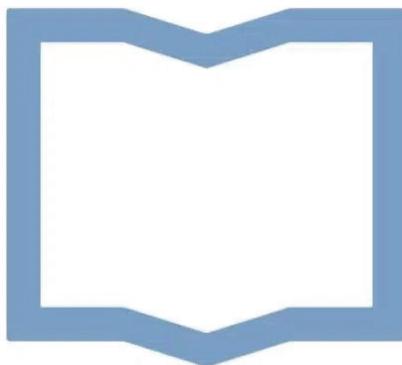


UNIVERSIDAD NACIONAL DE SAN MARTÍN



ESCUELA DE CIENCIA Y TECNOLOGÍA

**Automatización de laboratorio químico mediante brazo
robótico**

PROYECTO FINAL DE CARRERA DE INGENIERÍA EN ELECTRÓNICA

Autores:

Agüero Nicolás

Pedraza Sebastián

Tutor:

Dr. Carlos Rinaldi

Co-tutor:

Dr. Ing. Juan Vorobioff

PROYECTO FINAL DE CARRERA
INGENIERÍA EN ELECTRÓNICA

Automatización de laboratorio químico mediante brazo robótico

Autores: Agüero Nicolás, Pedraza Sebastián.

Tutor: Dr. Carlos Rinaldi

Co-tutor: Dr. Ing. Juan Vorobioff.

ESCUELA DE CIENCIA Y TECNOLOGÍA
UNIVERSIDAD NACIONAL DE SAN MARTÍN
BUENOS AIRES, ARGENTINA.

May 7, 2023



Agradecimientos

A Dios por darme salud, vida, perseverancia y voluntad de llegar hasta el final.

A los doctores Rinaldi Carlos y Vorobioff Juan por permitirme la participación en este proyecto, por el respeto a mis sugerencias e ideas y por confiar en mí.

A la Universidad Nacional de San Martín por la calidez de su gente y las excelentes personas que he conocido y me han ayudado durante mi crecimiento profesional.

A Stella por sus sabias palabras, su solidaridad constante y su contención en todos estos años.

A los docentes con los cuales he disfrutado de sus clases y por su calidez humana: Horacio, Héctor, Javier, Ignacio, Miguel, Jorge, Daniel, Nicolás, Romualdo, Liliana y otros tantos que he tenido el placer de conocer.

A mi país por darme la oportunidad de formarme.

A mi madre por los valores que me ha transmitido, sus consejos de vida, su apoyo incondicional y el tiempo cedido.

A mi mujer Samanta que siempre me ha brindado su compañía, su respaldo y el cariño tanto en momentos de dificultad como de éxitos, y por el tiempo que me ha concedido. Es por ello que este trabajo también es suyo.

A mis amigos David, Fabricio, Cristian, Joaquín, Nicolás por su apoyo moral y humano dentro y fuera de la universidad y a otros tantos que he conocido durante todo este tiempo.

A mi compañero de proyecto Nicolás por su paciencia y el profesionalismo con el que se desempeñó durante el desarrollo del proyecto.

A Lucas que desde su lugar siempre buscó la forma de ayudarnos con materiales o herramientas durante las etapas de construcción del trabajo.

Sebastián

A los doctores Rinaldi Carlos y Vorobioff Juan por su apoyo, guía, orientación, confianza y motivación durante todo el proyecto.

A mis padres y hermana por su constante apoyo y confianza en mí durante todo mi paso por la universidad.

A mi compañero de proyecto Sebastián por su inagotable paciencia, dedicación, entrega y apoyo incondicional durante todo el desarrollo del proyecto. Además estoy agradecido por su amistad durante el tiempo que hemos pasado trabajando juntos.

A mis amigos por su compañía y apoyo emocional durante los momentos más difíciles de mi formación universitaria.

A la Universidad Nacional de San Martín por proporcionarme las herramientas y recursos necesarios para llevar a cabo este proyecto.

A mis profesores por su inspiración, orientación e importancia en mi formación, así como su influencia en mi desarrollo profesional.

A mis colegas de trabajo por su amistad, colaboración y apoyo durante mi carrera.

A mi país por darme la oportunidad de formarme.

Nicolás

“ Si una persona es perseverante, aunque le cueste entender, se hará inteligente y aunque sea débil se volverá fuerte.”



Leonardo da Vinci

Resumen

La evolución de la robótica en la industria y su estudio en las instituciones académicas ha crecido exponencialmente en los últimos años. Dispositivos tales como robot del tipo spot de *Boston Dynamics* resultan útiles para la búsqueda de personas y la inspección en entornos peligrosos. Así también brazos robóticos de la empresa *KUKA* que son utilizados mayormente en plantas automotrices, han contribuido a lograr ambientes seguros y eficientes. La automatización de laboratorios consiste en utilizar tecnología para lograr incrementos de productividad, reducción del error humano en tareas repetitivas, producción de resultados repetibles, siendo estas algunas de las ventajas principales.

El proyecto describe la construcción y diseño de un sistema para la automatización de tareas en laboratorios, mediante la incorporación de un brazo robótico de código libre, impreso en 3D con 6 grados de libertad y conectividad inalámbrica. Teniendo como principal aplicación, el pesado de sustancias.

Abstract

The evolution of robotics in industry and its study in academic institutions has grown exponentially in recent years. Devices such as *Boston Dynamics'* Spot robot are useful for search and inspection in dangerous environments. Similarly, robotic arms from *KUKA*, mostly used in automotive plants, have contributed to achieving safe and efficient environments. Laboratory automation involves using technology to achieve increased productivity, reduced human error in repetitive tasks, and production of repeatable results, among other advantages.

The project describes the construction and design of a system for laboratory task automation by incorporating a 3D-printed, open-source robotic arm with 6 degrees of freedom and wireless connectivity, with its main application being substance weighing.

Motivación y objetivos

La motivación de este trabajo responde a la necesidad de modernizar la forma en la que se desarrollan las actividades en los laboratorios universitarios. Para ello, se propone incorporar un brazo robótico cuyo propósito será automatizar las tareas de laboratorio y a su vez integrar un sistema que fomente estas prácticas en forma remota. Se presenta entonces, un desafío implícito en la construcción y diseño del sistema final debido a la carencia de antecedentes sobre proyectos similares. De lo anterior, resulta el enriquecimiento no sólo en las actividades relacionadas con docentes y estudiantes, sino también contribuir a la mejora en la calidad de los proyectos vinculados a áreas de investigación y desarrollo de la alta casa de estudios.

Los objetivos que se plantean son:

- Modernización: incorporar nuevas tecnologías en espacios de trabajo.
- Simplificación: brindar a alumnos, docentes e investigadores, herramientas que faciliten la dinámica en actividades de laboratorio.
- Protección: disminuir el contacto o manipulación de elementos o sustancias que puedan ser peligrosas para el ser humano.
- Eficiencia: reducir tiempos y errores en las prácticas por tareas de carácter repetitivas.

Indice

1	Introducción	1
1.1	Antecedentes históricos	1
1.2	Definiciones	3
2	Estado del arte	5
3	Desarrollo	8
3.1	Componentes del sistema	8
3.1.1	Thor	8
3.1.2	GRBL	9
3.1.3	ESP32	11
3.1.4	Blynk	11
3.1.5	UART	13
3.1.6	WiFi	15
3.2	Presentación del sistema	17
3.3	Construcción de <i>Thor</i>	18
3.3.1	Materiales	18
3.3.2	Impresión 3D	20
3.3.3	Electrónica	21
3.4	Selección de la herramienta	27
3.5	Conectividad de <i>Thor</i>	28
3.5.1	Comunicación <i>ESP32-Blynk</i>	28
3.5.2	Comunicación <i>ESP32-Arduino MEGA</i>	29

3.6	Programación de <i>Thor</i>	30
3.7	Tablero Blynk	33
3.8	Diagrama de funcionamiento	34
3.9	Mejoras al modelo original	36
4	Resultados	41
4.1	Diseño	41
4.2	Electrónica	41
4.3	Aplicación de campo	42
4.4	Movimientos	44
4.5	Conectividad	46
4.6	Efactor final	47
4.7	Temperatura del sistema	47
5	Conclusiones	49
5.1	Hardware	49
5.2	Software	50
5.3	Ejecución de tarea	51
5.4	Conclusión final	55
5.5	Mejoras futuras	55
	Anexo	57
A	Conectividad con la aplicación Blynk	57
B	Especificaciones técnicas	60
C	G-Code soportados en V0.9	62
D	Documentación del proyecto	63
D.1	Acta de proyecto	63
D.2	Desglose de tareas del proyecto	65
E	Esquema PCB Buck	66
F	Hojas de datos	67
	Referencias	73

Lista de figuras

1.1	Robot Unimate de 1961	2
1.2	Ejemplo de articulaciones. Izquierda: articulación prismática. Derecha: articulación revoluta	3
1.3	Tipos de efector final	4
2.1	Densidad mundial de robots por país	5
2.2	Robot humanoide Ameca	7
3.1	Brazo robótico <i>Thor</i>	8
3.2	Medidas del brazo robótico <i>Thor</i>	9
3.3	Movimientos en brazo <i>Thor</i>	10
3.4	Módulo <i>NodeMCU32</i>	11
3.5	Plataforma <i>BlynkIoT</i>	12
3.6	Recepción de temperatura a través de un pin virtual.	13
3.7	Comunicación UART.	14
3.8	Trama UART.	15
3.9	Diagrama de flujo protocolo <i>CSMA/CA</i>	16
3.10	Esquema de conectividad de <i>Thor</i> y <i>Blynk</i>	18
3.11	Piezas de <i>Thor</i> impresas en material PLA.	21
3.12	Placa <i>Arduino MEGA 2560</i> utilizada en <i>Thor</i>	22
3.13	CNC Shield utilizada en <i>Thor</i>	23
3.14	Driver <i>DVR8825</i>	24
3.15	Configuración de micropasos en CNC shield.	24

3.16	Motores NEMA 17 implementados en <i>Thor</i>	26
3.17	Fuente de alimentación tipo buck con 5V de salida.	26
3.18	Placa de fuente de alimentación diseñada con <i>KiCad</i>	27
3.19	Efactor final seleccionado.	27
3.20	Servomotor para el gripper.	28
3.21	ID del dispositivo asignado por <i>Blynk</i>	29
3.22	Comunicación serie entre <i>Arduino Mega</i> y el módulo <i>NodeMCU32</i>	30
3.23	Tablero <i>Blynk</i> versión móvil. Izquierda: tareas predefinidas. Derecha: movimientos individuales de las articulaciones y gripper.	33
3.24	Tablero <i>Blynk</i> versión web. Izquierda: movimientos de articulaciones individuales. Derecha: Temperatura del sistema, consola para la visualización de instrucciones, funciones de parada y pausa.	34
3.25	Funcionamiento de <i>Thor</i>	35
3.26	Circuito para sensado de temperatura de <i>Thor</i>	37
3.27	Rediseño de tapa que contiene la electrónica.	38
3.28	Tapa rediseñada con software Fusion 360 de <i>Autodesk</i>	39
3.29	Luces leds indicadores de estado.	40
4.1	Descripción del movimiento de la tarea <i>pesada</i> a efectuar por <i>Thor</i>	43
4.2	Descripción del movimiento de la tarea <i>agregado</i> que debe efectuar <i>Thor</i>	44
4.3	Primera prueba de movimientos.	45
4.4	Segunda prueba de movimientos	45
4.5	Primeras pruebas de pesada	46
5.1	Tablero web para la ejecución de la tarea pesada	51
5.2	1 - Inicio, 2 - Toma de recipiente <i>A</i> , 3 - Traslación a zona de pesado y 4 - Colocación en balanza	52
5.3	5 - Toma de recipiente <i>B</i> . 6 - Traslación hacia zona de pesado. 7 - Vertido de contenido de recipiente <i>B</i> en <i>A</i>	53

5.4	8 - Apartado de recipiente <i>B</i> hacia zona segura. 9 - Movimiento a <i>home</i> . 10	
	- Fin	54
A.1	Correo electrónico con link de acceso a <i>Blynk</i>	57
A.2	Generación de clave de acceso	58
A.3	Selección de dispositivo	58
A.4	Tablero web de usuario	59
A.5	Tablero de usuario desde la aplicación	59

Lista de tablas

1	Lista de materiales del proyecto.	20
2	Resolución de micropasos para driver DVR8825 [13].	25
3	Ensayo de temperatura en impresiones de prueba.	48
B.1	Motores	60
B.2	Driver	60
B.3	Servo	60
B.4	CNC shield	60
B.5	Fuente switching	60
B.6	Módulo ESP32	61
B.7	Arduino Mega 2560	61

Lista de acrónimos

ABS Acrylonitrile Butadiene Styrene (Acrilonitrilo Butadieno Estireno). 20

AI Artificial Intelligence (Inteligencia Artificial). 5, 6, 56

CNC Computer Numerical Control (Control Numérico Computarizado). 4, 9, 22–24

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance (Acceso múltiple por detección de portadora y prevención de colisiones). 16

DOF Degree of freedom (Grados de libertad). 3

GRBL Garble. 21, 50, 56

IFR International Federation of Robotics (Federación Internacional de Robótica). 2, 5

IoT Internet of Things (Internet de las cosas). 5

ISM Industrial, Scientific, and Medical (Industriales, Científicas y Médicas). 15

JIRA Japan Industrial Robot Association (Asociación Japonesa de Robots Industriales). 2

NEMA National Electrical Manufacturers Association (Asociación Nacional de Fabricantes Eléctricos). 5, 25, 26

PCB Printed Circuit Board (Placa de circuito impreso). 26

PLA Polylactic Acid (Ácido Poliláctico). 4, 20, 21

R.U.R Rossum's Universal Robot (Robots Universales Rossum). 1

ROS Robot Operating System (Sistema Operativo Robótico). 56

STL STereoLithography (Estereolitografía). 20

UART Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal). 4, 13–15, 29, 31, 47, 50

VR Virtual Reality (Realidad Virtual). 5

1 Introducción

1.1 Antecedentes históricos

A lo largo del tiempo existió una fascinación por la construcción de máquinas y dispositivos que pudieran imitar movimientos y funciones que habitualmente realizaban los humanos [1]. Estas máquinas eran denominadas por los griegos con el nombre *automato* que posteriormente se convirtió en **autómata**. La primera aparición de la palabra robot fue en el 1921 cuando se estrenó la obra *Rossum's Universal Robot (Robots Universales Rossum)*. El origen de la palabra provino del eslavo *robota* y refería al concepto de *servidumbre* o *esclavo* [2]. En aquella época la concepción que se tenía sobre el robot se basaba en que debían servir a sus jefes humanos y ejecutar trabajos físicos que éstos le indicasen. Con el paso de los años, esa percepción fue modificándose y finalmente los robots se concibieron como una herramienta que permitía reducir la exposición a riesgos para el ser humano y a su vez, incrementar la eficiencia en los trabajos que debieran realizarse.

Las bases del robot industrial actual fueron patentadas por el norteamericano George C. Devol en el año 1961 y la primera máquina robótica de aplicación industrial se denominó Unimate y fue implementada en la fábrica de General Motors de Trenton, Nueva Jersey [3]. Los robots manipuladores fueron empleados para diversas tareas como por ejemplo: ensamblado, soldadura, pintura, manipulación de herramientas o dispositivos especiales. A diferencia de los robots móviles, poseen una base fija en la cual operan en un espacio restringido, garantizando la seguridad al quedar su operación limitada en un volumen definido y también la eficiencia dado que si el entorno en el que opera se encuentra ordenado, el robot puede ser rápido y preciso.



Figura 1.1: Robot Unimate de 1961

La primera asociación robótica del mundo la comenzó Kawasaki en el año 1972 y se denominó *Japan Industrial Robot Association (Asociación Japonesa de Robots Industriales) (JIRA)*. En Europa comenzó algunos años más tarde cuando en 1980 se fundó la *Federación Internacional de Robótica* o *IFR* en Suecia [4]. La evolución de la robótica fue tal que resulta habitual su incorporación en diversas industrias para las que desarrollan diversas aplicaciones tales como: carga y descarga, montaje, pulido, testing y análisis de laboratorio, empaquetado y paletizado, líneas de envasado, aplicaciones de soldadura, entre otros. Cada una de estas tareas son de tipo repetitivas y requieren un nivel de precisión tal que no exista diferencia entre procesos. Por lo tanto el uso de robots provee las siguientes ventajas:

- Puede operar las 24 hs y hasta de forma autónoma.
- Realizar tareas repetitivas con la misma precisión y eficiencia.
- Puede ser adaptable y versátil ante diferentes entornos.

1.2 Definiciones

Los robots han sido creados en base a ciertas características propias del ser humano. Por ejemplo en un brazo robótico, cada uno de sus elementos son rápidamente distinguibles cuando se compara con la anatomía del brazo humano. Debido a esto, se emplean los términos cuerpo, brazo, codo y muñeca. Las articulaciones están sostenidas por elementos o eslabones y cada una de ellas puede tener movimientos de tipo rotacional, de traslación o una combinación de ambos (figura 1.2).

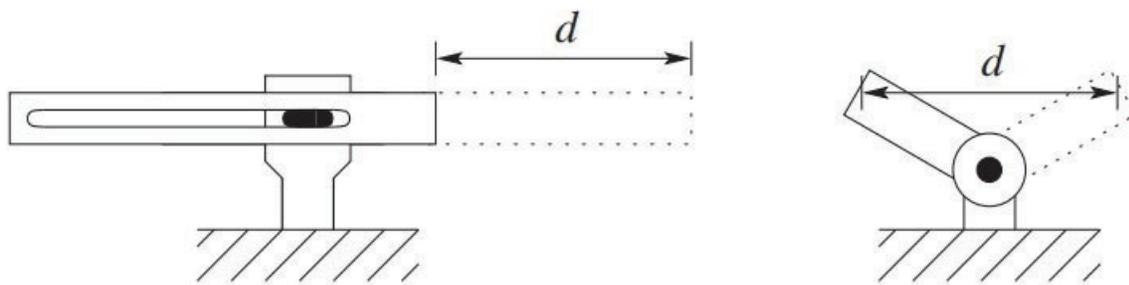


Figura 1.2: Ejemplo de articulaciones. Izquierda: articulación prismática. Derecha: articulación revoluta

Los distintos movimientos independientes que puede ejercer una articulación se denominan *grados de libertad* o *DOF* [5]. Para que un robot manipulador pueda tomar un objeto, debe utilizar una herramienta o un *efector final*. Los efectores pueden ser simples como un *gripper* el cual abre y cierra (figura 1.3 b), o sistemas más sofisticados como pinzas con aplicación de vacío para sujeción de láminas de vidrio o manos de tipo humano (figura 1.3 a).



Figura 1.3: Tipos de efector final

De acuerdo con los tipos o combinaciones de articulaciones prismáticas y/o revolutas existen distintos tipos de cadenas cinemáticas. Sin embargo los diseños más utilizados en la practica son los siguientes [6]:

- **Manipulador articulado (RRR):** llamado también manipulador antropomorfo o de tipo revoluto. La representación simbólica consiste en asignar el cuerpo, hombro y codo como tres articulaciones revolutas.
- **Manipulador esférico (RRP):** su representación consiste en dos articulaciones revolutas (cuerpo y hombro) y una tercera de tipo prismática.
- **Manipulador cilíndrico (RPP):** esta configuración consiste en una articulación revoluta (cuerpo) y dos prismáticas (hombro y codo).
- **Manipulador cartesiano (PPP):** Las tres articulaciones son de tipo prismáticas.

2 Estado del arte

El 14 de diciembre del 2021, la *IFR* emitió un comunicado de prensa en el cual estableció que la media de robots a nivel global fue de 126 por cada 10000 empleados, cuando en el año 2015 el estimado era de 66 unidades. Las regiones con mayor densidad pertenecen a Asia y Australia con 134 unidades, luego Europa con 123 y finalmente el Continente Americano con 111 unidades. Los 5 países de mayor automatización a nivel mundial son Corea del Sur, Singapur, Japón, Alemania y Suecia.

