



**INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ**

***TRABAJO PROFESIONAL***

***Informe Técnico de Residencia Profesional:***

**INGENIERO ELECTRÓNICO**

**PRESENTA:**

**CRISTIAN HERNANDEZ PEREZ**

**TEMA:**

**Desarrollo de un simulador basado en un robot cartesiano de tres grados de libertad**

**MEDIANTE**

***RESIDENCIA PROFESIONAL***

**ASESOR:**

**M.C. ILDEBERTO DE LOS SANTOS RUIZ**

**TUXTLA GUTIÉRREZ, CHIAPAS.**

**DICIEMBRE DE 2015.**

---

## Agradecimiento

Al creador, pues el, todo y mucho más lo hace posible, gracias.

A mis padres patrocinadores directos, de esta pasión.

A todo aquel que cree y hace posible cambiar los  
Paradigmas y prejuicios con invenciones y ciencia

---

## Resumen

En este trabajo se describe el proceso que se hace para controlar un Robot de cuatro grados de libertad y hacerle un simulador grafico del modelo mecánico del robot, es decir que este robot que se controlaba con un “control” alámbrico para cada uno de sus eslabones se modificó por completo, se midieron las piezas, se creó un modelo mecánico virtual del sistema y su movimiento, se diseñó un circuito eléctrico para poder moverlo desde Arduino, se diseña un sistema mecánico para implementar los sensores que tendrá y se implementan todos tal cual sobre el robot. Obviamente se modifica todo el circuito interno del robot, añadiéndole sensores ya mencionados, añadiéndole sistemas de potencia para que pueda ser operado desde serial monitor de Arduino.

Una vez la parte física ha terminado se pasa la parte de programación de sensores y actuadores tomando en cuenta limitaciones de giro, el espacio de trabajo y se realiza la programación adecuada en Arduino, se realizan pruebas y se hacen los análisis correspondientes, posteriormente se hacen todas las partes del robot en SolidWorks para hacer un sistema mecánico representativo al robot. Una vez realizado el sistema mecánico con ayuda de Simulink en Matlab con diagrama a bloques se crea un control de movimiento al sistema mecánico virtual, Solo restaría hacer una interfaz gráfica de Robot físico al robot virtual en tiempo real, teniendo así un sistema de monitoreo en tiempo real, cabe mencionar que por razones de tiempo esta última no se pudo concretar, así que no hubo interfaz gráfica en tiempo real.

---

## Índice

1	Informe Final de Residencia Profesional.....	1
1.1	Introducción.....	1
1.2	Datos generales de la empresa.....	2
1.2.1	Nombre de la empresa y razón social.....	2
1.2.2	Domicilio de la empresa.....	2
1.2.3	Giro.....	2
1.2.4	Organigrama.....	3
1.2.5	Breve descripción de la empresa.....	4
1.3	Justificación del proyecto.....	4
1.4	Objetivos generales y específicos.....	6
1.4.1	Objetivos generales.....	6
1.4.2	Objetivos específicos.....	6
1.5	Características del área en que participó.....	7
1.6	Alcances y limitaciones.....	7

---

<b>1.7</b>	<b>Fundamento o Marco Teórico.....</b>	<b>9</b>
<b>1.7.1</b>	<b>Antecedentes del problema.....</b>	<b>9</b>
<b>1.7.2</b>	<b>Robot industrial.....</b>	<b>10</b>
<b>1.7.3</b>	<b>Estructura de los robots industriales .....</b>	<b>13</b>
<b>1.7.4</b>	<b>Robots manipuladores.....</b>	<b>16</b>
<b>1.7.5</b>	<b>Grados de libertad en robots manipuladores.....</b>	<b>17</b>
<b>1.7.6</b>	<b>Configuraciones morfológicas y parámetros</b> <b>Característicos de los robots industriales.....</b>	<b>19</b>
<b>1.7.7</b>	<b>Robot Steren modelo k-680.....</b>	<b>23</b>
<b>1.7.8</b>	<b>Potenciómetro.....</b>	<b>24</b>
<b>1.7.9</b>	<b>Potenciómetro Multivoltas 10k Bourns 3590s-2-103 lineal.....</b>	<b>26.</b>
<b>1.7.10</b>	<b>Arduino.....</b>	<b>27</b>
<b>1.7.11</b>	<b>Puentes o jumper.....</b>	<b>30</b>
<b>1.7.12</b>	<b>Puente h.....</b>	<b>31</b>
<b>1.7.13</b>	<b>SolidWorks.....</b>	<b>33</b>
<b>1.7.14</b>	<b>Matlab.....</b>	<b>38</b>
<b>1.8</b>	<b>Procedimientos y descripción de las actividades realizadas.....</b>	<b>41</b>
<b>1.8.1</b>	<b>Diagrama del hardware.....</b>	<b>43</b>
<b>1.8.2</b>	<b>Pruebas del Potenciómetro 10K Bourns 3590S-2-103 Lineal.....</b>	<b>44</b>
<b>1.8.2.1</b>	<b>Circuito.....</b>	<b>44</b>
<b>1.8.2.2</b>	<b>Esquemático.....</b>	<b>45</b>
<b>1.8.2.3</b>	<b>Código y programación en</b> <b>Arduino.....</b>	<b>46</b>

---

<b>1.8.2.4</b>	Monitor serial de Arduino.....	47
<b>1.8.3</b>	Construcción de robot Steren modelo k-680.....	48
<b>1.8.4</b>	Construcción de robot Steren modelo K-680 en SOLIDWORKS® modelado mecánico en 3D.....	50
<b>1.8.4.1</b>	Primer eslabón.....	51
<b>1.8.4.2</b>	Segundo eslabón.....	52
<b>1.8.4.3</b>	Tercer eslabón.....	53
<b>1.8.4.4</b>	Cuarto eslabón.....	54
<b>1.8.4.5</b>	Ensamble final.....	55
<b>1.8.5</b>	Colocación de los sensores sobre cada codo el robot.....	55
<b>1.8.5.1</b>	Identificación de cada eslabón para robot Steren modelo k-680.....	56
<b>1.8.5.2</b>	Identificación de los puntos rotativos ideales para colocar sensores.....	57
<b>1.8.5.3</b>	Limitaciones en el espacio de trabajo.....	58
<b>1.8.5.4</b>	Adaptación y colocación de sensores.....	59
<b>a)</b>	Transmisión por levas.....	59
<b>b)</b>	Cintura y acoplamiento de Transmisión por levas.....	60
<b>c)</b>	Transmisión directa.....	61
<b>d)</b>	Hombro y acoplamiento de sensor.....	61
<b>e)</b>	Codo y acoplamiento de sensor.....	62

---

<b>f)</b>	Acoplamiento de sensor a muñeca y pinza.....	63
<b>1.8.5.5</b>	Montaje físico de sensores.....	64
<b>a)</b>	Montaje de sensor en cintura.....	65
<b>b)</b>	Montaje de sensor en hombro.....	65
<b>c)</b>	Montaje de sensor en codo.....	66
<b>d)</b>	Montaje de sensor sobre muñeca y pinza.....	67
<b>1.8.6</b>	Diagrama de conexiones.....	68
<b>1.8.7</b>	Creación de piezas en solidwork para modelado de Robot y simulación en Matlab.....	70
<b>a)</b>	Primer eslabón.....	70
<b>b)</b>	Segundo eslabón.....	71
<b>c)</b>	Tercer eslabón.....	72
<b>d)</b>	Cuarto eslabón.....	72
<b>e)</b>	Ensamble final.....	73
<b>1.8.8</b>	Diagrama de software.....	74
<b>1.8.9</b>	Programación de todos los sensores en conjunto.....	75
<b>1.8.9.1</b>	Programación de control manual por medio de Arduino.....	76
<b>1.8.9.2</b>	Programación final.....	77
<b>1.8.10</b>	Desarrollo de simulador del robot en Matlab.....	82
<b>1.9</b>	Evaluación o impacto económico, social o tecnológico.....	92
<b>1.10</b>	Resultados Obtenidos.....	92

---

---

1.11	Conclusiones y recomendaciones.....	93
1.12	Bibliografía.....	93



# 1. Reporte de la Residencia Profesional

## 1.1. Introducción

Las primeras máquinas simples sustituían una forma de esfuerzo en otra forma que fueran manejadas por el ser humano, tal como levantar un peso pesado con sistema de poleas o con una palanca. Posteriormente las máquinas fueron capaces de sustituir la energía humana o animal por formas naturales de energía renovable, tales como el viento, las mareas, o un flujo de agua, con las cuales crearon diferentes tipos de molinos. Todavía después, algunas formas de automatización fueron controladas por mecanismos de relojería o dispositivos similares utilizando algunas formas de fuentes de poder artificiales -algún resorte, un flujo canalizado de agua o vapor para producir acciones simples y repetitivas, tal como figuras en movimiento, estos abecés creaban música, o eran simples juegos. Dichos dispositivos caracterizaban a figuras humanas o parte de ellos, y fueron conocidos como autómatas y datan posiblemente desde el año 300 a.c.

Actualmente en las industrias esos autómatas han evolucionado hasta convertirse en verdaderas obras de arte en ingeniería, capaces de realizar tareas específicas y en muchos casos industriales las tareas son repetitivas formando parte de un proceso.

Retomando lo anterior, es muy importante para la comunidad de estudiantes de ingenierías (Ingeniería de procesos, Ingeniería de proyectos, Ingeniería de diseño, Ingeniería de automatización, Ingeniería electrónica, Ingeniería mecatrónica, Ingeniería electromecánica y otros) conocer la caracterización de un robot industrial, en este caso se caracterizará y se creará un simulador de un robot de 4 grados de libertad.

---

## **1.2. Datos generales de la empresa**

Con su Lema “Pensar bien, para vivir mejor” fue fundada el 14 de abril de 1578 hace 437 años. EL campus Abarca un dimensión de Terreno de: 1 010 000 m<sup>2</sup>, y sus construcciones logran los: 214 000 m<sup>2</sup>. La Mascota son los “Lobos de la BUAP”, Colores (oficiales): azul cielo y azul marino.

### **1.2.1.Nombre de la empresa y/o razón social**

La Benemérita Universidad Autónoma de Puebla (BUAP) es una institución de educación superior pública mexicana, autónoma y descentralizada, cuya sede es la ciudad de Puebla, capital del estado del mismo nombre.

### **1.2.2.Domicilio de la empresa**

Se localiza en 4 Sur #104 Col. Centro C.P. 72000 en Puebla de Zaragoza, Puebla México. Sus coordenadas en el globo son: 19°00'03"N 98°12'02"O Coordenadas: 19°00'03"N 98°12'02"O (mapa).

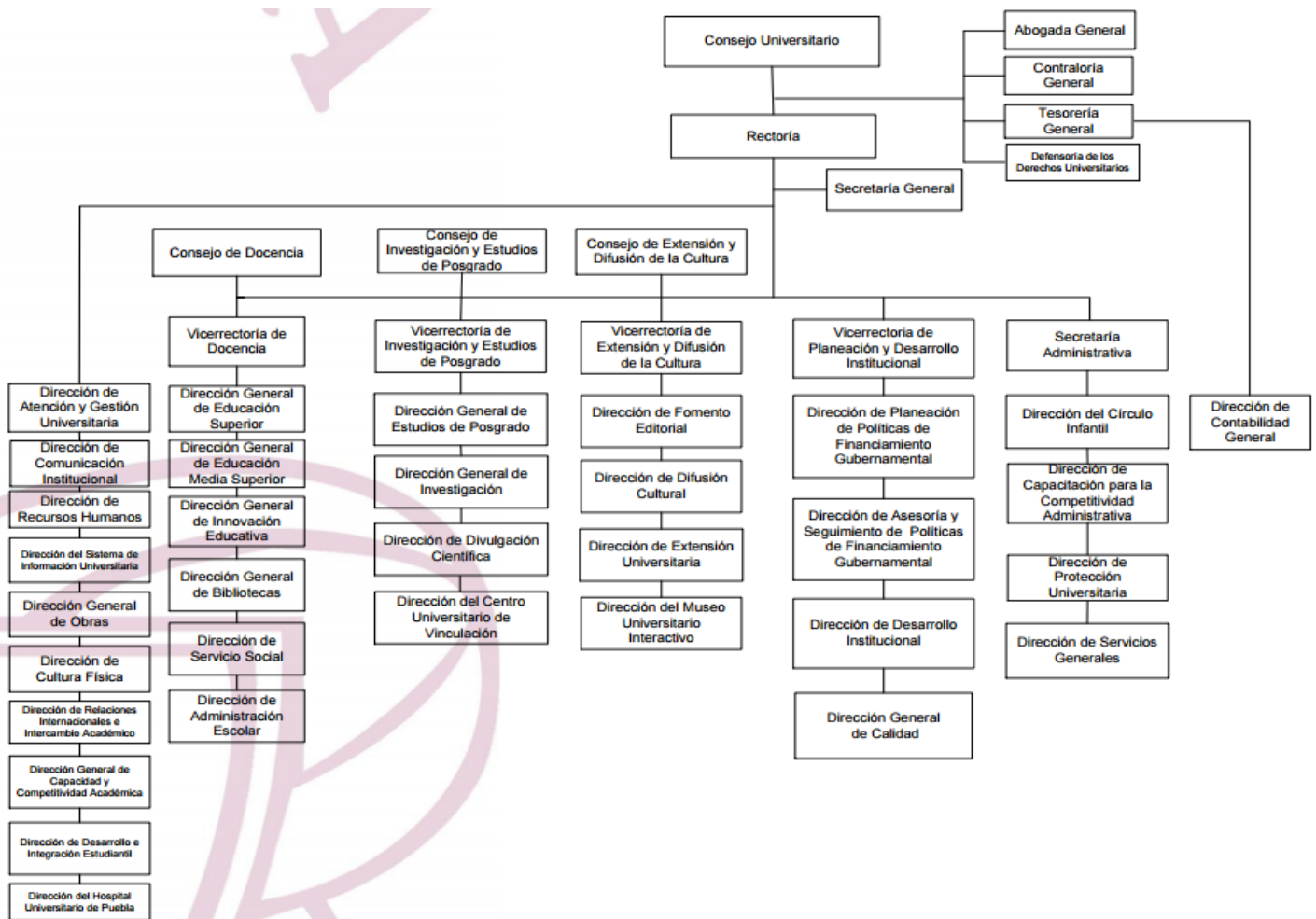
### **1.2.3.Giro**

La Benemérita Universidad Autónoma de Puebla es una institución educativa pública y autónoma consolidada a nivel nacional, comprometida con la formación integral de profesionales y ciudadanos críticos y reflexivos en los niveles de educación media superior, superior y posgrado, que son capaces de generar, adaptar, recrear, innovar y aplicar conocimientos de calidad y pertinencia social.

La Universidad fomenta la investigación, la creación y la divulgación del conocimiento, promueve la inclusión, la igualdad de oportunidades y la vinculación; coadyuva como comunidad del conocimiento al desarrollo del arte, la cultura, la solución de problemas económicos, ambientales, sociales y políticos de la región y del país, bajo una política

de transparencia y rendición de cuentas, principios éticos, desarrollo sustentable, en defensa de los derechos humanos, de tolerancia y honestidad; contribuyendo a la creación de una sociedad proactiva, productiva, justa y segura.

### 1.2.4. Organigrama



---

### **1.2.5. Breve descripción de la empresa**

La Benemérita Universidad Autónoma de Puebla es líder en el país y cuenta con posicionamiento internacional. La calidad y pertinencia de su oferta educativa y servicios académicos están sustentadas en una planta académica sólida y reconocida, un modelo educativo pertinente, flexible y enfocado en el estudiante, un desarrollo científico y tecnológico, una amplia influencia en la cultura y las artes, una estructura académica que funciona en redes de cooperación y colaboración nacionales e internacionales, una estructura administrativa y de gestión ágil, funcional, acreditada y flexible que apoya el quehacer académico, bajo una política de transparencia, rendición de cuentas y desarrollo sustentable.

### **1.3. Justificación del proyecto**

El monitoreo de maquinaria es la observación del curso de uno o varios parámetros para la detección de problemas. cada vez en las industrias el mantenimiento cobra un sentido de amplia relevancia, debido a muchos factores como el acelerado desarrollo del mundo contemporáneo, donde se exige cada vez más en aspectos tales como calidad, tiempo de proceso, tiempo de entrega etc. Es por ello que la industria actual no puede atrasarse y cada vez busca alternativas de solución para satisfacer sus necesidades.

Pero no solo se queda la búsqueda en el mantenimiento, sino que profundiza en el concepto y analiza posibilidades de evitar las fallas, y es por ello que se desarrollaron los conceptos de mantenimiento predictivo y preventivo, y por tanto, también se crearon, y se siguen creando, diversos métodos, herramientas, técnicas para facilitar estos estudios.

Por ello el monitoreo en tiempo real es algo sumamente importante, con un buen monitoreo de los equipos de trabajo disminuimos las probabilidades de fallas, ya que es

---

---

mucho más práctico visualizar el problema detectarlo y posteriormente dar el ajuste o mantenimiento necesario a la máquina y al software de la misma, si así se requiere.

Hoy en día el hecho de que existan personas que se encarguen de supervisar en las mismas industrias es una actividad más que destacada, sin importar el ramo de la industria ni a lo que se dedique cada una de ellas siempre es bueno que existan personas que puedan observar y de alguna forma también guiar las máquinas.

Los estudiantes de ingenierías o los ya ingenieros que laboran en industrias o plantas de procesos de manufactura deben conocer el funcionamiento de las máquinas, es parte del conocimiento vital si se aspira a una buena labor en ese ámbito.

En este caso se hace un proyecto en el cual se introduce al ámbito del control de robots industriales

Las interfaces hombre-máquina (HMI) o interfaces hombre-computadora (CHI) suelen emplearse en empresas e industrias donde las máquinas hacen gran parte del trabajo, Las interfaces HMI o CHI se emplean por lo regular para comunicarse con los PLCs y con otras computadoras, para labores tales como introducir y monitorear temperaturas o presiones para controles automáticos o respuesta a mensajes de alarma. El personal de servicio que monitorea y controla estas interfaces se conoce comúnmente como ingenieros de estación y el personal que opera directamente en la HMI o SCADA (Sistema de Control y Adquisición de Datos) es conocido como personal de operación.

Existen multitud de empresas industriales en las cuales hacen uso de robots para ciertas tareas específicas y las nuevas generaciones de ingenieros deben conocer el funcionamiento de una manera general de los robots industriales.

Este proyecto está inclinado a los estudiantes y próximos egresados aspirantes a trabajar en plantas industriales, donde comprender de una manera más profunda un robot industrial y la caracterización de sus partes vitales, como también los fundamentos para el control del mismo y su propio monitoreo es muy importante. Los

---

---

resultados de este proyecto sirven como base para distintas soluciones de proyectos futuros.

## **1.4. Objetivos generales y específicos**

### **1.4.1. Objetivos generales**

Implementar sensores y una programación de control en el robot de marca STEREN modelo K-680 para dar movimiento la mismo, al igual que crear un simulador basado en un diseño del robot STEREN K-680 en SolidWorks y posteriormente controlarlo con ayuda de Simulink Matlab.

### **1.4.2. Objetivos específicos**

- Diseñar un sistema con ayuda de sensores para cada grado de libertad del robot, permitiendo así tener una relación del censado con el Angulo de movimiento de cada grado de libertad (cada eslabón) del robot STEREN K-680.
- Diseñar un circuito con Arduino capaz de leer la información de los sensores propuestos para conocer la posición de cada eslabón
- En base a la plataforma y tarjeta Arduino diseñar y crear un sistema de potencia que permita controlar los actuadores que le dan movimiento al robot.
- Si es necesario crear en SolidWorks acoples Sensor-Eslabón para posteriormente imprimir las piezas (impresora 3D) para no dañar la estructura del robot.
- Diseñar un programa en Arduino capaz de leer y procesar la información censada en los “sensores” que se colocaron previamente en cada grado de libertad del robot, para saber la posición en la que están cada eslabón.
- Realizar las piezas necesarias para el movimiento mecánico en SolidWorks
- Hacer el ensamble de las piezas en solidwork y verificar que tengan el mismo movimiento que el Robot K-680.

- 
- Exportar el diseño del robot diseñado en solidwork a Matlab para su posterior simulación
  - Crear una programación en Simulink que simule el movimiento de cada eslabón
  - Controlar el movimiento del robot físico STEREN K-680 y el robot simulado en Matlab.

### **1.5. Características del área en que participé.**

Las características del área son de tipo técnico e “investigación”, es decir la participación y el aprendizaje autodidacta resaltan mucho, dado por la magnitud del proyecto trabajé en un cubo que se encuentra en el edificio de posgrado de la facultad de ingenierías de la electrónica, cabe mencionar que este espacio me fue otorgado por el Maestro en ciencias e investigador Sánchez Sánchez Pablo quien es asesor externo, también estuve en otros edificios ya que necesitaba aprender ciertos conocimientos que me ayudarían a mi labor, como lo fue asistir a clases del Software de modelado mecánico virtual de SolidWorks (en licenciatura) y clases de automatización industrial (en maestría).

### **1.6. Alcance y limitaciones.**

Dentro de lo necesario para este proyecto era conocer sobre automatización, y creación de piezas mecánicas en algún simulador, Dominar conocimiento en Control, al igual que contar con los recursos económicos necesarios para solventar gastos para comprar materiales que se utilizan en prácticas de electrónica, uno de los inconvenientes para empezar las practicas fue adquirir todo lo necesario para poder empezar el aprendizaje el software para la creación de piezas, es decir para SolidWorks es recomendable que el equipo en el cual estará instalado cuente con 4gb de memoria ram y cuente con procesador i5

---

cuando poco, ya que las tareas eran muchas, y tener una mala máquina de trabajo es sinónimo de lentitud para terminar, cuando las tareas y prácticas eran demasiadas.

Otra de las limitaciones era el tiempo, ya que idealmente el proyecto constaba de un simulador en el cual existiera la interfaz gráfica entre el robot físico controlado por Arduino y el modelo mecánico animado controlado por Simulink de Matlab, y para todo esto es necesario un sistema de control y las ecuaciones del modelo dinámico del robot, los dos últimos no pudieron concretarse ya que se necesitó más tiempo.



---

## 1.7. Fundamento o marco teórico

### 1.7.1. Antecedentes del problema

En las empresas se necesita tener el control total de las maquinas que se están utilizando

En lo que respecta al monitoreo industrial que tiene que ver con la calidad de los productos, debemos decir que es uno de los más importantes, más que nada para todos aquéllos productos que tienden a comerse. Es vital que un grupo de personas que esté acorde a la importancia de cada empresa, este a cargo de este sector, ya que cuando hablamos de alimentos, nos estamos refiriendo a un elemento importante que va a ser consumido por un gran número de personas, y el hecho de que los mismos se encuentren en perfecto estado desde que se fabrican hasta que llega a la mano del consumidor, es vital que esta cadena este perfectamente monitoreada porque de no ser así, podrían existir grandes problemas. Si nos basamos en la historia de las industrias, ha sucedido muchas veces que por diferentes problemas, se genere un lote de productos dañados o contaminados.

Lo anterior es un ejemplo de muchos problemas que puede haber en el monitoreo de máquinas industriales

Este tipo de problemas realmente son muy importantes y repercuten de manera negativa tanto en la gente como en la empresa misma que pierde cierta credibilidad al vender al consumidor productos de baja calidad o en casos extremos defectuosos e inservibles.

Cabe mencionar que en la compañía de Tesla motors que se dedica a fabricar autos eléctricos, se utilizan robots para ensamblar las piezas, los ingenieros programan cada robot a mano para cumplir sus tareas específicas, los últimos robots de la cadena de producción como los que se muestran en la imagen 1,

---

estos son programados como soldadores multipunto para unir la carrocería con el chasis, en este caso el trabajo de un robot es tan bueno como el que lo entrena (programa), pero cuando ocurre un error en el modo de trabajar de un robot toda la cadena de producción se detiene, como ocurrió meses antes de las primeras entregas de los autos modelo S de Tesla motors, pero gracias al monitoreo constante se detectó el problema y se reajustaron las piezas mecánicas del robot soldador que ocasionaban las fallas.



Imagen 1. Robot soldador utilizado en tesla motors

### 1.7.2. Robot industrial

Entre los robots considerados de más utilidad en la actualidad se encuentran los robots industriales o manipuladores. Existen ciertas dificultades a la hora de establecer una definición formal de lo que es un robot industrial. La primera de ellas surge de la diferencia conceptual entre el mercado japonés y el euro-americano de lo que es un robot y lo que es un manipulador. Así, mientras que para los japoneses un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación, el mercado occidental es más restrictivo, exigiendo una mayor complejidad, sobre todo en lo relativo al control. En segundo lugar, y centrándose ya en el concepto occidental, aunque existe una idea común acerca de lo que es un robot industrial, no es fácil

---

ponerse de acuerdo a la hora de determinar una definición formal. Además, la evolución de la robótica ha ido obligando a diferentes actualizaciones de su definición.

