

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e Implementación de un Sistema de Control para un
Brazo Robótico asistencial de 6 grados de libertad para
Procedimientos de Cirugías para el Diagnóstico de Epilepsia**

Trabajo de graduación presentado por Dennet Orbaugh Antillón para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2021

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e Implementación de un Sistema de Control para un
Brazo Robótico asistencial de 6 grados de libertad para
Procedimientos de Cirugías para el Diagnóstico de Epilepsia**

Trabajo de graduación presentado por Dennet Orbaugh Antillón para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,


2021


Vo.Bo.:

(f) 
M. Sc. Renato Conedera

Tribunal Examinador:

(f) 
M. Sc. Renato Conedera

(f) 
MAEB. Pablo Mazariegos

(f) 
MBA. Robert Fuertez

Fecha de aprobación: Guatemala, 20 de enero de 2021.

Inspiración

El ser humano es un animal extraordinario que ha creado invenciones tecnológicas impresionantes que le han ayudado a mejorar su vida, ya sea que se estudie desde el comienzo de la raza humana en la época de la prehistoria cuando se empiezan a generar avances como el uso de utensilios como piedras filosas para poder cortar y cazar animales para obtener una fuente de alimentación o nos adelantemos algunos millones de años cuando surge la implementación de comunidades para generar seguridad, las cuales llegan a dar lugar a grandes imperios como el Imperio de Alejandro Magno o el Imperio de Roma, por mencionar algunos. Estos imperios permiten un florecimiento de ideas que ayudan a propulsar el conocimiento de la raza humana. Sin embargo, es hasta que se crean las repúblicas, comenzando por Estados Unidos en 1776, cuando se da total libertad al individuo para actuar bajo su mejor juicio que se empiezan a demostrar avances tecnológicos impresionantes, como la máquina de vapor, que genera una revolución industrial fijando un sendero de conocimiento, el cual no es dictado por el trabajo del hombre, sino que acelerado por el ritmo de las máquinas comenzando así una etapa en el mundo de innovación audaz. Algunos individuos explotaron estas invenciones para generar atrocidades, llevando al mundo al borde de la guerra, sin embargo, existen otros que eligieron otro camino y lograron generar una perspectiva única en el mundo, llenándolo de asombro a través de sus obras, generando un cambio de ideas para poder mejorar la raza humana y así poder innovar nuestra forma de pensar. Dicha innovación impulsa hazañas impresionantes, como el poner a un hombre en la Luna o la creación de aparatos que pueden realizar cálculos complejos en cuestión de fracciones de segundo mejorando la calidad de vida de todos los individuos, lo cual es una evolución importante para poder llegar a mejorar a la raza humana. Es por eso, que fui inspirado a trabajar en este proyecto de graduación, para poder aportar a los avances tecnológicos y poder ayudar a mejorar la calidad de vida en Guatemala.

Agradecimientos

Agradezco a mi familia, sin la cual no hubiera logrado estudiar y terminar una educación en ingeniería mecatrónica en la Universidad del Valle de Guatemala. A mis padres, Warren y Claudia, quienes me apoyaron económicamente, siempre confiaron en mí y me han motivado a través de su ejemplo a superarme estudiando para mejorar mis habilidades y conocimientos. A mis hermanos Delaney, Derek y Deryn, quienes me han demostrado la posibilidad de superarse y seguir sus sueños. A Donald, quien me ha mostrado a mantener siempre un pensamiento positivo a lo largo de mi vida y me ha apoyado a lo largo de la carrera universitaria. Y a Ana Lucía, quien me apoyó y motivó en mis últimos años universitarios a continuar esforzándome para lograr obtener los conocimientos y alcanzar mis metas.

Dedicatoria

Este trabajo de graduación es dedicado a mis padres, Warren y Claudia, quienes han sido guías en mi vida y me han enseñado los valores y principios para vivir una vida plena y llena de alegría.

Prefacio	VI
Lista de figuras	XIII
Lista de cuadros	XV
Resumen	XVII
Abstract	XIX
1. Introducción	1
2. Antecedentes	3
2.1. Tecnología Rosa	3
2.2. Tecnología NeuroMate	3
2.3. Primera etapa	4
3. Justificación	7
4. Objetivos	9
4.1. Objetivo general	9
4.2. Objetivos específicos	9
5. Alcance	11
6. Marco teórico	13
6.1. Epilepsia	13
6.2. Motores paso a paso (Stepper)	13
6.3. Sistema de control PID	14
6.4. Cinemática de un manipulador	14
6.4.1. Representación de Denavit-Hartenberg	15
6.5. Microcontrolador	15

7. Sistema de control	17
7.1. Selección de componentes	17
7.1.1. Selección del microcontrolador	17
7.1.2. Selección de los motores	19
7.1.3. Selección de los <i>Divers</i>	20
7.2. Generación de movimiento a través de cinemática de movimiento	21
7.2.1. Matriz Denavit-Hartenberg	21
7.2.2. Cinemática directa	22
7.2.3. Obtener el jacobiano	23
7.2.4. Cinemática inversa	24
7.2.5. Generación del algoritmo en <i>Matlab</i>	24
7.3. Control PID	25
7.3.1. Obtención de la planta	25
7.3.2. Obtención de las constantes para el PID	29
7.4. Simulación	33
7.4.1. Crear un robot en <i>Webots</i> con un diseño 3D	33
7.4.2. Resultados de la simulación	37
7.5. Pruebas	40
8. Control de mandos	41
8.1. Selección de actuadores	41
8.2. Diseño del control de mandos	44
8.2.1. Primer prototipo	44
8.2.2. Segundo prototipo	45
8.2.3. Tercer prototipo	46
8.2.4. Cuarto prototipo (prototipo físico)	48
8.3. Manual de uso para el control de mandos	49
9. Diseño eléctrico	51
9.1. Selección de componentes	51
9.1.1. Fuente de voltaje 12V 240W	52
9.1.2. Conversor de voltaje (Stepdown Converter)	52
9.1.3. Conectores	52
9.1.4. <i>Nunchuky</i>	53
9.2. Primer prototipo	54
9.2.1. Esquemático	54
9.2.2. Placa de circuito impreso (PCB)	54
9.3. Segundo prototipo	55
9.3.1. Base para el robot	55
9.3.2. Esquemático	57
9.3.3. Placa de circuito impreso (PCB)	58
9.4. Tercer prototipo	60
9.4.1. Base para el robot	60
9.4.2. Esquemático	61
9.4.3. Placa de circuito impreso (PCB)	63
9.5. Cuarto prototipo (prototipo físico)	67
9.5.1. Base del robot	67
9.5.2. Placa del circuito impreso: Control de mandos	68

9.5.3. Placa del circuito impreso: Base del robot	69
10. Conclusiones	71
11. Recomendaciones	73
12. Bibliografía	75
13. Anexos	79
13.1. Manual de uso	79
13.2. Esquemáticos	79

Lista de figuras

1.	Cuarto prototipo primer etapa: brazo robótico [5]	4
2.	Cuarto prototipo primer etapa: control de mandos [6]	4
3.	Cuarto prototipo primer etapa: Arduino Mega [7]	5
4.	Segundo prototipo primer etapa: brazo robótico [6]	5
5.	Tiva C Series TM4C123GXL [18]	18
6.	Motores NEMA 23	19
7.	Motores NEMA 17	19
8.	<i>Drivers</i> que se utilizaron.	20
9.	Sistema de referencias	21
10.	Simulación del sistema en <i>Simulink</i>	25
11.	Respuesta al sistema simulado en <i>Simulink</i> ($\omega = 1KHz$)	26
12.	Planta obtenida por medio de <i>Simulink</i> y <i>PID Tuner</i>	26
13.	Parámetros de la planta obtenida por medio de <i>Simulink</i> y <i>PID Tuner</i>	27
14.	Gráfico de dispersión obtenido de los puntos generados	28
15.	Obtención de constantes PID en <i>PID Tuner</i> para la planta detectada	29
16.	Parámetros PID en <i>PID Tuner</i> para la planta detectada	29
17.	Obtención de constantes PID en <i>PID Tuner</i> para la planta de velocidad constante	30
18.	Parámetros PID en <i>PID Tuner</i> para la planta de velocidad constante	30
19.	Parámetros PID en <i>PID Tuner</i> para la planta de velocidad en rampa	30
20.	Obtención de constantes PID en <i>PID Tuner</i> para la planta de velocidad en rampa	31
21.	Verificación del PID de la planta detectada en respuesta al escalón	31
22.	Verificación del PID de la planta con velocidad constante en respuesta al escalón	32
23.	Verificación del PID de la planta con velocidad en rampa en respuesta al escalón	32
24.	Archivos aceptados por <i>Webots</i> para importar diseños 3D	33
25.	Importar un archivo CAD	34
26.	Exportar un archivo CAD	34
27.	Crear un robot en <i>Webots</i>	35
28.	Tipo de junta	35
29.	Tipo de dispositivo	36

30.	Aspecto de la junta	36
31.	Simulación 1, posición inicial	37
32.	Simulación 1, posición final	37
33.	Simulación 2, posición inicial	38
34.	Simulación 2, captura en movimiento 1	38
35.	Simulación 2, captura en movimiento 2	39
36.	Simulación 2, captura en movimiento 3	39
37.	Simulación 2, posición final	40
38.	Motor Stepper durante pruebas realizadas	40
39.	Control Nunchuk del Nintendo Wii	42
40.	Pantalla LCD TFT 3.2"	42
41.	Interruptor SPST [31]	42
42.	Selector rotativo [32]	43
43.	Botón de push [33]	43
44.	Control de mandos prototipo 1 vista superior en ángulo	44
45.	Control de mandos prototipo 1 vista superior	44
46.	Control de mandos prototipo 2 primera etapa vista isométrica	45
47.	Control de mandos prototipo 2 primera etapa vista superior	45
48.	Control de mandos prototipo 2 vista isométrica	46
49.	Control de mandos prototipo 2 vista superior	46
50.	Control de mandos prototipo 3 vista isométrica	47
51.	Control de mandos prototipo 3 vista superior	47
52.	Control de mandos prototipo 3 vista del PCB	48
53.	Control de mandos prototipo 4 vista isométrica	48
54.	Control de mandos prototipo 4 vista superior	49
55.	Fuente de voltaje	51
56.	Vista interna fuente de voltaje [36]	52
57.	Convertor de potencia [37]	52
58.	Cable de poder IEC 320 C14	53
59.	Conectores interconexiones [40]	53
60.	Conectores internos [41]	53
61.	<i>Nunchuky</i>	53
62.	Esquemático en EasyEDA	54
63.	PCB en EasyEDA	55
64.	Vista isométrica base del brazo robótico	55
65.	Vista frontal base del brazo robótico	56
66.	Brazo robótico con base	56
67.	Esquemático principal	57
68.	Esquemático control de mandos	58
69.	Esquemático base del brazo	58
70.	PCB control de mandos	59
71.	PCB base del brazo	59
72.	Base del robot vista isométrica	60
73.	Base del robot vista frontal	61
74.	Base del robot vista superior	61
75.	Base del robot vista del PCB	61

76.	Esquemático completo	62
77.	Esquemático control de mandos	62
78.	Esquemático base del robot	63
79.	Cálculos del grosor de vías requeridos	64
80.	PCB Control de mandos 2D	64
81.	PCB Control de mandos vista superior 3D	64
82.	PCB Control de mandos vista inferior 3D	65
83.	PCB Control de mandos vista isométrica 3D	65
84.	PCB Base del robot 2D	65
85.	PCB Base del robot vista superior 3D	66
86.	PCB Base del robot vista inferior 3D	66
87.	PCB Base del robot vista isométrica 3D	66
88.	Base del robot vista isométrica	67
89.	Base del robot vista superior	67
90.	Conjunto base del robot y control de mandos	68
91.	Placa del circuito impreso del control de mandos	68
92.	Placa del circuito impreso de la base del robot	69
93.	Placa del circuito impreso de la base del robot con fuente de voltaje	69

Lista de cuadros

1.	Matriz Denavit-Hartenberg Genérica	15
2.	Datos microcontroladores	18
3.	Ponderación de microcontroladores	18
4.	Comparación de <i>Drivers</i>	20
5.	Comparación de motores y <i>Drivers</i>	20
6.	Matriz Denavit-Hartenberg	22
7.	Generación de tabla para la función de rampa	28

Este trabajo consiste en el diseño del sistema de control de un brazo robótico asistencial para cirugías. Este fue dividido en tres partes: el sistema de control, el control de mandos y el diseño eléctrico. Sin embargo, por la crisis de salud mundial no fue posible fabricar los prototipos físicos del control de mandos y el diseño eléctrico.

Para el diseño del sistema de control, se realizó una selección del microcontrolador, los motores y los *drivers* a utilizar. Como resultado se obtuvo que se debe utilizar la Tiva C Series TM4C123GXL, dos motores NEMA 23, cuatro motores NEMA 17, un *Driver TB67S128FTG*, un *Driver TB67S249FTG Full Breakout* y cuatro *Drivers DRV8825*. Se implementó la cinemática inversa para poder generar el movimiento de las primeras tres juntas, obteniendo así la posición del efector final (de las primeras tres juntas). Además, se utilizó dos controladores PID (uno para la velocidad y uno para la posición) para lograr un movimiento suave y preciso en cada motor. Los cálculos y algoritmos ejecutados fueron escritos en *Matlab* y luego se realizó una simulación en *Webots*.

Como segundo paso, se diseñó el control de mandos. En esta parte del trabajo se seleccionaron los actuadores del control de mandos, tomando en consideración una interfaz simple para el usuario. Debido a esto, se optó por utilizar un control de *Wii Nunchuk* para implementar el *joystick* y los botones para el movimiento del brazo robótico asistencial. Además, se realizaron varios diseños en programas asistidos por computadora (CAD, por sus siglas en inglés) con el objetivo de fabricar el control de mandos en iteraciones futuras. Asimismo, se estableció un manual de uso para el mando de control.

Por último, se realizó el diseño eléctrico para poder alimentar y controlar los motores tomando en cuenta los actuadores y componentes previamente establecidos. Se seleccionaron los componentes eléctricos necesarios para el diseño. Se utilizará una fuente de voltaje de 12V y 240W, un convertidor *stepdown* y tres distintos tipos de cables para la conexión eléctrica. También se realizaron diseños de circuitos, para luego realizar los diseños de los circuitos impresos correspondientes. Se diseñaron dos circuitos impresos, uno para el control de mandos y uno para la base del robot, la cual también fue diseñada en esta sección del trabajo.

This thesis consists of the design of the control system of an assisting robotic arm for surgeries. The work was divided into 3 parts: the control system, the command control and the electrical design. However, due to the global health crisis, it was not possible to manufacture the physical prototypes of the command control and the electrical design.

For the design of the control system, a selection of the microcontroller, the motors and the drivers to be used was made. As a result, it was obtained that the Tiva C Series TM4C123GXL, two NEMA 23 motors, four NEMA 17 motors, a TB67S128FTG Driver, a TB67S249FTG Full Breakout Driver and four DRV8825 Drivers must be implemented. Inverse kinematics was implemented to be able to generate the movement of the first three joints, thus obtaining the position of the end effector (of the first three joints). Additionally, two PID controllers (one for speed and one for position) were designed to achieve a smooth and precise movement on each motor. The calculations and algorithms performed were written in Matlab and then a simulation was performed in Webots.

As a second step, the command control was designed. In this part of the work, the command control actuators were selected, taking into consideration a simple interface for the user. It was decided to use a Wii Nunchuk control to implement the joystick and buttons for the movement of the assistive robotic arm. In addition, several designs were made in computer-aided programs (CAD) with the aim of manufacturing the command control in future iterations. Likewise, a user manual for the command control was established.

Finally, the electrical design was carried out to power and control the motors taking into consideration the previously established actuators and components. The electrical components necessary for the design were selected: a 12V and 240W voltage source, a stepdown converter and three different types of cables that will be used for the electrical connection. Circuit designs were also designed, to later fabricate the corresponding printed circuit designs. Two printed circuits were designed, one for the command control and one for the robot base, which was also designed in this section of the work.

En Guatemala existen varias personas que padecen de accidentes cerebro-vasculares, los cuales pueden llegar a ocasionar epilepsia, una enfermedad que genera trastornos que interrumpen la actividad de las células nerviosas en el cerebro ocasionando convulsiones en las personas. Esto puede llegar a provocar pérdida de memoria en los pacientes afectados. Tomando esto en consideración y deseando mejorar el diagnóstico y tratamiento de la epilepsia, HUMANA, una organización guatemalteca dedicada al tratamiento de la epilepsia, creó una alianza con la Universidad del Valle de Guatemala para poder fabricar un brazo robótico asistencial para cirugías.

En el presente trabajo se explora la segunda fase del proyecto, enfocándose en la parte del movimiento del robot. En otras palabras, se realizó un sistema de control para obtener un movimiento suave y preciso para el brazo robótico asistencial para cirugías. Se dividió el trabajo en tres aspectos: el sistema de control, el control de mandos y el sistema eléctrico.

Como primer paso se define el sistema de control tomando en cuenta aspectos como la selección y justificación de componentes como los microcontroladores, motores y drivers. Asimismo, se genera la cinemática inversa para el movimiento de las primeras tres juntas y un controlador PID para el movimiento y la velocidad de todos los motores. Además, se realiza una simulación para verificar los resultados obtenidos.

Otro aspecto desarrollado en este documento es el control de mandos físico. Se propone la selección de los actuadores, un modelo 3D implementando los actuadores previamente definidos y un manual de uso para el control de mandos.

Por último, se genera un diseño del sistema eléctrico y electrónico que consiste en una selección de los componentes eléctricos y electrónicos a utilizar. También se diseña los circuitos y el diseño de los circuitos impresos.

2.1. Tecnología Rosa

ROSA (Asistencia Robótica Estereotáctica, por sus siglas en inglés) es una tecnología de robótica médica, la cual ayuda a minimizar la invasión en cirugías del sistema nervioso. También ayuda a médicos profesionales a realizar cirugías minimizando el tiempo requerido para estas y dando mayor precisión, disminuyendo riesgos de complicaciones por anestesia u otros contratiempos. Esta tecnología fue diseñada por Zimmer Biomet Robotics y es capaz de disminuir el procedimiento de la cirugía por 3 horas y 42 minutos, según un estudio realizado por SEEG Procedures. La tecnología ROSA utiliza un brazo robótico y una consola para poder controlar y posicionar los instrumentos requeridos por el cirujano para realizar la cirugía [1] .

La tecnología se utiliza para distintos procedimientos como electroencefalografía estereo (SEEG), estimulación cerebral profunda (DBS), biopsia estereotáctica, endoscopia ventricular y endoscopia transnasal, entre otros. Actualmente, se utiliza esta tecnología en varios hospitales en distintos países, siendo uno de ellos, el hospital Stanford Children's Health | Lucile Packard Children's Hospital Stanford, el cual realiza estos procedimientos en niños [2].

2.2. Tecnología NeuroMate

El sistema robotizado NeuroMate sirve para neurocirugía estereotáctica, dando precisión y rapidez en los procedimientos. Esto ayuda a minimizar el trauma de los pacientes, haciendo que la cirugía sea menos invasiva y se puedan insertar los electrodos necesarios con mayor facilidad. El robot cuenta con herramientas para la inserción de los instrumentos médicos y herramientas láser para mayor precisión, al marcar los puntos donde se ingresará en la cabeza del paciente, entre otras [3], [4].

2.3. Primera etapa

En la primera etapa del proyecto de un brazo robótico para cirugías de la Universidad del Valle de Guatemala, se realizaron diseños mecánicos para obtener un robot adecuado a los requisitos para las cirugías. Para esto se realizaron 4 prototipos. Dos de ellos fueron fabricados en fibra de densidad media (MDF) y los siguientes dos en PLA por medio de impresión 3D. El sistema de control era fallido, ya que el brazo robótico tenía demasiada incertidumbre al moverse, y siendo para una aplicación médica para cirugías cerebrales, es imperativo tener un control suave y seguro para el paciente. Además, es necesario diseñar un sistema mecánico más rígido, ya que el de la primera etapa es propenso a movimientos por medio de accidentes [5], [6], [7].



Figura 1: Cuarto prototipo primer etapa: brazo robótico [5]

El mando de control fue diseñado para poder controlar el brazo por medio de varios botones. Se enfocan en dos botones para cambiar en la etapa del movimiento, cuatro botones para el movimiento como tal y otros para poder aumentar o disminuir la velocidad y dar comandos de bloqueo [6].

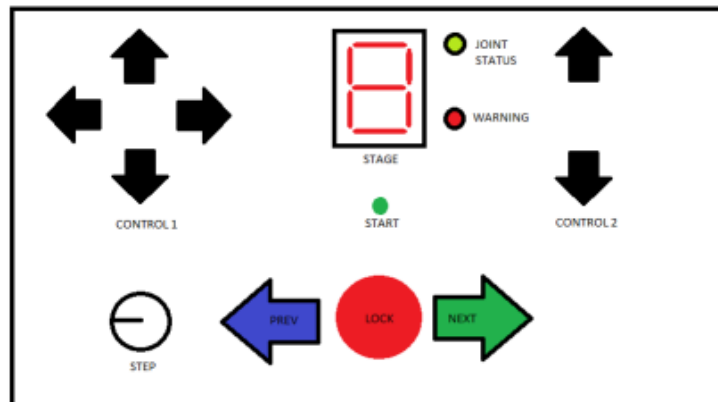


Figura 2: Cuarto prototipo primer etapa: control de mandos [6]

En la primera fase, se utilizó un Arduino Mega, debido a la cantidad de pines requeridos para poder completar el sistema de control de los servo motores [7].

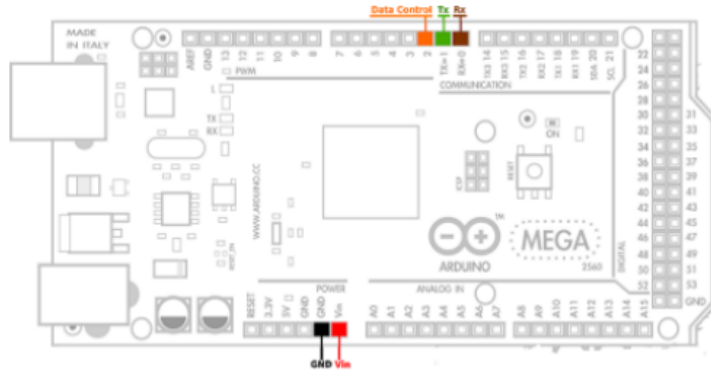


Figura 3: Cuarto prototipo primer etapa: Arduino Mega [7]

Para el posicionamiento se utilizó la cinemática inversa y directa. Después de obtener la cinemática directa, se obtuvo la matriz de Denavit-Hartenberg. Utilizaron Matlab para poder obtener correctamente el posicionamiento deseado. Sin embargo, a la hora de aplicarlo y unirlo al movimiento de los servo motores se tuvo una falla en el movimiento suave. Esto ocasionó que el movimiento, en lugar de ser paulatino, fue brusco y pausado [6].

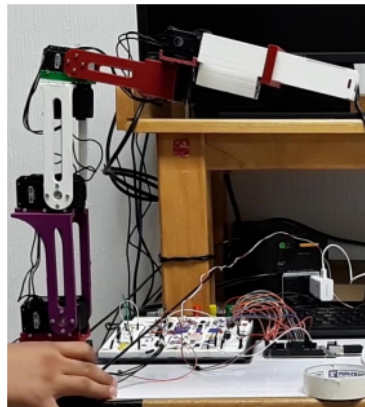


Figura 4: Segundo prototipo primer etapa: brazo robótico [6]

La epilepsia es una enfermedad del sistema nervioso que se presenta al tener actividad eléctrica anormal en la corteza cerebral. Esto ocasiona ataques repentinos caracterizados por convulsiones violentas de distinta duración y pérdida del conocimiento. Esta enfermedad afecta a 50 millones de personas alrededor del mundo, haciendo su tratamiento de suma importancia [8], [9], [10].

En Guatemala, la organización que ayuda a personas con epilepsia es la Asociación HUMANA, la cual fue fundada en el 2006, como el Centro de Epilepsia y Neurocirugía Funcional HUMANA. Esta organización trata personas con epilepsia, problemas neurológicos, de columna y espina bífida. Unas de estas cirugías constan en la inserción de electrodos en ciertas cavidades del cerebro, haciéndolo un procedimiento altamente delicado y con requerimiento de una muy alta precisión. Actualmente, para realizar estas cirugías utiliza un brazo mecánico para posicionar los electrodos, el cual tiene que ser movido manualmente ajustando las partes de este para llegar a la posición final deseada. Esto ocasiona que sea un trabajo muy cansado y tardado para los médicos profesionales. A lo largo de una de estas cirugías, se puede necesitar insertar hasta 8 electrodos, haciendo esto una cirugía de varias horas, ya que el reposicionamiento del brazo mecánico tarda aproximadamente 30 minutos, haciendo que el paciente este bajo anestesia por bastante tiempo [11] .

Para poder minimizar el riesgo en sus pacientes, HUMANA le solicitó a la Universidad del Valle de Guatemala un brazo robótico que pueda reposicionarse por medio de retroalimentación de la resonancia hecha al paciente. Sin embargo, esta solución no es una solución factible, debido a que la retroalimentación de la resonancia es demasiado tardada, por lo que se optó por una solución distinta. Se crearía un brazo robótico, el cuál sería controlado por el médico para poder posicionar los electrodos correctamente. Esto haría que la cirugía disminuyera su tiempo y que el paciente corra menos riesgos.

4.1. Objetivo general

Diseñar y simular un sistema de control para un brazo robótico asistencial de 6 grados de libertad para cirugías cerebrales.

4.2. Objetivos específicos

- Utilizar cinemática inversa para generar el movimiento para el efector final.
- Realizar un control de motores para obtener un movimiento suave y preciso.
- Realizar una simulación de todos los sistemas de control por medio de computadora.
- Utilizar un software de diseño asistido por computadora para modelar el control de mandos del robot asistencial.
- Fabricar por medio de impresión 3D y corte láser el control de mandos.
- Diseñar el sistema eléctrico y electrónico del brazo robótico asistencial.
- Diseñar y fabricar los circuitos impresos del sistema eléctrico y electrónico.
- Realizar los diagramas y esquemáticos del sistema eléctrico y electrónico.

El proyecto consiste en diseñar un sistema de control completo. Esto implica: diseñar el sistema de control y verificarlo con simulaciones realizadas en aplicaciones como *Webots*. Adicionalmente se diseñará un control de mandos físico para el sistema y el sistema eléctrico correspondiente.

En términos del sistema de control se pretende implementar una selección de los componentes y el diseño completo de la cinemática inversa para las primeras tres juntas utilizando la matriz de Denavit-Hartenberg, la cinemática directa, el jacobiano y la implementación de *Matlab*. Asimismo, se procura diseñar un controlador PID con apoyo de *Matlab* y una simulación en *Webots* importando el modelo físico diseñado por Bryan Raúl Santos Avila.

El diseño del control de mandos incluye el 3D en *Autodesk Inventor*, la selección de los actuadores, un manual del control de mandos y un prototipo físico.

Como último punto el diseño eléctrico abarca los diseños de los circuitos, los circuitos impresos (*PCB*) y el 3D de la base del robot, donde estará parte del sistema eléctrico. Asimismo, también se considera la fabricación de los mismos.

