



DEPARTAMENTO DE INGENIERÍA MECÁNICA
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE
ME4250 - MECATRÓNICA

MECATRÓNICA

Proyecto 1: Octo Magic Hat

Disfraz Animatrónico

Benjamín Rojas Bustamante

Profesora:

Carolina Silva

Auxiliar:

Leslie Cárdenas V.

Entrega:

9 de mayo de 2023

Resumen

En este documento se muestra el proceso de diseño y construcción de un disfraz animatrónico, en el marco del primer proyecto del curso Mecatrónica, consiste en diseñar un artículo electro-mecánico utilizando control de lazo abierto mediante la plataforma open source: *Arduino*.

El objetivo de este proyecto es integrar componentes electrónicos esenciales con herramientas de programación y técnicas de manufactura para el desarrollo de prototipos funcionales.

Este proyecto en particular, consiste en un sombrero similar a los utilizados en trucos de magia, dentro del cual se encuentra un modelo de un *pequeño pulpo* el cual sale del sombrero al accionar un pulsador y efectúa una breve interacción.



1. Diseño

Este proyecto, nace de la necesidad de crear un disfraz animatrónico. Las primeras ideas fueron disfraces icónicos como una máscara de Ironman, o una corbata retráctil pero luego de pensarlo, opté por un diseño más original y representativo del atuendo de un mago, un sombrero.

La idea de un sombrero que tenga oculto un personaje que pueda salir y efectuar una pequeña interacción con el público, la idea principal surge de una mezcla entre un concepto proveniente de la magia y de la forma de saludar, sosteniendo la solapa de un sombrero de copa como se hacía a comienzos del siglo XX. A continuación se muestra el proceso en detalle desde el primer concepto, pasando por algunas iteraciones hasta la versión final de este sombrero.

1.1. Primeros trazos...

Teniendo ya la idea en mente, decidí utilizar *Fusion 360* como herramienta de prototipado debido a la facilidad de modificar los elementos del proyecto así también para darme una idea acerca de posibles interferencias entre componentes y también visualizar el movimiento de los posibles mecanismos a utilizar.

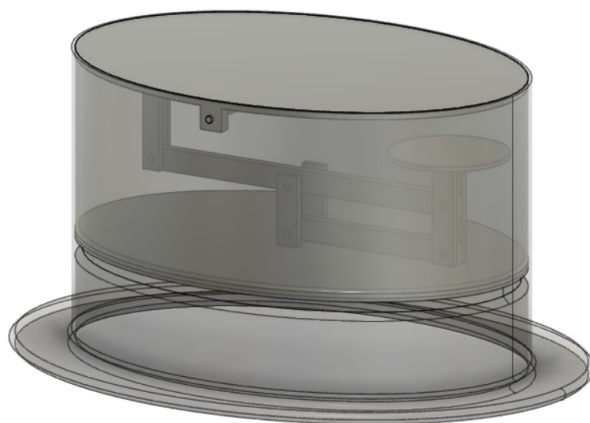


Figura 1: Primer bosquejo del sombrero.

1.2. Mecanismo de interacción

Lo primero que debía plantear era el mecanismo que debía ejecutar la salida del modelo, para esto pensé en una plataforma de doble brazo, la cual permite un movimiento rotacional de una plataforma dejando esta última siempre en una posición horizontal.

1.2.1. Primer bosquejo del mecanismo

El primer mecanismo planteado fue el que se muestra en la Figura 2, la idea central era que, mediante un servomotor, el mecanismo pudiera abrir la tapa del sombrero con una "bisagra", al mismo tiempo que exponía una plataforma hacia afuera. El problema con este mecanismo era que al salir la plataforma del sombrero, esta realizaba una rotación antihoraria hacia la parte posterior del sombrero, lo que hacía que el modelo escondido dentro, pudiera ser no visible por otra persona, además este mecanismo ocupaba un espacio considerable, lo que podría significar un problema al momento de ubicar los otros componentes dentro del sombrero.