- La definición más comúnmente aceptada posiblemente sea la de la Asociación de Industrias de Robótica (RIA, Robotic Industry Association), según la cual:

*"Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas"*

- Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO) que define al robot industrial como:

*"Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas"*

- Se incluye en esta definición la necesidad de que el robot tenga varios grados de libertad. Una definición más completa es la establecida por la Asociación Francesa de Normalización (AFNOR), que define primero el manipulador y, basándose en dicha definición, el robot:

Manipulador: *mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.*

---

Robot: *manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.*

- Por último, la Federación Internacional de Robótica (IFR, International Federation of Robotics) distingue entre robot industrial de manipulación y otros robots:

*"Por robot industrial de manipulación se entiende una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento"*

En esta definición se debe entender que la reprogramabilidad y la multifunción se consiguen sin modificaciones físicas del robot.

Común en todas las definiciones anteriores es la aceptación del robot industrial como un brazo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo. Un sistema robotizado, en cambio, es un concepto más amplio. Engloba todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y que pueden incorporar o no a uno o varios robots, siendo esto último lo más frecuente.

---

### 1.7.3. Estructura de los robots industriales

Un manipulador robótico consta de una secuencia de elementos estructurales rígidos, denominados enlaces o eslabones, conectados entre sí mediante juntas o articulaciones, que permiten el movimiento relativo de cada dos eslabones consecutivos como se muestra en la imagen 2.

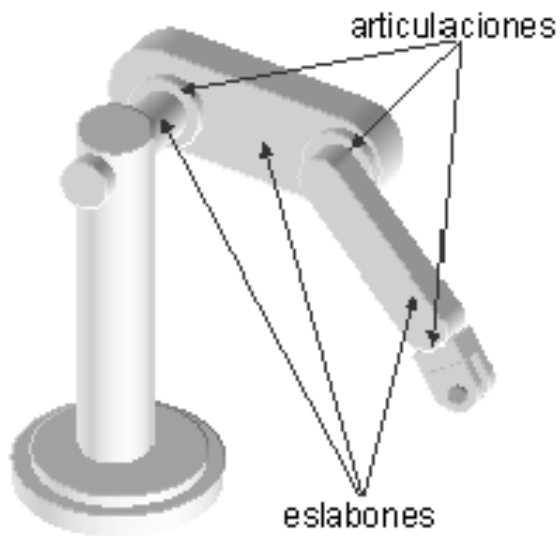


Imagen 2. Elementos estructurales de un robot industrial

Una articulación puede ser:

- **Lineal** (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior.
- **Rotacional**, en caso de que un eslabón gire en torno a un eje solidario al eslabón anterior.

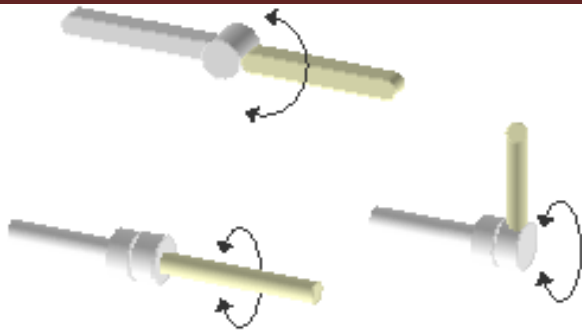


Imagen 3. Distintos tipos de articulaciones de un robot: a) lineal, b) rotacionales

El conjunto de eslabones y articulaciones se denomina cadena cinemática. Se dice que una cadena cinemática es abierta si cada eslabón se conecta mediante articulaciones exclusivamente al anterior y al siguiente, exceptuando el primero, que se suele fijar a un soporte, y el último, cuyo extremo final queda libre. A éste se puede conectar un elemento terminal o actuador final: una herramienta especial que permite al robot de uso general realizar una aplicación particular, que debe diseñarse específicamente para dicha aplicación: una herramienta de sujeción, de soldadura, de pintura, etc. El punto más significativo del elemento terminal se denomina punto terminal (PT). En el caso de una pinza, el punto terminal vendría a ser el centro de sujeción de la misma.

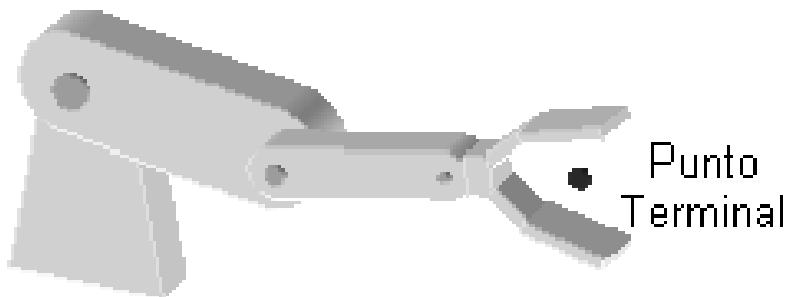


Imagen 4. Punto terminal de un manipulador

Los elementos terminales pueden dividirse en dos categorías:

- **pinzas** (*gripper*)
- **herramientas**

---

## a) pinzas

Las pinzas se utilizan para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot. Hay una diversidad de métodos de sujeción que pueden utilizarse, además de los métodos mecánicos obvios de agarre de la pieza entre dos o más dedos. Estos métodos suplementarios incluyen el empleo de casquillos de sujeción, imanes, ganchos, y cucharas.



Imagen 5. Pinzas (gripper)

## B) herramienta

Una herramienta se utiliza como actuador final en aplicaciones en donde se exija al robot realizar alguna operación sobre la pieza de trabajo. Estas aplicaciones incluyen la soldadura por puntos, la soldadura por arco, la pintura por pulverización y las operaciones de taladro. En cada caso, la herramienta particular está unida a la muñeca del robot para realizar la operación.



Imagen 6. Herramientas taladro (Izquierda), Pintura por pulverización (Derecha).

### 1.7.4. Robots manipuladores

A los manipuladores robóticos se les suele denominar también brazos de robot por la analogía que se puede establecer, en muchos casos, con las extremidades superiores del cuerpo humano.



Imagen 7. Brazo robot



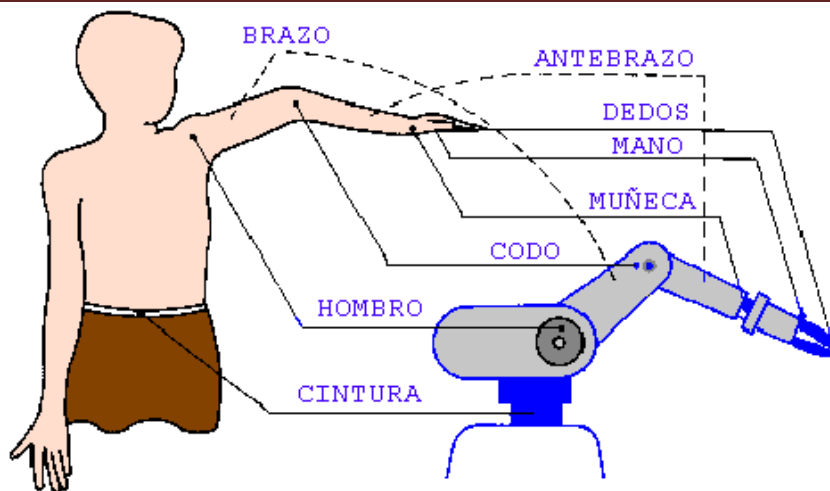


Imagen 8. Semejanza de un brazo manipulador con la anatomía humana

### 1.7.5. Grados de libertad en robots manipuladores

Se denomina grado de libertad (g.d.l.) a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación en el espacio de sus elementos). Normalmente, en cadenas cinemáticas abiertas, cada par eslabón-articulación tiene un solo grado de libertad, ya sea de rotación o de traslación. Pero una articulación podría tener dos o más g.d.l. que operan sobre ejes que se cortan entre, tal sí como se muestra en la imagen 9.

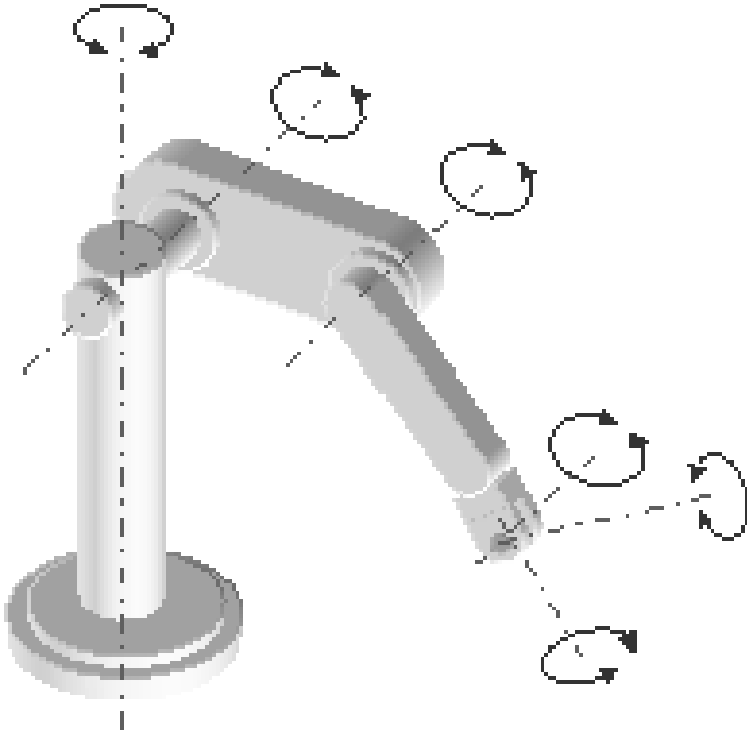


Imagen 9. Distintos grados de libertad de un brazo de robot

Para describir y controlar el estado de un brazo de robot es preciso determinar:

- La posición del punto terminal (o de cualquier otro punto) respecto de un sistema de coordenadas externo y fijo, denominado el sistema mundo.
- El movimiento del brazo cuando los elementos actuadores aplican sus fuerzas y momentos.

El análisis desde el punto de vista mecánico de un robot se puede efectuar atendiendo exclusivamente a sus movimientos (estudio cinemático) o atendiendo además a las fuerzas y momentos que actúan sobre sus partes (estudio dinámico) debidas a los elementos actuadores y a la carga transportada por el elemento terminal.

---

## 1.7.6. Configuraciones morfológicas y parámetros característicos de los robots industriales

Según la geometría de su estructura mecánica, un manipulador puede ser:

- **Cartesiano**, cuyo posicionamiento en el espacio se lleva a cabo mediante articulaciones lineales.
- **Cilíndrico**, con una articulación rotacional sobre una base y articulaciones lineales para el movimiento en altura y en radio.
- **Polar**, que cuenta con dos articulaciones rotacionales y una lineal.
- **Esférico** (o de brazo articulado), con tres articulaciones rotacionales.
- **Mixto**, que posee varios tipos de articulaciones, combinaciones de las anteriores. Es destacable la configuración SCARA (Selective Compliance Assembly Robot Arm)
- **Paralelo**, posee brazos con articulaciones prismáticas o rotacionales concurrentes.

Los principales parámetros que caracterizan a los robots industriales son:

- 
- **Número de grados de libertad.** Es el número total de grados de libertad de un robot, dado por la suma de g.d.l. de las articulaciones que lo componen. Aunque la mayoría de las aplicaciones industriales requieren 6 g.d.l., como las de soldadura, mecanizado y almacenamiento, otras más complejas requieren un número mayor, tal es el caso de las labores de montaje.
  
  - **Espacio de accesibilidad o espacio (volumen) de trabajo.** Es el conjunto de puntos del espacio accesibles al punto terminal, que depende de la configuración geométrica del manipulador. Un punto del espacio se dice totalmente accesible si el PT puede situarse en él en todas las orientaciones que permita la constitución del manipulador y se dice parcialmente accesible si es accesible por el PT pero no en todas las orientaciones posibles. En la figura inferior se aprecia el volumen de trabajo de robots de distintas configuraciones.
  
  - **Capacidad de posicionamiento del punto terminal.** Se concreta en tres magnitudes fundamentales: resolución espacial, precisión y repetibilidad, que miden el grado de exactitud en la realización de los movimientos de un manipulador al realizar una tarea programada.
  
  - **Capacidad de carga.** Es el peso que puede transportar el elemento terminal del manipulador. Es una de las características que más se tienen en cuenta en la selección de un robot dependiendo de la tarea a la que se destine.

- 
- **Velocidad.** Es la máxima velocidad que alcanzan el PT y las articulaciones.

Configuración geométrica	Estructura cinemática	Espacio de trabajo	Ejemplo
<p>cartesianos</p>  <p>tipo cantilever</p> <p>tipo pórtico</p>			
<p>cilíndrico</p> 			
<p>polar</p> 			
<p>esférico</p> 			
<p>SCARA</p> 			
<p>paralelo</p> 			

---

### 1.7.7. Robot Steren modelo k-680

El Robot de marca STEREN Kit es un brazo mecánico para armar, con control remoto alámbrico kit para estudiantes o hobbistas. El equipo contiene un paquete de componentes electrónicos con instructivo para armar un brazo mecánico. Puede enviar la señal de abrir o cerrar la tenaza, sostener objetos ligeros (de hasta 100 gr), bajar o subir el brazo y moverlo hacia arriba, abajo, derecha o izquierda, a través del control remoto integrado.



Imagen 10. Robot STEREN K-680 (Izquierda), control alámbrico que controla el movimiento hacia ambas direcciones de cada eslabón y la pinza (Derecha).

#### Características técnicas

- Abertura máxima de la tenaza: 1,77 pulgadas
- Movimiento vertical de la tenaza: 120°
- Movimiento vertical de la parte superior del brazo: 120°

- 
- Movimiento vertical de la parte inferior del brazo:  $180^\circ$
  - Movimiento horizontal de la base del brazo:  $270^\circ$
  - Alimentación: 6 Vcc (4 pilas tipo D)

### 1.7.8. Potenciómetro

Un potenciómetro es un resistor cuyo valor de resistencia es variable. De esta manera, indirectamente, se puede controlar la intensidad de corriente que fluye por un circuito si se conecta en paralelo, o la diferencia de potencial al conectarlo en serie.

Normalmente, los potenciómetros se utilizan en circuitos de poca corriente. Para circuitos de corrientes mayores, se utilizan los reóstatos, que pueden disipar más potencia.



Imagen 11. Potenciómetro lineal (izquierda), potenciómetro rotativo (derecha)

#### a) Tipos de potenciómetros

Según su aplicación se distinguen varios tipos:



---

**Potenciómetros de Mando.** Son adecuados para su uso como elemento de control en los aparatos electrónicos. El usuario acciona sobre ellos para variar los parámetros normales de funcionamiento. Por ejemplo, el volumen de una radio.

**Potenciómetros de ajuste.** Controlan parámetros pre ajustado, normalmente en fábrica, que el usuario no suele tener que retocar, por lo que no suelen ser accesibles desde el exterior. Existen tanto encapsulados en plástico como sin cápsula, y se suelen distinguir potenciómetros de ajuste vertical, cuyo eje de giro es vertical, y potenciómetros de ajuste horizontal, con el eje de giro paralelo al circuito impreso.

Según la ley de variación de la resistencia  $R = \rho(\theta)$ :

**Potenciómetros lineales.** La resistencia es proporcional al ángulo de giro. Denominados con una letra **B** según la normativa actual (anteriormente A).

**Logarítmicos.** La resistencia depende logarítmicamente del ángulo de giro. Denominados con una letra **A** según normativa actual (anteriormente B).

**Senoidales.** La resistencia es proporcional al seno del ángulo de giro. Dos potenciómetros senoidales solidarios y girados 90° proporcionan el seno y el coseno del ángulo de giro. Pueden tener topes de fin de carrera o no.

**Anti logarítmicos.** Generalmente denominados con una letra **F**.

En los potenciómetros impresos la ley de resistencia se consigue variando la anchura de la pista resistiva, mientras que en los bobinados se ajusta la curva a tramos, con hilos de distinto grosor.

**Potenciómetros Multivueeltas.** Para un ajuste fino de la resistencia existen potenciómetros Multivueeltas, en los que el cursor va unido a un tornillo desmultiplicado, de modo que para completar el recorrido necesita varias vueltas del órgano de mando.

---

## 1.7.9.POTENCIÓMETRO MULTIVUELTAS 10K BOURNS 3590S-2-103 LINEAL

### Descripción de potenciómetro Multivueeltas 10K Bourns 3590S-2-103 Lineal

Potenciómetro de precisión de 10 vueltas marca Bourns modelo 3590s-2-103, resistencia 10 k ohms, lineal, precisión .25%, potencia 2w, 5% de tolerancia, eje metálico. El 3590S - 2-103L es un potenciómetro de precisión con casquillo de montaje, el cuerpo totalmente sellado y los pines de PCB. El potenciómetro de precisión está diseñado para su uso en aplicaciones HMI (Interface Hombre Maquina). Plástico o metal Eje y bujes, Soldadura agarraderas o PC prendedores, Cuerpo completamente sellado



Imagen 12. Potenciómetro Multivueeltas 3590S-2-103 Lineal

### Detalles de Potenciómetro Multivueeltas 10K Bourns 3590S-2-103 lineal

- Tipo de AJUSTE : Ranura de destornillador
- Diámetro del Eje : 6.34mm
- Número de Pandillas : 1

- 
- Número de Giros : 10
  - Potencia nominal: 2W
  - Potenciómetro , Montaje : Buje
  - Rango de Producto : Serie 3590S
  - Tolerancia de Resistencia :  $\pm 5 \%$
  - Resistencia de la Pista : 10kohm
  - Sustancia Extremadamente Preocupante ( SEP ) : No SVHC ( 15 - Jun- 2015)
  - Longitud del eje : 20.63mm
  - Tensión nominal: 450VAC
  - Variación : Lineal
  -

### 1.7.10. ARDUINO

**Arduino** es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida.



Imagen 13. Placa de Arduino uno

---

## a) Hardware

Las placas Arduino son pequeños ordenadores con los que puedes leer información de diferentes sensores, así como controlar luces, motores y muchas otras cosas. La gran mayoría de los sistemas que nos rodean son ordenadores de diferentes tamaños. Los ordenadores no necesitan tener teclado ni pantalla. Hay ordenadores en el microondas de la cocina, dentro de los ascensores para detectar qué botón pulsas y, en los coches. Hay más de 70 ordenadores similares a Arduino... hay ordenadores por todas partes.

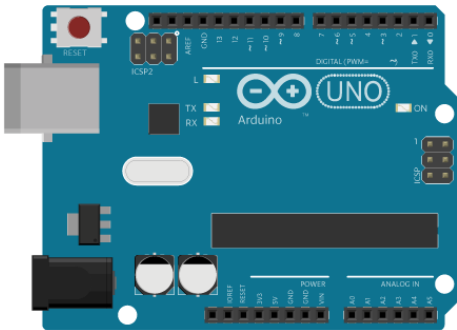


Imagen 14. Placa Arduino

Las placas Arduino se conectan a tu ordenador utilizando un cable USB, al igual que cualquier otro periférico, como la impresora, el teclado o incluso, un mando de videojuegos. Arduino necesita estar conectado al ordenador a través del cable USB para cargar un programa. El cable USB sirve también para suministrar energía a la placa, pero también puedes alimentarla usando una fuente de energía externa, como una batería o un transformador apropiado.

## b) Software

---

Puesto que Arduino, a diferencia del ordenador que usas normalmente, no tiene pantalla ni teclado, se necesita un programa externo ejecutado en otro ordenador para poder escribir programas para la placa Arduino. Éste software es lo que llamamos Arduino IDE. IDE significa “Integrated Development Environment” (Entorno de Desarrollo Integrado), y es un término común para llamar a este tipo de desarrollo de software. Escribes tu programa en el IDE, lo cargas en el Arduino, y el programa se ejecutará en la placa.

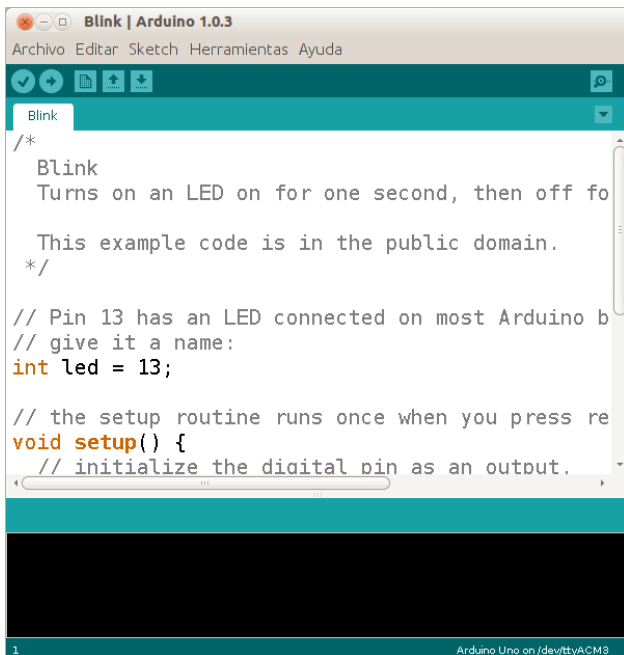


Imagen 15. ID de Arduino

El IDE de Arduino es muy sencillo y parecido a Processing. Hay una sencilla barra de herramientas que puedes utilizar para:

- Verificar si tu programa va a funcionar.
- Cargar el programa a la placa de Arduino.
- Crear un programa nuevo.
- Abrir un programa.
- Guardar el programa en el disco duro del ordenador.

---

(En la parte derecha de la barra de herramientas se encuentra el Monitor Serial) abre una ventana de comunicación con la placa Arduino.

### **c) aplicaciones**

La plataforma Arduino ha sido usada como base en diversas aplicaciones electrónicas:

- Xoscillo: Osciloscopio de código abierto.<sup>16</sup>
- Equipo científico para investigaciones.<sup>17</sup>
- Arduinome: Un dispositivo controlador MIDI.<sup>18</sup>
- OBduino: un económetro que usa una interfaz de diagnóstico a bordo que se halla en los automóviles modernos.
- SCA-ino: Sistema de cómputo automotriz capaz de monitorear sensores como el TPS, el MAP y el O2S y controlar actuadores automotrices como la bobina de ignición, la válvula IAC y aceleradores electrónicos.
- Humane Reader: dispositivo electrónico de bajo coste con salida de señal de TV que puede manejar una biblioteca de 5000 títulos en una tarjeta microSD.<sup>19</sup>
- The Humane PC: equipo que usa un módulo Arduino para emular un computador personal, con un monitor de televisión y un teclado para computadora.<sup>20</sup>
- Ardupilot: software y hardware de aeronaves no tripuladas.
- ArduinoPhone: un teléfono móvil construido sobre un módulo Arduino.<sup>21 22</sup>
- Impresoras 3D.

### **1.7.11. Puentes o jumper**

un jumper o puente es un elemento que permite interconectar dos terminales de manera temporal sin tener que efectuar una operación que requiera una herramienta adicional. Dicha unión de terminales cierra el circuito eléctrico del que forma parte. Son generalmente usados para configurar o ajustar circuitos impresos.



Imagen 16. Jumper

### 1.7.12. Puente h

Un Puente en H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, *avance* y *retroceso*. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos.

El término "puente H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores (mecánicos o mediante transistores). Cuando los interruptores S1 y S4 (ver primera figura) están cerrados (y S2 y S3 abiertos) se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 (y cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

### **PLACA DRIVER - CONTROLADORA PARA MOTOR DC MODELO STEPPER L9110**



Imagen 17. Puente H Stepper modelo L9110 DC

### a) Características

- Chip driver - controlador L9110
- Puede manejar dos motores DC o un motor stepper de 4 hilos y 2 fases
- Voltaje de entrada: 3~12V
- Corriente: 800mA un solo motor, 1600 mA manejando dos motores o un stepper de 2 fases.

### b) Aplicaciones

Este driver es un puente H común, el uso es muy simple:

MOTOR-A: conectar terminales del motor A

MOTOR-B: conectar terminales del motor B

GND: Conectar tierra del circuito

VCC: Conectar voltaje de alimentación ( OJO: 1600 mA de corriente)

A-IA: Entrada 1 del motor A



---

A-IB: Entrada 2 del motor A

B-IA: Entrada 1 del motor B

B-IB: Entrada 2 del motor B

### c) Lógica de funcionamiento para el Motor A:

A-IA | A-IB | Operación

-----

L | H | Giro derecha

H | L | Giro izquierda

L | L | Motor stop

H | H | Motor stop

(Idem para el Motor B)

La señal digital alta ("H") en la tabla anterior puede ser también una señal PWM con ciclo de trabajo variable, a fin de controlar la velocidad del motor.

### 1.7.13. SolidWorks

**SolidWorks®** es un software CAD (diseño asistido por computadora) para modelado mecánico en 3D, desarrollado en la actualidad por SolidWorks Corp., una filial de Dassault Systèmes, S.A. (Suresnes, Francia), para el sistema operativo Microsoft Windows. Su primera versión fue lanzada al mercado en 1995 con el propósito de hacer la tecnología CAD más accesible.

El programa permite modelar piezas y conjuntos y extraer de ellos tanto planos técnicos como otro tipo de información necesaria para la producción. Es un programa que funciona con base en las nuevas técnicas de modelado con sistemas CAD. El proceso consiste en trasvasar la idea mental del diseñador al sistema CAD, "construyendo

virtualmente" la pieza o conjunto. Posteriormente todas las extracciones (planos y ficheros de intercambio) se realizan de manera bastante automatizada.

