

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Mantenimiento Eléctrico

**IMPLEMENTACIÓN DE UN PROTOTIPO DE VEHÍCULO TERRESTRE NO
TRIPULADO AUTÓNOMO**

Trabajo de Grado presentado ante la Ilustre Universidad Técnica del Norte previo a
la obtención del título de Ingeniero en Mantenimiento Eléctrico.

Autor:

Carlosama Fauta Galo Israel

Director:

Ing. Naranjo Cobo Francisco Roberto MSc.

Ibarra – Ecuador

2020 – 2021



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004144182		
APELLIDOS Y NOMBRES:	Carlosama Fauta Galo Israel		
DIRECCIÓN:	Ibarra		
EMAIL:	gicarlosamaf@utn.edu.ec		
TELÉFONO FIJO:	062 580-162	TELÉFONO MÓVIL:	0999971251

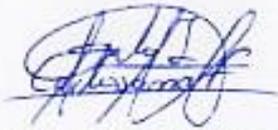
DATOS DE LA OBRA	
TÍTULO:	IMPLEMENTACIÓN DE UN PROTOTIPO DE VEHÍCULO TERRESTRE NO TRIPULADO AUTÓNOMO
AUTOR (ES):	Carlosama Fauta Galo Israel
FECHA: DD/MM/AAAA	28/06/2021
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero en Mantenimiento Eléctrico
ASESOR /DIRECTOR:	Ing. Naranjo Cobo Francisco Roberto MSc.

2. CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 30 días del mes de junio del 2021

EL AUTOR:



Nombre: Carlosama Fauta Galo Israel

C.C: 100414418-2

ACEPTACIÓN DEL DIRECTOR

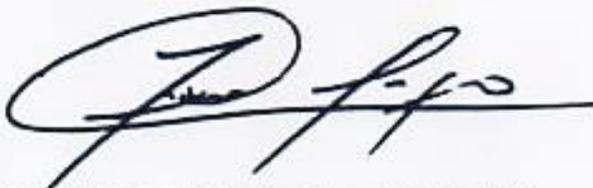
Ing. Francisco Roberto Naranjo Cobo MSc.

CERTIFICA

Que después de haber examinado el presente trabajo de investigación elaborado por el señor estudiante: Galo Israel Carlosama Fauta, certifico que ha cumplido con las normas establecidas en la elaboración del trabajo de investigación titulado: **"IMPLEMENTACIÓN DE UN PROTOTIPO DE VEHÍCULO TERRESTRE NO TRIPULADO AUTÓNOMO"** para la obtención del título de ingeniero en Mantenimiento Eléctrico.

En la ciudad de Ibarra, a los 28 días del mes de junio del 2021

Lo certifico:

A handwritten signature in black ink, appearing to read 'F. Naranjo Cobo', written over a horizontal line.

Ing. Francisco Roberto Naranjo Cobo MSc.

DIRECTOR DE TESIS

Dedicatoria

El presente trabajo de investigación lo dedicó principalmente a Dios, quién me brindó fuerzas para seguir adelante y no desmayar en las adversidades que se presentan y por tal razón haberme permitido llegar hasta este momento tan importante de mi formación profesional.

De manera especial a mis padres, por ser los principales promotores de este sueño hecho realidad, quienes, con su amor, trabajo y sacrificio en todos estos años, estuvieron apoyándome incondicionalmente en toda esta etapa de estudio.

A mi hermana, quién se ha convertido que mi mejor amiga, por apoyar mi condicional durante todo este proyecto y por estar conmigo en todo momento.

Agradecimiento

Le agradezco a Dios por haberme brindado la oportunidad de culminar con éxito mis estudios, por haberme acompañado y guiado a lo largo de mi carrera y por ser mi fortaleza en los momentos de debilidad.

Le doy gracias a mis padres por apoyarme en todo momento, por los valores que me han inculcado y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida.

También agradezco a los docentes de la carrera de Ingeniería en Mantenimiento Eléctrico de la Universidad Técnica del Norte, por haberme compartido sus conocimientos a lo largo de la preparación profesional, de igual manera a mi director de tesis Ing. Naranjo Cobo Francisco Roberto MSc, por la orientación, apoyo y la ayuda brindada para el desarrollo de esta tesis.

Tabla de Contenido

IDENTIFICACIÓN DE LA OBRA	II
CONSTANCIA	III
ACEPTACIÓN DEL DIRECTOR	IV
Dedicatoria	V
Agradecimiento.....	VI
Tabla de Contenido	VII
Índice de Imágenes	XI
Índice de Tablas	XIII
Resumen	XIV
Abstract	XV
Contexto	1
Planteamiento del problema	3
Objetivos	4
Objetivo General.....	4
Objetivos Específicos	4
Justificación	5
Alcance.....	6
Viabilidad.....	6
CAPÍTULO 1	
Marco Teórico.....	7
1.1 Vehículos terrestres no tripulados autónomos	7
1.1.1 Vehículos no tripulados terrestres tele operada o a control remoto.....	7
1.1.2 Vehículos no tripulados terrestres autónomo	8
1.2 Aplicaciones y usos de los vehículos terrestres no tripulados autónomos	8
1.3 Clasificación de los vehículos terrestres no tripulados autónomos	10
1.4 Partes que conforman un vehículo terrestre no tripulado autónomo.....	11
1.4.1 Control	12
1.4.2 Comunicación	13

1.4.3	Actuadores.....	14
1.4.4	Sensores.....	16
1.4.5	Sistema de posicionamiento global o navegación	16
1.4.6	Cámara en primera persona	18
1.4.7	Video transmisor y receptor	18
1.4.8	Fuente de alimentación	19
1.5	Requerimientos de calidad para el vehículo terrestre no tripulado autónomo.....	20
CAPÍTULO 2		22
Diseño electrónico del vehículo terrestre no tripulado autónomo.....		22
2.1	Metodología	22
2.1.1	Características técnicas del vehículo terrestre no tripulado autónomo.....	22
2.1.2	Matriz de los requerimientos y características técnicas el vehículo terrestre no tripulado autónomo	23
2.2	Selección de componentes electrónicos.....	25
2.2.1	Controladores	25
2.2.2	Motores o actuadores.....	26
2.2.3	Módulos de comunicación.....	27
2.2.4	Sensores de distancia.....	28
2.2.5	Módulos de sistemas de navegación	28
2.2.6	Cámaras en primera persona.....	29
2.2.7	Transmisor y receptor de video	30
2.2.8	Capacidad de batería	31
2.3	Resultados	32
2.3.1	Diagrama de bloques y descripción funcional.....	32
2.3.2	Diseño electrónico del vehículo terrestre no tripulado autónomo	33
2.3.3	Tipo del vehículo terrestre no tripulado autónomo.....	36
2.3.4	Software.....	38
CAPÍTULO 3		41

Implementación del vehículo terrestre no tripulado autónomo	41
3.1 Metodología	41
3.2 Descripción de costos comerciales de los componentes electrónicos elegidos para el vehículo terrestre no tripulado autónomo	41
3.3 Distribución y montaje de los componentes electrónicos.....	42
3.3.1 Montaje de los actuadores.	43
3.3.2 Montaje de sensores de distancia Sharp	47
3.3.3 Montaje del módulo Xbee S2c en el vehículo y estación remota.....	48
3.3.4 Montaje del módulo de sistema de posicionamiento global y magnetómetro HMC5883L.....	50
3.3.5 Montaje de la cámara en primera persona, el video transmisor TS832 y receptor RC832	51
3.4 Implementación del código de programación	52
3.4.1 Configuración de los Xbee.....	53
3.4.2 Programación del código del vehículo terrestre no tripulado autónomo	54
3.4.2.1 Programación de sensores Sharp.....	54
3.4.2.2 Programación del Magnetómetro.....	55
3.4.2.3 Programación del módulo GPS	56
3.4.2.4 Programación de los filtros o muestras de los datos GPS	57
3.4.2.5 Programación para el cálculo del vector posición.....	58
3.4.2.6 Programación de los actuadores.....	59
3.4.2.7 Programación del control de la base remota	61
3.4.2.8 Configuración de la cámara runcam con el transmisor de video	62
3.5 Ejecución del vehículo terrestre no tripulado autónomo pruebas y ajustes.....	63
3.6 Resultados	64
3.6.1 Funcionalidad del controlador	64
3.6.2 Funcionalidad de los actuadores.....	65
3.6.3 Capacidad de comunicación.....	67

3.6.4	Respuesta del módulo posicionamiento global neo Ublox 6m	67
3.6.5	Respuesta de sensores Sharp.....	68
3.6.6	Duración de batería	69
3.6.7	Calidad de transmisión y recepción de video.....	69
3.7	Discusión	70
	Conclusiones	71
	Recomendaciones.....	72
	Referencias	73
	Anexos	77
	Anexo A: Datos de la placa distribuidora de voltaje PDB.....	77
	Anexo B: Hoja de datos de RC832 Y TS832	78
	Anexo C: Hoja de datos de motor brushless 2300KV	79
	Anexo D: Hoja de datos de modulo GPS NEO 6m	80
	Anexo E: Prototipo de vehículo terrestre no tripulado autónomo	81
	Anexo F: Mediciones de velocidad	82
	Anexo G: Manual de Usuario.....	83
	Anexo H: Manual Técnico.....	94
	Anexo I: Manual de Mantenimiento	106
	Anexo J: Código de programación.....	108

Índice de Imágenes

Fig. 1. Proyecto CITIES Timanfaya bus eléctrico sin conductor	8
Fig. 2. Tractor para trabajar el campo	9
Fig. 3. Rover NASA	9
Fig. 4. Vehículo para transportar oro	9
Fig. 5. Guardium II.....	9
Fig. 6. RESCUE vehículo para transporte de personas heridas	9
Fig. 7. Vehículo 4 ruedas	10
Fig. 8. Vehículo Talón mediante deslizantes usos militares	10
Fig. 9. BigDog cuadrúpedo robótico	10
Fig. 10. PakBot de la empresa iRobot.....	11
Fig. 11. Arduino Mega 2560	12
Fig. 12. Raspberry Pi 3.....	12
Fig. 13. Pixhawk.....	13
Fig. 14. Módulo Bluetooth HC-05	13
Fig. 15. Módulo Wi-Fi ESP8266	14
Fig. 16. Módulo Xbee S2C	14
Fig. 17. Emisor y receptor RF.....	14
Fig. 18. Motor Brushless	15
Fig. 19. ESC Control Electrónico de Velocidad	15
Fig. 20. Servomotor.....	15
Fig. 21. Sensor de distancia Sharp	16
Fig. 22. Módulo ultrasónico	16
Fig. 23. Modulo GPS	17
Fig. 24. Magnetómetro HMC5883L	17
Fig. 25. Cámara CMOS FPV	18
Fig. 26. Transmisor y receptor de video FPV, antena cloverleaf	19
Fig. 27. Diagrama de bloques del vehículo terrestre no tripulado autónomo	33
Fig. 28. Diseño electrónico de un vehículo terrestre no tripulado autónomo.	34
Fig. 29. Diseño referencial del vehículo terrestre no tripulado autónomo	36
Fig. 30. Montaje de los actuadores	43
Fig. 31. Motorreductor de 4 engranes	44
Fig. 32. Motorreductor del servomotor.....	46
Fig. 33. Conexiones de actuadores.....	46
Fig. 34. Montaje de los sensores de distancia Sharp	47
Fig. 35. Conexiones de los sensores de distancia Sharp	48

Fig. 36. Montaje de los Xbee S2c.....	49
Fig. 37. Conexiones de los Xbee S2c.....	49
Fig. 38. Montaje del módulo GPS y magnetómetro HMC5883L	50
Fig. 39. Conexiones del módulo GPS y magnetómetro HMC5883L	50
Fig. 40. Montaje del transmisor TS832 de video	51
Fig. 41. Conexiones del transmisor de video TS832	52
Fig. 42. Configuración Xbee	53
Fig. 43. Diagrama de flujo de los sensores Sharp	54
Fig. 44. Diagrama de flujo del Magnetómetro.....	55
Fig. 45. Diagrama de flujo del módulo GPS	56
Fig. 46. Lecturas de valores GPS con filtro y sin filtro	57
Fig. 47. Diagrama de flujo del vector posición	58
Fig. 48. Diagrama de flujo del ángulo posición	59
Fig. 49. Diagrama de flujo de los actuadores	60
Fig. 50. Diagrama de flujo de la base de remota	61
Fig. 51. Configuración de la recepción de la cámara de video	62
Fig. 52. Ruta del vehículo terrestre no tripulado autónomo.	63
Fig. 53. Error de desviación máxima del vehículo con respecto a las coordenadas establecidas.....	64
Fig. 54. Pruebas en superficies planas, inclinadas, e irregulares.	66
Fig. 55. Lecturas de la trayectoria del vehículo.	67
Fig. 56. Respuesta del módulo GPS.	68
Fig. 57. Vehículo autónomo evitando obstáculo.....	69

Índice de Tablas

TABLA 1. Clasificación de las aplicaciones de los vehículos terrestres no tripulados.....	8
TABLA 2. Tipos de locomoción de vehículos terrestres no tripulados autónomos	10
TABLA 3. Clasificación de plataformas electrónicas	12
TABLA 4. Tipos de módulos de comunicación.....	13
TABLA 5. Tipos de actuadores	15
TABLA 6. Tipos de sensores de distancia	16
TABLA 7. Factores de batería Li-Po	20
TABLA 8. Lista de requerimientos del prototipo de vehículo terrestre no tripulado autónomo	21
TABLA 9. Características técnicas del vehículo terrestre no tripulado autónomo	22
TABLA 10. Matriz de relación entre los requerimientos y características técnicas.....	23
TABLA 11. Grado de importancia basado en la matriz de diseño de la función de calidad (QFD)	24
TABLA 12. Características de diferentes tipos de controladores	25
TABLA 13. Características de actuadores	26
TABLA 14. Características de los módulos de comunicación	27
TABLA 15. Características de sensores de distancia	28
TABLA 16. Especificaciones de módulos GPS	29
TABLA 17. Especificaciones de cámaras FPV.....	29
TABLA 18. Especificaciones de los tipos de transmisores y receptores de video	30
TABLA 19. Consumo de corriente de componentes electrónicos	31
TABLA 20. Característica del vehículo terrestre no tripulado autónomo	36
TABLA 21. Costo de los componentes electrónicos comerciales	42
TABLA 22. Conexiones de pines de los actuadores	47
TABLA 23. Conexiones de los sensores Sharp	48
TABLA 24. Conexiones de Xbee S2c	49
TABLA 25. Conexiones del módulo GPS y magnetómetro	51
TABLA 26. Conexiones de transmisor y receptor de video	52
TABLA 27. Configuración de Xbee coordinador y Router	53
TABLA 28. Resultados de la ejecución del controlador.....	65
TABLA 29. Resultados de la ejecución de actuadores	66
TABLA 30. Resultados de la ejecución de la batería Li-Po	69

Resumen

El desarrollo de nuevas tecnologías autónomas está diseñado para cumplir determinadas aplicaciones entre ellas exploración espacial, espionaje, búsqueda y rescate, además de diferenciarse por la variedad de diseños y así resolver los problemas que aquejan al mundo como desastres naturales que dan lugar a que se extravíen personas y no se logre una búsqueda eficiente.

El objetivo de esta investigación es la implementación de un prototipo de vehículo terrestre no tripulado autónomo utilizando la metodología del diseño de la función de calidad el cual permite la selección de materiales de acuerdo a las necesidades requeridas por el usuario, con el fin de realizar un diseño electrónico que da lugar a la implementación, así realizando pruebas y ajuste de funcionalidad en donde se obtiene que los vehículos autónomos sofisticados son costosos pero también puede ser implementado con tecnologías de menor gama y cumplir con los requerimientos mínimos, ya que entre más componentes electrónicos se incluya mayor recursos se requerirá en el controlador, de la misma forma sucede con la comunicación, transmisión y recepción de información entre más área de cobertura se quiera alcanzar mayor potencia de transmisión es necesaria.

El vehículo terrestre no tripulado autónomo transita por superficies planas, inclinadas y semiplanas en las que por medio de los sensores esquiva obstáculos y se dirige de manera autónoma a una o varias coordenadas declaradas como latitud y longitud.

Palabras clave: Vehículos terrestres no tripulados, navegación autónoma, sensores, telemetría, radio frecuencia.

Abstract

The development of new autonomous technologies is designed to meet certain applications including space exploration, espionage, search, and rescue, in addition to differentiating itself by the variety of designs and thus solving the problems that afflict the world such as natural disasters that lead to misplacement. people and an efficient search are not achieved.

The objective of this research is the implementation of an autonomous unmanned land vehicle prototype using the quality function design methodology which allows the selection of materials according to the needs required by the user, in order to carry out an electronic design that gives rise to the implementation, thus performing tests and adjustment of functionality where it is obtained that sophisticated autonomous vehicles are expensive but can also be implemented with lower-end technologies and meet the minimum requirements, since more electronic components More resources will be required in the controller, in the same way it happens with the communication, transmission and reception of information, the more coverage area you want to reach, the greater the transmission power is necessary.

The autonomous unmanned land vehicle travels on flat, inclined, and semi-flat surfaces in which, by means of sensors, it avoids obstacles and autonomously goes to one or more coordinates declared as latitude and longitude.

Keywords: Unmanned ground vehicles, autonomous navigation, sensors, telemetry, radio frequency.

Contexto

La tecnología autónoma en épocas pasadas se encontraba presente en áreas militares ya que tenía un elevado costo, pero a medida que pasaron los años fueron desarrollando nuevos dispositivos eléctricos y electrónicos, lo que hizo posible poder abordar este tema en el terreno civil, desde entonces se ha tenido avances muy grandes y varias compañías trabajan en vehículos autónomos. (Delgado, 2016)

Existen varias empresas que están dentro de potenciales mercados como son países de Hispanoamérica y Europa pero hasta el momento Estados Unidos es el que lleva la vanguardia en esta tecnología seguida de China. (Contreras, 2018), también otros países que están desarrollando esta tecnología es Francia con un 4,8%, Suecia con un 4,7%, y en menor medida Reino Unido y Países Bajos con un 2,9 y un 2,8 por ciento, respectivamente, también Alemania se encuentra entre los 5 países del mundo en mejor legislación y tecnología autónoma. (Romero, 2018)

Por otra parte, es una tecnología la cual se espera que dentro de 10 a 15 años ya veamos vehículos autónomos en las calles, Antonio Rojas, director de Tecnología y Ecosistemas en Grupo Premo en España, menciona que “Al final, el coche será un robot más en nuestra vida. Para su desarrollo solo hace falta que la tecnología permita mejorar la transmisión de datos, que reduzca los tiempos de latencia y que sea altamente flexible”. (Muñoz, 2018)

Conjuntamente con el avance de vehículos autónomos otra transformación fundamental es la conectividad, la que hace posible que los robots funcionen independientemente de un usuario, por lo que se espera que la conexión 5G incrementará el número de conexiones con más fluidez y rapidez que la conexión 4G. (Aguilera, 2019), Inés Sanz responsable de Hispasat indico que “El 5G mejorará notablemente la red de comunicaciones para que los vehículos puedan ser más inteligentes y más autónomos. Y en este ámbito, los satélites tendrán un papel fundamental donde no se pueda contar con infraestructuras terrestres. El satélite garantiza una cobertura universal (tierra, mar y aire), facilita la movilidad”. (Muñoz, 2018)

Toda esta tecnología da como resultado una infinidad de aplicaciones en las que se puede aplicar en áreas como agricultura, recolección de información flora y fauna, transporte de carga, búsquedas en lugares de difícil acceso, reconocimiento y exploración de terrenos, recolección de basura, en el área militar se usaría para exploración de áreas peligrosas para los soldados, vigilancia, transporte de suministros, fijar objetos, desactivación de bombas entre otros (Alkè, 2016). En áreas militares tiene un gran impacto ya que es el blanco perfecto para los hackers, como es un sistema autónomo pueden crearse formas de hackear el sistema ya que el FBI mencionan que "las tecnologías autónomas pueden ser alteradas y convertirse en un arma letal, del mismo modo que podrían utilizarse estos vehículos para secuestros y otras prácticas ilegales semejantes". (Fernández, 2019)

Una de las compañías que utilizan la tecnología autónoma para uso de fuerzas de defensa, seguridad, y servicios de rescate, es la compañía Milrem Robotics establecida en Europa en el 2013. Esta compañía tiene una variedad de vehículos autónomos los cuales tiene características variadas dependiendo al área que se vaya a utilizar, también trabajan haciendo pruebas en Malí en África Occidental ya que es una de las regiones más duras del mundo con suelos de roca de lava, arena, y altas temperaturas de hasta 50 grados a la sombra. (MilremAS, 2019)

La compañía Milrem Robotics se enfoca en vehículos autónomos para uso militar y sus costos son elevados pero es un gran aporte, el teniente segundo Mandis Parnpuu menciona que "El despliegue de un vehículo no tripulado permitirá a las unidades aumentar su poder de combate, reducir la carga física para los soldados y permite que se incluyan suministros adicionales en una operación como armamento pesado, agua adicional y municiones que no podrían transportarse sin el vehículo". (MilremRobotics, 2019)

Planteamiento del problema

Con el paso del tiempo el planeta ha sido presente ante desastres naturales los cuales perturban la vida cotidiana de los seres humanos, estos eventos no se saben con exactitud cuándo, dónde y de que magnitud van a ocurrir. En Indonesia el 28 de septiembre del 2018, la ciudad de Palu fue partícipe de un terremoto y tsunami que provocó alrededor de 5000 personas extraviadas, Sutopo Purwo Nugroho el portavoz de la agencia de Prevención de Desastres dijo que “se seguirá realizando una búsqueda limitada, pero aquellas a gran escala con mucho personal y equipo pesado se detendrán”. (Berry, 2018)

En el Ecuador también existieron casos eventuales como por ejemplo en Esmeraldas el 16 de abril del 2016 el cantón Muisne presencio un terremoto que provocó alrededor de 40 personas extraviadas. Otro evento similar ocurrió en la provincia de Imbabura el 16 de agosto de 1868, las ciudades como Ibarra, Atuntaqui, Cotacachi, Otavalo y Urcuquí fueron presentes de dos sismos seguidos de gran magnitud que dejo un gran número de fallecidos. (Instituto Geofísico, 2017)

La búsqueda de personas extraviadas ante catástrofes naturales conlleva tiempo, personas y recursos, por lo que la búsqueda no sería eficiente si las personas involucradas fallecen por tener heridas de gravedad, poco aire para respirar producto de gases tóxicos fruto de la combustión durante el fuego, por este motivo se debería optar por buscar nuevos desarrollos tecnológicos que faciliten y agilicen la búsqueda de personas extraviadas, salvaguardando la integridad física de las personas que realicen pesquisas en lugares peligrosos.

Objetivos

Objetivo General

Implementar un prototipo de vehículo terrestre no tripulado autónomo.

Objetivos Específicos

- Realizar un análisis sobre la tecnología de los vehículos terrestres no tripulados autónomos, que proporcionará las bases necesarias para la implementación del prototipo.
- Diseñar un prototipo electrónico de vehículo terrestre no tripulado autónomo.
- Implementar un vehículo terrestre no tripulado autónomo para realizar pruebas y ajustes.

Justificación

Ante los eventos de catástrofes naturales, en los que se ve implicadas personas extraviadas, por no contar con desarrollos de tecnologías para búsqueda, esto ha propiciado un clima de preocupación sobre lo que podría pasar si estos eventos naturales ocurrirían con más frecuencia, (Berry, 2018) por lo que resulta de especial interés conocer el nuevo desarrollo de tecnologías que contribuirán para beneficio de la sociedad.

El presente proyecto surge de la necesidad de agilizar búsquedas de rescate para personas extraviadas frente a eventualidades naturales que el Ecuador ha pasado en estos últimos años, de tal forma que con el estudio y desarrollo de un prototipo de vehículo terrestre no tripulado autónomo (UGV), se logre reducir el tiempo de búsqueda cuando se necesite encontrar a personas extraviada en territorios hostiles o de difícil acceso.

Con esto se logrará motivar a futuras generaciones para que este mismo sistema pueda ser replicado en cualquier actividad como es el caso de la agricultura con la fabricación de tractores que realicen tareas rutinarias de siembra y cosecha, en la pesca en las que podría recorrer rutas sin necesidad de que los marineros tengan que pasar varios días en el mar, por lo que numerosas industrias podría ser beneficiadas por el desarrollo de este prototipo.

Por otra parte, con el desarrollo de este prototipo, resulta un aporte importante del estado del arte de los vehículos terrestres no tripulados autónomos (UGV), ya que este sistema puede ser adaptable a cualquier necesidad de transporte que requiera navegación autónoma.

Alcance

Este trabajo se basa en el diseño y la implementación de un prototipo de vehículo terrestre no tripulado autónomo dirigido con Sistema de Posicionamiento Global (GPS), sensores de evasión de obstáculos, cámara en primera persona, uso de un controlador con software libre, y un sistema de alimentación para los dispositivos electrónicos, este prototipo tendrá la capacidad de moverse por navegación (GPS) en el que se indicará la trayectoria que debe realizar el vehículo de manera que se movilizará sobre superficies planas, el cual tendrá la capacidad de tomar decisiones para evitar obstáculos y llegar a su destino por medio de sensores y a su vez transmitir en tiempo real la visibilidad del entorno por el que transita, en el cual se realizarán las pruebas respectivas y ajustes para un mejor desempeño del prototipo.

Viabilidad

Para el desarrollo del prototipo, se necesitará el uso de un microcontrolador, existe una gran variedad de modelos con software libres, por lo que son componentes que se pueden encontrar en tiendas online, para la adquisición del resto de los materiales como son motores brushless, modulo GPS, cámara en primera persona, cables, etc. se los puede encontrar en tiendas online o tiendas electrónicas cerca de la zona por lo que son materiales de fácil acceso.

Los gastos de todos los componentes para la realización del prototipo serán provenientes de recursos del autor del proyecto, el costo varía dependiendo de la escala y el uso que se requiera, por tanto, el costo no sería elevado, por lo que no se requiere un financiamiento mayor.

CAPÍTULO 1

Marco Teórico

En este capítulo se desarrolla la base teórica del proyecto en el que se presentará las aplicaciones más comunes para los vehículos terrestres no tripulados autónomos, los componentes más usados para la fabricación y los requerimientos que se tendrá en cuenta para la implementación y el mejor desempeño.

1.1 Vehículos terrestres no tripulados autónomos

Los avances en tecnología móvil son principalmente los vehículos no tripulados terrestres autónomos o sus siglas en inglés UGV (Unmanned Ground Vehicle), estos sistemas autónomos son capaces de movilizarse en superficies terrestres sin necesidad de un conductor dentro del vehículo, evitar obstáculos y operar en condiciones donde peligre la integridad física de la persona. (Embation, 2016)

Los sistemas de control para estos vehículos son:

- a) UGV Tele operados o a control remoto.
- b) UGV Autónomo.

1.1.1 Vehículos no tripulados terrestres tele operada o a control remoto

Este tipo de vehículos son operados por una persona que es la encargada de manipular las acciones en función de la observación visual directa o por medio de cámaras de video digital o sensores, a través de una emisora de radiocontrol. (Morales, 2017)

1.1.2 Vehículos no tripulados terrestres autónomo

Un UGV autónomo es aquel que opera sin necesidad de un controlador humano, es decir realiza tareas predefinidas mediante el uso de sensores para tener un entorno limitado y esta toma acciones necesarias para cumplir con la tarea asignada en un inicio por el ser humano, eliminando así la necesidad de que una persona tenga que estar vigilando la tarea asignada al UGV. (Ortega, Darío, & Cárdenas, 2018)

Un UGV puede tener la capacidad de:

- a) Recopilar información del medio ambiente.
- b) Detectar objetos de interés como personas y vehículos.
- c) Realizar trabajos prolongados sin que intervenga el humano.
- d) Evitar a las personas situaciones que sean perjudiciales para sí mismos.
- e) Desarmar, o eliminar explosivos.
- f) Repararse sin asistencia externa.

1.2 Aplicaciones y usos de los vehículos terrestres no tripulados autónomos

Estos vehículos tienen un gran uso hoy en día en sectores civiles y militares, también en industrias como la agricultura, la minería, y la construcción como se muestra en la Tabla 1, esto con el fin de remplazar al humano en situaciones peligrosas (Embation, 2016).

TABLA 1. Clasificación de las aplicaciones de los vehículos terrestres no tripulados

Aplicaciones	Usos	Vehículo terrestre no tripulado	Referencia
Civiles y comerciales	En esta área la aparición de un autobús eléctrico de 12 plazas sin conductor ha causado que muchas personas visiten el parque tecnológico de Leganés en España ya que el vehículo realiza un recorrido como guía turística con una trayectoria de ida y vuelta una velocidad de 24 km/h controlado por cámaras y sensores. Este proyecto fue promovido desde 2018 por los centros de Arte, Cultura y Turismo del Cabildo de Lanzarote con la participación de la Universidad Carlos III.		(Cortes, 2019)

Fig. 1. Proyecto CITIES
Timanfaya bus eléctrico sin conductor

Agricultura	<p>Los tractores de recolección no tripulados pueden realizar trabajos en el campo reduciendo el tiempo recolección, controlar la salud de los cultivos y el ganado entre otras uno de estos vehículos fue presentado por Case IH en Silicon Valley y Detroit este vehículo cuenta con cámaras, GPS y sistema de radar el cual permite manejar mejor el entorno. La persona programa en su Tablet el trabajo a realizarse y puede trabajar simultáneamente con otros tractores.</p>		(Cruz, 2016)
Espaciales	<p>Una de las hazañas de exploración espacial más exitosa fue la misión del Rover de la NASA esto se atribuye a los sistemas redundantes, el manejo cuidadoso y la toma de decisiones de interfaz a largo plazo.</p>		(Lopez, 2019)
Minería	<p>Estos vehículos también pueden ser adaptados para atravesar y mapear túneles de minas. Combinando sensores de radar, laser y visuales se están desarrollando vehículos para mapear superficies de rocas en 3D en minas y cielos abiertos. Las tecnologías más usuales son radares, sistemas LiDar de posicionamiento por láser y la navegación GPS.</p>		(Pedraza, 2019)
Respuesta de emergencia	<p>Los UGV también son usados para situaciones de emergencia como es la búsqueda y rescate urbano, la lucha contra incendios y la respuesta nuclear. Un vehículo llamado Guardium II se fabricó para ser operado a distancia por un controlador a través de una interfaz, la cual se dirigen por medio de cámaras de video digitales y la ayuda de sensores.</p>		(Today, 2017)
Militar	<p>En el ámbito militar los sistemas UGV son utilizados para reconocer áreas peligrosas, transportar provisiones y equipos a las fuerzas militares, fijar objetivos, desactivar artefactos explosivos, una de las industrias que se dedica a la fabricación de vehículos terrestres no tripulados autónomos es Milrem Robotics.</p>		(MilremRobotics, 2019)

1.3 Clasificación de los vehículos terrestres no tripulados autónomos

En la actualidad existen varios modelos de vehículos terrestres no tripulados, cada uno de estos son diseñados para cruzar algún tipo de terreno para el cual se vaya a situar el vehículo, la velocidad y nivel de obstáculos que se encuentren para esto existen diferentes tipos de locomoción para vehículos terrestres no tripulados, mediante ruedas, deslizantes, extremidades o patas, e híbridos como se muestra en la Tabla 2. (Chiriboga, Chafla, & Carlos, 2019)

TABLA 2. Tipos de locomoción de vehículos terrestres no tripulados autónomos

Tipos de locomoción	Características	Modelo	Referencias
Ruedas	Este modelo es similar a los automóviles convencionales ya que consta con cuatro ruedas, también existe vehículos no tripulados de tres ruedas, la delantera sirve para mover la tracción y dirección, el eje trasero tiene acopladas dos ruedas paralelas que se mueven libremente pero este modelo tiene problemas de estabilidad ya que no tiene muchos puntos de apoyo.		(Garcia & Garcia, 2016)
Deslizantes	Estos vehículos son similares a los vehículos blindados de los militares son tipo oruga en los que la tracción y el direccionamiento se consigue mediante bandas de tracción o pistas de deslizamiento este sistema tiene desventajas como una reducción de velocidad comparado al uso de ruedas y consume más energía ya que en la rotación existe rozamiento entre el suelo.		(Chiriboga, Chafla, & Carlos, 2019)
Extremidades	Este tipo de modelo se los utiliza para terrenos en donde sea difícil evitar obstáculos para este tipo hay cuadrúpedos que son de 4 patas y hexápodos que son de 6 patas la gran desventaja es que requiere gran consumo de energía además que el control es más complejo que los de ruedas o deslizantes.		(Chiriboga, Chafla, & Carlos, 2019)

Híbrida

Este tipo de sistemas de locomoción son configuraciones que se basan en la unión de dos modelos de locomoción que lo único que buscan es satisfacer las necesidades de los consumidores en trabajos forzados o cruzar terrenos irregulares.



(Garcia & Garcia, 2016)

Fig. 10. PakBot de la empresa iRobot

1.4 Partes que conforman un vehículo terrestre no tripulado autónomo

Para los vehículos terrestres se tiene elementos mecánicos, electrónicos y de control que trabajan todos en conjunto para realizar su funcionamiento de acuerdo con las necesidades que se lo requiera estos son: control, actuadores, sensores, comunicación, estructura y chasis.

