

Control de Gripper por IIoT

Daniel Melo Avila
Ingeniería Mecatrónica
Universidad Nacional de Colombia
Bogotá, Colombia
danmeloavi@unal.edu.co

I. RESUMEN

Este informe consta de varias secciones que describen el diseño y construcción de una mano robótica con capacidad de agarre y sujeción de objetos. En la sección de requerimientos funcionales, se establecen los criterios necesarios, como el control remoto y la conexión a internet, así como la fuerza de sujeción y precisión de la mano. Luego, se detalla el mecanismo de actuación, que utiliza un motor stepper acoplado a un tornillo de potencia para transmitir el movimiento lineal a los mecanismos de agarre.

En la sección de diseño, se enfoca en la usabilidad y repetibilidad, con una interfaz de usuario intuitiva y un repositorio de archivos para la reproducción del diseño. Se analizan las fuerzas y torque requeridos para el tornillo de potencia, demostrando que un motor nema 17 es adecuado. Además, se comparan los agarres humanos con la funcionalidad de la mano robótica, destacando los agarres de precisión y potencia. Los resultados muestran que la mano es capaz de sujetar objetos de diferentes formas y tamaños, con la posibilidad de ajustar la posición de sujeción según las necesidades. En resumen, el informe presenta el diseño y construcción de una mano robótica funcional, con énfasis en los requerimientos funcionales y la comparación con los agarres humanos.

II. DISEÑO

II-A. Requerimientos funcionales

1. Poder ser controlada remotamente

- 1.1. Debe poder conectarse fácilmente con plataformas que la puedan controlar

DP: Implementación de Node-red para control de mano

- 1.2. Poder conectarse a internet

DP: ESP32 WROOM

2. Poder sujetar objetos de distintos tamaños **DP: Motor acoplado a tornillo de potencia que mueve axialmente al hexágono de presión**

- 2.1. Poder sostener una fuerza de 20N para sujetar al objeto.

DP: El motor, junto al sistema de transmisión de potencia

- 2.2. Se debe poder controlar la posición de la mano en su rango de movimiento con una precisión suficiente como para ejercer presión sobre los objetos en su rango de sujeción.

DP: Motor stepper que con definición de 200 pasos por vuelta

- 2.3. Debe permitir mover el hexágono de presión linealmente sobre su recorrido.

DP: Eje de motor acoplado a tornillo de potencia cuyo carro esta sujeto al hexágono de presión

- 2.3.1. Debe sujetar al motor rotatoriamente sobre su eje, ejerciendo un apoyo para que el motor ejerza su torque.

DP: Hexágono acrílico en la base del motor

- 2.3.2. Debe sujetar al motor y al sistema axialmente para que sobre el se deslice el hexágono de presión y sin ejercer un torque en contra que trabe al motor.

DP: Rodamiento de sujeción axial que encaja en escalon en la parte superior final del eje

3. El diseño debe ser usable

DP: Interfaz de usuario de node-red (dashboard) e instrucciones de uso

4. El diseño debe ser repetible **DP: Repositorio "stepper gripper" con los archivos necesarios para reproducción y el informe del readme del repositorio**

III. CONSTRUCCIÓN DEL MECANISMO DE ACTUACIÓN

El mecanismo de actuación consiste en un motor stepper nema 17 acoplado con un acople flexible a un tornillo de potencia sobre el cual se desplaza un carro. Este es un anillo de latón que se desplaza sobre la rosca del tornillo de potencia para transmitir la fuerza axialmente sobre el hexágono que transmite el movimiento lineal a los 4 mecanismos de 4 barras.

El tornillo de potencia está soportado radialmente, en un extremo por el acople, en el otro por un rodamiento 624ZZ. El diámetro del tornillo de potencia es de 8mm, por lo que se requirió de un torneado de un escalón en la punta superior del tornillo para un ajuste holgado del tornillo con el rodamiento y que de esta forma el rodamiento también pueda proveer un apoyo axial. También se cortó el tornillo a la medida para que no hallan juegos mecánicos.

Cuando el motor gira, el carro baja pero al entrar en contacto con el objeto a sujetar, se genera una fuerza de reacción en contra, que tiende subir al eje aplicando una carga hacia arriba sobre el anillo de latón. Esta fuerza busca subir al tornillo de potencia, el cual es soportado por el escalón que está en contacto con el rodamiento. De manera que puede seguir girando y aplicando la carga sin verse afectado por fuerzas de fricción que tendría el tornillo si no estuviese

el rodamiento y estubiese en contacto directamente con el plástico.

El motor esta soportado ante la rotación sobre su eje por medio de un hexágono acrílico cortado a láser, que esta sujeto a su parte posterior con los 4 tornillos que encajan en cada uno de los agujeros. Tiene la misma forma que el agujero y de esta manera no hay holguras en la rotación del tornillo. El plano de todas las piezas mencionadas esta en el repositorio "stepper gripper", sin embargo se deberá ajustar la longitud a la que se corta el eje y se maquina su escalón dependiendo de la longitud del motor disponible, el espesor del acrílico en la base y el rodamiento disponible.

