



UNIVERSIDAD POLITÉCNICA DE VALENCIA



TRABAJO FIN DE GRADO

Diseño de un manipulador controlado con el microcontrolador Arduino.

Grado en Ingeniería Mecánica.

Autor: **Octavio Amat Verdú**

Tutor: **Roberto Capilla Lladró**





Agradecimientos.

A mi padre y mi madre por el apoyo y sus ánimos a continuar estudiando, a seguir peleando y no tirar la toalla.

A mis compañeros de piso Pablo, Francisco y Patricio, por estar conmigo a las duras y a las maduras, y esos bocadillos tan épicos que nos hacíamos.

Cabe mencionar a Roberto, mi tutor de prácticas y de este proyecto, por los consejos proporcionados, la confianza en mí y, sobre todo, saber transmitir esa pasión por la robótica y la electrónica.

Por último, pero no menos importante, a Tatiana por soportar mis periodos de estrés y por el apoyo día tras día que me ha mantenido firme.



Resumen

En el presente proyecto titulado *Diseño de un manipulador controlado con el microcontrolador Arduino*, se muestra la impresión, el ensamblaje y la creación de un prototipo constituido por dos brazos. Uno de ellos será movido por el usuario y el otro, que denominaremos brazo robot o manipulador, imitará dicho movimiento. Los planos del brazo robot, se han obtenido de una página de internet OpenSource que proporciona modelos 3D con licencia libre para el uso de maquetas y prototipos. El brazo que gobierna al manipulador ha sido diseñado para este proyecto utilizando el programa *Autodesk Inventor Pro 2017* y para el control del sistema se ha utilizado una plataforma *Arduino* para la programación del modelo.

Este trabajo final de grado ha sido realizado por el alumno Octavio Amat Verdú, estudiante del Grado en Ingeniería Mecánica en el 5º año de matriculación, tutelado por Don Roberto Capilla Lladró perteneciente al Departamento de Ingeniería Electrónica (DIE), en Valencia a 15 de Septiembre de 2017.



Abstract

In this project named *Design of a manipulator controlled by microcontroller Arduino*, it is shown the printing, assembly and creation of a two arm model prototype. One of them named robotic arm, will be moved by the use, while the other one called robotic manipulator will imitate that movement. The manipulator's sketches are obtained via an OpenSource web that provides 3D models with open license with the porpoise of investigation and prototypes. The leading arm will be designed using *Autodesk Inventor Pro 2017*. Concerning the control system, we are using an OpenSource platform named *Arduino* that allows us to program an application.

This final project degree has been made by Octavio Amat Verdú, Mechanical Engineering student in its 5th career year, with the supervision of Mr Roberto Capilla Lladró belonging the Electronic Engineering Department (DIE), at Valencia on September the 15th.



Índice

1. Introducción y objetivos.	7
2. Antecedentes.	8
2.1 Introducción a la Robótica.	8
2.2 Brazos robóticos, tipos de configuraciones y aplicaciones.	9
2.3 Impresoras 3D.	12
2.4 Microcontrolador y microprocesador.	20
2.5 Arduino.	23
2.6 Thingiverse.	24
2.7 Circuitos eléctricos.	25
2.8 Potenciómetros.	28
2.9 Servomotores.	30
3. Ejecución del proyecto.	32
3.1 Aspectos a tener en cuenta.	32
3.2 Montaje del manipulador y del brazo gobernante.	33
3.2.1 Manipulador gobernado por servomotores.	33
3.2.1.1 Unión entre los módulos.	34
3.2.1.2 Módulo Base.	34
3.2.1.3 Módulo Brazo.	35
3.2.1.4 Módulo Antebrazo.	36
3.2.1.5 Módulo Muñeca.	36
3.2.1.6 Módulo Agarre.	37
3.2.1.7 Módulos ensamblados.	38
3.2.2 Diseño del brazo que controla los servos.	38
3.2.2.1 Unión entre piezas.	39
3.2.2.2 Base.	39
3.2.2.3 Base giratoria.	40
3.2.2.4 Brazo y antebrazo.	40
3.2.2.5 Muñeca y agarre.	41
3.2.2.6 Diseño final del brazo.	42
3.3 Programación del sistema de control de los servos y montaje del circuito electrónico.	42
3.3.1.1 Aplicación de control.	43



3.3.1.2	Declaración de variables.	43
3.3.1.3	Función Setup.....	44
3.3.1.4	Función Loop.....	44
3.4	Circuito electrónico.....	46
3.5	Resultado final.....	47
3.6	Posibles mejoras del proyecto.	48
4.	Planos.....	49
5.	Presupuesto.....	55
6.	Referencias.	57
7.	Anexos.....	58



1. Introducción y objetivos.

Hoy en día la tecnología se encuentra prácticamente en todos los ámbitos de nuestras vidas, ya sea dentro de nuestras casas, como fuera. El simple hecho de poner una lavadora conlleva detrás un largo estudio, cálculos y prototipos, para conseguir el producto final.

Y entonces, ¿qué es la tecnología? Podría decirse que es el conjunto de conocimientos y técnicas cuyo objetivo es desarrollar un entorno agradable y cómodo para el ser humano, ya sea facilitando el transporte entre ciudades o ayudarnos en la manufactura de productos, entre otras funciones. Debido al gran uso de dichas técnicas y conocimientos, la tecnología es un ámbito que está en constante desarrollo, y consecuentemente la industria también.

Actualmente, la mayoría de las industrias se benefician de lo que se llama “Producción en masa”, “Fabricación en serie” o “Cadena de montaje”. Este concepto se basa en delegar a cada operario una función específica a lo largo de la fabricación del producto, es decir, colocar cierto componente, conectar un cable, atornillar, etc. A lo largo de toda la historia de desarrollo de la industria, se han ido implementando progresivamente máquinas en las áreas de trabajo para ayudar a las personas a construir el producto, asegurando una mejor calidad y una mayor producción.

Más en detalle, me centraré en la aparición de los brazos robóticos, brazo mecánico o manipulador mecánico. Este tipo de máquina es un símil al de un brazo humano, compuesto por una base, un antebrazo, brazo y pinza o agarradera. El movimiento de éste se realiza gracias a que se incluyen unos motores que actúan sobre las articulaciones, proporcionándoles la fuerza necesaria.

El interés de la realización del montaje de este tipo de brazo mecánico para el Trabajo de Fin de Grado (TFG) surgió tras finalizar el periodo de prácticas en una empresa dedicada a la industria del automóvil, ya que me asombró la cantidad de máquinas y automatismos que utilizaban para muchos de los procesos de construcción y ensamblaje. Especialmente me llamó la atención los inmensos brazos robóticos que formaban parte de la cadena de montaje.

Por tanto, el objetivo de este proyecto es el de construir dos tipos de brazos: uno será el que nosotros manipulamos y movemos, y otro el que replique los movimientos que nosotros provocamos. Éste primer brazo constará de cinco juntas móviles gobernadas por un servomotor en cada una. El segundo brazo tendrá un diseño similar el primero, solo que en vez de motores llevará potenciómetros conectados al procesador Arduino los cuales controlarán la posición de dichos servos. La estructura del brazo estará diseñada mediante planos CAD, construida por impresoras 3D y la programación de los servomotores se llevará a cabo con la plataforma electrónica *Arduino*.

2. Antecedentes.

En el siguiente apartado se detallarán los puntos más importantes que se utilizarán a la hora del diseño y programación del manipulador, así como el equipo usado.

2.1 Introducción a la Robótica.

Por mucho tiempo, el ser humano ha tratado de recrear mediante “artefactos” o máquinas que imiten las partes del cuerpo, ya sea para delegar el trabajo tedioso y costoso en ellos o simplemente como mero hobby enfocado al ocio y entretenimiento. Durante los siglos XVII y XVIII en Europa se construyeron diversos muñecos que se asemejaban al cuerpo humano. Jacques de Vaucanson (un ingeniero e inventor francés) construyó varios músicos del tamaño de un humano, que se trataban de robots mecánicos cuyo propósito era entretener.

A principios del año 1800, un ingeniero mecánico suizo llamado Henri Maillardet, creó un autómatas, o robot mecánico, que consistía en una muñeca que dibujaba. Una serie de levas se utilizaban para programar el movimiento y mediante un mecanismo de muelles y resortes ésta muñeca podría escribir frases o dibujar.



Imagen 1. Muñeca escritora.

Todos estos autómatas, inventos, máquinas, ingenieros e inventores de las épocas pasadas, y gracias a la aparición de la electricidad y a los avances en tecnología que hoy en día se están consiguiendo, nace esta ciencia o disciplina llamada Robótica. Esta ciencia, o rama de la ciencia, se encarga del desarrollo de los robots y de sus aplicaciones. Otra de sus definiciones podría ser la fabricación o utilización de máquinas automáticas programables, con el fin de realizar tareas repetitivas. (Existen muchos trabajos que a las personas no les gusta hacer, o simplemente les sería más cómodo si lo hiciese

otra persona.)

La industria es un claro ejemplo de cómo los robots facilitan el trabajo, es un ámbito donde abundan los movimientos repetitivos y en el que la precisión y la calidad son cruciales para proporcionar un producto competente. Las personas se equivocan, se fatigan y necesitan descansos para poder continuar trabajando, los robots no. Otra de las ventajas de los robots es que se pueden construir de casi cualquier tipo y tamaño, es decir, podemos construir un manipulador más grande, más fuerte y ágil para poder manipular o transportar materiales dañinos para el hombre, o levantar pesos que haría falta de varias personas para poder lograrlo. Una industria que se beneficiaría de éstas ventajas sería aquella que se basase en la producción en masa o cadena de montaje, como por ejemplo la del automóvil. Estas fábricas producen miles de coches al día y como en cada puesto se repiten las mismas operaciones una y otra vez a lo largo del día la aplicación de robots es muy habitual.

En resumen, la robótica se ocupa de todo lo que concierne a los robots, esto implica programación, control de motores, mecanismos automáticos, neumática, sensores, etc. Es por ello que combina diferentes disciplinas como la ingeniería mecánica, eléctrica, mecatrónica,

electrónica o biomédica (entre otras). Como toda disciplina cuenta con sus problemas, sus ventajas y leyes. Se puede dividir en dos vertientes: teoría y práctica. La parte teórica reúne las aportaciones de la automática, informática y la inteligencia artificial. Por otro lado, en la práctica hay aspectos de construcción (mecánica y electrónica) y de la gestión (control y programación). La robótica cuenta con tres leyes o principios según Isaac Asimov (un escritor y profesor biomédico del siglo XX), estos son:

- Un robot no puede lastimar ni permitir que sea lastimado ningún ser humano.
- El robot debe obedecer todas las órdenes de los humanos, excepto las que contradigan la primera ley.
- El robot debe auto protegerse, salvo que para hacerlo entre en conflicto con la primera o segunda ley.

Dentro del ámbito de la robótica, podemos encontrarnos innumerables tipos de robots, a continuación se mencionan los que se han escogido como más comunes:

- **Robots industriales:** Este conjunto lo forman manipuladores automáticos, reprogramables y multifuncionales que suelen tener, como mínimo, 3 ejes. Son utilizados por las industrias (como su propio nombre indica) para orientar piezas, recolocar materiales o ejecutar trabajos diversos en las distintas etapas de la cadena de producción. Suelen ser de posición fija o en movimiento sobre raíles.
- **Robots de servicio:** Esta variedad engloba a aquellos dispositivos electromecánicos (habitualmente móviles) controlados por ordenador que realizan tareas no industriales. Estos robots se utilizan para reemplazar al hombre en entornos no estructurados y que puedan requerir de un cambio en el mismo. Un ejemplo claro de este tipo de máquinas serían los robots aspirador (imagen 2).



Imagen 2. Robot aspirador.

- **Robots de investigación:** Podemos encontrar una gran variedad de este tipo de robots en los laboratorios o talleres de investigación de muchas Universidades, no suelen tener una aplicación concreta mientras que en otras ocasiones son prototipos de determinados proyectos.
- **Juguetes robóticos:** Poseen una gran variedad en el mercado, y que últimamente está en auge aumentando la cantidad y tipos de robots que van apareciendo, enfocados principalmente al ocio. Podemos encontrar desde mascotas y dinosaurios, hasta drones y helicópteros voladores.

2.2 Brazos robóticos, tipos de configuraciones y aplicaciones.

En la clasificación de los robots, podemos incluir el caso concreto de éste trabajo en el grupo de los robots de investigación, lo que significa que estamos ante un prototipo de lo que podría ser un brazo robótico industrial. Ahora bien, ¿qué es un brazo robótico?

Los brazos robóticos o manipuladores robóticos, son un tipo de máquinas mecánicas, normalmente programables, que poseen funciones parecidas a las de un brazo humano. Las diferentes partes móviles de éste brazo están conectadas mediante articulaciones que permiten un movimiento rotacional o un desplazamiento lineal. Estos manipuladores, en sus extremos, poseen lo que se llama el efector final, o mano robótica, en la que se encuentra la herramienta necesaria para el proceso que esté realizando, ya sea una soldadura o colocar ciertas partes del producto. En situaciones especiales pueden poseer una réplica de una mano humana para desempeñar dicha función como podría ser la desactivación de explosivos.

A continuación nombraremos rápidamente las características principales de los brazos robóticos:

- **Grados de libertad:** Se conoce como grados de libertad (GDL) al número mínimo de variables necesarios para describir la distribución de velocidades en el sistema, es decir, cuáles de las 4 variables de posición en el espacio (X, Y, Z, α) no están restringidas.
- **Volumen de trabajo:** Según los propios fabricantes de robots, es el espacio dentro del cual puede desplazarse el extremo de su muñeca. Este espacio viene definido por el tamaño de sus eslabones. Por ejemplo, los robots antropomórficos (similares al mostrado en la “Imagen 9”) poseen un volumen parecido al de una esfera.

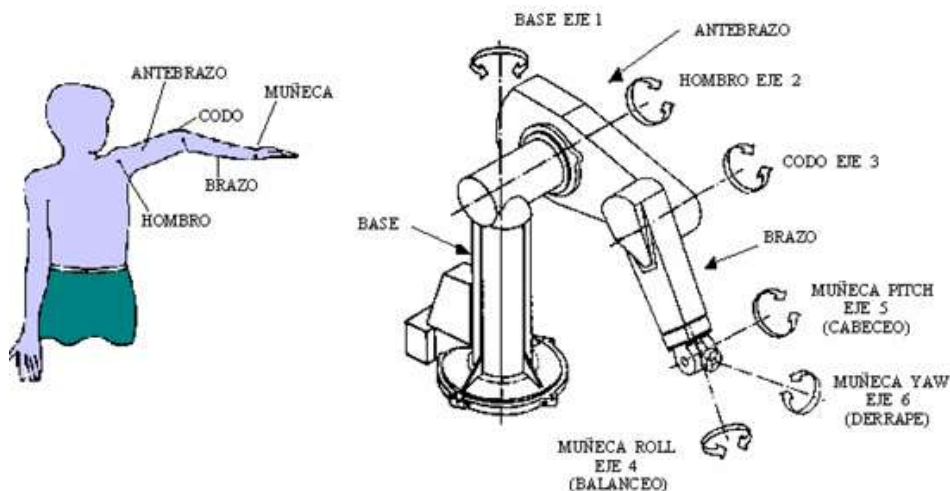


Imagen 3. Esquema de un brazo robótico de 6 GDL.

Dentro de los brazos robóticos podemos encontrar las siguientes configuraciones:

- **Robot cartesiano:** Se trata de un robot cuyo brazo tiene tres ejes principales que son lineales (se mueven en línea recta) y forman un ángulo recto entre ellos, asemejándose a los ejes cartesianos (X, Y, Z). Poseen altas prestaciones en cuanto a velocidad, para desempeñar tareas en zonas de trabajo estructuradas. Es por ello que para sus aplicaciones prácticas necesitan entornos que no sean complejos (pocos obstáculos). Destaca a su vez por la sencillez en la construcción, siendo idóneos a la hora de la alimentación de máquinas. Otra aplicación muy común de estos brazos es la acción de cortar sobre un plano X,Y. Un claro ejemplo es el corte de formas sobre una plancha metálica o en la industria del calzado para obtener las formas del tejido deseadas.

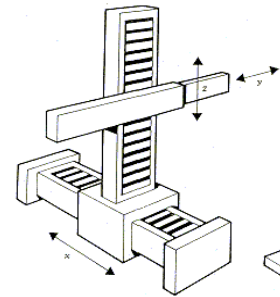


Imagen 4. Robot cartesiano.

- **Robot cilíndrico:** Este tipo de brazos mecánicos se caracterizan por tener un movimiento rotacional en la base y dos ejes lineales entre sí perpendiculares, siendo el segundo de ellos perpendicular a la base. En su mayoría poseen una base rotativa, capaz de deslizarse hacia arriba o hacia abajo y carga con un segmento horizontal telescópico.

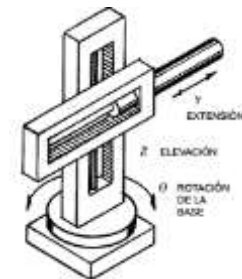


Imagen 5. Robot cilíndrico.

- **Robot esférico/polar:** Se compone de una base rotativa y sobre ella colocado perpendicularmente a la base, unido mediante una junta que permite el movimiento angular, a una tercera articulación que permite el movimiento lineal. El área de trabajo de esta configuración es una esfera hueca ya que no son capaces de desplazarse dentro del volumen que sus articulaciones describen. Estos brazos son muy versátiles debido a su área de trabajo, podemos encontrar este tipo de brazos en aplicaciones como cargar maquinaria para mantenimiento de ellas, o soldadura en líneas de fabricación como pueden ser las de automóviles.

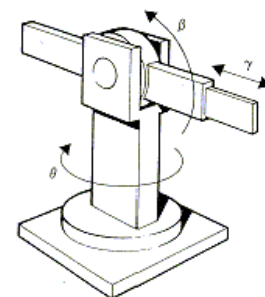


Imagen 6. Robot esférico.

- **Robot SCARA:** Los robots SCARA (Selective Compliant Assembly Robot Arm) tienen una configuración similar a la cilíndrica, solo que el radio y la rotación nos la proporcionan unos eslabones (similar a la unión de una cadena de bicicleta), en su mayoría suelen tener entre 3 y 4 grados de libertad. Son conocidos debido a sus rápidos ciclos de trabajo repetitivos y con una gran capacidad de carga. Este tipo de configuración, generalmente, es más rápida y sencilla que los robots cartesianos y fáciles de montar. A pesar de sus ventajas poseen un alto precio en comparación

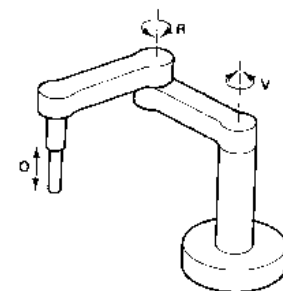


Imagen 7. Esquema robot SCARA.

con los robots cartesianos, también requieren un software de control más complejo. Son comúnmente usados para trabajos de “pick and place” (coger y colocar), operaciones de ensamblado y hasta manipulación de máquinas.

- **Robot de brazo articulado:** El robot se compone de 3 o más juntas de rotación para posicionarse. Su volumen de trabajo es esférico. La disposición morfológica del brazo puede llegar a recordarnos al de un humano, diferenciando el hombro y el codo. Presenta un movimiento rotacional (generalmente en su base) y dos movimientos angulares en el resto de articulaciones. Este robot permite un movimiento rotacional como un movimiento lineal (para este caso deberá existir un movimiento simultáneo de dos o tres de sus articulaciones). Éste tipo de robots se suelen utilizar para operaciones de ensamblaje, fundición a presión, soldadura y pintura en spray, entre otros.

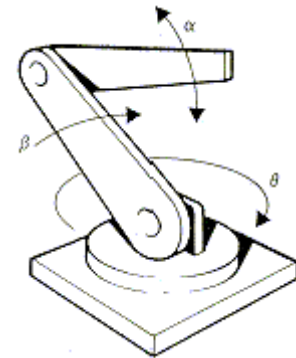


Imagen 8. Robot articulado.

- **Robot antropomórfico:** “Antropomórfico” se refiere a cualquier cosa que tenga el aspecto físico similar al del ser humano. Éstos robots se asemejan a un brazo del ser humano, hombro, brazo, antebrazo y muñeca, que le sirven para colocarse en una posición en el espacio y orientarse para realizar un determinado trabajo. Como se puede ver en la “Imagen 9”, poseen en su mayoría, 5 grados de libertad. Aprovechando la variedad de movimiento que poseen, son muy útiles y versátiles para tareas pesadas y repetitivas. Esta configuración preseta una gran maniobrabilidad, accesibilidad a lugares con obstáculos, una característica muy importante es que son muy rápidos y permiten la realización de trayectorias difíciles. Son los más comúnmente utilizados en líneas de montaje e industrias que requieran manejar objetos pesados y altos volúmenes de trabajo.

