Universidad de Oviedo

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ÁREA DE INGENIERÍA

CONTROL REMOTO DE UN BRAZO ROBOT

DÑA. PAULA FARPÓN FERNÁNDEZ

TUTOR: D. JUAN CARLOS ÁLVAREZ ÁLVAREZ

FECHA: Julio 2024

ÍNDICE

TABLA DE ILUSTRACIONES

TABLA DE ECUACIONES

1. Introducción

1.1. IDENTIFICACIÓN DEL PROYECTO

Título	Diseño basado en modelos de sistemas multi-robot
Tutor	Juan Carlos Álvarez Álvarez
Autor	Paula Farpón Fernández
Fecha	Julio, 2024

1.2. PROYECTO PREVIO

La creación de esta propuesta surge para poder complementar un proyecto previo, el "Rover Asturiosity", en la Cátedra MediaLab de la Universidad de Oviedo.

Medialab es un laboratorio colaborativo en el cual estudiantes que se encuentran en sus últimos años de carrera crean diferentes proyectos con la intención de acercar la tecnología a las personas. Surge de una colaboración entre el Gijón Impulsa, el Ayuntamiento de Gijón, y la Universidad de Oviedo.

El proyecto "Rover Asturiosity" es la creación de un prototipo basado en el "Curiosity", que es un robot (tipo Rover) dirigido por la NASA con la misión de explorar el planeta Marte mediante la captura de imágenes y la detección de la superficie gracias a diversos sensores. [1]

Se tiene como objetivo diseñar y desarrollar el robot para poder ser mostrado en un ámbito educativo, llevándolo a diferentes escuelas e institutos para poder mostrar la tecnología a la gente más joven con un proyecto llamativo.

A lo largo de los años, varios residentes han trabajado sobre ello, consiguiendo un autómata funcional que es capaz de moverse controlado de forma remota con un mando de drones.

Al haber conseguido un modelo básico, lo que se pretende en la actualidad, es encontrar formas de ampliar su utilidad y versatilidad. Se busca expandir la aplicación de este robot en diferentes entornos, integrando distintos tipos de tecnología.

De ahí surge la idea de implementarle un brazo robótico, que puede ser muy beneficioso para el proyecto. Además de hacer que el prototipo sea cada vez más preciso y parecido al original, se crea la capacidad del robot para llevar a cabo diferentes tareas que antes no eran posibles, como el traslado de material de un lugar a otro.

La base de la que se parte es un brazo robótico creado con anterioridad por diferentes estudiantes de la Escuela de Ingeniería de Gijón que diseñaron la parte mecánica completa e incorporaron una parte electrónica que controlaba el brazo con un guante. A continuación, se describe el trabajo previo sobre el que se basa este proyecto.

1.3. ROBÓTICA INDUSTRIAL

La robótica es una disciplina que fusiona varios campos tecnológicos como la ciencia o la ingeniería con la finalidad de crear máquinas que tienen la capacidad de llevar a cabo distintas labores automatizadas. Se basan en programas y algoritmos que son controlados por personas e incluye desde la elaboración de software hasta la fabricación de elementos físicos.

En la actualidad, este campo técnico ha ganado gran importancia gracias a la cantidad de avances tecnológicos que ha generado y la rápida progresión en su estudio.

Dentro de este campo hay varios aspectos fundamentales que son esenciales para comprender cómo los robots interactúan con su entorno y llevan a cabo la función para la que han sido construidos.

La más importante es la movilidad y el control de dicho movimiento, algunos robots constan de componentes mecánicos como brazos articulados, pinzas… que les permiten manipular objetos.

Otro aspecto es la percepción del entorno, la cual se lleva a cabo mediante sensores implementados en el sistema. Estos sensores tienen la finalidad de poder detectar objetos, calcular distancias o poder evaluar diferentes características del entorno. Gracias a esto, es posible obtener información en tiempo real sobre la situación en la que se encuentra el robot, facilitando así la capacidad de adaptarse a ella.

Si se producen cambios en el entorno, es donde el siguiente aspecto toma relevancia, la adaptabilidad, ya que los robots necesitan ser capaces de ajustar su comportamiento en consecuencia. Con los avances en inteligencia artificial se ha podido profundizar en este aspecto en los últimos años.

También se tiene en cuenta la interactividad, ya que muchos robots están diseñados para interactuar con los seres humanos siendo capaces de comunicarse y colaborar con las personas.

La robótica es tan extensa que puede utilizarse en múltiples campos ofreciendo soluciones e innovaciones. En la medicina, por ejemplo, se utilizan para realizar cirugías de alta precisión, mejorando los resultados de estas. En la manufactura, se agiliza el tiempo de fabricación de diferentes objetos**…**

TIPOS DE ROBOTS

La Real Academia Española recoge la palabra robot como: “Máquina o ingenio electrónico que es capaz de manipular objetos y realizar diversas operaciones.” [2]

Los robots pueden ser distinguidos según ciertos factores para poder categorizarlos.

SEGÚN SU CRONOLOGÍA

PRIMERA GENERACIÓN: ROBOTS MANIPULADORES

Los robots de primera generación trabajan con un movimiento repetitivo previamente programado debido a que no son capaces de reconocer su entorno. Estos robots están creados para tareas muy simples y por ello se les implementa una serie de instrucciones específicas que repetirán de forma cíclica.

No son capaces de reconocer el entorno que los rodea, limita el rango de movimiento de forma considerable. Por ello, son especialmente útiles en la industria donde este tipo de movimientos sirven para las cadenas de ensamblaje o traslado de objetos de un punto a otro.

Ilustración 1: Robot de primera generación [3]

SEGUNDA GENERACIÓN: ROBOTS EN APRENDIZAJE

A diferencia de los anteriores, esta nueva generación es capaz de reconocer y observar su entorno mediante sensores avanzados y sistemas de retroalimentación. Estos robots pueden interpretar cambios y tomar decisiones en consecuencia. Aunque sus movimientos siguen siendo limitados, son más complejos y precisos que sus predecesores.

Todo esto permite que las tareas sean más precisas y que se reduzca la producción de unidades defectuosas al realizarlas. Pueden estar coordinados entre ellos y se pueden agregar interfaces de usuario más básicas.

Siguen siendo muy utilizados en la industria en tareas de soldadura e inyección de plásticos, entre otros.

Ilustración 2: Robot segunda generación [4]

TERCERA GENERACIÓN: ROBOTS REPROGRAMABLES

Se considera que los robots reprogramables son el principio de los robots inteligentes. Esto es debido a que destacan por su capacidad de aprendizaje y adaptación.

La gran diferencia de esta generación es que además de ser capaces de estudiar su entorno y ajustar sus acciones en respuesta a los cambios que haya, mejora su desempeño en el tiempo. Esta capacidad es posible gracias a la integración de inteligencia artificial, que

hace que tenga un aprendizaje automático y puedan desenvolverse de manera más eficiente en un futuro.

Ilustración 3: Robots tercera generación [5]

CUARTA GENERACIÓN: ROBOTS MÓVILES

Son bastante parecidos a la generación anterior, utilizando la tecnología de inteligencia artificial para poder tomar decisiones según la situación en la que se encuentren.

Un avance con respecto a su predecesor es que también utilizan internet de las cosas (IoT) para poder crear un crear un flujo bidireccional de información, facilitando la toma de decisiones en tiempo real.

Ilustración 4: Robots cuarta generación [6]

QUINTA GENERACIÓN: ROBOTS CON INTELIGENCIA ARTIFICIAL

La última de todas las generaciones todavía está empezando. Pero se estima que tendrán la capacidad de imitar las funciones cognitivas humanas mediante la inteligencia artificial, llegando al nivel de la singularidad, que es el punto en el que se tiene una IA plena.

En la quinta generación se producen dispositivos con componentes que posibilitan su movimiento autónomo, como pueden ser ruedas o extremidades. Siendo de utilidad en trabajos de construcción, producción, creación, y similares.

SEGÚN SU MOVILIDAD

ARTICULADOS/ BRAZO ROBÓTICO

Es el tipo de robot que será estudiado en este trabajo de fin de grado, por lo que, se llevará un análisis más exhaustivo en secciones posteriores.

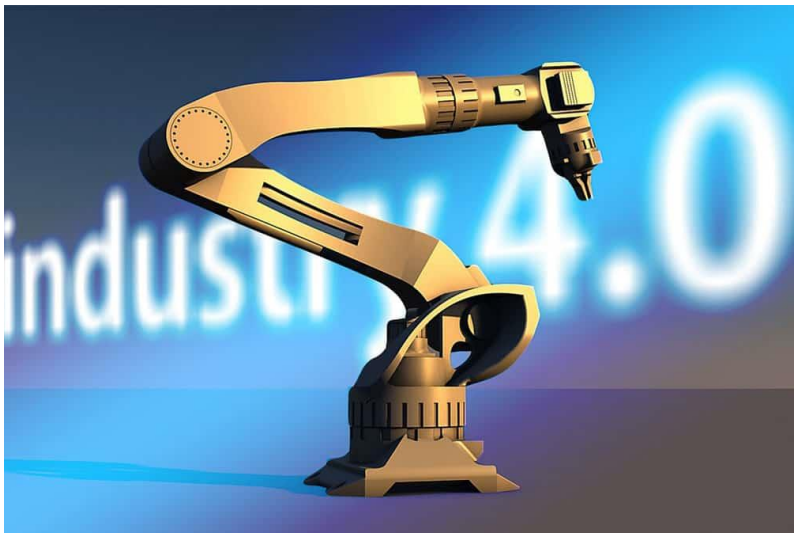

Ilustración 5: Brazo robótico [7]

GUIADO AUTOMATIZADO

Los vehículos guiados automatizados son sistemas capaces de desplazarse de un lugar a otro sin necesidad de intervención humana. Dependiendo del medio en el que se mueven pueden categorizarse en terrestre, aéreo o submarino.

La aplicación más utilizada se encuentra en el campo de la logística, creando un aumento en la producción y optimización.

