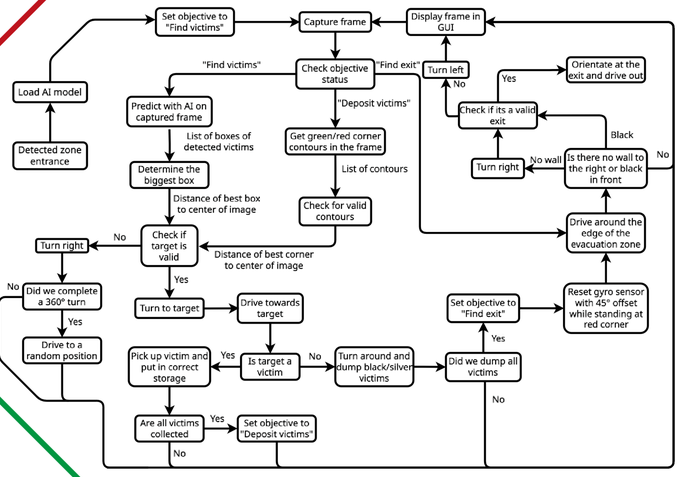
OVERENGINEERING²

EVACUATION ZONE



RESCUE LINE

Raspberry Pi 5

Opting for Raspberry Pi 5 as our main computing device was based on its impressive power as a single board computer and remarkable versatility.

Google Coral USB Accelerator

The USB Accelerator functions as a high speed AI coprocessor that relieves the Raspberry Pi 5 of the task of running the Victim Detection AI.

2x Arduino Nano

The Arduino Nano was selected for its proven reliability in managing readings from our diverse array of sensors. In addition, a second Arduino Nano is dedicated solely to driving the servos.

2x Adafruit BNO055

These gyro sensors were selected based on recommendations. Two sensors were integrated to mitigate occasional drift.

7x Pololu irs16a/irs17a 50 cm/130 cm Infrared Sensor

We use 130 cm infrared sensors on each side of the robot, plus two more at the front for obstacle detection, and also added a 50 cm infrared sensor on the gripper to confirm victim pickup.

1 m revoART 12 V COB LED-Strip

This specific LED-strip was selected for its densely packed LED-modules, enabling high illumination within a small space.

KY-019 Relay

This compact relay is utilized to toggle the light strip on and off based on whether we are in the evacuation zone.

5x EMAX-ES09MD Servo

After extensive testing with various small servo models, we settled on this most reliable servo due to its durable metal gears and high torque.

Diymore 35 kg Digital Servo

As the only servo discovered thus far capable of rotating 270° with durable metal gears, it has proven to be reliable during heavy use in previous projects.

SOFTWARE

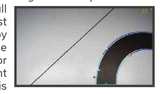
Much of the program relies on Python, a versatile and flexible programming language known for its user-friendly nature. Python was selected due to its widespread adoption, particularly in image processing use cases. The rich library ecosystem offered by Python was an important factor in our decision to use Python for our robot. Libraries like OpenCV and NumPy significantly contribute to the functionality of our codebase, handling substantial computational tasks. Moreover, we use Numba, a just in time compiler, to address Python's inherent speed limitations. The ability to swiftly launch the program without prolonged compilation further reinforced our preference for Python.

We use Arduino because of its ease of use and large user base, employing an Arduino Nano for sensor readings and another for controlling servos. Its foundation in C++ and extensive library support for our sensors and servos make it an ideal choice. The Arduino code communicates sensor data via serial to the Raspberry Pi and interprets digital pins to determine servo actions.

Bash scripts serve as a valuable utility for managing various smaller tasks. For instance, it automatically starts the program upon crashes or reboots of the Raspberry Pi. Additionally, we use these scripts to automatically install and update essential libraries and tools during the setup of a new SD card, making the configuration process much easier.

