



Bento Robotics

Team Nachtgeeker



Team Description Materials - Robocup 2024

Summary	2
General Information.....	3
Hardware	4
Chassis.....	4
Power	5
Drive train	5
Motors	5
Motor controllers.....	5
Cameras.....	6
Processors.....	6
Robot Arm	7
Modeling.....	7
Software.....	8
Human-Robot-Interface.....	8
Current Experiments and Testing	8
Movement Visualisation.....	9
Radio.....	9
Cable	11
Experiences	12
Plans until the Robocup	13
Costs.....	14
Our Team	15
Members.....	15

Summary

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

Our focus is creating the robot Schäufole, in a quad-flipper platform with enhanced mobility, especially on stairs, whereas our counterpart, Team Löhbotics, is working to improve Bento-Box, the German Open winning robot. This means most things are still in the R&D stage.

“Schäufole” is a traditional Bavarian-Franconian dish which consists of a pork roast that still contains the shovel-like shoulder bone. We chose the name because our robot will be designed with flippers that also resemble the shape of a shovel.



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: Nachtgeeker

Robot Name: Schäufele [ʃœʏfələ] [*github*]

Organization: Bento robotics (Wilhelm-Löhe-Schule)

Country: Germany

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

Plans: github.com/orgs/Bento-Robotics/projects/3

Contact: bento.robotics@gmail.com

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

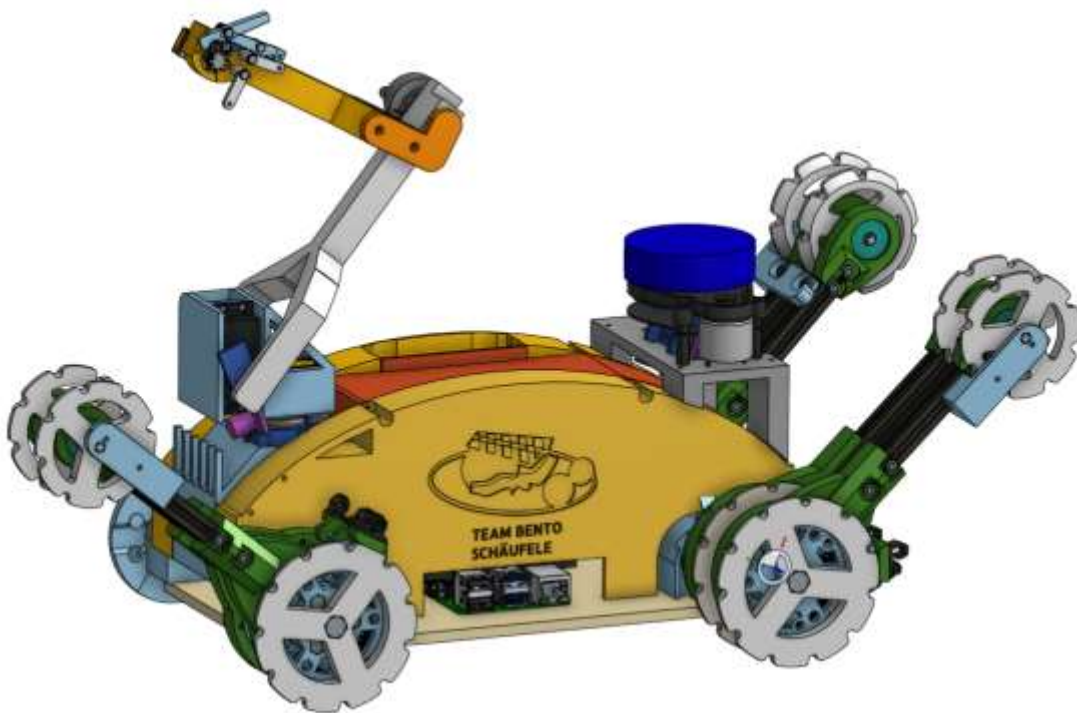
Hardware

This is our plan for the robot “Schäufele” for the upcoming RoboCup 2024.

