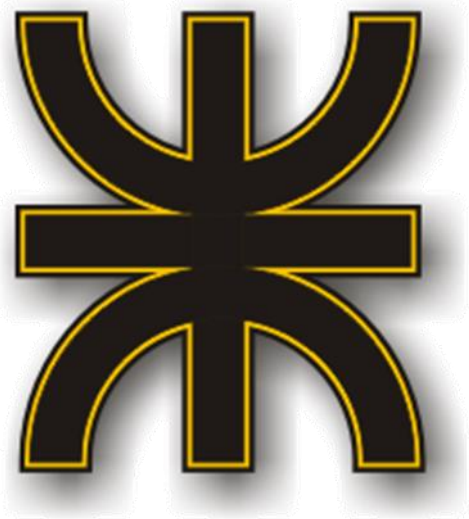
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL TUCUMÁN
Departamento de Ingeniería Electrónica



“Navegación para robots móviles: Implementación de una plataforma de bajo costo para navegación mediante algoritmo SLAM”

Proyecto Final

Estudiante: Robra Prieto, Abel Gonzalo

Tutor Docente: Ing. Ovejero, César Enrique

Profesor Titular: Ing. Rubén Egea

Universidad Tecnológica Nacional
Facultad Regional Tucumán
San Miguel de Tucumán, 2022

Agradecimientos

Este proyecto significa la culminación de una etapa muy importante en mi vida. Cerrar esta etapa me ha llevado mucho esfuerzo y dedicación, pero no lo habría podido lograr sin la constante colaboración de mi esposa Claudia, a quien se lo dedico.

Agradezco principalmente a mi familia por el apoyo y la confianza que me brindaron desde el inicio de mi carrera, especialmente, a mi padre, quien fue mi primer maestro de electrónica. A mi madre, quien con su ejemplo me enseñó que con constancia y sacrificio se puede lograr cualquier objetivo. A mis hijos, quienes me acompañaron y me enseñaron mucho de lo que necesité para realizar este trabajo.

Agradezco al Ingeniero Rubén Egea y al Ingeniero Fabio Poli por su disposición y su tiempo. Al Ingeniero César Ovejero, por haber aceptado ser mi tutor y colaborar en la elaboración de esta tesis. Al Ingeniero Juan Carlos Colombo, Jefe del Departamento de Electrónica, por su apoyo al retomar mis estudios.

A todos los docentes que he tenido la suerte de conocer a lo largo de la carrera y que me han enseñado y preparado en esta disciplina, les extiendo mi más profundo agradecimiento. Y a cada persona que durante estos años me ha ayudado a jamás bajar los brazos, les agradezco desde lo más profundo de mi corazón.

Abel Gonzalo Robra Prieto
San Miguel de Tucumán, 2022

Índice

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN Y OBJETIVOS

"Encontraremos permanentemente grandes oportunidades disfrazadas de problemas insolubles"
Anthony "Lee" Lacocca

En la actualidad está en desarrollo la cuarta revolución industrial, también conocida como Industria 4.0. Esta revolución implica la digitalización de los datos relevantes con el fin de conseguir la automatización de procesos para poder predecir, controlar, planear y producir de forma inteligente.

1.1 Introducción

Para poder transformarse en una industria 4.0, una empresa debe iniciar un proceso de incorporación gradual de distintos componentes tecnológicos novedosos, provenientes de los dominios digital y físico. Por ejemplo:

- Inteligencia artificial.
- Internet de las cosas.
- Robótica.
- Impresión 3D.
- Servicios en la nube.
- Ciberseguridad.

Hoy en día existe una proliferación de todo tipo de vehículos autónomos, que van desde coches que se conducen solos hasta aspiradoras de uso doméstico. Pero quisiera enfocarme en el uso de los robots autónomos, para automatización de la logística interna en la industria.

¿Por qué utilizar los recursos de los empleados para mover materiales cuando se puede automatizar estas tareas y hacer que los empleados se concentren en actividades de mayor valor?

Al automatizar el transporte de materiales, las organizaciones pueden optimizar la productividad y programar las entregas de manera más efectiva para reducir los cuellos de botella en la producción.

Figura 1-1. Robots en depósitos de mercadería.
Fuente: https://www.scmr.com/article/analyst_study_revenues_from_warehouse_robotics_to_top_51_billion_by_2030

Muchos hemos observado con asombro, imágenes de los depósitos de compañías como Amazon, donde un ejército de robots, se encargan de preparar y ordenar las cajas prácticamente sin intervención humana.
Obviamente esta tecnología se está comenzando a aplicar localmente, tanto en fábricas como en depósitos de grandes empresas. Por lo tanto, es necesario contar con personal especializado para instalarlos, mantenerlos, configurarlos, repararlos y tal vez, en algún momento diseñarlos y contruirlos. Existe un trabajo interdisciplinario en este tipo de tecnologías: mecánica, informática y electrónica. Personalmente, creo que el ingeniero electrónico, por la amplitud de temas de su formación académica, tiene cierta ventaja, respecto al resto, a la hora de involucrarse en esta tecnología. Por lo tanto, la idea de este trabajo es estudiar y explorar conceptos de esta tecnología con el objetivo de armar un vehículo autónomo de bajo costo con fines educativos.

1.2 Objetivos

Como se ha explicado en el apartado anterior este proyecto está pensado para adquirir conocimientos en las nuevas tecnologías relacionadas con vehículos autónomos en la industria y la logística de almacenes. Enfocándose especialmente en la implementación de la técnica SLAM, haciendo uso de un sensor LiDAR y utilizando herramientas de la plataforma ROS.
Para conseguir este objetivo principal, se han definido los siguientes objetivos de segundo nivel:
• Desarrollo e implementación de un robot móvil.
• Iniciarse y obtener habilidades en el uso de la plataforma ROS.
• Conectar de manera inalámbrica con el robot.
• Control de los motores DC de manera inalámbrica.
• Control de la trayectoria del robot móvil mediante RViz.
• Elaborar una plataforma en la que se trabaje de forma conjunta con las placas ATMega2560 y Raspberry Pi.
• Desarrollo de odometría mediante la utilización de distintos sensores.
• Analizar el funcionamiento y operatividad de los sensores utilizados.
• Desarrollo de la técnica SLAM en un entorno físico.
• Representar el mapa obtenido por el LiDAR en RViz.

Es importante comprender algunos conceptos que se van a utilizar a lo largo del trabajo. A continuación una breve introducción de ellos mismos

1.3 Rutas Fijas versus Navegación Inteligente

En robótica móvil aplicada a la industria, se pueden considerar dos tipos de vehículos, AGV y AMR.

Figura 1-2. Vehículos AGV y AMR se continuarán utilizando.
Pixar Animation Studios. (2008). Wall-E [Fotograma]. En Wall-E. Walt Disney Studios Motion Pictures.

Un **AGV** tiene una inteligencia mínima a bordo y solo puede obedecer instrucciones de programación simples. Para navegar, debe guiarse por cables, bandas magnéticas o sensores, que normalmente requieren actualizaciones extensas (y costosas) de las instalaciones para instalarse, tiempo durante el cual la producción puede verse interrumpida. El AGV está restringido a seguir estas rutas fijas, que requieren costos adicionales e interrupciones si se necesitan cambios en el futuro. El AGV puede detectar obstáculos frente a él, pero no puede sortearlos, por lo que simplemente se detiene en seco hasta que se elimina el obstáculo.

En contraste, el **AMR** navega a través de mapas que su software construye en el sitio o a través de dibujos de instalaciones precargados. Esta capacidad se puede comparar con un automóvil con un GPS y un conjunto de mapas precargados. Cuando se le enseña la dirección de la casa y del trabajo del propietario, genera la ruta más directa basada en posiciones simples en el mapa. Esto es similar a la forma en que se le enseña al AMR las ubicaciones para recoger y dejar piezas. El AMR utiliza datos de cámaras y sensores integrados y escáneres láser, así como un software sofisticado que le permite detectar su entorno y elegir la ruta más eficiente hacia el objetivo. Funciona de forma totalmente autónoma y si, frente a él, se encuentran carretillas elevadoras, palés, personas u otros obstáculos, el AMR maniobrará alrededor de ellos de forma segura, utilizando la mejor ruta alternativa. Esto optimiza la productividad al garantizar que el flujo de material se mantenga según lo programado.

Al comparar las ventajas entre ambos, nos podemos concentrar principalmente en la flexibilidad, autonomía, colaboración humano-robot y gestión de flotas. El siguiente cuadro representa de forma gráfica las comparativa:

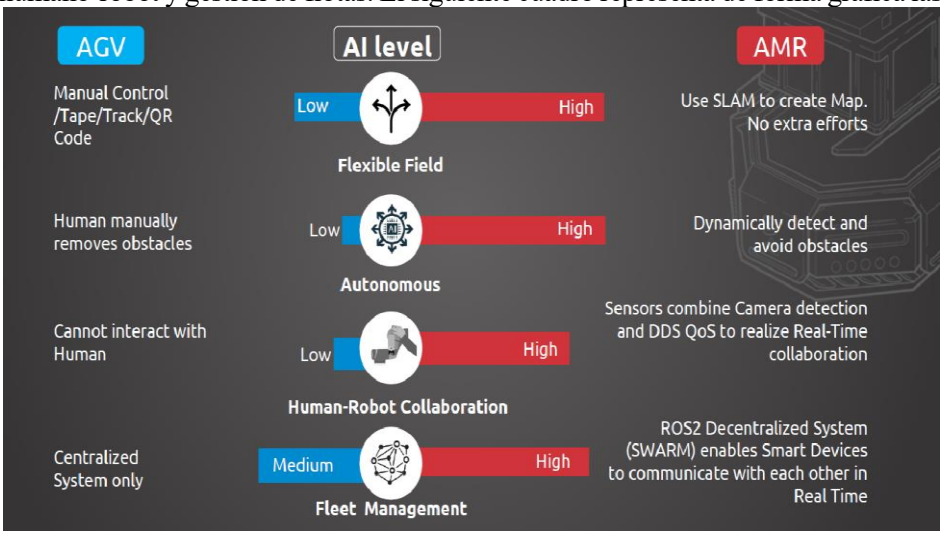


Figura 1-3. Comparación AGV vs AMR.
Fuente: Fuente: www.roboticstomorrow.com/article/2021/07/robots-from-automated-to-autonomous/17234

1.4 Introducción a SLAM

Enfocándonos en los vehículos AMR, surge la necesidad de estudiar el proceso de navegación a través de mapas construidos por el propio vehículo.
El problema de localización y construcción de mapas simultáneos (SLAM), pregunta si es posible que un vehículo autónomo, comience en una ubicación desconocida, en un entorno desconocido y luego construya gradualmente un mapa de este entorno, mientras usa simultáneamente este mapa para calcular la ubicación absoluta del vehículo. A partir de los fundamentos teóricos de la estimación de este problema, se podría demostrar, que una solución al problema SLAM, sí es posible.
Por lo tanto, un vehículo autónomo, colocado en un entorno desconocido, utilizando solamente observaciones relativas, mediante SLAM, podrá construir un mapa ambiental preciso y calcular y realizar un seguimiento de su ubicación en dicho mapa a lo largo del tiempo.
El problema SLAM es visto como uno de los temas más importantes a dominar al desarrollar robots móviles verdaderamente autónomos. Es significativamente más difícil que la mayoría de los problemas robóticos debido al hecho de que el mapa y las posiciones tienen que ser estimados en el camino.
Hay varias áreas que podrían beneficiarse de tener vehículos autónomos con algoritmos SLAM implementados. Ejemplos serían la industria minera, exploración submarina y exploración planetaria.

1.4.1 Movimiento del Robot Mediante SLAM

El desplazamiento de un robot mediante SLAM, es similar al de una persona tratando de encontrar su posición en un lugar desconocido. Primero, la persona observa a su alrededor, en busca de alguna marca que le resulte familiar. Una vez que la persona reconoce una marca en particular (landmark), se puede hacer una idea de dónde

se encuentra respecto a dicho landmark. Pero, si la persona no reconoce ningún landmark, se encontraría perdida. Sin embargo, cuanto más tiempo dedique a la observación del entorno, más landmarks podrá reconocer y empezar a generar una imagen mental (mapa), cada vez más completo, del entorno en el que se encuentra. A medida que la persona se desplace por dicho entorno, dicho mapa será más detallado, por lo tanto, resultará más sencillo desplazarse por el mismo.
Un robot que implemente la técnica SLAM, intenta mapear un entorno desconocido mientras observa donde se encuentra en él. Realizar ambas tareas a la vez, mientras el robot se continúa desplazando, es una de las tareas más complicadas en robótica. El robot necesita saber su posición antes de poder resolver la pregunta de cómo es el entorno. El SLAM resuelve este problema.
Para implementar SLAM, el robot utilizado, debe tener datos consistentes para la generación de la odometría. Cuanto mejor sea la odometría mejor podrá estimar el robot su posición a lo largo del tiempo. Esto normalmente se calcula a partir de mediciones obtenidas por distintos tipos de sensores, como

- Sensores propios del robot como encoders en las ruedas, sensores IMU, etc.
- Sensores de Rango: los cuales proveen información angular y de rango. Como sonares, sensores de distancia Sharp infrarrojos y sensores Lidar entre otros.
- Cámaras: Una cámara es un sensor proyectivo que mide el ángulo (bearing) respecto a los elementos de la imagen, por lo que la profundidad o rango no puede ser obtenida mediante una sola medición

Hay que tener en cuenta que un pequeño error con las lecturas de la odometría puede provocar que, en un trayecto largo, se produzcan desviaciones importantes respecto a la posición estimada. Estos errores deben de tenerse en cuenta en los algoritmos empleados en el cálculo.

1.4.2 Fundamentos Matemáticos

Un componente clave de lo que permite que un robot realice con éxito la navegación es el sensado. Para navegar correctamente, el robot necesita información sobre su entorno. Esta información es suministrada por sensores. Un problema importante, sin embargo, es que los datos del sensor siempre estarán corruptos por ruido hasta cierto punto. Ese significa que, si el robot solo usa los datos del sensor sin procesar, tal como están, la navegación puede producir resultados erróneos. En virtud del hecho de que todas las medidas están en la presencia de ruido, se necesita filtrado. Un enfoque para hacer esto es a través de la estimación Bayesiana recursiva [28, p. 13], que utiliza modelos probabilísticos en para filtrar el ruido.

Representación del Espacio de Estados:
Para definir los estados de un sistema, se utiliza la representación de espacio de estados.
Para entender mejor cómo funciona esta representación del sistema en la localización del robot se hará una explicación detallada paso a paso.
Un robot móvil se encuentra en un espacio desconocido empezando desde una localización con coordenadas conocidas x0.
El móvil se desplaza a lo largo del tiempo ‘t’, generando una secuencia de localizaciones del robot ‘Xt’. Esta secuencia proporciona su trayectoria (estados).
Mientras el robot se desplaza adquiere (entradas) mediciones de la odometría (ut) y observaciones del entorno (Zt), que es la información que establece el robot a través de las mediciones entre las características de ‘m’ y ‘xt’.
Se denomina ‘m’ el mapa real del entorno. El entorno se compone de landmarks, objetos, superficies, etc.., y ‘m’ describe sus localizaciones. El mapa del entorno ‘m’ se asume que es invariante en el tiempo.

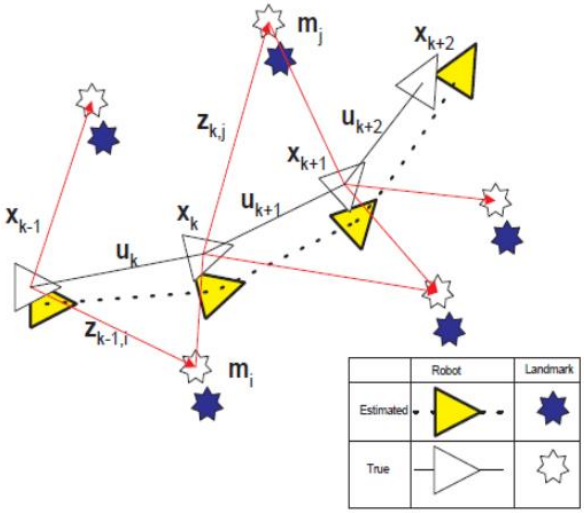


Figura 1-4. Representación del Espacio de Estados

Fuente: simultaneous localization and mapping (SLAM) problem [DWB06]

1.4.3 Filtro Bayesiano

En la siguiente figura, se muestra la relación entre controles, estados y mediciones. Sin profundizar en su desarrollo matemático, el filtro Bayesiano es un algoritmo que calcula y representa el impacto que la evidencia tiene en la hipótesis, es decir en la localización del robot.

Figura 1-5. Filtro Bayesiano

La pose de un robot se debe a su estado actual xt. Cuando en el sistema ingresa un nuevo estado xt+1, que se debe a algún comando de control del robot ut, el nuevo estado producirá una medida zt+1. La relación entre controles, estados y las medidas se muestran en la figura a continuación. Un filtro Bayes es un algoritmo que calcula la función de densidad de probabilidad (PDF) para el vector de estado xt.

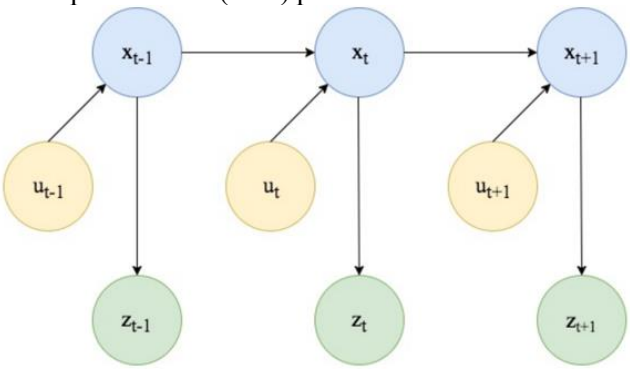


Figura 1-6. Relación entre Estados, Controles y Medidas

El filtro extendido de Kalman es una técnica que implementa el filtro Bayesiano, que a diferencia del filtro de Kalman también está destinado a la estimación de sistemas no lineales, debido a que linealiza el sistema mediante series de Taylor.

1.4.4 Filtro Extendido de Kalman (EKF)

El EKF (Extended Kalman Filter) se encarga de estimar la posición del robot a través de la odometría y los landmarks. Mediante una etapa de predicción en la que solo se tiene en cuenta la odometría, y una etapa de corrección en la que se utilizan los landmarks y las distribuciones de los errores, se haya la nueva posición.

Figura 1-7. Funcionamiento del Filtro de Kalman Extendido

1.5 ROS

ROS es un framework que simplifica la tarea de escribir programas complejos para robots. Con ROS, es posible tomar rápidamente programas de última generación para muchos aspectos del sistema, lo que libera tiempo del I+D para desarrollar los aspectos novedosos de un proyecto en particular" - Ph.D. Morgan Quigley, desarrollador de la primera versión de ROS

Cuando comenzamos a leer respecto a ROS, nos encontramos con el siguiente lema:

"No reinventes la rueda. ¡Cree algo nuevo y hágalo más rápido y mejor construyendo sobre ROS!"
(Don't reinvent the wheel. Create something new and do it faster and better by building on ROS!)

Figura 1-8. No reinventar la rueda
Fuente: https://www.sinologic.net/2021-11/la-importancia-de-no-reinventar-la-rueda.html

El objetivo fundamental de ROS, es justamente eso, dejar tiempo para lo importante.

Construir un robot es difícil, motores, los sensores, el software y las baterías deben funcionar todos juntos sin problemas para realizar una tarea.

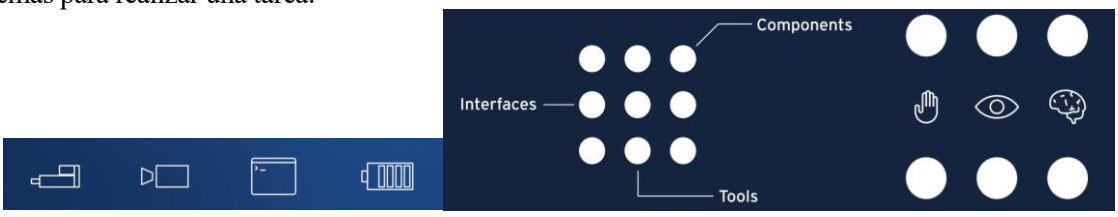


Figura 1-9. ROS – Introducción

Fuente: Página oficial ROS

Pero hay una manera de simplificar el proceso: El sistema operativo del robot en ROS, es un software libre y de código abierto, que define los componentes, las interfaces y las herramientas para construir robots avanzados.
La mayoría de los robots están hechos de actuadores: cosas que se mueven, sensores: cosas que leen el mundo y sistemas de control: o el cerebro de los robots.
Ros ayuda a los desarrolladores a construir rápidamente estos componentes y luego conectarlos fácilmente usando herramientas de ros llamadas temas y mensajes

estos mensajes
ROS es un acrónimo de Robot Operating System. A continuación, se expondrán aspectos básicos para la comprensión del funcionamiento general de ROS, no se intenta brindar los contenidos necesarios para aprender a programar o manejar ROS; solamente, brindar el material necesario para un análisis completo del proyecto que se está presentando.

1.5.1 ¿Qué es ROS?

ROS es un metasistema operativo de código abierto para robots. Proporciona los servicios que esperaría de un sistema operativo, incluida la abstracción de hardware, el control de dispositivos de bajo nivel, la implementación de funciones de uso común, el paso de mensajes entre procesos y la gestión de paquetes. También proporciona herramientas y bibliotecas para obtener, construir, escribir y ejecutar código en varias computadoras. ROS es similar en algunos aspectos a los "marcos de trabajo de robots", como Player, YARP, Orocos, CARMEN, Orca, MOOS y Microsoft Robotics Studio.
ROS es una red de procesos de igual a igual (peer-to-peer) (potencialmente distribuidos entre máquinas) que se acoplan libremente mediante la infraestructura de comunicación de ROS. ROS implementa varios estilos diferentes de comunicación, incluida la comunicación sincrónica de estilo RPC sobre servicios, transmisión asincrónica de datos sobre temas (topics) y almacenamiento de datos en un Servidor de Parámetros.

