



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“DESARROLLO DE UN SISTEMA BIOELECTRÓNICO PARA EL
ENTRENAMIENTO EN EL USO DE UNA PRÓTESIS
MIOELÉCTRICA EN LA REHABILITACIÓN DE PERSONAS CON
AMPUTACIÓN TRANSRADIAL”.**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR: ALVARO GABRIEL CASIERRA COTERA

DIRECTOR: ING. RAMIRO FERNANDO ISA JARA

Riobamba – Ecuador

2024

© 2024, Alvaro Gabriel Casierra Cotera

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Alvaro Gabriel Casierra Cotera, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 05 de abril de 2024.

A handwritten signature in blue ink, consisting of several overlapping loops and a long horizontal stroke extending to the right.

Alvaro Gabriel Casierra Cotera

C.I: 0803706431

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Proyecto Técnico: **“DESARROLLO DE UN SISTEMA BIOELECTRÓNICO PARA EL ENTRENAMIENTO EN EL USO DE UNA PRÓTESIS MIOELÉCTRICA EN LA REHABILITACIÓN DE PERSONAS CON AMPUTACIÓN TRANSRADIAL”**, realizado por el señor: **ALVARO GABRIEL CASIERRA COTERA**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Wilson Oswaldo Baldeon Lopez PRESIDENTE DEL TRIBUNAL		2024-04-05
Dr. Ramiro Fernando Isa Jara DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2024-04-05
Ing. Diego Ramiro Ñacato Estrella ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2024-04-05

DEDICATORIA

El presente trabajo de integración curricular está dedicado a mis padres Rene y Magali por apoyarme en todo momento, guiar mi camino y amarme a pesar de mis malos momentos, a mis hermanos Jairo y Alan, sabiendo que cuento con ellos de manera incondicional. A toda mi familia que de una u otra manera me han impulsado para seguir adelante. A mis amigos cuyo apoyo y compañía ha sido fuente inagotable de motivación para alcanzar esta meta.

Alvaro

AGRADECIMIENTO

Agradezco a todos aquellos que hicieron posible la realización de este trabajo de integración curricular, por su apoyo constante y aliento. De manera especial a mis profesores, por su orientación, compromiso, paciencia y conocimientos compartidos; y a todos mis amigos que de alguna manera contribuyeron a este trabajo. A la Escuela Superior Politécnica de Chimborazo por brindarme la oportunidad de llevar a cabo este trabajo. Su apoyo y recursos proporcionados fueron fundamentales para el desarrollo y éxito de este trabajo de integración curricular.

Alvaro

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE ILUSTRACIONES.....	xi
ÍNDICE DE ANEXOS.....	xiv
RESUMEN.....	xv
SUMMARY	xvi
INTRODUCCIÓN	1

CAPÍTULO I

1. DIAGNOSTICO DEL PROBLEMA.....	2
1.1. Antecedentes	2
1.2. Formulación del Problema	5
1.3. Sistematización del Problema	5
1.4. Justificación Teórica	5
1.5. Justificación Aplicativa.....	7
1.6. Objetivos	8
1.6.1. <i>Objetivo General</i>	8
1.6.2. <i>Objetivos Específicos</i>	8

CAPÍTULO II

2. MARCO TEÓRICO	9
2.1. Discapacidad.....	9
2.1.1. <i>Persona con discapacidad</i>	9
2.1.2. <i>Discapacidad física</i>	9
2.2. Amputación.....	10
2.2.1. <i>Amputación transradial</i>	10

2.3. Tipos de prótesis de miembro superior	11
2.3.1. <i>Estéticas</i>	11
2.3.2. <i>Mecánicas</i>	12
2.3.3. <i>Eléctricas</i>	13
2.3.4. <i>Neumáticas</i>	13
2.3.5. <i>Mioeléctricas</i>	14
2.3.6. <i>Híbridas</i>	15
2.4. Tipos de agarre con prótesis.....	15
2.5. Sistemas de control de lazo cerrado.....	16
2.6. Controladores	17
2.6.1. <i>Controlador PI</i>	18
2.6.2. <i>Controlador PD</i>	19
2.6.3. <i>Controlador PID</i>	19
2.7. Señales	20
2.7.1. <i>Señales mioeléctricas</i>	21
2.8. Sensores mioeléctricos	21
2.8.1. <i>Myo Ware</i>	21
2.8.2. <i>Myo Armband</i>	22
2.8.3. <i>EMG AD8832</i>	23
2.9. Modulación por ancho de pulso - PWM.....	24
2.10. Sensor de fuerza	25
2.10.1. <i>MF01</i>	25
2.10.2. <i>RFP-602</i>	26
2.10.3. <i>Análisis comparativo de los sensores de fuerza</i>	26
2.11. Tarjeta de desarrollo.....	27
2.11.1. <i>Arduino</i>	27
2.11.2. <i>ESP32</i>	28
2.11.3. <i>Raspberry PI</i>	29
2.11.4. <i>Análisis comparativo de las tarjetas de desarrollo</i>	30

2.12. Servomotores	31
2.12.1. <i>Micro servo MG90S</i>	32
2.13. Procesamiento de señales.....	33
2.13.1. <i>Matlab</i>	33
2.13.2. <i>Python</i>	34

CAPÍTULO III

3. MARCO METODOLÓGICO.....	36
3.1. Requerimientos del sistema bioelectrónico	36
3.2. Concepción general del sistema bioelectrónico.....	36
3.3. Etapa de adquisición	38
3.3.1. <i>Comunicación entre Myo Armband y el computador</i>	39
3.3.2. <i>Comunicación entre RFP 602 y Arduino UNO</i>	40
3.4. Etapa de procesamiento.....	41
3.4.1. <i>Identificación del gesto</i>	42
3.5. Etapa de activación y visualización	43
3.5.1. <i>Algoritmo de control</i>	43
3.5.2. <i>GUI</i>	43
3.6. Elementos Hardware del sistema bioelectrónico	43
3.6.1. <i>Myo Armband</i>	43
3.6.2. <i>Arduino UNO</i>	47
3.6.3. <i>RFP 602</i>	48
3.6.4. <i>Esquema de conexión del sistema bioelectrónico</i>	48
3.7. Software para el desarrollo del sistema bioelectrónico.....	49
3.7.1. <i>Myo Connect Installer</i>	49
3.7.2. <i>Matlab</i>	50
3.7.3. <i>Arduino</i>	53
3.8. Desarrollo de la GUI.....	56
3.8.1. <i>App Designer</i>	57

CAPÍTULO IV

4.	VALIDACIÓN DEL SISTEMA BIOELECTRÓNICO	60
4.1.	Validación del tiempo de conexión del sistema.....	60
4.2.	Validación del reconocimiento de gesto con el sensor Myo Armband	63
4.3.	Validación de la efectividad de agarre de la prótesis	71
4.3.1.	<i>Características de los objetos seleccionados.....</i>	71
4.4.	Tiempo de respuesta	72
4.5.	Identificación de objeto por presión	74
4.6.	Consumo de energía del sistema bioelectrónico	77
4.7.	Evaluación económica del prototipo.....	78
4.7.1.	<i>Costo de materiales y herramientas</i>	78
	CONCLUSIONES.....	79
	RECOMENDACIONES.....	80
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 2-1: Características principales de los tipos de agarre.	16
Tabla 2-2: Características de las ganancias del controlador PID.	18
Tabla 2-3: Características de los sensores de presión.	27
Tabla 2-4: Características de tarjetas de desarrollo.....	30
Tabla 2-5: Características principales micro servo MG90S.	32
Tabla 3-1: Distribución de pines en Arduino UNO.	47
Tabla 3-2: Componentes usados en la App Designer.	58
Tabla 4-1: Tiempos de conexión del sistema bioelectrónico.	61
Tabla 4-2: Resultados Media, Mediana y Moda de los tiempos de conexión.....	62
Tabla 4-3: Resultados T Student para el tiempo de conexión del sistema bioelectrónico.	62
Tabla 4-4: Datos obtenidos del hombre en el reconocimiento de gestos.	64
Tabla 4-5: Resultados de la frecuencia de reconocimiento de los gestos en el hombre.....	65
Tabla 4-6: Resultados Q de Cochran para el reconocimiento de gestos en el hombre.	65
Tabla 4-7: Pruebas correctas e incorrectas de los gestos en el hombre.....	66
Tabla 4-8: Datos obtenidos de la mujer en el reconocimiento de gestos.	67
Tabla 4-9: Resultados de la frecuencia de reconocimiento de los gestos en la mujer.	69
Tabla 4-10: Resultados Q de Cochran para el reconocimiento de gestos en la mujer.	69
Tabla 4-11: Pruebas correctas e incorrectas de los gestos en la mujer.	69
Tabla 4-12: Dimensiones de los objetos.	71
Tabla 4-13: Pruebas correctas e incorrectas de los objetos.....	72
Tabla 4-14: Tiempos de respuesta de agarres correctos.....	72
Tabla 4-15: Resultados media, mediana y moda de tiempos de agarre.	74
Tabla 4-16: Datos de los sensores de presión RFP para la harina.....	74
Tabla 4-17: Datos de los sensores de presión RFP para la pelota 2.....	75
Tabla 4-18: Datos de los sensores de presión RFP para la botella 2.....	76
Tabla 4-19: Consumo de corriente del sistema.	77
Tabla 4-20: Costo de materiales y herramientas.	78

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1:	Diagrama de bloques del proyecto.....	7
Ilustración 1-2:	Esquema del sistema bioelectrónico.	8
Ilustración 2-1:	Discapacidad física.	9
Ilustración 2-2:	Tipos de amputación.	10
Ilustración 2-3:	Niveles de amputación de miembro superior.....	11
Ilustración 2-4:	Tipos de prótesis de miembro superior.	11
Ilustración 2-5:	Prótesis estética.	12
Ilustración 2-6:	Prótesis mecánica.....	12
Ilustración 2-7:	Prótesis eléctrica.	13
Ilustración 2-8:	Prótesis neumática.	14
Ilustración 2-9:	Prótesis mioeléctrica.	14
Ilustración 2-10:	Prótesis híbrida.	15
Ilustración 2-11:	Tipos de agarre.....	16
Ilustración 2-12:	Sistema de control en lazo cerrado.	17
Ilustración 2-13:	Señales en el tiempo.	20
Ilustración 2-14:	Señal EMG.....	21
Ilustración 2-15:	Sensor Myo Ware.	22
Ilustración 2-16:	Sensor Myo Armband.....	23
Ilustración 2-17:	Sensor AD8832.....	24
Ilustración 2-18:	Ciclo de trabajo PWM.	25
Ilustración 2-19:	Sensor MF01.....	26
Ilustración 2-20:	Sensor RFP-602.	26
Ilustración 2-21:	Tipos de Arduino.	28
Ilustración 2-22:	Tipos de ESP32.....	29
Ilustración 2-23:	Tipos de Raspberry Pi.....	30
Ilustración 2-24:	Componentes servomotores.	31
Ilustración 2-25:	Microservo MG90S.	32
Ilustración 2-26:	GUI con Matlab.	34
Ilustración 2-27:	GUI con Python.	35
Ilustración 3-1:	Diagrama de bloques del sistema.....	37
Ilustración 3-2:	Diagrama de flujo del sistema bioelectrónico.....	38
Ilustración 3-3:	Diagrama de bloques de la etapa de adquisición.	39
Ilustración 3-4:	Comunicación Myo Armband-Computador.	40

Ilustración 3-5:	Comunicación RFP 602-Arduino UNO.....	41
Ilustración 3-6:	Diagrama de bloques de la etapa de procesamiento.	41
Ilustración 3-7:	Propiedades de los datos muestreados en la base de tiempo EMG.....	42
Ilustración 3-8:	Diagrama de flujo de la identificación de gesto.....	42
Ilustración 3-9:	Diagrama de bloques de la etapa de activación y visualización.	43
Ilustración 3-10:	Partes del Brazaletes Myo Armband.	44
Ilustración 3-11:	Elementos principales del brazaletes Myo Armband.	45
Ilustración 3-12:	Elementos ubicados en la placa base del Myo Armband.....	45
Ilustración 3-13:	Posición Myo Armband arriba del codo.	46
Ilustración 3-14:	Posición Myo Armband debajo del codo.....	46
Ilustración 3-15:	Sensores RFP 602 en Prótesis.....	48
Ilustración 3-16:	Esquema de conexión del sistema bioelectrónico.....	49
Ilustración 3-17:	Ventana principal de Myo Connect.	50
Ilustración 3-18:	Ubicación de MyoMex-master y myo-sdk-win.	51
Ilustración 3-19:	Archivos ejecutables en el directorio de trabajo actual.....	52
Ilustración 3-20:	Compilador MinGW-64 instalado.	52
Ilustración 3-21:	Librería para Arduino instalada.	53
Ilustración 3-22:	Código para iniciar la comunicación serial.....	53
Ilustración 3-23:	Diagrama de flujo Arduino.....	55
Ilustración 3-24:	Secuencia de funcionamiento de la GUI.....	57
Ilustración 3-25:	Panel de Inicio de la GUI.....	59
Ilustración 3-26:	Panel de gráficas de la GUI.	59
Ilustración 4-1:	Toma de datos del tiempo de conexión del sistema.....	60
Ilustración 4-2:	Tiempos de conexión del sistema bioelectrónico.	63
Ilustración 4-3:	Toma de datos del reconocimiento de gestos.....	64
Ilustración 4-4:	Diagrama de barras para el reconocimiento de gestos en el hombre.	67
Ilustración 4-5:	Efectividad de reconocimiento en el hombre.....	67
Ilustración 4-6:	Diagrama de barras para el reconocimiento de gestos en la mujer.	70
Ilustración 4-7:	Efectividad de reconocimiento en la mujer.	70
Ilustración 4-8:	Efectividad de agarre.	72
Ilustración 4-9:	Tiempos de respuesta de agarres correctos.	73
Ilustración 4-10:	Datos de los sensores de presión RFP para la harina.....	75
Ilustración 4-11:	Datos de los sensores de presión RFP para la pelota 2.	76
Ilustración 4-12:	Datos de los sensores de presión RFP para la botella 2.....	77

ÍNDICE DE ANEXOS

ANEXO A: PRIMEROS PASOS CON EL SENSOR MYO ARMBAND

ANEXO B: HOJA DE DATOS DEL ARDUINO UNO

ANEXO C: DATOS DEL SENSOR RFP 602

ANEXO D: HOJA DE DATOS DEL SERVOMOTOR MG90S

ANEXO E: DISEÑO DE LA INTERFAZ GRÁFICA DE USUARIO GUI

ANEXO F: PROGRAMACIÓN DEL SISTEMA DE CONTROL PID

ANEXO G: SISTEMA BIOELECTRÓNICO FINAL

ANEXO H: MANUAL DE USUARIO

RESUMEN

Adaptarse a una prótesis puede resultar difícil porque estas partes son extrañas al cuerpo y el proceso lleva tiempo para la mente y especialmente para el cuerpo. Los sistemas electrónicos de rehabilitación hoy en día han surgido como una solución o apoyo a la rehabilitación tradicional. En el presente trabajo se desarrolla un sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioelectrónica en la rehabilitación de personas con amputación transradial. El sistema consta de tres bloques: adquisición, procesamiento y activación. Durante la etapa de adquisición se obtienen las señales mioeléctricas en tiempo real mediante el sensor Myo Armband para ser enviadas al computador a través de comunicación inalámbrica Bluetooth para identificar los gestos. Además, se obtienen las señales de los sensores de presión RFP 602 mediante la tarjeta de desarrollo Arduino UNO. Con estas señales se realiza el procesamiento aplicando la técnica de control PID para la activación de los actuadores en función del gesto recibido y realizar el agarre de objetos de diferente composición y tamaños. La comunicación entre los elementos sensores y actuadores, el reconocimiento del gesto, así como la visualización de las señales de los sensores se la realiza a través de una interfaz gráfica de usuario (GUI) programada en Matlab. De las pruebas de validación de la conexión entre el sensor y la interfaz (GUI) se obtiene un tiempo promedio de 9 segundos, lo que permite verificar que se tiene una conexión exitosa. A partir de las pruebas de agarre y de tiempo de respuesta con la prótesis se obtiene un 84% de efectividad con un tiempo promedio de 2.93 segundos. Esto indica que el sistema cumple con la funcionalidad esperada. De esto se concluye que el sistema bioelectrónico cumple con los requerimientos previamente establecidos.

Palabras clave: <PROTOTIPO BIOELECTRÓNICO>, <PRÓTESIS MIOELÉCTRICA>, <CONTROL AUTOMÁTICO>, <INTERFÁZ GRÁFICA DE USUARIO (GUI)>, <PROGRAMACIÓN>.



15-04-2024

0395-DBRA-UPT-2024

SUMMARY

Adapting to a prosthesis can be difficult because these parts are foreign to the body, and the process takes time for the mind, especially the body. Electronic rehabilitation systems today have emerged as a solution or support to traditional rehabilitation. In the present work, a bioelectronic system is developed for training in using a myoelectronic prosthesis in rehabilitating people with transradial amputation. The system consists of three blocks: acquisition, processing, and activation. During the acquisition stage, myoelectric signals are obtained in real-time using the Myo Armband sensor to be sent to the computer through Bluetooth wireless communication to identify gestures. In addition, the signals from the RFP 602 pressure sensors are obtained using the Arduino UNO development board. With these signals, processing is done by applying the PID control technique to activate the actuators based on the gesture received and grabbing objects of different compositions and sizes. The communication between the sensor and actuator elements, the recognition of the gesture, as well as the visualization of the sensor signals is carried out through a graphical user interface (GUI) programmed in Matlab. From the validation tests of the connection between the sensor and the interface (GUI), an average time of 9 seconds is obtained, which allows verifying that there is a successful connection. From the grip and response time tests with the prosthesis, 84% effectiveness is obtained with an average time of 2.93 seconds. This indicates that the system meets the expected functionality. From this, the bioelectronic system meets the previously established requirements.

Keywords: <BIOELECTRONIC PROTOTYPE>, <MYOELECTRIC PROSTHESIS>, <AUTOMATIC CONTROL>, <GRAPHIC USER INTERFACE (GUI)>, <PROGRAMMING>.



Lic. Nelly Padilla P.

0603818717

DOCENTE FIE

15-04-2024

0395-DBRA-UPT-2024

INTRODUCCIÓN

Parte de la naturaleza humana es generar soluciones a los problemas que se presentan, el desarrollar dispositivos que puedan sustituir una extremidad con los diferentes materiales disponibles de tal manera que mejoren la calidad de vida de los pacientes ha sido una prioridad a lo largo de la historia. Las primeras prótesis fueron elaboradas con madera y hierro para realizar tareas básicas como el vestir y comer. Con el desarrollo de la tecnología se logró obtener señales eléctricas a partir de los movimientos de los músculos, denominadas señales EMG, gracias a esto las prótesis también evolucionaron hasta las actuales que se denominan prótesis mioeléctricas, generalmente desarrolladas para los miembros superiores debido a que es más fácil obtener las señales mioeléctricas necesarias para el control de esta.

Las amputaciones de miembros superiores son el resultado de diferentes factores como enfermedades y traumas, en el caso de enfermedades puede ser la gangrena, cáncer de hueso o algún defecto congénito y en el caso de traumas, los accidentes de tránsito y lesiones en el lugar de trabajo son las principales causas.

Según la OMS y la OPS cerca del 15% de la población mundial padece algún tipo de discapacidad. (CONADIS, 2022). Según el Consejo para la igualdad de discapacidades en 2022, en Ecuador el 2,73% (471.205) de personas padecen algún tipo de discapacidad, de los cuáles el 45,66% (215.156) corresponde a discapacidad física.

En la actualidad no existen sistemas que permitan a las personas que padecen de alguna discapacidad realizar un entrenamiento previo a la obtención de sus prótesis de tipo mioeléctricas y mucho menos si estas cuentan con un sistema de reflexión de fuerza que les permita realizar el agarre de objetos, por lo cual los pacientes al obtener sus prótesis no les dan uso por la falta de maniobrabilidad de esta.

El presente proyecto técnico denominado “Desarrollo de un sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioelectrónica en la rehabilitación de personas con amputación transradial ”, tiene como objetivo principal el desarrollo de una interfaz gráfica que permita a los pacientes realizar un entrenamiento en el uso de una prótesis de tipo mioeléctrica, en la cual podrá generar movimientos básicos, como la apertura y cierre de la mano y poder agarrar diferentes objetos, además de poder monitorear en tiempo real las señales mioeléctricas y de la instrumentación utilizada. Para la activación del sistema bioelectrónico se hace uso de sensores de tipo mioeléctricos, sensores de presión y servomotores, por medio de los cuales se consigue una activación acorde y controlada de la prótesis mioeléctrica.

CAPÍTULO I

1. DIAGNOSTICO DEL PROBLEMA

1.1 Antecedentes

La mano ha sido la compañera fundamental del cerebro para convertir el pensamiento en acción, ya que las ideas se traducen mecánicamente en acciones, estimulando así el cerebro para construir nuevos programas motores, rediseñar vías sinápticas, procesar estímulos visuales para crear patrones de movimiento, cambiar el plano de referencia y crear jerarquía de representaciones para afinar los procesos cognitivos útiles para el control motor. En otras palabras, el cerebro le habla a la mano como la mano le habla al cerebro. (Vargas et al., 2011)

Las prótesis para miembros apendiculares que funcionan electrónicamente más difundidas son las de tipo mioeléctricas, en donde el movimiento se facilita capturando potenciales eléctricos del músculo a nivel de la piel mediante electrodos de superficie o directamente en el músculo (mediante electrodos invasivos). Estas actividades eléctricas, a menudo presentadas como señales, sirven como información para controlar los movimientos que puede realizar la prótesis. (Bravo, 2003)

En Alicante, España, en la Escuela Politécnica Superior se desarrolló un entorno virtual para tareas de manipulación y agarre, con el objetivo de convertirlo en un entorno familiar y cotidiano, se realizaron diversas pruebas de usabilidad con muchas personas, de modo que el trabajo del proyecto se evaluó examinando señales mioeléctricas provenientes de sensores mecánicos. El resultado final fue positivo porque su sistema demostró que cumplía su objetivo principal de enseñar a varios usuarios cómo controlar eficazmente un brazo robótico mediante señales electromiográficas. En futuras investigaciones se pretende enviar los movimientos del brazo robótico del entorno virtual a un brazo robótico real, para realizar las mismas tareas de manipulación y agarre. (Máximo, 2022)

En México, estudios reflejan que muchos amputados no llegan a utilizar regularmente sus prótesis porque no son capaces de controlarlas eficientemente o porque no ofrecen suficientes funciones útiles para compensar el esfuerzo de aprendizaje necesario para utilizarlas. En este caso, la prótesis que utiliza el paciente es una imagen tridimensional de la mano en la pantalla de un ordenador. Este sistema es como un juego de computadora y se elimina la sensación de rechazo inicial que supone la colocación de un elemento extraño (la prótesis) sobre el brazo. (Alonso et al., 2002)

En Colombia, en la Escuela Colombiana de Ingeniería en 2017, se presentó una evaluación de un sistema sensorial para el control del agarre en una prótesis mioeléctrica, empleando realimentación de esfuerzos entre la prótesis y los objetos agarrados, utilizando sensores de presión y piezoeléctricos. Ésta se basa en un control híbrido, el cual le permite acciones deliberativas y reactivas. De esta manera pueden determinar la fuerza necesaria para realizar un agarre estable. Los resultados de la evaluación muestran que muchos usuarios se adaptan más fácilmente a las nuevas condiciones cuando utilizan el sistema de retroalimentación. (Rodríguez et al., 2017)

En la Pontificia Universidad Católica del Perú, en 2014, se diseñó una prótesis mioeléctrica para desarticulación de muñeca, que permite agarrar objetos con una superficie mayor a 3 cm³ y regular la fuerza de agarre, a través del procesamiento de datos y algoritmos matemáticos permite estimar la fuerza resultante de la relación entre la función de agarre y la contracción muscular. Señales en forma de datos y lógica controlan tres servomotores para abrir y cerrar el brazo protésico ajustando la fuerza. De esta forma se asegura una mejor experiencia al paciente y así mejorar su desempeño al contar con el dispositivo físico. (Salas, 2014)

A nivel nacional, en la Universidad de las Fuerzas Armadas -ESPE, se presentó un informe con el diseño de una prótesis biónica de mano de 7 grados de libertad para su validación. El dispositivo cuenta con un sistema de regulación de presión que, procesando señales musculares, permite controlar el agarre de objetos para obtener un modelo anatómicamente proporcional para el usuario, que posee un sistema de control compuesto por encapsulados de fluido magneto reológico y sensores de presión que permiten un control autónomo. (Arteaga et al., 2015)

En la Universidad de Azuay, implementaron un entrenador virtual de prótesis mioeléctricas es un software o interfaz que tiene como principal objetivo la sustitución de la prótesis real durante el proceso de aprendizaje y acoplamiento del usuario a los sistemas mioeléctricos para evitar inversiones económicas, muchas veces innecesarias, cuando una determinada persona no es apta para operar con este tipo de prótesis. Por otra parte, busca también reducir el impacto psicológico generado en la persona al momento de manipular una prótesis. (Burbano et al., 2016)

A nivel institucional se desarrolló un prototipo de prótesis mioeléctricas de mano de bajo costo, correspondiente al tipo de mango esférico, en el que el rango de agarre del objeto esférico se calcula de 2 a 12 cm. Se calcula que la relación de sujeción efectiva del prototipo es del 78,89%, lo que demuestra que es totalmente funcional. Al calcular los gastos se determinó que su precio está por debajo de las prótesis de tipo mioeléctricas comerciales. (Guevara, 2020)

En la facultad de mecánica en 2020 se ha implementado un sistema de control basado en el procesamiento de señales electromiográficas para la activación de una prótesis robótica. Utilizaron una pulsera Myo para recibir señales EMG y transmitir las de forma inalámbrica a un microcontrolador para su posterior análisis, utilizando servomotores como actuadores de movimiento. El sistema de control se lo realizó a través de una aplicación propia del sensor Myo Connect, logrando crear procesos de calibración para relacionar señales del antebrazo. Las pruebas con varios movimientos mostraron un rendimiento de reconocimiento de gestos del 89,2%. (Mora, 2020)

En 2017 un trabajo de magister desarrolló un sistema para la adquisición y procesamiento de señales electromiográficas con Matlab e identificar varios movimientos tradicionales como la extensión y flexión del brazo. El sistema consta de un sensor de electromiografía y un acelerómetro que transmite las lecturas a la placa Arduino y luego a la computadora de forma inalámbrica. Las señales se procesan y se muestran en tiempo real a través de una gráfica. Este dispositivo tiene una efectividad superior al 96%, aunque a distancias limitadas. (Zabala, 2017)

1.2 Formulación del Problema

¿Cómo desarrollar un sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica en la rehabilitación de personas con amputación transradial?

1.3 Sistematización del Problema

¿Cuáles son los requerimientos que debe cumplir el sistema bioelectrónico para la rehabilitación de personas con amputación transradial?

¿Cuáles son los sensores musculares para la adquisición de señales mioeléctricas y los softwares para el monitoreo y procesamiento de señales?

¿Cuál es el proceso para la adquisición de señales mioeléctricas para su posterior análisis y procesamiento?

¿Cuál es el controlador adecuado para la activación de los actuadores de la prótesis?

¿Como realizar las pruebas de validación del sistema bioelectrónico enfocado en el entrenamiento, uso y manejo de una prótesis?

1.4 Justificación Teórica

Las principales razones por las que muchas personas no utilizan los servicios de rehabilitación incluyen la falta de personal capacitado en rehabilitación y los altos costos, así como el aumento en el número de personas que, por diversas razones y situaciones diferentes, necesitan tratamiento. Hoy en día los sistemas electrónicos de rehabilitación social han surgido como una solución o apoyo a la rehabilitación social tradicional, es decir, se presentan como mecanismos de rehabilitación social. Incluso se espera que en un futuro próximo este tipo de desarrollo tecnológico reduzca el coste de recuperación en el largo plazo. La investigación sobre los usos y beneficios de este tipo de terapia se ha ampliado en los últimos años con el desarrollo del equipamiento necesario. Son muchos los avances en el campo de la terapia ocupacional en la actualidad, pero lamentablemente no existen avances tecnológicos que puedan solucionar la escasez de personal capacitado en este campo. Además, algunos avances tecnológicos tienen costos muy elevados. (Campos et al., 2021)

Según la OMS y la OPS, en su informe CD58/INF/7- Plan de acción sobre discapacidades y rehabilitación del 2019, elaborado en septiembre del 2020, indica que, aproximadamente el 15% de la población padece algún tipo de discapacidad. Las mujeres sufren discapacidades con más

frecuencia que los hombres, y las personas mayores lo son con más frecuencia que los jóvenes. En todo el mundo, estas personas enfrentan grandes dificultades en su desarrollo social, económico, educativo y profesional porque encuentran barreras que a menudo surgen debido al estigma de quienes los rodean. Este problema se agrava cuando son excluidos y no se identifica con precisión su discapacidad, lo que les dificulta desarrollar sus fortalezas e integrarse al mundo educativo, social y laboral. (CONADIS, 2022)

Según el Consejo Nacional para la Igualdad de Discapacidades en 2022, en Ecuador el 2,73% (471.205) de personas registran algún tipo de discapacidad, de los cuales el 45,66% (215.156) corresponden a discapacidad física, debido al gran número de personas con discapacidad, muchos programas de atención enfrentan problemas como: falta de aplicación de programas sociales para personas con discapacidad y planes de trabajo por falta de recursos económicos, incapacidad de implementación debido a crisis económica. Intentaron ser reales y tuvieron que trabajar con un presupuesto reducido. (Conadis, 2022)

En Colombia, el periódico UNAL expone que, adaptarse a una prótesis o implante puede resultar difícil porque estas partes son extrañas al cuerpo y el proceso lleva tiempo para la mente y especialmente para el cuerpo. Acostumbrar el cuerpo a las prótesis es una tarea difícil que la ciencia y la tecnología han tratado de facilitar, por lo que en los últimos veinte años se han vuelto más ligeras y funcionales gracias a nuevos materiales como el titanio y el metal. Sin embargo, 2 de cada 10 personas no pueden personalizar ni gestionar sus dispositivos. Las prótesis se descartan parcialmente porque pesan más de lo esperado o porque no se ha desarrollado su función durante la rehabilitación. Este proceso incluye ejercicios de fortalecimiento para restaurar la fuerza muscular previa a la amputación, así como la detección básica de señales producidas por los músculos durante la contracción y relajación (señales electromiográficas) para mejorar la movilidad. (Cruz, 2017)

Por todos los problemas de las personas con amputaciones transradiales, se plantea diseñar un entrenador para mejorar en la rehabilitación, en su experiencia y adaptabilidad a un elemento extraño a su cuerpo. El sistema bioelectrónico contará con un sistema mioeléctrico y sensores de presión, y a diferencia de los sistemas propuestos anteriormente se realizará un sistema de control de lazo cerrado o retroalimentado para mejorar su coordinación y desempeño en actividades del día a día, reduciendo la cantidad de elementos actuadores y sensores, además de realizar un monitoreo en tiempo real mediante una GUI, que permitirá al paciente ser consciente de su desempeño y poder enfocarse en las señales pertinentes.

