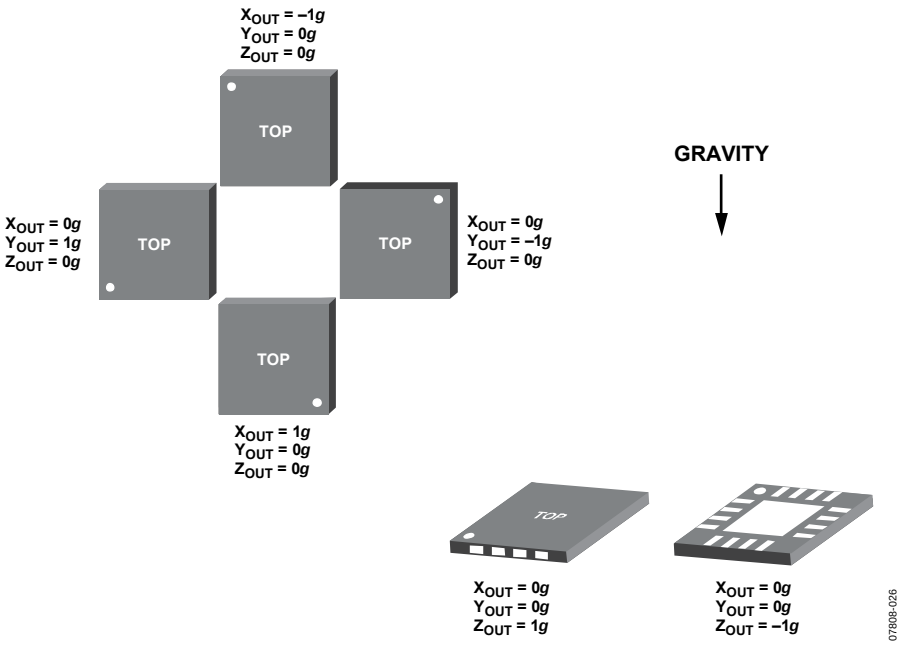


Figure 24. Output Response vs. Orientation to Gravity

LAYOUT AND DESIGN RECOMMENDATIONS

The recommended soldering profile is shown in Figure 25 followed by a description of the profile features in Table 6. The recommended PCB layout or solder land drawing is shown in Figure 26.

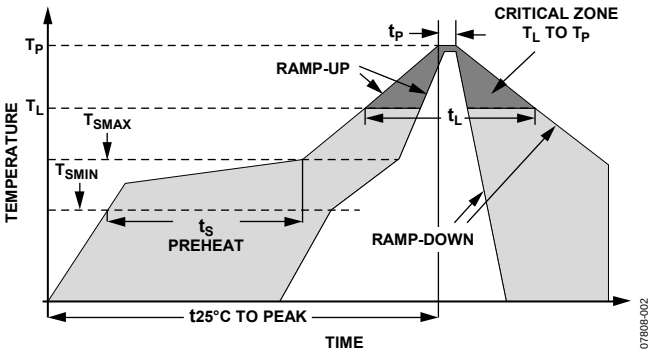


Figure 25. Recommended Soldering Profile

Table 6. Recommended Soldering Profile

Profile Feature	Sn63/Pb37	Pb-Free
Average Ramp Rate (T_L to T_P)	3°C/sec max	3°C/sec max
Preheat		
Minimum Temperature (T_{SMIN})	100°C	150°C
Maximum Temperature (T_{SMAX})	150°C	200°C
Time (T_{SMIN} to T_{SMAX})(t_S)	60 sec to 120 sec	60 sec to 180 sec
T_{SMAX} to T_L		
Ramp-Up Rate	3°C/sec max	3°C/sec max
Time Maintained Above Liquidous (T_L)		
Liquidous Temperature (T_L)	183°C	217°C
Time (t_L)	60 sec to 150 sec	60 sec to 150 sec
Peak Temperature (T_P)	240°C + 0°C/−5°C	260°C + 0°C/−5°C
Time Within 5°C of Actual Peak Temperature (t_P)	10 sec to 30 sec	20 sec to 40 sec
Ramp-Down Rate	6°C/sec max	6°C/sec max
Time 25°C to Peak Temperature	6 minutes max	8 minutes max

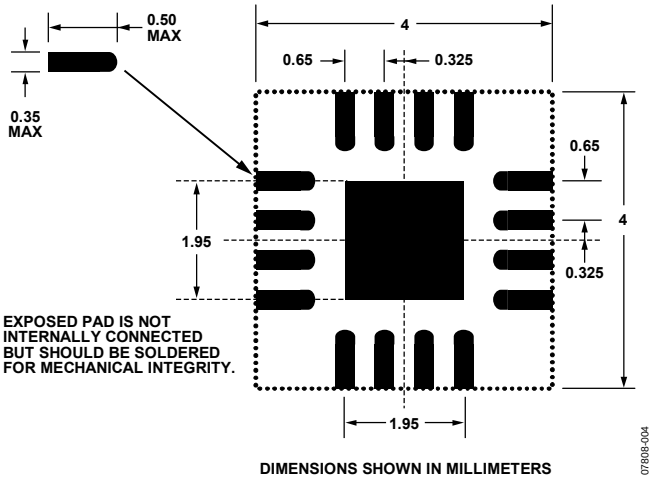


Figure 26. Recommended PCB Layout

OUTLINE DIMENSIONS

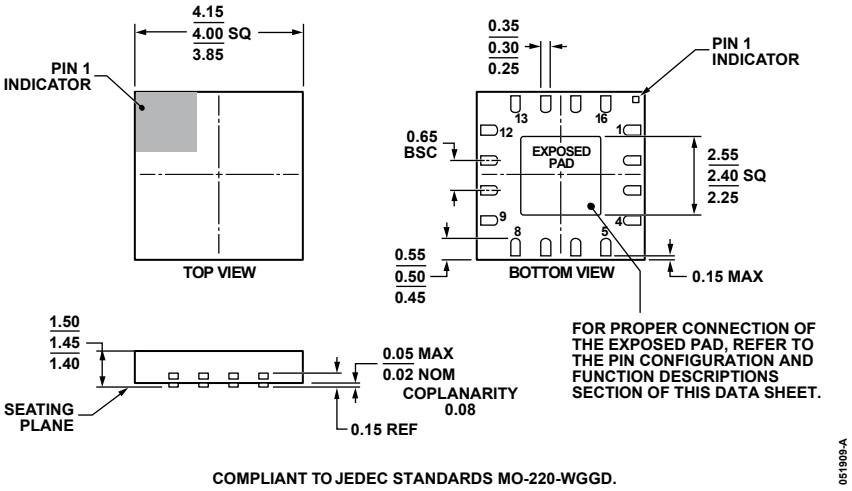


Figure 27. 16-Lead Lead Frame Chip Scale Package [LFCSP_LQ]
4 mm × 4 mm Body, 1.45 mm Thick Quad
(CP-16-14)
Dimensions shown in millimeters

ORDERING GUIDE

Model[1]	Measurement Range	Specified Voltage	Temperature Range	Package Description	Package Option
ADXL335BCPZ	±3 *g*	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ–RL	±3 *g*	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ–RL7	±3 *g*	3 V	−40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
EVAL-ADXL335Z				Evaluation Board	

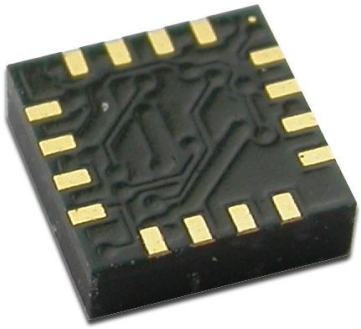
[1] Z = RoHS Compliant Part.

3-Axis Digital Compass IC HMC5883L

Honeywell

Advanced Information

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I^2C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.



The HMC5883L utilizes Honeywell's Anisotropic Magnetoresistive (AMR) technology that provides advantages over other magnetic sensor technologies. These anisotropic, directional sensors feature precision in-axis sensitivity and linearity. These sensors' solid-state construction with very low cross-axis sensitivity is designed to measure both the direction and the magnitude of Earth's magnetic fields, from milli-gauss to 8 gauss. Honeywell's Magnetic Sensors are among the most sensitive and reliable low-field sensors in the industry.

FEATURES	BENEFITS
3-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package	Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly
12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 2 milli-gauss Field Resolution in ±8 Gauss Fields	Enables 1° to 2° Degree Compass Heading Accuracy
Built-In Self Test	Enables Low-Cost Functionality Test after Assembly in Production
Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 µA)	Compatible for Battery Powered Applications
Built-In Strap Drive Circuits	Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation
I^2C Digital Interface	Popular Two-Wire Serial Data Interface for Consumer Electronics
Lead Free Package Construction	RoHS Compliance
Wide Magnetic Field Range (+/-8 Oe)	Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy
Software and Algorithm Support Available	Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available
Fast 160 Hz Maximum Output Rate	Enables Pedestrian Navigation and LBS Applications

SPECIFICATIONS (* Tested at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
Power Supply					
Supply Voltage	VDD Referenced to AGND VDDIO Referenced to DGND	2.16 1.71	2.5 1.8	3.6 VDD+0.1	Volts Volts
Average Current Draw	Idle Mode Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00) VDD = 2.5V, VDDIO = 1.8V (Dual Supply) VDD = VDDIO = 2.5V (Single Supply)	- -	2 100	- -	µA µA
Performance					
Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss
Digital Resolution	VDD=3.0V, GN=0 to 7, 1-LSb, 12-bit ADC	0.73		4.35	milli-gauss
Noise Floor (Field Resolution)	VDD=3.0V, GN=0, No measurement average, Standard Deviation 100 samples (See typical performance graphs below)		2		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Happlied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurment Mode Single Measurement Mode	0.75		75 160	Hz Hz
Measurement Period	From receiving command to data ready		6		ms
Turn-on Time	Ready for I2C commands Analog Circuit Ready for Measurements		200 50		µs ms
Gain Tolerance	All gain/dynamic range settings		±5		%
I^2C Address	8-bit read address 8-bit write address		0x3D 0x3C		hex hex
I^2C Rate	Controlled by I^2C Master			400	kHz
I^2C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V) Rise (VDDIO=1.8V)		0.2*VDDIO 0.8*VDDIO		Volts Volts
Self Test	X & Y Axes Z Axis		±1.16 ±1.08		gauss
	X & Y & Z Axes (GN=5) Positive Bias X & Y & Z Axes (GN=5) Negative Bias	243 -575		575 -243	LSb
Sensitivity Tempco	T_A = -40 to 125°C, Uncompensated Output		-0.3		%/°C
General					
ESD Voltage	Human Body Model (all pins) Charged Device Model (all pins)			2000 750	Volts
Operating Temperature	Ambient	-30		85	°C
Storage Temperature	Ambient, unbiased	-40		125	°C

Characteristics	Conditions*	Min	Typ	Max	Units
Reflow Classification	MSL 3, 260 °C Peak Temperature				
Package Size	Length and Width	2.85	3.00	3.15	mm
Package Height		0.8	0.9	1.0	mm
Package Weight			18		mg

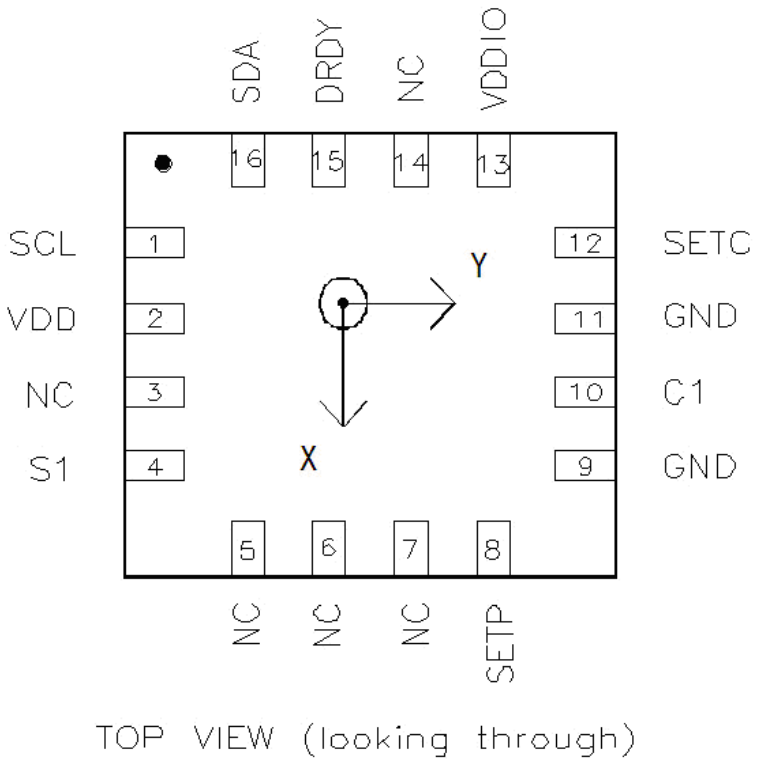
Absolute Maximum Ratings (* Tested at 25°C except stated otherwise.)

Characteristics	Min	Max	Units
Supply Voltage VDD	-0.3	4.8	Volts
Supply Voltage VDDIO	-0.3	4.8	Volts

PIN CONFIGURATIONS

Pin	Name	Description
1	SCL	Serial Clock – I^2C Master/Slave Clock
2	VDD	Power Supply (2.16V to 3.6V)
3	NC	Not to be Connected
4	S1	Tie to VDDIO
5	NC	Not to be Connected
6	NC	Not to be Connected
7	NC	Not to be Connected
8	SETP	Set/Reset Strap Positive – S/R Capacitor (C2) Connection
9	GND	Supply Ground
10	C1	Reservoir Capacitor (C1) Connection
11	GND	Supply Ground
12	SETC	S/R Capacitor (C2) Connection – Driver Side
13	VDDIO	IO Power Supply (1.71V to VDD)
14	NC	Not to be Connected
15	DRDY	Data Ready, Interrupt Pin. Internally pulled high. Optional connection. Low for 250 µsec when data is placed in the data output registers.
16	SDA	Serial Data – I^2C Master/Slave Data

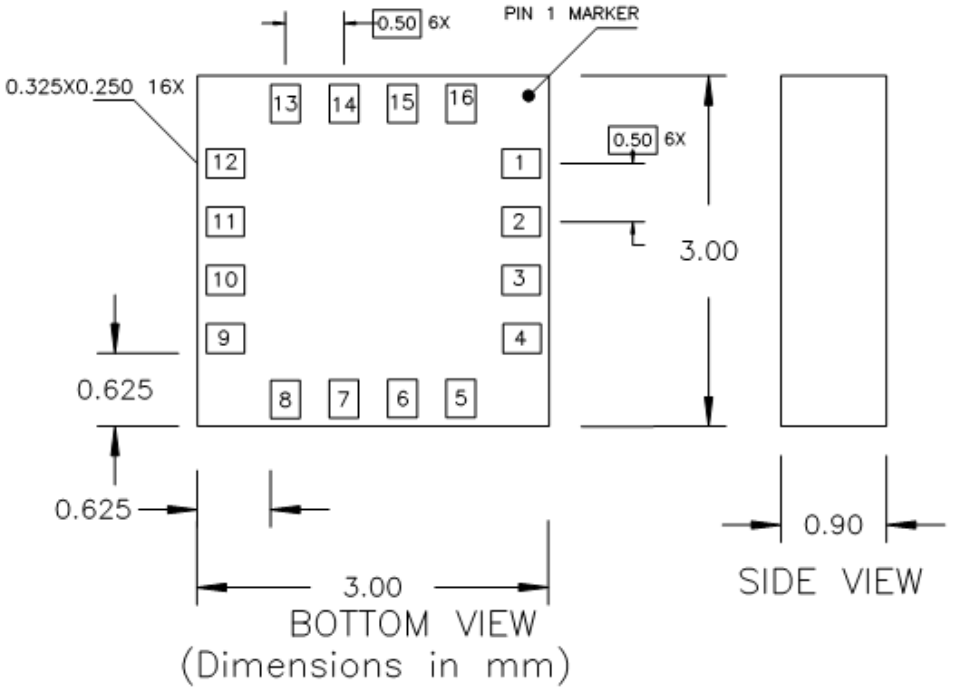
Table 1: Pin Configurations



Arrow indicates direction of magnetic field that generates a positive output reading in Normal Measurement configuration.

