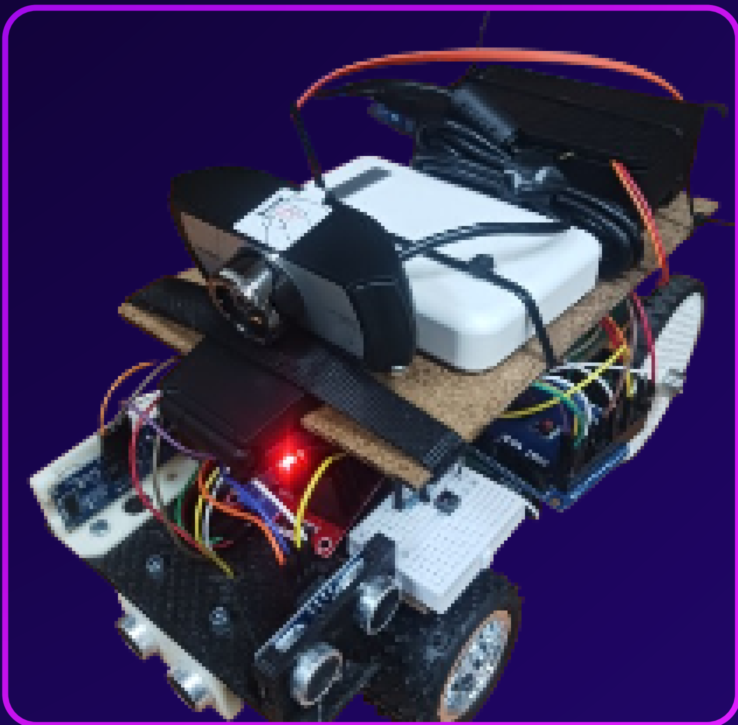
WORLD ROBOT OLYMPIAD™ WRO ITALY 2024

WORLD ROBOT OLYMPIAD™ WRO VENEZUELA

FUNDESTEAM Pensando • Construyendo • Creciendo

# OPEN CHAMPIONSHIP
# BRESCIA - ITALY 2024

## TEAM VIOLET



NUESTRA SEÑORA DE CHIQUINQUIRÁ
**HERMANOS MARISTAS**

EARTH ALLIES

FUTURE ENGINEERS

# CONTENTS

# INTRODUCTION



## TEAM PRESENTATION

We are Team Violet, the robotics team of the "Nuestra Señora de Chiquinquirá" Maristas school, in Zulia, Venezuela. We've been training for 3 years in a variety of projects and meetings in the area of robotics, both educational and competitive.

This is our second year participating in the Future Engineers category of the World Robotic Olympiad competition. This year, we achieved the second regional place, and we were able to attend the national edition, where we finished in ninth place nationwide.

## PARTICIPATIONS (REGIONAL)

To qualify for the nationals, we participated in 3 regionals, 2 of these at the "Los Robles" school and 1 more at the "Salto Angel" school.


### WRO "LOS ROBLES"

In these two editions, we put into practice the design that we had developed since May, achieving the scores of 24pts both in the first and second edition.

### WRO "SALTO ANGEL"

In this edition, after doing some minimum adjustments to the robot, we achieved the 53pts score, winning 1st place and qualifying to the national competition.

## TEAM MEMBERS

### Jose Miguel Zambrano

I am José Miguel Zambrano, the team's mechanic. I am in charge of the robot's assembly and its proper maintenance. For this season I hope, alongside my team, to deliver a project that meets the expectations, in addition to representing my institution and my country.

### Reinaldo Belmonte

My name is Reinaldo Belmonte. I am the team's programmer and I'm responsible for developing most (if not all) of the robot's moveset. My expectations for this season is to do my best with this project and, hopefully, end up in the first places.

### Adrian Alvarez

Hi, my name is Adrian Alvarez, I'm part of Team Violet and I'm responsible for the 3D designs used in the robot. My expectations of this competition are to learn and to test my skills in robotics.

# HARDWARE DESIGN



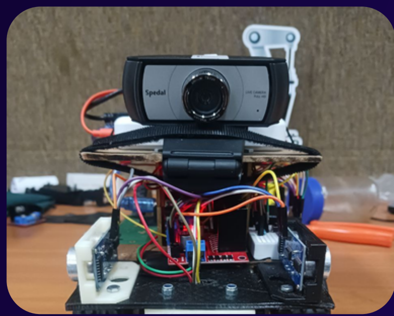
## DESIGN PROCESS

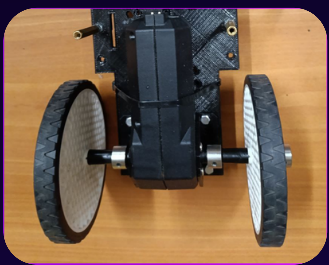
In the development process, one of the most important phases is logically the design stage, since it is the phase in which both technical and aesthetic decisions are generated. In this case, the design stage was vital to reflect and build on previous designs, learn from their failures and create something better.