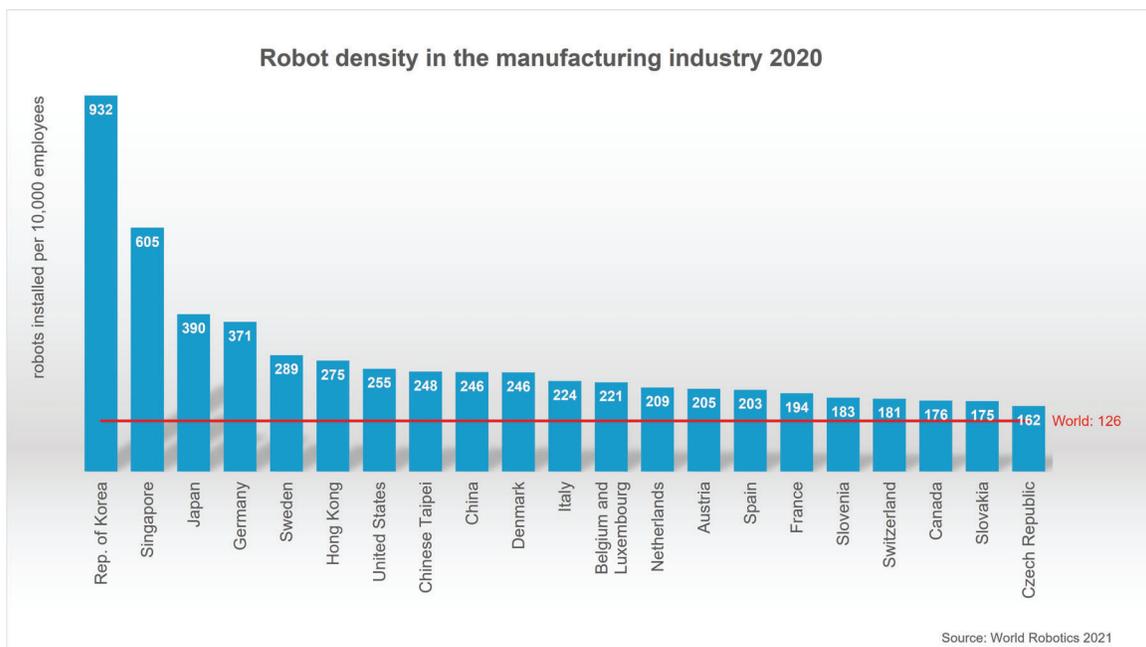


Figura 2.1: Densidad mundial de robots por país

A pesar de la pandemia mundial reciente, el año 2020 fue el tercer año más exitoso para la industria de la robótica con una instalación de 383.545 unidades respecto del 2017 (400.000) y 2018 (422.000). La perspectiva para el período 2021-2024 es de crecimiento sostenido, y se espera alcanzar las 500.000 unidades instaladas para el año 2024 [7]. En esta nueva revolución industrial cuyo nombre se conoce como *Industria 4.0*, se busca digitalizar líneas de producción y fabricación mediante realidad virtual (*VR*), inteligencia artificial (*AI*) o el internet de las cosas (*IoT*). Un robot dotado de IA permite que este pueda realizar tareas

en forma autónoma y además tomar decisiones en función de un entorno de trabajo y ciertos parámetros establecidos. Un ejemplo de robot dotado con AI es el denominado *Cobot* o robot colaborativo. Estos robots son más sencillos de operar respecto a los robots industriales y entre las aplicaciones prácticas que pueden desarrollar se encuentran:

- **Monitoreo:** detección de condiciones cambiantes de espacio en el lugar de trabajo.
- **Visión artificial:** detección y reconocimiento sobre la presencia y orientación de un objeto o pieza, realizar tareas de inspección y selección como así también analizar los resultados de una operación y tomar las decisiones correspondientes.
- **Adaptación:** ajuste de la orientación respecto a la tarea que debe ejercer, controlar la fuerza requerida para alguna aplicación o detectar posibles colisiones y evitarlas.
- **Aprendizaje:** predicción y diagnóstico de errores en la producción, identificando patrones en forma automática.
- **Implementación:** comienzo de una tarea en poco más de una hora, sin necesidad de programadores o personal externo. Gracias a la comunicación inteligente, los cobots se pueden conectarse entre sí e intercambiar flujos de datos.
- **Supervisión de maquinaria:** realización de aplicaciones de supervisión de maquinaria, control, organización de tareas y mejoras en la eficiencia de los equipos cercanos.

Otro ejemplo es *Ameca*, un robot con forma humana considerado el más avanzado del mundo. Fue desarrollado por Engineered Arts y puede interactuar con humanos y otros robots. Entre sus características posee AI y aprendizaje automático o *machine learning*¹.

¹Machine learning es un rama de la IA cuyo propósito es desarrollar algoritmos que permitan que las computadoras aprendan

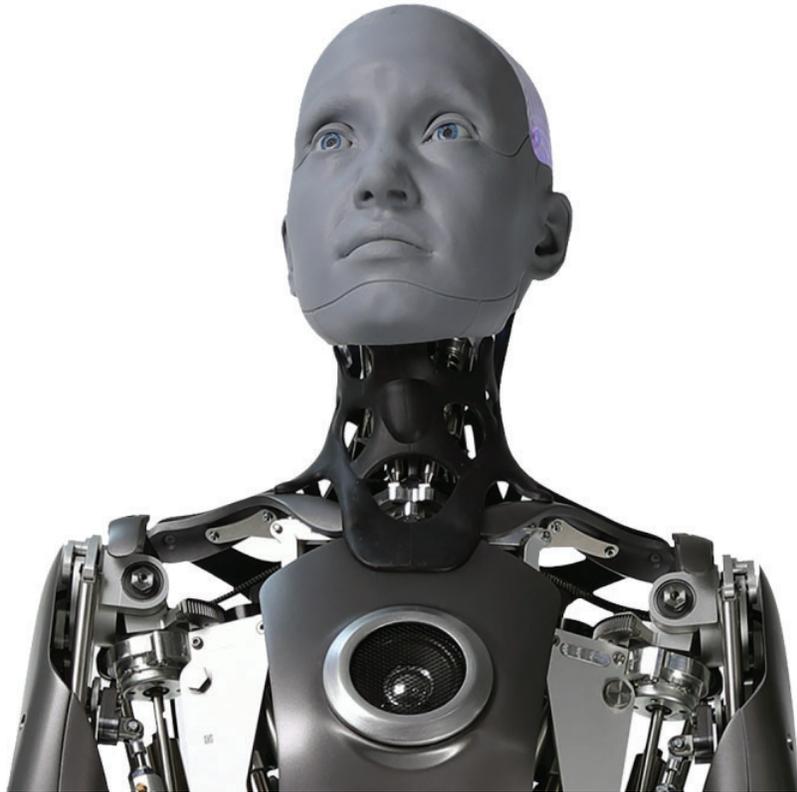


Figura 2.2: Robot humanoide Ameca

Ameca posee un poderoso sistema operativo robótico denominado *Tritium* el cual permite movimientos suaves y similares a los del ser humano, tanto expresiones como gestos de excelente naturalidad [8].

3 Desarrollo

3.1 Componentes del sistema

3.1.1 Thor

Como se mencionó anteriormente, el alcance de este proyecto era la automatización de un laboratorio mediante un brazo robótico y no la construcción en sí del mismo. Es por ello, que se efectuó la búsqueda de un robot ya modelado. Por lo tanto se adoptó un brazo robótico (figura 3.1) desarrollado por **Ángel L.M**, denominado *Thor* [9] cuyas características se detallan a continuación:

- 6 DOF con articulaciones de tipo revolutas.
- Piezas de material termoplástico denominado PLA.
- Equipado con 7 motores paso a paso.



Figura 3.1: Brazo robótico *Thor*.

Las dimensiones del brazo robótico se muestran en la figura 3.2:

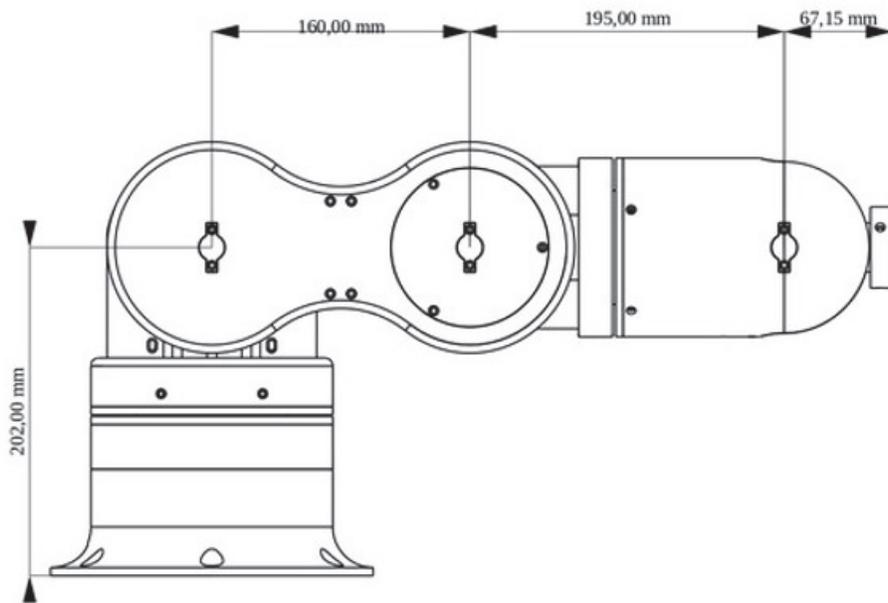


Figura 3.2: Medidas del brazo robótico *Thor*.

3.1.2 GRBL

El software de operación que se encuentra embebido en la placa de control de *Thor* se denomina **Grbl** y su versión es v0.9. Su característica principal es su alto rendimiento y debido a ello es habitualmente utilizado para el control de movimientos de equipos CNC ² y que ha adoptado la impresión 3D como método de movimiento de los motores. La forma en la que se producen los movimientos de los motores, generar una pausa o lograr que el equipo se dirija hacia su posición de reposo o *home*, es mediante un lenguaje de programación en el que se envían instrucciones que se denominan *g-code*³. Dichas

²CNC o control numérico computarizado es un método que permite mecanizar piezas en base un sistema de coordenadas de referencia que determinan la posición de la herramienta de trabajo empleando un programa que será ejecutado por un microcontrolador

³*g-code* o también conocido como RS-274 es ampliamente utilizado para decirle a una máquina herramienta a dónde debe moverse, con qué velocidad y qué trayectoria debe seguir.

instrucciones consisten en indicarle a una máquina que operación debe realizar. La sintaxis de este lenguaje se representa como un conjunto de coordenadas correspondientes a cada motor seguido de otros parámetros que pueden ser la velocidad o comandos especiales. Veamos algunos ejemplos:

```
1 G0 A10 B25 C25 D15 X45 Y3 Z3
2 G1 A90 F500
3 M3 S900
```

En la primera línea de comandos se definió una instrucción de movimiento (*G0*) y siete motores (A-Z) con sus respectivos valores de movimiento. La segunda línea también emplea un movimiento (*G1*), pero agrega además la velocidad del motor A. El último comando habilita la herramienta (*M3*) y le confiere una apertura total de la misma (*S900*). De esta manera, los movimientos requeridos por el usuario se tradujeron en diversas líneas g-code en las que se definieron qué motor debía rotar y en qué dirección, como así también la apertura o cierre del gripper. El esquema siguiente (figura 3.3) representa como se envían dichas instrucciones de movimiento:



Figura 3.3: Movimientos en brazo *Thor*.

3.1.3 ESP32

Para la conectividad en forma remota se empleó el chip *ESP32*, ampliamente utilizado en diversos proyectos IoT. Éste pertenece a la familia de chip SoC con tecnología bluetooth y wifi. Cuenta con un procesador doble núcleo de 32 bits, logrando alcanzar frecuencias de trabajo entre 160 y 240 MHz. Para este proyecto se utilizó la placa *NodeMCU-32* (figura 3.4) que contiene el chip ESP32 embebido, brindando facilidad en el acceso a los pines del mismo. El módulo cuenta con un puerto usb para la carga del firmware y a su vez, funciona como alimentación del chip embebido mediante un regulador de voltaje integrado. Las características adicionales se encuentran en el apéndice.



Figura 3.4: Módulo *NodeMCU32*.

3.1.4 Blynk

El control del brazo robótico *Thor* se efectuó en forma remota mediante la plataforma denominada *Blynk IoT*, la cual permite la gestión y supervisión de dispositivos que se encuentran conectados a su red. El hardware conectado a Blynk puede administrarse tanto desde su sitio web como desde su aplicación móvil [10].

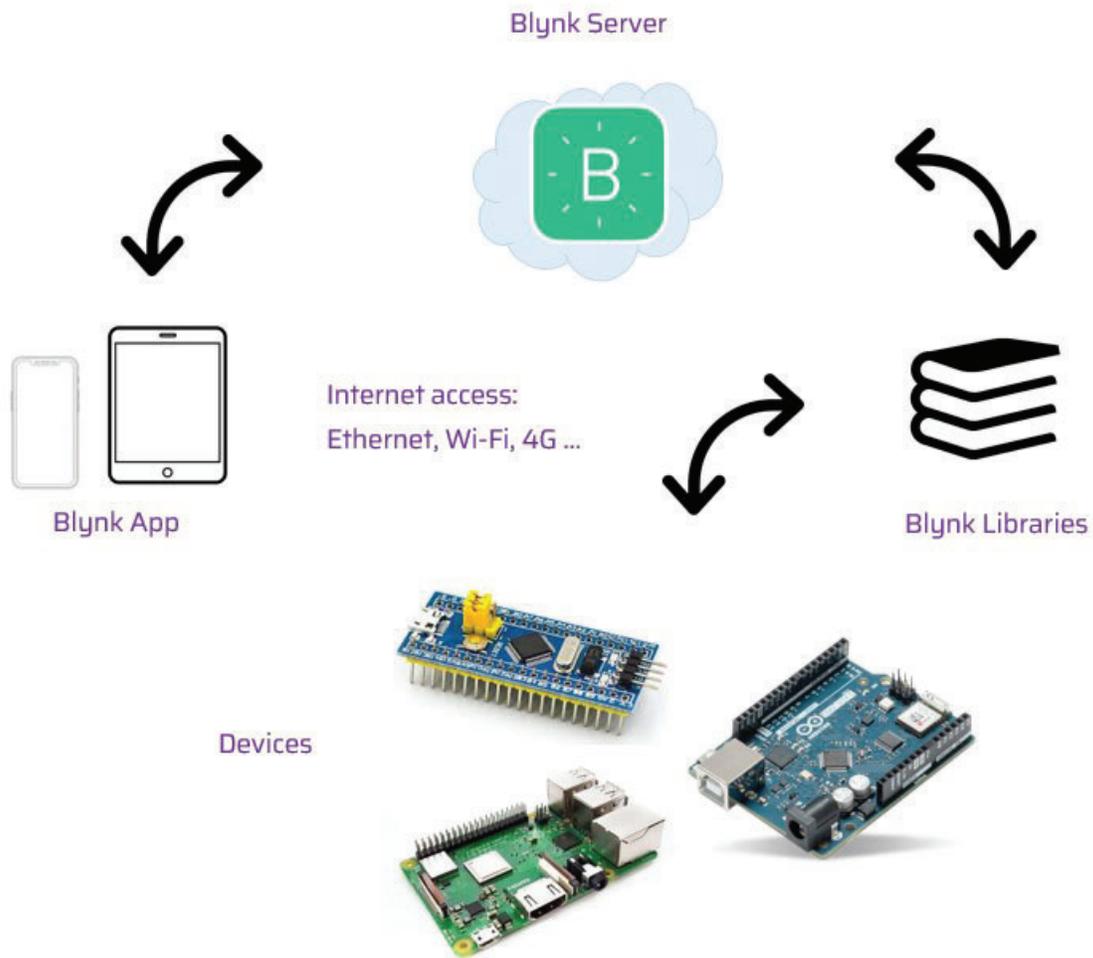


Figura 3.5: Plataforma *BlynkIoT*.

La forma en intercambiar información entre el servidor de Blynk y el hardware o dispositivo, fue mediante pines virtuales que se definen como “canales” de intercambio de varios tipos de datos (int, strings, float, double). Debe tenerse en cuenta que los pines virtuales no poseen representación física. Sin embargo, su empleo permite interactuar a nivel de hardware e implementar la lógica particular deseada.

Vale aclarar que no solamente pueden enviarse datos desde el pin virtual hacia el hardware conectado, sino que además pueden escribirse dichos pines al efecto de actualizar información como podría ser el caso de la figura 3.6 donde se monitorea la temperatura de un sistema en forma de gráfica temporal. El instructivo de conexión con la plataforma, se detalla en

el apéndice A.

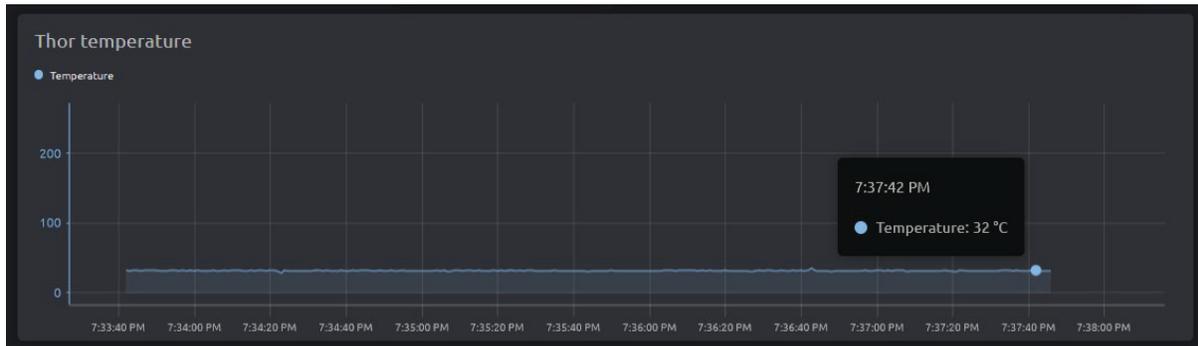


Figura 3.6: Recepción de temperatura a través de un pin virtual.

3.1.5 UART

Uno de los dispositivos mas utilizados en el intercambio de datos entre placas se denomina UART que significa *transmisor-receptor asíncrono universal*. El cual resulta de bajo costo y posee facilidad en su implementación. Esta conexión del tipo hardware emplea dos cables: uno para la transmisión y otro para recepción de datos, pero además deben compartir una conexión a tierra (figura 3.7). Los tipos de comunicación UART pueden ser unidireccionales (*simplex*), transmisión de ambos lados pero de a uno a la vez (*half-duplex*) y en ambas direcciones (*full-duplex*).

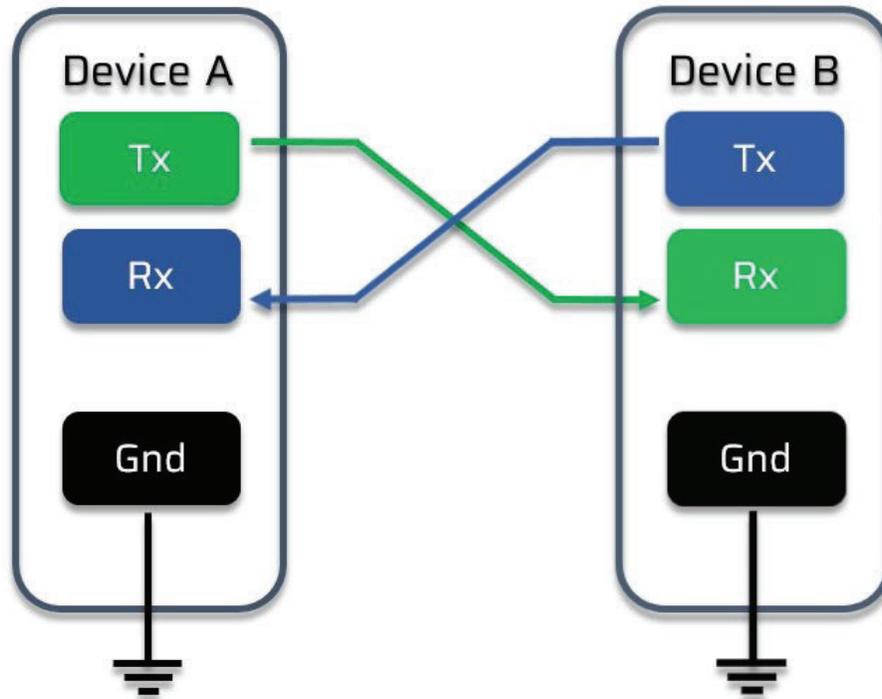


Figura 3.7: Comunicación UART.

La comunicación entre el transmisor y el receptor es asincrónica, esto significa que no comparten una señal de reloj común. Esto simplifica el protocolo, sin embargo requiere que los dispositivos estén configurados a la misma velocidad de transmisión de datos de manera de obtener la misma sincronización de bits. Las velocidades de transmisión se describen en *baudios*⁴ y los valores habituales son 4800, 9600, 19200, 56700, 115200 [11].

Durante la comunicación entre dispositivos, los datos viajan en serie desde el origen hacia el destino. Para lograr esto, la información se agrupa en tramas. Es importante que la velocidad de datos de los equipos que se comunican sea coincidente, así como que los parámetros y la estructura de la trama sean los mismos para que la comunicación sea exitosa. Antes de efectuar la transmisión de información primero se requiere que los datos en paralelo provenientes de un dispositivo maestro (como por ejemplo un CPU)

⁴Unidad de medida utilizada en telecomunicaciones que representa el número de símbolos por segundo en un medio de transmisión.

deban convertirse a serie. A continuación, los bits que componen el dato se agrupan en paquetes que pueden ser de 5 a 9 bits, aunque lo más habitual es de 7 o 8 bits. A este grupo se le coloca un bit de inicio y otro de parada (figura 3.8). El bit de inicio indica al receptor que los datos de usuario están llegando y el bit de parada denota el final de los mismos. Opcionalmente puede existir un bit de comprobación de errores (*paridad*) entre el final de los datos de usuario y el bit de parada.