(you can also view our OnShape via

<https://cad.onshape.com/documents/9c18033a030c6f56d8ed1210/w/dd191a8f46515620872bb010/e/40d26ac2d4b0547b41c2f2b4>)

Chassis



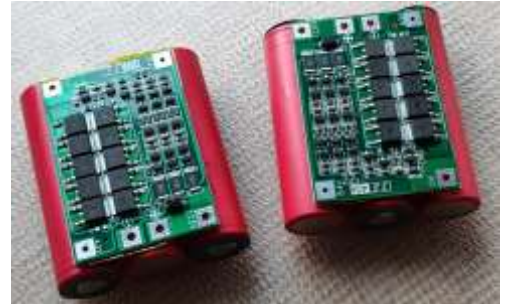
The flippers are made of 3d-printed parts for the frame and wheels and connected with Maker Beam, some screws, nuts and gears are made of metal. The flippers are then screwed onto a wooden base plate. In addition, we will print an outer shell to protect our hardware

Power

Our plan is to supply Schäumele the same way as Bento-Box, using two Li-Ion 3S1P batteries in series, each delivering a nominal 11.1V with a 3.2Ah 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

Motors

We use eight [2224U018SR](#) motors, four for tracks and four for flipper rotation, from Faulhaber with [IEH2-512](#) encoders on their rear ends.

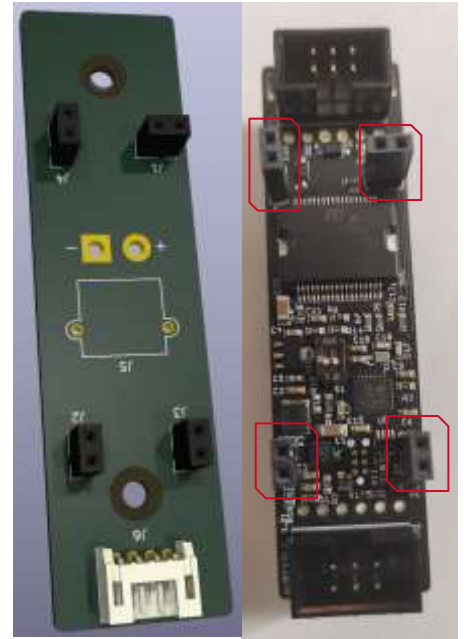


They are mounted to [22GPT](#) 44:1 planetary gearboxes, resulting in a continuous output torque of 0.8 Nm.

Motor controllers

Our motors are driven by EduArt's Motorshield modules. They communicate via CAN and a 3V enable signal.

Once again we don't plan on stacking them into a central tower, and instead plan on creating breakout boards to mount them closer to the motors. This Means we needn't modify them, they can benefit from motor cooling we plan on adding, and we won't need to extend the motor's cables.



*EduArt
Motorshield*

Cameras

Once again, we plan on using an ultra-wide lens for driving, and a non-distorting lens for Computer Vision. These could either be connected via USB or MIPI CSI.

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

Our powerhouse is a raspberry pi 5,
But we'd like to use a CM5 in future. (once it is released)

Unlike a regular pi 5, the CM5 gives us a much compactor form factor.

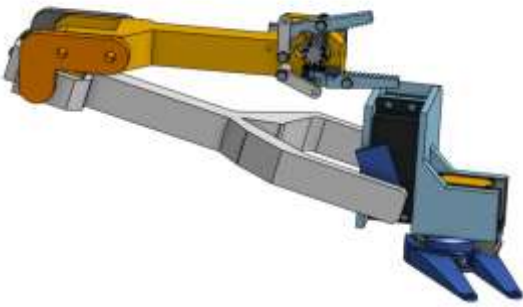


As stock linux isn't real-time, and there are better arduino libraries than linux libraries for most chips, we added a coprocessor, a RP2040 raspberry pi pico.

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

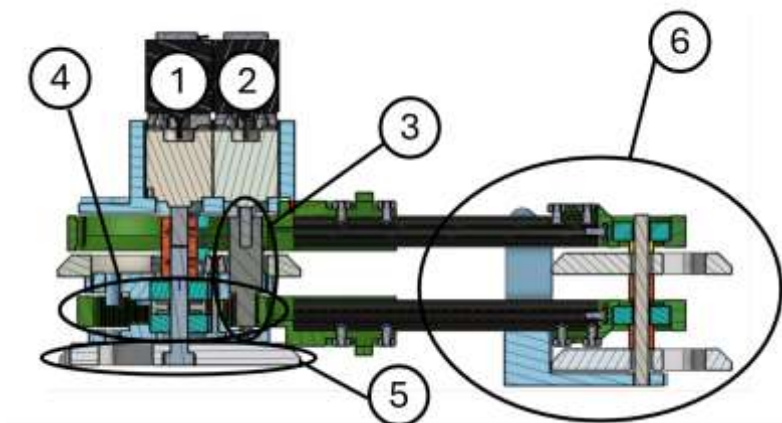
Robot Arm

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.