1.5 Justificación Aplicativa

Son diversas las razones por las que muchas personas no utilizan los servicios de rehabilitación de prótesis, entre las principales se incluyen la falta de personal capacitado en rehabilitación y los altos costos, por eso se presenta una propuesta de un sistema bioelectrónico para el entrenamiento en el uso de prótesis, el mismo que se divide en varias fases o bloques diferenciados para llegar a una aplicación plenamente funcional que cumpla con los requisitos estipulados, el cual se describe en la Ilustración 1.

La primera fase consiste en la adquisición de señales, por un lado el sensor muscular con los electrodos EMG captarán las señales mioeléctrica que filtrarán, amplificarán y acondicionarán, por otro lado los sensores de presión y fuerza por medio de un divisor de voltaje acondicionarán sus señales de salida, ambos tipos de señales son de tipo analógico, posteriormente todas las señales son recibidas por una tarjeta de desarrollo que cuenta con un módulo convertido analógico digital o ADC para ser entramadas.

La segunda fase es el desarrollo de una GUI, para la comunicación entre la tarjeta de desarrollo y una plataforma de programación y cálculo numérico, la información entramada es recibida a través del puerto serie por la plataforma en donde se procesarán las señales para monitorear y generar gráficas en tiempo real de las señales analógicas de entrada y digitales de salida. A las señales analógicas del sensor muscular se les aplicara un control por retroalimentación con un setpoint para obtener a la salida una señal de tipo PWM.

La tercera fase consiste en enviar las señales PWM de salida de la plataforma devuelta a la tarjeta de desarrollo, la señal es el dato de potencia necesaria para proceder con la activación moderada y controlada de los actuadores de la prótesis mioeléctrica.



Ilustración 1-1: Diagrama de bloques del proyecto

Realizado por: Casierra Alvaro, 2024.

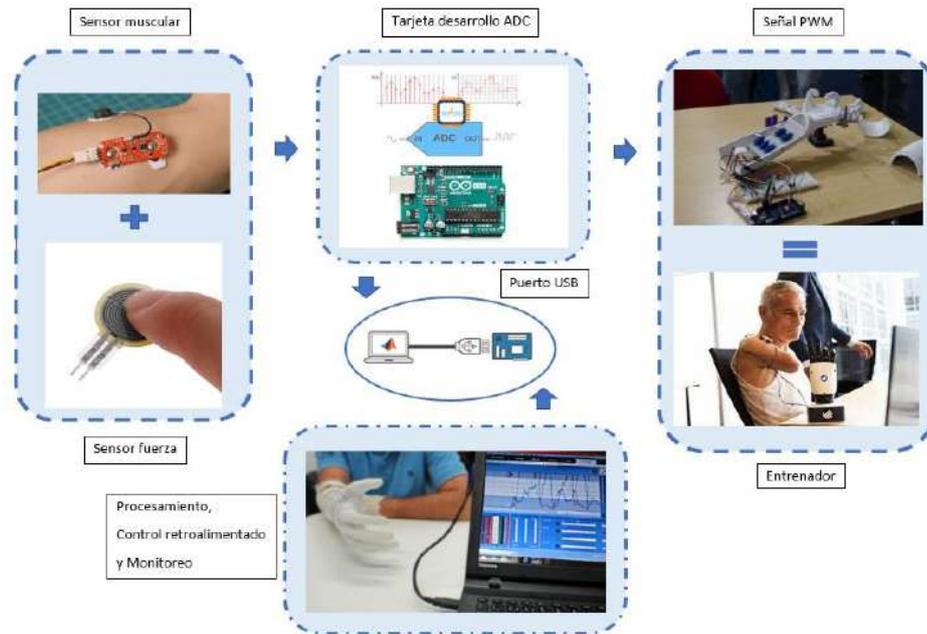


Ilustración 1-2: Esquema del sistema bioelectrónico

Realizado por: Casierra Alvaro, 2024.

1.6 Objetivos

1.6.1 *Objetivo General*

Desarrollar un sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica en la rehabilitación de personas con amputación transradial.

1.6.2 *Objetivos Específicos*

- Establecer los requerimientos que debe cumplir el sistema bioelectrónico para la rehabilitación de personas con amputación transradial.
- Estudiar los sensores musculares para la adquisición de señales mioeléctricas y los softwares para el monitoreo y procesamiento de señales.
- Definir el proceso para la adquisición de señales mioeléctricas para su posterior análisis y procesamiento.
- Diseñar un controlador apropiado para la activación de los actuadores de la prótesis.
- Evaluar si el sistema bioelectrónico enfocado en el entrenamiento, uso y manejo de una prótesis cumple con los requerimientos establecidos.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Discapacidad

La Organización Mundial de la Salud OMS define la discapacidad como “un fenómeno complejo que refleja una interacción entre las características del organismo humano y las características de la sociedad en la que vive”. El término incluye problemas que afectan la estructura o función del cuerpo y las limitaciones que causan en la actividad y participación adecuadas. (Observatori de la Discapacitat Física -ODF, 2016)

2.1.1 *Persona con discapacidad*

Las personas con discapacidad presentan alguna deficiencia física, mental, intelectual o sensorial de larga duración que, al enfrentarse a diversos desafíos, pueden impedirles participar plena y efectivamente en la sociedad con igualdad de condiciones respecto de los demás. Según un Informe Mundial, cerca del 15% de la población mundial padece algún tipo de discapacidad. Además, según estos informes las mujeres tienden a tener mayores probabilidades de sufrir alguna discapacidad. (Organización Panamericana de la Salud, 2019)

2.1.2 *Discapacidad física*

La discapacidad de la movilidad ocurre cuando la condición física de una persona le impide moverse de manera permanente e irreversible utilizando su sistema neuromotor. Generalmente afecta al sistema musculoesquelético, especialmente a las extremidades, aunque también puede manifestarse por una disminución de la movilidad de los músculos esqueléticos. (Observatori de la Discapacitat Física -ODF, 2016)



Ilustración 2-1: Discapacidad física

Fuente: (Observatori de la Discapacitat Física -ODF, 2016)

2.2 Amputación

Una amputación es una condición permanente, que provoca un impedimento para realizar actividades diarias de carácter laboral, recreacional y de ámbito social, si no se toman las acciones de rehabilitación necesarias. Son varios los factores que determinan el resultado funcional después de la amputación de una extremidad, siendo el más importante el nivel de esta. En la Ilustración 2-2 se muestran los tipos generales de amputación. (Espinoza y García, 2014)

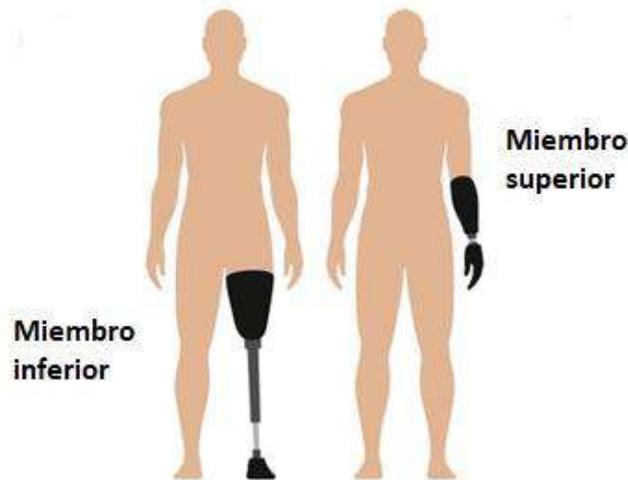


Ilustración 2-2: Tipos de amputación

Realizado por: Casierra Alvaro, 2024.

2.2.1 Amputación transradial

Las amputaciones a un nivel anatómico se las divide en función del sitio en el cual se localizan. La amputación transradial está entre la muñeca y el codo, en la cual se realiza un corte a través de los huesos radio y cubito. De acuerdo con la extensión del parte residual conocido como muñón, se clasifica en: tercio distal, medio o proximal. (Bejarano, 2019) En la Ilustración 2-3 se muestran los principales niveles de amputación de miembro superior.

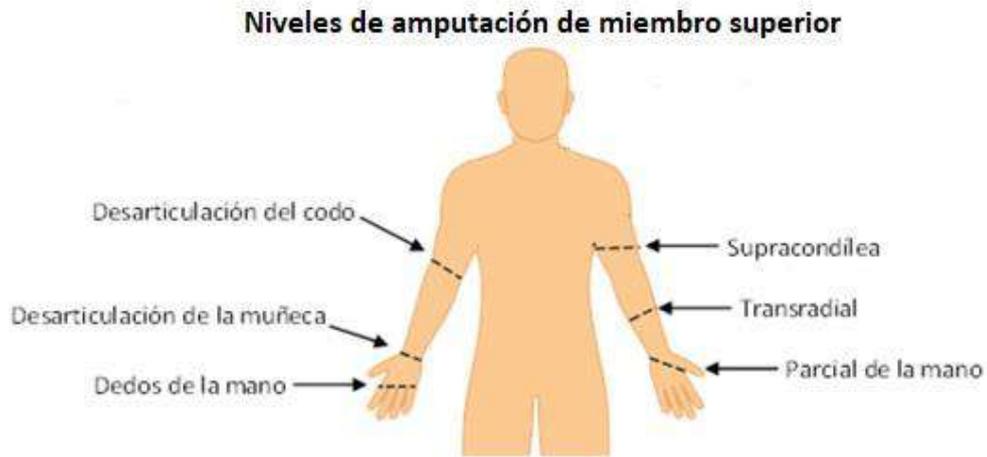


Ilustración 2-3: Niveles de amputación de miembro superior

Realizado por: Casierra Alvaro, 2024.

2.3 Tipos de prótesis de miembro superior

Se puede clasificar los tipos de prótesis, como pasivas y activas. Las activas pueden ser accionadas y controladas por señales generadas en los diferentes grupos musculares. (Cruz, 2018) En la Ilustración 2-4 se muestran los tipos de prótesis de miembro superior.

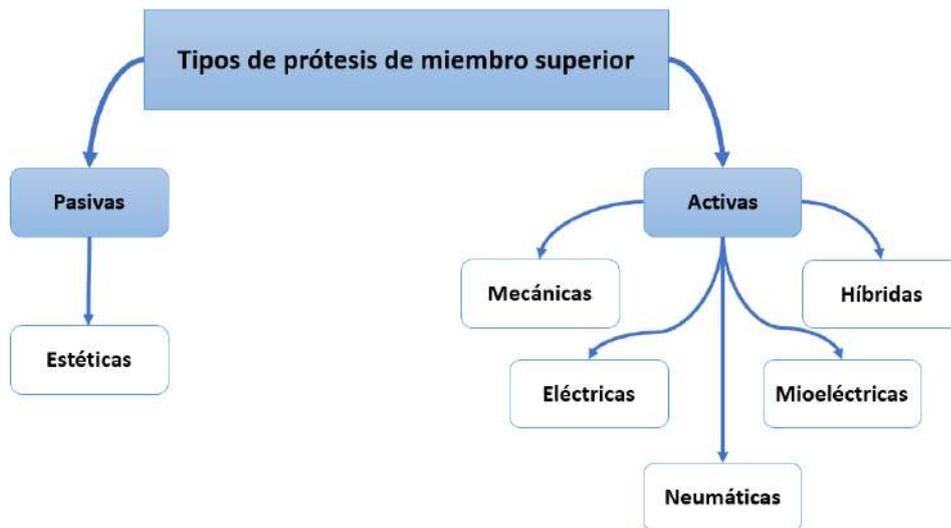


Ilustración 2-4: Tipos de prótesis de miembro superior

Fuente: (Cruz, 2018)

2.3.1 Estéticas

Son aquellas cuya función es generar comodidad y confort en los usuarios. Su aspecto tiene ser muy cercano a una extremidad natural, manteniendo un alto nivel de estética. Estas pueden usarse

dependiendo los niveles de amputación. (Cruz, 2018) En la Ilustración 2-5 se muestra un ejemplo de prótesis estética.



Ilustración 2-5: Prótesis estética

Realizado por: Casierra Alvaro, 2024.

2.3.2 *Mecánicas*

Estas se conectan al cuerpo mediante cables o arneses de materiales amigables con la piel. Utilizan los movimientos de diversas partes del cuerpo para controlar la prótesis. Generalmente el elemento final es una pinza para la toma de objetos. (Cruz, 2018) En la Ilustración 2-6 se muestra un ejemplo de prótesis mecánica.

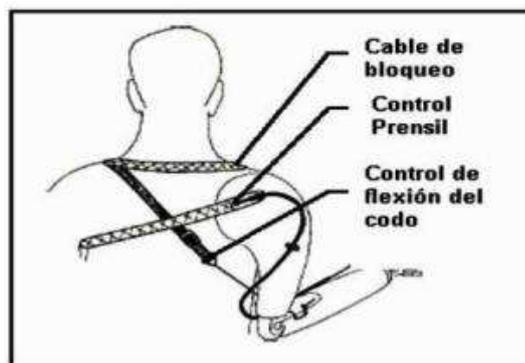


Ilustración 2-6: Prótesis mecánica

Fuente: (Cruz, 2018)

2.3.3 *Eléctricas*

Como su nombre lo indica son prótesis controladas con actuadores eléctricos como microcontroladores, servomotores, interruptores entre otros. Utilizan sockets como elementos intermedios entre la prótesis y el paciente. Al ser eléctricas se debe tener en consideración la exposición y las condiciones ambientales, tales como su contacto directo con agua. (Cruz, 2018) En la Ilustración 2-7 se muestra un ejemplo de prótesis eléctrica.



Ilustración 2-7: Prótesis eléctrica

Realizado por: Casierra Alvaro, 2024.

2.3.4 *Neumáticas*

Las prótesis neumáticas pueden ser accionadas mediante ácido carbónico comprimido, que genera una gran cantidad de fuerza. Estas prótesis cuentan con un gran número de accesorios que pueden genera incomodidad en los usuarios. El movimiento es transmitido a través de un sistema de tuberías especiales, lo cual permite tener una gran precisión y control sobre la fuerza al momento de tomar objetos de distintos tamaños y/o composición. (Cruz, 2018) En la Ilustración 2-8 se muestra un ejemplo de prótesis neumática.



Ilustración 2-8: Prótesis neumática

Fuente: (Cruz, 2018)

2.3.5 Mioeléctricas

El control de estas prótesis se realiza por medio de un sensor que permite adquirir las señales mioeléctricas generadas en los músculos. Estas tienen un alto nivel de adaptación y permiten tener un control independiente de cada elemento de la prótesis. Además, cuenta con un nivel estético alto y brinda una precisión en la velocidad y fuerza. Los sensores mioeléctricos adquieren las señales por medio de electrodos superficiales conectados a la piel. Estos monitorean la actividad eléctrica de los músculos para procesar e interpretar la generación de movimientos. Este tipo de prótesis tienen un valor económico elevado en el mercado. (Cruz, 2018) En la Ilustración 2-9 se muestra un ejemplo de prótesis mioeléctrica.



Ilustración 2-9: Prótesis mioeléctrica

Realizado por: Casierra Alvaro, 2024.

2.3.6 Híbridas

Una prótesis híbrida es aquella que combina una o varias técnicas de las mencionadas. Pueden utilizar los movimientos generados por el cuerpo humano y a la vez accionamientos de tipo eléctrico. (Cruz, 2018) En la Ilustración 2-10 se muestra un ejemplo de prótesis híbrida.



Ilustración 2-10: Prótesis híbrida

Fuente: (Cruz, 2018)

2.4 Tipos de agarre con prótesis

Existen distintos tipos de agarres de acuerdo con los propósitos que el usuario puede requerir. Al cambiar la cinemática de la mano se obtienen las distintas formas de agarre. En 1919, Schlesinger publicó una clasificación de seis movimientos de agarre que permiten realizar actividades simples o complejas, estos son: agarre cilíndrico, de punta, de gancho, de palma, esférico y de lado. Estos movimientos se pueden realizar en prótesis a través de actuadores eléctricos y sistemas de transmisión de fuerza. (Cuesta y Tamay, 2022)

En la Tabla 2-1 se presenta el uso o empleo de los principales tipos de agarre que se pueden generar con los diferentes tipos de prótesis de miembros superiores, donde se puede deducir que todos son necesarios para que el paciente pueda desempeñarse con normalidad en la mayoría de las actividades cotidianas, tanto a nivel profesional como personal.

Tabla 2-1: Características principales de los tipos de agarre.

Tipo de agarre	Característica
Cilíndrico	Se emplea para hacer un agarre tipo prensil con objetos con geometrías continuas.
Punta	Se usa para la sujeción de geometrías pequeñas y finas, con la punta de los dedos.
Gancho	Empleada para levantar objetos con geometrías variables de gran longitud.
Pinza	Utilizada para la sujeción de objetos pequeños con las yemas de los dedos.
Esférica	Se emplea para la sujeción de objetos con circunferencias definidas.
Lateral	Es para la sujeción de objetos planos y delgados.

Fuente: (Cuesta y Tamay, 2022)

Realizado por: Casierra Alvaro, 2024.

En la Ilustración 2-11 se muestran los tipos de agarre.

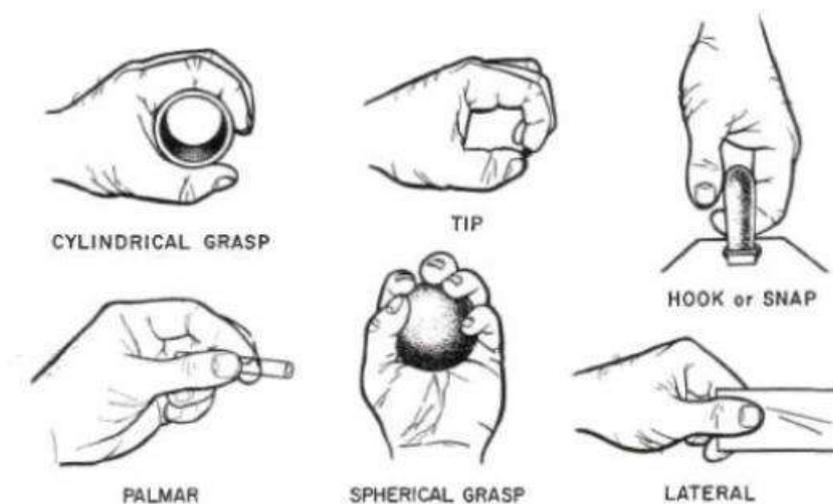


Ilustración 2-11: Tipos de agarre

Realizado por: Casierra Alvaro, 2024.

2.5 Sistemas de control de lazo cerrado

Un sistema de control lazo cerrado tiene a la señal de salida retroalimentada marcando una influencia directa en la operación de control. La diferencia entre la señal de entrada y la señal de

salida se considera la señal de error. Esta señal es la que va a actuar en el sistema, de tal modo que lleve la señal de salida a un valor deseado. El término "bucle cerrado" significa que se va a usar una señal de retroalimentación como negativa que permita la reducción de los errores del sistema. (Perez et al., 2007) En la Ilustración 2-12 se muestra un ejemplo de sistema de control en lazo cerrado.

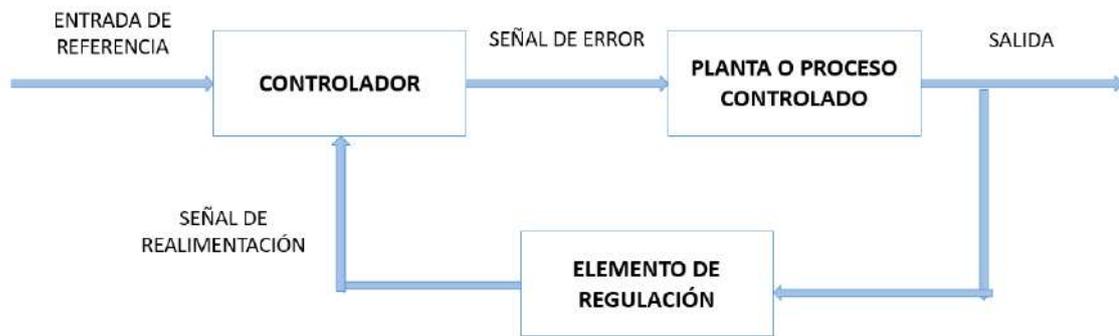


Ilustración 2-12: Sistema de control en lazo cerrado

Fuente: (Perez et al, 2007)

2.6 Controladores

Los controladores son dispositivos o algoritmos diseñados para regular el comportamiento de un sistema dinámico con el fin de lograr un rendimiento deseado. El control automático se utiliza en una amplia variedad de aplicaciones, desde sistemas industriales y procesos de fabricación hasta sistemas de control de vuelo en aeronaves. Dentro de los controladores con mejor desempeño y aplicabilidad en la industria, se tiene a los controladores de tipo proporcional, integral y derivativo (PID). (Anaya et al., 2014)

Se puede describir sus términos de la siguiente manera:

- El término proporcional provee una acción general de control a la señal de error a través de un factor de ganancia permitiendo pasar todo tipo de frecuencias.
- El término integral reduce los errores en estado estacionario a través de compensación de baja frecuencia.
- El término derivativo ayuda a mejorar la respuesta transitoria mediante una compensación a muy altas frecuencias.

(Anaya et al., 2014)

En la Tabla 2-2 se presentan las variaciones que se producen al aumentar las ganancias K_p , K_i y K_d en los controladores en función del tiempo, específicamente en el tiempo de subida, saturación, tiempo de estabilización, error en estado estacionario y estabilidad.

Tabla 2-2: Características de las ganancias del controlador PID.

	Tiempo de Saturación		Tiempo de estabilización	Error en estado estacionario	Estabilidad
Aumenta K_p	Decremento	Incremento	Pequeño incremento	Decremento	Reducción
Aumenta K_i o T_i	Pequeño decremento	Incremento	Incremento	Gran decremento	Reducción
Aumenta K_d o T_d	Pequeño decremento	Decremento	Decremento	Cambios menores	Mejora

Fuente: (Anaya et al., 2014)

Realizado por: Casierra Alvaro, 2024.

2.6.1 Controlador PI

El controlador proporcional - integral (PI) es muy usado en los sistemas de control automático, ya que, combina las características de un controlador proporcional (P) y un controlador integral (I). Cuando se combinan, el controlador PI ofrece beneficios tanto en la respuesta a corto plazo como en la eliminación de errores de estado estacionario. También se recomienda la acción PI cuando hay retardos en el proceso. Otro caso en el que se debería desconectar la acción derivativa es cuando el proceso está contaminado con niveles de ruido elevados. (Álamo, 2015)

La salida del controlador PI se calcula sumando la salida proporcional y la salida integral. La fórmula matemática para un controlador PI se presenta en la Ecuación 1:

$$U(t) = K_p * e(t) + k_i * \int_0^t e(\tau) d\tau \quad (1)$$

Donde, $U(t)$ es la señal de control que se aplica al sistema en el tiempo t , $e(t)$ es el error actual en el tiempo t (diferencia entre la referencia y la salida del sistema), K_p es la ganancia proporcional, K_i es la ganancia integral.

2.6.2 Controlador PD

Es un tipo específico de controlador que combina las características de un controlador proporcional (P) y un controlador derivativo (D). Cuando se combinan, el controlador PD ofrece beneficios tanto en la respuesta a corto plazo como en la estabilización del sistema. El control PD puede ser apropiado cuando el proceso a controlar incorpore ya un integrador, el diferencial es sensible al ruido porque tiene una ganancia relativamente alta a frecuencias elevadas, por lo que para niveles de ruido altos se debe reducir la ganancia o se debe omitir el diferenciador. De forma similar, en procesos con grandes tiempos muertos, el principal efecto del término derivativo ya no es efectivo debido a que la aproximación lineal solo es válida para valores pequeños de T_d . Debido al tiempo muerto, se presenta un retraso antes de que se pueda detectar el efecto de cualquier acción de control sobre la variable del proceso. (Álamo, 2015)

La salida del controlador PD se calcula sumando la salida proporcional y la salida derivativa. La fórmula matemática para un controlador PD se expresa de acuerdo con la Ecuación 2:

$$U(t) = k_p * e(t) + K_d * \frac{de(t)}{dt} \quad (2)$$

Donde, $U(t)$ es la señal de control que se aplica al sistema en el tiempo t , $e(t)$ es el error actual en el tiempo t (diferencia entre la referencia y la salida del sistema), K_p es la ganancia proporcional, K_d es la ganancia derivativa.

2.6.3 Controlador PID

Por sus siglas proporcional-integra-derivativo, es un tipo de control en el cual se establecen para las constantes, valores en función del resultado deseado. Este controlador es un caso específico de compensador en adelanto-retraso, proporcional-derivativo y proporcional-integral, respectivamente, obteniendo un controlador con dos ceros, con un polo en el origen y una ganancia. (Gil y Rubio, 2010)

La salida total del controlador PID se calcula sumando las salidas de cada componente. La fórmula matemática básica para un controlador PID se puede expresar de acuerdo con la Ecuación 3:

$$U(t) = K_p * e(t) + K_i * \int_0^t e(\tau) d\tau + K_d * \frac{de(t)}{dt} \quad (3)$$

Donde, $U(t)$ es la señal de control que se aplica al sistema en el tiempo t , $e(t)$ es el error actual en el tiempo t (diferencia entre la referencia y la salida del sistema), K_p , K_i y K_d son las ganancias proporcionales, integral y derivativa, respectivamente.

El controlador PID es ampliamente utilizado en sistemas de control automático debido a su capacidad para proporcionar un equilibrio óptimo entre la respuesta rápida, la eliminación de errores de estado estacionario y la estabilidad del sistema. La sintonización adecuada de las tres ganancias es esencial para lograr un rendimiento eficiente para aplicaciones específicas como el control de temperatura de un horno industrial o la posición de motores de corriente continua. (Gil y Rubio, 2010)

2.7 Señales

Son una manifestación de los fenómenos físicos que se encuentran presentes en el entorno. Se generan a partir de las actividades realizadas y contienen información relacionada a las mismas. Entre más detallada sea una señal, mejor se podrá conocer características y comportamiento de los fenómenos que la produjeron. La representación de sus datos puede ser en forma gráfica, tablas y matemáticamente en función de variables dependientes e independientes. (Mata, 2017) En la Ilustración 2-13 se muestran ejemplos de señales en el tiempo.

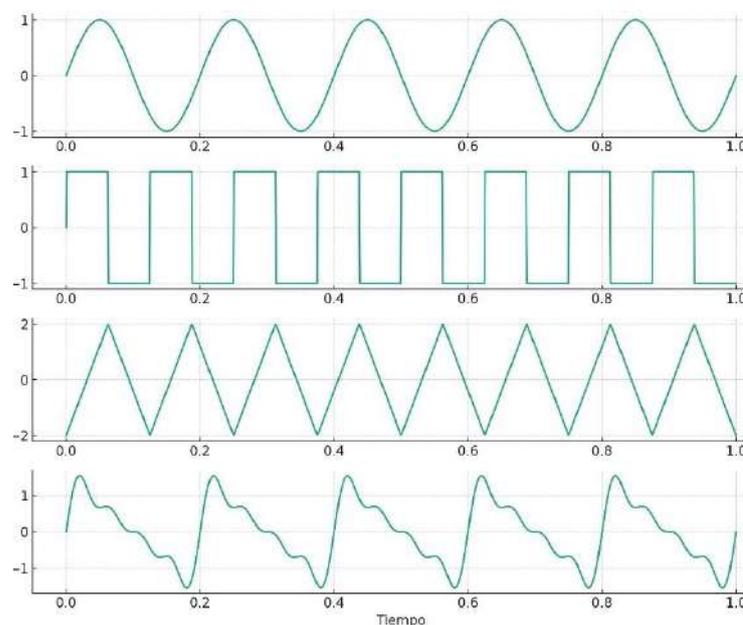


Ilustración 2-13: Señales en el tiempo

Realizado por: Casierra Alvaro, 2024.

2.7.1 Señales mioeléctricas

Las señales de electromiografía son producidas por la contracción de los músculos de cualquier extremidad, como brazos y piernas. Estas resultan de la actividad química producida por la proteína fibrosa miosina. Se pueden medir con equipos adecuados y así aprovechar la información que aportan para el diseño de las prótesis móviles, ya que, esta señal es de tipo eléctrica en el orden de los microvoltios (Vázquez et al., 2017). La señal mioeléctrica suele tener ruido proveniente de varias fuentes, una de ellas la frecuencia de 60 Hz de la línea eléctrica y ruido blanco del ambiente. Es por esto por lo que la señal debe pasar por filtros analógicos para obtener una señal pura. Esta señal permite la integración temporal y espacial de todos los potenciales detectados utilizando electrodos superficiales, a partir de un cierto volumen de tejido. (Alva et al., 2011) En la *Ilustración 2-14* se muestran una señal EMG.

