ROBOCUOJUNIOR RESCUE LINE 2024 ENGINEERING JOURNAL

OverEngineering^2

Tim Dumaschus and Marius Keiser

Contents

1 Introduction

1.1 The Team

The original team, #RoboEvolution, was founded in 2017 by two former members. Marius joined this team in 2019, and Tim joined in 2022. In the end of 2023, we renamed our team Overengineering², since we felt the old name didn't fit us anymore. We currently consist of the members Tim and Marius, since our third team member from the last season left to focus on the Abitur. Tim is the main programmer, CAD designer, and builder of the robot, while Marius, who joined in 2019, is responsible for the machine learning models and the evacuation zone.

1.2 Team History

From 2017 to 2022, the robot was a light-sensor-based robot, but sadly, it never achieved any qualification beyond the local qualification tournament in Hanover. In April 2022, when Tim joined, he proposed the idea of a camera-based robot instead, so we quickly went to work and built a robot that qualified for the German Open within 2 weeks. A few weeks later, at the German Open 2022, we scored 22nd place out of ~50 entries, which we think is quite good for a 6 week old robot with technology we have never used before. After developing and overhauling the robot completely for a year, we qualified for the German Open in Kassel 2023 at the local tournament in Hanover. We scored 2nd place at the German Open and qualified for the European Championship in Varaždin, Croatia, where we ended up in 3rd place. This season we won the local qualification tournament in Hanover and recently also won the German Open in Kassel, qualifying us for the World Championship in Eindhoven, Netherlands.

1.3 Operational Logistics

We mostly communicate through our own Discord server and via email with our teachers if we need to purchase something or send them a bill for a part that we purchased. Since 2022, we have been using GitHub for easy collaboration when coding. We buy most components twice unless they go directly on the robot, so we either have spare parts or both team members can test with them. Our expenses are mainly financed by sponsors, who are taken care of by our teachers. We meet in person every Thursday in the Robotics Club of our school to work, test, and innovate on the robot, as well as discuss future plans. Tim can also test the robot at home, since he has borrowed his own Rescue Line course from school.

1.4 This Document

With the choice to maintain our journal in a LaTeX document, we initially put in a lot of work, but over time, this has paid off, as it is now very easy for us to create entries. To enable easy collaboration we use the online LaTeX editor Overleaf. We tried to link to as many sources and documentation as possible, so please pay attention to underlined words, these are always hyperlinks.

2 Daily Log

Date: 2023-06-28

First AI testing

Tasks done today:

- Bought Google Colab Pro for AI training
- Trained a Ultralytics - YoloV8 AI model based on a few thousand ball images from the Open Images V7 dataset on a Google Colab Nvidia A100 instance
- Tested the model on some images taken in the evacuation zone, seen in Fig. 1

Issues and solutions:

Issues	Solutions
Many difficulties and poor results detecting victims in the evacuation zone with regular image processing through OpenCV's Hough Circles method, causing us to often wrongly detect victims in the background or miss silver victims entirely	After some research online we believe a custom YoloV8 detection model from Ultralytics can solve this issue, because detection AI's can be trained to work in any situation. That is why we trained a model based on images containing balls in the the Open Images V7 dataset to test if it is even feasable.
The free tier of Google Colab is very limited and I quickly ran out of credit. It only provides a Nvidia Tesla T4, a GPU based on the 20 series Nvidia GPU's, which are very slow and take long to train	Buying Google Colab Pro, since it provides Nvidia A100 GPU's which are many times faster and we get additional credit to train with

Conclusion / further planning:

Even though all of the images in the Open Images V7 dataset don't contain the type of balls that are used by the RoboCup Junior, the AI is able to detect the black victims with a quite high confidence of 86%, shown by the number next to "Ball" in Fig. 1. We chose to use a YoloV8 model by Ultralytics because they seem to be very popular for their simplicity and ease of use, which we are able to confirm because their training documentation is very exhaustive and it's possible to get training going with 3 lines of python code. We are very happy with the first results and the next step will be to create a custom dataset that is based on images taken in the evacuation zone.

Figure 1: The first test of the AI

by Marius

Date: 2023-06-29

Creating this file

Tasks done today:

- Create a shared LaTeX file with Overleaf
- Create an new LaTeX-Environment with all essential content for the daily log by using

```
\newenvironment{}
```

Issues and solutions:

Issues	Solutions
A daily log typically involves a significant amount of copying and pasting for formatting purposes Had problems with using provided \newenvironment{} arguments within the end_def section of the command	Using LaTeX-Environments, we minimized the need for excessive copying and pasting worked around this problem by initially defining these arguments as \newcommand{}

Conclusion / further planning:
While creating LaTeX files can be time consuming, it often proves to be more time saving and flexible when adding new content, resulting in significantly cleaner documents compared to, e.g., Word.

by Tim

Date: 2023-06-29

Disassembling the old robot

Tasks done today:

- Disassembled the whole top section of the robot
- Removed and then replaced some of the wheel-hubs
- Removed old solder connections and wires from certain parts we will reuse on our new robot (e.g. motors, digital voltmeters, IR-sensors, ...)
- Took measurements of part sizes

Issues and solutions:

Issues	Solutions
Some grub screws on the wheel-hubs were turned round and could no longer be gripped	Had to drill them out using a slightly undersized drill in order to prevent damaging the threads of the wheel-hubs; Nevertheless, I changed them later for safety reasons
Found a broken solder connection on the old step-up converter, which probably was the reason the robot randomly stopped working properly in the past	Using a higher quality solder as well as checking every solder connection in the future precisely should help prevent this type of issue (this connection was only added after the robot had already been built)

Conclusion / further planning:

Disassembling our robot from this year's finished competition was the first necessary step in improving our design for the next robot. For designing an improved chassis in CAD in the future, we needed to take some important measurements, such as the motors, which were hard to get without a complete disassembly (see Fig. 2). Because of their generally reliant performance over the last few months, we plan on reusing some of our old components and save some money there. With the new measurements, I will probably start redesigning the robot over the next few weeks while Marius tests the AI detection.

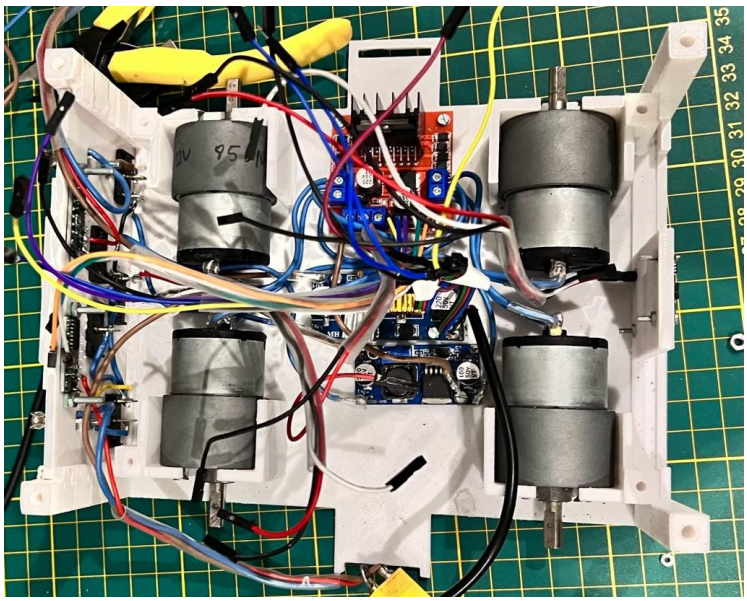


Figure 2: Our old, mostly disassembled robot

by Tim

Date: 2023-06-30

Settling on parts to use for the new robot

Tasks done today:

- Comparing camera modules from different manufactures
- Settling on the following part list to design our new robot around:
- Ordering:
  - two Coral USB Accelerators
  - additional irs16a and irs17 IR-distance-sensors
  - an additional BNO055 gyroscope sensor

Issues and solutions:

Issues	Solutions
Many camera modules with different characteristics to choose from and different requirements for line following and victim detection	Using different camera modules for line following an victim detection enables us to choose different camera modules for different purposes
The performance of running AI models on a Raspberry Pi 4B is pretty bad	Ordering two Coral USB Accelerators that add an additional TPU to the Pi, which hopefully improves its performance and enables us to detect victims with an acceptable frame rate (second one is a spare part)
After a few rotations of the robot, the X-axis readings of our single BNO055 became imprecise	We hope adding a second sensor, somehow reading them out simultaneously, and then averaging their reading will solve this problem

Conclusion / further planning:

Since we already used most of the parts, like motors, motor drivers, batteries, step-up/step-down converters, distance and gyroscope sensors, and the LED strip on the old robot, our choice for parts for the new robot was easy because they have already proven to be reliable and work well together. Therefore we settled on the following part list for our new build with a few upgrades compared to the last robot:

- Raspberry Pi 4B 8GB RAM (used a 4GB variant before, very reliant and well documented, but we already ran out of ram memory sometimes even without any AI)
- Coral USB Accelerator (hopefully enables us, running AI models with an acceptable frame rate)
- two Arduino Nanos (used before and been proven reliable; one with shield for reading out distance sensors an one for any servo controls)
- Raspberry Pi Camera Module V3 (for line following)
- Arducam B0268 16MP Wide Angle (for victim detection)
- two 11.1 V/7.4 V Conrad Lipo Batteries (used before and been proven reliable)

- two XL6009 Step-Up Converters (used before and been proven mostly reliable; for stepping up the LiPo's 11.1 V to 12.1 V to power the motors and LED strip)
- two XL4015 Step-Down Converters (used before and been proven mostly reliable; for stepping down the LiPo's 7.4 V to 5 V to power the Raspberry Pi, Arduino's and Servos)
- L298N Motor Driver (used before and been proven reliable; works well even with 3.3V GPIO pins from the Raspberry Pi)
- four 12 V DC Geared Motors (used before and been proven reliable)
- two BNO055 Gyroscope Sensor (used before and been proven mostly reliable;)
- at least 6 irs16a and irs17 IR-Distance-Sensors (used before and been proven reliable; one on each side and two in the front for obstacle detection)
- 1 m revoART, 12V COB LED strip (used before and been proven reliable; very bright)

A diagram of how we plan to connect all the components can be found in Fig. 3.
Besides the parts we were satisfied with on our old robot, especially the cameras needed an upgrade. The reasons for our camera choices are:

- one unused USB-A port on the Pi as well as only one camera module port
- 105° wide angle camera for victim detection - less need for turning while searching for victims
- high resolution camera for victim detection - for detecting even the smallest of victims
- reliable exposure compensation for line camera - needed because of very bright LED's
- higher resolution, image quality and frame rate than Camera Module V2
- experience with the Raspberry Pi Camera Module V2 and other USB cameras

We are currently planning to use at least one of the distance sensors on each side of the robot to determine, in combination with our gyroscope sensors, the current position of the robot in the evacuation zone.
I will probably have to wait to redesign the robot until all the new parts arrive to take measurements of them.

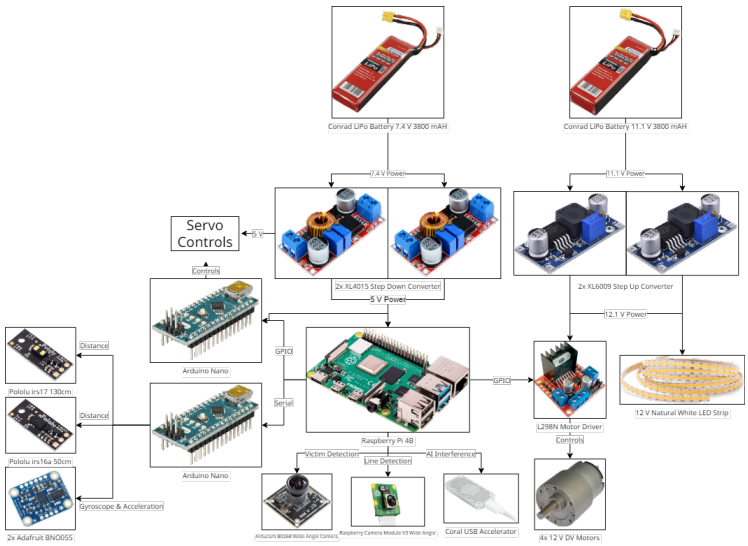


Figure 3: Component diagram for new robot

by Tim

Date: 2023-07-01

First testing of Coral Ai

Tasks done today:

- Installed the Coral TPU into a test Raspberry Pi 4B
- Ran benchmarks with pretrained models

Issues and solutions:

Issues	Solutions
Trying to run YoloV8 models with PyCoral produced weird outputs	None yet found
Issues with exporting YoloV8 PyTorch models to a Coral TPU compatible format and kept getting crashes due to libraries	Installing a slightly older version of flatbuffers that resolved the exporting issues

Conclusion / further planning:

We saw a significant performance increase over regular CPU inference on the Raspberry Pi 4B, so we will definitely keep using it in the future for our AI models, but we do need to first figure out how to convert the output from PyCoral into a usable box format, maybe someone online has figured that out already, but that is for the future.

by Marius

Date: 2023-07-07

Redesigning the old robot

Tasks done today:

- Finished the base plate of our new chassis, mostly copying proven dimensions of our old design using Fusion 360
- Measured and added the new, much bigger USB camera to the design

Issues and solutions:

Issues	Solutions
Cutouts for attaching side plates were very thin and broke with the old design	Moved them further in and made them smaller
The new camera lens is much longer than the old one and almost protrudes into the field of view of the camera, looking down on the line	Moving the attachment plate as close to the motors as possible and hoping for the best
Some nuts have come loose from the screws on the old robot	Already adding inserts for the nuts to the design and later pushing them in with a hot soldering iron
The two LiPo batteries lying on the motors were hard to reach	Adding big cutouts to the compartment they are in to reach them from below
Perhaps need to disassemble parts the robot in the future	Always use screws to connect different parts instead of relying on printed connections or glue

Conclusion / further planning:

As seen in Fig. 4 we solved all the issues we encountered with our old design, while keeping the overall dimension, which already worked well with the part we are planning to use again. Unfortunately, only I can work alone on the CAD file, but exchanging ideas and current designs with Marius still helps me a lot.

I will try finishing the whole CAD design until i go on vacation on July 13, 2023, so we can start printing the parts an assembling the robot when I am back.

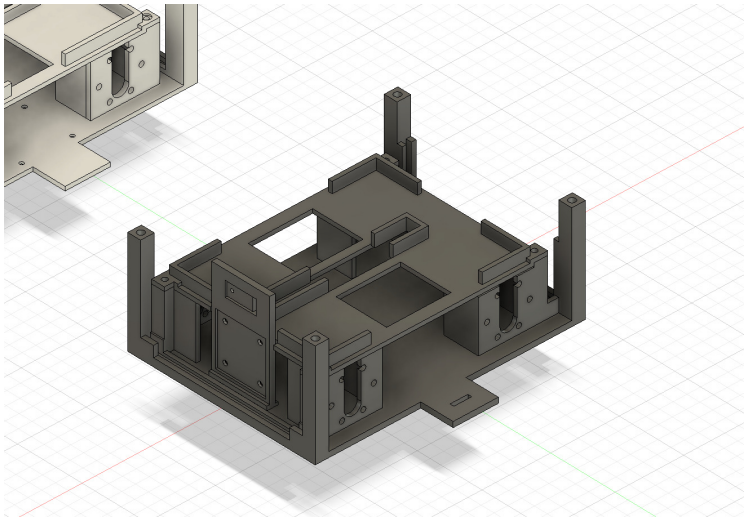


Figure 4: CAD design of the finished base plate of the new chassis

by Tim

Date: 2023-07-08

Designing the upper panel with high, rounded corners

Tasks done today:

- Modeling more components such as cameras, LiPo's, motors and wheels
- Start modeling the upper panel of the robot
- Prototyping a design for the new, rounded LED bar (see all of today's progress in Fig. 5)

Issues and solutions:

Issues	Solutions
The old robot often got stuck on the sharp edges of the upper panel	Back then the sharp corners were already filed round; now rounded corners are added directly to the CAD design
While driving against walls only 10 cm high in the evacuation zone, the upper panel sometimes slid over the walls, causing the robot to behave differently from driving against higher walls	Adding rounded, 1.5 cm high walls to the upper panel
Often got stuck on the old protruding LED strip mount	Designing a new, strongly rounded off mount, keeping the number of LED's, the angle to the ground, and cutout for the camera about the same as in the old design, while shortening the mount by about 2 cm and still making it printable - only the outer walls are finished yet

Conclusion / further planning:

As seen in Fig. 5 the basic, already improved design is finished so far, but many components and mechanisms need to be added, such as:

- Raspberry Pi, Coral TPU, 2x Arduino Nano, IR-Sensors, Display, H-bridge, step-up & step-down converters
- cable feed-through in the center of the upper panel
- front/back and side plates
- victim storage solution
- victim gripper arm
- finished LED mount

We try to complete the draft during the summer vacation, as there are massive delays in the following months due to school trips and exams.

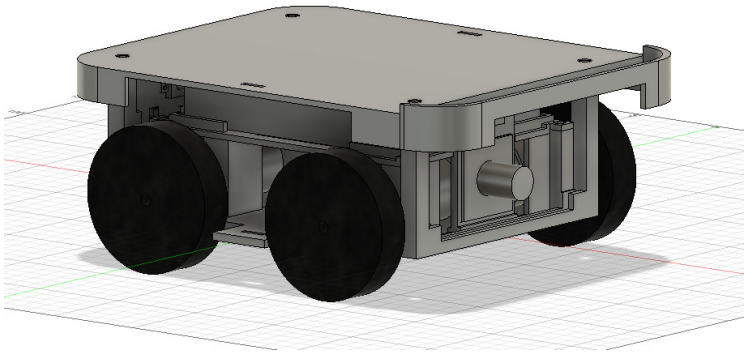


Figure 5: CAD design of the upper panel of the new chassis with added components

by Tim

Date: 2023-07-09

Victim storage prototyping and back plate design

Tasks done today:

- Added and arranged missing components (Raspberry Pi, Arduino's, Display)
- Improved old back plate design and adding easy access fuses, digital voltmeters, and switches to the design (as shown on the right-hand side of Fig. 6)
- Copied the old camera servo mount into the new design
- Decided to collect, store and unload all victims at once this time
- Ensured that height restrictions should also be observed with inclined victim storage and with display (as shown on the left-hand side of Fig. 6)

Issues and solutions:

Issues	Solutions
Meeting the height and width restriction of 25 cm	The robot reaches a height of 199 mm when choosing the flattest possible angle of the container where victims roll down and placing the screen directly above it,
Accidentally turned off a switch in the back plate in the last competition	Insert all the switches into the back plate
We may damage the back plate when driving against the corner to unload the victims	Adding protruding cube later printed with 100% infill to drive against them

Conclusion / further planning:

We chose to collect, store, and unload all victims at once with this design because of

- time saving advantages (less travel time to corners)
- reduced complexity (the robot has to find and navigate only to both corners once)
- lower risk of colliding with the dead victim when picking up the living victim, which makes picking it up more difficult

Now we just have to design the actual compartment and an arm that can lift the victims from the ground up onto the robot.
The fact that the robot is now much higher could perhaps lead to problems later that we are not yet aware of, but collecting the victims on the robot is certainly the right step for us.

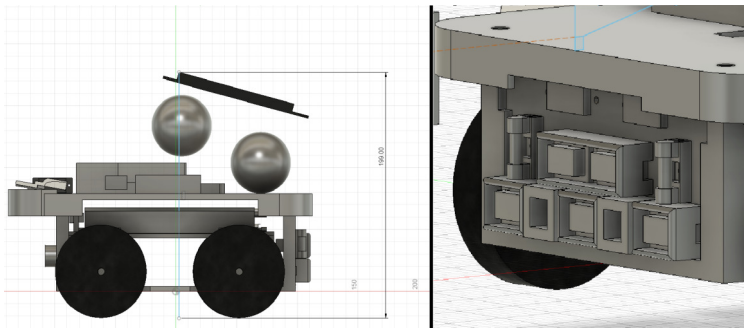


Figure 6: Observed height restrictions and back plate design

by Tim

Date: 2023-07-10

Redesigning the LED bar

Tasks done today:

- Redesigning the LED bar to fit as many LED's as possible
- Adding the camera cutout to the LED bar

Issues and solutions:

Issues	Solutions
Old design worked well with removing any shadows in the camera image, but the robot got stuck on the protruding strip mount because of its shape and length	Increase the angle of the LED's slightly, because of the shorter mounting surface and maximizing the surface area adapted to the width of the LED strips while keeping the shape 3D-printable

Conclusion / further planning:

As seen in Fig. 7 the robot should hopefully no longer get stuck on the LED strip mount, while the amount of LED's fitted to the mount only decreased slightly.

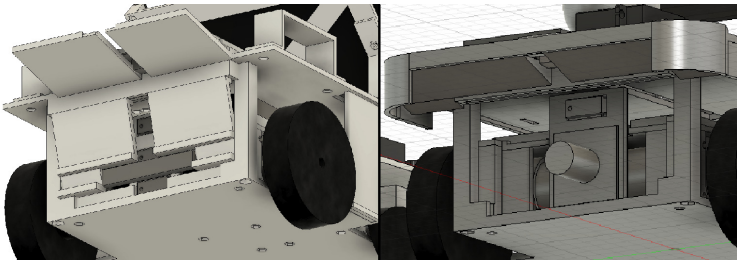


Figure 7: Old and new LED bar design

by Tim

Date: 2023-07-11

Searching for the optimal relay

Tasks done today:

- Searched through Amazon and different retailers for relays that are compatible with a Raspberry Pi 4B
- Discussed about the possibility of a team name change

Issues and solutions:

Issues	Solutions
The bright LED bar blocks the view of the bottom camera in the evacuation zone, making it unable to detect victims	Installing a relay so that we can turn the LED's on or off through the GPIO pins of the Raspberry Pi 4B
Not many relays are able to be toggled from the 3.3V and support 12V pass through	Found a relay that is specifically marketed towards compatibility with Raspberry Pi's and supports 12V

Conclusion / further planning:

The search for the right relay was somewhat difficult due to most being marketed towards the automotive world and not low power SBC's. In the end we settled on a KY-019 Relay.

by Marius

Date: 2023-07-11

Discussing unloading mechanisms for victims/rescue kit

Tasks done today:

- Creating multiple concepts for victim/rescue kit storage and unloading mechanisms
- Settling on a mechanism like illustrated in Fig. 8

Issues and solutions:

Issues	Solutions
Somehow need to sort, store and unload two alive victims and rescue kit before unloading the dead victim at a different location	Creating three different storage areas with two barriers in which any victim/rescue kit rolls from left to right because of the angle of the whole system, stopped by the barriers (1 and 2). Using one barrier twice reduces complexity and possible failing points of the system as well as reducing the over all size

Conclusion / further planning:

The storage system in Fig. 8 works in the following procedure:

- Close main barrier (2) | Open secondary barrier (1)
- Pick up rescue kit first and sort it in lower storage area (has to be located before any first victim)
- Wait until it slides against the closed main barrier (2) | Close secondary barrier (1)
- Pick-up the victims in any order and sort the alive ones in the upper storage area and the dead one in the lower storage area
- Unload alive victims and rescue kit first by only opening the main barrier (2)
- Unload the dead victim by opening the secondary barrier (1)

This design with only two victims in a row allows us to fit the system above the existing upper panel (see Fig. 5). To finally design it in CAD, however, I need to design the gripper arm first to know its degrees of freedom in sorting the victims into different storage areas.

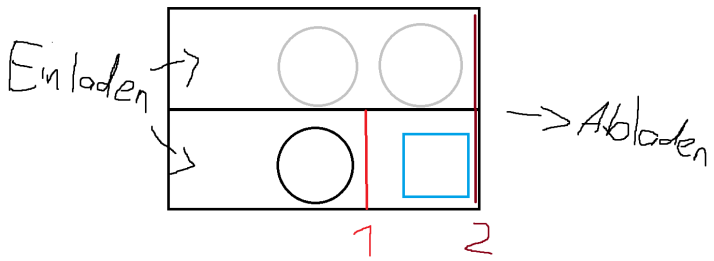


Figure 8: Sketch of final victim/rescue kit unloading mechanism

by Tim

Date: 2023-07-12

First recording of training footage

Tasks done today:

- Attaching the camera we will use for victim detection temporarily in the correct height to a cardboard box (see Fig. 9)
- Testing the camera their image formats for the first time and settling on

```
[0]: 'MJPG' (Motion-JPEG, compressed)
 Size: Discrete 640x480
 Interval: Discrete 0.033s (30.000 fps)
```

- Writing some simple Python code to save videos with the Arducam from our Raspberry Pi 4B using OpenCV (find documentation here)
- Recording videos of different victims in the evacuation zone with different lighting conditions

Issues and solutions:

Issues	Solutions
Chassis of the robot is not finished yet and I am soon on vacation, but Marius needs training data of victims in the evacuation zone to train the AI on them	Using a cardboard box and attaching the Arducam in the correct height from the CAD design to it, I was able to record authentic training footage of victims in the evacuation zone, which I have at home

Conclusion / further planning:

Due to some delays because of design decisions taking longer than expected I sadly was not able to finish the CAD design before my vacation. We are now trying to get as much done as possible during the rest of the summer vacation, but then there will soon be an exam phase until mid-September. This temporary solution was necessary so that Marius could at least continue working during this time.