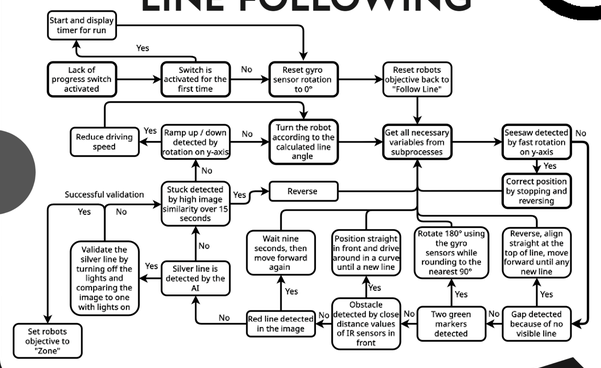
Line Following Algorithm

The robot follows the line using a wide angle camera mounted about 10 cm above the ground, looking directly down. By installing a very bright light, we completely eliminated any shadows that could be misinterpreted as the actual line. The robot follows the line by initially searching for seven points in the image. In addition to identifying the lowest point on the line, both for the full image and a version cropped in height, the highest, leftmost, and rightmost points on the line are identified. The robot typically follows the line by centering the highest point of the cropped image. If this point is not at the top edge of the cropped frame, either the left or right point is selected for following, depending on their distance to the edge of the frame, to prevent overrunning the line. The point that the robot is currently following is highlighted in red on our GUI.



At intersections, this becomes more challenging, particularly when standing at an angle, as the correct line may no longer be the only one at the upper edge of the frame. For this reason, a theoretical point (yellow in our GUI) is constantly calculated by connecting and extending the lowest and highest points of the cropped image to the upper edge of the frame. In the case of an intersection, the point at the top of the frame closest to the yellow point is selected to follow. If a green marker is detected, the point on the corresponding side of the line is chosen. The detection of these is based on checking the four sides around the marker for any black lines.

LINE FOLLOWING

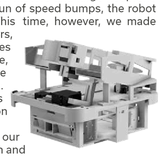


HARDWARE

Chassis

Over the past two years, we have been designing and optimizing the chassis of our robot. Autodesk Fusion 360 was used to realize our design ideas and take advantage of the freedom that 3D printing offers. We used PLA+ and various densities of infill to provide the necessary strength for navigating a robot weighing over 3 kg around the course.

Based on a number of components, including our powerful DC geared motors and large neoprene wheels that ensure good grip and safe overrun of speed bumps, the robot has already reached a considerable size. This time, however, we made sure that the robot had high, rounded corners, because we often got stuck on sharp edges before. Despite, or perhaps because of its size, we have once again decided to place a large touch screen above the victim storage system. This enhances the debugging process, adds aesthetic value, and has a minimal impact on the robot's overall size at this point.



To provide an image free of shadows for our camera, LED-strips are installed relatively high and straight above the line.

There are a lot of cables inside and on top of the robot, which have been organized in the best possible way. However, custom PCBs could be used in the future to simplify repairs.

Victim Gripper / Storage System

Inside the evacuation zone, we have to collect black and silver sphere-shaped victims and lift them into their designated corners. Our goal was to design a storage system that would sort the victims so they could all be collected at once, which greatly reduces driving times. We also designed a gripping mechanism that is attached to an arm to reliably pick up and lift the victims.

As shown in the render, the gripper opens using a small internal servo, which leaves enough space to accommodate a distance measuring IR sensor that confirms the successful pickup of victims. Furthermore, by adding two extra servos, the entire gripper mechanism can be lifted up and rotated on an additional axis. This allows us to sort the collected victims into different storage areas on the robot. The blades of the gripper are fitted with small edges that reach underneath the victims, enclosing and holding them in place during lifting.

Originally designed to fit a rescue kit as well, our storage system uses three storage areas and two independently controlled barriers. This system allows us to pick up the victims in any order while always being able to deposit the living victims first. The robot sorts the living victims to the left and the dead victims to the right. When the main barrier is opened, the living victims roll out first. After that, the barrier will close again, and the barrier in front of the dead victims will open, so they will roll up to the main barrier, which can now be opened again to deposit the dead victims. The procedure is illustrated in the following renders.

Raspberry Camera Module V3 Wide Angle

The choice of this camera as our primary line camera was driven by its superior sensor compared to the V2 module, enhancing overall performance and functionality.

Arducam B0268 Widge Angle Camera

This camera has a high resolution and a wide angle of 105°, which complements our Victim Detection AI model and provides sufficient coverage of the evacuation zone.

XL4016E1 Step Down Converter

In order to prevent crashes due to low voltage, a 12 A step down converter was incorporated to reduce the output of the LiPo battery from 7.4 V to 5.1 V for powering the Raspberry Pi 5.

2x Hailege XL6009 Step Up Converter

Two of these step up converters are used to increase the voltage from the 11.1 V LiPo battery output to the required 12 V level for the four DC motors and the light strip.

