



FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

## **TALLER 3**

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, 7 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 [1]
- Tarjeta Raspberry Pi 4 [2]
- Motores 9 voltios [3]
- Rueda loca metálica
- Placa de acrílico: 2mm de espesor
- Pilas Lipo 7.4V
- Regulador 5V [4]
- Motores Servo SG90 de 4.8-7.2V [5]

## 1.2 Planos Mecánicos

Por medio del Software Fusion 360 se realizó el montaje CAD del manipulador, 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 manipulador especificando las medidas o posiciones de cada uno de los objetos presentes.

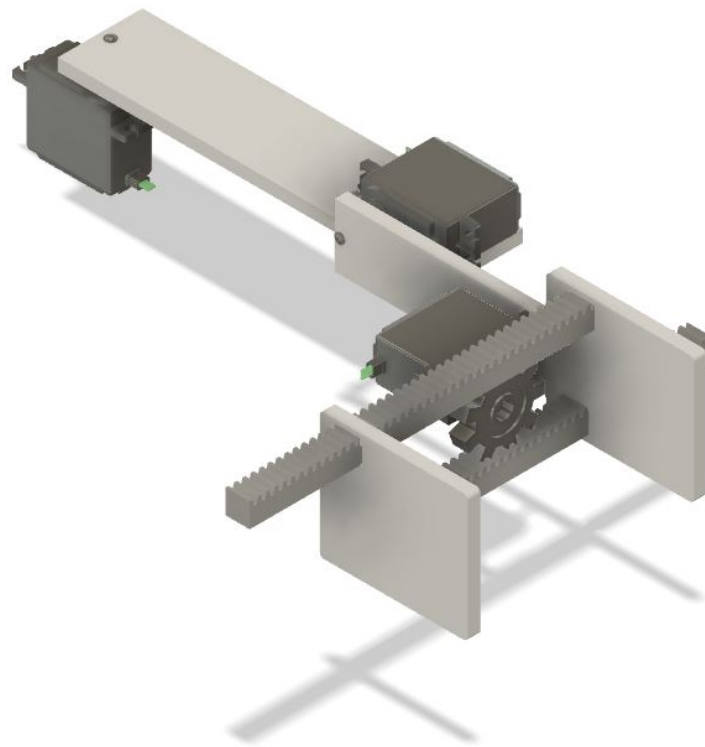


Figura 1: Vista isométrica del manipulador

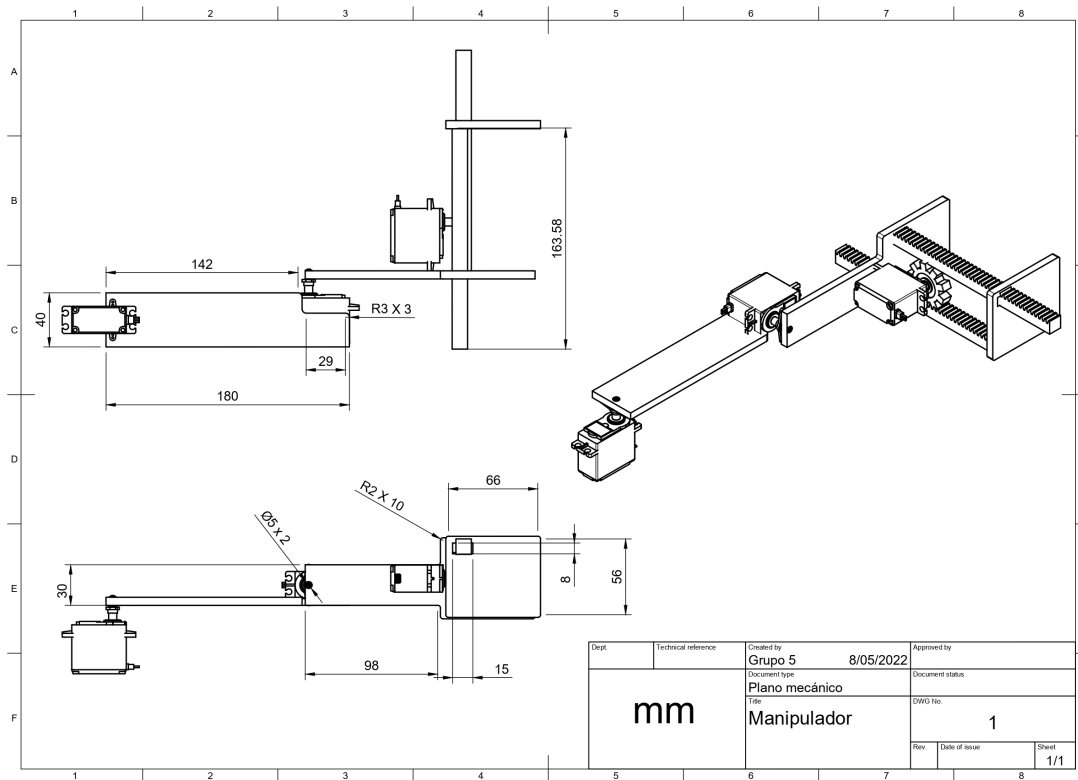


Figura 2: Planos Mecánicos del montaje

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

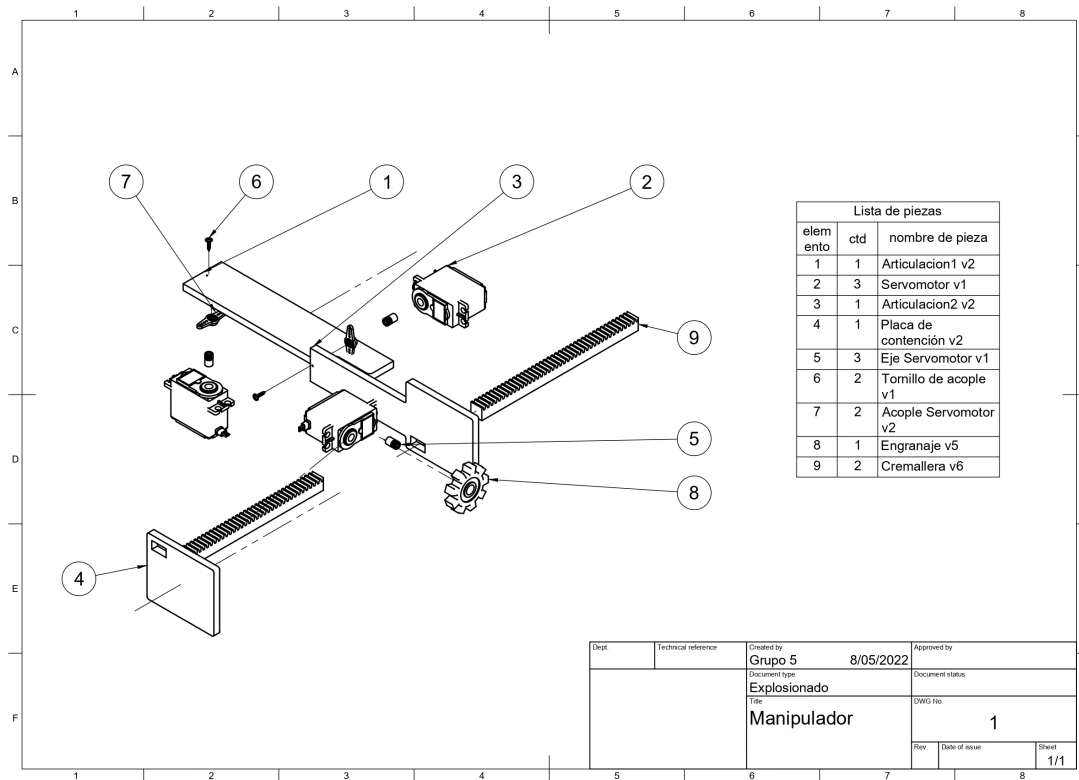


Figura 3: Planos del explosionado del manipulador

### 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.

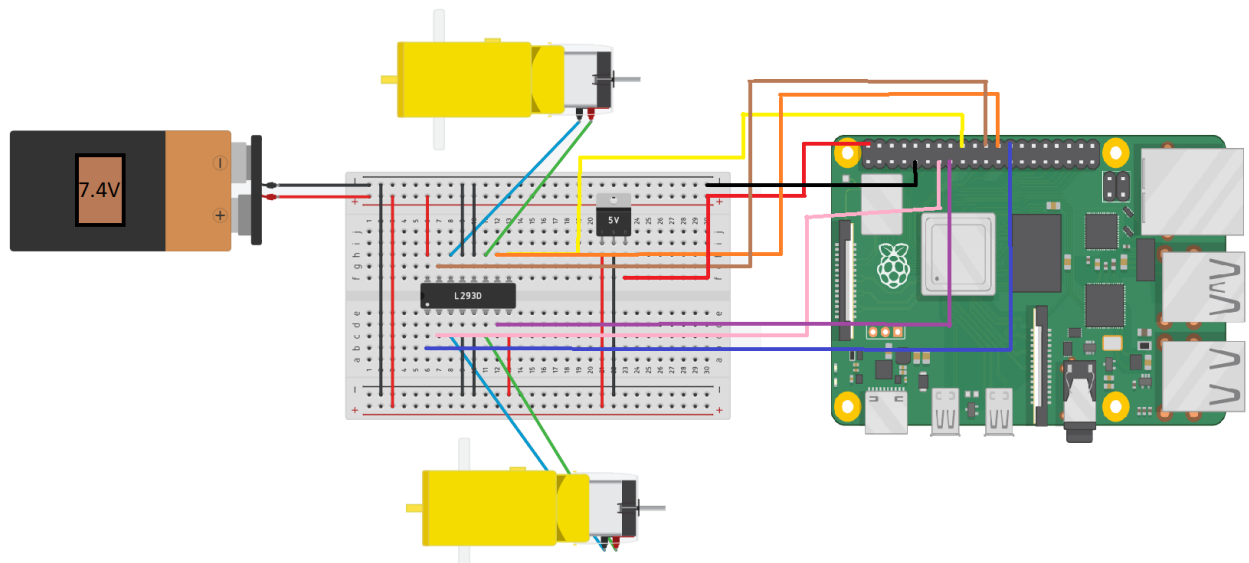


Figura 4: Planos Electrónicos del montaje del Robot

