



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería Electrónica Industrial y Automática**

**“Prototipado y control de un dedo protésico  
para amputaciones parciales de mano”**

**Autor: Pedraza Huarcaya, Juan Carlos**

**Tutor: Mansilla Gallo, Alberto**

Dpto. Ciencia de los Materiales e  
Ingeniería Metalúrgica, Expresión  
Gráfica en la Ingeniería, Ingeniería  
Cartográfica, Geodesia y  
Fotogrametría, Ingeniería Mecánica  
e Ingeniería de Procesos de  
Fabricación

**Valladolid, septiembre de 2021**



## RESUMEN BREVE

Este proyecto tiene como objetivo el diseño, fabricación y control de un dedo pulgar protésico para amputaciones parciales de mano. Para ello, se comenzará con un análisis de las soluciones existentes hoy en día en el mercado, detallándose las ventajas e inconvenientes. A continuación, se presentará un estudio de la anatomía del miembro con intención de identificar los elementos que permiten su cinemática, así como otras características que la prótesis deberá reproducir. Por otro lado, las especificaciones de diseño cubrirán desde requisitos de dimensionado, peso y estética, hasta los aspectos legales asociados al producto.

Por último, a este prototipo se le dotará de los elementos necesarios para generar un sistema de control mioeléctrico el cual será gobernado por un ordenador Raspberry Pi. Este micro ordenador hará uso de sensores electromiográficos para clasificar y ejecutar las ordenes que reproducirán los movimientos de flexión/extensión del dedo protésico.

**Palabras clave:** prótesis, Raspberry Pi, dedo pulgar, sEMG, control, diseño

## ABSTRACT

This project aims to design, manufacture and control a prosthetic thumb for partial hand amputations. To do this, it will begin with an analysis of the existing solutions on the market today, detailing the advantages and disadvantages. Next, a study of the anatomy of the limb will be presented with the intention of identifying the elements that allow its kinematics, as well as other characteristics that the prosthesis must reproduce. On the other hand, the design specifications will cover from dimensioning, weight and aesthetic requirements, to the legal aspects associated with the product.

Finally, this prototype will be equipped with the necessary elements to generate a myoelectric control system which will be governed by a Raspberry Pi computer. This microcomputer will use electromyographic sensors to classify and execute the commands that will reproduce the flexion / extension movements of the prosthetic finger.

**Keywords:** prosthetic, Raspberry Pi, thumb, sEMG, control, design

### ***Agradecimientos***

A mi tutor *Alberto Mansilla* por haberme ofrecido esta gran oportunidad para ampliar mis conocimientos.

Al *Laboratorio* por la ayuda prestada.

A mi familia, mis padres *Fortunato* y *Fidelia* y mis hermanas por acompañarme durante esta etapa con su apoyo incondicional, sin ellos no hubiese sido posible.

## ÍNDICE

CAPÍTULO 1.- INTRODUCCIÓN.....	1
<b>1.1 Motivación.....</b>	<b>2</b>
<b>1.2 Objetivos .....</b>	<b>3</b>
CAPÍTULO 2.- ESTADO DEL ARTE .....	4
<b>2.1 Antecedentes.....</b>	<b>5</b>
<b>2.2 Desarrollos actuales .....</b>	<b>8</b>
CAPÍTULO 3.- ANATOMÍA DE LA MANO .....	15
<b>3.1 Estudio de la mano.....</b>	<b>16</b>
<b>3.2 Antropometría del dedo pulgar .....</b>	<b>19</b>
CAPÍTULO 4.- ESPECIFICACIONES DEL DEDO PULGAR PRÓTESICO.....	21
<b>4.1 Parámetros básicos de diseño .....</b>	<b>22</b>
<b>4.2 Estructura conceptual del dedo protésico.....</b>	<b>24</b>
CAPÍTULO 5.- PROTOTIPADO Y CONTROL .....	25
<b>5.1 Diseño mecánico y fabricación .....</b>	<b>26</b>
<b>5.2 Diseño electrónico .....</b>	<b>39</b>
<b>5.1 Sistema de control mioeléctrico .....</b>	<b>54</b>
<b>5.2 Pruebas y resultados .....</b>	<b>74</b>
CAPÍTULO 6.- CONCLUSIONES Y LÍNEAS FUTURAS.....	80
<b>6.1 Conclusiones .....</b>	<b>81</b>
<b>6.2 Líneas futuras de trabajo.....</b>	<b>82</b>
Bibliografía .....	83
ANEXO I:.....	88

## Índice de figuras

<b>Figura 1.</b> Mano protésica de Alt-Ruppin. [11] .....	5
<b>Figura 2.</b> Fotografía de la primera prótesis mioeléctrica. [11] .....	6
<b>Figura 3.</b> Prótesis ProDigits versión años 80 (a) y actual (b). [13] .....	6
<b>Figura 4.</b> Prótesis mecánica X finger. [13] .....	7
<b>Figura 5.</b> Mecanismo X finger [13] .....	7
<b>Figura 6.</b> Prótesis comerciales. I-Limb (a), Bebionic (b), Michelangelo (c) y Cyber Hand (d). [10] .....	8
<b>Figura 7.</b> Prótesis i-Digits Quantum [15] .....	9
<b>Figura 8.</b> Dígitos de distintos tamaños(a) y placa de montaje de nudillos (b). [15] .....	10
<b>Figura 9.</b> Electrodo de perfil bajo (a) y Sensor de fuerza resistivo (b). [15] .....	10
<b>Figura 10.</b> Vincent Partial Hand. [16] .....	11
<b>Figura 11.</b> Componentes y brazaletes del Vincent Partial Hand. [16].....	12
<b>Figura 12.</b> Pulgar de la Mano Parcial Vincent. [16] .....	12
<b>Figura 13.</b> Modelo Renderizado (a) y prototipo terminado (b) de la mano de seis grados de libertad. [18] .....	13
<b>Figura 14.</b> Mecanismo de accionamiento de los dedos a excepción del pulgar. [18].....	14
<b>Figura 15.</b> Esquema del funcionamiento del pulgar con dos motores. [18] .....	14
<b>Figura 16.</b> Huesos de la mano. [29] .....	16
<b>Figura 17.</b> Anatomía del dedo. [19].....	17
<b>Figura 18.</b> Falanges y articulaciones del dedo pulgar. [19].....	18
<b>Figura 19 .</b> Pulgar en estado neutral (a) y pulgar flexionado (b). [20] .....	19
<b>Figura 20.</b> Falanges del pulgar.....	20
<b>Figura 21.</b> Dimensiones y movimiento de diseño. [20].....	22
<b>Figura 22.</b> Conjunto: Reductor (a) - motor (b) y encoder (c). [25].....	26
<b>Figura 23.</b> Motor CC con imanes permanentes y bobinado sin núcleo. [26] .....	26
<b>Figura 24.</b> Extracto de las características del motor RE8. [26] .....	27
<b>Figura 25.</b> Reductor Planetario RE [28] .....	28
<b>Figura 26.</b> Extracto de las características del reductor planetario GP8. [28] .....	28
<b>Figura 27.</b> Encoder Magneto Resistivo MR 100 [29].....	29
<b>Figura 28.</b> Extracto de las características del Encoder MR. [29].....	29
<b>Figura 29.</b> Esquema Motor RE8 + Reductora + Engranajes. [28].....	30
<b>Figura 30.</b> Conjunto Sin Fin – Corona. [30] .....	31
<b>Figura 31.</b> Esquema Motor RE8 + Reductor + Sistema tornillo sin fin – corona.....	32
<b>Figura 32.</b> Rango de funcionamiento del Motor RE8. [27] .....	33
<b>Figura 33.</b> Distancia D (desde el punto de giro hasta el punto de contacto) .....	33
<b>Figura 34.</b> Dimensiones del reductor con rosca. [28] .....	34
<b>Figura 35.</b> Dedo 3D en Inventor (a) y Dedo Impreso (b).....	35
<b>Figura 36.</b> Base del dedo .....	35
<b>Figura 37.</b> Base 3D en Inventor (a) y Base Impresa (b). .....	36
<b>Figura 38.</b> Pasador cilíndrico insertado entre la base y el dedo .....	36
<b>Figura 39.</b> Diseño en Inventor del soporte del pasador (a) y soporte impreso (b). .....	37
<b>Figura 40.</b> Partes fabricadas con impresora 3D. ....	37

<b>Figura 41.</b> Despiece de los elementos que conforman el dedo protésico. ....	38
<b>Figura 42.</b> Fotos desde distintos ángulos del dedo pulgar protésico. ....	38
<b>Figura 43.</b> Sistema: Reductor + Motor + Encoder + Controlador de Posición Epos4. [31] .....	39
<b>Figura 44.</b> Dibujo dimensional de la EPOS4 Compact en mm. [29].....	40
<b>Figura 45.</b> Esquema de conexiones Motor-Reductor-Encoder-Sensor Emg. [31].....	42
<b>Figura 46.</b> Esquema puesto en marcha con EposStudio.....	43
<b>Figura 47.</b> Parámetros del motor. ....	44
<b>Figura 48.</b> Parámetros del Reductor. ....	44
<b>Figura 49.</b> Parámetros del Encoder. ....	44
<b>Figura 50.</b> Límites de Operación.....	45
<b>Figura 51.</b> Modo de operación PPM. [31] .....	46
<b>Figura 52.</b> Descripción General de la arquitectura de control.....	47
<b>Figura 53.</b> Arquitectura de control del regulador de corriente. ....	47
<b>Figura 54.</b> Ajuste del regulador de intensidad .....	47
<b>Figura 55.</b> Holgura media del reductor planetario GP8. [28] .....	48
<b>Figura 56.</b> Ajuste del regulador de Velocidad .....	49
<b>Figura 57.</b> Arquitectura de control del regulador de posición con feedforward .....	49
<b>Figura 58.</b> Ajuste del regulador de Posición y pruebas realizadas con EposStudio.....	50
<b>Figura 59.</b> Raspberry Pi Zero. [34] .....	51
<b>Figura 60.</b> Pines de la Raspberry Pi Zero. [35].....	52
<b>Figura 61.</b> Esquema de funcionamiento. ....	53
<b>Figura 62.</b> Esquema general de funcionamiento .....	54
<b>Figura 63.</b> Desplazamiento del potencial de acción a través del axon. [34] .....	55
<b>Figura 64.</b> Diferencia de voltaje durante y después del potencial de acción. [35] .....	56
<b>Figura 65.</b> Esquema de dos ejemplos de unidades motoras y sus ramificaciones [37] ...	57
<b>Figura 66.</b> Propagación del potencial de acción a través de las fibras musculares. [36] .	57
<b>Figura 67.</b> Superposición de potenciales de acción para generar una señal EMG. [36] ..	58
<b>Figura 68.</b> Adquisición de la señal electromiográfica con electrodos de superficie. [39]	59
<b>Figura 69.</b> Electrodos con gel conductor adhesivo. ....	59
<b>Figura 70.</b> Diagrama de bloques del tratamiento y uso de la señal EMG.....	60
<b>Figura 71.</b> Sensor muscular MyoWare compuesto por tres electrodos. [40] .....	61
<b>Figura 72.</b> Disposición de los pines del sensor MyoWare. [40] .....	61
<b>Figura 73.</b> Conexionado Sensor MyoWare – Controladora Epos4. [29].....	62
<b>Figura 74.</b> Alternativas para la colocación de los diferentes sensores EMG superficiales en el antebrazo. [41].....	63
<b>Figura 75.</b> Colocación de los electrodos. [41] .....	63
<b>Figura 76.</b> Conexionado y posicionamiento del sensor MyoWare. ....	64
<b>Figura 77.</b> Esquema general de funcionamiento .....	65
<b>Figura 78.</b> Diagrama de bloques por módulos de la solución completa. ....	65
<b>Figura 79.</b> Comunicación vía USB. ....	66
<b>Figura 80.</b> Librerías de la Epos4 compatibles con Linux. [30] .....	66
<b>Figura 81.</b> Manual explicativo de la librería Epos Command y sus funciones en Linux. [30] .....	66

<b>Figura 82.</b> Entorno de programación Geany. ....	67
<b>Figura 83.</b> Extracto de la página 13 del documento Epos Command Library. [30] .....	67
<b>Figura 84.</b> Extracto del código donde se declaran los parámetros de comunicación.....	68
<b>Figura 85.</b> Extracto fraccionado del código encargado de la comunicación. ....	68
<b>Figura 86.</b> Bloques del módulo dos. ....	69
<b>Figura 87.</b> Conexión de lectura y procesamiento de señal EMG .....	69
<b>Figura 88.</b> Extracto de la página 117 del documento Epos Command Library. [30] .....	70
<b>Figura 89.</b> Extracto fraccionado del código encargado de la lectura del sensor EMG.....	70
<b>Figura 90.</b> Bloques que conforman el módulo del sistema final. ....	71
<b>Figura 91.</b> Conexión del sistema final.....	71
<b>Figura 92.</b> Conector Molex. [42].....	72
<b>Figura 93.</b> Fragmento del código que clasifica la señal electromiográfica.....	72
<b>Figura 94.</b> Fragmento del código que posiciona el dedo en flexión.....	73
<b>Figura 95.</b> Dedo pulgar protésico en extensión (a), en reposo (b) y en flexión (c). ....	73
<b>Figura 96.</b> Elementos que conforman el sistema del pulgar protésico accionado por control mioeléctrico. ....	74
<b>Figura 97.</b> Valores leídos en la entrada analógica de la Epos4. ....	75
<b>Figura 98.</b> Diagrama de flujo representativo del sistema de control mioeléctrico .....	76
<b>Figura 99.</b> Dedo protésico en posición de reposo (a) y en flexión (b). ....	77
<b>Figura 100.</b> Dedo protésico sometido a una carga de 0,225kg. ....	78
<b>Figura 101.</b> Evolución de la intensidad demandada, actual y el torque en el tiempo. ....	79



## Índice de tablas

<b>Tabla 1</b> Valores de los ángulos del dedo pulgar. [20] .....	19
<b>Tabla 2.</b> Longitud promedio de las falanges del dedo pulgar. [13].....	20
<b>Tabla 3.</b> Fuerzas de agarre en cada dedo protésico comercial. [7] .....	23
<b>Tabla 4.</b> Datos eléctricos de la Epos 4. [32].....	40
<b>Tabla 5.</b> Entradas de la Epos4. [32] .....	41
<b>Tabla 6.</b> Salidas de la Epos4. [32].....	41
<b>Tabla 7.</b> Interfaces de la Epos4. [32] .....	41
<b>Tabla 8.</b> Condiciones ambientales y datos mecánicos. [32] .....	42
<b>Tabla 9.</b> Plataformas, arquitecturas e interfaces compatibles con la EPOS4. [30].....	51
<b>Tabla 10.</b> Asignación de pines del conector analógico X8 de la EPOS4. [29] .....	62

# CAPÍTULO 1.- INTRODUCCIÓN

## 1.1 Motivación

La mano es una de las partes más importantes de nuestro cuerpo ya que interviene prácticamente en cada actividad que realizamos cumpliendo principalmente dos funciones: Presión y tacto.

La pérdida de la mano o parte de ella deriva en una nueva perspectiva a la hora de realizar actividades triviales del día a día. No solo es necesario aprender de nuevo a realizar tareas sencillas que, con el nuevo contexto se tornan más complejas, sino que, además, será necesaria la aceptación de la situación.

Siendo así, no es difícil entender la importancia de su pérdida. Las repercusiones no solo son funcionales, sino que también psicológicas.

Actualmente, las amputaciones son situaciones menos frecuentes que antes, debido sobre todo al progreso en materia de seguridad laboral, aun así estudios recientes indican que las amputaciones importantes de extremidades superiores tienen una prevalencia media en Europa de 11,6/100 000 individuos [1], siendo las amputaciones parciales de mano el tipo más común de amputación relacionada con el trabajo en todo el mundo [2].

Sin embargo, solo se ha desarrollado un número relativamente pequeño de prótesis de manos parciales, las principales razones son las siguientes [3]. Primero, las presentaciones parciales de la mano son anatómicamente muy diversas. Por lo tanto, es difícil estandarizar y escalar un diseño. En segundo lugar, el espacio disponible para la integración de componentes es limitado, lo que dificulta la aplicación de las soluciones utilizadas en las manos y brazos protésicos de última generación (por ejemplo, los de [4] – [5]), los cuales tienen además un precio muy elevado.

En un intento de abordar estos desafíos, este trabajo fin de grado presenta el diseño, la fabricación y el control de una prótesis mioeléctrica para amputaciones parciales de mano, específicamente para amputaciones a nivel de la articulación metacarpofalángica del dedo pulgar. Este dedo es el de mayor funcionalidad y la pérdida del mismo reduce un 40% la capacidad de funcionalidad de la mano [6] [7].

## 1.2 Objetivos

- 1) Diseño de un dedo pulgar protésico de carácter práctico y de bajo coste, en la que la mayoría de sus partes puedan ser impresas en una máquina de impresión 3D.
- 2) Dotar a la prótesis de los elementos necesarios para generar un eficiente sistema de control e implementar un algoritmo que utilizará los datos enviados por el sensor superficial electromiográfico (sEMG) para generar la posición que ha de alcanzar el dedo, el cuál será impulsado por un motor de corriente continua.
- 3) Puesta en funcionamiento del prototipo. Para ello, gracias a la técnica de impresión 3D, se imprimirán y se ensamblarán las piezas del prototipo desarrollado. Posteriormente se realizarán las pruebas de control correspondientes al posicionamiento de la prótesis mioeléctrica.
- 4) Crear un punto de partida sólido para el estudio y desarrollo de la próxima generación de prótesis para amputaciones parciales de la mano a bajo costo, utilizando componentes de calidad. La mayoría de los proyectos con bajo coste económico que se encuentran sobre prótesis están desarrollados con Arduino, lo cual no es muy fiable si se le quiere dar un uso profesional.
- 5) Indicar algunas de las líneas de trabajo futuro a seguir en base al proyecto realizado.

# CAPÍTULO 2.- ESTADO DEL ARTE

En este capítulo se procederá a la descripción del estado del arte, haciendo un breve recorrido por las distintas prótesis que fueron desarrollándose a lo largo de la historia concernientes al marco de este trabajo, se clasificará y analizará los desarrollos expuestos, con el objetivo de tener un punto de partida sólido y actualizado que hará de cimientos en el desarrollo de nuestro prototipo.

## 2.1 Antecedentes

Las prótesis nacen ante la necesidad del ser humano de volver a recuperar funciones en aquellas partes que, por diversos motivos, se encuentran parcial o completamente amputadas, o deformadas. Las primeras prótesis surgen con el único objetivo de satisfacer especificaciones estéticas. Más adelante, comienzan a ganar cierta funcionalidad gracias a las mejoras en herramientas y al mayor conocimiento en el tratamiento y uso de materiales. Actualmente, las prótesis son capaces de recrear, por ejemplo, el sentido del tacto de zonas amputadas, función que era impensable hace unos años.

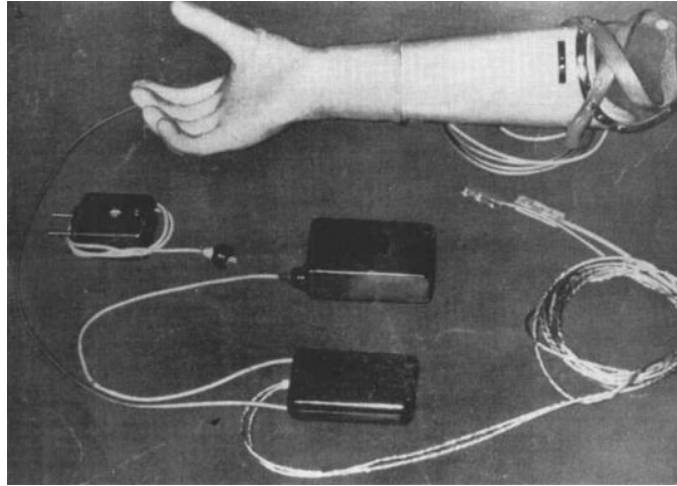
La primera prótesis de miembro superior registrada data del año 2000 a. C., la cual fue encontrada en una momia egipcia. La prótesis estaba sujeta al antebrazo a través de un cartucho adaptado al mismo [8]. Posteriormente, con el manejo del hierro, el hombre pudo construir prótesis más resistentes que pudieran ser empleadas para portar objetos pesados, tal es el caso del general romano Marcus Sergius que, durante la Segunda Guerra Púnica (218-202 a.C.) mando a que se le fabricara una mano de hierro con la cual pudiera sostener su escudo y volver a la batalla [9].

Más adelante, en 1400 se fabricó la mano de Alt-Ruppin (**Figura 1**). Construida también en hierro, que permitió una cinemática mayor debido a que tenía un mecanismo compuesto por una serie de resortes y correas en cuero. Esta prótesis constaba de un pulgar rígido en oposición y unos dedos flexibles, los cuáles eran flexionados pasivamente, éstos se podían fijar mediante un mecanismo de trinquete y además tenían una muñeca movable [10].



*Figura 1. Mano protésica de Alt-Ruppin. [11]*

Con el paso de los años, conforme el desarrollo tecnológico y el entendimiento de la biomecánica iban en aumento se fueron realizando grandes mejoras en el diseño. Pero no es hasta 1960 cuando surge en Rusia la prótesis mioeléctrica (ver **Figura 2**).



*Figura 2. Fotografía de la primera prótesis mioeléctrica. [11]*

El movimiento de esta prótesis es activado por los impulsos bioeléctricos generados durante la contracción de la masa muscular del muñón [11] .

Los primeros dedos protésicos comerciales aparecen sobre los años 80, de la mano de la marca escocesa ProDigits (powered prosthetic digit system). Este tipo de prótesis fueron presentadas para diferentes niveles de amputación de mano ya sea parcial o total. Este diseño fue mejorado en el año 1994, consiguiendo una mejora en el tamaño de los motores, engranajes y en la fuente de energía [12].



*Figura 3. Prótesis ProDigits versión años 80 (a) y actual (b). [13]*

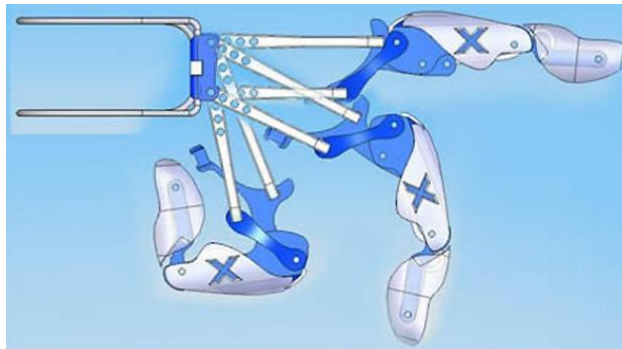
El ejemplo mostrado en la **Figura 3** es la solución a una pérdida parcial de mano. Podemos observar las diferencias y mejoras entre las dos versiones, destacando las soluciones a los problemas de espacio, de estética y de control.

Algunos años más tarde, en 2005, se desarrolla la prótesis mecánica X finger con la intención de solucionar específicamente amputaciones parciales de dedo (**Figura 4**).



*Figura 4. Prótesis mecánica X finger. [13]*

Este dispositivo tiene la peculiaridad de no utilizar dispositivos electrónicos para su accionamiento. Su funcionamiento se basa en la utilización del mecanismo de cuatro barras o también llamado cuadrilátero articulado [12], consiguiendo así imitar el movimiento natural de las falanges tal y como lo podemos ver en la **Figura 5**.



*Figura 5. Mecanismo X finger [13]*

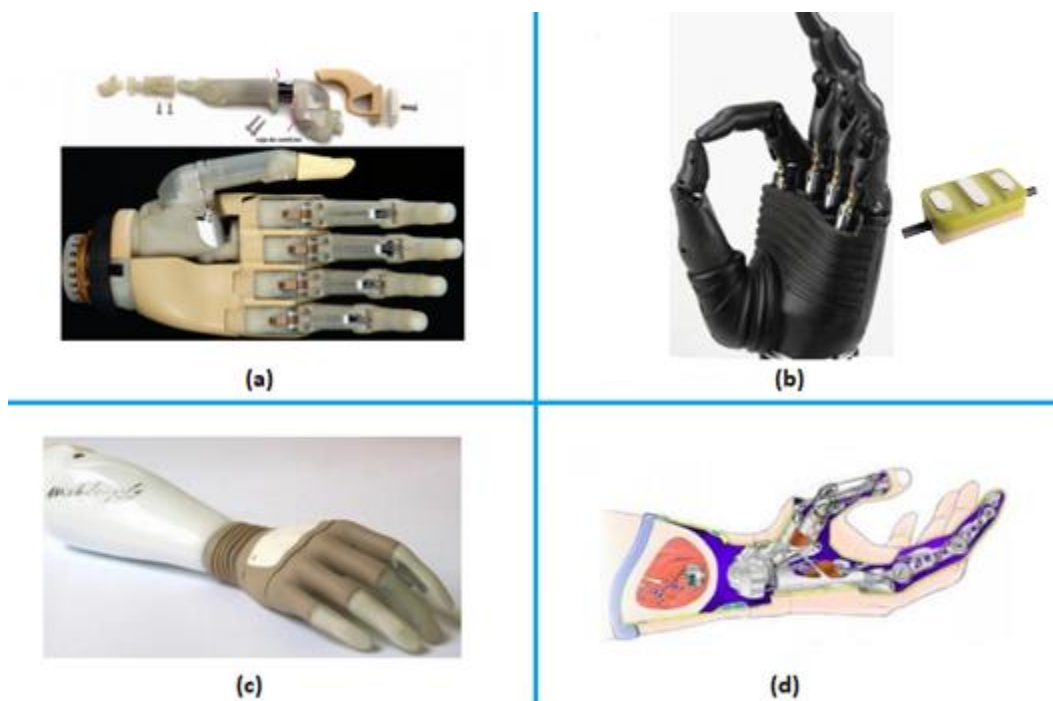
El funcionamiento de esta prótesis se debe a que cada una de sus falanges dispone de articulaciones naturales, que se activan cuando el dedo residual se mueve, evitando así el uso de motores y sistemas de control por cada grado de libertad.

La prótesis se recubre luego por una piel sintética de silicona, emulando casi a la perfección, la funcionalidad y la precisión de un dedo real humano no amputado [13].



## 2.2 Desarrollos actuales

En estos últimos años, el desarrollo tecnológico ha dado pasos exponenciales. Esto ha permitido la fabricación de prótesis avanzadas comerciales (**Figura 6**), las cuales permiten el control independiente de cada dedo, tanto en su posición, presión o velocidad, como es el caso de las prótesis I-Limb, que es la más difundida de su clase en la actualidad, o la prótesis Bebionic, que incluye además materiales inteligentes. También se pueden encontrar prótesis con gran fuerza de agarre y resistencia al agua, como es el caso de la prótesis Michelangelo o prótesis que se unen quirúrgicamente insertando electrodos en las terminaciones nerviosas del muñón del paciente para devolverle así la sensación de presión y temperatura, como es el caso de la Cyber Hand [9].



*Figura 6. Prótesis comerciales. I-Limb (a), Bebionic (b), Michelangelo (c) y Cyber Hand (d). [10]*

Los modelos de la **Figura 6** permiten los movimientos de flexión – extensión en el dedo pulgar y llevan implementado además un control mioeléctrico que gobierna la posición de un motor de corriente continua. Estas son hoy, el tipo de prótesis que presentan el más alto grado de rehabilitación, ya que no requieren de arneses o medios de suspensión [8].

Sin embargo, se puede observar observar que estas manos están diseñadas principalmente para personas con amputaciones por encima de la muñeca. Las prótesis para amputaciones parciales de mano, como por ejemplo un dedo protésico, han sido

difíciles de conseguir, sobre todo por el limitado espacio disponible para los mecanismos y demás componentes electrónicos. Para solucionar esto, los fabricantes se centraron en conseguir que el mecanismo de accionamiento se encuentre contenido en los dedos.

Actualmente, para las personas con una amputación parcial de la mano, solo hay dos prótesis mioeléctricas disponibles comercialmente. Una de ellas es el modelo i-Digits Quantum de la empresa TouchBionics y la otra el modelo Vincent Partial Hand de la empresa Vincent systems. Estas prótesis cuentan con unos dedos protésicos motorizados individualmente.

A continuación, pasaremos a analizarlos más detenidamente, ya que servirán de inspiración para el sistema que se desarrollará en este trabajo fin de grado.

### 2.2.1 i-Digits Quantum

i-Digits Quantum es una prótesis parcial comercial de mano mioeléctrica, adecuada para personas con ausencia de 1 a 5 dedos donde el nivel de ausencia es distal a la muñeca y proximal a la articulación metacarpofalángica [14].

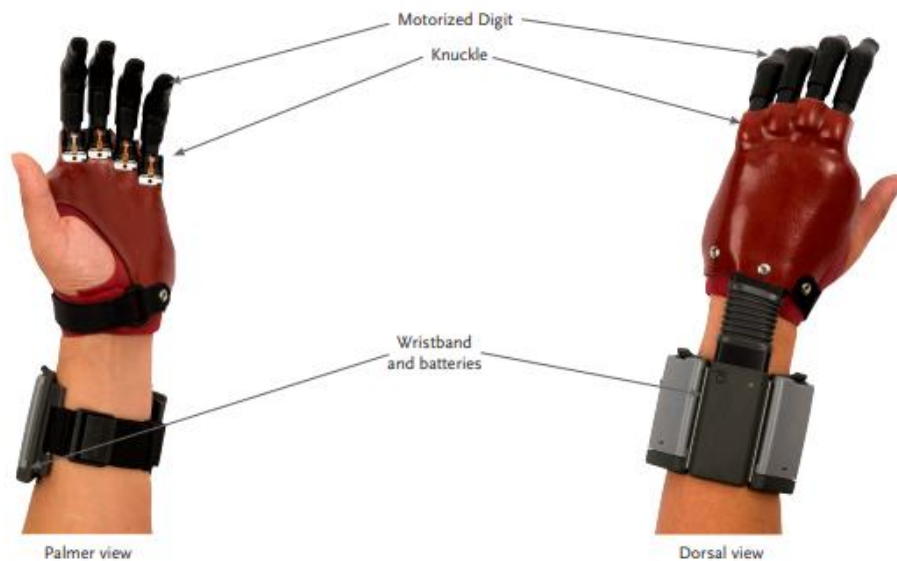
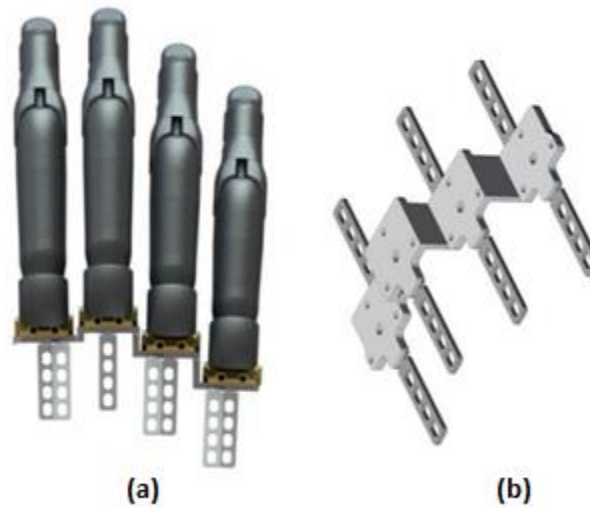


Figura 7. Prótesis i-Digits Quantum [15]

Las características más destacables son:

- Permite configurar hasta 32 agarres automatizados. El cambio entre un agarre y otro se da a través de gestos sencillos.
- Llevan un brazalete en la cual están alojadas las baterías y el microprocesador.
- Es posible sustituir desde uno hasta cinco dedos en cualquiera de las posiciones (incluido el pulgar)
- Permite controlar la velocidad de apertura/cierre de los dígitos, así como la fuerza que estos aplican.

Estas prótesis están disponibles en distintos tamaños para ofrecer al usuario un ajuste y unas funcionalidades óptimas. Además, cada dígito va montado de forma independiente sobre una placa de montaje de nudillos (**Figura 8**).



*Figura 8. Dígitos de distintos tamaños(a) y placa de montaje de nudillos (b). [15]*

Los dispositivos i-Digits se pueden controlar mediante electrodos remotos de perfil bajo (LP) o sensores de fuerza resistivos (FSR), todos ellos diseñados específicamente para su uso en dispositivos i-Digit (**Figura 9**).



*Figura 9. Electrodo de perfil bajo (a) y Sensor de fuerza resistivo (b). [15]*

En el control por electrodos remotos, la señal electromiográfica es medida a través de unos electrodos de perfil bajo, los cuales captan la actividad de los músculos restantes dentro de la mano residual para controlar los movimientos de los dedos.

Por otra parte, el control por FSR (Force Sensitive Sensor) se basa en los movimientos físicos de la mano residual para controlar los movimientos de los dígitos. Los FSR se utilizan idealmente en situaciones en las que hay un dedo remanente presente en la mano afectada que puede flexionarse y extenderse.

### 2.2.2 Vincent Partial Hand [15]

Esta prótesis es comercial al igual que la anterior y cuenta con dedos individuales motorizados. Disponen de un sistema modular, por el cual los dedos, mandos y pilas se pueden colocar individualmente para realizar una reconstrucción anatómica de la mano, en la medida de lo técnicamente posible [15].



Figura 10. Vincent Partial Hand. [16]

Sus características más significativas son las siguientes [16]:

- Dispone de multitud de agarres, con una gran precisión y control en velocidad y fuerza.
- Se adaptan a diferentes tamaños de palma y longitud de dedos.
- Los motores están incorporados en la base del dedo de forma que los componentes ocupen el menor espacio posible
- Están fabricados en una combinación de acero inoxidable y aleación de magnesio lo que la hace robusta y ligera.
- Sistema de feedback sensorial con un botón vibratorio que informa al usuario sobre el estado del agarre sustituyendo al tacto.