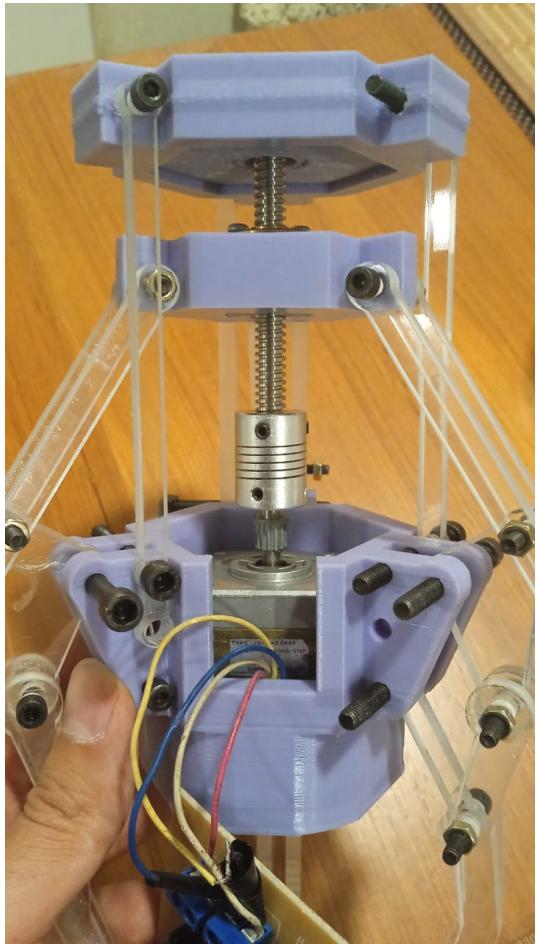


Figura 1. Gripper

IV. ANÁLISIS GEOMÉTRICO Y ESTÁTICO DEL MECANISMO:

El mecanismo esta conformado por 3 mecanismos de 4 barras. Los 3 se controlan por medio de dos eslabones y una pieza triangular al hexágono de presión, el cual tiene un recorrido de 3cm desde la posición superior hacia abajo. Esta conexión hace girar alrededor de su eje pinado al eslabón de entrada del mecanismo de 4 barras y por medio de esto logra su movimiento.

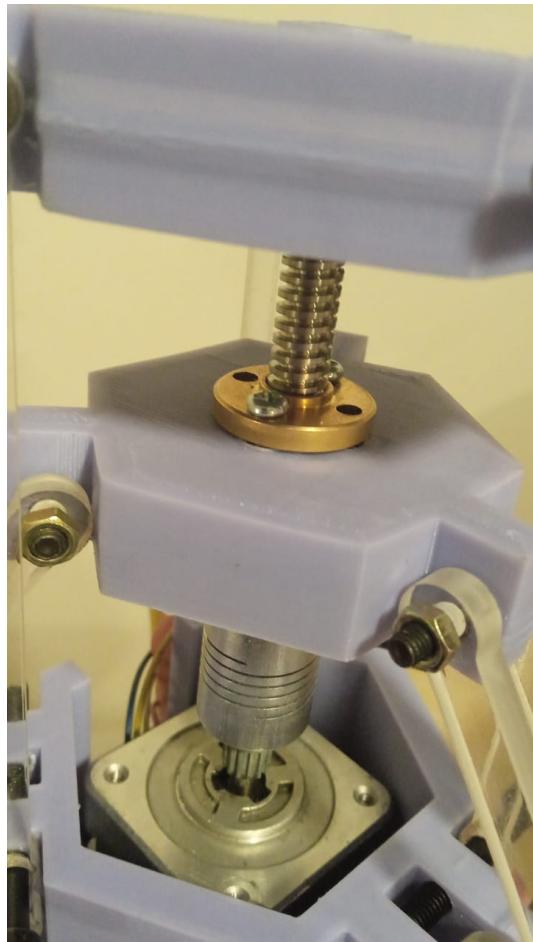


Figura 2. Hexágono de presión montado [1]

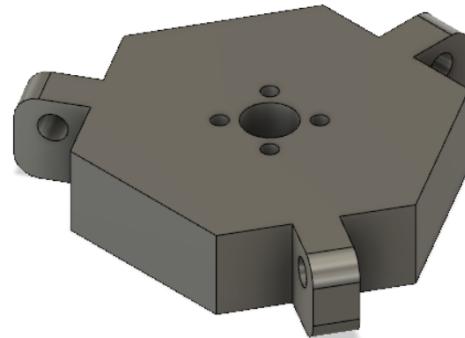


Figura 3. Hexágono de presión. Esta es la pieza que se desplaza sobre el tornillo de potencia [1]

El eslabón biela (el que se sujetta en sus extremos a los eslabones que giran alrededor de sus conexiones pinadas) del mecanismo es la misma pata de agarre que sujeta al objeto. Al trasladarse por el movimiento axial del hexágono presión 4, este eslabón disminuye su altura y se cierra sobre la posición de agarre. De esta manera sujeta al objeto.

