



FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

TALLER 2

Juan Diego Acosta García

201812486

Luisa Fernanda Fuentes Melo

201631093

Diego Alejandro Rojas Sánchez

201812307

Laura Stephania Suárez Sanmiguel

201715140

Robótica IELE-3338

Juan José García Cárdenas

Bogotá, 1 de Abril del 2022

1 Diseño del Robot

1.1 Materiales

La lista de materiales que se emplearon en el desarrollo y diseño del robot fueron:

- Puente H- Referencia L293D
- Tarjeta Raspberry Pi 4
- Motores 9 voltios
- Rueda loca metálica
- Placa de acrílico: 2mm de espesor
- Pilas Lipo 7.4V
- Regulador 5V

1.2 Planos Mecánicos

Por medio del Software Fusion 360 se realizó el montaje CAD del robot diferencial diseñado, el cual se puede observar en la figura 1. De esta manera, se logró obtener el plano mecánico que se observa en la figura 2, con el cual se da evidencia a la geometría del robot especificando las medidas o posiciones de cada uno de los objetos presentes.

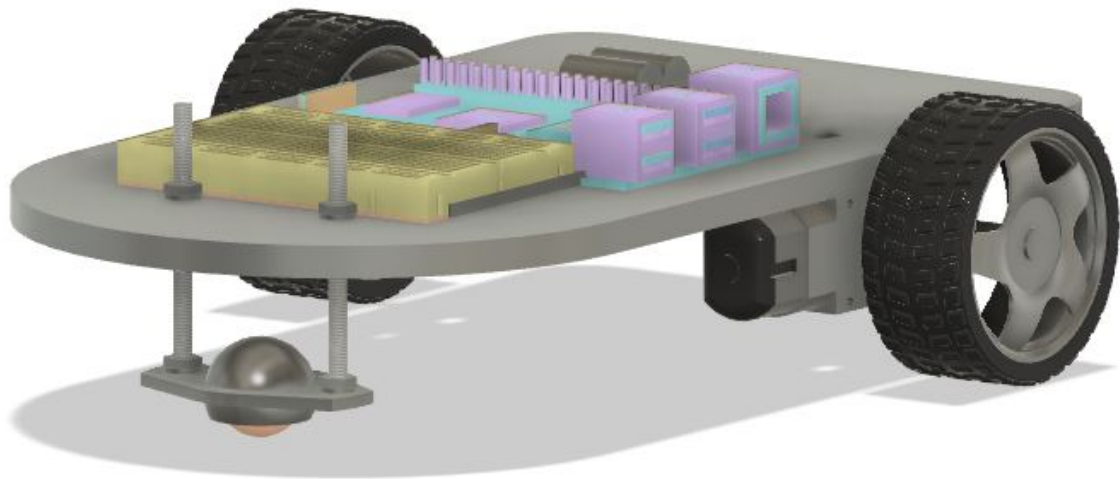


Figura 1: Vista isométrica del robot

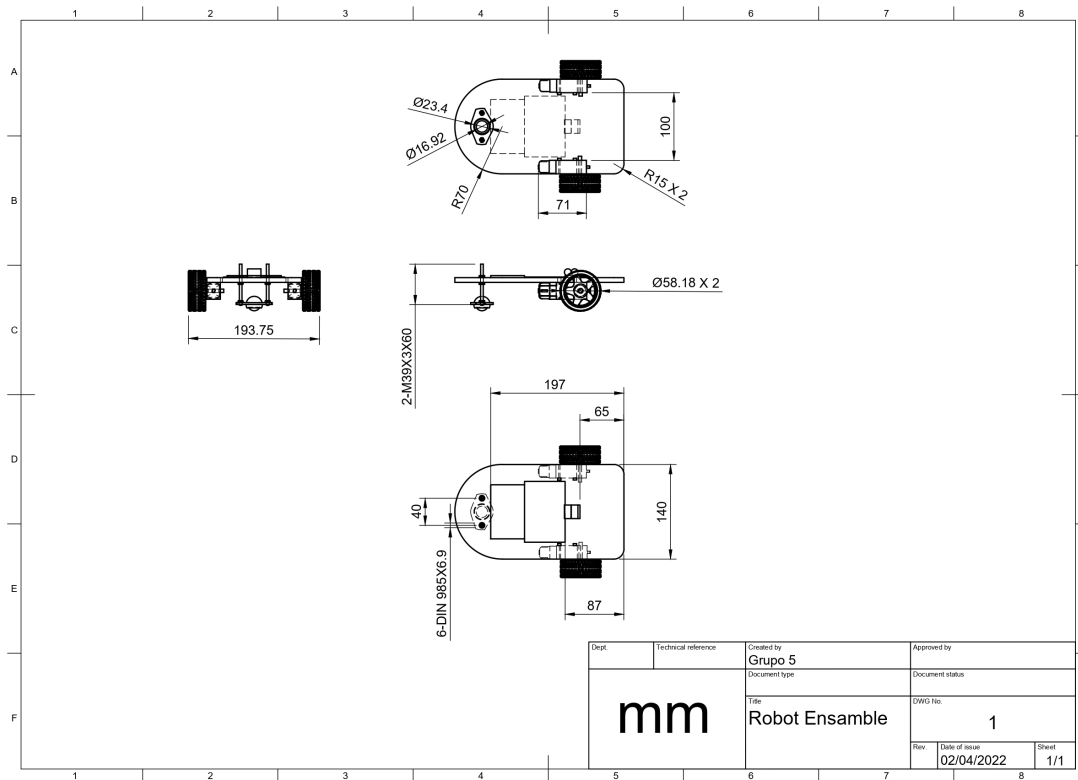


