Rescue Line Team BIGG-IRMI

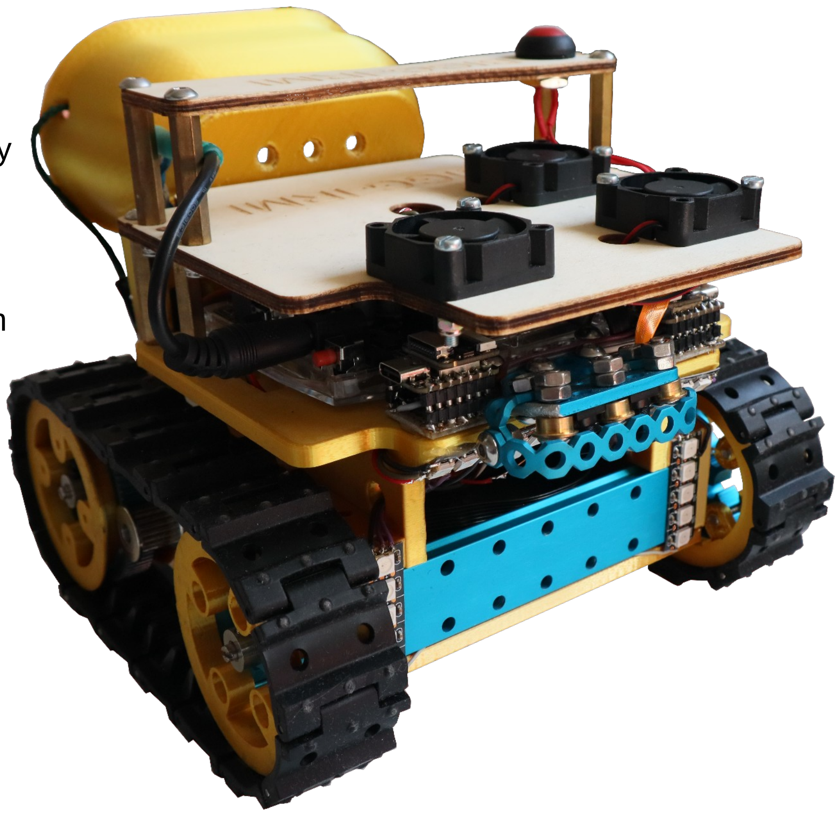
Technical Description Paper

Iren Biggel

- Born in 2008, Germany
- Team Captain
- Hardware Design
- Embedded Software Development
- Project Management

Mila Biggel

- Born in 2011, Germany
- Team Member
- Mechanical CAD
- 3D Manufacturing
- Media content creation
- Testing