6.1. Epilepsia

La epilepsia es una de las enfermedades neurológicas más comunes en el mundo. Aproximadamente 2 millones de personas sufren de esta enfermedad en los Estados Unidos y alrededor de 80 millones tienen epilepsia en todo el mundo. Esta se define como un trastorno cerebral caracterizado por ataques de convulsiones y pérdida de la conciencia [8], [9], [10].

6.2. Motores paso a paso (Stepper)

El motor paso a paso es un dispositivo electromecánico que convierte una secuencia de pulsos eléctricos a un desplazamiento angular discreto. Esto hace posible que pueda girar una cierta cantidad de grados por medio de pasos. Estos pasos pueden diferir dependiendo de la cantidad de entradas. El funcionamiento se basa en un estator construido por varios embobinados y un rotor que gira libremente en el estator. Los embobinados se alimentan en distintas secuencias para generar lo que conocemos como el paso del motor o el paso angular, lo que hace que el motor se desplace angularmente. Existen tres tipos de motores paso a paso. Estos son motores de reluctancia variable, de imán permanente o híbrido [12].

El motor paso a paso de reluctancia variable utiliza un campo magnético variable, lo que ocasiona que pueda moverse sin limitaciones, para poder conseguir desplazamientos más reducidos, sin tener que aumentar considerablemente el número de bobinados. Sin embargo, es el menos común. Se utiliza para aplicaciones que no requieren un alto grado de par de fuerza [12].

Los motores paso a paso de imán permanente se dividen en dos tipos, unipolares y bipolares. Los motores unipolares constan de 5 o 6 cables, dependiendo de su configuración interna. 4 de los cables son usados para indicar la secuencia y la duración de los pasos,

mientras que los cables restantes se utilizan para la alimentación. Se caracteriza por ser más sencillo de controlar. Los motores bipolares constan de dos bobinas, por lo que se utilizan cables para energizar las bobinas en una secuencia y obtener el movimiento por medio de pasos. Dependiendo de la configuración de la bobina, la corriente puede fluir en ambas direcciones haciendo necesario un control bidireccional o bipolar. A pesar de ser más difícil de controlar, el motor bipolar tiene mayor torque y anclaje comparado a un motor unipolar [12],[13].

Para la secuencia de los pasos, se debe controlar la cantidad de bobinas que se energizan y la secuencia en que se hace. Dependiendo de esto se pueden realizar pasos completos, energizando una bobina y luego la otra, mientras se apaga la bobina previamente energizada o se pueden realizar pasos medios, en los cuales se enciende la primera bobina y luego se enciende la bobina consiguiente sin apagar la bobina previamente energizada, ocasionando un movimiento de media totalidad [12].

El motor paso a paso híbrido es una combinación del funcionamiento del motor con reluctancia variable y el motor con imán permanente [12] .

6.3. Sistema de control PID

Un sistema de control PID, es un sistema que permite controlar la salida de una planta, minimizando los errores. En el caso de un PID, se utilizan tres controladores importantes. Un controlador proporcional (P), un controlador integral (I) y un controlador derivativo (D). El parámetro proporcional depende del error actual, el parámetro integral depende de los errores pasados y el derivativo es una predicción de los errores futuros [14],[15].

6.4. Cinemática de un manipulador

La cinemática de un manipulador se utiliza para describir el movimiento de este, para poder controlarlo y que realice las operaciones deseadas. Para esto se enfoca los elementos estructurales rígidos, mejor conocidos como eslabones, las juntas (de rotación o traslación) y sus grados de libertad. Dependiendo de la aplicación que se quiera dar y de los datos iniciales, es el tipo de cinemática que se utiliza. Esta puede ser cinemática directa o cinemática inversa [16].

Se utiliza la cinemática directa para obtener la posición y orientación del efector final. Esto se hace conociendo las geometrías del manipulador dado, como los ángulos para articulaciones rotacionales y desplazamientos para articulaciones traslacionales. La cinemática inversa se utiliza para obtener el conjunto de ángulos y desplazamientos que darán como resultado la posición del efector final. Para esta se debe conocer la posición inicial y final del efector final. En otras palabras, en la cinemática directa, se obtiene la información iniciando desde la base y se avanza hasta llegar al efector final, mientras que en la cinemática inversa se inicia en el efector final y se avanza hacia la base del manipulador [16].

6.4.1. Representación de Denavit-Hartenberg

La representación de Denavit-Hartenberg, fue propuesta por Jacques Denavit y Richard Hartenberg en 1955 para estandarizar los marcos de coordenadas para estructuras espaciales de la cinemática. Este es un sistema de un modelo matricial para establecer la forma sistemática de un cuerpo rígido. Esta matriz se obtiene al realizar una serie de procedimientos a las distintas poses de las juntas. Al obtener la matriz de Denavit-Hartenberg (Matriz DH), se puede utilizar para los cálculos de la cinemática inversa. La matriz DH se basa en una rotación en el eje Z (θ), una traslación en dirección del eje Z (d), una traslación en dirección del eje X (a) y una rotación respecto al eje X (α). La idea es llegar de la pose actual a la pose siguiente realizando esta serie de movimientos [17].

	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	α_1
2	θ_2	d_2	a_2	α_2
...
n	θ_n	d_n	a_n	α_n

Tabla 1: Matriz Denavit-Hartenberg Genérica

6.5. Microcontrolador

Un microcontrolador es un conjunto de sistemas integrados, los cuales sirven para una función en específico en un sistema embebido. El microcontrolador contiene una unidad central de procesamiento (CPU), unidades de memoria RAM y ROM, varios puertos de entradas y salidas, ya sean digitales o analógicos y en conjunto conforman una computadora que soluciona problemas específicos para distintas aplicaciones.

En este capítulo se explica lo que es el diseño del control, la selección de los controladores de motores que fueron seleccionados y los distintos métodos de control implementados. Esto nos permite generar un movimiento suave y preciso en el brazo robótico asistencial para cirugías.

7.1. Selección de componentes

7.1.1. Selección del microcontrolador

Para poder controlar el brazo robótico se utilizó un microcontrolador. Sin embargo, este debe cumplir ciertos requisitos para poder lograr el desempeño deseado. Entre estos se encuentran la cantidad de entradas (debido a que se estarán controlando 6 motores), la cantidad de pines que proveen control de modulación del ancho del pulso (PWM, por sus siglas en inglés), una frecuencia operacional alta, una memoria suficiente para la programación de matrices de control. Tomando estas características en cuenta se compararon varios microcontroladores por medio de una tabla de ponderación. un punteo a cada aspecto y logrando obtener el mejor microcontrolador para la aplicación deseada.

Como se puede observar en las tablas 2 y 3 el microcontrolador más adecuado es la Tiva C Series TM4C123GXL (5).

Microcontrolador	Ponderación	PIC16F887	Arduino Nano	Arduino UNO	Arduino Mega	Tiva C	Raspberry Pi 3
Cantidad de entradas y salidas digitales I/O pins (12)	5	40	14	14	54	35	40
Cantidad de PWM (6)	5	2 pero se pueden generar en GPIO	6	6	15	6-16	2
Frecuencia operacional	10	20 MHz E 8 MHz Int	16 MHz	16 MHz	16 MHz	80 MHz	1.2 GHz
RAM	10	368 B	2 KB	2 KB	8 KB	32 KB	1 GB
Program memory (Flash)	8	14 KB	32 KB	32 Kb	256 KB	256 KB	microSD
Programación para matrices	1	sí	sí	sí	sí	sí	sí

Tabla 2: Datos microcontroladores

Microcontrolador	PIC16F887	Arduino Nano	Arduino UNO	Arduino Mega	Tiva C	Raspberry Pi 3
Cantidad de entradas y salidas digitales I/O pins (12)	10	5	5	10	10	10
Cantidad de PWM (6)	8	6	6	10	10	6
Frecuencia operacional	6	8	8	8	9	10
RAM	3	5	5	8	8	10
Program memory (Flash)	3	5	5	10	10	8
Programación para matrices	10	10	10	10	10	10
Total	214	235	235	350	360	354

Tabla 3: Ponderación de microcontroladores

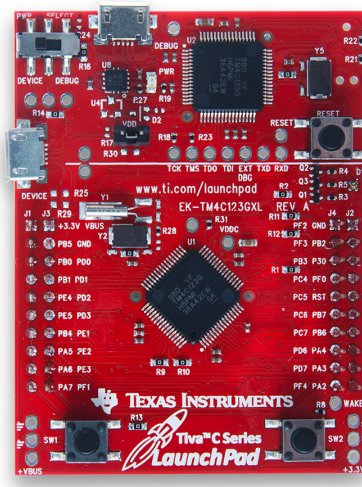
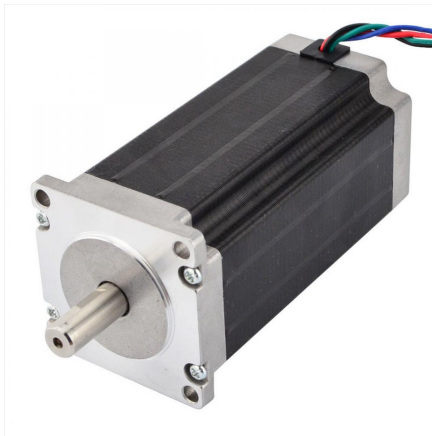


Figura 5: Tiva C Series TM4C123GXL [18]

7.1.2. Selección de los motores

En la fase anterior se implementaron servo motores y estos no proveyeron una buena estabilidad en el movimiento y no generaban el torque necesario requerido para el brazo robótico asistencial para cirugías. Tomando esto en consideración se optó por utilizar motores *stepper* (motor de pasos) para lograr un mayor torque, obteniendo mayor seguridad para el paciente. El proceso de selección fue basado en el torque necesario para soportar las distintas juntas del brazo robótico. Estos cálculos se pueden encontrar en el trabajo de graduación de mi compañero Bryan Raúl Santos Avila [19]. Los resultados de los mismos fueron que los motores adecuados eran: dos NEMA 23. Uno con un torque de parada de 3 N.m. y el otro con un torque de 1.2 N.m. El motor colocado en la base tiene un peso de 1.60 kg para agregar estabilidad al brazo. También se seleccionaron cuatro NEMA 17, dos con cajas reductoras para aumentar el torque, el cual es de 4.71 N.m. que son empleados en las juntas 3 y 4 y dos NEMA 17 sin caja reductoras para las últimas dos juntas con un torque de 0.34 N.m.



(a) NEMA 23 Junta 1 [20]



(b) NEMA 23 Junta 2 [21]

Figura 6: Motores NEMA 23



(a) NEMA 17 Juntas 3 y 4 [22]



(b) NEMA 17 Juntas 5 y 6 [23]

Figura 7: Motores NEMA 17

7.1.3. Selección de los *Drivers*

Al tener los motores que se iban a utilizar fue posible seleccionar los *drivers* que eran funcionales para poder controlar el brazo robótico asistencial. Para esto se buscó la información de los distintos *drivers* tomando en cuenta tanto la corriente y el voltaje requerido por los motores, como el tipo de control que se utiliza (por pasos y dirección (*Step/Dir*), programación SPI o programación UART) y el costo de los mismos. Se realizó una tabla comparativa (Tabla 4) de *drivers* comúnmente utilizados en aplicaciones de robótica.

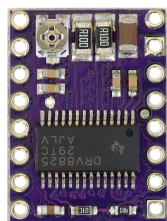
	Driver A4988	Driver DRV8825	Driver TB67S249FTG Full Breakout	Driver TB67S128FTG	TMC2226-SA	TMC2209-LA	TMC2660C-PA
Voltaje mínimo	8.2 V	8.2 V	10 V	6.5 V	4.75 V	4.75 V	9 V
Voltaje máximo	35 V	45 V	47 V	44 V	29 V	29 V	30 V
Corriente continua por fase	1 A	1.5 A	1.7 A	2.1 A	2 A	2 A	2.8 A
Corriente máxima por fase	2 A	2.2 A	4.5 A	5.0 A	-	-	-
Step/Dir	Si	Si	Si	Si	Si	Si	Si
MicroStep	1/16	1/32	1/32	1/128	1/256	1/256	1/256
Programación SPI	No	No	No	No	No	No	Si
Programación UART	No	No	No	No	Si	Si	No
Precio	\$5.95	\$8.95	\$11.95	\$13.95	~\$15	~\$15	~\$15

Tabla 4: Comparación de *Drivers*

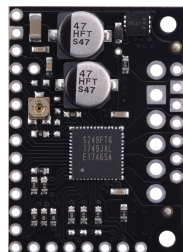
Basándose en la corriente requerida y en el voltaje de alimentación se seleccionó el *driver TB67S128FTG* para el motor NEMA 23 de la primera junta, el *driver TB67S249FTG Full Breakout* para el motor NEMA 23 de la segunda junta y para los motores NEMA 17 se seleccionó el *driver DRV8825*. Otro factor que afectó en la selección es que el control funciona correctamente sin programación SPI o UART. La comparación de los parámetros se puede observar en la Tabla 5. En esta se puede observar las parejas de motor y *driver*, donde se resume la relación de voltaje y corriente de estos.

Tipo	Motor NEMA 23 Junta 1	Driver TB67S128FTG	Motor NEMA 23 Junta 2	Driver TB67S249FTG Full Breakout	Motores NEMA 17 Juntas 3 y 4	Motores NEMA 17 Juntas 5 y 6	Driver DRV8825
Voltaje de operación	3.78 V	6.5 V - 44 V	-	10 V - 47 V	12 V	3.4 V	8.2 V - 45 V
Corriente requerida / entregada	4.2 A	5.0 A	2.8 A	4.5 A	1.7 A	1.7 A	2.2 A

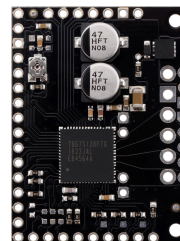
Tabla 5: Comparación de motores y *Drivers*



(a) *DRV8825 stepper motor driver carrier*. [24]



(b) *TB67S249FTG Stepper Motor Driver Carrier* (Vista Superior). [25]



(c) *TB67S128FTG Stepper Motor Driver Carrier* (Vista Superior). [26]

Figura 8: *Drivers* que se utilizaron.

7.2. Generación de movimiento a través de cinemática de movimiento

La cinemática inversa se implementó para poder mover las primeras tres juntas del brazo robótico asistencial. En otras palabras, esta se utilizó para mover los primeros tres motores (empezando desde la base) y poder especificar la posición del efector final de estas tres juntas, en este caso la junta número cuatro o el cuarto motor. Para lograr esto se generó un algoritmo que implementa la matriz de Denavit-Hartenberg, cinemática directa y el jacobiano de tal modo que se pueda obtener la configuración de los motores para poder llegar a esta posición deseada.

7.2.1. Matriz Denavit-Hartenberg

Para poder realizar movimientos en nuestro manipulador, es necesario conocer el sistema del mismo. Esto se puede obtener mediante la matriz de Denavit-Hartenberg (Tabla 1) la cual proporciona los datos necesarios de una forma clara y ordenada para poder trasladar los movimientos de junta a junta. Para esto se asignó los marcos de referencias correspondientes, como se puede observar en la Figura 9.

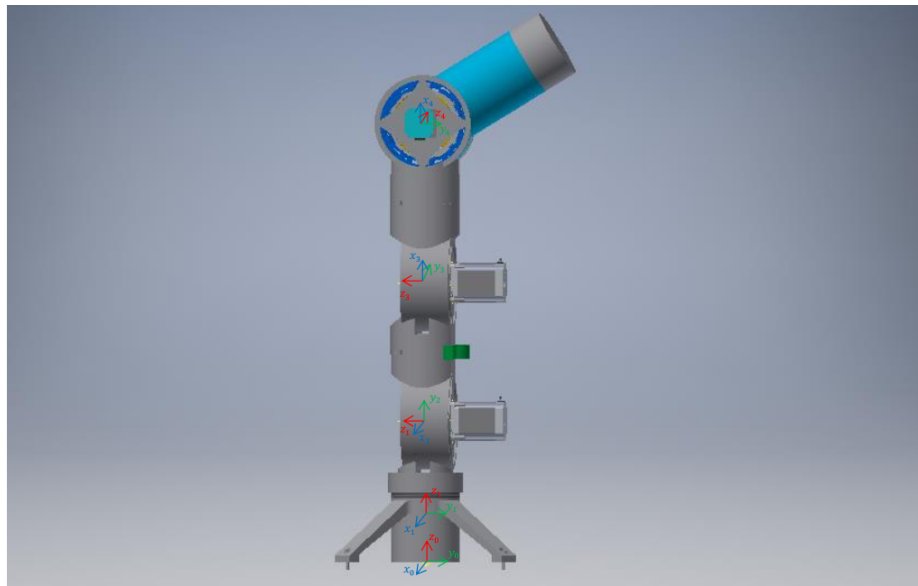


Figura 9: Sistema de referencias

Los marcos de referencia se deben colocar dependiendo del tipo de junta que tenemos. Esta puede ser prismática o revoluta (como es en este caso). Primero se debe colocar el eje Z en la misma dirección al movimiento de la junta. Luego se debe situar el eje X paralelo a la normal común ($X_n = Z_n \times Z_{n-1}$). Por último, se sitúa el eje Y para que este en concordancia con el eje Z y el eje X.

Después de colocar los ejes de referencia se puede obtener la matriz DH, la cual nos describe el movimiento que se debe realizar para llegar de la pose de una junta n a la pose

de una junta m . La matriz DH utilizada en esta aplicación se encuentra a continuación (Tabla 6).

	θ	d	a	α
1	$q(1)$	0.114	0	$\frac{\pi}{2}$
2	$q(2) + \frac{\pi}{2}$	0	0.200	0
3	$q(3)$	0	0.230	$-\frac{\pi}{2}$

Tabla 6: Matriz Denavit-Hartenberg

7.2.2. Cinemática directa

En la cinemática directa se utilizan transformaciones elementales para obtener un mapeo entre el espacio de configuración y el espacio de tarea del robot. Esta se obtiene por medio de la ecuación 1 Sin embargo, bajo la convención DH se puede obtener la cinemática directa de un manipulador serial por medio de la ecuación 2, donde la matriz A_j equivale a la matriz de traslación entre sistemas coordenados de referencia. Cabe recalcar que las matrices de transformaciones son matrices homogéneas y las transformaciones son del marco inercial de la base del manipulador "B" al marco inercial del efector final "E".

$$\mathcal{K}(\mathbf{q}) \subset B\mathbf{T}_E(\mathbf{q}) = \begin{bmatrix} {}^B\mathbf{R}_E(\mathbf{q}) & {}^B\mathbf{O}_E(\mathbf{q}) \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \in SE(3) \quad (1)$$

$$\mathcal{K}(\mathbf{q}) = \mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_{N-1} \mathbf{A}_N \quad (2)$$

$$\mathbf{A}_j = {}^{j-1}\mathbf{T}_j = \text{Rot}_z(\theta_j) \text{Transl}_z(d_j) \text{Transl}_x(a_j) \text{Rot}_x(\alpha_j) \quad (3)$$

$$\text{Rot}_z(\theta_j) = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & 0 \\ \sin(\theta_j) & \cos(\theta_j) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$\text{Transl}_z(d_j) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\text{Transl}_x(a_j) = \begin{bmatrix} 1 & 0 & 0 & a_j \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$\text{Rot}_x(\alpha_j) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_j) & -\sin(\theta_j) & 0 \\ 0 & \sin(\theta_j) & \cos(\theta_j) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

7.2.3. Obtener el jacobiano

El jacobiano es una matriz de $6 \times n$, en donde n es la dimensión de la configuración. Esta matriz representa la velocidad de nuestro efector (8) por medio de una combinación lineal de las velocidades de las revolutas. En la ecuación 10 se puede observar el jacobiano de posición (\mathbf{J}_v) y orientación (\mathbf{J}_ω). Esto nos da un mapeo entre la velocidad de la configuración y la velocidad (traslacional y rotacional) del efector final. En otras palabras, el jacobiano describe la cinemática directa diferencial del manipulador serial.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \vdots \\ z \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \mathbf{J}(\mathbf{q}) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \mathbf{q}_n \end{bmatrix} \quad (8)$$

$$\begin{aligned} \dot{x} &= \frac{\partial f_x}{\partial \mathbf{q}_1} \cdot \dot{q}_1 + \frac{\partial f_x}{\partial \mathbf{q}_2} \cdot \dot{q}_2 \cdots + \frac{\partial f_x}{\partial \mathbf{q}_n} \cdot \dot{q}_n \\ \dot{y} &= \frac{\partial f_s}{\partial \mathbf{q}_1} \cdot \dot{q}_1 + \frac{\partial f_v}{\partial \mathbf{q}_2} \cdot \dot{q}_2 \cdots + \frac{\partial f_v}{\partial \mathbf{q}_n} \cdot \dot{q}_n \\ \dot{z} &= \frac{\partial f_z}{\partial \mathbf{q}_1} \cdot \dot{q}_1 + \frac{\partial f_i}{\partial \mathbf{q}_2} \cdot \dot{q}_2 \cdots + \frac{\partial f_z}{\partial \mathbf{q}_n} \cdot \dot{q}_n \\ \dot{\alpha} &= \frac{\partial f_a}{\partial \mathbf{q}_1} \cdot \dot{q}_1 + \frac{\partial f_a}{\partial \mathbf{q}_2} \cdot \dot{q}_2 \cdots + \frac{\partial f_a}{\partial \mathbf{q}_n} \cdot \dot{q}_n \\ \dot{\beta} &= \frac{\partial f_\theta}{\partial \mathbf{q}_1} \cdot \dot{q}_1 + \frac{\partial f_\theta}{\partial \mathbf{q}_2} \cdot \dot{q}_2 \cdots + \frac{\partial f_\beta}{\partial \mathbf{q}_n} \cdot \dot{q}_n \\ \dot{\gamma} &= \frac{\partial f_r}{\partial \mathbf{q}_1} \cdot \dot{q}_1 + \frac{\partial f_r}{\partial \mathbf{q}_2} \cdot \dot{q}_2 \cdots + \frac{\partial f_r}{\partial \mathbf{q}_n} \cdot \dot{q}_n \end{aligned} \quad (9)$$

$$\begin{bmatrix} \mathbf{J}_v(\mathbf{q}) \\ \mathbf{J}_\omega(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (10)$$

Inversa del jacobiano

El mapeo inverso del jacobiano no se puede garantizar que exista, ya que por lo general no es una matriz cuadrada. Por este motivo, se utiliza la pseudo-inversa de Moore-Penrose

(ecuación 12 para jacobianos "gordos" (más columnas que filas) y la ecuación 13 para jacobianos "delgados" (más filas que columnas)).

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \begin{bmatrix} {}^B \mathbf{V}_E \\ {}^B \boldsymbol{\omega}_{BE} \end{bmatrix} \quad (11)$$

$$\mathbf{J}^\dagger = \mathbf{J}^\top \left(\mathbf{J} \mathbf{J}^\top \right)^{-1} \quad (12)$$

$$\mathbf{J}^\dagger = \left(\mathbf{J}^\top \mathbf{J} \right)^{-1} \mathbf{J}^\top \quad (13)$$

7.2.4. Cinemática inversa

Se puede obtener la configuración de nuestro manipulador a través de la cinemática inversa numérica como se puede observar en la ecuación 14.

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}_v^\dagger(\mathbf{q}_k) \mathbf{e}_p[k] \quad (14)$$

$$\mathbf{J}_v^\dagger(\mathbf{q}_k) = \mathbf{J}_v^\top(\mathbf{q}_k) \left(\mathbf{J}_v(\mathbf{q}_k) \mathbf{J}_v^\top(\mathbf{q}_k) \right)^{-1} \quad (15)$$

$$\mathbf{e}_p[k] = \mathbf{o}_d - {}^B \mathbf{o}_E(\mathbf{q}_k) \quad (16)$$

7.2.5. Generación del algoritmo en *Matlab*

Para obtener la configuración deseada del efector final es necesario conocer dos factores a priori: la configuración actual del manipulador y la posición final deseada del efector final. Al tener esta información se puede generar un algoritmo que nos ayudará a calcular la configuración deseada del manipulador. De esta forma se obtiene la posición final deseada para el efector final. Primero se debe conocer la matriz de Denavit-Hartenberg de nuestro manipulador para así poder generar una matriz K que nos permite trasladarnos desde la base del manipulador hasta el efector final. El algoritmo para generar la trayectoria del efector final se enfoca en verificar el error de la posición actual con la posición deseada como se puede observar en la ecuación 16, en la que $\mathbf{e}_p[k]$ es el error de posición, \mathbf{o}_d la configuración de la posición actual y ${}^B \mathbf{o}_E(\mathbf{q}_k)$ la configuración de la posición deseada. La idea es disminuir el error hasta llegar al margen previamente establecido como error mínimo aceptable para obtener una posición suficientemente cercana que sea aceptable para nuestra implementación.

Mientras el error no cumpla con este requerimiento se debe el jacobiano actual y se genera una sumatoria para calcular la nueva posición sumando la posición actual y el producto del jacobiano obtenido y el error (ecuación 14). Luego se guardan los datos obtenidos y se realiza el mismo procedimiento hasta que se cumpla con el criterio del error mínimo aceptable. En

caso este se cumplierse, se llegó a la solución del problema para llegar a la posición deseada. En caso este no se cumpla, se vuelve a calcular la nueva posición con ayuda del jacobiano y se realizan las mismas pruebas hasta llegar a un error que sí satisfaga.

7.3. Control PID

Los motores *stepper* (paso a paso) se pueden controlar por medio de *drivers* que utilizan un control simple. Este consiste en utilizar señales PWM en el pin *Step* del driver para indicar los pasos y una señal digital en el pin *Dir* para especificar la dirección. Los ángulos por paso se definen por medio de los motores a utilizar, en este caso, un paso equivale a 1.8° y los *drivers* permiten realizar fracciones de pasos por medio de tres entradas digitales en los pines designados al *Microstepping*. El *driver* utilizado con menor fraccionamiento de pasos es el *driver DRV8825* (ver figura: 8a), el cual cuenta con $\frac{1}{32}$ de paso. Este se seleccionó para obtener una mayor resolución y se puede generalizar a todos los motores.

7.3.1. Obtención de la planta

Para poder implementar un sistema de control PID es necesario obtener la planta del sistema. Esta se puede obtener de distintas maneras, por ejemplo: por medio de pruebas físicas por el método de Cecil-Smith (que utiliza una serie de pasos para poder generar la planta a partir de mediciones obtenidas), por medio de funciones matemáticas que describan el sistema o por medio de ayuda de un programa de software como *Matlab*. En este caso se utilizaron las últimas dos opciones para verificar que sí se tuviera una planta adecuada.

Obtención de planta por medio de *Simulink* y *Matlab*

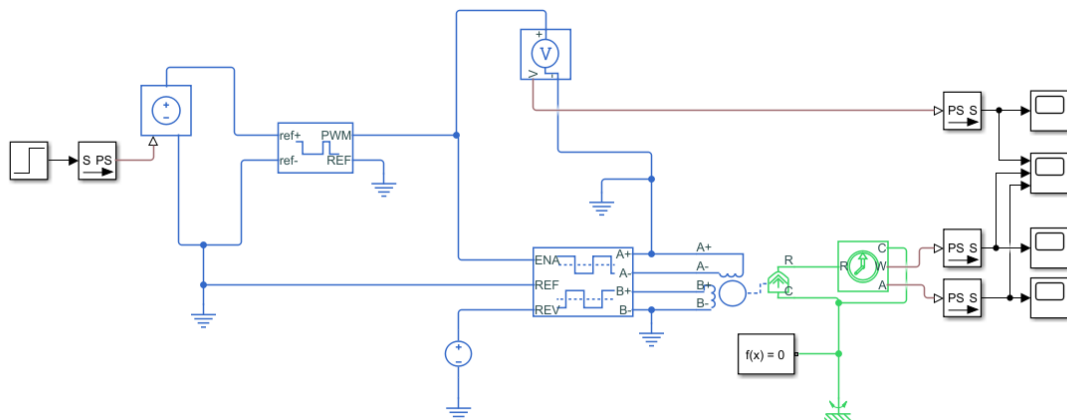


Figura 10: Simulación del sistema en *Simulink*

Primero se realizó una simulación del sistema por medio de componentes de *Simulink*. Estos fueron obtenidos de *Simulink* y *Simscape Electrical*. Se utilizó un motor *stepper* , un *driver* para motores *stepper* , un generador de PWM , medidores para el sistema y un lector de la salida (Scope). Al terminar las conexiones se obtuvo un sistema (ver Figura: 10) que simulaba el sistema físico.

