RoboCupJunior Rescue Line 2023

Team Description Paper

HACKATRONICI

Abstract

Our robot is a Lego Mindstorm EV3 31313 with Education expansion kit. The pair of color sensors, together with a course correction algorithm allows a very performant and precise line-following.
An additional sensor in reflected light mode has been used for the detection of the crucial points of the rescue line such as the beginning of an intersection or the entry into the evacuation zone. The use of this additional sensor has greatly improved the reaction to the occurrence of such events.
An ultrasonic sensor was used to detect objects in front of the robot and at the sides. This was used in obstacle avoidance and evacuation zone scenarios.
In the evacuation zone, the reflected light sensor changes to color modes and tilts to 90 degrees to detect lvl. two evacuation points.
In the realization the aim was maximizing the performance over the time of the rescue line, while maintaining excellent accuracy. This allowed more time availability in the evacuation zone scenario.

1. Introduction

a. Team

Our Group consists of:
Alessandro Chiarulli: 4th year, robot assembly. Rescue line programming.
Gabriele Montrone: 3rd year, green detection calibration. Programming.
Giuseppe Clemente: 3rd year. Evacuation zone programming.
Mario Recchia: 3rd year, evacuation point, entry and exit evacuation zone. Programming avoiding obstacle.

Past experiences of team members:

- on the podium for the Olympics of problem solving
- digital education hackathon global ambassador
- registered in the “albo nazionale delle eccellenze”
- Arduino projects: intelligent vase, prototype of robot for rescue line, control unit lights for dj set
- software development: web front end applications
- certifications obtained: astropy
- Certified courses: Cisco, Cambridge, EiPass

2. Project Planning

a. Overall Project Plan

The objective in the design phase was to maximize the points in the follow-line phase, covering all possible scenarios.

After completing the part of the line, we focused all our attention on the evacuation zone algorithm. The process of development of the evacuation zone has been long and tortuous, occupying a lot of time in the general development.

The total project duration was around 15 days, considering that we are at the end of the school year and there were many commitments.

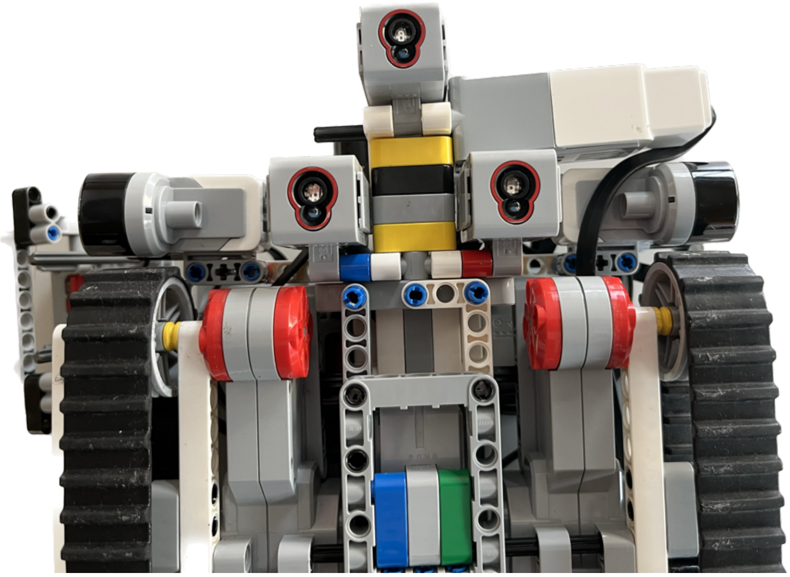
The activities were carried out in parallel, indicatively based on the tasks assigned.

b. Integration Plan

For the line we used a system with three sensors, two at the level of the engines and one ahead of the two. The two sensors behind recognize the line in the center and any green intersection, while the sensor in front is in reflected light mode to recognize the reflective sheet at the entrance of the evacuation zone and in the follow-line to recognize intersections.

For the obstacle and walls of the evacuation zone we used four ultrasonic sensors. To release the Rescue Kit, we used a medium engine, powerful enough to support its weight. To catch the victims, we used two large motors placed on top of the robot structure, carrying a tube that locks the balls.

