

ESTRUCTURA Y MECANIZADO DE UN BRAZO ROBOT



PSEUDÒNIM: MAZINGER

2018/2019
11/12/2018

“Sin práctica, la teoría es solo poesía”
-Eugenio Nieto - Fídestec.

AGRADECIMIENTOS

Gracias a mi familia por haberme apoyado en este proyecto especialmente a mis tías Pepi y Mari por aportar sus conocimientos, material documental y sus experiencias y recomendaciones, gracias a sus conocimientos trabajando con brazos robots.

También dar las gracias a mi padre por aportar todo el conocimiento y material necesario para hacer este brazo robot lo más funcional posible hasta lograr un brazo robot a escala haciendo uso de los mismos componentes.



ÍNDICE

1. INTRODUCCIÓN	5
2. OBJETIVOS DEL TRABAJO	5
3. JUSTIFICACIÓN	5
4. PLANIFICACIÓN	6
5. LA HISTORIA DEL BRAZO ROBOT	7
5.1. Robots de primera generación	8
5.2. Robots de segunda generación	8
5.3. Robots de tercera generación	9
6. ¿QUÉ ES UN BRAZO ROBÓTICO?	10
6.1. Aplicaciones de un brazo robot	10
7. ¿CÓMO FUNCIONA UN BRAZO ROBÓTICO?	11
8. COMPONENTES DE UN BRAZO ROBÓTICO	12
8.1. Estructura Mecánica	12
8.2. Sistema locomotor (actuadores)	12
8.3. Sistema sensorial	14
8.4. Sistema de decisión y planificación	14
8.5. Programación	15
8.6. Sistema de comunicación	16
8.7. Dispositivos de entrada y salida	16
9. TIPOS DE BRAZOS ROBOT	17
9.1. Estructura mecánica Cartesiana	17
9.2. Estructura mecánica Cilíndrica	17
9.3. Estructura mecánica Polar	17
9.4. Estructura mecánica Angular	18
9.5. Estructura mecánica Scara	18
9.6. Estructura mecánica en Paralelo	19
10. SEGURIDAD EN UN BRAZO ROBÓTICO	20
10.1. Seguridad en la fase del diseño del robot	20
10.2. Seguridad proporcionadas por los fabricantes	20
10.3. Normativa legal de seguridad en robots	22
11. PARTE PRÁCTICA	23
11.1. Fabricación caja controladora	23



11.2.	Componentes caja controladora	24
11.2.1.	Cuadro eléctrico y elementos de seguridad	26
11.2.2.	Instalación del cableado	27
11.2.3.	Fuentes de alimentación	27
11.2.4.	Ventilación	27
11.2.5.	PLC Siemens	28
11.2.6.	Pantalla TFT	28
11.2.7.	Arduino MEGA (2560)	28
11.2.8.	Drivers TB6600	29
11.3.	Fabricación TPU	29
11.4.	Fabricación Brazo robot	31
11.4.1.	Fabricación del brazo utilizando una impresora 3D	31
11.4.2.	Ensamblaje del brazo robot	32
11.4.3.	Distribución del cableado en el brazo robot	33
11.4.4.	Elementos con los que interactúa el brazo robot	33
11.5.	Componentes Brazo robot	34
11.5.1.	Motores P.A.P. (Paso a paso)	34
11.5.2.	Servomotor (Gripper)	35
11.6.	Presupuesto brazo robot	36
12.	CONCLUSIONES	38
13.	ÍNDICE DE ILUSTRACIONES	39
14.	BIBLIOGRAFÍA Y REFERENCIAS	42
15.	ANEXOS	43
15.1.	Datasheet Arduino UNO	43
15.1.1.	Código Arduino UNO	44
15.2.	Datasheet Arduino MEGA	45
15.2.1.	Código Arduino MEGA	46
15.3.	Datasheet Driver TB6600	49
15.3.1.	Código Arduino MEGA para los Drivers TB6600	50



1. INTRODUCCIÓN

El término Robot antiguamente fue empleado para referirse a esclavo o esclavo mecánico pero no fue hasta el siglo XVII y XVIII en Europa que se empezó a utilizar esta palabra para definir unos muñecos mecánicos con la función de mecanizar distintas actividades repetitivas en las fábricas, estos fueron los primeros robots.

Actualmente el mundo de la robótica es algo que está revolucionando tanto la industria como la vida cotidiana y gracias a su increíble versatilidad a la hora de hacer trabajos: te puede hacer la cena, un pedido sin tener que emplear mano de obra humana, puede llevar a cabo todo el proceso de una cadena de montaje, hacer actividades que pueden ser perjudiciales para los humanos, etc. La motivación por la que he hecho este trabajo es conseguir los conocimientos necesarios para diseñar y construir un brazo robótico totalmente funcional.

2. OBJETIVOS DEL TRABAJO

Los principales objetivos del trabajo son:

- Entender cómo funciona un brazo robot y poder identificar las distintas partes que lo componen así como saber también sus nombres.
- Conseguir una maqueta funcional de un brazo robótico donde se vea expresado todos los temas tratados en este TR, tales como, sus partes, los componentes que lo construyen ya sea la electrónica o las partes mecánicas y los programas utilizados para su respectivo manejo.

3. JUSTIFICACIÓN

La justificación de la viabilidad del trabajo se basa en el avance de la tecnología de estos últimos años. El avance de la tecnología ha permitido que los brazos robots sean cada vez una máquina más eficiente, inteligente, precisa y segura. Estos son los motivos por los que en un futuro gran parte de los trabajos serán sustituidos por autómatas, por estas razones es importante comprender cómo actúa un brazo robot y ver su utilidad en la sociedad de hoy en día ya que poco a poco se intentan integrar en la vida de las persona.



4. PLANIFICACIÓN

Esta es la planificación del trabajo expresado cronológicamente en una tabla de Excel. En esta tabla de planificación se hace un breve resumen de todas las partes en las que se desarrolla el TR en función de las tareas y el tiempo que han llevado a cabo realizarlas. El gráfico está dividido en dos partes: las teóricas que son las distintas fases en las que se ha ido realizando el TR y las que son prácticas es decir el desarrollo del brazo robótico impreso en 3D. Al principio de la parte teórica y la parte práctica se hace una pequeña planificación donde se esquematiza toda la información para después tener una idea clara de por donde llevar el temario en cuestión.

Actividades	2018										2019
	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Enero
Planificación TR	[Barra azul]										
Fase 1	[Barra roja]										
Fase 2	[Barra naranja]										
Fase 3		[Barra verde]									
Fase 4				[Barra marrón]							
Fase 5							[Barra magenta]				
Fase 6									[Barra azul]		
Fase 7										[Barra roja]	
Planificación Parte Práctica	[Barra azul]										
Impresión de piezas 3D						[Barra morada]					
Montaje Hardware						[Barra amarilla]					
Programación Software					[Barra naranja]						
Treabajo escrito Parte Práctica							[Barra azul]				
Preparación presentación										[Barra verde]	



5. LA HISTORIA DEL BRAZO ROBOT

El ser humano ha construido máquinas que imitan partes del cuerpo humano como por ejemplo los antiguos egipcios y los griegos. El principio de la robótica se desarrolló en la industria textil del siglo XVIII, donde un joven ingeniero llamado Joseph Jacquard inventa en 1801 una máquina textil programable mediante una PCB. Luego, la Revolución Industrial impulsó el desarrollo y construcción de los primeros robots. Además de esto, durante los siglos XVII y XVIII en Europa fueron construidos muñecos mecánicos muy ingeniosos que tenían algunas características de los robots. Otro ingeniero de la época llamado Jacques de Vaucansos construyó varios músicos de tamaño humano a mediados del siglo XVIII. En 1805, Henri Maillardert construyó una muñeca mecánica que era capaz de hacer dibujos y así fue evolucionando el mundo de la robótica cada vez consiguiendo funciones más complejas que las anteriores.

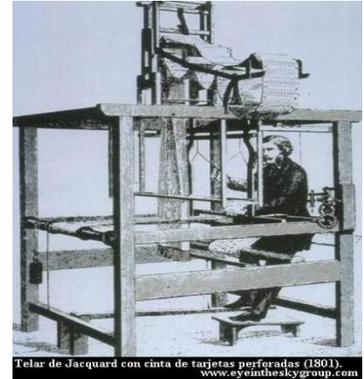


Fig. 1 Primer robot industrial

La palabra robot se utilizó por primera vez en 1920 en una obra llamada "Los Robots Universales de Rossum", escrita por el dramaturgo checo Karel Capek. Su trama trataba sobre un hombre que fabricó un robot y luego este último mata al hombre. La palabra checa 'Robota' significa trabajo forzado, y cuando se tradujo al inglés se convirtió en el término robot.

1938: Los americanos Willard Pollard y Harold Roselund fabrican la primera máquina para pintar con spray.

1942: Isaac Asimov publica las tres leyes de la robótica. Leyes que marcan la supeditación de los robots a la voluntad humana. Estas leyes se ponen de moda especialmente a finales de siglo XX, al introducirse la robótica en los hogares y plantearse un problema ético y de seguridad civil.

1951: Raymond Goertz diseña el primer brazo mecánico manejado a distancia para la Comisión de la Energía Atómica

1976: El robot de la NASA "Viking II" aterriza en Marte. Disponía de un brazo robótico articulado.

2013: se fabrican prótesis robóticas por medio de la impresión 3D.

2015: Se han alcanzado mayores perfecciones para los brazos robóticos, logrando agarrar objetos delicados sin quebrarlos.



Al largo de la historia se han ido clasificando todos estos tipos de robots en 3 grupos según sus utilidades de más sencillas a más complejas:

- Robots de primera generación
- Robots de segunda generación
- Robots de tercera generación

5.1. Robots de primera generación

Los robots de primera generación actúan como "esclavo" mecánico de un hombre, que funciona mediante la transmisión directa con las articulaciones y movimientos de los humanos. Esta transmisión tiene lugar mediante las extremidades superiores del hombre, unos ejemplos de esta primera generación de robots serían manipulación de materiales radiactivos, obtención de muestras submarinas, etc.



Fig. 2 Robot de primera generación

5.2. Robots de segunda generación

Los robots de segunda generación actúan automáticamente sin intervención humana frente a posiciones fijas en las que el trabajo ha sido preparado y ubicado ejecutando movimientos repetitivos, que funciona mediante una serie de programas insertados en una tarjeta que los interpreta y envía coordenadas al robot para que este se mueva siguiendo esa serie de coordenadas. Al ser códigos de coordenadas sencillos le da cierta versatilidad al robot, unos ejemplos de su versatilidad sería en el estampado en frío y en caliente asistiendo a la carga y descarga de piezas. En la inyección de termoplásticos, en los procesos de soldadura punto a punto y continúa, en tareas de pintado...



Fig. 3 Robot de segunda generación



5.3. Robots de tercera generación

Son dispositivos que habiendo sido contruidos para alcanzar determinados objetivos, son capaces de elegir la mejor forma de hacerlo teniendo en cuenta el ambiente que los rodea. Para obtener estos resultados es necesario que el robot posea algunas condiciones que posibiliten su interacción con el ambiente y los objetos, ya sean sensores o las distintas herramientas que utilice para trabajos específicos como soldar, atornillar, sellar, etc. Las mínimas aptitudes requeridas son:



Fig. 4 Robot de tercera generación

capacidad de reconocer un elemento determinado en el espacio y la capacidad de adoptar propias trayectorias para conseguir el objetivo deseado. Los métodos de identificación empleados hacen referencia a la imagen óptica por ser esta el lenguaje humano en la observación de los objetos.



6. ¿QUÉ ES UN BRAZO ROBÓTICO?

Según el país existen diferentes definiciones para el brazo robot, por ejemplo en el mercado japonés el brazo robot está considerado como un **manipulador** que significa mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico. Pero en el mercado euro-americano el brazo robot está considerado como manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.

6.1. Aplicaciones de un brazo robot

Según el sector donde se destine el brazo robot, ya sea en un ambiente industrial o cotidiano este tendrá una aplicación u otra. Unos ejemplos de la función que puede llegar a realizar un robot en un ambiente industrial son los siguientes: atornillar, pegar, soldar, pintar, desplazar la pieza de una cinta a otra, montaje de secciones...



Fig. 5 Brazo robot industrial Soldador



Fig. 5 Brazo robot industrial Traspalé



Fig. 7 Brazo robot industrial Pintor

Otros ejemplos de función en un ambiente cotidiano sería doblar ropa, planchar, hacer la comida...



Fig. 8 Brazo robot industrial Prototipo 1



Fig. 9 Brazo robot industrial Prototipo 2



7. ¿CÓMO FUNCIONA UN BRAZO ROBÓTICO?

Un brazo robot se compone principalmente de dos partes: el controlador (ordenador donde procesa y actúa toda la información en coordenadas que le llega directamente del programa previamente compilado en este ordenador) y el propio brazo robot.



