CJTec

Team Description Paper
RMRC 2024 World Championship

1.0 Logistical Info

- Team Name: CJTec
- Organisation: CJT-Gymnasium Lauf
- Country: Germany
- Contact person: Rudolf Pausenberger (our teacher) / Elias Schramm (Senior SD)
- Email: r@pausenberger.eu / elias-schramm@web.de

2.0 Introduction

2.1 Who are we?

We are a small team of students with a passion for Computer Science and state-of-the-art technology. In our German High school, the “CJT-Gymnasium Lauf”, you can select “voluntary subjects” and robotics is one of them. The beginner division usually just uses building sets, but since we, the advanced division, are fed up with “just following the manual” and want to implement our own ideas, we created our very own robot called Sam. We created the robot last year after we decided to split the advanced division into two teams with separate robots because of a too-high number of participants.

Our team is quite diverse with an age range from 14 to 18-year-old students. We also include different genders. Furthermore, we also employ a “specialization over generalization” policy, which lets us work more efficiently since everybody has his/her domain to work on.

2.1.1 Participants

Elias Schramm - Senior Developer
Writing code for 4 years, Elias is a force to be reckoned with and an integral part of the team effort. He likes to explore creative approaches and transform concepts from other domains e.g. the car industry to usable concepts for our small robot.

Nele Boghdan - Junior Developer & Driver
Having begun to code just a year prior, she steadily improved and gained proficiency in Python and C++. But most importantly, she uses her stunning talent to drive our robot through every obstacle.

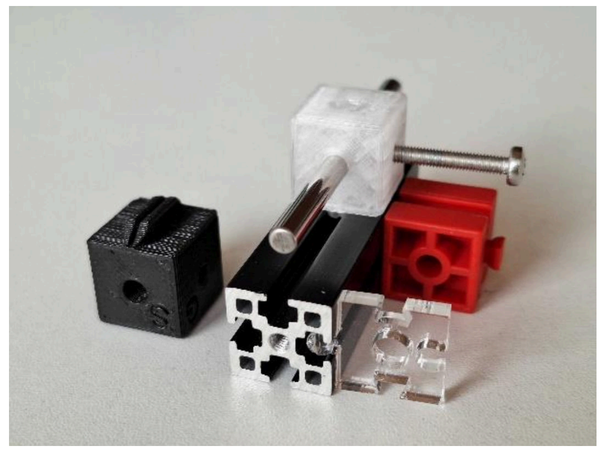
Jannis Arnet - Senior Mechanician
Even though he joined the team just a while ago, he constructed the robot in record time and used his expertise to account for many advanced concepts like for example the center of gravity to avoid toppling-overs.

2.2 What are we doing?

Our team focuses mainly on two aspects: Computer Science and Mechanics.
In the Computer Science domain, we try to enable the driver to do his/ her best. On the one hand, the handling of the vehicle has to be intuitive and easy-to-learn and on the other hand, the driver has to have a good understanding of his/her environment to make sensible decisions. In the Frontend we focus on Driver Assistance Tools (e.g. Motion Detection Tools) and a GUI that adapts to challenges and their specific requirements. The Backend mainly consists of embedded coding where we route data from the sensors to the GUI and vice versa. The routing is done via Ros2.
In the Mechanics domain, we mainly focus on the maneuverability and robustness of our robot. To achieve this, we try to incorporate things like the center of gravity in our design.
Furthermore, Electronics / Power supply plays an important role in our work.

3.0 System Description

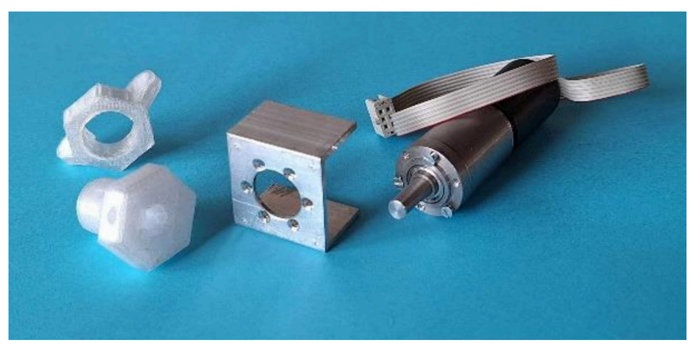
3.1 Hardware



3.1.1 Corpus

For the body of the robot, we essentially use the Makerbeam modular system with a width of 15 mm. This way new ideas can be tried out flexibly. Fischertechnik building blocks are compatible with its profile rails and significantly expand the possibilities that can be implemented quickly. Further practical parts can be created using 3D printing; at http://www.wanderausstellung.eu/robodat.html you will find the print files.