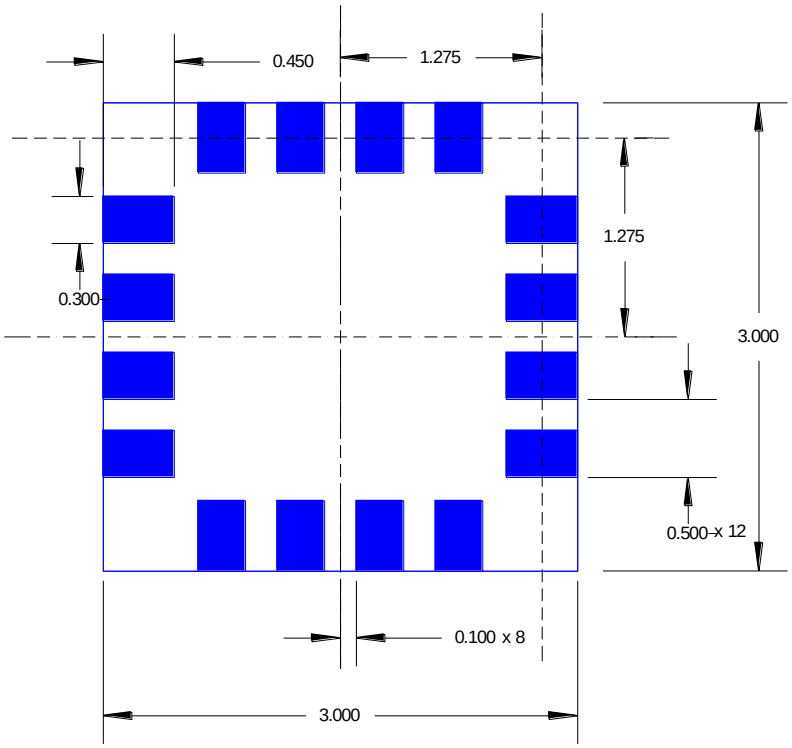
PACKAGE OUTLINES

PACKAGE DRAWING HMC5883L (16-PIN LPCC, dimensions in millimeters)



MOUNTING CONSIDERATIONS

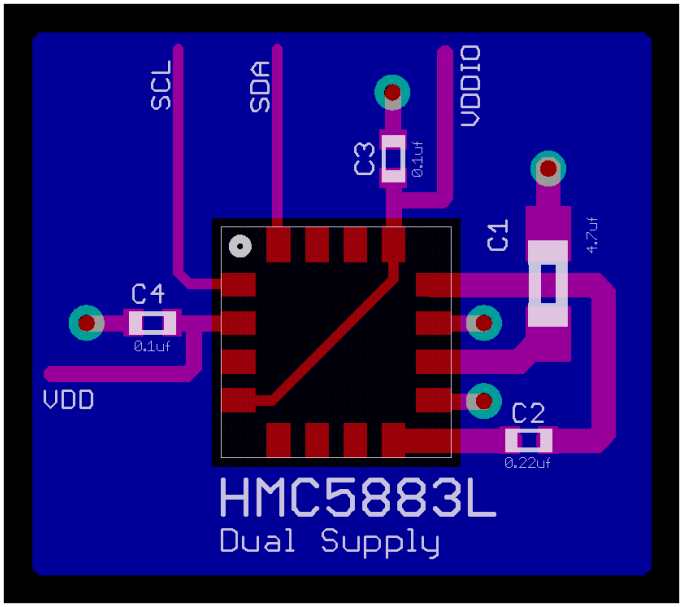
The following is the recommend printed circuit board (PCB) footprint for the HMC5883L.

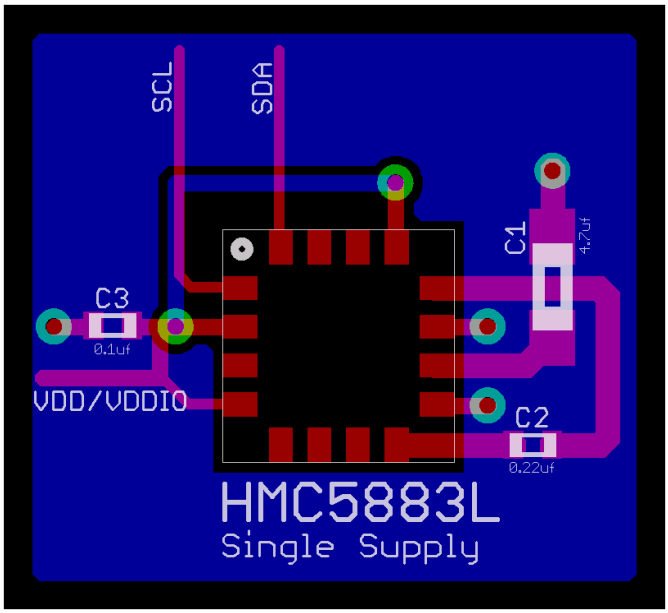


HMC5883 Land Pad Pattern
(All dimensions are in mm)

LAYOUT CONSIDERATIONS

Besides keeping all components that may contain ferrous materials (nickel, etc.) away from the sensor on both sides of the PCB, it is also recommended that there is no conducting copper under/near the sensor in any of the PCB layers. See recommended layout below. Notice that the one trace under the sensor in the dual supply mode is not expected to carry active current since it is for pin 4 pull-up to VDDIO. Power and ground planes are removed under the sensor to minimize possible source of magnetic noise. For best results, use non-ferrous materials for all exposed copper coding.





PCB Pad Definition and Traces

The HMC5883L is a fine pitch LCC package. Refer to previous figure for recommended PCB footprint for proper package centering. Size the traces between the HMC5883L and the external capacitors (C1 and C2) to handle the 1 ampere peak current pulses with low voltage drop on the traces.

Stencil Design and Solder Paste

A 4 mil stencil and 100% paste coverage is recommended for the electrical contact pads.

Reflow Assembly

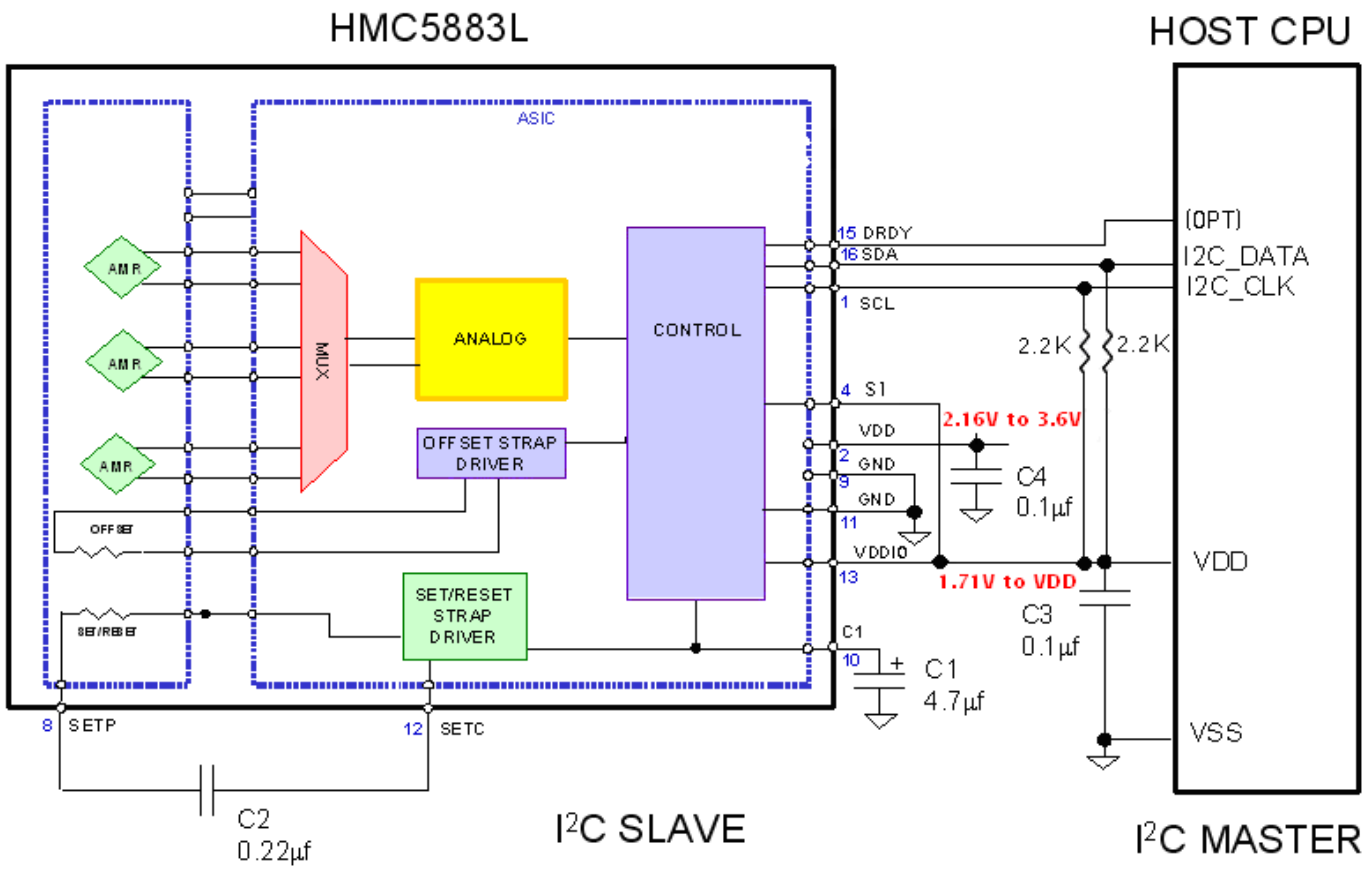
This device is classified as MSL 3 with 260°C peak reflow temperature. A baking process (125°C, 24 hrs) is required if device is not kept continuously in a dry (< 10% RH) environment before assembly. No special reflow profile is required for HMC5883L, which is compatible with lead eutectic and lead-free solder paste reflow profiles. Honeywell recommends adherence to solder paste manufacturer's guidelines. Hand soldering is not recommended. Built-in self test can be used to verify device functionalities after assembly.

External Capacitors

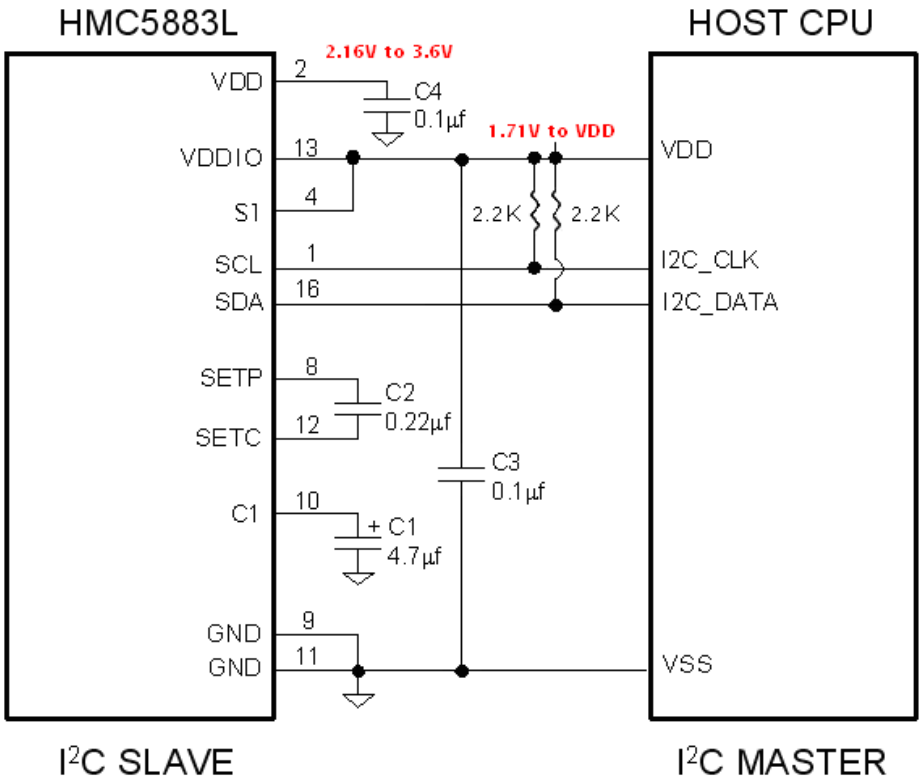
The two external capacitors should be ceramic type construction with low ESR characteristics. The exact ESR values are not critical but values less than 200 milli-ohms are recommended. Reservoir capacitor C1 is nominally 4.7 µF in capacitance, with the set/reset capacitor C2 nominally 0.22 µF in capacitance. Low ESR characteristics may not be in many small SMT ceramic capacitors (0402), so be prepared to up-size the capacitors to gain Low ESR characteristics.

INTERNAL SCHEMATIC DIAGRAM

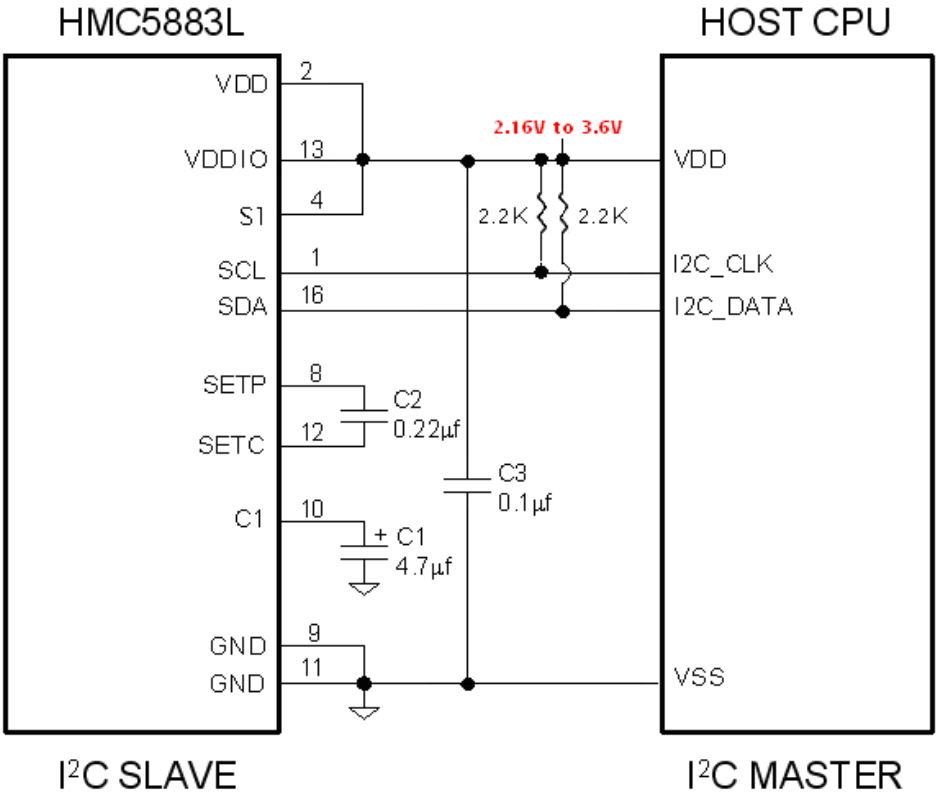
HMC5883L



DUAL SUPPLY REFERENCE DESIGN



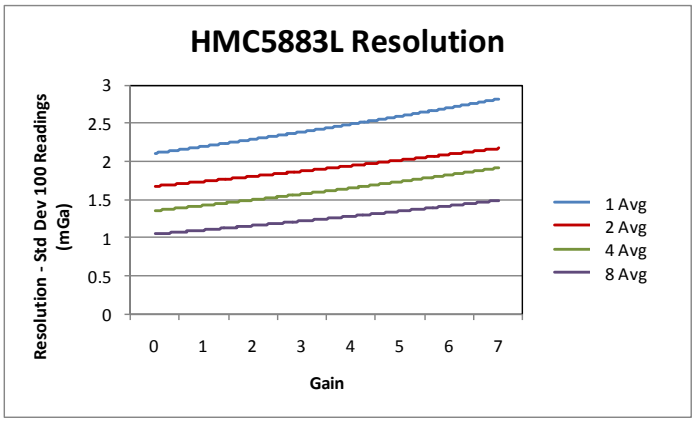
SINGLE SUPPLY REFERENCE DESIGN

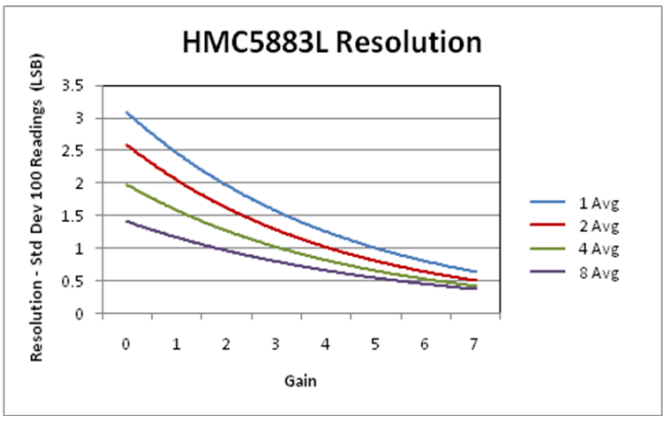


PERFORMANCE

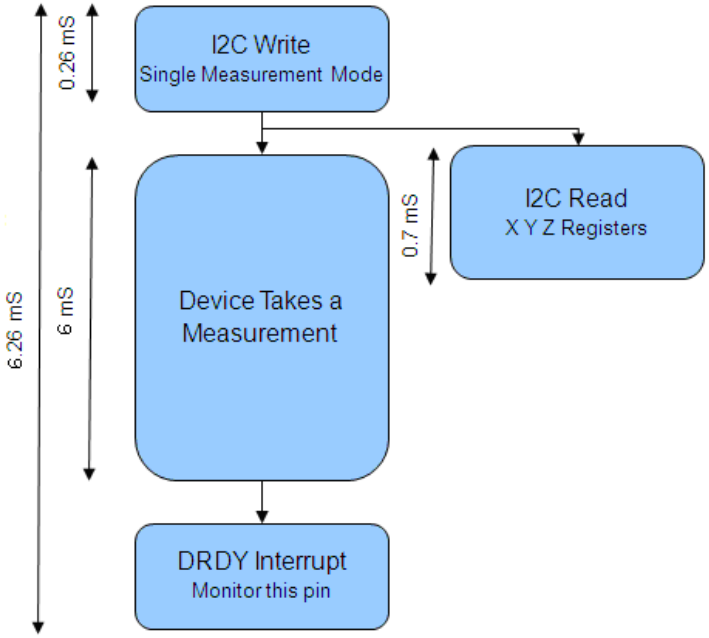
The following graph(s) highlight HMC5883L's performance.