Ilustración 6: Robot guiado automatizado [8]

HUMANOIDE

Están diseñados para tener apariencia y comportamientos similares a los humanos, utilizando la biomecánica para conseguir dicho objetivo.

Tienen elementos antropomórficos como cabeza, torso, brazos… para poder imitar de la forma más precisa posible el movimiento humano.

Aunque nacieron con el objetivo de poder entender mejor el funcionamiento del cuerpo humano para poder crear mejores prótesis actualmente la proyección que tienen se basa en la interacción social ya sea en usos de investigación, ocio o tareas domésticas.

Ilustración 7: Humanoide [9]

COBOTS

Son robots colaborativos que realizan trabajos en entornos industriales, llevando a cabo tareas repetitivas y manuales. Están diseñados para trabajar de forma directa con humanos por lo que tienen una alta capacidad de interacción.

Su programación es bastante sencilla y puede ser realizada de forma previa o bien en tiempo real.

Ilustración 8: Cobots [10]

SEGÚN SU FUNCIÓN

INDUSTRIALES:

Son aquellos más rudimentarios, con trabajos repetitivos para optimizar, sobre todo, cadenas de montaje o producción. Una de las tareas más significativas puede ser mover un objeto de un punto a otro.

DOMÉSTICOS

Ayudan en el hogar con diferentes tareas como limpieza o vigilancia, los más conocidos son las aspiradoras automáticas o robots de cocina que preparan comidas con programas ya implementados de forma previa.

EDUCATIVOS

Son kits básicos que enseñan el funcionamiento de una parte específica de la robótica para poder acercar la tecnología a una audiencia más joven haciendo así que se interesen por ello.

MÉDICOS

Son de gran utilidad, ya que, como se ha mencionado previamente, pueden llevar a cabo tareas muy complejas y precisas que el ser humano realizaría con mucha dificultad, como cirugías. También pueden ser robots que ayudan diariamente a gente con diferentes tipos de enfermedades.

MILITARES

Este tipo de robots son utilizados en misiones militares, dándoles tareas específicas como transporte, artificieros o incluso rastreadores.

1.4. INTRODUCCIÓN A LOS BRAZOS ROBÓTICOS Y CONTROL

Un brazo robótico es un dispositivo mecánico dotado de articulaciones para poder ejecutar un amplio rango de movimiento y libertad. Suelen disponer 6 articulaciones, dando así 7 grados de libertad al autómata, los mismos que un brazo humano.

Se pretende simular el movimiento humano para realizar tareas de forma automática con las ventajas que eso conlleva, como velocidad, fuerza o precisión.

Suelen implementar sensores, sistemas de visión o pinzas para poder ejecutar trabajos complejos y meticulosos. El sector que más se beneficia de este tipo de robots es el industrial, llevando a cabo labores como soldadura, ensamblaje…

Ilustración 9: Brazo robótico. [11]

1.4.1. CONCEPTOS BÁSICOS

PARTES

En esta sección se estudiarán las diferentes partes de un brazo mecánico, para ver como cada componente contribuye a su funcionamiento.

CONTROLADOR

Es la parte responsable de calcular las trayectorias que se utilizan para controlar el robot utilizando un software genérico, se utiliza un microordenador con unidad central. Dirige a los actuadores para que estos transformen la información recibida en movimiento.

ACTUADORES

Son los motores que cumplen la misión de realizar el movimiento. Convierten la energía eléctrica recibida por el controlador en energía mecánica que mueve las diferentes partes del brazo.

Existe una variedad considerable de tipos de motores que se pueden usar como actuadores. La decisión de cual utilizar se basa, sobre todo, en la cantidad de fuerza necesaria para mover el robot.

MANIPULADOR

El manipulador es la parte mecánica que realiza el movimiento designado por el sistema de control. Son segmentos conectados entre sí para poder realizar movimientos en diferentes direcciones. Están controladas por los actuadores que guían la trayectoria a seguir.

ARTICULACIONES/MUÑECAS

Son las conexiones que unen los diferentes manipuladores en el robot. Crean flexibilidad permitiendo generar los movimientos lineales y angulares con un amplio rango.

HOMBRO

Forma parte de las articulaciones, pero es la más importante de ellas, porque es la encargada de la rotación y elevación del brazo, creando movimientos tridimensionales en el espacio de trabajo.

MANO ROBÓTICA O GRIPPER

La mayoría de los brazos implementan como eslabón final un elemento específico para la realización de la tarea del brazo. Lo más habitual es una pinza para poder coger objetos.

SENSORES

No son estrictamente necesarios para la construcción de un brazo robótico, pero sí pueden ser de utilidad para poder detectar la posición y fuerza de este, lo que habilita un control más preciso y de la posibilidad de comprender el entorno en el que se encuentran para poder adaptarse al mismo.

Convierten diferentes magnitudes físicas, tanto del brazo robótico como del entorno, en magnitudes eléctricas que serán estudiadas por la unidad de control, que posteriormente llegarán a los actuadores.

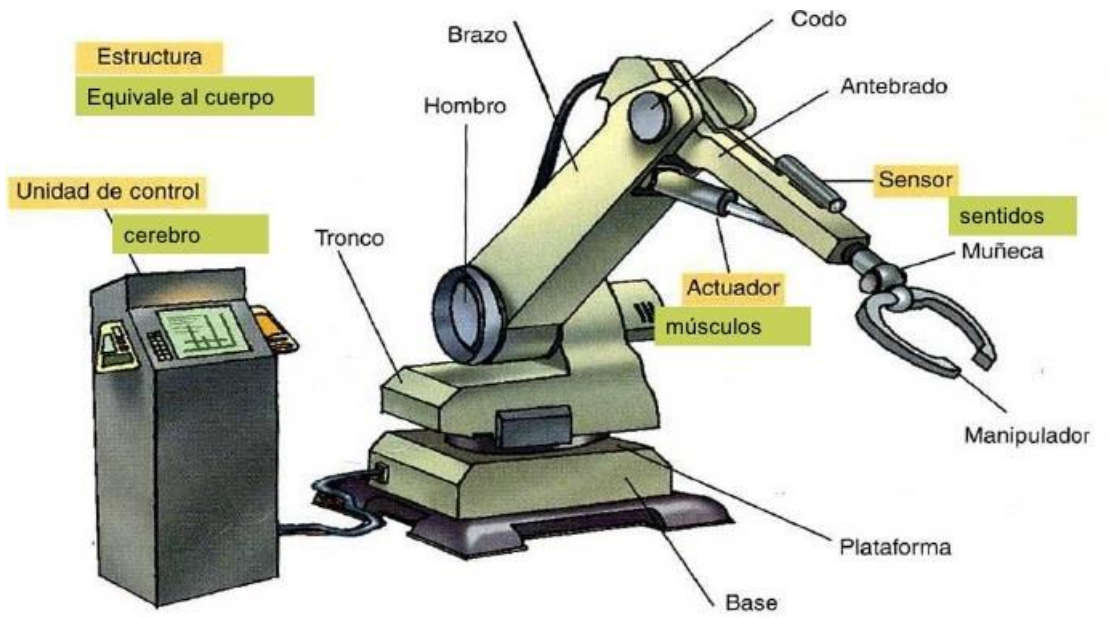

Ilustración 10: Partes de un brazo robótico. [12]

BENEFICIOS

Aportan muchos beneficios en el sector de la industria y a las personas que trabajan en ella.

La más significativa es la eficiencia y productividad, ya que pueden operar todo el día sin descanso, manteniendo la producción de forma continua y consistente.

Se utilizan en entornos peligrosos, de esta manera, se mantiene de forma segura a los trabajadores, bajando el riesgo laboral y lesiones en el lugar de trabajo de forma considerable.

Otro beneficio es que los brazos robóticos trabajan de forma más precisa que los humanos y con la capacidad de manipular cargas pesadas haciendo así que se puedan cumplir labores más complejas con mayor facilidad.

USOS

Este tipo de robots son muy versátiles, pudiendo realizar todo tipo de tareas en diferentes situaciones.

La más reconocida es el manejo y el transporte de material pesado, sobre todo en almacenes mejorando la logística de este.

Otra aplicación es la soldadura, en la fabricación automotriz son capaces de realizar piezas de gran calidad.

1.4.2. CLASIFICACIÓN

CARTESIANO

Está formado por dos o tres ejes en coordenadas cartesianas (X,Y,Z). Este tipo de brazo se mueve en trayectorias lineales y perpendiculares a lo largo de los ejes, en un espacio tridimensional. Son característicos por su simplicidad y facilidad de programación de sus trayectorias, haciéndolos muy versátiles. Es habitual su empleo en impresiones 3D.

CILÍNDRICO

Presentan un funcionamiento basado en articulaciones cilíndricas, creando así el mismo tipo de movimientos. Tiene una estructura tubular con tres articulaciones: una de ellas de revolución que le permite moverse en un plano horizontal y dos prismáticas que permite la extensión en sentido vertical. Aunque predomina el movimiento vertical. Es de utilidad en aplicaciones en las que se tiene un espacio limitado, en entornos pequeños.

ARTICULADO

Poseen múltiples componentes interconectados por articulaciones que se mueven en múltiples direcciones. Dichas articulaciones pueden ser tanto de revolución, para el control de la rotación, como de traslación, para crear el movimiento a lo largo de un eje.

SCARA

Las siglas de SCARA vienen de: Selective Compliance Assembly Robotic Arm. Consta de cuatro ejes, los cartesianos (X,Y,Z) con un eje rotativo adicional que le permite crear el movimiento de rotación para la orientación de la herramienta. Amplían el campo de trabajo en cuanto a los cartesianos pudiendo hacer tareas más complejas como atornillar.

ESFÉRICO/POLAR

Constan de una estructura esférica o semiesférica que se extiende desde la base. Tienen un funcionamiento muy similar a los cilíndricos, con la única diferencia de tener un eje rotativo en vez de uno lineal en el eje vertical.

PARALELO/DELTA

Este tipo de robots tiene varios brazos colocados en una base central, la gran ventaja que esto proporciona es la capacidad de llegar a altas velocidades.