3.1.2 Motors



The motors 2224U018S R IEH2-512 2081 22GPT 44:1 were made available to us through the organization and mediation of Nuremberg by the Faulhaber company free of charge. Unfortunately, according to Faulhaber, there is no commercially available flange, so that we had to rely on our own designs. The flange on the right picture is made by hand, the hubs are printed, see http://www.wanderausstellung.eu/robodat.html.

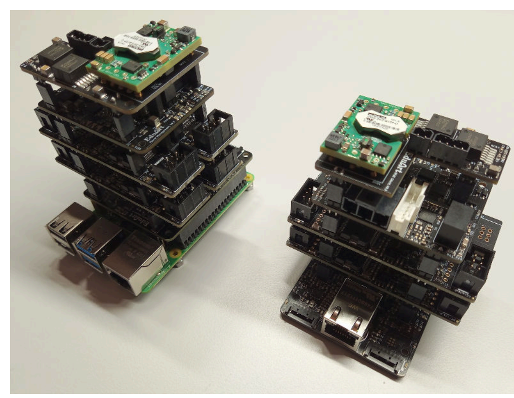
3.1.3 Off-road capability

The aim is to maintain the traditionally good off-road mobility of the robots from our working group in this model as well. We achieve this by using large tires and a concept of appropriate suspension.
In https://www.youtube.com/watch?v=XbbiSpJtkmM you can see a lead block on each of the two front booms. With this additional weight on the front wheels, it became possible to climb steps. The concept of installing the batteries there developed from this experience.

3.1.4 Power supply

16 rechargeable batteries of 1.2 V each provide our supply voltage of 19.2 V. Alternatively, we use blocks of the type Sub-C (d = 23 mm, h = 43 mm) or 4/3A (d = 17 mm, h = 67 mm) including temperature sensor.

3.1.5 Motor Controllers



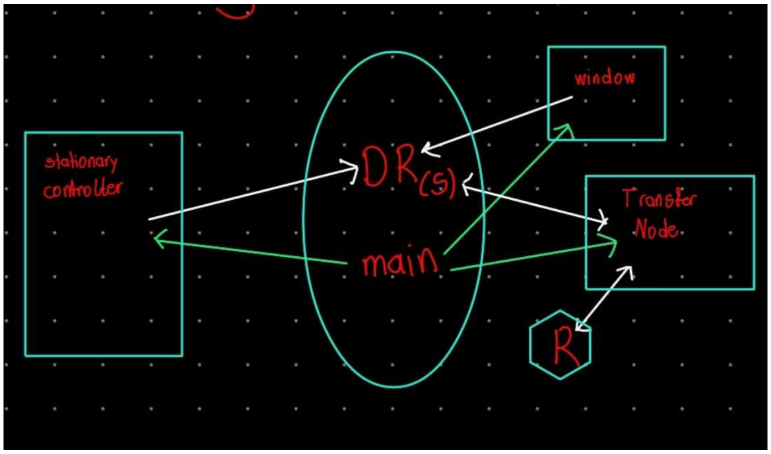
We use the Eduart Free Kinematics Kit as a Motor Controller. It also includes a Power Management Module. It enables us to focus on the things that really matter.

3.1.6 Cameras

Currently, we are using an old usb webcam as our only camera. However, we are in the process of buying two additional cameras for our robot. The old webcam will then cover the back of our vehicle. We will mount a wide-angle camera on top of our vehicle (where the webcam used to be). Furthermore, we also plan on mounting a small camera on the front of our vehicle, since the top-camera doesn't capture the front view in close proximity. This camera has to be quite small since the front of our vehicle already contains our batteries.

3.2 Software

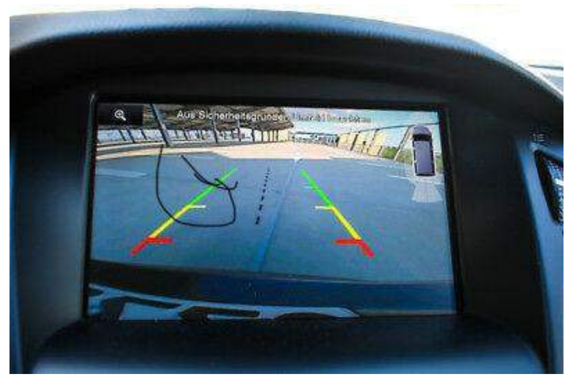
3.2.1 Frontend



We have created the GUI with Dear Imgui (GLFW + OpenGL3). However, it is still under development since the GUI heavily depends on everything else. And we are still unsure whether we can add some modifications in time. Therefore, it just wouldn't make sense to finish it now when there still exist some uncertainties about possible additional hardware. For example, we are currently determining the feasibility of a robot arm. However, the basic structure does already exist and can be seen in the above picture (and on our github https://github.com/CJT-Robotics/gui_small/). Each blue container represents a thread and the white arrows show data exchanges between the threads. The green arrows clarify that each thread is started from the main thread. On the left, you can see the stationary controller thread. It manages the input of the joystick with the <linux/joystick.h> library.