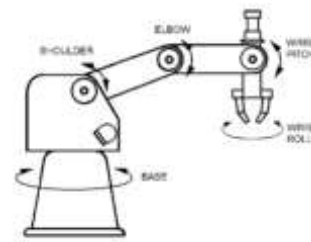


Imagen 9. Robot antropomórfico.

2.3 Impresoras 3D.

Una de las innovaciones más esperadas y revolucionaria de estos últimos años ha sido la aparición de las impresoras 3D (en tres dimensiones). Ha sido tal su aceptación y su innumerable demanda por todos nosotros que hoy en día podemos tener una de ellas en nuestras casas. Teniendo en cuenta que hablamos de un producto que, a priori, no todo el mundo tenga conocimientos para poder utilizarla adecuadamente, existen infinidad de tutoriales por internet explicando perfectamente cómo trabajar con ellas, y gracias a programas como Autocad, es posible diseñar y fabricar piezas en nuestras casas.

Las impresoras 3D se remontan al año 1976, el año de la invención de la impresión mediante inyección de tinta. Alrededor del año 1984, ciertas adaptaciones y avances

sobre el concepto de inyección de tinta hicieron que dicha tecnología sufriese una transformación y se investigase la impresión con materiales. Uno de los principales pioneros en el desarrollo de este tipo de máquinas fue Charles Hull quién inventó la estereolitografía, un proceso de impresión que permite la creación de objetos 3D a partir de datos digitales. Cabe mencionar que este avance supuso una reducción de costos enorme para las industrias ya que podrían crear modelos en 3D a partir de una imagen virtual y permitir a los usuarios probar el diseño evitando así gastar dinero y tiempo en la fabricación de un modelo definitivo.



Imagen 10. Charles Hull, desarrollador de la estereolitografía.

Existen varios tipos de impresoras 3D la cuales se diferencian principalmente en la tecnología usada para llevar a cabo la impresión. Dicho esto, podemos diferenciar las impresoras en:

- **Impresión por Estereolitografía (SLA):** Como hemos comentado anteriormente, esta técnica fue la primera en utilizarse y consistía en la aplicación de un haz de luz ultravioleta a una resina líquida (contenida en un cubo) sensible a la luz. La luz UV (ultra violeta) va solidificando la resina capa por capa. La base en la que se apoyaba la estructura se desplaza hacia abajo para que la luz incida sobre el nuevo baño, y así, capa por capa, se repetía este proceso hasta que el objeto alcanzase la forma deseada.

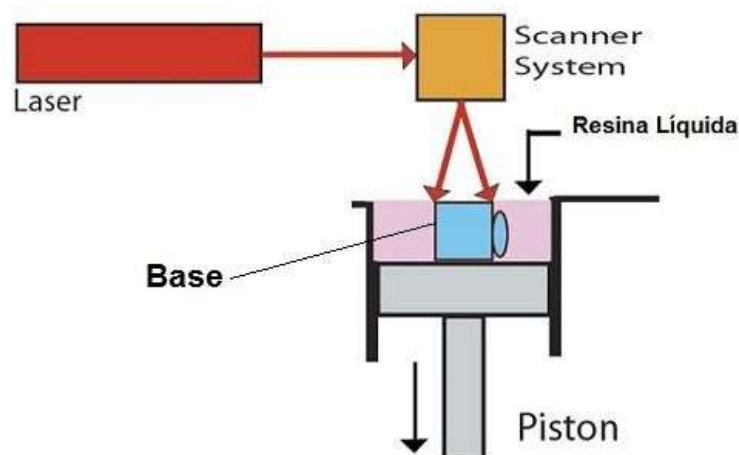


Imagen 11. Técnica impresión por estereolitografía.

- **Impresión mediante Sinterización Selectiva por Láser (SLS):** Conocida en inglés por *Selective Laser Sintering (SLS)*, esta tecnología utiliza un haz de láser para lograr la impresión. El procedimiento de fabricación sería el siguiente: el láser impacta en el polvo, funde el material y este se solidifica. El método de impresión por SLS permite utilizar numerosos materiales en polvo (cerámica, cristal, nylon y poliestireno, entre otros).

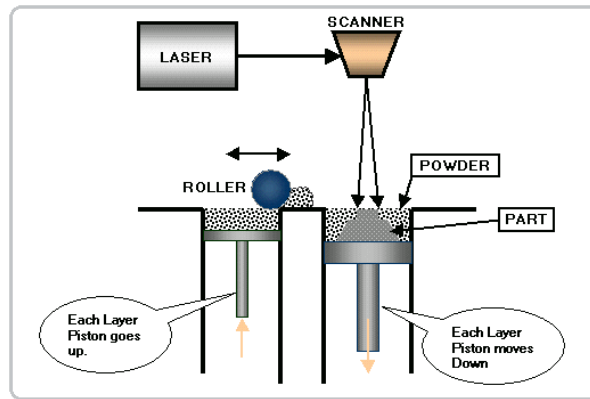


Imagen 12. Impresión por sinterización por láser.

- Impresión por inyección:** Este sistema de impresión es el más parecido a una impresora habitual (de tinta en folio), pero en lugar de inyectar gotas de tinta en el papel, inyectan capas de fotopolímero líquido que se pueden curar en la bandeja de construcción. La máquina deposita polvo composite aplicando una fina capa en la plataforma de impresión. El cartucho de impresión se desplaza por esta capa, aplicando aglutinante (y diferentes tintas si se trata de un modelo a color). El aglutinante solidifica sólo los puntos que correspondan con la geometría diseñada dejando el resto de polvo suelto. Y así sucesivamente, capa por capa se construiría el objeto deseado.

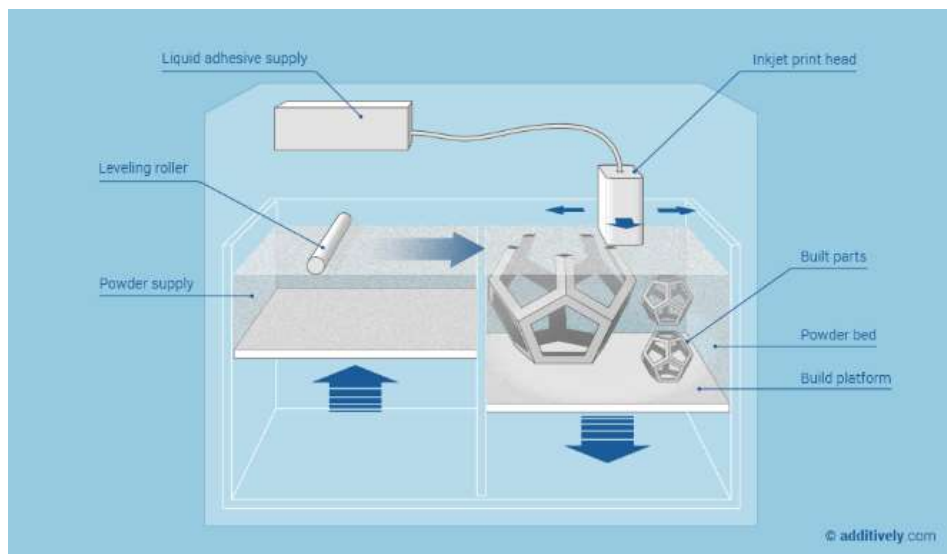


Imagen 13. Impresión por inyección.

- Impresión por deposición de material fundido (FDM):** También conocida como FFF (Fused Filament Fabrication). La técnica por deposición de material es una tecnología basada en la deposición de un polímero fundido sobre una base plana, capa por capa. El material, que inicialmente se encuentra en estado sólido almacenado en bobinas, se funde y se expulsa continuamente por la boquilla en diminutos hilos que se solidifican según van tomando la forma de cada capa. Es el método de impresión más común

utilizado a nivel de usuario doméstico. Aunque los resultados pueden ser muy buenos, no son comparables con los ofrecidos por las impresoras 3D por SLA. La ventaja e importancia de esta tecnología es que ha permitido llevar la impresión en tres dimensiones al alcance de cualquier persona. Los materiales más utilizados para este tipo de impresoras son el ABS y PLA.

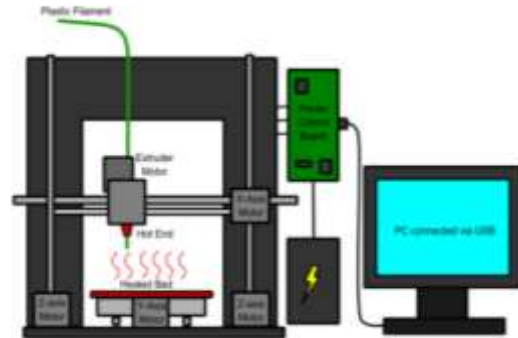


Imagen 14. Esquema impresora método FDM.

Como hemos visto, existen diversos métodos y tecnologías para la impresión de modelos tridimensionales pero todas ellas comparten la misma metodología a la hora de querer imprimir cualquier objeto. Si atendemos a los diferentes procesos de impresión 3D, podemos deducir que la construcción del objeto se lleva a cabo formando capas sucesivas, una encima de la otra hasta que se consigue la forma deseada.

Sin importar que tipo de impresora utilicemos para imprimir, los pasos a seguir son similares a los explicados a continuación:

1. El primer paso que se suele tomar previo a la impresión es la generación de un croquis (Imagen 15), plasmar la idea en papel para poder visualizarla y pensar en posibles cambios y mejoras en su diseño. Al comienzo de este párrafo hemos dicho “suele” pues no tiene por qué ser necesario este paso ya que no estamos generando ningún archivo digital que será cargado posteriormente en la impresora, pero generalmente los diseñadores prefieren exponer sus diseños en dibujos a mano alzada que se aproximarán al objeto final.
2. A continuación, con un programa CAD, diseñamos el objeto en nuestro ordenador (Imagen 16) realizando las mejoras o variaciones que se hayan considerado en la fase anterior. Una vez este el diseño completado se procederá a la exportación del archivo en formato “. STL” (Standard Triangle Language). Este formato de archivo informático de diseño asistido por computadora (CAD) define la geometría del objeto 3D excluyendo información como las texturas, el color u otras propiedades físicas que sí están incluidas en otros formatos CAD, proporcionándonos un mallado global de la pieza a imprimir.

3. Ya tenemos el objeto diseñado y exportado en el formato adecuado, ahora debemos cargarlo en el programa que la impresora 3D posea, que nos dividirá digitalmente el modelo 3D en cortes transversales (Imagen 17). Cada división se corresponderá a una capa que la impresora tendrá que generar cuando comience el proceso de impresión. Es en esta fase dónde se seleccionarán y se modificarán variables como la densidad de relleno, la altura de la capa, si se desean soportes o no, etc,

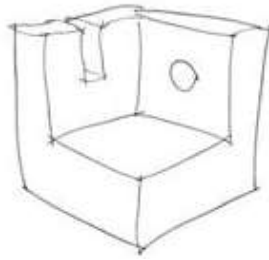


Imagen 15. Croquis del objeto.

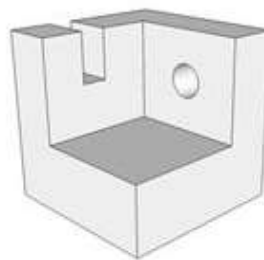


Imagen 16. Diseño CAD.

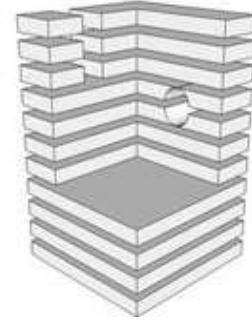


Imagen 17. Separación en capas para la impresión.

4. El cuarto paso a seguir será la preparación de la máquina para la impresión. Tendremos que colocar el material de impresión que hemos escogido para el objeto, limpiar la cama de impresión para evitar suciedades y que puedan aparecer defectos indeseados.
5. Entramos en la fase de impresión, ordenando a la impresora a que trabaje con el archivo que el programa de separación por capas nos ha generado, imprimiendo así, capa por capa, hasta que el modelo se complete. En este paso podemos modificar variables como la densidad de relleno, si queremos que genere soportes o la altura de capa que creamos conveniente.
6. Finalmente retiramos la pieza una vez se haya respetado el tiempo de espera para asegurarnos de que la plataforma de impresión se haya enfriado. Retiraremos el objeto de la plataforma de impresión y seguiremos un proceso de post-procesamiento, eliminando los soportes (si los hay), lijando bordes o cualquier acción necesaria para conseguir la pieza deseada.

A continuación, entraremos un poco más en detalle sobre las impresoras por deposición de material fundido (FDM) ya que son las más versátiles y con las que se han trabajado en este proyecto. Las principales ventajas de esta tecnología son:

- 1) Es un método limpio, de fácil utilización y perfecta para hogares y oficinas.
- 2) Los materiales termoplásticos con los que trabajan estas impresoras son estables mecánica y medioambientalmente.
- 3) Esta tecnología puede trabajar con geometrías y cavidades que podrían ser problemáticas al usar otros métodos de impresión.

Los programas que incluyen las impresoras FDM, por muy diferentes que sean entre sí, comparten ciertas variables que definirán el objeto, ya sea su peso o su acabado superficial. Seguidamente, nombraremos los más habituales:

- **Tipo de soporte:** Puesto que habrá casos a la hora de seccionar el objeto nos quedarán voladizos que la impresora deberá formar, y para garantizar la solidez del modelo deberemos echar mano a esta sección. La opción “Tipo de soporte” permite al usuario crear de forma automática una especie de andamios para depositar el material sobre ellos mismos. Son estructuras ligeras que se pueden retirar con facilidad. Existen 3 posibles configuraciones:
 - Sin soportes: No crea ningún soporte alrededor del objeto.
 - Soportes tocando la plataforma: Construirá soportes sólo desde la plataforma hasta la pieza en las zonas que se necesiten.
 - En todas partes: Generará soportes en todos los voladizos existentes.

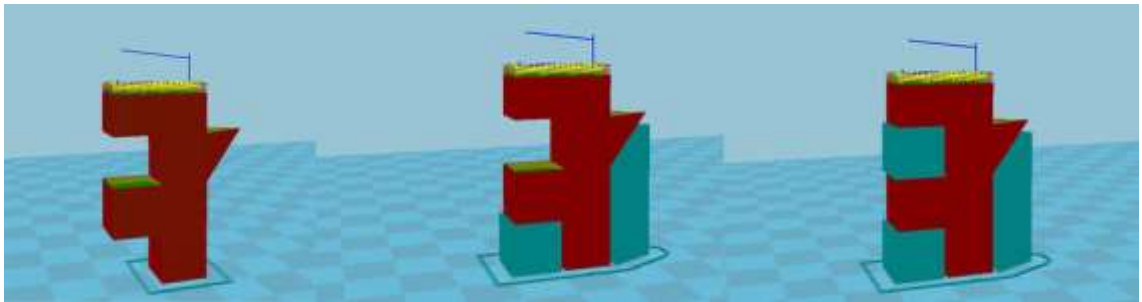


Imagen 18. Ejemplo de las diferentes opciones de soportes. El modelo de la izquierda nos muestra el objeto sin soportes, el central con soportes tocando la plataforma, y el de la derecha en todas partes.

- **Tipo de adhesión a la plataforma:** A la hora de imprimir piezas de gran tamaño, en ciertos casos se observa que las esquinas de estas suelen despegarse de la cama de impresión debido a la contracción del material al enfriarse. Este efecto de contracción del material y de su consecuente despegue de cama de impresión se le denomina “Warping” y puede provocar malformaciones durante el proceso de impresión, por ello es importante tener esta opción en cuenta. Existen 4 tipos de adhesiones a la plataforma:
 - Ninguna: No se crea ninguna base de adherencia.
 - Raft: El sistema Raft crea una celosía horizontal entre la base y la pieza. Ya que esa celosía tiene más área de contacto con la base que con la pieza, favorece la adherencia y evita problemas de nivelado y desajuste.
 - Brim: Este método es ideal para piezas con poca base o que contengan partes estrechas (una torre, engranajes...). Se crea de 1 a 2 capas de altura para conseguir

adhesión suficiente para evitar el efecto Warping comentado anteriormente. Por el contrario, utilizar la técnica Brim hace que se utilice más material para la creación de la base y nos encontramos con la dificultad de eliminar estas capas del objeto impreso.

- **Skirt:** Este parámetro elabora una copia del contorno del modelo a imprimir previo a la impresión. Con esto nos aseguramos de purgar el extrusor y de nivelar la base, evitando posibles acumulaciones de material indeseadas.

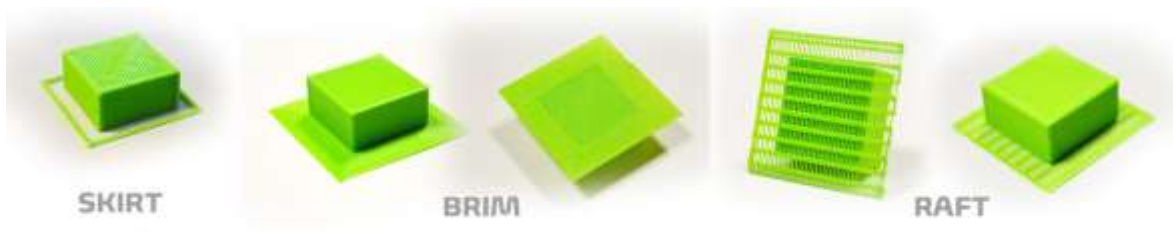


Imagen 19. Diferentes tipos de adhesión.

- **Altura de la capa:** Podríamos decir que es uno de los parámetros más importantes y útil ya que define la calidad superficial del objeto. Este valor controla la altura de cada capa de la pieza. Al disminuir la altura se notarán menos las capas y por ellos conseguiremos un acabado superficial más suave pero por el contrario aumentaremos el tiempo necesario de impresión. La altura se mide en micras (10^{-6} metros).



Imagen 20. Comparación de alturas de capas.

- **Densidad de relleno:** El relleno es la parte interior de la pieza que principalmente le da peso y este parámetro se encarga de proporcionarle tal densidad. Según para que función vaya destinada la pieza, el relleno podrá variar desde 0% (pieza totalmente hueca) al 100% (pieza totalmente sólida). Diversas fuentes aconsejan imprimir con rellenos del 20% al 60% pero como hemos comentado depende del modelo a imprimir y de la función que este desempeñe.

Al igual que podemos variar la densidad, también podemos modificar el patrón que seguirá la impresora para rellenarlo. En la Imagen 21 que se muestra a continuación, aparecen algunos de los patrones más utilizados y el porcentaje de relleno.

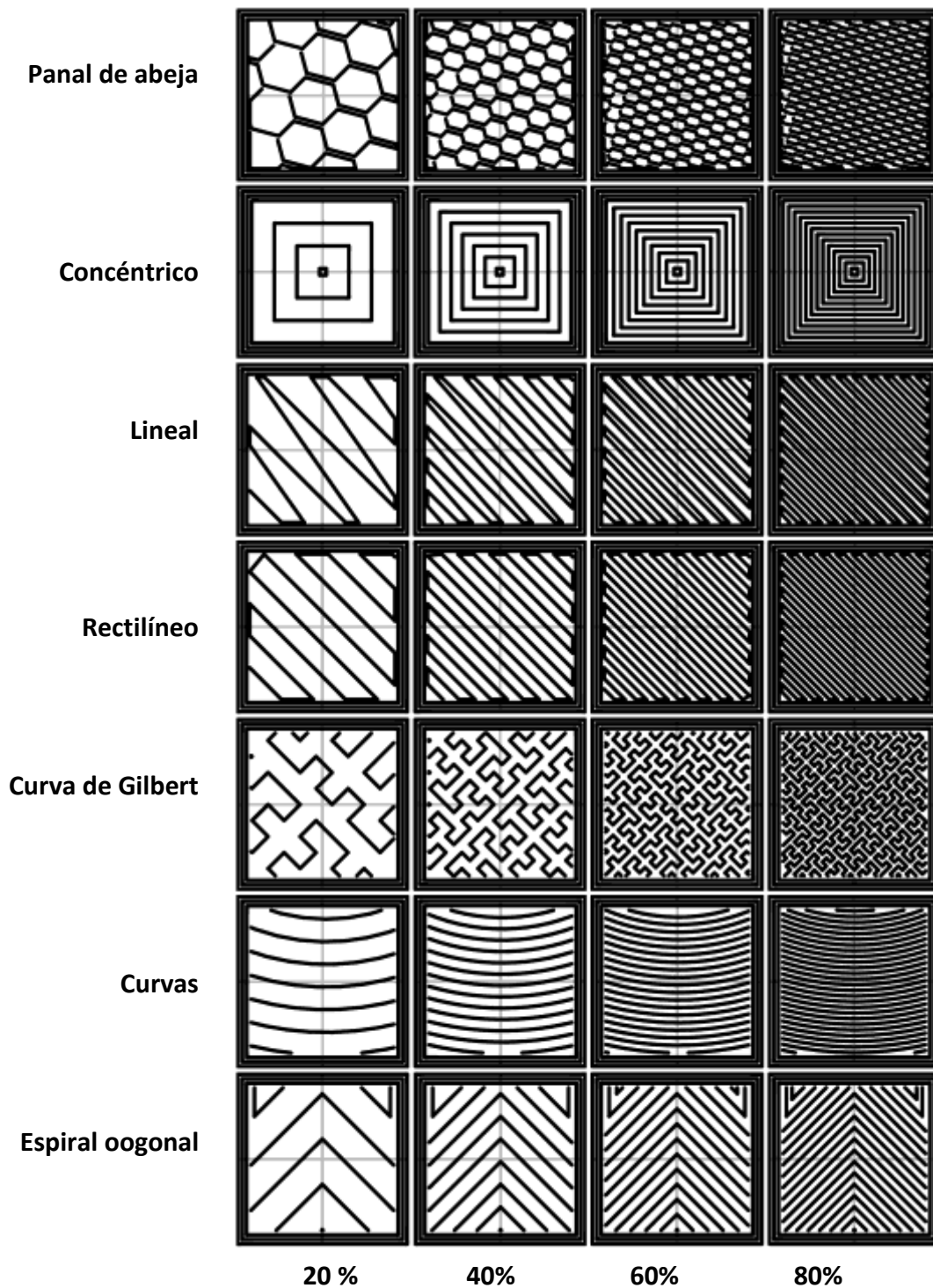


Imagen 21. Densidades y patrones de relleno.