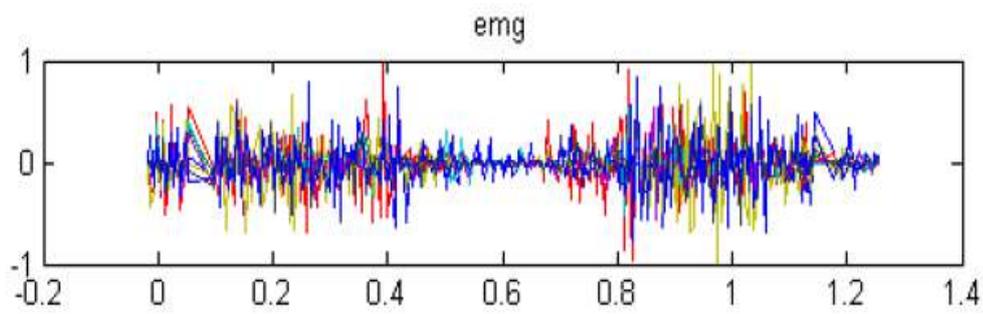


Ilustración 2-14: Señal EMG

Fuente: Thalmic Labs

2.8 Sensores mioeléctricos

Se usan para el registro de la actividad muscular al detectar las señales electromiográficas (EMG). En el mercado se puede encontrar una gran variedad de sensores mioeléctricos, los cuáles, permiten su adquisición y procesamiento. Entre los más usados se tiene a los siguientes:

2.8.1 Myo Ware

Es el último sensor EMG de ADVANCER TECHNOLOGIES. Esta empresa ha desarrollado un sensor con un diseño portátil que se puede utilizar mediante electrodos conectados directamente a un módulo MyoWare para obtener una señal EMG amplificada, suavizada e integrada con el ajuste de ganancia integrado. Además, este sensor permite tomar señales durante los movimientos musculares para controlar dispositivos externos como motores, luces, etc., mediante su procesamiento en una tarjeta de desarrollo. (Suprpto et al., 2023) En la *Ilustración 2-15* se muestra el sensor Myo Ware con los electrodos superficiales.

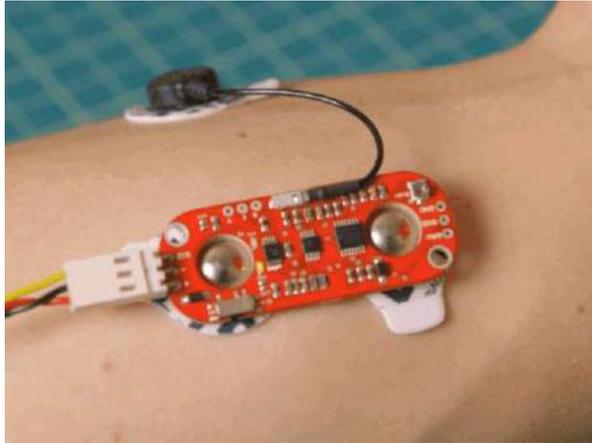


Ilustración 2-15: Sensor Myo Ware

Realizado por: Casierra Alvaro, 2024.

2.8.2 *Myo Armband*

Desarrollado por Thalmic Labs, es un brazalete con el cuál es posible controlar diferentes dispositivos de manera inalámbrica mediante los movimientos del brazo. El brazalete está conformado por ocho sensores de nivel médico de acero inoxidable y de alta sensibilidad, los cuales permiten adquirir las señales eléctricas emitidas por los músculos y enviarlos de forma inalámbrica usando comunicación Bluetooth. (Arteaga, 2015) Todos los sensores están dentro de una banda de plástico flexible, la cual debe estar bien ajustada al músculo para poder registrar correctamente los movimientos. El brazalete Myo además de las señales EMG combina con señales de un giroscopio, un acelerómetro y un magnetómetro, de tres ejes. (Benito, 2015) En la Ilustración 2-16 se muestra el sensor Myo Armband ubicado en el antebrazo.



Ilustración 2-16: Sensor Myo Armband

Realizado por: Casierra Alvaro, 2024.

2.8.3 *EMG AD8832*

Este sensor desarrollado por Advancer Technologies permite monitorear el potencial eléctrico producido por las células mide la actividad muscular. El sensor filtra, rectifica, amplifica y procesa todas las señales eléctricas producidas y realiza la conversión de señales analógicas en digitales para que puedan ser procesadas de una manera simple por tarjetas de desarrollo como los microcontroladores. Al tener dimensiones pequeñas lo convierte en un sensor atractivo para actividades relacionadas con la medicina. Además, cuenta con implementos como los electrodos superficiales y el cable EMG de nivel profesional con el cuál se obtienen las señales de alta calidad. (Ortega, 2021) En la Ilustración 2-17 se muestra el sensor EMG AD8832 con sus implementos.



Ilustración 2-17: Sensor AD8832

Fuente: (Ortega, 2021)

2.9 Modulación por ancho de pulso - PWM

Es una técnica de control que se basa en modificar el ciclo de trabajo de una señal periódica (senoidal o cuadrada), permitiendo controlar la cantidad de energía que se envía a una carga. Al variar la señal de referencia y la secuencia de conmutación obtenemos diferentes técnicas de PWM (Posada, 2005). Esta técnica nos permite regular la velocidad de motores de C.C. Este es un método con el que se consigue lograr un conversión digital-analógica variando su ciclo de trabajo, manteniendo la frecuencia constante. Gracias a estas características se usará esta técnica para el control de posición de los servomotores en función de las señales analógicas de entrada provenientes de los sensores de fuerza resistivos. (Pujol, 2005) En la Ilustración 2-18 se muestran varias señales PWM con diferentes ciclos de trabajo.

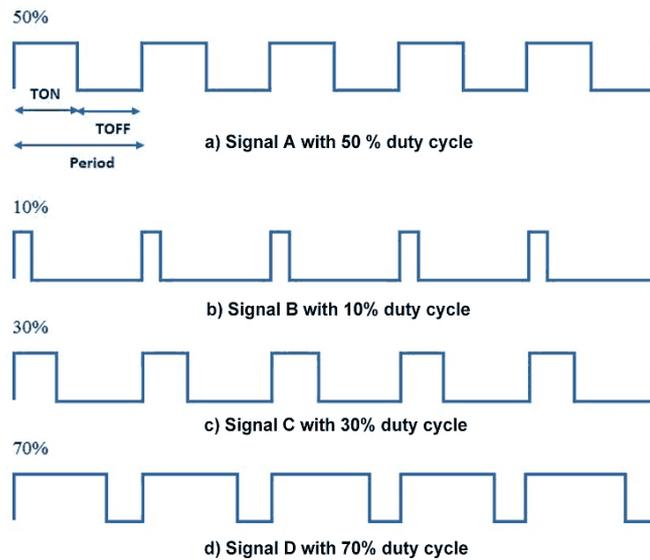


Ilustración 2-18: Ciclo de trabajo PWM

Fuente: (Pujol, 2005)

2.10 Sensor de fuerza

Son dispositivos diseñados para medir una fuerza aplicada sobre estos y, se utilizan en una amplia gama de aplicaciones con fines educativos y médicos. Su función principal es convertir una fuerza en una señal eléctrica, la cual se puede medir, cuantificar y procesar.

2.10.1 MF01

Este es un sensor de dos capas separadas por un espaciador. Su funcionamiento se basa en la presión aplicada sobre puntos de la superficie. De acuerdo con la cantidad de puntos activados, tiene un nivel de activación del semiconductor, lo que provoca que su resistencia disminuya. Además, es autoadherible lo que facilita su colocación en distintas superficies. Su resistencia cuando no está siendo presionado es de 20 MΩ. Su rango de presión varía de 30 gramos a 1 kilogramo. Para su correcto funcionamiento se debe complementar con un divisor de tensión para medir su señal de salida de tipo analógico. (HETPRO, 2015) En la Ilustración 2-19 se muestra el sensor MF01.



Ilustración 2-19: Sensor MF01

Realizado por: Casierra Alvaro, 2024.

2.10.2 RFP-602

Este sensor fabricado por Kuongshun Electronic está compuesto por una película resistiva delgada que permite medir la presión estática y dinámica en cualquier superficie de contacto, convirtiendo la presión ejercida en un área en una señal variable. En función de la curva de fuerza-resistencia, se obtiene la información de la presión externa. Cuanto mayor es la presión, menor es la resistencia en la salida del sensor. (KUONGSHUN, 2018) En la Ilustración 2-20 se muestra el sensor RFP 602.



Ilustración 2-20: Sensor RFP-602

Realizado por: Casierra Alvaro, 2024.

2.10.3 Análisis comparativo de los sensores de fuerza

En la Tabla 2-3 se realiza una comparativa entre dos sensores de presión, donde se detallan las características principales, como el diámetro de su área sensible, la temperatura de trabajo, tiempo de respuesta y el rango de medición. Con estas características se puede determinar cuál de las dos opciones sería la mejor para el proyecto.

Tabla 2-3: Características de los sensores de presión.

Características	MF01	RFP-602
Forma área sensible	Redonda	Redonda
Diámetro	14.7 mm	10 mm
Temperatura de trabajo	-30 °C a 70 °C	-25 °C a 70 °C
Tiempo de respuesta	< 1ms	< 1ms
Rango	30 g a 1 Kg	50 g a 1 Kg

Fuente: (HETPRO, 2015) (KUONGSHUN, 2018)

Realizado por: Casierra Alvaro, 2024.

En base a la Tabla 2-3, se define que el sensor RFP 602 reúne las características necesarias para su implementación en el proyecto.

2.11 Tarjeta de desarrollo

Es una herramienta que permite diseñar prototipos tecnológicos cuyo bloque principal es el microcontrolador. Permite integrar en un solo dispositivo los elementos necesarios para la realización de proyectos de investigación y desarrollo de diversos prototipos y dispositivos para la integración con el mundo físico. (GISI, 2018)

Algunos de los componentes clave de estas placas incluyen memoria, bus de comunicación, puertos de entrada y salida como unidad flash USB, Ethernet, pines de entrada y salida de uso general (que pueden ser digitales o analógicos) y, en algunos casos comunicaciones Wi-Fi. También pueden tener funciones especiales para la integración con sistemas de Internet de las Cosas e Industria 4.0. (GISI, 2018)

2.11.1 Arduino

Es una plataforma electrónica de fácil uso y manejo de código abierto basada en la integración de hardware y software utilizada a nivel mundial. El lenguaje de Arduino se utiliza para la programación y el IDE está basado en Processing. Esto significa que Arduino ofrece las bases para que cualquier persona pueda crear sus propias placas. Estas placas cuentan con conversores USB-UART llamado FTDI. (Arduino, 2018)

El tipo de periféricos que se puede utilizar para el envío de datos al microcontrolador depende en gran medida del uso que se pretenda dar, por ejemplo, teclados para introducir datos y diferentes tipos de sensores. También cuenta con una interfaz de salida, que se encarga de llevar la

información procesada a otros periféricos, que pueden ser pantallas y todo tipo de actuadores. (Arduino, 2018) En la Ilustración 2-21 se muestran los principales tipos de Arduino.

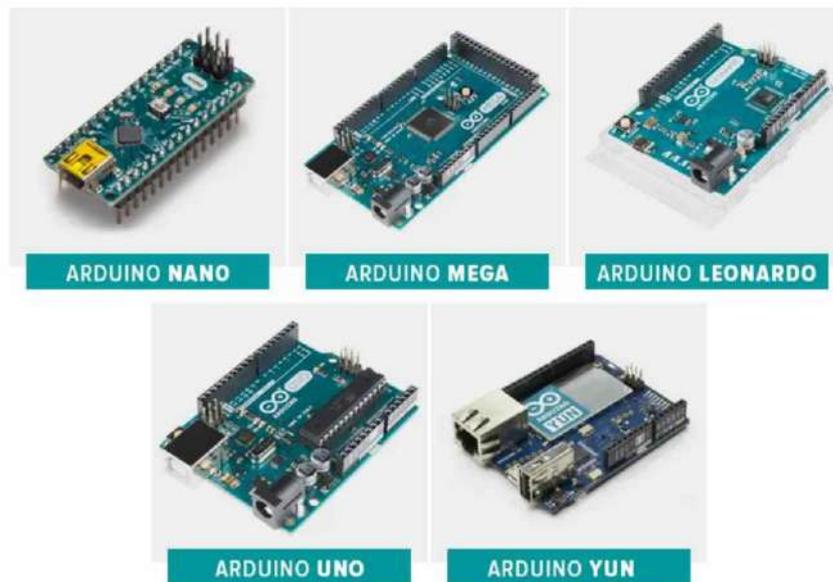


Ilustración 2-21: Tipos de Arduino

Fuente: (Arduino, 2018)

2.11.2 ESP32

Creado por Espressif Systems, es un sistema de bajo consumo y bajo costo en una serie de chips (SoC) con capacidades para la comunicación Wi-Fi y Bluetooth de modo dual. Equipados con amplificadores de potencia, filtros y módulos de administración de energía, logra un consumo de energía bajo a través de funciones de ahorro de energía que incluyen sincronización de reloj de resolución fina, múltiples modos de energía y escalado dinámico de energía. La ESP32 puede funcionar como un sistema completamente independiente o como esclavo de la MCU principal, reduciendo la carga en la pila de comunicación del procesador de aplicaciones principal y la eliminación automática de imperfecciones en los circuitos externos. (ESP32, 2016) En la Ilustración 2-22 se muestran los principales tipos de ESP32.



Ilustración 2-22: Tipos de ESP32

Fuente: (ESP32 2016)

2.11.3 Raspberry PI

Es una computadora de bajo costo y con un tamaño compacto, creada por Raspberry Pi Foundation. Cuenta con un sistema operativo Linux capaz de permitir la programación en lenguajes como Scratch y Python. Además, tiene la habilidad de interactuar en ambientes externos, por ello es usada en proyectos como estaciones meteorológicas y sistemas automáticos de bombeo de agua. Para conseguir que este ordenador sea lo más económico posible y al mismo tiempo tenga flexibilidad para poder ser modificado, la placa no incluye ni carcasa, disco duro, periféricos de entrada y salida lo que facilita su integración en cualquier proyecto. A pesar de ser un ordenador tiene algunas limitaciones al momento de intentar ejecutar programas que requieran un alto costo computacional. (Raspberry Pi, 2019) En la Ilustración 2-23 se muestran los principales tipos de Raspberry Pi.

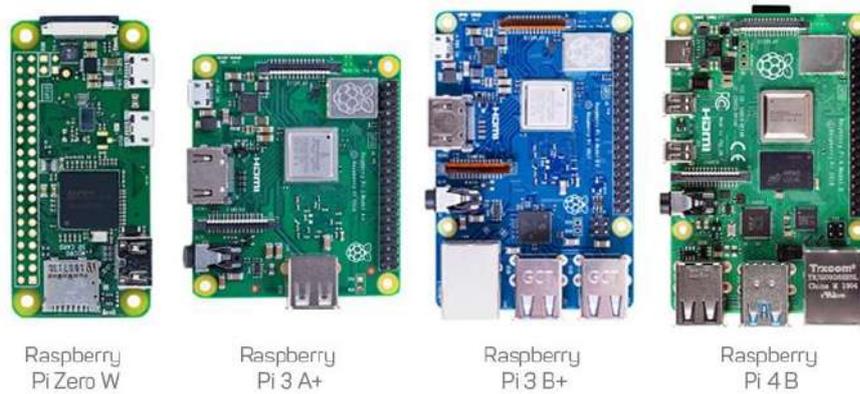


Ilustración 2-23: Tipos de Raspberry Pi

Fuente: (Raspberry Pi 2019)

2.11.4 Análisis comparativo de las tarjetas de desarrollo

En función del estudio realizado en la Tabla 2-4 se puede observar las tarjetas de desarrollo con mayor aceptación y utilidad del mercado, donde se detallan sus principales características como son el consumo de corriente, voltaje de operación, terminales analógicos y terminales digitales para PWM.

Tabla 2-4: Características de tarjetas de desarrollo.

Características	Arduino	ESP32	Raspberry Pi
Consumo de corriente	19 mA a 85mA	80mA a 225mA	140mA a 320mA
Voltaje de operación	2.7V a 5.5V	5V	5V
RAM	2Kb a 32Kb	96Kb	512Mb a 4Gb
Frecuencia	16MHz a 84MHz	80MHz a 160MHz	700MHz a 1.5GHz
Terminales digitales	14-54	17-36	40
Terminales PWM	6-15	16	4
Terminales analógicos	6-12	1-18	-
Compatibilidad Matlab	Si	Si	Si
Precio	11-70\$	8-40\$	30-470\$

Fuente: (Arduino, 2018) (ESP32, 2016) (Raspberry Pi, 2019)

Realizado por: Casierra Alvaro, 2024.

Esto permite elegir la mejor opción para la realización del proyecto técnico. En base a la Tabla 2-4 se define que Arduino reúne las características suficientes.

2.12 Servomotores

Son un tipo especial de motores que permiten controlar la posición específica del eje en un momento determinado. Está diseñado para moverse una cierta cantidad de grados y luego permanecer en una posición. Son dispositivos electromecánicos que constan de motores eléctricos, actuadores y paneles de control, contenidos en una carcasa generalmente de plástico. (Luna et al., 2016)

En la actualidad existen dos tipos de servomotores, los SDC (Servomotor Direct Current) y los SBLDC (Servomotor Brushless Direct Current). La diferencia entre estos radica en la distribución interna de los elementos que lo integran, ya que en los SDC los imanes permanentes se encuentran en la sección del estator y los electroimanes se encuentran integrados en el rotor junto con el anillo de conmutación, mientras que, en los SBLDC, los imanes permanentes se encuentran en el rotor y los electroimanes se encuentran en la sección del estator. (Martínez et al., 2017) En la Ilustración 2-24 se muestran los componentes del motor con escobillas y el BLDC.

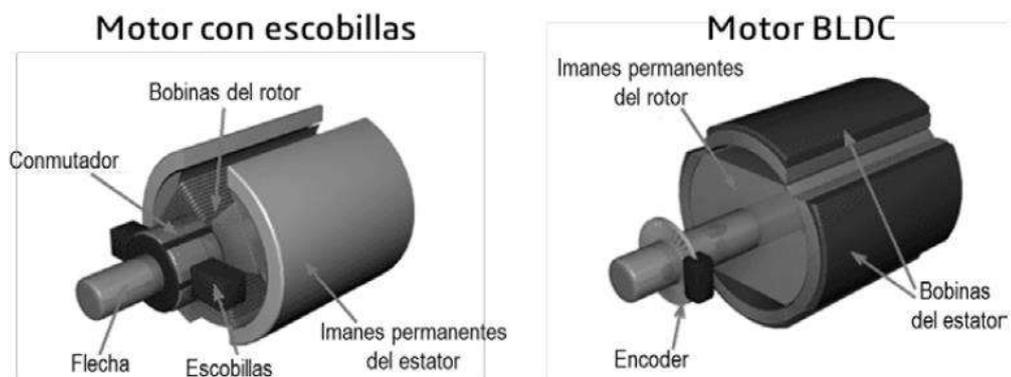


Ilustración 2-24: Componentes servomotores

Fuente: (Martínez et al., 2017)

Para lograr un rendimiento óptimo y una dinámica adecuada, los servomotores requieren la integración de sistemas electrónicos capaces de ajustar todas las variables durante el funcionamiento. Estos dispositivos se los conoce como controladores, los cuales están integrados por sensores, actuadores y dispositivos lógicos para determinar e implementar técnicas de control automático. (Martínez et al., 2017)

2.12.1 Micro servo MG90S

Es un servomotor de tamaño compacto ideal para proyectos de bajo torque. Cuenta con engranajes metálicos que permiten tener una mayor resistencia para el manejo de cargas. Este puede rotar 180° aproximadamente. Además, tiene una gran facilidad al momento de trabajar con diversas tarjetas de desarrollo como Arduino, ESP32 y Raspberry Pi. (Tower Pro, 2015) En la Tabla 2-5 se detallan las características del micro servo MG90S.

Tabla 2-5: Características principales micro servo MG90S.

Característica	MG90S
Peso	13.4g
Dimensión	22.8*12.2*28.5mm
Par de Parada	1.8kg/cm (4.8V); 2.2kg/cm (6.6V)
Velocidad funcionamiento	0.10sec/60° (4.8V); 0.08sec/60° (6V)
Ancho del pulso	600us y 2400us
Voltaje funcionamiento	4.8V
Rango temperatura	0°C – 55°C
Tipo conector	Universal tipo S

Fuente: (Tower Pro, 2015)

Realizado por: Casierra Alvaro, 2024.

En la Ilustración 2-25 se observa un Microservo MG90S.

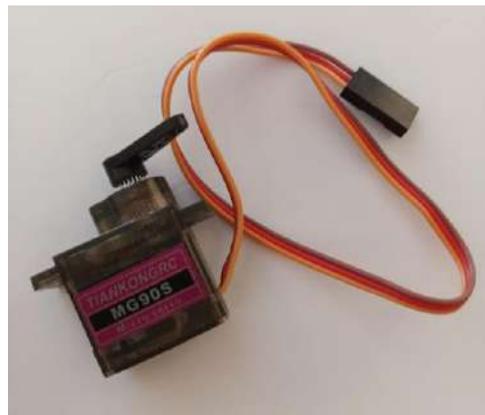


Ilustración 2-25: Microservo MG90S

Realizado por: Casierra Alvaro, 2024.

2.13 Procesamiento de señales

Para el procesamiento se utilizan algoritmos que se permiten analizar, transformar e interpretar las señales provenientes de los sensores. Las señales pueden ser de audio, video, eléctricas, biomédicas, etc. Dos de los softwares más utilizados y potentes para realizar el procesamiento en tiempo real son lo que se mencionan a continuación:

2.13.1 Matlab

Es una plataforma de programación y computación numérica utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos. Combina un entorno gráfico optimizado para procesos iterativos de análisis y diseño con un lenguaje de programación que permite cálculos matriciales. Matlab cuenta con una herramienta llamada SIMULINK, el cual, es un entorno de programación usando diagramas de bloques. Esta permite diseñar sistemas con modelos multidominio sin necesidad de escribir código de manera directa. (The MathWorks, 2018)

Dentro de Matlab se pueden desarrollar GUIs (interfaces gráficas de usuario) que permiten un control sencillo de las aplicaciones de software, lo cual elimina la necesidad de escribir comandos para ejecutar una aplicación. Generalmente las GUI incluyen controles tales como menús, barras de herramientas, botones y controles deslizantes. Existen también extensiones de Matlab llamadas Toolboxes, que son colecciones de funciones que permiten resolver problemas específicos en diversas áreas de ciencia e ingeniería, como por ejemplo en el área de control, redes neuronales, lógica difusa, procesamiento de señales, etc. (The MathWorks, 2018) En la Ilustración 2-26 se muestra un ejemplo de GUI con Matlab.

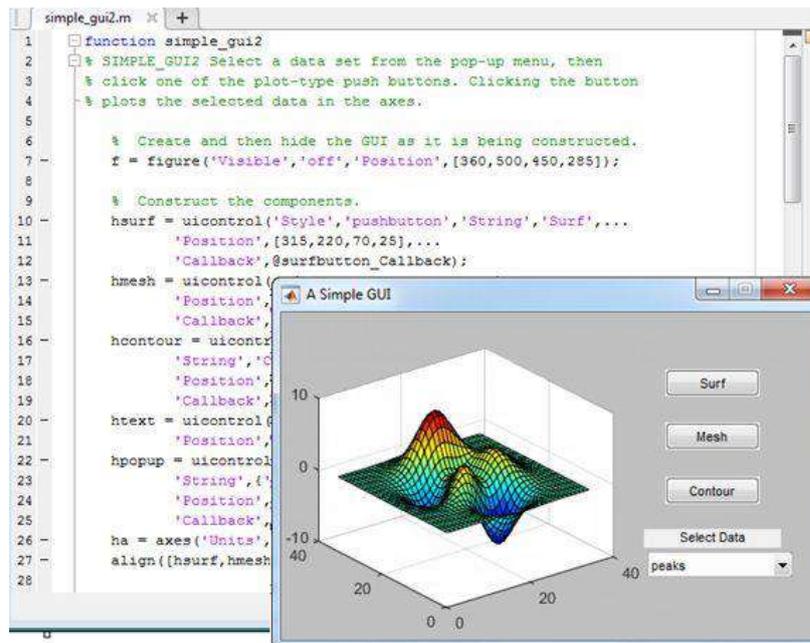


Ilustración 2-26: GUI con Matlab

Fuente: (The MathWorks, 2018)

2.13.2 Python

Es un lenguaje de programación potente y eficiente, con una estructura de datos eficiente de alto nivel, una sintaxis simple y un sistema de programación orientado a objetos. La sintaxis y la escritura dinámica de Python lo convierten en un lenguaje ideal para crear secuencias de comandos y crear rápidamente aplicaciones para muchos campos como las aplicaciones web, desarrollo de software, ciencia de datos, procesamiento de datos y machine learning. (Python Software Foundation, 2001)

Dentro de Python existen bibliotecas que son colecciones de códigos usados con frecuencia para optimizar el código. También cuentan con marcos que son una colección de paquetes y módulos, que son conjuntos de código relacionados que se usan generalmente para crear aplicaciones de forma más rápida. (Python Software Foundation, 2001) En la Ilustración 2-27 se muestra un esquema de las GUI en Python.



Ilustración 2-27: GUI con Python

Fuente: (Python Software Foundation, 2001)

CAPÍTULO III

3. MARCO METODOLÓGICO

En el presente capítulo se realiza un análisis detallado de los requerimientos que debe cumplir el sistema bioelectrónico en lo referente al diseño, funcionamiento, concepción, bloques de operación, componentes necesarios para su implementación hardware, así como los diferentes algoritmos implementados y diagramas de flujo del software desarrollado incluyendo los diagramas de conexión de los sensores y actuadores.

3.1 Requerimientos del sistema bioelectrónico

En función de la investigación realizada en el capítulo anterior, se plantearon los requerimientos necesarios para el diseño e implementación de un sistema bioelectrónico para el entrenamiento de prótesis mioeléctricas en personas con amputación transradial, los mismos que se describen a continuación:

- El sistema debe ser de fácil instalación, amigable, intuitivo y no ser invasivo para el uso de distintos usuarios.
- El sistema debe obtener las señales mioeléctricas de personas de todos los géneros, con diferentes texturas físicas y en rango de edad variable.
- Desarrollar una interfaz gráfica para la comunicación entre los sensores y los actuadores del sistema.
- Desarrollar un algoritmo de control que permita diferenciar varios tipos de agarre para la activación de una prótesis.
- La interfaz debe ser capaz de monitorear las señales EMG provenientes del sensor mioeléctrico.
- El dispositivo externo debe medir diferentes niveles de presión de acuerdo con los objetos que manipule.
- Escribir un protocolo general para el uso correcto del sistema.

3.2 Concepción general del sistema bioelectrónico

La concepción del sistema bioelectrónico se describe en tres bloques generales, que especifican el funcionamiento, como se puede observar en la Ilustración 3-1.

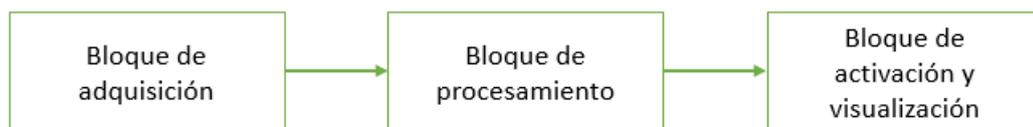


Ilustración 3-1: Diagrama de bloques del sistema

Realizado por: Casierra Alvaro, 2024.

Las señales mioeléctricas obtenidas en tiempo real mediante el sensor Myo Armband son enviadas a computador a través de comunicación inalámbrica Bluetooth, para posteriormente ser procesadas y utilizadas como señales de activación para el funcionamiento correcto del sistema. Las señales de los sensores de presión son adquiridas por la tarjeta de desarrollo y en función de las señales de activación provenientes del sensor se realiza el procesamiento para ejecutar el algoritmo de control pertinente que permita activar los actuadores de manera oportuna y eficiente para realizar el agarre de diferentes objetos.

El sistema realiza múltiples comunicaciones de tipo inalámbrica y serial entre los elementos software y hardware. La primera es entre el sensor mioeléctrico y el computador a través de su aplicación propietaria Myo Connect de manera inalámbrica mediante Bluetooth BLE 4.0 para su reconocimiento y calibración, para posteriormente utilizar dicha comunicación con la interfaz gráfica GUI. La segunda es de tipo Serial entre la tarjeta de desarrollo y la GUI. Una vez obtenida la señal de activación con el sensor, entran en funcionamiento los algoritmos de control preestablecidos para la activación de los actuadores en función de las señales de retroalimentación para generar un agarre o directamente establecer el puño.

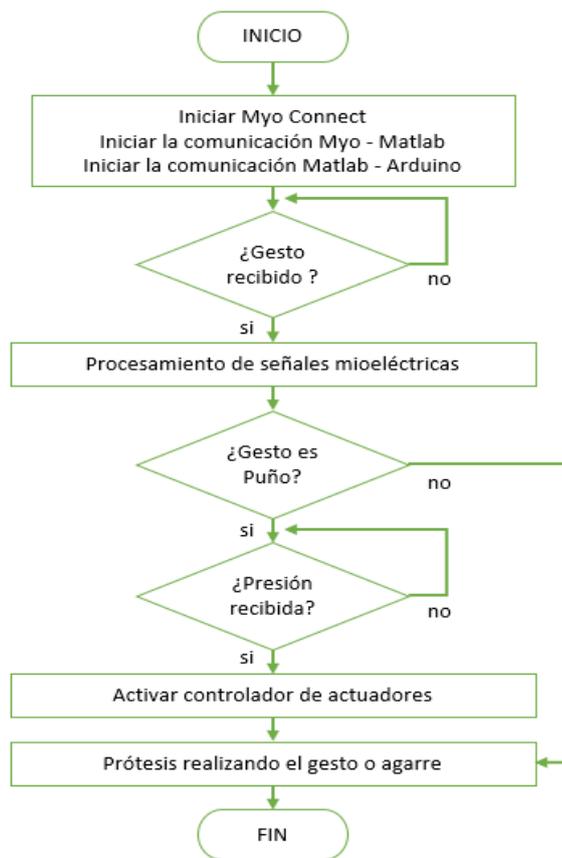


Ilustración 3-2: Diagrama de flujo del sistema bioelectrónico

Realizado por: Casierra Alvaro, 2024.