De este sistema se obtuvo la salida (ver Figura: 11) y se guardó esta como un listado de variables. Este se utilizó para obtener el modelo del sistema (ver Figura: 12) para poder ser utilizado en el control PID. Se utilizó la herramienta de *PID Tuner* de *Matlab* en la cual se le ingresaron los datos necesarios para que la herramienta estimará la planta y se pudiera obtener los datos necesarios (ver Figura: 13) para generar la función de transferencia (17).

$$G_1 = \frac{K}{(T_1s + 1)(T_2s + 1)} = \frac{0.3274}{(0.9999s + 1)(0.1345 + 1)} \quad (17)$$

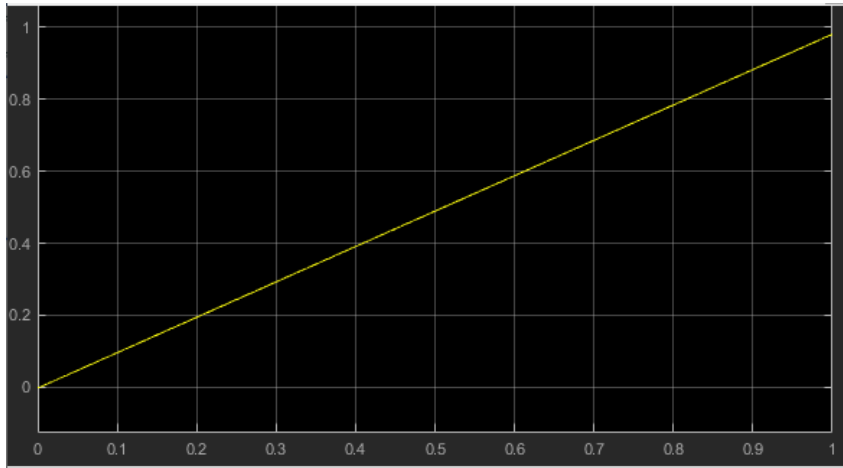


Figura 11: Respuesta al sistema simulado en *Simulink* ($\omega = 1KHz$)

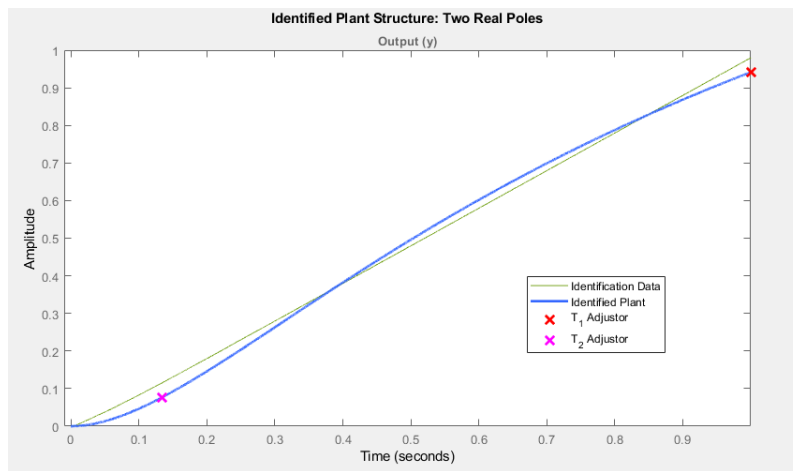


Figura 12: Planta obtenida por medio de *Simulink* y *PID Tuner*

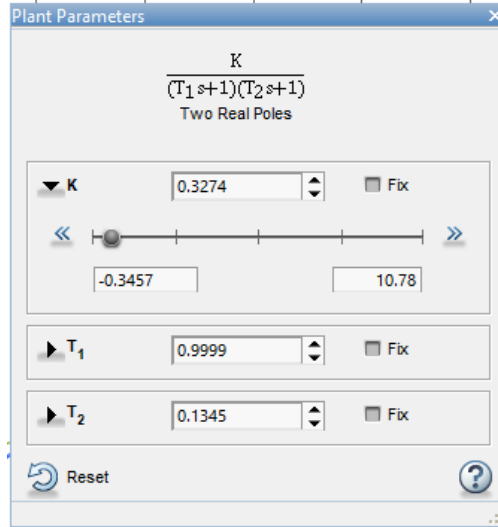


Figura 13: Parámetros de la planta obtenida por medio de *Simulink* y *PID Tuner*

Obtención de planta por medio de funciones matemáticas

Para obtener una planta general se utilizó una función que caracteriza el comportamiento de la posición a una velocidad constante y una función que caracteriza el comportamiento de la posición en una velocidad de rampa para obtener una velocidad con función trapezoidal, que da como resultado un movimiento suave.

La función general para la posición a velocidad constante (18) se obtuvo al comprender el comportamiento del sistema y los parámetros que se le entregan al *driver* para generar un movimiento, siendo estos los ángulos por paso (θ), la división de los pasos (ϕ), la cantidad de pulsos o frecuencia del PWM (ω) y el tiempo (t).

$$\Theta(t) = \frac{\theta}{\phi} \omega t \quad (18)$$

Para la función general de la posición a una velocidad trapezoidal se hizo la suposición que esta se puede obtener por medio de un cambio en la división de pasos (ϕ), aumentando la cantidad de grados por paso, obteniendo así una velocidad de rampa. El rango, el cual es establecido por los *drivers*, es de $\frac{1}{32}$ a 1. Al comprender esto se generó una Tabla (7) con el muestreo de las posiciones alterando la división de pasos hasta llegar a la menor división (1/1). Esto se realizó con la ecuación 18 con una frecuencia de 1 Hz. Al terminar, se graficaron los resultados en Excel y se obtuvo una ecuación a partir de la línea de tendencia exponencial (ver Figura 14).

ϕ	t	Θ
0	0	0
$\frac{1}{32}$	0.1	0.005625
$\frac{1}{16}$	0.2	0.028125
$\frac{1}{8}$	0.3	0.095625
$\frac{1}{4}$	0.4	0.275625
$\frac{1}{2}$	0.5	0.725625
1	0.6	1.805625

Tabla 7: Generación de tabla para la función de rampa

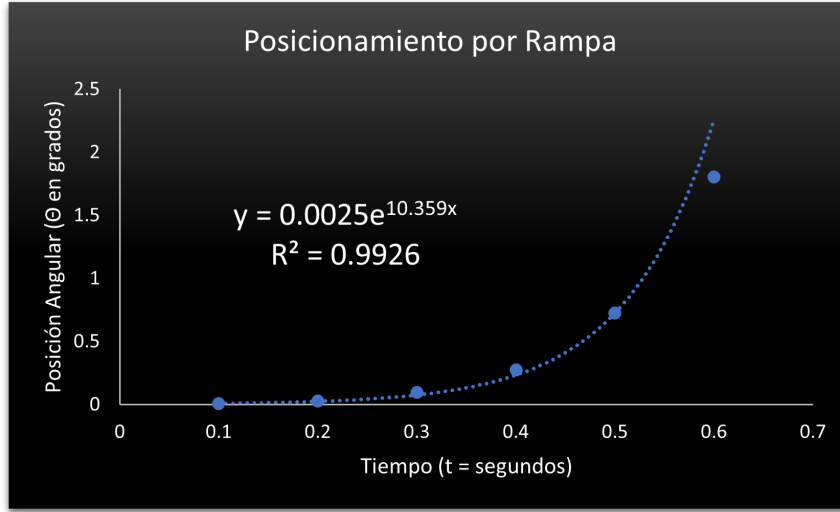


Figura 14: Gráfico de dispersión obtenido de los puntos generados

Por medio de la ecuación obtenida en Excel se pudo obtener la función para la posición angular al tener una velocidad trapezoidal (19).

$$\theta(t) = Ae^{-\alpha t} = \frac{100}{39413} \omega e^{\frac{10359}{1000} t} \quad (19)$$

Al tener ambas funciones que describen el sistema se deben pasar estas al dominio de la frecuencia por medio de la transformada de Laplace y así obtener la planta del sistema. Al realizar esto obtenemos la función de transferencia para ambas funciones.

$$G_2 = \Theta(s) = \frac{\theta}{\phi} \frac{\omega t}{s^2} \quad (20)$$

$$G_3 = \Theta(s) = \frac{\frac{100}{39413} \omega}{s + \frac{10359}{1000}} \quad (21)$$

7.3.2. Obtención de las constantes para el PID

Para obtener las constantes K_P , K_I y K_D para el control PID se utilizó el *PID Tuner* de *Matlab*, logrando obtener la respuesta deseada para cada sistema. Estas se obtuvieron ajustando la respuesta en el tiempo y el comportamiento transitorio en el *PID Tuner*.

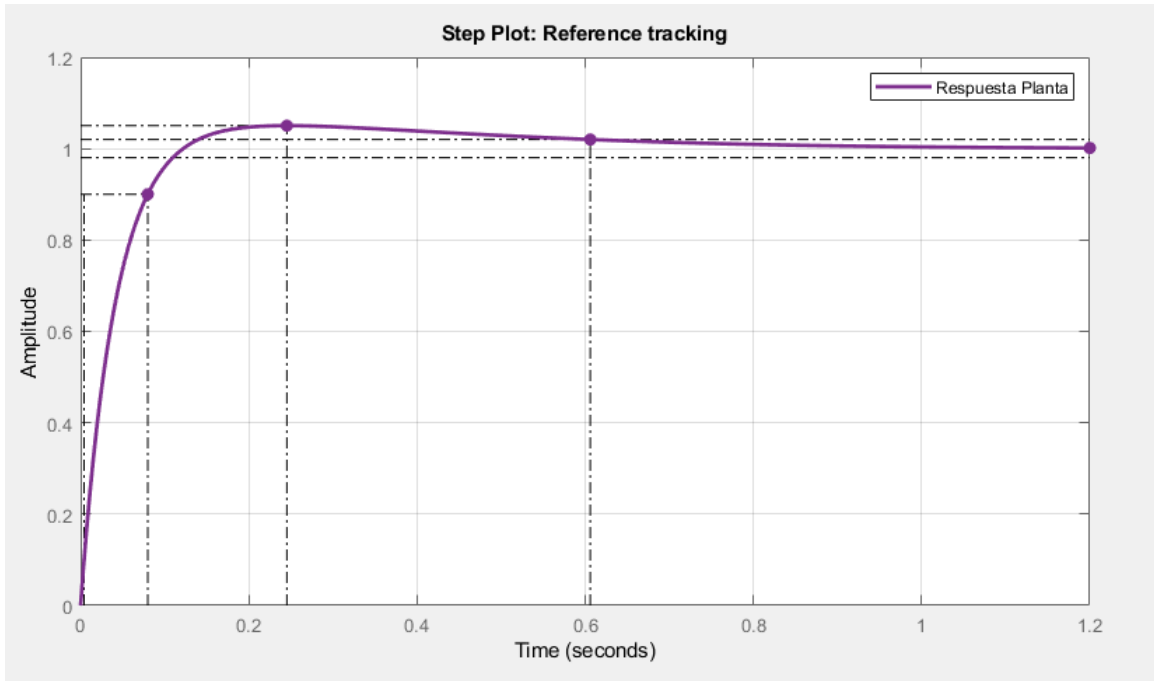


Figura 15: Obtención de constantes PID en *PID Tuner* para la planta detectada

Controller Parameters	
	Tuned
Kp	100.6854
Ki	243.7541
Kd	10.3973
Tf	n/a

Performance and Robustness	
	Tuned
Rise time	0.0757 seconds
Settling time	0.606 seconds
Overshoot	5.03 %
Peak	1.05
Gain margin	Inf dB @ NaN rad/s
Phase margin	86.9 deg @ 25.1 rad/s
Closed-loop stability	Stable

Figura 16: Parámetros PID en *PID Tuner* para la planta detectada

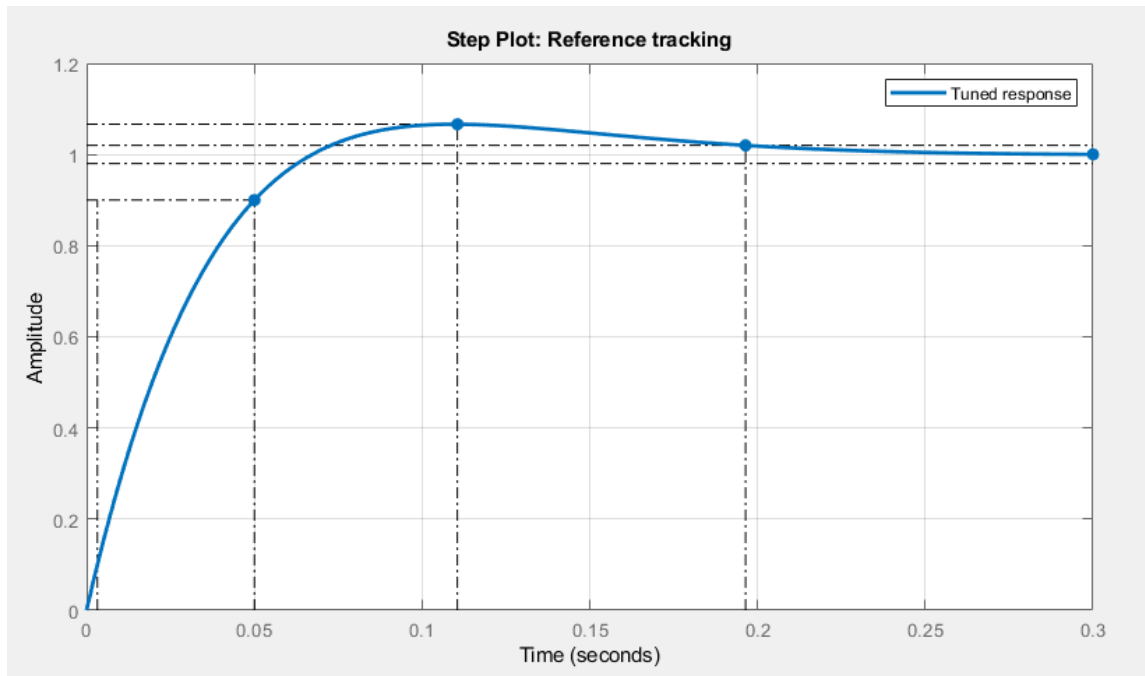


Figura 17: Obtención de constantes PID en *PID Tuner* para la planta de velocidad constante

Controller Parameters	
	Tuned
Kp	12664.1166
Ki	244235.8853
Kd	0
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	0.0468 seconds
Settling time	0.196 seconds
Overshoot	6.64 %
Peak	1.07
Gain margin	Inf dB @ NaN rad/s
Phase margin	77.7 deg @ 35.2 rad/s
Closed-loop stability	Stable

Figura 18: Parámetros PID en *PID Tuner* para la planta de velocidad constante

Controller Parameters	
	Tuned
Kp	84.1441
Ki	49.0951
Kd	23.0995
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	0.0433 seconds
Settling time	0.501 seconds
Overshoot	6.63 %
Peak	1.07
Gain margin	-37.1 dB @ 1.46 rad/s
Phase margin	85 deg @ 41.7 rad/s
Closed-loop stability	Stable

Figura 19: Parámetros PID en *PID Tuner* para la planta de velocidad en rampa

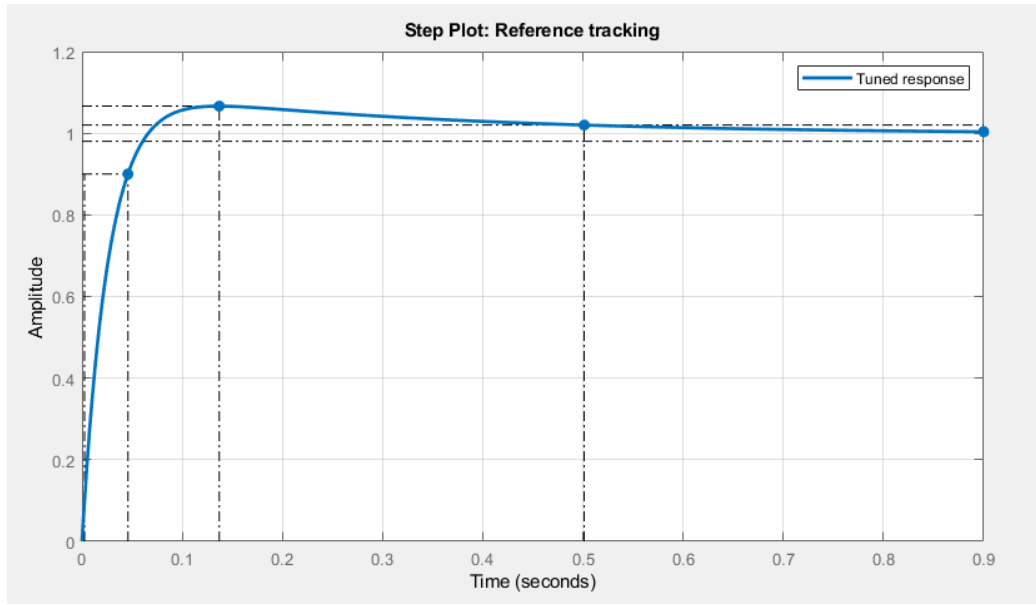


Figura 20: Obtención de constantes PID en *PID Tuner* para la planta de velocidad en rampa

Al obtener las constantes K_P , K_I y K_D se puede obtener la función de transferencia del sistema por medio de la función "feedback()" de *Matlab* y verificar la respuesta al escalón por medio de la función "step()".

$$TG_1 = \frac{3.404s^2 + 32.96s + 79.81}{0.1345s^3 + 4.538s^2 + 33.96s + 79.81} \quad (22)$$

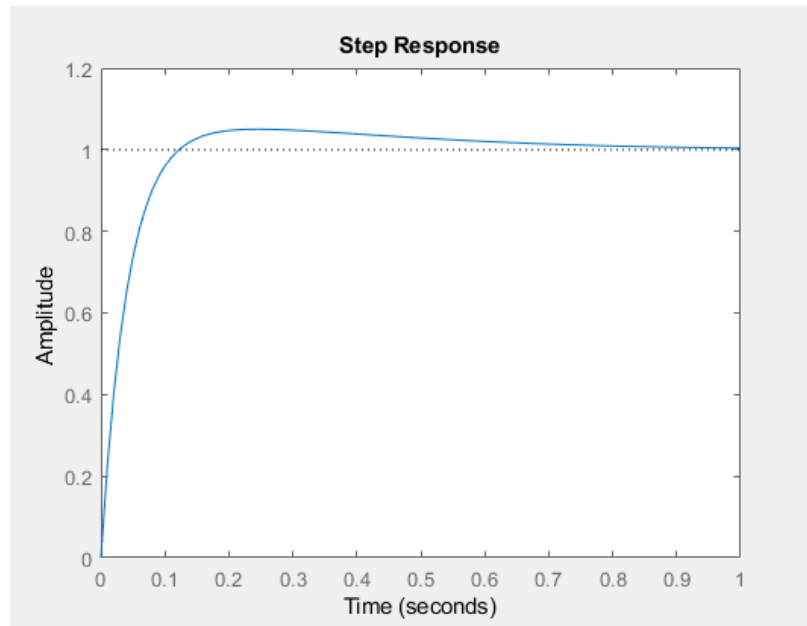


Figura 21: Verificación del PID de la planta detectada en respuesta al escalón

$$TG_2 = \frac{41.58s^2 + 151.5s + 88.37}{s^3 + 41.58s^2 + 151.5s + 88.37} \quad (23)$$

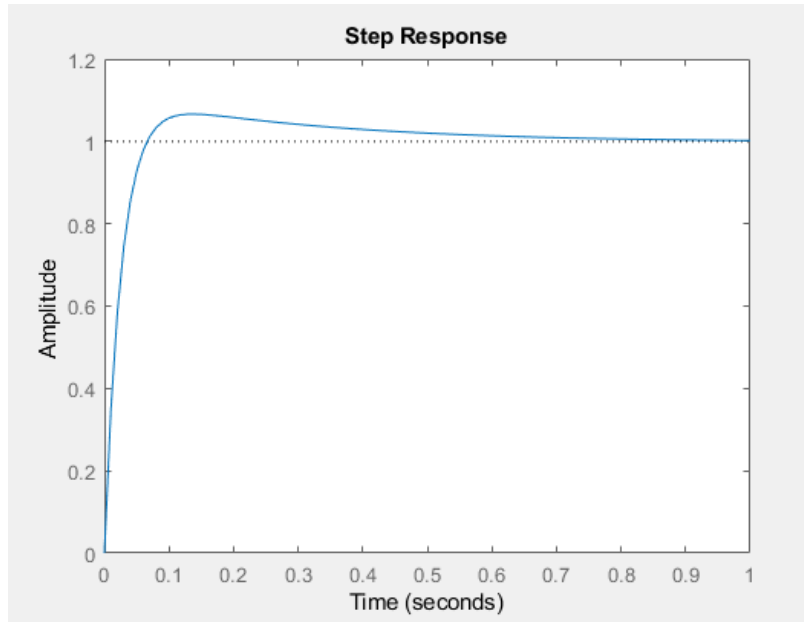


Figura 22: Verificación del PID de la planta con velocidad constante en respuesta al escalón

$$TG_3 = \frac{32.13s + 619.7}{s^2 + 42.49s + 619.7} \quad (24)$$

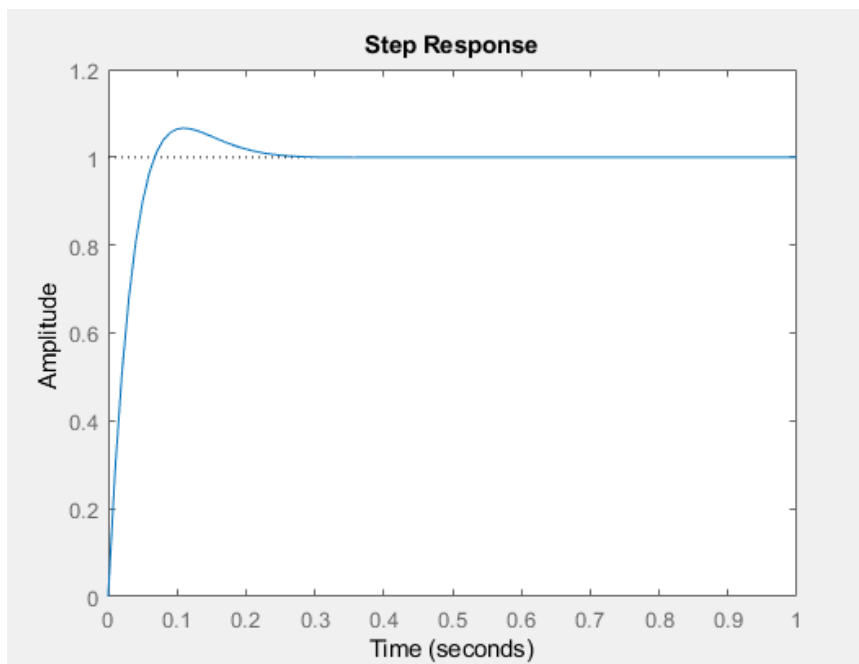


Figura 23: Verificación del PID de la planta con velocidad en rampa en respuesta al escalón

El control PID utiliza una realimentación visual, ya que el médico especialista estará generando el movimiento a través del control de mandos para llegar a la posición que el desee. Además, se utiliza el PID en el código para poder generar la cantidad de pulsos necesarios para llegar a la posición deseada.

7.4. Simulación

La crisis de salud mundial del Covid-19 causó limitantes en términos de la fabricación del prototipo del robot. Por lo tanto se decidió realizar una simulación del sistema de control en la herramienta *Webots*. Esta implementó el algoritmo de la cinemática inversa y los valores de las constantes PID para realizar el movimiento de las juntas.

7.4.1. Crear un robot en *Webots* con un diseño 3D

Webots ofrece varias opciones para realizar las simulaciones: desde utilizar robots diseñados y ya localizados en la base de datos hasta poder diseñar el robot dentro de *Webots*. Sin embargo, el programa también nos permite importar modelos 3D (ver Figura 24). Este es el método más adecuado para la implementación del brazo asistencial para cirugías cerebrales ya que se contaba con el archivo CAD del diseño previamente trabajado por mi compañero.

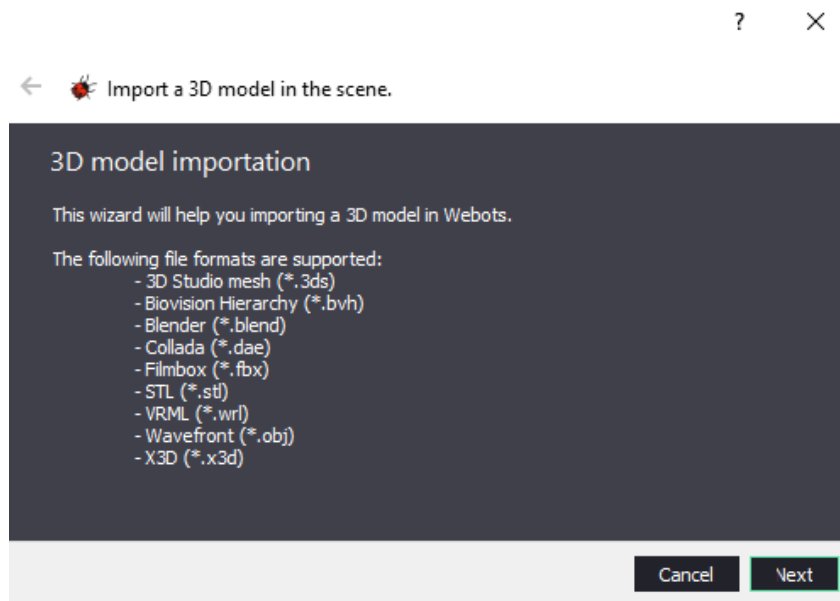


Figura 24: Archivos aceptados por *Webots* para importar diseños 3D

Antes de trabajar en *Webots* se debe obtener el archivo CAD en el formato aceptado por *Webots* (ver Figura 24). Es necesario planificar el eje central de nuestra figura ya que esto facilitará el posicionamiento de los ejes de coordenadas a la hora de pasar a *Webots*. En este caso, se utilizaron los ejes de los motores paso a paso y se exportaron las piezas a un formato de archivo de definición de geometría (.obj).

Crear el modelo 3D con extensión .wbo

Para poder crear un diseño 3D deseado en lugar de un diseño ya incluido en la base de datos de *Webots*, se puede importar un archivo con el formato aceptado por el programa para luego poder exportarlo a un formato .wbo y poder utilizarlo en futuras aplicaciones dentro del mismo. Primero se debe importar el archivo por medio de la pestaña de "Archivo", como se puede observar en la Figura 25. Luego se debe exportar y guardar bajo el nombre deseado. Se debe hacer clic derecho sobre el elemento que se desee exportar como se puede observar en la Figura 26.

