Bento Robotics

Team Löhbotics

Team Description Materials - Robocup 2024

Summary

We are Team Löhbotics, a subdivision of bento robotics, the winner of last year's German Open.

Our focus is improving the Bento-Box, the winning robot, whereas our counterpart, Team Nachtgeeker, is designing a new one from scratch.
This means major hardware and software changes, some of which are still in development.

The name “Löhbotics” consists of the two words “Löhe” and robotics. The word “Löhe” is short for the name of our school (Wilhelm-Löhe-Schule Nürnberg), where we work on our robots as a part of an extracurricular course.

Bento robotics formed in early 2021, as Prof. Dr. Stefan May of the OHM university stumbled across an extracurricular technology course offered by our school and decided to lend support and introduce the world of robotics.
Consequently we learned new programming languages and tools, such as Python, C++ and ROS, and began tinkering with proper robotics hardware.

We are very grateful to all the companies and organizations, especially the OHM university, that have supported us along the way and continue to do so.

General Information

Team Name: Löhbotics

Robot Name: Bento-Box *[github]*

Organization: Bento robotics (Wilhelm-Löhe-Schule)

Country: Germany

Website: https://bento-robotics.github.io/website/

Plans: github.com/orgs/Bento-Robotics/projects/1

Contact: bento.robotics@gmail.com

Contact Person: Dr. Markus Stammler (markus.stammler@loehe-schule.de)

Hardware

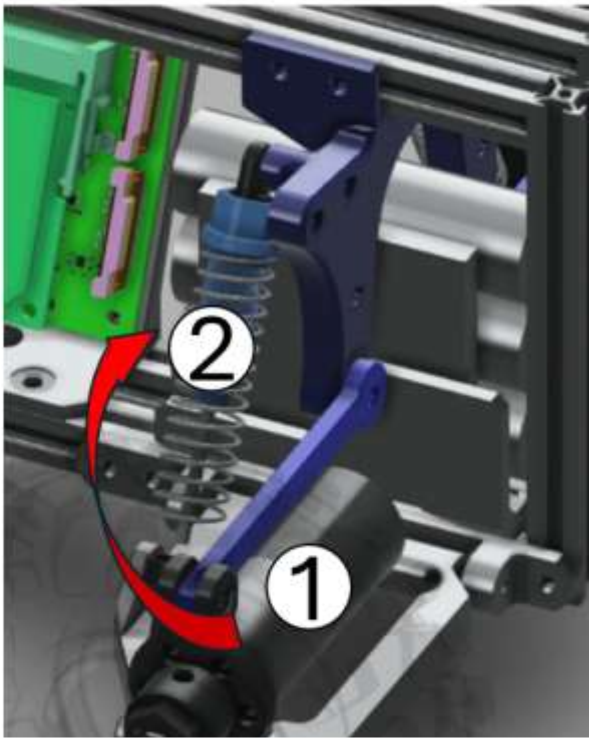
Chassis

The chassis consists of two Makerbeam rectangles made of 10 mm aluminum extrusions,
to which everything is mounted through 3D prints.

The suspension setup was inspired by RC cars,
and offers more than 60 mm of travel.

Everything together fills a space of 32 cm x 27 cm x 16 cm.

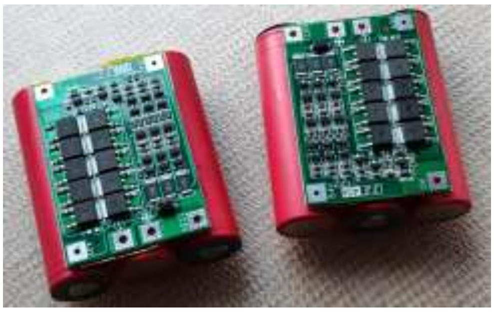
chassis, bare and with enclosure



Here you can see a render of a suspension. It has proved its reliability and also flexibility because the strength of the shock absorbers (2) is alterable. It is also quite compact because the motors (1) are mounted closely to the wheel (transparent)

Power

Power is supplied by two Li-Ion 3S1P batteries in series, each delivering a nominal 11.1 V with a 3.2 Ah capacity.



Each one incorporates three 18650 cells and a BMS, taking care of battery protection and under/overcharge.

We chose the dual battery setup to center the batteries' weight along the chassis and make more room for electronics.

Drive train

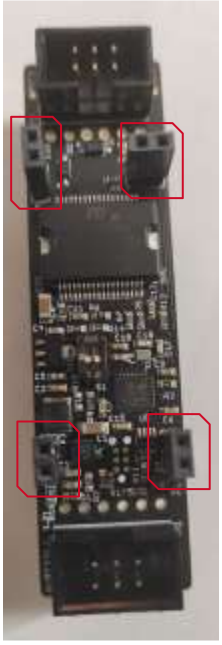
<u>Motors</u>

We use four 2224U018SR motors from Faulhaber with IEH2-512 encoders on their rear ends.

