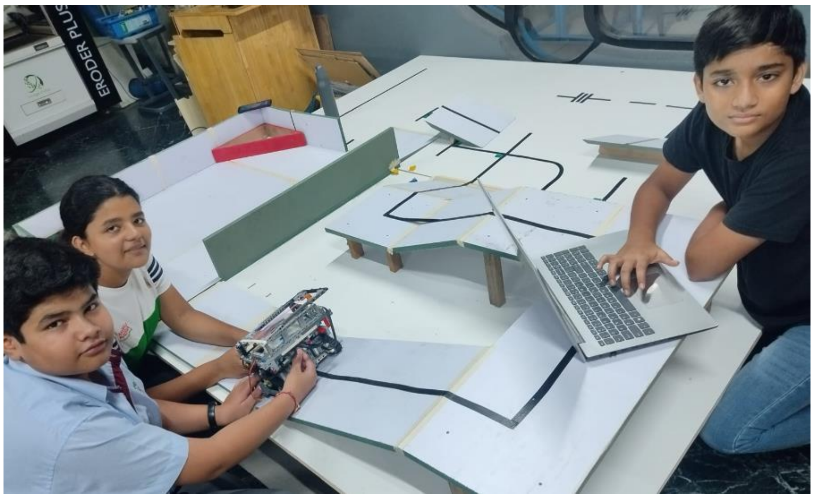
RoboCup Asia Pacific - Rescue Line 2024

Team Description Paper

League Name:	RCJ RESCUE LINE
Age Group:	Secondary
Team Name:	CODEX
Team Website:	Codex
Participants and Technical Roles	AI strategy designer and Programmer - Kashvi Khajanchi Hardware designer- Swayambhav Siddharth Madhur Programmer - Navya Singla
Team Photo	
Mentor Name:	Mr. Amanurrehman
Institution:	Manav Rachna International School, Charmwood
Region:	INDIA
Contact Email:	amanurrehman.mriscw@mris.edu.in

INTENT

Our team, CODEX, is dedicated to advancing the frontiers of robotics and collaboration by participating in **RoboCup Junior Asia Pacific 2024** for the category Rescue line. We are a group of three students from diverse backgrounds, united by our passion for **problem-solving through the integration of AI and programming.** While participating in the rescue line mission our intent was to innovate and construct a robot capable of analysing complex situations and performing rescue operations with **precision, reliability and ensuring the safety of the target keeping the sustainability aspect at the heart of our innovation.** For sustainability we strictly use only reusable plastic and have equipped our lab with e-waste collection box. We keep raising **awareness and conduct e-waste and plastic collection drives to reduce our carbon footprints.** We have a tie up with NAMO e-waste Management Ltd. for responsible disposal of e-waste. We believe in the positive outcome of the power of teamwork, collaboration, continuous learning, and inspiring one another to achieve our objectives. We are committed to excellence and a spirit of competition and hope to exhibit our efforts while making a meaningful impact in not only the world of robotics but its immense uses beyond. Together, we are excited to tackle the challenges ahead and contribute to the future of Rescue Robotics!

Benches made for community by Plastic collection drives June 2023- August 2024. 400 Kgs Plastic collected	E-Waste collection Bin placed in our Robotics lab

[illegible]	[illegible]	Kashvi	[illegible]
Amanur Rehman	**Navya Singla**	**Kashvi Khajanchi**	**Swayambhav S. Madhur**
(Mentor)	**(Team Member)**	**(Team Member)**	**(Team Member)**

INDEX

ABSTRACT

Our team aims to develop a robot that operates with exceptional targeted precision, rescuing victims and overcoming all obstacles in accordance with the Rescue Line rules. What makes our robot unique is the multi-layered approach we've created by blending OpenMV and EV3 technologies. Over the months, we have tested various cameras and algorithms, but the OpenMV camera, in conjunction with EV3, proved to be the most effective solution for us. The design includes colour sensors for line following, an EV3 ultrasonic sensor at the front for obstacle avoidance, two large motors for movement and steering, a medium motor for the claw mechanism to lift and place the ball on a slope, and another medium motor for opening the tailgate to drop the balls in the evacuation zone. The servo motor and the Lever in OpenMV are used to find the opening for the exit by sensing the boundaries.

1) INTRODUCTION

Our team comprises of three members - Navya Singla, Kashvi Khajanchi and Swayambhav Siddharth Madhur bringing together a blend of unique perspectives and experiences, all united by a shared goal: to collaborate and deliver the best possible outcome. We started our journey for the RoboCup Junior Rescue line in our school's robotics lab about a year and a half ago. During our participation in RoboCup Regionals we got an insight into the level of precision required to secure the target safely, pushing us to refine our skills.

