UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DISEÑO Y DESARROLLO DEL PROTOTIPO DE UN BRAZO ROBÓTICO ARTICULADO ANTROPOMÓRFICO CONTROLADO MEDIANTE UN MICROCONTROLADOR ARDUINO MEGA 2560

AUTORA: IRIS VILA CASTELLÁ

TUTOR: FRANCISCO EUGENIO ALBERT GIL

COTUTORA: MARÍA NURÍA ALEIXOS BORRÁS

Curso Académico: 2017-18

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

AGRADECIMIENTOS

"A mis padres y a mi hermano por apoyarme y alentarme incondicionalmente"

"A mi tutor y al técnico de laboratorio por haber estado siempre pendientes de cómo avanzaba el proyecto y por haberme resuelto todas las dudas que me iban surgiendo"

"A Guille por animarme y admirarme en todo aquello que hago"

"A Eli por haberme acompañado en esta etapa y ser un apoyo constante todos estos años, dentro y fuera del ambiente de la universidad"

"A Gemma, Jorge y Paula por confiar en mi desde el principio y estar siempre disponibles"

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

RESUMEN

Este proyecto consiste en el diseño y desarrollo de un brazo robot antropomórfico con 5 GDL que permite el agarre de una pieza dentro de su área de trabajo y el traslado de la misma mediante las órdenes enviadas desde una aplicación móvil.

Para la realización de este trabajo se han seguido las siguientes pautas:

- Elección de los elementos a emplear, tanto eléctricos como de programación.
- Diseño de las todas las partes del conjunto en software de CAD 3D
- Simulación del comportamiento mecánico del conjunto.
- Impresión de las piezas y montaje de la estructura
- Montaje del circuito eléctrico sobre la estructura
- Programación de la placa Arduino y de la aplicación móvil para conseguir el manejo del robot

Palabras clave

Diseño, CAD 3D, Impresión 3D, Programación de microcontroladores

RESUM

Aquest projecte consisteix en el disseny i desenvolupament d'un braç robòtic antorpomòrfic amb 5 GDL que permiteix que s'agafe una peça dins del seu àrea de treball i el trasllat de la mateixa mitjançant les ordres manades des d'una aplicació mobil.

Per a la realització d'aquest treball s'han seguit les següents pautes:

- Elecció dels elements a emprar, tant elèctrics com de programació
- Disseny de totes les parts del conjunt amb software de CAD 3D
- Simulació del comportament mecànic del conjunt
- Impressió de les peces i muntatge de l'estructura
- Muntatge del circuit elèctric sobre la estructura
- Programació de la placa Arduino i de l'aplicació mobil per a aconseguir el maneig del robot

Paraules clau

Disseny, CAD 3D, Impressió 3D, Programació de microcontroladors

ABSTRACT

This project consists in the design and development of an anthropomorphic robot arm with 5 DOF that allows the grip of a piece within its work area and the translation of it through the orders sent from a mobile application.

To carry out this work, the following guidelines have been followed:

- Choice of the elements to be used, both electrical and programming.
- Design of all the parts of the set in 3D CAD software
- Simulation of the mechanical behavior of the set.
- Printing of the parts and assembly of the structure
- Assembly of the electrical circuit on the structure
- Programming of the Arduino board and the mobile application in order to get the handling of the robot

Key words

Design, 3D CAD, 3D printing, Microcontroller programming

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

DOCUMENTOS CONTENIDOS EN EL TFG

- Memoria
- Presupuesto
- Planos
- Anexo de programación
- Anexo interfaz usuario

ÍNDICE DE LA MEMORIA

ÍNDICE DEL PRESUPUESTO

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

ÍNDICE DE FIGURAS DE LA MEMORIA

ÍNDICE DE TABLAS DE LA MEMORIA

ÍNDICE DE TABLAS DEL PRESUPUESTO

Memoria

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Capítulo 1-Introducción al proyecto

1.1-Objetivo del proyecto

El objetivo de este proyecto es el diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560, basándonos en la cinemática directa. A partir de este objetivo, se proponen los subobjetivos siguientes:

- Diseñar una estructura simple para que pueda ser impresa en 3D
- Estudiar las fuerzas, cargas y movimientos máximos que el brazo es capaz de soportar. De esta manera se estudiarán los puntos críticos de la estructura y se podrán proporcionar mejoras.
- Construir un prototipo virtual mediante el modelado 3D y el ensamblaje de las piezas
- Proporcionar a los diferentes brazos de los actuadores necesarios para conseguir su correcto movimiento
- Controlar el movimiento del brazo mediante una aplicación móvil de fácil uso.

1.2-Motivación del proyecto

La principal motivación para la realización de este TFG es el desarrollo de cada una de las partes de un proyecto, desde su creación y su diseño, pasando por el estudio de fuerzas, hasta su montaje y elaboración del circuito que lo controla. De esta manera, se aplicaran conocimientos de las asignaturas de Tecnología eléctrica, Ingeniería Gráfica, Informática e Impresión 3D.

Capítulo 2-Materiales y métodos

2.1- Aspectos teóricos de un brazo robot

2.1.1- Breve historia de los brazos robot

Desde miles de años atrás los seres humanos han intentado diseñar aparatos o máquinas que simulen comportamientos humanos.

Desde la cultura árabe (siglos VIII a XV) donde se crearon mecanismos destinados a aplicaciones prácticas como dispensadores automáticos de agua, pasando por el renacimiento y todas las invenciones creadas por sus representantes más relevantes como Leonardo Da Vinci, quien creó numeroso ingenios mecánicos, hasta la actualidad, donde se encuentra robótica muy sofisticada y especializada.

La palabra robot, la cual proviene del eslavo y alude al trabajo que se realiza de manera forzada, se usó por primera vez en una obra denominada *Rossum's Universal Robot* del escritor checo Karel Capek en el 1921.

2.1.2-Grados de libertad

Mecánicamente, un robot se forma con la unión diferentes partes o piezas mediante articulaciones. Dichas articulaciones posibilitan el movimiento relativo entre los elementos que se unen.

Exiten diferentes tipos de articulaciones que surgen de la combinación de desplazamientos y giros en diferentes direcciones. De este modo se encuentran 6 tipos diferentes de articulaciones, siendo las más empleadas la prismática y la de rotación.

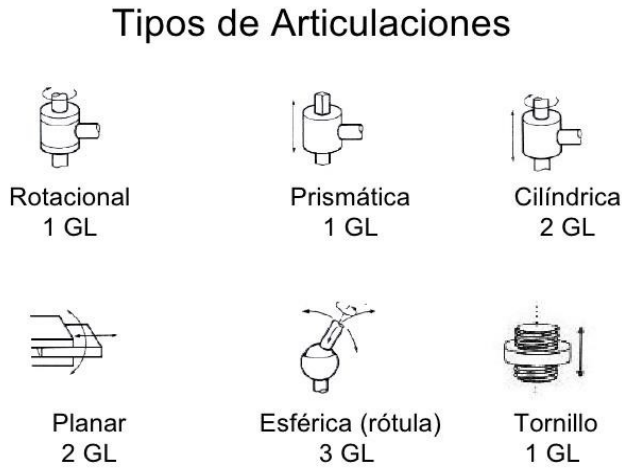


Figura 1. Tipos de articulaciones

[Fuente: https://es.slideshare.net/mgarofalo85/robotica-2865379]

Los movimientos que pueden realizar cada una de las articulaciones que posee un mismo robot con idependencia se denominan grados de libertad (GDL). Además el número total de GDL de un robot es la suma de los GDL de cada articulación que lo compone. A partir de las diferentes combinaciones de articulaciones surgen diferentes configuraciones para un robot que proporcionaran unas características de movimiento diferentes.

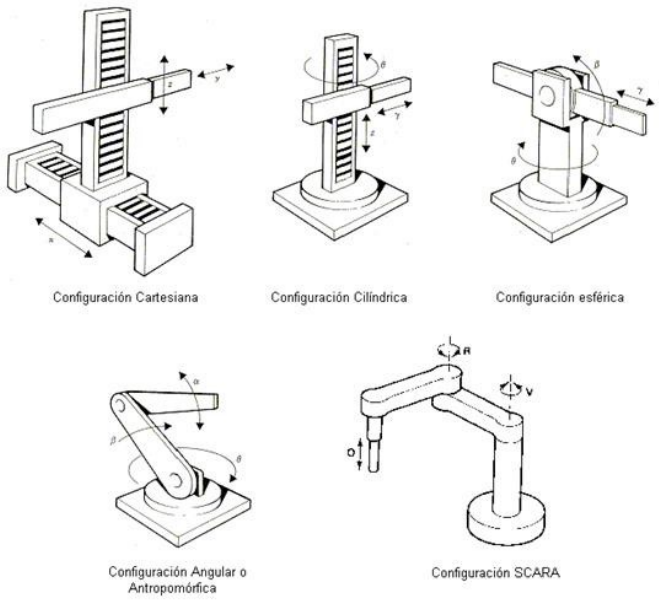


Figura 2 Tipos de configuraciones

[Fuente: https://tesisdegradohectorc2.es.tl/Capitulo-2.htm]

Para poder localizar y orientar el extremo de un robo t(o la pieza que manipula) en el espacio se necesitaran al menos 6 GDL. Esto se debe a que para posicionar y orientar cualquier cuerpo es imprescinble establecer 3 parámetros que definen la posición y 3 parámetros que definen la orientación.

Aun así, muchos robots industriales cuentan sólo con 4 o 5 GDL, ya que son no se requieren más GDL para las tareas que realizan.

Además, existen robots que necesitan tener más de 6 GDL para que pueda acceder a toda el área de trabajo que se le precisa. Por ello, los GDL adicionales permitirán, a los robots que se encuentran en lugares de trabajo obstaculizados, poder acceder a aquellas zonas que no hubieran podido llegar con tan solo 6 GDL.

Por otro lado, se dice que un robot es redundante si el número de GDL del robot es superior a los que precisa para hacer la tarea.

2.1.3-Capacidad de carga

La capacidad de carga de un robot vendrá marcada por su misma estructura, es decir, por el tamaño de sus piezas, la configuración que tengan y el sistema de accionamiento.

Además, en el momento de evaluar la carga que tendrá que manipular el robot, es necesario tener en cuenta el peso propio de la herramienta que empleará el robot para trabajar y el peso de la pieza que tendrá que manipular.. Por otra parte, un factor importante que se ha de tener en cuenta es el momento que se genera cuando se transporta una pieza en el extremo del robot.

Es habitual que el fabricante proporcione el dato de la carga nominal que el robot. Esta carga indica el peso que el robot es capaz de transportar sin disminuir sus características dinámicas, en la situación más desfavorable del mismo.. Sin embargo, cabe la posibilidad de aumentar dicha carga hasta un cierto límite pero asumiendo una pérdida de velocidad y de precisión en los movimientos.

La capacidad de carga de los robots de uso industrial oscilan entre 5 y 50 kg, aunque existen algunos que transportan cargas muy grandes (más de media tonelada).

2.1.4-Resolución

La resolución es el mínimo incremento que se puede aplicar en el dato de entrada y se aprecia en dato de salida.

El valor de la resolución se encuentra limitado por diversos factores como pueden ser la resolución de los convertidores, el número de bits de las operaciones aritméticas de la CPU o los motores paso a paso.

2.1.5-Precisión

La precisión se define como la diferencia entre el dato de entrada que se envía y la media de los puntos que alcanza el brazo robot, repitiendo el mismo movimiento con la carga y la temperatura nominal.

La mayor o menor precisión de un robot surge de fallos de calibración, errores de redondeo en los cálculos, diferencias entre la dimesion real y teorica del robot, etc…

2.1.6-Repetibilidad

Se trata del radio de la zona de forma esférica que contiene a los puntos que alcanza el robot tras múltiples ensayos indicándole que se dirija hacia el mismo punto de destino y en las mismas condiciones. Es decir, la diferencia entre sucesivas medidas de la misma entrada.

El error de repetabilidad se debe a problemas mecánicos como pueden ser rozamientos o perdidas por histéresis.

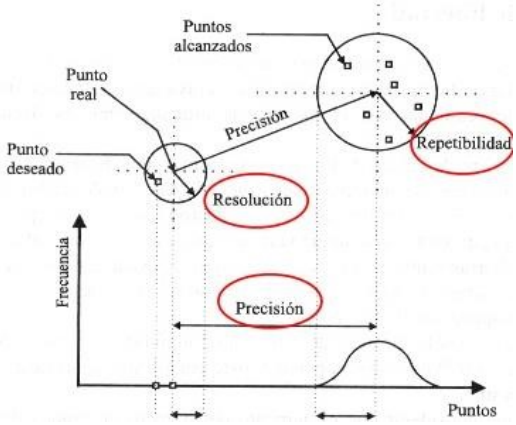


Figura 3. Gráfico sobre repetibilidad, resolución y precisión

[Fuente:https://es.slideshare.net/luispedraza/cim-07-produccin-fms-robtica-agvs-asrs]

2.1.7-Velocidad

La velocidad es una característica de los brazos robot inversamente proporcional a la carga que transporta. Este aspecto técnico de un robot puede encontrarse como la velocidad de cada articulación que conforma al brazo robot o como la velocidad media del extremo. Esta última es la que resulta más útil para el usuario aunque tiene más probablilidad de ser imprecisa.

El dato que se proporcionado en un robot es el de su velocidad nominal. Dicha velocidad se obtiene en régimen permanente, es decir, en un tiempo de movimiento suficientemente largo. Aunque se trata de un dato muy útil en los tiempos de ciclo, en la realidad los movimientos de los brazos robot suelen ser rápidos y cortos, por lo que no es habitual alcanzar el régimen permanente. Por ello, no se puede emplear el dato de la velocidad para obtener el tiempo de ciclo de un robot y, como consecuencia, los fabricantes indican el tiempo que se emplea en un movimiento común como un “pick & place” en vez de utilizar el dato de la velocidad.

Los valores frecuentes de velocidad se encuentran entre 1 m/s y 4 m/ con carga máxima.

2.1.8-Área de trabajo

El área de trabajo es volumen al que puede acceder el extremo del robot. Dichi volumen se determina con las características físicas del robot (tamaño, forma y tipo de sus piezas) y por las limitaciones de las que consta el mismo. Para el cálculo del área de trabajo no se tiene en cuenta la herramienta que se le añade al brazo robot ya que si se cambiara la herramienta se tendría que recalcular el área de trabajo.

Para elegir un robot u otro se debe tener en cuenta que el área de trabajo sea la necesaria para acceder a todos los puntos que necesita para poder realizar su tarea. El área de trabajo se indica mediante un

dibujo acotado o si se trata de información numérica, se representa el rango de recorrido que contiene cada articulación.

Por otro lado, hay que tener en cuenta que existirán puntos que el robot solo alcanzará con una orientación determinada.

Además, se encuentran puntos singulares en los que no es posible realizar, por ejemplo, una trayectoria rectilínea.

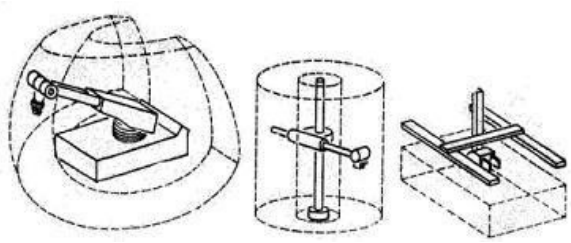


Figura 4. Área de trabajo

[Fuente: https://www.monografias.com/trabajos82/robotica-tipos-robot/robotica-tipos-robot2.shtml]

2.2-Modelado 3D

2.2.1- Autodesk Inventor

Inventor es una herramienta de modelado 3D desarrollada por Autodesk. Surgió en 1999 y se incorporóó a las Series de Diseño Mecánico de Autodesk.

Se basa en técnicas de modelado paramétrico, de esta manera, se pueden añadir tolerancias, notas y otros detalles al modelo 3D.

Los sistemas paramétricos se basan en restricciones que se capturan y resuelven en el orden en el que se han introducido: el diseño se controla por un grafo que refleja el proceso de creación de la pieza. Las variables que intervienen en una restricción algebraica deben de haber sido definidas en otra restricción; es decir cada entidad geométrica se posiciona con referencia a otras previamente definidas.

Las diferencias principales y ventajas que se encuentran en Inventor son la existencia de herramientas como:

-Design Accelerator: Empleado para acelerar el diseño de maquinaria y estandarizar componentes (diseño de poleas, engranajes, piñones…)

-Shape Generator: Es una herramienta que posibilita la validación de la capacidad de resistencia física de la pieza para así optimizar la forma y el peso de la estructura.

2.2.2- Simulación dinámica

Autodesk Inventor Simulation ofrece herramientas para poder simular y estudiar las características dinámicas de una estructura en movimiento con diferentes condiciones de carga. Esto supone una gran ventaja ya que con un mismo programa se puede diseñar y estudiar su funcionamiento y así, poder hacer las modificaciones que sean necesarias de una manera sencilla y rápida. La simulación nos muestra la respuesta del movimiento del conjunto a unas situaciones de carga indicadas desde un punto de vista estructural.

2.2.2.1-Teoría de elementos finitos

Se trata de un método numérico de gran utilidad en la práctica empleado para resolver problemas de mecánica de sólidos. A pesar de tener una gran funcionabilidad no se trata de un método exacto ya que ofrece soluciones aproximadas a los problemas. Aun así, dichas soluciones son suficientes a nivel práctico.

Se fundamenta en la subdivisión en pequeños elementos interconectados entre sí mediante los nudos de dichos elementos de un sólido que se encuentra sometido a un sistema de cargas. De esta manera, se estudia el desplazamiento en el interior de cada elemento en función del desplazamiento de los

nudos del mismo. A continuación, mediante la matriz de rigidez de cada elemento se obtiene los desplazamientos de los nudos de cada elemento y, por ello, se puede determinar las deformaciones y tensiones en el interior de los elementos.

Para este método se emplea una expresión del tipo:

$$\{F\} = [K]\{u\}$$

En dicha fórmula {F} se trata de un vector columna, [K] la matriz de rigidez de la estructura y {u} el vector de desplazamientos de los nudos.

2.2.3-Diseño de engranajes en Inventor

En este apartado se van a comentar las elecciones en el diseño de los diferentes engranajes que se necesitan para el funcionamiento del brazo robot.

Un engranaje viene definido por diversos conceptos que son fundamentales para el correcto cálculo de engranes. Entre ellos, se destacan los siguientes conceptos que, como se observa más adelante, son necesarios para crear los engranajes del robot:

-Módulo: Consta de la relación entre el diámetro primitivo (mm) y el número de dientes.

-Distancia entre centros: Es el equivalente a la distancia entre los ejes de los engranajes.

-Angulo de presión: Es el ángulo que forma la dirección de la fuerza de contacto entre los dientes y la dirección de la velocidad del engranaje conducido.

-Ángulo de hélice: Se define como la inclinación (grados) del diente en dirección longitudinal. En el caso en el que este dato sea 0 el diente es paralelo al eje de la rueda y, por tanto, se trata de un engranaje de dientes rectos.

-Relación de transmisión: Es a la relación entre las velocidades de rotación de los engranajes. Se comenta con más detalle en el momento de proceder al diseño de engranajes en el apartado “3.5-Modelado de los engranajes”.

Autodesk Inventor ofrece la posibilidad de crear conjuntos de engranajes mediante Design Accelerator. Cuando se selecciona en la pestaña superior “Diseño” la opción “Engranajes rectos”, que es la que nos interesa para el movimiento de nuestro conjunto, nos aparece una ventana emergente que nos ofrece la posibilidad de crear los engranajes de diferentes maneras.

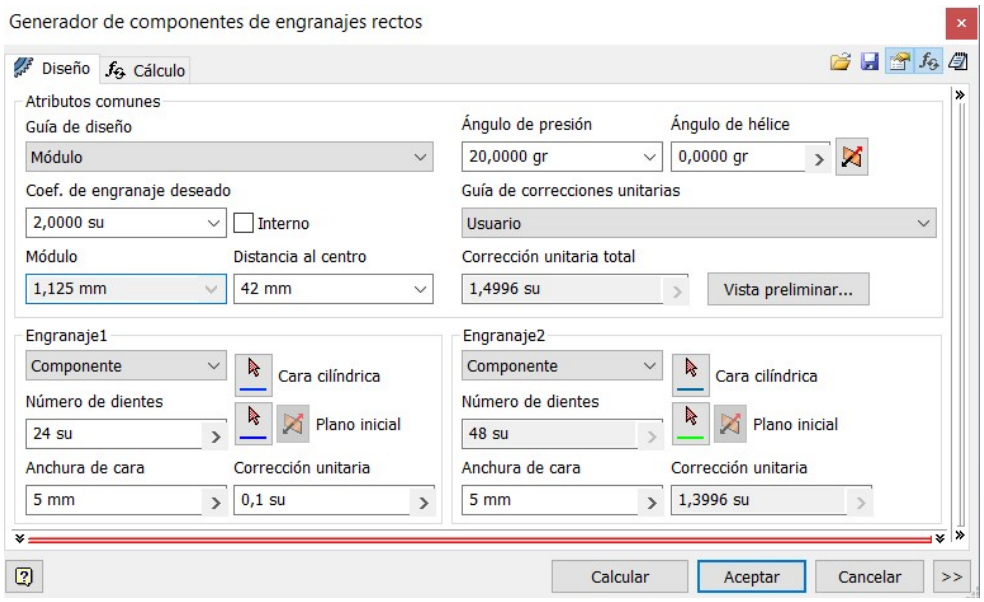


Figura 5. Ventana de diseño engranajes rectos

En la pestaña desplegable "Guía de diseño" se selecciona la opción que más nos convenga para la creación de los engranajes. Aquello que se seleccione será lo que no se pueda modificar y se fijará según el cálculo que haga Design Accelerator.

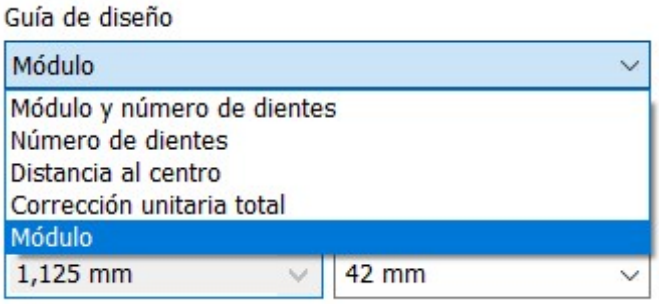


Figura 6. Ventana desplegable Guía de diseño

En nuestro caso, se ha seleccionado la opción de "Módulo y número de dientes" y, de esta manera, poder introducir la distancia entre centros y el coeficiente de engranaje deseado, que son los datos que más interesan.

2.3- Impresión de las piezas

2.3.3-Introducción a las impresoras 3D

Las impresoras 3D han supuesto una revolución importante en diversos ámbitos de la ciencia ya que son máquinas capaces de imprimir figuras con volumen a partir de un diseño que se realiza con algún programa de ordenador.

Este tipo de impresoras usa materiales de impresión como nylon, metales, ABS y muchas más tipos de materiales dependiendo de las condiciones que se necesiten para el producto. La mayoría de impresoras se basan en el funcionamiento de robots cartesianos. Estos robots permiten el movimiento en ejes ortogonales, siendo cada uno independiente del otro. Se puede tener máquinas de 3,4, incluso 5 ejes.

2.3.5-Aplicaciones

El factor interesante del uso de estas máquinas es la obtención de un buen resultado con un coste asequible. A pesar de que sea una aplicación que aún se encuentre en desarrollo en muchos campos, va creciendo a pasos agigantados. Hoy en día se usa la impresión 3D tanto para piezas de robots como para prótesis, incluso se está desarrollando para la construcción de edificios.

2.3.2-Funcionamiento

Las impresoras 3D basan su funcionamiento en la construcción de capas sucesivas hasta que se obtiene la pieza deseada. Este proceso el cual se denomina “proceso aditivo” se inicia con la creación de un archivo en 3d con un programa de modelado.

La ventaja de este tipo de funcionamiento es el hecho de que el coste por unidad es siempre el mismo, no encarece al disminuir o aumentar las unidades producidas.

2.3.4-Tipos de impresoras 3D

Aun que todas las impresoras usan el proceso aditivo, se pueden distinguir 3 tipos de impresoras 3D bien diferenciadas:

- Por disposición de material fundido o FDM

  Esta técnica consiste en depositar polímero fundido sobre una base plana, capa a capa. El material de impresión que se encuentra enrollado en estado sólido en unas bobinas, se funde y se expulsa por la boquilla en pequeños hilos que se solidifican conforme se crean las capas.
  Es la técnica más común de impresión en cuánto a uso de escritorio o individual. Se obtienen buenos resultados, pero las piezas no tienen un acabado comparable con los otros dos tipos de tecnología.

Entre los materiales más usados se encuentran el ABS y PLA.

- Estereolitografía o SLA

  Esta técnica se basa en el uso de un haz de luz ultravioleta a una resina líquida que se encuentra en un cubo. El haz de luz va solidificando la resina capa a capa. La base que soporta la estructura va desplazándose para que el haz de luz vuelva a actuar sobre un nuevo baño.
  Tiene la ventaja de que se consiguen piezas con una gran calidad a pesar de que se puede desperdiciar material dependiendo del soporte que se realice.

- Sinterización Selectiva por Láser (SLS)

  Tiene cierto parecido con la tecnología SLA pero con la ventaja de poder emplear diferentes tipos de materiales en polvo como cerámica o cristal.
  En esta técnica el láser impacta en el polvo, impactando sobre él y solidificándolo. Otra de sus ventajas es que no se desperdicia ningún material, ya que se almacena en el mismo lugar donde inicio la impresión.

- Por inyección

  Es la tecnología más parecida a una impresora habitual de tinta. En esta tecnología se inyectan capas de fotopolímero líquido hasta que se obtiene la pieza final. Se diferencia respecto a las anteriores en que no se compacta un material, sino se inyecta un aglomerante.

2.3.6-Fases de la impresión

Las fases que constan en la impresión de cualquier tipo de pieza son las siguientes:

1- Modelado 3D de una pieza: Para esta fase se usan programas de diseño como pueden ser OnShape, OpenScad o, en nuestro caso, Inventor. Durante esta etapa se ha de tener en cuenta diversos aspectos que faciliten la posterior impresión de las piezas. Por ello, es recomendable el diseño de piezas sencillas con base plana para evitar la creación de soportes de apoyo o pilares que tras la impresión se tendrían que eliminar con herramientas. Además, se debe tener en cuenta los grosores de las estructuras y su tamaño ya que cuanto más volumen abarque una pieza, más tiempo y material consumirá.

2- Convertir el archivo a STL: (Este formato define la geometría del objeto, a excepción del color, textura o propiedades físicas de otros formatos CAD. Es un formato estándar empleado en máquinas de fabricación aditiva. Se emplea una malla de triángulos cerrada para definir la forma. Conforme más pequeños se formen estos triángulos, mayor resolución y peso tendrá nuestra pieza.)

3- Conformación del código de impresión: Para la impresión de la pieza se necesita obtener el código G o G-Code. Este código es el lenguaje que se emplea para describir las operaciones que realizan las máquinas CNC. Es el lenguaje más empleado ya que se estandarizó en los 80 como ISO 6983. El G-Code se almacena como un texto así que puede leerse y modificarse.

4- Preparación de la máquina: Se prepara el material con el que se va a imprimir la figura. En nuestro caso, se emplean bobinas de PLA.