El análisis geométrico del mecanismo de 4 barras consistió

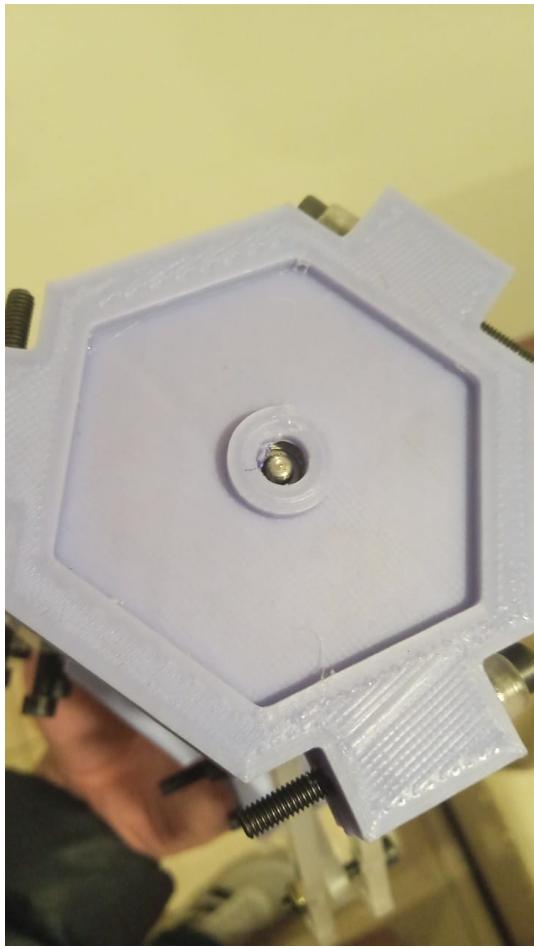


Figura 4. Foto superior del extremo maquinado del eje acoplado al rodamiento

en la determinación de todos los ángulos y posiciones de los eslabones de la geometría del mecanismo en función de la distancia recorrida por el movimiento lineal del carro del tornillo de potencia. Este análisis es un programa en un notebook de python llamado cálculos Mano, ya disponible en la carpeta de entrega. Es útil para el análisis de la geometría y el movimiento además de facilitar la operación de rediseño del mecanismo en caso de que se quiera rediseñar para sujetar piezas de menor tamaño.

Este análisis también es útil para poder realizar el análisis de fuerzas estático de la mano. El análisis estático permite conocer las fuerzas en cada punto de los eslabones del mecanismo en función de la posición del mecanismo, la fuerza de sujeción del objeto y el coeficiente de fricción entre la mano y el objeto. Una conclusión importante de este análisis es que la fuerza de fricción entre la pata de agarre y el objeto a sujetar es determinante para conocer la fuerza que se requiere aplicar axialmente en el mecanismo del tornillo de avance. Es necesario contar con una fuerza muy grande para poder sujetar un objeto si el coeficiente de fricción es bajo.

Entonces se requerirá gran torque del motor, el cual puede no disponer de este o, si lo puede proporcionar, puede ser muy grande y romperá la mano. Es por esto que se recomienda

añadir una superficie de caucho en los extremos de los dedos, pues así fácilmente se puede manejar más carga con el mismo motor o evitado comprar uno de mayor valor.

IV-A. Análisis de fuerzas del tornillo

El tornillo de potencia se analiza en un caso en el que la carga se debe "subir", como se muestra en la figura ???. Por lo tanto se utiliza la ecuación 1 para deducir el torque requerido del motor en función de la fuerza axial que debe ejercerse sobre el hexágono de presión.

En la gráfica 6 se visualiza el resultado de aplicar la ecuación 1 con los parámetros del tornillo de potencia (diámetro 8mm y paso de 2mm, rosca acme) teniendo en cuenta la incertidumbre que hay en el coeficiente de fricción entre latón y acero inoxidable. Se toma un rango para este coeficiente de entre 0.44 y 0.6, obteniéndose que para aplicar una fuerza de 100N, bastaría con un torque de 0.25Nm. De esta manera, se verifica que un motor nema 17 genérico es suficiente para el diseño. Se establecieron los 100N de fuerza al medir cuanta fuerza se requirió aplicar para la sujeción de una pieza pesada en el laboratorio manualmente.