Currently, we are using the Logitech G Extreme 3D Pro Joystick. Though, we are discussing the possibility of switching to PS4 dualshock controllers. The Data Repository class [abbreviated DR(s)] uses the Singleton design pattern and is responsible for all the Inter-Thread Data Exchange. It contains a bunch of private variables which are accessible through get and set functions. Each Variable is protected by a mutex which safeguards the get-and-set functions. The Singleton is initialized in the main thread first. I chose this approach to avoid any thread-related issues which are sometimes very difficult to fix. The Transfer Node is a Ros Node that manages the robot-Frontend communication. Currently, both the publisher and the subscriber are contained in the same Node. However, we intend to separate them into separate nodes in the future to promote maintenance and readability.

3.2.1.1 Driver Assistance Tools



This is also under development since we need some more experience to figure out what our driver struggles with and what can be done about it. For example, our camera doesn't include the wheels. Because of the small frame we are using, we cannot change this. However, we plan on including the projected path of the wheels in the camera.

3.2.2 Backend

3.2.2.1 Free Kinematics Kit

Setting up the Free Kinematics Kit from EduArt mainly involved configuring config-files. The whole process is explained well on their github:
https://github.com/EduArt-Robotik/edu_drive_ros2/ .

3.3 Communications

For Communications we use Ros2. To extend our range of operations, our raspberry also is plugged to a nano router.

4.0 Application

4.1 Mission

For Starting our robot, we need to press the start-button and switch on the power supply for the robot. Currently, we need to ssh into our robot to launch all ros nodes and software packages. However, we plan on automising it in the future. The Frontend can be launched by launching the appropriate Ros Node (gui_small). After having plugged in the joystick, you are ready to drive.

4.2 Testing

In the Computer Science domain, we use gTest to write tests for key segments of the code, whose failure might not be noticed easily.
Furthermore, whenever we launch a new feature, we first isolate the feature and try to test for any bugs. Afterwards, we test the whole robot to ensure that the new feature doesn't interfere with other old features.
In addition, we were able to test our robot last year on the consumenta exhibition at the "RMRC Robocup German Open 2023" competition.

5.0 Our Experiences

Our mechanics team met multiple difficulties:

- It has been very difficult to build the motor mount since even our small motors are quite powerful and they need to withstand a lot of stress; especially during a competition. Through Trial and Error, we found a working configuration in the end. Moreover, we hope that after the competition, when our design has proven itself, we could open source our motor mounts to help other teams in their struggle.
- Furthermore, we are currently working on finding an optimal angle for our suspension. This has proven to be difficult. Since most robots are quite unique, there doesn't exist an universally perfect solution.

Our software team was challenged by:

- The setup of the operating systems and frameworks has been very time consuming. There do exist many tutorials on the Internet. But many are outdated and many do not cover our exact requirements. Because of the uniqueness of robots, we do not think that sharing our experiences would help. Whenever we thought it to be helpful, we shared links to helpful sources in the documentation.
- We do struggle a lot with mathematics and for example are still trying to work out a way to implement the wheel projection. This can mostly be attributed to us being students. However, we do not want to rely on others too much regarding this topic since in our opinion the path is often more rewarding than the actual fulfillment of the challenge.

6.0 Appendix

6.1 Components

Quantity	Name	Price per piece	Source
1	old usb-webcam	(not sold anymore)	recycled from old project
1	Raspberry Pi 5 (8GB)	87.60€	https://www.berrybase.de/raspberry-pi-5-8gb-ram
1	EduArt Free Kinematics Kit	(no price given)	sponsored by EduArt
1	TP-Link WLAN-Router TL-WR802N	~20€	recycled from old project
1	MakerBeam Kit	109€	part of school workshop
4	motors (2224U018S R IEH2-512 2081 22GPT 44:1)	122,50 €	recycled from old project
4	Wheels	(price cannot be traced back by us)	recycled from old project
1	Logitech G Extreme 3D Pro Joystick	~50€	borrowed from another team
1	battery pack: 16 cells	(price cannot be traced back by us)	recycled from old project
1	base metal plate		part of school workshop
1	RPLidar A1M8	140€	sponsored

All in All: 800€ that we certainly know of, but probably close to 1000€ since we do not know the cost of some of our assets.

6.2 Additional Plans

Even though we have already mentioned many of our plans in the appropriate segments, we have still missed to mention some of them:

- add a light on the front
- add ultrasonic sensors on the front
- improving our appearance by adding a casing