Para la transmisión del motor al respectivo eje donde queremos que se produzca el movimiento dentro del brazo actúa un ingenioso mecanismo de poleas y cuerdas dentadas que facilita esta rotación de los ejes.

Fig. 6 Movimiento brazo robot

Aquí podemos ver un claro esquema de todos los elementos que actúan para conseguir que el brazo robot se mueva y cumpla sus funciones como autómatas.

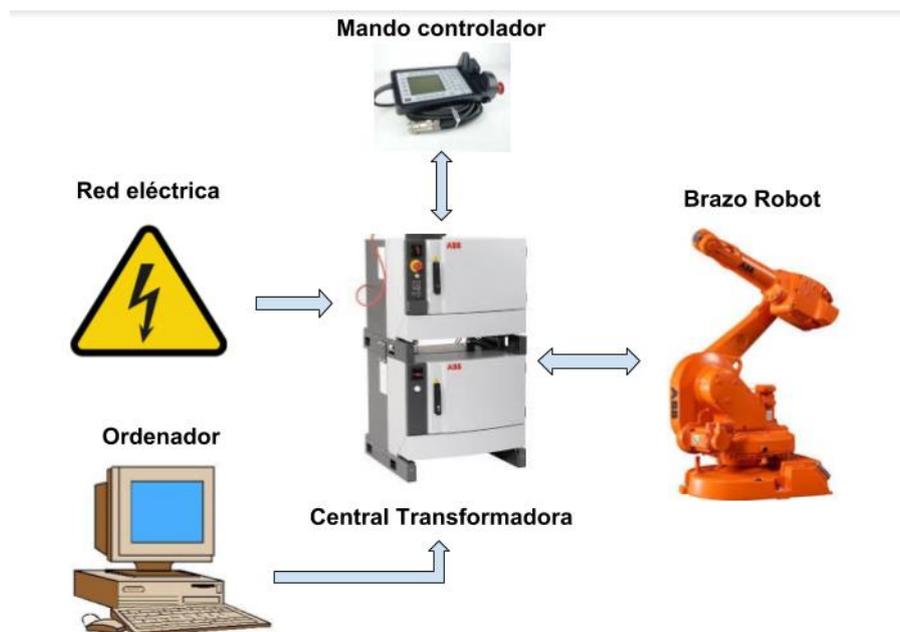


Fig. 7 Esquema hardware brazo robot

El mando controlador también es el TPU



8. COMPONENTES DE UN BRAZO ROBÓTICO

El complejo mecanismo que se encarga de hacer actuar un brazo robot en perfecta sincronía se puede dividir en 6 apartados y estos se pueden dividir en otros subapartados según el tipo de robot que se trate.

8.1. Estructura Mecánica

La estructura mecánica de un robot se secciona en las mismas partes que un brazo humano, dividiéndolo por los ejes como los codos el hombro o la muñeca. Estas son las siguientes partes en las que se divide un brazo robot:

- Cuerpo o Cintura
- Hombro
- Brazo
- Codo
- Antebrazo
- Muñeca
- Mano o Gripper

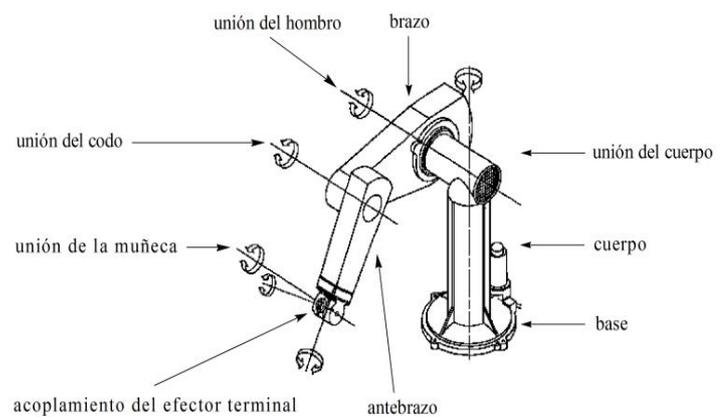


Fig. 8 Estructura brazo robot

8.2. Sistema locomotor (actuadores)

Los actuadores de los brazos robots pueden ser eléctricos, neumáticos o hidráulicos según el tipo de faena y el ambiente donde se tenga que instalar el robot.

Eléctricos

Los brazos robots que se componen por motores eléctricos, son los más complicados ya que la electrónica requerida para mover estos motores es más compleja que otros sistemas. Estos sistemas requieren de un transformador para modificar la tensión trifásica de la red a una tensión monofásica con el fin de alimentar los potentes drivers que son los encargados de recibir órdenes del ordenador incorporado en el robot y saberlas interpretar para así enviar los pulsos correspondientes a los motores para generar el movimiento en el brazo robot. Este sistema de actuadores eléctricos son mayormente utilizados ya que la mayoría de brazos robots de los que se compone el mercado son medianos y utilizan motores eléctricos, pues estos no necesitan tanta velocidad ni potencia.



Neumáticos

Los brazos robots que utilizan sistemas neumáticos para su respectivo funcionamiento son los más limpios de todos así que su funcionalidad se basa en trabajos repetitivos en salas blancas. Otra ventaja de los brazos robots movidos por pistones neumáticos es su increíble rapidez y actuación en trabajos repetitivos y sencillos como mover piezas de una cinta a otra y otros trabajos semejantes.

Hidráulicos

El sistema de actuadores hidráulicos son en los que se usa un fluido, generalmente un aceite especial para hidráulica. Estos sistemas de impulsión hidráulica son generalmente usados para robots grandes, los cuales presentan mayor velocidad y mayor resistencia mecánica.

Aquí se puede apreciar una tabla con las ventajas e inconvenientes de cada uno de estos tipos de actuadores en brazos robots:

	Ventajas	Desventajas
Robots Neumáticos	-Baratos	-Dificulta de control continuo
	-Rápidos	-Instalación especial (compresor, filtros)
	-Sencillos	-Ruidoso
	-Robustos	
Robots Hidráulicos	-Rápidos	-Difícil mantenimiento
	-Alta relación potencia peso	-Instalación especial(filtros, eliminación aire)
	-Auto lubricantes	-Frecuentes fugas
	-Alta capacidad de carga	-Costosos
	-Estabilidad frente a cargas estáticas	
Robots Eléctricos	-Precisos	-Potencia limitada
	-Fiables	
	-Fácil control	
	- Silenciosos	



8.3. Sistema sensorial

Un sensor es un dispositivo que detecta la presencia o la ausencia de algún objeto. Un brazo robot se compone de múltiples sensores para interactuar con su entorno y asegurar la seguridad requerida por la normativa para evitar posibles daños materiales o humanos. Algunos de estos sensores los podemos ver a continuación:

- Sensores de proximidad:** Un sensor de proximidad cumple la función de detectar la presencia de objetos cercanos sin la necesidad de contacto.
- Sensores de contacto:** Este tipo de sensor es utilizado para detectar la presencia de objetos con la necesidad de contacto físico.
- Sensores electromagnéticos:** Un sensor de estas características funciona como un interruptor eléctrico activado por un campo magnético cercano.
- Sensores fotoeléctricos:** Un sensor fotoeléctrico emite un haz de luz (puede ser infrarrojo o visible para el ojo humano) que al cortar este haz de luz por la presencia de un objeto el sensor detecta esta presencia de algún objeto.
- Sensor sin contacto:** Este sensor es utilizado para detectar la presencia de objetos sin la necesidad de contacto mutuo. Uno de estos sensores es el sensor de proximidad.
- Sensores de ultrasonidos:** Este tipo de sensor funciona con dos sensores, uno que se encarga de enviar una serie de ondas mientras que el otro se encarga de recibir esta misma señal. Esa señal rebota contra un objeto determinado pero si ese tiempo que ha tardado en recibir esa misma onda es menor o mayor significa que ese objeto está más cerca o más lejos.

8.4. Sistema de decisión y planificación

El sistema de decisión y planificación en un brazo robot va de la mano con la programación de un brazo robótico. Esta programación es la que le da órdenes al brazo robótico dependiendo del Hardware exterior como pueden ser los distintos sensores explicados con anterioridad. Un ejemplo sería la interrupción de una cadena de montaje, en este caso el



robot debe de ser autosuficiente para tomar la decisión de dejar de producir y pararse en una posición *Home* previamente preestablecida.

8.5. Programación

Existen dos formas de programar un brazo robótico, la textual o indirecta y la guiada o directa.

Para programar un robot de forma textual o indirecta es necesario crear un algoritmo que permita mover el robot de la forma deseada, seguidamente transformar este algoritmo en el estilo de lenguaje dependiendo la marca del robot y la tarea requerida, esta es una tabla donde se ve reflejado el tipo de lenguaje y para que robot y función está destinado:

LENGUAJE	UNIVERSIDAD /FABRICANTE	ORIGEN Y CARACTERISTICAS	APLICACIONES
WAVE (1973)	Stanford	- Nivel actuador - Compilador asociado al robot. - Sintaxis simple: MOVE, SEARCH, CENTER. - Objetos tratados: constantes, variables, vectores, contadores	- Manipulación de piezas mecánicas. - Montaje de bomba de agua. - Ensamblaje.
AL (Assembly Language) (1974)	Stanford	- Basado en PASCAL. - Nivel actuador. - Transformador de coordenadas. - Sintaxis compleja: MOVE TO, MOVE VIA, etc., SEARCH, ACROSS SEARCH, WITHOUT..., CENTER ON ... - Objetos tratados: los de Wave más planos, traslaciones, rotaciones. - Estructura Algol.	- Manipulación de piezas mecánicas. - Sistema de desarrollo de programas Pointy. - Robots Unimation.
RAPT (1978)	Sistemas de Toulouse (Universidad de Edimburgo)	- Interprete escrito en APT. - Nivel objeto.	- Montaje
VAL I (1979) VAL II (1983)	Unimation	- Se partió de AL, pero utilizando BASIC. - Nivel actuador. - Transformador de coordenadas. - Instrucciones de movimiento y de control (MOVE-, DEPART-). - instrucciones Aritméticas. - Estructura Algol.	- Robot Puma de Unimation
MCL	Cincinnati Milacron	- Desarrollo por ICAM y realizado en Fortram	
AML (1979) FUNKY MAPLE AUTOPASS (1977)	IBM	- Alto nivel interactivo estructurado. - Nivel objeto. - Compilador al nivel. - Sintaxis evolucionada, como colocar X sobre Y alineando Z y T.	- Corresponde al nivel de descripción de las gamas de montaje actuales.
LRP	ACMA	- Desarrollado en colaboración con la Universidad de Montpellier.	
V+ (1989)	ADEPT	- Lenguaje textual de alto nivel. - Ejecución de varios programas al mismo tiempo (multitarea). - Proceso asíncrono o ejecución de rutinas de reacción ante determinados eventos.	- Usado en Adept y Staübli
RAPID (1994)	ABB	- Lenguaje textual de alto nivel altamente estructurado. - Estructura modular del programa. - Uso de estructuras predefinidas para especificar la configuración del robot y las características de la herramienta.	
KAREL	FANUC	- Basado en PASCAL. - Soporta la multitarea. - Estructura de datos asociados modificable.	
KRL	KUKA	- Basado en el estándar IRL, descrito en la norma DIN 66312, lo que aporta transportabilidad.	



Los lenguajes de programación en los robots industriales no son generalizados, es decir cada robot tiene su propio lenguaje concreto, como es el caso de la marca KUKA que utiliza el lenguaje de programación KRL. Suelen estar diseñados para programar un modelo de robot particular y no pueden emplearse en otros tipos de robots.

Otra forma de programar un robot es de forma guiada o directa que consiste en guiarlo manualmente con el control o *Teach Pendant Unit* (TPU) utilizando un método que consiste en *Point To Point* (PTP) que trata en llevar el robot por una serie de coordenadas mientras se van guardando para crear un movimiento con el que pase por esa serie de coordenadas.

8.6. Sistema de comunicación

Últimamente son muchas las empresas que tienen cadenas de montaje únicamente operadas por robots y la única manera de tenerlos todos funcionando a máximo rendimiento es con un correcto mantenimiento y puesta a punto. La manera de mantener la comunicación con todos estos brazos robots es manteniéndolos conectados todos a una misma red donde puedan aparecer los múltiples fallos localizados en los distintos robots para mantener una eficiencia extraordinaria a la hora de localizar ese fallo y arreglarlo en el menor tiempo posible.

8.7. Dispositivos de entrada y salida

Los dispositivos de entrada y salida permiten introducir y, a su vez, ver los datos del controlador. Para mandar instrucciones al controlador y para dar de alta programas de control, comúnmente se utiliza una computadora externa para compilar los datos al ordenador del brazo. Es necesario aclarar que algunos robots únicamente poseen uno de estos componentes. Los más comunes son: teclado, monitor y caja de comandos (*teach pendant*).