Since we needed other sensors to cover all possible scenarios in the path, we decided to mount a second EV3 brick on the robot, in communication with the first via bluetooth. To the first brick are connected the three sensors for the line-following and the gyroscope, placed under the structure at the center of the robot. On the second brick we connected the four ultrasonic sensors to take all the measurements we need.



In addition to the hardware side, the integration has also been managed at the software level. To facilitate software integration, we used GitHub for versioning and code revision management. Besides, the source has been divided into modules to allow independent development after defining the interfaces of the modules. Splitting the program modules into multiple files allowed us to optimize the development work of the robot and work on multiple parts of the project simultaneously.

```
mhackrobocup
│
│   .gitignore
│   README.md
│
├───EV3_Python
    │
    ├──Caricare
    │     Stanza.py
    │     rescue_line_main.py
    │     rescue_line_functions.py
    │     rerscue_line_setup.py
    │     Slave_server.py
    │     client_functions.py
    │     stanza_guadagna_dentro.py
    │     stanza_lascia_pallina.py
    │     stanza_main.py
    │     stanza_prendi_palle.py
```

3. Hardware

Color Sensors

Color sensors can recognize seven colors: black, blue, green, yellow, red, white, brown.
For the line follower, an ad hoc calibration was necessary to manage the cases of overlap between black and white, a situation that for the sensor is part of the "blue" case.

Reflected Light Sensor

The reflected light sensor, on the other hand, works by emitting a red light and measuring the intensity with which it is reflected. It works on a scale from 0 to 100, where 0 is very dark and 100 is very bright. In our case, if the reflected light is greater than 95 it means that it has found the evacuation zone, while if it is less than 10 it has found the black line. In the range between 50 and 70 is the color white.

Once it enters the evacuation zone, the robot changes the order of the sensors, making the one in front enter color mode and those behind enter reflected light mode. The front sensor is also tilted by 90 degrees. In this way we can recognize the evacuation points in which to store the victims.

Ultrasonic Sensor

The ultrasonic sensor works by emitting sound waves and measuring the return echo. Depending on the time the sound takes to return to the sensor it’s able to measure the distance with a fairly high level of accuracy.

Gyroscope

The Gyro Sensor is a digital sensor that detects rotational motion on a single axis. If you rotate the Gyro Sensor in the direction of the arrows on the case of the sensor, the sensor can detect the rate of rotation in degrees per second. (The sensor can measure a maximum rate of spin of 440 degrees per second.) You can then use the rotation rate to detect, for example, when a part of your robot is turning, or when your robot is falling over.

In addition, the Gyro Sensor keeps track of the total rotation angle in degrees. You can use this rotation angle to detect, for example, how far your robot has turned. This feature means you are able to program turns (on the axis the Gyro Sensor is measuring) with an accuracy of +/- 3 degrees for a 90-degree turn.

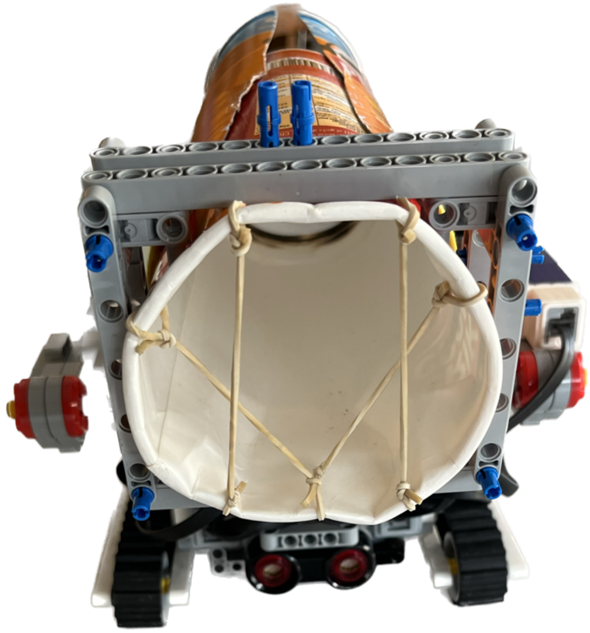
a. Mechanical Design and Manufacturing

The robot structure is made with the pieces of the EV3 kit. We tried to create a structure as efficient as possible in terms of weight, considering the ramps and the additional weight of the rescue kit and the victims holder. For this reason, the number of components of the robot is reduced to a minimum, looking for a compromise between weight and solidity.