With each of us bringing our expertise to the table, we divided tasks based on our strengths, yet our roles constantly overlapped as we worked as a unified team. Whether testing, refining the model setup, or making necessary adjustments in our robotics lab, we supported each other in every aspect. Kashvi focused on AI strategy design and programming, Navya took the lead as the programmer, and Swayambhav handled the hardware design. Together, we also managed our digital presence through our **website**, **Instagram** and **Youtube channel** ensuring our project's journey is shared with the world.

This competition has not only expanded our technical knowledge but also sharpened essential 21st-century skills like collaboration, problem-solving, and critical thinking.

2) PROJECT PLANNING

In our journey of about a year and a half of preparation and working on the project, our objective has been creating a cost effective yet efficient robot. With each trial we carefully analysed each aspect of the robot and reworked to get the best fit as per the requirements. We identified potential options for integrating the best possible available software and hardware carefully evaluating each to find the optimal solution.

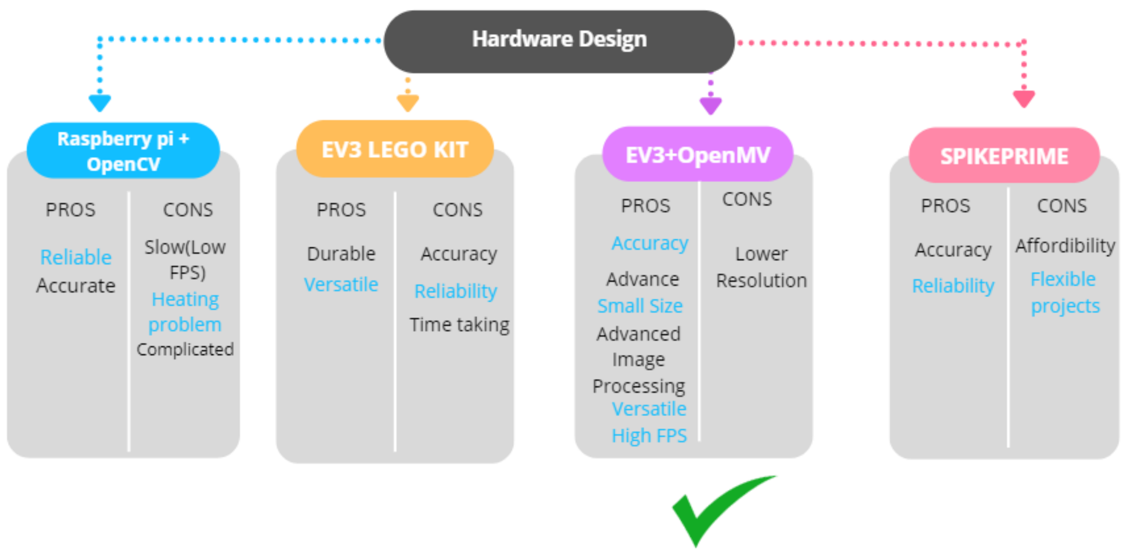


Figure 2a – Evaluation of Hardware Design Options

After multiple trials and testing, the model we decided to use was EV3 with OpenMV as it was the most suitable option that provided robot with accuracy and precision required to complete the line following and rescue mission. We planned to completely use the EV3 platform for line following, and obstacle avoidance. Our robot uses colour sensors to navigate around the map and will avoid obstacles with the help of an ultrasonic sensor. For the rescue mission, we have integrated OpenMV with EV3.

We initiated the process by looking at our previous runs and derived ideas from our observations based on them.

After rigorous brainstorming and rounds of discussions we started our project by creating a base model of the robot using EV3. The design was gradually improved with frequent experimentation and analysis followed by a series of changes made to finally set up a model capable of line following, avoiding obstacles, sensing turns and intersections.

Over the course of two months, we meticulously developed our algorithm and conducted multiple test runs. This process allowed us to make significant adjustments to the construction and coding of our robot, ensuring it met our precise requirements.

For the rescue mission, we integrated an OpenMV camera in our robot. The camera recognizes the circular objects to be rescued. The experimentation continued until a dependable hardware and software that can run without causing any deviations was developed.

We developed a strict schedule and deadlines to work on each segment of the process with the major focus on practise for the final set of obstacles. The design despite all this was potentially revised to get better than the previous models each time and ensure reliability and efficiency.