1.4.3. PRINCIPIOS DE CONTROL

Para poder llevar a cabo el control del robot se pueden tener en consideración múltiples parámetros como la velocidad, fuerza o trayectoria. En este trabajo de fin de grado se va a estudiar el control de la posición de los diferentes eslabones para crear el movimiento deseado en el brazo.

Es necesario traducir las coordenadas a las cuales se quiere llevar el movimiento a las correspondientes posiciones articulares para llegar a ellas. Esto se consigue mediante la cinemática, que es una rama de la física que estudia analíticamente la geometría del movimiento mecánico respecto a un eje de referencia.

Existen dos modelos cinemáticos para el estudio de lo nombrado:

CINEMÁTICA DIRECTA:

Se estudian las variables de posición, velocidad y aceleración, no se tiene en cuenta ni las fuerzas ni los pares. En este tipo de cinemática se estudia la relación entre dichas variables y la posición y orientación del extremo del brazo o del efector final en el espacio tridimensional.

Los principios básicos son, como se ha mencionado, las variables de articulación, el sistema de coordenadas y matrices de transformación.

La metodología por utilizar es la siguiente:

Primero, se asocia un sistema de coordenadas a cada uno de los eslabones que tiene el brazo robótico a estudiar. La forma óptima de hacerlo es aplicar el Convenio de Denavit-Hartenberb.

A continuación, se obtiene la matriz homogénea que conecta cada sistema con el enlace anterior.

Para finalizar, la configuración del efector final será el producto de todas las matrices calculadas anteriormente. [13]

CINEMÁTICA INVERSA:

Al contrario que la cinemática directa, la inversa se basa en el estudio de la posición y orientación del efector final para calcular así los ángulos y geometrías de los eslabones para llegar a esta.

Se estudia con un sistema de ecuaciones no lineales, aunque no siempre se puede conseguir un resultado analítico. Si es el caso deberá intentar resolver mediante métodos numéricos iterativos.

2. Antecedentes

2.1. ESTRUCTURA MECÁNICA

Se parte de un diseño mecánico del brazo realizado por el exalumno del grado de ingeniería mecánica Ángel García López para su trabajo de fin de grado. [14]

Todas las partes mecánicas han sido impresas en 3D utilizando PLA como material. Consta de 6 grados de libertad que le otorgan una amplia gama de movimientos, permitiendo una gran capacidad de alcanzar posiciones y ángulos específicos.

A causa de estar en una etapa de desarrollo, su capacidad de levantar cargas será limitada.

Consta de 5 eslabones principales:

- Eslabón 1: Es la base de la estructura, su función principal es la rotación completa del brazo, con un rango de 360º. Es accionado por un motor paso a paso.
- Eslabones 2, 3, 4: Son las partes responsables de variar la altura del brazo, con movimientos ascendentes y descendentes. Con la combinación de todos se pueden llegar a un amplio abanico de posiciones. Todos ellos están accionados por servomotores.
- Eslabón 5: Es el eslabón final, consta de una pinza para poder sujetar objetos.

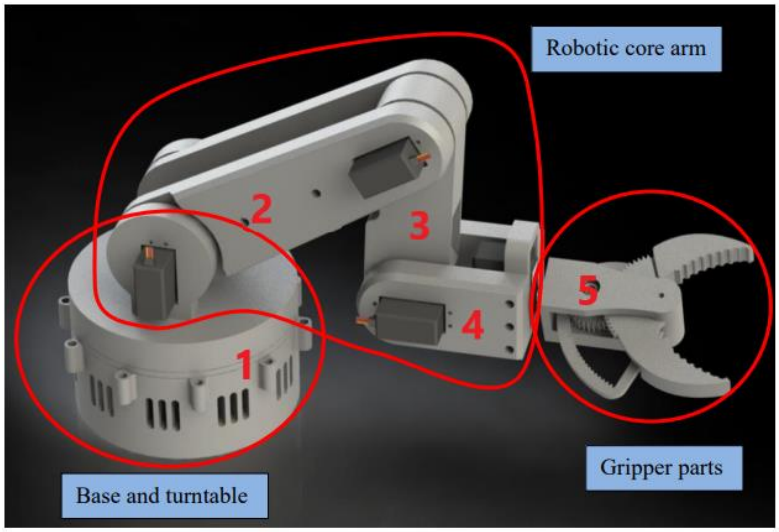

Ilustración 11. Estructura del brazo robótico. [14]

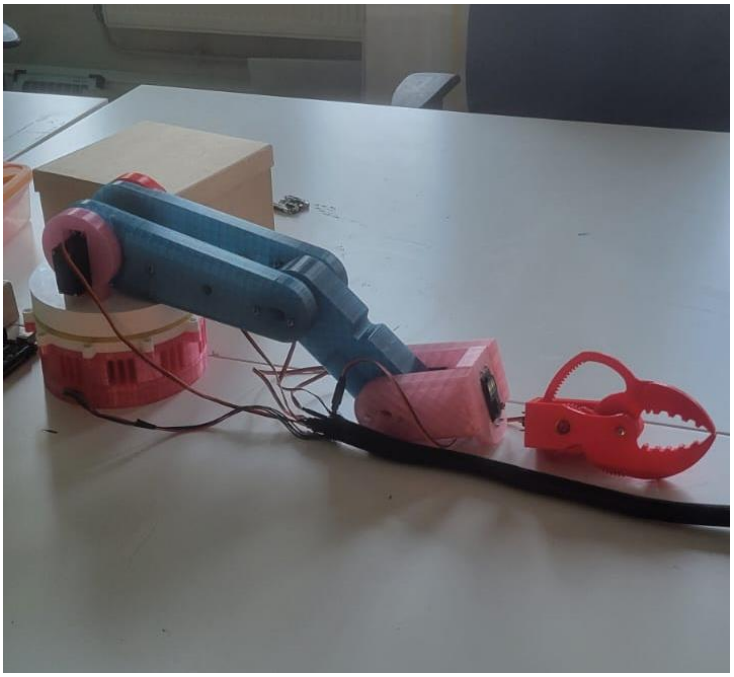

Ilustración 12. Estructura brazo real.

2.2. ESTRUCTURA ELECTRÓNICA

La parte electrónica, al igual que la mecánica fue diseñado por el exalumno Javier Martínez Becerra. [15]

Se basa en dos partes diferenciadas, la primera de ellas trata el emisor de los comandos, se hará mediante un guante dotado de sensores que enviará la información recogida del movimiento natural de la mano del usuario a la segunda parte, un brazo robótico para que así pueda replicar dicho movimiento.

Los componentes utilizados en cada una de las secciones se describen a continuación.

COMPONENTES BRAZO ROBOT

- Arduino UNO.
- 6x Servomotor MG995.
- Motor paso a paso NEMA 17.
- Driver para motor paso a paso 8825.
- Batería 4.8 V 2400 mAh.
- Batería 12 V 2500 mAh.
- Módulo bluetooth HC-05.
- PCB.
- Cables.

COMPONENTES GUANTE ROBOT

- Arduino UNO.
- Guante de constructor.
- Módulo bluetooth HC-05.
- 2x MPU6050 (Acelerómetro).
- 3x Sensores flexibles.

- 4x Resistencias de 10k.
- Batería de 4.8 V 2200 mAh.
- LED.
- PCB.
- Cables.

3. Objetivo

El principal propósito del trabajo es lograr dirigir el movimiento del brazo mediante órdenes separadas desde el ordenador, en lugar de sensores como se hace en la actualidad. Ya que esto hará que sea más fácil implementarlo en el robot "Asturiosity".

Para poder lograrlo, es necesario:

1) Verificar que todos los componentes estén operativos.
2) Modificar la manera en la que los actuadores reciben la información. Actualmente se recibe mediante sensores implementados en un guante, el objetivo final es hacerlo mediante comandos desde el ordenador.
3) Transformar la información para convertirla en movimiento.
4) Mandar a los actuadores las órdenes para realizar dicho movimiento.

Se utilizarán diversas tecnologías y conocimientos adquiridos en la carrera para poder conseguirlo.

4. Estructura del documento

El presente Trabajo Fin de Grado está compuesto por los siguientes documentos:

- Memoria: Se muestra toda la información sobre el proyecto realizado, pruebas realizadas y resultados finales.
- Anexos.
    - Planificación temporal.
    - Códigos.

5. Diseño mecatrónico del brazo robot

5.1. DISEÑO MECÁNICO

Los componentes principales del ensamblaje del robot pueden dividirse en tres partes diferenciadas que formarán el sistema completo:

5.1.1. BASE

Está constituida por 8 piezas que permiten la rotación completa del robot. Además de crear este movimiento también actúan como apoyo estructural, dando estabilidad al conjunto.

Las distintas secciones que componen la base son:

- Placa base
- Abrazadera de rodamiento
- Abrazadera de placa giratoria
- Base cilíndrica
- Tapa de base
- Anillo de rodamiento
- Engranaje_27T
- Plato giratorio

Dentro de la base cilíndrica se encuentra un motor paso a paso para poder crear la rotación del sistema.

En la siguiente figura se muestra un despiece de cómo están colocadas las diferentes partes que forman esta parte del conjunto.

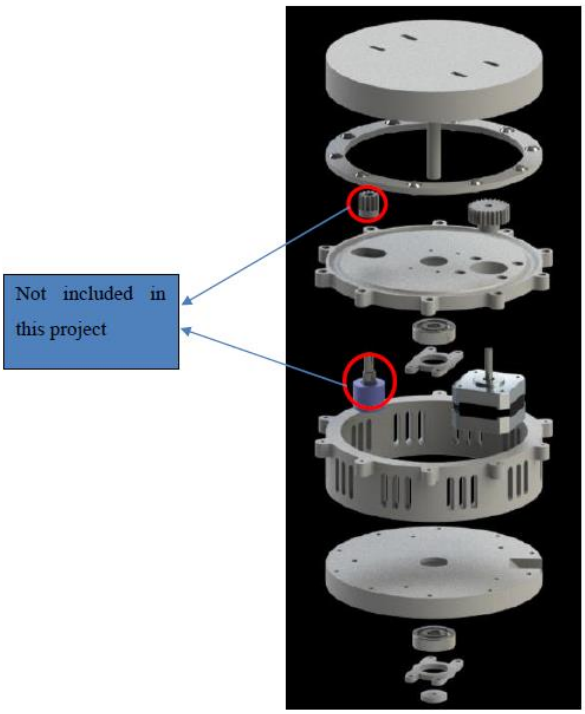

Ilustración 13: Piezas Base [14]

5.1.2. BRAZO

Es la zona central del sistema la que llevará el peso del movimiento del robot. Está conectada a la base mediante dos piezas intermedias, las cuales son soportes básicos para servos.

