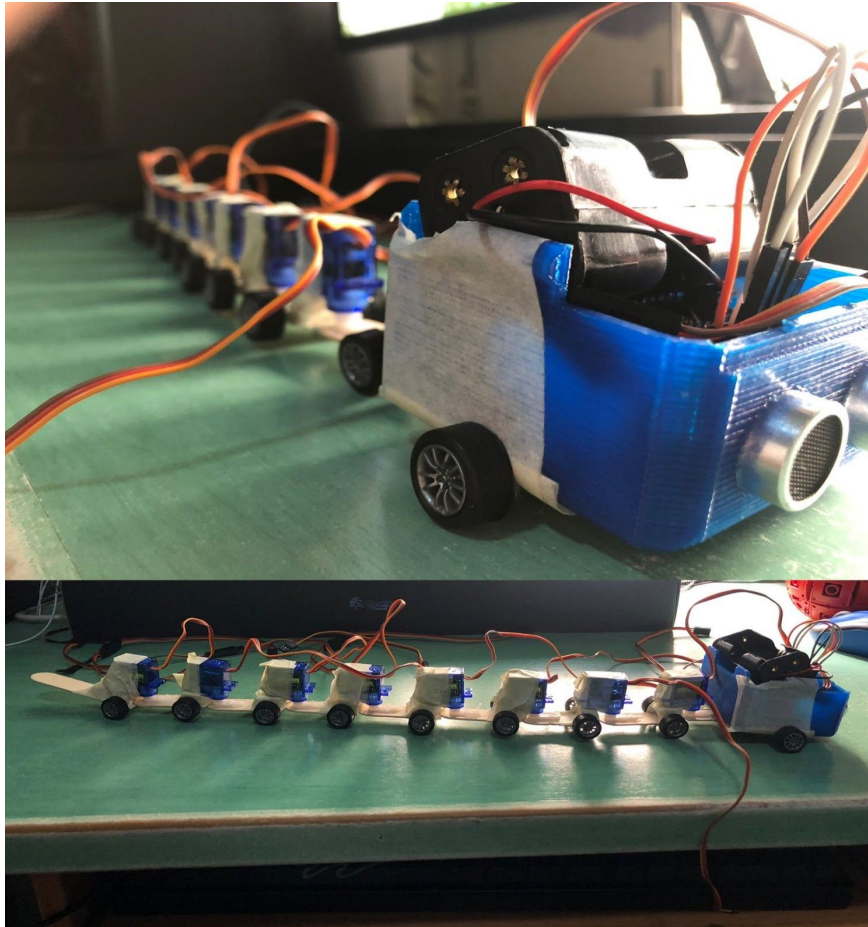




Institut Puig Castellar

Santa Coloma de Gramenet



Project: Robo Snake

CFGM Sistemes Microinformàtics i Xarxes

Joel Rodrigálvarez y Alan Rodríguez
SMX2A
2019-2020



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2020 Joel Rodrigalvarez y Alan Rodríguez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© Joel Rodrigalvarez Peñaranda
Alan Rodríguez Macia

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

Resumen del proyecto:

Nuestro proyecto consiste en crear y programar con arduino, un robot inspirado en una serpiente terrestre, para que pueda moverse de manera autónoma.

Tendrá sensores de distancia, de esta manera podrá esquivar los obstáculos. El cuerpo será un exoesqueleto, y tanto los servos y los cables irán dentro. Con Servo Motores, conseguiremos que se mueva de una forma muy similar a una serpiente real, con su característico zic-zac. En la parte de abajo tendrá unas pequeñas ruedas para poder moverse sin problemas.

Creemos que será un robot complicado pero muy satisfactorio de ver el resultado, poca gente ha hecho serpientes robot y puede ser una buena idea.

Palabras clave:

Arduino, robot, autónomo, serpiente.

Abstract:

Our project is to create and program with arduino, a robot inspired by a terrestrial snake, so that it can move autonomously.

It will have distance sensors, this way you can avoid obstacles. The body will be an exoskeleton, and both the servos and the cables will go inside. With Servo Motors, we will make it move in a very similar way to a real snake, with its characteristic zic-zac. At the bottom you will have small wheels to be able to move without problems.

We think it will be a complicated but very satisfactory robot to see the result, few people have made robot snakes and it may be a good idea.

Keywords:

Arduino, robot, autonomous, snake.

Índice

1. Introducción	7
1.1 Contexto	7
1.2 Justificación	7
1.3 Objetivos	7
1.3.1 Objetivos generales	7
1.3.2 Objetivos específicos	8
1.4 Estrategia y planificación del proyecto	8
1.5 Metodología de trabajo	9
1.6 Estudio económico y presupuestario.	9
2 Descripción del proyecto	11
2.1 Análisis de requisitos [proyecto de desenvolvimiento]	11
2.1.1 Requisitos funcionales	11
2.1.2 Requisitos no funcionales	11
2.2 Previsión de tareas de investigación [proyecto de investigación]	11
2.3 Tecnologías	11
2.3.1 Comparativa de las tecnologías valoradas	11
2.3.2 Tecnologías escogidas	12
2.4 Estructura del proyecto	13
2.5 Descripción de los componentes	15
2.5.1 Componente 1	15
2.5.2 Componente 2	16
2.5.3 Componente 3	16
2.5.4 Componente 4	16
2.5.5 Componente 5	17
2.5.6 Componente 6	17
2.5.7 Componente 7	17
2.6 Definición de las tareas [proyecto de investigación]	18
2.6.1 Prueba 1	18
2.6.2 Prueba 2	18
2.6.3 Prueba 3	18
2.6.4 Prueba 4	18
2.7 Definición de las funcionalidades [proyecto de desenvolvimiento]	18
2.7.1 Funcionalidad 1	18
2.7.2 Funcionalidad 2	19
2.7.3 Funcionalidad 3	19

3. Montaje	19
3.1 Desmontar	19
3.2 Diseño	20
3.2 Fuente de Alimentación	22
3.4 Montaje	26
4. Código de programación	31
4.1 Primera parte del código	31
4.2. Segunda parte del código	32
5. Pàgina web	37
5.1 Secciones	37
5.2 Navbar	37
5.3 Footer	38
5.4 Home	39
5.5 Information	39
5.6 Materials	40
5.7 Gallery	41
6. Incidencias	44
7. Conclusiones	44
7.1 Conclusiones generales del proyecto	44
7.2 Consecución de los objetivos	44
7.3 Valoración de la metodología y planificación	45
7.4 Visión de futuro	45
8. Glosario	46
9. Bibliografia	47
10. Anexos	48

Lista de figuras

Figura 1.1: Imagen arduino NANO

Figura 1.2: Imagen arduino shield

Figura 1.3: Imagen servo motor

Figura 1.4: Imagen ultrasonidos

Figura 1.5: Imagen bateria

Figura 1.6: Imagen cables

Figura 1.7: Imagen ruedas

Figura 1.8: Imagen palos de madera

Figura 2.1: Logo TinKerCad

Figura 2.2: Logo Arduino

Figura 2.3: Logo Fritzing

Figura 2.4: Esquema de los cables

Figura 2.5: Esquema conexion arduino nano

Figura 2.6: Esquema 3D vista de lado

Figura 2.7: Esquema 3D vista de frente

Figura 3.1: Fotografía de dos robots OTTO

Figura 3.2: Fotografía de los distintos palos

Figura 3.3: Fotografía del papel

Figura 3.4: Fotografía de la conexión al arduino nano

Figura 3.5: Fotografía del cable para conectar arduino nano

Figura 3.6: Fotografía de la conexión de alimentación del arduino shield

Figura 3.7: Fotografía del cable de alimentación

Figura 3.8: Fotografía de las conexiones UIN y GND del arduino shield

Figura 3.9: Fotografía de los 3 tipos de baterías

Figura 3.10: Fotografía de las baterías de litio

Figura 3.11: Fotografía de las pilas botón

Figura 3.12: Fotografía del portapilas, y de las pilas

Figura 3.13: Fotografía de las conexiones al arduino shield.

Figura 3.14: Fotografía extensa de las conexiones al arduino shield

Figura 3.15: Fotografía de la conexión de cables al arduino shield

Figura 3.16: Fotografía del conector de los servomotores

Figura 3.17: Fotografía de la conexión al arduino shield del cable del servo

Figura 3.18: Fotografía de cerca de la conexión del servo

Figura 3.19: Fotografía de un servo motor y un alargador

Figura 3.20: Fotografía del robot

Figura 3.21: Fotografía frontal del robot

Figura 3.22: Fotografía de la conexión del portapilas al arduino shield

Figura 3.23: Fotografía de una sección del robot con el alambre y una rueda

Figura 3.24: Fotografía de una sección del robot con las ruedas puestas

Figura 5.1:Navbar de la página web

Figura 5.2: Navbar oscuro

Figura 5.3: Navbar seleccionado "Home"

Figura 5.4: Navbar para móvil

Figura 5.5: Navbar oscuro de móvil

Figura 5.6: Navbar de móvil contraído

Figura 5.7: Footer de la página web

Figura 5.8: Image slider de la página web

Figura 5.9: Sección de información dentro de la web

Figura 5.10: Sección de materiales de la página web

Figura 5.11: Sección de materiales con el ratón encima de las imágenes

Figura 5.12: Sección de imágenes en la página web

Figura 5.13: Mostrando el funcionamiento del image slider de la seccion de imagenes

Figura 5.14: Sección del código en la página web

1. Introducción

1.1 Contexto

Nuestro proyecto consiste en crear y programar un robot inspirado en una serpiente terrestre programada con arduino para que pueda moverse de manera autónoma

Tendrá sensores de distancia, de esta manera podrá esquivar los obstáculos. El cuerpo será un exoesqueleto, y tanto los servos y los cables irán dentro. Con Servo Motores, conseguiremos que se mueva de una forma muy similar a una serpiente real, con su característico zic-zac. En la parte de abajo tendrá unas pequeñas ruedas para poder moverse sin problemas.

La motivación principal de nuestro proyecto es conseguir plasmar una idea compleja en base los conocimientos y materiales que tenemos, teniendo mucha libertad creativa y implementacion de mejoras.

Creemos que será un robot complicado pero muy satisfactorio de ver el resultado, poca gente ha hecho serpientes robot y puede ser una buena idea.

1.2 Justificación

Este proyecto lo empezamos porque somos unos grandes amantes de las serpientes y de los robots, esto se junta perfectamente en nuestro proyecto, nos parecía un gran reto y muy divertido.

1.3 Objetivos

1.3.1 Objetivos generales

El objetivo principal es montar una serpiente robótica que se mueva de forma autónoma, detecte obstáculos y los esquive por sí misma. con una estructura robusta para que no se rompa, y servos para simular el movimiento característico de una serpiente.

1.3.2 Objetivos específicos

Objetivo 1:

Construir un armazón resistente con los palos de madera y los servo, ha de ser resistente, con capacidad de movimiento y en el que entren la placa base, sensores, batería, etc... También hemos de colocar los sensores y las ruedas.

Objetivo 2:

Lograr programar el robot de tal manera que pueda moverse hacia delante, con el característico movimiento en zig-zag de una serpiente.

Objetivo 3:

Programar por una parte los sensores para que detecten a que distancia esta de los objetos, también programar la serpiente para que pueda girar.