- Control: Este sistema es el encargado de realizar los algoritmos para ejecutar las acciones que están programadas por el usuario el cual utilizará los sensores como entradas y los actuadores como salidas, por lo general se utilizan microcontroladores que son programables esto definirá el comportamiento del vehículo. (Mera & Ruano, 2019)
- Comunicación: Los sistemas de comunicación inalámbrica mejoran la estética de los vehículos para la transmisión de datos, audio y video sin la necesidad de cables, por medio de la comunicación de un emisor y receptor. Las señales de comunicación son vía radiofrecuencia o infrarrojo. (Mera & Ruano, 2019)
- Estructura y chasis: Es el armazón del vehículo en donde albergará todos los componentes electrónicos, el material utilizado depende de la aplicación para la que se vaya a utilizar ya que este brindará soporte y estabilidad a los materiales. (Garcia & Garcia, 2016)
- Actuador: Es el elemento de la señal de salida el cual permite la interacción del vehículo con el medio que lo rodea, ya que este realizará desplazamientos acordes a

las señales de entrada de sensores. los elementos más comunes son los motores eléctricos. (Garcia & Garcia, 2016)

- Sensores: Son los elementos que proporcionan información al sistema de control para que realice las acciones necesarias para ejecutar una orden que pueden ser de tipo digitales y analógicas; por ejemplo, encoders, ultrasónicos, laser LiDar, cámaras, etc. (Mera & Ruano, 2019)

1.4.1 Control

Hoy en día existe una gran variedad de dispositivos electrónicos programables que cuentan con variedad de microcontroladores, pero se debe tener en cuenta que cada tarjeta sirve para su propósito. Entre las plataformas electrónicas más comunes están Arduino, Raspberry Pi, Pixhawk, Pic las que se detallan en la Tabla 3. (Torres, 2017)

TABLA 3. Clasificación de plataformas electrónicas

Plataformas electrónicas	Detalles	Modelos	Referencias
Arduino	Arduino es una plataforma electrónica basada en hardware y software libre es decir de acceso público, puede ser programado en Windows como macOS y GNU/Linux. El Arduino está basado en un microcontrolador ATMEL que puede grabar instrucciones que permiten realizar programas para que interactúen con los circuitos de la placa.	 <p>Fig. 11. Arduino Mega 2560</p>	(Yúbal, 2018)
Raspberry Pi	La Raspberry cuenta con interfaces gráficas, comunicación mediante Wi-Fi, capacidad de manipular la cámara web mediante USB, cuenta con muchos programas y lenguajes de programación como es C++ y Python y cuenta con bibliotecas de programación libres para poder trabajar con módulos hardware a través de múltiples interfaces como SPI, I2C, USART, i2S, diversos puertos USB, HDMI, CSI para cámaras, DSI para pantallas,	 <p>Fig. 12. Raspberry Pi 3</p>	(Rodriguez, 2017)

Ethernet, Jack de 3,5mm para audio y vídeo compuesto.

Pixhawk

Es un módulo automático de alto rendimiento con hardware abierto que se lo utiliza principalmente para cualquier tipo de plataformas robóticas móviles como helicópteros, drones, coches, barcos, submarinos. Las principales aplicaciones con este dispositivo son para navegación, localización, investigación, educación, control y sistemas estabilizados.



Fig. 13. Pixhawk

(Samaniego, 2018)

Para la fabricación de cualquier invento mediante el uso de controladores se debe tener en cuenta aspectos como la velocidad de procesamiento, precio, rendimiento, facilidad de programación, todos estos aspectos se deben tener en cuenta para elegir un microcontrolador. (Torres, 2017)

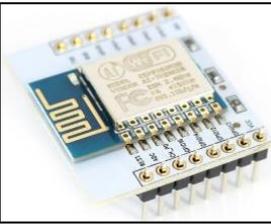
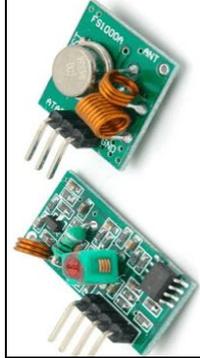
1.4.2 Comunicación

Para la comunicación entre una base de control y el vehículo existen diferentes dispositivos como Bluetooth, Wi-Fi, Zigbee, Radio Frecuencia RF mostrados en la Tabla 4.

TABLA 4. Tipos de módulos de comunicación

Tecnología	Detalles	Módulo	Referencias
Bluetooth	La tecnología Bluetooth permite establecer enlaces entre un microcontrolador y una PC o un teléfono móvil. La creación de los módulos Bluetooth facilita establecer un enlace serial con una PC o cualquier dispositivo con Bluetooth. La salida del módulo es una señal serial asíncrona a 9600 baudios y un alcance máximo de 10 metros.		(Mera & Ruano, 2019)

Fig. 14. Módulo Bluetooth HC-05

<p>Wi-Fi</p>	<p>La tecnología Wi-Fi innova frente a la tecnología Bluetooth ya que se puede mover archivos de gran tamaño y la transmisión de los datos la realiza 10 veces más rápido y tiene un alcance de 100 metros. Este medio servirá para permitir a cualquier robot comunicarse con otro procesador y su alcance máximo de comunicación es de 100 a 450 en línea de vista.</p>	 <p>Fig. 15. Módulo Wi-Fi ESP8266</p>	<p>(Germán, 2017)</p>
<p>Zigbee</p>	<p>Es un conjunto de protocolos de alto nivel de comunicación inalámbrica esta se basa en el estándar 802.15.4 definido por el IEEE, este dispositivo se ha desarrollado para implementarlo en un sistema de control el cual cumple con alta fiabilidad, bajo coste, muy bajo consumo, altamente seguro y con un alcance máximo con antenas normales a 1200 metros con línea de vista.</p>	 <p>Fig. 16. Módulo Xbee S2C</p>	<p>(Sandria, 2018)</p>
<p>Radio Frecuencia</p>	<p>Los módulos de radio frecuencia RF son transmisores y receptores inalámbricos los cuales se emplea para generar una comunicación con un procesador o estación remota, estas frecuencias varían dependiendo del módulo, son frecuencias con bandas libres lo que permite gratuidad para poder usar esos tipos de frecuencia con una línea de vista de 300 metros.</p>	 <p>Fig. 17. Emisor y receptor RF</p>	<p>(Llamas, 2016)</p>

1.4.3 Actuadores

Son dispositivos que proporciona fuerza para mover o actuar otro dispositivo, la fuerza que provoca el actuador proviene de presión neumática, presión hidráulica y fuerza motriz. Un ejemplo son los motores eléctricos como se detalla en la Tabla 5.

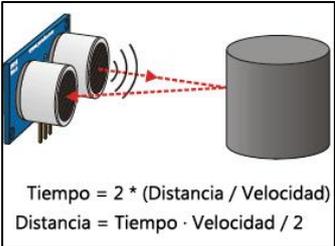
TABLA 5. Tipos de actuadores

Actuadores	Detalles	Dispositivos	Referencias
Motor eléctrico Brushless	<p>Estos motores funcionan sin escobillas a diferencia de los motores con escobillas que generan un roce al girar lo que produce que el rendimiento no sea favorable. Un motor brushless está compuesto por una parte móvil que hace girar el rotor y la carcasa o parte fija.</p>		(Barroso, 2019)
Control electrónico de velocidad	<p>Es un controlador que regula la velocidad y giro de motores mediante la generación de pulsos que son compatibles con motores brushless. Los variadores están en función de la corriente máxima que pueda soportar por lo que se debe conocer la corriente máxima del motor y es recomendable optar por uno que esté por encima de la demanda del motor.</p>		(FpvMax, 2017)
Servomotor	<p>Es un motor especial que permite controlar el eje para moverse en determinados grados para luego mantenerse fija en una posición, están compuesto por un sistema de componentes electromecánicos y electrónicos, existen dos tipos de servomotores: de rango de giro limitado y de rotación continua.</p>		(González, 2016)

1.4.4 Sensores

Los sensores de distancia son aquellos que miden el rango de distancia que hay entre un objeto y el sensor. Estos sensores se los utiliza para detectar obstáculos, crear trayectorias, resolver laberintos, etc. Entre estos existen algunos tipos como son los sensores fotoeléctricos, sensores ultrasónicos como los que se muestran en la Tabla 6.

TABLA 6. Tipos de sensores de distancia

Sensores	Detalles	Dispositivos	Referencias
Fotoeléctricos	Los sensores Sharp se basan en el rebote de un haz infrarrojo, este rebota en el objeto y es recibido por un array de fototransistor infrarrojo. Para estos sensores hay variedad de distancias que manejan como son de 4-30, 10-80, 20-150 metros.	 <p>Fig. 21. Sensor de distancia Sharp</p>	(Mera & Ruano, 2019)
Ultrasónicos	Son sensores que incorporan un par de transductores de ultrasonido el cual emite una onda de ultrasonido la cual rebota con el objeto y es recibida por el receptor el tiempo que tarda la onda en ir y regresar puede ocuparse para conocer la distancia que hay entre el objeto y el módulo ultrasónico, la distancia que maneja es de 2-400 metros.	 <p>Fig. 22. Módulo ultrasónico</p>	(Proserquisa, 2016), (Mera & Ruano, 2019)

1.4.5 Sistema de posicionamiento global o navegación

El Sistema de Posicionamiento Global (GPS) es un sistema de posicionamiento terrestre, la posición la calcula los receptores GPS por medio de satélites que están en órbita alrededor de la tierra, la cual tiene una red de 24 satélites que son propiedad de los Estados Unidos de América. Existen receptores para uso civil y que su precisión está en una oscilación de 15 a 100 metros RMS y esto ocasiona un error en la transmisión de la posición para los receptores de uso civil, con el fin de que los militares tengan una ventaja estratégica durante operaciones que lo requieran. (Enriquez, 2017)

Una forma de poder utilizar esta tecnología es por medio del módulo GPS que se muestra en la Figura 23, este elemento está diseñado para tener un pequeño tamaño, bajo coste, y bajo consumo. La intensidad de corriente necesaria es de unos 37mA en modo de medición continuo. (Llamas, 2016)

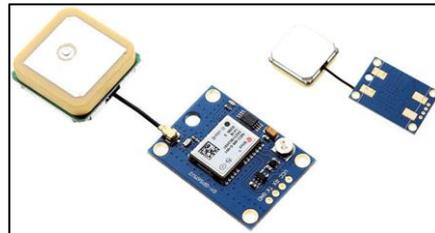


Fig. 23. Módulo GPS
Fuente. Tomado de (Llamas, 2016)

Este tipo de módulos presenta una alimentación de 2.7 a 3.6 V para los modelos NEO-6Q/6M y 1.75 a 2 V para los NEO-6G, también las frecuencias máximas de medición son de 5Hz que son valores aceptables para un sistema de posicionamiento global (GPS).

Otro dispositivo que trabaja juntamente con los módulos GPS son los sensores magnetómetro, estos módulos son brújulas digitales la cual mide el campo magnético de la tierra para poder calcular la orientación con respecto al norte magnético de la tierra. Un módulo magnetómetro es el GY-273 HMC5883L el que consta de 3 ejes x, y, z, como se muestra en la Figura 24. (García, 2017)

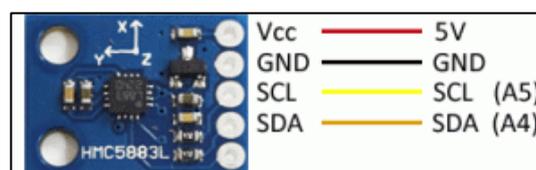


Fig. 24. Magnetómetro HMC5883L
Fuente. Tomado de (García, 2017)

1.4.6 Cámara en primera persona

Este tipo de cámaras son usuales en drones para maniobrar o poder observar en el instante de tiempo lo que el Dron o robot este observando, hoy en día la variedad de estas cámaras avanza y existen cámaras sumamente pequeñas con buenas prestaciones o similares a las cámaras normales. Estas cámaras tienen dos tipos de sensores CCD Y CMOS cada una de estas tiene sus características y ventajas únicas, las cámaras HD utilizan sensores CMOS como la que se muestra en la Figura 25. (Rodríguez, 2017)



Fig. 25. Cámara CMOS FPV
Fuente. Tomado de (Rodríguez, 2017)

Los sensores CCD presentan características como: mejor rango dinámico, en condiciones de poca luz presenta poco ruido, baja latencia, mayor manejo de luz a diferencia de los sensores CMOS presentan características como: mejor resolución, el color de la imagen es más real, presenta mayor ruido, son más económicas fabricarlas. (Rodríguez, 2017)

1.4.7 Video transmisor y receptor

Para transmitir y recibir una señal de video se debe tener en cuenta las frecuencias en las cuales se vaya a transmitir ya que algunas frecuencias son libres y legales otras no son permitidas en algunos países si no se tiene el respectivo permiso. Las frecuencias más usuales son: 900 MHz, 1.2 GHz, 1.3 GHz, 2.4 GHz y 5.8 GHz. (Dario, 2019)

Entre estas frecuencias la que más se utiliza es la 5.8 GHz con tiene 32 canales, las demás frecuencias tienen distintos tipos de canales, esto permite que al momento de

transmitir o recibir una señal el usuario elija algún canal para no interferir con otro usuario que se encuentre usando el mismo canal. Para transmitir una señal es importante el tipo de antena ya que esta proporcionará el alcance en el que el receptor pueda captar dicha señal, entre ellas están las omnidireccionales, cloverleaf, helicoidales mostradas en la Figura 26 las que se diferencian por el material utilizado al igual que su estructura y diseño. (Dario, 2019)

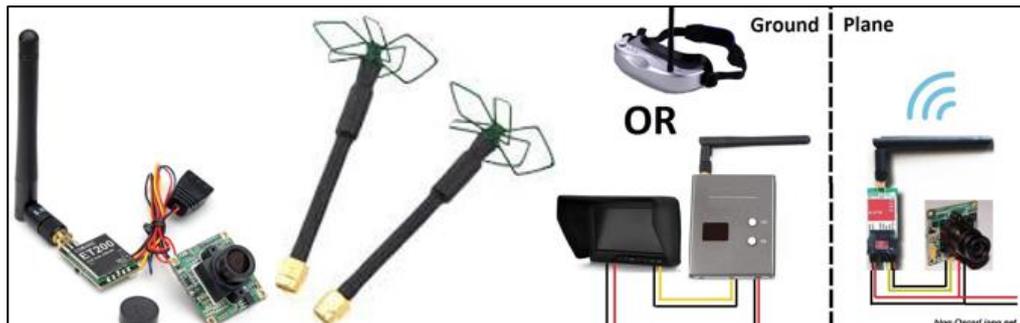


Fig. 26. Transmisor y receptor de video FPV, antena cloverleaf
Fuente. Tomada de (Macho, 2019)

- **Omnidireccionales:** son aquellas que radian energía en todas las direcciones con el mismo alcance.
- **Cloverleaf:** es similar a la omnidireccional solo que la punta tiene forma de trébol con polarización circular, es la más usada para drones con sistemas FPV.
- **Helicoidales:** estas antenas tienen un gran alcance, pero solo radian energía en una sola dirección lo que causa que la recepción de la imagen no sea la deseada.

1.4.8 Fuente de alimentación

Las baterías Li-Po cuya abreviatura significa Litio polímero son baterías usadas para la alimentación de robots móviles ya que son recargables, su estructura es pequeña, usada para aplicaciones que demanden un gran consumo de corriente, livianas y otros factores que se pueden mencionar son los que se detallan a continuación en la Tabla 7. (Mera & Ruano, 2019)

TABLA 7. Factores de batería Li-Po

Numero de celdas “S”	La sigla “S” muestra el número de celdas que la batería posee conectadas en serie. Las baterías Li-Po se componen de celdas de 3,7V las cuales al estar totalmente cargadas llegan a los 4,2V.	3.7V = 1 celda (1S) 7.4V= 2 celdas x 3.7v (2S) 11.1V= 3 celdas x 3.7v (3S) 14.8V= 4 celdas x 3.7v (4S) 18.5V= 5 celdas x 3.7v (5S) 22.2V= 6 celdas x 3.7v (6S)
Capacidad “mAh”	La capacidad es cuanta corriente suministra una batería la que puede ser empleada durante una hora antes de descargarse totalmente y se mide en (mAh).	Un ejemplo es con una batería Li-Po de 1000 mAh se descargaría completamente en una hora con una demanda de 1000mA. Si tiene una demanda de 500 mA en 2 horas se descargaría totalmente.
Tasa de descarga “C”	La tasa de descarga es la rapidez con que una batería se descarga de forma segura. Entre más rápido fluyen los iones del ánodo al cátodo indicará la velocidad de la descarga.	Una batería con una descarga de 10 C se descarga a una velocidad de 10 veces la capacidad de la batería, y así sucesivamente.

Fuente. Tomado de (Enriquez, 2017), (Mera & Ruano, 2019)

La variedad de baterías que existen en el mercado detalla el uso y la forma de carga para alargar la vida útil de la batería la cual dependerá de la marca del fabricante, pero un estimado puede ser de 3 años a unas 500 cargas completas. (Mera & Ruano, 2019)

1.5 Requerimientos de calidad para el vehículo terrestre no tripulado autónomo.

Los vehículos terrestres no tripulados son de diversas formas, tamaños y precios esto ya dependerá del uso que se vaya a dar para esto se debe tener en cuenta el propósito que el vehículo debe cumplir a continuación se presenta un listado de los requerimientos más comunes evaluados en una escala del 1 al 5 donde el valor 1 será el menos importante y el valor 5 con más importancia. (Chiriboga, Chafra, & Carlos, 2019)

TABLA 8. Lista de requerimientos del prototipo de vehículo terrestre no tripulado autónomo

N°	Requerimientos	Prioridad
1	Bajo costo	4
2	Uso para todo terreno	3
3	Componentes de fácil adquisición	4
4	Velocidad de movimiento adecuada	3
5	Fácil operación	3
6	Fácil mantenimiento	3
7	Diseño resistente	4
8	Capacidad de detectar obstáculos	5
9	Adecuada duración de la batería	5
10	Adecuada transmisión y recepción de video	5
11	Adecuada operación remota por GPS	5
12	Fácil reparación	3
13	Robusto	3
14	Liviano	3
15	Movilidad	5
16	Control	5

Fuente. Tomado de (Chiriboga, Chafla, & Carlos, 2019)

Los requerimientos establecidos en la Tabla 8 son los más principales para poder seleccionar un vehículo terrestre no tripulado ya que de esto dependerá su calidad, robustez, complejidad y costo, cabe recalcar que esta valoración es acorde a criterios propios.

CAPÍTULO 2

Diseño electrónico del vehículo terrestre no tripulado autónomo

En este capítulo se realiza un análisis de los diferentes elementos electrónicos que conforman un vehículo terrestre no tripulado autónomo, la metodología que se va a utilizar para la selección de los materiales que cumplan las mejores prestaciones para los requerimientos del vehículo terrestre no tripulado autónomo con la finalidad de cumplir con los objetivos propuestos.

2.1 Metodología

Para la realización del vehículo terrestre no tripulado autónomo se realiza el método del diseño de la función de calidad (QFD), esta metodología permite comparar la calidad de productos para que se adapten a las necesidades del usuario por medio de ponderaciones que da como prioridad enfocar los aspectos más importantes e inicio de partida para el desarrollo del prototipo. Como resultado se puede obtener el diseño y construcción del vehículo para el uso que lo requiera (Yepez, 2016).

2.1.1 Características técnicas del vehículo terrestre no tripulado autónomo

Se detalla en la Tabla 9 las especificaciones técnicas del vehículo terrestre no tripulado autónomo, para satisfacer a los objetivos planteados en el proyecto para garantizar un producto de calidad.

TABLA 9. Características técnicas del vehículo terrestre no tripulado autónomo

N°	Características técnicas
1	Dimensionamiento correcto de motores
2	Uso de software accesibles al usuario
3	Buena comunicación entre controladores
4	Buen diseño del vehículo
5	Materiales acorde a las necesidades
6	Diseño eficiente del circuito

Fuente. Tomado de (Chiriboga, Chafra, & Carlos, 2019)

Las características técnicas antes mencionadas son referentes a lo que debe cumplir el vehículo para su mejor desempeño, el orden en que se colocan no indica el grado de importancia.

2.1.2 Matriz de los requerimientos y características técnicas el vehículo terrestre no tripulado autónomo

Esta matriz analiza los aspectos técnicos juntamente con los requerimientos y por medio de esto ordenar los aspectos primordiales hasta los de menor valor para ello se evalúa en cuatro parámetros que son los siguientes:

- Relación fuerte: con un valor de 9.
- Relación media: con un valor de 3.
- Relación débil: con un valor de 1.
- Sin relación: con un valor de 0.

TABLA 10. Matriz de relación entre los requerimientos y características técnicas

Características técnicas Requerimientos	Prioridad	Dimensionamiento correcto de	Uso de software accesibles al	Buena comunicación	Buen diseño del vehículo	Diseño eficiente del circuito	Materiales acorde a las
Bajo costo	4	9	9	9	9	9	9
Uso para todo terreno	3	3	0	3	3	3	3
Componentes de fácil adquisición	4	3	1	3	3	0	3
Velocidad de movimiento adecuada	3	3	1	1	3	3	1
Fácil operación	3	1	3	0	0	3	0
Fácil mantenimiento	3	3	0	0	3	0	3
Diseño resistente	4	3	0	0	3	0	9
Capacidad de detectar obstáculos	5	0	0	0	0	0	3
Adecuada duración de la batería	5	9	1	1	1	1	1
Adecuada transmisión y recepción de video	5	0	1	3	0	3	3
Adecuada operación remota por GPS	5	0	3	3	0	3	3
Fácil reparación	3	3	0	3	3	0	0
Robusto	3	3	0	1	1	0	3
Liviano	3	3	0	1	1	0	3
Movilidad	5	3	0	3	1	3	3
Control	5	3	0	3	0	3	3
Evaluación total		192	77	140	112	128	203
Grado de importancia		2	6	3	5	4	1

Fuente. Tomado de (Chiriboga, Chafra, & Carlos, 2019)

Una vez relacionado las características técnicas y requerimientos se obtiene una valoración total, para organizar en orden cronológico el grado de importancia, estas valoraciones se las coloca con el fin de cumplir los requerimientos mínimos del prototipo ya que el vehículo debe tener la capacidad de dirigirse a una ubicación con ayuda de un módulo de navegación (GPS), por medio de coordenadas de latitud y longitud, de manera que se movilizará sobre superficies planas, el cual tendrá la capacidad de actuar autónomamente para evitar obstáculos por medio de sensores de distancia, así llegando a su destino y a su vez transmitir en tiempo real la visibilidad del entorno por medio de una cámara, para esto se ordena cronológicamente el grado de importancia de las características técnicas detalladas en la Tabla 11.

TABLA 11. Grado de importancia basado en la matriz de diseño de la función de calidad (QFD)

N°	Características técnicas	Porcentajes de prioridad %
1	Materiales acorde a las necesidades	23.83
2	Dimensionamiento correcto de motores	22.54
3	Buena comunicación entre controladores	16.43
4	Diseño eficiente del circuito	15.02
5	Buen diseño del vehículo	13.15
6	Uso de software accesibles al usuario	9.04

Fuente. Autoría propia

Una vez ordenado las prioridades de mayor a menor, se tiene como primera característica la selección de materiales acorde a las necesidades, juntamente con la segunda característica que es el dimensionamiento correcto de motores con el fin de que el vehículo pueda transitar por superficies planas, por lo que se debe seleccionar motores que cumplan con un buen torque y velocidad ya que sin estas características dificultará la movilidad del vehículo.

La tercera característica es la buena comunicación entre controladores que esto dependerá el vehículo ejecute las acciones de manera satisfactoria, como es la de esquivar obstáculos por medio de sensores, la navegación (GPS), y la transmisión de video.

Las siguientes características técnicas es el diseño en cuanto a robustez, confiable, reducido, etc. Por lo que una buena distribución de los equipos electrónicos permitirá un

buen desempeño del vehículo y por último el uso del software al momento de programar el vehículo.

2.2 Selección de componentes electrónicos

Se realiza una comparación entre materiales para seleccionar el componente que esté más acorde a los requerimientos para la implementación del vehículo con base en los resultados de la Tabla 11, por lo que se tendrá en cuenta aspectos como bajo costo, fácil adquisición del producto, torque, velocidad, comunicación, diseño, software libre entre otros aspectos.

2.2.1 Controladores

A continuación, se realiza un análisis entre algunas familias de dispositivos electrónicos programables teniendo en cuenta aspectos como la velocidad de procesamiento, precio, rendimiento, facilidad de programación, para lo cual se detalla en la Tabla 12 algunas características de estos dispositivos electrónicos.

TABLA 12. Características de diferentes tipos de controladores

Tecnologías	Clases	Voltaje (v)	Corriente (mA)	Voltaje de operación (v)	SRAM (kb)	Clock Speed (MHz)	Peso (g)	Precio \$ (25/04/20)
Arduino	Due	7 - 12	93-100	3.3	96	84	36	25
	Mega	7 - 12	93	3.3 – 5	8	16	37	16
	Uno	7 - 12	60	3.3 – 5	2	16	25	10
	Nano	7 - 12	19	3.3 – 5	2	16	7	8
Tecnologías	Clases	Voltaje (v)	Corriente (mA)	RAM (GB)	Clock Speed (GHz)	Peso (g)	Precio \$	
Raspberry	Pi 1 B+	5	700	0.512	0.7	45	43	
	Pi 2 B	5	820	1	0.9	45	60	
	Pi 3 B	5	1400	1	1.2	45	80	
	Pi 4 B+	5	3000	4	1.5	45	109	
Pixhawk	Cube	4.7 – 6	250	256 kb	0.168	23	238	
	Tres Pro	4.7 – 6	280	384 kb	0.168	63	260	
	Pix. 4	4.7 - 6	280	512 kb	0.216	15.8	290	

Fuente. Tomado de (Roda, 2020), (Llamas, 2017), (Meier, 2020), (Mercado Libre, 2020)

Con las características técnicas de la Tabla 12 se observó que estos controladores funcionan con voltajes entre 4.7 a 12 voltios, las tecnologías como la Raspberry y Pixhawk tienen un mayor consumo de corriente a comparación de Arduino, la velocidad de procesamiento la Raspberry tiene gran capacidad seguida de Pixhawk y Arduino por tanto se optó por el dispositivo electrónico programable de la familia de Arduino Due ya que cumple con los requerimientos de la Tabla 10 y el costo en comparación a los otros dispositivos electrónicos es bajo, de fácil adquisición, tiene un peso bajo, la interfaz de programación no es muy compleja con software libre, la velocidad de procesar información no es tan alta como la de los otros dispositivos pero es la necesaria para poder desarrollar el vehículo terrestre no tripulado autónomo.

2.2.2 Motores o actuadores

A continuación, se realiza un análisis entre algunos actuadores o motores teniendo en cuenta aspectos como la potencia, torque, corriente, para lo cual se detalla en la Tabla 13 algunas características de estos dispositivos electrónicos.

TABLA 13. Características de actuadores

Actuadores	Constante de revoluciones por minuto (KV)	Torque (kg/cm)	Voltaje (v)	Corriente máxima del control de velocidad del motor (A)	Precio \$ (25/04/20)
Motor eléctrico Brushless	1120	Torque alto	5 - 12	60	65
	2300	Torque normal	5 - 12		
	4500	Torque bajo	5 - 12		
Control electrónico de velocidad	---	---	5 - 12	60	75
Servomotor	---	12	4 - 7.2	0.07	11
		2.1	4.8 -	0.03	6
			6.6		

Fuente. Tomado de (Amazon, 2020) , (Mercado Libre, 2020)

Con la información detallada en la Tabla 13 para el motor eléctrico brushless se tuvo en cuenta que entre mayor sea su constante de revolución por minuto tendrá más velocidad, pero menos torque y viceversa, de acuerdo a la Tabla 10 se considera que el vehículo debe tener bajo costo, velocidad y torque adecuados para poder movilizarse por superficies

planas por tal motivo se optó por un motor brushless de 2300 KV (constante de revoluciones por minuto) con su control de velocidad de 60 Amperios que cuenta con un ventilador para mantener en una temperatura adecuada debido a los picos de corriente generados por el motor. Para la selección del servo se opta por el que tiene mayor torque con el fin de que la dirección del vehículo gire correctamente.

2.2.3 Módulos de comunicación

Se realiza un análisis entre algunos módulos de comunicación teniendo en cuenta aspectos como el alcance de línea de vista, corriente, precio, frecuencias, para lo cual se detalla en la Tabla 14 algunas características de estos módulos electrónicos.

TABLA 14. Características de los módulos de comunicación

Tecnologías	Módulos	Frecuencias (GHz)	Alcance línea de vista (m)	Ancho de banda (Mbps)	Corriente (mA)	Precio \$ (25/04/20)
Bluetooth	HC-05	2.4	10	1	50	6
	HC-06	2.4	10	1	40	7
Wi-Fi	ESP-8266 ESP01	2.4	100-450	54	80	11
Zigbee	Xbee S2C	2.4	1200	1	215	42
Radio Frecuencia	3DR	0.433	300	0.25	35	14
	Radio V2	0.915				

Fuente. Tomado de (Mera & Ruano, 2019), (Llamas, 2016), (Mercado Libre, 2020)

Después del analizar las características de estas tecnologías de comunicación de la Tabla 14 se procede a elegir un módulo el cual cumpla con los criterios de la Tabla 11 el cual permita una correcta comunicación entre el controlador el módulo por lo que se eligió la tecnología Zigbee con su módulo Xbee S2C ya que trabaja a altas frecuencias y presenta un gran alcance de línea de vista directa, lo que permite una transmisión y recepción de datos a distancias más grandes que los otros módulos, el consumo de corriente es un poco mayor a los otros, pero no excesiva y cumple con la adecuada operación remota mencionada en la Tabla 10.

2.2.4 Sensores de distancia

A continuación, se realiza un análisis entre dos tipos de sensores de distancia teniendo en cuenta la distancia adecuada para que el vehículo detecte, esquive obstáculos y el consumo de corriente, para lo cual se detalla en la Tabla 15 algunas especificaciones de estos tipos de sensores.

TABLA 15. Características de sensores de distancia

Tipos de Sensores	Clases	Voltaje (v)	Corriente (mA)	Distancia (cm)	Precio \$ (25/04/20)
Fotoeléctricos Sharp	GP2Y0A41SK0F	4.5 - 5.5	12	4 - 30	9
	GP2Y0A21YKF0	4.5 - 5.5	12	10 - 80	13
	GP2Y0A02YK0F	4.5 - 5.5	12	20 - 150	17
Ultrasónicos	Hc-sr04	5	15	2 - 400	3

Fuente. Tomada de (Amazon, 2020) , (Mercado Libre, 2020), (Leibson, 2018)

Una vez realizado el análisis de la Tabla 15 el sensor fotoeléctrico de distancia Sharp GP2Y0A41SK0F cumple con las características de la Tabla 11 que indica una buena comunicación con el controlador por tal motivo el Sharp GP2Y0A41SK0F presenta mejores prestaciones ya que la velocidad de respuesta es más rápida que la del ultrasónico, el consumo de corriente es bajo y trabaja a una distancia de 4 a 30 centímetros lo suficiente para cumplir con el requerimiento mencionado en la Tabla 10 que indica la adecuada capacidad de detectar obstáculos y así enviar la información al controlador para que este actúe y evada obstáculos.

2.2.5 Módulos de sistemas de navegación

A continuación, se realiza un análisis entre las familias de dispositivos de navegación Ublox NEO, teniendo en cuenta aspectos como la sensibilidad de navegación, precio, consumo de corriente, para lo cual se detalla en la Tabla 16 algunas características de estos dispositivos electrónicos.

TABLA 16. Especificaciones de módulos GPS

Módulos GPS	Sensibilidad de navegación (dBm)	Voltaje (v)	Corriente (mA)	Precio \$ (28/04/20)
Ublox NEO-6M v2	-161	2.5 - 5	35	15
Ublox NEO-7M v2	-162	2.7 - 5	35	20
Ublox NEO-M8 v2	-167	2.7 - 5	23	35
Modulo GY-273	Precisión			
Magnetómetro HMC5883L	1 a 2 grados Sexagesimales	3.3 - 5	20	5

Fuente. Tomado de (Amazon, 2020) , (Mercado Libre, 2020)

Después del analizar las características de la Tabla 16 el módulo más accesible y que tiene las mejores prestaciones para una adecuada operación remota por GPS como se indica en la Tabla 10 es el Ublox NEO-6M v2 y el magnetómetro HMC5883L, ya que el módulo GPS presenta cierta similitud a las dos versiones mencionadas como su alta sensibilidad de navegación y su costo bajo, lo que permite una adecuada comunicación con el magnetómetro y controlador para el vehículo terrestre no tripulado autónomo mencionada en la Tabla 11.

2.2.6 Cámaras en primera persona

Aquí, se realiza un análisis de dos minicámaras usadas con Arduino y drones, teniendo en cuenta aspectos como la resolución y el ángulo de visión, para lo cual se detalla en la Tabla 17 algunas especificaciones de estos dispositivos electrónicos.

TABLA 17. Especificaciones de cámaras FPV

Cámara	Resolución	Cuadros por segundo (fps)	Angulo de visión	Voltaje (v)	Corriente (mA)	Precio \$ (28/04/20)
VGA OV7670	640 x 480 (0.3 Mpx)	30	25°	3.3	20	8
Minicámara para dron	2.8mm 1/2 7" 3Mpx (1000TVL)	30	90°	3.5 - 5	70	40
Runcam Swift 2	2.5mm 600TVL	30	130°	5 - 36	70	60

Fuente. Tomado de (Amazon, 2020) , (Mercado Libre, 2020)

Una vez realizado el análisis de la Tabla 17 la cámara más acorde para cumplir con los requerimientos mencionados en la tabla 10 es la runcam swift 2 ya que presenta una resolución mayor a la cámara VGA también por tener un mayor ángulo de visión que la minicámara, lo que cubrirá más terreno para observar en el instante de tiempo que el vehículo terrestre transite.

2.2.7 Transmisor y receptor de video

A continuación, se realiza una comparación de los transmisores y receptores de video que usualmente son usados en drones, para lo que se tiene en cuenta aspectos como la frecuencia a las que se trabaja, potencias, y canales disponibles para usarse, para ello se detalla en la Tabla 18 algunas especificaciones de estos dispositivos de comunicación de video.

TABLA 18. Especificaciones de los tipos de transmisores y receptores de video

Transmisor	Frecuencia de trabajo (GHz)	Potencia de salida (mW)	Canales disponibles	Voltaje (v)	Corriente (mA)	Precio \$ (28/04/20)
TS832	5.8	600	40	7 - 16	600	25
TBS Unify Pro	5.8	200	40	7 - 16	600	40
Receptor	Frecuencia de trabajo (GHz)	Canales disponibles	Voltaje (v)	Precio \$		
RC832	5.8	40	5	30		
Gafas Fpv Para Dron	5.8	40	7.4	100		

Fuente. Tomado de (Amazon, 2020) , (Mercado Libre, 2020)

Luego de observar las características de los transmisores y receptores de la Tabla 18 cualquiera de los transmisores y receptores cumplen con una adecuada transmisión y recepción de video como indica en la Tabla 10, pero como un punto hace referencia a un bajo costo, se eligió el transmisor TS832 con su receptor RC832 se tiene en cuenta que el receptor no consta con una pantalla para visualizar la captura de la cámara pero es compatible para conectarse a una portátil, smartphone, Tablet o Tv por medio de un conversor USB a Jack Av.

2.2.8 Capacidad de batería

Para la elección de la batería se debe tener presente dos aspectos importantes, el tiempo de duración que el vehículo terrestre no tripulado autónomo va a estar en funcionamiento y el consumo de corriente de todo el vehículo terrestre es decir de todos los componentes que integran el prototipo. En la Tabla 19 se presenta un análisis del consumo de corriente de cada dispositivo electrónico que se eligió.

