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 entrance to a crossing 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. This was used in obstacle avoidance and evacuation zone scenarios.
In the evacuation zone, the reflected light sensor changes to color modes to detect 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, calibration detection, evacuation point, entry and exit evacuation zone. Programming bypassing 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.

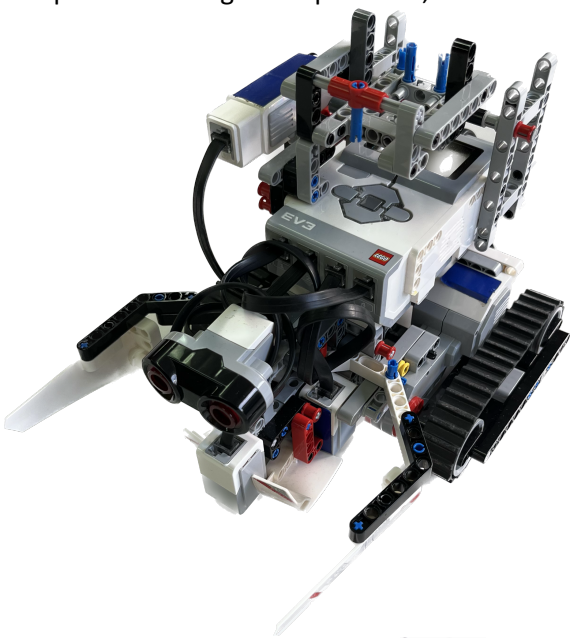
In the initial design phase, we decided to deal with the evacuation zone. A simple algorithm has been implemented for this phase. A more complex strategy was implemented as a prototype, but we did not consider it stable enough, so it was discarded in favor of the simpler algorithm.

Milestone

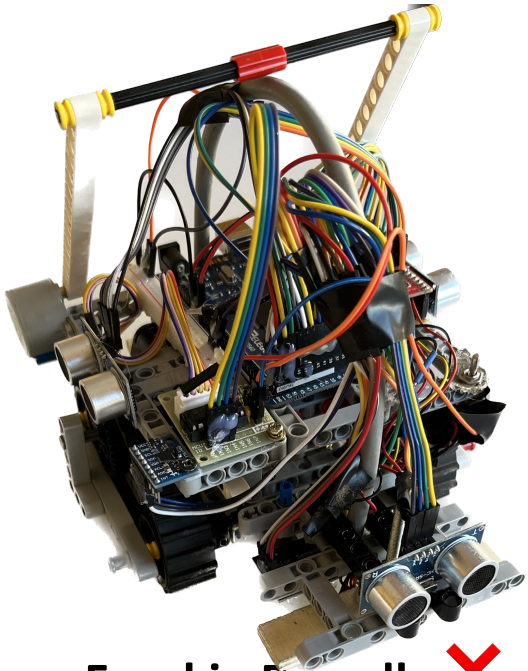
- Total project duration: 30 days, divided into:
  - robot realization: 4 days, in parallel with the other activities
  - line-follower programming: 3 days
  - programming: 5 days, after the line follow
  - Ball detection programming: 3 days
  - Detection evacuation point: 1 day
  - Evacuation zone output exit: 1 day
  - Rescue kit programming: 1 day
  - Test and refinement line: 5 days
  - Test and refinement speed bump: 1 day
  - Test and refinement ramps: 1 day
  - Test and refinement tilting ramp: 1 day
  - Test and refinement evacuation zone: 8 days

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

Two prototypes were developed in parallel, one using Arduino, the other an EV3 Kit. In the Test phase we opted for the EV3 solution because it proved to be more stable. The Arduino solution, although more complex and with greater potential, remained in a prototype phase, not stable to face the competition.



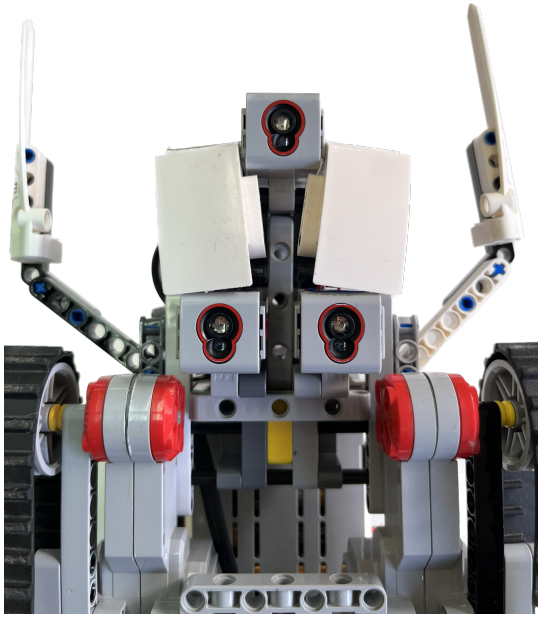
Gundam ✓



Frankie Brundle ✗

b. Integration Plan

For the line we used a system with three sensors, two at the level of the engines and one later. The two sensors behind recognize the line in the center and any green intesection, 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 an ultrasonic sensor. To release the Rescue Kit, we used a medium engine, powerful enough to support its weight.