Figure 9: Temporary recording setup for training footage

by Tim

Date: 2023-07-13

Preparation for AI training

Tasks done today:

- Converted the previously taken videos to image sequences with ffmpeg
- Removed similar, bad quality or blurry images resulting in 618 quality images
- Added a 5x5 blur to remove noise
- Cut off the up most 45% of the image to avoid detection of unwanted objects
- Began labeling the first 130 images with LabelImg, as seen in Fig. 10

Issues and solutions:

Issues	Solutions
A lot of images were very similar due to them being taken from a video	Adding the argument `-vf "select='not(mod(n,5))'"` to FFmpeg so we only convert every 5th frame so we avoid images that are too similar
We think that the AI may have issues with detecting the alive victims due to the reflections	Added a 5x5 blur using OpenCV to every image to avoid "randomness" in the reflections of the alive victims

Conclusion / further planning:

Labeling the first 130 images was quite easy because LabelImg is very simple to use and I developed a rhythm for it. I should be able to get all finished by tomorrow.

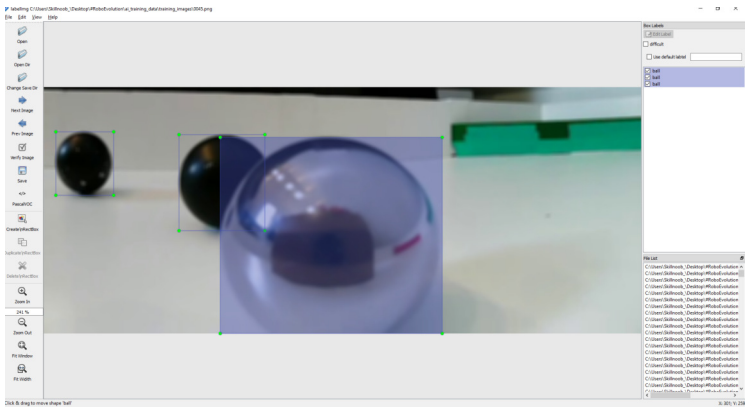


Figure 10: The labeling process with LabelImg

by Marius

Date: 2023-07-14

Training of the first AI

Tasks done today:

- Labeled the remaining 488 images with LabelImg
- Used Roboflow to augmentate and convert the dataset to the YoloV8 format
- Trained the first AI with a Google Colab Nvidia Tesla T4 instance

Issues and solutions:

Issues	Solutions
Difficulties figuring out how to properly train a YoloV8 model	Thanks to the extensive YoloV8 training Docs I could get the first training instance running fairly quickly and resolve all issues

Conclusion / further planning:
The last 488 images only took a few hours to label fully and thanks to Google Colab providing Tesla T4 GPU's for free we were able to quickly train the first model. The next step will be testing and improving this model.

by Marius

Date: 2023-07-15

Training new models for testing

Tasks done today:

- Trained a new AI model using yolov8n
- Trained a new AI model using yolov8m
- Ran benchmarks on both models

Issues and solutions:

Issues	Solutions
YoloV8n has some accuracy issues and sometimes incorrectly identifies dark spots in the background as victims	Training a YoloV8m model with more parameters that eliminates these issues but has performance drops

Conclusion / further planning:
We wanted to test out how the different scales affected the inference accuracy and concluded that even though using the m scale does improve accuracy alot, it's not worth the performance loss so we will continue using the n scale. More information can be found here.

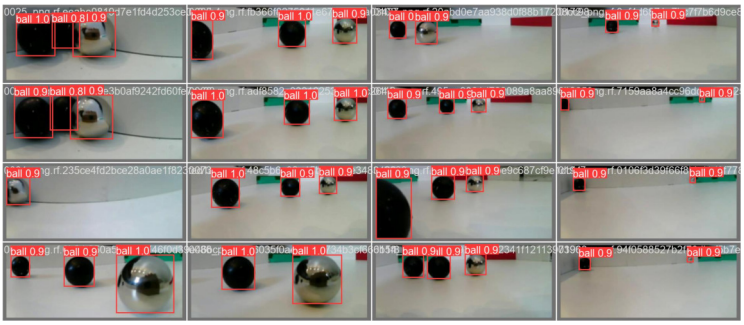


Figure 11: Results for YoloV8n

by Marius

Date: 2023-08-18

Labeling new images and new AI

Tasks done today:

- Captured and labeled a few hundred images in darker scenarios
- Labeled these images with LabelImg
- Trained a new AI with Google Colab

Issues and solutions:

Issues	Solutions
The AI has difficulties detecting the black victims in darker environments because of the victim blending into the background	Capturing a few hundred images in environments with few or artificial lighting to simulate dark scenarios and adding those to the existing dataset

Conclusion / further planning:

After capturing, labeling and training a new AI on these images we saw a great overall improvement in the detection ability of our AI which we are very satisfied with. Especially in dark scenarios.

by Marius

Date: 2023-08-23

Designing the arm to pick up victims

Tasks done today:

- Designed the front plate with cutouts for the camera lens and distance sensor as well as a protruding bar to protect the camera while driving against corners in the evacuation zone
- Designed the side panels with cutouts and mounting places for distance sensors
- Sketched the rough dimensions of the arm that comply with the height restrictions of 25 cm (see Fig. 12)
- Ordered new big (lifting the arm) and small servos (opening and closing the gripper):
  - EMAX-ES09MD Servo (used before and been proven reliable; small powerful servo with metal gears, requires a own power supply instead of usual Arduino pins to function correctly)
  - Diymore 35 kg Digital Servo (used before and been proven reliable; powerful servo with metal gears AND metal potentiometer inside (!), able to rotate nearly 270°, requires a own power supply instead of usual Arduino pins to function correctly)

Issues and solutions:

Issues	Solutions
Maximum height of 25 cm for the arm at highest point while picking up because of potential ramps above the evacuation zone	Designing the arm around these height restrictions
Arm might reach the maximum height of 25 cm in some orientations	Moving the arm further inwards while driving and adding a cutout in the upper panel to support this movement

Conclusion / further planning:

As shown in Fig. 12 the arm reaches in the current rough sketch a maximum height of 24 cm while picking up victims, while driving this height is reduced to only 21.5 cm.

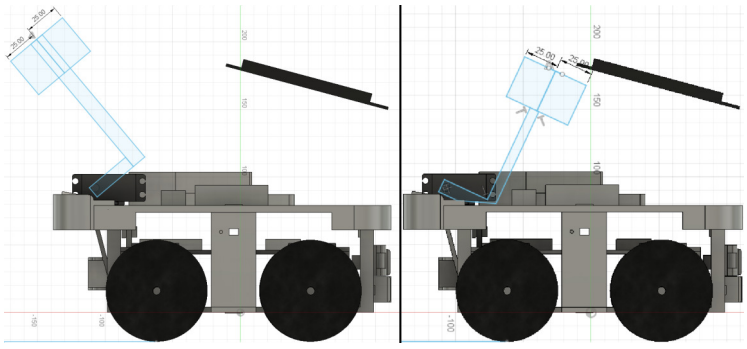


Figure 12: Sketch of the rough dimension of the arm

by Tim

Date: 2023-08-25

Fixed the weird Coral TPU output issue

Tasks done today:

- Alot of testing with the Coral TPU
- Fixed the issue of getting unusable outputs from the TPU already encountered here, (see Fig. 13)

Issues and solutions:

Issues	Solutions
TPU still produces unusable output, so we have no way of extracting the predictions	Using the python library Ultralytics instead, which does all the complicated things automatically

Conclusion / further planning:
Finding out Ultralytics directly supports Coral TPU models and automatically converts their output to a usable format was really the savior here, because the way of installing and running things in PyCoral is very annoying.

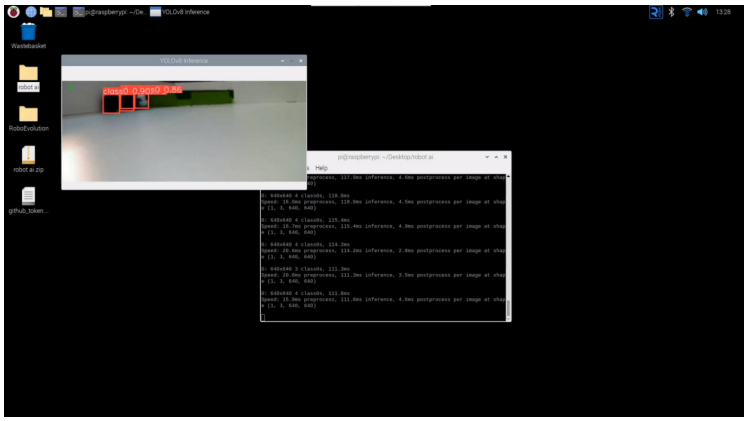


Figure 13: Working Coral TPU

by Marius

Date: 2023-08-30

Changing the group name

Tasks done today:

- Thinking about changing the group name and coming up with names
- Changed the group name to OverEngineering^2 and all relevant account names

Issues and solutions:

Issues	Solutions
Come up with a good team name	Asking friends and our teacher for some advise

Conclusion / further planning:
We didn't like our old name (#RoboEvolution) anymore as it felt outdated to us. We thought about some different new names like Pathfinder, IndexError, Race to Rescue, Bit Overflow and OverEngineering. In the end we settled on the name OverEngineering^2. The squared is because our robot is extremely over engineered.

by Tim & Marius

Date: 2023-09-03

First design of gripper

Tasks done today:

- Sketching some gripper blades that fit victims as well as the smaller rescue kit (see Fig. 14)

Issues and solutions:

Issues	Solutions
The rescue kit is smaller than most of the victims typically used, and the victims can vary in size (4 cm - 5 cm in diameter)	Choosing gripper blades that are curved in a way that fits all these objects well
Not sure about what diameter to choose for the gears connecting the gripper blades	Sketching gears with different diameters, turning the gripper blades around that circle within the sketch, and in the end, taking the one that seems to fit best

Conclusion / further planning:
After some discussion and multiple iterations of sketches, we decided to settle on the design in Fig. 14 on the right.

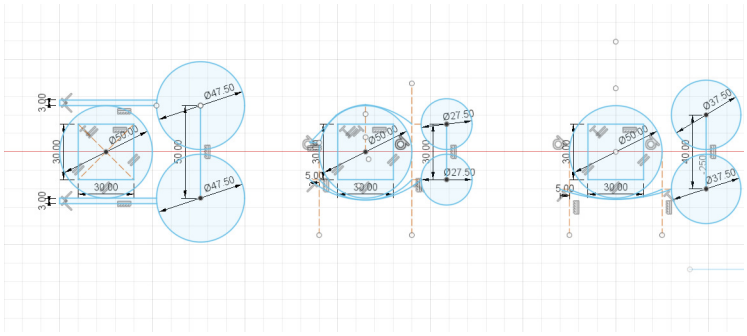


Figure 14: Sketches of different gripper blades and gear diameters

by Tim

Date: 2023-09-06

Finishing design of gripper arm

Tasks done today:

- Designed an enclosure to fit the servo for opening the gripper blades and a IR-distance-sensor
- Added a third servo to the arm, allowing rotation of the gripper to sort victims into different storage areas and serving as the attachment point between gripper and arm (see Fig. 15)
- Added small edges to the blades of the gripper

Issues and solutions:

Issues	Solutions
Need to approve whether a victim pickup was successful or not	Adding a short distance IR-distance-sensor in the center of the gripper blades to measure the distance to the closed blades thus determine if a victim is in between or not
Victims might slide out of the gripper blades while lifting	Adding small edges to both sides of the blades to prevent victims from moving at all
It is difficult to imagine, for example, how a component attached to a servo rotates	Using joints as relationships between components, you can easily simulate movements within given degrees of freedom using Fusion 360

Conclusion / further planning:

With the design of the gripper arm finished, I can finish designing the victim storage compartment within the next week, so we can finish the whole design shortly after that and finish printing the robot chassis before the start of our autumn holidays, spending them finally assembling the robot.

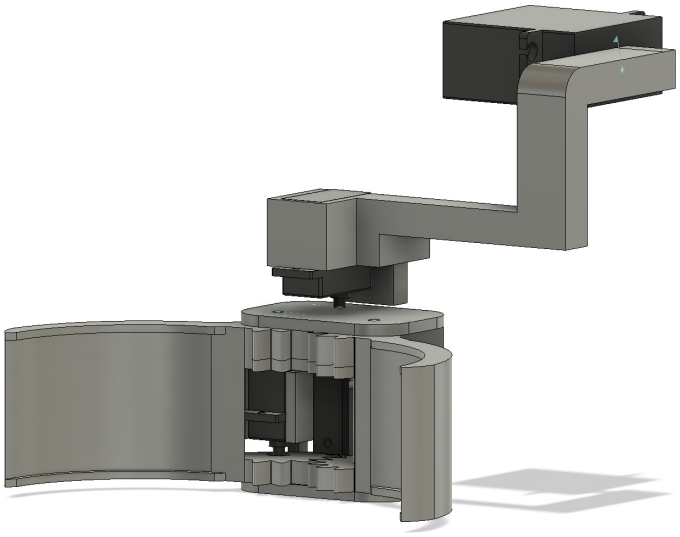


Figure 15: Finished CAD design of gripper arm

by Tim

Date: 2023-09-13

Designing the storage compartment for victims/rescue kit

Tasks done today:

- Added Coral TPU to upper panel
- Added two simple barrier attached to the small servos recently ordered
- Designed a mounting bracket for display on top of storage compartment
- Designed four pillars to support the storage compartment at the desired angel of 15°
- Added cutout for secondary barrier to storage compartment

Issues and solutions:

Issues	Solutions
Storage compartment requires much material to print	Adding cutouts acting as "windows" to save some material
Screws below the storage compartment could no longer be reached	Adding holes big enough to fit a screwdriver to the storage compartment
Opened main barrier currently intersecting with the upper panel	Cutting out a section of the upper panel

Conclusion / further planning:

After finishing the design of the storage compartment as seen in Fig. 16 there are only a few parts missing to finally finish the design and start printing.

We are not currently printing any part, as in the past there have often been changes where more parts than just the one currently being worked on had to be changed.

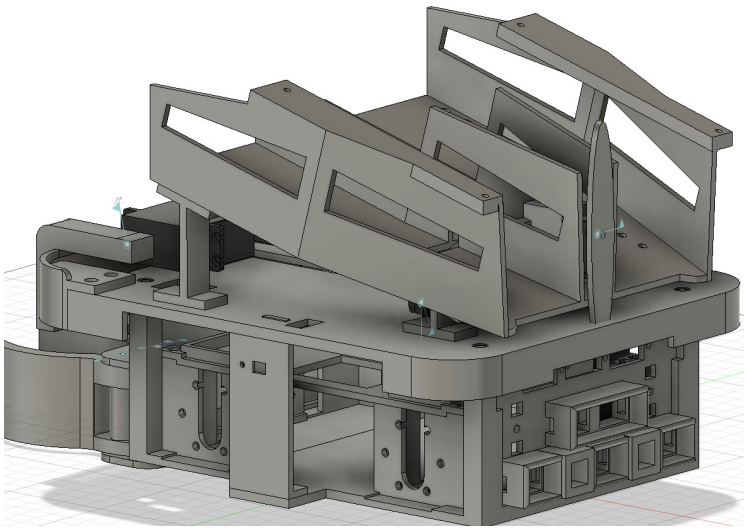


Figure 16: Finished CAD design of storage compartment

by Tim

Date: 2023-09-14

New wheels and new AI model

Tasks done today:

- Assembled new wheels to act as a backup
- Trained another AI model with 300 epochs on Google Colab
- Testing and benchmarking the new model

Issues and solutions:

Issues	Solutions
We are worried that our 3 year old neoprene wheels may fail in the future or break for a unknown reason	Put together a set of 4 brand new neoprene wheels with hubs as a replacement set if something happens
Accuracy issues with our yolov8n models. Quite often they detect the background as balls	Training a new model with 300 epochs instead of 200, which may increase accuracy and prevent false positives

Conclusion / further planning:

We had some new neoprene wheels left over, so we decided to assemble them so that we have a spare backup. After testing the new model, we conclude that it did not bring any significant improvement and the increased training time is not worth it.

by Marius

Date: 2023-09-14

Finding the right position for the gyro sensors

Tasks done today:

- Added and arranged missing components on the upper panel
- Noticed there was not any space left for the gyroscope sensors on the upper panel
- Added the gyroscope sensors to the side plates

Issues and solutions:

Issues	Solutions
No more space left on the upper panel to fit two horizontal BNO055 sensors	Adding them vertically to the side plates despite some problems with vertical sensors in the past

Conclusion / further planning:

After finding no different place for the two BNO055 sensors than vertically on the side plates (see Fig. 17), we hope that the problems encountered in the past are easy to fix.

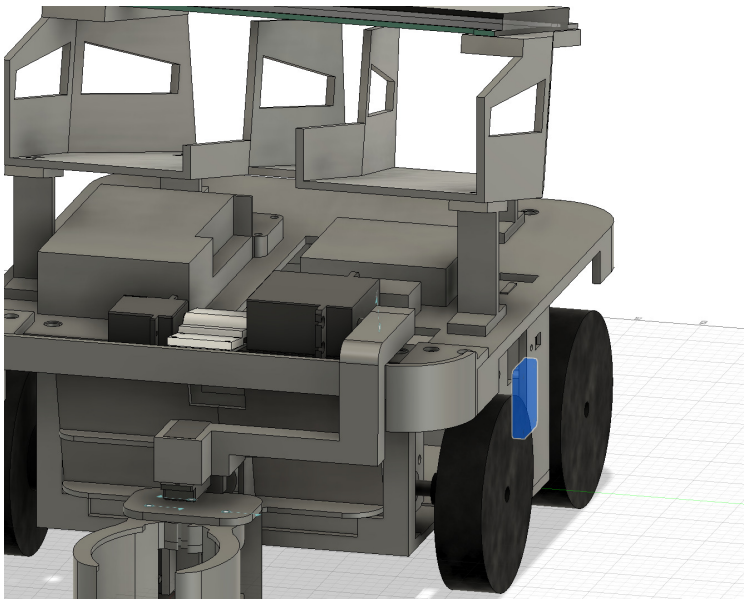


Figure 17: Gyroscope sensors positioned on the side plates

by Tim

Date: 2023-09-16

Finishing design of the robot

Tasks done today:

- Adding final touches such as:
  - victims
  - improving pillars to fit under the shorter side of the victim storage compartment
  - adding recesses for screw heads
  - clean up Fusion 360 project with adding descriptive names to sketches, bodys, and components
- Exporting all parts as STL files to move them into a slicer

Issues and solutions:

Issues	Solutions
Can easily export every body as an individual STL file when exporting the whole project	Exporting all body's individually by exporting the whole project while hiding everything except the body to export

Conclusion / further planning:

With a completely finished design of the whole robot, we can finally start printing. We will probably split up the printing between my Ender 3 V2 printer at home, a SNAPMAKER A350T at school, and maybe another SNAPMAKER A350T from a friend of Marius. Using all of these printers, the robot should be printed within a week or so.

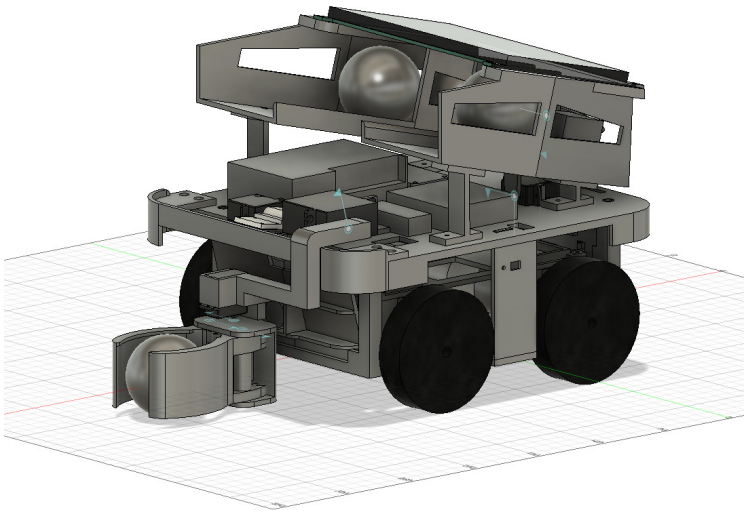


Figure 18: CAD design of finished robot

by Tim

Date: 2023-09-17

Training of new models

Tasks done today:

- Trained a new AI model with a image size of 448 px
- Tested and benchmarked the new model

Issues and solutions:

Issues	Solutions
We only got around 6 FPS with a model running on a image size of 512 px, which is no where near what we need	Training a new model with a much lower image size of 448 px and running it with a image size of 448x192 px made it possible to get 19-21 FPS using the edge TPU.

Conclusion / further planning:

I am very happy with the results of the 448 px model, as we are getting much more FPS now and after some testing I can say that the accuracy decrease is barely noticeable.

by Marius

Date: 2023-09-18

Printing the new Robot

Tasks done today:

- Leveled the heatbed and did some quick test prints to test the new filament
- Starting the first 19 hour printing session with printing the robots base plate on my Ender 3 V2 at home

Issues and solutions:

Issues	Solutions
The greatest load on the part will probably occur in the area of the motor mounts	Adding 100% infill in those areas in PrusaSlicer and using eSUN PLA+ as an easy to print PLA with improved toughness

Conclusion / further planning:

As the printer at school isn't working properly at the moment and Marius unfortunately couldn't use the other printer, I'm going to print most of the parts at home for now. We'll have to wait a few days until the printer at school is fixed to print the victim storage compartment, as this is the only part that doesn't fit on my Ender 3 V2.

by Tim

Date: 2023-09-18

Trained a new larger model

Tasks done today:

- Trained a yolov8s model, instead of the usual yolov8n
- Tested the bigger model with the Coral TPU

Issues and solutions:

Issues	Solutions
After many attempts to improve the accuracy of our yolov8n model, it still has many issues detecting things half way across the zone	Switching to using a yolov8s model with more than 3 times the parameter count improved accuracy greatly, while still running at 15 FPS

Conclusion / further planning:

The choice of switching to a yolov8s model was definitely the right decision, because it is now able to detect victims from much further away. More detail on the different model sizes can be found here.

by Marius

Date: 2023-09-19

Warping while printing the base plate

Tasks done today:

- Removing the print from the heatbed, discovering fairly strong warping of the print
- Did a few more test prints (find test model here) with higher heatbed temperature
- Starting anther try on printing the base plate with a higher headbed temperature

Issues and solutions:

Issues	Solutions
Printing with the new eSUN PLA+ resulted in fairly strong warping of the part especially at one corner	Reprinting the part with a higher headbed temperature of 80°C worked well on the test prints

Conclusion / further planning:

Deformation of 3D printed parts (as seen in Fig. 19) usually occurs when the adhesion of the part to the build plate is too weak to withstand the force of the cooling, i.e. shrinking, layers above the first layers, causing the first layer to detach from the build plate and be pulled inwards. A possible solution for this problem is increasing the headbed temperature to improve adhesion as well as preventing the upper layers from cooling down at the same time.

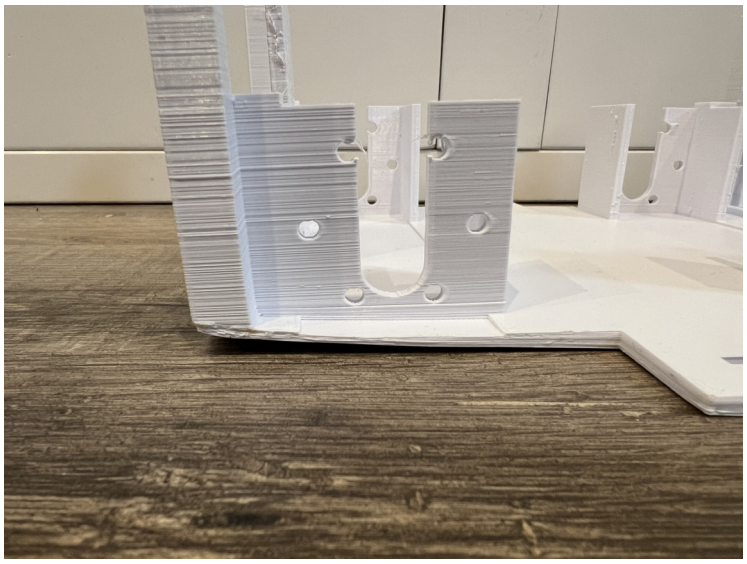


Figure 19: Warping while printing the base plate with eSUN PLA+

by Tim

Date: 2023-09-20

Adding handles to the robots design

Tasks done today:

- Noticed that our design is missing the handle required by the rules, therefore added two handles to both sides of the victim storage compartment as seen in Fig. 20
- Successfully finished the reprinted base plate without any warping this time
- Started printing the battery compartment fitted above the base plate

Issues and solutions:

Issues	Solutions
Robot is missing a handle required by the rules	Adding two handles to the design without interfering with any parts except from the upper panel

Conclusion / further planning:

Luckily there was enough space left on the upper panel of the robot to add some handles without editing any part except for the upper panel to add some mounting holes.
To improve strength, we will print the handles with 100% infill and lay them on their side so that the layer lines are opposite to the force exerted on the handles.