2.4 Microcontrolador y microprocesador.

Actualmente vivimos en el boom de la electrónica donde se dan grandes avances tanto a nivel de componentes como a nivel informático. Cada vez hacemos los componentes más pequeños y más eficientes. Antiguamente cuando los microprocesadores aún no existían, las personas diseñaban sus propios circuitos electrónicos que implicaban numerosos componentes y cálculos matemáticos. Construir un circuito lógico requería de numerosos elementos como transistores y resistencias. Inicialmente se creía que para manejar un microprocesador hacía falta tener un coeficiente intelectual muy alto puesto que eran sistemas complejos de entender, hasta que aparecieron los circuitos integrados que facilitaron el entendimiento y disminuyeron el tamaño y la complejidad de los diseños electrónicos.

En las últimas décadas se ha disparado la aparición de los microcontroladores, pequeños autómatas que gobiernan la mayor parte de los aparatos fabricados hoy en día. Aparecen más y más productos que incorporan microcontroladores con el fin de aumentar sus prestaciones, reducir su tamaño y el costo. Las aplicaciones de los microcontroladores y microprocesadores nos permiten profundizar en los aspectos tecnológicos de las nuevas computadoras, brindándonos una herramienta útil para desarrollar aplicaciones que contribuyen al avance tecnológico.

Estos dispositivos tienen como finalidad controlar multitudes de dispositivos ya sea el ratón el ordenador, teléfonos móvil, robots, etc. Las áreas de aplicación para los microcontroladores y microprocesadores se pueden considerar casi ilimitadas pero la principal industria que acapara la gran parte de microprocesadores y microcontroladores es la industria informática. Cuando optamos por comprar ya sea un microprocesador o un microcontrolador, encontramos dos principales productos: Arduino y Raspberry Pi. Es importante saber qué funciones y aplicaciones queremos que desempeñe nuestro proyecto para elegir correctamente que producto utilizar.

Tanto Arduino como Raspberry Pi fueron diseñadas como herramientas de enseñanza, debido a eso, al fácil uso y aprendizaje se han vuelto muy populares. Para poder entender la diferencia entre explicaremos que es cada uno de ellos:

- **Raspberry Pi:** En la página web del producto lo describen como “Un pequeño ordenador capaz, que puede ser utilizado por muchas de las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeos de alta definición”. Es un producto desarrollado en Reino Unido por la fundación Raspberry Pi por Eben Upton y sus compañeros de Informática, cuyo objetivo principal es la enseñanza de computación en las escuelas y universidades. Los primeros diseños de Raspberry Pi se basaban en el microcontrolador Atmel ATmega644 cuyos esquemas y diseños del circuito están disponibles para su descarga. Eben Upton (administrador de la fundación) se puso en contacto con un grupo de profesores, académicos y gurus de la informática para crear una computadora con el fin de promover la informática entre los niños.



Imagen 22.
Microprocesador
Raspberry Pi.

En Agosto de 2011, se fabricaron alrededor de 50 versiones alfa. Se hicieron variadas demostraciones de la placa ejecutando el escritorio del sistema operativo Debian, el videojuego Quake 3 en 1080p y vídeos Full HD. Fue en Octubre de ese mismo año cuando se seleccionó el logotipo de entre varios por miembros de la comunidad. Ese mismo mes se hizo una demostración en público.

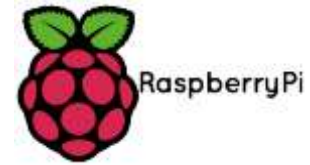


Imagen 23. Logotipo Raspberry Pi.

A inicios de Diciembre de 2011 se subastaron 10 placas raspberry en eBay (una centro de compra y venta en internet) de las cuales, la de número de serie No. 01 se vendió por 3.500£.

A la llegada de Febrero del 2012, la compañía anunció su venta anticipada al público y sufrió un colapso en sus servidores web debido a los numerosos refrescos de páginas desde los navegadores de la gente que quería comprar la placa.

Nos encontramos ante un producto Open Source (Código Abierto) que implica un software y desarrollo libre, siendo su sistema operativo Raspbian aunque permite el uso de otros sistemas operativos como Windows, Linux o Android.

Todas las versiones incluyen memoria RAM, una GPU (tarjeta gráfica), conexión HDMI, puertos USB y Ethernet. El principal inconveniente que nos encontramos es que no medio de almacenamiento, lo que implica la compra de una tarjeta SD o microSD externa de 4GB como mínimo.

- **Arduino:** Atendiendo a la propia página web del producto lo describen como "...una plataforma de prototipos electrónica de código abierto (open source) basada en hardware y software flexibles y fáciles de usar". *Arduino* se originó en Italia y debe su nombre al restaurante donde su fundador Massimo Banzi y sus compañeros dieron con la idea. Banzi era profesor del instituto IVREA en la localidad de Ivrea (Italia). Este instituto donde daba clase finalmente cerró y posteriormente Banzi afirmó que el proyecto nunca surgió para lucrarse sino para crear una plataforma libre que no podría ser embargada. La aparición de la primera placa de Arduino en el mercado fue en 2005, ofrecía bajo costo y de fácil uso para novatos. Más tarde, en 2011 se presentó la placa *Arduino* de 32 bit que realizaba tareas más pesadas.



Imagen 24. Logo Arduino.

Arduino es un microcontrolador de hardware libre cuya comunidad diseña y manufactura placas de desarrollo de hardware. Las placas son programadas a través de un software específico facilitado por la compañía, el cual compila el código y lo carga al modelo de placa seleccionado. Su principal enfoque es el de poner la electrónica y programación a nivel de cualquier persona siendo su fácil uso y entendimiento. Puesto que estamos hablando de una plataforma open source, todos sus componentes hardware y software están libres de licencias lo que permite libertad

de acceso a ellos. La placa consiste en un circuito impreso con un microcontrolador, contiene puertos analógicos y digitales que se pueden comportar tanto de entrada como de salida. Las funcionalidades de la placa se pueden ampliar conectando placas de expansión, lo que *Arduino* denomina “Shields”. Además, la placa contiene un puerto USB por donde se puede alimentar y establecer conexión con el ordenador. Cualquier placa de *Arduino* esta disponible de dos formas:

○ Hazlo tu mismo (“DIY”, siglas en inglés):

Hemos dicho que *Arduino* era una plataforma open source, gracias a este tipo de licencia, los esquemas de diseño de Hardware estan al alcance de cualquiera y puede fabricarlos por su propia cuenta y posteriormente. Siguiendo las indicaciones del fabricante para ensamblarlo todo, obtendríamos una placa similar que la que nos vendría montada con las mismas prestaciones.



Imagen 25. *Arduino* DIY.

- Ensambladas: La placa te viene ya montada y fabricada por la compañía sin necesidad de modificar absolutamente nada. Incluiría un cable USB para el flujo de datos entre la placa y el ordenador.



Imagen 27. Placa *Arduino* UNO.

Cabe mencionar que a simple vista podríamos decir que *Arduino* y *Raspberry Pi* son dos productos similares, pues ambos incluyen su unidad de procesamiento de información y ciertas entradas y salidas,

pero realmente estaríamos hablando de un Autómata programable (*Arduino*) y básicamente un ordenador (*Raspberry Pi*). Lo que si comparten estas plataformas es que ambas están destinadas a la iniciación tanto en la electrónica como en la informática, y son herramientas idóneas para su uso académico. Para elegir correctamente cuál de ellas nos vendrá mejor para el proyecto debemos tener en cuenta sus ventajas.



Imagen 26. Cable USB de tipo A a tipo B.

Comenzando con *Arduino*, es un microcontrolador mientras que *Raspberry Pi* es un miniordenador. Los microcontroladores forman una pequeña parte del conjunto de un ordenador por lo que el nivel de funcionalidades de *Raspberry Pi* será mucho más amplio que el de *Arduino*. También hay que destacar que *Arduino* es una placa donde se pueden programar aplicaciones en C++ mientras que en *Raspberry Pi* podemos ejecutar un sistema operativo entero.

Visto así, podríamos decir que *Raspberry Pi* al poseer mayor capacidad de funcionalidad, es más completa y por tanto, mejor. No es así, *Arduino* tiene la ventaja de la simplicidad en lo que a proyectos de electrónica se refiere.

Existe una gran facilidad para enchufar componentes a sus conectores, esto se traduce a hacer cambios rápidos y proyectos que no requieran tareas muy complejas. También es muy directa, tan pronto sea alimentada con corriente comenzará a ejecutar el programa que se le haya cargado. Anteriormente hemos dicho que *Arduino* se programa con un software que te facilita la propia firma y se puede descargar de su página web, gracias al cable USB podemos cargar las aplicaciones a la placa y una vez cargado el programa no necesitamos nada más que alimentarla, sin necesidad de teclados ni pantallas ni ejecutar manualmente la aplicación.

Por otra parte, *Raspberry Pi* posee todas las ventajas, y desventajas, de un ordenador. Tiene una gran velocidad de procesamiento, gran potencia de cálculo y puede manejar cantidades de memorias grandes. Cuenta también con una conectividad Wifi y Ethernet. Además, encontramos un chip gráfico (GPU) y su conexión HDMI. Como se observa, tenemos una capacidad de procesamiento de datos importante en un espacio reducido (89 x 49 mm).

Podemos concluir que *Arduino* sería ideal para proyectos electrónicos que no requieran de tareas muy complejas y requieran un mínimo de conocimientos de programación. Sin embargo, utilizaríamos *Raspberry Pi* para proyectos que necesiten más potencia de procesamiento de datos, así como cálculos complicados y alta complejidad en las aplicaciones.

2.5 Arduino.

En el apartado anterior se han comentado diversas propiedades sobre la plataforma *Arduino*, mencionando que se trata de un microprocesador, de una categoría de Hardware open source es por esto que las placas pueden ser fabricadas por uno mismo o compradas montadas de fábrica.

El lenguaje de programación para el microcontrolador de la placa es propio de *Arduino* y en internet podemos encontrar numerosos tutoriales y manuales que nos explican su programación así como su montaje, en el caso que hayamos optado por la opción DIY. La comunidad de usuarios de *Arduino* ayuda mucho a la gente que tenga problemas a través de un foro creado por ellos mismos, de esta manera los mismos usuarios ayudan al desarrollo y mejora de la plataforma.

Arduino nos proporciona una gran variedad de placas desde la más barata y sencilla con capacidad para ejecutar aplicaciones poco complejas y de poco tamaño, hasta placas con más de 50 pines digitales y un poderoso microcontrolador.

Este proyecto utilizará una placa de *Arduino UNO*, cuenta con 14 pines digitales que se pueden configurar como entradas o salidas. Funcionan a 5V, con un máximo de corriente de 40mA. Poseen una resistencia interna pull-up de entre 20k Ω y 50k Ω (este tipo de resistencias se utilizan normalmente en electrónica

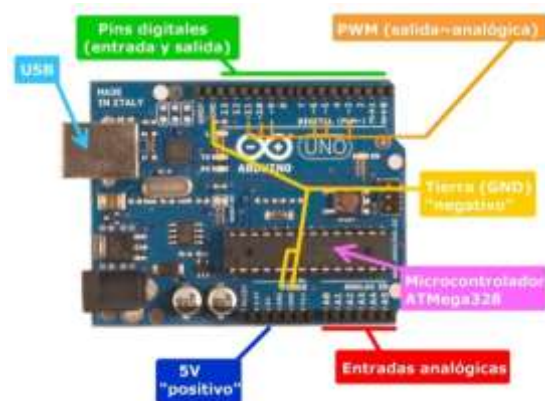


Imagen 27. Placa Arduino UNO.



digital para asegurarnos de tener los niveles lógicos correctos, 1 o 0, HIGH o LOW, 5V o 0V respectivamente). *Arduino UNO*, a su vez, brinda a nuestra disposición 6 pines de entrada analógicos con una resolución de 10bits (1024 valores diferentes) que nos convierten la señal analógico/digital internamente. La alimentación de la placa puede llevarse a cabo de dos formas: a través del propio cable USB o mediante una fuente de alimentación externa con un conector de 2,1 mm con el positivo en el centro o utilizando los pines Vin (+) y GND (-) de la placa. La placa trabaja entre 6 y 12 V. En resumidas cuentas, las características técnicas de la placa son las siguientes:

- Microcontrolador: Atmega328
- Voltaje de operación: 5V
- Voltaje de entrada: 7 – 12V
- Voltaje Límite: 6 – 20V
- Pines entrada/salida digital: 14 (6 de ellos soportan salida PWM)
- Pines entrada analógica: 6
- Corriente continua por pin Entrada/Salida: 40mA
- Corriente continua por pin 3.3V: 50mA
- Memoria Flash: 32KB
- SRAM: 2KB
- EEPROM: 1KB
- Frecuencia de reloj: 16MHz

Es de gran interés mencionar ciertos pines de entrada y salida como los pines PWM capaz de generar señales PWM de hasta 8 bits. Estos pines están señalizados con el símbolo “ ~ ”. También encontramos que los pines 10, 11, 12 y 13 están habilitados para comunicaciones SPI, que permiten trasladar información full dúplex en un entorno Maestro/Esclavo.

Por último, otra característica de la placa es que incluye el pin I²C la cual permite comunicaciones a través de un bus I²C, que fue creado por Philips para la interconexión de sistemas embebidos. Hoy en día se pueden encontrar una gran diversidad de dispositivos que utilizan esta interfaz, sensores, pantallas LCD, memorias EEPROM...

2.6 Thingiverse.

La plataforma *Thingiverse* se basa en la publicación de diseños 3D creada por la marca *Makerbot*. Actualmente es la mejor página para compartir piezas, figuras, copias y mejoras de otros diseños para imprimir uno mismo en casa. La página web se basa en la licencia OpenSource, todos los diseños subidos se categorizan de licencia libre, accesibles al pública sin restricción alguna.

Una curiosidad de la plataforma es que existen diversos proyectos que contienen los planos necesarios para imprimir tu propia impresora 3D, mostrándote los pasos a seguir y como ensamblar las piezas así como los materiales necesarios para su correcto montaje. De este modo conseguiríamos mediante impresión tridimensional nuestra propia impresora 3D. El

objetivo principal de la página web es promover la difusión de proyectos y que otros usuarios pueden trabajar sobre ellos para mejorarlos o para cambiar el enfoque.

El caso concreto de este proyecto se centra en la construcción de un brazo robótico utilizando unos servo-motores específicos (micro servo 9g) utilizando la impresión 3D. La impresora que utilizaremos para poder imprimir las piezas será la proporcionada por el laboratorio de impresión 3D de la propia escuela. Usaremos una impresora Zortrax m200. Los archivos se pueden descargar del siguiente enlace: <https://www.thingiverse.com/thing:65081>

2.7 Circuitos eléctricos.

El presente proyecto contiene elementos eléctricos y electrónicos que en su conjunto forman un circuito cerrado, por ello hemos decidido mostrar unas nociones básicas y pequeñas pinceladas para entender lo mejor posible el funcionamiento de este.

Una definición clara y directa sería: “Un circuito eléctrico es un conjunto de elementos conectados entre sí por los que puede circular una corriente eléctrica”. Ésta corriente eléctrica se desplaza de un punto con mayor potencial eléctrico a otro que tiene menor potencial. Esa diferencia de potencial se llama Voltaje o tensión entre los extremos del conductor. Para lograr ese movimiento de electrones llamado corriente eléctrica, necesitamos un generador o fuente que recoja las cargas de un extremo y las impulse hacia el otro.



Imagen 28.
Parámetro de la ley de Ohm.

Existe una ley que rige la mayoría de los circuitos y es la más básica de todas, es la conocida ley de Ohm. Postulada por Georg Simon Ohm, establece que la diferencia de potencial o voltaje (V , medido en Voltios) existente en los extremos de un determinado conductor es directamente proporcional a la intensidad de corriente que lo atraviesa (I , medido en Amperios) y su resistencia eléctrica (R , medida en Ohmios).

Pero no todos los circuitos pueden acogerse a esta ley pues existen diferentes tipos de corrientes y según cuál de ellas lo atravesase se comportará de una forma u otra. Podemos distinguir dos tipos de corrientes:

- **Corriente continua:** En los circuitos de corriente continua (también conocida como CC o DC en inglés) la polaridad no varía a su vez que los electrones fluyen en el mismo sentido y cantidad. Este tipo de corriente es muy común en los circuitos electrónicos portátiles, pues requieren de un voltaje relativamente bajo.



Imagen 29. Ejemplo corriente CC.

- **Corriente alterna:** Por el contrario, llamaremos corriente alterna aquella que varíe su sentido en el tiempo, es decir, circulará durante un periodo de tiempo en un sentido y después en el sentido opuesto, repitiéndose de forma constante. De la misma manera

que la corriente en ambos sentidos, la polaridad se comporta de igual forma. En castellano también se la conoce como CA y como AC en inglés. El ejemplo más sencillo de corriente alterna la tenemos día tras día en nuestras casas. Todos los enchufes que nos encontramos funcionan con corriente alterna.

Todos los circuitos eléctricos, o en su totalidad, están compuestos de los siguientes elementos:

1. **Generadores:** Se encargan de producir y mantener la corriente eléctrica. Son la fuente de energía y nos proporcionan esa diferencia de potencial necesaria para el flujo de electrones. Encontramos dos tipos de generadores: pilas y baterías que nos brindan corriente continua y alternadores que producen corriente alterna.
2. **Conductores:** Son elementos por los cuales circula la corriente eléctrica de un elemento a otro en los circuitos. Están fabricados de materiales que ofrecen poca resistencia al paso de la corriente como el cobre o el aluminio.
3. **Receptores:** Dichos elementos transforman la energía que reciben en otro tipo de energía. Por ejemplo, las resistencias de los hornos transforman la electricidad en calor, las bombillas en luz o los motores que transforman la energía eléctrica en movimiento.
4. **Elementos de control:** Nos permiten actuar sobre el comportamiento del circuito ya sea abriendo o cerrando el circuito o conmutando el paso de corriente entre dos conductores. Existen numerosos elementos para controlar el circuito ya sean interruptores, pulsadores, potenciómetros, relés...
5. **Elementos de protección:** Los catalogaríamos como los elementos más importantes de los circuitos pues su función es la de proteger los circuitos y a las personas de posibles daños debido a que la corriente es muy elevada o haya peligro de electrocución. Dentro de este grupo encontramos componentes como los fusibles, interruptores magneto térmicos, diferenciales, etc.

Para poder identificar todos estos elementos en un circuito eléctrico debemos familiarizarnos con la simbología que representan los componentes en los circuitos. A continuación en la imagen 30 se muestran los elementos comúnmente utilizados.