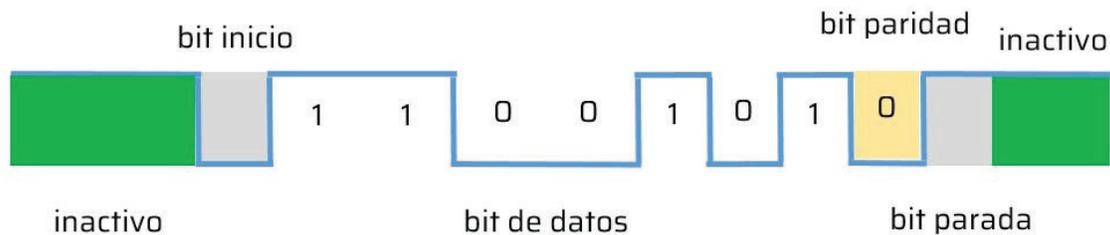


Figura 3.8: Trama UART.

Cuando se va a transmitir una trama, los datos de usuario invierten sus posiciones de manera tal que el bit menos significativo pasa a ser el primero que se transmite. Una vez recibida la trama se deberán convertir los bits nuevamente a paralelo para posteriormente dirigirlos hacia el dispositivo receptor.

3.1.6 WiFi

El estándar 802.11 conocido popularmente como *WiFi* es un medio de conectividad inalámbrica que utiliza ondas de radio para la transmisión de señales. Opera en banda de frecuencias sin licencia⁵ establecidas por el ITU-R⁶. Todos los dispositivos pueden utilizar

⁵Las bandas sin licencia como las ISM son bandas de radio como por ejemplo 902-929 MHz, 2.4-2.5 GHz, 5.725-5.825 GHz que están reservadas internacionalmente para el uso de energía de radiofrecuencia para fines industriales, científicos y médicos.

⁶La ITU-R es un organismo internacional ubicado en Ginebra (Suiza) encargado de la regulación de las telecomunicaciones a nivel mundial entre las administraciones y empresas operadoras.

este espectro pero con la condición de limitar su potencia de transmisión (menor a 1 W) para no afectar la coexistencia de los distintos equipos conectados [12]. En el caso del WiFi, la transmisión y recepción de datos se realiza en el rango de los GHz.

El funcionamiento de WiFi se basa en el protocolo denominado CSMA/CA el cual permite que múltiples estaciones utilicen un mismo medio de transmisión. En este protocolo cada equipo debe anunciar su intención de transmitir con el propósito de evitar colisiones entre los paquetes de datos. De esta manera, los demás dispositivos pertenecientes a la red detectarán cuando hayan colisiones y en caso de querer transmitir un paquete, deberán primero aguardar un tiempo aleatorio (y además que durante ese intervalo de espera el canal se encuentre libre), para luego proceder a la transmisión reduciendo la probabilidad de colisiones en el canal (figura 3.9).

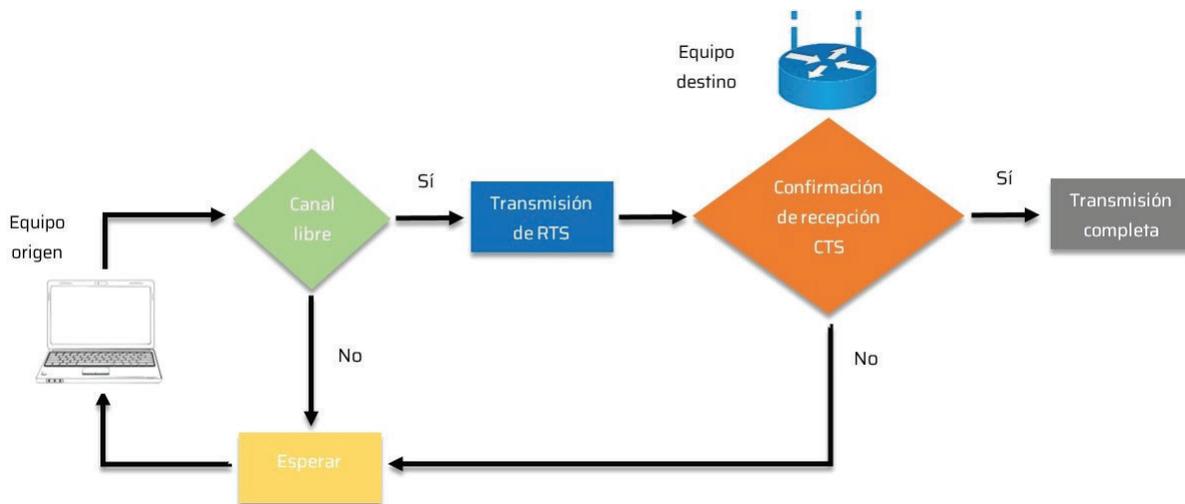


Figura 3.9: Diagrama de flujo protocolo CSMA/CA

Cuando se envía una trama, el dispositivo de origen envía un paquete de solicitud de transmisión (*RTS o Request To Send*) el cual contiene las direcciones del equipo de origen y de destino. Cuando el equipo de destino recibe esta solicitud, significa que está en condición de recibir la trama de datos. La respuesta del equipo receptor es una trama que puede tener dos contestaciones posibles: recepción correcta (*CTS o Clear To Send*) o receptor ocupado (*RxBusy*). Si responde que se encuentra libre, el transmisor envía el

paquete que se encontraba en espera. Si dicho paquete se recibe correctamente el receptor devuelve un nuevo mensaje de confirmación positivo *ACK*. En caso contrario, devolverá un mensaje de confirmación negativa *NAK* y entonces el equipo de origen tratará de enviar la trama nuevamente. El proceso se repite una cantidad de veces hasta lograr la transmisión correcta de la trama de datos.

3.2 Presentación del sistema

La automatización del sistema implicó los siguientes elementos:

- *Thor*: dispositivo como herramienta de automatización.
- Servidor *Blynk*: transferencia de datos entre el hardware de usuario y el módulo *NodeMCU32*.
- Módulo *NodeMCU32*: conexión entre el servidor *Blynk* y el hardware de *Thor*.
- Dispositivos de usuario: relación entre el usuario y los movimientos del brazo a través de la aplicación *Blynk IoT* mediante tablets, teléfonos inteligentes, computadoras.

En la figura 3.10 se representan los diferentes elementos que componen al sistema completo.

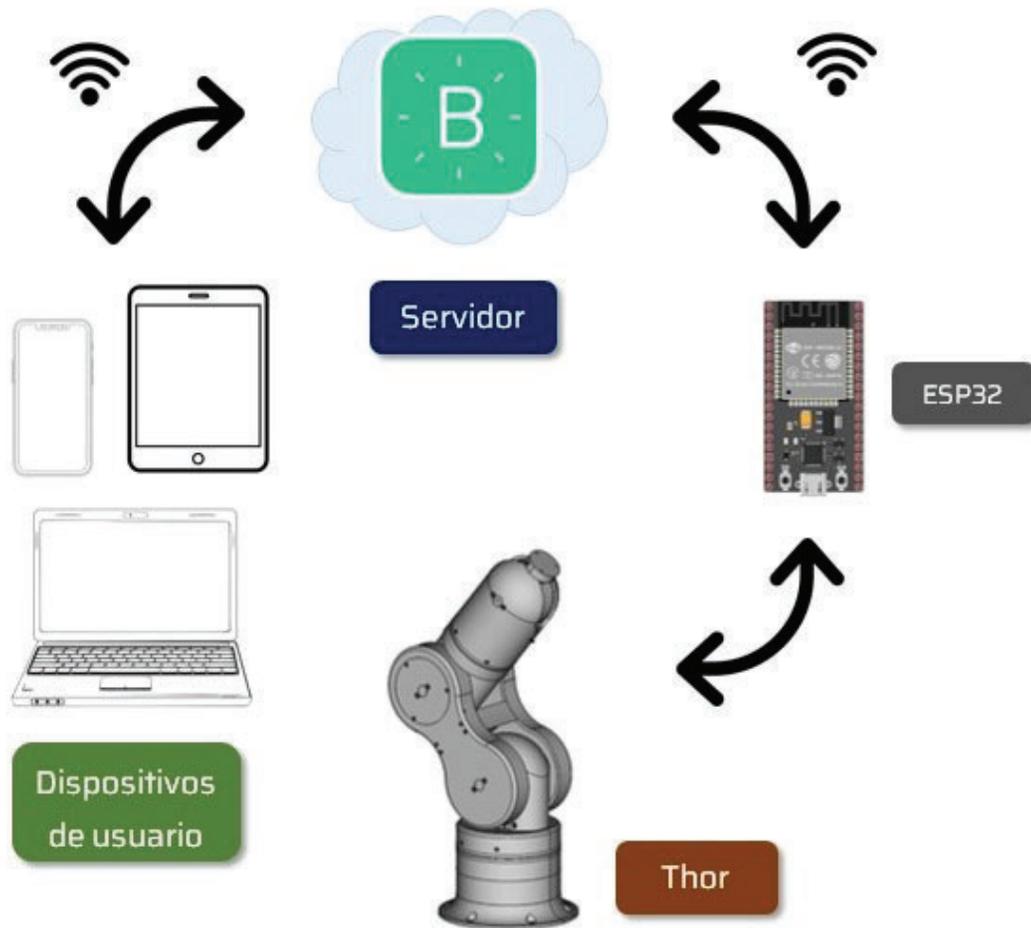


Figura 3.10: Esquema de conectividad de *Thor* y *Blynk*

3.3 Construcción de *Thor*

3.3.1 Materiales

Para la construcción de *Thor* fueron necesarios los siguientes componentes:

Materiales	Cantidad (unidades)
Rollo filamento PLA 1kg	5
Rollo filamento ABS 1kg	1

Ventiladores 12V 40mm	7
Correa dentada abierta 6 mm x 1m	1
Correa dentada cerrada 6mm 104 dientes	1
Tecla 2 contactos 12V	4
Placa CNC shield V3	2
Placa Arduino Mega 2560	1
Placa perforada 50x70mm	1
Borneras PCB 10A	3
Tira de pines macho 2.54mm	2
Resistencia 22k Ω 1/4W	1
Motor 17HS4401	1
Motor KS42STH34-1504A	3
Motor 17HS13-0404S-PG5	3
Driver DVR8825	7
Servo motor MG996R	1
Placa NodeMCU ESP32	1
Rodamiento 16014ZZ	1
Rodamiento 625ZZ	11
Varilla de acero inoxidable 100x5 mm	1
Tuercas M2/M3/M4/M5	4/155/4/1
Módulo buck MP1584	1
Polea dentada GT2 20 dientes 5mm	3
Polea dentada GT2 40 dientes 5 mm	2
Rodamiento MF84ZZ	2
Bolillas de acero 6mm	40
Tornillos M2/M3/M4/M5 largos varios	7/190/4/1
Fuente de Switching 12V 40A	1

Final de carrera óptico 5V	4
Final de carrera microswitch	1
Cable 1mm x metro	8

Tabla 1: Lista de materiales del proyecto.

3.3.2 Impresión 3D

Las partes necesarias para el armado del brazo se componen en 49 archivos STL⁷, de los cuales algunos de ellos fueron necesarios en múltiples etapas del ensamblado por lo que debieron imprimirse más de una vez. Algunas de las piezas fueron de gran tamaño lo que implicó recurrir a equipos con superficie de impresión mayores a 20x20 cm. Piezas específicas como engranajes y otras sobre las que se ejerce mayor fuerza, fueron impresas con material ABS, el cual es de elevada dureza en comparación con el PLA. Valiéndose de esta propiedad, resultó la impresión de dos engranajes dado que ejercen una fuerza superior al soportar el peso del resto de las articulaciones. Las piezas remanentes fueron impresas en material PLA y algunas de ellas se muestran en la figura 3.11:

⁷STL es un formato de archivo informático de diseño asistido por computadora (CAD) que define geometría de objetos 3D, excluyendo información como color, texturas o propiedades físicas que sí incluyen otros formatos CAD.



Figura 3.11: Piezas de *Thor* impresas en material PLA.

3.3.3 Electrónica

La placa de desarrollo responsable de gestionar el software de operación de *Thor* fue *Arduino MEGA 2560* (figura 3.12), la cual contiene grabado GRBL de versión 0.9. Dicha placa efectuó el envío de señales necesarias para el movimiento de los motores.

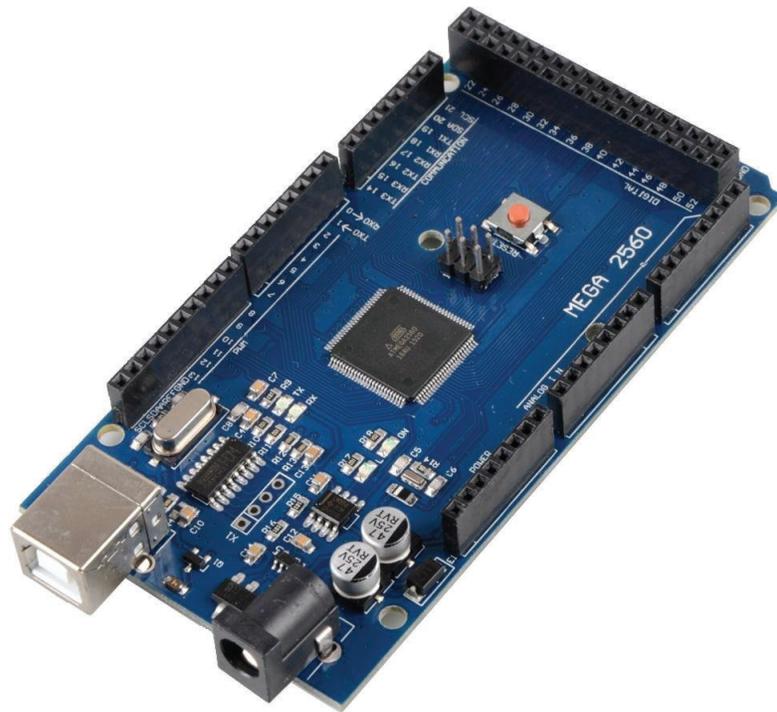


Figura 3.12: Placa *Arduino MEGA 2560* utilizada en *Thor*.

El módulo de expansión CNC shield fue el dispositivo que recibió las señales provenientes del *Arduino MEGA 2560* y además vinculó los drivers con los motores paso a paso. En la figura 3.13, donde se lo puede visualizar, se aprecian las conexiones para el control de hasta 4 motores de tipo paso a paso (recuadros X, Y, Z y A) a los cuales se les puede indicar la cantidad de pulsos que permitirán la rotación (*step*), como así también la dirección o el sentido de giro (*dir*).

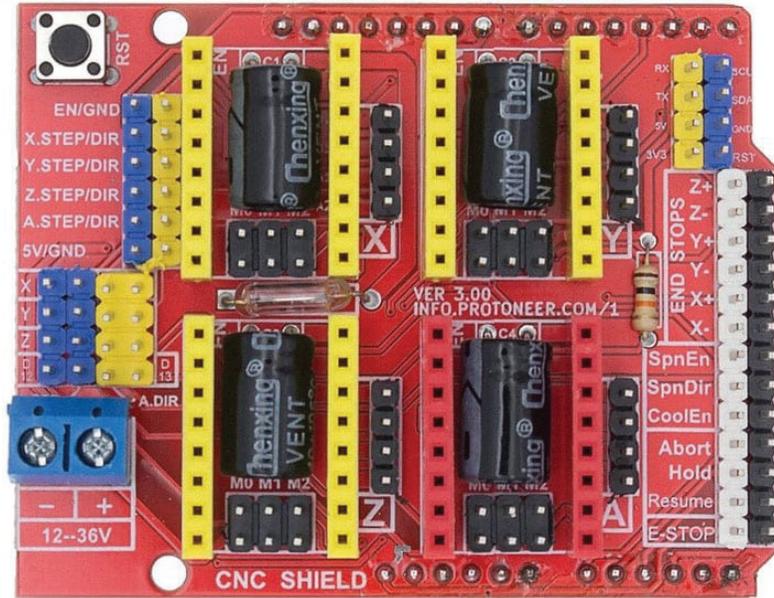


Figura 3.13: CNC Shield utilizada en *Thor*.

Los tipos de drivers que admite el shield son DVR8825 y A4988. Estos drivers tienen como función proveer los pulsos de corriente en las bobinas para que estas últimas produzcan el giro del rotor. A pesar de poseer conexiones similares, se diferencian en términos de la cantidad de corriente que pueden suministrar. El driver DVR8825 (figura 3.14) puede proveer de una corriente máxima por fase de 2.2A siempre y cuando se cuente con ventilación adecuada. Mientras que el driver A4988 puede suministrar hasta 2A.

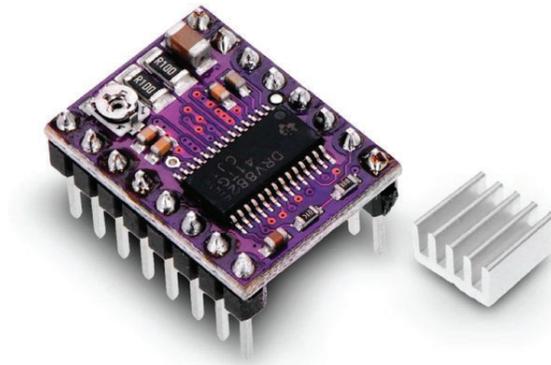


Figura 3.14: Driver DVR8825.

Indistintamente del tipo de driver utilizado, existe la posibilidad de configurar la resolución de pasos que los motores pueden efectuar de manera de lograr movimientos angulares de mayor precisión respecto al mínimo por defecto.

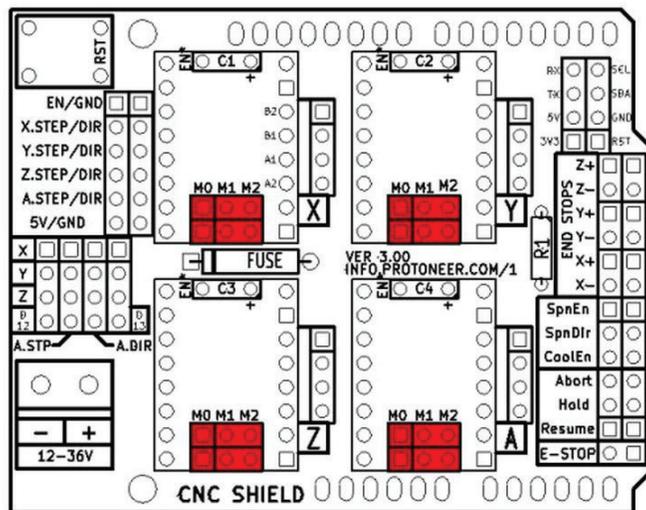


Figura 3.15: Configuración de micropasos en CNC shield.

Se utiliza la técnica de *microstepping* (micropasos) que permite obtener una mayor precisión

posicional como así también una operación más silenciosa y una mayor eficiencia en el consumo de energía. Las configuraciones posibles de micropasos que se le pueden aplicar a los motores son las que se presentan en la tabla 2:

M0	M1	M2	Resolución de micropaso
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

Tabla 2: Resolución de micropasos para driver DVR8825 [13].

El brazo empleó en total 7 motores paso a paso NEMA 17 (cuyos modelos se especifican en la tabla 1) para las articulaciones y se corresponden con los de la figura 3.16. La resolución de paso típico especificado en estos tipos de motores es un ángulo rotacional de 1.8° por paso, siendo 200 pasos un giro completo. Entonces, adoptando una configuración de micropasos de 1/32 de la tabla 2, se obtiene que la resolución del motor es de 6400 pasos por cada vuelta completa o un ángulo rotacional de $0,05625^\circ$ por paso. El resultado anterior denota que pueden lograrse pasos de mayor resolución que se pueden traducir en movimientos rotacionales suavizados. De acuerdo a las hojas de datos de los motores utilizados, la corriente máxima requerida es de 1,7 A. Las características de los tipos de motores utilizados se detallan en el anexo F.



Figura 3.16: Motores NEMA 17 implementados en *Thor*.

La alimentación del sistema completo se implementó mediante una fuente switching de 12V y 40A, cuyas características técnicas se encuentran en el anexo B (tabla B.5). Algunos de los dispositivos instalados requirieron una tensión de 5V para su funcionamiento. Para lograr suministrar esa alimentación se construyó un PCB en el que se incorporó una fuente buck de 5V-3A (figura 3.17) .

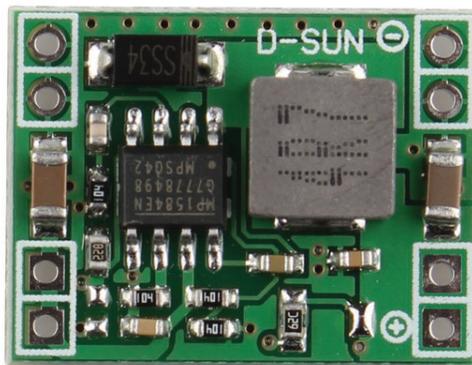


Figura 3.17: Fuente de alimentación tipo buck con 5V de salida.

