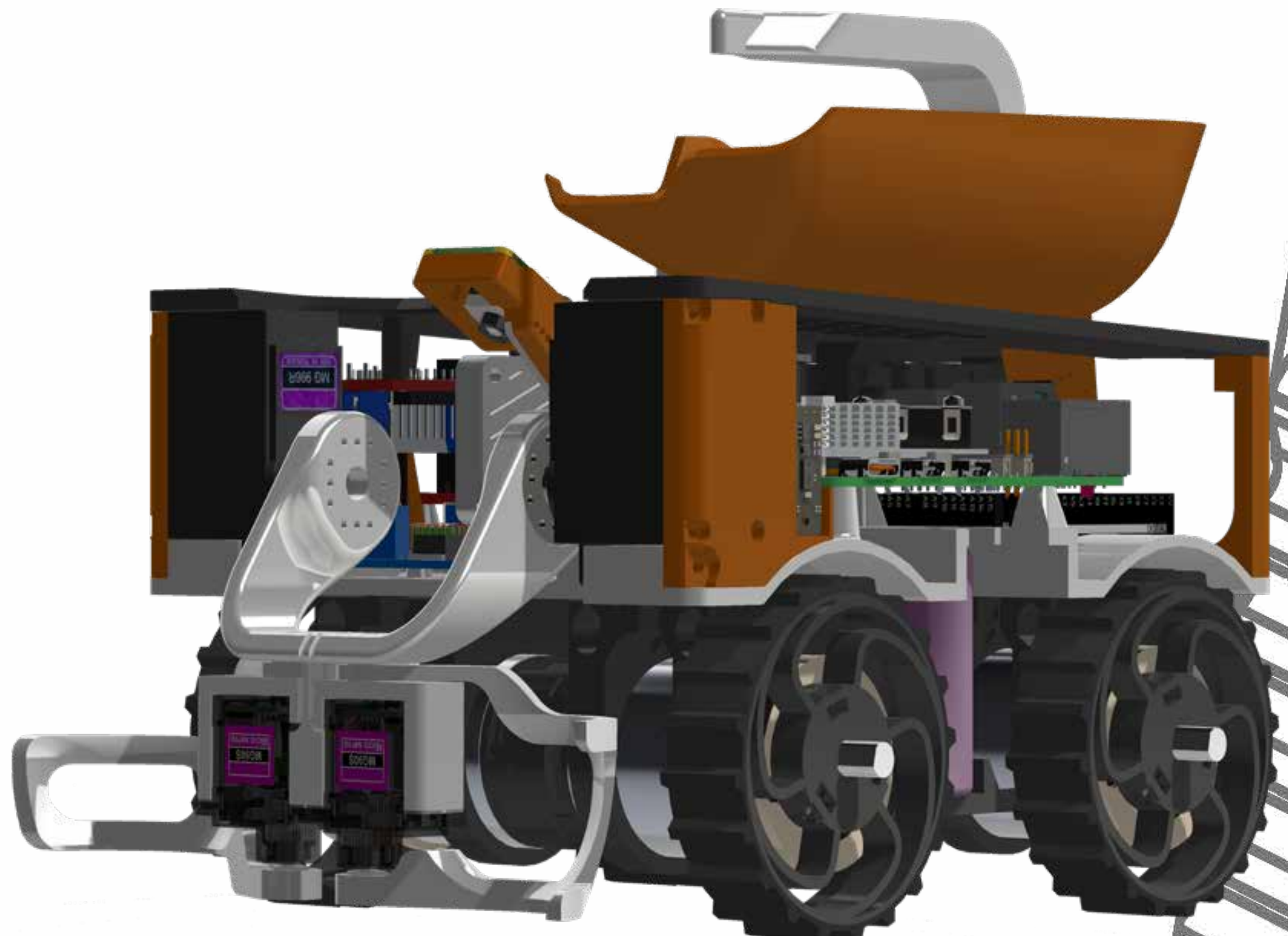
NETTUNO



GIOVANNI PEGORARO

CAPTAIN

His main roles are the ones of mechanical designer and software developer. He designed the structure of the robot and, after trials and optimizations, 3D printed it. Whereas, for the software development, he focused on the rescuing of the victims. He therefore worked on traditional coding as well as machine learning (training the model and deploying it).
He also covert parts of the firmware design.