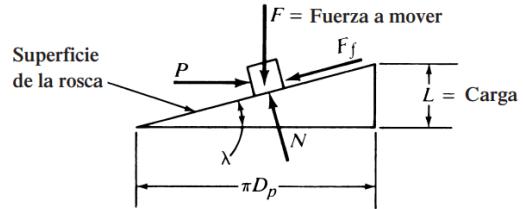


Figura 5. Diagrama de cuerpo libre para el caso del tornillo de potencia analizado

$$T_u = \frac{FD_p}{2} \left[\frac{L + \pi f D_p}{\pi D_p - f L} \right] \quad (1)$$

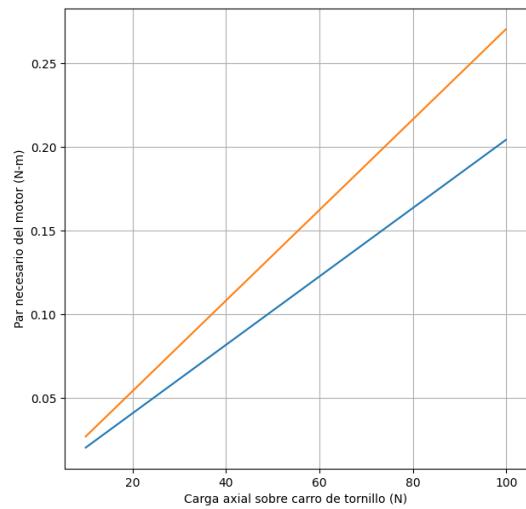


Figura 6. Gráfica del torque requerido del motor vs la fuerza realizada por el tornillo de potencia sobre el carro

IV-B. Análisis relativo a la mano humana

Cutkosky, basándose en el trabajo de Napier, desarrollo una clasificación de movimientos funcionalmente estudiando las poses de las manos de maquinistas en operaciones de manufactura. Según Napier, la mano adquiere una posición de acuerdo a lo que quiere lograr con el objeto.

La poses de la mano se clasifican entonces posiciones de potencia y de precisión.

El mecanismo esta diseñado para permitir efectuar 2 posiciones de precisión: trípode y esférica.

En los seres humanos, el agarre trípode se utiliza para elevar objetos pequeños, pues es preciso. Además, los 3 dedos virtuales que componen el agarre permiten ejercer 3 puntos fuerza lateral y momentos, los cuales lo hace ideal para sujetar el movimiento de un objeto en una trayectoria cuyas fuerzas están en dirección alrededor de los 360 grados del gripper. Un ejemplo en humanos de esta tarea es la escritura con lápiz o desenroscar un objeto sosteniéndolo de 3 lugares con los dedos.

El agarre esférico también forma parte de los agarres de potencia. Por medio de tener un punto de apoyo directo en el marco de la máquina (o en el efecto de mayor potencia como el brazo en el caso humano), el movimiento realizado ahora puede involucrar mucha mayor fuerza, pues los dedos que rodean al objeto no están haciendo empuje sino que están haciendo fuerza en dirección radial para sujetar al objeto. De esta manera, se puede controlar el movimiento de un objeto con mayor potencia que con el agarre de trípode.

Ambos agarres también tienen la ventaja de permitir el agarre de objetos de distintas formas y tamaños, pues no solo se pueden sujetar objetos esféricos o cilíndricos sino también objetos rectangulares o de forma irregular.

Solo con realizar un cambio en la posición de sujeción en node-red, se puede ajustar el mecanismo para sujetar distintos objetos.

V.

VI. KPI'S DE LA MANO:

VI-A. Máxima carga sujetada-probeta:

Tiene el objetivo de medir la capacidad de sujetar de la mano. Depende totalmente de la geometría del eslabón final de agarre y de el coeficiente de fricción de este con la pieza.

El procedimiento para se muestra a continuación. Este KPI esta medido respecto a una sola forma de probeta, con el material definido por el usuario de la mano. En este caso se utilizo un recipiente de plástico HDPE circular en el que se deposita la carga en el centro.

1. Se posiciona la mano directamente encima de la posición de agarre. De manera que solo se requiera un movimiento vertical para que la pieza quede entre las pinzas.
2. Se baja la mano hasta la posición de sujeción en donde este el diámetro de sujeción de la probeta.
3. Se realiza el movimiento de la mano hasta llegar a un punto cercano al objeto. Apartir de este punto, para el

apriete, se realiza el movimiento secuencial del gripper hasta lograr apriete de la pieza verificando alzando el gripper.

4. Se eleva la mano logrando agarre rígido de la pieza.

Si el anterior procedimiento fue exitoso, se toma el peso de la probeta utilizada como el valor de este KPI. Este KPI no solo consta del peso cargado sino también de la probeta utilizada (peso y forma).

Por esto se deben informar ambas informaciones al informar sobre este KPI.