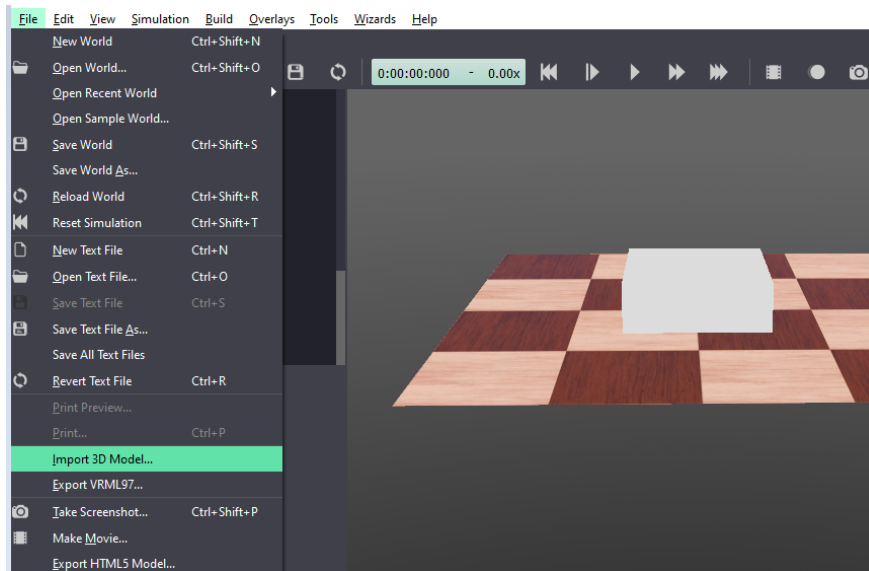


Figura 25: Importar un archivo CAD

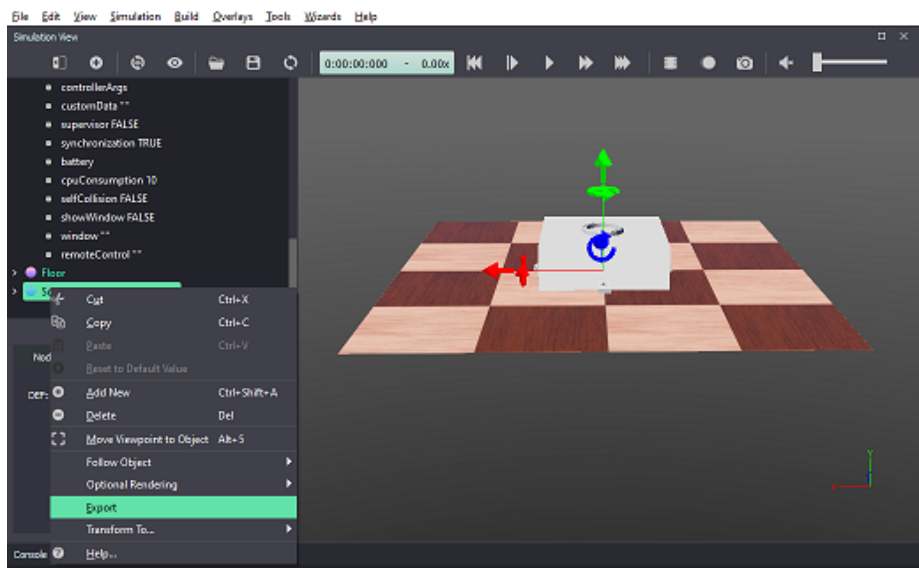


Figura 26: Exportar un archivo CAD

Crear el elemento de robot

Al estar en un nuevo mundo dentro de *Webots* se debe presionar la opción de agregar un nuevo elemento y añadir el nodo de base "Robot" (ver Figura 27). Esto creará un robot como nodo con el cual podremos interactuar.

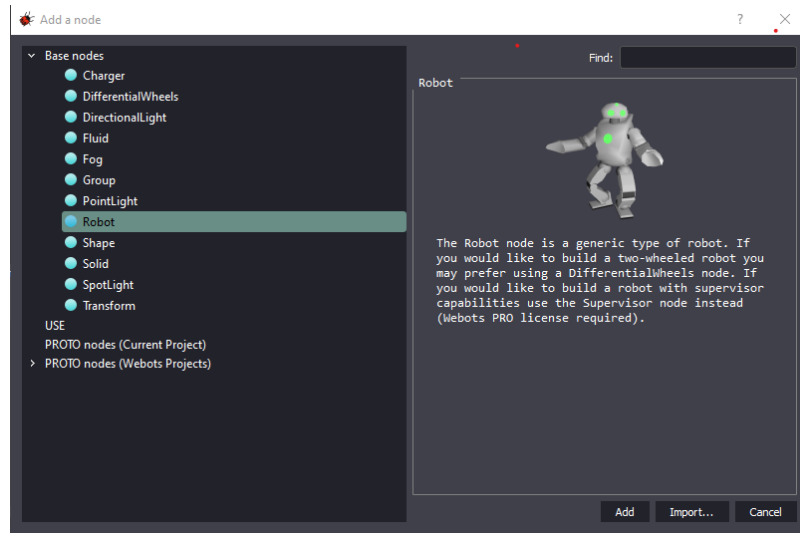


Figura 27: Crear un robot en *Webots*

Sin embargo, para poder tener un robot funcional se deben agregar las especificaciones del mismo. En el caso del brazo asistencial para cirugías cerebrales nuestro robot consta de 6 juntas de bisagra (HingeJoint). Estas se pueden agregar al seleccionar la subdivisión de "children" dentro de nuestro robot y hacer *clic derecho* → *agregar nuevo*. Se busca la opción de junta de bisagra en los nodos de base (ver Figura 28).

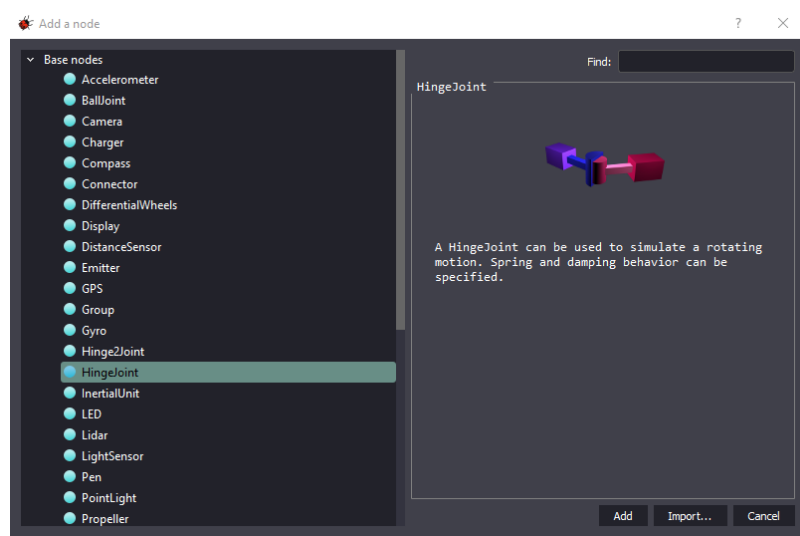


Figura 28: Tipo de junta

Una vez ya tenemos nuestra junta es importante especificar que tipo de dispositivo es la junta. *Webots* nos permite seleccionar para juntas de bisagra tres dispositivos: Un Freno, un Sensor de Posición o un Motor Rotacional (ver Figura 29). En nuestro caso todas las juntas son motores paso a paso por lo que se seleccionó la opción de Motor Rotacional.

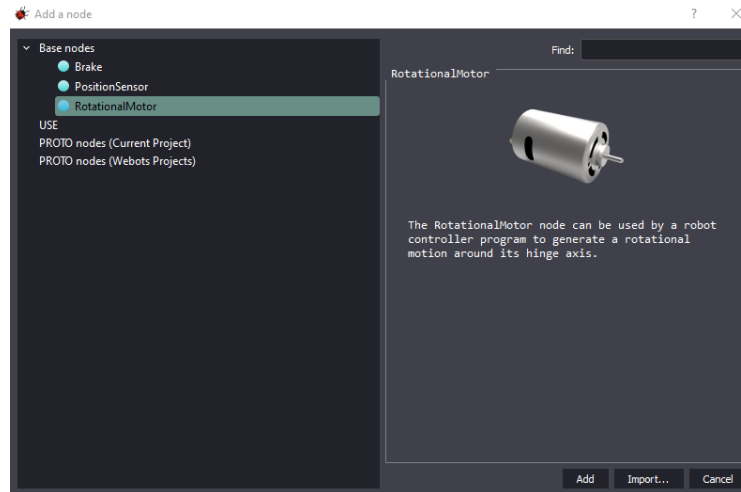


Figura 29: Tipo de dispositivo

Al haber terminado con las especificaciones de la junta se debe seleccionar que tipo de forma tendrá por lo que se debe seleccionar un sólido (ver Figura 30). Esto creará un sólido donde podemos importar nuestro archivo previamente exportado a un formato *.wbo*.

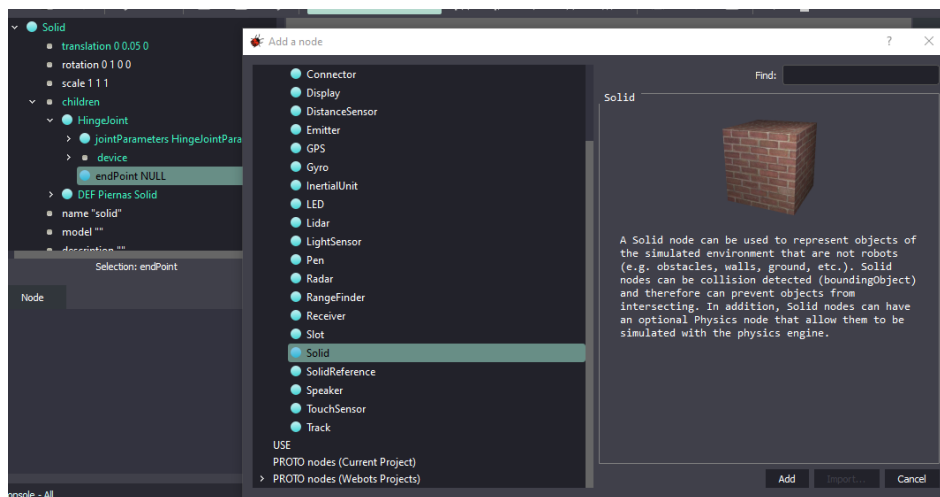


Figura 30: Aspecto de la junta

Al haber realizado esto se puede continuar con las siguientes juntas. Es importante tomar en consideración que debe tener movimiento al accionar una junta en específico. Para poder obtener un movimiento del brazo completo simulando un brazo verdadero fue necesario utilizar una metodología de "árbol" o "cadena", ya que cada junta debía crearse dentro de la junta anterior. Al no hacer esto, se tenía movimiento únicamente en la junta seleccionada y

no en las juntas dependientes. En otras palabras, esto ocasionaba que al mover la junta de la base únicamente se moviera la base y no el cuerpo completo, lo cual no tiene congruencia con la realidad.

7.4.2. Resultados de la simulación

Se realizó una simulación en la cual se varió todos los motores. Los primeros tres por medio de cinemática inversa y un controlador PID para todos los motores.

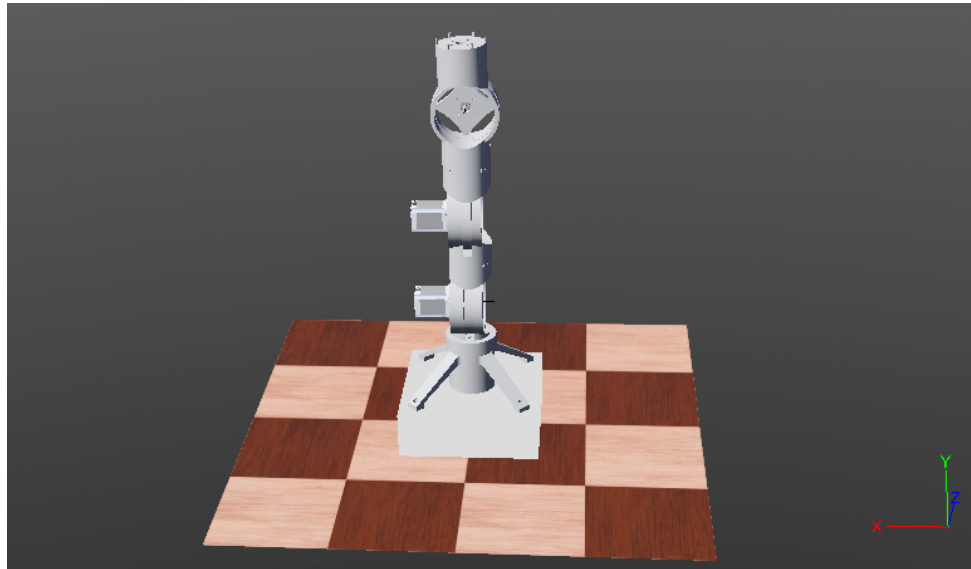


Figura 31: Simulación 1, posición inicial

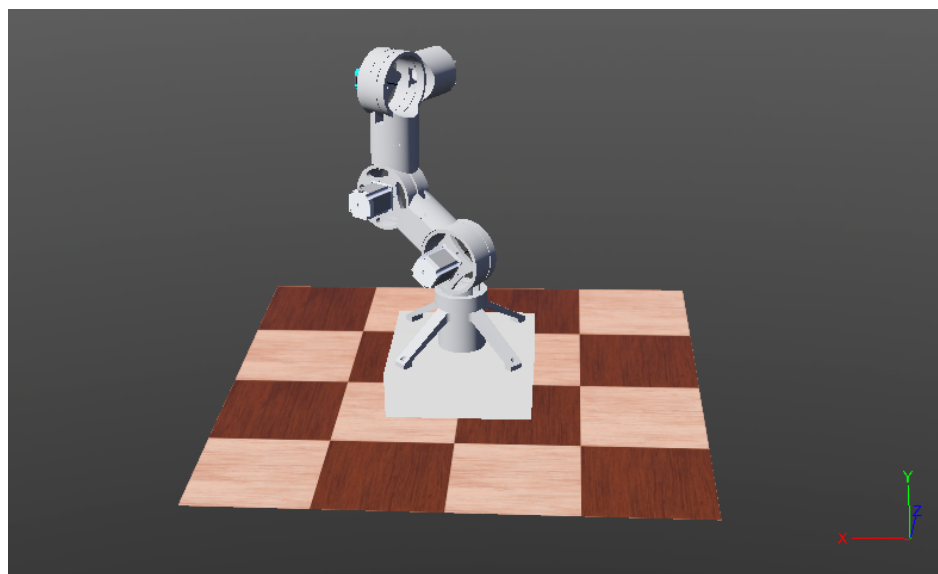


Figura 32: Simulación 1, posición final

Se realizó una simulación en la cual se varió algunos de los motores. Sin embargo, el control de estos fue por medio de cinemática inversa para los primeros tres y un controlador PID para todos los motores.

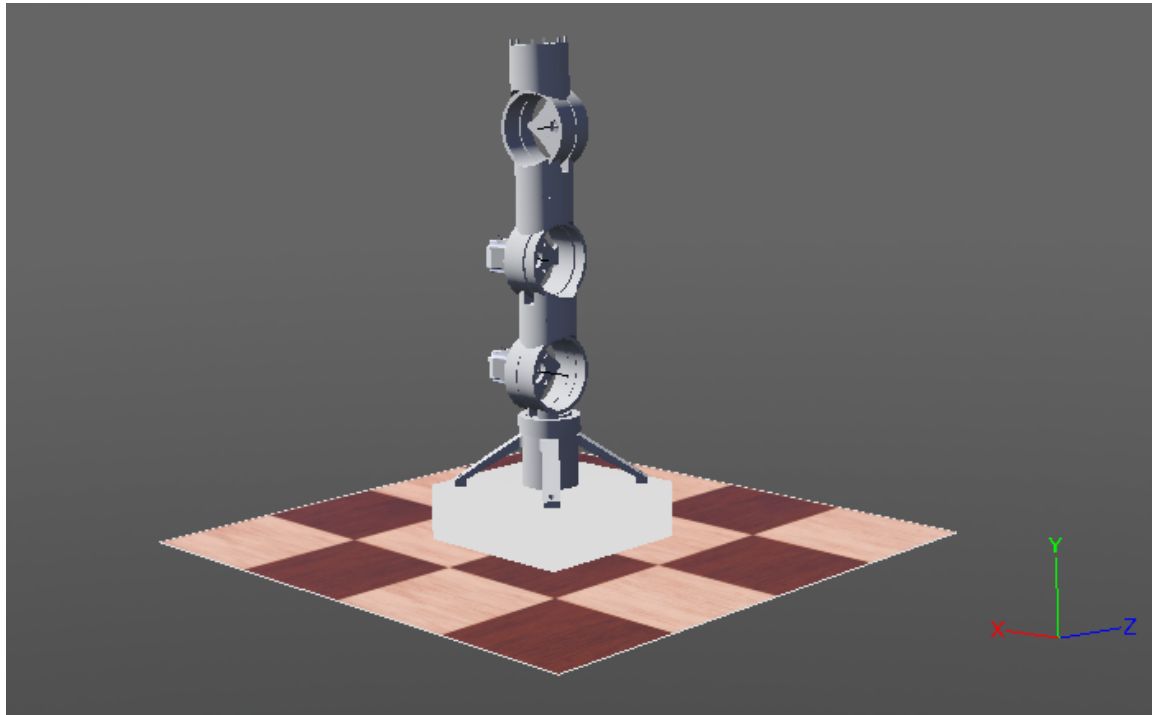


Figura 33: Simulación 2, posición inicial

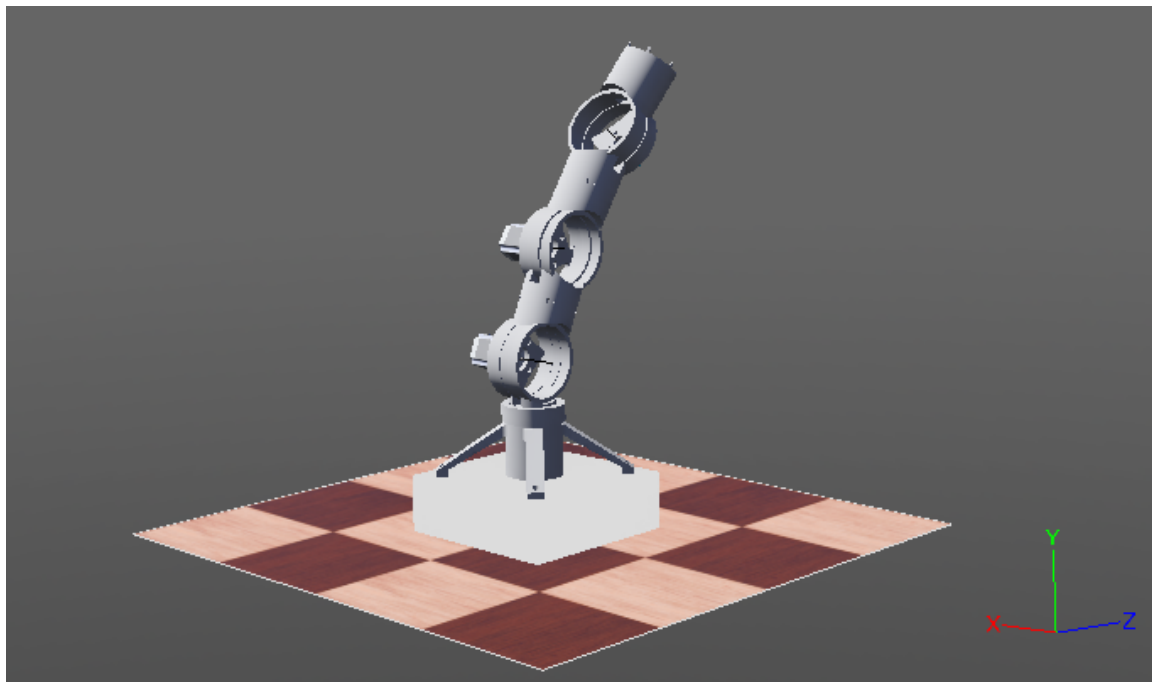


Figura 34: Simulación 2, captura en movimiento 1



Figura 35: Simulación 2, captura en movimiento 2



Figura 36: Simulación 2, captura en movimiento 3

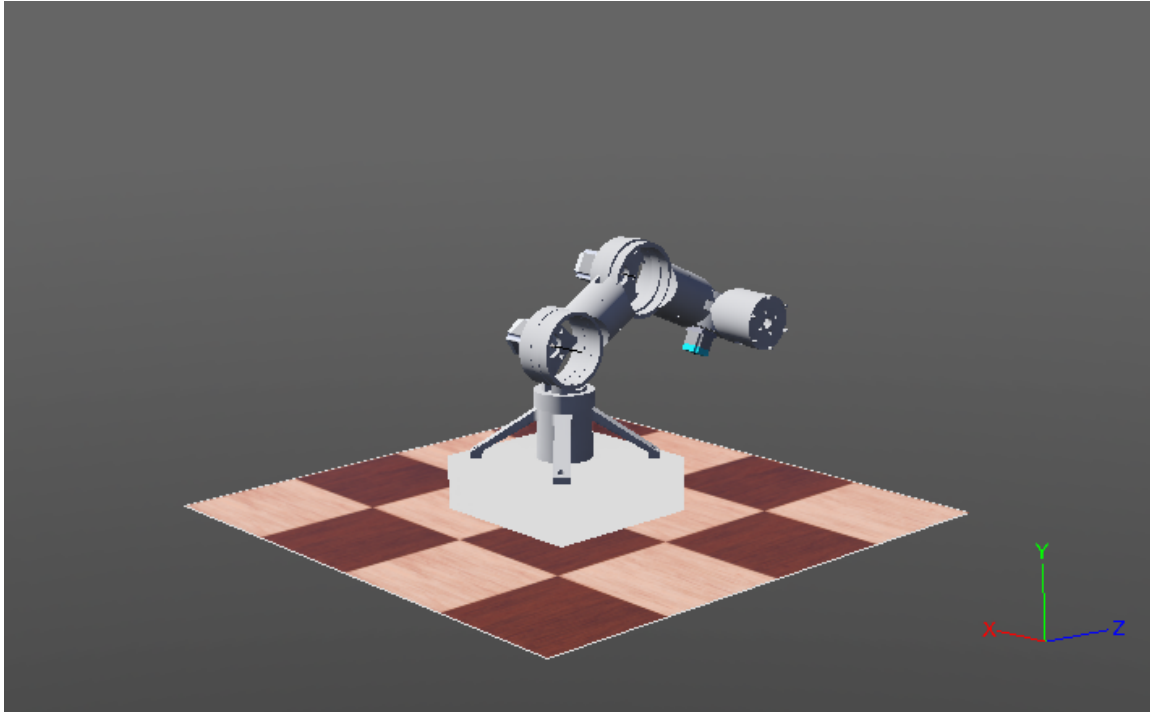


Figura 37: Simulación 2, posición final

7.5. Pruebas

Después de realizar pruebas de la rampa idealizada con el cambio de ángulos por paso (Stepangle), se pudo observar que el movimiento del motor era brusco. Esto fue debido a que al estar utilizando ángulos de paso grandes (1 a 1/4 veces la cantidad de ángulos designados por paso) el tiempo entre los pasos ocasionaba vibraciones las cuales podrían afectar el movimiento suave del robot. Tomando esto en cuenta, se decidió controlar la rampa por medio del tiempo y no por medio del cambio de ángulos por paso.

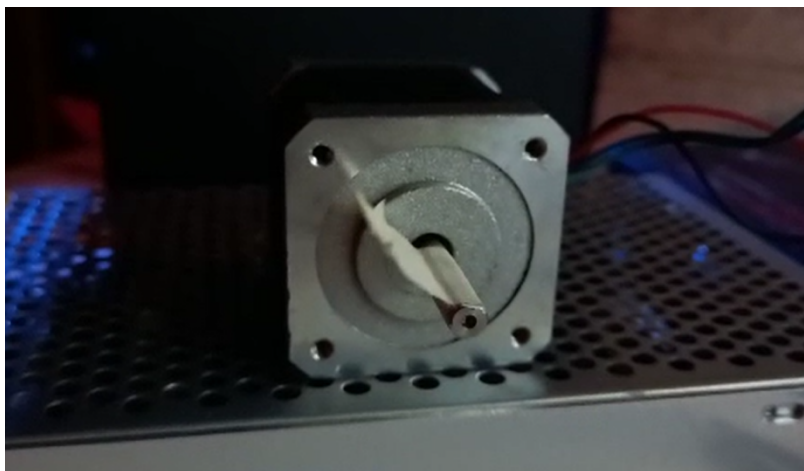


Figura 38: Motor Stepper durante pruebas realizadas

En este capítulo se explica el diseño del control de mandos. Tanto la selección de los actuadores que fueron utilizados como los distintos prototipos que se realizaron para poder llegar a un diseño funcional que se puede fabricar por medio de tecnologías aditivas como la impresión 3d. También se explica el funcionamiento del mismo.

Debido a la crisis mundial de salud ocasionada por el Covid-19 no fue posible fabricar un prototipo, pero se propone a continuación el diseño completo.

8.1. Selección de actuadores

Para poder obtener un control de mandos amigable para el usuario se decidieron incorporar distintos actuadores que ayudarán a realizar este objetivo. Se implementaron cuatro actuadores y una pantalla LCD para poder mostrar los parámetros de velocidad.

Control de *Wii Nunchuk*

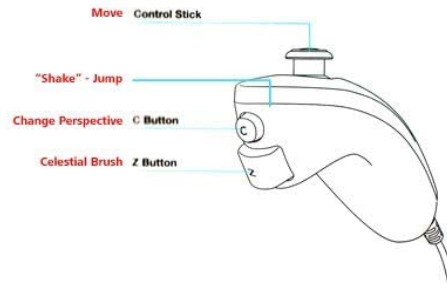
Se seleccionó un control de *Wii Nunchuk* (Figura: 39) para poder facilitar el movimiento del brazo. El joystick controla la posición y los botones la velocidad.

Pantalla LCD

Se seleccionó una pantalla LCD (Figura: 40) para poder desplegar información relevante para el usuario como la velocidad seleccionada y las juntas o junta a mover.

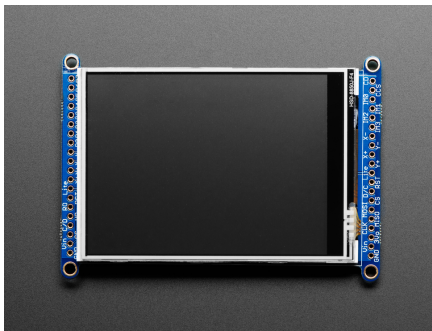


(a) Vista del control [27]

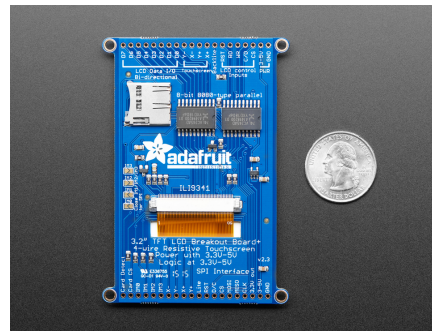


(b) Vista de los botones [28]

Figura 39: Control Nunchuk del Nintendo Wii



(a) Vista Frontal de la Pantalla [29]



(b) Vista Posterior de la Pantalla [30]

Figura 40: Pantalla LCD TFT 3.2"

Interruptor SPST (Switch)

Se seleccionó un interruptor SPST (Figura: 41) de dos estados.