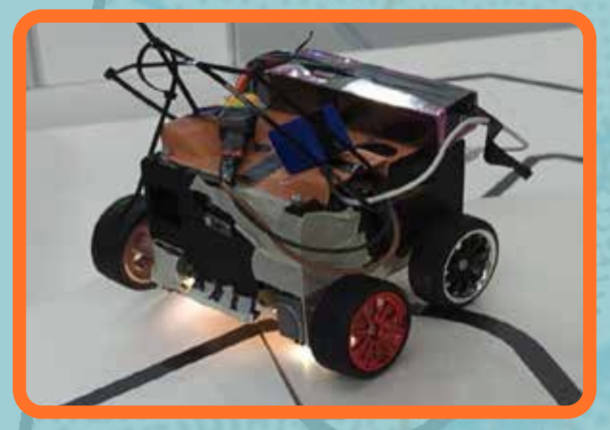
GIOVANNI MANZARDO

VICE - CAPTAIN

He is in charge of the software development for what concerns the line following part. This task consists of traditional computer vision and ranges from image acquisition and filtering to line detection. Furthermore he takes care of all the electronic aspects, from wiring to PCB designing.
He is firmware developer as well.

HISTORY

MKo
A piece of wood with 4 wheels



MK1
A piece of plastic with the same 4 wheels

MK2
A piece of plastic with other 4 wheels

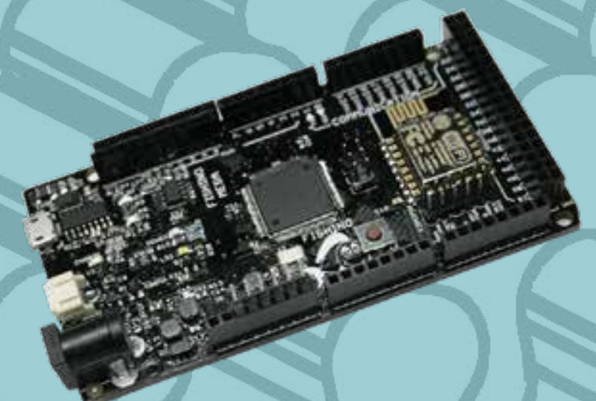
PREVIOUS COMPETITIONS

2023
- Robocup Rescue line - Regional Championship Vicenza
- Robocup Rescue line - National Championship Italia