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.

```
mhackrobocup
│
│    .gitignore
│    README.md
│
├────definitivo
│         040_sensor_distanza.ino
│         050_momimento.ino
│         060_aggira_ostacolo.ino
│         070_evacuation.ino
│         definitivo.ino
│         definizioni.h
│
└────ev3
          GaraVicenza.lmsp
          Stanza.lmsp
```

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.

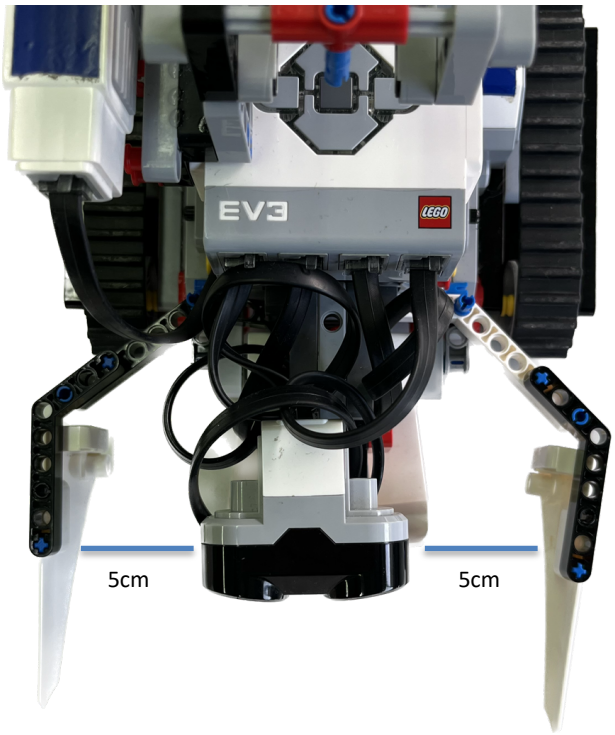
Once it enters the evacuation zone, the robot changes the order of the sensors, making the one in front enter color mode and those behind in reflected light mode. In this way we can recognize the safe areas in which to store the victims.

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 is able to measure the distance with a fairly high degree of accuracy.

a. Mechanical Design and Manufacturing

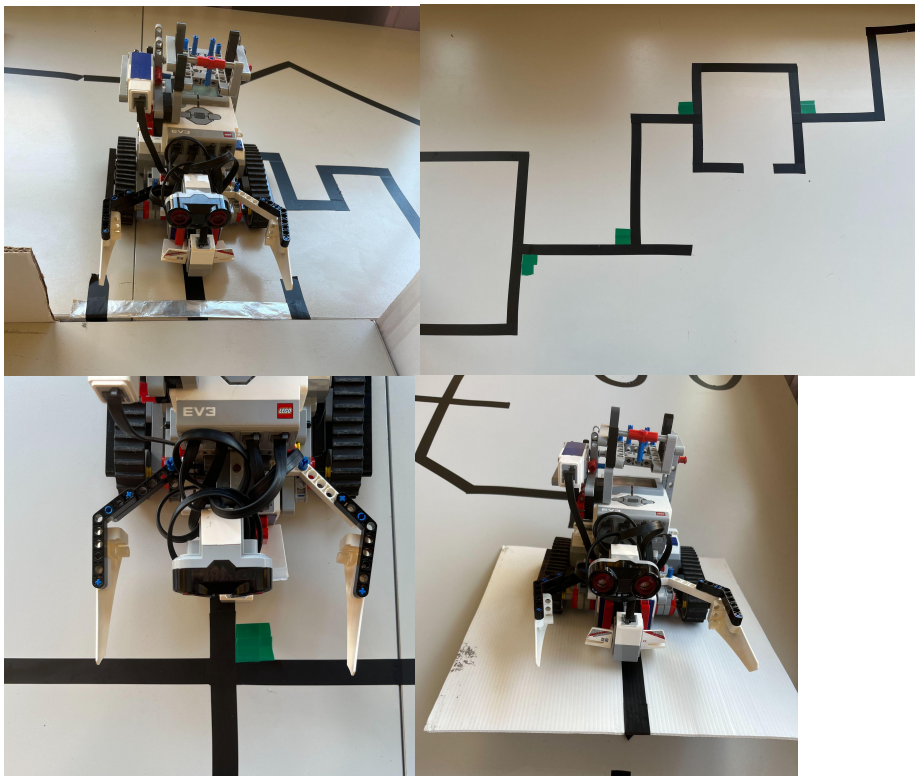
The robot structure is made with the pieces of the EV3 kit. We tried to create a structure as efficient as possible in terms of weight, considering the ramps and the additional weight of the rescue kit. For this reason, the number of components of the robot is reduced to a minimum, looking for a compromise 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 135 degrees several times to be sure to release the rescue kit from the basket.
For the rescue we used an extremely simple mechanism: two arms on the sides of the ultrasonic sensor, with a space from the sensors of just over 5cm, to accommodate the balls and not make them escape.
Once the green or red triangle has been identified in the Evacuation Zone, the robot moves away suddenly, to unhook the balls, makes a curve of 180 and then goes back to put down the Rescue Kit. The robot is able to move on.