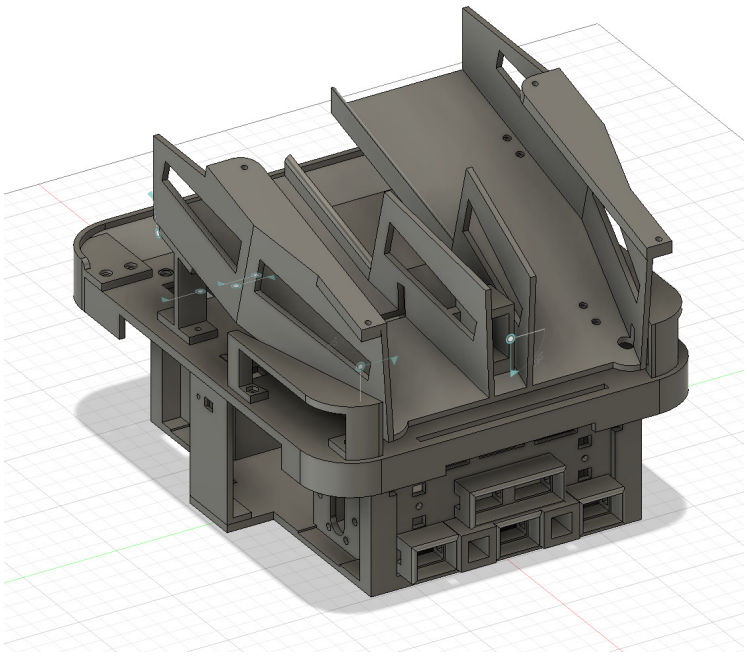


Figure 20: CAD design with added handles

by Tim

Date: 2023-09-21

Assembling first parts

Tasks done today:

- Finished printing of battery compartment and back plate
- Fitted all the components to the base plate for the first time
- Started printing the upper panel

Issues and solutions:

Issues	Solutions
Arrangement of the step-up and step-down converters as well as the h-bridge was not done in CAD	Arranged them in such a way shorting two contacts next to each other won't have major consequences for safety reason

Conclusion / further planning:

After we saw that the first components fit well into the printed parts, we were able to continue printing without any worries.

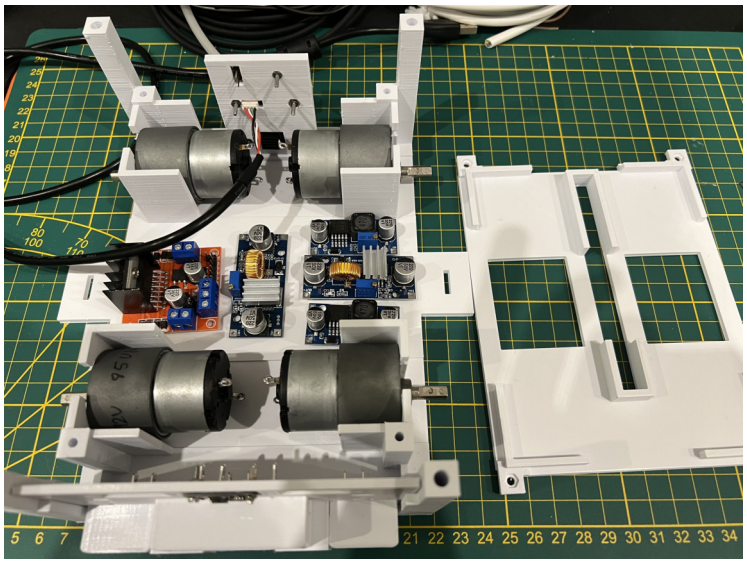


Figure 21: Fitting components to the base plate for the first time

by Tim

Date: 2023-09-22

Removing support material from bad print

Tasks done today:

- Printed the victim storage compartment on the SNAPMAKER A350T at school
- Printed some remaining parts like handles, pillars, barriers at home
- Made a jumper wire with three connectors myself using pliers, connectors and some solder

Issues and solutions:

Issues	Solutions
The Victim Storage Compartment won't fit on my Ender 3 V2	Printing it on the repaired SNAPMAKER A350T at school with a build plate size of 350 mm x 350 mm
Choosing bad support settings in the slicer of the printer at school	Needing to tediously remove them over the course of two hours using pliers, side cutters and a dremel
Individual jumper wires have often fallen off or become loose in the past	Make your own jumper wire with the required number of connectors (the more connectors, the less likely they are to become loose)

Conclusion / further planning:

After the tedious removal process of the poorly configured support material, as shown in Fig. 22, most parts of the robot are finally printed so that I can start assembling them and solder the internals of our new robot. During this time, Marius tests and writes the Arduino code needed to control and read out all the sensors.

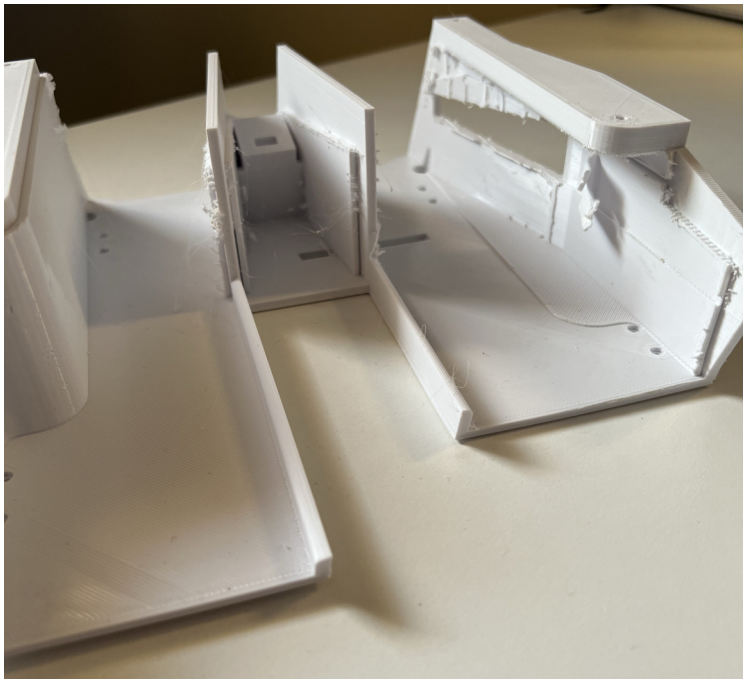


Figure 22: Removal process of poorly configured support material

by Tim

Date: 2023-09-23

Fixing the gyro shift

Tasks done today:

- Adding support for a 2nd gyro sensor

Issues and solutions:

Issues	Solutions
The gyro sensor we currently use undergoes a shift of approximately 5° in the X-axis after a specific duration, causing challenges in parkour as the PI relies on its stability	Attaching an additional gyro sensor to the Arduino and altering its address to 0x29 (found out how here) to enable the use of a 2nd sensor. The new sensor enables the PI to compensate for the values during its operation

Conclusion / further planning:

After often having reliability issues with the gyro sensor as it would shift randomly by around 5° in a random direction and having to reset it every time a lack of progress occurs, we added a 2nd sensor to average out the rotation in the opposite direction. This should in conclusion improve the orientation skill of the robot by not having to rely on unreliable factors.

by Marius

Date: 2023-09-25

Using self made jumper wire

Tasks done today:

- Spend a whole day making self made jumper wires out of high quality thin solid core wire for all necessary connections between Raspberry Pi and the motor driver as well as the Arduino Nano with all six distance sensors and the two gyroscope sensors

Issues and solutions:

Issues	Solutions
In the past, individual jumper wires have often fallen off or become loose	Make your own jumper wire with the number of connectors you need
Self made wire sometimes slides out of the connector	Using a drip of solder on each connector to secure the wire to the connector before squeezing the connector using pliers

Conclusion / further planning:

As shown in Fig. 3, six pins of the Raspberry Pi must be connected to the motor driver to control the motors, and each of the six distance sensors on every side of the robot needs three pins connected to the Arduino Nano using a shield for an increased number of pins. Using self made jumper wire not only lowers the probability of them getting loose but also makes it easier to connect and disconnect components because not every cable has to be connected individually. The connected connectors also prevent errors when reconnecting components, with a set like the one in Fig. 23, you can easily find a connector with the right number of sockets.

Figure 23: Various connectors for self-made jumper wires

by Tim

Date: 2023-09-27

First school intern presentation of unfinished prototype

Tasks done today:

- Assembling previously printed part and adding some components like the USB camera, motors and wheels
- Presented the unfinished prototype to some sponsors and our principal
- Discussed possible issues especially about our chosen dimensions of the robot with other members of the robotics club

Issues and solutions:

Issues	Solutions
Developing and building a vision based robot gets very expensive, especially when testing new components which might turn out to be pointless	Keeping sponsors informed about the value of their support and show them regular successes

Conclusion / further planning:

Although the robot is still far from being ready to drive, it is already easy to see what the robot will look like one day (see Fig. 24). Through meetings with our sponsors like this one today, we can provide them with updates on our current progress and demonstrate the impact of their support. Without the invaluable support of our sponsors (who can always be found here), this project would not be possible for us. Thank you for your continued support!

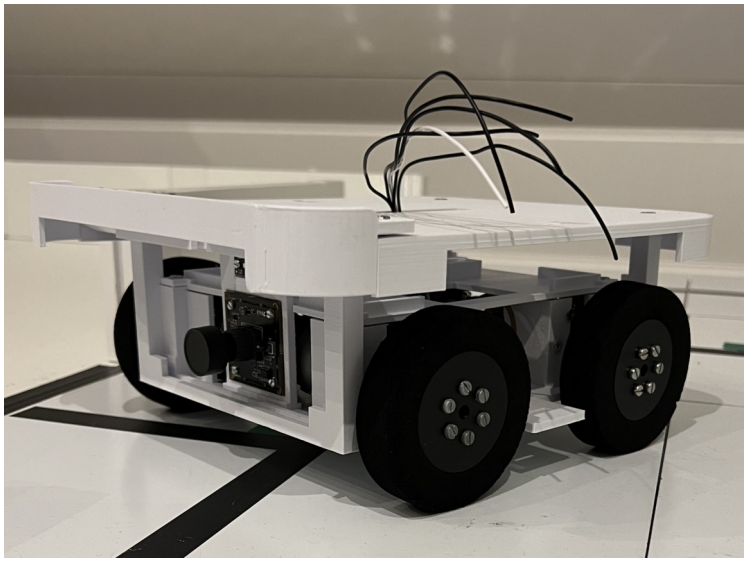


Figure 24: Unfinished prototype of new robot

by Tim

Date: 2023-09-28

Implementation of new camera module

Tasks done today:

- Added support for Camera Module V3 in the program

Issues and solutions:

Issues	Solutions
The Camera Module V3 isn't functional due to incompatibility issues between the picamera module and the newer camera, and there's also a lack of compatibility with OpenCV	The picamera2 module emerged as the solution: initially, using the wrong capture format caused a hue shift. However, after implementing picamera2, it seamlessly functioned with the AI and later with the line recognition program

Conclusion / further planning:

It took quite a while to get the new camera working well. The entirely different API from the picamera2 module proved incredibly powerful but also difficult to get used to as some basic functionality was hidden away and took a lot of documentation reading to get used to. In the end the new camera works well and the improved sensor of the V3 module produces some great looking images, result seen in Fig. 25.

We also found out that the Raspberry Pi 5 is confirmed to be released through this video and are very exited for it, mostly because of the promised performance increase of up to 3x. That would enable us to run at higher frame rates and then run the robot faster. We are planning to buy it as soon as it is available, probably somewhere around new year.

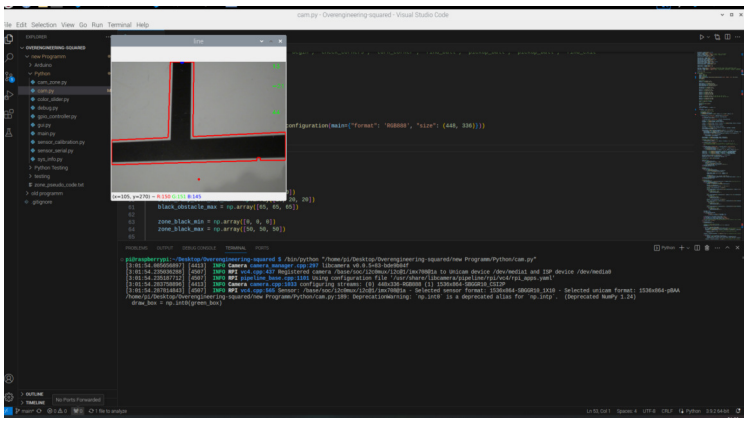


Figure 25: The first capture with the new camera

by Marius

Date: 2023-10-01

Finished soldering most of the internals Self Made Jumper Wires

Tasks done today:

- Spend the last few days cutting wire to length and soldering all the connections as shown in Fig. 3
- Integrated a switch to disconnect each LiPo battery individually from the whole circuit, switch to turn on/off the LED's later, switch to disconnect the motor diver from any power and a switch with a soldered resistor to connect it to the Raspberry Pi later
- Integrated two fuses for each LiPo battery as well as a digital voltmeter for each battery to determine their state of charge
- Connected the self made jumper wires to the motor driver
- Connected some self made jumper wires to create a "5V hub of connectors" to connect all servos directly to the step-down converter later

Issues and solutions:

Issues	Solutions
Stripped contacts could possibly touch each other and lead to a short circuit	Always using shrink tubing on stripped contacts to prevent this
Not much space inside the robot	Using cables that are as short as possible to take up less space

Conclusion / further planning:

Soldering all the cables and squeezing them under the battery compartment took some time, especially with the solid core jumper wires (see Fig. 26). Nevertheless, i am confident that this time i was able to avoid most of the mistakes during soldering as described in Fig. 2.

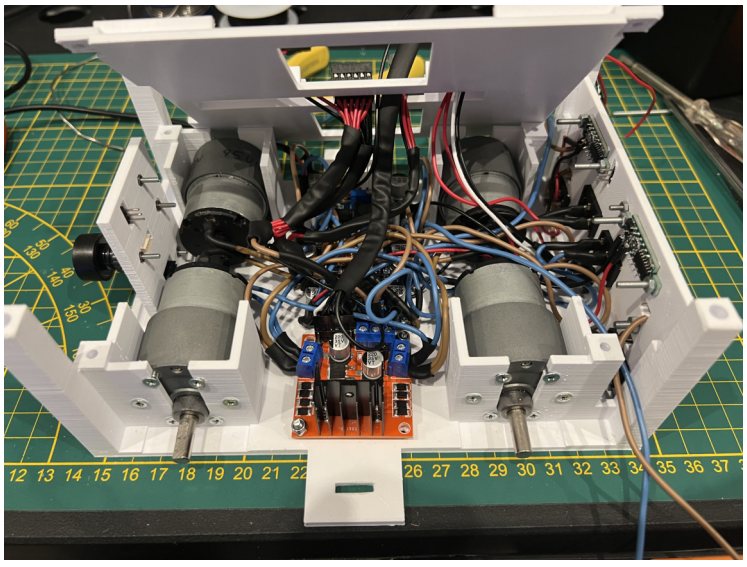


Figure 26: Fully soldered internals

by Tim

Date: 2023-10-02

First tests with LiPo's

Tasks done today:

- Added LiPo's to the circuit the first time
- Replaced a blown ground cable
- Printed the gripper mechanism as well as the LED mount

Issues and solutions:

Issues	Solutions
An already stripped ground cable has touched the exposed metal of one of the fuses causing a short circuit and the ground cable to burn out	Replacing the damaged cable and making sure to only strip any cables if they get connected shortly thereafter in the future

Conclusion / further planning:

Except of one mistake with an already stripped ground cable the first power tests were flawless. An image with working voltmeters is provided in Fig. 27. Now we can continue adding the upper panel with all of its components.

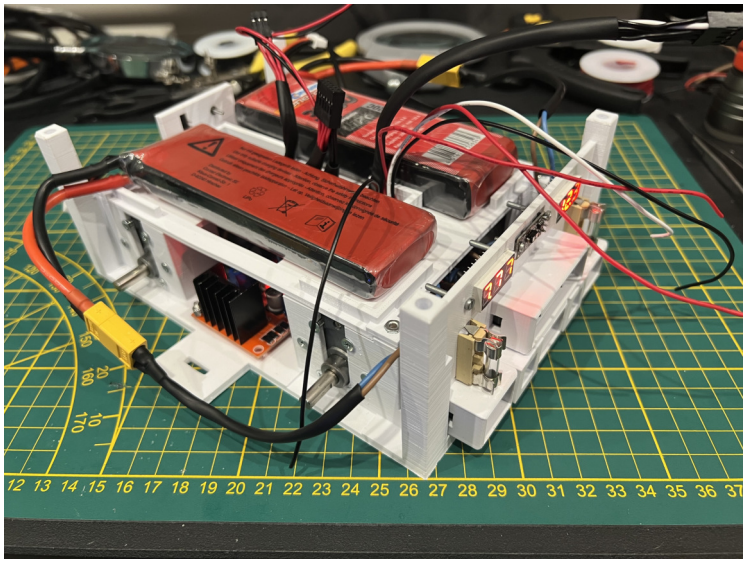


Figure 27: First power test all the internal components

by Tim

Date: 2023-10-03

Fitting all necessary parts on top of the robot

Tasks done today:

- Assembled the gripper mechanism with one EMAX-ES09MD servo and one 50 cm Pololu Irs16a inside
- Added the LED mount and arranged all necessary components on the upper panel

Issues and solutions:

Issues	Solutions
The new Raspberry Pi 5 has a different arrangement of the USB ports on the front than the Raspberry Pi 4B, so the arrangement of the components we created in CAD no longer works	Rearranging all the components, this time using a Raspberry Pi 3B+, because its USB port layout is the same as it will be on the Raspberry Pi 5

Conclusion / further planning:

Unlike what was planned in CAD, we had to change the arrangement of all the components on the upper panel, as shown in Fig. 28. Next we have to assemble the victim storage system, the gripper arm, add LED's to the LED mount and finally connect all the components.
While these are my tasks, because i have the robot at home, Marius is already able to start improving our old program.

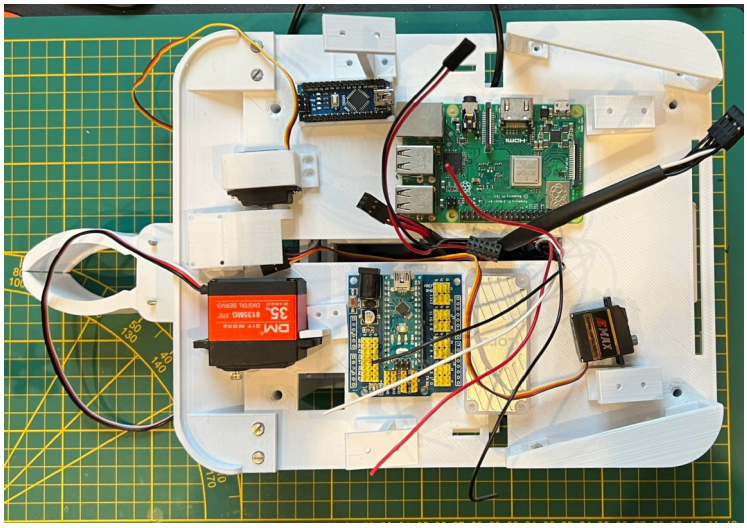


Figure 28: Rearranged components on the upper panel

by Tim

Date: 2023-10-06

Optimizing the program to gain FPS

Tasks done today:

- Finding time intensive in the program using line_profiler
- Rewriting these code sections and jit compiling them with Numba

Issues and solutions:

Issues	Solutions
The camera script gets about 6-7 FPS while running with multiprocessing which is not enough to drive at acceptable speeds. It can also lead to the Program missing or noticing changes in the line too late and having the to slow the robot down which can lead to time problems	Using the Numba library (a jit compiler for python) to compile certain pieces of code to native code which doesn't use the python interpreter. This is helpful since the python interpreter is extremely slow and C/C++ code runs about 80 times faster. Additionally Numba can handle NumPy very well providing another speedup. Optimizing some code and putting a part of it into Numba gained a FPS

Conclusion / further planning:

Using Numba to compile parts of the code down to native code did bring us a few FPS at the cost of the program taking longer to start up, but the entire program needs a full rewrite where we will focus more on performance. For the future we plan to upgrade the Raspberry Pi 4B with 8GB of ram to a Raspberry Pi 5 with 8GB of ram once it releases on the planned date of 23.10.2023. The Version 5 Pi has a approximately 2-3x CPU speedup over the 4B and a 4x GPU speedup. Since the Raspberry Pi 5's operating system is based on Debian bookworm, it will have Python 3.11.x unlike Raspberry Pi Os Bullseye which ships with Python 3.9.x which is not up-gradable to 3.11.x unless you use something like anaconda which creates more overhead. That is because Python 3.11.x has a 10-60% speedup over Python 3.10.x and presumably a even larger one over 3.9.x.

by Marius

Date: 2023-10-08

Assembling victim storage system

Tasks done today:

- Cleaned up left over support from the servo bracket of the main barrier of the storage system because the EMAX-ES09MD servo won't fit otherwise

Issues and solutions:

Issues	Solutions
Forgot to add ports under the display to the CAD model, causing the display to not fit on the bracket	Adding M4 nuts on M3 screws as spacers simply solved this problem

Conclusion / further planning:

Today's progress doesn't seem much (see Fig. 29), but removing the excess support material has once again taken much longer than expected.

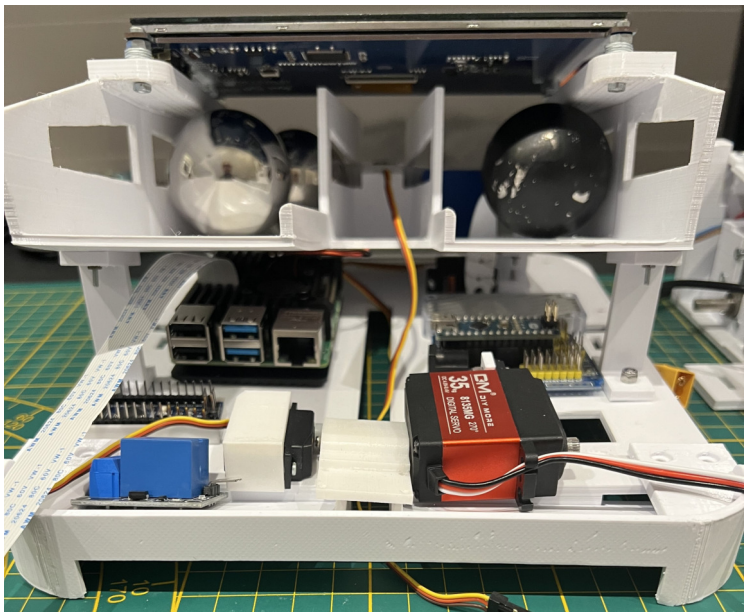


Figure 29: Added fully assembled victim storage system to the upper panel

by Tim

Date: 2023-10-10

Printing new red/green corners

Tasks done today:

- Started the print of a red and green corner with the school's 3d printer

Issues and solutions:

Issues	Solutions
The current red and green corners we have in school are in a bad shape and consist out of a black corner with tape on it	3D printing new corners with red and green filament and much better quality

Conclusion / further planning:
We needed some better corners, because our current ones were just the old ones with colored tape on them.

by Marius

Date: 2023-10-14

Realising there will be no rescue kit

Tasks done today:

- Finding out that in 2024 there will be no rescue kit
- Adjusting our rescue plans

Issues and solutions:

Issues	Solutions
Collection system is built onto the fact that there is a rescue kit and 3 victims	Luckily it our collection system will still work with some small adjustments in the code for how the dead victim gets placed inside

Conclusion / further planning:
We were shocked that the organisers would suddenly decide to remove the rescue kit, especially because many teams had modified their robots a year before, expecting the rule change to last and now have to modify them again.

by Tim & Marius

Date: 2023-10-15

Testing CustomTkinter as GUI

Tasks done today:

- Created the first test iterations of a new GUI using CustomTkinter

Issues and solutions:

Issues	Solutions
Our old GUI, which is mainly built with OpenCV, is very cluttered and doesn't look very good	Building a new GUI with CustomTkinter, which looks alot better and is hopefully also more performant

Conclusion / further planning:

We thought of using PyQt5 before, but it is very complicated compared to CustomTkinter and requires a lot of work to not look like a GUI which was made during the Windows XP era. The first markups of the GUI look much better than the old one already, like in Fig. 30.