5- Impresión: Para esta fase se pueden seleccionar varios parámetros de impresión. Una vez seleccionados, la impresión es un proceso automático.

6- Obtención de la pieza y post-proceso: Una vez finalizada la impresión, se tiene que dejar un tiempo de reposo y extraer la pieza. A continuación se tendrá que extraer el material excedente que se ha formado como apoyo para la creación de la pieza.

2.3.7-Repetier

El Repetier es un programa que se emplea para enviar la pieza (en formato STL) a la impresora 3D. Este programa se encarga de convertir nuestro archivo 3D al código que sea necesario para que pueda ser impreso.

En este software u otros del mismo tipo se puede importar uno o más modelos, rotarlos o duplicarlos. Además, se puede elegir factores como la resolución de nuestra pieza o la elección de estructuras auxiliares para una correcta impresión.

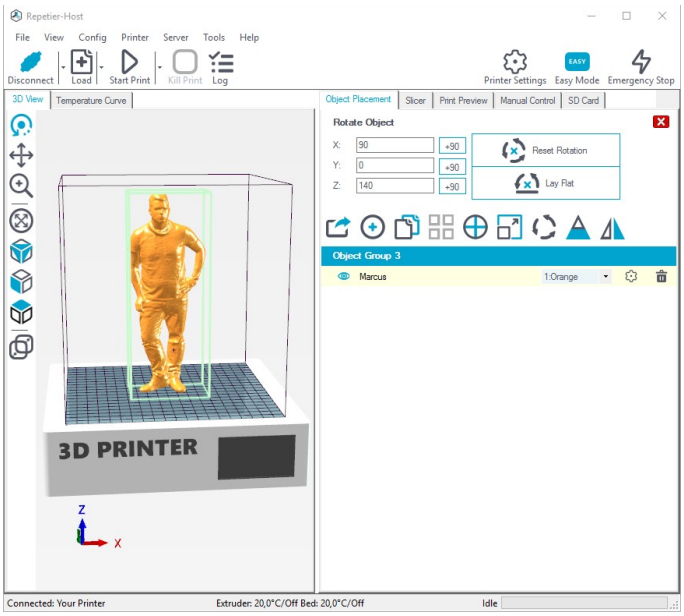


Figura 7. Interfaz de Repetier [Fuente: https://www.repetier.com/documentation/repetier-host/object-placement]

Como se ha mencionado en el punto 5 del apartado anterior “2.2.6-Fases de la impresión”, se han de elegir diferentes parámetros de impresión que variaran la calidad final de la pieza y la cantidad de material consumido. Algunos de dichos parámetros son:

-Altura de capa: Indica la altura que tendrá cada capa de impresión de la pieza. Este parámetro determina en mayor parte la calidad de la pieza ya que cuanta menos altura de capa menos visible serán las capas.

-Densidad de relleno: Mediante la fijación de esta propiedad se regula la densidad interior de la pieza, es decir, el porcentaje de huecos que se desea que tenga la pieza. Por ello, si se fija un valor del 100% se imprimiría una pieza totalmente sólida y si se fijara un valor del 0% se crea una pieza hueca.

-Tipo de soporte: Este parámetro nos permite realizar la impresión de piezas donde no se pueden evitar los voladizos. Además, cabe la posibilidad de elegir la opción de crear soportes solo en las zonas que se necesite o en todos los voladizos existentes.

-Tipo de adhesión a la plataforma: Esta propiedad permite escoger entre diferentes técnicas para mantener a la pieza pegada a la plataforma. Se puede elegir entre: ningún tipo de adherencia, una base cuadriculada sobre la cual se imprimirá la pieza (“raft”) o rodear la primera capa con más material de manera que se cree un contorno amplio para una mejor adherencia (“brim”).

2.3.8-Lion Pro 3D

Lion Pro 3D es la impresora que se va a usar para la realización del proyecto. Se trata de una impresora desarrollada por una empresa española llamada LEON3D.

Este tipo de impresoras admite más de 30 materiales de impresión diferentes entre los que se encuentran ABS, Nylon, bronce o cobre. Incluye la controladora LIONHEART PRO.

Para trabajar de una manera más autónoma, incluye un controlador LCD con ranura para tarjeta MicroSD.

2.4-Equipo electrónico

2.4.1-Placa Arduino

Arduino es un controlador que se basa en software y hardware fáciles de usar de código abierto (open-source). El hecho de ser una plataforma de código abierto le hace destacar frente a otras, ya que los planos de los módulos están publicados bajo licencia. Además, ofrece la facilidad de ejecutarse en sistemas operativos varios, como son Windows, Linux o Macintosh OSX.

Mediante Arduino se puede programar y desarrollar elementos autónomos o, simplemente usarlo como captador de información.

Arduino ofrece una gran cantidad de diferentes modelos de placa, algunos de los más conocidos son:

- Arduino Uno
- Arduino Mega
- Arduino Primo
- Arduino Nano
- Arduino Pro

A continuación se definen las características básicas de las placas Arduino más usadas:

- Arduino Uno

Se trata del primer modelo y más extendido, por ello, la mayoría de placas restantes posee las características de este modelo. En cuanto a memoria es uno de los modelos más limitados, aunque resulta suficiente para muchos tipos de proyectos. Consta de 14 pines digitales, de los cuales 6 pueden usarse como PWM (Modulación por ancho de pulso) y hasta 6 pines analógicos.

- Arduino Mega

Se trata del modelo más potente de Arduino aunque es posible que tenga que sacrificar algo de espacio. De todos los modelos, es la cuenta con más RAM. Respecto a las propiedades eléctricas es muy similar a la placa Arduino Uno aunque posee 54 pines digitales (15 de ellos PWM) y 16 pines analógicos.

- Arduino Nano

Es una placa de pequeñas dimensiones, tan sólo 18.5x43.2 mm. A pesar de su tamaño reducido se trata de un modelo muy completo, con conexión mini-USB pero sin conector de alimentación externa. Fue creada para aplicaciones de coste reducido y donde importe la grandaria de la placa. Sus propiedades eléctricas son como las de la placa Arduino UNO con la diferencia de que el modelo NANO incluye 2 pines analógicos más.

-Arduino Pro

Este modelo de placa fue creada para usuarios que buscan flexibilidad y un precio bajo. No se trata de la placa más potente de Arduino, ya que las características son muy similares a las de Arduino UNO (mismo número de pines). Está destinada a la instalación semipermanente en objetos ya que la placa viene sin encabezados preinstalados y, por ello, se permite el uso de diferente conector o la soldadura de cables.

2.4.2-Servomotores

2.4.2.1-Definición de Servomotor

Los servomotores son motores DC con la propiedad característica de controlar la posición. El servomotor engloba a una serie de componentes electromecánicos y electrónicos:

La parte eléctrica consta de un motor de corriente continua, cuyo eje se encuentra acoplado a una caja de engranajes que aumentan el par del motor y permiten mantener una posición fija. La parte electrónica, formada por el controlador, es la encargada de manejar el movimiento y posición del motor.

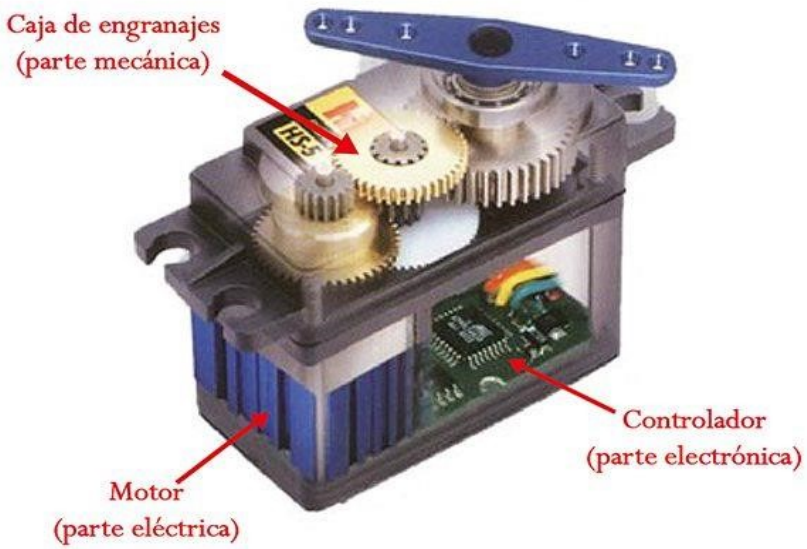


Figura 8. Partes de un servomotor

[Fuente: http://panamahitek.com/que-es-y-como-funciona-un-servomotor/]

2.4.2.2-Tipos de Servomotores

La primera división que se encuentran es entre servomotores de tipo industrial y servomotores de modelismo. En el último tipo ya que son aquellos que se emplean en prototipos de robótica. Los servomotores de modelismo operan a bajos voltajes y en corriente directa (DC).

Dentro de este tipo de servomotores, se clasifican según las características de rotación:

-Servomotores de rango de giro limitado: Son los más empleados. Permiten un giro de 180 grados, es decir, no tienen la capacidad de dar la vuelta completa.

-Servomotores de rotación continua: Son similares a un motor convencional, pero tienen la capacidad de poder ser controlados en su posición y velocidad de giro. A diferencia de los anteriores, permiten realizar giros de 360 grados.

2.4.2.3-Funcionamiento de los Servomotores

Para el control de los servomotores se dispone de 3 cables. Para un mejor reconocimiento de ellos, se emplea siempre el mismo código de colores:

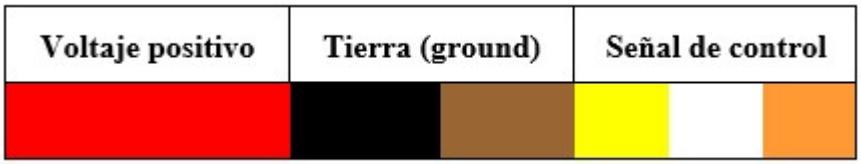


Figura 9. Colores de los cables de un servomotor

[Fuente: http://panamahitek.com/brazo-robotico-con-arduino/]

El primer cable, de color rojo, se emplea para recibir la alimentación eléctrica. El segundo cable, de color negro o marrón, ha de conectarse a la tierra común del circuito. Por último, el cable de control, de color amarillo, blanco o naranja, se conecta a algún pin digital de la placa Arduino, por donde se enviaran los pulsos que controlaran el desplazamiento angular del eje.

Una de las diferencias con los motores DC es que los Servomotores no necesitan invertir la polaridad de la alimentación y, por tanto, no es necesario incluir ningún “puente H”.

Los Servomotores basan su funcionamiento en la modulación PWM (Pulse Width Modulation). En este tipo de señal se emite una señal cuadrada formada por pulsos de frecuencia constante, sobre 500 Hz y consta de dos valores fijos de tensión: uno ALTO, que es la amplitud, y otro BAJO, que es el valor nulo. Consiste en variar la duración de este pulso para variar la tensión de salida. Es decir, cuanto más

corto sean los pulsos, más distantes estarán entre sí en el tiempo y, por tanto, menor será la tensión promedio de salida. Por otra parte, si los pulsos tienen mayor duración, la distancia entre ellos será menor y, con ello, la tensión promedio de salida será mayor.

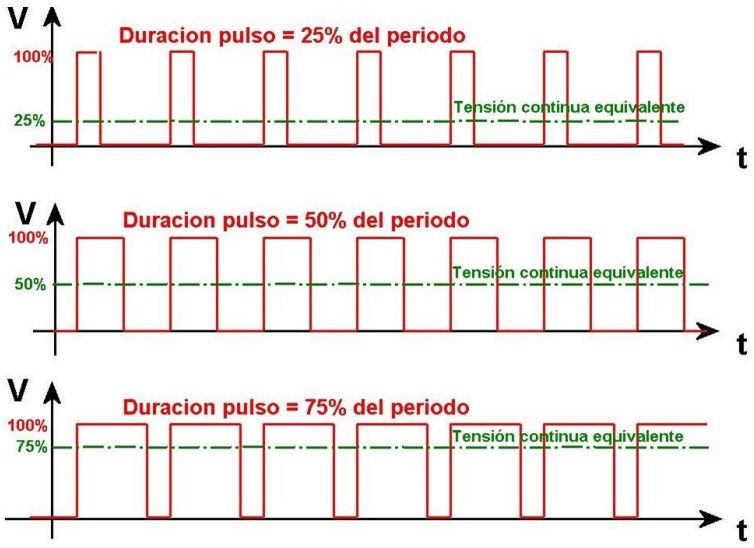


Figura 10. Gráficas funcionamiento PWM

[Fuente: https://hetpro-store.com/TUTORIALES/beaglebone-black-pwm/]

De forma más concreta, si el valor ALTO (5 V) se mantiene durante 1.5 milisegundos el eje del servo se ubica en la posición central de su recorrido, es decir, el eje se habrá situado a 90º respecto al origen. Si la duración del pulso se encuentra entre 1.5 y 2 milisegundos, el eje del servo se situará en una posición angular proporcional entre 90 y 180º. Por último, si la duración del pulso se encontrara entre 1 y 1.5 milisegundos, el eje del servo estaría situado en una posición angular proporcional entre 0 y 90º.

2.4.3-Bluetooth

El Bluetooth es una especificación industrial que define las características de un tipo de redes inalámbricas de corto alcance. Su principal uso es posibilitar la comunicación entre distintos dispositivos consumo que se encuentran relativamente próximos.

La principal ventaja es que permite simplificar la configuración de dispositivos cercanos.

El Bluetooth estándar empleado utiliza un enlace de radiofrecuencia en la banda ISM de los 2.4 GHz para la transmisión de voz y datos. Dichas bandas, cuyo acrónimo es “Industrial, Scientific and Medical”, están reservadas internacionalmente para un uso de forma abierta por todo el mundo sin necesidad de licencia, sunicamente se ha de respetar las regulaciones que limitan los niveles de potencia transmitida.

Por otra parte, cabe comentar que existen dos tipos de módulos bluetooth: Esclavo y maestro. La diferencia entre ellos consta de qué elemento comienza la conexión, es decir, el módulo maestro es aquel que inicia la conexión y se conecta a otro dispositivo y, el módulo esclavo es al que otro dispositivo le solicita la conexión y se conecta.

2.4.4-Regulador de tensión

Un regulador de tensión es un componente electrónico que protege partes de un circuito de un elevado voltaje o de variaciones notables de este. Su función es proporcionar, a partir de un voltaje recibido que varía dentro de un determinado rango (voltaje o tensión de entrada), otro voltaje (voltaje o tensión de salida) regulado a un valor estable y menor.

Para conseguir que disminuya la tensión se emplea la Ley de Ohm, ya que se puede aumentar o disminuir la corriente interna que lo atraviesa en cualquier momento y, de esta manera, se puede elevar o disminuir la tensión de salida.

2.5- Equipo de programación

2.5.1-ARDUINO IDE

Para desarrollar la programación de las placas arduino se emplea un IDE. Un IDE es un conjunto concreto de instrucciones, agrupadas de forma correcta y sin ambigüedades que pretende obtener un resultado determinado. Las siglas IDE provienen de "Integrated Development Environment", es decir, Entorno de Desarrollo Integrado y engloban a las herramientas software que permiten desarrollar diferentes programas. Para el diseño de aplicaciones se emplea la programación en Java.

En el ejemplo de Arduino, se necesita un IDE que permita escribir y editar el programa (llamado "sketch" en Arduino), donde se pueda comprobar que no se haya cometido ningún error y que permita grabar el programa en la memoria del microcontrolador para que éste se convierta en ejecutor autónomo de dicho programa. Para ello, se procede a la descarga del IDE Arduino de la página oficial de Arduino, ya que es de acceso gratuito.

Lo primero que se tiene que hacer cuando se inicia el programa es seleccionar el tipo de placa a usar y el puerto serie que tiene que utilizar el computador para comunicarse vía USB con la placa en la opción de herramientas. En este caso, se tiene que señalar que se emplea una placa Arduino Nano y, en el caso del puerto, aparece la opción de "Arduino Nano (COM3)".

Una vez se realiza este paso, existen varias opciones a elegir en la barra de botones:

-Verify: Este botón se emplea para comprobar que no hay ningún error en el sketch y compilarlo si no encuentra ningún error. Cada vez que se haga una modificación en el código se tendrá que pulsar.

-Upload: Dicho botón se pulsará inmediatamente después que el botón de "Verify". Se encarga de, mediante el comando "arvdude", cargar en la memoria del microcontrolador de la placa Arduino el sketch que ya ha sido verificado. Al activar este botón se activa automáticamente el "auto-reset" del microcontrolador y, de esta manera, "arvdude" es capaz de realizar su tarea.

-New: Se emplea para crear un sketch vacío.

-Open: Muestra todos los sketches disponibles para abrir. Se encuentran tanto sketches nuestros como sketches de ejemplo listos para probar, clasificados por categorías.

- Save: Guarda el código del sketch creado en un fichero. Estos archivos tendrán una extensión ".ino". Para cada proyecto se creará una subcarpeta, donde se guardaran los sketches correspondientes.

-Serial monitor: Abre el "monitor serie". Se trata de una ventana IDE que nos permite enviar y recibir datos textuales a la placa Arduino desde un computador empleando el cabe USB (micro USB en placa Nano)

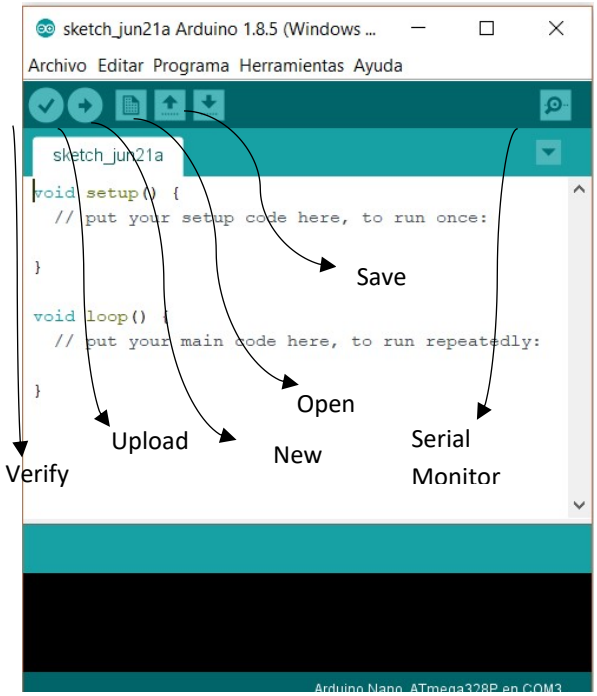


Figura 11. Funciones principales Arduino IDE

2.5.1.1-Estructura general de un Sketch

La estructura de un Sketch en el IDE de Arduino se divide en tres secciones:

-La sección donde se declaran las variables globales, que se encuentra ubicada directamente al principio del sketch y no tiene ningún delimitador de principio o final.

-La sección "void setup ()", que viene delimitada por las llaves de apertura y cierre. Dentro de esta sección se escriben las instrucciones que se ejecutaran una sola vez, cuando se encienda o resetee la placa Arduino. Se emplea para realizar configuraciones iniciales.

La sección "void loop ()" que, al igual que la anterior, viene delimitada por las llaves de apertura y cierre. Las instrucciones se ejecutaran justo después de las escritas en la sección "void setup ()" pero, con la diferencia de que se ejecutaran infinitas veces hasta que la placa se apague o resetee. Se emplea para escribir el programa en sí que está funcionando continuamente.

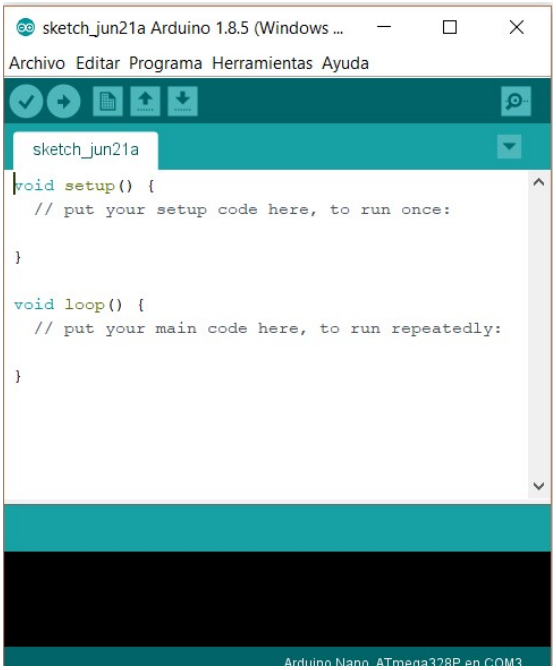


Figura 12. Presentación sketch Arduino IDE

2.5.2-Programacion aplicación en dispositivo móvil

2.5.2.1-Plataforma Android Studio

Android Studio es una plataforma de desarrollo libre y código abierto. Una plataforma de desarrollo libre es aquella que se guía por unos principios básicos de libertad:

-Libertad de ejecutar el programa

-Libertad de poder estudiar el funcionamiento del programa y, por tanto, poder acceder al código fuente.

-Libertad de poder redistribuir copias del programa.

-Libertad de introducir mejoras al programa y, compartir éstas de forma pública para así beneficiar al resto de comunidad.

Por otro lado, cuando se denomina software de código abierto aquellos que son distribuidos con licencia de uso, modificación y redistribución. Para que un software sea considerado de código libre tiene que cumplir ciertas características:

-Se debe poder redistribuir y modificar.

-Se tiene que garantizar la integridad del programa original.

-No se debe discriminar a ningún grupo de personas.

-La licencia se tiene que distribuir con el software, deberá ser la misma y no debe restringir otros programas.

-El software debe tener permiso para cualquier fin.

2.5.2.2Estructura del Sistema Operativo Android

La arquitectura de Android Studio está formada por capas diferentes:

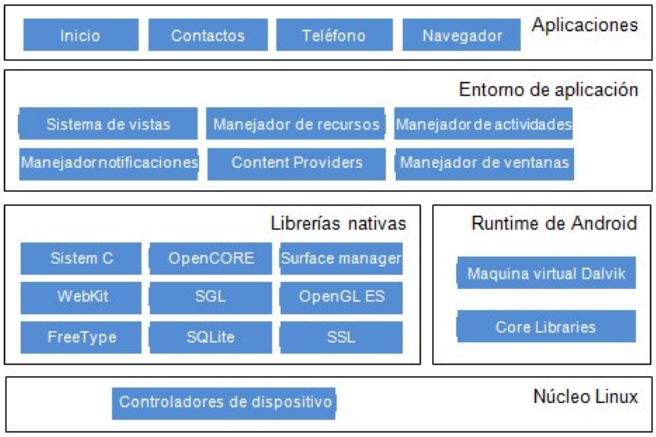


Figura 13. Arquitectura de Android Studio

[Fuente: https://sites.google.com/site/pruebajoseog/arquitectura-de-android/]

La primera capa que se encuentra de manera ascendente es el núcleo de Android, que está basado en el Kernel del sistema operativo del Linux Version 2.6. El núcleo es la única capa que está en contacto directo con el hardware y es una capa de abstracción entre el éste y el resto de software. Su funcionalidad es proporcionar servicios del sistema, tales como pueden ser seguridad, gestión de memoria, soporte a driver u otros.

A continuación se hallan de forma paralela el RunTime de Android (entorno de ejecución) y las librerías nativas.

Por una parte, el entorno de ejecución está formado por dos componentes distinguidos:

-Core libraries: Son aquellas librerias que proporcionan las funciones de Java.

-Máquina virtual Dalvik (DVM): Esta máquina está basada en la máquina virtual de Java aunque, por limitaciones respecto a la memoria y al procesador, no se pudo emplear una máquina virtual de Java estándar.

Además, por otro lado, el ART (Android Runtime) se encarga de compilar el Java bytecode durante la instalación de una aplicación y, por ello, se produce un arranque más rápido de las aplicaciones

Más tarde,

Por último, se encuentra la capa “Aplicaciones”. En él se incluyen tanto las aplicaciones que incluye por defecto Android (calendario, mapas, navegador, contactos…) como las que el usuario va añadiendo.

Para la creación de aplicaciones se emplea la programación en Java (mediante Android SDK) o C/C++ (empleando Android BDK; Native Development Kit).

2.5.2.3-Componentes de una aplicación

Las aplicaciones se componen de 4 elementos que realizan diferentes tareas.

Primeramente, las “Actividades (Activity)” nos expresan la cantidad de pantallas de las que consta una aplicación. Este elemento se divide en dos partes que almacenan distintos datos:

-Parte lógica: Se trata de un archivo .java y es la que se crea para definir el código de la actividad.

-Parte gráfica: Está formada por los archivos XML y éstos se encargan de contener los controles que se observa en la pantalla (Botones, textos…).

El segundo componente de una aplicación es el “Layout” y se definen como contenedores de elementos. Éstos son elementos no visuales que se emplean para distribuir y organizar las distintas que se inserten en su interior.

En tercer lugar se encuentran los “widgets”, “controles” o “vistas”. Estos elementos son los que componen la interfaz de usuario y se observan múltiples tipos de control. Aunque pueden ser definidas mediante código Java, es más usual definir los controles a través de un fichero XML ya que el sistema nos crea los objetos a partir del fichero insertado.

El último elemento del que se compone una aplicación es el “Intent” o “Intención”. Dicho elemento es el encargado de realizar la comunicación entre las diferentes actividades y/o elementos de la aplicación. Se emplea por tanto para lanzar una actividad o para realizar el pase de parámetros (comunicarse con una actividad).

Capítulo 3-Diseño brazo robot

3.1-Especificaciones

En el diseño de nuestro brazo robot se ha decidido hacer una estructura de 5 grados de libertad, ya que son suficientes para el trabajo que se desea realizar(descenso del brazo- apertura de la pinza – agarre de un objeto – cierre de la pinza- giro del brazo-ascensión del brazo).

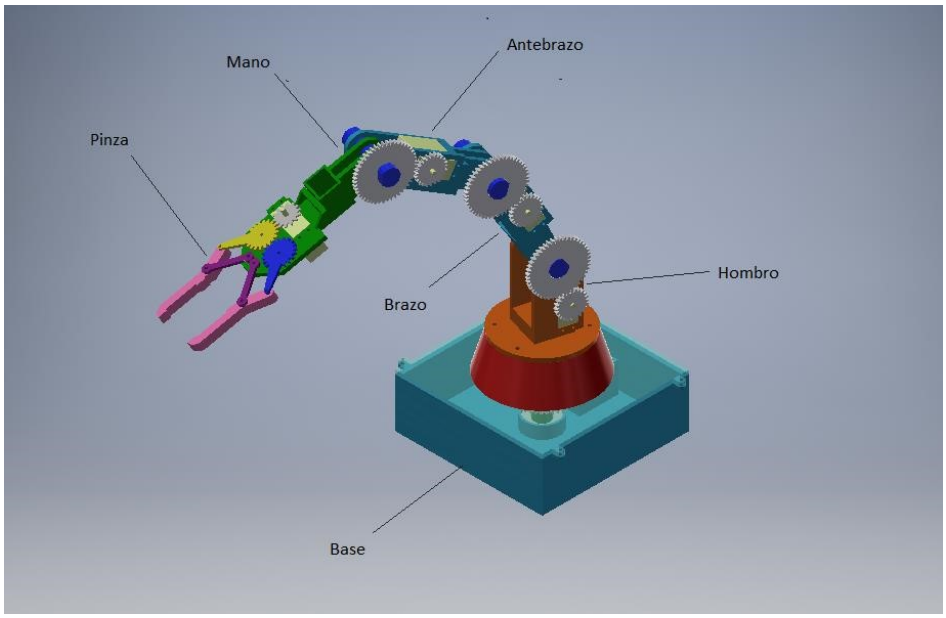


Figura 14. Articulaciones brazo robot

Los movimientos que se encuentran son:

1- Giro rotatorio de la base
2- Giro del brazo respecto al hombro
3- Giro del antebrazo respecto al brazo
4- Giro de la mano respecto al antebrazo
5- Apertura y cierre de la pinza

Para valorar el área de trabajo del robot que se ha creado, se hace un estudio de la longitud máxima a la que podrá llegar la pinza para agarrar un objeto. Para ello, se proyecta el perfil del brazo en su posición de máxima distancia y se mide la longitud.