## EVOLUTION OF THE DESIGN

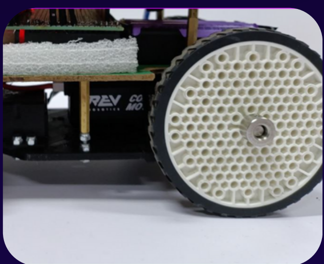
The robot is composed by a combination of 3D Printed parts for the most of its structure, the front wheels from a recicled toy car, the back wheels from a REV Robotics kit, and the electronics that give the robot its functionality.

it is worth mentioning that the robot went through a multitude of changes until it reached its final iteration:

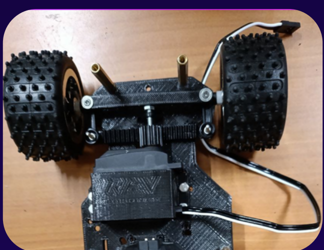




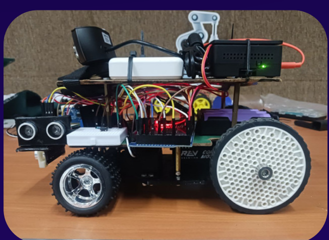
One of these changes was the replacement of the traditional DC motor that we used before with a «Hex Motor» , which provided a greater torque to the rear-wheel drive.



As a result, the wheels used for the rear-wheel drive had to be also replaced, since the old ones couldn't adapt to the torque provided by the Hex Motor. This way we could take advantage of all the potential the Hex Motor had to offer.



Alongside the rear-wheel drive, we also made changes to the turning system, replacing the old Servomotor with a «Smart Robot Servo» , allowing for better and more precise turns when meeting a corner.



Another of these changes was the reorganization of its elements, given the changes in the motors used and the space they now occupied, we made a redistribution to most of the elements present in the robot, this time taking into account to make it even more accessible for repairs or adjustments to any of its parts.

The final design of the robot is intended as a modular design, consisting of 3 modules: lower «traction» module, middle «sensor processing» module, and upper «power supply and color processing» module. This makes it easier to make repairs on any of the modules or to replace or rearrange parts.

# HARDWARE DESIGN



## LOWER "TRACTION" MODULE

This module is in responsible for managing the movement and direction of the robot, being where the Core Hex Motor, used as rear-wheel drive, and the Smart Robot Servo, used as steering system, are located.

**ELEMENTS PRESENT IN THIS MODULE:**


REV ROBOTICS'S SMART ROBOT SERVO


REV ROBOTICS'S CORE HEX MOTOR

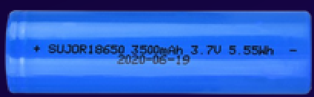
## MIDDLE "SENSOR PROCESSING" MODULE

This module is responsible for processing and managing the information supplied by the sensors that compose it. Additionally, in this module are present most of the electronic connections of the robot, one of the power sources of the robot, the 18650 batteries, and one of the microcontrollers of the robot, this being the Arduino Mega 2560 R3.

**ELEMENTS PRESENT IN THIS MODULE:**

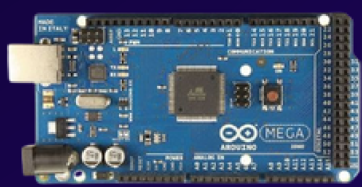

x3

HC-SR04

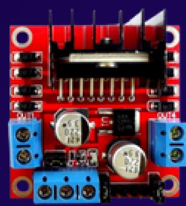


x3

18650 BATTERIES




ARDUINO MEGA 2560 R3



L298N

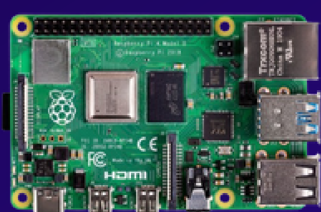
## UPPER "POWER SUPPLY AND COLOR PROCESSING" MODULE

This module handles the image and color processing functions through the other microcontroller of the robot, the Raspberry Pi 5. Additionally, to supply it with energy, there is an LDNIO Power Bank of 10000mAh.