2a. Project Timeline

Timeline	Work Done
Mid August 2023 -	Building a robot that can do Line Following
End of August 2023 -	Testing the robot for Line Following and Green Detection
Start of September 2023 -	Construction of a Fork for a Drag method for Rescue.
End of September 2023 -	Coding for the Rescue Mission Drag and Release
Start of October 2023 -	Final testing of the robot
Start of October 2023 -	RoboCup India Regionals
Start of December 2023 -	Strategy planning of what changes needed to make it better for Nationals
Mid December 2023 -	Modifying the Regionals robot for the Nationals
End of December 2023 -	Coding for Line Following and Green Detection
Start of January 2024 -	Testing the robot for Line following
Start of January 2024 -	Writing the code for Rescue Mission using the Sweep-Pickup-Drop Method
Mid January 2024 -	Trial and Testing of the Final robot
End of January 2024 -	RoboCup India Nationals
Start of March 2024 -	Planning, Rebuilding the robot for Asia Pacific
Mid March 2024 -	Strategic Design for Rescue Mission
End of March 2024 -	Deciding on which software to use
Start/Mid April 2024 -	Building an Advanced robot for Line Following
End of April 2024 -	Testing
Start of May 2024 -	Building a Claw Machine with a Unique Mechanism
Mid/End of May 2024 -	Testing the new mechanism for the Rescue Mission Victims
Start of June 2024 -	Started writing a Code for OpenMV
Mid/End of June 2024 -	Trying to Integrate OpenMV in the robot
July 2024 -	Testing the Rescue Mission
August 2024 -	Mixing Rescue Mission Code with the Line Following Code
Start of September 2024 -	Enhancing the structural capabilities of the robot
Start/Mid of September 2024 -	Working on the Posters, Journal and the TDP
End of September 2024 -	Final testing
Start of October 2024 -	Testing the Final Programs and removing all final bugs
October 2024 -	RoboCup 2024 Asia Pacific

Figure – 2a Project Timeline

3) SOFTWARE

- EV3 lab is used for Line following and obstacle avoidance
- OpenMV with micro-python is used for Rescue mission.

3a) Line following

The line following system of our robot is implemented using EV3 Colour Sensors, which employs a **Proportional Line following** system. This system uses the sensor readings from the two-Colour Sensors to calculate the error value, which represents the variation between the robot's current position and the desired position on the line. The error value is then used to adjust the motor speed and direction, ensuring that the robot stays on course and follows the line. Our robot is a **Differential Drive Robot.** (If the error value indicates that the robot is deviating from the line, we increase the motor speed on one side of the robot and decrease the motor speed on the other side of the robot to correct its path.)

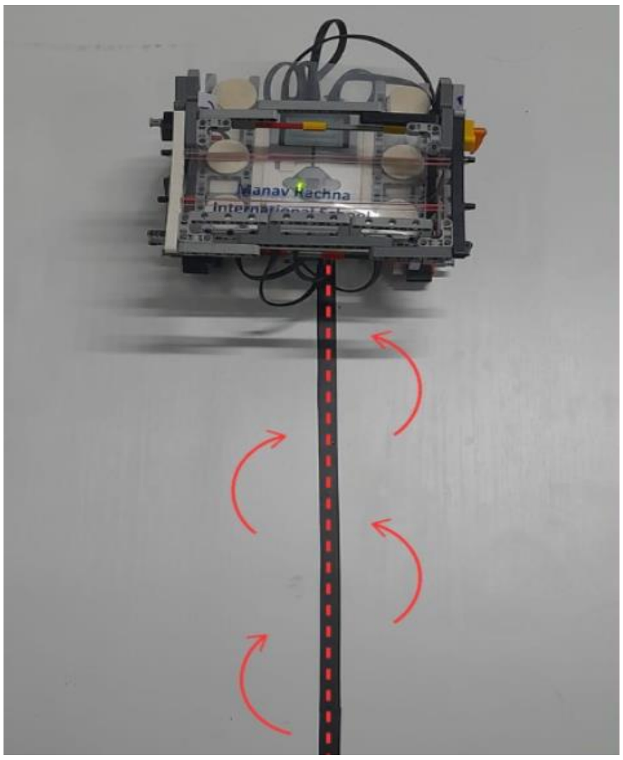


Figure- 3a

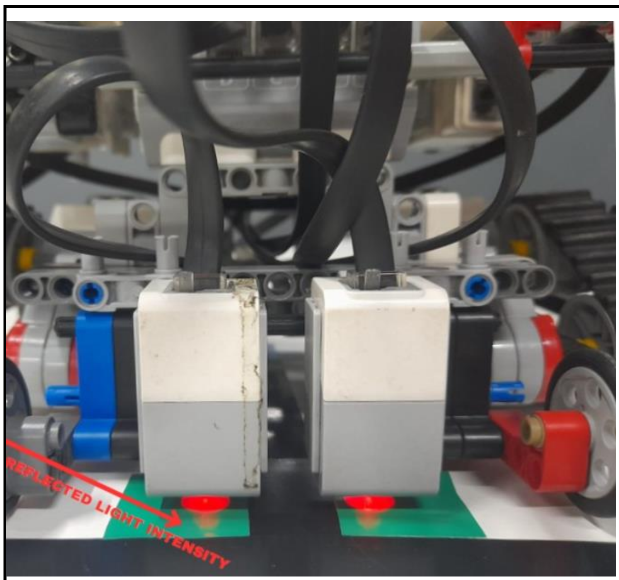


Figure 3b

3b) Turn Detection

Our robot takes turns on the basis of the previous error value, when there is a turn the robot tilts a little bit and stores that value, then it decides in which direction it should turn in based on that error value.

