



SufniTech

SufniTech's TDM

Team members:

Róbert Krisztián Czirják, Olivér Gonda, Gábor Nándor Szücs, Ádám Vincze

Mentor:

András Tamás Bakti



Table of Content

Team information	3
Introduction	4
Who are we?	4
Meet with our team.....	5
Robot's introduction	7
Chassis	10
Drivetrain.....	10
Processing Unit	10
Power Supply	10
Communication System	11
Camera Systems	11
Robotic Arm:.....	11
Stabilizer Wheel.....	11
Future Development Plans.....	11
Pictures of the robot	13
About the software	14
Used libraries	14



Team information

Team Name: SufniTech

Team Institution: Nyíregyházi Digitális Tudásközpont

Team Country: Hungary

Team Leader: Gábor Nándor Szücs

Team Email: info@sufnitech.hu

Team Website: <https://sufnitech.hu>

Project GitHub: <https://github.com/gondaoliver/SufniBot-v3>



SufniTech in Cologne, at RoboCup German Open



Introduction

Who are we?

We are SufniTech, a Hungarian robotic team, which was founded in 2024 by two 9th grade student from Nyíregyházi SZC Széchenyi István Technikum és Kollégium, Olivér Gonda and Ádám Vincze. They both had interest and experience in robotics, so it was a no-brainer that they would join their schools robotic study group RobotTÉP.

2 months later Gábor Nándor Szücs joined to our team for our first competition (Nyíregyházi DKC Advance) where we achieved 5th place. We were not disappointed at all because we gained a lot of knowledge in those 2 weeks while we built our first machine.

At the time we were preparing for our second competition MIRK (which is a qualification competition for RoboCup) Róbert Krisztián Czirják joined to us. For that robot we used tank movement for the rotation, which failed. Our code was crashing almost every time we opened it; it was an upsetting experience for all of us.

In 2026, we participated in the RoboCup German Open for the first time. We returned with valuable experience, a clearer understanding of the challenges ahead, and a strong 4th place finish.

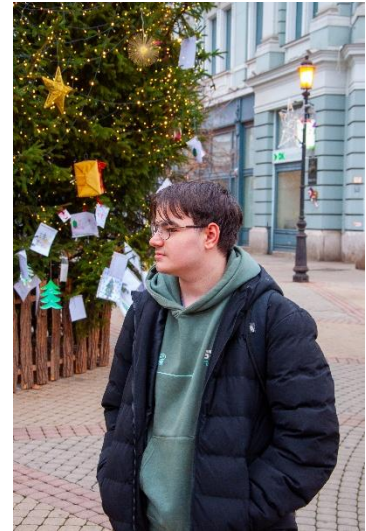
Now we are preparing for RoboCup 2026 using new components, new methods and all our experience to make a better robot we have ever made.



Meet with our team

Gábor Nándor Szücs

Gábor is the project's leader and the chief mechatronics engineer. Gábor is not a founder member of the team, but he took control at the second competitions. Gábor also the financial manager of the project. He works on the robot the most. He had a big part of designing all our previous robots and building them, but he built the latest robot entirely by himself in terms of hardware.



Olivér Gonda



Olivér is the CTO and former founder of SufniTech. His job is to make software and bring robots to life. Olivér has made the code for every single robot yet and plans to make them for a long time. He is currently using Python but planning to use other languages, so the software will be more optimized. Olivér is very ambitious about this project and always tries to stay optimistic.



Róbert Krisztián Czirják

Robert is the newest member of our team. In the robotics team, he is responsible for various mechanical tasks. He built the robotic arm and assisted with the electrical circuitry. Finally, together with **Ádám**, he is known as one of the team's jokesters, helping keep everyone smiling and maintaining a positive atmosphere within the group.



Ádám Vincze



Ádám is a founding member of the SufniTech Robotics Team, where he works as a hardware developer. His role involves designing and building the robot's frame and mechanical structure, and he personally designed and constructed the first and partaly the second robot's frame. In addition to hardware development, he also designing and developing the SufniTech website.



Robot's introduction

Our robot is the third version of our concept, which means we have eliminated numerous errors in the current design of our robot. Our robot has a plexiglass chassis to which we attach the other components. The robot can perform all competition tasks. QR code scanning and robot operation are provided by several cameras on the robot's surface, while a robotic arm is responsible for picking up items and moving smaller objects. Four 60 rpm motors provide stable drive, and a ParkSide battery, which is not only excellent in terms of performance, but also outstanding in terms of operating time. Ultimately, the entire system is operated by a Raspberry Pi 5 with 8GB of RAM, which we use to communicate via cable or, if necessary, via a Wi-Fi network.