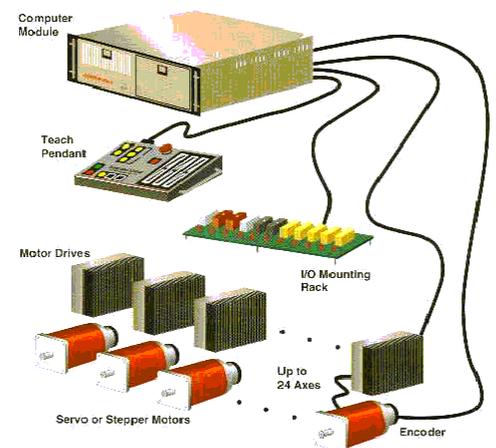


Fig. 9 Dispositivos de entrada y salida

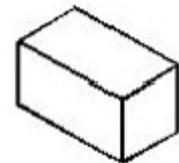
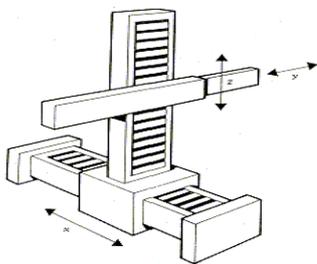


9. TIPOS DE BRAZOS ROBOT

Según el tipo de faena que queramos llevar a cabo se empleará un tipo de estructura en el brazo robot u otra, siempre buscando la mayor eficiencia posible en la faena requerida. Actualmente existen 6 tipos de estructuras compuestas por la estructura Cartesiana, la estructura Cilíndrica, la estructura Polar, la estructura Angular, la estructura Scara y por último la estructura en Paralelo.

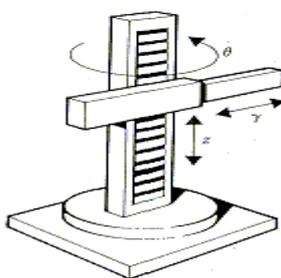
9.1. Estructura mecánica Cartesiana

Este tipo de estructura se usa normalmente en espacios de trabajo con grandes dimensiones. Este brazo posee 3 ejes o 3 grados de libertad reflejados como un eje de coordenadas X, Y y Z.



9.2. Estructura mecánica Cilíndrica

Este tipo de estructura mecánica destaca por su movimiento de rotación sobre la base del brazo facilitando así el trabajo en superficies cilíndricas. Este robot se compone de 3 ejes o grados de libertad pudiendo realizar dos ejes movimientos lineales y un eje movimientos rotacionales.

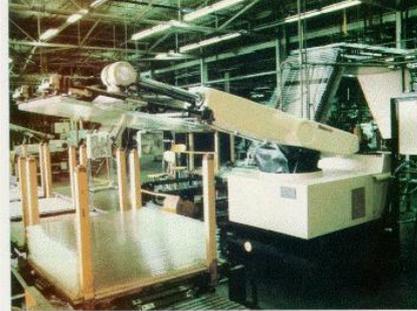
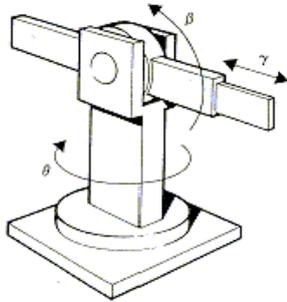


9.3. Estructura mecánica Polar

Una estructura mecánica utilizada mayormente en superficies de grandes medidas donde haya que transportar un objeto a una distancia considerable de la base del robot. Esto se

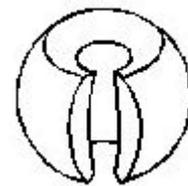
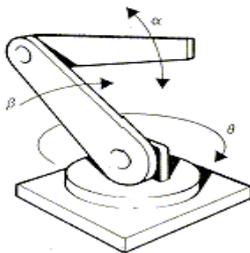


consigue con 3 ejes o grados de libertad dividiéndose en un movimiento rotacional de la base, un movimiento angular entre la base y el brazo y otro lineal para la extensión del brazo.



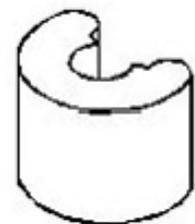
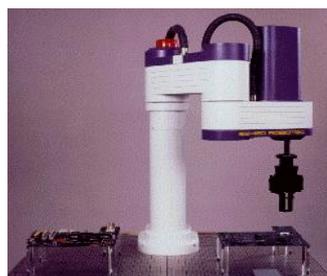
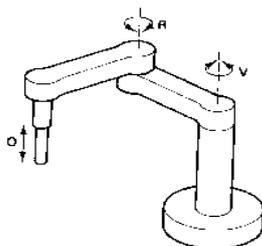
9.4. Estructura mecánica Angular

Para esta estructura generalmente se utilizan superficies cilíndricas. La característica de este tipo de estructura es la similitud con un brazo humano intentando así realizar las tareas de nuestra misma manera. El robot utiliza 3 ejes o ejes de rotación para posicionarse, siendo así un eje rotacional y dos angulares.



9.5. Estructura mecánica Scara

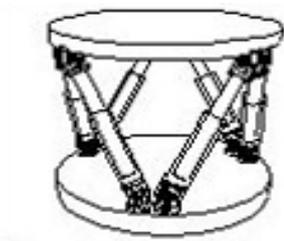
Esta estructura Scara es similar a la configuración cilíndrica pero el radio y la rotación se obtiene por uno o dos ejes. Este brazo puede realizar movimientos horizontales de mayor alcance debido a sus dos articulaciones de movimiento rotatorio. El robot de configuración Scara también puede hacer un movimiento lineal (mediante su tercera articulación).





9.6. Estructura mecánica en Paralelo

Este tipo de estructura en paralelo es la menos usada de todas debido a sus grandes dimensiones. El brazo posee articulaciones lineales consiguiendo un volumen de trabajo cóncavo.





10. SEGURIDAD EN UN BRAZO ROBÓTICO

Un robot industrial está categorizado como una máquina industrial que se encuentra en zonas de trabajo compartida con personas y por esto es susceptible a producir accidentes laborales entre los más comunes el impacto de su potente brazo contra algún operario/a produciendo así lesiones graves. Por ello existen muchas medidas de seguridad para intentar evitar estos accidentes.

10.1. Seguridad en la fase del diseño del robot

En el diseño del robot y su sistema de control debe considerarse siempre el posible accidente en el momento de estar operando el robot, tomándose las acciones oportunas para evitarlo en la medida de lo posible. Así, el propio robot debe contar con una serie de medidas internas encaminadas a evitar posible accidentes:

- **Supervisión del sistema de control:** El sistema de control debe realizar una continua supervisión del correcto funcionamiento de todos los subsistemas (bucles de realimentación, accionamientos, etc.) y de incluso él mismo (match-dog).
- **Paradas de emergencia:** Deben disponerse paradas de emergencia que desenergicen completamente al robot.
- **Velocidad máxima limitada:** El sistema de control asegurará que la velocidad máxima de los movimientos cuando una persona se encuentra en las proximidades del robot (fase de propagación por ejemplo) sea inferior a la nominal (como referencia debe ser inferior a 0,3 m/s).
- **Detectores de sobreesfuerzo:** Se incluirán detectores de sobreesfuerzo en los accionamientos que los desactiven cuando sobrepase un valor excesivo (caso de colisión o de atrapar a una persona contra una parte fija).
- **Códigos de acceso:** El acceso a la unidad de control y el arranque, parada y modificación del programa, estarán limitadas mediante el empleo de llaves, códigos de seguridad, etc.
- **Frenos mecánicos adicionales:** Si el robot maneja grandes cargas, se deberán incluir frenos mecánicos que entren en funcionamiento cuando se corte la alimentación de los accionadores. Asimismo, se dispondrán medios para desactivar estos frenos de forma manual.

10.2. Seguridad proporcionadas por los fabricantes

Existen una serie de elementos de seguridad con los que debe contar el brazo robot en su momento de fabricación para que en su momento de puesta en marcha haya unos requisitos de seguridad que aseguren al operario que en caso de emergencia o prevención de accidente laboral el robot se parará para evitar posibles daños.



Estas medidas proporcionadas por el fabricante para evitar estos posibles riesgos laborales son las siguientes:

1. No permitir que el personal no formado trabaje con los robots.
2. Instalar un mecanismo de acceso a la célula con código para impedir la entrada de personal no autorizado, así como barreras de seguridad fotoeléctricas industriales, sensores de presencia o proximidad y sistemas de visión para reforzar la seguridad.



Fig. 10 Célula de seguridad

3. Identificar claramente la zona máxima en la que se produce el movimiento del robot con marcas en el suelo, señales y barreras especiales, también es recomendable colocar todos los controles de los equipos fuera de esa zona. Los operarios deben conocer perfectamente el área de la célula de trabajo completa (robot y su área de trabajo).

4. Instalar botones o interruptores de <parada de emergencia> para el operador y en puntos críticos dentro y alrededor de la célula de trabajo. Los operarios deben conocer dónde están colocados dichos botones.



Fig. 11 Seta parada de emergencia

5. Instalar luces y/o alarmas que se activen cuando el robot esté funcionando.
6. Revisar los mecanismos de seguridad periódicamente.
7. Proporcionar suficiente espacio dentro de la célula de trabajo para que el personal pueda realizar operaciones de mantenimiento de forma segura.
8. Antes de una operación manual con el *teaching* del robot, verificar que no hay peligro dentro del entorno del robot.
9. El área cercana al robot debe estar limpia y sin aceite, agua o residuos.
10. Asegurarse de que todo el personal está fuera del área de trabajo antes de comenzar con la producción, y por supuesto, no entrar nunca en el área de trabajo si el robot están en operación.
11. Nunca intentar parar el movimiento del robot con el cuerpo.



10.3. Normativa legal de seguridad en robots

Antiguamente las normativas legales no se actualizaron hasta los años noventa cuando hubo una revolución en el mundo de los brazos robóticos industriales. Aparecieron dos normativas, una de ellas internacional y otra para Europa que comparte la misma que la española al ser territorio europeo.

Normativa internacional ISO 10218:1992

Se trata de una normativa realizada por el Instituto Nacional de Normalización de Estados Unidos (ANSI) [ANSI-92]. También data del año 1992, siendo una revisión de la normativa publicada en 1986. Es relativamente breve pero presenta algunas características que merecen destacarse. Por ejemplo, la inclusión, en el apartado sobre la definición de riesgos, de algunos epígrafes que versan sobre la probabilidad de la aparición de un accidente y la severidad del posible daño físico a una persona, dependientes del nivel de experiencia del operador y la frecuencia en la que éste se encuentra en zona de peligro.

Normativa europea EN 775 y española UNE-EN 775

El Comité Europeo de Normalización (CEN) aprobó en el año 1992 la normativa EN 775, adaptación de la norma intencional ISO 10218:1992. A su vez la Asociación Española de Normalización (AENOR) adoptó en marzo de 1996 esta normativa (traducida al español) denominándose UNE-EN 775 y que lleva por título: “Robots manipuladores industriales. Seguridad” [UNE-96]. Por lo tanto, la normativa española (coincide con la europea) se ha desarrollado sobre la normativa internacional ISO 10218:1992, incorporando y adaptando aspectos relacionados en la normativa europea “EN 292:1991 Safety of Machinery – Basic concepts, general principles for design” [EN-91].



11. PARTE PRÁCTICA

En la parte práctica de este trabajo explicare como se elaboró el brazo robot fabricado con un total de 28 piezas impresas en 3D a partir de un modelo de robot ROBOTEURS RBX1, un robot que consta de 6 ejes movidos por 7 motores.

A continuación la parte práctica se dividirá en 2 partes la explicación de la caja controladora y la explicación del propio brazo robot.

11.1. Fabricación caja controladora

El proceso comienza haciendo un dibujo en papel con las distintas perspectivas de la caja para después poner las cotas de las medidas que va a tener una vez finalizada. Después de tener todas las medidas, estas se tienen que pasar a una plancha de metal, en este caso se decidió emplear acero. Cuando se tienen las medidas planteadas en la plancha de acero se procede a cortar y hacer los agujeros necesarios a las medidas y ubicaciones marcadas para después doblar y soldar consiguiendo así la estructura de la caja. Para la puerta se sigue este mismo proceso. Esto es un proceso muy laborioso ya que se trata de conseguir el mínimo error posible, en el caso de la caja y la puerta se invirtieron 8 horas solamente para conseguir una estructura como la que se puede observar en la foto (fig. 11).