El control de estas prótesis puede hacerse a través de sensores electromiográficos (EMG) o también mediante sensores de fuerza resistivos (FSR). Estos sensores al igual que la batería y el microprocesador van dentro de un brazalete (**Figura 11**).



*Figura 11. Componentes y brazaletes del Vincent Partial Hand. [16]*

La mano parcial Vincent tiene un pulgar protésico que lleva dentro un motor de corriente continua, el cual permite realizar el movimiento de flexión/extensión con energía eléctrica. Sin embargo, el pulgar protésico no es capaz de realizar el movimiento de abducción/aducción, debido en gran parte a su diseño modular y compacto (**Figura 12**).



*Figura 12. Pulgar de la Mano Parcial Vincent. [16]*

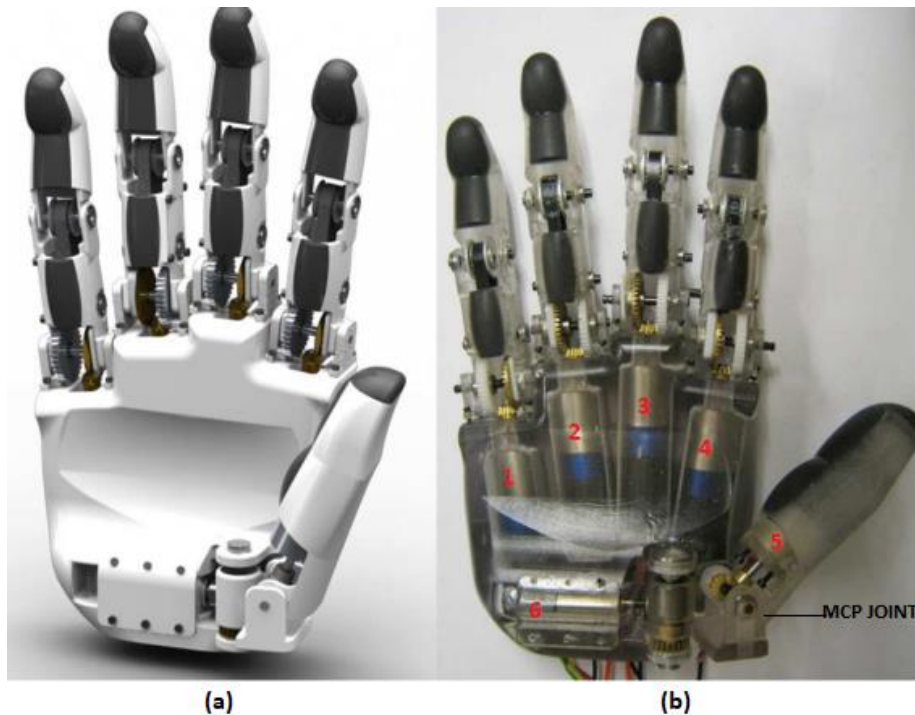
### 2.2.3 6 DOF Open Source Hand:

El diseño de esta mano es de código abierto a diferencia de los dos modelos comerciales estudiados anteriormente, los cuales tienen sus diseños patentados.

Este diseño lleva un motor de corriente continua alojado dentro del pulgar, por lo cual será de especial interés en este TFG. El propósito general de sus creadores es el de desarrollar una mano física, que permita desarrollar, implementar y probar diversos algoritmos de control independientes en cada dedo.

Los parámetros de diseño en este proyecto son [17]:

- Diseño para realizar cualquier tipo de agarre (punta, palmar, lateral, cilíndrica, esférica y gancho).
- Control independiente de cada dedo, con un motor para cada uno de los dedos, y dos motores en el pulgar (uno controla la flexión-extensión y otro la abducción-aducción).
- Costes reducidos para poder hacer más accesible el uso de la prótesis.
- Carcasas de los dedos diseñadas con una cavidad interna para que los sensores de fuerza puedan incrustarse en las yemas de los dedos.



*Figura 13. Modelo Renderizado (a) y prototipo terminado (b) de la mano de seis grados de libertad. [18]*

Como podemos observar en la **Figura 13**, el diseño de esta mano dispone de seis motores y diez articulaciones móviles.



En este diseño, cada dedo a excepción del pulgar, lleva un motor Faulhaber MM1724-T-006-SR, con un reductor con relación 246:1. El eje del reductor impulsa un juego de engranajes cónicos 3:1 accionando así el giro de la articulación metacarpofalángica (MCP). A su vez, este par en la articulación MCP se transmite a través de un conjunto de engranajes rectos con relación 2:1 a un sistema de correa de distribución que activa la articulación interfalángica proximal (PIP). Todo esto lo podemos ver en la **Figura 14**.

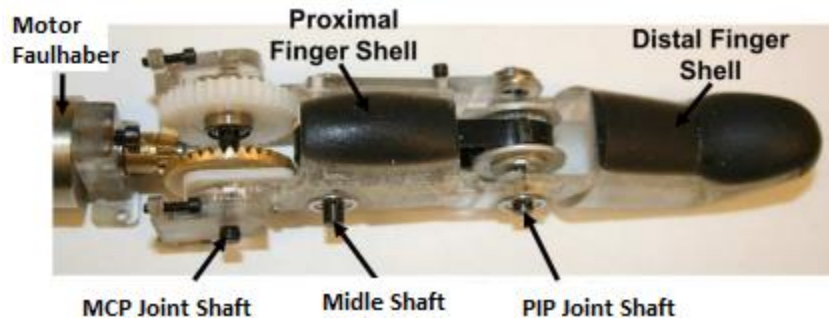


Figura 14. Mecanismo de accionamiento de los dedos a excepción del pulgar. [18]

Respecto al dedo pulgar (objetivo de este proyecto fin de grado) podemos ver la solución adoptada en la **Figura 15**.

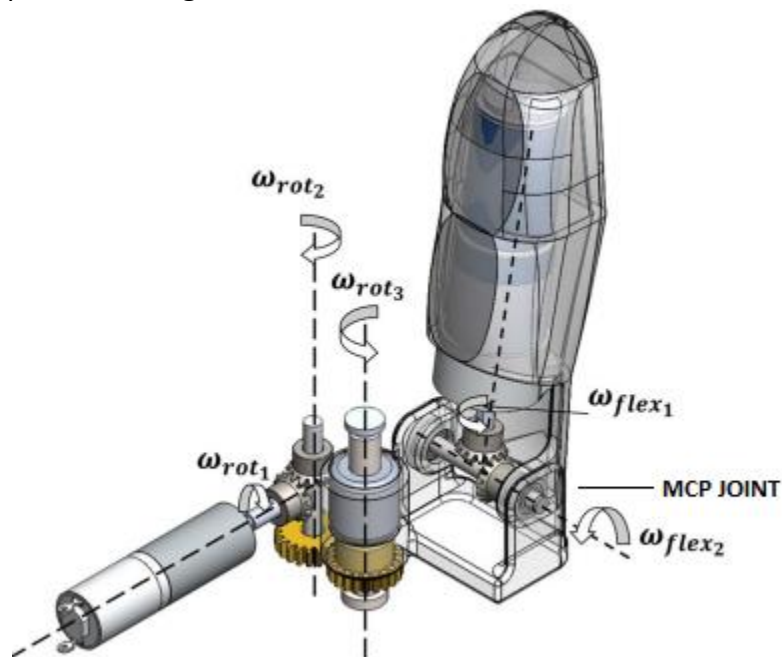


Figura 15. Esquema del funcionamiento del pulgar con dos motores. [18]

Para el movimiento del pulgar se dispone de dos motores. Uno de ellos alojados en la carcasa del dedo que acciona el movimiento de flexión/extensión en la articulación MCP que corresponde con  $\omega_{flex1}$ . Por otro lado, el motor que acciona el movimiento de abducción/aducción se alojaría en la palma y correspondería con  $\omega_{rot1}$ .

# CAPÍTULO 3.- ANATOMÍA DE LA MANO



En este capítulo se hará un estudio sobre la anatomía de la mano, haciendo especial hincapié en el dedo pulgar. Es imprescindible conocer los fundamentos anatómicos del dedo pulgar para diseñar un prototipo lo más funcional posible. También se estudiará los ángulos máximos de los movimientos del dedo pulgar y el tamaño de los mismos.

### 3.1 Estudio de la mano

Desde un punto de vista fisiológico, la mano representa la extremidad efectora del miembro superior. Sin embargo, esta no es sólo un órgano de ejecución, es también un receptor sensorial extremadamente sensible y preciso cuya información es indispensable para retroalimentar su propia acción [12].

La complejidad de la mano, y en consecuencia de los dedos, radica en la compacidad y el pequeño tamaño que presentan frente a su alta destreza, a través de la cual se permiten numerosos modos de agarre. Estas características son implementadas por la mano a través del uso de tendones, los cuales transmiten el movimiento generado por los músculos ubicados en el antebrazo.

#### 3.1.1 Características funcionales

Hay tres tipos principales de huesos en la mano entre los cuales se incluyen: falanges, huesos metacarpianos y carpianos (**Figura 16**).

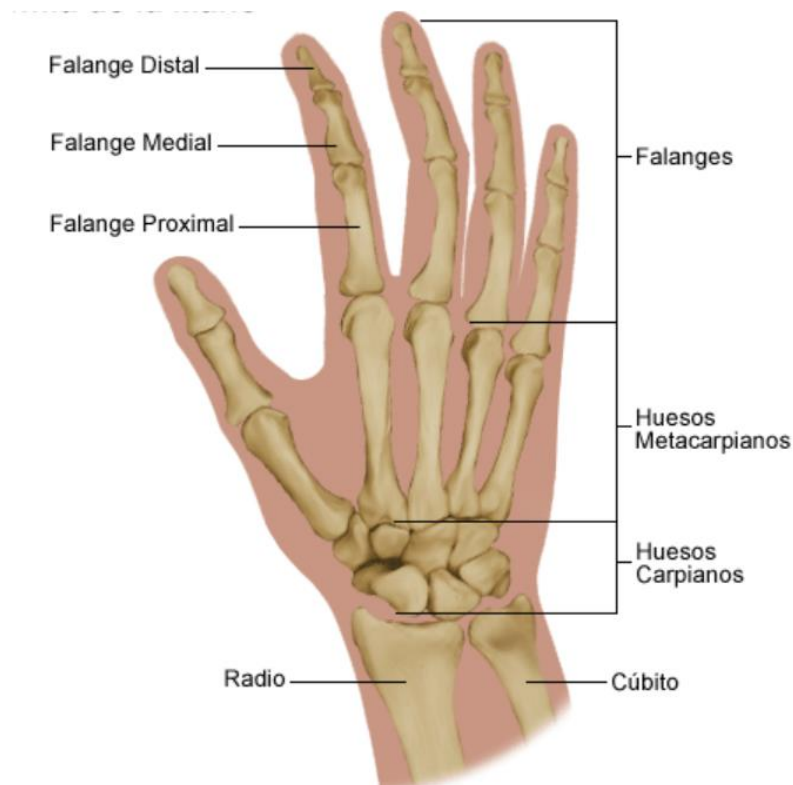
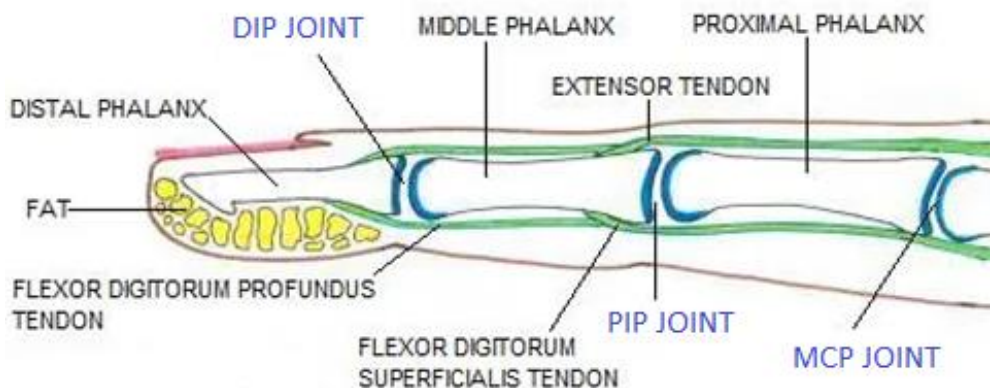


Figura 16. Huesos de la mano. [29]

- **Falanges:** Son los 14 huesos que están en los dedos de cada mano y también en los dedos de cada pie. Cada dedo tiene tres falanges (distal, media y proximal). Excepto el pulgar que solamente tiene dos (falange proximal y distal).
- **Huesos metacarpianos:** Son los cinco huesos que componen la parte media de la mano
- **Huesos carpianos:** Corresponden con los ocho huesos que forman la muñeca. Estos huesos están conectados a dos huesos del brazo (el cúbito y el radio).

Dentro de la mano se pueden encontrar numerosos músculos, ligamentos, tendones y vainas (**Figura 17**). Los músculos son las estructuras que pueden contraerse, permitiendo el movimiento de los huesos de la mano. Los tendones conectan los músculos del brazo o la mano con el hueso para permitir el movimiento. Los ligamentos son tejidos fibrosos que ayudan a unir las articulaciones de la mano. Las vainas son estructuras tubulares que rodean parte del dedo y se encargan de mantener a los tendones en posición y cerca de los huesos. Además, hay arterias, venas y nervios dentro de la mano que proporcionan flujo sanguíneo y sensación a la mano y los dedos [18].

Cada dedo, a excepción del pulgar, tiene tres huesos (falanges) y tres articulaciones: la articulación metacarpofalángica (MCP) que une el dedo a la mano, la articulación interfalángica proximal (PIP) y la articulación falángica distal (DIP). Todos los dedos están conectados a los metacarpianos por cinco articulaciones MCP (metacarpofalángicas).



*Figura 17. Anatomía del dedo. [19]*

Como se puede observar en la **Figura 17**, los tendones corren tanto en el lado palmar como en el dorsal para permitir que los dedos se flexionen y extiendan, pero se entrelazan a medida que cruzan las articulaciones MCP, PIP y DIP.

### 3.1.2 Anatomía del dedo pulgar

Como hemos comentado anteriormente, el dedo pulgar tiene únicamente dos huesos (falanges) es por ello que tiene una movilidad más compleja gracias a los dos grados de libertad de la articulación trapezometacarpiana.

Las articulaciones de la columna del pulgar son cuatro [19]:

1. La trapezoescafoidea (TE): Es de tipo artrodia. Permite al trapecio realizar un corto desplazamiento hacia delante, se esboza un movimiento de flexión de escasa amplitud.
2. La trapezometacarpiana (TM): Es de tipo Silla de montar y está dotada de dos grados de libertad.
3. La metacarpofalángica (MCP): Es de tipo Condilea y posee dos grados de libertad. Condilea.
4. La interfalángica (I.P) es de tipo Troclear y tiene un grado de libertad.

En total CINCO GRADOS DE LIBERTAD necesarios y suficientes para realizar la oposición del pulgar.

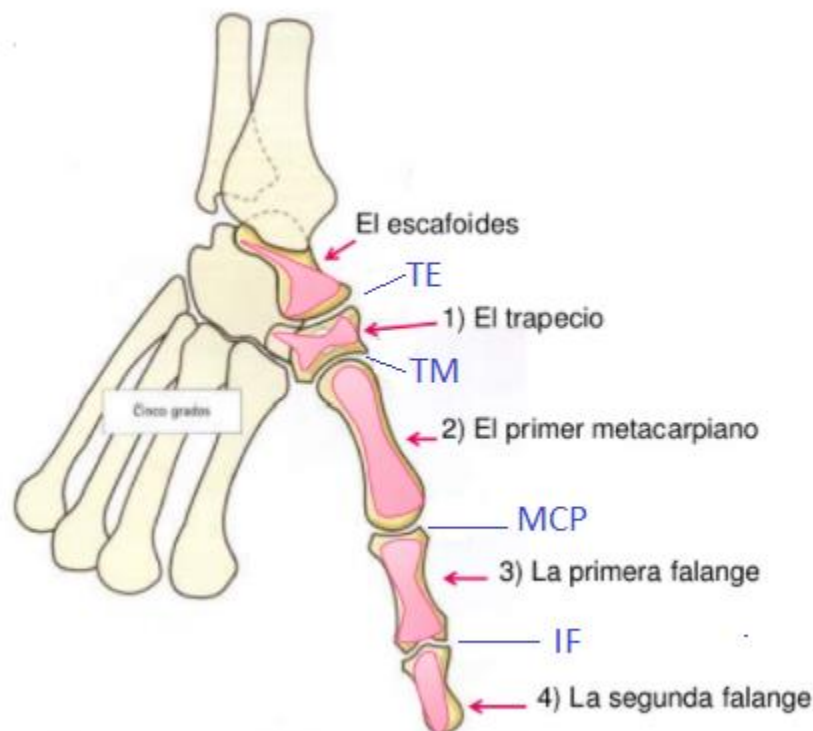


Figura 18. Falanges y articulaciones del dedo pulgar. [19]

En nuestro caso particular solo contemplaremos la articulación metacarpofalángica ya que será la que permita el movimiento objetivo (flexión/extensión) de este trabajo fin de grado.

### 3.2 Antropometría del dedo pulgar

Para dotar a nuestro prototipo de mayor funcionalidad y naturalidad, será necesario conocer los ángulos de flexión/extensión que se pueden alcanzar, así como las dimensiones del dedo pulgar.

#### 3.2.1 Ángulos máximos de la articulación MCP del pulgar

La articulación metacarpofalángica se encuentra entre la cabeza del primer metacarpiano, y la base de la falange proximal (**Figura 19**). Permite los movimientos de flexión y extensión del dedo pulgar. Además, esta articulación también permite el movimiento de rotación axial que ayuda a la oposición del pulgar.

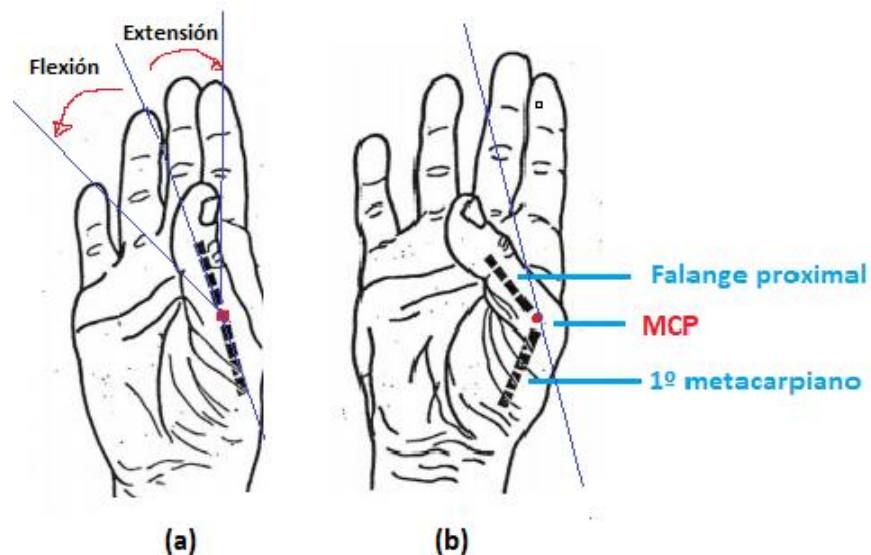


Figura 19 . Pulgar en estado neutral (a) y pulgar flexionado (b). [20]

Si se toma como referencia la postura neutral del dedo pulgar de la **Figura 19**. Según [20] los ángulos de flexión/extensión obtenidos a través de un goniómetro electrónico son los siguientes:

	Valor medio (°)	Rango (°)
<i>Flex. Máximo</i>	-42,3	-61 / 23,1
<i>Ext. Máximo</i>	57,7	44,0 / 69,0

Tabla 1 Valores de los ángulos del dedo pulgar. [20]

### 3.2.2 Tamaños del dedo pulgar

Como vimos anteriormente, el dedo pulgar tiene solo dos falanges (**Figura 20**).



*Figura 20. Falanges del pulgar.*

Para determinar el tamaño promedio del dedo pulgar, se hará uso del estudio [21], en el cual nos aparece el tamaño medio de cada una de las dos falanges del dedo pulgar diferenciados por intervalos de edad.

Edades	Falange proximal		Falange Distal	
	Mano der	Mano izq	Mano der	Mano izq
18-19	41±7	41±7	32±4	32±4
20-24	40±6	40±6	32±3	32±3
25-29	42±7	42±7	34±4	34±5
30-34	41±9	41±8	32±5	31±5
35-39	39±8	38±8	32±4	31±5
40-44	40±10	40±11	31±9	31±8
45-49	44±4	43±6	33±3	32±4
50-54	41±7	40±7	32±4	32±4
55-59	39±11	40±12	33±4	33±4
60-64	38±9	38±10	33±4	33±5

*Tabla 2. Longitud promedio de las falanges del dedo pulgar. [13]*

Para este TFG haremos uso de las medidas de las falanges para el intervalo de 20-24 años.

# CAPÍTULO 4.- ESPECIFICACIONES DEL DEDO PULGAR PROTÉSICO

En este capítulo se definirán los requerimientos que se tendrán en cuenta a la hora de diseñar el dedo pulgar protésico. El objetivo será reproducir los movimientos de flexión/extensión del dedo pulgar de la forma más natural posible. Para ello se usarán de base los conocimientos anatómicos expuestos en el capítulo anterior y la experiencia recogida al analizar el estado del arte.

#### 4.1 Parámetros básicos de diseño

Con la definición de estos parámetros, se podrá comenzar a dar forma al producto final, ya que determinarán el tipo de motor y a su vez también los espacios que restringirán el diseño estructural del dedo.

- **Dimensiones**

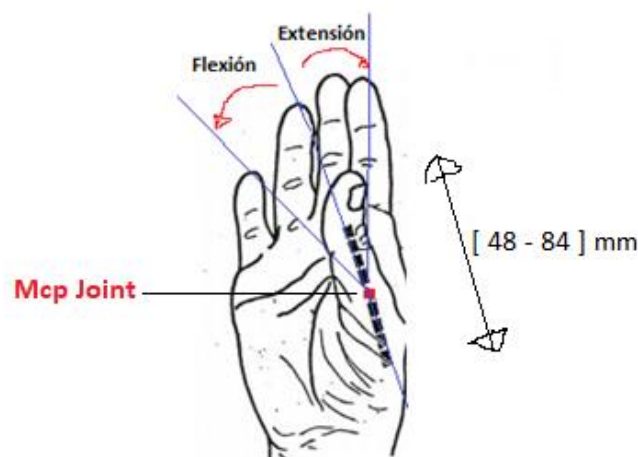
De acuerdo al capítulo anterior se determina un rango de tamaños basado en la media de ambos, tomando la edad de referencia en 20 años. Este rango queda entre 48 mm y 84 mm (**Figura 21**).

- **Ángulo de la articulación del dedo pulgar**

Basándonos de nuevo en el capítulo anterior, el ángulo correspondiente a la flexión-extensión del dedo pulgar se dará entre los -42 y los 57,7 grados.

- **Velocidad**

De acuerdo a [22], se ha determinado que las velocidades medias para tareas como la selección y colocación en actividades diarias, se mueven en un rango entre 2,5 rad/s y 3,5 rad/s.



*Figura 21. Dimensiones y movimiento de diseño. [20]*

- **Fuerza de agarre**

Para conocer la fuerza que deberá de alcanzar nuestro dedo protésico, se tomará de referencia los estudios de Krausz [17]. Krausz nos dice que para tener una mano protésica lo más funcional posible, cada dedo deberá tener una fuerza de agarre de  $9,28 \pm 2,46$  N de media.

Si observamos la **Tabla 3** podemos comprobar que los valores de fuerza elegidos anteriormente están dentro del intervalo de fuerza que manejan los dedos de las prótesis comerciales [6].

Individual finger holding force at tip.

Finger	Force (N)	Number of Trials	Standard Deviation
Vincent Large (ring, middle, and index)	4.82 or 8.44*	14 or 8*	0.8 or 1.3*
Vincent Small (little)	3.00	2	0.1
iLimb Large (middle)	7.66	2	0.2
iLimb Med (index/ring)	5.39	4	0.1
iLimb Small (little)	5.17	2	0.1
iLimb Pulse Med (index)	4.15 or 6.54*	1	—
iLimb Pulse Large (middle)	3.09 or 6.24*	2 or 2*	0.7 or 0.4*
iLimb Pulse Med (ring)	6.43 or 11.18*	2 or 2*	0 or 0.3*
iLimb Pulse Small (little)	4.09 or 8.56*	2 or 2*	0.1 or 0*
Bebionic (index)	12.47	1	—
Bebionic (middle)	12.25	2	1.0
Bebionic (ring)	12.53	2	1.1
Bebionic Small (little)	16.11	2	0.2
Bebionic v2 Large (ring, middle, and index)	14.5	2	1.2

\*Holding force after pulse mode.  
med = medium.

**Tabla 3.** Fuerzas de agarre en cada dedo protésico comercial. [7]

- **Proceso de fabricación**

Dado el bajo número de piezas a fabricar y su variedad, el proceso de fabricación óptimo que se utilizará será la fabricación aditiva. La fabricación aditiva o por adición, comúnmente conocida como impresión 3D, es una tecnología actualmente en auge que consiste en la producción de nuevas piezas depositando capas del material según el modelo diseñado mediante un programa de modelado en el ordenador. Ofrece una gran flexibilidad y es la opción perfecta para pequeñas series intermitentes de producción. Este tipo de fabricación permite obtener piezas más personalizadas y con mayor nivel de detalle. Además, disminuye el tiempo de producción manual, ya que tan solo sería necesaria la supervisión durante la impresión de la pieza y no requiere tratamientos intermedios, más allá de la reposición de material si fuese necesario.

- **Marco legal**

La calidad de los productos está regulada por ciertas legislaciones según en la región en la que se quiera comercializar. Existen normas de carácter general. Sin



embargo, no serán las únicas. De acuerdo con la naturaleza del producto, será requisito fundamental atenerse a una directiva legal de obligado cumplimiento. Concretamente en España, la organización que rige la normativa de los productos sanitarios es la Agencia Española de Medicamentos y Productos Sanitarios (AEMPS). Dentro de la clasificación hecha por la AEMPS, la prótesis de dedo será un producto de categoría 9-Instrumentos reutilizables y dentro de la misma, de clase o genérico: Instrumental y prótesis de prueba [23]. En consecuencia, deberá de cumplir los requisitos descritos en la directiva de dispositivos médicos: 93/42/EEC Medical devices directive (MDD) [23]. Para poder comercializar el producto, además de cumplir con los requisitos ya mencionados, debe ser una empresa autorizada la que certifique que ciertamente se alcanzan los requerimientos. La institución ENAC puede ser esa entidad de certificación, ya que posee las certificaciones EN ISO/IEC 17021 [24].

## 4.2 Estructura conceptual del dedo protésico

De acuerdo a la antropometría del dedo pulgar, las características definidas en el punto anterior y las conclusiones extraídas del estado del arte, se puede realizar una conceptualización de las partes que compondrán el producto final.

- **Dedo pulgar:** en primer lugar, se realizará una estructura que simulará las funciones del dedo humano, tales como agarrar, señalar o presionar superficies.
- **Soporte móvil:** este elemento servirá para integrar el movimiento de flexión/extensión en el dedo pulgar protésico.
- **Sistemas de transmisión:** finalmente para poder transmitir el movimiento de los motores al propio dedo se requerirán sistemas de transmisión como engranajes, o sistemas sinfín-corona.
- **Sensor Electromiográfico (EMG):** servirá como señal de control a la hora de activar los motores. Se elegirá el sensor EMG más adecuado a nuestros objetivos.
- **Conjunto Motor-Reductor-Encoder:** puesto que el diseño atiende a la realización de flexión/extensión con unos parámetros determinados, será necesario disponer de un conjunto Motor-Reductor que cumpla con las especificaciones del proyecto. También se necesitará un codificador (Encoder) para permitir la retroalimentación y las pruebas de la posición del motor.
- **Controlador de motores (Esclavo):** Se necesitará un controlador que gobierne la posición del motor a través de un control PID.
- **Controlador Maestro:** Este elemento será el más importante, ya que se encargará de leer, sincronizar y ejecutar todas las instrucciones del algoritmo de control.

# CAPÍTULO 5.-

## PROTOTIPADO Y CONTROL DEL DEDO PULGAR PROTÉSICO

Este capítulo es sin duda, el más importante de este TFG. Hay que tener en cuenta que este proyecto forma parte de uno previo, basado en el desarrollo del resto de dedos de la mano. En dicho proyecto se optó por usar un motor de corriente continua de la casa suiza Maxon con su propia controladora (EPOS 4 Compact). Esta selección nos servirá como punto de partida para la búsqueda de los demás componentes que nos ayudarán a alcanzar las especificaciones fijadas en el capítulo anterior. Teniendo esto en cuenta, se diseñará y fabricará el dedo pulgar protésico. Después se seleccionarán los componentes electrónicos necesarios para implementar un algoritmo de control mioeléctrico en la prótesis. Por último, se realizarán unas pruebas para observar la funcionalidad de la prótesis y se analizarán los resultados obtenidos.

### 5.1 Diseño mecánico y fabricación

En el laboratorio en el que se ha desarrollado el TFG, se dispone de un sistema modular Maxon compuesto por un motor de corriente continua acoplado a un reductor planetario y a un encoder magneto-resistivo. Estos dispositivos fueron utilizados para una aplicación similar y servirán como punto de partida para este proyecto (**Figura 22**).

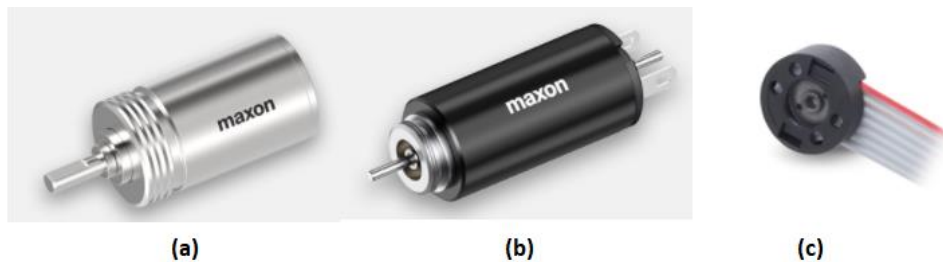


Figura 22. Conjunto: Reductor (a) - motor (b) y encoder (c). [25]

A continuación, se realizará una pequeña descripción de los componentes con los que se ha partido para la realización de este proyecto:

- **Motor de corriente continua**

Como se comentó anteriormente, se dispone de un motor RE8 de Maxon. Este tipo de motores RE de Maxon, están equipados con imanes permanentes y disponen de un sistema de bobinado sin núcleo o sin hierro. El imán permanente se encuentra debajo del devanado, y la carcasa exterior del motor es el retorno magnético, lo que lo convierte en un diseño muy compacto (**Figura 23**).

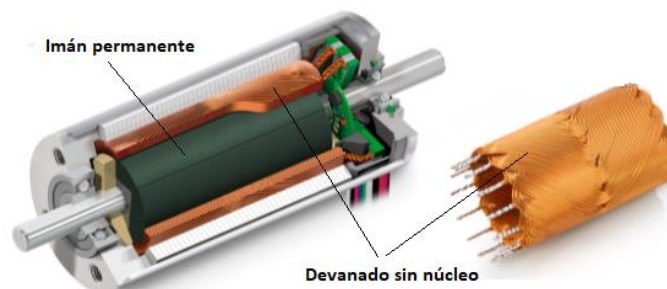


Figura 23. Motor CC con imanes permanentes y bobinado sin núcleo. [26]

Estos motores se pueden encontrar en muchos dispositivos donde se requieren prestaciones de servo de alta velocidad, como la robótica; Unidades de rayos X; Prótesis; Equipo de análisis de laboratorio; Imágenes térmicas; Unidades de radar y plataformas de registro de alta gama.

Las características más importantes son [26]:

- El bobinado del motor tiene una inercia muy baja ya que no hay núcleo de hierro para acelerar. El motor puede acelerar y desacelerar a la velocidad máxima en milisegundos, lo que ofrece un rendimiento de muy alta velocidad.
- Característica lineal de velocidad / par. Lo que proporciona un control lineal de velocidad y par dentro de una alta relación potencia/tamaño.
- Eficiencia. Los motores son altamente eficientes al 90%. El motor no tiene que romper los campos magnéticos. El devanado casi puede llenar el espacio de aire entre el imán y su retorno magnético.

En este caso se dispone del motor RE8 de Maxon con número de referencia 347724, el cual dispone de las siguientes características [27].


		Referencia
		B con cables
		A con terminales
<b>Datos del motor</b>		
<b>Valores a tensión nominal</b>		
1	Tensión nominal	V 4.2
2	Velocidad en vacío	rpm 14200
3	Corriente en vacío	mA 11.2
4	Velocidad nominal	rpm 4480
5	Par nominal (máx. par en continuo)	mNm 0.624
6	Corriente nominal (máx. corriente en continuo)	A 0.237
7	Par de arranque	mNm 0.932
8	Corriente de arranque	A 0.34
9	Máx. rendimiento	% 67
<b>Características</b>		
10	Resistencia en bornes	$\Omega$ 12.3
11	Inductancia en bornes	mH 0.09
12	Constante de par	mNm/A 2.74
13	Constante de velocidad	rpm/V 3490
14	Relación velocidad/par	rpm/mNm 15700
15	Constante de tiempo mecánica	ms 6.3
16	Inercia del rotor	gcm <sup>2</sup> 0.0383

Figura 24. Extracto de las características del motor RE8. [26]

El motor RE 8 tiene un diámetro exterior de 8mm, por lo tanto, es ideal para el proyecto propuesto ya que nos ayudará a fijar dos características: dimensiones muy reducidas y tipología de corriente continua. Además, el rango de velocidad del motor oscila entre 4480 rpm (velocidad nominal) y 14200 rpm (velocidad en vacío).