Typical Noise Floor (Field Resolution)





Typical Measurement Period in Single-Measurement Mode



** Monitoring of the DRDY Interrupt pin is only required if maximum output rate is desired.*

BASIC DEVICE OPERATION

Anisotropic Magneto-Resistive Sensors

The Honeywell HMC5883L magnetoresistive sensor circuit is a trio of sensors and application specific support circuits to measure magnetic fields. With power supply applied, the sensor converts any incident magnetic field in the sensitive axis directions to a differential voltage output. The magnetoresistive sensors are made of a nickel-iron (Permalloy) thin-film and patterned as a resistive strip element. In the presence of a magnetic field, a change in the bridge resistive elements causes a corresponding change in voltage across the bridge outputs.

These resistive elements are aligned together to have a common sensitive axis (indicated by arrows in the pinout diagram) that will provide positive voltage change with magnetic fields increasing in the sensitive direction. Because the output is only proportional to the magnetic field component along its axis, additional sensor bridges are placed at orthogonal directions to permit accurate measurement of magnetic field in any orientation.

Self Test

To check the HMC5883L for proper operation, a self test feature in incorporated in which the sensor is internally excited with a nominal magnetic field (in either positive or negative bias configuration). This field is then measured and reported. This function is enabled and the polarity is set by bits MS[n] in the configuration register A. An internal current source generates DC current (about 10 mA) from the VDD supply. This DC current is applied to the offset straps of the magneto-resistive sensor, which creates an artificial magnetic field bias on the sensor. The difference of this measurement and the measurement of the ambient field will be put in the data output register for each of the three axes. By using this built-in function, the manufacturer can quickly verify the sensor's full functionality after the assembly without additional test setup. The self test results can also be used to estimate/compensate the sensor's sensitivity drift due to temperature.

For each "self test measurement", the ASIC:

1. Sends a "Set" pulse
2. Takes one measurement (M1)
3. Sends the (~10 mA) offset current to generate the (~1.1 Gauss) offset field and takes another measurement (M2)
4. Puts the difference of the two measurements in sensor's data output register:

   Output = [M2 – M1] (i.e. output = offset field only)

See SELF TEST OPERATION section later in this datasheet for additional details.

Power Management

This device has two different domains of power supply. The first one is VDD that is the power supply for internal operations and the second one is VDDIO that is dedicated to IO interface. It is possible to work with VDDIO equal to VDD; Single Supply mode, or with VDDIO lower than VDD allowing HMC5883L to be compatible with other devices on board.

I^2C Interface

Control of this device is carried out via the I^2C bus. This device will be connected to this bus as a slave device under the control of a master device, such as the processor.

This device is compliant with *I^2C-Bus Specification*, document number: 9398 393 40011. As an I^2C compatible device, this device has a 7-bit serial address and supports I^2C protocols. This device supports standard and fast modes, 100kHz and 400kHz, respectively, but does not support the high speed mode (Hs). External pull-up resistors are required to support these standard and fast speed modes.

Activities required by the master (register read and write) have priority over internal activities, such as the measurement. The purpose of this priority is to not keep the master waiting and the I^2C bus engaged for longer than necessary.

Internal Clock

The device has an internal clock for internal digital logic functions and timing management. This clock is not available to external usage.

H-Bridge for Set/Reset Strap Drive

The ASIC contains large switching FETs capable of delivering a large but brief pulse to the Set/Reset strap of the sensor. This strap is largely a resistive load. There is no need for an external Set/Reset circuit. The controlling of the Set/Reset function is done automatically by the ASIC for each measurement. One half of the difference from the measurements taken after a set pulse and after a reset pulse will be put in the data output register for each of the three axes. By doing so, the sensor's internal offset and its temperature dependence is removed/cancelled for all measurements. The set/reset pulses also effectively remove the past magnetic history (magnetism) in the sensor, if any.

For each "measurement", the ASIC:

1. Sends a "Set" pulse
2. Takes one measurement (Mset)
3. Sends a "Reset" pulse
4. Takes another measurement (Mreset)
5. Puts the following result in sensor's data output register:

Output = [Mset – Mreset] / 2

Charge Current Limit

The current that reservoir capacitor (C1) can draw when charging is limited for both single supply and dual supply configurations. This prevents drawing down the supply voltage (VDD).

MODES OF OPERATION

This device has several operating modes whose primary purpose is power management and is controlled by the Mode Register. This section describes these modes.

Continuous-Measurement Mode

During continuous-measurement mode, the device continuously makes measurements, at user selectable rate, and places measured data in data output registers. Data can be re-read from the data output registers if necessary; however, if the master does not ensure that the data register is accessed before the completion of the next measurement, the data output registers are updated with the new measurement. To conserve current between measurements, the device is placed in a state similar to idle mode, but the Mode Register is not changed to Idle Mode. That is, MD[n] bits are unchanged. Settings in the Configuration Register A affect the data output rate (bits DO[n]), the measurement configuration (bits MS[n]), when in continuous-measurement mode. All registers maintain values while in continuous-measurement mode. The I^2C bus is enabled for use by other devices on the network in while continuous-measurement mode.

Single-Measurement Mode

This is the default power-up mode. During single-measurement mode, the device makes a single measurement and places the measured data in data output registers. After the measurement is complete and output data registers are updated, the device is placed in idle mode, and the Mode Register *is* changed to idle mode by setting MD[n] bits. Settings in the configuration register affect the measurement configuration (bits MS[n])when in single-measurement mode. All registers maintain values while in single-measurement mode. The I^2C bus is enabled for use by other devices on the network while in single-measurement mode.

Idle Mode

During this mode the device is accessible through the I^2C bus, but major sources of power consumption are disabled, such as, but not limited to, the ADC, the amplifier, and the sensor bias current. All registers maintain values while in idle mode. The I^2C bus is enabled for use by other devices on the network while in idle mode.

REGISTERS

This device is controlled and configured via a number of on-chip registers, which are described in this section. In the following descriptions, *set* implies a logic 1, and *reset* or *clear* implies a logic 0, unless stated otherwise.

Register List
The table below lists the registers and their access. All address locations are 8 bits.

Address Location	Name	Access
00	Configuration Register A	Read/Write
01	Configuration Register B	Read/Write
02	Mode Register	Read/Write
03	Data Output X MSB Register	Read
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read
08	Data Output Y LSB Register	Read
09	Status Register	Read
10	Identification Register A	Read
11	Identification Register B	Read
12	Identification Register C	Read

Table2: Register List

Register Access

This section describes the process of reading from and writing to this device. The devices uses an address pointer to indicate which register location is to be read from or written to. These pointer locations are sent from the master to this slave device and succeed the 7-bit address (0x1E) plus 1 bit read/write identifier, i.e. 0x3D for read and 0x3C for write.

To minimize the communication between the master and this device, the address pointer updated automatically without master intervention. The register pointer will be incremented by 1 automatically after the current register has been read successfully.

The address pointer value itself cannot be read via the I^2C bus.
Any attempt to read an invalid address location returns 0's, and any write to an invalid address location or an undefined bit within a valid address location is ignored by this device.

To move the address pointer to a random register location, first issue a "write" to that register location with no data byte following the commend. For example, to move the address pointer to register 10, send 0x3C 0x0A.

Configuration Register A

The configuration register is used to configure the device for setting the data output rate and measurement configuration. CRA0 through CRA7 indicate bit locations, with *CRA* denoting the bits that are in the configuration register. CRA7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit.CRA default is 0x10.

CRA7	CRA6	CRA5	CRA4	CRA3	CRA2	CRA1	CRA0
(0)	MA1(0)	MA0(0)	DO2 (1)	DO1 (0)	DO0 (0)	MS1 (0)	MS0 (0)

Table 3: Configuration Register A

Location	Name	Description
CRA7	CRA7	Bit CRA7 is reserved for future function. Set to 0 when configuring CRA.
CRA6 to CRA5	MA1 to MA0	Select number of samples averaged (1 to 8) per measurement output. 00 = 1(Default); 01 = 2; 10 = 4; 11 = 8
CRA4 to CRA2	DO2 to DO0	Data Output Rate Bits. These bits set the rate at which data is written to all three data output registers.
CRA1 to CRA0	MS1 to MS0	Measurement Configuration Bits. These bits define the measurement flow of the device, specifically whether or not to incorporate an applied bias into the measurement.

Table 4: Configuration Register A Bit Designations

The Table below shows all selectable output rates in continuous measurement mode. All three channels shall be measured within a given output rate. Other output rates with maximum rate of 160 Hz can be achieved by monitoring DRDY interrupt pin in single measurement mode.

DO2	DO1	DO0	Typical Data Output Rate (Hz)
0	0	0	0.75
0	0	1	1.5
0	1	0	3
0	1	1	7.5
1	0	0	15 (Default)
1	0	1	30
1	1	0	75
1	1	1	Reserved

Table 5: Data Output Rates

MS1	MS0	Measurement Mode
0	0	Normal measurement configuration (Default). In normal measurement configuration the device follows normal measurement flow. The positive and negative pins of the resistive load are left floating and high impedance.
0	1	Positive bias configuration for X, Y, and Z axes. In this configuration, a positive current is forced across the resistive load for all three axes.
1	0	Negative bias configuration for X, Y and Z axes. In this configuration, a negative current is forced across the resistive load for all three axes..
1	1	This configuration is reserved.

Table 6: Measurement Modes

Configuration Register B

The configuration register B for setting the device gain. CRB0 through CRB7 indicate bit locations, with *CRB* denoting the bits that are in the configuration register. CRB7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit. CRB default is 0x20.

CRB7	CRB6	CRB5	CRB4	CRB3	CRB2	CRB1	CRB0
GN2 (0)	GN1 (0)	GN0 (1)	(0)	(0)	(0)	(0)	(0)

Table 7: Configuration B Register

Location	Name	Description
CRB7 to CRB5	GN2 to GN0	Gain Configuration Bits. These bits configure the gain for the device. The gain configuration is common for all channels.
CRB4 to CRB0	0	These bits must be cleared for correct operation.

Table 8: Configuration Register B Bit Designations

The table below shows nominal gain settings. Use the “Gain” column to convert counts to Gauss. The “Digital Resolution” column is the theoretical value in term of milli-Gauss per count (LSb) which is the inverse of the values in the “Gain” column. The effective resolution of the usable signal also depends on the noise floor of the system, i.e.

Effective Resolution = Max (Digital Resolution, Noise Floor)

Choose a lower gain value (higher GN#) when total field strength causes overflow in one of the data output registers (saturation). Note that the very first measurement after a gain change maintains the same gain as the previous setting. **The new gain setting is effective from the second measurement and on.**

GN2	GN1	GN0	Recommended Sensor Field Range	Gain (LSb/ Gauss)	Digital Resolution (mG/LSb)	Output Range
0	0	0	± 0.88 Ga	1370	0.73	0xF800–0x07FF (-2048–2047)
0	0	1	± 1.3 Ga	1090 (default)	0.92	0xF800–0x07FF (-2048–2047)
0	1	0	± 1.9 Ga	820	1.22	0xF800–0x07FF (-2048–2047)
0	1	1	± 2.5 Ga	660	1.52	0xF800–0x07FF (-2048–2047)
1	0	0	± 4.0 Ga	440	2.27	0xF800–0x07FF (-2048–2047)
1	0	1	± 4.7 Ga	390	2.56	0xF800–0x07FF (-2048–2047)
1	1	0	± 5.6 Ga	330	3.03	0xF800–0x07FF (-2048–2047)
1	1	1	± 8.1 Ga	230	4.35	0xF800–0x07FF (-2048–2047)

Table 9: Gain Settings

Mode Register

The mode register is an 8-bit register from which data can be read or to which data can be written. This register is used to select the operating mode of the device. MR0 through MR7 indicate bit locations, with *MR* denoting the bits that are in the mode register. MR7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit. Mode register default is 0x01.

MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
HS(0)	(0)	(0)	(0)	(0)	(0)	MD1 (0)	MD0 (1)

Table 10: Mode Register

Location	Name	Description
MR7 to MR2	HS	Set this pin to enable High Speed I2C, 3400kHz.
MR1 to MR0	MD1 to MD0	Mode Select Bits. These bits select the operation mode of this device.

Table 11: Mode Register Bit Designations

MD1	MD0	Operating Mode
0	0	Continuous-Measurement Mode. In continuous-measurement mode, the device continuously performs measurements and places the result in the data register. RDY goes high when new data is placed in all three registers. After a power-on or a write to the mode or configuration register, the first measurement set is available from all three data output registers after a period of $2/f_{DO}$ and subsequent measurements are available at a frequency of f_{DO}, where f_{DO} is the frequency of data output.
0	1	Single-Measurement Mode (Default). When single-measurement mode is selected, device performs a single measurement, sets RDY high and returned to idle mode. Mode register returns to idle mode bit values. The measurement remains in the data output register and RDY remains high until the data output register is read or another measurement is performed.
1	0	Idle Mode. Device is placed in idle mode.
1	1	Idle Mode. Device is placed in idle mode.

Table 12: Operating Modes

Data Output X Registers A and B

The data output X registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel X. Data output X register A contains the MSB from the measurement result, and data output X register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DXRA0 through DXRA7 and DXRB0 through DXRB7 indicate bit locations, with *DXRA* and *DXRB* denoting the bits that are in the data output X registers. DXRA7 and DXRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DXRA7	**DXRA6**	**DXRA5**	**DXRA4**	**DXRA3**	**DXRA2**	**DXRA1**	**DXRA0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DXRB7	**DXRB6**	**DXRB5**	**DXRB4**	**DXRB3**	**DXRB2**	**DXRB1**	**DXRB0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 13: Data Output X Registers A and B

Data Output Y Registers A and B

The data output Y registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel Y. Data output Y register A contains the MSB from the measurement result, and data output Y register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DYRA0 through DYRA7 and DYRB0 through DYRB7 indicate bit locations, with *DYRA* and *DYRB* denoting the bits that are in the data output Y registers. DYRA7 and DYRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DYRA7	**DYRA6**	**DYRA5**	**DYRA4**	**DYRA3**	**DYRA2**	**DYRA1**	**DYRA0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DYRB7	**DYRB6**	**DYRB5**	**DYRB4**	**DYRB3**	**DYRB2**	**DYRB1**	**DYRB0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 14: Data Output Y Registers A and B

Data Output Z Registers A and B

The data output Z registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel Z. Data output Z register A contains the MSB from the measurement result, and data output Z register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DZRA0 through DZRA7 and DZRB0 through DZRB7 indicate bit locations, with *DZRA* and *DZRB* denoting the bits that are in the data output Z registers. DZRA7 and DZRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DZRA7	DZRA6	DZRA5	DZRA4	DZRA3	DZRA2	DZRA1	DZRA0
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DZRB7	**DZRB6**	**DZRB5**	**DZRB4**	**DZRB3**	**DZRB2**	**DZRB1**	**DZRB0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 15: Data Output Z Registers A and B

Data Output Register Operation

When one or more of the output registers are read, new data cannot be placed in any of the output data registers until all six data output registers are read. This requirement also impacts DRDY and RDY, which cannot be cleared until new data is placed in all the output registers.

