



**SISTEMA DE ASISTENCIA PARA LA
REHABILITACIÓN O
EJERCITACIÓN DE UNA MANO**
Universidad de Pamplona

Autor: John Edison Sánchez Fernández



SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Autor

JOHN EDISON SÁNCHEZ FERNÁNDEZ

Optante al título de ingeniero en Mecatrónica

Director

Ph.D. CÉSAR AUGUSTO PEÑA CORTÉS

Ingeniero Electromecánico
Doctor en Automática y Robótica

FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍA MECÁNICA, MECATRÓNICA E
INDUSTRIAL
INGENIERÍA MECATRÓNICA



UNIVERSIDAD DE PAMPLONA
PAMPLONA, Mayo 29 de 2018

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



AGRADECIMIENTOS

Toda meta inicia, se mantiene y se alcanza con el apoyo, motivación e inspiración de quienes nos rodean y alientan a seguir adelante, como no agradecer a Dios que guío mis pasos en el camino correcto, me da la sabiduría y la fortaleza para no desfallecer en cada aspecto de mi vida.

A mis padres Luis Sánchez y Marina Fernández quienes formaron mi carácter con su ejemplo y dedicación, quienes a pesar de los obstáculos que se presentaron nunca dejaron de darme su apoyo incondicional y con su gran esfuerzo me brindaron los medios posibles para completar mi formación profesional.

A mi hermana Paola Sánchez la motivación para ser un buen ejemplo y perseverar sin bajar la cabeza, por su atención constante y completa disposición para escucharme y darme su consejo oportuno y sincero.

A mi novia, amiga y compañera Tatiana Prieto por recordarme lo importante de creer en mí y ayudarme con su punto de vista objetivo y certero para encontrar la mejor solución a cada detalle de esta investigación, por su amor y su maravillosa compañía durante estos años.

Al PhD. César Peña director de esta investigación, quien me brindo su tiempo y compartió sus conocimientos para enfocar las herramientas disponibles que permitieron alcanzar los objetivos planteados con el fundamento y la innovación propios de la ingeniería.

A mis profesores guías en el proceso de aprendizaje y a mis amigos y compañeros, que fueron parte de la experiencia universitaria y son parte esencial de mi formación como ingeniero.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Tabla de contenido.

Capítulo I.....	9
1.1 RESUMEN	9
1.2 ABSTRACT.....	10
1.3 OBJETIVOS.....	11
1.4 INTRODUCCIÓN.....	12
1.5 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	13
1.6 GLOSARIO	14
Capítulo II.....	16
2.1 MARCO TEÓRICO	16
2.2 ESTADO DEL ARTE	28
Capítulo III.....	38
DESARROLLO DE LA INVESTIGACIÓN.....	38
3.1 CRITERIOS DE DISEÑO	38
3.1.1 INGENIERÍA DEL DETALLE.....	39
3.1.2 DISEÑO MECÁNICO	47
3.1.3 DISEÑO ELECTRÓNICO.....	64
3.1.4 DISEÑO DE SOFTWARE.....	66
Capítulo IV.....	85
RESULTADOS	85
4.1 DESCRIPCIÓN DEL FUNCIONAMIENTO DEL DISPOSITIVO	85
4.2 EVALUACIÓN DEL DISPOSITIVO.....	91
4.3 CONSTRUCCIÓN DEL DISPOSITIVO.....	96
Capítulo V.....	99
CONCLUSIONES.....	99
5.1 CONCLUSIONES.....	99
Capítulo VI.....	100
REFERENCIAS BIBLIOGRÁFICAS.....	100

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Departamento MMI- Ingeniería Mecatrónica



6.1 BIBLIOGRAFÍA.....	100
Capítulo VII.....	102
ANEXOS.....	102
7.1 PIEZAS IMPRESAS EN 3D.....	102
7.2 IMÁGENES Y MENÚS MOSTRADOS EN PANTALLA LCD.....	105
7.3 CODIGO DE PROGRAMACIÓN.....	109

Autor:
Ing.(c) John Edison Sánchez Fernández

Director:
PhD. César Augusto Peña Cortés.



Tabla de imágenes.

FIG. 1 MODELO DE UN SISTEMA MECATRÓNICO CONVENCIONAL. FUENTE: LOS AUTORES.	14
FIG. 2 DISPOSICIÓN HUESOS DE LA MANO. FUENTE:(NETTER, 2015).....	17
FIG. 3 ARTICULACIONES DEL DEDO DE LA MANO. FUENTE: (NETTER, 2015).....	17
FIG. 4 DISPOSICIÓN HUESOS DE LA MUÑECA. FUENTE: (NETTER, 2015).....	18
FIG. 5 ARTICULACIONES DE LA MUÑECA. FUENTE: (NETTER, 2015).....	21
FIG. 6 DISPOSICIÓN LIGAMENTOS DE LA MUÑECA. FUENTE: (NETTER, 2015).....	23
FIG. 7 MOVIMIENTOS DE EXTENSIÓN Y FLEXIÓN DE LOS DEDOS DE LA MANO. FUENTE: LOS AUTORES.....	24
FIG. 8 MOVIMIENTOS DE ABDUCCIÓN Y ADUCCIÓN DE LOS DEDOS DE LA MANO. FUENTE: LOS AUTORES.	25
FIG. 9 MOVIMIENTOS DE FLEXIÓN Y EXTENSIÓN DE LA MUÑECA. FUENTE: LOS AUTORES.	26
FIG. 10 ESQUEMA DE NIVELES DEL SISTEMA MECATRÓNICO. FUENTE LOS AUTORES.....	39
FIG. 11 DIAGRAMA DE FLUJO DEL FUNCIONAMIENTO GENERAL DEL SISTEMA. FUENTE: LOS AUTORES.	40
FIG. 12 TARJE ARDUINO MEGA 2560. FUENTE: LOS AUTORES.	42
FIG. 13 SERVOMOTOR FUTABA S3003. FUENTE: LOS AUTORES.	44
FIG. 14 SENSOR FINAL DE CARRERA. FUENTE: LOS AUTORES.....	44
FIG. 15 PANTALLA LCD TFT 2.4 PULGADAS. FUENTE: LOS AUTORES.	47
FIG. 16 MECANISMO TRONILLO SIN FIN. FUENTE: LOS AUTORES.....	48
FIG. 17 MECANISMO DE TRANSMISIÓN POR CORREA DENTADA. FUENTE: LOS AUTORES.	48
FIG. 18 MECANISMO MANIVELA BIELA CORREDERA. FUENTE: LOS AUTORES.	49
FIG. 19 RENDERIZADO MECANISMO MANIVELA BIELA CORREDERA CON DEDAL. FUENTE: LOS AUTORES.	51
FIG. 20 SIMULACIÓN DESPLAZAMIENTO MECANISMO MANIVELA BIELA CORREDERA (1) POSICIÓN INICIAL (2) POSICIÓN FINAL. FUENTE: LOS AUTORES.....	53
FIG. 21 RENDERIZADO CORREDERA Y CORREDERA CON DEDAL ACOPLADO. FUENTE: LOS AUTORES.	54
FIG. 22 RENDERIZADO DEDAL. FUENTE: LOS AUTORES.	55
FIG. 23 GRÁFICAS DE DESPLAZAMIENTO VELOCIDAD Y ACELERACIÓN DE UNA LEVA. FUENTE: (JOSEPH EDWARD SHIGLEY, 2014).....	56
FIG. 24 GRÁFICA DE DESPLAZAMIENTO DE LA LEVA DISEÑADA. FUENTE: LOS AUTORES.	58
FIG. 25 RENDERIZADO LEVA. FUENTE: LOS AUTORES.....	58
FIG. 26 RENDERIZADO CASCARON SUPERIOR. FUENTE: LOS AUTORES.	59
FIG. 27 RENDERIZADO CASCARON IZQUIERDO. FUENTE: LOS AUTORES.....	59
FIG. 28 RENDERIZADO CASCARON DERECHO. FUENTE: LOS AUTORES.....	60
FIG. 29 RENDERIZADO SOPORTE ANTEBRAZO. FUENTE: LOS AUTORES.....	60
FIG. 30 RENDERIZADO ENSAMBLE TOTAL DEL SISTEMA VISTA ISOMÉTRICA. FUENTE: LOS AUTORES.....	61
FIG. 31 RENDERIZADO ENSAMBLE TOTAL DEL SISTEMA VISTA LATERAL. FUENTE: LOS AUTORES.....	61
FIG. 32 RENDERIZADO EXPLOSIONADO DEL SISTEMA VISTA ISOMÉTRICA. FUENTE: LOS AUTORES.....	62
FIG. 33 RENDERIZADO EXPLOSIONADO DEL SISTEMA VISTA FRONTAL. FUENTE: LOS AUTORES.....	62
FIG. 34 ESQUEMA DE CONEXIONES CIRCUITO ELECTRÓNICO. FUENTE: LOS AUTORES.	64
FIG. 35 DISEÑO PCB. FUENTE: LOS AUTORES.	64
FIG. 36 ESQUEMA GENERAL DE CONEXIONES. FUENTE: LOS AUTORES.	65
FIG. 37 DIAGRAMA DE FLUJO GENERAL DELA PROGRAMACIÓN DEL SISTEMA. FUENTE: LOS AUTORES.	67
FIG. 38 DECLARACIÓN DE LIBRERÍAS. FUENTE: LOS AUTORES.....	68
FIG. 39 DECLARACIÓN DE VARIABLES. FUENTE: LOS AUTORES.	69
FIG. 40 DIAGRAMA DE FLUJO PROGRAMACIÓN IMÁGENES DE PRESENTACIÓN. FUENTE: LOS AUTORES.....	70
FIG. 41 CÓDIGO IMÁGENES DE PRESENTACIÓN. FUENTE: LOS AUTORES.....	71

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



FIG. 42 DIAGRAMA DE FLUJO SELECCIÓN TIPO DE MOVIMIENTO. FUENTE: LOS AUTORES.....	72
FIG. 43 CÓDIGO SELECCIÓN DE TIPO DE MOVIMIENTO. FUENTE: LOS AUTORES.....	73
FIG. 44 CÓDIGO ENTRADA DE DATO TÁCTIL. FUENTE: LOS AUTORES.	74
FIG. 45 DIAGRAMA DE FLUJO SELECCIÓN DE FLUIDEZ DEL MOVIMIENTO. FUENTE: LOS AUTORES.	75
FIG. 46 CÓDIGO SELECCIÓN DE TIPO DE FLUIDEZ. FUENTE: LOS AUTORES.	76
FIG. 47 DIAGRAMA DE FLUJO SELECCIÓN CANTIDAD DE REPETICIONES. FUENTE: LOS AUTORES.	77
FIG. 48 CÓDIGO SELECCIÓN CANTIDAD DE REPETICIONES. FUENTE: LOS AUTORES	78
FIG. 49 CÓDIGO FUNCIÓN DE MOVIMIENTO (1). FUENTE: LOS AUTORES.....	79
FIG. 50 CÓDIGO FUNCIÓN DE MOVIMIENTO (2). FUENTE: LOS AUTORES.....	80
FIG. 51 CÓDIGO FUNCIÓN DE MOVIMIENTO (3). FUENTE: LOS AUTORES.....	81
FIG. 52 DIAGRAMA DE FLUJO CONTINUACIÓN O FINALIZACIÓN DEL MOVIMIENTO. FUENTE. LOS AUTORES	83
FIG. 53 CÓDIGO CONTINUACIÓN O FINALIZACIÓN DEL MOVIMIENTO. FUENTE: LOS AUTORES.....	84
FIG. 54 IMAGEN DE PRESENTACIÓN ESCUDO DE LA UNIVERSIDAD DE PAMPLONA. FUENTE: LOS AUTORES.	85
FIG. 55 IMAGEN DE PRESENTACIÓN 2 LOGO INGENIERÍA MECATRÓNICA. FUENTE: JAIRO ROBERTO.....	86
FIG. 56 IMAGEN DE PRESENTACIÓN 3 INFORMACIÓN DE LOS AUTORES. FUENTE: LOS AUTORES.	86
FIG. 57 MENÚ SELECCIÓN TIPO DE MOVIMIENTO. FUENTE: LOS AUTORES.....	87
FIG. 58 MENÚ SELECCIÓN TIPO DE FLUIDEZ. FUENTE: LOS AUTORES.	87
FIG. 59 MENÚ SELECCIÓN CANTIDAD DE REPETICIONES. FUENTE: LOS AUTORES	87
FIG. 60 MENSAJE MOVIMIENTO EN PROCESO MOSTRADO EN PANTALLA. FUENTE: LOS AUTORES.....	88
FIG. 61 MENSAJE MOVIMIENTO FINALIZADO MOSTRADO EN PANTALLA. FUENTE: LOS AUTORES.....	88
FIG. 62 MENÚ REPETICIÓN DE MOVIMIENTOS. FUENTE: LOS AUTORES.	89
FIG. 63 MENÚ SELECCIÓN DE NUEVOS MOVIMIENTOS. FUENTE: LOS AUTORES.....	89
FIG. 64 MENSAJE APAGAR EL DISPOSITIVO MOSTRADO EN PANTALLA. FUENTE: LOS AUTORES.....	90
FIG. 65 DISEÑO CUESTIONARIO APLICADO A FISIOTERAPEUTAS. FUENTE: LOS AUTORES.....	92
FIG. 66 CUESTIONARIO DILIGENCIADO POR EL FISIOTERAPEUTA HUMBERTO PISCITI. FUENTE: LOS AUTORES.....	93
FIG. 67 CUESTIONARIO DILIGENCIADO POR LA FISIOTERAPEUTA SONIA MANTILLA (1). FUENTE: LOS AUTORES.....	94
FIG. 68 CUESTIONARIO DILIGENCIADO POR LA FISIOTERAPEUTA SONIA MANTILLA (1). FUENTE: LOS AUTORES.....	95
FIG. 69 ENSAMBLE DEL DISPOSITIVO CONSTRUIDO. FUENTE: LOS AUTORES.....	96
FIG. 70 MOVIMIENTO DE EXTENSIÓN DEL DEDO DE LA MANO. FUENTE: LOS AUTORES.....	97
FIG. 71 MOVIMIENTO DE FLEXIÓN DEL DEDO DE LA MANO. FUENTE: LOS AUTORES.....	97
FIG. 72 MOVIMIENTO DE FLEXIÓN DE LA MUÑECA. FUENTE: LOS AUTORES.	98
FIG. 73 MANIVELA Y BIELA DEL MECANISMO MANIVELA VIELCA CORREDERA. FUENTE: LOS AUTORES.	102
FIG. 74 CORREDERA DEL MECANISMO MANIVELA BIELA CORREDERA. FUENTE: LOS AUTORES.....	102
FIG. 75 CASCARON SUPERIOR. FUENTE: LOS AUTORES.	102
FIG. 76 CASCARON DERECHO. FUENTE: LOS AUTORES.	103
FIG. 77 CASCARON IZQUIERDO CON SERVOMOTORES Y TARJETA ARDUINO. FUENTE: LOS AUTORES.....	103
FIG. 78 ENSAMBLE CASCARON DERECHO, IZQUIERDO Y SUPERIOR VISTA ISOMÉTRICA. FUENTE: LOS AUTORES.....	104
FIG. 79 ENSAMBLE CASCARON DERECHO, IZQUIERDO Y SUPERIOR VISTA LATERAL. FUENTE: LOS AUTORES.	104
FIG. 80 ENSAMBLE DEL SISTEMA VISTA SUPERIOR SIN DEDAL. FUENTE: LOS AUTORES.	104
FIG. 81 IMAGEN DE PRESENTACIÓN 1. FUENTE: LOS AUTORES.....	105
FIG. 82 IMAGEN DE PRESENTACIÓN 2. FUENTE: LOS AUTORES.....	105

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Departamento MMI- Ingeniería Mecatrónica



FIG. 83 IMAGEN DE PRESENTACIÓN 3. FUENTE: LOS AUTORES.....	105
FIG. 84 SELECCIÓN TIPO DE MOVIMIENTO. FUENTE: LOS AUTORES.	106
FIG. 85 SELECCIÓN TIPO DE FLUIDEZ. FUENTE: LOS AUTORES.....	106
FIG. 86 SELECCIÓN CANTIDAD DE REPETICIONES. FUENTE: LOS AUTORES.	106
FIG. 87 MOVIMIENTO EN PROCESO. FUENTE: LOS AUTORES.	107
FIG. 88 MOVIMIENTO FINALIZADO. FUENTE: LOS AUTORES.	107
FIG. 89 OPCIÓN PARA REPETIR LA SERIE DE MOVIMIENTOS. FUENTE: LOS AUTORES.	108
FIG. 90 MENSAJE PARA APAGAR EL SISTEMA AL TERMINAR DE USARLO. FUENTE: LOS AUTORES.	108

Autor:
Ing.(c) John Edison Sánchez Fernández

Director:
PhD. César Augusto Peña Cortés.



Índice de tablas.

TABLA 1 COMPARACIÓN DE TARJETAS DE CONTROL. FUENTE: LOS AUTORES.	41
TABLA 2 CARACTERÍSTICAS ARDUINO MEGA 2560. FUENTE: LOS AUTORES.	42
TABLA 3 CARACTERÍSTICAS SERVO MOTOR FUTABA S3003. FUENTE: LOS AUTORES.	43
TABLA 4 COMPARACIÓN ENTRE PANTALLAS LCD. FUENTE: LOS AUTORES.	45
TABLA 5 CARACTERÍSTICAS PANTALLA LCD TFT 2.4 PULGADAS. FUENTE: LOS AUTORES.	46
TABLA 6 COMPARACIÓN DISEÑOS REALIZADOS. FUENTE: LOS AUTORES.	50
TABLA 7 MEDICIONES TOMADAS A MANOS DE PERSONAS COMO REFERENCIA PARA EL DISEÑO DEL PROTOTIPO. FUENTE: LOS AUTORES.	52
TABLA 8 TABLA DE VOLTAJES Y CORRIENTES. FUENTE: LOS AUTORES.	65



Capítulo I.

1.1 RESUMEN

En esta investigación se muestra el desarrollo de un sistema mecatrónico capaz de realizar movimientos de ejercitación para algunas articulaciones de la mano, que sirva como apoyo para tareas de rehabilitación practicadas por especialistas en el campo de la fisioterapia, orientadas directamente a pacientes en estado de inmovilidad prolongada como personas en estado de coma o con hemiplejía. Para cumplir con este objetivo se estudia la estructura anatómica de la mano con el fin de determinar las partes del cuerpo involucradas en los movimientos de ejercitación entre las cuales están las articulaciones interfalángicas, metacarpofalángicas, y la estructura ósea estudiando su comportamiento y disposición natural. Además de informar sobre los movimientos realizados en terapias convencionales de rehabilitación motriz para la mano.

Esta información, es la base para el inicio y desarrollo del mecanismo del sistema que cumple con la finalidad de automatizar el proceso elegido y además brinda la posibilidad al usuario de manipular opciones de tipo de movimiento y cantidad de repeticiones de los mismos programados, mediante una interfaz de usuario – maquina, que controla el inicio y desarrollo de los mismos. Mediante el uso de una pantalla LCD táctil de bajo costo, con un menú intuitivo y fácil de usar.

De igual manera, se muestran los parámetros de diseño que se toman en cuenta para la construcción final de un prototipo compacto de bajo costo, fácil de replicar y totalmente reciclable que cumple con el objetivo de lograr movimientos de flexión y extensión de un dedo de la mano, ejercitando las articulaciones interfalángicas distal, media y proximal del dedo seleccionado; y movimientos de extensión de la muñeca de la mano de manera separada o en simultaneo con series de movimientos alternados.

Los resultados obtenidos fueron evaluados por especialistas en el área de la fisioterapia los cuales dieron un concepto positivo del sistema desarrollado, avalando el cumplimiento del objetivo planteado y la utilidad del uso del dispositivo en terapias de rehabilitación motriz para los dedos de la mano.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



1.2 ABSTRACT

This research shows the development of a mechatronic system capable of exercising movements for some joints of the hand, which serves as support for rehabilitation tasks practiced by specialists in the field of physiotherapy, aimed directly at patients in a state of immobility prolonged as people in a coma or with hemiplegia. To fulfill this objective, the anatomical structure of the hand is studied in order to determine the parts of the body involved in the exercise movements, among which are the interphalangeal, metacarpophalangeal joints, and the bone structure, studying their behavior and natural disposition. In addition to reporting on movements made in conventional motor rehabilitation therapies for the hand.

This information is the basis for the start and development of the mechanism of the system that fulfills the purpose of automating the chosen process and also provides the possibility for the user to manipulate options of type of movement and number of repetitions of the same programmed, by means of a user interface - machine, which controls the start and development of the same. Through the use of a low cost LCD touch screen, with an intuitive and easy to use menu.

Likewise, the design parameters that are taken into account for the final construction of a low cost compact prototype, easy to replicate and totally recyclable that meets the objective of achieving movements of flexion and extension of a finger of the hand, exercising the distal, medial and proximal interphalangeal joints of the selected finger; and movements of extension of the wrist of the hand separately or simultaneously with series of alternating movements.

The results obtained were evaluated by specialists in the area of physiotherapy which gave a positive concept of the developed system, guaranteeing the fulfillment of the proposed objective and the usefulness of the use of the device in motor rehabilitation therapies for the fingers of the hand.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



1.3 OBJETIVOS

OBJETIVO GENERAL

Diseñar y construir un sistema mecatrónico capaz de realizar movimientos que apoyen ejercicios de rehabilitación motriz en algunas articulaciones de la mano, para personas en estado de inmovilidad.

OBJETIVOS ESPECÍFICOS

- Seleccionar un tipo de movimiento utilizado en una terapia de rehabilitación.
- Diseñar el sistema mecatrónico que permita el movimiento establecido.
- Desarrollar el sistema mecatrónico.
- Crear una interfaz que permita a los usuarios, definir el tiempo y/o repeticiones de los movimientos programados.
- Verificar el correcto funcionamiento del sistema desarrollado.



1.4 INTRODUCCIÓN

Durante el transcurso de las actividades cotidianas, los movimientos necesarios para el desarrollo de las mismas requieren el correcto funcionamiento del conjunto de músculos, huesos, articulaciones y demás componentes de las extremidades, útiles para ejecutar una determinada acción. Estos componentes se pueden ver afectados por el paso del tiempo, enfermedades o accidentes a los que cualquier persona está expuesta, que pueden causar entre otros problemas, dificultad en las funciones motrices de las personas que las sufren llegando incluso hasta la pérdida parcial o total de la capacidad de movimiento de una o más extremidades.

Sin embargo, en la actualidad existen diferentes tratamientos de rehabilitación trabajados desde el campo de la fisioterapia que permiten la recuperación paulatina de la función motora del paciente afectado. Las terapias existentes, se pueden dividir en dos tipos, basándose en la forma y en la fuerza que se debe ejercer para la ejercitación de las articulaciones de la extremidad a tratar, ya que existen una serie de ejercicios que aprovechan la capacidad del paciente para mover parcialmente el área afectada. Y por otro lado existe el caso en el cual el paciente es incapaz de realizar cualquier tipo de movimiento; por lo que los ejercicios deben ser realizados completamente por un especialista, lo que abarca un mayor tiempo de ejecución.

Con el fin de optimizar los métodos actualmente implementados para la recuperación de las personas afectadas por las razones ya expuestas, se construye un sistema mecatrónico que realiza movimientos que ayudan en tareas de ejercitación y rehabilitación motriz, a personas que se encuentren en cama debido a diferentes razones médicas, por ejemplo, personas en estado de coma. Dicho sistema se enfoca en las articulaciones de la mano, permitiendo un tipo de movimiento de estiramiento y contracción de las mismas durante periodos de tiempo previamente establecidos, por medio de una interfaz hombre - máquina.

Los movimientos realizados por el mecanismo, son completamente autónomos por lo cual no requieren de una fuerza externa por parte del usuario para que se cumpla el ciclo programado. La principal función del sistema se basa en ayudar a evitar lesiones o atrofias futuras, causadas por el estado de inmovilidad prolongada.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



1.5 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

En la actualidad las enfermedades cardiovasculares presentan una alta tasa de recurrencia, algunas de estas enfermedades son conocidas como ICTUS las cuales se tratan de una interrupción brusca del aporte de sangre a una región del cerebro en la cual, la sangre no llega al cerebro en la cantidad necesaria y por tanto tampoco lo hacen los nutrientes y el oxígeno que el cerebro necesita para funcionar correctamente (Silva, Zarruk, Quintero, Arenas, & Silva, 2006) (Asiain José, Sánchez Gema, Cueto Joxean, 2014).

Como consecuencia de este tipo de enfermedades, las personas que las sufren pueden llegar a presentar estados de cama o de inmovilidad prolongada, lo que a su vez puede causar atrofias motrices que van desde pequeñas retracciones musculares y úlceras en la piel hasta deformidades en las articulaciones expuestas al estado de inmovilidad, afectando de manera significativa la calidad de vida de los pacientes expuestos a esta condición (“Ictus – Causas y Tratamiento del Ictus,” 2018).

Con el fin de ayudar a evitar las lesiones motrices ya mencionadas, nace la idea de desarrollar un sistema mecatrónico que cumpla con tareas de apoyo en ejercicios de fisioterapia para algunas articulaciones de la mano, en el cual se integran mecanismos, electrónica y programación con los que se automatizará un tipo de movimiento adecuado para la rehabilitación de los mismos.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



1.6 GLOSARIO

- Sistema mecatrónico: Un sistema mecatrónico es aquel que recibe información mediante sensores que se encargan de convertirlos en señales medibles, las procesa para dar una orden de activación o desactivación a los respectivos actuadores involucrados, generando acciones de control sobre el sistema en el que se esté trabajando.

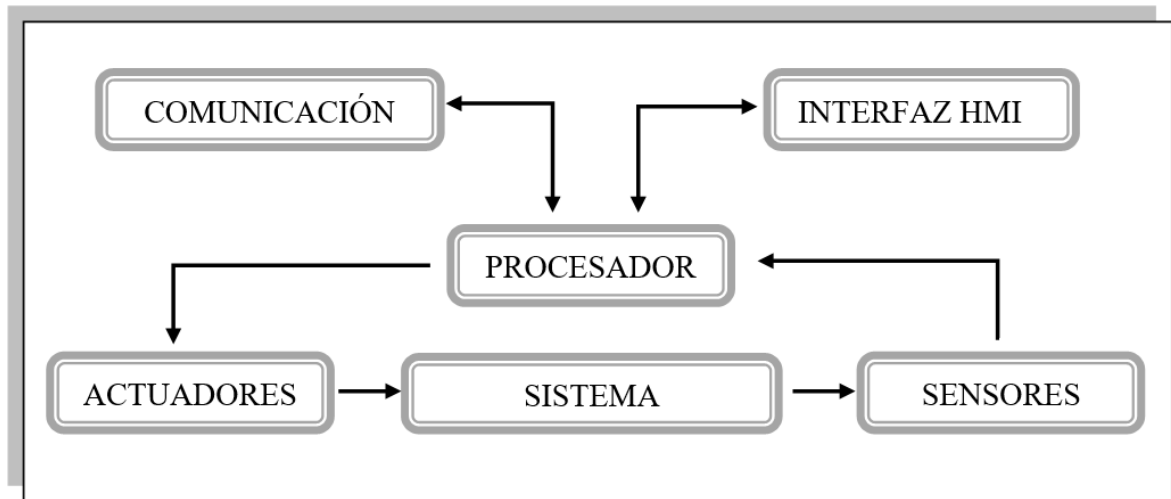


Fig. 1 Modelo de un sistema mecatrónico convencional. Fuente: los autores.

- Sensores: Un sensor es un instrumento que mide alguna variable del entorno y la traduce en señal medible. A menudo se utiliza la palabra transductor para hacer referencia al sensor. El transductor es el elemento del sensor que transforma la energía asociada a lo que se desea medir en otra forma de energía. (Escolano, 2003)
- Actuadores: Es un dispositivo mecánico que se encarga de suministrar la fuerza necesaria para mover otro dispositivo o elemento mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica.
- Procesador: El procesador o la etapa de control se encarga de realizarle el debido tratamiento a la información captada por dispositivos como los sensores o la interfaz hombre – máquina, tomando decisiones de acuerdo a la programación

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



realizada y a la vez envía las ordenes correspondientes a los actuadores del sistema.

- Sistema: Hace referencia a la planta sobre la que se está trabajando.
- Arco de movilidad: Máxima extensión con la que se mueve una articulación desde la extensión total hasta la flexión total.
- Movimiento pasivo: Movimiento dentro de la amplitud sin restricciones que realiza un segmento y que genera por entero una fuerza externa. No hay contracciones musculares voluntarias (infomed, 1999)

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Capítulo II.

2.1 MARCO TEÓRICO

Para el desarrollo de este proyecto es necesario conocer los fundamentos referentes a las articulaciones de una mano, incluyendo sus nombres, ubicación y los diferentes tipos de movimientos que pueden realizar las mismas. Basándose en sus comportamientos biomecánicos, sin entrar en detalle en temas de índole médico ya que no es el objetivo de esta investigación.

Anatomía de los dedos de la mano

El conjunto de huesos, ligamentos y articulaciones se da en una disposición anatómica permite dar a la mano una gran versatilidad en la manipulación de objetos y ajustes posicionales de acuerdo a las necesidades propias de las actividades cotidianas de los humanos. Siendo así la mano una estructura compleja en cuanto a su construcción y funcionamiento, lo que le permite ser un órgano de prensión y sensibilidad.

El esqueleto óseo de la mano consiste en 8 huesos carpianos divididos en dos filas principales. La fila proximal articulada con las porciones distales del radio y el cúbito, a excepción del pisiforme que se encuentra en disposición palmar y se articula con el triquetrum; los cuatro huesos cárpales distales están articulados con los cinco metacarpianos (Netter, 2015).