Test procedures

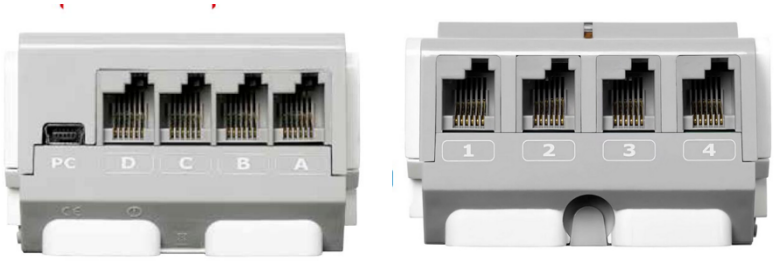
From a mechanical point of view, the following tests were planned:

- 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 6 1.5V AA batteries.



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.

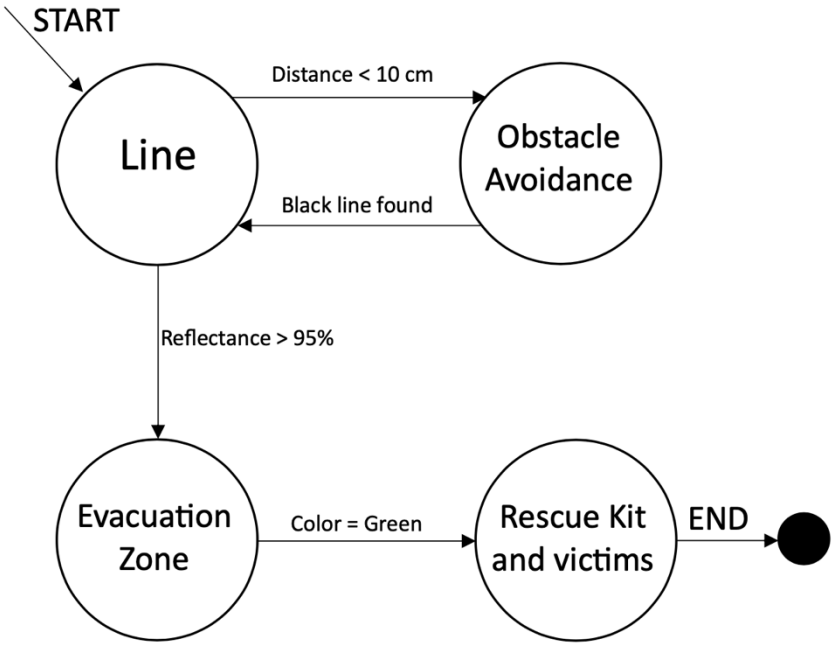
4. Software

For the robot code we used the integrated development environment of the EV3, with simple block programming but including all the blocks needed to make a complete path.

GitHub was used to facilitate collaborative development, track the various versions developed over time, label the tested versions and manage source conflicts. Each team member worked independently on their module. The final software was then finalized by integrating the various sources.

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 room mode, first check which side the entrance is on, then start working on "snowplow", covering all the available surface until it finds the green triangle. When he finds it leaves the rescue kit.

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, in particular 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.

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 because of the simple solution adopted. Not having a precise mechanism to collect the balls, different situations could cause problems to the robot. Generally, if the balls are well distributed in the room, there is a good chance that the robot will be able to complete its tasks. The rescue kit is released correctly with a good chance of success.

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] Open Roberta