Objetivo final:

Por último tendremos que juntarlo todo, hacer que si los ultrasonidos detectan un objeto a más de 10 cm la serpiente vaya hacia delante, por el contrario si hay un objeto delante a menos de 10 cm, deberá girar hacia un lado para evitar el obstáculo.

1.4 Estrategia y planificación del proyecto

La estrategia es desarrollar un producto nuevo a partir de un producto con el que aprendimos mucho y hemos intentado implementar todos los elementos que constituían el anterior producto. Estamos muy de acuerdo que es la estrategia más apropiada gracias a que el producto antiguo era muy completo.

Planificamos la faena de la siguiente manera:

1. Materiales:

- Desmontar los robots antiguos para conseguir el mayor número de piezas útiles.
- Comprar las piezas que nos falten.

2. Montage:

- Montar todos los componentes para crear la estructura del robot.

3. Código:

- Buscar información sobre código arduino que pueda interesarnos.
- Crear nuestro código con lo que ya sabíamos más la información recopilada en internet.

4. Documentacion:

- Realizar la documentación del proyecto.

5. Página web:

- Creación de la página web.




1.5 Metodología de trabajo

La metodología de trabajo es sencilla, lo que hacemos es ir trabajando cada día un poco para que el proyecto no pare y siga adelante. Pero del mismo modo no trabajamos todo el día sin parar, eso haría que no descansemos del proyecto y no lo hagamos con tantas ganas.

1.6 Estudio económico y presupuestario.

Nuestro inventario de componentes no es muy amplio, pero está todo lo necesario para completar el robot. tenemos componentes reciclados, es decir, comprados hace años que no han salido gratis, y otros componentes que hemos tenido que comprar.

Finalmente a salido bastante barato, ya que como he dicho antes, hay partes recicladas que han salido gratis, por otra parte la que hemos tenido que comprar son muy baratas porque han sido compradas por aliexpress.

 <p>Figura 1.1: Imagen arduino NANO</p>	<p>Arduino NANO: Este es el Arduino NANO en el cual compilamos el programa, irá conectado a una placa base.</p>	<p>Componente reciclado. 0€</p>
 <p>Figura 1.2: Imagen arduino shield</p>	<p>Placa base: En esta placa base conectamos Arduino NANO, también es donde están todos los pines de conexión para los demás componentes.</p>	<p>Componente reciclado. 0€</p>
	<p>Servo motor: Este es el componente que hará que la serpiente se mueva.</p>	<p>4 reciclados. 0€ 4 comprados.</p>

<p>Figura 1.3: Imagen servo motor</p>		4.99€
 <p>Figura 1.4: Imagen ultrasonidos</p>	<p>Sensores ultrasónicos: Se ponen en la parte más adelantada del robot para que detecte obstáculos, y posteriormente puedan ser esquivados</p>	<p>Componente reciclado. 0€</p>
 <p>Figura 1.5: Imagen batería</p>	<p>Baterías: Son para que el robot pueda ir sin ayuda de un cable de alimentación .</p>	4.94€
 <p>Figura 1.6: Imagen cables</p>	<p>Alargadores: Los utilizamos para alargar los cables de los servos, ya que eran muy cortos y no llegaban a la placa base.</p>	0.99€
 <p>Figura 1.7: Imagen ruedas</p>	<p>Ruedas: Estas pequeñas ruedas servirán para que el robot se apoye en el suelo, y en el movimiento de ir hacia delante le sea más fácil, ya que giraran</p>	0.50€
 <p>Figura 1.8: Imagen palos de madera</p>	<p>Palos de madera: Estos palos unieron de una forma firme los servomotor, a primera vista no parecen gran cosa, pero bien unidos son muy resistentes.</p>	0.99€

Finalmente el proyecto a costado un total aproximado de 12,41€, nuestro presupuesto inicial era de 10€ así que tuvimos que ampliar el presupuesto a 15€ para poder comprar todas las piezas necesarias, y tener mas dinero por si habia algun imprevisto, finalmente sobraron 2.59€.

2 Descripción del proyecto

2.1 Anàlisis de requisitos [proyecto de desenvolvimiento]

2.1.1 Requisitos funcionales

Movimiento:

- Moverse hacia delante: Será la manera principal de movimiento de la serpiente.
- Girar hacia la derecha: Este movimiento servirá para esquivar obstáculos.

Componentes:

- Arduino NANO: Esta será nuestra placa base en la que compilamos el programa.
- Ultrasonidos: Este sera el sensor que utilizaremos para detectar obstáculos.

2.1.2 Requisitos no funcionales

Materiales baratos: Queremos que nuestros materiales sean baratos, para que cualquier persona pueda permitírselo.

2.2 Previsión de tareas de investigación [proyecto de investigación]

2.3 Tecnologías

2.3.1 Comparativa de las tecnologías valoradas

Hemos tenido en cuenta muchos tipos de tecnologías, necesitábamos aplicaciones para hacer el programa, otras para hacer el esquema, incluso estuvimos mirando algún tipo de tecnología con la que pudiésemos hacer piezas para el montaje.

Tecnologías para el programa: Aquí no tuvimos mucha complicación, la aplicación que necesitábamos era la de arduino, va perfectamente además es la oficial de los componentes que utilizabamos. Pero estuvimos viendo que se puede hacer el programa con aplicaciones como Atom¹.

¹ [Atom](#)

Tecnologías para el esquema: En este punto nos costó bastante encontrar una buena aplicación con la que hacer el esquema que fuese barata. Primero probamos con “easyEDA”, pero no nos convenció, era bastante complicada de entender, desde nuestro punto de vista no estaba bien organizada. Después intentamos utilizar “TinKerCad”, iba perfectamente pero era 3D, y esto no nos convenció, ya que no se podrán conectar cables para que se viese perfecto el esquema, aunque si la usamos para hacer un modelo 3D del robot. Por último probamos “Fritzing”, esta funcionaba perfectamente, podías conectar cables y estaban todos los componentes que nosotros íbamos a utilizar, así que nos quedamos con esta.

Tecnologías para las piezas: Esta parte fue bastante complicada, teníamos dos opciones; impresión 3D y piezas meccano². Después de pensarlo mucho decidimos que utilizaremos piezas meccano, pero había un grave problema ya que se tenían que pedir por aliexpress, y por culpa de la crisis del coronavirus los paquetes no llegaban a tiempo.

2.3.2 Tecnologías escogidas

Después de mucho pensar y comparar tecnologías, nos quedamos con las siguientes:

Modelo 3D	TinKerCad  AUTODESK® TINKERCAD® Figura 2.1: Logo TinKerCad	Tinkercad es un programa gratuito de modelado 3D en línea que se ejecuta en un navegador web, conocido por su simplicidad y facilidad de uso.
Programación	Arduino IDE  ARDUINO Figura 2.2: Logo Arduino	Arduino IDE es una aplicación multiplataforma que está escrita en Java. Se utiliza para escribir y cargar programas en placas Arduino.
Esquema	Fritzing	Fritzing es un programa libre, de automatización de diseño electrónico que busca ayudar a

² [Meccano](#)

	 Figura 2.3: Logo Fritzing	diseñadores y artistas para que puedan pasar de prototipos a productos finales.
--	--	---

2.4 Estructura del proyecto

La estructura del proyecto es sencilla, primero va la cabeza, es una estructura donde van la placa base y los sensores de ultrasonidos en la parte delantera, esta estructura está sujeta a una la conexión de madera del primer servo. En total hay 8 servos que están unidos entre ellos con una estructura de madera echa con palos.

A continuación se muestran dos esquemas de la construcción del robot:

- Esquema de componentes:

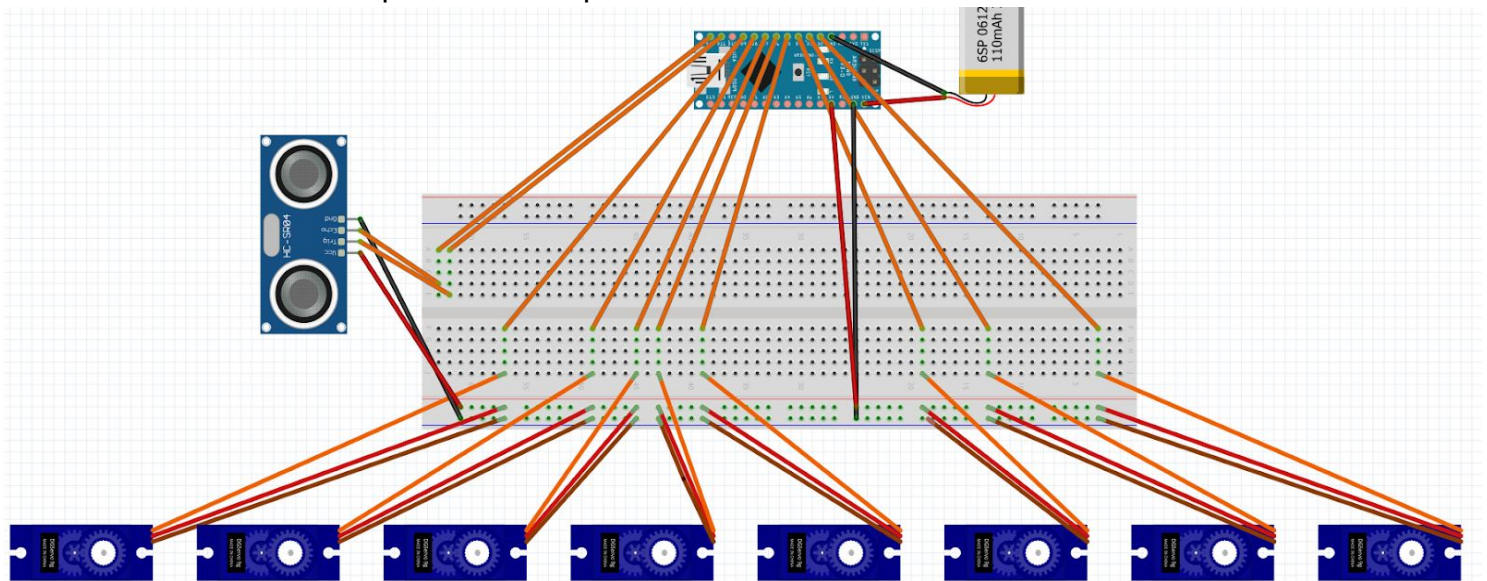


Figura 2.4: Esquema de los cables

// En este esquema ha sido cambiada la placa base por una protoboard para que sea más sencillo de ver y entender, en el producto final se conectan los cables de la misma forma pero a la placa base.

En este esquema, el cableado funciona de la siguiente manera:

- Naranjas: Cables de información, se conectan a la placa del D1 hasta el D12 y es por el que va la información al componente.
- Rojos: Cable positivo, los componentes se conectan al conector de 5V de la placa, y la batería se conecta al VIN.
- Negros: Cable negativo, se conecta a GND, la placa tiene 2 conectores, 1 para la batería y otro para los componentes.