Project Planning – from Design to Deployment

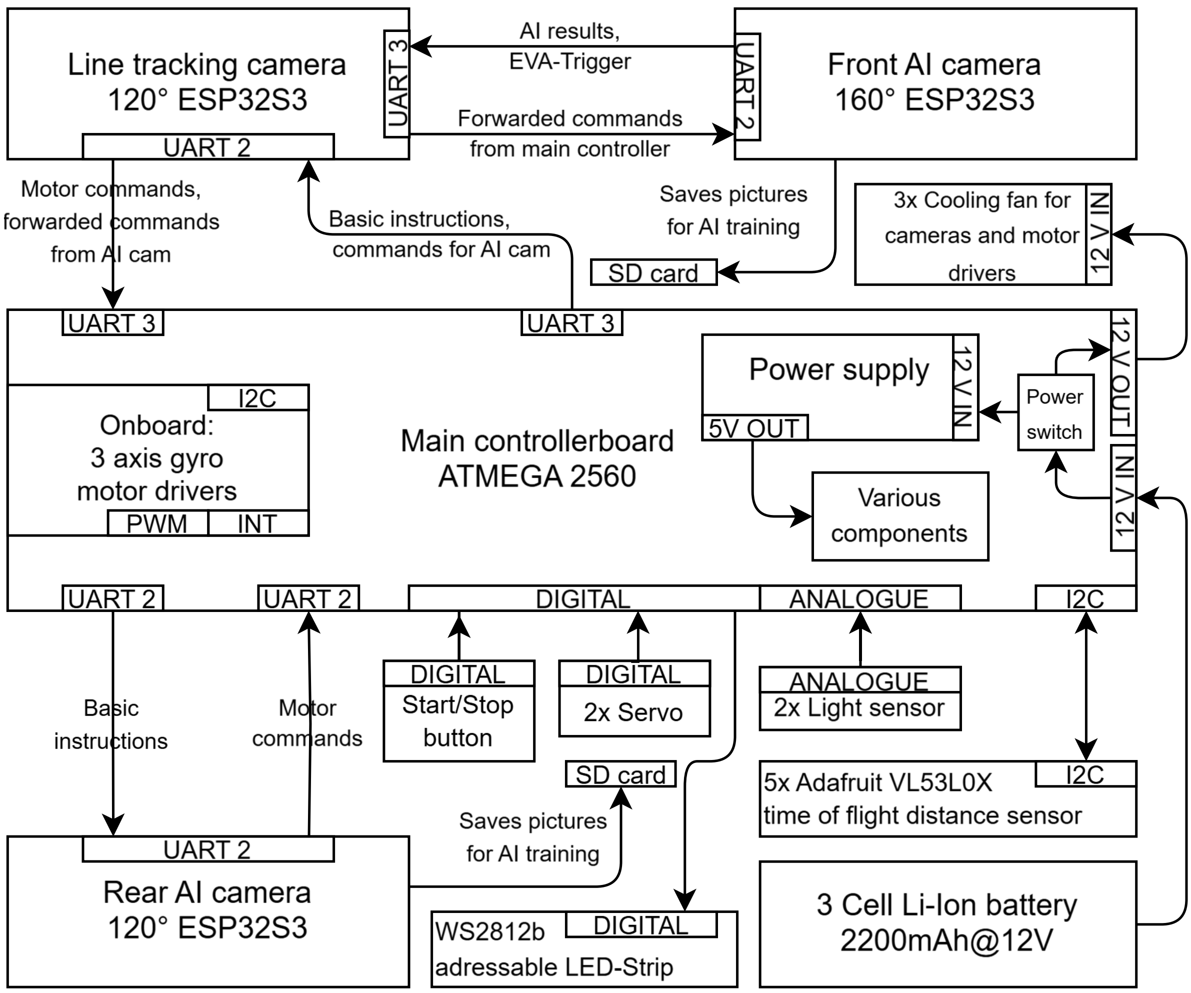
Requirements definition

Our robot shall be able to follow lines and make the right turning decisions at intersections. It should be capable of navigating obstacles, ramps, seesaws, bumpers and debris. Furthermore, the robot shall be able to recognise the evacuation zone and locate all victims and deposit them correctly in the colour-coded corners. It shall exit the Evacuation zone correctly and stop at the goal tile.

Overall Project Plan

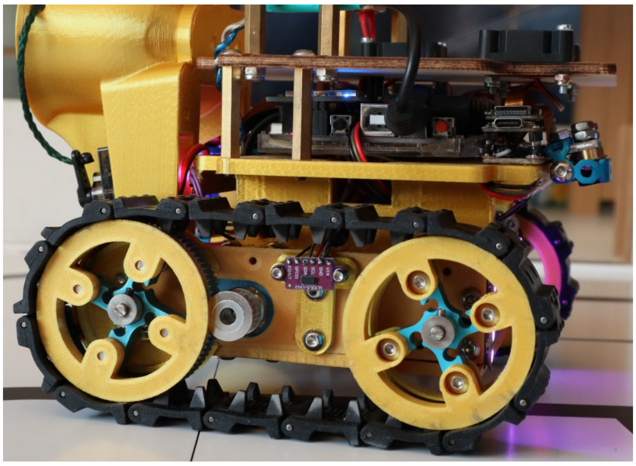
First of all, we need a chassis that meets our requirements, develop the power supply, select suitable sensors and camera modules and then attach them to the chassis in a useful way. After that, we select the appropriate electronic components, microcontrollers, drivers and HIDs and place them in the chassis. This is followed by software development for the selected components. It is imperative that we carry out regular tests to ensure correct functioning.