La prueba realizada fue satisfactoria para el espécimen de cera de la figura 7, la cual se encuentra en el laboratorio. Por lo tanto este KPI tiene un valor de 259 gramos, como se observa en la figura.



Figura 7. Probeta utilizada en el ensayo del KPI de carga sujetada

VI-B. Tiempo de primer uso

Este KPI busca medir la usabilidad de la mano, es decir, la facilidad con la que se puede realizar la función que desempeña. A este punto, la mano esta enfocada a poder hacer uso del sistema de actuación de la mano para hacer el testeo y verificación de este.

De esta forma, se define este KPI por medio de medir la cantidad de tiempo que le toma a una persona mover por primera vez la mano.

Metodología para la obtención de este KPI:

1. La persona lee las instrucciones o ve el tutorial de utilización de la mano por completo.
2. Se le indica a la persona que haga que el gripper se mueva y se toma el tiempo desde que se le indica esto hasta que mueve por primera vez la mano.

Este KPI asume que la persona que lo utilizara es el laboratorista Juan Camilo de la universidad o cualquier persona que halla estudiado el pregrado en ingeniería Mecánica o Mecatronics que ya halla cursado los prerrequisitos de la materia APM o materias equivalentes.

También se asume que la persona sabe como acceder a node-red y también al dashboard de este.

VII. GEMELO DIGITAL

El gemelo digital permite la simulación del mecanismo de la mano en su rango de movimiento. El método de reproducción de su geometría fue la medición de las medidas con calibrador con una resolución de 0.02mm. Los ángulos fueron medida con el goniómetro disponible en el laboratorio con las medidas con una resolución de 1 grado.

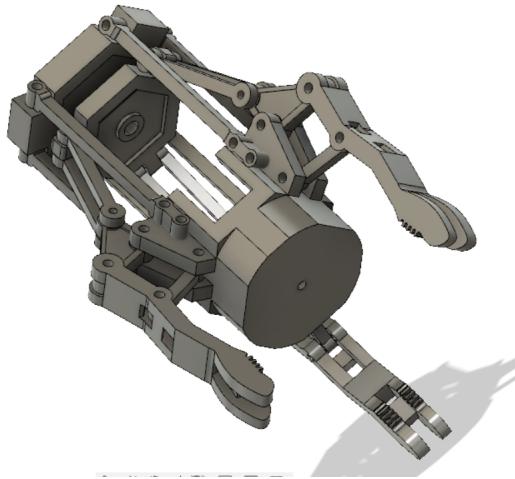


Figura 8. Gemelo digital de la mano [1]

Así mismo el gemelo digital se verifico al imprimir las piezas en material PLA. Se encontraron desviaciones respecto a las dimensiones principales de $\pm 1.3\text{mm}$.

El gemelo digital permite la simulación del movimiento del mecanismo, lo cual sera útil para previsualizar el movimiento del actuador una vez armado.

VIII. IIoT

VIII-A. Conexión wifi

El ESP32 utiliza las credenciales de la red, nombre de la red y contraseña de esta, para conectarse al inicio de su funcionamiento. Las credenciales se ingresan directamente en el código del ESP32.

VIII-B. Conexión MQTT

El esquema de conexión se puede apreciar en el diagrama POA estático. El ESP32 se conecta gracias a la librería PubSubClient con el broker MQTT de hive. Allí crea un tópico llamado StepperDMAOut y publica a este tópico. También se suscribe a un tópico llamado stepperDMAin, a donde se envía la posición del actuador.

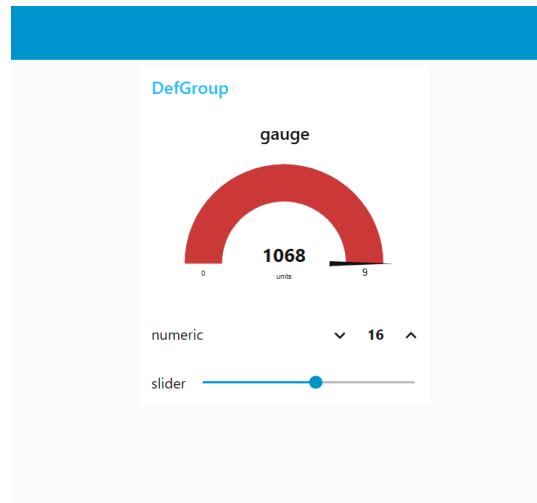


Figura 9. Interfaz de usuario node-red [1]

```
void setup_wifi() {
  delay(10);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
```

Figura 10. Conexion wifi

El servidor de Node-red, instalado en alguna raspberry-pi o computador con conexión a internet, publica la información de posición del actuador al tópico stepperDMAin y se suscribe al tópico stepperDMAout, del mismo broker Hive.

```
if (client.connect(clientId.c_str())) {
  Serial.println("connected");
  // Once connected, publish an announcement...
  client.publish("outTopicDMA", pos);
  // ... and resubscribe
  client.subscribe("stepperDMAtopic");
```

Figura 11. Código para conexión con broker

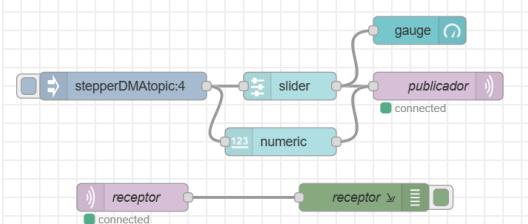


Figura 12. Diagrama de node-red utilizado para la conexión con el broker

La información del usuario se captura por medio de los nodos slider y numérica, los cuales representan componentes de la interfaz de usuario en el dashboard de nodered.