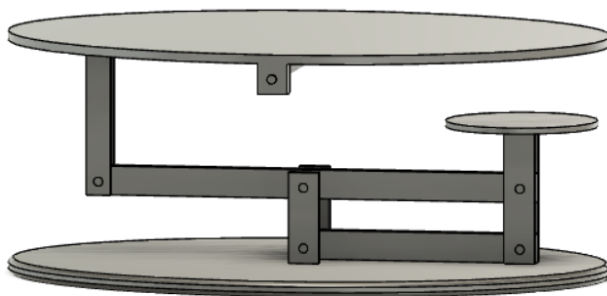


Figura 2: Diseño preliminar del mecanismo de interacción.

1.2.2. Mecanismo definitivo

Luego de un tiempo de reflexión, decidí inspirarme en el mecanismo utilizado en las "Useless Box" las cuales tienen un funcionamiento mecánico similar a lo que necesitaba, lo que hacen estas cajas es, al accionar un interruptor, un pequeño brazo sale desde la caja y cambia el estado del interruptor. En la Figura 3 se muestra una de estas cajas.



Figura 3: Useless Box.

En base a esta inspiración, nace el primer bosquejo de este diseño el cual se muestra a continuación en la Figura 4.

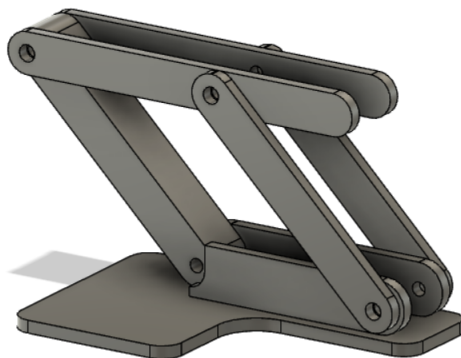


Figura 4: Primer bosquejo del mecanismo definitivo.

Este modelo fue impreso en 3D para testear el funcionamiento real del mecanismo utilizando un servomotor, sin embargo, presentaba ciertos problemas tales como el exceso de trabajo que debía ejercer el servo al mover el brazo posterior que se aprecia en la Figura 5, además de la flexión que presentaban algunos elementos del mecanismo. Esto provocaba que el mecanismo no tuviera un recorrido completo y forzaba en exceso el servomotor, de igual manera, durante esta etapa, decidí agregar un segundo servomotor el cual permitía el movimiento independiente del modelo de interacción.



Figura 5: Primera versión del mecanismo impreso.

Los problemas mencionados anteriormente se abordaron cambiando el diseño de algunos elementos, favoreciendo la resistencia a la flexión de la estructura, cambiando la posición del servo hacia el brazo frontal y añadiendo un soporte para el segundo servomotor en la plataforma. Luego de pasar por 3 iteraciones de diseño, se llegó al diseño mostrado en la Figura 6.

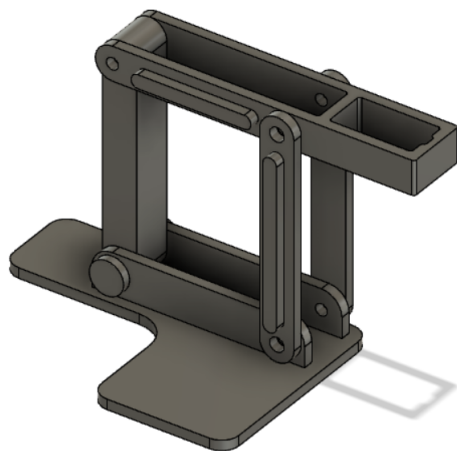


Figura 6: Segunda versión del mecanismo definitivo.

Con este diseño se logra disminuir la carga del servomotor y se integra un espacio para el segundo servomotor al mismo tiempo que la estructura es más resistente.

Finalmente, el mecanismo final se muestra en la Figura 7 junto con lo cual, tomé la decisión de incluir como modelo de interacción un pulpo con tentáculos móviles el cual es bastante compacto y llamativo mostrado en la Figura 8.