3.3 Etapa de adquisición

Está compuesto por cuatro sensores: uno mioeléctrico y los otros tres de presión de tipo resistivo. Estos sensores permitirán obtener la información necesaria, relacionado con las señales mioeléctricas, como la señal de activación para mantener una postura neutral cuando el paciente este realizando el gesto de palma afuera, palma adentro y reposo, y la señal de activación del algoritmo de control cuando el paciente este realizando el gesto de puño que permitirá activar los actuadores cuando se detecte un objeto que no permita cerrar el puño, generando así el agarre de dicho objeto.

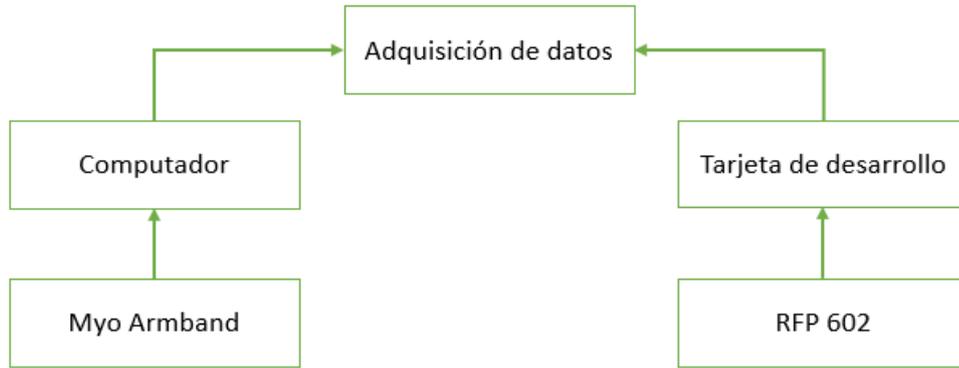


Ilustración 3-3: Diagrama de bloques de la etapa de adquisición

Realizado por: Casierra Alvaro, 2024.

3.3.1 *Comunicación entre Myo Armband y el computador*

El brazalete Myo Armband es el encargado de enviar o transmitir la información obtenida en el antebrazo o brazo del paciente hacia el computador para ser recibido por la aplicación Myo Connect y posteriormente mediante el software de procesamiento de señales desarrollado en Matlab. Este envío de información se realiza por el protocolo de comunicación inalámbrica Bluetooth BLE 4.0, siendo este una versión de bajo consumo de energía y baja latencia. Dentro de las principales características del protocolo de comunicación tenemos las siguientes:

- Frecuencia de trabajo 2.4 GHz.
- Velocidad de transmisión de datos de 24 Mbps.
- Distancia va de 10 a 100 metros.
- El consumo de corriente pico menor a 15mA.

En la Ilustración 3-4 se puede observar la comunicación del brazalete Myo Armband con el computador para la adquisición de las señales mioeléctricas de activación proporcionada por la EMG.



Ilustración 3-4: Comunicación Myo Armband-Computador

Realizado por: Casierra Alvaro, 2024.

3.3.2 Comunicación entre RFP 602 y Arduino UNO

El sensor de presión es el encargado de dar la retroalimentación en caso de detectarse el gesto de puño. Esta señal se envía a la tarjeta de desarrollo Arduino UNO mediante las entradas analógica, las entradas analógicas. Estas entradas se caracterizan por leer valores dentro del rango de 0 a 5 V, con una resolución de 1024 (10 bits), lo que supone una precisión de la medición de $\pm 2.44\text{mV}$. El sensor de presión debe conectarse con una configuración de divisor de voltaje, ubicado como segundo elemento junto a una resistencia con valores en $\text{k}\Omega$ que permita obtener un rango total en la lectura de la señal de salida.

En la Ilustración 3-5 se observa la conexión del sensor de presión a través de un divisor de voltaje a las entradas analógicas de la tarjeta de desarrollo Arduino UNO.

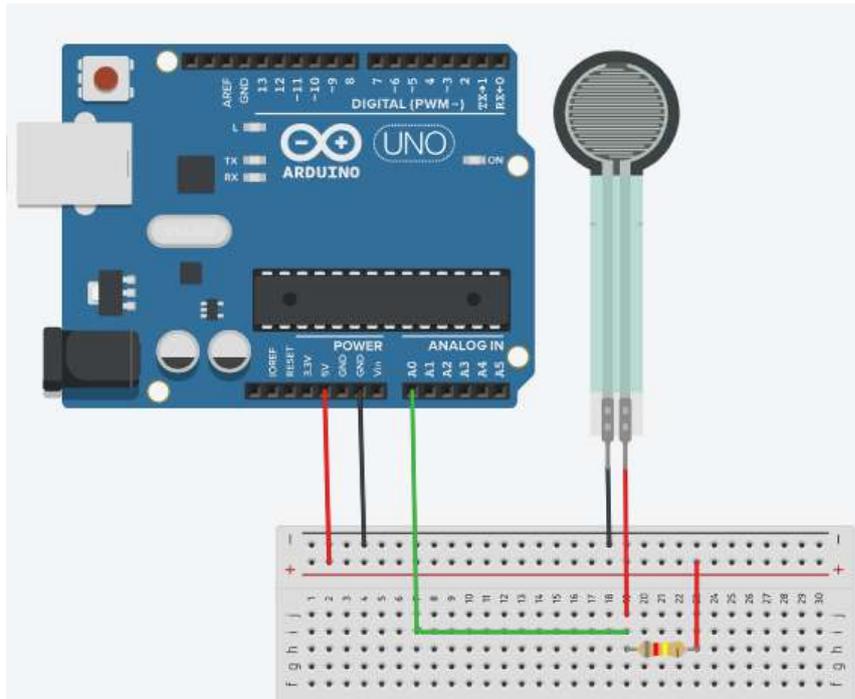


Ilustración 3-5: Comunicación RFP 602-Arduino UNO

Realizado por: Casierra Alvaro, 2024.

3.4 Etapa de procesamiento

Se hace uso del software Matlab para el procesamiento de las señales y la identificación de los diferentes gestos como puño, palma adentro, palma afuera, reposo, dedos expandidos y doble clic. Con los gestos identificados se realizan tres acciones puntuales, las cuáles son el envío de la señal de activación del algoritmo de control cuando se identifique el gesto puño, envío de la señal de activación para mantener una posición neutral cuando se detecten los gestos de palma adentro, palma afuera y reposo y cancelación del envío de señales de activación cuando se identifique el gesto de dedos extendidos.



Ilustración 3-6: Diagrama de bloques de la etapa de procesamiento

Realizado por: Casierra Alvaro, 2024.

3.4.1 Identificación del gesto

Dado que los datos más recientes de Myo se almacenan en el objeto MyoData, para realizar la identificación de gestos, se utilizan las propiedades de datos muestreados en la base de tiempo EMG (200 Hz).

```
m1.timeEMG
m1.emg
m1.pose
m1.pose_rest
m1.pose_fist
m1.pose_wave_in
m1.pose_wave_out
m1.pose_fingers_spread
m1.pose_double_tap
```

Ilustración 3-7: Propiedades de los datos muestreados en la base de tiempo EMG

Realizado por: Casierra Alvaro, 2024.

En la Ilustración 3-8 podemos observar el diagrama de flujo que corresponde al proceso de identificación de gestos a través de Matlab.

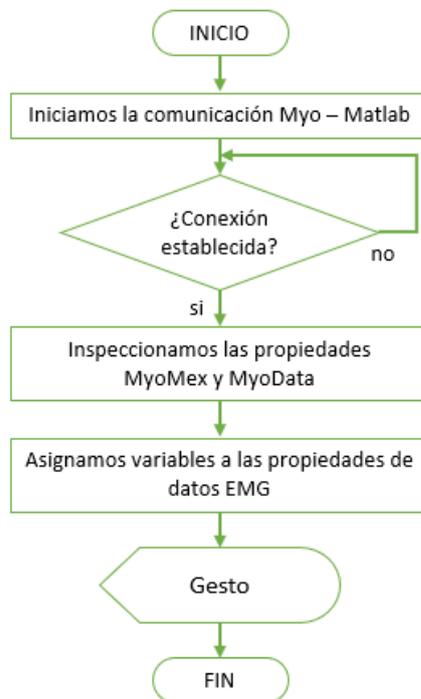


Ilustración 3-8: Diagrama de flujo de la identificación de gesto

Realizado por: Casierra Alvaro, 2024.

3.5 Etapa de activación y visualización

Una vez adquiridos los datos e identificado el gesto correspondiente, se procede a realizar la activación controlada de los actuadores y la visualización las señales EMG de la instrumentación del sistema bioelectrónico. El proceso se muestra en la Ilustración 3-9.

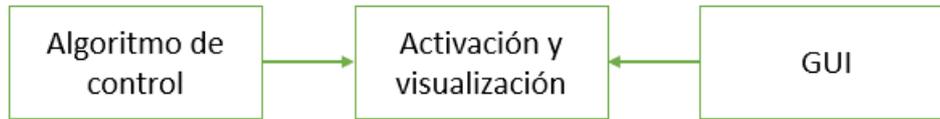


Ilustración 3-9: Diagrama de bloques de la etapa de activación y visualización

Realizado por: Casierra Alvaro, 2024.

3.5.1 Algoritmo de control

Para realizar la activación de los actuadores cuando el gesto sea el de puño, se ha diseñado un controlador tipo PID, donde se estableció una referencia que permita a la prótesis mantener, sujetar o agarrar diferentes objetos con firmeza sin llegar a deformar o dañar dicho objeto, y usando como retroalimentación la señal de los sensores de presión. Si el gesto recibido no es puño, no se utiliza el controlador PID.

3.5.2 GUI

Para la adquisición y procesamiento de las señales EMG recibidas por el brazalete Myo Armband es necesario una GUI que permita llevar un control y monitoreo de todo el proceso. La GUI se implementa en Matlab con la ayuda del entorno de desarrollo interactivo APP DESIGNER.

3.6 Elementos Hardware del sistema bioelectrónico

En este apartado se describen los elementos con sus respectivas características, que hacen parte del hardware para la construcción del sistema bioelectrónico.

3.6.1 Myo Armband

A través del control de los músculos de los brazos, el brazalete Myo Armband puede controlar un gran número de dispositivos electrónicos con el movimiento del brazo o realizando diferentes gestos preestablecidos como puede ser apretando el puño. Todos los sensores son de tipo no invasivo y están colocados en una banda flexible de plástico, dicha banda tiene que estar lo suficientemente ajustada para poder reconocer los gestos de manera eficiente, características principales por lo cual fue elegido para su uso en el desarrollo del sistema bioelectrónico. No dispone de botones físicos y se apaga una vez retirada del brazo o desde la aplicación Myo Connect.

Las especificaciones técnicas se describen a continuación:

- El tamaño del brazo es ampliable entre 19 y 34 cm (7,5 y 13 pulgadas) de circunferencia del antebrazo.
- Pesa 93 gramos.
- Talla única, a partir de 12 años.
- 8 sensores EMG (electromiografía) de acero inoxidable de calidad médica.
- 1 IMU de nueve ejes con giroscopio de tres ejes, acelerómetro de tres ejes y magnetómetro de tres ejes.
- Doble indicador LED.
- Procesador ARM Cortex M4.
- Feedback háptico con vibraciones cortas, medias y largas.
- Utiliza Bluetooth Smart.
- Carga Micro-USB.
- Batería de iones de litio recargable integrada.
- Un día completo de uso con una sola carga.



Ilustración 3-10: Partes del Brazaletes Myo Armband

Realizado por: Casierra Alvaro, 2024.

Composición del Myo Armband

Cuando se desmonta un brazalete Myo Armband se puede observar en su interior con detalle los elementos que lo componen como se muestra en la Ilustración 3-11. Como elementos principales se pueden observar las unidades de electrodos EMG integrado en el interior del recubrimiento de plástico, también se observa la placa base y las dos baterías de Litio. (Stern, 2016)

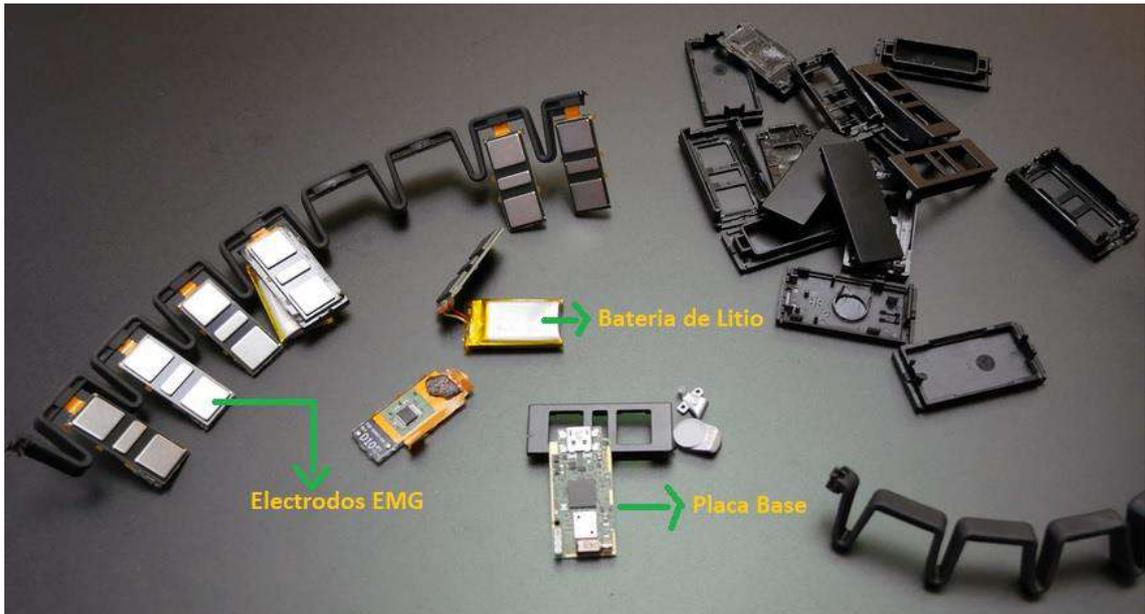


Ilustración 3-11: Elementos principales del brazalete Myo Armband

Realizado por: Casierra Alvaro, 2024.

La placa base del brazalete está conformada por el microcontrolador, el sensor IMU y el módulo Bluetooth (BLE), en la misma placa se encuentra un motor de vibración unido a la placa de la batería como se observa en la Ilustración 3-12. El chip IMU es un modelo de 9 ejes que contiene un procesador digital de movimiento a bordo. Adicionalmente el MPU-9150 cuenta con un magnetómetro, un giroscopio y un acelerómetro, todos de 3 ejes. (Stern, 2016)

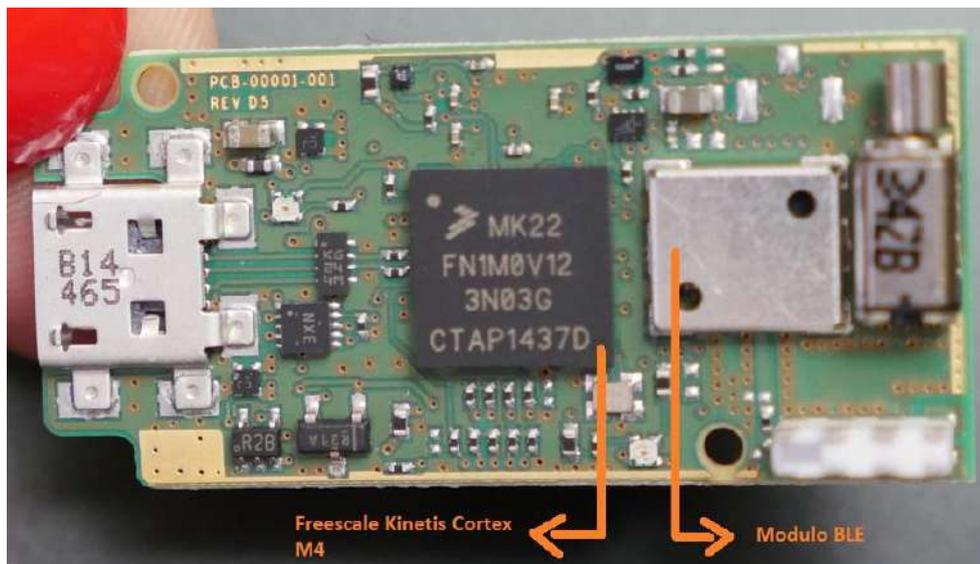


Ilustración 3-12: Elementos ubicados en la placa base del Myo Armband

Realizado por: Casierra Alvaro, 2024.

Posicionamiento del dispositivo

Cuando se obtienen datos con la pulsera Myo, es importante ubicarla correctamente en el brazo. Dentro de lo posible, el dispositivo siempre debe colocarse en la misma ubicación y orientación. Esto asegura una mayor fiabilidad de los datos obtenidos, incluso cuando se vaya a tomar datos de manera repetida se recomienda hacerlo durante la misma sesión, caso contrario tocará realizar la calibración del brazalete repetidamente. En las Ilustraciones 3-13 y 3-14 se observan dos posiciones correctas para el brazalete Myo.



Ilustración 3-13: Posición Myo Armband arriba del codo

Realizado por: Casierra Alvaro, 2024.



Ilustración 3-14: Posición Myo Armband debajo del codo

Realizado por: Casierra Alvaro, 2024.

Se recomienda colocar el electrodo principal de la pulsera Myo entre los músculos braquiorradial y extensor radial de la muñeca, siendo el electrodo principal el que tiene el logo y la luz LED.

3.6.2 *Arduino UNO*

El poder y la eficiencia de Arduino proviene de que es una plataforma de código abierto. Es posiblemente la placa más utilizada y robusta, basada en el microcontrolador ATmega328P, mediante la cual se realiza la adquisición de las señales analógicas de los sensores de presión y la activación de los servomotores por los pines digitales, la distribución de los pines se describe en la placa Arduino UNO se describen a continuación:

Tabla 3-1: Distribución de pines en Arduino UNO.

Pines Arduino UNO	Descripción
5 V	Alimentación positiva de sensores y actuadores.
GND	Alimentación negativa de sensores y actuadores.
Pin Analógico A0	Lectura analógica del sensor de presión 1
Pin Analógico A1	Lectura analógica del sensor de presión 2
Pin Analógico A2	Lectura analógica del sensor de presión 3
Pin Digital 2	Señal PWM de control de servomotor 1
Pin Digital 3	Señal PWM de control de servomotor 2
Pin Digital 4	Señal PWM de control de servomotor 3
Pin Digital 5	Señal PWM de control de servomotor 4
Pin Digital 6	Señal PWM de control de servomotor 5
Conector Power	Alimentación externa
Puerto USB	Puerto de programación

Realizado por: Casierra Alvaro, 2024.

La placa Arduino UNO puede alimentarse a través de la conexión USB o con fuentes externas. El adaptador se puede conectar al enchufar un conector 2.1 mm de alimentación de la placa, con centro positivo. También puede conectarse una batería insertando en los pines GND y Vin del conector POWER. El rango recomendado de alimentación es entre 7 y 12 V, ya que si se usa un valor superior el regulador puede sobrecalentarse y posteriormente dañar la placa.

3.6.3 RFP 602

Debido a las características que posee este sensor descritas en la Tabla 2-3, entre las que destacan sus dimensiones de área de trabajo y su rango de presión admitido, se lo ha montado directamente en la prótesis. Este es el encargado de realizar la retroalimentación necesaria para el control y activación adecuada de la prótesis. En la Ilustración 3-15 se puede ver su colocación en la prótesis mioeléctrica.



Ilustración 3-15: Sensores RFP 602 en Prótesis

Realizado por: Casierra Alvaro, 2024.

3.6.4 Esquema de conexión del sistema bioelectrónico

En la Ilustración 3-16 se presenta el esquema de conexión de los componentes físicos del sistema bioelectrónico, compuestos por los sensores de presión y los servomotores.

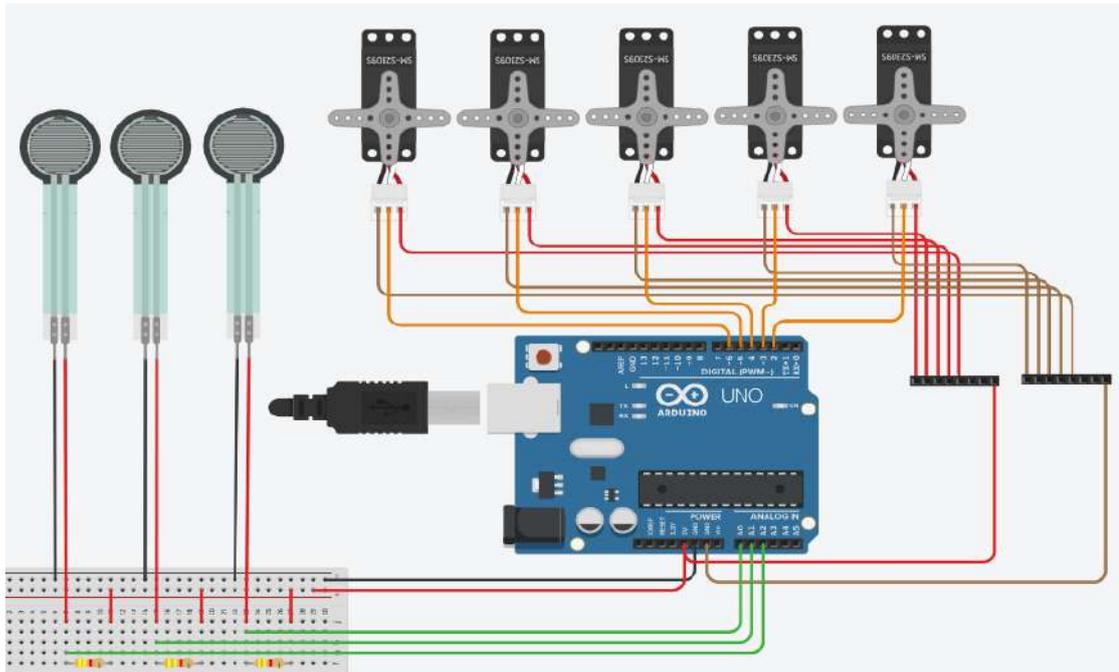


Ilustración 3-16: Esquema de conexión del sistema bioelectrónico

Realizado por: Casierra Alvaro, 2024.

3.7 Software para el desarrollo del sistema bioelectrónico

En esta sección se va a describir la parte software con la cual se ha desarrollado el sistema bioelectrónico, los distintos entornos de programación, librerías y toolbox necesarios para la adquisición de datos, procesamiento y activación de actuadores.

3.7.1 *Myo Connect Installer*

Este es un software propietario responsable de realizar diversas operaciones básicas de comunicación entre la banda Myo y la computadora y tiene varias opciones de configuración, que incluyen:

- Botón de encendido y apagado.
- Visualización de estado de la conexión.
- Visualización de nivel de batería.
- Prueba de conexión ping con señal de vibración.
- Calibración del brazalete.
- Reconocimiento de los principales gestos a través de señales electromiográficas.
- Manejo del computador simulando el ratón.
- Control de principales plataformas de entretenimiento.

Cuando se realiza la instalación del Myo Connect en el computador, se mostrará un icono de brazaletes en los iconos ocultos, ubicado en la barra de tareas, desde donde se puede acceder a las configuraciones básicas antes descritas como se puede observar en la Ilustración 3-17, que son indispensables para el correcto funcionamiento del Myo Armband.

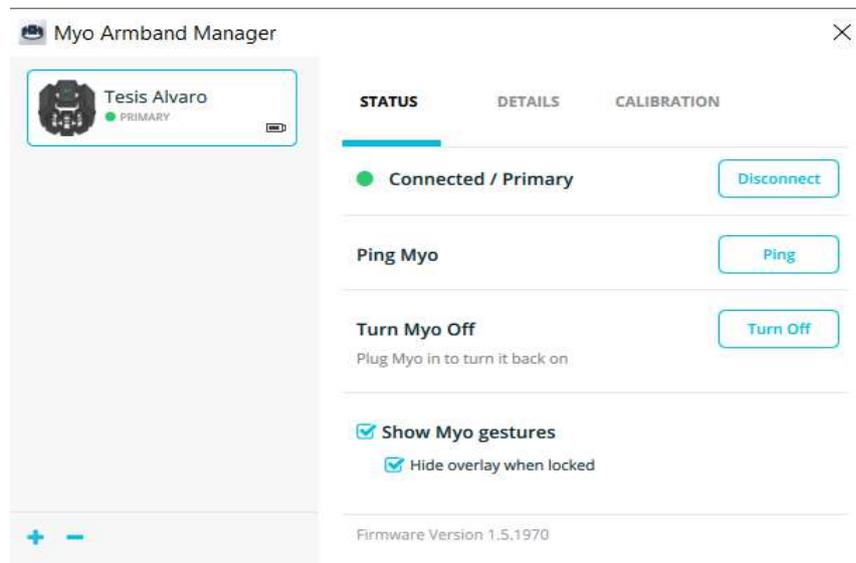


Ilustración 3-17: Ventana principal de Myo Connect

Realizado por: Casierra Alvaro, 2024.

3.7.2 *Matlab*

Las principales características del software Matlab son las siguientes:

- Está disponible para varios sistemas operativos como Windows, Apple, Unix, etc.
- Cuenta con un lenguaje de alto nivel enfocado en aplicaciones de ingeniería.
- Cuenta con herramientas para la creación de gráficos que permiten visualizar datos en tiempo real.
- Tiene disponibles herramientas para la creación de aplicaciones con interfaces de usuario (GUI).

El algoritmo para la adquisición de señales mioeléctricas se lo realizó en Matlab, debido a que ofrece una facilidad del manejo de las señales antes mencionadas gracias a las librerías con las que dispone para su comunicación con el brazaletes Myo Armband.

3.7.2.1 *Myo SDK y MyoMex Master*

El desarrollador del brazaletes Myo Armband Thalmics Labs dispone de SDKs (Software Development Kit), compatible con varias plataformas de desarrollo de aplicaciones para crear comunicación entre dispositivos, Myo Armband cuenta con una IMU con 9 grados de libertad y

8 sensores de electromiografía, donde el SDK permitirá la transmisión de datos desde múltiples dispositivos. (Andagama, 2022)

La biblioteca MyoMex es responsable de brindar a los usuarios del software de procesamiento de señales Matlab, en términos generales la capacidad de transmitir datos desde la pulsera hasta 50 Hz a la IMU y 200 Hz al sensor EMG mediante programación o línea de comando. (Andagama, 2022)

Hay que tener en cuenta dos consideraciones para poder hacer uso de estas herramientas y que no se genere algún error al momento de compilar o ejecutar los programas, las cuáles son:

- Las carpetas de MyoMex-Master y myo-sdk-win-0.9.0 deben estar en la unidad de disco C: o bien dentro de archivos de programa.
- Dentro de Matlab agregar “C:\Program Files\myo-sdk-win-0.9.0\myo-sdk-win-0.9.0\bin” a la current folder o directorio de trabajo actual.
- Dentro de Matlab agregar las carpetas MyoMex-Master y myo-sdk-win-0.9.0 y sus subcarpetas al path mediante la opción “Set Path”.

En la Ilustración 3-18 podemos ver la ubicación de las carpetas que contienen los archivos necesarios dentro de archivos de programa.

Por otro lado, en la Ilustración 3-19 se muestra la agregación de los archivos ejecutables necesarios en la current folder en el software de procesamiento de señales.

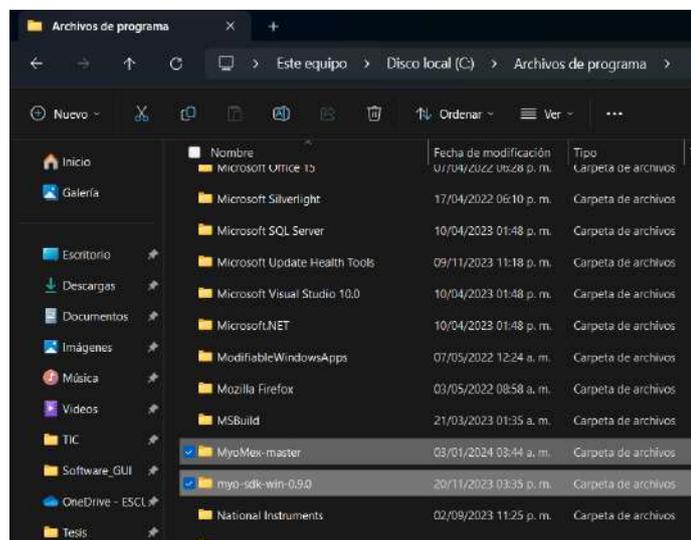


Ilustración 3-18: Ubicación de MyoMex-master y myo-sdk-win

Realizado por: Casierra Alvaro, 2024.

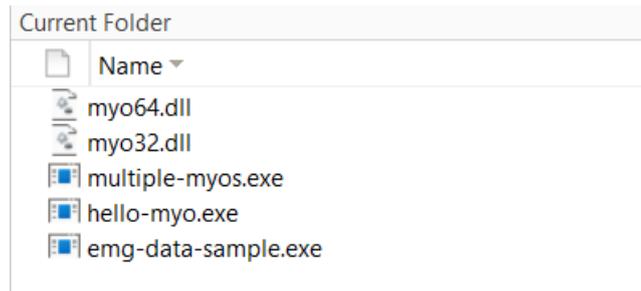


Ilustración 3-19: Archivos ejecutables en el directorio de trabajo actual

Realizado por: Casierra Alvaro, 2024.