Las partes en las que está dividida son las siguientes:

- Eslabón 2 (2 piezas)
- Guía rodamiento
- Eslabón 3 (1 pieza)
- Eslabón 4 (2 piezas)
- Montura final de servo
- Conectores eslabón 2 (2 piezas)

Tanto en el eslabón dos, como en el cuatro y en la montura final de servo, irán implementados servomotores. Haciendo así que el total de servomotores en esta sección sea de 5.

Ilustración 14: Piezas Brazo [14]

5.1.3. PINZA

Es el efector final de la estructura, encargado de coger objetos mediante la apertura y cierre de la pinza, trabajo que necesita gran precisión, de modo que todas las partes de la pinza tiene que estar bien diseñadas y ensambladas para poder ejecutar la tarea ordenada. Es accionado por el último servomotor del robot (el sexto) que está atornillado a este segmento.

Las secciones de las que consta esta parte son:

- Montura del servo
- Tapa
- Guía de rodamiento
- Pinza_120T
- Pinza_72T
- Engranaje de servo
- Casquillo de rodamiento

Ilustración 15: Piezas Pinza [14]

5.2. ACCIONAMIENTOS

Los accionadores forman la parte principal de este proyecto debido a que son aquellos que crean los movimientos físicos concretos para conseguir llegar a la posición deseada.

5.2.1. SERVOMOTORES

Es un dispositivo electromecánico de alto rendimiento que transforma las señales de control recibidas por el sistema de control, en este caso, un Arduino UNO en movimientos mecánicos.

Utiliza un sistema de control en bucle cerrado con un sensor de posición. Este sensor proporciona la información necesaria para ajustar su posición según las órdenes que reciba mediante la retroalimentación de posición.

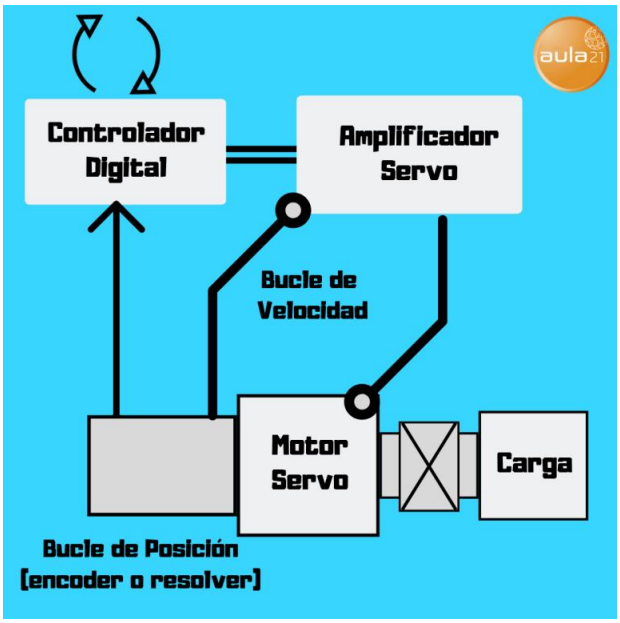

Ilustración 16: Funcionamiento básico de un servomotor [16]

Pueden ser de tipo corriente alterna o corriente continua, los primeros son utilizados en ocasiones en las que se utilice la velocidad constante, mientras que el segundo tipo es cuando la velocidad es variable.

MODELO UTILIZADO Y CARACTERÍSTICAS

Para la realización de este proyecto se utilizarán dos modelos diferentes de servomotores. La razón de esto es que en la parte más baja del brazo, en el eslabón 2, se necesita más fuerza para mover el brazo, ya que carga con todo el peso de este, si se pusieran unos servomotores como el que se necesita en el efector final no se podría realizar el movimiento correctamente.

Por ello, los servomotores que se pondrían en la parte inferior del eslabón dos, conectado con la base son los DS3225.

Las especificaciones eléctricas que resultan útiles para este trabajo son los siguientes:

- Rango de voltaje: 4.8-6.8V
- Velocidad de operación (5V): 0.15 s/60º
- Torque (5-6.8V): 21-25 kg/cm
- Ángulo de giro: 180º

Ilustración 17: Servomotor DS3225 [17]

En el resto del sistema al no necesitar llevar tanta carga de manera se utilizaron el modelo MG995.

Las características importantes para este trabajo son:

- Rango de voltaje: 4.8-7.2V
- Velocidad de operación: 0.2s/60º (4.8V) – 0.16s/60º (6V)

- Torque (4.8-6V): 8.5-1. kg/cm
- Ángulo de giro: 120º

Ilustración 18: Servomotor MG995 [18]

COMPONENTES PRINCIPALES

En el interior de un servomotor se encuentran tres componentes: un motor, que realiza el movimiento, un amplificador o variador que controla diferentes variables como el torque o la frecuencia y un mecanismo de retroalimentación.

Generalmente también incluyen una fuente de alimentación y un servocontrolador.

Los servomotores tienen tres cables:

- Rojo: Alimentación
- Negro/Marrón: GND (Ground o masa)
- Amarillo/Naranja: Señal

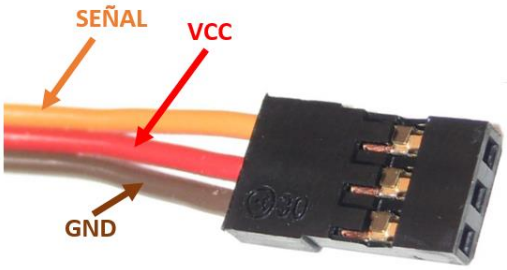

Ilustración 19: Conexión servomotores [19]

SEÑALES DE CONTROL DE SERVOMOTORES (PWM)

Las señales analógicas son aquellas que pueden adoptar cualquier valor en un rango específico. Mientras que las digitales son discretas, por lo que solo pueden tomar los valores 0 y 1.

En esta técnica, Pulse Width Modulation (PWM), o modulación por ancho de pulso, se utilizan señales digitales para transmitir información analógica. Esto se realiza modulando el ancho de los pulsos digitales.

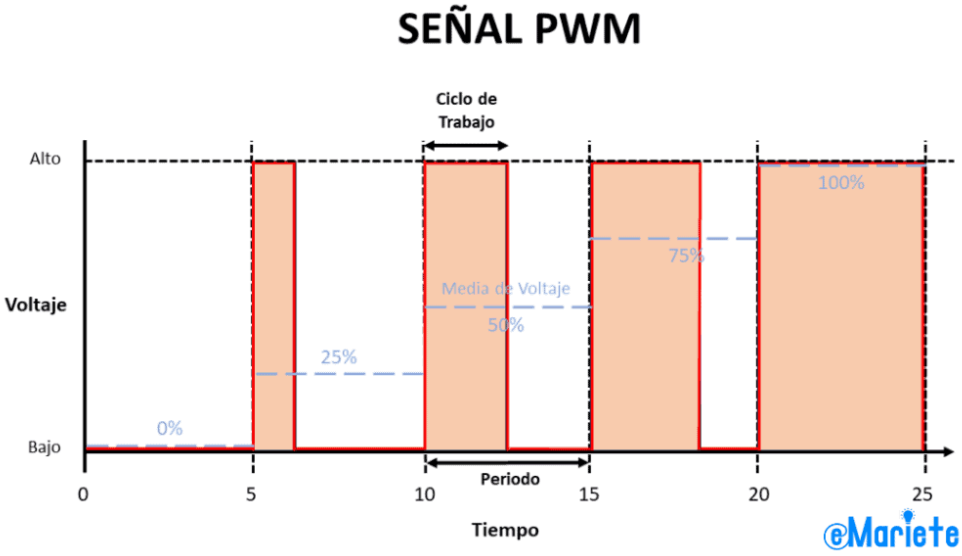

Ilustración 20: Señal PWM [20]

El tiempo en el cual el pulso se encuentre alto en un periodo determina el ciclo de trabajo o duty cycle.

Dicho parámetro es calculado con la siguiente fórmula:

$$Ciclo\ de\ trabajo(\%) = \frac{tiempo\ en\ parte\ positiva}{Periodo}$$

Ecuación 1: Ciclo de trabajo

USO DE LA SEÑAL PWM PARA CONTROLAR LA POSICIÓN

En la forma en la que trabaja la señal PWM se puede utilizar para el control de los diferentes servomotores del sistema.

Si el ciclo de trabajo es del 100%, el servomotor recibe potencia, creando el movimiento con la velocidad máxima que admite el motor.

Si, por el contrario, el ciclo es del 0% no recibiría ninguna información por lo que el motor se quedaría quieto.

Para que el servomotor los interprete se deberá trabajar con una frecuencia de 50Hz o 20ms.

Cuanto más ancho sea el pulso, de más grados constará el movimiento, como se puede observar de forma gráfica en la Ilustración 21.