Los metacarpianos y las falanges son huesos largos. Cada uno tiene una parte tubular característica. El extremo proximal o base de cada hueso tiene una superficie articular en forma conjunta con el hueso adyacente. El extremo distal o en la cabeza de cada hueso (con excepción de las falanges distales) también tiene una superficie articular. En las falanges distales, el extremo distal se llama el penacho, no tiene una superficie articular, pero proporciona una conexión para el tejido blando de la punta digital. Las distribuciones de los huesos de la mano se muestran en la figura 2.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

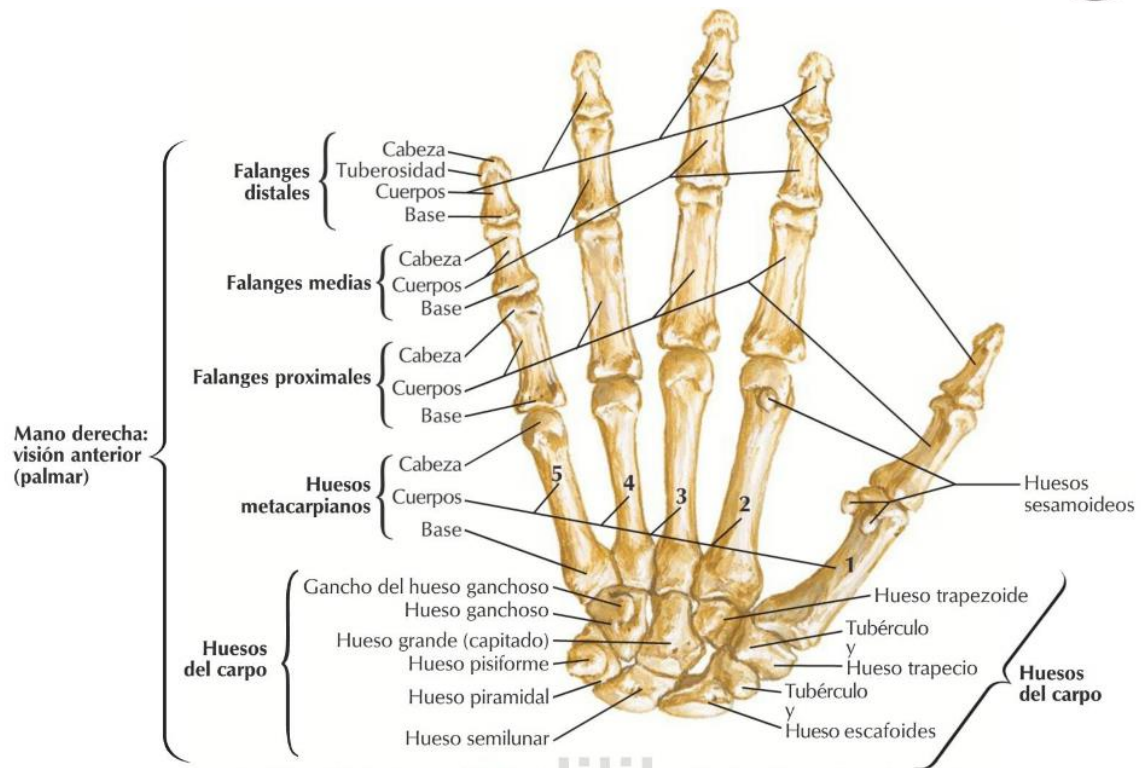


Fig. 2 disposición huesos de la mano. Fuente:(Netter, 2015)

Las articulaciones de la mano y la muñeca tienen el nombre de los huesos que se conectan. Cada dedo tiene dos articulaciones interfalángica, distal y proximal. El pulgar tiene sólo una articulación interfalángica. Entre las falanges proximal y metacarpianos están los nudillos o articulaciones metacarpofalángicas.

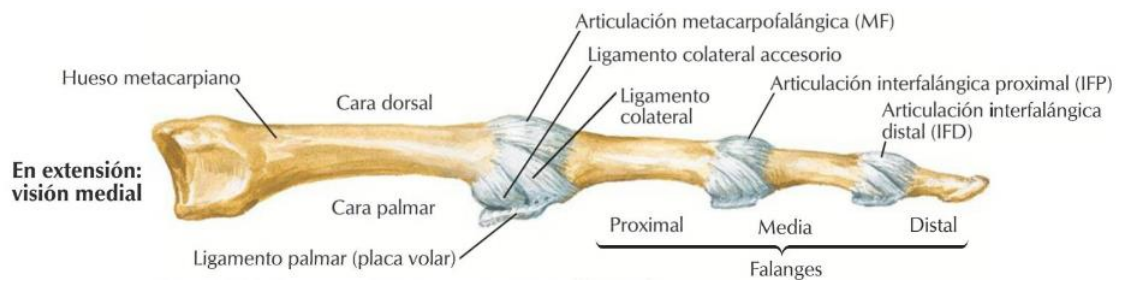


Fig. 3 Articulaciones del dedo de la mano. Fuente: (Netter, 2015)



A su vez, los huesos metacarpianos se articulan con la muñeca para formar cinco carpometacarpianos de las articulaciones. La primera articulación tiene una superficie que le permite moverse en todas direcciones. Esto particularmente da al pulgar su libertad de movimiento: extensión y flexión (paralela a la palma de la mano), abducción y aducción (en ángulo recto con la palma de la mano), así como la rotación.

Además, el primer metacarpiano se ha fijado en un ángulo de 90 grados desde el plano de los otros cuatro metacarpianos, de modo que el pulgar puede entrar en contacto con cada uno de los dedos. Este movimiento opuesto del pulgar permite pellizcar, agarrar y manipular objetos de manera eficiente.

Se debe resaltar que, en anatomía, el carpo es una parte del esqueleto de la extremidad superior que se encuentra en la muñeca. Está compuesta por ocho huesos unidos por ligamentos y que forman el esqueleto de la muñeca. Se disponen en dos filas: proximal y distal como se muestra en la figura 4.

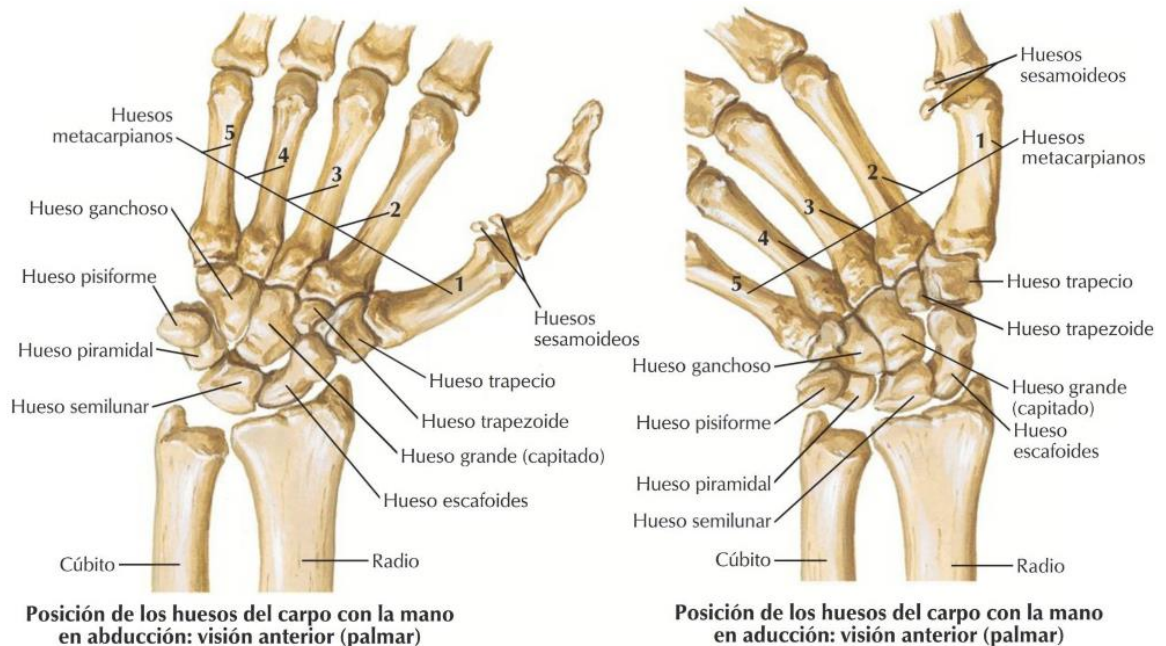


Fig. 4 Disposición huesos de la muñeca. Fuente: (Netter, 2015)



Patrones Funcionales

Esta compleja organización anatómica y funcional de la mano converge en la prensión. La función prensil de la mano depende de la integridad de la cadena cinética de huesos y articulaciones extendida desde la muñeca hasta las falanges distales. La interrupción en los sistemas de arcos transversales y longitudinales resulta en inestabilidad, deformidad y pérdida de función. Los patrones de función prensil son movimientos en los que se agarra un objeto y éste se mantiene en parte o de forma completa dentro de la superficie de la mano, que a su vez se pueden sub dividir en dos tipos.

Los agarres de fuerza que son aquellos en los cuales los dedos están flexionados en las tres articulaciones, el objeto se encuentra entre los dedos y la palma, el pulgar se aduce y queda posicionado sobre la cara palmar del objeto, hay una ligera desviación cubital y se realiza una ligera dorsiflexión para aumentar la tensión de los tendones flexores

Los agarres de precisión son aquellos utilizados para la manipulación de pequeños objetos entre el pulgar y las caras de flexión de los dedos, la muñeca se posiciona en dorsiflexión, los dedos permanecen semiflexionados y el pulgar se aduce y se opone. Los agarres de precisión se clasifican de acuerdo a las partes de las falanges utilizadas para soportar el objeto que se está manipulando, así: pinza terminal, pinza palmar, pinza lateral o de llave, pinza de pulpejo o cubital (Amparo, López, Segundo, Maestría, & Departamento, 2012).

Articulaciones de la muñeca

La muñeca en sí misma es un conjunto de articulaciones que une los huesos cúbito y radio al carpo, es decir, el antebrazo y la mano. Debido a su complejidad permite realizar movimientos en un eje transversal y en un eje anteroposterior, efectuando movimientos de flexión y extensión en el primero de los ejes, y movimientos de inclinación radial o cubital en el segundo de los ejes. Esta comprende desde el punto de vista topográfico la articulación radiocubital inferior y la articulación radiocarpiana.

Articulación radiocubital inferior

Es una trocoide, es decir, una articulación sinovial que permite movimientos rotatorios en todos los planos está formada por un cilindro óseo que encaja en un

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



anillo o cilindro hueco. En este caso reúne el extremo inferior de ambos huesos del antebrazo.

Sus superficies articulares esta representadas en el radio por la cavidad sigmoidea, situada en la cara interna del hueso, plana cóncava en sentido sagital y revestida por el cartílago hialino. En el cúbito por la cabeza cubital, redondeada, convexa y en estado fresco revestida también por el cartílago hialino.

En lo referente a los medios de unión, se centra en los siguientes. Un ligamento interóseo, que es a la vez un ligamento y superficie articular, este es el ligamento triangular, dispuesto horizontalmente en la cara inferior de la cabeza cubital. De forma triangular con vértice interno, se inserta por dentro, por su vértice en la base de la apófisis estiloides del cúbito y por dentro por su base, en el borde inferior de la cavidad sigmoidea. Adherente por sus bordes a la cápsula articular revestida por el cartílago en sus dos caras, más grueso por dentro que por fuera, presenta generalmente en su parte media una dehiscencia que comunica la sinovial de la articulación radiocubital inferior con la de la radiocarpiana.

Una capsula, muy delgada y laxa, insertada en el límite de las superficies óseas y de los bordes del ligamento triangular; se continúa por abajo con la cápsula de la articulación radiocarpiana (Bouchet Alain, 1987).

Un ligamento anterior, simple engrosamiento de la cápsula, formado por fibras oblicuas por abajo y adentro que desbordan ligeramente sobre el borde marginal anterior del radio. Un ligamento radiocubital posterior, también mal individualizado respecto de la cápsula y de disposición simétrica a la del ligamento anterior.

En las articulaciones también está presente el líquido sinovial, el cual reduce la fricción entre los cartílagos y otros tejidos en las articulaciones para lubricarlas y amortiguarlas durante el movimiento. En la articulación radiocubital inferior tapiza la cara profunda de la cápsula y se inserta en el límite de las superficies cartilaginosas. A menudo envía un fondo de saco superior al espacio interóseo; el receso sacciforme. Su cavidad comunica generalmente a través del ligamento triangular con la de la sinovial radiocarpiana.

Articulación radiocarpiana

Esta es una articulación del tipo de las condíleas, consiste de una superficie convexa que encaja en una superficie cóncava, produciendo movimientos en tres planos sobre tres ejes. Une el antebrazo a la primera fila del carpo.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Sus superficies articulares están representadas así. Del lado antebraquial por la glena antebraquial, formada por la cara inferior de la epífisis radial inferior y por la cara inferior del ligamento triangular. Así pues, la extremidad inferior del cúbito no interviene directamente en la constitución de la articulación radiocarpiana.

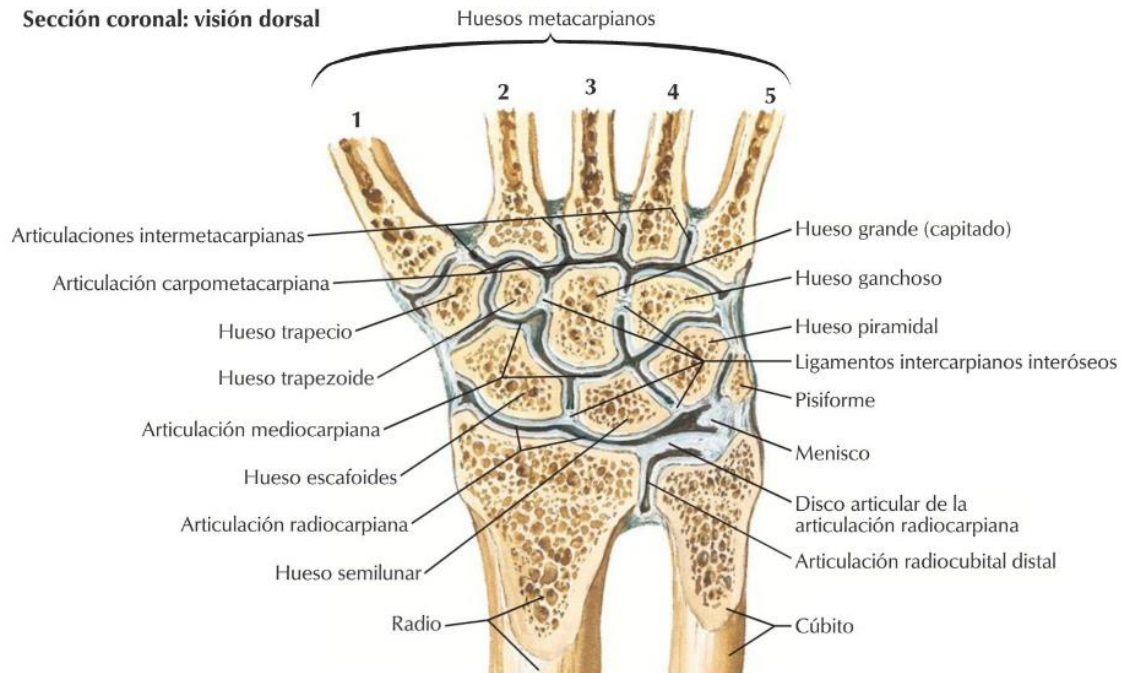


Fig. 5 Articulaciones de la muñeca. Fuente: (Netter, 2015)

La glena antebraquial, por lo tanto, comprende de fuera a dentro la superficie triangular escafoidea de la extremidad inferior del radio, la superficie cuadrilátera, lunar, de este mismo extremo y la cara inferior plana del ligamento triangular. Este conjunto, revestido por lo común de cartílago hialino más grueso en la periferia que en la parte central, forma una cavidad que mira abajo, adelante y adentro. Su concavidad es netamente más marcada en sentido anteroposterior que en sentido transversal.

Del lado carpiano la superficie articular constituye el cóndilo carpiano. Este se halla formado, de afuera adentro por la cara superior del escafoides, la cara superior del semilunar y parte de la cara superior del piramidal. Es pisiforme, situado en un plano más anterior, no participa en la constitución del cóndilo carpiano. Escafoides semilunar y piramidal están solidarizados entre sí mediante ligamentos interóseos. El conjunto de los tres huesos y de los ligamentos interóseos que los unen está revestido por una capa continua de cartílago hialino. El cóndilo carpiano así

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



constituido es regularmente convexo, apenas más amplio que la glena a la que desborda por detrás y a los lados, y mira arriba, atrás y afuera, orientación exactamente inversa a la de la glena.

La correspondencia de los elementos constitutivos de la glena y del cóndilo se establecen de la siguiente manera: el escafoides se articula con la faceta triangular del extremo inferior del radio; el semilunar se articula en sus dos tercios extremos con el radio, en su tercio interno con el ligamento triangular, el piramidal, que no corresponde más que en una parte reducida en la constitución del cóndilo, se articula únicamente con la parte interna del ligamento triangular.

De igual manera a la articulación radiocubital inferior, ésta tiene sus propios medios de unión comprendidos como se describe a continuación. La cápsula común a las articulaciones radiocarpiana y radiocubital inferior, se inserta por arriba en los bordes marginales del extremo inferior del radio y en los bordes del ligamento triangular. Por abajo se fija en el límite de las superficies cartilaginosas del cóndilo carpiano. Relativamente gruesa y sólida por delante es, en cambio, muy delgada a nivel lateral y atrás, donde a menudo presenta dehiscencias por las que la sinovial puede hacer hernia; en ellas se origina la mayoría de los quistes sinoviales de la muñeca.

Respectivamente, los ligamentos comprenden a los laterales, como los más sólidos e importantes, los ligamentos anteriores y posteriores y el ligamento radioescafolunar. Cuya disposición se muestra en la figura 6 y se explica a continuación.

Ligamento lateral externo: corto y grueso con forma de triángulo de vértice superior, se inserta por arriba en el vértice de la estiloides radial y desciende verticalmente a fijarse en la cara externa y en el tubérculo del escafoides.

Ligamento lateral interno: más resistente y alargado que el precedente, el ligamento lateral interno se inserta por arriba en el vértice y la cara interna de la estiloides cubital. Se divide en dos fascículos; uno anterior que va a fijarse en el pisiforme y otro posterior que va a terminar en el tubérculo posterior del piramidal.

Ligamento anterior: está formado por dos fascículos, nacidos uno del radio y otro del cúbito y que van a entrecruzarse en la cara anterior del carpo. El conjunto forma una especie de media luna fibrosa de concavidad superior, de allí el nombre de ligamento arqueado de Weitbrecht que ha recibido a veces. Como se ha dicho comprende dos fascículos, un fascículo radiocarpiano y un fascículo cubitocarpiano (Bouchet Alain, 1987).

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Fig. 6 Disposición ligamentos de la muñeca. Fuente: (Netter, 2015)

Ligamento radiocarpiano posterior: mucho menos importante, comprende dos fascículos, uno principal y otro accesorio. El principal es ancho y extendido, nace en la parte media del borde marginal posterior del radio y desciende oblicuo abajo y adentro para terminar en la cara dorsal piramidal. El fascículo accesorio nace aisladamente en el borde posterior de la estiloides radial y desciende en dirección vertical a finarse en la cara posterior del escafoides.

Ligamento radio escafolunar profundo: es un ligamento poco resistente con forma de una estrella de tres puntas que se inserta, por arriba, en la escotadura escafolunar del borde marginal anterior del radio y por abajo, por un lado, en la cresta obtusa que separa las carillas radial y lunar del escafoides y por el otro en la parte superior de la fosa perforada de la cara de externa del semilunar (Amparo et al., 2012).

Finalizando, el sinovial presenten en esta articulación, tapiza la cara profunda de la cápsula se inserta en el contorno de la glena antebraquial y del cóndilo carpiano. En el 40% de los casos tiene comunicación con la sinovial de la radiocubital inferior y en el 12% con la sinovial de la mediocarpiana. Envía, además, cierto número de fondos de saco y de prolongaciones; en la cara palmar se halla la prolongación

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



preestiloidea. En la cara dorsal la sinovial envía a través de las dehiscencias de la cápsula cierto número de botones sinoviales, punto de partida habitual de los quistes sinoviales de la muñeca.

Tipos de movimiento articular

Los movimientos naturales de los dedos de la mano, se pueden encerrar en cinco tipos que son: Movimiento de flexión y extensión, movimiento de abducción y aducción y los movimientos giratorios desde la base de la articulación de la falange proximal de cada dedo.

El movimiento de flexión y extensión de los dedos, involucra la ejercitación de las articulaciones interfalángicas proximal, medial y distal. El movimiento de flexión se desarrolla desde la punta del dedo hacia la palma de la mano, caso contrario sucede con el movimiento de extensión el cual inicia desde la palma de la mano hacia afuera de la misma.

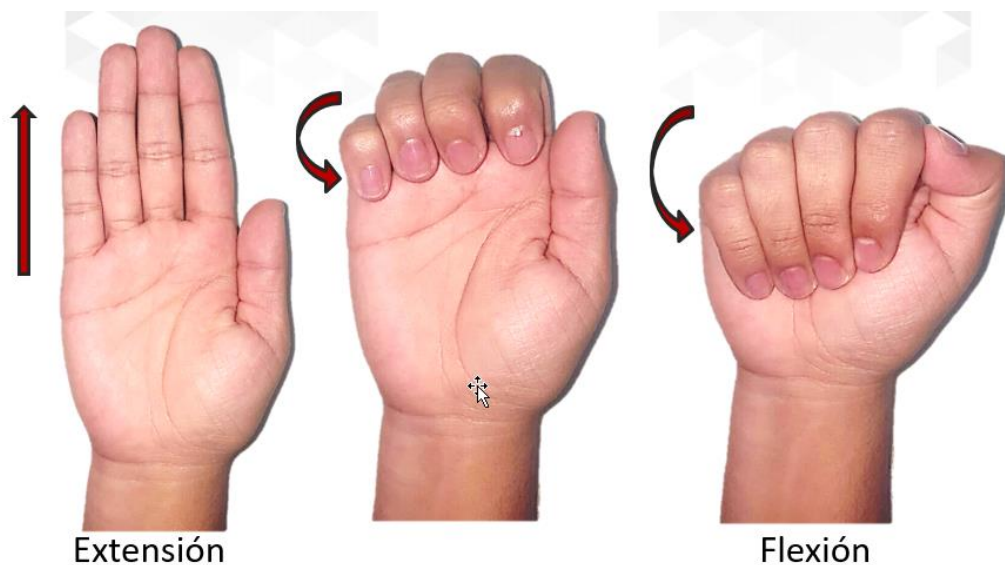


Fig. 7 Movimientos de extensión y flexión de los dedos de la mano. Fuente: los autores.

De igual manera se presentan los movimientos de aducción y abducción que se realiza situado en el plano de la palma de la mano y se centra en el dedo corazón. La abducción es el movimiento de los dedos alejándose del dedo corazón, mientras



que la aducción es el movimiento de los demás dedos aproximándose a este. E involucra la ejercitación de las articulaciones metacarpofalángicas y la interfalángica proximal.

La amplitud del movimiento se puede describirse como la distancia en que pueden extenderse los dedos, medida desde la punta del dedo índice hasta la punta del meñique. También puede medirse el movimiento entre dedos concretos, de la punta de uno a la del otro.

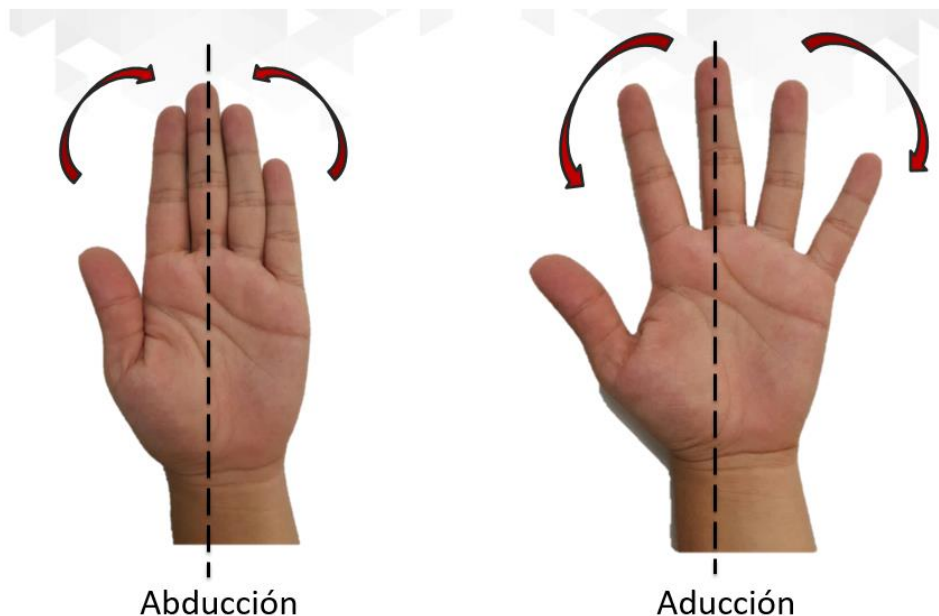


Fig. 8 Movimientos de abducción y aducción de los dedos de la mano. Fuente: los autores.

Al igual que los dedos de la mano, la muñeca presenta el movimiento de flexión y extensión el cual no se limita por la rotación del antebrazo, la posición de partida cero con la cara de la pala de la mano hacia abajo corresponde al borde cubital del tercer metacarpiano alineado con el eje del antebrazo distal. Esta posición facilita la colocación del eje para el movimiento de la muñeca.

La amplitud del movimiento se puede dar de cero a noventa grados, para los dos movimientos y en el caso de la extensión se parte de la posición neutra elevando la mano y atrayéndola hacia el antebrazo. El movimiento de flexión de igual manera parte desde la posición neutra, pero en este caso se desplaza hacia debajo de esta posición atrayéndolo hacia el antebrazo.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Fig. 9 Movimientos de flexión y extensión de la muñeca. Fuente: los autores.

Ejercitación pasiva de la mano.

Los ejercicios de fisioterapia aplicados convencionalmente a las articulaciones de la mano, se dividen en dos tipos; ejercicios pasivos y ejercicios activos. Los cuales se diferencian principalmente por el origen de la fuerza que da inicio al movimiento de ejercitación. De igual manera según el tipo de ejercicio que se realice es necesario o no el uso de herramientas de apoyo.

Los ejercicios pasivos sirven para que las articulaciones se mantengan tan saludables como sea posible y permitan que el paciente pueda realizar los movimientos con naturalidad y sin ningún tipo de ayuda. Por esto son ejercicios recomendados para terapias preventivas, son utilizados cuando las personas afectadas son incapaces de realizar los movimientos por si mismos. Se debe aclarar que los movimientos pasivos en el arco de movimiento ayudan a mantener las articulaciones flexibles, pero no fortifican los músculos involucrados en el movimiento realizado.

La aplicación de este tipo de ejercicios puede ser muy sencilla y no requiere del uso de herramientas de apoyo, basta con la aplicación de una serie de movimientos por parte del terapeuta.

Para la ejercitación pasiva flexo – extensiva de los dedos de la mano, el terapeuta toma la mano del paciente con las dos manos o ubica la mano en una superficie plana que permita la disposición de la mano del paciente de manera neutra. Luego

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



sostiene la mano de la persona hacia él fisioterapeuta, con los dedos extendidos; para luego doblar los dedos formando un puño. Después se estiran los dedos otra vez. De igual manera este ejercicio se puede repetir de manera individual doblando y estirando cada dedo uno a la vez.

Otra forma de realizar ejercicios de extensión pasiva en los dedos de la mano es extendiendo el pulgar y el dedo índice separados, a continuación, poniéndolos juntos nuevamente. Extendiendo el dedo índice y el dedo medio separados, a continuación, poniéndolos juntos nuevamente. Haciendo lo mismo con el resto de los dedos de manera alternada se completa el ejercicio por parte del fisioterapeuta.

En lo referente a los ejercicios para la muñeca, se apoya la muñeca de la persona sobre una superficie plana o en una mano del fisioterapeuta. Sosteniendo sus dedos de la mano a ejercitar desde la punta de los mismos. Luego de esto se dobla la mano de la persona hacia su hombro para que sus dedos estén apuntando hacia el techo. A continuación, se dobla la mano hacia abajo para que sus dedos apunten hacia el suelo. De esta manera se completan los ejercicios de flexión y extensión de la muñeca.

De manera adicional se puede balancear la mano de la persona hacia atrás y adelante lateralmente. Realizando giros suavemente la mano en círculos en una dirección. Posteriormente, gira la mano en círculos para la otra dirección.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



2.2 ESTADO DEL ARTE

La importancia de la inclusión de la robótica en el área de la salud, ha aumentado de manera significativa debido a los logros alcanzados enfocados en múltiples tareas como las cirugías de alta precisión, el uso de tecnología en el campo de laboratorio clínico, prótesis que sustituyen en cuanto a apariencia y funcionamiento algunas partes humanas, y el uso de diferentes sistemas automatizados que ayudan a recuperar la movilidad de extremidades del cuerpo. Sin embargo, los desarrollos tecnológicos adelantados en la recuperación de movilidad se centran en articulaciones mayores, un ejemplo de ello es la implementación de exoesqueletos que permiten a los usuarios recuperar habilidades como caminar o mantener una posición erguida.

En cuanto a la aplicación de estas tecnologías, en movimientos articulares como los de los dedos de la mano, un gran referente es el robot Amadeo. Un sistema de rehabilitación moderno, mecatrónico que se utiliza en el campo de la neuro rehabilitación robótica. El brazo y la mano del paciente se sujetan a un dispositivo compuesto por pequeñas palancas a las que se acoplan los dedos a través de unos imanes. A partir de ese momento, el robot ofrece diferentes resistencias de flexión y extensión de cada dedo a la vez que mide los rangos de movimiento, fuerzas recibidas y potencia de movilidad (“Rehabilitación de la mano con Amadeo® | Red Menni,” n.d.).

De igual manera existe el robot Pablo, el cual mide la fuerza en todas las formas de agarre, como el agarre cilíndrico y el agarre de pinza. El dispositivo también mide el rango de movimiento del brazo, así como la extensión y flexión de la muñeca o cualquier articulación. Por otro lado, existen sistemas dirigidos específicamente a usuarios que han avanzado en la recuperación de la funcionalidad de la mano, un ejemplo de ello es el Handtutor, un guante sensorizado que permite trabajar por separado con cada uno de los dedos por medio de una interfaz diseñada con juegos interactivos por medio de los cuales se lee el avance en la recuperación de los movimientos y sugiere el tipo de terapia que se debe seguir (“PABLO,” n.d.) (“HandTutor - Telerehabilitación,” n.d.).

El proyecto que se propone desarrollar durante este trabajo, se diferencia de los sistemas ya existentes en cuanto al tipo de operatividad que se propone ya que se busca permitir un ciclo de movimientos completamente autónomos, es decir no se necesita de la utilización de fuerza por parte del usuario, cabe resaltar que los movimientos serán pasivos sin ningún tipo de realimentación.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



El proceso de recopilación de información a través de la búsqueda y posterior lectura de artículos científicos, relacionados con la aplicación de la tecnología al área de fisioterapia específicamente en cuanto a rehabilitación motriz, se presenta a continuación mediante la descripción de algunas de las investigaciones más sobresalientes o similares a la investigación que se presenta en el presente libro.