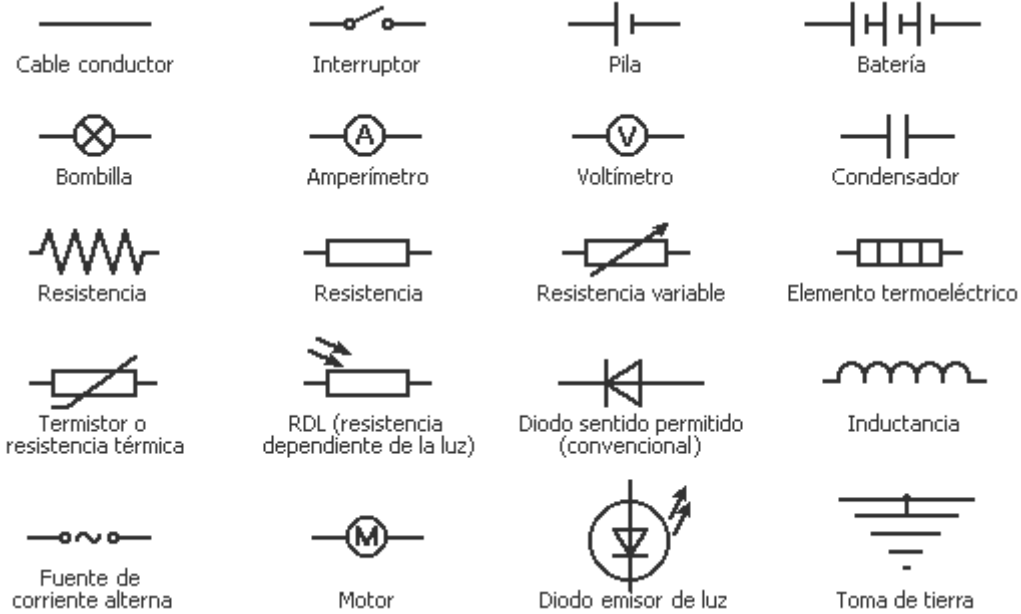


Imagen 30. Simbología de elementos comunes en un circuito eléctrico.

Hemos visto los diferentes elementos que, por norma general, suelen tener los circuitos, también hemos mostrado que esquemas representan esos elementos para poder identificarlos a la hora de ojear un esquema eléctrico. Para finalizar, explicaremos las posibles disposiciones de los receptores que según se conecten tendrán unas propiedades u otras. Ya que este proyecto se basa en circuitos de corriente continua, nos centraremos en los de corriente continua que pueden ser de 3 tipos:

- **Circuitos en serie:** Los elementos están conectados de manera que la salida de uno está unida con la entrada del siguiente (imagen 30). De modo que la corriente circula uniforme por todos los elementos del circuito ya que dispone de un único camino. Sin embargo el voltaje es la suma de la diferencia de potencial entre los bornes de cada receptor del circuito y la resistencia la suma directa de ellas.

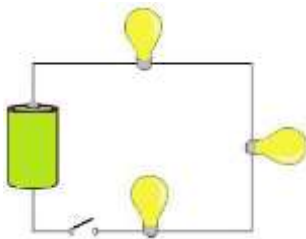


Imagen 30. Circuito en serie.

- **Circuitos en paralelo:** Cuando conectamos los elementos en paralelo, las entradas de estos están conectadas entre sí formando una división en el camino de la corriente (ver imagen 31). De este modo conseguimos que la tensión es la misma en todas sus entradas y que el total de la intensidad de corriente que proporciona el generador sea la suma de la intensidad que trasiega cada receptor. Se considera la conexión más estable y es la más utilizada por ello.

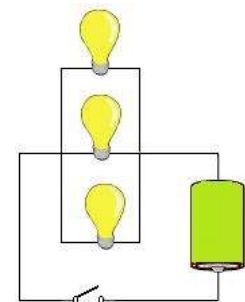


Imagen 31. Circuito en paralelo.

- **Circuitos mixtos:** Denominamos circuitos mixtos a la combinación entre las disposiciones anteriormente explicadas, varios elementos conectados en paralelo y a su vez otros conectados en serie.

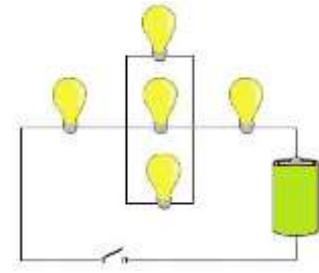


Imagen 32. Circuito mixto.

2.8 Potenciómetros.

Un potenciómetro es una resistencia eléctrica de valor variable que se ajusta manualmente. Existen múltiples tipos de potenciómetros, pero todos contienen 3 terminales de conexión. En diversos dispositivos, estos elementos se utilizan como elementos de control para establecer el nivel de la salida. Un ejemplo claro sería el control del volumen de un altavoz. Su misión es limitar el paso de la corriente eléctrica creando una caída de tensión. Esta caída de tensión variará según modifiquemos su resistencia.

El funcionamiento de estos elementos es relativamente simple. Fijándonos en la Imagen 33, observamos los 3 terminales que lo componen A, B y C y el cable enrollado representaría el

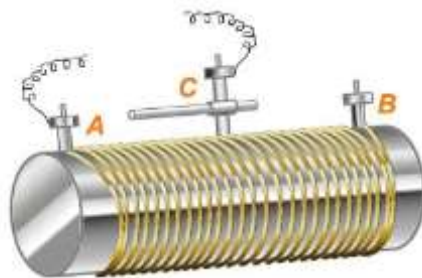


Imagen 33. Esquema potenciómetro.

hilo conductor por el que circulará la corriente. Si conectásemos los terminales A y B obtendríamos una resistencia de valor constante igual al valor máximo del potenciómetro. Por el contrario si lo que queremos es trabajar con una resistencia variable, deberemos conectar uno de los terminales extremos (A o B) con el central (C). De este modo, la resistencia dependerá de en qué posición se encuentre el terminal C que podrá moverse de un lado hacia otro. A esta clase de potenciómetros que varían su resistencia

mecánicamente se les denomina reóstatos, suelen tener resistencias altas y se suelen utilizar en circuitos eléctricos con mucha intensidad de corriente. La Imagen 34 nos muestra su representación en los esquemas eléctricos.



Imagen 34. Símbolo del potenciómetro.

Las resistencias de variación mecánica se diferencian en:

- **Impresas:** Se fabrican mediante una pista de carbón o cermet (el cermet es un material compuesto de cerámica y metal) sobre un soporte de baquelita, vidrio... La pista tiene un cursor conectado a un patín que se desliza por la pista resistiva.
- **Bobinadas:** Consiste en arrollar un hilo resistivo y un cursor que se desplaza sobre el mismo (similar al de la Imagen 33).

También diferenciamos los potenciómetros según la disposición del cursor para variar la resistencia:

- **Potenciómetros lineales (o deslizantes):** Se les llama de esta forma porque su valor varía a lo largo de una línea.
- **Potenciómetros rotatorios:** El mecanismo es similar pero el cursor describe una trayectoria circular. Este tipo de resistencias variables son más compactas y por ello más utilizadas.

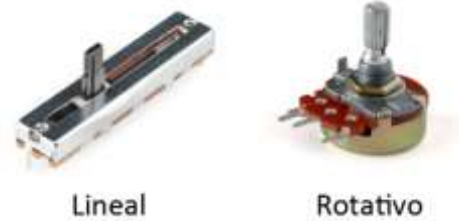


Imagen 35. Potenciómetros lineal y rotativo.

Otra distinción que podemos hacer entre los potenciómetros es la de la variación de la resistencia a lo largo de la pista. Los hay de 3 tipos: lineales, logarítmicos y senoidales:

- **Lineales:** La resistencia varía linealmente con el recorrido del cursor. El aumento de resistencia es igual en los primeros instantes del recorrido que al final.
- **Logarítmicos:** El incremento en los primeros instantes de la resistencia son muy leves, hasta llegar a un punto en el que el incremento entre valores será mucho mayor.
- **Senoidales:** Este tipo se da en los potenciómetros rotativos ya que dependen del seno del ángulo girado.

Como último punto, comentaremos unos componentes que varían su resistencia pero no se consideran potenciómetros propiamente dichos. Estos son:

- **LDR (Light Dependent Resistor):** Estas resistencias varían según la cantidad de luz que incida sobre sus receptores.
- **Resistencias variables con la temperatura:** Existen de dos tipos: NTC y PTC. Las NTC aumentan su resistencia al disminuir la temperatura. Y las PTC aumentan su resistencia al aumentar la temperatura.



Imagen 36. LDR y símbolo esquemático.



Imagen 37. Resistencias PTC y NTC.

2.9 Servomotores.

Los servomotores son un tipo especial de motores eléctricos que permiten controlar la posición del eje de giro, pudiendo mantenerlos en una posición fija deseada. La diferencia de estos motores con los de corriente continua es que los de CC son capaces de girar sin detenerse, no pueden pararse en una posición determinada. Es por ello que no se utilizan para aplicaciones de robótica ya que en este ámbito se necesitan movimientos precisos y en ciertos casos mantener una posición fija. Otra característica de los servomotores frente a los de corriente continua, es que son capaces de entregar mayor par de torsión.

Cuando hablamos de un servomotor, hablamos de un conjunto de componentes mecánicos y electrónicos. El motor que estos contienen son motores de corriente continua comunes. En el eje de éste, se encuentra un pequeño sistema de transmisión de engranajes que además de potenciar el par del motor y permitir que se pueda mantener una posición fija, contiene un engranaje unido al vástago de un potenciómetro rotativo. Por último, la parte electrónica se encarga de dirigir el movimiento y la posición del motor.



Con anterioridad, los servomotores no ofrecían ángulos de giro superiores a 180°. Pero hoy en día es posible controlar la posición y la velocidad de giro en los 360°. Las características principales de estos motores especiales son el par y la velocidad. El par mide la fuerza que su eje es capaz de ejercer y se expresa como Kg/cm. La velocidad se refiere a la velocidad de rotación o angular.

Para poder trabajar adecuadamente con estos motores debemos prestar atención a los cables que poseen. Son 3 y siempre tienen los mismos colores para facilitar su identificación. En la Imagen 39 observamos el sistema de colores.

Voltaje positivo	Tierra (ground)	Señal de control

Imagen 39. Sistema de colores de los cables de un servomotor.

Con este sistema difícilmente nos equivocaremos a la hora de reconocer la función de cada cable. Ya que se precisa de una señal de control, necesitamos un sistema de control adecuado. Será necesaria una señal de control modulada para que el circuito electrónico interno funcione adecuadamente, es decir, una señal PWM. La imagen 40 muestra un diagrama de funcionamiento del servomotor.

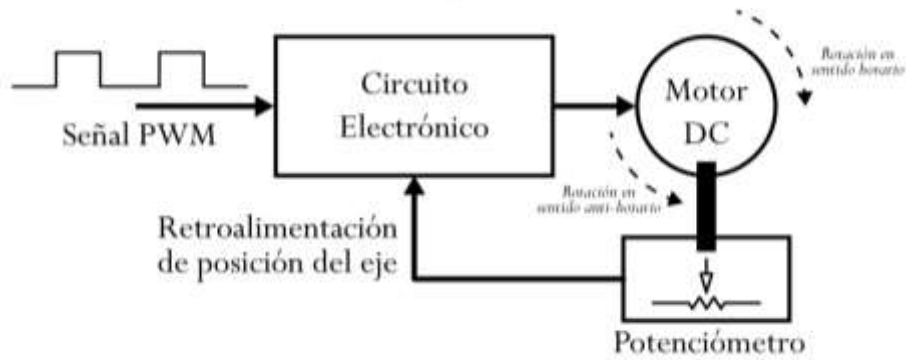
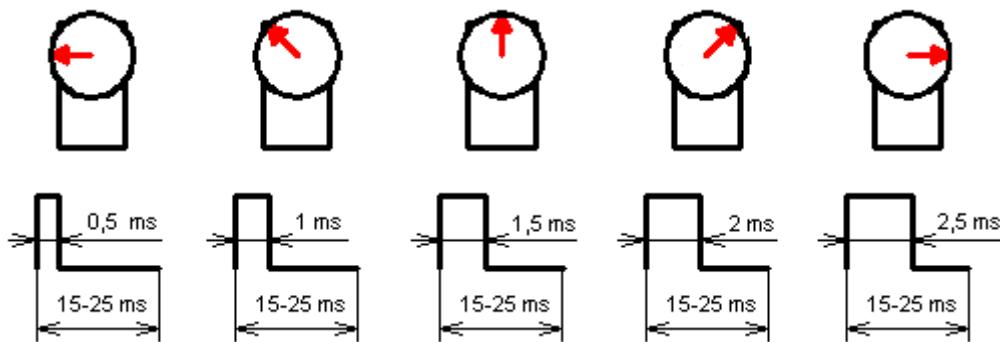


Imagen 40. Diagrama de los componentes de un servomotor.

Anteriormente comentado, los servomotores de corriente continua hacen uso de señales que será traducida por el circuito electrónico. Las señales PWM tienen forma de onda cuadrada y dependiendo del ancho del pulso adoptará una posición fija. En la imagen mostrada a continuación observamos distintas posiciones que responden a un determinado ancho de banda.



Para que el motor mantenga la posición fija, estas señales han de mantenerse en el tiempo. La frecuencia de las señales PWM variará entre 15 y 25 milisegundos, aunque muchos de los servos trabajan a una frecuencia de 20 milisegundos (50 hercios).

El uso de estos motores peculiares se ha extendido por todos los ámbitos, desde el zoom de una cámara de fotos hasta un robot industrial, pasando por aviones, helicópteros, trenes y sin duda, la comunidad Arduino para la creación de sistemas automatizados, aviones radiocontrol y pequeños brazos robóticos.

3. Ejecución del proyecto.

3.1 Aspectos a tener en cuenta.

Primero de todo, para lograr con éxito la realización del presente proyecto necesitamos tener las ideas claras de que queremos crear, como lo haremos y que medios utilizaremos para llevarlo a cabo.

La idea principal del proyecto es construir un brazo robótico que sea controlado por un modelo similar a él. Es decir, estaremos ante dos modelos, uno será el que envíe las órdenes y otro el que las ejecutará. Obtendremos un diseño ya creado del modelo que ejecuta las acciones con licencia OpenSource y diseñaremos con programas CAD el brazo que emite las órdenes.

Queremos que nuestro brazo robótico se adecúe al entorno del laboratorio dónde llevaremos el proyecto a cabo, fácil de manejar y trabajar, de este modo nos quitamos pesadas tareas de transporte y manipulación. Haciendo modelos pequeños y, por consiguiente, más ligeros evitamos utilizar servomotores potentes y caros. Buscaremos modelos OpenSource imprimibles en impresoras 3D y para ello recurriremos a la página *Thingiverse*. El modelo que busquemos debe estar dividido en 5 partes principales: base, hombro, codo, muñeca y pinza. Los motores que formen las articulaciones entre las diferentes partes del brazo serán servomotres *Tower Pro SG90 9g Micro Servo* (la ficha técnica será adjuntada en los anexos) y las partes del modelo tienen que estar adecuadas para instalar este modelo de servo. Para controlar los servomotores utilizaremos potenciómetros rotativos de entre 10 y 20kΩ. Hemos decidido las resistencias variables como elemento de control en lugar de encoders ya que son mucho más baratos y fáciles de programar. Haremos uso de la plataforma *Arduino* para programar y controlar todos los aspectos del brazo robótico, ya sea para recibir las señales de los potenciómetros como transformarlas con el programa adecuado y enviar órdenes a los servomotores.



Imagen 40. Micro servo SG 90.

Teniendo todo lo anteriormente dicho en cuenta podemos resumir que:

- El brazo se imprimirá en 3D con una impresora por deposición de material fundido proporcionada por la escuela.
- La obtención del modelo que contendrá los servomotores será a través de la página *Thingiverse* que posee licencia libre.
- Los modelos no deben ser excesivamente grande para evitar problemas de transporte y construcción.
- Nuestro brazo se compondrá de 5 partes principales. Esto se traduce a 4 articulaciones giratorias y una pinza para el agarre de objetos.
- Los modelos obtenidos deberán soportar el montaje de los servomotores “Micro servo SG90”.
- Como elementos de control utilizaremos potenciómetros.

- Diseñaremos el brazo que emite las órdenes de tal forma que podamos acoplarle las resistencias variables.
- El control y gestión de los distintos elementos que conformarán nuestro sistema se llevará a cabo mediante la plataforma *Arduino*. Más en detalle, con la placa *Arduino UNO*.

3.2 Montaje del manipulador y del brazo gobernante.

3.2.1 Manipulador gobernado por servomotores.

Habiendo acotado ya nuestro proyecto nos disponemos a la obtención de la estructura del manipulador que albergará los servos.

Comenzaremos por la descarga del modelo 3D que se podrán obtener en la página *Thingiverse* a través del siguiente enlace: <https://www.thingiverse.com/thing:34829>. En una primera impresión de un par de piezas observamos que las piezas impresas no correspondían con las medidas que el autor proporcionaba y más tarde encontramos que el fallo residía en la forma en la que al autor había exportado los archivos .stl. Indagando por la web, nos encontramos con un segundo autor que había recogido los archivos del enlace anteriormente mostrado y había mejorado su impresión 3D, unificando las unidades de medida en milímetros y proponiendo nuevos soportes para su correcta impresión. Sabiendo esto, haremos uso de estos segundos archivos para evitar malgastar más material de impresión. La dirección de descarga es: <https://www.thingiverse.com/thing:65081>. El autor propone un pequeño brazo robótico de 4 ejes más el agarre. Gobernado por 5 servomotores *Tower Pro SG90 9g Micro Servo*. Se decanta por un programa en *Arduino* controlado a remoto por un mando.

Lo que nosotros realmente queremos es el modelo del brazo robótico que utiliza ya que cumple con las necesidades que hemos definido anteriormente. La Imagen 41 nos muestra el modelo 3D donde se puede apreciar claramente la disposición de los 5 servos. Nombraremos los servos en orden ascendente por números, llamándose el de la base *Servo1*, el siguiente *Servo2*, y así sucesivamente.

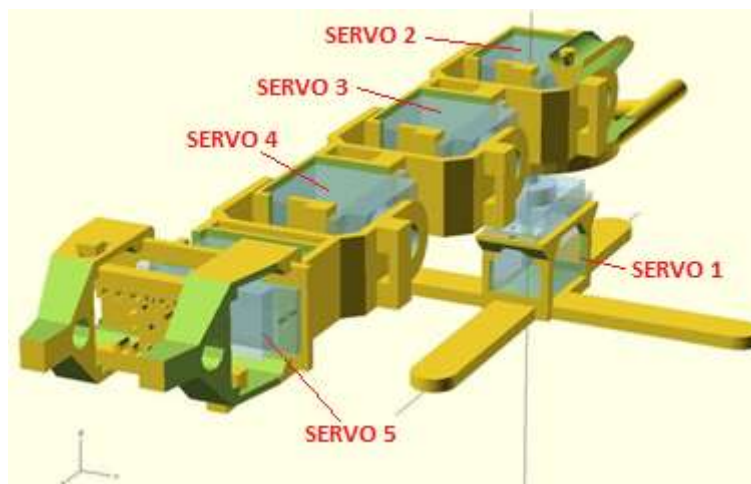


Imagen 41. Modelo brazo robótico 3D.

El archivo descargado contiene 9 ficheros diferentes que se muestran en la Imagen 42. Cada uno tendrá que ser impreso una sola vez excepto los llamados *Gripper.stl* y *GripperRack.stl*, que tendrán que imprimirse dos copias.

- Arm.stl
- ArmFront.stl
- ArmSpring.stl
- Foot.stl
- FrontPlate.stl
- Gripper.stl
- GripperGear.stl
- GripperRack.stl
- RoboArm.scad
- Swing.stl

Imagen 42.
Archivos
descargados.

El departamento de impresión de la escuela nos recomienda imprimir las piezas con las opciones por defecto del programa para agilizar la impresión ya que no van a ser expuestas a grandes esfuerzos. Como material de impresión se utilizará plástico ABS adecuado para impresoras en 3D. El tiempo total de impresión de las 11 piezas (recordamos que dos de ellas había que imprimirlas dos veces) es de aproximadamente 15 horas y media. Continuamos con la fase de post procesado donde antes de nada las piezas se separan de la cama adhesiva que genera la impresora 3D para evitar el efecto “Warping”. Procederemos a eliminar los soportes y lijar las zonas donde haya quedado resto de material indeseado.

Ya lo tenemos todo listo para comenzar el montaje de las diferentes partes del brazo.

3.2.1.1 Unión entre los módulos.

El objetivo principal es conseguir que el servo se mantenga inmóvil mientras que la pieza que está unida al eje se mueva. Para conseguirlo utilizaremos el mismo método de unión entre los módulos: introduciremos el servo en su correspondiente hueco, lo atornillaremos al módulo que hará de parte fija, colocaremos el módulo siguiente alineado con el eje del servomotor en su posición de giro y finalmente colocaremos la corona del servo. Se puede ver un ejemplo claro de cómo quedarán las uniones entre módulos en la Imagen 43, el objeto de color verde representa el servomotor, el objeto azul representa la corona del servo y los objetos amarillos los distintos módulos del brazo.