A continuación se toman las medidas del metacrilato para la ventana de la puerta y de la baquelita* situada detrás del cuadro eléctrico y así impedir una posible derivación de corriente. El último proceso después de asegurarse que todas las medidas son correctas es pasar a la fase de pintar todos los elementos, en este caso se darán dos manos de imprimación para conseguir una superficie lisa y regular. Después de dar las 2 manos de imprimación se aplicará la pintura que será un gris como color principal y un naranja de color secundario para dar un toque más estético (fig. 12).

Una vez se han secado todas las piezas y cortado el metacrilato y la baquelita se procede a montar e instalar todas las piezas con tornillos inoxidables de cabeza hexagonal para darle un toque de rigidez y aspecto industrial.

Aquí podemos observar algunas de las especificaciones de la caja como sus medidas, peso, etc.



Fig. 12 Caja de drivers sin pintar



Fig. 13 Caja de drivers pintada



- Alto: 50 cm
- Ancho: 20 cm
- Largo: 45 cm
- Peso: 23 kg

Para finalizar este apartado de la caja controladora hace falta decir que las horas empleadas han sido 11 horas en total consiguiendo así el resultado de la foto (fig. 13).

Para decorar la caja por el exterior se han aplicado unos vinilos de color naranja con la figura de un brazo robot.



Fig. 14 Caja de drivers montada

11.2. Componentes caja controladora

La caja controladora es la que se encarga de enviar las órdenes al brazo robot según el programa compilado en la placa controladora Arduino Mega. La principal función de la caja es alojar y refrigerar los drivers cuyo trabajo consiste en leer los pulsos enviados desde el Arduino y modificar estos pulsos para que el motor paso a paso conectado al driver funcione correctamente. Los motores paso a paso siempre tienen que ir conectados a un driver para que el motor pueda moverse.

A continuación se muestran diversas perspectivas de la caja controladora para poder ver todo su contenido y así poder explicar sus funciones posteriormente:

Parte de arriba



Parte trasera

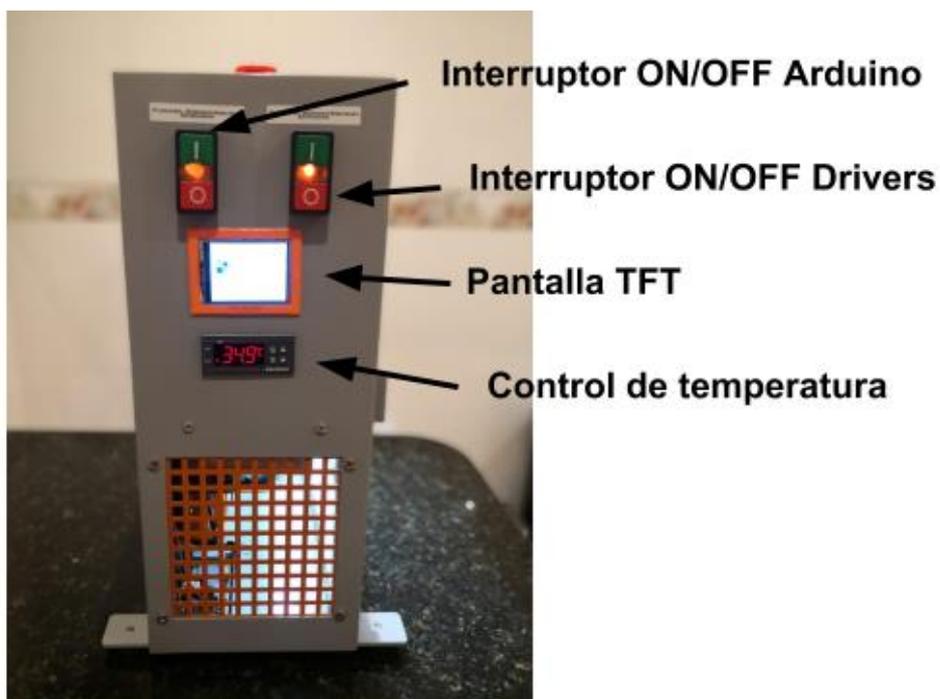




Parte lateral



Parte frontal





11.2.2. Instalación del cableado

En la instalación de la caja contamos con 3 tipos de secciones distintas: el primer tipo de sección de cable es de 2.5mm^2 , esta sección la podemos encontrar desde el tramo de la clavija de entrada situada en la parte trasera de la caja hasta la bornera pasando por el magnetotérmico y el diferencial. Otra sección que podemos encontrar es la de 1.5mm^2 , esta es la que se encuentra en la salida de la bornera para distribuir la electricidad por toda la instalación como la alimentación de los transformadores o la alimentación del ventilador de 220V. Algunos tramos de esta sección de cable también funcionan a 24V como la alimentación de los drivers o la alimentación del PLC Siemens. Por último la sección de 0.5mm utilizada en toda la instalación de electrónica como el tramo del Arduino a los drivers y de los drivers a los conectores para los motores P.A.P. (Paso A Paso).

11.2.3. Fuentes de alimentación

La caja contiene en total 3 fuentes de alimentación, una de 220V a 24V y 10A para alimentar a los 7 drivers instalados en el interior de la caja. La siguiente fuente es de 220V a 24V para alimentar el PLC Siemens, las 2 tiras LEDs y los 7 drivers. La última fuente de alimentación es de 220V a 8.5V para alimentar las dos placas Arduino y un ventilador para ventilar toda la electrónica.



Fig. 16 Fuente de alimentación

11.2.4. Ventilación

La ventilación era un reto muy importante en este proyecto ya que es necesario conseguir un flujo de aire continuo y fresco para disipar el calor de la batería de drivers. El sistema empleado consiste en un ventilador de 220v controlado por un controlador de temperatura de doble relé. El controlador ha sido programado para que por debajo de los 27 grados el ventilador permanezca quieto pero por encima de los 27 grados el relé cierra el circuito y el ventilador comienza a funcionar hasta que nuevamente la temperatura se sitúe por debajo de los 27 grados y el ventilador deje de funcionar.

Aparte, la electrónica tiene un ventilador de 9V que funciona independientemente de todo el sistema anterior. Una vez se energizan las dos placas Arduino el ventilador también empieza a funcionar.



Fig. 17 Sistema de ventilación



11.2.5. PLC Siemens

El PLC o también conocido como *Programmable Logic Controller* es un programador que cuenta con una pantalla LCD por donde se puede programar las distintas entradas y salidas de relés. Sus usos en muchas máquinas industriales lo hace un dispositivo comúnmente utilizado. Los usos más frecuentes son automatizar procesos, procesos en bucle... En este caso el PLC ha sido programado con la función de conmutador ya que al utilizar 2 botones On/Off momentáneos era imprescindible el uso del conmutador.



Fig. 22 LOGO Siemens

11.2.6. Pantalla TFT

La pantalla TFT es un elemento que no interviene en el control del brazo robot pero era interesante tener algún accesorio que ilustre imágenes para hacer un poco más interactiva la caja de drivers, en este caso la pantalla ilustrara 3 imágenes que irán alternándose con una fracción de tiempo de 5 Segundos cada una. Los 3 elementos a ilustrar son el nombre del brazo robot, el logo de TAM y el logo de la empresa Helcarinox S.L. (empresa encargada de la planificación y construcción de la caja de drivers y la base del brazo robot). Una



Fig. 23 Pantalla TFT y Arduino UNO

pantalla TFT al contrario de una LCD puede mostrar imágenes en color y tiene la capacidad de ser también una pantalla táctil. Esta pantalla necesita la ayuda de una placa controladora, en este caso se ha utilizado una placa Arduino UNO por la comodidad a la hora de programarla.

11.2.7. Arduino MEGA (2560)

Esta placa desarrolla una función muy importante en la caja controladora ya que es la encargada de controlar todo lo referente al brazo robot. Se ha elegido este modelo de placa porque tiene hasta 53 pines digitales y 15 pines analógicos ideal para controlar muchas tareas. Los elementos que controla son el TPU, el servomotor del *gripper* y los drivers que a su vez controlan los motores paso a paso.

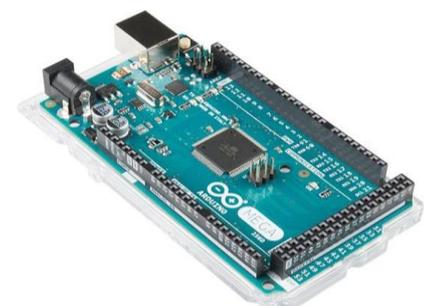


Fig. 24 Arduino MEGA

La programación del Arduino se divide en: librerías de la LCD y del servomotor, declaración de los pines de los drivers, de la pantalla LCD, del servomotor y de los botones del TPU, seguidamente el programa en cuestión donde desde los botones del TPU se puede seleccionar el programa que se quiera llevar a



cabo mientras en la LCD del TPU te muestra el programa seleccionado y el tiempo de duración que este tiene.

11.2.8. Drivers TB6600

Los Drivers son los encargados de interpretar los pulsos de ENA (Enable), DIR (Dirección) y PUL (Pulso) que llegan desde el Arduino MEGA y transformarlos de tal manera que salgan unos pulsos que indiquen al motor la dirección o sentido y el pulso digital de trabajo (frecuencia de trabajo). La característica de estos drivers es que aunque al motor P.A.P (Paso a Paso) no le esté llegando señal del Arduino este seguirá energizado en la última posición en la que se quedó es decir se quedará estático sin posibilidad de moverse.

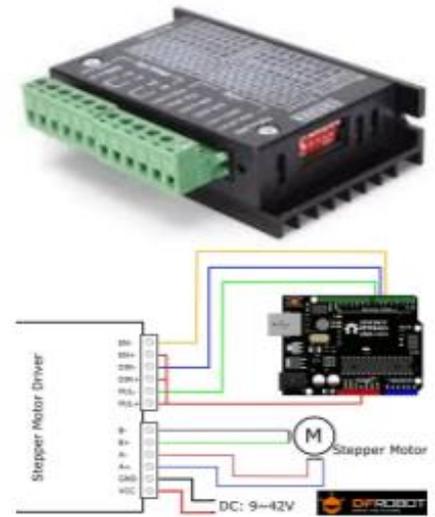


Fig. 25 Driver TB6600 y esquema

A continuación se muestra una tabla donde se ven reflejadas las conexiones de los distintos drivers al Arduino MEGA:

Drivers	ENA(Enable)	DIR(Dirección)	PUL(Pulso)	A(Amperios)
Driver 1	Pin Digital 2	Pin Digital 3	Pin Digital 4	2.8A
Driver 2	Pin Digital 5	Pin Digital 6	Pin Digital 7	2.8A
Driver 3	Pin Digital 8	Pin Digital 9	Pin Digital 10	2A
Driver 4	Pin Digital 16	Pin Digital 15	Pin Digital 14	1.7A
Driver 5	Pin Digital 19	Pin Digital 18	Pin Digital 17	1.5A
Driver 6	Pin Digital 22	Pin Digital 21	Pin Digital 20	0.5A
Driver 7	Pin Digital 25	Pin Digital 24	Pin Digital 23	0.5A

11.3. Fabricación TPU

Para poder seleccionar el programa que quieras que el robot ejecute y lleve a cabo, se ha planificado la construcción de un mando que imita un TPU (Teach Pendant Unit) utilizado en los brazos robots industriales para crear y seleccionar los programas que quieres que ejecute el brazo robot. En este caso el TPU que se ha recreado tiene la función de seleccionar y visualizar en la pantalla LCD el programa que se esté ejecutando en ese momento y la duración que tendrá hasta finalizarlo y volver a su posición de *Home*.

El motivo por el que se ha construido un TPU en este proyecto es para evitar tener que estar continuamente compilando los diversos programas desde el ordenador y conseguir más profesionalidad en el proyecto.

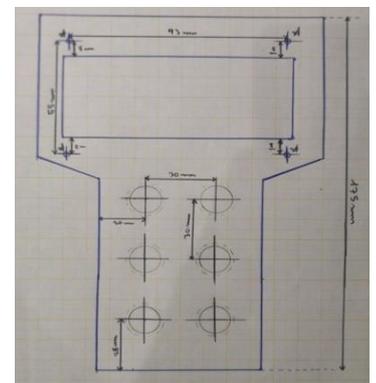


Fig. 26 Diseño a escala TPU



El material con el que se ha llevado a cabo la construcción del mando ha sido el aluminio para reducir el peso lo máximo posible ya que debe de ser lo más manejable posible. El proceso de fabricación ha consistido en plantear las medidas en una chapa de aluminio, hacer los respectivos cortes para doblar y soldar las 2 partes del TPU.



Fig. 27 Estructura en aluminio TPU



Fig. 28 Teach Pendant Unit (TPU)

El TPU cuenta con 6 botones momentáneos para los 6 distintos programas que se puedan compilar, además contiene una LCD de 20x4 para mostrar el programa que esté ejecutando el brazo robot y el tiempo que este tardará en finalizar. La conexión entre la caja de drivers y el TPU ha sido mediante el conector VGA que tiene 15 pines, 1 de los cuales no ha sido utilizado por si en algún futuro se decide ampliar las funciones del TPU.