Figura 41: Interruptor SPST [31]

Selector

Se implementó un selector rotativo (Figura: 42) para poder seleccionar qué juntas son las que se deben mover. Este tiene cuatro posiciones para las primeras tres juntas, y las otras tres restantes.



Figura 42: Selector rotativo [32]

Botón de pulso (Pushbutton)

El botón de pulso (Figura: 43) se implementó como recurso de auto-bloqueo. Esto es debido a que los procedimientos para los cuales es utilizado el brazo robótico asistencial para cirugías son muy delicados y pueden lastimar gravemente al paciente si existe algún error. Para mitigar este se decidió implementar un sistema de auto-bloqueo mecánico, trabajado en la Tesis de mi compañero Bryan Raúl Santos Avila [19] y un auto-bloqueo controlado digitalmente. Este bloqueo manual es accionado por medio del botón de pulso.



Figura 43: Botón de push [33]

8.2. Diseño del control de mandos

El control de mandos fue diseñado en programas de diseño asistido por computadora (CAD). El diseño implementa un pantalla para poder desplegar la velocidad y las juntas que se están moviendo. Aparte de eso, el control de mandos debe considerar el espacio para el circuito impreso (PCB) del mismo.

8.2.1. Primer prototipo

El primer prototipo se realizó en Fusion360 implementando una carcasa para los componentes. Este prototipo fue diseñado para ser impreso en 3D. Sin embargo, es una forma muy ineficiente de fabricar, ya que es mucho material de PLA que se utilizaría y se podría utilizar algún otro material como MDF para fabricar la carcasa. Se utilizó una pantalla LCD TFT 2.4", ya que está es la que se iba a utilizar a la hora de realizar el primer prototipo. Además, se utilizó un switch de tres posiciones, un botón de emergencia y un selector.

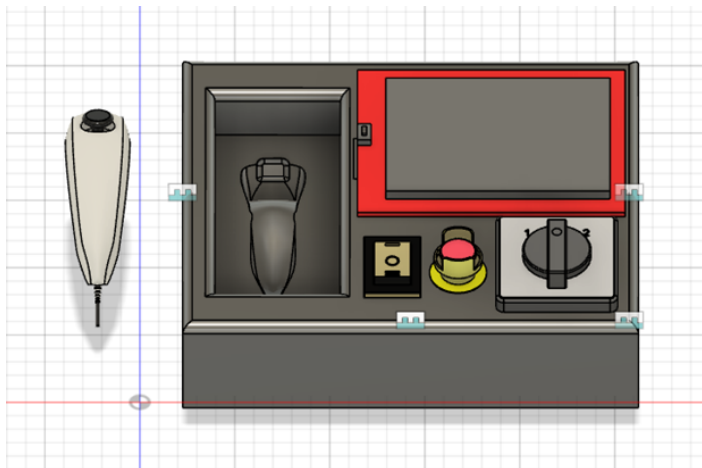


Figura 44: Control de mandos prototipo 1 vista superior en ángulo

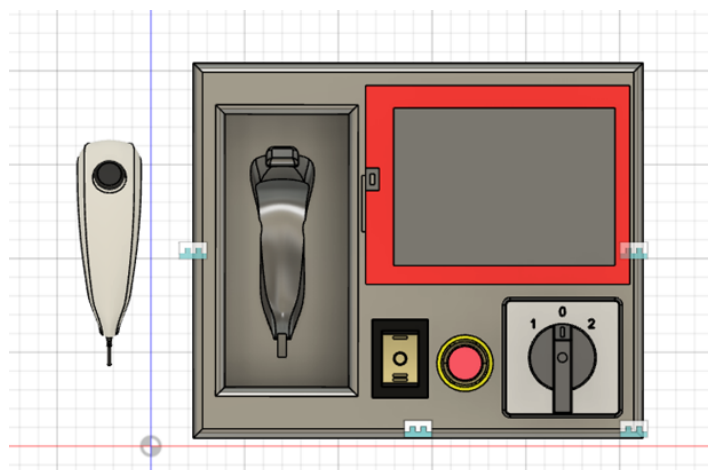


Figura 45: Control de mandos prototipo 1 vista superior

8.2.2. Segundo prototipo

El segundo prototipo fue diseñado en Inventor. En esta etapa del proyecto se consideró el material a utilizar para la fabricación y se concluyó que el mejor material es MDF (*Medium Density Fiber*) para la carcasa y utilizar impresión 3D para algunos componente. Como se puede observar en el control de mandos el lugar del control de *Wii Nunchuck* fue cambiado de un compartimiento de impresión 3D a dos apoyadores para el control. El switch fue cambiado a uno con únicamente dos estados, encendido y apagado. Asimismo, se cambió el selector y el pushbuton. Todos los componentes que se indican en el diseño final (Figura: 48) son utilizados ya que esos son los actuadores seleccionados para la fabricación. También se le agregaron los conectores para el cableado interno entre el control de mandos y el brazo robótico asistencial.

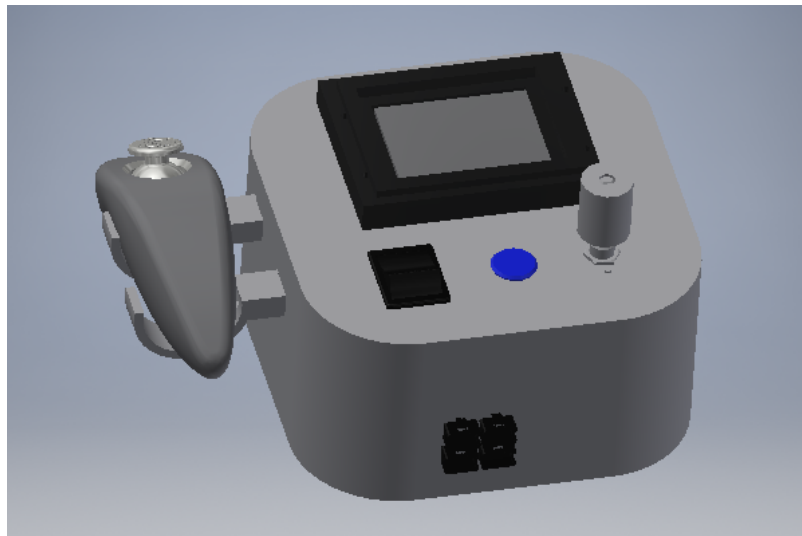


Figura 46: Control de mandos prototipo 2 primera etapa vista isométrica



Figura 47: Control de mandos prototipo 2 primera etapa vista superior

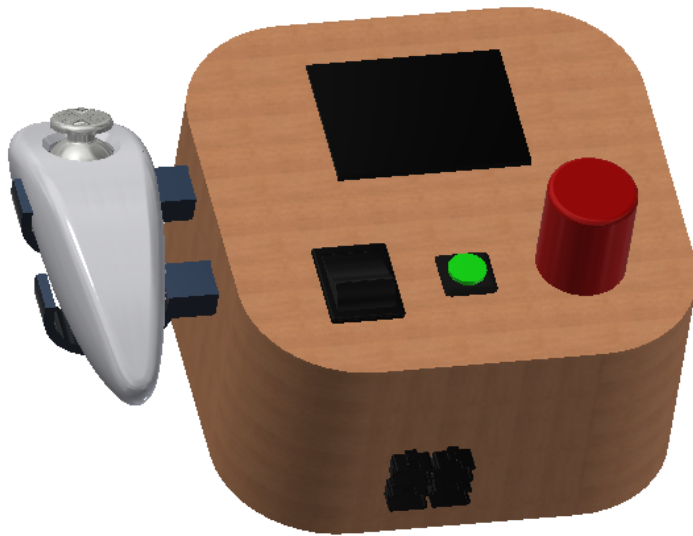


Figura 48: Control de mandos prototipo 2 vista isométrica



Figura 49: Control de mandos prototipo 2 vista superior

8.2.3. Tercer prototipo

Para el tercer prototipo se continuó trabajando en inventor. En esta iteración se enfocó en los detalles del control de mandos como las conexiones, tanto entre el control de Wii Nunchuk y el Nunchuky como entre ambas placas de circuitos impresos empleadas. También se diseño una nueva cubierta para la perilla del selector.



Figura 50: Control de mandos prototipo 3 vista isométrica



Figura 51: Control de mandos prototipo 3 vista superior

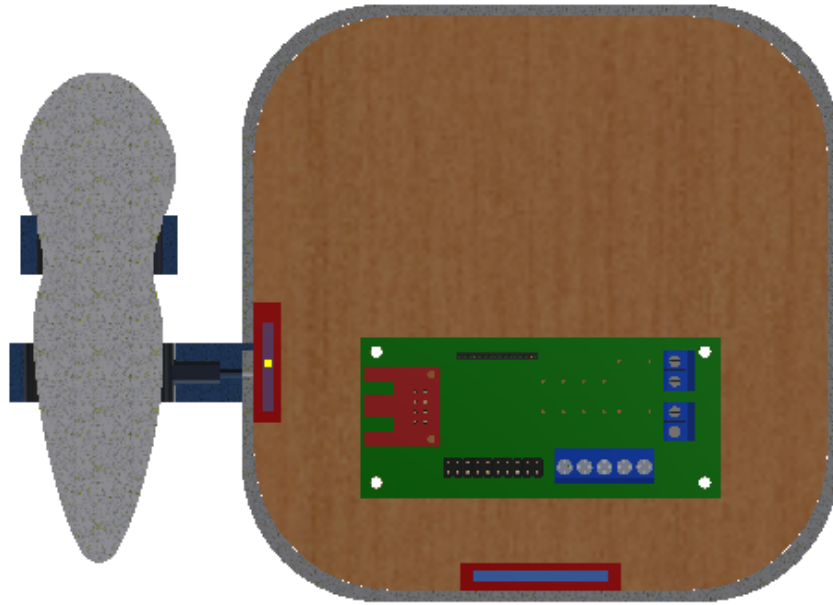


Figura 52: Control de mandos prototipo 3 vista del PCB

8.2.4. Cuarto prototipo (prototipo físico)

Para el cuarto prototipo se fabricó el control de mandos. Este se realizó por medio de impresión 3D y corte láser de MDF.



Figura 53: Control de mandos prototipo 4 vista isométrica

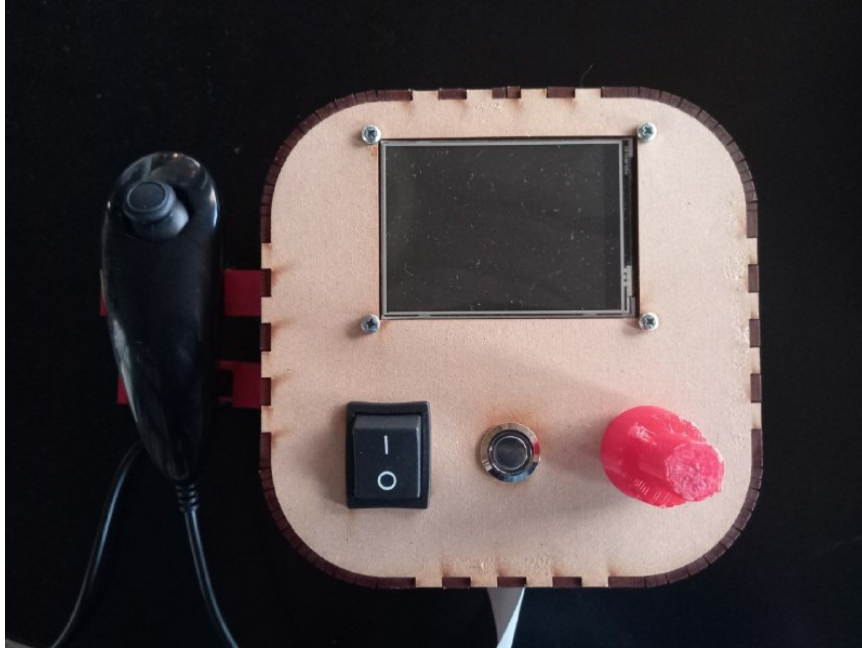


Figura 54: Control de mandos prototipo 4 vista superior

8.3. Manual de uso para el control de mandos

Se redactó un manual de uso para el control de mandos para especificar los componentes del mismo y el uso adecuado para mover el brazo robótico asistencial para cirugías (Ver Anexos).

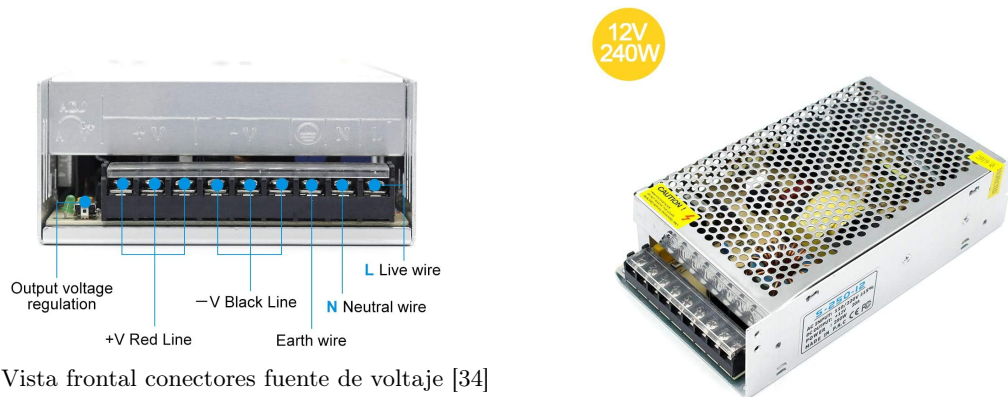
El manual incluye los siguientes temas:

1. Los componentes y actuadores del mismo.
2. Una sección de la metodología de uso.

En este capítulo se explica lo que es el diseño eléctrico: la selección de los componentes que fueron utilizados y los distintos prototipos que se realizaron para poder llegar a un diseño de PCB funcional que se puede fabricar para el uso en el brazo robótico asistencial para cirugías.

Debido a la crisis mundial de salud ocasionada por el Covid-19 no fue posible fabricar un prototipo físico de los circuitos impresos y de la base del robot. Sin embargo, se propone a continuación el diseño completo.

9.1. Selección de componentes



(a) Vista frontal conectores fuente de voltaje [34]

(b) Vista isométrica [35]

Figura 55: Fuente de voltaje

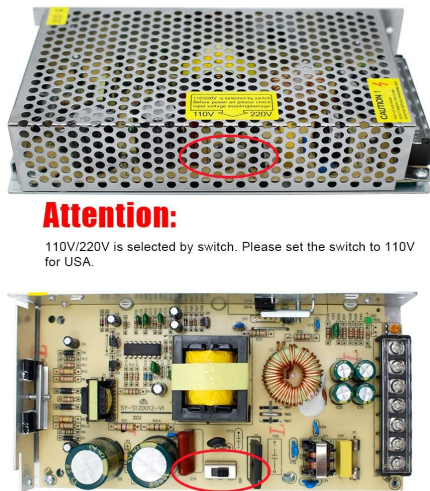


Figura 56: Vista interna fuente de voltaje [36]

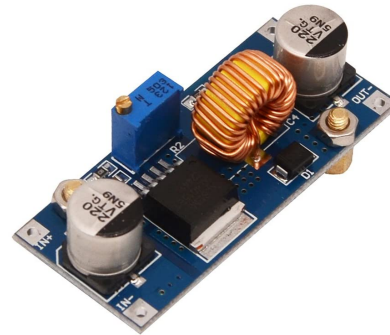


Figura 57: Conversor de potencia [37]

9.1.1. Fuente de voltaje 12V 240W

La fuente de voltaje se seleccionó basándose en el pico de corriente de cada motor. Los NEMA 23 utilizan 5A cada uno y los NEMA 17 aproximadamente 2.5A. El total de amperaje es de 20A ($5 \cdot 2 + 2.5 \cdot 4 = 20$). El voltaje utilizado por los motores es 12 V. Dado esto, se pudo obtener la potencia requerida por medio de la ecuación (25), obteniendo como resultado una potencia de 240 W. Esta se puede observar en las figuras 55 y 56.

$$P = V * I \quad (25)$$

9.1.2. Conversor de voltaje (Stepdown Converter)

Este conversor de voltaje tiene un rango de voltaje de entrada de 5V a 32V y tiene un voltaje de salida ajustable por medio de un potenciómetro desde 0.8V hasta 30V. El voltaje debe ser un voltaje de corriente directa. Debido a estas características se decidió utilizar un conversor de potencia (Figura: 57) para poder obtenerlos 5V requeridos por la Tiva C y los otros actuadores.

9.1.3. Conectores

Se implementaron tres tipos de conexión de cables para poder simplificar el montaje y desmontaje del control de mandos y de la base del motor. Se utilizó un cable (Figura: 58a) y un adaptador de panel IEC 320 C14 (Figura: 58b) para poder transmitir la potencia a la fuente de poder de 12 V. Se utilizó conectores de 4-pines JST SM (Figura: 59) para la conexión entre el control de mandos y la base del robot. El tercer tipo de cable se utilizó para las conexiones internas entre los componentes para poder tener una forma fácil de desensamblar el control de mandos y la base del robot. Estos son conectores JST de dos pines macho y hembra (Figura: 60).

El uso de estos conectores nos entrega una forma estética y práctica para poder adjuntarlos a la carcasa del mando de control y de la base del robot. De esta forma se tiene una fácil movilidad, es seguro y simple de conectar.



(a) Conectores del cable [38]



(b) Montaje de panel [39]

Figura 58: Cable de poder IEC 320 C14



Figura 59: Conectores interconexiones [40]



Figura 60: Conectores internos [41]

9.1.4. *Nunchuky*

Se utilizó un *Nunchuky* (Figura: 61) para poder el control de *Wii Nunchuk* a al PCB. Este *board* utiliza protocolo I2C.

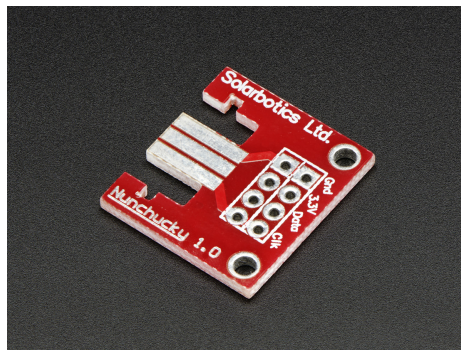


Figura 61: *Nunchuky*

9.2. Primer prototipo

Para el primer prototipo se consideraron únicamente los *drivers* del motor y los actuadores a utilizar. Debido a esto se realizó un esquemático y una placa de circuito impreso (PCB).

9.2.1. Esquemático

El esquemático (Figura: 62) fue diseñado en EasyEDA utilizando componentes diseñados por otros contribuyentes en la misma plataforma. El esquemático detalla la conexión entre los componentes pensados en utilizarse durante el primer prototipado. Estos son la Tiva C, los *drivers* A4988, una pantalla LCD TFT de 2.4" y los actuadores del control. Sin embargo, no se incluyó el *nunchuky* ni la división de los voltajes. Se concluyó que el prototipo podía mejorarse en otro programa de cómputo por lo que se procedió a la segunda iteración del mismo en lugar de terminarlo.

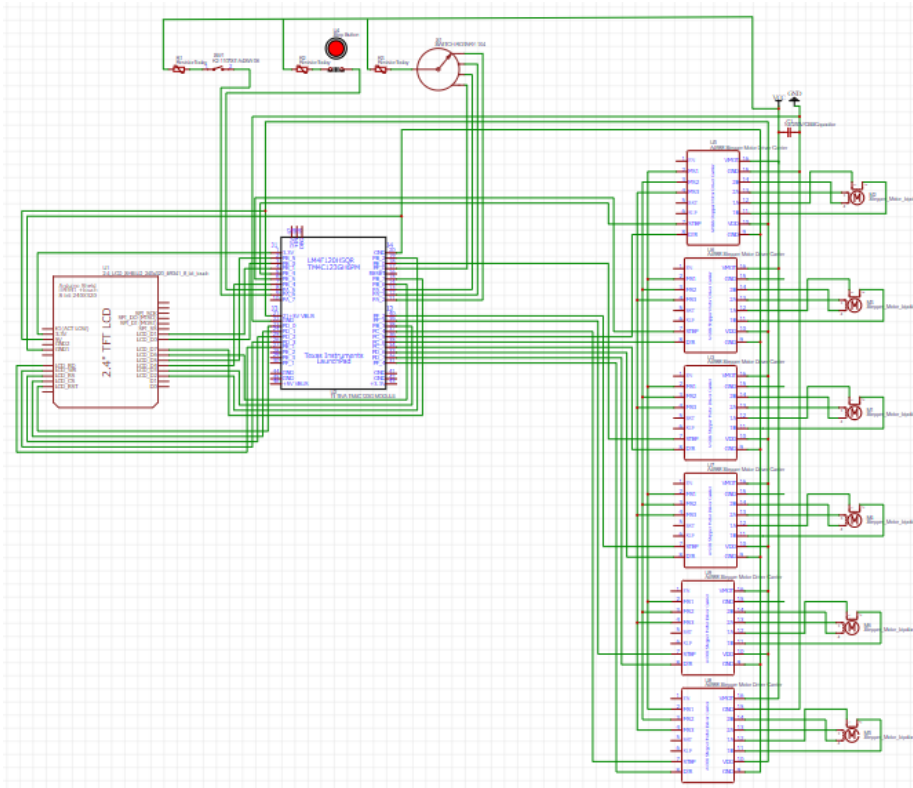


Figura 62: Esquemático en EasyEDA

9.2.2. Placa de circuito impreso (PCB)

La placa de circuito impreso (PCB) fue diseñada a partir del esquemático en EasyEDA (Figura: 63). Para este prototipo se pensó que el PCB estaría únicamente en el control de mandos por lo que se diseñó únicamente un PCB.

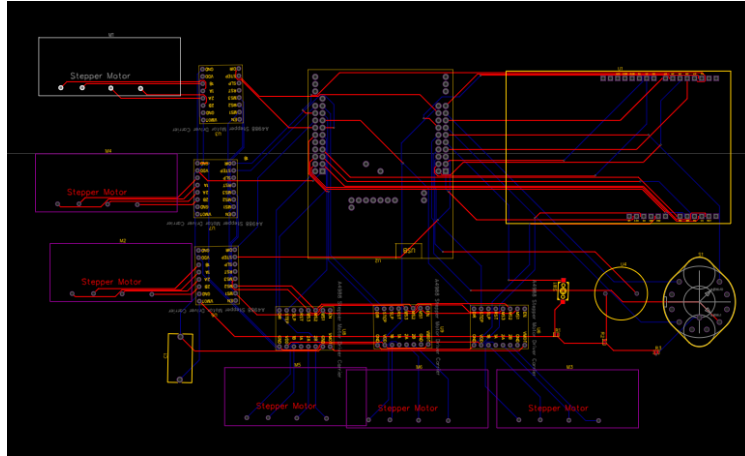


Figura 63: PCB en EasyEDA

9.3. Segundo prototipo

Para el segundo prototipo se tomaron en cuenta los componentes finales a utilizar y se crearon librerías de los componentes para los esquemáticos y para los PCBs. También se diseñó una base para el robot donde estarían situados los *drivers* para los motores y la fuente de poder.

9.3.1. Base para el robot

La base para el robot se diseñó pensando que sería cortada en MDF ya que debido al tamaño es más eficiente fabricar las piezas en material MDF con una cortadora láser y algunas piezas más detalladas con impresión 3d. Esta fue diseñada con los conectores que fueron descritos previamente en la sección de "Selección de Componentes".

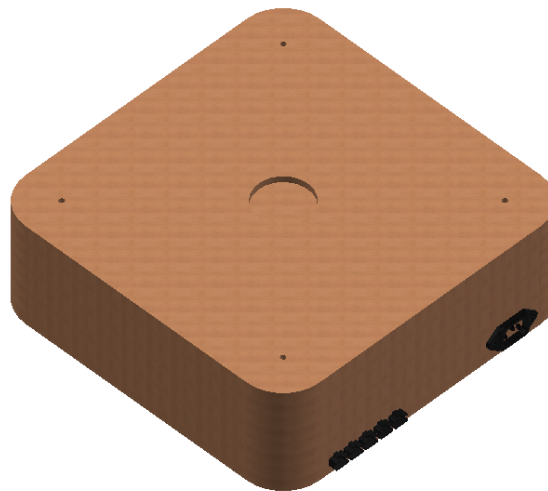


Figura 64: Vista isométrica base del brazo robótico



Figura 65: Vista frontal base del brazo robótico

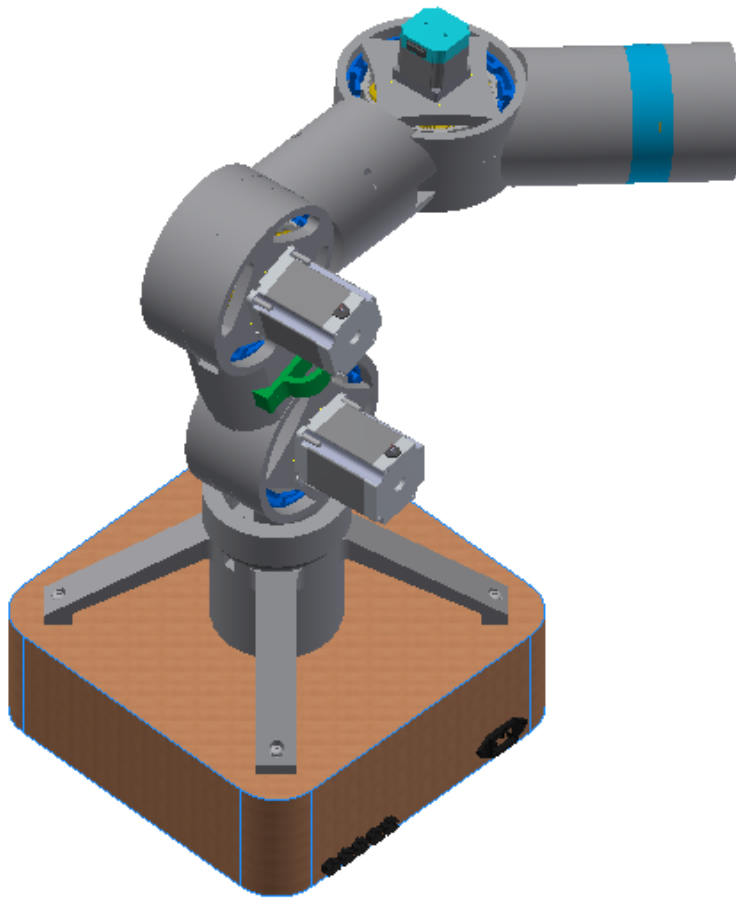


Figura 66: Brazo robótico con base

9.3.2. Esquemático

Para el segundo prototipo se realizaron varios cambios. Se utilizaron *drivers* de mayor potencia para ajustarse a las necesidades de los motores. También se cambió la pantalla a utilizar por una pantalla TFT 3.2" para no perder margen de imagen en el display. También se agregó el convertidor *stepdown*, para poder obtener el voltaje requerido para la Tiva C y no tener que incluir otra fuente de voltaje de 5V. Esto se utilizó en lugar de un divisor de voltaje para no insertar nuevas impedancias al circuito.