**ELEMENTS PRESENT IN THIS MODULE:**

LDNIO 10000MAH POWER BANK

SPEDAL MF920P WEBCAM



RASPBERRY PI 5

# HARDWARE DESIGN



## ASSEMBLY PROCESS

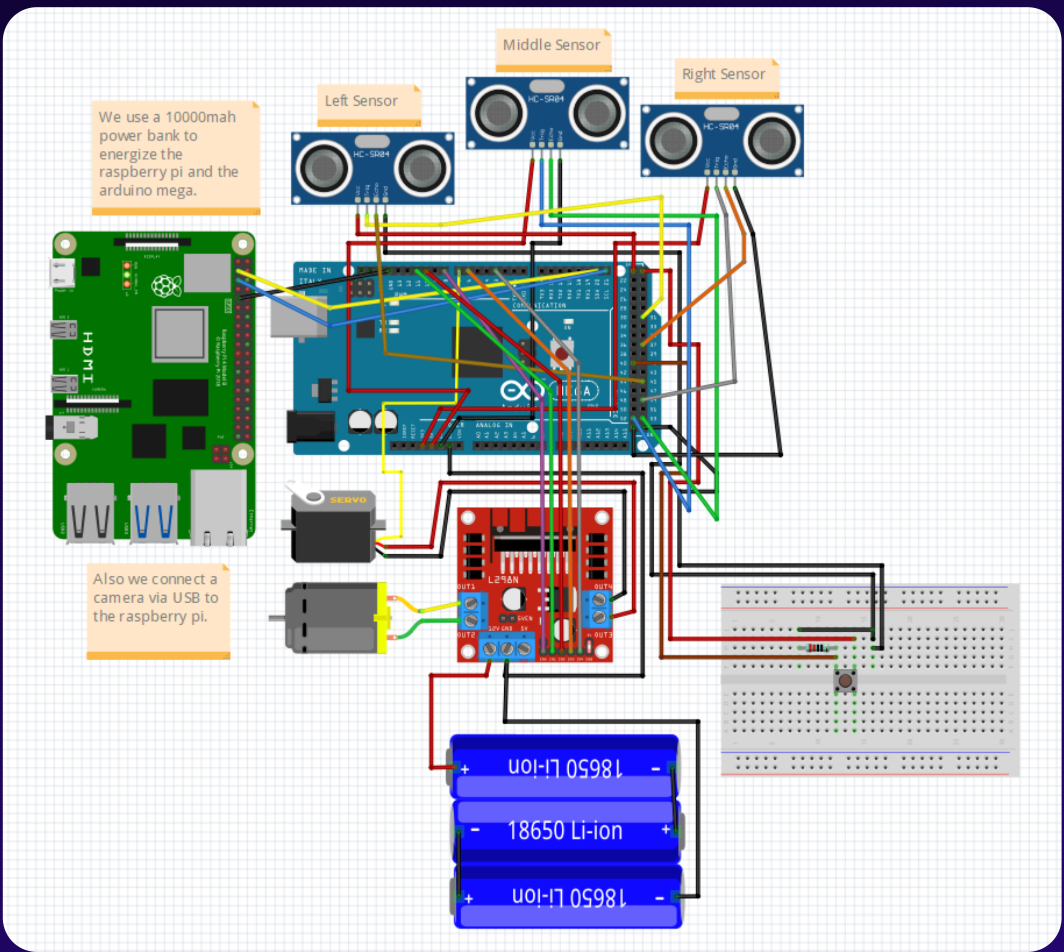
The assembly process of the robot was divided into each module separately to later unify all the modules into a single robot. The assembly of the lower module started looking for different references on the internet to find an ideal location to place all the corresponding motors. After finding the ideal locations for each motor, the parts of the main chassis and the turning system were designed and 3D printed, tailored to each element to optimize the performance of the robot.

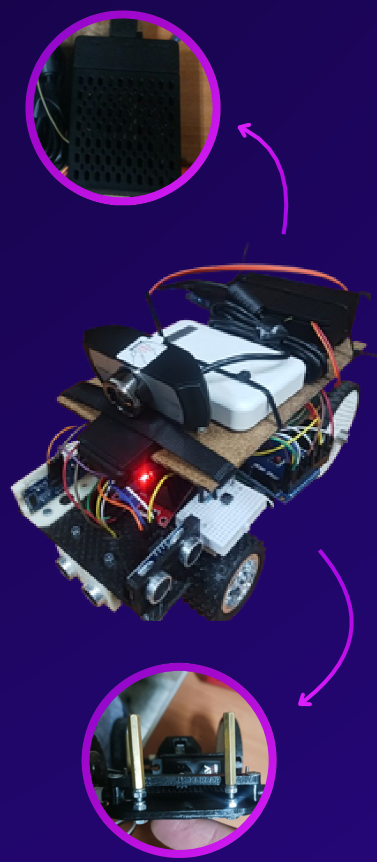
Following this, we started the assembly of the second module, being much easier to carry out, since instead of using a 3d printed piece, we used a piece of compressed cardboard as a base to place all the elements of the second module. When performing this part of the assembly, we performed some functional tests to ensure that everything worked properly; there we realized that the HC-SR04 sensors suffered from detection failures due to how they were placed in the robot, so we designed an extension of the base to place them there, thus solving the problem.

Finally, we start the assembly of the 3rd and last module, following a sequence very similar to the previous one: We use a piece of compressed cardboard as the base of the 3rd module and place the corresponding elements.

And to unify the design, we drilled holes in the 3 bases of each module (except the main chassis which was already designed with these holes in mind) and joined them with multiple "pillar shafts" in a tower shape.