De igual manera, se observa a continuación las conexiones de los Servomotores con la tarjeta *Raspberry Pi 4*. Estos permiten principalmente el movimiento del manipulador diseñado.

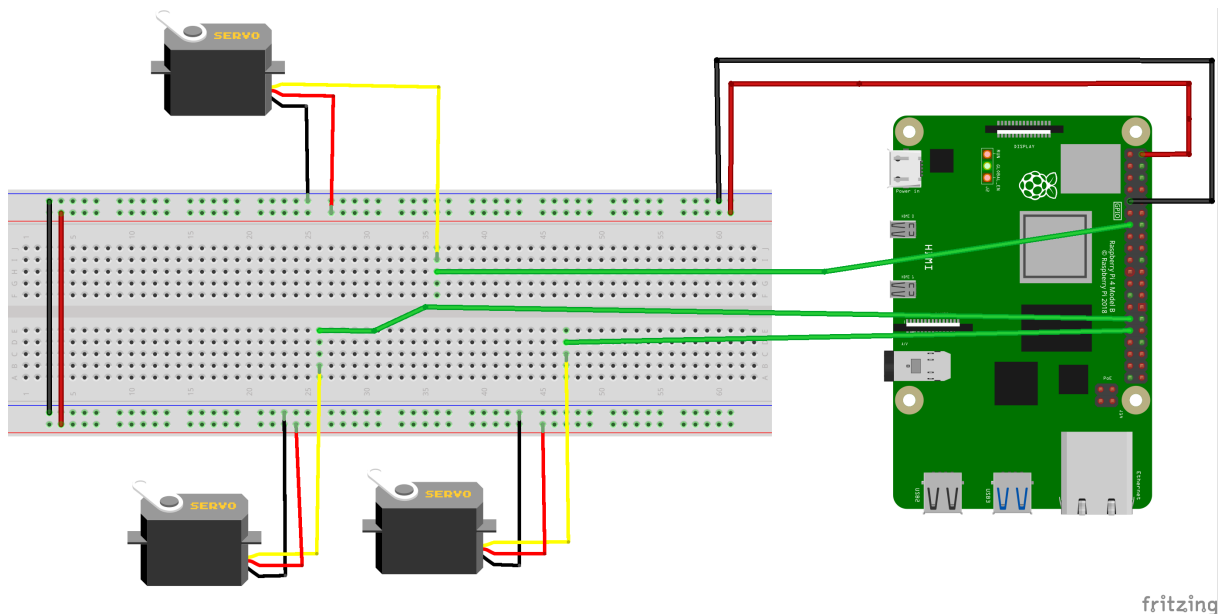


Figura 5: Planos Electrónicos del montaje del Manipulador

## 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_manipulator_teleop` que es de tipo publisher. Allí, se utilizan las funciones `on_press(key)` y `on_release(key)`. Una vez se identifica qué tecla

fue presionada, se modifica el valor de la velocidad de cada *joint*, según el caso, y se calcula el ángulo del movimiento que debe realizar cada uno de los servomotores.