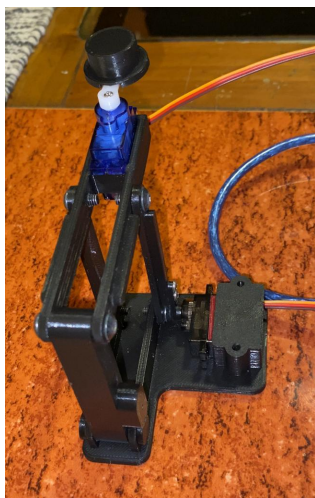


Figura 7: Segunda versión del mecanismo impreso.

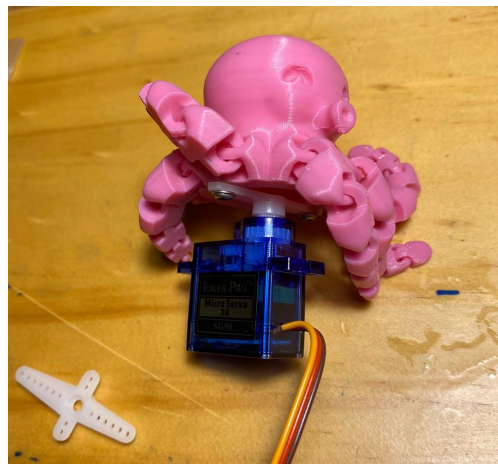


Figura 8: Pequeño octopus impreso y unido a un servomotor.

Para terminar, realicé una pequeña modificación la que consistió en añadir un acople en el servomotor azul, el cual se aprecia en la Figura 7 y tiene como función aprovechar mejor el movimiento del modelo y tener una interacción más llamativa, ya que permite que el octopus salga un poco más hacia afuera del sombrero.

2. Electrónica

Al momento de pensar en la interacción de este proyecto, quería que esta fuese llamativa y simple para no hacer más compleja la programación.

El mecanismo de interacción contempla el uso de 2 *micro servomotores*, un *pulsador* para accionar la interacción y un *buzzer* o emisor de sonido. Estos componentes son puestos en marcha mediante un programa de control por lazo abierto utilizando una placa *Arduino NANO*. Además, la electrónica es alimentada mediante una batería externa de 5 volts. El esquema de conexiones y componentes se puede ver a continuación en la Figura 9.

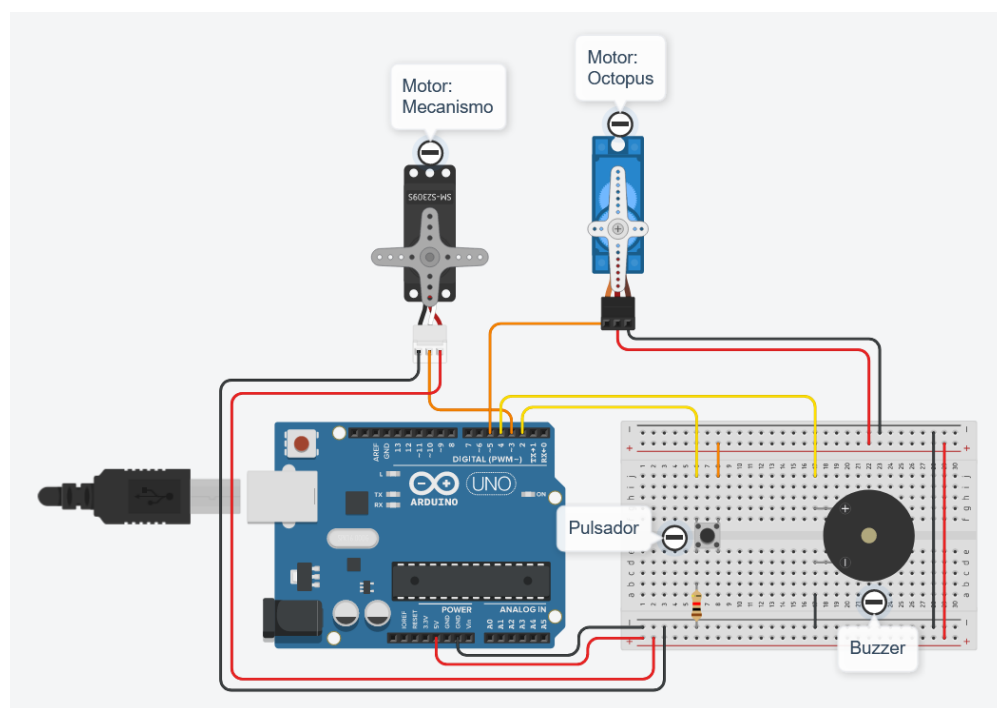


Figura 9: Esquema de conexiones.