1.5.2 Ros Objetivos

El objetivo de ROS no es ser un marco con la mayoría de las funciones. En cambio, el objetivo principal de ROS es admitir la reutilización de código en la investigación y el desarrollo de robótica. ROS es un marco distribuido de procesos (también conocido como Nodos) que permite que los ejecutables se diseñen individualmente y se acoplen libremente en tiempo de ejecución. Estos procesos se pueden agrupar en paquetes y pilas, que se pueden compartir y distribuir fácilmente. ROS también admite un sistema federado de repositorios de código que también permite la distribución de la colaboración. Este diseño, desde el nivel del sistema de archivos hasta el nivel de la comunidad, permite tomar decisiones independientes sobre el desarrollo y la implementación, pero todo se puede unir con las herramientas de infraestructura de ROS.

En apoyo de este objetivo principal de compartir y colaborar, existen varios otros objetivos del marco ROS:

- Delgado (Thin): ROS está diseñado para ser lo más delgado posible, de tal forma que el código escrito para ROS se pueda usar con otros marcos de software de robots.
- Bibliotecas independientes de ROS: el modelo de desarrollo preferido es escribir bibliotecas independientes de ROS con interfaces funcionales limpias.
- Independencia del lenguaje: el marco ROS es fácil de implementar en cualquier lenguaje de programación moderno. (Python, C++, etc.).
- Pruebas sencillas: ROS tiene un marco de prueba de unidad/integración incorporado llamado rostest que facilita la activación y desactivación de dispositivos de prueba.
- Escalado: ROS es apropiado para grandes sistemas de tiempo de ejecución y para grandes procesos de desarrollo.

2 FORMULACIÓN DEL PROYECTO

2.1 Requisitos técnicos, económicos y alcance

2.1.1 Requisitos Técnicos

- Conocimientos en lenguajes de programación: Arduino, C++, Python, Processing Code.
- Conocimientos sobre SO UBUNTU
- Estudio y comprensión de ROS (Robot Operating System)
- Estudio y comprensión de Algoritmos SLAM
- Conocimientos sobre protocolos de comunicación, principalmente RS232 y orientados a comunicación por tópicos (MQTT o similares).
- Estudio y comprensión del funcionamiento de distintos tipos de sensores orientados a la robótica
- Comprensión de teoría de la cinemática de vehítulos robotizados
- Conocimiento y operación de instrumental electrónico, multímetro, osciloscopio, generador de funciones.
- Electrónica de baja potencia.
- Técnicas para el armado, prueba, calibración y ajuste de dispositivos orientados a la robótica.
- Configuración e implementación de redes WiFi.
- Conocimientos de Software de acceso remoto multiplataforma: Team Viewer, VNC.

2.1.2 Requisitos Económicos

Recursos Humanos Utilizados

#	Cantidad	Descripción	Monto
1	1	Recurso Humano	$0.00
		TOTAL	**$0.00**

A continuación, se detalla en una tabla la totalidad de de bienes de consumo utilizados en el prototipo final (no se consideraron los elementos utilizados en prototipos previos y que fueron descartados por diversas razones).

#	Cantidad	Descripción	Valor Unitario	Total (USD)
1	1	Sensor Ydlidar X2L escáner láser de 360°	$266.10	$266.10
2	1	Raspberry Pi 4 Model B 8gb	$424.57	$424.57
3	1	Disipador Aluminio Dual Fan Raspberry Pi 4	$16.94	$16.94
4	1	Memoria MicroSD 128GB	$25.42	$25.42
5	2	Bases Chasis Auto, Acrílicas cortadas con Laser	$8.07	$16.15
6	1	Kit 4 motores con ruedas y encoders plásticos	$18.97	$18.97
7	1	Mega2560 R3	$29.14	$29.14
8	1	Motor Shield Driver L293d	$4.06	$4.06
9	1	Cargador Portatil - Powerbank - 10000mah (Dash)	$12.71	$12.71
10	1	Cargador Portatil - Powerbank - 12000mah (FoxBox)	$42.37	$42.37
11	1	Cargador Portatil - Powerbank (TrV)	$7.63	$7.63
12	1	Mini Voltimetro Panel Digital Dc Azul	$4.14	$4.14
13	2	Sensor Optico Tipo Ranura Encoder	$2.87	$5.75
14	1	Sensor Gy-9250 9-axis, Acel+girosc+campo Mag (GY-91)	$14.03	$14.03
15	1	Sensor 10DOF L3GD20 LSM303D BMP180 Gyro Accelerometer	$40.09	$40.09
16	1	Cables Fichas Conectores, varios	$16.95	$16.95
		TOTAL		**$945.04**

Nota: Los valores están unificados en dólares estadounidenes, al tipo de cambio del Banco Nación, a la fecha de compra de cada elemento. En caso de corresponder, el flete, se encuentra incluido en dichos valores.

2.1.3 Alcance

El estudio de las nuevas tecnologías orientadas a vehículos autónomos, tiene una gran limitación práctica. La falta de plataformas de ensayo y los costos de los elementos específicos, nos limitan a un entorno prácticamente teórico y muy limitado.
El presente prototipo pretende ser una herramienta para implementar las fases prácticas para el estudio, ensayo y comprensión de vehículos autónomos dentro de espacios cerrados.
El vehículo presentado, provee una plataforma funcional con CPU, sensores, motores, baterías y comunicación. Dicho vehículo tiene instalado un sistema ROS (Robot Operating System) Kinetic con las herramientas necesarias para este tipo de desarrollos.
El sensor LiDaR de 360 grados es un sensor de avanzada que permite mapear entornos con un scaner laser. El vehículo posee un sensor de este tipo implementado.
El vehículo presentado, podrá mapear con su escaner laser, la planta completa de un edificio, guardar dicho mapa. El vehículo podrá ser controlado remotamente (joystick inalámbrico, teclado de PC remota, mouse) o trabajar de modo autónomo mediante soft instalado.

Este proyecto esá orientado a fines educativos y pretende ser la base de futuros desarrollos aplicables a nivel industrial.

2.2 Planificación: metodología de trabajo, etapas, plazos, recursos humanos y materiales.

Planificación:

- Identificación de ámbito de aplicación.
- Necesidades.
- Definición de objetivos.

Análisis de situación:

• Tecnologías vinculantes en componentes electrónicos: comunicación, industrial, servicios, sensores, robótica, etc.
• Definición preliminar del producto a desarrollar.
• Financiación.

Desarrollo conceptual:

• Diseños orientativos y cálculos preliminares
• Analizar diseño y construcción de esquemas electrónicos funcionales con tecnología de última generación.
• Estudios previos de materiales e insumos. Logística de los mismos.
• Documentación

Diseño Preliminar:

• Diseño de un cabezal para un sensor de rango, SONAR, con tecnología de ultrasonido.
• Montajes mecánicos y electrónicos con tecnología actual.
• Pruebas, mediciones y ensayos.
• Documentación

Diseño Definitivo:

• Prototipo funcional con tecnología actual.
• Correcciones.
• Ensayos y pruebas finales de laboratorio y de campo

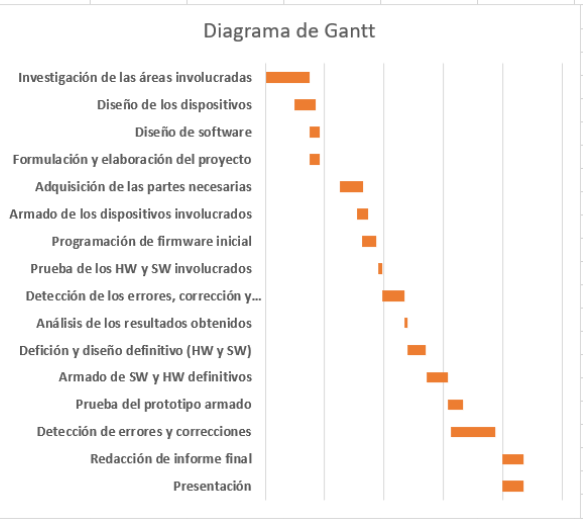
Evaluación técnica, económica y financiera:

• Análisis de costos
• Documentación Final.
• Conclusiones.

El proyecto consta de las siguientes etapas:

El tiempo total para el desarrollo del proyecto final ha sido cercano a los 6 meses, desde noviembre de 2021 hasta abril de 2022.

CRONOGRAMA DE ACTIVIDADES	INICIO	DURACIÓN	FINALIZACIÓN
PROYECTO FINAL	**02/11/2021**	174	**25/04/2022**
FASE 1: Investicación y diseño	02/11/2021	66	07/01/2022
Investigación de las áreas involucradas	02/11/2021	30	02/12/2021
Diseño de los dispositivos	22/11/2021	14	06/12/2021
Diseño de software	02/12/2021	7	09/12/2021
Formulación y elaboración del proyecto	02/12/2021	7	09/12/2021
Adquisición de las partes necesarias	22/12/2021	16	07/01/2022
FASE 2: Prototipo y prueba de concepto	03/01/2022	46	18/02/2022
Armado de los dispositivos involucrados	03/01/2022	7	10/01/2022
Programación de firmware inicial	06/01/2022	10	16/01/2022
Prueba de los HW y SW involucrados	17/01/2022	3	20/01/2022
Detección de los errores, corrección y mejoras	20/01/2022	15	04/02/2022
Análisis de los resultados obtenidos	04/02/2022	2	06/02/2022
Defición y diseño definitivo (HW y SW)	06/02/2022	12	18/02/2022
FASE 3: Implementación y puesta a punto	19/02/2022	46	06/04/2022
Armado de SW y HW definitivos	19/02/2022	14	05/03/2022
Prueba del prototipo armado	05/03/2022	10	15/03/2022
Detección de errores y correcciones	07/03/2022	30	06/04/2022
FASE 4: Presentación	11/04/2022	14	25/04/2022
Redacción de informe final	11/04/2022	14	25/04/2022
Presentación	11/04/2022	14	25/04/2022



3 EVALUACIÓN DEL PROYECTO

3.1 Evaluación económica – financiera de los recursos necesarios para la realización del proyecto en su etapa de pre diseño, diseño y construcción a nivel prototipo o referencial.

3.1.1 Bienes de Consumo

A continuación, se detalla en una tabla la totalidad de bienes de consumo utilizados, el monto por unidad, la cantidad y el monto total.

#	Cantidad	Descripción	Valor Unitario	Total (USD)
1	1	Sensor Ydlidar X2L escáner láser de 360°	$266.10	$266.10
2	1	Raspberry Pi 4 Model B 8gb	$424.57	$424.57
3	1	Disipador Aluminio Dual Fan Raspberry Pi 4	$16.94	$16.94
4	1	Memoria MicroSD 128GB	$25.42	$25.42
5	2	Bases Chasis Auto, Acrílicas cortadas con Laser	$8.07	$16.15
6	1	Kit 4 motores con ruedas y encoders plásticos	$18.97	$18.97
7	1	Mega2560 R3	$29.14	$29.14
8	1	Motor Shield Driver L293d	$4.06	$4.06
9	1	Cargador Portatil - Powerbank - 10000mah (Dash)	$12.71	$12.71
10	1	Cargador Portatil - Powerbank - 12000mah (FoxBox)	$42.37	$42.37
11	1	Cargador Portatil - Powerbank (TrV)	$7.63	$7.63
12	1	Mini Voltimetro Panel Digital Dc Azul	$4.14	$4.14
13	2	Sensor Optico Tipo Ranura Encoder	$2.87	$5.75
14	1	Sensor Gy-9250 9-axis, Acel+girosc+campo Mag (GY-91)	$14.03	$14.03
15	1	Sensor 10DOF L3GD20 LSM303D BMP180 Gyro Accelerometer	$40.09	$40.09
16	1	Cables Fichas Conectores, varios	$16.95	$16.95
		TOTAL		**$945.04**

Nota: Los valores están unificados en dólares estadounidenes, al tipo de cambio del Banco Nación, a la fecha de compra de cada elemento. En caso de corresponder, el flete, se encuentra incluido en dichos valores.

3.1.2 Recursos Humanos

Para la producción planteada se toma la siguiente estructura de recursos humanos

Estructura de sueldos	Hs	$ (ARS)	Total
Gerente de fabrica	160	$1,200.00	$192,000.00
Tecnicos / operario de producción	240	$850.00	$204,000.00
Encargado de comercialización	80	$550.00	$44,000.00
Sector administrativo	160	$500.00	$80,000.00
Encargado de RRHH	80	$550.00	$44,000.00
Total			**$564,000.00**

Gastos fijos		$ (ARS)	Total
Alquileres	1	$45,000.00	$45,000.00
Servicios varios	1	$15,000.00	$15,000.00
Total			**$60,000.00**

3.2 Evaluación económica-financiera de los recursos necesarios para la producción y comercialización del proyecto

3.2.1 Inversión Inicial año 0

Para la puesta en marcha de la producción se considera los gastos iniciales de

#	Concepto	Cantidad	Precio (AR$)	Total (AR$)
1	Equipamento Informático	1	$400,000	$400,000
2	Stock Inicial	17	$113,050	$1,921,850
3	Mobiliario	1	$80,000	$80,000
4	Gastos de Alquileres	1	$100,000	$100,000
	Total			**$2,501,850**

El advenimiento de la Industria 4.0 y la creciente necesidad de actualización del parque industrial, permitiría inferir, que el producto va a tener una captación exitosa. Y considerando que prácticamtne no existen competidores directos a nivel nacional, considero que existe una demanda de absorción de al menos **11 unidades mensuales**.
Considerando como referencia productos similares, cuyo precio final (incluyendo, compra, traslado y nacionalización) colocados en el país, se podría estimar en USD2000 dólares oficiales (aproximadamente 1200 dólares billete). Se podría estimar su **precio de venta en $177.300**. Obviamente se están considerando las condiciones actuales para la importación y la situación especial respecto a las distintas cotizaciones del dólar.

Dolar oficial	$119.00
Dolar turista	$197.00

3.2.2 Flujo de caja de forma anual

Primeramente, se debe anualizar producción y costos:

Producción estimada		
Producción mensual estimada	UN	11
Producción anual	UN	132

Costos variables		
Costo unitario materiales		$112,574.00
Costo de materiales anual		$14,859,768.00

Inversión inicial		
Inversión inicial		$2,501,850.00

Costos fijos		
Sueldos mensual		$564,000.00
Sueldos anual		**$6,768,000.00**
Costo fijo mensual		$60,000.00
Costo fijo anual		**$720,000.00**

Ingresos por ventas esperado		
Precio de venta unitario		**$177,300.00**
Ingreso anual		**$23,403,600.00**

Flujo de caja	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
1 Detalle de los ingresos						
1.1 Ingreso por venta	0	$23,403,600.00	$23,403,600.00	$23,403,600.00	$23,403,600.00	$23,403,600.00
2 Detalle de los egresos						
2.2 Costos variables	$0.00	$14,859,768.00	$14,859,768.00	$14,859,768.00	$14,859,768.00	$14,859,768.00
2.3 Costos fijos	$0.00	$7,488,000.00	$7,488,000.00	$7,488,000.00	$7,488,000.00	$7,488,000.00
2.4 Inversión inicial	$2,501,850.00	$0.00	$0.00	$0.00	$0.00	$0.00
Saldo neto	-$2,501,850.00	$1,055,832.00	$1,055,832.00	$1,055,832.00	$1,055,832.00	$1,055,832.00

3.2.3 Cálculo de VAN

El VAN o valor actual neto es una herramienta financiera para evaluar la rentabilidad de un proyecto de inversión, entendiéndose por proyecto de inversión no sólo como la creación de un nuevo negocio, sino también, como inversiones que podemos hacer en un negocio en marcha, tales como el desarrollo de un nuevo producto, la adquisición de nueva maquinaria, el ingreso en un nuevo rubro de negocio, etc. El VAN también nos permite determinar cuál proyecto es el más rentable entre varias opciones de inversión. Incluso, si alguien nos ofrece comprar nuestro negocio, con este indicador podemos determinar si el precio ofrecido está por encima o por debajo de lo que ganaríamos de no venderlo. Según el resultado arrojado por el VAN podemos clasificar un proyecto en:

• VAN > 0 → el proyecto es rentable.
• VAN = 0 → el proyecto es rentable también, porque ya está incorporado ganancia de la TD.
• VAN < 0 → el proyecto no es rentable.

Para el cálculo de VAN, se necesita una tasa de descuento, como el flujo no considera inflación, voy a tomar una tasa de descuento real, la cual será la tasa libre de riesgo del Treasure Bills (o bono de EEUU a diez años). Para la cual se tomó **2,56% anual**.

Por último, para nuestro cálculo de VAN, se aplica la siguiente formula, en una panilla de Excel, con los datos obtenidos en los cuadros anteriores,

$$\mathrm{VAN} = \sum_{t=1}^{n} \frac{V_t}{(1+k)^t} - I_0$$

VAN = $ 2.394.903,58

Como VAN es positiva, se puede considerar que es un proyecto rentable contra un proyecto basado en la tasa libre de riesgo considerada.

El VAN junto con otra herramienta TIR (tasa interna de retorno) trabajan en conjunto para determinar con mayor certeza la rentabilidad de un proyecto.

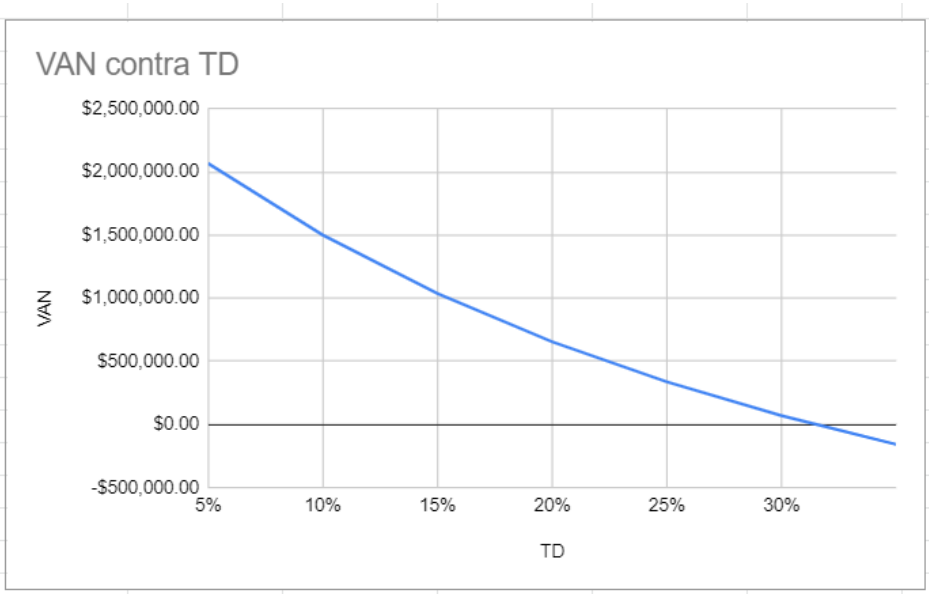
3.2.4 Cálculo de TIR

Para encontrar el valor del TIR, tasa interna del retorno del proyecto, se puede hacer de dos maneras:

- **Mediante hoja de cálculo:** Excel provee una función que puede calcular TIR, utilizando los cuadros y datos anteriores
- **Mediante iteraciones:** se procede al cálculo del VAN, para distintos valores de la tasa de interés, hasta que alguna haga el VAN=0.

Utilizando el segundo método, obtenemos los siguientes resultados:

TD	VAN
5%	$2,069,350.01
10%	$1,500,583.98
15%	$1,037,462.62
20%	$655,734.00
25%	$337,577.88
30%	$69,702.48
35%	-$157,943.69



TIR = 31%

4 DESARROLLO DEL DISPOSITIVO ELECTRÓNICO

4.1 Descripción del proyecto

Se diseñará y se construirá un vehículo autónomo a escala, orientado al estudio de las nuevas tecnologías basadas en ROS, algoritmos SLAM y mapeos mediante LiDAR. El presente trabajo pretende ampliar las fronteras que imponen la falta de recursos para hacer prácticas sobre estas nuevas tecnologías. También pretende ser la base de un trabajo evolutivo para el diseño y fabricación de este tipo de dispositivos, para su empleo en la industria.

El vehículo se lo podría describir en dos niveles:

- Nivel base: Este nivel es prácticamente se lo podría describir como un vehículo a control remoto. Posee una plataforma con cuatro ruedas, motores, encoders, una CPU basada en ATMega2560, un driver para los cuatro motores, un sensor IMU, un receptor para un joystick tipo PS2 y conjunto de baterías tipo PowerBank (Por razones constructivas, las baterías mencionadas, más adelante, tuvieron que colocarse en el nivel superior)
- Nivel superior: Este nivel es el corazón del proyecto, posee los "ojos", el "cerebro" y la conectividad del vehículo. El vehículo percibe su alrededor mediante un escaner laser 2D de 360 grados, tipo YDLIDAR modelo X2L. La CPU principal es una Raspberry Pi 4B, con un procesador Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz, 8GB de SRAM y 128GB de memoria microSD. Además posee conectividad 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE y Gigabit Ethernet. Además, tiene instaladas todas las herramientas de software que provee el SO ROS Kinetic.

La idea, es contar con un dispositivo móvil, con el cual recorrer la planta de un edificio mientras el dispositivo lo mapea mediante laser. Dicho mapa se almacena y permite luego que el operador pueda definir recorridos sobre dicho mapa, para que el vehículo los realice de manera autónoma.

4.2 Descripción del diseño del dispositivo electrónico. Tecnología de los elementos utilizados

Para el diseño del dispositivo existen dos grandes "cuellos de botella" que implican la utilización elementos con un grado de complejidad y un costo relativamente elevado. Estos elementos son: la CPU principal y el escaner LiDAR.