- **Reductor Planetario**

Los reductores planetarios de Maxon están fabricados para ser utilizados con los motores de la familia RE. Este tipo de reductores son particularmente adecuados en la transmisión de pares elevados (**Figura 25**).

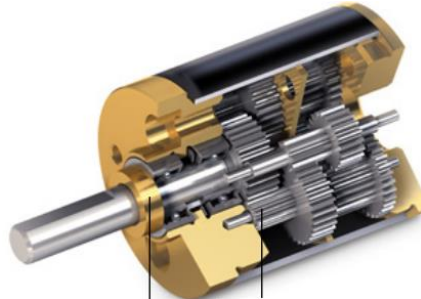


Figura 25. Reductor Planetario RE [28]

Este tipo de reductor planetario destaca por su altísima transmisión de fuerza en un formato muy pequeño. Es capaz de otorgar grandes velocidades y pares con poco ruido y holgura.

En este caso se dispone del engranaje planetario GP 8 con numero de referencia 474124 y dispone de las siguientes características [28].

Datos del Reductor			Referencia
1	Reducción		474124
2	Reducción absoluta		36:1
3	Diámetro máx. del eje del motor	mm	36
4	Número de etapas		0.65
5	Máx. par en continuo	Nm	2
6	Máx. par admisible de forma intermitente	Nm	0.008
7	Máx. rendimiento	%	0.012
8	Peso	g	76
9	Holgura media en vacío	°	3.2
10	Momento de inercia	gcm <sup>2</sup>	2.4
11	Longitud reductor L1	mm	0.004
			8.3

Figura 26. Extracto de las características del reductor planetario GP8. [28]

Este reductor tiene un diámetro exterior de 8mm y nos proporcionará una reducción de 36 a 1.

- **Encoder MR**

El encoder MR de Maxon es del tipo magnético-resistivo (MR) ya que posee un disco magnético multipolar montado en el eje del motor que genera una variación de tensión sinusoidal en el sensor MR. Este tipo de encoder se instalan principalmente en motores con eje pasante. En la siguiente figura se muestra el interior de este tipo de encoders, además se señalan las partes más importantes que la componen (**Figura 27**).

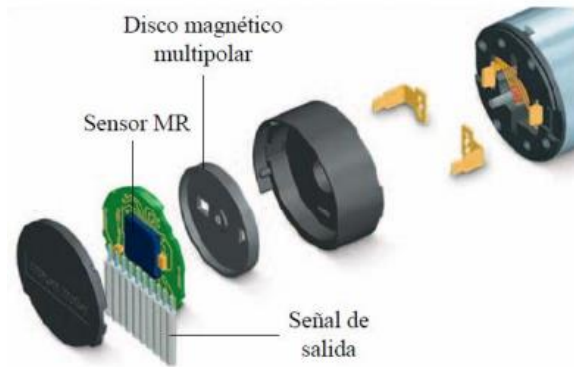


Figura 27. Encoder Magneto Resistivo MR 100 [29].

En este caso se dispone del Encoder MR tipo S con numero de referencia 468998 y dispone de las siguientes características [27]:



Tipo	Referencia
Número de pulsos por vuelta	334910
Número de canales	100
Máx. frecuencia de funcionamiento (kHz)	2
Máx. velocidad (rpm)	100
	60 000

Figura 28. Extracto de las características del Encoder MR. [29]

Este encoder dispone de dos canales y tiene una resolución de 100 pulsos por vuelta. Esto le hace una excelente elección para aplicaciones de posicionamiento de alta precisión y control de velocidad.

Teniendo estos componentes como punto de partida. Se calculará la relación de transmisión y el tipo de engranaje que se necesitará para cumplir con las especificaciones, tanto de velocidad como de fuerza. Para ello se realizará un análisis cinemático y dinámico.

### 5.1.1 Análisis cinemático

Tal y como se fijó en el apartado de parámetros básicos de diseño, la velocidad de flexión/extensión del dedo pulgar se sitúa entre los 2 rad/s y 3,2 rad/s.

En este caso, para el análisis cinemático se usará el valor de 2.6 rad/s que corresponden a una velocidad de flexión/extensión de 24,82 rpm en el dedo. Estos 24,82 rpm será la velocidad objetivo a alcanzar (**Figura 29**).

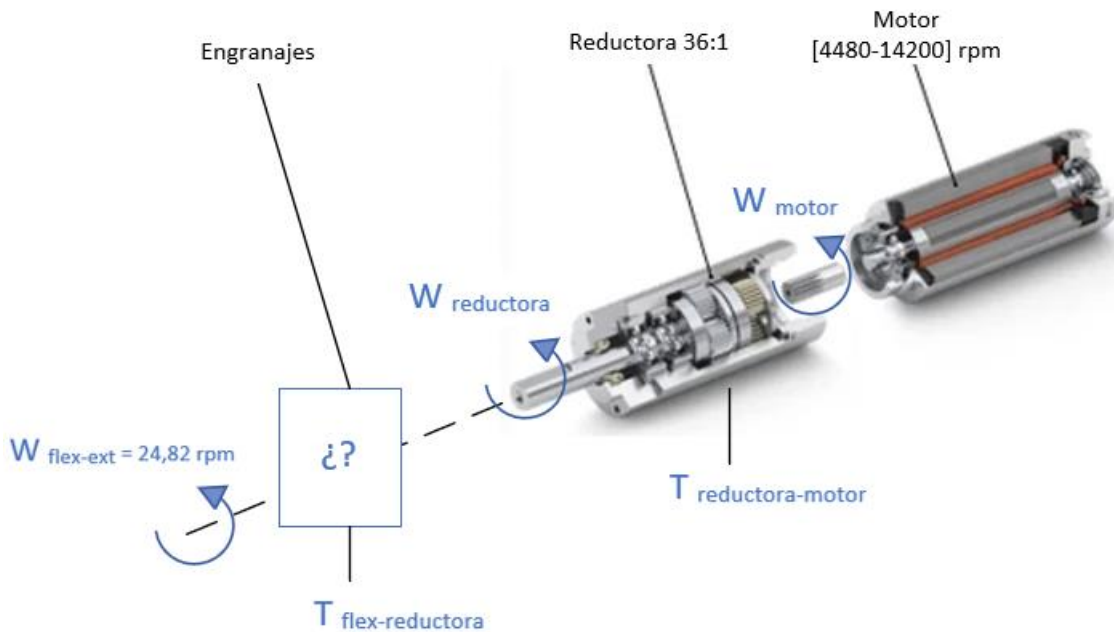


Figura 29. Esquema Motor RE8 + Reductora + Engranajes. [28]

Para los cálculos usaremos la siguiente nomenclatura:

- $W_{motor}$  será la velocidad de giro en el eje del motor.
- $W_{reductora}$  será la velocidad de giro a la salida del reductor planetario.
- $W_{flex-ext}$  será la velocidad que llevará el dedo medido.
- $T_{reductora-motor}$  es la relación de transmisión entre la reductora y el motor.
- $T_{flex-reductora}$  es la relación de transmisión entre los engranajes que se deberán poner para conseguir los 24,82 rpm en el dedo.

Como se vio anteriormente, el motor RE8 tiene una velocidad que oscila entre los 4480 y los 14200 rpm. Y va acoplado a un reductor planetario con una reducción de 36 a 1.

Para saber que engranajes se deberán de poner y con qué relación de transmisión para conseguir los 24,82 rpm de velocidad en el dedo, se harán los siguientes cálculos:

$$T_{flex-reductora} = \frac{W_{flex-ext}}{W_{reductora}} \quad T_{reductora-motor} = \frac{W_{reductora}}{W_{motor}}$$

-En el caso de tener el motor funcionando en condiciones nominales:  $W_{motor} = 4480 \text{ rpm}$

$$\tau_{\text{flex-reductora}} = \frac{W_{\text{flex-ext}}}{W_{\text{reductora}}} = \frac{W_{\text{flex-ext}}}{(T_{\text{reductora-motor}}) \times (W_{\text{motor}})} = \frac{24,82 \text{ rpm}}{\frac{1}{36} \times 4480 \text{ rpm}} = \frac{2}{10}$$

-Con el motor funcionando a máxima velocidad:  $W_{\text{motor}} = 14200 \text{ rpm}$

$$\tau_{\text{flex-reductora}} = \frac{W_{\text{flex-ext}}}{W_{\text{reductora}}} = \frac{W_{\text{flex-ext}}}{(T_{\text{reductora-motor}}) \times (W_{\text{motor}})} = \frac{24,82 \text{ rpm}}{\frac{1}{36} \times 14200 \text{ rpm}} = \frac{6}{100}$$

Con estos datos, se buscará en el mercado los engranajes que cumplan con una relación de transmisión en el intervalo  $[\frac{6}{100}, \frac{2}{10}]$ . Y que sean además compatibles con nuestro sistema Motor-Reductora.

Como es un prototipo se optó por adquirir un sistema tornillo sin fin-corona el cual tiene las siguientes características (**Figura 30**).



Figura 30. Conjunto Sin Fin – Corona. [30]

Como podemos observar en la figura 30, este sistema sin fin-corona consta de un tornillo sin fin de dos entradas que va engranado a una corona de 27 dientes.

$$\tau_{\text{flex-red}} = \frac{e1}{z2} = \frac{2}{27} = \frac{1}{13,5}$$

Esta relación de transmisión está dentro del intervalo en el que nos movíamos. En concreto para alcanzar la velocidad objetivo de 24,82 rpm, el motor tendría que girar a 12062 rpm.

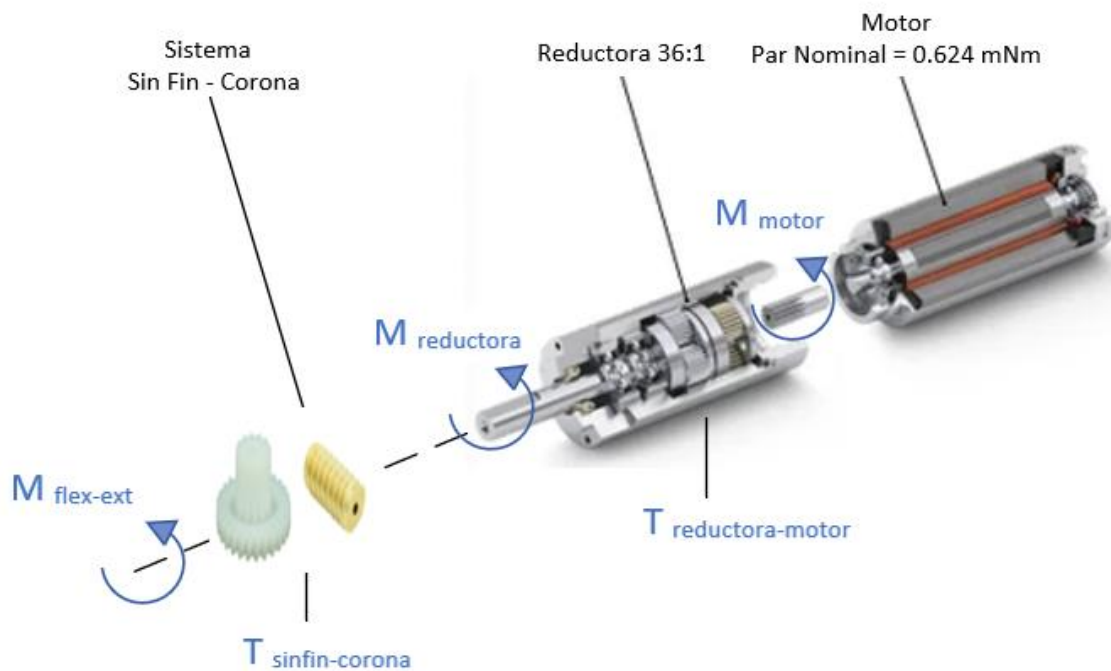
$$W_{\text{motor}} = \frac{W_{\text{flex-ext}}}{(T_{\text{reduc-motor}}) \times (T_{\text{flex-red}})} = \frac{24,82 \text{ rpm}}{\left(\frac{1}{36}\right) \times \left(\frac{1}{13,5}\right)} = 12062 \text{ rpm}$$

El motor RE8 tiene una velocidad que oscila entre los 4480 y los 14200 rpm. Por lo tanto, es capaz de llegar a la velocidad de 12062 rpm sin ningún problema. Esto nos daría una velocidad de 2,6 rad/s en la flexión del dedo, proporcionándonos un movimiento natural.



### 5.1.2 Análisis dinámico

Del mismo modo que con la velocidad calculada en el apartado anterior, se partirá de las especificaciones del motor para conocer la fuerza que podría soportar el dedo, teniendo en cuenta dos cosas: En primer lugar el sistema sinfín-corona con una relación de 2:27 y en segundo lugar el objetivo de mantener la velocidad del dedo en 24,82rpm para simular así un movimiento natural (**Figura 31**).



**Figura 31.** Esquema Motor RE8 + Reductor + Sistema tornillo sin fin – corona.

Para el análisis dinámico se usará la siguiente nomenclatura:

- $M_{motor}$  será el momento generado por el motor.
- $M_{reductor}$  será el momento generado a la salida del reductor planetario
- $M_{flex-ext}$  será el momento en el la corona del sistema tornillo sin fin.
- $T_{reductora-motor}$  es la relación de transmisión entre la reductora y el motor.
- $T_{sin\ fin-corona}$  es la relación de transmisión entre el tornillo sin y la corona. El movimiento de la corona coincide con el del dedo ya que las dos giran solidarios al mismo eje.

El par que nos dará el motor vendrá determinado por su velocidad de giro. En este caso a una velocidad de 24,82 rpm, el par será de 0,42 mNm (**Figura 32**).

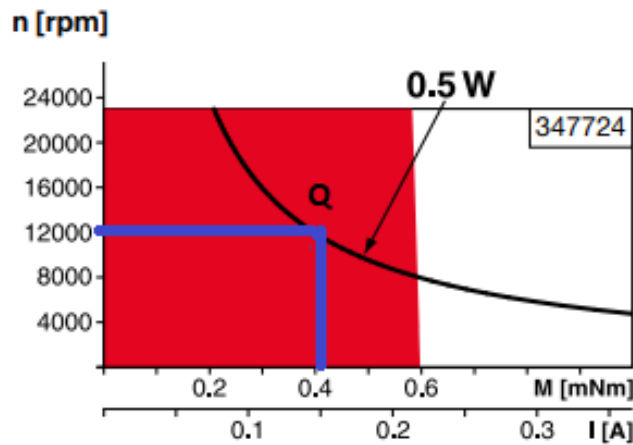


Figura 32. Rango de funcionamiento del Motor RE8. [27]

La corona del conjunto sin fin, irá acoplada directamente a un eje que será el que transmita el movimiento de flexión/extensión al dedo (Figura 33).

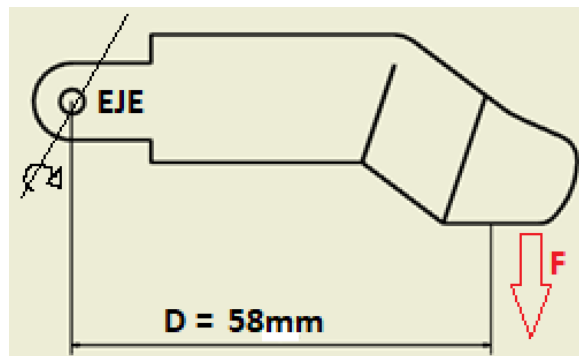


Figura 33. Distancia  $D$  (desde el punto de giro hasta el punto de contacto)

Teniendo todo esto en cuenta se realizará el cálculo para determinar cuánto será el par máximo que podrá soportar el dedo protésico:

$$\tau_{\text{sin fin-corona}} = \frac{M_{\text{reductor}}}{M_{\text{flex-ext}}} \quad \tau_{\text{reductor-motor}} = \frac{M_{\text{motor}}}{M_{\text{reductor}}}$$

$$M_{\text{flex-ext}} = \frac{M_{\text{reductor}}}{\tau_{\text{sin fin-corona}}} = \frac{M_{\text{motor}}}{\tau_{\text{reductor-motor}}} = \frac{0,42\text{Nm}}{\frac{1/36}{27}} = 204,12 \text{ mNm} = 0,2 \text{ Nm}$$

$$F = \frac{M_{\text{flex-ext}}}{D} = \frac{0,2 \text{ Nm}}{0,058\text{m}} = 3,448 \text{ N}$$

$$\text{Masa} = \frac{F}{a} = \frac{3,448\text{N}}{9,81 \text{ m/s}^2} = 0,3515 \text{ kg} = 351,5 \text{ g}$$

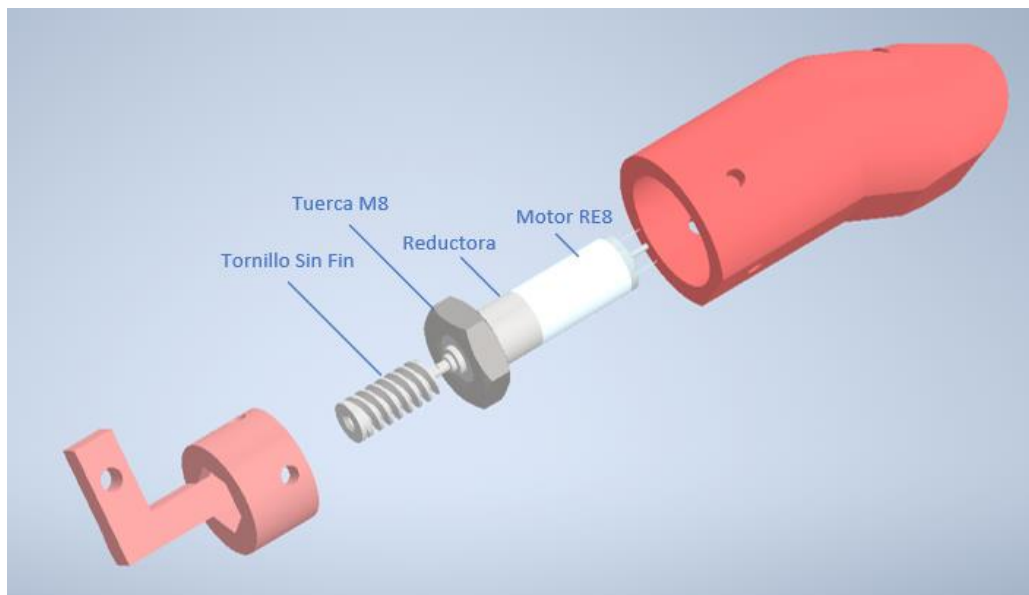
Como podemos observar, no se cumplirá uno de los objetivos fijados en el capítulo anterior de especificaciones, el cuál era conseguir que el dedo tuviera una fuerza de agarre de aproximadamente 9,28N. En nuestro caso, el dedo protésico llegará a tener una fuerza de agarre de 3,448N con una velocidad de 24,82 rpm, lo cual es un buen punto de partida para empezar con el diseño de los componentes.

### 5.1.3 Dedo: diseño y fabricación

El diseño del dedo se ha basado en el estado del arte del Capítulo 2, en las dimensiones expuestas en el Capítulo 3, en las dimensiones del conjunto motor-encoder y en otras adaptaciones basadas en la experimentación y en las pruebas realizadas. Por otro lado, la fabricación de todos los productos se llevará a cabo en una impresora Ender 3 Pro con el material PLA, el cual un coste muy reducido y nos permitirá realizar pruebas de forma rápida y conseguir un producto final con la calidad suficiente. Los parámetros de impresión se realizarán con el software Cura.

Como se vio en el estado del arte del Capítulo 2, los dedos protésicos actuales, llevan un motor de corriente continua en su interior. Es por ello que se diseñará un dedo que llevará en su interior un motor que permita el movimiento de flexión/extensión. Por lo tanto, el dedo llevará un hueco cilíndrico que permitirá el ajuste del motor en su interior.

Por otro lado, se necesitará disponer de un agujero roscado M8 de paso fino para poder fijar el conjunto motor-reductor-encoder en el interior del dedo. En este agujero irá roscado el reductor planetario el cual tiene una rosca M8 en uno de sus extremos. En nuestro caso, al ser el primer prototipo, se usará una tuerca M8 de paso fino para fijar el conjunto motor-reductor-encoder. Además, se realizará una base donde irá colocada la tuerca a presión. Esta base irá insertada en el hueco cilíndrico del dedo y estarán unidas por dos tornillos M3, formando entre ambas piezas el dedo (**Figura 34**).



*Figura 34. Dimensiones del reductor con rosca. [28]*

Tal como se puede ver en la **Figura 34**, el dedo estará compuesto por dos sólidos rígidos que además se fabricarán cada una como una pieza única. A continuación, se muestra el diseño impreso en plástico PLA.

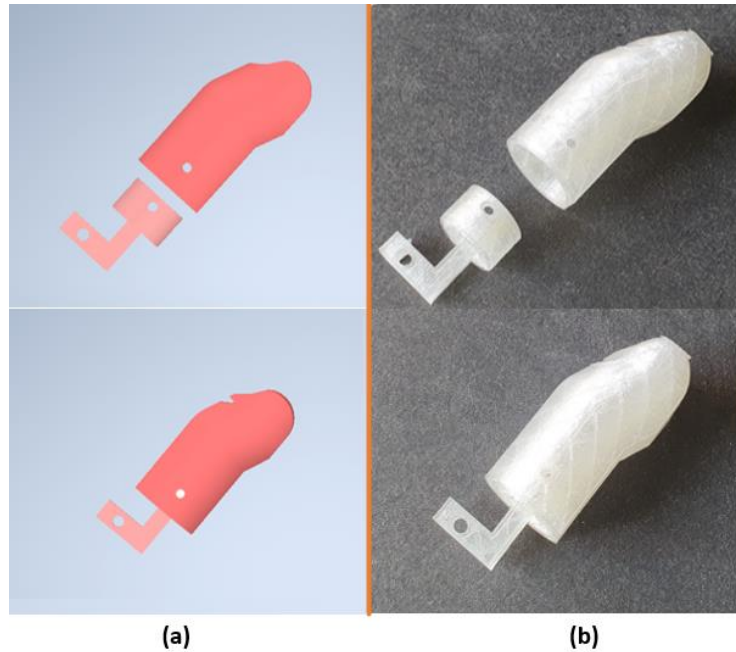


Figura 35. Dedo 3D en Inventor (a) y Dedo Impreso (b).

En este diseño se trató de asemejar las formas de un dedo, se dispuso de una zona plana que correspondería con la yema y se trataron de imitar el resto de formas y curvaturas del dedo en un estado relajado.

#### 5.1.4 Base del Dedo: diseño y fabricación

La base que se diseñará tendrá la función de unirse con el dedo a través de un pasador para poder generar el movimiento de flexión – extensión, y además tendrá un agujero en el cual irá insertado a presión la corona secundaria del sistema de engranajes.

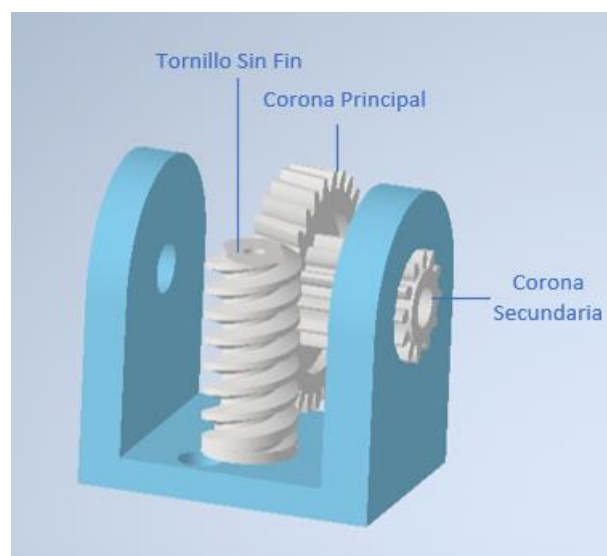


Figura 36. Base del dedo

A continuación, se muestra la base en su versión 3D y en su versión impresa.

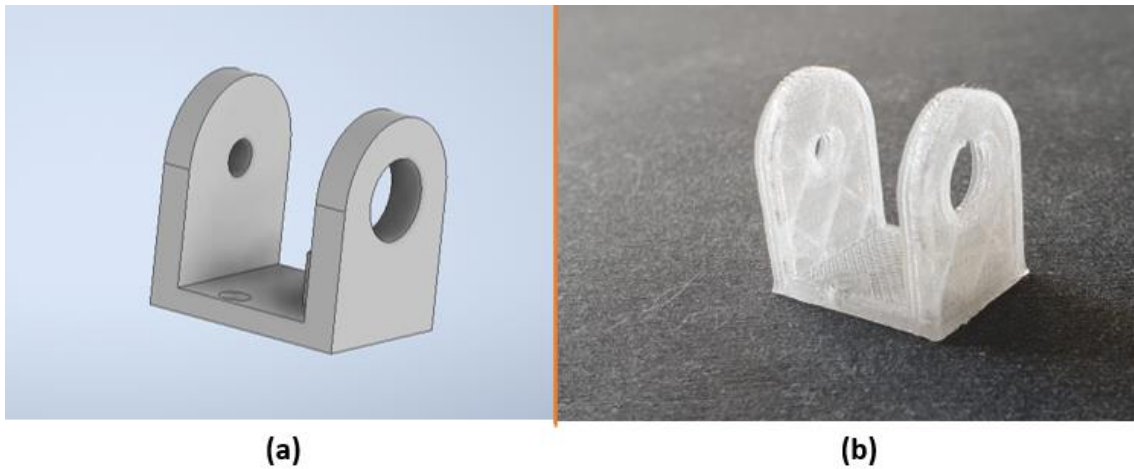


Figura 37. Base 3D en Inventor (a) y Base Impresa (b).

### 5.1.5 Pasador cilíndrico y soporte

Se usará un pasador cilíndrico metálico que atravesará la base y también el dedo, provocando así la unión solidaria de ambas piezas y permitiendo el giro de una respecto a la otra (**Figura 38**).

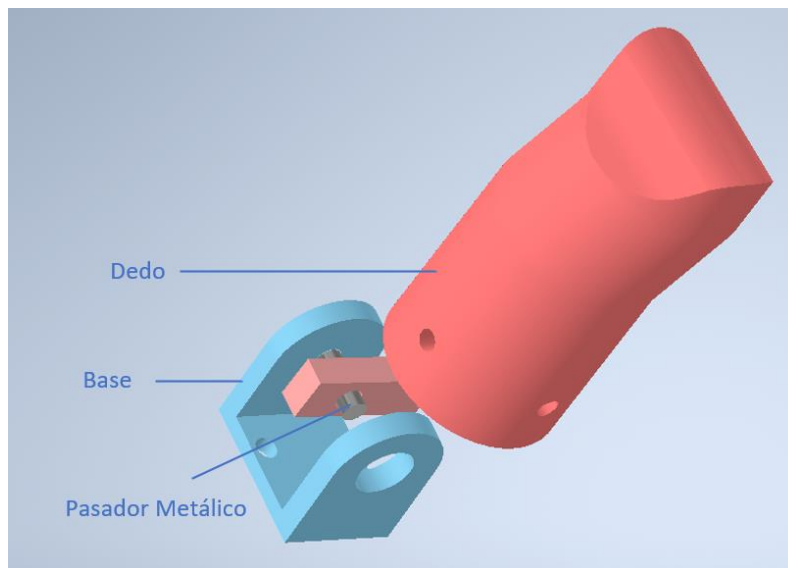


Figura 38. Pasador cilíndrico insertado entre la base y el dedo

Como vemos en la figura anterior, el pasador tiene que ser lo suficientemente robusto para mantener el peso de todo el dedo, por ello se decidió añadir un soporte en el cual el pasador metálico se inserte a presión, quedando el diseño de la **Figura 39**.

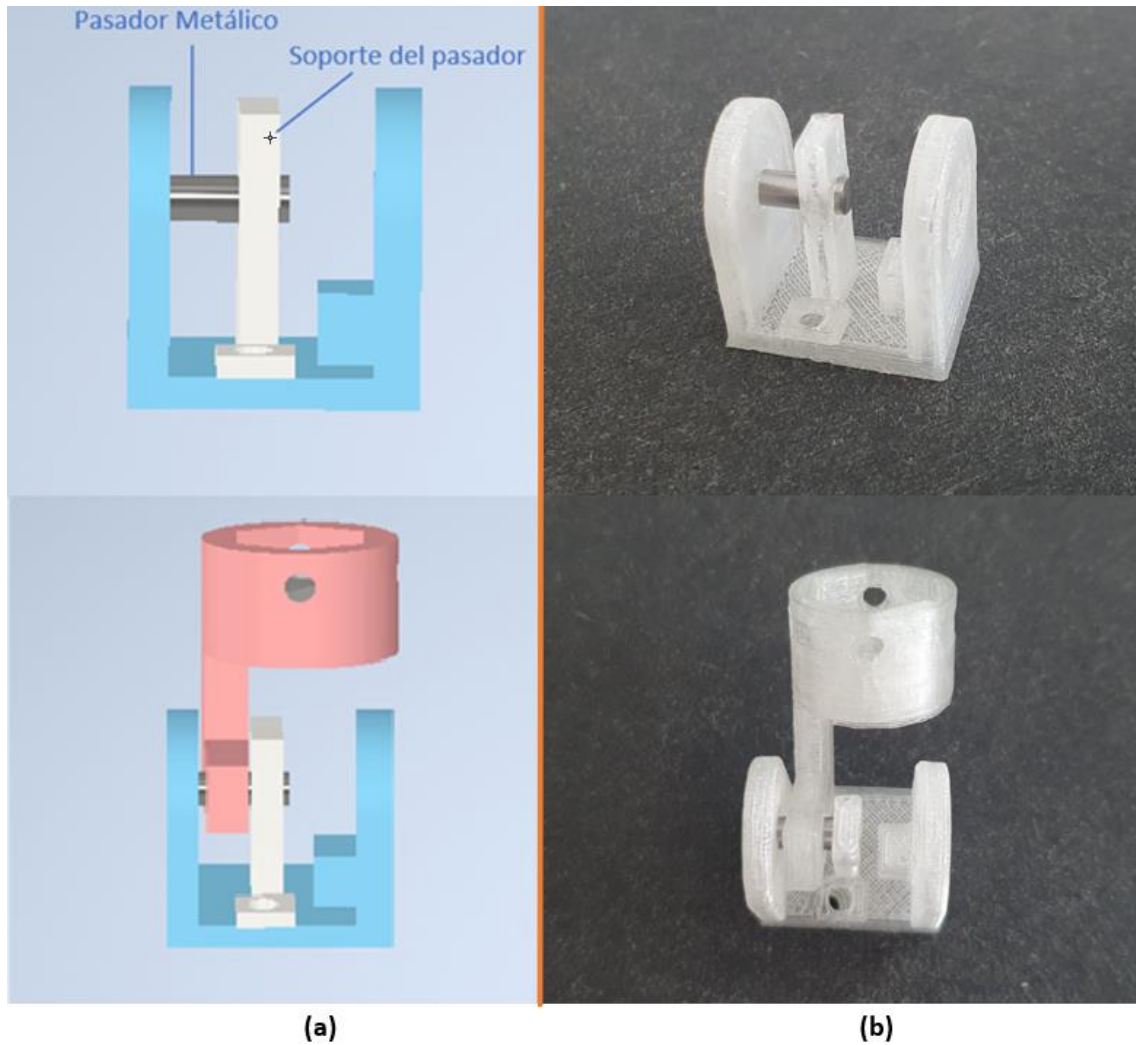


Figura 39. Diseño en Inventor del soporte del pasador (a) y soporte impreso (b).

En el Anexo A, al finalizar este documento se encuentran los planos del diseño.

### 5.1.6 Ensamblaje

Para realizar el ensamblaje del conjunto total, se partirá de las siguientes partes diseñadas e impresas en 3D.