If there is a U- Turn the robot simply looks for green on both sensors then it takes a U- turn if it does not sense green it will look for red and stop moving. If none of these situations work it will move forward.

3c) Obstacle avoidance

To sense obstacles, we have used an EV3 ultrasonic sensor. If the distance of an object is greater than 6 cm the robot will continue line following, but if the distance is less than 6 cm the robot will turn to avoid the obstacle and then find the line again.

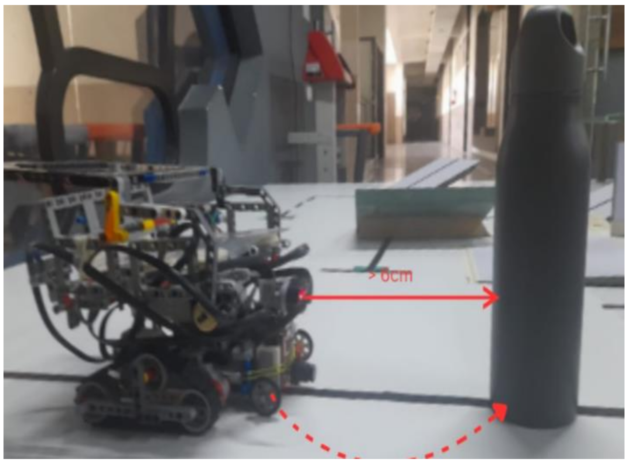


Figure 3c

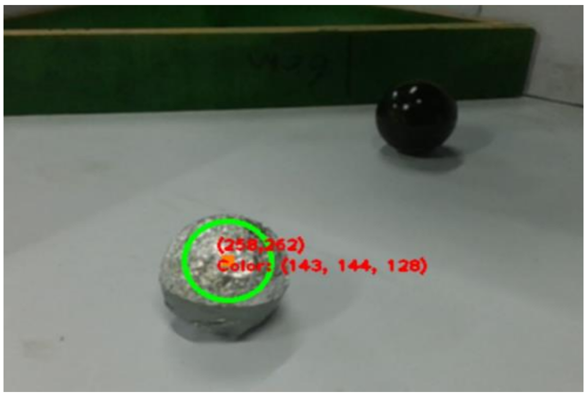


Fig 3d

3d) Detect Victims

We have used machine learning for the whole rescue mission using FOMO, first the camera detects the balls after which it sends a signal to Ev3 lego, using U-art connection.

After the signal is sent, Ev3 performs actions according to that signal.

First the robot looks for the black ball, by continuously moving right as soon as it enters the rescue zone

3e) Detecting corners

After the robot has detected the ball and picked it up, the camera starts looking for red corner (machine learning), as soon as it has found a corner the camera will send a signal to Ev3 to turn right, now as it will move right it will again start to look for red and then align with the corner.

Same for green corners.

3f) Exit

After the rescue mission has ended the robot will be parallel to the diagonal side of triangle (rescue evacuation zone), then using the servo motor an axle will hit the wall continuously. If the motor rotates 135 degrees, the robot will assume it is at the entry or exit. The robot will then turn: if the colour sensors detect a silver line, it means the robot is at the entry; if they detect a black line, it means it has found the exit. If it is at the entry, the robot will move forward and continue checking otherwise it will exit and resume line following.

3g) Connecting OpenMV with EV3

In our robot the OpenMV camera is connected to the EV3 brick using U-ART connections, where it functions as a sensor for the EV3. To program the OpenMV camera, we downloaded and installed the OpenMV IDE, and we used Micro Python to write the code. The OpenMV camera communicates with the EV3 by sending instructions in the form of numbers, allowing the EV3 to perform operations accordingly based on these instructions..

4) HARDWARE

Based on the guidelines set by the RCAP, we created a draft of the robot. We planned the design with a list of hardware components required. While we were exploring our options, we took into consideration the cost effectiveness of the component, accuracy, availability, reliability and efficiency and of course our competence to use that particular component.