4.2.1 CPU Principal

Para poder instalar el SO ROS Kinetic, es necesario tener contar con un sistema operativo de base. En este caso se eligió UBUNTU, que es una es una distribución GNU/Linux, gratuita, que ofrece un interesante sistema operativo para equipos de escritorio y servidores en el ámbito educativo. Es una distribución basada en Debian cuyas principales características son: Facilidad de manejo. Actualizaciones frecuentes.
De lo enunciado, se puede inferir que la CPU principal requiere de una serie de especificaciones mínimas similares a las de un PC de escritorio, lo cual hace que su fabricación sea inviable.
Para este prototipo, se decidió adquirir un ordenador de placa reducida, modelo RaspBerry Pi 4B, por ser un dispositivo que se consigue en el mercado local y que cumple con las especificaciones mínimas del proyecto. Además, dicha plataforma nos permitirá, en futuros desarrollos, la implementación de cámaras para el reconocimiento del entorno.
Estas son las especificaciones que podemos destacar de la RaspBerry seleccionada:

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 8GB LPDDR4-3200 SDRAM

- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI ports (up to 4kp60 supported)
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- Micro-SD card slot for loading operating system and data storage. En este caso se instaló una MicroSD de 128 GB
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)

4.2.2 Sensores de Rango

Para el presente proyecto, se utilizará algoritmos SLAM (Simultaneous Localization And Mapping). Para poder hacer un mapeo, se utilizan sensores de rango (Sonar, Lidar o cámaras).
Los sensores de rango son los dispositivos que permiten al robot detectar los posibles obstáculos del entorno en el que debe operar. También pueden ser usados para la medición de distancias. Existen muchos tipos de sensores de rango para la percepción del entorno, como son:
- Sensores basados en ondas acústicas: SONAR
- Sensores basados en señales electromagnéticas: infrarrojos, RADAR.
- Sensores basados en luz coherente: LiDAR.

4.2.2.1 Sensor de Rango Tipo SONAR

El sonar (acrónimo de Sound Navigation And Ranging, 'navegación por sonido') es una técnica que usa la propagación del sonido para detectar objetos y medir distancias.

4.2.2.1.1 SONAR: Principio de Funcionamiento

El transmisor emite una onda ultrasónica direccional cuando es activado y comienza un temporizador. Los pulsos ultrasónicos viajan hacia afuera hasta que se encuentran un objeto, el objeto hace que la onda se refleje hacia la unidad. El receptor ultrasónico detectará la onda reflejada y detendrá el temporizador.

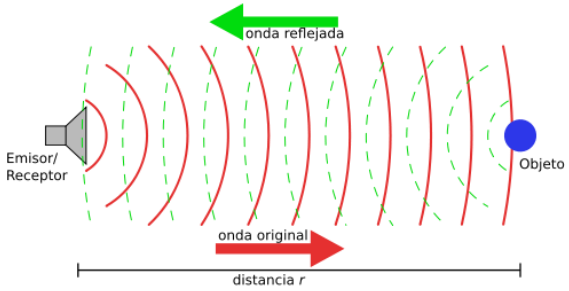


Figura 4-1. Medición de distancias mediante SONAR o RADAR.
Fuente: Principle of a sonar or radar distance measurement Georg Wiora (Dr. Schorsch)

Principle of a sonar or radar distance measurement Georg Wiora (Dr. Schorsch)

$$V_s = \frac{D}{t}$$

Donde:

$\boldsymbol{V_s}$ es la velocidad del sonido. 340 $^{m}/_{seg}$ en el aire.
D es la distancia total recorrida (ida y vuelta)
t es el tiempo empleado para recorrer dicha distancia

Despejando y considerando que **D** representa el doble de la distancia al objeto (ida y vuelta), tenemos:

$$\frac{D}{2} = D_r = \frac{V_s * t}{2}$$

Donde:

$\boldsymbol{D_r}$ es la distancia al objeto.

Por lo tanto, disponemos de toda la información necesaria para calcular la distancia al objeto D_r: el tiempo **t** determinado por el temporizador y la velocidad V_s, que es la velocidad del sonido en el aire, 340 $^{m}/_{seg}$.

4.2.2.1.2 El sensor HC SR104

El HC SR104 es un módulo ultrasónico económico, que consta de un transmisor, un receptor y una lógica de control. Su rango de medición de distancia es de 2cm a 450cm aproximadamente (en la práctica, se utilizan hasta los 200cm).

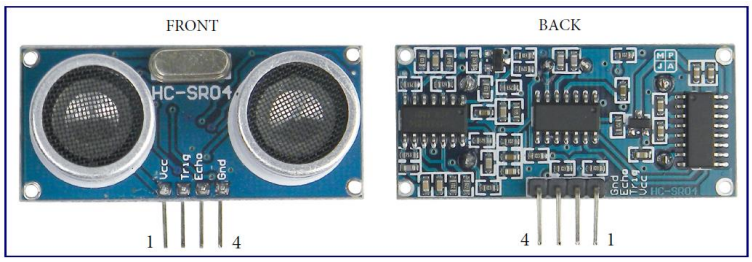


Figura 4-2. Sensor HC SR104.

Fuente HC SR104 User Guide

El principio de funcionamiento es el que fue descripto en el apartado anterior. Comienza con un pulso de disparo de 10 uSeg en el Pin Trigger, el transmisor emite una ráfaga de 8 pulsos ultrasónicos direccionales de 40 KHz y comienza un temporizador hasta recibir el rebote de la señal en algún objeto.

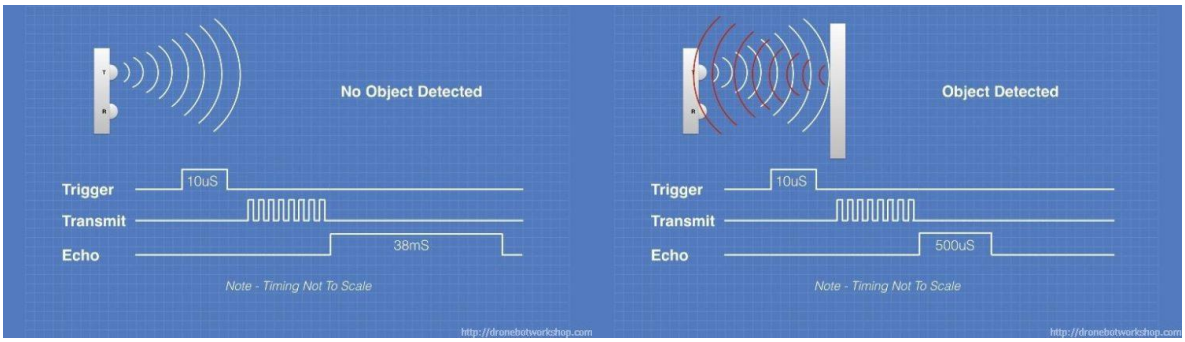


Figura 4-3. Sensor HC SR104 Detección de Objetos.

Su funcionamiento no se vé afectado por la luz solar o material negro como los telémetros de Sharp (aunque los materiales acústicamente suaves como la tela pueden ser difícil de detectar).
La siguiente figura resume las características principales:

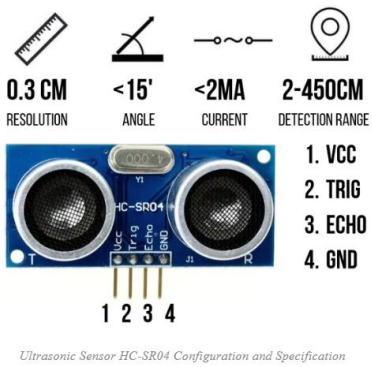


Figura 4-4. Sensor HC SR104 - Características.

Consideraciones:

- La velocidad del sonido es variable: $\boldsymbol{V_s}$ es la velocidad del sonido en el aire, 340 $^{m}/_{seg}$. Pero dicha velocidad depende de la temperatura T y se calcula aproximadamente:

$$V_s = V_{s_0} + 0{,}6\ T\ {}^{m}/_{S}$$

- Atenuación
- Tiempo en blanco
- Ángulo de medida: Es importante tener en cuenta que el sensor va a detectar objetos que se encuentren frente al mismo dentro de un determinado ángulo de cobertura. Las especificaciones mencionan un ángulo efectivo menor a 15 grados y un ángulo de medición de 30 grados.

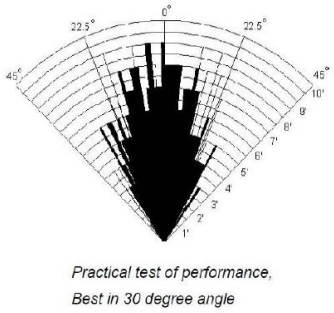


Figura 4-5. Sensor HC SR104 - Ángulo de medida.

- Reflexiones: En la siguiente figura se representa el efecto de la reflexión.

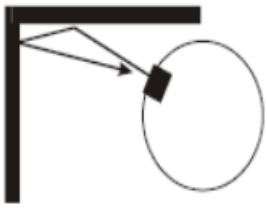


Figura 4-6. Sensor HC SR104 – Efecto Reflexión.

- Cross Talk: La figura representa el efecto Cross Talk

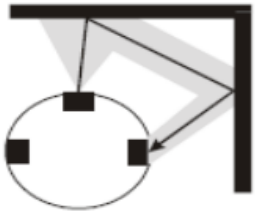


Figura 4-7. Sensor HC SR104 - Efecto Cross Talk.

Primeramente, se desarrolló, contruyó y ensayó un protipo de sensor de rango, tipo SONAR, basado en cuatro sensores ultrasónicos dispuestos a 90 grados sobre una base móvil acoplada a un servomotor.

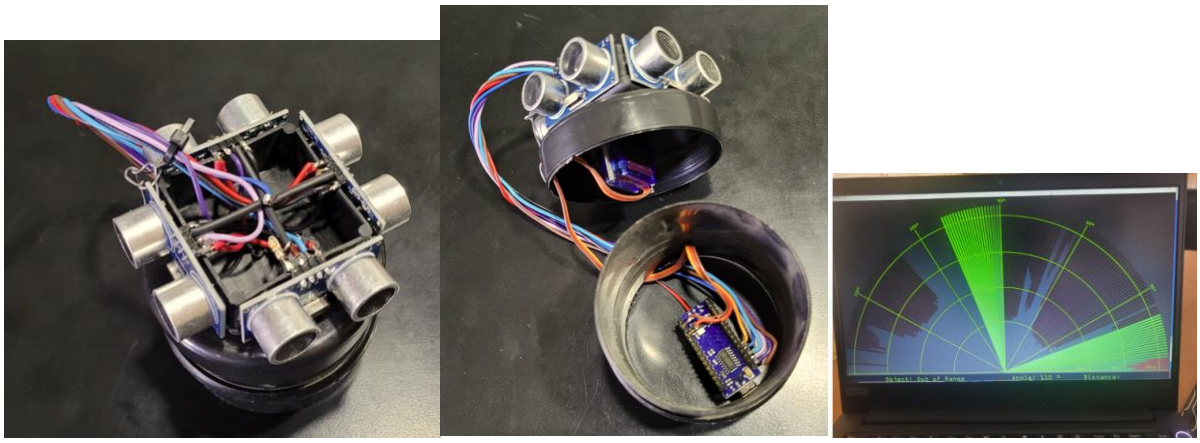
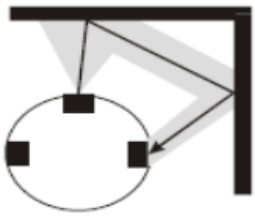


Figura 4-8. Sonar con 4 sensores desarrollado y construido.

También se inició un software basado en Processing Code, para visualización en 360 grados, el cual está en etapa de desarrollo.
Se esperaba un resultado económico y de prestaciones aceptables, pero se tuvo que descartar, en esta etapa del proyecto, por dos motivos:

- Implementación de un procolo de comunicación con ROS basado en tópicos. En la primera etapa del desarrollo, no se disponía de un entorno funcional para pruebas y ensayos. Eso dificultaba demasiado la implementación de una adecuada comunicación entre el dispositivo y el ROS. Actualmente, al disponer de un entorno funcional, sería posible retomar el desarrollo del sensor.

- Errores en mediciones resultantes del efecto Cross Talk. Es posible solucionarlo, a futuro, mediante un replanteo en el conexionado de los sensores y modificaciones en el soft que los comanda



Aunque este sensor fue descartado del modelo final, me parece interesante tenerlo en cuenta, considerando los costos de un sensor LiDAR o lo complejo del análisis de imágenes obtenidas por cámara para un mapeo del entorno.

4.2.2.2 Sensor de Rango Tipo LiDAR

Un LiDAR (acrónimo del inglés LiDAR, Light Detection and Ranging o Laser Imaging Detection and Ranging) es un dispositivo que permite determinar la distancia desde un emisor láser a un objeto o superficie utilizando un haz láser pulsado. La distancia al objeto se determina midiendo el tiempo de retraso entre la emisión del pulso y su detección a través de la señal reflejada.

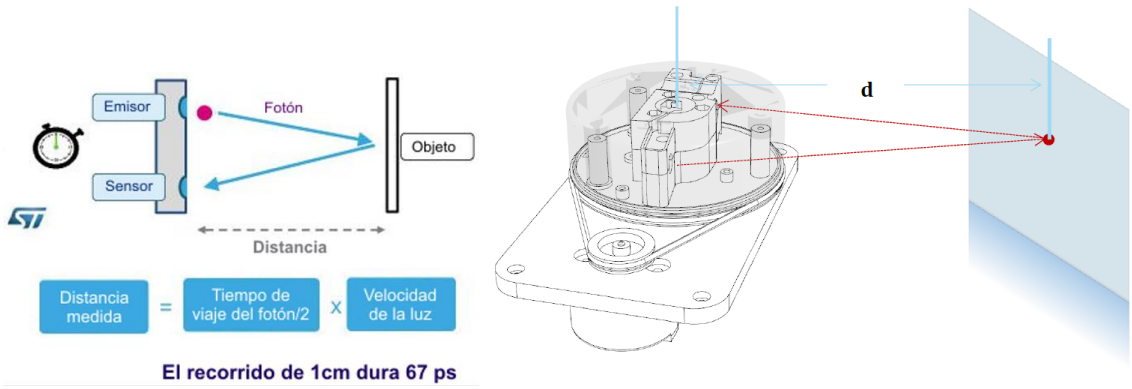


Figura 4-9. Sensor LiDAR – Principio de Funcionamiento.

Fuente: The RPLIDAR A1 Working Schematic (pdf LD108)

Un escaner LiDAR 2D de 360 grados, es un dispositivo al que se le ha incorporado una base giratoria, acoplada a un motor, lo que le permite hacer un barrido de todo el entorno circundante (los 360 grados).

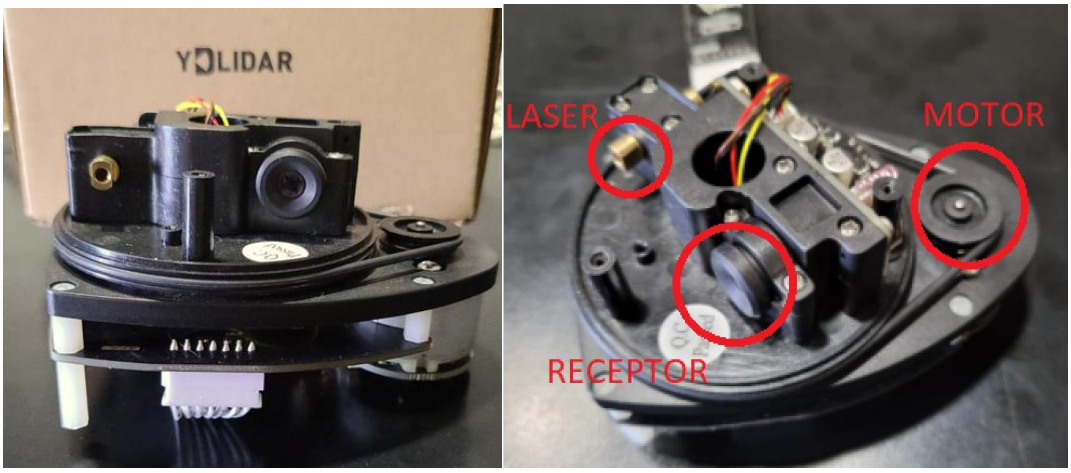


Figura 4-10. Sensor RPLiDAR – Laser, Receptor y Motor.

A lo largo del desarrollo he utilizado dos modelos de LiDAR:

- RPLiDAR, 360° Laser Scanner

Este dispositivo se utilizó durante la mayor parte del proyecto, pero lamentablemente comenzó a presentar lecturas erróneas debidas a una falla mecánica que no se logró subsanar todavía.

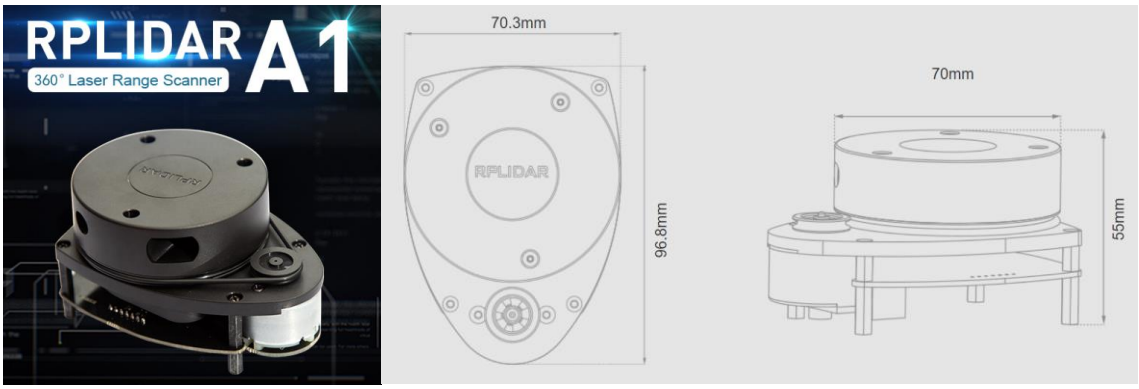


Figura 4-11. Sensor RPLiDAR – Dimensiones.

Fuente: https://www.slamtec.com/en/Lidar/A1Spec

- Fabricante: SlamTec, Modelo A1M1-R3
- Measuring Range 0.15m - 12m
- Sampling Frequency 8K
- Rotational Speed 5.5Hz
- Angular Resolution ≤1°
- Dimensions 96.8 x 70.3 x 55mm
- System Voltage 5V
- System Current 100 mA
- Power Consumption 0.5W
- Output UART Serial (3.3 voltage level)
- Temperature Range 0°C-40°C
- Angular Range 360°
- Range Resolution
  - ≤1% of the range （≤12m)
  - ≤2% of the range （12m ~ 16m)
- Accuracy
  - 1% of the range （≤3 m)
  - 2% of the range （3-5 m)
  - 2.5% of the range （5-25m)

- YDLiDAR

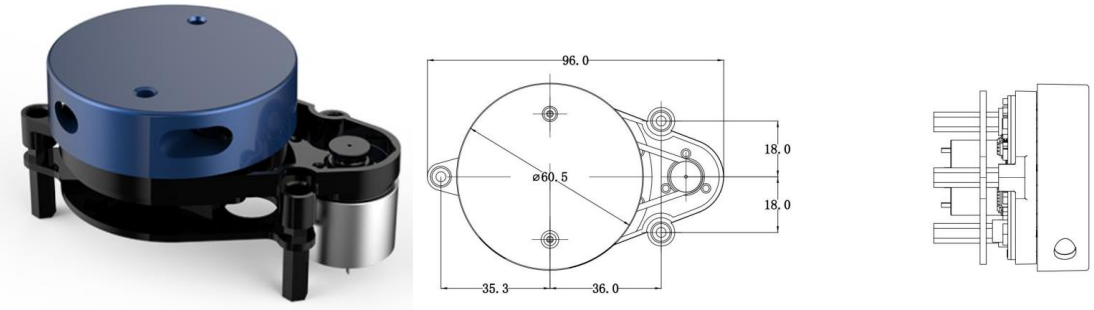


Figura 4-12. Sensor YDLiDAR X2 – Dimensiones.
Fuente: FIG2 YDLIDAR X2 MECHANICAL DIMENSIONS
https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/datasheet/application/lidarbot/YDLIDAR%20X2%20Datasheet.pdf

- Fabricante YDLIDAR, Modelo X2L
- Especificaciones:

Item	Min	Typical	Max	Unit	Remarks
Ranging frequency	-	3000	-	Hz	3000 times per second
Motor frequency	-	7	-	Hz	PWM or Voltage Regulation
Ranging distance	0.10	-	>8	m	Indoor
Scanning angle	-	0~360	-	Deg	-
Absolute error	-	2	-	cm	Distance≤0.5m
Relative error	-	1.5%	-	-	0.5m<Distance≤6m
	-	2.0%	-	-	6m<Distance≤8m
Angle resolution	0.82	0.84	0.86	Deg	Scanning frequency=7

- Especificaciones eléctricas:

Item	Min	Typical	Max	Unit	Remarks
Supply voltage	4.8	5	5.2	V	Excessive voltage might damage the Lidar while low affect normal performance
Voltage ripple	0	50	100	mV	Excessive ripple affect normal performance
Starting current	300	400	500	mA	Higher current required at start-up
Working current	200	350	380	mA	Normal working

4.2.3 Sensor IMU

Una unidad de medición inercial o IMU (del inglés inertial measurement unit), es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giróscopos.

4.2.3.1 Giróscopo

Los giroscopios han sido ampliamente utilizados con fines militares y civiles durante el último siglo. No solo para fines aeronáuticos sino también por la marina. Tradicionalmente, los giroscopos, solían ser mecánicos. Sin embargo, lo que se está utilizando hoy en día, en muchas aplicaciones, son los giroscopios MEMS (sistemas micro-electromecánicos), que se utilizan en teléfonos móviles, proyectos robóticos y con fines militares.

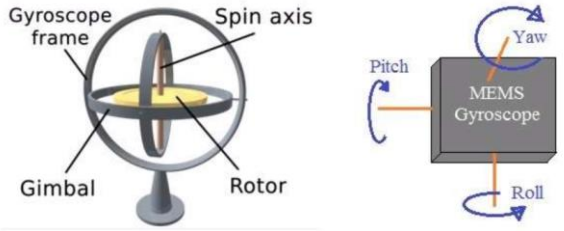


Figura 4-13. Giróscopo.
Fuente: https://www.rfwireless-world.com/Terminology/MEMS-Gyroscope.html

4.2.3.1.1 Giróscopo: Principio de funcionamiento

Los giroscopios en general pueden describirse mediante la fórmula de la fuerza de Coriolis.

$$F_c = 2 * m * (\omega * v)$$

En la ecuación, se observa que las fuerzas de Coriolis son dependientes de la masa **m** de un objeto, su velocidad angular **ω** como así también de su velocidad **v**. Este objeto mencionado, es una parte del girócopo y su masa, así como su velocidad, son conocidas. Dado que el objeto gira, y al tener información sobre la masa y la velocidad del objeto se puede detectar la velocidad angular. [14, p.173] Dada la velocidad angular, el ángulo total que ha girado el vehículo puede ser calculado como

$$a = \int_0^t \psi \, dt$$

4.2.3.1.2 Giróscopo Tridimensional

Un giroscopio MEMS también se basa en la variación de capacitancia entre el silicio y los elementos mecánicos, pero con esta configuración, el sensor genera cambios capacitivos con cambios de velocidad angular.