El diseño del PCB se efectuó mediante el software *Kicad*. En la figura 3.18, se logran

apreciar a la izquierda los terminales de 12V destinados a los ventiladores y a la derecha los terminales de 5V. También se incluyó un circuito de sensado de temperatura para los motores el cual se detallará en el capítulo 3.9.

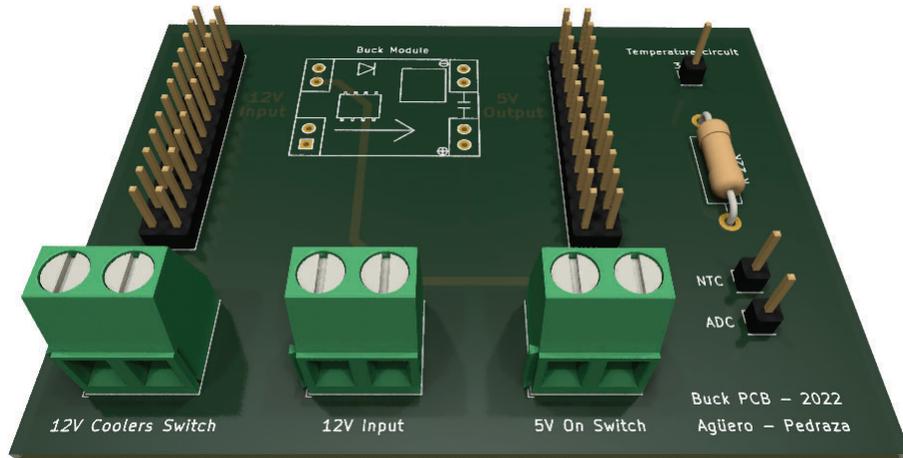


Figura 3.18: Placa de fuente de alimentación diseñada con *KiCad*.

3.4 Selección de la herramienta

De los tipos de efectores finales mencionados en el apartado 1.2 y en función de la tarea que el brazo debió efectuar, la opción adecuada fue seleccionar un gripper como herramienta de agarre (figura 3.19).

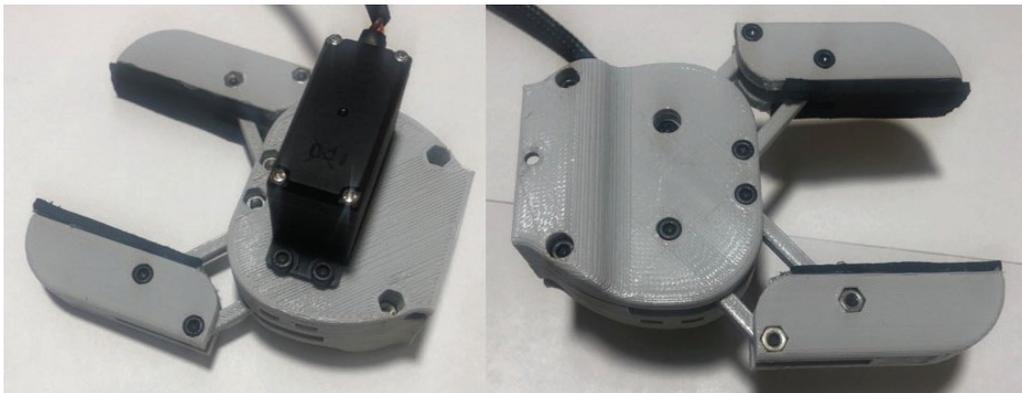


Figura 3.19: Efecto final seleccionado.

La construcción del mismo se realizó con un servomotor cuyo torque máximo es de 0.98 N*m (figura 3.20). Las especificaciones del mismo se encuentran en el anexo B (tabla B.3) y la hoja de datos en el anexo F.



Figura 3.20: Servomotor para el gripper.

3.5 Conectividad de *Thor*

3.5.1 Comunicación *ESP32-Blynk*

El brazo robótico *Thor* se conectó al servidor de *Blynk* a través de conexión WiFi. Para asociar el módulo inalámbrico *NodeMCU32* al servidor, primero es necesario crear una plantilla definiendo el tipo de hardware a conectar, como así también el medio de conexión. Posteriormente *Blynk* genera los datos que se muestran en la figura 3.21.

```
FIRMWARE CONFIGURATION

#define BLYNK_TEMPLATE_ID
  "TMPLmqzuDUgW"
#define BLYNK_DEVICE_NAME "Thor 1"
#define BLYNK_AUTH_TOKEN
  "P5zHbN9QNKXlH8WoQUEqLVTKsW1Tu08X"
```

Figura 3.21: ID del dispositivo asignado por *Blynk*.

La configuración en *Blynk* asigna los siguientes elementos:

- *BLYNK_TEMPLATE_ID*: identificación del dispositivo asociado.
- *BLYNK_DEVICE_NAME*: nombre que le asigna el usuario al dispositivo que desea conectar.
- *BLYNK_AUTH_TOKEN*: código de autenticación del dispositivo.

Los datos anteriores son grabados en el dispositivo para que posteriormente el servidor pueda identificarlo. Una vez identificado, se procede a la transmisión y recepción de datos.

3.5.2 Comunicación ESP32-Arduino MEGA

La comunicación entre ambas placas se realizó a través de la conexión UART descrita en el apartado 3.1.5. Dicha comunicación se basa en el envío de comandos en formato *g-code* (presentado en capítulo 3.1.2) desde el módulo *NodeMCU32* hacia el *Arduino Mega 2560* acusando éste último la recepción de los mismos.

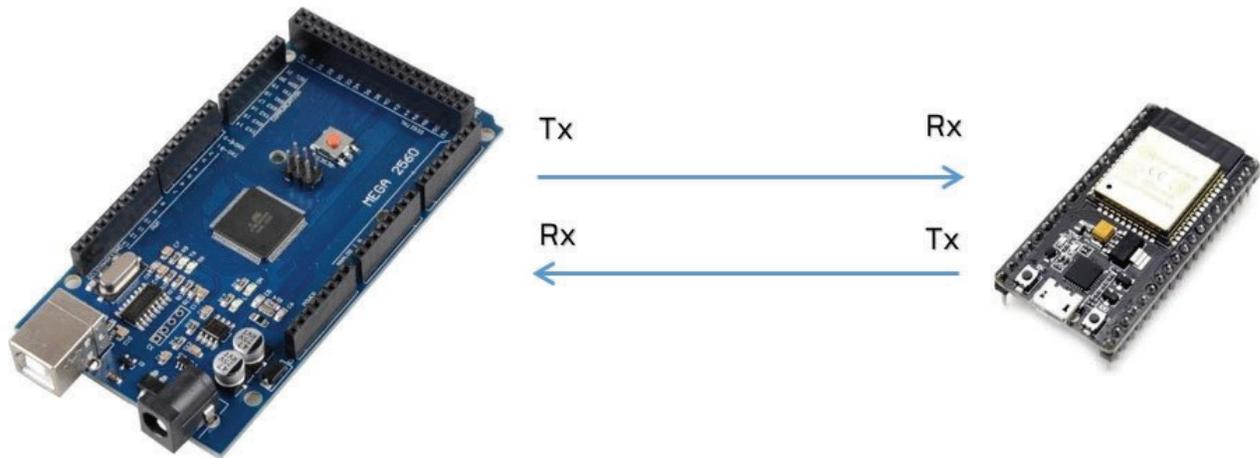


Figura 3.22: Comunicación serie entre *Arduino Mega* y el módulo *NodeMCU32*.

3.6 Programación de *Thor*

Como se mencionó en el capítulo 3.1.4 cada pin virtual definido fue asociado a un ícono o botón presente en el tablero de operación *Blynk*, de manera que al presionarlo se genera una instrucción particular (que fue programada dentro de la placa *NodeMCU32*) y que una vez recibida por el *Arduino Mega* generó las señales de salida para el movimiento de los motores.

Un ejemplo de instrucción a enviar debido a la lectura de un pin virtual podría ser la siguiente:

```

1 BLYNK_WRITE (V20)
2 {
3     byte state = param.asInt();
4     if (state == 0){
5         intruccion_A();
6     }
7     if (state == 1){
8         intruccion_B();
9     }

```

```
10 }
```

El fragmento de código anterior evalúa el pin virtual “V20” y de acuerdo al valor de tipo entero que se le asigne a la variable *state*, determinará la instrucción pertinente.

Para el desarrollo de código, las bibliotecas implementadas fueron:

- Conexión con el servidor Blynk: *BlynkEdgent.h*
- Comunicación serie: *SoftwareSerial.h*
- Iluminación: *Adafruit_NeoPixel*

Cuando se envían instrucciones *g-code* de pocas líneas, se ejecutan normalmente sin pérdida de comandos. Sin embargo, algunos movimientos del brazo requirieron varias líneas de código que fueron enviadas a gran velocidad por UART y que la placa responsable de procesarlas sólo pudo almacenar en memoria en forma temporal algunas de ellas. Por lo tanto, cuando se ejecutó una tarea predefinida del brazo robótico, debió establecerse la forma en la cual las distintas instrucciones del movimiento físico completo tenían que ejecutarse sin pérdida de datos. Para ello, implementamos el uso de *timers* (temporizadores). Los timers junto con el uso de interrupciones, permiten enviar datos cada “x” tiempo sin ocupar poder de procesamiento del microcontrolador. Esto conlleva una eficiencia mayor en el uso del sistema, dado que el microcontrolador no debe estar sensando continuamente la llegada de nuevos datos, liberándolo para otras tareas. De nuevo a nuestra aplicación, cada línea de código de instrucción *g-code* fue proporcionada en intervalos de 2 segundos logrando así suministrar el tiempo suficiente para lograr el movimiento completo de los motores para cada directiva y a su vez mantener en memoria la próxima instrucción de movimiento a ser ejecutada. Al igual que en el caso de los pines virtuales, se provee de un ejemplo de uso de temporizadores:

```
1 void funcion_ejemplo()  
2 {
```

```

3  switch (state)
4  {
5      case 0:
6          instruccion_A();
7          state += 1;
8          break;
9      case 1:
10         instruccion_B();
11         state += 1;
12         break;
13     case 2:
14         instruccion_C();
15         state += 1;
16         break;
17     case 3:
18         instruccion_D();
19         state += 1;
20         break;
21 }
22 }

```

El ejemplo anterior presenta una función llamada por el timer en la que evalúa la variable *state* según cada caso: suponiendo que inicializa en cero, entrará en el caso 0, ejecutará la instrucción A, incrementará su valor en uno y saldrá de la función debido al “break”. Pasado el tiempo configurado en el timer, *funcion_ejemplo* volverá a ser llamada, entrará en el caso 1 y ejecutará la instrucción B, incrementará nuevamente la variable *state* en uno y saldrá de la función. El proceso se repite hasta finalizar todas las instrucciones definidas en la función. Esta es la lógica planteada para poder enviar fragmentos de código de varias líneas evitando la pérdida de los mismos durante la ejecución.

Una segunda aplicación con timers fue la adquisición de temperatura por parte del sensor ubicado en el interior de la articulación 2, en la que cada 500 ms se extrae el valor de temperatura presente en la misma. Y una tercera se realizó para la lectura cada 10 ms

de los sensores ópticos, con el propósito de indicarle al usuario si cada articulación se encuentra en la posición *home*. En caso afirmativo, se genera la activación de las luces externas las cuales se corresponden con cada uno de los sensores ópticos en cuestión.

3.7 Tablero Blynk

El tablero de operación Blynk para el brazo robótico fue diseñado para dos plataformas, una para aplicación web y otra para dispositivos móviles. En ambos casos sus opciones de operación disponibles fueron: configuración de posición home, movimientos individuales de articulaciones, pausado de movimiento, apertura de gripper y dos movimientos predefinidos (pesada, agregado). En las figuras 3.23 y 3.24 se presentan las opciones disponibles del tablero.

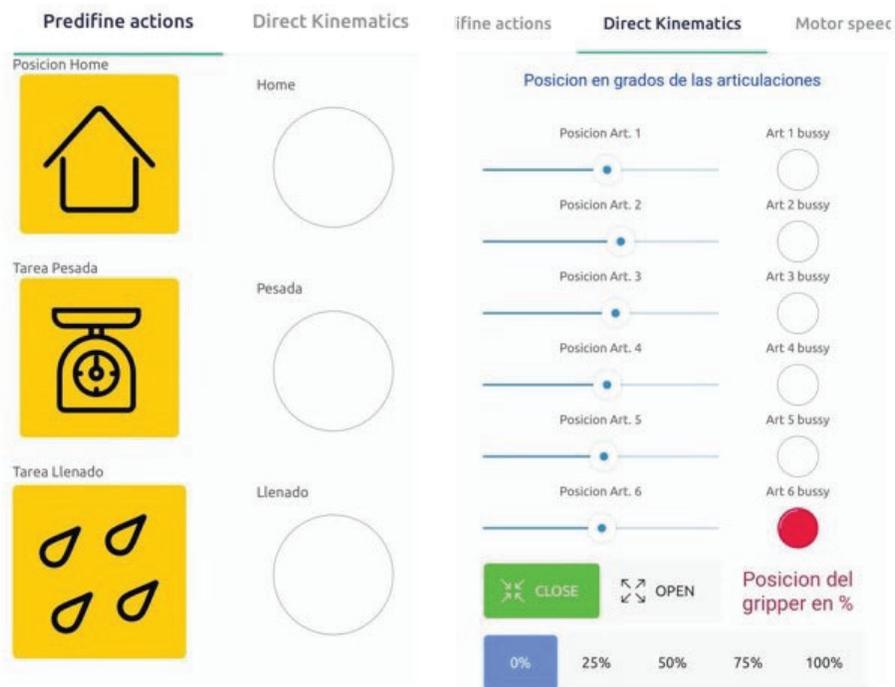


Figura 3.23: Tablero *Blynk* versión móvil. Izquierda: tareas predefinidas. Derecha: movimientos individuales de las articulaciones y gripper.

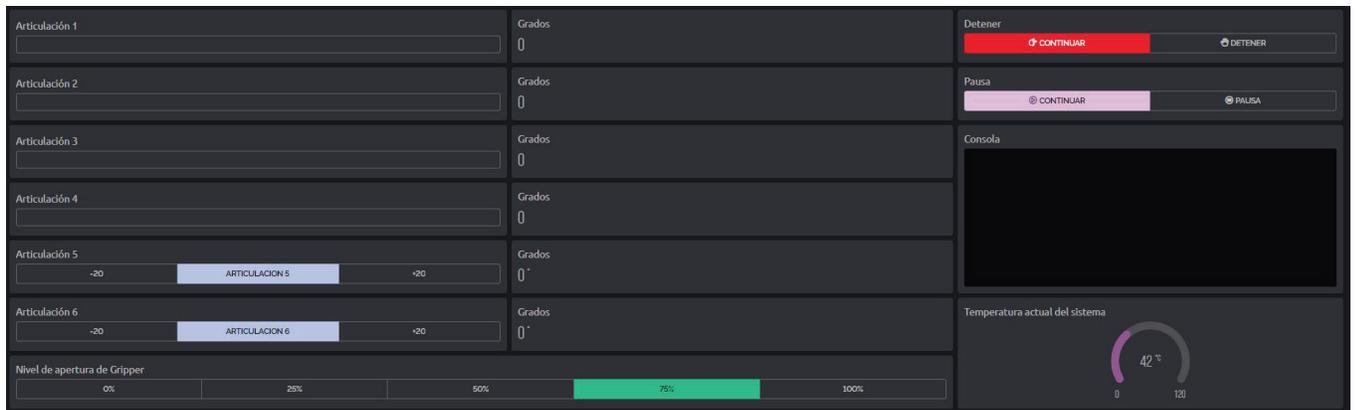


Figura 3.24: Tablero *Blynk* versión web. Izquierda: movimientos de articulaciones individuales. Derecha: Temperatura del sistema, consola para la visualización de instrucciones, funciones de parada y pausa.

Adicionalmente el tablero fue configurado para monitorear la temperatura mediante un gráfico del sistema (figura 3.6).

3.8 Diagrama de funcionamiento

El funcionamiento del brazo robótico se detalla en el esquema de la figura 3.25

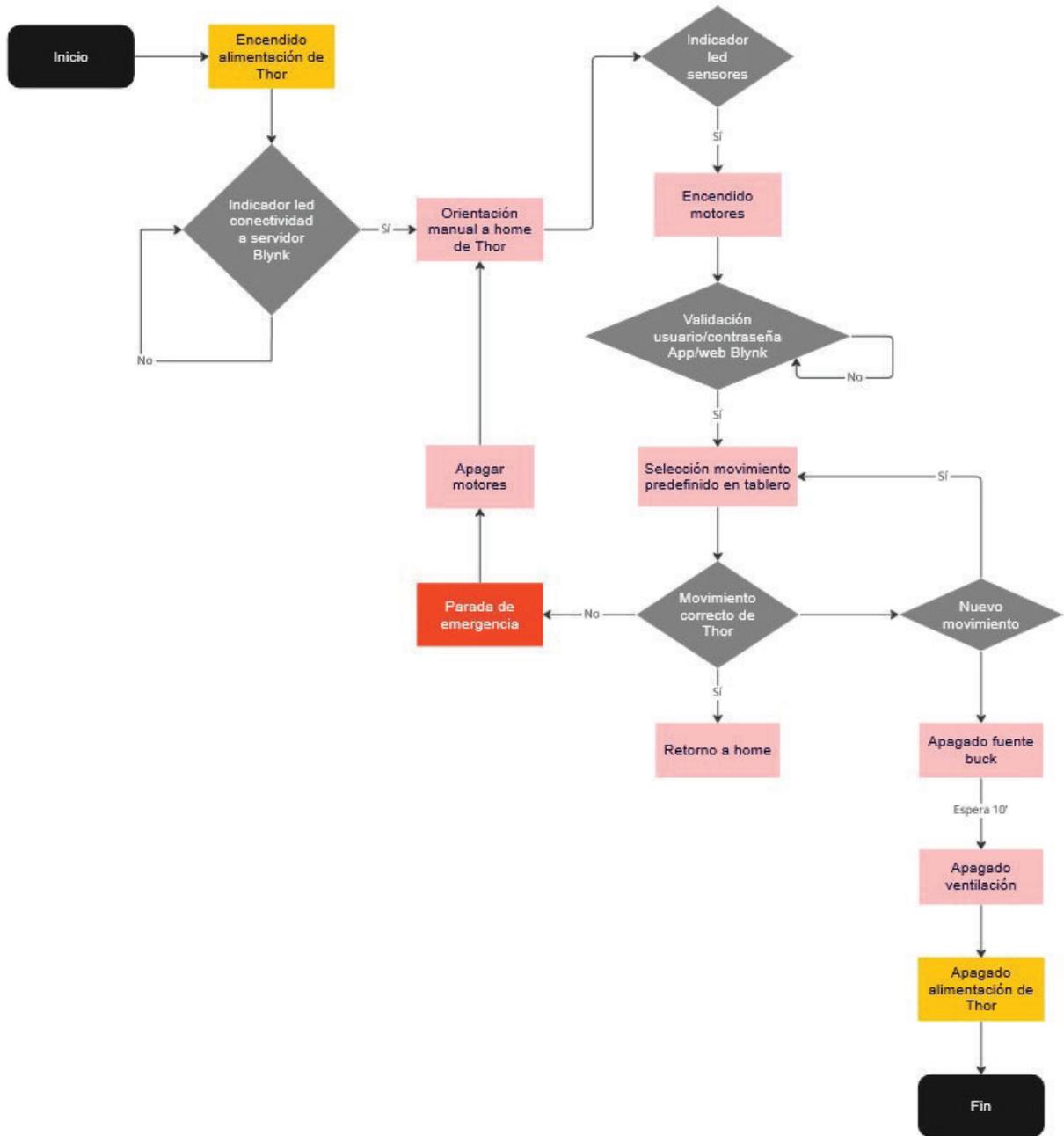


Figura 3.25: Funcionamiento de *Thor*.

3.9 Mejoras al modelo original

La forma de enviar instrucciones de movimiento al brazo robótico original se basaba en una comunicación de tipo cableada entre una PC y una placa *Arduino Mega*. Una de las mejoras incorporadas consistió en proveer, a través del servidor *Blynk* con el módulo *ESP32*, conectividad inalámbrica entre el usuario y *Thor*.

Otra de las mejoras que se incorporó fue el agregado de ventiladores en los laterales de la base de manera de favorecer el flujo de aire. Dado que el material de impresión es sensible a altas temperaturas, una vez energizados los motores debió considerarse cómo se comporta el calor producido por estos con el paso del tiempo. Para ello se diseñó un circuito de sensado en tiempo real de la temperatura con propósitos de supervisión. Dicho circuito consistió en un divisor resistivo en el cual la tensión medida se efectuó sobre un termistor de coeficiente negativo ⁸.