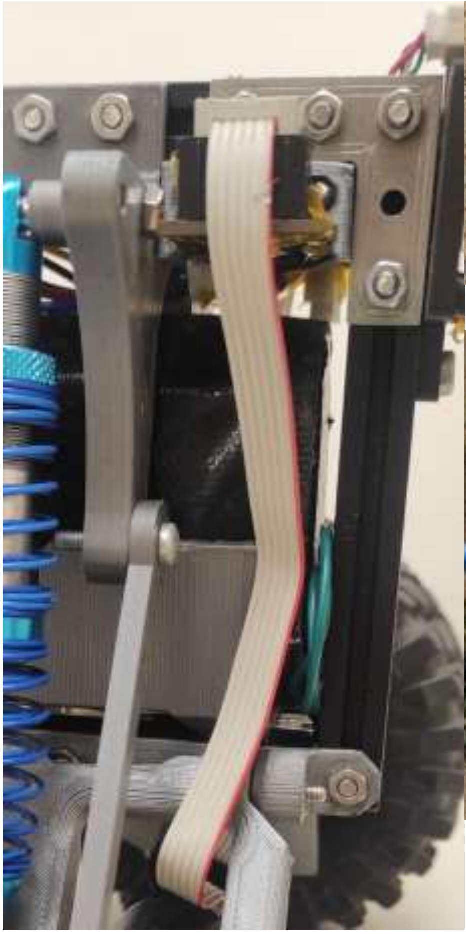
They are mounted to 22GPT 44:1 planetary gearboxes, resulting in a continuous output torque of 0.8 Nm.

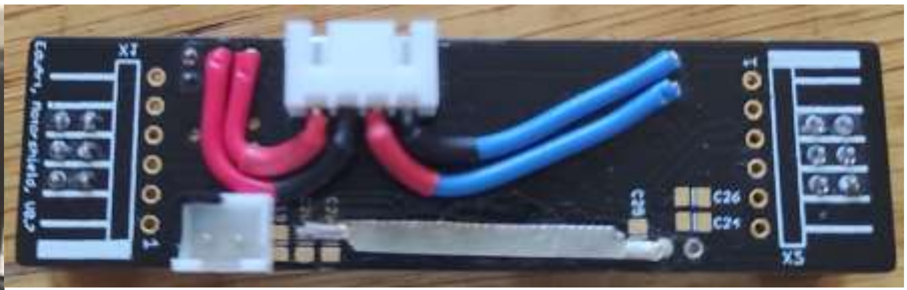
Motor controllers

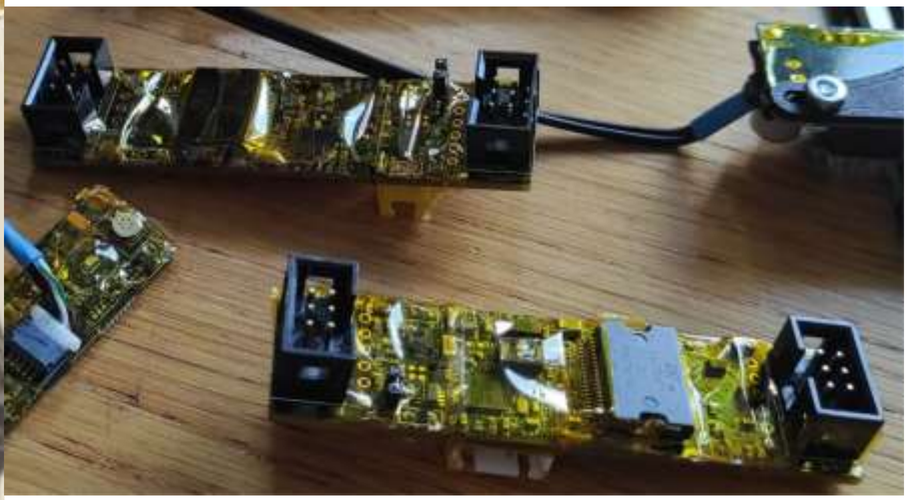
Our motors are driven by EduArt's Motorshield modules. They communicate via CAN and a 3 V enable signal.
Usually they are used stacked atop each other through dupont connectors (marked red),
This however wasn't feasible in our form factor so we soldered on some JST connectors.



EduArt Motorshield module







Taped up and Modded Motorcontrollers ↑

Cameras

Two USB cameras are placed at either end of the robot.
The front camera has an ultra-wide angle lens, giving operators a better sense of orientation.
The rear camera has a non-distorting lens, and is used for computer vision. We plan on adding an MLX90640 infrared camera for low resolution but budget friendly thermal imaging, and perhaps an intel D435i depth camera to map terrain and avoid obstacles.