Integration Plan / System Engineering

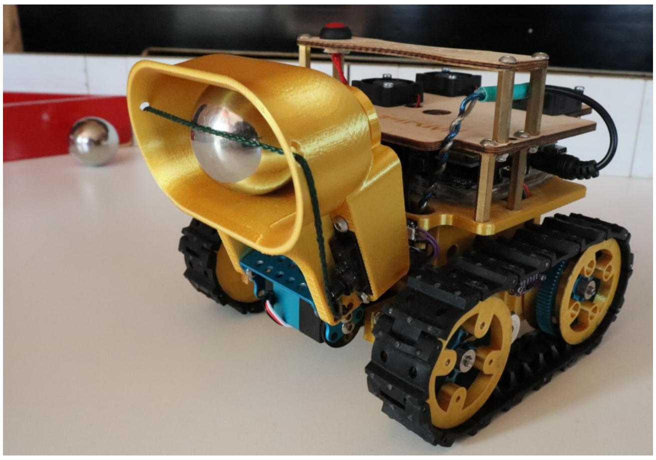


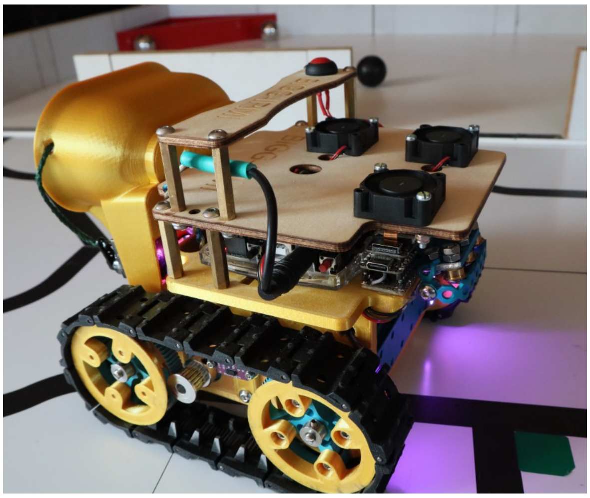
- 3D FDM printed chassis:
    - Individual customisation
    - Optimum dimensions for the course
    - Weight saving
    - Driven by rubber tracks for optimum traction and to overcome all obstacles.
- Power supply:
    - 3 cell Li-Ion battery with 2200mAh at 12V
    - Quick-charging capability
- Controller:
    - With on-board sensors
    - Integrated motor driver, controlled by digital and interrupt pins
    - At least 3 UART interfaces to communicate with other hardware
- Front camera #1, communicates with the controller via UART and enables line tracking and correct turning at intersections
- Front camera #2, communicates with front camera #1 via UART. The front camera #1 forwards the commands to the controller if necessary. It uses AI to check for Evacuation zone entrances / exits.
- Rear camera, communicates via UART with the controller and enables AI based victim detection, including AI supported victim depositing.
- Adafruit VL53L0X time of flight distance sensors, connected via our I2C-Bus
- WS2812b LED strips, controlled via digital output to ensure good lighting for our cameras during runs
- Light sensors for victim amount detection runs via analogue
- All servos are also controlled via digital outputs