2024
- 5th place in Robocup Rescue line - Regional Championship Vicenza

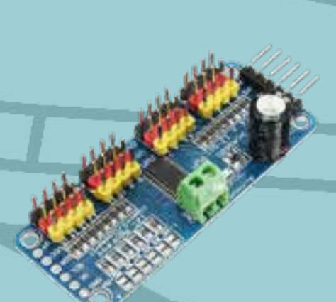
HARDWARE

RASPBERRY PI5
For line processing and victims finding



FISHINO MEGA
To control every sensor or motor

PICAMERA V3
Our main sensor

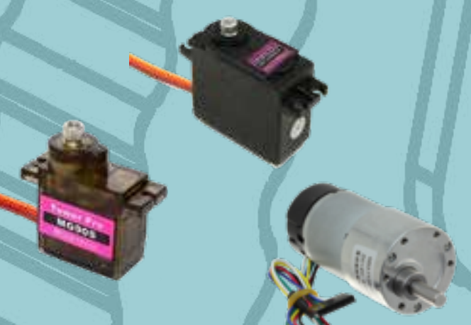


PCA 9685
A servo control device

BNO 055
A gyroscope

MINILIDAR
A tof sensor to detect distance

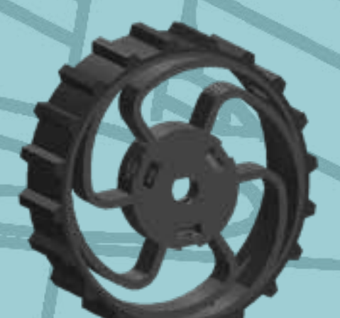
BATTERY
Our power supply



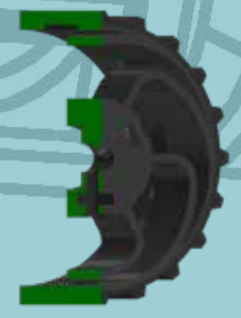
MOTORS
Our power train

The robot has been designed on Solidworks and 3D printed using PETG filament. This allowed a great flexibility while designing since we haven't been constrained by the use of off-the-shelf parts. The most distinctive mechanical features of our robot are the gripper and the wheels, both are covered in the section below.

INNOVATIVE SOLUTION

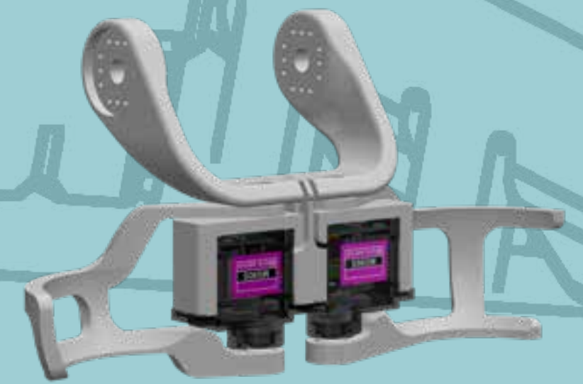


The wheels are inspired by the ones of a NASA rover, the shape is optimize in order to have a robot which is as narrow as possible. They are shock-absorbing as well.

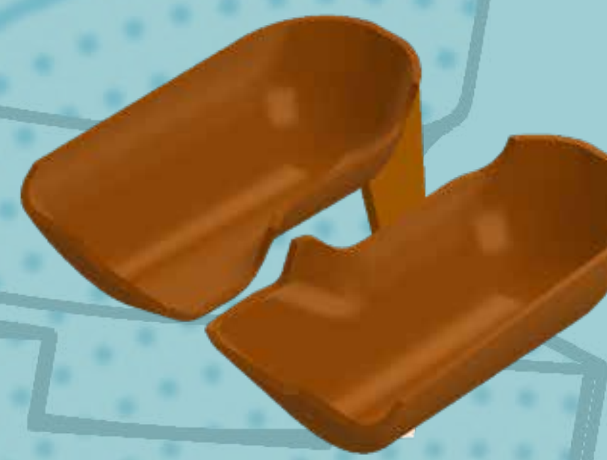


Cross section of a wheel

Computer-vision based drive systems are increasingly popular, nonetheless we think it's still innovative to use them because they require to overcome several difficulties at first. But then you can obtain a flawless line-following algorithm and superior performance in the evacuation room.



The gripper is an important part of the robot and its shape is its main feature. Our has been inspired by the design of the claws used in forestry-machinery, then it has been modified accordingly to the shape of the victims.



Another not-so-common feature is the dual basket setup. This allows us to carry more than one victim at a time, and even to differentiate the survivals from the deaths. That really simplifies the ball picking algorithm. It also reduces the risk of having to deal with a victim which is on the way of the robot while this is already rescuing another one.

ITT GIACOMO CHILESOTTI

RESCUE LINE

We choose the name "Nettuno" bacause it was the battle name of Giacomo Chilesotti, who was a partisian in Second World War. Our school is named after him and, since we believe in resistance values and want to keep them alive, we chose this name as well.