Estos componentes fueron colocados sobre una base de policarbonato la cual irá escondida dentro del sombrero. A continuación, en la Figura 10 se muestran los componentes mencionados más arriba, a excepción del pulsador, el cual va escondido dentro de la solapa del sombrero, pero eso se mostrará más adelante, sin embargo, se pueden ver 3 cables saliendo de la placa, los cuales corresponden a las conexiones pulsador.

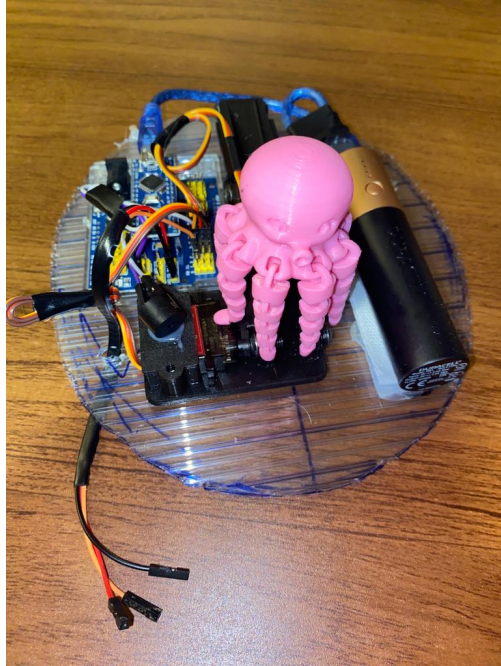


Figura 10: Electrónica utilizada en el proyecto.

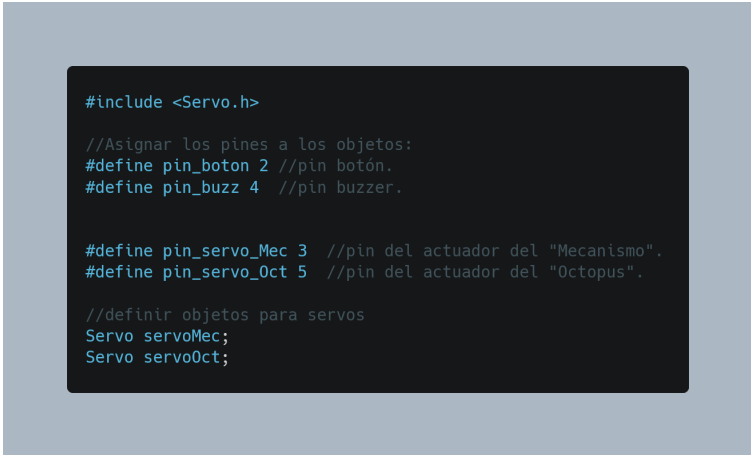
3. Código

En cuanto a la programación del proyecto opté por confeccionar un pequeño guión para la interacción que debe realizar el sombrero, estas instrucciones son:

1. Establecer una posición de reposo o inicial dentro del sombrero.
2. El mecanismo es activado para dar salida al modelo de octopus.
3. Se emite un sonido emulando un "saludo".
4. El modelo de octopus se mueve ligeramente de un lado a otro, emulando una "mirada de inspección".
5. Se emite un sonido emulando una "despedida".
6. El mecanismo es activado para esconder nuevamente el modelo de octopus dentro del sombrero.

Cabe mencionar que la programación se realizó en *Tinkercad* con el fin de poder ver la simulación instantánea de los componentes, luego el código se traspasó a la IDE de *Arduino* para cargar el programa en la placa *NANO*.