Mechanical design and manufacturing

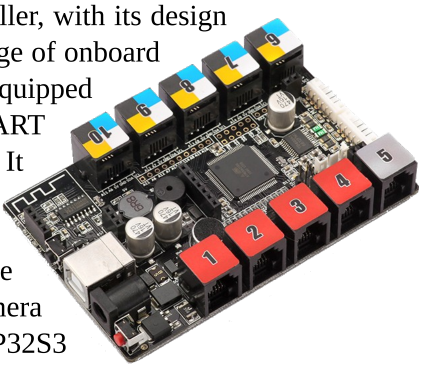


The majority of our robot is composed of 3D-printed components, enabling us to tailor them to our specific requirements. We use rubber tracks to ensure optimal traction and manoeuvrability. These components are driven via a gearbox and encoder motors, ensuring the motors operate within the optimal speed range. In order to achieve a low centre of gravity, we placed the power supply and motors at the bottom of the robot. In order to guarantee an unobstructed view of the front camera, we have strategically positioned the victim grabber device at the rear of the robot. As with the chassis, the component has been manufactured using 3D printing. The victim grabber device consists of a large victim container and a holding mechanism consisting of a cord that can be tensioned and released by a servo. Furthermore, the robot is equipped with light sensors that accurately determine the number of victims present. Conducting a conductivity test is not necessary as our AI-supported victim rescue system is able to differentiate between living and dead victims. The device is designed to rescue the victims by moving towards the floor, while the cord servo tightens the cord. At the point at which the servo has moved up, the victim is rescued. The existing hardware is then used to position the victims in their respective corners. To facilitate the identification of corners, we utilise our AI-based victim recognition system. The large servo is responsible for the downward movement of the entire victim grabber, while the cord servo is responsible for releasing the victims. The large servo is responsible for moving the entire victim grabber downwards, while the cord servo is responsible for releasing the victims. The rear camera is mounted low to improve victim detection. As the power supply had already been installed in the lower area, it was necessary to install the remaining components in the upper part of the robot, as well as the front cameras for tracking the line and detecting the entrance and exit of the evacuation zone. In order to protect the main board we have installed a laser cut wooden interlayer. This interlayer also serves the function of mounting the cooling fans for the cameras and motor driver, as these components generate a significant amount of heat during normal operation. The handle and the start/stop button have been mounted in the desired location.

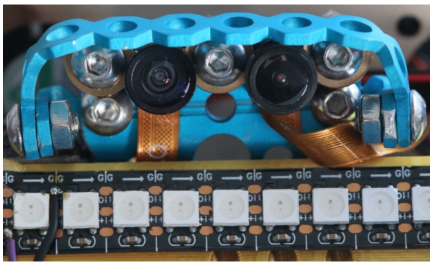




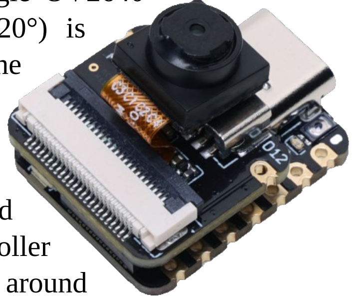
Electronic design and manufacturing



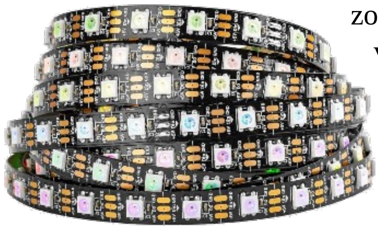
The Makeblock Me Auriga controller board is our primary controller, with its design based on an ATMEGA2560 µC. The board already provides a range of onboard sensors and high-quality motor drivers. The ATMEGA2560 is equipped with four hardware UART registers, which are essential for our UART communication protocols. It has proven to be very reliable for us. It



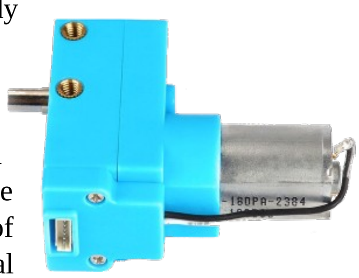
communicates with the main camera module via UART to facilitate line tracking. The primary camera module comprises two ESP32S3 microcontrollers. Each of these is connected to a 120/160° wide-angle OV2640 camera. The primary camera (120°) is



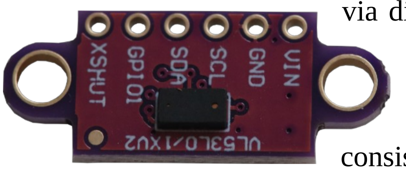
responsible for line tracking. Additionally, it transmits commands to the secondary camera (160°). It is responsible for the AI-based detection of the evacuation zone entrance. At the rear, we have deployed our 3rd ESP32S3 µC. The system utilises a 120° camera and employs AI to accurately locate the victims and evacuation points within the designated



zone. It communicates with the main controller via UART. We have installed LED strips around each camera to ensure that environmental conditions do not impact the camera's view. These are of type WS2812b and are controlled by the main controller via a digital signal. The encoder motors are controlled via the motor driver that is already provided by the main board