Processors

Currently we are still using a jetson nano, but it has
proven to be good on paper but a hassle to use.
We would like to switch to a raspberry pi 4,
and plan to use a CM4 in our final design.

Unfortunately we didn't want to change too much so we ended up keeping the Jetson for at least the RoboCup.

Unlike a regular pi 4, the CM4 is more compact and gives us some PCIe lanes that we have already succeeded in using for dedicated WiFi modules.

We'd like to design a CM4 carrier board to integrate everything we need.

As stock linux isn't real-time, and there are better arduino libraries than linux libraries for most chips, we added a coprocessor, a RP2040 raspberry pi pico.

This makes adding hardware far easier, and by using the µROS library it is fairly easy to integrate with ROS2.

Robot Arm

We are currently still working on a robotic arm. The code is written in Python and Arduino. We intend to use the Adafruit PWM Servo Driver because of its compatibility via I2C to our main board . Trying out several other methods of controlling the four Servo Motors we figured this as the best way to do so.

Software

Bento-Box's software is based on ROS2, using a [modified EduArt motorcontroller interface](#) and free software such as the [usb_cam](#) & joy_linux ROS2 Nodes and RQT, and platformio for direct hardware interfacing.

Containers

We cross-build our '[rosbox](#)' container for amd64 and arm64 on stronger hardware, so that the robot computer needn't do any strenuous compiling tasks.
Configuration files are mounted into the container, and a Launch file is executed as the main container process.

Overall this makes for an incredibly dynamic system, where things such as automatically starting software on boot are a single line of configuration.

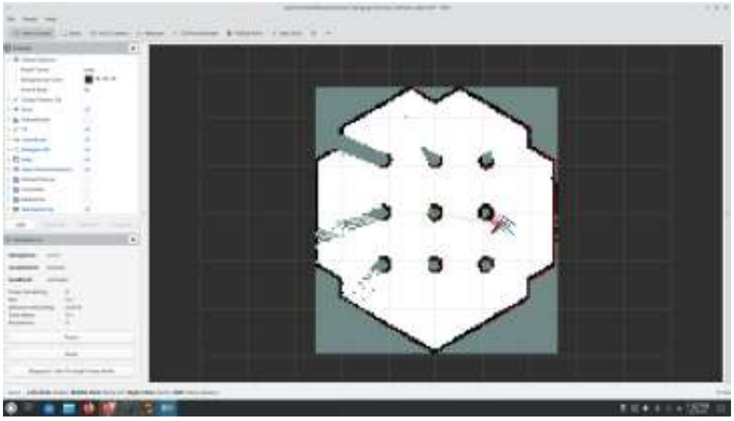
All of our robots use this setup [Zy, Bb, Sch].

Human-Robot-Interface

Our current setup uses RQT to display camera streams and status data, and a Joystick for input.

The Ultra-wide camera is for driving forwards, as you can see where the wheels are, and the non-distorting camera helps when reversing.

LiDAR mapping



example map:

We are currently on the verge of creating maps using the LiDAR scans. To achieve this we are using Nav2 and slam toolbox.

QR-code reader

To scan QR-codes we use the ros2 package zbar_ros. We also have a python script running that prevents QR-codes from being scanned more than once.

Camera Software

We use libcamera in the container to power our camera software.