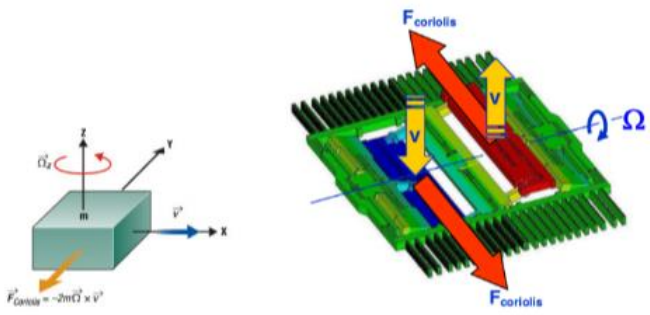


Figura 4-14. Giróscopo MEMS.

Fuente: https://www.researchgate.net/publication/308074268_Robust_smartphone_fingerprint_by_mixing_device_sensors_features_for_mobile_strong_authentication

Un giroscopio 3D tiene tres sensores giroscópicos montados ortogonalmente. Una medición de la fuerza-g se expresa en $^{m}/_{Seg^2}$, donde 1g es igual a la fuerza gravitacional de la tierra. El mecanismo de detección de los tres giroscopios es también capacitivo.

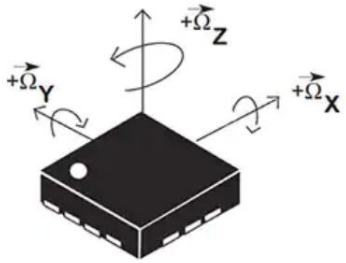


Figura 4-15. Giróscopo 3D.
Fuente: https://www.digikey.com/es/articles/apply-sensor-fusion-to-accelerometers-and-gyroscopes

Un giroscopio tridimensional proporciona datos de salida para la aceleración angular de giro alrededor de los ejes X, Y y Z.
Un giroscopio 3D mide la aceleración angular alrededor de los ejes X, Y y Z. Si la aceleración lineal se impone en un giroscopio, las distancias entre la placa fija interna y la masa permanecen invariables. Posteriormente, el giroscopio no responderá a una velocidad lineal.
Con este atributo, el giroscopio 3D es apropiado en aplicaciones tales como el control de movimiento, aparatos y la robótica. Sin embargo, la combinación de un giroscopio y un acelerómetro puede empezar a cumplir los requisitos de detección de un avión teledirigido.

4.2.3.2 Acelerómetro

Un acelerómetro actual (incluido en teléfonos inteligentes y tablets), está compuesto por un circuito de masa sísmica (compuesto por silicio) que cambia su posición de acuerdo con la orientación y está conectado al circuito del dispositivo. En realidad, un acelerómetro es un circuito basado en MEMS (Sistema Micro Electro Mecánico), que mide las fuerzas de aceleración que pueden ser causadas por la gravedad, por el movimiento o por la acción de basculamiento. Estas aceleraciones se miden en términos de fuerza g (m/s2) en los tres ejes (x,y,z).

4.2.3.2.1 Acelerómetro: Principio de Funcionamiento

Los acelerómetros basados en MEMS pueden consistir en condensadores diferenciales. La siguiente figura muestra la arquitectura interna de un acelerómetro basado en MEMS.

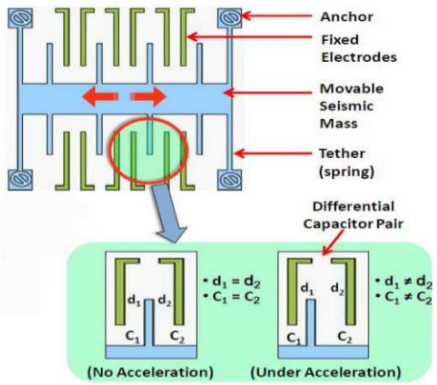


Figura 4-16. Acelerómetro MEMS.

Fuente: https://www.researchgate.net/publication/308074268_Robust_smartphone_fingerprint_by_mixing_device_sensors_features_for_mobile_strong_authentication

Como podemos ver hay varios pares de electrodos fijos y una masa sísmica móvil. Sin aceleración, las distancias d1 y d2 son iguales y, como resultado, los dos condensadores son iguales, pero un cambio en la aceleración hará

que la masa sísmica móvil se desplace más cerca de uno de los electrodos fijos provocando un cambio en la capacitancia generada. Esta diferencia de capacitancia se detecta y amplifica para producir un voltaje que es proporcional a la aceleración. La mínima imprecisión en la estructura electromecánica induce imperfecciones entre los chips del acelerómetro.
En ecuaciones:

$$F = m * a$$

Fuerza es igual a masa por aceleración. La masa es conocida y la fuerza se puede determinar de las constantes de los resortes (spring en la figura).

Por otro lado, la capacidad se determina por:

$$c = \frac{\varepsilon_0 * \varepsilon_r * L * W}{d}$$

Donde:

ε_0 es la constante dieléctrica del aire $8{,}85 * 10^{-12}$ Faradio/metro
ε_r es la constante dieléctrica del sustrato en relación al aire
L = longitud de la placa fija adyacente y la masa
W = espesor de la placa fija y la masa
d = separación entre placas fijas y la masa

4.2.3.2.2 Acelerómetors 3D

En un acelerómetro 3D, hay tres sensores de acelerómetro montados ortogonalmente

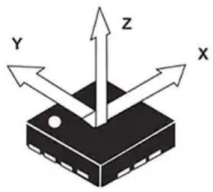


Figura 4-17. Acelerómetro 3D.

Un acelerómetro 3D proporciona datos de salida para las aceleraciones posicionales de los ejes Z, Y y Z. (Fuente de la imagen: STMicroelectronics)
El mecanismo de detección de los tres acelerómetros es también capacitivo.
Un acelerómetro 3D mide la aceleración lineal a lo largo de los ejes X, Y y Z. Bajo rotación, como en una bobina, las distancias entre la placa fija interna y la masa permanecen invariables. Posteriormente, el acelerómetro no responderá a una velocidad angular.
Con este atributo, el acelerómetro 3D es apropiado en aplicaciones tales como la detección de movimiento, el reconocimiento de gestos, la orientación de pantalla y la detección de caída libre. Sin embargo, sólo puede satisfacer una parte de las necesidades de detección de un avión teledirigido.

4.2.3.3 Magnetómetro

Magnetómetros MEMS: Los magnetómetros son sensores de medición de campo magnético y en este caso específico la del campo magnético terrestre.

4.2.3.3.1 Magnetómetro: Princpio de Funcionamiento

El principio de funcionamiento, es la medición a través de magneto-resistencias que cambian su valor en función del campo que las atraviesa en su dirección usando sensores AMR(Anisotropic Magneto-Resistive por sus siglas en ingles), este fenómeno se produce en materiales ferrosos y se trata de un cambio en la resistencia, cuando un campo magnético se aplica en una tira delgada de material ferroso llamada permaloy.

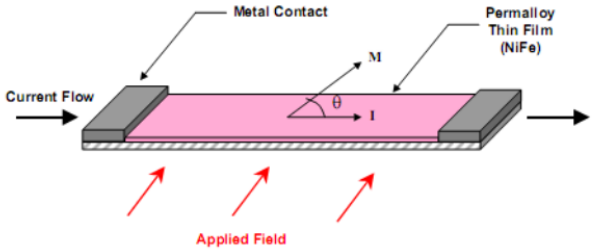


Figura 4-18. Magnetómetro – Principio de Funcionamiento.

Fuente: https://docplayer.es/5045507-Diseno-e-implementacion-del-control-de-estabilizacion-para-una-plataforma-altazimut-para-uso-en-fotografia-ing-mauricio-jorge-machado-buritica.html
HONEYWELL. HONEYWELL Magnetic Sensors. [Online].
http://www.magneticsensors.com/literature.php

Para su construcción, se utilizan cuatro elementos resistivos, de los mencionados anteriormente, para formar un sensor de puente de Wheatstone.

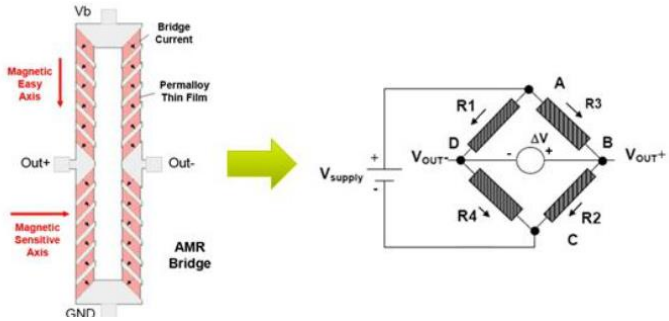


Figura 4-19. Magnetómetro – Puente de Wheatstone.

ST Microelectronics. DigiKey Corporation. [Online].
http://dkc1.digikey.com/il/en/tod/STMicroelectronics/MEMS

El fabricante, para crear el sensor con los elementos AMR, usualmente los orienta en forma de diamante con los extremos conectados entre sí por metalización para formar el puente de Wheatstone, y la diagonal de los elementos (R1, R2, R3 y R4) son iguales. Pero R1 y R2 tienen polaridad opuesta a R3 y R4. Cuando un campo magnético externo es aplicado, si R1-R2 incrementa la resistencia ΔR, y R3-R4 decrece la resistencia ΔR.
Explicando un poco más el principio de medición con las AMR, con el puente de Weatstone se realiza una conexión superior e inferior de los cuatro elementos idénticos y se les da un estímulo de corriente directa (DC) en la forma de una tensión de alimentación (Vsupply), con las conexiones de los lados restantes a ser medidos (Vout-, Vout+). Sin un campo magnético suministrado (0gauss), los contactos laterales deben estar a la misma tensión, a excepción de una pequeña tensión de offset debido a las tolerancias de fabricación de los elementos de AMR. Estos contactos secundarios producen una tensión diferencial (ΔV) como una función de la tensión de alimentación, la relación de AMR, y el ángulo theta (θ); que es el ángulo entre el flujo de corriente y el elemento de magnetización (M). En otras palabras, el cambio en el ángulo θ resulta expresado en el elemento AMR resistivo con el cambio de ΔR y lo que puede detectarse con el cambio de la salida de voltaje ΔV en el monitoreo del puente de Weatstone.

4.2.4 Sensor de Velocidad: Encoder en las ruedas

Un encoder es un sensor electro-opto-mecánico que, unido a un eje, proporciona información de la posición angular. Su fin, es actuar como un dispositivo de realimentación en sistemas de control integrado.

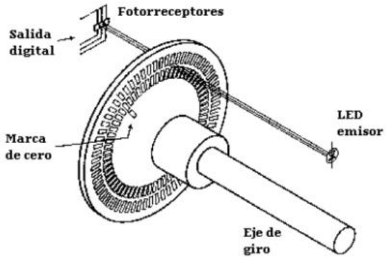


Figura 4-20. Encoder – Principio de Funcionamiento.

Los codificadores ópticos o encoders incrementales constan, en su forma más simple, de un disco transparente con una serie de marcas opacas colocadas radialmente y equidistantes entre sí, de un sistema de iluminación en el que la luz es colimada de forma adecuada, y de un elemento fotorreceptor.
Este tipo de encoder, se caracteriza porque determina su posición, contando el número de impulsos que se generan cuando un rayo de luz, es atravesado por marcas opacas en la superficie de un disco unido al eje.

4.2.4.1 Ejemplo de uso

Para poder calcular la velocidad del vehículo (suponiendo que se desplace en línea recta, con igual velocidad en todas las ruedas), se necesita conocer el perímetro de la rueda, la cantidad de marcas del disco del encoder y la cantidad de Ticks (marcas) por segundo que interrumpen el haz infrarrojo.

Figura 4-21. Encoder – Ejémplo aplicado al proyecto.

En nuestro caso:

Diámetro $D = 69mm = 0{,}069m$
Perímetro $P_e = \pi * D = 0{,}217m$
Cantidad de marcas (Ticks) = 20
Desplazamiento por Tick $D_{Tick} = \frac{P_e}{Ticks} = 0{,}0108m$
Cantidad de Ticks en tiempo transcurrido n_{Ticks}
Distancia recorrida (en dicho tiempo) = $n_{Ticks} * D_{Tick}$
Tiempo de muestreo $T_m = 100mSeg = 0{,}1Seg$
Velocidad $v_{100mSeg} = \frac{distancia\ recorrida}{tiempo} = \frac{n_{Ticks} * D_{Tick}}{0{,}1Seg} = n_{Ticks} * 0{,}108[\frac{m}{Seg}]$

4.2.5 Conjunto Motores, Caja Reductora, Ruedas y Rueda de Encoder

El vehículo consta de cuatro ruedas y para cada una de ellas se utilizará un kit comercial que incluye motor, caja reductora, rueda y marcador para encoder. El kit seleccionado, tiene la ventaja de su bajo coste y el hecho de que es bastante fácil de conseguir localmente. La desventaja es que estaría trabajando muy cerca de los límites de sus prestaciones respecto a torque, rigidez mecánica y regulación de velocidad. Las otras alternativas analizadas y mucho más recomendables, representan costos significativamente superiores, sumado a las dificultades relacionadas con la logística y los tiempos para su importación.
Debido a lo anterior, se decidió armar el proyecto con estos motores, tratando de:

- Compensar el bajo torque con el uso de 4 motores en lugar de 2.
- Compensar los defectos en la linealidad de la regulación mediante la implementación de un PID por software.
- Compensar la baja resolución del encoder modificando los tiempos de muestreo del software.

Figura 4-22. Conjunto motor caja reductora rueda y encoder (Kit TA0132).

Como referencia, cuando se suministran los 5 V CC, estos conjuntos se mueven aproximadamente a una velocidad de 740 mm/s. En caso de quitar el neumático y reemplazarlo por una goma de menor espesor, reduciendo el diámetro exterior de 69mm a 52mm, se obtendría como resultado una disminución de la velocidad del 74 % y un aumento del par del 35 %.

4.2.5.1 Motor y Caja Reductora

Se trata de un motor de 3 a 6 V CC con una caja reductora de relación de reducción de 48,75:1. El eje de salida está en ángulo recto con el eje del motor. Estos motores, también llamados "motores TT" (en este caso: motor TT de 3V a 6V – 200RPM).

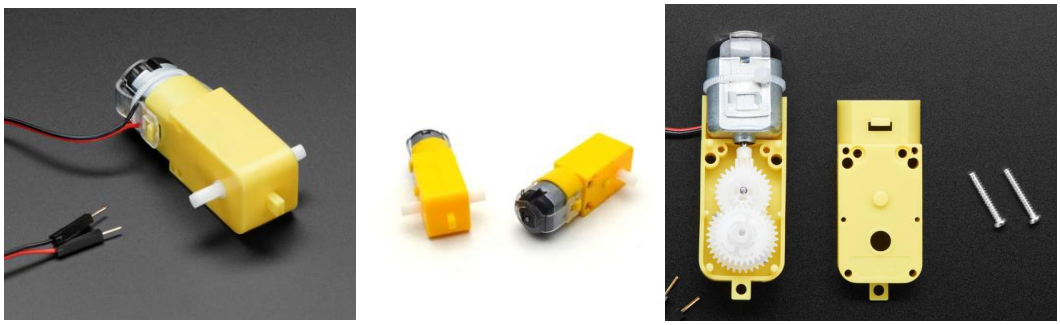


Figura 4-23. Detalle del Motor y la Caja Reductora.

El tren de engranajes consta de un engranaje de piñón de 8 dientes en el motor, luego un engranaje de 26:9, 28:14, 36:16 y un engranaje de 30 dientes en el eje de salida. Esto da como resultado una relación de 48,75:1, una reducción de 1:0,025 o, a la inversa, 1 vuelta del eje de salida requiere 48,75 vueltas del motor. Esto también aumenta el par del motor por el mismo factor de 48,75.

Unos ensayos a distintas tensiones nos pueden dar una idea de consumo y de las RPM que serían esperables al ensamblarlos:

- A 3 VCC: medimos 150 mA, 120 RPM sin carga. Y 1,1 amperios cuando está bloqueado
- A 4,5 VCC: medimos 155 mA a 185 RPM sin carga y 1,2 amperios cuando está bloqueado
- A 6 VCC, medimos 160 mA a 250 RPM sin carga y 1,5 amperios cuando está bloqueado

Reduction Ratios: 1/48 1/120 1/220 1/288

MODEL		VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
		OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			V	r/min	A	r/min	A	N m	kgf cm	W	N m	kgf cm	A
TGP01D-A130	10300-288	3.0 - 12.0	6	21	0.10	19	0.17	0.05	0.47	0.18	0.18	1.8	0.6
	14175-120	3.0 - 6.0	4.5	90	0.20	72	0.50	0.03	0.3	0.38	0.14	1.4	1.25
	12215-48	3.0 - 6.0	3.0	110	0.12	85	0.20	0.01	0.11	0.13	0.027	0.27	0.45
	18100-220	3.0 - 6.0	3.0	50	0.25	34	0.45	0.07	0.71	0.52	0.24	2.4	1.1

Figura 4-24. Especificaciones Técnicas del Conjunto Motor - Caja.

Estos son motores muy básicos y no tienen codificadores incorporados, control de velocidad o retroalimentación posicional. Habrá variación de motor a motor, por lo que se requiere un sistema de retroalimentación separado si necesita un movimiento de precisión. La modulación de ancho de pulso (PWM) se puede usar para sincronizar los motores si hay una gran discrepancia cuando se aplica la carga.

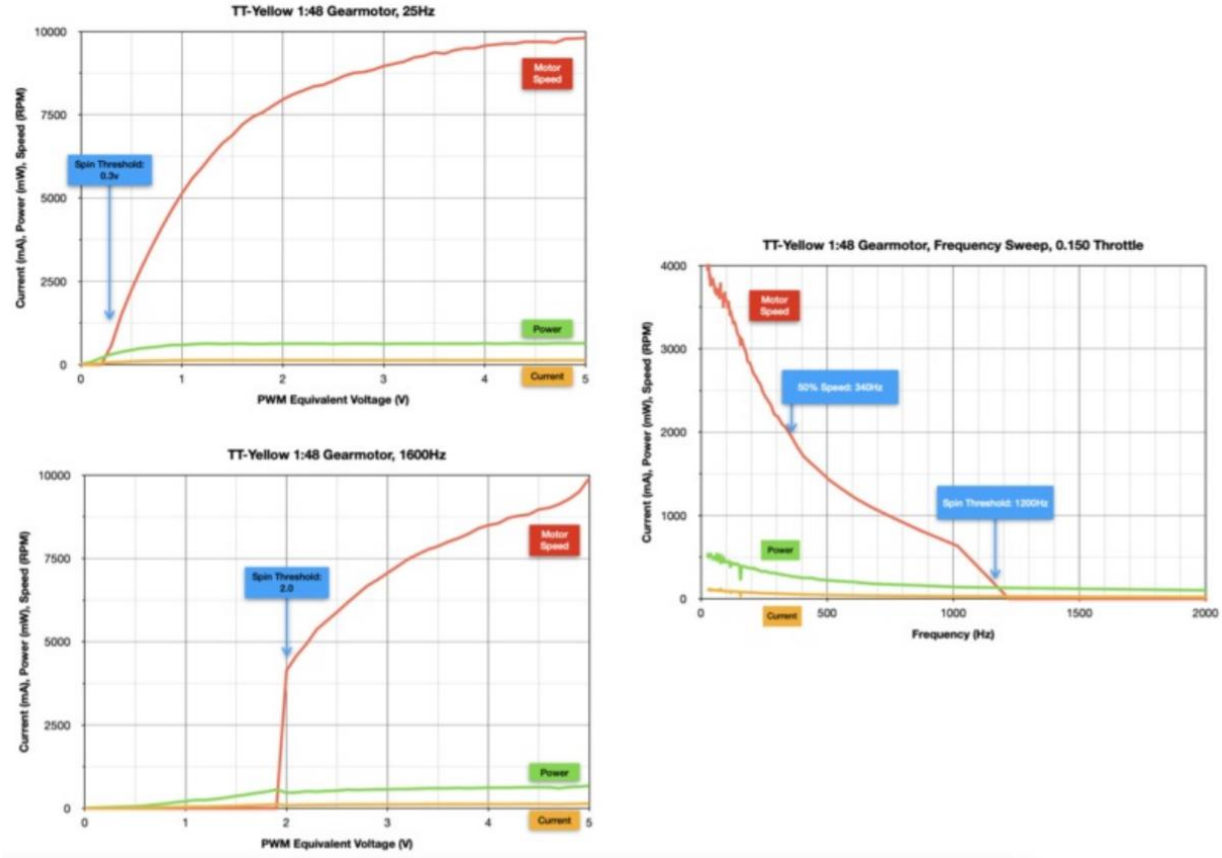


Figura 4-25. Respuestas de los motores controlados por PWM.

A continuación, se complementa la información técnica de los conjuntos, con las dimensiones físicas de los mismos.