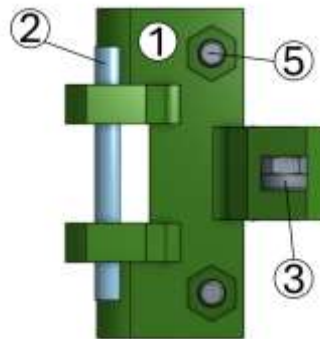
Modeling

We use Onshape to create models, which we then 3D print. Here is an overview of our robot in Onshape.



Here you can see a section of our flipper. Motor 1 is moving the wheel (5) which is moving the chain. We removed the planetary gearbox in that position because it caused too much friction. The chain elements are then mounted on the wheel 5 and the wheels on the other side of the flipper (6). Motor 2 is powering

another planetary gearbox (4) via a machined aluminum gear (3) that moves the rest of the flipper (6) with a ratio of ~ 0.15 so that there is enough torque to lift the robot using the flippers. We use standard lego axles and gears for the other 5 planetary gears.



For the chain elements, we connect a 3D-printed base with headless screws (2) and a torque prevailing nut.

The wheels are in contact with both ends of the screws. Finally, a 2-layer tread consisting of a cut, slip-proof mat and a yoga mat are glued onto the chain elements. We discarded the idea of using 3d-printed flexible filament because it did not provide enough friction.

The 4 flippers are then screwed onto a wooden base plate, the outer shell and the mountings for the batteries and the RP4 are already designed, but not printed yet.



The shell is designed to be easily removable and provide a mounting for the arm and LIDAR

Software

Our software uses ROS2 and runs in containers.

This means it can theoretically run on any hardware with little-to-no modifications. Things like network interfaces and driver shenanigans however still need to be configured on the host computer.

Schäufele uses a [modified EduArt motorcontroller interface](#) and free software such as [camera_ros](#), joy_linux 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](#)]. Schäufele however uses different software than the rest, so we added some extra

build steps as a Dockerfile. This does compile things on the robot, but they are rather lightweight so it works fine.

Human-Robot-Interface

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

We plan on making our own UI using QT or GTK, as this allows better visualization of data and an overall cleaner look.

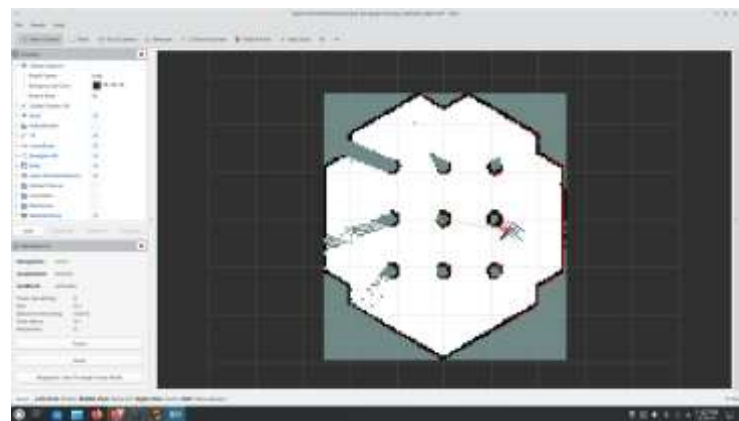
Concerning the control of the flippers, we either have to move them via the additional buttons on the joystick or we toggle the joystick into controlling them.



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.



Camera Software

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

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, as well as one that allows us to print all scanned QR-codes to the console with the push of a button.



Current Experiments and Testing



Now that the design of the robot, especially the flippers, is almost complete, and we already have a great part of our robot constructed, we are currently working on optimizing the mechanics so that the chains can move better. In addition, we are making great progress concerning software development so that we should be able to do test runs in a parcour very soon

Concerning hard- and software, we are currently working with a simple "hardware tower" consisting of the motor controllers and the Raspberry Pi so we can already do some testing.