TABLA 19. Consumo de corriente de componentes electrónicos

Materiales	Voltaje (v)	Corriente (mA)	Corriente Total (mA)	Cantidad	Precio \$ unidad	Precio \$ Total
Arduino Due	7 - 12	93	93	1	30	30
Xbee Zigbee Modulo S2c	3,3	215	430	2	42	84
Motor brushless 2300KV + ESC 60A	5 - 12	4000	4000	1	75	75
Servomotor 12kg Mg995	4 - 7,2	70	70	1	11	11
Sensor Sharp GP2Y0A41SK0F 4-30cm	4,5 - 5,5	12	36	3	11	33
Modulo GPS NEO-6M v2	2,7 - 5	35	35	1	15	15
Magnetómetro HMC5883L	3,3 - 5	20	20	1	5	5
Runcam Swift 2	5 - 36	70	70	1	60	60
Transmisor de video TS832 y RC832	7,4 - 16	600	600	1	55	55
		Corriente Total (mA)	5354		Costo Total	368

Fuente. Tomado de (Amazon, 2020), (Mercado Libre, 2020), (Roda, 2020), (Mera & Ruano, 2019)

A continuación, con la información de la Tabla 19 obtenemos la corriente total de consumo de los equipos electrónicos la cual es 5354 mA por lo que se tendrá un consumo por hora de 5.35 A/hora, se realiza el cálculo de la capacidad de corriente que la batería

debe tener para lo cual se utiliza la Ecuación (1) la cual es el resultado del producto de consumo total del vehículo (Amperio/hora) y el tiempo de uso del vehículo (horas).

$$\text{Consumo por hora} = 5.35 \frac{\text{A}}{\text{h}}$$

$$\text{Tiempo de uso} = 45 \text{ min} = 0.75 \text{ h}$$

$$\text{Batería} = \text{consumo} * \text{tiempo} \quad (1)$$

$$\text{Batería} = 5.35 \frac{\text{A}}{\text{h}} * 0.75 \text{ h} = 4.01 \text{ A}$$

Una vez realizado el cálculo se concluye que para un tiempo de uso del vehículo de 45 minutos y un consumo de corriente de 5.35 Amperio/hora es necesario una batería de 4.01 Amperios, para lo cual es recomendable que sobrepase este valor ya que al momento de que el vehículo esté en funcionamiento los dispositivos electrónicos consumen un poco más de lo que el fabricante establece, por lo que con una batería Li-Po de 5000 mA/hora (5 A/hora) y 50 C (tasa de descarga), permita que el vehículo cumpla con una adecuada duración de la batería acorde a los requerimientos mencionados en la Tabla 10.

2.3 Resultados

Con base en la selección de materiales se realiza el diseño electrónico en el cual se especifica todos los componentes con su respectiva conexión con el fin de entender cómo es la estructura del vehículo.

2.3.1 Diagrama de bloques y descripción funcional

Para entender de una manera gráfica cómo es el funcionamiento del vehículo se indica en un diagrama de bloques como se observa en la Figura 27.

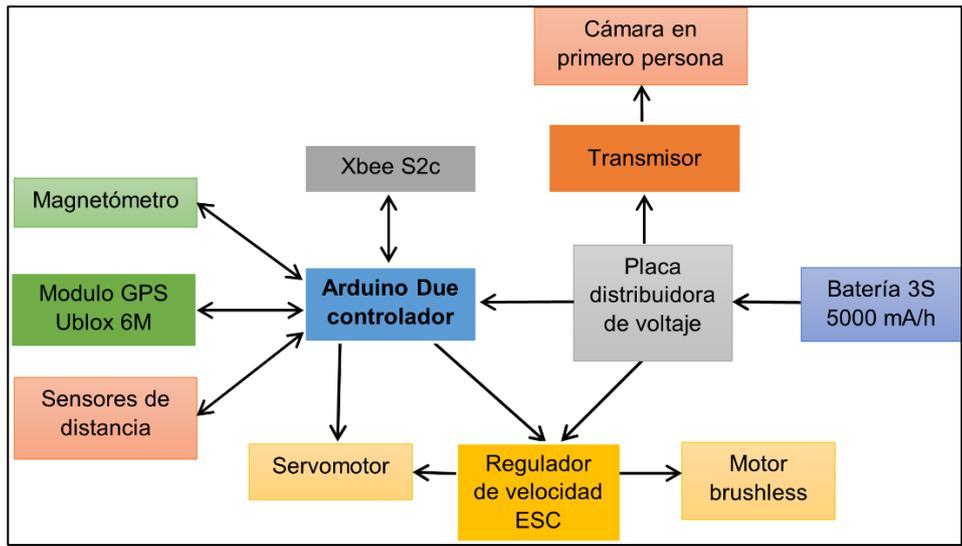


Fig. 27. Diagrama de bloques del vehículo terrestre no tripulado autónomo
Fuente. Autoría propia

En este diagrama de bloques se tiene los principales componentes del vehículo, en el cual la fuente de alimentación es una batería Li-Po que está conectada a una placa de distribución de voltaje, encargada de distribuir la energía a todo el sistema, también el módulo GPS que es el encargado de calcular la distancia que existe desde un punto inicial hasta un punto final con ayuda del magnetómetro, el cual ayuda a determinar la dirección en la que se encuentra un punto final por medio de la latitud y longitud que obtiene el módulo GPS.

El vehículo consta de un motor brushless para movilizarse y un servomotor encargado de generar la dirección o giro, este servomotor está alimentado del regulador de velocidad y la señal es enviada por el Arduino. A la placa de distribución de voltaje está conectado el transmisor de vídeo y éste a la cámara en primera persona para transmitir el video del recorrido que hace el vehículo al momento de trasladarse a la ubicación indicada.

2.3.2 Diseño electrónico del vehículo terrestre no tripulado autónomo

Una vez seleccionado los materiales que integraran el vehículo terrestre no tripulado autónomo, se realizó un diagrama de las conexiones de todo el diseño eléctrico con su respectiva estación base, la cual se encarga de monitorear el vehículo y en donde se observa la transmisión de video de la cámara como se muestra en la Figura 28.

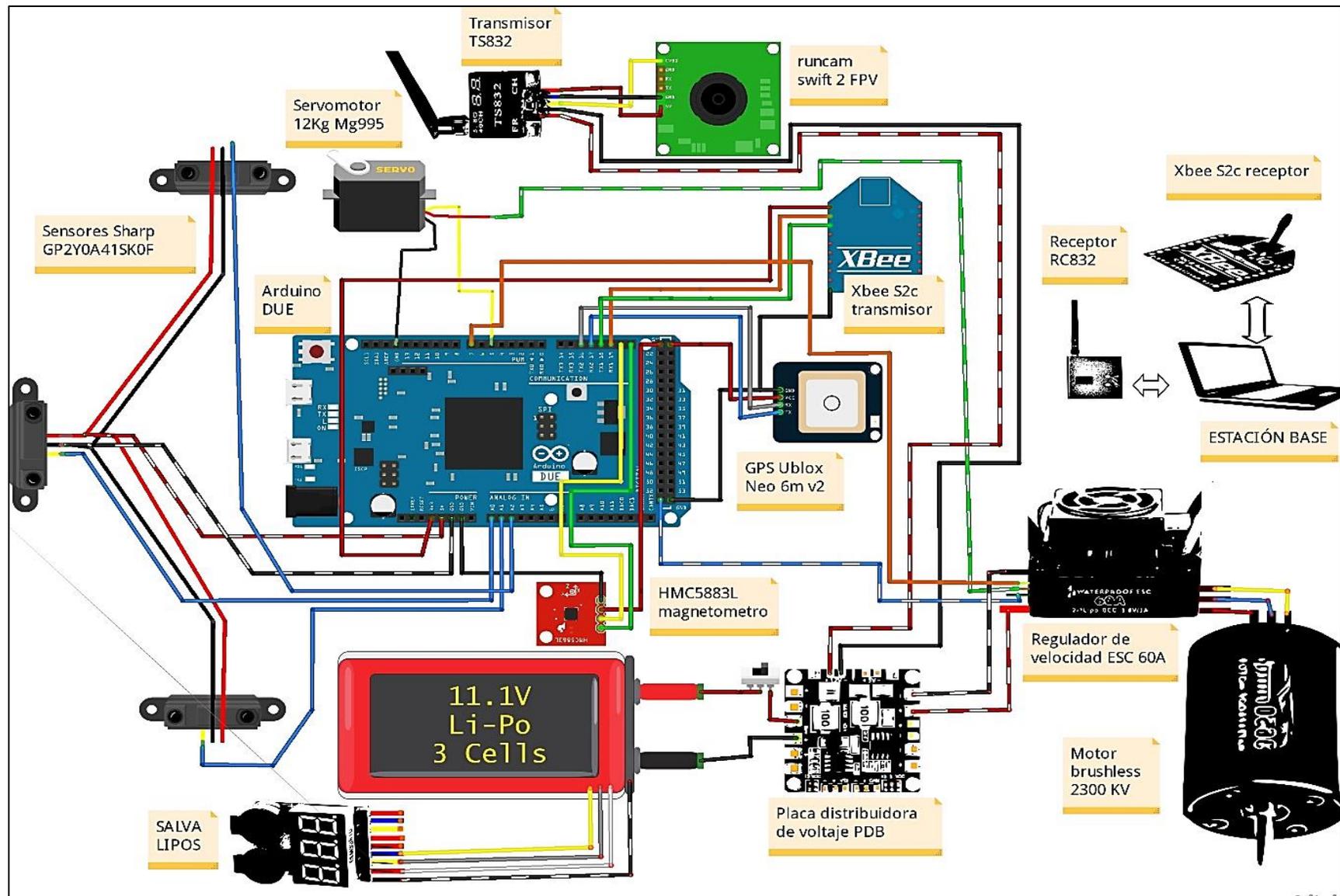


Fig. 28. Diseño electrónico de un vehículo terrestre no tripulado autónomo.
Fuente. Autoría propia.

La batería Li-Po es la encargada de alimentar todos los componentes, la que está conectada a una salva Li-Po para indicar el nivel de batería y proteger la vida útil de la misma. Esta batería también está conectada a una placa distribuidora de voltaje (PBD), la encargada de distribuir los voltajes adecuados para los componentes electrónicos, con el fin de no sobrecargar al Arduino y así conectar la alimentación del servomotor a la placa distribuidora de voltaje con el fin de reducir el ruido que genera el servo.

El circuito consta con 3 sensores fotoeléctricos Sharp análogos el cual tiene 3 pines que son voltaje (+5V), GND (-), y señal analógica. Estos sensores se ubican en la parte frontal del vehículo ya que serán los encargados de detectar obstáculos y enviar esa información al controlador Arduino para que este efectúe alguna acción para esquivar el obstáculo.

El módulo GPS Ublox Neo 6M v2 cuenta con los siguientes pines VCC (+5V), GND (-), RX y TX los cuales el RX se conecta al Tx del Arduino, y el TX se conecta al Rx del Arduino. Estos pines son los que transmiten la información por el puerto serial del software Arduino para indicar a que coordenadas el vehículo tiene que dirigirse con ayuda del magnetómetro el cual se conecta el pin VCC a 5v del Arduino, GND al Arduino y las señales de comunicación I2C que son los pines SDA, SCL al Arduino.

El módulo Xbee S2c cuenta con un pin para el voltaje VCC (+3.3V), un TX y RX que son conectados de la misma forma que el módulo GPS y un GND (-). Este dispositivo recibe la información por el puerto serial, enviada desde la base remota por medio de otro módulo Xbee S2c el cual será el transmisor de los datos y se conecta por medio de un Arduino Mega a la estación remota (portátil).

El transmisor TS832 cuenta con un pin VCC (7-20V) el cual es conectado directo a la placa distribuidora de voltaje, GND (-), la entrada de video la cual se conecta a la cámara, la salida de voltaje que se conecta a la cámara y GND conectado a la cámara. Este transmisor es el encargado de enviar la información de audio y video al receptor RS832 el cual se encuentra conectado a la estación remota (portátil) el cual se configura para que el reproductor VLC Media Player se observe la imagen de la cámara y poder hacer seguimiento del vehículo.

El motor que moviliza al vehículo está conectado a su driver o regulador de velocidad el cual es ESC de 60 amperios este consta de una entrada de alimentación de 7 a 12v, GND los cuales estarán conectados a la placa distribuidora de voltaje. También cuenta con 2 conectores que son de GND (-), y señal el cual se conecta a un pin digital con PWM con el fin de programar al Arduino para que este envíe la información al driver y por ende este envíe la corriente necesaria para hacer girar el motor a mayor o menor velocidad.

2.3.3 Tipo del vehículo terrestre no tripulado autónomo

Para el tipo de vehículo se optó utilizar la estructura de vehículos de control remoto el cual cuenta con el chasis, ruedas, dirección, y carrocería con el fin de adaptarle a conveniencia propia ya que más enfoque se da a la parte del diseño electrónico para esto se detalla en la Tabla 20 las características del vehículo.

TABLA 20. Característica del vehículo terrestre no tripulado autónomo

Características	Detalles	Tipo
Tipo de uso	El uso del vehículo es para respuesta de emergencia que hace referencia a búsqueda de personas extraviadas.	
Tipo de locomoción	Para este vehículo el tipo de locomoción es de cuatro ruedas las cuales las dos ruedas traseras son las que brindan la fuerza para mover el vehículo y las dos ruedas delanteras brindan la dirección de giro del vehículo.	
Tamaño	El tamaño real del vehículo terrestre no tripulado autónomo puede variar dependiendo del uso que se vaya a dar teniendo en cuenta todos los elementos que se añadirán al mismo, por tal razón las medidas aproximadas estarían entre 30 cm de largo, 25 cm de ancho, 30 cm de alto.	

Fig. 29. Diseño referencial del vehículo terrestre no tripulado autónomo
Fuente. Tomado de (FreeShopping, 2019)

Fuente. Autoría propia

Para el vehículo terrestre no tripulado se opta por un diseño 4x4, tipo de locomoción de cuatro ruedas en el que cabe recalcar que la Figura 26 es un diseño referencial del modelo

del vehículo, ya que se opta por utilizar la base de vehículos de control remoto la que se modifica conforme a todos los materiales a usarse en el vehículo.

Para que el vehículo posea suficiente torque para movilizarse por superficies planas es necesario la ayuda de piñones los cuales se encargan de reducir la velocidad y aumentar la fuerza por lo que se opta por usar un tren de engranajes o más conocidos como motorreductores.

Para conocer cuál es la relación de transmisión del tren de engranes y así conocer la velocidad a la que se reduce o aumenta un motor, se utiliza la ecuación (2) para un piñón y engranaje de tracción. Para un tren de engranajes se usa la ecuación (3) y para el cálculo de la relación de velocidad la ecuación (4) ya que una vez encontrado la relación de transmisión resta despejar la velocidad de salida y se halla la velocidad reducida. (Galbarro, 2021)

$$R_T = z_e / z_s \quad (2)$$

$$R_T = \frac{\text{conductoras}}{\text{conducidas}} = \frac{z_1 \cdot z_3}{z_2 \cdot z_4} \quad (3)$$

$$R_T = \frac{N_s}{N_e} \quad (4)$$

Donde:

R_T : relacion de transmision

Z_e : numero de dientes de entrada

Z_s : numero de dientes de salida

N_e : velocidad de entrada en rpm

N_s : velocidad de salida en rpm

Para el cálculo del torque que genera el tren de engranes se aplica la conversión de las revoluciones por minuto a radianes son sobre segundos utilizando la ecuación (5) y para el cálculo del torque se procede a utilizar la potencia del motor y su velocidad angular aplicando la ecuación (6).

$$\omega_e = N_e \cdot \frac{\pi}{30} \quad , \quad \omega_s = N_s \cdot \frac{\pi}{30} \quad (5)$$

$$T_e = \frac{P_e}{\omega_e} \quad , \quad T_s = \frac{P_e}{\omega_s} \quad (6)$$

2.3.4 Software

Para la programación se utiliza el software Arduino en el que se realiza el código para los sensores fotoeléctricos Sharp, la comunicación con los Xbee transmisor y receptor, la comunicación con el módulo GPS juntamente con el magnetómetro HMC5883L, y los actuadores que son los motores, para esto se debe obtener los respectivos cálculos para la dirección de giro del vehículo el cual depende del cálculo del vector de navegación, para esto el magnetómetro ubica la posición del vehículo (norte magnético) y el módulo GPS obtiene las coordenadas (latitud, longitud) en donde se sitúa el vehículo y las coordenadas a las cuales queremos que se dirija el vehículo.

Para esto se utiliza las fórmulas de Haversine la cual nos permite calcular las distancias entre la superficie de la tierra esférica mostradas en las Ecuación (7) (8) (9), las cuales son nada más un desglose de la formula general (Alvarez, 2019).

$$a = \text{sen}^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \text{sen}^2\left(\frac{\Delta\lambda}{2}\right) \quad (7)$$

$$c = 2 \cdot \arctan 2(\sqrt{a}, \sqrt{1-a}) \quad (8)$$

$$d = r \cdot c \quad (9)$$

Donde:

φ_1 : latitud 1 λ_1 : longitud 1

φ_2 : latitud 2 λ_2 : longitud 2

$\Delta\varphi$: $\varphi_2 - \varphi_1$ $\Delta\lambda$: $\lambda_2 - \lambda_1$

r : Radio de la tierra d : Distancia

Para el uso de esta fórmula es necesario tener las dos posiciones principales que son la posición en donde se encuentra ubicado el vehículo y la posición a dónde se requiera que se dirija el vehículo (latitud y longitud), para el radio de la tierra se utiliza el valor del radio Ecuatorial que 6378 km, pero esto viene con un margen de error ya que la tierra no es redonda completamente por lo que es un valor aproximado.

Una vez calculada la distancia que existe entre el vehículo y el punto de destino se calcula el ángulo horizontal con el fin de hallar la línea entre la posición del vehículo y el norte, y la línea entre la posición del vehículo y punto de destino para esto se utiliza la ecuación (10). (Alvarez, 2019)

$$\theta = \text{atan2}(\text{sen}\Delta\lambda \cdot \cos\varphi_2, \cos\varphi_1 \cdot \text{sen}\varphi_2 - \text{sen}\varphi_1 \cdot \cos\varphi_2 \cdot \cos\Delta\lambda) \quad (10)$$

Donde:

φ_1 : latitud 1 λ_1 : longitud 1

φ_2 : latitud 2 λ_2 : longitud 2

θ : Angulo de punto de destino $\Delta\lambda$: $\lambda_2 - \lambda_1$

Este método itera hasta que el vehículo llegue al punto de destino para esto se calcula el error de rumbo que no es más que la diferencia entre el ángulo de punto de destino y la dirección del vehículo con respecto al norte cómo se ejemplifica la ecuación (11).

$$\text{error de rumbo} = \theta - \theta_{vehiculo} \quad (11)$$

Cabe recalcar que la dirección del vehículo con respecto al norte se obtiene de la brújula digital y de este error de rumbo depende que el vehículo tome la decisión de girar hacia la izquierda, delante o derecha e ir comparando el error de rumbo nuevo con el anterior hasta que el vehículo se ubique en el punto de destino ordenado.

CAPÍTULO 3

Implementación del vehículo terrestre no tripulado autónomo

Este capítulo se explica la implementación del prototipo de vehículo terrestre no tripulado autónomo con los componentes seleccionados en el segundo capítulo, también el montaje, comunicación entre la estación remota y el vehículo para poder realizar pruebas y ajustes necesarios para observar los resultados, desempeño de los actuadores, sensores, módulo de navegación GPS lo que permitirá llegar a la correcta funcionalidad.

3.1 Metodología

La metodología utilizada para el desarrollo de este capítulo es el método en espiral lo que permite estar interactuando con los anteriores subtemas para ir encontrando una mejora del prototipo, esto solo cuando se presentan riesgos que impidan el correcto funcionamiento del vehículo terrestre no tripulado autónomo.

3.2 Descripción de costos comerciales de los componentes electrónicos elegidos para el vehículo terrestre no tripulado autónomo

Para la implementación de los componentes en el capítulo dos se detalló los costos de tiendas en línea ya que se indago valores encontrados en mercado libre, electrónicas online, etc. Una vez elegido los materiales se realizó la adquisición de los materiales y su valor real comercial se presenta en la Tabla 21.

TABLA 21. Costo de los componentes electrónicos comerciales

Materiales	Cantidad	Precio unidad	Precio Total (12/01/21)
Arduino Due	1	35,00	35,00
Xbee Zigbee Módulo S2c	2	40,99	81,98
Xbee Explorer USB	1	11,50	11,50
Shield Xbee Explorer	1	13,30	13,30
Motor brushless sin escobillas 2300KV + ESC 60A	1	74,98	74,98
Servomotor 12kg 180g Mg995	1	12,50	12,50
Sensor Distancia Sharp GP2Y0A41SK0F 4- 30cm	3	14,44	43,31
Módulo GPS Ublox NEO-6M v2	1	15,50	15,50
Magnetómetro HMC5883L	1	7,50	7,50
Cámara Fpv Runcam Swift 2	1	60,00	60,00
Batería Li-Po 3S 5A/h 50C	1	91,67	91,67
Transmisor De Video TS832 y RC832	1	54,99	54,99
Capturador de Video USB a RCA	1	14.50	14.50
Batería Li-Po 2S 500mA/h 30C	1	13.30	13.30
Placa distribuidora de voltaje	1	17,00	17,00
Otros	1	20	20
		Total	567,04

Fuente. Autoría propia

El total de todos los componentes electrónicos es de 547.04 dólares, pero a este valor se le añade el valor de los elementos extras (otros), como son cables de conexión, tornillos, tuercas, etc.

3.3 Distribución y montaje de los componentes electrónicos.

Para distribuir los componentes electrónicos se utilizó la base de un carro control remoto el cual ya dispone de chasis, ruedas, dirección y carrocería para ello se empieza montando los actuadores que son el motor brushless junto con el regulador de velocidad (ESC), el servo motor para la dirección, luego de ello los sensores, el Xbee S2c, el magnetómetro, el módulo del Sistema de Posicionamiento Global (GPS), el transmisor de video junto con la cámara en primera persona (FPV), y por último el controlador al cual se le conecta todos los componentes, para ello se explica por separado la forma de conexión de cada componente.

3.3.1 Montaje de los actuadores.

Para los actuadores se tiene de dos tipos, el primero que proporciona movilidad al vehículo por medio del motor brushless junto con su regulador de velocidad y el segundo que es la dirección por medio del servomotor.

a) Motor brushless.

Para el montaje del motor brushless con su regulador de velocidad se colocó en la base del vehículo, el cual se acopla a un tren de engranes de un motorreductor el que consta de 4 engranes, con el fin de reducir la velocidad del motor y aumentar el torque, para ello se hizo una pequeña comprobación para verificar la funcionalidad.



Fig. 30. Montaje de los actuadores
Fuente. Autoría propia

Para hallar la relación del motorreductor se utilizó los datos de los engranajes los cuales se toman los números de dientes de los engranes conductores y el número de dientes de los engranes conducidos.



Fig. 31. Motorreductor de 4 engranes
Fuente. Autoría propia

Este motorreductor consta de 3 engranes compuestos o dobles uno que es el conductor y el otro el conducido para ello se toma los datos y se procede a realizar el cálculo haciendo uso de la ecuación (3) para hallar la relación de transmisión.

Nº dientes conductores	Nº dientes conducidos
$Z_e = 15$	$Z_2 = 36$
$Z_1 = 12$	$Z_4 = 32$
$Z_3 = 12$	$Z_6 = 32$
$Z_5 = 12$	$Z_8 = 32$
$W_e = 50000 \text{ rpm}$	$P_e = 900 \text{ W}$

$$R_T = \frac{15 \cdot 12 \cdot 12 \cdot 12}{36 \cdot 32 \cdot 32 \cdot 32} = \frac{45}{2048}$$

Una vez obtenida la relación de transmisión se calculó la velocidad reducida del motor por medio de este motorreductor utilizando la ecuación (4) la cual despejaremos la velocidad de salida en rpm (N_s) para este motor los rpm máximos son 50000 acorde a los datos del fabricante.

$$N_s = N_e * R_T = 50000 \text{ rpm} * \frac{45}{2048} = 1098.63 \text{ rpm}$$

Luego se procede a calcular el torque que se aumentara con este motorreductor utilizando primero la ecuación (5) para hallar la velocidad angular del motor en radianes/segundos.

$$\omega_e = 50000 \text{ rpm} * \frac{\pi}{30} = 5235.98 \text{ rad/s}$$

$$\omega_s = 1098.63 \text{ rpm} * \frac{\pi}{30} = 115.048 \text{ rad/s}$$

Seguido de la ecuación (6) para encontrar el torque de salida con respecto al torque de entrada así observando cuanto es el incremento del torque gracias al motorreductor así el vehículo podrá soportar todo el peso de los materiales y poder movilizarse sin problemas.

$$T_e = \frac{900w}{5235,98 \text{ rad/s}} = 0.17 \text{ N} \cdot \text{m}$$

$$T_s = \frac{900w}{115.048 \text{ rad/s}} = 7.82 \text{ N} \cdot \text{m}$$

b) Servomotor

Este actuador se colocó en la parte delantera del vehículo ya que este proporciona la dirección o giro del vehículo. La alimentación es proporcionada por el regulador de velocidad del motor brushless ya que este cuenta con una salida de voltaje de 5v, también se le añadió la placa de distribución de voltaje para alimentar todo el vehículo.

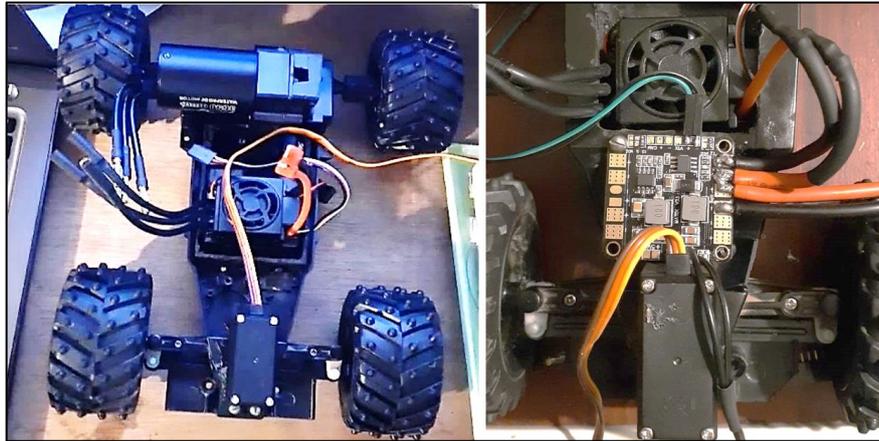


Fig. 32. Motorreductor del servomotor
Fuente. Autoría propia

Para observar de manera más clara las conexiones de los dos actuadores se realizó un diagrama separando todos los componentes del vehículo, para poder visualizar y entender de mejor manera el montaje como se muestra en la Figura 33.

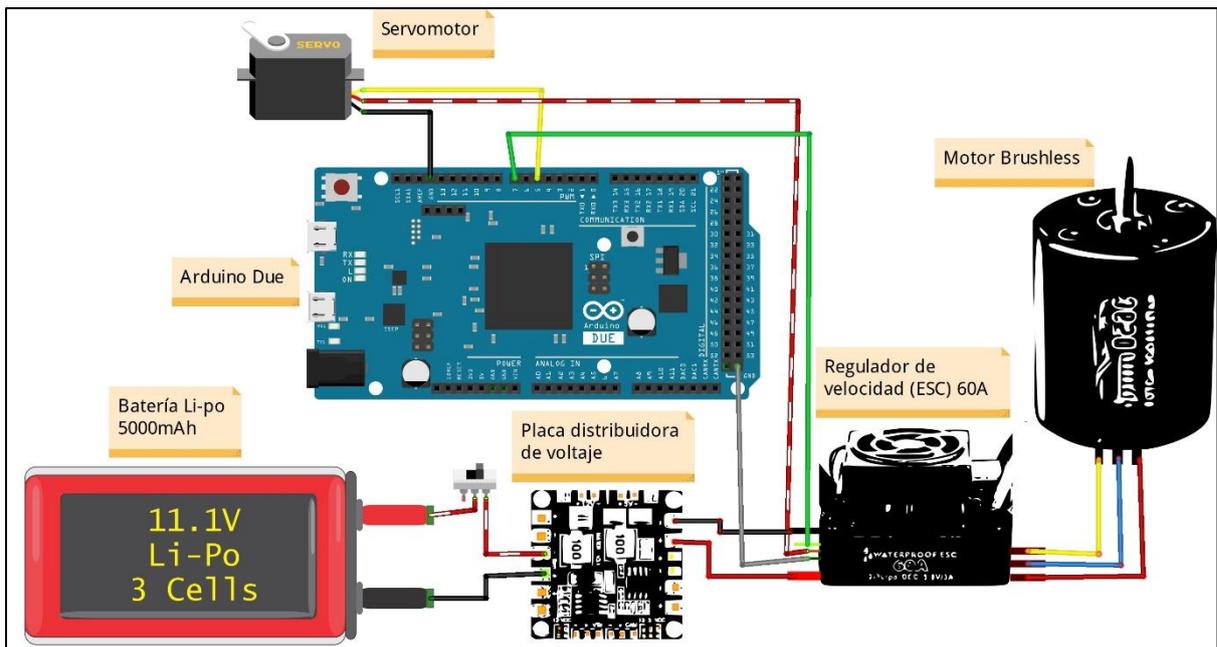


Fig. 33. Conexiones de actuadores
Fuente. Autoría propia

Como se observó en la Figura 33 la batería Li-Po junto con su placa distribuidora de voltaje son las encargadas de distribuir todos los voltajes necesarios a los distintos componentes electrónicos y a continuación la Tabla 22 que muestra los pines a conectarse.

TABLA 22. Conexiones de pines de los actuadores

Servomotor	Conexiones
VCC	5V (Regulador de velocidad ESC)
GND	GND (Arduino Due)
Señal	Pin 5 (Arduino Due)
Motor brushless	
Load -	Load - (Regulador de velocidad ESC)
Load señal	Load señal (Regulador de velocidad ESC)
Load +	Load + (Regulador de velocidad ESC)
Regulador de velocidad (ESC)	
GND	GND (Arduino Due)
Load señal	Pin 7 (Arduino Due)
5V	VCC (Servomotor)
12V	+ (Placa distribuidora de voltaje)
GND	- (Placa distribuidora de voltaje)

Fuente. Autoría propia

3.3.2 Montaje de sensores de distancia Sharp

Para el montaje de los tres sensores Sharp se ubicó en la parte delantera del vehículo el cual uno apunta con rección al centro, el segundo con dirección a la izquierda y el tercero con dirección a la derecha esto con el fin de que el vehículo tenga más ángulos de visión para poder evadir obstáculos.



Fig. 34. Montaje de los sensores de distancia Sharp

Fuente. Autoría propia

Para las conexiones se unió todos los negativos y todos los positivos de voltaje para evitar más conexiones y reducir cables, para ejemplificar como está la conexión de los sensores se muestran en la Figura 35 el esquema de conexión.

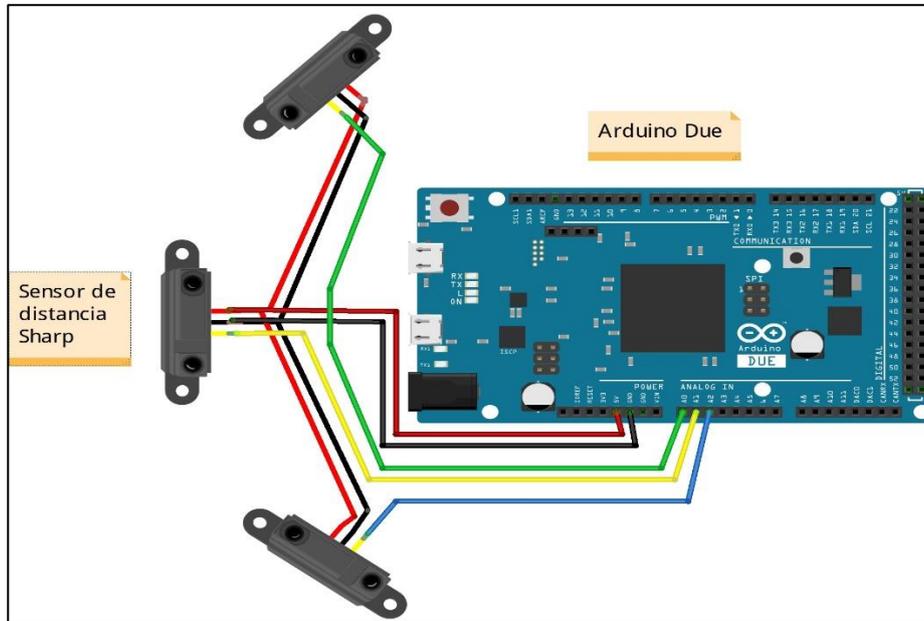


Fig. 35. Conexiones de los sensores de distancia Sharp
Fuente. Autoría propia

TABLA 23. Conexiones de los sensores Sharp

Sensores Sharp	Conexiones
+	5V (Arduino Due)
-	GND (Arduino Due)
señal	A1, A2, A3 (Arduino Due)

Fuente. Autoría propia

3.3.3 Montaje del módulo Xbee S2c en el vehículo y estación remota

Para el primer módulo Xbee se conectó al controlador el cual es el que reciba la información enviada por el transmisor, este es el segundo modulo Xbee conectado a la estación remota (portátil) por medio de un Arduino Mega el cual contiene la programación de control, los componentes se ubican en la parte de la cajuela de la carrocería del vehículo.



Fig. 36. Montaje de los Xbee S2c
Fuente. Autoría propia

Para las conexiones del módulo xbee se lo representa en el siguiente Figura 37 que es el diagrama de conexión para ejemplificar de manera una manera más clara.