La conexión con el broker MQTT se realiza por medio de los nodos receptor y publicador, los cuales utilizan la

información de la dirección del broker, el nombre de los tópicos y el puerto de este para realizar la conexión.

El dato enviado desde node red es la carga paga, la cual tiene el tipo de dato payload. Al llegar y salir del borker sale con tipo byte y en el ESP 32 se realiza su conversión a char y luego a int, para así finalmente ser utilizada como el numero de pasos que el motor dará desde su posición de inicio.

La posición de inicio corresponde a 0 pasos. Por esto la posición del hexágono de presión debe comenzar desde el principio del recorrido del tornillo de potencia, en la parte superior.

IX. ELECTRÓNICA Y CONTROL DEL MOTOR

El circuito se compone del ESP32 y del driver del motor stepper. También se utiliza un regulador de 5V para darle poder al ESP32, un capacitor de desacople para compensar cambios de corriente bruscos, y unos leds indicadores. Las borneras se utilizan para conectar con el motor y la bornera-conector para conectar la fuente de 12V.

El control del motor se realizo por medio de los pines step y dir. Una vez conectado el driver como se observa en la figura [?], el pin step se debe conectar a algún GPIO del ESP32. Puede ser cualquiera pero se utilizo el GPIO 1. Cada vez que el motor da un paso, se envía un pulso por este pin. El pin dir se utiliza para controlar la dirección del motor. También se puede conectar a cualquier GPIO pero se utilizo el GPIO 3.

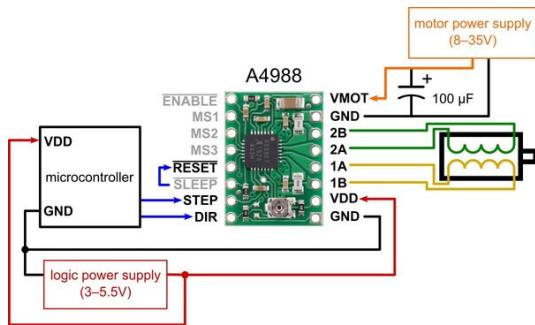


Figura 13. Esquema de conexión para el driver

El circuito se manufaturo per medio de serigrafía en PCB y los archivos de Kicad se encuentran disponibles en el repositorio "stepper gripper"de github.

X. CONTROL DEL MOVIMIENTO DEL MOTOR

El control de movimiento se realizo por medio de la librería accelstepper de Arduino.

Se debe instanciar una interfaz de stepper indicando los pines de pasos y dirección. 16

El setup del stepper se muestra en la figura ???. Estos parámetros se utilizaran cuando se le solicite al motor alcanzar una posición. setAcceleration configura la aceleración para el perfil trapezoidal que sigue el stepper entre posición y posición. La velocidad máxima durante recorrido trapezoidal también se configura con setMaxSpeed.