3.7.2.2 *MATLAB Support for MinGW-w64 C/C++ Compiler*

Son un conjunto de compiladores para Windows que se basa en una cadena de herramientas GNU. Las aplicaciones C y C++ compiladas con MinGW-w64 se pueden llamar desde Matlab usando MEX.

Para hacer uso de este compilador desde Matlab, es necesario instalarlo, desde el Home de Matlab en la pestaña Add-Ons se accede al buscador, ingresamos el nombre del compilador y descargar el mismo. Una vez descargado se agrega al PATH la ubicación del compilador y con doble clic procedemos a instalarlo. En la Ilustración 3-20 se observa el compilador correctamente instalado.



Ilustración 3-20: Compilador MinGW-64 instalado

Realizado por: Casierra Alvaro, 2024.

3.7.2.3 *MATLAB Support Package for Arduino Hardware*

Para utilizar esta biblioteca con Matlab, debes instalarla; desde Matlab Home, en la pestaña Add-Ons, ir al buscador, introducir el nombre de la biblioteca y descargar. Una vez descargado, se agrega la ubicación de la biblioteca al PATH y con doble clic para instalarla. En la Ilustración 3-21 se observa la librería correctamente instalada.

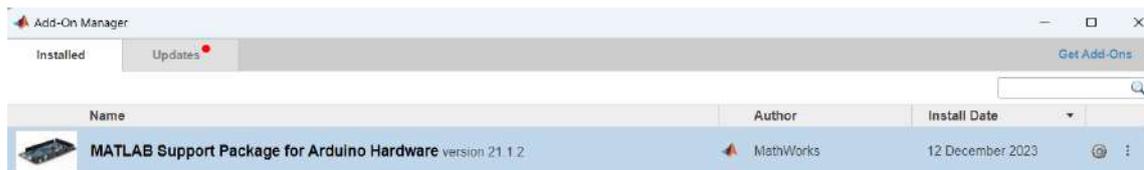


Ilustración 3-21: Librería para Arduino instalada

Realizado por: Casierra Alvaro, 2024.

Una vez instalada la librería podemos iniciar la comunicación serial, especificando el puerto serial, la velocidad de transmisión y habilitando el puerto como se puede observar en la Ilustración 3-22.

```
global t
app.comu = serialport('COM3',115200);
tic;
```

Ilustración 3-22: Código para iniciar la comunicación serial

Realizado por: Casierra Alvaro, 2024.

3.7.3 Arduino

Las librerías son colecciones de código que facilitan la interconexión de sensores, pantallas, módulos electrónicos, etc.

3.7.3.1 Arduino IDE

Es el entorno que permite a los desarrolladores importar las librerías necesarias para el desarrollo del sistema bioelectrónico como son la librería de PID_v1, Servo y SoftwareSerial, además de generar el código que permita realizar la comunicación con la interfaz gráfica de usuario GUI y adquirir las señales, procesarlas y posteriormente controlar los actuadores de una manera sencilla, debido a que su lenguaje y su entorno se basan en Wiring, siendo este compatible con varios sistemas operativos. La versión utilizada para la creación del código del sistema bioelectrónico es 2.2.1.

3.7.3.2 Librería PID by Brett Beauregard

La biblioteca PID de Brett Beauregard es una biblioteca de software desarrollada para la plataforma Arduino que facilita la implementación del control proporcional-integral-derivativo (PID). Probablemente la librería más utilizada para control PID en Arduino, tiene una explicación sencilla y efectiva, lo que lo hace relativamente conveniente para realizar proyectos utilizando

controladores. Dispone de protección anti-windup, la cual permite especificar la acción de control y a partir de la versión 1.2 incluyó la posibilidad de elegir entre las ecuaciones PI-D e I-PD indicando los parámetros P_ON_E y P_ON_M respectivamente.

3.7.3.3 Programación de Arduino

Para el desarrollo del sistema bioelectrónico se realiza una programación en lenguaje C++ en el IDE de Arduino, para lo cual se define el procedimiento del diagrama de flujo mostrado en la Ilustración 3-23.

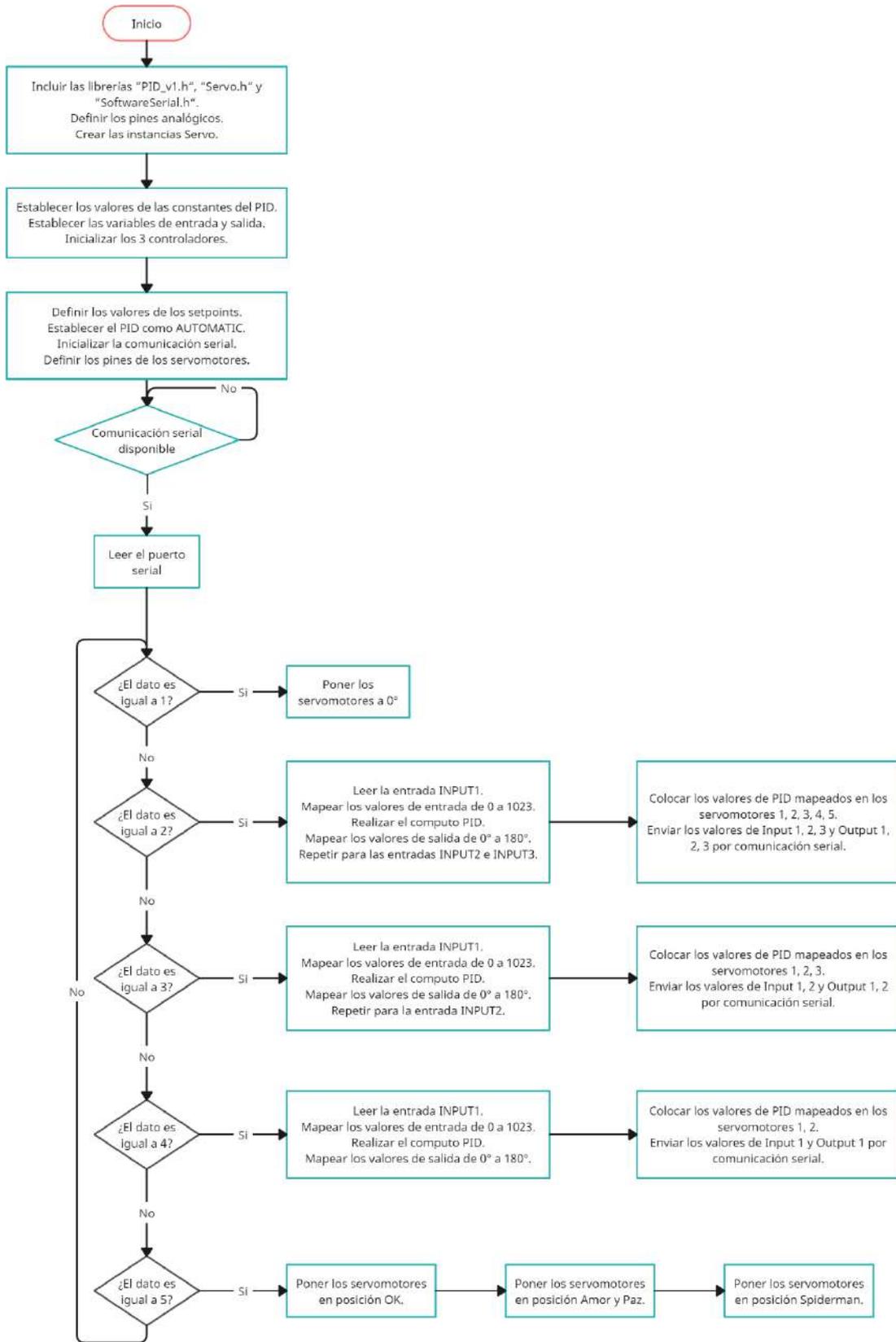


Ilustración 3-23: Diagrama de flujo Arduino

Realizado por: Casierra Alvaro, 2024.

3.8 Desarrollo de la GUI

La interfaz gráfica de usuario es un programa informático que utiliza imágenes y objetos gráficos que permiten visualizar diversas acciones e información pertinente. La función principal de la interfaz es proporcionar un entorno visual simple que permita que dos o más sistemas que utilizan diferentes lenguajes de programación interactúen entre sí.

Cuando se diseñan interfaces gráficas de usuarios se sigue habitualmente el siguiente procedimiento:

- 1) Comprender y definir el entorno de uso. – Conocer las especificaciones técnicas del entorno en el que se va a utilizar la interfaz de forma que se pueda desarrollar una aplicación que se adapte a dichas condiciones.
- 2) Identificar las necesidades de los usuarios. – Definir junto con el usuario el objetivo principal de la aplicación, así como todas las funciones adicionales que debe incorporar.
- 3) Generar soluciones de diseño. – En esta etapa es importante recordar que los elementos que componen la interfaz están ordenados según su importancia en áreas de mayor o menor jerarquía visual, y todos deben ser accesibles. Otro aspecto importante es la creación de una interfaz intuitiva que resulte atractiva para los usuarios.
- 4) Evaluar el diseño en función de las necesidades. – Analiza las diferentes opciones y elige la que mejor se adapta a tus necesidades, tanto a nivel de prestaciones como de diseño.
- 5) Implementar el diseño elegido. – Desarrollar el código de la alternativa establecida como válida. Es importante evaluar de forma periódica todos los elementos para saber si continúan funcionando correctamente.
- 6) Comprobar el funcionamiento. – Esta es una de las etapas más importantes del proyecto y en ella se realiza un recorrido a lo largo de la aplicación con el objetivo de identificar posibles errores y problemas de diseño.
- 7) Entregar el producto. – Una vez comprobado que el funcionamiento de la interfaz es el correcto, se procede a la entrega de esta. Esta parte es muy importante porque el primer contacto determinará en gran medida el sentimiento del usuario sobre la aplicación.

En la Ilustración 3-24 se muestra la secuencia de funcionamiento de la interfaz gráfica de usuario GUI.



Ilustración 3-24: Secuencia de funcionamiento de la GUI

Realizado por: Casierra Alvaro, 2024.

3.8.1 *App Designer*

Es un entorno gráfico de desarrollo de interfaces que incluye una versión totalmente integrada del editor de MATLAB, motivo por el cual se ha utilizado para el desarrollo de la GUI del sistema bioelectrónico. Las vistas de diseño y código están estrechamente vinculadas, de modo que los cambios que se realizan en una afectan inmediatamente a la otra.

Esta herramienta tiene como objetivo reducir el tiempo necesario para desarrollar interfaces, así como simplificar su diseño al facilitar el acceso a las propiedades de los componentes, las declaraciones de devolución de llamadas y cómo intercambiar información entre ellos.

Los principales componentes usados para el desarrollo de nuestra interfaz se muestran en la Tabla 3-2:

Tabla 3-2: Componentes usados en la App Designer.

Componente	Descripción
 <p>Axes Display graphical data visualization</p>	<p>El objeto UIAxes se usa para representar datos en un diagrama de líneas 2D o en un diagrama de dispersión.</p>
 <p>Button Execute command when pressed</p>	<p>Los objetos Button ejecutan una callback cuando el usuario pulsa sobre ellos.</p>
 <p>Date Picker Select and display date in specified format</p>	<p>El objeto Date Picker nos permite visualizar la fecha en la cual se está ejecutando la aplicación en diferentes formatos.</p>
 <p>Image Display icon or logo</p>	<p>Image nos permite mostrar diferentes imágenes como pueden ser logos o componentes para una mejor interpretación.</p>
 <p>Label Display text to describe component</p>	<p>Las labels se usan para transmitir información, como el nombre de las variables que se van a medir o las unidades de estas, al usuario.</p>
 <p>Text Area Enter multiline text</p>	<p>Text Area permite introducir datos de tipo texto en una o en varias líneas, respectivamente.</p>
 <p>Panel Group components</p>	<p>El Objeto panel agrupa a los componentes en colecciones.</p>
 <p>Lamp Illuminate to indicate status</p>	<p>El objeto lamp se utiliza para indicar estados en la aplicación de forma visual. Por lo general, al ejecutar una callback se produce un cambio de color que muestra una modificación en el estado.</p>

Realizado por: Casierra Alvaro, 2024.

Al hacer la integración de los diferentes componentes en función de la programación necesaria para la conexión, control y monitoreo de las señales EMG provenientes de los ocho electrodos del sensor usado, se obtiene como resultado una interfaz amigable e interactiva como se observa en la Ilustración 3-25 y 3-26.

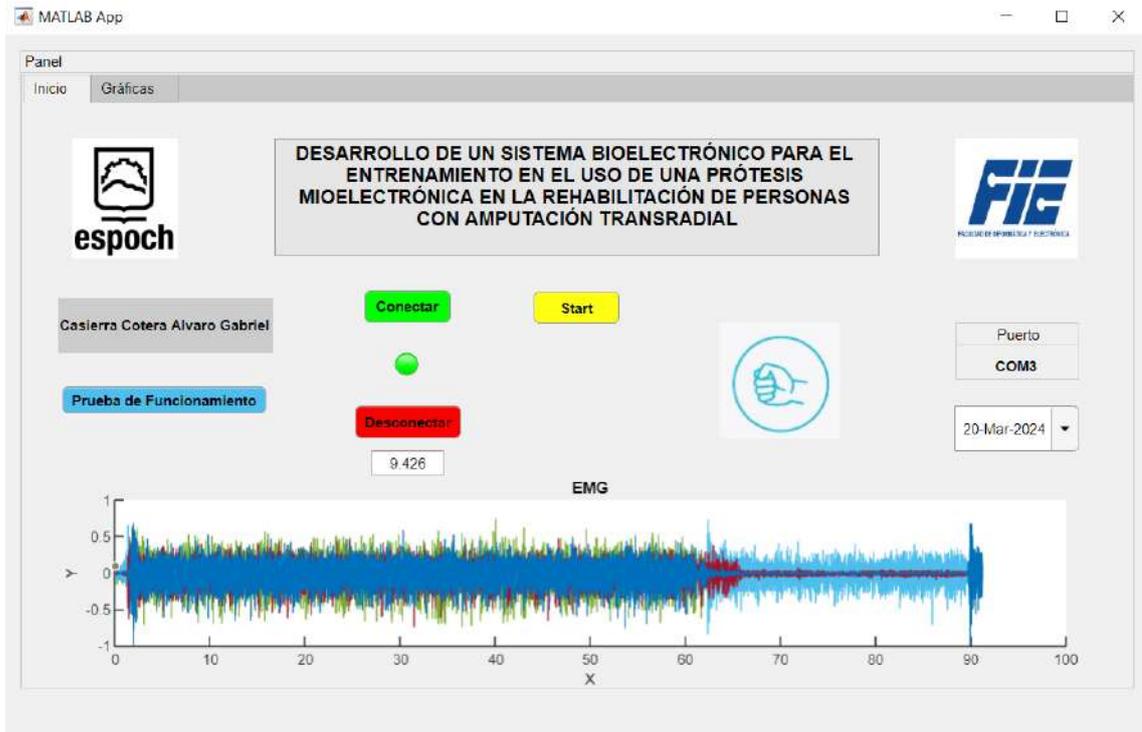


Ilustración 3-25: Panel de Inicio de la GUI

Realizado por: Casierra Alvaro, 2024.

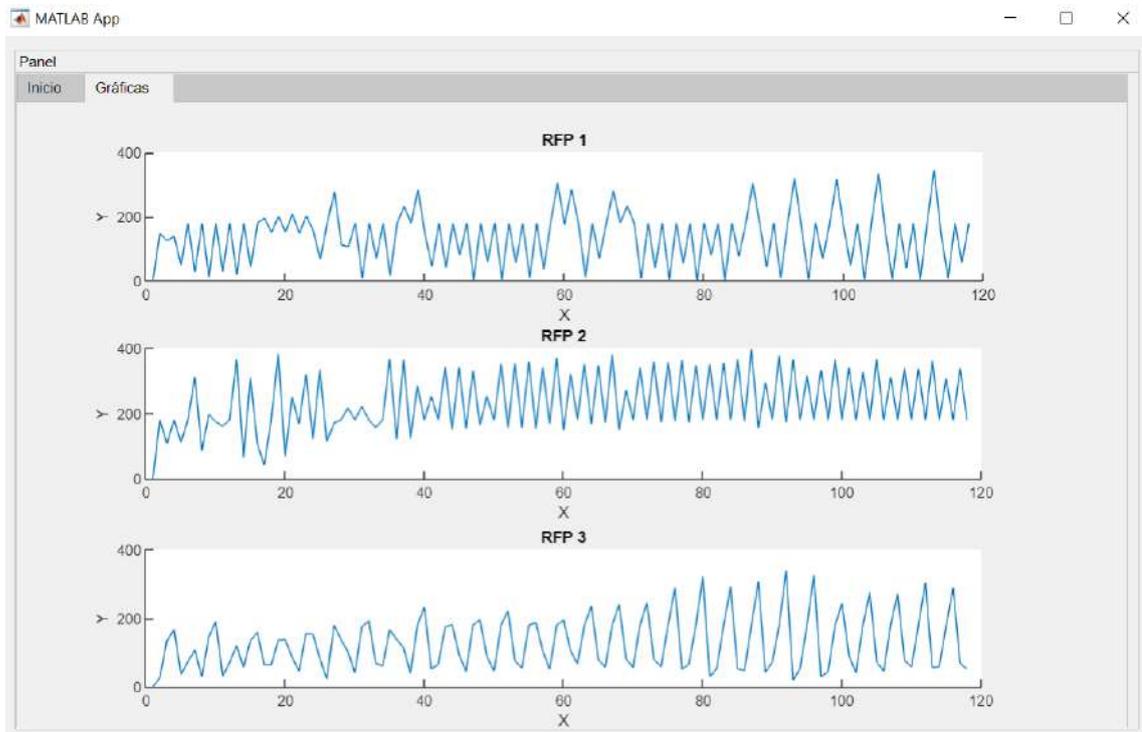


Ilustración 3-26: Panel de gráficas de la GUI

Realizado por: Casierra Alvaro, 2024.

CAPÍTULO IV

4. VALIDACIÓN DEL SISTEMA BIOELECTRÓNICO

En este capítulo se evalúa la funcionalidad del software y hardware del sistema bioelectrónico al realizar pruebas de validación con el sensor mioelectrónico Myo Armband, los sensores de presión, la comunicación entre los equipos y sensores, actuadores y sistemas de control, el tiempo de respuesta y el funcionamiento general. Adicionalmente se presenta un análisis de costos económicos del sistema.

4.1 Validación del tiempo de conexión del sistema.

El objetivo es verificar que el tiempo de conexión del sistema sea constante en todas las sesiones y no presente retardos en los tiempos de conexión del sistema. Se recolectaron 30 muestras para llevar a cabo la validación del tiempo de conexión del sistema.

Para recopilar los datos se utilizó la función TIC - TOC que proporciona Matlab a través de un indicador y un Label dentro de la GUI que muestra el tiempo transcurrido desde el momento de iniciar hasta que ya se ha establecido la comunicación. Para el procesamiento de estos se utiliza el software IBM SPSS Statistics 26. En la Ilustración 4-1 se muestra la toma de datos con la GUI.

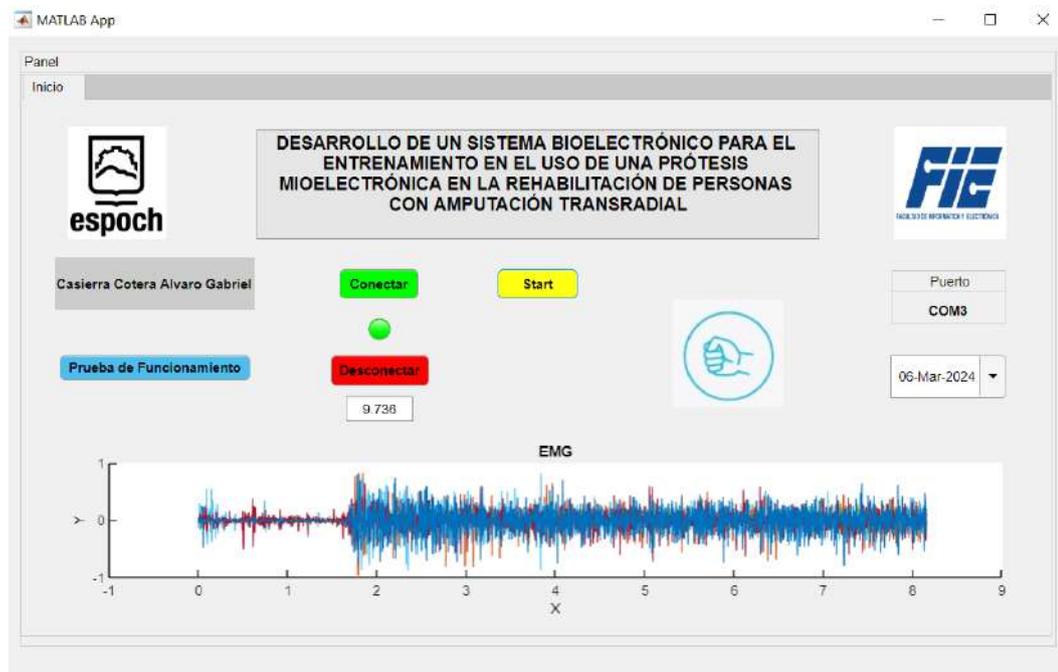


Ilustración 4-1: Toma de datos del tiempo de conexión del sistema

Realizado por: Casierra Alvaro, 2024.

En la Tabla 4-1 se muestran los datos obtenidos del tiempo de conexión del sistema bioelectrónico con la descripción de funcionamiento y verificación de conexión.

Tabla 4-1: Tiempos de conexión del sistema bioelectrónico.

N.º Prueba	Tiempo (s)	Conexión establecida
1	8.505	Si
2	9.15	Si
3	8.637	Si
4	8.824	Si
5	9.265	Si
6	8.709	Si
7	8.905	Si
8	8.737	Si
9	8.679	Si
10	8.876	Si
11	8.982	Si
12	9.459	Si
13	9.056	Si
14	8.939	Si
15	8.811	Si
16	8.644	Si
17	9.052	Si
18	8.903	Si
19	9.246	Si
20	8.873	Si
21	9.123	Si
22	9.036	Si
23	9.13	Si
24	9.186	Si
25	9.359	Si
26	8.973	Si
27	9.188	Si
28	9.026	Si
29	8.972	Si
30	8.687	Si

Realizado por: Casierra Alvaro, 2024.

Se calcula la media, mediana y moda de los tiempos de conexión del sistema bioelectrónico para determinar el valor en segundos utilizado para la conexión.

Tabla 4-2: Resultados Media, Mediana y Moda

de los tiempos de conexión.

Estadísticos		
Tiempo conexión		
N	Válido	30
	Perdidos	7
Media		8.96440
Mediana		8.97250
Moda		8.505 ^a

Realizado por: Casierra Alvaro, 2024.

Los resultados mostrados en la Tabla 4-2, indican que el valor de media es de 8.9644 seg, la mediana es de 8.9725 seg y el mínimo valor de moda es de 8.505 seg. Se plantea probar si el tiempo de conexión del sistema bioelectrónico es de 9 segundos en todas las sesiones de conexión con una significancia de 0.05. Para esto se plantean las siguientes hipótesis estadísticas.

1. Plantear la hipótesis

H0: El tiempo de conexión del sistema bioelectrónico es de 9 segundos.

H1: El tiempo de conexión del sistema bioelectrónico es diferente de 9 segundos.

2. Nivel de significancia: $\alpha= 0.05$.

3. Tipo de Prueba: t de Student

Supuestos de la prueba

Tamaño de muestra: igual a 30

Distribución de datos: Normal

Tabla 4-3: Resultados T Student para el tiempo de conexión del sistema bioelectrónico.

Prueba para una muestra						
Valor de prueba = 9						
	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
Datos	-0.845	29	0.405	-0.035600	-0.12179	0.05059

Realizado por: Casierra Alvaro, 2024.

Según los resultados de la prueba T-Student mostrados en la Tabla 4-3, el estadístico alcanza un valor de -0.845 y un p-value de $0.405 > 0.05$ por lo que se acepta la hipótesis nula con un nivel de significancia del 0.05 y se concluye que el tiempo promedio de conexión del sistema bioelectrónico es de 9 segundos.

En la Ilustración 4-2 se muestran los tiempos de conexión del sistema bioelectrónico en un diagrama de líneas.

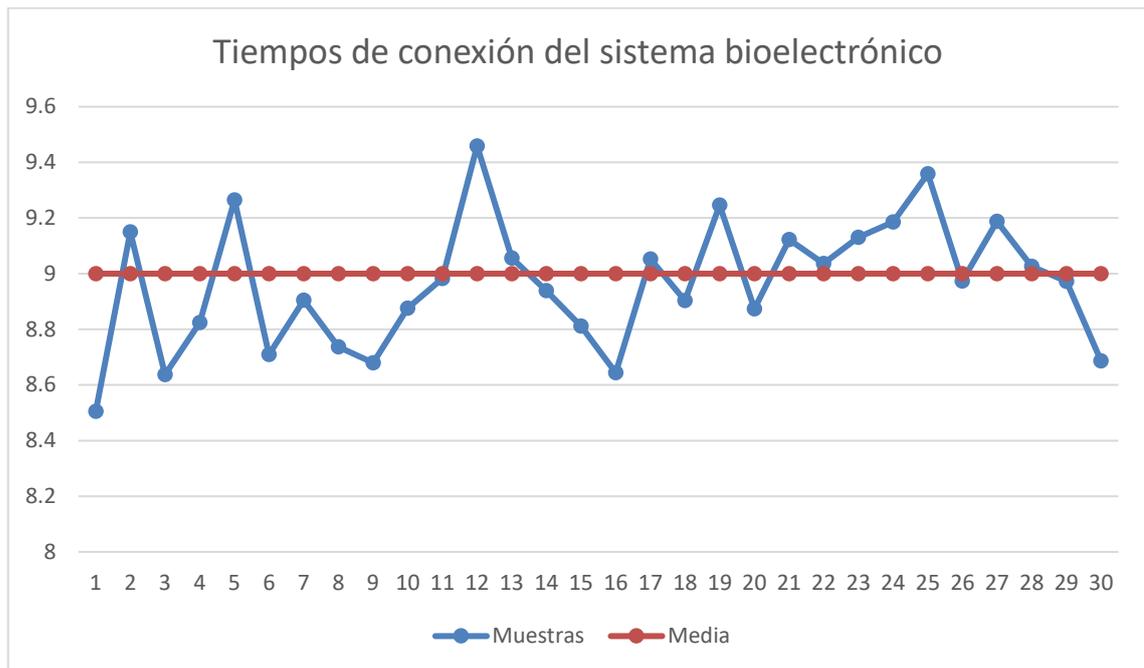


Ilustración 4-2: Tiempos de conexión del sistema bioelectrónico

Realizado por: Casierra Alvaro, 2024.

4.2 Validación del reconocimiento de gesto con el sensor Myo Armband

El objetivo es verificar la eficiencia del sensor Myo Armband al momento de detectar los principales gestos relacionados con el funcionamiento del sistema bioelectrónico como son el gesto de puño, palma adentro, palma afuera y dedos extendidos. Se recolectaron 30 muestras por cada gesto, y con dos personas diferentes, un hombre y una mujer para la validación del reconocimiento de los gestos.

Para obtener los datos del reconocimiento de los gestos se usa la GUI que presentará el gesto detectado y la aplicación Myo Connect que cuenta con un identificador de los gestos y los muestra en pantalla en tiempo real. En la Ilustración 4-3 se observa el proceso de recolección de datos.

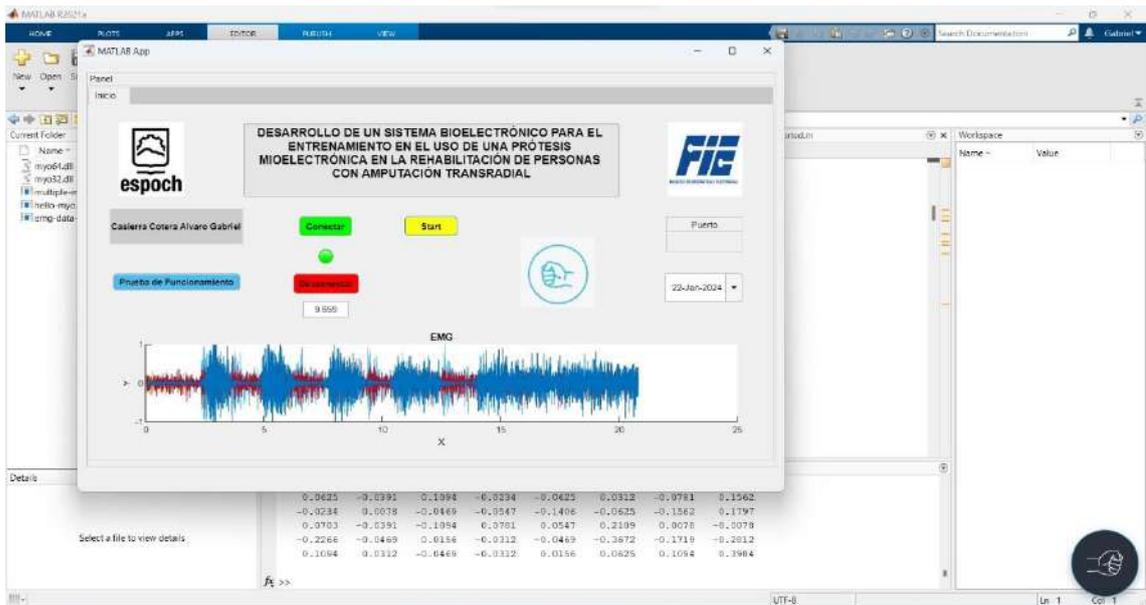


Ilustración 4-3: Toma de datos del reconocimiento de gestos

Realizado por: Casierra Alvaro, 2024.