(6612FNG). In order to use the encoder, it is necessary to implement an interrupt-driven PD regulator. The single binary switch on the handle is used to start and stop the robot. The main controller interprets the signal



via digital. In order to successfully navigate around obstacles and locate the exit of the evacuation zone, we mounted a total of 4 VL53L0X time of flight laser distance sensors. The configuration consists of one on each side and two at the front, one of which is oriented in a straight line, while the other is positioned at a specific angle. They have demonstrated consistent reliability in competitive environments and are highly precise. In order to supply

our robot with power, we have selected a 3-cell Li-Ion battery with a total capacity of 2200mAh at 12V. This battery also has a quick-charge function,

which means it can be fully charged in less than an hour, which is sometimes necessary, especially during testing. Our victim rescue device is equipped with two light sensors. These sensors are fitted with a simple voltage divider, which reliably produces a readable voltage for our µC. The voltage

is then read by the main controller via analogue. In an effort to reduce noise levels, we have installed filter capacitors. Our rescue system also utilises a small MG90 micro servo to tighten the cord, as well as a large lifting servo from Makeblock. Extensive testing over many years has demonstrated the servo to be robust, reliable and powerful. Both servos are controlled by the main controller via digital PWM. We also use the on-board MP-6050 gyro sensor to detect and react to ramps and bumps, which is read via I2C. Three 12V cooling fans are fitted as the motor drivers and camera controllers get quite hot during operation.

Software

Architecture design

Our software consists of several elements, each running on a separate microcontroller. Each element contains our basic UART communication module and can therefore easily send motor commands, read sensor values and control servos. All programs are written in C++.

Overview of the elements:

- Line tracking, intersection handling and end detection
- Evacuation Zone entry detection (AI)
- Sensor interface, motor control, lighting
- Victim and Evacuation Point location (AI)

Line tracking, intersection handling and end detection

The program operates using the primary front camera (camera #1), which contains a bespoke line tracking algorithm that has been developed from ground up. A variety of sophisticated techniques were employed to optimise runtime performance. These optimisations enable us to process approximately 20 FPS (30 ms capture, 29 ms decode, 800 µs line tracking). For further information, please refer to the Innovative Solutions section.

Evacuation Zone entry detection (AI)

It runs on the auxiliary front camera (camera #2), with a custom-trained AI. Our AI model is based on MobileNetV2 and has been trained on a dataset of over 1,500 images that have been manually labelled. The system is capable of detecting the silver tape at the entrance to the evacuation zone, as well as differentiating between the entrance and exit of the evacuation zone during exit search. It is capable of processing approximately 0.6 FPS on a 96×96 RGB frame.