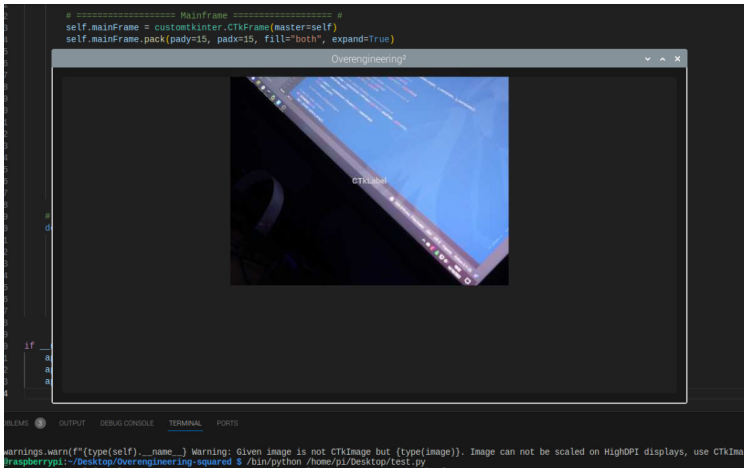


Figure 30: The first working GUI

by Marius

Date: 2023-10-16

Creating a layout for the new GUI

Tasks done today:

- Gathered a list of requirements for the new GUI
  - Camera streams of both cameras
  - Real time sensor data of at least eight sensors (seven distance sensors and one averaged rotation value from both gyroscope sensors)
  - An illustration of the evacuation zone with the robot's current position, the detected red and green corners, and the entry and exit points.
  - Free space to add more buttons and our logo later
- Drew a Layout sketch for the new GUI in Microsoft Paint (see Fig. 31)

Issues and solutions:

Issues	Solutions
Ended up with some free space between the sensor data frame and the evacuation zone frame	Decided to add a real time rotating model of our robot based on the data from the gyroscope sensors without any function just for aesthetic reasons

Conclusion / further planning:

We have ended up with a pretty simple design for our GUI as you can see in Fig. 31. Such a simple design is easier to create and offers plenty of room for additions later on. Buttons and our logo will later be added to the gray bar at the bottom and the free space between the two camera frames.

The rotating model of the robot should be fairly easy to add based on a STL file we can export from Fusion 360.

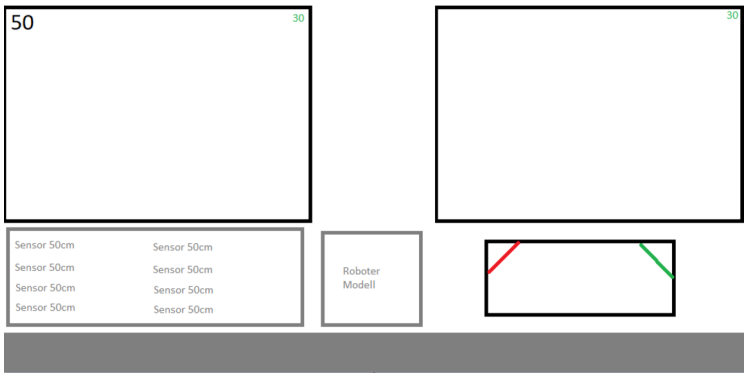


Figure 31: Layout sketch for new GUI

by Tim

Date: 2023-10-17

Mounting the LED strips

Tasks done today:

- Tested how much material I could cut off the LED strips before they stopped working
- Trim as much material as possible from the LED strips and cut to length
- Soldered new connections between the pieces of the LED strip

Issues and solutions:

Issues	Solutions
Less space for LED's on the new mount that on the old mount	Cutting off as much useless material as possible to fit the most amount of LED's possible
Soldering the wires between the contacts of the LED strips would often cause them to come loose, which would turn off the light strips on the old robot	Bridging two connection using just solder in the places where it was possible

Conclusion / further planning:

With this technique we have maximized the amount of LED's that can fit on the mount, we will see if this new light bar is as effective as the old one when we test everything later. However, it should definitely be much more reliable than before because the plastic edge around the mount protects the cable connections from direct impacts (see Fig. 32).

Figure 32: Finished LED bar

by Tim

Date: 2023-10-18

Assembling the whole robot for the first time

Tasks done today:

- Finally assembled all the parts like arranged in Fig. 29 (until the Raspberry 5 arrives, we are using our old Raspberry Pi 4B) and connected all the cables
- Did a lot of cable management, especially the stiff solid core jumper wires were once again a problem in this regard
- Attached and connected all of the distance sensors

Issues and solutions:

Issues	Solutions
During cable management one of the self made jumper wires has came loose connected to one of the distance sensors	Disassembling some part of the robot again to get to the disconnected cable and soldering the wires of each sensor directly to the sensors
While I was wiggling a distance sensor in the front plate, because the LED indicating something was detected in front occasionally malfunctioned, the sensor suddenly stopped working completely	After disassembling part of the robot again and removing the broken sensor, I discovered that the sensor was missing a resistor, which caused it to malfunction. Simply replacing the sensor solved this problem for now

Conclusion / further planning:

Even with the discovered weak spot of an easily broken resistor on the Irs17a (see Fig. 33), the general problem of incorrectly detecting the front panel as an obstacle is not yet solved.
I will need to actually read the sensor data, which is the next goal, to investigate this issue further.

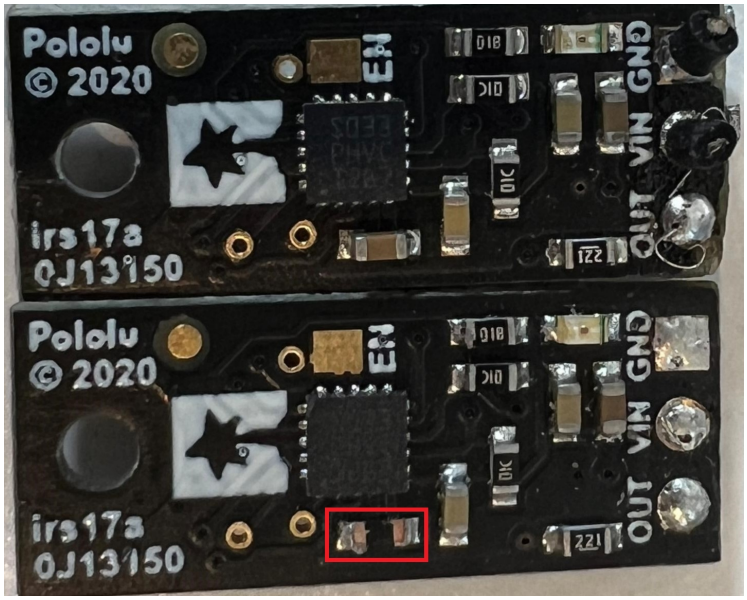


Figure 33: Broken Irs17a sensor on the bottom; undamaged sensor on the top for comparison

by Tim

Date: 2023-10-19

Improving old Arduino scripts & delay issue with Irs17a

Tasks done today:

- Added function based support to the Arduino for reading and printing seven distance sensors and two gyro sensor readings (the Arduino and Raspberry Pi are connected via USB enabling simple print and read serial communication as shown in Fig. 34)
- Discovered very slow sensor input readings by the Arduino as seen in Fig. 34
- Among other things, I tried to solve this problem by adding a timeout according to the documentation, as follows

  ```
  pulseIn(pin, value, timeout)
  ```

  Like the others, this attempt did not work and only resulted in the sensor no longer being recognized
- Ordered a new set to make your own jumper wires with a special crimping tool this time as well as new more flexible wire

Issues and solutions:

Issues	Solutions
If no distance sensor is detected on the given pin (indicated by the sensor index "S*" follow by "No") the function `pulseIn(pin, value)` seems to cause a delay of up to 0.7 seconds as seen by the timestamps in the Arduino's serial output	Ensure that each distance sensor is securely connected, as loss of connection would result in each distance sensor, as well as the gyro sensors, becoming unusable
Connection to the distance sensors often breaks off because the self made, much too stiff jumper wire broke	Reluctantly deciding to redo all jumper wires, this time with much more flexible cables and a special pliers (whose existence we only recently discovered)

Conclusion / further planning:

After discovering huge delays in case of a lost connection to just one of our distance sensors, a reliable connection to them became even more important than before, which finally led to the decision to do the work of disassembling the robot again and redoing all the jumper cables to be more flexible and reliable.

After redoing all the jumper wires soon, we will software-wise focus on finishing the GUI fist which will hopefully help us massively debug the actual Rescue Line code.

```
01:31:42.675 -> G1 X: No Y: No AX: No AY: No
01:31:42.675 -> G2 X: No Y: No AX: No AY: No
01:31:42.863 -> 1
01:31:43.564 -> S1 No
01:31:44.265 -> S2 No
01:31:44.918 -> S3 No
01:31:45.615 -> S4 No
01:31:46.314 -> S5 No
01:31:47.011 -> S6 No
01:31:47.011 -> G1 X: No Y: No AX: No AY: No
01:31:47.011 -> G2 X: No Y: No AX: No AY: No
01:31:47.198 -> 1
```

Figure 34: Arduino serial output with huge delays

by Tim

Date: 2023-10-20

Redoing all the self made jumper wires & Raspberry Pi setup

Tasks done today:

- Redoing all the jumper wires took about eight hours (which was way faster than expected because of the new special crimping tool)
- Installed the new Raspberry Pi OS Bookworm, which had trouble detecting our display, so after a lot of debugging I gave up and switched back to Raspberry Pi OS Bullseye for now which previously worked with our display by adding

  ```
  hdmi_cvt:0=1024 600 60 3 0 0 0
  hdmi_group:0=2
  hdmi_mode:0=87
  ```

  to the raspi-config.txt file.
- Cloned our GitHub repository, disabled legacy camera with raspi-config to get the Camera Module V3 working properly
- Updated the old Python serial read script to use a higher baud rate of 115200 and store data from seven distance sensors without any problems (and make the data accessible in any multi-process by using (multiprocessing.Value())
- Tested the old motor driver script as well as our old Rescue Line code, which worked fine except for the USB camera (probably because I am currently using mostly the same setup as on the old robot)

Issues and solutions:

Issues	Solutions
Our display does not work with the new Raspberry Pi OS Bookworm installed on the Raspberry Pi 4B	Trying to fix this problem by, among other things, following this very helpful article, but eventually giving up and using the old Raspberry Pi OS Bullseye

Conclusion / further planning:

Except for some problems with new software and components, the robot now works similarly to our old robot except for rescuing victims. Fixing the problems with the new software and camera support is not worth it for us now, because we are planning to switch to the Raspberry Pi 5 soon anyway, where these fixes might not be needed or work.
The goal for now is to get the GUI to a stage where we can integrate it into our existing code.

by Tim

Date: 2023-10-22

Adding a grid system to the GUI

Tasks done today:

- Add a grid system for precise widget alignment by using

```
self.grid_columnconfigure()
```

- Split the GUI in multiple frames by using

```
customtkinter.CTkFrame(master=self)
```

- Try to buy the new Raspberry Pi 5, but failed because of issues with the dealer
- Add buttons and status indicators to the GUI

Issues and solutions:

Issues	Solutions
Camera frames won't align as intended due to their fixed larger size	Add weights in the form of weight=2 to the first and last columns to accommodate for their larger size

Conclusion / further planning:
Particularly, the reorganization achieved by dividing the GUI into multiple frames is crucial for maintaining better organization. The result looks great so far (see Fig. 35), but we need to add more buttons, display actual data, and add support for displaying two different camera streams at the same time to make the GUI really useful.

Figure 35: GUI with added grid system

by Tim

Date: 2023-10-23

Rotating STL file within GUI

Tasks done today:

- Try to add a real-time rotating model of our robot into the GUI

Issues and solutions:

Issues	Solutions
The python Tkinter package lacks support for 3D rendering	Consider the use of alternative packages and later embedding them into the Tkinter GUI
Successfully rendered 3D models through matplotlib and integrated them into the GUI. However the STL file obtained from Fusion360 contained too many polygons, so the model looked bad and performed weak	Attempted to simplify the mesh of the STL file using both online mesh simplification and NumPy functions. However gave that approach due to a of a new idea for resolving the issue up

Conclusion / further planning:

Spent several hours on the STL approach and achieved only poor results (see probably the best looking attempt in Fig. 36), but ultimately shifted focus. We devised a new strategy: generating individual images from various angles of the model using Blender and then mapping them to the current rotation within the GUI.

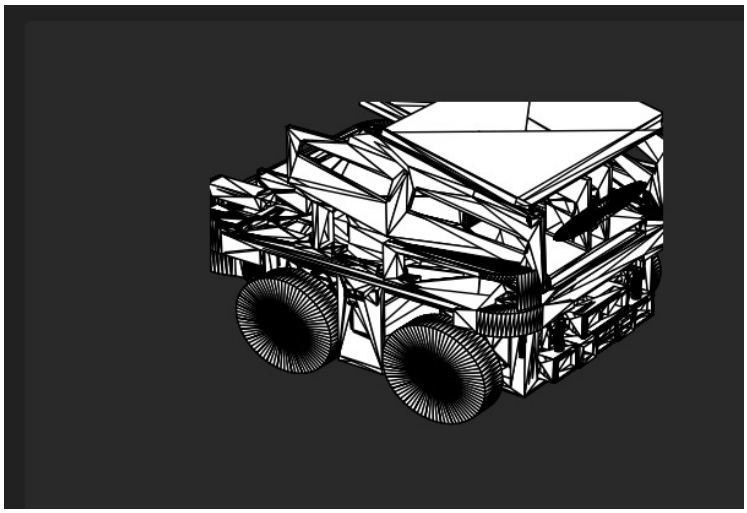


Figure 36: Robots STL file rendered through Matplotlib

by Tim

Date: 2023-10-27

Add sensor data and second camera stream to GUI

Tasks done today:

- Added "Exit" and "Expand" button to GUI using

```
ctk.CTkButton()
```

- Added real time sensor data to the bottom left frame using

```
ctk.CTkLabel()
```

  and writing the current multiprocessing.Value() to this label using

```
label_sensor_1_var.set(str(int(sensor_one.value)) + " mm")
```

  within the main update loop of the GUI (find more information here)

- Replaced the old rescue code within the USB camera script with a simple pass-through of the camera image at the correct resolution of 640px x 480px
- Added the second camera stream to the GUI (the used approach will also allow us to run the GUI within its own multi-process, so it won't slow down the main image processing)

Issues and solutions:

Issues	Solutions
Somehow need to access the camera stream processed in two different multi-processes to display them simultaneously within the GUI (multiprocessing.Value() won't work efficiently on data the size of entire camera frames)	Creating two shared memories, creating NumPy arrays backed by these shared memories and then overwriting these backed NumPy arrays with the processed frames we can access the processed frames of both camera streams in any multi-process (you can find more documentation on this approach here)

Conclusion / further planning:

With the GUI now much more practical than before, Marius will take over the task of adding the rotating robot model to the GUI, since he has more experience with Blender than I do.
I will now try to get the robot to follow the line more correctly and much faster than before, starting with fixing the weird camera zoom seen in Fig. 37 and finally integrating the GUI into our existing code.

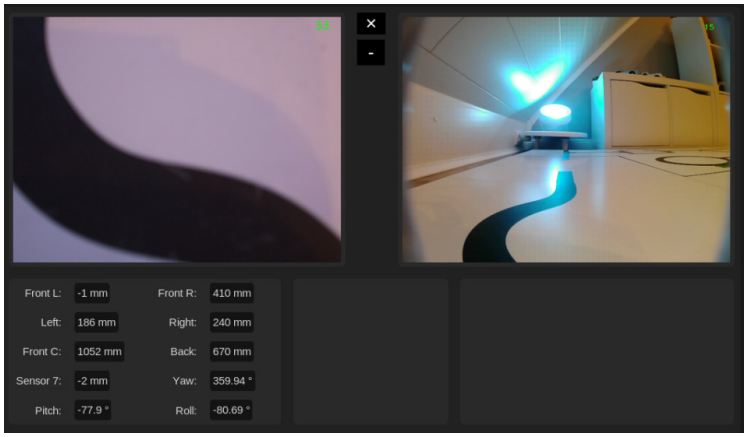


Figure 37: GUI with added sensor data and second camera stream

by Tim

Date: 2023-10-29

Image Sequence Generation using Blender

Tasks done today:

- Using the 3D Model which Tim created in Fusion360 to create a image sequence in Blender where only every 2nd angle is a image to save on memory as seen in Fig. 38
- Pre-convert the images to a NumPy hash map for fast access times and not having to convert every image to a CTkImage which reduces the strain on the CPU
- Save the hash map to a compressed NumPy file so it can be loaded when the program is started

Issues and solutions:

Issues	Solutions
The exported OBJ file having many different materials and the mesh being split up into non-regular chunks which complicates selecting the wheels individually and changing their color	Using a STL file instead which has the entire mesh as one piece and doesn't contain any materials
The hash map having the wrong images assigned to a different key than it should and the convert script trying to read more images than there are available	Changing the loops to first go over the pitch and then the yaw since in the image sequence for every yaw step a full 360° rotation happens. The yaw range should not include 360° as a value because since 0° and 360° are the same so the images are being offset instead of being assigned to their correct keys

Conclusion / further planning:

It was a struggle to get python's arrays and hash maps to work correctly, but with enough time i got it to work and the data is now fully prepared to be implemented into the GUI.

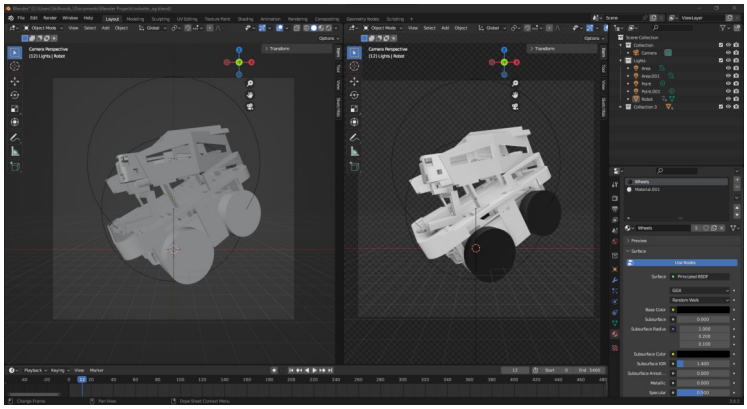


Figure 38: Blender Workspace

by Marius

Date: 2023-10-30

Rotating Robot Model GUI integration

Tasks done today:

- Integrated the 3d model into the GUI (see Fig. 39)
- Made the model rotate select based on the rotation of the gyro sensors

Issues and solutions:

Issues	Solutions
The first time the images were to small, only taking up about a quarter of the allocated space	Re rendering the images in blender with a slightly higher resolution and converting them to a array again
The transparent background won't work when displayed with CustomTkinter	Re rendering the images in blender with the same background color as the GUI frame

Conclusion / further planning:
After fixing the hiccup with the too small images and the not working transparent background shown in Fig. 39 as well as integrating it into the CustomTkinter GUI, it works nicely and is very responsive to rotation changes, even if we only use ever 2nd angle. It does slow down the start of the program though, because we have to load and unpack a 100MB npz file.

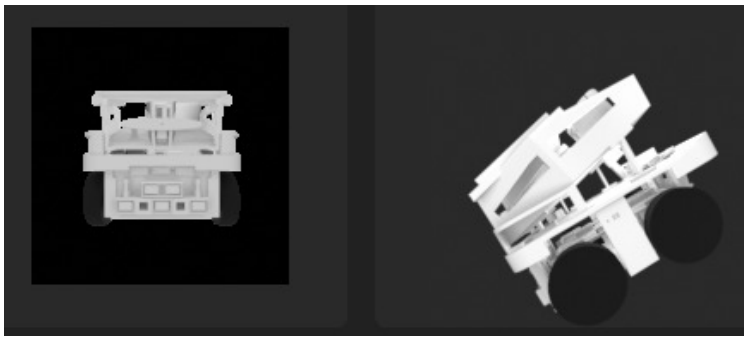


Figure 39: The 3D model in the GUI

by Marius

Date: 2023-10-31

Camera Zoom Issue & GUI integration

Tasks done today:

- Tried to get a more zoomed out image from the Raspberry Pi Camera Module V3 (see Fig. 37), including using the lores stream or adjusting the zoom with

  ```
  full_res = picam2.camera_properties['PixelArraySize']
  ```

  as described in the documentation here, but failed
- Switched back to the Raspberry Pi Camera Module V2 we used on the old robot for now
- Ordered a new Raspberry Pi Camera Module V3 Wide Angle
- Integrated the GUI into our existing code and tested the whole line following with new GUI for the first time
- Added a label indicating the detection of any green markers to the GUI (see Fig. 40)

Issues and solutions:

Issues	Solutions
The Raspberry Pi Camera Module V3 seems to zoom in much more than our old Camera Module V2. With just this small portion of the line, our program is unable to follow it correctly	As you can see here the focal length of the two cameras is very different (3.04mm to 4.74mm) which might explain this "zoom". For now, we are just using the Camera Module V2 again, but we also ordered a Camera Module V3 Wide Angle, which should solve this problem

Conclusion / further planning:

With this achievement of a completely new, much more performant and useful GUI that can run while following the line (see Fig. 40), at least i won't be able to do much more on the robot until mid December due to school trips and a subsequent exam period.
From then on, the aim is to improve the old code so that we can safely drive faster (the raspberry Pi 5 will hopefully contribute a lot to this) as well as finally attaching the arm and taking care of the whole Rescue part.

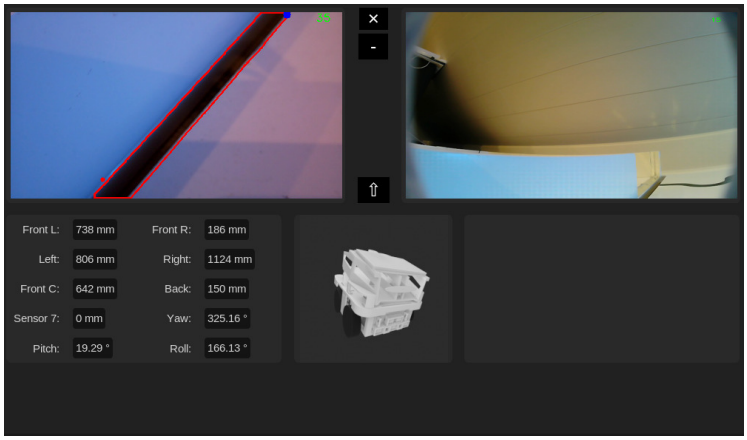


Figure 40: GUI integrated into existing line following code

by Tim

Date: 2023-11-07

Managed to order the new Raspberry Pi 5

Tasks done today:

- Marius: Ordered two Raspberry Pi 5's on BerryBase
- Tim: Ordered one Raspberry Pi 5 on BerryBase

Issues and solutions:

Issues	Solutions
Being limited to order only one Raspberry Pi 5 per account. Additonally BerryBase requires the account to have made a purchase previously to be eligible for ordering a Raspberry Pi 5.	Using my 2nd email that I have ordered something with regarding the robot before, so I was able to get 2, one for the robot and one as a replacement.

Conclusion / further planning:

After waiting for a long time and multiple delays of the actual release date, we were finally able to order some. BerryBase said they were allocating the Raspberry Pi 5's randomly to users but that doesn't seem to be the case since Tim and I were able to order three of them in total when there are only 55 Pi's available as seen in Fig. 41.

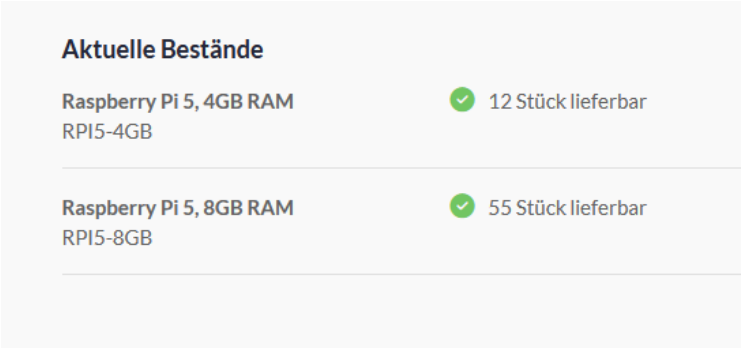


Figure 41: Stock of the Raspberry Pi 5's on BerryBase

by Marius

Date: 2023-11-10

Receiving the new Raspberry Pi 5

Tasks done today:

- Received 2 Raspberry Pi 5's
- Installed the official active cooler onto both PI's
- Installed Raspberry Pi OS Bookworm and did some performance testing
- Fixed VNC not working
- Fixed Python PIP modules refusing to install

Issues and solutions:

Issues	Solutions
When I try install a python package using pip, python refuses and gives me a error with externally managed environments and warns me about PEP-668.	Executing the command `sudo rm /usr/lib/python3.11/EXTERNALLY-MANAGED` fixed the issue permanently. I added it to our install script and update script to prevent it from ever happening again.
When trying to connect with RealVNC to the Raspberry Pi 5, RealVNC disconnects and warns me about issues with the encryption level, even though it is set to automatic.	By executing `sudo raspi-config` in a terminal and then going to Advanced Options → Wayland → X11 and rebooting, the Raspberry Pi 5 now uses the older X11 backend which is compatible with VNC. When trying to connect now it works correctly.

Conclusion / further planning:

After doing some quick performance testing with our program, I can say that performance has increased, as expected, by 2-3 times over the Raspberry Pi 4B. We will definitely keep using the Raspberry Pi 5 in the future and swap out our Raspberry Pi 4B in the Robot.