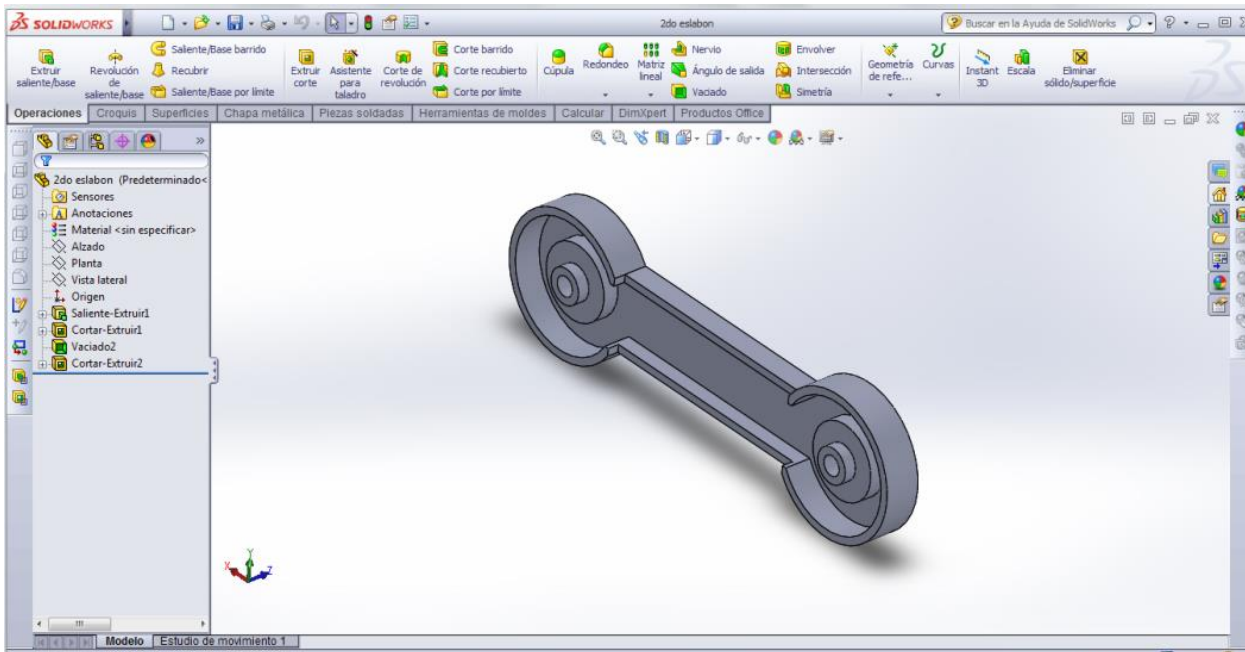


Imagen 18. Pieza para un sistema mecánico dibujado en SolidWorks

El software de CAD en 3D SolidWorks es una aplicación intuitiva con la que podrá desarrollar productos más perfectos, pues permite a su equipo de diseño trabajar de una manera más rápida y productiva. Al igual que las versiones anteriores del software, SolidWorks ofrece innovaciones líderes en el sector, así como cientos de mejoras derivadas de las solicitudes de los clientes. Con ello, su organización ocupará una posición de ventaja con respecto a la competencia.

Diseñe productos más perfectos con prestaciones de CAD en 3D sin precedentes y fáciles de utilizar. Con el software SolidWorks, los datos de diseño son 100% editables y las relaciones entre las piezas, los ensamblajes y los dibujos están siempre actualizados.

---

## **a) Usos y facilidades**

El proceso de diseño con decenas de innovaciones que le permiten ahorrar Tiempo. Disminuya la confusión visual y el cansancio con las interacciones que ocurren en frente de la vista del usuario, un conjunto de funciones de control y de visualización intuitivas.

## **b) Transición e integración de datos 2D a 3D.**

Modificar y mantener archivos DWG en formato

nativo con DWGeditor™, una herramienta de edición que proporciona una interfaz familiar a los usuarios de AutoCAD®. Conserve el valor de sus datos de diseño con la mejor herramienta disponible para convertir datos 2D a 3D, adaptar geometría reutilizable en 2D y permitir una adopción sin complicaciones de la tecnología de CAD en 3D; incluye una exhaustiva documentación de ayuda para usuarios de AutoCAD.

## **c) Prestaciones únicas.**

Gama de herramientas integradas y funciones innovadoras que ofrece en exclusiva el software de diseño mecánico SolidWorks:

- Tecnología SWIFT™ (SolidWorks Intelligent Feature Technology): simplifique el proceso de diseño con la primera tecnología que pone al servicio de los usuarios técnicas expertas para el desarrollo de las operaciones de diseño de CAD en 3D más complicadas. Por ejemplo, SWIFT le permite ordenar de una forma automática y correcta operaciones de piezas tales como ángulos de salida y redondeos.

---

---

- Análisis de piezas integrados: garantice la integridad del diseño y el empleo de materiales de bajo coste gracias a COSMOSXpress™, un asistente para analizar tensiones con sólo seleccionar y hacer clic que permite comprobar los diseños de piezas de una forma rápida y sencilla.

- Comunicación de diseño: comparta con facilidad los conceptos de diseño con eDrawings®, la primera herramienta habilitada para correo electrónico y que facilita en gran medida la comunicación a la hora de diseñar el producto. Gracias a la posibilidad de trabajar con documentos PDF de Adobe® de los diseños en 3D de SolidWorks, el intercambio de conceptos

de diseño a través de los grupos de fabricación e ingeniería resultará todavía más sencillo.

- Herramientas de diseño de maquinaria: saque partido a un completo conjunto de herramientas de diseño, análisis y documentación de piezas soldadas. Obtenga las mejores prestaciones para chapas metálicas, totalmente asociativas. Gracias a ellas, podrá desplazarse con rapidez desde la fase de diseño hasta los dibujos de fabricación definitivos.

Ahorre tiempo consultando una biblioteca de operaciones de diseño de maquinaria.

- Herramientas de diseño de moldes: automatice la creación de núcleos y cavidades con herramientas de diseño de moldes integradas. Utilice MoldflowXpress, una herramienta de validación de diseños basada en un asistente, para comprobar de forma rápida y sencilla la viabilidad de la fabricación de piezas de plástico moldeadas por inyección.

- Herramientas de diseño de productos de consumo: acelere el diseño de productos de Consumo con herramientas mejoradas para la creación y manipulación sencillas de

---

---

## Superficies complejas.

- Prestaciones de búsqueda universales: encuentre con rapidez todos los archivos de SolidWorks, independientemente de que estén almacenados localmente o en una red compartida.
- Acceso en línea a componentes listos para usar: ahorre tiempo con 3D ContentCentral®,

un recurso en línea que proporciona archivos de CAD de componentes listos para usar por proveedores líderes en el sector.

### **d) Modelado de piezas.**

Crear fácilmente diseños con extrusiones, revoluciones, operaciones lámina, vaciados complejos, patrones de área rayada y Taladros aprovechando las prestaciones del modelado de piezas basado en operaciones. Acelere el modelado de piezas con un control exclusivo a nivel de operaciones sobre las piezas multicuerpo. Realice cambios de diseño en tiempo real con la sencilla función de arrastrar y colocar durante la edición dinámica de operaciones y croquis.

### **e) Modelado de ensamblajes.**

Hacer referencia a otras piezas directamente y mantenga sus relaciones al crear piezas nuevas. Beneficiarse de un rendimiento sin precedentes en el diseño de ensamblajes de gran tamaño con decenas de miles de piezas. Trabaje más deprisa con el modo Aligerado sin prescindir de las prestaciones de diseño y documentación. Arrastre y coloque las piezas y operaciones en su lugar.

- Acelere el diseño de ensamblajes con la función de enganche automático SmartMates

y los Componentes inteligentes reutilizables, que se ajustan automáticamente a otros componentes del diseño. Simule el movimiento real y la interacción mecánica entre sólidos con las prestaciones de la Simulación física.

- Simule el movimiento de correas, cadenas, cremalleras, piñones y engranajes, y visualice con facilidad distintos colores, texturas y otras características en pantalla.

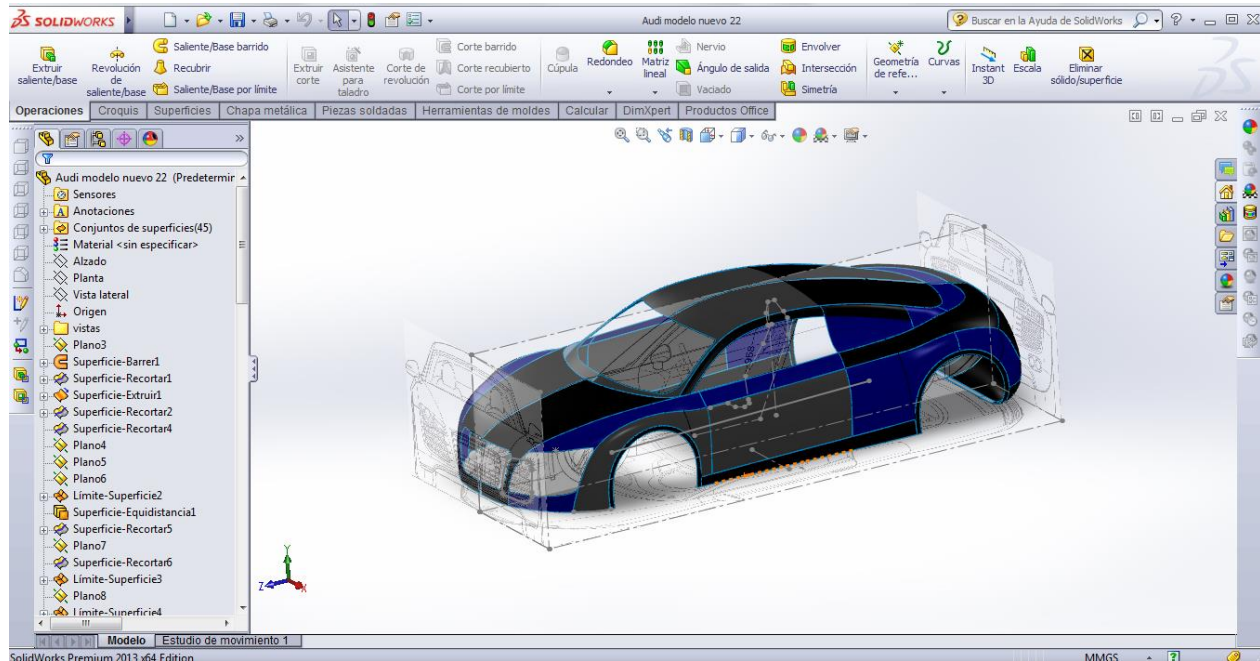


Imagen 19. Recreación de chasis de Audi que realice en residencia profesional en software CAD SolidWorks

## 1.7.14. Matlab

MATLAB es un entorno de computación técnica que posibilita la ejecución del cálculo numérico y simbólico de forma rápida y precisa, acompañado de características gráficas y de visualización avanzadas aptas para el trabajo científico y la ingeniería. MATLAB es un entorno interactivo para el análisis y el modelado que implementa más de 500 funciones para el trabajo en distintos campos de la ciencia.

---

---

Por otra parte, MATLAB presenta un lenguaje de programación de muy alto nivel basado en vectores, arrays y matrices.

Además, el entorno básico de MATLAB se complementa con una amplia colección de toolboxes que contienen funciones específicas para determinadas aplicaciones en diferentes ramas de las ciencias y la ingeniería.

La arquitectura de MATLAB es abierta y ampliamente extensible, permitiendo la relación con Excel, C, Fortran y otras aplicaciones externas muy utilizadas e importantes. Entre otras cosas, el código escrito en lenguaje de MATLAB puede ser traducido a C de forma inmediata.

MATLAB también permite la operatividad entre plataformas posibilitando trabajar con distintos sistemas operativos y relacionar el trabajo realizado en las distintas plataformas.

MATLAB es un software en continuo crecimiento y muy adaptable a los avances científicos y al trabajo en laboratorios I+D, que resuelve los problemas que presenta la ingeniería en el desarrollo de productos innovadores.

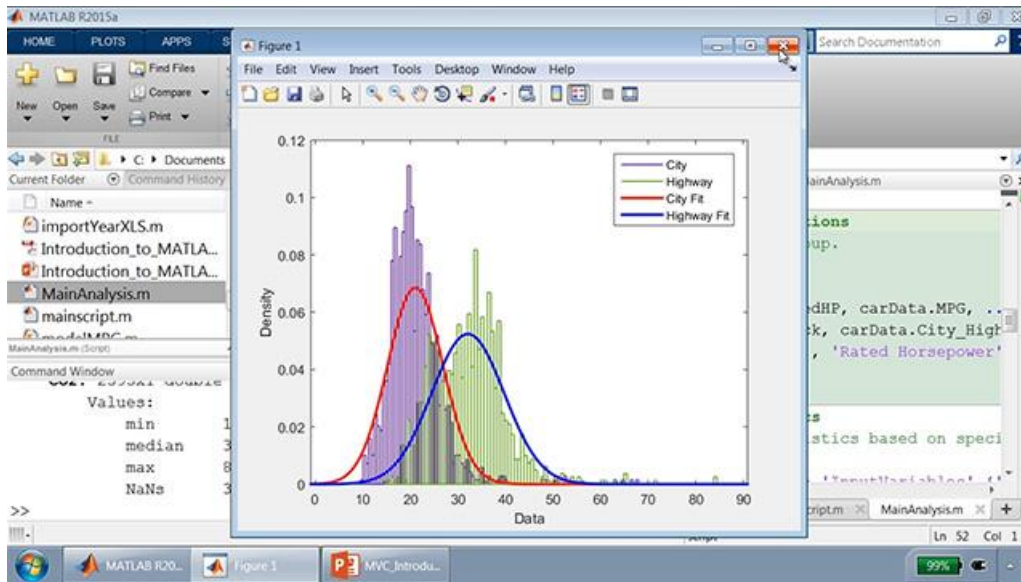


Imagen 20. Grafica simulada el MATLAB

- a) **Comunicaciones:** MATLAB permite realizar modelado y diseño de sistemas DSP, trabajar con sistemas conmutados, con telefonía fija/móvil o ADSL y con modelado de canal/emisor/receptor.
- b) **Periféricos para ordenadores,** MATLAB dispone de drivers para discos, de periféricos de control para posición/velocidad y de instrumentación.
- c) **Campo Aeroespacial/Defensa:** MATLAB permite trabajar en sistemas radar, unidades de seguimiento y rastreo, aviónica, modelado y control de sistemas de potencia y guiado, y navegación y control.
- d) **Campo de la Automoción:** MATLAB posibilita aplicaciones para trabajar en la ingeniería de control, sistemas de suspensión, sistemas ABS y diseño de bloques de embrague.



---

**e) Finanzas cuantitativas:** pudiendo utilizarse como un entorno de cálculo para el análisis de datos, para la valoración y análisis de opciones e instrumentos financieros, para la optimización de carteras y análisis de riesgos y para el desarrollo de modelos y su validación. Asimismo, MATLAB se puede utilizar como un entorno de desarrollo de aplicaciones de renta fija, de opciones derivadas, de distribución de activos/gestión de cartera y de gestión de riesgo y reporting.

Materias como la Estadística, el Álgebra lineal, el Análisis matemático, el Análisis numérico, el Análisis de series temporales, las Bases de datos y la Geometría encuentran en el módulo básico de MATLAB y en sus toolboxes adicionales una herramienta esencial para su desarrollo.

**f) MATLAB, a través de Simulink:** permite diseñar sistemas dinámicos sencillos o complejos y realizar modelado y simulación mediante un lenguaje agradable basado en diagramas de bloques. Admite sistemas en tiempo continuo, sistemas de control y control inteligente, y aplicaciones de procesamiento de señal digital y comunicaciones.

## **1.8. Procedimiento y descripción de las actividades realizadas**

Para efectuar una investigación sobre la implementación del desarrollo de un simulador basado en un robot cartesiano de cuatro grados de libertad y un sistema básico de control industrial utilizando como base el robot STEREN modelo K-680. el robot STEREN modelo K-680 es por definición como modelo de fábrica un “Manipulador multifuncional controlable con 4 grados de libertad, capaz de sujetar y manipular materiales con el punto terminal que es una pinza que se utiliza para realizar tareas diversas según el operador de control”.

Entonces se creara a partir de este su simulador del robot con apoyo de SolidWorks® un software CAD de diseño asistido por computadora y Matlab, más la automatización

---

---

de un movimiento en específico o controlado físicamente alterando parcialmente su morfología para implantar en el los “gadgets” necesarios.

El diseño ideal debió de haber llevado una interfaz gráfica en la que el simulador y el robot físico estarán comunicados y mostrarían movimiento coordinado en tiempo real, también con un movimiento repetitivo.

Una vez establecida la base del proyecto que queda expuesta en el apartado anterior, comenzó el desarrollo del mismo.

En este capítulo se expondrán los pasos seguidos en orden cronológico, dejando para el próximo apartado las modificaciones, ensayos y pruebas que se realizaron hasta llegar al prototipo final

---

## 1.8.1 Diagrama del hardware

Para el buen funcionamiento del sistema primero localizamos las articulaciones, esto para encontrar el lugar ideal donde colocar los potenciómetros Multivuelatas 10K bourns 3590S-2-103 Lineal, que serán sensores con los cuales aremos una relación de valor de: resistencia - ángulo de giro, es decir respecto al giro del potenciómetro, y para poder tener esa relación utilizaremos transportadores geométricos, concéntricos a el eje del potenciómetro, posteriormente el valor de los sensores (potenciómetros) iran a una placa Arduino y sabremos cuanto valor resistivo equivale a cada posición angular de cada eslabón, dado esto también Arduino tendrá salidas conectadas a puentes H tipo DC modelo Stepper L9110 con los cuales se controlaran los motores de cada eslabón y pinza.

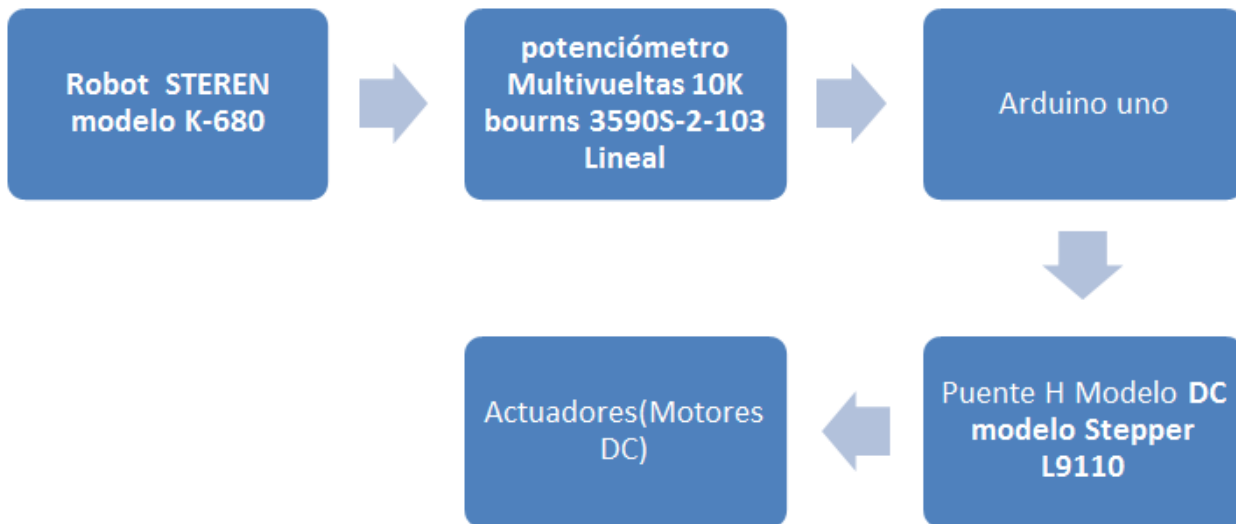


Imagen 21. Diagrama del hardware

---

## 1.8.2 Pruebas del Potenciómetro 10K Bourns 3590S-2-103 Lineal

### 1.8.2.1 Circuito

El circuito es muy simple. El potenciómetro es un componente que tiene tres conectores o patas. Dos de ellas son los extremos de una resistencia fija. La tercera, la central, es la de un contacto móvil, que al deslizarse de un extremo a otro de la resistencia fija la divide en dos y nos da los distintos valores del voltaje en cada punto.

Para utilizarlo, conectamos una de las patas a la alimentación de 5 Volts, la otra la conectamos a tierra y la pata central, donde tenemos la resistencia variable, la conectamos a un pin analógico del Arduino. Es en esta pata central donde censaremos los valores del voltaje que nos da el potenciómetro.

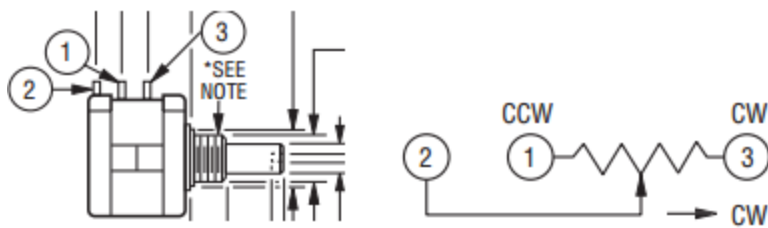


Imagen 22. Izquierda terminales y conexiones del potenciómetro 3590s-2-103

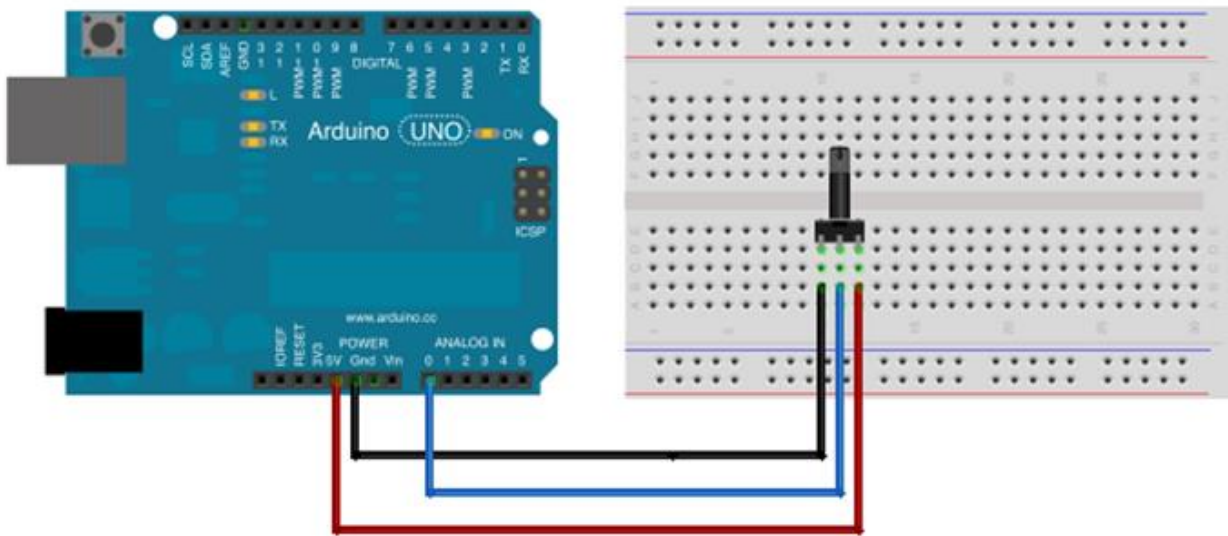


Imagen 23. Conexionado de potenciómetro

### 1.8.2.2 Esquemático

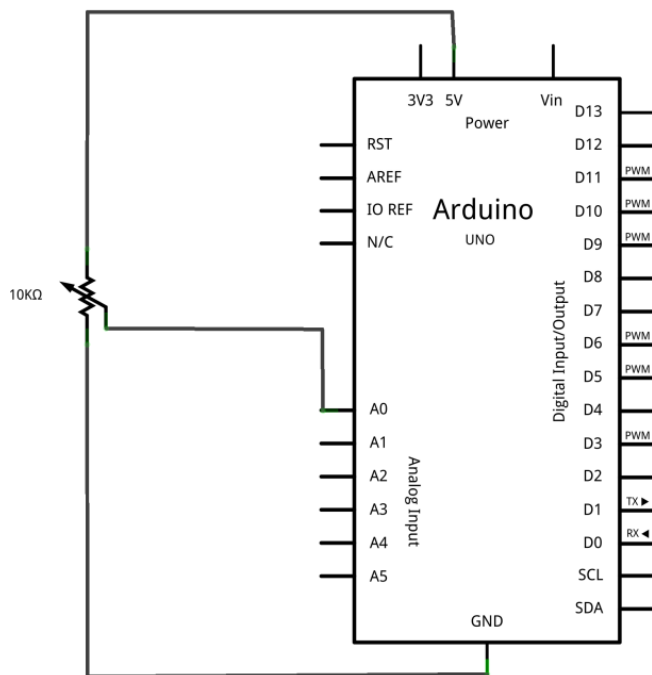


Imagen 24. Esquema del circuito

---

### 1.8.2.3 Código y programación en Arduino

La tarjeta Arduino tiene un circuito interno llamado convertidor análogo-digital (también conocido como ADC, por sus siglas en inglés Analog to Digital Converter) que lee un voltaje de 0V a 5V y devuelve un número en una escala entre 0 a 1023. El comando `analogRead()` retorna el valor del ADC que es aproximadamente proporcional al valor de voltaje aplicado al pin de entrada analógica.