En el primer estudio consultado, denominado “Diseño, desarrollo y validación de dispositivo robótico para la supresión del temblor patológico”, escrito por E. Rocon, A. F. Ruíz, J. M. Belda-Lois, J. C. Moreno, J. L. Pons, R. Raya y R. Ceres en el año 2008, publicado en el Revista Iberoamericana de Automática e Informática Industrial RIAI; se describe la implementación de un dispositivo robótico en forma de exoesqueleto para el miembro superior humano, destinado a la monitorización y supresión de temblor patológico, el cual fue validado clínicamente realizando pruebas con pacientes de dos hospitales.

El contenido del artículo mencionado con anterioridad presenta el diseño mecánico, la aplicación de fuerzas en el brazo, la medición de cinemática y cinética, los sistemas de actuación, la arquitectura y estrategias del control, entre otros aspectos significativos del prototipo WOTAS (Wearable Orthosis for Tremor Assessment and Suppression), que traducido al español significa “Ortesis usable para la evaluación y supresión del temblor”, el cual es un exoesqueleto robótico capaz de aplicar fuerzas internas dinámicas sobre el miembro superior. Un aspecto relevante es el protocolo de medidas, el cual se llevó a cabo mediante experimentos donde los objetivos principales eran los siguientes:

- Evaluar la plataforma hardware que se ha desarrollado.
- Validar las estrategias de control desarrolladas para supresión mecánica de temblor patológico.
- Determinar las diferencias entre aproximaciones activa y pasiva (control).
- Determinar la mejor combinación de parámetros para cada aproximación.
- Evaluar la funcionalidad de las diferentes alternativas de actuadores.
- Estimar el posible impacto y aceptación de un modelo futuro de ortesis para suprimir temblor.

Las pruebas fueron realizadas en dos países distintos. En la primera etapa de evaluación se realizaron pruebas pre-clínicas en el Departamento de Neurología del Hospital Erasme en Bruselas, Bélgica. La segunda etapa de pruebas clínicas, fue realizada en el departamento de Neurología del Hospital General de Valencia,

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



en España. El protocolo de los experimentos fue aprobado por los comités éticos de cada hospital donde se realizaron las pruebas.

Finalmente, en la etapa de pruebas de la implementación de este prototipo los pacientes llevaron adaptado el dispositivo mientras éste operaba en tres posibles modos de operación: monitorización, supresión pasiva y supresión activa, obteniendo reducciones del orden del 80% en la potencia del temblor en los pacientes que padecían temblor severo. Gracias a los excelentes resultados de esa investigación, en el instituto de Automática Industrial donde se desarrolló la misma, adelantarán trabajo en el desarrollo de otro dispositivo similar al WOTAS, que además de cancelar el temblor, también genere y aplique fuerzas de diversas índoles sobre el miembro superior, para evaluar otros tipos de patologías y estudiar a nivel neuromotriz a los pacientes (Rocon et al., 2008).

La segunda consulta se realizó en la versión del 2014 Revista Iberoamericana de Automática e Informática industrial RIAII, donde llamó la atención el artículo escrito por Ricardo López, Hipólito Aguilar, Sergio Salazar, Rogelio Lozano y Jorge Torres, acerca del “Modelado y control de un Exoesqueleto para la rehabilitación de extremidad inferior de dos grados de libertad”. En el artículo presentaron un exoesqueleto de dos grados de libertad para realizar ejercicios de rehabilitación para tobillo y rodilla. El diseño y fabricación del exoesqueleto se basó en la instrumentación de una ortesis del miembro inferior derecho. El Exoesqueleto utiliza sensores que estiman la fuerza producida por el humano y se encuentran incorporados en los actuadores de tipo SEA (Series Elastic Actuator), que traducido al español significa “Actuador elástico en serie”, los cuales se utilizan para amplificar la fuerza humana.

Los autores inician dando al lector una contextualización del tema de exoesqueletos a manera de introducción, donde se aclara la importancia y ventajas que estos presentan a la hora de tratar a pacientes que presentan algún tipo de enfermedad como la hemiplejía, que antiguamente se trataban de manera exclusiva a través de la inmovilización por medio de férulas, las cuales pueden producir algunos problemas dentro de los que se encuentra la atrofia en las fibras musculares. Entonces, buscando contrarrestar estos efectos negativos, consideran necesario buscar no solo la movilización pasiva de la musculatura afectada y activa cuando sea posible, sino también la elongación controlada de esta musculatura, ya que si no se produce movimiento en los músculos y tendones se pierde la capacidad de resistir y generar fuerza.

Posteriormente dedican una sección al modelado dinámico del actuador SEA, del exoesqueleto y de la pierna humana, seguido de una sección exclusiva para la explicación del control, donde se expone por qué el controlador PD no garantiza el

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



cumplimiento del objetivo de control de posición en sistemas cuyos modelos dinámicos tienen en cuenta los pares gravitacionales, por lo cual deciden aplicar una compensación de gravedad al controlador PD. Además, en esta sección del artículo, presentan el análisis de estabilidad en lazo cerrado del controlador.

Los resultados numéricos de la simulación de los movimientos de flexión y extensión de las articulaciones son expuestos de manera general en la cuarta sección del artículo, donde utilizan rangos de movimientos de ángulos alrededor de los 35° y 80° en el tobillo y la rodilla respectivamente. Estas trayectorias las generan a través de la aplicación de funciones cosenoidales, obteniendo las salidas de control deseadas sin sobretiros y con tiempos cortos de respuesta.

La integración de los resultados de las simulaciones, del análisis dinámico mencionado y de todo el trabajo ingenieril aplicado en esa investigación, dieron como resultado el prototipo experimental del exoesqueleto, siendo este el diseño de un exoesqueleto capaz de soportar el peso de una persona de hasta 75 kg, el cual presentan mediante diagramas de bloques con los componentes del prototipo, y los modelos mecánicos de los actuadores para la rodilla. Con este prototipo se realizaron pruebas experimentales que consistieron en ejercicios similares a los simulados. En conclusión, el prototipo experimental construido por los autores se utiliza de manera activa y también puede ser utilizado de manera pasiva, es decir como un sistema estacionario. Los sistemas estacionarios son los mecanismos robóticos diseñados para ejercitar el tobillo y los movimientos de rodilla sin caminar, donde el usuario no ejerce ningún tipo de fuerza (López, Aguilar, Salazar, Lozano, & Torres, 2014).

En la siguiente versión de la misma revista, del año 2015, se encontró el tercer artículo de interés para la realización del presente estado del arte, denominado "Evaluación en un paciente con ictus en fase crónica de un sistema autoadaptativo de reur rehabilitación robótica", publicado por Ricardo Morales, Francisco Badesa, Nicolas García, Joa Aranda y Alicia Casals, donde presentan un nuevo sistema de rehabilitación capaz de adaptarse al estado psicofisiológico del paciente durante tareas de rehabilitación robótica, siendo una continuación de otra investigación presentada en una versión anterior de esta revista, con la diferencia de que se utilizaba en pacientes sanos, y en el artículo consultado se hace uso en pacientes que hayan sufrido de algún tipo de accidente cerebrovascular.

Debido a este enfoque que le otorgaron los autores a la investigación, se hace necesario una contextualización en cuando al daño cerebral sobrevenido (DCS), que corresponde a un concepto clínico heterogéneo asociado a una afectación del sistema nervioso central (SNC) de etiología variable, cuyo denominador común

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



consiste en una lesión cerebral que irrumpe de forma brusca y decisiva en el desarrollo vital de la persona, donde la más común es el accidente cerebrovascular (ACV), que se conoce como ictus. La cual presentan a manera de introducción, donde incluyen también datos estadísticos tomados de la organización mundial de la salud. Evidentemente se enfocan en el ACV como causa del deterioro crónico de la función del brazo para efectuar muchas actividades cotidianas, ya que es precisamente lo que presente evaluarse en el desenlace del artículo.

Después de las correspondientes aclaraciones se detalla el sistema autoadaptativo de rehabilitación robótica, donde se hace énfasis en la incorporación de la realimentación en el lazo del control del sistema, a través de la extracción de señales fisiológicas del usuario que permiten identificar su estado psicofisiológico y usar esta información para modificar automáticamente el nivel de dificultad de la tarea a realizar. La descripción del sistema se da mediante un esquema del concepto de sistema autoadaptativo, donde se observan cuatro bloques diferenciados:

- Bloque del sistema de neuro-rehabilitación asistida por robot.
- Control a alto nivel.
- Estimación del estado del usuario.
- Usuario.

Luego se representa el proceso general de la medida, interpretación y uso de la respuesta fisiológica para su uso en sistemas informatizados, el cual también se divide en cuatro fases:

- Tareas de adquisición, procesado y extracción de características de las señales fisiológicas del usuario.
- Normalización de señales y reducción de dimensión del espacio de entrada.
- Clasificación y estimación.
- Implementación del sistema adaptativo usando la información extraída de las señales fisiológicas.

En el detalle del sistema autoadaptativo también se describen las técnicas de aprendizaje automático, que corresponden al bloque de estimación del estado del usuario, que los autores decidieron realizar a través de algoritmos de aprendizaje supervisado, los cuales deducen una función a partir de datos de entrenamiento, que corresponden a vectores constituidos por dos datos, uno corresponde al par de entrada y el otro al resultado esperado. El objetivo del aprendizaje supervisado es el de crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente. Por esta razón, los autores deciden utilizar 9 algoritmos supervisados de clasificación para estimar el estado fisiológico:

- Perceptron Learning Algorithm (PLA)
- Logistic Regresión (LR)
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- Support Vector Machine lineal (SVML)
- Support Vector Machine con RBF kernel (SVMK)
- Naïve Bayes (NB)
- K-Nearest Neighbor (KNN)
- K-Center con Radial Basis Functions (RBF)

En la sección correspondiente a resultados experimentales, donde se da a conocer la descripción de la experimentación, la adquisición de las señales fisiológicas, el entrenamiento y selección de los algoritmos de aprendizaje automáticos, la cual se hace mediante el análisis de componentes principales (PCA), tabulando los aciertos para seleccionar el clasificador más idóneo; y finalmente la adaptación automática de la dificultad de la terapia. En conclusión, del desarrollo de estos experimentos se estableció seleccionando un modelo SVM con una función RBF como kernel, y el uso de las 3 primeras componentes principales del algoritmo PCA sobre los datos de entrada del clasificador, con el cual obtuvieron un 67,57% de acierto en la validación (Morales, Badesa, Garcia-Aracil, Aranda, & Casals, 2015).

La cuarta investigación consultada corresponde a un artículo aceptado para publicación en una versión del 2016 de la revista "IEEE Transactions on industrial electronics" (IEEE transacciones en la electrónica industrial), escrito por Yong Yang, Lei Ma, y Deqing Huang, a cerca del "Development and Repetitive Learning Control of Lower Limb Exoskeleton Driven by Electro-Hydraulic Actuators" (Desarrollo y Aprendizaje Repetitivo Control del Exoesqueleto de las Extremidades Inferiores Impulsado por Actuadores Electrohidráulicos), donde presentan un novedoso exoesqueleto de las extremidades inferiores denominado CASWELL de 5 grados de libertad por pierna, donde dos usan actuadores electrohidráulicos.

A manera de introducción presentan algunos de los exoesqueletos más reconocidos a nivel de investigación, dentro de los que se encuentran los desarrollados por DARPA, que se enfocan en aumentar el rendimiento humano en cuanto a la capacidad de carga al transportar equipo pesado, el exoesqueleto

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Berkeley implementado en la universidad de California, la investigación de la Universidad de Tsukuba donde crearon una especie de pierna asistida híbrida (HAL) mejorar la capacidad normal de caminar y cargar, así como ayudar a las personas a subir escaleras y levantar objetos pesados, entre otros. También se destacó la importancia de los actuadores hidráulicos al proporcionar potencias superiores a otros tipos de actuadores. Dejando claro que el objetivo de la investigación desarrollada por los autores es desarrollar un exoesqueleto para el aumento de potencia humana que pueda rastrear el movimiento humano de forma adecuada, se establecen algunas estrategias de control donde la dinámica del cuerpo rígido y los actuadores electrohidráulicos se consideran simultáneamente, donde el controlador propuesto se implementa en el sistema electrónico integrado con un microcontrolador de 32 bits y se prueba en el CASWELL desarrollado.

También se presenta el diseño mecatrónico del prototipo CASWELL, iniciando con el diseño de la estructura mecánica, donde pueden identificarse los 5 grados de libertad de cada pierna:

- Flexión y extensión de cadera (1 grado de libertad).
- Flexión y extensión de rodilla (1 grado de libertad).
- Flexión plantar y dorsiflexión de tobillo (3 grados de libertad).

Donde las articulaciones de la cadera y la rodilla son accionadas por actuadores electrohidráulicos mientras que las articulaciones del tobillo son pasivas.

También hace parte del diseño mecatrónico de CASWELL el diseño de sistema electrónico incorporado, que incluye la recopilación y el procesamiento de datos, la predicción de la marcha, la implementación del controlador y la activación de los actuadores electrohidráulicos. Además, se hace mención a algunos dispositivos adicionales que son esenciales para el control de CASWELL, dentro de los que se encuentran sensores y válvulas

Se debe resaltar el modelo dinámico completo del prototipo, donde abordan de manera secuencial la dinámica del cuerpo rígido y los actuadores electrohidráulicos, a través de la respectiva matemática que aclara al lector el análisis dinámico. Posteriormente dedican una sección del artículo al análisis del diseño y convergencia de RLC "Repetitive learning control" (Control por aprendizaje repetitivo), primero con el diseño del controlador con la técnica de diseño de retroceso, y finalmente con sus principales resultados, donde destacan algunas convergencias de componentes del modelado.

Los resultados de las pruebas del esquema RLC en el prototipo CASWELL los presentan en la quinta sección del artículo, mediante tres etapas:

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



- Configuración del experimento.
- Parámetros de Implementación y Experimentación del Controlador.
- Resultados y análisis del experimento.

A manera de conclusión, se hace referencia al desarrollo del novedoso sistema de exoesqueleto de miembro inferior, CASWELL, que es accionado por actuadores electrohidráulicos, donde gracias a su estructura mecánica, el sistema electrónico integrado para el exoesqueleto y el esquema RLC implementado se amplía la aplicabilidad potencial de RLC en la industria y la ingeniería (Yang, Ma, & Huang, 2016).

Otro sistema analizado en el presente estado del arte se denomina "Shoulder Flexion Rehabilitation in Patients with Monoparesia Using an Exergame" (Rehabilitación de flexión de hombro en pacientes con monoparesia usando un juego de ejercicios), escrito por Javier Bedoya, Juan Daniv Hoyos, John Edison Muñoz y José Fernando López, y aprobado por la IEEE, donde proponen una combinación de los dos métodos para encontrar mejoras en el ángulo de flexión del hombro afectado por la lesión de la neurona motora superior en pacientes de la "Clínica de Dolor del Eje Cafetero". Mediante sesiones de rehabilitación con una duración de 30 minutos de terapia asistida y otros 30 minutos con terapias de realidad virtual, donde los movimientos de flexión del hombro afectado a través del sensor de Kinect se registraron mientras el paciente interactuaba con el videojuego.

Después del respectivo resumen los autores inician el artículo con una introducción donde atribuyen el síndrome piramidal a la lesión de la neurona motora superior o a la lesión de la neurona piramidal, de la cual desciende la carretera principal que controla los movimientos voluntarios, y presentan algunos tratamientos comúnmente usados por fisioterapeutas, dentro de los que se encuentran:

- Técnica de Bobath o neurodesarrollo.
- Método de Facilitación Neuromuscular de Kabat o Proprioceptivo.
- Reaprendizaje motor.

Según los autores, aunque la rehabilitación estándar (es decir, la fisioterapia y la terapia ocupacional) ayuda a mejorar la función motora después del accidente cerebrovascular, solo se han demostrado hasta el momento beneficios modestos, ya que se presentan algunas limitaciones de los enfoques de rehabilitación convencionales como el tiempo que requiere, la dependencia del cumplimiento del paciente, la disponibilidad limitada de acuerdo a la geografía, los efectos moderados y retrasos de algunos pacientes, el requerimiento de transporte a instalaciones especiales, los costos, entre otros.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



La metodología aplicada en el desarrollo de la investigación se presenta después de la introducción al tema, donde iniciar aclarando el objetivo principal del proyecto, definiéndolo como la evaluación de la flexión hemiparética del hombro en pacientes con síndrome piramidal utilizando un sensor Kinect. Para la inclusión de los pacientes en este estudio tuvieron en cuenta que se cumplieran los siguientes requisitos:

- Tener monoparesía en la extremidad superior causada por una lesión de la neurona motora superior debido a una apoplejía, CP (parálisis cerebral) o TBI (lesión cerebral traumática).
- Tener la capacidad de estar de pie en el momento de interactuar con el Exergame.

Un total de seis pacientes cumplieron estas características de los cuales 3 sufrieron ACV, 1 LCT y 2 IMOC. Las sesiones fueron realizadas en una sala ubicada en el centro de rehabilitación de la "Clínica de Dolor del Eje Cafetero" llamada "salón interactivo para la rehabilitación" (SIR) o Sala Interactiva de Rehabilitación en inglés, que tenía un espacio adecuado para desarrollar prácticas y herramientas necesarias para interactuar, registrar y evaluar el movimiento del hombro afectado en cada paciente; entre ellos se encontraban: una computadora de escritorio, un sensor Kinect, un software para capturar y analizar movimientos (MoCap), un video proyector, un dispositivo de sonido y un Exergame diseñado específicamente para que los pacientes tuvieran que realizar el movimiento de flexión con el hombro afectado.

El objetivo del juego es romper cierto número de mesas a medida que avanza el juego, para ello el paciente debe realizar un movimiento de flexión del hombro lo más ampliamente posible y, a partir de ahí, generar el movimiento de extensión necesario para romper las tablas virtuales. Una vez que el paciente puede romper la primera mesa representada como el primer nivel, se agregarán más tablas a medida que avance el juego, agregando más complejidad al juego.

Los resultados de la aplicación del videojuego a los pacientes con síndrome piramidal se demostraron a través de imágenes donde se puede observar la evolución del movimiento del ángulo del paciente de acuerdo con el gráfico angular en el plano sagital, además de los respectivos análisis de las sesiones con los pacientes, con lo cual pudieron concluir que encontraron un evento emocionante en la terapia de realidad virtual para los pacientes. Los autores evidenciaron una gran importancia entre las relaciones establecidas por el participante y el terapeuta por la correcta ejecución del movimiento cuando las indicaciones eran claras y

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



había una motivación constante y los resultados fueron motivadores en las articulaciones logradas con la combinación de métodos asistidos (Kabat y Bobath) y realidad virtual durante el curso de las sesiones, aumentando las posibilidades de cumplir algunas de las tareas básicas de la vida diaria y mejorando así su calidad de vida (Castaño, Cardona, Escobar, & Herrera, 2014).



Capítulo III. DESARROLLO DE LA INVESTIGACIÓN.

3.1 CRITERIOS DE DISEÑO

Una vez analizados los movimientos propios de las articulaciones de la mano, y de la muñeca. Se determinó el tipo de movimiento a automatizar que cumpla con las especificaciones planteadas al inicio de ésta investigación. Teniendo en cuenta que el objetivo principal es desarrollar un sistema mecatrónico que realice tareas de ayuda en ejercicios de rehabilitación motriz, los criterios de diseño abarcan desde los aspectos mecánicos y electrónicos hasta lograr un prototipo que sea compacto y agradable para el medio en el que se pretende será usado.

Dentro de los movimientos naturales de la mano estudiados, se observa que la extensión y flexión de un dedo de la mano involucra la ejercitación de tres articulaciones de manera simultánea; las correspondientes a las uniones interfalángicas (proximal, media y distal). A demás de ser una serie de movimientos utilizados frecuentemente en diferentes tipos de terapias convencionales. De esta manera se cumple con el objetivo planteado, ejercitando no solo una sino tres articulaciones de la mano.

La realización de los movimientos de extensión y flexión de los dedos de la mano, implican que la mano se disponga en una posición neutra, con el fin de que las acciones se den de manera cómoda, segura y práctica. Ante esta situación, se presenta la oportunidad de aprovechar dicha posición para agregar un tipo de movimiento más a la serie de ejercicios de rehabilitación motriz, siendo este la extensión de la muñeca.

Luego de establecer el tipo de movimiento que se va a automatizar, es necesario definir los componentes necesarios para que el sistema realice los ejercicios indicados con un correcto funcionamiento, esto incluye los actuadores, tarjeta de control, mecanismo y estructura diseñada. Lo que representa un sistema mecatrónico como se muestra en la figura 10. En el cual se muestra la relación directa entre la tarjeta de control con los demás componentes del sistema, y se resalta la interacción del especialista quien programa los movimientos del prototipo que a su vez son efectuados en la mano del usuario o paciente.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

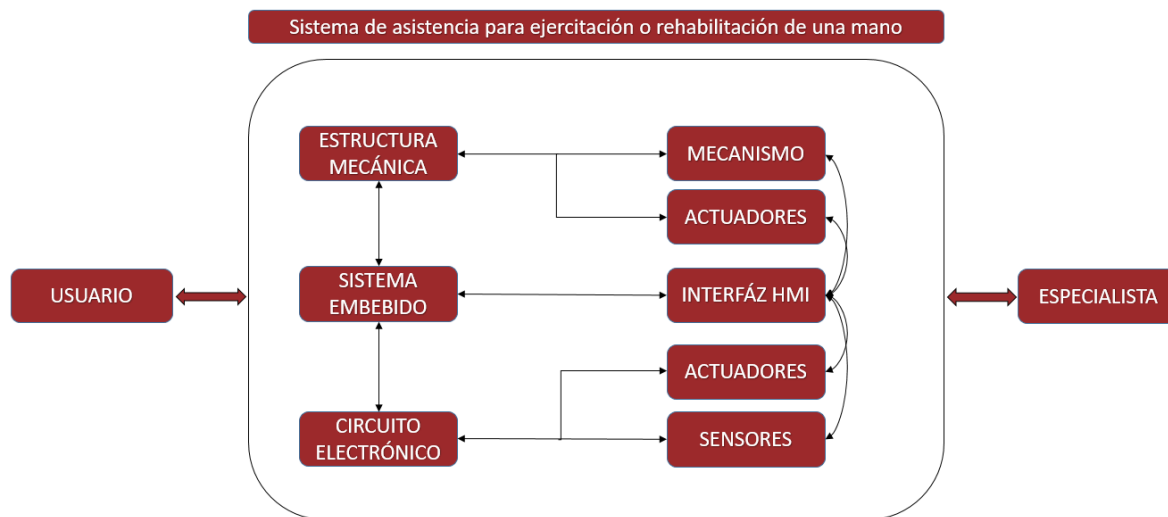


Fig. 10 Esquema de niveles del sistema mecatrónico. Fuente los autores.

3.1.1 INGENIERÍA DEL DETALLE

El sistema desarrollado cumple con la realización de ejercicios de flexo extensión de las articulaciones ya mencionadas mediante un mecanismo que permite ejercitar cuatro dedos de la mano (índice, corazón, anular y meñique), haciendo la aclaración de que la ejercitación se realiza un dedo a la vez, por lo cual para llevar a cabo una serie de movimientos en los cuatro dedos es necesario que el especialista a cargo de la terapia intercale los dedos sobre los cuales se requiera realizar la ejercitación. A demás el sistema permite que de manera alternada se realicen movimientos en el dedo y la muñeca.

La elección del tipo de movimiento a ejecutar se puede realizar mediante un menú mostrado en una pantalla LCD por medio de la cual además se da la opción de elegir la fluidez con la que se efectúa el movimiento y el número de repeticiones deseadas y otra serie de opciones que se mostraran con detalle más adelante. Un esquema general del funcionamiento del sistema desarrollado se muestra en la figura 11.

Teniendo en cuenta que los componentes del sistema diseñado deben cumplir con una función específica, a continuación, se muestra de manera detallada las razones por la cuales son elegidos, al igual que la forma en que son usados con el fin de



cumplir el objetivo de ésta investigación. Lo anterior con base al tipo de movimiento a automatizar y las implicaciones que conlleva.

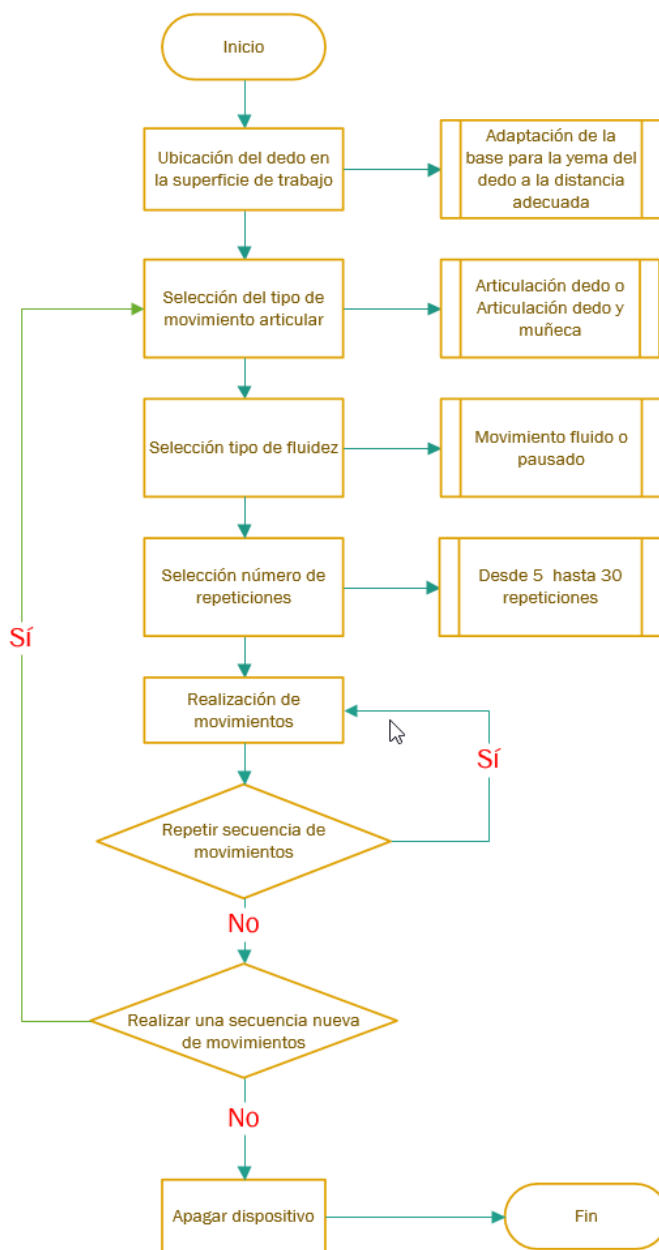


Fig. 11 Diagrama de flujo del funcionamiento general del sistema. Fuente: los autores.



Tarjeta de control

Basándose en la premisa de que el sistema a desarrollar servirá como herramienta de trabajo a fisioterapeutas, este debe ser compacto y fácil de transportar. Esto hace que el tamaño de la tarjeta de control elegida cobre una gran importancia al momento de la selección de la misma. De esta manera se analizaron tres tarjetas de control disponibles en el mercado que cumplan con las especificaciones de tamaño deseada sin que esto afecte sus capacidades técnicas.

Las tarjetas comparadas se muestran en la siguiente tabla.

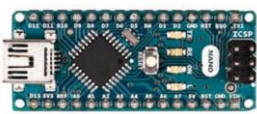


Tarjeta	Arduino Nano	Arduino Uno	Arduino Mega 2560
			
Dimensiones	18 x 45 mm	68.6 x 53.4 mm	53.3 x 101.52 mm
Microcontrolador	ATmega328	ATmega328P	ATmega2560
Alimentación	7-12 V	6-20V	6-20V
Velocidad de reloj	16 MHz	16 MHz	16 MHz
Pines disponibles	22 (6 de los cuales son PWM)	14 (de los cuales 6 proporcionan salida PWM)	54 (de los cuales 15 proporcionan salida de PWM)

Tabla 1 Comparación de tarjetas de control. Fuente: los autores.

Para realizar el control del sistema, se escogió una tarjeta Arduino Mega 2560 mostrada en la figura 12. Debido a su versatilidad para el control PWM (modulación por ancho de pulso); necesario en para el control de los actuadores y la disponibilidad de pines digitales para la conexión de sensores y demás dispositivos que se implementaran como la pantalla LCD que funcionara como interfaz MHI; razón por la cual no se seleccionaron las otras tarjetas analizadas ya que la cantidad de pines digitales requeridos (23 aproximadamente) no están disponibles en ellas o requieren de acondicionamientos que pueden afectar la comunicación con la pantalla LCD. A demás de ser un dispositivo de Software y Hardware libre, permitiendo la construcción de un prototipo de bajo costo. Las principales características de esta tarjeta son:



Microcontrolador	ATmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pines de E / S digitales	54 (de los cuales 15 proporcionan salida de PWM)
Clavijas de entrada analógica	dieciséis
Corriente DC por Pin E / S	20 mA
Corriente DC para 3.3V Pin	50 mA
Memoria flash	256 KB de los cuales 8 KB utilizados por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	101.52 mm
Anchura	53.3 mm
Peso	37 g

Tabla 2 Características Arduino mega 2560. Fuente: los autores.



Fig. 12 Tarje Arduino mega 2560. Fuente: los autores.

Autor:
Ing.(c) John Edison Sánchez Fernández

Director:
PhD. César Augusto Peña Cortés.



Actuadores

Para efectuar el movimiento de flexión y extensión de las articulaciones interfalángicas del dedo de la mano, y de igual manera, para los movimientos de la articulación de la muñeca; se decidió usar servomotores Futaba S3003 como el mostrado en la figura 13 compatibles para realizar el control de los mismo mediante la tarjeta Arduino ya mencionada.

Esta elección se da basándose en aspectos como el control de posicionamiento, par ejercido, la corriente de operación y el beneficio económico. Las características técnicas del motor se muestran en la siguiente tabla.

Modelo	Futaba S3003
Modulación	Analógica
Voltaje de operación	4.8V~6V
Corriente de operación (4.8V)	7.2mA
Corriente de operación (6V)	8mA
Torque (4.8V)	3.2Kg*cm
Torque (6V)	4.1Kg*cm
Velocidad de funcionamiento (4.8V)	0.23sec/60° sin carga
Velocidad de funcionamiento (6V)	0.19sec/60° sin carga
Tipo de motor	3 polos tipo ferrita
Piñonería	Plástica
Ancho de pulso	1520us~1900us
Longitud del cable	30cm
Temperatura de operación	-20°C~60°C

Tabla 3 Características servo motor Futaba S3003. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Fig. 13 Servomotor Futaba S3003. Fuente: los autores.

Sensores de posición

Para controlar el avance de los motores, se implementan sensores de posición conocidos como interruptores de límite o finales de carrera. Los cuales son interruptores de acción mecánica que al detectar la presencia del mecanismo de movimiento cierran el circuito enviando una señal positiva (o negativa según sea el esquema de conexión) al dispositivo de control, permitiendo tomar las medidas de acción que sean programadas. Estos sensores tienen un rango de alimentación de 0 a 24 voltios y una corriente de ruptura de 5 amperios.