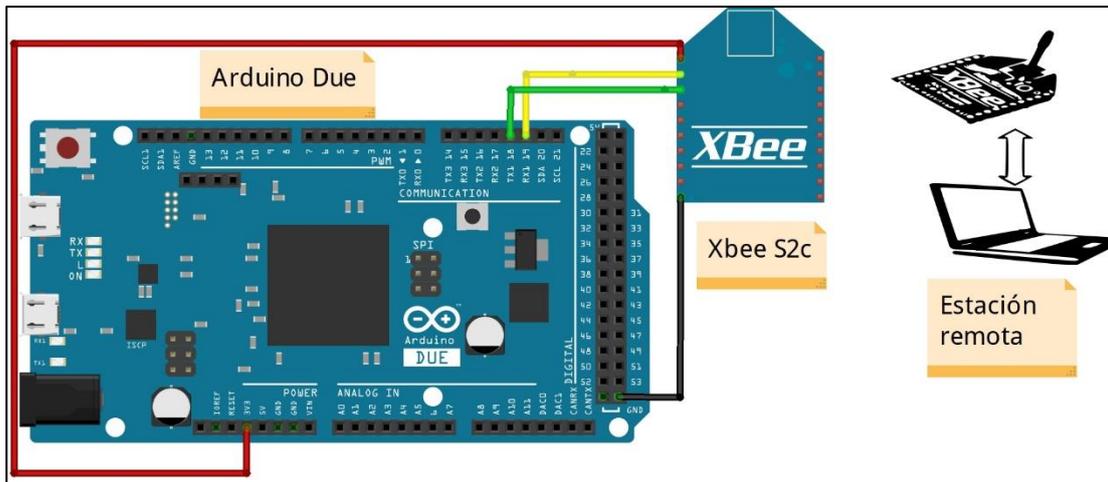


Fig. 37. Conexiones de los Xbee S2c
Fuente. Autoría propia

TABLA 24. Conexiones de Xbee S2c

Xbee S2c	Conexiones
1 VCC	3.3V (Arduino Due)
2 Data out (TX)	Pin RX1 (Arduino Due)
3 Data in (RX)	Pin TX1 (Arduino Due)
10 GND	GND (Arduino Due)

Fuente. Autoría propia

3.3.4 Montaje del módulo de sistema de posicionamiento global y magnetómetro HMC5883L

Para el módulo GPS se ubicó en la parte de arriba de la carrocería en una base que se colocó a una distancia de 15 cm del vehículo, para evitar interferencias, y que el módulo tome lecturas de mejor forma al momento de conectarse con los satélites, para el magnetómetro se colocó junto con el módulo GPS ya que este también debe estar alejado de los demás componentes para evitar interferencias magnéticas y que se vea afectado su desempeño.

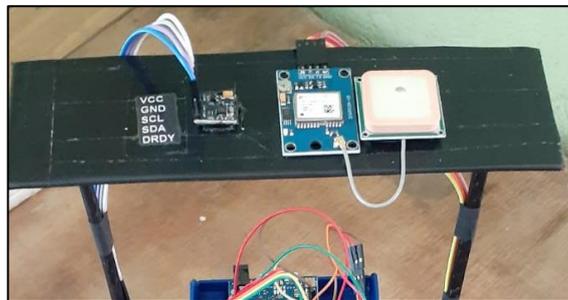


Fig. 38. Montaje del módulo GPS y magnetómetro HMC5883L
Fuente. Autoría propia

Para ejemplificar las conexiones se representa el diagrama en la Figura 39 separado de los demás componentes para comprender y observar de mejor manera el diseño electrónico.

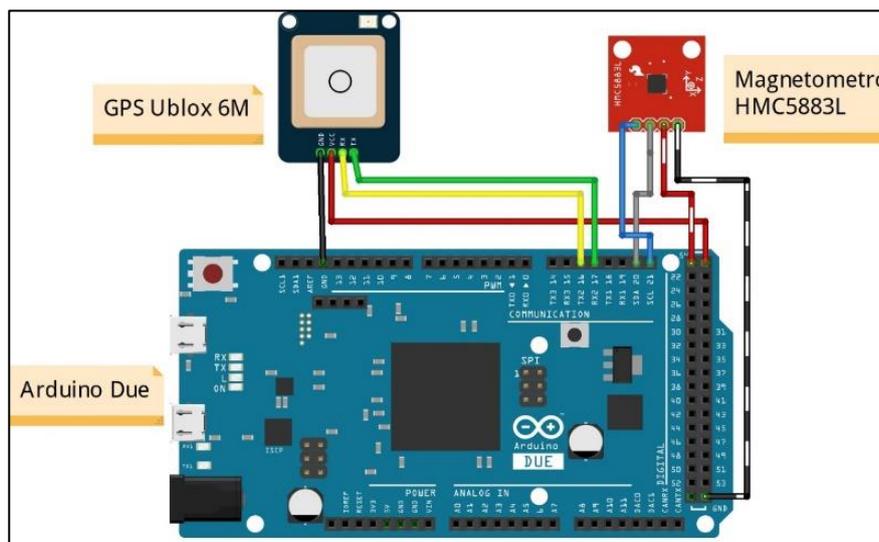


Fig. 39. Conexiones del módulo GPS y magnetómetro HMC5883L
Fuente. Autoría propia

TABLA 25. Conexiones del módulo GPS y magnetómetro

Módulo GPS Ublox 6m	Conexiones
VCC	5V (Arduino Due)
GND	GND (Arduino Due)
TX	Pin RX2 (Arduino Due)
RX	Pin TX2 (Arduino Due)
Magnetómetro HMC5883L	
VCC	5V (Arduino Due)
GND	GND (Arduino Due)
SDA	SDA (Arduino Due)
SCL	SCL (Arduino Due)

Fuente. Autoría propia

3.3.5 Montaje de la cámara en primera persona, el video transmisor TS832 y receptor RC832

Para el montaje del transmisor de video se ubicó a un lado del vehículo para no acumular todo en la parte de la cajuela de la carrocería.



Fig. 40. Montaje del transmisor TS832 de video
Fuente. Autoría propia

Para poder visualizar de manera más clara las conexiones se realizó el siguiente diagrama en la Figura 41 con su respectiva tabla de conexiones.

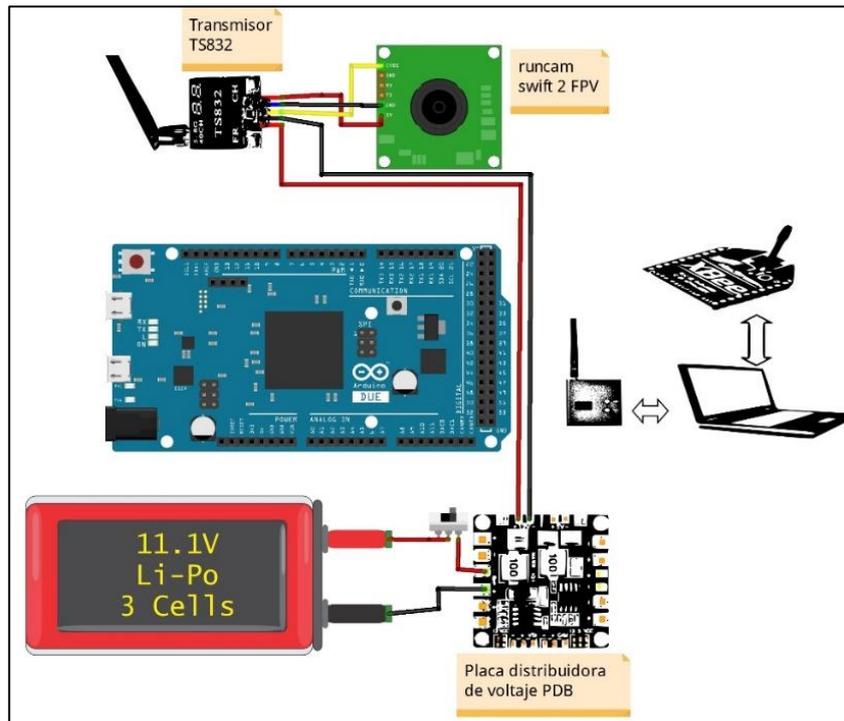


Fig. 41. Conexiones del transmisor de video TS832
Fuente. Autoría propia

TABLA 26. Conexiones de transmisor y receptor de video

Receptor RC832	Conexiones
DC IN	12V (Placa distribuidora de voltaje)
GND	GND (Placa distribuidora de voltaje)
VIDEO IN	SEÑAL (Cámara)
GND	GND (Cámara)
Voltaje OUT	VCC (Cámara)
Transmisor TS832	Portátil

Fuente. Autoría propia

3.4 Implementación del código de programación

Para realizar la programación del vehículo primero se configuró los dispositivos Xbee para lo cual se utilizó el programa X-CTU, luego de ello se realizó la programación en el programa Arduino en el que se desglosa en partes de programación como la de sensores de distancia, magnetómetro, módulo GPS, y actuadores.

3.4.1 Configuración de los Xbee

El programa XCTU permite configurar a los módulos Xbee en 3 diferentes tipos cómo es el modo Coordinador, Router y dispositivos finales para ello sí utilizó la configuración de un coordinador el cual está ubicado en la estación remota y un receptor el cual se ubica en el vehículo la configuración que se utiliza es punto a punto, en la cual la comunicación es bidireccional, la comunicación de estos dispositivos se la puede realizar en modo transparente (AT) o en modo comando (API) qué son mensajes por medio de tramas en este caso se utilizó la configuración en modo AT.

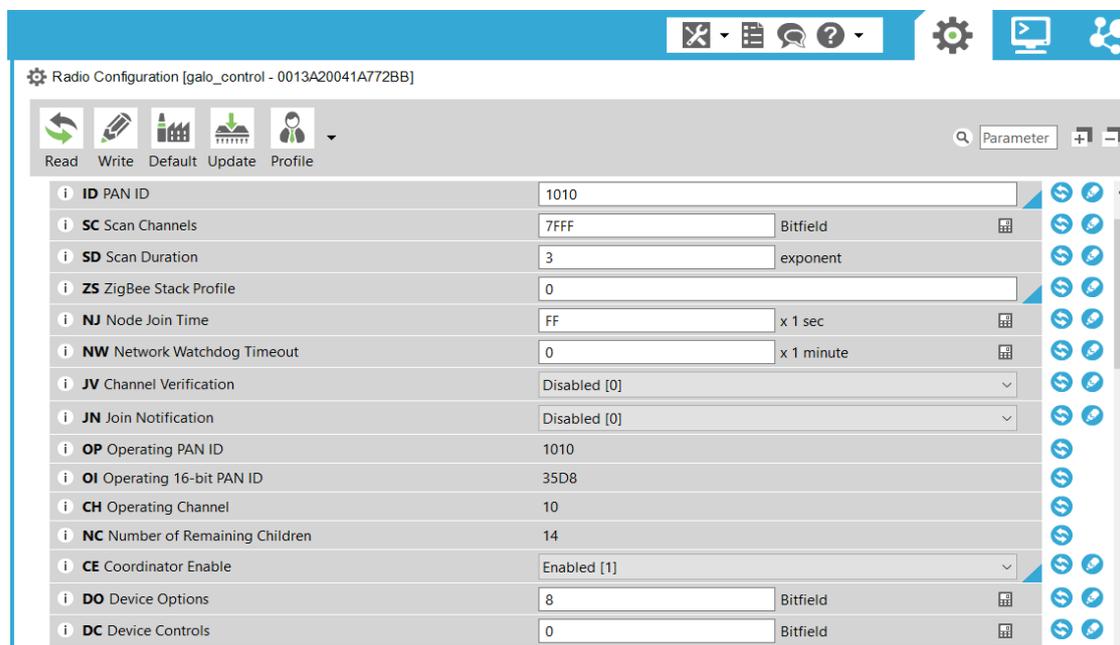


Fig. 42. Configuración Xbee
Fuente. Autoría propia

La Figura 42 muestra el entorno del programa XCTU, no es muy complejo y para la configuración AT los únicos campos que se configura son los mostrados en la Tabla 27.

TABLA 27. Configuración de Xbee coordinador y Router

Coordinador	Router
Cambiar PAN ID = 1010	Cambiar PAN ID = 1010
Habilitar coordinador	Habilitar verificación de canal
DH = 13A200	DH = 13A200
DL = 41A772BF	DL = 41A772BB

Fuente. Autoría propia

Con este tipo de configuración los Xbee se pueden comunicar bidireccionalmente el Coordinador puede enviar y recibir información, el Router de igual manera recibir y enviar información al coordinador.

3.4.2 Programación del código del vehículo terrestre no tripulado autónomo

Para la programación se distribuye en 7 partes, la cual se explica mediante flujogramas la programación, éstas están conformadas de la siguiente forma, programación de los sensores Sharp, magnetómetro, módulo GPS, filtros de los datos del GPS, cálculos del vector dirección, actuadores y control de la base remota.

3.4.2.1 Programación de sensores Sharp

Para realizar la programación de los sensores Sharp, se realizó 20 lecturas de los datos tomados del sensor de distancia para sacar un promedio, con el fin de reducir los picos que se producían al tomar las lecturas y trabajar con un valor más preciso. Este proceso se lo detalla en la Figura 43 que es un flujograma del proceso que realiza el Arduino con el sensor Sharp.

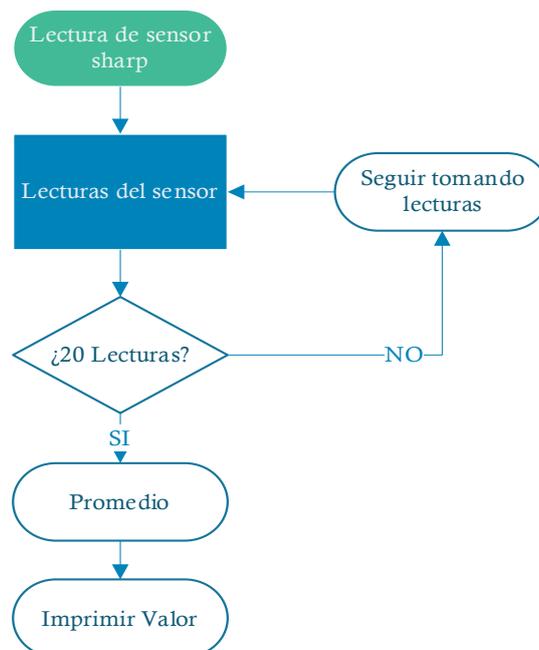


Fig. 43. Diagrama de flujo de los sensores Sharp
Fuente. Autoría propia

3.4.2.2 Programación del Magnetómetro

Para la programación del magnetómetro se utilizó las librerías “Wire, I2Cdev, HMC5883L”, para lo cual este tomaba el norte magnético el cual situaba a la variable mx para luego sacar el arco tangente de my/mx y transformar ese valor a grados ya que está en radianes, a qué se utiliza la declinación magnética que existe en cada zona del mundo para ello se consulta en la página y se verifica que la declinación magnética de Ibarra es -4°25'. (Magnetic declination, 2021)

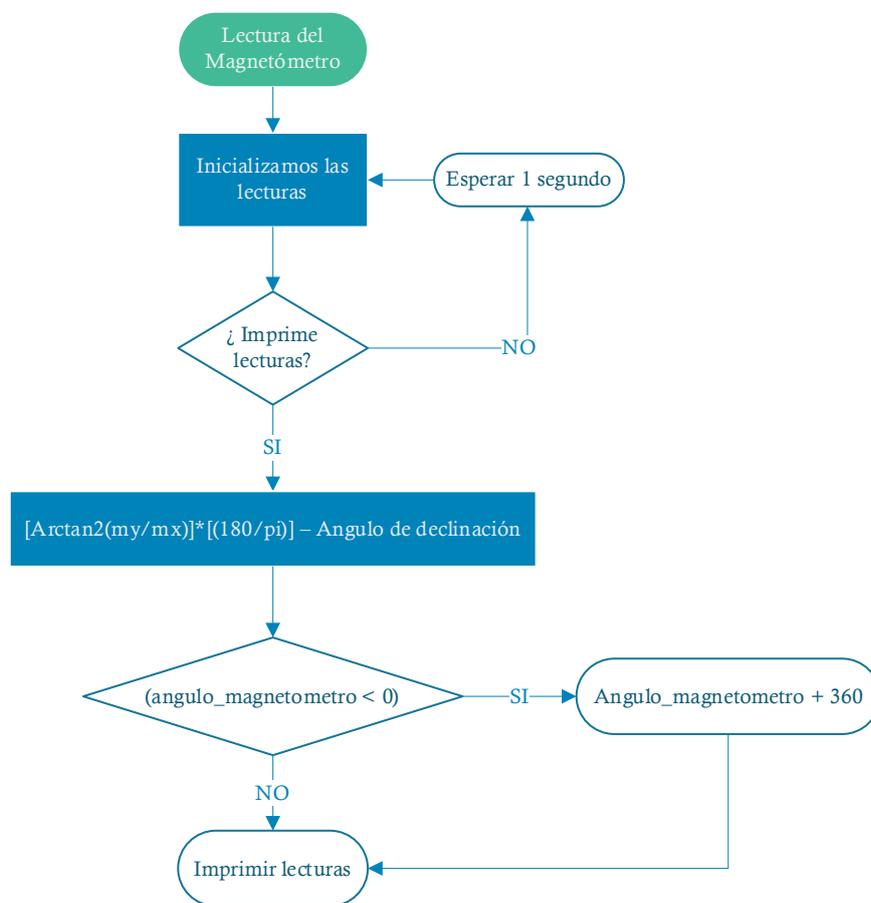


Fig. 44. Diagrama de flujo del Magnetómetro
Fuente. Autoría propia

Una vez que se calcula el ángulo del magnetómetro se realiza una comparación si ese ángulo es menor a 0 se realiza una suma de 360° para imprimir caso contrario solo se imprime el valor.

3.4.2.3 Programación del módulo GPS

Para esta programación se hace uso de la librería "TinyGPS" como el Arduino Due no viene incorporado una librería del puerto serial se trabaja directamente con los puertos seriales que tiene por defecto.

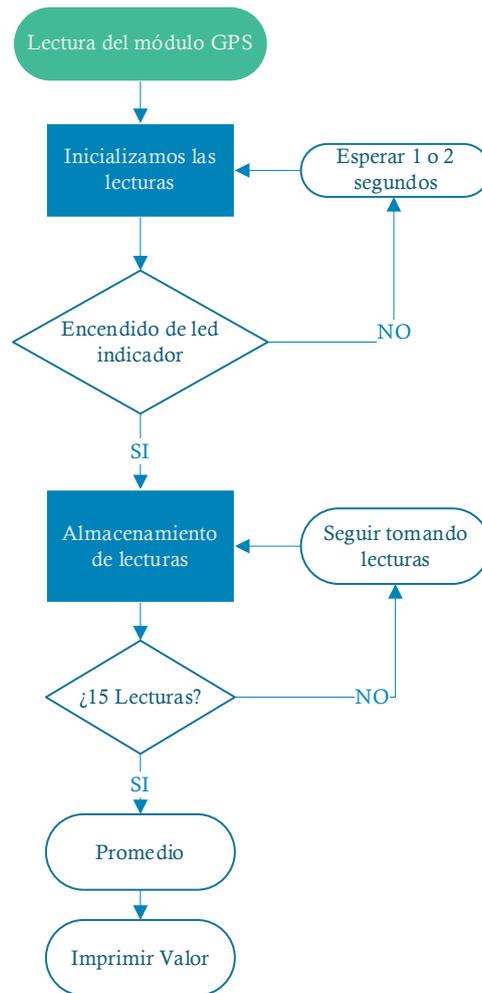


Fig. 45. Diagrama de flujo del módulo GPS
Fuente. Autoría propia

Para verificar que el módulo GPS está tomando lecturas se debe verificar que el led indicador parpadee y así se procede a tomar 15 lecturas para sacar un promedio para proceder a realizar el resto del proceso.

3.4.2.4 Programación de los filtros o muestras de los datos GPS

En esta parte se realizó el filtrado de todos los datos obtenidos del GPS para mejorar los resultados finales ya que en un inicio se tomó sólo los valores directos del GPS y el vehículo enloquecía al momento de tomar el rumbo hacia el punto marcar por medio del filtrado de los datos se mejoró la parte de dirigir el vehículo a su destino para ello se muestra la curva que se genera de un dato directo del GPS y un dato filtrado por medio del promedio de muestras en la Figura 46.

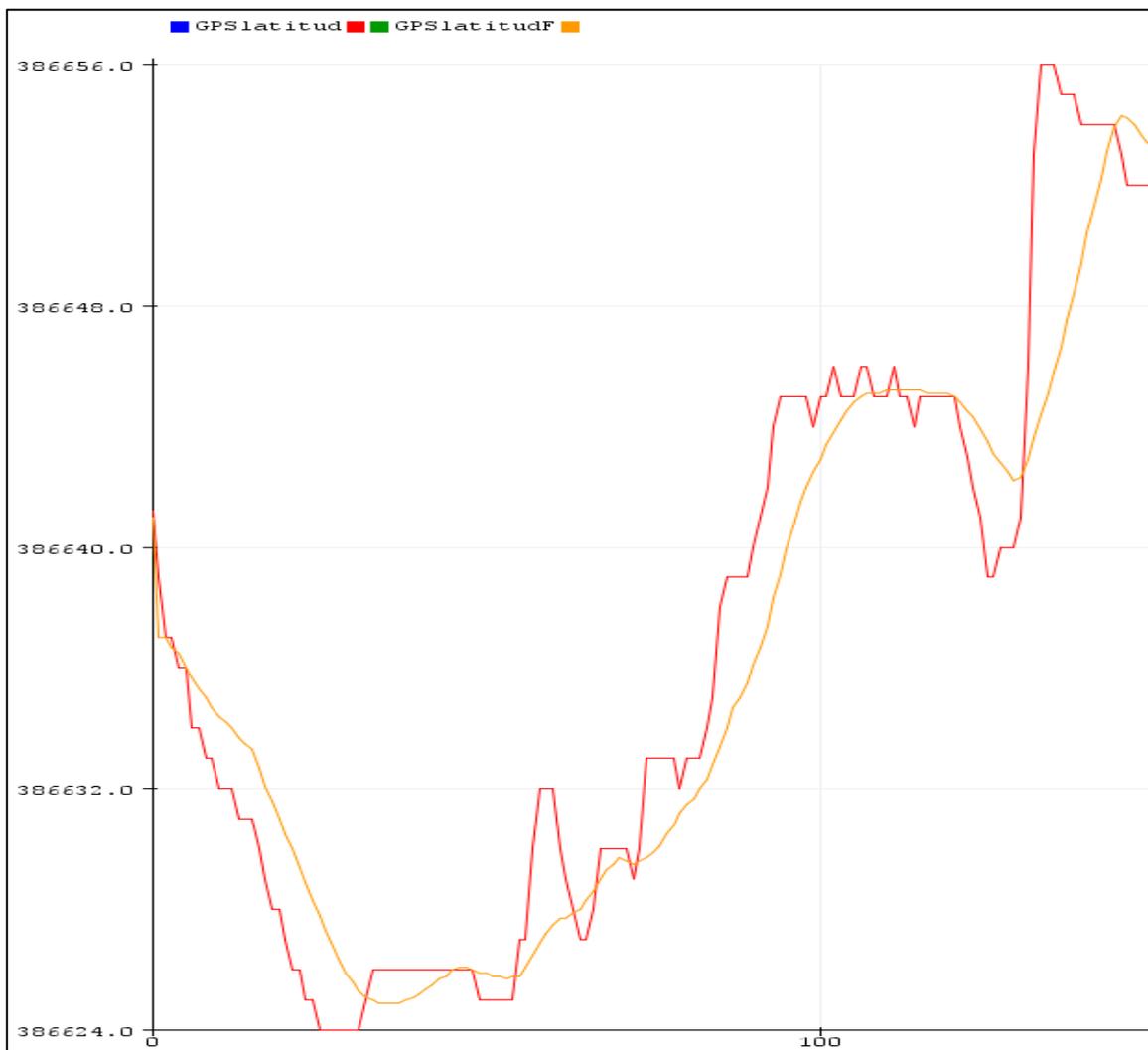


Fig. 46. Lecturas de valores GPS con filtro y sin filtro
Fuente. Autoría propia

Como se observa en la Figura 46 el color rojo es el dato recogido directo del GPS y el color amarillo el dato filtrado, se observa que se eliminan los picos reduciendo la inestabilidad y mejorando el comportamiento del vehículo.

3.4.2.5 Programación para el cálculo del vector posición

La programación para encontrar el vector posición es mediante el método de haversine que no es más que la medición de distancia que existe entre el vehículo y la ubicación a dirigirse para ello se utilizó las ecuaciones (7), (8), (9), y también el radio de la tierra que es un valor aproximado más 6378 km, pero aquí se genera un error ya que este valor asemeja a la tierra como que fuese completamente esférica.

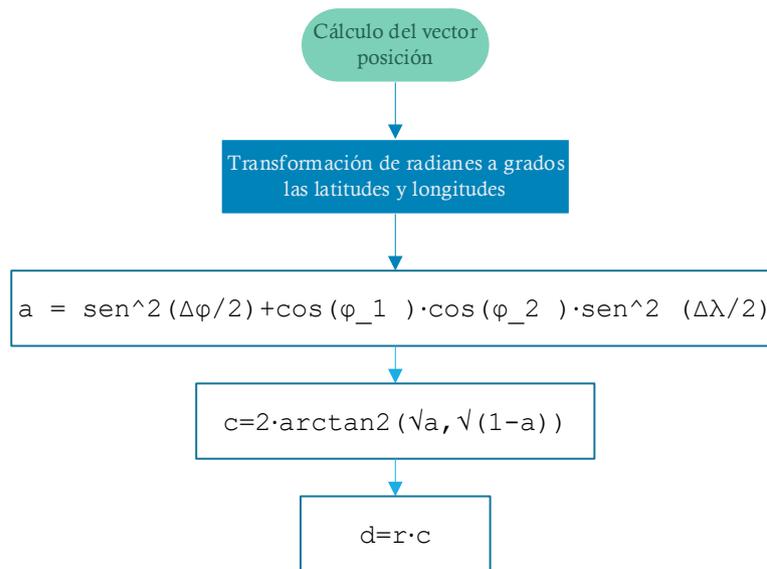


Fig. 47. Diagrama de flujo del vector posición
Fuente. Autoría propia

Luego de haber calculado la distancia se calcula el ángulo que tiene el vehículo con respecto a la ubicación establecida y por último se calcula el error de rumbo por medio del cual el vehículo tomará la decisión de dirigirse a la izquierda, derecha o centro haciendo uso de las ecuaciones (10), (11).

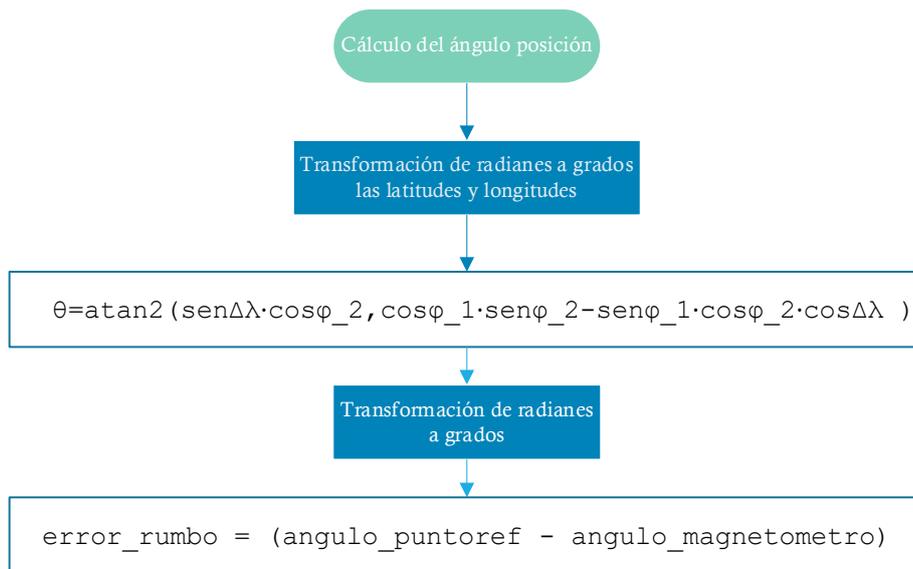


Fig. 48. Diagrama de flujo del ángulo posición
Fuente. Autoría propia

3.4.2.6 Programación de los actuadores

En esta programación en un principio se tenía declarado tres estados los cuales que eran izquierda, derecha y centro distribuidos de -180° a -90° izquierda, -90° a 90° centro, 90° a 180° derecha pero presentó desviaciones muy grandes hacia el punto marcado por lo que se optó por distribuir en 5 estados para reducir el ángulo en el cual el vehículo tome la acción de dirigir se a la izquierda derecha o centro de la siguiente forma -180° a -45° izquierda rápido, -45° a -5° izquierda, -5° a 5° centro, 5° a 45° derecha lento, 45° a 180° derecha rápido, por medio de estos se logró que el vehículo tenga menos grados de desviación que logra llegar al punto indicado correctamente.

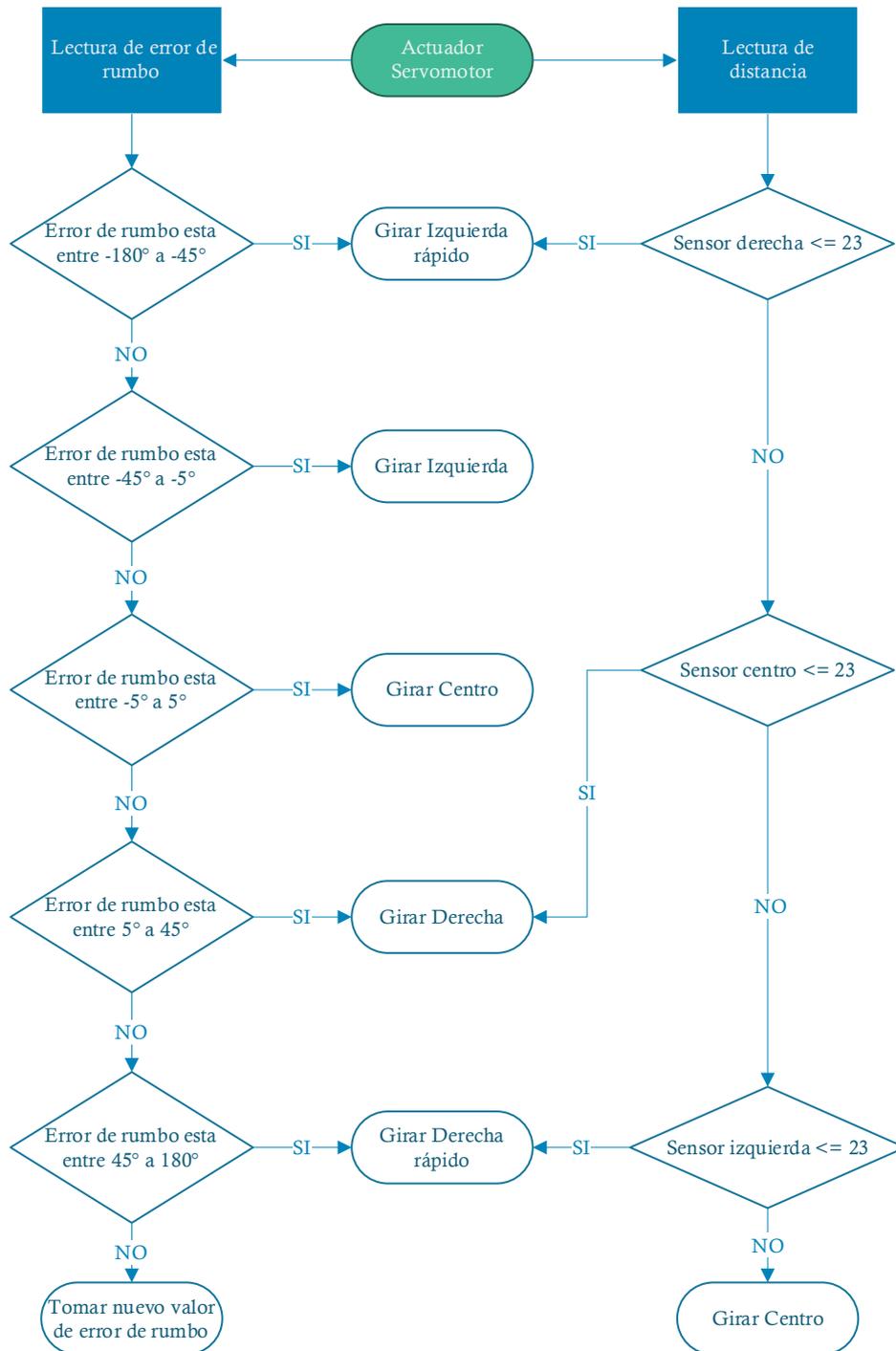


Fig. 49. Diagrama de flujo de los actuadores
 Fuente. Autoría propia

Para las condiciones de los sensores se cómo se observa en la Figura 49 se declara distancias para los tres sensores ubicados en puntos específicos para que al momento de detectar un objeto proceda a esquivar realizando la acción indicada.

3.4.2.7 Programación del control de la base remota

Para el control de la base remota se ocupó la librería “SoftwareSerial” la cual permitirá una comunicación por el puerto serial por medio de los Xbee para poder enviar la información que los puntos o coordenadas para que el vehículo se dirija y el vehículo enviaba las coordenadas para poder visualizar el recorrido del vehículo haciendo uso de Google Maps.