Al conectar el potenciómetro a la entrada analógica vamos a poder observar que podemos variar el voltaje en el pin central entre 0 y 5V.

En la función `setup` se inicia la comunicación serial, a 9600 bits de información por segundo, entre la tarjeta Arduino y la computadora con el comando:

- `Serial.begin(9600);`

Luego, en el loop principal del código, se necesita establecer una variable para almacenar el valor de la resistencia (el cual está entre 0 y 1023) proveniente del potenciómetro.

- `int valor_sensor = analogRead(A0);`

Finalmente, se necesita mostrar esta información usando el monitor serial. Se puede hacer esto usando el comando `Serial.println()` en la última línea del código:

- `Serial.println(valor_sensor, DEC);`

Cuando se abra el monitor serial, se deberá ver una cadena de números en el rango de 0 a 1023, que cambiará cada vez que giremos el potenciómetro mostrando el número equivalente al voltaje en la escala.

#### Lectura Analógica Serial

Leer una entrada analógica del pin 0, muestra el resultado por el Serial Monitor.

---

```
*/  
  
// la rutina de setup corre una vez o cuando se presiona reset  
void setup() {  
  // inicializa la comunicación serial en 9600 bits por segundo  
  Serial.begin(9600);  
}  
// la rutina loop corre constantemente  
void loop() {  
  // lee la entrada analógica desde el pin 0  
  int valor_sensor = analogRead(A0);  
  // muestra el valor que se leyó  
  Serial.println(valor_sensor);  
  delay(1);    // retraso entre lectura y lectura, para la estabilidad  
}
```

#### 1.8.2.4 Monitor serial de Arduino

En base a esto podemos tener en el potenciómetro una lectura que va de 0-1024 y se muestra en el monitor serial como se muestra en la imagen siguiente

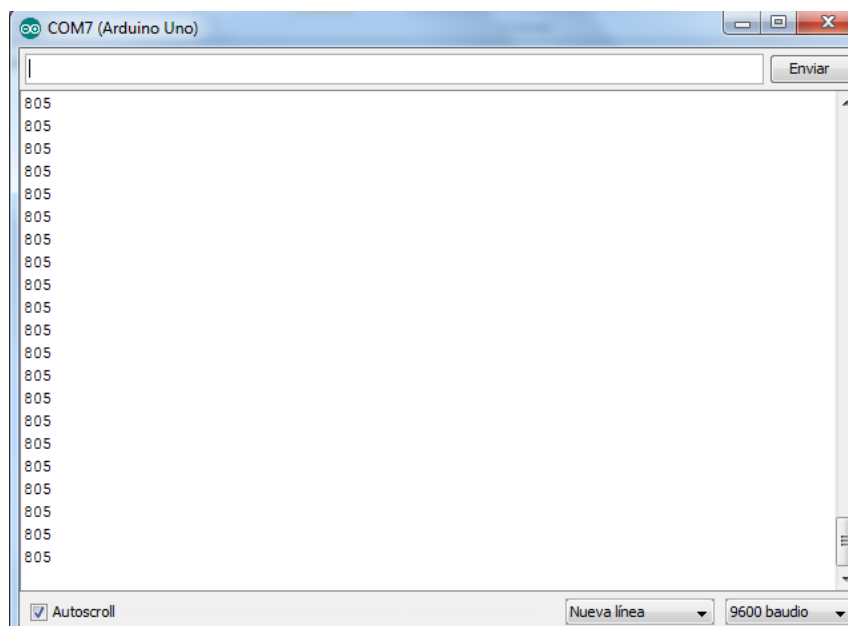


Imagen 25. Monitor serial con calores resistivos de potenciómetro

---

### 1.8.3 Construcción de robot Steren modelo K-680

Dado que el robot STEREN modelo K-680 es parte de un Kit didáctico y es un brazo mecánico para armar, con control remoto alámbrico, contiene un paquete de componentes electrónicos con instructivo para armarlo.

A continuación

#### PLASTIC PARTS

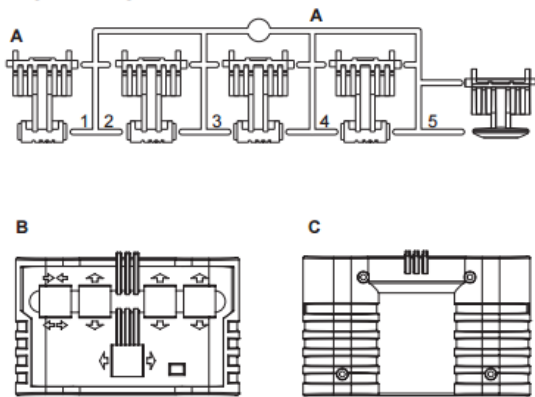


Imagen 26. Partes del control alámbrico

#### PLASTIC PARTS

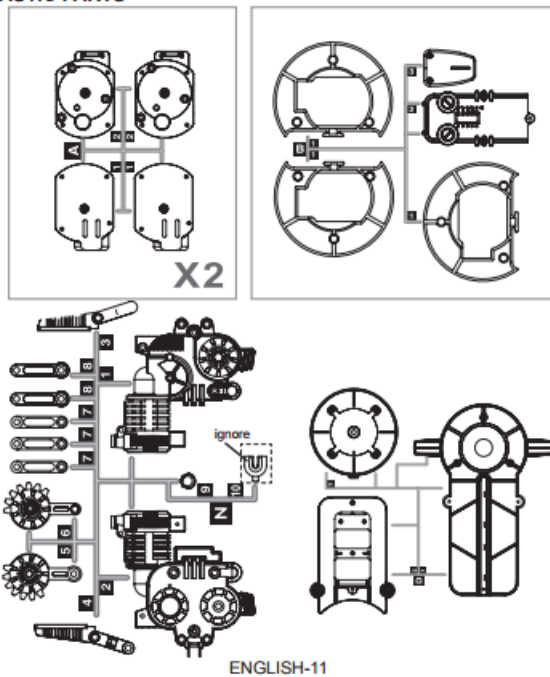


Imagen 27. Partes armables de los motores eslabones y pinza de brazo Robot



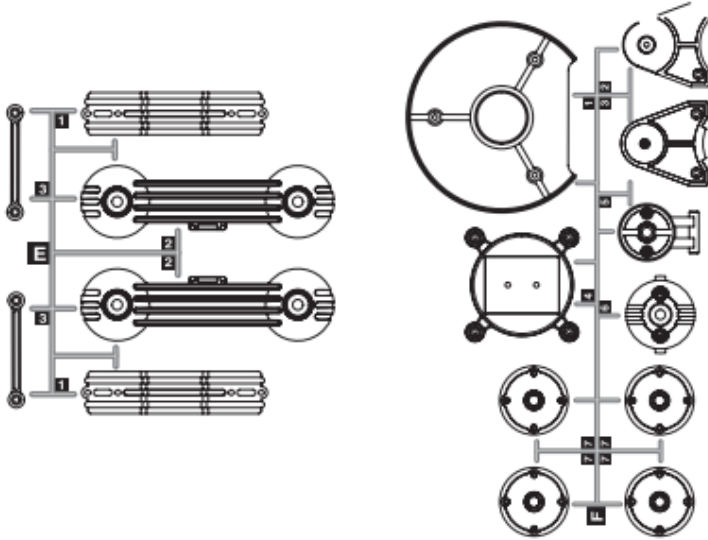


Imagen 28. Partes del segundo eslabón y base o cintura

Una vez armado se hacen pruebas de su funcionamiento con el control alámbrico y se verifica el funcionamiento, se acotan dimensiones.



Imagen 29. Robot totalmente armado con algunos potenciómetros que posteriormente serán utilizados como sensores, Derecha: partes totalmente desarmadas.

---

## 1.8.4 Cconstrucción de robot Steren modelo K-680 en SOLIDWORKS® modelado mecánico en 3D

Como se mencionó con anterioridad para hacer el simulador es necesaria la versión computarizada en el software CAD SolidWorks® para el modelado de cada una de las piezas y el ensamble respectivo posteriormente.

Para la realización de cada pieza, utilizamos un vernier con el cual medimos las longitudes de cada parte para hacer lo más real a la versión original, dividí el cuerpo completo 4 eslabones y la pinza en el último extremo basándome completamente en el robot real como se muestra en la imagen.



Imagen 30. Robot STEREN K- 680

En imágenes siguientes vemos cada una de los eslabones con las medidas similares, utilizando vernier digital para medir las piezas físicas que anteriormente se armaron y esto para hacer la escala uno a uno de cada eslabón, gracias a las clases de SolidWorks® que el asesor externo brindo pude realizar con facilidad cada una de las piezas y ensambles requeridos, cabe mencionar que el procedimiento de medición se excluyó, ya que fue un proceso un tanto laborioso, solo se muestran piezas y los respectivos ensambles.

### 1.8.4.1 primer eslabón

El primer eslabón es la base de o cintura que será una parte rotativa, entonces está compuesta por dos piezas una fija y un rotativa.

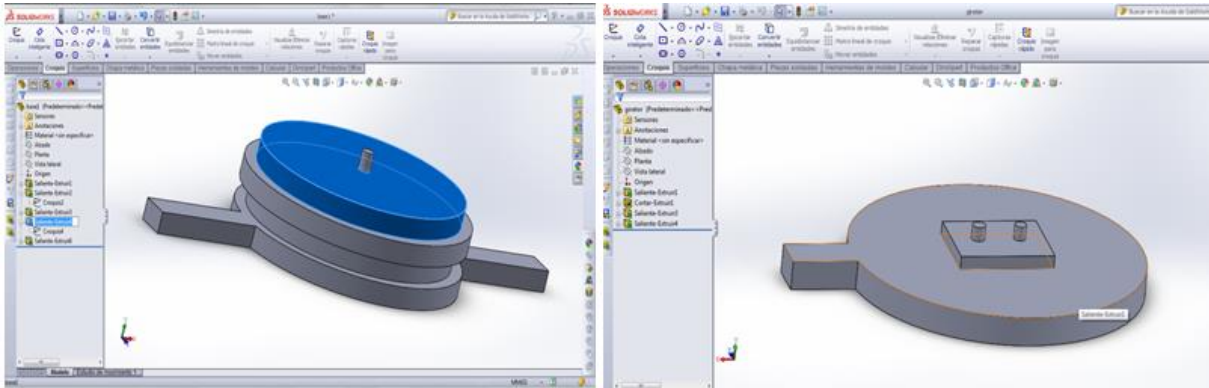


Imagen 31. Primer eslabón o cintura (pieza 1), Derecha: Creación de pieza en SolidWorks (pieza 2)

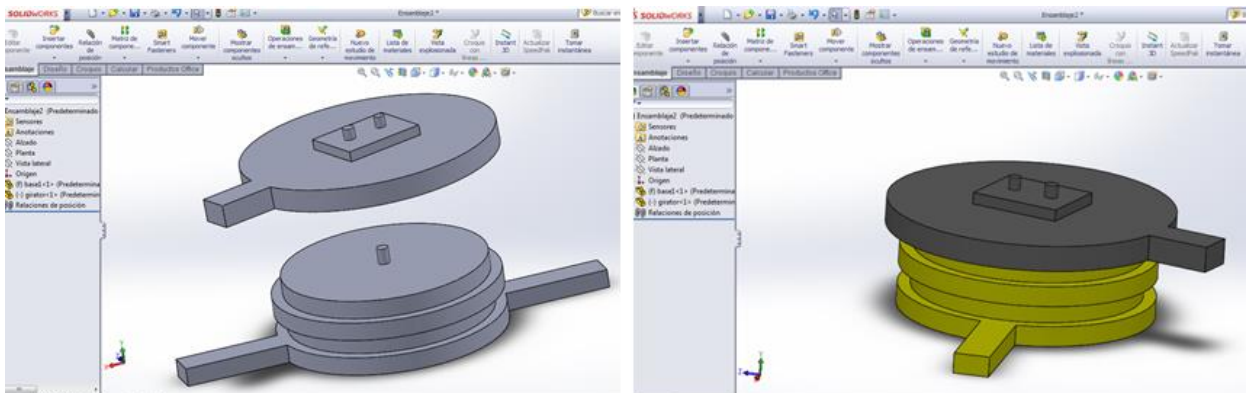


Imagen 32. Izquierda: ensamble de parte fija y cintura, Derecha: Ensamble con movimiento y adición de apariencia (ensamble 1)

## 1.8.4.2 Segundo eslabón

El segundo eslabón es la suma de varias partes movibles pero para hacer esto más real, realice el modelado de esas partes para posteriormente hacer el ensamble

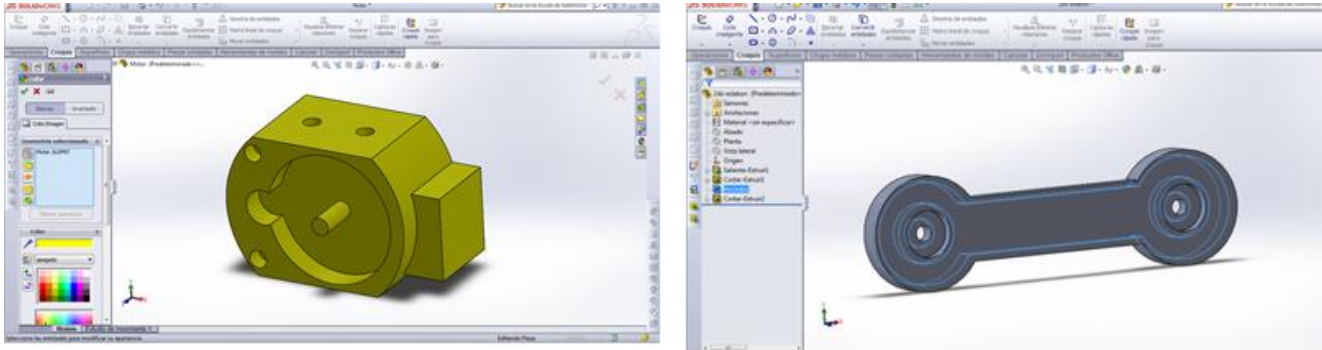


Imagen 33. Izquierda compartimento del motor (pieza 3) , derecha pieza rotativa (pieza 4).

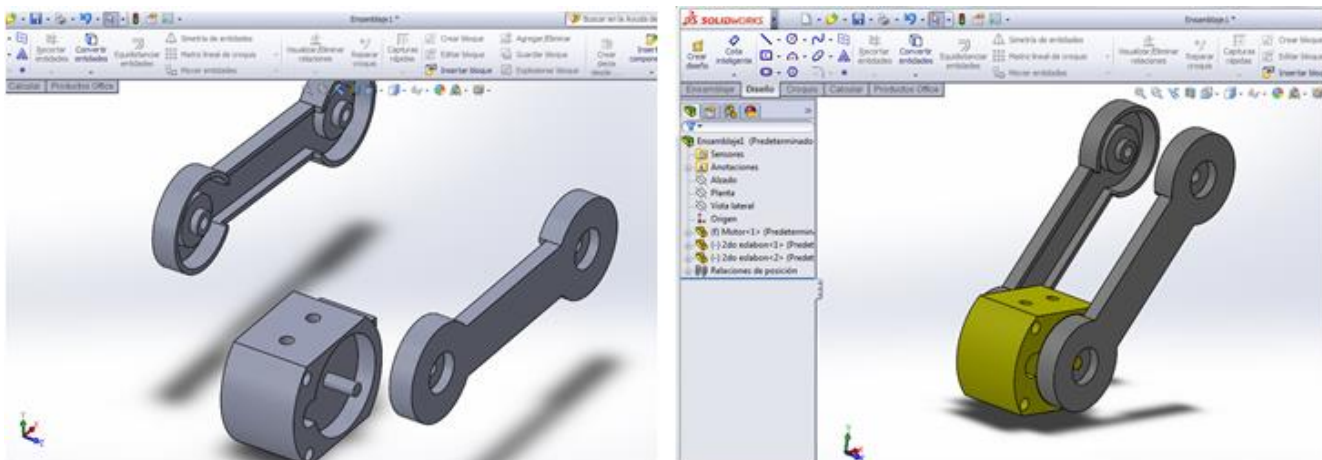


Imagen 34. . Izquierda: piezas antes del ensamble. Derecha Piezas ensambladas y adición de apariencia (Ensamble 2).

### 1.8.4.3 Tercer eslabón

Para esta parte se unieron más piezas, y así como el modelo original utilice tornillos para la sujeción

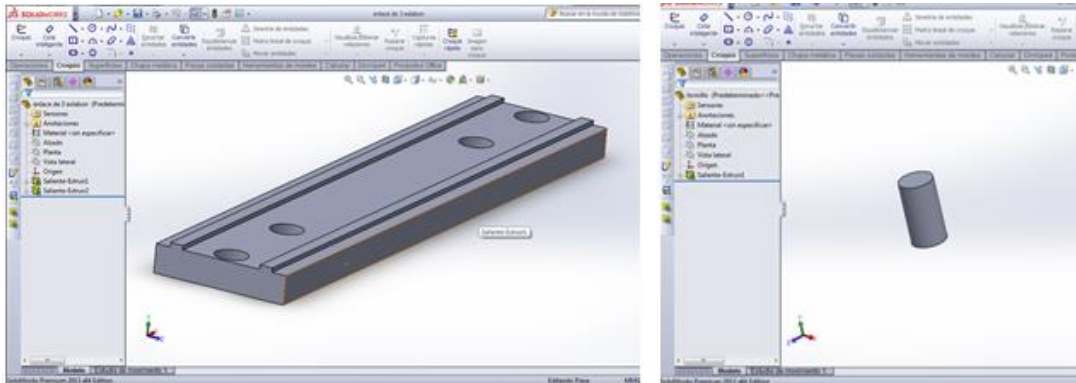


Imagen 35. Izquierda pieza para sujeción (pieza 5), Derecha "tornillo" (pieza 6).

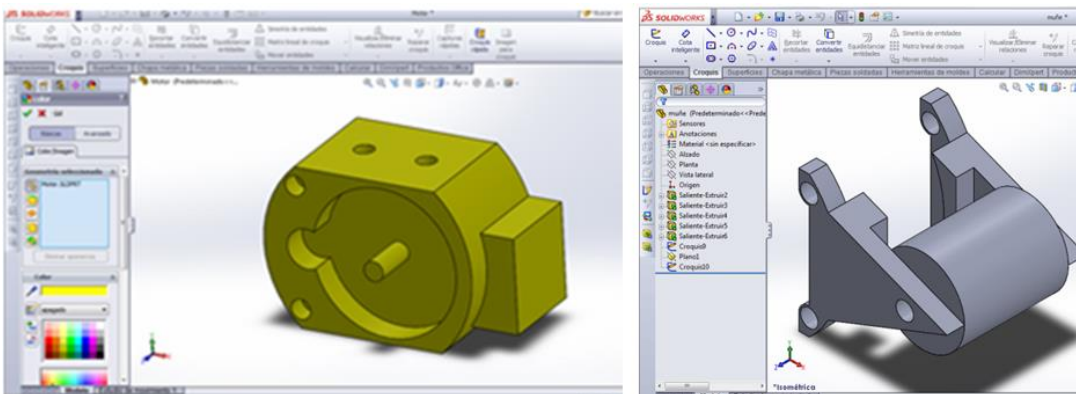


Imagen 36. Izquierda pieza para sujeción (pieza 5), Derecha "tornillo" (pieza 6).

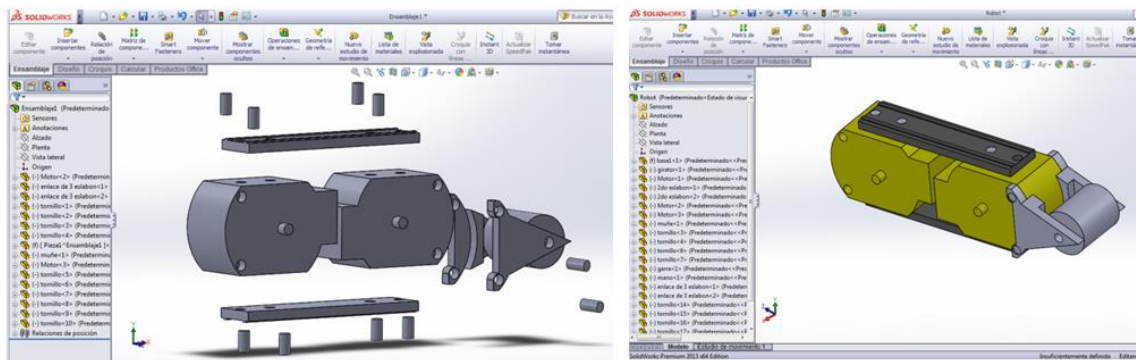


Imagen 37. Izquierda piezas antes del ensamble, Derecha ensamble terminado y adición de apariencia (Ensamble 3).

### 1.8.4.4 Cuarto eslabón

En el cuarto y último eslabón se encuetra el complemento del codo de la muñeca, y en la muñeca una pinza de sujeción como se muestra a continuación, cada pieza y el ensamble final.

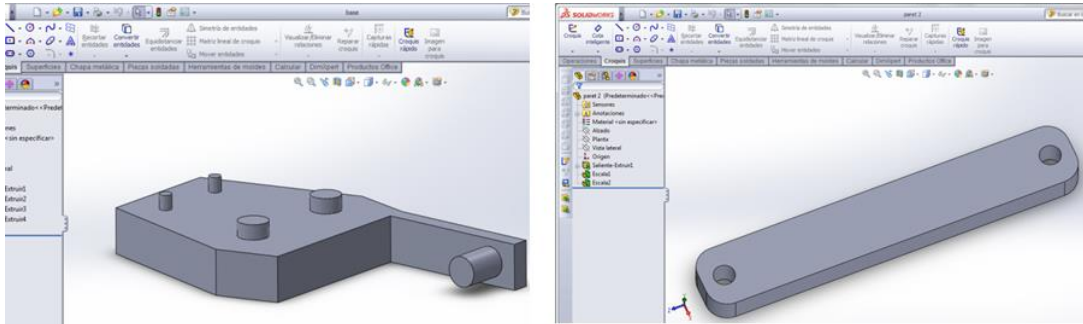


Imagen 38. Izquierda: Base de muñeca (Pieza 8), Derecha: Pieza uno para pinza (Pieza 9).

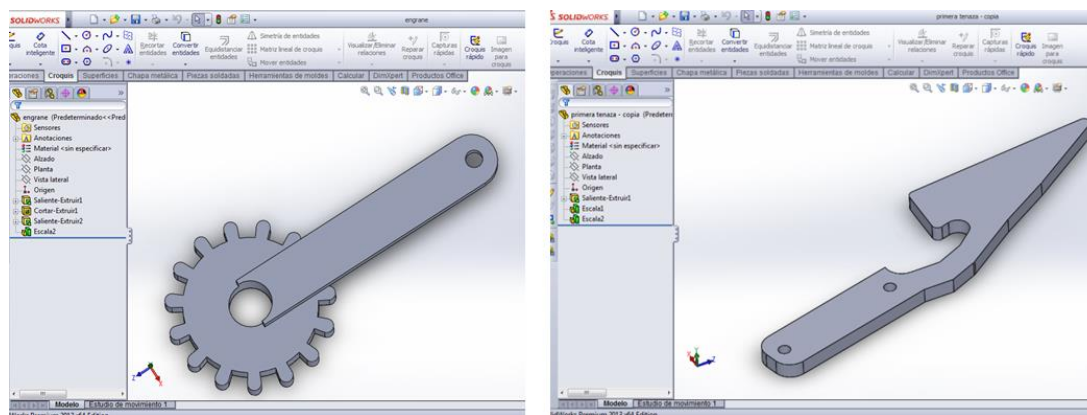


Imagen 39. Izquierda: pieza para pinza (Pieza 9), Derecha: para pinza (Pieza 10).

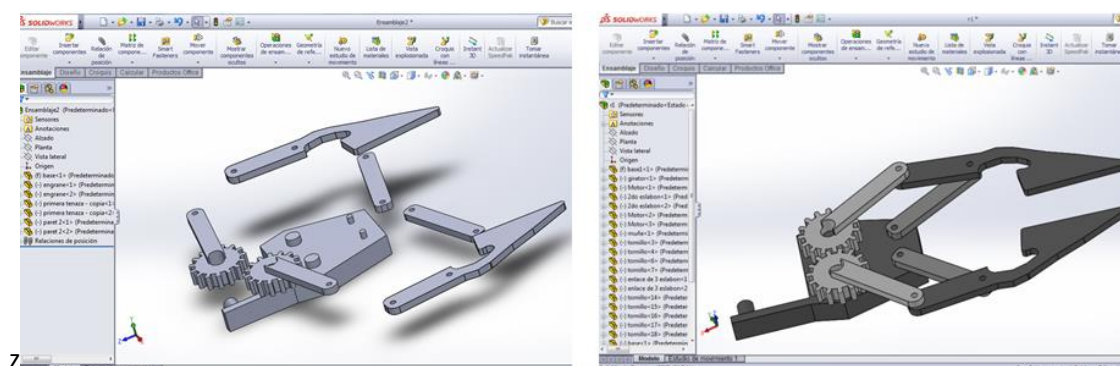


Imagen 40. Izquierda: piezas antes de ensamble, Derecha: Ensamble y adición de color (Ensamble 4).