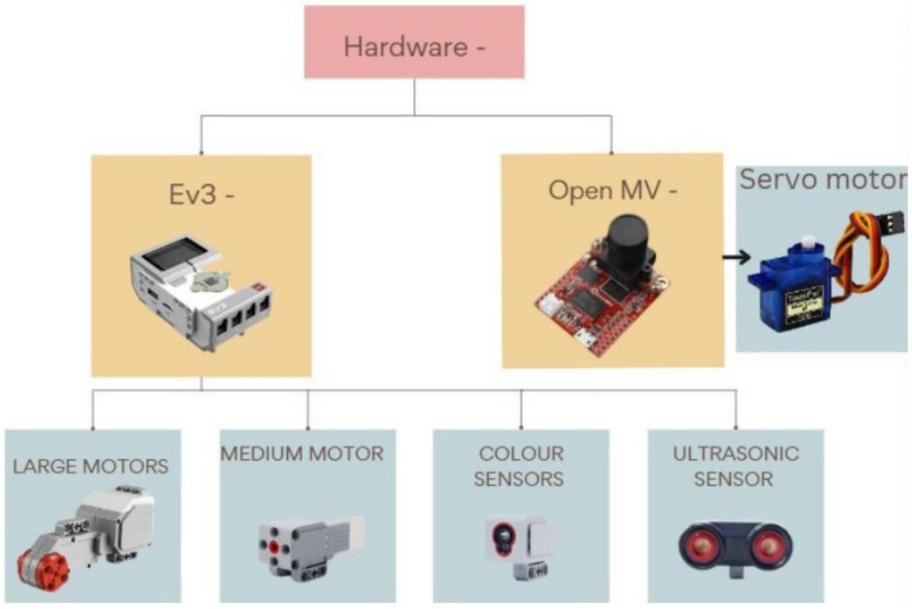


Figure 4a hardware integration plan

We can roughly categorise the hardware into 3 sections

- The main structure of the vehicle
- The rescue mission mechanisms
- OpenMV Camera

4a) Main Structure

COMPONENT	USAGE

Key Component	EV3 Lego Kit
Colour Sensors	• The colour sensors are mainly used for line following and intersections. • The sensors reflect an RGB rays (red, green, blue) on the desired place. The reflection of the rays will indicate the colours that were absorbed by the object. The sensors will measure the intensity of the reflected light using the photodiodes and finally can determine the colour of the object.
Two large motors	Two large motors are placed on both sides of the robot which is then attached to the tyres, these motors are responsible for movement and steering of the whole robot.
The caterpillar tracks	• These are made with sprocket technology. • One sprocket is connected with the motor It controls the movement - pivot or turn enabling better manoeuvrability by changing the speed of the sprockets on either side of the robot. • Provide good traction for climbing ramps and seesaws as compared to normal wheels.
The EV3 Brick	• The EV3 brick could be considered as the "Brain" of the robot, it is responsible for interpreting the code, supplying power, connecting with a variety of sensors and motors. • The brick also displays information and communicates with the code. • It also controls the motors that cause movement of the robot allowing precise movement such as navigating towards the target and avoiding obstacles.

4b) Rescue mission mechanisms

COMPONENT	USAGE
The Claw	• The claw is made by using rubber bands attached horizontally to beams with connectors. • Logic: When the claw is dropped vertically, the elasticity of the rubber bands will allow the claw to pick up the victims. • This claw is attached to a medium motor which allows it to move up and down.
The Platform	• After picking the victims the claw will place them on the inclined plane from here it will slide and collect near the Tailgate.
The Tailgate	• After the victims have gathered to one point with the help of the inclined plane, they are stopped by a Tailgate. • When the robot has detected the rescue corners using OpenMV it will align with the corners. • The Tailgate is dropped using a medium motor causing all the victims to fall in their designated spots.