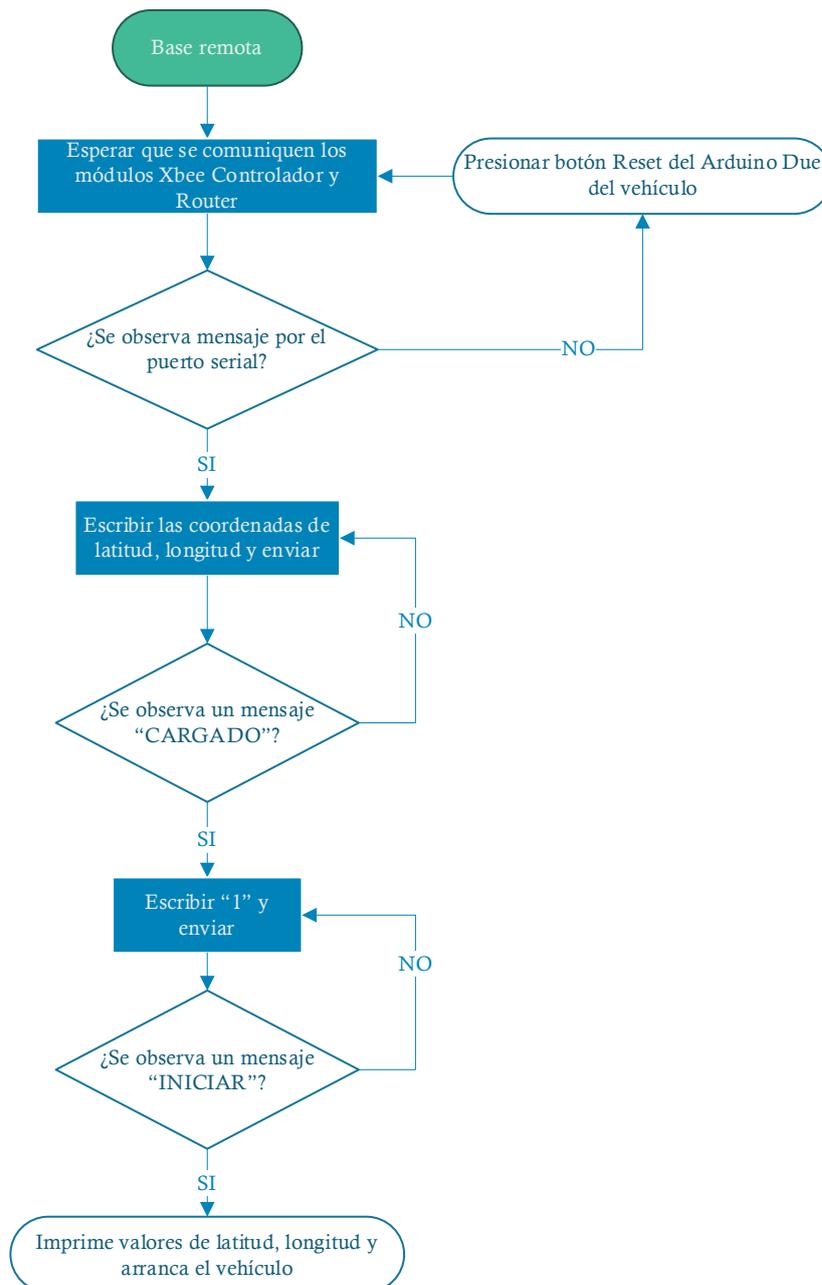


Fig. 50. Diagrama de flujo de la base de remota
Fuente. Autoría propia

3.4.2.8 Configuración de la cámara runcam con el transmisor de video

Para la obtención de video de la cámara ronca se utilizó la aplicación reproductora multimedia VLC el cual nos permite visualizar la transmisión de la cámara por medio de un Capturador de Video USB a RCA que se conecta al receptor RC832 para lo que se configura el reproductor VLC como se muestra en la Figura 51.

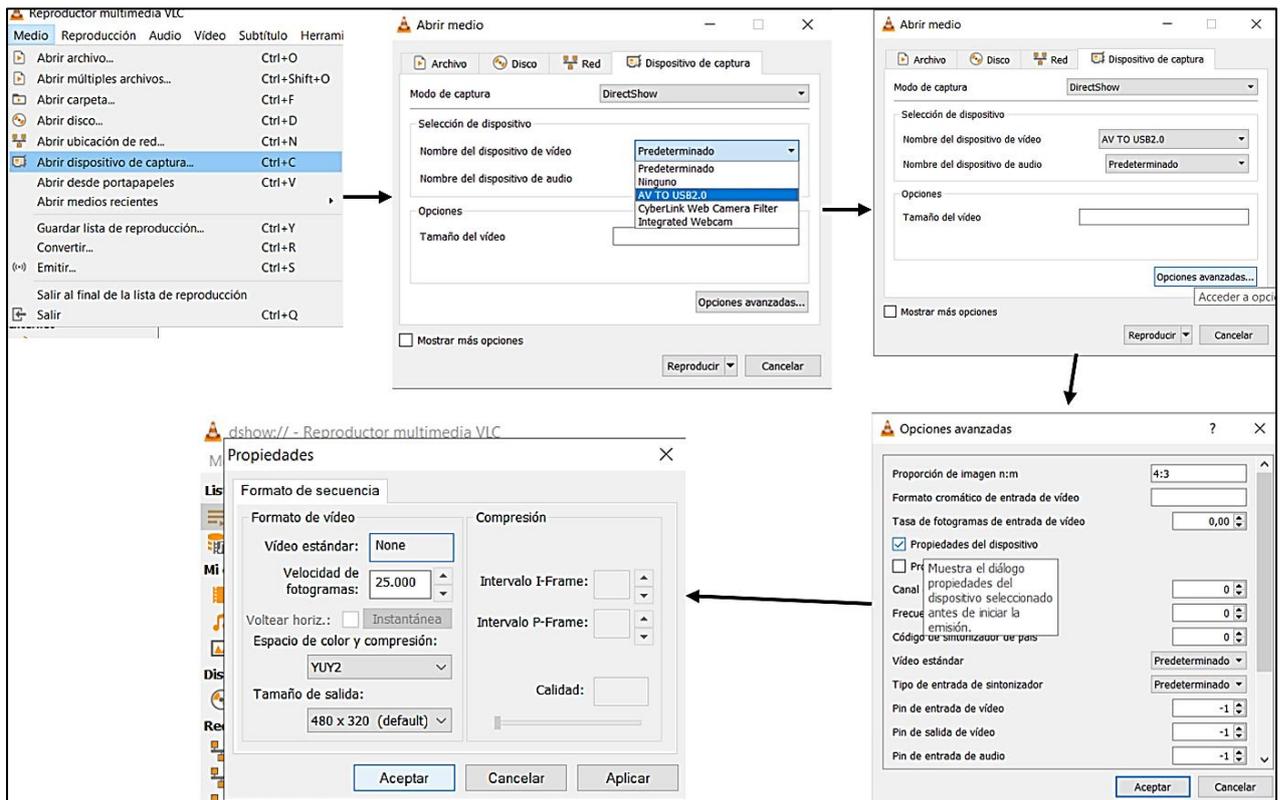


Fig. 51. Configuración de la recepción de la cámara de video
Fuente. Autoría propia

Con la configuración mostrada en la Figura 51 el transmisor de video comunicaría con el receptor y se transmitiría el video de la cámara al computador para poder visualizar el recorrido del vehículo. La cámara cuenta con un OSD para configurar directamente a la cámara como es el brillo, nitidez, etc.

3.5 Ejecución del vehículo terrestre no tripulado autónomo pruebas y ajustes

Para la ejecución en el vehículo terrestre no tripulado autónomo se realizó las pruebas en el autódromo de Yahuarcocha con el permiso de los bomberos. Para la ejecución se realizó pruebas con obstáculos y sin obstáculos para una y cuatro ubicaciones y se obtuvo los siguientes resultados como se muestra en la Figura 52.



Fig. 52. Ruta del vehículo terrestre no tripulado autónomo.
Fuente. Tomada de (Google Maps, 2021)

En la Figura 52 la línea azul indica la ruta que se estableció para que el vehículo se dirija del punto de salida al punto llegada, la línea roja indica la prueba de una sola ubicación, la línea verde indica la prueba de las 4 ubicaciones, para ello se indica el recorrido que tuvo y el margen de error del vehículo al ubicarse en cada punto marcado.

Cabe mencionar que el módulo GPS ofrece una precisión de posicionamiento horizontal de 2.5 metros en condiciones ideales y el Magnetometro de 1 a 2 grados de precisión sin contar que el vehículo cuenta con sensores de distancia los que al momento de esquivar un obstáculo generan que el vehículo se desvíe de la ruta así generando más errores. (Alvarez, 2019)

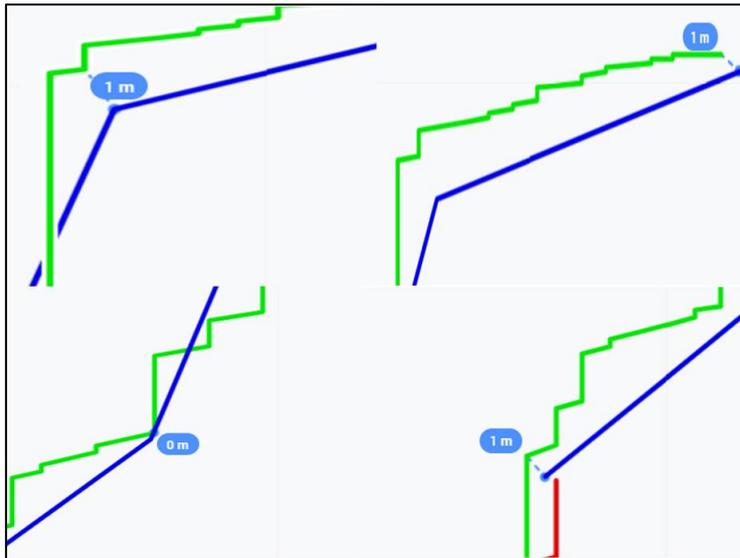


Fig. 53. Error de desviación máxima del vehículo con respecto a las coordenadas establecidas.
Fuente. Tomada de (Google Maps, 2021)

Como se observa en la Figura 53 la distancia más alejada de la ruta trazada es de 1 metro presenta más desviaciones en el transcurso de dirigirse al otro punto esto se debe a que los sensores están actuando y tratan de esquivar obstáculos generando desviaciones y añadiendo más errores a los cálculos, pero mientras más se acerca a la ubicación su error de desviación se acorta.

3.6 Resultados

Con base en las pruebas realizadas se detalla la funcionalidad que se obtuvo componente a la hora de su ejecución y permitió verificar como fue el desempeño total del vehículo terrestre no tripulado autónomo.

3.6.1 Funcionalidad del controlador

En un principio se empezó a utilizar el Arduino mega para el desarrollo del prototipo, pero conforme se realizaban pruebas y se implementaban más componentes al vehículo, la velocidad de respuesta de procesamiento empezaba a verse afectada por lo que se utilizó un Arduino Due el cual tenía el doble de prestaciones que el Arduino mega. Se ubicó al Arduino Due en el vehículo para comandar todos los componentes del vehículo y el Arduino mega se utilizó como estación remota para adquirir los datos y comandar al vehículo a

distancia ya que sólo efectuaba el trabajo de comunicación serial mediante el Xbee coordinador. Los parámetros que se consideraron para las pruebas del vehículo se detallan en la Tabla 28.

TABLA 28. Resultados de la ejecución del controlador

Parámetros	Resultados
Conexiones	Correctas
Envío de datos	Adecuada
Recepción de datos	Adecuada
Velocidad de respuesta a instrucciones	Adecuada

Fuente. Autoría propia

3.6.2 Funcionalidad de los actuadores

Como se había mencionado en el apartado 3.3.2.6 en un inicio se realizó tres estados para los motores, los cual era derecha centro izquierda y se había establecido una velocidad constante esto generó que el error de ángulo se distribuya -180° a -90° , -90° a 90° , y 90° a 180° ocasionado que el rango de estado sea muy grande y el vehículo no corrija su ubicación constantemente para ello se establecieron a cinco estados ya antes mencionados.

El vehículo tiene un peso de 4 libras con dimensiones como alto 40 cm, largo 35 cm, ancho 23 cm, para lo cual el motor soporta el peso sin problemas, sin sobrecalentamientos y sin bajar su rendimiento. Por otra parte, el servomotor que se encuentra en la parte delantera también responde perfectamente a los giros considerando el peso total del vehículo. El prototipo está programado para ir a una velocidad de 5 a 6 km/h aproximadamente este valor se obtuvo por medio de una aplicación de celular llamada velocímetro GPS, el valor completo de la velocidad esta entre 10 a 15 km/h ya que se encuentra colocado un motorreductor el cual disminuye su velocidad, pero aumentando su torque.

El vehículo fue probado en subidas con superficie plana, superficies completamente planas, semiplanas como es piso arenoso y por último en superficies irregulares como se

observa en la Figura 54. También se identificó que en superficies planas rectas o inclinadas el vehículo desempeña con normalidad, en superficies semiplanas se redujo a un 80% de su desempeño normal, pero en superficies irregulares, rocosas, el vehículo transita, pero con dificultad debido a que el chasis no cuenta con sistema de amortiguación por lo que el vehículo se sacude en ese tipo de superficies.



Fig. 54. Pruebas en superficies planas, inclinadas, e irregulares.
Fuente. Autoría propia

TABLA 29. Resultados de la ejecución de actuadores

Parámetros	Resultados		
	Motor Brushless	Regulador de velocidad ESC	Servomotor
Conexiones	Correctas	Correctas	Correctas
Voltaje de operación	Correcto	Correcto	Correcto
Ensamblaje	Correcto	Correcto	Correcto
Funcionamiento continuo	Si	Si	Si
Sobrecalentamientos	No	60%	No
Vibraciones	No	No	No

Fuente. Autoría propia

En la Tabla 29 se describen los resultados que se tomó en cuenta para verificar el rendimiento del motor brushless y el servomotor, concluyendo así que si se desea trabajar con velocidades más elevadas es recomendable usar un regulador de velocidad que soporte más corriente, con el fin de no dañar el regulador de velocidad ya que el motor en el arranque tiende a generar una gran corriente por periodos cortos igual o mayor a 60 Amperios.

3.6.3 Capacidad de comunicación

Para los módulos Xbee coordinador y Xbee Router se tomó el tiempo que tardan en establecer conexión entre ellos, este tiempo variaba en cada prueba realizada que era de 2 a 10 segundos y en ocasiones de 30 a 60 segundos. Una vez enlazado, la comunicación era fluida, a distancias cercanas no había retrasos ni desconexiones en las pruebas realizadas. La distancia máxima que se realizó fue de 106 metros y la comunicación de los Xbee no presentaba problemas, gracias a ello se obtuvo todas las coordenadas de la ruta completa del vehículo.

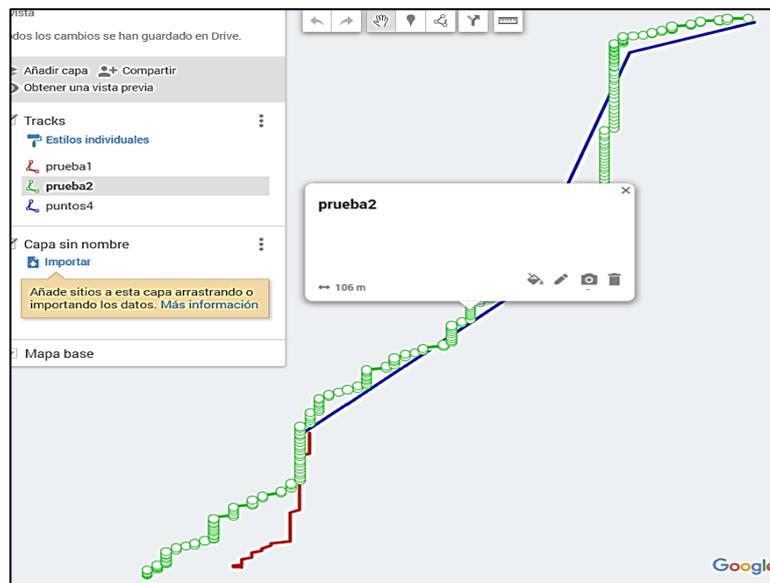


Fig. 55. Lecturas de la trayectoria del vehículo.
Fuente. Tomada de (Google Maps, 2021)

Como se observa en la Figura 55 los puntos verdes son las lecturas captadas por el Xbee coordinador cada lectura es un círculo graficado que llega hasta el final de la ruta del vehículo.

3.6.4 Respuesta del módulo posicionamiento global neo Ublox 6m

En las diferentes pruebas que se realizó, el vehículo se posicionaba a una distancia de 30 centímetros a 1 metro, aunque varias veces llegaba a coincidir con el punto indicado. Para lugares abiertos la respuesta era mucho mejor que en lugares cerrados y los puntos

ubicados a una distancia de 20 metros o más respondía correctamente, cuando los puntos se encontraban a una distancia de unos 5 a 10 metros el vehículo se acercaba al punto, pero divagaba en círculos con respecto a la ubicación.



Fig. 56. Respuesta del módulo GPS.
Fuente. Tomada de (Google Maps, 2021)

Otro aspecto que se menciona es que los sensores al momento de detectar un obstáculo desvían al vehículo de la ruta indicada lo que hace que genere más errores de desviación, pero no impide que el vehículo termine su recorrido como se muestra en la Figura 56.

3.6.5 Respuesta de sensores Sharp

Para el funcionamiento de los sensores de distancia se realizó en la programación un promedio de 20 muestras para reducir mediciones erróneas y no exista desviaciones que interfiera con el recorrido del vehículo. En la Figura 57 se muestra la prueba realizada con un obstáculo, al final se realizó con tres obstáculos y con una coordenada ingresada.



Fig. 57. Vehículo autónomo evitando obstáculo
Fuente. Autoría propia

3.6.6 Duración de batería

Mediante el cálculo de la selección de la batería se había establecido una batería de 5 amperios hora para un tiempo de 45 minutos la batería seleccionada en el transcurso de las pruebas no presentó deficiencia alguna y sobrepasó el tiempo de operación estimado y su duración real fue de 2:30 minutos aproximadamente esto se debe a que no se trabajó con la velocidad máxima del motor por lo que el consumo de corriente fue menor al previsto, los parámetros a considerar fueron los que se muestran en la Tabla 30.

TABLA 30. Resultados de la ejecución de la batería Li-Po

Parámetros	Resultados
Conexiones	Correctas
Voltaje de operación	Correcto
Ensamblaje	Correcto
Funcionamiento continuo	Si
Sobrecalentamientos	No

Fuente. Autoría propia

3.6.7 Calidad de transmisión y recepción de video

El transmisor y receptor de video se comunicó de una manera correcta, se mantuvo una línea de alcance de 100 metros pasados los 100 metros se distorsionó el video y luego se perdió para ello se grabó el recorrido del vehículo mediante el programa VLC el cual permitió observar en el instante de tiempo lo que transmitía la cámara por el transmisores y receptor de vídeo.

El receptor de vídeo no contaba con batería incluida por lo que fue necesario adquirir una batería Li-Po para su funcionamiento para el tiempo que se requería se realizó los respectivos cálculos utilizando la ecuación (1) y se optó por adquirir una batería Li-Po de 2 celdas 7.4 voltios 500 mA/h ya que su consumo es de 200mA, pero en el transcurso de funcionamiento se calentaba por ende consumía más amperaje.

3.7 Discusión

El vehículo terrestre no tripulado autónomo es capaz de dirigirse a las coordenadas ingresadas de manera autónoma ya que por medio de los sensores de distancia el vehículo detecta y esquiva con el fin de no quedarse bloqueado y así poder recorrer toda la ruta sin problemas.

Se realizó las pruebas del vehículo para verificar que cumpla con los objetivos, también que los materiales seleccionados sean accesibles para la fabricación de este tipo de prototipos y así poder cumplir los requerimientos mínimos de un vehículo terrestre no tripulado autónomo.

El vehículo puede transitar en superficies planas, inclinadas y semiplanas ya que los actuadores poseen el suficiente torque para movilizarse por estas áreas y a la vez esquivando obstáculos e ir transmitiendo lo que capta la cámara en el instante de tiempo con el fin de realizar búsquedas de personas extraviadas.

Las pruebas que se realizaron tuvieron buenos resultados entre la comunicación de la estación remota y el vehículo terrestre no tripulado autónomo. Se tuvo en cuenta las condiciones ambientales por donde transita el vehículo, ya que el módulo GPS junto al magnetómetro se encuentran ubicados en la parte superior del vehículo, con el fin de evitar interferencias con los componentes electrónicos, al igual que el transmisor de video disipa calor y en un ambiente caluroso esto puede influir para que no se transmita correctamente el video de la cámara, lo mismo sucede con los demás componentes que pueden sufrir sobrecalentamientos así acortando la vida útil de todos los componentes.

Conclusiones

- En la investigación que se realizó en este trabajo se concluyó, que la fabricación de vehículos terrestres no tripulados autónomos están diseñados para cumplir determinadas aplicaciones, entre ellas búsqueda y rescate, exploración espacial, espionaje, además se diferencian por su variedad de tipos de locomoción lo que ocasiona que su fabricación sea costosa, pero también puede ser implementada con tecnologías de menor gama y cumplir con los requerimientos mínimos de un vehículo terrestre no tripulado autónomo.
- Al ir implementando más componentes electrónicos al vehículo el diseño electrónico se hace más robusto por lo que el controlador utiliza más recursos y su rendimiento disminuye debido a que ejecuta varios procesos en cuestión de segundos, esto dependerá de la velocidad de procesamiento del controlador entre mayor sea la capacidad menor problemas presenta en la fluidez del vehículo.
- Durante las pruebas realizadas se concluyó que los módulos de comunicación, transmisión y recepción de video son un factor importante para tomar en cuenta ya que de esto dependerá que el vehículo envíe la información pertinente al usuario en un área de cobertura determinada, además se evidenció que las especificaciones de las hojas técnicas de los módulos varían con respecto a la cobertura que estos trabajan.
- En el momento que el vehículo se dirige a las coordenadas establecidas se evidenció que el recorrido sin obstáculos es más fluido, mientras que cuando se presentaba obstáculos el vehículo tendía a oscilar y generar más error en el recorrido, debido a que los sensores de distancia actuaban para evadir el objeto generando desviaciones al vehículo, pero sin impedir que el vehículo llegue al destino, pero con mayor alejamiento de la coordenada tomada.

Recomendaciones

- Para mejoras futuras relacionados a los vehículos terrestres no tripulados autónomos se podría diseñar y construir un sistema de suspensión para el chasis lo que proporcionará que el vehículo pueda transitar por superficies irregulares y los dispositivos colocados en el vehículo no sufran golpes así manteniendo su estática y correcto funcionamiento.
- El uso de comunicación de largo alcance conlleva a generar elevados costos, pero permite obtener mejores resultados con respecto a la comunicación o transmisión de video uno de los componentes que ayudan a generar más alcance son las antenas omnidireccionales que proporciona que los dispositivos de radiofrecuencia aumente su rango de alcance por lo que se podría añadir este tipo de antenas al módulo Xbee, y transmisores de vídeo.
- Para el módulo GPS podría ser reemplazado por un dispositivo de mayor gama para tener mediciones más precisas y que el vehículo no genere grandes errores de rumbo y se podría utilizar los controladores Pixhawk los cuales cuentan con una interfaz para este tipo de vehículos y juntamente con la incorporación de sensores de detección de objetos más sofisticados ayudaría a que el vehículo se desarrolle de una manera más fluida, rápida y precisa, pero con un mayor costo de inversión.

Referencias

- Aguilera, C. (11 de Agosto de 2019). *Nueva Modalidad*. Recuperado el 2 de Noviembre de 2019, de Vehículos Autónomos: <https://www.nuevamovilidad.com/historia-actualidad-y-futuro-de-los-coches-autonomos/#>
- Alkè. (2016). *Alke*. (Alke, Productor) Recuperado el 3 de Noviembre de 2019, de <https://www.alke.eu/es/ugv-vehiculos-autonomos>
- Alvarez, R. (23 de Mayo de 2019). *TecBolivia*. Recuperado el 25 de Diciembre de 2020, de <http://tecbolivia.com/index.php/articulos-y-tutoriales-microcontroladores/100-carro-robot-guiado-por-gps>
- Amazon. (1 de Enero de 2020). Recuperado el 28 de Abril de 2020, de <https://www.amazon.com>
- Barroso, R. (6 de Agosto de 2019). *Motores Brushless*. Recuperado el 29 de Diciembre de 2019, de <http://www.aerodelismofacil.com/Motores%20brushless.htm>
- Berry, A. (7 de Octubre de 2018). Terremoto. *Al menos 5.000 desaparecidos tras el terremoto en Indonesia*. Recuperado el 14 de Novimebre de 2019, de <https://www.elmundo.es/internacional/2018/10/07/5bb9c36e468aeb4d728b45e6.html>
- Chiriboga, F., Chafra, A., & Carlos, A. (2019). *Diseño y construcción de un prototipo de robot todo terreno, tele-operado y dotado de visión remota para el centro de investigación científica y tecnológica del ejército CICTE*. Tesis, Universidad de las Fuerzas Armadas ESPE, Sangolquí. Recuperado el 21 de Noviembre de 2019
- Contreras, P. (1 de octubre de 2018). *Nueva Modalidad*. Recuperado el 1 de noviembre de 2019, de <https://www.nuevamovilidad.com/somos-nueva-movilidad/>
- Cortes, A. (21 de Septiembre de 2019). Un minibús eléctrico sin conductor como guía turístico de la Ruta de los Volcanes. España. Recuperado el 26 de Diciembre de 2019, de https://elpais.com/tecnologia/2019/09/20/actualidad/1568985433_488958.html
- Cruz, F. S. (6 de Octubre de 2016). Robótica y agricultura: Nuevas tecnologías en el campo. *Robótica y agricultura: Vehículos autónomos, drones y más*. Recuperado el 26 de Diciembre de 2019, de <http://sistemaagricola.com.mx/blog/robots-en-la-agricultura/>
- D.I.Y Robotics. (2021). Recuperado el 25 de Febrero de 2021, de <http://diybot.co.za/product/arduino-due-board-compatible-with-arduino/>
- Dario. (6 de Junio de 2019). *Todo sobre drone*. Recuperado el 23 de Enero de 2020, de <https://dronewiki.net/drone-fpv>
- Delgado, V. (1 de junio de 2016). *ELDRONE.ES*. (Empresa Operadora de Drones habilitada por AESA) Recuperado el 1 de novimebre de 2019, de Historia de los drones: <http://eldrone.es/historia-de-los-drones/>
- Embention. (15 de enero de 2016). UGV, Aplicaciones y funciones para uso profesional. España. Recuperado el 23 de 12 de 2019, de <https://www.embention.com/es/news/ugv-aplicaciones-funciones-profesionales/>

- Enriquez, J. I. (2017). *Diseño e implementación de un vehículo terrestre no tripulado con sistemas de navegación autónoma*. Tesis, Universidad Nacional Autónoma de México, México. Recuperado el 23 de Noviembre de 2019
- Fernández, I. (2019). *clicachoches*. Recuperado el 3 de Noviembre de 2019, de Los 8 posibles problemas de los coches autónomos: <https://clicachoches.com/noticia/problemas-coches-autonomos>
- FpvMax. (3 de Agosto de 2017). Recuperado el 29 de Diciembre de 2019, de Variador electrónico (ESC): <http://fpvmax.com/2016/12/21/variador-electronico-esc-funciona/>
- FreeShopping. (24 de Mayo de 2019). *Associated CR12 Ford F-150*. Obtenido de <https://www.rcplanet.com/associated-cr12-ford-f-150-pick-up-rtr-black-asc40001/>
- Galbarro, H. (2021). *Ingemecánica*. Recuperado el 27 de Enero de 2021, de <https://ingemecanica.com/tutorialsemanal/tutorialn151.html>
- García, R., & García, A. (2016). *Prototipo virtual de un robot móvil multi-terreno para*. Artículo Científico, Universidad La Salle Oaxaca. Recuperado el 28 de Diciembre de 2019, de <file:///C:/Users/DELL/Downloads/Prototipovirtualdeunrobotmvmulti-terrenoparaaplicacionesdebsquedayrescate.pdf>
- García, V. (31 de Diciembre de 2017). *EPA*. Recuperado el 20 de Diciembre de 2020, de Electrónica práctica aplicada: <https://www.diarioelectronico hoy.com/blog/magnetometro-hmc5883#:~:text=EL%20HMC5883L%20es%20un%20sensor,al%20norte%20magnético%20e%20la>
- Germán, M. (2017). *Programar facil*. Recuperado el 9 de Enero de 2020, de <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>
- González, A. G. (2 de Diciembre de 2016). *PanamaHitek*. Recuperado el 29 de Diciembre de 2019, de <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>
- Google Maps. (2021). *Google maps*. Recuperado el 24 de Febrero de 2021, de https://www.google.com/maps/d/edit?hl=es&hl=es&mid=1UH3UHuiOMn2pnBBInzzSxald7aQh_EJb&ll=0.38001402772847004%2C-78.09869709908274&z=21
- Instituto Geofísico. (16 de Agosto de 2017). Sismos. *El terremoto de Ibarra del 16 de Agosto de 1868*. Recuperado el 14 de Noviembre de 2019, de <https://www.igeppn.edu.ec/servicios/noticias/1505-el-terremoto-de-ibarra-del-16-de-agosto-de-1868>
- Jecrespom. (13 de Noviembre de 2016). *Arduino open source community*. Recuperado el 9 de Enero de 2020, de <https://aprendiendoarduino.wordpress.com/tag/hc-05/>
- Leibson, S. (28 de Noviembre de 2018). *Digi-Key*. Recuperado el 3 de Mayo de 2020, de <https://www.digikey.com/es/articles/fundamentals-distance-measurement-gesture-recognition-tof-sensors>
- Llamas, L. (27 de Septiembre de 2016). Recuperado el 30 de Diciembre de 2019, de Localización con arduino y los módulos GPS NEO-6: <https://www.luisllamas.es/localizacion-gps-con-arduino-y-los-modulos-gps-neo-6/>

- Llamas, L. (6 de Diciembre de 2016). *Ingeniería, informática y diseño*. Recuperado el 23 de Enero de 2020, de Comunicación inalámbrica en arduino con módulos RF:
<https://www.luisllamas.es/comunicacion-inalambrica-en-arduino-con-modulos-rf-433mhz/>
- Llamas, L. (17 de Noviembre de 2017). *Ingeniería, informática y diseño*. Recuperado el 18 de Mayo de 2020, de <https://www.luisllamas.es/modelos-de-raspberry-pi/>
- Llamas, L. (17 de Noviembre de 2017). *Ingeniería, informática y diseño*. Recuperado el 18 de Mayo de 2020, de <https://www.luisllamas.es/modelos-de-raspberry-pi/>
- Lopez, M. I. (13 de Febrero de 2019). La Misión del Rover Opportunity de la NASA en Marte Llega a su Fin. Recuperado el 26 de Diciembre de 2019, de
<https://www.lanasa.net/misiones/marte/la-mision-del-rover-opportunity-de-la-nasa-en-marte-llega-su-fin>
- Macho, J. C. (2019). *Prometec*. Recuperado el 23 de Enero de 2020, de
<https://www.prometec.net/guia-fpv-para-multirrotores/>
- Magnetic declination. (2021). *Magnetic declination*. Recuperado el 15 de Febrero de 2021, de
<https://www.magnetic-declination.com>
- Meier, L. (3 de Marzo de 2020). *Pixhawk*. Recuperado el 18 de Mayo de 2020, de
<https://pixhawk.org/products/>
- Mera, A., & Ruano, D. (2019). *Implementacion de un robot de navegacion autonoma terrestre para evasión de obstáculos usando el dispositivo PIXHAWK*. Tesis, Universidad de las Américas, Quito. Recuperado el 1 de Diciembre de 2019
- Mercado Libre. (1 de Enero de 2020). Recuperado el 25 de Abril de 2020, de
<https://www.mercadolibre.com.ec>
- MilremAS. (1 de Octubre de 2019). Transporte de rescate multicapa. Europa. Recuperado el 13 de Noviembre de 2019, de <https://milremrobotics.com/product/multiscope-rescue-transport/>
- MilremRobotics. (1 de Octubre de 2019). El pelotón de infantería estonia despliega el UGV THEMIS de Milrem Robotics en patrulla por primera vez en Mali. Malí, Europa. Recuperado el 13 de Noviembre de 2019, de <https://milremrobotics.com/estonian-infantry-platoon-deploys-milrem-robotics-themis-ugv-on-patrol-for-the-first-time-in-mali/>
- Morales, E. (2017). *Sistema de captura de imágenes panorámicas mediante un vehículo aéreo no tripulado (drone) para el departamento de comunicación de la universidad técnica del norte*. Tesis, Universidad Técnica del Norte, Ibarra. Recuperado el 15 de Noviembre de 2019
- Muñoz, D. (23 de 10 de 2018). S-Moving analiza el futuro del vehículo inteligente, autónomo y no tripulado. *Interpresas.net*. Recuperado el 2 de Noviembre de 2019, de
<https://www.interempresas.net/Sector-Automocion/Articulos/227678-S-Moving-analiza-el-futuro-del-vehiculo-inteligente-autonomo-y-no-tripulado.html>
- Ortega, L., Darío, A., & Cárdenas, S. (2018). *Percepción 3D y planificación de trayectoria basado en una combinación de lildar 2D y Vision para vehiculos terrestres no tripulados*. Tesis, Universidad de las Fuerzas Armadas ESPE, Ingenieria en Electrónica, Automatización y Control, Sangolqui. Recuperado el 27 de 12 de 2019

- Pedraza, J. Á. (26 de Septiembre de 2019). Nuevas tecnologías llegan a los camiones de transporte de las minas de oro. Recuperado el 27 de Diciembre de 2019, de <https://oroinformacion.com/las-nuevas-tecnologias-llegan-a-los-camiones-de-las-minas-de-oro/>
- Proserquisa. (5 de Octubre de 2016). *Curso Arduino*. Recuperado el 9 de Enero de 2020, de <http://cursoarduino.proserquisa.com/2016/10/05/tutorial-13-modulo-sensor-ultrasonico-haz-una-alarma/>
- Roda, G. (18 de Mayo de 2020). *Arduino, Genuino, Raspberry Pi. Noticias y proyectos*. Recuperado el 19 de Mayo de 2020, de <https://descubrearduino.com/comparamos-varias-de-las-arduino-mas-utilizadas/>
- Rodriguez, A. (2017). *Sistema de vision en primera persona*. Tesis, Universidad Politecnica de Madrid, Madrid. Recuperado el 23 de Noviembre de 2019
- Rodriguez, A. (2017). *Sistema de Visión en Primera Persona*. Tesis, Universidad Politécnica de Madrid. Recuperado el 28 de Diciembre de 2019
- Romero, F. (8 de Novimebre de 2018). *motor.es*. (R. M. MURCIA, Productor) Recuperado el 1 de Nomvubre de 2019, de Motor Internet S.L.: <https://www.motor.es/noticias/patentes-conduccion-autonoma-crecimiento-tecnologias-201851709.html>
- Salazar, E. F. (2015). *Construcción e implementación de un pequeño prototipo de vehiculo electrico terrestre no tripulado para reconocimiento*. Tesis, Quito. Recuperado el 11 de Noviembre de 2019
- Samaniego, A. (2018). *Diseño y construcción de un cuatricóptero autónomo de vigilancia*. Tesis, Universidad Nacional de Chimborazo. Recuperado el 29 de Diciembre de 2019
- Sandria, J. (28 de Enero de 2018). *Robotica*. Recuperado el 9 de Enero de 2020, de <http://robotica.sandria.org/2018/01/conexion-punto-a-punto-de-modulos-xbee-s2c/#!/Energía/c/26897154/offset=0&sort=normal>
- Today. (5 de Marzo de 2017). *TheObjective*. Recuperado el 30 de Noviembre de 2020, de <https://theobjective.com/el-guardium-ii-es-un-vehiculo-terrestre-no-tripulado-2410f/>
- Torres, H. (7 de Diciembre de 2017). *Hetpro*. Recuperado el 28 de Diciembre de 2019, de <https://hetpro-store.com/TUTORIALES/arduino-vs-raspberry-pi/>
- Way, S. (14 de Septiembre de 2017). New x-2 platform and control system to be launched at dsei 2017. Reino Unido. Recuperado el 27 de Diciembre de 2019, de http://www.army-guide.com/eng/article/article_3125.html
- Yopez, V. (26 de Octubre de 2016). *Universidad Politécnica de valencia*. Recuperado el 25 de Noviembre de 2020, de poli(Blogs): <https://victoryepes.blogs.upv.es/2016/10/26/qfd-despliegue-de-la-funcion-de-calidad/>
- Yúbal. (3 de Agosto de 2018). *Xataka*. Recuperado el 29 de Diciembre de 2019, de Xataka Basics: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