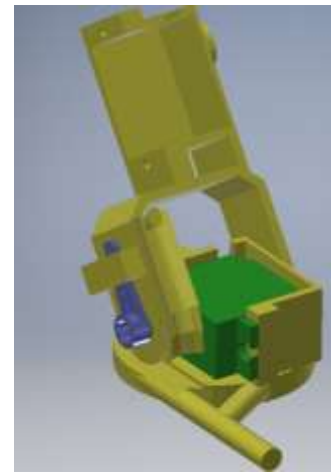


Imagen 43. Ejemplo de unión entre módulos.

Mediante esta técnica nos aseguramos de que el servo quede fijo en el su hueco y conseguimos transmitir el movimiento a través de la corona que va introducida en la ranura del módulo siguiente. Esta técnica es idónea pues prácticamente no se necesita más material del que viene cuando compramos los servos, dos tornillos y la corona.

3.2.1.2 Módulo Base.

Este módulo está formado por dos partes: la base fija y la base rotativa. La base fija tiene como función soportar el peso total de la estructura y dar estabilidad al brazo, por ello cuenta con un apoyo en cruz amplio para evitar que vuelque cuando se extienda totalmente. Está formada por una única pieza sólida y aloja en su centro el *Servo1* que será el encargado orientar la dirección a la que apunta el brazo. En segundo lugar, la base rotativa es la que está

directamente unida con el *Servo1* y sufrirá los esfuerzos de giro de dicho motor. Alojará el segundo servo del sistema, el *Servo2*, que se encargará del cabeceo principal del brazo.

Para montar correctamente las piezas comenzaremos por introducir el *Servo1* en su alojamiento atornillándolo (Imagen 45). Paralelamente a esto, atornillamos el *Servo2* en el alojamiento de la base giratoria (Imagen 46). Después atornillamos la corona en cruz al servo y con una pistola de cola caliente, aplicamos un poco de adhesivo en la superficie de la corona y la pegamos en la parte inferior de la base giratoria obteniendo un resultado similar al de la Imagen 47.



Imagen 44. Corona en cruz.

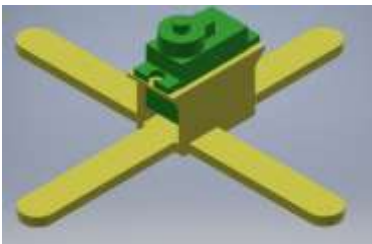


Imagen 45. Colocación del servo en la base.

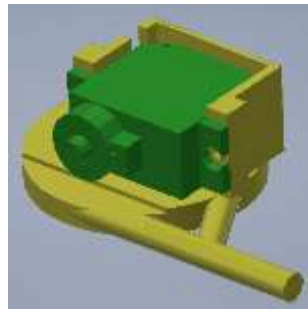


Imagen 46. Colocación del servo en la base giratoria.

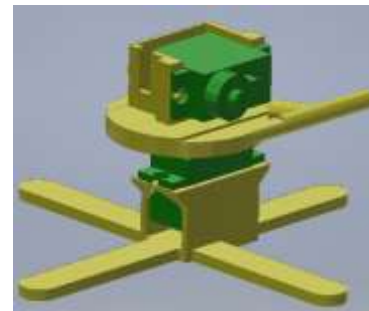


Imagen 47. Unión entre la base y base giratoria.

3.2.1.3 Módulo Brazo.

Con el montaje de este módulo ya empezamos a aumentar considerablemente la altura del manipulador. Cabe destacar la ranura que se encuentra en su parte izquierda, diseñada como guía para los cables de los servos situados aguas arriba. Otra observación interesante sobre esta pieza es la ranura que se encuentra en el exterior de la pared izquierda que sirve para alojar la corona del servo. Es una forma eficaz transmitir el movimiento del servo al módulo disminuyendo el uso de materiales.

Siguiendo el mismo procedimiento que en el módulo anterior, primero atornillaremos el *Servo3* a la pieza y posteriormente, utilizando el método de *Unión entre los módulos*, montaremos el módulo actual al *Módulo Base*. Quedándonos una apariencia como la de la Imagen 49.



Imagen 48. Módulo brazo con servo atornillado.

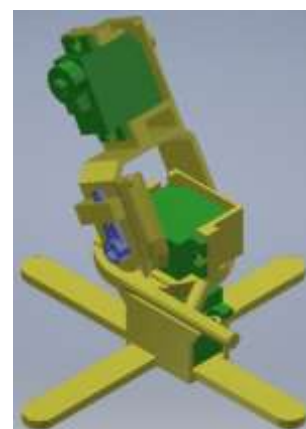


Imagen 49. Unión Módulo base y Módulo Brazo.

3.2.1.4 Módulo Antebrazo.

Continuamos con el ensamblaje del siguiente nivel de articulaciones. Este módulo será arrastrado por *Servo3* montado en el apartado anterior. No se trata de la parte más importante del brazo, pues estas son el *Módulo Base* y el *Agarre*, pero si imprescindible para el ensamblaje del brazo. A estas alturas nos habremos percatado que poco a poco el modelo va cogiendo peso, por ello debemos de ser muy cautelosos a la hora de manipularlo. Hay que hacerlo suave y sin movimientos bruscos para evitar daños en las uniones, especialmente en la unión de la base con la base giratoria del *Módulo Base*. Esta está unida con cola y si le propinamos un golpe lo suficientemente fuerte la despegaremos y habrá que adherirla nuevamente.

Esta pieza se diferencia de la del *Módulo Brazo* en que desaparece la guía para los cables, por lo demás conserva la misma forma y medidas. Seguimos utilizando el método de unión explicado, atornillamos el servo correspondiente en su respectivo lugar (en este caso se trata del *Servo4*) y seguidamente acoplamos el *Módulo Antebrazo* al *Módulo Brazo*.



Imagen 50. Módulo antebrazo.

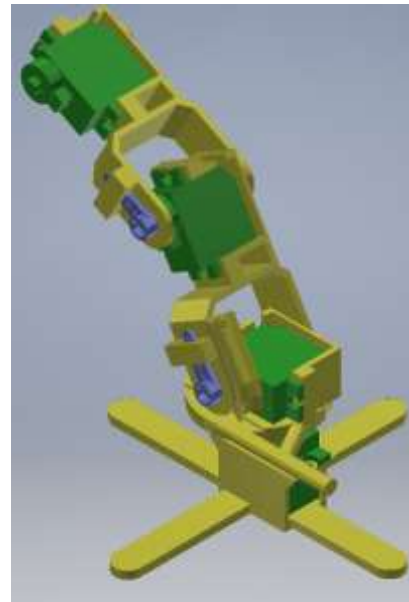


Imagen 51. Conjunto Módulo Base, Módulo Brazo y Módulo antebrazo.

3.2.1.5 Módulo Muñeca.

Estamos ante el montaje del penúltimo módulo que contendrá el *Servo5* encargado de abrir y cerrar el agarre. Este módulo es más corto que los anteriores y se compone de dos partes: la placa frontal y el cuerpo sólido. La placa frontal nos servirá como guía para que las partes del *Módulo Agarre* que se abren y cierran. Y el cuerpo sólido será el que contenga el servomotor atornillado.

A la hora de atornillar el servo en el cuerpo tenemos que introducirlo primeramente en la placa frontal, después atornillar el conjunto y nos quedará de la misma forma que muestra la Imagen 53.

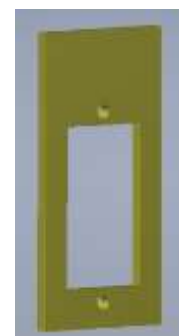


Imagen 52. Placa frontal.



Imagen 53. Módulo muñeca ensamblado.

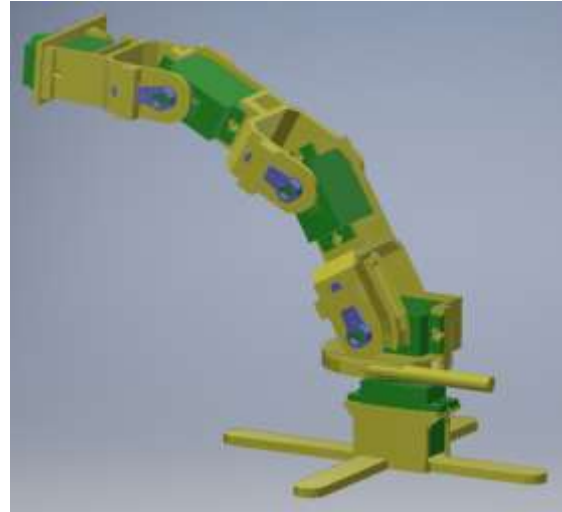


Imagen 54. Placa frontal.

Al observar detenidamente los módulos montados hasta ahora vemos una pequeña ranura en el alojamiento de los servos cuya función es la de dejar paso a los cables de los servos y es vital para su ensamblaje. Sin ella no seríamos capaces de colocar los servos y habría que buscar otras alternativas que dificultarían su montaje.

3.2.1.6 Módulo Agarre.

Finalizamos el montaje de este brazo con el ensamblaje de uno de los módulos más importantes del sistema, el agarre. Este módulo se encargará de agarrar los objetos por poder desplazarlos de lugar. Utilizando el sistema de piñón-cremallera logramos la apertura y el cierre de la pinza. Consta de 5 piezas: el engranaje, las dos cremalleras y las dos partes móviles que serán las que abran y cierren.

En primer lugar pegaremos con cola caliente una de las dos cremalleras a una de las dos partes móviles como se muestra en la Imagen 55. Repetimos el mismo proceso con la cremallera y la parte móvil restante. Mientras esperamos a que se solidifique la cola atornillaremos el engranaje al eje del *Servo5*. Cuando nos hayamos asegurado que la cremallera y la parte móvil poseen una unión sólida nos disponemos a introducir cada una por un extremo de la guía montada en el *Módulo Muñeca* (ver Imagen 56).

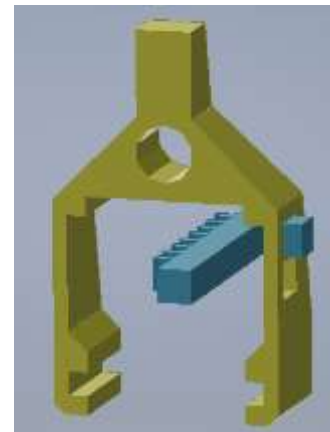


Imagen 55. Unión entre la parte móvil y la cremallera.

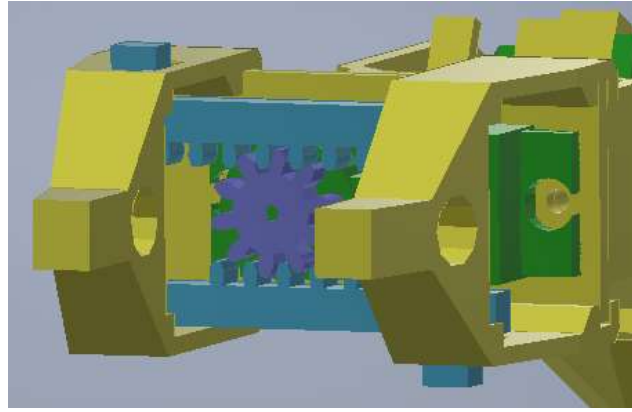


Imagen 56. Módulo agarre finalizado.

3.2.1.7 Módulos ensamblados.

El manipulador robótico tendrá un aspecto similar al siguiente:

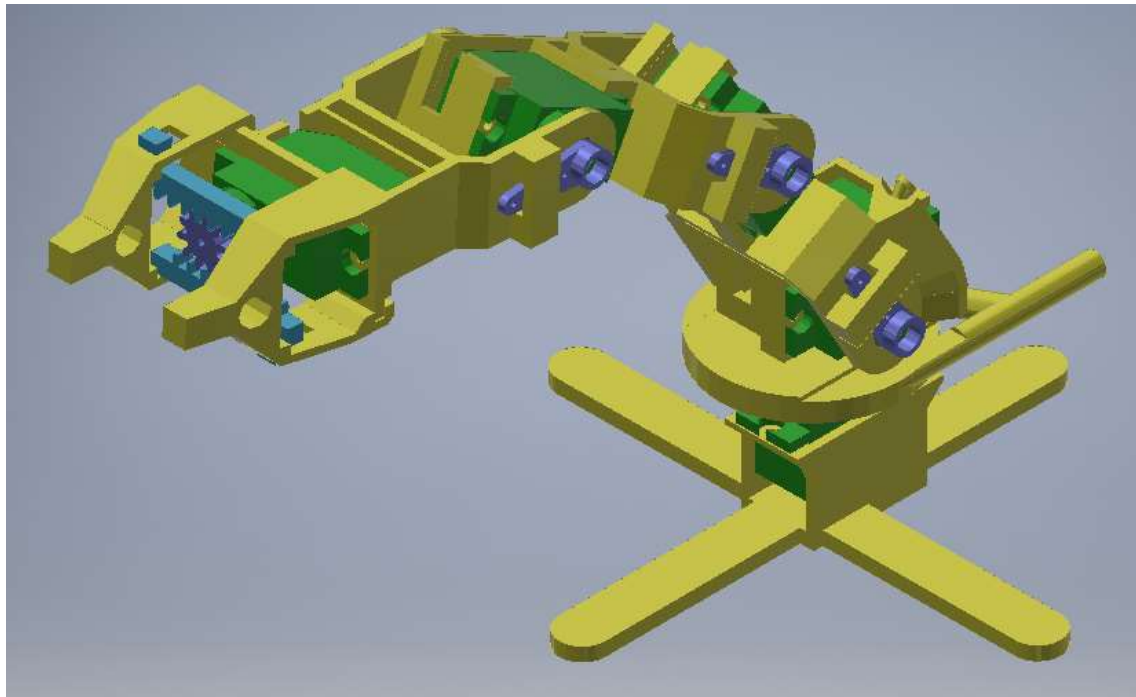


Imagen 56. Módulo agarre finalizado.

3.2.2 Diseño del brazo que controla los servos.

Para poder controlar el brazo que hemos montado anteriormente necesitamos un modelo que nos envíe las señales de control de los servomotores. El método de control se basará en el diseño de un brazo de dimensiones similares al modelo 3D descargado. Comenzaremos por tomar la medida de las alturas de cada módulo montado, a fin y al cabo las alturas nos indican el alcance que el brazo tendrá y es la única dimensión que importa en el diseño de este. A su vez tomaremos medidas del potenciómetro que usaremos para poder ajustar los puntos donde los acoplaremos.

Debemos mantener la morfología del conjunto, es decir:

- Contará con 5 potenciómetros, uno por cada servomotor.
- Su estructura estará formada por 5 sólidos que representarán los 5 módulos que posee el modelo construido.

3.2.2.1 Unión entre piezas.

El método que usaremos para ensamblar los potenciómetros al modelo consistirá en uno muy similar al empleado con los servos salvo que no utilizaremos tornillos para su fijación. En este caso utilizaremos cola caliente para unir la base del potenciómetro a una de las partes y el vástago se introducirá por unos orificios en la segunda pieza a unir. Utilizaremos cola caliente para fijar el vástago a esta. De esta forma permitimos el giro de ambas piezas entre ellas.

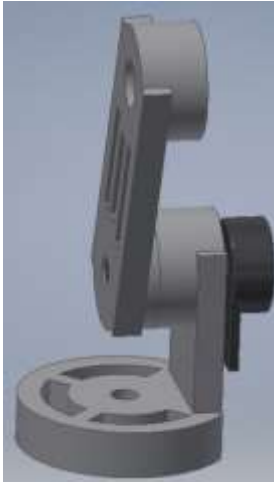


Imagen 56. Unión Potenciómetro2.

Para garantizar una unión sólida entre las articulaciones hemos diseñado un encaje macho-hembra muy parecido al sistema de los puzles. Una de las piezas tiene un desnivel de forma circular (hembra) y la otra contiene un saliente de igual medida y forma (macho) que se acoplan formando la unión.

3.2.2.2 Base.

Conservaremos el diseño en cruz de la base que es vital para la estabilidad del modelo. En el centro de ella habilitaremos el punto de montaje del *Potenciómetro1*, encargado de transmitir las órdenes al *Servo1*. Durante el proceso de diseño hemos intentado evitar el uso excesivo de material de impresión creando orificios pasantes.

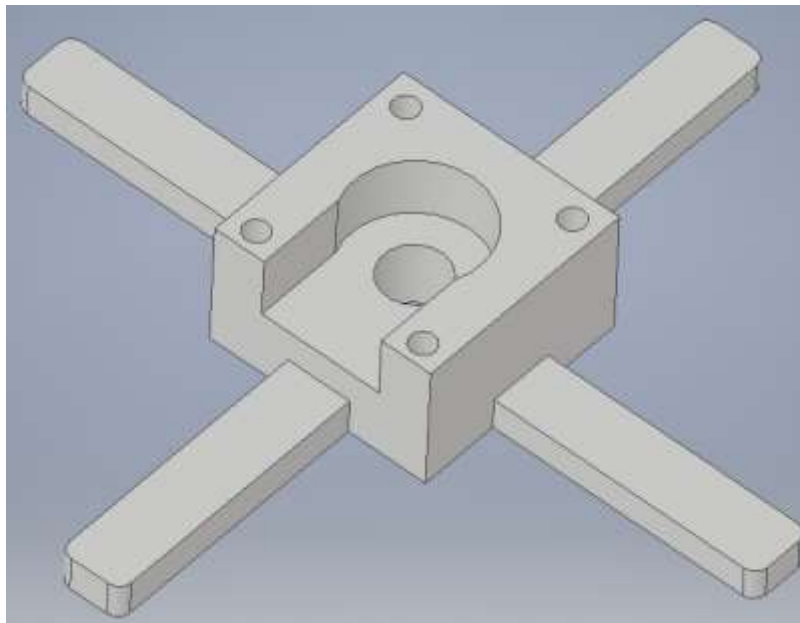


Imagen 57. Base del Potenciómetro1.

3.2.2.3 Base giratoria.

La base giratoria contiene ya dos puntos de acople para los potenciómetros. Uno de ellos se encuentra en la cara inferior que corresponde al acople del vástago de la resistencia variable de la base y la otra se encuentra en la cara externa del pilar donde irá acoplado el *Potenciómetro2*. Este pilar saliente de la base está preparado para que la pieza que le precede encaje perfectamente. Antes de introducir el vástago del *Potenciómetro1* a través de la base giratoria nos aseguraremos de fijar el *Potenciómetro2* al pilar (Imagen 58). El resultado final se muestra en la Imagen 59.

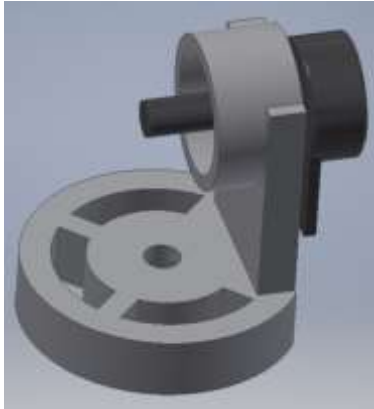


Imagen 58. Base giratoria con el potenciómetro acoplado.

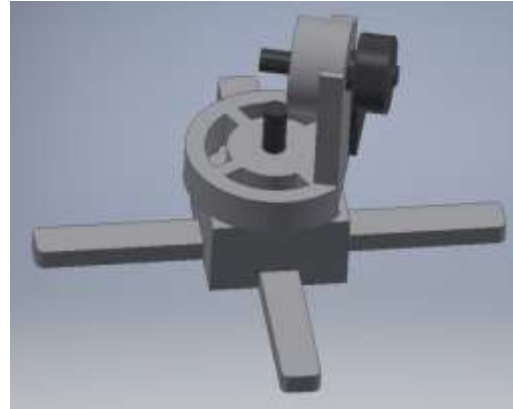


Imagen 59. Ensamblaje de la Base con la Base giratoria.

Hemos de asegurarnos que aplicamos la cola adecuadamente, un excesivo uso de esta podría dar pie a la unión del vástago del potenciómetro con la misma pieza que lo aloja, evitando la rotación de este.

3.2.2.4 Brazo y antebrazo.

Hemos decidido incluir las dos partes en el mismo punto porque no existe variación alguna en el modelo creado. Sus dimensiones y diseño son los mismos a contrario que en el brazo de los servomotores que una de ellas incluía una guía para los cables. Al igual que en los diseños anteriores, existen zonas huecas para el ahorro de material. Estas piezas muestran claramente el método macho-hembra que hemos utilizado para unir las articulaciones. Un extremo hace de macho mientras que el contrario hace de hembra.

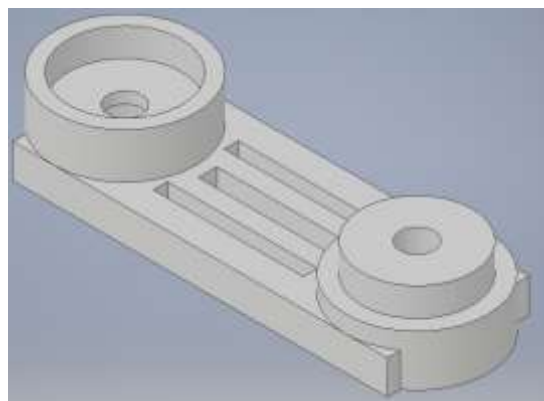


Imagen 60. Diseño común para las piezas Brazo y Antebrazo donde se observa el método de unión macho-hembra.