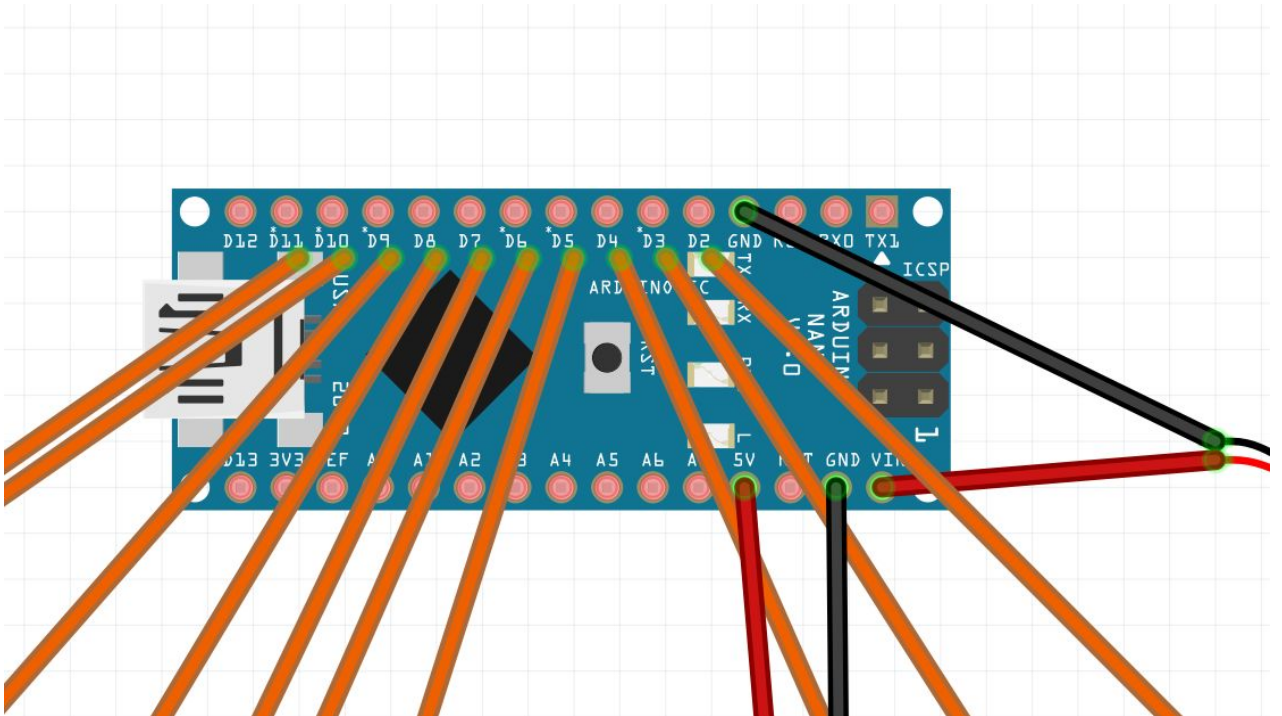


Figura 2.5: Esquema conexion arduino nano

En este zoom se puede ver mas detenidamente donde estan conectados los cables comentados anteriormente.
- Esquema completo:

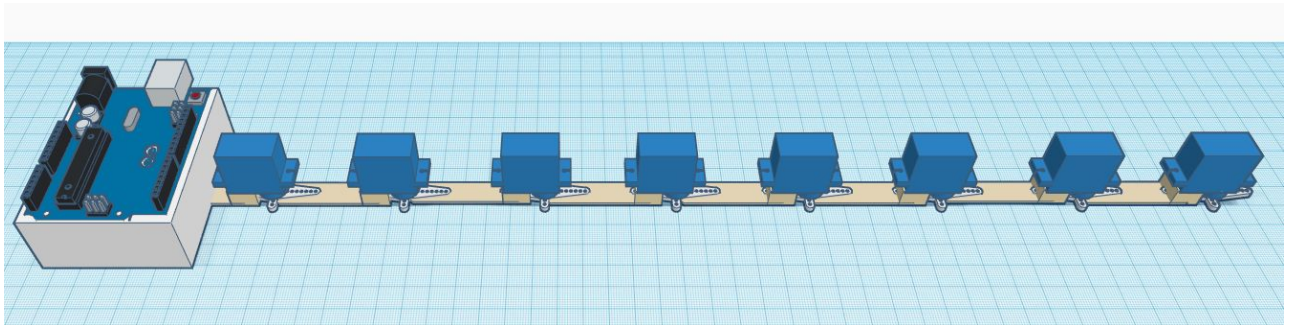


Figura 2.6: Esquema 3D vista de lado

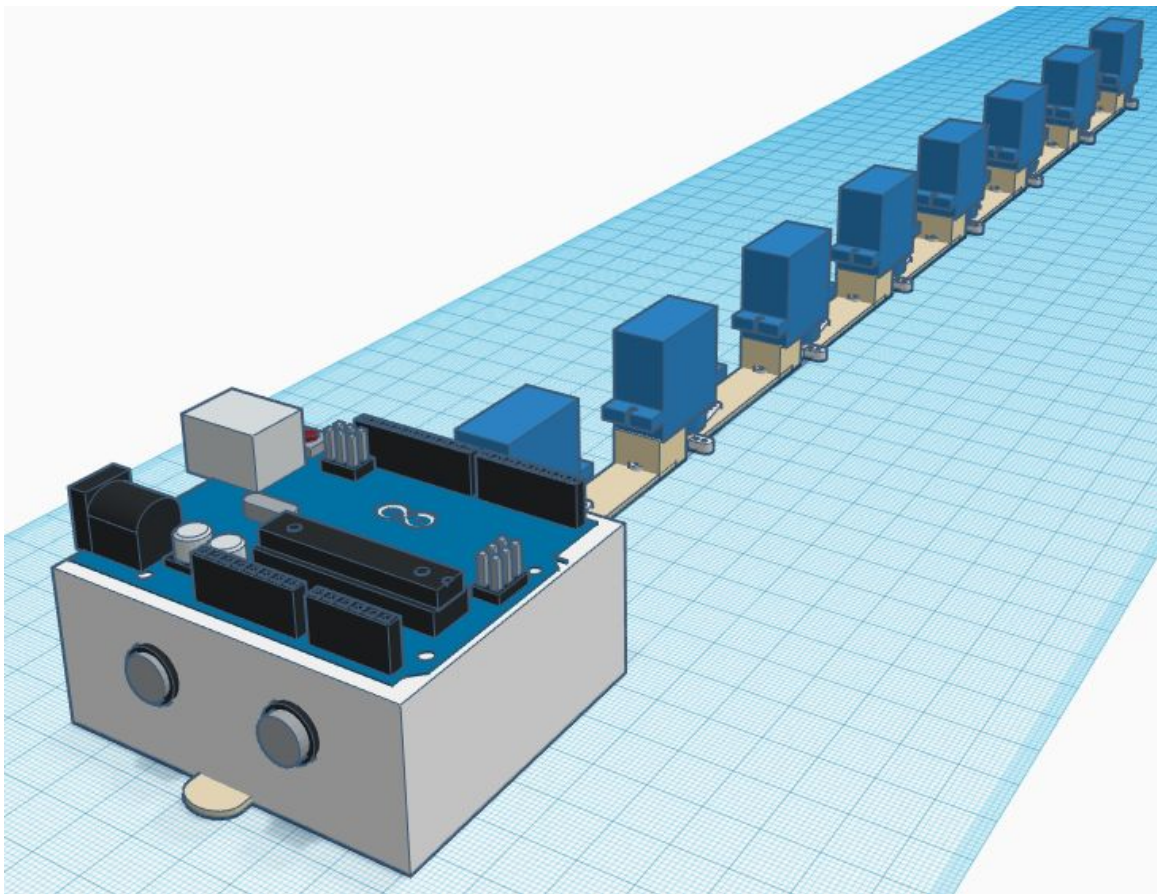


Figura 2.7: Esquema 3D vista de frente

En este diseño 3D podemos ver el resultado de la serpiente, podemos ver los servos unidos con los palos de madera, en la parte delantera hay un cubículo hecho con impresora 3D en el cual está la placa base, los sensores ultrasónicos, y aunque no se ve, también esta la batería.

2.5 Descripción de los componentes

Los componentes explicados a continuación, son los componentes informaticos y electronicos

2.5.1 Componente 1

Micro servo SG90	Voltaje	Precio
Servo motor capaz de rotar entre 0 y 180 grados muy completo con distintos cabezales y tornillos	4,2 - 6 V	1.25€ (por servo)

2.5.2 Componente 2

Modulo ultrasonico hc-SR04	Voltaje	Precio
Sensor de distancia ultrasónico de alta precisión con alcance de detección de 2cm a 450cm	5 V	3,04 €

2.5.3 Componente 3

Arduino Nano V3.0	Voltaje	Precio
Microcontrolador específico para compilar programa Arduino, su fuente de alimentación es con un cable USB Mini-B	5 V	1.03€

2.5.4 Componente 4

Batería Litio Fytoo 5PCS 3.7V 1200mAh	Voltaje	Precio
La batería de iones de litio, son un dispositivo diseñado para almacenamiento de energía eléctrica.	3.7 V	4.94€

2.5.5 Componente 5

Arduino nano shield	Voltaje	Precio
Tarjeta de expansión compatible con la Arduino Nano, con pines I/O análogos y digitales, interfaz I2C y conector para alimentación.	Funciona con el voltaje de la Arduino Nano (5V)	2.75€

2.5.6 Componente 6

Palos de madera	Voltaje	Precio
Son el armazón principal del robot, se encarga de unir los servos.	-	0.99€

2.5.7 Componente 7

Ruedas	Voltaje	Precio
Servirán para que el robot se apoye en el suelo, y en el movimiento de ir hacia delante le sea más fácil, ya que giraran	-	0.50€

2.6 Definición de las tareas [proyecto de investigación]

2.6.1 Prueba 1

Calibración de servos.

Esta prueba consistirá en hacer un programa para encontrar el grado en el que se encuentra el centro del servo($\pm 90^\circ$) y así no tendremos que preocuparnos de que todos vayan en la misma posición.

2.6.2 Prueba 2

Velocidad de servos

En esta prueba consistirá en comprobar las distintas velocidades de los servos para que en el programa final sepamos aproximadamente a qué velocidad se tienen que mover.

2.6.3 Prueba 3

Movimiento serpenteante.

Para la tercera prueba nos dispondremos a avanzar y a acercarnos más al proyecto con su característico movimiento, haremos que cuatro servos sigan el patrón del primero pero con retraso es decir delay algo parecido al juego de la serpiente.

2.6.4 Prueba 4

Desplazamiento.

Ya hemos llegado a la prueba más importante, la manera que determinará cómo conseguirá desplazarse sobre unas ruedas nos llevará tiempo encontrar la manera de fusionar el movimiento serpenteante con el desplazamiento.

2.7 Definición de las funcionalidades [proyecto de desenvolvimiento]

2.7.1 Funcionalidad 1

La primera función de la serpiente, y la más básica, es moverse hacia delante, los servomotores oscilan entre 125 y 55, con un retraso de 30

grados. De esta manera conseguimos que vaya haciendo zigzag y junto a la fricción que ejerce, se mueve hacia delante.

2.7.2 Funcionalidad 2

La segunda función de la serpiente, es detectar la distancia. Utiliza los ultrasonido para calcular a cuantos centímetros se encuentra un objeto que hay delante.