Status Register

The status register is an 8-bit read-only register. This register is used to indicate device status. SR0 through SR7 indicate bit locations, with *SR* denoting the bits that are in the status register. SR7 denotes the first bit of the data stream.

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
(0)	(0)	(0)	(0)	(0)	(0)	LOCK (0)	RDY(0)

Table 16: Status Register

Location	Name	Description
SR7 to SR2	0	These bits are reserved.
SR1	LOCK	Data output register lock. This bit is set when: 1.some but not all for of the six data output registers have been read, 2. Mode register has been read. When this bit is set, the six data output registers are locked and any new data will not be placed in these register until one of these conditions are met: 1.all six bytes have been read, 2. the mode register is changed, 3. the measurement configuration (CRA) is changed, 4. power is reset.
SR0	RDY	Ready Bit. Set when data is written to all six data registers. Cleared when device initiates a write to the data output registers and after one or more of the data output registers are written to. When RDY bit is clear it shall remain cleared for a 250 µs. DRDY pin can be used as an alternative to the status register for monitoring the device for measurement data.

Table 17: Status Register Bit Designations

Identification Register A

The identification register A is used to identify the device. IRA0 through IRA7 indicate bit locations, with *IRA* denoting the bits that are in the identification register A. IRA7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

The identification value for this device is stored in this register. This is a read-only register.
Register values. ASCII value *H*

IRA7	IRA6	IRA5	IRA4	IRA3	IRA2	IRA1	IRA0
0	1	0	0	1	0	0	0

Table 18: Identification Register A Default Values

Identification Register B

The identification register B is used to identify the device. IRB0 through IRB7 indicate bit locations, with *IRB* denoting the bits that are in the identification register A. IRB7 denotes the first bit of the data stream.

Register values. ASCII value *4*

IRB7	IRB6	IRB5	IRB4	IRB3	IRB2	IRB1	IRB0
0	0	1	1	0	1	0	0

Table 19: Identification Register B Default Values

Identification Register C

The identification register C is used to identify the device. IRC0 through IRC7 indicate bit locations, with *IRC* denoting the bits that are in the identification register A. IRC7 denotes the first bit of the data stream.

Register values. ASCII value *3*

IRC7	IRC6	IRC5	IRC4	IRC3	IRC2	IRC1	IRC0
0	0	1	1	0	0	1	1

Table 20: Identification Register C Default Values

I^2C COMMUNICATION PROTOCOL

The HMC5883L communicates via a two-wire I^2C bus system as a slave device. The HMC5883L uses a simple protocol with the interface protocol defined by the I^2C bus specification, and by this document. The data rate is at the standard-mode 100kbps or 400kbps rates as defined in the I^2C Bus Specifications. The bus bit format is an 8-bit Data/Address send and a 1-bit acknowledge bit. The format of the data bytes (payload) shall be case sensitive ASCII characters or binary data to the HMC5883L slave, and binary data returned. Negative binary values will be in two's complement form. The default (factory) HMC5883L 8-bit slave address is 0x3C for write operations, or 0x3D for read operations.

The HMC5883L Serial Clock (SCL) and Serial Data (SDA) lines require resistive pull-ups (Rp) between the master device (usually a host microprocessor) and the HMC5883L. Pull-up resistance values of about 2.2K to 10K ohms are recommended with a nominal VDDIO voltage. Other resistor values may be used as defined in the I^2C Bus Specifications that can be tied to VDDIO.

The SCL and SDA lines in this bus specification may be connected to multiple devices. The bus can be a single master to multiple slaves, or it can be a multiple master configuration. All data transfers are initiated by the master device, which is responsible for generating the clock signal, and the data transfers are 8 bit long. All devices are addressed by I^2C's unique 7-bit address. After each 8-bit transfer, the master device generates a 9th clock pulse, and releases the SDA line. The receiving device (addressed slave) will pull the SDA line low to acknowledge (ACK) the successful transfer or leave the SDA high to negative acknowledge (NACK).

Per the I^2C spec, all transitions in the SDA line must occur when SCL is low. This requirement leads to two unique conditions on the bus associated with the SDA transitions when SCL is high. Master device pulling the SDA line low while the SCL line is high indicates the Start (S) condition, and the Stop (P) condition is when the SDA line is pulled high while the SCL line is high. The I^2C protocol also allows for the Restart condition in which the master device issues a second start condition without issuing a stop.

All bus transactions begin with the master device issuing the start sequence followed by the slave address byte. The address byte contains the slave address; the upper 7 bits (bits7-1), and the Least Significant bit (LSb). The LSb of the address byte designates if the operation is a read (LSb=1) or a write (LSb=0). At the 9th clock pulse, the receiving slave device will issue the ACK (or NACK). Following these bus events, the master will send data bytes for a write operation, or the slave will clock out data with a read operation. All bus transactions are terminated with the master issuing a stop sequence.

I^2C bus control can be implemented with either hardware logic or in software. Typical hardware designs will release the SDA and SCL lines as appropriate to allow the slave device to manipulate these lines. In a software implementation, care must be taken to perform these tasks in code.

OPERATIONAL EXAMPLES

The HMC5883L has a fairly quick stabilization time from no voltage to stable and ready for data retrieval. The nominal 56 milli-seconds with the factory default single measurement mode means that the six bytes of magnetic data registers (DXRA, DXRB, DZRA, DZRB, DYRA, and DYRB) are filled with a valid first measurement.

To change the measurement mode to continuous measurement mode, after the power-up time send the three bytes:

0x3C 0x02 0x00

This writes the 00 into the second register or mode register to switch from single to continuous measurement mode setting. With the data rate at the factory default of 15Hz updates, a 67 milli-second typical delay should be allowed by the I^2C master before querying the HMC5883L data registers for new measurements. To clock out the new data, send:

0x3D, and clock out DXRA, DXRB, DZRA, DZRB, DYRA, and DYRB located in registers 3 through 8. The HMC5883L will automatically re-point back to register 3 for the next 0x3D query. All six data registers must be read properly before new data can be placed in any of these data registers.

Below is an example of a (power-on) initialization process for “continuous-measurement mode”:

1. Write CRA (00) – send **0x3C 0x00 0x70** (8-average, 15 Hz default, normal measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5, or any other desired gain)
3. Write Mode (02) – send **0x3C 0x02 0x00** (Continuous-measurement mode)
4. Wait 6 ms or monitor status register or DRDY hardware interrupt pin
5. Loop
   Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
   Convert three 16-bit 2’s compliment hex values to decimal values and assign to X, Z, Y, respectively.
   Send **0x3C 0x03** (point to first data register 03)
   Wait about 67 ms (if 15 Hz rate) or monitor status register or DRDY hardware interrupt pin
   End_loop

Below is an example of a (power-on) initialization process for “single-measurement mode”:

1. Write CRA (00) – send **0x3C 0x00 0x70** (8-average, 15 Hz default or any other rate, normal measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5, or any other desired gain)
3. For each measurement query:
   Write Mode (02) – send **0x3C 0x02 0x01** (Single-measurement mode)
   Wait 6 ms or monitor status register or DRDY hardware interrupt pin
   Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
   Convert three 16-bit 2’s compliment hex values to decimal values and assign to X, Z, Y, respectively.

SELF TEST OPERATION

To check the HMC5883L for proper operation, a self test feature in incorporated in which the sensor offset straps are excited to create a nominal field strength (bias field) to be measured. To implement self test, the least significant bits (MS1 and MS0) of configuration register A are changed from 00 to 01 (positive bias) or 10 (negetive bias).

Then, by placing the mode register into single or continuous-measurement mode, two data acquisition cycles will be made on each magnetic vector. The first acquisition will be a set pulse followed shortly by measurement data of the external field. The second acquisition will have the offset strap excited (about 10 mA) in the positive bias mode for X, Y, and Z axes to create about a 1.1 gauss self test field plus the external field. The first acquisition values will be subtracted from the second acquisition, and the net measurement will be placed into the data output registers.

Since self test adds ~1.1 Gauss additional field to the existing field strength, using a reduced gain setting prevents sensor from being saturated and data registers overflowed. For example, if the configuration register B is set to 0xA0 (Gain=5), values around +452 LSb (1.16 Ga * 390 LSb/Ga) will be placed in the X and Y data output registers and around +421 (1.08 Ga * 390 LSb/Ga) will be placed in Z data output register. To leave the self test mode, change MS1 and MS0 bit of the configuration register A back to 00 (Normal Measurement Mode). Acceptable limits of the self test values depend on the gain setting. Limits for Gain=5 is provided in the specification table.

Below is an example of a “positive self test” process using continuous-measurement mode:

1. Write CRA (00) – send **0x3C 0x00 0x71** (8-average, 15 Hz default, positive self test measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5)
3. Write Mode (02) – send **0x3C 0x02 0x00** (Continuous-measurement mode)
4. Wait 6 ms or monitor status register or DRDY hardware interrupt pin
5. Loop
   - Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
   - Convert three 16-bit 2’s compliment hex values to decimal values and assign to X, Z, Y, respectively.
   - Send **0x3C 0x03** (point to first data register 03)
   - Wait about 67 ms (if 15 Hz rate) or monitor status register or DRDY hardware interrupt pin

   End_loop
6. Check limits –
   If all 3 axes (X, Y, and Z) are within reasonable limits (243 to 575 for Gain=5, adjust these limits basing on the gain setting used. See an example below.) Then
   - All 3 axes pass positive self test
   - Write CRA (00) – send **0x3C 0x00 0x70** (Exit self test mode and this procedure)

   Else
   - If Gain<7
     - Write CRB (01) – send **0x3C 0x01 0x_0** (Increase gain setting and retry, skip the next data set)
   - Else
     - At least one axis did not pass positive self test
     - Write CRA (00) – send **0x3C 0x00 0x70** (Exit self test mode and this procedure)

   End If

Below is an example of how to adjust the “positive self” test limits basing on the gain setting:

1. If Gain = 6, self test limits are:
   Low Limit = 243 * 330/390 = 206
   High Limit = 575 * 330/390 = 487

2. If Gain = 7, self test limits are:
   Low Limit = 243 * 230/390 = 143
   High Limit = 575 * 230/390 = 339

SCALE FACTOR TEMPERATURE COMPENSATION

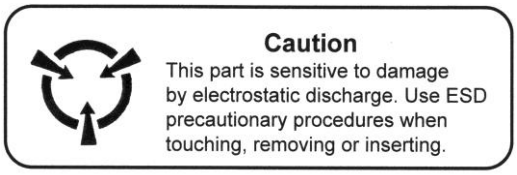
The built-in self test can also be used to periodically compensate the scaling errors due to temperature variations. A compensation factor can be found by comparing the self test outputs with the ones obtained at a known temperature. For example, if the self test output is 400 at room temperature and 300 at the current temperature then a compensation factor of (400/300) should be applied to all current magnetic readings. A temperature sensor is not required using this method.

Below is an example of a temperature compensation process using positive self test method:

1. If self test measurement at a temperature “when the last magnetic calibration was done”:
   X_STP = 400
   Y_STP = 410
   Z_STP = 420
2. If self test measurement at a different tmperature:
   X_STP = 300 (Lower than before)
   Y_STP = 310 (Lower than before)
   Z_STP = 320 (Lower than before)
   Then
   X_TempComp = 400/300
   Y_TempComp = 410/310
   Z_TempComp = 420/320
3. Applying to all new measurements:
   X = X * X_TempComp
   Y = Y * Y_TempComp
   Z = Z * Z_TempComp
   Now all 3 axes are temperature compensated, i.e. sensitivity is same as “when the last magnetic calibration was done”; therefore, the calibration coefficients can be applied without modification.
4. Repeat this process periodically or,for every Δt degrees of temperature change measured, if available.

ORDERING INFORMATION

Ordering Number	Product
HMC5883L-T	Cut Tape
HMC5883L-TR	Tape and Reel 4k pieces/reel



CAUTION: ESDS CAT. 1B

FIND OUT MORE

For more information on Honeywell’s Magnetic Sensors visit us online at www.magneticsensors.com or contact us at 1-800-323-8295 (763-954-2474 internationally).

The application circuits herein constitute typical usage and interface of Honeywell product. Honeywell does not warranty or assume liability of customer-designed circuits derived from this description or depiction.

Honeywell reserves the right to make changes to improve reliability, function or design. Honeywell does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

U.S. Patents 4,441,072, 4,533,872, 4,569,742, 4,681,812, 4,847,584 and 6,529,114 apply to the technology described

Honeywell
12001 Highway 55
Plymouth, MN 55441
Tel: 800-323-8295
www.magneticsensors.com

Form # 900405 Rev E
February 2013
©2010 Honeywell International Inc.

Honeywell

ACS712 Current Sensor

Basic Overview

The ACS712 Current Sensors offered on the internet are designed to be easily used with micro controllers like the Arduino.

These sensors are based on the Allegro ACS712ELC chip.

These current sensors are offered with full scale values of 5A, 20A and 30A.

The basic functional operation of each of these devices is identical. The only difference is with the scale factor at the output as detailed below.

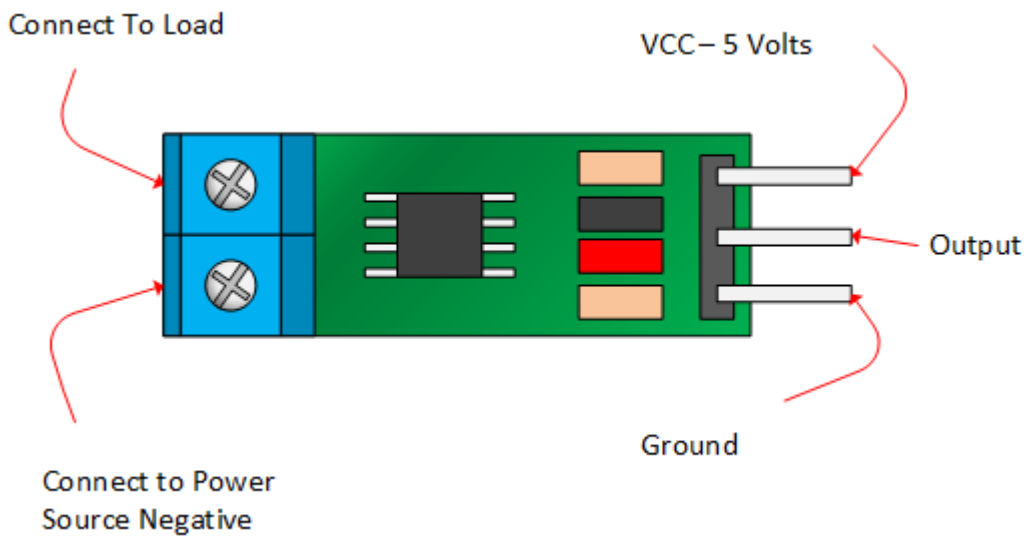
Sensor Specifications

	5A Module	20A Module	30A Module
Supply Voltage (VCC)	5Vdc Nominal	5Vdc Nominal	5Vdc Nominal
Measurement Range	-5 to +5 Amps	-20 to +20 Amps	-30 to +30 Amps
Voltage at 0A	VCC/2 (nominally 2.5Vdc)	VCC/2 (nominally 2.5Vdc)	VCC/2 (nominally 2.5VDC)
Scale Factor	185 mV per Amp	100 mV per Amp	66 mV per Amp
Chip	ACS712ELC-05A	ACS712ELC-10A	ACS712ELC-30A

ACS712 Module Pin Outs and Connections

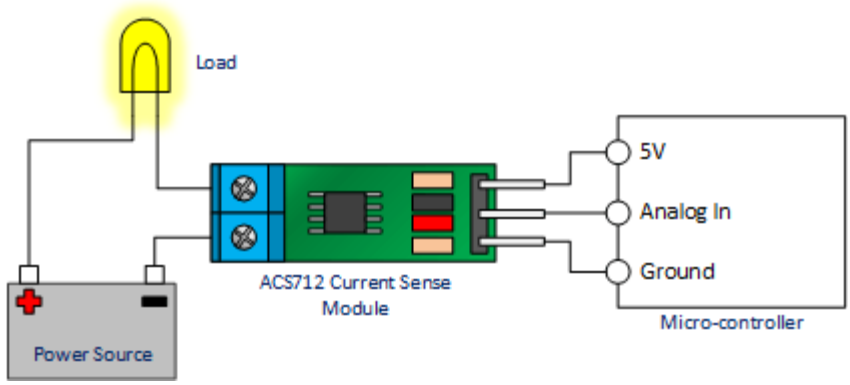
The picture below identifies the pin outs for the ACS172 Modules.