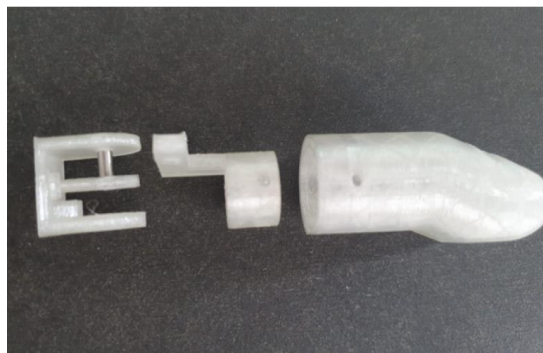
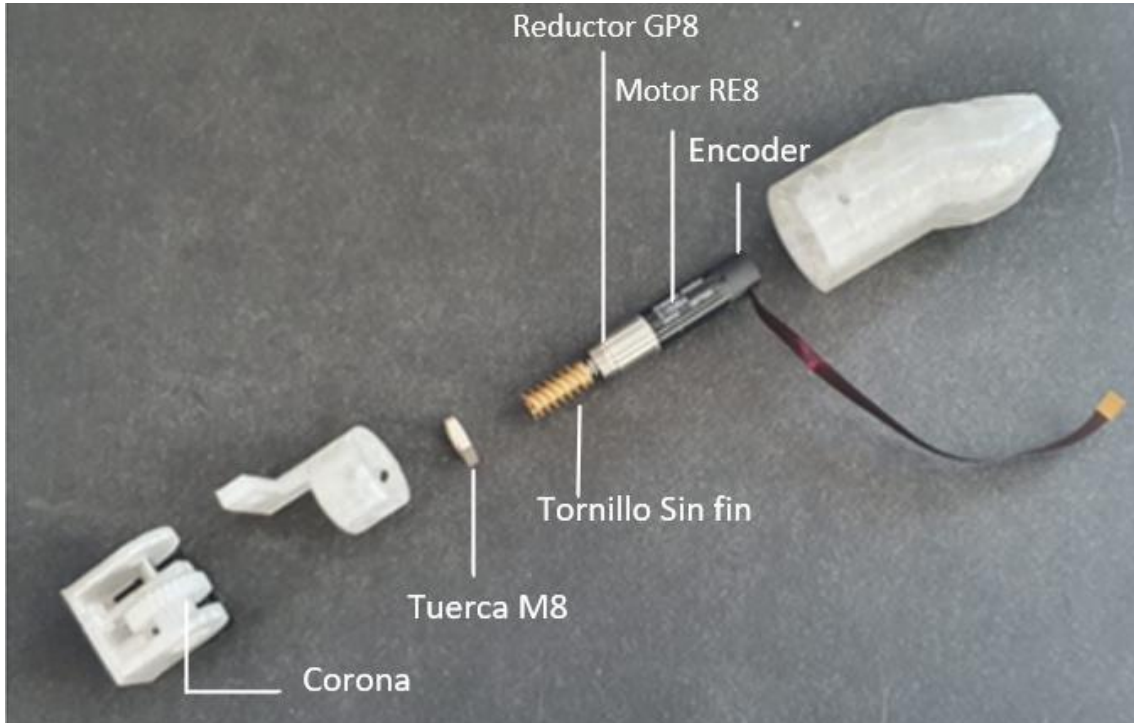


Figura 40. Partes fabricadas con impresora 3D.

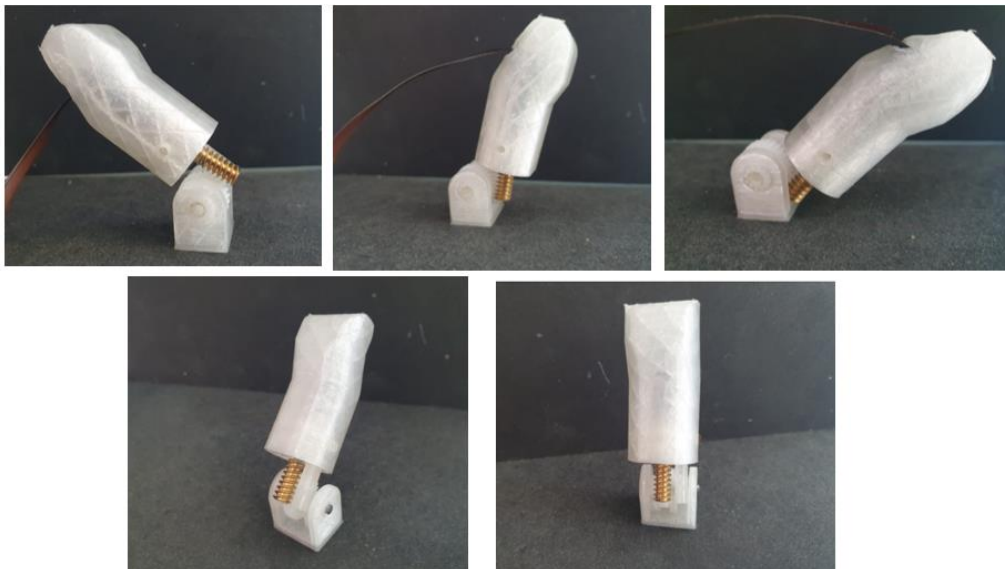


Para ensamblar el dedo pulgar protésico se han añadido a las partes fabricadas por impresora 3D los siguientes elementos: encoder, motor RE8 reductor planetario GP8, tornillo sin fin, tuerca M8 de paso fino y una corona de 23 dientes (**Figura 41**).



*Figura 41. Despiece de los elementos que conforman el dedo protésico.*

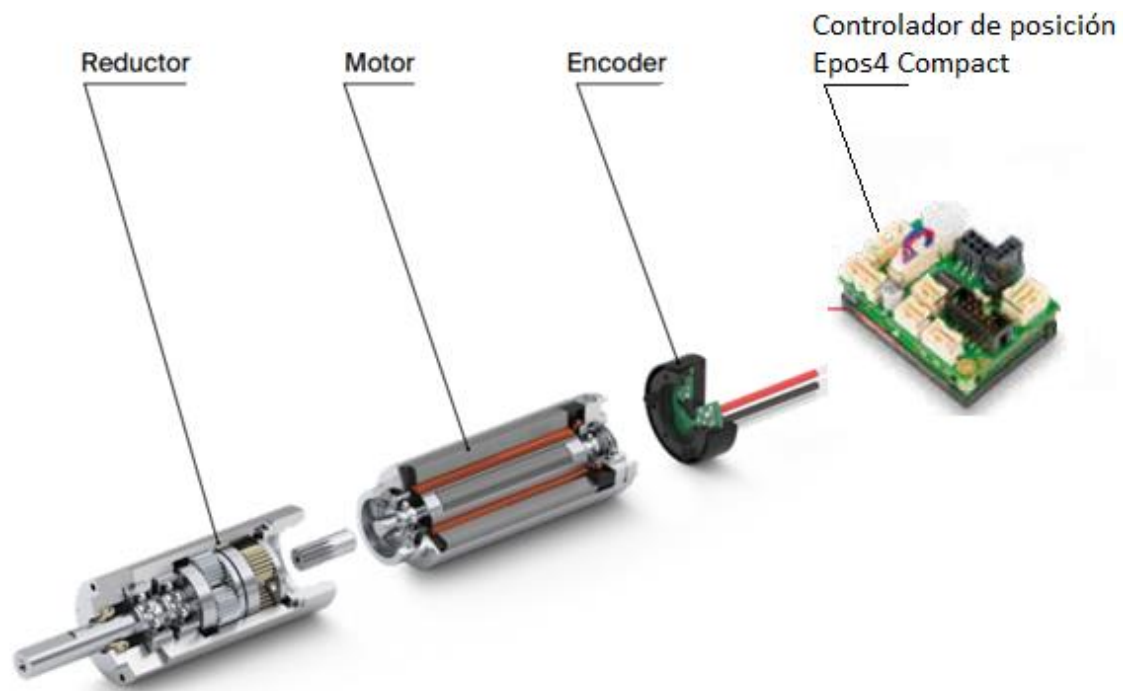
Finalmente, se ha ensamblado todo el conjunto, quedando el dedo pulgar protésico con la apariencia de **Figura 42**.



*Figura 42. Fotos desde distintos ángulos del dedo pulgar protésico.*

## 5.2 Diseño electrónico

Como se comentó anteriormente, este proyecto forma parte de uno previo, basado en el desarrollo del resto de dedos de la mano. En dicho proyecto se optó por usar un motor de corriente continua del fabricante Maxon con su propio controlador de motores EPOS4 Compact, por lo tanto, el sistema de partida fue el de la **Figura 43**.



*Figura 43. Sistema: Reductor + Motor + Encoder + Controlador de Posición Epos4. [31]*

El controlador de posición Epos4 tiene la peculiaridad de que es un controlador de tipo esclavo. Esto significa que necesita de un controlador maestro que le mande las instrucciones a ejecutar. Por lo tanto, hará falta elegir una placa que se encargue de dar las ordenes al controlador Epos4. Esta placa que hará de controlador maestro, tendrá que ser de tamaño reducido, portátil y además tener la suficiente capacidad computacional como para gobernar a la EPOS.

A continuación, se explicará las características principales de la Epos4 y su puesta en marcha usando un ordenador con Windows 10 que hará de Maestro. Después se elegirá un ordenador de tamaño reducido que haga de maestro (reemplazando al ordenador con Windows 10), teniendo en cuenta las funcionalidades de la Epos4 y las características del proyecto. Luego se implementará la comunicación entre los dos controladores (maestro-esclavo) y por último se le añadirá un sensor muscular el cual se utilizará para implementar un control mioeléctrico.



### 5.2.1 Controlador Esclavo - Epos4 Compact

El controlador de posición Epos4 Compact 24/1.5 CAN será el encargado de regular el flujo de intensidad que recibirá el motor para poder realizar los movimientos de flexión/extensión en el dedo protésico. Tiene un tamaño compacto lo cual lo hace ideal para montarlo en aplicaciones donde se dispone de un espacio muy reducido. El tamaño del controlador Epos4 se puede observar en la siguiente figura.

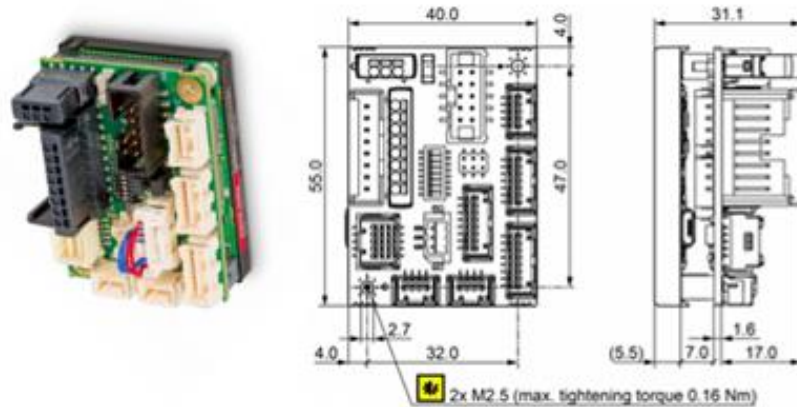


Figura 44. Dibujo dimensional de la EPOS4 Compact en mm. [29]

Las principales características se pueden ver en el catálogo de su página web, destacando las siguientes [25]:

- Datos eléctricos:

Tipos de controlador	
	<b>CANopen Slave</b>
Datos eléctricos	
Tensión de alimentación $V_{CC}$	10 - 24 VDC
Voltaje de alimentación de la lógica $V_C$ (opcional)	10 - 24 VDC
Máx. tensión de salida	$0.9 \times V_{CC}$
Máx. corriente de salida $I_{m\acute{a}x}$	4.5 A (<30 s)
Corriente en continuo de salida $I_{cont}$	1.5 A
Frecuencia de conmutación	100 kHz
Velocidad de muestreo del PI = control de corriente	25 kHz (40 $\mu$ s)
Velocidad de muestreo del PI = control de velocidad	2.5 kHz (400 $\mu$ s)
Velocidad de muestreo del PI = control de posición	2.5 kHz (400 $\mu$ s)
Máx. velocidad (1 par de polos)	50 000 rpm (sinusoidal), 100 000 rpm (en bloque)
Choque motor incorporado por fase	94 $\mu$ H / 1.5 A

Tabla 4. Datos eléctricos de la Epos 4. [32]

Como podemos ver en la tabla 4, se trata de un controlador del tipo Slave (esclavo). Se puede alimentar con una tensión entre 10 y 24 VDC y es capaz de suministrar hasta 1,5 amperios en continuo. De ahí el nombre Epos4 Compact 24/1.5.

- Entradas

Señales de sensor Hall	H1, H2, H3
Señales encoder	A, A\, B, B\, I, I\ (máx. 6.25 MHz)
Señales de sensor	A, A\, B, B\, I, I\, Clock, Clock\, Data, Data\
Entradas digitales	4 (nivel conmutable: Logic/PLC)
Entradas digitales "High-speed"	4, diferencial
Entradas analógicas	2 (resolución de 12 bits, -10...+10 V)
CAN ID / DEV ID	configurable con microinterruptores 1...5

Tabla 5. Entradas de la Epos4. [32]

En la tabla 5 se puede ver que la Epos4 dispone de dos entradas analógicas. Esto es de especial interés ya que será en estas entradas donde se conectarán los sensores musculares. Por otro lado, vemos que dispone de entradas dedicadas al encoder, las cuales se usarán para conectar el encoder MR que viene acoplado al motor RE8.

- Salidas

Salidas digitales	2
Salidas digitales "High-speed"	1, diferencial
Salidas analógicas	2 (resolución de 12 bits, -4...+4 V, máx. 1 mA)
Voltajes de salida encoder	+5 VDC, máx. 70 mA
Voltajes de salida sensores Hall	+5 VDC, máx. 30 mA
Voltajes de salida auxiliar	+5 VDC, máx. 150 mA

Tabla 6. Salidas de la Epos4. [32]

En este caso, se utilizarán dos salidas. La primera se utilizará para alimentar al encoder y la segunda de +5 VDC para alimentar al sensor muscular.

- Interfaces

RS232	RxD; TxD (máx. 115 200 bit/s)
CAN	high; low (máx. 1 Mbit/s)
USB 2.0/3.0	Data+; Data- (Full Speed)
EtherCAT	-

Tabla 7. Interfaces de la Epos4. [32]

En este caso como se controlará un único motor, es suficiente con la interfaz de comunicaciones USB. Además, USB es un estándar muy utilizado lo cual facilitará la elección del controlador maestro. Aunque en un futuro se utilizará la interfaz CAN para controlar múltiples dedos.

- Condiciones ambientales y datos mecánicos

Condiciones ambientales	
Temperatura de funcionamiento	-30...+45°C
Temperatura, rango ampliado	+45...+70°C; Derating: -0.060 A/°C
Temperatura de almacenamiento	-40...+85°C
Humedad (sin condensación)	5...90%
Datos mecánicos	
Peso	aprox. 58 g
Dimensiones (L x A x H)	55.0 x 40.0 x 31.1 mm
Montaje	Tornillos M2.5

Tabla 8. Condiciones ambientales y datos mecánicos. [32]

Como se puede observar en la tabla 8, el controlador Epos4 tiene un peso de 58 gramos. Esto lo hace ideal para proyectos en los que el peso es un factor importante, como es el de este TFG en el que desarrollará una prótesis que tendrá que ser llevada por una persona. Además, la electrónica de control de posición tiene circuitos contra sobre corriente, temperatura excesiva, tensión insuficiente o excesiva, contra transitorios de tensión y cortocircuitos del cable del motor, así como para pérdida de señal de feedback.

Teniendo todo esto en cuenta, la conexión del conjunto Motor-Reductor-Encoder con el controlador Epos4 quedaría de la siguiente manera:

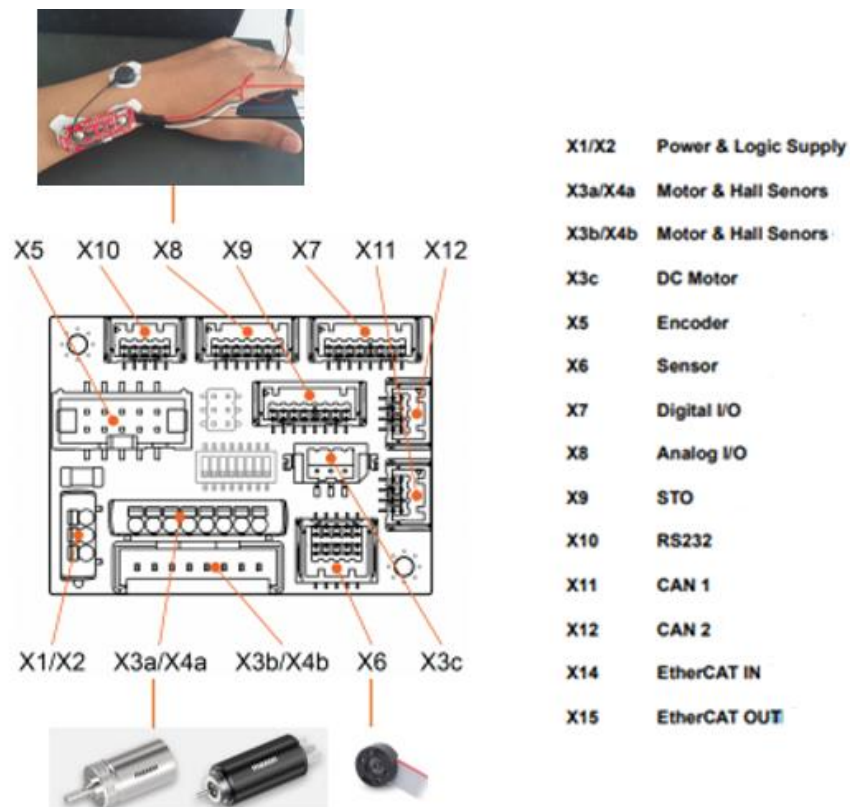


Figura 45. Esquema de conexiones Motor-Reductor-Encoder-Sensor Emg. [31]

### 5.2.2 Puesta en marcha con EPOS Studio

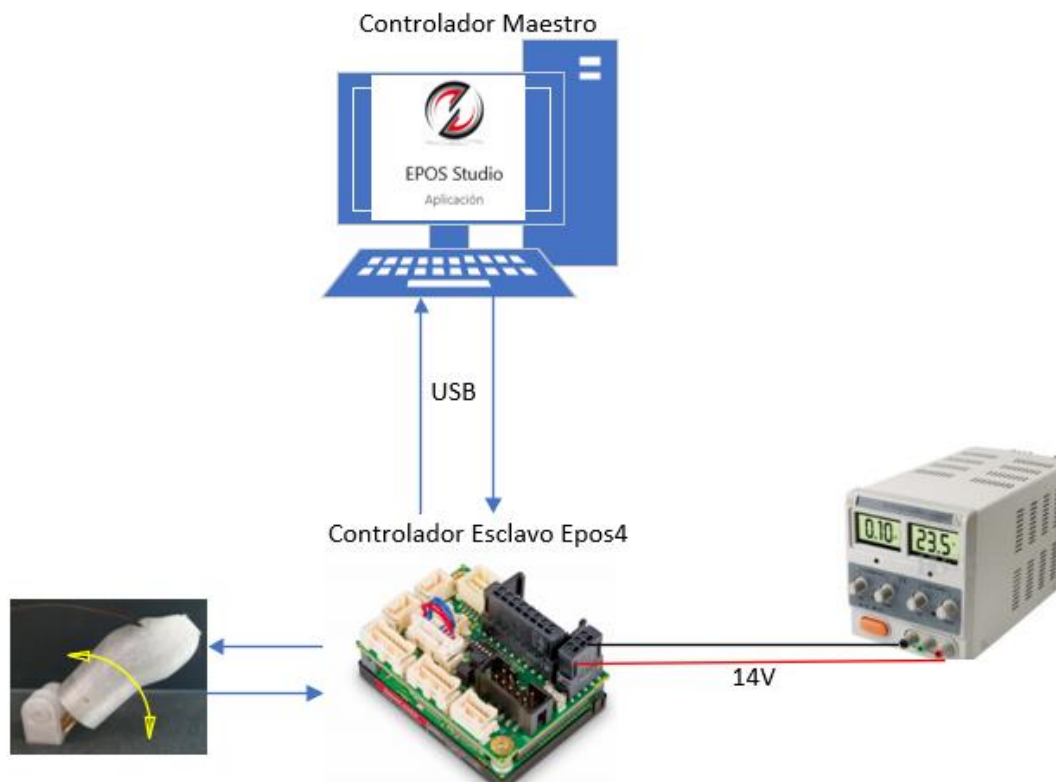
Los controladores de posición EPOS4 necesitan ser configurados antes de ser utilizados en cualquier aplicación.

La puesta en marcha y la configuración del Epos4 se realizan a través de la interfaz gráfica EPOS Studio, mediante programas de ayuda (asistentes) guiados por menú y de fácil manejo. También forma parte del sistema, un sofisticado procedimiento automático para el auto ajuste del regulador. Para la puesta en marcha se utilizará un ordenador con Windows 10 que hará de maestro y en el cual se instalará el software EposStudio.

- **Configuración de la Epos4 a través de EPOS Studio**

Para configurar la Epos4, lo primero que se hizo fue instalar el software EposStudio, este programa está disponible de forma gratuita en la página web de Maxon [30].

Después de instalar el software EPOS Studio en la PC, se alimentó a la Epos4 con 14V y se le conectó el conjunto motor-reductor-encoder, los cuales están dentro del dedo pulgar protésico. Finalmente se conectó la PC con la Epos4 a través de un cable USB, quedando así el esquema de **Figura 46**.



*Figura 46. Esquema puesto en marcha con EposStudio.*

Al abrir el software EPOS Studio, el programa reconoce automáticamente que está conectado una controladora EPOS4 por USB y crea un proyecto con ese nombre.

La configuración inicial se realiza de la siguiente manera:

- 1) Parámetros básicos:** Se abre la pestaña Startup (puesta en marcha) y se configura las características del conjunto Motor-reductor-encoder que se va a utilizar, en el caso de este proyecto, disponemos de un motor RE8, con un reductor planetario GP8 y un encoder MR100. Las características de cada uno de ellos ya se vieron en el apartado de diseño mecánico.

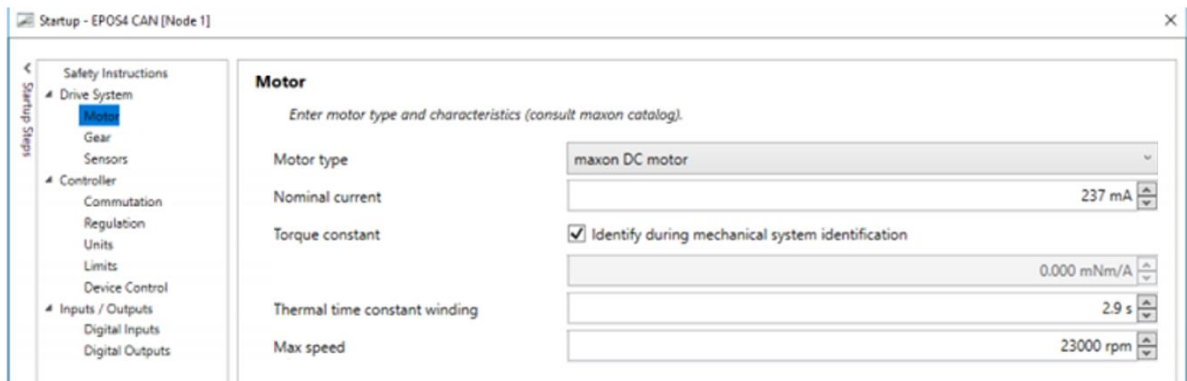


Figura 47. Parámetros del motor.

### Gear

Enter gear data.

System with gear

Absolute reduction: 36 : 1

Direction of rotation, drive to output:  Same,  Inverted

Max continuous input speed: 12000 rpm

Figura 48. Parámetros del Reductor.

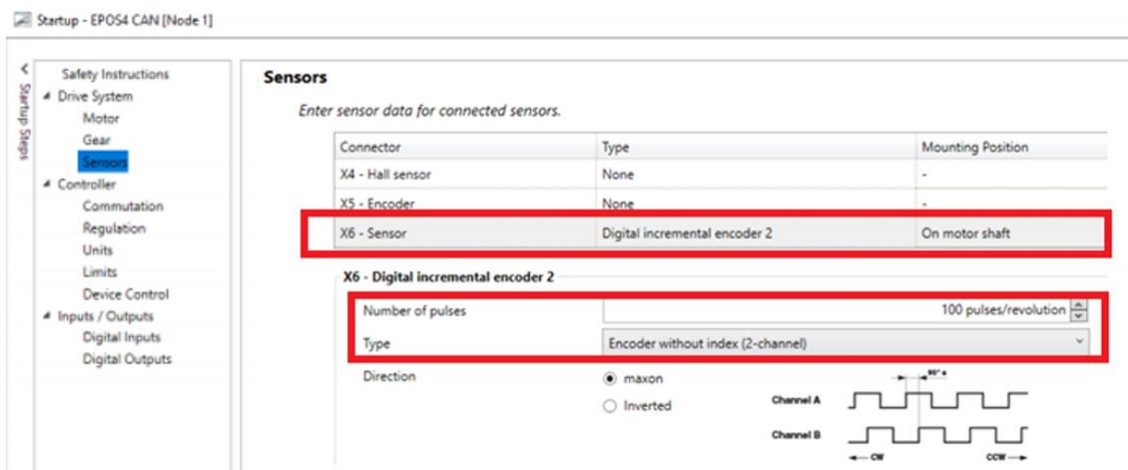


Figura 49. Parámetros del Encoder.

**2) Límites de Operación:** Se especifica los límites del sistema. Este apartado es uno de los más importantes ya que nos garantizará el buen funcionamiento y mantenimiento de la electrónica, así como de los motores.

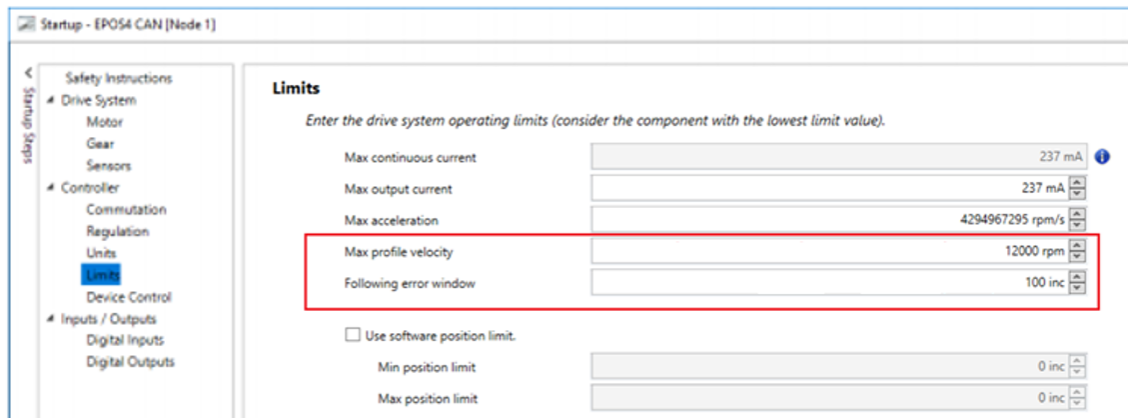


Figura 50. Límites de Operación

Es importante observar que, aunque el motor sea capaz de girar a 14400 rpm, el reductor planetario solo admite hasta los 12000 rpm como entrada. Al poner estos 12000 rpm como límite nos aseguramos que el reductor no sufra daños.

La configuración debe realizarse en el controlador Epos4 una sola vez, esta información se almacena de manera permanente hasta la siguiente reconfiguración. En este caso no la volveremos a tocar.

**3) Ajuste de parámetros de regulación:** El controlador Epos4 tiene varios modos de operación, cada uno de ellos con un regulador específico. Algunos de los modos más afines a este proyecto son los siguientes:

- Cyclic Synchronuous Position (CSP): En este modo de operación, el controlador maestro calcula el recorrido y transmite a la EPOS4 la posición de destino cíclicamente y de forma sincronizada a través la red. En este caso estaría operando el regulador de posición en la Epos4. La EPOS4 suministra al Master los valores actuales de posición, velocidad y corriente medidos por los sensores

- Cyclic Synchronuous Velocity (CSV): En este modo, el controlador maestro calcula el recorrido y transmite a la EPOS4 la velocidad de consigna cíclicamente y de forma sincronizada a través de la red. En este caso estaría el regulador de velocidad en la Epos4. La EPOS4 suministra al Master los valores actuales de posición, velocidad y corriente medidos por los sensores.

-Cyclic Synchronuous Torque (CST): En este modo de operación, el controlador maestro calcula el recorrido y transmite a la EPOS4 el par de consigna cíclicamente y de forma sincronizada a través de la red. En este caso estaría operando el regulador de corriente



en la Epos4. La EPOS4 suministra al Master los valores actuales de posición, velocidad y corriente medidos por los sensores.

-Profile Position Mode (PPM) / Punto a Punto: Este modo de operación permite el posicionamiento del eje del motor del punto A al punto B. Este posicionamiento se puede realizar respecto al punto cero del eje (absoluto) o a la posición actual del eje (relativo). Este será el modo de operación que se utilizará para situar el dedo en distintas posiciones y con una velocidad determinada. Habrá una posición para el dedo en flexión (punto A) y otra posición correspondiente al dedo en extensión (Punto B). Para ir de una posición a otra se puede hacer con distintos perfiles de velocidad y de aceleración. Estos serán los parámetros que se deberán de ajustar en el modo PPM el cuál utiliza un perfil de velocidad trapezoidal (**Figura 51**).

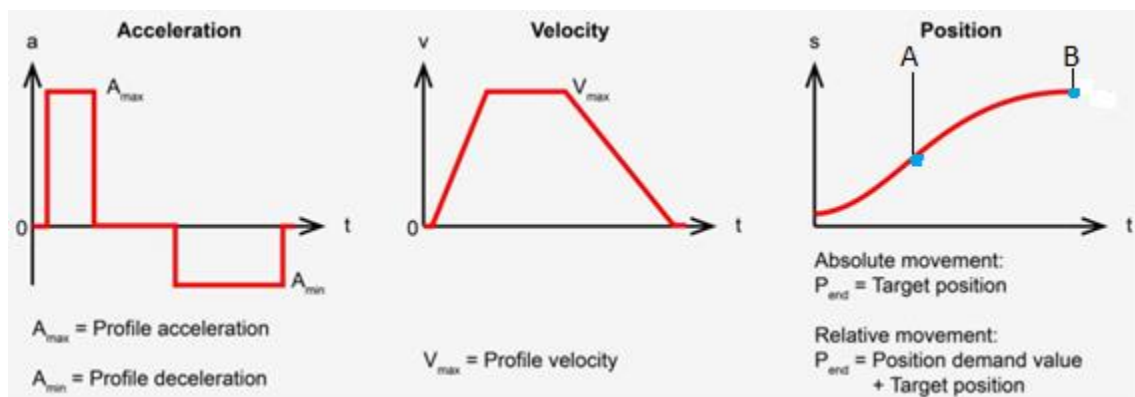


Figura 51. Modo de operación PPM. [31]

Para poder utilizar el modo de posicionamiento PPM, antes se tiene que regular los lazos de control de intensidad, velocidad y de posición.

Para realizar este ajuste, es necesario tener acoplado los motores en la estructura del dedo protésico, para que de esta manera las constantes sean calculadas con respecto a la fricción real a la que van a estar sujetas.

La herramienta que ofrece Epos Studio para realizar el ajuste, se llama "Regulation Tunning". En el podemos se puede regular tanto la intensidad, como la velocidad y la posición. Es importante hacer un ajuste previo del lazo de intensidad, ya que está subordinada al lazo de control de la velocidad y al lazo de control de la posición (**Figura 52**).

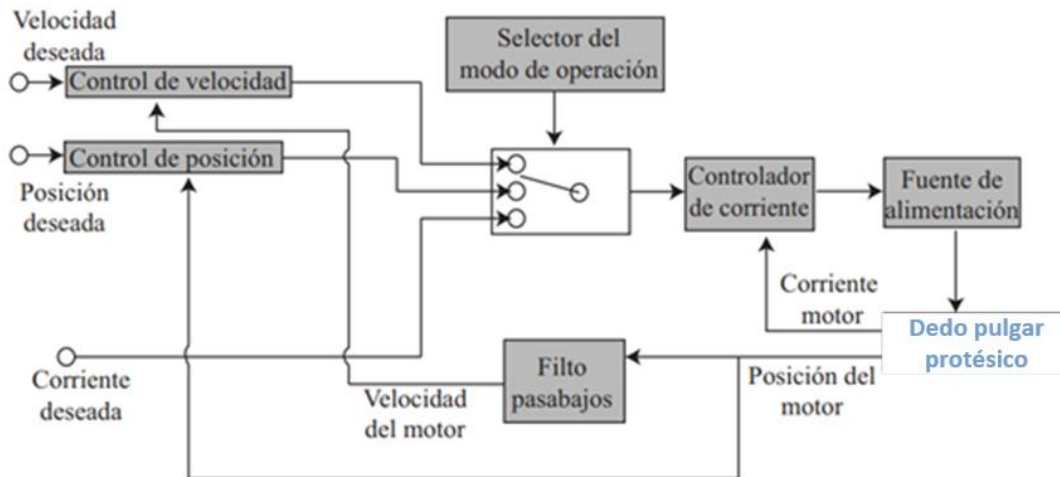


Figura 52. Descripción General de la arquitectura de control.

Como podemos observar en la **figura 52**, la estructura de la arquitectura de control esta principalmente enfocada en un controlador de corriente que esta subordinada a un control de posición y un control de velocidad.

A continuación, se realizará el ajuste de cada uno de los reguladores.

### -Ajuste del regulador de intensidad

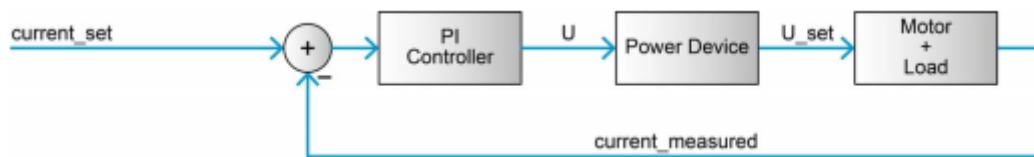


Figura 53. Arquitectura de control del regulador de corriente.

Se usará una amplitud de 115mA, para que el ajuste automático identifique el modelo del sistema.

