RLP – Computer Engineering UAB (Barcelona) 2022-2023

EYECAR

A self-driving car controlled by gaze, using Computer Vision eye-tracking, which also detects the environment and prevents the car from crashing.

PROJECT FINAL SPRINT #6.
DATE: 05 June 2023

Aguilera Oliver, JOSE FRANCISCO
Barranco Aguilar, SERGIO
Bermúdez Valle, PAU
Karzazi El Bachiri, SAMYA

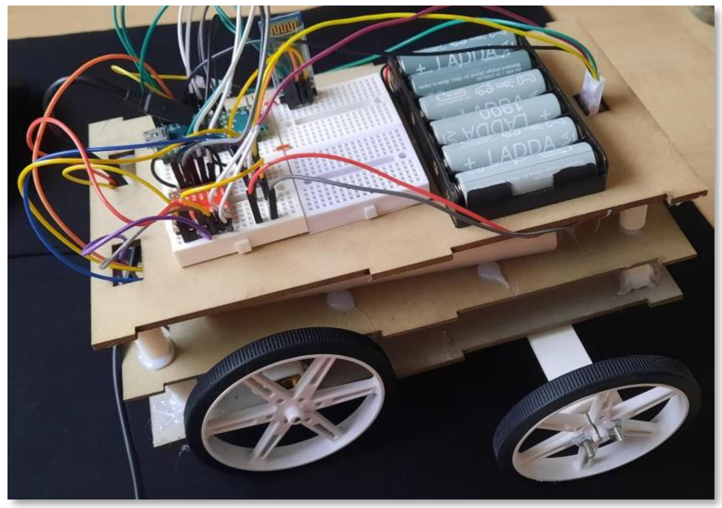
Table of Contents

A self-driving car controlled by gaze, using Computer Vision eye-tracking, which also detects the environment and prevents the car from crashing.

Project description

eyeCar is a self-driving car controlled by gaze, using Computer Vision eye-tracking. Create for people that cannot drive.

This car includes a camera that also detects the environment such as the road, objects, people, other cars and prevents it from crashing by stopping and not turning in the direction you have ordered. By doing so we ensure that no accidents happen and everything around us is safe.

