

# TERRY

## I. DESCRIPCIÓN

El proyecto consiste en el desarrollo de un coche con dos modalidades, una de control remoto con un mando y otra capaz de seguir una línea negra de manera autónoma.

## II. DESARROLLO

Este proyecto consta de los siguientes elementos de desarrollo de software:

- Scripts de Python para adaptar los sensores
- Scripts de Python para el control del coche
- Scripts de Bash para automatizar tareas: inicio de programas, conexión de mando y descarga de dependencias
- Desarrollo de una interfaz gráfica con Flask
- Entorno virtual tervenv con las librerías necesarias
- README.md con las instrucciones para ejecutar



```

drivers
> pyache
# init.py
# car_eyes.py
# car_lines.py
# carpy
# clase_infrarrojo.py
# clase_ultrasonidos.py
# controller.py
# eyes.py
# lines.py
# motors_eyes.py
# motors.py
> tervenv

utils
$ remote.sh
$ requirements.txt
$ setup.sh

web
> __pycache__
$ main.py
$ README.md
$ terry_eyes.sh
$ terry_lines.sh
$ terry.sh

$ static
$ css
# style.css
$ images
$ ghibli_car.png
$ templates
$ index.html
$ app.py

```

Figura 1. Estructura de módulos.

En cuanto al desarrollo de hardware:

- Carrocería como soporte del coche
- Motores
- Ruedas
- Pilas
- Power Bank
- 2 sensores infrarrojos
- 2 sensores ultrasonidos
- Mando para control remoto
- Raspberry Pi

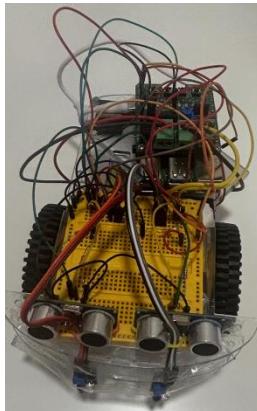


Figura 2. Componentes.

## III. RESULTADOS

Se dividen los programas del coche en 3 modalidades debido a la limitación de la corriente.

En el modo de exhibición, auxiliar, los motores están desactivados, pero se puede apreciar la interfaz gráfica y la activación de los sensores



Figura 3. Interfaz gráfica

En el manual, el mando se conecta a la placa para guiar la dirección, mediante los botones A (acelerar recto), B (marcha atrás), X (parada), Y (parada completa, fin del programa), ZL (giro a la izquierda) y ZR (giro a la derecha). El coche para automáticamente si hay un obstáculo cerca.

En el automático, El coche se sirve de una línea negra en el suelo y los infrarrojos para guiar la dirección sin necesidad de intervención por parte del usuario.

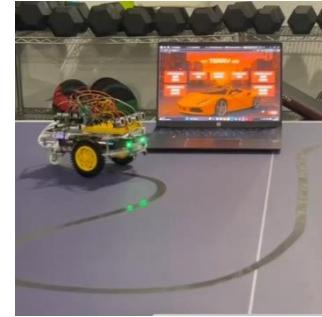


Figura 4. Control automático

## IV. LIMITACIONES

- Se ha tenido que separar en distintos programas, la idea principal era poder cambiar entre modos mediante el mando. Sin embargo, es necesario conectar y desconectar los sensores para que cada modo funcione correctamente.
- Los motores funcionan con un factor de corrección distinto. Además, una rueda no colocada correctamente dificulta la trayectoria del coche.

## V. SOLUCIONES

- Crear scripts de bash para activar el entorno y conectar el mando de manera automática para agilizar el cambio entre modos
- Corregir la dirección del coche cada poco tiempo apagando y encendiendo los motores. Disminuye la velocidad, pero aumenta la precisión en el recorrido y ayuda a controlar la rueda descolocada.