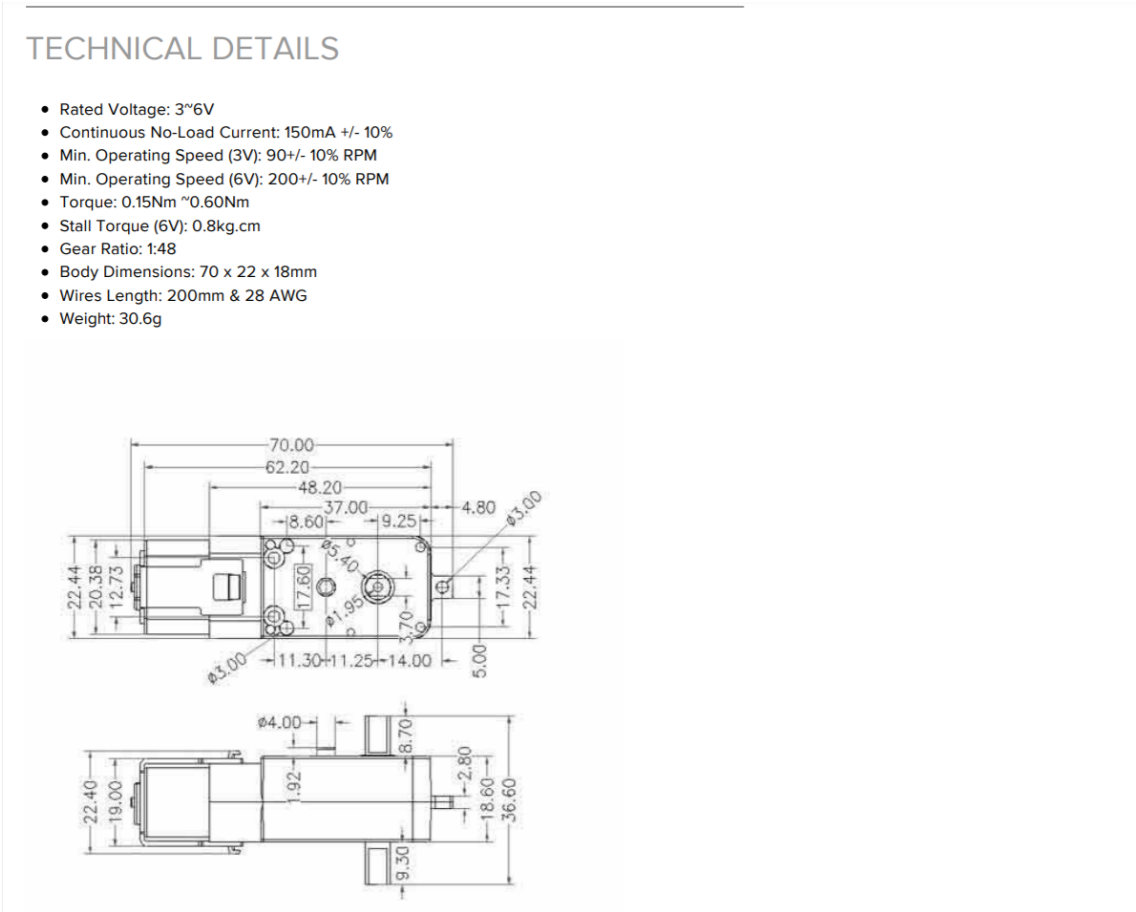


Figura 4-26. Detalles Técnicos y dimensiones del conjunto Motor-Caja reductora.

Nota: Estos motores son de bajo costo y prestaciones en el límite mínimo de lo que el proyecto requiere. Se espera un prototipo terminado, con un peso aproximado a los 2 kg y que necesita moverse a velocidades muy bajas para poder hacer un mapeo correcto. Para ello se deberá reducir bastante el duty cicle del PWM, lo que redundará en una pérdida notable de torque en los motores. Para tratar de compensar esto, se utilizó una configuración de 4 motores. Los ensayos determinarán si esta configuracíon y modelo de motores logran resultados aceptables.

4.2.5.2 Ruedas

Ruedas pásticas con cubiertas de goma, aptas para acoplar directamente al eje de la caja reductora del motor.

Figura 4-27. Ruedas Dimensiones.

4.2.5.3 Rueda Ranurada - Encoder

Es una rueda ranurada que se coloca en el otro extremo del eje de la caja reductora del motor. Agregando un sensor infrarojo de ranura, se puede determinar velocidad velocidad y desplazamiento de cada rueda.

Figura 4-28. Detalle del montaje de la Rueda del Encoder.

La rueda ranurada se complementa con un sensor infrarrojo tipo ranura. Consta de un emisor y un receptor. Su principio de funcionamiento fue descripto anteriormente

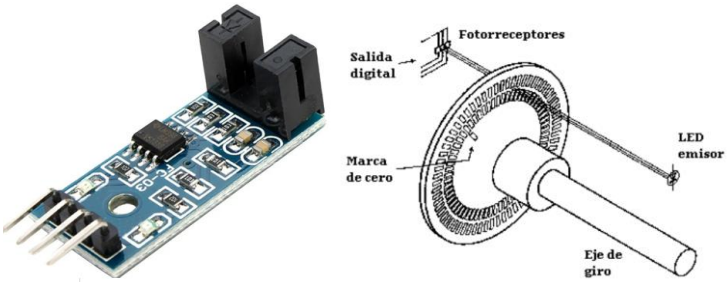


Figura 4-29. Detalle del sensor de ranura y principio de funcionamiento.

Nota: La rueda ranurada posee sólo 20 ranuras. Por lo tanto, en una vuelta completa, 360 grados, tendríamos 20 "ticks" o cuentas de las interrupciones del haz de luz. Esto nos dá una resolución de 18 grados por tick, la cual es bastante baja y puede significar un trabajo extra al ajustar los tiempos de muestreo del soft que debe ajustar la velocidad de cada rueda.

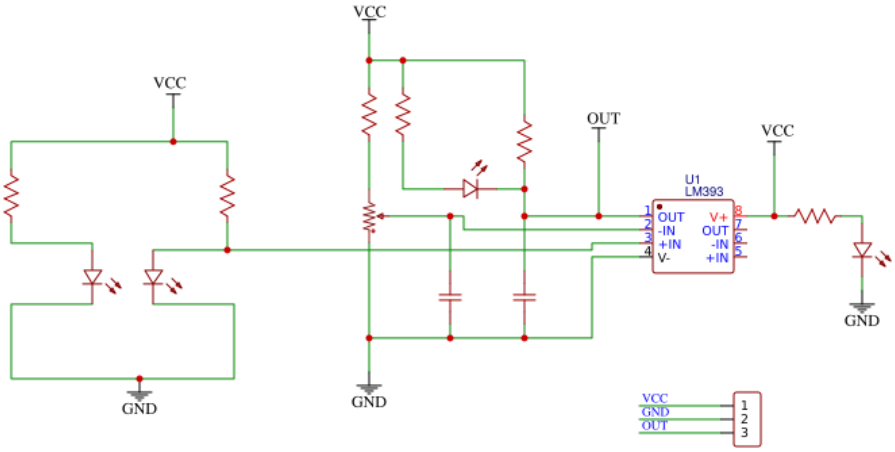
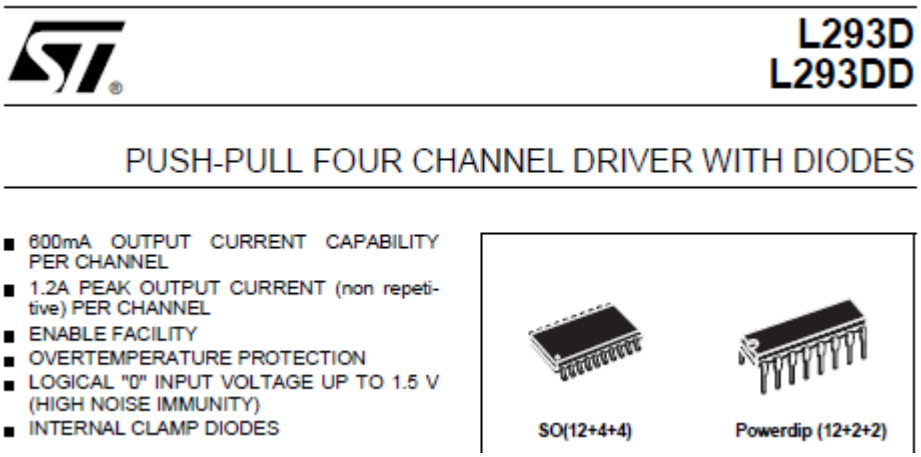


Figura 4-30. Diagrama esquemático de cada sensor de ranura del encoder.

4.2.6 Driver para motores

Los motores no pueden ser alimentados directamente por el microcontrolador. En el punto anterior se expresaron las características eléctricas de los motores seleccionados. De ellas se deduce que se espera manejar 4 motores con corrientes que varían entre los 200mA y 1.2Amperes cuando están bloqueados y alimentados con CC. (Esto

se vé notablemente reducido al usar PWM).
El driver más utilizado para este tipo de control es el L293D. Dicho circuito integrado es un cuadruple driver Push Pull, con él se pueden implementar 2 puentes "H" para controlar 2 motores de 600mA cada uno y una corriente de pico (no repetitivo) de 1,2A. Lo cual se ajustaría perfectamente a nuestro proyecto. Necesitando utilizar 2 CI de este tipo.



L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

SO(12+4+4) Powerdip (12+2+2)

Figura 4-31. Extracto del DataSheet del L293D (web de ST).

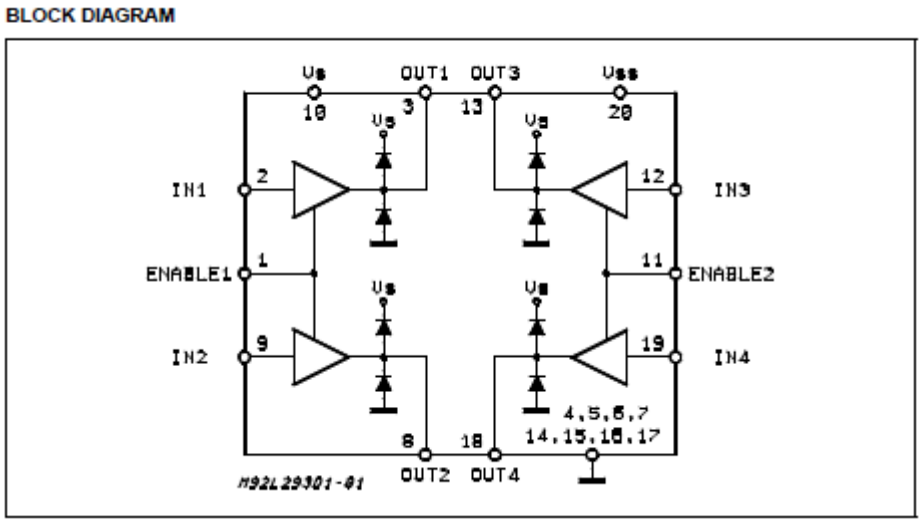


Figura 4-32. Diagrama en bloques del L293D (web de ST).

Un tema interesante se presenta en este punto. Se necesitan comprar 2 CI L293D, diseñar un circuito impreso, fabricarlo, montarlo, comprar algún conector o componente extra, etc. Analizando rápidamente los costos, observamos lo siguiente:

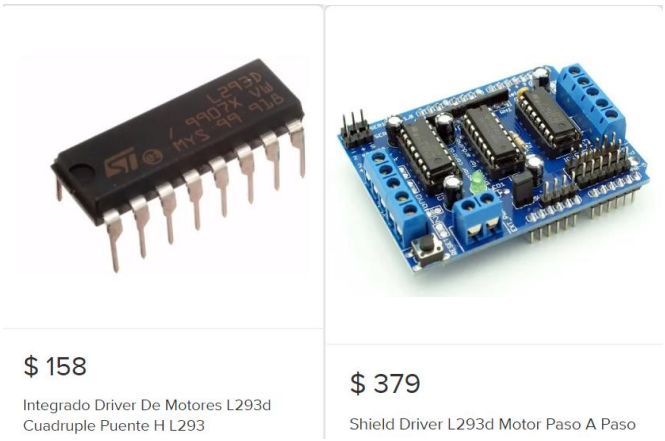


Figura 4-33. Comparación de precios L293D vs Shield para 4 motores.

De lo que se deduce, que, para este proyecto, con los alcances y requerimientos definidos previamente, es poco práctico fabricar un circuito impreso dedicado para esta funcionalidad.

Se decide utilizar un shield comercial con las siguientes características:

- Capaz de manejar 4 motores de CC y 2 servos
- Capaz de manejar 2 motores paso a paso y 2 servos
- El voltaje de control lógico Vss: 4.5 ~ 5.5V
- Voltaje de suministro del motor: 4.5 ~ 12V
- Pines de control reducidos necesarios a través de registros de desplazamiento en serie de 8 etapas
- Conducir parte de la corriente de funcionamiento Io: 1.2ª

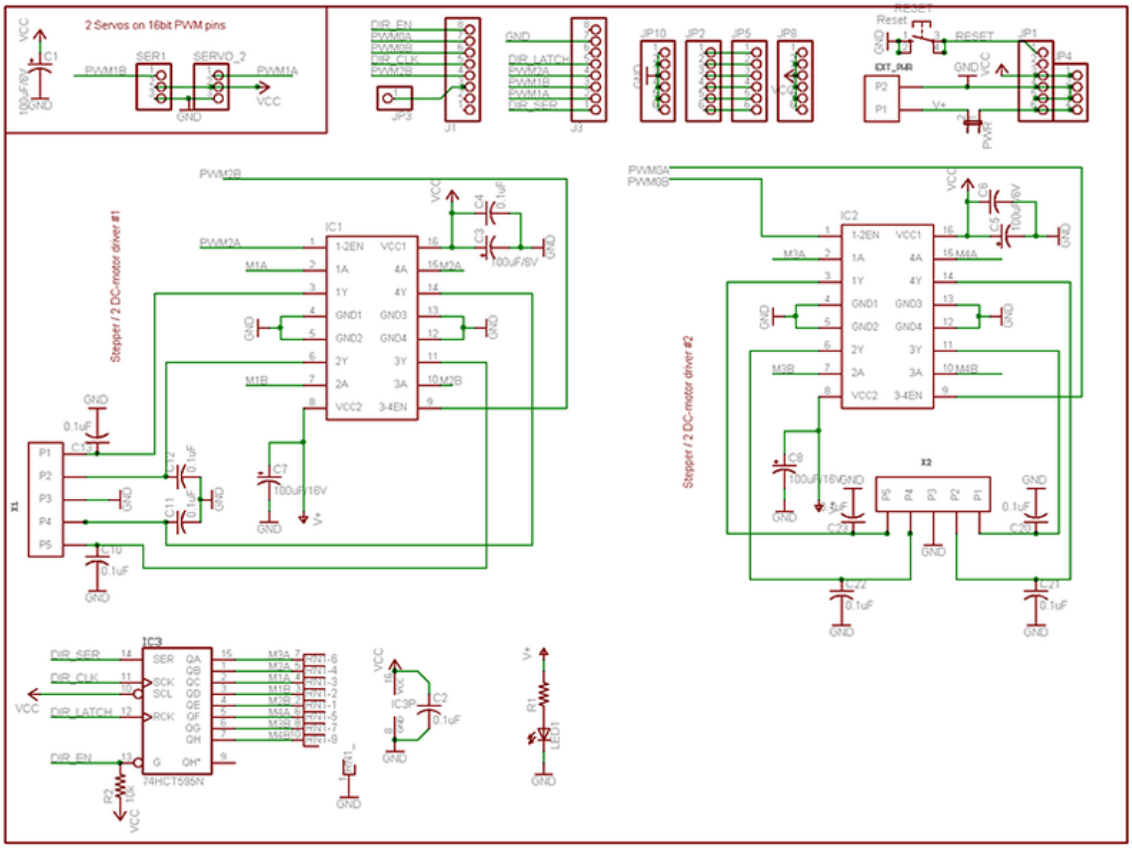


Figura 4-34. Diagrama esquemático del driver para control de los 4 motores.

4.2.7 CPU para control de motores

La configuración definida para el vehículo está dividida en 2 bloques o niveles. El nivel base necesita una CPU capaz de:

- Controlar 4 motores (PWM) (Mediante Driver).
- Procesar los pulsos de 2 o 4 encoders de velocidad.
- Comunicación I2C para conectarse a Módulo IMU.
- Conexión USB para simplificar el conexionado con la CPU principal.
- Memoria SRAM igual o superior a 3K, debido a los requerimientos necesarios para poder utilizar ROS_LIB. Para subscribir al tópico “TWIST”, los mensajes de ODOMETRIA requieren bastante memoria SRAM.

Se decide utilizar un módulo basado en ATMega2560 que satisface los requerimientos mencionados a un precio razonable, sumado a la compatibilidad con el módulo driver de motores seleccionado en el punto anterior.

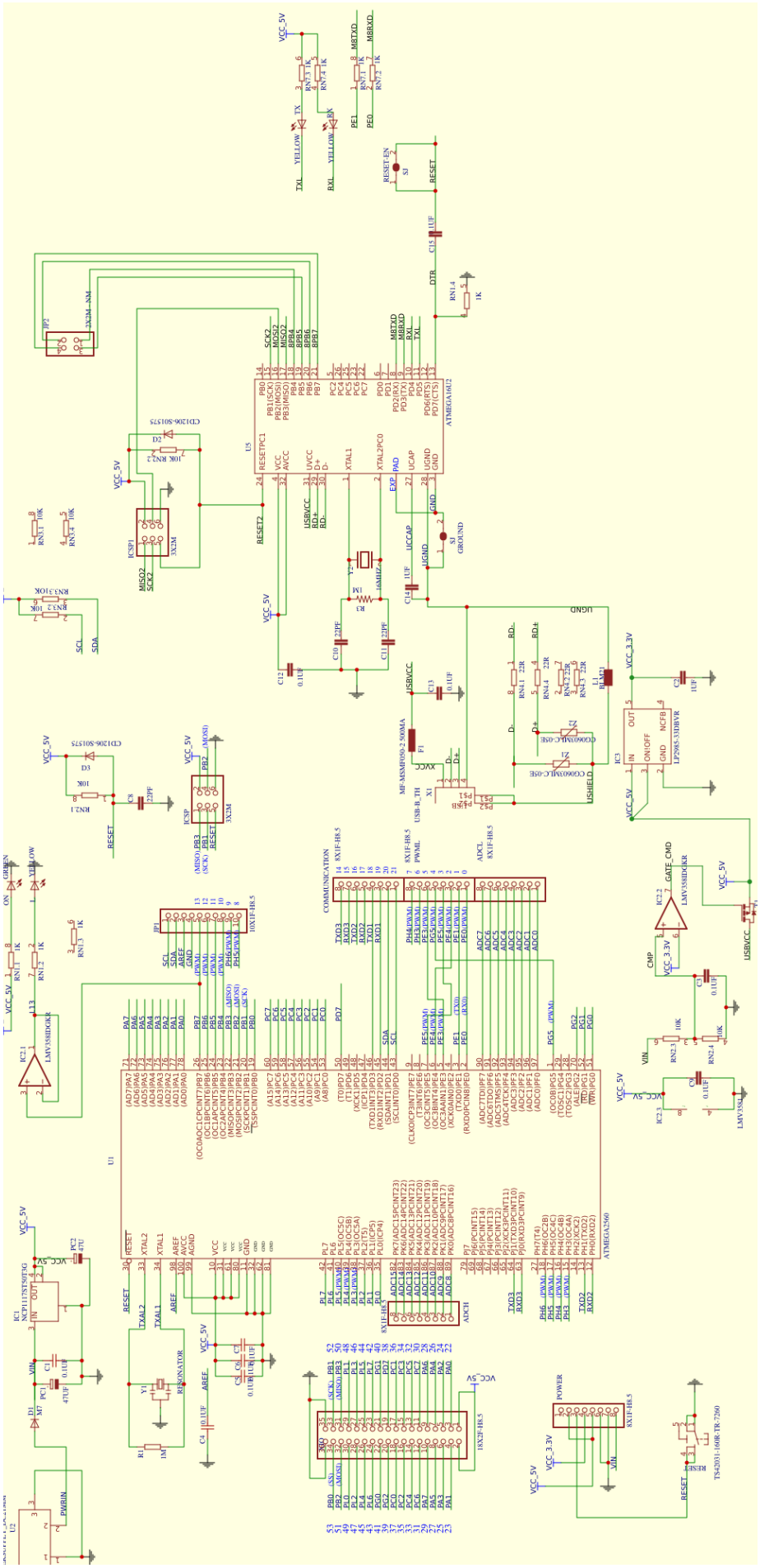


Figura 4-35. Diagrama esquemático de la CPU secundaria ATMega2560.

Especificaciones:

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 64K/128K/256KBytes of In-System Self-Programmable Flash
  - 4Kbytes EEPROM
  - 8Kbytes Internal SRAM
  - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/ 100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
    - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
  - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change

4.2.8 Joystick

Se consideró la posibilidad de poder controlar el vehículo mediante un Joystick, para poder facilitar las pruebas de control de motores y mapeos.

Se utiliza un Joystick Dual Shock de PS2 con un receptor para acoplar al módulo de CPU secundaria

Figura 4-36. Conjunto Joystick PS2 mas receptor.

Figura 4-37. PinOut del receptor.

Descripción de funciones:

- DATOS: Esta es la línea de datos del controlador a la PS2. Esta es una salida de colector abierto y requiere una resistencia pull-up (1 a 10k).
- COMANDO: Esta es la línea de datos de PS2 al controlador.
- POTENCIA DEL MOTOR DE VIBRACIÓN
- TIERRA: Tierra
- VCC: VCC puede variar desde 5V hasta 3V.
- ATT: ATT se utiliza para llamar la atención del controlador. Esta línea debe bajarse antes de enviar/recibir cada grupo de bytes, y luego volver a establecerse en alto. Este pin se considera como una línea de "selección de chip" o "selección de esclavo" que se usa para direccionar diferentes controladores en el mismo bus.
- CLK: 500kH/z, normalmente alto. La comunicación parece ser bus SPI.
- No conectado
- ACK: Señal de reconocimiento del controlador a PS2. Esta línea normalmente alta cae alrededor de 12 us después de cada byte durante medio ciclo de reloj, pero no después del último bit de un conjunto. Esta es una salida de colector abierto y requiere una resistencia pull-up (1 a 10k).