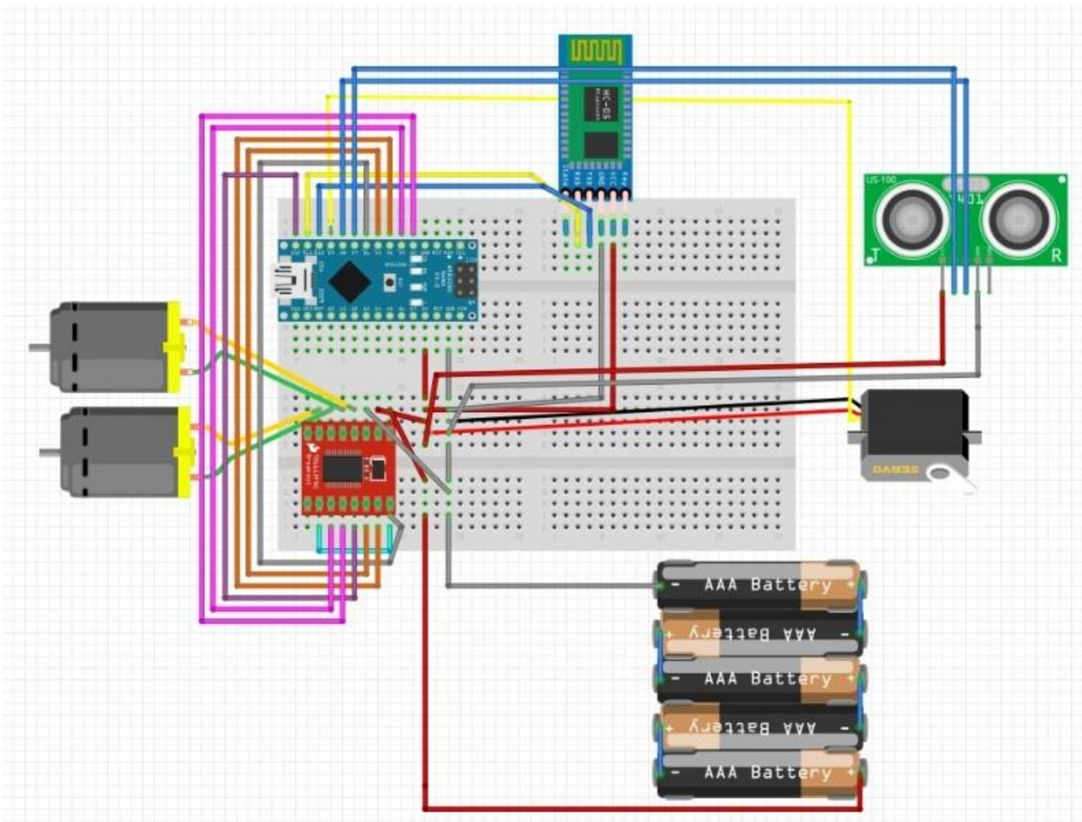


Electronic components

This is the list of the proposed components:

- *HC-SR04 (ultrasonic sensors)*
- *Arduino Nano*
- *Micro servomotor SG90*
- *x2 Micromotor with gear reducer 10:1*
- *Motor controller driver 1a Dual TB6612FNG*
- *Battery holder 6xAA*
- *Protoboard*
- *HC-05 Bluetooth*

Hardware Scheme



Arduino D12 --> STBY controller

Arduino D11 --> RDX HC-05

Arduino D10 --> TDX HC-05

Arduino D9 --> Servo

Arduino D8--> Echo Sensor

Arduino D7--> Trigger Sensor

Arduino D6--> Power Motor 1 & 2

Arduino D5--> Ain1 controller

Arduino D4-> Ain2 controller

Arduino D3--> Bin1 controller

Arduino D2--> Bin2 controller

Arduino 5v --> VCC controller & VCC HC-05 & VCC Servo & VCC Sensor

Arduino GND--> GND controller & GND HC-05 & GND Servo & GND Sensor

Software Architecture

main():

Responsible for executing all the functions needed to action our eyeCar.

lectorPupila():

This function uses the laptop's camera to detect the position of the pupil to estimate its movement. So, this function updates the movement values. Depending on the gaze direction, it will modify the speed or braking direction value.

- Look to the left: turn left.

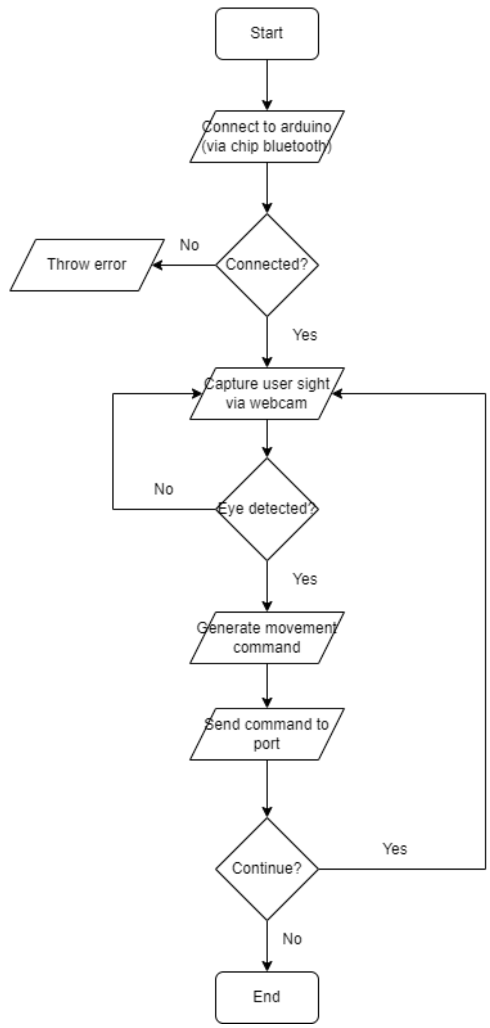
- Look to the right: turn right.

- Look up: accelerate.

- Look down: brake.

- Other actions not yet decided.

To make this possible we utilize a pretrained model that detects and crops the face region of the person in front of the camera. Then we split this region in two, to get the right and left sides of the face. We use another pretrained model that crops the eye region on both sides, giving us a real time frame of our left and right eye. We could use the two eyes to estimate the pupil position, but in our case, we only used the right eye for the further calculations.