You can see the respective connections in the diagram below:



# ELEMENTS



The electronic components used in the design are divided into 5 main sections:

## SENSORS

### • HC-SR04

The robot has 3 HC-SR04 ultrasonic sensors on the left, right and front of the "Bumper" piece, located in front of the middle module. They are used to provide orientation to the robot when moving in any environment, allowing it to detect obstacles or adjacent walls.


**SPECIFICATIONS**

- Supply voltage: 5 Vcc
- Working frequency: 40 KHz
- Minimum trigger pulse duration (TTL level): 10 µS.
- Minimum waiting time between one measurement and the start of another: 20 mS.
- Maximum range: 4.5 m
- Minimum range: 2.0 cm

### • SPEDAL MF920 WEBCAM

The robot uses an Spedal MF920P Webcam to recognize the colors in the environment. It has a Full HD 1080p resolution at 30 fps, with a H.264 Video Compression and a microphone incorporated.

## MOTORS

### • REV ROBOTICS'S CORE HEX MOTOR

The robot uses an Spedal MF920P Webcam to recognize the colors in the environment. It has a Full HD 1080p resolution at 30 fps, with a H.264 Video Compression and a microphone incorporated.


**SPECIFICATIONS**

- Output Shaft: 5mm Female Hex
- Voltage: 12V DC
- Stall Torque: 3.2 N-m
- Gear Ratio: 72:1
- At the motor: 4 counts/revolution
- Weight: 199g
- Free Speed: 125 RPM
- Stall Current: 4.4 A
- Encoder Counts per Revolution
- At the output: 288 counts/revolution

### • REV ROBOTICS'S SMART ROBOT SERVO

Located alongside the Hex Motor, it gives the robot the ability to rotate the front wheels, changing the orientation of the robot, thus allowing the robot to move freely in any direction on a flat surface.

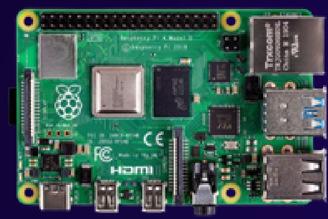

**SPECIFICATIONS**

- Speed: 0.14 s/60° (at 6V)
- Stall Current: 2A (at 6V)
- Stall Torque: 13.5 kg-cm / 187.8 oz-in (at 6V)
- Voltage Rating: 4.8V – 7.4V, 6V nominal
- Spline type: 25T
- Spline Internal Thread Size: M3
- Weight: 58g
- Input Pulse Range: 500µs – 2500µs
- Gear Material: Metal
- Default Angular Range: 270°
- Maximum programmable range in angular mode: 280°
- Spline Internal Thread Size: M3

# ELEMENTS 

## MICROCONTROLLERS

- **RASPBERRY PI 5**

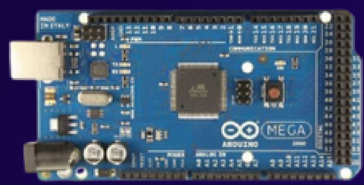


The Raspberry Pi 4 is a single-board development board used for entry-level computing. It is located in the upper module and it has the purpose of managing the "Color Processing" section, thus allowing the robot to recognize adjacent colors and use this information advantegiously.

### SPECIFICATIONS

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- 4GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- 2 × micro-HDMI® ports (up to 4kp60 supported)
- Micro-SD card slot for loading operating system and data storage
- Operating temperature: 0 – 50 degrees Celsius ambient
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- OpenGL ES 3.1, Vulkan 1.0
- 5V DC via GPIO header (minimum 3A*)
- 5V DC via USB-C connector (minimum 3A*)

- **ARDUINO MEGA 2560 R3**



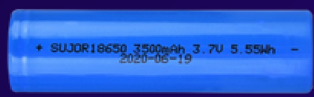

Located in the middle module, it is used to manage the robot's mobility alongside the L298N Motor Driver, thus allowing the robot to move freely on a flat surface.

### SPECIFICATIONS

- Microcontroller: ATmega2560
- Input voltage: 7-12V.
- 16 analog inputs.
- 256k flash memory.
- 16Mhz clock speed.
- 54 digital Input/Output pins (14 of them are PWM outputs).

## POWER SUPPLY

- **18650 BATTERIES**



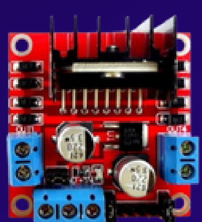

Used to provide power to the Hex Motor and Smart Robot Servo on the lower module, this batteries have a nominal voltage of 3.7V and a maximum voltage of 4.2V and a capacity of 3500 mAh each.

- **LDNIO 10000MAH POWER BANK**

We use a LDNIO 10000mAh Power Bank to supply Raspberry Pi and the Arduino Mega with energy. It has 1 USB ports, along with a Micro-USB port for charging, with an output voltage of 5V and 2.4A.

## OTHER

- **L298N MOTOR DRIVER**



Located in the middle module, it is used to manage the Hex Motor and provide both the Hex Motor and the Smart Robot Servo with energy.