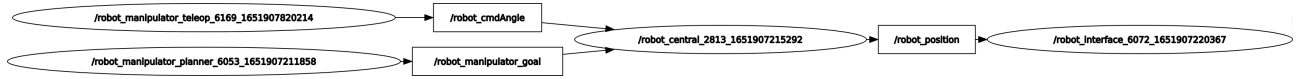


Figura 6: Grafo de ROS del punto 1.

El diagrama de flujo que se muestra a continuación define detalladamente el proceso de elección que computa nuestro algoritmo para mover el manipulador.

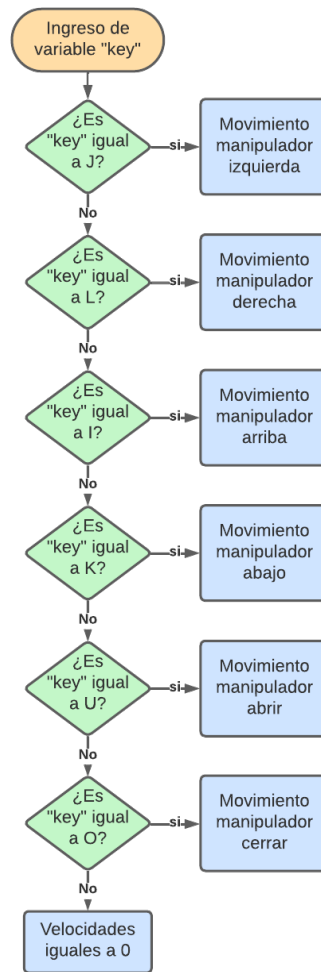


Figura 7: Algoritmo de cambio de velocidad en cada joint

## 2.2 Punto 2.

Para el desarrollo de este punto, se crea un nuevo nodo llamado */robot\_manipulator\_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 para obtener las posiciones instantáneas del robot basados en las velocidades de cada *joint*. 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, se puede observa 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*.

Por otro lado, se aprecia cómo mediante el tópico **robot\_cmdAngle** se envían los valores del ángulo de los motores de cada uno de los *joints* deseados al nodo *robot\_central*. Y cómo el nodo *robot\_central* a su vez realiza un envío de datos de posición en *x*, *y* y *z* del ultimo *joint* frente al marco global a la interfaz diseñada mediante **/robot\_manipulator\_position**.

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 Vector3 donde se contienen todas las variables de velocidad o posición.

### 2.3 Punto 3.

Los cálculos realizados para la definición del volumen del trabajo se tuvo en cuenta que todos los puntos  $(x, y, z) \in R^3$  que cumplan las siguientes condiciones:

$$\begin{aligned} x &= (r_2 \cdot \cos(\Phi) + r_1) \cdot \cos(\theta) \\ y &= (r_2 \cdot \cos(\Phi) + r_1) \cdot \sin(\theta) \\ z &= (r_2 \cdot \sin(\Phi)) + Z_{inicial} \end{aligned} \quad [2.1]$$

Con  $\theta$ ,  $\Phi$  ángulos de  $45^\circ$  a  $135^\circ$  y  $0^\circ$  a  $90^\circ$  respectivamente y con  $r_1$  como radio del brazo 1 correspondiente a 15.8 cm y  $r_2$  radio del brazo 2 que tiene una medida de 14 cm.

Esto permitió calcular el volumen de trabajo, el cual se expone en la figura 8.

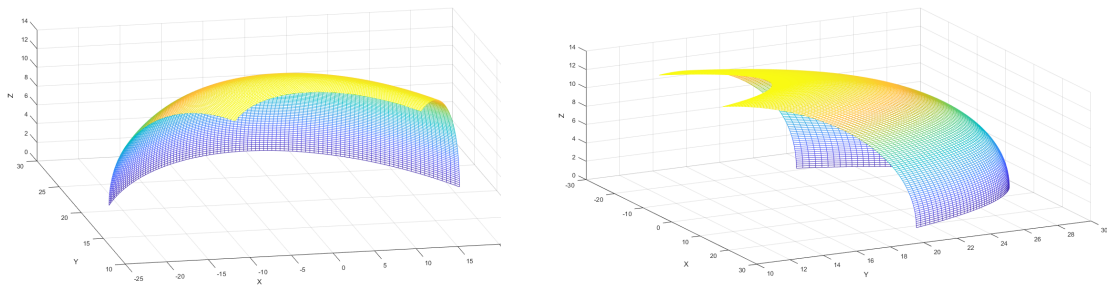


Figura 8: Volumen de trabajo del manipulador

Este volumen de trabajo se ve verificado mediante la gráfica de recorrido del end-effector que se puede observar en la figura 9.