El esquemático principal (Figura 67) representa las conexiones del sistema completo. Se muestra la conexión entre la Tiva C y los distintos actuadores. Además de este esquemático también se diseñaron dos esquemáticos adicionales que representan la parte del esquemático principal que va en el control de mandos (Figura: 68) y en la base del robot (Figura: 69) enfocándose únicamente en las conexiones internas que son las que se representarán en los PCBs.

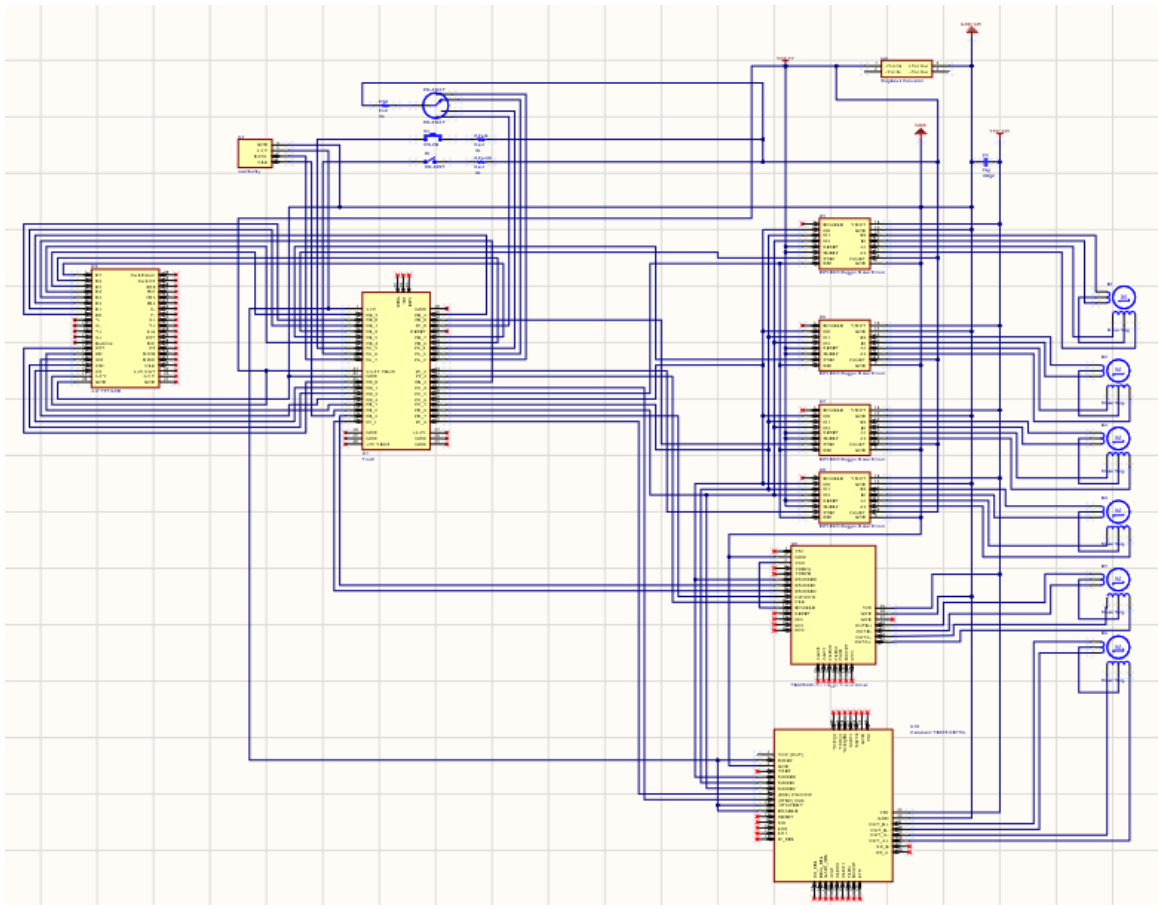


Figura 67: Esquemático principal

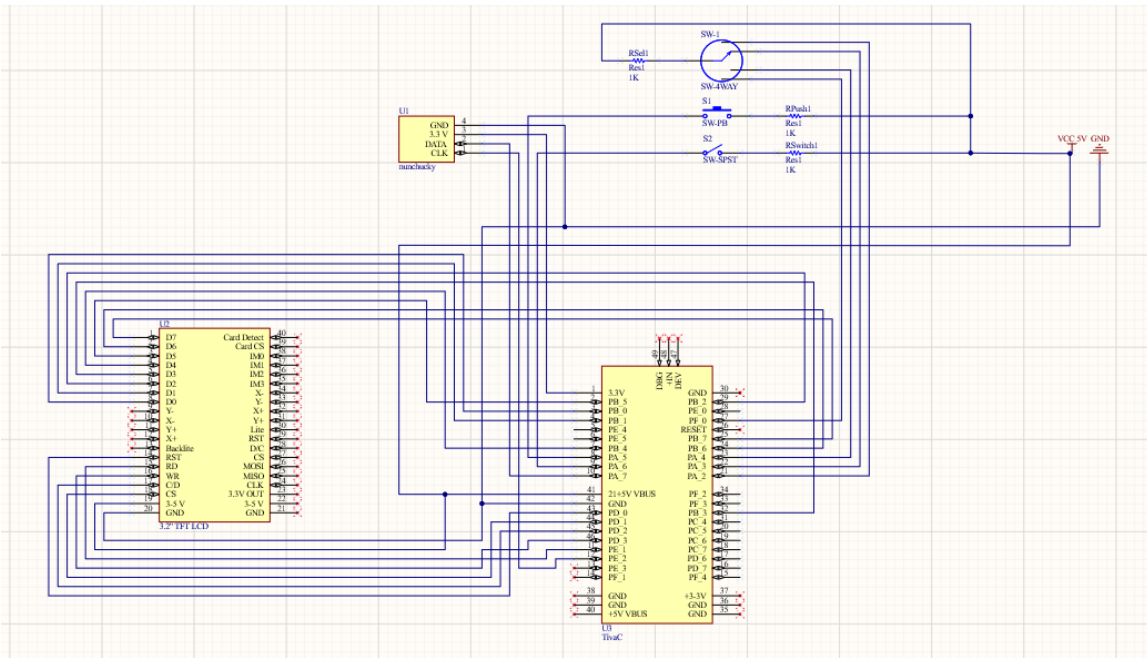


Figura 68: Esquemático control de mandos

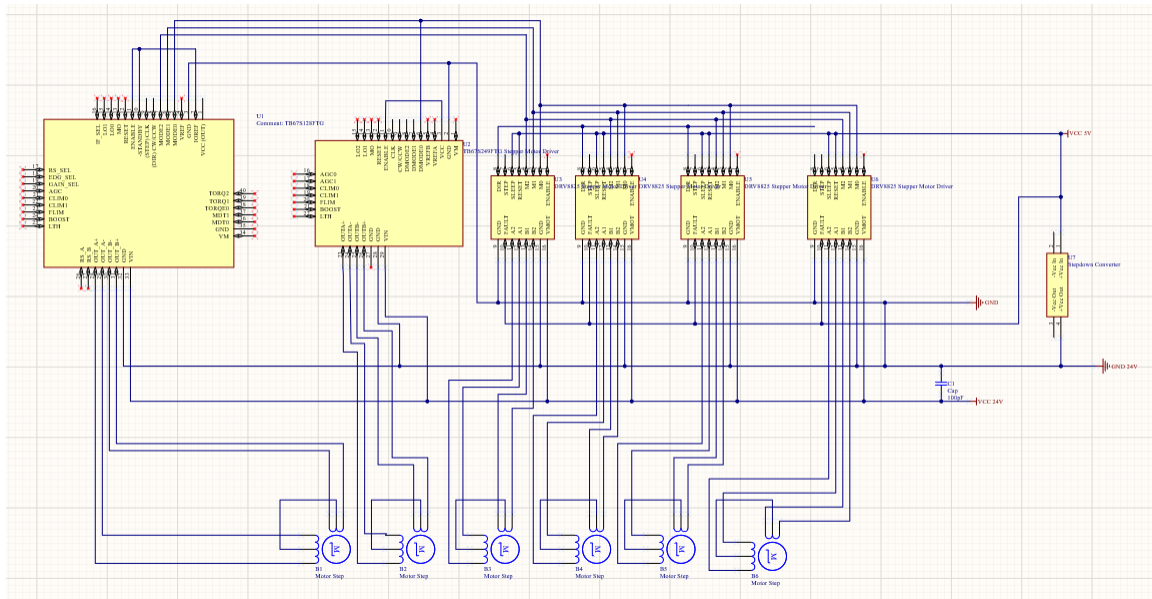


Figura 69: Esquemático base del brazo

9.3.3. Placa de circuito impreso (PCB)

Se diseñaron dos PCB para poder tener las conexiones dentro del control de mandos y de la base del robot. Estos fueron derivados de los esquemáticos específicos.

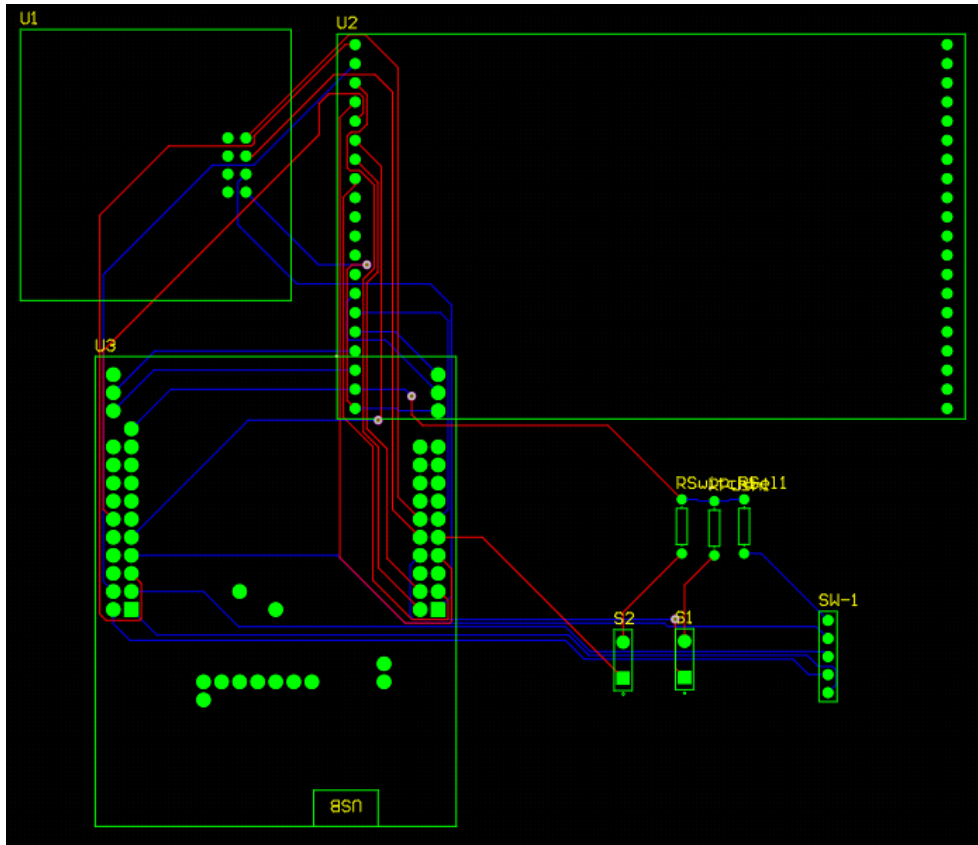


Figura 70: PCB control de mandos

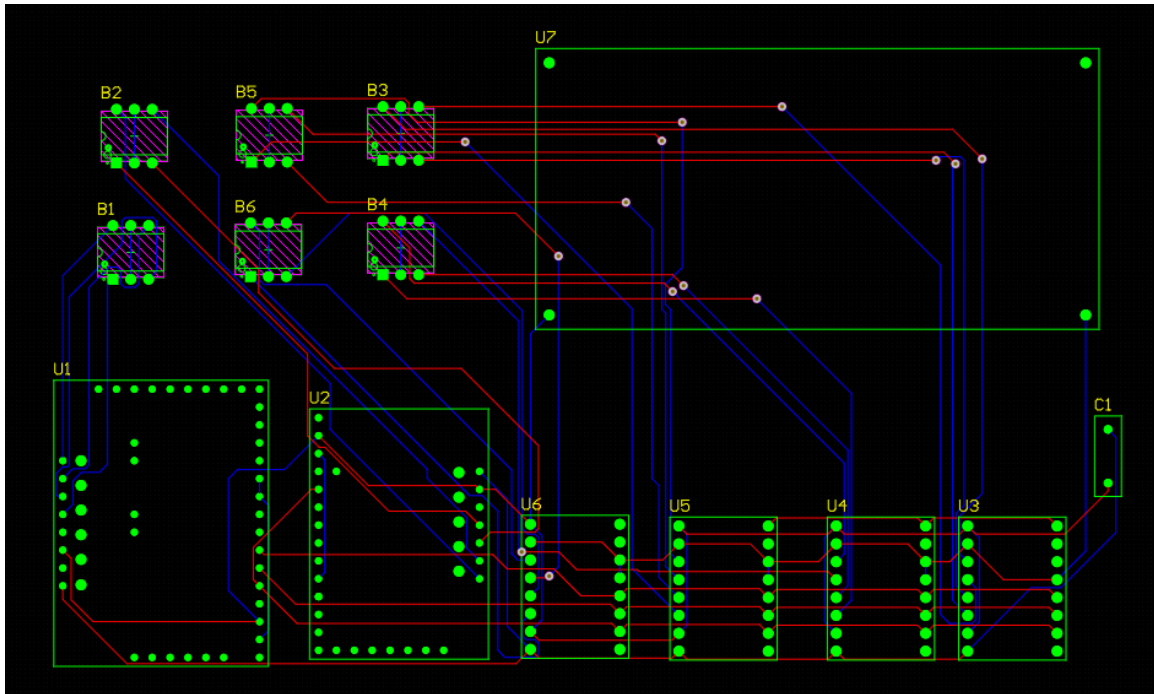


Figura 71: PCB base del brazo

9.4. Tercer prototipo

Para diseñar el tercer prototipo se verificaron varios cambios necesarios tanto al diseño de la base del robot como a los diseños de los esquemáticos y los PCBs.

9.4.1. Base para el robot

En la base del robot se cambiaron las entradas de los cables de las señales y se incorporó la entrada adecuada para el montaje de la entrada de potencia. Además, se realizó un cambio en el ángulo de las partes circulares de la caja para poder realizar el doblado con el MDF.

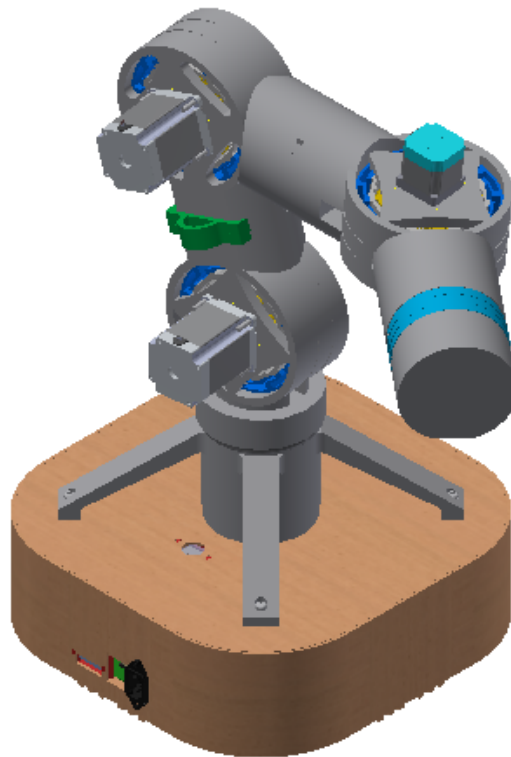


Figura 72: Base del robot vista isométrica

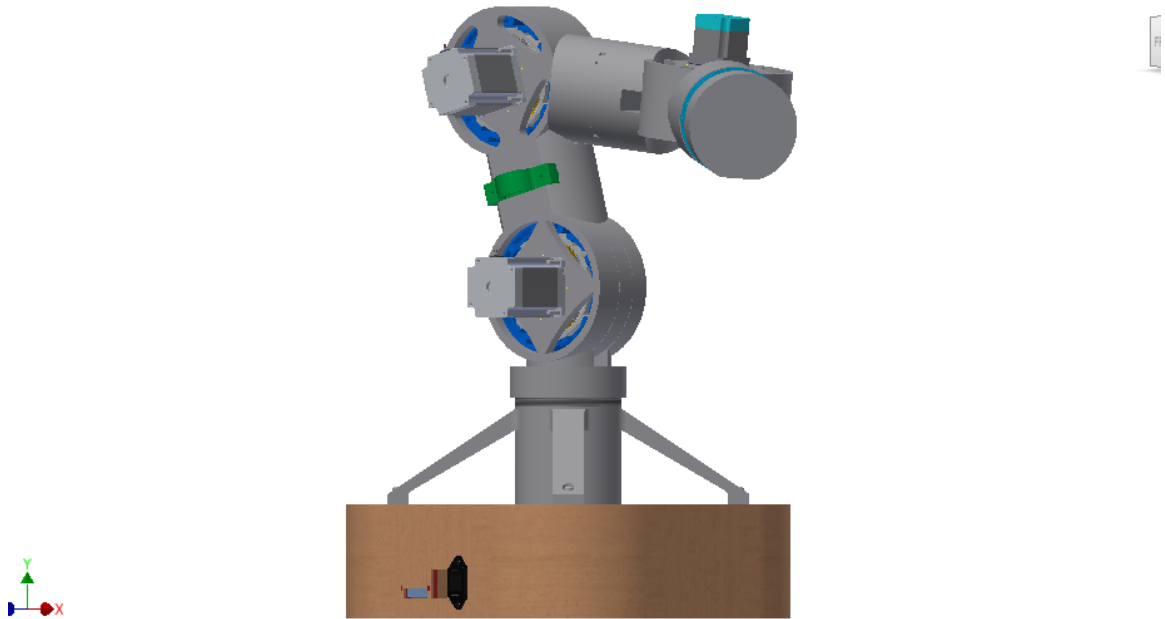


Figura 73: Base del robot vista frontal



Figura 74: Base del robot vista superior

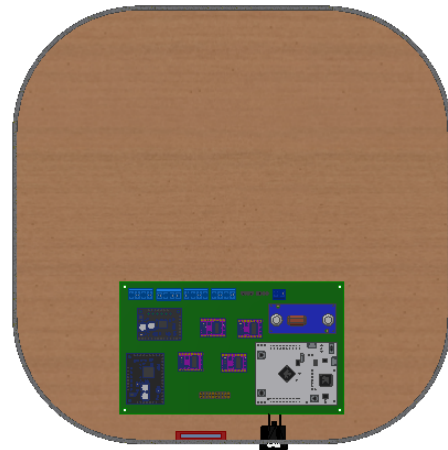


Figura 75: Base del robot vista del PCB

9.4.2. Esquemático

Para los esquemáticos se consideró la posición de los componentes en los distintos PCBs, tomando como cambio que la Tiva C se incorporó al PCB de la base del robot, disminuyendo la cantidad de señales que se deben transportar entre ambos PCBs.

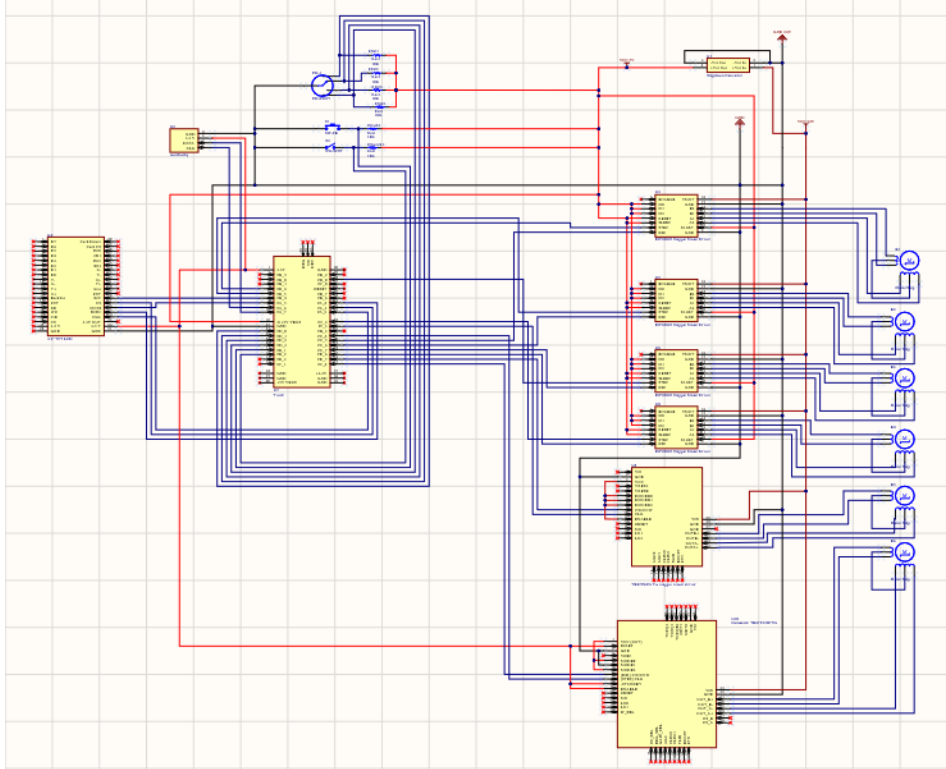


Figura 76: Esquemático completo

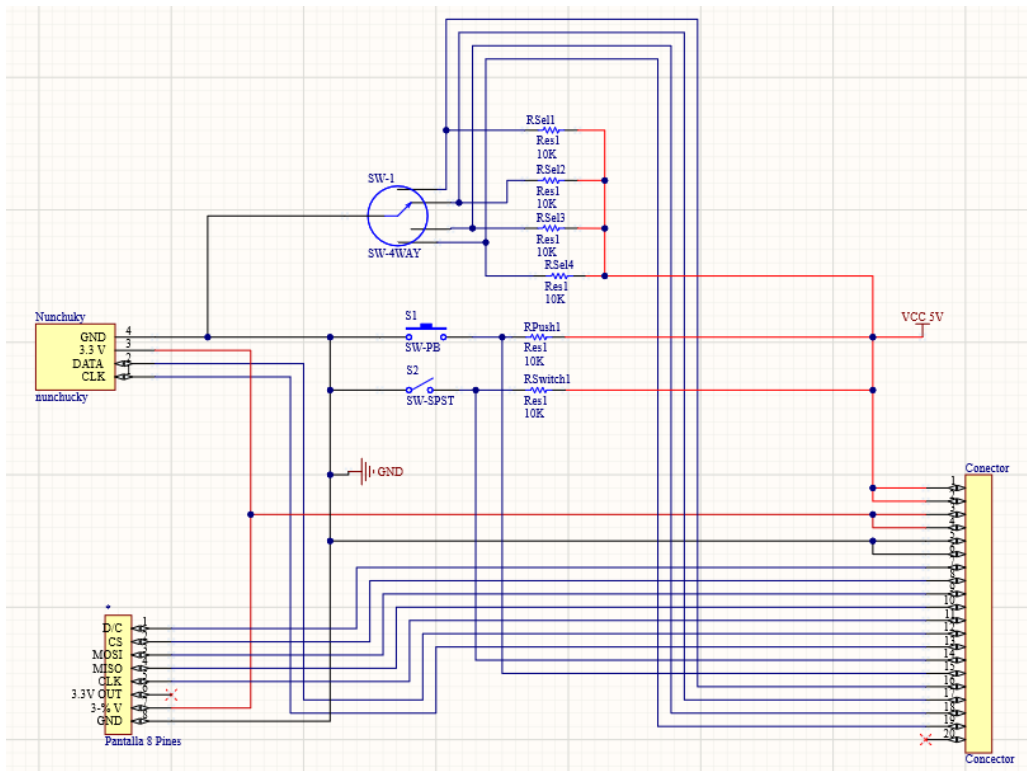


Figura 77: Esquemático control de mandos

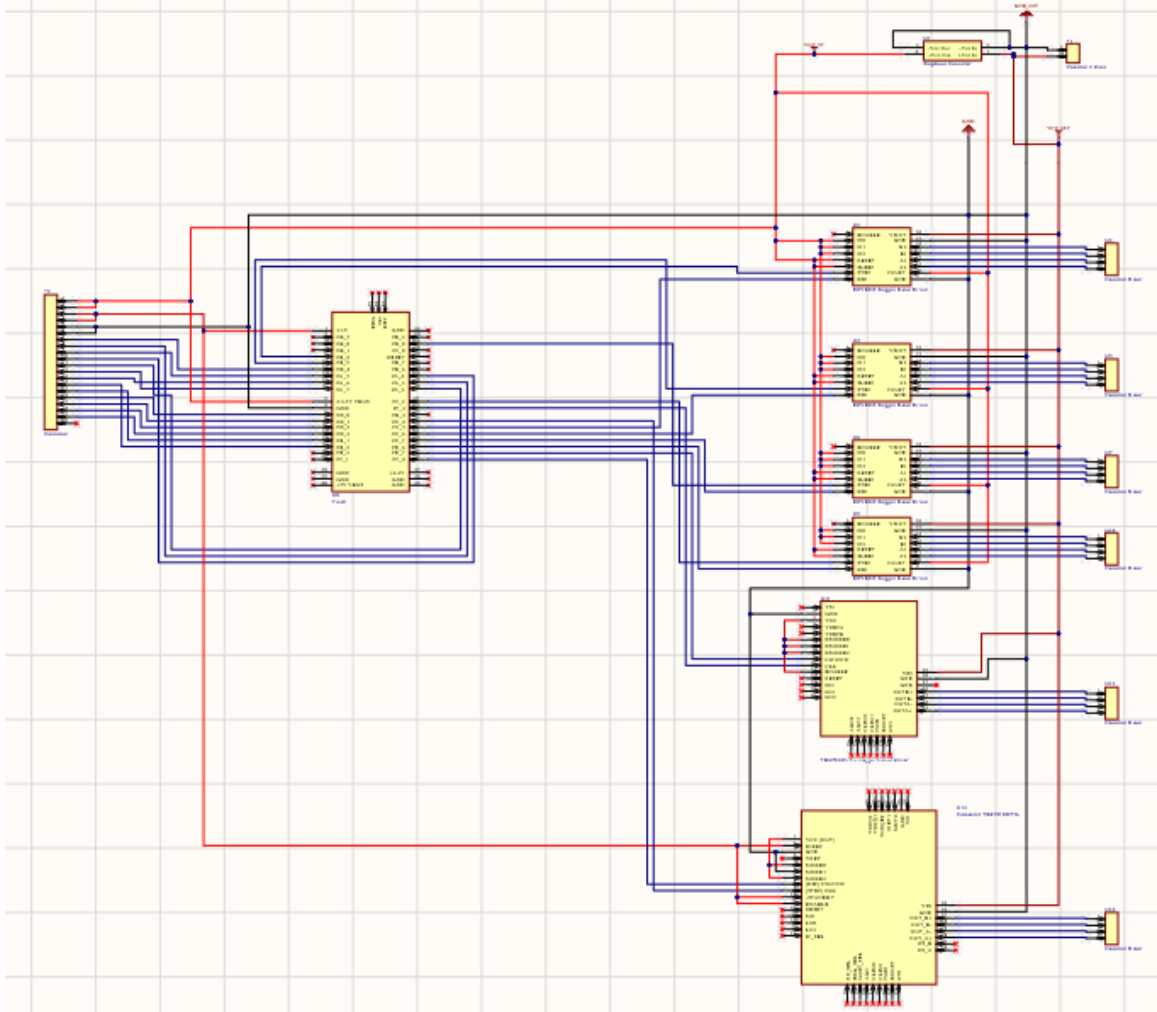


Figura 78: Esquemático base del robot

9.4.3. Placa de circuito impreso (PCB)

Para los PCBs, se realizaron los cambios necesarios para poder realizar la fabricación de los mismos, siendo estos la posición de los componentes, el tamaño de las vías, las pistas y la capa de las conexiones, entre otros. También se realizaron los diseños de los 3Ds para poder incorporarlos a los diseños realizados en Inventor.