3.0.1. Configuración



```
#include <Servo.h>

//Asignar los pines a los objetos:
#define pin_boton 2 //pin botón.
#define pin_buzz 4 //pin buzzer.

#define pin_servo_Mec 3 //pin del actuador del "Mecanismo".
#define pin_servo_Oct 5 //pin del actuador del "Octopus".

//definir objetos para servos
Servo servoMec;
Servo servoOct;
```

Figura 11: Configuración y definición de variables.

3.0.2. Setup

A continuación, se muestra el *setup* del código, cabe destacar de esta sección del código el movimiento inicial del modelo de tal manera que al energizar el circuito, el modelo se mueva a su posición de reposo, dentro del sombrero.

```
void setup()
{
  //attach de los pines a los objetos "servos".
  servoMec.attach(pin_servo_Mec);
  servoOct.attach(pin_servo_Oct);

  //Pin Modes...
  pinMode(pin_servo_Mec, OUTPUT); //Mecanismo.
  pinMode(pin_servo_Oct, OUTPUT); //Octopus.
  pinMode(pin_boton, INPUT); //Botón.
  pinMode(pin_buzz, OUTPUT); //Buzzer.

  //predefinir posición de los servos:
  servoMec.write(10); //Posición dentro del sombrero.
  servoOct.write(60); //Octopus en dirección hacia el frente.
}
```

Figura 12: Setup del código.

3.0.3. Loop

Luego se muestra el *loop* del programa en donde se debe destacar el uso de funciones para realizar ciertas acciones, decidí utilizar funciones para facilitar la modificación de las acciones de interacción, además de poder refinar la misma o incluso mejorarla a futuro. Las funciones mencionadas en el código se definirán a continuación.

```
void loop()
{
  //el programa espera hasta que se presione el pulsador:
  if (digitalRead(pin_boton)) {

    servoMec.write(90); //activación del mecanismo.
    delay(500);
    buzz_hola(); //función definida más abajo.
    delay(500);
    headServo_hola(); //función definida más abajo.
    delay(500);
    buzz_adios(); //función definida más abajo.
    delay(500);

  }
  //condicional de reposo:
  else {
    servoMec.write(10); //mecanismo vuelve a su posición de reposo.
    servoOct.write(60); //El octopus vuelve a su posición de reposo.
  }
}
```

Figura 13: Código loop del programa.

3.0.4. Funciones de interacción

```
//Melodia de "saludo" en buzzer:  
//Emite 3 sonidos de distintas frecuencias emulando un saludo...  
void buzz_hola()  
{  
    tone(pin_buzz, 440, 100); //tone(pin, frecuencia, t_on)  
    delay(100);  
    tone(pin_buzz, 340, 100);  
    delay(100);  
    tone(pin_buzz, 500, 100);  
    delay(100);  
}
```

Figura 14: Función: Sonido de "saludo".

```
//Melodia de "despedida" en buzzer:  
//Emite 2 sonidos de distintas frecuencias emulando una despedida...  
void buzz_adios()  
{  
    tone(pin_buzz, 500, 100);  
    delay(100);  
    tone(pin_buzz, 440, 100);  
    delay(100);  
}
```

Figura 15: Función: Sonido de "despedida".

```
//Movimiento del pulpito de lado a lado:  
//Nota: el servo apunta al frente a 60°.  
void headServo_hola()  
{  
  servo0ct.write(30);  
  delay(200);  
  servo0ct.write(90);  
  delay(500);  
  servo0ct.write(60);  
}
```

Figura 16: Función: movimiento de interacción del octopus.

4. Construcción sombrero

La construcción del sombrero fue realizada de manera muy concreta, consiste en una estructura de cartón corrugado y policarbonato para la solapa, todos estos elementos están forrados con tela de felpa.



Figura 17: Sombrero forrado en felpa, con un pulsador oculto y una plataforma interna de soporte para la electrónica.