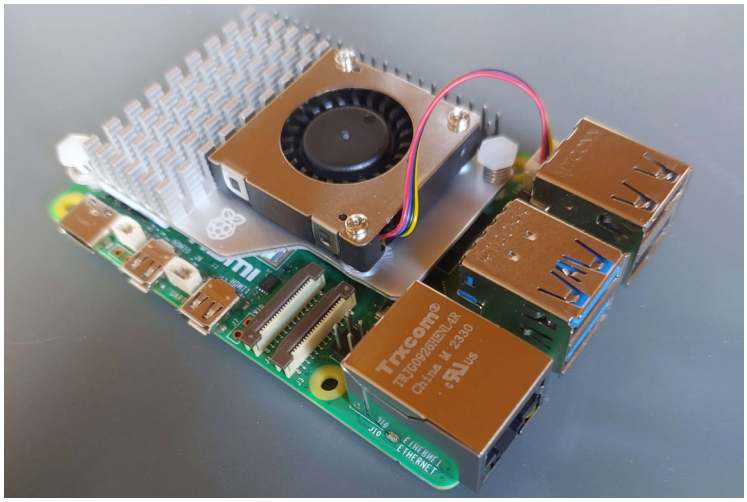


Figure 42: The new Raspberry Pi 5

by Marius

Date: 2023-11-20

Solution for inference speed without coral usb accelerator

Tasks done today:

- Diagnosed the problem when running the ball detection ai using the coral usb accelerator
- Found a solution so we don't have to use the usb accelerator when running the ai and implemented it

Issues and solutions:

Issues	Solutions
When running the ai model using the Ultralytics python module on the coral usb accelerator it segmentation faults. I created a issue on the tensorflow GitHub here but there was no help yet.	Instead of using the coral usb accelerator to increase inference speed from the measly 5 fps with pytorch, I converted the model using the Ultralytics export function into tensorflow lite with full int8 quantization which yields about 45fps using the small model and 17 with the big model

Conclusion / further planning:
As the coral usb accelerator is currently not usable we moved to just running the ball detection model on the CPU of the raspberry pi 5 which yields fast inference speeds. It still has to be tested though how this affects the rest of the program due to it also running on the CPU.

by Marius

Date: 2023-11-23

Fixing screen detection on Raspberry Pi OS Bookworm

Tasks done today:

- Fixed an issue with screen detection with Raspberry Pi OS Bookworm that we had discovered a while back. In the past, we successfully resolved this issue by adding

  ```
  hdmi_cvt:0=1024 600 60 3 0 0 0
  hdmi_group:0=2
  hdmi_mode:0=87
  ```

  to the raspi-config.txt file.
- Add a on-screen keyboard to Raspberry Pi OS

Issues and solutions:

Issues	Solutions
Our Raspberry Pi 5, running the latest Bookworm Raspberry Pi OS, is no longer detecting our screen. Due to the unique resolution of 1024x600px we previously resolved this issue by editing the raspi-config.txt	After some investigation, we found that adding `video=HDMI-A-1:1024x600M@59.96D` to "boot/cmdline.txt" and ensuring the correct framerate ("@59.96D") for this video mode with `~$ xrandr`

Conclusion / further planning:

It was a struggle to get the display to work with the Raspberry Pi 5 because we didn't expect it to have a framerate of 59.96 and always assumed it would be 60. This lead to us always putting 60fps into the config settings and it never working until one of us randomly decided to test out the odd fps setting that xrander gave out for our screen size.

by Tim

Date: 2023-11-23

Fixing the installation script for the Raspberry Pi 5

Tasks done today:

- Updating the installation script to work for the Raspberry Pi 5 as it was made for the Raspberry Pi 4B

Issues and solutions:

Issues	Solutions
Onboard, the on-screen keyboard we use not being auto installed so its impossible to do anything on the display	Adding `sudo apt-get install onboard` to the install script and replacing the configuration file with a preconfigured one
The inability to use RealVNC with the raspberry pi 5 due to the new OS being based on the Wayland and not X11 window manager	By reading the shell script "raspi-config" the lines that configure the pi to use X11 instead of Wayland have been added to the install script to enable the use of RealVNC
Python by default is configured to not allow the installation of modules via pip and says you should install the via apt. This is bad due to alot of the modules we need not being in the apt repository	Deleting the python flag that forbids the installation of pip modules using the "rm" shell command

Conclusion / further planning:

Since we need certain packages and python modules installed for the program to work correctly we have a install script which auto installs those. The problem is that it was made for the Raspberry Pi 4B and does not cover all the things that have changed with the new Raspberry Pi OS Bookworm.

by Marius

Date: 2023-11-25

Improving detection accuracy for ball detection model

Tasks done today:

- Using onnx instead of tensorflow lite for inference
- Inference speed and precision testing with the onnx model format

Issues and solutions:

Issues	Solutions
Using the tensorflow lite (tflite) model format with fully quantized tensors did provide very fast inference speed, but the precision of the model was very bad compared to the raw pytorch model. Often detection boxes were too large or did not include the entire ball. In addition to that the detections had low confidence value often only being in the 50-70% ranges	After testing various model formats including onnx, openvino, tensorflow lite, pytorch and tensorflow lite edge tpu, it proved that the onnx model format provided acceptable inference speeds (~25fps) and much better detection precision compared to tensorflow lite as seen in the picture below

Conclusion / further planning:

Even though we are not quite getting the full 30fps which the camera can deliver, ~25fps is still enough with the benefit of getting much better detection accuracy. This wouldn't have been possible using the raspberry pi 4B as its CPU would be too slow to run inference on it and the program at the same time. The raspberry pi 5 has a extreme speedup over the pi 4B and sits at around 20-30% CPU usage while running inference, providing enough CPU leftover for the rest of the program.

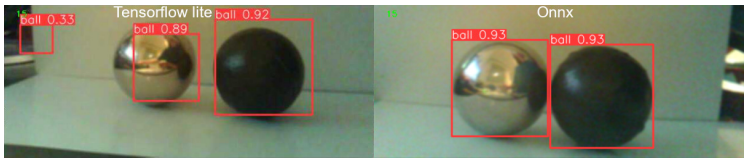


Figure 43: Tensorflow lite and onnx comparison

by Marius

Date: 2023-12-05

Create YouTube channel

Tasks done today:

- Created a new YouTube channel with the name OverEngineering[2]
- Edited all our past runs in Davinci Resolve
- Uploaded the first few runs and set them to private

Issues and solutions:

Issues	Solutions
Difficulties finding the scores and times for the older runs, especially for the ones in 2022, because most websites that had these scores don't exist anymore	Luckily the website ran by our teacher kept track of most of the results and we found pictures of the result papers in our phone galleries after some digging

Conclusion / further planning:

Since I was somewhat familiar with Davinci Resolve, the editing of the runs was finished in about a day and uploaded to YouTube as a private video, because we wanted to publish them later on. We chose to upload the runs to YouTube because we wanted some way of archiving the runs and it may help inspire other teams.

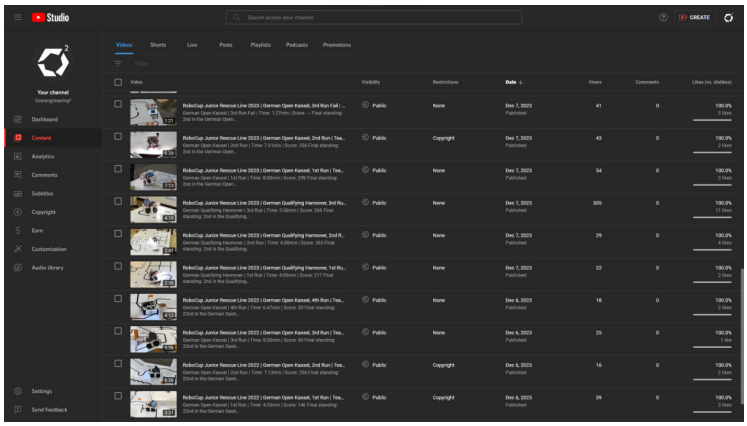


Figure 44: The first runs in YouTube Studio

by Marius

Date: 2023-12-06

Model format speed and accuracy testing

Tasks done today:

- Converted 3 different model sizes to 4 different formats each
- Ran validation tests on the total of 12 different models and created a graph using mathplotlib for visualisation

Issues and solutions:

Issues	Solutions
As seen in Fig. 45, using onnx instead of tensorflow lite did improve model accuracy, but since ultralytics supports many more export formats such as openvino, I was unsure whether onnx is the best choice or not	Exporting different model sizes such as n, s and m (l and x are infeasible to run as they would take multiple seconds to process a single image) to onnx, tensorflow lite and openvino to get insight on the differences in accuracy and inference speed
Exporting and running the different models is very time intensive and produces many files that are complicated to manage	Creating a script that automatically exports the models to the different formats and afterwards run each one using the validation function on the coco128 dataset
Difficulties creating a graph which compares the results of the validations	Using matplotlib to plot the map 50-95 and total inference time (including pre and postprocessing) while the validations are running

Conclusion / further planning:

Even though tensorflow lite provides the best inference speed, it's accuracy is atrocious and cannot be considered a valid candidate for our usecase. Pytorch is also useless as it has much slower inference speeds compared to model formats like onnx or openvino. Since openvino and onnx are CPU optimized model formats they naturally run much better compared to pytorch, but not as fast as tensorflow lite due to the inability to quantize the model. It is surprising that onnx and openvino somehow manage to get better map 50-95 scores compared to pytorch, because the pytorch model is the one all others are based off. In the end it is clear that onnx is the best choice as it has slightly lower inference times and slightly better map 50-95 scores compared to openvino.

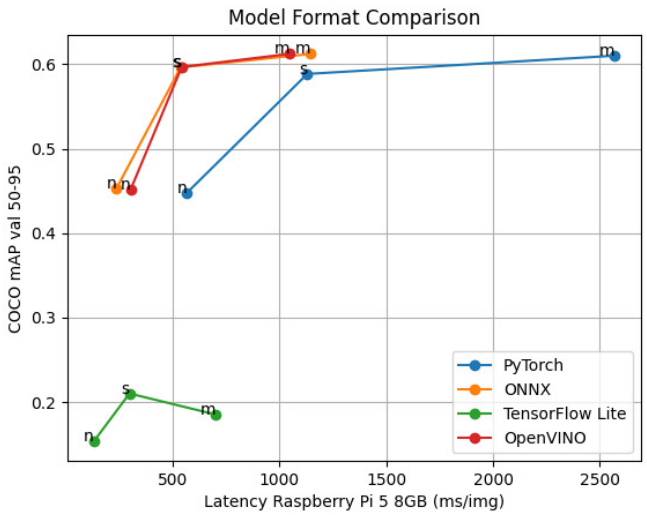


Figure 45: Map 50-95 and inference speed comparison

by Marius

Date: 2023-12-09

Rewriting the GPIO controller

Tasks done today:

- Rewrote our custom GPIOcontroller with a newer library

Issues and solutions:

Issues	Solutions
After switching to the raspberry pi 5, our GPIO controller using the RPI.GPIO library crashes when initializing the GPIO pins	Rewriting the entire GPIO controller to use the gpiozero library instead, as the raspberry pi documentation recommends to use this library now instead of RPI.GPIO

Conclusion / further planning:

Thanks to tips on how to migrate from RPI.GPIO to gpiozero in their documentation, we were able to quickly do the switch. After testing the new GPIO controller we determined that it works fine and works just like the old one.

by Marius

Date: 2023-12-20

Switched to the Raspberry Pi 5 and Camera Module V3 Wide Angle

Tasks done today:

- Connected all the USB cables to the new Raspberry Pi 5 and and placed it on top of the Raspberry Pi 4 for now
- Temporarily powered the new Raspberry Pi 5 by using the original power supply
- Tested different sensor modes of the Camera Module V3 Wide Angle using

  ```
  mode = camera.sensor_modes[0]
  ```

  to get the different sensor modes of the camera followed by

  ```
  camera.configure(camera.create_video_configuration(sensor={'output_size': mode['size'], 'bit_depth': mode['bit_depth']}))
  ```

  to set the camera to this sensor mode using the Picamera2 Library
- Changed the aspect ratio of both camera images to 16:9, which is the correct ratio without stretching
  (similar to the Camera Module V2's documentation, different sensor modes result in different fields of view for the camera)

Issues and solutions:

Issues	Solutions
The camera delivers a 2304px x 1296px image with our chosen sensor mode, which is too large for the image processing to run at an optimal speed	Resizing the camera image to our desired size for image processing by using `image = cv2.resize(image, (448, 252))`
It takes a long time to replace the old Raspberry Pi and we don't know if everything will work as before with the Raspberry Pi 5	Connecting both cameras as well as all the other USB cables to the Raspberry Pi 5 and and placing it on top of the Raspberry Pi 4 (the power cable is too short, so I am using the original power supply for now)

Conclusion / further planning:

With the combination of the new Raspberry Pi 5, code optimizations and the switch back to a Camera Module V3, we manage to achieve a frame rate of about 60 FPS with the image processing running as seen in Fig. 46. This is a huge improvement, especially compared to the old robot with a frame rate of about 20 FPS.

The aim of the upcoming Christmas vacations is to

- integrate the victim AI
- install the Raspberry Pi 5 to the robot
- add the gripper arm
- improve line following, intersections, gaps and obstacles
- detect silver
- add evacuation zone program (including the exact position of the robot within the zone by analyzing the readings from the distance sensors in combination with the rotation angles and acceleration from the gyro sensors)

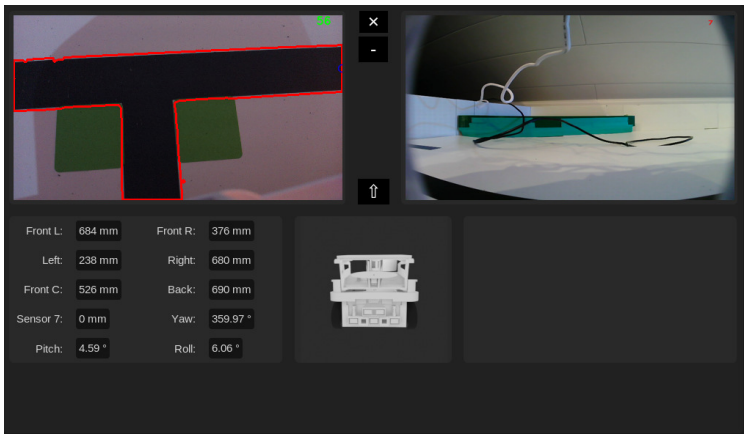


Figure 46: Improved camera image using the Raspberry Pi Camera Module V3 Wide Angle

by Tim

Date: 2023-12-21

Gathering and labeling of new training data

Tasks done today:

- Tim: Captured a few hundred new images in the evacuation zone with around 10% background images
- Marius: Labeled all images with Roboflow
- Marius: Trained a new yolov8n model with a Google Colab A100 instance

Issues and solutions:

Issues	Solutions
We still lack some accuracy in darker areas and sections where the victims partially blend in with the background.	Adding more images of these scenarios to our dataset. After training we see improvements in those areas, making our model more robust.

Conclusion / further planning:

After expanding our dataset again, we see the improvements we hoped for. Adding the background images eliminated all of our issues with false positives.

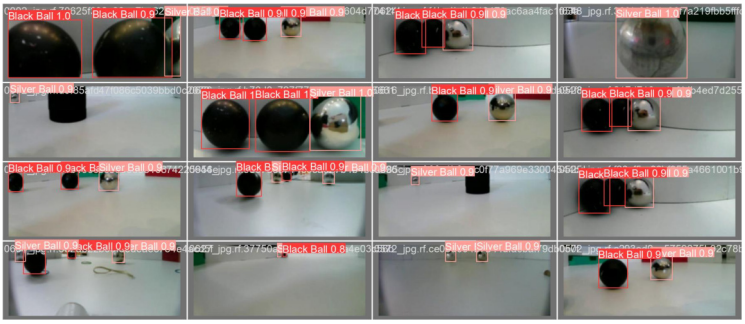


Figure 47: Training results

by Tim & Marius

Date: 2023-12-22

Integrated victim AI into existing code

Tasks done today:

- Changed the aspect ratio of the zone camera frame in the GUI to match the cropping at the top and bottom
- Added a capture button to the GUI to make it easier to save new training images
- Tested the performance of the AI models provided by Marius
- Integrated the AI's output boxes to the camera frame zone within the GUI.
- Added an averaging function to average the gyro angles from both gyroscope sensors (before that, we had just used one)

Issues and solutions:

Issues	Solutions
The CPU usage while running any AI model is abnormally high and the frame rate therefore bad	Not fixed yet (Marius will try solving this problem soon)

Conclusion / further planning:
The first goal on the list for this Christmas holiday is now complete.
Hopefully Marius will manage to solve the performance problem, in the meantime i would like to finally get the hardware ready, i.e. attach the arm and install the Raspberry Pi 5

by Tim

Date: 2023-12-23

Train a new models with 2 classes

Tasks done today:

- Trained a new AI model with 2 classes with Google Colab
- Tested the new model on validation images and parkour

Issues and solutions:

Issues	Solutions
Old model did not perform well with a single class that combines both victim types	Giving each victim type their own class, significantly improving detection accuracy

Conclusion / further planning:
The decision to go with a 2-class model is definitely worth it because we see immediate improvements over our previous single-class AI.
We still need to gather better and more training data though as the model is not good enough yet.

by Marius

Date: 2023-12-24

Fix CPU usage being abnormally high

Tasks done today:

- Exporting the current model to tensorflow lite fp32
- Comparing CPU usage between tensorflow lite fp32 and onnx

Issues and solutions:

Issues	Solutions
While running the program it became a problem that the AI model ran at only 3 fps when started under multiprocessing with the rest also being active. Also predictions on objects that are further away often doesn't work	After testing some more different model formats such as ncnn and tensorflow lite fp32. The result is that tensorflow lite fp32 seems to be the best because it results in about 7-8 fps even though in the comparison its worse

Conclusion / further planning:

In the end its concluded that tensorflow lite fp32 is the best because it seems to only use one core for the same inference speed as onnx or openvino, leaving more processing power for the rest of the program and being able to pump out more fps.

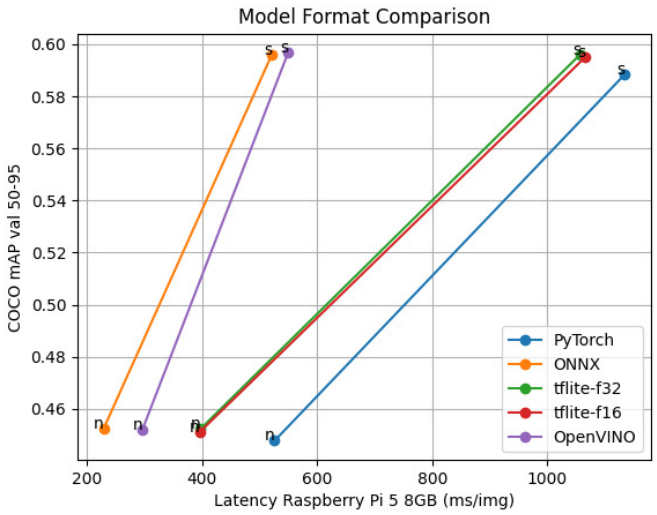


Figure 48: Comparison of different machine learning frameworks

by Marius

Date: 2023-12-25

Installing the Raspberry Pi 5

Tasks done today:

- Replaced the old Raspberry Pi 4B on the robot with the new Raspberry Pi 5
- Added a CPU usage label to the GUI using psutil and the function

  ```
  psutil.CPU_percent()
  ```

  within the GUI's main loop

Issues and solutions:

Issues	Solutions
After booting the Pi from the robot's power supply, some power-related warnings appeared	Not solved yet, but probably related to the high power requirements of the Raspberry Pi 5. We will try to increase the output voltage of the step-down converter to 5.1V first

Conclusion / further planning:

Replacing the Raspberry Pi took about 3 hours as expected. Hopefully the power-related warnings are not a major problem so that we can still stay on schedule.

by Tim

Date: 2023-12-26

Fixing the power problem

Tasks done today:

- Changing the output of the step-down converters to be 5.1V 5A
- Config change in /boot/config.txt to enable max power draw for USB devices

Issues and solutions:

Issues	Solutions
When booting up the raspberry pi 5 on the power supply of the robot we noticed that the screen flickers and the front camera not working when running the program	Adding `usb_max_current_enable=1` to `/boot/config.txt` and configuring the step-down converters to have 5.1V instead of 5V since the spec for the official power brick is 5.1V at 5A

Conclusion / further planning:

While diagnosing the problem we noticed that when we unplug the screen the camera starts working which made us suspect it to be a power issue. To be sure we plugged the pi into the official power brick and had no issues. After further diagnosing we noticed that usb_max_current_enable was disabled when the pi is running over our internal power supply. As seen in Fig: 49 the EXT5V_V voltage was previously at around 4.95V and after increasing the step-down converters 5.13V which is in line with the official spec. In conclusion we should have accounted for the increased power draw of the pi 5 and configured the step-down converters accordingly.

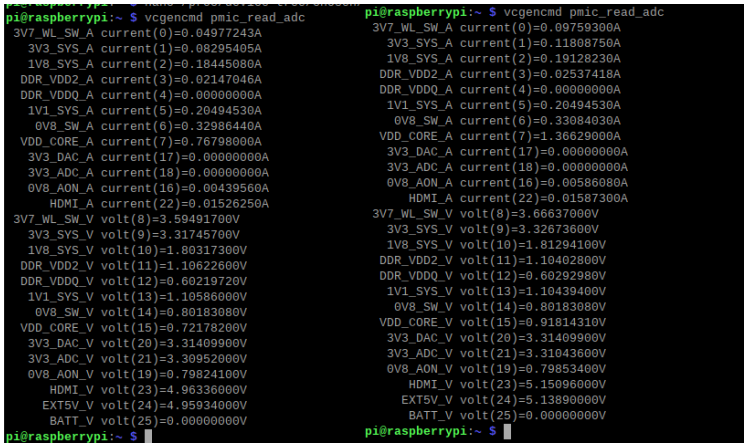


Figure 49: Old- vs. new voltages

by Tim

Date: 2023-12-30

Adding the gripper arm to the robot

Tasks done today:

- Cut and soldered the cables for the servo motors and distance sensor so that they share the same power supply cables (reducing the amount of cables moving with the arm)
- Taped all loose cables for safety, so the robot can't get caught in them
- Wrote and tested Arduino code to create motion sequences to pick up dead/alive victims with the arm
- Wrote a simple communication script between the Raspberry Pi and the Arduino Nano using 4 jumper wires between them

Issues and solutions:

Issues	Solutions
Multiple servos need to rotate simultaneously for the most time-efficient pick-up, but an Arduino can only control one servo at a time, and start/stop angles need to be easily adjustable during testing	Rotating the servos in a loop alternately by only one degree (i wrote the code so that by specifying a start and end angle, it automatically calculates how much a servo must be rotated per iteration so that in the end all servos that are rotated at the same time reach their specified end angle at the same time)
The two mounting holes in the metal arm of the servo are very close together, which doesn't look particularly stable	Drilling an additional hole and thread further down in the arm
Had some inconsistencies when reading the LOW/HIGH values of Raspberry Pi's GPIO pins	Solved this by adding an additional ground wire between Pi and Arduino, although they are powered by the same power supply, so should share a common ground

Conclusion / further planning:

Sorting the victims in the designated storage area works as intended, although the advertised 180° servo rotation is not possible with the EMAX-ES09MD servos (more like 160°, which we need to fully utilize). The 270° rotation of the Dymore 35 kg digital servo is needed to lower the arm to a resting position.

With this last upgrade, the robot should be complete, at least hardware-wise. Now it's time to finalize the software.

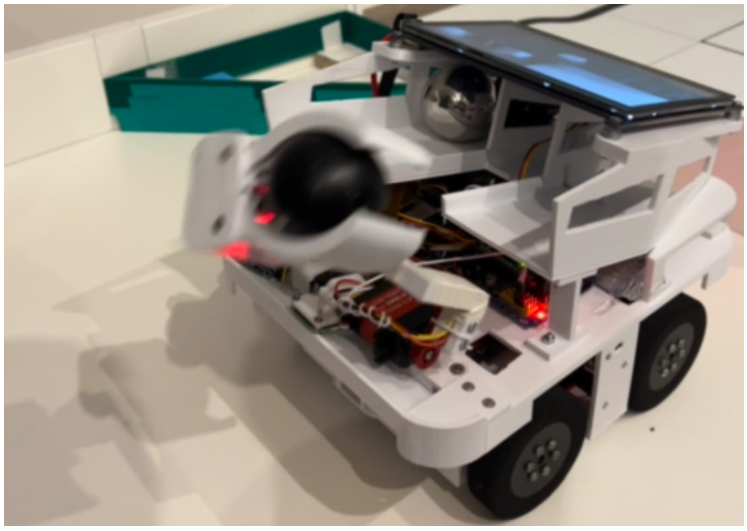


Figure 50: Gripper arm sorting test

by Tim

Date: 2024-01-01

Code cleanup and zone cam update

Tasks done today:

- Ran quick code cleanup by PyCharm to make code uniform and more readable
- Add multiprocessing manager values for victim type and angle to victim
- Refactor math for calculating the angle between the middle of screen and victim

Issues and solutions:

Issues	Solutions
Currently no way of processing the angle and victim type for the zone camera in the control loop	Adding multiprocessing manager values for both which the control loop will use in the future for deciding how to turn and in what compartment to place the victim
The code is difficult to read due to being compacted a lot and many things being inlined	Using the code refactor feature of PyCharm to refactor the code confirming to the PEP 8 style standard, making the code much easier to read

Conclusion / further planning:

Some progress on the code for the zone victim detection, now properly implementing angle calculation and sharing that to the control loop which is to be implemented in the future.

by Marius

Date: 2024-01-02

GUI improvements & Sensor averaging

Tasks done today:

- Captured around 800 images of victims in different lighting situations
- Added a timer label to the GUI that activates the moment the LOP switch is moved from hold to run for the first time within a program restart
- Added an angle label to the GUI that shows the current angle to the black line determined by the image processing
- Added a status bar label to the GUI that shows what part of the program the robot is currently in, making it easier to debug
- Cleaned up the old line code by making it more understandable using functions that can be easily compiled with Numba in some cases, making the code much faster
- Fixed an issue with recurring image names when using the capture button (image names are now based on

  ```
  num = len(os.listdir(path)) - 1
  ```

  which returns the number of files in the given directory minus one folder)
- Added an averaging function for every sensor data

Issues and solutions:

Issues	Solutions
Working with only one current value that a sensor is currently returning is dangerous because of potential measurement errors	Adding a system that stores each value received from each sensor, along with the current time, in a NumPy array, so that you can e.g. easily obtain an average value over the last second

Conclusion / further planning:

By adding a function with 4 lines of code:

```
def get_time_average(time_value_array, time_range):
  time_value_array = time_value_array[np.where(time_value_array[:, 0] > time.perf_counter() - time_range)]
  if time_value_array.size > 0: return np.mean(time_value_array[:, 1])
  else: return -1
```

we can now easily work with an average sensor value from both distance and gyro sensors, making us less susceptible to random measurement errors.