Figura 2: Planos Mecánicos del montaje

Finalmente, se puede observar en la figura 3 el plano explosionado del robot, con el que se pretende dar evidencia al ensamblaje de las partes del robot. De la misma manera, se pueden observar cada uno de los elementos que se utilizaron para la construcción de este.

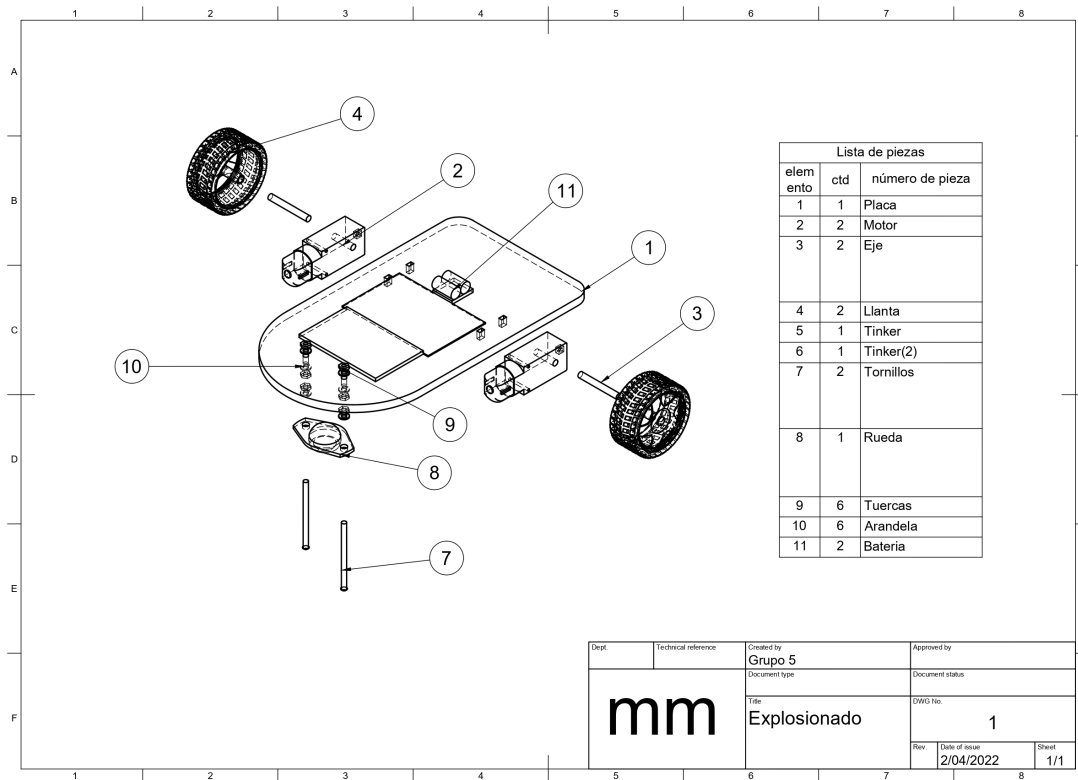


Figura 3: Planos del explosionado del robot

1.3 Planos Electrónicos

A continuación, observamos el plano electrónico del montaje entre la tarjeta *Raspberry Pi* 4, los motores, el puente H, el regulador de voltaje y la batería. Es importante destacar que según el puente H, hay unos pines que corresponden a los *Enable* los cuales son los encargados de variar la velocidad de los motores. Para el adecuado funcionamiento de estos se debe tener en cuenta emplear los pines PWM (señal cuadrada de reloj) a la cual se puede ir alterando su frecuencia para controlar la velocidad de los motores. Otras conexiones presentes son los pines de entrada y salida teniendo en cuenta el datasheet del puente H y la tarjeta programable.