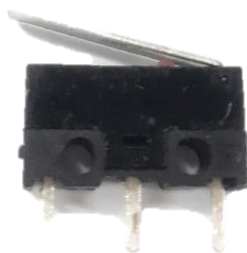


Fig. 14 Sensor final de carrera. Fuente: los autores

Interfaz HMI

Siendo uno de los objetivos específicos de esta investigación, implementar una interfaz de usuario que permita configurar e iniciar el ciclo de movimientos de la ejercitación de las articulaciones elegidas. Se compararon tres tipos de pantallas que permiten implementar botoneras para recibir datos de entrada y así desplazarse por un menú de opciones para el control del proceso.



Pantalla	LCD KeyPad shield	LCD TFT shield	Nextion LCD
			
Resolución	1602 caracteres 2 filas x 16 columnas	240 x 320	320 x 240
Dimensiones	80 x 58 mm	2.4 pulgadas	2.8 pulgadas
Control	Librerías para Arduino	Librerías para Arduino	Interfaz propia (Nextion editor Guide)
Adquisición de datos	5 pulsadores con funciones predeterminadas	Pantalla táctil resistiva	Pantalla táctil resistiva
Alimentación	5V	5 V	5 V
Precio	\$ 15.000	\$ 20.000	\$ 90.000 (aproximadamente)

Tabla 4 Comparación entre pantallas LCD. Fuente: los autores.

Luego de comparar las pantallas mencionadas, se implementa una pantalla táctil de 2.4 pulgadas LCD TFT disponible a manera de shield para Arduino uno, sin embargo, es compatible con la tarjeta Arduino Mega 2560. Por lo cual se realizan ligeras modificaciones para su uso, entre ellos la disposición de pines de comunicación SPI. Esta pantalla es mostrada en la figura 15.

Con el uso de esta pantalla se muestra al usuario los dos tipos de movimiento previamente programados, la fluidez del movimiento y la cantidad de repeticiones que se pueden llegar a realizar, visualizando el menú de opciones de manera secuencial.

Esta pantalla es de tipo resistiva y tiene una comunicación de 8 bits, por 4 líneas de control, con una alimentación de 5 voltios, una corriente de funcionamiento de 300 mili amperios y una resolución de 240 x 320 pixeles la disposición de pines de conexión se muestra en la tabla 5. La programación de esta pantalla se explica en la sección de “diseño de software”.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Pin Arduino	Pin LCD Shield	Uso
3.3V	3.3V	Alimentación
5V	5V	Alimentación
GND	GND	Alimentación
A0	LCD_RD	Control LCD
A1	LCD_WR TOUCH_YP	Control LCD / Dato Touch
A2	LCD_RS TOUCH_XM	Control LCD / Dato Touch
A3	LCD_CS	Control LCD
A4	LCD_RST	Reset LCD
D2	LCD_D2	Dato LCD
D3	LCD_D3	Dato LCD
D4	LCD_D4	Dato LCD
D5	LCD_D5	Dato LCD
D6	LCD_D6 / TOUCH XP	Dato LCD / Dato Touch
D7	LCD_D7 / TOUCH YM	Dato LCD / Dato Touch
D8	LCD_D0	Dato LCD
D9	LCD_D1	Dato LCD
D53	SD_SS	SD Select
D51	SD_DI	Dato SD
D50	SD_DO	Dato SD
D52	SD_SCK	SD Clock

Tabla 5 Características pantalla LCD TFT 2.4 pulgadas. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

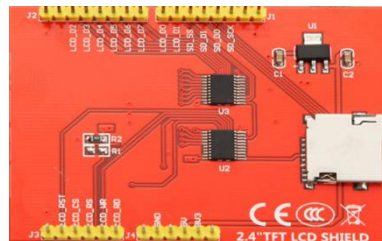


Fig. 15 Pantalla LCD TFT 2.4 pulgadas. Fuente: los autores.

3.1.2 DISEÑO MECÁNICO

En cuanto al diseño mecánico, se tomaron en cuenta diferentes mecanismos que permitieran un desplazamiento horizontal en dos sentidos, para que a su vez se transmitiera dicho desplazamiento al dedo de una mano, logrando así los movimientos de flexión y extensión del dedo, ejercitando las articulaciones interfalángicas como ya se explicó con anterioridad en el marco teórico de esta investigación.

Los ejercicios de flexión y extensión son elegidos debido al número de articulaciones que se pueden ejercitar a la vez con un solo tipo de movimiento, además de ser uno de los tipos de ejercitación más comunes en terapias de rehabilitación motriz.

Una vez determinado el movimiento a realizar, se busca la manera de obtener un diseño compacto que permita desarrollar un sistema portátil, pero que a la vez cumpla con la finalidad de ejercitar las articulaciones del dedo de la mano sin que se vea afectada la integridad del usuario. Luego de estudiar diferentes mecanismos y realizar una serie de bocetos a mano se llegó a tres que cumplen con este requerimiento, los diseños escogidos se muestran a continuación.

Mecanismo tornillo sin fin.

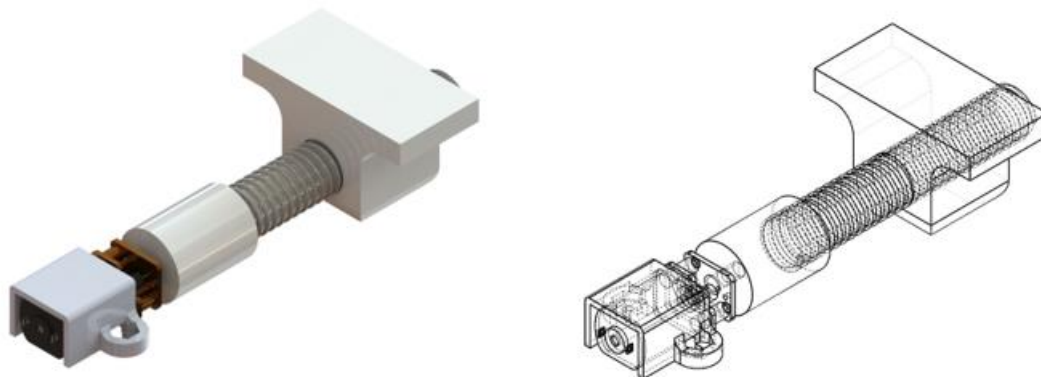


Fig. 16 Mecanismo tronillo sin fin. Fuente: los autores.

El mecanismo mostrado en la figura 16 consiste en un motor de giro continuo con transmisión de movimiento directa a un tornillo sin fin por medio de un acople metálico, este tornillo tiene unida una base que se desplaza de manera ascendente o descendente según el sentido de giro del motor. De esta manera se obtiene el movimiento de tipo lineal deseado en el efector final del mecanismo. En la base móvil sujeta al tornillo sin fin se puede acoplar la yema del dedo a ejercitar y se transmitirían los movimientos a las articulaciones del mismo

Mecanismo de transmisión por correa dentada

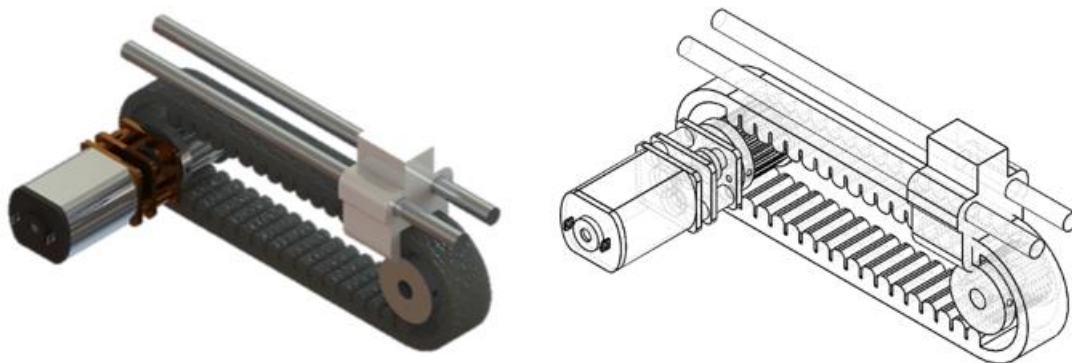


Fig. 17 Mecanismo de transmisión por correa dentada. Fuente: los autores.



Este mecanismo utiliza el mismo principio al mecanismo de tornillo sin fin, con la diferencia de que el motor de giro continuo se une a un piñón que transmite el movimiento a una correa plástica dentada con dos puntos de tensión, que a su vez tiene una base sujeta en uno de sus lados planos como se muestra en la figura 17.

Al ser una correa plástica que solo tiene un punto de unión con la base que recibe el movimiento final, esta puede permitir un desacople o juego en el movimiento lineal. Para corregir este posible error la base que se sujeta a la correa cuenta con dos orificios por los cuales atraviesan dos varillas lisas de forma que sirvan como guía para el desplazamiento deseado y a la vez como soporte para el peso que se pueda agregar al mecanismo.

Mecanismo manivela biela corredera.

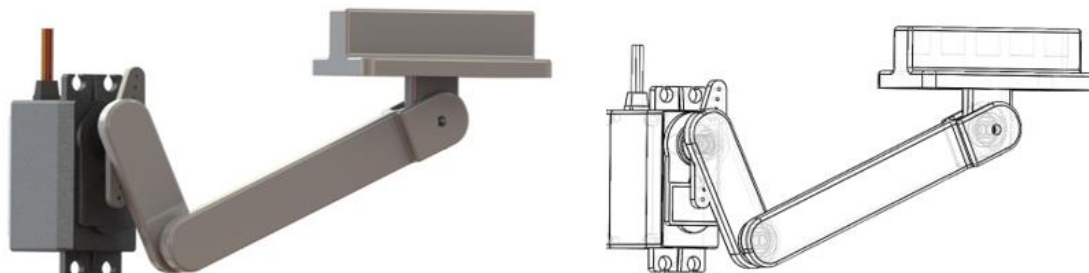


Fig. 18 Mecanismo manivela biela corredera. Fuente: los autores.

Por último, se muestra un mecanismo manivela biela corredera, que cuenta con un servomotor como actuador del movimiento. La corredera del mecanismo sobre la cual se ve el movimiento lineal, está diseñada como base para el soporte sobre el cual se posicionará el dedo a ejercitar. El cambio del sentido del movimiento, se da por el cambio en el sentido de giro del servomotor y no por el giro continuo de la manivela como en los mecanismos convencionales de este tipo.

Luego de realizar los tres diseños mostrados anteriormente, se realizó una comparación entre la efectividad del movimiento esperados, teniendo en cuenta parámetros como velocidad, fuerza de ejecución y estabilidad del movimiento. Dicha comparación se resume en la tabla 6.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Mecanismo	Tornillo sin fin	Correa dentada	Manivela biela corredera
Velocidad de movimiento	Baja	Alta	Alta
Fuerza de ejecución	Alta	Baja	Media
Estabilidad del mecanismo	Alta	Alta	Alta

Tabla 6 Comparación diseños realizados. Fuente: los autores.

Con base en esto se elige el mecanismo manivela biela corredera, ya que presenta ventajas en la implementación del diseño por ser un mecanismo compacto y poco aparatoso, caso contrario al sistema con transmisión por correa dentada el cual por necesitar varillas lisas como guías del movimiento puede ser aparatoso en la implementación; ventajas en la velocidad de ejecución con respecto al mecanismo de tornillo sin fin, el cual garantiza un buen par de fuerza para la ejecución del movimiento debido al motoreductor utilizado como actuador, pero limita su velocidad por la misma razón. Y un nivel medio con respecto a la fuerza del movimiento en comparación a los otros dos sistemas mencionados.

La corredera del mecanismo entra en contacto directo con el dedo de la mano; y ejercita las articulaciones interfalángicas distal, medial y proximal del mismo, cumpliendo con el objetivo planteado, y por su ubicación y tamaño permite la inclusión de un mecanismo que ayude a ejercitar la muñeca de la mano de forma complementaria, aumentando el número de articulaciones ejercitadas, lo que representa una mejora en el diseño final.

Con la elección del mecanismo que mejor se adapta a las necesidades de esta investigación definida, el paso siguiente es el diseño de cada una de las piezas que forman parte del sistema, junto con el entorno en el cual será implementado.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

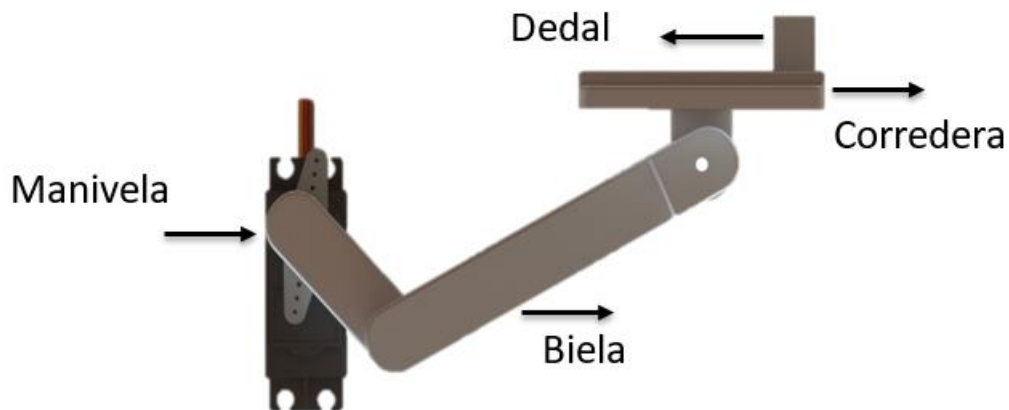


Fig. 19 Renderizado mecanismo manivela biela corredera con dedal. Fuente: los autores.

El mecanismo manivela biela corredera, es accionado por un servo motor futaba s3003 según los parámetros analizados en esta sección en el apartado de actuadores, conectado a la manivela del mecanismo, que a su vez se conecta a una biela que transmite el movimiento a la corredera, sobre la cual se ubica un dedal en el cual se posiciona la yema del dedo a ejercitar. Para la ejercitación de la muñeca en movimiento de extensión se usa una leva que aprovecha el punto de apoyo del antebrazo con la superficie del sistema desarrollado para realizar un movimiento vertical transmitido a la articulación a ejercitar.

Como ya se ha mencionado el mecanismo manivela biela corredera se encarga de generar el movimiento de flexión y extensión de las articulaciones del dedo de la mano, en los puntos de conexión de cada uno de los componentes, se usan rodamientos 623-2rs que permiten disminuir la fricción entre las piezas y dar un movimiento más fluido, y son acoplados por un eje pasante de 3 milímetros.

El desplazamiento lineal ejecutado por la corredera del sistema, se limitó a un máximo de dos centímetros y medio, esta medida se determinó mediante la medición de las manos de diferentes personas a las cuales se les pidió que posicionaran su mano de manera neutra en completa extensión y luego en manera neutra en flexión, midiendo un delta de la diferencia de estas dos medidas y sacando un promedio de la totalidad de las mediciones tomadas. Estas medidas se muestran en la tabla 7.



Medición número	Medición en extensión (cm)	Medición en flexión (cm)	Delta (cm)
1	16	13,4	2,6
2	15,3	12,8	2,5
3	16,5	13,7	2,8
4	15,8	13,3	2,5
5	16,8	14,6	2,2
6	16	13,6	2,4
7	14,9	12,6	2,3
8	17,2	14,8	2,4
9	16,5	14	2,5
10	16,2	13,8	2,4
11	13,5	11	2,5
12	15,3	12,7	2,6
13	17	14,7	2,3
14	15	12,7	2,3
15	15,2	12,8	2,4
Promedio	15,81333333	13,36666667	2,44666667

Tabla 7 Mediciones tomadas a manos de personas como referencia para el diseño del prototipo. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Para elegir las dimensiones de la manivela y la biela del mecanismo se recurre a la ecuación de desplazamiento y posición de la corredera del mecanismo que se muestra a continuación:

$$Bx = r - r * \cos\beta + \frac{r^2}{2R} * \text{sen}^2\beta$$

Donde r representa el radio de la manivela, R es el radio de la Biela y β es el ángulo de desplazamiento. Conociendo el desplazamiento deseado en la corredera del mecanismo, se utilizó una simulación en el software Matlab que se basa en la ecuación de desplazamiento mostrada anteriormente y permite variar parámetros como las medidas de la manivela y la biela del mecanismo desde la ecuación del desplazamiento. Estas medidas se modificaron hasta llegar a una distancia de 3.2 centímetros para la manivela y 8.3 centímetros para la biela, con lo que se obtiene un desplazamiento horizontal en la corredera de 2.56 centímetros aproximadamente, que es el deseado según el promedio obtenido en las mediciones de la tabla 7. Los resultados del desplazamiento mostrado por la simulación se pueden observar en la figura 20, en la cual el punto Bx representa el desplazamiento horizontal de la corredera del mecanismo.

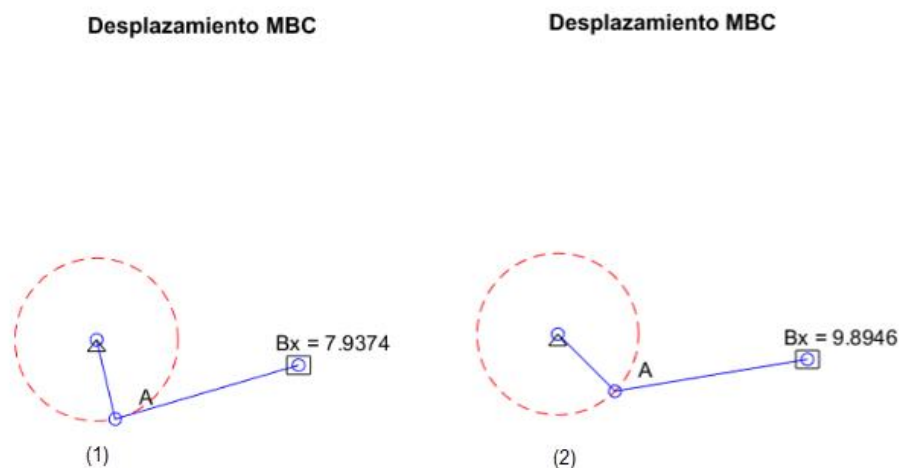


Fig. 20 Simulación desplazamiento mecanismo manivela biela corredera (1) posición inicial (2) posición final. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Teniendo en cuenta que las manos de las personas tienen diferentes tamaños y que el prototipo es diseñado para la ejercitación de cuatro dedos diferentes de la mano además de la intención de crear un prototipo generalizado; la corredera de este mecanismo se diseñó de forma que sirva como soporte para un dedal extraíble que pueda ser ubicado en 5 puntos circulares diferentes separados a una distancia de 12 milímetros de centro a centro como se muestra en la figura 21.

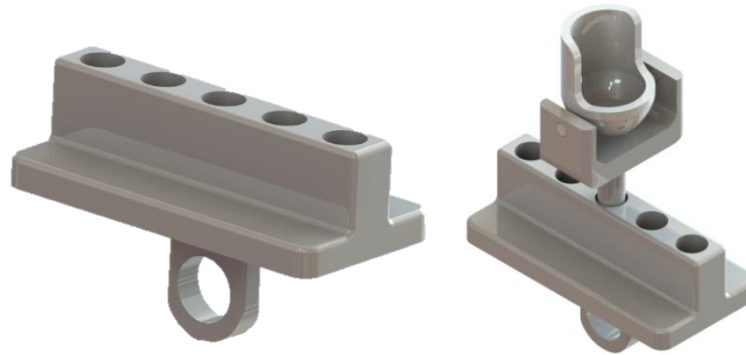


Fig. 21 Renderizado corredera y corredera con dedal acoplado. Fuente: los autores.

Dedal

El dedal del sistema, es la pieza en la que se posiciona la yema del dedo de la mano, está compuesto por dos piezas, un soporte conectado a la corredera del mecanismo manivela biela corredera, que al mismo tiempo se conecta con una base que se adapta a la anatomía de la punta del dedo. Esta conexión se hace con el fin de generar un punto de pivote que de fluidez al movimiento. Permitiendo vencer la inercia para iniciar el desplazamiento por parte del mecanismo y evitando sobre esforzar el dedo del usuario o llevarlo a posiciones críticas que podrían causar lesiones. Adicionalmente el dedal cuenta con una cinta de velcro que permite asegurar de manera efectiva el dedo al mecanismo evitando que el dedo se desacople de la base en caso de que por error se ejecute una fuerza superior a la programada en el mecanismo.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Fig. 22 Renderizado dedal. Fuente: los autores.

Leva

Como ya se ha mencionado para aprovechar el espacio, y complementa la cantidad de articulaciones de la mano que se puedan ejercitar con el prototipo. Se implementa una leva de disco con palpador de cuña. Para lograr un movimiento de elevación en un el extremo del cascaron superior del prototipo de más o menos 1.5 centímetros. Este movimiento, permite la ejercitación de la muñeca de la mano ya que realiza movimientos de flexo – extensión.

Para que la leva que se pretende implementar cumpla con el desplazamiento deseado se realizaron cálculos respectivos para el diseño mecánico de levas, para los cuales es necesario determinar el tipo de leva que se quiere diseñar y se varían parámetros como los ángulos de rotación. Las ecuaciones utilizadas para estos cálculos y los resultados obtenidos se muestran en esta sección.

Lo primero que se debe aclarar es que el perfil del desplazamiento efectuado por el seguidor de la leva corresponde a una función parabólica, establecida de la siguiente manera:

$$y = A\theta^2 + B\theta^2 + C$$

Donde la velocidad se comporta de manera triangular y la aceleración como escalones, así como se observa en la figura 23.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

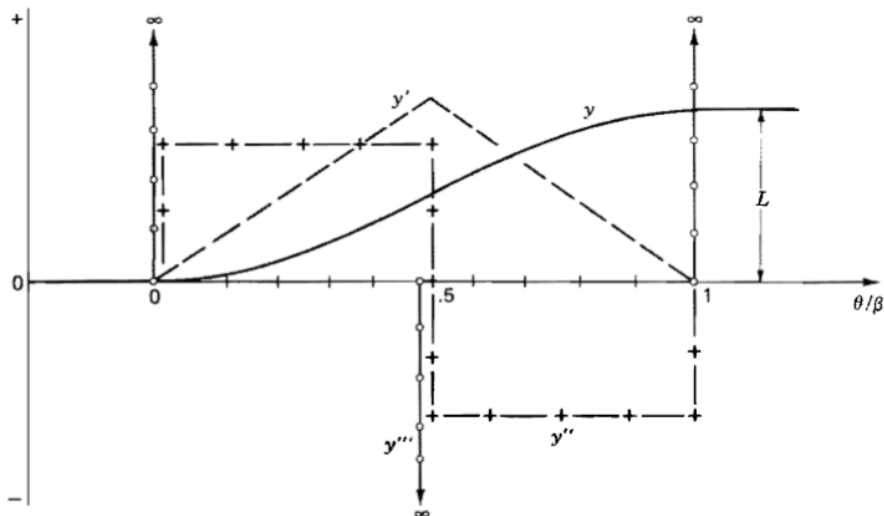


Fig. 23 Gráficas de desplazamiento, velocidad y aceleración de una leva. Fuente: (Joseph Edward Shigley, 2014)

Para obtener la función que represente la velocidad del seguidor de la leva, se realiza la primera derivación:

$$y' = 2A\theta + B$$

De igual forma, se realiza la segunda derivada para determinar la ecuación de aceleración del movimiento:

$$y'' = 2A$$

Primer tramo del movimiento:

De la ecuación de desplazamiento y la ecuación de velocidad, se puede calcular C, asignando el valor de desplazamiento cuando el ángulo θ es igual a 0, es decir, las condiciones iniciales del sistema.

$$0 = A(0)^2 + B(0) + C$$

$$C = 0$$

Para el cálculo de A y B, se reemplazan en la ecuación de desplazamiento dos condiciones, una cuando el ángulo θ es la mitad del ángulo total y otra cuando el ángulo llega al punto máximo del desplazamiento.



Condición 1 $\theta = 45^\circ$, $y = 0.015m$

Condición 2 $\theta = 22.5^\circ$, $y = 0.0075m$

Para la condición 1 se obtiene:

$$\begin{aligned}0.015m &= A(45)^2 + B(45) \\0.015m &= 2025 * A + 45 * B\end{aligned}$$

Para la condición 2 se obtiene:

$$\begin{aligned}0.0075m &= A(22.5)^2 + B(22.5) \\0.0075m &= 506.25 * A + 22.5 * B\end{aligned}$$

De esta manera y haciendo los despejes pertinentes se obtienen los valores de A y B que son:

$$A = 5.3582x10^{-10} \text{ y } B = 3.3334x10^{-4}$$

Siendo así, las ecuaciones resultantes son:

$$\begin{aligned}y &= 5.3582x10^{-10} * \theta^2 + 3.3334x10^{-4} * \theta \\y' &= 1.0764x10^{-9} * \theta \\y'' &= 1.0764x10^{-9}\end{aligned}$$

Mediante el uso de estas ecuaciones se pueden realizar la gráfica de desplazamiento del sistema. Con la que se puede comprobar que al aplicar un ángulo de rotación de 45 grados el sistema tendrá un desplazamiento vertical de 1.5 centímetros. Esta gráfica se muestra en la figura 24 y corresponde a la superficie inclinada de contacto de la leva con el palpador, se debe aclarar que el movimiento se calcula a los 45 grados debido a que la leva diseñada no da un giro continuo de 360 grados para retornar a la posición inicial, debido a que esto se controla con el sentido de giro del motor por medio de programación.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

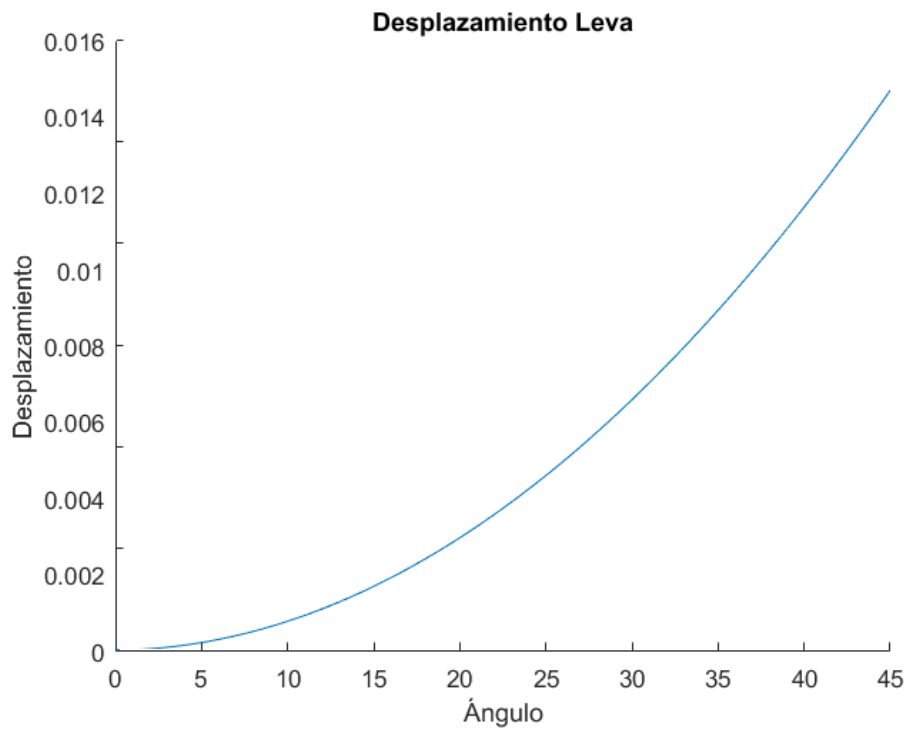


Fig. 24 Gráfica de desplazamiento de la leva diseñada. Fuente: los autores.

Finalmente, el diseño de la leva se muestra en la figura 25.

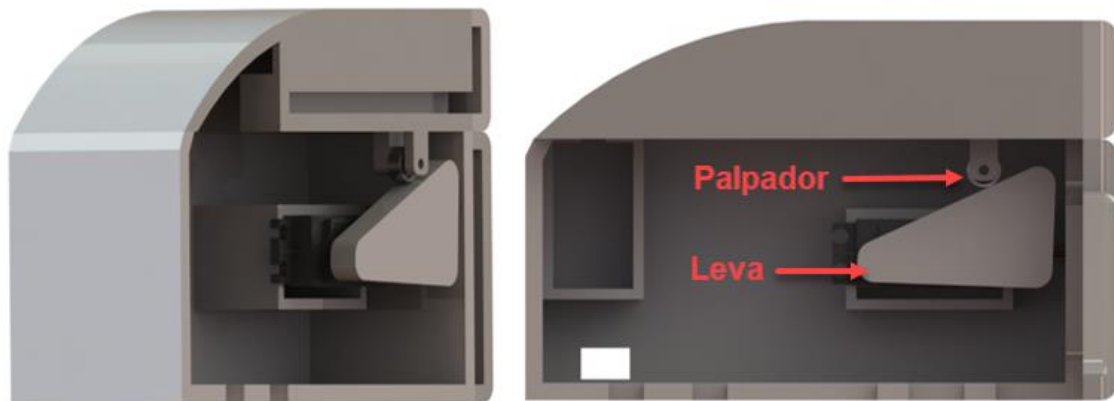


Fig. 25 Renderizado leva. Fuente: los autores.



Cascaron superior

Se diseña con el fin de servir como soporte para la mano del usuario, de igual manera esta pieza tiene una guía para la corredera del mecanismo con el fin de garantizar el desplazamiento horizontal deseado. Para que la guía de la corredera no presentara una fricción significativa en el movimiento, se imprimió en tres partes que fueron ensambladas haciendo uso del cloruro de metileno, un disolvente útil para pegar y dar acabado a las piezas impresas en PLA. Esto con el fin de garantizar una forma de impresión que mantenga las caras de contacto lo más lisas posibles.

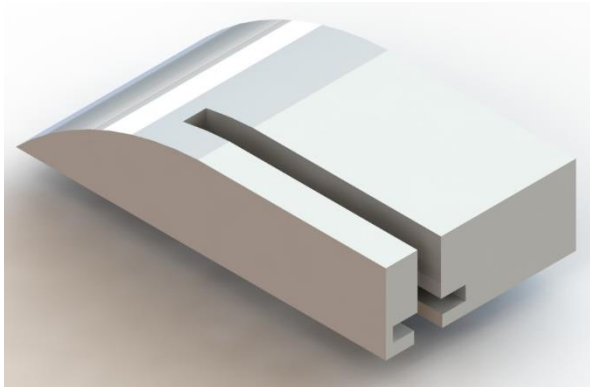


Fig. 26 Renderizado cascaron superior. Fuente: los autores.