### 1.8.4.5 Ensamble final

Para completar el modelado virtual total del robot se ensamblan los “ensambles” que anteriormente se realizaron, colocando cada uno de los cuatro eslabones tomando en cuenta la base y pinza.

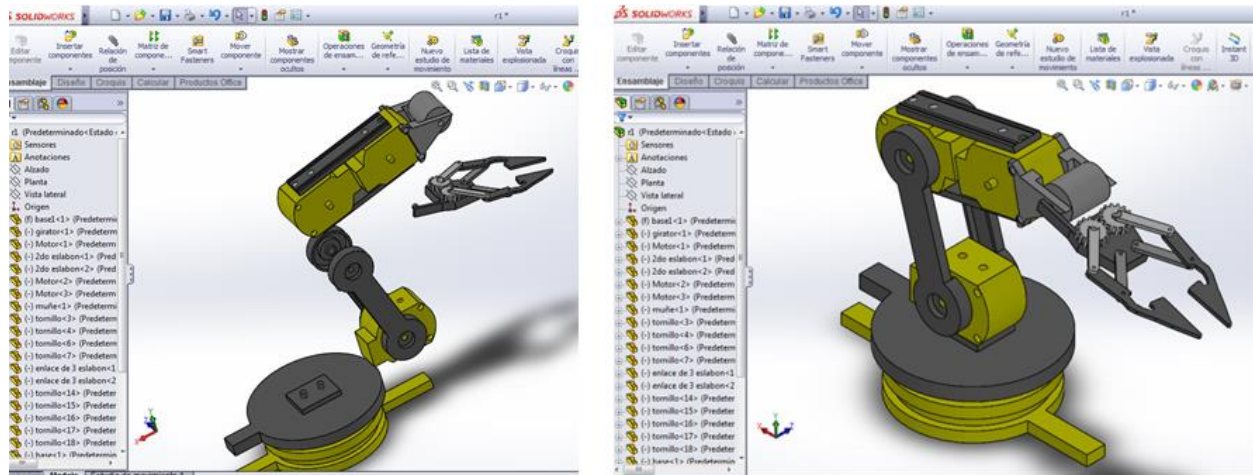


Imagen 41. Izquierda: Piezas sin ensamblar, Derecha Robot ensamblado

Al observar se nota el parecido del modelo virtual al modelo mecánico real.

### 1.8.5 Colocaciones de los sensores sobre cada codo el robot.

Diseñar el mecanismo físico de manera en que los sensores (potenciómetros) puedan yacer sobre la estructura del robot, de tal manera que se obtenga la mejor relación resistencia y Angulo de giro de cada eslabón, es decir: dicho sensor propuesto en el lugar específico es capaz de captar las variaciones de movimiento.

Para esto se conocen las dimensiones de la morfología del el robot y potenciómetro.

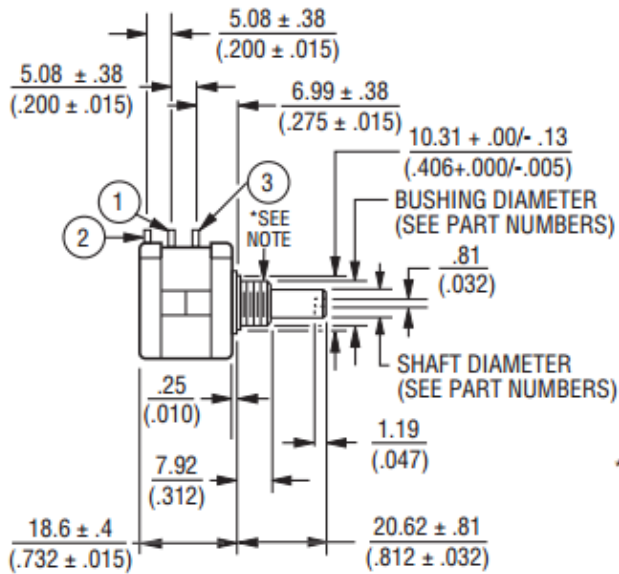


Imagen 42. Dimensiones de Bourns 3590S-2-103

### 1.8.5.1 Identificación de cada eslabón para robot Steren modelo k-680

En la siguiente imagen se observan los eslabones y las articulaciones, por ende se tomaran como referencia el codo de unión de cada eslabón para colocar allí los sensores, Permitiendo así que cuando este giren los eslabones sobre sus respectivos codos lo haga también los cada potenciómetro fijado de manera efectiva.

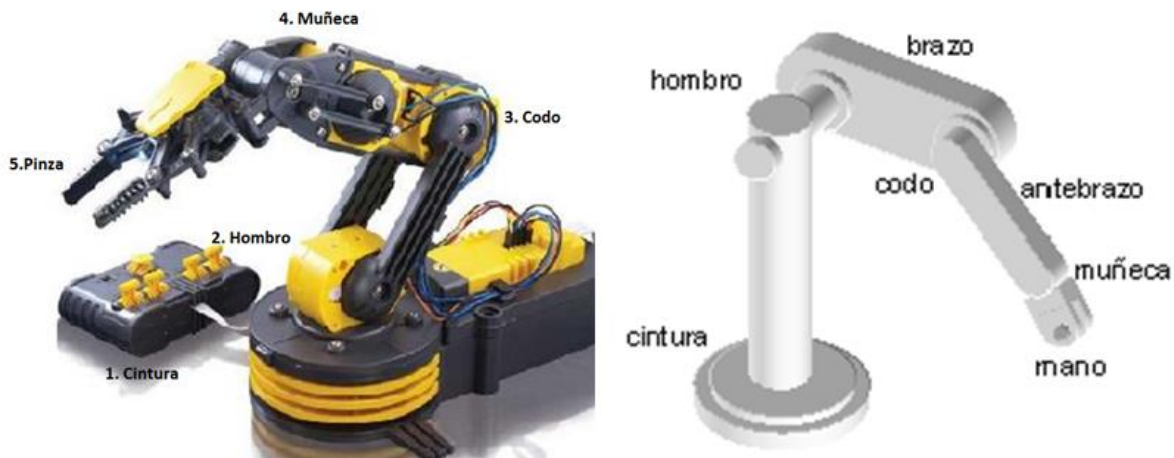


Imagen 43. Eslabones y uniones en la morfología de Robot STEREN modelo K-608



### 1.8.5.2 Identificación de los puntos rotativos ideales para colocar sensores.

Con ayuda del software CAD se pueden proponer partes indicadas en la morfología del brazo físico para colocar allí cada sensor de giro (potenciómetro lineal), y como se menciona anteriormente en cada codo es un lugar en que no se necesita hacer un sistema de engranes o un diseño mecánico con algún tipo de transmisión a excepción del giro de la cintura.

#### Cintura

#### Hombro

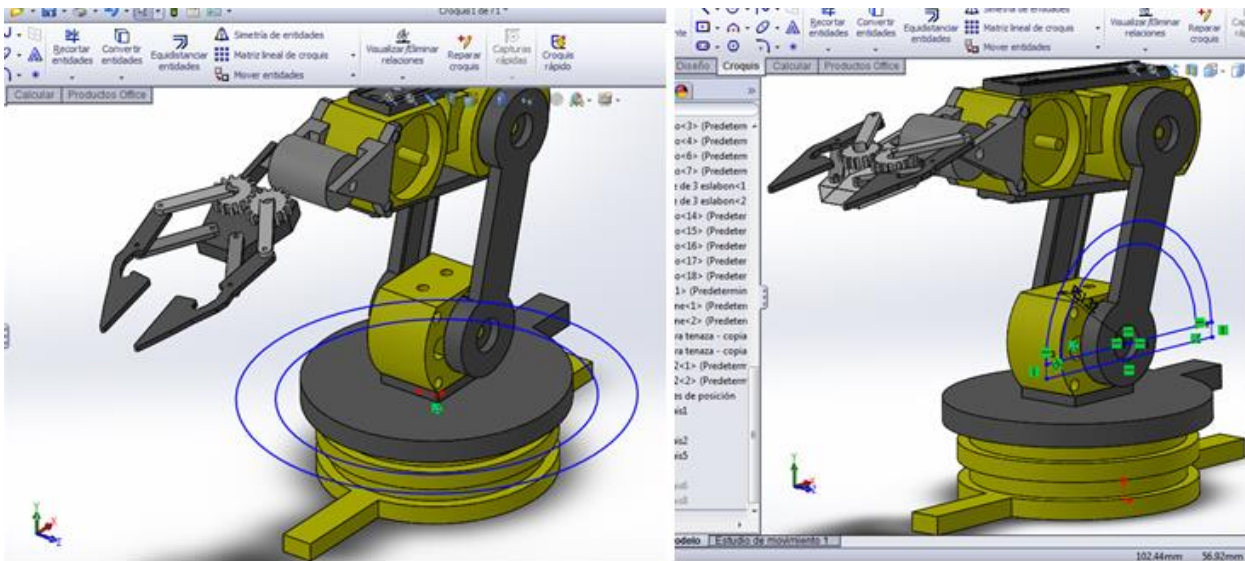


Imagen 44. Movimiento rotativo de la cintura y moviendo de hombro.

Como se ve en la imagen 44 se aprecia la dimensión de movimiento rotativo para cada eslabón y se puede observar que la cintura puede dar los  $180^\circ$  de libertad, cabe mencionar que ese rango de movimiento se corrobora primero en el modelo del robot real, tomando en cuenta la extensión del cableado que alimenta a todos los motorreductores.

## Codo

## Muñeca

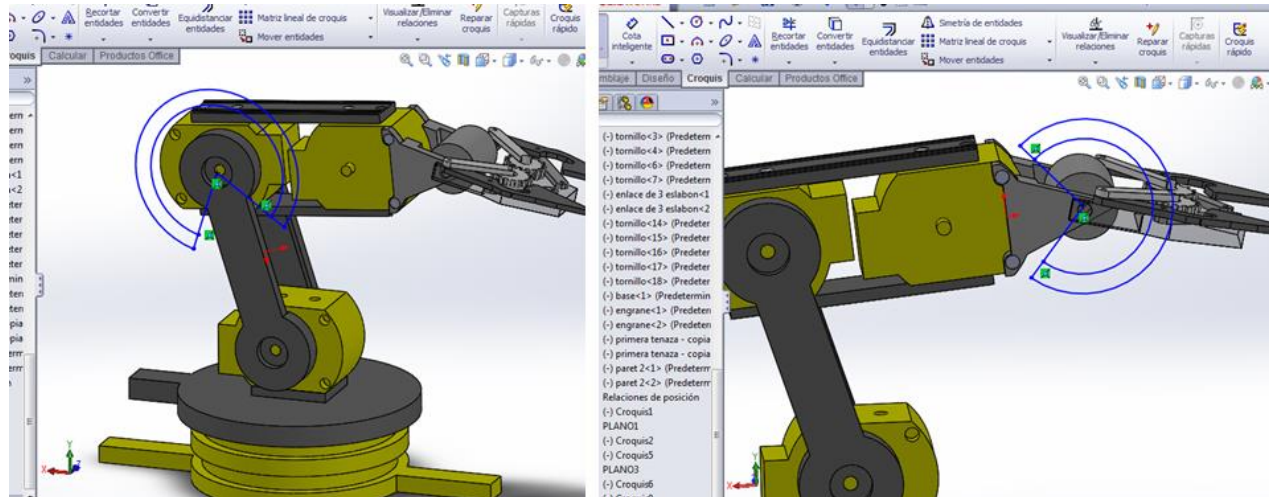


Imagen 45. Movimiento de codo y movimiento de muñeca.

### 1.8.5.3 Limitaciones en el espacio de trabajo

Las limitaciones de movimiento o Angulo máximo de giro fueron propuestas por el funcionamiento óptimo del mecanismo del que lo controla que también están en las especificaciones de fábrica como se muestra en la imagen 46, ya que este no cuenta con sensores de final de carrera es necesario conocer las limitaciones de movimiento, pues cuando se le es imposible seguir moviéndose físicamente porque existe algo que obstruye su libre movimiento ocurre un sobreesfuerzo de la caja de reducción, es decir por ejemplo si el tercer eslabón gira sin detenerse impactara contra alguna parte de él, mientras que el operador o la programación no lo detenga, este seguirá y ocasionara daños en el mecanismo interno.

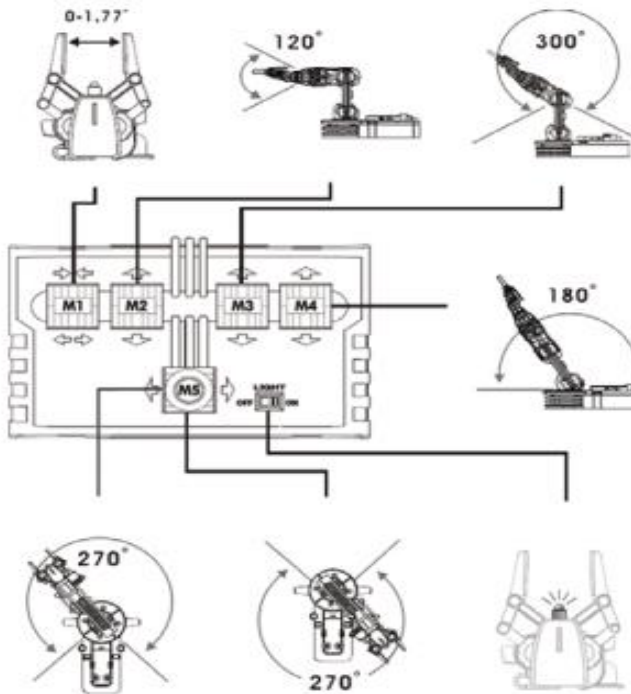


Imagen 46. Especificaciones de espacio de trabajo del robot

#### 1.8.5.4 Adaptación y colocación de sensores

##### a) Transmisión por levas

Para el giro de la cintura se propone un sistema de movimiento por levas, ya que en este sistema se mueve por fricción y es fácil de acoplar, un ejemplo es el de la imagen siguiente.

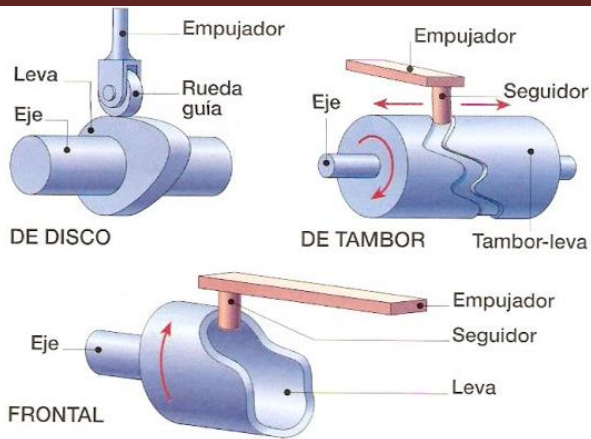


Imagen 47. Tipos de levas: de disco, frontal y de tambor.

### b) Cintura y acoplamiento de transmisión por levas

Como se muestra en la imagen anterior las levas son un sistema de transmisión eficiente y en este caso aremos uso del principio de transmisión por disco, ambos discos circulares y se utilizara caucho para el contorno de cada rueda para mejorar el agarre. A continuación se muestra el modelado mecánico de la propuesta.

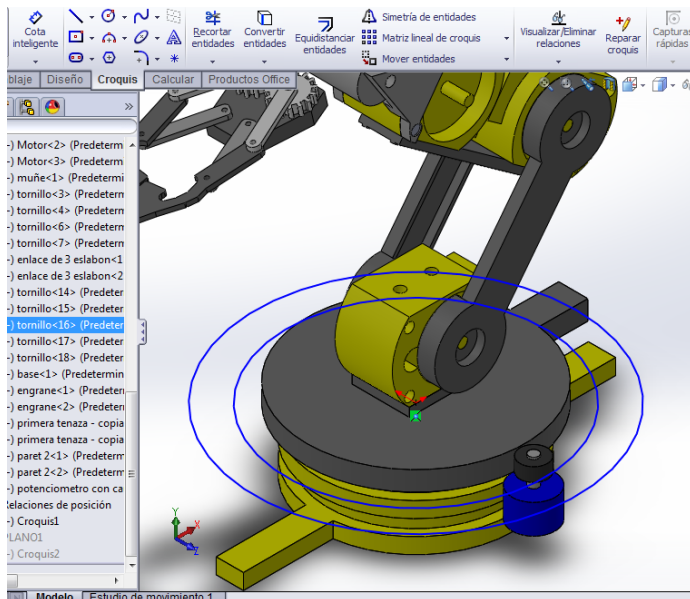


Imagen 48. Propuesta para la transmisión de giro por principio de levas “pieza 2 - Potenciómetro lineal”.

---

La opción de transmisión de movimiento al motor por medio de levas es la más conveniente tomando en cuenta que no se modificaran más cosas del modelo mecánico real para lograr el resultado y las respuestas mecánicas que buscamos.

### **c) Transmisión directa**

En ingeniería mecánica se conoce como eje de transmisión a todo objeto asimétrico especialmente diseñado para transmitir potencia y en este caso se usara el mismo principio aplicado al movimiento de los sensores.

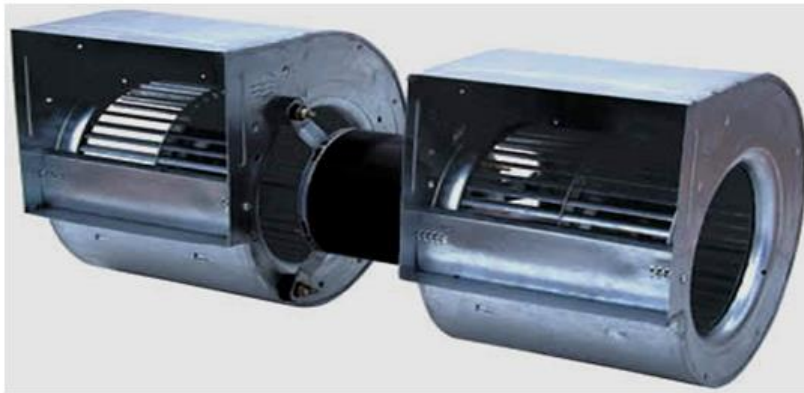


Imagen 49. Motor eléctrico transmitiendo movimiento y potencia directamente a otro.

### **d) Hombro y acoplamiento de sensor**

Se deduce al ver las imágenes del robot que los sensores se pueden colocar directamente sobre cada “articulación” a excepción de la cintura, y dado esto se procede a proponer en CAD la manera en que cada potenciómetro será montado.

Dada la morfología del robot y la posición en la que este tiene el “hombro” puedo colocar en el eje de giro de la articulación que una a cada eslabón con el anterior.

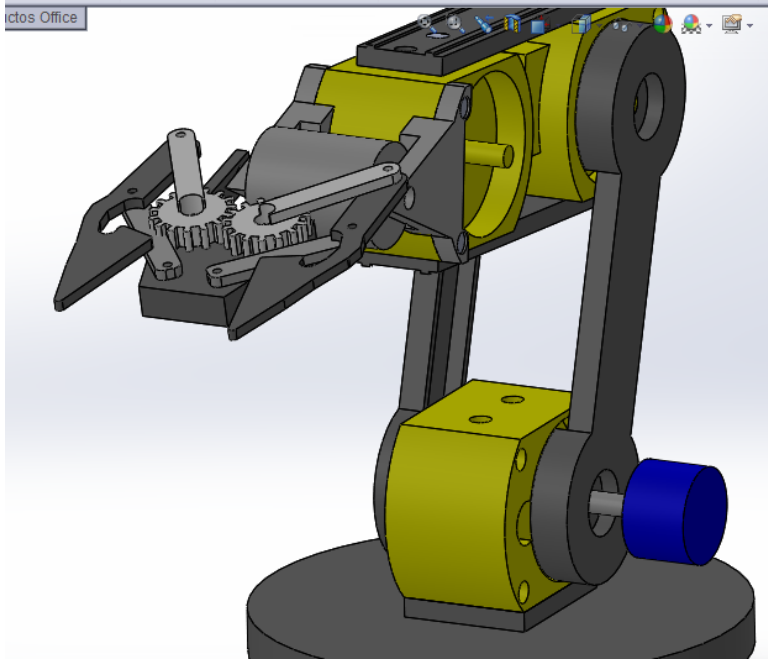


Imagen 50. Potenciómetro colocado en lo que llamaremos segunda articulación.

### e) Codo y acoplamiento de sensor

Para el codo también tomamos el mismo principio que la articulación de hombro puesto que el eje que transmite un esfuerzo motor y está es la misma torsión que girara nuestro sensor.

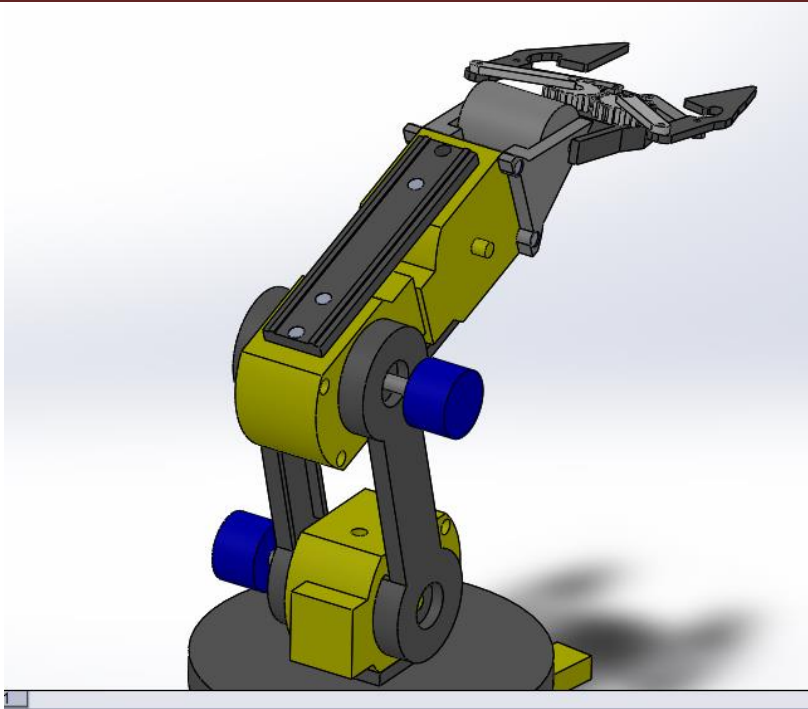


Imagen 51. Simulación de acople de sensor de tercer articulación.

**f) Acoplamiento de sensor a muñeca y pinza.**

En esta parte esencial del robot existe una ligera diferencia en cuanto al mecanismo real, el modelo virtual y el posicionamiento de algunas piezas de la muñeca difieren de la real, pero cabe recalcar que el principio de movimiento es el mismo, mecánicamente encontramos la misma respuesta ante las fuerzas de torque respectivamente con el movimiento angular.

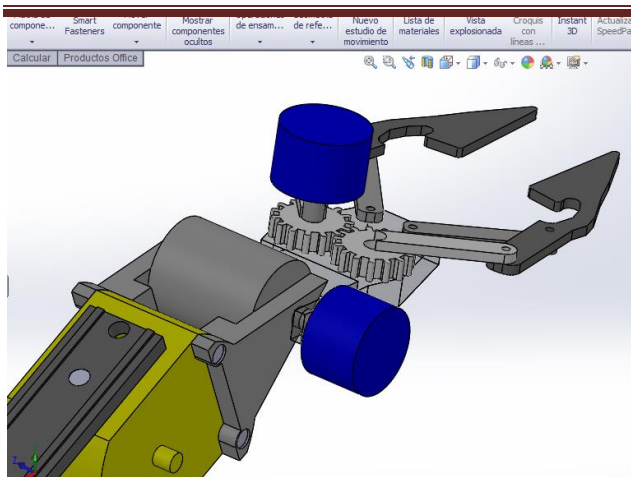


Imagen 52. Acople de sensores en cuarta articulación y en pinza.

### 1.8.5.5 Montaje físico de sensores

Para esta tarea se montaron los sensores como se determinaron en los pasos anteriores, pero se halló un inconveniente en el acople de los sensores a excepción del movimiento de la cintura, ya que para que el movimiento mecánico fuera eficiente es necesario que la perilla en los potenciómetros sean colocados de manera concéntrica en los ejes rotacionales de las articulaciones, para esto se creó con ayuda de SolidWorks® en una impresora 3D el acople necesario, puesto se es vital que tanto la perilla de cada potenciómetro sea concéntrico con el eje rotacional como ya se mencionó.

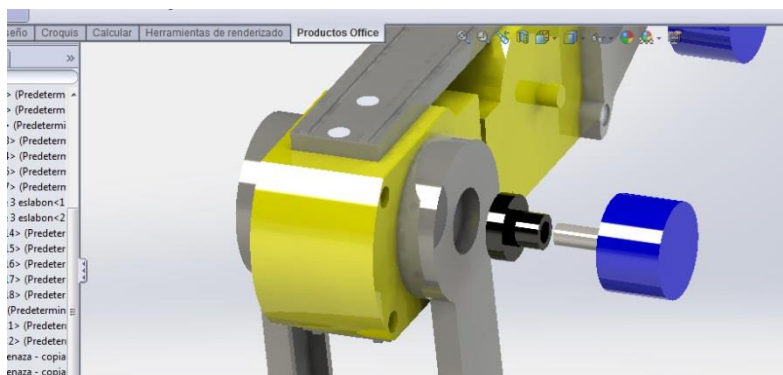


Imagen 53. En color negro acoplamiento para la sujeción concéntrica de sensores.