### SPECIFICATIONS

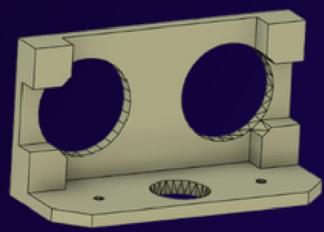
- Logic voltage: 5V.
- Operating Voltage: 5V-35V.
- Current consumption (Digital): 0 to 36mA.
- Current capacity: 2A (peaks up to 3A).
- Channels: 2 (supports 2 DC motors or 1 PAP motor)
- Maximum power: 25W.
- Weight: 30g.

## 3D PRINTED

When designing the robot, many times we needed very specific parts to allow the correct performance of the components that conform it, it is in these occasions that the 3D design proved to be very useful, as it allowed us to create and print these parts with high precision to optimize the design and maximize efficiency.

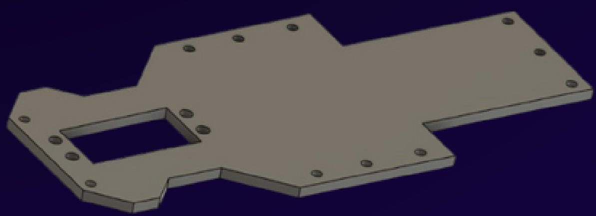

**SMART ROBOT SERVO SUPPORT BRACKET**

A piece that helps fasten the Servo Motor to the Main Chassis structure.



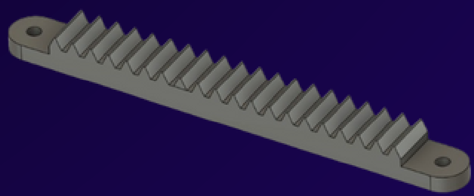
**HC-SR04 SUPPORT BRACKET**

Located in the "Bumper" piece, it serves as support for the HC-SR04 ultrasonic sensor, allowing it to function properly.



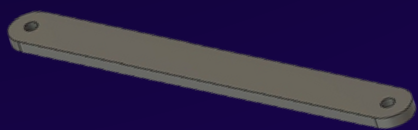
**MAIN CHASSIS**

It's the piece that contains and supports all the components found in the robot.



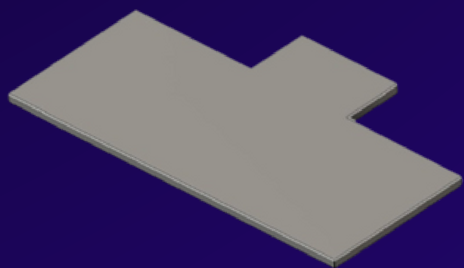
**CROSSING SYSTEM**

As it's name says, this part makes it possible to transmit the movement of the servomotor to the front wheels, thus allowing the robot to change direction.



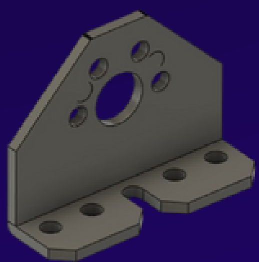
**CROSSING SYSTEM SUPPORT PIECE**

the function of this piece is to join or support the two wheels, in other words, it is the one that produces a support point for them, which allows them to rotate on a fixed axle.



**BUMPER**

Used as a extension of the Main Chassis piece, it is used to locate the HC-SR04 sensors a little further ahead so it can detect the walls more easily.



**HEX MOTOR SUPPORT BRACKET**

A piece that helps fasten the Hex Motor to the Main Chassis structure.

# SOFTWARE DESIGN



## CODING ENVIRONMENTS

### ARDUINO IDE (C++)

It is a cross-platform software written in the Java programming language. It is used to write and upload code to Arduino-compatible boards, as well as other boards from different manufacturers with the help of additional cores.

### PYTHON

It is a high-level interpreted programming language whose philosophy emphasizes the readability of its code. It is a multi-paradigm programming language, since it partially supports object orientation, imperative programming and, to a lesser extent, functional programming.

## IMAGE AND COLOR PROCESSING

### IMAGE CAPTURING

For color detection, the Spedal MF920P camera is used to capture a frame which is processed with Python thanks to the OpenCV image manipulation library.

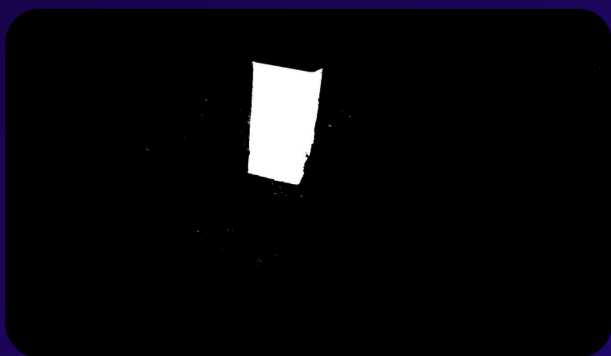
### CREATING RED AND GREEN MASKS

OpenCV reads the images or frames in BGR, therefore it is necessary to transform them to the HSV color space. This is because the parameters that we are going to give to identify the colors work in the HSV model (Hue, Saturation, Value). To do this we will use the function **cv2.cvtColor**, as first argument we will give the image to transform, and then **cv2.COLOR_BGR2HSV**, to indicate that we will transform from BGR to HSV.