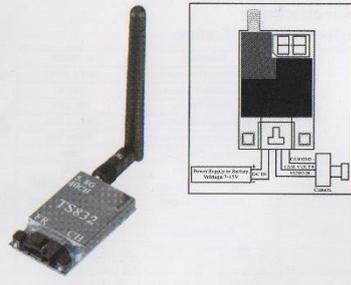
Anexo B: Hoja de datos de RC832 Y TS832

TS832 Instruction Manual

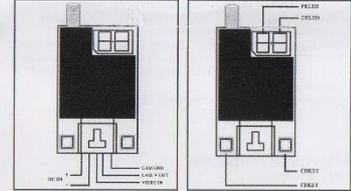
MODEL: TS832					
5.8G 600mW 40 Channels AV Transmitter Module					
Modulate		Wideband FM Modulate			
Video Format		NTSC/PAL			
Characteristics		Value		Units	
1	Output Impedance	50	---	Ohm	
2	Output Power	27	27.5	28	
				dBm	
Frequency Range		Channel Frequency			
FR1	CH1	---	5865	---	MHz
	CH2	---	5844	---	MHz
	CH3	---	5825	---	MHz
	CH4	---	5805	---	MHz
	CH5	---	5785	---	MHz
	CH6	---	5765	---	MHz
	CH7	---	5745	---	MHz
	CH8	---	5725	---	MHz
FR2	CH1	---	5733	---	MHz
	CH2	---	5753	---	MHz
	CH3	---	5771	---	MHz
	CH4	---	5790	---	MHz
	CH5	---	5809	---	MHz
	CH6	---	5828	---	MHz
	CH7	---	5847	---	MHz
	CH8	---	5866	---	MHz
FR3	CH1	---	5705	---	MHz
	CH2	---	5685	---	MHz
	CH3	---	5665	---	MHz
	CH4	---	5645	---	MHz
	CH5	---	5885	---	MHz
	CH6	---	5905	---	MHz
	CH7	---	5925	---	MHz
	CH8	---	5945	---	MHz
FR4	CH1	---	5740	---	MHz
	CH2	---	5760	---	MHz
	CH3	---	5780	---	MHz
	CH4	---	5800	---	MHz
	CH5	---	5820	---	MHz
	CH6	---	5840	---	MHz
	CH7	---	5860	---	MHz
	CH8	---	5880	---	MHz

3	FR5	CH1	---	5658	---	MHz
		CH2	---	5695	---	MHz
		CH3	---	5732	---	MHz
		CH4	---	5769	---	MHz
		CH5	---	5806	---	MHz
		CH6	---	5843	---	MHz
		CH7	---	5880	---	MHz
		CH8	---	5917	---	MHz
4	Operating Voltage	7	12	16	---	V
5	Supply Current	---	220	---	---	mA
6	Operating Temperature	-10	---	+	85	°C
7	Video Band Width	0	---	8.0	---	MHz
8	Audio Carrier Frequency	---	6.5	---	---	MHz
9	Video Input Level	0.8	1.0	1.2	---	Vp-p
10	Video Input Impedance	---	75	---	---	Ohm
11	Audio Input Level	0.5	---	2.0	---	Vp-p
12	Audio Input Impedance	---	10K	---	---	Ohm
13	Weight	---	22	---	---	g
14	Antenna Connector	SMA Connector				
15	Dimension(L×W×H)	54×32×10mm				

Picture: Application examples:



Connect: Channel Switch:



Frequency and channel frequency table:

	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FR1	5865M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
FR2	5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
FR3	5705M	5685M	5665M	5645M	5885M	5905M	5925M	5945M
FR4	5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M
FR5	5658M	5695M	5732M	5769M	5806M	5843M	5880M	5917M

RC832 Instruction Manual

Specification:

1. Technical Specification:

RC832 5.8G Wireless receiver	Sensitivity	< -90dBm
	Working Frequency	ISM 5.8GHz
	Available Channel	40 CH
	Power Supply	DC 12V
	Consumption Current	200mA Max
	Antenna Input Impedance	50ohm Typ
	Antenna Connector	SMA
	Antenna Gain	2dBi
	Video Output Level	1.0Vp-p Typ 75Ω
	Audio Output Level	1.0Vp-p Typ 10KΩ
	Audio Carrier	6.5 MHz
	Type Standard	NTSC/PAL
	Dimension(L×W×H)	80×65×15mm
Weight	85g	

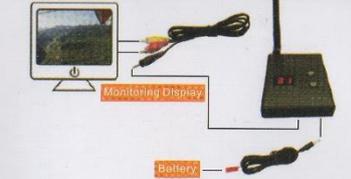
2. Features:

- 40 channels: 5 bands and all frequencies compatible.
- Two switching buttons: CH button for frequencies channels switching, FR button for frequencies bands switching.
- Two digital display: one for CH and the other for FR, real-time positioning which frequency band and which channel received.
- Power off memory: Replay the very last frequency band and channel.
- Independent video and audio signal outputs.

3. Operation Instruction:

A. Pin Description

B. Application example:



C. Channel and Digital tube display:
Power on, press the "CH" button to change channel, "CH" digital display will change synchronously, "CH" digital display changes cyclically from 1 to 8.

D. Frequency and Digital tube display:
Power on, press the "FR" button to change frequency band, "FR" digital display will change synchronously, "FR" digital display changes cyclically from 1 to 5.

FR 1 for band A, FR 2 for band B, FR 3 for band E, FR 4 for band F, FR 5 for band r.

E. Frequency and Channel Frequency table:

	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FR1	5865M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
FR2	5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
FR3	5705M	5685M	5665M	5645M	5885M	5905M	5925M	5945M
FR4	5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M
FR5	5658M	5695M	5732M	5769M	5806M	5843M	5880M	5917M

4. Standard Accessories:
- SMA Standard Type Dipole Antenna*2
 - AV Output Cable*1
 - Power cable for RC832 *1
 - Power cable for TS832 *1
 - User Manual*1

NOTICE:
Please connect every part according to the instruction, Please don't take down and reft this product.

RC832 & TS832

Product
Instruction manual

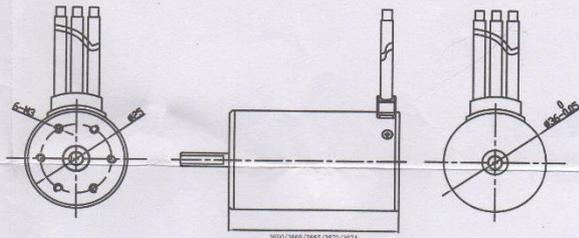
Anexo C: Hoja de datos de motor brushless 2300KV

36 SERIES BRUSHLESS MOTOR FOR CAR

■ Features:

- ★ 4 Poles 12 Slot Hi-torque Motor Design
- ★ CNC Machined 6061 T6 Billet Aluminum Heatsink Can
- ★ High Purity Copper Windings Maximizes Efficiency
- ★ High RPM ABEC5 Oversized Bearings
- ★ Universal fit, fits all 1/10 brushless cars-multi-mounting system for M3 screws
- ★ Precision balanced rotor, smoothness for best reliability and maximum RPM
- ★ Stator - Super Thin (0.35mm) Laminations
- ★ Removable/Replaceable Rotor
- ★ Precision Engineered for Maximum Energy Conversion

■ The product structure :



At the bottom of the motor

The side

At the top of the motor

■ Motor Specification

Product	Watts	Max voltage	Max Amps	Rotor Poles	IO	Resistance (Ω)	Kv (RPM/Volt)	Max RPM	Length	Diameters	Weight(g)	Shaft(mm)	Length of extend shaft	Connector
3650/2.5T	900	<8.4V	107A	4	5.2A	0.0043	5900	50000	50mm	36mm	180	∅ 3.175	15mm	4.0mm gold
3650/3T	900	<9.6V	94A	4	3.7A	0.0051	5200	50000	50mm	36mm	180	∅ 3.175	15mm	
3650/3.5T	900	<12V	75A	4	2.7A	0.0075	4300	50000	50mm	36mm	180	∅ 3.175	15mm	
3650/4T	900	<13V	69A	4	2.0A	0.0081	3900	50000	50mm	36mm	180	∅ 3.175	15mm	
3650/5T	900	<16V	56A	4	1.4A	0.0126	3100	50000	50mm	36mm	180	∅ 3.175	15mm	
3650/3.5Y	900	<22V	41A	4	1.2A	0.0182	2300	50000	50mm	36mm	180	∅ 3.175	15mm	
3650/4Y	900	<24V	38A	4	1.0A	0.0212	2050	50000	50mm	36mm	180	∅ 3.175	15mm	
3650/5Y	900	<30V	30A	4	0.7A	0.0318	1650	50000	50mm	36mm	180	∅ 3.175	15mm	

Product	Watts	Max voltage	Max Amps	Rotor Poles	IO	Resistance (Ω)	Kv (RPM/Volt)	Max RPM	Length	Diameters	Weight(g)	Shaft(mm)	Length of extend shaft	Connector
3660/3T	1200	<12V	100A	4	5.0A	0.0058	4300	50000	60mm	36mm	195	∅ 5	17mm	4.0mm gold
3660/3.5T	1200	<13V	92A	4	4.0A	0.0083	3800	50000	60mm	36mm	195	∅ 5	17mm	
3660/4T	1200	<15V	80A	4	3.5A	0.0095	3300	50000	60mm	36mm	195	∅ 5	17mm	
3660/5T	1200	<19V	63A	4	3.0A	0.0146	2600	50000	60mm	36mm	195	∅ 5	17mm	
3660/3Y	1200	<21V	57A	4	2.8A	0.0152	2350	50000	60mm	36mm	195	∅ 5	17mm	
3660/3.5Y	1200	<26V	46A	4	2.5A	0.0231	1950	50000	60mm	36mm	195	∅ 5	17mm	
3660/4Y	1200	<28V	43A	4	2.3A	0.0258	1750	50000	60mm	36mm	195	∅ 5	17mm	
3660/5Y	1200	<36V	33A	4	2.0A	0.0378	1400	50000	60mm	36mm	195	∅ 5	17mm	

Product	Watts	Max voltage	Max Amps	Rotor Poles	IO	Resistance (Ω)	Kv (RPM/Volt)	Max RPM	Length	Diameters	Weight(g)	Shaft(mm)	Length of extend shaft	Connector
3665/2Y	1400	<16.8V	80A	4	3.8A	0.0061	3100	50000	65mm	36mm	245	∅ 5	17mm	4.0mm gold
3665/2.5Y	1400	<16.8V	75A	4	2.9A	0.0116	2600	50000	65mm	36mm	245	∅ 5	17mm	
3665/3Y	1400	<16.8V	60A	4	1.9A	0.0162	2100	50000	65mm	36mm	245	∅ 5	17mm	
3665/3.5Y	1400	<16.8V	52A	4	1.6A	0.0235	1800	50000	65mm	36mm	245	∅ 5	17mm	
3665/4.0Y	1400	<16.8V	45A	4	1.2A	0.0266	1550	50000	65mm	36mm	245	∅ 5	17mm	

Product	Watts	Max voltage	Max Amps	Rotor Poles	IO	Resistance (Ω)	Kv (RPM/Volt)	Max RPM	Length	Diameters	Weight(g)	Shaft(mm)	Length of extend shaft	Connector
3670/2Y	1600	<18.5V	86A	4	4.2A	0.0069	2650	50000	70mm	36mm	275	∅ 5	17mm	4.0mm gold
3670/2.5Y	1600	<23V	70A	4	3.0A	0.0121	2150	50000	70mm	36mm	275	∅ 5	17mm	
3670/3Y	1600	<26V	60A	4	2.4A	0.0165	1900	50000	70mm	36mm	275	∅ 5	17mm	
3670/4Y	1600	<37V	43A	4	1.5A	0.0281	1350	50000	70mm	36mm	275	∅ 5	17mm	
3670/5Y	1600	<42V	38A	4	1.0A	0.0386	1100	50000	70mm	36mm	275	∅ 5	17mm	

Product	Watts	Max voltage	Max Amps	Rotor Poles	IO	Resistance (Ω)	Kv (RPM/Volt)	Max RPM	Length	Diameters	Weight(g)	Shaft(mm)	Length of extend shaft	Connector
3674/2Y	1800	<22.2V	80A	4	4.0A	0.0076	2250	50000	74mm	36mm	305	∅ 5	17mm	4.0mm gold
3674/2.5Y	1800	<26V	70A	4	3.1A	0.0132	1900	50000	74mm	36mm	305	∅ 5	17mm	
3674/3Y	1800	<32V	56A	4	2.6A	0.0161	1580	50000	74mm	36mm	305	∅ 5	17mm	
3674/3.5Y	1800	<37V	50A	4	2.1A	0.0258	1350	50000	74mm	36mm	305	∅ 5	17mm	
3674/4Y	1800	<42V	43A	4	1.6A	0.0298	1150	50000	74mm	36mm	305	∅ 5	17mm	

Anexo D: Hoja de datos de modulo GPS NEO 6m



NEO-6 - Data Sheet

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	2.7	3.0	3.6	V	
		NEO-6P/V/T					
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
		Icc Acquisition	All	47 ¹⁹		mA	
		Icc Tracking (Max Performance mode)	NEO-6G/Q/T	40 ²⁰		mA	
Average supply current ¹⁸	Icc Tracking (Eco mode)	NEO-6M/P/V	39 ²⁰		mA		VCC = 3.0 V ¹⁶ / 1.8 V ¹⁷
		NEO-6G/Q/T	38 ²⁰		mA		
	Icc Tracking (Power Save mode / 1 Hz)	NEO-6M/P/V	37 ²⁰		mA		
		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

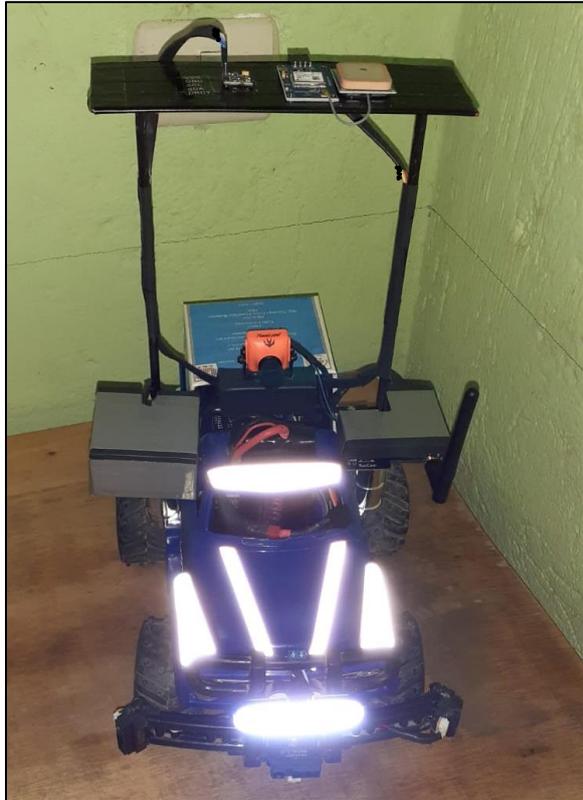
¹⁷ NEO-6G

¹⁸ Use this figure to determine required battery capacity.

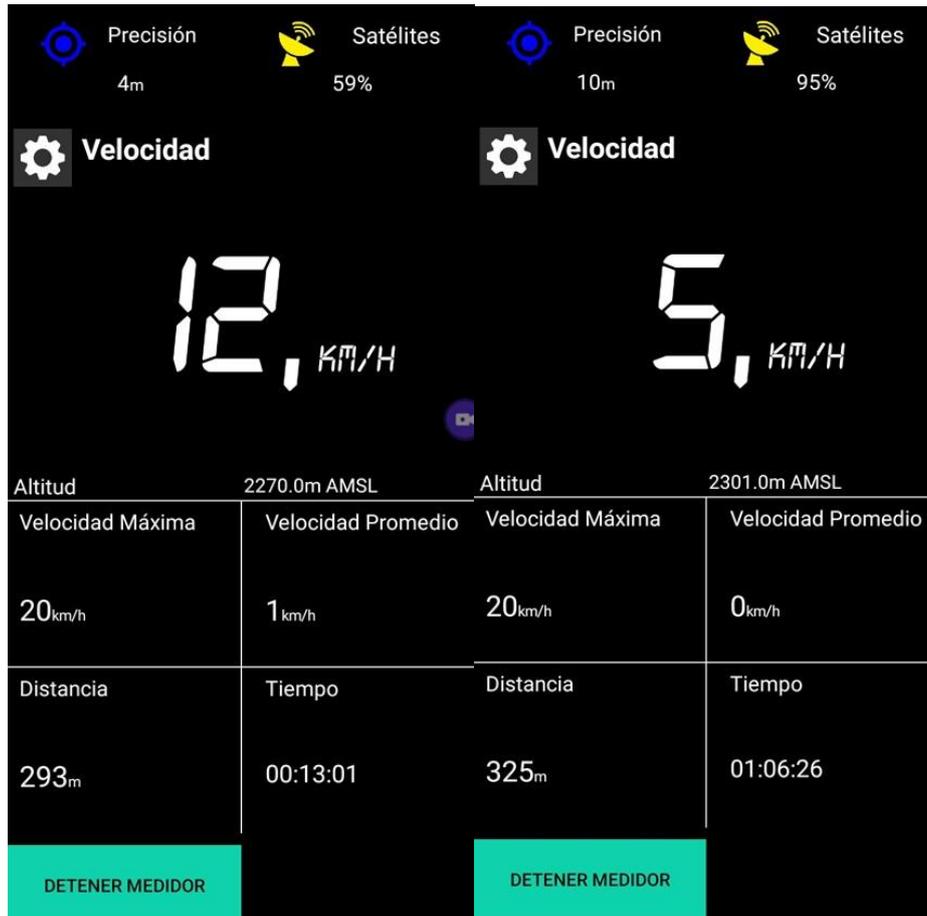
¹⁹ >8 SVs in view, CNo >40 dBHz, current average of 30 sec after cold start.

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.

Anexo E: Prototipo de vehículo terrestre no tripulado autónomo



Anexo F: Mediciones de velocidad



Anexo G: Manual de Usuario

INTRODUCCIÓN

La implementación de cualquier tipo de prototipo conlleva la realización de un manual de usuario con el fin de detallar toda la configuración de los dispositivos utilizados para que cualquier usuario que tenga acceso al prototipo tenga la facilidad de controlarlo y conozca cuál es el funcionamiento del prototipo al igual que su modo de ejecución. Para ello se detalla desde la configuración inicial que se realizó, hasta la ejecución y funcionamiento del prototipo en la que se interactúa con el puerto serial de la programación de la estación remota, el programa VLC media player que permite observar lo captado por la cámara y el vehículo.

1. Configuración de los Xbee

El programa XCTU permite configurar a los módulos Xbee en 3 diferentes tipos cómo es el modo Coordinador, Router y dispositivos finales para ello se utilizó la configuración de coordinador, el cual estará ubicado en la estación remota y un receptor el cual se ubicará en el vehículo. La configuración que se utiliza es punto a punto, en la cual la comunicación es bidireccional, la comunicación de estos dispositivos se la puede realizar en modo transparente (AT) o en modo comando (API), que son mensajes por medio de tramas en nuestro caso se utiliza la configuración en modo AT para ello se configura de la siguiente forma.

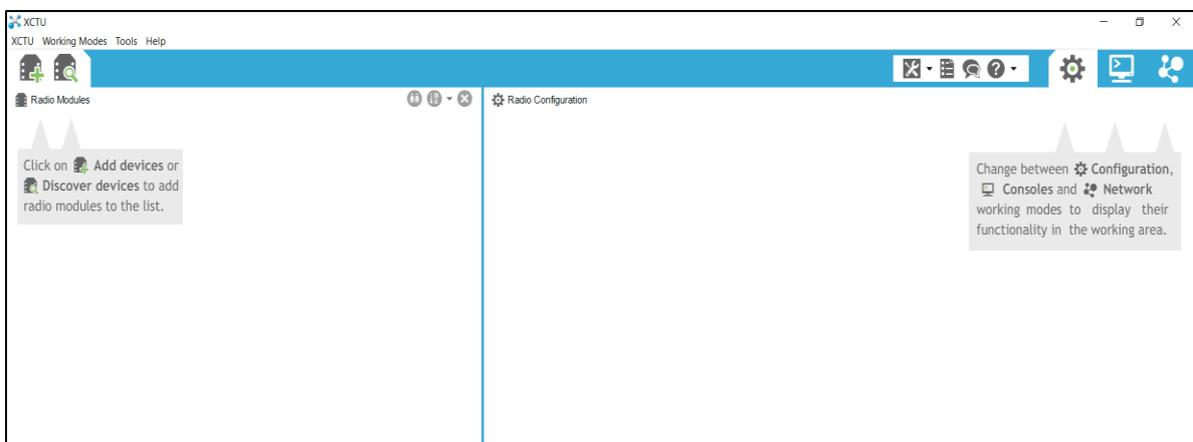


Imagen 1. Programa XCTU
Fuente. Tomado de XCTU

1. Se ingresa al programa XCTU, dar clic en Módulos.

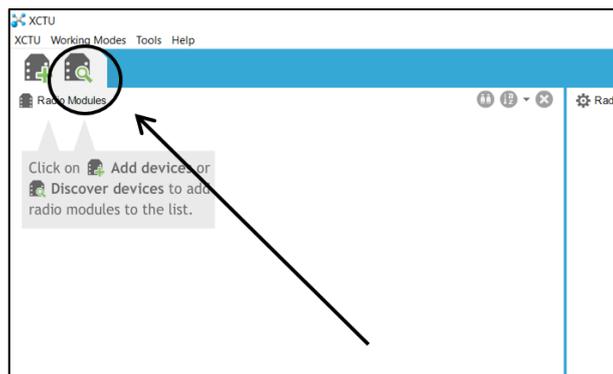


Imagen 2. Programa XCTU
Fuente. Tomado de XCTU

2. Seleccionar el puerto y dar clic en Next.

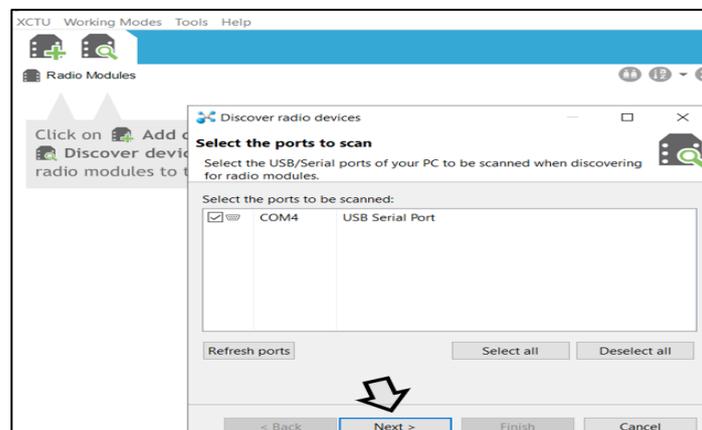


Imagen 3. Programa XCTU
Fuente. Tomado de XCTU

3. Marcar las casillas correspondientes y dar en clic en Finish.

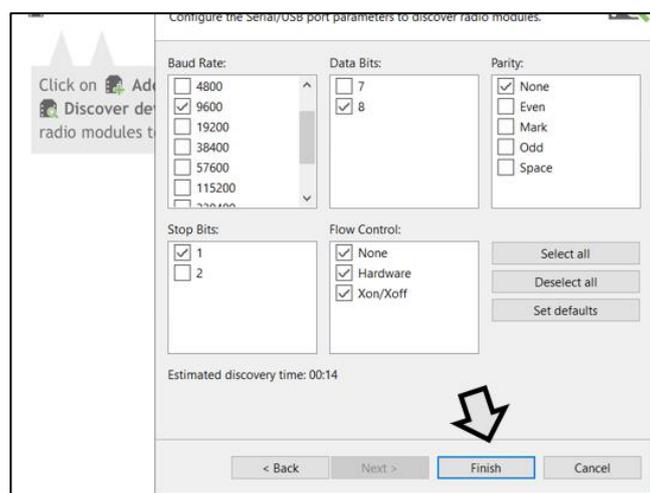


Imagen 4. Programa XCTU
Fuente. Tomado de XCTU

4. Brevemente el programa empezará a buscar el dispositivo Xbee, dar clic en Add selected devices.

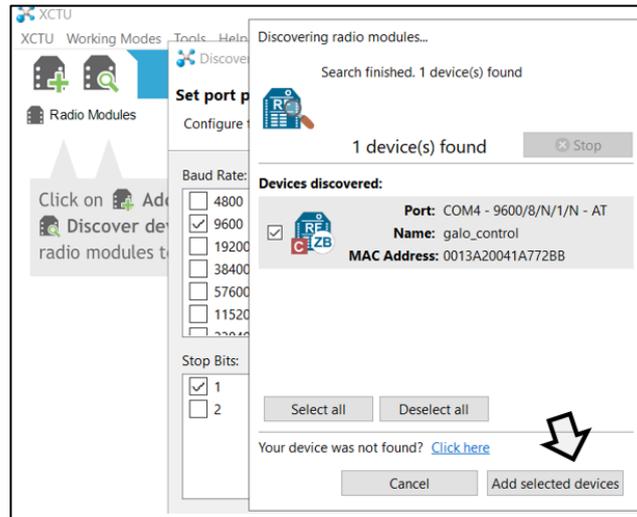


Imagen 5. Programa XCTU
Fuente. Tomado de XCTU

5. Luego se selecciona el dispositivo y se cargará la configuración.

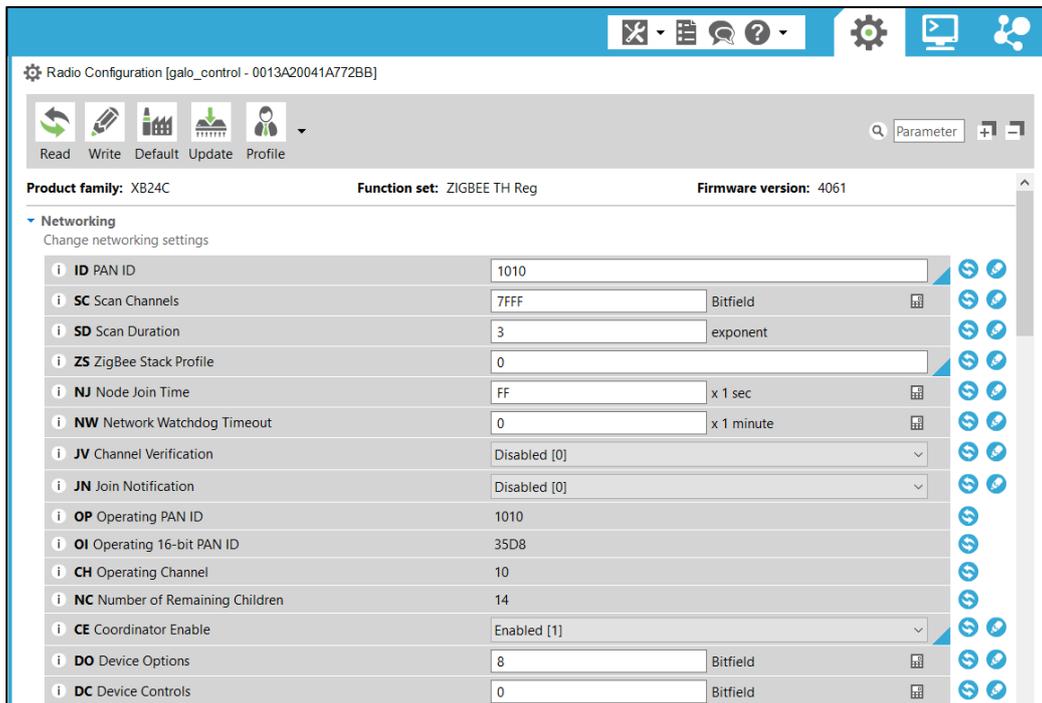


Imagen 6. Programa XCTU
Fuente. Tomado de XCTU

El entorno del programa XCTU no es muy complejo y para la configuración AT los únicos campos que se configura son los señalados en la Tabla 1.

TABLA 1. Configuración de Xbee coordinador y Router

Coordinador	Router
Cambiar PAN ID = 1010	Cambiar PAN ID = 1010
Habilitar coordinador	Habilitar verificación de canal
DH = 13A200	DH = 13A200
DL = 41A772BF	DL = 41A772BB

Fuente. Autoría propia

Con este tipo de configuración los Xbee se pueden comunicar bidireccionalmente el Coordinador puede enviar y recibir información, el Router de igual manera recibir y enviar información al coordinador.

- Una vez configurado, para verificar que el Router comunique con el coordinador y viceversa, dar clic en la imagen del computador y Open.



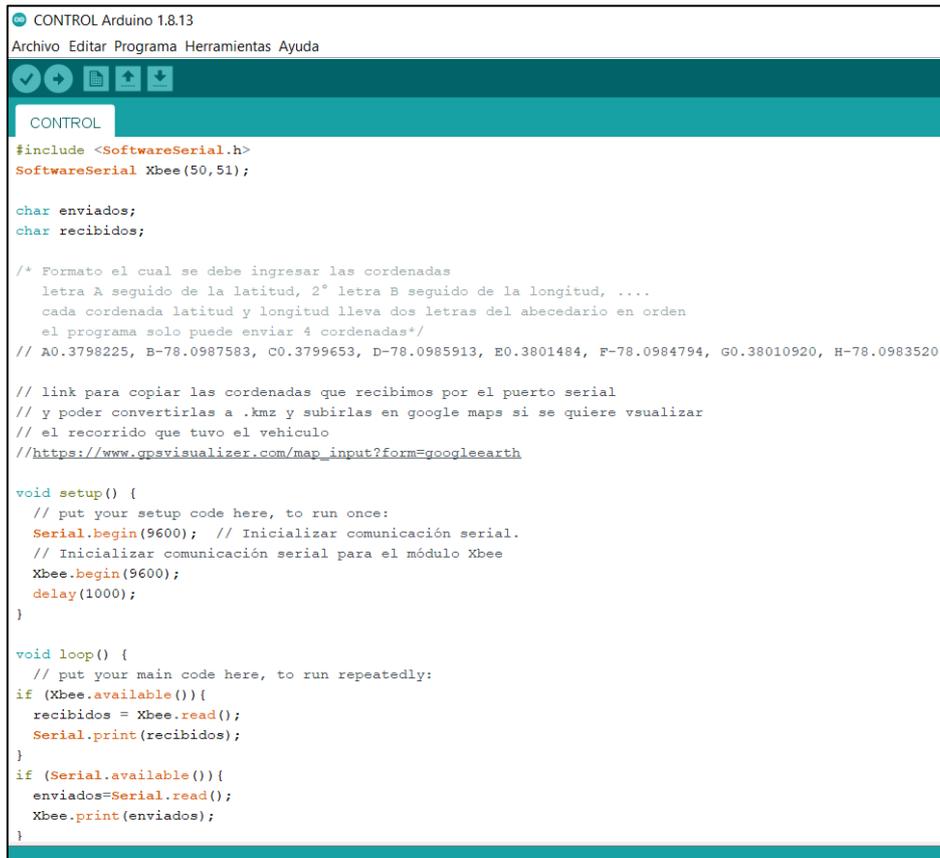
Imagen 7. Programa XCTU
Fuente. Tomado de XCTU

Una vez aquí todo lo que se escriba en “Console log” se transmitirá al otro Xbee Router.

2. Control del vehículo

Para realizar la disposición del vehículo a donde se necesite que se dirija, se usa la programación de control realizada en Arduino la cual estará cargada al Arduino Mega, que será el que reciba los datos por medio del Xbee coordinador y las transmita por el puerto serie.