With most of the functions of our image processing using Numba now, it runs even more efficient. Nevertheless, we still have a problem with Numba in a function that needs to be fixed. After that, we should be able to run the program at significantly higher FPS, which will allow us to drive faster.

by Tim

Date: 2024-01-02

Fix distance/angle calculation in zone

Tasks done today:

- Remove old angle calculation that uses "real" angles
- Add new distance calculation that uses distance instead
- Update our requirements.txt to force tensorflow at version 2.13.1

Issues and solutions:

Issues	Solutions
Our current angle calculation doesn't make much sense, since we view the victim from the front and not on top, so using a angle is pointless	Change to using a simpler distance calculation instead, that just calculates the distance between the middle of the screen and the middle of the victim
When exporting our AI to tensorflow lite it constantly crashes	Forcing tensorflow to be at version 2.13.1 resolves the issue, which is also what ultralytics recommends

Conclusion / further planning:

Since we can ignore the vertical aspect of orienting the robot to the victims, we chose a much simpler approach which only involves the distance to the middle of the image.

by Marius

Date: 2024-01-03

Numba fix and huge speed increases

Tasks done today:

- Fixed random Numba error (more on that below)
- Fixing the last Numba bug allowed us to increase the motor speed to the maximum the motors are capable of, resulting in a speed we have never seen before with this robot
- Adapted the GPIO control script to the new motor speed (reduced speed as the angle to the line increases, which prevents losing the line)

Issues and solutions:

Issues	Solutions
Numba crashed while trying to compile the largest function of our image processing	Finding out that standard python functions like `max()` aren't supported by Numba and we have to use `np.maximum()` instead

Conclusion / further planning:

The robot is now very fast on a continuous line (completing the course seen in Fig. 51 in about 16 seconds), but as always it is too imprecise for gaps, which is why we still have to find our own solution for these. We also need to rework the 180° rotation method, which has been very slow in the past.

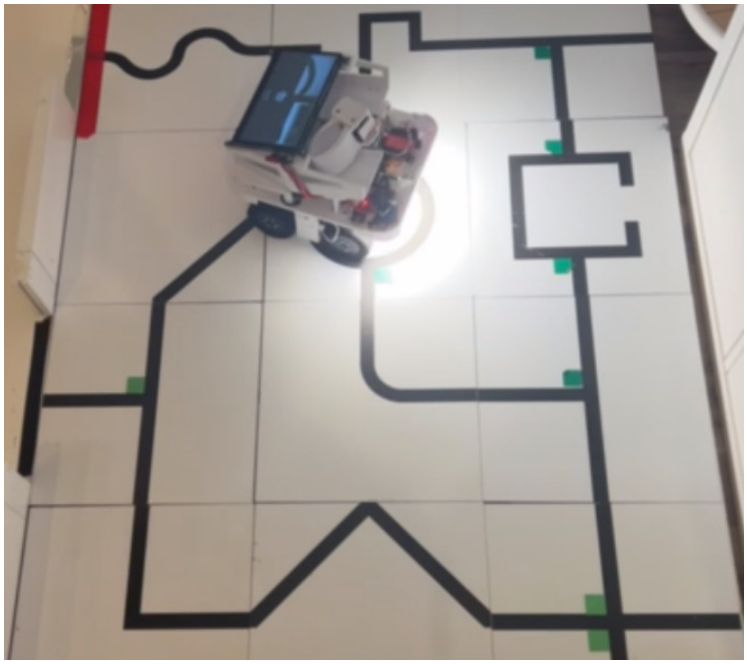


Figure 51: Test run with motor speed increased to maximum

by Tim

Date: 2024-01-04

LabelImg crashing and image annotating

Tasks done today:

- Fixed LabelImg constantly crashing when trying to annotate
- Annotated 500 more images for the victim detection AI
- Finish the method that turns the robot towards victims
- Adding the "agnostic_nms" argument to AI inference to improve box clarity

Issues and solutions:

Issues	Solutions
LabelImg crashes every time when trying to draw a box label. Additonally it also crashes when trying to zoom in our out	Finding out turning off the the "Auto Save" feature fixes the first crash and thanks to this Github issue I was able to fix the remaining errors

Conclusion / further planning:

After fixing the annoying LabelImg issues, I was able to quickly annotate 500 new images for our AI. The control loop for the zone still needs a lot of attention. We currently have no way of detecting the silver line and will have to look at different solutions in the future like dedicated sensors, image processing methods or maybe another AI. The problem with silver currently is that it is not reflective due to our camera lowering the exposure so much. A example can be found in Fig. 52.

Figure 52: The non reflective silver strip

by Marius

Date: 2024-01-05

Improved angle turning Speed

Tasks done today:

- Improved long If/Else statement using arrays and indexes, making it more readable and able to debug
- Added new function to turn the robot to a given angle using the rotation value from the gyro sensors
- Tried to find a mathematical equation linking the time turned at maximum speed to the angle the robot turned within that time using custom written code and regression (This approach failed due to inconsistencies caused by the grip of the wheel and the surface of the ground)

- Switched back to turning only by the angle provided by the gyro sensor, turning faster the greater the difference from the desired angle. However, we have improved this old method by using a new, more aggressive function (see Fig. 53) to determine the rotation speed from the angle difference. This is achieved with the following line of code (both the minimum rotation speed of 0.3 and the increasing factor of 3 here will probably be improved with more testing over time)

```
speed = max(1 - pow(abs(abs(angle_to_turn) / 360 - 1), 3), 0.3)
```

Issues and solutions:

Issues	Solutions
Turning to a specific angle often took more than 10 seconds because the robot turned with the same speed all the time	Finding a function that allows the robot to turn faster the greater the difference from the desired angle, without being too fast and overshooting all the time

Conclusion / further planning:
With this new function, shown in Fig. 53, the robot turns much faster in dead ends as well as when navigating the evacuation zone, which is still to be done.

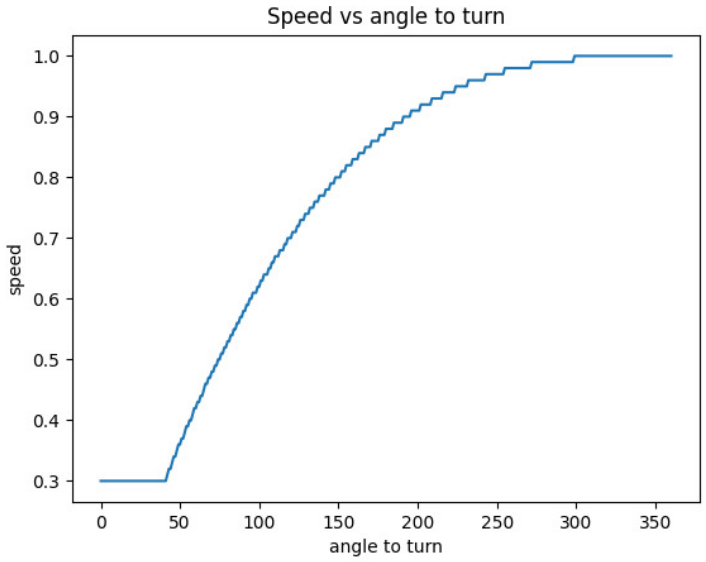


Figure 53: Turning speed per angle to turn chart

by Tim

Date: 2024-01-06

New victim model and ncnn testing

Tasks done today:

- Augmented the new dataset using Roboflow
- Trained a new yolov8 model with Google Colab and an A100 on the new dataset
- Exported the model to tensorflow int8 and ncnn by Tencent to test speeds

Issues and solutions:

Issues	Solutions
I am not really happy with the way the boxes are not very consistent in size and change alot when using tensorflow int8. This is because int8 models have way less information in the model and are usually less than half the size of full FP32 models	Testing if ncnn can be fast enough to run on the CPU while also having the program running, which may help with accuracy

Conclusion / further planning:

The new models has the improvements we wanted to happen through the new dataset and using ncnn to run the model may be worth it to keep for the future, as it is more accurate than tensorflow int8 while still being relatively fast.

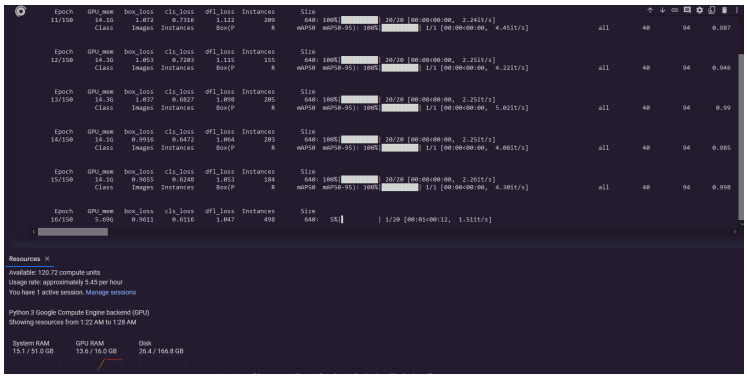


Figure 54: Training of the new model on Google Colab

by Marius

Date: 2024-01-07

Remove not needed model and update install scripts

Tasks done today:

- Removed the FP16 ncnn model
- Change flatbuffers to version 23.5.9
- Change tensorflow to tensorflow-aarch64 version 2.13.1
- Add automatic ncnn installation to the install shell script

Issues and solutions:

Issues	Solutions
The experimental FP16 ncnn model, which we hoped would have performance gains, mostly bugs out and produces random boxes that make no sense	Deleting the model due to it not being alot faster than the regular FP32 model and having many issues
High CPU usage with the ncnn model, shown in Fig. 55, impacting the rest of the program significantly	Using a tensorflow lite FP32 model instead, as we noticed it delivers the same performance while only using one core of the CPU

Conclusion / further planning:

The quest to finding a suitable alternative for the Coral TPU is very difficult, because we need a AI model format that is capable of many FPS while still being fairly accurate and not impacting the rest of the program too much. We have to do this because for a unknown reason the Coral TPU decided to not work anymore and constantly produces segmentation faults.

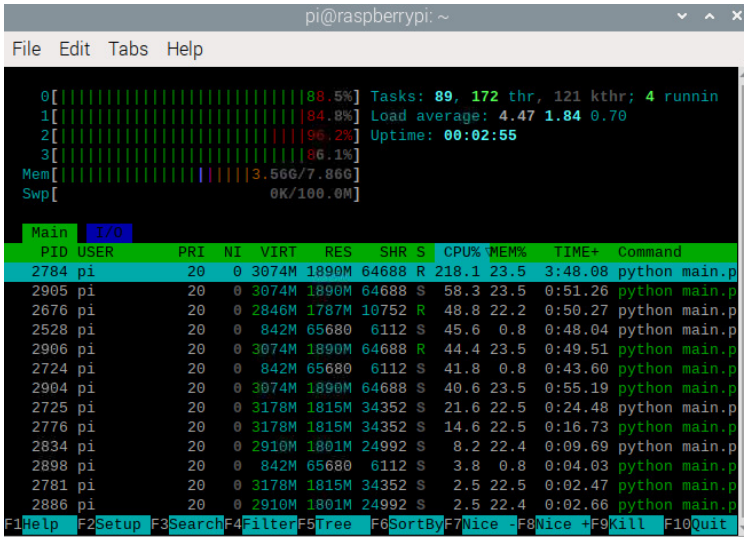


Figure 55: The high CPU usage of the ncnn model

by Marius

Date: 2024-01-08

Testing with lower resolution models

Tasks done today:

- Exported a ncnn model with a image size of 448x192
- Tested said model thoroughly for performance and accuracy

Issues and solutions:

Issues	Solutions
Still having fps issues with the current ncnn models, we are heavily CPU limited	Training and exporting a model with a lower image size of 448x192 which does increase FPS, but reduces the accuracy alot and makes the detection of distant objects alot worse

Conclusion / further planning:

Even though the lower resolution model improves FPS to about 10, the accuracy is far worse than what we require. I'm gonna have to get the Coral TPU working somehow because it is just not possible to run the AI models with acceptable FPS and accuracy while having the rest of the program open as well.

by Marius

Date: 2024-01-09

New technique for managing gaps

Tasks done today:

- Rearranged some buttons within the GUI and added a second timer tracking the total time spend in the evacuation zone (starts when silver is detected and will end when the exit is found)
- Added a simple gap detection (if the area of the black line is below a certain threshold)
- Added a calculation of the "gap angle" which is the angle the angle of the upper side of the line to the horizontal axis (the blue line seen in Fig. 56). We calculate this angle by drawing a min area rectangle around the black line using OpenCV, sorting the corner points by their height, and calculating the angle of the vector between these two points
- Added a repositioning program until this angle is 0°

Issues and solutions:

Issues	Solutions
We often lost track of the line when facing gaps due to an imprecise but abbreviated driving style	Stopping and precisely repositioning when a gap is detected (in the past, the robot would turn based on its angle from the gyro sensor, but this approach won't work with gaps at 45°)

Conclusion / further planning:

With this new tactic, gaps are no longer a problem, so the main objective for line driving now is to add obstacles and test ramps, but this is only possible later in the school. In addition, we still recognize silver poorly and have not yet tested and debugged the code for the evacuation zone.

Figure 56: Determining the gap angel

by Tim

Date: 2024-01-09

New models and tflite int8 testing

Tasks done today:

- Trained new yolov8 model with a image size of 544
- Exported the model to tflite int8 and ncnn
- Tested an alternative way of running tflite int8 models

Issues and solutions:

Issues	Solutions
We still have the issue that tflite int8 models run slower than regular tflite FP32 models and need to achieve more FPS inside the zone to be able to detect the victims faster	Trying a custom tensorflow lite implementation from here, but ultimately failing, because it did not improve the inference speed

Conclusion / further planning:

Even though the new model with a higher image size performs better in terms of accuracy, it is slower. I hoped a custom implementation of tensorflow lite would fix the issue, but it doesn't. It looks like its some underlying issue with tensorflow on Raspberry Pi's because as seen in Fig. 45, a tflite int8 model should be much faster. I will continue debugging this issue and hope to some day get it working properly and maybe also the Coral TPU.

by Marius

Date: 2024-01-12

Annotating the new images and train new model

Tasks done today:

- Tim captures 727 new training images of different victims with different lighting
- Annotated all 727 images with LabelImg
- Trained the new model with a Google Colab A100 instance
- Export the model to tensorflow lite FP32 using the ultralytics export mode
- Test the model in a testing parkour

Issues and solutions:

Issues	Solutions
The model still lacks in some areas, especially detecting from further away causing us to miss victims on the other end of the zone	The new model trained on the new dataset improved the ability to detect from further away quite a bit, shown in Fig. 57, which we are very happy with. That means that we hopefully don't have to drive around in the zone much

Conclusion / further planning:

Annotating all the images took about a day and the model training less than a hour thanks to the fast Nvidia A100 GPU's provided by Google Colab. We are now very happy with the accuracy of the current model, though we still need more FPS so we can traverse the zone quicker. As usual I will continue to debug the issue with tflite int8 and the Coral TPU, but no updates so far.

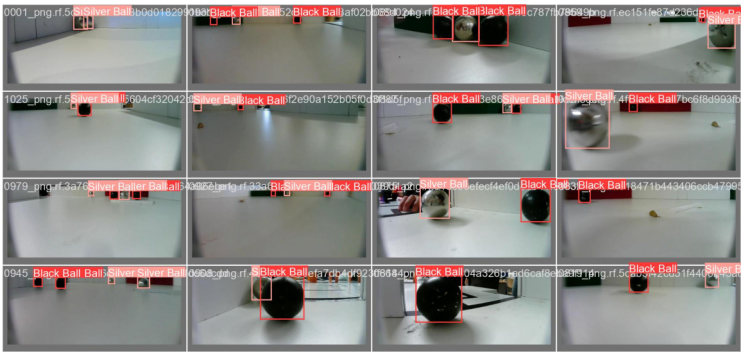


Figure 57: Test prediction with the new AI

by Marius

Date: 2024-01-13

Began working on the evacuation zone program

Tasks done today:

- Implemented the code Marius had written so far for rotating to and picking up victims, and fixed some simple errors in reasoning that occurred because Marius could not test the code, such as different angle ranges being used for the same angle, causing the robot to stop moving if a victim is at the edge of the frame
- Tested and made sure that the old method of detecting the silver tape by searching for very bright reflections of our LED's will definitely not work because of the shorter LED bar compared to the old robot
- Tested new ideas of detecting the silver tape by using a different color-space than RGB. HSV seemed promising at first glance as seen in Fig. 58. This attempt also turned out to be useless, as black insulating tape looks very similar to silver tape in both RGB and HSV with our bright light
- Re-implemented simple orientation at the entrance to the evacuation zone by rounding the robot's current rotation angle to 90° increments and rotating to that value
- Added a function to turn the LED's on/off using our Relay (we turn them off during the evacuation zone to get better images from the lower camera)

Issues and solutions:

Issues	Solutions
Reliable detection of the silver reflective tape does not work with any of the methods tested so far	Trying to use AI detection for the silver tape as well seems to be our only promising option left at this moment

Conclusion / further planning:

Although I worked a lot on the robot today, the results are sobering. In order to finish everything necessary to do well in Hanover, Marius and I should collect and divide up the remaining tasks as best we can. Without a working silver detection, however, there is no point in continuing to work on the evacuation zone for the time being, which is why i will first take care of obstacles avoidance while Marius tries the approach with AI silver detection.

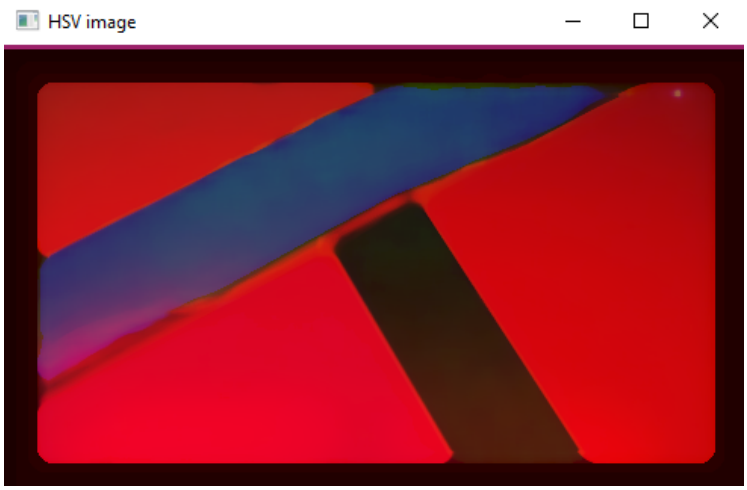


Figure 58: HSV image of silver reflective tape

by Tim

Date: 2024-01-14

Test the idea of using an AI for silver detection

Tasks done today:

- Test the possibility of using an AI for silver detection
- Simple python implementation of Bitflip's silver AI for basic testing
- Begin working on a YoloV8 classification model by taking images of the silver line and regular line
- Trained the first classification model with Google Colab

Issues and solutions:

Issues	Solutions
We still have no way of detecting the silver strip in a reliable manner. Regular image processig methods do not work due to the silver strip being detected as a regular black line.	Using the silver AI from the German team Bitflip, we were able to confirm that the possibility of using an AI to detect the silver strip is possible.
The first YoloV8 classification AI always detected the probability for silver to be 1, so 100% even if there is no silver visible, effectively rendering the model useless because we can't differentiate between silver and non-silver.	The cause of that seems to be that Ultralytics uses softmax to distribute the probabilities between multiple classes and because there is only one, it will always be 1. I trained a new model with a 2nd class called "None" which includes images of the line. The results of the new model were much better, but definitely lacks training data. Fig: 59 shows a example of a image where the AI predicted it to be silver instead of none.

Conclusion / further planning:

After seeing the basic silver model from Bitflip work, we began working on creating a dataset so we can train a YoloV8 classification model. We chose this type of model because we only care about if there is a silver strip in the image or not. Additionally we chose to use a Ultralytics YoloV8 model because we already use them for our victim detection model. We captured around 1k images and annotated them with Roboflow. We chose to train the model on Google Colab with a Nvidia Tesla T4 GPU because we don't have a lot of images. Additionally we chose a image size of 96x96 pixels because we need to run the model on the CPU, requiring very low inference times.

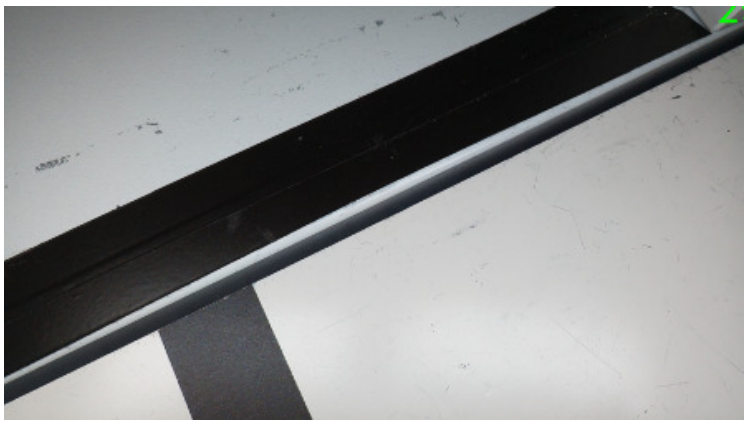


Figure 59: Silver AI failure

by Marius & Tim

Date: 2024-01-15

Creating a to-do-list until the qualification tournament in Hanover

Tasks done today:

- Test the robot thoroughly
- Create a to-do-list with things to fix or change for the local competition in Hanover
- Tim captured a few hundred more images of the black line as well as the silver tape to improve our existing training data

Issues and solutions:

Issues	Solutions
None	None

Conclusion / further planning:

After testing the robot on our school owned test parkour and Tim's parkour at home, we determined the following things that need to be done:

- Implement the avoiding of a obstacle on the course, we currently don't have anything for that
- Fix the front infrared sensors sometimes displaying a distance of 0, could be a electrical issue
- Improve silver detection, the current AI is not very good, probably needs more data
- Implement that we orientate the robot at the silver line, as we are often not straight in front of the evacuation zone
- Fix tipping over when driving over the seesaw
- Fix tipping over when transitioning into driving down a ramp, because we are sometimes too fast and the robot's center of gravity is quite high

- Implement detection when the robot is stuck somewhere in the course and a un-stuck method
- Improve driving over T-intersections, sometimes the robot weirdly turns and we think it could become a issue
- Implement a auto start of the program when the Raspberry Pi 5 crashes, it rarely happens but we want to have a backup if it happens
- Implement orientating the robot to the victims in the evacuation zone
- Implement orientating the robot to the corners in the evacuation zone
- Implement entire procedure for dumping the victims in the corners
- Implement finding the evacuation zone exit

A lot of things still to do, but not all are necessary to score a lot of points in runs, for example finding the exit is not as important as collecting and dumping the victims in the corners, because we can still get a multiplier of ~1,95. Some other things also aren't as important like fixing the tipping over and T-intersection because they only sometimes happen and we can always call a LOP if it does. We hope to get most of these done before the competition in Hanover.

by Tim & Marius

Date: 2024-01-16

Implementing obstacle avoidance

Tasks done today:

- Added our old obstacle avoidance code and modified it to use all three distance sensors in the front
- Added orientation in front of any obstacle by turning either left or right until the obstacle is no longer detected by the center distance sensor. With this orientation, we can turn for another 0.3 seconds and then start driving a curve around the obstacle, which works with obstacles of any width now
- Fixed the issue of randomly detected obstacles as described below
- Captured ~1000 images of the black line and the silver tape for improving the AI

Issues and solutions:

Issues	Solutions
Sometimes very small distance values are output by the distance sensors in the front, even though there is nothing in front of them	The problem probably came from the fact that a small part of the front panel (see Fig. 60 for an illustration) was very close to the measuring range of the sensors, screwing nuts behind the sensors so that they protrude further out solved this problem, which we had been struggling with for a long time.