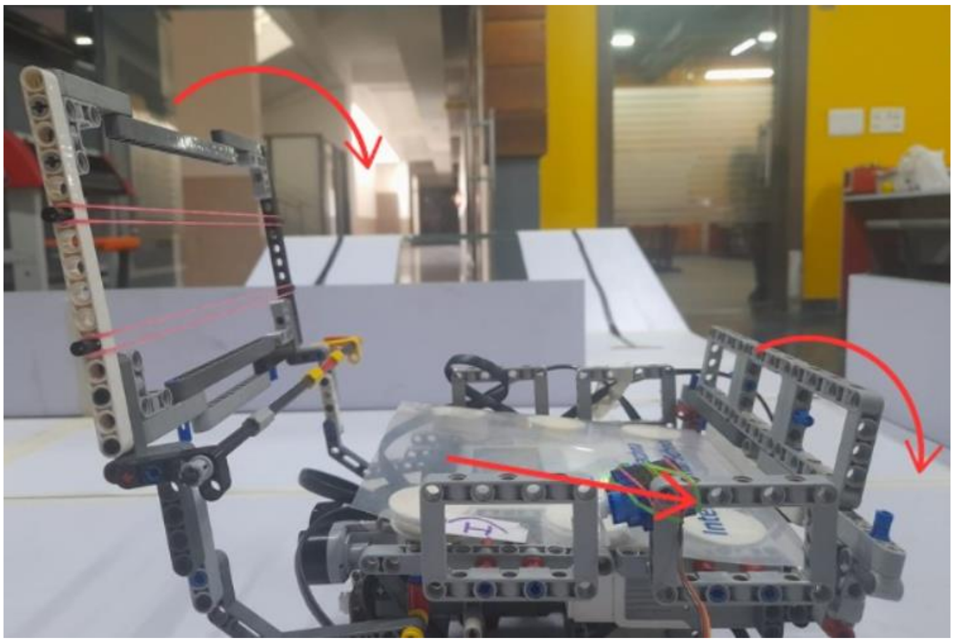


Figure -4b Rescue mission mechanism

4c) OpenMV Camera

- It communicates with EV3 with serial connection.
- In our robot the OpenMV camera is connected to the EV3 brick using an I2C connector, where it functions as a sensor for the EV3.
- To program the OpenMV camera, we downloaded and installed the OpenMV IDE, and we used MicroPython to write the code.
- The OpenMV camera communicates with the EV3 by sending instructions in the form of numbers, allowing the EV3 to perform operations accordingly based on these instructions.
- A major challenge we faced was incorporating OpenMV in our robot as it was crucial for our camera's height to be parallel to the radius of the balls (about 2.2cm - 2.4cm) to provide us with maximum range in which the balls can be detected.

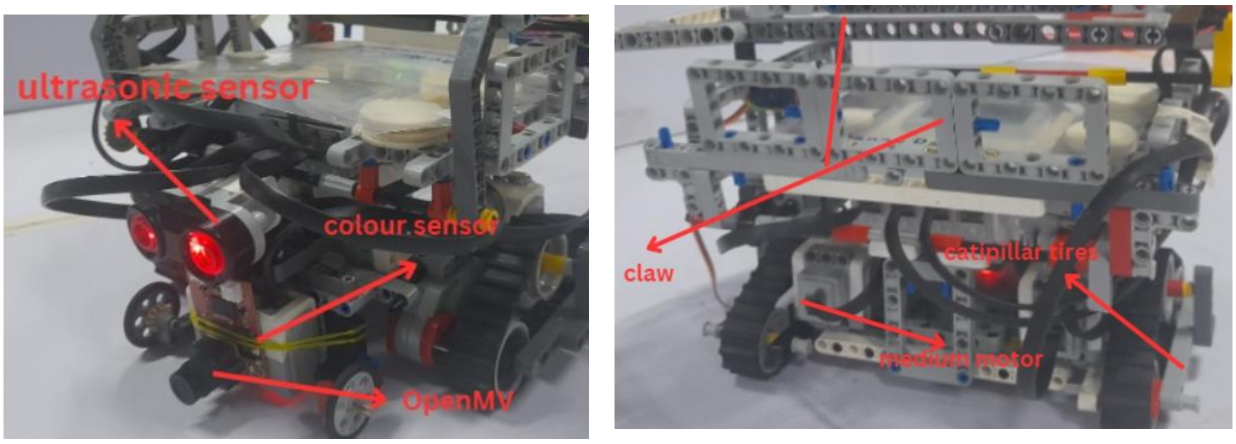
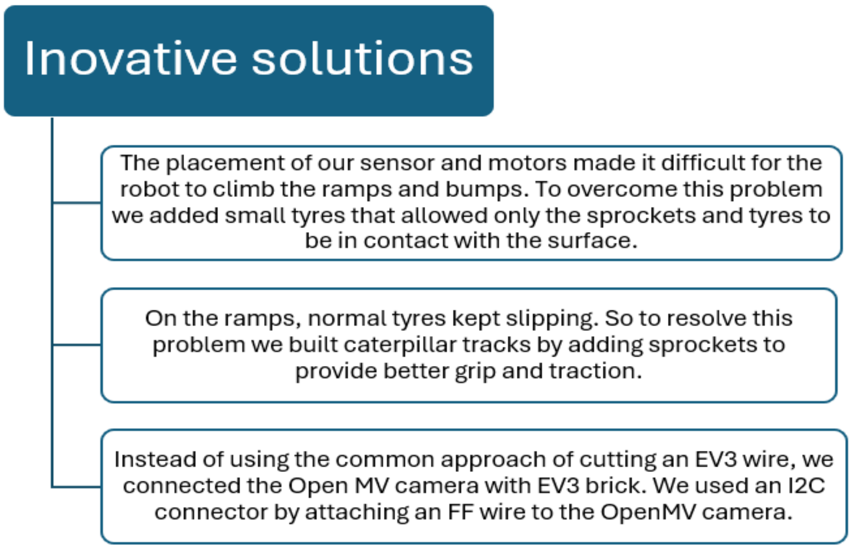


Figure 4c Overview of Robot

Innovative solutions:



Inovative solutions

The placement of our sensor and motors made it difficult for the robot to climb the ramps and bumps. To overcome this problem we added small tyres that allowed only the sprockets and tyres to be in contact with the surface.