Para esto se detalla los pasos a seguir para ingresar las coordenadas y enviar la ejecución de la ruta asignada.



```
CONTROL Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

CONTROL

#include <SoftwareSerial.h>
SoftwareSerial Xbee(50, 51);

char enviados;
char recibidos;

/* Formato el cual se debe ingresar las cordenadas
  letra A seguido de la latitud, 2° letra B seguido de la longitud, ...
  cada cordenada latitud y longitud lleva dos letras del abecedario en orden
  el programa solo puede enviar 4 cordenadas*/
// A0.3798225, B-78.0987583, C0.3799653, D-78.0985913, E0.3801484, F-78.0984794, G0.38010920, H-78.0983520

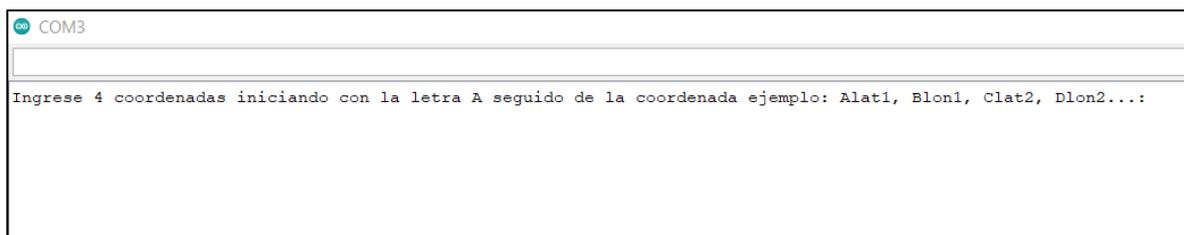
// link para copiar las cordenadas que recibimos por el puerto serial
// y poder convertirlas a .kmz y subirlas en google maps si se quiere vsualizar
// el recorrido que tuvo el vehiculo
//https://www.gpsvisualizer.com/map_input?form=googleearth

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Inicializar comunicación serial.
  // Inicializar comunicación serial para el módulo Xbee
  Xbee.begin(9600);
  delay(1000);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (Xbee.available()){
    recibidos = Xbee.read();
    Serial.print(recibidos);
  }
  if (Serial.available()){
    enviados=Serial.read();
    Xbee.print(enviados);
  }
}
```

Imagen 8. Programa Arduino
Fuente. Tomado de Arduino

1. Encender el vehículo para que se energice el Xbee Router, al igual que dar energía al Arduino Mega para que se encienda el Xbee Coordinador, en caso de haberse borrado la programación del control y del vehículo, cargar la programación de CONTROL en Arduino Mega y en Arduino Due la programación de TESIS.
2. Una vez conectados se espera a que se emparejen los módulos Xbee y se reciba un mensaje por el puerto serial de CONTROL como se observa en la Imagen 9.



```
COM3

Ingrese 4 coordenadas iniciando con la letra A seguido de la coordenada ejemplo: Alat1, Blon1, Clat2, Dlon2...:
```

Imagen 9. Puerto Serial de Arduino
Fuente. Tomado de Arduino

Formato el cual se debe ingresar las coordenadas letra A seguido de la latitud, 2° letra B seguido de la longitud, etc. Cada coordenada latitud y longitud lleva una letra del abecedario en orden alfabético, el programa solo puede recibir 4 coordenadas, por ejemplo:

'A'Latitud1,'B'longitud1, 'C'Latitud2,'D'longitud2, 'E'Latitud3,'F'longitud3, 'G'Latitud4,'H'longitud4
A0.3798225,B-78.0987583,C0.3799653,D-78.0985913,E0.3801484,F-78.0984794,G0.38010920, H-78.0983520

3. Ingresar las coordenadas como se menciona en el ejemplo anterior, damos clic en enviar y se recibe como respuesta “CARGANDO”

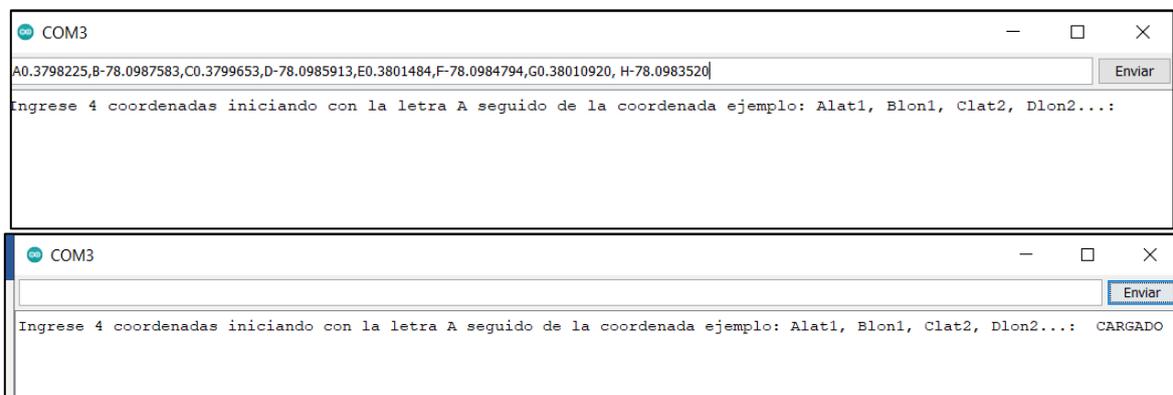


Imagen 10. Puerto Serial de Arduino
Fuente. Tomado de Arduino

4. Una vez realizado eso se ingresa el “1” sin comillas y dar clic en enviar para que el vehículo empiece a recorrer la ruta enviada y por el puerto serial se transmitirá un mensaje de “INICIAR” con las coordenadas por donde el vehículo está transitando.

3. Toma de ubicaciones de latitud y longitud en google maps

Para tomar lecturas de puntos GPS se puede ingresar en la aplicación de google maps o en la página de google maps. Primero se debe pulsar en la ubicación y enseguida google maps brinda información de ese punto seleccionado en el mapa y la información es latitud y longitud en la siguiente imagen se indica donde verificar los datos mencionados.

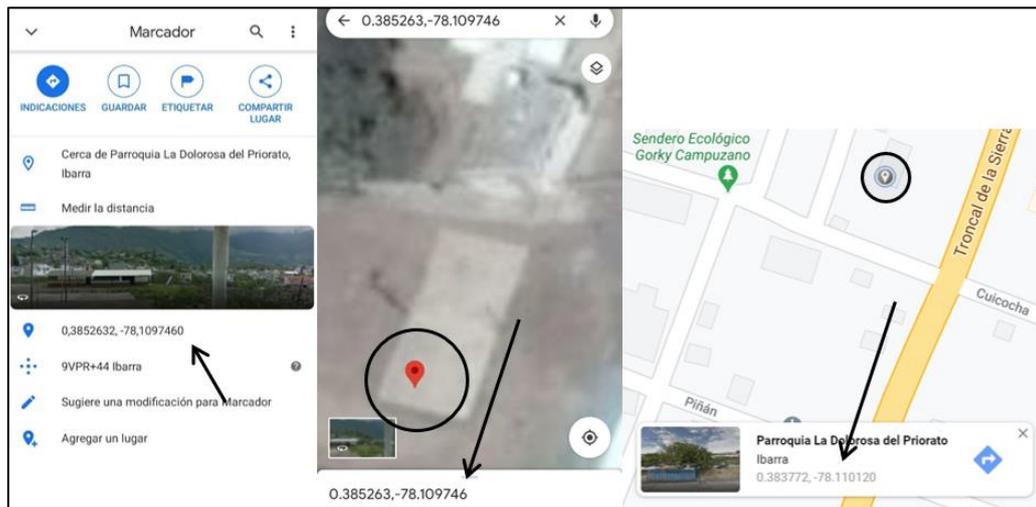


Imagen 13. Toma de puntos GPS
Fuente. Tomado de Google Maps

4. Conexión con la transmisión de video

Para observa la transmisión de la cámara se hace uso del reproductor Media player VLC el cual se le configura de la siguiente forma para poder ver lo que observa la cámara.

1. Abrir el programa Media Player VLC e ir a Medio en la opción abrir dispositivo de captura.

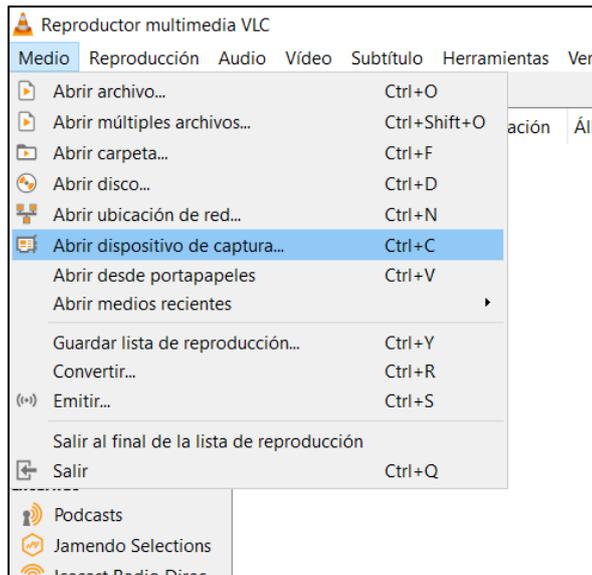


Imagen 14. Configuración 1 VLC media player
Fuente. Tomado de VLC media player

2. Luego seleccionar AV TO USB2.0 en Nombre del dispositivo de video.

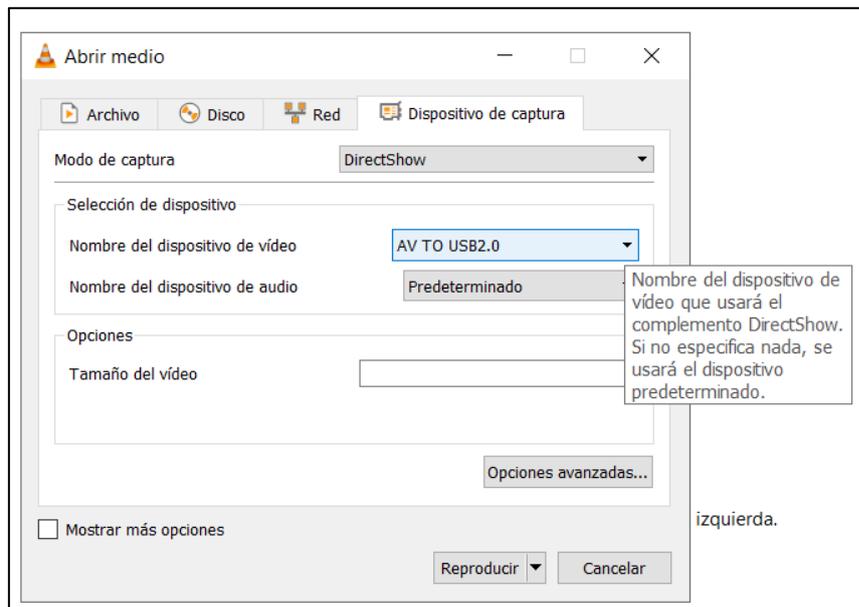


Imagen 15. Configuración 2 VLC media player
Fuente. Tomado de VLC media player

3. Dar clic en opciones avanzadas, marcar el casillero propiedades de dispositivo y aceptar, por último, dar clic en producir.

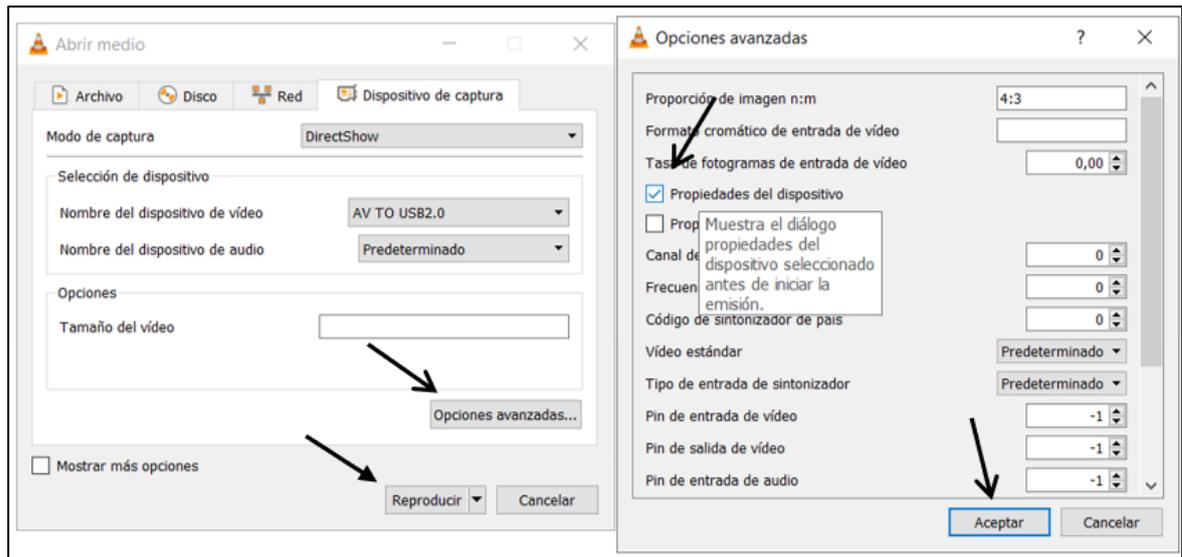


Imagen 16. Configuración 3 VLC media player
Fuente. Tomado de VLC media player

4. Se abren varias ventanas y dar clic en aceptar todo.

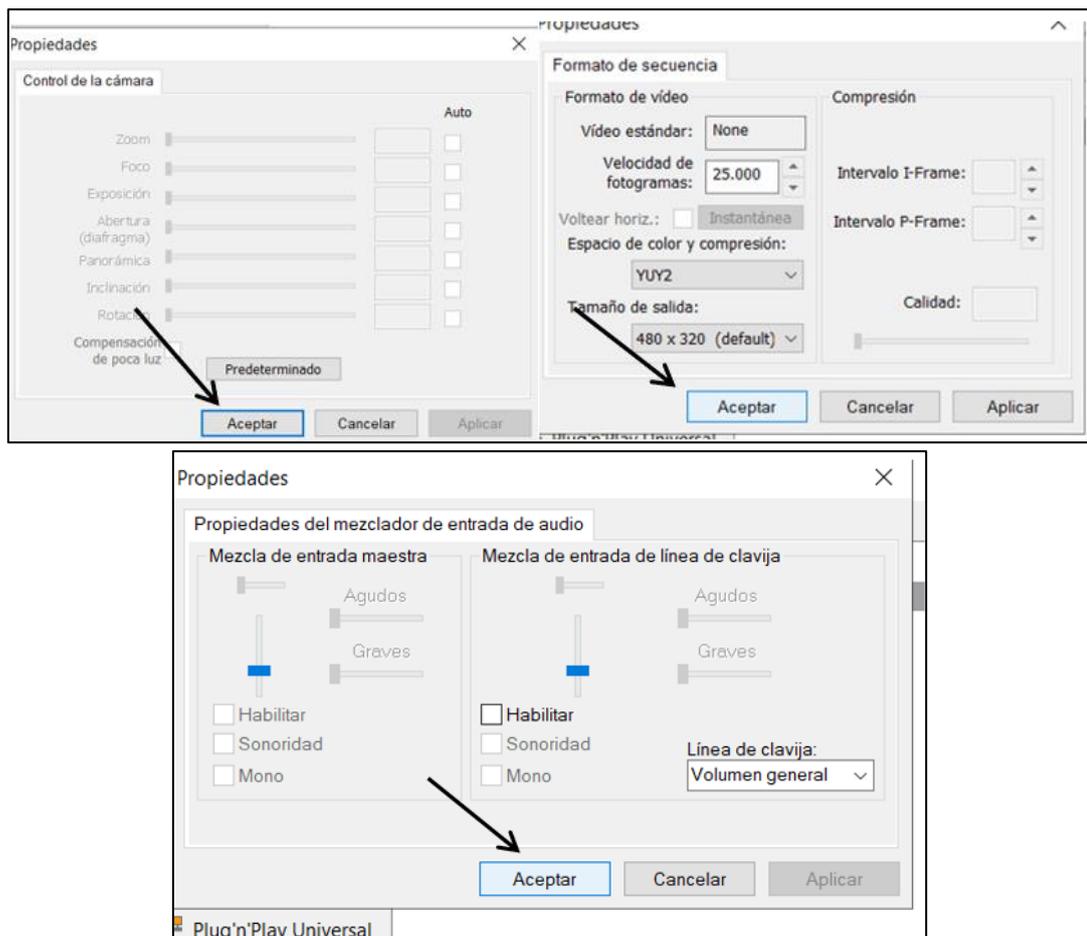


Imagen 17. Configuración 4 VLC media player
Fuente. Tomado de VLC media player



Imagen 18. Transmisión de la Cámara VLC media player
Fuente. Tomado de VLC media player

5. Verificar que tanto el transmisor como receptor deben estar encendidos antes de configurar el reproductor VLC para no tener inconvenientes. Si no se llega a transmitir bien el video ir probando frecuencia y canal ubicando en ambos dispositivos la misma frecuencia y canal.

5. Problemas y soluciones

1. Cuando los Xbee no se comunican entre sí, se debe apagar y encender los dos Arduinos para que se reinicie el emparejamiento entre ellos, seguido aplastar el botón Reset del Arduino Due hasta que envíe un mensaje al Arduino Mega mediante el puerto serie por medio de los Xbee.
2. Si el módulo GPS no tiene comunicación y no emite las coordenadas por el puerto serie, verificar que el led del módulo se encuentre parpadeando, caso contrario esperar unos minutos a que se empareje con los satélites para empezar a trabajar.
3. Si existe mala recepción de video probar cambiando a la misma frecuencia y el canal en el transmisor y receptor de video, luego verificar si las antenas se encuentran bien sujetas.

Anexo H: Manual Técnico

INTRODUCCIÓN

La implementación de cualquier tipo de prototipo conlleva la realización de un manual técnico con el fin de detallar toda la conexión de los dispositivos utilizados para que cualquier usuario que tenga acceso al prototipo tenga la facilidad de conectarlo y programarlo. Para ello se detalla las conexiones del vehículo, montaje y programación del prototipo.

1. Conexiones de actuadores

Para las conexiones de los actuadores que está montado en la parte del chasis del vehículo el diagrama de conexión es el siguiente:

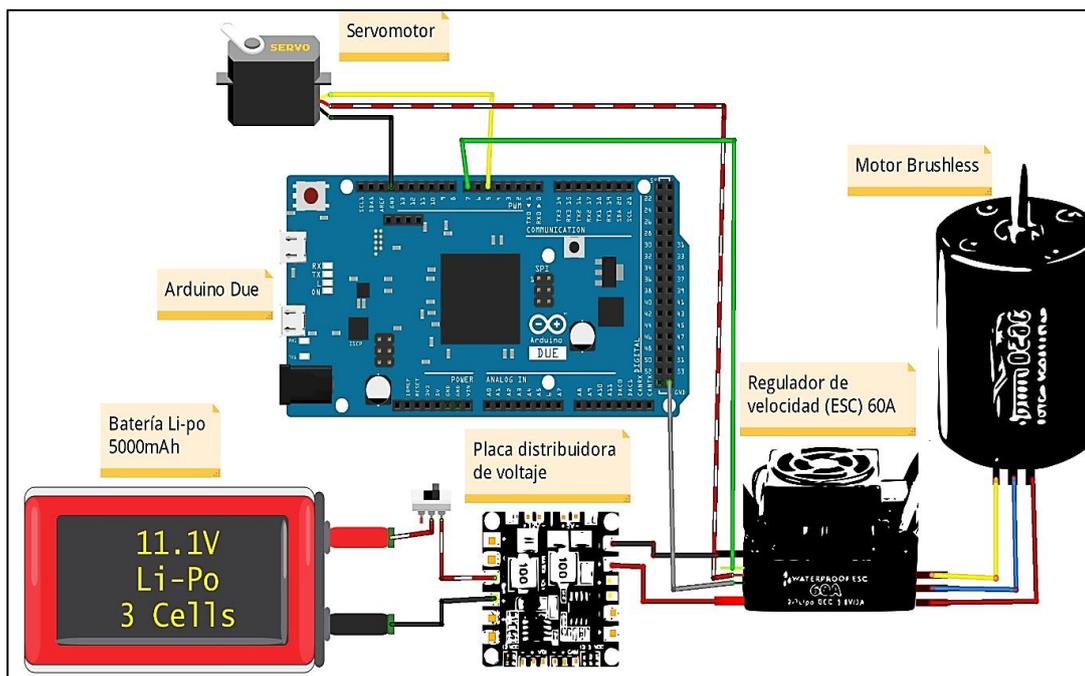


Imagen 1: Conexiones de actuadores
Fuente. Autoría propia

Como se observa en la Imagen 1 la batería Li-Po junto con su placa distribuidora de voltaje son las encargadas de distribuir todos los voltajes necesarios a los distintos componentes electrónicos. Algunos pines de la placa de distribución de voltaje quedan libres para poder acoplar más componentes en un futuro.

Para el caso de la conexión entre el motor brushless y el regulador de velocidad el color de los cables es negro por lo que se colocó señalización en los cables para indicar con cual se conecta

TABLA 1. Conexiones de pines de los actuadores

Servomotor	Conexiones
VCC	5V (Regulador de velocidad ESC)
GND	GND (Arduino Due)
Señal	Pin 5 (Arduino Due)
Motor brushless	
Load -	Load - (Regulador de velocidad ESC)
Load señal	Load señal (Regulador de velocidad ESC)
Load +	Load + (Regulador de velocidad ESC)
Regulador de velocidad (ESC)	
GND	GND (Arduino Due)
Load señal	Pin 7 (Arduino Due)
5V	VCC (Servomotor)
12V	+ (Placa distribuidora de voltaje)
GND	- (Placa distribuidora de voltaje)

Fuente. Autoría propia

2. Conexión de sensores Sharp

Para las conexiones de los sensores se unió todos los negativos y todos los positivos de voltaje para evitar más conexiones y reducir cables para ejemplificar como estaría la conexión solo de los sensores se representa de la siguiente manera en la Imagen 2.

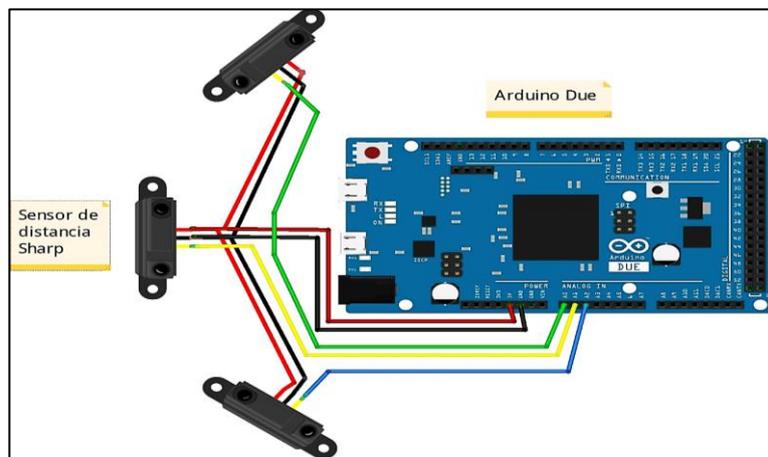


Imagen 2. Conexiones de los sensores de distancia Sharp

Fuente. Autoría propia

Queda más pines del Arduino Due para añadir más sensores al vehículo o algún componente que necesite de lecturas analógicas.

TABLA 2. Conexiones de los sensores Sharp

Sensores Sharp	Conexiones
+	5V (Arduino Due)
-	GND (Arduino Due)
señal	A1, A2, A3 (Arduino Due)

Fuente. Autoría propia

3. Conexión de módulos Xbee S2c

Para las conexiones del módulo Xbee se lo representa en el siguiente diagrama de conexión para ejemplificar de manera una manera más clara.

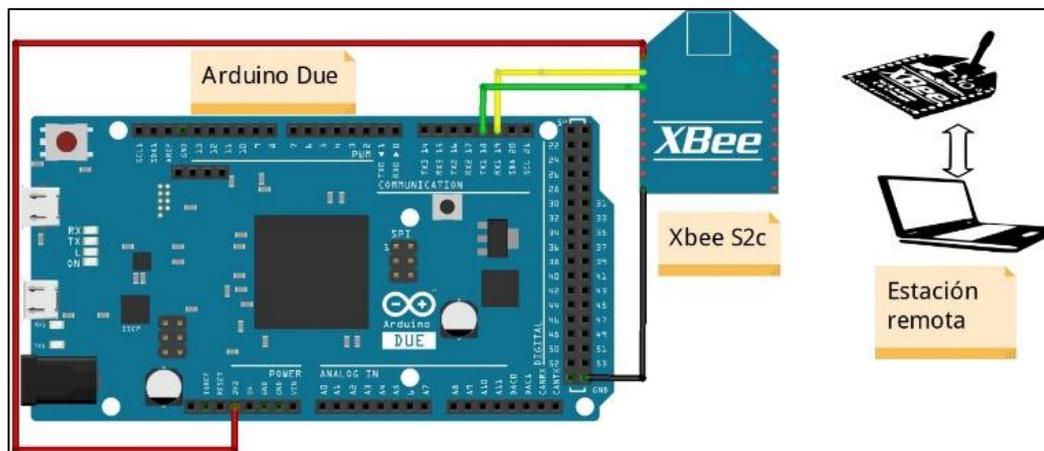


Imagen 3. Conexiones de los Xbee S2c

Fuente. Autoría propia

TABLA3. Conexiones de Xbee S2c

Xbee S2c	Conexiones
1 VCC	3.3V (Arduino Due)
2 Data out (TX)	Pin RX1 (Arduino Due)
3 Data in (RX)	Pin TX1 (Arduino Due)
10 GND	GND (Arduino Due)

Fuente. Autoría propia

4. Conexiones del módulo de sistema de posicionamiento global y magnetómetro HMC5883L.

Para ejemplificar las conexiones se representa el diagrama separado de los demás componentes para comprender y observar de mejor manera el diseño electrónico.

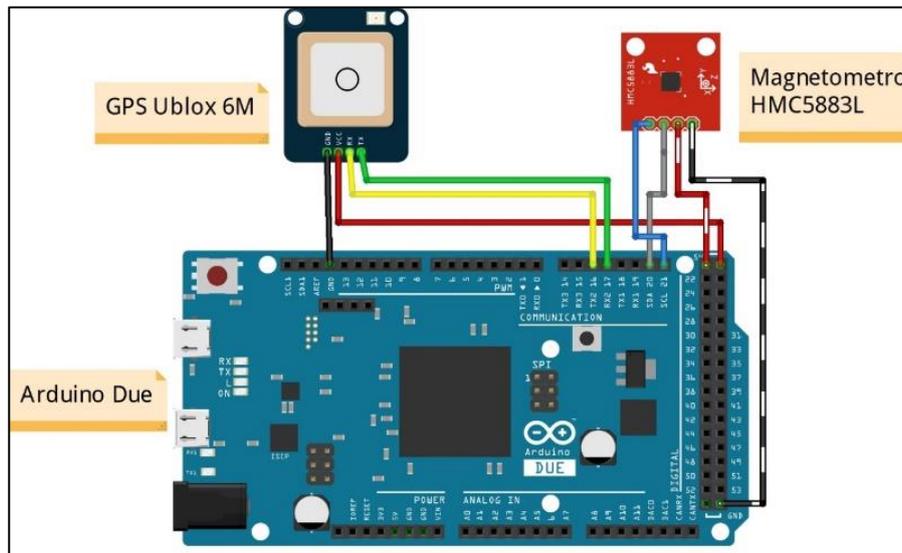


Imagen 4. Conexiones del módulo GPS y magnetómetro HMC5883L
Fuente. Autoría propia

TABLA 4. Conexiones del módulo GPS y magnetómetro

Módulo GPS Ublox 6m	Conexiones
VCC	5V (Arduino Due)
GND	GND (Arduino Due)
TX	Pin RX2 (Arduino Due)
RX	Pin TX2 (Arduino Due)
Magnetómetro HMC5883L	
VCC	5V (Arduino Due)
GND	GND (Arduino Due)
SDA	SDA (Arduino Due)
SCL	SCL (Arduino Due)

Fuente. Autoría propia

5. Conexiones de la cámara, transmisor y receptor de video TS832 y RC832

Para poder visualizar de manera más clara las conexiones se realiza el siguiente diagrama como se muestra en la Imagen 5 con su respectiva conexión en la Tabla 5.

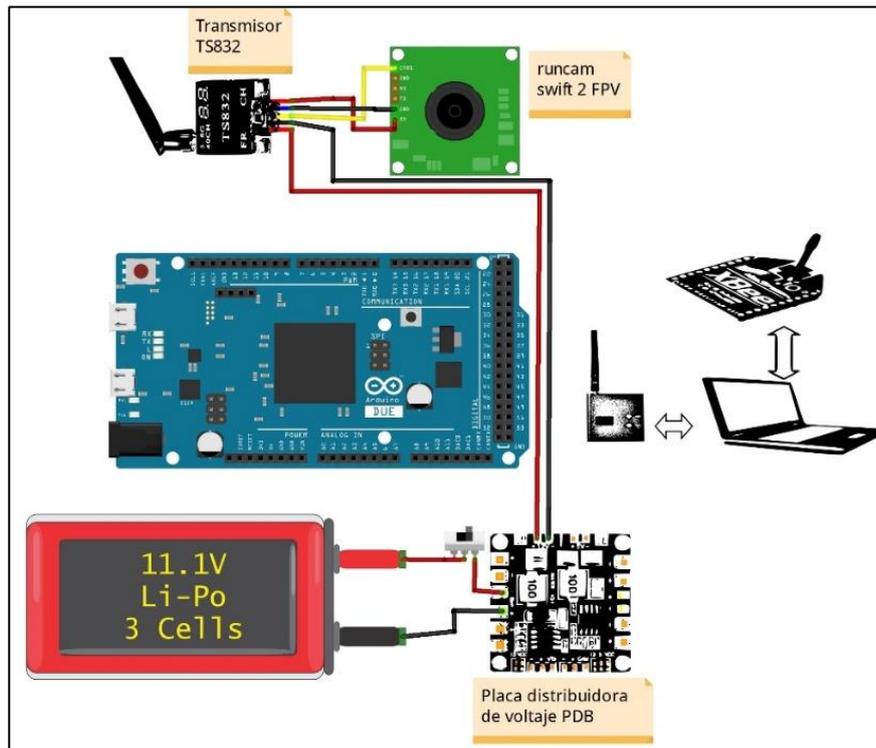


Imagen 5. Conexiones del transmisor de video TS832

Fuente. Autoría propia

TABLA 5. Conexiones de transmisor y receptor de video

Receptor RC832	Conexiones
DC IN	12V (Placa distribuidora de voltaje)
GND	GND (Placa distribuidora de voltaje)
VIDEO IN	SEÑAL (Cámara)
GND	GND (Cámara)
Voltaje OUT	VCC (Cámara)
Transmisor TS832	Portátil

Fuente. Autoría propia

6. Programación del código del vehículo terrestre no tripulado autónomo

Para la programación se distribuye en 7 partes, la cual se explica la programación como tal, mas no las variables ingresadas. Se conforma de la siguiente manera: programación de los sensores Sharp, magnetómetro, módulo GPS, filtros de los datos del GPS, cálculos del vector dirección, actuadores y control de la base remota.

a. Programación de sensores Sharp

Para realizar la programación de los sensores Sharp, se realizó 20 lecturas de los datos tomados del sensor de distancia para sacar un promedio, con el fin de reducir los picos que se producían al tomar las lecturas y trabajar con un valor más preciso.

```
float dist_izq(int n) {
    long suma = 0;
    for (int i = 0; i < n; i++) {
        suma = suma + analogRead (Sensor_izquierda);}
    float prom = suma / n;
    float adc = prom * 0.0048875855; // 5v/1023 =0.0048875855
    float dist_izq_cm = 13 * pow(adc, -1);
    return (dist_izq_cm);}
```

b. Programación del Magnetometro

Para la programación del magnetómetro se utilizó las librerías “Wire, I2Cdev, HMC5883L”, para lo cual este tomaba el norte magnético el cual situaba a la variable mx para luego sacar el arco tangente de mx/my y transformar ese valor a grados ya que está en radianes, a qué se utiliza la declinación magnética que existe en cada zona del mundo para ello se consulta en la página y se verifica que la declinación magnética de Ibarra es -4° 25'. (Magnetic declination, 2021)

```
void Lectura_magnetometro(void) {
    // Obtenga los componentes del campo magnético en X, Y, Z
    magnetometro.getHeading(&mx, &my, &mz);
    //Calculo del ángulo del eje X norte
    angulo_magnetometro = atan2(my, mx);
    angulo_magnetometro = angulo_magnetometro * (180 / M_PI);
    angulo_magnetometro = angulo_magnetometro - mag_declinacion; //
    Compensa la declinación magnética
    // Siempre convierta a ángulos positivos
    if (angulo_magnetometro <
0)angulo_magnetometro=angulo_magnetometro+360;
```

c. Programación del módulo GPS

Para esta programación se hace uso de la librería “TinyGPS” como el Arduino Due no viene incorporado una librería del puerto serial se trabaja directamente con los puertos seriales que tiene por defecto.

```
bool Inicio_GPS() {
  while (Serial2.available()) {
    if (gps.encode(Serial2.read())) {
      return true;
    }
  }
  return false;
}

void Lectura_GPS(TinyGPS &gps) {
  gps.get_position(&latitud, &longitud, &age);
  Inicio_GPS(); // Si no alimentamos el gps durante toda la rutina,
  podemos eliminar caracteres y obtener errores de suma de comprobación
  gps.f_get_position(&lec_latitud, &lec_longitud, &age);
  Inicio_GPS();
}
```

d. Programación de los filtros o muestras de los datos GPS

En esta parte se realizó el filtrado de todos los datos obtenidos del GPS para mejorar los resultados finales ya que en un inicio se tomó sólo los valores directos del GPS y el vehículo enloquecía al momento de tomar el rumbo hacia el punto marcar por medio del filtrado de los datos se mejoró la parte de dirigir el vehículo a su destino para ello se muestra la curva que se genera de un dato directo del GPS y un dato filtrado por medio del promedio de muestras en la Imagen 6.