Hardware components actively involved in the robot's operation:

Name	Quantity	Price
12V DC Motor	4	17,03 €
L298N	2	7,51€
MG996R	4	40,00€



PCA9685	1	7,72€
120mm memory foam tire	4	53,23€
ABS filament 1kg	2	31,94€
Plexiglas	4	7,98€
80mm tire	1	4,11€
Parkside Performance X20V Battery	1	39,92€
Raspberry Pi 5 8gb + heatsink with fan	1	106,46€
5A Buckconverter	1	6,11€
10A Buckconverter	1	6,42€
20A Buckconverer	1	7,00€
Iris Cam 1080p 2MP	2	21,29€
Iris Cam 720p 1MP	1	7,98€
Copper Cables	≈45	≈10,00€

Total Price of The Robot: **374,7€** (These prices are always changing due to the EUR to HUF exchange rate)



Spare parts reserved for troubleshooting problems that may arise during robot operation:

Name	Quantity	Price
20A Buckconverer	1	7,00€
Parkside Performance X20V Battery	2	79,84€
12V DC Motor	2	8,52€
PCA9685	1	7,72€
L298N	4	15,02€
Raspberry Pi 5 8gb + heatsink with fan	1	106,46€
MG996R	2	20,00€

Total Price of The Robot's Reserve Components: **244,56€** (These prices are always changing due to the EUR to HUF exchange rate)

Total Price of Robot Development (This is the whole budget of the robot): **619.26€** (These prices are always changing due to the EUR to HUF exchange rate)

The equipment prices are approximately \approx **200,00€**

The whole project is around \approx **800,00€**



Chassis: The robot's chassis is made of plexiglass, which we found to be the most suitable material as it is durable, lightweight, and transparent. Transparency is beneficial because if anything happens to the robot during operation, we can immediately see whether something occurred inside it. Our plexiglass components are 150mm×250mm and 250mm×60mm in size. However, the robot is not entirely made of plexiglass - it also contains parts printed from ABS plastic.

Drivetrain: The robot is driven by 4 DC motors operating at 12 volts. The robot's steering is not achieved through a steering mechanism but rather through "tank" movement. When turning right, the right-side wheels rotate backward while the left-side wheels rotate forward, achieving the change of direction. Since the motors are not PWM-controlled, the robot can only rotate in place around its own axis.

Processing Unit: A Raspberry Pi 5 with 8GB of RAM is responsible for the robot's complete control. The 8GB of RAM is sufficient for full robot control as well as for streaming camera footage to the control unit. The RPi does not directly control any process - each process has been assigned its own dedicated controller, so the Pi only manages these controllers, reducing the risk of short circuits.

Power Supply: As this is the 3rd version of our robot, we are using ParkSide batteries that have proven reliable in previous versions. The robot carries one 4Ah 20V battery, which can be quickly and easily swapped when needed. This unit is rechargeable, so as soon as a replacement is needed, the swapped-out unit can immediately be put on charge. Since the battery is hot-swappable, a 3D-printed holder is also required to connect the battery to the robot's circuitry. The robot has 3



major power circuits, none of which operate at 20 volts, so converters are needed to achieve the desired voltage levels.

Communication System: Our robot supports both wired UTP cable control and wireless communication over a Wi-Fi network. However, the robot is primarily intended for wired use, as the RPi delivers lower quality video signal streaming over Wi-Fi.

Camera Systems: The robot is equipped with two 1080p cameras and one 720p camera. Cameras are positioned at the front and rear of the robot, and one camera is also mounted on the robotic arm. This system handles all visual tasks - it provides the operator with a live view and is also responsible for QR code scanning.

Robotic Arm: A robotic arm is mounted on the robot's surface, enabling it to move and manipulate objects. The arm is currently unable to rotate around its own axis, though we aim to achieve this in the future. The arm consists of metal components, 3 servo motors, and a gripper mechanism, controlled by a PCA9685 servo controller.

Stabilizer Wheel: To ensure stable movement, the robot is fitted with a caster wheel that can be positioned in various orientations using a high-torque servo motor. If necessary, this system can be used to put it back on four wheels.