2.7.3 Funcionalidad 3

La tercera función de la serpiente, se divide en dos partes, y está relacionada con la segunda funcionalidad. Lo que hace es, poner todos los servos de 0 a 45 grados para que se enrosque, y después de 3 segundos se ponen todos los servos a 90 grados para desenroscarse, gracias a la fricción, conseguimos que se gire. Este movimiento lo efectúa cuando hay un obstáculo a menos de 10 cm, de esta manera al girar no se choca.

3. Montaje

3.1 Desmontar

No podemos empezar el montaje de nuestro proyecto no sin antes reciclar todos los componentes y piezas 3D de nuestros proyectos pasados muy conocidos como OTTO³. No solo aprovecharemos la parte física, también haremos servir muchos de los códigos ejemplo, esquemas virtuales para conectar componentes etc... . Aquí adjuntamos algunas fotos de ejemplo de nuestros proyectos pasados junto a algunos esquemas que nos han sido de gran utilidad.

Separamos con mucho cuidado la cabeza del cuerpo presionando en los laterales del cráneo para abrirlo, desconectamos de los pines todos ya cada uno de los cables, los 4 servos, los 4 cables del ultrasonido, los dos cables del buzzer⁴ y la batería al estar soldada con estaño no nos quedará más remedio que arrancarla afectando lo mínimo posible a la placa. Con un destornillador pequeño separaremos las piernas del torso desatornillando las de sus sujeciones. Por último la parte más complicada separar los pies de la pierna, estos están sujetos a presión con el servo maximizando la probabilidad de partirse.

³ [OTTO](#)

⁴ [Buzzer](#)

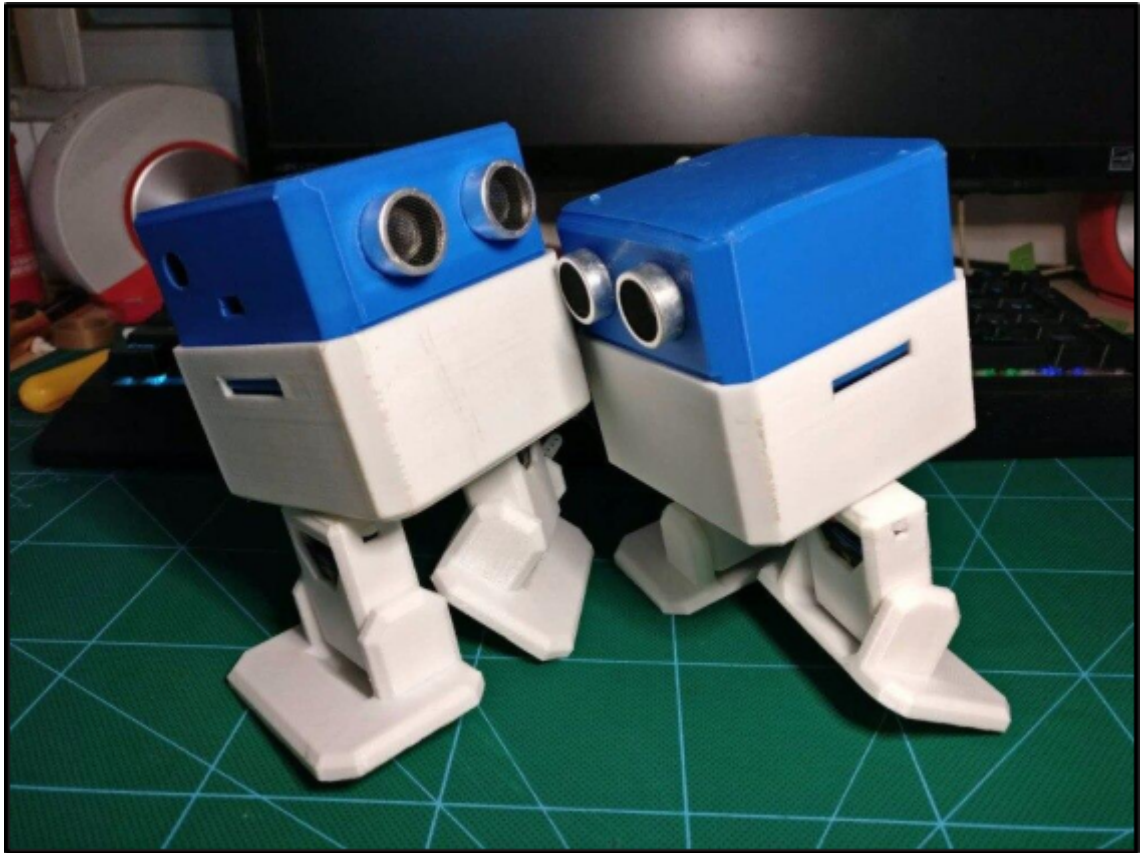


Figura 3.1: Fotografía de dos robots OTTO

3.2 Diseño

Antes de empezar el montaje no solo necesitaremos las piezas recicladas y las piezas compradas, necesitaremos diseñar nuestras propias piezas que en este caso constituirán la estructura donde irán enlazados los servos unos con otros. Adjuntamos algunas fotos del proceso de diseño de la estructura del robot.

Para el diseño de la estructura decidimos utilizar palitos de madera bastante finos y largos, los cortamos una tercera parte del palito de cada extremo dejando el tercio de en medio fuera de lugar, pegaremos con silicona caliente mediante una pistola de silicona el accesorio blanco que disponemos de muchos ya que cada servo viene equipado con 4 recambios de accesorios totalmente diferentes, al tercio del palito exactamente por la parte redonda aproximadamente. Hay un palito

distinto a los demás que es el primero que debe aguantar la placa base no un servo, cortamos un palito por la mitad aproximadamente.

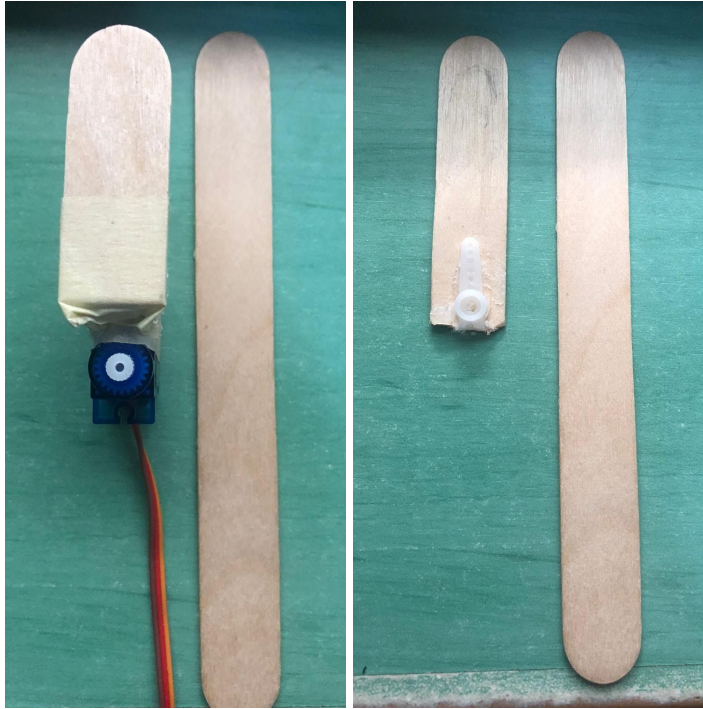


Figura 3.2: Fotografía de los distintos palos

Por último, finalizamos la construcción diseñando unas piezas de papel que alzarán al servo para que pueda conectarse al siguiente segmento. Doblamos un din A4 (reciclado) por la mitad, doblamos por la mitad las dos mitades y así sucesivamente hasta que queden marcadas tiras finas de 1 cm aproximadamente. Cortamos la hoja de papel por las guías dejando un gran montón de tiras de papel, doblamos estas por la mitad 4 o 5 veces sin pasarnos. Con distintos pegamentos probamos a pegar 3 o 4 tiras ya dobladas una encima de la otra formando así una pieza compacta, finalmente decidimos utilizar silicona caliente. Recubrimos los servos de cinta de carroceros para no dañarlos a la hora de pegar las piezas. Primero pegamos el montoncito compacto formado por 3 o 4 tiras dobladas en el palito de madera justo al final, después pegamos el palito con la parte superior trasera del servo fusionando así la pieza de papel.

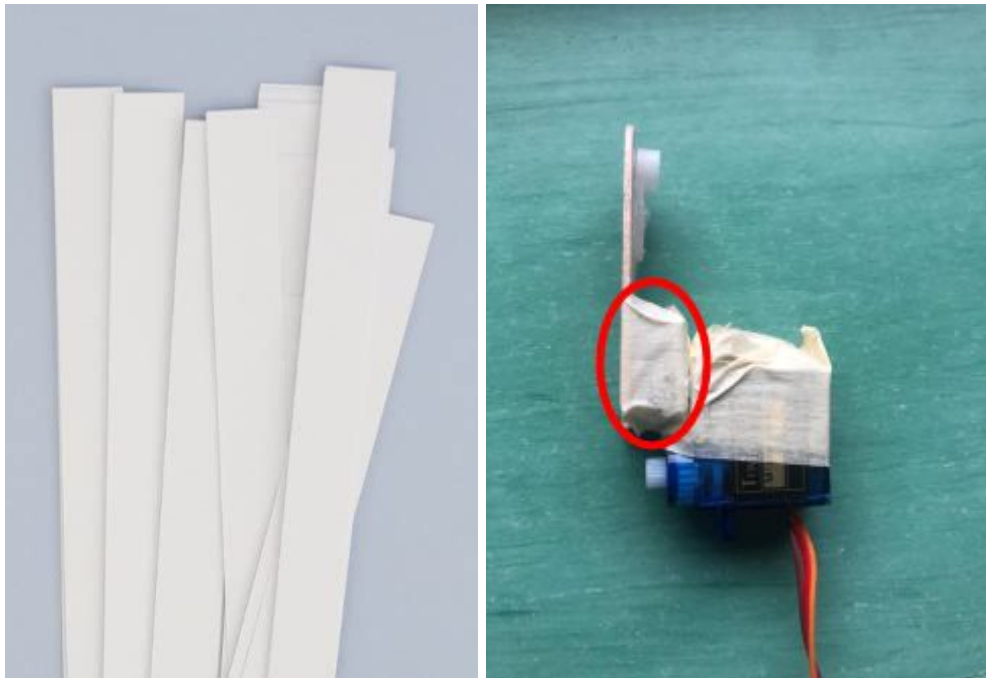


Figura 3.3: Fotografía del papel

3.2 Fuente de Alimentación

El robot requerirá energía para funcionar, el nuestro está diseñado para obtenerla de 3 lugares diferentes. Necesita bastante energía ya que se compone de 8 servos y un sensor, así que se la proporcionaremos.

La manera mas facil y mas usual es con el cable Mini-B Usb jack es decir el cable con el que subimos el programa/script a la placa base también vale sin ningún problema el del mando de PS3 sony ya que es el mismo.

La parte Usb puede ir en un ordenador, portátil, cargador que acepte entrada Usb etc...