4.2.9 Baterías

Las baterías representan un desafío que tal vez no es comprendido en su totalidad al comenzar un proyecto de este tipo. Consideraciones:

- Tensiones y corrientes requeridas por las diversas partes a alimentar.
  - La CPU RaspBerry requiere una alimentación de 5V con un consumo entre 3.8W (idle) hasta un máximo de 6.0W con 4 cores trabajando.
  - Los motores funcionan entre 3v a 6v con una corriente entre 200mA (en vacío) y 1.2A (bloqueados), alimentados con CC. Pero se espera mucho menor consumo debido al PWM. Los motores probaron funcionar mucho mejor con PWM en tensiones cercanas a los 8V. Con 5V y PWM a bajas velocidad, el torque obtenido es muy bajo.
  - La CPU ATMega funciona entre 7V y 12V si se alimenta a través de su propio regulador. O también se puede alimentar con 5V a través del USB. El consumo esperado es del orden de los 50mA
  - El LiDAR se alimenta con 5V con una corriente de arranque del orden de los 500mA que luego se estabiliza en el orden de los 350mA
- Peso y calidad de las Baterías: durante las pruebas se ensayó con un sistema mixto: un PowerBank de 10.000mA para RaspBerry, Laser y CPU secundaria; y un array serie/paralelo, conectado a un voltímetro de panel, para los motores, de 2 baterías tipo 18650, de 3.7V y 2200mA cada una, obteniéndose 7.4V y 2200mA. Con esta configuración obtuve los siguientes resultados:
  - Dentro de los 20 minutos de operación, la RaspBerry se reiniciaba debido a la caída de tensión del PowerBank, a causa del consumo propio, del LiDAR y de la CPU secundaria.
  - La tensión en el Pack 18650 sufría grandes variaciones que imposibilitaban calibrar los PWM para lograr una velocidad uniforme para el mapeo durante más de diez minutos. También, el consumo de los motores al momento de tener que doblar, es siginificativamente superior que

al conducir en línea recta. Esta variación de consumo se traduce en variaciones de tensión inferiores al voltio, pero dicho valor representa un impacto superior al 10% de la tensión que alimenta los PWM de los motores. ROS, mediante un tópico llamado CMD_VEL, solicita al robot un desplazamiento determinado en m/s (metros por segundo) en sus tres ejes. Además, las ordenes para doblar, se experan en rad/seg (radianes por segundo). Tratar de mantener el vehículo a una determinada velocidad con valores de tensión que pueden variar entre 8V con baterías cargadas y cercanos a los 6V con baterías descargadas. O fluctuaciones superiores al 10% en la rensión de acuerdo al torque que requiera la maniobra, requieren un esfuerzo considerable de programación, los resultados obtenidos con esta configuración de baterías, no resultó satisfactorio. Luego se agregó otro conjunto de 7.4V 2200mA, mejorando la respuesta en tensión, pero el aumento de peso en el robot, pejudicaba algún motor de acuerdo a la ubicación de la batería anexada. Parte del problema, es también, la calidad de las baterías, las cuales, a pesar de tener las mismas características, presentaban rendimientos muy distintos, lo cual es imposible de conocer antes de comprarlas y testearlas.

Finalmente, luego de horas de ensayos, se logró una configuración aceptable dentro de ciertos límites, unificando todos los dispositivos a 5V y utilizando 3 PowerBank de celulares:

- 1 PowerBank de 12.000 mA/h
- 1 PowerBank de 10.000 mA/h
- 1 PowerBank antiguo sin especificaciones

Un dato interesante a tener en cuenta es que los PowerBank, a pesar de contar con baterías que superan la demanda de corriente del conjunto conectado, tienen un circuito regulador que limita la corriente de salida. El dispositivo utiliza 3 PowerBank en paralelo para poder subsanar el inconveniente de esa limitación de corriente, contrariamente a lo que se podría pensar respecto a la capacidad de las baterías internas.

Para futuros desarrollos sería interesante explorar el uso de Baterías de polímero de litio (LIPO). Estas baterías son una variación de las baterías de iones de litio (Li-ion). Sus características son muy similares, pero permiten una mayor densidad de energía, así como una tasa de descarga bastante superior. Estas baterías tienen un tamaño más reducido respecto a las de otros componentes. Cada celda tiene un voltaje nominal de 3,7 V, voltaje máximo 4,2 V y mínimo 3,0 V. Requieren de un cargador especial.

4.2.10 Chasis

El vehículo está pensado para ser montado en 2 pisos o niveles. Se espera un peso del orden de los 2Kg del conjunto armado. Debido a las medidas y montajes propios de los kits de ruedas y de los encoders de las mismas, se decidió tomar como base una plataforma comercial dimensionada para esos kits. La plataforma original es de un acrílico muy delgado, por lo tanto, se levantó un modelo en AutoCad, con las mismas medidas y se hizo cortar con laser, nuevas piezas en acrílico de 4mm.

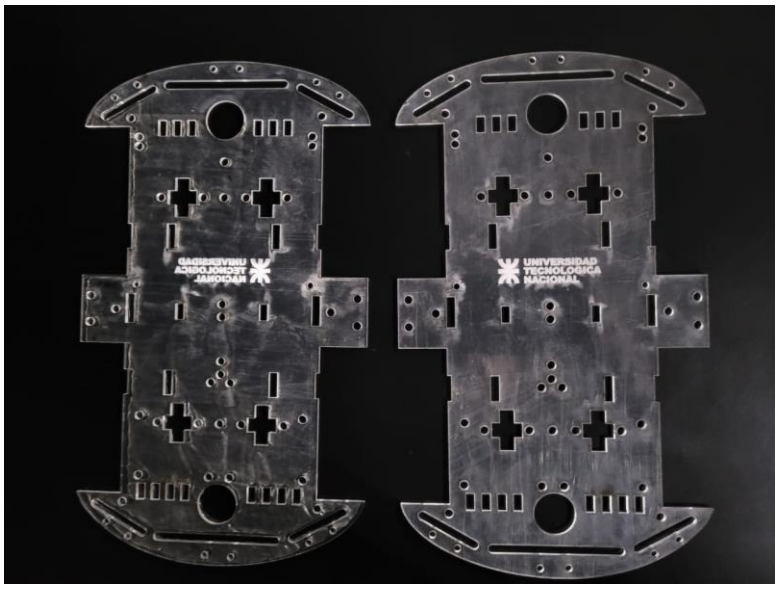


Figura 4-38. Bases acrílicas construidas con acrílico cortado por laser.

4.3 Diagrama en bloque general del diseño.

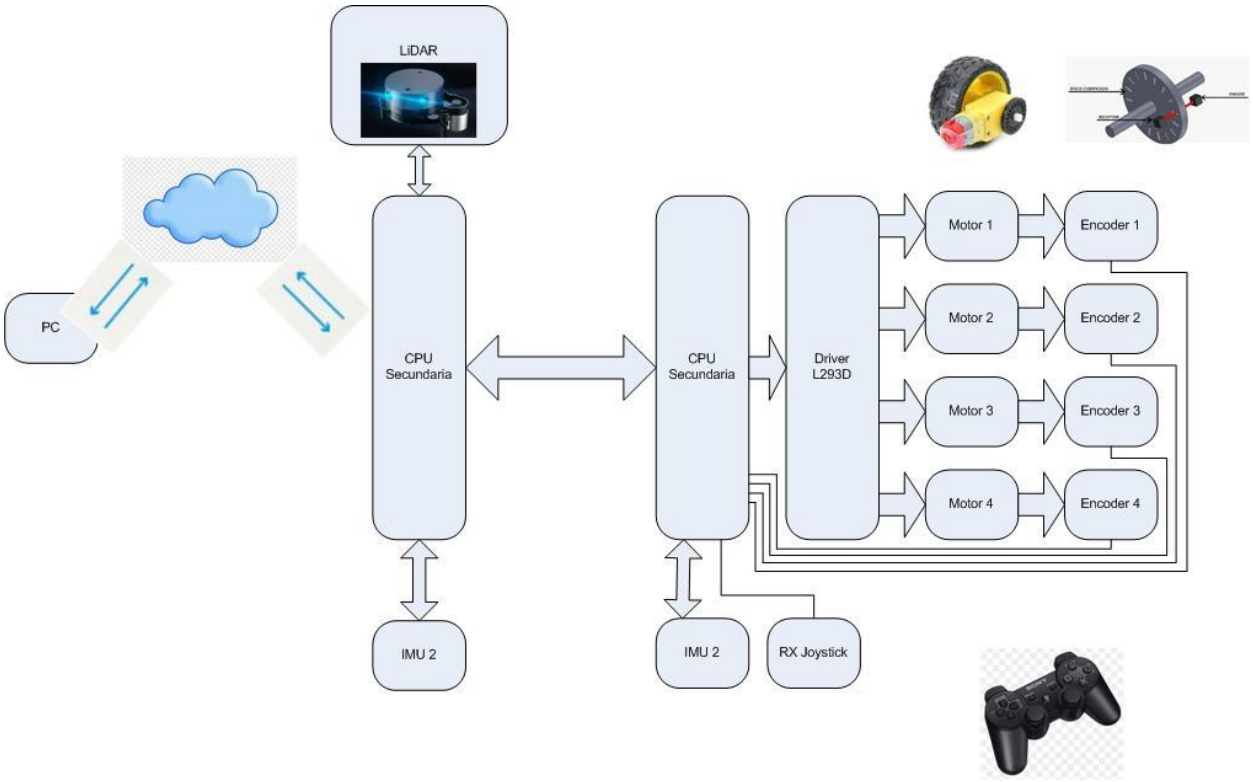
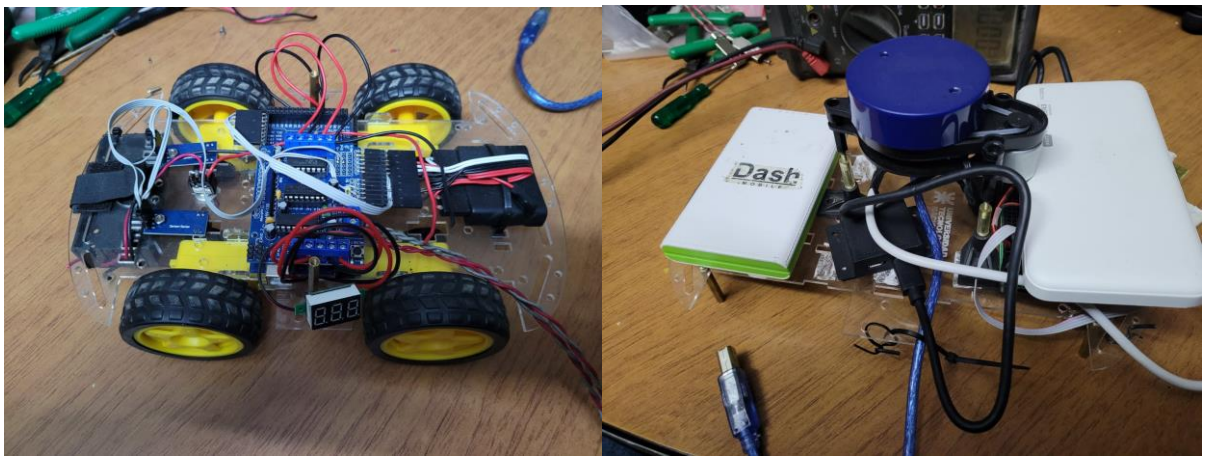


Figura 4-39. Diagrama en Bloques General del Proyecto.



Pisos Inferior y Superior por separado

Modelo Ensamblado

4.4 Funcionamiento y Operación

En esta sección se deben contemplar una serie de descripciones de acuerdo a la acción que se desea realizar

4.4.1 Comprobaciones iniciales

Debido a que el vehículo es un sistema que involucra una serie de elementos de software y hardware, se deben chequear que todos se encuentren en línea para poder comenzar a operarlo.

4.4.1.1 Comprobación Joystick y Control mediante Joystick

Al conectar alimentación al vehículo, se debe enlazar el Joystick BlueTooth, y comprobar su funcionamiento. Los pasos a seguir:

- Encender Joystick, las luces roja y verde van a parpadear.
- Alimentar el vehículo y observar que la luz verde queda fija y la roja apagada.
- Presionar MODE y observar que la luz roja queda fija.
- Con las 4 flechas (arriba, abajo, izquierda y derecha) comprobar movimiento de las ruedas

Figura 4-40. Enlazando Joystick BlueTooth.

Una vez que el Joystick se enlazó correctamente, ya es posible controlar al robot desde el mismo. El receptor de BlueTooth se encuentra conectado directamente a la CPU secundaria, que a su vez es la que controla los motores. Mediante las flechas (arriba, abajo, derecha o izquierda) es posible ordenarle una trayectoria al robot. Obsérvese en el diagrame de flujo, que los módulos adelante, atrás, derecha, izquierda y parar; también se encuentran en el apartado control por teleop_twist_keyboard, que se analizará más adelante. Esto se debe a que el control del sentido de giro y la velocidad de las ruedas a través de PWM es el mismo, independientemente del tipo de control que se realice.

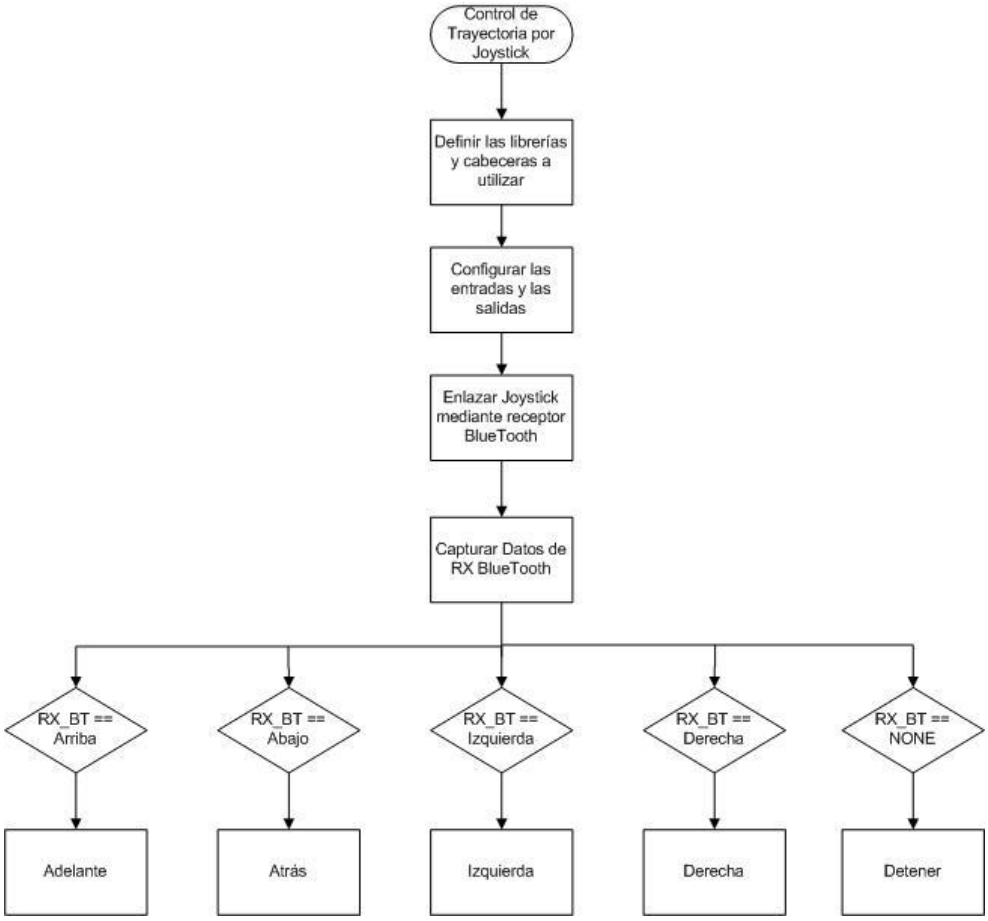


Figura 4-41. Algoritmo de funcionamiento de control de trayectoria mediante Joystick.

4.4.1.2 Comprobación Conexión remota al vehículo

En la RaspBerry se encuentran pre configuradas las redes WiFi donde se supone que el vehículo va a funcionar. Se debe corroborar que el vehículo esté conectado al WiFi y que pueda ser accedido remotamente.
Pasos a seguir:

- Comprobación de está conectado al WiFi. La manera más sencilla es hacer ping a la IP propia del vehítulo.

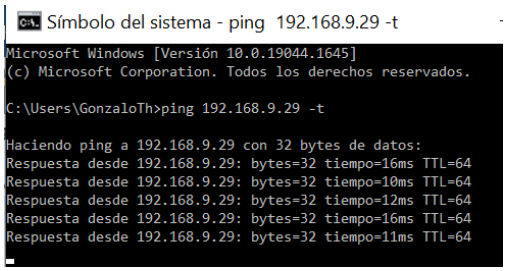


Figura 4-42. Haciendo Ping al robot.

- Comprobación del acceso remoto. En este caso se accede mediante el software VNC para acceso remoto. Si todo está OK, se solicitará la autenticación para poder conectarse (Usuario de VNC y contraseña).

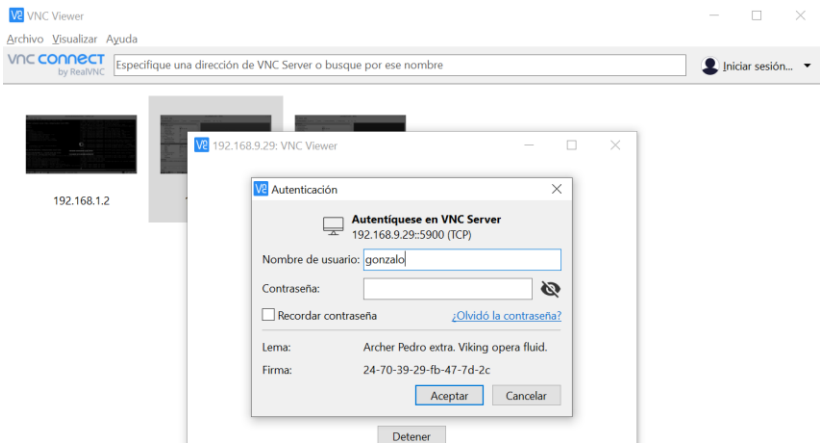


Figura 4-43. Accediendo a control remoto por VNC.

- Inmediatemente aparecerá la pantalla de la RaspBerry solicitando loguearse con usuario y contraseña

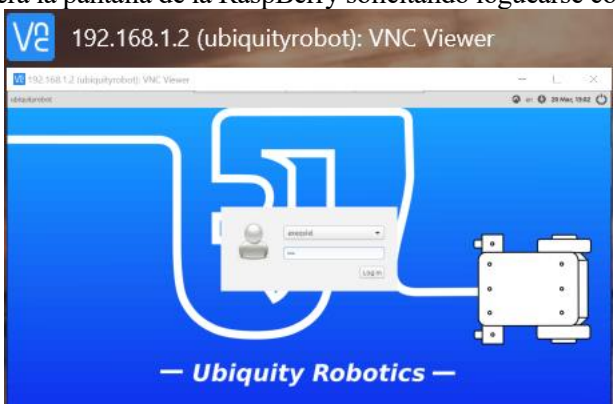


Figura 4-44. Logueando en Ubiquity Robotics.

- Una vez realizado exitosamente todo lo anterior, se puede proceder a trabajar con el equipo.

4.4.1.3 Preparando entorno en Ubuntu

Va a ser necesario trabajar con el usuario UBUNTU para acceder a los permisos necesarios. Además se utilzarán varias terminales para ejecutar los distintos módulos.

- Primeramente, abrir una terminal (Cntrol Alt T) y cambiar a usuario Ubuntu (y Password)

Figura 4-45. Usuario UBUNTU.

- Va a ser necesario trabajar en varias terminales a la vez, en este caso ejecuto TMUX y abro 4 terminales

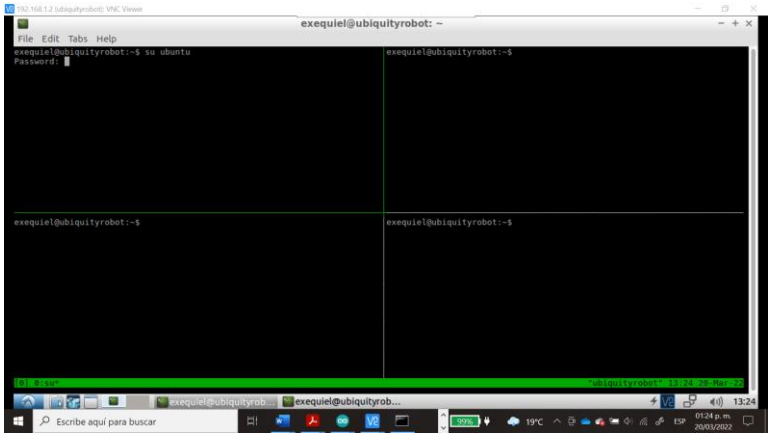


Figura 4-46. Abriendo entorno TMUX.

4.4.1.4 Ejecucíon ROSSERIAL

Rosserial es un protocolo para envolver mensajes serializados ROS estándar y multiplexar múltiples tópicos y servicios a través de un dispositivo de envío de caracteres, como un puerto serie o un conector de red.

- Author: Michael Ferguson
- License: BSD
- Source: git https://github.com/ros-drivers/rosserial.git (branch: noetic-devel)

Se debe comprender que se dispone de 2 CPU´s, la CPU principal ejecuta un entorno ROS que se debe comunicar mediante tópicos con la CPU secundaria. Estos tópicos se encuentran encapsulados en un protocolo ROS. Rosserial se encarga de agregar una capa más a ese protocolo para poder transmitirlo por un puerto determinado.

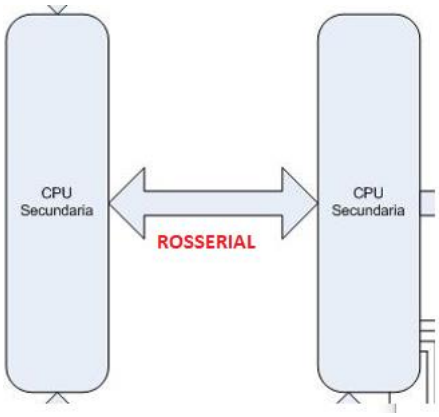


Figura 4-47. RosSerial entre ambas CPUs.