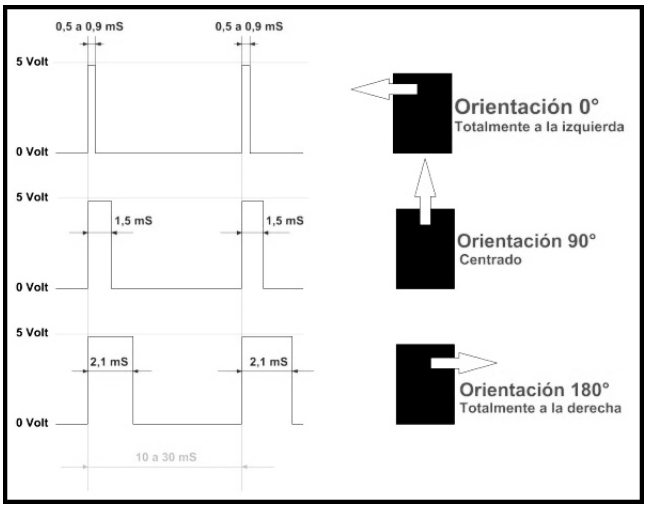

Ilustración 21: PWM [21]

INTERFAZ Y CONEXIÓN DE SERVOMOTORES CON ARDUINO

La conexión directa de los servomotores al Arduino es bastante simple. Como se ha comentado previamente, se tienen 3 cables por motor.

El primero es el de GND, por lo que habría que juntarlos todos a un punto común que posteriormente será conectado al *ground* del Arduino. Lo mismo pasaría con la alimentación.

Para el cable de la señal se asocia un pin a cada uno de los servos utilizados y se conecta al propio.

La conexión final debería verse de la siguiente forma:

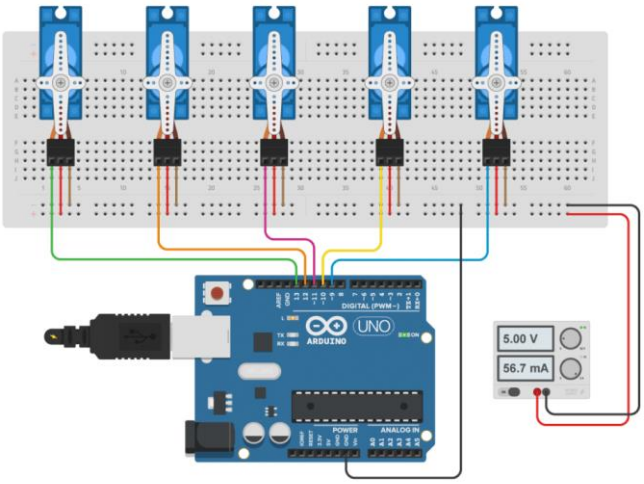

Ilustración 22: Conexión Servo-Arduino [22]

5.2.2. PLACA PCA9685

DESCRIPCIÓN GENERAL DE LA PLACA

Esta placa es un controlador de ancho de pulso (PWM), con la capacidad de controlar 16 salidas de este tipo. Es útil para poder controlar múltiples servomotores a la vez, lo que será de gran utilidad para este proyecto.

Se comunica mediante el protocolo I2C que será estudiado en profundidad en siguientes apartados.

CARACTERÍSTICAS TÉCNICAS

Este componente presenta una serie de características que son:

- Alimentación: 3-5V
- Alimentación motores: 6V
- Frecuencia de salida: 40-1000Hz
- Canales: 16
- Resolución por canal: 12 bits
- Interface: I2C

[23]

DESCRIPCIÓN DE LOS COMPONENTES PRINCIPALES

La placa PCA9685 está compuesta por varios componentes:

CHIP PCA9685:

Es el componente principal de la placa, encargado de controlar las salidas y entradas que comunican con el Arduino.

PINES DE SEÑAL/CONTROL:

- SCL: Pin del reloj I2C
- SDA: Pin de datos I2C

- OE: Salida habilitada, su uso es para deshabilitar todas las salidas. Si está bajo todos los pines están habilitados, si está alto pasará lo contrario.

PINES DE ALIMENTACIÓN:

- GND: Pin que conecta la placa con tierra o *ground*.
- VCC: Pin de alimentación lógica
- V+: No es necesario conectarlo, suministra energía a los servos de forma más directa y distribuida.

TERMINALES DE SALIDA PWM:

Esta placa consta de 16 puertos de salida, cada uno de ellos tiene tres pines: V+, GND y PWM, que son los puntos donde se conectan los servomotores.

PUENTES DE DIRECCIÓN (A0 A A5):

Son los responsables de configurar la configuración I2C y cambiar la dirección del chip (0x40) si se diera el caso de estar utilizando varias placas PCA9685 en el mismo bus I2C.

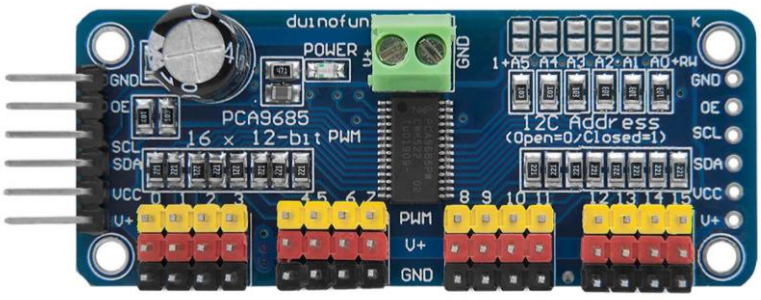

Ilustración 23: Placa PCA9685 [24]

PROTOCOLO DE COMUNICACIÓN I2C

I2C es una abreviatura de Inter-Ic (Inter integrated circuits) que es un tipo de bus utilizado para conectar circuitos integrados.

Es un protocolo que trabaja con dos hilos donde diferencia los componentes que están comunicados entre sí en dos grupos: maestros y esclavos.

Trabaja con dos líneas de colector abierto que son SCL y SDA, la primera de ellas hace referencia a una señal de reloj, y la segunda a la línea que lleva los datos.

Los maestros se encargan de la gestión del reloj controlando el flujo de la comunicación y de iniciar y parar la comunicación entre los dispositivos.

Los esclavos suministran información al maestro y responden ante las solicitudes de este.

CONEXIÓN PLACA PCA9685 – ARDUINO

La conexión correcta entre la placa y el Arduino se muestra en la siguiente ilustración:

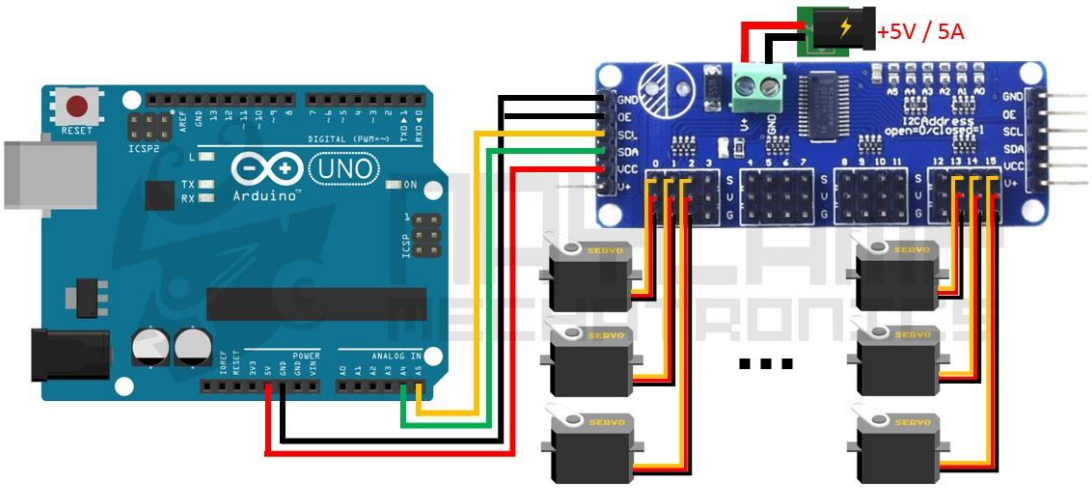

Ilustración 24: Conexión Placa PCA9685 – Arduino [25]

5.3. DISEÑO ELECTRÓNICA

A continuación, van a describirse los diferentes elementos utilizados en el conjunto y la forma de conexión entre ellos.

ARDUINO

Es uno de los microcontroladores de código abierto más extendido debido a su simplicidad y amplia gama de aplicaciones. Es una herramienta potente y versátil tanto para principiantes como para usuarios más avanzados en la materia.

COMPONENTES PRINCIPALES:

- Microcontrolador ATmega328: Componente central de la placa, es la parte capaz de ejecutar el código y llevarlo a la placa para que esta cumpla con las funciones requeridas.
- Regulador de Voltaje: Transforma la entrada de 7-12V a 5V que es el voltaje utilizado por la placa.
- Conector de alimentación: Jack de 2.1mm para poder conectar una alimentación externa.
- Conector USB Tipo B: Es la parte que conecta el Arduino con un ordenador y proporciona alimentación.
- Pines de entrada y salida: Hay dos tipos de pines, digitales y analógicos.

  DIGITALES: Pueden ser configurados tanto de entradas como salidas. Su uso es la lectura o envío de señales digitales. Hay 14 pines de este tipo.

  ANALÓGICOS: Son los nombrados A0, A1, A2, A3, A4 y A5. El rango en el que puede variar los valores es entre 0 y 1023.
- Pines PWM: Hay 6 de los pines digitales que pueden proporcionar una salida PWM, vienen indicados con el símbolo “~”
- Conector ICSP: Se puede programar el microcontrolador mediante un programador ICSP.

- Botón de *reset*: Reinicia el programa de 0, no borra la memoria del Arduino, simplemente vuelve a la parte inicial del código.
- LED TX RX: Reflejan la transmisión de datos (TX) y de recepción de datos (RX). Este LED se encienden cuando hay algún tipo de transmisión de este tipo.

CARACTERÍSTICAS PRINCIPALES:

- Voltaje de funcionamiento: 5V
- Voltaje de entrada: 7-12V
- Voltaje de entrada límite: 6-20V
- Máxima corriente para entradas: 20mA
- Máxima corriente para pines: 3.3V 50mA
- Memoria flash: 32KB
- Velocidad del reloj: 16Mhz
- EEPROM: 512B
- SRAM: 512B
- Conector de alimentación: Jack de 2.1mm

[26]

ARDUINO IDE

IDE (siglas en inglés de Integrated Development Environment), es un entorno de programación de software libre y código abierto.