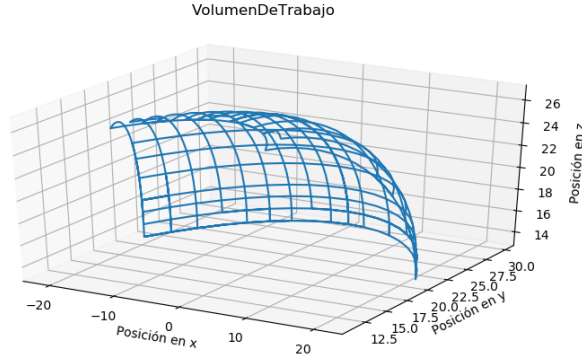


Figura 9: Volumen de trabajo obtenido de manera experimental

El conjunto de ecuaciones 2.1 se permitió definir  $\theta$  y  $\Phi$  con  $(x, y, z)$  dados:

1<sup>ro</sup> Hallar  $\Phi$ :

$$\Phi = \sin^{-1} \left( \frac{Z - Z_{fijo}}{r_2} \right)$$

2<sup>do</sup> Hallar  $\theta$  con  $x$  ó  $y$ :

$$\theta = \sin^{-1} \left( \frac{y}{r_2 \cdot \cos(\Phi) + r_1} \right) = \cos^{-1} \left( \frac{x}{r_2 \cdot \cos(\Phi) + r_1} \right)$$

Estas expresiones son utilizadas para enviar un Vector3 con la ubicación deseada en el tópico `/robot_manipulator_goal`. Este envía la posición deseada

## 2.4 Punto 4.

Para el reconocimiento de los pimpones de colores, se realizó un código que permite obtener imágenes a partir de un vídeo y procesarlas primero transformando en el espacio de color adecuado para su lectura en este caso HSV [6]. Este espacio es el más adecuado dado que sus componentes son Matiz, Saturación y Brillo y es más sencillo determinar los colores a detectar. Para esto se emplea principalmente el componente H que corresponde al matiz [6]. Este proceso fue realizado gracias a la referencia [6] el cual nos brinda un código base para implementar un proceso de visualización, detección y visualización con máscaras del color deseado.

De igual manera, nos basamos en un código empleado para la detección de color a tiempo real que permite generar un contorno al rededor del objeto de interés y encontrar el punto central de este para facilitar el proceso del manipulador en seleccionar, moverse y levantar el pimpón del color indicado por parámetro [7][8]. Este código se realizó gracias a la referencia [7] que nos brinda el proceso antes mencionado que combinado con el anterior permite una mejor identificación de los pimpones en la prueba a realizar.

## References

- [1] "Datasheet l293d puente h", ArduinoQue, 2021. [Online]. Available: <https://arduinoque.com/arduino/datasheet-l293d-puente-h/>.



- [2] "Raspberry Pi 4 Model B DATASHEET", Datasheets.raspberrypi.com, 2019. [Online]. Available: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>.
- [3] "Motorreductor 3v-9v con llanta 65mm / Motor con caja reductora - Tecnopura", Tecnopura. [Online]. Available: <https://www.tecnopura.com/producto/motorreductor-3v-9v-con-llanta-65mm-motor-con-caja-reductora/>.
- [4] "LM7805 Datasheet(PDF) - Fairchild Semiconductor", Alldatasheet.com, 2001. [Online]. Available: <https://www.alldatasheet.com/datasheet-pdf/pdf/82833/FAIRCHILD/LM7805.html>.
- [5] "SERVO MOTOR SG90 DATASHEET", Ee.ic.ac.uk. [Online]. Available: [http://www.ee.ic.ac.uk/pcheung/teaching/DE1\\_EE/stores/sg90\\_datasheet.pdf](http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf).
- [6] G. Solano, "Detección de colores en OpenCV – Python (En 4 pasos)", Omer-va, 2019. [Online]. Available: <https://omes-va.com/deteccion-de-colores/>.
- [7] L. del Valle Hernandez, "Detección de movimiento con OpenCV y Python", Programar fácil con Arduino, 2018. [Online]. Available: <https://programarfácil.com/blog/vision-artificial/deteccion-de-movimiento-con-opencv-python/>.
- [8] "OpenCV: Contours : Getting Started", Docs.opencv.org. [Online]. Available: [https://docs.opencv.org/4.x/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html).