Conclusion / further planning:

Despite some as yet uncorrected errors, the robot should now theoretically be able to complete an entire line part of the course. However, especially T-intersections and ramps need more testing and improvement and the evacuation zone will have to be worked on anyway as soon as it becomes clear that the new silver method works.

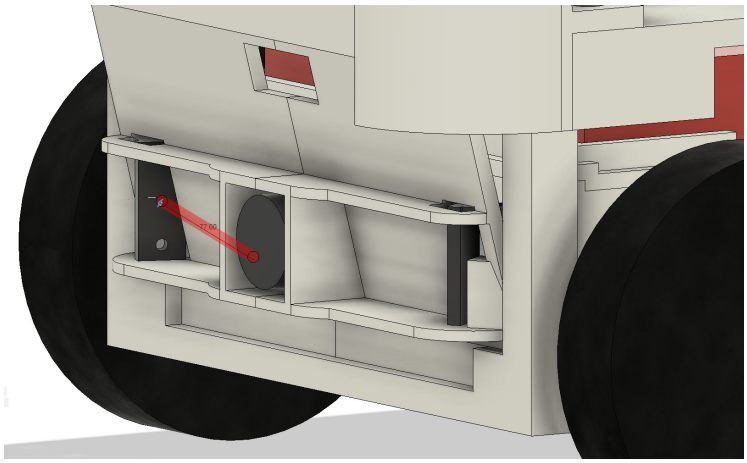


Figure 60: Distance sensor issue illustrated in CAD

by Tim

Date: 2024-01-17

New iteration of the silver classification AI

Tasks done today:

- Cleaned up the silver dataset by removing bad images
- Added new images to the dataset that Tim captured
- Trained a new YoloV8 classification model on the new dataset with a Google Colab T4 instance
- Began implementing the silver model into the line following code
- Tim: Captured more images at test parkour

Issues and solutions:

Issues	Solutions
As previously described the silver model doesn't perform well at the moment, predicting the wrong class quite often. The issue is especially present when there are gaps between the tiles.	Training the AI on the new dataset that is roughly 800 images larger, we saw improvements in those areas, which is great. Additionally we chose to train the model on a image size of 128x128px because the model was able to run at 250+ FPS with a image size of 96. The larger image size should also improve detection accuracy.

Conclusion / further planning:
We are happy with the performance of the new model and began a rough implementation in our line following code with the addition of required timer arrays and the prediction code being added to the cam loop. We still need the model to be more reliable though, so Tim already began capturing more images on his test parkour at home.

by Marius

Date: 2024-01-19

Discovered the guide for the new victims rules

Tasks done today:

- Discovered the new guide for creating victims here
- Create a few victims according to the guide, the result can be seen in Fig. 61
- Test the victim AI on the new
- Capture images of the new victims and add to the dataset
- Train new AI on the new dataset

Issues and solutions:

Issues	Solutions
The AI does detect the new victims, but much worse than on the old victims, probably because of the bumps and uneven texture of the silver victims seen in Fig. 61. The detection is mostly fine on the black victims, but is not on par with the regular black victims.	We captured images of the new victims and trained another yolov8n AI on the new dataset. The results did improve, but made the detection of the old victims slightly worse.

Conclusion / further planning:

When we discovered the guide on the new victims, we were confused because the new victims are nothing like the old ones, which everyone has adapted to. We do like the change that victims are now filled with some material that they can't roll around, but we would have liked it more if the guide was about how to modify existing victims to the 2024 rule book. Adapting the AI to the new victims went well, but we don't know if we can expect them in official competitions. Additionally we still need more training data.

Figure 61: New and old victims

by Tim & Marius

Date: 2024-01-20

Fix the T intersection problem & add sharp turn detection

Tasks done today:

- Added code to detect sharp turns like the one seen in Fig. 62 (see more detailed description below)
- Fixed a problem where the robot sometimes ignored green markers at T intersections (see more detailed description below)

Issues and solutions:

Issues	Solutions
On very sharp curves, as shown in Fig. 62, the robot often loses the line and simply drives straight ahead (along the red line)	Added detection for sharp turns like this one by looking for a contour of the black line that touches the bottom edge of the camera frame at two different positions (we look for a gap between each black pixel in the bottom-most column) that does not touch the top edge of the frame. If these indications are fulfilled, we turn for a certain time on the spot to the side from which the new second line has appeared at the bottom edge
In some edge cases, when the robot approaches a T intersection at a very sharp angle, it will ignore any green markers and go straight. This problem occurs only at very sharp angles because the point on the black line that is furthest to the side of the green marker in the cropped camera image is still rather in the center of the picture causing the robot to go straight.	As the line on the respective side of the green marker is only visible later, when the green marker is no longer visible, it is sufficient to memorize the green marker for a certain time (in our case approx. 0.2 seconds), which solves this issue

Conclusion / further planning:

Even if it is unlikely that a line will be drawn as in Fig. 62 at the qualifying tournament in Hanover, it definitely makes sense to add a solution for such sharp bends for the following events. Due to our pre-baccalaureate exams (Vorabitur) in the next few weeks, we won't be able to work on the robot much, so for Hanover we are now prioritizing only picking up and dropping off victims (we probably won't be able to implement a method to find the exit in time) as well as driving on ramps, which we can only test well at school.

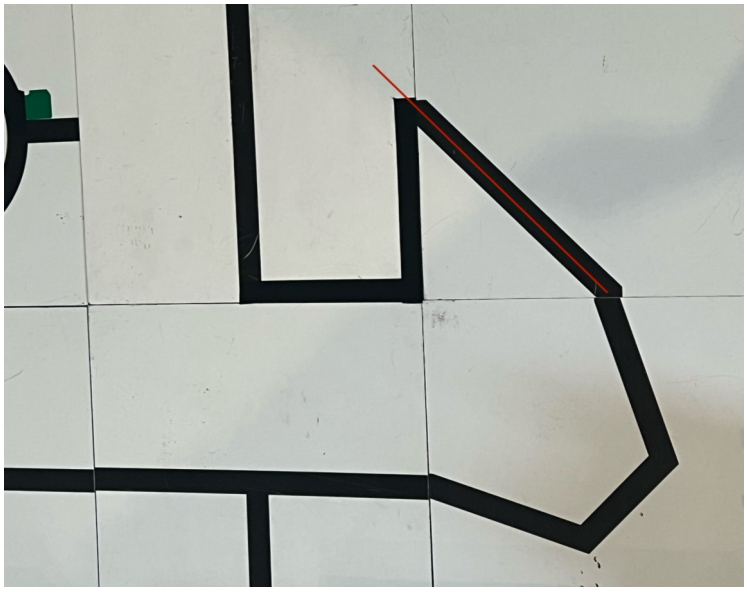


Figure 62: Tile that requires a sharp turn

by Tim

Date: 2024-01-21

Debugging the Coral TPU issue

Tasks done today:

- Debug and document the Coral TPU crashing with a segmentation fault

Issues and solutions:

Issues	Solutions
Every time I try starting a program using the Coral TPU for inference, it begins loading the Coral TPU delegate and then crashes with a segmentation fault.	None yet found.

Conclusion / further planning:

I mostly thought this to be a software issue, not a hardware issue because we did do updates in the past and the Coral TPU did work when we received it. Here is a list of things that I tried and didn't work:

- Python 3.11 which comes preinstalled with Raspberry Pi Os Bookworm
- Python 3.9.2 which comes preinstalled with the Legacy Raspberry Pi Os
- Python 3.9.18 through miniconda
- Backup kernel through the Raspberry Pi's config.txt
- The example script from the GitHub repository of PyCoral
- Python 3.7 through miniconda
- Flashing a brand new Raspberry Pi Os Bookworm image on a SD card
- Trying a brand new Raspberry Pi 5 to rule out the Pi being a issue
- Trying out our old Raspberry Pi 4B with the legacy Os and the new Bookworm Os
- Python 3.10 through miniconda
- Trying a Ultralytics version 8.0.193
- Retraining the model on Ultralytics version 8.0.193
- Trying every supported tflite-runtime version on Python 3.9.2 and 3.11
- Creating a GitHub issue on the Google Coral GitHub repository

So far none of these have worked, though I will not give up on trying to get the Coral TPU working. I am starting to suspect that it is a issue with the libedgetpu library, because it hasn't been updated in more than 3 years. But I will need to postpone investigating this further, due to our pre-baccalaureate (Vorabitur) beginning soon.

by Marius

Date: 2024-01-30

Fixed the Coral TPU

Tasks done today:

- Fixed the constant segmentation fault issue with the Coral TPU

Issues and solutions:

Issues	Solutions
Every time I try starting a program using the Coral TPU for inference, it begins loading the Coral TPU delegate and then crashes with a segmentation fault.	Installing a custom build of the libedgetpu library from here.

Conclusion / further planning:

After many days of debugging the issue and opening or asking in GitHub issues here, here, here and here, I fixed the issue by installing a custom libedgetpu build from here which was created by feranick. Additionally I installed a custom tflite-runtime build for tensorflow 2.15 found here. I am now able to run our YoloV8 victim detection model with more than 50 FPS as seen in Fig. 63. We can now free the Raspberry Pi 5 of running the AI on the CPU, making it possible for our line program to run at more FPS.

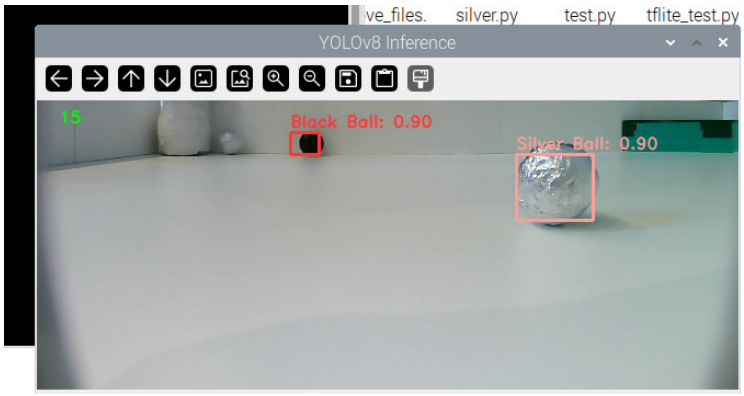


Figure 63: Working Coral TPU inference

by Marius

Date: 2024-02-03

Add better victim prioritisation

Tasks done today:

- Add better victim sorting and prioritisation to zone cam
- Add dump victim methods to the control loop

Issues and solutions:

Issues	Solutions
When there are multiple victims seen by the robot, it sometimes jumps around between the victims because we always target the biggest box, which can cause time loss and the robot missing a victim.	Using python's min function with a key to determine the box that is the closest to the box we originally targeted. Here is a simplified example: `min(boxes, key=lambda x:  abs(x[1]-current[1]))` Current is the box we last selected and boxes is the array of detected boxes.

Conclusion / further planning:

After adding the new prioritisation to the selection of the best box to use, we no longer sporadically jump around targeting different victims, making the orientation up much cleaner and faster. We are going to continue working off the list from the 2024-01-15, getting ready for Hanover.

by Marius

Date: 2024-02-04

Update GUI and picking up victims works

Tasks done today:

- Added detection and guidance method for the red and green evacuation points using OpenCV's inRange() function to binaries the image from the zone camera regarding green pixels in the image followed by the findContours() function as well as the contourArea() function to get each detected contour with its size within the camera frame. If the contour is big enough, we use the boundingRect() function to determine the center of the contour, which the robots try to center while moving towards the evacuation point
- Improved and tested the depositing script Marius which works according to the following procedure
  - drive to the center of the biggest green or red contour in the image
  - if the readings by the distance sensors show a close wall (probably the evacuation point), turn 180°
  - reverse as quickly as possible to align yourself with the evacuation point

- Added a victim counter to the GUI using silver and black circles by adding a canvas to the GUI and using

```
canvas.create_oval()
```

  with different colors. An easy method to draw circles can be found here

Issues and solutions:

Issues	Solutions
I couldn't figure out how to delete anything from the canvas after drawing it, making it difficult to display a lower number of collected victims as before, e.g. after a LOP.	Simply drawing new circles over the old ones in the same color as the background solved the problem.

Conclusion / further planning:
With these new additions, and Marius' confidence that he can fix the silver detection before Hanover, the robot should be able to rescue some victims there.

by Tim

Date: 2024-02-05

Fixing the silver classification AI

Tasks done today:

- Fixed the YoloV8 classification AI randomly detecting white or the line as silver

Issues and solutions:

Issues	Solutions
Sometimes when driving the line, the classification model would randomly detect silver, causing us to go into the evacuation zone mode and causing a LOP.	Adding the argument `erasing=0.0` to the classification training command fixed the issue. More information about the erasing argument and others can be found here.

Conclusion / further planning:

Because of the black boxes that erasing adds to the image through the automated augmentations from Ultralytics, as seen in Fig. 64, our guess is that when it adds a black box over the silver line the model learns silver examples with a black box over them, effectively telling it a black line is silver. After turning that augmentation off and retraining the model using Google Colab, we saw immediate improvements and the model now very rarely detects the line as silver. Those cases can most likely be traced back to bad data, so I will have to check over the entire silver dataset in the future, sorting out bad examples and adding new unique training data.

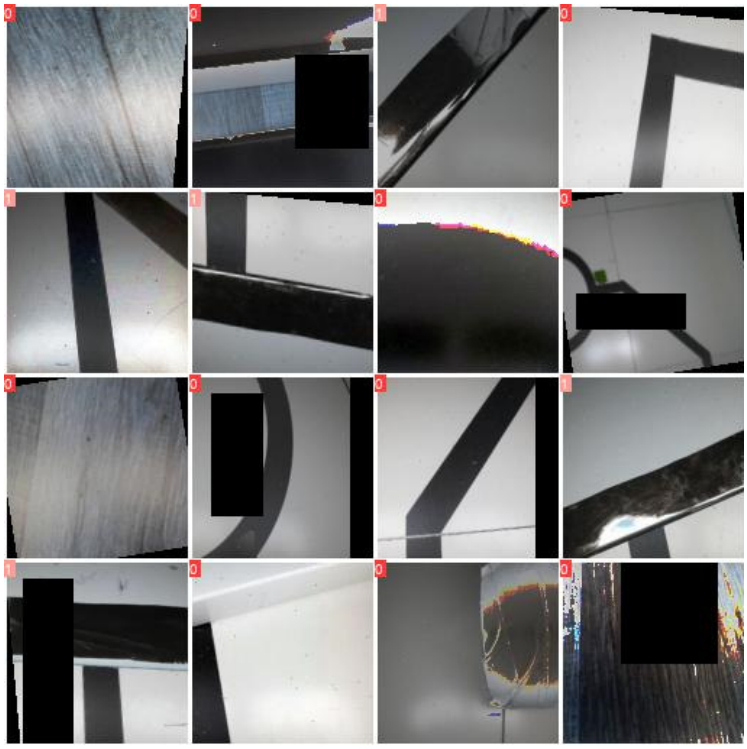


Figure 64: Erasing augmentation examples

by Marius

Date: 2024-02-10

Dealing with the problem of ramps down & silver validation

Tasks done today:

- Tested various techniques, such as driving slower / faster backwards / faster forwards, to avoid rolling over when driving down ramp
- Added a silver validation after the AI model allegedly detects silver tape in the frame

Issues and solutions:

Issues	Solutions
The robot often rolls over when driving down ramps	We have already tried slowing down / going fast backwards / going fast forwards after detecting a ramp down with our gyro sensors, nothing worked perfectly, but going fast forwards is definitely the best solution
The AI still frequently makes mistakes when detecting silver	Adding a validation by comparing the average black area with the LED's on (the silver reflective tape looks black) and with the LED's off (the silver reflective tape now looks bright, but the black line still looks black) expecting a smaller black area if the model is working correctly
The Raspberry Pi sometimes crashes for no apparent reason	Not fixed yet but probably power supply related (since we would have to disassemble the whole robot to modify the power supply, we can only hope that the Pi does not crash during a run)

Conclusion / further planning:

At the moment it seems that adding additional weights is the only possible solution to the problem of overturning. However, we would need a lot of these as our robot is very heavy (~3 kg), but we have absolutely no space for them. It is unrealistic that we will be able to solve the problem before Hanover, as weights would have to be ordered first. Ramps with a maximum incline of 25 degrees have also seemed to work without problems for the most part, but some of them give way under the weight of the robot, which is why we can only hope for good ramps in Hanover.

by Tim

Date: 2024-02-12

Writing a guide for the Coral TPU

Tasks done today:

- Wrote a guide for the Ultralytics documentation on how to get the Coral TPU working with a Raspberry Pi

Issues and solutions:

Issues	Solutions
While going through the issues with the Coral TPU, I saw in many GitHub issues that a lot of people had issues with the Coral TPU on a Raspberry Pi or other SBC's.	Writing a tutorial on how to get the Coral TPU working with Raspberry Pi's and creating a pull request that merges it into the Ultralytics GitHub repository.

Conclusion / further planning:

Since I saw that many people had the same or similar issue and had gained a lot of experience about the Coral TPU, I wanted to help those people out by writing a guide/tutorial ranging from installing a fixed libedgetpu version to exporting and running the YoloV8 model. This is, because I don't think any inexperienced YoloV8 user can be expected to go down the same rabbit hole as I did to get the Coral TPU working. The guide can be found here.

by Marius

Date: 2024-02-17

Creating new logos

Tasks done today:

- Got a friend to create new logos for our new team name, the result can be seen in Fig: 65

Issues and solutions:

Issues	Solutions
We still have the old logo for our Discord server, YouTube account and GitHub organisation, which doesn't fit to our new name anymore	I asked a friend of mine, who isn't in the team, to design new logo's based on the old design, as we did like that design.

Conclusion / further planning:

We are happy with the new simple design and have changed all the referenced accounts to the new design. We also have each one with a translucent background, mainly used for the YouTube watermark and logo in the beginning.

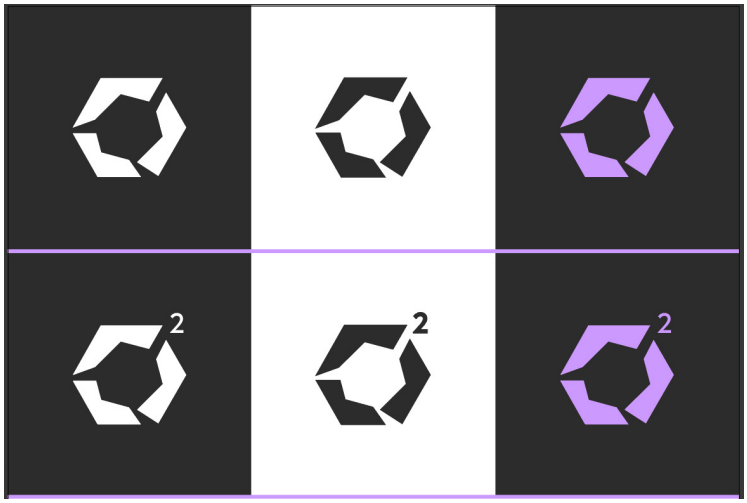


Figure 65: The new logos

by Marius

Date: 2024-02-18

Auto start script creation

Tasks done today:

- Created a auto start script with automatic logging, in case the Raspberry Pi 5 crashes

Issues and solutions:

Issues	Solutions
In rare instances our Raspberry Pi 5 crashes due to extreme power spikes and we have no way of restarting the program, since that would require a few screen interactions, which is forbidden by the rules. That would effectively ruin a run and we'd have to end it prematurely	Using the .desktop file format supported by Raspberry Pi OS Bookworm to automatically start the program once the Raspberry Pi 5 boots up, since the usage of an additional power button is allowed.

Conclusion / further planning:

Thanks to this guide on medium, I was able to figure out how the auto start files work on Raspberry Pi OS. I have also created a script that automatically creates the file with the execution path adjusted to where it was executed from, so we are able to clone the repository anywhere. The .desktop file executes a "start.sh" script that starts the Python script and automatically writes everything that gets printed out into a log with the current date as its name, since I was unable to find a way to start a console with a command being auto executed. A correct .desktop file looks something like this:

```
[Desktop Entry]
Name=Robot
Type=Application
Exec=bash -c 'cd <script directory> && ./start.sh'
```

by Marius

Date: 2024-02-19

Begin of the Hanover competition

Tasks done today:

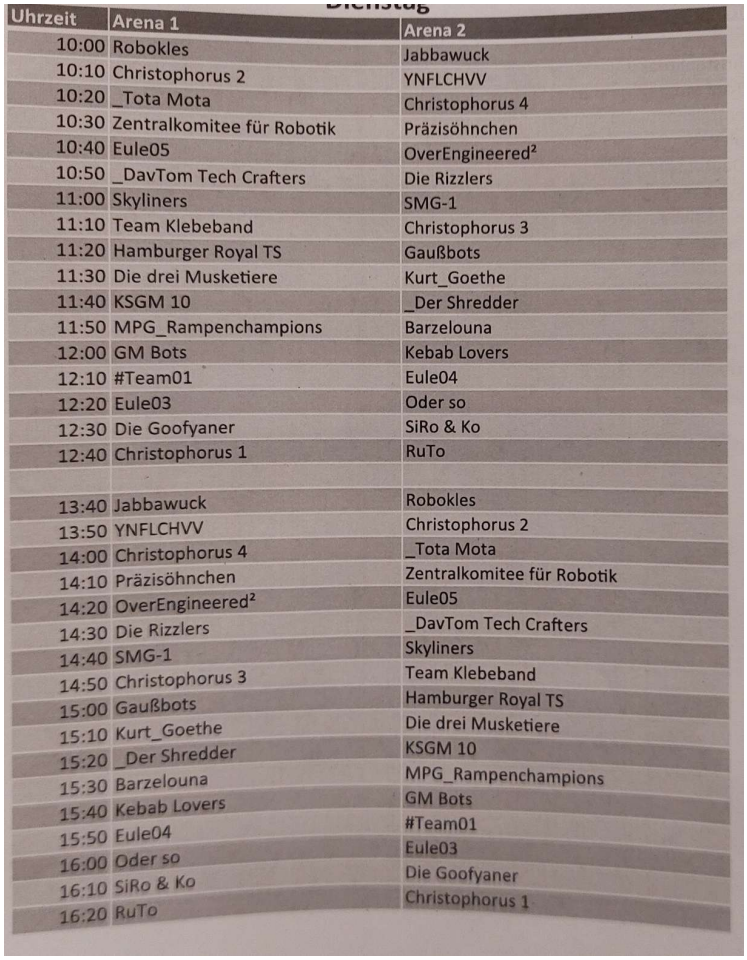
- Arrived at the Hanover competition
- Took a look at the other teams participating and their robots
- Prepared for the first runs on the next day, the plan can be found in Fig. 66
- Fixed a small logic error in the gap program that caused the robot to get stuck in the gap orientation
- Captured a few hundred training images of silver tape and black line

Issues and solutions:

Issues	Solutions
When driving around obstacles in a curve (one side of the tires turns faster than the other), the robot often drives too far behind the obstacle, where there no longer needs to be a line	Although the motors cannot turn any slower or faster than they currently do, we have added a turn on the spot at a time interval of 0.3 seconds and then continue on the curve, solving this issue very efficient

Conclusion / further planning:

The arrival at the competition in Hanover went smoothly and the robot works fine. We are confident that we will score a high result and hopefully even win. Though the organisers did manage to spell our name wrong (It's Overengineering² not OverEngineered²).