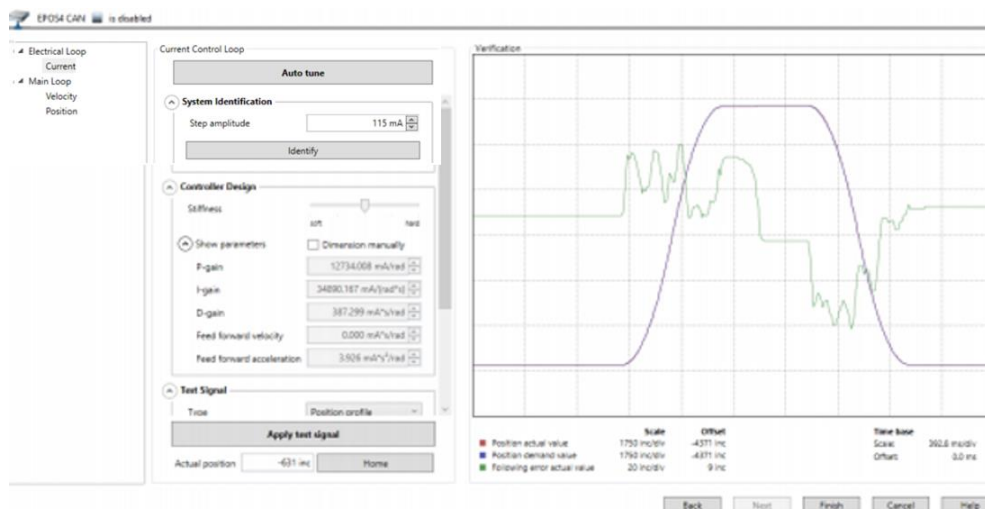


Figura 54. Ajuste del regulador de intensidad



### -Ajuste del regulador de velocidad

A la hora de sintonizar los reguladores de velocidad y posición, hay que tener en cuenta la holgura de los engranajes que componen nuestro sistema. La holgura o juego lateral, es el espacio que queda libre al engranar una pareja de dientes. Esta holgura siempre será necesaria para permitir una cierta deformación o deflexión que se produce en los dientes, además de para permitir el paso del lubricante o para la expansión térmica que sufre el metal del engranaje al calentarse.

En este caso, el reductor planetario tiene una holgura media de 2, 4º en vacío (**Figura 55**).

DATOS DEL REDUCTOR	
Relación de reducción	36 : 1
Relación de reducción absoluta	36/1
Máx. diámetro del eje del motor	0.65 mm
Número de etapas	2
Máx. par en continuo	0.008 Nm
Par de fuerza admisible brevemente	0.012 Nm
Sentido de giro, de motor a eje de salida	=
Máx. rendimiento	76 %
Holgura media del reductor sin carga	2.4 °
Momento de inercia	0.004 gcm <sup>2</sup>

*Figura 55. Holgura media del reductor planetario GP8. [28]*

Para pasar de grados a incrementos hay que tener en cuenta la resolución del encoder y se hace de la siguiente manera:

$$4 \times \frac{\text{pulses}}{\text{revolutions}} = \frac{\text{increments[inc]}}{\text{revolutions[rev]}} = \frac{\text{quadcounts[qc]}}{\text{revolutions[rev]}}$$

Teniendo en cuenta que el encoder tiene una resolución de 100 pulsos por vuelta:

$$\text{Holgura GP8} = 2.4^\circ \times \frac{1 \text{ Vuelta Output}}{360^\circ} \times \frac{36 \text{ vueltas motor}}{1 \text{ vuelta Output}} \times \frac{4 \times 100 \text{ pulsos}}{1 \text{ vuelta motor}} = 96 \text{ inc}$$

El fabricante recomienda colocar incrementos de más del doble que el de la holgura del sistema. En nuestro caso colocaremos 4 veces la holgura del reductor. De esta manera se estará teniendo en cuenta también la holgura del mecanismo tornillo sin fin – corona.

$$\text{Holgura total del sistema} = 4 \times 96 = 384 \text{ inc}$$

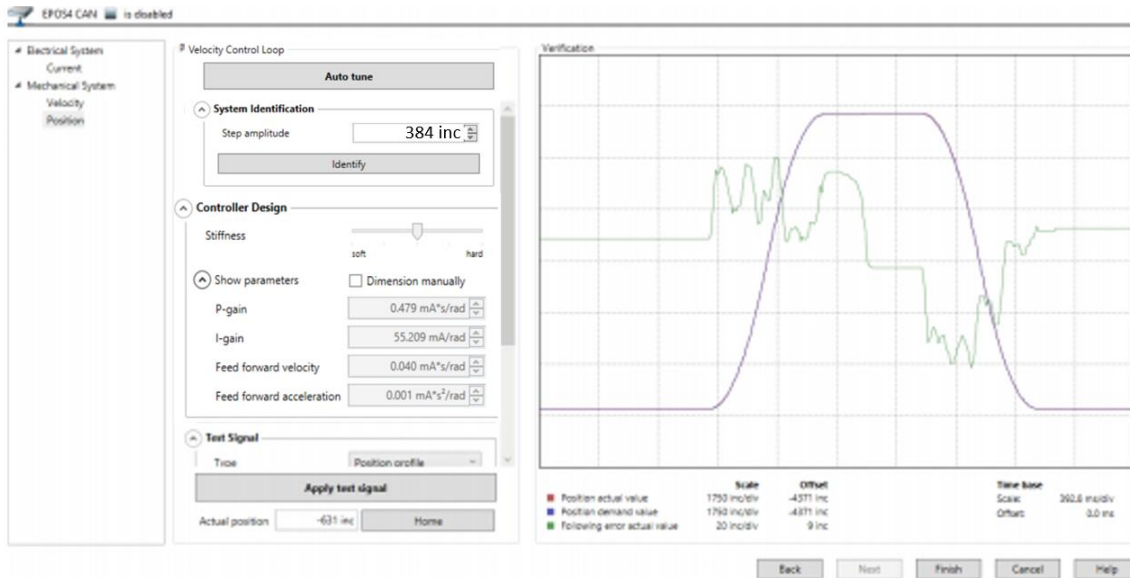


Figura 56. Ajuste del regulador de Velocidad

### -Ajuste del regulador de posición

La arquitectura de este controlador es el de la **Figura 57**. Este regulador será el encargado de posicionar el dedo protésico en un punto u otro, con una velocidad y una aceleración configurada previamente. Esto nos permitirá cumplir los objetivos fijados en las especificaciones de diseño.

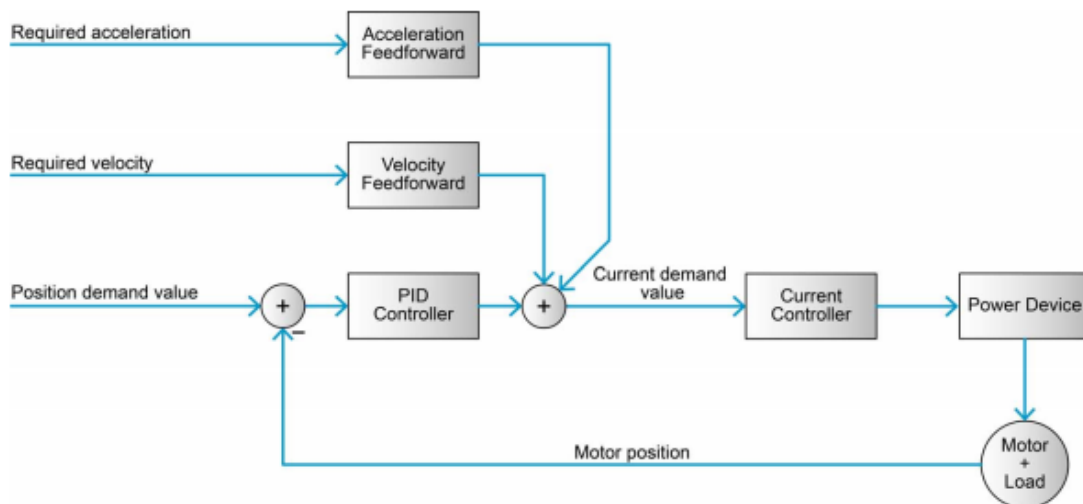


Figura 57. Arquitectura de control del regulador de posición con feedforward

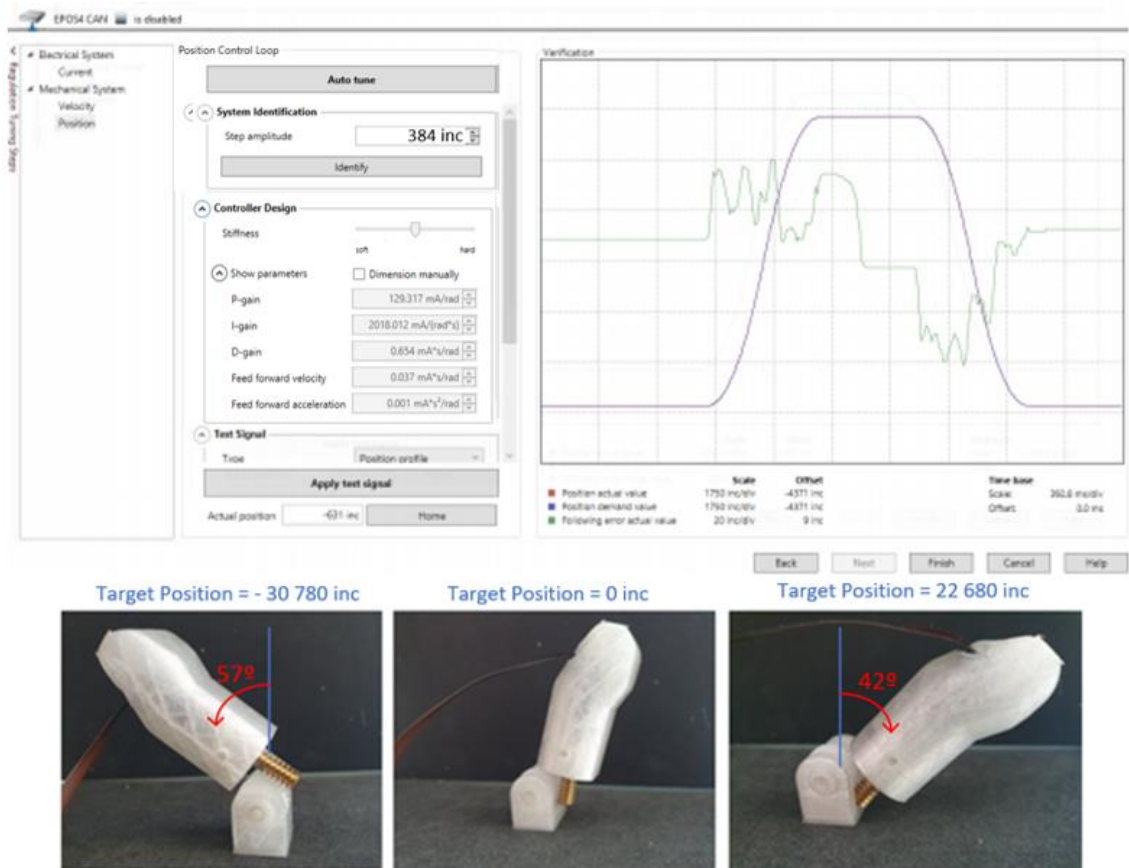


Figura 58. Ajuste del regulador de Posición y pruebas realizadas con EposStudio.

Como se puede observar en la **Figura 58**, se han obtenido buenos resultados de ajuste utilizando la herramienta "Auto tune", por lo que no fue necesario modificar las constantes del PID de manera manual.

Una vez configurada y sintonizado los lazos de la Epos4 con ayuda del software Epos Studio, se procederá a elegir un controlador maestro.

### 5.2.3 Controlador Maestro - Raspberry Pi Zero

La elección de la Raspberry Pi Zero como controlador maestro, se debe a los siguientes motivos:

- El controlador Epos4 únicamente es compatible con los procesadores que lleven una de las arquitecturas de la tabla 9:

Interface	Architecture				
	Intel		ARM		
	32-bit X86	64-bit X86_64	32-bit V6	V7/V8	64-bit V8
RS232	X	X	—	X	—
USB	X	X	X	X	X
CAN	IXXAT	X	X	—	X
	Kvaser	X	X	—	X
	PiCAN2	—	—	—	X
	MTTCAN	—	—	—	—

Tabla 9. Plataformas, arquitecturas e interfaces compatibles con la EPOS4. [30]

- El controlador Epos4 ofrece una librería con funciones compatibles con placas que tengan el sistema operativo Linux
- Al tratarse de un dedo protésico se requiere una placa que sea compacta, potente y de un precio asequible.

La placa que reúne esas características es la Raspberry Pi Zero, la cual tiene una arquitectura ARMV6 de 32 bits. La familia de instrucciones de procesamiento que forman esta arquitectura esta soportada por distintos sistemas operativos, entre ellos Linux con su distribución Raspbian. Este será el sistema operativo que se instalará en la placa Raspberry Pi Zero [31], la cual hará de controlador maestro.



Figura 59. Raspberry Pi Zero. [34]

Entre sus principales características destacan [32]:

- **Procesador:** Posee un integrado Broadcom BCM2835. Este tiene un procesador ARM1176JZFS que trabaja a 700 MHz, pero puede llegar a una frecuencia de trabajo de 1 GHz. La arquitectura del procesador que usa es ARM, en este caso ARM11.
- **Memoria:** En cuanto a la memoria tiene 512 MB de memoria RAM. Bastante aceptable tomando en cuenta que la Raspberry Pi modelo 3 B que era la más potente en ese momento tenía 1 GB de memoria RAM.
- **Puerto USB:** Debido a su pequeño tamaño, la Raspberry Pi Zero W tiene un puerto micro-USB. Por ese motivo, debemos usar un adaptador para conectar los periféricos como teclado, ratón, etc. Los kits normalmente incluyen este adaptador.
- **GPIO:** La Raspberry Pi Zero W cuenta con 40 pines GPIO. Entre esos se incluyen pines de voltaje y de tierra o GND. Entre estos pines GPIO tenemos:

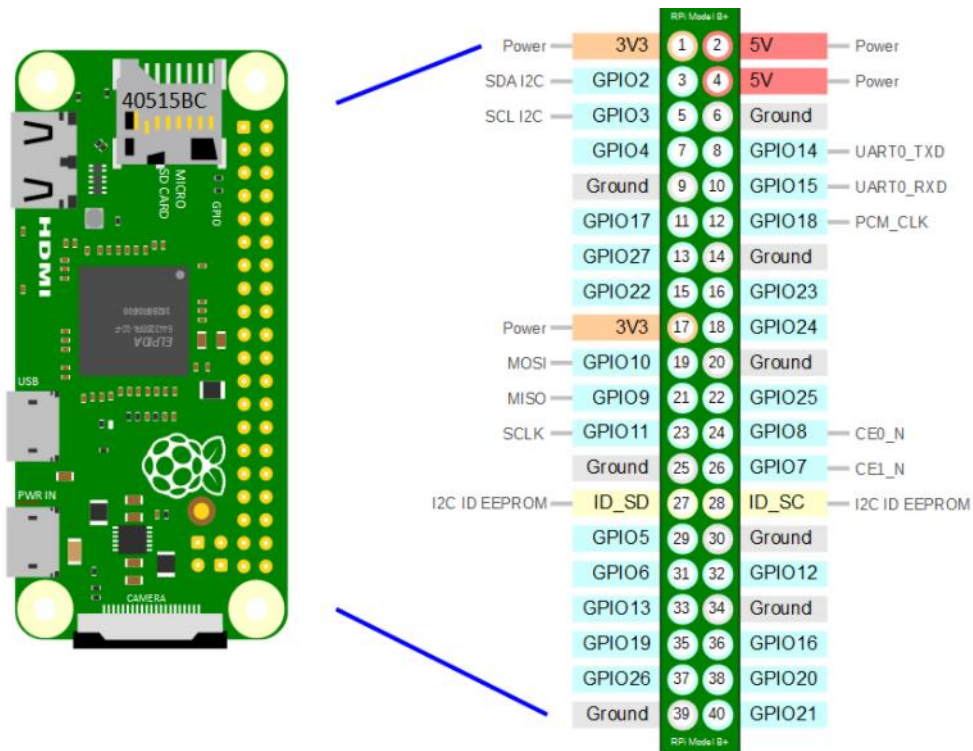


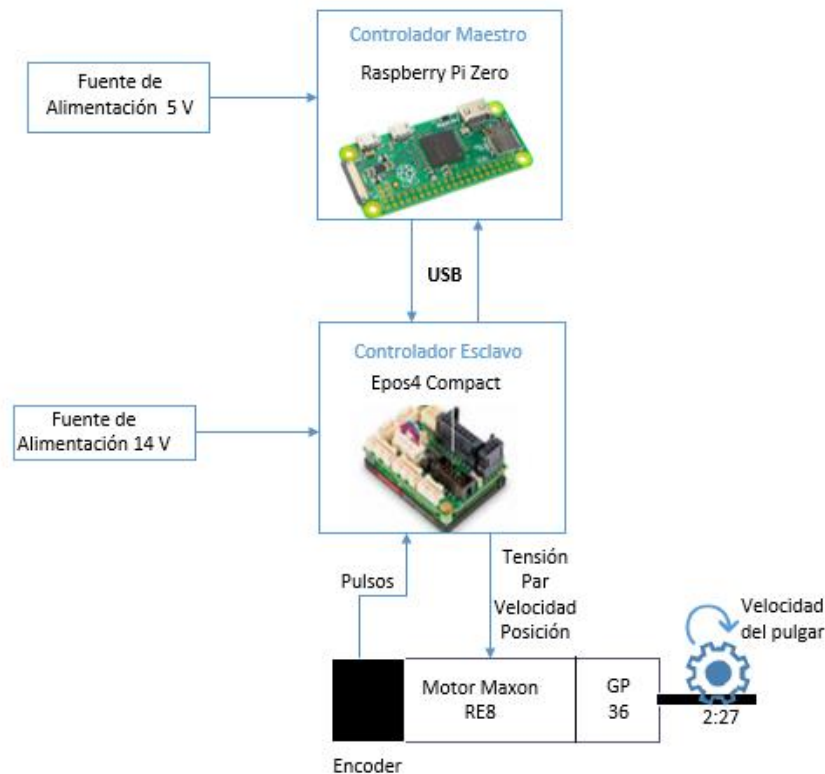
Figura 60. Pines de la Raspberry Pi Zero. [35]

Es importante tener en cuenta que los pines trabajan con una tensión de 3,3V y que la corriente máxima con la que se puede trabajar en esta placa es de 16mA.

- **Conectividad:** la Raspberry Pi Zero W tiene conectividad a redes wifi, en este caso es compatible con redes del tipo 802.11 b/g/n. Esto nos permite conectarla a un red wifi para enviar información a Internet o conectarnos a ella dentro de nuestra red en el hogar.

También posee conectividad Bluetooth 4.1 y BLE, lo cual nos permite conectarle otros dispositivos periféricos o conectarnos a ella usando nuestro teléfono inteligente.

En conclusión, por su tamaño y características es ideal para todo tipo de proyectos que requieran de computadores de tamaño micro. En este caso, la Raspberry Pi Zero hará de controlador maestro de la Epos4. El esquema de funcionamiento nos quedaría de la siguiente forma (**Figura 61**).



*Figura 61. Esquema de funcionamiento.*

Llegados a este punto, para cerrar el lazo de control e implementar un sistema de control mioeléctrico, necesitamos un sensor muscular que dependiendo del valor de la señal electromiográfica detectada en el músculo, el motor se mueva a una posición determinada, posicionando así el dedo pulgar en flexión o en extensión.

## 5.1 Sistema de control mioeléctrico

Uno de los objetivos planteados en este proyecto, es el de dotar a la prótesis de la mayor funcionalidad posible en cuanto a su forma de control. Es por ello, que para la realización de este proyecto se empleará un sensor muscular conectado a un músculo sano del paciente. Este sensor recogerá las señales electromiográficas emitidas a voluntad por el paciente y las enviará a un controlador, el cual se encargará de mover el motor a una posición determinada y producir así el movimiento de flexión/extensión en el dedo pulgar protésico (**Figura 62**).

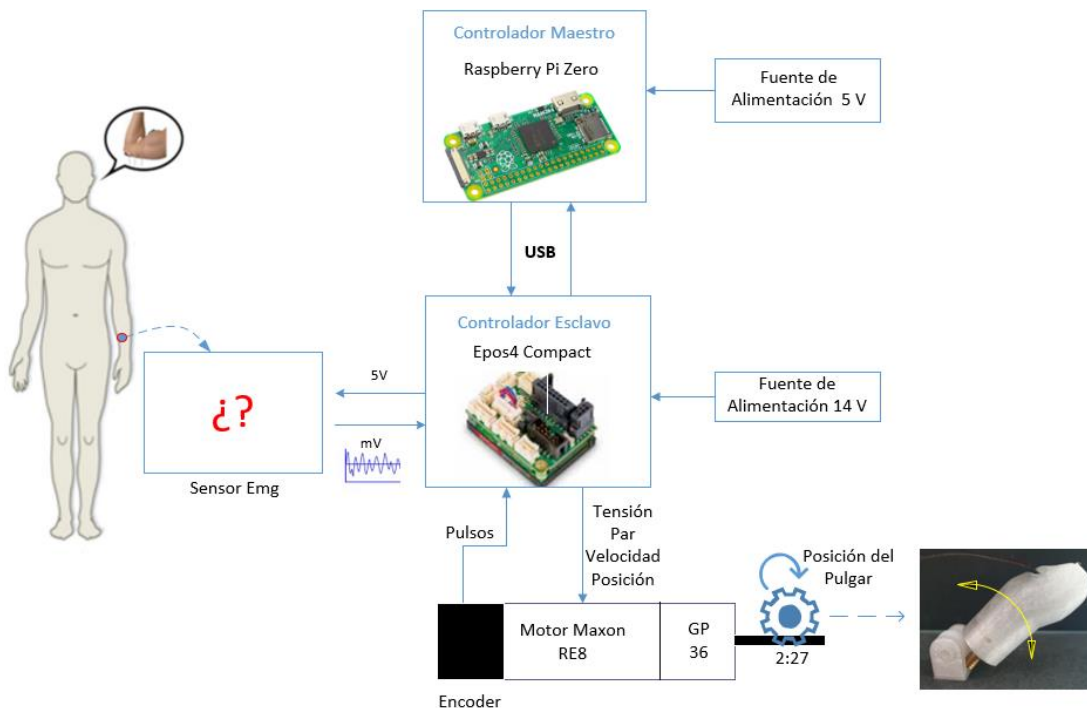


Figura 62. Esquema general de funcionamiento

A continuación, se elegirá el sensor muscular más apropiado para este proyecto teniendo en cuenta un análisis previo de las señales electromiográficas y posteriormente se implementará un algoritmo de control mioeléctrico.

### 5.1.1 Sensor muscular de electromiografía

Antes de elegir el sensor muscular, conviene conocer un poco sobre la electromiografía.

Etimológicamente, la palabra electromiografía está formado por las raíces griegas elektron (electricidad), myos (músculo) y el sufijo -grafía (escritura), por lo que se puede definir como el registro grafico de la actividad eléctrica producida por los músculos. Esta técnica engloba el proceso de detección, análisis y tratamiento de la señal eléctrica generada por la contracción de los músculos.



Estas señales son ricas en información sobre la funcionalidad del músculo y pueden explotarse para el control de prótesis mioeléctricas. Pero antes de examinar esta señal eléctrica, es relevante comprender cómo se generan en el cuerpo humano.

### Potencial de Acción

En los seres vivos la recepción de información del entorno, así como el control de los efectores requiere mecanismos que permitan transmitir información codificada desde y hacia los centros nerviosos en forma rápida y eficiente. Los animales superiores cuentan con un sistema nervioso desarrollado capaz de satisfacer esta demanda y que utiliza como mecanismo fundamental para este fin el potencial de acción, un cambio estereotipado del potencial de la membrana celular, que ocurre en el rango de los milisegundos y que se propaga a lo largo de los nervios [33].

La generación de señales eléctricas en el músculo se relaciona con el concepto de potencial de acción. Los potenciales de acción son las señales eléctricas que originan la contracción muscular, estas se generan debido a la capacidad de las membranas de las células nerviosas (neuronas) para permitir el paso de iones  $\text{Na}^+$  e iones  $\text{K}^+$ . Estas señales se propagan desde el cerebro o la médula espinal a lo largo de los axones de las células nerviosas hasta las fibras del músculo esquelético y hacen que se contraigan.

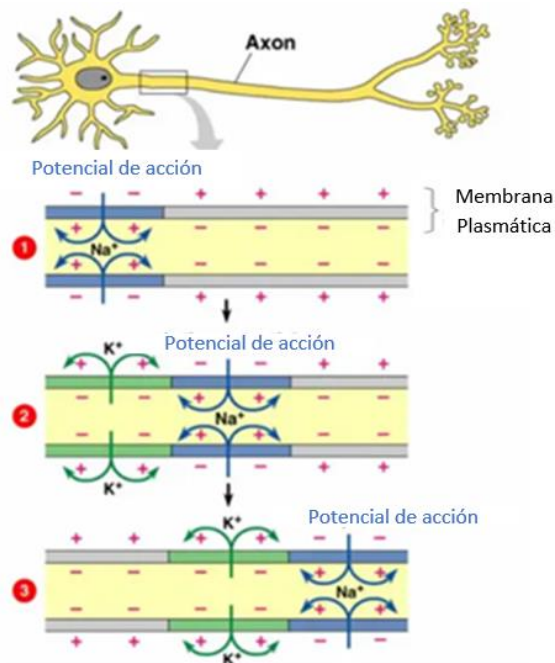


Figura 63. Desplazamiento del potencial de acción a través del axon. [34]

Como podemos observar en **Figura 63**, las membranas plasmáticas en ausencia de un potencial de acción, están polarizadas, lo que significa que hay una diferencia de voltaje en cada membrana plasmática. La diferencia de voltaje de una célula no estimulada se denomina potencial de membrana en reposo, que suele ser de unos  $-70$  mV. Sin embargo, la estimulación de una célula nerviosa puede provocar la despolarización de



su membrana plasmática. Si la despolarización hace que el potencial de membrana alcance un valor llamado 'umbral', se activa un potencial de acción.

Un potencial de acción, por lo general, tarda entre 1 milisegundo y algunos milisegundos en ocurrir y está constituido por dos fases: despolarización y repolarización (**Figura 64**).

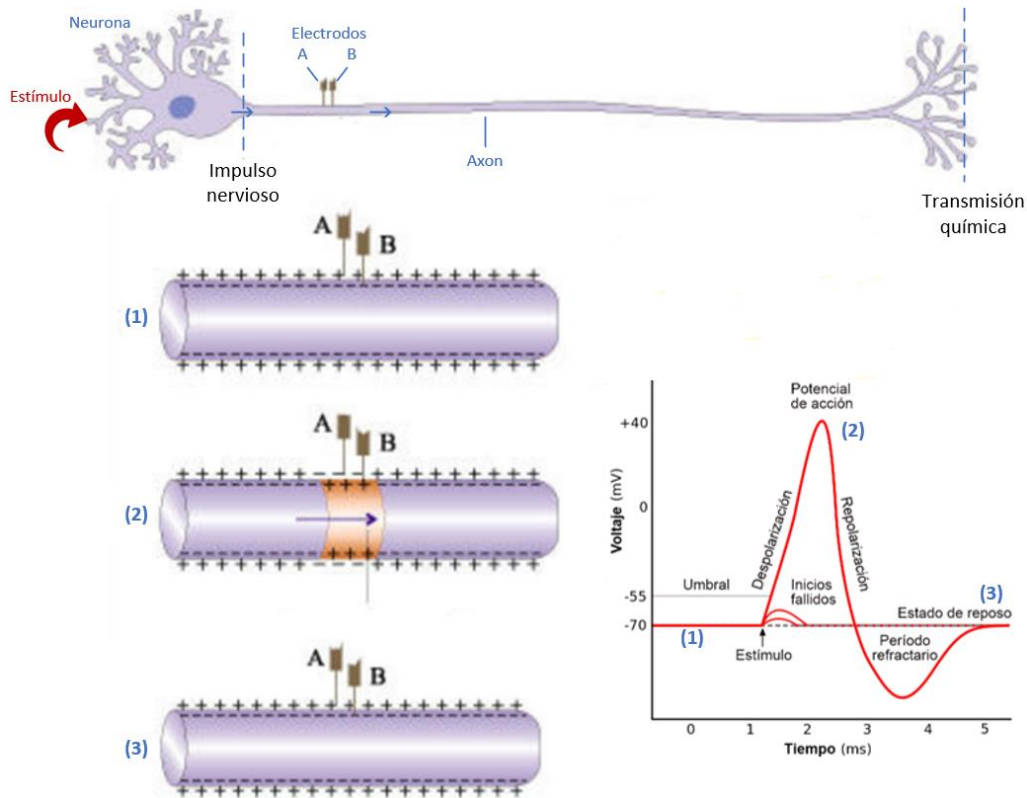


Figura 64. Diferencia de voltaje durante y después del potencial de acción. [35]

Durante la fase de despolarización, el interior de la célula se carga positivamente después de la estimulación celular debido a la apertura de los canales de  $\text{Na}^+$  presentes en la membrana. Estos iones cargados positivamente hacen que el interior de la membrana sea menos negativa y, si se alcanza el valor umbral, más canales de  $\text{Na}^+$  dependientes de voltaje se abrirán rápidamente, haciendo que el interior de la membrana sea positivo por un breve período de tiempo (aproximadamente +40 mV). Este cambio de voltaje provoca cambios de permeabilidad adicionales en la membrana plasmática, lo que hace que se detenga la despolarización y comience la repolarización. Durante esta fase, los canales de  $\text{Na}^+$  se cierran y el movimiento de  $\text{K}^+$  hacia el exterior de la célula aumenta, haciendo que el interior de la membrana plasmática se vuelva más negativo y el exterior más positivo. Una vez que se restablece el potencial de membrana en reposo, los canales de  $\text{K}^+$  activados por voltaje se cierran y el potencial de acción termina. Por tanto, la generación de un potencial de acción depende de si el estímulo es lo suficientemente fuerte como para alcanzar el valor umbral y provocar la despolarización. A esto se le llama el principio de todo o nada.

Aunque un potencial de acción ocurre en un área muy pequeña de la membrana plasmática, puede propagarse a través de ella estimulando la generación de otro potencial de acción en una ubicación adyacente, lo que también genera otro potencial de acción, y así sucesivamente.

### Unidad Motora

Por otra parte, la generación de señales eléctricas en el músculo se relaciona también con el concepto de unidad motora, que es el conjunto formado por una única neurona motora (o motoneurona) de la médula espinal o de un núcleo motor del tallo cerebral, y el grupo de fibras musculares que inerva. [36]

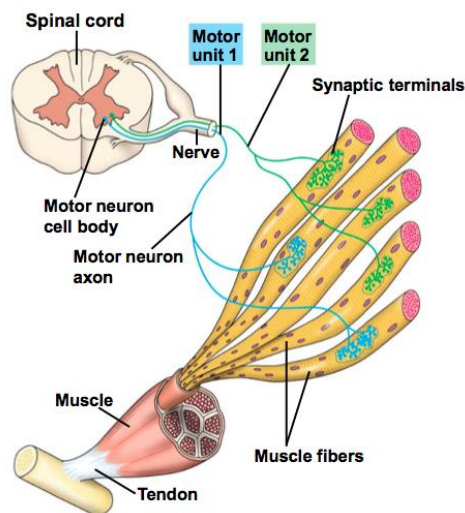


Figura 65. Esquema de dos ejemplos de unidades motoras y sus ramificaciones [37]

Si la motoneurona sufre una despolarización, ésta recorre todo el axón hasta las terminaciones sinápticas y provoca la despolarización, casi sincrónica, de todo el conjunto de fibras musculares de la unidad motora, en otras palabras, conduce los impulsos del cerebro y la médula espinal hacia los efectores (músculos) provocando la estimulación y consiguiente contracción de estos [33].

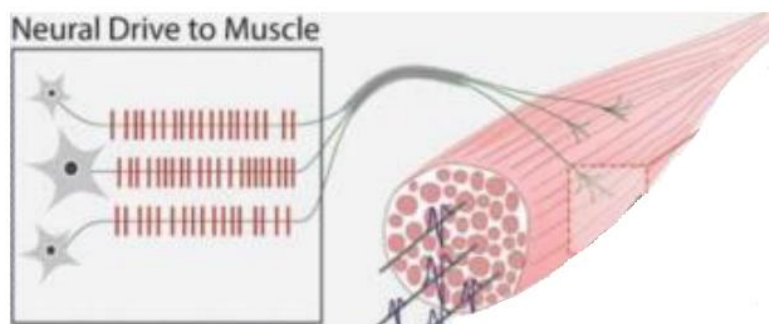


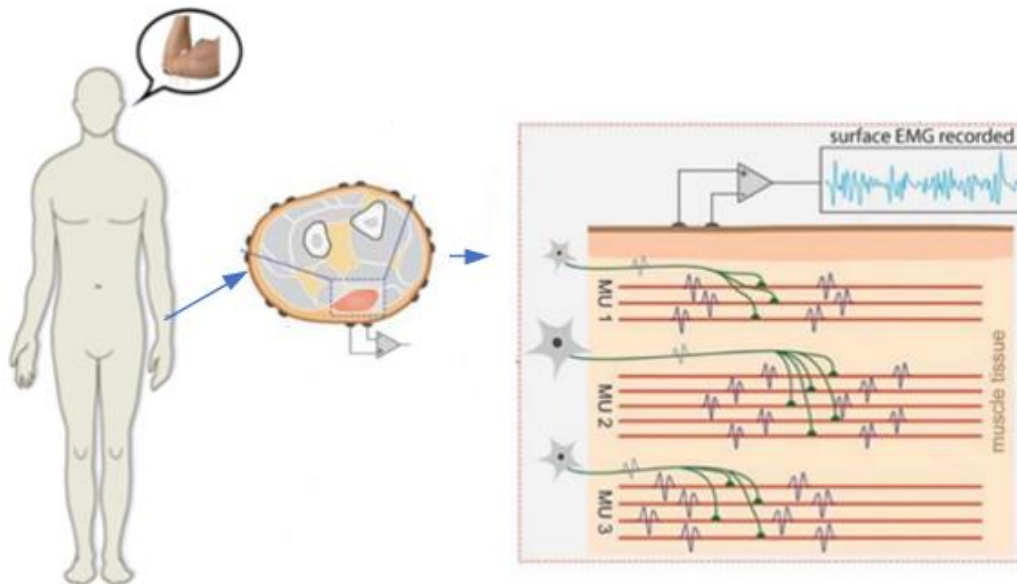
Figura 66. Propagación del potencial de acción a través de las fibras musculares. [36]

Como se puede observar en la **Figura 66**, el potencial de acción se propaga a lo largo de la fibra muscular en ambas direcciones a través de un sistema tubular. Esta excitación

provoca el desencadenamiento de diversos procesos químicos que terminan produciendo un acortamiento de los elementos contráctiles de la célula muscular, obteniéndose así el movimiento en el músculo efector a partir de un impulso nervioso.