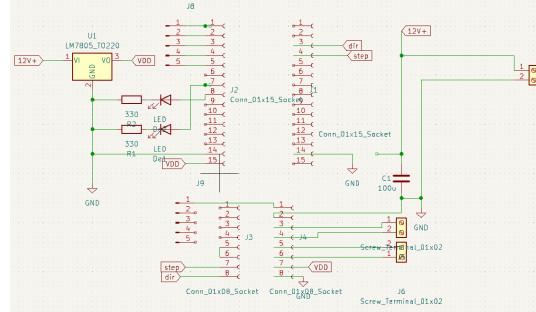


Figura 14. Esquemático del circuito

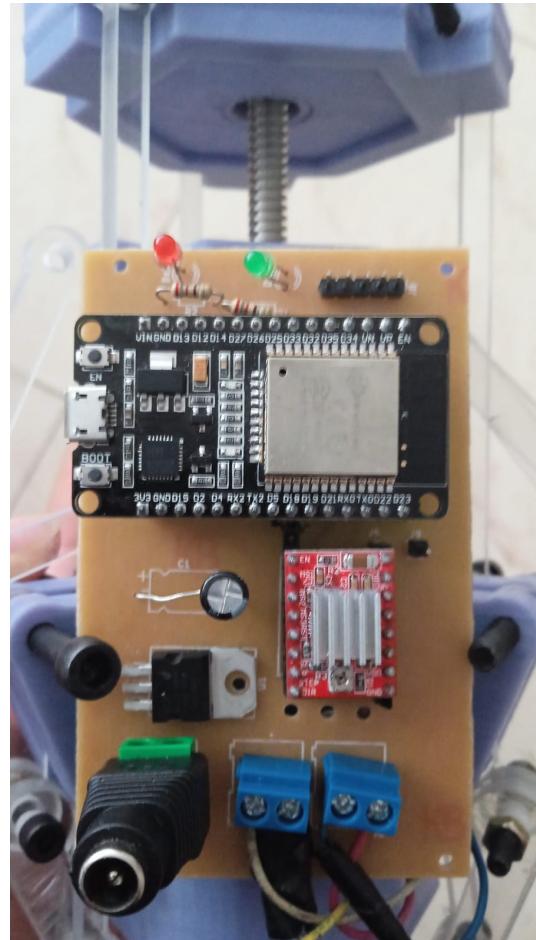


Figura 15. Circuito en PCB

setSpeed se utiliza para configurar la velocidad a la que el motor se moverá cuando lo haga a velocidad constante sin aceleración o desaceleración.

El motor se controla en el ciclo loop por medio del código de la figura 20. Requestedposition es la posición solicitada y reachingPosition es la posición a la que se esta moviendo en ese momento en particular. Si estas son distintas, se realiza el comando moveTo, el cual interrumpe el movimiento y hace que el motor inmediatamente comience a moverse hacia la posición señalada.

```
AccelStepper myStepper(motorInterfaceType, stepPin, dirPin);
```

Figura 16. Instanciación de interfaz para motor

```
// Setup stepper
myStepper.setMaxSpeed(500);
myStepper.setAcceleration(20);
myStepper.setMinPulseWidth(55);
myStepper.setSpeed(500);
myStepper.moveTo(requestedPosition);
myStepper.setSpeed(500);
```

Figura 17. Establecimiento de parámetros iniciales para motor

XI. VALIDACIÓN

■ Ítem 1 y 3 del contrato:

- Prueba de medición de KPI de la mano (video "Medición de KPI.en la carpeta de drive).
- Prueba de control inalámbrico de la mano en drive (video "prueba IOT.en la carpeta de drive).

■ Ítem 2 del contrato:

- Testeo de clasificador con datos de testeo del dataset (se obtuvo precisión de 81 % en el dataset)

■ Ítem 4 del contrato:

- Medición de medidas del gripper con calibrador y comparación con fotografías (carpeta de drive).

XII. ALGORITMO DE CLASIFICACIÓN DE SEÑALES

Para el algoritmo de clasificación se utilizo el dataset ninapro 5, en el 10 individuos utilizan un guante con sensores de

En el algoritmo se utilizaron 12 tipos de movimientos de la mano. Se busca poder hacer la identificación de 2 de entre todos estos porque así un dispositivo puede saber cuando se efectúa una u otra pose de la mano y también sabe cuando la mano no esta realizando ninguna de las dos sino que otra distinta. Esto es útil para hacer clasificación binaria sin la necesidad de clasificar a todos los movimiento dentro de las 2 opciones sino que, además de diferenciar entre 2 estímulos, identificar que no se debe reaccionar ante un estímulo distinto a los otros 2.

En el dataset, se utilizo la información del sujeto 2, las cuales son señales electromiográficas muestreadas a 200Hz. Hay 16 canales. 8 electrodos cada uno proveniente de cada una de las dos bandas de medición de ECG.

La información provee la secuencia de 6 repeticiones seguidas de 12 poses con la mano. Para extraer la información de cada uno de los ensayos ("trials"), se utilizo la información de etiquetado, la cual etiqueta cada muestra de tiempo según el estímulo que se mostraba en ese momento en pantalla (imagen de la pose que la mano de la persona debía adquirir) y según la repetición del mismo.

```
if (requestedPosition != reachingPosition){
    reachingPosition = requestedPosition;
    myStepper.moveTo(requestedPosition);
    if (requestedPosition < myStepper.currentPosition() ) {
        myStepper.setSpeed(-500);

    }else{
        myStepper.setSpeed(500);
    }
}
if (myStepper.currentPosition() == requestedPosition){
    myStepper.stop();
}
else{
    myStepper.runSpeed();
}
```