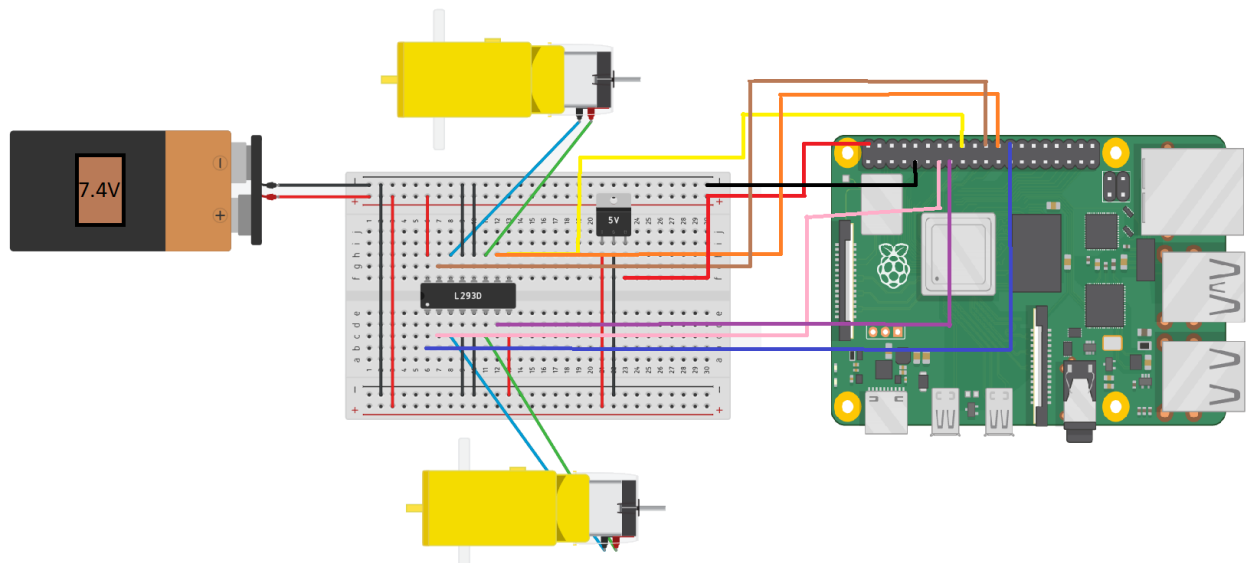


Figura 4: Planos Electrónicos del montaje

2 Desarrollo

2.1 Punto 1.

Para identificar cuándo las teclas del teclado son presionadas para indicar el movimiento deseado se crea un primer nodo *robot_bot_teleop* que es de tipo publisher. Allí, se utilizan las librerías *sys* y *termios*, las cuales permiten primeramente obtener la información que llega a consola y en segundo lugar, solo permite la acción de una sola tecla por iteración. Una vez se identifica qué tecla que fue presionada, se modifica el valor de la velocidad lineal o angular, según el caso, y se publica el mensaje. Este mensaje es enviado a través del tópico *robot_cmdVel*. Esto es percibido por el nodo *robot_central* el cual hace las veces de nodo suscriptor, en este nodo se emplea *reverse* y *forward*, las cuales son funciones que permiten la transmisión de la señal a los motores del robot. Cabe resaltar, que este nodo *robot_central* toma los datos sobre la velocidad lineal y angular del robot ingresados para procesarlos y evidenciar su equivalencia en velocidades angulares de cada llanta. Esto con el fin, de obtener el valor de la señal de 0 a 100 que acepta cada uno de los motores equivalentes a su velocidad angular.

Con la información de la velocidad el robot realiza su movimiento en tiempo real mientras se presionan las teclas. Esta etapa de envío se puede observar en el grafo de ROS:



Figura 5: Grafo de ROS del punto 1.

Este grafo permite apreciar la conexión entre el *robot_bot_teleop* y el nodo *robot_central* con el tópico previamente mencionado. Por otro lado, se imprimen en la terminal los valores de velocidad que son enviados desde el nodo teleop hacia el nodo central de conexión con el robot físico. Se puede observar en la figura ?? las 4 variaciones de velocidad lineal y angular que se envían al simulador al presionar las teclas: 'a', 'w', 's' y 'd' respectivamente.