Next step is to convert the eye frame to grey scale, apply some noise reducing filters and apply a thresholder to binarize the pupil from the rest of the eye. Once we have the pupil binarized we extract the centroids, which represent the centres of the main binarized sections of the image. The centroid of the biggest section will be, theoretically, the centre of our pupil.

We now do a process of synchronization, where we ask the user to look at each direction and press enter. With this process we save the coordinates of the centroid detected on each direction so that, thereafter, we can predict where the user is looking comparing the actual coordinates with the coordinates of the sync process and giving as a result the direction closest to.

movement():
This function is responsible for giving instructions to the two motors and servo, making the car move as a result. We send the commands as numbers to our robot via a port of our Wi-Fi chip.
The directions, speed and motor braking commands will be given by the *readerPupila()* function, with the received values we will update the direction using the rotation of the servomotor or increasing and decreasing the speed of the motors at will.

Amazing contributions

Our project aims to provide and contribute to people with reduced mobility and added disabilities to ease their use of wheelchairs or other mobility devices.

We intend to achieve this goal through a comfortable use and a low difficulty curve, through "eye-tracking", since it will adapt to the directions desired by the user without least effort.

This project works testing the features using a miniature car model as a prototype, of course with safety in mind, being able to detect obstacles in both directions of traffic, to avoid any type of accident that could affect the user directly.

Taking all these points into account, we can affirm that our project has a large component related to the computer vision field, this being its incentive, although the structuring and designing part of the car must also be included, where we emphasized above all on speed control, user negligence probability and other factors.

We truly believe that our project has excellent potential and a goal of contributing our grain of sand to a needy and important part of society.

Extra components and 3D pieces

- *Ultrasonic sensor cover*
- *Wheel bolt bracket*
- *Micromotor bracket*
- *Layer support*
- *Front wheels axis*
- *Layer 1 and 2 – Power supplies and microelectronics*
- *Layer 3 – Wheel support*

In the annex we have included the sketch of what contains each layer visually.

Ultrasonic sensor cover: used to cover the ultrasonic sensor and stick it in the upper layer.

Wheel bolt bracket: used to hold the bolts of the wheels and attach them to the wheels' axis.

Micromotor bracket: designed to hold the micromotors and attach them to the base.

Layer support: to create a separation between layers.

Front wheels axis: used to turn the front wheels to the right or left using only one servomotor.

Layer 1 and 2 – Power supplies and microelectronics support: used to hold the protoboard, every cable, Arduino, the ultrasonic sensor, the portable battery, and the batteries.

Layer 3 – Wheel support: used to screw the motors and servomotor to hold the wheels.