Utiliza resaltado de sintaxis, lo que significa el uso de colores en diferentes partes del código para una escritura y lectura más sencilla.

El monitor serial sirve para obtener información directamente de la placa y poder interactuar con esta, ayudando a la detección de errores.

Otra de las ventajas de este entorno es el director de bibliotecas, posee una gran variedad de bibliotecas donde poder encontrar comandos que faciliten la creación del código deseado.

COMUNICACIÓN Y PROTOCOLOS (UART, I2C)

Dentro de la comunicación con Arduino hay dos tipos que resaltan: UART, e I2C. Dependiendo del tipo de conexión que se quiera, se utilizará una u otra.

Generalmente cuando solo se necesita la conexión entre dos dispositivos se utiliza la UART debido a que no necesita configuración de direcciones. Por el contrario, aun necesitando dicha configuración, la conexión I2C permite la conexión entre más dispositivos.

6. Conexiones

6.1. ANTIGUA CONEXIÓN

La forma en la que las diferentes partes estaban conectadas previamente era mediante una placa PCB que juntaba el Arduino con los distintos componentes que crean el movimiento y control del brazo.

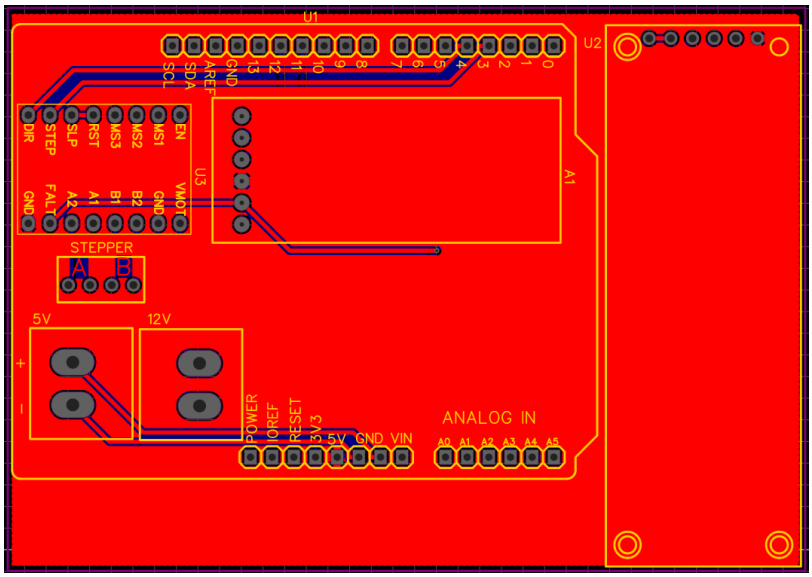

Ilustración 25: Placa PCB Brazo [15]

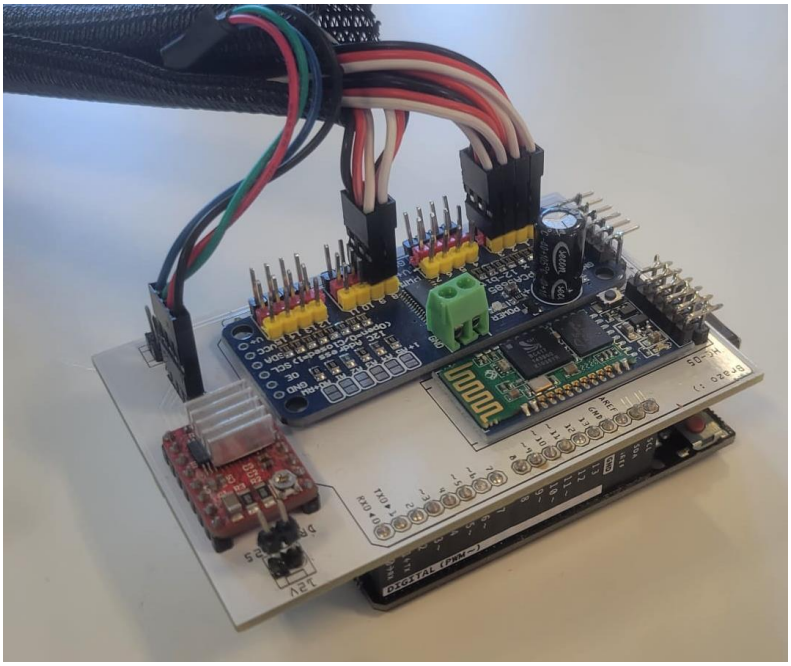

Ilustración 26: Placa PCB Conexiones

6.2. NUEVA CONEXIÓN

Debido a múltiples razones que serán comentadas en el apartado siguiente, la conexión final es bastante dispar a la previa.

Actualmente tanto los servomotores, están conectados directamente al Arduino.

Tras el estudio de las posibilidades de implementación en el proyecto final del Rover tras varias pruebas con el motor paso a paso se decidió no utilizarlo para la rotación de la base, debido a que el movimiento puede ser realizado directamente con el robot final.

Tampoco es necesario el uso del módulo bluetooth, debido a que la recepción de información ahora será realizada mediante el serial y no por sensores. Tampoco se utilizará la placa PCA9685 por las complicaciones que ha creado en el funcionamiento correcto del brazo explicadas en secciones siguientes.

Los componentes que están presentes son:

- Arduino Uno
- 4 servomotores MG995
- 2 servomotores DS3225

La conexión actual se representa en la Ilustración 28:

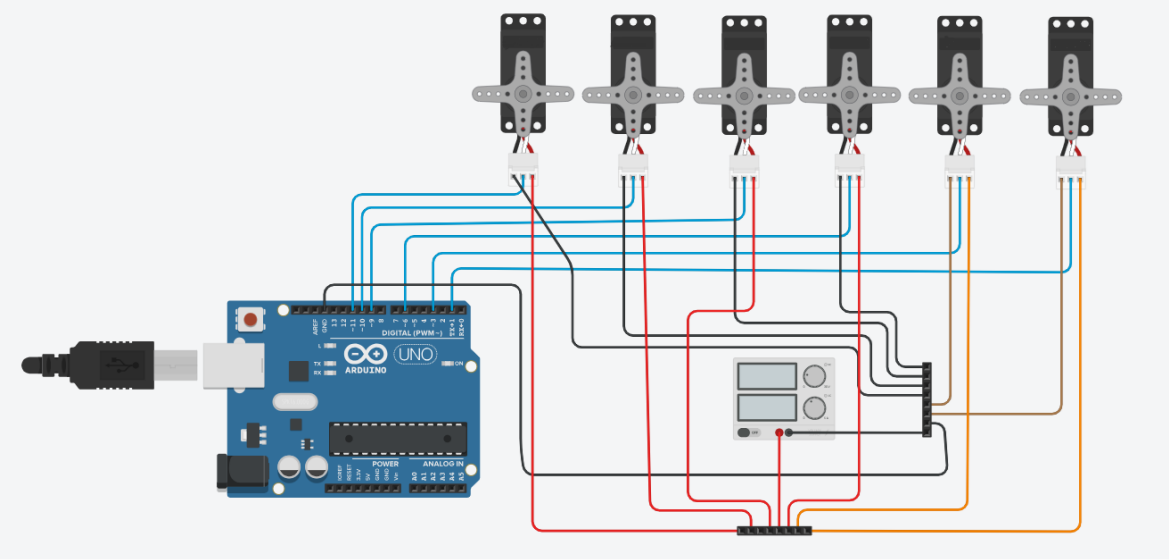

Ilustración 27: Conexión actual servos

7. Pruebas electrónicas

7.1. PLACA PCA9685

La razón del fallo inicial de esta placa era la conexión entre el elemento y la PCB que no estaba bien realizada, haciendo así que se desconectara de forma continua. Para solucionarlo, se cambió la PCB por una nueva para poder asegurar que todos los enlaces estuvieran correctos.

Aun estando bien conectada, no les llegaba suficiente voltaje a los servomotores debido a que el pin V+ no funcionaba correctamente, no estaba conectado a los distintos pines asociados a este. Es un problema bastante común que puede suceder cuando por error se conecta VCC a V+, resultando en un daño irreparable a la placa.

Tras descubrir esto, se procedió a cambiarla por una nueva. En principio no hubo más problemas con esta, pero por culpa del uso de la librería finalmente no se utilizó para el proyecto.

7.2. SERVOMOTORES

Tras los cambios realizados en la parte mecánica (Parte 4.3) se pudo percibir que a los servomotores de la base les costaba realizar el movimiento debido al peso de las piezas de todo el brazo, por ello se sustituyeron por un modelo más potente para poder llevar a cabo la tarea.

Este nuevo modelo por utilizar fue el DS3225.

Otro problema que hubo en cuanto a los servomotores fue que aquellos colocados en el eslabón 2 y 4 creaban un pico de amperaje que además de ser peligroso por poder estropear el resto de los componentes, no llegaba a realizar la trayectoria que debería.

Se comprobó si el problema residía en los pines de la placa PCA en los que estaban conectados, pero no era así, ya que si se conectaba un servo externo a dichos pines no se creaba ese problema. Pero sí sucedía si se conectaban esos servos a cualquier otro pin. Debido a esto se optó por cambiar los servos por unos nuevos que no dieron problemas futuros.

7.2.1. LIBRERÍA ADAFRUIT

Para crear la comunicación entre el Arduino y la placa PCA al principio se utilizaron las librerías Wire.h y Adafruit_PWMServoDriver.h

Wire.h es una biblioteca que permite la comunicación I2C entre los componentes. Mientras que Adafruit es una biblioteca específica para la placa PCA9685 que es de gran uso para la configuración de los movimientos a realizar por los servomotores y el control de esta.

En la mayor parte del proyecto se estuvieron utilizando estas dos librerías, pero llegó un punto en el que el código estaba sin ningún fallo, pero no se movían los motores. Tras múltiples revisiones se identificó que el único lugar donde podría haber un fallo era dentro de un comando de la librería, en concreto “pwm.setPWM()” que es el encargado de crear el movimiento.

No hubo forma de encontrar una alternativa para poder hacer que se movieran los servos, por lo que se tomó la decisión de usar la librería servo, para poder solucionarlo.