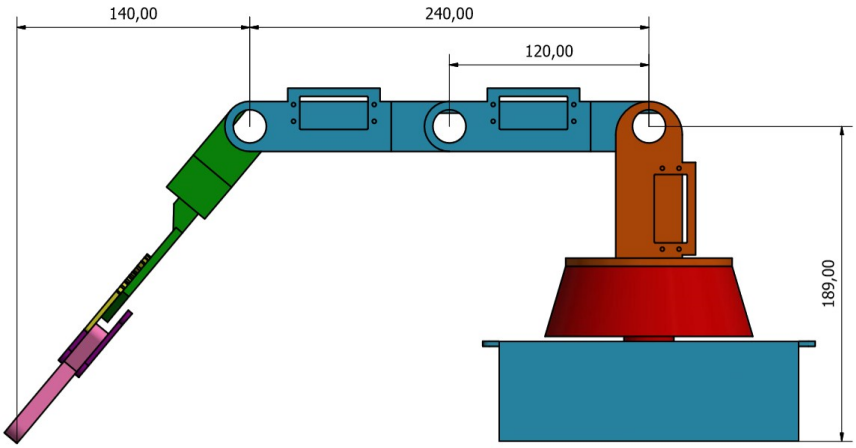


Figura 15. Cotas del alcance de piezas del brazo robot

Como se puede observar en la imagen anterior, la longitud máxima constará de unos 380 mm aproximadamente y, como máximo podrá llegar a piezas que estén situadas

Por ello, el volumen de trabajo en el que podrá agarrar objetos será, basándonos en la fórmula del tronco de cono:

$$V\ trabajo = \frac{1}{3}.\pi.h.(R^2 + r^2 + R.r) - \pi\ .r_{min}^2\ .h \quad (1)$$

Por tanto, conforme se ha planteado anteriormente, se sustituyen los datos en la fórmula:

$$V\ trabajo = \frac{1}{3}.\pi.189.(380^2 + 240^2 + 380.240) - \pi\ .120^2\ .189 = 49480084\ mm^3$$

Además, se debe valorar el área de movimiento que podrá alcanzar el robot. Para ello, se cuenta con la altura máxima que a la que podrá llegar el brazo, que será de aproximadamente 475 mm.

Como el brazo robot podrá desplazarse 180º el volumen de desplazamiento será igual a un cuarto de una esfera. Es decir:

$$V\ trabajo = \frac{4}{3}.\pi.r^3 \quad (2)$$

$$V\ trabajo = \frac{4}{3}.\pi.(475)^3 = 448920500{,}2\ mm^3$$

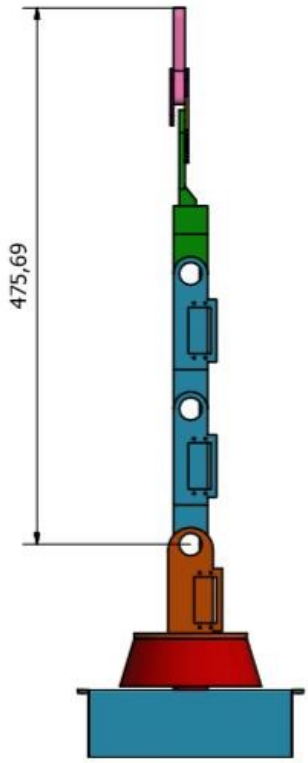


Figura 16. Alcance máximo del brazo robot

Por otra parte la carga nominal que podrá soportar el brazo sin perder sus propiedades físicas es de 40 gramos. Se realiza el estudio de la misma en las condiciones más desfavorables en el apartado “3.7.2- Simulación fuerza desconocida”.

3.2-Desarrollo electrónico

3.2.1-Electrónica escogida

3.2.1.1-Servomotor MG996R

Para conseguir realizar todas las funciones que se necesitan sin problemas, se ha elegido el Servo MG996R, ya que nos proporciona un torque detenido de 9.4 kgf.cm con 4.8 V y hasta 11 kgf.cm de torque con un voltaje de 6 V. Para las aplicaciones para las que está destinado el brazo robot el torque que proporciona el servo seleccionado es más que suficiente.

Este servo, además, opera con un voltaje entre 4.8 V y 7.2 V, que puede ser proporcionado mediante una batería.

Respecto a otros servomotores, el MG996R es un poco más pesado (55g). Esta propiedad puede ser crítica ya que aumenta el peso del brazo robot pero, a pesar de ello, el torque que proporciona puede suplir este aumento de peso.

Por último, es de interés la propiedad de velocidad de operación, que será de 0.17 s/60º con 4.8 V y hasta de 0.14 s/60º con 6 V.

Figura 17. Servomotor Mg996r

[Fuente: http://www.hobbyandyou.com/towerpro-mg996r-metal-gear-servo-motor*]*

3.2.1.2-Placa Arduino Nano

La placa elegida es la Arduino Nano ya que su tamaño reducido hace que se pueda introducir la placa dentro de la caja de la base sin tener que hacer una estructura excesivamente grande.

A pesar de algunas desventajas respecto a otras placas, como un menor número de puertos de entrada/salida o un menor espacio en la memoria, las características que tiene son suficientes para la aplicación que se va a realizar.

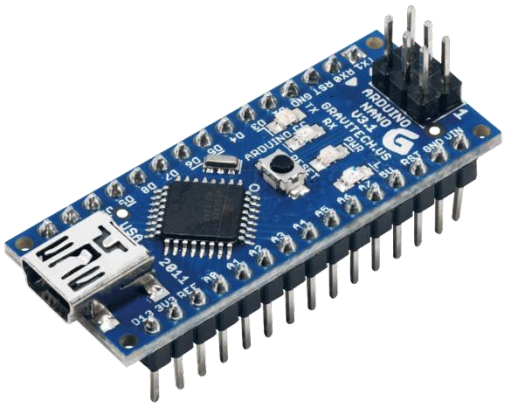


Figura 18. Placa Arduino Nano

[Fuente: http://www.nova.com.bo/arduino-nano.html*]*

3.2.1.3-Regulador de tensión

La placa Arduino Nano incluye un regulador de tensión ya integrado pero, sin embargo, este regulador no es suficiente y, al incluir otros módulos (como el módulo Bluetooth) puede sobrecalentarse el circuito y dejar de funcionar. Por ello, se ha decidido incluir un regulador externo, ya que así se puede asegurar que los componentes resulten dañados.

Se ha escogido un regulador de tensión 7805. Se trata de un regulador de tensión positiva (regulador 78XX) que tiene una tensión de salida de 5V (regulador XX05) y soporta hasta 1 A de intensidad. Además se trata de un regulador bastante económico y que, contando con sus limitaciones (tensión de entrada menor que 35 V), se adapta a las condiciones que se aplicaran en el proyecto.

Indiferentemente del modelo de regulador escogido, los reguladores están formados por tres patillas diferentes: una de entrada (input), otra de salida (output) y una última de tierra (GND).

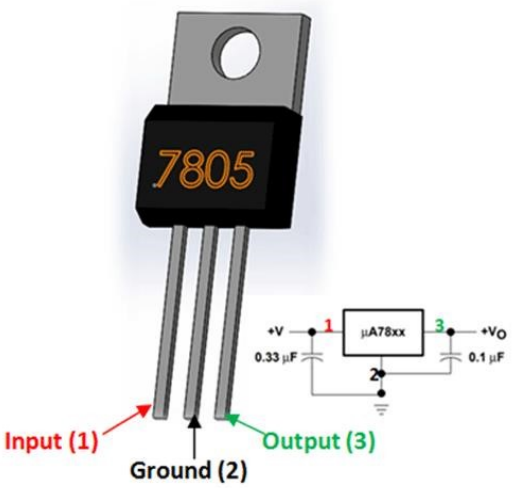


Figura 19. Regulador de tensión 7805

[Fuente: http://supertech.rw/sample-page/voltage-regulatorsth/78l05-positive-5v-regulatorsmd-2/]

3.2.1.4-Módulo Bluetooth

Para nuestra aplicación, se va a emplear un módulo bluetooth para poder manejar el brazo robot desde una aplicación en un dispositivo móvil. De esta manera, se pueden evitar las molestias que causan los cables y, además, nos permite manejar el brazo a una cierta distancia.

Se ha escogido el módulo bluetooth HC-06 ya que se trata de un módulo esclavo y, por ello, será el dispositivo móvil el que inicie la conexión para conectarse a la placa Arduino Nano.

Este módulo consta de 4 patillas que vienen señalizadas:

-VCC: Es la patilla que va conectada al voltaje positivo de alimentación, es decir, al pin de tensión

de la placa Arduino.

-RX: Se define como la patilla de recepción de datos. Recoge los datos de Arduino y los envía al dispositivo móvil

-TX: Se trata de la patilla de transmisión de datos. Envía los datos recibidos desde la aplicación

Móvil a la placa Arduino.

-GND: Es la conexión de toma tierra que garantiza la protección del módulo.

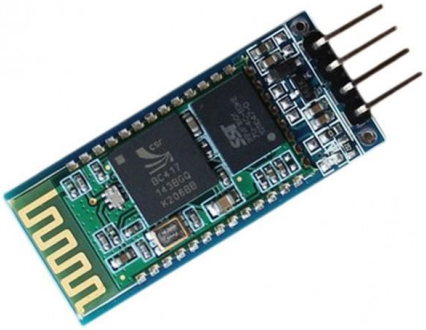


Figura 20. Módulo bluetooth HC-06

[Fuente: https://www.iberobotics.com/producto/modulo-bluetooth-hc06/*]*

3.2.1.5-Placa de prototipado

Para poder realizar la conexión se ha elegido emplear una "protoboard". Las "protoboards" son placas perforadas con conexiones internas donde se 31pueden insertar las patas de los componentes electrónicos tantas veces como se quiera, evitando así la necesidad de soldar. El objetivo del empleo de este tipo de placas es poder montar prototipos totalmente funcionales de manera más rápida.

Para poder conectar correctamente los componentes electrónicos se tiene que conocer cómo se estructuran las propias conexiones internas de la placa seleccionada.

3.2.2-Circuito electrónico

El circuito electrónico del conexionado es de la siguiente manera.

La batería externa se conecta al INPUT de los reguladores de tensión y al pin VIN de la placa Arduino.

El OUPUT de uno de los reguladores de tensión se conecta a todas las conexiones VCC de los servomotores y el otro al VCC del módulo Bluetooth.

Además todas las tomas a tierra de los elementos se conectan al mismo pin GND de la placa Arduino Nano.

Por otra parte, las patillas TX y RX del módulo HC-06 se tienen que conectar a los pines RX y TX de la placa respectivamente. Estos pines son los que permiten la lectura y envío de datos entre el modulo y la placa Arduino.

Por último, los cables de color amarillo de los servomotores se conectan cada uno a un pin digital de la placa. Este pin es el que más tarde se asociará a cada servomotor para indicar que está conectado.

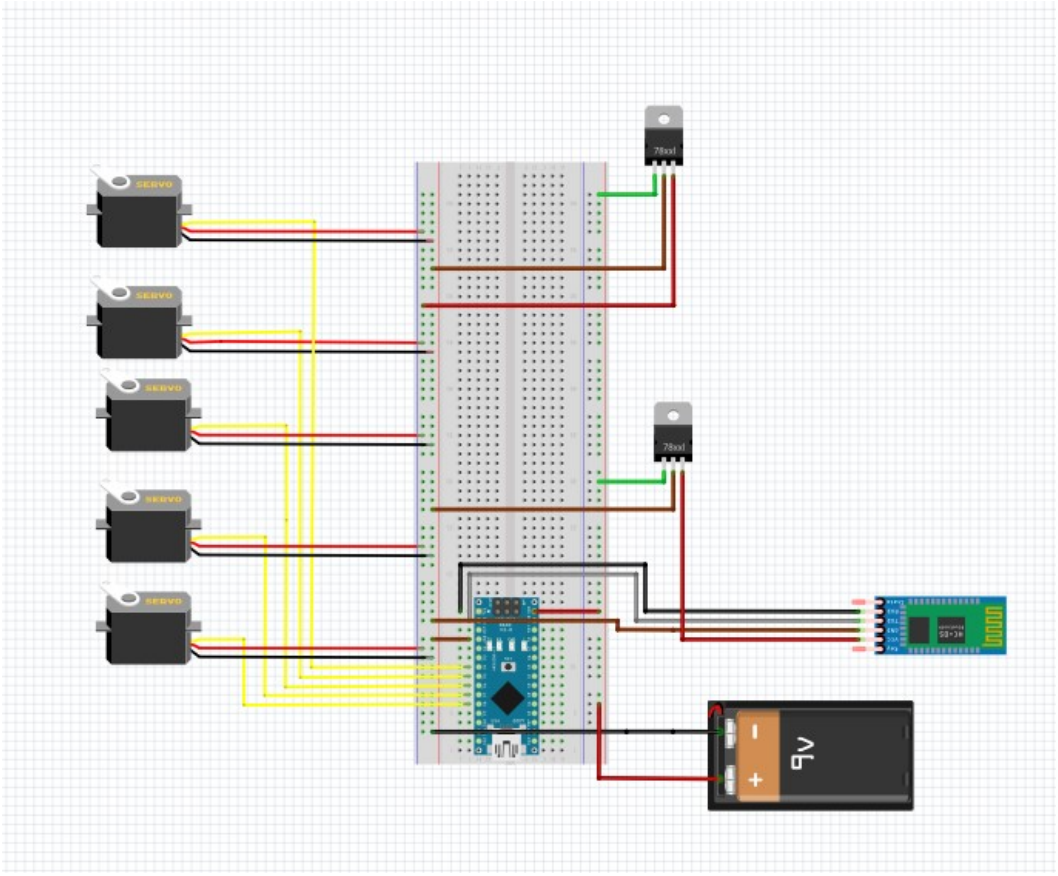


Figura 21. Esquema conexionado del circuito electrónico

3.2- Diseño del brazo robot

3.2.1-Subconjunto Base brazo robot

Este subconjunto se encarga del movimiento de rotación de la base. Para ello, mediante la activación de un servo que se encuentra ubicado en la caja de la base se mueve un par de engranajes, uno alojado en el eje del servo y otro insertado en el eje de la base. Este último eje está introducido en el interior de la base y, dicha pieza, atornillada al brazo inferior. De esta manera, se desplazan las piezas de forma solidaria y se consigue el movimiento buscado.

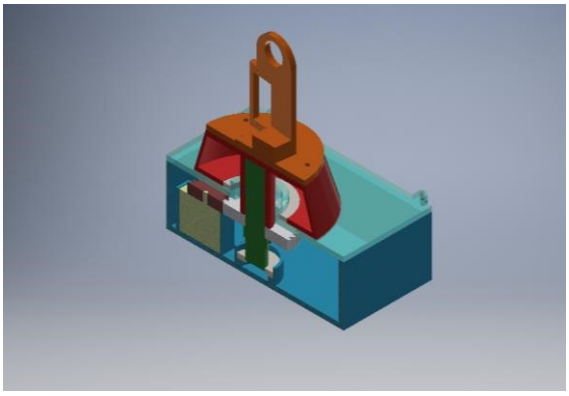


Figura 22. Vista partida del subconjunto base

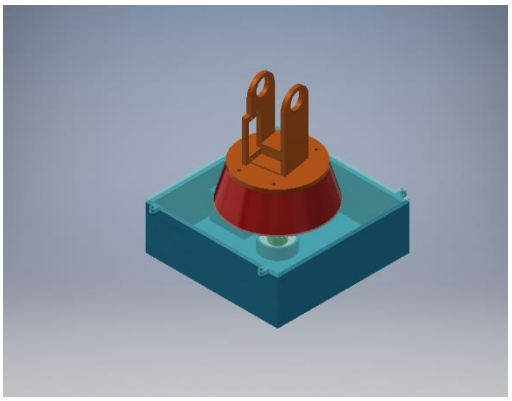


Figura 23 Vista completa del subconjunto base

3.2.1-Caja inferior

La caja inferior tiene que tener una estructura capaz de almacenar el servomotor que se empleará para el giro rotatorio de la base y, además, tendrá que alojar el rodamiento pequeño que se ha empleado para el movimiento ya mencionado.

Para una mejor disposición del rodamiento, se ha decidido hacer una estructura de manera que éste no roce la parte inferior de la base y, evitar así, pérdidas por rozamiento. Por ello, la estructura está elaborada para sujetar el aro exterior del rodamiento pequeño y así poder dejar un giro libre para el aro interior del mismo.

En cuanto a los rodamientos, se ha escogido rodamientos rígidos de bolas de contacto radial. Se han seleccionado este tipo de rodamientos ya que no son desmontables, requieren poco mantenimiento y soportan todo tipo de velocidades.

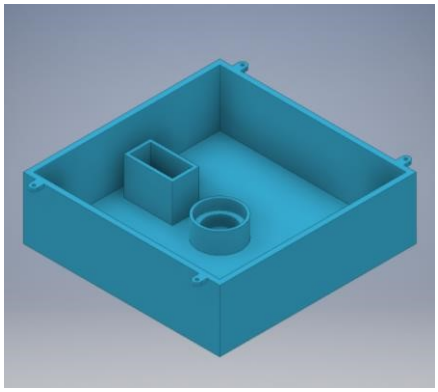


Figura 24. Modelado caja inferior

3.2.2-Tapa base

Para cerrar la caja de la base se ha decidido realizar una tapa que ira atornillada a la pieza mencionada con anterioridad.

Por otro lado, para mejorar el movimiento rotatorio de la base, se ha realizado una protuberancia de forma cilíndrica. Esta estructura adicional va insertada en el aro interior del rodamiento grande para, de esta manera, solidarizarse con el movimiento giratorio y conseguir que el rodamiento pequeño no cargue con todo el movimiento generado.

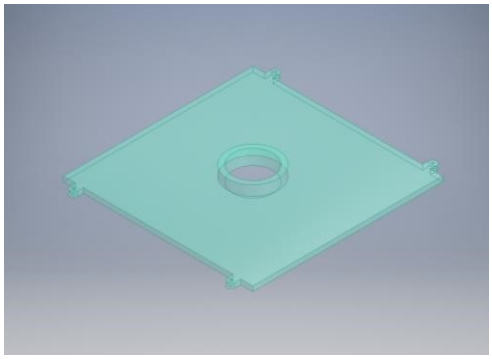


Figura 25. Modelado tapa base

3.2.3-Base

La pieza designada con el nombre de “Base” tiene que ser apta para alojar al eje estriado que comunica el movimiento giratorio al subconjunto base. Asimismo, su estructura está diseñada para ser insertada en el aro exterior del rodamiento grande y conseguir un movimiento del subconjunto más ligero y con mucho menor rozamiento.

Por otro lado, consta de 4 agujeros en la parte superior para poder atornillar el brazo inferior a dicha pieza.

Se observa que se trata de una estructura más rígida que las anteriores y esto se debe a la necesidad de estabilidad en la base para evitar el vuelco del brazo cuando este en las condiciones de menor estabilidad.

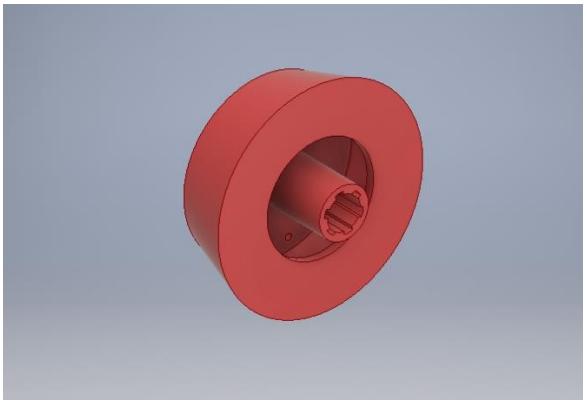


Figura 26. Modelado base

3.2.4-Brazo inferior

Esta pieza, la última que compondría el subconjunto "Base brazo robot" está diseñada con la capacidad de almacenar el servo mg996r, pudiendo éste ser atornillado directamente a la pieza.

Adicionalmente, se ha elaborado una estructura cilíndrica en la base de la pieza con las mismas medidas que la parte superior de la pieza "Base" para, de esta manera, conseguir un mejor acople entre piezas, uniéndolas mediante tornillos y tuercas para una mejor fijación.

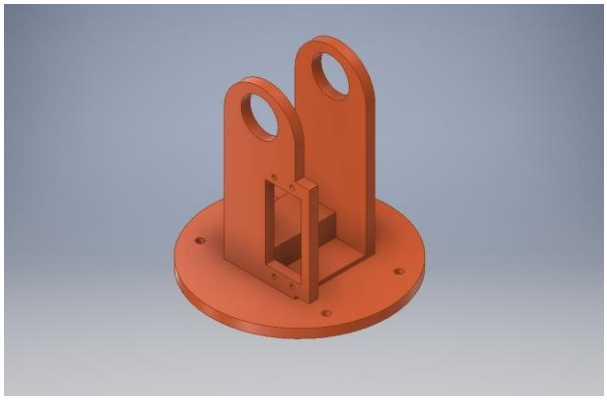


Figura 27. Modelado brazo inferior

3.2.5- Eje estriado

El eje estriado se ha creado con el objetivo de conseguir realizar un buen acople con el engranaje y con la base. Con él se consigue una mayor rigidez al acoplamiento y poder transmitir el esfuerzo del giro proporcionado por el servomotor.

Además, para conseguir un movimiento más deslizante, se ha realizado una extrusión cilíndrica en la parte inferior que ira acoplada en el aro interno del rodamiento pequeño.

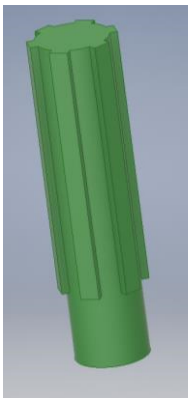


Figura 28. Modelado eje estriado

3.3-Subconjunto brazo y antebrazo

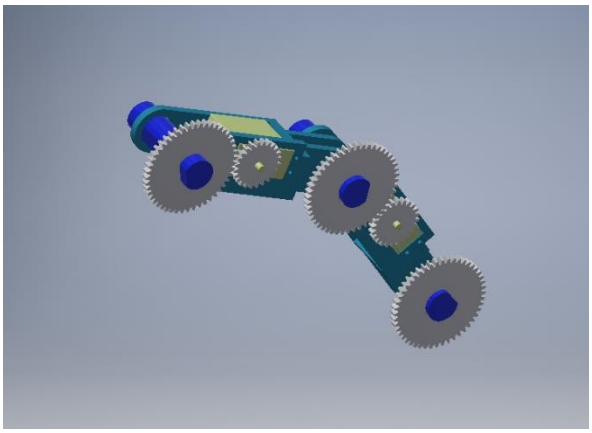


Figura 29. Vista modelado del subconjunto brazo y antebrazo

Dicho subconjunto se encarga del movimiento del hombro, del codo y de la muñeca. Para poder conseguir estos movimientos se posiciona un engranaje en el eje del servo que engrana con otro más que se encuentra introducido en el eje de los brazos. Mediante este ensamblaje se produce la rotación del eje y, con ello, la rotación del brazo que se encuentra ensamblado al eje (aquel que tiene el agujero aplanado).

Por otro lado el brazo que no se encuentra ensamblado al eje, permitirá el deslizamiento de él, generándose unas perdidas por fricción.

3.3.1-Brazo

Esta estructura ha sido diseñada de manera que tenga una forma sencilla y sea capaz de alojar tanto al servomotor que producirá el movimiento del brazo respecto al hombro o el movimiento del antebrazo respecto al brazo.

Para conseguir que el brazo se mueva de forma solidaria al eje se ha hecho una extrusión en los agujeros cilíndricos para así fijar el eje al brazo que tenga que moverse.

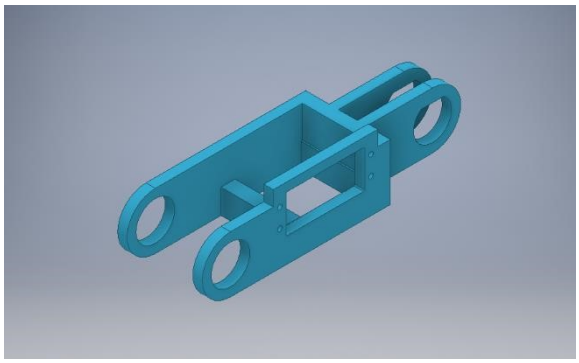


Figura 30. Modelado brazo

3.3.2-Eje brazo

Esta pieza tiene una estructura sencilla para imprimir y montar con facilidad. Para encajar con el brazo y poder hacer que se produzcan los movimientos correctos se le ha realizado una extrusión negativa, aplanando así parte del eje.

Se emplea para poder realizar el movimiento del brazo respecto al hombro, el del antebrazo respecto al brazo y el de la pinza respecto al antebrazo.

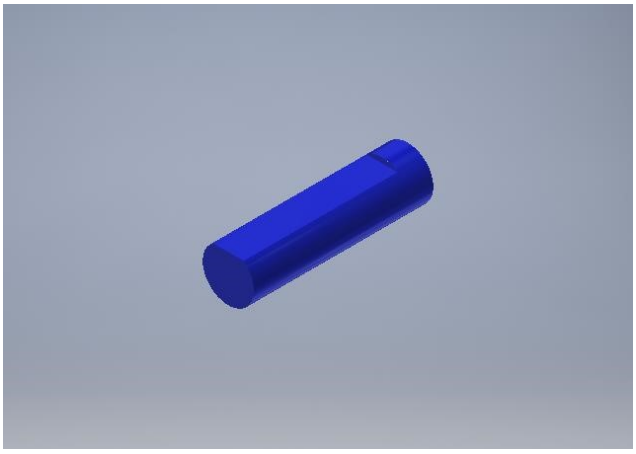


Figura 31. Modelado eje brazo

3.4-Subconjunto pinza

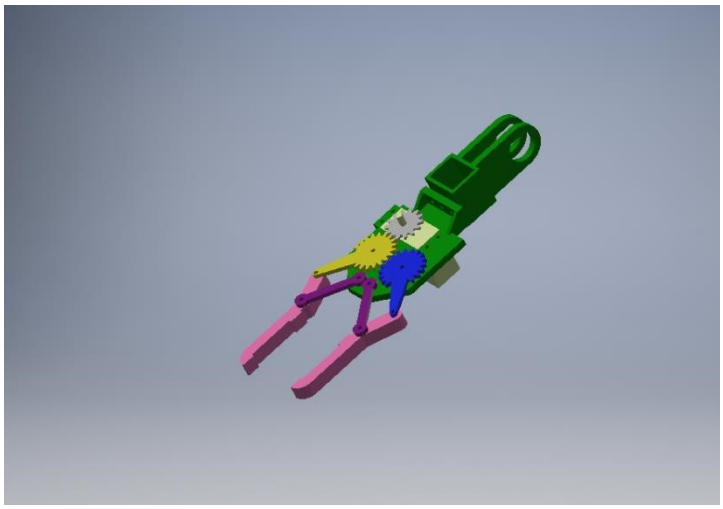


Figura 32. Ensamblaje subconjunto pinza

El subconjunto que se comenta a continuación es el encargado de la apertura y cierre de la pinza. Para conseguir este movimiento se dispone de un engranaje en el eje del servomotor que engrana con otro par de engranajes que van sujetos a la pinza en sí. Mediante la unión de estos engranajes con la pinza y de las piezas auxiliares con la citada pinza, se consigue el movimiento deseado.