La relación entre la temperatura y la resistencia del termistor se caracteriza por tener un comportamiento exponencial (ecuación 1).

$$R_{ntc} = R_0 * e^{\beta(\frac{1}{T} - \frac{1}{T_0})} \quad (1)$$

Siendo:

- R_{ntc} : resistencia del termistor a la temperatura T (K).
- R_0 : resistencia del termistor a una temperatura de referencia, habitualmente a temperatura ambiente T_0 (K)
- T_0 : temperatura ambiente en Kelvin.
- T : temperatura deseada en Kelvin.
- β : constante de temperatura del material.

⁸NTC es un componente que posee la propiedad de disminuir su valor de resistencia en función del incremento de temperatura.

Debido a que necesitamos conocer la temperatura resultante en función del valor de resistencia R_{ntc} , se propone el siguiente circuito conocido como divisor resistivo:

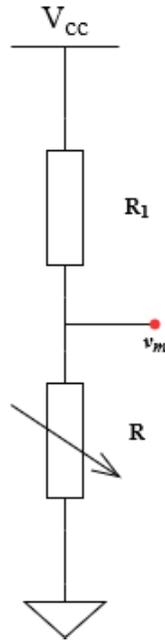


Figura 3.26: Circuito para sensado de temperatura de *Thor*.

Del circuito anterior resulta que la tensión v_m medida en R es:

$$v_m = \left(\frac{V_{cc}}{R_1 + R} \right) R \quad (2)$$

R_1 es una resistencia cuyo valor se define en términos de la resolución configurada en el ADC. Reordenando los términos, se tiene:

$$R = \left(\frac{v_m}{V_{cc} - v_m} \right) R_1 \quad (3)$$

De la ecuación 3, podemos calcular la resistencia del termistor en función de valores conocidos. Despejando la temperatura T de la ecuación 1:

$$T = \frac{1}{\left(\frac{\ln\left(\frac{R_{ntc}}{R_0}\right)}{\beta} + \frac{1}{T_0} \right)} \quad (4)$$

Para obtener la temperatura en grados celsius, en la ecuación 4 debe restarse el valor 273,15 equivalente a cero grados Kelvin.

Durante la etapa de ensamblado debieron realizarse algunas modificaciones a ciertas piezas debido a la dificultad para la obtención de bulonería y engranajes indicados en la lista de materiales original. Para el diseño y rediseño de piezas se utilizó el software **Fusion 360** de *Autodesk* dada la simplicidad de su interfaz y las innumerables herramientas disponibles para la creación de objetos 3D. Una de las partes modificadas se correspondió a la caja destinada al alojamiento de la electrónica (figura 3.27). Debido a la gran cantidad de cableado generado por los sensores ópticos, motores del brazo, los CNC shields, la placa de control, el PCB buck y el módulo NodeMCU32, debió ampliarse el volumen de la base y además se incorporaron cinco botones que cumplen las funciones de

- Encendido de fuente buck para alimentar el módulo NodeMCU32 y los dispositivos de 5V mencionados anteriormente.
- Activación de refrigeración a través de los ventiladores dispuestos en todo el brazo.
- Energizado de motores.
- Parada de emergencia.
- Botón para grabado de firmware de módulo NodeMCU32.



Figura 3.27: Rediseño de tapa que contiene la electrónica.

La necesidad de agregar el botón para grabado de firmware, se debió a que cada modificación producida en el código requiere una actualización de los cambios realizados y su posterior subida al módulo. Para ello, la placa NodeMCU32 debe configurarse en modo *flash* y

esto se logra presionando un botón denominado *boot* que se encuentra integrado en el módulo. Dado que el módulo se alojó dentro de la caja que contiene toda la electrónica, resultó necesario diseñar un acceso externo evitando de esta manera la apertura y cierre constante de la misma.

La tapa rediseñada debió dividirse en tres partes para poder ser impresa en equipos de tamaño estándar y luego ser unificada con soportes internos a fin de obtener un cuerpo único.

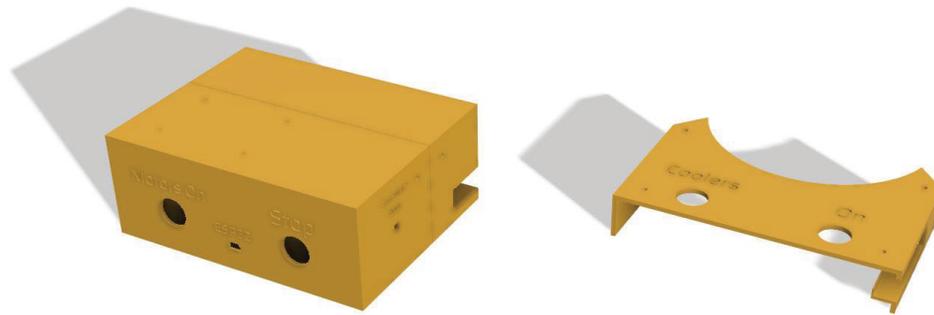


Figura 3.28: Tapa rediseñada con software **Fusion 360** de *Autodesk*.

También se adicionaron luces leds de distintos colores (figura 3.29), para indicarle al usuario los distintos estados de funcionamiento del brazo que abarcan la posición de *homming*, conectividad hacia el servidor Blynk, parada de emergencia, un estado de movimiento o de disponibilidad de uso.

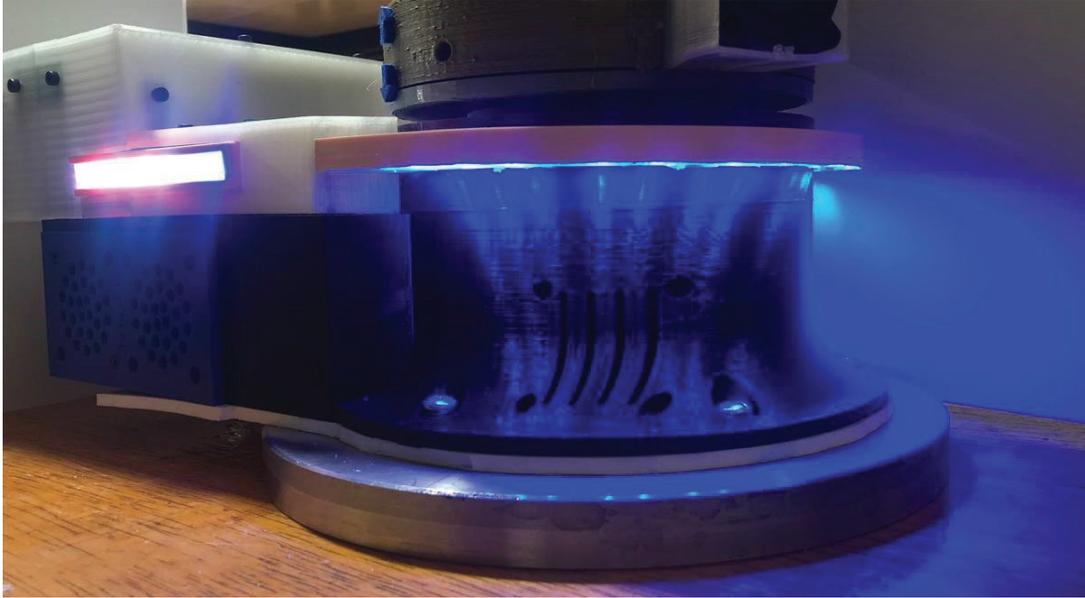


Figura 3.29: Luces leds indicadores de estado.

4 Resultados

4.1 Diseño

Comparado con métodos de fabricación industriales como el moldeo de plásticos, la impresión 3D se caracteriza por su bajo costo en la elaboración de piezas convirtiéndose en una excelente opción para realizar prototipos. La modificación sobre una pieza resulta simple dado que requiere solamente un software de edición 3D. Este beneficio permite realizar diversas configuraciones de relleno, nivel de capa de impresión, velocidad, temperatura, para obtener versiones preliminares hasta lograr la impresión de un diseño correcto y de buena calidad. La impresión de piezas que conforman el brazo *Thor* de este trabajo presentan una excelente resolución y terminación, lo cual favorece el ensamble del mismo.

La incorporación de un disco metálico a la base con un peso que supera al de *Thor* proporciona firmeza y estabilidad en todo el brazo cuando efectúa el movimiento de sus articulaciones.

Por último, *Thor* fue desarrollado de manera tal que todo el cableado de su electrónica viaja por el interior de su estructura. Estos cables fueron agrupados dentro de mallas de tipo helicoidales a fin de tener conexiones ordenadas y prolijas para facilitar el conexionado e identificación de los cables correspondientes a cada motor, sensor y ventilador.

4.2 Electrónica

La implementación de luces leds facilitan y mejoran la percepción sobre cual es el estado actual en el que se encuentra el brazo robótico. Es decir, se puede determinar si *Thor* está energizado, conectado al servidor de *Blynk*, disponible para un movimiento, en parada de emergencia o en posición de reposo. El resultado de tener indicadores de estado, evita por ejemplo que el usuario se acerque al brazo mientras éste se encuentre en movimiento, previniendo colisiones y/o accidentes.

Adicionalmente, los botones incorporados en la parte superior de la tapa añaden funciones

individuales al brazo, como la separación del sistema de ventilación respecto de la alimentación de la placa principal o el encendido/apagado de los motores. El hecho de mantener la refrigeración activa sin estar energizados los motores brinda la posibilidad de enfriar el sistema en menor tiempo. Además, no alimentar los motores favorece su vida útil y evita el calentamiento del brazo.

4.3 Aplicación de campo

La aplicación principal que el brazo robótico debió efectuar fue la pesada de un material en una balanza. Para el desarrollo de la tarea se dispuso de dos regiones, una correspondiente al equipo de pesado y otra en donde se encontraban los dos recipientes que debían ser manipulados (zona segura). En base a esto, se ha diseñado en el tablero blynk un botón para la tarea principal denominado *Pesar* y que estuvo compuesta de los siguientes movimientos:

1. Toma de recipiente vacío *A*.
2. Traslación hacia la zona de pesado.
3. Colocación en balanza.
4. Toma de recipiente *B*.
5. Traslación hacia la zona de pesado.
6. Vertido de contenido en recipiente *A*.
7. Apartado de recipiente *B* hacia zona segura.
8. Posicionado del brazo a *home*.

La figura 4.1 resume el proceso completo descrito anteriormente.

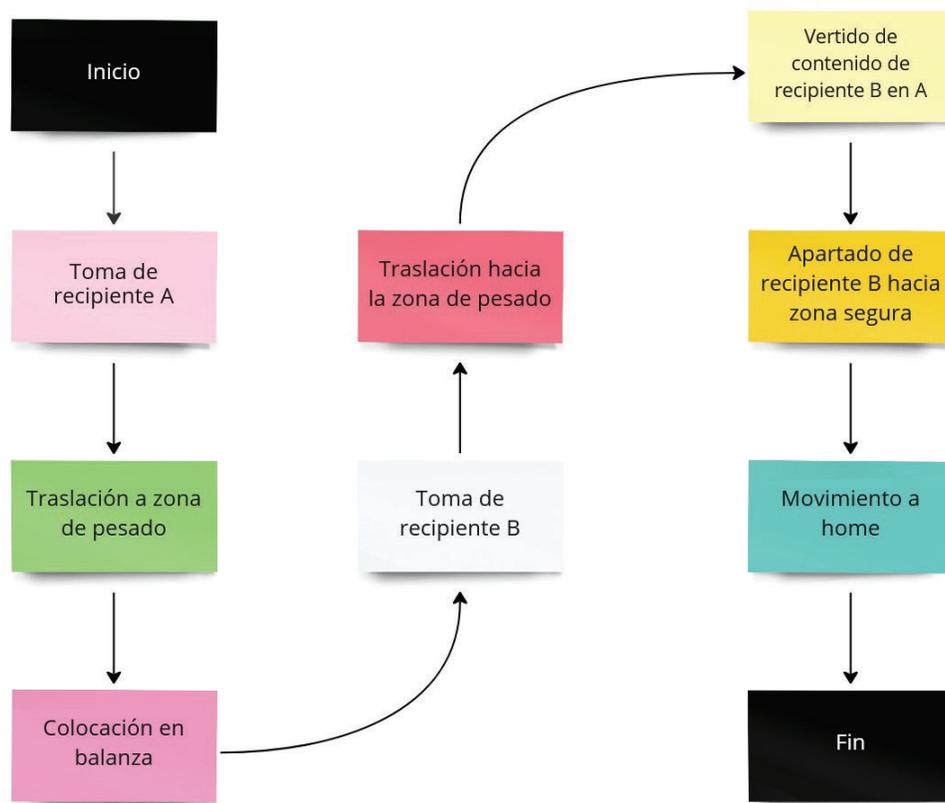


Figura 4.1: Descripción del movimiento de la tarea *pesada* a efectuar por *Thor*.

Adicionalmente se definió una tarea secundaria de llenado que consistió en el agregado de un material presente en el recipiente C hacia el recipiente A dispuesto en la balanza. El propósito de esta segunda tarea es contemplar el caso en el que se requiera añadir mayor cantidad de un material o sustancia. De esta manera, se evita efectuar todo el conjunto de movimientos involucrados en la tarea principal presentada previamente. El botón correspondiente a esta acción se denominó *Agregado* y sus movimientos se detallan en la figura 4.2.

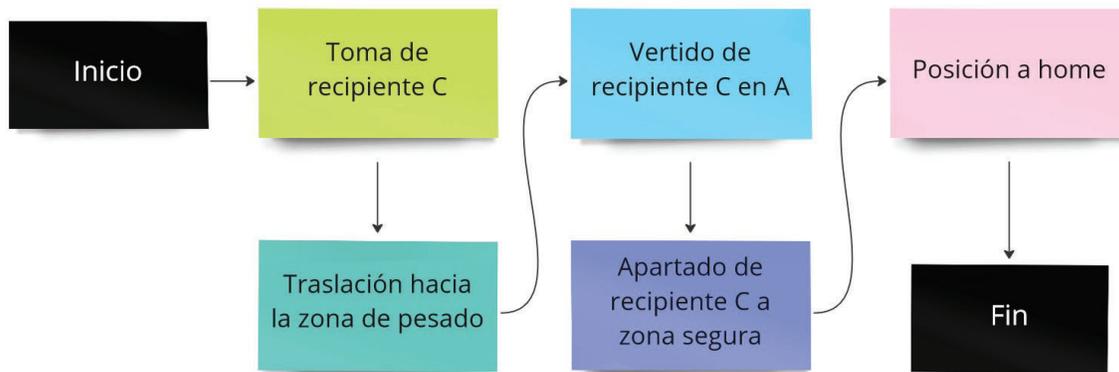


Figura 4.2: Descripción del movimiento de la tarea *agregado* que debe efectuar *Thor*.

4.4 Movimientos

Al ejecutar las tareas predefinidas, el brazo realiza cada uno de los movimientos siguiendo la secuencia de pasos definidos por el esquema 4.1 y/o 4.2 logrando efectuar el pesado y/o agregado de la sustancia correctamente. Durante la tarea, el desplazamiento de las articulaciones presentan un movimiento fluido y natural similar al que puede efectuar un brazo humano. Estas características son atribuibles a la configuración de pasos en los drivers de los motores que le otorgan un giro con una mayor resolución. Estas cualidades brindan flexibilidad al sistema ya que permiten la programación de otras tareas mas complejas que *Thor* puede ejercer. En las figuras 4.3 y 4.4 se presentan algunas pruebas de movimientos realizadas por el brazo.

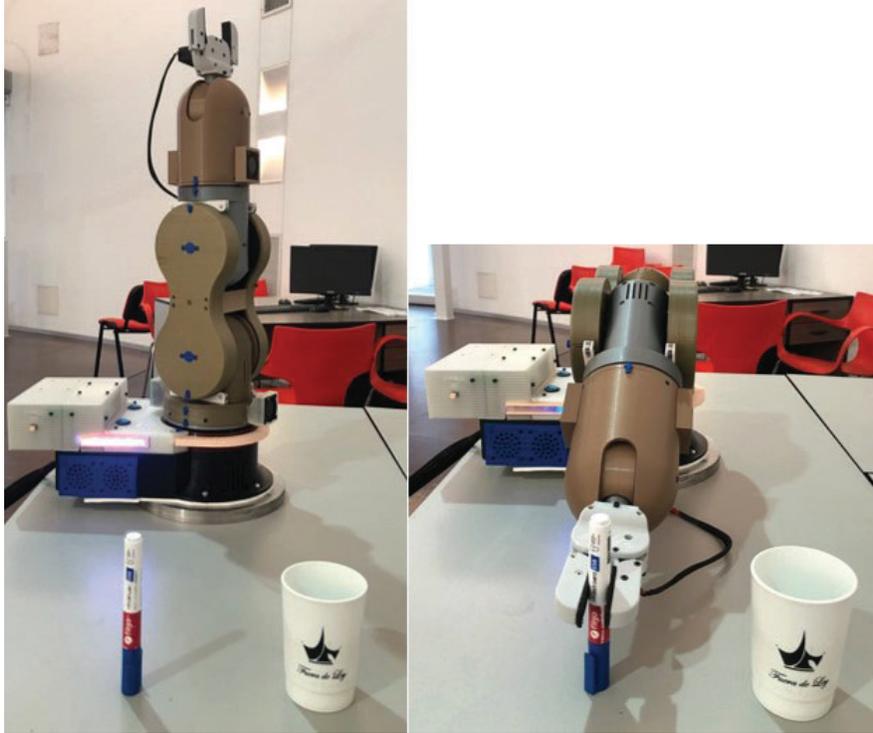


Figura 4.3: Primera prueba de movimientos.

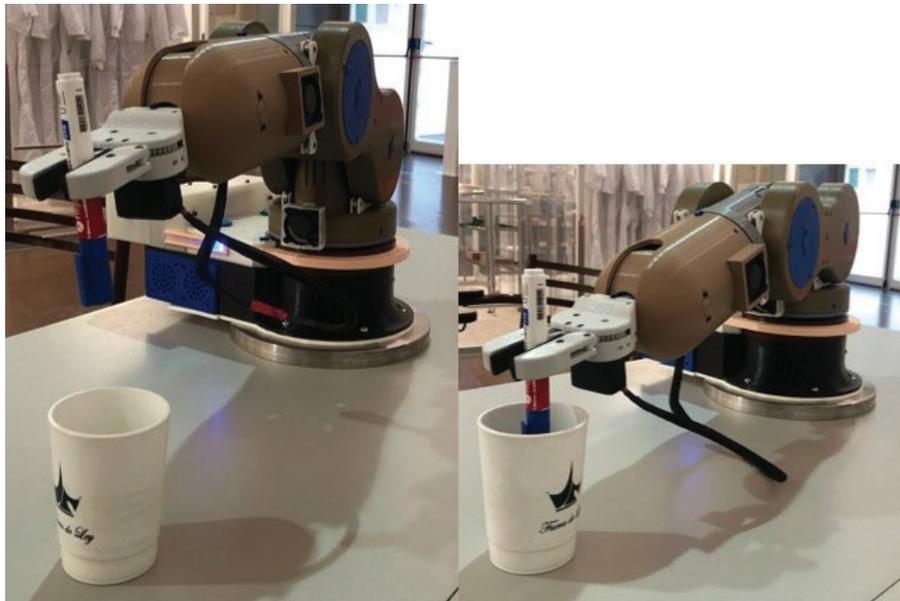


Figura 4.4: Segunda prueba de movimientos

Una vez efectuados los movimientos descritos anteriormente se procedió a modificar el escenario incorporando una balanza y un recipiente de manera de acondicionar la futura aplicación real del brazo. En la figura 4.5 se visualizan tres pruebas que consistieron en trasladar el recipiente hacia la balanza.

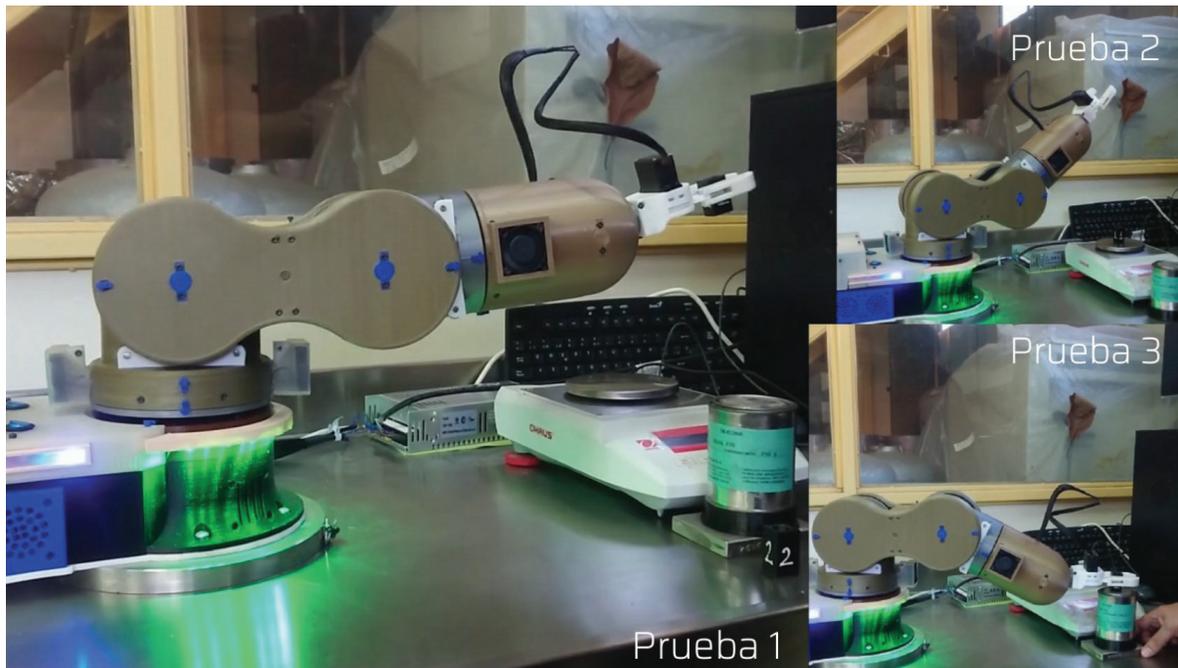


Figura 4.5: Primeras pruebas de pesada

El video de la la prueba completa se encuentra en el siguiente link https://drive.google.com/file/d/1gDqoqTywuE5-XUGu_8fyrazTnZQygEd-/view?usp=share_link/.