Cascaron izquierdo

Cumple las funciones de servir como soporte para los actuadores implementados, y además en él se ubican el interruptor general, el circuito electrónico y la tarjeta de control. A la vez es parte de la carcasa total del sistema.

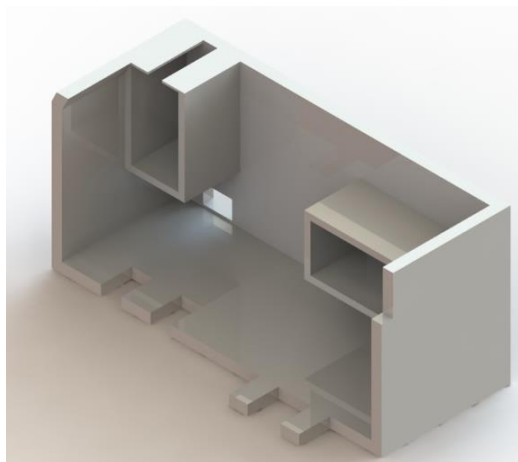


Fig. 27 Renderizado cascaron izquierdo. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Cascaron derecho

Esta pieza sirve como soporte para la pantalla LCD TFT, utilizada como interfaz HMI en para el control del dispositivo. De igual manera hace parte de la carcasa total del sistema.

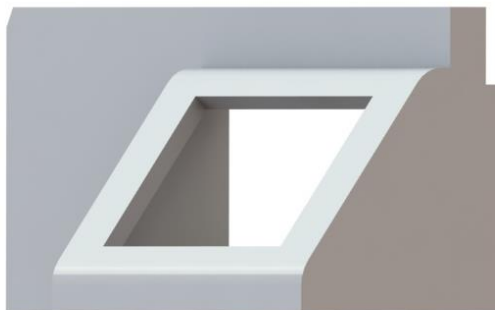


Fig. 28 Renderizado cascaron derecho. Fuente: los autores.

Soporte antebrazo

Para brindar comodidad al usuario del sistema, y a la vez ayudar a soportar el peso de la mano del mismo, se diseña un soporte sobre el cual se ubique parte del antebrazo del usuario, logrando el posicionamiento correcto para permitir el funcionamiento óptimo del sistema, la cual debe iniciar en una ubicación neutra.



Fig. 29 Renderizado soporte antebrazo. Fuente: los autores



El ensamble total de las piezas del sistema diseñado se muestra en los siguientes renderizados, en los cuales se incluyen vistas explosionadas del mismo. Una vista isométrica del diseño ensamblado se muestra en la figura 30.



Fig. 30 Renderizado ensamble total del sistema vista isométrica. Fuente: los autores.

En la figura 31 se muestra un renderizado de la vista lateral del ensamble total del sistema en la que se alcanza a observar la cara frontal del soporte para el antebrazo.



Fig. 31 Renderizado ensamble total del sistema vista lateral. Fuente: los autores.

La vista isométrica explosionada del mecanismo en el cual se observan cada uno de los componentes que conforman el prototipo se muestra en la figura 32, adicionalmente se muestra una vista frontal del explosionado en la figura 33.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

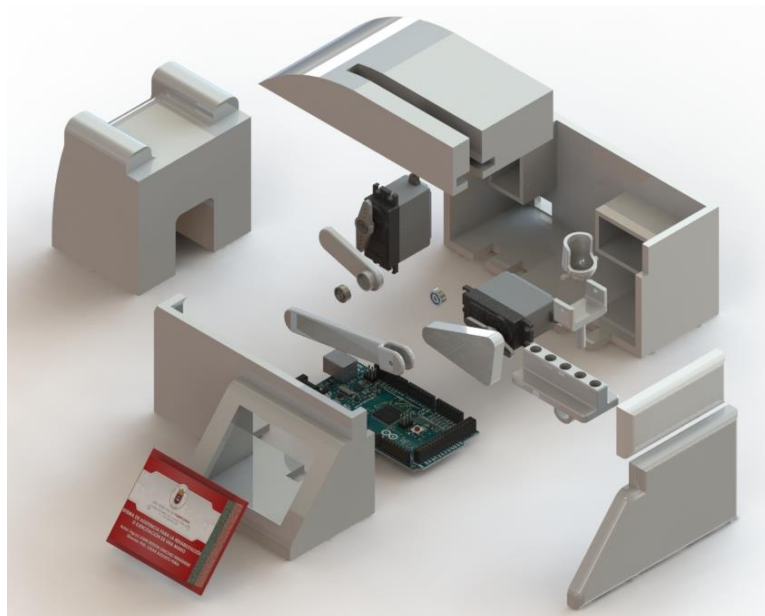


Fig. 32 Renderizado explotado del sistema vista isométrica. Fuente: los autores.



Fig. 33 Renderizado explotado del sistema vista frontal. Fuente: los autores.

En la construcción del prototipo se emplea PLA (poliácido láctico) como material de construcción de todas las piezas diseñadas para el sistema. Esto debido a que se construyen mediante impresión 3D y entre los diferentes filamentos disponibles



como el ABS (Acrilonitrilo Butadieno Estireno), HIPS (Poliestireno de alto impacto), PET (Polietileno de Tereftalato), el PLA es un material biodegradable que por sus propiedades no presenta problemas para ser usado en aplicaciones que impliquen el contacto directo con la piel, esto en cuanto a componentes tóxicos. Sin que esto afecte las propiedades mecánicas del material lo que genera una ventaja respecto a los demás filamentos usados para la impresión 3D derivados de hidrocarburos.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



3.1.3 DISEÑO ELECTRÓNICO.

Las conexiones electrónicas de los diferentes componentes del sistema desarrollado, se realizan en dos etapas, una con conexión a la tarjeta de control Arduino y otra en una etapa de alimentación ordenada mediante una PCB diseñada en el programa Eagle. La etapa de alimentación tiene como entrada un voltaje de 12 voltios que se conectan directamente a la tarjeta Arduino y regulados a 5 voltios por un regulador de voltaje lm7805, para los demás componentes, actuadores y sensores del sistema.

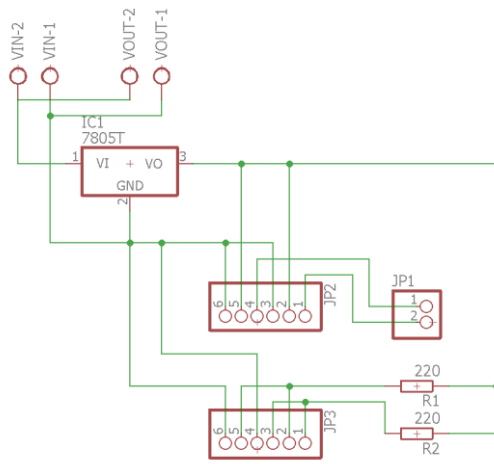


Fig. 34 Esquema de conexiones circuito electrónico. Fuente: los autores.

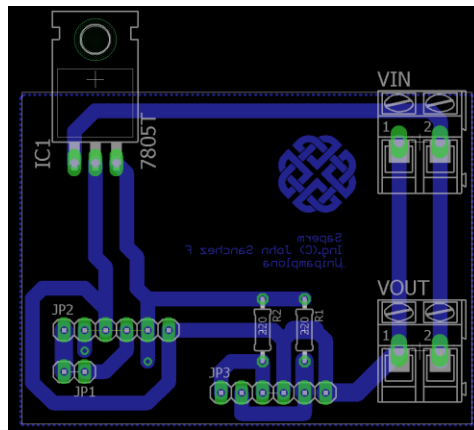


Fig. 35 Diseño PCB. Fuente: los autores.

La elección de la fuente de alimentación del sistema, se realizó teniendo en cuenta los datos de voltajes y corrientes de operación proporcionados por las hojas de características de cada uno de los componentes implementados, al igual que mediciones de consumo de corriente realizadas con el sistema en funcionamiento haciendo uso de una fuente de voltaje variable. Con los datos obtenidos se determinó usar un cargador de 12 voltios y 1.5 amperios suficientes para alimentar por completo el sistema sin ningún tipo de dificultad ni sobredimensionamiento.

Componente	Voltaje de operación (hoja de características)	Corriente de operación (hoja de características)
Servomotores	4.8 V – 6 V	0.6 A
Tarjeta Arduino	5 V – 20 V	0,2 A
Pantalla LCD TFT	5 V	0,3 A



Componente	Voltaje de operación (medida)	Corriente de operación (medida)
Servomotores	5 V	0,57 A - 0,62 A
Tarjeta Arduino	12 V	0,47 A
Pantalla LCD TFT	5 V	0,3 A

Tabla 8 Tabla de voltajes y corrientes. Fuente: los autores.

La interacción de los diferentes componentes de sistema desarrollado y la fuente de alimentación se muestra en la figura 36 en la cual se ve la comunicación entre los servomotores que reciben las ordenes de activación y desactivación de la tarjeta de control Arduino 2560, que a su vez recibe datos de entrada desde los sensores de posición finales de carrera los cuales limitan el movimiento del mecanismo diseñado. Del lado derecho de la imagen se ve la pantalla LCD TFT que se comunica con la tarje Arduino para recibir datos de entrada ingresados por el usuario para determinar el control del funcionamiento del sistema.

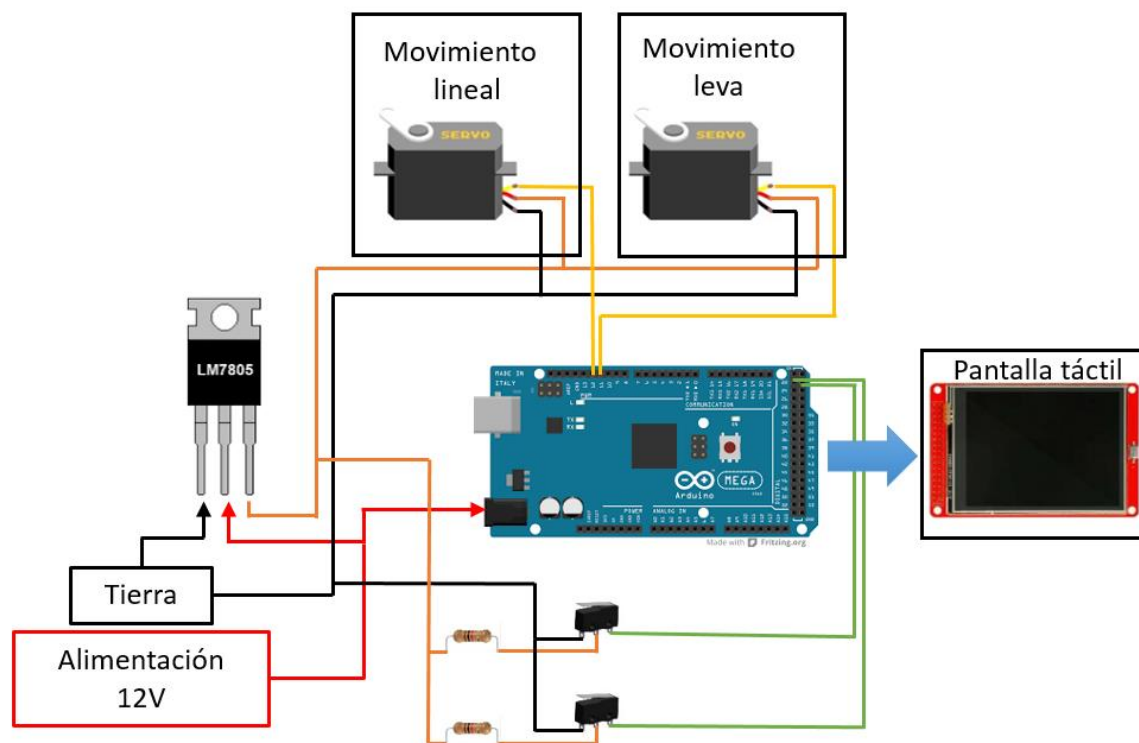


Fig. 36 Esquema general de conexiones. Fuente: los autores.



3.1.4 DISEÑO DE SOFTWARE

La programación del sistema se realiza mediante el guide de software libre propio de Arduino, con la importación de las librerías especializadas para el control de servomotores y la pantalla LCD TFT que como ya se ha mencionado funciona como una shield para Arduino.

En un principio se realizó la programación de la pantalla por aparte con el fin de aprender y entender la forma correcta de utilizar la librería Adafruit_TFTLCD, con la cual se hace el llamado de las diferentes funciones que permiten graficar texto y formas geométricas, en una escala de colores RGB. A demás de esto la pantalla permite la importación de imágenes en formato BMP de 24 bits desde una tarjeta micro SD que debe tener un formato Fat. Para esto se utilizan las librerías SD.h y SPI.h que permiten la comunicación de la pantalla con la SD y los pines SPI de la tarjeta Arduino. Esta contiene a la función BUFFPIXEL 20 que es la encargada de traducir los pixeles de las imágenes a una matriz de datos para ser interpretados por la tarjeta Arduino y así puedan ser visualizados mediante la pantalla LCD TFT.

Al ser esta una pantalla de tipo táctil, los gráficos realizados corresponden a un texto guía que muestra las opciones programadas a las que tiene acceso el usuario por medio de botoneras, que al ser seleccionadas envían una señal de activación a la tarjeta Arduino, la cual es interpretada posteriormente como un dato de memoria que es una condición para la ejecución de los movimientos de los actuadores programados. Esta señal es enviada mediante los pines digitales de conexión expuestos en la sección de ingeniería del detalle.

El esquema general de la programación realizada en el dispositivo se muestra en el diagrama de flujo de la figura 37. El ciclo de funcionamiento comienza con la muestra de imágenes en la pantalla LCD. Luego de las imágenes de presentación se entra en un menú por medio del cual se obtienen los datos de usuario necesarios para la activación y ejecución de los movimientos del sistema. Mientras se ejecutan los movimientos se muestra en pantalla el estado del movimiento en ejecución, para luego entrar nuevamente a un menú de opciones que permiten continuar con los movimientos programados, seleccionar una nueva serie de movimientos o en caso contrario finalizar la ejecución del programa y apagar el dispositivo.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

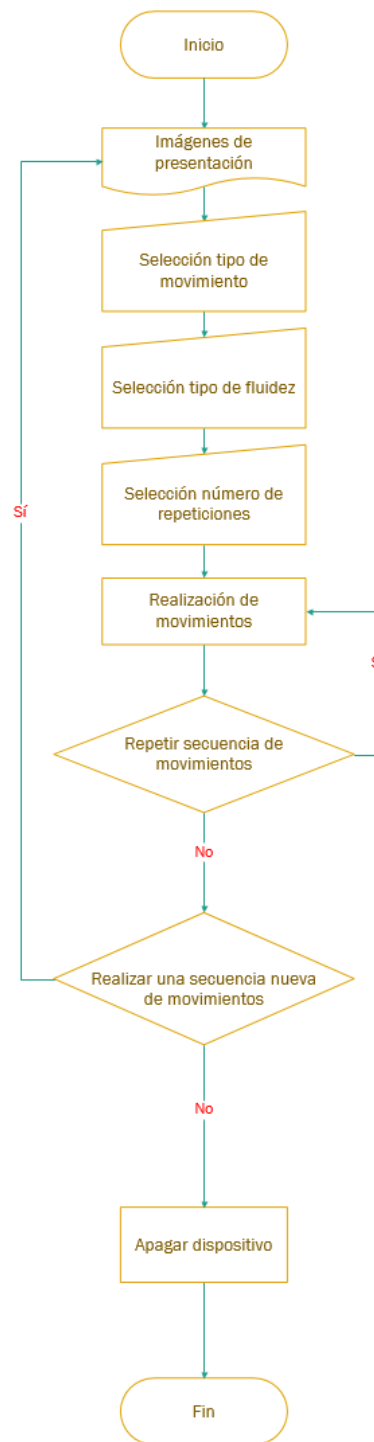


Fig. 37 Diagrama de flujo general de la programación del sistema. Fuente: los autores.



El llamado de las librerías utilizadas para el control de la pantalla y los servo motores, al igual que la definición de variables básicas para el correcto funcionamiento de la pantalla junto con el panel táctil se muestra a continuación.

```
#include <Adafruit_GFX.h> // Libreria de graficos
#include <Adafruit_TFTLCD.h> // Libreria de LCD
#include <SD.h> // Libreria de tarjeta SD
#include <SPI.h> // Libreria bus SPI
#include <TouchScreen.h> // Libreria del panel tactil
#include <Servo.h>

Servo myservo1;
Servo myservo2;
Servo myservo3;
#define LCD_CS A3 // Definimos los pines del LCD
#define LCD_CD A2 // para poder visualizar elementos graficos
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4

#define YP A3 // Pin analogico A1 para ADC
#define XM A2 // Pin analogico A2 para ADC
#define YM 9
#define XP 8

// Definimos la presion máxima y minima que podemos realizar sobre el panel
#define MINPRESSURE 1
#define MAXPRESSURE 1000
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 654); // 654 valor resistencia de la pantalla

short TS_MINX = 200; // Coordenadas del panel tactil para delimitar
short TS_MINY = 120; // el tamaño de la zona donde podemos presionar
short TS_MAXX = 850; // y que coincida con el tamaño del LCD
short TS_MAXY = 891;
```

Fig. 38 Declaración de librerías. Fuente: los autores.

Para poder usar de manera sencilla la gama de 8 colores disponibles por la pantalla LCD se declaran los nombres de estos colores, para no llamarlos según su referente hexadecimal en cada línea de código que lo requiera. Este paso y la declaración de las variables utilizadas se muestra en la figura 39.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET); // Instancia LCD

#define BLACK 0x0000 // Definimos los colores para poder referirnos a ellos con su nombre
#define RED 0xF800 // en lugar de usar el código hexadecimal de cada uno.
#define GREEN 0x07E0
#define WHITE 0xFFFF
#define BLUE 0x001F
#define CYAN 0x07FF
#define YELLOW 0xFFE0
#define MAGENTA 0xF81F

int X; // Variables que almacenaran la coordenada
int Y; // X, Y donde presionemos y la variable Z
int Z; // almacenara la presion realizada

int a = 0; // variable "flag" para control rebotes
int sm= 0; // variable tipo de movimiento seleccionado
int sf= 0; // variable tipo de movimiento seleccionado
int sr= 0; // variable número repeticiones seleccionado
int inicio=0;
int i=0;

int mov=0;
int rep;
int pausa;
int volver=0;
int bandera;
int ang, ang1;

int sad=22, saf=23, pos=0, conteo=0;
```

Fig. 39 Declaración de variables. Fuente: los autores.

En la ejecución del programa se inicializa la lectura de la tarjeta micro SD para importar las imágenes en formato BPN de 24 bits. En este punto se debe hacer una salvedad, debido a que las pantallas TFT de 2.4 pulgadas disponibles en el mercado, presentan una variedad de drivers de control (ILI9325, ILI9341, HX8347G, HX8357), el cual debe ser declarado para la inicialización de la pantalla. Es necesario conocer con exactitud el controlador adecuado. En el caso pantalla utilizada el driver es el 0x9341.

El no conocer el driver exacto con el que opera la pantalla, genera conflictos de comunicación al momento de inicializar la lectura de la tarjeta SD. Además de que

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



se ve afectada la forma en la que se lee la matriz resistiva de la pantalla táctil en cuando a la orientación de las coordenadas X y Y.

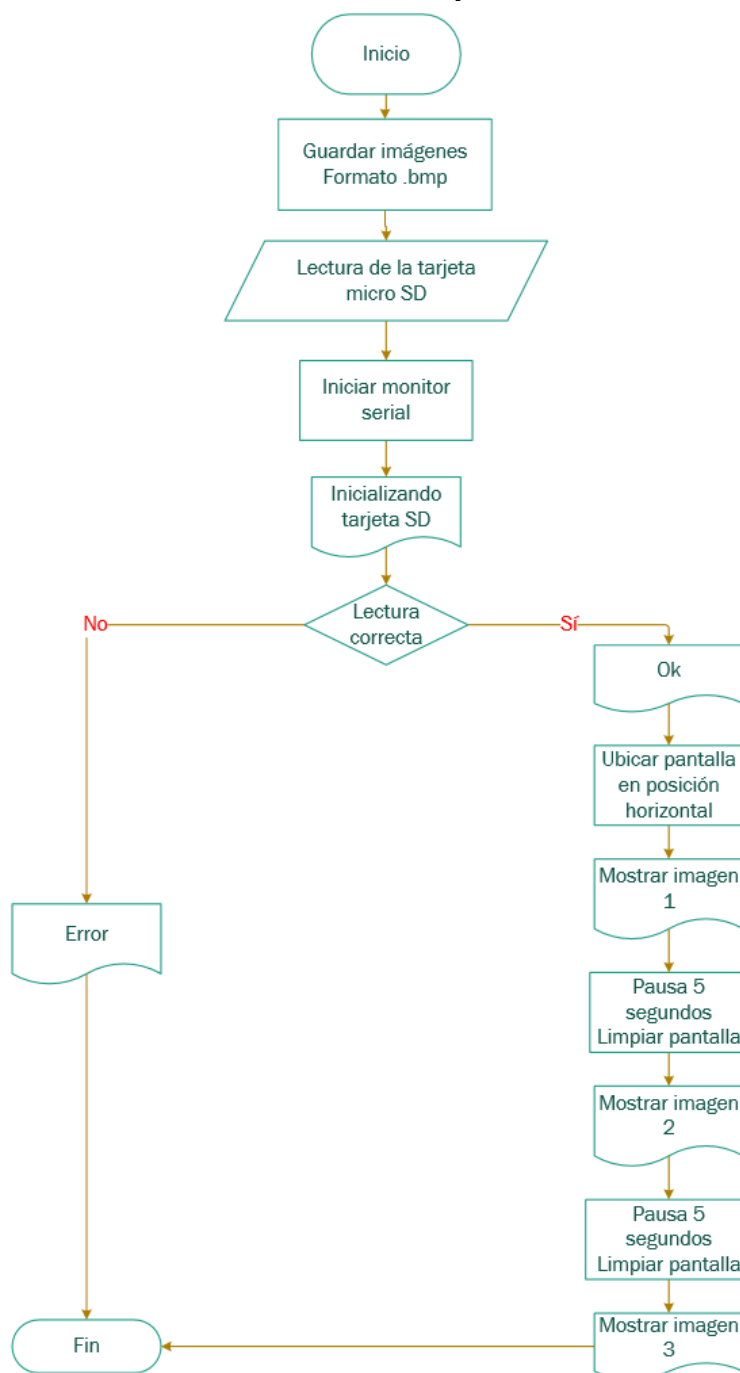


Fig. 40 Diagrama de flujo programación imágenes de presentación. Fuente: los autores.



La lectura de la SD es necesaria para poder mostrar en pantalla las imágenes de presentación del sistema. La lógica y el orden en el que se presentan las imágenes se muestra en la figura 40. Para confirmar que la lectura de las imágenes guardadas en la SD se realizó correctamente, se recurre al monitor serial, en el cual se muestra un mensaje de error en caso de no leer la SD y detiene el proceso del código de programación. Los comandos utilizados para esta parte del código son los mostrados en la figura 41.

```
void setup()
{
  myservo1.attach(12);
  myservo2.attach(11);
  myservo3.attach(13);
  myservo1.write(45);
  myservo2.write(0);
  Serial.begin(9600); // Iniciamos el puerto serie para comprobar
                    // la comunicacion con la tarjeta microSD
  tft.reset();
  tft.begin(0x9341); // Iniciamos el LCD especificando el controlador ILI9341.

  Serial.print(F("Inicializando tarjeta SD..."));

  if (!SD.begin(SD_CS)) // Si se produce un error al intentar acceder
  { // a la tarjeta SD, lo mostramos por el Serial Monitor
    Serial.println(F("Error!"));
    return;
  }
  Serial.println(F("OK!"));

  tft.setRotation(3); // Establecemos la posicion de la pantalla Vertical

  bmpDraw("1.bmp", 0, 0); // Mostramos una imagen en las coordenadas 0,0

  delay(1000);

  tft.setRotation(3); // Establecemos la posicion de la pantalla Horizontal

  bmpDraw("2.bmp",0,0); // // Mostramos otra imagen en las coordenadas 0,0

  delay(1000);
}
```

Fig. 41 Código imágenes de presentación. Fuente: los autores.



El paso siguiente, es el ingreso a la función de selección de tipo de movimiento desde la cual se recibe como dato de entrada la elección del usuario entre dos opciones presentadas, movimiento de articulación del dedo de la mano o el movimiento de la articulación del dedo de la mano y de la muñeca de manera alternada, según esta elección se da la activación de uno o dos motores tal y como se muestra en el diagrama de flujo de la figura 42.

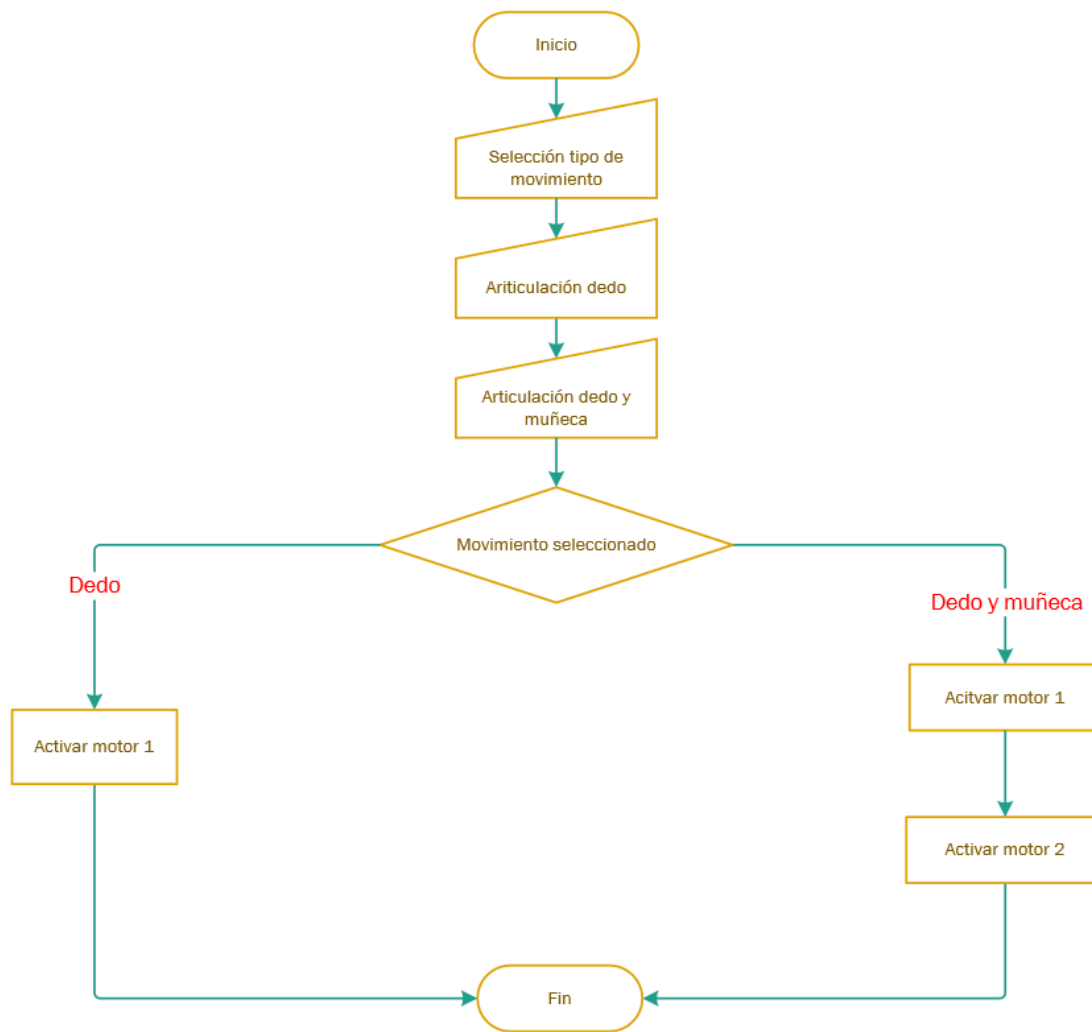


Fig. 42 Diagrama de flujo selección tipo de movimiento. Fuente: los autores.



El menú para la selección de tipo de movimiento y demás opciones que se dan al usuario para el control del sistema, se realiza con el uso de comandos específicos, propios de la librería Adafruit_TFTLCD como lo son `tft.fillRect` que imprime un rectángulo sólido con bordes redondeados recibiendo como parámetros las coordenadas en las cuales se desea que este ubicado el rectángulo, el radio del redondeo que se realiza a las esquinas y el color de relleno seleccionado de los 8 disponibles por la pantalla. La definición, uso y explicación de cada comando utilizado se muestra en las secciones de código correspondientes.

La figura 43 muestra el código correspondiente a la selección de opciones para el tipo de movimiento que se ejecutara en el sistema, bien sea para la ejercitación del dedo o la ejercitación del dedo y la muñeca.

```
tft.fillRect(WHITE); // Se pone la pantalla en blanco
delay(500);
tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y inicial x final y final
tft.setCursor(20,15); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Elija el tipo de"); // Escribimos por pantalla
tft.setCursor(20,31);
tft.println("fluidez que desea ");
tft.setCursor(20,46);
tft.println("para el movimiento");

tft.fillRect(40, 100, 250, 40, 15, CYAN); //Dibujamos un rectangulo con relleno de color CYAN
tft.drawRoundRect(40, 100, 250, 40, 15, RED); // Dibujamos un rectangulo sin relleno con border ROJOS
tft.setCursor(50,110);
tft.println("Pausada");

tft.fillRect(40, 150, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 250, 40, 15, RED);
tft.setCursor(50,160);
tft.println("Fluida");
```

Fig. 43 Código selección de tipo de movimiento. Fuente: los autores.

Una vez se muestran en pantalla las botoneras con las opciones disponibles, se requiere capturar el dato de entrada según la selección del usuario del sistema, para esto se utiliza la función "lecturaPanel" que se encarga de leer la presión ejercida en la pantalla táctil para luego ser comparada en las coordenadas

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



habilidades para cada opción de selección mostrada. El uso de esta función se muestra en la figura 44.

```
void lecturaPanel()
{
    digitalWrite(13, HIGH);
    TSPoint p = ts.getPoint(); // Realizamos lectura de las coordenadas
    digitalWrite(13, LOW);

    pinMode(XM, OUTPUT); // La librería utiliza estos pines como entrada y salida
    pinMode(YP, OUTPUT); // por lo que es necesario declararlos como salida justo
                          // despues de realizar una lectura de coordenadas.