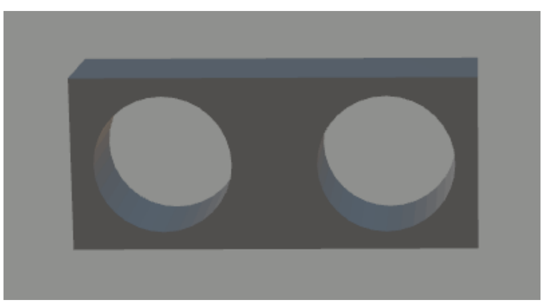


Fig. 1 - Ultrasonic sensor cover

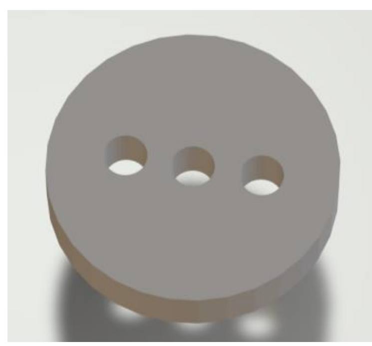


Fig. 2 – Wheel bolt bracket

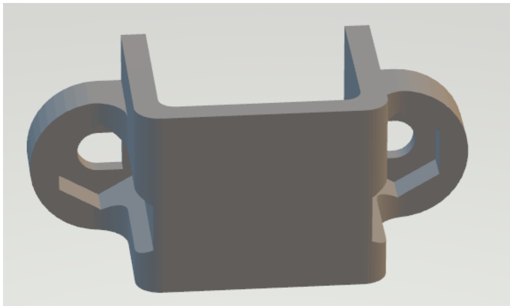


Fig. 3 – Micromotor bracket

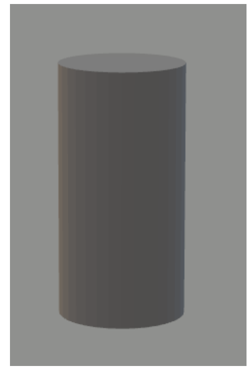


Fig. 4 – Layer support

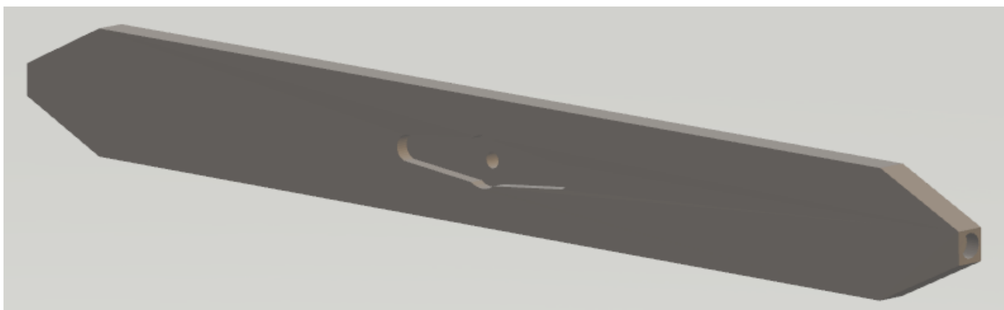


Fig. 5 – Front wheels axis

Fig. 6 - Layer 1 and 2 – Power supplies and microelectronics

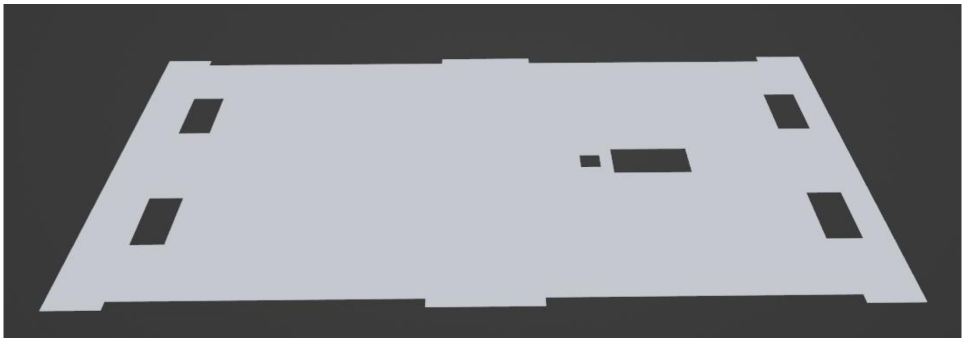


Fig. 7 - Layer 3 – Wheel support

Strategy for validation, testing and simulation

Our strategy for validation is to evaluate the car in a car circuit mounted by us and see if it follows our orders of changing the route. We also want to test if it detects objects by adding "people", objects, and other cars in the circuit. This strategy of validation will be done with a challenging simulation and a simple real circuit.

Foreseen risks and contingency plan

Risk #	Description	Probability (High/Medium/Low)	Impact (High/Medium/Low)	Contingency plan
Collision	If our robot moves too fast or our remote-control system fails, there may be a risk of collision with other objects or people.	Medium	Medium	Implement speed control systems and proximity sensors that detect obstacles and stop the robot before a possible impact. As well as stopping the robot immediately if the connection with the remote-control system by eye tracking is lost.

Internal damage	If the robot is handled incorrectly or receives an external shock, the electronic components could be damaged.	Medium	Medium	Design and print a 3D housing that offers some protection to all components of the robot. In addition, the housing should be as aerodynamic as possible to ensure a good displacement of the robot.
Burn	If our robot overheats or there is a short circuit, there may be a risk of the robot catching fire.	Medium	High	Add temperature sensors that stop robot operation if certain temperatures are exceeded. (Not implemented on final model)