Lo importante a destacar en esta sección es que en la solapa del sombrero, está oculto el pulsador en la zona en donde usualmente se sostenía al saludar. El circuito del pulsador se puede apreciar en la Figura 9 y para lo mostrado en la Figura 19, se aprecian 3 cables, uno de señal (naranja), uno para el positivo (rojo) y uno de tierra con una resistencia conectada (negro).

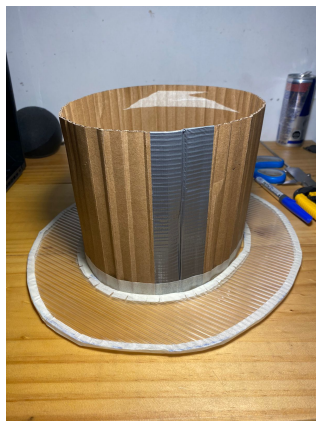


Figura 18: Estructura interna del sombrero



Figura 19: Pulsador dentro de la solapa.

5. Montaje

A continuación se muestra el montaje de cada uno de los componentes mencionados en los puntos anteriores. Se inicia con la base del sombrero mostrada en la Figura 17, luego se introduce la base que contiene la electrónica dentro del sombrero quedando como se muestra a continuación.

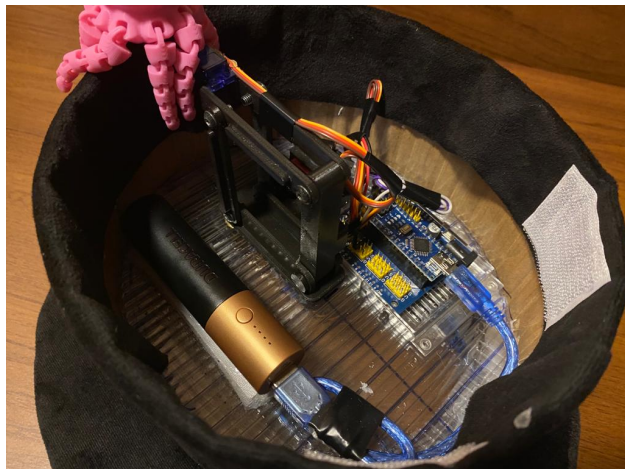


Figura 20: Electrónica dentro de del sombrero.

Posteriormente, se realiza la conexión del pulsador por debajo de la plataforma de electrónica.



Figura 21: Conexiones del pulsador.

Posteriormente, se realiza el montaje de la tapa del sombrero, construida con un trozo de policarbonato que posee unas ranuras que facilitan la apertura por flexión, esta se conecta mediante dos cintas de *velcro*. Junto a esto, la tapa incluye un pequeño imán el cual permite que mientras el modelo esté dentro del sombrero, la solapa permanezca cerrada.



Figura 22: Fijación de la tapa usando cintas de *velcro*.

Finalmente, se tiene el sombrero ya montado, sólo falta encender la batería y ejecutar el programa presionando el pulsador.



Figura 23: Sombrero mágico.

6. Comentarios y consideraciones para siguientes iteraciones

6.1. Mejorar el código para interacciones más pulidas

En un futuro cercano, pretendo mejorar el código de manera que la interacción del modelo dentro del sombrero, pueda realizar acciones más fluidas y llamativas, al mismo tiempo, incluir más de un programa de interacción para favorecer la dinámica y variación de funcionamiento.

6.2. Sistema de seguridad de la batería externa

La mayoría de las baterías externas o "power banks" cuentan con un sistema de seguridad el cual permite apagar la batería cuando se consume muy poca corriente o incluso detecta un "no consumo", esto es un problema para este proyecto debido a que cuando el sombrero se encuentra en reposo, el consumo del Arduino es muy pequeño, por lo cual la batería entiende que no hay nada conectado a ella y se apaga, interrumpiendo por completo el funcionamiento del circuito.

Existen circuitos que "simulan" un consumo de tal manera de dejar encendida la batería para que el programa funcione sin tener que están encendiendo la batería todo el tiempo. El uso de estos circuitos u otra estrategia se evaluará luego de la primera entrega del proyecto.