Pay attention to the polarity at the load end of the device. If you are connected as illustrated below, the output will raise. If you connect it opposite of this picture, the output will decrease from the 2.5 volt offset.



Basic Hook Up and Functional Description

As mentioned before, these modules are primarily designed for use with micro-controllers like the Arduino. In those applications, the connections would be as picture below:



If the light bulb shown in the picture above were disconnected, the output of the ACS712 module would be 2.500 volts.

Once connected, the output would be scaled to the current drawn through the bulb. If this were a 5 Amp module and the light bulb pulled 1 Amp, the output of the module would be 2.685 volts.

Now imagine the battery polarity reversed. Using the same 5A module, the output would be 2.315 volts.

IMPORTANT NOTE – This device is a Hall Effect transducer. It should not be used near significant magnetic fields.

GPS

locate, communicate, accelerate

NEO-6

u-blox 6 GPS Modules

Data Sheet

Abstract

Technical data sheet describing the cost effective, high-performance u-blox 6 based NEO-6 series of GPS modules, that brings the high performance of the u-blox 6 positioning engine to the miniature NEO form factor.
These receivers combine a high level of integration capability with flexible connectivity options in a small package. This makes them perfectly suited for mass-market end products with strict size and cost requirements.

16.0 x 12.2 x 2.4 mm

www.u-blox.com

Document Information	
Title	NEO-6
Subtitle	u-blox 6 GPS Modules
Document type	Data Sheet
Document number	GPS.G6-HW-09005-E
Document status	

Document status information	
Objective Specification	This document contains target values. Revised and supplementary data will be published later.
Advance Information	This document contains data based on early testing. Revised and supplementary data will be published later.
Preliminary	This document contains data from product verification. Revised and supplementary data may be published later.
Released	This document contains the final product specification.

This document applies to the following products:

Name	Type number	ROM/FLASH version	PCN reference
NEO-6G	NEO-6G-0-001	ROM7.03	UBX-TN-11047-1
NEO-6Q	NEO-6Q-0-001	ROM7.03	UBX-TN-11047-1
NEO-6M	NEO-6M-0-001	ROM7.03	UBX-TN-11047-1
NEO-6P	NEO-6P-0-000	ROM6.02	N/A
NEO-6V	NEO-6V-0-000	ROM7.03	N/A
NEO-6T	NEO-6T-0-000	ROM7.03	N/A

This document and the use of any information contained therein, is subject to the acceptance of the u-blox terms and conditions. They can be downloaded from www.u-blox.com.
u-blox makes no warranties based on the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Reproduction, use or disclosure to third parties without express permission is strictly prohibited. Copyright © 2011, u-blox AG.
u-blox® is a registered trademark of u-blox Holding AG in the EU and other countries. ARM® is the registered trademark of ARM Limited in the EU and other countries.

Contents

Contents 3

1 Functional description 5
1.1 Overview 5
1.2 Product features 5
1.3 GPS performance 6
1.4 Block diagram 7
1.5 Assisted GPS (A-GPS) 7
1.6 AssistNow Autonomous 7
1.7 Precision Timing 8
1.7.1 Time mode 8
1.7.2 Timepulse and frequency reference 8
1.7.3 Time mark 8
1.8 Raw data 8
1.9 Automotive Dead Reckoning 8
1.10 Precise Point Positioning 9
1.11 Oscillators 9
1.12 Protocols and interfaces 9
1.12.1 UART 9
1.12.2 USB 9
1.12.3 Serial Peripheral Interface (SPI) 9
1.12.4 Display Data Channel (DDC) 10
1.13 Antenna 10
1.14 Power Management 10
1.14.1 Maximum Performance Mode 10
1.14.2 Eco Mode 10
1.14.3 Power Save Mode 11
1.15 Configuration 11
1.15.1 Boot-time configuration 11
1.16 Design-in 11

2 Pin Definition 12
2.1 Pin assignment 12

3 Electrical specifications 14
3.1 Absolute maximum ratings 14
3.2 Operating conditions 15
3.3 Indicative power requirements 15
3.4 SPI timing diagrams 16
3.4.1 Timing recommendations 16

4 Mechanical specifications 17

5 Qualification and certification 18
5.1 Reliability tests 18
5.2 Approvals 18

6 Product handling & soldering 19
6.1 Packaging 19
6.1.1 Reels 19
6.1.1 Tapes 20
6.2 Moisture Sensitivity Levels 21
6.3 Reflow soldering 21
6.4 ESD handling precautions 21

7 Default settings 22

8 Labeling and ordering information 23
8.1 Product labeling 23
8.2 Explanation of codes 23
8.3 Ordering information 24

Related documents 24

Revision history 24

Contact 25

1 Functional description

1.1 Overview

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.

The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

1.2 Product features

Model	Type					Supply		Interfaces				Features						
	GPS	PPP	Timing	Raw Data	Dead Reckoning	1.75 V - 2.0 V	2.7 V - 3.6 V	UART	USB	SPI	DDC (I^2C compliant)	Programmable (Flash) FW update	TCXO	RTC crystal	Antenna supply and supervisor	Configuration pins	Timepulse	External interrupt/ Wakeup
NEO-6G	●					●		●	●	●	●		●	●	O	3	1	●
NEO-6Q	●						●	●	●	●	●		●	●	O	3	1	●
NEO-6M	●						●	●	●	●	●			●	O	3	1	●
NEO-6P	●	●		●			●	●	●	●	●			●	O	3	1	●
NEO-6V	●				●		●	●	●	●	●			●	O	3	1	●
NEO-6T	●		●	●			●	●	●	●	●		●	●	O	3	1	●

O = Requires external components and integration on application processor

Table 1: Features of the NEO-6 Series

☞ All NEO-6 modules are based on GPS chips qualified according to AEC-Q100. See Chapter 5.1 for further information.

1.3 GPS performance

Parameter	Specification			
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix[1]		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Cold Start[2]	26 s	27 s	32 s
	Warm Start[2]	26 s	27 s	32 s
	Hot Start[2]	1 s	1 s	1 s
	Aided Starts[3]	1 s	<3 s	<3 s
Sensitivity[4]		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition[5]	-160 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-157 dBm	-156 dBm	-155 dBm
Maximum Navigation update rate		NEO-6G/Q/M/T	NEO-6P/V	
		5Hz	1 Hz	
Horizontal position accuracy[6]	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP[7]	< 1 m (2D, R50)[8)]		
	SBAS + PPP[7]	< 2 m (3D, R50)[8]		
Configurable Timepulse frequency range		NEO-6G/Q/M/P/V	NEO-6T	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated[9]	15 ns		
Velocity accuracy[6]		0.1m/s		
Heading accuracy[6]		0.5 degrees		
Operational Limits	Dynamics	≤ 4 g		
	Altitude[10]	50,000 m		
	Velocity[10]	500 m/s		

Table 2: NEO-6 GPS performance

[1] All satellites at -130 dBm
[2] Without aiding
[3] Dependent on aiding data connection speed and latency
[4] Demonstrated with a good active antenna
[5] For an outage duration ≤10s
[6] CEP, 50%, 24 hours static, -130dBm, SEP: <3.5m
[7] NEO-6P only
[8] Demonstrated under following conditions: 24 hours, stationary, first 600 seconds of data discarded. HDOP < 1.5 during measurement period, strong signals. Continuous availability of valid SBAS correction data during full test period.
[9] Quantization error information can be used with NEO-6T to compensate the granularity related error of the timepulse signal
[10] Assuming Airborne <4g platform

1.4 Block diagram

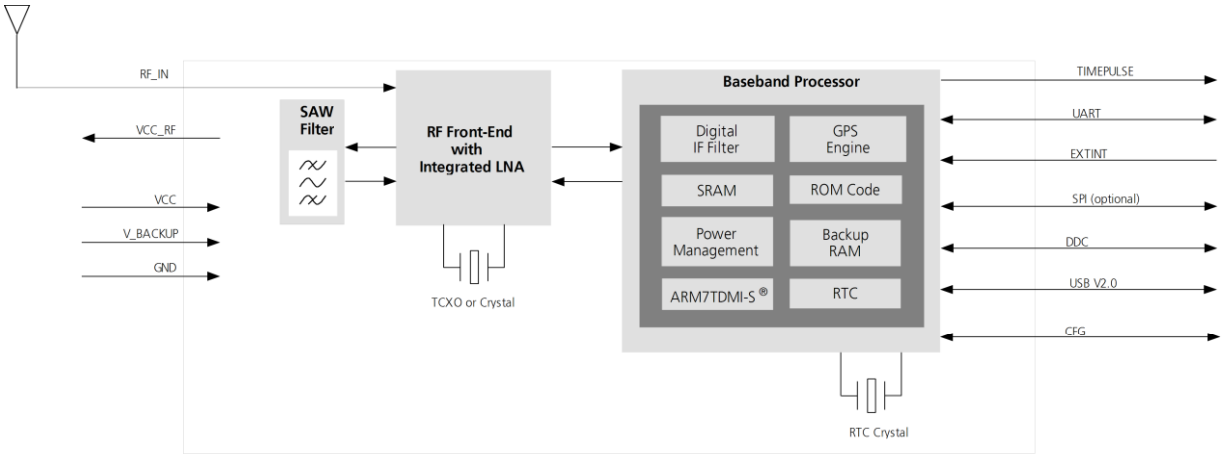


Figure 1: Block diagram (For available options refer to the product features table in section 1.2).

1.5 Assisted GPS (A-GPS)

Supply of aiding information like ephemeris, almanac, rough last position and time and satellite status and an optional time synchronization signal will reduce time to first fix significantly and improve the acquisition sensitivity. All NEO-6 modules support the u-blox AssistNow Online and AssistNow Offline A-GPS services[11] and are OMA SUPL compliant.

1.6 AssistNow Autonomous

AssistNow Autonomous provides functionality similar to Assisted-GPS without the need for a host or external network connection. Based on previously broadcast satellite ephemeris data downloaded to and stored by the GPS receiver, AssistNow Autonomous automatically generates accurate satellite orbital data ("AssistNow Autonomous data") that is usable for future GPS position fixes. AssistNow Autonomous data is reliable for up to 3 days after initial capture.

u-blox' AssistNow Autonomous benefits are:

- Faster position fix
- No connectivity required
- Complementary with AssistNow Online and Offline services
- No integration effort, calculations are done in the background

☞ For more details see the u-blox 6 Receiver Description including Protocol Specification [2].

[11] AssistNow Offline requires external memory.

1.7 Precision Timing

1.7.1 Time mode

NEO-6T provides a special Time Mode to provide higher timing accuracy. The NEO-6T is designed for use with stationary antenna setups. The Time Mode features three different settings described in Table 3: Disabled, Survey-In and Fixed Mode. For optimal performance entering the position of the antenna (when known) is recommended as potential source of errors will be reduced.

Time Mode Settings	Description
Disabled	Standard PVT operation
Survey-In	The GPS receiver computes the average position over an extended time period until a predefined maximum standard deviation has been reached. Afterwards the receiver will be automatically set to Fixed Mode and the timing features will be activated.
Fixed Mode	In this mode, a fixed 3D position and known standard deviation is assumed and the timing features are activated. Fixed Mode can either be activated directly by feeding pre-defined position coordinates (ECEF - Earth Center Earth Fixed format) or by performing a Survey-In. In Fixed mode, the timing errors in the TIMEPULSE signal which otherwise result from positioning errors are eliminated. Single-satellite operation is supported. For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

Table 3: Time mode settings

1.7.2 Timepulse and frequency reference

NEO-6T comes with a timepulse output which can be configured from 0.25 Hz up to 10 MHz. The timepulse can either be used for time synchronization (i.e. 1 pulse per second) or as a reference frequency in the MHz range. A timepulse in the MHz range provides excellent long-term frequency accuracy and stability.

1.7.3 Time mark

NEO-6T can be used for precise time measurements with sub-microsecond resolution using the external interrupt (EXTINT0). Rising and falling edges of these signals are time-stamped to the GPS or UTC time and counted. The Time Mark functionality can be enabled with the UBX-CFG-TM2 message

For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

1.8 Raw data

Raw data output is supported at an update rate of 5 Hz on the NEO-6T and NEO-6P. The UBX-RXM-RAW message includes carrier phase with half-cycle ambiguity resolved, code phase and Doppler measurements, which can be used in external applications that offer precision positioning, real-time kinematics (RTK) and attitude sensing.

1.9 Automotive Dead Reckoning

Automotive Dead Reckoning (ADR) is u-blox' industry proven off-the-shelf Dead Reckoning solution for tier-one automotive customers. u-blox' ADR solution combines GPS and sensor digital data using a tightly coupled Kalman filter. This improves position accuracy during periods of no or degraded GPS signal.

The NEO-6V provides ADR functionality over its software sensor interface. A variety of sensors (such as wheel ticks and gyroscope) are supported, with the sensor data received via UBX messages from the application processor. This allows for easy integration and a simple hardware interface, lowering costs. By using digital sensor data available on the vehicle bus, hardware costs are minimized since no extra sensors are required for Dead Reckoning functionality. ADR is designed for simple integration and easy configuration of different sensor options (e.g. with or without gyroscope) and vehicle variants, and is completely self-calibrating.

For more details contact the u-blox support representative nearest you to receive dedicated u-blox 6 Receiver Description Including Protocol Specification [3].

1.10 Precise Point Positioning

u-blox' industry proven PPP algorithm provides extremely high levels of position accuracy in static and slow moving applications, and makes the NEO-6P an ideal solution for a variety of high precision applications such as surveying, mapping, marine, agriculture or leisure activities.

Ionospheric corrections such as those received from local SBAS[12] geostationary satellites (WAAS, EGNOS, MSAS) or from GPS enable the highest positioning accuracy with the PPP algorithm. The maximum improvement of positioning accuracy is reached with PPP+SBAS and can only be expected in an environment with unobstructed sky view during a period in the order of minutes.

1.11 Oscillators

NEO-6 GPS modules are available in Crystal and TCXO versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

1.12 Protocols and interfaces

Protocol	Type
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0)
UBX	Input/output, binary, u-blox proprietary
RTCM	Input, 2.3

Table 4: Available protocols

All listed protocols are available on UART, USB, and DDC. For specification of the various protocols see the u-blox 6 Receiver Description including Protocol Specification [2].

1.12.1 UART

NEO-6 modules include one configurable UART interface for serial communication (for information about configuration see section 1.15).

1.12.2 USB

NEO-6 modules provide a USB version 2.0 FS (Full Speed, 12Mbit/s) interface as an alternative to the UART. The pull-up resistor on USB_DP is integrated to signal a full-speed device to the host. The VDDUSB pin supplies the USB interface. u-blox provides a Microsoft® certified USB driver for Windows XP, Windows Vista and Windows 7 operating systems.

1.12.3 Serial Peripheral Interface (SPI)

The SPI interface allows for the connection of external devices with a serial interface, e.g. serial flash to save configuration and AssistNow Offline A-GPS data or to interface to a host CPU. The interface can be operated in master or slave mode. In master mode, one chip select signal is available to select external slaves. In slave mode a single chip select signal enables communication with the host.

☞ The maximum bandwidth is 100kbit/s.

[12] Satellite Based Augmentation System

1.12.4 Display Data Channel (DDC)

The I²C compatible DDC interface can be used either to access external devices with a serial interface EEPROM or to interface with a host CPU. It is capable of master and slave operation. The DDC interface is I²C Standard Mode compliant. For timing parameters consult the I²C standard.

☞ The DDC Interface supports serial communication with u-blox wireless modules. See the specification of the applicable wireless module to confirm compatibility.