3.4.1-Enganche de la pinza

Esta pieza se emplea para poder realizar la unión del subconjunto pinza con el resto del brazo robot. Por ello, consta de los agujeros pertinentes para el correcto alojamiento del eje de los brazos y de tres agujeros donde se introducirán tornillos para una mejor fijación con la pieza “Base pinza”.

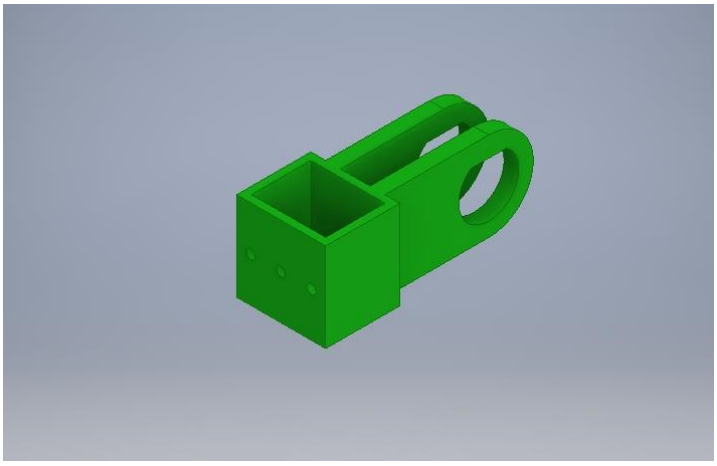


Figura 33. Modelado del enganche de la pinza

3.4.2-Base pinza

Esta pieza está diseñada para que tenga la capacidad de sujetar mediante tornillos a los mecanismos que realizaran el movimiento de la pinza. Es decir, tiene que tener una estructura capaz de soportar los engranajes de la pinza y las piezas auxiliares.

Asimismo, esta pieza tendrá que contener al servo que generara el movimiento de apertura y cierre de la pinza y, de igual manera que la anterior pieza, consta de 3 agujeros para la fijación mediante tornillos y tuercas.

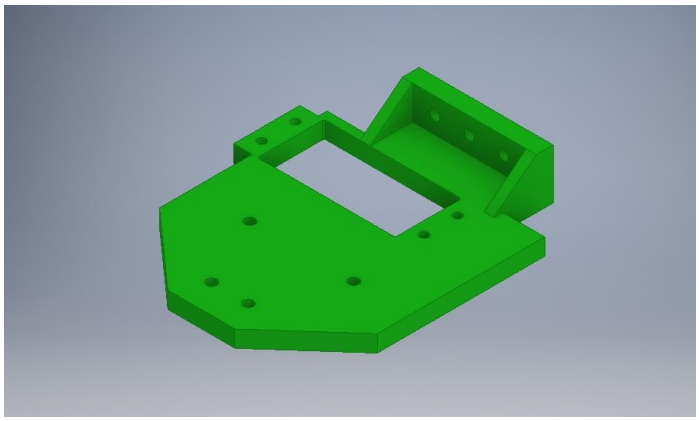


Figura 34. Modelado base pinza

En un primer momento, la pieza comentada y la descrita anteriormente se idearon como una sola pieza. No obstante, para un mejor acabado y precisión en su impresión, se decidió dividirla en dos estructuras que acoplan entre sí mediante tornillos.

3.4.3-Pieza auxiliar

La estructura de la pieza auxiliar es sencilla para conseguir una fácil impresión y se tiene que diseñar con el tamaño adecuado para cuadrar la unión entre los engranajes y la pieza que se emplea como pinza.

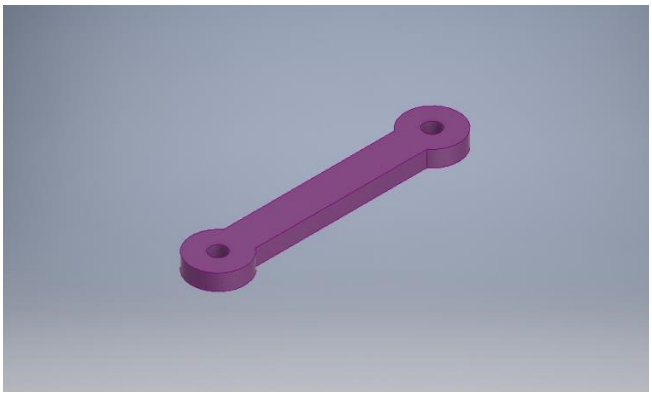


Figura 35. Modelado pieza auxiliar

3.4.4-Pinza

Esta pieza esta creada con una estructura de fácil impresión. Dispone de un tope de la misma altura que el final de la pinza para un mejor cierre de ésta.

Además, la distancia entre sus agujeros tiene que tener una distancia concreta para que pueda sujetarse correctamente con el resto de piezas que forman el subconjunto pinza.

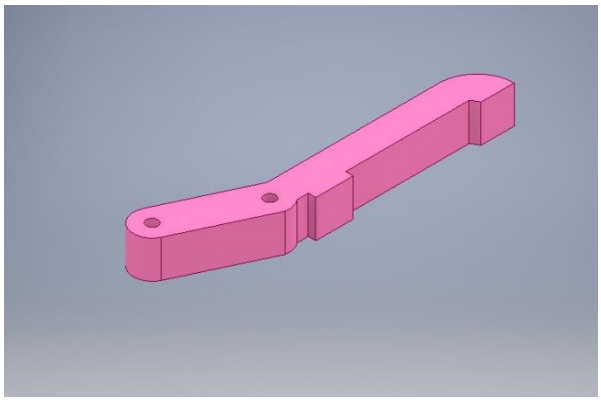


Figura 36. Modelado pinza

3.5-Modelado de los engranajes

3.5.1-Engranajes base

Para el diseño de los engranajes que se encargaran del movimiento giratorio de la base se midió la distancia desde el centro del eje estriado al centro del eje de giro del servomotor Mg996r, basándonos en la posición donde se decidió diseñar el espacio que alojaría al servomotor en la caja de la base. Para un buen funcionamiento del conjunto y del movimiento se decidió un coeficiente de engranaje de 2, ya que, como se explica a continuación se puede reducir suficientemente la velocidad sin llegar a generar un engranaje demasiado grande. Según las leyes de engrane y siendo n1 y n2 revoluciones por minuto de los engranajes 1 y 2 respectivamente y Z1 y Z2 los dientes de los engranajes 1 y 2 respectivamente:

$$i = \frac{n1}{n2} = \frac{Z2}{Z1} \tag{3}$$

De esta manera, al establecer el coeficiente de engranaje (i) en 2, se obtiene que el engranaje 2 tendrá el doble número de dientes y funcionara a la mitad de revoluciones por minuto que el engranaje 1.

Una vez seleccionadas las condiciones de diseño, se solicita al programa que calcule el resto de parámetros y se obtienen los engranajes deseados.

Por otra parte, se realizaron los orificios pertinentes para que los engranajes encajaran tanto en el eje estriado como en el eje del servomotor.

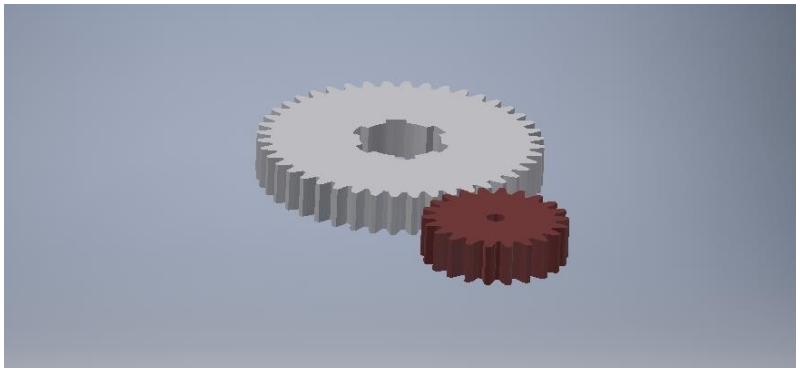


Figura 37. Diseño en Inventor de los engranajes de la base

3.5.2-Engranajes brazos

Para el movimiento de los brazos (hombro, codo y muñeca), se procede de la misma manera, midiendo así la distancia entre los ejes donde se alojarían los engranajes según las condiciones de diseño establecidas. Al igual que en el conjunto anterior, se fija un coeficiente de engranaje de 2. Por tanto, se tienen las mismas condiciones de movimiento anteriores.

Además, en este caso, el engranaje de mayor tamaño ha sido diseñado para poder ser encajado en el eje de los brazos.

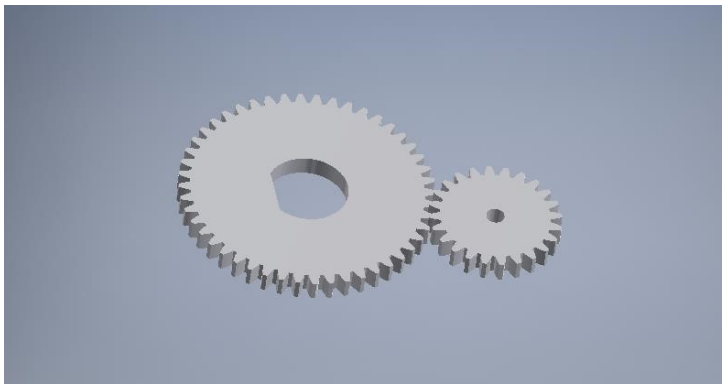


Figura 38. Diseño en Inventor de los engranajes de los brazos

3.5.3-Engranajes pinzas

Para la apertura y cierre de la pinza se necesitan tres engranajes: dos de ellos para cada lado de la pinza y otro más que se introduce en el eje del servo.

Se eligió un coeficiente de engranaje de 1 entre los engranajes de cada pinza, ya que así el movimiento de cada parte será simétrico. Para el movimiento de uno de los engranajes de la pinza y el engranaje del servo se fijó un coeficiente de 2.

Recurriendo a las ecuaciones sobre engrane anteriormente planteadas, los engranajes de las pinzas tendrán los mismos dientes y se moverán a las mismas revoluciones por minuto.

Finalmente, para adecuarse al diseño, se realizan una extrusión en los engranajes de la pinza y agujeros para poder crear la estructura del robot correctamente y atornillar unas piezas con otras.

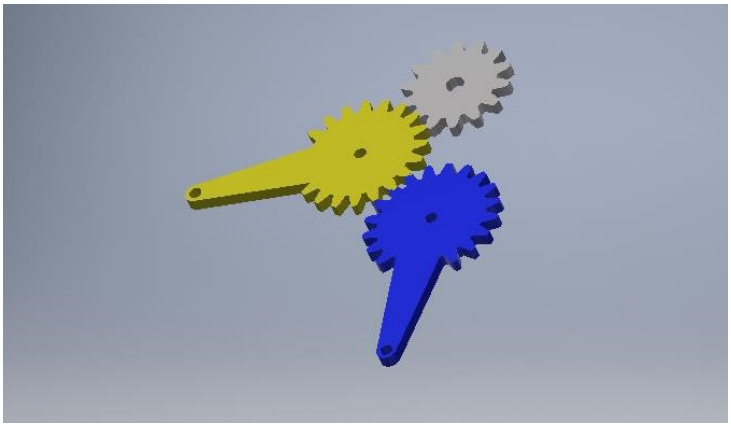


Figura 39. Diseño en Inventor de los engranajes de la pinza

3.6-Ensamblaje

Para realizar un correcto ensamblaje de las piezas del brazo robot se han empleado fundamentalmente dos tipos de restricciones:

-Restricción de coincidencia eje con eje: Éste tipo de restricción se ha empleado para todas aquellas piezas que van ensambladas de manera concéntrica. Es decir, para uniones de tipo eje-cubo.

-Restricción de coincidencia plano con plano: Ésta se utiliza para ensamblar piezas cara con cara o bien para alinear caras de diferentes piezas. Además se puede usar para generar el movimiento de rotación de dos elementos que comparten eje, restringiendo un plano en el que compartan eje respecto al otro.

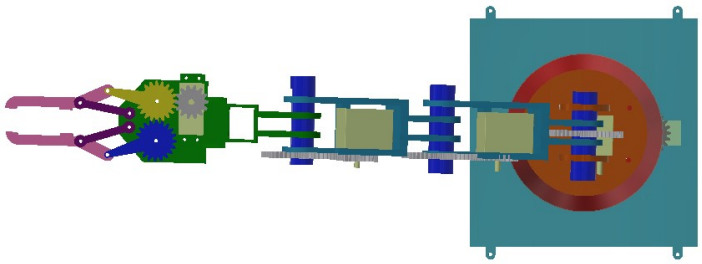


Figura 40. Vista superior del ensamblaje

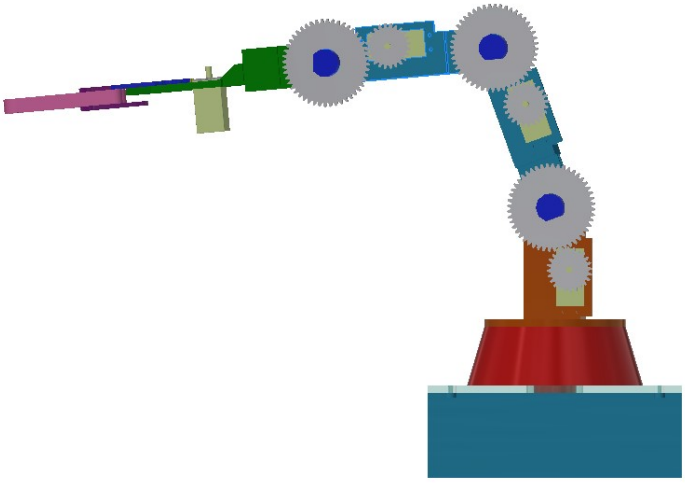


Figura 41. Vista de perfil del ensamblaje

3.7-Simulación dinámica

Autodesk Inventor ofrece la posibilidad de realizar una simulación del movimiento que tendría nuestro ensamblaje mediante las restricciones que le se han impuesto. Las uniones son el producto de las restricciones que se aplican entre dos piezas. Es decir, las restricciones crean un tipo de movimiento entre dos elementos (unión) mediante el bloqueo de diferentes combinaciones de grados de libertad.

Para ello, Inventor proporciona la posibilidad de transformar automáticamente las restricciones creadas en uniones estándar y nos indica entre que piezas o grupo de piezas se produce la unión y de qué tipo es. De esta manera, se pueden encontrar las siguientes uniones estándar:

Tabla 1-Tipos de uniones en Inventor [Fuente: https://knowledge.autodesk.com]

Tipo de unión estándar	**De traslación** **Grado de libertad**	**De rotación** **Grado de libertad**	**Sistema de coordenadas** **Restricciones**
Revolución	Ninguna	Eje Z	$Z_2 = Z_1$ $O_2 = O_1$
Prismática	Eje Z	Ninguna	$X_2 Y_2 Z_2 = X_1 Y_1 Z_1$ O_2 en $O_1 Z_1$
Cilíndrica	Eje Z	Eje Z	$Z_2 = Z_1$ O_2 en $O_1 Z_1$
Esférica	Ninguna	Eje X Eje Y Eje Z	$O_2 = O_1$
Plana	Eje X Eje Z	Eje Y	$Y_2 = Y_1$ O_2 en $O_1 X_1 Z_1$
Punto-línea	Eje Z	Eje X Eje Y Eje Z	O_2 en $O_1 Z_1$
Línea-plano	Eje X Eje Z	Eje Y Eje Z	$O_2 Z_2$ en $O_1 X_1 Z_1$
Punto-plano	Eje X Eje Z	Eje X Eje Y Eje Z	O_2 en $O_1 X_1 Z_1$
Espacial	Eje X Eje Y Eje Z	Eje X Eje Y Eje Z	Ninguna
Soldadura	Ninguna	Ninguna	$O_1 X_1 Y_Y Z_1 =$ $0_2 X_2 Y_2 Z_2$

Como se ha observado en el apartado “2.1.2-Grados de libertad” las uniones que se pueden crear en el entorno de simulación de Inventor son muy similares a los tipos de articulación de un brazo robot ya que es una aproximación a un entorno real de aquello que se ha diseñado.

En este entorno se pueden definir diferentes fuerzas y momentos externos y visualizar la mediante la simulación la respuesta del ensamblaje a dichas acciones. Además, se podría realizar la simulación del

movimiento basándose en diferentes variables, como son la posición, la velocidad, la aceleración y el par de torsión.

Por otra parte, se puede emplear la simulación para calcular la fuerza que sería necesaria para mantener el conjunto en una posición de equilibrio, contando con acciones externas que aplican fuerza o crean momentos en el conjunto, por ejemplo, la gravedad, la cual se podría definir en el eje y dirección que se decida.

3.7.1-Fuerza de rozamiento

Mediante todas estas posibilidades, se ha definido la fuerza de rozamiento que aparece en cada uno de los ejes donde hay una articulación. Para ello, se consulta el coeficiente de fricción del material PLA y se trata de un valor de 0,048. La fricción entre dos componentes a lo largo del tiempo produce un desgaste del material haciendo que surja holgura entre ellos. El problema de este fenómeno es que los componentes pueden dejar de funcionar correctamente y tengan que ser sustituidos ya que se pierde precisión en el movimiento.

Se define en las propiedades de la restricción que sea de nuestro interés el coeficiente de fricción y el radio donde se encuentra rozamiento, ya que se trata de una revolución.

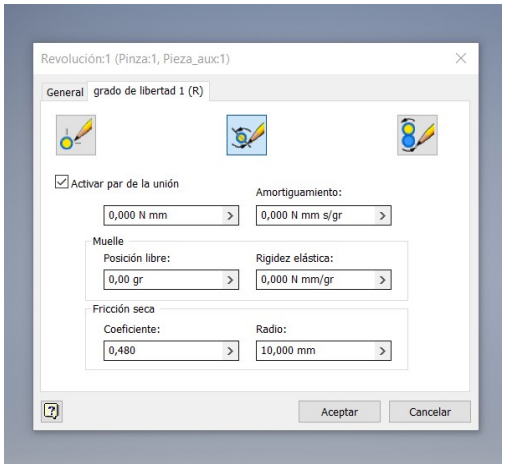


Figura 42. Ventana de propiedades de una restricción

3.7.2-Simulación fuerza desconocida

Inventor ofrece la posibilidad de saber que par será necesario transmitir a una articulación para mover el conjunto desde una posición inicial hasta un determinado punto. Para ello, se estudia el caso más desfavorable de movimiento, donde aparecerá un mayor momento. En este caso será cuando el brazo se encuentre estirado y tenga que realizar la acción de subir 90 grados para posicionarse verticalmente.

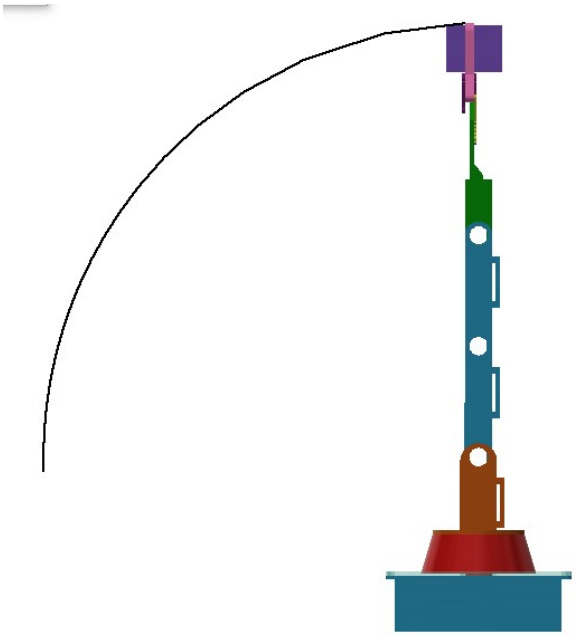


Figura 43. Representación del trazo del movimiento más desfavorable

Para realizar el estudio de este movimiento se considera la carga nominal del robot, que son aproximadamente 40 gramos. Con el fin de simplificar el estudio se ha hecho una versión reducida del brazo robot en la que no se incluyen engranajes ni ejes. Por ello, el par desconocido se aplicará a la articulación inferior del brazo y, mediante las relaciones de momentos en engranajes:

$$i = \frac{Z2}{Z1} = \frac{M2}{M1} \tag{4}$$

Siendo i, Z1 y Z2 parámetros ya comentados en el apartado "3.5-Modelado de los engranajes" y M1 y M2 el momento transmitido al engranaje 1 y el momento que transmite el engranaje 2 respectivamente.

Como ya se ha comentado en el apartado "3.5.2-Engranajes del brazo" la relación de transmisión de los engranajes que conforman el movimiento a estudiar es de 2 por lo que, se tiene que tener en cuenta que cuando se obtenga el momento máximo que será necesario para el movimiento se tendrá que obtener el que tendrá que transmitir el engranaje del Servomotor despejando de la fórmula anterior:

$$M1 = \frac{M2}{i} = \frac{M2}{2}$$

Se selecciona la opción "Fuerza desconocida" dentro del entorno de simulación dinámica de Inventor y se aplica un par de torsión a la articulación a estudiar. En las especificaciones que se tiene que introducir para poder simular la fuerza desconocida se introduce la posición final a -90 grados (brazo vertical) y el número de pasos en 100, ya que serán suficientes datos para poder realizar el estudio.

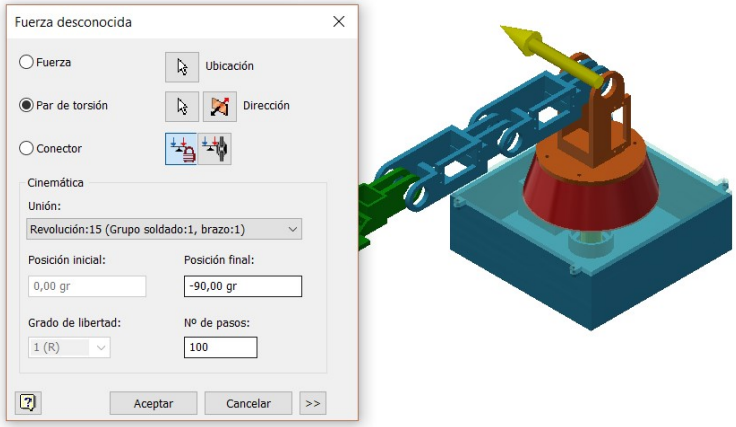


Figura 44. Simulación fuerza desconocida en articulación más desfavorable

Al realizar la simulación aparece el gráfico de salida que muestra el momento de la fuerza desconocida y la posición donde aparece dicho momento. Para estudiar correctamente los datos de salida se exportan a Excel y se realiza un gráfico de representación del momento respecto al ángulo de posición del brazo robot. Además se representa también la línea de tendencia (línea en punteado) y la ecuación de la misma.

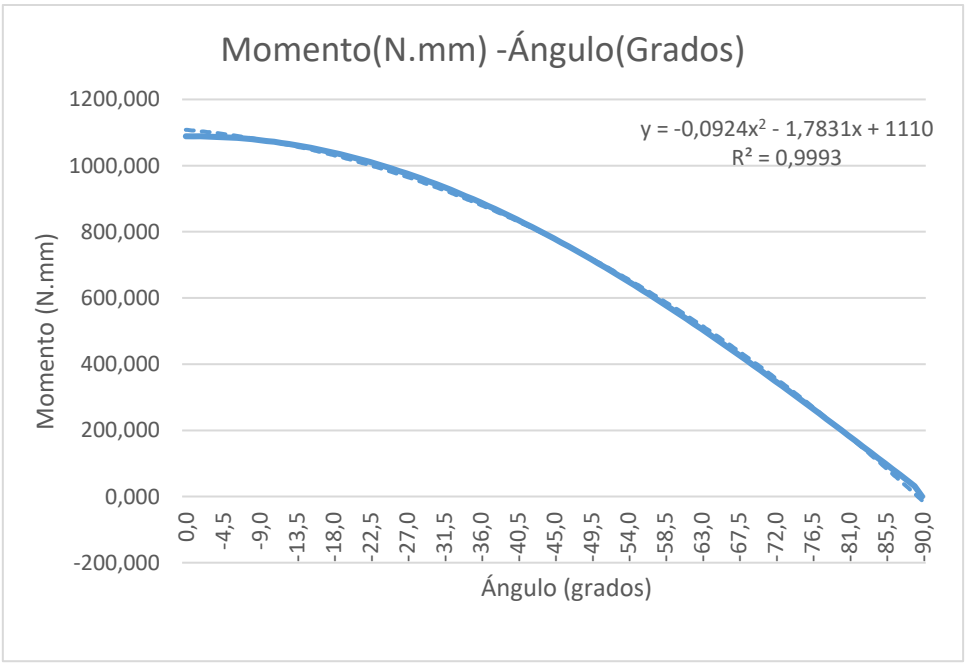


Figura 45. Gráfico del momento respecto al ángulo girado de la articulación del hombro

A continuación, se simula el brazo de nuevo pero esta vez estableciéndole un par de torsión conocido, introduciendo un gráfico de entrada. En el cuadro emergente se introduce la constante de la ecuación

de la gráfica plasmada pero dándole un poco de margen de seguridad para asegurarse el funcionamiento. Por tanto, introducimos un valor constante de 1120 N.mm y se comprueba que efectivamente, puede realizar el movimiento.

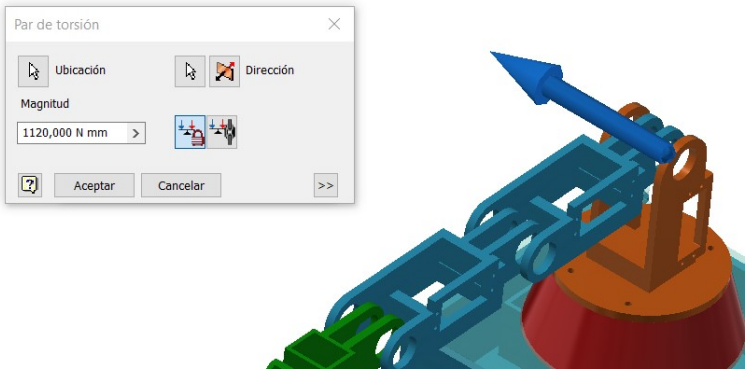


Figura 46. Simulación par de torsión conocido

En conclusión, se tendrá que proporcionar un par mínimo de 1120 N.mm para poder garantizar un buen funcionamiento del brazo robot. Y según la fórmula planteada anteriormente se obtiene el par que tendrá que proporcionar el servomotor:

$$M1 = \frac{1120}{2} = 560\ N.mm \sim 5{,}71\ kgf.cm$$

Conforme se ha comentado en el apartado "3.2.1.1-Servomotor Mg996r" los servomotores que se han seleccionado tienen un par máximo de 9,4 kgf.cm cuando se ha conectado a 4,8V y de 11 kgf.cm cuando se ha conectado a 6 V. Por ello, los servomotores sí que serán capaces de realizar los movimientos que se deseen con una pieza de 48 gramos.