### Composición de las señales EMG

Hasta el momento, se han considerado los potenciales de acción que recorren la fibra muscular como eventos individuales y distinguibles, sin embargo, dado que las despolarizaciones de las fibras de una unidad motora se superponen en el tiempo, la señal presente en la zona de detección (bajo los electrodos) consistirá en la superposición tanto espacial como temporal de las contribuciones de cada potencial de acción individual. La señal que se obtiene de esa superposición se denomina potencial de acción de la unidad motora (MUAP). [38]



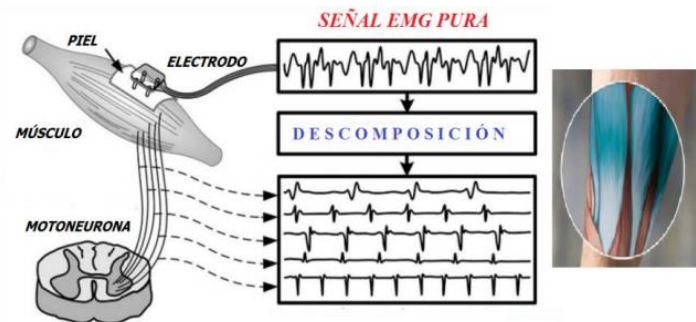
*Figura 67. Superposición de potenciales de acción para generar una señal EMG. [36]*

En conclusión, tal como se puede observar en la **Figura 67**, la señal EMG tomada de un músculo surge de la superposición de los potenciales de acción de las unidades motoras detectadas en el mismo.

### Adquisición y acondicionamiento de la señal EMG

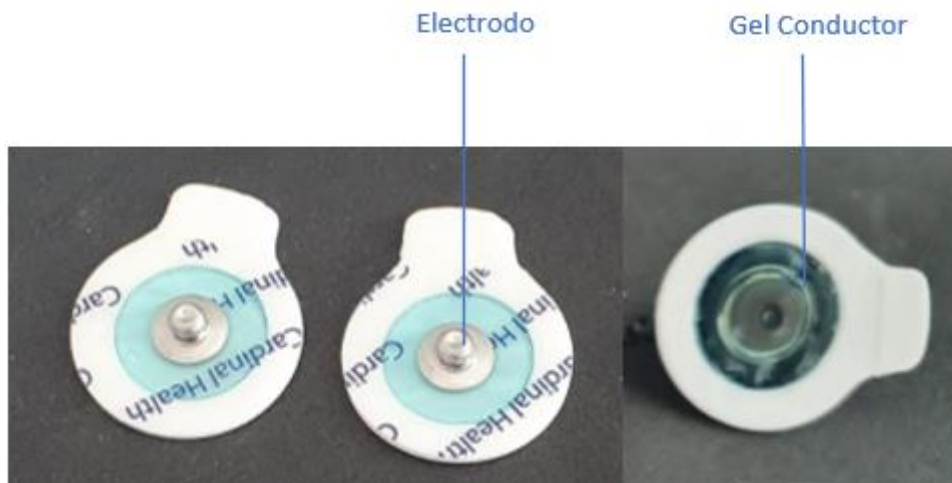
La forma de obtener información acerca de nuestro entorno y transferirla a algún aparato electrónico para su posterior procesado, se lleva a cabo mediante un transductor, el cual es un dispositivo capaz de transformar una magnitud física en una señal eléctrica. En el campo biomédico los transductores utilizados son llamados electrodos. Estos electrodos hacen una transferencia iónica del tejido vivo del cuerpo hacia un dispositivo electrónico, el cual se encargará posteriormente de su acondicionamiento.

En este proyecto se hará uso de electrodos de superficie, se les llama así porque son colocados directamente en contacto con la superficie de la piel. (**Figura 68**)



*Figura 68. Adquisición de la señal electromiográfica con electrodos de superficie. [39]*

Como podemos observar en la **Figura 68**, las señales EMGS serán recogidas a través de electrodos bipolares de superficie. A la hora de utilizar estos electrodos se debe tener en cuenta la elevada resistencia eléctrica natural de la piel, para ello es recomendable la aplicación de un gel que mejore la conductividad con el fin de lograr una mejor superficie de contacto que mejore la adherencia de los electrodos, de esta forma se logra mejorar la captura de señales EMG.



*Figura 69. Electrodos con gel conductor adhesivo.*

A pesar de estas disposiciones, las señales EMG recogidas suelen ser demasiado débiles, por lo que es necesario un procesamiento previo de filtraje y de amplificación antes de su uso para implementar un control mioeléctrico.

El proceso a seguir, una vez que la señal es recolectada por los electrodos, sería el de la **Figura 70**.

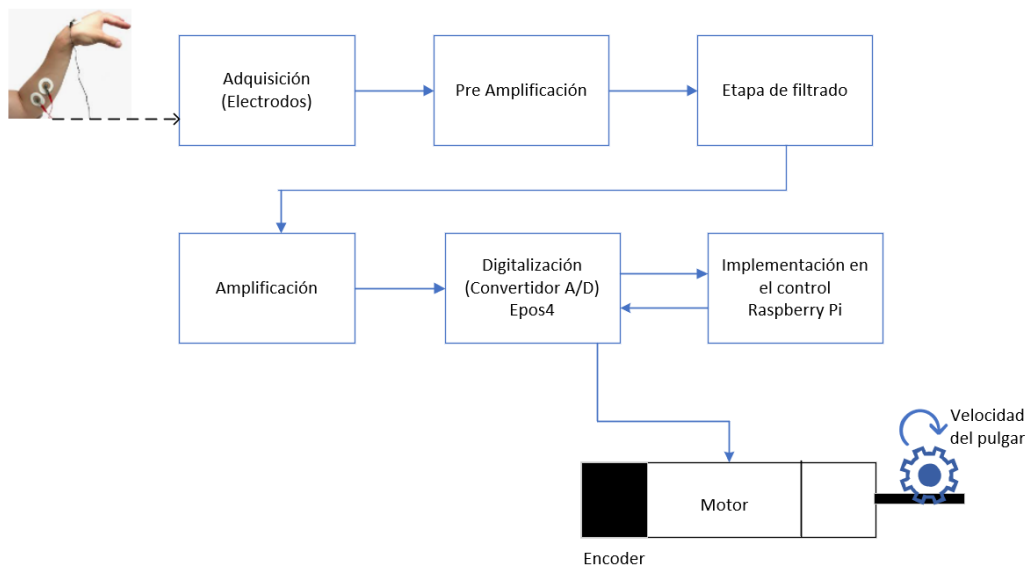


Figura 70. Diagrama de bloques del tratamiento y uso de la señal EMG

Para facilitar la implementación del control mioeléctrico, se ha decidido utilizar un sensor muscular electromiográfico que llevase incorporado de forma interna todo el proceso amplificación y filtrado.

Teniendo esto en cuenta, el sensor muscular que se ha elegido fue el de la empresa MyoWare [40], el cual nos entrega una señal EMG ya tratada, lista para ser leída directamente por algún controlador que disponga de entradas analógicas, como es el caso de la Epos4 compact.

### **Sensor Muscular MyoWare**

Como se comentó anteriormente, este proyecto hará uso del sensor muscular EMG de la empresa MyoWare, el cual hace uso de electrodos de superficie, que se colocan directamente sobre la superficie de la piel para detectar la señal electromiográfica, la cual como vimos anteriormente, está compuesta por la superposición de los potenciales de acción de las unidades motoras activas. Actualmente los electrodos de Ag/AgCL son muy utilizados en el ámbito médico. Estos electrodos utilizan un gel electrolítico para incrementar la conductividad entre la piel y el electrodo.

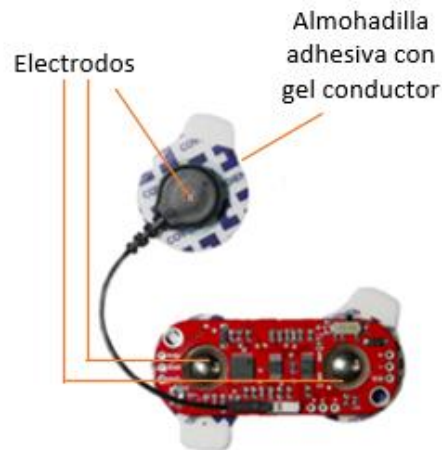


Figura 71. Sensor muscular MyoWare compuesto por tres electrodos. [40]

Entre las características más importantes del sensor muscular MyoWare destacan [40]:

- Voltaje de operación: [2.9V – 5.7V]
- Ganancia Ajustable
- Salida RAW EMG sin procesar y procesada
- Pines protegidos contra polaridad inversa
- Diseño específico para el uso con microcontroladores
- Indicadores LED
- Expansión vía shields
- Dimensiones: [0,82"-2,06"]
- Soporte con una gran comunidad de desarrollo detrás

La ubicación de los pines más importantes los podemos ver en la **Figura 72**. De entre todos ellos destacan los pines 1, 2 y 3. A través del pin 1 se alimentará al sensor con 5V, y a través del pin 2 y 3 se enviará la señal EMG a la entrada analógica de la Epos4.

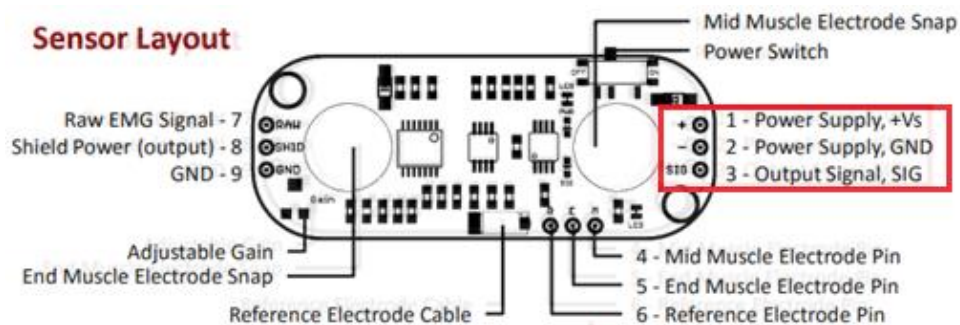


Figura 72. Disposición de los pines del sensor MyoWare. [40]

En este proyecto se hará un uso directo de la señal EMG entregada por el sensor, para ello se hará la conexión con una de las entradas analógicas de la Epos4. Para saber cuáles son los pines analógicos de la controladora Epos4, el fabricante proporciona el documento Hardware Reference [29]. En ese documento se puede observar, por un



lado, que las entradas analógicas de la EPOS4 corresponden con el conector x8 y por otro lado la distribución y la descripción del uso de cada uno de los 7 pines que conforman al conector X8.

Conector X8



X8 Head A Pin	Prefab Cable Color	Head B Pin	Signal	Description
1	white	1	AnIN1+	Analog input 1, positive signal
2	brown	2	AnIN1-	Analog input 1, negative signal
3	green	3	AnIN2+	Analog input 2, positive signal
4	yellow	4	AnIN2-	Analog input 2, negative signal
5	grey	5	AnOUT1	Analog output 1
6	pink	6	AnOUT2	Analog output 2
7	blue	7	GND	Ground

Tabla 10. Asignación de pines del conector analógico X8 de la EPOS4. [29]

Teniendo esto en cuenta, la conexión entre el sensor muscular MyoWare y la controladora Epos4 nos quedaría como en la **Figura 73**.

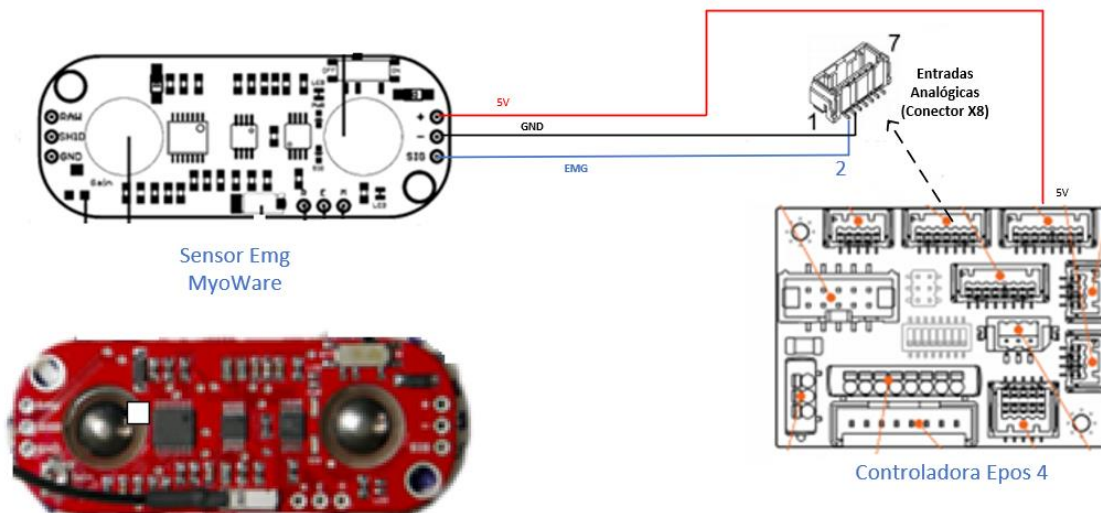


Figura 73. Conexión Sensor MyoWare – Controladora Epos4. [29]

### Colocación del sensor

Antes de colocar los electrodos que componen el sensor, se tiene que preparar la piel sobre la que se van a colocar los electrodos. Para ello es importante exfoliar y limpiar con alcohol la piel, de esta forma se reducirá la impedancia (traducida en ruido) que pueda sufrir la señal EMG.

En este proyecto se ha seguido las recomendaciones del experimento [41], en el cual se dan recomendaciones sobre el posicionamiento de los electrodos en el antebrazo de cara a clasificar las señales electromiográficas.

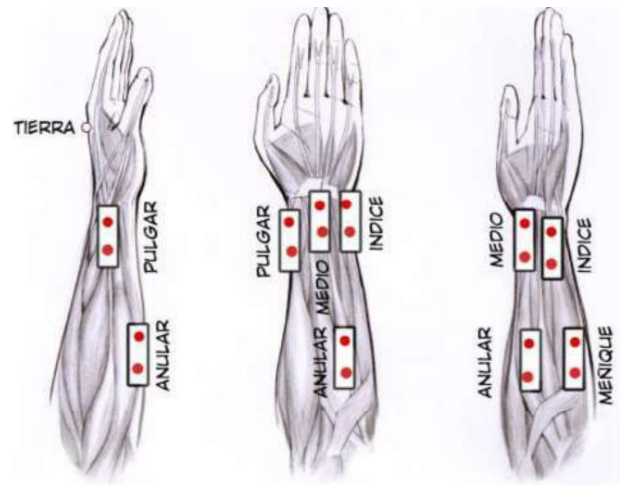


Figura 74. Alternativas para la colocación de los diferentes sensores EMG superficiales en el antebrazo. [41]

De entre todas las opciones que se muestran en la **Figura 74**, se ha prestado especial atención a aquella en la que se vea involucrada activamente la musculatura que interviene en el movimiento del dedo pulgar, como son el abductor largo (Abductor longus pollicis) y el extensor corto del pulgar (Extensor pollicis brevis). Teniendo esto en cuenta, la colocación de los electrodos será como el representado en la **Figura 75**.

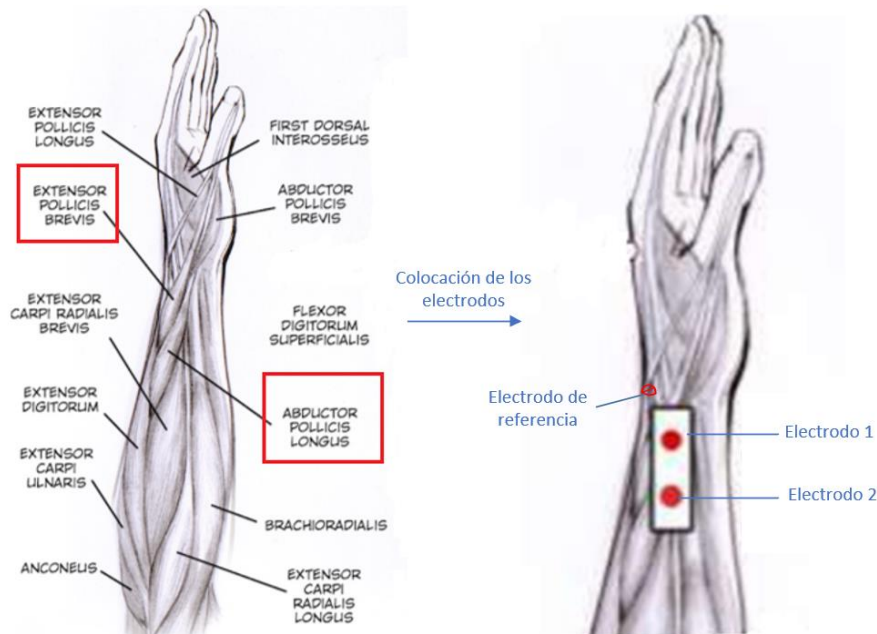
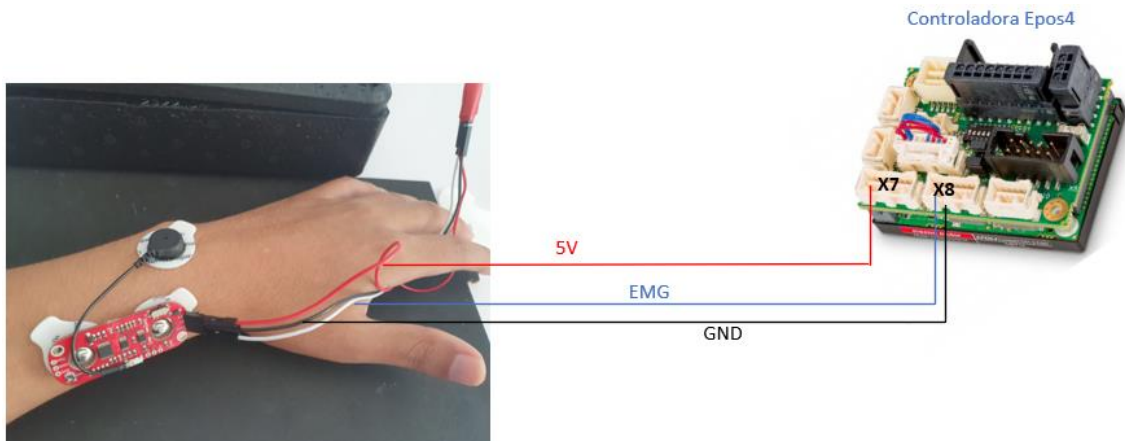


Figura 75. Colocación de los electrodos. [41]

En conclusión, el sensor muscular MyoWare, irá adherido a la muñeca. Es importante tener en cuenta que la localización de los electrodos tiene que ser paralela a las fibras del músculo. Es por esto que dos de sus electrodos harán contacto con los músculos



involucrados en el movimiento del pulgar y el tercero será de referencia. Por otro lado, la señal electromiográfica acondicionada y leída por el sensor será enviada hacia la controladora Epos4, la cual dispone de lectura de señales analógicas a través de su conector X8. A su vez, la controladora Epos4 suministrará la alimentación de 5V al sensor muscular a través de una de sus salidas auxiliares incorporadas en su conector X7. Este proceso queda esquematizado en la **Figura 76**.



*Figura 76. Conexión y posicionamiento del sensor MyoWare.*

Con esta disposición, se facilitará el cumplir con uno de los objetivos propuestos en este proyecto, el cual era posicionar, todos los elementos que componen la prótesis del dedo pulgar protésico, en un soporte en forma de banda colocada alrededor de la muñeca. Esta disposición sería similar a la que utiliza el brazalete del modelo comercial Vincent Partial Hand [15], el cual ya fue comentado en el capítulo en el que se ha tratado el estado del arte.

Llegados a este punto, ya es posible la implementación de un algoritmo de control mioeléctrico en el dedo pulgar protésico.

### 5.1.2 Algoritmo de control mioeléctrico

Finalmente, en la **Figura 77** se puede observar el esquema general de conexión e interacción de los diferentes elementos que forman el sistema del dedo pulgar protésico.

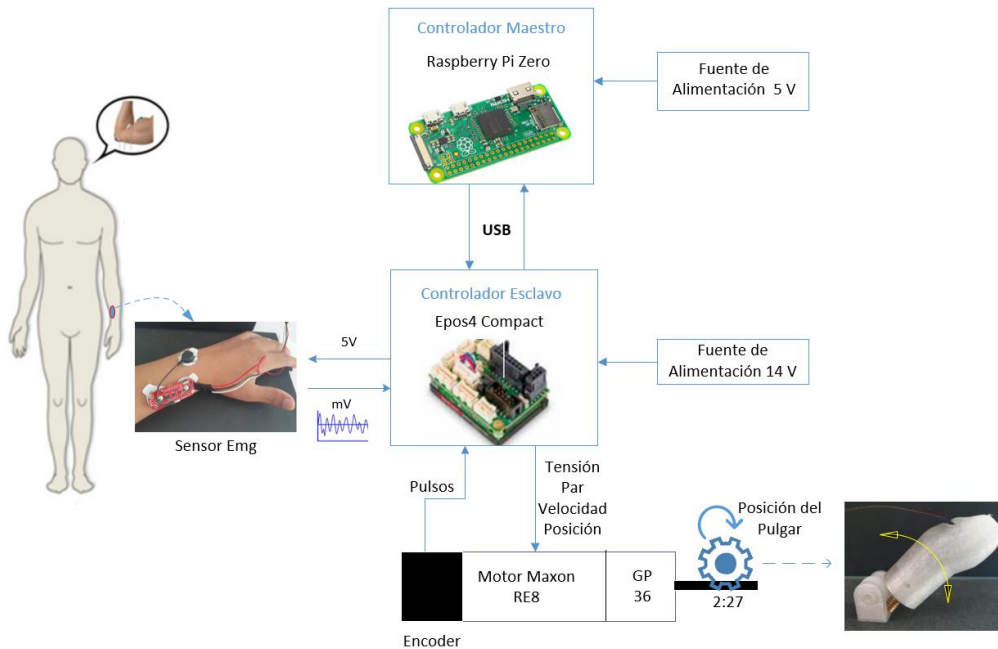


Figura 77. Esquema general de funcionamiento

El algoritmo de control se implementará por módulos. Cada módulo funcionará de forma independiente. Se empezará con un primer módulo que llevará la programación correspondiente a la comunicación vía USB entre la raspberry pi y la Epos4. Posteriormente sobre la base anterior, se creará el segundo módulo que contendrá el código correspondiente a la interacción con el sensor electromiográfico, y finalmente el tercer módulo contendrá a los módulos anteriores y posicionará al dedo protésico en flexión o en extensión (**Figura 78**).

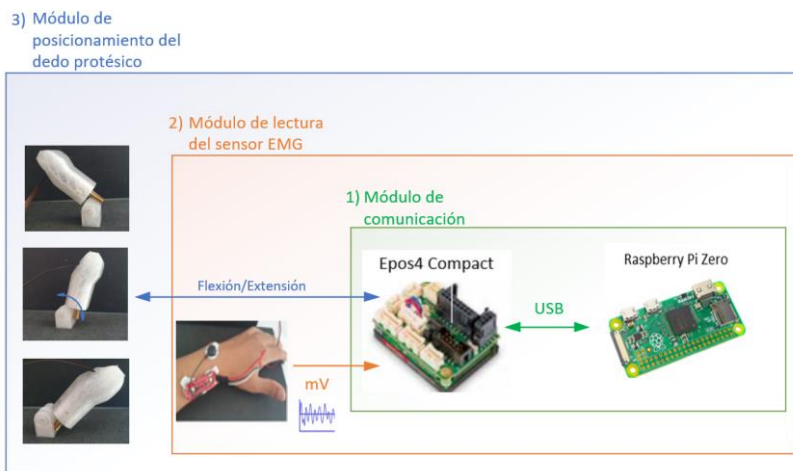


Figura 78. Diagrama de bloques por módulos de la solución completa.

### 1) Módulo de comunicación Epos4 – Raspberry Pi

Lo primero que se ha hecho es alimentar a la Raspberry con 5 V y a la Epos4 con 14V, posteriormente se los ha comunicado con un cable vía puerto USB.

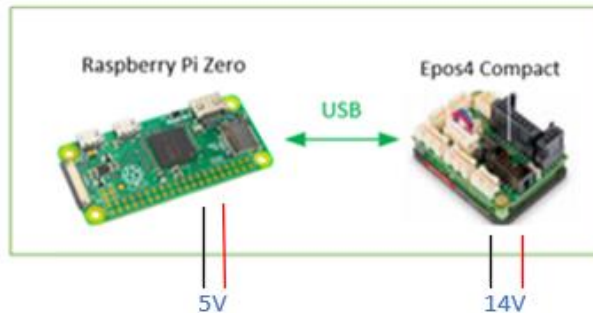


Figura 79. Comunicación vía USB.

Para lograr comunicación entre estos dos dispositivos, se ha instalado en la Raspberry Pi, la librería “Epos Linux Library” la cual nos proporciona el fabricante de la Epos4 en su página web en el apartado de descargas [30].

SOFTWARE/FIRMWARE	
ENGLISH	
↓ Setup EPOS, IDX - Studio 3.7 (english, ZIP -1 bytes)	Version 35
↓ EPOS IDX Setup Release Notes (english, PDF 581 KB)	Version April 2021
↓ EPOS Command Library Documentation (english, PDF 11 MB)	Version March 2021
↓ EPOS Windows 32-Bit and 64-Bit DLL (english, ZIP 97 MB)	Version 6.7.1.0
↓ EPOS .Net Library (english, ZIP 32 MB)	Version 6.7.1.0
↓ EPOS LabVIEW VI's for PC (english, ZIP 19 MB)	Version 6.7.1.0
↓ EPOS Linux Library (ARM/Intel 32/64-Bit) (english, ZIP 7 MB)	Version 6.7.1.0

Figura 80. Librerías de la Epos4 compatibles con Linux. [30]

Las instrucciones y los pasos a seguir en la instalación, se encuentran en la página 163 del documento “EPOS Command Library Documentation” [30], también proporcionados por el fabricante. De la misma forma, dentro del documento “EPOS Command Library Documentation”, se encuentran las indicaciones en detalle para utilizar cada una de las funciones disponibles en la librería (Figura 81).

SOFTWARE/FIRMWARE	
ENGLISH	
↓ Setup EPOS, IDX - Studio 3.7 (english, ZIP -1 bytes)	Version 35
↓ EPOS IDX Setup Release Notes (english, PDF 581 KB)	Version April 2021
↓ EPOS Command Library Documentation (english, PDF 11 MB)	Version March 2021
↓ EPOS Windows 32-Bit and 64-Bit DLL (english, ZIP 97 MB)	Version 6.7.1.0

9 Supported Operating Systems	149
9.1 Windows	149
9.1.1 Overview	149
9.1.2 Integration into Programming Environment	151
9.1.3 Programming	160
9.2 Linux	163
9.2.1 Overview	163
9.2.2 Installation / Uninstallation	164
9.2.3 Integration into Programming Environment	165
9.2.4 Programming	166

Figura 81. Manual explicativo de la librería Epos Command y sus funciones en Linux. [30]

Después de seguir los pasos e instalar la librería Epos Command en la Raspberry Pi Zero, la cual recordemos tiene un sistema operativo basado en una distribución Linux, se ha empezado a programar la Raspberry Pi Zero en el lenguaje de programación C++, utilizando las funciones incorporadas en la librería Epos Command. Por otro lado, la escritura y la compilación del código se han realizado a través del entorno de programación Geany, el cual fue instalado previamente en la raspberry pi (**Figura 82**).

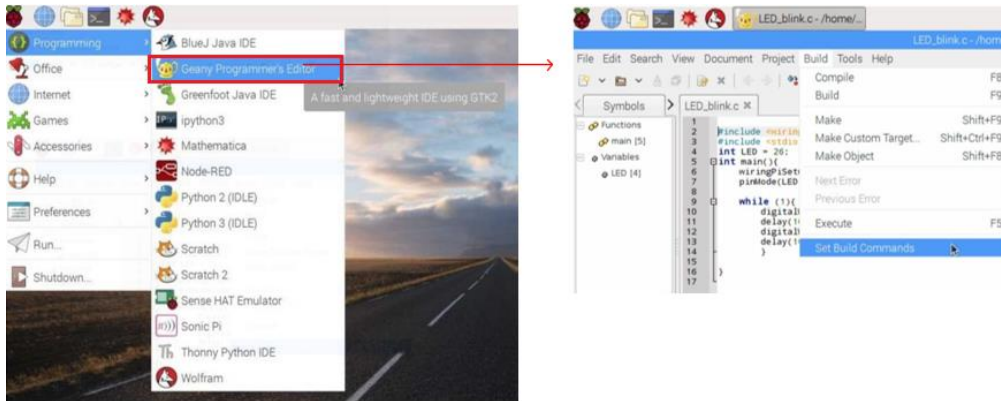


Figura 82. Entorno de programación Geany.

Dentro de las funciones disponibles en la librería Epos Command, se encuentra una función específica que se encarga de establecer la comunicación entre la EPOS4 y otro dispositivo basado en una arquitectura Linux. Esta función es “VCS\_OpenDevice ()”, su estructura esta descrita en la **Figura 83** y es la encargada de abrir el puerto de la Epos4 para enviar y recibir comandos. Los puertos pueden ser interfaces RS232, USB y CanOpen. En este proyecto se ha usado el puerto USB.

### VCS\_OpenDevice

#### FUNCTION

HANDLE VCS\_OpenDevice(char\* DeviceName, char\* ProtocolStackName, char\* InterfaceName, char\* PortName, DWORD\* pErrorCode)

#### PARAMETERS

Parameter Name	Type	Description
DeviceName	char*	Name of connected device: <ul style="list-style-type: none"> <li>EPOS</li> <li>EPOS2</li> <li>EPOS4 (Note: Also used for IDX drives)</li> </ul>
ProtocolStackName	char*	Name of used communication protocol: <ul style="list-style-type: none"> <li>MAXON_RS232</li> <li>MAXON SERIAL V2</li> <li>CANopen</li> </ul>
InterfaceName	char*	Name of interface: <ul style="list-style-type: none"> <li>RS232</li> <li>USB</li> <li>IXXAT_&lt;&lt;BoardName&gt;&gt; &lt;&lt;DeviceNumber&gt;&gt;</li> <li>Kvaser_&lt;&lt;BoardName&gt;&gt; &lt;&lt;DeviceNumber&gt;&gt;</li> <li>NI_&lt;&lt;BoardName&gt;&gt; &lt;&lt;DeviceNumber&gt;&gt;</li> <li>Vector_&lt;&lt;BoardName&gt;&gt; &lt;&lt;DeviceNumber&gt;&gt;</li> </ul> Remark: Use "VCS_OpenDeviceDlg" or "VCS_GetInterfaceNameSel" to identify the exact name
PortName	char*	Name of port: <ul style="list-style-type: none"> <li>COM1, COM2, ...</li> <li>USB0, USB1, ...</li> <li>CAN0, CAN1, ...</li> </ul>

Figura 83. Extracto de la página 13 del documento Epos Command Library. [30]

Siguiendo la estructura de la función VCS\_OpenDevice descrita anteriormente, se ha programado la parte del algoritmo que se encarga de la comunicación vía USB entre la

Raspberry Pi y la Epos4. Primero se han declarado los parámetros que se pasarán a la función VCS\_OpenDevice (**Figura 84**).

```
void SetDefaultParameters()
{
    //USB
    g_usNodeId = 1;
    g_deviceName = "EPOS4";
    g_protocolStackName = "MAXON SERIAL V2";
    g_interfaceName = "USB";
    g_portName = "USB0";
    g_baudrate = 1000000; //tasa de baudios. Velocidad de transmisión.
}
```

*Figura 84.* Extracto del código donde se declaran los parámetros de comunicación.