    // Mapeamos los valores analogicos leidos del panel tactil (0-1023)
    // y los convertimos en valor correspondiente a la medida del LCD 320x240
    X= map(p.y, TS_MAXX, TS_MINX, tft.width(), 0);
    Y = map(p.x, TS_MINY, TS_MAXY, tft.height(), 0);
    Z = p.z;
}
```

Fig. 44 Código entrada de dato táctil. Fuente: los autores.

Inmediatamente después de recibir la selección del tipo de movimiento que debe realizar el sistema, se pasa a seleccionar la fluidez con la que se debe ejecutar el movimiento que se trasmite al dedo del paciente, de acuerdo a esto se modifica el avance el ángulo de avance de los servo motores, como se muestra en el diagrama de flujo de la figura 45.

Como ya se mencionó la lectura del dato ingresado por medio de la pantalla táctil se hace mediante la función “lecturaPanel” que es necesaria nuevamente en este menú, el código correspondiente a esta sección es mostrado en la figura 46.

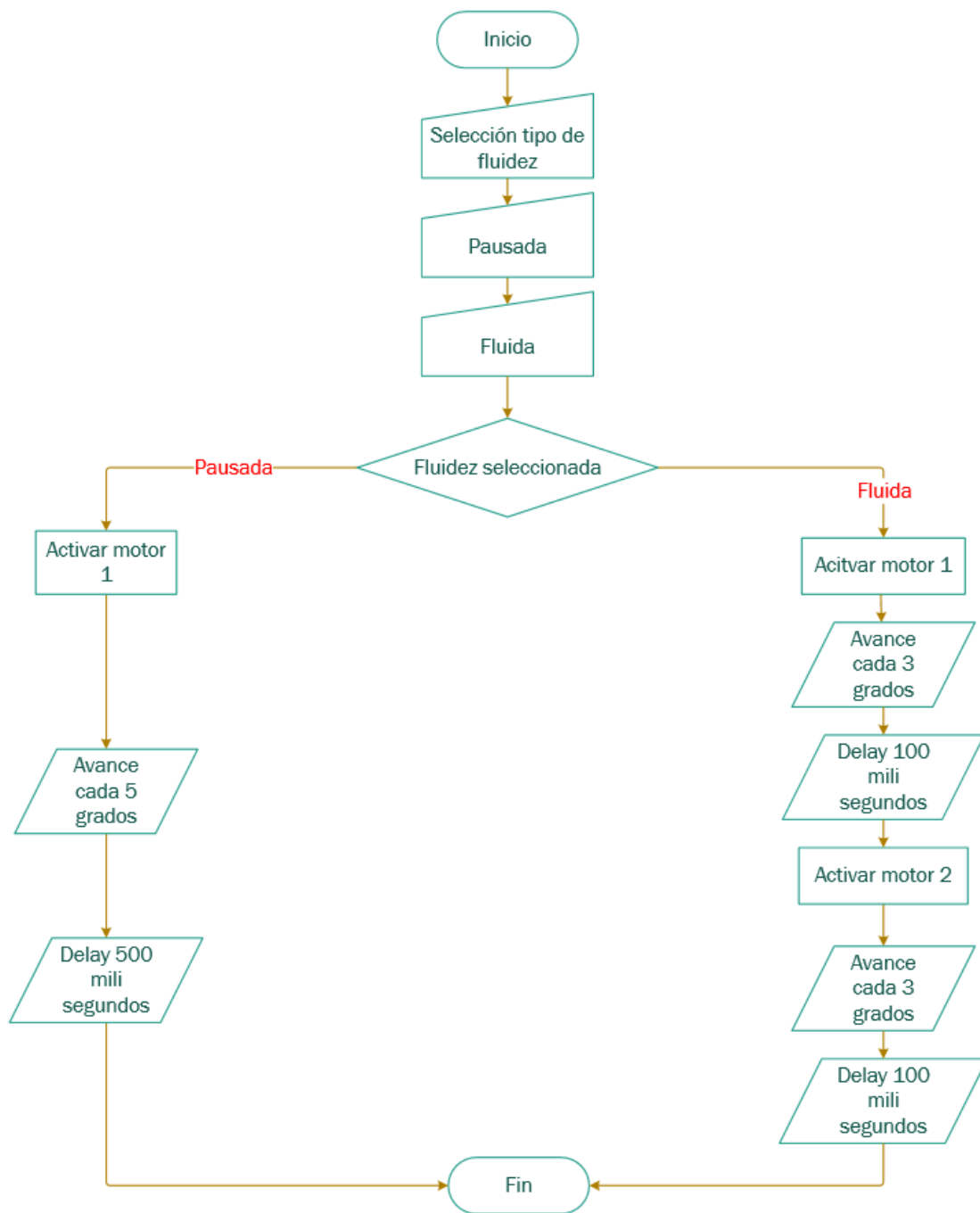


Fig. 45 Diagrama de flujo selección de fluidez del movimiento. Fuente: los autores.



```
tft.fillScreen(WHITE); // Se pone la pantalla en blanco
delay(500);
tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y inicial x final y final
tft.setCursor(20,15); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Elija el tipo de"); // Escribimos por pantalla
tft.setCursor(20,31);
tft.println("fluidez que desea ");
tft.setCursor(20,46);
tft.println("para el movimiento");

tft.fillRoundRect(40, 100, 250, 40, 15, CYAN); //Dibujamos un rectangulo con relleno de color CYAN
tft.drawRoundRect(40, 100, 250, 40, 15, RED); // Dibujamos un rectangulo sin relleno con border ROJOS
tft.setCursor(50,110);
tft.println("Pausada");

tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 250, 40, 15, RED);
tft.setCursor(50,160);
tft.println("Fluida");
```

Fig. 46 Código selección de tipo de fluidez. Fuente: los autores.

Por último, se debe leer la cantidad de veces que se repiten los movimientos seleccionados, esta opción se da mediante 6 botoneras con un incremento de 5 repeticiones desde 5 a 30. De acuerdo a la opción seleccionada se define el límite de ciclo de movimientos, esto se muestra en el flujograma de la figura 47.

El segmento de código para la lectura del número de movimientos seleccionados es mostrado en la figura 48. En este caso se puede evidenciar de mejor manera la distribución de la pantalla con diferentes puntos de presión habilitados para tomar un dato de entrada táctil. Para lo cual nueva mente se hace uso de la función de "lecturaPanel".

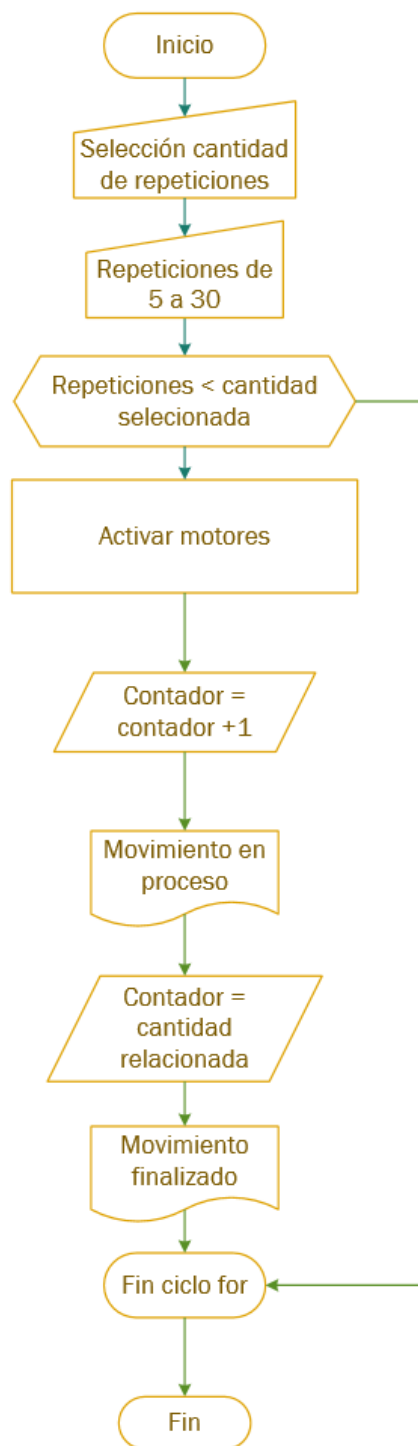


Fig. 47 Diagrama de flujo selección cantidad de repeticiones. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
tft.fillScreen(WHITE);// Ponemos la pantalla en blanco
delay(500);
tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y inicial x final y final
tft.setCursor(20,15); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Elija la cantidad"); // Escribimos por pantalla
tft.setCursor(20,31);
tft.println("de repeticiones ");
tft.setCursor(20,46);
tft.println("que desea");

tft.fillRoundRect(40, 100, 80, 40, 15, CYAN);
tft.drawRoundRect(40, 100, 80, 40, 15, RED);
tft.setCursor(50,110);
tft.println("5");

tft.fillRoundRect(40, 150, 80, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 80, 40, 15, RED);
tft.setCursor(50,160);
tft.println("10");

tft.fillRoundRect(130, 100, 80, 40, 15, CYAN);
tft.drawRoundRect(130, 100, 80, 40, 15, RED);
tft.setCursor(140,110);
tft.println("15");

tft.fillRoundRect(130, 150, 80, 40, 15, CYAN);
tft.drawRoundRect(130, 150, 80, 40, 15, RED);
tft.setCursor(140,160);
tft.println("20");

tft.fillRoundRect(220, 100, 80, 40, 15, CYAN);
tft.drawRoundRect(220, 100, 80, 40, 15, RED);
tft.setCursor(230,110);
tft.println("25");

tft.fillRoundRect(220, 150, 80, 40, 15, CYAN);
tft.drawRoundRect(220, 150, 80, 40, 15, RED);
tft.setCursor(230,160);
tft.println("30");
```

Fig. 48 Código selección cantidad de repeticiones. Fuente: los autores

Con los datos de entrada completados, se hace el llamado de la función "movimiento" en la cual con una serie de condicionales y variables de lectura se dan los valores de ángulos y esperas de avance a cada uno de los motores que para así iniciar con el movimiento programado tal y como se muestra en las siguientes imágenes.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Al mismo tiempo que los motores mueven el mecanismo, se leen datos digitales de entrada proporcionados por los sensores finales de carrera que ayudan a llevar un conteo en una variable de tipo entero que se actualiza cada vez que el mecanismo realiza un movimiento completo, dato que es comparado con el número de repeticiones seleccionada por el usuario para cumplir y en el momento que sean iguales da por terminado el ciclo y pasa a la parte final del programa.

```
void movimiento(int sm, int sf, int sr)
{
  //////////////////////////////////////
  mov=0;
  myservol.write(0);
  if (sm==1)
  {
    if(sf==1)
    {
      pausa=500;
      for(rep = 1; rep <= sr; rep ++)
      {
        if(digitalRead(sad)==0)
        {
          //MOVER ADELANTE
          while(digitalRead(saf)==1)
          {
            Serial.println("MOVIENDO HACIA ADELANTE");
            Serial.println("SAF");
            Serial.println(digitalRead(saf));
            myservol.write(pos);
            pos=pos+3;
            delay(pausa);
          }
          Serial.println("SAF");

          Serial.println(digitalRead(saf));
          Serial.println("SAD");

          Serial.println(digitalRead(saf));
        }
      }
    }
  }
}
```

Fig. 49 Código función de movimiento (1). Fuente: los autores.



```
if(digitalRead(saf)==0)
{
  //MOVER ATRAS
  while(digitalRead(sad)==1)
  {
    Serial.println("MOVIENDO HACIA ATRAS");
    Serial.println("SAD");
    Serial.println(digitalRead(sad));
    myservol.write(pos);
    pos=pos-3;
    delay(pausa);
  }
  Serial.println("CONTEO");
  Serial.println(rep);
}

tft.fillCircle(130, 215, 5, WHITE);
tft.drawCircle(130, 215, 8, WHITE);
tft.fillCircle(160, 215, 5, WHITE);
tft.drawCircle(160, 215, 8, WHITE);
tft.fillCircle(190, 215, 5, WHITE);
tft.drawCircle(190, 215, 8, WHITE);
delay(100);
tft.fillCircle(130, 215, 5, RED);
tft.drawCircle(130, 215, 8, BLACK);
delay(100);
tft.fillCircle(160, 215, 5, RED);
tft.drawCircle(160, 215, 8, BLACK);
delay(100);
tft.fillCircle(190, 215, 5, RED);
tft.drawCircle(190, 215, 8, BLACK);
}
}
```

Fig. 50 Código función de movimiento (2). Fuente: los autores.



```
if(sf==2)
{
  pausa=100;
  for(rep = 1; rep <= sr; rep ++ )
  {
    if(digitalRead(sad)==0)
    {
      //MOVER ADELANTE
      while(digitalRead(saf)==1)
      {
        Serial.println("MOVIENDO HACIA ADELANTE");
        Serial.println("SAD");
        Serial.println(digitalRead(saf));
        myservol.write(pos);
        pos=pos+2;
        delay(pausa);
      }
    }
  }

  if(digitalRead(saf)==0)
  {
    //MOVER ATRAS
    while(digitalRead(sad)==1)
    {
      Serial.println("MOVIENDO HACIA ATRAS");
      Serial.println("SAF");
      Serial.println(digitalRead(sad));
      myservol.write(pos);
      pos=pos-2;
      delay(pausa);
    }
    Serial.println("CONTEO");
    Serial.println(rep);
  }
}
```

Fig. 51 Código función de movimiento (3). Fuente: los autores.



Como se puede ver el sentido de giro de los servo motores cambia según los datos recibidos por las variables saf (sensor afuera) y sad (sensor adentro) que corresponden a los finales de carrera ya mencionados. Este ciclo se repite hasta terminar con el ciclo asignado por usuario y retorna una variable de bandera con la cual se toman las últimas acciones a las que tiene acceso el usuario.

Con el fin de garantizar que los ejercicios realizados por el sistema puedan ser aplicados a los diferentes dedos de la mano, por medio de la programación se da la opción al usuario de repetir la última serie de movimientos programada, o por el contrario de programar una nueva serie de movimientos, esto debido a que según el criterio del fisioterapeuta la cantidad de repeticiones puede aumentar o disminuir, se puede intercalar la fluidez de aplicación de los movimientos y los diferentes parámetros para que la terapia sea efectiva. Esto se logra llamando nuevamente a las funciones encargadas del movimiento o reiniciando las variables del programa para que reciban los nuevos requerimientos del usuario según sea el caso.

Si el usuario no quiere efectuar ninguna de las dos opciones mencionadas el programa muestra en pantalla una imagen desde la memoria micro SD que indica que el sistema puede ser apagado, finalizando por completo el ciclo de programación como se muestra en el diagrama de flujo de la figura 52 y el segmento de código de la figura 53.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

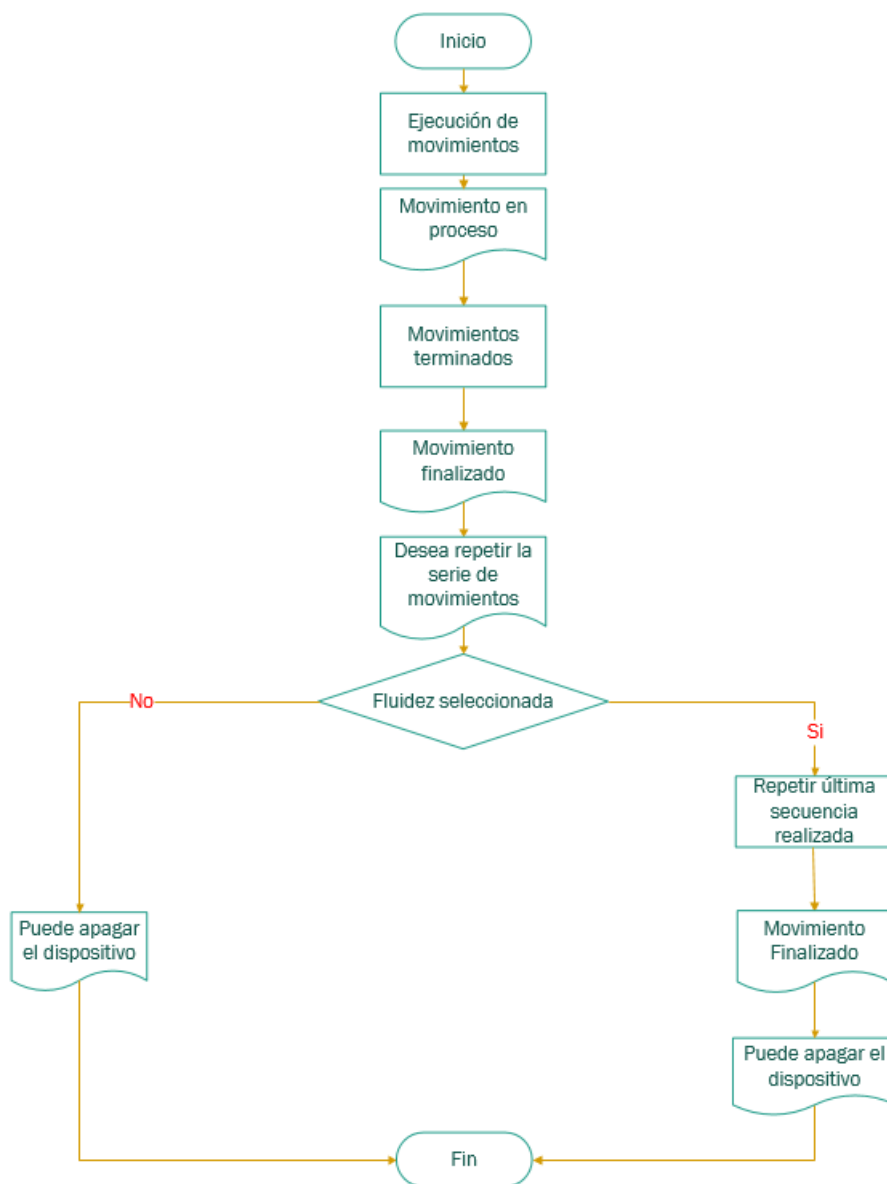


Fig. 52 Diagrama de flujo continuación o finalización del movimiento. Fuente. Los autores



```
tft.fillScreen(WHITE);
delay(500);
tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y inicial x final y final
tft.setCursor(20,18); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Desea repetir la "); // Escribimos por pantalla
tft.setCursor(20,34);
tft.println("serie de movimientos ? ");

tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 100, 250, 40, 15, RED);
tft.setCursor(50,110);
tft.println("Si");

tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 250, 40, 15, RED);
tft.setCursor(50,160);
tft.println("No");

tft.fillScreen(WHITE);
delay(500);
tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y inicial x final y final
tft.setCursor(20,18); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Desea repetir la "); // Escribimos por pantalla
tft.setCursor(20,34);
tft.println("serie de movimientos ? ");

tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 100, 250, 40, 15, RED);
tft.setCursor(50,110);
tft.println("Si");

tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 250, 40, 15, RED);
tft.setCursor(50,160);
tft.println("No");
    if (bandera==0)
    {
        tft.fillScreen(WHITE);
    }
    delay(500);
    bandera=bandera+1;
    tft.setRotation(3); // Establecemos la posicion de la pantalla Horizontal
    bmpDraw("4.bmp",0,0); // // Mostramos otra imagen en las coordenadas 0,0
                        // Para apagar finalizar el código y apagar el sistema
}
    bandera=bandera+1;
}

//fin del codigo
```

Fig. 53 Código continuación o finalización del movimiento. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Capítulo IV. RESULTADOS.

4.1 DESCRIPCIÓN DEL FUNCIONAMIENTO DEL DISPOSITIVO

Se obtiene como resultado un prototipo compacto de bajo costo, compuesto por un mecanismo manivela biela corredera, más la implementación de una leva para un movimiento vertical complementario, controlado por una tarjeta Arduino mega 2560, que a su vez recibe órdenes desde una pantalla táctil. Que cumple con los objetivos de la investigación.

Para el uso adecuado del dispositivo se deben seguir los siguientes pasos:

- Energizar el sistema activando el interruptor ubicado en la parte posterior del dispositivo.
- Ubicar sobre la superficie del prototipo la palma de la mano, y de igual manera del antebrazo en el soporte indicado. Ubicar el dedal a la distancia necesaria según la mano del paciente y por ultimo sujetar de manera adecuada la yema del dedo en el dedal diseñado para este fin.
- Al estar energizado en la pantalla táctil se iniciará una presentación de tres imágenes por un espacio de tiempo de dos segundos para cada imagen, relacionadas al nombre y los autores del proyecto como se muestra a continuación.



Fig. 54 Imagen de presentación escudo de la universidad de Pamplona. Fuente: los autores.



Fig. 55 Imagen de presentación 2 logo Ingeniería mecatrónica. Fuente: Jairo Roberto.



Fig. 56 Imagen de presentación 3 información de los autores. Fuente: los autores.

- Seguido de las imágenes de presentación, se inician las opciones de elección de tipo de movimiento, elección de tipo de fluidez del movimiento y por último la cantidad de veces que se desea realizar el ciclo de movimiento. La cuales solo se actualizan hasta que el usuario elige entre las opciones mostradas.



Elija el tipo de movimiento que desea ejecutar

Articulacion dedo

Art. dedo y muñeca

Fig. 57 Menú selección tipo de movimiento. Fuente: los autores.

Elija el tipo de fluidez que desea para el movimiento

Pausada

Fluida

Fig. 58 Menú selección tipo de fluidez. Fuente: los autores.

Elija la cantidad de repeticiones que desea

5

15

25

10

20

30

Fig. 59 Menú selección cantidad de repeticiones. Fuente: los autores

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



- Luego de seleccionar los parámetros de funcionamiento para el sistema anteriormente mostrados, se da inicio a la ejecución del movimiento. Mientras se esté efectuando el movimiento seleccionado, se muestra en la pantalla una imagen de espera, que indica que el movimiento está en proceso.



Fig. 60 Mensaje movimiento en proceso mostrado en pantalla. Fuente: los autores.

- Al terminar el ciclo de movimientos, se muestra en pantalla un mensaje que indica que el movimiento programado a finalizado.

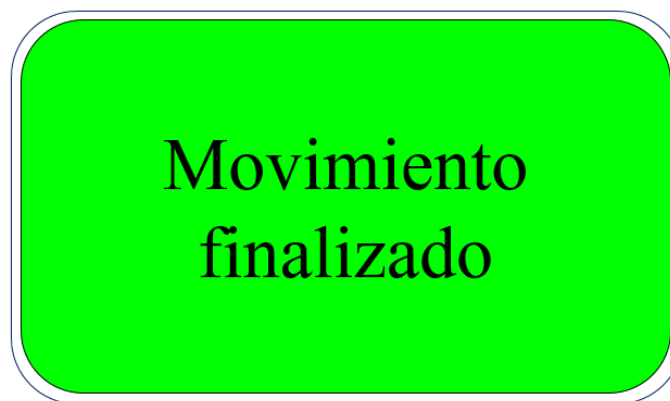


Fig. 61 Mensaje movimiento finalizado mostrado en pantalla. Fuente: los autores.



- Al terminar la serie de movimientos programada se da la opción al usuario de repetir la última configuración de movimientos programada como se muestra en la figura 62. En caso de que no quiera repetirla, se muestra un mensaje en el cual se da la opción de programar una nueva serie de movimientos, lo que reiniciaría las opciones mostradas en pantalla desde la elección del tipo de movimiento como se ve en la figura 63.

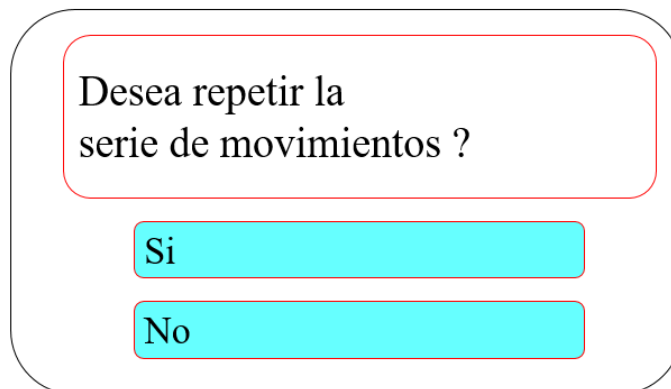


Fig. 62 Menú repetición de movimientos. Fuente: los autores.

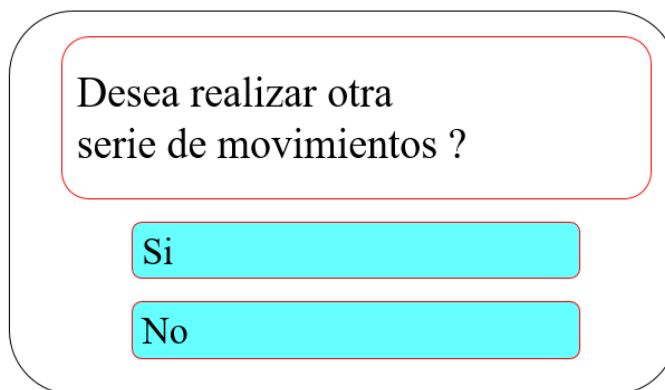


Fig. 63 Menú selección de nuevos movimientos. Fuente: los autores.



- Por último, si el usuario decide no repetir la serie de movimiento ni programar una nueva secuencia, se muestra en pantalla un aviso que indica que se puede apagar el sistema desenergizando lo desde el interruptor ubicado en la parte posterior. Después de lo cual se puede retirar la mano del paciente de manera segura.



Fig. 64 Mensaje apagar el dispositivo mostrado en pantalla. Fuente: los autores.



4.2 EVALUACIÓN DEL DISPOSITIVO

Con el objetivo de evaluar la efectividad del movimiento ejecutado por el sistema para realizar ejercicios terapéuticos en los dedos de la mano, se consultó a profesionales en fisioterapia con la finalidad de mostrarles la construcción, funcionamiento y objetivo del dispositivo construido, y de esta manera conocer una opinión experta y complementaria que valide el cumplimiento de los objetivos planteados.

Para este fin, se diseñó un cuestionario el cual fue diligenciado por los fisioterapeutas a los que se les mostro el dispositivo. En este cuestionario se hacen preguntas referentes a la utilidad de los movimientos del mecanismo diseñado, la viabilidad de la aplicación de este dispositivo en labores de fisioterapia, la facilidad de uso, seguridad y observaciones pertinentes a la finalidad de esta investigación. El diseño del cuestionario aplicado se muestra en la figura 65.

El primer cuestionario fue diligenciado por el fisioterapeuta Humberto Piscioti, quien resalto la posibilidad de que pacientes con estado de inmovilidad en la mitad del cuerpo causada por accidentes cardiovasculares, aprovechen el uso del prototipo ya que por la autonomía que brinda el sistema se puede implementar como herramienta de terapias en casa. La respuesta del cuestionario diligenciado a mano se muestra en la figura 66.

El segundo cuestionario fue respondido por la fisioterapeuta Sonia Carolina Mantilla Tolosa, docente de la universidad de pamplona, la cual valoro la economía, practicidad y limpieza con la cuenta el dispositivo debido al material usado para la construcción del prototipo. Estas respuestas se ven en las figuras 67 y 68.

De las respuestas obtenidas se resalta el hecho de que el sistema desarrollado demostró ser útil y viable en ejercicios de fisioterapia para los dedos de la mano. Los dos fisioterapeutas recomendaron la ampliación del sistema para realizar movimientos de flexo – extensión de manera simultánea en los cuatro dedos de la mano. De igual manera recomendaron continuar con el desarrollo de la investigación para poder llegar incluso a ejecutar movimientos de pinza que involucren al dedo pulgar.

Para trabajos futuros, se sugirió por parte de los fisioterapeutas aumentar la amplitud del movimiento que se usa para ejercitar la muñeca, con el fin de lograr otro tipo de movimiento en las terapias en las que puede ser utilizado el sistema.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

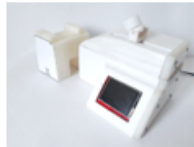


SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Universidad de pamplona- Facultad de Ingenierías y Arquitectura

Departamento MMI- Ingeniería Mecatrónica

ENCUESTA DE IMPACTO DEL PROYECTO APLICADA A PROFESIONALES EN FISIOTERAPIA



Responda las siguientes preguntas con respecto a la información recibida.

- Después de conocer el funcionamiento y propósito del prototipo ¿Considera que éste sistema cumple con la ejercitación de la mano? Si__ No__ ¿Por qué?

- Teniendo en cuenta los tipos de movimientos para los que fue diseñado el prototipo y con base en su experiencia ¿Considera que la implementación de éste sistema es viable para la realización de terapias para la mano? Si__ No__ ¿Por qué?

- En cuanto a la implementación del sistema como herramienta de trabajo ¿Considera que éste sistema es fácil de usar ? Si__ No__ ¿Por qué?

- Teniendo en cuenta la aplicación del sistema ¿cree que es seguro? __ No__ ¿Por qué?

- OBSERVACIONES:

Nombre _____

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortes.



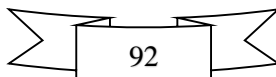
Fig. 65 Diseño cuestionario aplicado a fisioterapeutas. Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



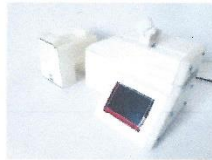


SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Universidad de pamplona- Facultad de Ingenierías y Arquitectura

Departamento MMI- Ingeniería Mecatrónica

ENCUESTA DE IMPACTO DEL PROYECTO APLICADA A PROFESIONALES EN FISIOTERAPIA



Responda las siguientes preguntas con respecto a la información recibida.

- Después de conocer el funcionamiento y propósito del prototipo ¿Considera que éste sistema es útil para la ejercitación de la mano? Si No ¿Por qué?
Se comprueba que en su aplicación (que) cumple con el objetivo de recuperación de movilidad articular.
- Teniendo en cuenta los tipos de movimientos para los que fue diseñado el prototipo y con base en su experiencia ¿Considera que la implementación de éste sistema es viable para la realización de terapias para la mano? Si No ¿Por qué?
El sistema está construido específicamente para el objeto en mención (flexión-extensión articular)
- En cuanto a la implementación del sistema como herramienta de trabajo ¿Considera que éste sistema es fácil de usar? Si No ¿Por qué?
Permite el trabajo autónomo y programado.
- Teniendo en cuenta la aplicación del sistema ¿cree que es seguro? No ¿Por qué?
El diseño del sistema ofrece garantías de seguridad.
- OBSERVACIONES:
Se considera que el sistema va a contribuir con el trabajo del fisioterapeuta, labor o en el trabajo de recuperación articular.

Humberto Piscioti Quintero

Nombre

Autor:
Ing.(c) John Edison Sánchez Fernández

Director:
PhD. César Augusto Peña Cortes.

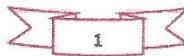
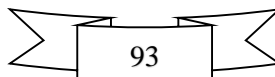


Fig. 66 Cuestionario diligenciado por el fisioterapeuta Humberto Piscioti. Fuente: los autores.