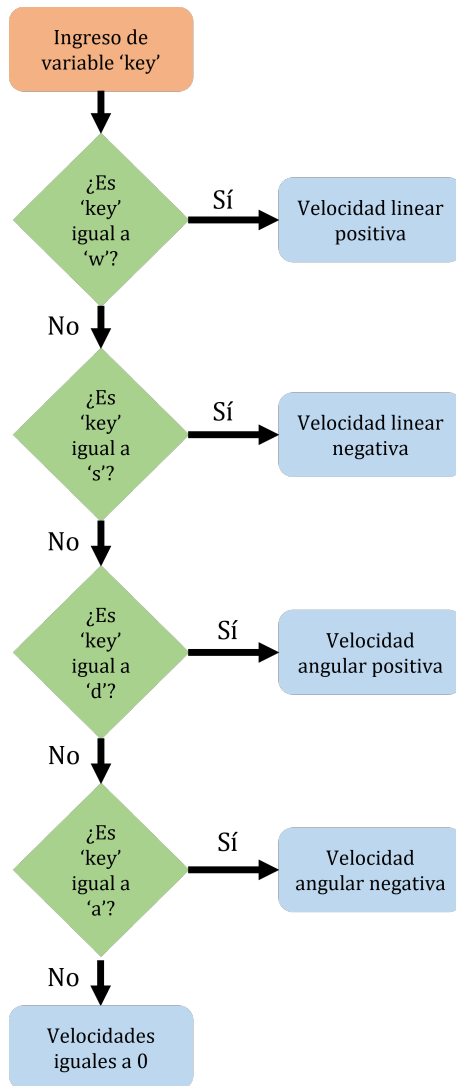


Figura 6: Algoritmo de cambio de velocidad en el mensaje a enviar por parte del Teleop

2.2 Punto 2.

Para el desarrollo de este punto, se crea un nuevo nodo llamado *robot_bot_interface* de tipo suscribir. Esta interfaz recibe la posición del robot por medio del tópico *robot_central*, el cual actúa como un nodo publisher para esta sección. En donde, se usa un proceso de odometría se obtienen las posiciones instantáneas del robot basados en las velocidades lineales de cada llanta. Con lo anterior, se puede generar el movimiento del robot en la interfaz y se brinda la posibilidad de guardar un archivo *.png* que tenga la gráfica de los movimientos del robot.

Así, en la figura 7 se puede observar la interfaz del sistema donde se puede evidenciar los botones de guardar y pausar. Esto con el fin de extraer la gráfica de la interfaz en formato *png* y obtener el recorrido realizado en un formato *txt*. De la misma manera, se puede evidenciar un espacio para ingresar el nombre que se desea tener como título de la gráfica como el nombre del archivo *.png* y *.txt*.

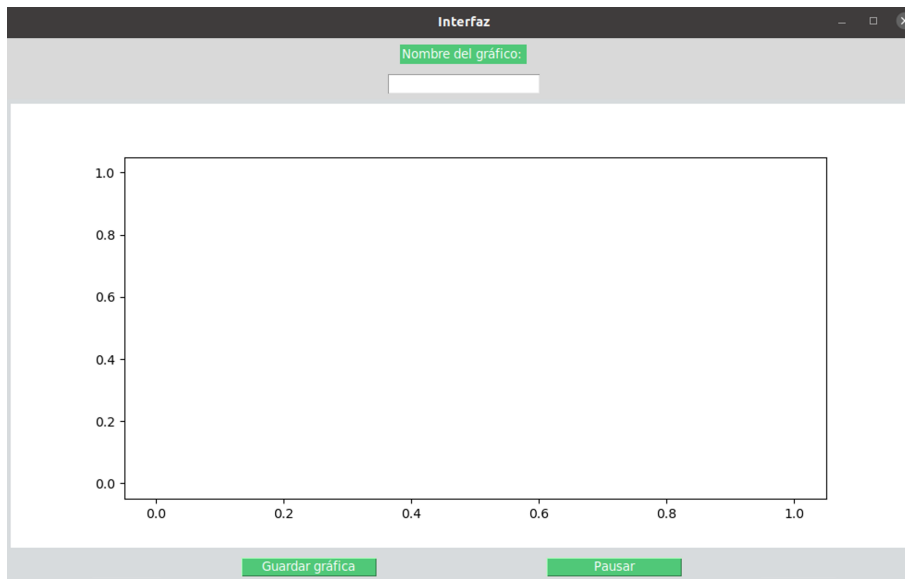


Figura 7: Interfaz diseñada inicialmente

En términos de resultados, se implementó un recorrido y se guardó en la casilla de la interfaz con el nombre GrafoPrueba1. Este recorrido se puede observar en la figura 8 y se guarda en la carpeta de Results del paquete de trabajo. En este gráfico se evidencia con claridad el recorrido del robot sobre un marco de referencia global generados por la odometría del nodo *robot_central*.