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 human

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 5.

To get PCIe Wi-Fi modules working, you need to recompile the 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.



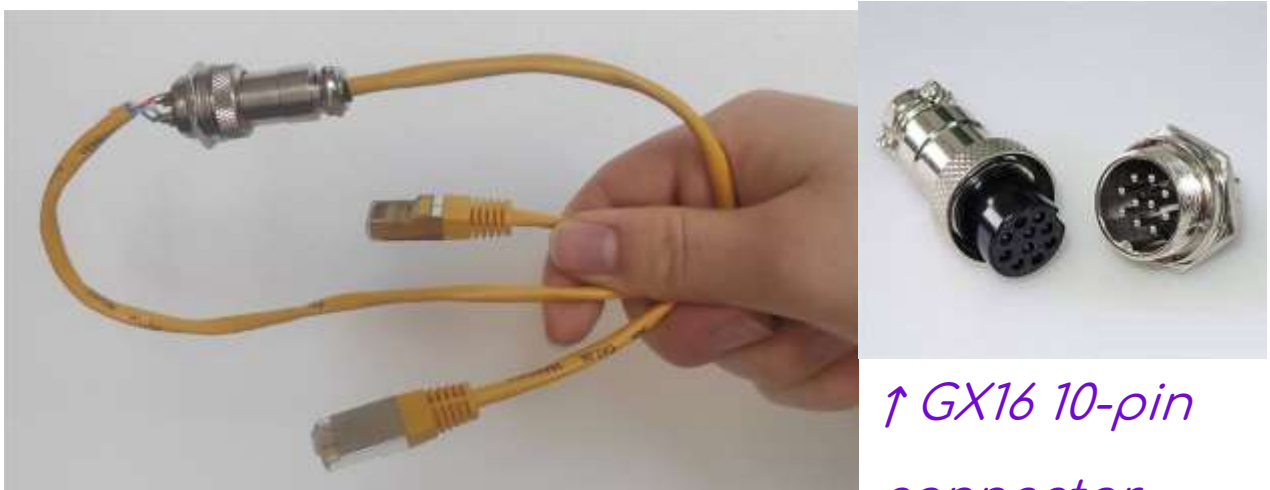
Cable

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.



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).

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.
- **Construction flaws of Bento-Box:** Although it was well suited for standard terrain, we had lots of problems with stairs, one of the main reasons we decided to use flippers on the new robot.
- **Construction Development:** Our old plan was to design the flippers using mainly lego technic parts made of metal. However, the design was quite bulky and expensive so we switched to a layout based on 3D-printed inner gears



Old vs. new: The new design is cheaper and more compact

Plans until the Robocup

Our next steps will be the finalizing of the flipper design: We need to determine whether the planetary gearbox in the wheel that is moving the chain is necessary and if we need the additional wheels to apply tension on the chain. Furthermore, we want to create a dedicated mainboard to achieve more compact and capable electronics. After that, we will create a layout for all the parts on a wooden base plate and design an enclosure. The wood will be replaced with metal in the final version.

We also plan on installing a LIDAR and a robotic arm on our robot, but both are still in an early R&D phase.

Costs

Item	Quantity	Price/unit	Notes
Batteries	2	~10.00€	reclaimed Cells
Raspberry PI 5	1	~90.0€	depends on specs
Various Bolts and nuts	-	~40.00€	
MakerBeam kit	1	106.00€	amazon.de
Motor Drivers	4	100€	sponsored by EduArt
Adafruit board	1	-15€	adafruit.com
Motors	8	209€	faulhaber.com
Total		~2288 €	subject to change

Our Team

Members

- **Christian Hopfengärtner: Supervisor, CAD, Construction**
- **Dr. Markus Stammler: Supervisor**
- Roman Deuerling: Programming and PR
- Noah Schuller: Main Builder and Constructor
- Jakob Halbig: Driver and Co-Constructor
- Sanyukt Mishra: OpenCV, Simulation
- Daniel Jaksch: Programming novice and IT
- Oskar Paul: Programming and IT

(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.