The large motors used for movement are the most powerful of the EV3 kit, with an integrated rotation sensor that can provide accuracy to the degree. They are powerful enough to support the weight of the robot in the path and on the ramps, considering also the cases when there is a speed-bump on the ramp.

For the Rescue Kit we used the medium engine, as less power is required to store it in the safe area. This also has a built-in rotation sensor, so you can make it rotate 100 degrees several times to be sure to release the rescue kit from the basket.

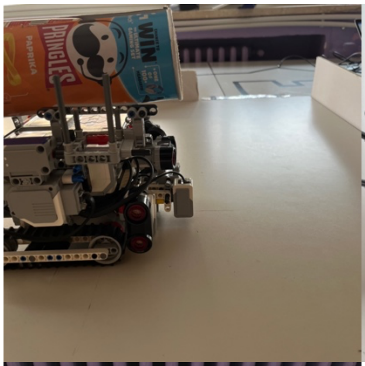
For the rescue of the victims we used an extremely simple mechanism. The structure is formed by a 25 cm high tube, with elastics on the bottom. When a ball is recognized, the tube quickly lowers on it, widening the elastic bands and fitting the ball inside. The tube will rise to prevent the ball from falling on the other side. When the right evacuation point is recognized, the robot will lift the tube completely and drop the ball from the opposite side.

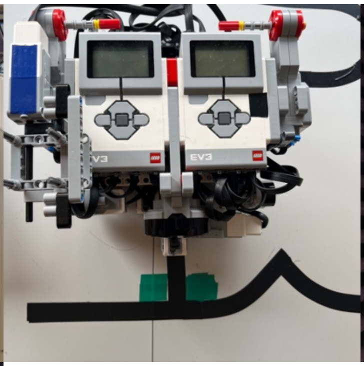


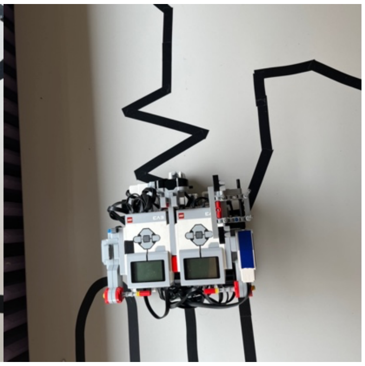
Test procedures

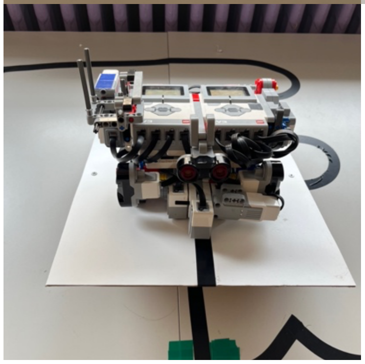
From the mechanical point of view, the following tests were planned:

- ramp progress (power/weight test, track friction)
- rescue kit release (release procedure)
- balls-collecting









b. Electronic Design and Manufacturing

Control of the onboard electronics is handled by the main brick of the EV3 31313. It has 8 ports available, 4 input for sensor control and 4 output for motors. If the robot is connected to the computer, we can have real time values read by the sensors, useful for calibration and color control.
The robot is powered by its native battery pack.

Test procedures
For each scenario to be managed (line, turn, bumper, fixed and mobile ramp, obstacle, evacuation zone) an independent test procedure has been provided, activating from the software only the part of code involved.
After testing the individual procedures, we went to the **integration test**, where a complete scenario was tested several times, activating this time all the software developed.