---

### a) Montaje de sensor en cintura

Se puede observar en la figura 53 que en la sujeción se utilizaron piezas de caucho alrededor de cada disco para que funcione con el principio de transmisión de movimiento por levas de disco.

Dado que el sistema mecánico no tenía que ser interferido en la libertad de movimiento se adecuo y se pusieron con ayuda de ciertos materiales como plásticos y ligeras barras de madera todo para un buen desempeño mecánico y no haciendo obstrucción del área de movimiento.

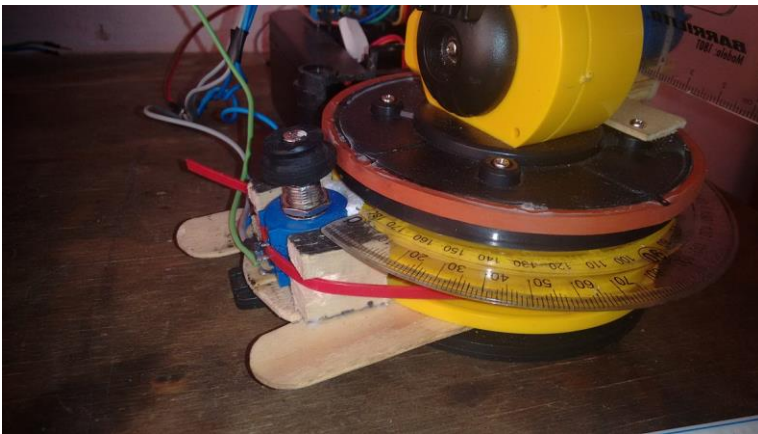


Imagen 54. Acoplamiento de sensor en cintura de robot

También como se puede observar en la imagen 54 para conocer mejor la relación “Angulo de giro- Resistencia eléctrica” y las limitaciones del espacio de trabajo se colocaron transportadores geométricos y así saber en qué rango corresponde a cada valor del sensor posteriormente cuando se pueda medir.

### b) Montaje de sensor en hombro

Como se mencionó anteriormente, a diferencia de la transmisión por levas de la cintura rotativa del brazo, es en este caso es una transmisión directa y con un acople diseñado en el software CAD de Solidwork, se puede apreciar la pequeña pieza que sirve como acoplador entre la perilla del potenciómetro y la parte cóncava de esta articulación.

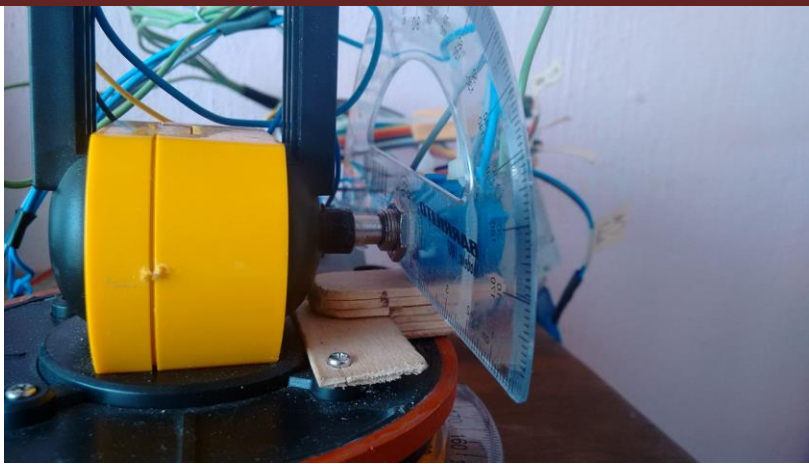


Imagen 55. Acoplamiento de sensores en codo de robot

Para conocer la ubicación angular se adaptó un transportador geométrico que brinda la oportunidad monitorear visualmente el movimiento

### c) **Montaje de sensor en codo**

Las especificaciones anteriores para la sujeción y los acoles son los mismos al igual que el mismo principio de movimiento y utilidad transportadores geométricos.

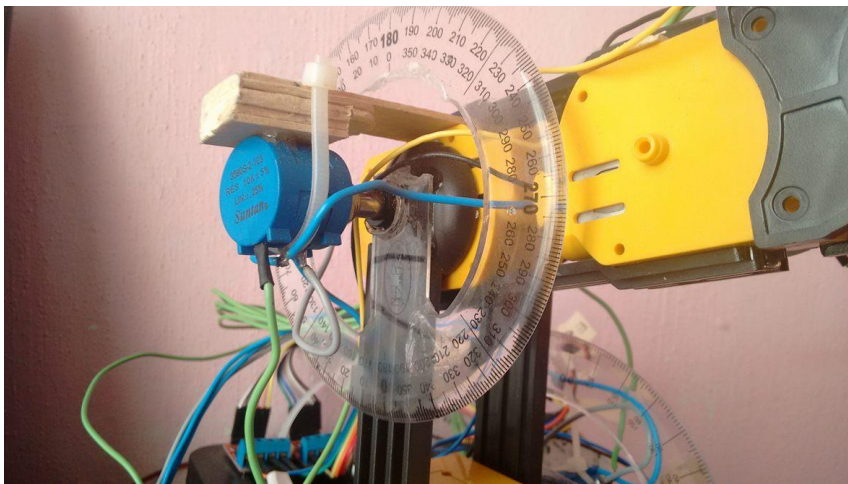


Imagen 56. . Articulación codo al igual que sujeción de sensor y transportador.

---

**d) Montaje de sensor sobre muñeca y pinza**

Las especificaciones anteriores para la sujeción y los acoples son los mismos al igual que el mismo principio de movimiento y utilidad transportadores geométricos para conocer el espacio de trabajo

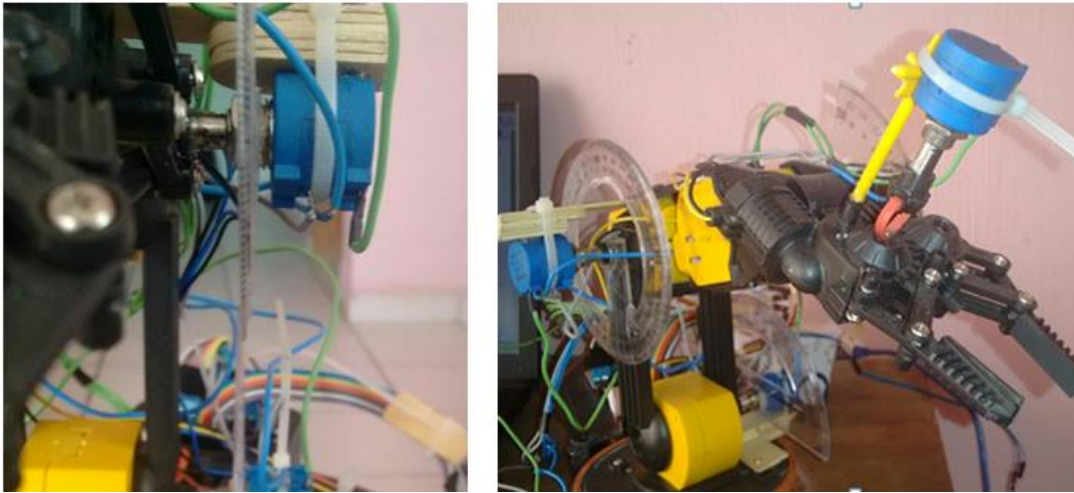


Imagen 57. Izquierda Sujeción de sensor en muñeca, Derecha: sujeción de sensor en pinza.

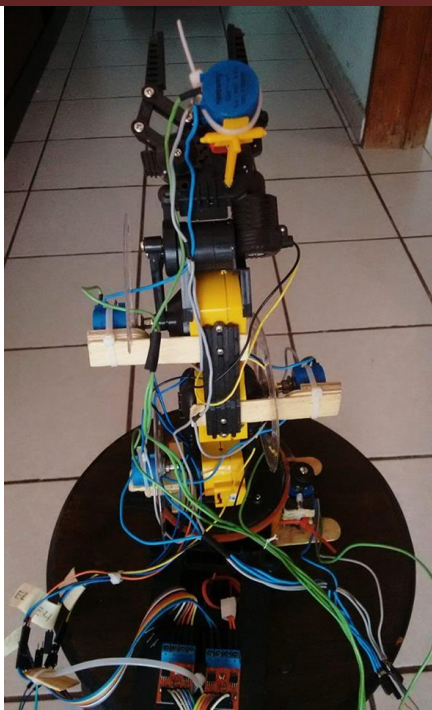


Imagen 58. Potenciómetros montados sobre los ejes rotativos del robot

### 1.8.6 Diagrama de conexiones

El sistema del diseño electrónico propuesto está representado por la imagen siguiente. Consiste de un sistema de sensores y actuadores para permitir una mejor interacción con el entorno, los robots autónomos son capaces de reconocer el ambiente y sus limitaciones, en este caso esa información es proporcionada por los potenciómetros.

Los potenciómetros “Bourns 3590S-2-103 Lineales” con ayuda de “Arduino Uno” serán los responsables de guiar nuestro movimiento o mostrarnos en qué posición se encuentra, para esto se conectan como se muestra en la imagen, dado que es un sensor analógico, por obvias razones el pin de la señal será acoplado con los pin de entrada analógica de Arduino, los 2 pines restantes se conectan a VCC y Ground de Arduino, y como se ve la imagen se repite el mismo circuito para los demás potenciómetros. Con todos los potenciómetros conectados respectivamente podremos

---

ver con ayuda del software y programación apropiada los valores en serial Monitor casi en tiempo real.

Posteriormente con la misma placa Arduino se enviarán las señales a los actuadores que son los motores, cabe mencionar que el sistema de control alámbrico por el usuario fue retirado ya que no será de utilidad para lo propuesto. Las señales son enviadas a “Placa Driver - Controladora Para Motor DC modelo Stepper L9110” (puente H) que es un módulo de control de motor DC que opera el funcionamiento de dirección de giro de motor, en este caso se muestra de manera representativa con los tres circuitos integrados que está sobre el protoboard.

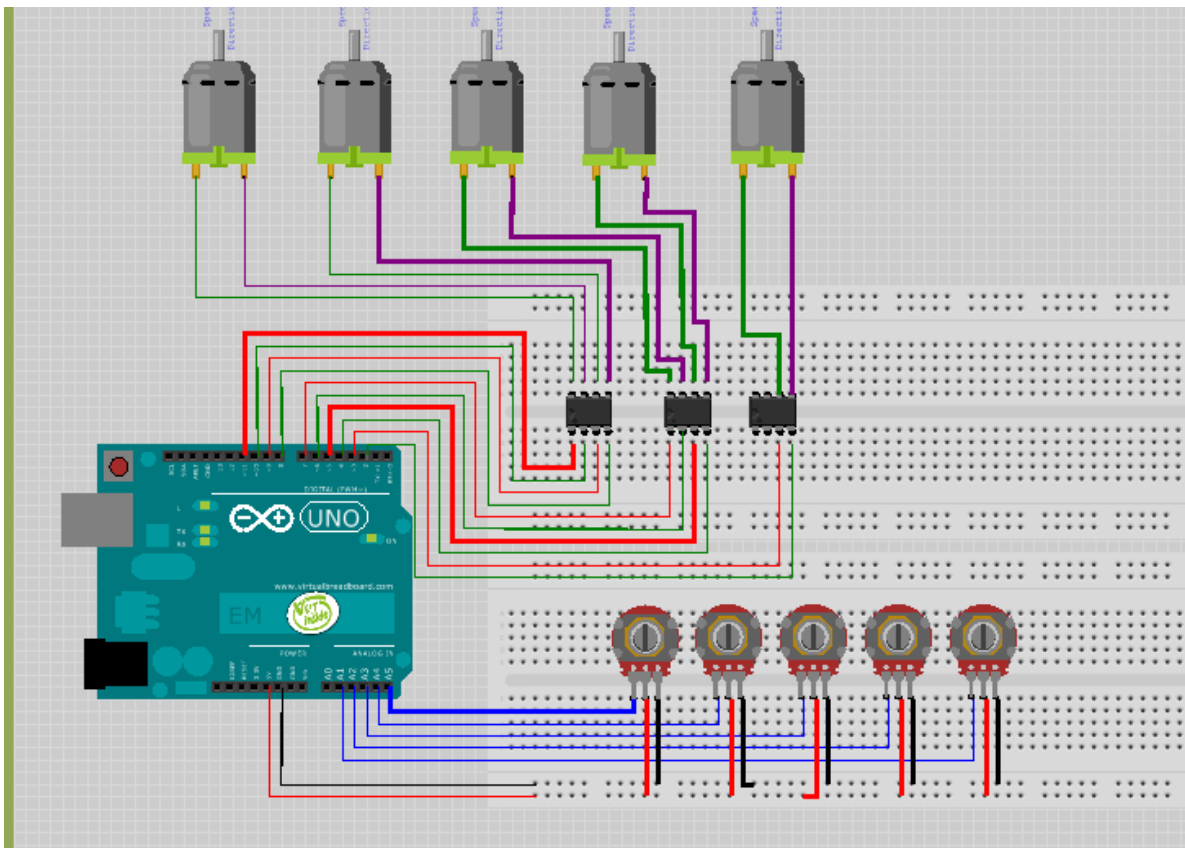


Imagen 59. Representación de conexión de circuito de control.

## 1.8.7 Creación de piezas en solidwork para modelado de robot y simulación en Matlab.

Justo como el modelado de piezas mecánicas para crear una réplica del “robot STEREN modelo K-680” en ejemplos anteriores, también es necesario crear un diseño apegado al mismo robot pero de menor detalle gráfico y mecánico ya que eso imposibilita un buen desempeño en el simulador que utiliza el software matemático Matlab para una interfaz de usuarios en simulaciones mecánicas.

### a) Primer eslabón

El primer eslabón como ya vimos es la base o cintura que será una parte rotativa, y está compuesta por dos piezas una fija y un rotativa.

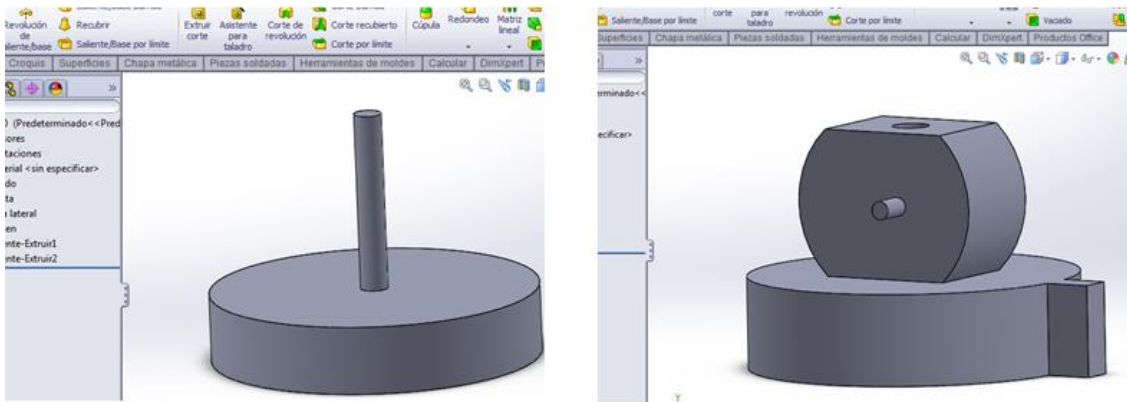


Imagen 60. Primer eslabón o cintura (pieza 1), Derecha: Creación de pieza en SolidWorks (pieza 2)

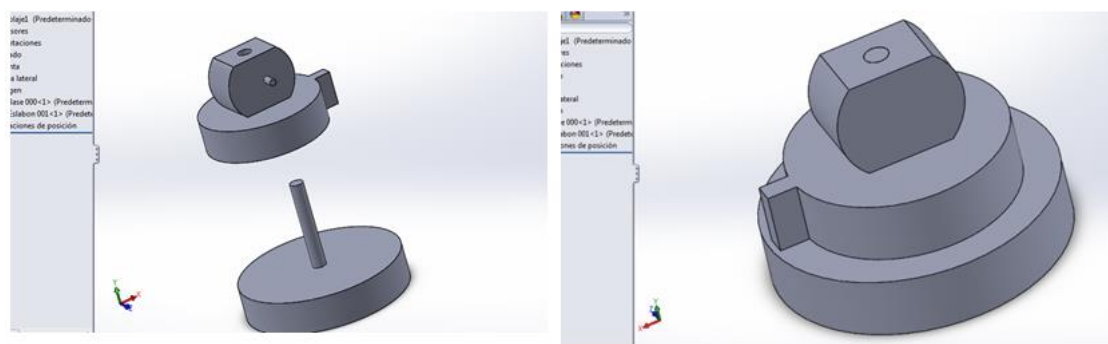


Imagen 61. Izquierda: ensamble de parte fija y cintura, Derecha: Ensamble con movimiento

---

En este caso la cintura tiene también parte del segundo eslabón que se hizo anteriormente.

**b) Segundo eslabón**

En este segundo eslabón solo contamos con una pieza ya que como he mencionado, para poder hacer una apropiada simulación mecánica en Matlab es conveniente utilizar el menor número de piezas ensambladas, esto para los gráficos y movimiento.

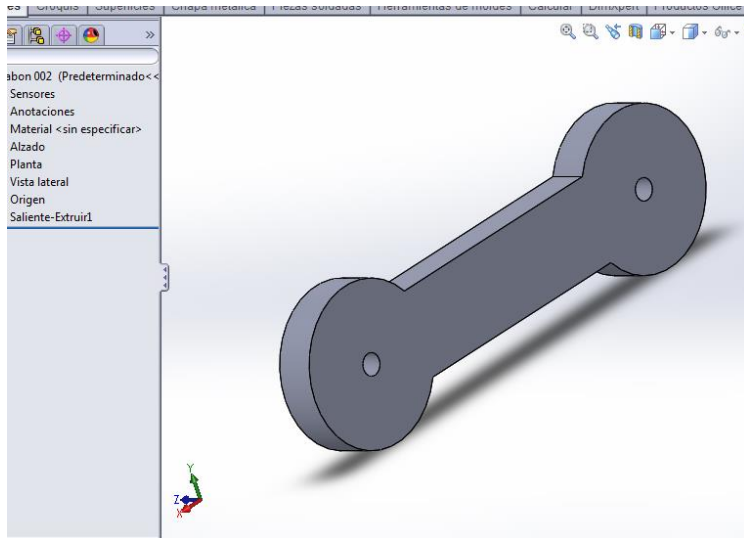


Imagen 62. Segundo eslabón

---

### c) Tercer eslabón

Para el tercer eslabón ocurre lo mismo que en los anteriores ya que solo necesitamos una pieza.

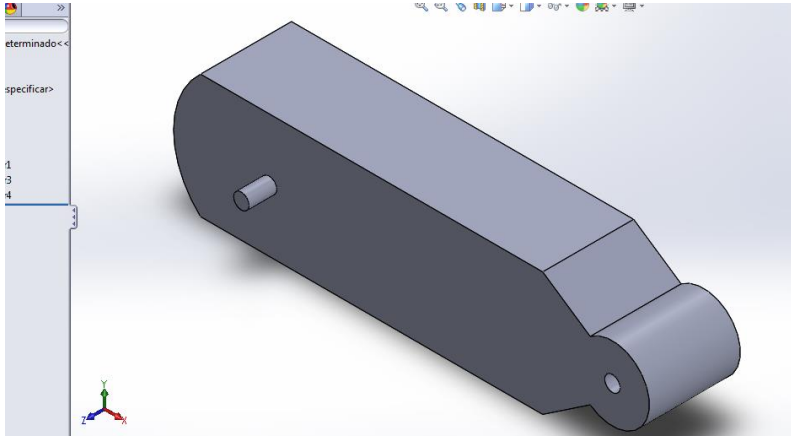


Imagen 63. Tercer eslabón.

### d) Cuarto eslabón

Este eslabón forma parte de la muñeca y como vemos el punto terminal es una pinza.

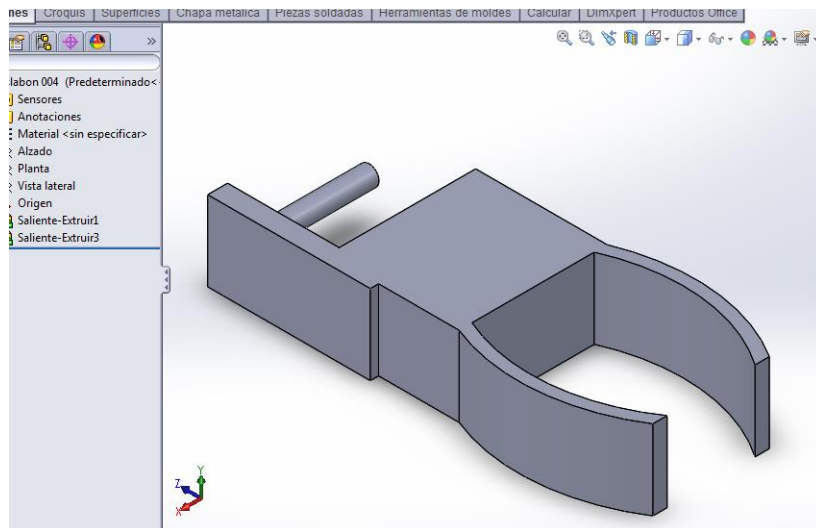


Imagen 64. Muñeca y punto terminal representativo de pinza.



## e) Ensamble final

Para completar el modelado virtual total del robot se ensamblan las piezas modeladas anteriormente.

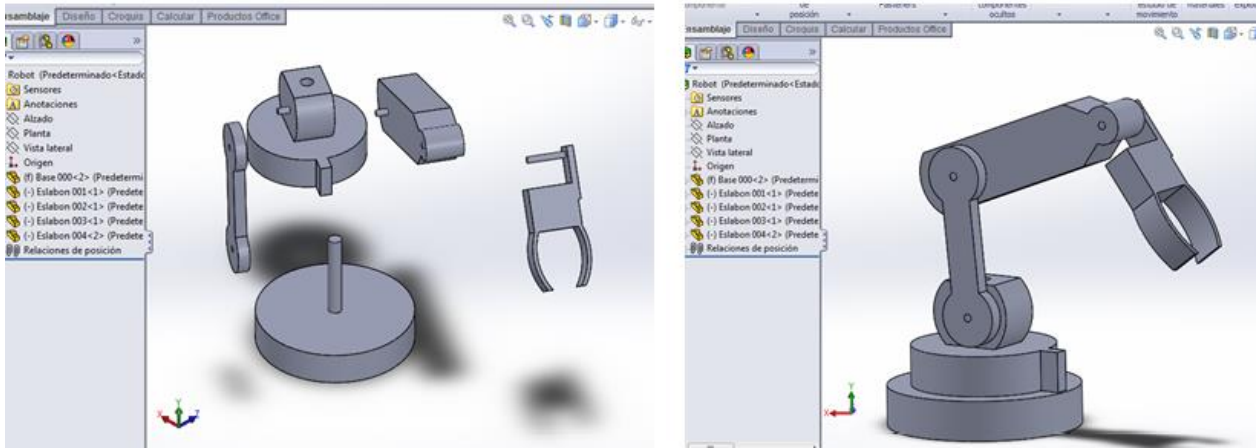


Imagen 65. Izquierda: Piezas sin ensamblar, Derecha Robot ensamblado

---

### 1.8.8 Diagrama de software

Como se aprecia en la siguiente imagen podemos apreciar el funcionamiento, al iniciar el sistema para controlar el movimiento de cada articulación del brazo, lo primero que se observa es el valor de cada sensor que es enviado a la placa Arduino y este al serial monitor de Arduino, la programación interpreta la señal analógica y la convierte en un valor digital con una rango de valores de 0-1024, tomando como ideal referencia 500, y posteriormente dar movimiento por el usuario y deteniéndolo mediante el mismo serial monitor de Arduino al insertar variables de entrada.