En la Tabla 4-4 se muestran los datos dicotómicos obtenidos del reconocimiento de gestos por parte de la GUI y la aplicación Myo Connect para el hombre y en la Tabla 4-8 para la mujer.

Tabla 4-4: Datos obtenidos del hombre en el reconocimiento de gestos.

N.º Prueba	Puño	Dedos extendidos	Palma adentro	Palma afuera
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	0	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	0	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	1	1	1	1
19	1	1	1	1
20	1	0	1	1
21	0	1	1	1

22	1	1	1	1
23	1	1	1	1
24	1	0	1	1
25	1	1	1	1
26	1	1	1	1
27	1	1	1	1
28	1	1	1	1
29	1	1	1	1
30	1	1	1	1

Realizado por: Casierra Alvaro, 2024.

Se plantea probar si el reconocimiento tiene una efectividad igual con los cuatro gestos planteados por parte del sensor Myo Armband en el hombre con una significancia de 0.05. Las hipótesis estadísticas planteadas son:

1. Plantear la hipótesis

H0: No hay diferencia en el reconocimiento de los cuatro gestos por parte del sensor Myo Armband en el hombre.

H1: Hay diferencia en el reconocimiento de los cuatro gestos por parte del sensor Myo Armband en el hombre.

2. Nivel de significancia: $\alpha = 0.05$.

3. Tipo de Prueba: Cochran

Supuestos de la prueba

Variable nominal

Más de dos muestras relacionadas

Tabla 4-5: Resultados de la frecuencia de reconocimiento

de los gestos en el hombre.

Frecuencias		
	Valor	
	0	1
Puño	2	28
Dedos extendidos	3	27
Palma adentro	0	30
Palma afuera	0	30

Realizado por: Casierra Alvaro, 2024.

Tabla 4-6: Resultados Q de Cochran para el reconocimiento

de gestos en el hombre.

Estadísticos de prueba

N	30
Q de Cochran	3.667 ^a
gl	3
Sig. asintótica	0.300

Realizado por: Casierra Alvaro, 2024.

Tabla 4-7: Pruebas correctas e incorrectas de los gestos en el hombre.

Gesto	Puño	D. extendidos	P. adentro	P. afuera	Total
Gesto correcto	28	27	30	30	115
Gesto incorrecto	2	3	0	0	5
Total	30	30	30	30	120

Realizado por: Casierra Alvaro, 2024.

Según los resultados de la prueba Q de Cochran mostrados en la Tabla 4-6, el estadístico alcanzo un valor de 3.667 y un p-value de $0.300 > 0.05$ por lo tanto aceptamos la hipótesis nula H_0 , y se concluye que no hay diferencia en el reconocimiento de los cuatro gestos por parte del sensor Myo Armband en el hombre. Además, los resultados de la Tabla 4-7 determinan que aproximadamente un 95.83% de los reconocimientos con los diferentes gestos fueron correctos y un 4.17% fueron reconocimientos incorrectos.

En la Ilustración 4-4 se muestra la distribución de gestos reconocidos y no reconocidos en un diagrama de barras y en la Ilustración 4-5 se muestran gráficamente los resultados de los reconocimientos correctos e incorrectos.

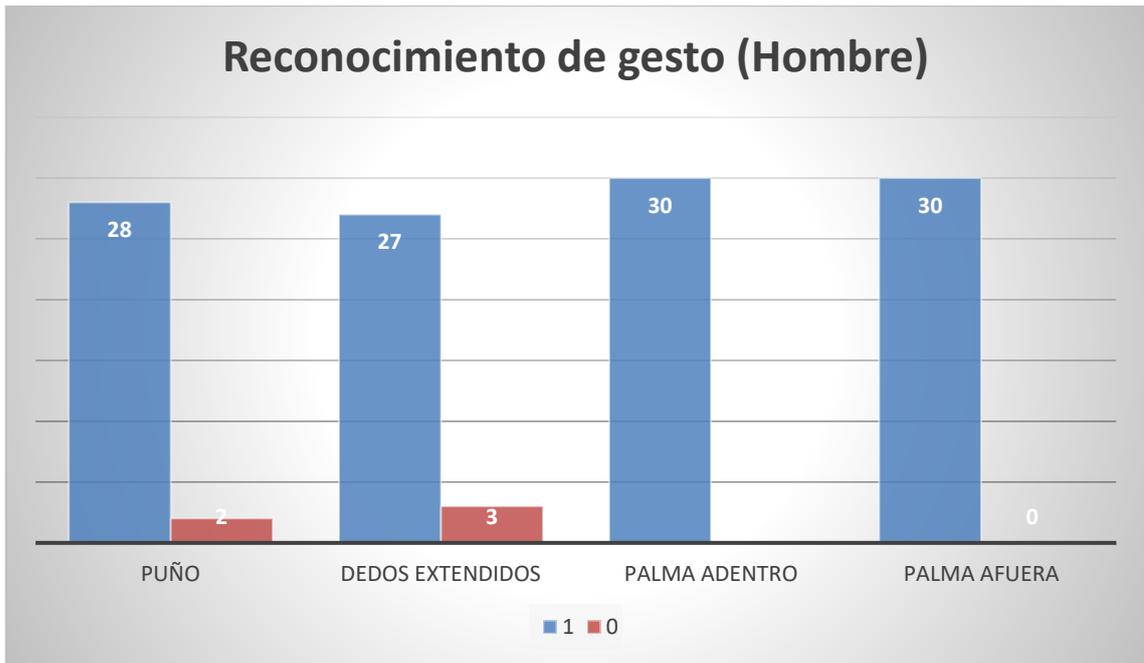


Ilustración 4-4: Diagrama de barras para el reconocimiento de gestos en el hombre

Realizado por: Casierra Alvaro, 2024.

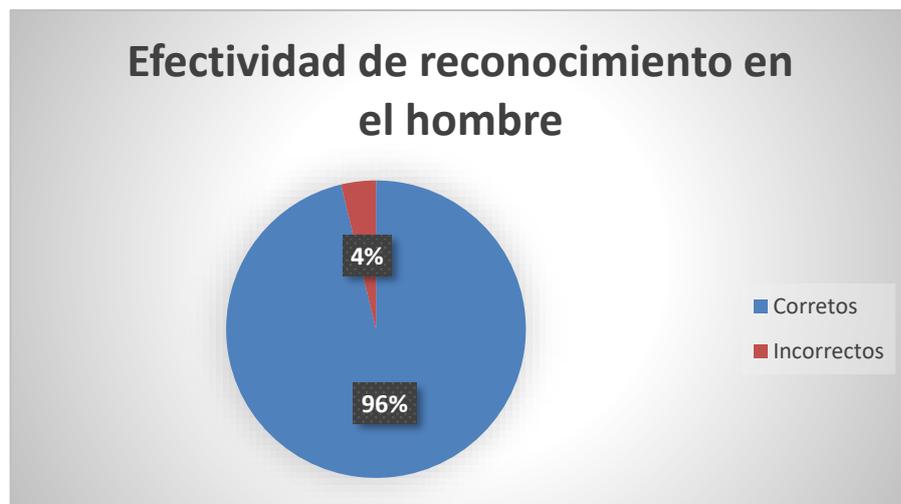


Ilustración 4-5: Efectividad de reconocimiento en el hombre

Realizado por: Casierra Alvaro, 2024.

Tabla 4-8: Datos obtenidos de la mujer en el reconocimiento de gestos.

N.º Prueba	Puño	Dedos extendidos	Palma adentro	Palma afuera
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1

5	0	1	0	1
6	1	1	1	1
7	1	1	1	1
8	1	0	1	1
9	1	1	1	1
10	1	1	0	1
11	1	1	1	1
12	0	1	1	0
13	1	1	1	1
14	1	0	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	0	1	1	0
19	1	1	1	1
20	1	0	1	1
21	1	1	1	1
22	1	1	0	0
23	0	1	1	1
24	1	1	1	1
25	1	0	1	1
26	1	1	1	0
27	1	1	0	1
28	1	1	1	1
29	1	0	1	1
30	1	1	1	1
	1	1	1	1

Realizado por: Casierra Alvaro, 2024.

Se plantea probar si el reconocimiento tiene una efectividad igual con los cuatro gestos planteados por parte del sensor Myo Armband en la mujer con una significancia de 0.05. Las hipótesis estadísticas planteadas son:

1. Plantear la hipótesis

H0: No hay diferencia en el reconocimiento de los cuatro gestos por parte del sensor Myo Armband en la mujer.

H1: Hay diferencia en el reconocimiento de los cuatro gestos por parte del sensor Myo Armband en la mujer.

2. Nivel de significancia: $\alpha = 0.05$.

3. Tipo de Prueba: Cochran

Supuestos de la prueba

Variable nominal

Más de dos muestras relacionadas

Tabla 4-9: Resultados de la frecuencia de reconocimiento de los gestos en la mujer.

Frecuencias		
	Valor	
	0	1
Puño	4	26
Dedos extendidos	5	25
Palma adentro	4	26
Palma afuera	4	26

Realizado por: Casierra Alvaro, 2024.

Tabla 4-10: Resultados Q de Cochran para el reconocimiento de gestos en la mujer.

Estadísticos de prueba	
N	30
Q de Cochran	.500 ^a
gl	3
Sig. asintótica	0.919

Realizado por: Casierra Alvaro, 2024.

Tabla 4-11: Pruebas correctas e incorrectas de los gestos en la mujer.

Gesto	Puño	D. extendidos	P. adentro	P. afuera	Total
Gesto correcto	26	26	25	26	103
Gesto incorrecto	4	4	5	4	17
Total	30	30	30	30	120

Realizado por: Casierra Alvaro, 2024.

Según los resultados de la prueba Q de Cochran mostrados en la Tabla 4-10, el estadístico alcanzo un valor de 0.500 y un p-value de $0.919 > 0.05$ por lo tanto aceptamos la hipótesis nula H_0 , y se concluye que no hay diferencia en el reconocimiento de los cuatro gestos por parte del sensor Myo Armband en la mujer. Además, los resultados de la Tabla 4-11 determinan que aproximadamente un 85.83% de los reconocimientos con los diferentes gestos fueron correctos y un 14.17% fueron reconocimientos incorrectos.

En la Ilustración 4-6 se muestra la distribución de gestos reconocidos y no reconocidos en un diagrama de barras y en la Ilustración 4-7 se muestran gráficamente los resultados de los reconocimientos correctos e incorrectos.

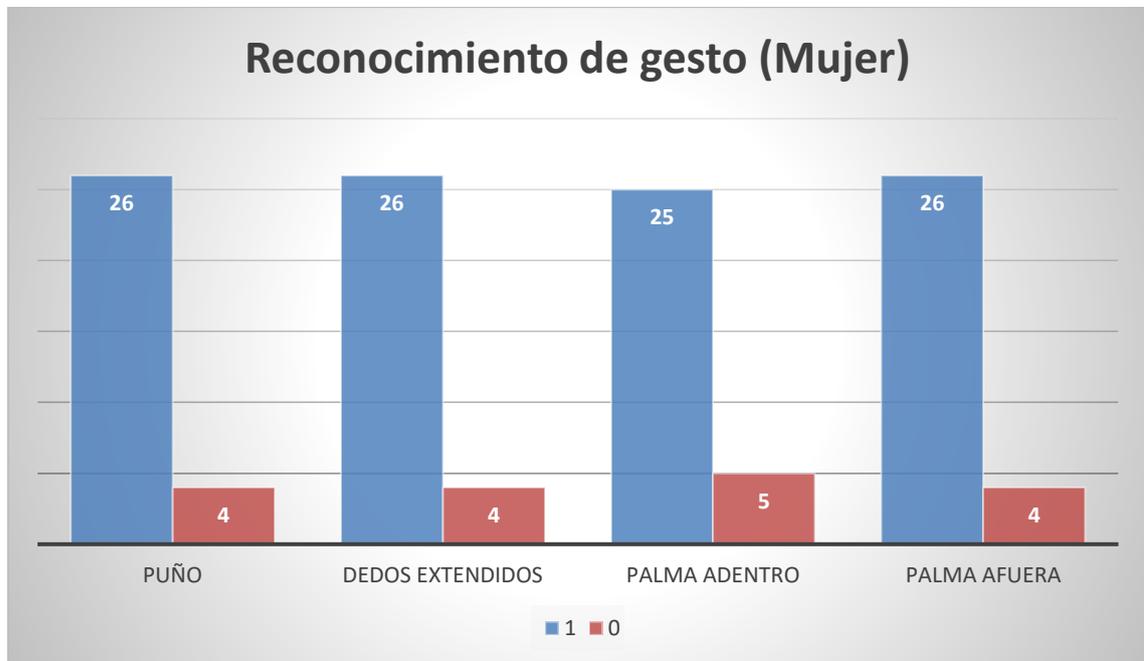


Ilustración 4-6: Diagrama de barras para el reconocimiento de gestos en la mujer

Realizado por: Casierra Alvaro, 2024.



Ilustración 4-7: Efectividad de reconocimiento en la mujer

Realizado por: Casierra Alvaro, 2024.

Al realizar las mediciones en dos usuarios diferentes se concluye que el sensor Myo Armband no genera errores significativos a la hora de la identificación del gesto, sin embargo, si existe una leve diferencia de 10% en la efectividad del reconocimiento entre el hombre y la mujer.

4.3 Validación de la efectividad de agarre de la prótesis

El objetivo de esta prueba es verificar que el sistema bioelectrónico, mediante una prótesis mioeléctrica acoplada con los sensores de presión puede efectuar el agarre de diferentes objetos, de los cuáles dos son esféricos de diámetros y colores diferentes, otro de forma prismática, otro de forma cilíndrica y los tres últimos tienen formas irregulares y amoldables.

Para lograr un agarre de forma correcta se deben tomar en consideración varios factores como lo son la fricción y el diámetro de los objetos. Se recolectaron diez muestras por cada objeto para la validación de la efectividad de agarre del sistema con la prótesis mioeléctrica. Para la verificación se hizo uso de un calibrador que permitió la medición de los objetos.

4.3.1 Características de los objetos seleccionados

Para realizar las pruebas se seleccionaron dos objetos esféricos de espuma de poliuretano, dos objetos de tipo recipiente de plástico, y los otros tres objetos son elementos comunes ubicados en bolsas delgadas de plástico, cuyas dimensiones de diámetro y ancho se muestran en la Tabla 4-12, para una mejor visualización cada objeto cuenta con colores diferentes.

Tabla 4-12: Dimensiones de los objetos.

N° Objeto	Objeto	Diámetro (cm)	Ancho (cm)
1	Botella 1	5.9	2.8
2	Azúcar	12.8	6
3	Fideo	8.3	6.1
4	Harina	8.3	6.3
5	Botella 2	2.6	-
6	Pelota 1	6	-
7	Pelota 2	7	-

Realizado por: Casierra Alvaro, 2024.

Las 30 muestras realizadas por cada objeto se las clasifica como un Agarre correcto o uno incorrecto. De esta manera se detallan un total de 210 muestras, las cuáles se muestran en la Tabla 4-13.

Tabla 4-13: Pruebas correctas e incorrectas de los objetos.

N° Objeto	1	2	3	4	5	6	7	Total
Agarre correcto	24	30	30	30	18	21	24	177
Agarre incorrecto	6	0	0	0	12	9	6	33
Total	30	30	30	30	30	30	30	210

Realizado por: Casierra Alvaro, 2024.

Según los resultados mostrados en la Tabla 4-13, se puede deducir que aproximadamente un 84.28% de los agarres con los diferentes objetos fueron correctos y un 15.72% fueron agarres incorrectos. En la Ilustración 4-8 se muestran gráficamente los resultados.

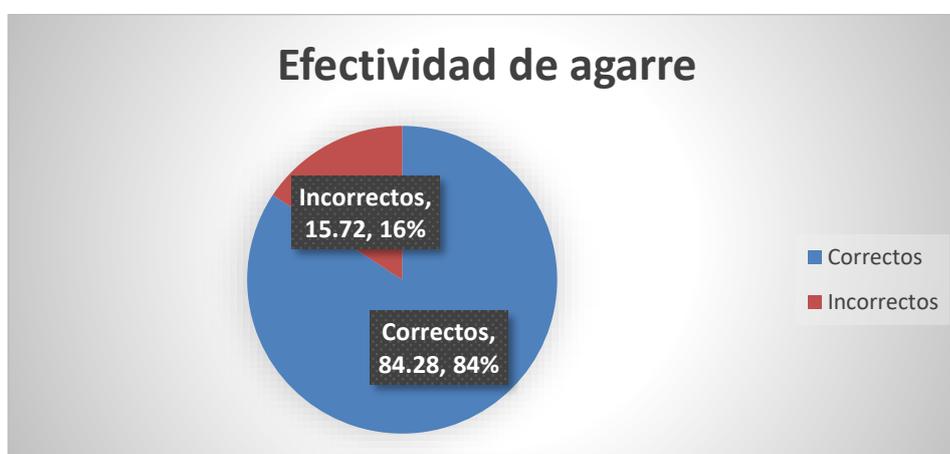


Ilustración 4-8: Efectividad de agarre

Realizado por: Casierra Alvaro, 2024.

4.4 Tiempo de respuesta

Para obtener un tiempo de respuesta promedio de los agarres correctos, se tomaron 30 muestras de los tiempos de respuesta, 6 por cada objeto como se muestra en la Tabla 4-14.

Tabla 4-14: Tiempos de respuesta de agarres correctos.

N° Muestra	Objeto	Tiempo (s)
1	Botella 1	2.96
2	Botella 1	2.83
3	Botella 1	3.05
4	Botella 1	3.07
5	Botella 1	2.90
6	Botella 1	2.93
7	Harina	3.79
8	Harina	3.60

9	Harina	3.71
10	Harina	3.68
11	Harina	3.64
12	Harina	3.78
13	Botella 2	2.81
14	Botella 2	2.88
15	Botella 2	2.85
16	Botella 2	2.77
17	Botella 2	2.74
18	Botella 2	2.75
19	Pelota 1	2.93
20	Pelota 1	3.00
21	Pelota 1	3.02
22	Pelota 1	2.94
23	Pelota 1	2.92
24	Pelota 1	2.93
25	Pelota 2	2.19
26	Pelota 2	2.31
27	Pelota 2	2.35
28	Pelota 2	2.22
29	Pelota 2	2.18
30	Pelota 2	2.34

Realizado por: Casierra Alvaro, 2024.

Se calcula la media, mediana y moda de los tiempos de respuesta de agarres efectivos del sistema bioelectrónico para determinar el valor en segundos. En la Ilustración 4-9 se muestran los datos de la Tabla 4-14.

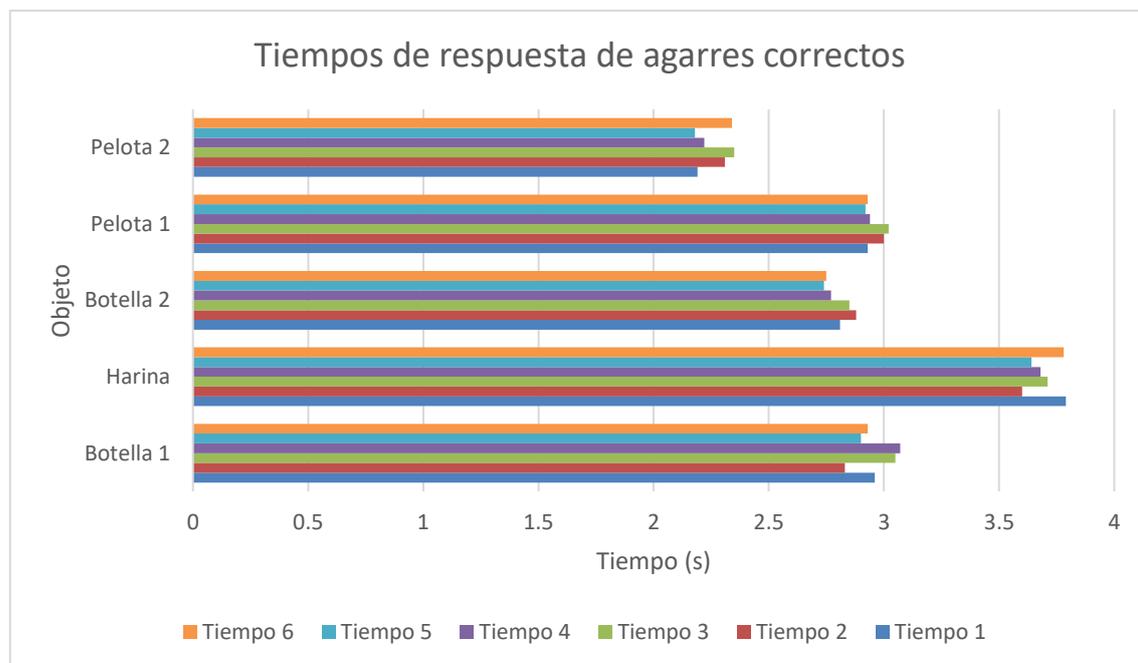


Ilustración 4-9: Tiempos de respuesta de agarres correctos

Realizado por: Casierra Alvaro, 2024.

Tabla 4-15: Resultados media, mediana y moda de tiempos de agarre.

Estadísticos		
Tiempo (s)		
N	Válido	30
	Perdidos	0
Media		2.9357
Mediana		2.9250
Moda		2.93

Realizado por: Casierra Alvaro, 2024.

Los resultados mostrados en la Tabla 4-15, indica que el valor de media es de 2.9357 seg, la mediana es de 2.925 seg y el mínimo valor de moda es de 2.93 seg.

4.5 Identificación de objeto por presión

Debido a que cada objeto tiene una forma diferente, no siempre será posible que los tres sensores detecten el objeto, sin embargo, no se generan problemas al momento de realizar un agarre efectivo. Para la realización de esta prueba se hizo uso de tres objetos, los cuáles son la harina, la pelota 2 y la botella 2. Recolectando 10 muestras por cada sensor y objeto para la validación de la identificación de objeto por presión con la prótesis mioeléctrica. La toma de muestras se la realizó a través de gráficas generadas en tiempo. En la Tabla 4-16 se muestran los datos obtenidos para la harina, en la Tabla 4-17 para la pelota dos y en la Tabla 4-18 para la botella 2.

Tabla 4-16: Datos de los sensores de presión RFP para la harina.

Nº Muestra	RFP 1	RFP 2	RFP 3
1	171	171	305
2	176	208	414
3	200	221	291
4	168	161	373
5	109	222	358
6	161	226	323
7	174	217	314
8	108	169	295
9	136	196	336
10	182	205	371
Promedio	158.5	199.6	338

Realizado por: Casierra Alvaro, 2024.

Se calcula la media de los tres sensores de presión del sistema bioelectrónico con las diez muestras tomadas, obteniendo así un valor de media de 158.5, 199.6 y 338 respectivamente para cada

sensor, datos con lo cuáles en conjunto, se podrá deducir que el objeto que está siendo manipulado por la prótesis es harina. En la Ilustración 4-10 se muestran los datos de los sensores en el reconocimiento de la harina en un diagrama de líneas.

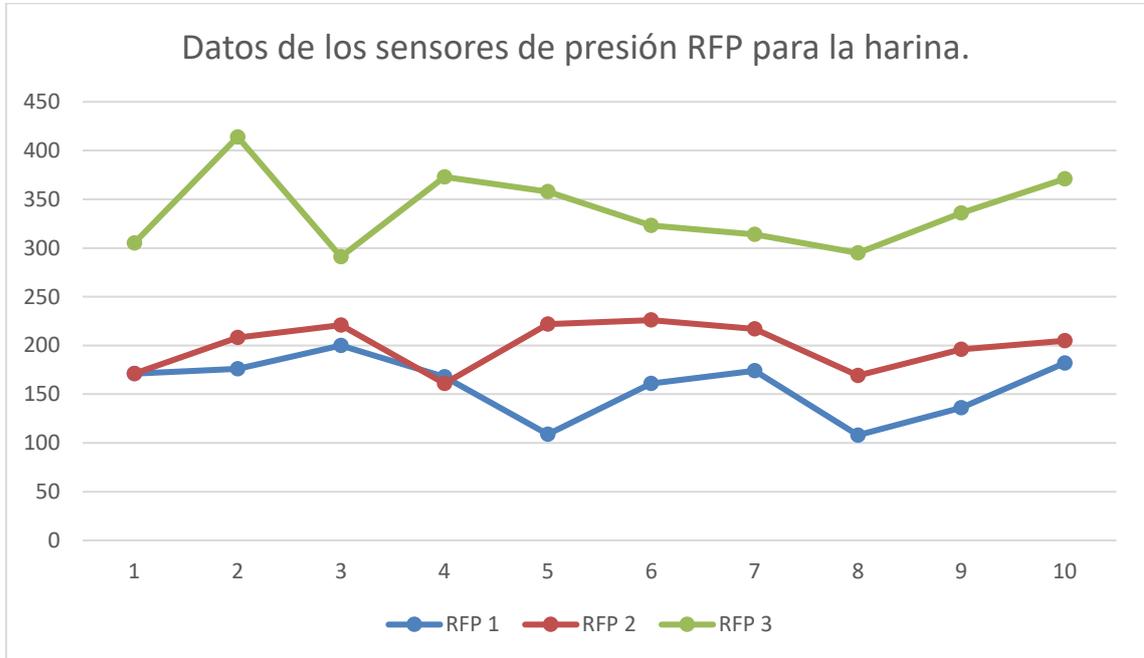


Ilustración 4-10: Datos de los sensores de presión RFP para la harina

Realizado por: Casierra Alvaro, 2024.

Tabla 4-17: Datos de los sensores de presión RFP para la pelota 2.

N° Muestra	RFP 1	RFP 2	RFP 3
1	41	112	252
2	39	114	246
3	46	175	135
4	23	191	123
5	30	215	98
6	31	208	58
7	27	263	104
8	42	184	216
9	41	172	228
10	38	202	193
Promedio	35.8	183.6	165.3

Realizado por: Casierra Alvaro, 2024.

Se calcula la media de los tres sensores de presión del sistema bioelectrónico con las diez muestras tomadas, obteniendo así un valor de media de 35.8, 183.6 y 165.3 respectivamente para cada sensor, datos con lo cuáles en conjunto, se podrá deducir que el objeto que está siendo manipulado

por la prótesis es la pelota dos. En la Ilustración 4-11 se muestran los datos de los sensores en el reconocimiento de la pelota 2 en un diagrama de líneas.

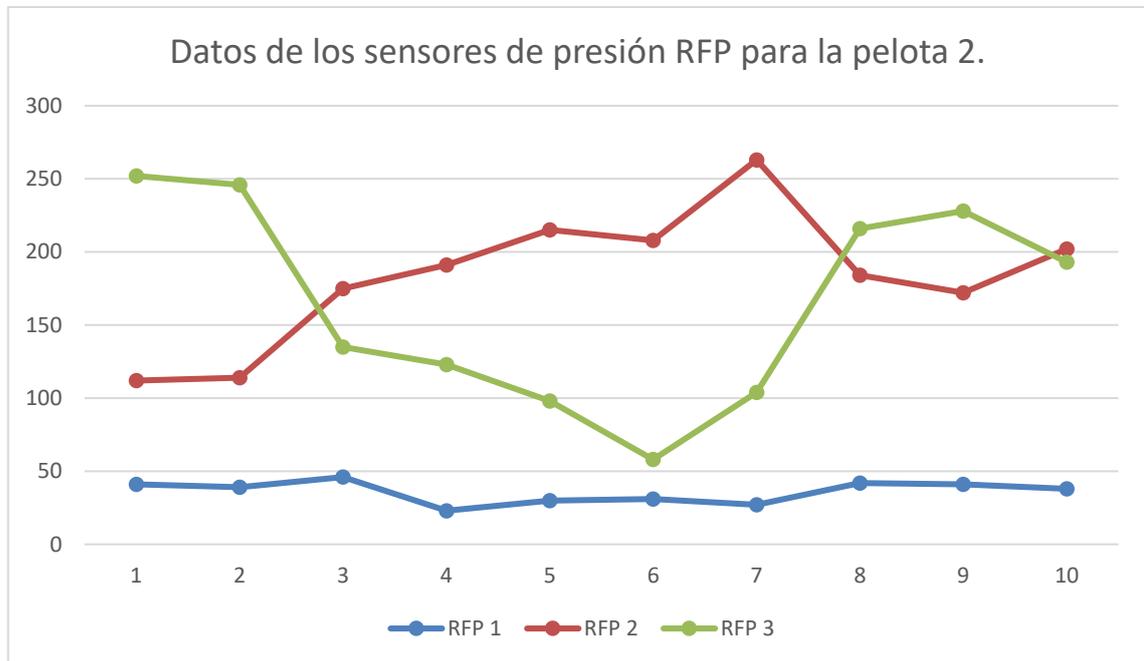


Ilustración 4-11: Datos de los sensores de presión RFP para la pelota 2

Realizado por: Casierra Alvaro, 2024.

Tabla 4-18: Datos de los sensores de presión RFP para la botella 2.

Nº Muestra	RFP 1	RFP 2	RFP 3
1	40	144	146
2	38	102	154
3	25	156	128
4	41	99	146
5	35	138	124
6	39	161	129
7	29	147	151
8	34	136	103
9	40	109	115
10	48	118	136
Promedio	36.9	131	133.2

Realizado por: Casierra Alvaro, 2024.

Se calcula la media de los tres sensores de presión del sistema bioelectrónico con las diez muestras tomadas, obteniendo así un valor de media de 36.9, 131 y 133.2 respectivamente para cada sensor, datos con los cuales en conjunto, se podrá deducir que el objeto que está siendo manipulado por la prótesis es la botella 2. En la Ilustración 4-12 se muestran los datos de los sensores en el reconocimiento de la botella 2 en un diagrama de líneas.