On the ramps, normal tyres kept slipping. So to resolve this problem we built caterpillar tracks by adding sprockets to provide better grip and traction.

Instead of using the common approach of cutting an EV3 wire, we connected the Open MV camera with EV3 brick. We used an I2C connector by attaching an FF wire to the OpenMV camera.

- We created a blue blob on top of our claw so that as soon as the robot detects the silver line the claw will drop, as soon as the camera detects the blue blob it will start the rescue code.

5) PERFORMANCE EVALUATION AND CONCLUSION

In reference to the goals and expectations we had set for ourselves, we can say that we have really come a long way. During the course, we have significantly improved the performance of our robot in context to line following, obstacles and rescue mission. Throughout the journey we have, extensively tested the hardware and software aspects of our robot. We have done multiple practice runs by allowing the robot to manoeuvre its way around the map. By recording each one of our runs we were able to monitor what our robot is doing as opposed to what it should be doing. This also helped us to make numerous changes to our program to improve its efficacy. We also stored essential frames on the SD card such as the entrance of the evacuation zone, victim detection, aligning to the centre of the triangle etc. to allow us to post- evaluate its performance and implement changes that showed improvement in that respective field. Real time performance evaluation was indeed a challenging task for which we tested each component separately before installing it to the final design of the robot. Reliability and efficacy of the robot was tested and evaluated by setting up random obstacles like Lego pieces, stones, speed breakers etc.

Lastly, it is important to note that with strategic planning and set timelines which were closely met, we were able to improve the robot design and its working as per guidelines set by RCAP. We look forward to showcasing our Robot in actual arena and get to interact with our competing team members to understand their ideology and engineering used by them for their robots.

6) ACKNOWLEDGEMENTS

We extend our sincere gratitude to RoboCup team for organising such an event and giving us the amazing opportunity to participate and interact with best of the brains. A special thanks to our mentor, Mr. Amanurrehman, for his unwavering guidance and support in every step of the way. We are grateful to our school principal, Ms. Divjot Kaur - a constant source of motivation, for her timely intervention in any problem we might be facing during our journey. We also extend our deepest appreciation to our school teachers for their help in balancing our academic commitments with our participation in this competition. Finally, we would like to thank our parents for making such experiences and learning opportunities a part of our lives. We sincerely thank all those who have helped us throughout this journey including the amin and housekeeping staff of our school for smooth operations and their assistance whenever required.

7) APPENDIX

REFERENCES

Hardware-

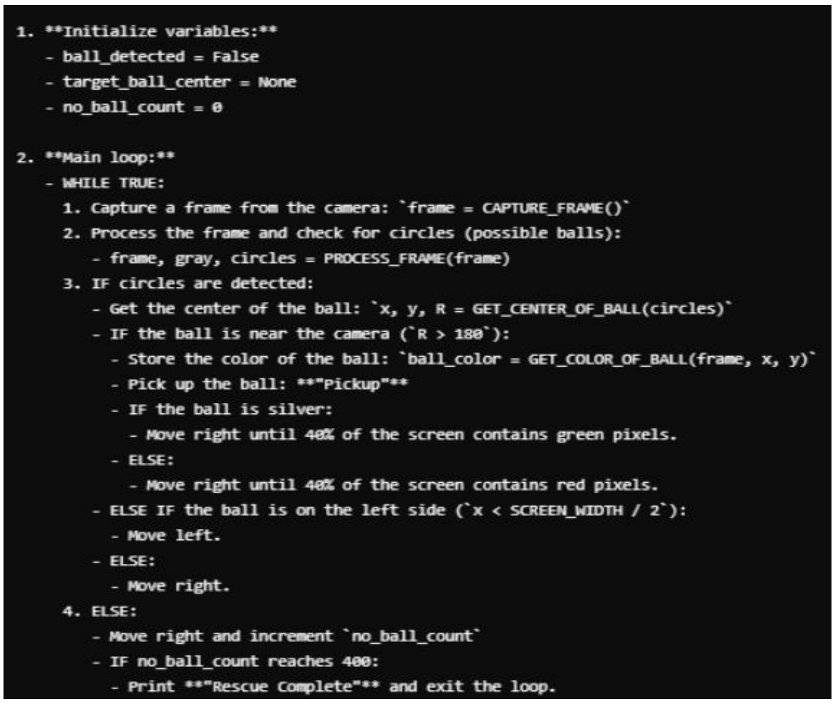
- EV3
- OpenMV
- Servo motor

Software-

- Python
- OpenMV IDE
- EV3 LAB
- Github

If you have suggestions, comments, or questions about our development you can also write us an email at: **codex.robots@gmail.com**