Figura 18. Establecimiento de parámetros iniciales para motor

Esta información se almacena en un tensor de 3 dimensiones, en donde la primera dimensión es el ensayo, la segunda el tiempo y la tercera el numero del canal de señal EMG.

```
import numpy as np

trials = np.empty((6,1100,16))
X = np.empty((6,1100,16))
y = []
print(trials.shape)

for j in range(12):
    for i in range(6):
        stim = (stimulus == j+1) & (repetition == i+1)
        n = stimulus.shape[0]
        indexes = [i for i in range(n)]
        arr = np.array(indexes)
        stim = stim.reshape(-1)
        indexes = arr[stim]
        index_list = indexes.tolist()
        # plt.plot(indexes)
        indexes.shape
        trial = all_data[index_list,:]
        trials[i,:,:] = trial[0:1100,:]
        # _ = plt.plot(trials[:, :, 4])
        if j == 0 or j == 1:
            y.append(j)
        else:
            y.append(3)

X = np.concatenate((X,trials), axis = 0)

y = np.array(y)
print(X.shape, y)
# all_data[[1,3],:].shape
plt.figure(figsize=(25,10))
_= plt.plot(X[17,:,:])
```

Figura 19. Extracción de la información y ordenamiento

También, a la par se genera el vector de etiquetas para las 3 clases.

Estos datos generados se dividen en información de entrenamiento y de testeo. La validación de este ítem en el contrato se realiza por medio de la evaluación en la red de los datos de testeo.

Así las cosas, la entrada a la red es una imagen de 2 dimensiones de 16 x 1100. La salida es una activación softmax para 3 clases. Es decir, se activara la correspondiente a la clase que crea que corresponde la red.

```

train_len = x_train.shape[0]
test_len = x_test.shape[0]
input_shape = (train_len, 1100, 16, 1)

# plt.plot(run_.T)

model = keras.Sequential(
    [
        keras.Input(shape=input_shape[1:]),
        layers.Conv2D(32, kernel_size=(100,8), activation="relu"),
        layers.MaxPooling2D(pool_size=(10,2),padding = 'same'),
        layers.Conv2D(64, kernel_size=(10,2), activation="relu"),
        layers.MaxPooling2D(pool_size=(2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)
model.summary()

```

Figura 20. Red neuronal convolución de clasificación

La red solo consta de 2 capas con filtros de convolución, 2 de maxpooling y 1 capa densa. Finalmente, se entrena la red y se logra el desempeño de 81 % de precisión en los datos de testeo, como se aprecia en la figura 21.

```

score = model.evaluate(x_test, y_test, verbose=0)
print("Test loss:", score[0])
print("Test accuracy:", score[1])
✓ 0.6s
Test loss: 0.584884762763977
Test accuracy: 0.8181818127632141

```

Figura 21. Resultados del testeo de la red

XIII. CONCLUSIONES:

- La manufactura aditiva es poderosa cuando es utilizada en conjunto con un gemelo digital por medio de permitir la reproducción exacta de la pieza y de esta forma desacoplando el proceso de prototipado del objeto real.
- Node-red facilita la conexión con protocolo MQTT con brokers, por medio de la representación intuitiva con nodos y flujos del sistema de información
- Los motores steppers facilitan el control de movimiento por medio de no requerir un sistema de feedback para controlar su posición y al permitir un torque sostenido.
- La manufactura es de valor permitiendo desacoplar el proceso de prototipado y testeo de mecanismos prototípico del funcionamiento del dispositivo final, pues por medio de testear un prototípico en la pieza, se mitigan y detectan riesgos o fallas que comprometerían una pieza clave para el dispositivo final. Este fue el caso en el proyecto con la testear piezas impresas en 3d con las modificaciones que se le realizarían a las piezas finales.
- Fdm es valiosísima para el prototipado rápido por medio de permitir hacer piezas con geometrías casi arbitrarias y con suficiente fortaleza física como para aplicaciones en donde se quiere hacer un prototípico de otro material plástico. De esta manera verificando que el dispositivo real no se valla a romper, pues prototípico y pieza real tiene propiedades mecánicas muy similares. De esta manera se puede casi que tener el mismo producto funcional ya

disponible a la salida del proceso de FDM. La complejidad geométrica alcanzable es agradecida pues también elimina la necesidad de invertir tiempo y destreza humana en procesos secundarios que vendrían con maquinados o pegamentos en otros procesos de manufactura como con madera o láminas de acrílico.

- A pesar de ser valiosísima, FDM también puede requerir mucho tiempo de manufactura para sus piezas y esto puede ser limitante en una situación crítica en tiempo (Se requirieron 12 horas para la pieza del marco en este proyecto, por ejemplo). Mas aun, requiere de disponer de por lo menos el doble de material y tiempo requerido para la fabricación de la pieza debido a siempre presente posibilidad de atasco de la máquina, y en este caso, se requeriría entonces volver a comenzar el proceso de impresión. Esto en resumen hace que este en desventaja ante otros procesos como el corte de acrílico en situaciones que no dan espera.

REFERENCIAS

- [1] H. University. 4d printing: Shapeshifting architecture. [Online]. Available: https://www.youtube.com/watch?v=7Q_Fu1KIVac&ab_channel=HarvardUniversity