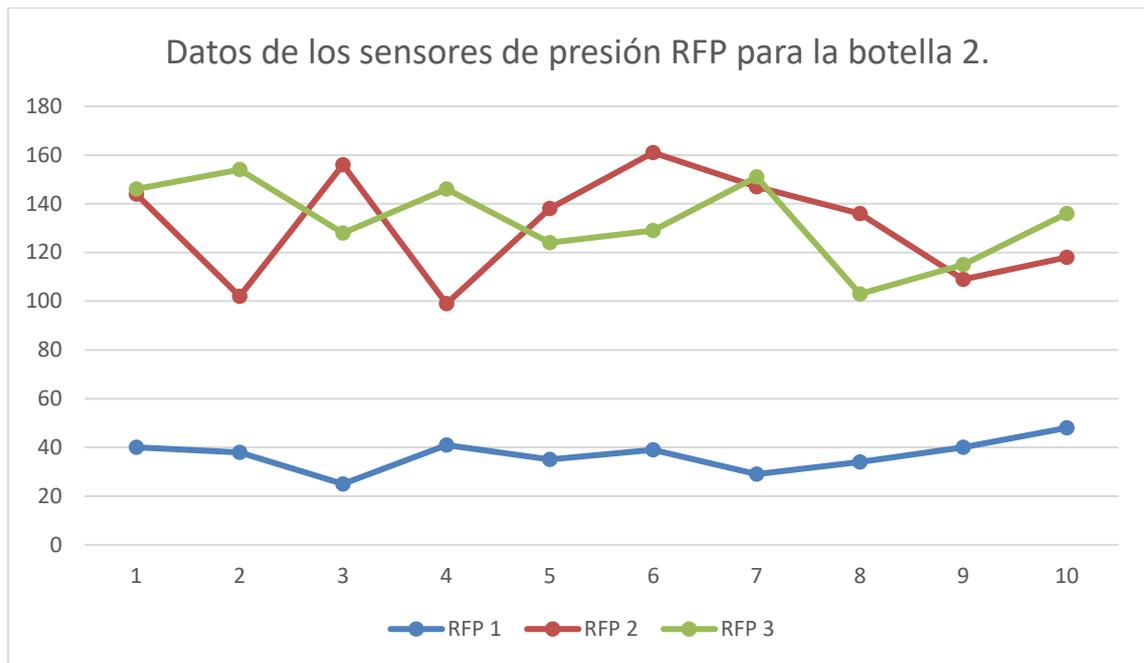


Ilustración 4-12: Datos de los sensores de presión RFP para la botella 2

Realizado por: Casierra Alvaro, 2024.

4.6 Consumo de energía del sistema bioelectrónico

El tiempo promedio que toma el sistema mediante la prótesis en lograr un agarre correcto de los objetos es de 2.9357 segundos, entonces para poder determinar la corriente consumida por el sistema se evalúa la corriente que consume cada elemento de manera individual, como se detalla en la Tabla 4-19.

Tabla 4-19: Consumo de corriente del sistema.

Componente	Corriente
Servomotor (x5)	0.5A
Myo Armband	100mA
RFP 602 (x3)	45.45uA
Arduino UNO	45mA
Computador	2.37A
Total	3.015A

Realizado por: Casierra Alvaro, 2024.

Obtenido el valor promedio o media del tiempo de agarre efectivo se continuará con el análisis del consumo del sistema en funcionamiento como se muestra en la Ecuación 4.

$$\begin{aligned}
 \text{Capacidad} &= \text{Corriente} \times \text{tiempo} & (4) \\
 \text{Capacidad} &= 3.015A \times 815.472e^{-6} \\
 \text{Capacidad} &= 2.4586mAh
 \end{aligned}$$

La energía total aproximada consumida por el dispositivo se calcula como se muestra en la Ecuación 5:

$$\begin{aligned}
 \text{Energía} &= \text{Voltaje} \times \text{Capacidad} & (5) \\
 \text{Energía} &= 5V \times 2.4586mAh \\
 \text{Energía} &= 12.293mWh
 \end{aligned}$$

4.7 Evaluación económica del prototipo

4.7.1 Costo de materiales y herramientas

Las herramientas y materiales utilizados en la elaboración del sistema bioelectrónico se presentan en la Tabla 4-20.

Tabla 4-20: Costo de materiales y herramientas.

Cantidad	Descripción	Valor Unitario	Total
1	Myo Armband	\$200	\$200
5	Servomotor	\$5	\$25
3	RFP 602	\$15	\$45
1	Arduino UNO	\$50	\$50
1	Fuente DC	\$5	\$5
1	Placa	\$5	\$5
1	Cables	\$5	\$5
1	Prótesis	\$100	\$100
TOTAL			\$435

Realizado por: Casierra Alvaro, 2024.

El sistema tiene un costo accesible para su implementación, ya que, en el mercado no existen sistemas que permitan realizar un entrenamiento previo a la obtención de sus prótesis mioeléctricas. Además, el sistema brinda grandes aportes a los usuarios que harán uso de este, desde los técnicos que podrán seguir realizando pruebas, hasta los pacientes en sus periodos de rehabilitación de manera asistida y monitoreada.

CONCLUSIONES

- Para el desarrollo del sistema bioelectrónico es necesario un sensor muscular de alta calidad y precisión que permita obtener las señales mioeléctricas correctas. Se seleccionó el sensor Myo Armband debido a que presenta un diseño ergonómico y con comunicación inalámbrica eliminando así el uso de cables.
- La posición de los sensores de presión es de gran importancia ya que estos son quienes aportarán la retroalimentación necesaria para poder realizar el agarre de los objetos, por ello es necesario realizar un ensayo después de su instalación que permita determinar cuál será el nivel de presión adecuado para agarrar los diferentes objetos.
- Los parámetros del PID, k_p , k_i y k_d , son quienes determinan la fuerza, precisión y la velocidad con la que se realiza el agarre de los objetos, por ello mediante las pruebas de efectividad de agarre realizadas se determinó que un sistema PI es el más adecuado para lograr un agarre efectivo y controlado.
- La calibración del sensor se debe realizar mediante la aplicación Myo Connect para cada usuario, ya que esto asegurará mejores resultados en función de las condiciones físicas de cada usuario. Cuando el sensor está correctamente calibrado, el reconocimiento de los gestos es superior al 86% tanto en hombres como en mujeres.
- Al llevar a cabo las pruebas de latencia del sistema, se evidencia que el sistema logra una conexión exitosa en un tiempo promedio de 9 segundos permitiendo la interacción entre el paciente, el sensor y los actuadores.
- Mediante el análisis estadístico se determinó que el rango de diámetros de objetos con diferentes formas, que el sistema puede agarrar de manera correcta es de 4 a 13 cm con un tiempo promedio de agarre de 2.93 segundos. Además, se obtuvo un 84.28% en la efectividad de agarre, lo que indica que el sistema cumple con su función.
- Se realizaron los cálculos de costos del sistema bioelectrónico y se determinó que su precio final no supera los \$500, por lo que es un sistema de fácil implementación, lo cual permite su aplicabilidad en una variedad de locaciones en las cuáles se necesite un sistema de entrenamiento.

RECOMENDACIONES

- Investigar las diferentes técnicas de control inteligente y realizar una comparativa con los sistemas de control tradicionales.
- Estudiar los archivos MEX y su compatibilidad con los diferentes lenguajes de programación, para el desarrollo de interfaces gráficas.
- Desarrollar interfaces gráficas en diferentes lenguajes y realizar comparativas de efectividad en el tiempo de conexión.
- Configurar un algoritmo que permita incluir las librerías y los archivos MEX para la interfaz gráfica de manera automática.
- Implementar sistemas de aislamiento del ruido estático y eléctrico para mejorar la adquisición de datos.
- Implementar tecnología inmersiva para desarrollar un sistema de entrenamiento físico y virtual.
- Agregar movimientos independientes que permitan una mayor funcionalidad en la prótesis.

BIBLIOGRAFÍA

1. **ÁLAMO, Teodoro.** *Diseño del Controlador PID*. Sevilla-España: Departamento de Ingeniería de Sistemas y Automática, 2015, págs. 3-5.
2. **ALONSO, A; et al.** “Entrenador mioeléctrico de prótesis para amputados de brazo y mano”. *Mapfre Medicina*, vol. 13, n° 1, (2002), (España). págs. 11-19.
3. **ALVA, Carlos; et al.** “Procesamiento de señales mioeléctricas aplicado a un robot de cinco grados de libertad”. *Intercon* [en línea], 2011, (Perú), págs. 1-6. [Consulta: 22 noviembre 2023]. Disponible en: <https://www.urp.edu.pe/pdf/id/2836/n/processing-of->
4. **ANAYA, María; et al.** “Diseño de controladores P, PI y PID para el comportamiento dinámico de un servo-sistema hidráulico, basado en desarrollo experimental”. *Epistemus* [en línea], 2014, (México), vol. 16 (8), págs. 13-21. [Consulta: 12 noviembre 2023]. ISSN 2007-4530. Disponible en: www.epistemus.uson.mx
5. **ANDAGAMA, Miguel.** Implementación de un algoritmo para el reconocimiento de gestos del brazo humano en tiempo real utilizando la imu del dispositivo myo armband. (Trabajo de titulación). Escuela Politécnica Nacional. Quito-Ecuador. 2022. págs. 16-20.
6. **ARDUINO.** *¿Qué es Arduino?*. [blog]. [Consulta: 4 diciembre 2023]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>
7. **ARTEAGA, Oscar; et al.** “Diseño y construcción de una prótesis biónica de mano de 7 grados de libertad utilizando materiales inteligentes y control mioeléctrico adaptada para varios patrones de sujeción”. [en línea], 2015, (Ecuador), págs. 1-10. [Consulta: 9 noviembre 2023]. Disponible en: <http://repositorio.espe.edu.ec/xmlui/handle/21000/10191>
8. **ARTEAGA, Sandra.** *Myo, un brazalete para controlar los dispositivos con gestos*. [blog]. [Consulta: 17 noviembre 2023]. Disponible en: <https://computerhoy.com/noticias/hardware/myo-brazalete-controlar-dispositivos-gestos-35559>
9. **BEJARANO, Eliana.** Diseño de un objeto lúdico para la rehabilitación de pacientes con amputación transradial. (Trabajo de titulación). Universidad del Valle. Santiago de Cali-Colombia. 2019. págs. 19-45.
10. **BENITO, Iván.** *Myo Armband – El futuro ya está entre nosotros, pero todavía le falta algo*. [blog]. [Consulta: 17 noviembre 2023]. Disponible en: <https://gouforit.com/myo-aramband-el-futuro-ya-esta-entre-nosotros-pero-todavia-le-falta-algo-opinion/>
11. **BRAVO, Luis.** Brazo artificial adaptable como prótesis mioeléctrica. (Trabajo de titulación). Instituto Politécnico Nacional. México D.F.-México. 2003. págs. 30-35.

12. **BURBANO, Jaime; et al.** Entrenador virtual y real de prótesis mioeléctrica de mano derecha. (Trabajo de titulación). Universidad del Azuay. Cuenca-Ecuador. 2016. págs. 2-30.
13. **CAMPOS, Nancy; et al.** “Diseño de sistema mecatrónico para terapia de extremidad superior de niños”. *Xikua* [en línea], 2021, (México), vol. 9 (18), págs. 6-13. [Consulta: 20 noviembre 2023]. ISSN 2007-4948. Disponible en: <https://repository.uaeh.edu.mx/revistas/index.php/xikua/issue/archive>
14. **CONADIS.** *Personas con discapacidad en el Ecuador*. [blog]. [Consulta: 5 septiembre 2023]. Disponible en: <https://obest.uta.edu.ec/wp-content/uploads/2022/04/Discapacidades-en-el-Ecuador-2.pdf>
15. **CONADIS.** *Estadísticas de Discapacidad*. [blog]. [Consulta: 5 septiembre 2023]. Disponible en: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>
16. **CRUZ, Andy.** Desarrollo y construcción de prótesis robótica para personas con amputación transradial. [En línea]. (Trabajo de titulación). Universidad autónoma del estado de México, Estado de México-México. 2018. págs. 18-24. [Consulta: 2023-11-11]. Disponible en: <http://ri.uaemex.mx/bitstream/handle/20.500.11799/99086/CRUZ%20CALDER%c3%93N%20ADNY%20SAYTO.pdf?sequence=1&isAllowed=y>
17. **CRUZ, Diana.** *Entrenador virtual para adaptarse a las prótesis*. [blog]. [Consulta: 5 septiembre 2023]. Disponible en: <https://periodico.unal.edu.co/articulos/entrenador-virtual-para-adaptarse-a-las-protesis/>
18. **CUESTA, Juan & TAMAY, Manuel.** Diseño y construcción de una prótesis para amputación transradial de miembro superior derecho utilizando control mioeléctrico. [En línea]. (Trabajo de titulación). Universidad Politécnica Salesiana sede Cuenca, Cuenca-Ecuador. 2022. págs. 58-63 [Consulta: 2023-12-2]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/23040/1/UPS-CT010009.pdf>
19. **ESP32.** *The internet of things with esp32*. [blog]. [Consulta: 18 noviembre 2023]. Disponible en: <http://esp32.net/>
20. **ESPINOZA, María & GARCÍA, Daniela.** “Niveles de amputación en extremidades inferiores: Repercusión en el futuro del paciente”. *Las Condes*, vol. 25, n° 2, (2014), (Argentina). págs. 276-280.
21. **GIL, Juan & RUBIO, Ángel.** *Fundamentos de control automático de sistemas continuos y muestreados* [en línea]. San Sebastián-España: Unicopia, C.B., 2010. [Consulta: 4 noviembre 2023]. ISBN 978-84-613-4618-9. Disponible en: <https://dadun.unav.edu/bitstream/10171/7096/4/Gil-Control.pdf>

22. **GISI.** *Tarjetas de desarrollo.* [blog]. [Consulta: 15 septiembre 2023]. Disponible en: <https://virtual.cuautitlan.unam.mx/intar/sistdig/tarjetas-de-desarrollo/>
23. **GUEVARA, Brenda.** Desarrollo de un prototipo de prótesis mioeléctrica de mano de bajo costo empleando actuadores lineales y sensores musculares. (Trabajo de titulación). Escuela Superior Politécnica de Chimborazo. Riobamba-Ecuador. 2020. págs. 62-79.
24. **HETPRO.** *Sensor de Fuerza o Presión MF01.* [blog]. [Consulta: 5 diciembre 2023]. Disponible en: <https://hetpro-store.com/TUTORIALES/sensor-de-fuerza-o-presion-mf01/>
25. **KUONGSHUN.** *Sensor de presión de película delgada RFP-602.* [blog]. [Consulta: 5 diciembre 2023]. Disponible en: <https://www.kuongshun-ks.com/uno/uno-accessories/thin-film-pressure-sensor-rfp-602.html>
26. **LUNA, Joshua; et al.** *Control PID de un Sistema Balancín.* [blog]. [Consulta: 06 noviembre 2023]. Disponible en: <https://pdfcoffee.com/mecanismo-balancin-pdf-free.html>
27. **MARTÍNEZ, Gerardo; et al.** “Controlador multieje de posicionamiento de servomotores bldc implementado en fpga”. *Celaya Pistas Educativas*, vol. 39, n° 125, (2017), (México). págs. 340-361.
28. **MATA, Gloria; et al.** *Análisis de sistemas y señales con cómputo avanzado* [en línea]. 2017. México D.F.-México: Córima Books, 2017. [Consulta: 13 noviembre 2023]. Disponible en: <http://www.librosoa.unam.mx/bitstream/handle/123456789/293/Libro%20Coord%20G-LORIA%20MATA-SyS.pdf?sequence=3&isAllowed=y>
29. **MÁXIMO, Miriam.** Entorno virtual para tareas de manipulación y agarre con un brazo robótico asistencial. (Trabajo de titulación) (Maestría). Universidad de Alicante. Alicante-España. 2022. págs. 54-70.
30. **MORA, Anthony.** Integración de señales electromiográficas a sistemas de control para manipulación de una prótesis robótica de mano. (Trabajo de titulación). Escuela Superior Politécnica de Chimborazo. Riobamba-Ecuador. 2020. págs. 40-54.
31. **OBSERVATORI DE LA DISCAPACITAT FÍSICA -ODF.** *La discapacidad física: ¿qué es y qué tipos hay?* [blog]. [Consulta: 6 septiembre 2023]. Disponible en: <https://www.observatoridiscapitat.org/es/la-discapacidad-fisica-que-es-y-que-tipos-hay>
32. **ORGANIZACIÓN PANAMERICANA DE LA SALUD.** *Discapacidad.* [blog]. [Consulta: 5 septiembre 2023]. Disponible en: <https://www.paho.org/es/temas/discapacidad>

33. **ORTEGA, Carlos.** Sistema electrónico de rehabilitación para personas que padecen la enfermedad neuropática de parálisis facial periférica. [En línea]. (Trabajo de titulación). Universidad Técnica de Ambato, Ambato-Ecuador. 2021. págs. 40-42. [Consulta: 2023-11-17]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/33853/1/t1898ec.pdf>
34. **PEREZ, Mario; et al.** *Introducción a los sistemas de control y modelo matemático para sistemas lineales invariantes en el tiempo.* [en línea]. San Juan-Argentina: 2007. [Consulta: 24 septiembre 2023]. Disponible en: <http://dea.unsj.edu.ar/control1/apuntes/unidad1y2.pdf>
35. **POSADA, Johnny.** “Modulación por ancho de pulso (PWM) y modulación vectorial (SVM). Una introducción a las técnicas de modulación”. *Redalyc* [en línea], 2005, (Colombia), vol. 1 (25), págs. 70-83. [Consulta: 30 noviembre 2023]. ISSN 0121-0777. Disponible en: <http://www.redalyc.org/articulo.oa?id=47802507>
36. **PUJOL, José.** Control de articulación mediante PWM y sistemas embebidos [En línea]. (Trabajo de titulación). Universidad de Sevilla, Sevilla-España. 2005. págs. 7-19. [Consulta: 2023-11-14]. Disponible en: https://biblus.us.es/bibing/proyectos/abreproy/50024/fichero/Memoria_JPujol.pdf
37. **PYTHON SOFTWARE FOUNDATION.** *El tutorial de Python.* [blog]. [Consulta: 18 noviembre 2023]. Disponible en: <https://docs.python.org/es/3/tutorial/>
38. **RASPBERRY PI.** *¿Qué es Raspberry Pi?* [blog]. [Consulta: 18 noviembre 2023]. Disponible en: <https://raspberrypi.cl/que-es-raspberry/>
39. **RODRIGUEZ, Luis; et al.** “Evaluación de un Sistema Sensorial para el Control del Agarre en una Prótesis Mioeléctrica de Mano”. *Iberdiscap* [en línea], 2017, (Colombia), vol. 1, págs. 217-222. [Consulta: 22 diciembre 2023]. Disponible en: <https://www.researchgate.net/publication/321479888>
40. **SALAS, Carlos.** Diseño de una Prótesis Mioeléctrica para desarticulación de muñeca. (Trabajo de titulación). Pontificia Universidad Católica del Perú. Lima-Perú. 2014. págs. 21-30.
41. **STERN, Becky.** *Inside Myo.* [blog]. [Consulta: 26 febrero 2024]. Disponible en: <https://learn.adafruit.com/myo-armband-teardown/inside-myo>
42. **SUPRAPTO, Sena; et al.** “Gym training muscle fatigue monitoring using EMG myoware and arduino with envelope and sliding window methods”. *International Journal of Reconfigurable and Embedded Systems* [en línea], 2023, (Indonesia) vol. 12 (3), págs. 345-350. [Consulta: 22 diciembre 2023]. ISSN 27222608. DOI 10.11591/ijres.v12.i3pp345-350. Disponible en: https://www.researchgate.net/publication/373292132_Gym_training_muscle_fatigue_m

onitoring_using_EMG_myoware_and_arduino_with_envelope_and_sliding_window_methods

43. **THE MATHWORKS, Inc.** *¿Qué es MATLAB?* [blog]. [Consulta: 24 septiembre 2023]. Disponible en: <https://la.mathworks.com/products/matlab.html>
44. **TOWER PRO.** *Servo digital MG90S.* [blog]. [Consulta: 4 diciembre 2023]. Disponible en: <https://www.towerpro.com.tw/product/mg90s-3/>
45. **VARGAS, David; et al.** “Entrenador virtual para el aprendizaje de rutinas motoras, en personas con ausencia de un miembro superior”. *UNAD* [en línea], 2011, (Colombia), vol. 10 (2), págs. 167-182. [Consulta: 8 diciembre 2023]. ISSN 0124 793X. Disponible en: <https://hemeroteca.unad.edu.co/index.php/revista-de-investigaciones-unad/article/view/762/1425>
46. **VÁZQUEZ, Vicario; et al.** “Sistema de reconocimiento inteligente de señales mioeléctricas del movimiento de mano humana”. *Redalyc* [en línea], 2017, (México), vol. 21 (2), págs. 41-53. [Consulta: 5 septiembre 2023]. ISSN 2448-8364. Disponible en: <http://www.redalyc.org/articulo.oa?id=46753192008>
47. **ZABALA, Luis.** Desarrollo de un sistema de adquisición y procesamiento de señales electromiográficas con matlab para la identificación de los movimientos extensión, flexión, pronación y supinación de un brazo humano. (Trabajo de titulación). Escuela Superior Politécnica de Chimborazo. Riobamba-Ecuador. 2017. págs. 7-10.



ANEXOS

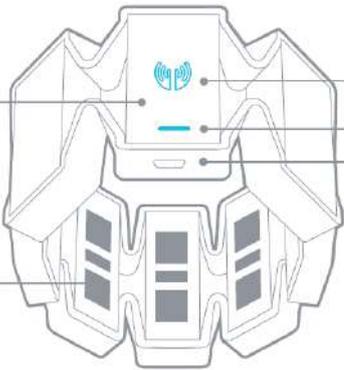
ANEXO A: PRIMEROS PASOS CON EL SENSOR MYO ARMBAND




WHAT DOES THE MYO (MY-OH) ARMBAND DO?

The Myo armband reads the muscle activity in your forearm and gives you touch-free control of technology with hand gestures and motion.

Tech Specs



- Nine-axis IMU tracks motions and rotations
- Muscle activity sensors detect hand gestures
- Dual indicator LEDs
- Standard Micro-USB recharging

Comfortable design allows for one-size fits all
Expandable between 7.5in - 13in (19-34 cm)
Built-in rechargeable lithium ion battery
Bluetooth Smart connection
For ages 12+

COMPATIBILITY



Windows



Mac



iOS



Android

GESTURES



DOUBLE TAP



WAVE RIGHT



WAVE LEFT



SPREAD FINGERS



MAKE A FIST



ROTATE



MOVE

ANEXO B: HOJA DE DATOS DEL ARDUINO UNO

Arduino UNO R3

Features

- ATMega328P Processor**
 - Memory**
 - AVR CPU at up to 16 MHz
 - 32KB Flash
 - 2KB SRAM
 - 1KB EEPROM
 - Security**
 - Power On Reset (POR)
 - Brown Out Detection (BOD)
 - Peripherals**
 - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
 - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
 - 1x USART with fractional baud rate generator and start-of-frame detection
 - 1x controller/peripheral Serial Peripheral Interface (SPI)
 - 1x Dual mode controller/peripheral I2C
 - 1x Analog Comparator (AC) with a scalable reference input
 - Watchdog Timer with separate on-chip oscillator
 - Six PWM channels
 - Interrupt and wake-up on pin change
- ATMega16U2 Processor**
 - 8-bit AVR® RISC-based microcontroller
- Memory**
 - 16 KB ISP Flash
 - 512B EEPROM
 - 512B SRAM
 - debugWIRE interface for on-chip debugging and programming
- Power**
 - 2.7-5.5 volts

Arduino UNO R3

5 Connector Pinouts

The diagram shows the physical layout of the Arduino Uno R3 board with color-coded labels for each connector:

- DC Power:** Includes pins for +5V, GND, and VIN.
- Digital:** Lists pins from D0 to D13, including headers for SPI (MISO, MOSI, SCK), I2C (SDA, SCL), and UART (RX, TX).
- Analog:** Lists pins A0 through A5.
- USB:** Shows the USB Type-B connector with pins D+, D-, VCC, and GND.
- Power:** Shows the ATX power connector with pins +5V, GND, and Pin 1.

Arduino UNO R3

5.1 ANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	I/REF	I/REF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+5V	Power	+5V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 DIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

6 Certifications

6.1 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

ROHS 2 Directive 2011/65/EU	
Conforms to:	EN50581:2012
Directive 2014/35/EU (LVD)	
Conforms to:	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
Directive 2004/40/EC & 2008/46/EC & 2013/35/EU, EMF	
Conforms to:	EN 62311:2008

6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl) phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

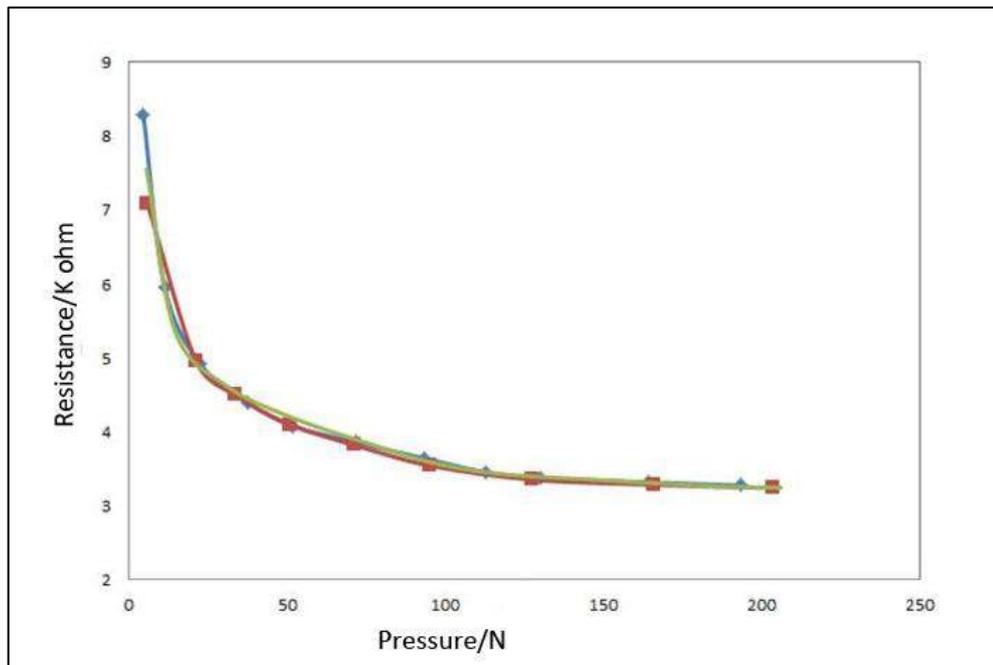
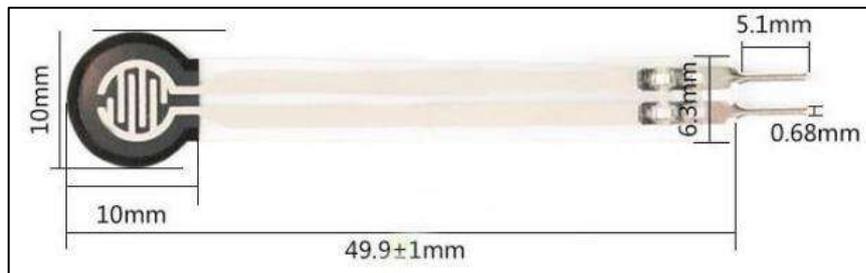
Exemptions: No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907/2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (<https://echa.europa.eu/web/guest/candidate-list-table>), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907/2006/EC.