Para verificar el grosor de las vías y los espacios entre vías, se utilizó una calculadora en línea en la página web: <https://www.desmith.net/NMdS/Electronics/TraceWidth.html>, en la cual se deben ingresar los datos de nuestro circuito como la corriente requerida, el voltaje y el grosor de la capa de cobre. Luego se obtiene automáticamente el calor de las vías necesario, siendo este 42 mils. Este resultado se hizo para el motor con mayor demanda de corriente y el resultado se puede observar en la Figura 79.

ANSI PCB TRACE WIDTH CALCULATOR							
Input Data			Results Data				
Field	Value	Units	Trace Data	Value	Units	Value	Units
Current (max. 35A)	4	Amps	Required Trace Width	106.75	mil	41.03	mil
Temperature Rise (max. 100°C)	10	°C	Cross-section Area	286.89	mil ²	110.28	mil ²
Cu thickness	2	oz/ft ²	Resistance	0.01	Ω Ohms	0.03	Ω Ohms
Ambient Temperature	25	°C	Voltage Drop	0.04	Volts	0.1	Volts
Conductor Length	10	cm	Loss	0.16	Watts	0.41	Watts
Peak Voltage	12	Volts	Required Track Clearance	25.4	mil		

Figura 79: Cálculos del grosor de vías requeridos

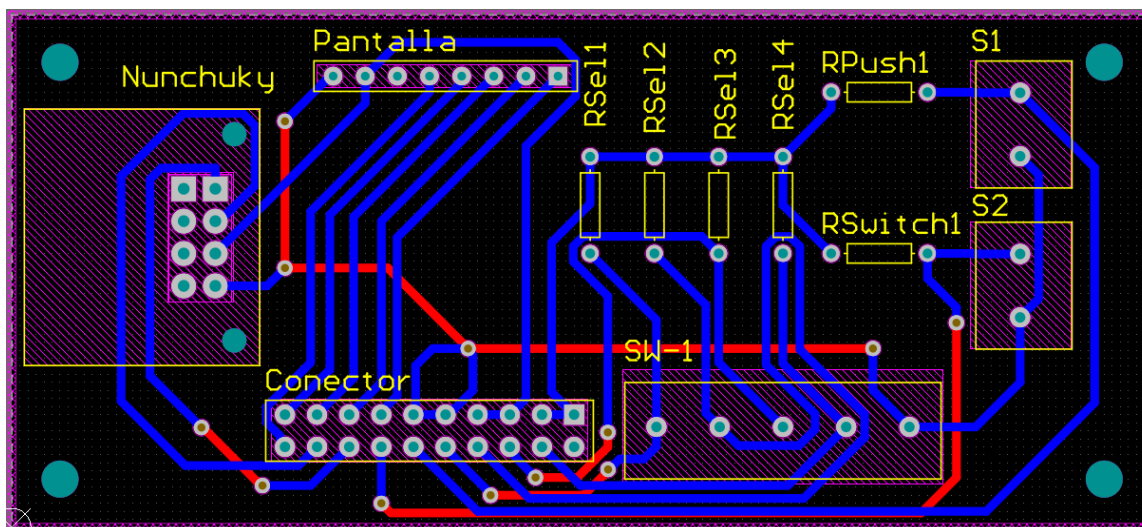


Figura 80: PCB Control de mandos 2D

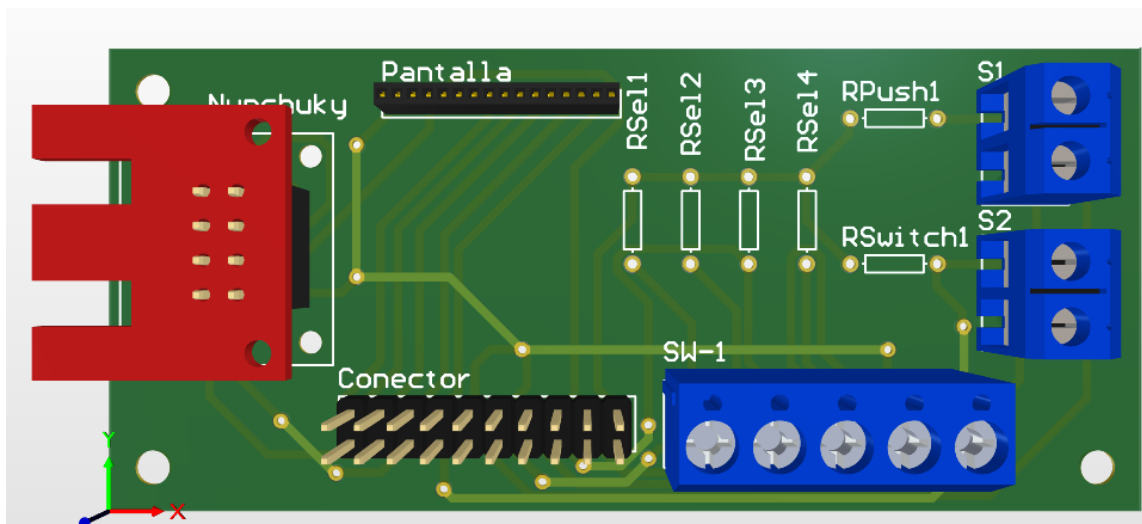


Figura 81: PCB Control de mandos vista superior 3D

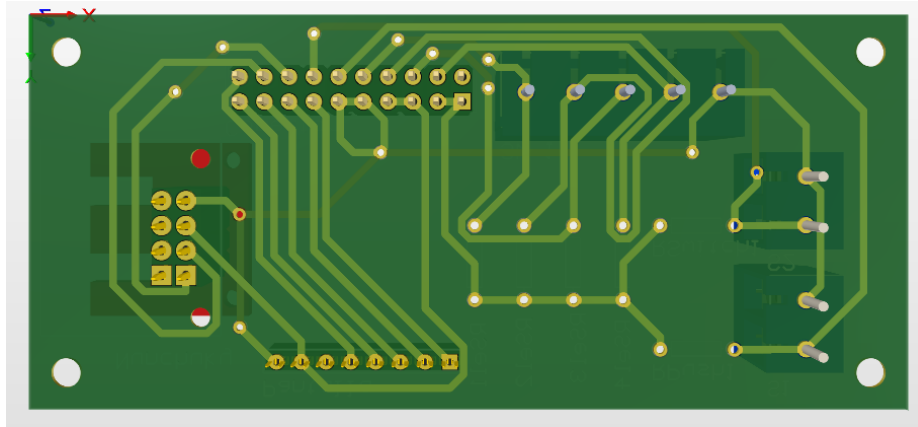


Figura 82: PCB Control de mandos vista inferior 3D

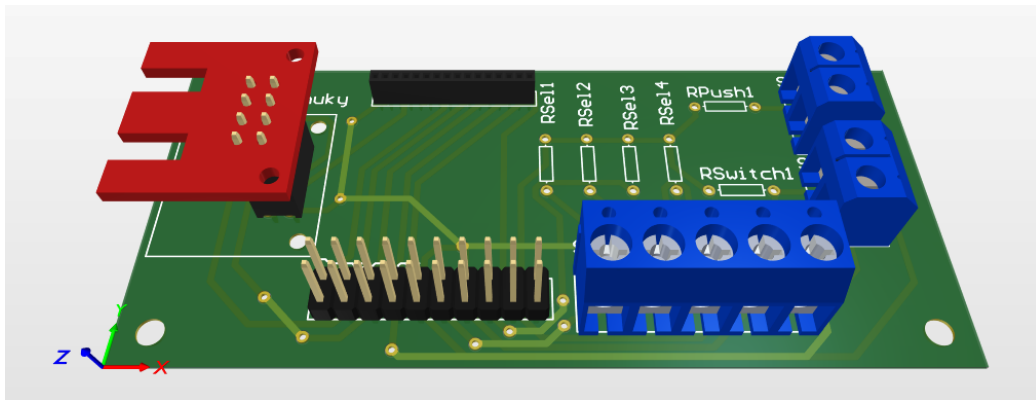


Figura 83: PCB Control de mandos vista isométrica 3D

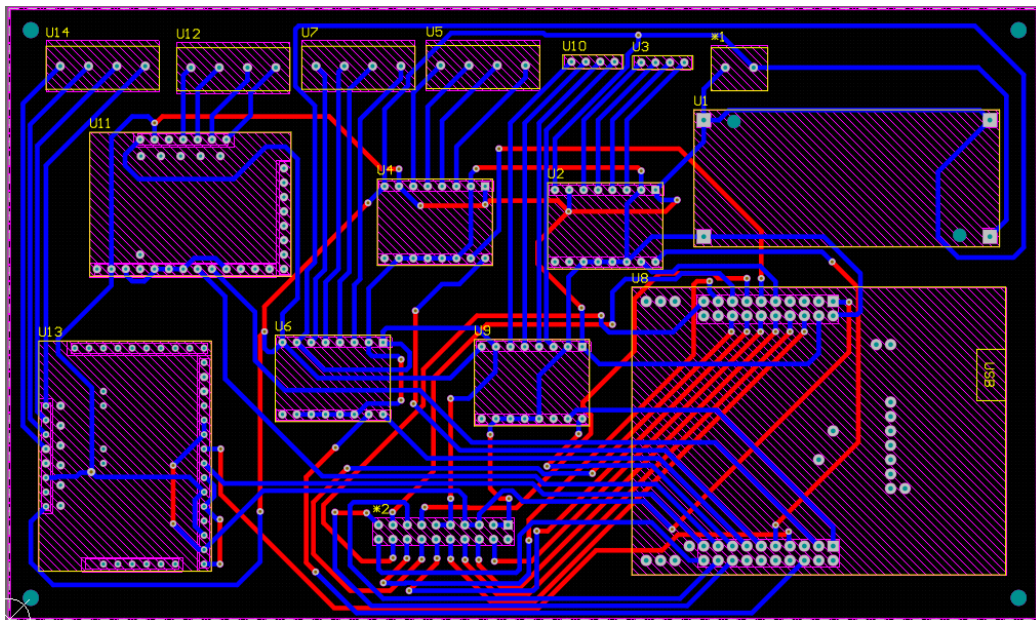


Figura 84: PCB Base del robot 2D

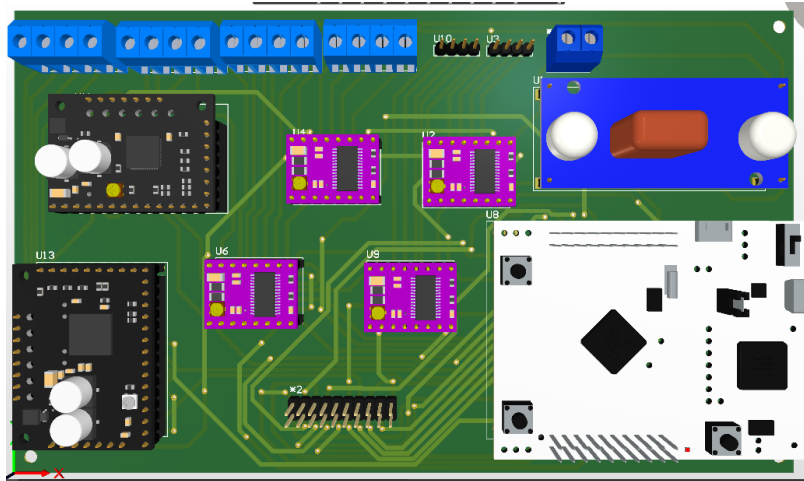


Figura 85: PCB Base del robot vista superior 3D

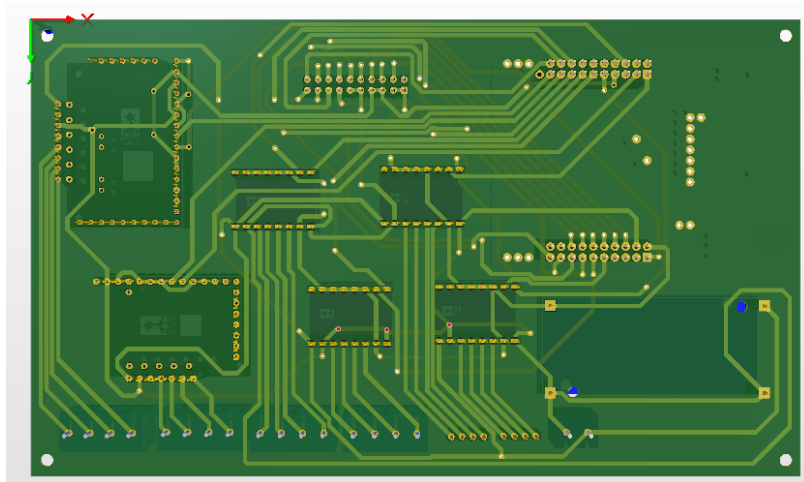


Figura 86: PCB Base del robot vista inferior 3D

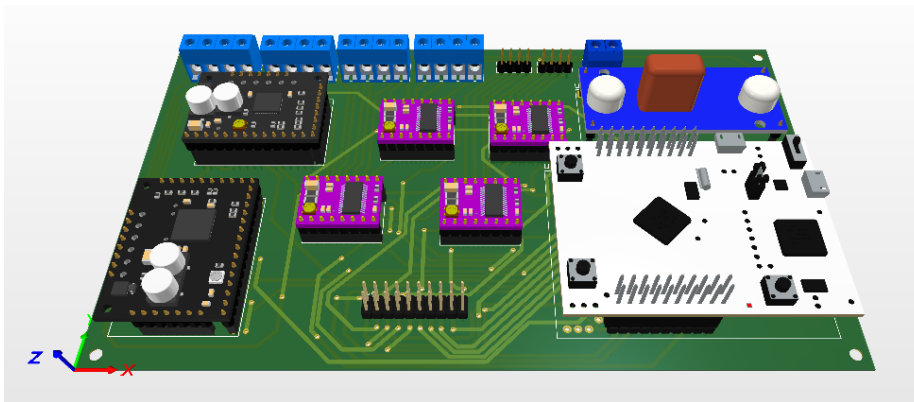


Figura 87: PCB Base del robot vista isométrica 3D

9.5. Cuarto prototipo (prototipo físico)

Para le cuarto prototipo se realizó la fabricación de ambas placas de circuito impreso y la base del robot con ayuda de corte láser de MDF e impresión 3D de PLA.

9.5.1. Base del robot

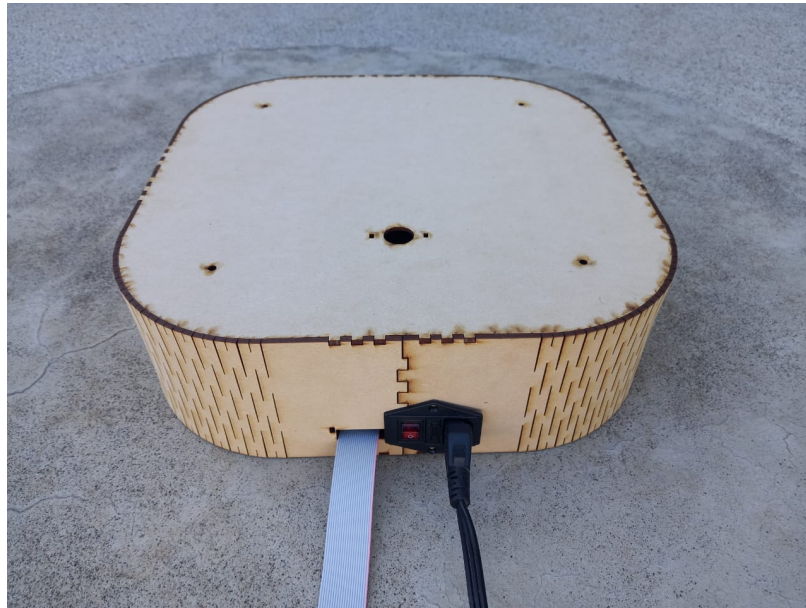


Figura 88: Base del robot vista isométrica

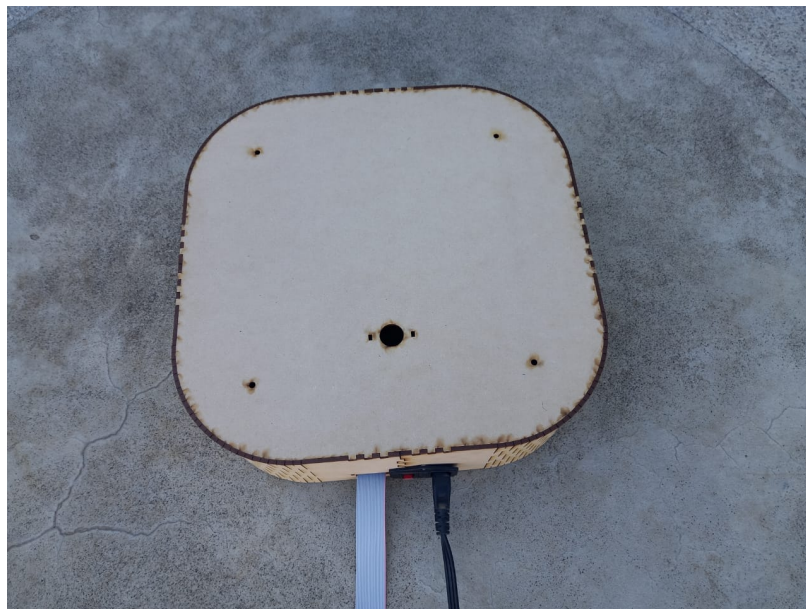


Figura 89: Base del robot vista superior

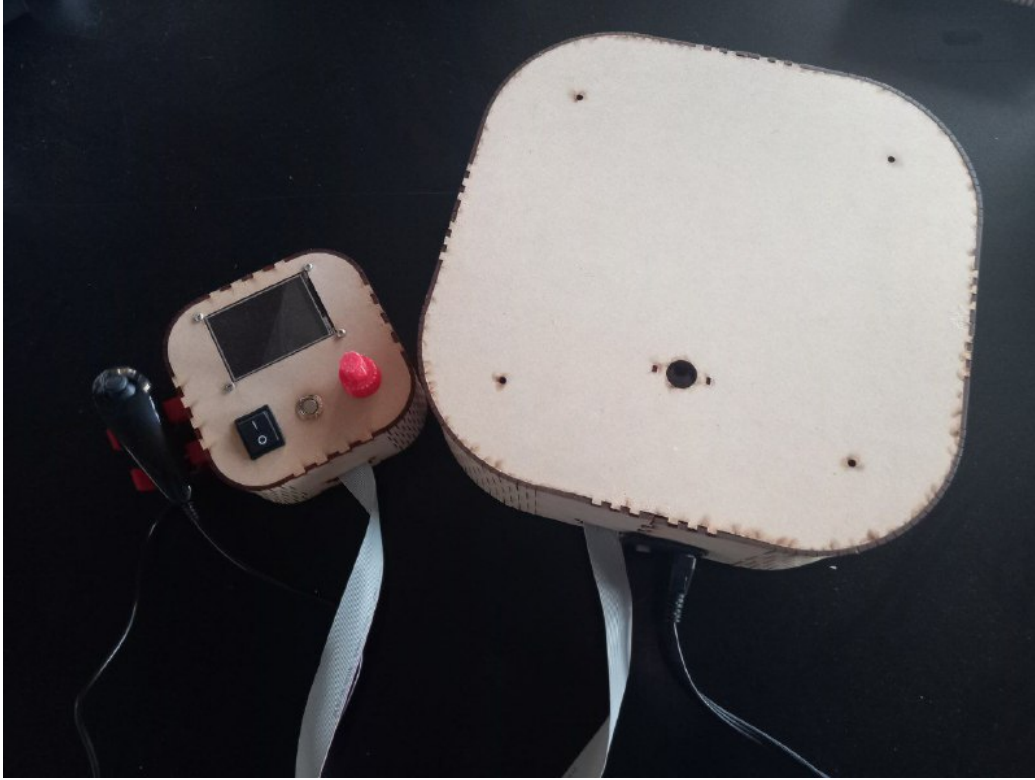


Figura 90: Conjunto base del robot y control de mandos

9.5.2. Placa del circuito impreso: Control de mandos

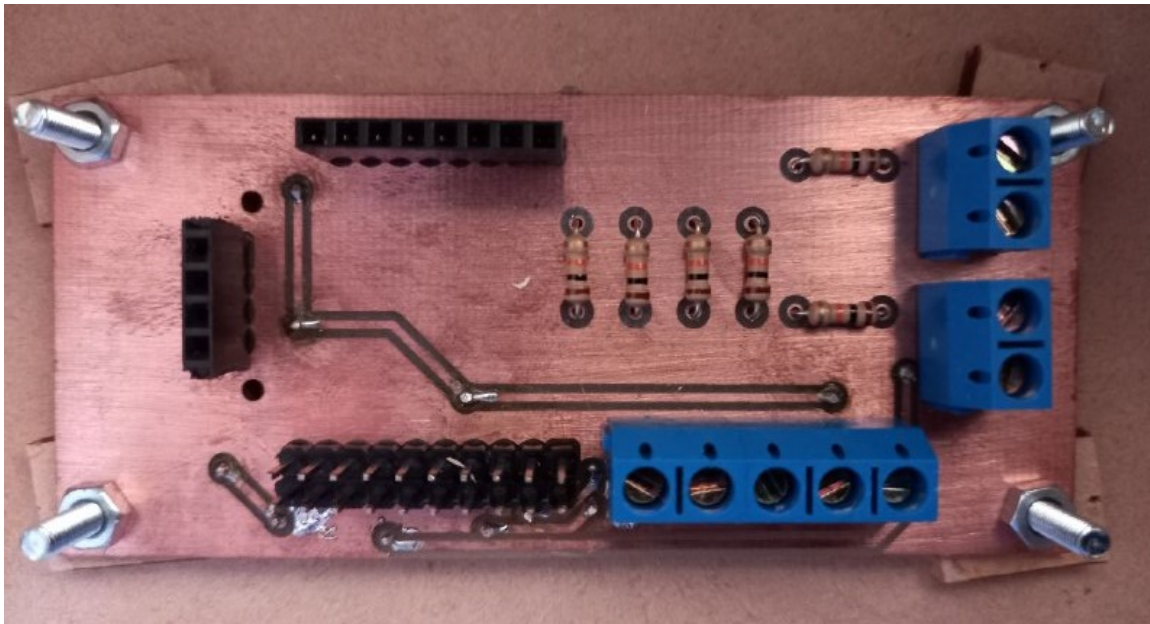


Figura 91: Placa del circuito impreso del control de mandos

9.5.3. Placa del circuito impreso: Base del robot

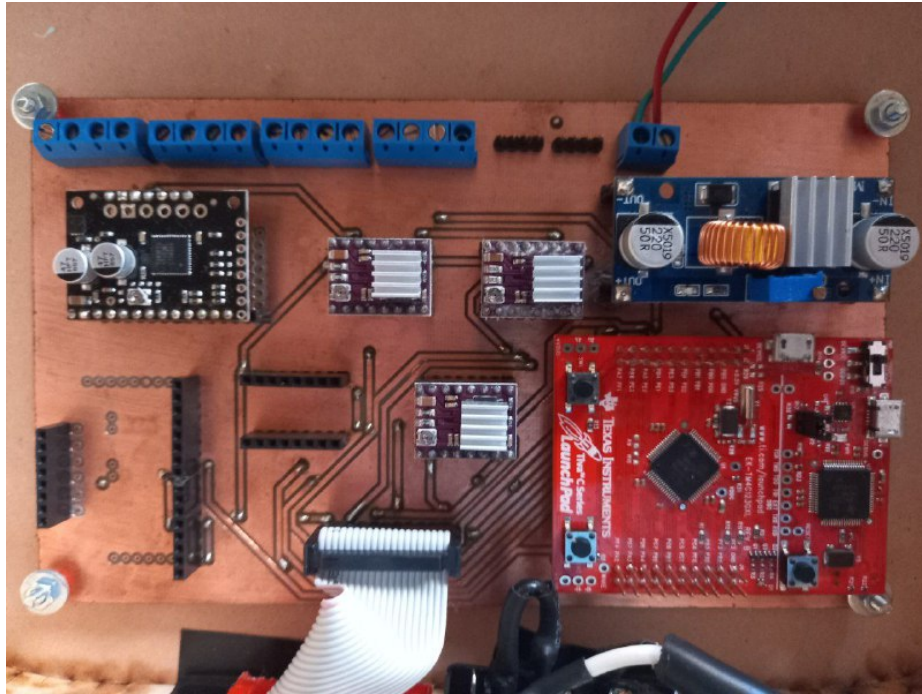


Figura 92: Placa del circuito impreso de la base del robot

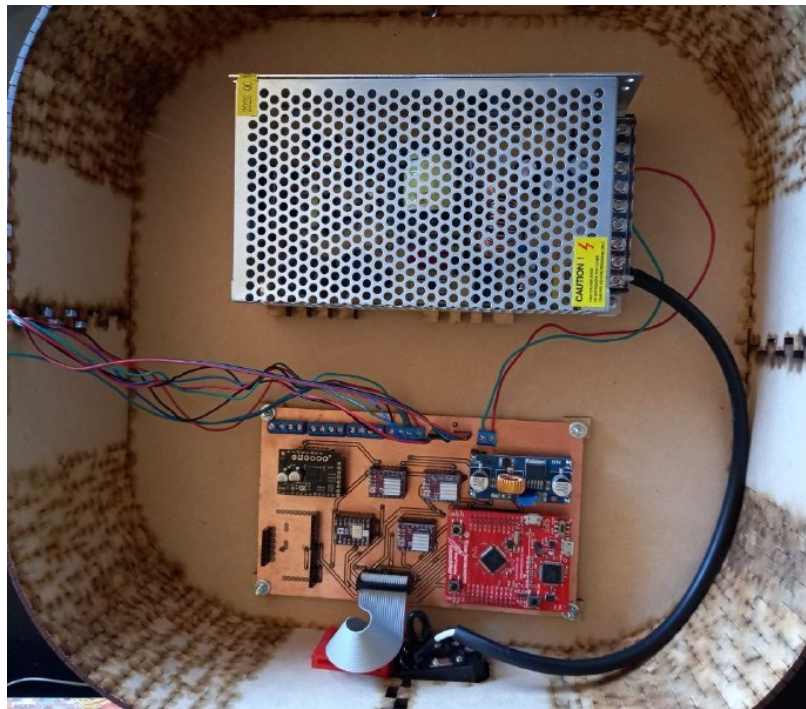


Figura 93: Placa del circuito impreso de la base del robot con fuente de voltaje

- Se utilizó cinemática inversa para generar el movimiento de las primeras tres juntas del brazo.
- Se diseñó un sistema de control para el brazo robótico que implementa un sistema de PIDs para controlar la velocidad y la posición de cada motor.
- Se generó una simulación en Webots con el modelo 3D del robot para observar y verificar el movimiento y sistema de control.
- Se diseñó y fabricó un control de mandos funcional.
- Se diseñaron y fabricaron circuitos impresos para el mando de control y la base del robot.
- Se realizaron los esquemáticos del sistema eléctrico y electrónico
- Se creó un manual de uso para el control de mandos.