Estos parámetros se pasarán a la función VCS\_OpenDevice (), la cual será llamada a través de la función OpenDevice () que a su vez es llamada desde la función principal main ().

```
int main(int argc, char** argv)
{
    int lResult = MMC_FAILED;
    unsigned int ulErrorCode = 0;
    SetDefaultParameters();
    if ((lResult = OpenDevice(&ulErrorCode)) != MMC_SUCCESS)
    {
        cout<<"OpenDevice ha fallado"<<lResult<<ulErrorCode;
        return lResult;
    }
}

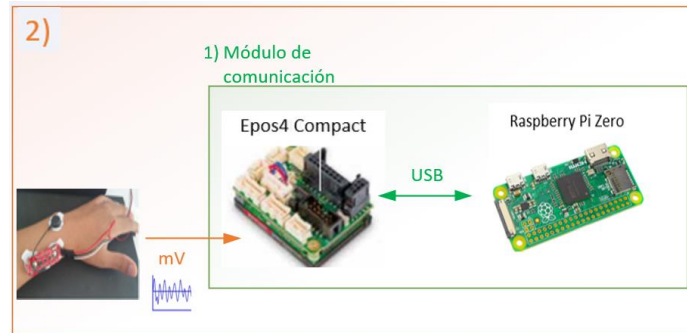
int OpenDevice(unsigned int *p_pErrorCode)
{
    g_pKeyHandle = VCS_OpenDevice(pDeviceName, pProtocolStackName, pInterfaceName, pPortName, p_pErrorCode);
}
```

*Figura 85.* Extracto fraccionado del código encargado de la comunicación.

Al compilar y ejecutar el código de este primer módulo en la raspberry Pi se enviará la orden a la Epos4 de abrir los puertos para establecer el envío y el recibimiento de comandos. Sobre esta base se añadirán los demás fragmentos de código hasta completar el sistema el sistema de control mioeléctrico al completo.

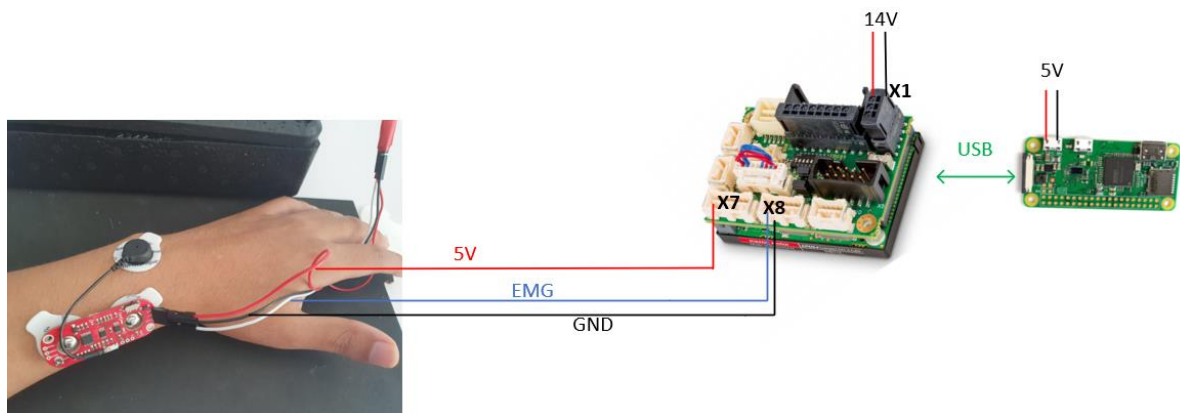
## 2) Módulo de lectura del sensor muscular

El módulo dos, será la parte encargada de la lectura y el procesamiento de la señal electromiográfica enviada por el sensor muscular. A su vez llevará incluido y hará uso del módulo uno de comunicación que vimos anteriormente (**Figura 86**).



*Figura 86. Bloques del módulo dos.*

Para programar este módulo primero se realizará la conexión correspondiente. Los pines a utilizar de la Epos4 serán: X1 para alimentar la Epos4 con 14V, X7 para alimentar al sensor muscular con 5V y X8 para la leer la señal electromiográfica enviada por el sensor muscular. Por otro lado, se alimentará la raspberry Pi con 5V y se le conectará vía USB con la Epos4 (**Figura 87**).



*Figura 87. Conexión de lectura y procesamiento de señal EMG*

Después de realizar el conexionado, se ha programado el segundo módulo, tomando como base el código implementado en el módulo uno que corresponde a la comunicación.

La función que se ha añadido al algoritmo ha sido:

- **VCS\_GetAnalogInputVoltage:** Esta función devuelve el valor leído en milivoltios en una de las entradas analógicas de la Epos4. La estructura y el cómo hacer uso de ella vienen explicadas en el documento Epos Command Library (**Figura 88**).



**VCS\_GetAnalogInputVoltage**

**FUNCTION**  
 BOOL VCS\_GetAnalogInput(HANDLE KeyHandle, WORD NodeId, WORD InputNumber, WORD\* pAnalogValue, DWORD\* pErrorCode)

**PARAMETERS**

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node-ID of the addressed device
InputNumber	WORD	Analog input number

**RETURN PARAMETERS**

pAnalogValue	WORD*	Analog value from input
pErrorCode	DWORD*	Error information on the executed function
<b>Return Value</b>	BOOL	Nonzero if successful; otherwise "0"

Figura 88. Extracto de la página 117 del documento Epos Command Library. [30]

Teniendo esto en cuenta, se ha añadido una nueva función en el main, la cual hace uso de VCS\_GetAnalogInputVoltage (), para obtener el valor en mV de la señal electromiográfica enviada por el sensor muscular MyoWare (Figura 89).

```
int main(int argc, char** argv)
{
    int lResult = MMC_FAILED;
    unsigned int ulErrorCode = 0;
    SetDefaultParameters();
    if ((lResult = OpenDevice(&ulErrorCode)) != MMC_SUCCESS)
    {
        cout<<"OpenDevice ha fallado"<<lResult<<ulErrorCode;
        return lResult;
    }
    if(lResult = AnalogInput(g_pKeyHandle, g_usNodeId, &ulErrorCode)!=MMC_SUCCESS)
    {
        cout<<"AnalogInput ha fallado"<<lResult<<ulErrorCode<<endl;
        return lResult;
    }
}

int AnalogInput(HANDLE p_DeviceHandle, unsigned short p_usNodeId, unsigned int * p_rlErrorCode)
{
    int lResult=MMC_SUCCESS;
    unsigned int lErrorCode = 0;
    unsigned short InputNumber=1; //WORD InputNumber, Corresponde con A1+

    if(VCS_GetAnalogInputVoltage(p_DeviceHandle, p_usNodeId, InputNumber, pVoltageValue, p_rlErrorCode) ==0)
    {
        cout<<"Error en VCS_GetAnalogInputVoltage"<<endl;
        lResult = MMC_FAILED;
    }
}
```

Figura 89. Extracto fraccionado del código encargado de la lectura del sensor EMG.

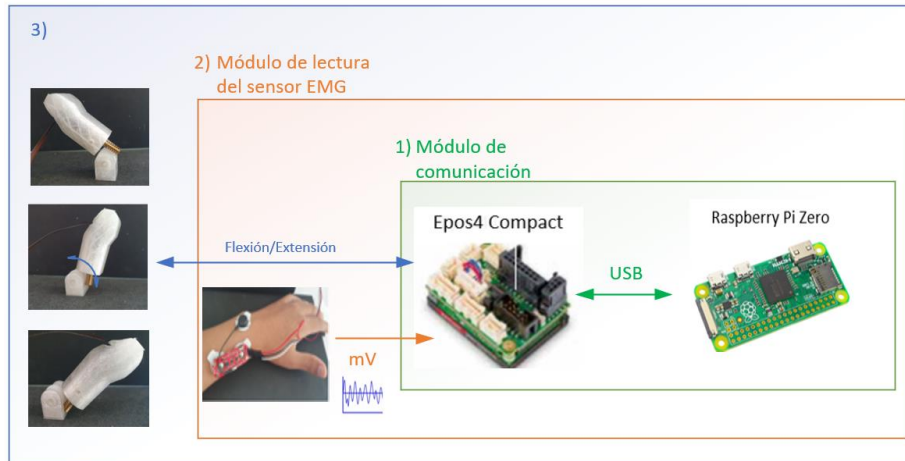
En la figura anterior se puede observar la variable “pVoltageValue”, esta variable será la encargada de almacenar el valor leído en el sensor muscular en un rango entre 0 y 5000 mV.

La idea es declarar una variable de control con un determinado valor umbral. Tal que cuando el valor leído del sensor muscular y almacenado en “pVoltageValue” sobrepase este valor umbral, se envíen las órdenes para mover el motor a una posición determinada, posicionando así el dedo protésico. Esto se implementará en el algoritmo de control con un bucle if en el siguiente módulo.



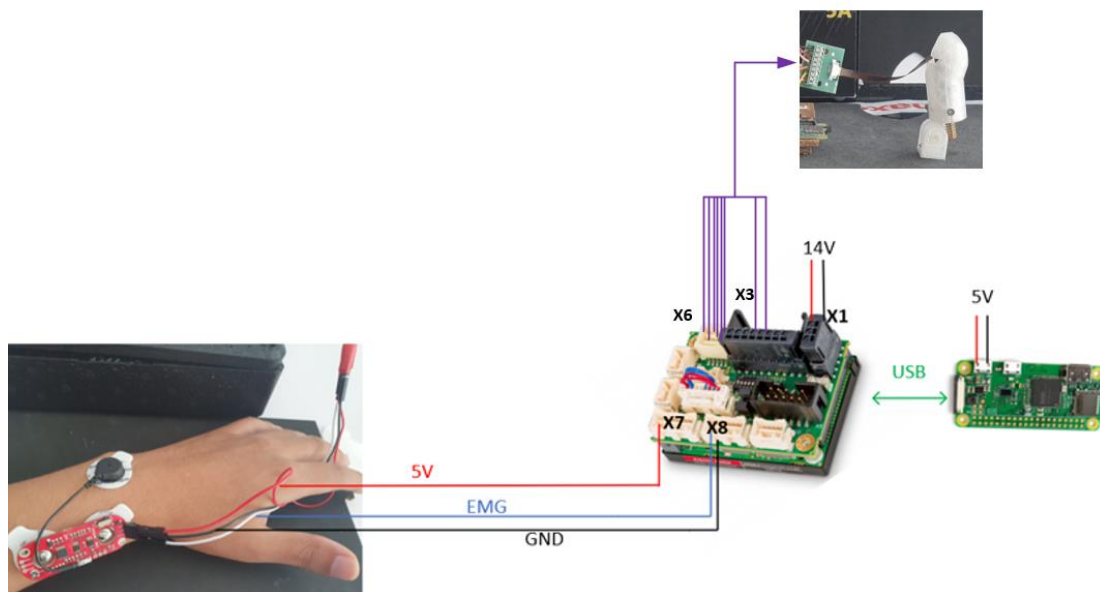
### 3) Módulo de posicionamiento del pulgar protésico

Este último módulo hará uso de los dos anteriores y en él se llevará a cabo la programación del algoritmo de control mioeléctrico que posicionará el dedo protésico en flexión/extensión en función de la señal electromiográfica leída por el sensor muscular (**Figura 90**).



*Figura 90. Bloques que conforman el módulo del sistema final.*

Para programar este último módulo primero se realizará la conexión correspondiente. Los conectores de la Epos4 que se utilizarán serán: X1 para alimentar la Epos4 con 14V, X7 para alimentar al sensor muscular con 5V, X8 para la leer la señal electromiográfica enviada por el sensor muscular, X3 para alimentar al motor y X6 para alimentar y leer los pulsos del encoder. Por otro lado, se alimentará la raspberry Pi con 5V y se le conectará vía USB con la Epos4 (**Figura 91**).



*Figura 91. Conexión del sistema final.*

Es importante observar que el motor lleva un conector Molex a la salida del encoder, este conector consta de 8 pines, los cuales ya están preparados para conectar directamente el conector X6 y X3 de la Epos4 (**Figura 92**).

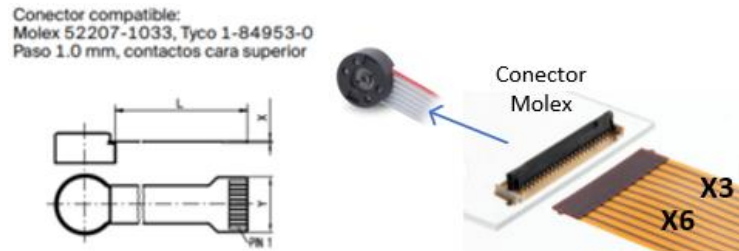


Figura 92. Conector Molex. [42]

Las principales funciones que se han añadido al algoritmo respecto a los ya implementados anteriormente han sido:

- **VCS\_GetEnableState:** Esta función comprueba si la Epos4 está habilitada.
- **VCS\_SetEnableState:** Esta función cambia el estado de la Epos4 ha habilitado.
- **VCS\_ActivateProfilePositionMode:** Esta función cambia el modo de operación a “Modo de operación de perfil de posición o punto a punto”.
- **VCS\_MoveToPosition:** Esta función inicia el movimiento con el perfil de posición seleccionado a la posición de destino.
- **VCS\_HalfPositionMovement:** Esta función detiene el movimiento con un perfil de desaceleración.

Teniendo esto en cuenta, se ha implementado el posicionamiento del motor condicionado a un valor umbral de 1,5V en la lectura del sensor muscular.

```

if(VCS_GetAnalogInputVoltage(p_DeviceHandle, p_usNodeId, InputNumber, pVoltageValue)
{
    cout<<"Error en VCS_GetAnalogInputVoltage"<<endl;
    lResult = MMC_FAILED;
}

cout<<"El valor de pVoltageValue es : "<<*pVoltageValue<<endl;

if((*pVoltageValue > 1500) && (cerrado == 0))
{
    cout<<"Dedo pulgar en flexión"<<endl;
    lResult = DemoProfilePositionMode(g_pKeyHandle, g_usNodeId, lErrorCode);
}
    
```

Figura 93. Fragmento del código que clasifica la señal electromiográfica.

Como podemos observar en la **Figura 93**, si el sensor muscular entrega un valor superior a 1500mV, el dedo pulgar estará en flexión. Para lograr esto, se llamará a la función DemoProfilePositionMode (), la cual contiene a las funciones VCS\_ActivateProfilePositionMode () y a VCS\_MoveToPosition (). Estas dos últimas son necesarias para mover el motor desde una posición a otra y su uso queda reflejado en la siguiente figura.

```

if(cerrado == 0)
{
    targetPosition = 22680;
}
cout << "Set profile position mode, node = " << p_usNodeId << endl;
if(VCS_ActivateProfilePositionMode(p_DeviceHandle, p_usNodeId, &p_rlErrorCode) == 0)
{
    cout << "Error en el activar el perfil de posicionamiento" << endl;
    lResult = MMC_FAILED;
}
else
{
    cout << "move to position = " << targetPosition << ", node = " << p_usNodeId << endl;
    if(VCS_MoveToPosition(p_DeviceHandle, p_usNodeId, targetPosition, 0, 1, &p_rlError
    {
        cout << "Ha fallado el llamado a VCS_MoveToPosition" << endl;
        lResult = MMC_FAILED;
        //break;
    }
}
    
```

Figura 94. Fragmento del código que posiciona el dedo en flexión.

Como podemos ver en la figura anterior, se ha declarado una variable llamada `targetPosition`. En esta variable se almacenará las posiciones a las que queremos mover el dedo protésico. Estas posiciones se almacenan en unidades de incrementos. Las conversiones de grados a incrementos se hacen de la siguiente manera:

$$4 \times \frac{\text{pulses}}{\text{revolutions}} = \frac{\text{increments[inc]}}{\text{revolutions[rev]}} = \frac{\text{quadcounts[qc]}}{\text{revolutions[rev]}}$$

Teniendo en cuenta que el encoder tiene una resolución de 100 pulsos por vuelta:

$$1 \text{ vuelta del dedo} = \frac{13,5 \text{ vueltas sin fin}}{1 \text{ vuelta del dedo}} \times \frac{36 \text{ vueltas motor}}{1 \text{ vuelta sin fin}} \times \frac{4 \times 100 \text{ pulsos}}{1 \text{ vuelta motor}} = 194400 \text{ inc}$$

Por lo tanto, para posicionar el dedo protésico en flexión, lo cual corresponde a 42° respecto a su posición de reposo, la variable `targetPosition` tiene que tener un valor aproximado de 22680 inc tal y como se puede ver en la **Figura 95**.

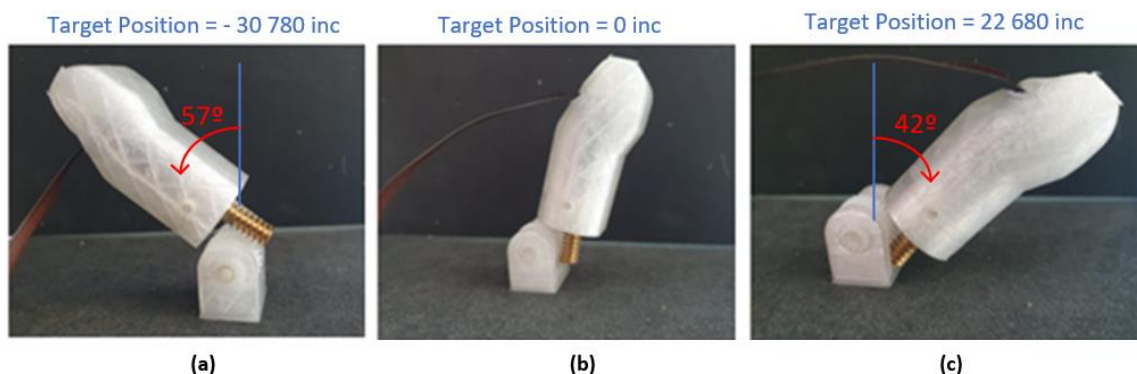
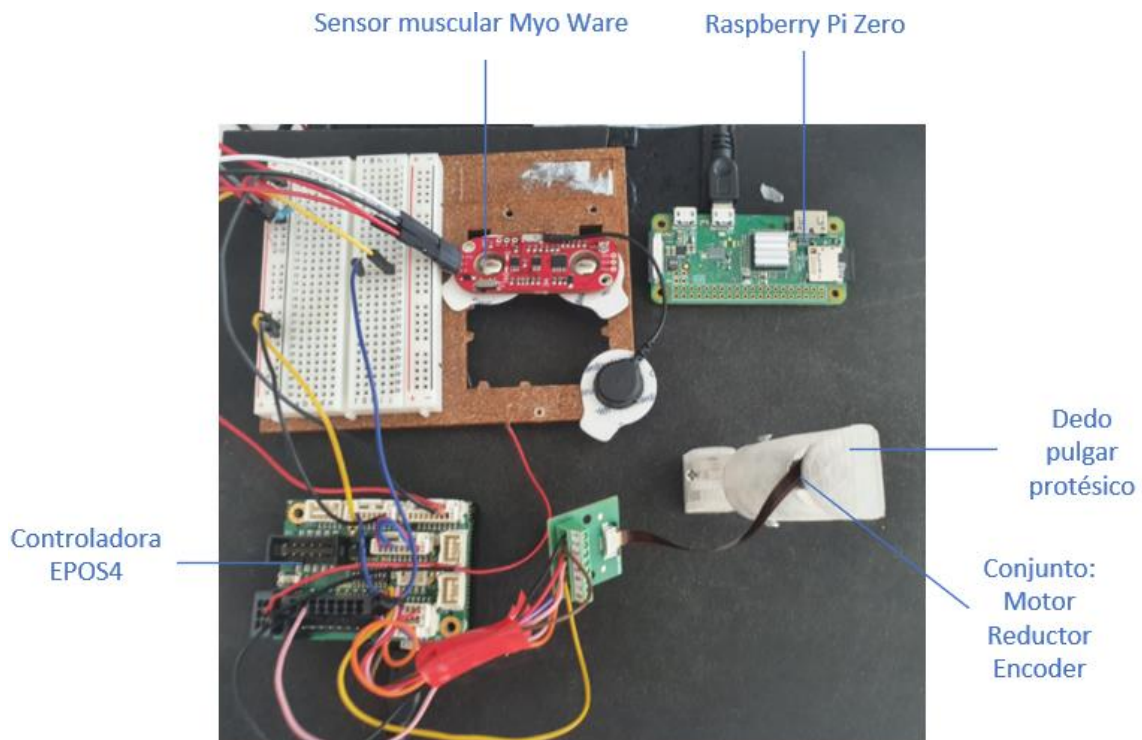


Figura 95. Dedo pulgar protésico en extensión (a), en reposo (b) y en flexión (c).

## 5.2 Pruebas y resultados

En este apartado se realizarán unas pruebas para comprobar el correcto funcionamiento de todo el sistema en su conjunto. Los elementos que conforman el sistema del dedo pulgar protésico accionado por control mioeléctrico son los de la **Figura 96**.



*Figura 96. Elementos que conforman el sistema del pulgar protésico accionado por control mioeléctrico.*

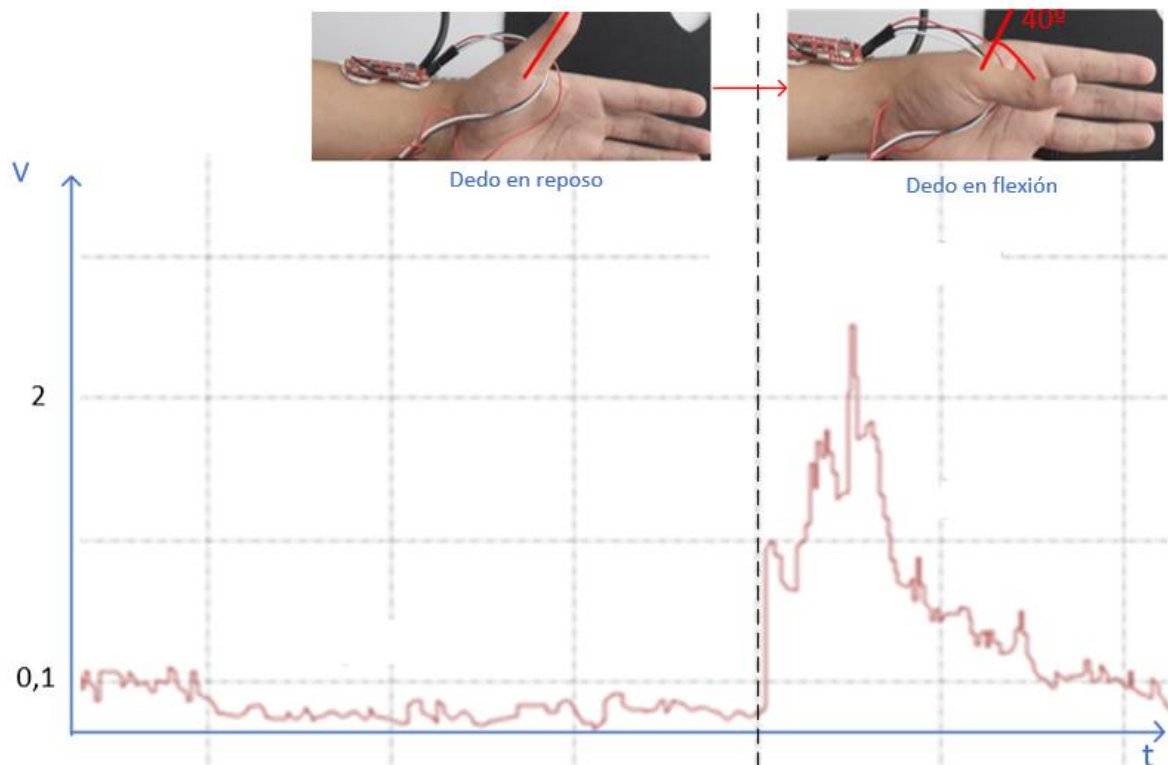
El conexionado y el uso de cada uno de los elementos que conforman el sistema, ya han sido descritos en capítulos anteriores. Teniendo todo esto en cuenta se realizarán dos pruebas. La primera prueba consistirá en posicionar el pulgar protésico en flexión/extensión dependiendo de la lectura de la señal electromiográfica enviada por el sensor muscular. La segunda prueba tratará de un ensayo, en el cual se añadirá una carga al dedo protésico y se comprobará si este es capaz de mantener su posición

### 5.2.1 Prueba 1: Control mioeléctrico del pulgar protésico

#### **Calibración del sensor muscular MyoWare**

Antes de realizar la prueba, se ha calibrado el sensor muscular, para ello se ha utilizado la aplicación EposStudio la cual nos proporciona unas gráficas a tiempo real de los valores leídos en la entrada analógica de la Epos4 a la cual esta conectada el sensor muscular.

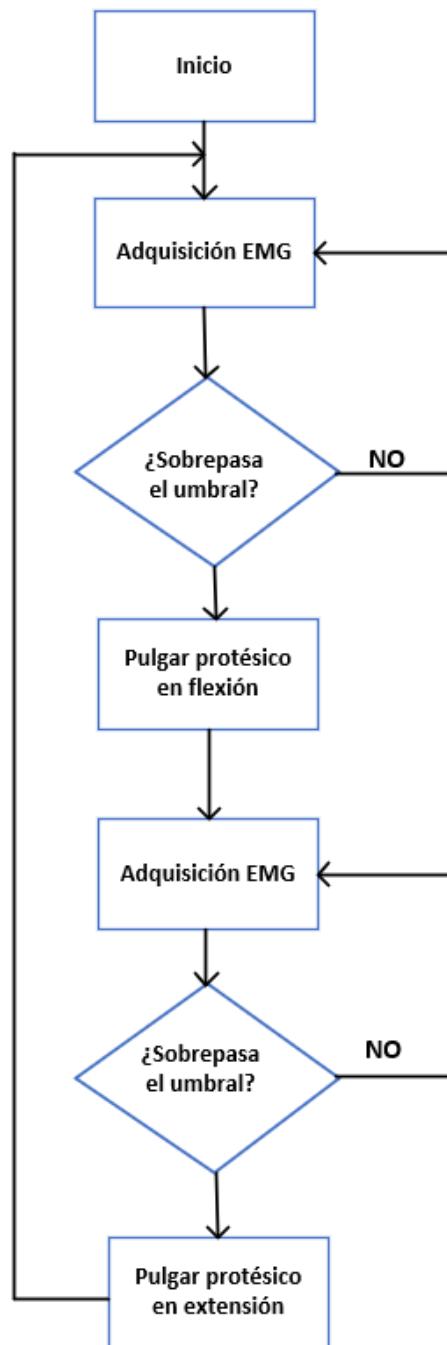
Cuando se inicia el movimiento de flexión en el pulgar, se produce una extensión de los músculos que se encuentran en la muñeca en contacto con los electrodos del sensor MyoWare. Esta extensión se traduce en una diferencia de potencial que es recogida por el sensor MyoWare y enviado a la Epos4 a través de su entrada analógica. Por otro lado, el sensor MyoWare dispone de un potenciómetro para ajustar la ganancia de la señal EMG que entrega a su salida. En esta prueba se ha hecho uso de ese potenciómetro hasta conseguir que el sensor entregue una señal con un valor en un rango de 0 y 2V. Este proceso de cuando el pulgar pasa de un estado de reposo a un estado de flexión lo podemos apreciar en la **Figura 97**.



*Figura 97. Valores leídos en la entrada analógica de la Epos4.*

Para esta primera prueba, se usará el valor de 1,5V como valor umbral a sobrepasar para posicionar al dedo protésico desde un estado inicial a un estado de flexión. Y posteriormente desde ese estado de flexión pasar a extensión.

El diagrama de flujo del funcionamiento general, se muestra en la siguiente figura.



*Figura 98. Diagrama de flujo representativo del sistema de control mioeléctrico*

Para la implementación de la **Figura 98** en lenguaje C++ se ha utilizado un bucle condicional DoWhile (). El algoritmo de control completo, escrito en C++ estará disponible en el apartado de Anexos.

Teniendo todos los elementos conectados. Al compilar y ejecutar el código se ha obtenido lo mostrado en la **Figura 99**.



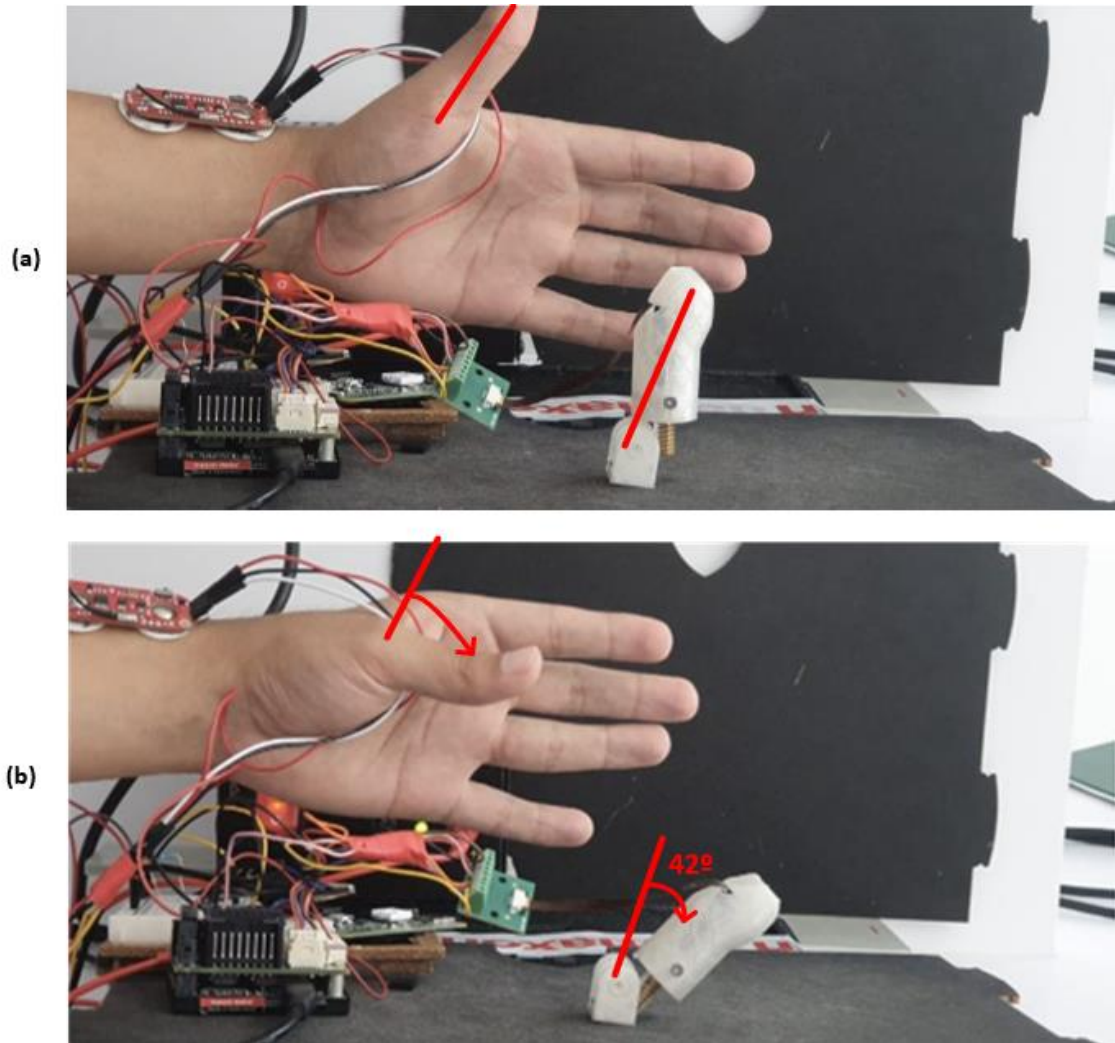


Figura 99. Dedo protésico en posición de reposo (a) y en flexión (b).

Como se puede observar en la figura anterior, el sistema de control mioeléctrico implementado en la raspberry Pi funciona de manera correcta.

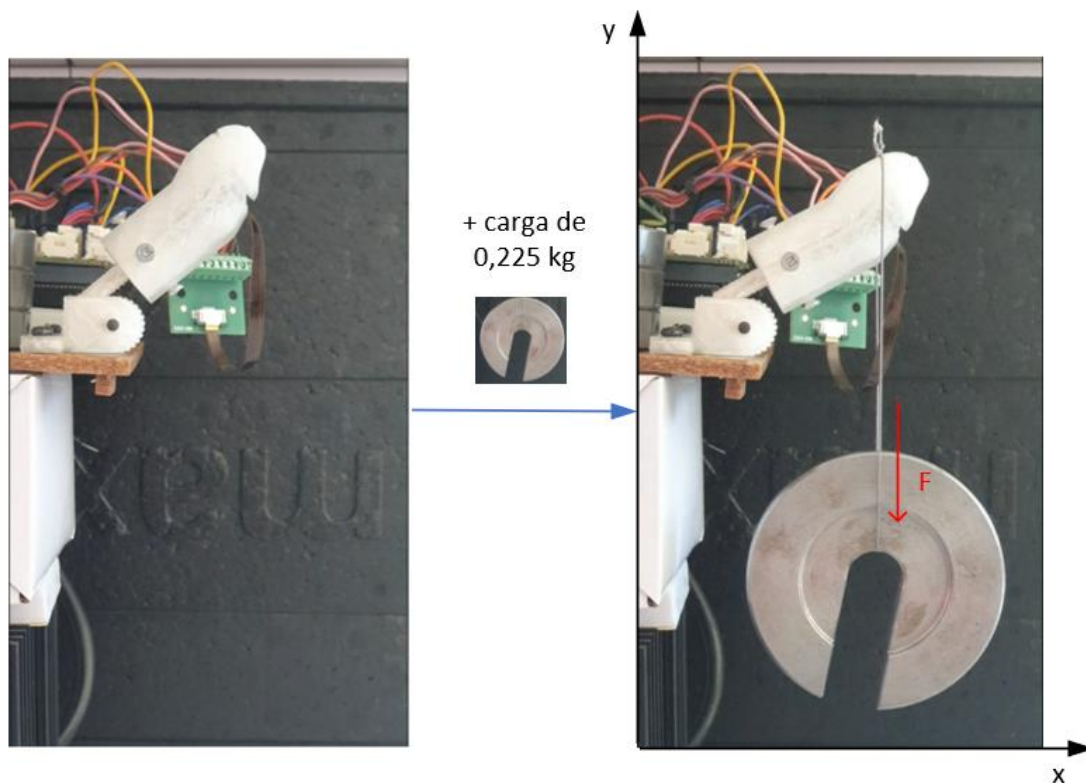
En resumen, al realizar un gesto de flexión en el dedo pulgar sano, el valor enviado por el sensor muscular a la Epos4 sobrepasará un valor umbral. Al sobrepasarse este valor umbral, la Raspberry Pi enviará las órdenes a la Epos4 para que este haga girar el eje del motor hasta que posicione al dedo pulgar protésico en flexión o en extensión.