Esta resolución llevo al hecho de que la placa PCA9865 ya no sería de utilidad en el proyecto, por lo que las conexiones tendrían que ser sin ella, conectando los servomotores de forma directa al Arduino.

7.2.2. LIBRERÍA SERVO

Esta librería es una de las más utilizadas en cuando a control de servos se refiere. Fue la elegida finalmente en este trabajo de fin de grado debido a las complicaciones obtenidas con otras librerías. Se trabajó sin ningún problema y el código final cumple con la tarea definida.

8. Pruebas mecánicas

8.1. BASE

Con el diseño anterior había un gran problema de estabilidad cuando se realizaba cualquier movimiento en el brazo. La inercia y velocidad hacían que la base volcara. De este modo, el funcionamiento se veía parado para poder restaurar la posición ideal.

La primera idea para arreglarlo de forma rudimentaria fue añadirle un peso en la parte superior de la base, pero no era una solución a largo plazo dado el hecho de que en ciertos ángulos del brazo podía llevar a interferencias entre el peso y el brazo.

La solución final fue el diseño de una pieza que aumentara el tamaño de la superficie, a la vez que implementaba unas pestañas para atornillar la base a unas maderas creando mucho más soporte.

Para realizar el diseño se utilizó el software Autodesk Inventor, tomando las medidas pertinentes de la base para que esta nueva pieza encajara en el sistema. Posteriormente, para su impresión 3D se utilizó el programa PrusaSlicer.

La pieza final sería la siguiente:

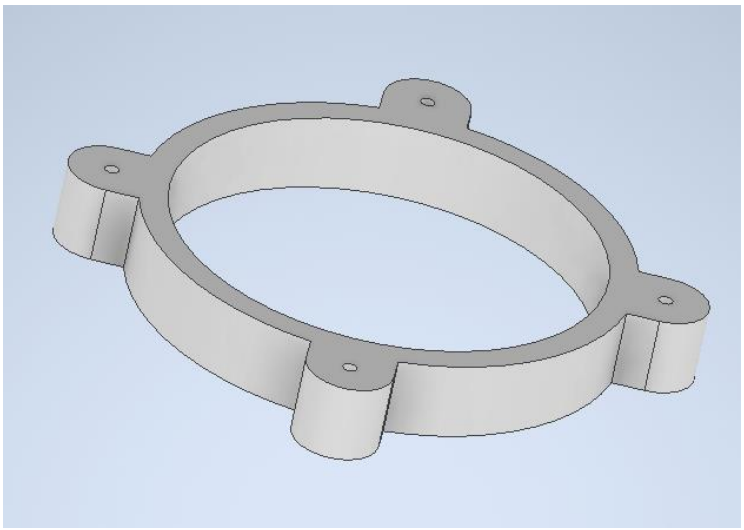

Ilustración 28: Nueva pieza base

8.2. BRAZO

Las piezas que constituían el brazo no estaban bien ensambladas, y se desmontaban de forma continua, algunas de las conexiones entre las piezas y los servomotores estaban desgastados por lo que no se llegaba a realizar el movimiento.

Se procedió a un cambio completo de todos los componentes, se le subió el porcentaje de relleno para poder tener más robustez. Después de ello se pudo observar que los motores tenían dificultad para moverse, esto era por el incremento que se había realizado, de manera que se realizaron nuevas piezas con la misma cantidad de relleno que las iniciales.

Finalmente, tras un cauteloso montaje se consiguió obtener un brazo útil.

8.3. PINZA

El funcionamiento de la pinza no era del todo correcto, se podía observar que las piezas estaban sueltas y no tenía la fluidez ni exactitud necesaria.

La pieza que estaba implementada estaba desgastada, aunque el problema principal fue el hecho de que los arcos no estaban bien formados.

Al diseñar, el programa en vez de crear circunferencias interpola y crea poliedros de muchos lados, cuando se exporta la malla para imprimirlo hay que seleccionar un grado de calidad, si no se opta por un grado alto suficiente las piezas no tendrán la precisión suficiente para encajar y poder realizar la tarea.

La solución fue volver a imprimir la pieza con la mayor calidad posible para poder resolver el problema. Una vez realizado se procedió al montaje de la pinza y se comprobó su correcto funcionamiento.

Otro factor que se debió tener en cuenta era la conexión entre el brazo y la pinza. Inicialmente, estaba unida directamente, es decir, el servomotor iba introducido dentro de la pieza misma de la pinza, creando un desgaste del material en cada movimiento. Por ello, el agujero estaba dado de si y se desmontaba de forma constante.

El remedio utilizado fue la implementación de un brazo de servo para poder reforzar la unión entre las diferentes partes sin que se creara dicho desgaste.

9. Software (Código brazo)

En este apartado se procede a explicar el código utilizado finalmente que puede observarse en el apartado de Anexos.

La parte inicial se basa en iniciar el código, tanto en la librería como los diferentes servos que van a utilizarse.

Se han creado dos funciones principales a utilizar:

La primera “moveServoSmoothly” es para realizar el movimiento del servomotor de una forma sutil y uniforme.

Se lee la posición a la que se quiere ir y la guarde en la variable currentPos, después crea un bucle en el cual si la posición a la que se quiere llegar es mayor a la actual hace que el servo se mueva de forma uniforme de un grado de cada vez hasta llegar al ángulo deseado.

La otra función se llama “setAnguloBase”, sirve para mover los dos servos de la base que debido a que están enfrentados, tienen que trabajar de forma síncrona. Si solo se moviera uno, haría el movimiento solo sobre una parte del eslabón, haciendo que se partiera el brazo.

Por ello cuando se le introduce un ángulo al que llegar hace que un motor se mueva tres grados, a continuación, se moverá el otro 3 grados, y así de forma continua hasta llegar a la posición deseada.

En la parte del *setup* se define en que pin va a estar cada uno de los motores y se mueve cada uno a un ángulo específico para que el brazo quede con un estado neutro para poder realizar un rango de movimientos más amplio.

Ya en la parte del *loop* es en la que se pregunta al usuario cuál de los servos se quiere mover, se introduce el número deseado mediante el serial y a continuación se pregunta de nuevo de cuántos grados se quiere realizar el movimiento. Seguidamente, el brazo creará el movimiento realizado y volverá a preguntar por el servo a mover.

10. Implementación al "Asturiosity"

A la vez que se trabajaba en el funcionamiento del brazo también se estudió la forma de poder implementar el brazo al robot actual.

El "Asturiosity" ya consta de toda la electrónica para poder moverse de forma autónoma mediante los comandos recibidos por un mando de drones.

La idea principal era controlar todas las partes desde un mismo dispositivo para hacer la experiencia del usuario más sencilla e intuitiva. Se conectaría todos los componentes a una Raspberry Pi por serial.

A su vez, esta creará una red wifi para conectarse a un dispositivo desde el que se reciban los comandos, como un móvil.

Tras la instalación del sistema operativo en la Raspberry, se creó un Hotspot para poder realizar la conexión de los componentes.

10.1. INTERFAZ

Para poder cohesionar todos los elementos que forman el robot, que son, la parte central, siendo el Rover como tal, la cámara y el brazo se creó una interfaz en la plataforma AppInventor que es la siguiente:

Ilustración 29: Interfaz Raspberry

Consta de un joystick para poder controlar los diferentes movimientos.

También, tiene tres elementos que determinan el movimiento de la pinza, que son: dos botones para rotar hacia las dos direcciones y un slider para poder abrir y cerrarla de forma meticulosa.

Finalmente, hay cuatro botones en la esquina de arriba a la derecha que están destinados para dirigir hacia donde apunta la cámara web. A la izquierda de estos botones se encontraría la visualización de dicha cámara.

10.2. CAMBIO DE DIRECCIÓN

Tras varias reuniones del equipo que trabaja en el robot se decidió que, en vez de utilizar una app o la interfaz previa, el control de todo ser realizaría mediante https que crea la misma Raspberry, no con AppInventor.

Los comandos del brazo se recibirían en forma de lista desde la Raspberry, por lo que habría que realizar un código que lea esos datos y sea capaz de transmitirlos para crear el movimiento deseado.

11. Conclusiones y posibles mejoras

Como conclusión diría que trabajar con un elemento físico conlleva mucho más control sobre los elementos que se están estudiando que cuando se está trabajando con una simulación. Debido a que en cada cambio que se realice, hay que comprobar que el resto de los componentes siguen teniendo la misma efectividad y no hay nada que falle respecto a ese cambio. Muchas veces los fallos a los que me he tenido que enfrentar han sido físicos, como, por ejemplo, una conexión mal realizada, y son cosas que no se tienen en cuenta al principio del proyecto.

En cuanto al brazo, aun habiendo cambiado las piezas, el diseño de estas no es el óptimo, por lo que habría que rediseñar alguna de ellas para que no se cree desgaste tan rápido. También se podría gestionar la optimización de las conexiones creando una nueva PCB que conecte todos los elementos sin necesidad de tantos jumpers. Además de hacer que sea más difícil el deterioro de estos, también serviría a modo optimización del espacio y de la estética.

También habría que implementarle el movimiento de rotación de la base, no se tuvo en consideración su estudio debido a que como el "Asturiosity" puede gestionar movimientos y giros no era de primordial importancia que rotara el brazo.

Tras haber conseguido el movimiento del robot, el siguiente paso a seguir sería continuar con la implementación al "Asturiosity" con la creación del código que lea los valores recibidos de la Raspberry y su transformación en movimiento.