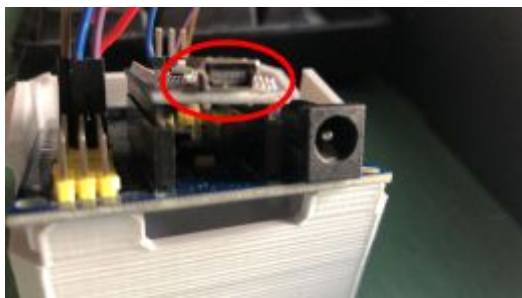


Figura 3.4: Fotografía de la conexión al arduino nano



Figura 3.5: Fotografía del cable para conectar arduino nano

Otra manera también muy común es con el cable DC Jack 5,5mm así se les da energía a la gran mayoría de placas de arduino, nosotros hemos utilizado un cargador de Walkie talkie.

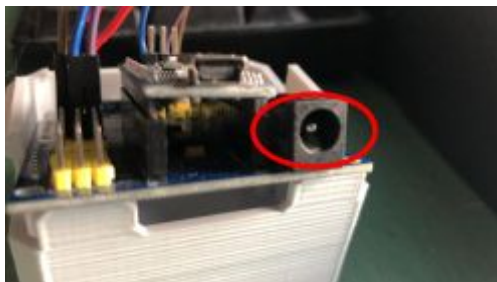


Figura 3.6: Fotografía de la conexión de alimentación del arduino shield



Figura 3.7: Fotografía del cable de alimentación

Una manera compleja y complicada seria soldar una batería directamente a la placa, es algo muy peligroso ya que podríamos dañar la placa base. Tendríamos que utilizar un soldador

Nosotros nos decantamos por esta manera ya que así el robot podría llevar encima su propia energía.

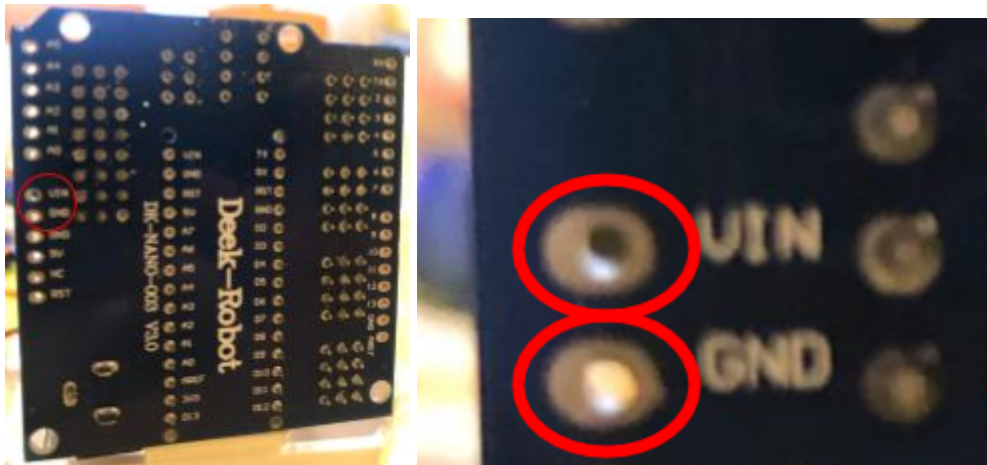


Figura 3.8: Fotografía de las conexiones UIN y GND del arduino shield

Probamos un total de 3 tipos de baterías diferentes para ver cuál se adapta mejor.



Figura 3.9: Fotografía de los 3 tipos de baterías

Las baterías de litio lipo eran nuestra principal idea para la alimentación pero no teníamos el cable hembra para soldar en la placa base.



Figura 3.10: Fotografía de las baterías de litio

Intentamos sustituirlo por estas pilas de botón, soldamos el portapilas a la placa, colocamos 3 pilas y parecía que no daba suficiente energía.

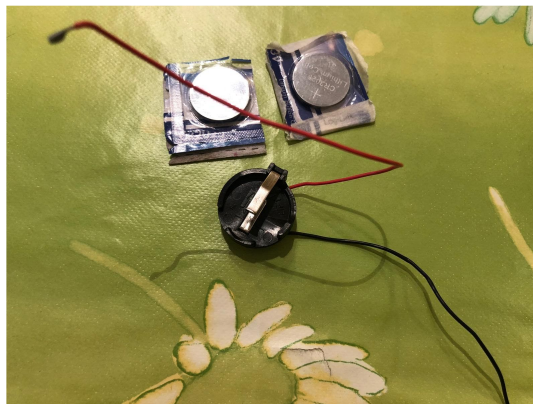


Figura 3.11: Fotografía de las pilas botón

Por último reciclamos el antiguo portapilas a pesar de que no buscábamos tanto peso soldamos el portapilas con un poco de estaño y con mucho cuidado a la placa.



Figura 3.12: Fotografía del portapilas, y de las pilas

3.4 Montaje

Finalmente después de todo, empezamos la parte del montaje o sea unir y conectar todo en su sitio.

Para conectar el sensor usamos estos dos esquemas, tiene 4 pines cada uno diferente, es de gran importancia conectarlos donde se debe de lo contrario no funcionara correctamente y incluso podría dañarse.

El nombre de los pines son Vcc, Trig, Echo y Gnd.

La manera de conectarse varía según el programa por eso tuvimos cautela al guiarnos del esquema.

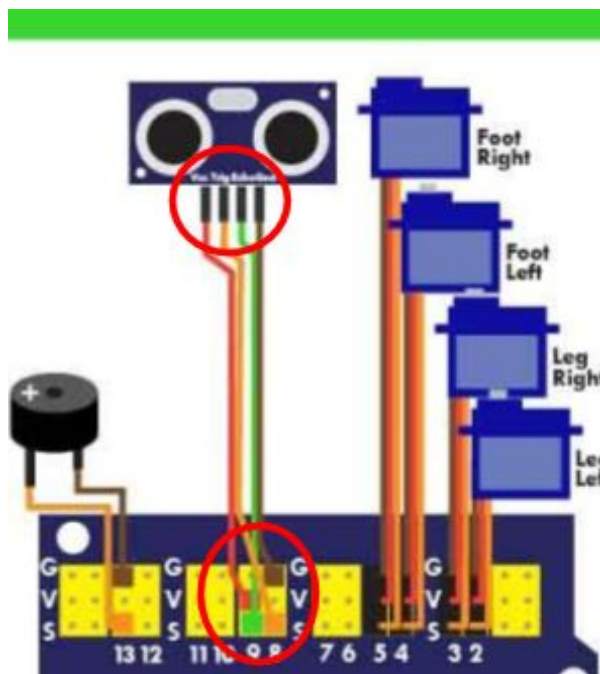


Figura 3.13: Fotografía de las conexiones al arduino shield.

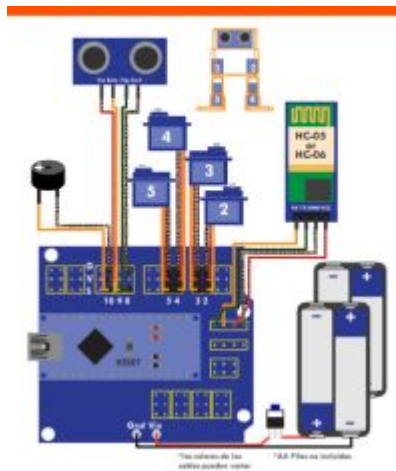


Figura 3.14: Fotografía extensa de las conexiones al arduino shield

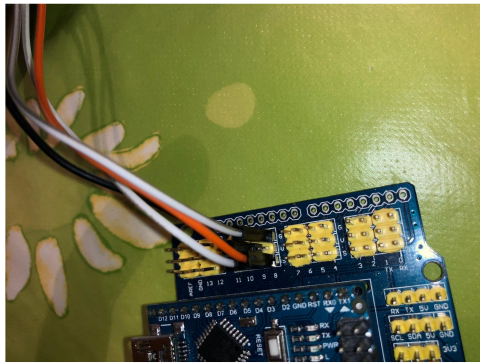


Figura 3.15: Fotografía de la conexión de cables al arduino shield

Los servos sin embargo son más sencillos, tienen 3 pines pero la dificultad solo reside en conectarlos al revés, la manera correcta de conectarlos es que la parte plateada esté mirando hacia las entradas de energía, otra manera de entenderlo sería viendo los colores del cable, el cable más oscuro más lejos del núcleo de la placa y el cable mas clarito lo contrario.

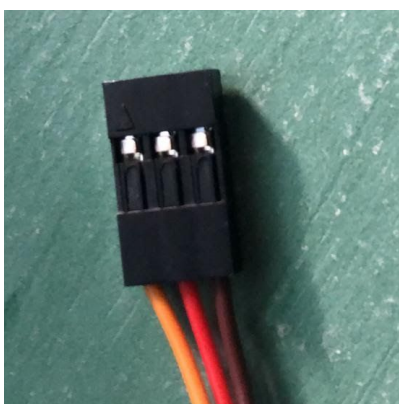


Figura 3.16: Fotografía del conector de los servomotores

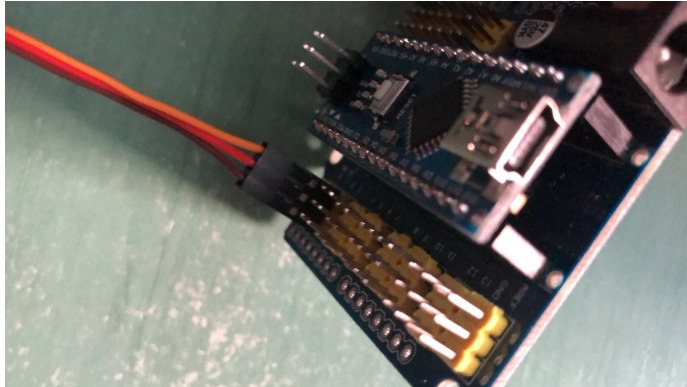


Figura 3.17: Fotografía de la conexión al arduino shield del cable del servo

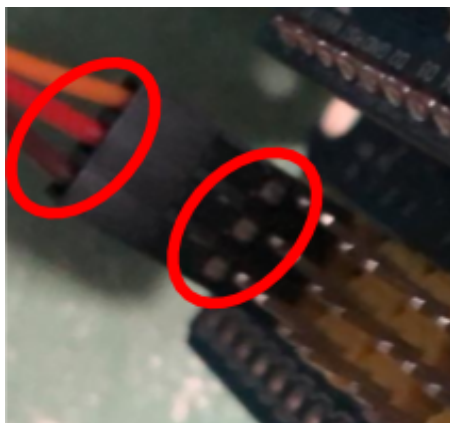


Figura 3.18: Fotografía de cerca de la conexión del servo

Los alargadores se conectan siguiendo el ejemplo de los colores del cable, siguiendo el patrón de los colores. Al principio pensábamos en solo poner alargadores en los servos más lejanos, pero corregimos y decidimos ponerlo a todos los que no lleguen a la placa y no tengan maleabilidad es decir que no estén tensos. Depende el programario, es importante que en el programario no nos obligue a conectarlos en el pin 0, 1, 2, ya que creemos que nos son para lo que buscamos que en este caso son los servos.



Figura 3.19: Fotografía de un servo motor y un alargador

Uniremos cada servo con el siguiente diferenciando solamente dos, la cola y la cabeza encajando la cola primero y al final la cabeza o viceversa.