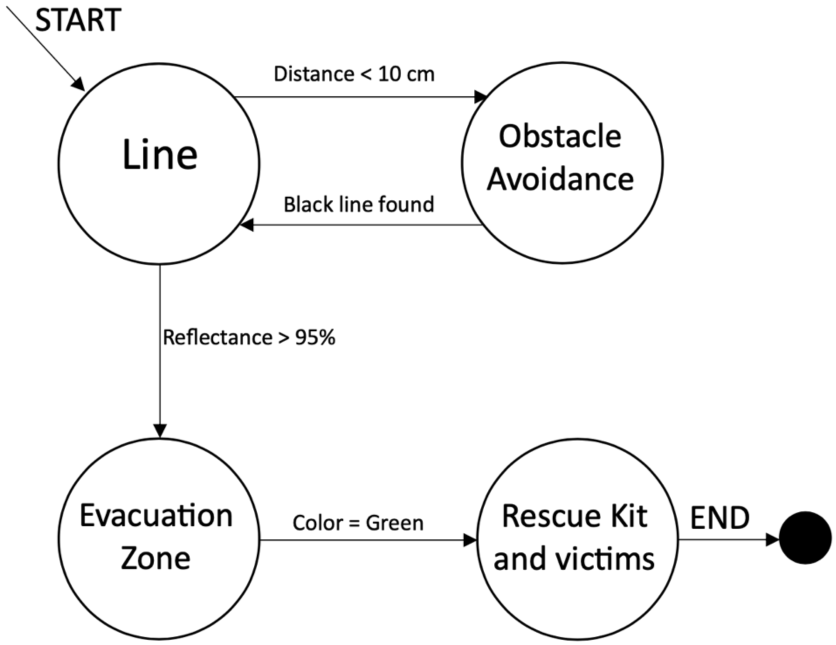
4. Software

Per il software abbiamo deciso di non usare il linguaggio a blocchi di EV3, in quanto molto limitante a livello di funzioni in generale. Abbiamo usato MicroPython, una versione "semplificata" di Python, in cui sono state eliminate tutte le librerie che non vengono usate nel robot. La libreria principale che abbiamo usato per programmare il robot è stata PyBricks, con una serie di funzioni aggiuntive per garantire la compatibilità con tutti i sensori e attuatori.

Per la comunicazione tra i due brick abbiamo usato il bluetooth. Abbiamo scelto un brick come server, quello che ha tutti i sensori ad ultrasuoni, che riceve i comandi, e uno come client che manda i comandi al server, su cui abbiamo montato i sensori per il seguilinea e il giroscopio. Il programma principale è eseguito sul client, che richiede al server i dati del sensore ad ultrasuoni quando sono necessari.

a. General software architecture

The general structure of our code is as follows:



If the front sensor detects reflected light greater than 95%, it enters evacuation zone mode, otherwise the line-following is performed. If the ultrasonic sensor detects a distance smaller than 10 cm, it enters the obstacle-avoidance mode. If the robot has entered Evacuation Zone mode the first step will be to gain the center and understand where the evacuation points are located. He will leave the rescue kit and start a scan to look for the balls. Once identified the ball it will catch it with the tube and bring it into the right triangle.

b. Innovative solutions

The functionality we are most proud of is the line-follower. Finding the right compromise between robustness, precision and speed has been a challenge. For tuning the parameters and to keep track of the results for each new configuration we used a spreadsheet to write down values and evidence.

The use of a simulator as an open Roberta proved to be very useful, it allowed us to start with preliminary tests when the robot was unavailable (during its construction, modification and structural improvement).

Splitting into modules and integrating with GitHub was also a winning move for teamwork.

Another innovative solution was to install a different operating system on the EV3 brick, in order to make the most of the robot with the Python programming language. Finding all the right functions for the robot was not easy, it took many hours of testing, but the results obtained are much better than what we could have done only with the native block language, making it also extremely readable.

The union between mathematics, physics and computer science was fundamental for the development of a correct algorithm for evacuation zone. Given the high inaccuracy of the cone of view of Lego ultrasonic sensors, it was not easy to clean up the signal and make it usable to detect spikes in the measurements corresponding to the victims. It was necessary to apply all our knowledge in the analysis of the collected data to create an accurate algorithm enough to cover all possible cases.

5. Performance evaluation

We found that the robot was very precise in following the line. Almost all cases of crossing are covered and, unless extremely unlucky cases, it is always able to follow the line without getting lost. For the evacuation zone we have great room for improvement. Due to the great inaccuracy of the sensors it was not easy to create a perfect algorithm in a short time. But given the amount of work done to refine the code we are very happy with the result, hoping for a future opportunity to use more advanced and accurate sensors.

6. Conclusion

To participate in this competition, we tried to use 100% of the potential of our robot, using all the features also present by the software. We prepared the robot to deal with all possible cases present in the paths, with particular attention to the line-follower. Despite the limited possibilities from the Lego kit, we tried to create a robot capable of performing the main purpose of the category.

References

[1] EV3 Site - Tutorial

[2] Lego Mindstorm PyBricks documentation

[3] Lego Mindstorm EV3 Hardware Developer Kit