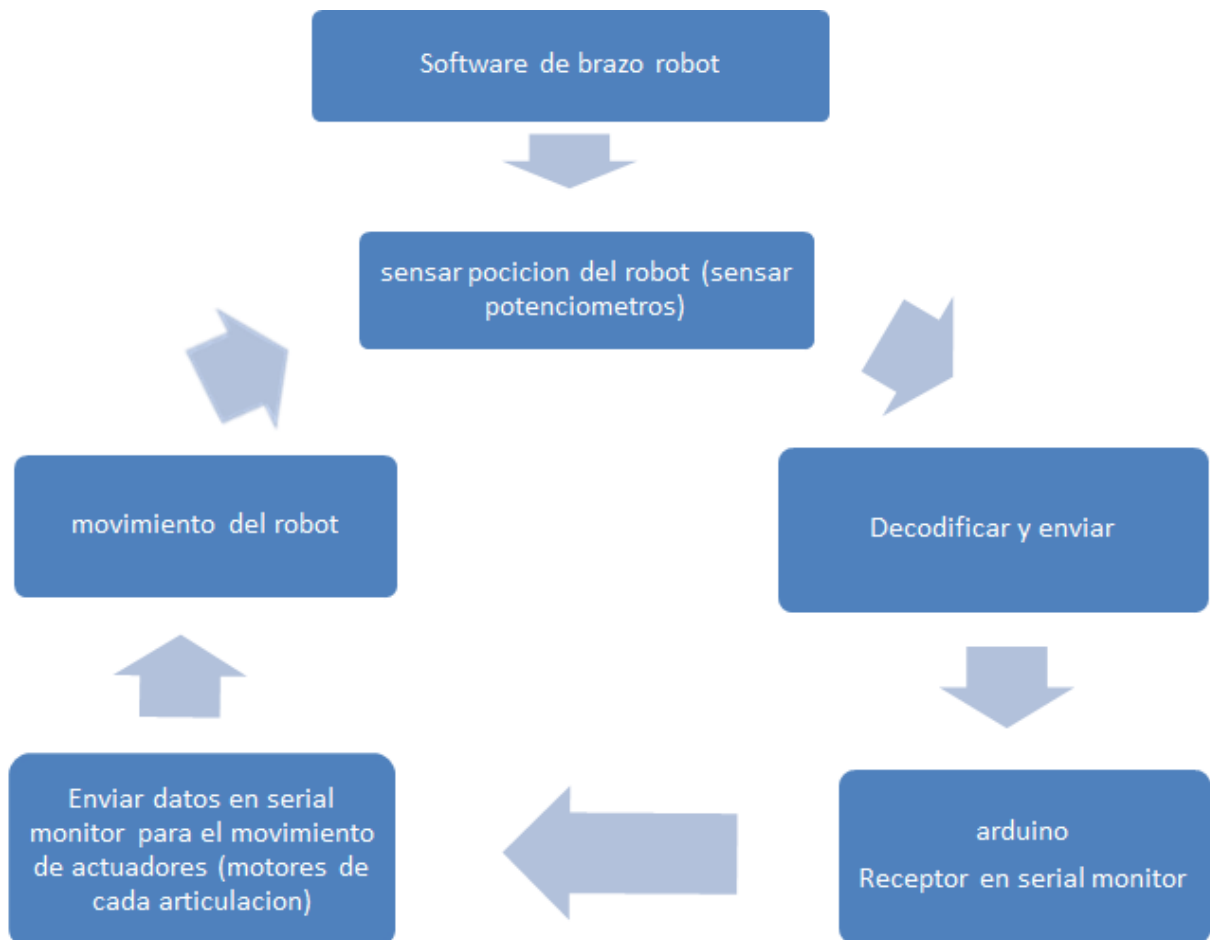


Imagen 66. Diagrama de software

---

### 1.8.9 programación de todos los sensores en conjunto

Al haber montado todos los componentes y posteriormente haber conectado los sensores y motores tal como se observa en la imagen 59, en donde se especifican las conexiones de las entradas y salidas de Arduino hacia los potenciómetros y puentes H (que están conectados a los motores) tal como se mostró en las imágenes de montaje y acople de potenciómetros en la imagen 67. Entonces procedemos a generar un código en la plataforma Arduino para poder sincronizar cada uno de los sensores y salidas a los actuadores, además de visualizar en el serial monitor los valores de entrada de cada sensor, y también enviar información para el movimiento de los motores.

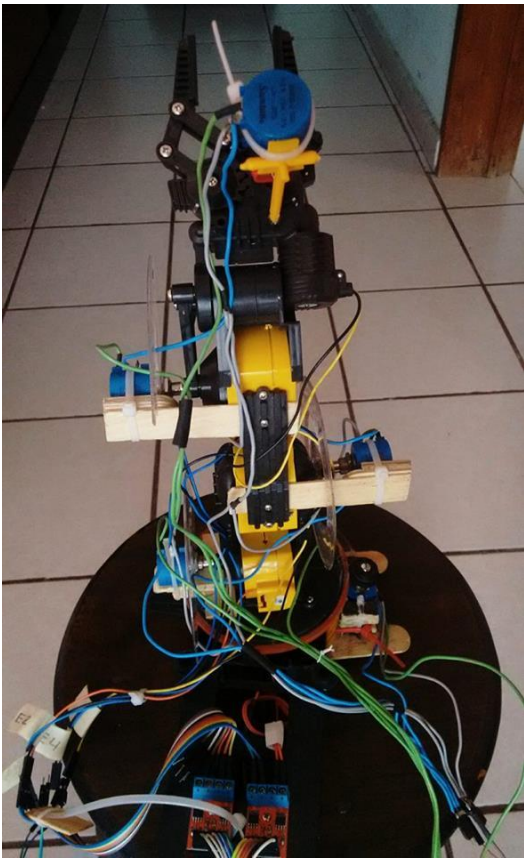


Imagen 67. Puentes H conectados a los motores y potenciómetros a Arduino.

---

### 1.8.9.1 Programación de control manual por medio de Arduino

En Arduino se programaron en base a el programa de censado del primer potenciómetro y logramos hacer el censado de cada potenciómetro, tomando en cuenta que al ser sensores analógicos los resultados de estos son convertidos y enviados a serial monitos en una escala de 0-1024, la programación se aprecia en la Imagen 68.

```
// la rutina de setup corre una vez o cuando se presiona reset
void setup() {
// inicializa la comunicación serial en 9600 bits por segundo
  Serial.begin(9600);
}

// la rutina loop corre constantemente
void loop() {
  // lee la entrada analógica desde el pin 0
  int esla1 = analogRead(A1);
  int esla2 = analogRead(A2);
  int esla3 = analogRead(A3);
  int esla4 = analogRead(A4);
  int garra = analogRead(A5);

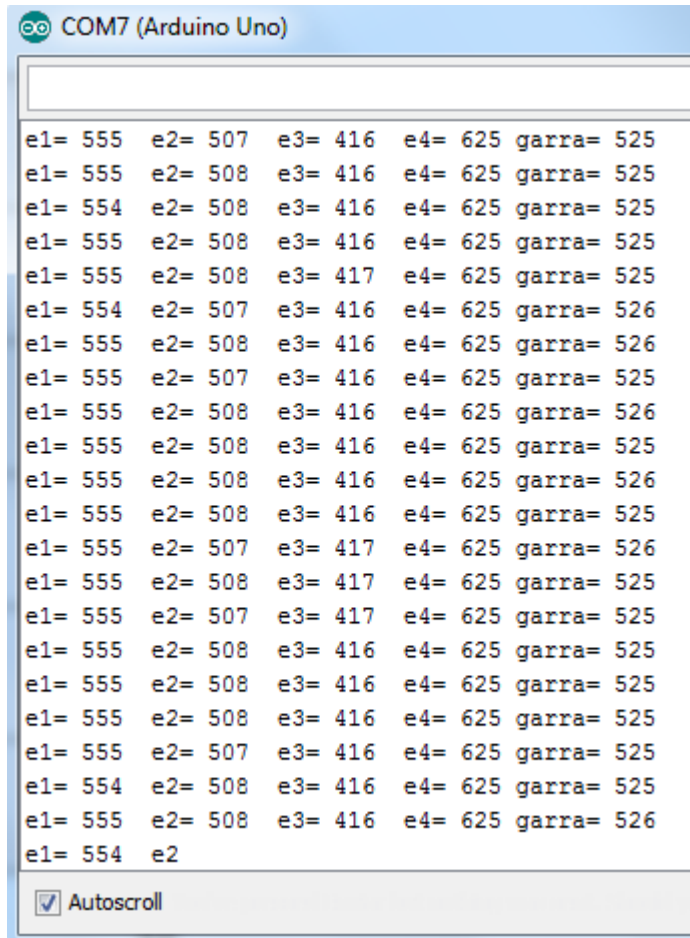
  // muestra el valor que se leyó
  Serial.print("e1= ");
  Serial.print(esla1);
  Serial.print(" e2= ");
  Serial.print(esla2);
  Serial.print(" e3= ");
  Serial.print(esla3);
  Serial.print(" e4= ");
  Serial.print(esla4);
  Serial.print(" garra= ");
  Serial.println(garra);
  // retraso entre lectura y lectura, para la estabilidad
}
}
```

Imagen 68. Programación de lectura de sensores.

La intención de la programación anterior es mostrar en serial monitor cada valor de cada potenciómetro con un valor que va de 0-1024 como se ha mencionado anteriormente. En la imagen 69 se puede apreciar la lectura en serial monitor donde se muestra que el valor de cada eslabón está representado con una abreviatura. Para

---

referirse a eslabón uno (base o cintura) se utilizó la abreviatura esla1, para el eslabón dos la abreviatura esla2, para el eslabón tres la abreviatura esla3, para el eslabón cuatro la abreviatura esla4 y la pinza la abreviatura garra, y cada una abreviatura seguida del signo igual y el valor censado.



```
COM7 (Arduino Uno)
e1= 555 e2= 507 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 525
e1= 554 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 417 e4= 625 garra= 525
e1= 554 e2= 507 e3= 416 e4= 625 garra= 526
e1= 555 e2= 508 e3= 416 e4= 625 garra= 526
e1= 555 e2= 507 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 526
e1= 555 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 526
e1= 555 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 507 e3= 417 e4= 625 garra= 526
e1= 555 e2= 508 e3= 417 e4= 625 garra= 525
e1= 555 e2= 507 e3= 417 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 507 e3= 416 e4= 625 garra= 525
e1= 554 e2= 508 e3= 416 e4= 625 garra= 525
e1= 555 e2= 508 e3= 416 e4= 625 garra= 526
e1= 554 e2
```

Imagen 69. Valor del sensor en cada uno de los eslabones y garra

## Programación Final

Posteriormente lo que se realiza es la programación que ara que Arduino envíe al controlador Para Motor DC modelo Stepper L9110 Lineal que es el pequeño sistema de potencia, y este a su vez energizara a los motores. A continuación se muestra la

---

programación terminada la cual es capaz de mover el robot y detenerlo cuando el usuario lo requiera, en esta programación se pusieron límites de movimiento, es decir las limitaciones para seguridad del robot

```
int input;
// la rutina de setup corre una vez o cuando se presiona reset
void setup() {
  pinMode(2, OUTPUT); // Declaramos que utilizaremos el pin como salida
  pinMode(3, OUTPUT); // Declaramos que utilizaremos el pin como salida
  pinMode(4, OUTPUT); // Declaramos que utilizaremos el pin como salida
  pinMode(5, OUTPUT); // Declaramos que utilizaremos el pin como salida
  pinMode(6, OUTPUT); // Declaramos que utilizaremos el pin como salida
  pinMode(7, OUTPUT); // Declaramos que utilizaremos el pin como salida
  pinMode(8, OUTPUT); // Declaramos que utilizaremos el pin como salida
  pinMode(9, OUTPUT); // Declaramos que utilizaremos el pin como salida
  // inicializa la comunicación serial en 9600 bits por segundo
  Serial.begin(9600);
}

// la rutina loop corre constantemente
void loop() {

  input=Serial.read();
  // lee la entrada analógica de cada pin
  int esla1 = analogRead(A1);
  int esla2 = analogRead(A2);
  int esla3 = analogRead(A3);
  int esla4 = analogRead(A4);
  int garra = analogRead(A5);

  // muestra el valor que se leyó
  Serial.print("e1= ");
  Serial.print(esla1); //muestra el valor del sensor 1
  Serial.print(" e2= ");
  Serial.print(esla2); //muestra el valor del sensor 2
  Serial.print(" e3= ");
  Serial.print(esla3); //muestra el valor del sensor 3
  Serial.print(" e4= ");
  Serial.print(esla4); //muestra el valor del sensor 4
  Serial.print(" garra= ");
```

---

```

Serial.println(garra); //muestra el valor del sensor de la pinza

// limitaciones eslabon UNO
if (esla1 > 700){
  digitalWrite(2, LOW); //giro hacia la izquierda apagado cuando llegue a este valor
}
if (esla1 < 500){
  digitalWrite(3, LOW); //giro hacia la derecha apagado cuando llegue a este valor
}
////////////////////////////////////
// limitaciones eslabon DOS
if (esla2 < 495){ //delante
  digitalWrite(4, LOW); //giro hacia la izquierda apagado cuando llegue a este valor
}
if (esla2 > 514){ //atras
  digitalWrite(5, LOW); //giro hacia la derecha apagado cuando llegue a este valor
}
////////////////////////////////////
// limitaciones eslabon tres
if (esla3 >507 ){ // "w" atras
  digitalWrite(6, LOW); //giro hacia la izquierda apagado cuando llegue a este valor
}
if (esla3 < 480){ // "q" delnate
  digitalWrite(7, LOW); //giro hacia la derecha apagado cuando llegue a este valor
}
////////////////////////////////////
/* // limitaciones eslabon Cuatro Cuatro Cuatro
if (esla3 >< ){ // "1" atras
  digitalWrite(, LOW); //giro hacia la izquierda apagado cuando llegue a este valor
}
if (esla3 <> ){ // "2" delnate
  digitalWrite(, LOW); //giro hacia la derecha apagado cuando llegue a este valor
}
*/

if (Serial.available(>0){

  //ESLABON 1 UNO UNO UNO UNO MOVIMIENTOS Y PAROS // 450 - 750, 500 es 90 grados
  if (input=='z'){
    digitalWrite(3, HIGH); //giro hacia la izquierda encendido
    digitalWrite(2, LOW); //giro hacia la derecha apagado

  }
  if (input=='x'){

```

---

```

digitalWrite(3, LOW); //giro hacia la izquierda apagado
digitalWrite(2, HIGH); //giro hacia la derecha encendido

}
if (input=='c'){
  digitalWrite(3, LOW); //giro hacia la izquierda apagado
  digitalWrite(2, LOW); //giro hacia la derecha apagado
}
//ESLABON 2 DOS DOS DOS MOVIMIENTOS Y PAROS
if (input=='a'){ //atras
  digitalWrite(5, HIGH); //giro hacia la izquierda encendido
  digitalWrite(4, LOW); //giro hacia la derecha apagado
}
if (input=='s'){ //al frente
  digitalWrite(5, LOW); //giro hacia la izquierda apagado
  digitalWrite(4, HIGH); //giro hacia la derecha encendido
}
if (input=='d'){
  digitalWrite(5, LOW); //giro hacia la izquierda apagado
  digitalWrite(4, LOW); //giro hacia la derecha apagado
}
//ESLABON 3 TRES TRES TRES TRES MOVIMIENTOS Y PAROS
if (input=='q'){ //delante
  digitalWrite(7, HIGH); //giro hacia la izquierda encendido
  digitalWrite(6, LOW); //giro hacia la derecha apagado
}
if (input=='w'){ //atras
  digitalWrite(7, LOW); //giro hacia la izquierda apagado
  digitalWrite(6, HIGH); //giro hacia la derecha encendidoD
}
if (input=='e'){
  digitalWrite(7, LOW); //giro hacia la izquierda apagado
  digitalWrite(6, LOW); //giro hacia la derecha apagado
}
//ESLABON 4 CUATRO CUATRO CUATRO CUATRO MOVIMIENTOS Y PAROS
if (input=='1'){
  digitalWrite(9, HIGH); //giro hacia la izquierda encendido
  digitalWrite(8, LOW); //giro hacia la derecha apagado
}
if (input=='2'){
  digitalWrite(9, LOW); //giro hacia la izquierda apagado
  digitalWrite(8, HIGH); //giro hacia la derecha encendidoD
}
if (input=='3'){

```



---

```

digitalWrite(9, LOW); //giro hacia la izquierda apagado
digitalWrite(8, LOW); //giro hacia la derecha apagado
}

//GARRA GARRA GARRA
if (input=='4'){
digitalWrite(10, HIGH); //giro hacia la izquierda encendido
digitalWrite(11, LOW); //giro hacia la derecha apagado
}
if (input=='5'){
digitalWrite(10, LOW); //giro hacia la izquierda apagado
digitalWrite(11, HIGH); //giro hacia la derecha encendidoD
}
if (input=='6'){
digitalWrite(10, LOW); //giro hacia la izquierda apagado
digitalWrite(11, LOW); //giro hacia la derecha apagado
}

}

}

```

Al correr esta programación puedo mover cada uno de los eslabones tomaremos en cuenta que la posición para empezar recomendable para cada eslabón es cuando los sensores están cerca o iguales que 500 ya que se considera limitad, cambé mencionar que esta calibración la hace el operario pues en Arduino solo controlamos movimiento y paro, para poder calibrarlo automáticamente se necesita una programación de control,

Entonces ya podemos controlar el Angulo de giro de cada eslabón, poder hacer ciertos movimientos múltiples si en el Serial monitor escribimos combinaciones de todos los eslabones, haciendo pruebas logramos tener un movimiento óptimo para lo que se desea.

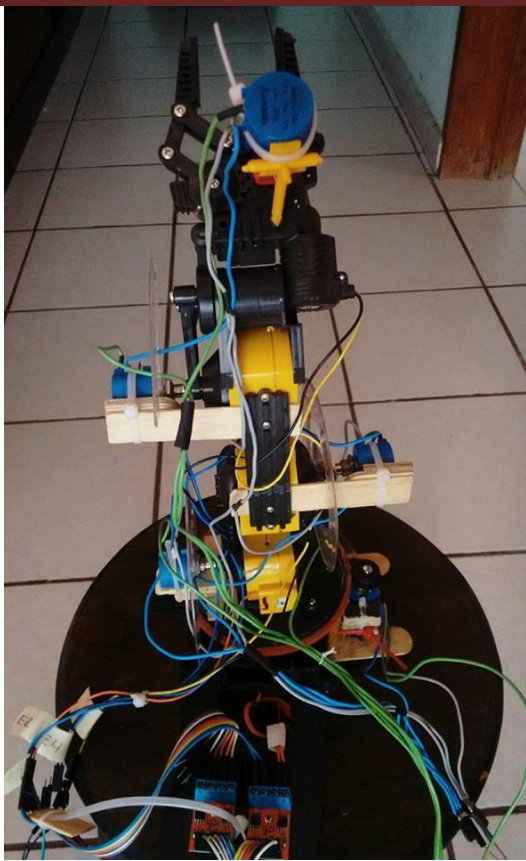


Imagen 70. Posición en la cual los sensores están todos mandando señales de al rededor o iguales a 500

## DESARROLLO DE SIMULADOR DEL ROBOT EN MATLAB

Para este simulador tomamos las piezas utilizadas en SolidWorks que se realizaron con anterioridad y con ayuda de las propiedades de Simulink de Matlab logramos exportar el modelo dinámico y hacer pruebas de simulación. En la imagen siguiente podemos ver como Simulink muestra el sistema mecánico y casa uno de las articulaciones, y también los eslabones y la dirección de movimiento, cabe mencionar que eliminamos variables externas como la gravedad, pero se dejaron informaciones como la base, el centro de masa de cada eslabón.

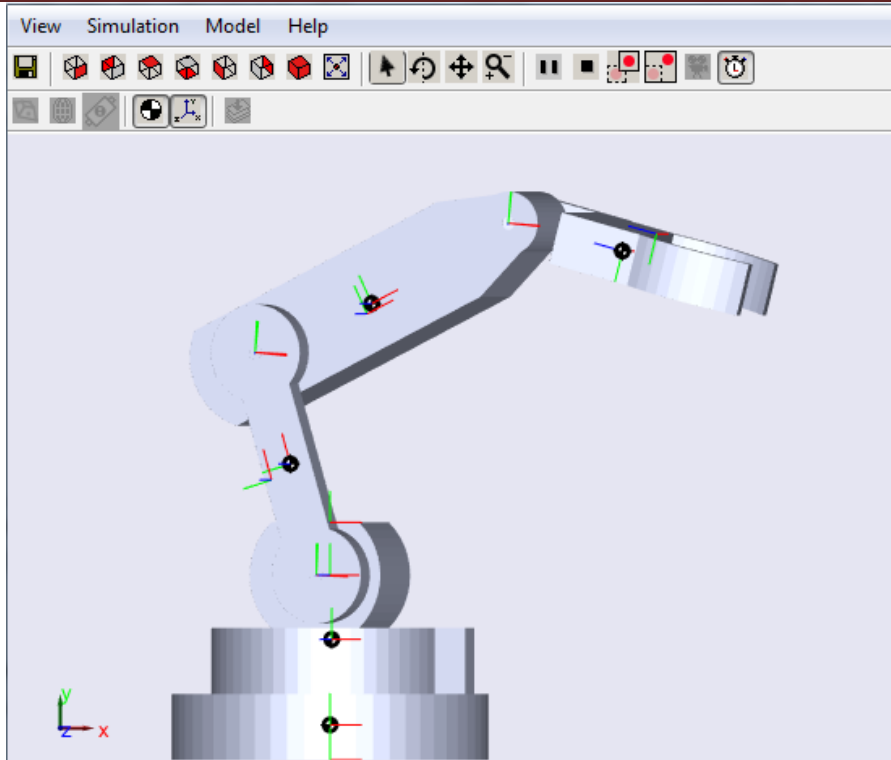


Imagen 71. Simulador de sistema mecánico en Simulink

Para este tipo de simulación se omitió la pinza ya que lo que importa es encontrar el movimiento de los eslabones

### Diagrama a bloques del sistema Mecánico

Para controlar cada uno de los eslabones se creó un sistema a bloques de cada uno. Como se puede ver en la imagen siguiente se puede ver el sistema total que es el robot, pero también se crearon subsistemas que mandan la información de cada eslabón, a estos se les llamo Subsystem , Subsystem 1, Subsystem 2, Subsystem 3, y se puede ver las entradas de cada uno

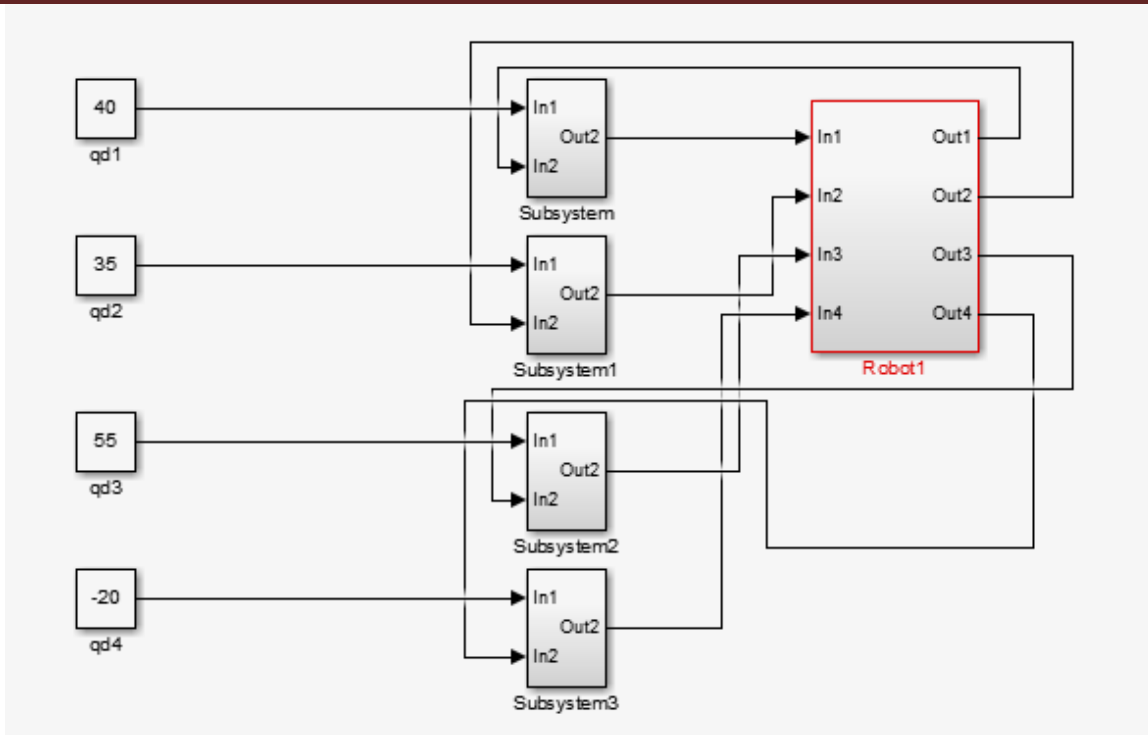


Imagen 72. Diagrama a bloque del sistema de control del modelo virtual

## Subsystem 1

A continuación podemos ver el diagrama a bloques de Subsystem , en este caso recibí asesorías del asesor externo, allí podemos ver la señal de entrada un sumador, integradores y lo necesario para poder mandar la salida y un elemento capas de retroalimentar para cuando se haya llegado al movimiento deseado, Que es este caso son Out2 y In2 como salida y entrada retroalimentativa.

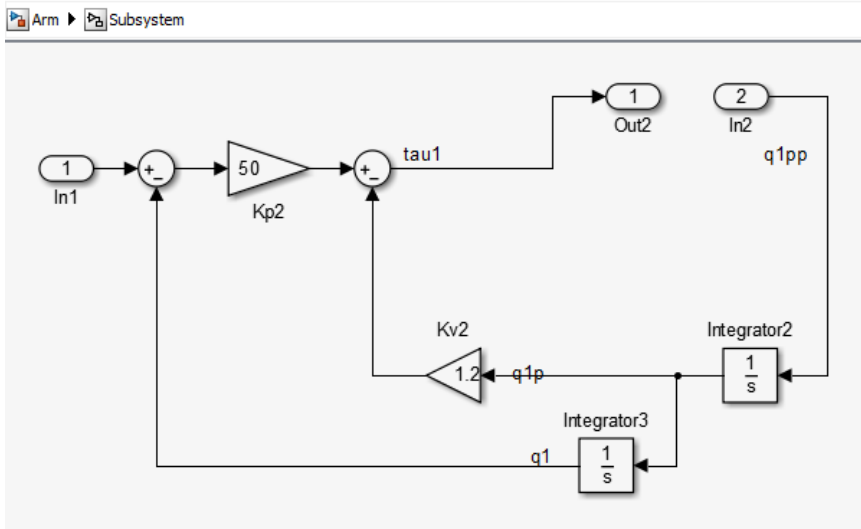


Imagen 73. Sistema de control para enviar señal al actuador de primer eslabón

### Subsystem 1

Para el Subsystem 1 es exactamente lo mismo.