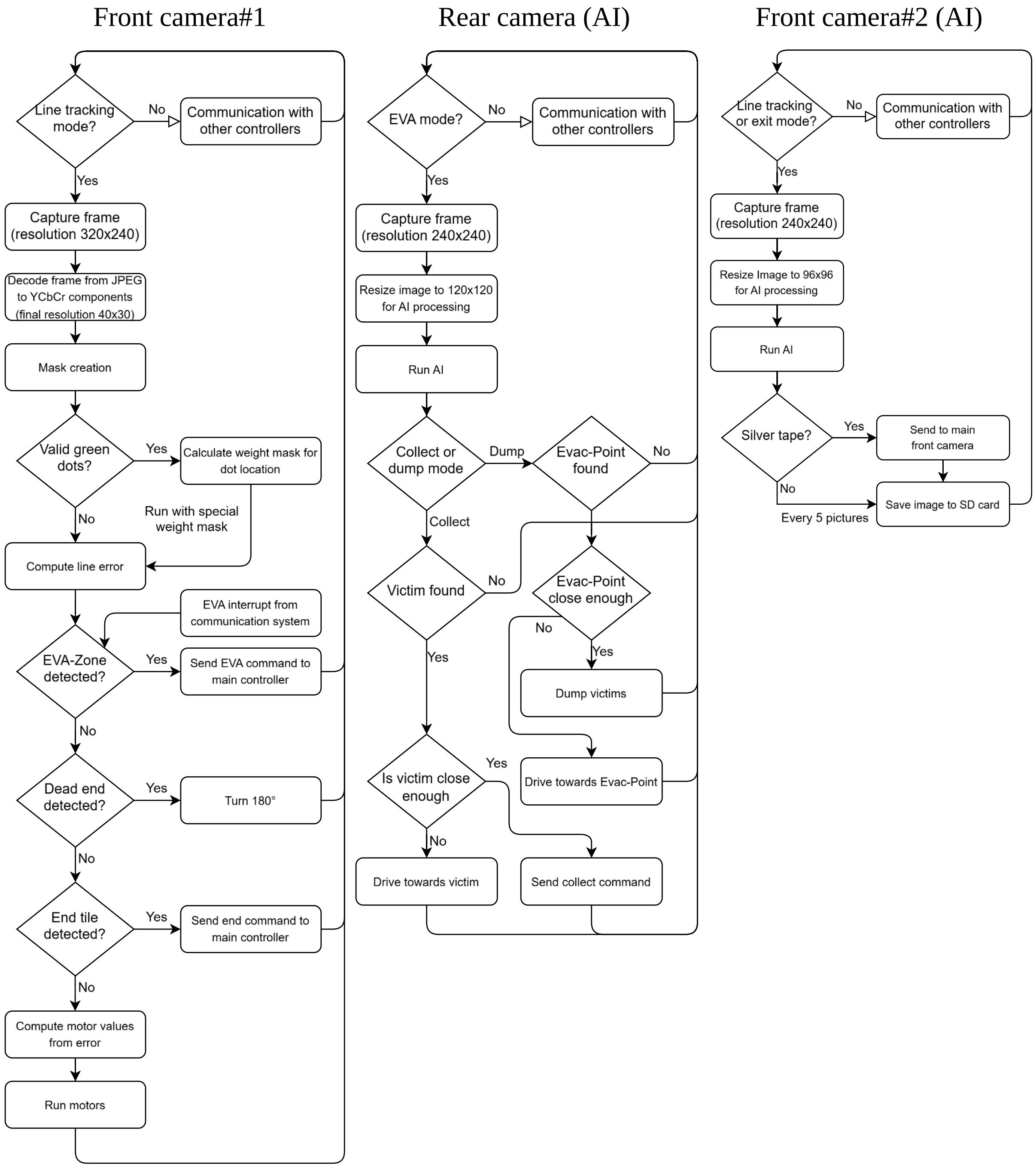
Sensor interface, motor control, lighting

It runs on the main controller board and is responsible for controlling the motors, the victim presence sensor and our 4 VL53L0X distance sensors. It receives commands from the front/rear elements and performs tasks for them, such as turning on the lights or measuring the distances. In line tracking mode, it constantly checks the front distance sensor for obstacles. It also handles the start/stop commands given by the button. In addition, a watchdog is employed to guarantee optimal performance during operation.

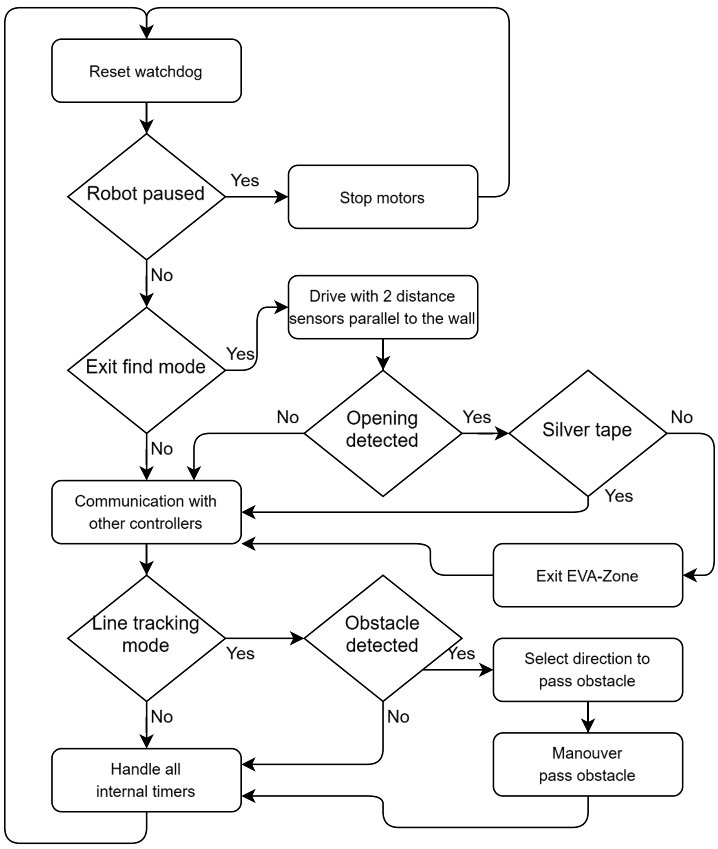
Victim and Evacuation Point location (AI)