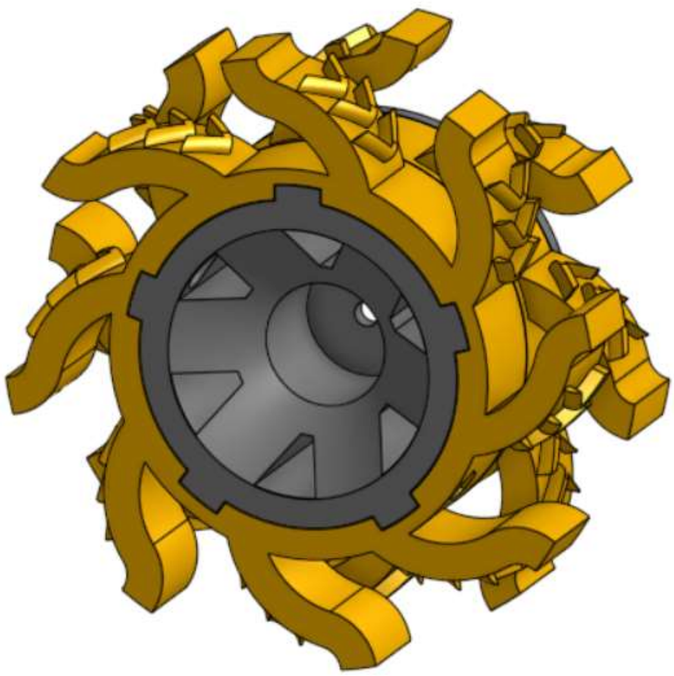
Experiences

- **Loose team:** Our first proper drive was on the same day as the competition, due to the team only rarely being able to meet and contribute. That is why we now meet 2-3 times a week to work on the new robot since late 2023.
- **Jetson Nano:** The Jetson Nano is fantastic on paper but terribly supported in practice. This handicapped us massively in software and hardware in the end, with countless hours wasted on trying to get things working.
- **Bad cameras:** During the last robocup, we simply mounted off-the-shelf webcams to a pole on top of the robot which increased our total height dramatically. These often glitched and crashed, blame to the Jetson. Now we use dedicated Camera modules with M12 lenses, with wider angles and more reliability.
- **No fallbacks:** Last robocup we stopped thinking about connectivity after getting Wi-Fi working. Wi-Fi promptly failed at the competition and we had to scramble together an alternative.
- **The Fuse:** When making the power board our tech added a fuse for the battery for good measure. As it turns out this has single-handedly saved our electronics on multiple occasions. Plugging in a charger backwards is all too easy.

Current Experiments and Testing

Wheels

3D-printed wheels for better traction: The yellow parts are made of flexible filament that is mounted onto the solid base. These are pain to print, so we are considering casting them using rubber caulk.



Movement Visualisation

By subtracting camera Frames with a static background from each other (and applying some filters to improve contrast) we can display movement.

This can simply be humans, or, if the background is suitable, even air movements.

Demo at [gist.github.com]

Gas exhaust of a bunsen

Radio

Due to previous experiences we have put more thought into the data link between Operator and Robot.
We now run a dedicated PCIe Wi-Fi module for radio contact, theoretically giving us access to technologies such as wifi 6e (6GHz), but this is against the RMRC rules, so we use 5GHz.

Adding hardware like this on the jetson never worked correctly, and the old kernel version forced us to use old modules, hence the switch to raspi 4.

To get PCIe Wi-Fi modules working, you need to recompile Pi OS' kernel to include support for them. It's easier than it sounds :P
We've documented this process over at github.com/Bento-Robotics/Schaeufele/wiki/Software.

Cable

If all goes wrong and wireless communication fails, a wired connection is a reliable fallback.
We tried many standards, but nothing beats ethernet when it comes to ease of use and cost to implement.

As an added bonus, ethernet can carry power and data simultaneously (PoE), which makes charging whilst driving and therefore extended mission durations possible.

It became apparent however, that the regular RJ45 jack isn't built with moving cables in mind - the retaining mechanism kept breaking. So we opted for a far more robust GX16 connector.

↑ GX16 10-pin connector

The male side will be mounted to the robot's chassis, and the female side attached to a long CAT5e ethernet cable.

Either a 4-wire or an 8-wire cable can be used (GX16 is available in 2 through 10-pin variants), with a 4-wire cable being lighter, but only capable of 100 Mbit/s, whereas an 8-wire cable can supply more PoE power and gigabit ethernet at the cost of more weight and stiffness.

The test cable shown didn't show any signal degradation, in spite of lacking shielding - speed tests run before and after had the same result (100 Mbit/s).

Costs

Item	Quantity	Price/unit	Notes
Batteries	2	~10.00€	reclaimed Cells
Bingfu WLAN Antenna	1 (2/pack)	12.98€	amazon
AMD RZ608 Wi-Fi	1	12€	
Raspi CM4	1	~50€	depends on specs
Adafruit board	1	-15€	adafruit.com
CM4 Heatsink	1	7€	waveshare
CM4 Carrier Board	1	?€	cost not yet known
Cameras	2	16.50€	sponsored by Ross Robotics
MakerBeam kit	1	115.00€	amazon
Shock Absorber	4	8.51€	
Motor Drivers	2	100€	sponsored by EduArt
Motors	4	209€	faulhaber.com
Wheels	4	4.35€	
Total		~ 1333 €	subject to change

Our Team

Members

- **Samuel Pelz** - *Instructor, Technician, Electrical Engineer*
- **Luis Herzog** - *Team Supervisor and Website Programmer*
- **Daniel Rudyy** - *Electronics, Mechanical maintenance*
- **Michael Meyer** - *Electronics, Co-Constructor*
- **Christian Baussmerth** - *Mechanic, Designer*
- **Dennis Gulz** - *Driver, Social Media Manager, Mechanic*
- **Till Lutz** - *Team Spokesperson and Manager*

(mentors in **red**)

All members are students from the Wilhelm-Löhe school from grade 10 up to grade 12.

Team personnel is bound to change, especially with people switching between Team Nachtgeeker and Team Löhbotics.