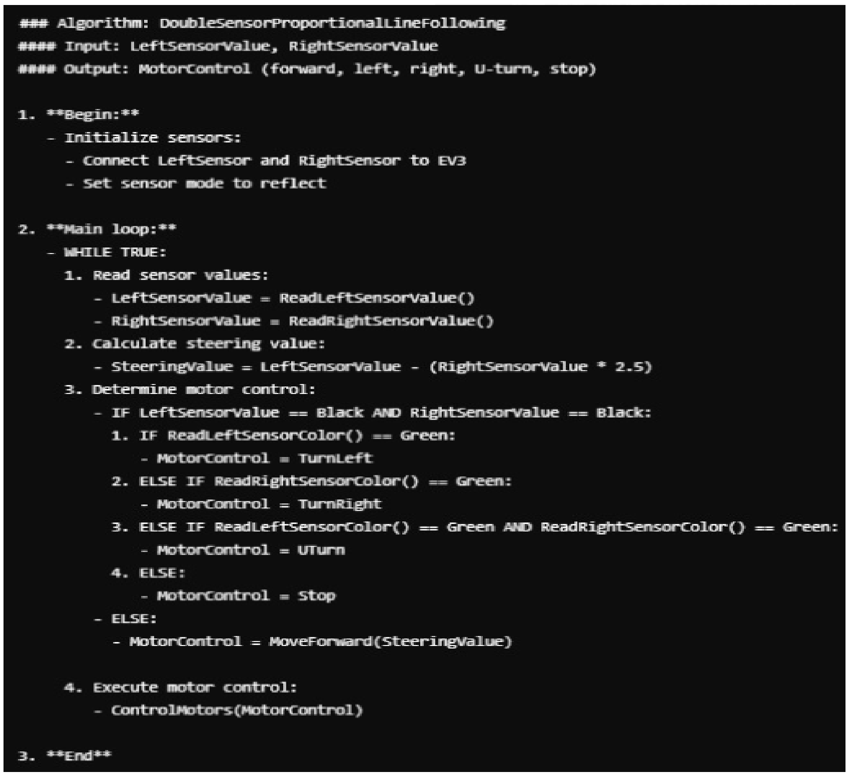
PSEUDO CODE FOR RESCUE MISSION



```
1. **Initialize variables:**
   - ball_detected = False
   - target_ball_center = None
   - no_ball_count = 0

2. **Main loop:**
   - WHILE TRUE:
     1. Capture a frame from the camera: `frame = CAPTURE_FRAME()`
     2. Process the frame and check for circles (possible balls):
        - frame, gray, circles = PROCESS_FRAME(frame)
     3. IF circles are detected:
        - Get the center of the ball: `x, y, R = GET_CENTER_OF_BALL(circles)`
        - IF the ball is near the camera (`R > 180`):
          - Store the color of the ball: `ball_color = GET_COLOR_OF_BALL(frame, x, y)`
          - Pick up the ball: **"Pickup"**
          - IF the ball is silver:
            - Move right until 40% of the screen contains green pixels.
          - ELSE:
            - Move right until 40% of the screen contains red pixels.
        - ELSE IF the ball is on the left side (`x < SCREEN_WIDTH / 2`):
          - Move left.
        - ELSE:
          - Move right.
     4. ELSE:
        - Move right and increment `no_ball_count`
        - IF no_ball_count reaches 400:
          - Print **"Rescue Complete"** and exit the loop.
```

PSEUDO CODE FOR LINE FOLLOWING



```
### Algorithm: DoubleSensorProportionalLineFollowing
#### Input: LeftSensorValue, RightSensorValue
#### Output: MotorControl (forward, left, right, U-turn, stop)

1. **Begin:**
   - Initialize sensors:
     - Connect LeftSensor and RightSensor to EV3
     - Set sensor mode to reflect

2. **Main loop:**
   - WHILE TRUE:
     1. Read sensor values:
        - LeftSensorValue = ReadLeftSensorValue()
        - RightSensorValue = ReadRightSensorValue()
     2. Calculate steering value:
        - SteeringValue = LeftSensorValue - (RightSensorValue * 2.5)
     3. Determine motor control:
        - IF LeftSensorValue == Black AND RightSensorValue == Black:
          1. IF ReadLeftSensorColor() == Green:
             - MotorControl = TurnLeft
          2. ELSE IF ReadRightSensorColor() == Green:
             - MotorControl = TurnRight
          3. ELSE IF ReadLeftSensorColor() == Green AND ReadRightSensorColor() == Green:
             - MotorControl = UTurn
          4. ELSE:
             - MotorControl = Stop
        - ELSE:
          - MotorControl = MoveForward(SteeringValue)

     4. Execute motor control:
        - ControlMotors(MotorControl)

3. **End**
```