4.5 Conectividad

Al realizar las pruebas de conectividad se obtiene que la distancia máxima de conexión del brazo robótico respecto al router debe ser menor a 6 metros⁹. Sin embargo con el agregado de un repetidor se logra ampliar la cobertura.

⁹Pruebas realizadas con un router marca Cisco modelo WRT54G2 y cuya potencia de salida de RF es de 18 dbm

Una observación a comentar es que la velocidad del servicio de internet requerida para el uso de *Blynk* (ya sea con teléfono celular o computadora) no es un limitante debido a las conexiones de banda ancha ofrecidas actualmente en el mercado.

En términos de interacción entre el usuario y la aplicación *Blynk* (móvil/pc), no se observan retrasos de tiempo significativos entre la activación de una acción presente en el tablero (figura 3.23) y la ejecución física de la misma por parte de *Thor*.

El envío de datos mediante conexión UART (apartado 3.1.5) entre el módulo *ESP32* y la placa *Arduino Mega* resulta satisfactoria dado que la ejecución de la tarea se realiza en forma completa.

4.6 Efecto final

La elección del efector final (apartado 3.4) presenta los siguientes resultados: los límites, en términos de agarre, comprenden un área cuadrada de 5 cm de lado o un diámetro de 5 cm en caso de superficies cilíndricas. Según pruebas realizadas, el peso máximo que puede sujetar es de 750 g. Respecto a los objetos que el gripper puede manipular, se comprueba que no resulta recomendable el uso de materiales que sean flexibles o blandos debido a la deformación como consecuencia de la presión ejercida en las paredes de estos.

4.7 Temperatura del sistema

Se realizaron ensayos de temperatura sobre impresiones de prueba con el propósito de observar su rigidez estructural a medida que se incrementa la temperatura. Para ello utilizamos rectángulos impresos en distintos espesores que fueron sometidos a un flujo de aire caliente de distintas temperaturas durante un lapso de tiempo de dos minutos. Este tiempo se consideró como el necesario para ejecutar el movimiento de mayor duración en la secuencia de la tarea *pesada* dado que cuando los motores comienzan a girar el consumo de corriente se eleva y en consecuencia también la temperatura. El resultado obtenido fue el siguiente:

Espesor (mm)	Temperatura (°C)			
	40	50	60	70
1	✓	✓	✗	✗
1,5	✓	✓	✓	✗
2	✓	✓	✓	✗
2,5	✓	✓	✓	✗
3	✓	✓	✓	✗

Tabla 3: Ensayo de temperatura en impresiones de prueba.

De la tabla 3 se observa que una impresión de 1 mm de espesor puede tolerar temperaturas por debajo de los 60°C y para espesores mayores a este, la máxima temperatura que puede alcanzar son 70°C. Por lo tanto se considera un valor crítico una temperatura de 60°C.

La incorporación del sistema de ventilación adicional tanto en el interior de la articulación 2 como en los laterales externos de la misma, mantienen el brazo robótico por debajo de los 54°C. Logrando de esta manera estar por debajo del valor crítico mencionado anteriormente.

5 Conclusiones

Las conclusiones del sistema se presentan realizando un análisis respecto de su hardware, software y ejecución de tareas. Principalmente debemos recordar que este brazo es de código libre, siendo esta característica el motivo de su incorporación como elemento de automatización para este trabajo final.

5.1 Hardware

Como se mencionó en el apartado 3.3.2, la estructura de *Thor* está realizada mediante impresión 3D logrando así una reducción de costos y tiempo en su construcción. Sin embargo, se puede considerar el mecanizado de su estructura con otro tipo de material. El cableado interno puede reubicarse en el exterior tal cual ocurre en los brazos robóticos de tipo industrial favoreciendo aún más la ventilación interna de la estructura interna del brazo. Por ejemplo, en caso de existir espacio interno donde se encuentran los motores pueden incorporarse a éstos disipadores de aluminio logrando un descenso mayor de temperatura del sistema.

El consumo de corriente generado por la ejecución de las tareas, que involucran el movimiento de todos los motores en simultáneo, es de 15 A. Para este valor notamos que la fuente de alimentación utilizada en *Thor* está sobredimensionada ya que la potencia que puede entregar supera ampliamente el consumo requerido.

Los ensayos efectuados para las tareas, demuestran que *Thor* puede manipular recipientes con contenido sin mayores dificultades e inclusive puede elevar objetos de mayor peso que se encuentren cerca del máximo descripto. Como se mencionó anteriormente, la precisión de sus motores permite que este brazo pueda realizar tareas de mayor complejidad en comparación con la que fue definida.

El agregado de los botones funcionales ubicados en la tapa de la base del brazo contribuyeron a la reducción de consumo de energía del brazo ya que existe una separación entre la alimentación de las placas y los motores, siendo estos últimos los dispositivos de mayor

consumo energético. A su vez, esta independencia permite mantener activa la ventilación del sistema indistintamente del estado de activación de los motores.

La incorporación de los indicadores lumínicos aportan al usuario una mejor comprensión sobre el estado actual en el que se encuentra *Thor*. Un caso particular, son las luces que se corresponden al movimiento. Éstas evitan que puedan ocurrir accidentes dada la proximidad del usuario con el brazo. De esta manera, se logra un uso más seguro protegiendo tanto a la persona que lo opera como así también al brazo robótico.

5.2 Software

El uso de bibliotecas pertenecientes a *Blynk* simplifican la configuración correspondiente a la conexión entre servidor y la placa ESP32. En relación al tablero, el intercambio de datos entre los botones presentes en este y la lógica desarrollada para cada uno de ellos, resulta óptima gracias a la abundante documentación provista por *Blynk* en su sitio. Por otro lado, la aplicación móvil brinda una forma sencilla de reconfigurar la red a la que se necesita conectar el sistema en caso requerir reubicarlo hacia otro sitio.

Por otra parte, se destaca el uso en simultáneo de la aplicación *Blynk* que puede ejecutarse ya sea desde un teléfono móvil y/o a través del panel web desde una PC. Por lo tanto, al brindar la posibilidad de elegir entre dichos dispositivos evita una dependencia particular para acceder al sistema. Esto favorece la inclusión de usuarios en caso de que éstos no cuenten con una computadora, pero sí con un equipo móvil o viceversa.

Las opciones de diseño para el tablero son variadas con gran cantidad de botones e íconos disponibles para utilizar permitiendo desarrollar diversas opciones para el usuario.

Cabe destacar que la conexión del módulo *ESP32* y la placa *Arduino Mega* mediante comunicación serie (utilizando una UART) no presenta problemas de envío ni de recepción de datos debido a las mejoras implementadas. Ninguno de los comandos *g-code* enviados se han perdido como así tampoco se han generado errores durante la comunicación, lo que contribuyó a la correcta interpretación y ejecución por parte de GRBL, logrando realizar en forma correcta las tareas predefinidas.

5.3 Ejecución de tarea

Cada uno de los movimientos involucrados se corresponde con la secuencia detallada previamente en la figura 4.1 y que se encuentra en el tablero web como se observa en 5.1

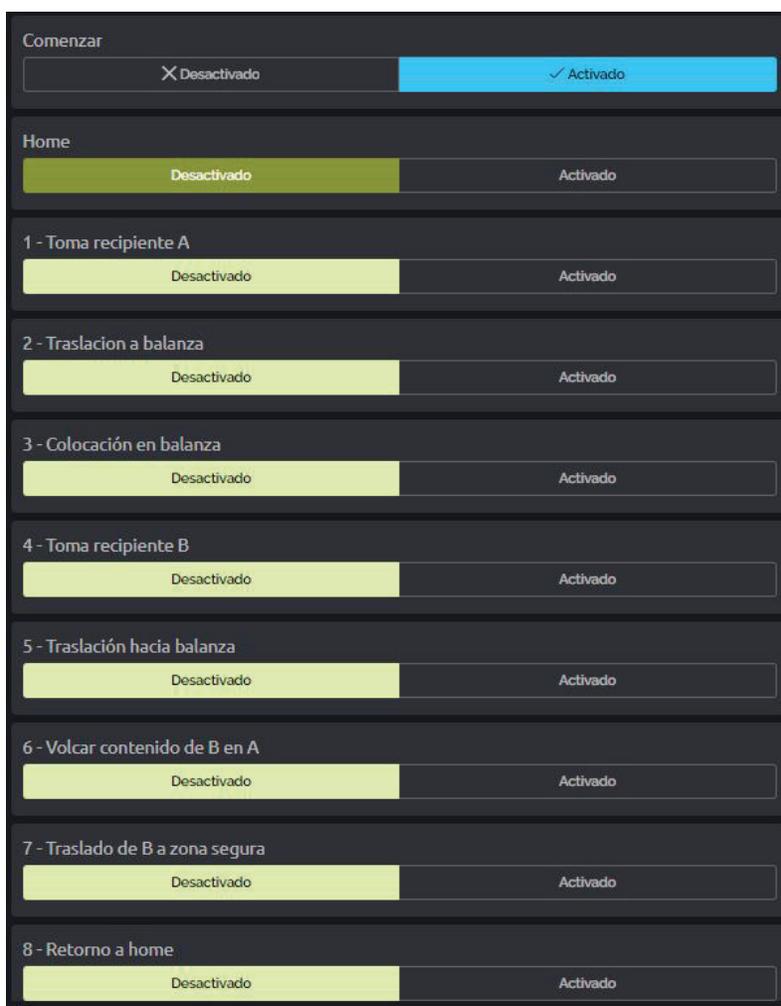


Figura 5.1: Tablero web para la ejecución de la tarea pesada

La ejecución de la tarea de pesada efectuada por *Thor* se muestra en la figura 5.2

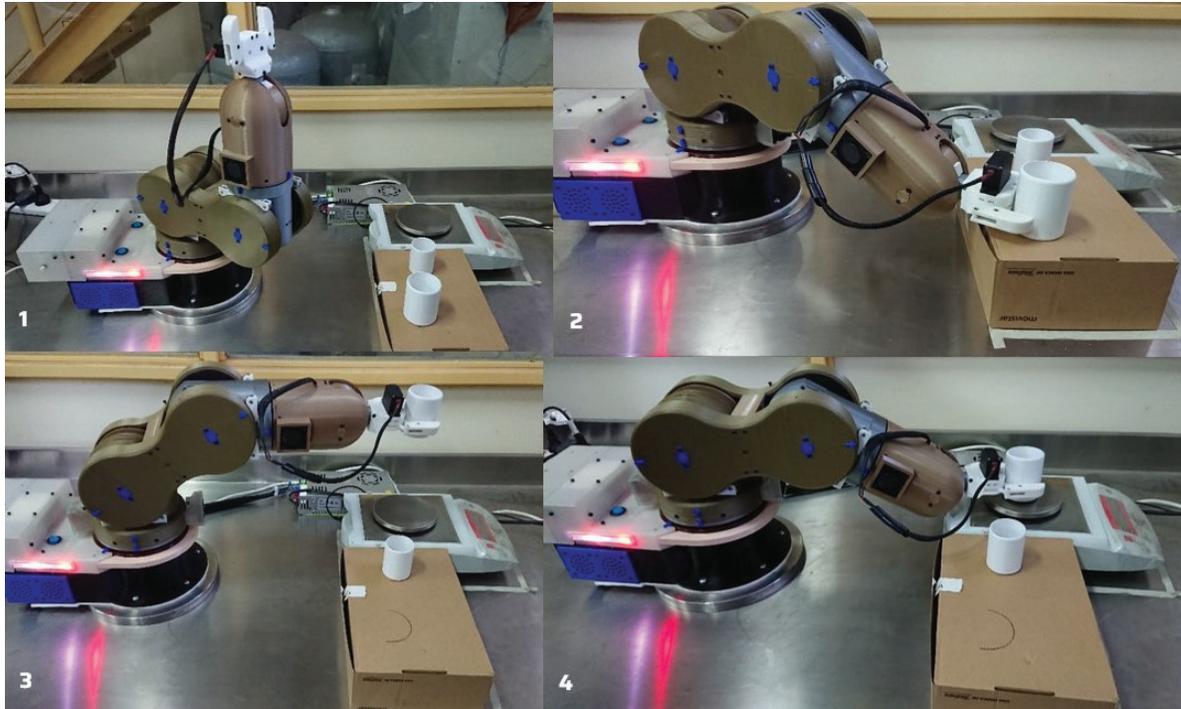


Figura 5.2: 1 - Inicio, 2 - Toma de recipiente A, 3 - Traslación a zona de pesado y 4 - Colocación en balanza

Una vez depositado el recipiente vacío A en la balanza, se procede a la búsqueda, toma y traslado del recipiente B hacia la zona de pesado como se muestra en la figura 5.3

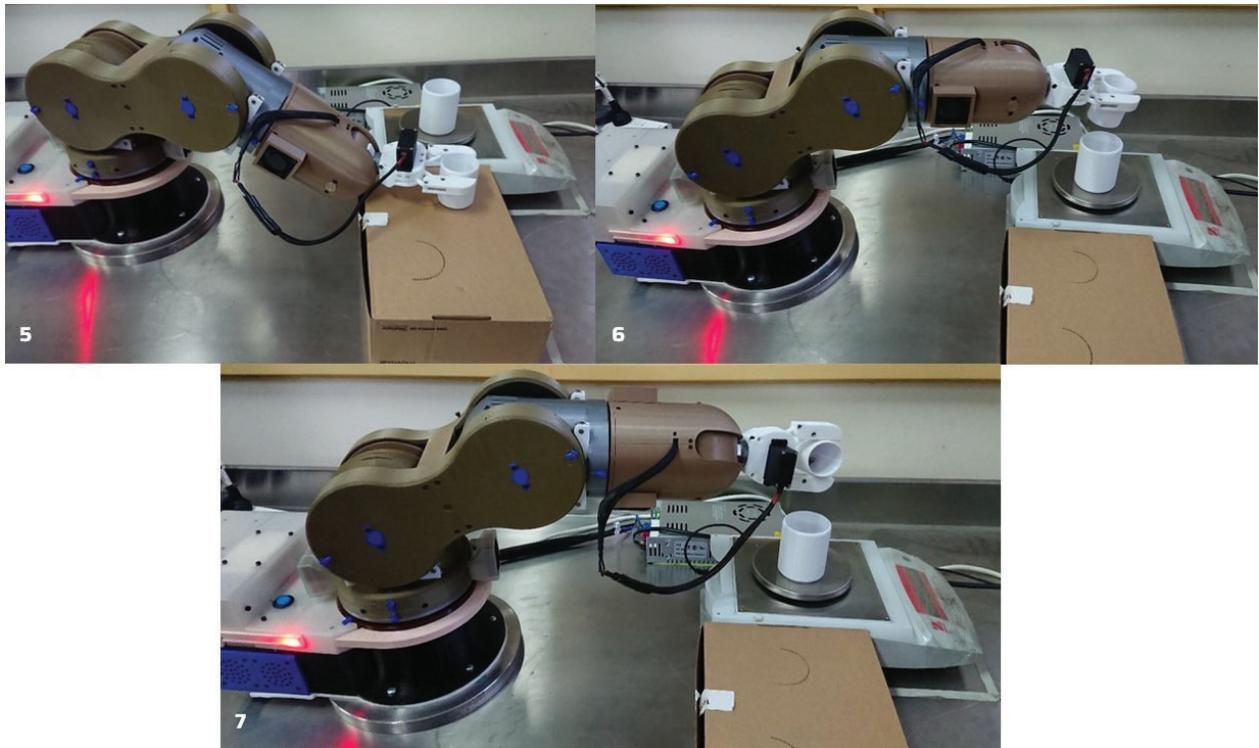


Figura 5.3: 5 - Toma de recipiente *B*. 6 - Traslación hacia zona de pesado. 7 - Vertido de contenido de recipiente *B* en *A*

Luego que el contenido del recipiente *B* es vertido en el recipiente *A*, se ejecutan los últimos movimientos que comprenden el alejamiento de *B* hacia zona segura y la restitución del brazo a su posición de reposo como se aprecia en la figura 5.4.

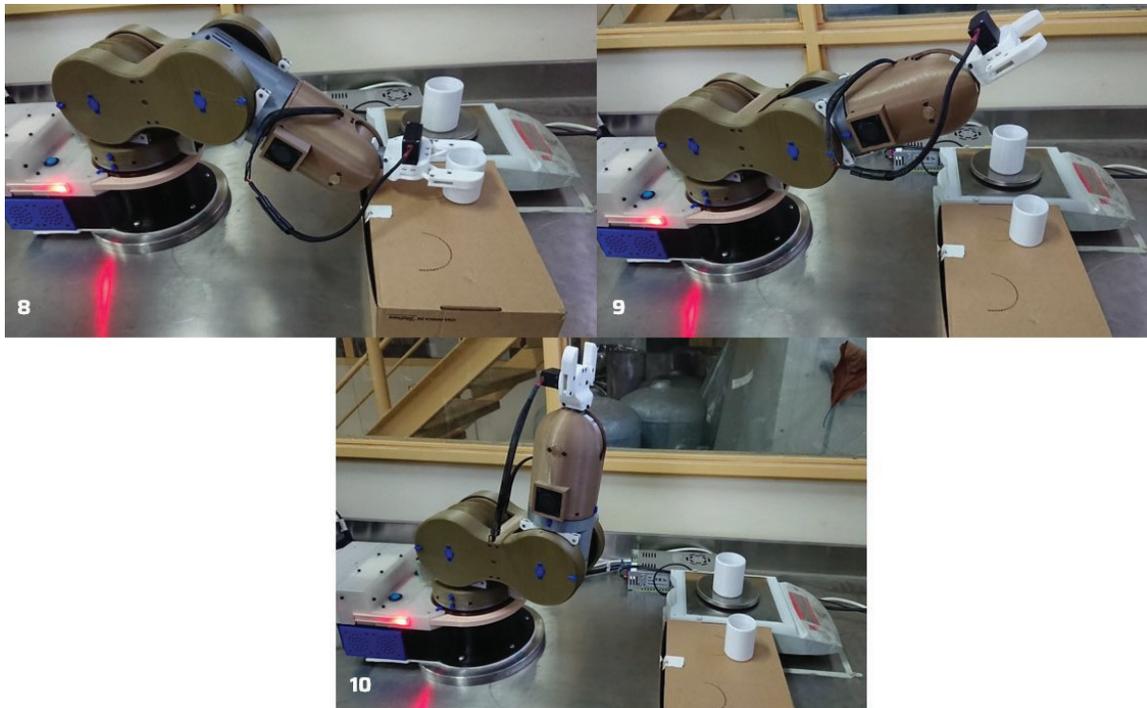


Figura 5.4: 8 - Apartado de recipiente *B* hacia zona segura. 9 - Movimiento a *home*. 10 - Fin

Durante la ejecución de la tarea de pesada se observó que al ejecutar movimientos de los motores en forma grupal se obtiene un tiempo de ejecución inferior en comparación con la activación de motores en forma individual. Además este movimiento conjunto genera una fluidez en el desplazamiento similar al que puede realizar el brazo de un humano. Sin embargo, la necesidad de activar varios motores simultáneamente implica un incremento en el consumo de corriente y como consecuencia, una elevación en la temperatura del sistema.

Respecto a la temperatura vale mencionar que durante el proceso de ejecución de la pesada, el sistema de ventilación mantuvo la temperatura por debajo del valor crítico, definido en la tabla 3, refrigerando el brazo sin mayores dificultades.

La ejecución de la tarea de pesada se puede observar en el siguiente link: https://drive.google.com/file/d/1fQGTnvZEh9wFFsPj2x8P2bu5EuZ_60Ao/view?usp=sharing.

5.4 Conclusión final

Dado que *Thor* es un brazo robótico de 6 grados de libertad y además de código libre, se presenta como una excelente opción para incorporarlo como principal herramienta de automatización de laboratorios o inclusive integrarlo hacia otros ámbitos de trabajo. Este brazo presenta un costo competitivo comparado con otros modelos que se ofrecen en el mercado para realizar tareas similares.