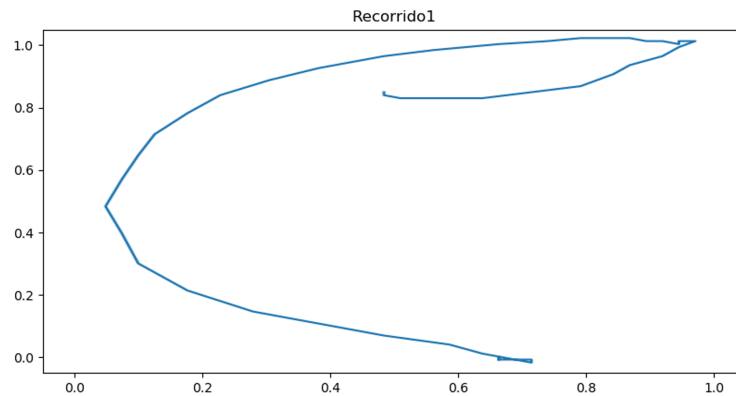


Figura 8: Gráfico guardado por la interfaz de un recorrido

Por otro lado, el grafo de ROS de este proceso se encuentra en la figura 9. Allí se aprecia cómo mediante el tópico **robot_cmdVel** se envían los valores de velocidad lineal y angular deseados al nodo *robot_central*. Y cómo el nodo *robot_central* a su vez realiza un envío de datos de posición lineal y angular a la interfaz diseñada mediante **robot_position**.

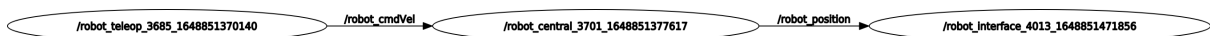


Figura 9: Grafo de nodos y tópicos con interfaz activa

En estas imágenes se puede observar que el correcto uso de los tópicos permite hacer una

comunicación sencilla de los mensajes entre nodos. En este caso, aunque los datos son muchos y variados, su envío y recibimiento es sencillo debido a que se utiliza el tipo de dato Twist donde se contienen todas las variables de velocidad o posición.

2.3 Punto 4.

Para llevar a cabo este punto, fue necesario crear el servicio *server*. Con este, se ingresa el archivo *.txt* con la memoria de movimientos realizados por el usuario y el robot para ingresarlos en un algoritmo similar al de ingreso de teclas por teclado utilizado en *robot_bot_teleop* y se envía nuevamente la información de velocidad a *robot_central* que es el encargado de leer la información y mover el robot físico, esto nuevamente envía el mensaje con la posición del robot y se grafica en la interfaz. Buscando que la relación entre el recorrido realizado con el nodo *robot_bot_teleop* y el recorrido replicado por el servicio fuera la mayor posible, se estableció un *rospy.Rate(28)* dentro del servidor. Esto debido a que se observó una diferencia de velocidades de lectura de datos en ambos procedimientos. De manera similar, se agregó un *time.sleep(3)* dentro del servicio al momento en que se recibía un valor de velocidades 0 (caso en el que el robot recibe la instrucción de quedarse quieto), ya que el servidor no cuenta con la posibilidad de identificar cuanto tiempo se estuvo sin oprimir una tecla simplemente con la lectura de la lista, por lo que se aplicó un delay al recibir la instrucción de detenerse pensando en el comportamiento inercial del sistema.

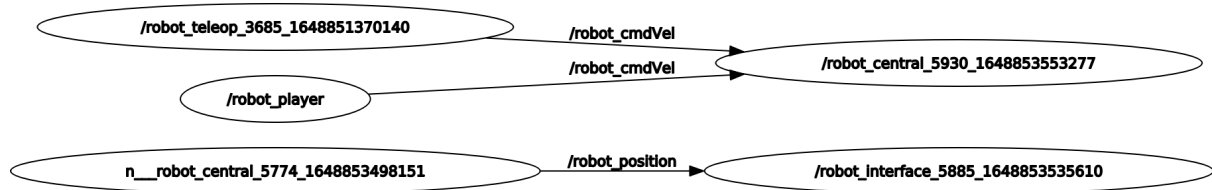


Figura 10: Grafo de nodos y tópicos del sistema completo.