3.7.3- Simulación elementos finitos

Para realizar el estudio de la respuesta estructural al movimiento se crea una simulación del robot completo pero, en este caso, se establece una carga de 5 kg para crear un margen de seguridad y obtener unos resultados más visibles que con una carga de 48 gramos como la del apartado anterior.

Se imponen los movimientos que se desean que haga cada articulación para crear la simulación dinámica. En este caso, se opta por definir el movimiento mediante la posición de las articulaciones (en grados) respecto al tiempo.

Para conseguir una simulación más real, se ha dividido la simulación en 3 partes:

-Parte primera: Apertura de la pinza y descenso de la misma hasta alcanzar una pieza

-Parte segunda: Se ha creado una pieza cúbica a la que se le ha impuesto un peso de 5 kilogramos. La pinza se elevará unos grados y trasladará la pieza 180 grados.

-Parte tercera: La pieza se ha soltado y se realiza el ascenso de la pinza hasta volver a la posición inicial, donde todos los brazos se encuentran en posición vertical.

3.8-Análisis de elementos finitos

Para elaborar el informe de análisis de elementos finitos se proyecta el gráfico de salida a la vez que se simula dinámicamente.

Se va a realizar el análisis a la pieza “Brazo” del ensamblaje en la posición de antebrazo. Para ello, se selecciona el momento máximo en el gráfico de salida en las restricciones de la pieza y se exporta este valor a la opción de Análisis de elementos finitos que proporciona Autodesk Inventor. Además, se observa también la fuerza que se aplica en los momentos máximos de cada pieza.

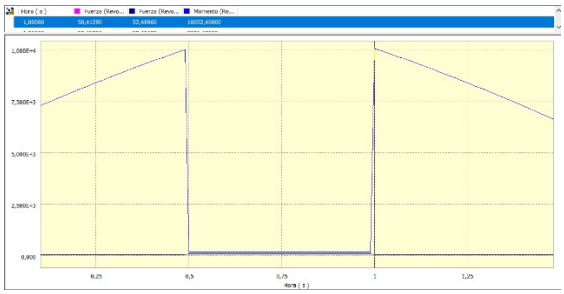


Figura 47. Gráfica del momento en las uniones del antebrazo

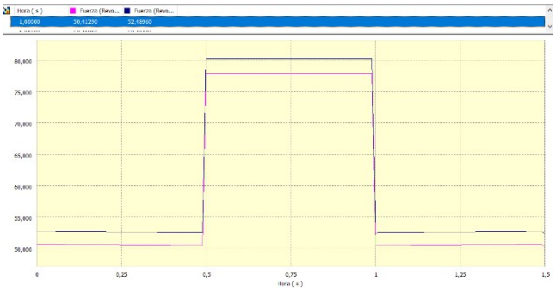


Figura 48. Gráfica de la fuerza en las uniones del antebrazo

A continuación, se crea un nuevo estudio estático de la pieza seleccionada y se crea la vista malla de dicha pieza. Para hacer una mejor aproximación de la malla triangular se configura la malla modificando el tamaño medio del elemento a 0,02. Esta propiedad modifica la fracción de la longitud del cuadro delimitador y, por tanto, crea una malla triangular más ajustada al objeto dibujado ya que los triángulos que configuran la malla son más pequeños.

Previamente a realizar la simulación, se realiza la configuración de convergencia, modificando algunas de las propiedades que aparecen.

Por un lado, el número máximo de refinados se aumenta a 5. Mediante esta característica se indica el número máximo de ciclos de refinado que se realizaran en la convergencia. Este número se alcanzará si no se cumplen los criterios de parada.

Por otra parte, los criterios de parada (%) se disminuyen a 2%. Mediante este valor se indica el porcentaje límite que suspende el refinado cuando la diferencia entre los dos últimos resultados es menor que dicho porcentaje.

Una vez establecidos todos los parámetros, se simula y se observan los resultados obtenidos en la pieza a estudiar.

Primero, se observan los resultados estableciendo como máximo el límite elástico del plástico ABS que es 41 MPa.

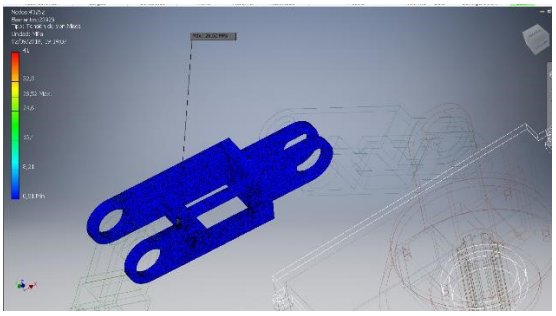


Figura 49. Análisis de fuerzas en el antebrazo fijando el LE como máximo

Como se puede comprobar, la estructura podrá resistir los esfuerzos que se le sometan sin romper ya que, como se muestra en la foto anterior, los valores de tensión en la pieza son mucho menores que su límite elástico.

Aun así, se estudian las zonas más problemáticas de cada pieza fijando como máximo la tension máxima que se ha obtenido.

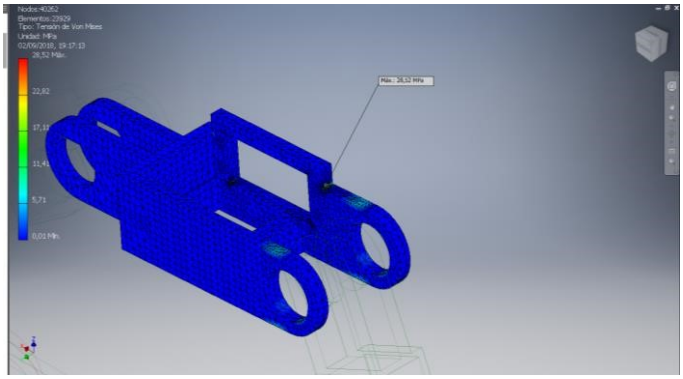


Figura 50. Simulación dinámica del antebrazo fijando como máximo la tensión máxima

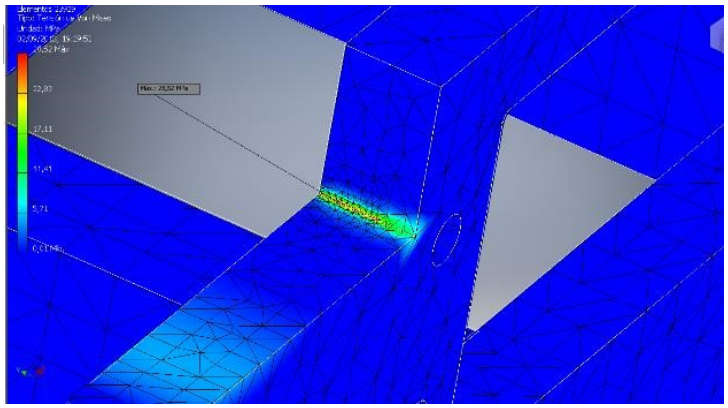


Figura 51. Detalle 1 simulación dinámica

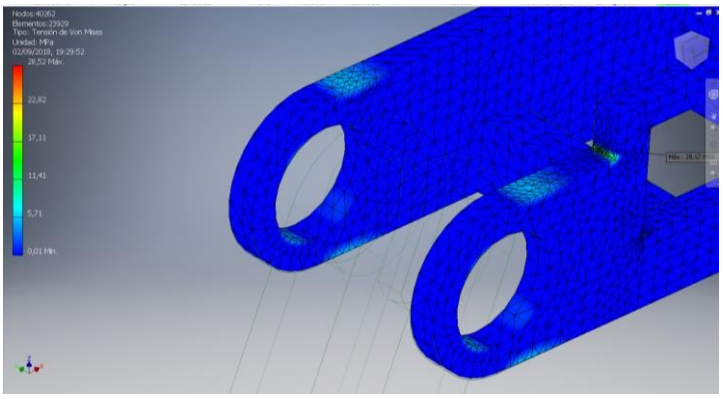


Figura 52. Detalle 2 simulación dinámica

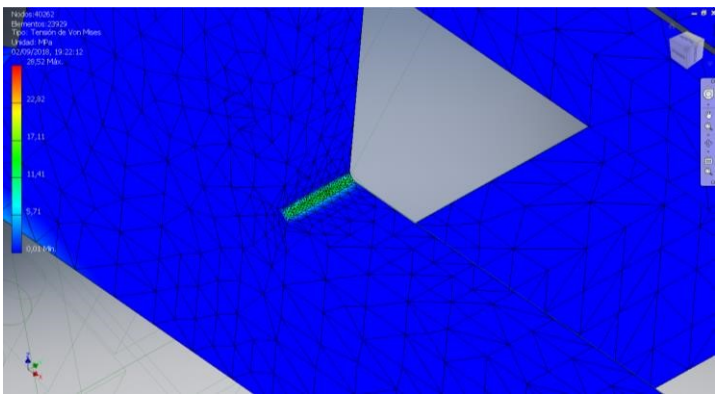


Figura 53. Detalle 3 simulación dinámica

Como se ha podido apreciar en las imágenes anteriores, las zonas más conflictivas son las esquinas, ya que, a menor radio de empalme la esquina actuará más como un concentrador de tensiones. Además también se observan problemas en las zonas cercanas a un agujero ya que dichas zonas son más delgadas y es más fácil que rompa por esos lugares.

3.9-Análisis de datos y mejora de estructura

Observando los resultados obtenidos en el análisis de elementos finitos de cada una de las piezas se llega a la conclusión de que se encuentran problemas en las esquinas, los bordes y los agujeros extruidos (no realizan la rotación).

Se plantean las siguientes mejoras, para conseguir una mejor respuesta de la pieza.

-Realizar empalmes en todas las aristas problemáticas de la pieza y así evitar puntos de concentración de tensiones.

-Reducir el diámetro del agujero problemático a la mitad para, de esta manera, evitar zonas estrechas por donde pueda romper la pieza.

Una vez planteadas las opciones de mejora, se crean una nueva pieza mejorada y se incluye en un nuevo montaje.

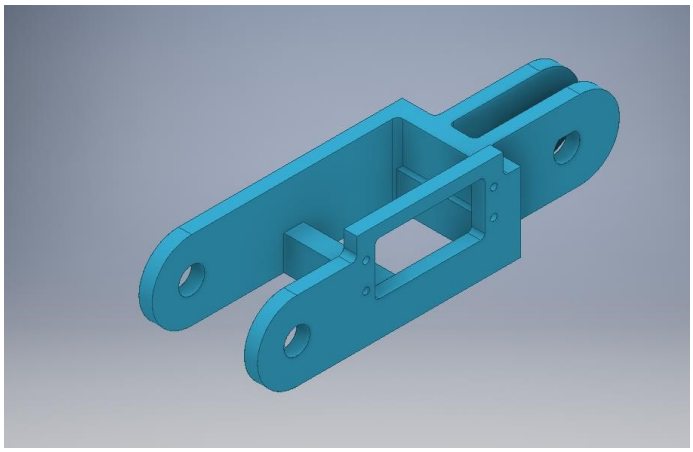


Figura 54. Modelado del antebrazo con mejoras estructurales

Una vez diseñadas e insertadas en el montaje, se realiza el análisis de elementos finitos de nuevo y, se observa la respuesta del montaje a las mejoras. Al igual que en el estudio que se ha realizado en las piezas sin mejora, se observa primero el resultado respecto al límite elástico del ABS como referencia y, se comprueba que la respuesta sigue siendo buena y, además, la tensión máxima que se aplica a la pieza disminuye.

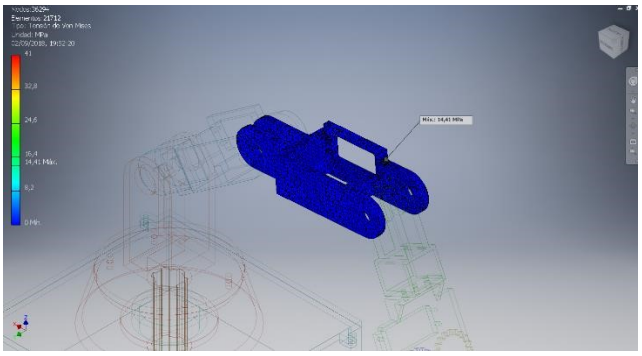


Figura 55. Simulación dinámica antebrazo tras las mejoras con el LE como máximo

Se ha hecho el estudio detallado de cada una de las zonas que se han determinado como conflictivas en el análisis de fuerzas de la pieza en el apartado anterior. Se puede observar una mejora evidente en la respuesta de la pieza. Además, para poder apreciar dichas mejoras, se fija como máximo el máximo que se había obtenido en el análisis de las piezas sin mejora.

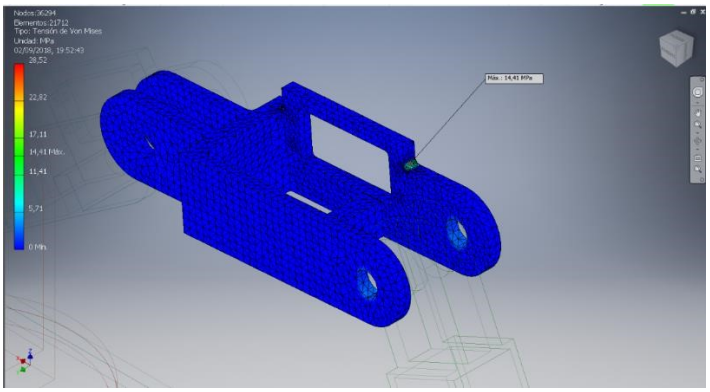


Figura 56. - Simulación dinámica del antebrazo tras las mejoras fijando como máximo la tensión máxima

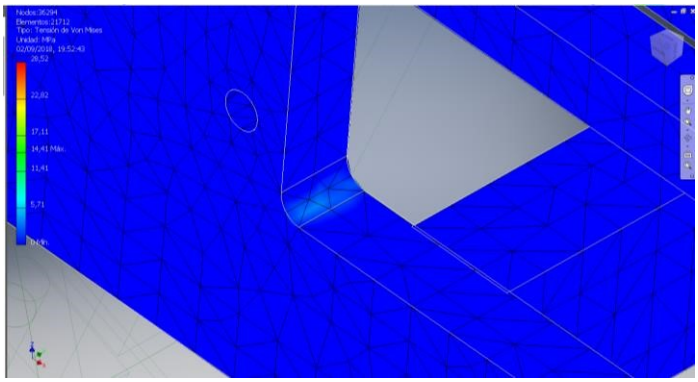


Figura 57. Detalle 1 de la simulación con las mejoras

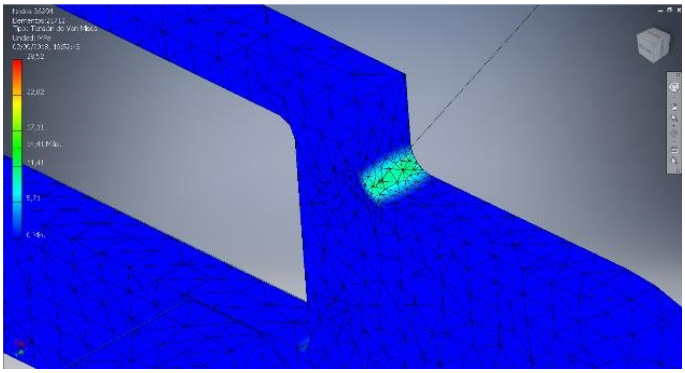


Figura 58. Detalle 2 de la simulación con las mejoras

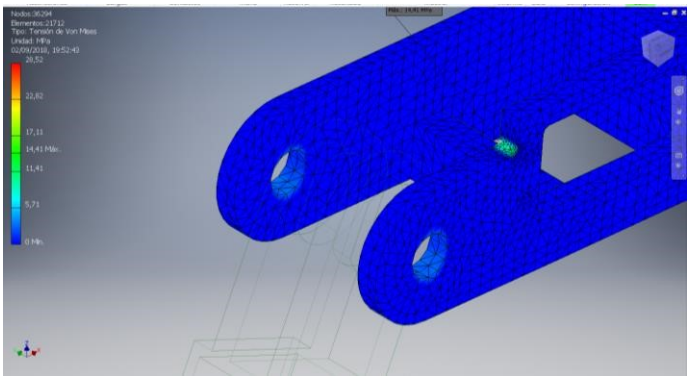


Figura 59. Detalle 3 de la simulación con las mejoras

Por tanto, se puede concluir que, a pesar de que las piezas creadas cumplen con la resistencia a rotura, cabe la posibilidad de mejorar la estructura tratando de evitar roturas o dobleces de la pieza antes de tiempo mediante unos cambios rápidos y sencillos en el diseño como son empalmes o reducción de diámetros y zonas estrechas.

Capítulo 4-Conclusiones

Se puede concluir como resultado que se ha podido realizar un brazo robot antropomórfico controlado a través de bluetooth con un microcontrolador y poder desplazar con él piezas desde un punto a otro. Se han cumplido todos los objetivos marcados satisfactoriamente

Aun así, se podrían realizar muchas mejoras que optimizarían las condiciones del robot. Por un lado se podrían diseñar las piezas de forma más ergonómica y con la capacidad de alojar a los servos en la base del robot y, de esta manera, evitar un extra de cargar en el robot que aumenta en gran cantidad el par que ha de suministrar para poder realizar el movimiento.

Por otro lado, se podría programar el robot empleando la cinemática inversa. Es decir, la localización de un punto y el movimiento de todas sus articulaciones a la vez y, con ello, tener desarrollado un robot mucho más potente.

Además, cabe comentar que se ha realizado un prototipo en pequeña escala pero que, sin embargo, demuestra la posibilidad de implantación de robots más potentes en empresas con un coste bastante más asequible que si se comprara el robot. Queda demostrado pues, la funcionabilidad de la impresión 3D con la que se pueden realizar múltiples diseños de utillajes para una empresa teniendo un coste reducido y un diseño totalmente personalizado.

REFERENCIAS BIBLIOGRÁFICAS

1. Apuntes de la asignatura de Impresión 3D

2. Apuntes de la asignatura de Diseño y desarrollo de aplicaciones móviles

3. Apuntes de la asignatura de Métodos matemáticos.

4. Apuntes de la asignatura de Ingeniería Gráfica.

5. Barrientos, Antonio; Peñín, Luis Felipe y otros (Ed. McGraw-Hill). (2007) . Fundamentos de robótica

6. Torrente Artero, Óscar (Ed. RC Libros). (2013) . Arduino. Curso práctico de formación

7. Millán Gómez, Simón (Ed. Ediciones Paraninfo). (2003) . Procedimientos de mecanizado

8. Younis, Wasin (Ed. Marcombo). (2012) . Inventor y su simulacion con ejercicios prácticos.

9. Autodesk Inventor:
https://www.autodesk.es/products/inventor/overview

10. Diseño paramétrico:
http://www.3dcadportal.com/diseno-parametrico-modelado-parametrico.html

11. Simulación dinámica Inventor:
https://knowledge.autodesk.com/es/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2014/ESP/Inventor/files/GUID-0540FCFE-D9C2-456A-8256-6D014C65CC54-htm.html

12. Impresoras 3D:
http://www.areatecnologia.com/informatica/impresoras-3d.html

13. Formato de impresión:
http://www.r3ald.com/que-es-un-fichero-stl

14. Repetier:
https://www.repetier.com

15. Arduino:
http://arduino.cl/que-es-arduino/

16. Fabricación aditiva:
https://www.sculpteo.com/es/servicios/fabricacion-aditiva/

17. G-CODE:
https://polaridad.es/que-es-g-code/

18. SolidWorks:
http://www.3dcadportal.com/solid-works.html

19. FreeCad:
https://www.freecadweb.org/?lang=es_ES

20. OpenScad:
http://www.pulso.uniovi.es/wiki/index.php/OpenScad

21. Placas Arduino:
https://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/#Arduino_UNO

22. Funcionamiento de un servomotor:
http://panamahitek.com/que-es-y-como-funciona-un-servomotor/

23. Parámetros para la impresión:
http://wiki.ikaslab.org/index.php/Impresi%C3%B3n_3D_paso_a_paso#Variables_que_hay_que_definir_aunque_usemos_perfiles

24. Engranajes rectos:
http://www.mecapedia.uji.es/angulo_de_presion.htm

PRESUPUESTO

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Capítulo 1-Contenido del presupuesto

1.1-Unidades de obra

Se ha dividido el presupuesto en las siguientes unidades de obra:

-Mano de obra: En este cuadro se incluye el salario de la Graduada en Ingeniería Industrial. Se valoran las horas dedicadas a cada fase del proyecto según la implicación de la mano de obra y la dificultad de la tarea.

-Material y herramientas: En este cuadro se han incluido todos los productos, software y hardware que se han empleado valorando su amortización y el precio.

1.2-Cuadro de precios

1.2.1-Cuadro de precios nº1- Mano de obra.

El número de horas totales para la realización del proyecto teniendo en cuenta que se han empleado 12 semanas de trabajo, invirtiendo 5 horas diarias en cada día laborable son:

Horas trabajadas= 5h. 5 días/semana. 12 semanas= 300 *horas*

Tabla 1: Secuenciación del trabajo por semanas

Duración total(semanas)	**Descripción actividad**
3	Planteamiento de diseño
1	Modelado
1	Generación de planos
1	Estudio y simulación del comportamiento mecánico
3	Impresión 3D de las piezas y montaje del prototipo
3	Programación del microcontrolador

En todas las etapas se ha contado con el trabajo de un graduado en Ingeniería en Tecnologías Industriales.

Para una mejor secuenciación de las tareas, se dividió el proyecto en diferentes etapas:

-Planteamiento de diseño: previamente a comenzar el modelado 3D se realizó un diseño conceptual del producto, donde se plantearon diferentes opciones estructurales de las piezas que componen el brazo, así como de los dispositivos electrónicos a escoger para el control del

mismo. Esta etapa del proyecto se valoró en 75 horas aproximadamente y el coste es de 20 euros la hora,

-Modelado: En esta parte del trabajo se engloba el diseño de cada una de las partes del brazo robot y la elaboración del ensamblaje 3D, donde se observa el prototipo virtual. Se han invertido sobre 25 horas totales y se valora en 45 euros.

-Generación de planos: La realización de planos se realizó aproximadamente al mismo tiempo que el modelado de las piezas y se realizó en 25 horas con un coste estimado de 25 euros la hora.

-Estudio y simulación del comportamiento mecánico: Durante esta etapa en la cual se simuló el prototipo creado para observar la respuesta a diferentes esfuerzos antes de realizar la impresión 3D se emplearon 25 horas con un coste de 40 euros la hora.

- Impresión 3D de las piezas y montaje del prototipo: Para la impresión de las piezas y el ensamble del mismo se emplearon 75 horas con un coste estimado en 20 euros la hora.

-Programación del microcontrolador: En el transcurso de esta etapa se realizó el montaje de los componentes electrónicos, la programación en Arduino IDE y la elaboración de la aplicación móvil en Android Studio. Para esta etapa se emplean 75 horas con un coste de 35 euros la hora.

-Redacción del documento del proyecto: Durante la ejecución del proyecto se redacta de forma paralela los documentos del mismo. Se estima el tiempo destinado a esta etapa en 75 horas. Como no se requieren conocimientos específicos se valora el coste en 20 euros la hora.

Tabla 2- Costes de la mano de obra

Descripción	Participante	Unidad básica	Importe unitario	Cantidad (h)	Precio total
Planteamiento de diseño	Graduada en Ingenierías industriales	h	20	75	1500 €
Modelado	Graduada en Ingenierías industriales	h	45	25	1125 €
Generación de planos	Graduada en Ingenierías industriales	h	25	25	625 €
Estudio y simulación del comportamiento mecánico	Graduada en Ingenierías industriales	h	40	25	1000 €
Impresión 3D de las piezas y montaje del prototipo	Graduada en Ingenierías industriales	h	20	75	1500 €
Programación del microcontrolador	Graduada en Ingenierías industriales	h	35	75	2625 €
Redacción del documento del proyecto:	Graduada en Ingenierías industriales	H	20	75	1500 €
Total					9875 €

1.2.2-Cuadro de precios nº2- Materiales y herramientas

En este apartado se comentan las herramientas y el material empleado para la realización del TFG.

Primero, se comentan las herramientas que se han empleado.

-Impresión 3D: El precio del material para la impresión de todas las piezas es de 18 euros cada unidad de bobina de plástico PLA. Para este proyecto en concreto se han consumido 1,5 bobinas.

-Servomotor Mg996r: Para el proyecto se han empleado 5 unidades de este producto. El precio unitario de este producto es de 6.85 euros.

-Arduino Nano: La placa seleccionada para la realización del proyecto tiene un precio de 3.74 la unidad y solo se ha empleado una unidad de este producto.

-Kit componentes eléctricos: Para la realización del circuito eléctrico se han empleado componentes como una Protoboard, reguladores, cables jumper macho-macho o Leds para hacer pruebas con la placa y el módulo bluetooth. Para abaratar costes se decidió comprar un kit que incluyese estos productos y su precio es de 6.69 euros.

-Módulo Bluetooth HC-06: El coste del módulo empleado para la transmisión de datos es de 3.99 euros por unidad.

-Reprografía: Para la impresión de los documentos del proyecto, de los planos y para la encuadernación de todos ellos se destina un presupuesto de 60 euros.

-Rodamientos rígidos de bolas: Se emplean 2 rodamientos de distintos tamaños para facilitar el giro del brazo robot en la base. Se eligen dos rodamientos Zokol de 50*65*7mm y de 20*37*9mm, y su coste es de 3,94 euros la unidad y 2.85 euros la unidad, respectivamente.

Tabla 3-Precio de los materiales

Concepto	**Precio unitario concepto (€/unidad)**
Impresión 3D	18 €
Servomotor Mg996r	6,85 €
Arduino Nano	3,74 €
Kit componentes eléctricos	6,69 €
Módulo Bluetooth HC-06	3,99 €
Rodamiento rígido de bolas pequeño	2,85 €
Rodamiento rígido de bolas grande	3,94 €
Reprografía	60 €

Para el cálculo del precio a amortizar de cada programa que se va a comentar a continuación se ha empleado la siguiente fórmula:

$$\textit{Precio a amortizar (€/hora)} = \frac{Precio\ licencia\left(\frac{€}{año}\right)}{Amortización(h/año)} \quad (5)$$

Se estima que las licencias anuales se emplean aproximadamente 1400 horas al año.

Por otro lado, se realiza un estudio del uso y la amortización de los diferentes sistemas de software que se han empleado.

-Autodesk Inventor Professional 2014: El precio de dicho programa vendrá dado por el coste de la suscripción anual a Inventor Professional, por lo que se cuenta con una amortización de un año. Consultando en la página oficial de Autodesk se puede fijar el precio en 2553,10 euros.

-Microsoft Office Professional 2016: Este software ha sido empleado tanto para la redacción del Proyecto como para el control de datos en simulación dinámica. El precio de la licencia del pack Office 365 Personal es de 69 euros al año. Como en el caso anterior, se fija la amortización en un año de suscripción. Se considera un uso de 150 horas.