Por otro lado, su conectividad inalámbrica brinda al usuario mayores opciones para su accionamiento y/o monitoreo.

La integración de estas tecnologías establecen actuales estándares para el desarrollo de las prácticas de laboratorios, y además la necesidad de incorporar nuevos conocimientos a los usuarios para su utilización.

Este sistema establece las bases que permitirán a futuro tener laboratorios más eficientes, productivos y seguros.

Existen también varias mejoras que pueden incorporarse y que se dejan planteadas en el apartado 5.5, en caso de explorar investigaciones futuras.

5.5 Mejoras futuras

En este apartado se presentan algunas ideas de mejoras futuras que pueden ser integradas al sistema:

- Incorporación de una cámara como herramienta de visualización para el usuario de manera de permitir monitorear los movimientos del brazo.
- Desarrollo de un sistema de conexión rápida de efectores finales para proveer distintas herramientas que otorguen una mayor funcionalidad de aplicación.
- Desarrollo y aplicación de un sistema de control del tipo realimentado a través de encoders o sensores de manera de obtener las posiciones angulares de los motores.
- Opción de control de movimiento mediante joystick.

- Reemplazo de impresión 3D por mecanización en metal u otro material de mayor rigidez.
- Sustitución de lenguaje GRBL por un sistema operativo robótico más avanzado como por ejemplo *ROS*.
- Configuración y/o desarrollo de un servidor propio o empleo de otro de características de código libre.
- Incorporación de AI de manera que el brazo integre la toma rápida de decisiones en forma independiente incrementando la calidad de los procesos.
- Sustitución de la placa *Arduino Mega* por otra de mejores prestaciones como por ejemplo *Raspberry Pi* o aquellas que presenten microcontrolador del tipo *Cortex*.

Anexo

A Conectividad con la aplicación Blynk

Antes de comenzar la operación del brazo robótico se requiere contar con la aplicación de Blynk instalada en teléfonos o tablets que posean tanto *Android* como *iOS*. Otra manera de acceso puede realizarse a través del sitio web <https://blynk.cloud/>. Indistintamente del canal de acceso, el o los administradores, otorgarán los permisos de invitación a través de correo electrónico.

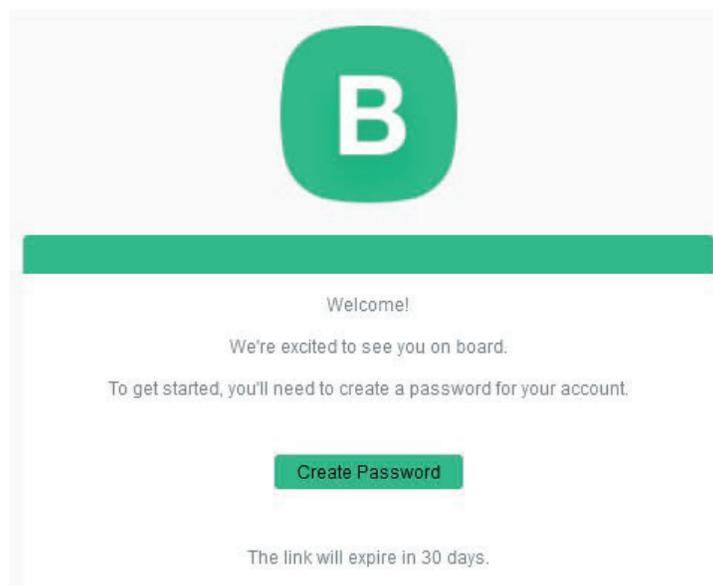


Figura A.1: Correo electrónico con link de acceso a *Blynk*

El nuevo usuario deberá generar una clave que le facilitará el acceso al tablero de *Blynk*.

Create Password

Create a password which is hard to guess.

PASSWORD

🔒
👁️

A password is required to log in

- Make it at least 8 symbols long

Other tips:

- Use uncommon words
- Use non-standard uPPercasIng
- Use creatif spelllllllling
- Use non-obvi0u\$ number\$ & symbo1s

I agree to Terms and Conditions and accept Privacy Policy

Continue

[I already have an account. Log in.](#)

Figura A.2: Generación de clave de acceso

Efectuado lo anterior, deberá seleccionarse el dispositivo (debajo de la opción *device name*) y posteriormente se presentará a la derecha de la pantalla el tablero de usuario e información adicional referido al dispositivo.

The screenshot shows the 'My Devices' page in the UNSAM Remote Laboratory system. On the left, a sidebar contains a search icon and a list of device categories: 'DEVICES', 'My Devices' (with a count of 1), and 'All' (with a count of 1). The main area is titled 'My Devices' and shows '1 Device'. Below this, a table lists the device details:

Device name	Device owner	Status	Device model	Actions
Thor 1 D3387	Robotizacion de laboratorios	Offline		

At the bottom right of the interface, there is a footer that reads 'Region: ny3 Privacy Policy'.

Figura A.3: Selección de dispositivo

Dependiendo si el acceso es a través de la web o con la aplicación de *Blynk*, existirán

tableros diferentes ya que, como se mencionó anteriormente, los diseños no tienen ninguna relación entre sí.

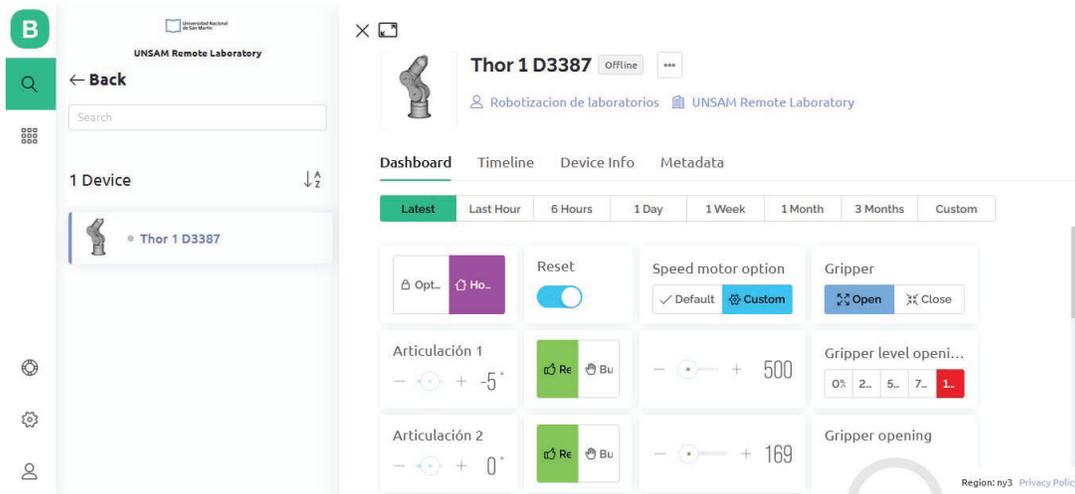


Figura A.4: Tablero web de usuario

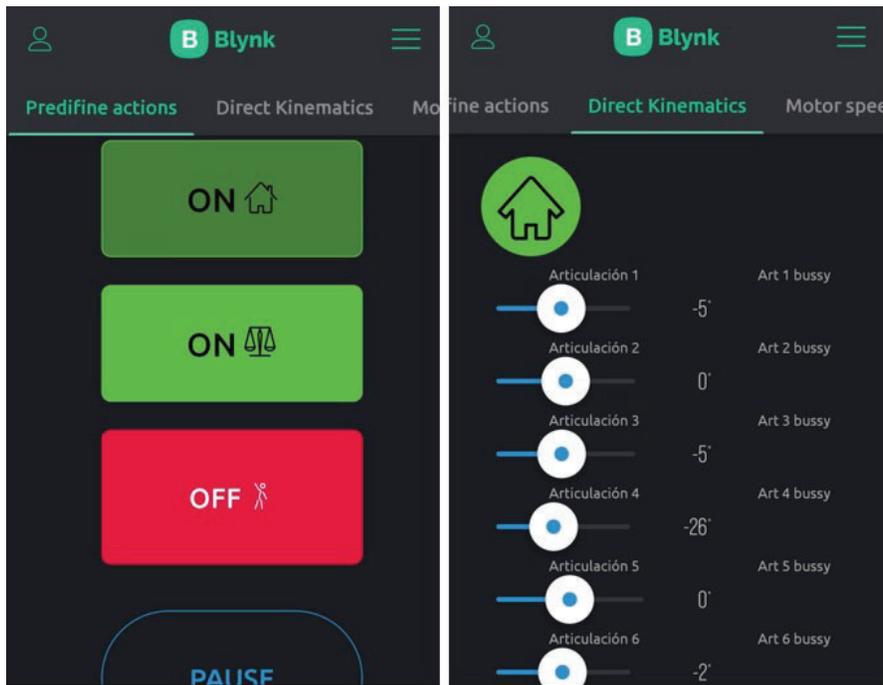


Figura A.5: Tablero de usuario desde la aplicación

B Especificaciones técnicas

Model	Step Angle (deg)	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm ²)	Lead Wire (No.)	Motor Weight (g)
17HS4401	1.8	40	1.7	1.5	2.8	40	2.2	54	4	280
KS42STH34-1504A	1.8	34	1.5	1.5	2.8	22	-	-	4	220
17HS13-0404S-PG5	1.8*	34*	-	27	28.5	22*	-	-	4	-

Tabla B.1: Motores

Model	Operating voltage (logic)	Operating voltage (vmot)	Max current (A)	Step resolution
DVR8825	3 – 5.25V	12 – 24V	2.5	1/2°/1/4°/1/8°/1/16°/1/32°

Tabla B.2: Driver

Model	Operating voltage	Running current (A)	Stall current (A)	Stall torque (kg.f.cm)	Weight (g)
MG996R	4.8 – 7.2V	0.5 – 0.9(6V)	2.5 (6V)	9.4 (4.8V) 11 (6V)	55

Tabla B.3: Servo

Model	Operating voltage	Axis support	Ends stops (per axis)	Drivers supported	Control software
CNC shield V3	12 – 36V	4 (X, Y, Z, A)	2	DVR8825 A4988	GRBL

Tabla B.4: CNC shield

Operating voltage (VCA)	Output voltage (VDC)	Output current (A)	Power (W)	Operating frequency (Hz)	Weight (g)
110 – 220	12	40	480	50 – 60	880

Tabla B.5: Fuente switching

Model	Clock speed (MHz)	Operating voltage (V)	Flash memory/SRAM	Digital I/O pin	WiFi Built-in
NodeMCU ESP32	240	3.3 (5 via microUSB)	4MB/512KB	11	802.11 b/g/n

Tabla B.6: Módulo ESP32

Microcontroller	Clock speed (MHz)	Input voltage (V)	Flash memory/SRAM	Digital I/O pin	Weigth (g)
ATmega2560	16	7 – 12	256KB/8KB	54	37

Tabla B.7: Arduino Mega 2560

C G-Code soportados en V0.9

```
1      G0, G1: Linear Motions
2      G2, G3: Arc and Helical Motions
3      G4: Dwell
4      G10 L2, G10 L20: Set Work Coordinate Offsets
5      G17, G18, G19: Plane Selection
6      G20, G21: Units
7      G28, G30: Go to Pre-Defined Position
8      G28.1, G30.1: Set Pre-Defined Position
9      G38.2: Probing
10     G38.3, G38.4, G38.5: Probing
11     G40: Cutter Radius Compensation Modes OFF (Only)
12     G43.1, G49: Dynamic Tool Length Offsets
13     G53: Move in Absolute Coordinates
14     G54, G55, G56, G57, G58, G59: Work Coordinate Systems
15     G61: Path Control Modes
16     G80: Motion Mode Cancel
17     G90, G91: Distance Modes
18     G91.1: Arc IJK Distance Modes
19     G92: Coordinate Offset
20     G92.1: Clear Coordinate System Offsets
21     G93, G94: Feedrate Modes
22     M0, M2, M30: Program Pause and End
23     M3, M4, M5: Spindle Control
24     M7* , M8, M9: Coolant Control
25
```

Repositorio: <https://github.com/grbl/grbl>

D Documentación del proyecto

D.1 Acta de proyecto

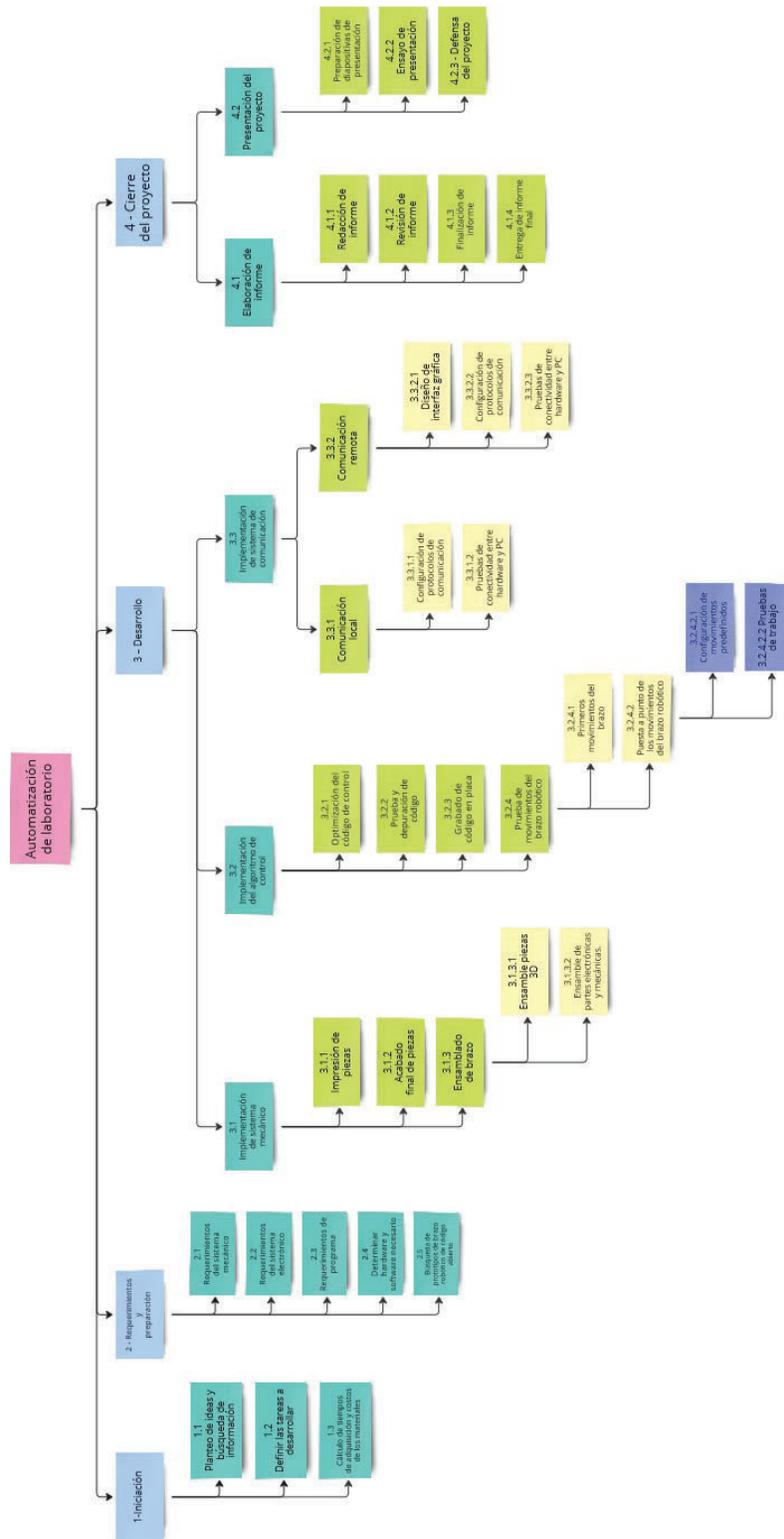


Acta Constitutiva

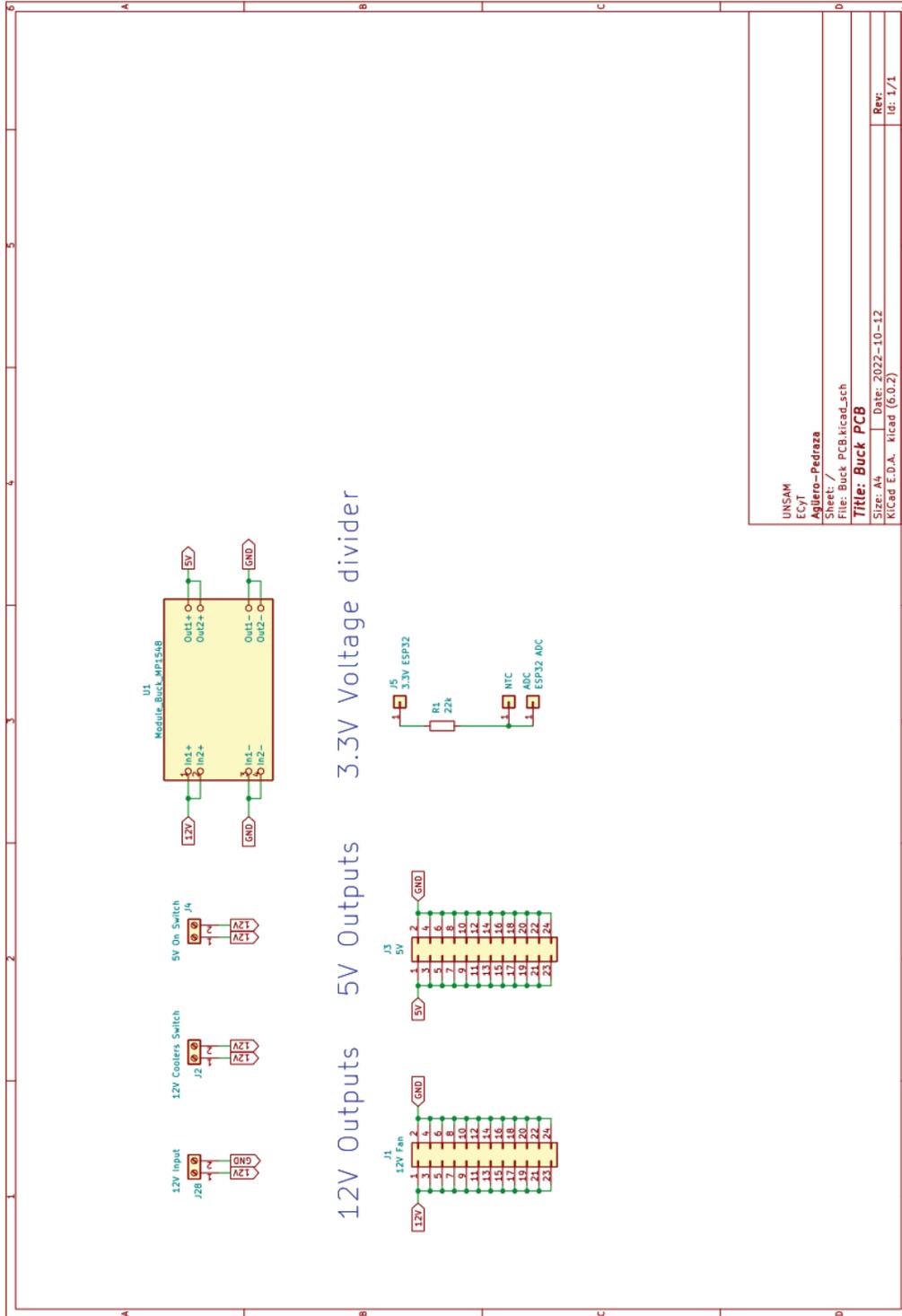
Título del proyecto		
Automatización de laboratorio químico mediante brazo robótico		
Estudiantes	Legajo	Firma
Agüero Nicolás - Pedraza Sebastián	CYT-4809 CYT-3622	
Fecha de inicio del proyecto		Fecha tentativa de finalización del Proyecto
01 de <u>Enero</u> del 2022		01 de Julio del 2022
Objetivo General		
Implementar un sistema de robotización basado en modelos de código abierto, para su aplicación en laboratorios de química. Este se ejecutará en tiempo real y de forma remota a través de una aplicación web.		
Objetivos Especificos		
<ul style="list-style-type: none">• Adquirir experiencia en el campo de la robótica.• Implementar conceptos avanzados en sistemas de control.• Manipular objetos correspondientes a las actividades prácticas de los laboratorios, mediante la utilización de un brazo robótico controlado en forma remota, a través de un software presente en una PC que cuente con conectividad a Internet.		
Antecedentes		
<ul style="list-style-type: none">• Robotización de laboratorios en universidades del resto del mundo.• Implementación de brazos robóticos en la industria automotriz y farmacéutica.• Automatización en áreas de logística para la selección de mercadería.		
Justificación o propósito del Proyecto (Aportes y resultados esperados)		
La robotización de los laboratorios de química brindará la posibilidad a docentes y alumnos de poder operarlos en forma remota e incorporando tecnología de vanguardia. Cabe destacar que este proyecto será pionero y permitirá dar el primer paso para la automatización de espacios en universidades del país.		