The parameters used to identify red were:
**rojobajo1=np.array([0,150,20],np.uint8)**
**rojoalto1=np.array([3,255,255],np.uint8)**
**rojobajo2=np.array([175,100,20],np.uint8)**
**rojoalto2=np.array([179,255,255],np.uint8)**

and the parameters used for green were:
**verdebajo=np.array([35,100,20],np.uint8)**
**verdealto=np.array([80,255,255],np.uint8)}**

The result of these procedures is a binary image in which the whites are the colors detected with the parameters given above. This image will be used later to obtain the final image.



### FINAL IMAGE

To obtain the final image we will use the functions cv2.findContours and cv2.drawContours from the OpenCV library. The function cv2.findContours is used to identify the colors detected in the binary image that together with the function cv2.drawContours is used to mark the obstacles in the final image.

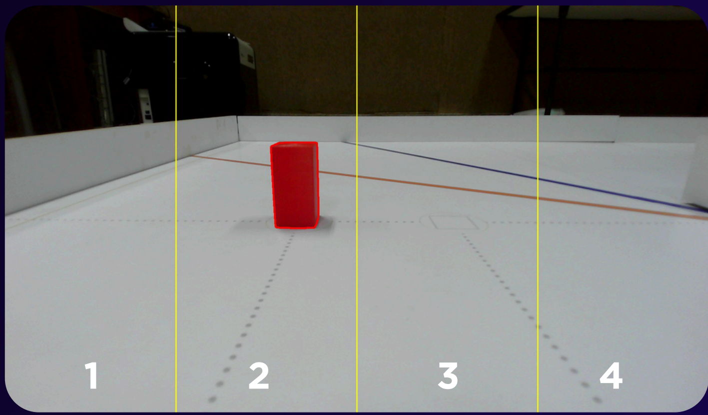
# SOFTWARE DESIGN



## DISTANCE FROM THE CONES

To identify how far the cones are from the robot, the image is divided into four positions: (see image)

The higher the position means that the obstacle is farther away from the robot, and with this information be able to calculate how much the robot has to move to avoid the cones more effectively.




## ROBOT MOVEMENT

### SMART ROBOT SERVO CONFIGURATION

The Smart Robot Servo is set on Pin 7 of the Arduino Mega and is configured with Servo.write function. The moveset of the Smart Robot Servo is defined by the distance given by the HC-SR04 sensors depending on whether the walls are near, far away or in a corner.

### ROBOT ORIENTATION

The orientation of the robot is defined by the first corner, if the corner is of the rigth side it means that the direction is clockwise and vice versa.

### ROUTE DETERMINATION

The determination of the routes is based on the color of the cone next to the robot. If the python code detects a color this information will be sent to the arduino and transformed into a left or right turn depending on the detected color.