Autor:
Ing.(c) John Edison Sánchez Fernández

Director:
PhD. César Augusto Peña Cortés.



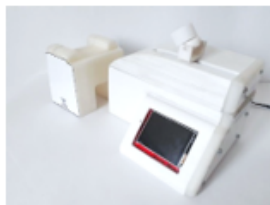


SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Universidad de pamplona- Facultad de Ingenierías y Arquitectura

Departamento MMI- Ingeniería Mecatrónica

ENCUESTA DE IMPACTO DEL PROYECTO APLICADA A PROFESIONALES EN FISIOTERAPIA



Responda las siguientes preguntas con respecto a la información recibida.

- Después de conocer el funcionamiento y propósito del prototipo ¿Considera que éste sistema cumple con la ejercitación de la mano? Si__ No_X_ ¿Por qué?
El sistema cumple con la ejercitación de los dedos índice, anular, corazón y meñique.
- Teniendo en cuenta los tipos de movimientos para los que fue diseñado el prototipo y con base en su experiencia ¿Considera que la implementación de éste sistema es viable para la realización de terapias para la mano? Si_X_ No__ ¿Por qué?
Insisto en que el sistema es viable para apoyar la ejecución de movimientos pasivos para los dedos de la mano, excepto el pulgar. No apoya el trabajo de movilización de la mano en general.
- En cuanto a la implementación del sistema como herramienta de trabajo ¿Considera que éste sistema es fácil de usar ? Si_X_ No__ ¿Por qué?
El mecanismo de funcionamiento es sencillo, fácil de ajustar, transportar y la pantalla permite visualizar y programar el funcionamiento del equipo.
- Teniendo en cuenta la aplicación del sistema ¿cree que es seguro? _X_ No__ ¿Por qué?
Está elaborado con material adecuado para su función, de fácil acceso, limpieza y sin ningún riesgo para la salud, integridad y vida de quien lo utiliza.
- OBSERVACIONES: El dispositivo diseñado presenta un sistema novedoso que puede complementar o apoyar la aplicación de ejercicio terapéutico durante la ejecución de fisioterapia, específicamente movilizaciones pasivas. Es un equipo de

Autor:
Ing.(c) John Edison Sánchez Fernández

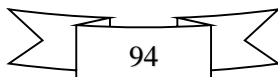
Director:
PhD. César Augusto Peña Cortes.



Fig. 67 Cuestionario diligenciado por la fisioterapeuta Sonia Mantilla (1). Fuente: los autores.

Autor:
Ing.(c) John Edison Sánchez Fernández

Director:
PhD. César Augusto Peña Cortés.



SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Departamento MMI- Ingeniería Mecatrónica



SISTEMA DE ASISTENCIA PARA LA REHABILITACIÓN O EJERCITACIÓN DE UNA MANO

Universidad de pamplona- Facultad de Ingenierías y Arquitectura

Departamento MMI- Ingeniería Mecatrónica

fácil acceso, económico, sencillo para activar su funcionamiento y también fácil de transportar y limpiar. Se sugiere cambiar en el título "rehabilitación" por "fisioterapia", ya que el término rehabilitación es muy general e involucra una gran cantidad de aspectos: el dispositivo realmente está diseñado para el apoyo de la aplicación de ejercicio terapéutico. Además, se sugiere cambiar "ejercitación de una mano" por "ejercitación de los dedos de la mano".

Sonia Carolina Mantilla Toloza

Nombre

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortes.



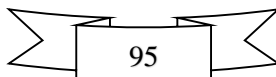
Fig. 68 Cuestionario diligenciado por la fisioterapeuta Sonia Mantilla (1). Fuente: los autores.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.





4.3 CONSTRUCCIÓN DEL DISPOSITIVO

En esta sección se muestran fotografías del prototipo construido mediante impresión 3D, en las cuales se evidencian las partes del mecanismo mencionadas y la ejecución de los movimientos de flexo - extensión para los que fue diseñado. En la figura 69 se muestra el ensamble total del sistema construido.

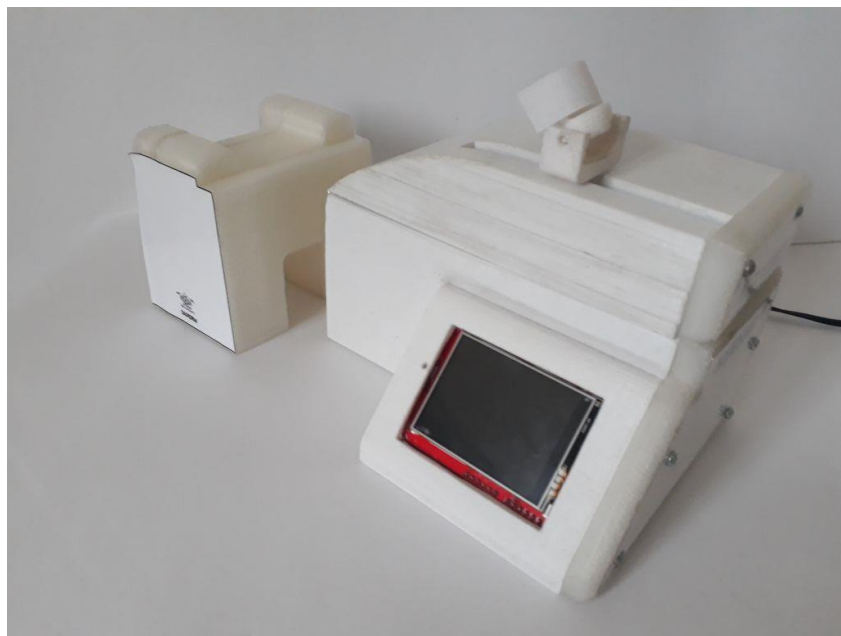


Fig. 69 Ensamble del dispositivo construido. Fuente: los autores.

En la figura 70 se muestra el dispositivo realizando el movimiento de extensión del dedo índice de una mano, en esta imagen también puedes observar el mensaje de “movimiento en proceso” mostrado por la pantalla LCD. El siguiente movimiento ejecutado por el sistema es el de flexión del dedo índice de la mano que se puede observar en la figura 71.



Fig. 70 Movimiento de extensión del dedo de la mano. Fuente: los autores.



Fig. 71 Movimiento de flexión del dedo de la mano. Fuente: los autores.



Fig. 72 Movimiento de flexión de la muñeca. Fuente: los autores.



Capítulo V. CONCLUSIONES.

5.1 CONCLUSIONES

- El sistema mecatrónico desarrollado en esta investigación, cumple con la finalidad de ejercitar articulaciones de la mano en personas en estado de inmovilidad. Se demostró que el prototipo es viable para la implementación como herramienta de trabajo de profesionales en fisioterapia y áreas afines. Y por los resultados obtenidos se puede ampliar para ejercitar la totalidad de la mano.
- Debido a la autonomía del sistema en la ejecución de los movimientos, este prototipo es óptimo para realizar terapias de ejercitación pasiva de tipo preventivo en personas en estado de inmovilidad parcial o total.
- Mediante el estudio de las características anatómicas de la mano, en cuanto a su composición y disposición, se logró diseñar un sistema mecatrónico que cumple con la función de transmitir movimientos para la ejercitación las articulaciones interfalángicas, distal, media y proximal de un dedo de la mano, además de ejercitar levemente la articulación de la muñeca.
- Con base en los ejercicios de rehabilitación motriz comúnmente utilizados por terapeutas especializados, y analizando los diferentes mecanismos utilizados para la transmisión de movimientos horizontales, se decidió implementar un mecanismo manivela biela corredera, debido a la posibilidad de desarrollar un prototipo compacto y portátil que pueda ser utilizado en diferentes ámbitos y por diferentes pacientes sin tener que establecer piezas personalizadas para cada posible paciente.
- La implementación de una pantalla táctil compatible con Arduino, permitió realizar una interfaz usuario - máquina, intuitiva y de operación sencilla. Que permite al operario controlar características de los movimientos que pueden ser adecuadas a las necesidades de cada paciente, y evaluar la evolución de la efectividad de los ejercicios realizados.

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



Capítulo VI. REFERENCIAS BIBLIOGRÁFICAS.

6.1 BIBLIOGRAFÍA

Amparo, L., López, A., Segundo, M., Maestría, S., & Departamento, H. (2012). A RTÍCULO ORIGINAL B iomecánica y patrones funcionales de la mano, 4(1), 14–24.

Asiain José, Sánchez Gema, Cueto Joxean, V. C. (n.d.). EL DAÑO CEREBRAL ADQUIRIDO, IMPRESCINDIBLE ACTUAR CUANTO ANTES. | ADACEN Noticias. Retrieved May 18, 2018, from <http://www.adacen.org/blog/index.php/la-importancia-de-la-rehabilitacion-en-dano-cerebral/>

Bouchet Alain, C. J. (1987). *Anatomía descriptiva, topográfica y funcional*. Panamericana.

Castaño, J. B., Cardona, J. E. M., Escobar, J. D. H., & Herrera, J. F. L. (2014). Shoulder flexion rehabilitation in patients with monoparesia using an exergame. *SeGAH 2014 - IEEE 3rd International Conference on Serious Games and Applications for Health, Books of Proceedings*. <https://doi.org/10.1109/SeGAH.2014.7067072>

Escolano, F. E. R. (2003). *Inteligencia artificial : modelos, técnicas y áreas de aplicación*. Thomson.

HandTutor - Telerehabilitación. (n.d.). Retrieved May 5, 2017, from <http://handtutor.com/>

Ictus – Causas y Tratamiento del Ictus. (2018). Retrieved May 18, 2018, from <https://www.neurorhb.com/ictus/>

infomed, red telemática de salud en cuba. (1999). medicina de rehabilitacion - ejercicios. Retrieved from <http://www.sld.cu/sitios/rehabilitacion-ejer/temas.php?idv=21360>

Joseph Edward Shigley. (2014). *Teoría de máquinas y mecanismos. Igarss 2014*. McGraw Hill. <https://doi.org/10.1007/s13398-014-0173-7.2>



- López, R., Aguilar, H., Salazar, S., Lozano, R., & Torres, J. A. (2014). Modelado y control de un exoesqueleto para la rehabilitación de extremidad inferior con dos grados de libertad. *RIAI - Revista Iberoamericana de Automatica E Informatica Industrial*, 11(3), 304–314.
<https://doi.org/10.1016/j.riai.2014.02.008>
- Morales, R., Badesa, F. J., Garcia-Aracil, N., Aranda, J., & Casals, A. (2015). Evaluación en un paciente con ictus en fase crónica de un sistema autoadaptativo de neurorehabilitación robótica. *RIAI - Revista Iberoamericana de Automatica E Informatica Industrial*, 12(1), 92–98.
<https://doi.org/10.1016/j.riai.2014.11.007>
- Netter, F. H. (2015). *Atlas de anatomía humana* (6th ed.). Elsevier Masson.
- PABLO. (n.d.). Retrieved May 5, 2017, from <http://tyromotion.com/en/products/pablo>
- Rehabilitación de la mano con Amadeo® | Red Menni. (n.d.). Retrieved May 5, 2017, from <http://dañocerebral.es/analisis-de-la-rehabilitacion-robotica-de-la-mano-con-el-robot-amadeo/>
- Rocon, E., Ruíz, A. F., Belda-Lois, J. M., Moreno, J. C., Pons, J. L., Raya, R., & Ceres, R. (2008). Diseño, Desarrollo y Validación de Dispositivo Robótico para la Supresión del Temblor Patológico. *Revista Iberoamericana de Automática E Informática Industrial RIAI*, 5(2), 79–92.
[https://doi.org/10.1016/S1697-7912\(08\)70147-X](https://doi.org/10.1016/S1697-7912(08)70147-X)
- Silva, F. a, Zarruk, J. G., Quintero, C., Arenas, W., & Silva, S. Y. (2006). Enfermedad cerebrovascular en Colombia. *Revista Colombiana de Cardiología*, 13(2), 85–89.
- Yang, Y., Ma, L., & Huang, D. (2016). Development and Repetitive Learning Control of Lower Limb Exoskeleton Driven by Electro-Hydraulic Actuators. *IEEE Transactions on Industrial Electronics*, 46(c), 1–1.
<https://doi.org/10.1109/TIE.2016.2622665>

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.

**Capítulo VII.
ANEXOS.**

7.1 PIEZAS IMPRESAS EN 3D.



Fig. 73 Manivela y biela del mecanismo manivela bielca corredera. Fuente: los autores.



Fig. 74 Corredera del mecanismo manivela bielca corredera. Fuente: los autores.



Fig. 75 Cascaron superior. Fuente: los autores.



Fig. 76 Cascaron derecho. Fuente: los autores.

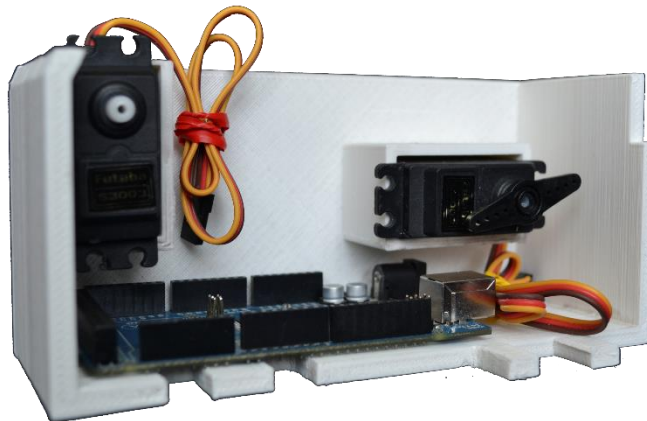


Fig. 77 Cascaron izquierdo con servomotores y tarjeta Arduino. Fuente: los autores



Fig. 78 Ensamble cascaron derecho, izquierdo y superior vista isométrica. Fuente: los autores.

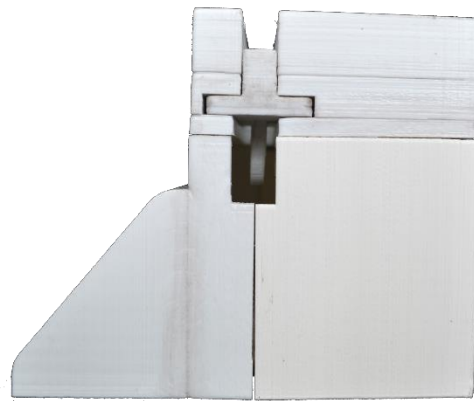


Fig. 79 Ensamble cascaron derecho, izquierdo y superior vista lateral. Fuente: los autores.

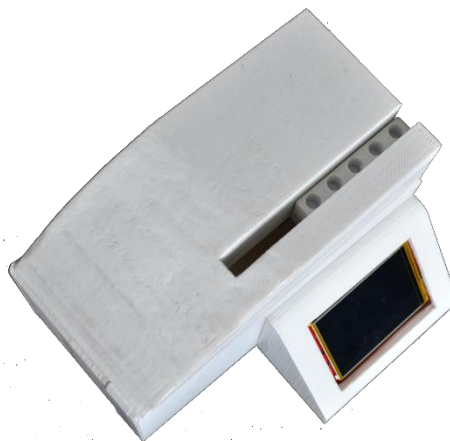


Fig. 80 Ensamble del sistema vista superior sin dedal. Fuente: los autores.

7.2 IMÁGENES Y MENÚS MOSTRADOS EN PANTALLA LCD.



Fig. 81 Imagen de presentación 1. Fuente: los autores.



Fig. 82 Imagen de presentación 2. Fuente: los autores.



Fig. 83 Imagen de presentación 3. Fuente: los autores.

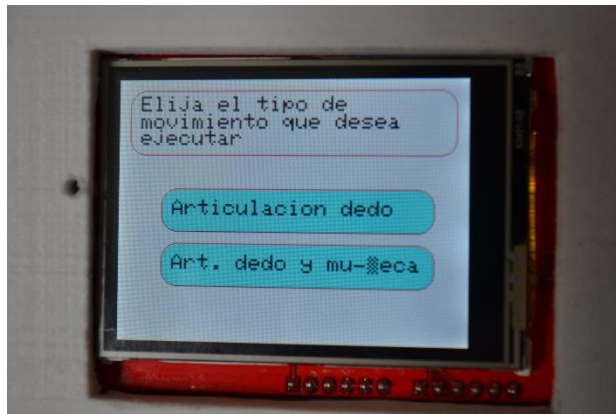


Fig. 84 Selección tipo de movimiento. Fuente: los autores.



Fig. 85 Selección tipo de fluidez. Fuente: los autores.

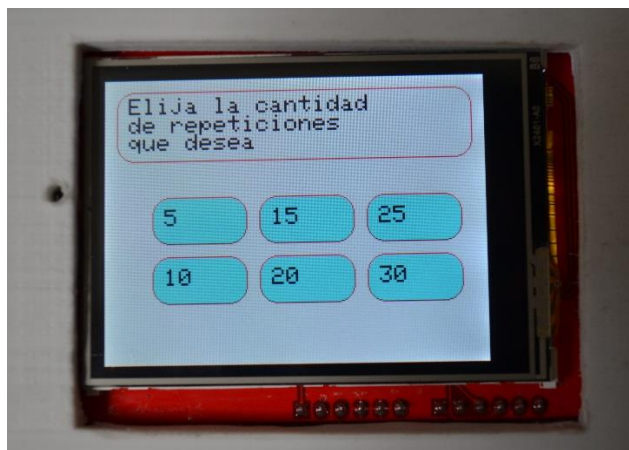


Fig. 86 Selección cantidad de repeticiones. Fuente: los autores.



Fig. 87 Movimiento en proceso. Fuente: los autores.



Fig. 88 Movimiento finalizado. Fuente: los autores.

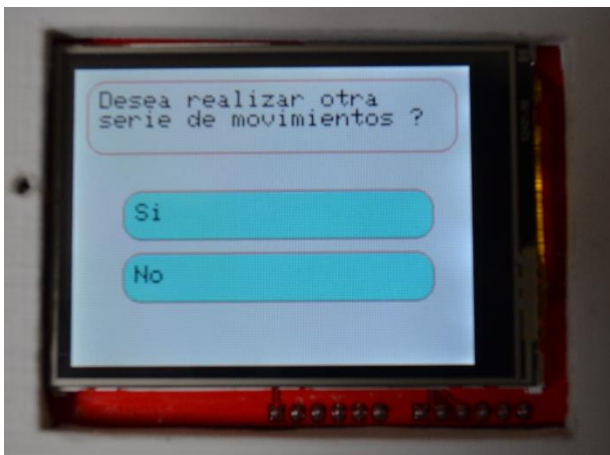


Fig. 89 Opción para repetir la serie de movimientos. Fuente: los autores.



Fig. 90 Mensaje para apagar el sistema al terminar de usarlo. Fuente: los autores.