☞ The maximum bandwidth is 100kbit/s.

1.12.4.1 External serial EEPROM

NEO-6 modules allow an optional external serial EEPROM to be connected to the DDC interface. This can be used to store Configurations permanently.

☞ For more information see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

⚠ **Use caution when implementing since forward compatibility is not guaranteed.**

1.13 Antenna

NEO-6 modules are designed for use with passive and active[13] antennas.

Parameter	Specification	
Antenna Type		Passive and active antenna
Active Antenna Recommendations	Minimum gain	15 dB (to compensate signal loss in RF cable)
	Maximum gain	50 dB
	Maximum noise figure	1.5 dB

Table 5: Antenna Specifications for all NEO-6 modules

1.14 Power Management

u-blox receivers support different power modes. These modes represent strategies of how to control the acquisition and tracking engines in order to achieve either the best possible performance or good performance with reduced power consumption.

☞ For more information about power management strategies, see the u-blox 6 Receiver Description including Protocol Specification [2].

1.14.1 Maximum Performance Mode

During a Cold start, a receiver in Maximum Performance Mode continuously deploys the acquisition engine to search for all satellites. Once the receiver has a position fix (or if pre-positioning information is available), the acquisition engine continues to be used to search for all visible satellites that are not being tracked.

1.14.2 Eco Mode

During a Cold start, a receiver in Eco Mode works exactly as in Maximum Performance Mode. Once a position can be calculated and a sufficient number of satellites are being tracked, the acquisition engine is powered off resulting in significant power savings. The tracking engine continuously tracks acquired satellites and acquires other available or emerging satellites.

☞ Note that even if the acquisition engine is powered off, satellites continue to be acquired.

[13] For information on using active antennas with NEO-6 modules, see the LEA-6/NEO-6 Hardware Integration Manual [1].

1.14.3 Power Save Mode

Power Save Mode (PSM) allows a reduction in system power consumption by selectively switching parts of the receiver on and off.

Power Save mode is not available with NEO-6P, NEO-6T and NEO-6V.

1.15 Configuration

1.15.1 Boot-time configuration

NEO-6 modules provide configuration pins for boot-time configuration. These become effective immediately after start-up. Once the module has started, the configuration settings can be modified with UBX configuration messages. The modified settings remain effective until power-down or reset. If these settings have been stored in battery-backup RAM, then the modified configuration will be retained, as long as the backup battery supply is not interrupted.

NEO-6 modules include both **CFG_COM0** and **CFG_COM1** pins and can be configured as seen in Table 6. Default settings in bold.

CFG_COM1	CFG_COM0	Protocol	Messages	UARTBaud rate	USB power
1	**1**	**NMEA**	**GSV, RMC, GSA, GGA, GLL, VTG, TXT**	**9600**	**BUS Powered**
1	0	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT	38400	Self Powered
0	1	NMEA	GSV[14], RMC, GSA, GGA, VTG, TXT	4800	BUS Powered
0	0	UBX	NAV-SOL, NAV-STATUS, NAV-SVINFO, NAV-CLOCK, INF, MON-EXCEPT, AID-ALPSERV	57600	BUS Powered

Table 6: Supported COM settings

NEO-6 modules include a **CFG_GPS0** pin, which enables the boot-time configuration of the power mode. These settings are described in Table 7. Default settings in bold.

CFG_GPS0	Power Mode
0	Eco Mode
1	**Maximum Performance Mode**

Table 7: Supported CFG_GPS0 settings

Static activation of the **CFG_COM** and **CFG_GPS** pins is not compatible with use of the SPI interface.

1.16 Design-in

In order to obtain the necessary information to conduct a proper design-in, u-blox strongly recommends consulting the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

[14] Every 5th fix.

2 Pin Definition

2.1 Pin assignment

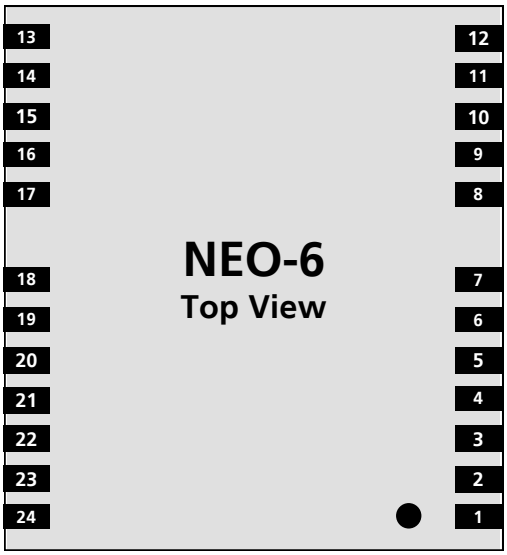


Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SPI Slave Select
3	All	TIMEPULSE	O	Timepulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage RF section Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MOSI/CFG_COM0	O/I	SPI MOSI / Configuration Pin. Leave open if not used.
15	All	MISO/CFG_COM1	I	SPI MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPS0/SCK	I	Power Mode Configuration Pin / SPI Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	DDC Data
19	All	SCL2	I/O	DDC Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1

No	Module	Name	I/O	Description
22	All	V_BCKP	I	Backup voltage supply
23	All	VCC	I	Supply voltage
24	All	GND	I	Ground

Table 8: Pinout

☞ Pins designated Reserved should not be used. For more information about Pinouts see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

3 Electrical specifications

3.1 Absolute maximum ratings

Parameter	Symbol	Module	Min	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	-0.5	2.0	V	
		NEO-6Q, 6M, 6P, 6V, 6T	-0.5	3.6	V	
Backup battery voltage	V_BCKP	All	-0.5	3.6	V	
USB supply voltage	VDDUSB	All	-0.5	3.6	V	
Input pin voltage	Vin	All	-0.5	3.6	V	
	Vin_usb	All	-0.5	VDDUSB	V	
DC current trough any digital I/O pin (except supplies)	Ipin			10	mA	
VCC_RF output current	ICC_RF	All		100	mA	
Input power at RF_IN	Prfin	NEO-6Q, 6M, 6G, 6V, 6T		15	dBm	source impedance = 50Ω, continuous wave
		NEO-6P		-5	dBm	
Storage temperature	Tstg	All	-40	85	°C	

Table 9: Absolute maximum ratings

GPS receivers are Electrostatic Sensitive Devices (ESD) and require special precautions when handling. For more information see chapter 6.4.

Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. The product is not protected against overvoltage or reversed voltages. If necessary, voltage spikes exceeding the power supply voltage specification, given in table above, must be limited to values within the specified boundaries by using appropriate protection diodes. For more information see the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].

3.2 Operating conditions

☞ All specifications are at an ambient temperature of 25°C.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	1.75	1.8	1.95	V	
		NEO-6Q/M NEO-6P/V/T	2.7	3.0	3.6	V	
Supply voltage USB	VDDUSB	All	3.0	3.3	3.6	V	
Backup battery voltage	V_BCKP	All	1.4		3.6	V	
Backup battery current	I_BCKP	All		22		µA	V_BCKP = 1.8 V, VCC = 0V
Input pin voltage range	Vin	All	0		VCC	V	
Digital IO Pin Low level input voltage	Vil	All	0		0.2*VCC	V	
Digital IO Pin High level input voltage	Vih	All	0.7*VCC		VCC	V	
Digital IO Pin Low level output voltage	Vol	All			0.4	V	Iol=4mA
Digital IO Pin High level output voltage	Voh	All	VCC -0.4			V	Ioh=4mA
USB_DM, USB_DP	VinU	All	Compatible with USB with 22 Ohms series resistance				
VCC_RF voltage	VCC_RF	All		VCC-0.1		V	
VCC_RF output current	ICC_RF	All			50	mA	
Antenna gain	Gant	All			50	dB	
Receiver Chain Noise Figure	NFtot	All		3.0		dB	
Operating temperature	Topr	All	-40		85	°C	

Table 10: Operating conditions

☞ Operation beyond the specified operating conditions can affect device reliability.

3.3 Indicative power requirements

Table 11 lists examples of the total system supply current for a possible application.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Max. supply current [15]	Iccp	All			67	mA	VCC = 3.6 V[16] / 1.95 V[17]
Average supply current[18]	Icc Acquisition	All		47[19]		mA	VCC = 3.0 V[16] / 1.8 V[17]
	Icc Tracking (Max Performance mode)	NEO-6G/Q/T		40[20]		mA	
		NEO-6M/P/V		39[20]		mA	
	Icc Tracking (Eco mode)	NEO-6G/Q/T		38[20]		mA	
		NEO-6M/P/V		37[20]		mA	
	Icc Tracking (Power Save mode / 1 Hz)	NEO-6G/Q		12[20]		mA	
		NEO-6M		11[20]		mA	

Table 11: Indicative power requirements

☞ Values in Table 11 are provided for customer information only as an example of typical power requirements. Values are characterized on samples, actual power requirements can vary depending on FW version used, external circuitry, number of SVs tracked, signal strength, type of start as well as time, duration and conditions of test.

[15] Use this figure to dimension maximum current capability of power supply. Measurement of this parameter with 1 Hz bandwidth.
[16] NEO-6Q, NEO-6M, NEO-6P, NEO-6V, NEO-6T
[17] NEO-6G
[18] Use this figure to determine required battery capacity.
[19] >8 SVs in view, CNo >40 dBHz, current average of 30 sec after cold start.
[20] With strong signals, all orbits available. For Cold Starts typical 12 min after first fix. For Hot Starts typical 15 s after first fix.

3.4 SPI timing diagrams

In order to avoid a faulty usage of the SPI, the user needs to comply with certain timing conditions. The following signals need to be considered for timing constraints:

Symbol	Description
SS_N	Slave Select signal
SCK	Slave Clock signal

Table 12: Symbol description

Figure 3: SPI timing diagram

3.4.1 Timing recommendations

Parameter	Description	Recommendation
t_{INIT}	Initialization Time	500 µs
t_{DES}	Deselect Time	1 ms
Bitrate		100 kbit/s

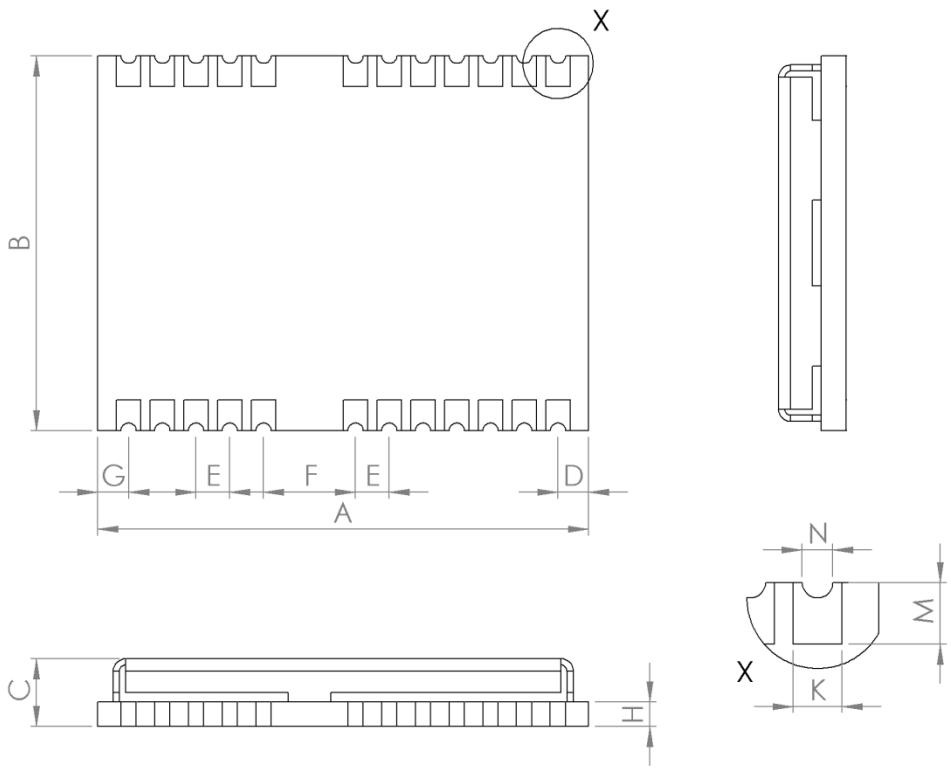
Table 13: SPI timing recommendations

☞ The values in the above table result from the requirement of an error-free transmission. By allowing just a few errors, the byte rate could be increased considerably. These timings – and therefore the byte rate – could also be improved by disabling other interfaces, e.g. the UART.

☞ The maximum bandwidth is 100 kbit/s[21].

[21] This is a theoretical maximum, the protocol overhead is not considered.

4 Mechanical specifications



Symbol	Min. (mm)	Typ. (mm)	Max. (mm)
A	15.9	16.0	16.6
B	12.1	12.2	12.3
C	2.2	2.4	2.6
D	0.9	1.0	1.3
E	1.0	1.1	1.2
F	2.9	3.0	3.1
G	0.9	1.0	1.3
H		0.82	
K	0.7	0.8	0.9
M	0.8	0.9	1.0
N	0.4	0.5	0.6
Weight		1.6g	

Figure 4: Dimensions

☞ For information regarding the Paste Mask and Footprint see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

5 Qualification and certification

5.1 Reliability tests

☞ All NEO-6 modules are based on AEC-Q100 qualified GPS chips.

Tests for product family qualifications according to ISO 16750 "Road vehicles - Environmental conditions and testing for electrical and electronic equipment", and appropriate standards.

5.2 Approvals

Products marked with this lead-free symbol on the product label comply with the "Directive 2002/95/EC of the European Parliament and the Council on the Restriction of Use of certain Hazardous Substances in Electrical and Electronic Equipment" (RoHS).

All u-blox 6 GPS modules are RoHS compliant.

6 Product handling & soldering

6.1 Packaging

NEO-6 modules are delivered as hermetically sealed, reeled tapes in order to enable efficient production, production lot set-up and tear-down. For more information about packaging, see the u-blox Package Information Guide [4].

Figure 5: Reeled u-blox 6 modules

6.1.1 Reels

NEO-6 GPS modules are deliverable in quantities of 250pcs on a reel. NEO-6 modules are delivered using reel Type B as described in the u-blox Package Information Guide [4].

Parameter	Specification
Reel Type	B
Delivery Quantity	250

Table 14: Reel information for NEO-6 modules

6.1.1 Tapes

Figure 6 shows the position and orientation of NEO-6 modules as they are delivered on tape. The dimensions of the tapes are specified in Figure 7.