Para conseguir mayor protección y comodidad a la hora de usar el TPU se ha diseñado con FilaFlex (material flexible que se puede imprimir en impresora 3D) dos tapas para proteger de golpes y caídas la parte superior e inferior.

Para resumir el cableado del interior del TPU se ha diseñado un circuito donde están soldadas las resistencias pertinentes a cada botón a más a más un LED de encendido. Esta placa se ha diseñado y soldado sobre una placa de circuito impreso ya que es la forma más fácil de diseñar tus propias placas con los elementos que necesites incorporar, en este caso se han incorporado 6 resistencias de 10K Ohmios para los botones, una resistencia de 220 Ohmios para un LED color amarillo y unos conectores para conectar todos estos elementos a la placa.

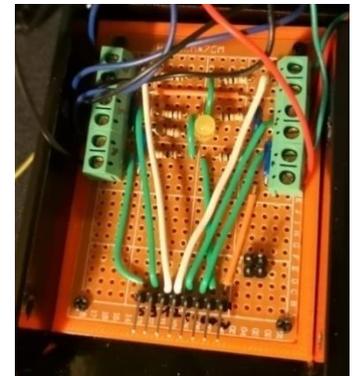


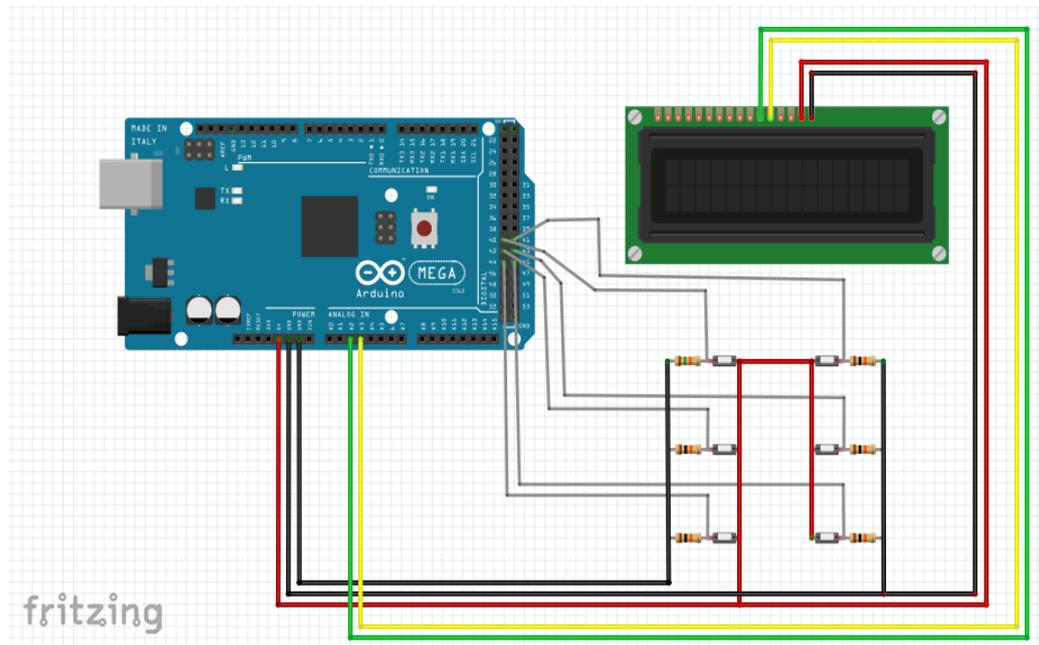
Fig. 29 Diseño propio en placa de prototipado

Los pines utilizados han sido los siguientes:

Pines Digitales Arduino MEGA	Pines LCD	Pines Botones Momentaneos
Pin 32	RS	-
Pin 33	E	-
Pin 35	D4	-
Pin 36	D5	-
Pin 37	D6	-
Pin 32	D7	-
Pin 26	-	Botón 1
Pin 27	-	Botón 2
Pin 28	-	Botón 3
Pin 29	-	Botón 4
Pin 30	-	Botón 5
Pin 31	-	Botón 6



A continuación se mostrará el esquema de conexionado:



11.4. Fabricación Brazo robot

El proceso de construir el brazo robot comenzó pensando qué tipo de brazo iba a construir y cómo lo iba a construir, finalmente decidí hacer un brazo de estructura mecánica angular ya que mi pensamiento era hacer un brazo robot lo más funcional y semejante a la vida real así que la mejor opción fue esta. El siguiente paso consistió en hallar la forma de construir la estructura, esta fue utilizando la impresora 3D.

11.4.1. Fabricación del brazo utilizando una impresora 3D

El primer paso consistió en buscar alguna estructura de brazo robot angular *open source* para poder descargar los archivos de la estructura del brazo robot y ahorrar tiempo de diseño y posibles fallos de medida. El archivo que encontré fue de un brazo robot llamado Arm Bot de la casa Roboteurs, este brazo cuenta con 6 grados de libertad movidos por 7 motores paso a paso de los cuales se hablará en los componentes del brazo robot.

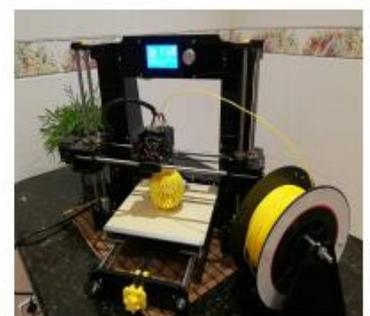


Fig. 30 Anet A6

Una vez encontrados cada uno de los más de 20 archivos de piezas del brazo lo siguiente era laminar los archivos “.stl” con el programa CURA y poner los parámetros de grosor de capa, relleno y tipo de relleno que se adecue a las funciones que van a tener esas piezas



en el brazo robot. Cuando está laminada la pieza en CURA pasa a tener un archivo “.gcode”, este es el que leerá la impresora para poder imprimir la pieza. Las piezas han sido impresas en dos impresoras distintas, una de ellas es la Hephestos 2 de bq y la otra es la Anet A6. El tipo de material utilizado en la estructura del brazo robot es el Ácido Poliláctico o comúnmente conocido como PLA, este material tiene características excelentes como torsión mecánica, resistencia y dureza lo que lo hace perfecto para cada una de las piezas del brazo robot. Los colores elegidos son el naranja como color principal y el negro como secundario imitando así los colores de un brazo robot de la marca KUKA. En total se ha utilizado más de 2 kg de PLA en el brazo y más de 100 horas de impresión siendo 26 horas de impresión continua la pieza más grande del brazo.

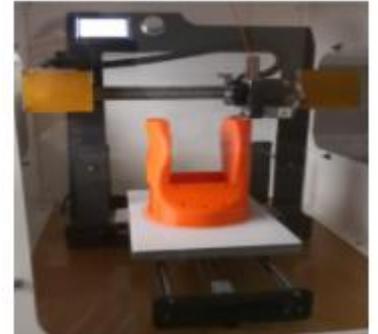
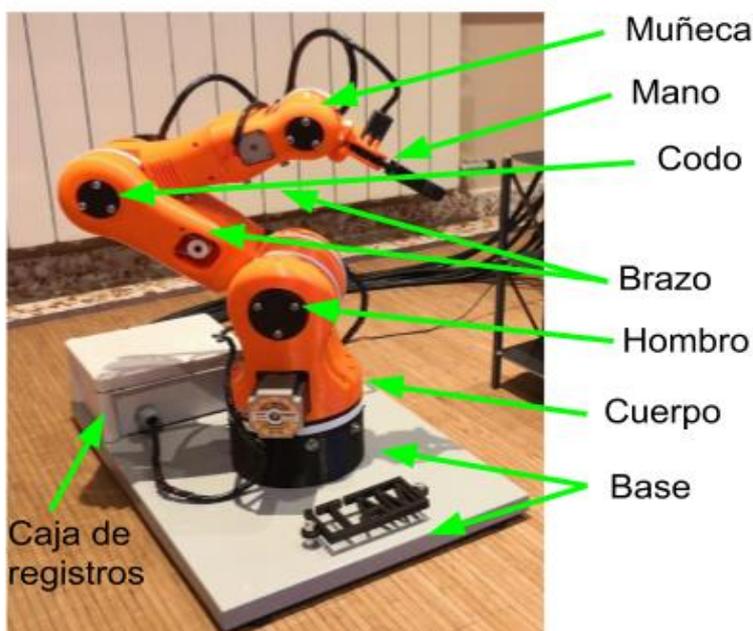


Fig. 31 Hephestos 2

11.4.2. Ensamblaje del brazo robot

Para ensamblar todas las piezas de brazo robot se utilizaron tornillos de acero inoxidable con cabeza hexagonal de 16 tamaños distintos, se eligieron estos tornillos por la resistencia al tiempo y a la corrosión. Para el movimiento de los 6 distintos ejes que componen el brazo se utilizaron cojinetes japoneses de 3 tamaños distintos y 5 varillas lisas para hacer de eje entre los cojinetes. Toda la estructura del brazo robot está atornillada a una base hecha de metal y pintada de color gris para que esté bien sujeto y que sea cómodo de transportar. La transmisión de movimiento motor paso a paso - eje se debe al uso de poleas dentadas y correas dentadas, estas están reforzadas por alambre para asegurar durabilidad en el brazo y conseguir un mayor rendimiento.

Su estructura es la siguiente:





11.4.3. Distribución del cableado en el brazo robot

Es importante hacer una correcta instalación del cableado proveniente de los 7 motores paso a paso y 1 servo porque esto permitirá tenerlo todo bien organizado y en caso de avería poder detectarla y solucionarla de la manera más fácil posible, también su correcta organización es garantía de que los cables no impedirán al brazo moverse con libertad sin estar impedido por algún cable entrelazado.

En el caso de este robot la instalación se ha hecho de la siguiente manera: primero los cables de todos los motores han sido recogidos por un pasa cables trenzado y este conducido por el brazo, siempre intentando no impedir la fluidez de los ejes, hasta una caja estanca donde serán agrupados en una regleta de conexiones y posteriormente identificados. En el otro lado de la bornera se conecta otro grupo de cables de la misma sección del anterior y estos salen de la caja por un tubo de *TubiClair* recubierto por una funda de nylon trenzado para darle una mejor estética, al final de este tubo va insertado un conector macho de tres pines estanco que es el que va al conector hembra de la caja. Cada motor tiene su propia manguera con su respectivo conector, en total 7 mangueras que van conectadas a la caja controladora. También de la caja instalada en la base del brazo robot sale un cable de 4 hilos con su respectivo conector que también va a la caja controladora que es el que pertenece al servo.



Fig. 32 Cableado brazo robot

11.4.4. Elementos con los que interactúa el brazo robot

Para poder simular el movimiento de un brazo robot industrial la mejor opción es verlo trabajar, para esto a su alrededor tiene un conjunto de estructuras como una estantería o un dispensador de piezas donde el brazo puede desarrollar varios movimientos como coger piezas del dispensador para colocarla en un lugar concreto de la estantería o sacar la pieza para colocarla en otro sitio, etc. Estas estructuras están instaladas sobre una base del mismo material, color y estructura que la del brazo robot pero con diferentes dimensiones, esto está hecho así para poder modular las estructuras pudiendo así conseguir mucha variedad de movimientos y procesos. La pieza que se utiliza como ejemplo está diseñada para poder dejarla en cualquier posición, está impresa en la impresora 3D con PLA rojo y tiene unas dimensiones de 90cm de largo por 30cm de ancho.



Fig. 33 Modulo 1



Fig. 34 Modulo 2



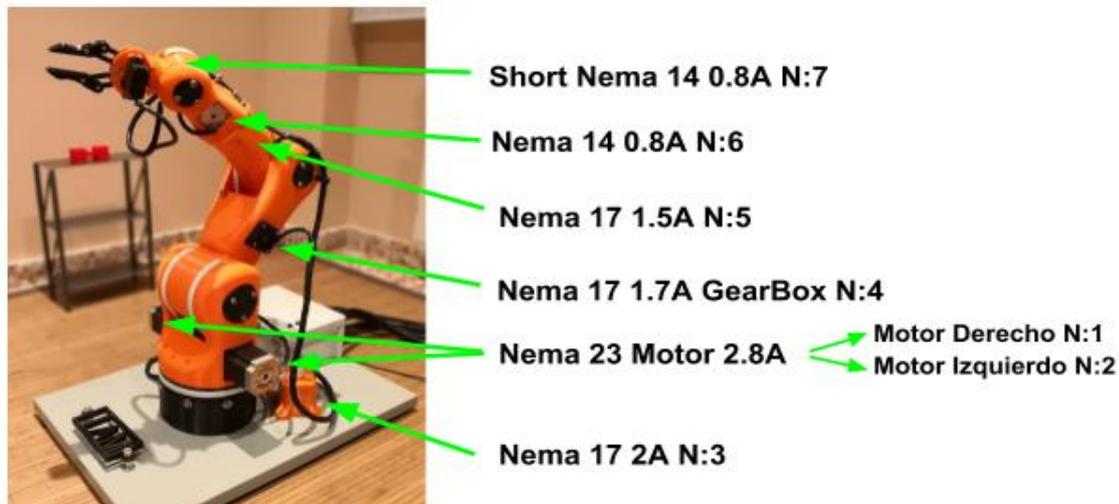
11.5. Componentes Brazo robot

El brazo robot como tal no tiene ningún circuito que le haga pensar por sí solo, este recibe las órdenes que le llegan desde la caja controladora por lo tanto si el brazo no está conectado a la caja controladora no va a actuar por sí solo. Los únicos elementos electrónicos que reciben órdenes desde la caja de drivers son los 7 motores paso a paso y el servo instalado en la pinza.