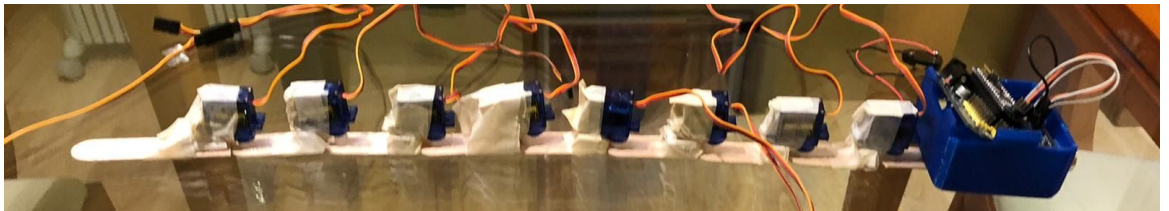


Figura 3.20: Fotografía del robot

En nuestra pieza reciclada colocaremos los sensores, de manera que queden los pinchos de pines hacia arriba. Después colocaremos la placa base de manera que queden los agujeros de alimentación al lado de los de la pieza reciclada. Dejando fuera el portapilas listo para colocar las pilas.

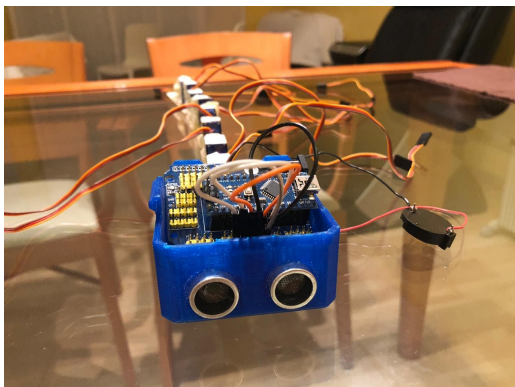


Figura 3.21: Fotografía frontal del robot



Figura 3.22: Fotografía de la conexión del portapilas al arduino shield

Por último montaremos las ruedas, fue un proceso complicado ya que no teníamos mucho soporte para pasar el alambre con el que uniremos las ruedas.

Finalmente conseguimos pasarlo entre el servo y el palo de madera un alambre en el cual enganchamos las ruedas, una a cada.

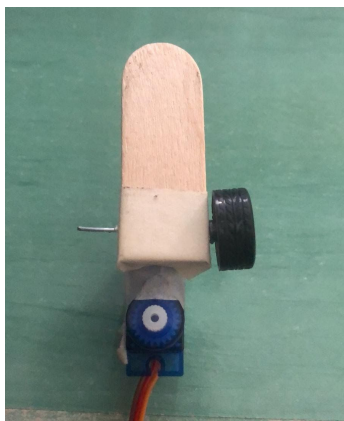


Figura 3.23: Fotografía de una sección del robot con el alambre y una rueda

Finalmente debemos comprobar que giran bien, si no es así las podemos ajustar sin problema ejerciendo un poco de fuerza hacia abajo para que se coloquen bien.

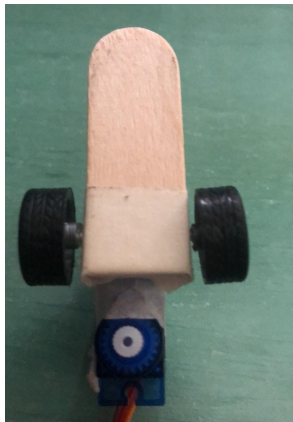


Figura 3.24: Fotografía de una sección del robot con las ruedas puestas

4. Código de programación

El código está diseñado en dos partes, una en la que se incluyen las librerías y se dice en que pines están los servos y los ultrasonido. También se pone una pequeña parte del código que explicaremos más adelante.

Por otra parte tenemos un if, en el cual ejecutamos los voids de que escribimos más abajo en el código.

4.1 Primera parte del código

En esta primera parte del código, declaramos todas las variables

```
/*
  Joel Rodrigalvarez y Alan Rodriguez
  Trabajo de sintesis 2n SMX
  Codigo libre
*/

// Incluimos la libreria de los servos
#include <Servo.h>

// Declaramos la variable servo
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;
Servo myservo6;
Servo myservo7;
Servo myservo8;

//Declaramos las variables array, una de tipo servo y una de tipo boolean
Servo myservos[]={myservo1,myservo2,myservo3,myservo4,myservo5,myservo6,myservo7,myservo8};
boolean activado[]={false,false,false,false,false,false,false,false};
//Declaramos un array de tipo integer de longitud 8, pero vacia
int dxs[8];
```



```
//Declaramos los pines en los que conectar el Ultrasonidos
const int Trigger = 8; //Pin digital 8 para el Trigger del sensor
const int Echo = 9; //Pin digital 9 para el Echo del sensor

//Declaramos la funcion setup (es el inicialización principal de arduino)
void setup() {
  // Llamamos a las funciones begin y println
  Serial.begin(19200); //Escogemos los baudios de nuestro monitor serie
  Serial.println("test");

  // Declaramos la variable array de tipo integer que nos servira para determinar en que pines iran los servos
  int pines[]={3,4,5,6,7,10,11,12};
  for(int i=0;i<8;i++){
    myservos[i].attach(pines[i]);
  }

  // Ultrasonidos
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT); //pin como entrada
  digitalWrite(Trigger, LOW); //Inicializamos el pin con 0

  // Declaramos la variable array dxrecta para utilizarlo en la funcion inicializadx
  int dxrecta[]={1,1,1,1,1,1,1,1};
  inicializadx(dxrecta);

  // Llamamos a la funcion recta
  recta();
}

//Declaramos la funcion inicializadx, que necesitara que le pasen un integer cuando se la llame
void inicializadx(int* losdx) {
  copyB(losdx,dxs,8);
}
```

4.2. Segunda parte del código

```
void loop() {
  //Hacemos un if, que comprueba si tiene un obstaculo
  if (distanciaMenorQue(10)) {

    //Llamamos a la funcion enroscar
    enroscar();
  }
  else{

    // Llamamos a la funcion avanzar
    avanzar();
  }
}
```

//Hacemos un if, que comprueba si tiene un obstáculo a menos de 10cm (con la función distancia Menor Que(10)), si lo tiene pues llamara la función enroscar, si no se llamara la función avanzar

En la siguiente captura creamos un booleano para saber si la distancia detectada por el sensor es mayor de 10cm.

```
// Declaramos una funcion que devuelve un boolean
boolean distanciaMenorQue(int distancia){
  //Ejecutamos el sensor
  long t; //tiempo que demora en llegar el eco
  long d; //distancia en centimetros

  digitalWrite(Triquer, HIGH);
  delayMicroseconds(10);          //Enviamos un pulso de 10us
  digitalWrite(Triquer, LOW);

  t = pulseIn(Echo, HIGH); //obtenemos el tiempo del pulso
  d = t/59;                 //escalamos el tiempo a una distancia en cm

  Serial.print("Distancia: ");
  Serial.print(d);          //Enviamos al monitor serie el valor de la distancia
  Serial.print("cm");
  Serial.println();
  //delay(100);              //Hacemos una pausa de 100ms

  if (d < distancia){
    return true;
  }else {
    return false;
  }
}
```

Creamos 2 funciones que servirán para ver los grados a los que están los servos en la consola.

```
// Declaramos la función imprimeServo (Solo es para el monitor serie)
void imprimeServo(Servo myservo, int n){
  String Tinicio = "IN";
  String dospuntos = ": ";
  String Sinicio = Tinicio + n + dospuntos + myservo.read();
  Serial.println(Sinicio);
}

// Declaramos la función imprimeServo (Solo es para el monitor serie)
void imprimeServos(){
  for(int i=0, y=1;i<8;i++){
    imprimeServo(myservos[i], y);
    y++;
  }
}
```

Creamos un array para que los servos se activen. Lo utilizaremos más adelante.

```
// Declaramos la función activar que sirve para activar los servos
void activar(int i) {
    for(int index=1,grados=0;1<=8;i++){
        grados=grados+30;
        if(i>=grados) activado[i] = true;
    }
}
```

Declaramos la funcion copyB.

```
// Declaramos la función copyB
void copyB(int* src, int* dst, int len) {
    for (int i = 0; i < len; i++) {
        *dst++ = *src++;
    }
}
```

Declaramos la función recta, la cual llamamos al principio del documento para que se ponga completamente a 90° el robot (completamente recto).

```
// Declaramos la función recta que sirve para que se ponga a 90 grados
void recta() {
    //Servos a 90
    for(int i=0;i<8;i++){
        myservos[i].write(90);
    }
}
```

Creamos la función avanzar, es la parte más importante ya que consigue que el robot avance, los servos oscilan entre 125° y 55° con una desviación entre ellos de 30°, para conseguir el movimiento de una serpiente.

```
// Declaramos la función avanzar
void avanzar() {

    int i = 0;
    int vuelta = 240;
    int total[8];

    //Activar primer servo
    activado[0] = true;

    while (i < vuelta) {
        //Llamamos a la función imprimeServos
        imprimeServos();
        for (int j=0; j<8; j++){
            total[j]=myservos[j].read();
            if (total[j] >= 125) dxs[j] = -1;
            if (total[j] <= 55) dxs[j] = 1;
        }

        //Sumar los grados a los servos
        for(int j=0;j<8;j++){
            if (activado[j]){
                total[j]=total[j]+dxs[j];
                myservos[j].write(total[j]);
            }
        }

        // Llamamos a la función activar
        activar(i);

        i++; // Incrementa en 1 la i

        // Si la función distanciaMenorQue devuelve true, entonces fuerza a salir del bucle
        if (distanciaMenorQue(10)){
            break;
        }
    }
}
```

Creamos la función enroscar, esta función servirá para pararse, enroscarse y, al desenroscarse girara para evitar un obstáculo.

```
// Declaramos la función enroscar
void enroscar(){
//enroscar el robot
  int gra2[]={15,20,30,30,30,55,65,35};
  for(int i=7;i>=0;i--){
    myservos[i].write(gra2[7-i]);
    delay(1000);
  }
  //desenroscar el robot
  for(int i=0;i<=7;i++){
    myservos[i].write(90);
    delay(1000);
  }
}
```

Y este es el código con el que la serpiente se mueve de la manera más óptima posible.

5. Pàgina web

La pàgina web està pensada per presentar el nostre projecte d'una manera més elegant i gustosa de veure. La pàgina reflecteix en gran part els coneixements adquirits en el disseny de pàgines web durant el curs. Hem intentat implementar el màxim d'elements possibles diferents. Que sigui possible veure la pàgina des d'un dispositiu mòbil o tablet. La gran majoria d'imatges han tingut que ser editades en Paint i Paint 3D per adaptar-les a la pàgina web. La pàgina web està en anglès per que sigui més universal.

Link para ver el código entero en gitlab

<https://gitlab.com/Alanegra/m12>

Link para ver la página web

<https://alanegra.gitlab.io/m12>

5.1 Secciones

Tiene incorporada 5 secciones totalmente diferentes aunque me gustaría comentar la navbar⁵ y el footer⁶.

- Navbar
- Footer
- Home
- Introduction
- Materials
- Gallery
- Code

5.2 Navbar

Tenemos una Navbar de los más minimalista con nuestro logo a la izquierda, un botón que a la derecha del logo que su función es la de un interruptor cambia la página web de luminosa es decir que predominen los colores blancos a oscura. Tiene las secciones preparadas para pulsar.