- Se debe continuar con el uso de motores Stepper, debido a su estabilidad y par de fuerza.
- Es recomendable continuar con el uso de la cinemática inversa para el movimiento de las primeras juntas para obtener un posicionamiento aproximado al deseado.
- Utilizar algún sistema de realimentación (encoders) para obtener posición de los motores.
- Verificar el uso de drivers con control de temperatura para tener un sistema de paro de respaldo a la hora de alguna falla.
- Cambio a un material que proporcione más rigidez y sea fácil de desinfectar debido a que es para uso de cirugías.
- Tener presente la normativa eléctrica para la seguridad eléctrica para el equipo médico (IEC60601).

-
- [1] autor desconocido, 2020. dirección: <https://www.stanfordchildrens.org/es/service/neurosurgery/rosa>.
 - [2] autor desconocido, *ROSA ONE® Brain*, 2020. dirección: <https://www.zimmerbiomet.com/medical-professionals/cmf/rosa-brain.html>.
 - [3] —, *neuromate® - Robot de cirugía posicionamiento de electrodos by Renishaw: MedicalExpo*, 2020. dirección: <https://www.medicaexpo.es/prod/renishaw/product-73826-475862.html>.
 - [4] Renishawplc, *Robot estereotáctico neuromate®*, 2020. dirección: <https://www.renishaw.es/es/sistema-robotizado-neuromate-para-neurocirugia-estereotactica--10712>.
 - [5] D. F. Salazar, “Diseño mecánico y modelado matemático para brazo robótico asistencial en cirugía estereotáctica del cerebro.”, Tesis doct., 2019.
 - [6] J. J. Cifuentes, “Diseño e implementación de mando de control para brazo robótico asistencial en cirugía estereotáctica del cerebro”, Tesis doct., 2019.
 - [7] D. R. Valdez, “Control y Alimentación de Actuadores para Brazo Robótico Asistencial en Cirugía Estereotáctica del Cerebro”, Tesis doct., 2019.
 - [8] B. S. Chang, *Epilepsy: NEJM*, sep. de 2003. dirección: <https://www.nejm.org/doi/full/10.1056/NEJMra022308>.
 - [9] *Epilepsy*, jun. de 2019. dirección: <https://www.who.int/news-room/fact-sheets/detail/epilepsy>.
 - [10] D. Y. Ko, *Epilepsy and Seizures*, S. R. Benbadis, ed., abr. de 2020. dirección: <https://emedicine.medscape.com/article/1184846-overview>.
 - [11] a. desconocido autor, *HUMANA?*, 2020. dirección: <https://humanagt.org/>.
 - [12] autor desconocido, *Motor paso a paso ¿que es y como funciona?*, abr. de 2017. dirección: <https://www.ingmecafenix.com/electricidad-industrial/motor-paso-a-paso/>.

- [13] P. 3. -, *Motores a pasos... ¿unipolares o bipolares?*, feb. de 2016. dirección: <https://blog.330ohms.com/2016/02/09/motores-a-pasos-unipolares-o-bipolares/>.
- [14] K. Ogata, *Ingeniería de control moderna*, quinta. Prentice Hall, 1982.
- [15] G. F. Franklin, J. D. Powell y M. L. Workman, *Digital Control of Dynamic Systems*, Third Edition. Addison Wesley Longman, Inc., 1998.
- [16] L. F. Giraldo, E. Delgado y G. Castellanos, “Cinemática inversa de un brazo robot utilizando algoritmos genéticos”, *Revista Avances en Sistemas e Informática*, vol. 3, n.º 1, págs. 29-34, 2006.
- [17] F. M. Alonzo y M. E. Bravo, “Diseño, construcción y control de un brazo robótico”, Tesis doct., 2014.
- [18] *TM4C123G, LAUNCHPAD, TIVA C, EVAL KIT Part TEXAS INSTRUMENTS EK-TM4C123GXL*. LogicWare. dirección: <http://www.logicware.com.au/tm4c123g-launchpad-tiva-c-eval-kit-part-texas-inst>.
- [19] B. R. Santos Avila, “Diseño Mecánico y Fabricación de un Brazo Robótico de 6 Grados de Libertad Asistencial en Cirugía Estereotáctica del Cerebro”, Tesis doct., 2020.
- [20] *Nema 23 Bipolar 425oz.in 4.2A 57x57x114mm 4 Wires Stepper Motor*. Robotshop. dirección: <https://www.robotshop.com/en/nema-23-bipolar-425ozin-42a-57x57x114mm-4-wires-stepper-motor.html>.
- [21] *57BYG Stepper Motor*. Robotshop. dirección: <https://www.robotshop.com/en/12v-17a-667oz-in-nema-17-bipolar-stepper-motor.html>.
- [22] *12V, 1.7A, 667oz-in NEMA-17 Bipolar Stepper Motor*. Robotshop. dirección: <https://www.robotshop.com/en/12v-17a-667oz-in-nema-17-bipolar-stepper-motor.html>.
- [23] *Hybrid Stepper Motor for 3D Printer*. Robotshop. dirección: <https://www.robotshop.com/en/hybrid-stepper-motor-3d-printer.html>.
- [24] *DRV8825 stepper motor driver carrier*. Pololu. dirección: <https://www.pololu.com/product/2133/pictures>.
- [25] *TB67S249FTG Stepper Motor Driver Carrier (top view)*. Pololu. dirección: <https://www.pololu.com/product/2973/pictures>.
- [26] *TB67S128FTG Stepper Motor Driver Carrier (top view)*. Pololu. dirección: <https://www.pololu.com/product/2998/pictures>.
- [27] *Nunchuk - REFURBISHED (Wii U, Wii, Wii mini)*. nintendo. dirección: <https://store.nintendo.com/nunchuk-refurbished-wii-u-wii-wii-mini.html>.
- [28] *BEIARA Wii Nunchuck Controllers Reemplazo para Nintendo WII U Motion Controllers Video Game - 2Packs (Blanco)*. Amazon. dirección: <https://www.amazon.com/BEIARA-Nunchuck-Controllers-Replacement-Nintendo/dp/B08819YZW9>.
- [29] *3.2" TFT LCD with Touchscreen Breakout Board w/MicroSD Socket - ILI9341*. Adafruit. dirección: <https://www.adafruit.com/product/1743>.
- [30] *3.2" TFT LCD with Touchscreen Breakout Board w/MicroSD Socket - ILI9341*. Adafruit. dirección: <https://www.adafruit.com/product/1743>.
- [31] *R5ABLKBLKFF0*. Digikey. dirección: <https://www.digikey.com/product-detail/es/e-switch/R5ABLKBLKFF0/EG1529-ND/301976>.

- [32] *KC14A30.001NPS*. Digikey. dirección: <https://www.digikey.com/product-detail/es/e-switch/KC14A30-001NPS/KC14A30-001NPS-ND/1804514>.
- [33] *16mm Illuminated Pushbutton - Red Latching On/Off Switch*. Adafruit. dirección: <https://www.adafruit.com/product/1442>.
- [34] *inShareplus Fuente de alimentación de conmutación LED de 12 V, Plateado*. Amazon. dirección: https://www.amazon.com/dp/B07RJMG3FB/ref=sspa_dk_detail_1?pd_rd_i=B07RFGH2ZM&pd_rd_w=xrY0g&pf_rd_p=48d372c1-f7e1-4b8b-9d02-4bd86f5158c5&pd_rd_wg=Xghg7&pf_rd_r=3HK8W3MRTXVGWKR1TG9B&pd_rd_r=4d59fb08-935f-4b86-98c3-e91ed29321a1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzSE&th=1.
- [35] *inShareplus Fuente de alimentación de conmutación LED de 12 V, Plateado*. Amazon. dirección: https://www.amazon.com/dp/B07RJMG3FB/ref=sspa_dk_detail_1?pd_rd_i=B07RFGH2ZM&pd_rd_w=xrY0g&pf_rd_p=48d372c1-f7e1-4b8b-9d02-4bd86f5158c5&pd_rd_wg=Xghg7&pf_rd_r=3HK8W3MRTXVGWKR1TG9B&pd_rd_r=4d59fb08-935f-4b86-98c3-e91ed29321a1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzSE&th=1.
- [36] *inShareplus Fuente de alimentación de conmutación LED de 12 V, Plateado*. Amazon. dirección: https://www.amazon.com/dp/B07RJMG3FB/ref=sspa_dk_detail_1?pd_rd_i=B07RFGH2ZM&pd_rd_w=xrY0g&pf_rd_p=48d372c1-f7e1-4b8b-9d02-4bd86f5158c5&pd_rd_wg=Xghg7&pf_rd_r=3HK8W3MRTXVGWKR1TG9B&pd_rd_r=4d59fb08-935f-4b86-98c3-e91ed29321a1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzSE&th=1.
- [37] *D-PLANET [paquete de 4] Convertidor de Buck ajustable de 5 A CC-CC de 4 38 V a 1,25 - 36 V, fuente de alimentación de baja potencia, módulo regulador de voltaje de alta eficiencia*. Amazon. dirección: https://www.amazon.com/Adjustable-Converter-1-25-36v-Efficiency-Regulator/dp/B079N9BFZC/ref=sr_1_7?crd=5BB4415QBQNV&dchild=1&keywords=12v+to+5v+step+down+converter&qid=1599887520&srefix=12v+to+5v+s%5C%2Caps%5C%2C245&sr=8-7.
- [38] *Micro conectores, Inc. 6 pies 2 dientes cable de alimentación para ordenador portátil aprobado por UL (M05-125) 6 ft*. Amazon. dirección: https://www.amazon.com/Micro-Connectors-Approved-18AWG-Power/dp/B07H3QN7H3/ref=sr_1_13?dchild=1&keywords=iec320%5C%2Bc14%5C%2Bpanel%5C%2Bmount%5C%2Bmale%5C%2Bband%5C%2Bfemale&qid=1599889019&sr=8-13&th=1.
- [39] *Antrader IEC320 C14 - Enchufe con 3 clavijas macho (250 V CA, 10 A), color negro Negro*. Amazon. dirección: https://www.amazon.com/Antrader-Screw-Adapter-Connector-Socket/dp/B081VJ22G4/ref=sr_1_4?dchild=1&keywords=iec320+c14+panel+mount+male+and+female&qid=1599889019&sr=8-4.
- [40] *4-pin JST SM Plug + Receptacle Cable Set*. Adafruit. dirección: <https://www.adafruit.com/product/578>.
- [41] *VANDESAIL JST - Conector micro eléctrico macho y hembra (2 pines)*. Amazon. dirección: https://www.amazon.com/Connector-VANDESAIL-Female-Adapter-Electrical/dp/B076HLQ4FX/ref=sr_1_17?dchild=1&keywords=conexion+de+cable+40+pines&qid=1599888446&sr=8-17.

CAPÍTULO 13

Anexos

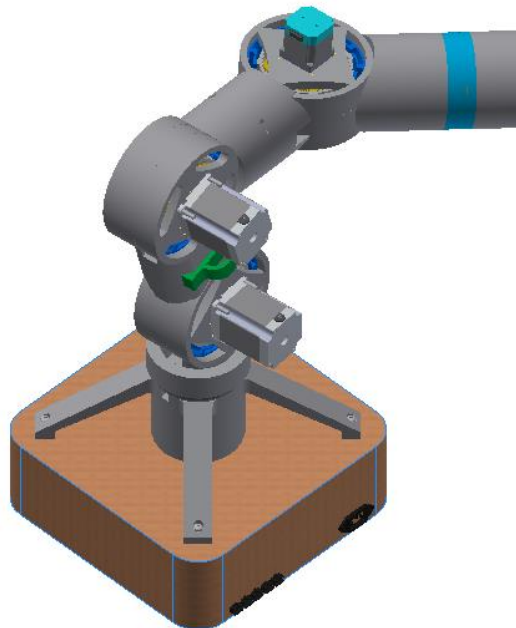
13.1. Manual de uso

13.2. Esquemáticos



MANUAL DE USO

Control para el brazo robótico



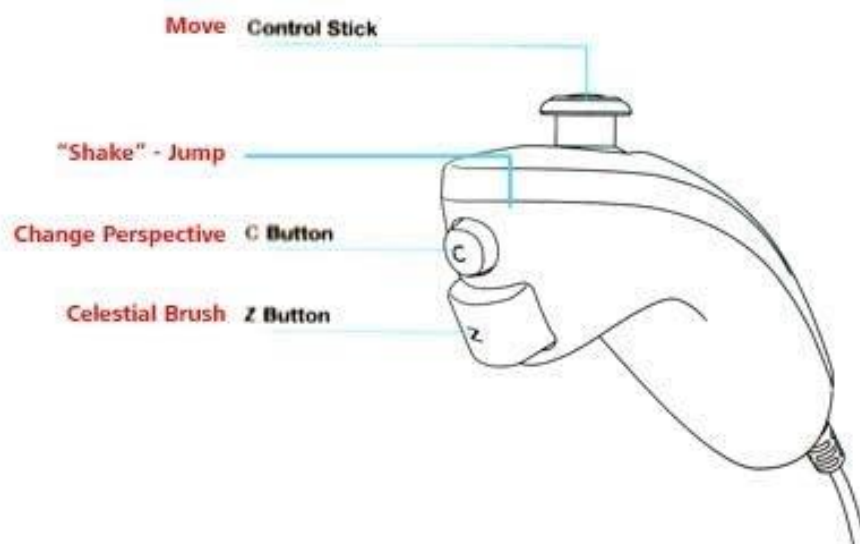
ENERO 20, 2021

UNIVERSIDAD DEL VALLE DE GUATEMALA

Dennet Orbaugh

Componentes

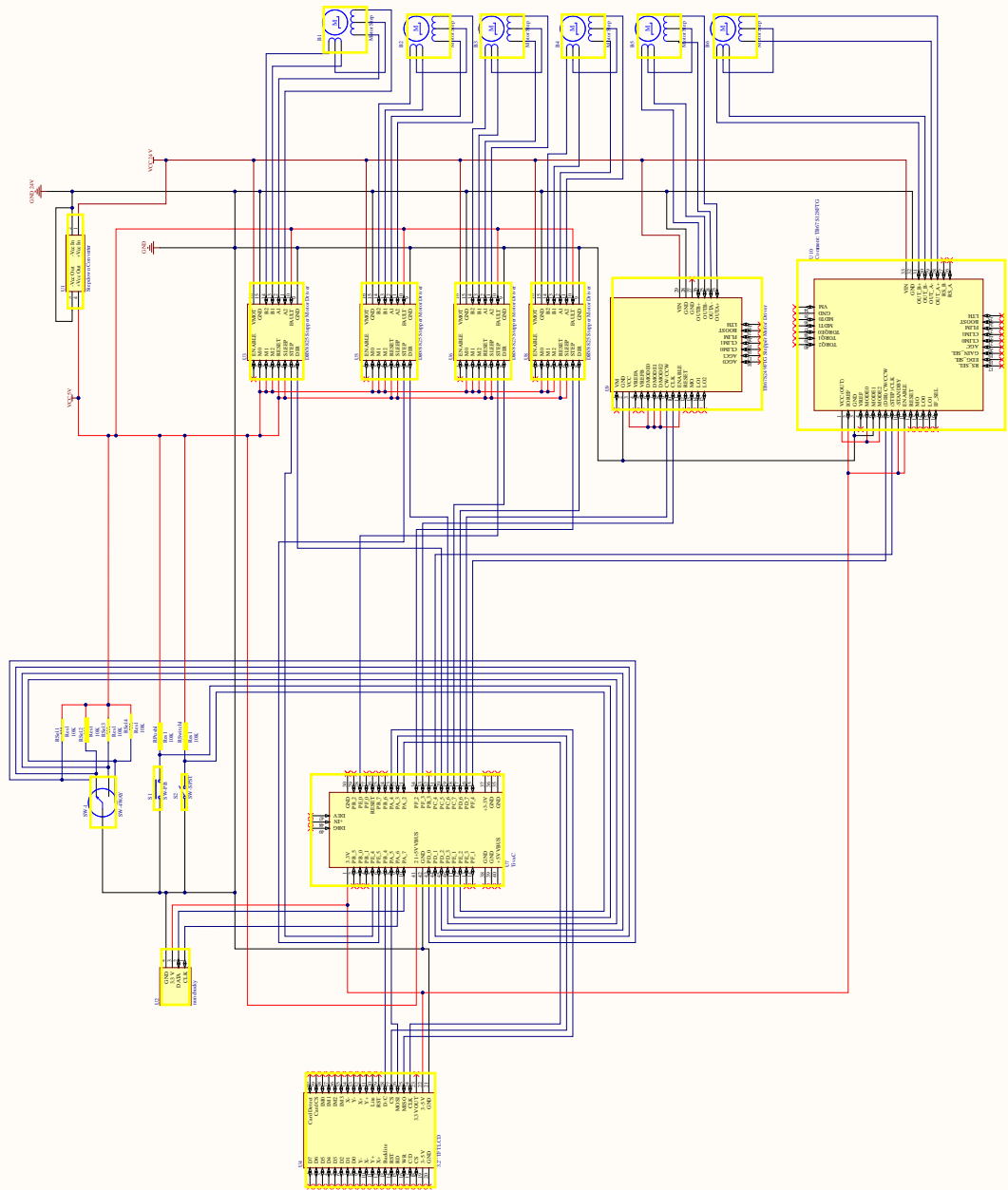
1. Selector rotativo
2. Interruptor SPST (Switch)
3. Botón de Push
4. Pantalla LCD
5. Control de Wii: Nunchuk

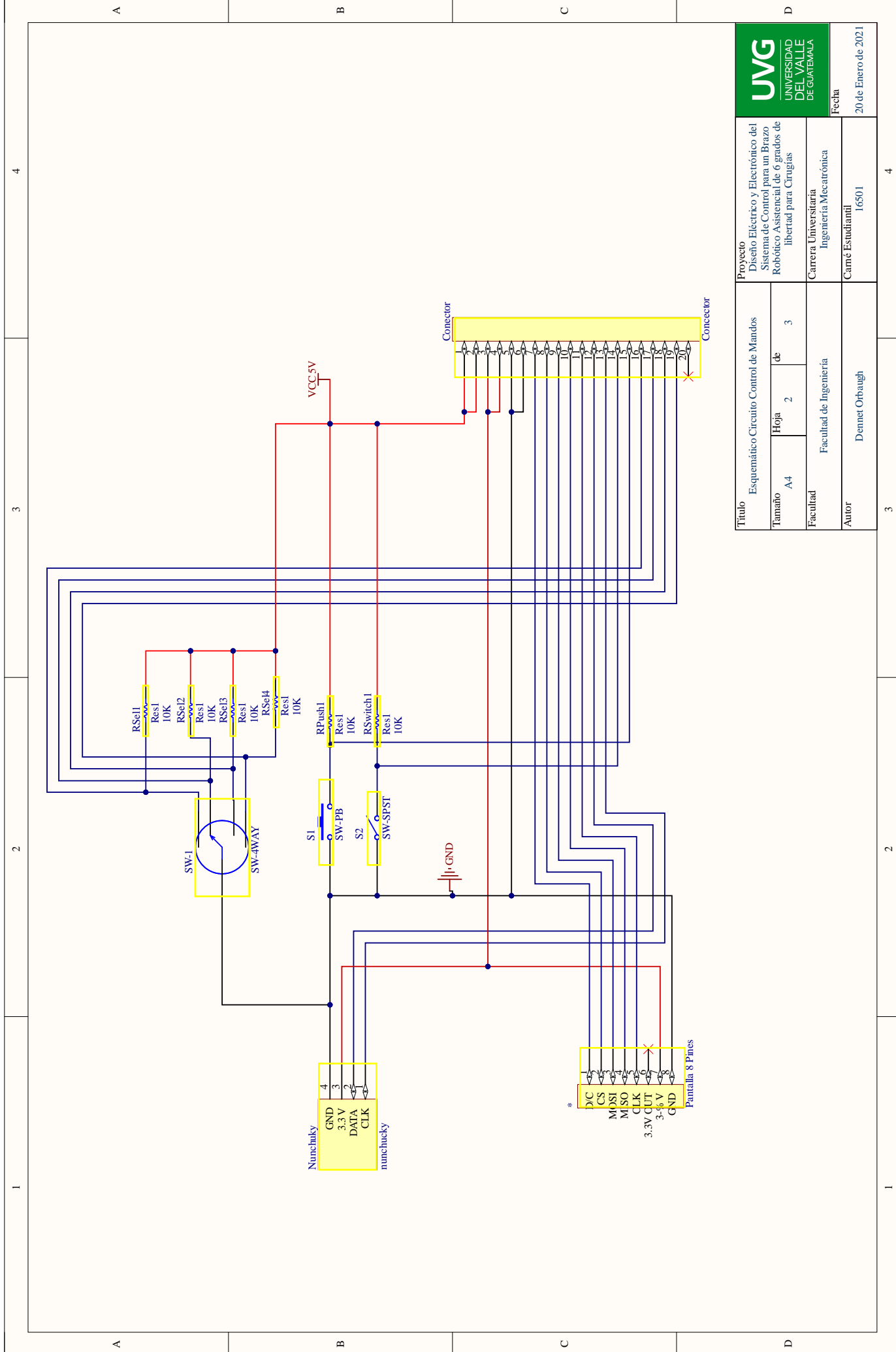


Modo de uso


1. Prender el control al accionar el interruptor SPST
2. Verificar que el botón de paro de emergencia este desactivado
3. Seleccionar la opción deseada en el selector
 - a. Movimiento de las juntas 1 a 3 **
 - b. Movimiento de la junta 4
 - c. Movimiento de la junta 5
 - d. Movimiento de la junta 6
4. Utilizar el joystick del Nunchuk para mover la junta seleccionada.
 - a. Mover el joystick en dirección del eje Y (de arriba hacia abajo) para controlar el movimiento de la junta seleccionada.
** En la primera opción del selector rotativo (juntas 1 a 3), utilizar el joystick para mover el efector final en el plano X-Y. Mantener el botón “c” presionado para mover el efector final en el eje Z al mover el joystick en dirección del eje Y.
5. Mantener el botón “z” del joystick para controlar la velocidad.
6. Al presionar el botón para el paro de emergencia, ninguna otra función del control esta activa, hasta que este se desactive.
7. En la pantalla se despliega:
 - a. la junta(s) a mover
 - b. la velocidad actual
 - c. estado del paro de emergencia



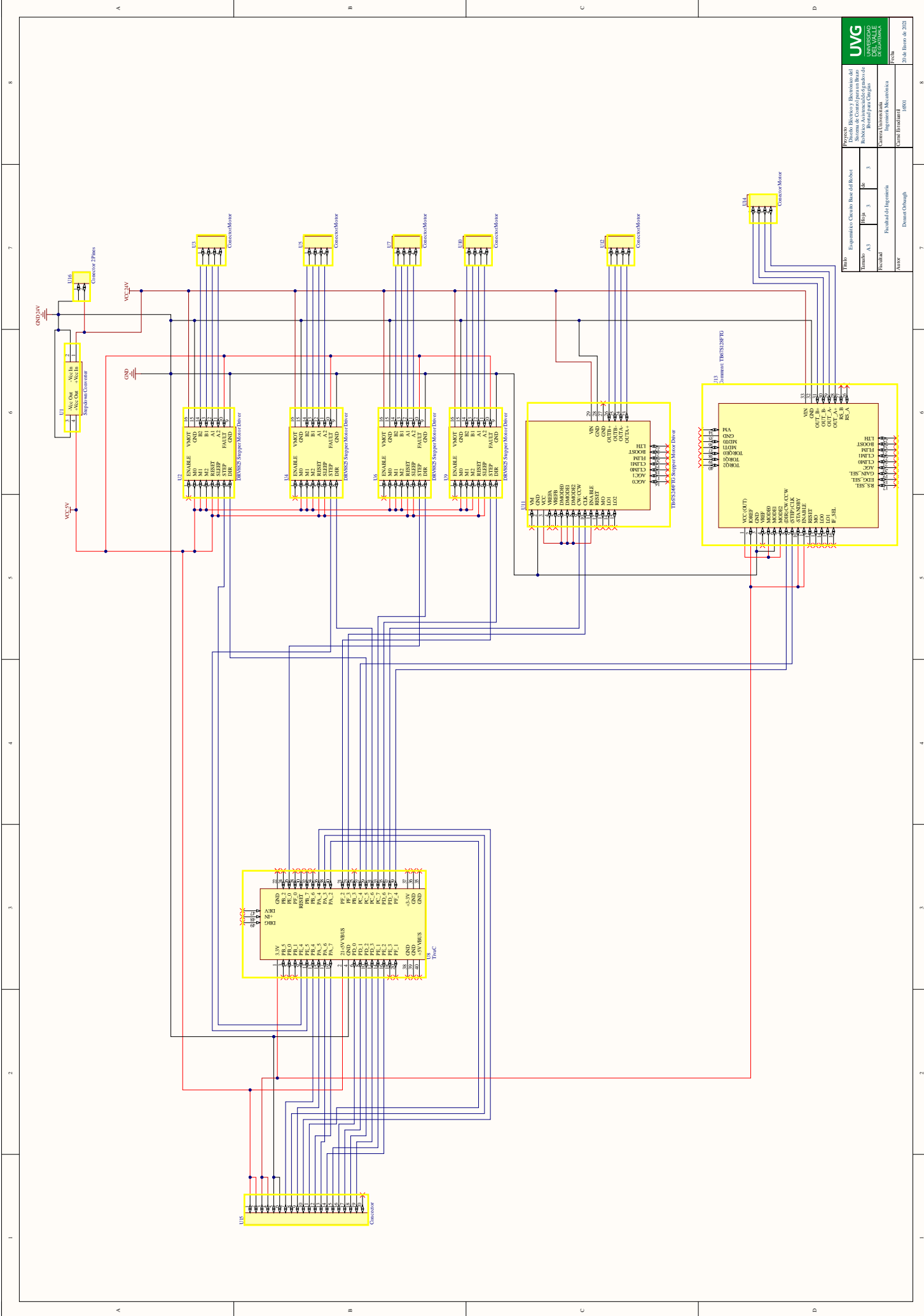




Título		Esquemático Circuito Control de Mandos	
Tamaño	A4	Hoja	2 de 3
Facultad	Facultad de Ingeniería		
Autor	Dennet Orbaugh		


UNIVERSIDAD DEL VALLE DE GUATEMALA
 Fecha: 20 de Enero de 2021

Proyecto: Diseño Eléctrico y Electrónico del Sistema de Control para un Brazo Robótico Asistencial de 6 grados de libertad para Cirugías
 Carrera: Ingeniería Mecatrónica
 Carné Estudiantil: 16501



UVG UNIVERSIDAD DE GUATEMALA	
Facultad de Ingeniería Ingeniería Mecatrónica	Nombre: [Blank] No. de Matrícula: [Blank]
Tema: Estructura Mecánica	Fecha: [Blank]
Título: Estructura Mecánica	Año: [Blank]
Autor: [Blank]	Fecha: [Blank]
Fecha: [Blank]	Año: [Blank]

Tema: Estructura Mecánica	Fecha: [Blank]
Título: Estructura Mecánica	Año: [Blank]
Autor: [Blank]	Fecha: [Blank]
Fecha: [Blank]	Año: [Blank]