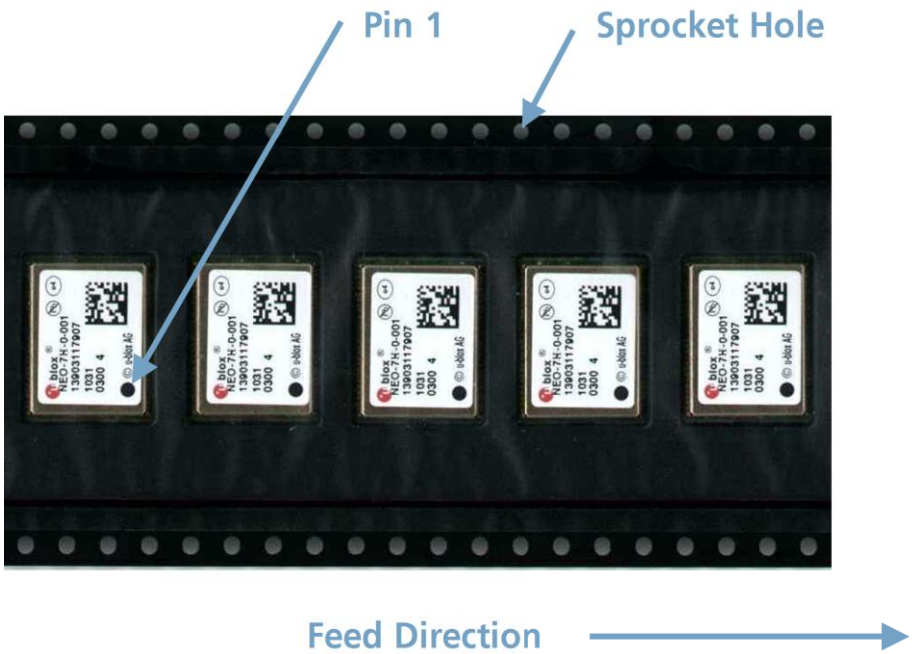


Figure 6: Orientation for NEO-6 modules on tape

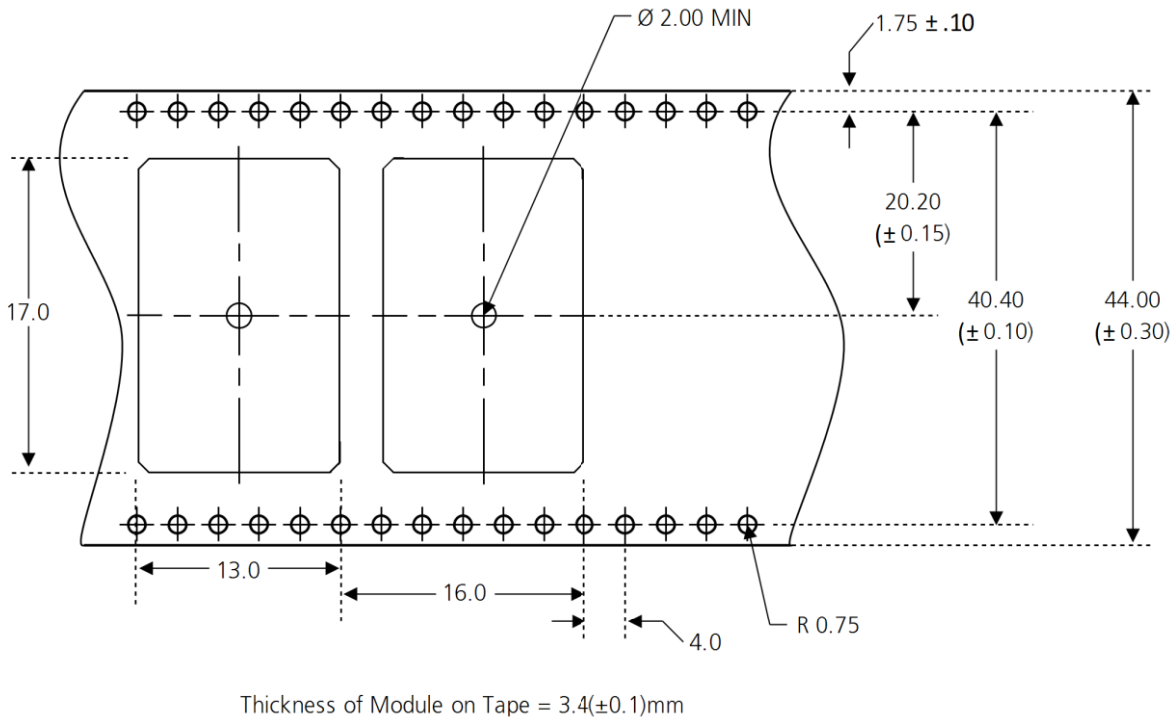


Figure 7: NEO tape dimensions (mm)

6.2 Moisture Sensitivity Levels

NEO-6 modules are Moisture Sensitive Devices (MSD) in accordance to the IPC/JEDEC specification.

NEO-6 modules are rated at MSL level 4. For more information regarding moisture sensitivity levels, labeling, storage and drying see the u-blox Package Information Guide [4].

For MSL standard see IPC/JEDEC J-STD-020, which can be downloaded from www.jedec.org.

6.3 Reflow soldering

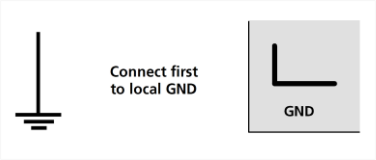
Reflow profiles are to be selected according to u-blox recommendations (see LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1]).

6.4 ESD handling precautions

NEO-6 modules contain highly sensitive electronic circuitry and are Electrostatic Sensitive Devices (ESD). Observe precautions for handling! Failure to observe these precautions can result in severe damage to the GPS receiver!

GPS receivers are Electrostatic Sensitive Devices (ESD) and require special precautions when handling. Particular care must be exercised when handling patch antennas, due to the risk of electrostatic charges. In addition to standard ESD safety practices, the following measures should be taken into account whenever handling the receiver:

- Unless there is a galvanic coupling between the local GND (i.e. the work table) and the PCB GND, then the first point of contact when handling the PCB must always be between the local GND and PCB GND.
- Before mounting an antenna patch, connect ground of the device



- When handling the RF pin, do not come into contact with any charged capacitors and be careful when contacting materials that can develop charges (e.g. patch antenna ~10pF, coax cable ~50-80pF/m, soldering iron, …)

- To prevent electrostatic discharge through the RF input, do not touch any exposed antenna area. If there is any risk that such exposed antenna area is touched in non ESD protected work area, implement proper ESD protection measures in the design.

- When soldering RF connectors and patch antennas to the receiver's RF pin, make sure to use an ESD safe soldering iron (tip).

7 Default settings

Interface	Settings
Serial Port 1 Output	9600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (In addition to the 6 standard NMEA messages the NEO-6T includes **ZDA**).
USB Output	Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (In addition to the 6 standard NMEA messages the NEO-6T includes **ZDA**). USB Power Mode: Bus-Powered
Serial Port 1 Input	9600 Baud, 8 bits, no parity bit, 1 stop bit Automatically accepts following protocols without need of explicit configuration: **UBX, NMEA** The GPS receiver supports interleaved UBX and NMEA messages.
USB Input	Automatically accepts following protocols without need of explicit configuration: **UBX, NMEA** The GPS receiver supports interleaved UBX and NMEA messages. USB Power Mode: Bus-Powered
TIMEPULSE (1Hz Nav)	1 pulse per second, synchronized at rising edge, pulse length 100ms
Power Mode	Maximum Performance mode
AssistNow Autonomous	Disabled.

Table 15: Default settings

Refer to the u-blox 6 Receiver Description including Protocol Specification [2] for information about further settings.

8 Labeling and ordering information

8.1 Product labeling

The labeling of u-blox 6 GPS modules includes important product information. The location of the product type number is shown in Figure 8.

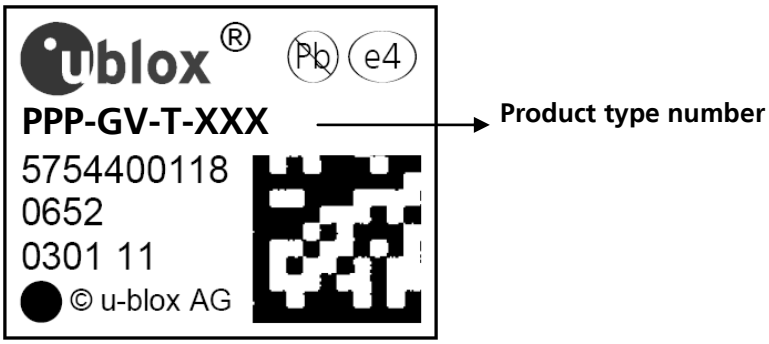


Figure 8: Location of product type number on u-blox 6 module label

8.2 Explanation of codes

3 different product code formats are used. The **Product Name** is used in documentation such as this data sheet and identifies all u-blox 6 products, independent of packaging and quality grade. The **Ordering Code** includes options and quality, while the **Type Number** includes the hardware and firmware versions. Table 16 below details these 3 different formats:

Format	Structure
Product Name	PPP-GV
Ordering Code	PPP-GV-T
Type Number	PPP-GV-T-XXX

Table 16: Product Code Formats

The parts of the product code are explained in Table 17.

Code	Meaning	Example
PPP	Product Family	NEO
G	Product Generation	6 = u-blox6
V	Variant	T = Timing, R = DR, etc.
T	Option / Quality Grade	Describes standardized functional element or quality grade such as Flash size, automotive grade etc.
XXX	Product Detail	Describes product details or options such as hard- and software revision, cable length, etc.

Table 17: part identification code

8.3 Ordering information

Ordering No.	Product
NEO-6G-0	u-blox 6 GPS Module, 1.8V, TCXO, 12x16mm, 250 pcs/reel
NEO-6M-0	u-blox 6 GPS Module, 12x16mm, 250 pcs/reel
NEO-6Q-0	u-blox 6 GPS Module, TCXO, 12x16mm, 250 pcs/reel
NEO-6P-0	u-blox 6 GPS Module, PPP, 12x16mm, 250 pcs/reel
NEO-6V-0	u-blox 6 GPS Module, Dead Reckoning SW sensor, 12x16mm, 250 pcs/reel
NEO-6T-0	u-blox 6 GPS Module, Precision Timing, TCXO, 12x16mm, 250 pcs/reel

Table 18: Product Ordering Codes

☞ Product changes affecting form, fit or function are documented by u-blox. For a list of Product Change Notifications (PCNs) see our website at: http://www.u-blox.com/en/notifications.html

Related documents

[1] LEA-6/NEO-6/MAX-6 Hardware Integration Manual, Docu. GPS.G6-HW-09007

[2] u-blox 6 Receiver Description Including Protocol Specification (Public version), Docu. No. GPS.G6-SW-10018

[3] u-blox 6 Receiver Description Including Protocol Specification (Confidential version), Docu. No. GPS.G6-SW-10019

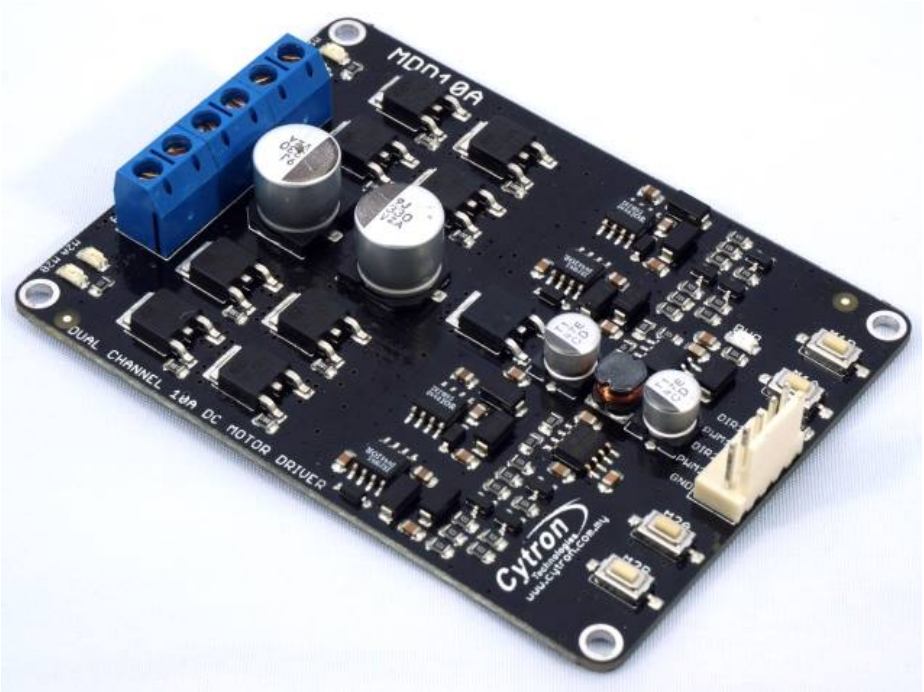
[4] u-blox Package Information Guide, Docu. No GPS-X-11004

☞ For regular updates to u-blox documentation and to receive product change notifications please register on our homepage.

Revision history

Revision	Date	Name	Status / Comments
	31/08/2009	tgri	Initial Version
1	21/09/2009	tgri	update of section 1.3 GPS performance, section 1.4 block diagram, section 3.2 peak supply current
A	25/02/2010	tgri	Change of status to Advance Information. Addition of NEO-6G. Update of section 1.8.2, removed reference to Vddio – added USB driver certification. Update of section 3.2 table 11: average supply current, Added section 3.3-3.4: SPI & DDC timing, section 5.1: addition of table 12.
B	24/06/2010	dhur	Change of status to Preliminary. Update of section 1.2, 1.8.4, 1.10.4, 3.1, 3.2 and chapter 2 and 4. General clean-up and consistency check.
B1	11/08/2010	dhur	Replaced graphic in figure 2.
C	18/07/2011	dhur	Added chapter 1.6, update to FW7.03.
D	19/10/2011	dhur	Added NEO-6P and NEO-6V. Added chapter 1.7 and 1.8. Revised Chapter 6.
E	05/12/2011	dhur	Added NEO-6T. Added chapter 1.7 and 1.8. Added Accuracy for Timepulse signal in Table 2. Corrected Maximum Input power at RF_IN for NEO-6P in Table 9.

MDD10A
Dual Channel 10A DC Motor Driver



User's Manual

V1.0

Dec 2013

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

INDEX

1. Introduction/Overview 3
2. Packing List 4
3. Product Specification and Limitation 5
4. Dimension 6
5. Board Layout 7
6. Getting Started 9
7. Warranty 10

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

1. INTRODUCTION/OVERVIEW

MDD10A is the dual channel version of MD10C which is designed to drive 2 brushed DC motors with high current up to 10A continuously. Just like MD10C, the MDD10A also supports locked-antiphase and sign-magnitude PWM signal. It is also using full solid state components which result in faster response time and eliminate the wear and tear of the mechanical relay.

MDD10A has been designed with the capabilities and features of:

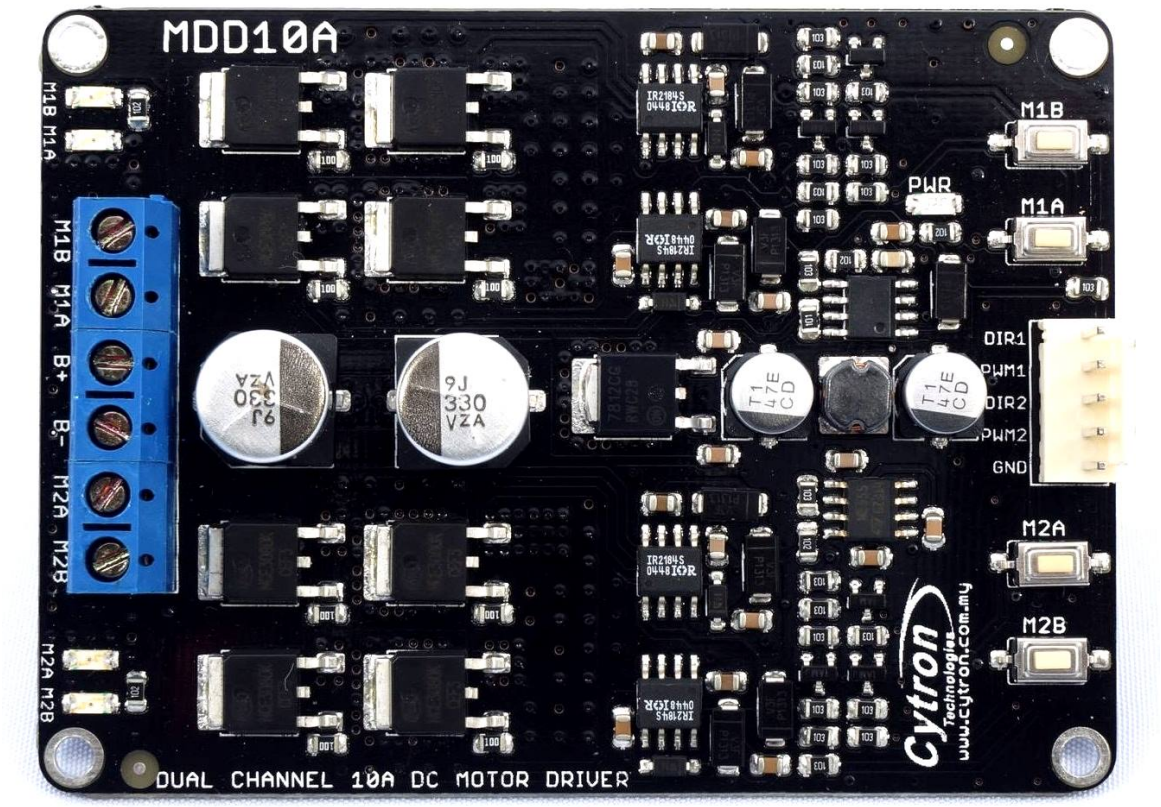
- Bi-directional control for 2 brushed DC motors.
- Support motor voltage ranges from 5V to 25V.
- Maximum current up to 10A continuous and 30A peak (10 second) for each channel.
- Solid state components provide faster response time and eliminate the wear and tear of mechanical relay.
- Fully NMOS H-Bridge for better efficiency and no heat sink is required.
- Speed control PWM frequency up to 20KHz.
- Support both locked-antiphase and sign-magnitude PWM operation.
- Onboard push button to control the motor manually.