Los siguientes elementos los veremos a continuación bien detallados:

11.5.1. Motores P.A.P. (Paso a paso)

Un motor paso a paso es un dispositivo electromagnético que convierte una serie de impulsos electromagnéticos en desplazamiento angular, permitiendo así girar una cantidad de grados llamados pasos, estos se controlan mediante la frecuencia y el tiempo de reacción. Un motor paso a paso presenta las ventajas de tener precisión y fuerza lo que es fundamental cuando hablamos de brazos robóticos industriales pero la desventaja de que no puede trabajar sin la ayuda de un Driver/Controlador que le envíe los pulsos necesarios para trabajar.



Este Brazo robot consta de 7 motores paso a paso de amperajes distintos distribuidos dependiendo la zona y la cantidad de peso que tenga que mover.

En la imagen podemos visualizar la ubicación de cada motor paso a paso, su correspondiente modelo, número de motor y el amperaje con el que trabaja.

Como se puede observar en la foto los motores con mayor amperaje son los que están situados en la parte más baja del brazo ya que son los que tienen que levantar toda la estructura.



Algunas cosas a destacar de estos motores son el ruido que hacen causado por la frecuencia a la que los motores paso a paso trabajan. Otra cosa a destacar es el estado de energización, el estado de energización es cuando un motor está energizado pero no le está llegando información para que el motor se mueva por lo tanto el motor está parado pero manteniendo la fuerza en el eje consiguiendo así que el brazo robot se mantenga estable y no pierde la fuerza.

11.5.2. Servomotor (Gripper)

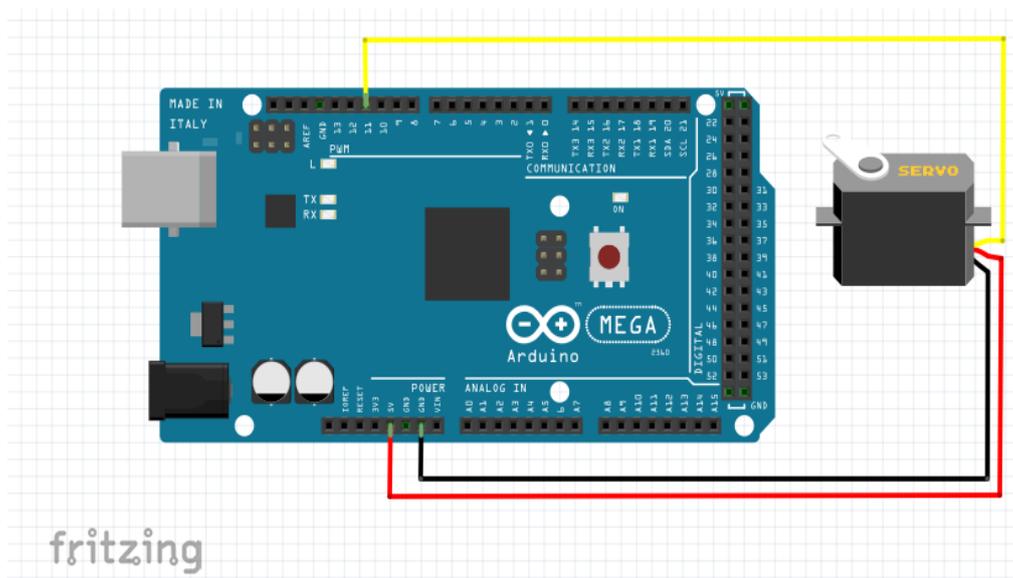
El servomotor cumple la función de abrir y cerrar el *gripper* (pinza). Un servomotor funciona por los grados que tiene que girar, en este caso los grados empleados han sido 60° para abrirse la pinza totalmente y -60° para volver a su posición original de cierre.

El programa de un servomotor se complementa con una librería llamada `<Servo.h>` con la que quedaría una estructura del programa como la siguiente: declarar la librería del servo, declarar el pin al cual está conectado el servo y por último el programa donde se le ordena al servo girar 60° para abrir la pinza y esperar unos segundos para volver a girar -60° y volver a cerrarse la pinza con el objeto.

El esquema de conexionado sería el siguiente:



Fig. 35 Servomotor





11.6. Presupuesto brazo robot

El presupuesto se ha hecho en base a los costes que tienen fabricar un robot de 6 ejes con una estructura hecha en impresión en 3D (las horas impresas y el mantenimiento de la máquina no han sido valorados en este presupuesto) y programado con unos drivers TB6600 acompañado de un Arduino Mega. La base del robot, la estructura metálica y el TPU están hechos a medida por Helcarinox para alojar todos los componentes electrónicos correctamente.

Aquí podemos ver representado dicho presupuesto:

PRESUPUESTO				
CATEGORÍA	CANTIDAD	ITEM	VALOR UNITARIO	VALOR TOTAL
Estructura	2	Bobina color Naranja (SMARTFIL)	23,23 €	46,46 €
	1	Bobina color Negro (SMARTFIL)	23,23 €	23,23 €
Electrónica	1	Enchufe corriente entrada macho y cable	3,13 €	3,13 €
	7	Clavijas SODIAL(R) 4 Conectores	3,89 €	27,23 €
	1	Arduino MEGA	40,00 €	40,00 €
	1	Arduino UNO	20,00 €	20,00 €
	1	Fuente de alimentación 24V 10A (220V)	14,53 €	14,53 €
	7	Drivers Quimat TB6600	12,59 €	88,13 €
	1	Pantalla TFT 2.8 Pulgadas	13,99 €	13,99 €
	1	Inkibird Controlador de temperatura relé	14,99 €	14,99 €
	1	Tira LED	15,00 €	15,00 €
	1	Ventilador 220V	30,00 €	30,00 €
	2	Botón Encendido Apagado SODIAL(R)	2,34 €	4,68 €
	1	Seta parada de emergencia	1,46 €	1,46 €
	1	8 metros de cable 1.5mm ²	0,40 euros el metro	3,20 €
	1	2 metros de cable 2.5mm ²	1,20 euros el metro	2,40 €
	1	20m Cable 4 hilos 0.5mm ²	1,04 euros el metro	20,86 €
	1	Funda de nylon trenzado 15m	6,08 €	6,08 €
	1	Manguera TubiClair 10.5m	0,37 euros el metro	3,88 €
	1	Caja estanca para conexiones	6,20 €	6,20 €
	1	Regleta con encaje caja estanca	11,79 €	11,79 €
	Cuadro eléctrico	1	Scheinder Interruptor diferencial 40A	14,96 €
1		LOGO Siemens	145,00 €	145,00 €
1		Regleta con encaje	25,00 €	25,00 €
1		Transformador 24-28V	19,09 €	19,09 €
2		Scheinder Interruptor magnetotérmico	9,94 €	19,88 €



Movimiento	3	Nema 23 Motor 2.8A	20,00 €	60,00 €
	1	Nema 17 Motor 2A	15,99 €	15,99 €
	1	Nema 17 Motor 1.7A	14,99 €	14,99 €
	1	Nema 17 Motor 1.7A 1:5 Gear Box	38,54 €	38,54 €
	2	Nema 14 Motor 0.8A	13,69 €	27,38 €
	1	Micro Servo Motor 9g	3,00 €	3,00 €
	1	Polea dentada	40,99 €	40,99 €
Tornilleria	-	Tornillería Varia	-	24,00 €
	-	Cojinetes	-	245,00 €
Estructura Metalica	1	Caja contenedora de la electrónica	319,00 €	319,00 €
	1	Base Brazo Robot	159,00 €	159,00 €
TPU	6	Pulsador momentáneo 12mm	1,08 €	6,49 €
	2	Conectores Hembra VGA	0,27 €	0,54 €
	1	Cable conector VGA Macho-Macho 5m	7,69 €	7,69 €
	1	LCD 20x4 + Modulo i2c	10,69 €	10,69 €
	1	Estructura TPU	80,00 €	80,00 €
	1	Bobina FilaFlex Goma protectora del TPU	15,95 €	15,95 €
TOTAL(EUROS)				1.684,24 €



12. CONCLUSIONES

Con este trabajo el objetivo era llegar a replicar de una manera reducida el funcionamiento y la estructura de un brazo robot así como todos los componentes que se encargan de que este funcione. El objetivo finalmente se ha conseguido pero esto ha llevado ciertos problemas y frustraciones que son necesarios comentar.

Antes de empezar un proyecto tan grande y complejo es necesario plantearse cómo lo vas a construir, qué materiales y herramientas vas a usar, en caso de tener algún imprevisto donde vas a acudir (en mi caso fueron foros y páginas web) y el coste que esto va a conllevar. Una vez se ha planteado todo lo anterior y estás decidido a seguir adelante el siguiente paso será planificar todo el trabajo, quizás esta sea la parte más importante. Unos de mis principales retos fueron encontrar todos los componentes y piezas como motores, drivers, fuentes de alimentación... Yo recurrí a Amazon ya que tiene una gran variedad de productos en stock y el envío es bastante rápido. Otro principal problema fue alimentar todos los componentes situados en la caja de drivers como controladores y arduinos, este problema lo solventé dividiendo todos los componentes en dos grupos, el grupo de arduinos alimentado por una fuente de alimentación convencional y el otro grupo que eran los drivers a los que les conecte una fuente con la capacidad de alimentarlos a todos y cada uno de los 7 drivers. Otro problema fue la disipación del aire acumulado dentro de la caja y provocado por los potentes drivers, esto se solventó instalando un sistema de circulación del aire compuesto por un ventilador y su respectivo controlador. Y el último de los problemas a tener en cuenta fue el transporte ya que las dimensiones y el peso eran un gran hándicap a la hora de transportarlo, la solución a esto fue dividirlo en módulos como la caja de drivers, el brazo robot y los dos módulos con los que el brazo robot interactúa.

Uno de los problemas que se ha presentado y que no se ha podido solucionar ha sido conseguir un movimiento fluido a la hora de moverse el brazo. Para conseguir este movimiento es necesario que el brazo mueva más de un solo eje al mismo tiempo, esto no era posible de conseguir por la manera con la que se programa el movimiento del brazo robot y me hubiera gustado solventarlo.

En conclusión la construcción de un brazo robot es un proceso laborioso por la cantidad de problemas que se presentan al momento de estar fabricando un brazo robot a escala, pero al conseguir lograr retos como cablear la caja controladora de la forma más ordenada posible o ver el proyecto acabado devuelve una satisfacción que hace querer seguir ampliando el proyecto. Algunas mejoras que se le podrían aplicar al brazo robot serían perfeccionar el sistema de programación o ampliar las posibilidades de movimiento.



13. ÍNDICE DE ILUSTRACIONES

Fig. 1 Primer robot industrial:.....Pag. 7

Internet < https://es.wikipedia.org/wiki/Joseph_Marie_Jacquard#/media/File:Jacquard_loom.jpg >

Fig. 2 Robot de primera generación:.....Pag. 8

Internet < https://es.wikipedia.org/wiki/Caja_de_guantes >

Fig. 3 Robot de segunda generación:.....Pag. 8

Internet < https://es.wikipedia.org/wiki/Robot_industrial >

Fig. 4 Robot de tercera generación:.....Pag. 9

Internet < <https://itcomunicacion.com.mx/que-industrias-estan-ganando-en-inteligencia-artificial/> >

Fig. 5 Brazo robot industrial Soldador:.....Pag. 10

Internet < <https://www.thefabricator.com/article/automationrobotics/un-robot-para-soldaduraaen-continuo-movimiento> >

Fig. 6 Brazo robot industrial Traspalé:.....Pag. 10

Internet < <https://www.alamy.es/imagenes/industrial-robot-automation.html> >

Fig. 7 Brazo robot industrial Pintor:.....Pag. 10

Internet < <https://www.logismarket.com.mx/fanuc-robotics/robot-de-pintura/1639267833-1179566037-p.html> >

Fig. 8 Brazo robot industrial Prototipo 1:.....Pag. 10

Internet < <https://blog.bricogeek.com/noticias/robotica/doblando-ropa-con-dos-brazos-robots-industriales/> >

Fig. 9 Brazo robot industrial Prototipo 2:.....Pag. 10

Internet<https://www.google.com/search?biw=1366&bih=657&tbm=isch&sa=1&ei=MAYMXOmAG8P_sxgOr2o54&q=brazo+robot+industrial+que+cocina+vodafone&oq=brazo+robot+industrial+que+cocina+vodafone&gs_l=img.3...66959.71799..72251...0.0..0.152.19r62.8j11.....1....1..g7ws-wiz-img.hf_gZww-PU0#imgrc=Ba0BdQ6xGjMRWM >

Fig. 10 Movimiento brazo robot:.....Pag. 11

Internet < <https://www.lincolnelectric.com/es-es/equipment/robotic-automation/products-and-systems/Pages/automation-wire-feeders.aspx> >



Fig. 11 Esquema hardware brazo robot:.....Pag. 11

Elaboración propia.