CONTACT US:

𝕏 @NettunoRobotica

@nettuno_robotica

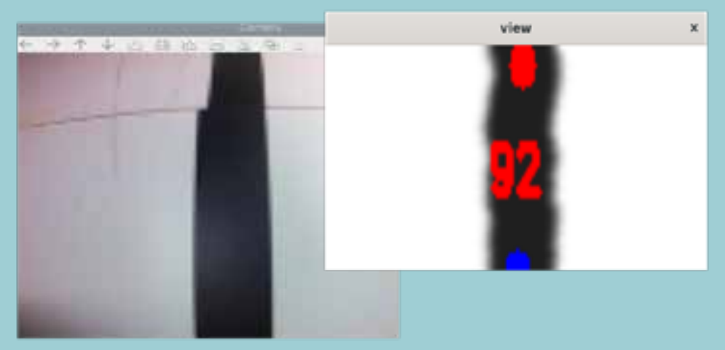
Nettuno

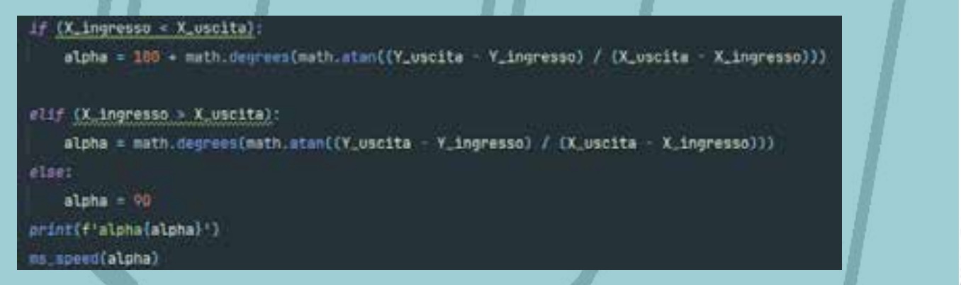
A picture of Giacomo Chilesotti

SOFTWARE

LINE

The line-following is entirely carried out using conventional computer-vision techniques, implemented using Python and the OpenCV library running on a Raspberry Pi 5.
The image is first acquired by the camera, then filtered to remove noise. At this point a threshold is applied to isolate the black pixels of the line.
The robot then tracks the end of the line and in this way it knows when to turn.

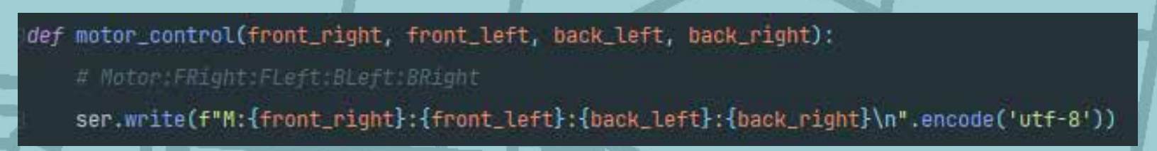




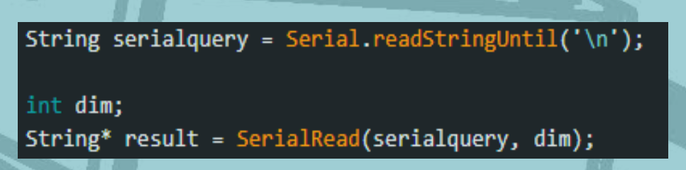
```
if (X_ingresso < X_uscita):
    alpha = 180 + math.degrees(math.atan((Y_uscita - Y_ingresso) / (X_uscita - X_ingresso)))

elif (X_ingresso > X_uscita):
    alpha = math.degrees(math.atan((Y_uscita - Y_ingresso) / (X_uscita - X_ingresso)))
else:
    alpha = 90
print(f'alpha{alpha}')
ms_speed(alpha)
```

Example of Input and Output Tracking

Low-level calculations are carried out by the Fishino Mega (similar to an Arduino Mega), which controls the motors through the drivers.
It also read data from the sensor and sends them to the Raspberry, so that this knows when there is an obstacle.