-Windows 10: El coste de la licencia anual de la cual dispone el ordenador que se emplea para la realización del proyecto es de 50 € al año y se considera una amortización de un año.

Tabla 4-Precio a amortizar de Software

Software	**Precio licencia (€/año)**	**Amortización (h/año)**	**Precio que amortizar (€/hora)**
Autodesk Inventor Professional 2014	2553,10	1400	1.82364
Microsoft Office Professional 2016	69	1400	0.0493
Windows 10	50	1400	0.0357

Por último, respecto a los sistemas hardware utilizado también se ha realizado un estudio sobre la amortización.

$$\textit{Precio a amortizar (€/año)} = \frac{Precio\ licencia\left(\frac{€}{\text{año}}\right)}{Amortización(h/\text{año})}$$

-Ordenador portátil: Para la redacción del proyecto y la programación de las aplicaciones se ha empleado un ACER Aspire v5. El precio fijado de este modelo de ordenador portátil en el momento de su compra fue de 1000 euros. La devaluación del precio del ordenador no será considerada en el cuadro de precios. Se estima el uso del ordenador en 8 horas de jornada laboral, considerando 264 días laborales y, por tanto, 2112 horas al año.

-Impresora Lion Pro 3D: El precio de la impresora empleada para el proyecto es de 1399 euros en el momento de su adquisición. Para realizar el trabajo se han empleado 50 horas de impresión. Como en el caso anterior, no se cuenta la devaluación del precio de la impresora. La impresora 3D no se emplea tanto, así que se considera una amortización de 800 horas al año.

Tabla 5-Precio a amortizar de Hardware

Hardware	**Precio total (€/año)**	**Amortización(h/año)**	**Precio que amortizar (€/hora)**
Ordenador portátil	1000	2112	0.4734
-Impresora Lion Pro 3D	1399	800	1.7487

Por último, se obtiene el presupuesto de todos los materiales y herramientas:

Tabla 6-Costes de materiales y herramientas

Descripción	**Importe unitario**	**Cantidad**	**Coste calculado (€)**
Materiales			
Impresión 3D	18 €/bobina	1,5 unidades	27 €
Servomotor Mg996r	6,85 €/unidad	5 unidades	34,25 €
Arduino Nano	3,74 €/unidad	1 unidad	3,74 €
Kit componentes eléctricos	6,69 €/unidad	1 unidad	6,69 €
Módulo Bluetooth HC-06	3,99 €/unidad	1 unidad	3,99 €
Rodamiento rígido de bolas pequeño	2,85 €/unidad	1 unidad	2,85 €
Rodamiento rígido de bolas grande	3,94 €/unidad	1 unidad	3,94 €
Montaje brazo robot			
Amortización Autodesk Inventor	1.82364	75 h	136,773 €
Amortización impresora 3D	1.7487	50 h	87,435 €
Amortización Windows 10	0.0357	150 h	5,355 €
Amortización ordenador	0.4734	150 h	71,01 €
Desarrollo de la aplicación			
Amortización ordenador	0.4734	150 h	71,01 €
Amortización Microsoft office	0.0493	150 h	7,395 €
Amortización Windows 10	0.0357	150 h	5,355 €
UO4-Redacción del documento			
Reprografía	60	1 unidad	60 €
Subtotal			526,793 €

Capítulo 2-Presupuesto general

Partiendo de los costes subtotales obtenidos, se le añaden un 2% de costes indirectos y a dicho resultado un 21% de IVA.

Tabla 7-Costes totales del proyecto

Descripción	**Coste**
Mano de obra	9875 €
Materiales y herramientas	526,793 €
Subtotal	10401,793 €
Costes indirectos	2%
Subtotal	208,03 €
Coste sin IVA	10609,82 €
IVA	21%
Coste total del proyecto	12837,89 €

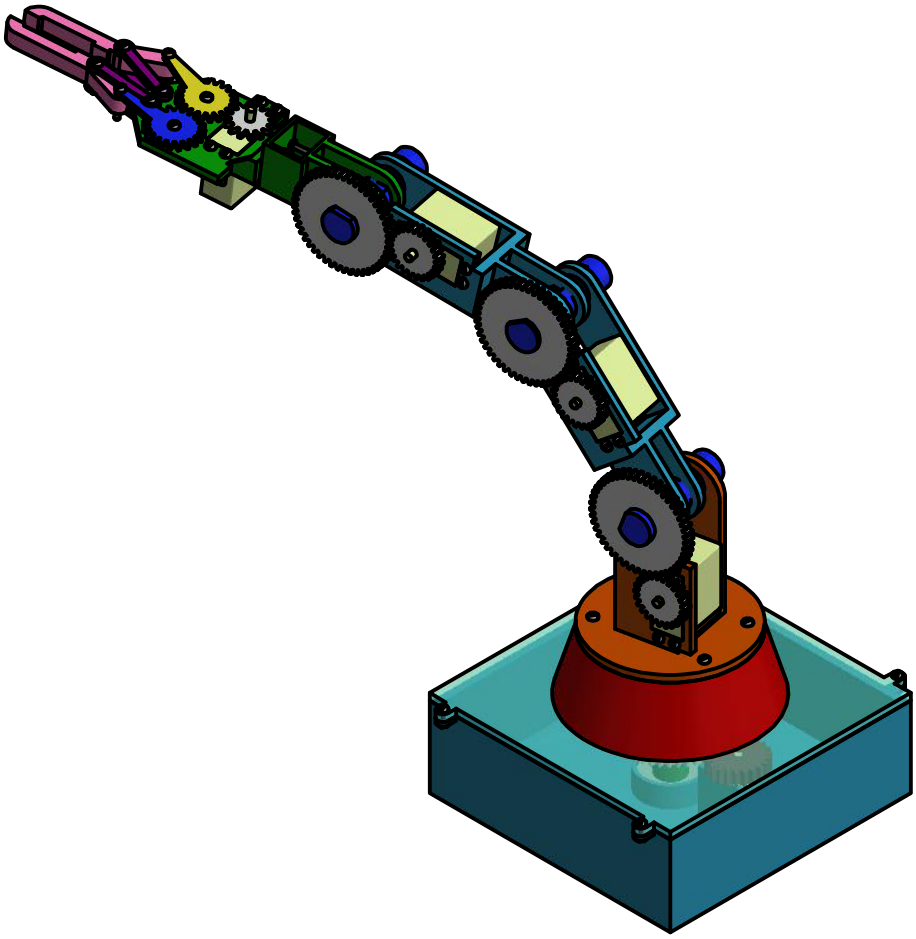
Por tanto, el coste total de este TFG es de:

DOCE MIL OCHOCIENTOS TREINTA Y SIETE EUROS CON OCHENTA Y NUEVE CÉNTIMOS

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

PLANOS

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560



TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

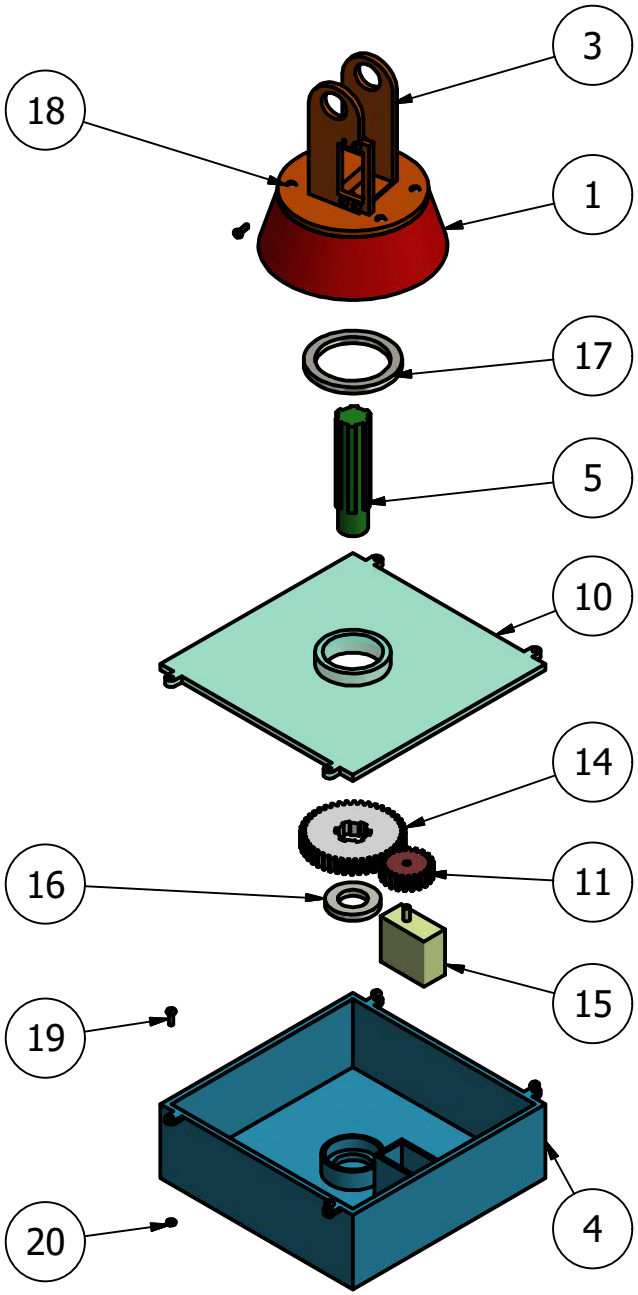
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Autor: Iris Vila Castellá

Proyecto:

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Fecha:	Escala:
28.08.18	1:2
Plano:	**Nº Plano:**
Plano de conjunto	1/13



LISTA DE PIEZAS			
ELEMENTO	CTDAD	Nº DE PIEZA	DESCRIPCIÓN
1	1	Base del brazo robot	
3	1	Brazo inferior	
4	1	Caja- Parte inferior	
5	1	Eje estriado	
10	1	Caja-Tapa	
11	1	Engranaje recto 1	
14	1	Engranaje recto 2	
15	1	Servomotor Mg996r	
16	1	Rodamiento pequeño	
17	1	Rodamiento grande	
18	4	AS 1427 - M3 x 20	Tornillos métricos para maquinaria ISO
19	8	AS 1427 - M3 x 10	Tornillos métricos para maquinaria ISO
20	12	ISO 4035 - M3	Tuerca hexagonal

TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Autor: Iris Vila Castellá

Proyecto:

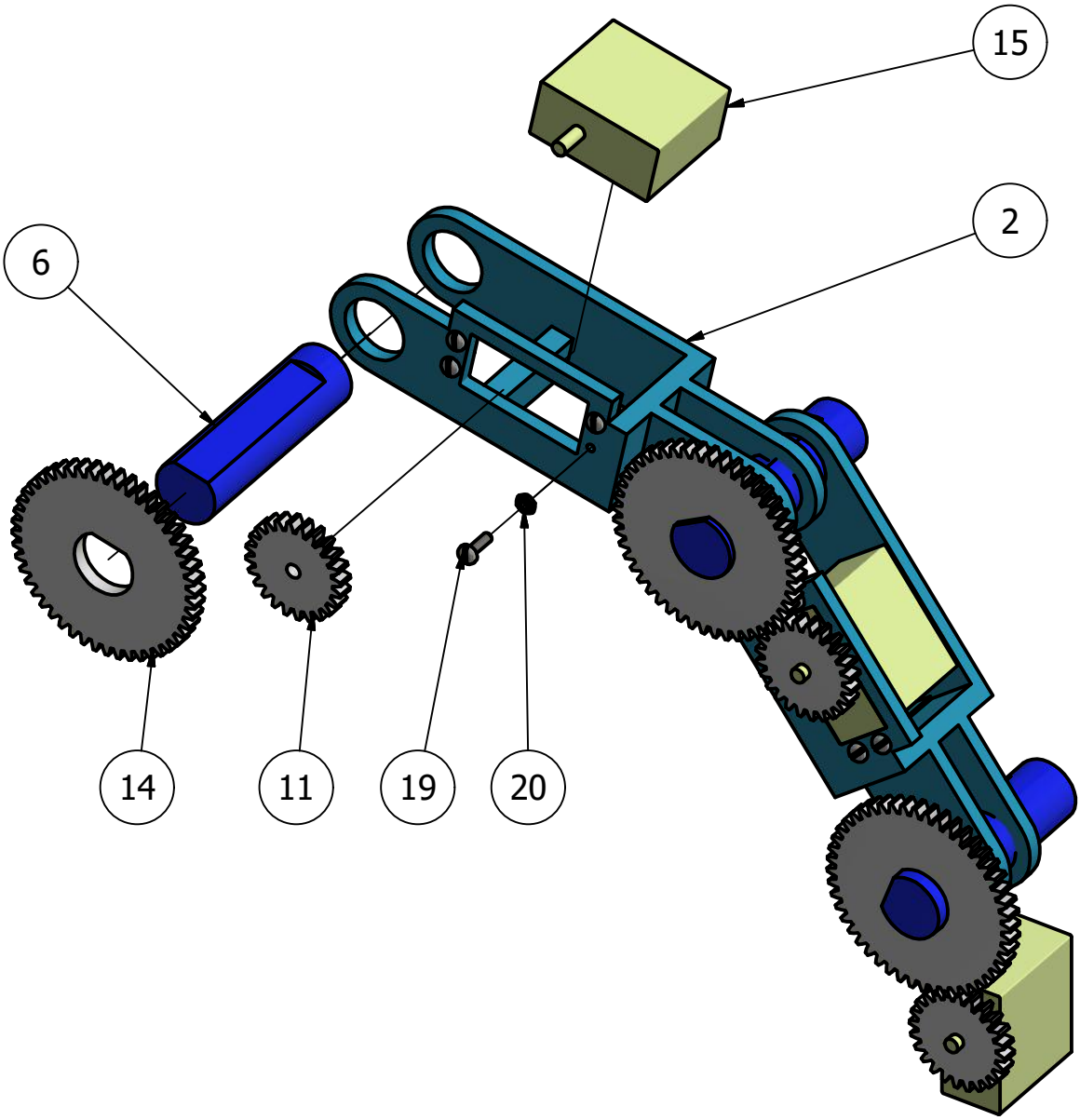
Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Fecha: 28.08.18

Escala: 1:5

Plano: Subconjunto base robot

Nº Plano: 2/13



LISTA DE PIEZAS			
ELEMENTO	CTDAD	Nº DE PIEZA	DESCRIPCIÓN
2	2	Brazo del robot	
6	3	Eje del brazo	
11	3	Engranaje recto 1	
14	3	Engranaje recto 2	
15	3	Servomotor Mg996r	
19	8	AS 1427 - M3 x 10	Tornillos métricos para maquinaria ISO
20	8	ISO 4035 - M3	Tuerca hexagonal

TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

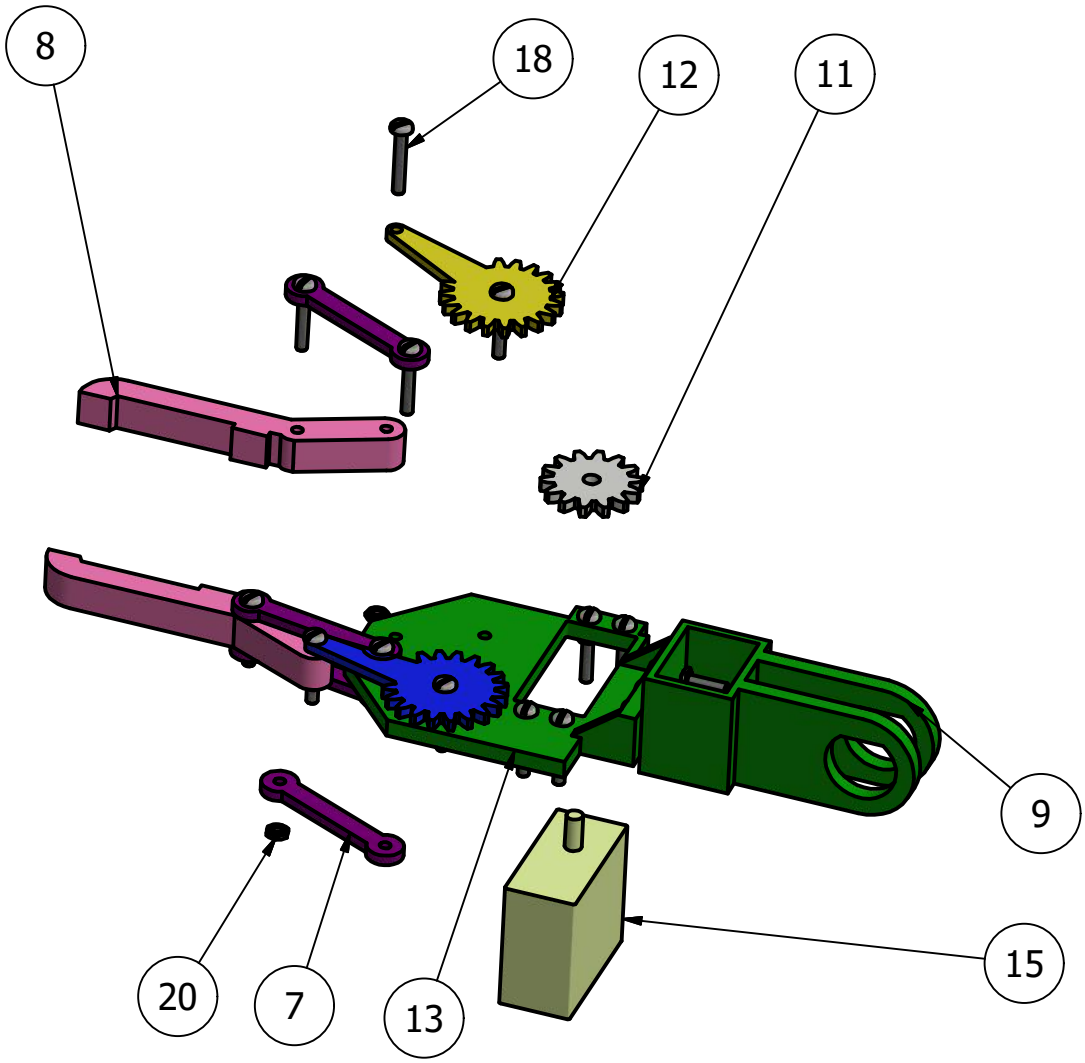
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Autor: Iris Vila Castellá

Proyecto:

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Fecha:	Escala:
28.08.18	1:2
Plano:	Nº Plano:
Subconjunto brazo robot	3/13



LISTA DE PIEZAS			
ELEMENTO	CTDAD	Nº DE PIEZA	DESCRIPCIÓN
7	4	Pieza auxiliar	
8	2	Pinza	
9	1	Base de la pinza	
11	1	Engranaje recto 1	
12	2	Engranaje pinzas	
13	1	Enganche de la pinza	
15	1	Servomotor Mg996r	
18	15	AS 1427 - M3 x 20	Tornillos métricos para maquinaria ISO
20	15	ISO 4035 - M3	Tuerca hexagonal

TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Autor: Iris Vila Castellá

Proyecto:

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

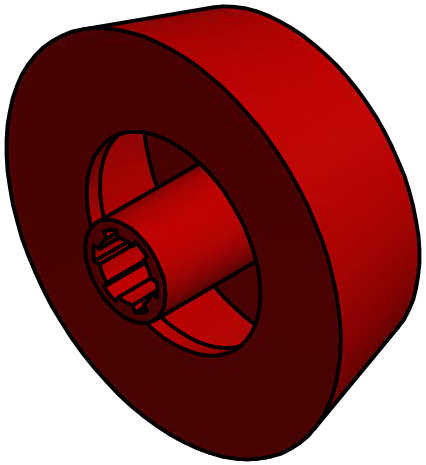
Fecha: 28.08.18

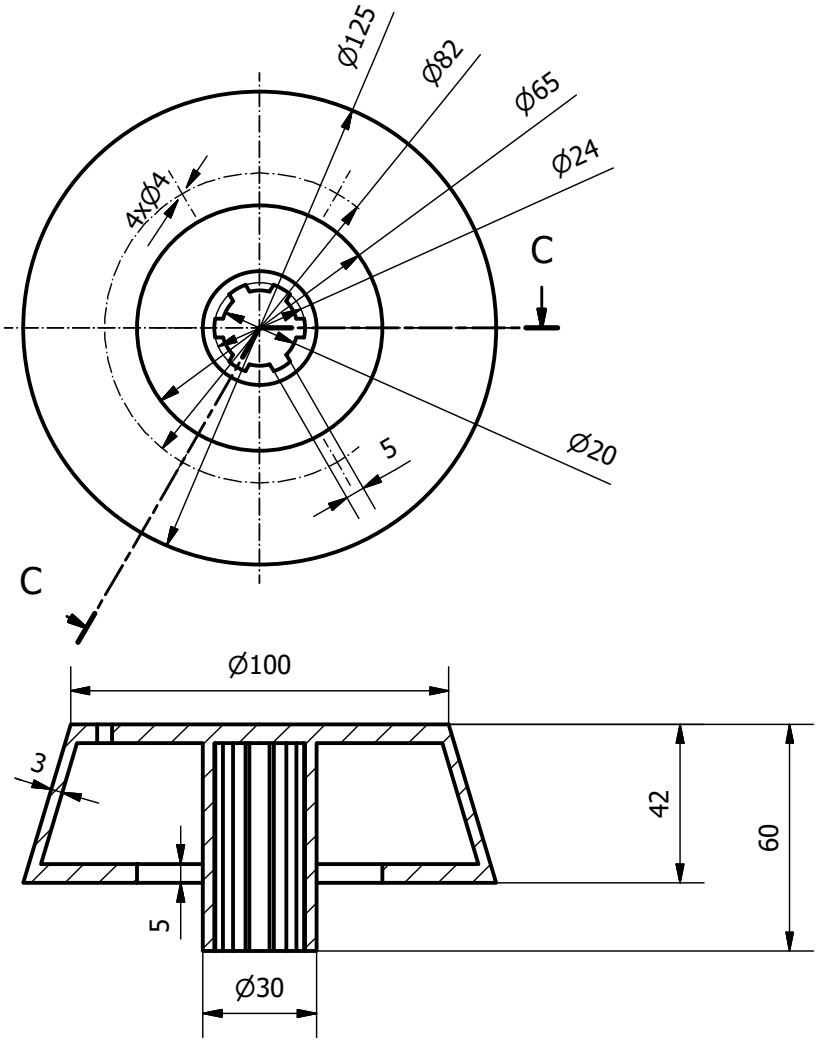
Escala: 1:2

Plano: Subconjunto pinza robot

Nº Plano: 4/13

1





TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Autor: Iris Vila Castellá

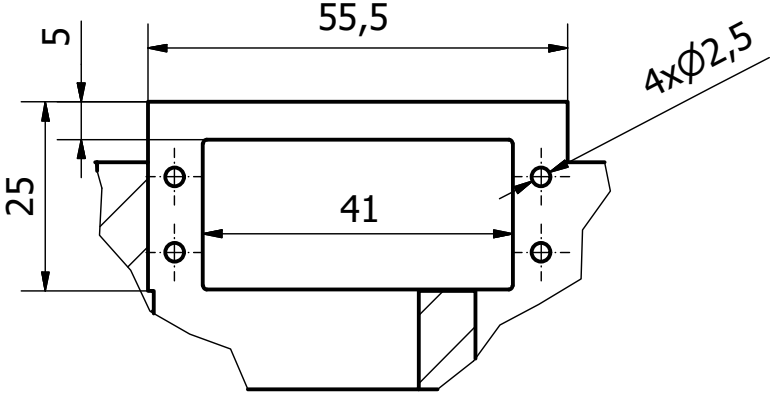
Proyecto:

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

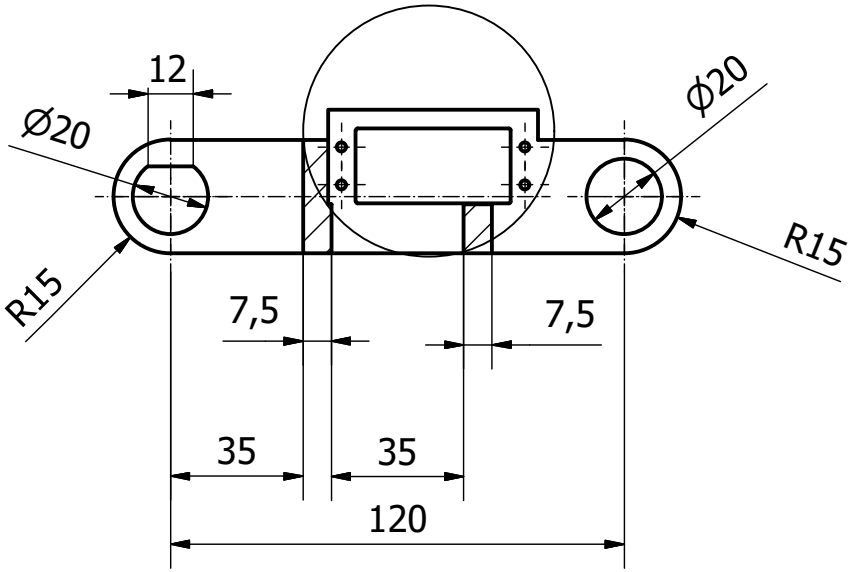
Fecha:	Escala:
28.08.18	1:2
Plano:	Nº Plano:
Base del brazo robot	5/13

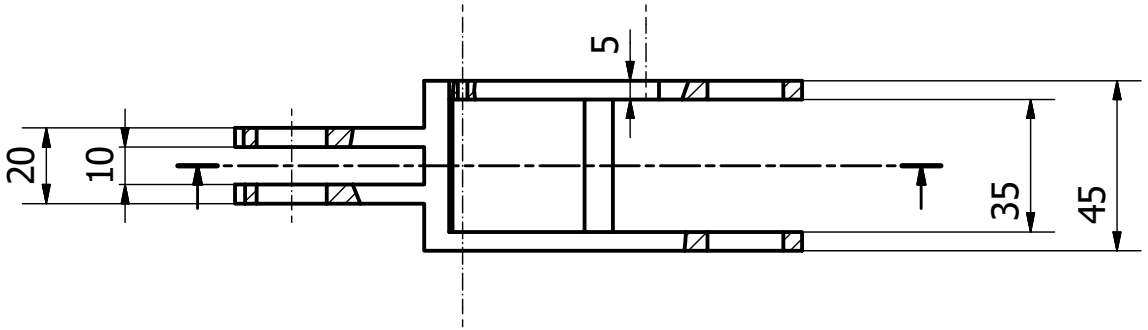
2

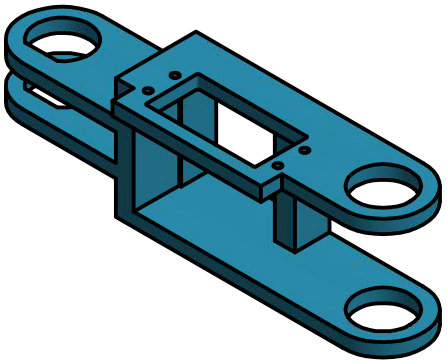
B (1 : 1)



B







TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Autor: Iris Vila Castellá

Proyecto:

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Fecha: 28.08.18

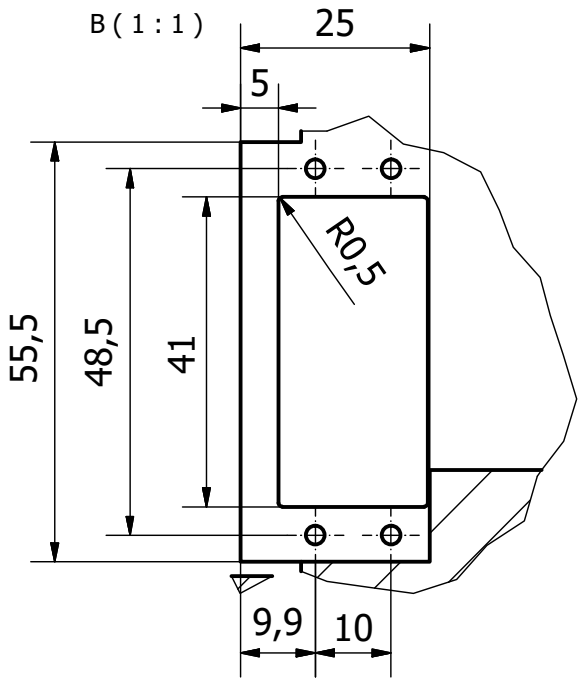
Escala: 1:2

Plano: Brazo del robot

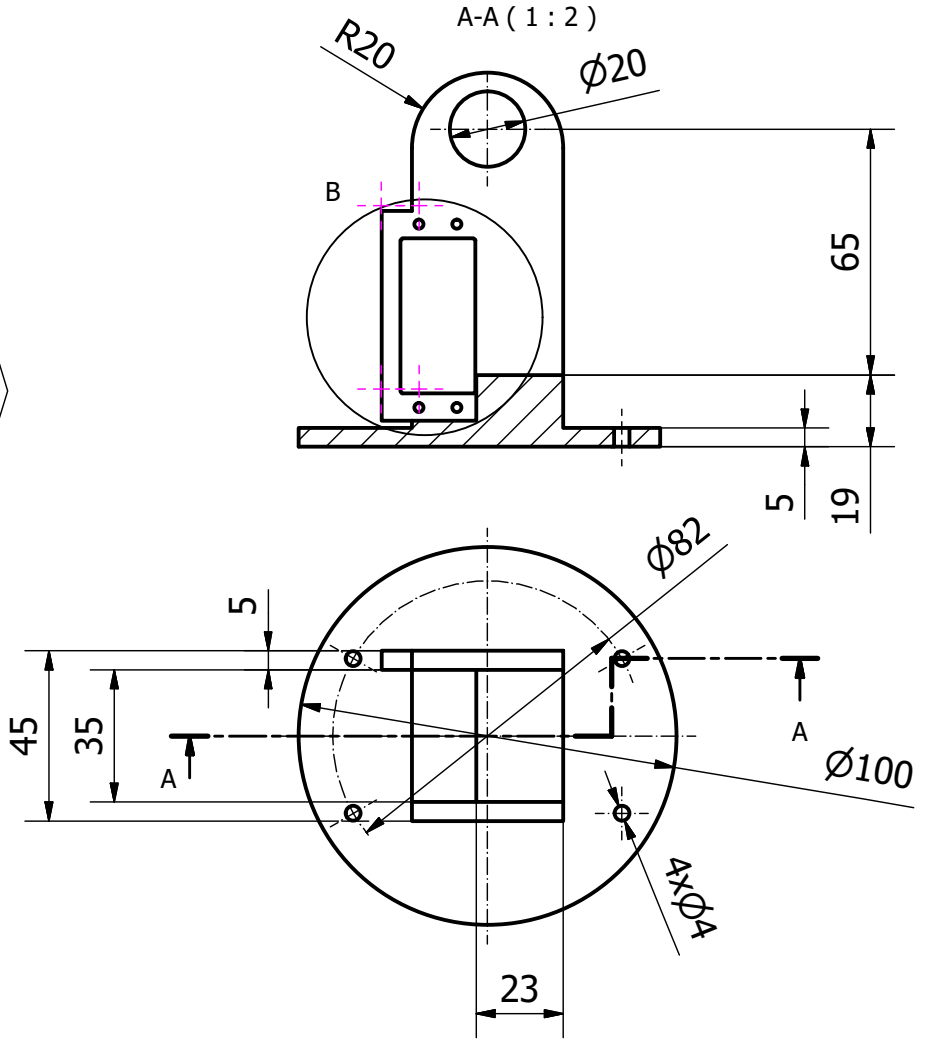
Nº Plano: 6/13

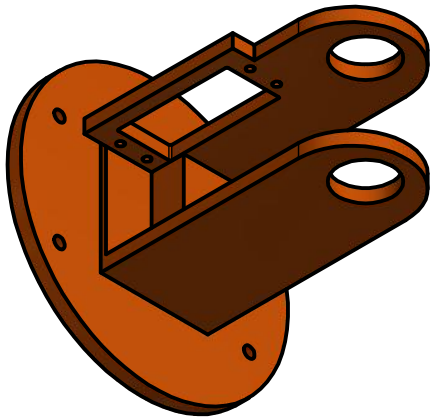
3

B (1 : 1)



A-A (1 : 2)





TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

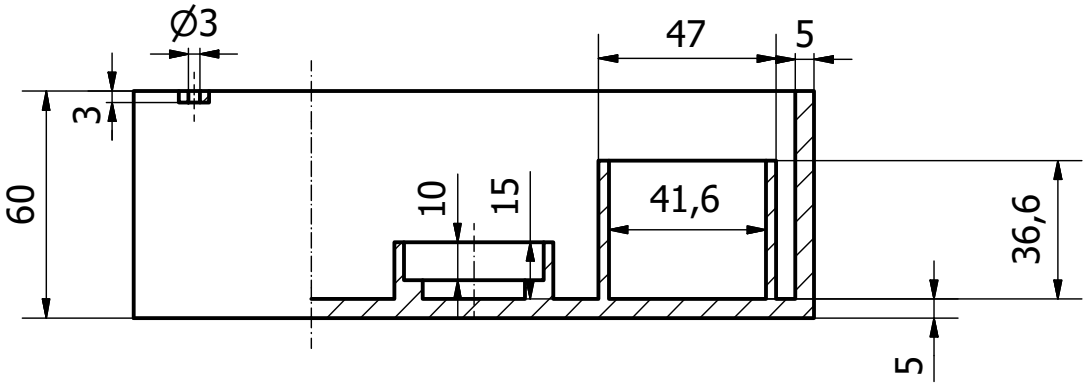
Autor: Iris Vila Castellá

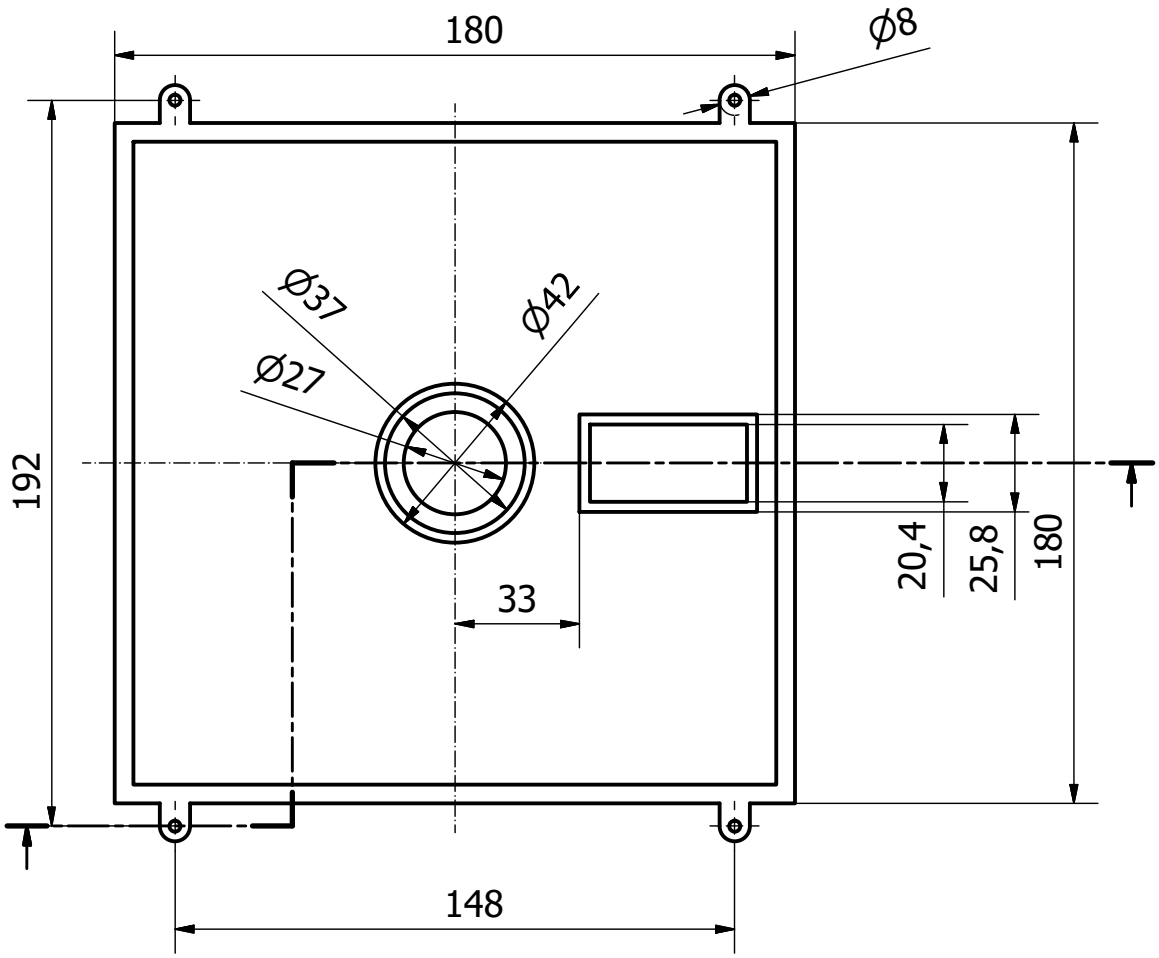
Proyecto:

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Fecha:	Escala:
28.08.18	1:2
Plano:	**Nº Plano:**
Brazo inferior	7/13

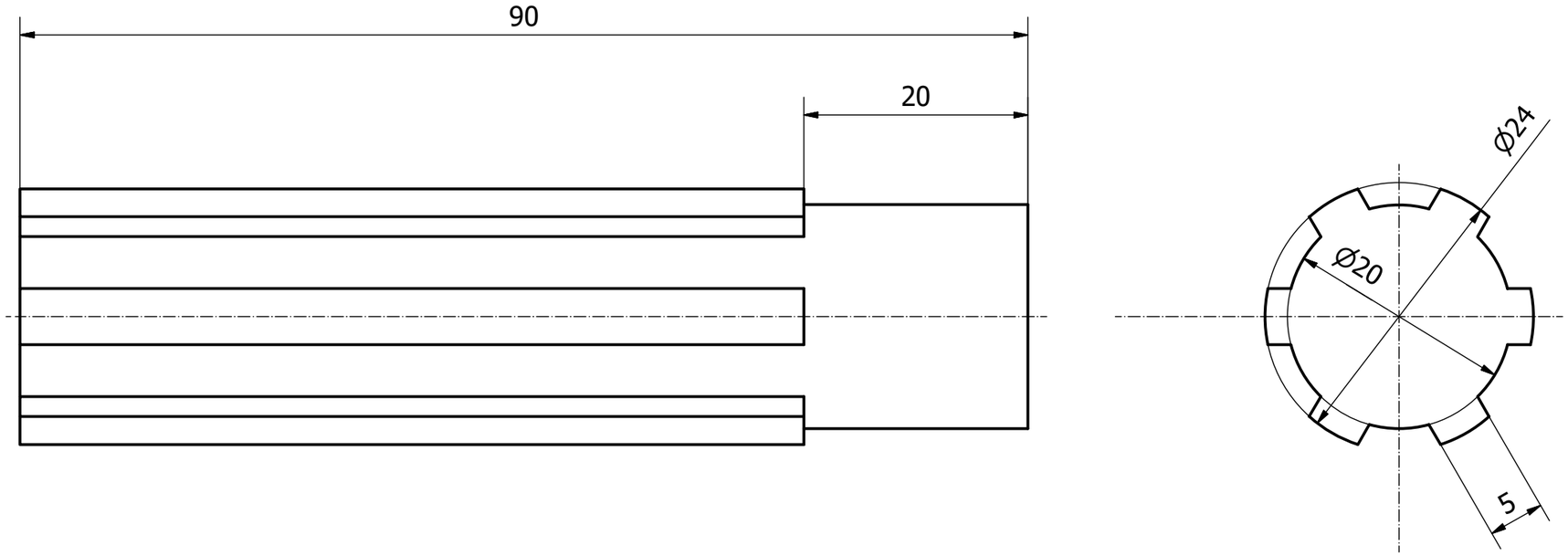
4



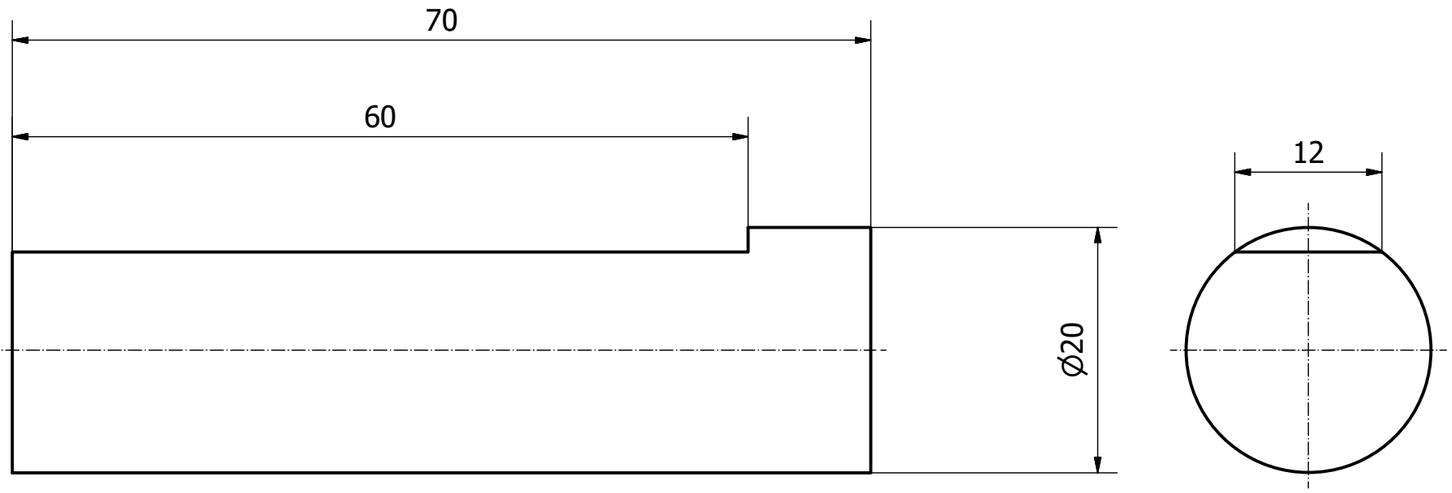


TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL	Proyecto: Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA	Fecha: 28.08.18 / Escala: 1:2
Autor: Iris Vila Castellá	Plano: Caja - Parte inferior / Nº Plano: 8/13

5



6



TRABAJO FIN DE GRADO EN INGENIERÍA INDUSTRIAL

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

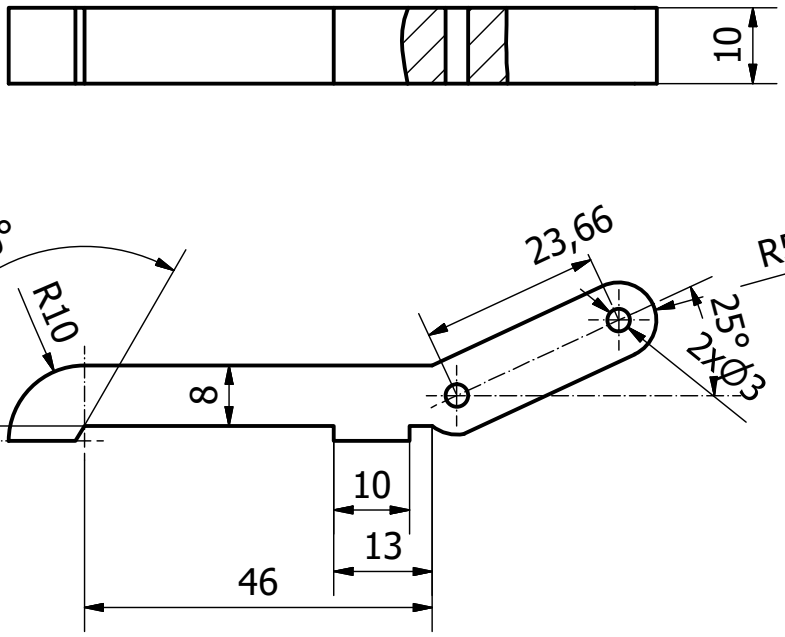
Autor: Iris Vila Castellá

Proyecto:

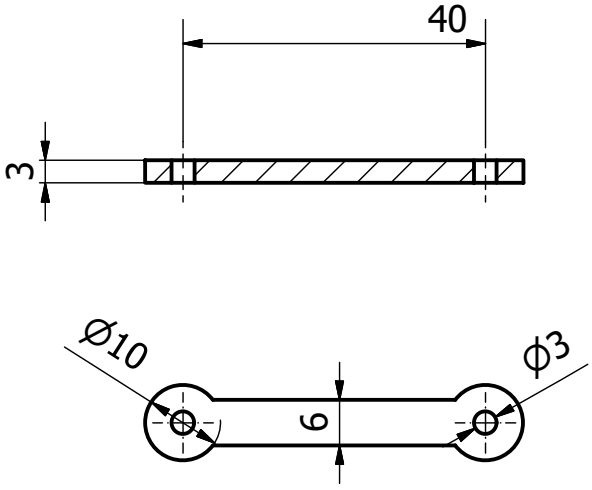
Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Fecha:	Escala:
28.08.18	2:1
Plano:	**Nº Plano:**
Eje estriado	9/13

8

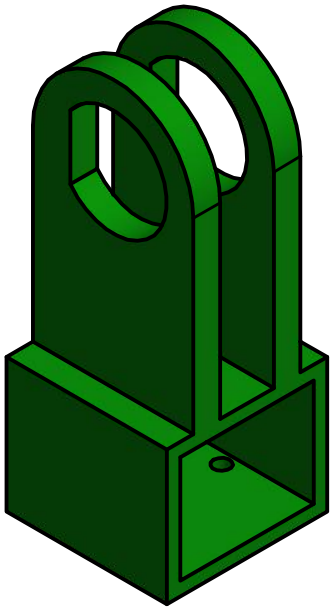


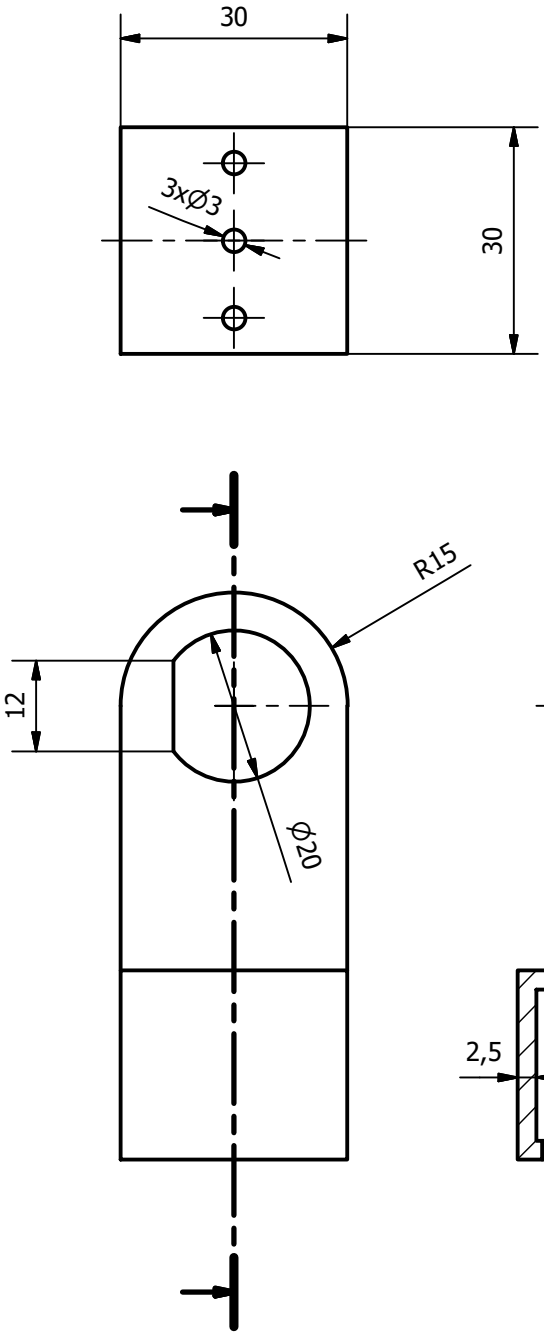
7



TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL	Proyecto: Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA	Fecha: 28.08.18 / Escala: 1:1
Autor: Iris Vila Castellá	Plano: Pinza y Pieza auxiliar / Nº Plano: 10/13

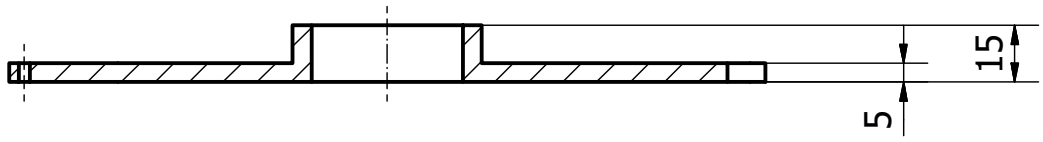
13

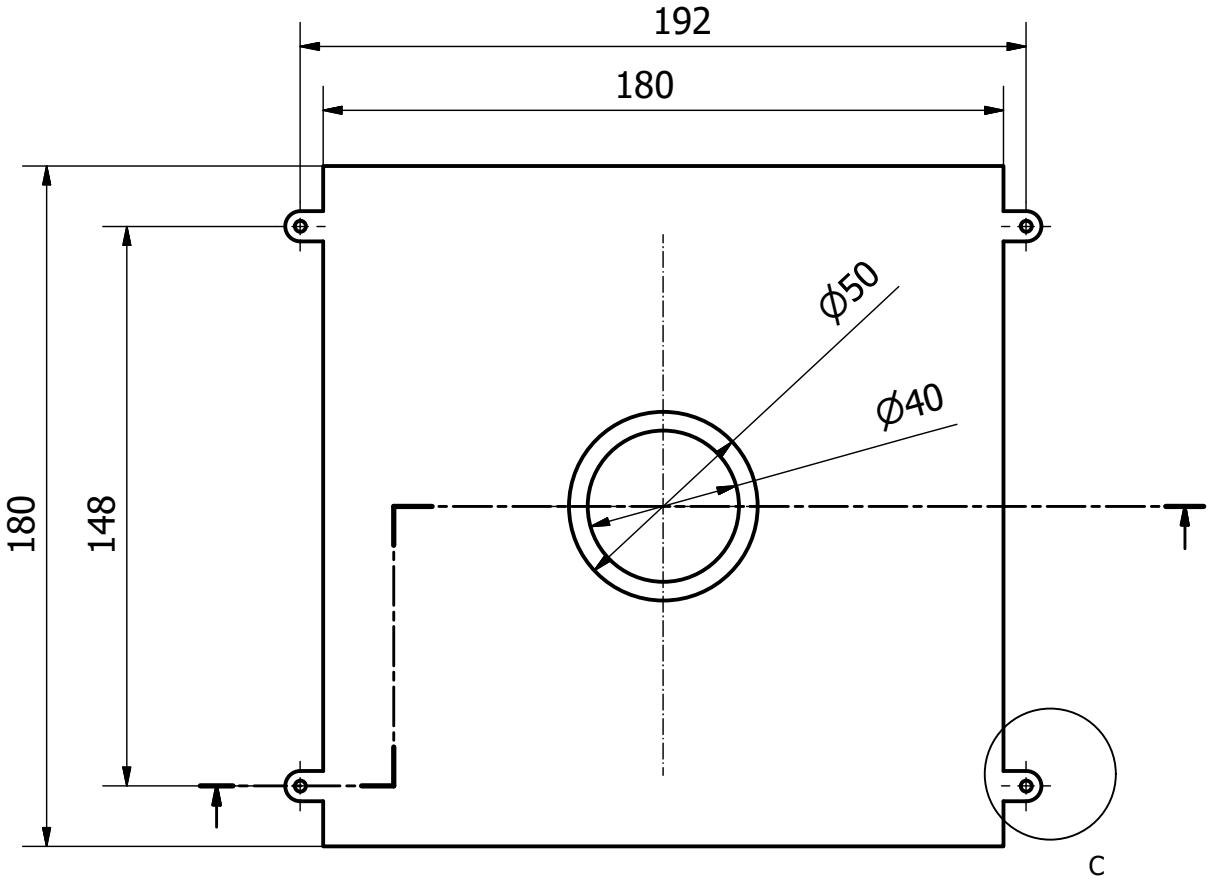




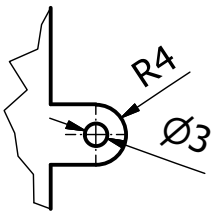
TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL	Proyecto: Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA	Fecha: 28.08.18 / Escala: 1:1
Autor: Iris Vila Castellá	Plano: Enganche de la pinza / Nº Plano: 13/13

10

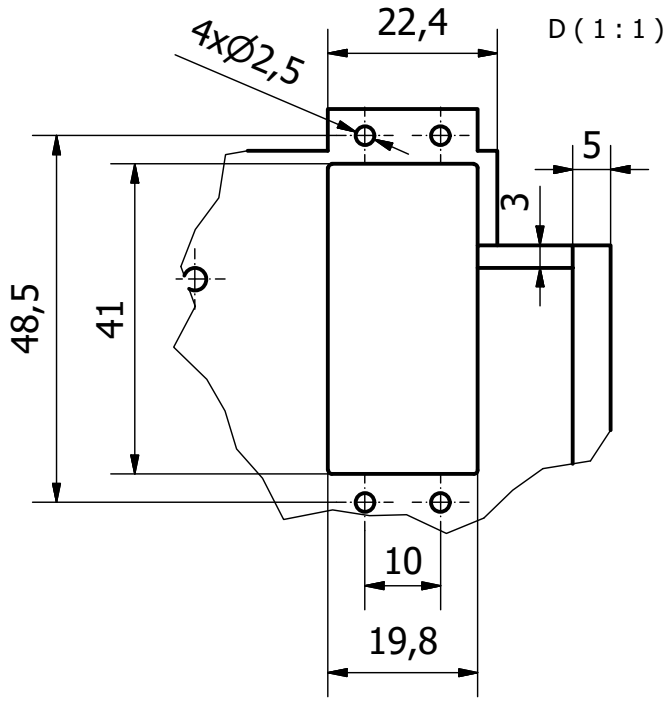




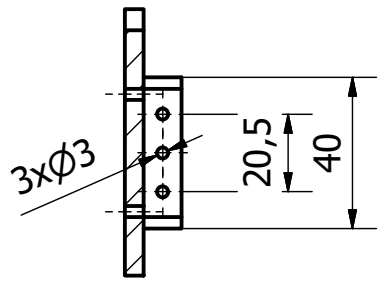
C (1 : 1)



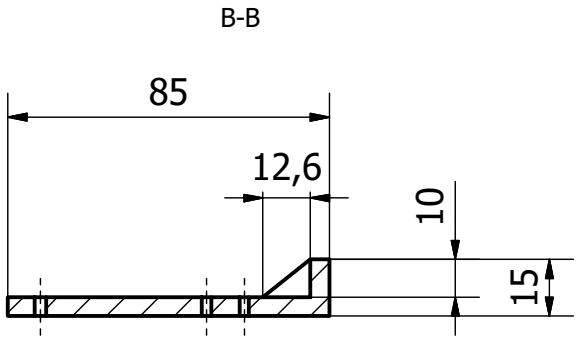
TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL	Proyecto: Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA	Fecha: 28.08.18 / Escala: 1:1
Autor: Iris Vila Castellá	Plano: Caja - Tapa / Nº Plano: 12/13



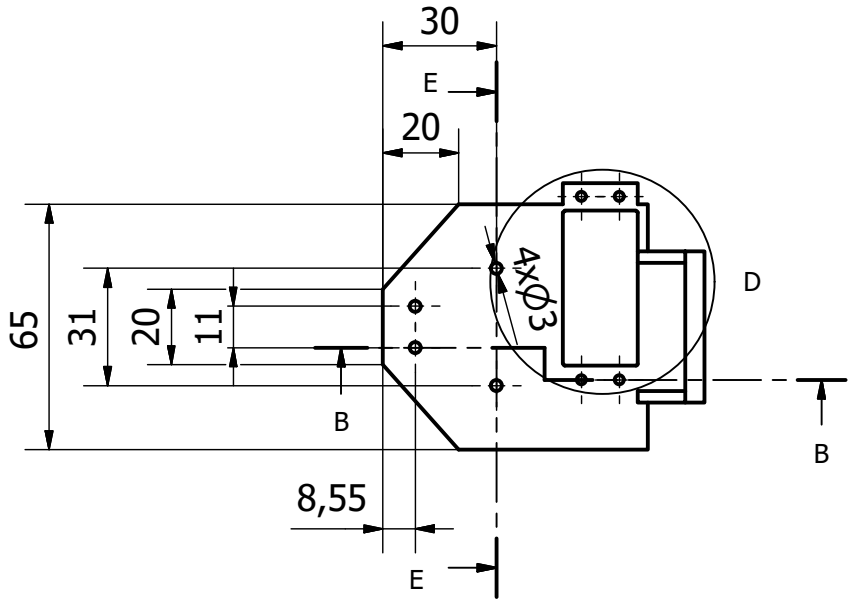
D (1 : 1)



E-E



B-B



TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Autor: Iris Vila Castellá

Proyecto:

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

Fecha: 28.08.18

Escala: 1:2

Plano: Pinza - base

Nº Plano: 11/13

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

ANEXO I: PROGRAMACIÓN

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

ÍNDICE ANEXO I: PROGRAMACIÓN

ÍNDICE DE FIGURAS ANEXO I: PROGRAMACIÓN

1-Estructura Sketch Arduino

Para comentar el código que se ha empleado para programar la placa se separa en diferentes partes.