Hacemos uso de la cola caliente para fijar los potenciómetros que se ubicarán en la parte exterior de las piezas, introduciendo el vástago por el orificio de la unión hembra. Esperamos un tiempo a que la cola se solidifique y ensamblamos el *Brazo* con el *Antebrazo* (Imagen 61) y posteriormente los acoplamos al conjunto de la *Base giratoria* con la *Base* (Imagen 62). Finalmente encolamos los vástagos a su pieza correspondiente.

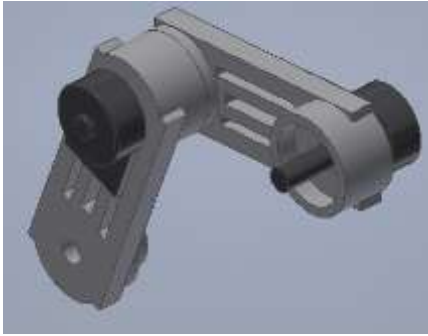


Imagen 61. Acople del Brazo con el Antebrazo y colocación del Potenciómetros3 y Potenciómetro4.



Imagen 62. Conjunto de las partes Base, Base giratoria, Brazo y Antebrazo.

3.2.2.5 Muñeca y agarre.

Finalizaremos el montaje de este segundo brazo con el montaje del agarre y la articulación de la muñeca. Solo quedará introducir el vástago del *Potenciómetro4* por el orificio de esta pieza aplicando cola caliente entre el vástago y esta.

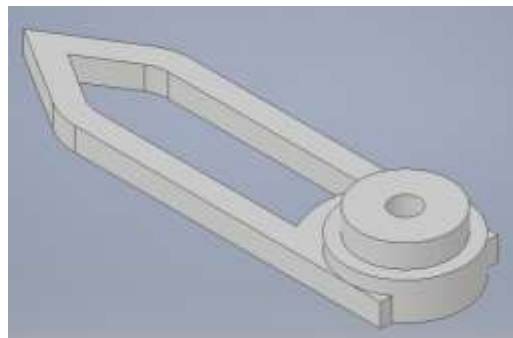


Imagen 63. Pieza del Agarre.

El *Potenciómetro5* que controla el *Servo5* encargado de abrir y cerrar el agarre no se incluirá en el modelo para facilitar su acceso y manipulación.

3.2.2.6 *Diseño final del brazo.*

Finalmente obtendremos el modelo mostrado en la Imagen 64. Dejamos que transcurran un par de horas para asegurarnos que la cola caliente se ha asentado adecuadamente y comprobamos que todas las articulaciones giran correctamente. De no ser así cabría la posibilidad que parte de la cola utilizada en fijar el potenciómetro se haya adentrado en el orificio del vástago, impidiendo la rotación. En este caso habría que desmontar dicha unión, eliminando todo resto de cola caliente y procediendo de nuevo al ensamblaje.

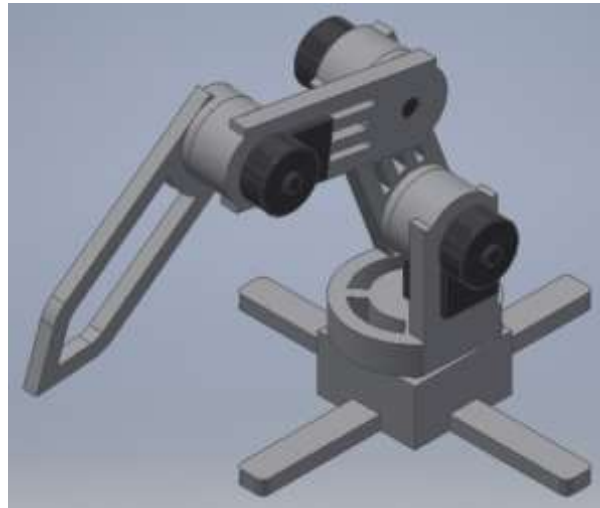


Imagen 64. Diseño final del brazo ensamblado.

3.3 Programación del sistema de control de los servos y montaje del circuito electrónico.

A estas alturas tendremos en nuestras manos los dos brazos robóticos: el manipulador que ejecuta el trabajo (Imagen 65) y el brazo gobernante (Imagen 66).

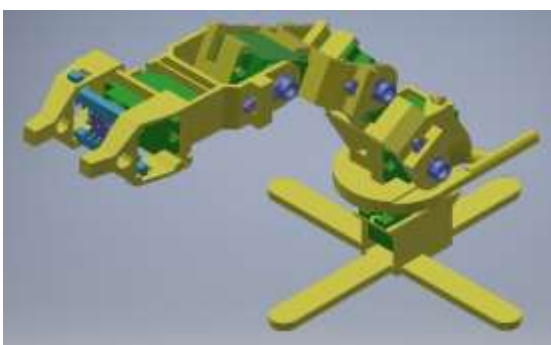


Imagen 65. Manipulador robótico.

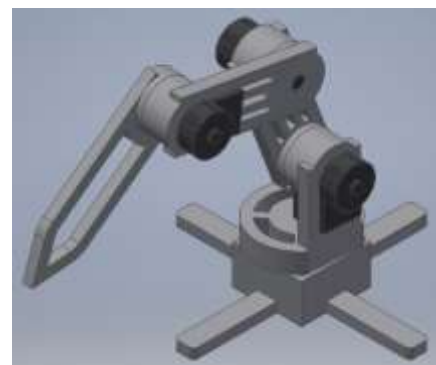


Imagen 66. Brazo gobernante.

Estos brazos no poseen comunicación alguna entre sí de manera que nuestra siguiente tarea será la de crear un sistema de control y de comunicación entre ambos. Para lograrlo utilizaremos la placa *Arduino UNO* proporcionada por la empresa *Arduino*.

Dicha plataforma cuenta con un software propio para la programación del microcontrolador que podemos adquirir gratuitamente desde su página web. Este software está basado en el lenguaje de programación C++. La estructura de programación en *Arduino* se divide en dos partes: *setup* y *loop*. La primera se encarga de la preparación del programa, se incluyen las configuraciones de los pines. La función *loop* incluye el código que se ejecutará indefinidamente, leyendo las entradas y salidas e interpretándolas. Este software de programación incorpora lo que llamamos librerías que poseen líneas de código que facilitan y simplifican la programación. En el presente proyecto hacemos uso de los servomotores y para su correcta programación utilizaremos una librería específica que nos ayudará a su manejo.

3.3.1.1 Aplicación de control.

La función principal del programa será la de controlar los servos con los potenciómetros y para ello transformaremos la lectura de analógica que reciba la placa de estos en la posición correspondiente de los motores. El método de lectura de potenciómetros se basará en promediar la lectura de 5 muestras consecutivas para reducir el ruido y las pequeñas variaciones de las medidas instantáneas. Seguidamente haremos una interpolación lineal de los valores mínimos y máximos que irán ligados al ángulo mayor y menor del servo. Dividiremos la explicación del programa en 3 apartados distintos: declaración de variables, función *setup* y función *loop*.

3.3.1.2 Declaración de variables.

Lo primero de todo será incluir la librería para trabajar con servomotores (*Servo.h*) seguido de la declaración de estos nombrándolos como ya hemos comentado: *Servo1*, *Servo2*... *Servo5*. Seguidamente se declaran los vectores que recogerán 5 muestras de cada potenciómetro para poder aplicar el filtro que hemos explicado y las variables necesarias para utilizar los bucles *for*.

```
#include <Servo.h>

Servo servo1, servo2, servo3, servo4, servo5;

int lectura1[5];
int lectura2[5];
int lectura3[5];
int lectura4[5];
int lectura5[5];

int i = 0;
int suma1, suma2, suma3, suma4, suma5;
int promedio1=0, promedio2=0, promedio3=0, promedio4=0, promedio5=0;
int salida1=0, salida2=0, salida3=0, salida4=0, salida5=0;

int N = 5;
```

Imagen 67. Declaración de variables.

3.3.1.3 Función Setup.

Damos paso a la función que se ejecutará una sola vez y será al inicio del programa. Aquí inicializamos todos los valores de los vectores a 0 para eliminar posibles datos residuales. Acto seguido, asignamos a cada servo su pin de control por donde recibirán las órdenes de la placa.

Hemos escogido estos pines debido a que tienen habilitada la salida PWM. Como se explicó en el apartado *Servomotores*, este tipo de elementos necesitan una señal modulada para poder controlarlos. Los pines digitales con señal PWM habilitada vendrán acompañados del siguiente símbolo: ~.

3.3.1.4 Función Loop.

Se encarga de ejecutar las acciones constantemente. Aquí encontraremos la mayoría de las líneas de programación. Nuestra primera tarea será la de asignar a cada posición del vector una lectura del potenciómetro (Imagen 69), en este caso hemos optado por realizar 5 medidas. Utilizaremos el comando *analogRead()* para leer los valores del potenciómetro. La asignación de pines a los potenciómetros será la siguiente:

- Potenciómetro1 – A0
- Potenciómetro2 – A1
- Potenciómetro3 – A2
- Potenciómetro4 – A3
- Potenciómetro5 – A5

Observamos que el quinto potenciómetro no sigue el orden de la serie ya que debería estar conectado al pin analógico A4. Esto se debe a un problema que dificultaba la conexión del cable a este pin. El estar conectado al pin A5 no influye absolutamente en nada en el resultado final siempre y cuando esté reflejado en el programa.

Continuamos con el siguiente paso del programa, el promedio de las diferentes medidas. Recurrimos de nuevo al bucle *for* para sumar cada valor de los componentes del vector en una variable denominada *suma* para después hacer el promedio entre *N* (referido a la cantidad de medidas tomadas, en este caso 5).

La variable *promedio* tendrá una resolución de 10bits que será un valor comprendido entre 0 y 1023. El valor 0 significará que no hay flujo de datos por lo que el potenciómetro estará en una posición de máxima resistencia en un extremo. Por el contrario, un valor de 1023

```
void setup() {  
  
  for(i=0;i<5;i++){  
    lectura1[i] = 0;  
    lectura2[i] = 0;  
    lectura3[i] = 0;  
    lectura4[i] = 0;  
    lectura5[i] = 0;  
  }  
  
  servo1.attach(5);  
  servo2.attach(6);  
  servo3.attach(9);  
  servo4.attach(10);  
  servo5.attach(11);  
}
```

Imagen 68. Función Setup.

```
void loop() {  
  
  for(i=0;i<N-1;i++){  
    lectura1[i] = lectura1[i+1];  
    lectura2[i] = lectura2[i+1];  
    lectura3[i] = lectura3[i+1];  
    lectura4[i] = lectura4[i+1];  
    lectura5[i] = lectura5[i+1];  
  }  
  
  lectura1[N-1] = analogRead(A0);  
  lectura2[N-1] = analogRead(A1);  
  lectura3[N-1] = analogRead(A2);  
  lectura4[N-1] = analogRead(A3);  
  lectura5[N-1] = analogRead(A5);  
  
  promedio1 = suma1 / N;  
  promedio2 = suma2 / N;  
  promedio3 = suma3 / N;  
  promedio4 = suma4 / N;  
  promedio5 = suma5 / N;  
}
```

Imagen 69. Función Loop, lectura de señales analógicas.

```
promedio1 = suma1 / N;  
promedio2 = suma2 / N;  
promedio3 = suma3 / N;  
promedio4 = suma4 / N;  
promedio5 = suma5 / N;
```

Imagen 70. Promedio de las medidas.

corresponderá al extremo opuesto del potenciómetro con un valor de resistencia de 0Ω . Ahora debemos traducir ese valor de datos en un ángulo de posición del servomotor. Para ello usaremos la función *map()* que nos permite convertir un rango de variación en otro. Nuestro rango de lectura hemos comentado que está comprendido entre 0 y 1023 y el rango de giro de los servos varía de 0 a 180 grados. Como 0 también es un valor, el rango de acción del servomotor que escribiremos será de 0 a 179. Dado que los potenciómetros tienen más de 180 grados entre su valor mínimo y máximo de resistencia deberemos ajustar esta diferencia de rango de valores para que las posiciones angulares de ambos coincidan y que a la vista estén en la misma posición. Sin el ajuste necesario, el valor de mínima resistencia coincidiría con 0° y el de máxima con 180° (Imagen 71). La Imagen 72 muestra el rango de giro después de aplicar el ajuste. Cada potenciómetro se ajustará dependiendo de la articulación que represente.

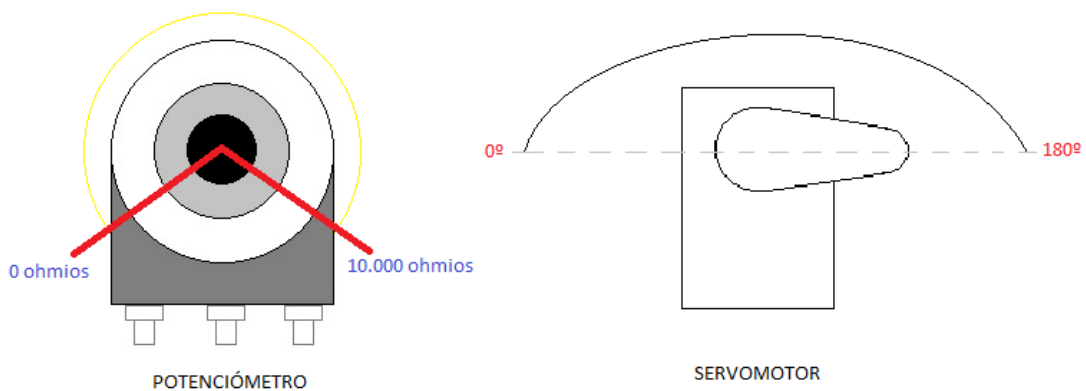


Imagen 71. Diferencia entre rangos.

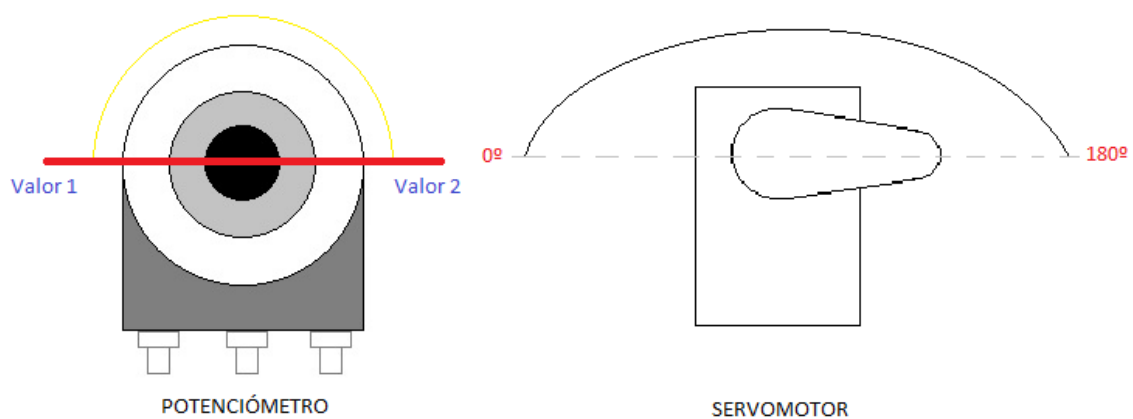


Imagen 72. Ajuste de rango de valores entre la resistencia y el ángulo de giro.

Una vez realizado el ajuste de cada uno de los 5 potenciómetros nos quedará un código como el siguiente mostrado en la Imagen 73.

```
salida1 = map(promedio1, 1023, 380, 179, 0);  
salida2 = map(promedio2, 900, 450, 80, 150);  
salida3 = map(promedio3, 891, 450, 24, 102);  
salida4 = map(promedio4, 891, 450, 24, 102);  
salida5 = map(promedio5, 1023, 0, 179, 0);
```

Imagen 73. Función map () con el ajuste de rangos entre los potenciómetros y los servos.

Para finalizar nuestro programa de control usaremos la función `servo.write ()` que envía los datos al pin de cada servomotor. Para evitar saturar el microprocesador y dar tiempo a que los servos se muevan a la posición deseada escribiremos una última línea de código utilizando la función `delay ()` que hará que el programa espere los milisegundos que le indiquemos.

```
servo1.write(salida1);  
servo2.write(salida2);  
servo3.write(salida3);  
servo4.write(salida4);  
servo5.write(salida5);
```

```
delay(50);
```

Imagen 74. Orden de enviar los datos de las variables “salida” a cada servo.

Con este proceso hemos conseguido crear la aplicación que nos servirá para controlar el sistema en conjunto formado por el manipulador y el brazo gobernante. El código que hemos descrito anteriormente vendrá adjuntado en el apartado *Anexos*.

3.4 Circuito electrónico.

Para concluir con el proyecto nos queda una de las fases más importantes, el montaje del circuito electrónico. Sin él, no existiría comunicaciones entre los diferentes elementos y el programa de control no tendría efecto alguno.

El montaje lo lograremos con los siguientes materiales:

- x1 Protoboard.
- x1 Microprocesador *Arduino UNO*.
- x5 Potenciómetro de 10kΩ.
- x5 Servomotores *Tower Pro SG90 / Microservo 9g*.
- x1 Fuente de alimentación de 5-6 Voltios, >1 Amperio.

Seguiremos el esquema eléctrico de la Imagen 75 para conectar todos los elementos entre sí. La imagen no representa los elementos en su localización real simplemente está colocados a modo de esquema para facilitar su comprensión. Para facilitar su transporte y solidez hemos optado por fijar y atornillar el manipulador, la placa y el brazo gobernante a una plancha de madera.

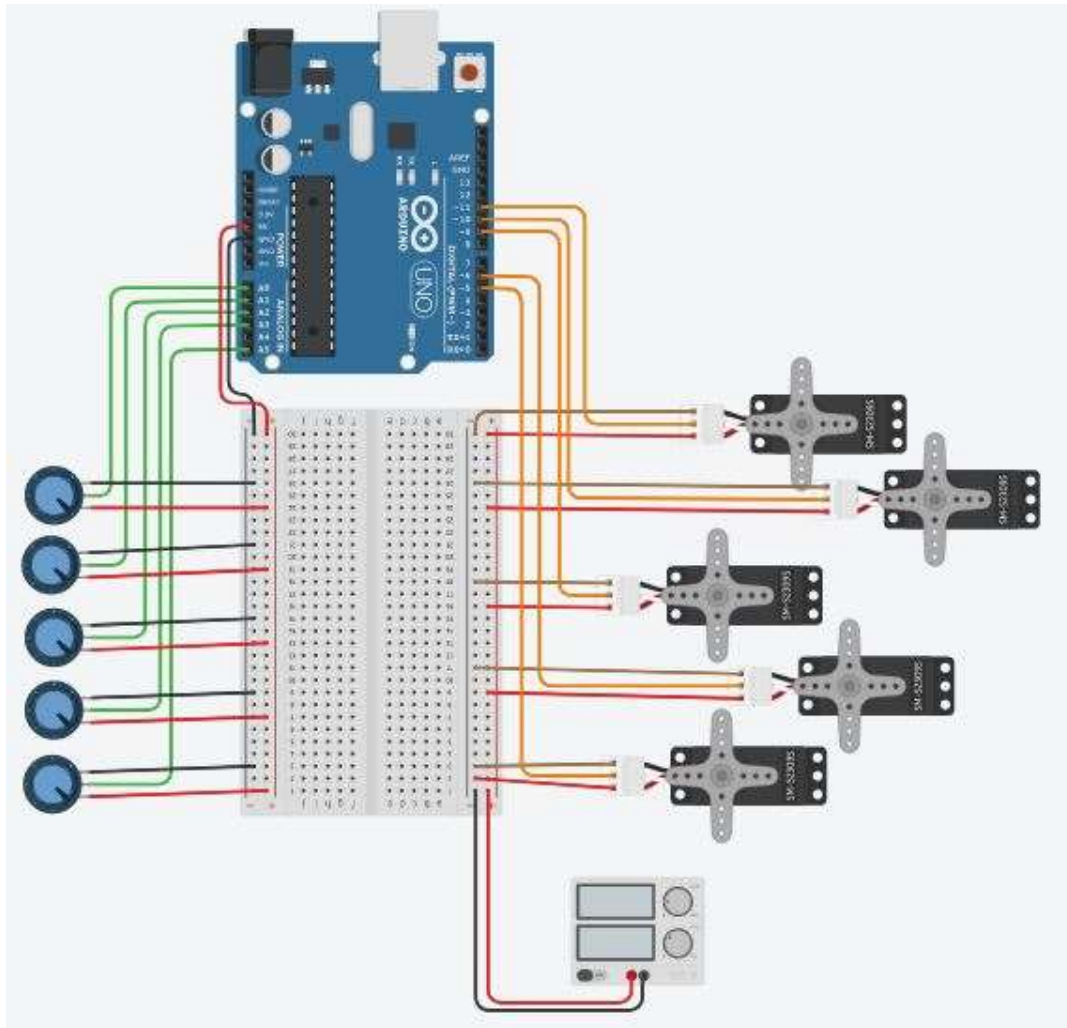


Imagen 75. Esquema de conexiones del circuito electrónico.