### 5.2.2 Prueba 2: Control de posición ante una carga externa

Para realizar esta prueba, lo que se ha hecho es modificar un poco el diseño de la base del dedo protésico, para dotarle de mayor estabilidad y colocarlo en una posición acorde al ensayo que se quiere realizar.

El ensayo consistirá en posicionar el dedo protésico en flexión y posteriormente colgarle un disco de 0,225 kg y comprobar si el dedo es capaz de mantener su posición (**Figura 100**).



*Figura 100. Dedo protésico sometido a una carga de 0,225kg.*

Se hará uso del software EposStudio para graficar la evolución de tres variables:

- **Intensidad demandada:** Es la intensidad que demanda el lazo de control para mantener el dedo en una posición de referencia ante perturbaciones.
- **Intensidad actual:** Es la intensidad actual que está suministrando la Epos4 al motor.
- **Valor medio actual del torque:** Es el par resistente que tenemos en el motor.

Al añadir la carga de 0,225kg se observa que el dedo mantiene su posición y las tres variables descritas anteriormente siguen la evolución de la **Figura 101**.

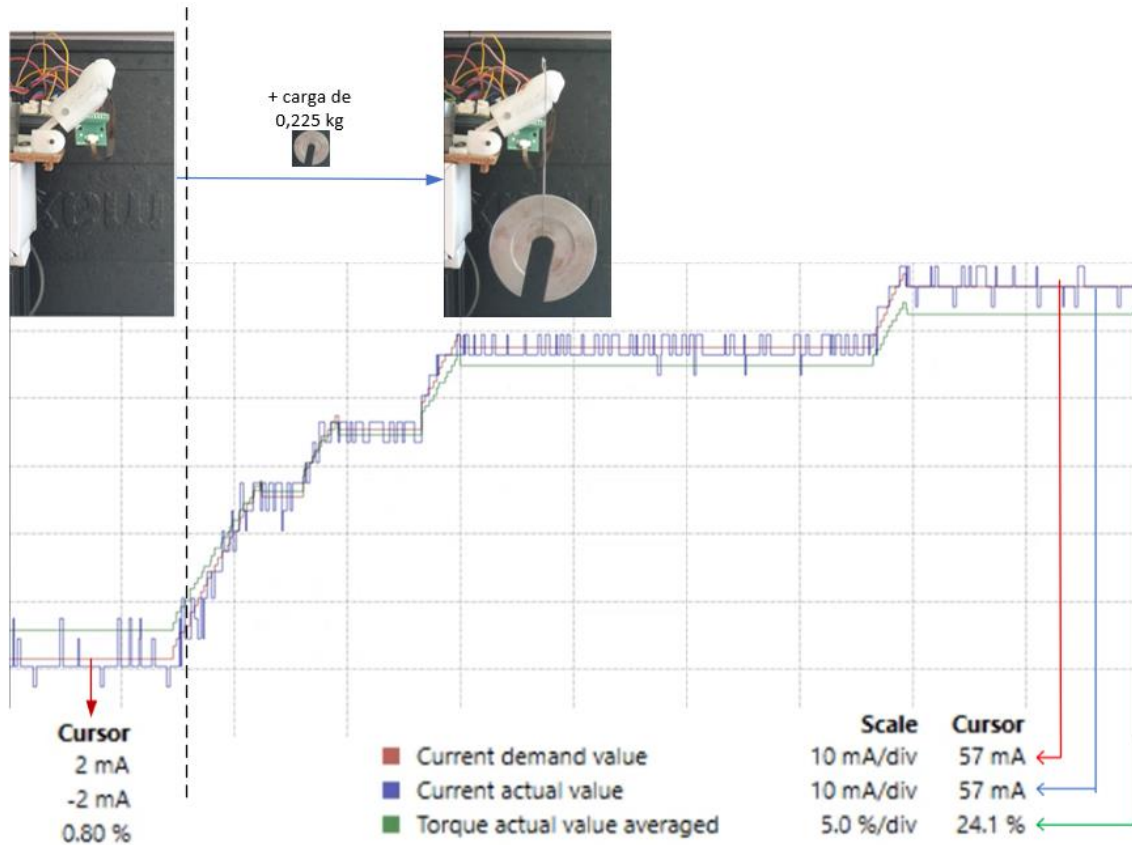


Figura 101. Evolución de la intensidad demandada, actual y el torque en el tiempo.

Como se puede observar en la figura anterior, el dedo mantiene su posición ante una perturbación externa, como puede ser el añadir una carga de 0,225kg. Para lograrlo el regulador de posición demanda una intensidad de 57mA, que corresponde con un par resistente del 24,1% del par nominal. Al quitar la carga, estas variables volverían a un valor cercano a 0.

Por otro lado, estos resultados eran lo esperado, ya que, en el estudio dinámico realizado en un capítulo anterior, se comprobó que el motor es capaz de soportar hasta una masa de 0,351kg en el tornillo sin fin.

De esta forma comprobamos, que el regulador PID de posición de la Epos4 esta sintonizado correctamente y reacciona de manera óptima ante perturbaciones.

# CAPÍTULO 6.- CONCLUSIONES Y APLICACIONES FUTURAS

## 6.1 Conclusiones

En el presente trabajo se ha realizado el diseño, fabricación, montaje y programación de un dedo pulgar protésico accionado por control mioeléctrico para amputaciones parciales al nivel de la articulación metacarpofalángica.

Una de las motivaciones que me ha llevado a realizar este proyecto se basa en mi interés por los dispositivos mecatrónicos, ya que concentran todas las capacidades que debe tener un ingeniero: análisis de un problema, búsqueda de soluciones, realización de un prototipo y presentación de resultados.

Una vez completado el desarrollo del dedo pulgar protésico y probado el sistema de control mioeléctrico, se han llegado a las siguientes conclusiones concernientes a los dispositivos elegidos y la implementación del sistema:

- El diseño mecánico de este primer prototipo es lo suficientemente funcional para realizar pruebas e implementar distintos tipos de algoritmos de control.
- La integración de los componentes elegidos en el diseño electrónico ha sido efectiva y suficiente para implementar un control preciso.
- Se ha conseguido una correcta integración del hardware junto con el software.
- El sensor muscular MyoWare es un excelente paquete de captura de datos para aplicaciones biónicas, suficiente para controlar un único dedo. Así se ha demostrado en las pruebas expuestas.
- El valor leído por el sensor EMG varía de unos sujetos a otros. De hecho, ocurre lo mismo para un mismo sujeto en días distintos, esto puede ser debido a las condiciones ambientales e incluso a la mayor o menor capacidad del paciente de generar impulsos. Por ello es necesario ajustar el valor umbral usado en el algoritmo de control mioeléctrico en cada prueba.
- La posición de los electrodos en la zona donde están los músculos del abductor largo y el extensor corto del pulgar se ha demostrado muy adecuada, ya que se ha conseguido detectar la flexión del pulgar en la mayoría de los casos.
- Con el control mioeléctrico implementado se detecta con exactitud el momento en el que el dedo pulgar protésico empieza su recorrido. Sin embargo, debido a las holguras que presentan las piezas impresas en 3D, este posicionamiento no llega a ser del todo preciso.

El prototipo actual permite su uso en entornos de prueba, aunque requiere de un estudio de viabilidad en grupos de sujetos con amputaciones al nivel de la articulación metacarpofalángica.

Para concluir y en resumen, todo el trabajo desarrollado en este TFG ha conseguido los objetivos que se plantearon al inicio.

## 6.2 Líneas futuras de trabajo

Mencionadas estas conclusiones, cabe destacar que el objetivo de este trabajo fin de grado es académico, aunque con la pretensión de generar un trabajo sólido que sirva como punto de partida para futuros desarrollos, ya que el sistema de control mioeléctrico implementado servirán como base para el desarrollo de diferentes algoritmos de control, más eficientes y precisos que el usado en este trabajo.

Teniendo esto en cuenta, quedan asentadas las bases para futuros desarrollos en las siguientes líneas generales:

- Diseños mecánicos: uso de materiales más resistentes y robustos que el PLA, e investigar en profundidad el uso de sistemas de engranajes para ejes de motores pequeños.
- Sensorización: investigar sobre los sensores electromiográficos que usan electrodos secos. Por otro lado, además de los sensores musculares usar sensores hápticos que capten la fuerza ejercida en el dedo protésico, para regular la fuerza de actuación según la misma.
- Espacio: Para reducir el espacio que ocupan los componentes, usar la Epos micro en lugar de la Epos4 Compact.
- Alimentación: investigar sobre fuentes de alimentación reguladas, capaces de alimentar tanto a la controladora de motores como a la raspberry Pi.
- Protocolo de comunicación: Usar el protocolo CAN BUS en lugar del USB entre la raspberry y la Epos. Esto mejorará notablemente la velocidad de transmisión de las instrucciones. Por otro lado, el protocolo CAN BUS permite conectar varias Epos entre ellas a través de este puerto. Esto es útil si se quiere controlar varios motores con una única Epos.
- Resto de dedos: Ampliar el diseño de este primer prototipo hasta conseguir una mano completa. Basarse en los diseños expuestos en el capítulo del estado del arte. Los cuales son diseños de código abierto.

## BIBLIOGRAFÍA

- [1] G. Mattiassich, F. Rittenschober, L. Dorninger, J. Rois, R. Mittermayr, R. Ortmaier, M. Ponschab, K. Katzensteiner, & Larcher, «Long-term outcome following upper extremity replantation after major traumatic amputation,» *BMC Musculoskeletal Disord*, vol. 9, nº 4, p. 80, 2020.
- [2] P. Alvia, G. Bravo, M.P Bustos, G Moreno, R Alfaro, R Cancino, JC Zagal, «Quantitative functional evaluation of a 3D-printed silicone-embedded prosthesis for partial hand amputation: A case report,» *Journal of Hand Therapy*, nº 31, pp. 129-136, 2018.
- [3] I. Imbinto et al., «Treatment of the partial hand amputation: An engineering perspective,» *IEEE rev. Biomed Eng.*, vol. 9, pp. 32-48, 2016.
- [4] H. Kawasaki, T. Komatsu, and K. Uchiyama, «Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand,» *IEEE/ASME Trans. Mechatronics*, vol. 7, nº 3, pp. 296-303, 2002.
- [5] S. A. Dalley, T. E. Wiste, T. J. Withrow, and M. Goldfarb, «Design of a multifunctional anthropomorphic prosthetic hand with extrinsic actuation,» *IEEE/ASME Trans. Mechatronics*, vol. 14, nº 6, pp. 699-706, 2009.
- [6] J. T. Belter and A. M. Dollar, «Performance characteristics of anthropomorphic prosthetic hands,» *Rehabilitation Robotics (ICORR)*, pp. 1-7, 2011.
- [7] J. L. Loaiza and N. Arzola, «Evolución y tendencias en el desarrollo de prótesis de mano,» nº 169, pp. 191-200, 2011.
- [8] J.M. Dorador, «Robótica y prótesis inteligentes,» *Revista Digital Universitaria*, vol. 6, nº 1, pp. 1067-6079, 2004.
- [9] J. L. Brito, M. X. Quinde, D. Cusco y J. I. Calle, «Estudio del estado del arte de las prótesis de mano,» *INGENIUS*, nº 9, 2013.
- [10] I. Sánchez, «Prótesis biónicas, biología y tecnología,» *Panorama Actual del Medicamento*, nº 42, pp. 256-259, 2018.
- [11] S. Dudley, «Historial Aspects of Powered Limb Prostheses,» *Childress*, pp. 5-6, 1985.
- [12] C. A. S. Castellanos, J. E. Muñoz, D. A. Garzón, N. S. Landinez, and O. Silva, «Mechanical and cosmetic design of a hand partial prosthesis,» *Revista Cubana de Investigaciones Biomédicas*, vol. 1, nº 30, pp. 15-41, 2011.

- [13] D. Didrick, «Articulated Artificial Finger Assembly». US Patente 6908489 B2, 21 Junio 2005.
- [14] «Ossur,» i-Digits, [En línea]. Available: <https://www.ossur.com/es-es/recursos-profesionales/catalogo-protésica-y-otros-recursos>. [Último acceso: 06 Julio 2021].
- [15] «VINCENT SYSTEMS,» [En línea]. Available: <https://www.vincentystems.de/partial-active3?lang=en>. [Último acceso: 06 Julio 2021].
- [16] «OrtoSur,» Dedos Bionicos Vincent, [En línea]. Available: <https://www.ortosur.es/catalogo-de-productos/protesis/miembro-superior/dedos-bionicos/dedos-bionicos/>. [Último acceso: 06 Julio 2021].
- [17] N.E. Krausz, R. Rorrer, R. Weir, «Design and Fabrication of a Six Degree-of-Freedom Open Source Hand,» *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, vol. 24, nº 5, pp. 562-571, 2016.
- [18] «MDdirect,» [En línea]. Available: <https://mddirect.org/tag/finger-anatomy/>. [Último acceso: 07 Julio 2021].
- [19] «unizar,» [En línea]. Available: [http://wzar.unizar.es/acad/cinesio/Documentos/Pulgar\\_Apuntes\\_2010.pdf](http://wzar.unizar.es/acad/cinesio/Documentos/Pulgar_Apuntes_2010.pdf). [Último acceso: 07 Julio 2021].
- [20] P. Jonsson, PW. Joshnson, M. Hagberg, «Accuracy and feasibility of using an electrogoniometer for measuring simple thumb movements,» *Ergonomics*, vol. 5, nº 50, pp. 657-659, 2007.
- [21] O. B. Gutiérrez, A. Almagia, P. Lizana, E. Olave, «Aspectros biometricos de la mano de individuos chilenos,» *Int J Morphol*, vol. 30, nº 2, pp. 599-606, 2012.
- [22] R F .ff. Weir, «DESIGN OF ARTIFICIAL ARMS AND HAND FOR PROSTHETIC APPLICATIONS,» *STANDARD HANDBOOK OF BIOMEDICAL ENGINEERING AND DESIGN*, vol. 32, pp. 32.1-32.61, 2003.
- [23] « Agencia Española de Medicamentos y Productos Sanitarios. <<Legislación sobre Productos Sanitarios >>,» [En línea]. Available: <https://www.aemps.gob.es/productos-sanitarios/prodsanitarios/legislacion-sobre-productos-sanitarios/>. [Último acceso: 11 Febrero 2020].
- [24] «Certificación de sistemas de gestión y requisitos de la norma,» [En línea]. Available: <https://www.enac.es/curso/iso17021-requisitos-norma>. [Último acceso: 09 Julio 2021].



- [25] [En línea]. Available: [https://www.maxongroup.es/maxon/view/content/catalog\\_request](https://www.maxongroup.es/maxon/view/content/catalog_request). [Último acceso: 14 Julio 2021].
- [26] «Environmental Engineering,» [En línea]. Available: <https://www.environmentalengineering.org.uk/news/iron-core-vs-ironless-dc-motors-4819/>. [Último acceso: 11 Julio 2021].
- [27] «Maxon Group,» [En línea]. Available: <https://www.maxongroup.es/maxon/view/product/334910>. [Último acceso: 11 Julio 2021].
- [28] «Maxon Group,» [En línea]. Available: <https://www.maxongroup.es/maxon/view/product/474124>. [Último acceso: 11 Julio 2021].
- [29] «Maxon Group Hardware Reference,» [En línea]. Available: [https://www.maxongroup.com/medias/sys\\_master/root/8884070449182/EPOS4-Module-Compact-24-1-5-Hardware-Reference-En.pdf](https://www.maxongroup.com/medias/sys_master/root/8884070449182/EPOS4-Module-Compact-24-1-5-Hardware-Reference-En.pdf). [Último acceso: 07 Julio 2021].
- [30] «Maxon Grop España,» [En línea]. Available: <https://www.maxongroup.es/maxon/view/product/546714>. [Último acceso: 07 Julio 2021].
- [31] «<https://www.raspberrypi.org/products/raspberry-pi-zero/>,» Raspberry Pi, [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-zero/>. [Último acceso: 14 Julio 2021].
- [32] [En línea]. Available: <https://www.rjconcepcion.com/podcast/conoce-todo-sobre-la-raspberry-pi-zero-w/>. [Último acceso: 14 Julio 2021].
- [33] S. Franco, «Ingeniería Biomédica perspectivas desde el Uruguay,» *Núcleo de Ingeniería Biomédica*, p. 448, 2007.
- [34] «jralonso,» [En línea]. Available: <https://jralonso.es/2015/12/01/el-axon-gigante-del-calamar/>. [Último acceso: 30 08 2021].
- [35] «upcommons.upc,» [En línea]. Available: <https://upcommons.upc.edu/bitstream/handle/2099.1/6277/4.NEURONA.pdf?sequence=6&isAllowed=y>. [Último acceso: 30 08 2021].
- [36] K. Anam, «Bio-driven control system for the rehabilitation hand device : a new approach,» *Open Publication of UTS Scholars*, nº 21, p. 233, 2016.

- [37] «Chegg,» [En línea]. Available: <https://www.chegg.com/flashcards/chapter-11-muscular-tissue-dafe72c1-3df6-4284-aba9-26e7d1a60eab/deck>. [Último acceso: 30 08 2021].
- [38] D. Gutiérrez, «Diseño del sistema de control de un dispositivo mecatrónico de rehabilitación de mano,» 2017. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/38/browse?authority=2765c5b0-81f0-4c43-8216-26a10e608665&type=author>. [Último acceso: 02 09 2017].
- [39] L.F. Marín, E.I. Vera, «Repositorios espe,» [En línea]. Available: <http://repositorio.espe.edu.ec/bitstream/21000/13219/1/T-ESPEL-MEC-0096.pdf>. [Último acceso: 02 09 2021].
- [40] «sparkfun,» [En línea]. Available: <https://www.sparkfun.com/products/14409>. [Último acceso: 02 09 2021].
- [41] Backyard brains, «Backyardbrains,» [En línea]. Available: <https://backyardbrains.com/experiments/RobotHand>. [Último acceso: 04 09 2021].
- [42] «molex,» [En línea]. Available: [https://www.molex.com/molex/search/partSearch?query=52745%20-%200897&pQuery=.](https://www.molex.com/molex/search/partSearch?query=52745%20-%200897&pQuery=) [Último acceso: 05 09 2021].
- [43] H. Rouviere and A. Delmas, Anatomía Humana, Barcelona: Masson, 2007.
- [44] «Stanford Children's Health,» [En línea]. Available: <https://www.stanfordchildrens.org/es/topic/default?id=anatomyofthehand-85-P04195>. [Último acceso: 07 Julio 2021].
- [45] «Maxon Group,» [En línea]. Available: <https://www.maxongroup.es/maxon/view/product/347724>. [Último acceso: 07 Julio 2021].
- [46] «Mootio Components,» [En línea]. Available: [http://www.mootio-components.com/\\_ref\\_016304.html#.YOWhaugzYmL](http://www.mootio-components.com/_ref_016304.html#.YOWhaugzYmL). [Último acceso: 12 Julio 2021].
- [47] «Instrumentacion Biomedica,» [En línea]. Available: <https://sites.google.com/site/biomedicaeinstrumentacion/biopotenciales/biosenales/biosenales2?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>. [Último acceso: 29 08 2021].
- [48] R.R. Seeley, T.D. Stephens, P. Tate, Anatomy and Physiology, McGraw Hill, 2008.

- [49] M. Flores, J. Mojica, «pardell,» [En línea]. Available: <https://www.pardell.es/electromiografo.html>. [Último acceso: 02 09 2021].

# ANEXO I: código

## Código generado en lenguaje C++

```
1 //Epos Print
2 #include <iostream> //std::cout
3 //#include "Definitions.h"
4 #include <string.h> //std::string
5 #include <sstream> //std::stringstream
6 using namespace std;
7 #include "Definitions.h"
8
9 //Librerias Extra para Position
10 #include <getopt.h>
11 #include <stdlib.h>
12 #include <stdio.h>
13 #include <math.h>
14 #include <sys/types.h>
15 #include <unistd.h>
16 #include <sys/times.h>
17 #include <sys/time.h>
18
19 //Parametros de comunicacion
20 unsigned short g_usNodeId = 1;
21 string g_deviceName;
22 string g_protocolStackName;
23 string g_interfaceName;
24 string g_portName;
25 int g_baudrate = 0;
26
27 //Variable Control Mano cerrada
28 int cerrado = 0;
29
30 //A mayores respecto a 7.1.EposPrint. Asigno la direccion 0
31 void* g_pKeyHandle = 0;
32 //Es igual que poner void* g_pKeyHandle; g_pKeyHandle = 0
33 #ifndef MMC_SUCCESS
34 | #define MMC_SUCCESS 0
35 #endif
36
37 #ifndef MMC_FAILED
38 | #define MMC_FAILED 1
39 #endif
40
```

```

41 #ifndef MMC_MAX_LOG_MSG_SIZE
42 | #define MMC_MAX_LOG_MSG_SIZE 512
43 #endif
44 //Defino variable tipo Handle
45 typedef void* HANDLE;
46 typedef int BOOL;
47 void PrintHeader();
48 void PrintSettings();
49 void SeparatorLine();
50 void SetDefaultParameters();
51 void LogInfo(string message);
52 int OpenDevice(unsigned int* p_pErrorCode);
53 int PrepareDemo(unsigned int* p_pErrorCode);
54 int AnalogInput(HANDLE p_DeviceHandle, unsigned short p_usNodeId, unsigned int *p_r1ErrorCode);
55 int PrepareDemo(unsigned int* p_pErrorCode);
56 int DemoProfilePositionMode(HANDLE p_DeviceHandle, unsigned short p_usNodeId, unsigned int & p_r1ErrorCode);
57
58
59 void PrintHeader()
60 {
61 | SeparatorLine();
62
63 | LogInfo("Programa Ejemplo");
64
65 | SeparatorLine();
66 }
67
68 void PrintSettings()
69 {
70 | stringstream msg;
71 | msg << "Configuración por defecto: " << endl;
72 | msg << "node id = " << g_usNodeId << endl;
73 | msg << "device name = " << g_deviceName << "" << endl;
74 | msg << "protocol stack name = " << g_protocolStackName << "" << endl;
75 | msg << "interface name = " << g_interfaceName << "" << endl;
76 | msg << "port name = " << g_portName << "" << endl;
77 | msg << "baudrate = " << g_baudrate << "" << endl;
78 | LogInfo(msg.str());
79 | SeparatorLine();
80 | cout << "node id = " << g_usNodeId<<endl;
81 | cout << "device name = " << g_portName<<endl;
82 | SeparatorLine();
83 }
84 void LogInfo(string message)
85 {
86 | cout << message << endl;
87 }
88 void SetDefaultParameters()
89 {
90 | //USB
91 | g_usNodeId = 1;
92 | g_deviceName = "EPOS4";
93 | g_protocolStackName = "MAXON SERIAL V2";
94 | g_interfaceName = "USB";
95 | g_portName = "USB0";
96 | g_baudrate = 1000000; //tasa de baudios. Velocidad de transmisión.

```

```

97 void SeparatorLine()
98 {
99     const int lineLength = 65;
100    for(int i=0; i<lineLength; i++)
101    {
102        cout<<"-";
103    }
104    cout <<endl;
105 }
106 int OpenDevice(unsigned int *p_pErrorCode)
107 {
108     int lResult = MMC_FAILED;
109     //char pDeviceName [255]= "hasta una longitud de 255".. o usar char*
110     char *pDeviceName = new char[255];
111     char *pProtocolStackName = new char[255];
112     char *pInterfaceName = new char[255];
113     char *pPortName = new char[255];
114
115     strcpy(pDeviceName, g_deviceName.c_str());
116     strcpy(pProtocolStackName, g_protocolStackName.c_str());
117     strcpy(pInterfaceName, g_interfaceName.c_str());
118     strcpy(pPortName, g_portName.c_str());
119
120     cout<<"VCS_OpenDevice abre el puerto para enviar y recibir comandos"<<"\n";
121     cout<<"Dispositivo Abriéndose empezando comunicación con EPOS"<<"\n";
122     cout<<"VCS_OpenDevice va a devolver una direccion de memoria "<<"\n";
123     // *g_pKeyHandle = 100 No se puede porque es un puntero void
124     // Asignas direcciones de memoria da igual el tipo de dato
125     g_pKeyHandle = VCS_OpenDevice(pDeviceName, pProtocolStackName, pInterfaceName, pPortName, p_pErrorCode);

126     cout<<"La direccion de g_pKeyHandle paso de 0 ---> "<<g_pKeyHandle<<endl;
127     if(g_pKeyHandle!=0 && *p_pErrorCode ==0 )
128     {
129         unsigned int lBaudrate = 0; //Tasa de baudios real del puerto abierto [bit / s]
130         unsigned int lTimeout = 0; // Tiempo de espera real del puerto abierto [ms]
131         //obtener la configuración de la pila de protocolos
132         if(VCS_GetProtocolStackSettings(g_pKeyHandle, &lBaudrate, &lTimeout, p_pErrorCode)!=0)
133         {
134             cout<<"Estamos dentro de VCS_GetProcol esta funcion OBTIENE los parametros a tiempo real de la comunicacion"<<endl;
135             cout<<"lBaudrate -->Tasa de baudios real del puerto abierto es : "<<lBaudrate<<" bit/s "<<endl;
136             cout<<"lTimeout -->Tiempo de espera real del puerto abierto es : "<<lTimeout<<" ms "<<endl;
137             //configurar- Poner la configuración de la pila de protocolos
138             if(VCS_SetProtocolStackSettings(g_pKeyHandle, g_baudrate, lTimeout, p_pErrorCode)!=0)
139             {
140                 cout<<"Estamos dentro de VCS_Set protocol, colocando, poniendo los parametros reales de la Epos"<<endl;
141                 cout<<"Verificando la tasa de baudios que quiero comunicar con la tasa real ..."<<endl;
142                 if(VCS_GetProtocolStackSettings(g_pKeyHandle, &lBaudrate, &lTimeout, p_pErrorCode)!=0)
143                 {
144                     if(g_baudrate==(int)lBaudrate) // 1000000 = 1000000
145                     {
146                         cout<<"La tasa de baudios que coloque coincide con la tasa de baudios real"<<endl;
147                         cout<<"La comunicación se ha logrado correctamente"<<endl;
148                         lResult = MMC_SUCCESS; // = 0 y lo retorno
149                     }
150                 }
151             }
152         }
153     }
154     else
155     {
156         g_pKeyHandle = 0;
157     }
158
159     delete []pDeviceName;
160     delete []pProtocolStackName;
161     delete []pInterfaceName;
162     delete []pPortName;
163     return lResult;
164 }

```



```

165
166 int AnalogInput(HANDLE p_DeviceHandle, unsigned short p_usNodeId, unsigned int * p_r1ErrorCode)
167 {
168     int lResult=MMC_SUCCESS;
169     unsigned int lErrorCode = 0;
170     unsigned short InputNumber=1; //WORD InputNumber, Corresponde con A1+
171     int long* pVoltageValue; //WORD* pAnalogValue, No hace falta declarar puntero
172     do
173     {
174         if(VCS_GetAnalogInputVoltage(p_DeviceHandle, p_usNodeId, InputNumber, pVoltageValue, p_r1ErrorCode) ==0)
175         {
176             cout<<"Error en VCS_GetAnalogInputVoltage"<<endl;
177             lResult = MMC_FAILED;
178         }
179
180         cout<<"El valor de pVoltageValue es : "<<*pVoltageValue<<endl;
181
182         if((*pVoltageValue > 1500) && (cerrado == 0))
183         {
184             cout<<"Dedo pulgar en flexión"<<endl;
185             lResult = DemoProfilePositionMode(g_pKeyHandle, g_usNodeId, lErrorCode);
186         }
187         if(VCS_GetAnalogInputVoltage(p_DeviceHandle, p_usNodeId, InputNumber, pVoltageValue, p_r1ErrorCode) ==0)
188         {
189             cout<<"Error en VCS_GetAnalogInputVoltage"<<endl;
190             lResult = MMC_FAILED;
191         }
192         if((*pVoltageValue > 1500) && (cerrado == 1))
193         {
194             cout<<"Dedo pulgar en extensión"<<endl;
195             lResult = DemoProfilePositionMode(g_pKeyHandle, g_usNodeId, lErrorCode);
196             cerrado = cerrado +1 ;
197         }
198         sleep(1);
199     } while(cerrado < 2);
200     return lResult;
201 }
202
203 int PrepareDemo(unsigned int* p_pErrorCode)
204 {
205     int lResult = MMC_SUCCESS;
206     BOOL oIsFault = 0;
207
208     if(VCS_GetFaultState(g_pKeyHandle, g_usNodeId, &oIsFault, p_pErrorCode ) == 0)
209     {
210         lResult = MMC_FAILED;
211     }
212
213     if(lResult==0)
214     {
215         if(oIsFault)
216         {
217             stringstream msg;
218             msg << "clear fault, node = " << g_usNodeId << " ";
219             LogInfo(msg.str());
220
221             if(VCS_ClearFault(g_pKeyHandle, g_usNodeId, p_pErrorCode) == 0)
222             {
223                 lResult = MMC_FAILED;
224             }
225         }
226
227         if(lResult==0)
228         {
229             BOOL oIsEnabled = 0;
230             //Partes de la no habilitacion. Internamente Verificado si esta en 0 o 1.
231             //Si coincide y esta en 0 (No habilitado) te lanza un 1. No entra al bucle.
232             //Y en la siguiente instrucción te habilita el dispositivo Epos.
233             if(VCS_GetEnableState(g_pKeyHandle, g_usNodeId, &oIsEnabled, p_pErrorCode) == 0)
234             {
235                 lResult = MMC_FAILED;
236             }
237             //Si esta deshabilitado lo habilitas. Si no esta deshabilitado no hace nada.
238             if(lResult==0)

```

```

239     {
240         if(!IsEnabled)
241         {
242             if(VCS_SetEnableState(g_pKeyHandle, g_usNodeId, p_pErrorCode) == 0)
243             {
244                 {
245                     lResult = MMC_FAILED;
246                 }
247             }
248         }
249     }
250     return lResult;
251 }
252
253 int DemoProfilePositionMode(HANDLE p_DeviceHandle, unsigned short p_usNodeId, unsigned int & p_rlErrorCode)
254 {
255     SeparatorLine();
256     int lResult = MMC_SUCCESS;
257     //long targetPosition = 14400; // Una unica vuelta
258     long targetPosition; //13 vueltas
259     if(cerrado == 0)
260     {
261         targetPosition = 22680;
262     }
263     cout <<"Set profile position mode, node = " <<p_usNodeId<<endl;
264     if(VCS_ActivateProfilePositionMode(p_DeviceHandle, p_usNodeId, &p_rlErrorCode) == 0)
265     {
266         cout<<"Error en el activar el perfil de posicionamiento"<<endl;
267         lResult = MMC_FAILED;
268     }
269     else
270     {
271         cout <<"move to position = " << targetPosition << ", node = " << p_usNodeId<<endl;
272         if(VCS_MoveToPosition(p_DeviceHandle, p_usNodeId, targetPosition, 0, 1, &p_rlErrorCode) == 0)
273         {
274             cout<<"Ha fallado el llamado a VCS_MoveToPosition"<<endl;
275             lResult = MMC_FAILED;
276             //break;
277         }
278         sleep(1); //
279         SeparatorLine();
280         if(lResult == MMC_SUCCESS)
281         {
282             cout<<"Iniciando la desaceleración"<<endl;
283             if(VCS_HaltPositionMovement(p_DeviceHandle, p_usNodeId, &p_rlErrorCode) == 0)
284             {
285                 cout<<"Ha fallado la desaceleración"<<endl;
286                 lResult = MMC_FAILED;
287             }
288             sleep(6);
289         }
290     }
291 }
292
293 cerrado = cerrado +1 ;
294 return lResult;
295 }
296
297 int main(int argc, char** argv)
298 {
299     int lResult = MMC_FAILED;
300     unsigned int ulErrorCode = 0;
301     SetDefaultParameters();
302     if ((lResult = OpenDevice(&ulErrorCode)) != MMC_SUCCESS)
303     {
304         cout<<"OpenDevice ha fallado"<<lResult<<ulErrorCode;
305         return lResult;
306     }
307     if(lResult = AnalogInput(g_pKeyHandle, g_usNodeId, &ulErrorCode)!=MMC_SUCCESS)
308     {
309         cout<<"AnalogInput ha fallado"<<lResult<<ulErrorCode<<endl;
310         return lResult;
311     }
312     if(lResult = 1)
313     {
314         cout<<"\n"<<"Todo fue correctamente, seguimos con lo demas del main"<<endl;
315     }
316     return lResult;
317 }
318 }

```