12. Bibliografía

[1] El historiador, «Astromovil», www.elhistoriador.es. Accedido: 17 de mayo de 2024. [En línea]. Disponible en: https://elhistoriadores.wordpress.com/tag/astromovil/

[2] RAE- y RAE, «Definición Robot». Accedido: 18 de mayo de 2024. [En línea]. Disponible en: https://dle.rae.es/robot

[3] J. Romero, «Robots de primera generación», GoConqr. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://www.goconqr.com/mapamental/19562462/robots-de-primera-generacion

[4] E. B. School, «Clasificación de los robots», Euroinnova Business School. Accedido: 18 de mayo de 2024. [En línea]. Disponible en: https://www.euroinnova.edu.es/blog/clasificacion-de-los-robots

[5] «Robots 3 generación», www.google.com. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://n9.cl/jgyzt

[6] «Características de la Robótica». Accedido: 18 de mayo de 2024. [En línea]. Disponible en: https://aeromaquinados.com/caracteristicas-de-la-robotica/

[7] Esneca, «Brazo Robótico», Esneca. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://www.esneca.com/blog/brazo-robotico-industrias/

[8] «Robot guiado automatizado». Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://lc.cx/sd4Wvb

[9] «Robots humanoides», Fundación Aquae. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://www.fundacionaquae.org/creando-robots-humanoides/

[10] «Cobots», https://www.manutencionyalmacenaje.com/. Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://www.manutencionyalmacenaje.com/Articulos/238751-El-mercado-de-cobots-crecera-un-60-por-ciento-en-2018-en-un-entorno-competitivo-y.html

[11] «Robot industrial imagen». Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://www.klipartz.com/es/sticker-png-hjapp/download

[12] «5.3. Arquitectura de un robot | Control y robótica 4o E.S.O.» Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://angelmicelti.github.io/4ESO/CYR/53_arquitectura_de_un_robot.html

[13] "Control cinemático", apuntes de la materia Robótica Industrial, Departamento de electrónica, Universidad de Ovieo, EPI Gijón, 2024

[14] Á. García López, «TFG -Brazo robótico controlado por gestos humanos», Universidad de Oviedo, 2022.

[15] J. Martinez Becerra, «TFG -Brazo robótico controlado con gestos manuales», Universidad de Oviedo, 2023.

[16] «Servomotores: Héroes Silenciosos de la Tecnología Moderna». Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.cursosaula21.com/que-es-un-servomotor/

[17] «DS3225 servo». Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.amazon.es/Digital-Torque-Impermeable-DS3225-Control/dp/B08BZNSLQF

[18] «Servo Towerpro MG995», Tienda Prometec. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://store.prometec.net/producto/servo-towerpro-mg995/

[19] «Servomotores». Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://alquimetricos.com/conectar-todo/

[20] «PWM». Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://solectroshop.com/es/blog/que-es-pwm-y-como-usarlo--n38

[21] «Tresdeland - Tu mundo 3D», Tr3sDland. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.tr3sdland.com/2011/12/componentes-el-servomotor/

[22] B. de Bakker, «Servomotores con arduino», Makerguides.com. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.makerguides.com/es/servo-arduino-tutorial/

[23] «Modulo PCA9685», turibot.es. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.turibot.es/modulo-pca9685-controlador-servos-16-canales

[24] «PCA9685», AZ-Delivery. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.az-delivery.de/es/products/pca9685-servotreiber

[25] «PCA9685 con Arduino», Naylamp Mechatronics - Perú. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://naylampmechatronics.com/blog/41_tutorial-modulo-controlador-de-servos-pca9685-con-arduino.html

[26] «A000066-datasheet.pdf». Accedido: 24 de mayo de 2024.

[27] «MG995-datasheet.pdf». Accedido: 22 de mayo de 2024. [En línea]. Disponible en: http://pdf1.alldatasheet.com/datasheet-pdf/download/1132435/ETC2/MG995.html

[28] «Servomotores». Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.cursosaula21.com/que-es-un-servomotor/

[29] «¿Qué es PWM y cómo usarlo?» Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://solectroshop.com/es/blog/que-es-pwm-y-como-usarlo--n38

[30] «Modulo PCA9685 controlador servos 16 canales», turibot.es. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://www.turibot.es/modulo-pca9685-controlador-servos-16-canales

[31] «CinematicaRobot.pdf». Accedido: 20 de mayo de 2024. [En línea]. Disponible en: https://www.kramirez.net/Robotica/Material/Presentaciones/CinematicaRobot.pdf

[32] «DS3225-270, Metal Gear Digital Servo with 25KG High Torque». Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://abra-electronics.com/electromechanical/motors/servo-motors/ds3225-270.html

[33] «I2C - Puerto, Introducción, trama y protocolo», HeTPro-Tutoriales. Accedido: 22 de mayo de 2024. [En línea]. Disponible en: https://hetpro-store.com/TUTORIALES/i2c/

13. Anexos

13.1. PLANIFICACIÓN TEMPORAL

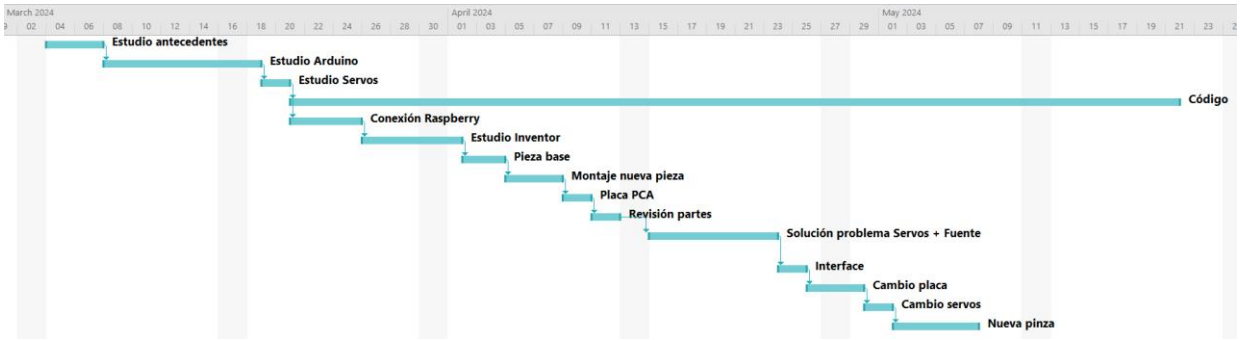

Ilustración 30: Programación temporal

13.2. CÓDIGOS

CÓDIGO BRAZO:

```
#include <Servo.h>

// Declaramos los objetos de la clase Servo para cada servo
Servo servo0;
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo8;
Servo servo9;

// Definimos un arreglo para almacenar las posiciones previas de los
servos
int Prev_servo[10] = {90, 90, 90, 90, 90, 90, 90, 90, 90, 90};

// Función para mover un servo suavemente a una posición específica
void moveServoSmoothly(Servo& servo, int servoPos) {
  int currentPos = servo.read();
  if (currentPos == servoPos) return;

  int step = (currentPos < servoPos) ? 1 : -1;
  for (int pos = currentPos; pos != servoPos; pos += step) {
    servo.write(pos);
    delay(30);
  }
  servo.write(servoPos);
  delay(300);
}
```

```
// Función para mover dos servos juntos a una posición base
void setAnguloBase(int angulo, int prev_angulo) {
  if (angulo < prev_angulo) {
    for (int i = prev_angulo; i > angulo; i -= 3) {
      moveServoSmoothly(servo0, i);
      moveServoSmoothly(servo1, 180 - i);
      delay(20);
    }
  } else if (prev_angulo < angulo) {
    for (int i = prev_angulo; i < angulo; i += 3) {
      moveServoSmoothly(servo0, i);
      moveServoSmoothly(servo1, 180 - i);
      delay(20);
    }
  }
}

void setup() {
  Serial.begin(9600);

  // Inicializamos los pines de los servos y los posicionamos en 90
grados
  servo0.attach(3);
  servo1.attach(5);
  servo2.attach(6);
  servo3.attach(9);
  servo8.attach(10);
  servo9.attach(11);

  servo0.write(90);
  servo1.write(90);
  // servo2.write(90);
  // servo3.write(90);
  // servo8.write(90);
  // servo9.write(90);
  // moveServoSmoothly(servo0, 90);
```

```
  // moveServoSmoothly(servo1, 90);
  delay(500);
  moveServoSmoothly(servo2, 100);
  delay(1000);
  moveServoSmoothly(servo3, 65);
  delay(1000);
  moveServoSmoothly(servo8, 60);
  delay(1000);
  moveServoSmoothly(servo9, 90);
  delay(1000);

  Serial.println("Inicialización completa");
}

void loop() {
  // Definimos variables para el número del servo y la posición del servo
  int servoNumber = 0;
  int servoPos = 0;
  int ang = 180;
  static int prev_ang = 90;

  // Pedimos al usuario que introduzca el número del servo y el ángulo
deseado
  Serial.println("....");
  Serial.println("¿Qué servo quieres mover?");
  delay(7000);

  // Leemos el número del servo desde el puerto serial
  if (Serial.available()) {
    servoNumber = Serial.parseInt();
    Serial.println(servoNumber);
    delay(1000);

    // Pedimos al usuario que introduzca el ángulo deseado para el servo
    Serial.println("¿Qué ángulo quieres?");
    delay(5000);
```

```
    // Leemos el ángulo deseado desde el puerto serial
    if (Serial.available()) {
      servoPos = Serial.parseInt();
      Serial.println(servoPos);
      delay(1000);

      // Verificamos que el ángulo esté en el rango válido y movemos el
servo correspondiente
      if (servoPos >= 0 && servoPos <= 180) {
        switch (servoNumber) {
          case 0:
            setAnguloBase(servoPos, prev_ang);
            prev_ang = servoPos;
            break;
          case 2:
            moveServoSmoothly(servo2, servoPos);
            break;
          case 3:
            moveServoSmoothly(servo3, servoPos);
            break;
          case 8:
            moveServoSmoothly(servo8, servoPos);
            break;
          case 9:
            moveServoSmoothly(servo9, servoPos);
            break;
          default:
            Serial.println("Servo inválido.");
            break;
        }
        Serial.print("Se movió el servo ");
        Serial.print(servoNumber);
        Serial.print(" al ángulo ");
        Serial.println(servoPos);
      } else {
        Serial.println("Ángulo inválido. Debe estar entre 0 y 180
grados.");
```

```
      }
    } else {
      Serial.println("Entrada de ángulo no recibida.");
    }
  } else {
    Serial.println("Entrada de servo no recibida.");
  }
}
```