Dienstag

Uhrzeit	Arena 1	Arena 2
10:00	Robokles	Jabbawuck
10:10	Christophorus 2	YNFLCHVV
10:20	_Tota Mota	Christophorus 4
10:30	Zentralkomitee für Robotik	Präzisöhnchen
10:40	Eule05	OverEngineered²
10:50	_DavTom Tech Crafters	Die Rizzlers
11:00	Skyliners	SMG-1
11:10	Team Klebeband	Christophorus 3
11:20	Hamburger Royal TS	Gaußbots
11:30	Die drei Musketiere	Kurt_Goethe
11:40	KSGM 10	_Der Shredder
11:50	MPG_Rampenchampions	Barzelouna
12:00	GM Bots	Kebab Lovers
12:10	#Team01	Eule04
12:20	Eule03	Oder so
12:30	Die Goofyaner	SiRo & Ko
12:40	Christophorus 1	RuTo
13:40	Jabbawuck	Robokles
13:50	YNFLCHVV	Christophorus 2
14:00	Christophorus 4	_Tota Mota
14:10	Präzisöhnchen	Zentralkomitee für Robotik
14:20	OverEngineered²	Eule05
14:30	Die Rizzlers	_DavTom Tech Crafters
14:40	SMG-1	Skyliners
14:50	Christophorus 3	Team Klebeband
15:00	Gaußbots	Hamburger Royal TS
15:10	Kurt_Goethe	Die drei Musketiere
15:20	_Der Shredder	KSGM 10
15:30	Barzelouna	MPG_Rampenchampions
15:40	Kebab Lovers	GM Bots
15:50	Eule04	#Team01
16:00	Oder so	Eule03
16:10	SiRo & Ko	Die Goofyaner
16:20	RuTo	Christophorus 1

Figure 66: The order of the runs

by Tim & Marius

Date: 2024-02-20

2nd day of the Hanover competition

Tasks done today:

- A lot of testing and adjusting timing/calibration variables
- Complete the first two runs
- Got interviewed by SAT1 Regional

Issues and solutions:

Issues	Solutions
We still can't exit the evacuation zone and entering the evacuation zone did fail a few times, because the robot turned into the wrong direction	Marius: Writing a quick method that drives along the horizontal and vertical axis and checks if there is a sudden big distance to try solve the exiting of the evacuation zone
The robot sometimes rolls over while driving ramps down even on the training course	Tried changing the speed and time of the faster forward motion with no noticeable improvement until I noticed that the ramp down increase was greater than 25 degrees (as the ramp on the competition course had a gradient of exactly 25 degrees, we had no problems here at the end)
The robot mistakenly did not go straight when crossing an X-intersection without a marker, but instead turned left	Not yet found, should be fixed before Kassel

Conclusion / further planning:

The first two runs of the went well and the first one is already up on YouTube. The video can be found here. We don't know yet when our next run is tomorrow. The method Marius wrote for exiting the evacuation zone is very simple, but can't hurt to give it a try, we may complete a run with a successful exit. We were also interviewed about our robot by SAT1 Regional and the interview/story should be published in a few days.

by Tim & Marius

Date: 2024-02-21

3rd day of the Hanover competition

Tasks done today:

- Completed the 3rd run

Issues and solutions:

Issues	Solutions
None	None

Conclusion / further planning:

In the end we won the competition with a 404 point lead and advanced to the German championship in Kassel, which will be in a few months, around when we are writing our Abitur. The scores can be found here. We are very happy with the robots performance, but we still need to improve a lot of things to be able to compete properly at the German championship in Kassel. The interview by SAT1 Regional is also published already and can be found here. Lastly we are happy that finding the exit in the last run actually succeeded, earning us some more points because we don't get the multiplier reduced by a lot.

Uhrzeit	Arena 1	Arena 2
10:00	Robokles	Jabbawuck
10:10	Christophorus 2	YNFLCHVV
10:20	_Tota Mota	Christophorus 4
10:30	Zentralkomitee für Robotik	Präzisöhnchen
10:40	Eule05	OverEngineered²
10:50	_DavTom Tech Crafters	Die Rizzlers
11:00	Skyliners	SMG-1
11:10	Team Klebeband	Christophorus 3
11:20	Hamburger Royal TS	Gaußbots
11:30	Die drei Musketiere	Kurt_Goethe
11:40	KSGM 10	_Der Shredder
11:50	MPG_Rampenchampions	Barzelouna
12:00	GM Bots	Kebab Lovers
12:10	#Team01	Eule04
12:20	Eule03	Oder so
12:30	Die Goofyaner	SiRo & Ko
12:40	Christophorus 1	RuTo

Figure 67: Time of the last run

by Tim & Marius

Date: 2024-03-03

Buying a sim card router

Tasks done today:

- Bought a Cudy sim card router for the German championship in Kassel

Issues and solutions:

Issues	Solutions
We always have to use a mobile hotspot, or the router of our school which doesn't have internet access. That is inconvenient because we often require a VNC connection through RealVNC and using a mobile hotspot can be unreliable. Additionally we need internet access because our code is hosted in a GitHub repository.	Buying a router with a sim card slot, so we can buy a data only sim card, when the competition in Kassel arrives, to access our GitHub repository or other useful resources directly on the Raspberry Pi 5. Having a dedicated router should also fix the unreliability and range issues that mobile phones typically have.

Conclusion / further planning:

The new router should be able to solve all the reliability issues that we have with internet while at competitions. The router having a sim card is also convenient because we can then directly connect to the cellular network and don't have to waste our mobile data on downloading our GitHub repository.

by Marius

Date: 2024-03-06

Creating a cooler for the Coral TPU

Tasks done today:

- Used Fusion 360 to create a cooler for the Coral TPU in CAD
- 3D printed the design with Tim's Bambu Lab X1-Carbon for a high quality result, seen in Fig. 68

Issues and solutions:

Issues	Solutions
We want to use a bigger AI model with the Coral TPU for increased accuracy, but we are quickly bottle necked by the Coral TPU's limited speed. This could be solved by installing the "max" version of the libary which essentially overclocks the Coral TPU, but we there run into thermal issues where the Coral TPU downclocks because of overheating.	Since I had some small fans from a older Cooler for the Raspberry Pi 4B laying around, I designed a cooler that fits ontop of the metal heatsink for the Coral TPU and houses two small 5V fans. After printing the design we are now able to run the Coral TPU at increased clock speeds.

Conclusion / further planning:

The new cooler works nicely and keeps the Coral TPU cool even when it is running under full load for longer periods of time. The 3D model is downloadable from this MakerWorld page with a guide that includes what parts will be needed. The 5V fans simply get plugged into a 5V GPIO pin of the Raspberry Pi 5.

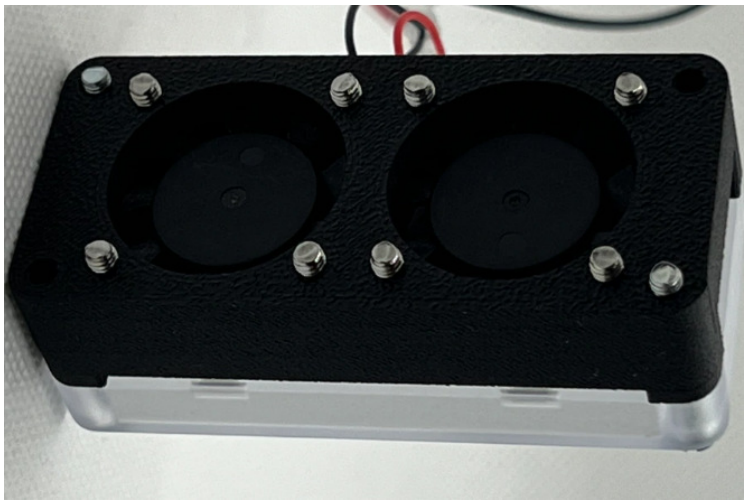


Figure 68: The cooler mounted to a Coral TPU

by Marius

Date: 2024-03-10

Add stuck detection & fixed the robot rolling over

Tasks done today:

- Marius: Added a method that uses scikit's structural_similarity to determine if the robot is stuck
- Tim: Simplified the existing method and added it into a timer array that checks the similarity every 0.5 seconds

Issues and solutions:

Issues	Solutions
Because of the immense size of our robot we sometimes get stuck on the edges of the poles under a ramp, most times resulting in a LOP.	Using scitkit-image's structural_similarity method to calculate the similarity of two images every 0.5 seconds and add the result to a timer array. We calculate the similarity with a image from half a second ago and the current time, because this way we avoid calculating the similarity for two almost identical images. If the average of the timer array is over 0.9, the robot performs a wiggle maneuver to try and unstuck itself.
The robot is rolling over some times while driving ramps down (as already stated here, but not fully solved yet)	Found a logic error between the detection of ramps down via the camera (large black area in the image) and via the gyroscope sensors causing the robot to accelerate to normal speed while standing just before a ramp down. With this error fixed we are now able to drive really slow at ramps down, preventing any rolling over

Conclusion / further planning:

Through this article we found out about the scikit-image library and its structural_similarity method. The end result works great and is able to unstuck itself almost every time. If it failed the next try often works. This looses us only about 15 seconds, which is much preferred over a LOP.

by Marius & Tim

Date: 2024-03-18

Fixed the bottom camera returning no image sometimes

Tasks done today:

- Tried to automatically reconnect the USB camera when it returns a NONE (without success)
- Rewrote the bottom camera initialization this time using Picamera2 instead of OpenCV

Issues and solutions:

Issues	Solutions

When we started the program, in some cases the bottom USB camera seemed to return a NONE, causing an error in the first line where we expected to see an actual camera image (we never bothered to fix it because it did not always happen and once the program was running it was no longer a problem)	We were able to rule out a loose contact or similar hardware problems, since the camera works fine on a PC, so the software we use to get the camera image must be the problem. Until now we had simply used the OpenCV function
	`camera = cv2.VideoCapture(0)`
	to get the camera image. When we realized that you can use any USB camera with Picamera2, which we already use for the Raspberry Pi Camera Module V3, we decided to test this library's function
	`camera = Picamera2(1)`
	to get the camera image. Since this rewrite the error has not occurred anymore

Conclusion / further planning:

This bug hasn't cost us any points yet, but it was quite annoying when debugging, so it's a great relief that we were able to fix it so easily.

Apart from finding the exit of the zone, our biggest problem until kassel is now to prevent the random crashes of the Raspberry Pi, which could cost us the points of an entire run.

by Tim

Date: 2024-03-20

Fix the Raspberry Pi 5 crashing

Tasks done today:

- Had collected a number of possible reasons for the crashes of the Raspberry Pi 5:
  - Connecting two step down converters in parallel does not result in double the current (so we would only have 5A available for the Raspberry Pi, all servos, the Arduino's, display and TPU) (most likely)
  - Some strange error with the TPU or the Raspberry Pi OS (possible)
  - A loose contact somewhere in the power supply (excludable by testing)
- Selected and ordered a new 12 A step-down converter, since this was the most likely cause of the problem
- Ordered a small cooling fan since the step-down converter requires active cooling according to its documentation
- Once again disassembled half the robot to replace one of the step down converters with this one as seen in Fig. 69
- Designed and printed a new side plate with a cutout for the fan and mounting holes for four threaded inserts securing it

Issues and solutions:

Issues	Solutions
Step-down converter is big and requires active cooling but the space inside the robot is limited	Fixing the step-down converter lying on its side with adhesive tape, cutting a part out of the battery compartment above it and integrating the cooler into one of the side plates as seen in Fig. 69

Conclusion / further planning:

As expected, it took quite a long time to disassemble the robot this far (about 8 hours), but after the first test, the crashes seem to have been fixed, which fully compensates for this effort.

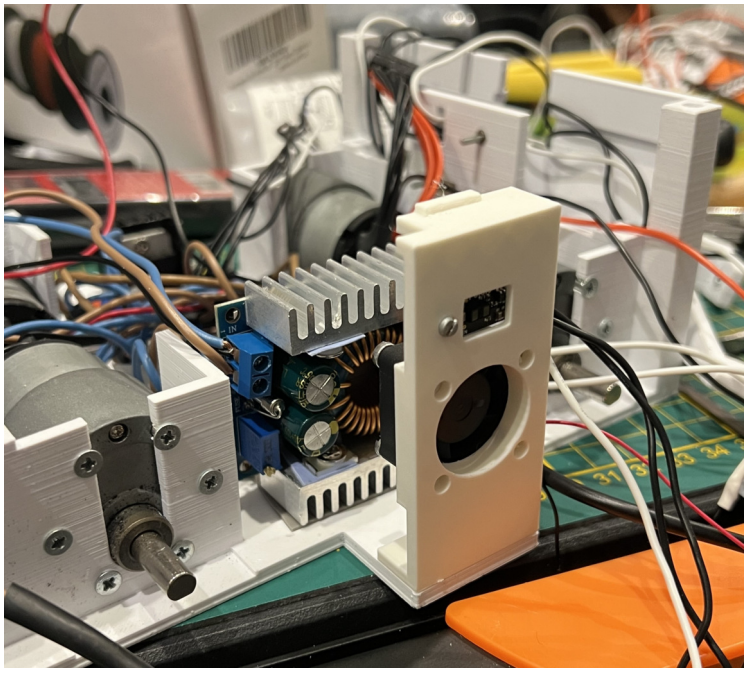


Figure 69: Installed the new 12A active cooled step-down converter

by Tim

Date: 2024-03-23

Added a config manager & a new handle

Tasks done today:

- Added a config manager class that can save and load values from/to a config file

Issues and solutions:

Issues	Solutions
We have no way of easily modifying the values and going through the python files takes time. Additionally Tim wants to add color calibration soon so we need a way to save the calibrated values.	Using Python's configparser module from the standard library, I was able to create a class that can load floats, integers, Boolean's and lists from and to a config.ini file. After consultation with the organizer of the qualifying tournament in Hanover, we have added a new handle made of wire and insulating tape to the robot, which should make it easier to catch the robot quickly (see Fig. 70)

Conclusion / further planning:

Thanks to the simple configparser module, writing this didn't take too long. A example config file looks like this:

```
[color_values_zone]
green_min = [35, 195, 24]
green_max = [85, 255, 154]
red_min_1 = [0, 149, 112]
red_max_1 = [10, 255, 255]
red_min_2 = [170, 149, 112]
red_max_2 = [180, 255, 255]
```

This should be very useful for Tim when he integrates the planned color calibration for the evacuation zone.

Figure 70: Added handle made of wire and insulating tape

by Marius&Tim

Date: 2024-03-24

Added color calibration for evacuation points

Tasks done today:

- Added function to find average color within an image using NumPy:

```
avg_color_per_row = np.average(image, axis=0)
avg_color = np.average(avg_color_per_row, axis=0)
```

- Added a color calibration button to the GUI
- Added a colour calibration interface to the GUI, which first displays a rectangle for positioning the evacuation point, after confirmation of the image section, the zone camera switches to a binarized image using the new average colours. This new average color is saved in the config file and will be used for evacuation point detection from now on (see Fig. 71)

Issues and solutions:

Issues	Solutions
The green/red shades of the evacuation points vary greatly in different competitions and lighting conditions, making it difficult to find values that still detect green/red in all situations, but at the same time do not detect shadows as such	Implementing an easy and quick-to-use color calibration that determines the correct color in a narrower range

Conclusion / further planning:

After a number of tests where victims were dropped in the wrong place because a shadow was interpreted as an evacuation point, this new solution saves a lot of time at competitions and makes such mistakes much less likely.

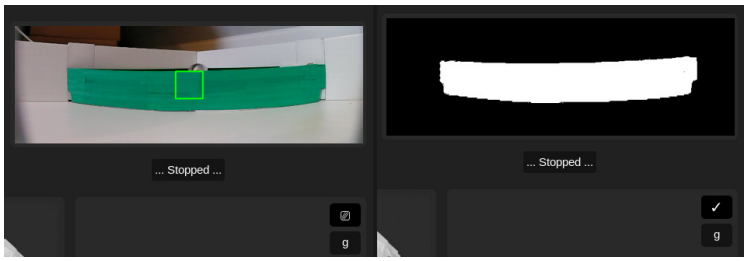


Figure 71: Color Calibration Interface

by Tim

Date: 2024-03-25

Fix X-intersection path selection

Tasks done today:

- Adjusted image processing color scheme from red/yellow to shades of blue
- Fixed the issue found here regarding the path selection at X-intersections

Issues and solutions:

Issues	Solutions
As you can see on the left in Fig. 72, the point (blue) that the robot follows at an X-intersection is incorrectly selected when the robot approaches it at an angle.	Selection between the two sections at the top of the frame is no longer by the distance to the last position of the blue dot, but by the distance to a new theoretical yellow dot, calculated by extending the centre of the line at the bottom of the frame (light blue) and the centre of the line approximately at the centre of the frame (red dot), as shown in Fig. 72

Conclusion / further planning:

Like the blue dot previously, both the light blue dot and the red dot are calculated by calculating the centre between the leftmost and rightmost white pixels of the binarised line image with NumPy within a given row. The theoretical yellow dot (see Fig. 72) is then calculated by extending the light blue (bottom_point) dot through the red dot (poi) to the upper edge of the frame using the function

```
if bottom_point[0] != poi[0] and bottom_point[1] != poi[1]:
  slope = (bottom_point[1] - poi[1]) / (bottom_point[0] - poi[0])
  x = min(max(poi[0] + (0 - poi[1]) / slope, 0), camera_x)
else: x = poi[0]
```

With this fix, the robot will now take the correct path regardless of the angle at which it arrives at an x-intersection.

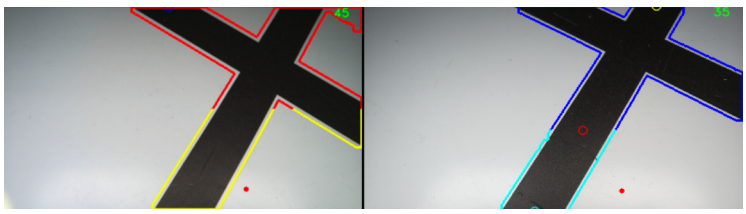


Figure 72: Left before intersection selection fixed; Right after intersection selection fixed

by Tim

Date: 2024-03-27

Complete rewrite of the evacuation zone exit search

Tasks done today:

- Decided against the approach of tracking the robot's position inside the evacuation zone entirely using the distance and gyro sensors, as this seems like a lot of work with only a few benefits
- Added and debugged a new wall following approach using the distance sensor on the right side of the robot according to the following procedure
  - 1. Unload the dead victims
  - 2. Turn 90° right
  - 3. Drive forward until the front distance sensors detects a wall or the camera detects black or silver tape
  - 4. If a wall is detected, turn 45° left otherwise turn 45° right and check if the tape is silver or black by comparing the size of the black area with the LED's turned on and off, expecting a larger difference if the tape is silver (similar to our silver validation), drive out if the tape is black and turn 90° left if the tape is silver
  - 5. Drive forward until the front distance sensors detect a wall, the line camera detects black or silver tape, the line camera detects red or green or the right distance sensor detects an exit
  - 6. If a wall is detected, turn 90° left and proceed as described in 5.; if black or silver tape is detected, proceed as described in 4.; if red or green is detected, turn 60° left and drive curved around the evacuation point; if an exit is detected, turn 90° right and proceed as described in 4.

Issues and solutions:

Issues	Solutions
If there is an exit right after an evacuation point, the robot will stand in the exit at a 45° angle and simply get stuck on the wall	Turning 60° after a evacuation point is detected and driving curved around it results in the robot standing directly in front of the wall or the exit after the curve

Conclusion / further planning:

With this approach, the robot reliably finds the exit and even obstacles in the middle of the zone should pose no problems. This approach is also much simpler and less error-prone than position determination, thus perfect for a task. A video of the approach is available here.

by Tim

Date: 2024-03-29

Overengineering the entering of the evacuation zone

Tasks done today:

- Fixed a logic error in the evacuation zone exit search, causing the robot to turn the wrong direction in some edge cases
- Edited the evacuation zone exit search so obstacles placed at the edge of the zone should be avoided
- Added a function to determine the angle of the silver tape to decide which 90° angle to turn to
  Since the silver AI does not determine the position of the silver stripe, but only whether one is in the picture or not, the position has to be determined in a different way.
  The fact that the silver strip is almost black with the LED on and very bright without the LED on makes it possible to save a binarized image of all reasonably black pixels in the image when the light is on, then turn off the light and subtract the current binarized image of all reasonably black pixels from the saved image using NumPy

  ```
  silver_image = saved_black_image.copy() - black_image.copy()
  ```

  After some noise reduction, you now have an image that contains only the silver tape. By the same approach our gap angle determinations works (as described in Fig. 56) we now calculate the angle of the silver tape and deciding which 90° angle to turn to (find an example of this procedure in Fig. 73).

Issues and solutions:

Issues	Solutions
The robot is often positioned at a great angle at the entrance to the evacuation zone when the silver line is detected, which is why a simple turn to a rounded 90° angle can easily end up in the wrong direction	Determining the angle of the silver tape in the image and based on this decide in which direction to round to 90°

Conclusion / further planning:

With these probably last changes before Kassel, we are pretty sure that we should be able to reliably rescue victims and do well in the rest of the course, as we already showed in Hanover. From now until Kassel, however, preparing for our final exams (Abitur) simply has priority for us which is why we cannot continue working on the robot for the last three weeks until the competition.

Figure 73: Silver tape angle determination

by Tim

Date: 2024-04-19

First day of Kassel and a gigantic problem arises

Tasks done today:

- Arrived in Kassel for the German championship a day late because of our physics Abitur exam
- Find out that the program runs slow after software update

Issues and solutions:

Issues	Solutions
After we ran a automated software update on our Raspberry Pi 5, the program suddenly ran with only 20 FPS instead of the regular 40 FPS. After a minute or so of the program running it would drop down to 7-15 FPS and fluctuate a lot. Without the ability of running the line cam at 40 FPS or more, we would have to slow down the robot significantly and run the risk of time issues.	We were unable to find a fix after a whole night of debugging. We tried reinstalling Raspberry Pi OS Bookworm onto a new SD card, trying the same and new SD card with the 2 backup Raspberry Pi 5's, removing Numba from our program or disabling the AI's. So far nothing was able to fix that issue.

Conclusion / further planning:

Since we arrived late due to our Abitur (high school diploma) exams, where the physics exam is on the same day as the first day of the competition, we had very little time preparing. Thanks to the organisers we are allowed to move all of our runs to tomorrow, giving us some more time to prepare. After a thing as simple as a software update our program suddenly ran at less than 20 FPS. We require at least 35 FPS so the robot can run at the speed we want it to and otherwise we would need to slow the robot down by a big margin. There would also be other complications with turning speed. So far, after debugging through the entire night and Tim getting no sleep (this is written on the 20th because we were busy debugging the issue) and Marius got one hour of sleep, we were not able to fix the issue. The robot is driving line so unreliably now that we are contemplating giving up. But we do still have time so we are going to continue to debug this.

by Tim & Marius

Date: 2024-04-20

Second day of Kassel and fixing the problem

Tasks done today:

- Fixed the big issue from yesterday
- Tested the entire robot and made sure everything still works
- Caught up with the runs by doing 5 in a row

Issues and solutions:

Issues	Solutions
Same issue from previous entry	Marius: Since I had my own Raspberry Pi 5 with me and a SD card with a Raspberry Pi OS Bookworm that hasn't been updated in a while, I plugged that SD card into our "broken" Raspberry Pi 5, the one in the robot and opened "raspi-config" through the terminal and went to "Advanced Options" → "Bootloader Version" → "Default", this resets the eeprom software, which is kind of similar to a BIOS. This somehow fixed the Raspberry Pi 5 permanently and the program now runs like before with 35-40 FPS. I think it is because the eeprom software is the only thing that carries over if we put in a different SD card, since we tried to run the program with my SD card before without changing the bootloader version, which didn't work. It seems that something in the eeprom got changed or corrupted so the Raspberry Pi 5 lost more than half of it's performance.

Conclusion / further planning:

The entire task of debugging and fixing the issue was a gigantic nightmare, firstly because we had written our physics finals on the day of the begin of the runs and arrived a day later without being able to test anything, secondly we only had about half a day to debug the issue, including the entire night and until 16:00 where we would be forced to do our first runs. Thankfully we were able to fix the issue at around midday and could then test the robot for a few hours and fix issues that weren't related to awful performance. We were told we could tell the organisers when we were ready to do our first run after we notified them that we fixed the issue.

Surprisingly the first run we did was a perfect run without any LOP's. We completed this run in 3:47 min with a score of 1427. We did 4 more runs after that which were also really good, but all contained one or two LOP's. The 6th run will be tomorrow. We are currently first in the standings with a sizable lead, but we can still not get first because of the runs that get cut out.
At last, our team wants to thank the organisers for even allowing us to arrive late because of our Abitur exams. Additionally we want to thank them for giving us a lot of time to be able to debug and fix the issue with our Raspberry Pi 5 performance and being very open in general. One last thank you goes out to the teams who were willing to lend us Raspberry Pi 5's and other helping means such as knowledge.

by Tim & Marius

Date: 2024-04-21

Third day of Kassel

Tasks done today:

- Completed our 6th and last run
- Picked up the certificate for the best poster
- Picked up the trophy (see Fig. 74) and certificate for winning the German championship

Issues and solutions:

Issues	Solutions
None	None

Conclusion / further planning:
We are extremely relieved and happy after the last day of the competition. We won the German championship with a 940 point lead and are now qualified to go the world championship in Eindhoven, Netherlands. Even though the last run went poorly because of a obstacle in the evacuation zone, we now know what we need to improve because we can most likely expect them in the world championship. The scores and times of all our runs can be found here. We will upload all our runs on YouTube and they can be found in this playlist.

Figure 74: Our 1st place trophy

by Tim & Marius

3 Appendix

If you want to learn more about our robot, feel free to visit these sites:

- Our YouTube channel
- Our Github oranisation (Code will be available once we finish all our competitions)

References

3.1 Software tools & Platforms

- Arduino
- Github
- Google Colab
- Fusion 360
- Python
- Raspberry Pi OS
- Roboflow

3.2 Libraries

- CustomTkinter
- gpiozero
- NumPy
- Numba
- OpenCV-Python
- scikit-image
- Ultralytics

3.3 Hardware

- 12 V DC Geared Motor
- Arducam B0268 16MP Wide Angle
- Arduino Nano
- 11.1 V/7.4 V Conrad Lipo Batteries
- Diymore 35 kg Digital Servo
- EMAX-ES09MD Servo
- Coral USB Accelerator
- KY-019 Relay
- L298N Motor Driver
- Pololu irs16a/irs17a 50 cm/130 cm Infrared Sensor

- Raspberry Pi 5
- Raspberry Pi Camera Module V3
- revoART, 12V COB LED strip
- XL4015 Step Down-Converter
- XL4016E1 Step Down-Converter
- XL6009 Step-Up Converter

List of Figures

Index

W

Y

Z