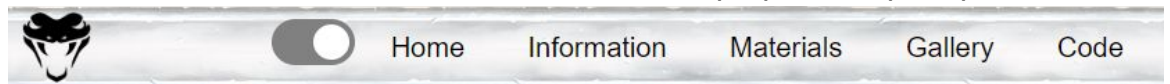
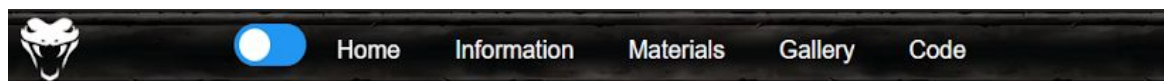


Figura 5.1:Navbar de la página web



⁵ [Navbar](#)

⁶ [Footer](#)

Figura 5.2: Navbar oscuro



Figura 5.3: Navbar seleccionado "Home"

La navbar se ajusta perfectamente a los dispositivos móviles, incluso tiene un botón que no dispone la navbar por defecto, es la de esconder las secciones, muy útil y cómoda para el usuario.

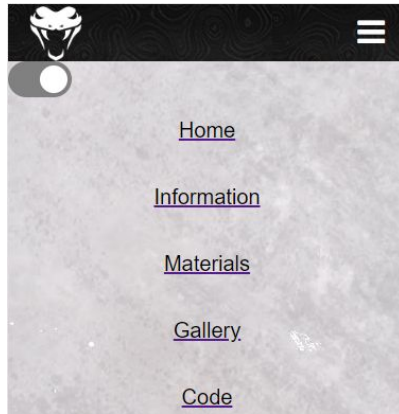


Figura 5.4: Navbar para móvil

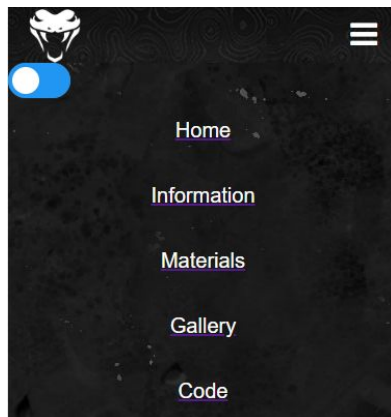


Figura 5.5: Navbar oscuro de móvil



Figura 5.6: Navbar de móvil contraído

5.3 Footer

Buscábamos un footer atrevido y sobretodo épico solo con acercar el cursor hacia abajo se desplegará hacia arriba el footer este incluye

nuestros mails y 3 iconos totalmente equipados con sus hyperlinks⁷ . Sin embargo en móviles habrá que hacer clic a la bolita magenta.



Figura 5.7: Footer de la página web

5.4 Home

En Home disponemos del elemento Image slider⁸ y un botón para descargar y bajarse de la web dos PDF y un archivo txt. Hay 3 imágenes ya que nos lo indica los números que aparecen en la parte inferior de la página, tenemos la posibilidad de pasar de página pulsando las flechas de los extremos o simplemente esperamos y al cabo de un tiempo se pasan automaticamente las imagenes.

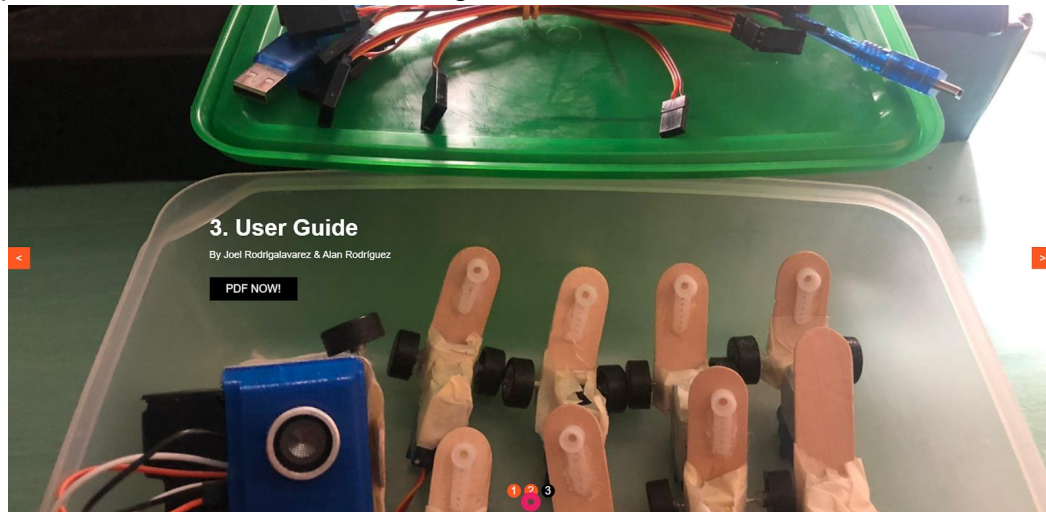


Figura 5.8: Image slider de la página web

5.5 Information

En Information hemos escrito nuestro resumen del proyecto en versión inglesa con un gif y una imagen para acompañar el texto. En la parte inferior encontramos otra vez el mensaje de PDF NOW! también está en dos de las imágenes del imageslider de Home. Al hacer clic nos bajaremos el PDF de esta misma memoria.

⁷ [Hyperlink](#)

⁸ [Image slider](#)

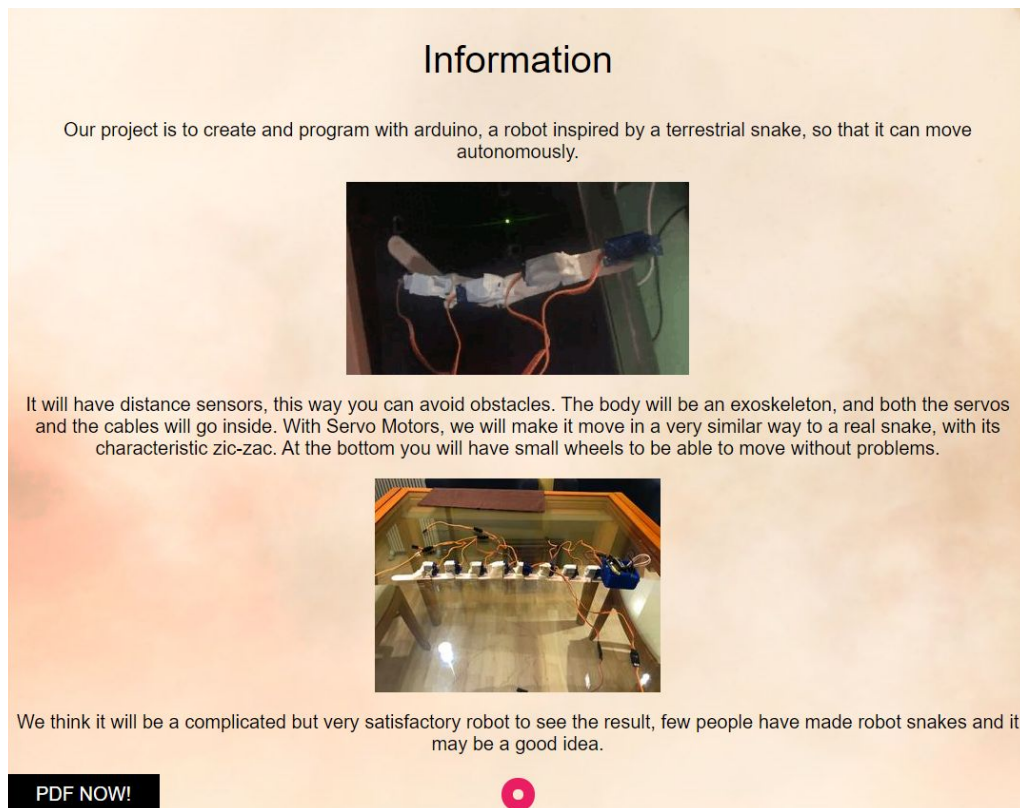


Figura 5.9: Sección de información dentro de la web

5.6 Materials

En Materials tenemos 4 cajitas responsives⁹ donde se encuentran los componentes básicos de nuestro robot si pasamos el ratón por encima o pulsamos con un dispositivo móvil se mostrará la fotografía desde otra perspectiva. En la segunda foto podemos ver claramente en funcionamiento los elementos comentados, el sensor tiene otra perspectiva.



Figura 5.10: Sección de materiales de la página web

⁹ [Responsives](#)

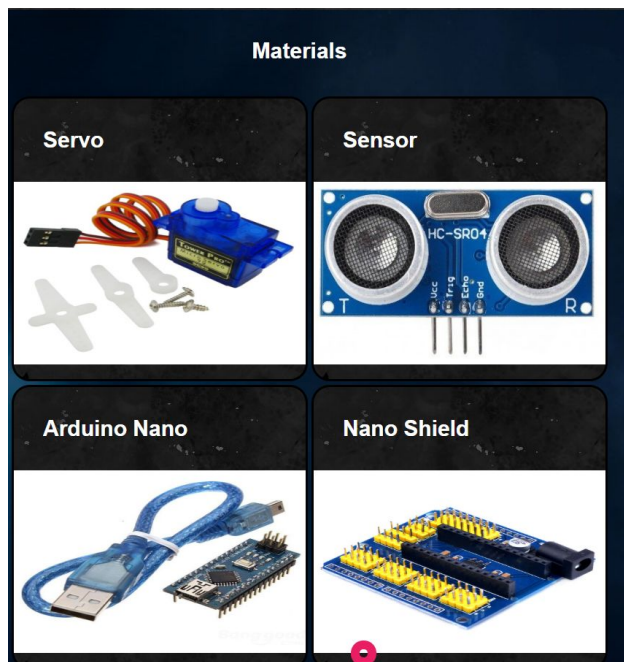


Figura 5.11: Sección de materiales con el ratón encima de las imágenes

5.7 Gallery

En Gallery disponemos de un gran catálogo de fotografías, exactamente hay 22. Son responsives como las anteriores pero estas si pulsamos con clic se nos abrirá un imageslider diferente al de la página de Home. Volverá a tener flechas a cada lado de la imagen pero a diferencia del otro podemos elegir cual queremos ver en zoom bajando un poco la pantalla sin salir del imageslider, para salir tendremos que hacer clic a la cruz superior a la derecha.