References

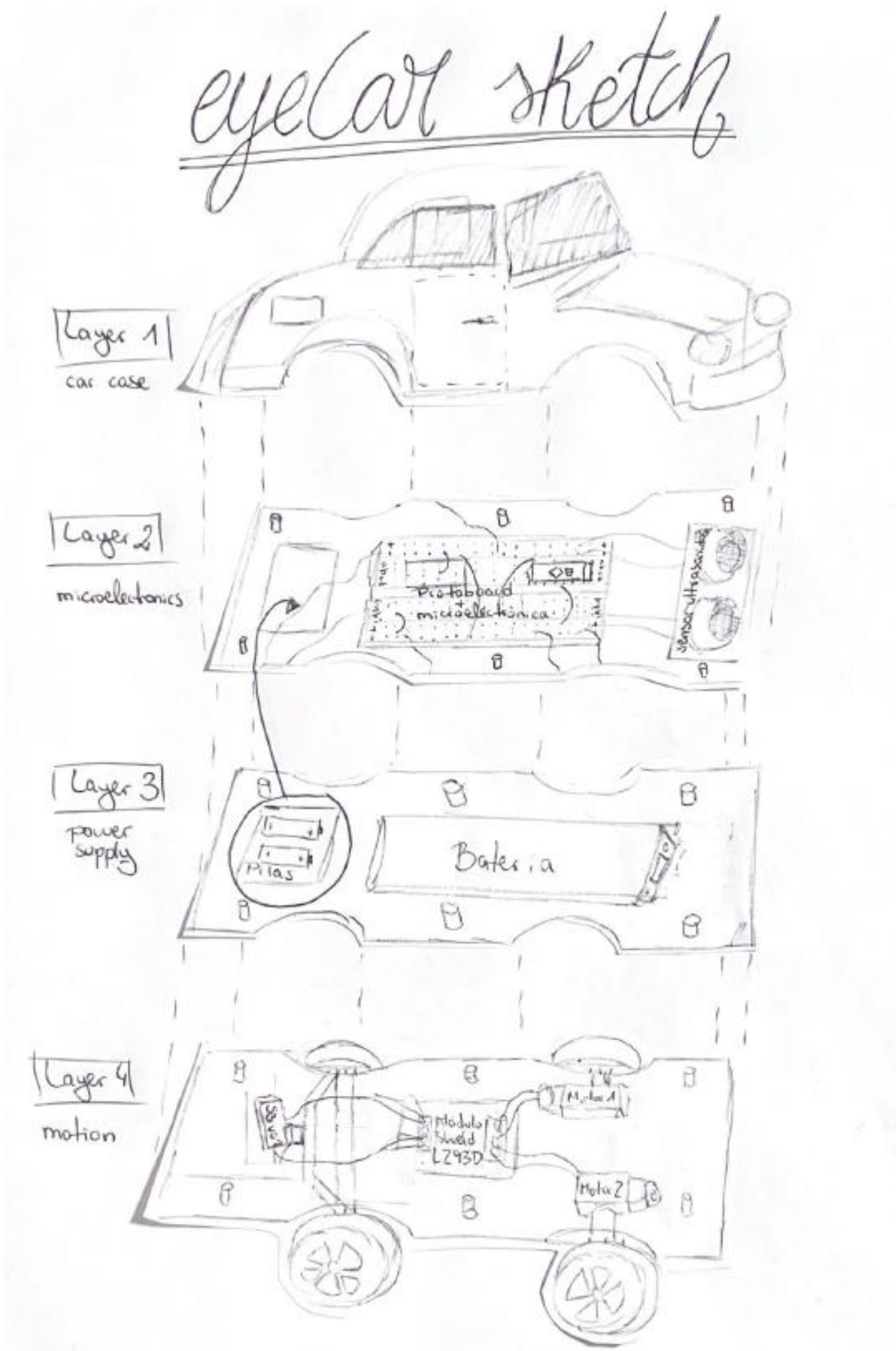
This project is inspired by the following Internet projects:

[1] https://projecthub.arduino.cc/aadhuniklabs/3eb4a951-a51e-4999-82ab-94a6128e91c4?ref=platform&ref_id=424_trending_part__&offset=6

[2] https://www.ijert.org/research/hand-gesture-control-car-IJERTCONV8IS05066.pdf

[3] https://visnav.in/ijabs/wp-content/uploads/sites/11/2021/12/Wireless-Robotic-car-control-through-human-interface-using-eye-movement.pdf

ANNEX


eyeCar sketch
Layer 1
car case
Layer 2
microelectronics
Protoboard
+
microelectronica
Layer 3
power supply
Pilas
Bateria
Layer 4
motion
Modulo
Shield
L293D
Motor 1
Motor 2