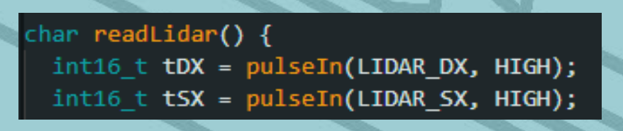


```
def motor_control(front_right, front_left, back_left, back_right):
    # Motor:FRight:FLeft:BLeft:BRight
    ser.write(f"M:{front_right}:{front_left}:{back_left}:{back_right}\n".encode('utf-8'))
```



```
String serialquery = Serial.readStringUntil('\n');

int dim;
String* result = SerialRead(serialquery, dim);
```

Example of Serial communication between Raspberry and Arduino

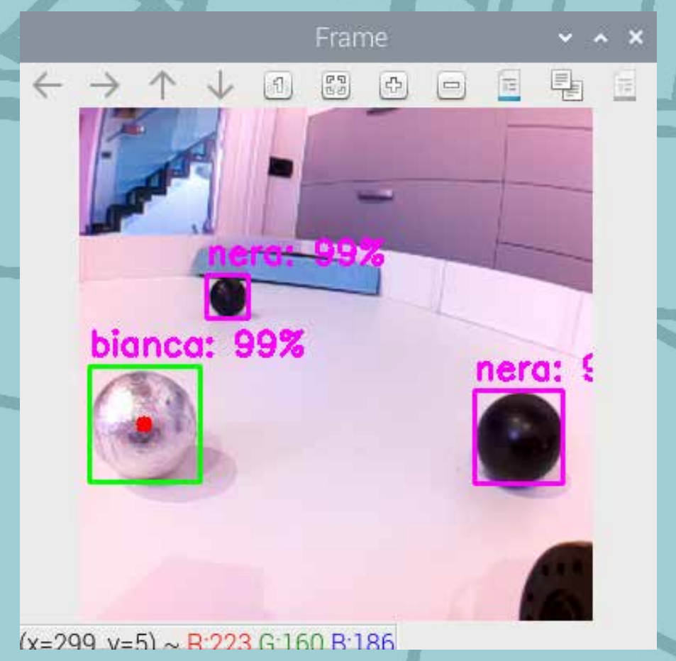


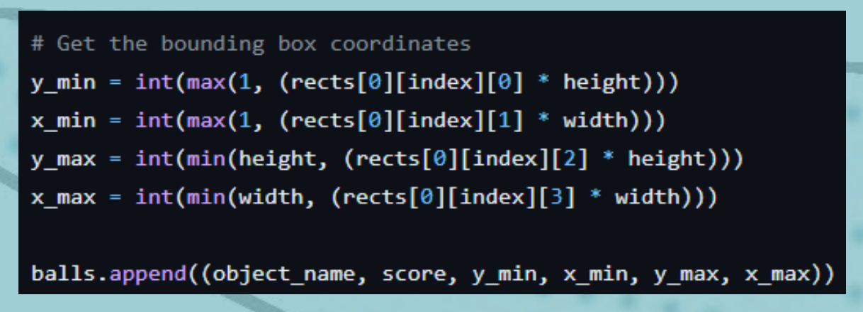
```
char readLidar() {
  int16_t tDX = pulseIn(LIDAR_DX, HIGH);
  int16_t tSX = pulseIn(LIDAR_SX, HIGH);
```

Example of Obstacle detection

EVACUATION ZONE

On the other hand the ball-picking algorithm relies on both traditional computer vision and machine learning. A TensorFlow object detection algorithm is in charge of finding the victims.
The script then tracks them while the robot gets closer.
When a ball is low enough in the camera frame,
the robot stops and picks it. It then goes to the green and red baskets.
In this part standard OpenCV algorithms are used.





```
# Get the bounding box coordinates
y_min = int(max(1, (rects[0][index][0] * height)))
x_min = int(max(1, (rects[0][index][1] * width)))
y_max = int(min(height, (rects[0][index][2] * height)))
x_max = int(min(width, (rects[0][index][3] * width)))

balls.append((object_name, score, y_min, x_min, y_max, x_max))
```

Here we can see how the data related to objects classes and bounding boxes are retrieved from the output of the inference (the process which actually runs the model on the input image which we provide).

TEAM'S PICTURE

Us and other Chilesotti Team after National Championship 2023

Us and other Chilesotti Team after Regional Championship 2024

Us after this year Regional Championship