3.5 Resultado final.

Teniendo todos los elementos conectados como hemos explicado anteriormente, solo faltaría alimentar la placa con el USB tipo A a tipo B y conectar la fuente de alimentación. El sistema funcionará respondiendo a las ordenes emitas por los potenciómetros.

Con este proyecto hemos conseguido crear un manipulador robótico de 4 grados de libertad más el agarre. Hemos utilizado máquinas que se utilizan hoy en día como las impresoras 3D, al igual que diversos elementos que están a nuestro alrededor y no nos damos cuenta como los potenciómetros o servomotores. Hemos sido capaces de codificar un pequeño programa en un idioma de programación similar al C++. Al igual que somos capaces de diseñar objetos en CAD para su posterior impresión y pudiendo modificar ciertas variables para adecuar la impresión a nuestras necesidades.

El sistema en sí, manipulador robótico, brazo gobernante y todo el conjunto del sistema electrónico son un prototipo que podría formar parte de un proyecto mucho mayor como la construcción de brazos industriales.



3.6 Posibles mejoras del proyecto.

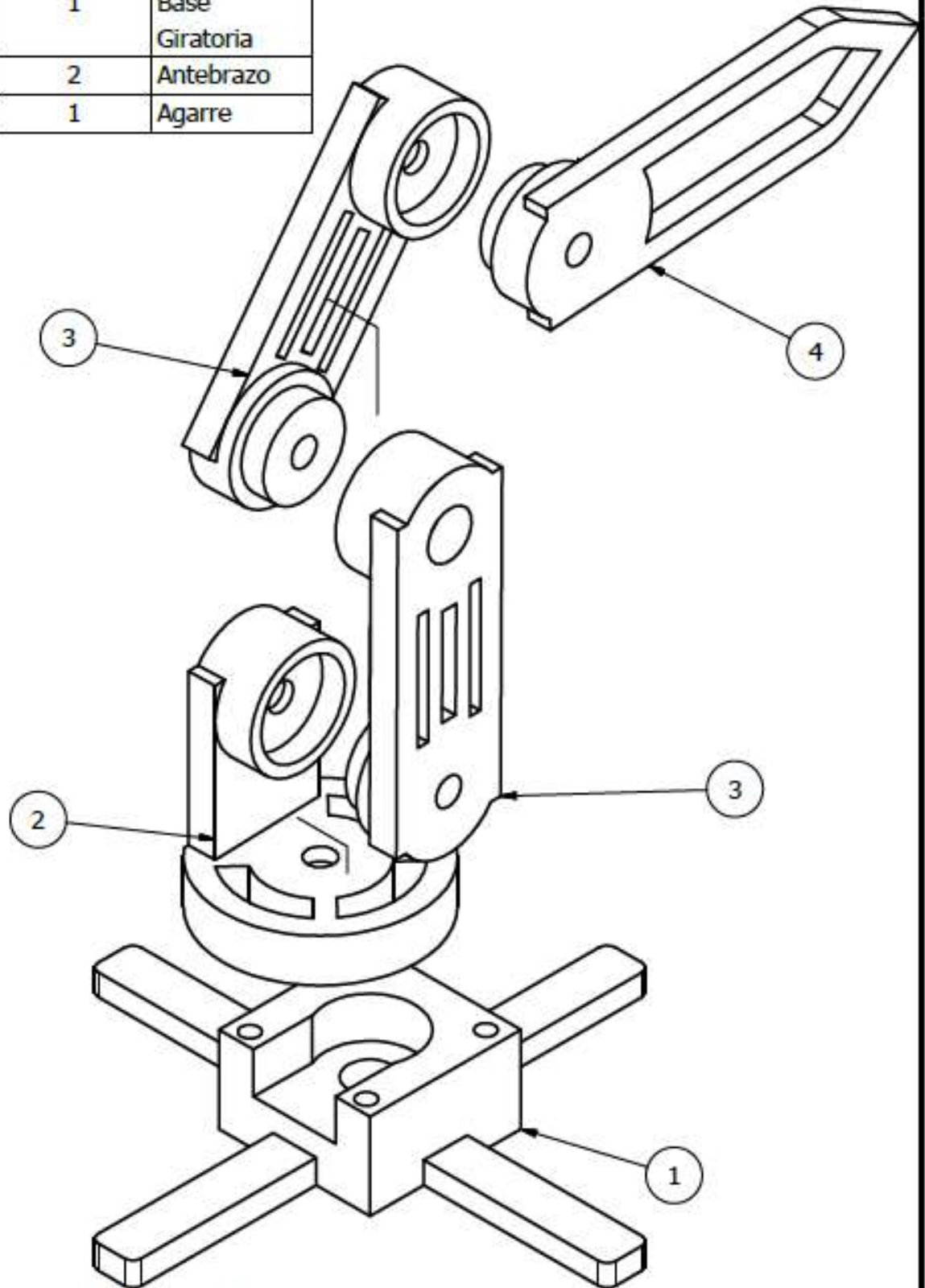
Como prácticamente todo en lo que se refiere a las máquinas y dispositivos existe una posible mejora en las prestaciones de ellos. En lo referente al actual proyecto podríamos hacer varias mejoras referentes a los servos.

El principal inconveniente que hemos observado a la hora de del movimiento de los servomotores es la escasez de par en el *Servo2*. Es el encargado del cabeceo principal del brazo y ha de soportar el peso de los 3 servos restantes. Una posible solución a este problema sería la de sustituir dicho servo por otro más potente. Otra posible mejora de servomotor se daría en el motor del agarre, para tener más potencia y evitar que se puedan escurrir los objetos transportados. También notamos en ciertas pruebas de laboratorio que existían muchas interferencias con el circuito. Llegamos a la conclusión que debido a la cantidad de cables y su longitud estos se comportaban con antena de captación de radiofrecuencias y ruido que alteraban las señales del sistema y posiblemente partes de las conexiones no fuesen lo suficientemente sólidas. Una salida a este problema podría ser la de crear un circuito impreso para deshacernos de los cables.



4. Planos.

LISTA DE ELEMENTOS		
ELEMENTO	CANTIDAD	DESCRIPCIÓN
1	1	Base
2	1	Base Giratoria
3	2	Antebrazo
4	1	Agarre

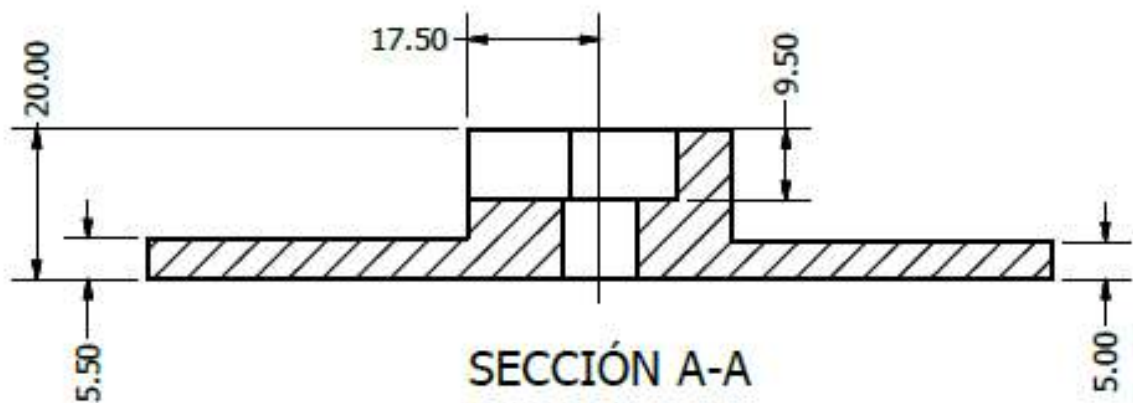
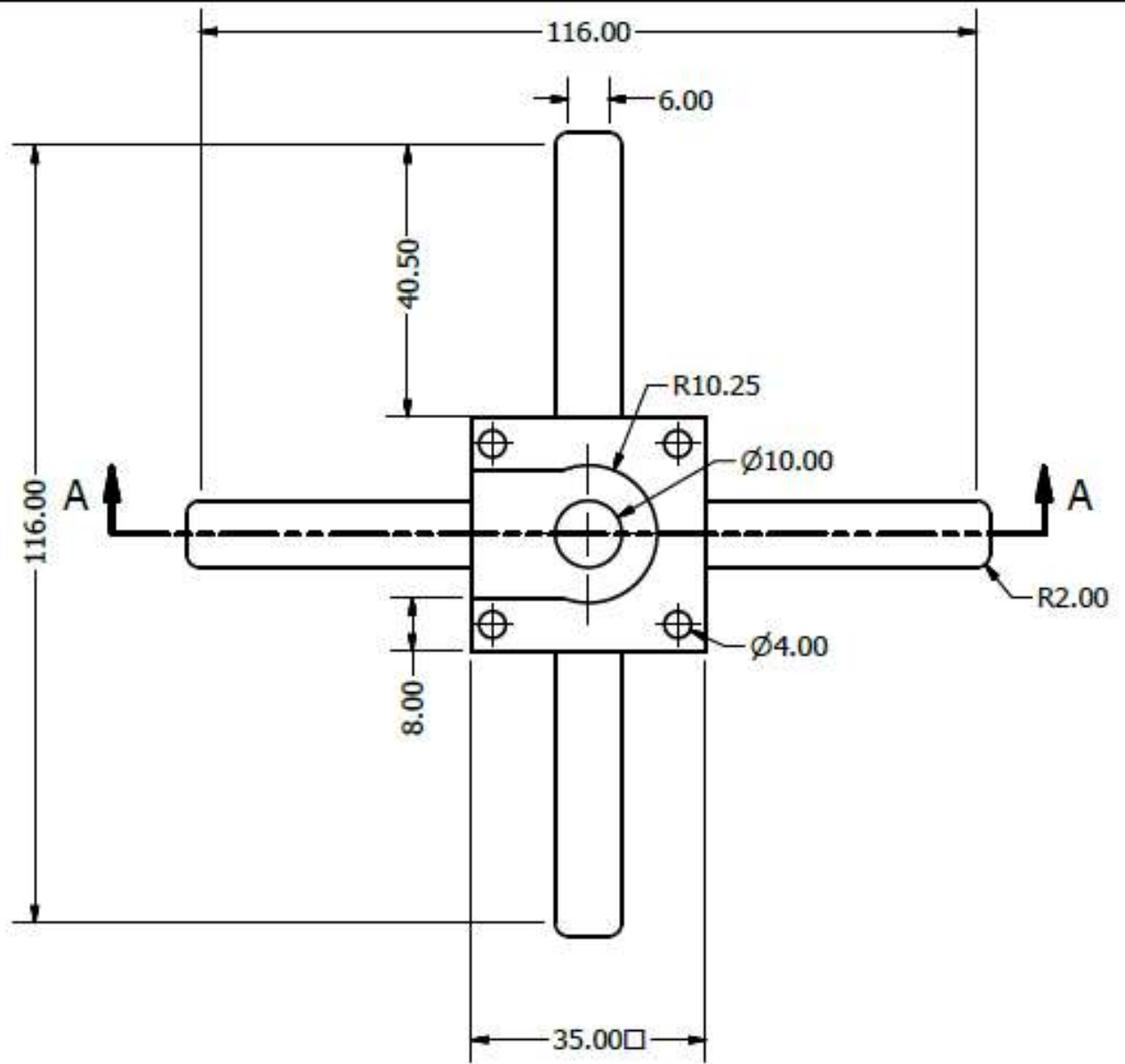


	Fecha	Nombre
Dibujado	Julio 20	Octavio Amat
Comprobado	Agosto 17	Octavio Amat


Escuela Técnica Superior de Ingeniería del Diseño

Escala: 1 / 1	Nombre del archivo: Conjunto Brazo
Unidad: mm	

PLANO Nº: 1
<i>Proyecto Final de Grado: Diseño y programación de un manipulador controlado por arduino</i>



SECCIÓN A-A
ESCALA 1 : 1

	Fecha	Nombre
Dibujado	Julio 20	Octavio Amat
Comprobado	Agosto 17	Octavio Amat



Escuela Técnica Superior de Ingeniería del Diseño

Escala:
1 / 1

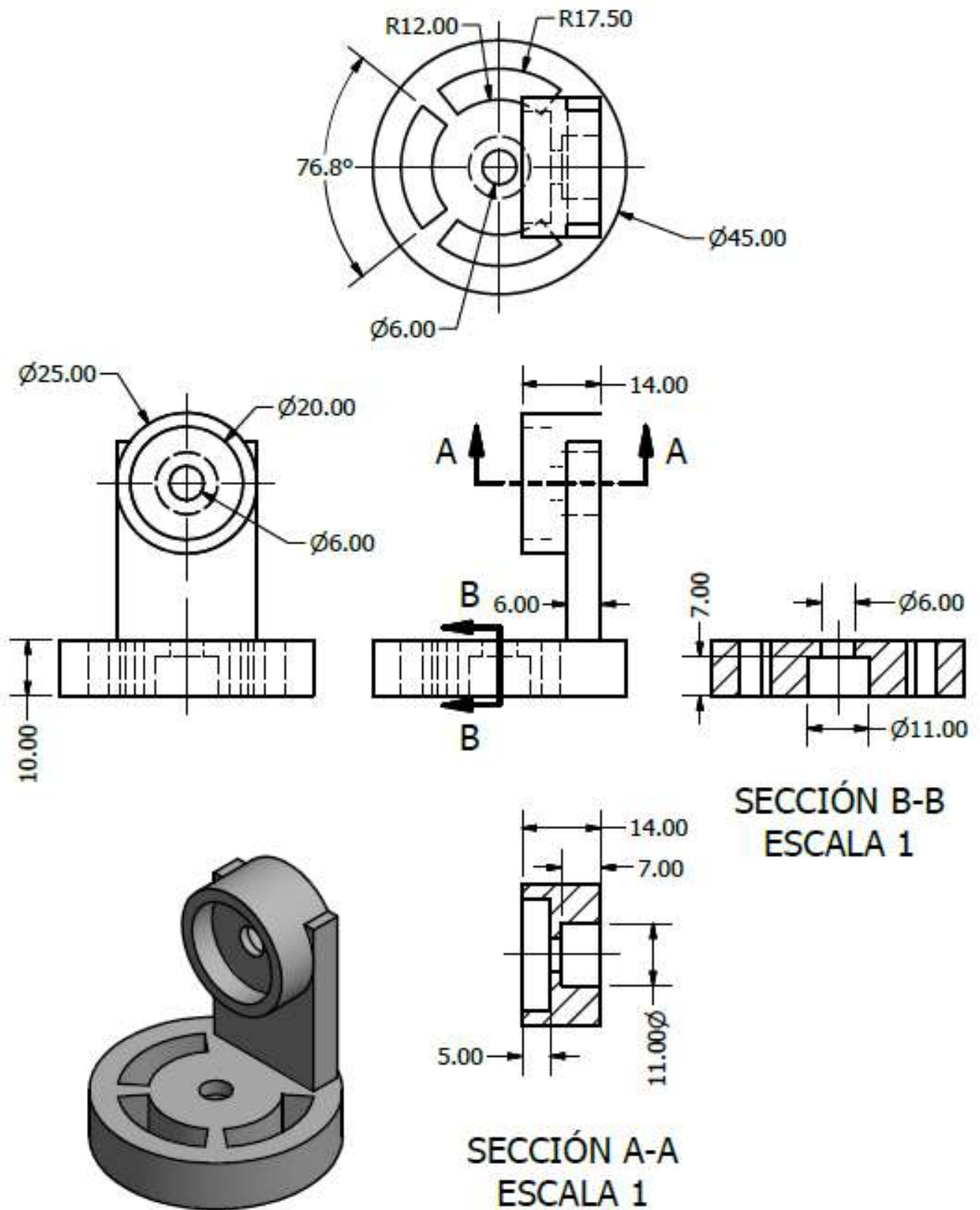
Nombre del archivo:

Base

PLANO Nº: 2

Unidad:
mm

Proyecto Final de Grado:
Diseño y programación de un manipulador controlado por arduino



SECCIÓN B-B
ESCALA 1

SECCIÓN A-A
ESCALA 1

	Fecha	Nombre
Dibujado	Julio 20	Octavio Amat
Comprobado	Agosto 17	Octavio Amat

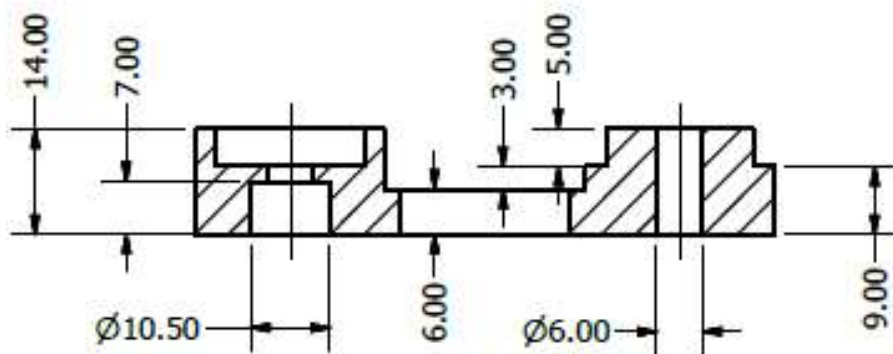
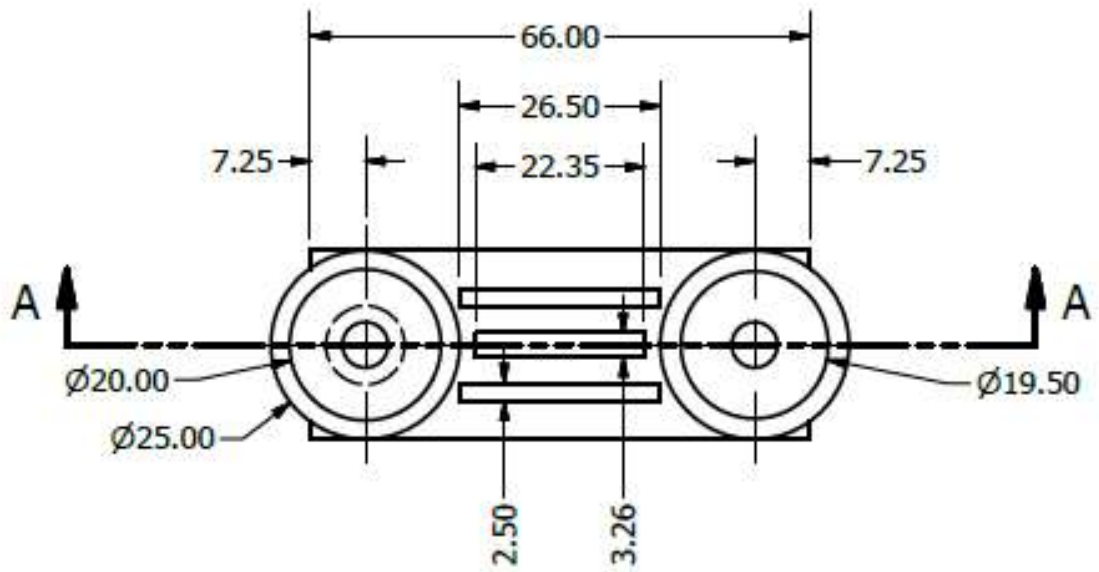


Escuela Técnica Superior de Ingeniería del Diseño

Escala: 1 / 1	Nombre del archivo: Base Giratoria
Unidad: mm	

PLANO N°: 3

Proyecto Final de Grado:
Diseño y programación de un manipulador controlado por arduino