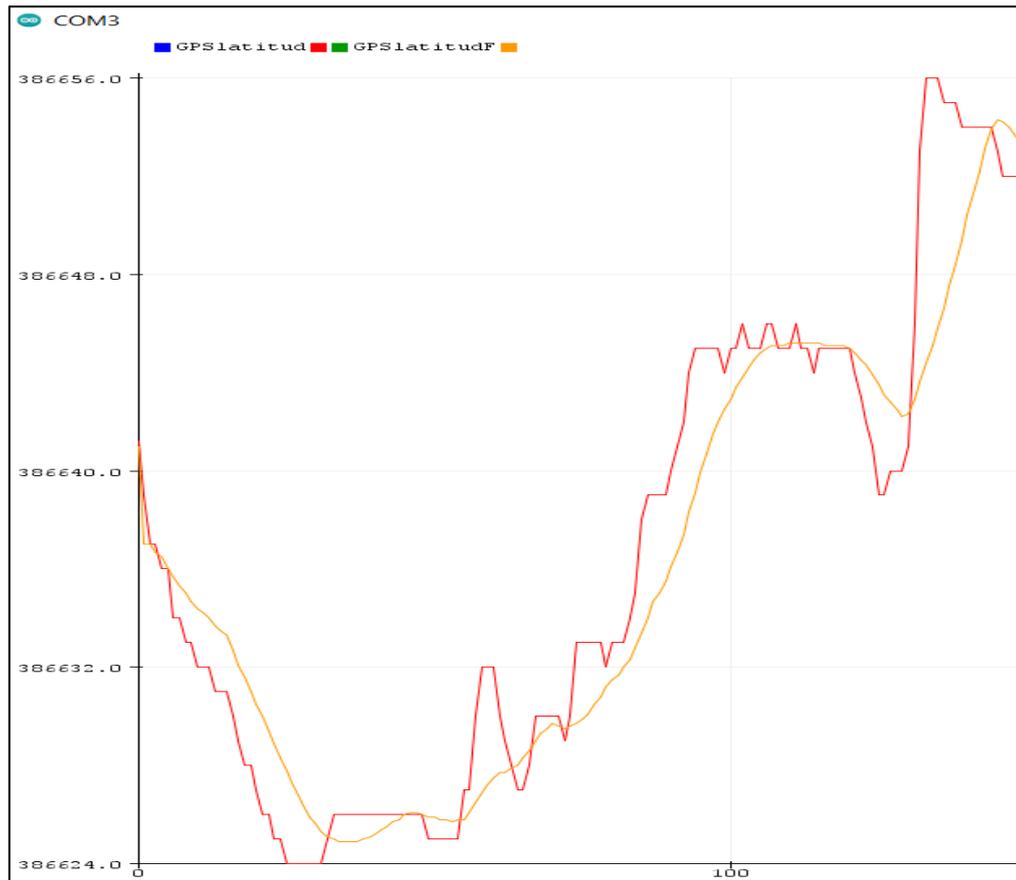


Imagen 6. Lecturas de valores GPS con filtro y sin filtro
Fuente. Autoría propia

Como se observa en la Imagen 6 el color rojo es el dato recogido directo del GPS y el color amarillo el dato filtrado, se observa que se eliminan los picos reduciendo la inestabilidad y mejorando el comportamiento del vehículo.

```

puntos_referencia Calculos_filtros_GPS() {
    float latitud_sum;
    float longitud_sum;
    puntos_referencia puntos_filtrados_ref;
    latitud_sum = 0;
    longitud_sum = 0;
    // Agrega todos los valores en cada búfer
    for (int i = 0; i < inicio_llenado_filtro; ++i) {
        latitud_sum = latitud_sum + filtro_GPS_latitud[i];
        longitud_sum = longitud_sum + filtro_GPS_longitud[i];
    }
    // Toma el promedio
    puntos_filtrados_ref.latitud = latitud_sum /
float(inicio_llenado_filtro);
    puntos_filtrados_ref.longitud = longitud_sum /
float(inicio_llenado_filtro);
    Serial1.print(puntos_filtrados_ref.latitud, 7); Serial1.print(",");
    Serial1.println(puntos_filtrados_ref.longitud, 7);
    return puntos_filtrados_ref; // Devuelve valores filtrados}

```

e. Programación para el cálculo del vector posición

La programación para encontrar el vector posición es mediante el método de haversine que no es más que la medición de distancia que existe entre el vehículo y la ubicación a dirigirse para ello se utilizó las ecuaciones (7), (8), (9), y también el radio de la tierra que es un valor aproximado más 6378 km, pero aquí se genera un error ya que este valor asemeja a la tierra como que fuese completamente esférica.

```
delta_latitud = radians(punto_ruta.latitud - gps_latitud);
gps_f_latitud_rad = radians(gps_latitud);
puntoref_latitud_rad = radians(punto_ruta.latitud);
delta_longitud = radians(punto_ruta.longitud - gps_longitud);

// a = sen^2(Δφ/2) + cos(φ1)·cos(φ2) · sen^2 (Δλ/2)
a_haversine = sin(delta_latitud / 2.0) * sin(delta_latitud / 2.0);
parcial = cos(gps_f_latitud_rad) * cos(puntoref_latitud_rad);
parcial = parcial * sin(delta_longitud / 2.0) * sin(delta_longitud / 2.0);
a_haversine += parcial;

// c=2·arctan2(√a,√(1-a))
c_haversine = 2 * atan2(sqrt(a_haversine), sqrt(1.0 - a_haversine));

// d=r·c
d_haversine = 6378000.0 * c_haversine; // Radio ecuatorial de la tierra
6378000 metros
```

Luego de haber calculado la distancia se calcula el ángulo que tiene el vehículo con respecto a la ubicación establecida y por último se calcula el error de rumbo por medio del cual el vehículo tomará la decisión de dirigirse a la izquierda, derecha o centro haciendo uso de las ecuaciones (10), (11).

```
// θ=atan2(senΔλ·cosφ2, cosφ1·senφ2-senφ1·cosφ2·cosΔλ )
angulo_puntoref = atan2(sin(puntoref_longitud_rad -
gps_f_longitud_rad) * cos(puntoref_latitud_rad), cos(gps_f_latitud_rad) *
sin(puntoref_latitud_rad) - sin(gps_f_latitud_rad) *
cos(puntoref_latitud_rad) * cos(puntoref_longitud_rad -
gps_f_longitud_rad));

angulo_puntoref = angulo_puntoref * 180 / PI; // Convertir de radianes
a grados

/* Controla la navegación del coche robot hacia la meta. */
void Control_navegacion() {
    error_rumbo = (angulo_puntoref - angulo_magnetometro);
```

f. Programación de los actuadores

En esta programación en un principio se tenía declarado tres estados los cuales que eran izquierda, derecha y centro distribuidos de -180° a -90° izquierda, -90° a 90° centro, 90° a 180° derecha pero presentó desviaciones muy grandes hacia el punto marcado por lo que se optó por distribuir en 5 estados para reducir el ángulo en el cual el vehículo tome la acción de dirigir se a la izquierda derecha o centro de la siguiente forma -180° a -45° izquierda rápido, -45° a -5° izquierda, -5° a 5° centro, 5° a 45° derecha lento, 45° a 180° derecha rápido, por medio de estos se logró que el vehículo tenga menos grados de desviación que logra llegar al punto indicado correctamente.

```
// El error está entre +5 y +45 (para Angulo_rango_div = 0.25):
if (error_rumbo > Angulo_direccion_min && error_rumbo <=
Angulo_direccion_max * Angulo_rango_div) {
    DERECHA(); // Gire a la derecha
    accionamiento = "Derecha";
}
// El error está entre +45 y +180 (para Angulo_rango_div = 0.25):
else if (error_rumbo > Angulo_direccion_max * Angulo_rango_div &&
error_rumbo <= Angulo_direccion_max) {
    DERECHA_RAPIDO(); // Gire a la derecha rápido
    accionamiento = "Derecha rapido";
}
// El error está entre -5 y -45 (para Angulo_rango_div = 0.25):
else if (error_rumbo < -Angulo_direccion_min && error_rumbo >= -
Angulo_direccion_max * Angulo_rango_div) {
    IZQUIERDA(); // Girar a la izquierda
    accionamiento = "Izquierda";
}
// El error está entre -45 y -180 (para Angulo_rango_div = 0.25):
else if (error_rumbo < -Angulo_direccion_max * Angulo_rango_div &&
error_rumbo >= -Angulo_direccion_max) {
    IZQUIERDA_RAPIDO(); // Gire a la izquierda rápido
    accionamiento = "Izquierda rapido";
}
// El error está entre -5 y +5 (para Angulo_rango_div = 0.25):
else if (error_rumbo >= -Angulo_direccion_min && error_rumbo <=
Angulo_direccion_min) {
    DELANTE(); // Hacia adelante
    accionamiento = "Delante";}
```

g. Programación del control de la base remota

Para el control de la base remota se ocupó la librería "SoftwareSerial" la cual permitirá una comunicación por el puerto serial por medio de los Xbee para poder enviar la información que los puntos o coordenadas para que el vehículo se dirija y el vehículo

enviaba las coordenadas para poder visualizar el recorrido del vehículo haciendo uso te Google Maps.

```
#include <SoftwareSerial.h>
SoftwareSerial Xbee(50,51);
char enviados;
char recibidos;
/* Formato el cual se debe ingresar las cordenadas
   letra A seguido de la latitud, 2° letra B seguido de la longitud, ....
   cada cordenada latitud y longitud lleva dos letras del abecedario en
orden
   el programa solo puede enviar 4 cordenadas*/
// A0.3798225, B-78.0987583, C0.3799653, D-78.0985913, E0.3801484, F-
78.0984794, G0.38010920, H-78.0983520
// link para copiar las cordenadas que recibimos por el puerto serial
// y poder convertirlas a .kmz y subirlas en google maps si se quiere
vsualizar
// el recorrido que tuvo el vehiculo
//https://www.gpsvisualizer.com/map_input?form=googleearth

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Inicializar comunicación serial.
  // Inicializar comunicación serial para el módulo Xbee
  Xbee.begin(9600);
  delay(1000);
}
void loop() {
  // put your main code here, to run repeatedly:
  if (Xbee.available()){
    recibidos = Xbee.read();
    Serial.print(recibidos);
  }
  if (Serial.available()){
    enviados=Serial.read();
    Xbee.print(enviados); }
}
```

7. Aspectos técnicos para tener en cuenta

- Antes de conectar la batería se debe asegurar que el interruptor este apagado (0), al igual que el interruptor del variador de velocidad, el cual se encuentra en la parte de abajo del chasis y desconectar el conector que energiza al Arduino.
- Antes de encender el vehículo colocar la antena del transmisor de video TS832 ya que si no se la coloca y se energiza el vehículo podría estropear el transmisor por sobrecalentamientos generados.
- Colocar la cubierta del lente de la cámara al finalizar las pruebas, así se evita rayones, golpes, suciedades que afecten al lente de la cámara.

- Cuando se pruebe la comunicación con el transmisor de video verificar que se encuentren en la misma frecuencia y canal caso contrario no receptara nada de lo que capte la cámara.
- El conector de carga de la batería está conectado a una salva Li-Po, la cual indica el nivel de voltaje total, por cada celda, e informa cuando la batería está llegando a su nivel mínimo para proceder a cargar la batería, se recomienda no forzar el funcionamiento cuando la salva Li-Po indique que la batería ya necesita ser cargada ya que un mal uso acabaría con la vida útil de la batería.
- Para cargar la batería Li-Po se debe conectar los cables más delgados que vienen con un conector macho blanco a un cargador de baterías Li-Po. Al momento de cargar las baterías verificar el voltaje total y de cada celda. Por lo general el voltaje de carga máximo es 12.9v, pero no es recomendable cargarle hasta ese valor por lo que se recomienda cargar la batería hasta un máximo de 12.3v a 12.5v y de descarga 9.2v a 9.5 v así alargando la vida útil de la batería.

8. Mejoras futuras

- El vehículo cuenta con una placa distribuidora de voltaje la cual tiene salidas de voltaje de 12v y 5v que aún pueden ser utilizadas para poder conectar algún otro dispositivo electrónico.
- El controlador Arduino Due dispone de más salidas digitales y analógicas si se quiere añadir algún tipo de sensor u otro componente.
- Al chasis se le podría añadir un sistema de suspensión para que el vehículo transite con normalidad por superficies irregulares y los componentes no sufran golpes por medio de las vibraciones.
- El motor brushless y regulador de velocidad soporta salpicaduras de agua según datos de fabricante, pero no se realizó las pruebas en terrenos húmedos para evitar contratiempos.
- El uso de antenas de señal mejora la comunicación y transmisión de video ya que brinda un alcance a distancias más largas por lo que se podría adaptarle al módulo Xbee, transmisor y receptor de videos mejores tipos de antenas.

Anexo I: Manual de Mantenimiento

INTRODUCCIÓN

La implementación de cualquier tipo de prototipo conlleva la realización de un manual de mantenimiento con el fin de detallar las revisiones que se debe realizar para que cualquier usuario que tenga acceso al prototipo tenga en cuenta el mantenimiento que se deba realizar para alargar la vida útil del prototipo.

1. Mantenimiento para realizarse antes de su uso

Cada vez que el prototipo vaya a ser usado se debe verificar algunos aspectos para no tener problemas a futuro como los que se menciona a continuación.

- Antes de conectar la batería al vehículo, revisar que los interruptores de encendido y apagado de la alimentación y del regulador de velocidad se encuentren apagados.
- Revisar si existe impurezas en los ejes de las llantas para evitar acumulaciones de basura y desgaste en el eje de la llanta.
- Verificar que la antena del transmisor de video este bien colocada y sujeta, para evitar daños en el equipo o que su funcionamiento no sea eficiente.
- Verificar que la batería este cargada por medio de la salva Li-Po ya que usar el vehículo con la batería baja, puede reducir la vida útil de la misma e incluso dañarla.
- Revisar que no exista impurezas en los sensores ya que podría provocar un mal funcionamiento.
- Al finalizar el uso del vehículo primero apagar el interruptor del regulador de velocidad, seguido el interruptor de la fuente de alimentación, luego desconectar el conector del Arduino, por último, desconectar el conector de la batería de la fuente de alimentación, colocar su respectivo protector al lente de la cámara.

2. Mantenimiento preventivo

Este tipo de mantenimiento se lo debe realizar dependiendo el uso del vehículo, si su uso es muy frecuente lo más aconsejable es cada 15 días, si el uso es frecuente cada mes, y si es poco frecuente cada dos meses. Los aspectos para revisar son los siguientes:

- Verificar si los tornillos estén bien sujetos o exista una alguna parte móvil realizar un ajuste.
- Verificar que el ventilador del regulador de velocidad no tenga impurezas caso contrario limpiar las obstrucciones.
- Realizar un engrase al eje del motor para que no exista ningún roce o calentamiento.
- Verificar que no existe impurezas en el servomotor para que pueda girar libremente el vehículo, así evitando desgaste del eje del servomotor.
- Verificar continuidad en los conductores.
- Limpiar las impurezas como polvos en la placa de distribución de voltaje y todos los componentes electrónicos.

Anexo J: Código de programación

```
// UNIVERSIDAD TECNICA DEL NORTE
// CARRERA DE INGENIERIA EN MATENIMIENTO ELECTRICO
// GALO CARLOSAMA
// VEHICULO TERRESTRE NO TRIPULADO AUTONOMO
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Librerias
#include <TinyGPS.h>
#include <Wire.h>
#include <Servo.h>
#include "I2Cdev.h"
#include "HMC5883L.h"

// Variables decalradas
HMC5883L magnetometro; // Declaramos el objeto de brújula
TinyGPS gps; // Declaramos el objeto GPS
int RX2 = 16; // Declaramos el pin 16tx (RX del GPS)
int TX2 = 17; // Declaramos el pin 17tx (TX del GPS)
Servo ServoMotor;
Servo Esc;
int RX1 = 19; // Declaramos el pin 18tx (RX del xbee)
int TX1 = 18; // Declaramos el pin 19rx (TX del xbee)
int servo = 5; // Declaramos pin del servo
int motor = 7; // Declaramos pin del motor
int Sensor_izquierda = A0; // Declaramos pin del SENSOR 1
int Sensor_centro = A1; // Declaramos pin del SENSOR 2
int Sensor_derecha = A2; // Declaramos pin del SENSOR 3

// Variables para los sensores
int D_cm1 = 0;
int D_cm2 = 0;
int D_cm3 = 0;
// variables para recibir datos
char datos_recibidos;
char datos_enviados;

#define radio_tolerancia 1.0 // Radio de tolerancia para alcanzar la meta

// Velocidades de los coches robot
// (rango de velocidades atras:74-83 ~ delante:104-180)
const int Velocidad = 109;
const int Velocidad_normal = 107;
const int Velocidad_lenta = 105;
const int Velocidad_paro = 95;
const int vel1 = 104; // delante
const int vel2 = 83; // reversa
const int Velocidad_reversa = 80;
// Giros del servomotor
const int Izq = 10;
const int Izq_rapido = 6;
const int Delante = 23;
const int Dere = 38;
const int Dere_rapido = 44;

// Constantes para definir rangos de error para el control de navegacion
#define Angulo_direccion_max 180
#define Angulo_direccion_min 5
#define Angulo_rango_div 0.25
```

```

int iniciar = 0;
// Almacena la siguiente acción que debe realizar el vehiculo
String accionamiento;

// Variables para almacenar las cordenadas ingresadas
float A, B, C, D, E, F, G, H;

// Variables para almacenar waypoints
// Estructura para almacenar waypoints (coordenadas GPS):
struct puntos_referencia {
    float latitud;
    float longitud;
};

puntos_referencia puntos_navegacion[] = {
    { A, B}, // Punto 1
    { C, D}, // Punto 2
    { E, F}, // Punto 3
    { G, H} // Punto 4
};

byte punto_ref_inicial = 0; // Índice del punto de referencia actual en
matriz
// Obtener el número de waypoints:
int numero_puntos_referencia = sizeof(puntos_navegacion) /
sizeof(puntos_navegacion[0]);

// Variables auxiliares para calcular el vector de navegación:
float puntoref_latitud_rad;
float puntoref_longitud_rad;

// Variables para lecturas de magnetometro o brujula
int16_t mx, my, mz; // Componentes del campo magnético de la Tierra
float angulo_magnetometro; // Almacena el ángulo de la brújula o
magnetometro
const float mag_declinacion = -4.4; //declinación de -4° 25' Ibarra
//https://www.magnetic-declination.com
int error_rumbo; // Almacena el error de rumbo

// Variables para lecturas de GPS
float lec_latitud, lec_longitud; // Almacena la lectura del receptor GPS
float angulo_puntoref; // Almacena el acimut del objetivo actual
float calculo_dist_meta = 0; // La distancia calculada hasta la meta

// Declaramos la variables para la obtención de datos GPS
long latitud, longitud;
unsigned long age, date, hour, chars;
unsigned short sentences, failed;

// Variables para el filtro de medida. Incrementa Num_puntos_filtrado si se
quiere más filtrados de las lecturas del GPS,
// también agrega más demora al sistema.
#define Num_puntos_filtrado 15
float filtro_GPS_latitud[Num_puntos_filtrado]; // Amortiguador para latitud
float filtro_GPS_longitud[Num_puntos_filtrado]; // Amortiguador para
longitud
puntos_referencia filtros_GPS; // Para almacenar una lectura de GPS
filtrada
unsigned int inicio_llenado_filtro = 0; // inicio de llenado de tampones de
filtro

```

```

/* Función de configuración */
void setup() {
  // Inicializar pines de control para el controlador del motor
  PARAR(); // Para el coche
  Serial.begin(9600); // Inicializar comunicación serial.
  Serial2.begin(9600); // Inicializar comunicación serial para el módulo
GPS
  Serial1.begin(9600); // Inicializar comunicación serial para el módulo
Xbee
  Wire.begin(); // Inicializa la comunicación I2C. para el
magnetometro o brújula
  Esc.attach(motor); // Declarar el regulador de velocidad al pin 7
  ServoMotor.attach(servo); // Declarar el servomotor al pin 5
  Serial1.print("Ingrese 4 coordenadas iniciando con la letra A seguido de
la coordenada ejemplo: Alat1, Blon1, Clat2, Dlon2...: ");
  magnetometro.initialize(); // Inicializar el magnetometro o brújula
  delay(1000);
}

/* Bucle principal */
void loop() {
  if (Serial1.available()) {
    datos_recibidos = Serial1.read();
    Cordenadas();
    //Serial.print(datos_recibidos);
  }
  if (Serial.available()) {
    datos_enviados = Serial.read();
    Serial1.print(datos_enviados);
  }
  D_cm1 = dist_izq(20); //lectura de dist_izq
  D_cm2 = dist_centro(20); //lectura de dist_izq
  D_cm3 = dist_dere(20); //lectura de dist_izq
  /*Serial.print("Dist 1: "); Serial.print(D_cm1); Serial.print(" cm
");
  Serial.print("Dist 2: "); Serial.print(D_cm2); Serial.print(" cm
");
  Serial.print("Dist 3: "); Serial.print(D_cm3); Serial.println("
cm");*/
  delay(50);

  while (iniciar == 1) {
    //Serial1.print("INICIAR");
    //Serial1.println();*/
    Lectura_magnetometro(); // Leer la brújula digital

    if (Inicio_GPS()) { // Consulta el GPS
      Lectura_GPS(gps); // Leer el GPS

      // Almacena la nueva lectura de GPS en búferes de filtro
      Almacenamiento_lecturas_GPS(lec_latitud, lec_longitud);

      // Obtén latitud y longitud de GPS filtradas
      filtros_GPS = Calculos_filtros_GPS();

      // Calcular la distancia y el rumbo a la meta
      Calculo_vector_navegacion(filtros_GPS.latitud, filtros_GPS.longitud);

      // Imprime datos en la PC solo para depurar
      Imprimir_datos();
    }
  }
}

```

```

    }
    // Mueve el coche a la meta
    Control_navegacion();
    Control_esquivar();
}
}

void Cordenadas (void) {
    if (datos_recibidos == 'A') {
        A = Serial1.parseFloat();
    }
    else if (datos_recibidos == 'B') {
        B = Serial1.parseFloat();
    }
    else if (datos_recibidos == 'C') {
        C = Serial1.parseFloat();
    }
    else if (datos_recibidos == 'D') {
        D = Serial1.parseFloat();
    }
    else if (datos_recibidos == 'E') {
        E = Serial1.parseFloat();
    }
    else if (datos_recibidos == 'F') {
        F = Serial1.parseFloat();
    }
    else if (datos_recibidos == 'G') {
        G = Serial1.parseFloat();
    }
    else if (datos_recibidos == 'H') {
        H = Serial1.parseFloat();
        Serial1.print(" CARGADO ");
        Serial1.println();
    } else if (datos_recibidos == '1') {
        iniciar = 1;
        Serial1.print(" INICIAR ");
    }
    /*Serial.print("lat1: ");    Serial.print(A,8);
    Serial.print(" lon1: ");    Serial.print(B,8);
    Serial.print(" lat2: ");    Serial.print(C,8);
    Serial.print(" long2: ");   Serial.print(D,8);
    Serial.print(" lat3: ");    Serial.print(E,8);
    Serial.print(" long3: ");   Serial.print(F,8);
    Serial.print(" lat4: ");    Serial.print(G,8);
    Serial.print(" long4: ");   Serial.println(H,8);*/
}

float dist_izq(int n) {
    long suma = 0;
    for (int i = 0; i < n; i++) {
        suma = suma + analogRead(Sensor_izquierda);
    }
    float prom = suma / n;
    float adc = prom * 0.0048875855; // 5v/1023 =0.0048875855
    float dist_izq_cm = 13 * pow(adc, -1);
    return (dist_izq_cm);
}

float dist_centro(int m) {
    long suma2 = 0;

```

```

    for (int j = 0; j < m; j++) {
        suma2 = suma2 + analogRead(Sensor_centro);
    }
    float prom2 = suma2 / m;
    float adc2 = prom2 * 0.0048875855; // 5v/1023 =0.0048875855
    float dist_izq_cm2 = 13 * pow(adc2, -1);
    return (dist_izq_cm2);
}

float dist_dere(int y) {
    long suma3 = 0;
    for (int k = 0; k < y; k++) {
        suma3 = suma3 + analogRead(Sensor_derecha);
    }
    float prom3 = suma3 / y;
    float adc3 = prom3 * 0.0048875855; // 5v/1023 =0.0048875855
    float dist_izq_cm3 = 13 * pow(adc3, -1);
    return (dist_izq_cm3);
}

/* Almacena una lectura de GPS en filtro */
void Almacenamiento_lecturas_GPS(float latitud, float longitud) {
    // Desplaza todos los valores del búfer hacia la cola
    for (int i = (Num_puntos_filtrado - 1); i > 0; --i) {
        filtro_GPS_latitud[i] = filtro_GPS_latitud[i - 1];
        filtro_GPS_longitud[i] = filtro_GPS_longitud[i - 1];
    }

    // Insertar nuevos valores en la cabecera
    filtro_GPS_latitud[0] = latitud;
    filtro_GPS_longitud[0] = longitud;

    // Incrementar el número de lecturas almacenadas en búfer
    if (inicio_llenado_filtro < Num_puntos_filtrado) {
        ++inicio_llenado_filtro;
    }
}

/***** Calcula la latitud y la longitud filtradas mediante un filtro *****/
puntos_referencia Calculos_filtros_GPS() {
    float latitud_sum;
    float longitud_sum;

    puntos_referencia puntos_filtrados_ref;

    latitud_sum = 0;
    longitud_sum = 0;

    // Agrega todos los valores en cada búfer
    for (int i = 0; i < inicio_llenado_filtro; ++i) {
        latitud_sum = latitud_sum + filtro_GPS_latitud[i];
        longitud_sum = longitud_sum + filtro_GPS_longitud[i];
    }

    // Toma el promedio
    puntos_filtrados_ref.latitud = latitud_sum /
float(inicio_llenado_filtro);
    puntos_filtrados_ref.longitud = longitud_sum /
float(inicio_llenado_filtro);
}

```

```

Serial1.print(puntos_filtrados_ref.latitud, 7); Serial1.print(",");
Serial1.println(puntos_filtrados_ref.longitud, 7);
return puntos_filtrados_ref; // Devuelve valores filtrados
}

/***** Consulta el receptor GPS *****/
bool Inicio_GPS() {
    while (Serial2.available()) {
        if (gps.encode(Serial2.read())) {
            return true;
        }
    }
    return false;
}

/***** Obtiene una lectura de la brújula *****/
void Lectura_magnetometro(void) {
    // Obtenga los componentes del campo magnético en X, Y, Z
    magnetometro.getHeading(&mx, &my, &mz);

    //Calculo del ángulo del eje X norte
    angulo_magnetometro = atan2(my, mx);
    angulo_magnetometro = angulo_magnetometro * (180 / M_PI); //
    angulo_magnetometro = angulo_magnetometro - mag_declinacion; // Compensa
la declinación magnética

    // Siempre convierta a ángulos positivos
    if (angulo_magnetometro < 0) angulo_magnetometro = angulo_magnetometro +
360;
    //Serial.print("angulo_magnetometro: ");
    //Serial.println(angulo_magnetometro,0);
    //delay(100);
}

/* Calcula el vector de navegación */
void Calculo_vector_navegacion(float gps_latitud, float gps_longitud) {
    puntos_referencia punto_ruta;

    // Obtener el objetivo actual
    punto_ruta = obtener_puntoref_inicial(punto_ref_inicial);

    float gps_f_latitud_rad;
    float gps_f_longitud_rad;
    float a_haversine = 0;
    float c_haversine = 0;
    float d_haversine = 0;
    float parcial = 0;
    float delta_latitud = 0;
    float delta_longitud = 0;

    // Calculo de la distancia a la meta con la fórmula de Haversine
    // *****
    delta_latitud = radians(punto_ruta.latitud - gps_latitud);
    gps_f_latitud_rad = radians(gps_latitud);
    puntoref_latitud_rad = radians(punto_ruta.latitud);
    delta_longitud = radians(punto_ruta.longitud - gps_longitud);

    //  $a = \sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2)$ 
    a_haversine = sin(delta_latitud / 2.0) * sin(delta_latitud / 2.0);
    parcial = cos(gps_f_latitud_rad) * cos(puntoref_latitud_rad);
}

```

```

    parcial = parcial * sin(delta_longitud / 2.0) * sin(delta_longitud /
2.0);
    a_haversine += parcial;

    // c=2*arctan2(√a,√(1-a))
    c_haversine = 2 * atan2(sqrt(a_haversine), sqrt(1.0 - a_haversine));

    // d=r*c
    d_haversine = 6378000.0 * c_haversine; // Radio ecuatorial de la tierra
6378000 metros

    calculo_dist_meta = d_haversine;
    //Serial.print(calculo_dist_meta);

    // Comprueba si estamos dentro del radio de tolerancia de la meta.
    if (d_haversine < radio_tolerancia) {
        PARAR(); // Para el coche
        Serial1.print("***** PUNTO ALCANSADO *****");
        Serial1.println();
        delay(5000); // Retraso solo para comprobar visualmente si el coche
alcanzó exactamente la meta
        punto_ref_inicial++; // Cambiar al siguiente punto de referencia
    }

    // Comprueba si llegamos a todos los puntos de referencia
    if (punto_ref_inicial == numero_puntos_referencia) {
        PARAR(); // Para el coche
        Serial1.print("***** RECORRIDO EXITOSO *****");
        Serial1.println();
        delay(5000);
        while (1); // Detén el programa (reinicia la placa Arduino para
repetir)
    }

    // Calculo del angulo punto de referencia
    // *****
    gps_f_longitud_rad = radians(gps_longitud);
    puntoref_longitud_rad = radians(punto_ruta.longitud);

    //  $\theta = \text{atan2}(\sin\Delta\lambda \cdot \cos\phi_2, \cos\phi_1 \cdot \sin\phi_2 - \sin\phi_1 \cdot \cos\phi_2 \cdot \cos\Delta\lambda)$ 
    angulo_puntoref = atan2(sin(puntoref_longitud_rad - gps_f_longitud_rad) *
cos(puntoref_latitud_rad),
        cos(gps_f_latitud_rad) *
sin(puntoref_latitud_rad) - sin(gps_f_latitud_rad) *
cos(puntoref_latitud_rad) * cos(puntoref_longitud_rad -
gps_f_longitud_rad));
    //Serial.print(angulo_puntoref);
    angulo_puntoref = angulo_puntoref * 180 / PI; // Convertir de radianes a
grados

    // Convierta siempre a ángulos positivos
    if (angulo_puntoref < 0) {
        angulo_puntoref += 360;
    }
}

/* Controla la navegación del coche robot hacia la meta. */
void Control_navegacion() {
    error_rumbo = (angulo_puntoref - angulo_magnetometro); // Calculo del
error de rumbo
    // Ángulo correcto para envolver

```

```

if (error_rumbo < -180) {
    error_rumbo = error_rumbo + 360;
}
if (error_rumbo > 180) {
    error_rumbo = error_rumbo - 360;
}
// El error está entre +5 y +45 (para Angulo_rango_div = 0.25):
if (error_rumbo > Angulo_direccion_min && error_rumbo <=
Angulo_direccion_max * Angulo_rango_div) {
    DERECHA(); // Gire a la derecha
    accionamiento = "Derecha";
}

// El error está entre +45 y +180 (para Angulo_rango_div = 0.25):
else if (error_rumbo > Angulo_direccion_max * Angulo_rango_div &&
error_rumbo <= Angulo_direccion_max) {
    DERECHA_RAPIDO(); // Gire a la derecha rápido
    accionamiento = "Derecha rapido";
}

// El error está entre -5 y -45 (para Angulo_rango_div = 0.25):
else if (error_rumbo < -Angulo_direccion_min && error_rumbo >= -
Angulo_direccion_max * Angulo_rango_div) {
    IZQUIERDA(); // Girar a la izquierda
    accionamiento = "Izquierda";
}

// El error está entre -45 y -180 (para Angulo_rango_div = 0.25):
else if (error_rumbo < -Angulo_direccion_max * Angulo_rango_div &&
error_rumbo >= -Angulo_direccion_max) {
    IZQUIERDA_RAPIDO(); // Gire a la izquierda rápido
    accionamiento = "Izquierda rapido";
}

// El error está entre -5 y +5 (para Angulo_rango_div = 0.25):
else if (error_rumbo >= -Angulo_direccion_min && error_rumbo <=
Angulo_direccion_min) {
    DELANTE(); // Hacia adelante
    accionamiento = "Delante";
}
else {
    accionamiento = "No definido";
}
Serial.print(" Error rumbo: "); Serial.print(error_rumbo);
Serial.print(" Condicion: "); Serial.println(accionamiento);
//Serial1.print(" Error rumbo: "); Serial1.print(error_rumbo);
//Serial1.print(" Condicion: "); Serial1.println(accionamiento);
}

void Control_esquivar() {
    if (D_cm1 <= 23) { // Sensor izquierda detecta obstaculo
        accionamiento = "Esquivar sensor izquierda";
        Serial.print(" Dist izq: "); Serial.print(D_cm1);
        Serial.print(" Condicion: "); Serial.println(accionamiento);
        DERECHA_RAPIDO();
    }
    else if (D_cm2 <= 23) { // Sensor centro detecta obstaculo
        accionamiento = "Esquivar sensor centro";
        Serial.print(" Dist centro: "); Serial.print(D_cm2);
        Serial.print(" Condicion: "); Serial.println(accionamiento);
        DERECHA_RAPIDO();
    }
}

```

```

    }
    else if (D_cm3 <= 23) { // Sensor derecho detecta obstaculo
        accionamiento = "Esquivar sensor derecha";
        Serial.print(" Dist dere: "); Serial.print(D_cm3);
        Serial.print(" Condicion: "); Serial.println(accionamiento);
        IZQUIERDA_RAPIDO();
    }
}

/* Rutina auxiliar TinyGPS, viene con la biblioteca */
void Lectura_GPS(TinyGPS &gps) {
    gps.get_position(&latitud, &longitud, &age);
    Inicio_GPS(); // Si no alimentamos el gps durante esta larga rutina,
    podemos eliminar caracteres y obtener errores de suma de comprobación
    gps.f_get_position(&lec_latitud, &lec_longitud, &age);
    Inicio_GPS();

    //Serial.print("LAT=");Serial.print(lec_latitud ==
    TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : lec_latitud, 6);
    //Serial.print(" LON=");Serial.print(lec_longitud ==
    TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : lec_longitud, 6);
    //gps.stats(&chars, &sentences, &failed);Serial.print("Estadísticas:
    characters: ");
    //Serial.print(chars); Serial.print(" sentences: ");
    Serial.print(sentences);
    //Serial.print(" suma de comprobación fallida: ");
    Serial.println(failed);
}

/* Devuelve un punto de referencia de la matriz */
struct puntos_referencia obtener_puntoref_inicial(int inicial) {
    return puntos_navegacion[inicial];
}

/***** GIROS DE ACTUADORES *****/
// Gira el auto a la izquierda lentamente
void DELANTE() {
    ServoMotor.write(Delante);
    Esc.write(Velocidad);
}

// Gira el auto a la izquierda lentamente
void IZQUIERDA() {
    ServoMotor.write(Izq);
    Esc.write(Velocidad_normal);
}

// Gira el auto a la derecha lentamente
void DERECHA() {
    ServoMotor.write(Dere);
    Esc.write(Velocidad_normal);
}

// Gira el coche a la izquierda rápidamente
void IZQUIERDA_RAPIDO() {
    ServoMotor.write(Izq_rapido);
    Esc.write(Velocidad_lenta);
}

// Gira el coche a la derecha rápidamente
void DERECHA_RAPIDO() {

```

```

    ServoMotor.write(Dere_rapido);
    Esc.write(Velocidad_lenta);
}

// Detiene el coche
void PARAR() {
    Esc.write(Velocidad_paro);
    ServoMotor.write(Delante);
}

// imprimir datos y graficar los valores de lat y lon que resultan por
medio del filtro en serial plotter
void Imprimir_datos(void)
{
    // latitud original y filtrada (multiplicada por un factor para ayudar al
    // ;Capacidades mínimas de Serial Plotter!)
    //Serial.print(" GPS latitud: "); Serial.print(1000000*lec_latitud, 1);
    //Serial.print(" GPS latitud Filtros: ");
    Serial.print(1000000*filtros_GPS.latitud, 1);
    //Serial.print(" GPS latitud Filtros: ");
    Serial.print(filtros_GPS.latitud,5);

    // longitud original y filtrada
    //Serial.print(" GPS longitud: "); Serial.print(1000000*(lec_longitud),
1);
    //Serial.print(" GPS longitud Filtros: ");
    Serial.print(1000000*(filtros_GPS.longitud), 1);
    //Serial.print(" GPS longitud Filtros: ");
    Serial.print(filtros_GPS.longitud,5);

    // Ángulo del waypoint, ángulo del coche robot y error de rumbo
    //Serial.print(" Ang Magnetometro: "); Serial.print(angulo_magnetometro);
    //Serial.print(" Cal dis meta: ");Serial.print(calculo_dist_meta);
    //Serial.print(" Ang pun ref: ");Serial.print(angulo_puntoref);
    //Serial.print(" Error rumbo: ");Serial.print(error_rumbo);
    //Serial.print(" Condicion: ");Serial.print(accionamiento);
    //Serial.println();
}

```