Heemol XL4015 Step Down Converter

We use this step down converter to transform the 11.1 V output from the LiPo battery into a stable 5 V supply, powering the five servos and their Arduino Nano.

2x Conrad 7.4 V/11.1 V Lipo

Our batteries were selected based on their proven performance with other teams and availability within our Robotics Club's inventory.

L298N Motor Driver

This motor driver has proven to be reliable in previous projects, effectively driving our four motors. Its compatibility with the Raspberry Pi 5 adds convenience.

4x 12 V DC Geared Motor

These motors were chosen due to their reliability and remarkable power. Utilizing four of these motors combined with neoprene wheels ensures that our robot can easily navigate ramps and speed bumps.

2nd Place Qualifying Tournament Hanover 2023

2nd Place German Open Kassel 2023

3rd Place European Robocup Varaždin 2023

1st Place Qualifying Tournament Hanover 2024

1st Place German Open Kassel 2024

GUI

We utilized CustomTKinter, a Python library with predesigned interface elements, to create a visually appealing graphical user interface (GUI) for the display on our robot. This GUI shows the feeds from the two cameras, readings from installed sensors, timers that track total run time and time spent in the evacuation zone, and various other debugging information. Furthermore, it contains a detailed, rotating model of our robot, displayed with 5580 images that were previously rendered in Blender and assigned to the corresponding real time rotation values obtained from the gyro sensors.

AI Victim Detection

To detect victims in the evacuation zone we rely on a self trained AI that is based on the YoloV8 architecture developed by Ultralytics. This state of the art AI model was chosen due to previous challenges encountered with tools like OpenCV in accurately locating victims, particularly the silver ones. After extensive labeling of over 3200 images and training many AI models through different iterations on Google Colab, a service that provides powerful GPUs for AI training, the model is now good at identifying both black and silver victims across the evacuation zone. Coupled with a wide angle camera and a Coral USB Accelerator, a coprocessor specifically made for running AI models, that ensures inference at about 17 FPS, we can efficiently complete the picking up and depositing of all victims in approximately 2 minutes. This significantly reduces the time required compared to our previous methods, which often failed to deposit all victims.

TEAM

German Open 2023

European Championship 2023

The original team **#RoboEvolution** was founded in 2017 by two former members. Current team member Marius joined the team in 2019. Using a light sensor based robot, however, never resulted in any qualification for the German Open, as there were often issues with different hardware components and the light sensors being unreliable.

In April 2022, Tim joined the team, proposed the idea of using a camera instead, and within two weeks we built a robot that qualified for the German Open at an internal school competition. Four weeks later, at the German Open 2022, we scored 22nd place in the end. A year of development later, we achieved our first podium finish at the Hanover 2023 Qualification Tournament, coming in second place. This result qualified us once again for the German Open 2023, where we also secured second place, earning us a spot in the European Championship in Croatia. Competing in Varaždin, Croatia, we managed to secure third place. This year, our team consist of only two members under the new name **OverEngineering²**. We have so far successfully won the local Qualificaion Tournament in Hanover, as well as the German Open and are excited to compete at the Word Championship 2024 in Eindhoven.

Tim Dumaschus

Main Programmer, CAD-Designer, Construction of the Robot

In April 2022, the Robotics Club and the former team **#RoboEvolution,** consisting of Marius and then one more team member welcomed me as a new member. I proposed using a camera-based line following system to compete at RCJ Rescue Line and took the lead on the project. To design the chassis, gripper, and storage system, I taught myself how to use Fusion 360. As the designer of all the different parts, I was also responsible for their correct assembly. Having the robot at home and being the only member of our team with prior programming experience in Python, the main programming language we use, I was able to program and test a significant portion of our code.

Marius Keiser

Machine Learning Specialist, Main Evacuation Zone Programmer

After being interested in robotics for over a year, I learned about the Robotics Club. Between me joining in 2019 and Tim joining in 2022 I gained a lot of experience, but never qualified for the German Open. I came up with the idea of using an AI model to detect victims in the zone, since we had many issues with that previously. Now I am responsible for the logic while inside the evacuation zone and the development of all our AI models, recently adding a classification model to identify the silver strip at the entrance of the evacuation zone.

Georg-Büchner-Gymnasium

RoboCup Junior