- En una de las terminales del apartado anterior, se va a ejecutar Rossrerial. Pero primeramente, es necesario conocer, en qué puerto, del RaspBerry, se encuentra conectado el USB de la CPU secundaria.

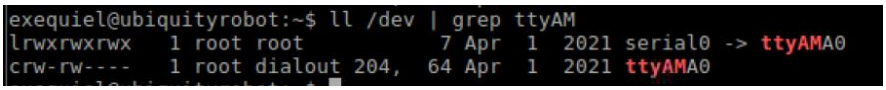


Figura 4-48. Determinando puerto de CPU secundaria.

- Y luego ejecutar Rosserial para ese puerto.

rosrun rosserial_python serial_node.py port= /dev/ttyAMA0

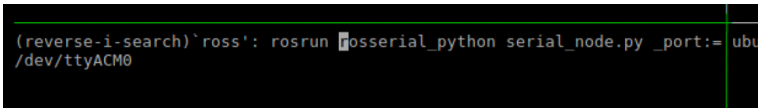


Figura 4-49. Ejecutando RosSerial.

Mediante este comando se establece una conexión ROS Serial entre ambas placas que permite a la Raspberry Pi suscribir o publicar mensajes, en los temas que indiquemos, en el programa de la CPU secundaria, en este caso, en el tópico **cmd_vel**. Esto permitirá suscribir los mensajes del nodo teleop_twist_keyboard que se explica en el siguiente apartado.

- Sería importante que los tópicos cmd_vel y speed

En este punto es interesante entender el significado de los valores del tópico cmd_vel.

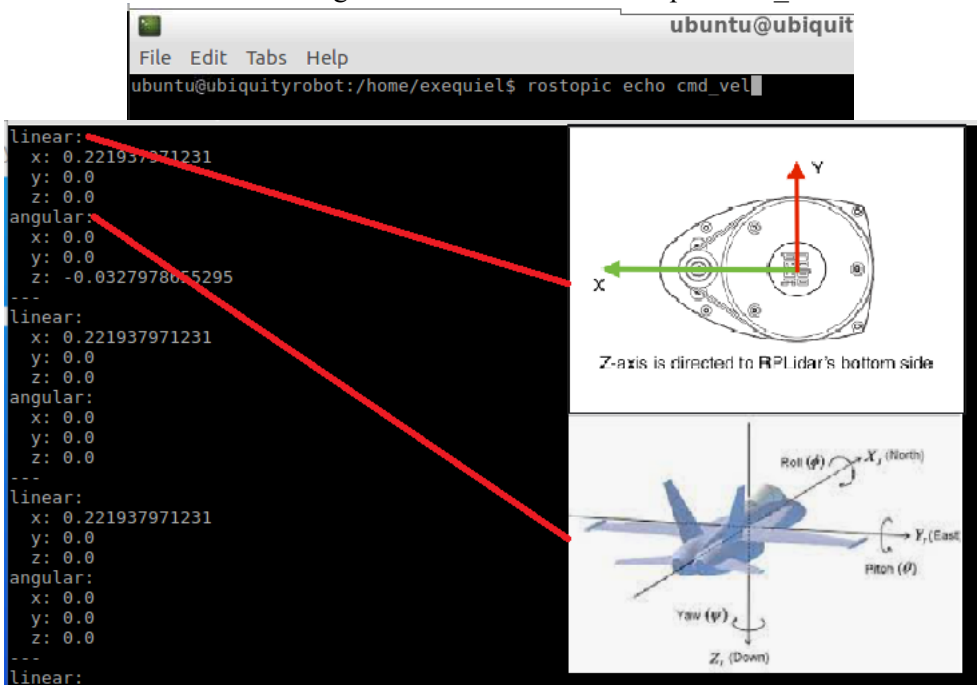


Figura 4-50. Interpretación del tópico cmd_vel.

Al observar los datos presentes en el tópico cmd_vel, encontramos:

- Linear: hace referencia a la velocidad lineal, expresada en metros por segundo, en cada eje.
- Angular: hace referencia a la velocidad de rotación, expresada en radianes por segundo, en cada eje. Por ejmplo: -1,047 rad/seg serían 60 grados/seg en sentido anti horario

Se debe tener en cuenta solo velocidad lineal en el eje X, pues este tipo de vehiculo no puede desplazarse hacia los costados ni hacia arriba o abajo. Un valor positivo en velocidad lineal en X, significa avanzar y un valor negativo, significa retroceder.
Respecto a la velocidad angular, sólo consideramos velocidad angular en Z, pues el vehículo no puede volcar hacia los ldados, ni subir o bajar la nariz, como se muestra en el avión de la figura. Los valores positivos indican rotación o giro en sentido horario y viceversa.

Una vez realizada la conexión entre ambas placas será necesario publicar el tema al que está
suscrito el serial node. Esto se realizará mediante el paquete de ROS teleop_twist_keyboard.
o Teleop_twist_keyboard
Fig. 27. Interface paquete teleop_twist_keyboard

4.4.1.5 Control mediante teleop_twist_keyboard

En este apartado se explica cómo controlar el robot móvil haciendo uso del teclado del PC desde el que se está controlando la Raspberry Pi.
Teleop es un soft realizado en Python, que captura ciertas teclas del teclado de la PC y las convierte en parámetros que publica en el tópico cmd_vel, con el fin de mover un robot en una determinada trayectoria.

- Generic keyboard teleop for twist robots.
- Author: Graylin Trevor Jay
- License: BSD

A continuación, se observa el algoritmo de funcionamiento del control de la trayectoria del robot mediante el uso del paquete teleop_twist_keyboard y conexión entre las CPU Principal y Secundaria:

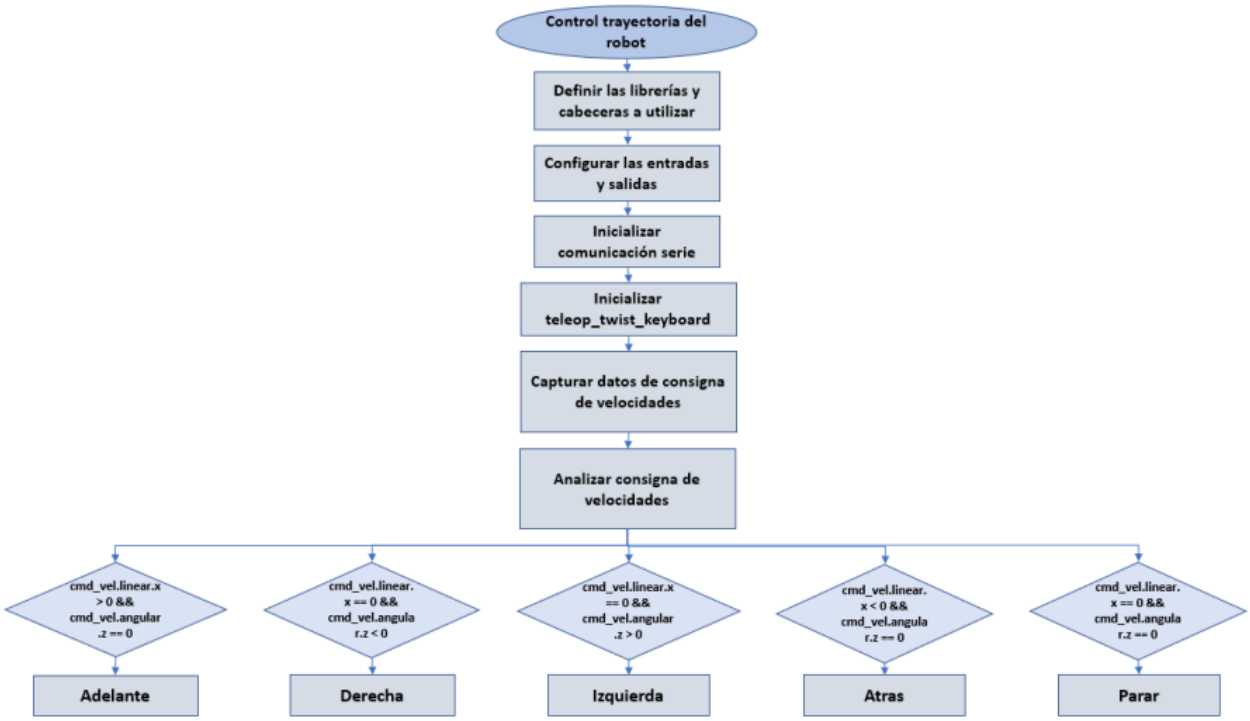


Figura 4-51. Algoritmo de funcionamiento de control de trayectoria.

El robot monitorea constantemente el tópico cmd_vel y ejecuta acciones de acuerdo a las velocidades lineales y angulares que les son comunicadas. Obsérvese que sólo atiende cmd_vel.linear.X y cmd_vel.angular.Z, como se explicó en el apartado anterior.
La CPU secundaria ajusta los PWM de los motores para tratar de ajustarse a la trayectoria ordenada. Los encoders en las ruedas proporcionan información que se realimenta para ajustar los PWM a las velocidades ordenadas.
Las teclas a utilizar y su función aparecen en pantalla al ejecutar Teleop.

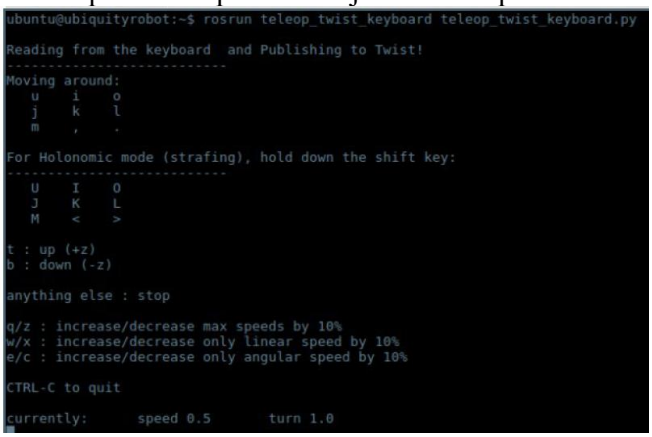


Figura 4-52. Control mediante teleop_twist_keyboard.

4.4.1.6 Control Mediante mouse_teleop

Una herramienta teleoperativa de dispositivo señalador (p. ej., mouse, panel táctil) para robots móviles, compatible con plataformas de accionamiento holonómico y diferencial.
Al igual que la anterior, captura la entrada de un dispositivo (antes era el teclado, ahora un dispositivo señalador), interpreta esa información y la convierte en parámetros de velocidades que se envían al robot mediante el tópico cmd_vel.
Del lado del robot, es exactamente igual al caso anterior. El robot no sabe, si esas ordenes se generaron con un teclado o un mouse. El robot sólo atiende los pedidos a través de cmd_vel.

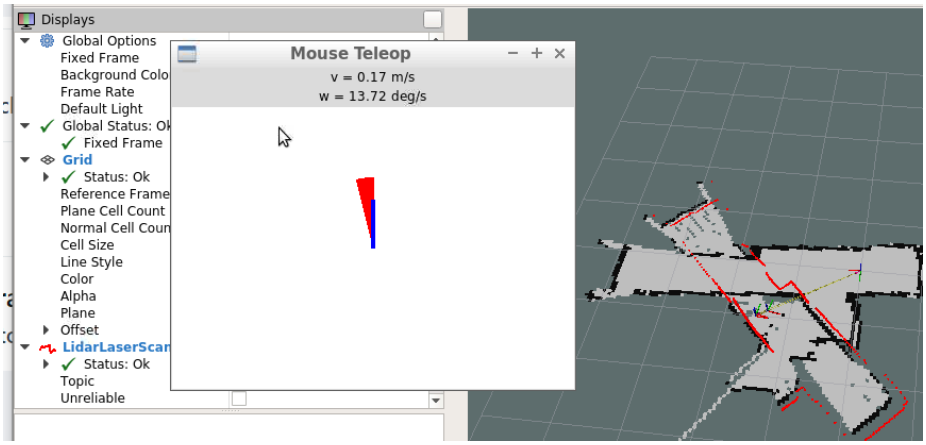


Figura 4-53. Control mediante mouse_teleop.

4.4.2 Mapeo de la planta de un edificio

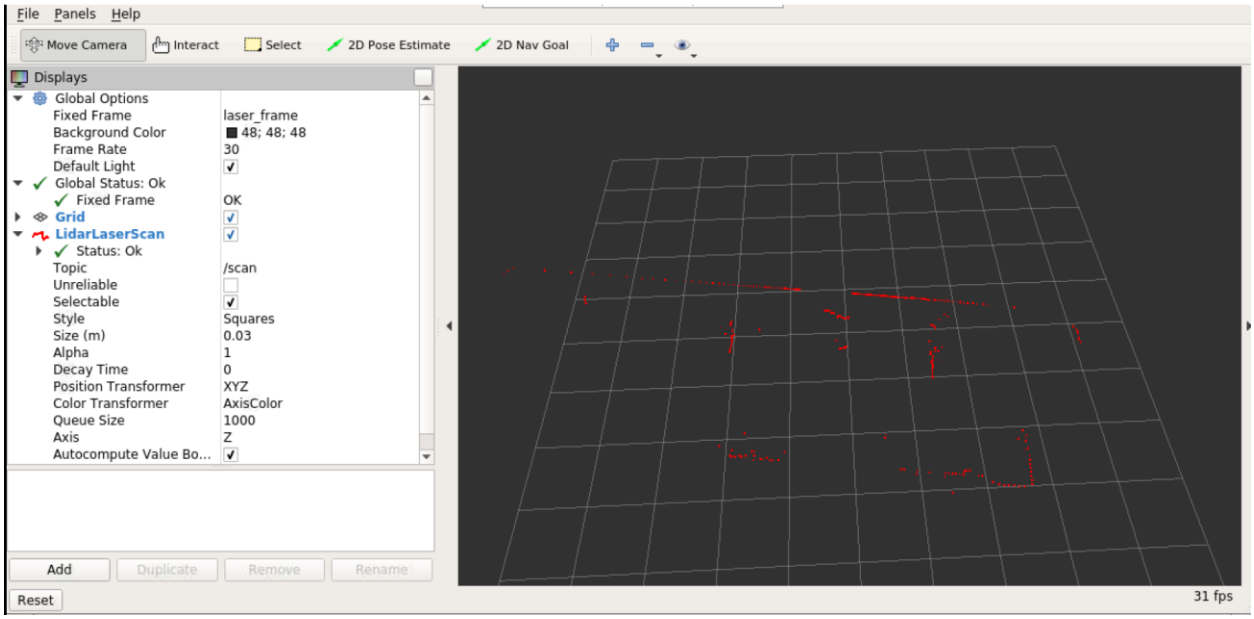
Primero se ejecuta Hector SLAM
hector_slam: Es un metapaquete que instala hector_mapping y paquetes relacionados para mapeo.

```
ubuntu@ubiquityrobot:~$ roslaunch hector_slam_launch tutorial.launch
```

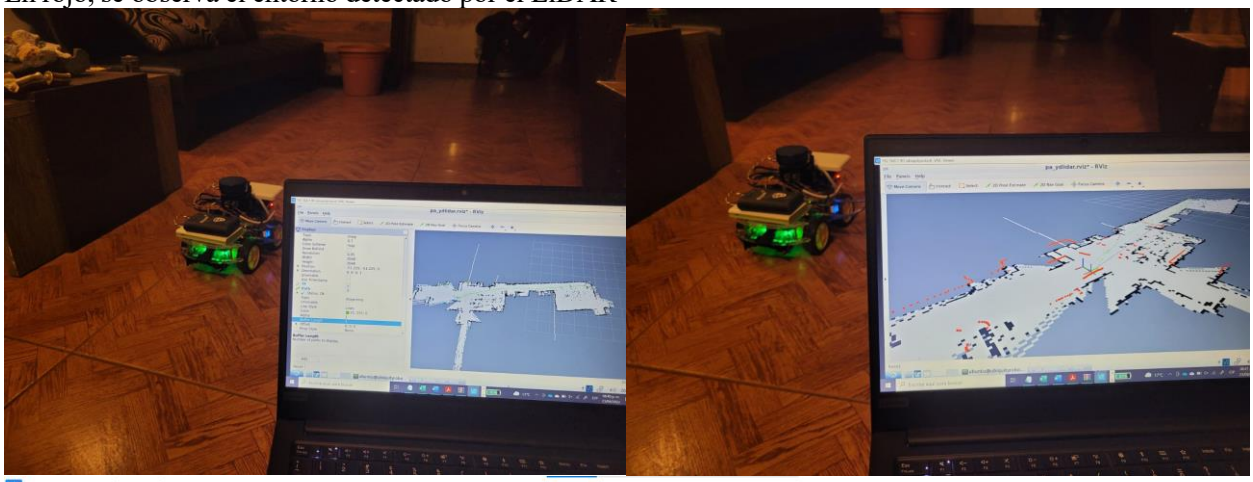
Luego se ejecuta ydlidar_ros X2L.lanuch
Levanta la interfaz de usuario de RViZ junto con los sensores asociados, en este caso el LiDAR.

RVIZ: ROS dispone de una herramienta de visualización 3D, llamada Rviz, donde se pueden representar robots, nubes de puntos, datos de sensores, etc. que gracias a estar basada en una arquitectura de plugins, nos permite no sólo desarrollar nuestras herramientas, si no reutilizar otras ya existentes.

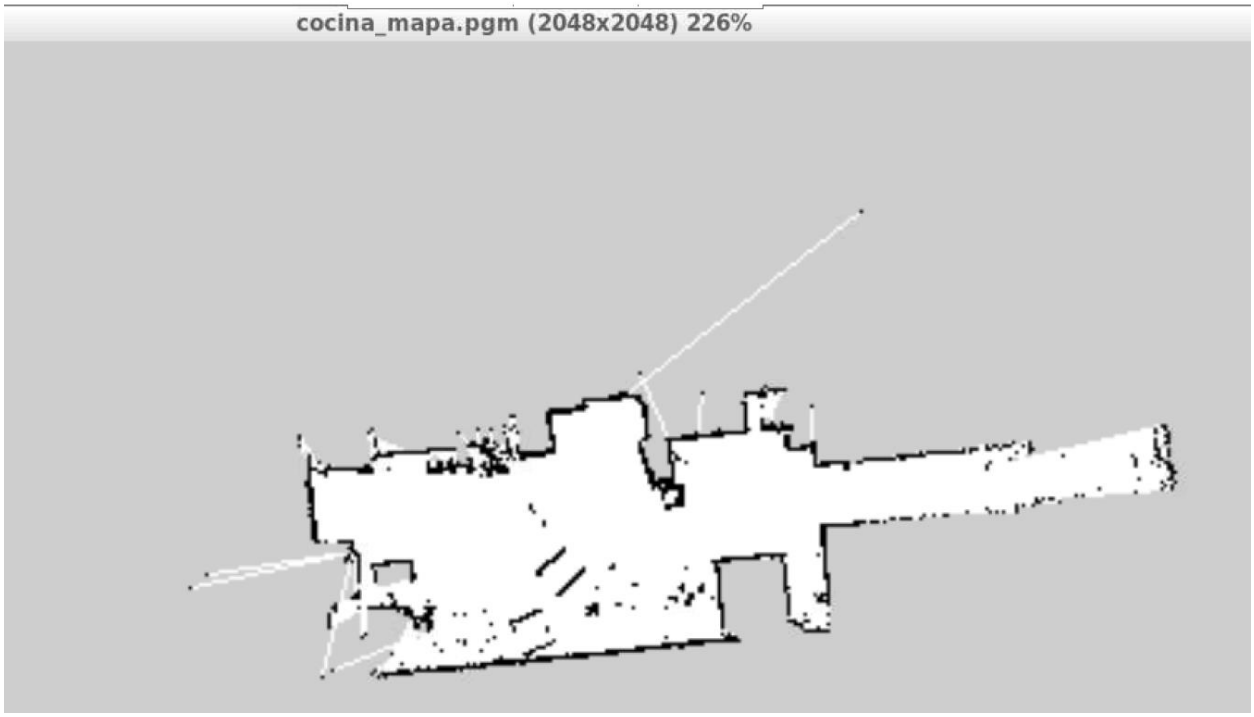
```
root@ubiquityrobot:/home/ubuntu# roslaunch ydlidar_ros X2L.launch
```



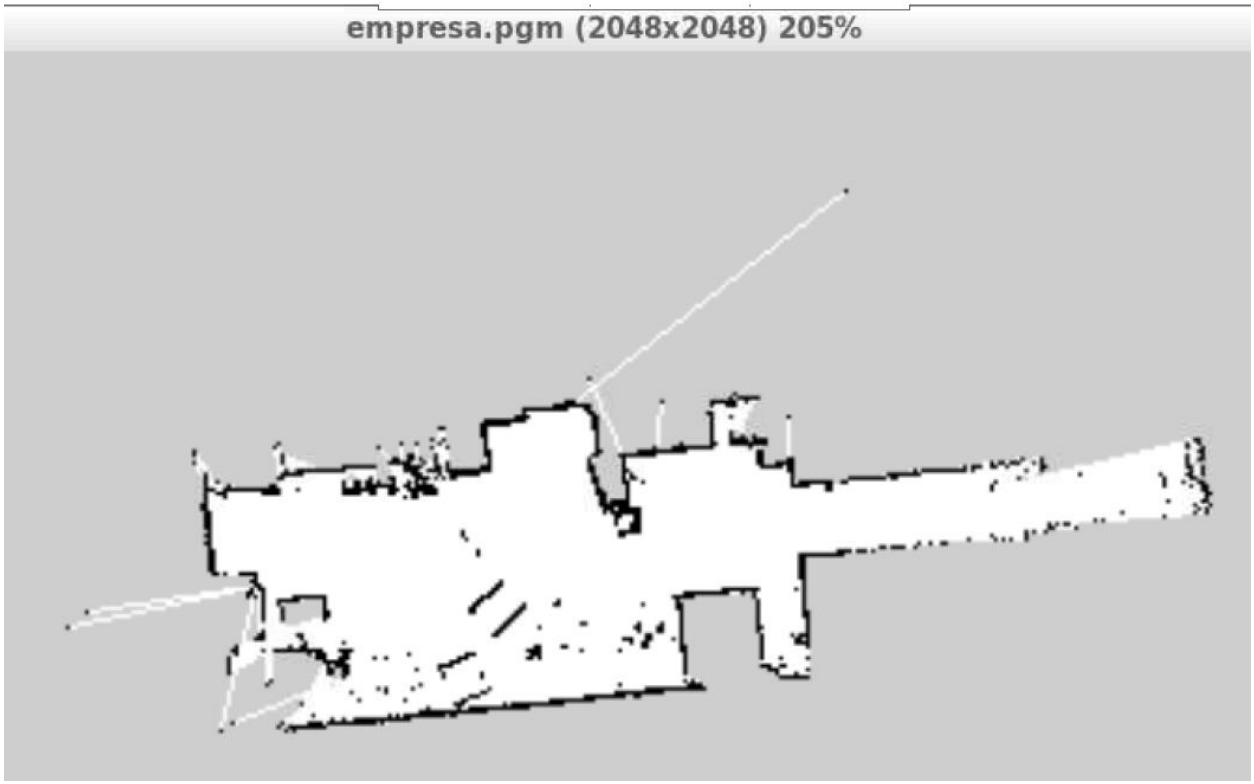
En rojo, se observa el entorno detectado por el LiDAR



En verde se observa la trayectoria del robot mientras realizó el mapeo



Otro mapa obtenido I



Otro mapa obtenido II

4.4.3 Desplazamiento Autónomo

Para el desplazamiento autónomo se deben ejecutar:

Rosserial:

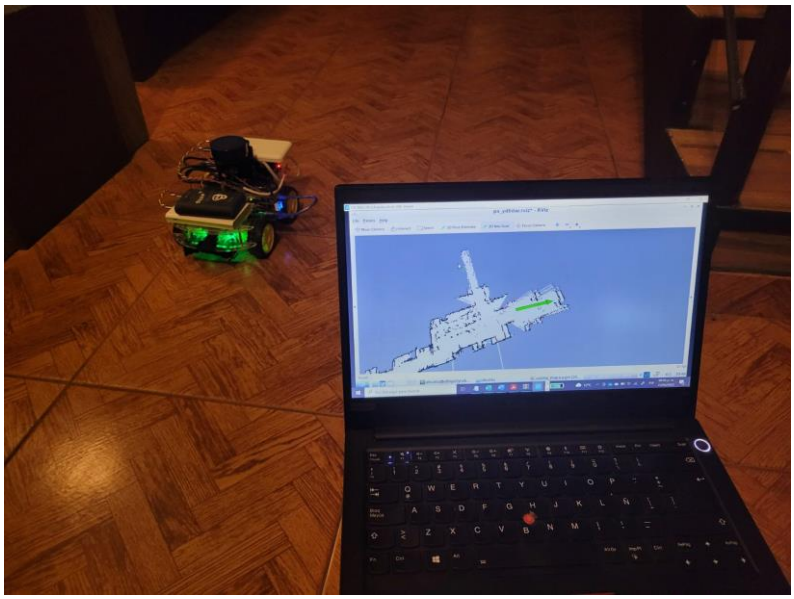
La biblioteca rosserial implementa un protocolo para empaquetar mensajes serializados estándares de ROS, además, permite multiplexar múltiples tópicos y servicios sobre dispositivos como puertos seriales o sockets.