It runs on the rear camera and employs AI that has been specially trained. Our AI model is based on MobileNetV2 and has been trained on a dataset of over 2,000 images that have been manually labelled. The system is capable of identifying live and dead victims within the evacuation zone as well as the evacuation points. It is capable of processing approximately 0.5 FPS on a 120×120 RGB frame.

Flow Charts for each software element



Main controller board



Innovative Solutions

Custom line tracking algorithm

Our robot follows the line using a custom algorithm that only works on the currently captured frame. It is evident that a direct comparison between frames is not applicable. The algorithm is first given a 40×30 pixel frame of YCbCr data. The Y component is then used to apply a tresholding algorithm, creating a boolean mask of 40×30 line data. Secondly, it corrects some errors (00100 > 00000 or 11011 > 11111) to provide the main part with clean data. The primary process involves iterating over each pixel to mark the centre of the line in a given row. The coordinates of the centre pixel are used to read it from a weight mask, after which the error value is applied. The weight mask remains constant, with the robot loading different versions of the mask depending on the situation. Lines not pointing in the direction indicated by the green dots will be ignored. In order to ensure the smooth handling of intersections, a calculated factor is applied to the error.

Distance based pixel weight mask (green is 1.0, black is 0.0)

AI-based evacuation zone detection

We have been facing challenges in detecting the evacuation zone for some time. However, we have now implemented an AI-based detection system to address this issue. The AI is capable of processing 96×96 pixel RGB images at a rate of approximately 1.2 seconds per image. This processing speed is more than adequate, even when tracking the line at high speeds, due to the 160° lens of the camera providing a wide field of view. Our AI model has been specifically trained to detect the silver stripe, and is based on Mobile Net v2. We are now able to detect the entrance to the evacuation zone in most cases.

Quality assurance

In order to ensure good quality of our software, we have an automated check that verifies the communication between each µC and the main board, and vice versa. In order to verify other components of the software, we build test parcours to evaluate various aspects of the code. To streamline the debugging process, we have incorporated optional serial printouts, as we do not utilise JTAG debugging. Finally, we perform full test runs to ensure everything functions correctly.

Performance Evaluation

In order to evaluate the performance of our robot, we conduct weekly tests to assess new parts and modules. These tests include multiple full runs, as well as extensive testing of newly installed components or newly developed software. In order to simulate a competitive environment, we systematically alter the lighting conditions between tests and introduce increasingly complex tiles until our robot reaches its limit. We then implement improvements and continue to challenge it. In addition, during the course of competitions, we analyse of the solutions developed by other teams. A selection of these solutions are then incorporated into our own development process.

Conclusion

After a year of developing this robot, we can say that we have improved in many areas. We reached many of the goals we set ourselves after last year’s season. We invested greater effort in the planning and conceptual design of our robot. Our development process was much more elaborate, and we planned, prototyped and tested more. The robot's design is also much more mature and reliable. The robot hardware is more straightforward to work on because each area (e.g. controller, electronics and drive) has its own layer in the robot. The software and electronics have also become much more robust due to the integration of new camera systems and a new controller architecture. We are anticipating this year's Robocup season with interest. It will provide an opportunity to assess the performance of our robot in a competitive environment, as well as facilitating dialogue with other teams and the exchange of ideas regarding their mechanical and software solutions.