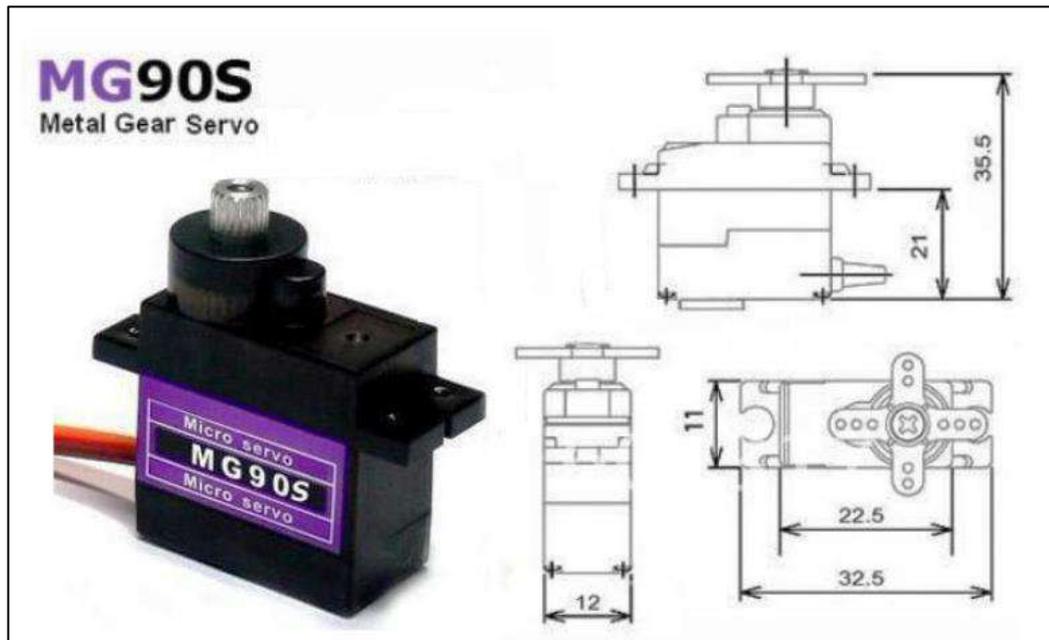
ANEXO C: DATOS DEL SENSOR RFP 602

Product Parameters:

Sensor Name: RFP Membrane Pressure Sensor
Model: RFP-602e
Sensor type: single point
Sensitive area shape: round
Total length of sensor: 10mm (± 1 mm) in diameter
Sensor thickness: 0.2mm
Terminal Type: Male Terminal
Sensor principle: piezoresistive
Service life: > 500,000 times
Response time: μ s level
Working temperature: $-25^{\circ}\text{C} \sim 70^{\circ}\text{C}$
Film material: polyester
Range: 100g, 500g, 1kg, 2kg, 5kg, 10kg, 20kg

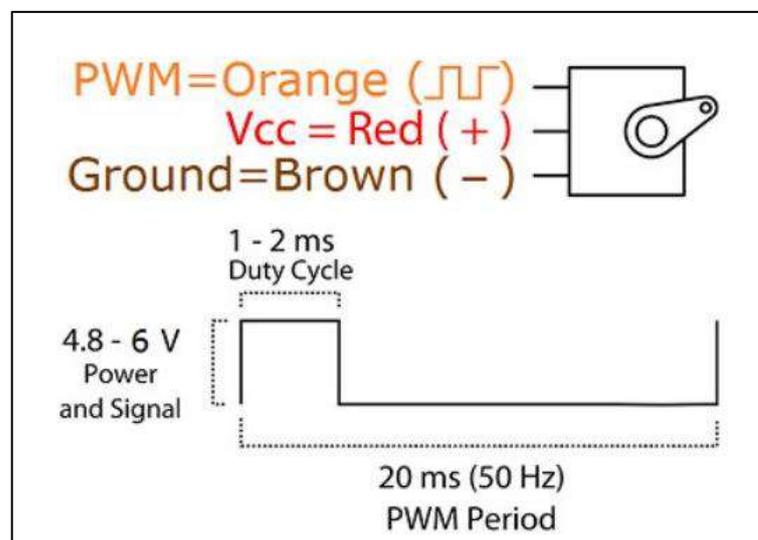


ANEXO D: HOJA DE DATOS DEL SERVOMOTOR MG90S



CARACTERÍSTICAS:

- Voltaje de operación: 4.8~6v
- Velocidad de operación: 0.1 s/60° (4.8v), 0.08 s/60° (6v)
- Torque detenido: 1.8 kg/cm (4.8v), 2.2 kg/cm (6v)
- Con cojinete
- Banda muerta: 5 μ s
- Peso ligero: 14 g
- Dimensiones compactas: Largo 22.5 mm, ancho 12 mm, altura 35.5 mm aprox.
- Largo del cable: 25 cm aprox.
- Piñonería metálica



ANEXO E: DISEÑO DE LA INTERFAZ GRÁFICA DE USUARIO GUI

```

1 classdef Tesis_Alvaro_Casierro_exported < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure          matlab.ui.Figure
6         Panel            matlab.ui.container.Panel
7         TabGroup         matlab.ui.container.TabGroup
8         InicioTab       matlab.ui.container.Tab
9         SEMG            matlab.ui.control.UIAxes
10        Fecha           matlab.ui.control.DatePicker
11        Image           matlab.ui.control.Image
12        Image_2         matlab.ui.control.Image
13        PuertoPanel     matlab.ui.container.Panel
14        LabelP          matlab.ui.control.Label
15        LabelNombre     matlab.ui.control.Label
16        Start           matlab.ui.control.Button
17        Stop            matlab.ui.control.Button
18        TextAreaTitulo  matlab.ui.control.TextArea
19        PruebadeFuncionamientoButton matlab.ui.control.Button
20        Lamp            matlab.ui.control.Lamp
21        Conectar       matlab.ui.control.Button
22        tiempo          matlab.ui.control.NumericEditField
23        Image2          matlab.ui.control.Image
24        Image3          matlab.ui.control.Image
25        Image4          matlab.ui.control.Image
26        Image5          matlab.ui.control.Image
27        Graficas        matlab.ui.container.Tab
28        RFP1            matlab.ui.control.UIAxes
29        RFP2            matlab.ui.control.UIAxes
30
31        RFP3            matlab.ui.control.UIAxes
32    end
33
34    properties (Access = private)
35
36        comu % Description
37
38    end
39
40    % Callbacks that handle component events
41    methods (Access = private)
42
43        % Code that executes after component creation
44        function startupFcn(app)
45
46            % Parámetros para la comunicación serial
47
48
49            app.Image2.Visible = 0;
50            app.Image3.Visible = 0;
51            app.Image4.Visible = 0;
52            app.Image5.Visible = 0;
53            app.Lamp.Color = 'red';
54
55        end
56
57        % Close request function: UIFigure
58
59        function UIFigureCloseRequest(app, event)
60
61            delete(app)
62
63        end
64
65        % Value changed function: Fecha
66        function FechaValueChanged(app, event)
67            value = app.Fecha.Value;
68            disp(value);
69        end
70
71        % Button pushed function: Conectar
72        function ConectarButtonPushed(app, event)
73
74            global t
75            app.comu = serialport('COM3',115200);
76            tic;
77
78            sdk_path = 'C:\Program Files\myo-sdk-win-0.9.0\myo-sdk-win-0.9.0';
79            build_myo_mex(sdk_path);
80
81            app.Lamp.Color = 'green';
82
83            t = toc;
84            app.tiempo.Value = t;
85            set(app.LabelP,'Text',app.comu.Port);
86
87        end
88
89        % Button pushed function: Stop
90        function StopButtonPushed(app, event)
91            global cerrar;
92
93            cerrar = 0;
94            app.comu = [];
95            app.Lamp.Color = 'red';
96            app.tiempo.Value = 0;
97            set(app.LabelP,'Text','No conectado');
98            app.Image2.Visible = 0;
99            app.Image3.Visible = 0;
100           app.Image4.Visible = 0;
101           app.Image5.Visible = 0;
102           clear all;
103
104        end
105
106        % Callback function: Image2, Start
107        function StartButtonPushed(app, event)
108            global cerrar const RFP1 RFP2 RFP3
109
110            cerrar = 1;
111            countMyos = 1;
112            nm = MyoMex(countMyos);
113
114
115            m1 = nm.myoData(1);
116            pause(0.1);
117
118            i = 1;
119
120            while cerrar
121
122                TiempoEMG = m1.timeEMG_log;
123                EMG = m1.emg_log;
124                Pose = m1.pose;
125                Pose_des = m1.pose_rest;
126                Pose_puno = m1.pose_fist;
127                Pose_adentro = m1.pose_wave_in;
128                Pose_afuera = m1.pose_wave_out;
129                Pose_cerrar = m1.pose_fingers_spread;
130
131                if Pose_des == 1
132                    app.Image5.Visible = 0;
133                    drawnow
134                    app.Image4.Visible = 0;
135                    drawnow
136                    app.Image3.Visible = 0;
137                    drawnow
138                    app.Image2.Visible = 0;
139                    drawnow
140                    write(app.comu,'1','char');
141
142                end
143
144                if Pose_puno == 1
145                    app.Image5.Visible = 1;
146                    drawnow
147                    app.Image4.Visible = 0;
148                    drawnow
149                    app.Image3.Visible = 0;
150                    drawnow
151                    app.Image2.Visible = 0;
152                    drawnow
153                    write(app.comu,'2','char');
154
155                    const(i) = 1;
156                    RFP1(i) = str2double(readline(app.comu));
157                    pause(0.1);
158                    RFP2(i) = str2double(readline(app.comu));
159                    pause(0.1);
160                    RFP3(i) = str2double(readline(app.comu));
161                    pause(0.1);
162
163                    plot(app.RFP1,RFP1);
164                    plot(app.RFP2,RFP2);
165                    plot(app.RFP3,RFP3);
166
167                    i = i+1;
168
169                end
170            end

```

```

170 -         if Pose_afuera == 1
171 -             app.Image5.Visible=0;
172 -             drawnow
173 -             app.Image4.Visible=0;
174 -             drawnow
175 -             app.Image3.Visible=1;
176 -             drawnow
177 -             app.Image2.Visible=0;
178 -             drawnow
179 -             write(app.comu,'3','char');
180
181 -             const(i) = 1;
182 -             RFP1(i) = str2double(readline(app.comu));
183 -             pause(0.1);
184 -             RFP2(i) = str2double(readline(app.comu));
185 -             pause(0.1);
186
187 -             plot(app.RFP1,RFP1);
188 -             plot(app.RFP2,RFP2);
189
190 -             i = i+1;
191
192 -         end
193
194 -         if Pose_adentro == 1
195 -             app.Image5.Visible = 0;
196 -             drawnow
197 -             app.Image4.Visible = 0;

```

```

198 -             drawnow
199 -             app.Image3.Visible = 0;
200 -             drawnow
201 -             app.Image2.Visible = 1;
202 -             drawnow
203 -             write(app.comu,'4','char');
204
205 -             const(i) = 1;
206 -             RFP1(i) = str2double(readline(app.comu));
207 -             pause(0.1);
208
209 -             plot(app.RFP1,RFP1);
210
211 -             i = i+1;
212
213 -         end
214
215 -         if Pose_cerrar == 1
216 -             app.Image5.Visible = 0;
217 -             drawnow
218 -             app.Image4.Visible = 1;
219 -             drawnow
220 -             app.Image3.Visible = 0;
221 -             drawnow
222 -             app.Image2.Visible = 0;
223 -             drawnow
224
225 -         break;

```

```

226 -
227 -         end
228
229 -         plot(app.SEMG,TiempoEMG,EMG);
230 -         drawnow
231
232 -     end
233
234
235 -     end
236
237 -     % Button pushed function: PruebadeFuncionamientoButton
238 -     function PruebadeFuncionamientoButtonPushed(app, event)
239
240 -         write(app.comu,'5','char');
241
242 -     end
243
244 - end

```

ANEXO F: PROGRAMACIÓN DEL SISTEMA DE CONTROL PID

```

1 // Librerías necesarias
2 #include <PID_v1.h>
3 #include <Servo.h>
4 #include <SoftwareSerial.h>
5
6 // Definir los pines de las entradas analógicas
7 #define INPUT1 0
8 #define INPUT2 1
9 #define INPUT3 2
10
11 // Crear las instancias Servo para cada servomotor
12 Servo servoMotor1;
13 Servo servoMotor2;
14 Servo servoMotor3;
15 Servo servoMotor4;
16 Servo servoMotor5;
17
18 // Establecer los valores de las constantes del PID
19 double Kp=1, Ki=0.5, Kd=0;
20
21 // Establecer las variables de entrada y salida
22 // Inicializar los 3 controladores PID
23 double Setpoint1, Input1, Output1, Output1_1;
24 double in1, in1_1;
25 PID myPID1(&Input1, &Output1, &Setpoint1, Kp, Ki, Kd, DIRECT);
26
27 double Setpoint2, Input2, Output2, Output2_2;
28 double in2, in2_2;
29 PID myPID2(&Input2, &Output2, &Setpoint2, Kp, Ki, Kd, DIRECT);
30
31 double Setpoint3, Input3, Output3, Output3_3;
32 double in3, in3_3;
33 PID myPID3(&Input3, &Output3, &Setpoint3, Kp, Ki, Kd, DIRECT);
34
35
36 void setup() {
37
38 // Definir los valores de los setpoints
39 Setpoint1 = 200;
40 Setpoint2 = 300;
41 Setpoint3 = 400;
42
43 // Establecer los PID como AUTOMATIC
44 myPID1.SetMode(AUTOMATIC);
45 myPID2.SetMode(AUTOMATIC);
46 myPID3.SetMode(AUTOMATIC);
47
48 // Inicializar la comunicación serial
49 Serial.begin(115200);
50 Serial.setTimeout(10);
51
52 pinMode(8,OUTPUT);
53
54 // Definir los pines de salida para cada servomotor
55 servoMotor1.attach(2);
56 servoMotor2.attach(3);
57 servoMotor3.attach(4);
58 servoMotor4.attach(5);
59 servoMotor5.attach(6);
60
61 // Colocar los servomotores en 0° al iniciar
62 servoMotor1.write(0);
63 servoMotor2.write(0);
64 servoMotor3.write(0);
65 servoMotor4.write(0);
66 servoMotor5.write(0);
67
68 delay(1000);
69
70 }
71
72 void loop() {
73
74 // Verificar que exista comunicación serial
75 while(Serial.available()){
76
77 // Leer la información recibida por comunicación serial
78 char dato=Serial.read();
79 delay(10);
80
81 if(dato=='1'){ // Poner los servomotores en 0°, reposo
82
83     digitalWrite(8,LOW);
84
85     servoMotor1.write(0);
86     servoMotor2.write(0);
87     servoMotor3.write(0);
88     servoMotor4.write(0);
89     servoMotor5.write(0);
90
91 }
92
93 if(dato=='2'){ // Realizar el agarre de objetos con 5 dedos
94
95     digitalWrite(8,HIGH);
96
97     in1=analogRead(INPUT1); // Leer la entrada A0
98     // Mapear los valores de entrada de 0 a 1023
99     in1_1=map(in1,1023,0,0,1023);
100
101     Input1 = in1_1;
102     myPID1.Compute(); // Se realiza el computo del PID
103     // Mapear los valores de salida de 0° a 180°
104     Output1_1=map(Output1, 0, 255, 0, 180);
105
106     in2=analogRead(INPUT2); // Leer la entrada A1
107     in2_2=map(in2,1023,0,0,1023);
108     Input2 = in2_2;
109     myPID2.Compute();
110     Output2_2=map(Output2, 0, 255, 0, 180);
111
112     in3=analogRead(INPUT3); // Leer la entrada A3
113     in3_3=map(in3,1023,0,0,1023);
114     Input3 = in3_3;
115     myPID3.Compute();
116     Output3_3=map(Output3, 0, 255, 0, 180);
117
118 // Colocar los valores del PID en los servomotores
119 servoMotor1.write(Output1_1);
120 servoMotor2.write(Output2_2);
121 servoMotor3.write(Output3_3);
122 servoMotor4.write(Output3_3);
123 servoMotor5.write(Output3_3);
124
125 // Enviar los valores de los sensores por comunicación serial
126 Serial.println(in1_1);
127 delay(100);
128 Serial.println(in2_2);
129 delay(100);
130 Serial.println(in3_3);
131 delay(100);
132 }

```

```

133
134 if(dato=='3'){ // Realizar el agarre de objetos con 3 dedos
135
136     digitalWrite(8,HIGH);
137
138     in1=analogRead(INPUT1); // Leer la entrada A0
139     // Mapear los valores de entrada de 0 a 1023
140     in1_1=map(in1,1023,0,0,1023);
141     Input1 = in1_1;
142     myPID1.Compute(); // Se realiza el computo del PID
143     // Mapear los valores de salida de 0° a 180°
144     Output1_1=map(Output1, 0, 255, 0, 180);
145
146     in2=analogRead(INPUT2); // Leer la entrada A1
147     in2_2=map(in2,1023,0,0,1023);
148     Input2 = in2_2;
149     myPID2.Compute();
150     Output2_2=map(Output2, 0, 255, 0, 180);
151
152     // Colocar los valores del PID en los servomotores
153     servoMotor1.write(Output1_1);
154     servoMotor2.write(Output1_1);
155     servoMotor3.write(Output2_2);
156
157     // Enviar los valores de los sensores por comunicación serial
158     Serial.println(in1_1);
159     delay(100);
160     Serial.println(in2_2);
161     delay(100);
162
163 }
164
165 if(dato=='4'){ // Realizar el agarre de objetos con 2 dedos

```

```

166
167     digitalWrite(8,HIGH);
168
169     in1=analogRead(INPUT1); // Leer la entrada A0
170     // Mapear los valores de entrada de 0 a 1023
171     in1_1=map(in1,1023,0,0,1023);
172     Input1 = in1_1;
173     myPID1.Compute(); // Se realiza el computo del PID
174     // Mapear los valores de salida de 0° a 180°
175     Output1_1=map(Output1, 0, 255, 0, 180);
176
177     // Colocar los valores del PID en los servomotores
178     servoMotor1.write(Output1_1);
179     servoMotor2.write(Output1_1);
180
181     // Enviar los valores de los sensores por comunicación serial
182     Serial.println(in1_1);
183     delay(100);
184
185 }
186
187 // Realizar la prueba de funcionamiento de los servomotores
188 if(dato=='5'){
189     digitalWrite(8,LOW);
190
191
192     // Poner los servomotores en posición ok
193     servoMotor1.write(180);
194     servoMotor2.write(180);
195     servoMotor3.write(0);
196     servoMotor4.write(0);
197     servoMotor5.write(0);
198     delay(2000);

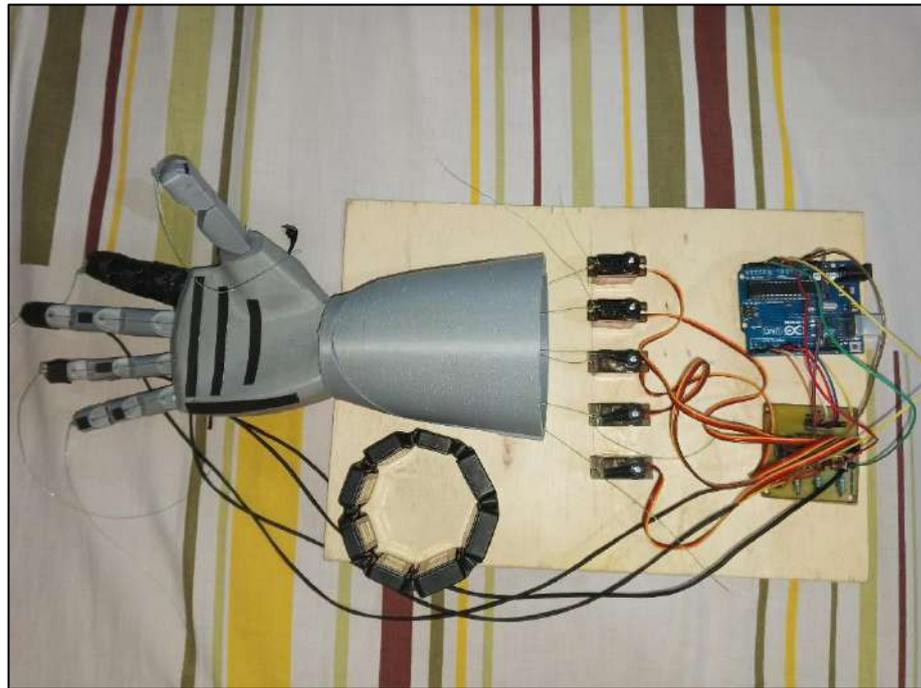
```

```

199
200     // Poner los servomotores en posición Amor y Paz
201     servoMotor1.write(180);
202     servoMotor2.write(0);
203     servoMotor3.write(0);
204     servoMotor4.write(180);
205     servoMotor5.write(180);
206     delay(2000);
207
208     // Poner los servomotores en posición Spiderman
209     servoMotor1.write(0);
210     servoMotor2.write(0);
211     servoMotor3.write(180);
212     servoMotor4.write(180);
213     servoMotor5.write(0);
214     delay(2000);
215
216 }
217
218 }
219
220 }
221
222

```

ANEXO G: SISTEMA BIOELECTRÓNICO FINAL



ANEXO H: MANUAL DE USUARIO



Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica para la rehabilitación de personas con amputación transradial.

Manual de Usuario

Versión: 1.0
Fecha: 25/03/2024

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito por la Escuela Superior Politécnica de Chimborazo.



Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica

Alvaro Gabriel Casaña Cotera

HOJA DE CONTROL

Organismo	Escuela Superior Politécnica de Chimborazo		
Proyecto	Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica para la rehabilitación de personas con amputación transradial		
Entregable	Manual de Usuario		
Autor	Alvaro Gabriel Casaña Cotera		
Versión	1.0	Fecha versión	25/03/2024
Aprobado por		Fecha aprobación	25/03/2024
		Nº páginas	15

Requerimientos mínimos:

- Sistema operativo:
 - Windows: Windows 7 SP1 en adelante, Windows Server 2008 SP2 en adelante.
 - Mac: macOS 10.10 - 10.11
 - Linux: Kernel 2.6 o superior, glibc 2.11 o superior.
- Procesador: Intel o AMD x86-64 con soporte de instrucciones AVX2.
- Disco: 2 GB solo para MATLAB, 4.6 GB para una instalación típica.
- RAM: 1GB mínimo, 4 GB recomendado.
- Tarjeta gráfica: Soporte para OpenGL 3.3 recomendado con 1 GB en GPU.



Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica

Alvaro Gabriel Casaña Cotera

ÍNDICE

Contenido

1. DESCRIPCIÓN DEL SISTEMA	4
1.1 Objetivo.....	4
1.2 Funcionalidad.....	4
1.3 Consideraciones.....	4
2. INSTALACIÓN DE COMPLEMENTOS	4
2.1 Myo Connect Installer.....	4
2.2 MyoMex-master y myo-sdk-win.....	4
2.3 MATLAB Support for MinGW-w64 C/C++ Compiler.....	4
2.4 MATLAB Support Package for Arduino Hardware.....	4
2.5 Set Path.....	4
2.6 Librería PID by Brett Beaurogard.....	4
2.7 Arduino.....	4
3. USO DEL SISTEMA BIOELECTRÓNICO	4
3.1 Myo ArmBand.....	5
3.2 GUI.....	5
3.3 Procedimiento.....	5



Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica

Alvaro Gabriel Casaña Cotera

1. DESCRIPCIÓN DEL SISTEMA

1.1 Objetivo
Desarrollar un sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica en la rehabilitación de personas con amputación transradial.

1.2 Funcionalidad
El sistema consiste en realizar el agarre controlado de objetos mediante una prótesis de miembro superior. Esto lo logra al recibir los señales mioeléctricas provenientes del sensor Myo ArmBand y de los sensores de presión RFP-602. Una vez recibidas y procesadas las señales se las muestra gráficamente a través de una interfaz gráfica de usuario GUI en tiempo real.

1.3 Consideraciones
Tenar previamente instalado los softwares Matlab 2020a o superior y Arduino IDE 2.1.0 o superior.
La carpeta "Sistema_bioelectronico" será proporcionada con el Manual de Usuario.

2. INSTALACIÓN DE COMPLEMENTOS

2.1 Myo Connect Installer
Abrir la carpeta "Sistema_bioelectronico" y dar clic derecho en el archivo ejecutable "myo_connect_installer" y luego se ejecuta como administrador.

- matMYO-master
- MyoMex-master
- myo-sdk-win-0.9.0
- Tesis_graficas
- arduino.mipkginstall
- mingw.mipkginstall
- myo_connect_installer
- Tesis_Alvaro_Casiera_exported.m

Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica Alvaro Gabriel Casiera Cotera

espóch

Abrir

- Ejecutar como administrador
- Compartir con Skype
- Solucionar problemas de compatibilidad
- Anclar a Inicio

Aceptar y seguir el proceso de instalación.

2.2 MyoMex-master y myo-sdk-win

Abra la carpeta "Sistema_bioelectrónico", seleccione las carpetas "matMYO-master", "MyoMex-master" y "myo-sdk-win-0.9.0" y copie:

- matMYO-master
- MyoMex-master
- myo-sdk-win-0.9.0
- Tesis_graficas
- arduino.mipkginstall
- mingw.mipkginstall
- myo_connect_installer
- Tesis_Alvaro_Casiera_exported.m

En el Disco Local (C:) y luego a "Archivos de programa" y pegue:

- matMYO-master
- Microsoft
- Microsoft Office
- Microsoft Office 15
- Microsoft OneDrive
- Microsoft Update Health Tools
- ModifiableWindowsApps
- MyoMex-master
- myo-sdk-win-0.9.0

Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica Alvaro Gabriel Casiera Cotera

espóch

2.3 MATLAB Support for MinGW v64 C/C++ Compiler

Abra el software MATLAB, busque y seleccione la carpeta "Sistema_bioelectrónico" en la current folder:

Current Folder

- Name -
- matMYO-master
- MyoMex-master
- myo-sdk-win-0.9.0
- Tesis_graficas
- arduino.mipkginstall
- mingw.mipkginstall
- myo_connect_installer.exe
- Tesis_Alvaro_Casiera_exported.m

De doble clic sobre el archivo "mingw.mipkginstall" y proceda a la instalación automáticamente.

2.4 MATLAB Support Package for Arduino Hardware

Vea una vez en el software MATLAB, busque y seleccione la carpeta "Sistema_bioelectrónico" en la current folder:

Current Folder

- Name -
- matMYO-master
- MyoMex-master
- myo-sdk-win-0.9.0
- Tesis_graficas
- arduino.mipkginstall
- mingw.mipkginstall
- myo_connect_installer.exe
- Tesis_Alvaro_Casiera_exported.m

De doble clic sobre el archivo "arduino.mipkginstall" y proceda a la instalación automáticamente.

2.5 Set Path

Una vez instalados los complementos en MATLAB, diríjase a la opción "Set Path" en la pantalla Home.

Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica Alvaro Gabriel Casiera Cotera

espóch

Preferences Add-Ons Help Request Support

Set Path

Parallel Change the search path used by MATLAB to look for files

ENVIRONMENT RESOURCES

Una vez dentro seleccionar la opción "Add with Subfolders".

Set Path

All changes take effect (immediate)

Add Folder...

Add with Subfolders...

Buscar la carpeta la carpeta "Sistema_bioelectrónico", seleccione las carpetas "matMYO-master", "MyoMex-master" y "myo-sdk-win-0.9.0", una vez seleccionadas y agregadas dar clic en "Save".

Remove Save Close Abort Default Help

2.6 Librería PID by Brett Beauregard

Abra el software Arduino IDE, diríjase al "Gestor de bibliotecas", busque "PID by Brett Beauregard" y dar clic en instalar.

Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica Alvaro Gabriel Casiera Cotera

espóch

Arduino Uno

GESTOR DE BIBLIOTECAS

PID

Tipo: Todo

Tema: Todo

PID by Brett Beauregard

1.2 Instalado

PID controller: A PID controller needs to keep some input variable close to a desired setpoint by adjusting an...

Más información

2.7 Arduino

Una vez instaladas las librerías, ir a Archivo y luego en Abrir:

Archivo Editar Sketch Herramientas Ayuda

New Sketch Ctrl+N

Nuevo Sketch en la Nube Alt+Ctrl+N

Abrir... Ctrl+O

Buscar la carpeta "Sistema_bioelectrónico" y seleccionar el archivo "Tesis_graficas".

Tesis_graficas

Seleccionar la placa Arduino UNO y el puerto COM correspondiente y cargar el programa.

Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica

Alvaro Gabriel Casiera Colera

espóch

Archivo Editar Sketch Herramientas Ayuda

Arduino Uno

Id. Cargar

```

1 // Librerías necesarias
2 #include <PID_V1.h>
3 #include <Servo.h>
4 #include <SoftwareSerial.h>

```

3. USO DEL SISTEMA BIOELECTRÓNICO

3.1 Myo Armband

Para encender el sensor debemos conectarlo a través de su puerto microUSB.



Colocar el sensor en el antebrazo, debajo del codo, con el Logo encima del artículo braquiocrural, como se muestra a continuación.

Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica

Alvaro Gabriel Casiera Colera

espóch



Colocar el plug de conexión Bluetooth en un puerto del computador y abrir la aplicación Myo Connect, ubicada en la barra de tareas, en ícono oculto, dar clic derecho y seleccionar "Armband Manager".

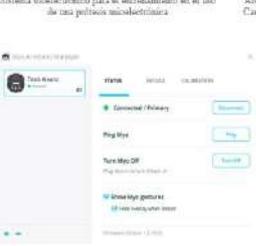


Dar clic en conectar y sincronizar el sensor.

Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica

Alvaro Gabriel Casiera Colera

espóch



3.2 GUI

Dirigirse a MATLAB y abrir el archivo "Tesis_Alvaro_Casiera_exported.m", que se encuentra en la carpeta "Sistema_bioelectrónico".

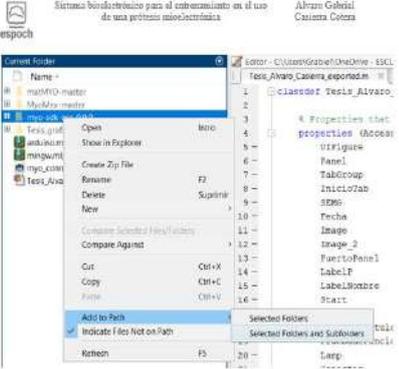
- mathMIO-master
- MyoMex-master
- myo-sdk-win-0.9.0
- Tesis_graficas
- arduino.mipkginstall
- mingw.mipkginstall
- myo_connect_installer
- Tesis_Alvaro_Casiera_exported.m**

Ir a la current folder, buscar la carpeta "myo-sdk-win-0.9.0". Dar clic derecho y seleccionar "Add to Path" y luego "Selected Folders and subfolders".

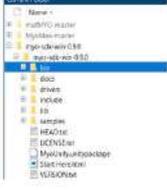
Sistema bioelectrónico para el entrenamiento en el uso de una prótesis mioeléctrica

Alvaro Gabriel Casiera Colera

espóch



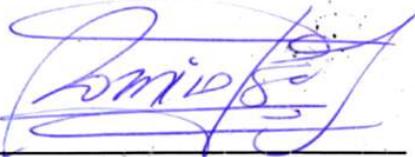
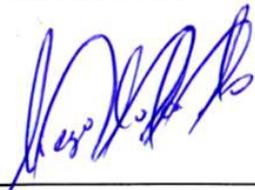
Una vez agregada, expandir la carpeta, seleccionar la carpeta "bin" y dar doble clic.





ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
CERTIFICADO DE CUMPLIMIENTO DE LA GUÍA PARA
NORMALIZACIÓN DE TRABAJOS DE FIN DE GRADO

Fecha de entrega: 27 / 03 / 2024

INFORMACIÓN DE LOS AUTORES
Nombres – Apellidos: Alvaro Gabriel Casierra Cotera
INFORMACIÓN INSTITUCIONAL
Facultad: Informática y Electrónica
Carrera: Electrónica y Automatización
Título a optar: Ingeniero en Electrónica y Automatización
 _____ Dr. Ramiro Fernando Isa Jara. Director del Trabajo de Integración Curricular
 _____ Ing. Diego Ramiro Ñacato Estrella Asesor del Trabajo de Integración Curricular