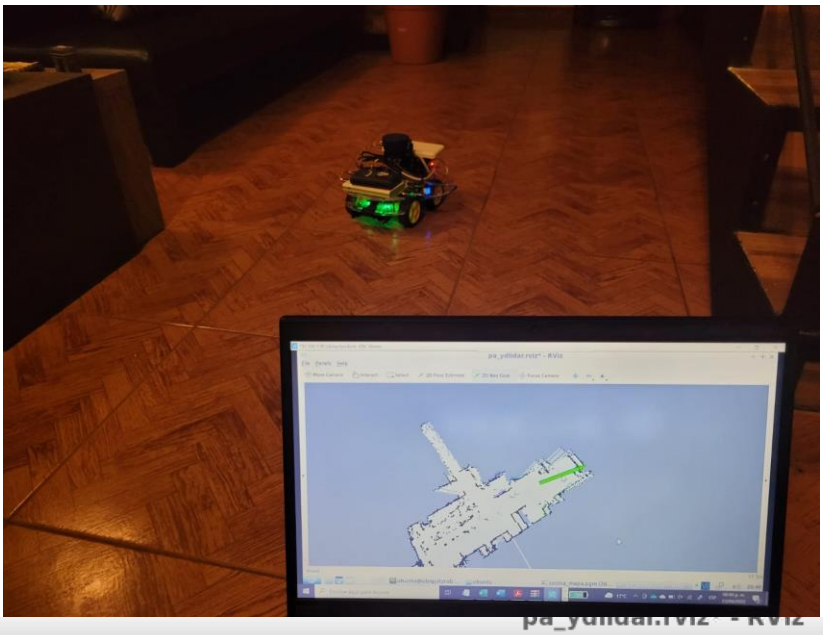
```
ubuntu@ubiquityrobot:~$ rosrun rosserial_python serial_node.py _po[3/3$
dev/ttyACM
[INFO] [1650757556.554667]: ROS Serial Python Node
```

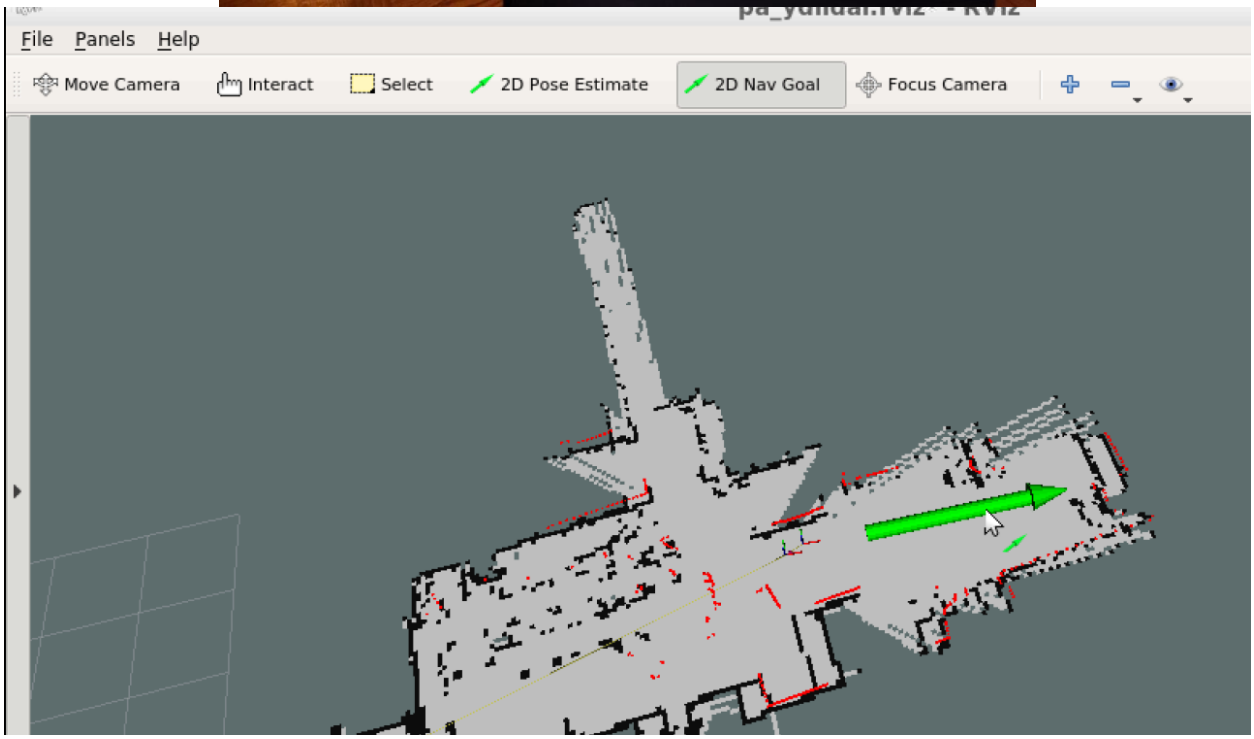
A continuación, se ejecuta el nodo move_basic
El nodo move_basic realiza tareas básicas de navegación. La planificación de la ruta consiste en girar en el lugar para mirar hacia la meta y luego conducir directamente hacia ella. Está diseñado para proporcionar las mismas interfaces de software que move_base, es decir, implementa un QueuedActionServer (similar a SimpleActionServer, consulte la documentación de actionlib), que pone en cola hasta dos objetivos que contienen mensajes geometric_msgs/PoseStamped.

Se supone que estamos tratando con datos de localización imperfectos: map->base_link es preciso pero puede retrasarse y tiene un ritmo lento y odom->base_link es más frecuente, pero se desvía, particularmente después de rotar. Para contrarrestar estos problemas, move_basic planifica en el marco del mapa, espera un poco después de cada paso y ejecuta el movimiento en el marco odom. Si se reciben objetivos en el marco base_link, se interpretan como relativos a la posición actual del robot. Este comportamiento es diferente al de move_base, que no aceptará objetivos en el marco base_link.

```
ubuntu@ubiquityrobot:~$ rosrun move_basic move_basic
[ WARN] [1650757487.366715495]: Parameter change detected
```



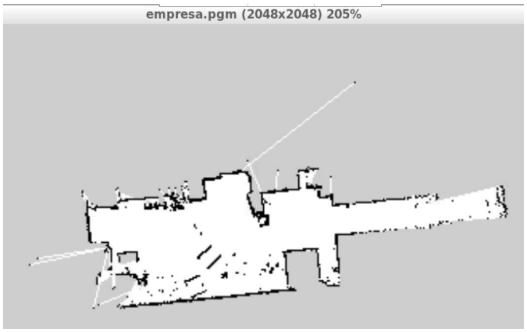



File
Panels
Help
Move Camera
Interact
Select
2D Pose Estimate
2D Nav Goal
Focus Camera

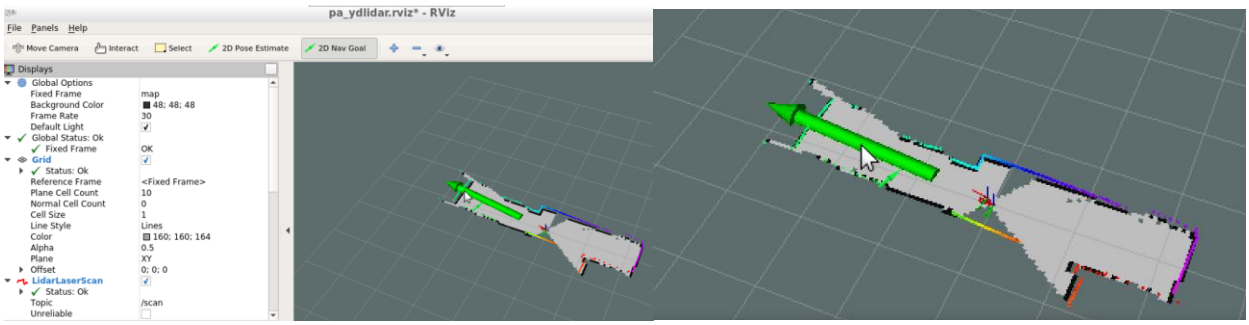
5 RESULTADOS

5.1 Análisis de resultados individuales y generales

El objetivo principal de este proyecto es el diseño y fabricación de un vehículo de bajo costo para implementación de las técnicas de SLAM, haciendo uso de un sensor LiDAR y utilizando herramientas de la plataforma ROS. Todo lo anterior, orientado a fines educativos con la intención de ser la base de trabajos posteriores orientados a elaboración de productos con fines industriales.
El hecho de no disponer plataformas similares para poder realizar comparativas, ni experiencias previas, enfoca el análisis de los resultados, a un contraste entre las expectativas al comenzar el proyecto versus los logros obtenidos. En ese sentido, se puede decir que la mayoría de las expectativas iniciales fueron alcanzadas, pues la plataforma es funcional y se ha logrado hacer los mapeos propuestos mediante el LiDAR instalado. Los mapas son satisfactorios y se lograron con las mismas herramientas que se pueden visualizar en la bibliografía oficial de ROS y experiencias similares volcadas en la WEB.



El movimiento autónomo del robot, se logró mediante el uso de la aplicación RVIZ de ROS, utilizando la herramienta “2D Nav Goal”, que permite marcar un punto en el mapa y mediante el algoritmo Move_basic, mueve el robot desde el punto donde se encuentra ubicado hasta el punto marcado. Mientras el robot se desplaza, se actualiza automáticamente su posición en el mapa. Cumpliendo con las premisas de definen a SLAM (Simultaneous Localization And Mapping).



También se podría analizar desde el punto de vista del aprendizaje. Esta plataforma permite ensayar, de forma práctica, una amplia variedad conceptos y herramientas orientados ROS, que, al momento de redactar este documento, no existe manera de ponerlos en práctica con los elementos que dispone la institución. A lo largo del desarrollo del proyecto se utilizaron las mismas herramientas que sugiere la bibliografía oficial de ROS, para depuración y prueba de cada módulo.

Es importante mencionar que quedaron expectativas sin alcanzar. El robot presenta limitaciones en los motores que se utilizaron. Dichos motores se eligieron en función de disponibilidad en comercios nacionales y de su costo reducido. Las especificaciones fueron analizadas al comienzo del proyecto con las hojas técnicas de los conjuntos (motor + caja reductora), era evidente de que dichas especificaciones, estaban muy próximas a los

mínimos aceptables. Lamentablemente las opciones más aconsejables estaban fuera del presupuesto y del tiempo disponible. Por lo que se decidió intentar con ellos. Si se analizan los resultados obtenidos, se podría decir que se obtuvieron resultados muy aceptables para la calidad de motores utilizados. Observando experiencias similares publicadas en la WEB, es notable la diferencia entre los motores que se utilizaron para lograr resultados similares a los obtenidos en el presente trabajo.
Como consecuencia de las limitaciones mencionadas, el robot presenta problemas de torque que le impiden ajustarse adecuadamente a trayectorias precisas. Recordando, ROS envía la trayectoria, en forma de velocidades lineales (m/s) y angulares (rad/seg), y el robot debe ser capaz de reaccionar a cada cambio de velocidad o de ángulo con cierta facilidad. En este caso, el robot debe estar compensando constantemente y los valores quedan fuera de rango a la hora de esquivar ciertos objetos o de moverse en espacios reducidos.

A partir del presente proyecto, se obtuvieron múltiples experiencias, que permitirán mejorar futuros diseños con los conocimientos adquiridos.

5.2 Análisis del alcance de los objetivos del proyecto

Los objetivos del proyecto fueron enunciados como objetivos primarios y objetivos secundarios.

- Objetivos primarios
  - Elaborar un vehículo autónomo de bajo costo: Existe una gran cantidad de factores que se deben tener en cuenta al comparar costos a nivel internacional con costos locales. Teniéndose en cuenta dichas asimetrías, se puede decir que el objetivo fue alcanzado. Los costos más elevados están en la CPU principal y el LiDAR. Ambos no podrían ser reemplazados sin una pérdida notable de prestaciones. Al contrario, la RaspBerry adquirida, permite, con un costo muy bajo, incorporar una cámara para el tratamiento de imágenes del entorno. También a lo largo del proyecto se elaboró un SONAR de 360° que se podría mejorar y completar para reemplazar al LiDAR, obviamente con prestaciones inferiores, pero con un costo significativamente menor.
  - Utilización de un sensor LiDAR para reconocimiento del entorno. Se logró exitosamente utilizar esta tecnología de última generación.
  - Mapeo e implementación de técnicas de SLAM. Se lograron con resultados similares a los publicados en trabajos similares consultados en la WEB.
  - Utilización de herramientas ROS. Se hizo a lo largo de todo el proyecto.
  - Autonomía completa, precisión para esquivar objetos y seguir trayectorias muy precisas. Esto se logró parcialmente debido a las causas enunciadas en el apartado anterior.
- Objetivos de segundo Nivel
  - Iniciarse y obtener habilidades en el uso de la plataforma ROS. Se obtuvieron valiosos concimientos y experiencias al respecto.
  - Conectar de manera inalámbrica con el robot. Se logró de diversas maneras:
    - a través de un software de escritorio remoto y de conexión WiFi, se accede a la CPU principal.
    - A través de un Joystick, mediante BlueTooth
    - A través del teclado y del mouse de una PC remota, se puede controlar el robot.
  - Control de la trayectoria del robot móvil mediante RViz. El movimiento autónomo del robot, se logró mediante el uso de la aplicación RVIZ de ROS, utilizando la herramienta “2D Nav Goal”, que permite marcar un punto en el mapa y mediante el algoritmo Move_basic, mueve el robot desde el punto donde se encuentra ubicado hasta el punto marcado.
  - Elaborar una plataforma en la que se trabaje de forma conjunta con las placas ATMega2560 y Raspberry Pi. El robot dispone de ambas CPUs las cuales se comunican mediante protocolos ROS encapsulados en RosSerial.
  - Desarrollo de odometría mediante la utilización de distintos sensores. El robot es capaz de medir su velocidad y de medir distancias de objetos en un radio de 8 metros (y 12 metros para el RPLiDAR). También posee sensores de aceleración, brújula y giróscopos.

Con lo referido se puede decir que el robot alcanzó prácticamtente todos los objetivos planteados al comienzo del proyecto.
Obviamente, desde la luz de la experiencia obtenida, surjen muchas ideas para mejorar y nuevos horizontes para alcanzar.

5.3 Ventajas y desventajas del diseño y construcción del dispositivo

5.3.1 Ventajas del diseño y construcción del dispositivo electrónico

- A nivel educativo, es una plataforma funcional que permite hacer prácticas sobre nuevas tecnologías a las que, prácticamente, no se tendría acceso de otra manera.
- La implementación de algoritmos SLAM, requiere de un sistema completo para los ensayos prácticos. El dispositivo provee de dicho sistema.
- Instalar el SO ROS en un proceso un tanto complicado y requiere, en este caso, de bastantes conocimientos de Linux. El disponer del SO previamente instalado y probado, permite enfocarse en ensayos específicos.
- Reutilizabilidad: Partiendo de este modelo funcional, es relativamente modificar ciertos bloques o funcionalidades. Por ejemplo, ROS también se utiliza en brazos robots. Por lo tanto, las 2 CPUs utilizadas en el proyecto, más la placa de control de motores instalada, podrían ser removidas y reinstaladas en un brazo robot. Simplificando gran parte del proceso de desarrollo.
- Hay experiencia muy valiosa adquirida a lo largo del diseño y la construcción del presente proyecto. A partir de ésta, se puede avanzar en modelos mejorados con orientación industrial.

5.3.2 Desventajas del diseño y construcción del dispositivo electrónico

- Los precios y la logística para la importación de ciertos componentes, complican demasiado la posibilidad de hacer un producto, de producción a escala y competitivo. Se puede pensar en desarrollar soluciones específicas y customizadas para ciertos clientes, lo cual genera un valor agregado, que puede compensar los precios y la logística. La evaluación económica financiera presentada se basa en soluciones específicas orientadas a la industria.

5.3.3 Análisis de factibilidad de producción y comercialización

Habría que considerar algunos aspectos, es una tecnología muy novedosa y que aún está en etapa de desarrollo e investigación a nivel mundial. Esto condiciona las demandas concretas y algunas cuestiones de integración dentro los sistemas actuales. Por lo tanto, se consideraron dos escenarios: un escenario actual, transitorio y en evolución a un nuevo escenario orientado a la industria 4.0.

Actualmente y considerando el mercado local, se puede decir que existe una demanda muy limitada de este tipo de productos (al nivel alcanzado en esta etapa). Pero es importante tener en cuenta, que la industria actual existe una gran necesidad de este tipo de productos. Esta necesidad se va a convertir en una demanda concreta, a medida que esta tecnología se encuentre más difundida localmente, que los precios sean más accesibles y que se encuentre mano de obra especializada para instalación, operación y mantenimiento de este tipo de dispositivos.

Considero el desarrollo realizado, como un módulo, que se puede producir y comercializar, para agregarle una valiosa funcionalidad, a vehículos industriales diseñados para tareas específicas. Existe una diversidad de tareas que pueden realizar este tipo de vehículos que no podrían ser alcanzadas por un único diseño o estructura. Sin embargo, el módulo presentado, sería prácticamente el mismo para todos los casos.

5.3.4 Influencia en el medio ambiente y la sociedad

- Este tipo de vehículos son enteramente eléctricos, su utilización a gran escala en fábricas y depósitos irá desplazando a los sampis tradicionales con motores de combustión.
- Los vehículos autónomos optimizan la productividad en almacenes y fábricas. Es probable que fábricas y almacenes en el futuro, necesiten menos superficie para producir lo mismo que actualmente. Lo cual evitaría de seguir avanzando innecesariamente sobre la naturaleza.
- Con el Covid 19 se presentó la necesidad de implementar cuarentenas y aislamientos. Aun así, la gente necesitaba alimentos, medicamentos y mercadería en general. Los robots podrían encargarse de preparar los pedidos en los grandes depósitos con mínima intervención humana, evitando algunos cuellos de botella en la cadena de distribución de elementos esenciales.

REFERENCIAS

1. Dissanayake, M. W. M. Gamini, Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*.

2. Coria, A. (2020). Aplicaciones de IoT para el mundo real. En mcelectronics.

3. Coria, A., Balsamo, A., Blaha, F., & Castellucci, I. (2021). Aplicaciones de IoT Industriales. En mcelectronics.

4. ROS Wiki. (s. f.). *Documentation*. Recuperado de http://wiki.ros.org/Documentation

5. Roboticstomorrow.com. (2015). Low-cost Lidar-Based Navigation for Mobile Robotics. Recuperado de https://www.roboticstomorrow.com/article/2015/11/low-cost-lidar-based-navigation-for-mobile-robotics/7270

6. Instructables.com. (s. f.). Using Raspberry Pi 4 With Ubuntu, ROS, Rplidar, Arduino. Recuperado de https://www.instructables.com/Using-Raspberry-Pi-4-With-Ubuntu-Ros-Rplidar-Ardui/

7. TizianoFiorenzani. (2017). ROS Tutorial: Understanding ROS Nodes. [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=Qrtz0a7HaQ4&t=373s&ab_channel=TizianoFiorenzani