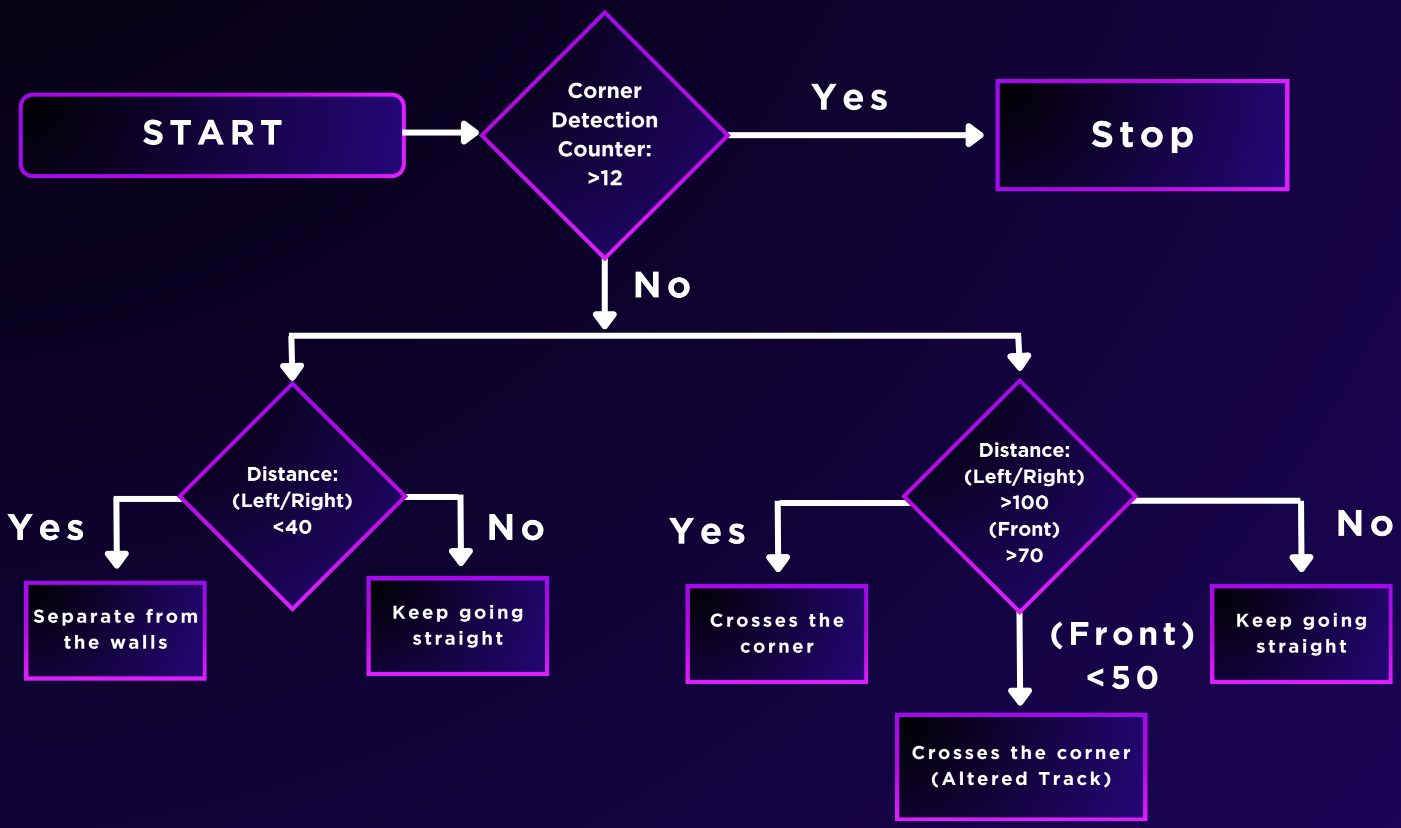
## DATA SENDING

First of all, we import the **smbus** library and then declare the *i2c identifier* that the Raspberry Pi 5 will have, additionally, the *i2c address* that the Raspberry Pi 5 has is placed, which is obtained thanks to a command executed in the terminal; then we create a function in charge of sending the information to the Arduino Mega 2560 R3 through a i2c connection that consist in three wires SDA (Serial Data Line), SCL (Serial Clock Line), and GND.

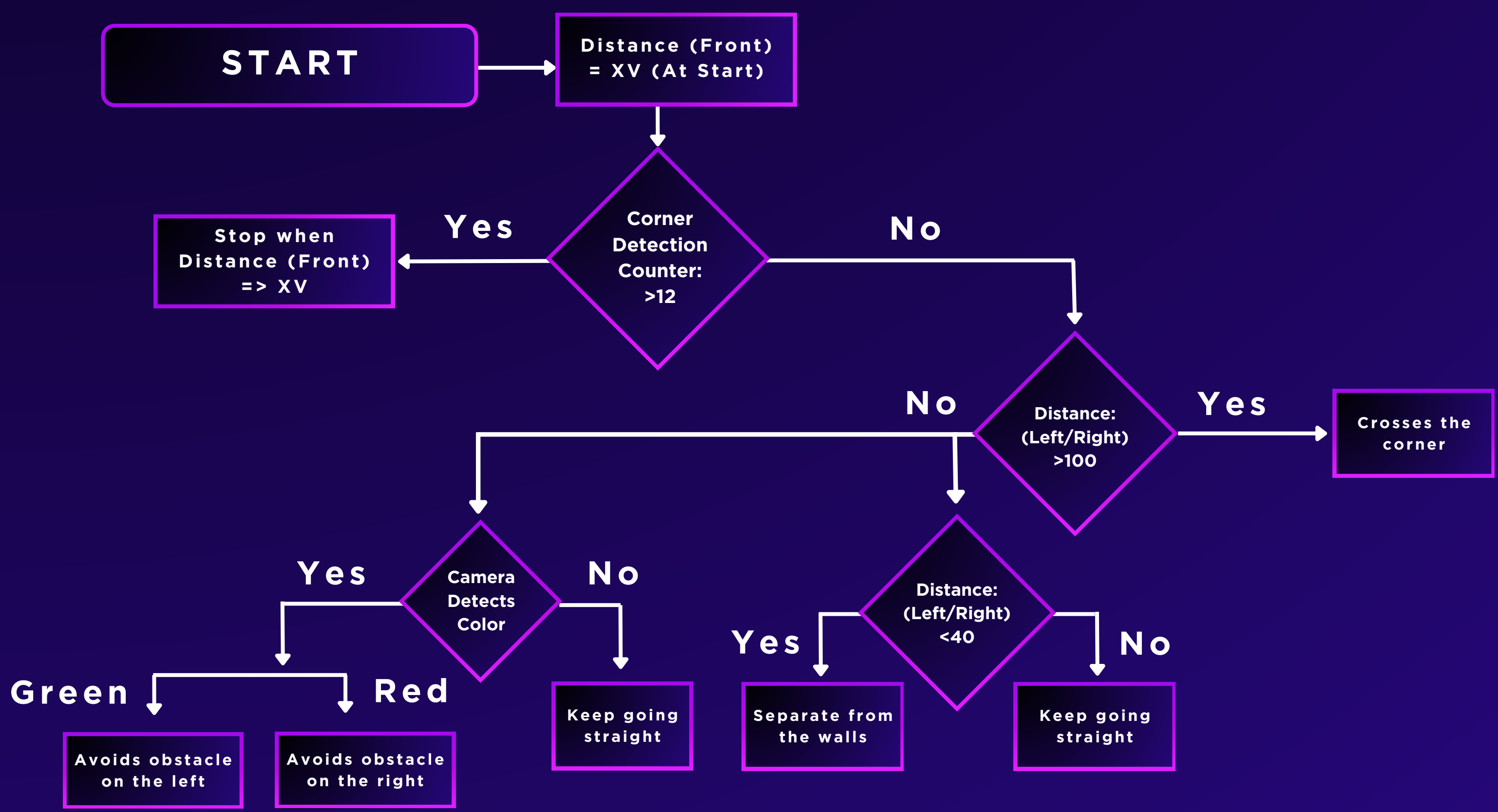
```
from smbus import SMBus
import cv2
import numpy as np
import serial
import time
b=""
w=0
clientAddr = 0x8
bus = SMBus(1)
def i2cWrite(msg):
  for c in msg:
    bus.write_byte(clientAddr, ord(c))
  return -1
```