SECCIÓN A-A
ESCALA 1 : 1

	Fecha	Nombre
Dibujado	Julio 20	Octavio Amat
Comprobado	Agosto 17	Octavio Amat


Escuela Técnica Superior de Ingeniería del Diseño

Escala:
1 / 1

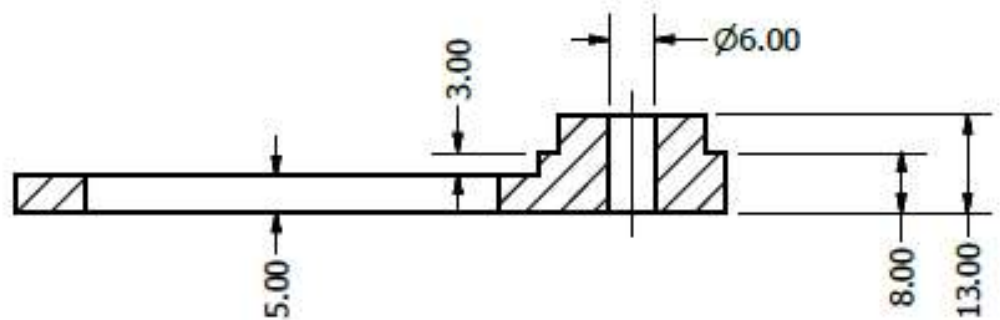
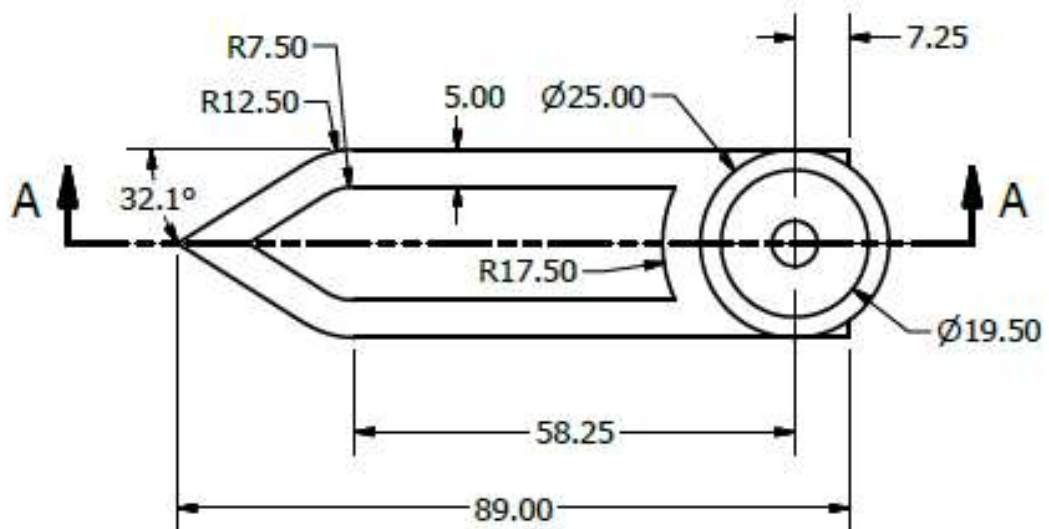
Nombre del archivo:

Antebrazo

PLANO Nº: 4

Unidad:
mm

Proyecto Final de Grado:
Diseño y programación de un manipulador controlado por arduino



SECCIÓN A-A
ESCALA 1 : 1

	Fecha	Nombre
Dibujado	Julio 20	Octavio Amat
Comprobado	Agosto 17	Octavio Amat


Escuela Técnica Superior de Ingeniería del Diseño

Escala:
1 / 1

Nombre del archivo:

Agarre

PLANO Nº: 5

Unidad:
mm

Proyecto Final de Grado:
Diseño y programación de un manipulador controlado por arduino

5. Presupuesto.

En el presente documento se especifican los costes asociados a la elaboración del presente proyecto, así como los costes de las posibles mejoras propuestas.

Desglosaremos los costes en los siguientes apartados:

- Material
- Licencia de software
- Mano de obra

El valor de los costes asociados a la compra y las licencias no llevan el IVA incluido por lo que se incluirá al final, una vez calculado el total de todos los costos.

Los materiales utilizados para el proyecto se ven reflejado en la siguiente tabla (Tabla 1) en donde apreciamos los costes por unidad y la cantidad.

Tabla 1. Coste de los materiales empleados en el proyecto.

Material	Concepto	Cantidad	Coste unitario [€/u]	Coste total [€]
Microservo Tower Peo SG90	Servomotor	5 u.	1.80	9
Placa Arduino UNO	Microprocesador	1 u.	6.05	6.05
Potenciómetro B10k 10k Lineal	Resistencia variable	5 u.	1.89	9.45
PROTOBOARD 400 puntos	Placa prototipo	1 u.	1.55	1.55
x40 Cables hembra macho 20cm, jumpers dupont 2.54	Cables	1 u.	2.85	2.85
x40 Cables macho macho 20cm, jumpers dupont 2.54	Cables	1 u.	1.82	1.82
Fuente de alimentación 5V y 2A	Fuente de alimentación	1 u.	4	4
SUBTOTAL				34.72

A continuación (Tabla 2) se muestran los costes de la licencia de software que han sido empleadas para este proyecto.

Tabla 2. Coste de las licencias de los programas utilizados.

Licencias de software	Tipo	Cantidad	Coste unitario [€/u]	Coste total [€]
Autodesk Inventor Professional 2017	Licencia de estudiante	1	314.60	314.60
SUBTOTAL				314.60

La licencia mostrada es para un mes de trabajo con el programa de la versión estudiante. Se ha necesitado de unas pequeñas nociones básicas de video tutoriales para poder entender el entorno de trabajo del programa.

En cuanto a la mano de obra, la Tabla 3 nos muestra las horas aproximadas en finalizar el proyecto y que valor tendría cada hora trabajada.

Tabla 3. Mano de obra empleada en el proyecto.

Personal	Cantidad de horas trabajadas	Coste por hora trabajada [€ / hora]	Coste total [€]
Departamento técnico	~360	8.5	3060
Departamento impresión 3D	15	8.5	127.5
SUBTOTAL			3187.5

Para finalizar, añadiremos el IVA a los costes ya mencionados y calcularemos el total de todos ellos.

Tabla 4. Total de costes.

	Coste [€]
Materiales empleados	34.72
IVA (21% s/materiales empleados)	7.3
Licencias de software	314.60
IVA (21% s/Licencias)	66.06
Mano de obra	3187.5
IVA (21% s/Mano de obra)	669.38
TOTAL	4279.56

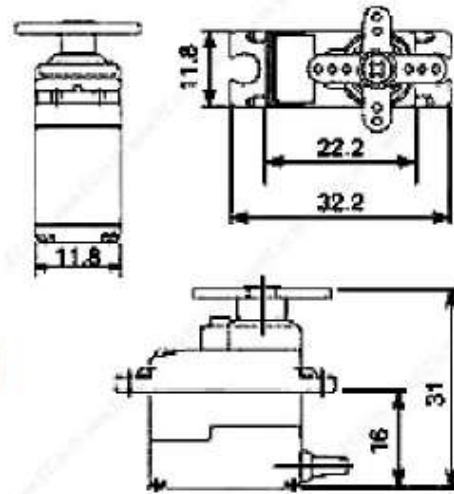
6. Referencias.

- [1] "Sitio web *CURIOSITE*", www.curiosite.com
- [2] "Sitio web *TIPOSDE*", www.tiposde.org
- [3] "Sitio web *PLATEA*", <http://platea.pntic.mec.es>
- [4] Universidad politécnica Salesiana. Francisco Aruajo, Luis Ayala, Cristhian Bermeo, "*Robots Atropomórficos*", 2014.
- [5] "Sitio web *JENNIMILY*", jennimily.wordpress.com, 2013.
- [6] "*Arduino Notebook: A Beginner's Reference*", Brian W. Evans, Creative Commons Attribution-Noncommercial-Share Alike 3.0, 2007.
- [7] "*La historia del futuro industrial: cómo surgió la impresión 3D*", Alberto Marcos, *Tech*, 2013.
- [8] "*¿Cómo surgió el mundo de la impresión 3D? Impresión 3D antes de RepRap*", Alba Sánchez, 2015.
- [9] "Sitio web *AREA TECNOLOGÍA*", <http://www.areatecnologia.com>
- [10] "Sitio web *WIKIPEDIA*", <https://es.wikipedia.org>
- [11] Zortrax. (2016). "*Z-Suite 2.0 Manual (Print Preparations)*". Recuperado de <http://support.zortrax.com/z-suite-2-0-manual-print-preparations/>
- [12] "*¿Qué es y cómo funciona un servomotor?*", Antony García González, 2016.
- [13] Raspberry Pi. "*Frequently Asked Questions*". Recuperado de <https://www.raspberrypi.org/help/faqs/>
- [14] "*Microcontrolador vs Microprocesador*", Enrique Crespo, 2015. Recuperado de <https://aprendiendoarduino.wordpress.com/2015/03/29/microcontrolador-vs-microprocesador/>
- [15] "Sitio web *LEANTEC*", <http://www.leantec.es/>
- [16] "*Diferencias entre Raspberry Pi y Arduino*", Olvido Medina Aguado, Blog Mundo Raspberry, 2014.
- [17] "*Arduino Programming Notebook*", Oscar Gonzalez, *Arduino*, 2009.
- [18] "Sitio web *QUÍMICA WEB*", <http://www.quimicaweb.net/>.
- [19] "*¿Qué es y cómo funciona un potenciómetro?*", Antony García González, 2016.
- [20] "*¿Qué son los servomotores?*", Uriel Méndez, 2016. Recuperado de <https://www.330ohms.com/blogs/blog/112837444-que-son-los-servomotores>.
- [21] "Sitio web *THINGIVERSE*", <https://www.thingiverse.com/>.



7. Anexos.

SG90 9 g Micro Servo

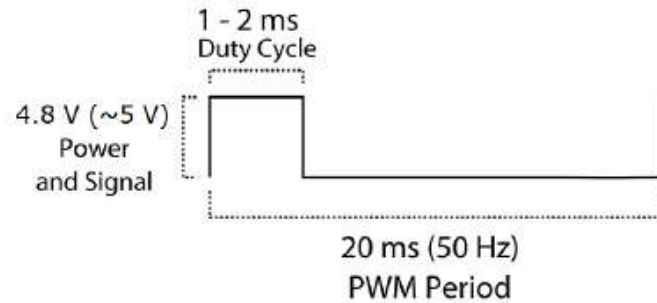
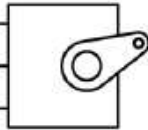


Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf-cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 μ s
- Temperature range: 0 °C – 55 °C

PWM=Orange (⌋⌋)
Vcc=Red (+)
Ground=Brown (-)



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

Placa Arduino Uno

-
-
-
-
-
-



Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteado. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

Con esta placa, Arduino conmemoraba el lanzamiento de Arduino 1.0., convirtiéndose en la versión de referencia de Arduino.

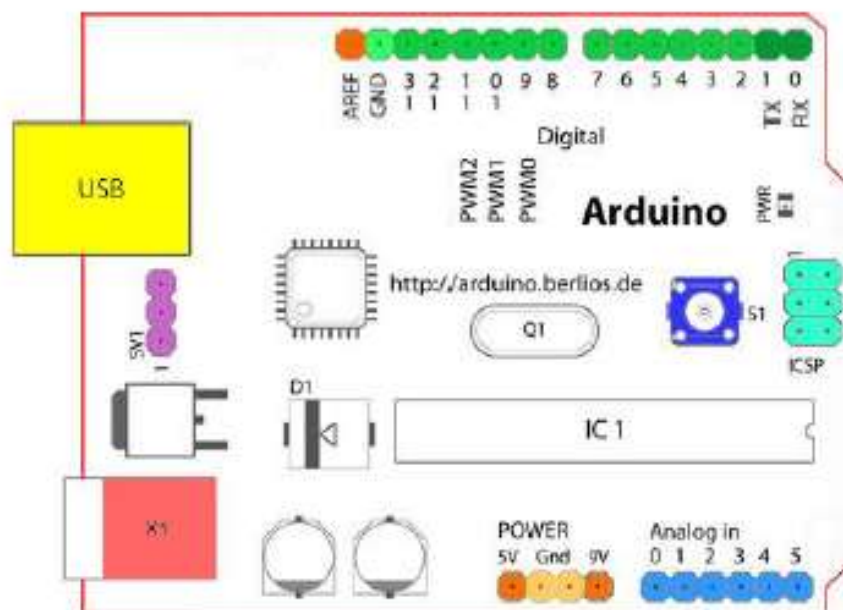
Características técnicas de Arduino Uno r3

▶ Microcontrolador:	ATmega328
▶ Voltage:	5V
▶ Voltage entrada (recomendado):	7-12V
▶ Voltage entrada (limites):	6-20V
▶ Digital I/O Pins:	14 (de los cuales 6 son salida PWM)

- ▶ Entradas Analógicas: 6
- ▶ DC Current per I/O Pin: 40 mA
- ▶ DC Current parar 3.3V Pin: 50 mA
- ▶ Flash Memory: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados para el arranque
- ▶ SRAM: 2 KB (ATmega328)
- ▶ EEPROM: 1 KB (ATmega328)
- ▶ Clock Speed: 16 MHz

Introducción general a una placa Arduino

Mirando a la placa desde la parte de arriba, este es el esquema de lo que puedes ver (los componentes de la placa con los que puedes interactuar en su uso normal están resaltados):



Empezando según las agujas del reloj:

- ▶ Terminal de referencia analógica (naranja)
- ▶ Tierra digital (verde claro)
- ▶ Terminales digitales 2-13 (verde)
- ▶ Terminales digitales 0-1/ E/S serie - TX/RX (verde oscuro) - Estos pines no se pueden utilizar como e/s digitales (`digitalRead()` y `digitalWrite()`) si estás utilizando comunicación serie (por ejemplo `Serial.begin()`).
- ▶ Botón de reinicio - S1 (azul oscuro)
- ▶ Programador serie en circuito "In-circuit Serial Programmer" o "ICSP" (azul celeste).
- ▶ Terminales de entrada analógica 0-5 (azul claro)
- ▶ Terminales de alimentación y tierra (alimentación: naranja, tierras: naranja claro)
- ▶ Entrada de alimentación externa (9-12VDC) - X1 (rosa)
- ▶ Selector de alimentación externa o por USB (coloca un jumper en los dos pines mas

cercanos de la alimentación que quieras) - SV1 (púrpura). En las versiones nuevas de Arduino la selección de alimentación es automática por lo que puede que no tengas este selector.

► USB (utilizado para subir programas a la placa y para comunicaciones serie entre la placa y el ordenador; puede utilizarse como alimentación de la placa) (amarillo)

Entradas y salidas digitales/analógicas

Un sistema electrónico es cualquier disposición de componentes electrónicos con un conjunto definido de entradas y salidas. Una placa Arduino, por tanto, puede pensarse de forma simplificada como un sistema que acepta información en forma de señal de entrada, desarrolla ciertas operaciones sobre ésta y luego produce señales de salida.

Justamente, una de las opciones que hacen más potente a Arduino son sus entradas/salidas digitales. ¿Entonces por qué hablamos de analógicas?

En los sistemas electrónicos, una magnitud física variable se representa generalmente mediante una señal eléctrica que varía de manera tal que describe esa magnitud. Por lo general, se hace referencia a las señales continuas como señales analógicas, mientras que asociamos las señales discretas a señales digitales: el ejemplo más claro es el de las señales binarias, donde la señal sólo pueden tomar dos niveles, 0 o 1.

Arduino incorpora terminales digitales (señales discretas) pero de tal forma que tenemos un gran abanico de valores con los que trabajar (por ejemplo, 255 valores de luz en un fotosensor, siendo 0 ausencia de luz y 255 el máximo valor lumínico).

Terminales Digitales

Las terminales digitales de una placa Arduino pueden ser utilizadas para entradas o salidas de propósito general a través de los comandos de programación `pinMode()`, `digitalRead()`, y `digitalWrite()`. Cada terminal tiene una resistencia pull-up que puede activarse o desactivarse utilizando `DigitalWrite()` (con un valor de HIGH o LOW, respectivamente) cuando el pin está configurado como entrada. La corriente máxima por salida es 40 mA.

► Serial: 0 (RX) y 1 (TX). Utilizado para recibir (RX) y transmitir (TX) datos serie TTL. En el Arduino Diacemila, estas terminales están conectadas a las correspondientes patas del circuito integrado conversor FTDI USB a TTL serie. En el Arduino BT, están conectados al las terminales correspondientes del módulo Bluetooth WT11. En el Arduino Mini y el Arduino LilyPad, están destinados para el uso de un módulo serie TTL externo (por ejemplo el adaptador Mini-USB).

- ▶ Interruptores externos: 2 y 3. Estas terminales pueden ser configuradas para disparar una interrupción con un valor bajo, un pulso de subida o bajada, o un cambio de valor. Mira la función `attachInterrupt()` para más detalles.
- ▶ PWM: 3, 5, 6, 9, 10, y 11. Proporcionan salidas PWM de 8 bit con la función `analogWrite()`. En placas con ATmega8, las salidas PWM solo están disponibles en los pines 9, 10, y 11.
- ▶ Reset BT: 7. (solo en Arduino BT) Conectado a la línea de reset del módulo bluetooth.
- ▶ SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estas terminales soportan comunicación SPI. Aunque esta funcionalidad está proporcionada por el hardware, no está incluida actualmente en el lenguaje Arduino.
- ▶ LED: 13. En el Diacemila y el LilyPad hay un led en placa conectado al pin digital 13. cuando el pin tiene valor HIGH, el LED está encendido, cuando el pin está en LOW, está apagado

Pines Analógicos

- ▶ Los pines de entrada analógicos soportan conversiones analógico-digital (ADC) de 10 bit utilizando la función `analogRead()`. Las entradas analógicas pueden ser también usadas como pines digitales: entrada analógica 0 como pin digital 14 hasta la entrada analógica 5 como pin digital 19. Las entradas analógicas 6 y 7 (presentes en el Mini y el BT) no pueden ser utilizadas como pines digitales.
- ▶ I2C: 4 (SDA) y 5 (SCL). Soportan comunicaciones I2C (TWI) utilizando la librería `Wire` (documentación en la página web de Wiring).

Pines de alimentación

- ▶ VIN (a veces marcada como "9V"). Es el voltaje de entrada a la placa Arduino cuando se está utilizando una fuente de alimentación externa (En comparación con los 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Puedes proporcionar voltaje a través de este pin. Date cuenta que diferentes placas aceptan distintos rangos de voltaje de entrada, por favor, mira la documentación de tu placa. También date cuenta que el LilyPad no tiene pin VIN y acepta solo una entrada regulada.
- ▶ 5V. La alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esta puede venir de VIN a través de un regulador en placa o ser proporcionada por USB u otra fuente regulada de 5V.

- ▶ Interruptores externos: 2 y 3. Estas terminales pueden ser configuradas para disparar una interrupción con un valor bajo, un pulso de subida o bajada, o un cambio de valor. Mira la función `attachInterrupt()` para mas detalles.
- ▶ PWM: 3, 5, 6, 9, 10, y 11. Proporcionan salidas PWM de 8 bit con la función `analogWrite()`. En placas con ATmega8, las salidas PWM solo están disponibles en los pines 9, 10, y 11.
- ▶ Reset BT: 7. (solo en Arduino BT) Conectado a la línea de reset del módulo bluetooth.
- ▶ SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estas terminales soportan comunicación SPI. Aunque esta funcionalidad esta proporcionada por el hardware, no está incluida actualmente en el lenguaje Arduino.
- ▶ LED: 13. En el Diacemila y el LilyPad hay un led en placa conectado al pin digital 13. cuando el pin tiene valor HIGH, el LED está encendido, cuando el pin está en LOW, está apagado

Pines Analógicos

- ▶ Los pines de entrada analógicos soportan conversiones analógico-digital (ADC) de 10 bit utilizando la función `analogRead()`. Las entradas analógicas pueden ser también usadas como pines digitales: entrada analógica 0 como pin digital 14 hasta la entrada analógica 5 como pin digital 19. Las entradas analógicas 6 y 7 (presentes en el Mini y el BT) no pueden ser utilizadas como pines digitales.
- ▶ I2C: 4 (SDA) y 5 (SCL). Soportan comunicaciones I2C (TWI) utilizando la librería `Wire` (documentación en la página web de Wiring).

Pines de alimentación

- ▶ VIN (a veces marcada como "9V"). Es el voltaje de entrada a la placa Arduino cuando se está utilizando una fuente de alimentación externa (En comparación con los 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Puedes proporcionar voltaje a través de este pin. Date cuenta que diferentes placas aceptan distintos rangos de voltaje de entrada, por favor, mira la documentación de tu placa. También date cuenta que el LilyPad no tiene pin VIN y acepta solo una entrada regulada.
- ▶ 5V. La alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esta puede venir de VIN a través de un regulador en placa o ser proporcionada por USB u otra fuente regulada de 5V.

▶ 3V3. (solo en el Diacemila) Una fuente de 3.3 voltios generada por el chip FTDI de la placa.

▶ GND. Pines de tierra.

Otros Pines

▶ AREF. Referencia de voltaje para las entradas analógicas. Utilizada con la función `analogReference()`.

▶ Reset. (Solo en el Diacemila) Pon esta línea a LOW para resetear el microcontrolador. Utilizada típicamente para añadir un botón de reset a shields que bloquean el de la placa principal.

Más información en www.arduino.cc