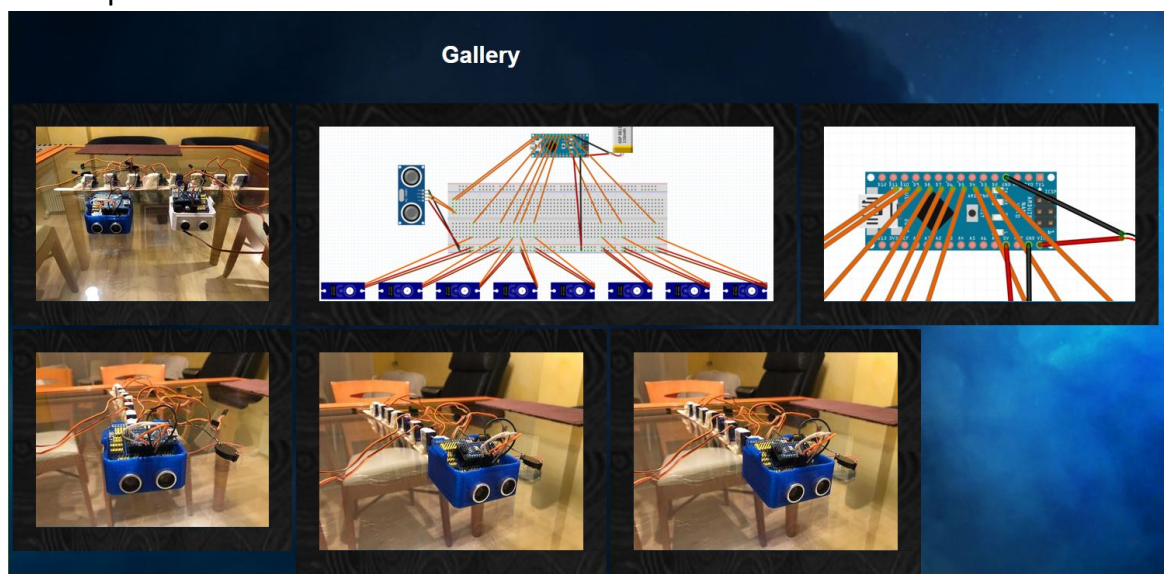


Figura 5.12: Sección de imágenes en la página web

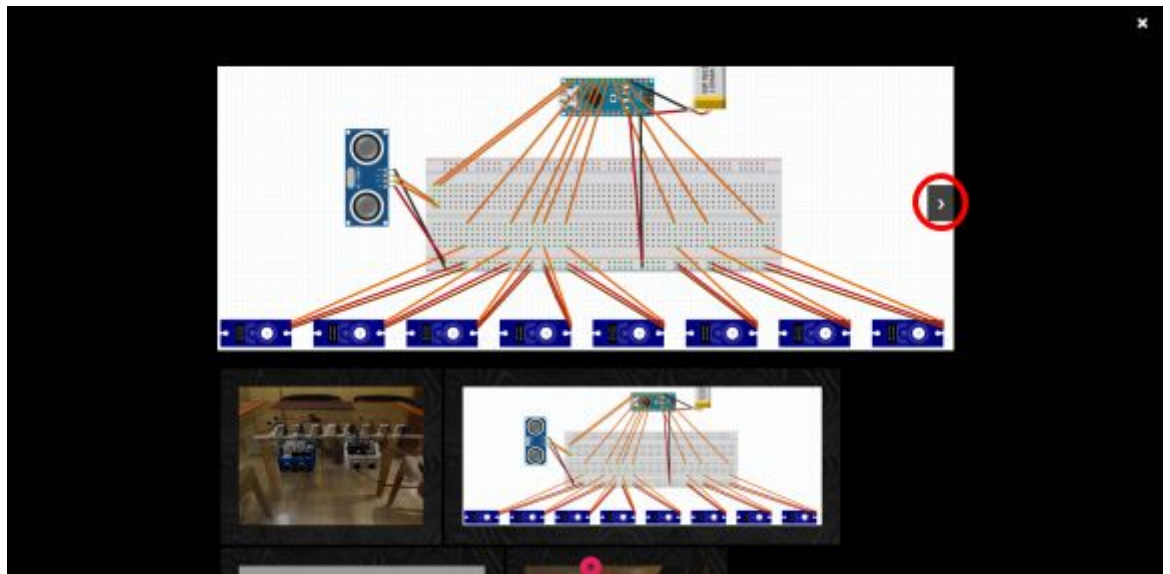


Figura 5.13: Mostrando el funcionamiento del image slider de la seccion de imagenes

5.8 Code

En Code simplemente pondremos el código completo incluso el mismo botón de la página de home que proporciona un archivo txt con el código.

```
int i = 0;
int vuelta = 140;

int total[8];
for(int i=0; i<8;i++){
    total[i]=myservos[i].read();
}

while (i < vuelta) {
    //Escribir servo
    for(int i=0;i<8;i++){
        myservos[i].write(total[i]); // El segundo valor que le pasamos a la funcion es la velocidad que entre 1-255
    }

    //Preparar Strings para la consola
    String iter = "ITERACION";
    String Siter = iter + i;

    imprimeServos();

    for (int i=0; i<8; i++){
        if (total[i] >= 125) dxs[i] = -1;
        if (total[i] <= 55) dxs[i] = 1;
    }
    //Sumar los grados
    for(int i=0;i<8;i++){
        total[i]=total[i]+dxs[i];
    }

    //Incrementar la iteracion

    i++;

    if (distanciaMenorQue(10)){
        break;
    }
}
}
```

FREE CODE!

Figura 5.14: Sección del código en la página web

6. Incidencias

Hemos tenido bastantes problemas, sobre todo por culpa del confinamiento del covid-19. Nuestro plan inicial se vio afectado, ya que no nos llegaron las piezas que estaban en camino. Partes muy importantes como el armazón, el cual en un principio iba a ser de piezas meccano pero que como no llegaron, dejamos las piezas de madera.

Por otra parte también faltaron los cables necesarios para utilizar las baterías de litio, que eran nuestra principal idea, intentamos comprar los en las tiendas locales y en amazon pero no tuvimos ningún éxito.

Podríamos decir que esta situación nos a hecho cambiar totalmente la trayectoria del proyecto.

7. Conclusiones

7.1 Conclusiones generales del proyecto

La realización del proyecto nos ha servido para formarnos en diferentes tecnologías y asimilar mucho mejor conceptos del ciclo. Los lenguajes utilizados para la página web (HTML, CSS, Javascript) y los utilizados en la programación del robot (C++), nos han nutrido mucho académicamente y nos vemos más preparados para empezar el ciclo superior el año que viene.

Llevar a cabo este proyecto ha nos a supuesto un gran reto, buscando constantemente información y formándonos en las tecnologías que hemos utilizado. Nos ha servido como motivación y a incrementado nuestra motivación para los siguientes cursos, además que ahora estamos más preparados para solucionar problemas que nos sucedan, sobretodo a nivel de programación. Gracias al gran esfuerzo que hemos hecho, se ha notado una mejoría de nuestros conocimientos académicos. Estamos muy orgullosos del proyecto y satisfechos del resultado obtenido.

7.2 Consecución de los objetivos

El objetivo del proyecto era logran un robot con forma de serpiente pero para completarlo necesitamos plantearnos distintos objetivos más pequeños. A sido muy satisfactorio completar los objetivos propuestos a tiempo, y ver como avanzaba el proyecto.

Además hemos cumplido con las fechas propuestas para los objetivos, y aunque en algunos teníamos problemas y eran difícil de solucionar, al final se ha podido y a sido muy gratificante.

7.3 Valoración de la metodología y planificación

Creemos que la metodología de trabajo a sido buena, no hemos parado de trabajar, hemos repartido muy bien el trabajo entre nosotros, además de revisarlo siempre que acabamos entre los dos para ver que todo estuviese correcto.

Por otro lado, la planificación también a sido buena, tuvimos en cuenta desde un principio que podrían haber errores, y en el momento que los hubo, teníamos tiempo para solucionarlos y seguir adelante. Además no se nos acumulaba trabajo.

7.4 Visión de futuro

Creemos que este proyecto puede tener futuro como kit arduino, en el cual te vienen todas las piezas y unas instrucciones de montaje, junto a un enlace a una web donde estarían los distintos códigos.

Sería un producto dirigido a los más jóvenes, para que puedan aprender un poquito más sobre arduino, los componentes, he incluso para jugar con el robot una vez montado.

8. Glosario

1. [↑](#) Atom:
Es un editor de código fuente, de código abierto.
2. [↑](#) Meccano:
Es un sistema de construcción de modelos. Consiste en piezas de diversos tamaños, forma y color construidas en metal con filas de agujeros para sujetarlas a otras piezas por medio de tornillos y tuercas.
3. [↑](#) OTTO:
Es un bípedo (dos patas) que puede andar, bailar, hacer sonidos y evitar obstáculos. Es totalmente Open Source y se programa fácilmente desde el entorno de Arduino, así que es ideal para empezar en Arduino.
4. [↑](#) Navbar:
Es dispositivo que permite convertir una señal eléctrica en una onda de sonido.
5. [↑](#) Navbar:
Es una parte de la interfaz gráfica del usuario, destinada a ayudar a los visitantes de la web a acceder a la información.
6. [↑](#) Footer:
Es una sección ubicada debajo de la parte principal de la página web.
7. [↑](#) Image slider:
Se colocan generalmente en el inicio y consisten en una serie de diapositivas que van pasando cada cierto tiempo.
8. [↑](#) HyperLink:
Un HyperLink es un elemento de navegación en un document que conduce a otra sección del mismo documento.
9. [↑](#) Responsives:
La idea de una pagina web responsive, es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visitarlas.

9. Bibliografia

<https://www.w3schools.com/>

<https://create.arduino.cc/>

https://naylampmechatronics.com/blog/10_Tutorial-de-Arduino-y-sensor-ultras%C3%B3nico-HC-S.html

<https://www.youtube.com/watch?v=YKICzcH8oWk>

<https://www.youtube.com/watch?v=yzTJ-KY-FC8>

<https://www.instructables.com/id/Bioinspired-Robotic-Snake/>

<https://www.youtube.com/watch?v=GeirgHVc9WE&t=6s>

<https://fritzing.org/home/>

<https://www.tinkercad.com/dashboard>

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

<https://es.aliexpress.com/>

10. Anexos

El código mencionado anteriormente, es el principal, pero no es el único. Hay más código escrito pero que no está en funcionamiento, esta por si alguien lo quiere usar, que pueda decidir el movimiento que haga el robot.

El código es el siguiente:

```
void iniciar(){
int inicios[]={90,125,90,55,90,125,90,55};

    for(int i=0;i<8;i++){
        myservos[i].write(inicios[i]);
    }

}

void inicializardxdeavanzar() {

    dxs[0]= -1;
    dxs[1]= -1;
    dxs[2]= 1;
    dxs[3]= 1;
    dxs[4]= -1;
    dxs[5]= -1;
    dxs[6]= 1;
    dxs[7]= 1;
}

void oldavanzar{

    int i = 0;
    int vuelta = 140;
```

```

int total[8];
for(int i=0; i<8;i++){
    total[i]=myservos[i].read();
}

while (i < vuelta) {
    //Escribir servo
    for(int i=0;i<8;i++){
myservos[i].write(total[i]); // El segundo
    }

    imprimeServos();

    for (int i=0; i<8; i++){
        if (total[i] >= 125) dxs[i] = -1;
        if (total[i] <= 55) dxs[i] = 1;
    }
    //Sumar los grados
    for(int i=0;i<8;i++){
        total[i]=total[i]+dxs[i];
    }

    //Incrementar la iteracion

    i++;

    if (distanciaMenorQue(10)){
        break;
    }
}

```

Para poder usarlo hay que cambiar una pequeña parte del documento, dentro de void setup. Hay que borrar los elementos que hay a continuación:

```

// Declaramos la variable array dxrecta para i
int dxrecta[]={1,1,1,1,1,1,1,1};
inicializadx(dxrecta);

```

```
// Llamamos a la funcion recta
recta();
}

//Declaramos la funcion inicializardx, que necesi
void inicializardx(int* losdx) {
    copyB(losdx,dxs,8);
}
```

Y sustituirlo por lo siguiente:

```
inicializardxdeavanzar();

iniciar();
```

Por último hay que cambiar dentro del if que hay dentro de void loop, “avanzar” por “oldavanzar”.

De esta manera estaríamos utilizando el otro código.