Fig. 12 Estructura brazo robot:.....Pag. 12

Internet<http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/morfologia.htm>

Fig. 13 Dispositivos de entrada y salida:.....Pag. 16

Internet<http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/morfologia.htm>

Fig. 14 Célula de seguridad:.....Pag. 21

Internet <<https://safework.es/seguridad-de-una-celula-robotizada-mas/>>

Fig. 15 Seta parada de emergencia:.....Pag. 21

Internet <<https://www.pilz.com/es-ES/eshop/00104002147028/PITestop-emergency-stop-pushbuttons>>

Fig. 16 Caja de drivers sin pintar:.....Pag. 23

Elaboración propia.

Fig. 17 Caja de drivers pintada:.....Pag. 23

Elaboración propia.

Fig. 18 Caja de drivers montada:.....Pag. 24

Elaboración propia.

Fig. 19 Elementos de seguridad:.....Pag. 26

Internet <<https://blogmaterialelectrico.wordpress.com/2013/04/17/interruptores-automaticos-magnetotermico/>>

Fig. 20 Fuente de alimentación:.....Pag. 27

Internet <<http://www.cablematic.es/producto/Fuente-de-alimentacion-industrial-24VDC-10A/>>

Fig. 21 Sistema de ventilación:.....Pag. 27

Internet <<https://www.amazon.es/Inkbird-Temperatura-Controlador-Refrigeraci%C3%B3n-Calefaccion/dp/B01NAIAH4S>>

Fig. 22 LOGO Siemens:.....Pag. 28

Internet <<http://www.zonaindustrial.cl/product/6ed1052-1fb00-0ba6>>



Fig. 23 Pantalla TFT y Arduino UNO:.....Pag. 28

Internet <<http://www.electrohobby.org/tft-lcd-touch-2-4-shield-arduino-uno/>>

Fig. 24 Arduino MEGA:.....Pag. 28

Internet <<https://330ohms.com/products/arduino-mega-2560-r3>>

Fig. 25 Driver TB6600 y esquema:.....Pag. 29

Internet <<https://tienda.bricogeek.com/controladores-motores-tb6600.html>>

Fig. 26 Diseño a escala TPU:.....Pag. 29

Elaboración propia.

Fig. 27 Estructura en aluminio TPU:.....Pag. 30

Elaboración propia.

Fig. 28 Teach Pendant Unit (TPU):.....Pag. 30

Elaboración propia.

Fig. 29 Diseño propio en placa de prototipado:.....Pag. 30

Elaboración propia.

Fig. 30 Anet A6:.....Pag. 31

Elaboración propia.

Fig. 31 Hephestos 2:.....Pag. 32

Elaboración propia.

Fig. 32 Cableado brazo robot:.....Pag. 33

Elaboración propia.

Fig. 33 Modulo 1:.....Pag. 33

Elaboración propia.

Fig. 34 Modulo 2:.....Pag. 33

Elaboración propia.

Fig. 35 Servomotor:.....Pag. 35

Internet <<https://tienda.bricogeek.com/servomotores/564-servomotor-360-grados.html>>



14. BIBLIOGRAFÍA Y REFERENCIAS

Toda la información relacionada con el origen y la evolución ha sido extraída de: <<http://brazorobotico2.blogspot.com.es/2015/08/evolucion-y-origen.html>>.

Las fuentes consultadas para explicar que es un brazo robótico han sido en: <http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/industrial.htm>.

Para redactar el apartado de los tipos de brazos robots que pueden haber en la industria se han visitado las siguientes fuentes documentales:

<<http://automata.cps.unizar.es/robotica/Morfologia.pdf>>

<<http://rivasrobotics.com/tipos-de-robots-industriales/>>

Para saber que brazos robots son los más utilizados y porque es interesante la el siguiente enlace:

<<https://www.eurobots.es/>>.

Para entender cómo funciona un brazo robot es necesario entender su estructura: <http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/morfologia.htm>

Para comprender el funcionamiento de un brazo robot la información ha sido extraída del siguiente link:

<https://www.youtube.com/watch?v=il5Z9rJ_lwM>.

En el apartado de elementos que componen un brazo robot se hace especial mención en los motores paso a paso, pues bien la información se ha extraído de:

< <http://automata.cps.unizar.es/robotica/Morfologia.pdf>>.

El apartado de seguridad en un brazo robot era muy importante saber las distintas normas que están en vigor actualmente para hacer el trabajo de un brazo robot lo más seguro posible, estas leyes y datos han sido extraídas de 3 distintas localizaciones en internet:

< <http://www.conectapyme.com/documentacion/2006robotica.pdf>>

<<http://www.mekkam.com/robotica-industrial/medidas-de-seguridad-en-los-robots-industriales/>>

<<https://ocw.upc.edu/sites/all/modules/ocw/estadistiques/download.php?file=14589/2011/1/54326/40604-3452.pdf>>.

Para dar pie a la idea de que la extensión del brazo robot al ámbito industrial está siendo más productivo que el ser humano era indispensable consultar algún documento donde se vea reflejado el aumento de la productividad, pues bien este es el documento:

< <https://www.escoeuniversitas.com/robots-250-mas-productivos/>>.

La tabla donde se muestran los programas asociados a cada una de las marcas de los brazos robots ha sido extraída de la siguiente página:

<https://revistadigital.inesem.es/gestion-integrada/files/2015/09/ScreenHunter_12-Sep.-02-12.151.jpg>



15. ANEXOS

En este apartado se hará mención de las características de los componentes principales incluidos en el brazo robot y de los programas empleados en este proyecto.

15.1. Datasheet Arduino UNO

Descripción general

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 16 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 conector USB, un conector de alimentación, una cabecera ICSP y un botón de reset. 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.

Especificaciones técnicas

Microcontrolador	ATmega328
Tensión de trabajo	5V
Tensión de entrada (recomendada)	7-12V
Tensión de entrada (límite)	6-20V
Pines Digitales I/O	16 (de los cuales 6 proporcionan salida PWM)
Pines de entradas Analógicas	6
DC Corriente por Pin I/O	40 mA
DC Corriente por Pin 3.3V	50 mA
Memoria Flash	32 KB de los cuales 0.5 KB se usan por el bootloader



15.1.1. Código Arduino UNO

La placa Arduino UNO tiene la función de controlar únicamente la pantalla TFT. La programación de la pantalla TFT está dividida en: Librerías requeridas, definición de los pines utilizados, código de colores, iniciación de la pantalla, primera ilustración con el nombre del robot, segunda ilustración con el logo TAM y tercera ilustración con el logo del Helcarinox S.L., todo esto en un bucle de 5 segundos por pantalla.

El código es el siguiente:

```
#include <Adafruit_TFTLCD.h>
#include <pin_magic.h>
#include <registers.h>
#include <Adafruit_GFX.h> // Librería de gráficos
#include <Adafruit_TFTLCD.h> // Librería de LCD

// Pines de conexión del LCD
#define LCD_CS A3 // Chip Select - Pin analógico 3
#define LCD_CD A2 // Command/Data - Pin Analógico 2
#define LCD_WR A1 // LCD Write - Pin Analógico 1
#define LCD_RD A0 // LCD Read - Pin Analógico 0
#define LCD_RESET A4 // LCD Reset - Pin Analógico 4

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

//Códigos de colores
#define BLACK 0x0000 // Definimos los colores para poder referirnos a ellos con su nombre
#define RED 0xF800
#define Navy 0x000F
#define DarkGreen 0x03E0
#define DarkCyan 0x03EF
#define Maroon 0x7800
#define Purple 0x780F
#define Olive 0x7BE0
#define LightGrey 0xC618
#define DarkGrey 0x7BEF
#define Blue 0x001F
#define Green 0x07E0
#define Cyan 0x07FF
#define Red 0xF800
#define Magenta 0xF81F
#define Yellow 0xFFE0
#define White 0xFFFF
#define Orange 0xFD20
#define GreenYellow 0xAFE5
#define Pink 0xF81F

void setup(){

tft.begin(0x9341); // Iniciamos el LCD especificando el controlador de nuestro LC. En este caso el ILI9341.

tft.fillScreen(LightGrey); // Colocamos el fondo del LCD en Negro
}

void loop(){

tft.setRotation(1); // Establecemos la posición de la pantalla Vertical u Horizontal
////////////////////////////////////
tft.setCursor(70, 85); // Situamos el cursor en la posición del LCD deseada,

tft.fillScreen(LightGrey); // Colocamos el fondo del LCD en Negro

tft.drawRoundRect(45, 65, 220, 110, 20, Orange); // Dibujamos un cuadrado/rectángulo relleno de color
//((Punto inicial X, Punto inicial Y, Longitud X, Longitud Y, Color)

tft.setTextSize(10); // Definimos tamaño del texto.
tft.setTextColor(Orange); // Definimos el color del texto

tft.println("TAM"); // Escribimos nuestro texto en el LCD. Realizará un salto de línea

delay(5000);
```



```
tft.fillScreen(LightGrey); // Colocamos el fondo del LCD

tft.setCursor(65,35); // Situamos el cursor en la posición del LCD deseada,
tft.setTextSize(8); // Definimos tamaño del texto.
tft.setTextColor(Orange); // Definimos el color del texto
tft.println("RBX1"); // Escribimos nuestro texto en el LCD. Realizará un salto de línea
tft.setCursor(50, 115 ); // Situamos el cursor en la posición del LCD deseada,
tft.setTextSize(4); // Definimos tamaño del texto.
tft.setTextColor(Orange); // Definimos el color del texto
tft.println("Robot Arm"); // Escribimos nuestro texto en el LCD. Realizará un salto de línea
tft.setCursor(25, 170); // Situamos el cursor en la posición del LCD deseada,
tft.setTextSize(5); // Definimos tamaño del texto.
tft.setTextColor(Orange); // Definimos el color del texto
tft.println("ROBOTEURS"); // Escribimos nuestro texto en el LCD. Realizará un salto de línea

delay(5000);

tft.fillScreen(White); // Colocamos el fondo del LCD en Negro
tft.fillRoundRect(24, 68, 26, 26, 5,LightGrey); // Dibujamos un cuadrado/rectángulo relleno de color
//((Punto inicial X, Punto inicial Y, Longitud X,Longitud Y, Color)
tft.fillRoundRect(24, 97, 26, 26, 5,DarkGrey); // Dibujamos un cuadrado/rectángulo relleno de color
tft.fillRoundRect(52, 75, 20, 20, 5,DarkGrey); // Dibujamos un cuadrado/rectángulo relleno de color
tft.fillRoundRect(52, 97, 20, 20, 5,LightGrey); // Dibujamos un cuadrado/rectángulo relleno de color
tft.fillRoundRect(74, 81, 14, 14, 5,LightGrey); // Dibujamos un cuadrado/rectángulo relleno de color

tft.drawLine( 60, 170, 256, 170, BLACK); //Dibujamos una línea recta Negra
tft.setCursor(75, 175); // Situamos el cursor en la posición del LCD deseada,
tft.setTextSize(1); // Definimos tamaño del texto.
tft.setTextColor(Blue); // Definimos el color del texto
tft.println("FABRICACION Y CONSTRUCCIONES"); // Escribimos nuestro texto en el LCD. Realizará un salto de línea
tft.setCursor(100, 187); // Situamos el cursor en la posición del LCD deseada,
tft.setTextSize(1); // Definimos tamaño del texto.
tft.setTextColor(Blue); // Definimos el color del texto
tft.println("EN ACERO INOXIDABLE"); // Escribimos nuestro texto en el LCD. Realizará un salto de línea
tft.drawLine( 60, 197, 256, 197, BLACK); //Dibujamos una línea recta Negra
tft.setCursor(30, 120); // Situamos el cursor en la posición del LCD deseada,
tft.setTextSize(3); // Definimos tamaño del texto.
tft.setTextColor(Blue); // Definimos el color del texto
tft.println("Helcarinox s.l."); // Escribimos nuestro texto en el LCD. Realizará un salto de línea

delay(5000);
}
```

15.2. Datasheet Arduino MEGA

Descripción general

La Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serie de hardware), una conexión USB, un conector de alimentación, un conector ICSP, y un botón de reset. Con conectar la placa a un ordenador con un cable USB o a la corriente con un adaptador de CA a CC o una batería la placa estaría lista para empezar. La placa Mega 2560 es compatible con la mayoría de los shield para el Uno y las placas anteriores Duemilanove o Diecimila.