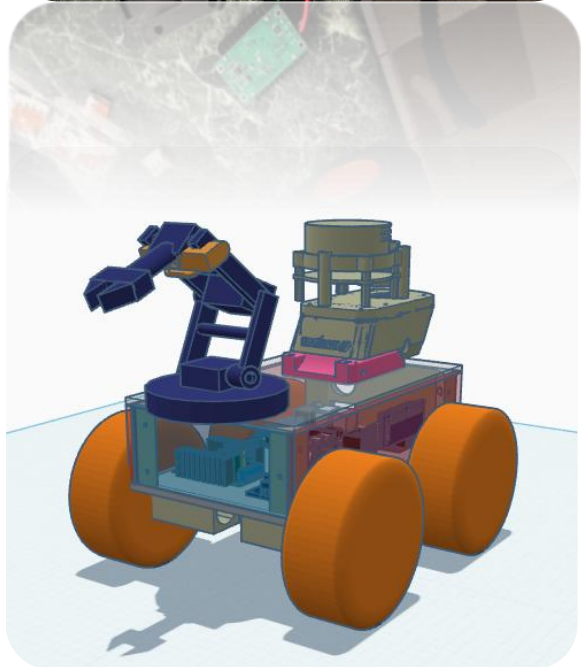
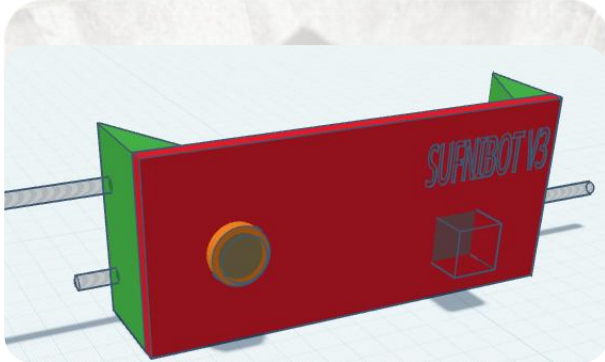
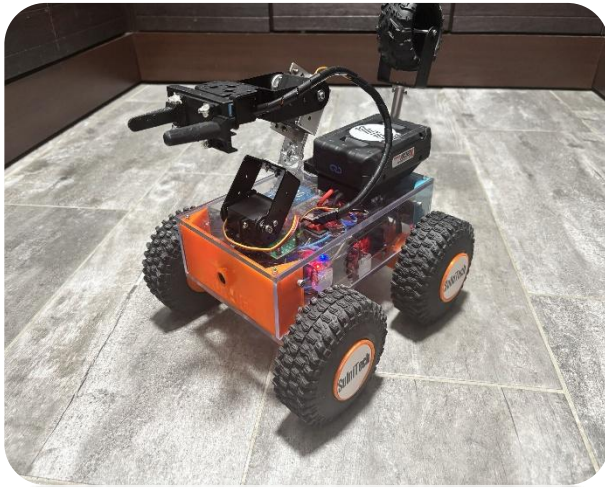
Future Development Plans: Since our robot performed well at the 2026 German Open in Cologne, where we have achieved a 4th place, we believe that no major technical changes are necessary. However, due to the robot's navigation and computational capabilities, we feel it is necessary to equip our robot with a new controller, specifically the



LattePanda IOTA microcomputer. In addition, we would like to modify the mobility of our robotic arm by connecting it to a high-performance servo motor mounted vertically in the plexiglass, which would provide more precise movement for our robot; we also aim to enhance this precision through software.



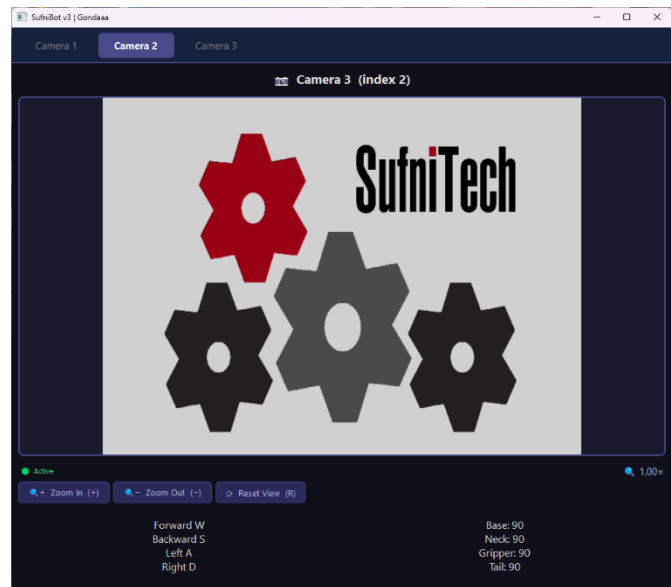
Pictures of the robot:





About the software:

Our RMRC robot's software is made fully with Python. Our previous robots were controlled by a DualSense controller, but for this robot we decided to use keyboard for more precise movements. For the first time, we made a GUI for the code, so we don't have to jump between thousands of open windows. We



also added a function to zoom into the camera picture, so some symbols like Landolt C can be recognized more easily.

We used the library pygame for every software our earlier, but it was not stable and we couldn't implement camera preview, this is why we started using PyQt 5.

In the future, we are planning to make a "robot side" and an "operator side" software, so we wouldn't have to rely on the protocol VNC, which is a bit unstable for us. We tried using SSH and X11 forwarding, but then our camera preview would have been laggy.

Used libraries:

- PyQt: We used this library to make the GUI fast and easy.
- CV2 (opencv-python): The cameras we use are managed and displayed using this library.
- Pyzbar: Used for QR code recognition and decode.



- **gpiozero**: This library is responsible for the whole movement. It communicates with our motor controller and gives signals.
- **adafruit_servokit**: To communicate with the servo motor controller we use this library.