7.3 CODIGO DE PROGRAMACIÓN.

```
// Programación Sistema de Asistencia para la Ejercitacion o Rehabilitacion de una Mano (SAPERM)
```

```
// Diseñado por ING(C)JOHN EDISON SANCHEZ FERNANDEZ
```

```
// Ing. Mecatronica Universidad de Pamplona
```

```
// Modified TFTbmp sketch from Adafruit_TFTLCD Library for
```

```
#include <Adafruit_GFX.h> // Libreria de graficos
```

```
#include <Adafruit_TFTLCD.h> // Libreria de LCD
```

```
#include <SD.h> // Libreria de tarjeta SD
```

```
#include <SPI.h> // Libreria bus SPI
```

```
#include <TouchScreen.h> // Libreria del panel tactil
```

```
#include <Servo.h>
```

```
Servo myservo1;
```

```
Servo myservo2;
```

```
Servo myservo3;
```

```
#define LCD_CS A3 // Definimos los pines del LCD
```

```
#define LCD_CD A2 // para poder visualizar elementos graficos
```

```
#define LCD_WR A1
```

```
#define LCD_RD A0
```

```
#define LCD_RESET A4
```

```
#define YP A3 // Pin analogico A1 para ADC
```

```
#define XM A2 // Pin analogico A2 para ADC
```

```
#define YM 9
```

```
#define XP 8
```

```
// Definimos la presion máxima y minima que podemos realizar sobre el panel
```

```
#define MINPRESSURE 1
```

```
#define MAXPRESSURE 1000
```

```
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 654); // 654 valor resistencia de la pantalla
```

```
short TS_MINX = 200; // Coordenadas del panel tactil para delimitar
```

```
short TS_MINY = 120; // el tamaño de la zona donde podemos presionar
```

```
short TS_MAXX = 850; // y que coincida con el tamaño del LCD
```

```
short TS_MAXY = 891;
```

```
// Los pines del puerto SPI vienen configurados por libreria, por lo que
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
// solamente debemos colocar el pin correspondiente al Chip Select del
// bus SPI correspondiente a la conexión con la tarjeta SD
#define SD_CS 53
```

```
// En la tarjeta SD debemos colocar imágenes en formato BMP de 24 Bits!
// Otro tipo de formato de imagen no se puede visualizar por pantalla.
```

```
Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET); //
Instancia LCD
```

```
#define BLACK 0x0000 // Definimos los colores para poder referirnos a ellos con su nombre
#define RED 0xF800 // en lugar de usar el código hexadecimal de cada uno.
#define GREEN 0x07E0
#define WHITE 0xFFFF
#define BLUE 0x001F
#define CYAN 0x07FF
#define YELLOW 0xFFE0
#define MAGENTA 0xF81F
```

```
int X; // Variables que almacenaran la coordenada
int Y; // X, Y donde presionemos y la variable Z
int Z; // almacenara la presión realizada
```

```
int a = 0; // variable "flag" para control rebotes
int sm= 0; // variable tipo de movimiento seleccionado
int sf= 0; // variable tipo de movimiento seleccionado
int sr= 0; // variable número repeticiones seleccionado
int inicio=0;
int i=0;
```

```
int mov=0;
int rep;
int pausa;
int volver=0;
int volver2=0;
int bandera;
int bandera1;
int ang, ang1;
```

```
int sad=22, saf=23, pos=0, conteo=0;
void setup()
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
{
  pinMode(saf, INPUT_PULLUP);
  pinMode(sad, INPUT_PULLUP);
  myservo1.attach(11);
  myservo2.attach(10);
  myservo1.write(0);
  myservo2.write(10);
  Serial.begin(9600); // Iniciamos el puerto serie para comprobar
                    // la comunicacion con la tarjeta microSD
  tft.reset();
  tft.begin(0x9341); // Iniciamos el LCD especificando el controlador ILI9341.

  Serial.print(F("Inicializando tarjeta SD..."));

  if (!SD.begin(SD_CS)) // Si se produce un error al intentar acceder
  {                      // a la tarjeta SD, lo mostramos por el Serial Monitor
    Serial.println(F("Error!"));
    return;
  }
  Serial.println(F("OK!"));

  tft.setRotation(3); // Establecemos la posicion de la pantalla Vertical

  bmpDraw("1.bmp", 0, 0); // Mostramos una imagen en las coordenadas 0,0

  delay(1000);

  tft.setRotation(3); // Establecemos la posicion de la pantalla Horizontal

  bmpDraw("2.bmp",0,0); // // Mostramos otra imagen en las coordenadas 0,0

  delay(1000);

  tft.setRotation(3); // Establecemos la posicion de la pantalla Horizontal

  bmpDraw("3.bmp",0,0); // // Mostramos otra imagen en las coordenadas 0,0
  //
  delay(2000);

}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
void loop()
{
while (i == 0)
{
    tft.fillScreen(WHITE);
    delay(500);
    tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y inicial
x final y final
    tft.setCursor(20,15); // Colocamos el cursor
    tft.setTextSize(2); // Especificamos el tamaño del texto
    tft.setTextColor(BLACK); // Definimos el color del texto
    tft.println("Elija el tipo de"); // Escribimos por pantalla
    tft.setCursor(20,31);
    tft.println("movimiento que desea ");
    tft.setCursor(20,46);
    tft.println("ejecutar");

    tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
    tft.drawRoundRect(40, 100, 250, 40, 15, RED);
    tft.setCursor(50,110);
    tft.println("Articulacion dedo");

    tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
    tft.drawRoundRect(40, 150, 250, 40, 15, RED);
    tft.setCursor(50,160);
    tft.println("Art. dedo y muneca");

    while(sm==0)
    {
        selecmov();
    }
    //////////////////////////////////////////////////// SELECCION FLUIDEZ DEL MOVIMIENTO
    if (sm!=0)
    {
        a=0;
        tft.fillScreen(WHITE); // Se pone la pantalla en blanco
        delay(500);
        tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y
inicial x final y final
        tft.setCursor(20,15); // Colocamos el cursor
```



```
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Elija el tipo de"); // Escribimos por pantalla
tft.setCursor(20,31);
tft.println("fluidez que desea ");
tft.setCursor(20,46);
tft.println("para el movimiento");

tft.fillRoundRect(40, 100, 250, 40, 15, CYAN); //Dibujamos un rectangulo con
relleno de color CYAN
tft.drawRoundRect(40, 100, 250, 40, 15, RED); // Dibijamos un rectangulo sin
relleno con border ROJOS
tft.setCursor(50,110);
tft.println("Pausada");

tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 250, 40, 15, RED);
tft.setCursor(50,160);
tft.println("Fluida");

while(sf==0)
{
    selecflu();
}

if (sf!=0)
{

tft.fillScreen(WHITE);// Ponemos la pantalla en blanco
delay(500);
tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y
inicial x final y final
tft.setCursor(20,15); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Elija la cantidad"); // Escribimos por pantalla
tft.setCursor(20,31);
tft.println("de repeticiones ");
tft.setCursor(20,46);
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
tft.println("que desea");

tft.fillRoundRect(40, 100, 80, 40, 15, CYAN);
tft.drawRoundRect(40, 100, 80, 40, 15, RED);
tft.setCursor(50,110);
tft.println("5");

tft.fillRoundRect(40, 150, 80, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 80, 40, 15, RED);
tft.setCursor(50,160);
tft.println("10");

tft.fillRoundRect(130, 100, 80, 40, 15, CYAN);
tft.drawRoundRect(130, 100, 80, 40, 15, RED);
tft.setCursor(140,110);
tft.println("15");

tft.fillRoundRect(130, 150, 80, 40, 15, CYAN);
tft.drawRoundRect(130, 150, 80, 40, 15, RED);
tft.setCursor(140,160);
tft.println("20");

tft.fillRoundRect(220, 100, 80, 40, 15, CYAN);
tft.drawRoundRect(220, 100, 80, 40, 15, RED);
tft.setCursor(230,110);
tft.println("25");

tft.fillRoundRect(220, 150, 80, 40, 15, CYAN);
tft.drawRoundRect(220, 150, 80, 40, 15, RED);
tft.setCursor(230,160);
tft.println("30");

while(sr==0)
{
    seleep();
}

if(sr!=0)
```



```
{
    // ciclo movimiento
    ciclomovimiento();
}

i=i+1;
}

}

// Esta funcion abre un archivo Windows bitmap (BMP) y lo muestra por
// pantalla en las coordenadas especificadas. Se puede acelerar el
// proceso de muestreo leyendo muchos pixeles a la vez en lugar de
// leer pixel a pixel, incrementando el tamaño de la siguiente variable
// BUFFPIXEL, utilizaremos mas memoria RAM del Arduino pero se realizará
// la carga de la imagen mas rapido.
// Un buffer de 20 pixeles es un valor equilibrado.

#define BUFFPIXEL 20

void bmpDraw(char *filename, int x, int y) {

    File  bmpFile;
    int   bmpWidth, bmpHeight; // W+H in pixels
    uint8_t bmpDepth;          // Bit depth (currently must be 24)
    uint32_t bmpImageoffset;   // Start of image data in file
    uint32_t rowSize;          // Not always = bmpWidth; may have padding
    uint8_t  sdbuffer[3*BUFFPIXEL]; // pixel in buffer (R+G+B per pixel)
    uint16_t lcdbuffer[BUFFPIXEL]; // pixel out buffer (16-bit per pixel)
    uint8_t  buffidx = sizeof(sdbuffer); // Current position in sdbuffer
    boolean  goodBmp = false;    // Set to true on valid header parse
    boolean  flip    = true;     // BMP is stored bottom-to-top
    int      w, h, row, col;
    uint8_t  r, g, b;
    uint32_t pos = 0, startTime = millis();
    uint8_t  lcdidx = 0;
    boolean  first = true;

    if((x >= tft.width()) || (y >= tft.height())) return;
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
Serial.println();
Serial.print(F("Loading image "));
Serial.print(filename);
Serial.println("\n");
// Open requested file on SD card
if ((bmpFile = SD.open(filename)) == NULL) {
  Serial.println(F("File not found"));
  return;
}

// Parse BMP header
if(read16(bmpFile) == 0x4D42) { // BMP signature
  Serial.println(F("File size: ")); Serial.println(read32(bmpFile));
  (void)read32(bmpFile); // Read & ignore creator bytes
  bmpImageoffset = read32(bmpFile); // Start of image data
  Serial.print(F("Image Offset: ")); Serial.println(bmpImageoffset, DEC);
  // Read DIB header
  Serial.print(F("Header size: ")); Serial.println(read32(bmpFile));
  bmpWidth = read32(bmpFile);
  bmpHeight = read32(bmpFile);
  if(read16(bmpFile) == 1) { // # planes -- must be '1'
    bmpDepth = read16(bmpFile); // bits per pixel
    Serial.print(F("Bit Depth: ")); Serial.println(bmpDepth);
    if((bmpDepth == 24) && (read32(bmpFile) == 0)) { // 0 = uncompressed

      goodBmp = true; // Supported BMP format -- proceed!
      Serial.print(F("Image size: "));
      Serial.print(bmpWidth);
      Serial.print('x');
      Serial.println(bmpHeight);

      // BMP rows are padded (if needed) to 4-byte boundary
      rowSize = (bmpWidth * 3 + 3) & ~3;

      // If bmpHeight is negative, image is in top-down order.
      // This is not canon but has been observed in the wild.
      if(bmpHeight < 0) {
        bmpHeight = -bmpHeight;
        flip = false;
      }
    }
  }
}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
// Crop area to be loaded
w = bmpWidth;
h = bmpHeight;
if((x+w-1) >= tft.width()) w = tft.width() - x;
if((y+h-1) >= tft.height()) h = tft.height() - y;

// Set TFT address window to clipped image bounds
tft.setAddrWindow(x, y, x+w-1, y+h-1);

for (row=0; row<h; row++) { // For each scanline...
  // Seek to start of scan line. It might seem labor-
  // intensive to be doing this on every line, but this
  // method covers a lot of gritty details like cropping
  // and scanline padding. Also, the seek only takes
  // place if the file position actually needs to change
  // (avoids a lot of cluster math in SD library).
  if(flip) // Bitmap is stored bottom-to-top order (normal BMP)
    pos = bmpImageoffset + (bmpHeight - 1 - row) * rowSize;
  else // Bitmap is stored top-to-bottom
    pos = bmpImageoffset + row * rowSize;
  if(bmpFile.position() != pos) { // Need seek?
    bmpFile.seek(pos);
    buffidx = sizeof(sdbuffer); // Force buffer reload
  }

  for (col=0; col<w; col++) { // For each column...
    // Time to read more pixel data?
    if (buffidx >= sizeof(sdbuffer)) { // Indeed
      // Push LCD buffer to the display first
      if(lcdidx > 0) {
        tft.pushColors(lcdbuffer, lcdidx, first);
        lcdidx = 0;
        first = false;
      }
      bmpFile.read(sdbuffer, sizeof(sdbuffer));
      buffidx = 0; // Set index to beginning
    }

    // Convert pixel from BMP to TFT format
    b = sdbuffer[buffidx++];
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
        g = sdbuffer[bufidx++];
        r = sdbuffer[bufidx++];
        lcdbuffer[lcdidx++] = tft.color565(r,g,b);
    } // end pixel
} // end scanline
// Write any remaining data to LCD
if(lcdidx > 0) {
    tft.pushColors(lcdbuffer, lcdidx, first);
}
Serial.print(F("Loaded in "));
Serial.print(millis() - startTime);
Serial.println(" ms");
} // end goodBmp
}
}

bmpFile.close();
if(!goodBmp) Serial.println(F("BMP format not recognized."));
}

// These read 16- and 32-bit types from the SD card file.
// BMP data is stored little-endian, Arduino is little-endian too.
// May need to reverse subscript order if porting elsewhere.

uint16_t read16(File f) {
    uint16_t result;
    ((uint8_t *)&result)[0] = f.read(); // LSB
    ((uint8_t *)&result)[1] = f.read(); // MSB
    return result;
}

uint32_t read32(File f) {
    uint32_t result;
    ((uint8_t *)&result)[0] = f.read(); // LSB
    ((uint8_t *)&result)[1] = f.read();
    ((uint8_t *)&result)[2] = f.read();
    ((uint8_t *)&result)[3] = f.read(); // MSB
    return result;
}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
// FUNCION LECTURA DEL TACTIL DE LA PANTALLA HORIZONTAL
```

```
void lecturaPanel()
{
    digitalWrite(13, HIGH);
    TSPoint p = ts.getPoint(); // Realizamos lectura de las coordenadas
    digitalWrite(13, LOW);

    pinMode(XM, OUTPUT); // La librería utiliza estos pines como entrada y salida
    pinMode(YP, OUTPUT); // por lo que es necesario declararlos como salida justo
    // despues de realizar una lectura de coordenadas.

    // Mapeamos los valores analogicos leidos del panel tactil (0-1023)
    // y los convertimos en valor correspondiente a la medida del LCD 320x240
    X= map(p.y, TS_MAXX, TS_MINX, tft.width(), 0);
    Y = map(p.x, TS_MINY, TS_MAXY, tft.height(), 0);
    Z = p.z;
}

void selecmov()
{
    lecturaPanel(); // Realizamos lectura del panel para detectar presion y coordenadas

    // Si la pulsación del eje X se produce entre los puntos 40 y 100
    // Y la pulsacion del eje Y se produce entre los puntos 20 y 60
    // Y la presión realizada esta entre el margen determinado
    if((X > 40 && X < 250) && (Y > 100 && Y < 140) && (Z > MINPRESSURE && Z <
MAXPRESSURE))
    {
        if (a == 0) // Si la variable flag esta en 0
        {
            tft.fillRoundRect(40, 100, 250, 40, 15, GREEN);
            tft.drawRoundRect(40, 100, 250, 40, 15, RED);
            tft.setCursor(50,110);
            tft.println("Articulacion dedo");
            sm=1;
            a = 1; // Ponemos la variable flag en 1
            delay(150);
        }
        else if (a == 1) // Si la variable flag esta en 1
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
{
  tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
  tft.drawRoundRect(40, 100, 250, 40, 15, RED);
  tft.setCursor(50,110);
  tft.println("Articulacion dedo");
  sm=1;
  a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
  delay(150);
}
}
else if((X > 40 && X < 250) && (Y > 150 && Y < 190) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{
  if (a == 0) // Si la variable flag esta en 0
  {
    tft.fillRoundRect(40, 150, 250, 40, 15, GREEN);
    tft.drawRoundRect(40, 150, 250, 40, 15, RED);
    tft.setCursor(50,160);
    tft.println("Art. dedo y muneca");
    sm=2;
    a = 1; // Ponemos la variable flag en 1
    delay(150);
  }
  else if (a == 1) // Si la variable flag esta en 1
  {
    tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
    tft.drawRoundRect(40, 150, 250, 40, 15, RED);
    tft.setCursor(50,160);
    tft.println("Art. dedo y muneca");
    sm=2;
    a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
    delay(150);
  }
}
}
a=0;
return sm;
}

void selecflu()
{
```



```
lecturaPanel(); // Realizamos lectura del panel para detectar presion y coordenadas
  if((X > 40 && X < 250) && (Y > 100 && Y < 140) && (Z > MINPRESSURE && Z <
MAXPRESSURE))
  {
    if (a == 0) // Si la variable flag esta en 0
    {
      tft.fillRoundRect(40, 100, 250, 40, 15, GREEN);
      tft.drawRoundRect(40, 100, 250, 40, 15, RED);
      tft.setCursor(50,110);
      tft.println("Pausada");
      //digitalWrite(led, HIGH);
      sf=1;
      a = 1; // Ponemos la variable flag en 1
      delay(150);
    }
    else if (a == 1) // Si la variable flag esta en 1
    {
      tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
      tft.drawRoundRect(40, 100, 250, 40, 15, RED);
      tft.setCursor(50,110);
      tft.println("Pausada");
      sf=1;
      a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
      delay(150);
    }
  }
  else if((X > 40 && X < 250) && (Y > 150 && Y < 190) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
  {
    if (a == 0) // Si la variable flag esta en 0
    {
      tft.fillRoundRect(40, 150, 250, 40, 15, GREEN);
      tft.drawRoundRect(40, 150, 250, 40, 15, RED);
      tft.setCursor(50,160);
      tft.println("Fluida");
      //digitalWrite(led, HIGH);
      sf=2;
      a = 1; // Ponemos la variable flag en 1
      delay(150);
    }
    else if (a == 1) // Si la variable flag esta en 1
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
{
  tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
  tft.drawRoundRect(40, 150, 250, 40, 15, RED);
  tft.setCursor(50,160);
  tft.println("Fluida");
  sf=2;
  a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
  delay(150);
}
}
a=0;
return sf;
}

void selecrep()
{
  lecturaPanel(); // Realizamos lectura del panel para detectar presion y coordenadas

  if((X > 40 && X <80 )&& (Y > 100 && Y < 140) && (Z > MINPRESSURE && Z <
MAXPRESSURE))
  {
    if (a == 0) // Si la variable flag esta en 0
    {
      tft.fillRoundRect(40, 100, 80, 40, 15, GREEN);
      tft.drawRoundRect(40, 100, 80, 40, 15, RED);
      tft.setCursor(50,110);
      tft.println("5");
      sr=5;
      a = 1; // Ponemos la variable flag en 1
      delay(150);
    }
    else if (a == 1) // Si la variable flag esta en 1
    {
      tft.fillRoundRect(40, 100, 80, 40, 15, CYAN);
      tft.drawRoundRect(40, 100, 80, 40, 15, RED);
      tft.setCursor(50,110);
      tft.println("5");
      sr=5;
      a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
      delay(150);
    }
  }
}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
}
else if((X > 40 && X < 80) && (Y > 150 && Y < 190) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{
  if (a == 0) // Si la variable flag esta en 0
  {
    tft.fillRoundRect(40, 150, 80, 40, 15, GREEN);
    tft.drawRoundRect(40, 150, 80, 40, 15, RED);
    tft.setCursor(50,160);
    tft.println("10");
    //digitalWrite(led, HIGH);
    sr=10;
    a = 1; // Ponemos la variable flag en 1
    delay(150);
  }
  else if (a == 1) // Si la variable flag esta en 1
  {
    tft.fillRoundRect(40, 150, 80, 40, 15, CYAN);
    tft.drawRoundRect(40, 150, 80, 40, 15, RED);
    tft.setCursor(50,160);
    tft.println("10");
    //digitalWrite(led, LOW);
    sr=10;
    a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
    delay(150);
  }
}
else if((X > 130 && X < 210) && (Y > 100 && Y < 140) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{
  if (a == 0) // Si la variable flag esta en 0
  {
    tft.fillRoundRect(130, 100, 80, 40, 15, GREEN);
    tft.drawRoundRect(130, 100, 80, 40, 15, RED);
    tft.setCursor(140,110);
    tft.println("15");
    //digitalWrite(led, HIGH);
    sr=15;
    a = 1; // Ponemos la variable flag en 1
    delay(150);
  }
}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
else if (a == 1) // Si la variable flag esta en 1
{
  tft.fillRoundRect(130, 100, 80, 40, 15, CYAN);
  tft.drawRoundRect(130, 100, 80, 40, 15, RED);
  tft.setCursor(140,110);
  tft.println("15");
  sr=15;
  a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
  delay(150);
}
}

else if((X > 130 && X < 210) && (Y > 150 && Y < 190) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{
  if (a == 0) // Si la variable flag esta en 0
  {
    tft.fillRoundRect(130, 150, 80, 40, 15, GREEN);
    tft.drawRoundRect(130, 150, 80, 40, 15, RED);
    tft.setCursor(140,160);
    tft.println("20");
    sr=20;
    a = 1; // Ponemos la variable flag en 1
    delay(150);
  }
  else if (a == 1) // Si la variable flag esta en 1
  {
    tft.fillRoundRect(130, 150, 80, 40, 15, CYAN);
    tft.drawRoundRect(130, 150, 80, 40, 15, RED);
    tft.setCursor(140,160);
    tft.println("20");
    sr=20;
    a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
    delay(150);
  }
}

else if((X > 220 && X < 300) && (Y > 100 && Y < 140) && (Z > MINPRESSURE &&
Z < MAXPRESSURE))
{
  if (a == 0) // Si la variable flag esta en 0
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
{
  tft.fillRoundRect(220, 100, 80, 40, 15, GREEN);
  tft.drawRoundRect(220, 100, 80, 40, 15, RED);
  tft.setCursor(230,110);
  tft.println("25");
  sr=25;
  a = 1; // Ponemos la variable flag en 1
  delay(150);
}
else if (a == 1) // Si la variable flag esta en 1
{
  tft.fillRoundRect(220, 100, 80, 40, 15, CYAN);
  tft.drawRoundRect(220, 100, 80, 40, 15, RED);
  tft.setCursor(230,110);
  tft.println("25");
  sr=25;
  a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
  delay(150);
}
}

else if((X > 220 && X < 300) && (Y > 150 && Y < 190) && (Z > MINPRESSURE
&& Z < MAXPRESSURE))
{
  if (a == 0) // Si la variable flag esta en 0
  {
    tft.fillRoundRect(220, 150, 80, 40, 15, GREEN);
    tft.drawRoundRect(220, 150, 80, 40, 15, RED);
    tft.setCursor(230,160);
    tft.println("30");
    sr=30;
    a = 1; // Ponemos la variable flag en 1
    delay(150);
  }
  else if (a == 1) // Si la variable flag esta en 1
  {
    tft.fillRoundRect(220, 150, 80, 40, 15, CYAN);
    tft.drawRoundRect(220, 150, 80, 40, 15, RED);
    tft.setCursor(230,160);
    tft.println("30");
    sr=30;
  }
}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
        a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
        delay(150);
    }
}
return sr;

}

void movimiento(int sm, int sf, int sr)
{
    //////////////////////////////////////
    mov=0;
    myservo1.write(0);
    if (sm==1)
    {
        if(sf==1)
        {
            pausa=500;
            for(rep = 1; rep <= sr; rep ++)
            {
                if(digitalRead(sad)==0)
                {
                    //MOVER ADELANTE
                    while(digitalRead(saf)==1)
                    {
                        Serial.println("MOVIENDO HACIA ADELANTE");
                        Serial.println("SAF");
                        Serial.println(digitalRead(saf));
                        myservo1.write(pos);
                        pos=pos+3;
                        delay(pausa);
                    }
                }
                Serial.println("SAF");

                Serial.println(digitalRead(saf));
                Serial.println("SAD");

                Serial.println(digitalRead(saf));

            }
            if(digitalRead(saf)==0)
```



```
{
  //MOVER ATRAS
  while(digitalRead(sad)==1)
  {
    Serial.println("MOVIENDO HACIA ATRAS");
    Serial.println("SAD");
    Serial.println(digitalRead(sad));
    myservo1.write(pos);
    pos=pos-3;
    delay(pausa);
  }
  Serial.println("CONTEO");
  Serial.println(rep);
}

tft.fillCircle(130, 215, 5, WHITE);
tft.drawCircle(130, 215, 8, WHITE);
tft.fillCircle(160, 215, 5, WHITE);
tft.drawCircle(160, 215, 8, WHITE);
tft.fillCircle(190, 215, 5, WHITE);
tft.drawCircle(190, 215, 8, WHITE);
delay(100);
tft.fillCircle(130, 215, 5, RED);
tft.drawCircle(130, 215, 8, BLACK);
delay(100);
tft.fillCircle(160, 215, 5, RED);
tft.drawCircle(160, 215, 8, BLACK);
delay(100);
tft.fillCircle(190, 215, 5, RED);
tft.drawCircle(190, 215, 8, BLACK);

}
}
if(sf==2)
{
  pausa=100;
  for(rep = 1; rep <= sr; rep ++)
  {
    if(digitalRead(sad)==0)
    {
      //MOVER ADELANTE
      while(digitalRead(saf)==1)
```



```
{
  Serial.println("MOVIENDO HACIA ADELANTE");
  Serial.println("SAD");
  Serial.println(digitalRead(saf));
  myservo1.write(pos);
  pos=pos+2;
  delay(pausa);
}
}

if(digitalRead(saf)==0)
{
  //MOVER ATRAS
  while(digitalRead(sad)==1)
  {
    Serial.println("MOVIENDO HACIA ATRAS");
    Serial.println("SAF");
    Serial.println(digitalRead(sad));
    myservo1.write(pos);
    pos=pos-2;
    delay(pausa);
  }
  Serial.println("CONTEO");
  Serial.println(rep);
}

tft.fillCircle(130, 215, 5, WHITE);
tft.drawCircle(130, 215, 8, WHITE);
tft.fillCircle(160, 215, 5, WHITE);
tft.drawCircle(160, 215, 8, WHITE);
tft.fillCircle(190, 215, 5, WHITE);
tft.drawCircle(190, 215, 8, WHITE);
delay(100);
tft.fillCircle(130, 215, 5, RED);
tft.drawCircle(130, 215, 8, BLACK);
delay(100);
tft.fillCircle(160, 215, 5, RED);
tft.drawCircle(160, 215, 8, BLACK);
delay(100);
tft.fillCircle(190, 215, 5, RED);
tft.drawCircle(190, 215, 8, BLACK);
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
    }
  }
}
else
{
  if (sm==2)
  {
    if(sf==1)
    {
      pausa=500;
      for(rep = 1; rep <= sr; rep ++ )
      {
        if(digitalRead(sad)==0)
        {
          //MOVER ADELANTE
          while(digitalRead(saf)==1)
          {
            Serial.println("MOVIENDO HACIA ADELANTE");
            Serial.println("SAD");
            Serial.println(digitalRead(saf));
            myservo1.write(pos);
            pos=pos+3;
            delay(pausa);
          }
        }
      }

      if(digitalRead(saf)==0)
      {
        //MOVER ATRAS
        while(digitalRead(sad)==1)
        {
          Serial.println("MOVIENDO HACIA ATRAS");
          Serial.println("SAF");
          Serial.println(digitalRead(sad));
          myservo1.write(pos);
          pos=pos-3;
          delay(pausa);
        }
        Serial.println("CONTEO");
        Serial.println(rep);
      }
    }
  }
}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
delay(1000);
  for(ang = 10; ang <= 45; ang=ang+5)
  {
    myservo2.write(ang);
    Serial.println("MOVIENDO MUNECA");
    delay(pausa);
  }
delay(2000);
  for(ang1 = 45; ang1 >= 10; ang1=ang1-5)
  {
    myservo2.write(ang1);
    Serial.println("MOVIENDO MUNECA");
    delay(pausa);
  }
```

```
tft.fillCircle(130, 215, 5, WHITE);
tft.drawCircle(130, 215, 8, WHITE);
tft.fillCircle(160, 215, 5, WHITE);
tft.drawCircle(160, 215, 8, WHITE);
tft.fillCircle(190, 215, 5, WHITE);
tft.drawCircle(190, 215, 8, WHITE);
delay(100);
tft.fillCircle(130, 215, 5, RED);
tft.drawCircle(130, 215, 8, BLACK);
delay(100);
tft.fillCircle(160, 215, 5, RED);
tft.drawCircle(160, 215, 8, BLACK);
delay(100);
tft.fillCircle(190, 215, 5, RED);
tft.drawCircle(190, 215, 8, BLACK);
```

```
  }
}
if(sf==2)
{
  pausa=100;
  for(rep = 1; rep <= sr; rep ++ )
  {
    if(digitalRead(sad)==0)
    {
```




```
//MOVER ADELANTE
while(digitalRead(saf)==1)
{
  Serial.println("MOVIENDO HACIA ADELANTE");
  Serial.println("SAD");
  Serial.println(digitalRead(saf));
  myservo1.write(pos);
  pos=pos+2;
  delay(pausa);
}
}

if(digitalRead(saf)==0)
{
  //MOVER ATRAS
  while(digitalRead(sad)==1)
  {
    Serial.println("MOVIENDO HACIA ATRAS");
    Serial.println("SAF");
    Serial.println(digitalRead(sad));
    myservo1.write(pos);
    pos=pos-2;
    delay(pausa);
  }
  Serial.println("CONTEO");
  Serial.println(rep);
}

delay(1000);
for(ang = 10; ang <= 45; ang=ang+5)
{
  myservo2.write(ang);
  Serial.println("MOVIENDO MUNECA");
  delay(pausa);
}
delay(2000);
for(ang1 = 45; ang1 >= 10; ang1=ang1-5)
{
  myservo2.write(ang1);
  Serial.println("MOVIENDO MUNECA");
  delay(pausa);
}
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
    }
    tft.fillCircle(130, 215, 5, WHITE);
    tft.drawCircle(130, 215, 8, WHITE);
    tft.fillCircle(160, 215, 5, WHITE);
    tft.drawCircle(160, 215, 8, WHITE);
    tft.fillCircle(190, 215, 5, WHITE);
    tft.drawCircle(190, 215, 8, WHITE);
    delay(100);
    tft.fillCircle(130, 215, 5, RED);
    tft.drawCircle(130, 215, 8, BLACK);
    delay(100);
    tft.fillCircle(160, 215, 5, RED);
    tft.drawCircle(160, 215, 8, BLACK);
    delay(100);
    tft.fillCircle(190, 215, 5, RED);
    tft.drawCircle(190, 215, 8, BLACK);
}
}
}
}
    mov=1;
    return mov;
}

void volver()
{

    lecturaPanel();

    if((X > 40 && X < 250) && (Y > 100 && Y < 140) && (Z > MINPRESSURE && Z
< MAXPRESSURE))
    {
        if (a == 0) // Si la variable flag esta en 0
        {
            tft.fillRoundRect(40, 100, 250, 40, 15, GREEN);
            tft.drawRoundRect(40, 100, 250, 40, 15, RED);
            tft.setCursor(50,110);
            tft.println("Si");
            volver=2;
            a = 1; // Ponemos la variable flag en 1
```



```
        delay(150);
    }
    else if (a == 1) // Si la variable flag esta en 1
    {
        tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
        tft.drawRoundRect(40, 100, 250, 40, 15, RED);
        tft.setCursor(50,110);
        tft.println("Si");
        volver=2;
        a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
        delay(150);
    }
}
else if((X > 40 && X < 250) && (Y > 150 && Y < 190) && (Z > MINPRESSURE
&& Z < MAXPRESSURE))
{
    if (a == 0) // Si la variable flag esta en 0
    {
        tft.fillRoundRect(40, 150, 250, 40, 15, GREEN);
        tft.drawRoundRect(40, 150, 250, 40, 15, RED);
        tft.setCursor(50,160);
        tft.println("No");
        volver=1;
        a = 1; // Ponemos la variable flag en 1
        delay(150);
    }
    else if (a == 1) // Si la variable flag esta en 1
    {
        tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
        tft.drawRoundRect(40, 150, 250, 40, 15, RED);
        tft.setCursor(50,160);
        tft.println("No");
        volver=1;
    }

    a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
    delay(150);
}
}
return volver;
}
```



```
void volver2()
{
    lecturaPanel();

    if((X > 40 && X < 250) && (Y > 100 && Y < 140) && (Z > MINPRESSURE && Z
< MAXPRESSURE))
    {
        if (a == 0) // Si la variable flag esta en 0
        {
            tft.fillRoundRect(40, 100, 250, 40, 15, GREEN);
            tft.drawRoundRect(40, 100, 250, 40, 15, RED);
            tft.setCursor(50,110);
            tft.println("Si");
            volver2=2;
            a = 1; // Ponemos la variable flag en 1
            delay(150);
        }
        else if (a == 1) // Si la variable flag esta en 1
        {
            tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
            tft.drawRoundRect(40, 100, 250, 40, 15, RED);
            tft.setCursor(50,110);
            tft.println("Si");
            volver2=2;
            a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
            delay(150);
        }
    }
    else if((X > 40 && X < 250) && (Y > 150 && Y < 190) && (Z > MINPRESSURE
&& Z < MAXPRESSURE))
    {
        if (a == 0) // Si la variable flag esta en 0
        {
            tft.fillRoundRect(40, 150, 250, 40, 15, GREEN);
            tft.drawRoundRect(40, 150, 250, 40, 15, RED);
            tft.setCursor(50,160);
            tft.println("No");
            volver2=1;
            a = 1; // Ponemos la variable flag en 1
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
        delay(150);
    }
    else if (a == 1) // Si la variable flag esta en 1
    {
        tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
        tft.drawRoundRect(40, 150, 250, 40, 15, RED);
        tft.setCursor(50,160);
        tft.println("No");
        volver2=1;

        a = 0; // Ponemos la variable flag en 0 para evitar los rebotes
        delay(150);
    }
    }
    return volver2;
}

void seleccion2()
{
    sm=0;
    sf=0;
    sr=0;

    tft.fillScreen(WHITE);
    delay(500);
    tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicio y inicio
x final y final
    tft.setCursor(20,15); // Colocamos el cursor
    tft.setTextSize(2); // Especificamos el tamaño del texto
    tft.setTextColor(BLACK); // Definimos el color del texto
    tft.println("Elija el tipo de"); // Escribimos por pantalla
    tft.setCursor(20,31);
    tft.println("movimiento que desea ");
    tft.setCursor(20,46);
    tft.println("ejecutar");

    tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
    tft.drawRoundRect(40, 100, 250, 40, 15, RED);
    tft.setCursor(50,110);
    tft.println("Articulacion dedo");
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);  
tft.drawRoundRect(40, 150, 250, 40, 15, RED);  
tft.setCursor(50,160);  
tft.println("Art. dedo y muñeca");
```

```
while(sm==0)  
{  
  selecmov();  
}  
  ////////////////////////////////////////////////// SELECCION FLUIDEZ DEL MOVIMIENTO  
if (sm!=0)  
{  
  a=0;  
  tft.fillScreen(WHITE);  
  delay(500);  
  tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicio y  
inicio x final y final  
  tft.setCursor(20,15); // Colocamos el cursor  
  tft.setTextSize(2); // Especificamos el tamaño del texto  
  tft.setTextColor(BLACK); // Definimos el color del texto  
  tft.println("Elija el tipo de"); // Escribimos por pantalla  
  tft.setCursor(20,31);  
  tft.println("fluidez que desea ");  
  tft.setCursor(20,46);  
  tft.println("para el movimiento");  
  
  tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);  
  tft.drawRoundRect(40, 100, 250, 40, 15, RED);  
  tft.setCursor(50,110);  
  tft.println("Pausada");  
  
  tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);  
  tft.drawRoundRect(40, 150, 250, 40, 15, RED);  
  tft.setCursor(50,160);  
  tft.println("Fluida");  
  
  while(sf==0)  
  {  
    selecflu();
```



```
    }  
  }  
  
  if (sf!=0)  
  {  
  
    tft.fillScreen(WHITE);  
    delay(500);  
    tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicio y  
inicio x final y final  
    tft.setCursor(20,15); // Colocamos el cursor  
    tft.setTextSize(2); // Especificamos el tamaño del texto  
    tft.setTextColor(BLACK); // Definimos el color del texto  
    tft.println("Elija la cantidad"); // Escribimos por pantalla  
    tft.setCursor(20,31);  
    tft.println("de repeticiones ");  
    tft.setCursor(20,46);  
    tft.println("que desea");  
  
    tft.fillRoundRect(40, 100, 80, 40, 15, CYAN);  
    tft.drawRoundRect(40, 100, 80, 40, 15, RED);  
    tft.setCursor(50,110);  
    tft.println("5");  
  
    tft.fillRoundRect(40, 150, 80, 40, 15, CYAN);  
    tft.drawRoundRect(40, 150, 80, 40, 15, RED);  
    tft.setCursor(50,160);  
    tft.println("10");  
  
    tft.fillRoundRect(130, 100, 80, 40, 15, CYAN);  
    tft.drawRoundRect(130, 100, 80, 40, 15, RED);  
    tft.setCursor(140,110);  
    tft.println("15");  
  
    tft.fillRoundRect(130, 150, 80, 40, 15, CYAN);  
    tft.drawRoundRect(130, 150, 80, 40, 15, RED);  
    tft.setCursor(140,160);  
    tft.println("20");  
  
    tft.fillRoundRect(220, 100, 80, 40, 15, CYAN);  
    tft.drawRoundRect(220, 100, 80, 40, 15, RED);
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
tft.setCursor(230,110);
tft.println("25");

tft.fillRoundRect(220, 150, 80, 40, 15, CYAN);
tft.drawRoundRect(220, 150, 80, 40, 15, RED);
tft.setCursor(230,160);
tft.println("30");

while(sr==0)
{
  seleep();
}

if(sr!=0)
{
  // ciclo movimiento
  ciclomovimiento();
}

void ciclomovimiento()
{
  volver=0;
  volver2=0;
  while(volver==0 || volver==2 || volver2==0 || volver2==2)
  {
    ////////////PRUEBA
    tft.fillScreen(WHITE);
    delay(500);
    tft.fillRoundRect(40, 40, 240, 150, 15, YELLOW); // Dibujamos un "boton" x inicio
y inicio x final y final
    tft.drawRoundRect(35, 35, 250, 160, 15, BLUE);
    tft.setTextSize(3); // Especificamos el tamaño del texto
    tft.setTextColor(BLACK); // Definimos el color del texto
    tft.setCursor(75,75);
    tft.println("Movimiento "); // Escribimos por pantalla
    tft.setCursor(130,105);
    tft.println("en"); // Escribimos por pantalla
```

Autor:

Ing.(c) John Edison Sánchez Fernández

Director:

PhD. César Augusto Peña Cortés.



```
tft.setCursor(90,135);
tft.println("proceso"); // Escribimos por pantalla
delay(100);
tft.fillCircle(130, 215, 5, RED);
tft.drawCircle(130, 215, 8, BLACK);
delay(100);
tft.fillCircle(160, 215, 5, RED);
tft.drawCircle(160, 215, 8, BLACK);
delay(100);
tft.fillCircle(190, 215, 5, RED);
tft.drawCircle(190, 215, 8, BLACK);
//////////
volver=0;
volver2=0;
//MOVIMIENTO
movimiento(sm, sf, sr);

if (mov==1)
{
tft.fillScreen(WHITE);
delay(500);
tft.fillRoundRect(40, 40, 240, 150, 15, GREEN); // Dibujamos un "boton" x inicio y
inicio x final y final
tft.drawRoundRect(35, 35, 250, 160, 15, BLUE);
tft.setTextSize(3); // Especificamos el tamaño del texto
tft.setTextColor(BLACK); // Definimos el color del texto
tft.setCursor(75,85);
tft.println("Movimiento "); // Escribimos por pantalla
tft.setCursor(70,115);
tft.println("Finalizado"); // Escribimos por pantalla
delay(3000);
}

//CAMBIO DE PANTALLA

tft.fillScreen(WHITE);
delay(500);
tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicial y
inicial x final y final
tft.setCursor(20,18); // Colocamos el cursor
tft.setTextSize(2); // Especificamos el tamaño del texto
```



```
tft.setTextColor(BLACK); // Definimos el color del texto
tft.println("Desea repetir la "); // Escribimos por pantalla
tft.setCursor(20,34);
tft.println("serie de movimientos ? ");

tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 100, 250, 40, 15, RED);
tft.setCursor(50,110);
tft.println("Si");

tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 250, 40, 15, RED);
tft.setCursor(50,160);
tft.println("No");

while(volver==0)
{
  volver();
}
bandera=0;
while(volver==1)
{
  if(bandera==0)
  {
    //PREGUNTAR LAS OPCIONES
    tft.fillScreen(WHITE);
    delay(500);
    tft.drawRoundRect(10, 10, 300, 60,15, RED); // Dibujamos un "boton" x inicio y
inicio x final y final
    tft.setCursor(20,18); // Colocamos el cursor
    tft.setTextSize(2); // Especificamos el tamaño del texto
    tft.setTextColor(BLACK); // Definimos el color del texto
    tft.println("Desea realizar otra "); // Escribimos por pantalla
    tft.setCursor(20,34);
    tft.println("serie de movimientos ? ");

    tft.fillRoundRect(40, 100, 250, 40, 15, CYAN);
    tft.drawRoundRect(40, 100, 250, 40, 15, RED);
    tft.setCursor(50,110);
    tft.println("Si");
```



```
tft.fillRoundRect(40, 150, 250, 40, 15, CYAN);
tft.drawRoundRect(40, 150, 250, 40, 15, RED);
tft.setCursor(50,160);
tft.println("No");

while(volver2==0)
{
  volver2();
}
bandera1=0;
if(volver2==2)
{
  seleccion2();
}
while(volver2==1)
{
  if(bandera1==0)
  {
    tft.fillScreen(WHITE);
  }
  delay(500);
  bandera1=bandera1+1;
  tft.setRotation(3); // Establecemos la posición de la pantalla Horizontal
  bmpDraw("4.bmp",0,0); // // Mostramos otra imagen en las coordenadas 0,0
                        // Para apagar finalizar el código y apagar el sistema
}
bandera=bandera+1;

}
}

}

tft.fillScreen(WHITE);
delay(500);

myservo1.write(45);
myservo2.write(10);
}
```