Especificaciones técnicas

Microcontrolador

ATmega2560



Tensión de trabajo	5V
Tensión de entrada (recomendada)	7-12V
Tensión de entrada (límite)	6-20V
Pines Digitales I/O	54 (de los cuales 15 proporcionan salida PWM)
Pines de entradas Analógicas	16
DC Corriente por Pin I/O	20 mA
DC Corriente por Pin 3.3V	50 mA
Memoria Flash	256 KB de los cuales 8 KB se usan por el bootloader

15.2.1. Código Arduino MEGA

La placa Arduino MEGA al ser la encargada de controlar todos los componentes que componen el brazo robot es la que mayor extensión de código tiene. El código empleado se puede dividir en 3 partes: la primera sería la declaración de las dos librerías encargadas de facilitar el código de la pantalla LCD 20*4 ubicada en el TPU y la segunda encargada del servomotor con la función de cerrar y abrir la pinza, a más a más de esto en la primera parte del código podemos encontrar la declaración de los 38 pines digitales 25 de los cuales dirigidos a los 7 drivers, 12 al TPU y 1 al servomotor. La segunda parte sería la relacionada con el *Void Setup** declarando todos los pines según sean entradas a la placa o salidas, aparte de esto también podemos encontrar el inicio del código que escribe en la pantalla "PROGRAMA SELECCIÓN:" para que sea lo primero que veamos en la pantalla antes de pulsar ningún botón. La tercera y última parte sería el bucle *Void Loop** donde aparece el código de todas las funciones y secuencias que el brazo tiene que seguir, inicialmente en este bucle la placa esta a la espera de que alguno de los 6 pulsadores este presionado para poder escribir en la pantalla el programa seleccionado, su duración y ejecutar el tipo de movimiento programado y seleccionado para ese programa. Cada botón tiene un programa distinto al otro para poder ampliar el abanico de posibles movimientos interactuando con el entorno.



El código es el siguiente:

```
#include <LiquidCrystal.h> //Incluye la Librería LiquidCrystal para la LCD del TPU
LiquidCrystal lcd(32, 33, 35, 36, 37, 38); //Declaración de los pines de la LCD Parámetros: (rs, enable, d4, d5, d6, d7)

// Declaración de los 6 botones del TPU
int boton1 = 26;
int boton2 = 27;
int boton3 = 28;
int boton4 = 29;
int boton5 = 30;
int boton6 = 31;

#include <Servo.h> //Librería del Servomotor
Servo myservo;
const int pinServo = 11; // Declaración del pin del Servomotor

//Declaración de los pines de los Drivers
int PUL1=4; //Pines Driver Motor 1
int DIR1=3;
int ENA1=2;

int PUL2=7; //Pines Driver Motor 2
int DIR2=6;
int ENA2=5;

int PUL3=10; //Pines Driver Motor 3
int DIR3=9;
int ENA3=8;

int PUL4=14; //Pines Driver Motor 4
int DIR4=15;
int ENA4=16;

int PUL5=17; //Pines Driver Motor 5
int DIR5=18;
int ENA5=19;

int PUL6=20; //Pines Driver Motor 6
int DIR6=21;
int ENA6=22;

int PUL7=23; //Pines Driver Motor 7
int DIR7=24;
int ENA7=25;

void setup() {
  pinMode(boton1, INPUT); //Declaración entrada de los pines de los 6 botones del TPU
  pinMode(boton2, INPUT);
  pinMode(boton3, INPUT);
  pinMode(boton4, INPUT);
  pinMode(boton5, INPUT);
  pinMode(boton6, INPUT);

  //La pantalla se inicia ilustrando "Programa Selección:"
  lcd.begin(20,4);
  lcd.setCursor(0,0);
  lcd.print("PROGRAMA SELECCION:");

  //Declaración salida de los pines de los drivers
  pinMode (PUL1, OUTPUT); //Pines Driver Motor 1
  pinMode (DIR1, OUTPUT);
  pinMode (ENA1, OUTPUT);

  pinMode (PUL2, OUTPUT); //Pines Driver Motor 2
  pinMode (DIR2, OUTPUT);
  pinMode (ENA2, OUTPUT);

  pinMode (PUL3, OUTPUT); //Pines Driver Motor 3
  pinMode (DIR3, OUTPUT);
  pinMode (ENA3, OUTPUT);

  pinMode (PUL4, OUTPUT); //Pines Driver Motor 4
  pinMode (DIR4, OUTPUT);
```



```
pinMode (ENA4, OUTPUT);

pinMode (PUL5, OUTPUT); //Pines Driver Motor 5
pinMode (DIR5, OUTPUT);
pinMode (ENA5, OUTPUT);

pinMode (PUL6, OUTPUT); //Pines Driver Motor 6
pinMode (DIR6, OUTPUT);
pinMode (ENA6, OUTPUT);

pinMode (PUL7, OUTPUT); //Pines Driver Motor 7
pinMode (DIR7, OUTPUT);
pinMode (ENA7, OUTPUT);

//El servomotor se inicia teniendo las pinzas abiertas
myservo.attach(pinServo);
myservo.write(0);
}

void loop() {
  //Bloque de condición del Programa 1 donde se pregunta el estado del botón 1, si el botón está en estado High=Pulsado se inicia el
  bucle pero si está en Low = NoPulsado el bucle no se iniciará.
  if(digitalRead(boton1) == HIGH && digitalRead(boton2) == LOW && digitalRead(boton3) == LOW && digitalRead(boton4) == LOW &&
  digitalRead(boton5) == LOW && digitalRead(boton6) == LOW) {

    lcd.setCursor(2,2);           //El cursor se situará en el pixel 2 de alto y 2 ancho
    lcd.print("Programa 1");      //Se imprime en la pantalla "Programa 1"
    lcd.setCursor(13,2);         //El cursor se situará en el pixel 2 de alto y 13 ancho
    lcd.print("02:30");           //Tiempo que tarda en finalizar el programa
    delay(500);                  //Tiempo de medio segundo

    //Conjunto de movimientos para el Brazo Robot guardados en el Programa 1

  }

  if(digitalRead(boton2) == HIGH && digitalRead(boton1) == LOW && digitalRead(boton3) == LOW && digitalRead(boton4) == LOW &&
  digitalRead(boton5) == LOW && digitalRead(boton6) == LOW ) {

    lcd.setCursor(2,2);
    lcd.print("Programa 2");
    lcd.setCursor(13,2);
    lcd.print("04:56");
    delay(500);

    //Conjunto de movimientos para el Brazo Robot guardados en el Programa 2

  }

  if(digitalRead(boton3) == HIGH && digitalRead(boton1) == LOW && digitalRead(boton2) == LOW && digitalRead(boton4) == LOW &&
  digitalRead(boton5) == LOW && digitalRead(boton6) == LOW ) {

    lcd.setCursor(2,2);
    lcd.print("Programa 3");
    lcd.setCursor(13,2);
    lcd.print("07:46");
    delay(500);

    //Conjunto de movimientos para el Brazo Robot guardados en el Programa 3

  }

  if(digitalRead(boton4) == HIGH && digitalRead(boton1) == LOW && digitalRead(boton2) == LOW && digitalRead(boton3) == LOW &&
  digitalRead(boton5) == LOW && digitalRead(boton6) == LOW ) {

    lcd.setCursor(2,2);
    lcd.print("Programa 4");
    lcd.setCursor(13,2);
    lcd.print("07:46");
    delay(500);

    //Conjunto de movimientos para el Brazo Robot guardados en el Programa 4

  }
}
```



```
if(digitalRead(boton5) == HIGH && digitalRead(boton1) == LOW && digitalRead(boton2) == LOW && digitalRead(boton3) == LOW &&
digitalRead(boton4) == LOW && digitalRead(boton6) == LOW ) {
  lcd.setCursor(2,2);
  lcd.print("Programa 5");
  lcd.setCursor(13,2);
  lcd.print("07:46");
  delay(500);

  //Conjunto de movimientos para el Brazo Robot guardados en el Programa 5
}

if(digitalRead(boton6) == HIGH && digitalRead(boton1) == LOW && digitalRead(boton2) == LOW && digitalRead(boton3) == LOW &&
digitalRead(boton4) == LOW && digitalRead(boton5) == LOW ) {

  lcd.setCursor(2,2);
  lcd.print("Programa 6");
  lcd.setCursor(13,2);
  lcd.print("07:46");
  delay(500);

  //Conjunto de movimientos para el Brazo Robot guardados en el Programa 6
}
}
```

15.3. Datasheet Driver TB6600

Descripción general

Es un controlador de motor paso a paso profesional y fácil de usar, que podría controlar un motor paso a paso de dos fases. Es compatible con Arduino y otros microcontroladores que pueden emitir una señal de pulso digital de 5V. El controlador del motor paso a paso TB6600 tiene una entrada de alimentación de amplio rango, fuente de alimentación de 9 ~ 42VDC. Y es capaz de generar una corriente máxima de 4A, que es suficiente para la mayoría de los motores paso a paso.

El controlador paso a paso soporta control de velocidad y dirección. Tiene la función de configurar su micro paso y la corriente de salida con 6 interruptores DIP. Hay 7 tipos de micro pasos (1, 2 / A, 2 / B, 4, 8, 16, 32) y 8 maneras de control de corriente (0.5A, 1A, 1.5A, 2A, 2.5A, 2.8A, 3.0 A, 3.5A).

Como dispositivo profesional, es capaz de conducir motores paso a paso tales como: 57, 42 de tipo bifásico, cuatro fases, motor paso a paso híbrido.

Nota: ésta es una versión de actualización del controlador del motor paso a paso TB6600.

Especificaciones técnicas

Corriente de entrada:	0 ~ 5A
Corriente de salida:	0.5 ~ 4.0A
Voltaje de entrada:	3.3 ~ 24V



Potencia (MAX):	160W
Paso micro:	1, 2 / A, 2 / B, 4, 8, 16, 32
Temperatura:	-10 ~ 45 °C
Humedad:	Sin condensación
Peso:	0,2 kg
Dimensión:	96 * 71 * 37 mm

15.3.1. Código Arduino MEGA para los Drivers TB6600

Este código es el que iría dentro de los espacios donde pone “//Conjunto de movimientos para el Brazo Robot guardados en el Programa (1, 2, 3, 4, 5, 6)” para que cuando se pulse el botón del bucle **if** se active ejecutando este programa dentro del anterior programa. Pues bien este es el programa encargado de mover los motores paso a paso según el tiempo que estén activos y su velocidad.

El código es el siguiente:

```
//Definir los 3 pines de un Driver
int PUL=14; //define Pulse pin
int DIR=15; //define Direction pin
int ENA=16; //define Enable Pin

void setup() {
  pinMode (PUL, OUTPUT);
  pinMode (DIR, OUTPUT);
  pinMode (ENA, OUTPUT);
}

void loop() {
  for (int i=0; i<7000; i++) //Controlas las vueltas y así el tiempo de funcionamiento del motor, cuanto mayor es el numero más vueltas
    dará y más tiempo está girando.
  {
    digitalWrite(DIR,LOW); //dirección PAP si es LOW(Derecha) si es HIGH(Izquierda)
    digitalWrite(ENA,HIGH);
    digitalWrite(PUL,HIGH);
    delayMicroseconds(100); //Velocidad cuanto mayor es el número menor es la velocidad
    digitalWrite(PUL,LOW);
    delayMicroseconds(300); //Velocidad cuanto mayor es el número menor es la velocidad
  }
  for (int i=0; i<4000; i++) //Controlas las vueltas y así el tiempo de funcionamiento del motor, cuanto mayor es el numero más vueltas
    está y más tiempo está girando.
  {
    digitalWrite(DIR,HIGH); //dirección PAP si es LOW(Derecha) si es HIGH(Izquierda)
    digitalWrite(ENA,HIGH);
    digitalWrite(PUL,HIGH);
    delayMicroseconds(100); //Velocidad cuanto mayor es el número menor es la velocidad
    digitalWrite(PUL,LOW);
    delayMicroseconds(1000); //Velocidad cuanto mayor es el número menor es la velocidad
  }
}
```