# FLOW CHARTS

## ROBOT MOVEMENT FLOWCHART




## COLOR DETECTION FLOWCHART



# STRATEGIES

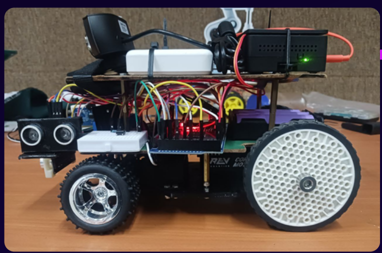


**PRACTICE SCORES**

**Open Testing: 30/30pts**
This is the 3-lap run without any obstacles.

**Obstacle Testing: 28/70pts**
This is the 3-lap run with the color cones. (Approximated)

## ROBOT REQUIREMENTS

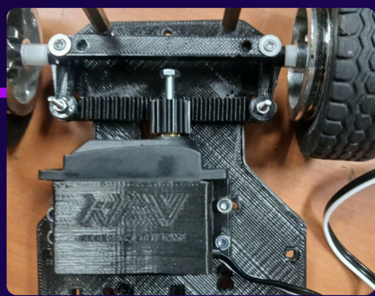


### EASY ACCESS TO ALL PARTS

The design is meant to be accesible in case of needing a repair or a minor adjustment in one of the modules.

### OPTIMIZED CROSSING SYSTEM

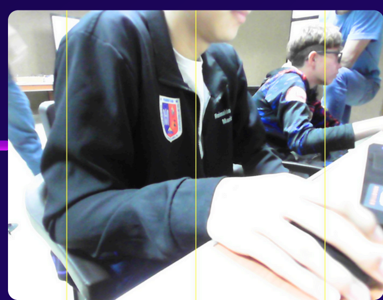
The design is meant to be accesible in case of needing a repair or a minor adjustment in one of the modules.



```
const int intervaloder=1900;
int esquinatrs=0;
void setup() {
 Serial.begin(115200);
 Wire.begin(I2C_DEVICE_ADDRESS);
  Wire.onReceive(receiveMsg);
  servo.attach(7);
  pinMode(IN4, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(triger, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(triger2, OUTPUT);
  pinMode(echo2, INPUT);
  pinMode(triger3, OUTPUT);
  pinMode(echo3, INPUT);
  pinMode(pulsador,INPUT);
  motor.setSpeed(120);
  motor.stop();
  servo.write(90);
```

### FASTER i2C DATA SENDING

We made the communication of the two boards by i2c, which gives us the advantage of having a higher sending speed.



### WIDE-ANGLE CAMERA

Use a wide-angle camera, which gives a greater range of vision and better detection of obstacles. This is complemented by high quality recording.

# THANKS FOR READING

NUESTRA SEÑORA DE
CHIQUINQUIRÁ
**HERMANOS MARISTAS**

**TEAM VIOLET**

WORLD ROBOT OLYMPIAD™ WRO ITALY 2024

WORLD ROBOT OLYMPIAD™ WRO VENEZUELA

EARTH ALLIES

FUTURE ENGINEERS