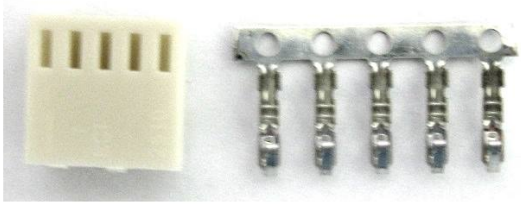
Dimension: **84.5mm x 62mm**

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

2. PACKING LIST

Please check the parts and components according to the packing list. If there are any parts missing, please contact us at sales@cytron.com.my immediately.





1. 1 x MDD10A Dual Channel 10A DC Motor Driver
2. 1 x 2510 PCB Connector - 5 Ways (Female)
3. 5 x 2510 Iron Pin
4. User's manual can be downloaded from http://www.cytron.com.my

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

3. PRODUCT SPECIFICATION AND LIMITATIONS

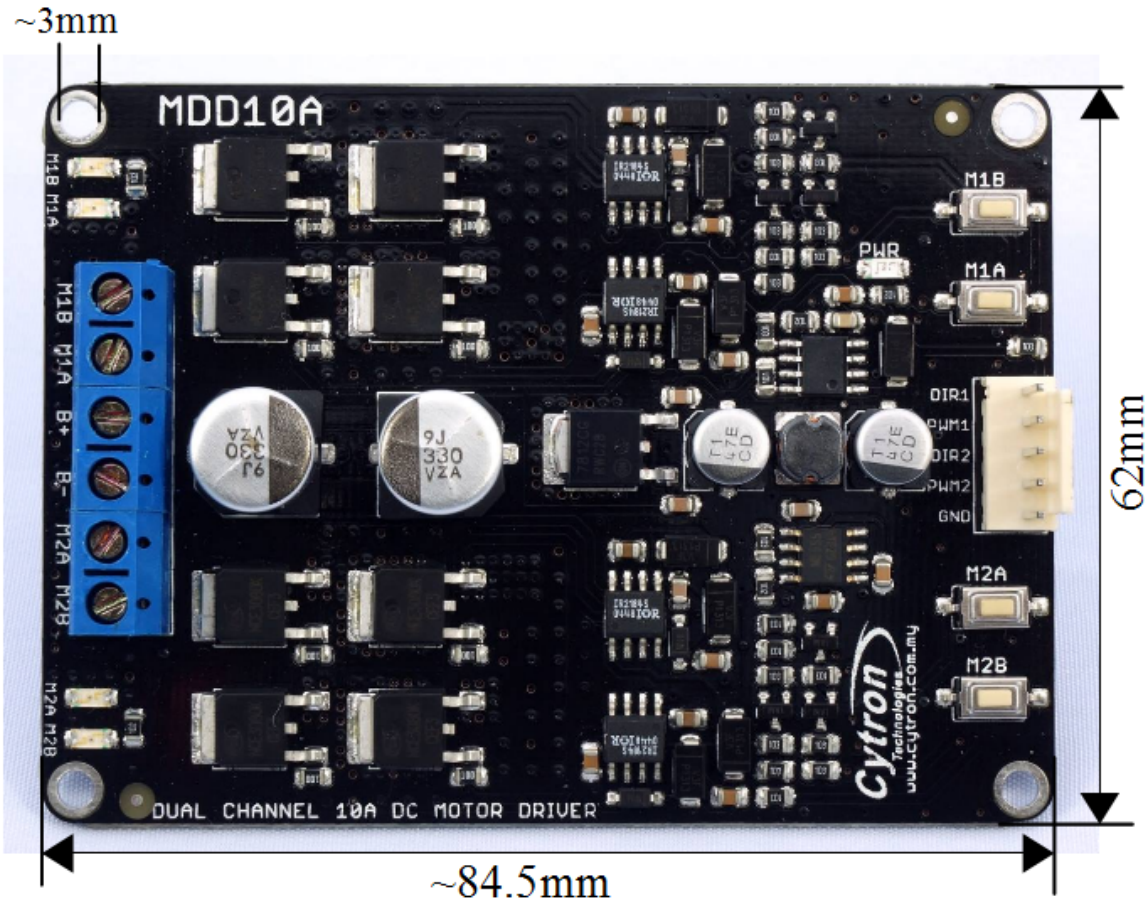
Absolute Maximum Rating

No.	Parameters	Min	Typical	Max	Unit
1	Power Input Voltage	5	-	25	V
2	I_{MAX} (Maximum Continuous Motor Curren	-	-	10	A
3	I_{PEAK} – (Peak Motor Current) *	-	-	30	A
4	V_{IOH} (Logic Input – High Level)	3	-	5.5	V
5	V_{IOL} (Logic Input – Low Level)	0	0	0.5	V
6	Maximum PWM Frequency	-	-	20	KHz

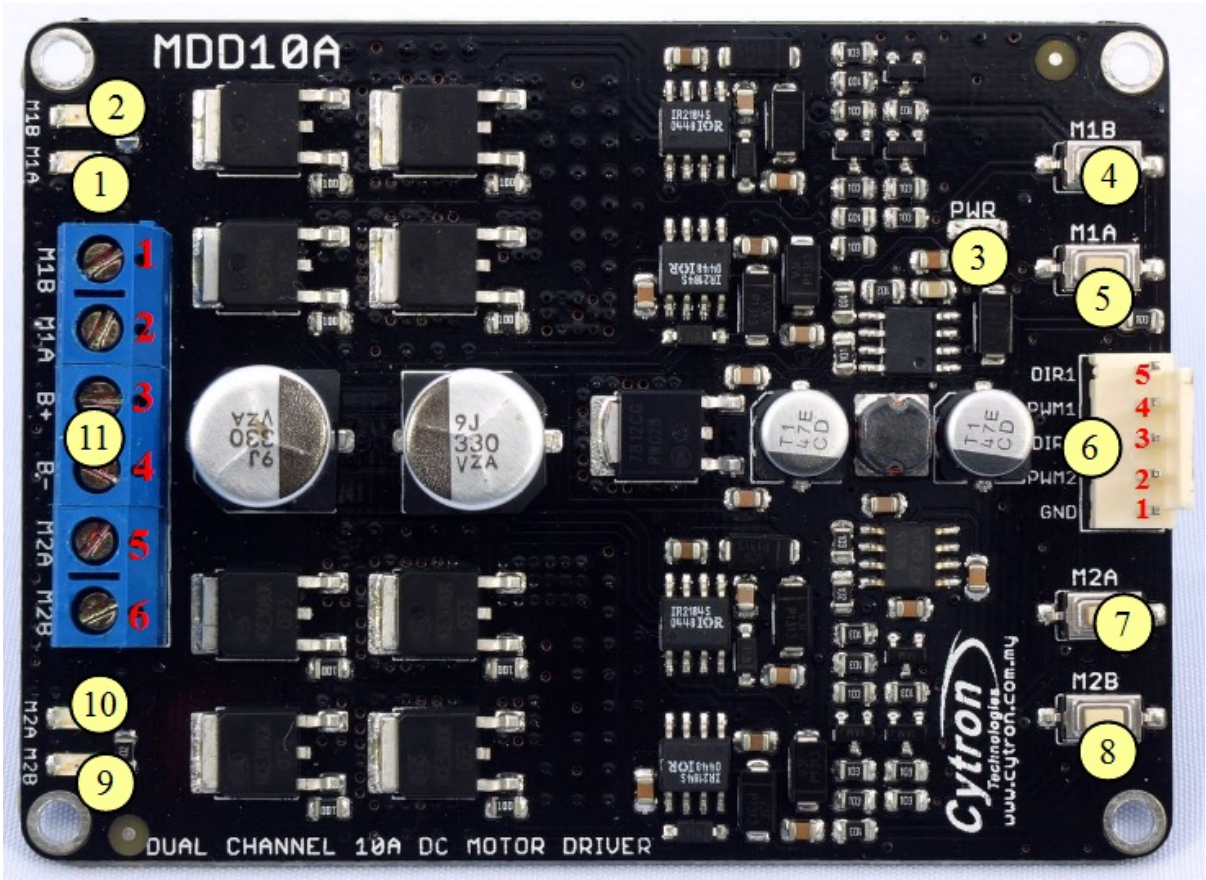
**** Must not exceed 10 seconds.***

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

4. DIMENSION


~3mm
MDD10A
M1B
M1A
PWR
DIR1
PWM1
DIR2
PWM2
GND
M2A
M2B
B+
B-
Cytron
Technologies
www.cytron.com.my
DUAL CHANNEL 10A DC MOTOR DRIVER
62mm
~84.5mm

5. BOARD LAYOUT



1. Red LED M1A – Turns on when the output M1A is high and output M1B is low. Indicates the current flows from output M1A to M1B.

2. Red LED M1B – Turns on when the output M1A is low and output M1B is high. Indicates the current flows from output M1B to M1A.

3. Green LED – Power LED. Should be on when the board is powered on.

4. Test Button M1B – When this button is pressed, current flows from output M1B to M1A and motor will turn CCW (or CW depending on the connection).

5. Test Button M1A – When this button is pressed, current flows from output M1A to M1B and motor will turn CW (or CCW depending on the connection).

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

6. Input

Pin No.	Pin Name	Description
1	GND	Ground
2	PWM2	PWM input for speed control (Motor 2)
3	DIR2	Direction input (Motor 2)
4	PWM1	PWM input for speed control (Motor 1)
5	DIR1	Direction input (Motor 1)

The truth table for the control logic for motor 1 and motor 2 are as follow:

PWM	DIR	Output A	Output B
Low	X(Don't care)	Low	Low
High	Low	High	Low
High	High	Low	High

7. Test Button M2A – When this button is pressed, current flows from output M2A to M2B and motor will turn CW (or CCW depending on the connection).

8. Test Button M2B – When this button is pressed, current flows from output M2B to M2A and motor will turn CCW (or CW depending on the connection).

9. Red LED M2B – Turns on when the output M2A is low and output M2B is high. Indicates the current flows from output M2B to M2A.

10. Red LED M2A – Turns on when the output M2A is high and output M2B is low. Indicates the current flows from output M2A to M2B.

11. Terminal Block – Connect to motor and power source.

Pin No	Pin Name	Description
1	Motor 1 Output B	Connect to motor 1 terminal B
2	Motor 1 Output A	Connect to motor 1 terminal A
3	POWER +	Positive Supply
4	POWER -	Negative Supply
5	Motor 2 Output A	Connect to motor 2 terminal A
6	Motor 2 Output B	Connect to motor 2 terminal B

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

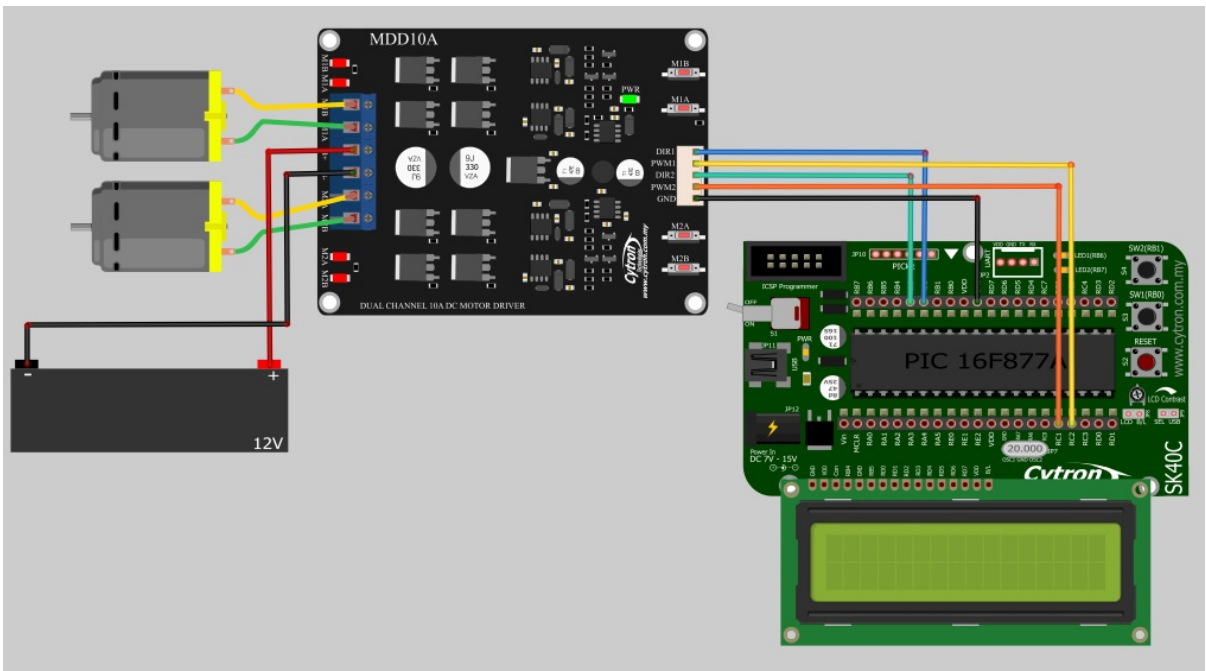
6. GETTING STARTED

6.1 Getting Started MDD10A with SK40C

MDD10A is compatible with 2 types of PWM operation, which are:

1. Sign-Magnitude PWM – For sign-magnitude PWM operation, 2 control signals are used to control the speed and direction of the motor. PWM is feed to the PWM pin to control the speed while DIR pin is used to control the direction of the motor.

2. Locked-Antiphase PWM – For locked-antiphase PWM operation, only 1 control signal is needed to control the speed and direction of the motor. PWM pin is connected to logic high while the DIR pin is being feed with the PWM signal. When the PWM signal has 50% duty cycle, the motor stops running. If the PWM has less than 50% duty cycle, the motor will turn CW (or CCW depending on the connection). If the PWM signal has more than 50% duty cycle, motor will turn CCW (or CW depending on the connection).



MOTOR
MOTEUR 12V – 20W
C-6044

Technical Characteristics

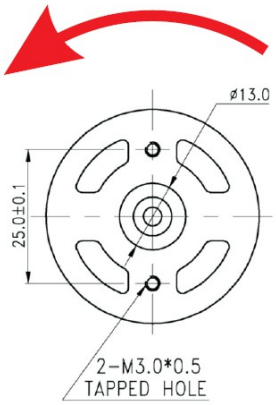
Nominal voltage	7 V
Working Range	6 - 12 V
No load velocity	20400 rpm
No-load current	max 1,80 A
Speed at max. efficiency	19500 rpm ±10%
Current at max. efficiency	4,70 A
Torque	261 g·cm
Output power	47 W
Torque limit	1850 g·cm
Noise	70 dB a 30 cm
Rotation	counterclockwise (viewed from theshaft end)
Approximate weight	150 g

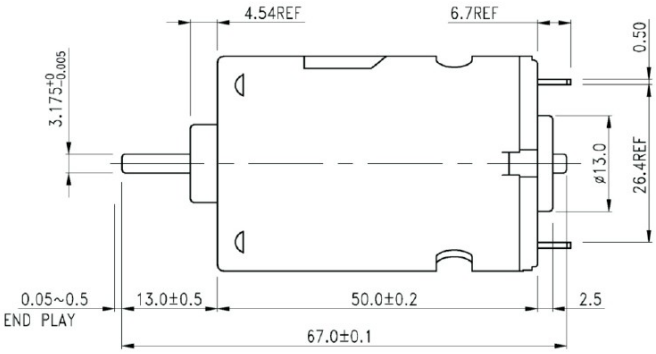
- Motor powerful 3-pole. Specific for professional model makers, small industrial and robotics applications.

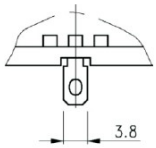
- Moteur puissant à 3 pôles. Spécifique pour les modélistes professionnels, applications industrielles et de la robotique .

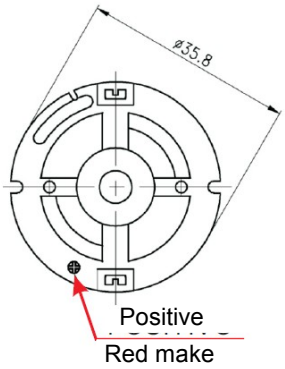
- Motor de 3 polos muy potente. Específico para maquetistas profesionales, pequeñas aplicaciones industriales y robótica.

Counterclockwise rotation direction
Sens inverse des aiguilles d'une rotation
Sentido rotación antihorario









Cebekit ® is a registered trademark of the Fadisel group

www.cebekit.com - info@cebekit.com