Supuestos			
<ul style="list-style-type: none"> • Se cuenta con el presupuesto para la adquisición del hardware necesario. • Se tiene conocimiento sobre lenguajes de programación. • Dominio de conceptos relacionados a sistemas de control. • Se cuenta con experiencia en términos de hardware y software para llevar a cabo los diferentes esquemas de conexión y desarrollo del algoritmo. • Se dispone de conocimiento en diseño de modelado 3D. 			
Restricciones			
<ul style="list-style-type: none"> • Tener acceso a los laboratorios de la universidad. • Disponer de una impresora 3D para realizar las piezas requeridas. • Obtener todos los componentes mecánicos y electrónicos para el armado del brazo robótico. • Contar con herramientas de software para la programación y simulación. 			
Referencias (bibliografía inicial para comenzar el proyecto)			
<ul style="list-style-type: none"> • Get started with cobots. Available in https://www.universal-robots.com • Thor Project. Available in http://thor.angel-lm.com/ • https://www.generationrobots.com/img/Kuka-YouBot-Technical-Specs.pdf • Fundamentos de robótica - 2° Ed, McGraw-Hill. 			
Dirigido por:	Dr. Carlos Rinaldi	Firma:	
Co-Dirigido por:	Dr. Juan Vorobioff	Firma:	
Aprobado por:		Firma:	

D.2 Desglose de tareas del proyecto



E Esquema PCB Buck



UNSAM
ECYT
Agüero-Pedraza
Sheet: /
File: Buck_PCB.kicad_sch
Title: Buck PCB
Size: A4
RCad E.D.A. kicad (6.0.2)
Date: 2022-10-12
Rev: 1/1
Rev: 1/1

F Hojas de datos

NTC Thermistor 3950

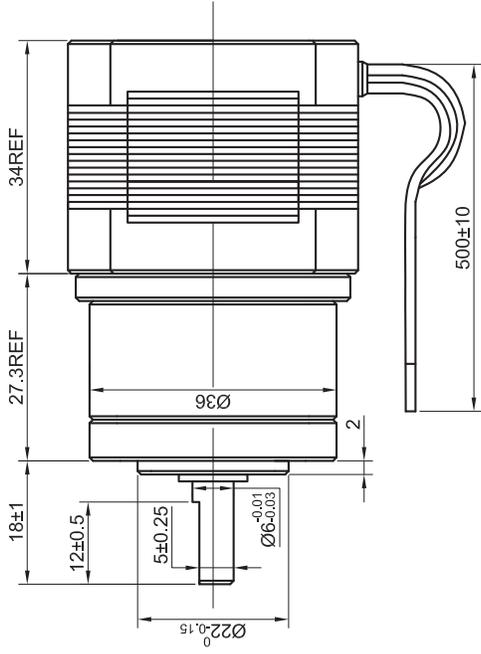
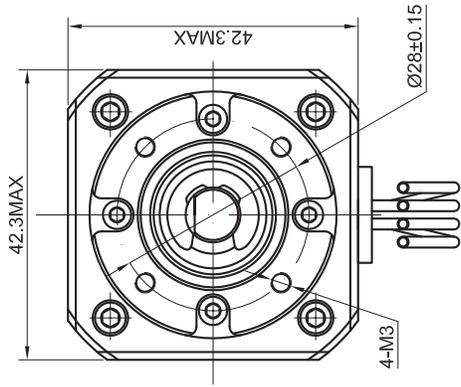
100 kΩ



Nennwiderstand:	$R_{25^{\circ}\text{C}} = 100 \text{ k}\Omega \pm 1\%$
Temperatur-Koeffizient:	$B_{25/50} = 3950 \text{ K} \pm 1\%$
Temperaturbereich:	-30°C bis +270°C

Temperatur-Widerstandstabelle

T [°C]	T [K]	R [kΩ]	T [°C]	T [K]	R [kΩ]	T [°C]	T [K]	R [kΩ]	T [°C]	T [K]	R [kΩ]
-30	243,15	1733,200	55	328,15	29,990	140	413,15	2,271	225	498,15	0,360
-25	248,15	1283,000	60	333,15	25,000	145	418,15	2,000	230	503,15	0,328
-20	253,15	959,050	65	338,15	20,900	150	423,15	1,770	235	508,15	0,2993
-15	258,15	724,800	70	343,15	17,550	155	428,15	1,589	240	513,15	0,2737
-10	263,15	551,410	75	348,15	14,760	160	433,15	1,414	245	518,15	0,2507
-5	268,15	423,700	80	353,15	12,540	165	438,15	1,259	250	523,15	0,2300
0	273,15	327,240	85	358,15	10,660	170	443,15	1,122	255	528,15	0,2114
5	278,15	254,800	90	363,15	9,100	175	448,15	0,997	260	533,15	0,1946
10	283,15	199,990	95	368,15	7,784	180	453,15	0,896	265	538,15	0,1794
15	288,15	157,600	100	373,15	6,710	185	458,15	0,797	270	543,15	0,1656
20	293,15	125,245	105	378,15	5,850	190	463,15	0,719	275	548,15	0,1531
25	298,15	100,000	110	383,15	5,070	195	468,15	0,643	280	553,15	0,1417
30	303,15	81,000	115	388,15	4,410	200	473,15	0,582	285	558,15	0,1314
35	308,15	65,650	120	393,15	3,850	205	478,15	0,533	290	563,15	0,1220
40	313,15	53,500	125	398,15	3,340	210	483,15	0,483	295	568,15	0,1134
45	318,15	43,780	130	403,15	2,940	215	488,15	0,437	300	573,15	0,1056
50	323,15	35,899	135	408,15	2,580	220	493,15	0,396			



SPECIFICATION	CONNECTION	BIPOLAR
AMPS/PHASE		0.4
RESISTANCE/PHASE(Ohms)@25°C		27.00±10%
INDUCTANCE/PHASE(mH)@1KHz		28.50±20%
HOLDING TORQUE w/o GEARBOX(Nm)[lb-in]		0.25[2.21]
GEAR RATIO		5 $\frac{1}{11}$
EFFICIENCY		90.00%
STEP ANGLE w/o GEARBOX(°)		1.80
BACKLASH@NO-LOAD		<=1°
MAX.PERMISSIBLE TORQUE(Nm)		2.00
MOMENT PERMISSIBLE TORQUE(Nm)		4.00
SHAFT MAXIMUM AXIAL LOAD(N)		50.00
SHAFT MAXIMUM RADIAL LOAD(N)		100.00
WEIGHT(Kg)[lb]		—
TEMPERATURE RISE:MAX.80°C (MOTOR STANDSTILL;FOR 2PHASE ENERGIZED)		
AMBIENT TEMPERATURE -10°C-50°C[14°F-122°F]		
INSULATION CLASS B 130°C[266°F]		

TYPE OF CONNECTION (EXTERN)		MOTOR	
PIN NO	BIPOLAR	LEADS	WINDING
1	A+ —	BLK	A+ A-
2	A- —	GRN	B+ B-
3	B+ —	RED	
4	B- —	BLU	

FULL STEP 2 PHASE-Ex. ,
WHEN FACING MOUNTING END (X)

STEP	A+	B+	A-	B-	CCW
1	+	+	-	-	↑
2	-	+	+	-	→
3	-	-	+	+	↓
4	+	-	-	+	←

BLK GRN RED BLU

STEPPERONLINE®

APVD 10.30.2020

CHKD

DRN

SCALE 1:1

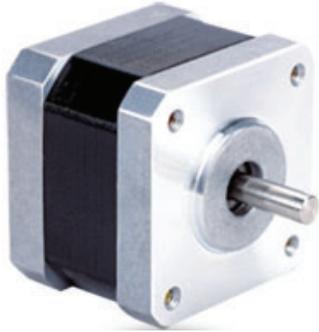
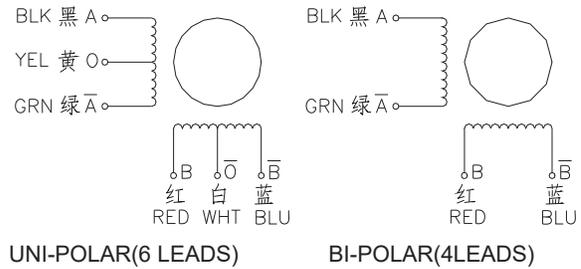
SIGNATURE

DATE

STEPPER MOTOR

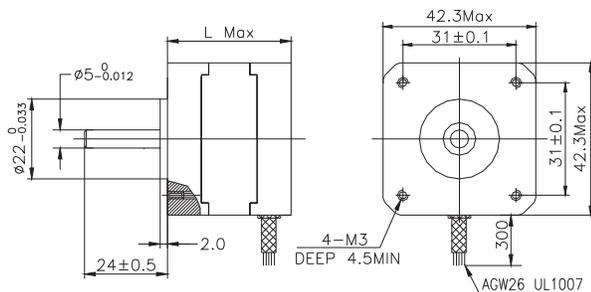
17HS13-0404S-PG5

2 Phase Hybrid Stepper Motor 17HS series-Size 42mm(1.8 degree)


Wiring Diagram:

Electrical Specifications:

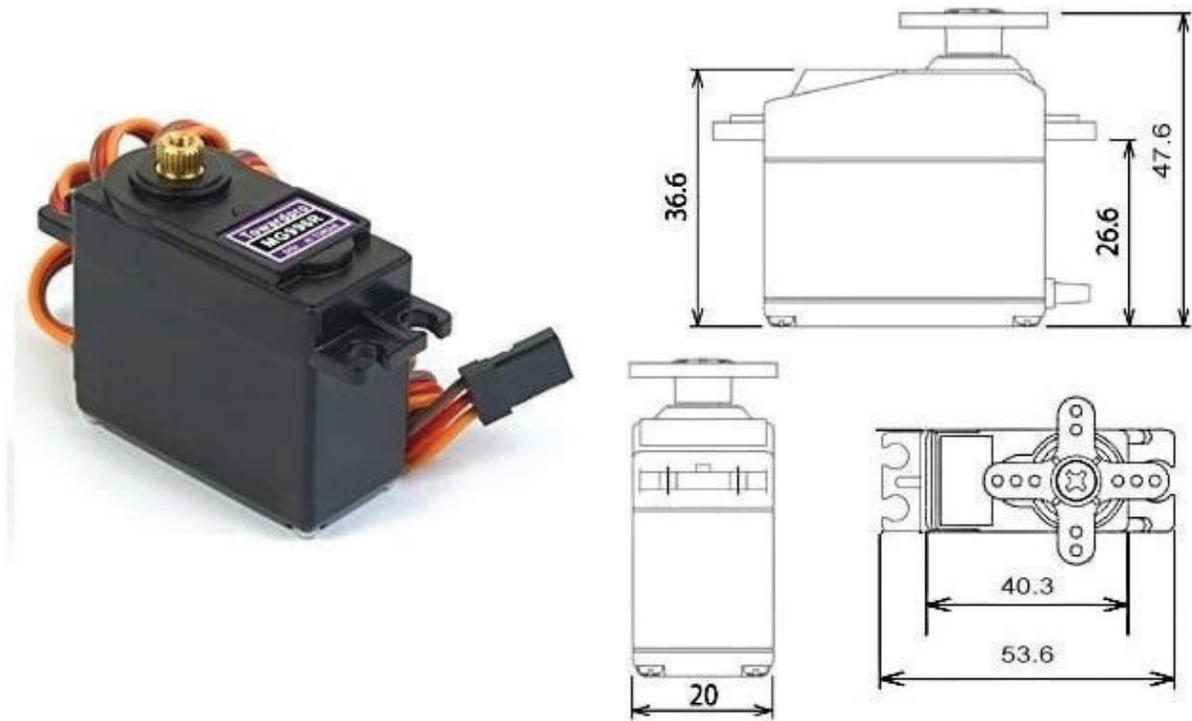
Series Model	Step Angle (deg)	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm ²)	Lead Wire (No.)	Motor Weight (g)
17HS2408	1.8	28	0.6	8	10	12	1.6	34	4	150
17HS3401	1.8	34	1.3	2.4	2.8	28	1.6	34	4	220
17HS3410	1.8	34	1.7	1.2	1.8	28	1.6	34	4	220
17HS3430	1.8	34	0.4	30	35	28	1.6	34	4	220
17HS3630	1.8	34	0.4	30	18	21	1.6	34	6	220
17HS3616	1.8	34	0.16	75	40	14	1.6	34	6	220
17HS4401	1.8	40	1.7	1.5	2.8	40	2.2	54	4	280
17HS4402	1.8	40	1.3	2.5	5.0	40	2.2	54	4	280
17HS4602	1.8	40	1.2	3.2	2.8	28	2.2	54	6	280
17HS4630	1.8	40	0.4	30	28	28	2.2	54	6	280
17HS8401	1.8	48	1.7	1.8	3.2	52	2.6	68	4	350
17HS8402	1.8	48	1.3	3.2	5.5	52	2.6	68	4	350
17HS8403	1.8	48	2.3	1.2	1.6	46	2.6	68	4	350
17HS8630	1.8	48	0.4	30	38	34	2.6	68	6	350

*Note: We can manufacture products according to customer's requirements.

Dimensions: unit=mm

Motor Length:

Model	Length
17HS2XXX	28 mm
17HS3XXX	34 mm
16HS4XXX	40 mm
16HS8XXX	48 mm

MG996R High Torque Metal Gear Dual Ball Bearing Servo



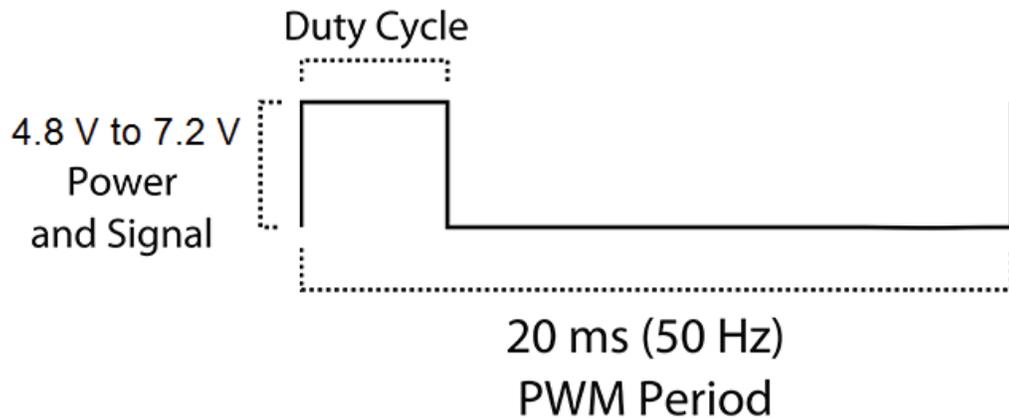
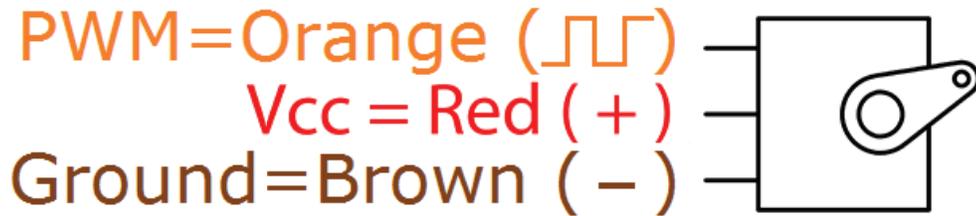
This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwith and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)

- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 μ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 $^{\circ}$ C – 55 $^{\circ}$ C



NodeMCU ESP32 Wireless & Bluetooth Development Board

NodeMCU is an open source IoT platform. ESP32 is a series of low cost, low power system-on-chip (SoC) microcontrollers with integrated Wi-Fi & dual-mode Bluetooth. The ESP32 series employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, with a clock rate of up to 240 MHz. ESP32 is highly integrated with built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules.

Features

- ▶ Manufactured by TSMC using their 40 nm process.
- ▶ Able to achieve ultra-low power consumption.
- ▶ Built-in ESP-WROOM-32 chip.
- ▶ Breadboard Friendly module.
- ▶ Light Weight and small size.
- ▶ On-chip Hall and temperature sensor
- ▶ Uses wireless protocol 802.11b/g/n.
- ▶ Built-in wireless connectivity capabilities.
- ▶ Built-in PCB antenna on the ESP32-WROOM-32
- ▶ Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- ▶ Uses CP2102 USB Serial Communication interface module.
- ▶ Programmable with ESP-IDF Toolchain, LuaNode SDK supports Eclipse project (C language).





Wireless Connectivity



Breadboard Friendly



USB Compatible



Lightweight

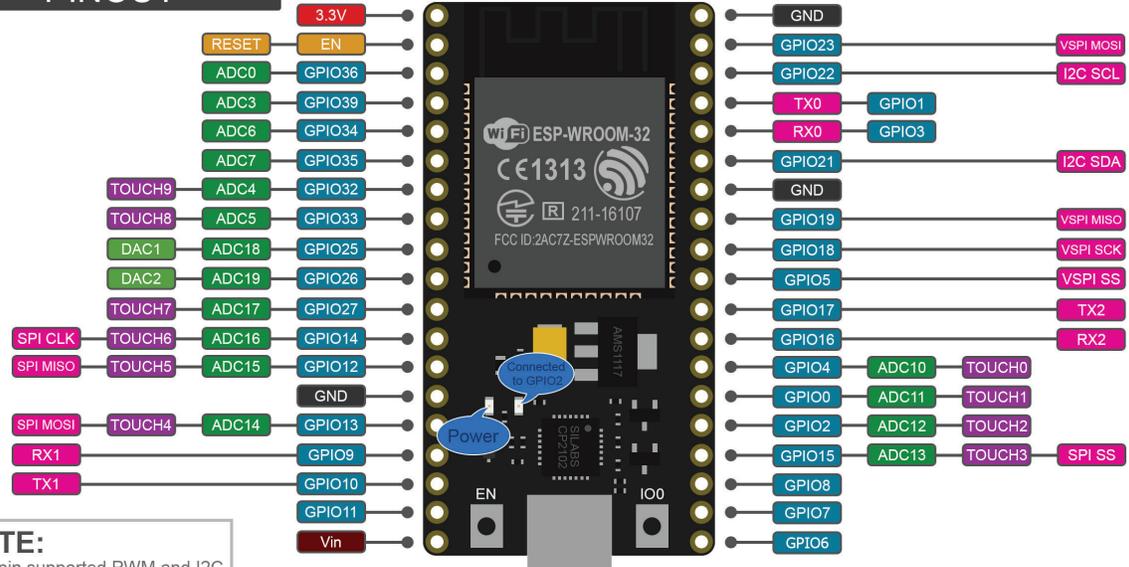


Bluetooth
4.0
Class 1 + BLE

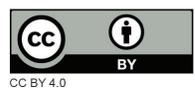
3.3V
POWERED
Low Power Consumption

NodeMCU-32S

PINOUT



NOTE:
All pin supported PWM and I2C
Pin current 6mA (Max. 12mA)



Source
<https://www.ioxhop.com/product/532/nodemcu-32s-esp32-wifiblueetooth-development-board>

Safety Precaution
All GPIO runs at 3.3V !!

Referencias

- [1] B. Niku. *Introduction to robotics: analysis, control, application*. John Wiley & Sons, 2011. ISBN: 978-0-470-60446-5.
- [2] S. Kumar Saha. *Introducción a la robótica*. McGraw-Hill, 2010. ISBN: 978-607-707-357-4.
- [3] P. Corke. *Robotics, Vision and Control Fundamental Algorithms in MATLAB*. Springer, 2011. ISBN: 978-3-319-54412-0.
- [4] A. Barrientos et al. *Fundamentos de robótica*. McGraw-Hill, 2014. ISBN: 978-84-481-5636-7.
- [5] et al. A. Pérez Cisneros. *Fundamentos de robótica y mecatrónica con Matlab© y Simulink©*. McGraw-Hill, 2007. ISBN: 978-84-9964-451-6.
- [6] M. Spong et al. *Robot Modeling and Control*. John Wiley & Sons, 2020. ISBN: 9781119523994.
- [7] IFR. *Executive Summary World Robotics 2021 Industrial Robots*. URL: https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2021.pdf.
- [8] Engineered Arts Limited. *State-of-the-art wonder*. URL: <https://www.engineeredarts.co.uk/es/inicio/>.
- [9] L.M. Ángel. *An Open Source 3D Printable 6 DOF Robotic Arm*. URL: <http://thor.angel-lm.com/>.
- [10] Blynk. *We make internet of things simple*. URL: <https://blynk.io/>.
- [11] Rhode & Schwarz. *Oscilloscope and probe fundamentals - Understanding UART*. URL: https://www.rohde-schwarz.com/us/products/test-and-measurement/oscilloscopes/educational-content/understanding-uart_254524.html.
- [12] A. Tenenbaum & J. Wetherall. *Redes de computadoras*. Pearson, 2012. ISBN: 978-607-32-0817-8.
- [13] Pololu Robotics & Electronics. *DRV8825 Stepper Motor Driver Carrier, High Current*. URL: <https://www.pololu.com/product/2133>.