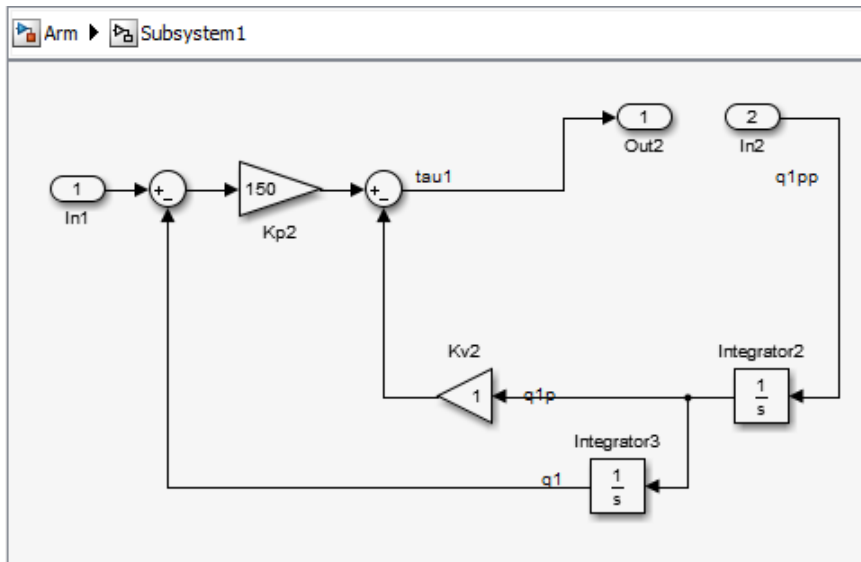


Imagen 74. Sistema de control para enviar señal al actuador de segundo eslabón

### Subsystem 2

Para el Subsystem 2 es exactamente lo mismo.

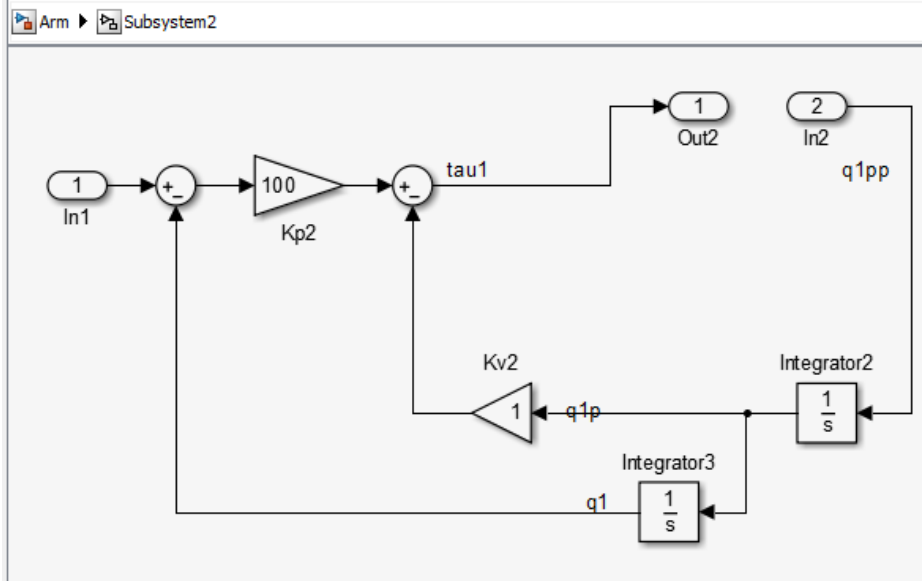


Imagen 75. Sistema de control para enviar señal al actuador de tercer eslabón

### Subsystem 3

Para el Subsystem 3 es exactamente lo mismo.

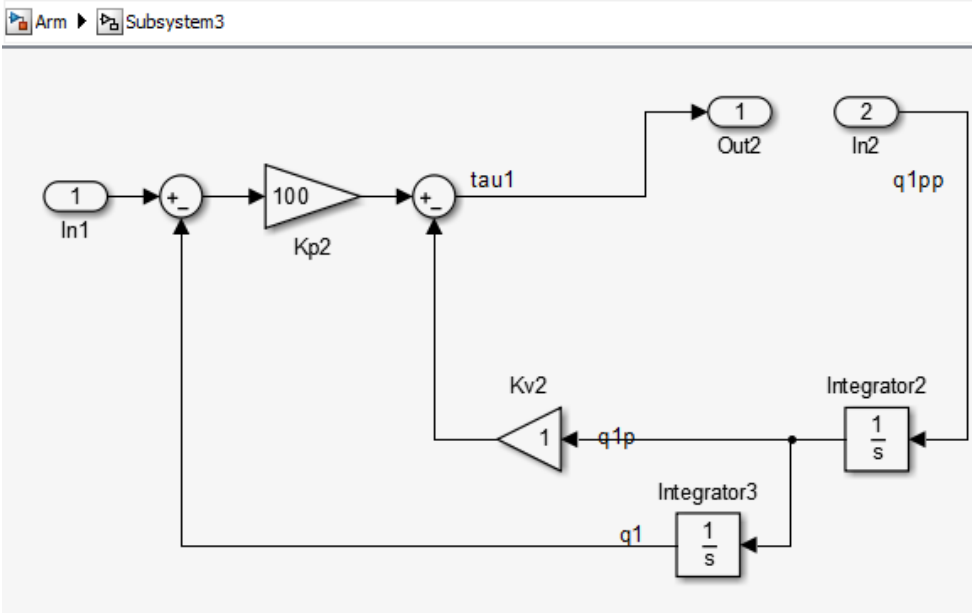


Imagen 76. Sistema de control para enviar señal al actuador de cuarto eslabón

---

## Robot 1

### Movimiento de eslabón 1

Para este sistema implementamos sensores y actuadores analizando al o que se utilizó físicamente, en la imagen siguiente podemos observar que el eslabón 1 esta después de la base y el eslabón es el que ara el movimiento físico de revolución dependiendo de la entrada que está representada como **In1** que proviene de Subsystem y este se dirige a un actuador, Dicho actuador ara el movimiento de revolución ya antes mencionado, y el sensor que esta también conectado a el eslabón que revoluciona es el encargado de detener el movimiento cuando este cumpla el valor que le fue enviado, y allí podemos observar que Out1 es la entrada del diagrama Subsystem que se retroalimentara el detendrá elproceso.

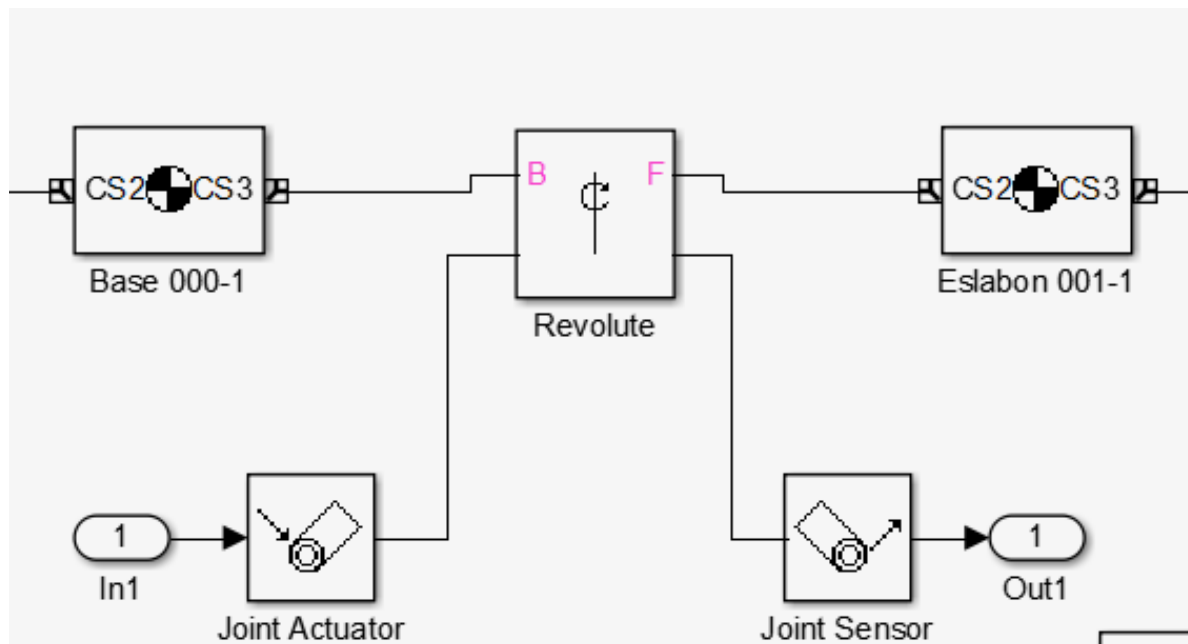


Imagen 77. sistema de control de movimiento de eslabón 1

---

## Movimiento de eslabón 2

Para este eslabón es exactamente lo mismo en cuanto al propósito de cada uno de las entradas actuadores y sensores, solo que la señal proviene deSubsystem 1

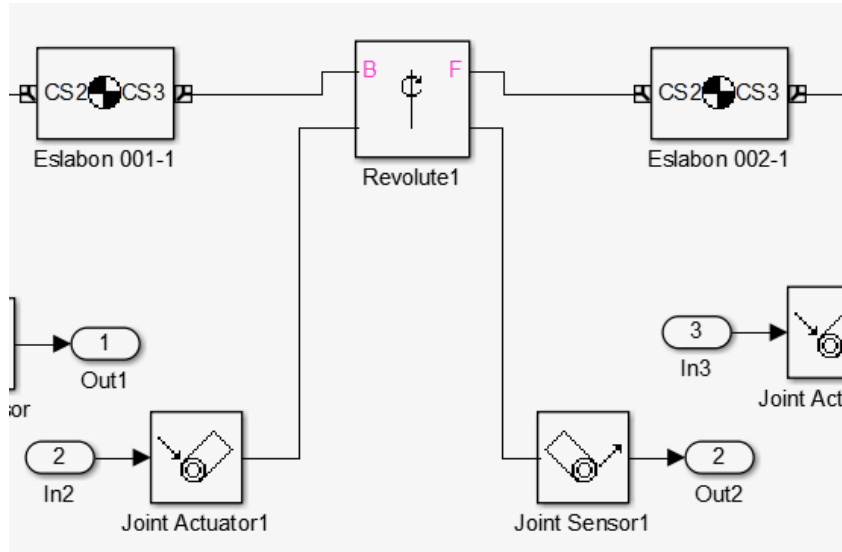


Imagen 78. Sistema de control de movimiento de eslabón 2

## Movimiento de eslabón 3

Para este eslabón es exactamente lo mismo en cuanto al propósito de cada uno de las entradas actuadores y sensores, solo que la señal proviene deSubsystem 2



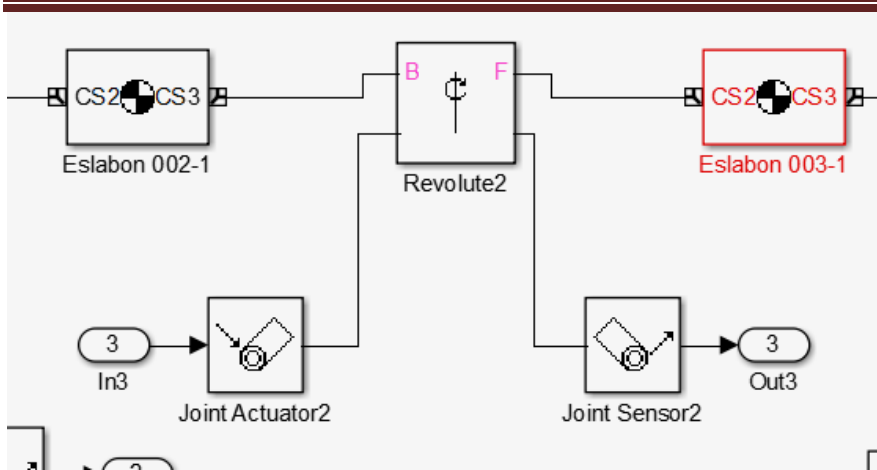


Imagen 79. Sistema de control de movimiento de eslabón 3

### Movimiento de eslabón 4

Para este eslabón es exactamente lo mismo en cuanto al propósito de cada uno de las entradas actuadores y sensores, solo que la señal proviene deSubsystem 3

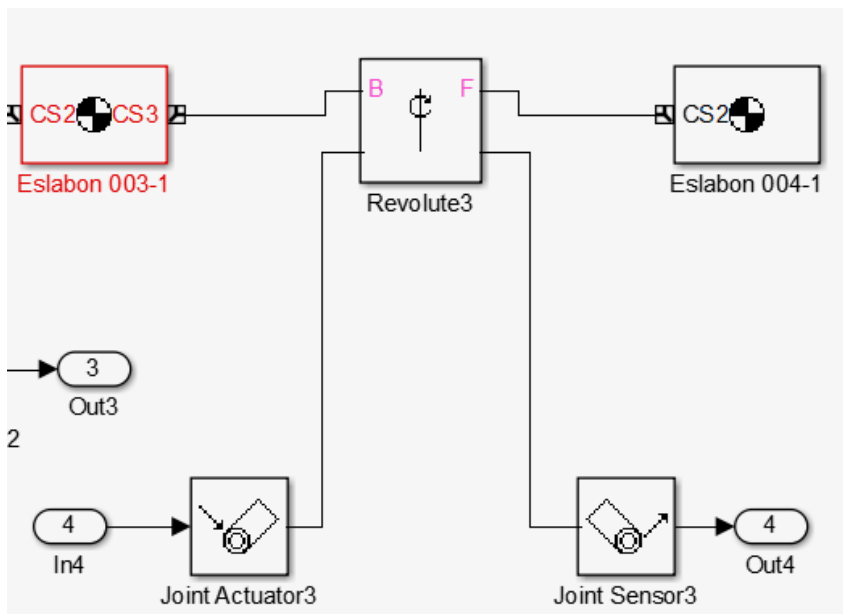


Imagen 80. Sistema de control de movimiento de eslabón 1

---

## Conexión final

Todas las configuraciones están interconectadas en serie y va a una salida, en la imagen 81 siguiente podemos observar que tenemos como partes fijas la base, y por lo tanto en Weld no ha actuadores ni sensores, lo mismo pasa en Robotpart que es el ensamble mecánico de todas las piezas y este hace k las piezas estén en su lugar y solo tengan movimiento las articulaciones.

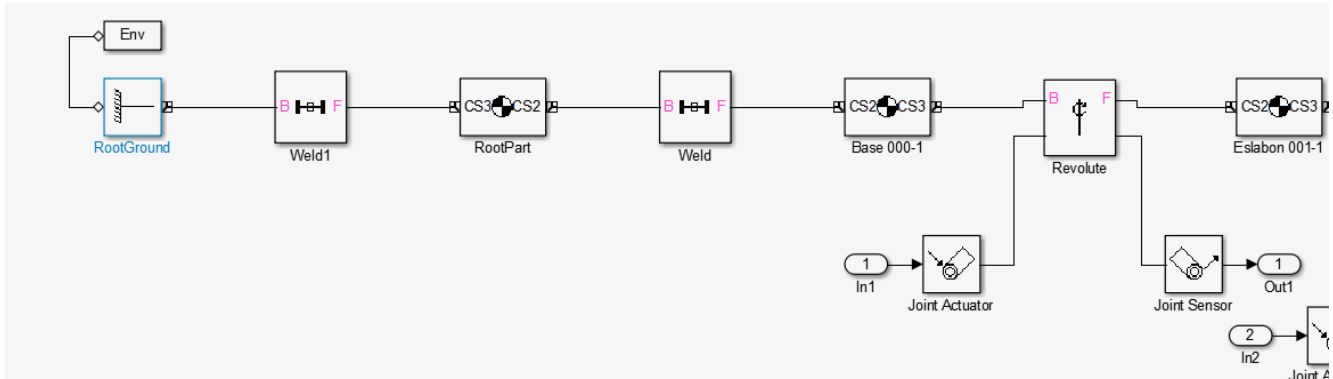


Imagen 81. Interconexión final

## Movimientos

A continuación podemos ver el movimiento logrado en cada uno de los eslabones y articulaciones.

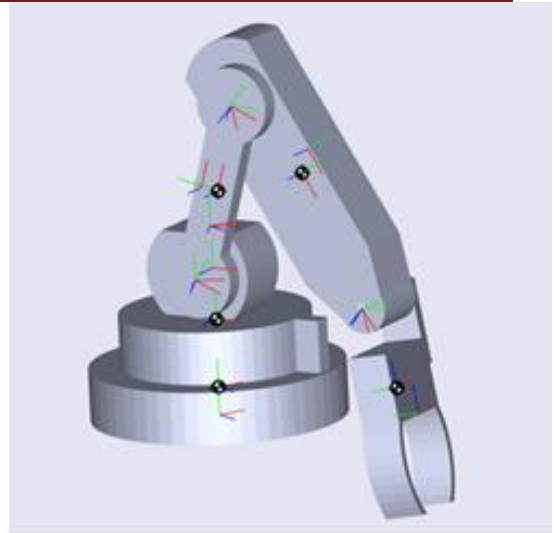
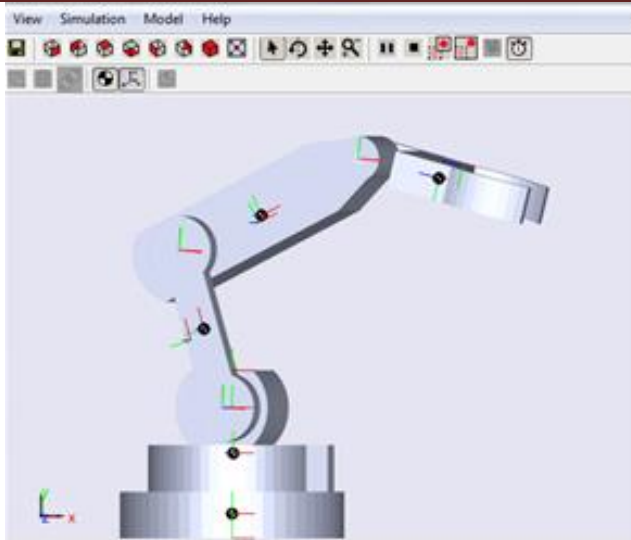


Imagen 82. Movimiento de simulador mecánico en Simulink

---

## 1.9 Evaluación o impacto económico, social o tecnológico

El impacto social en este caso sería para la comunidad estudiantil de ingeniería ya que con este modelo “didáctico” de implementación sobre el robot se pueden conocer nuevos métodos de monitoreo de sistemas mecánicos, y en ese sentido también puedo mencionar que el impacto tecnológico es sobre nuestros educandos, ya que habré muchas posibilidades, es decir en la carrera de Ing. Electrónica son escasas las materias en las que conocemos sobre interfaces graficas de sistemas mecánicos ya que en su mayoría somos conocedores de diseño y creación de sistemas eléctricos y electrónicos y no de este otro tipo, que si bien pertenece a otra rama de carreras como la mecatrónica o la mecánica, pero la suma de estos conocimientos crea un abanico de invenciones por implementar, como impacto económico diré que a mayor conocimiento sobre industrias, métodos y sistemas de control electrónicos podemos aspirar a la creación, adaptación y mantenimiento de tecnologías rentables propias o ya sea laborando en una empresa, generando a corto, mediano o largo plazo ingresos para el emprendedor o trabajador industrial.

## 1.10 Resultados

Como resultados finales podemos comentar a manera de descripción que el conocimiento básico para el funcionamiento, control y monitoreo de un robot es una tarea no complicada pero si meticulosa si se tiene los conocimientos previos, si lo que se desea es un buen funcionamiento, el movimiento de cada eslabón fue posible gracias a la programación en Arduino, los parámetros como el espacio de trabajo son casos complicados pero no difíciles de resolver puesto que lo que se evita es golpear el robot contra sus mismos componentes en sus diferentes configuraciones de movimiento, entonces como resultados tenemos que el movimiento del robot fue óptimo para el cumplimiento de objetivos, al censar los potenciómetros encontrando una relación de “Angulo de giro - resistencia” , El movimiento del robot fue un tanto rápido rápido y es de una poca resolución, si se deseaba mover un eslabón el rango de error es de  $\pm 5^\circ$ grados pues la velocidad, la inercia y el tiempo de respuesta crean un margen de error de movimiento, por otro lado el tener restringido ciertos ángulos de movimiento fue

---

muy efectivo ya que esto crea un sistema de protección para movimientos que pongan en peligro todo el mecanismo.

En el modelo virtual tenemos que ser necesario accesorio para convertir el modelo del ensamble mecánico a un sistema controlado en Simulink y además de esto crear el diagrama bloques correspondiente para controlarlo, conocimientos que no se terminaron de conocer del todo de mi parte.

## 1.11 Conclusiones y recomendaciones

Destacaremos que el objetivo siempre fue controlar el robot desde Arduino, y cuidando en mecanismo mecánico evitando colisiones en sus múltiples combinaciones de movimientos.

En el simulador gráfico de Simulink de Matlab encontramos que se debe tener conocimientos de diagramas a bloques y conocimiento sobre control, para poder idear la manera de moverlo y en este sentido recibí ayuda del asesor externo ya que encontraba complicaciones por falta de conocimiento en esto, una de las limitaciones fue que no se hicieron los análisis matemáticos y físicos para encontrar la ecuación del modelo dinámico para poder hacer una interfaz gráfica, Una vez echo los cálculos matemáticos del sistema físico haciendo los procedimientos necesarios entonces el proyecto en sí sería más robusto y fundamentado.

## 1.12 Bibliografía

### Fuentes electrónicas

1. Universidad de Atacama. “MORFOLOGÍA DEL ROBOT”. [industriaynegocios.cl](http://industriaynegocios.cl). 16/12/2015

[http://www.industriaynegocios.cl/Academicos/AlexanderBorger/Docts%20Docencia/Seminario%20de%20Aut/trabajos/2004/Rob%C3%B3tica/seminario%202004%20robotica/Seminario\\_Robotica/Documentos/MORFOLOG%C3%8DA%20DEL%20ROBOT.htm](http://www.industriaynegocios.cl/Academicos/AlexanderBorger/Docts%20Docencia/Seminario%20de%20Aut/trabajos/2004/Rob%C3%B3tica/seminario%202004%20robotica/Seminario_Robotica/Documentos/MORFOLOG%C3%8DA%20DEL%20ROBOT.htm)

2. MecanESO. “Sistema de levas”. [concurso.cnice.mec.es](http://concurso.cnice.mec.es). 2005. 17/12/2015

[http://concurso.cnice.mec.es/cnice2006/material107/mecanismos/mec\\_levas.htm](http://concurso.cnice.mec.es/cnice2006/material107/mecanismos/mec_levas.htm)

3. AREA TECNOLOGIA “potenciometro”. [areatecnologia.com](http://areatecnologia.com). 14 de diciembre de 2015

<http://www.areatecnologia.com/electronica/potenciometro.html>

- 
4. platea.pntic.“Robots industriales”. platea.pntic.mec.e. 28 de noviembre de 2015  
[http://platea.pntic.mec.es/vgonzale/cyr\\_0708/archivos/\\_15/Tema\\_5.4.htm](http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.4.htm)
  5. Universidad de Guadalajara.” Estructura Mecánica de un Robot”.proton.ucting.udg.mx.  
25 de noviembre de 2015  
<http://proton.ucting.udg.mx/materias/robotica/r166/r66/r66.htm>
  6. El Centro de Tesis, Documentos, Publicaciones y Recursos Educativos más amplio de la Red  
“Robótica Industrial”. monografias.com.6 de diciembre de 2015  
<http://www.monografias.com/trabajos99/robotica-industrial/robotica-industrial.shtml>
  7. Arduino. “QUÉ ES ARDUINO”.arduino.cl. 1 de diciembre de 2015  
<http://arduino.cl/que-es-arduino/>
  8. Galeon. “SolidWorks®”.solidworks.galeon.com. 4 de diciembre de 2015  
<http://solidworks.galeon.com/>
  9. Felipe “Matlab”. monografias.com. 3 de diciembre de 2015  
<http://www.monografias.com/trabajos5/matlab/matlab.shtml>
  10. Marco Fidel Orozco, mecánico industrial Sena “reductores y motorreductores”.  
monografias.com. 2 de diciembre de 2015  
<http://www.monografias.com/trabajos13/reducty/reducty.shtml>

## **Libros**

1. paul H. Lewis,C.Y.(2011).Sistemas de control en ingenieria(3 ed.).Madrid:Prentice-Hall
2. Gaviño.,R.H(2010).Introducción a los sistemas de control aplicación y simulación con matlab(1ed.).Mexico,estado de Mexico:PERSON.