Para empezar, se han declarado variables que se emplea más tarde para la recogida de datos o para el movimiento del brazo.

```
1  //Librerias necesarias
2  #include <SoftwareSerial.h>
3  #include <Servo.h>
4
5  //Variables a usar
6  int num;
7  int servo;
8  int ang;
9
10 //Declaración servomotores
11 Servo miservo1;
12 Servo miservo2;
13 Servo miservo3;
14 Servo miservo4;
15 Servo miservo5;
16
```

Figura 1. Código Arduino (Parte I)

A continuación, en la sección “void setup ()” se inicia la conexión Bluetooth con el dispositivo móvil y se enlazan los servos con el pin digital al que se encuentren conectados

```
17 void setup() {
18   // put your setup code here, to run once:
19
20
21   Serial.begin(9600);
22
23   //Enlazamos los servomotores con el pin digital al que se encuentran conectado
24   miservo1.attach(4);
25   miservo2.attach(5);
26   miservo3.attach(6);
27   miservo4.attach(7);
28   miservo5.attach(8);
29
30
31 }
```

Figura 2. Código Arduino (Parte II)

Por último, en la sección “void loop ()” se realiza la lectura de datos del bluetooth mediante la función “Serial.read ()” en el caso en el que hayan datos disponibles. Para comprobar si hay datos que leer se hace uso de la función “Serial.available ()” y, si se encuentra disponible, se recibirá como respuesta un “true” por lo que la condición del “if ()” se cumplirá y se realizará la

lectura de datos. Como se ha comentado anteriormente, los datos se han convertido para poder pasarse en un espacio de 255 valores así que, para volver a tener el número de grados que se han enviado, se realiza la conversión de nuevo.

```
36 void loop() {
37   // put your main code here, to run repeatedly:
38       if (Serial.available())
39 {
40
41   int valor=(Serial.read());
42   delay(1);
43   num=valor*(904.0/255.0);
44
```

Figura 3. Código Arduino (Parte III)

Se emplea el uso de la condición “if ()” imponiendo ente sus paréntesis el numero entre el que se debe encontrar el dato para mover una u otra articulación. Para escribir el ángulo deseado en el servomotor elegido se emplea la función “write ()”, introduciendo entre los paréntesis el ángulo de movimiento. Para poder poner el número de grados que se desplaza cada servomotor realmente se tiene que restar el número que se le ha sumado a cada intervalo y, de esta manera, se tiene un ángulo entre 0 y 180 grados.

```
44
45   if(num>=0&&num<=180){
46     miservo1.write(num);
47     delay(1);
48   }
49   else if (num>=181&&num<=361){
50       ang=num-181;
51     delay(1);
52   miservo2.write(ang);
53     delay(1);
54   }
55   else if (num>=362&&num<=542){
56     ang=num-362;
57     delay(1);
58     miservo3.write(ang);
59     delay(1);
60   }
61   else if (num>=543&&num<723){
62     ang=num-543;
63     delay(1);
64     miservo4.write(ang);
65     delay(1);
66   }
67   else if (num>=724&&num<=904){
68     ang=num-724;
69     delay(10);
70     miservo5.write(ang);
71     delay(10);
72   }
73
74 }    }
75
```

Figura 4. Código Arduino (Parte IV)

2-Programación Android Studio

2.1- Programación conexión bluetooth

En este apartado se comenta el código empleado para realizar la conexión bluetooth con la placa Arduino Nano.

En primer lugar, para poder usar las funciones de Bluetooth que nos ofrece Android Studio se necesita dar los permisos necesarios en el "AndroidManifest.XML". Para ello, se abre el XML que se encuentra dentro de la carpeta manifests y se introduce el siguiente código:

```
<uses-permission android:name="android.permission.BLUETOOTH" />

<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Mediante el permiso de bluetooth se puede solicitar, aceptar y transferir datos, Además, se ha incluido el permiso de "BLUETOOTH_ADMIN" para poder detectar dispositivos y controlar los diferentes ajustes de Bluetooth.

A tal efecto, se emplean las siguientes clases e interfaces para crear la conexión bluetooth:

-BluetoothAdapter: Se utiliza como un adaptador del dispositivo local. Mediante esta clase se puede buscar dispositivos (tanto sincronizados como disponibles en la zona local), configurar el bluetooth, transferir datos o conectar con otros dispositivos.

Además, se usan diferentes funciones con esta clase para poder generar el código necesario para la conexión con otro dispositivo

-getDefaultAdapter (): Mediante esta función se obtiene el adaptador bluetooth o, se indica la inexistencia de adaptador bluetooth en el dispositivo, en caso de recibir como respuesta "null" (nulo).

-isEnabled (): Se recibe un boolean como respuesta a dicha función. En el caso en el que el Bluetooth se encuentre habilitado y preparado para su uso, se obtendrá "true" (verdadero). Por otro lado, si la respuesta es "false" (falso), se iniciará una actividad denominada "BluetoothAdapter.ACTION_REQUEST_ENABLE" en la cual se solicitará al usuario la conexión del bluetooth del dispositivo. La actividad citada finalizara cuando el usuario rechace la solicitud o cuando termine de encenderse.

-BluetoothDevice: Esta clase representa un dispositivo remoto de Bluetooth. Permite crear una conexión con el dispositivo o modificar información sobre él (nombre, dirección, clase y estado de vinculación). Las operaciones que se realicen se harán mediante el "BluetoothAdapter" que se empleó para la creación de dicho "BluetoothDevice".

-getBondedDevices (): Dicha función nos retornará el conjunto de dispositivos bluetooth que ya estén emparejados al bluetooth local y, en caso de no encontrarse conectado el bluetooth, nos devolverá un conjunto vacío.

Una vez recibidos los dispositivos emparejados, se emplean las siguientes funciones que nos permitirán mostrar dichos dispositivos, seleccionar el elegido, enlazar la placa Arduino con el dispositivo y empezar la actividad del movimiento del brazo robot:

-ListView: Se trata de una clase que plasma una lista de elementos desplazables. Los elementos se incluyen de forma automática con un "Adapter" que coge el contenido de una matriz o base de datos. Se emplean algunas funciones de "ListView":

-setAdapter (ListAdapter): Esta función nos permite mostrar los elementos en la lista, asociando un adaptador con ella.

-setOnItemClickListener: El uso de este método se realiza cuando el usuario selecciona o toca un elemento del ListView creado. Una vez selecciona el Bluetooth con el que desea emparejarlo se inicia la conexión y, si ha resultado exitosa, empieza la actividad para mover el brazo robot.

-ArrayList: Dicha clase implementa la intefaz de "List" y proporciona métodos para poder modificar el tamaño del matriz empleado para almacenar la lista. Podría compararse con un vector con capacidad de almacenaje ya que mediante la función ".add ()" se añade un tamaño al arraylist con el nombre del dispositivo bluetooth y su dirección.

-ArrayAdapter: Esta clase permite la introducción de datos desde el Array hasta el ListView. El constructor que se declara en el código utilizaos tres parámetros:

-this: Indica el contexto en el que se sitúa, es decir, el entorno actual de la aplicación.

-android.R.layout.*simple_list_item_1*: Indica la vista que será empleada para formar la lista.

-list: Señala el vector de "String" que contiene los textos que serán visualizados en cada "TextView". Como se ha definido anteriormente "list" es un "Arraylist" donde se ha acumulado los dispositivos Bluetooth.

Una vez seleccionado el dispositivo Bluetooth y realizada la conexión, mediante un "intent" se pasa de la actividad "Lista_de_dispositivos" a la actividad "servoControl", donde se realiza el movimiento del brazo robot enviando los datos desde el dispositivo móvil al módulo HC-06 de Arduino.

A continuación, se comenta el código empleado para el envío de datos en la actividad "servoControl". En ella, se ha hecho uso de las siguientes actividades para conseguir el objetivo:

-BluetoothSocket: Dicha clase representa la interfaz de un socket de Bluetooth. Un socket es un punto de conexión que nos permite el envío de información desde el dispositivo a través de getOutputStream. Dentro de esta clase, se emplea diversas funciones que nos ayudan al conexionado:

-connect (): Con ella se intenta realizar la conexión con el dispositivo remoto y, hasta que se realice la conexión o falle, la llamada queda bloqueada.

En esta parte de la actividad se emplean otras funciones que nos ayudaran a cumplir el objetivo del envío de datos:

-getRemoveDevice (adress): Esta función se aplica al "BluetoothDevice" creado y se emplea para obtener el dispositivo de Bluetooth mediante la dirección MAC ya conocida. La dirección MAC (Media Acces Control) es un identificador que el fabricante de un dispositivo hardware le asigna a dicho dispositivo.

-createInsecureRfcommSocketToServiceRecord (myUUID): Crea un RFCOMM "BluetoothSocket" preparado para una conexión saliente segura al dispositivo ubicado en el área local empleando la búsqueda SDP de UUID.

-onPreExecute (): Este método se crea para proyectar un diálogo antes de establecer la conexión y nos indicará que se está realizando la conexión.

-onPostExecute (): Dicho método nos mostrará dos textos dependiendo de si se ha realizado la conexión con éxito o no. En el caso en que no se consiga realizar, nos mostrará un diálogo preguntando si realmente es una dirección Bluetooth. Por otro lado, si se ha realiza con éxito, el diálogo nos indicará que se ha conectado.

```
package com.servo.servo;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ListView;

import java.io.OutputStream;
import java.util.Set;
import java.util.ArrayList;
import android.widget.Toast;
import android.widget.ArrayAdapter;
import android.widget.AdapterView;
import android.widget.TextView;
import android.content.Intent;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;

public class Lista_de_dispositivos extends AppCompatActivity {

    Button btnPaired;
    ListView devicelist;
    private BluetoothAdapter myBluetooth = null;
    private Set<BluetoothDevice> pairedDevices;
    private OutputStream outStream=null;
    public static String EXTRA_ADRESS="device_address";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lista_de_dispositivos);

        btnPaired = (Button)findViewById(R.id.button);
        devicelist = (ListView)findViewById(R.id.listView);

        myBluetooth = BluetoothAdapter.getDefaultAdapter();
        if(myBluetooth == null) {

            Toast.makeText(getApplicationContext(), "Dispositivo
Bluetooth no disponible", Toast.LENGTH_LONG).show();
            finish(); }
            else {
            if (myBluetooth.isEnabled()) {

            } else {
Intent turnBTon = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(turnBTon,1);
            }
        }
        btnPaired.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                pairedDevicesList();                } });



            }
    private void pairedDevicesList(){
            pairedDevices = myBluetooth.getBondedDevices();
            ArrayList list = new ArrayList();
    if (pairedDevices.size()>0) {
        for(BluetoothDevice bt : pairedDevices) {
            list.add(bt.getName() + "\n" + bt.getAddress());
            } } else { Toast.makeText(getApplicationContext(), "No se
encuentran dispositivos emparejados", Toast.LENGTH_LONG).show();
    }
    final ArrayAdapter adapter = new
ArrayAdapter(this,android.R.layout.simple_list_item_1, list);
    devicelist.setAdapter(adapter);
    devicelist.setOnItemClickListener(myListClickListener);

        }
    private AdapterView.OnItemClickListener myListClickListener = new
AdapterView.OnItemClickListener() {
        public void onItemClick (AdapterView av, View v, int arg2,
long arg3) {

            String info = ((TextView) v).getText().toString();
            String address = info.substring(info.length() - 17);
            Intent i = new Intent(Lista_de_dispositivos.this,
servoControl.class);
            Bundle mi_bundle =new Bundle();
            mi_bundle.putString("EXTRA_ADDRESS",address);
            i.putExtras(mi_bundle);
            startActivity(i); } };
        }
```

2.2-Programacion movimiento brazo

Para conseguir un correcto funcionamiento del brazo robot se ha decidido emplear los siguientes controles:

-Seekbar: Se trata de una barra de progresión que permite el movimiento de sí misma mediante la pulsación táctil. El usuario puede tocar la "Seekbar" del servomotor que desee mover y arrastrarlo hacia la izquierda o derecha para incrementar o disminuir el ángulo del brazo robot. Se utiliza la clase anidada "Seekbar.OnSeekBarChangeListener" para notificar al usuario cuando se ha cambiado el nivel de progreso. Además para modificar las propiedades de las barras de progresión se emplean las funciones "setMax ()" y "setProgress ()", en las cuales se indica entre los paréntesis el máximo al que tiene que llegar la barra de progresión y el número por el cual empieza.

-Textview: Se emplea este control de texto para indicar que barra de progresión está ligada con que servomotor y plasmar los grados que se está moviendo cada servomotor, para lo cual se ha empleado la función "setText ()" que nos permite cambiar el texto que contiene el "TextView".

-Button: Mediante el uso de los botones se ofrece al usuario la posibilidad de desconexión de la red Bluetooth a la que se encuentra conectada en ese mismo instante. Para poder detectar el instante en el que el usuario aprieta el botón se emplea el método "setOnClickListener ()", que se activa cuando el usuario presiona el botón indicado.

Para poder garantizar la diferencia de movimiento entre las diferentes articulaciones al enviar los datos a Arduino se decide realizar el envío de la siguiente manera:

-Si se modifica los grados del servomotor de la base se enviaran los datos de 0 a 180 grados.

-Si se modifican los grados del servomotor del hombro se enviaran los datos de 181 a 361

-Si se modifican los grados del servomotor del codo se enviaran los datos de 362 a 542.

--Si se modifican los grados del servomotor de la muñeca se enviaran los datos de 543 a 723.

-Si se modifican los grados del servomotor de la apertura o cierre de la pinza se enviaran los datos de 724 a 904.

Como se puede observar, para poder diferenciar que servomotor hay que mover y cuantos grados según lo que varíe el usuario, se realizan cinco intervalos de 180 números consecutivos y, de esta manera, al leer el dato desde Arduino, se sabrá que articulación mover.

Por otro lado, en el envío de datos solo pueden enviarse hasta palabras de 8 bytes, es decir un espacio de 2^8 (256) bits. Como el envío se realiza en bytes, se ha decidido hacer una conversión del dato a enviar para que Arduino pueda leerlo correctamente, ya que si no, solo leería hasta el número 255.

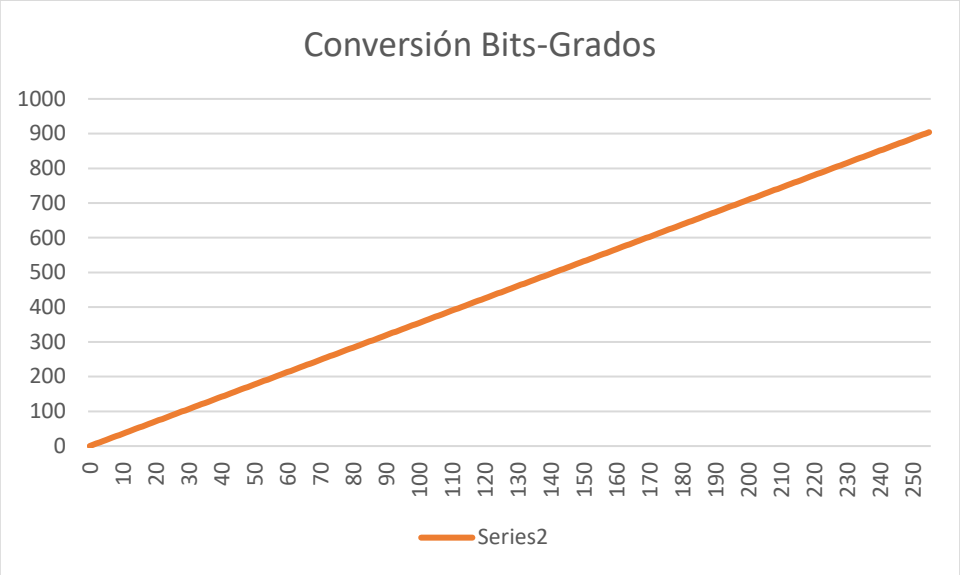


Figura 5. Relación entre los grados enviados y los bit recibidos

```
servo1.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {

        public void onProgressChanged(SeekBar servo1, int progress,
boolean a) {
            if (a == true) {
                ang.setText(String.valueOf(progress));
                valor = progress;
                Log.d("DEBUG", "Clicas servo1 " +valor);
                try {

btSocket.getOutputStream().write((int) (valor*(255.0/904.0)));

                } catch (IOException e) {

                }
            }
        }


        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });

    servo2.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress,
boolean fromUser) {
            if (fromUser == true) {
```

```
                ang2.setText(String.valueOf(progress));
                valor = progress;
                Log.d("DEBUG", "Clicas servo2 " + valor);
                try {


  btSocket.getOutputStream().write((int) ((valor+181)*(255.0/904.0)));
                } catch (IOException e) {

                }
            }
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
    servo3.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {

        public void onProgressChanged(SeekBar servo1, int progress,
boolean a) {
            if (a == true) {
                ang3.setText(String.valueOf(progress));
                valor = progress;
                Log.d("DEBUG", "Clicas servo1 " +valor);
                try {

  btSocket.getOutputStream().write((int) ((valor+362)*(255.0/904.0)));

                } catch (IOException e) {

                }
            }
        }


        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
    servo4.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {

        public void onProgressChanged(SeekBar servo1, int progress,
boolean a) {
            if (a == true) {
                ang4.setText(String.valueOf(progress);
```

```
                valor = progress;
                Log.d("DEBUG", "Clicas servo1 " +valor);
                try {

 btSocket.getOutputStream().write((int) ((valor+543)*(255.0/904.0)));

                } catch (IOException e) {

                }
            }
        }


        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
    servo5.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {

        public void onProgressChanged(SeekBar servo1, int progress,
boolean a) {
            if (a == true) {
                ang5.setText(String.valueOf(progress);
                valor = progress;
                Log.d("DEBUG", "Clicas servo5 " +valor);
                try {

 btSocket.getOutputStream().write((int) ((valor+724)*(255.0/904.0)));

                } catch (IOException e) {

                }
            }
        }
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
}
```

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

ANEXO II: INTERFAZ DE USUARIO

Diseño y desarrollo del prototipo de un brazo robótico articulado antropomórfico controlado mediante un microcontrolador Arduino Mega 2560

ÍNDICE ANEXO II: INTERFAZ DE USUARIO

ÍNDICE DE FIGURAS ANEXO II: INTERFAZ DE USUARIO

1-Pantalla de selección de dispositivos

Esta pantalla es la que observa el usuario en el instante en el que abre la aplicación. En el caso en el que el usuario no tenga activado el Bluetooth de su dispositivo apararecerá un diálogo solicitándole la activación de este.

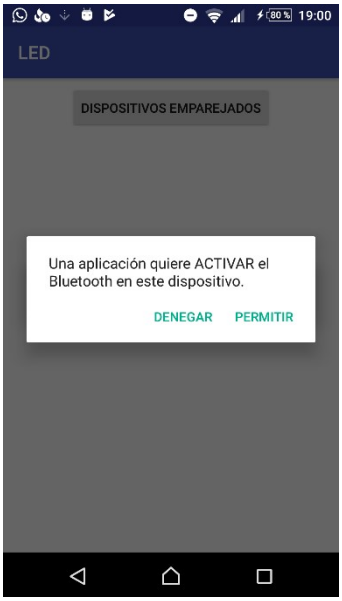


Figura 1- Dialogo solicitud Bluetooth

Si el usuario permite que se encienda el bluetooth aparecerá un cartel mientras se estÁ produciendo la activación que desaparecerá cuando ya se haya iniciado, dejando ver la pantalla de selección de dispositivos.

Si en cambio el usuario deniega esta conexión, el cuadro de dialogo desaparecerá y se mostrará directamente la pantalla de selección.

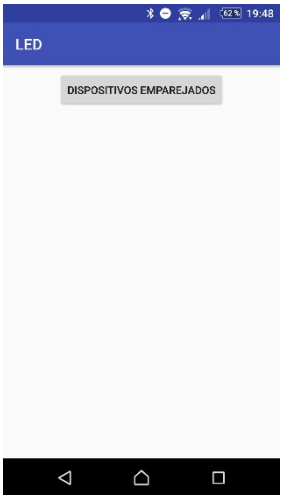


Figura 2- Pantalla Lista de Dispositivos

En ella, se encuentra un botón llamado “Dispositivos emparejados” que nos permite mostrar aquellos dispositivos que ya han sido vinculados con el nuestro. En el caso en el que no haya ningún dispositivo emparejado, la aplicación muestra un cuadro de diálogo temporal que nos indica que no se han encontrado dispositivos vinculados

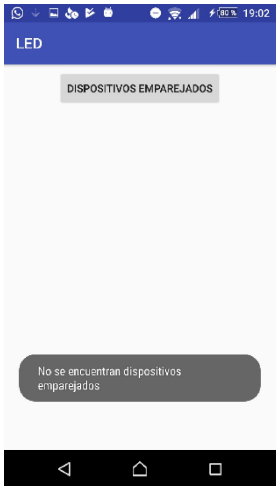


Figura 3-Pantalla que indica ausencia de dispositivos emparejados

En el momento de la primera conexión con la placa arduino, el usuario tendrá que ir a los ajustes de bluetooth y vincular su dispositivo con el módulo Bluetooth de la placa. Por defecto, el pin para conectar ambos dispositivos es “0000”.

Una vez ya se han emparejado los dispositivos, el usuario podrá regresar a la aplicación y cuando pulse el botón “dispositivos emparejados” aparecerá el nombre del bluetooth al que se ha vinculado. Si el usuario encuentra varios dispositivos, podrá elegir aquel con el que quiere vincularse. Para emparejarse con el módulo Arduino HC-06 de la placa, deberá seleccionar aquel con nombre “BlueSlave”.

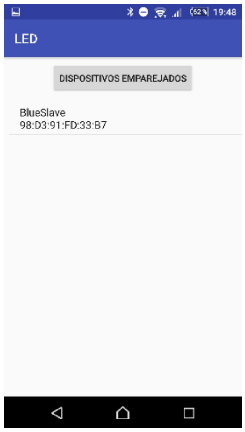


Figura 4- Pantalla con dispositivo vinculado

A continuación, cuando el usuario seleccione un dispositivo se iniciará la conexión, añadiendo un cuadro de diálogo que notifica esta operación.

Figura 5-Cuadro de diálogo de conexión

Si el usuario selecciona un dispositivo con el que no se puede emparejar, se mostrará por pantalla un cuadro de dialogo temporal que notificará del error de conexión.

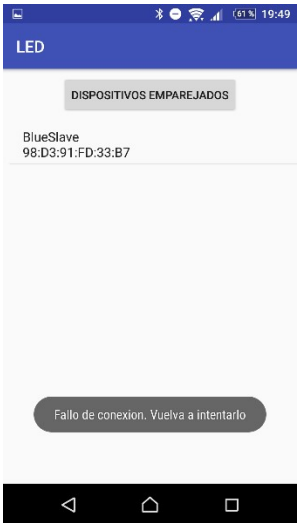


Figura 6- Pantalla con error de conexión

Cuando se realice con éxito la conexión, se mostrará la otra pantalla de la aplicación “Movimiento robot”.

2-Pantalla de movimiento robot

Como se puede observar está pantalla consta de todos los objetos necesarios para poder mover los servomotores del brazo robot o desconectar la conexión con el dispositivo.

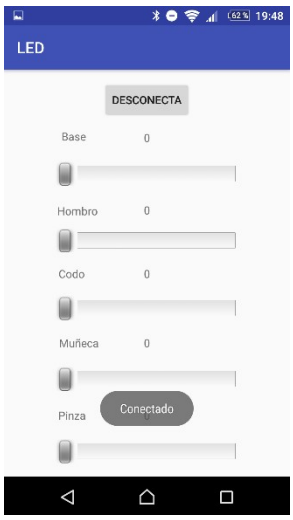


Figura 7- Pantalla movimiento servomotores con aviso inicial

Mediante la pulsación del botón "Desconecta", se retornará a la pantalla anterior y se desconectará el dispositivo del módulo bluetooth HC-06.

Por otro lado, el usuario encontrará cinco barras de progresión que indican el grado de movimiento de cada grado de libertad del robot. Además, cada barra tiene un texto que le indica al usuario que articulación estaría desplazándose.

Por último, el usuario podrá ver los grados que se ha desplazado la articulación en un texto situado encima de cada barra de progresión.

Figura 8- Pantalla movimiento servomotores