Region: Croatia

ETC autoPrikratki

Category: RCJ Soccer LightWeight

Abstract

Our team from Croatia consists of four students: Juraj Kolarić (team captain), Franko Prikratki, Duje Handabaka and Vinka Fištrek, guided by mentor Ivan Kolarić. Juraj led the programming, implementing core logic in Arduino C++ and camera processing in Python using the OpenMV environment. Franko was in charge of the complete mechanical design using SolidWorks, while Vinka developed the electronic systems and designed custom PCBs in EasyEDA. Duje coordinated team activities, assembled mechanical components, and maintained team motivation throughout the project. In preparation for the competition, we developed two robots named Purple Push and Bubble Blast. All parts were printed using the Bambu Lab X1C 3D printer located at our school. Special attention was given to the design of omni-wheels to optimize movement. The PCBs were produced through JLCPCB. Both robots were developed entirely from scratch, representing a new generation of our team's work. You can access all our project data by scanning the GitHub QR code located to the right of our team photo. And if You want to follow our journey and see behind-the-scenes moments, scan the Instagram QR code positioned to the left of the photo.

@ETC_AUTOPRIKRATKI

EasyEDA ARDUINO

Check Our GitHub!

SOLIDWORKS

Method, Production & Design

1 CAMERA

Each of our robots is equipped with an OpenMV H7 Plus camera. This component plays a crucial role, particularly on the goalkeeper robot, where the camera is mounted facing backward. It enables the robot to continuously track the position of its own goal, enhancing spatial awareness and defensive positioning.
The attacker robot also utilizes the same camera, mounted facing forward. This forward-facing configuration assists the robot in aiming at the opponent's goal, significantly improving chance of scoring during offensive plays.

2 BALL SENSOR

MRMS IR ball finder 3 is the sensor we have been using for a few years now and it has never let us down. It is connected to two analog pins on Teensy 4.1, one to read distance from the ball and the other to read the angle of the ball relative to the robot. this sensor is reliable and easy to use and we haven't had ay difficulties reading the data it provides us. Even though we are happy with this sensor some of our plans for the future include developing our own ball sensor.

3 MAIN PCB

The main PCB, now in its 9th generation, serves as the core of our system. With each iteration, we have focused on reducing the overall size of the board and minimizing the number of wires required within the robot. This PCB houses the microcontrollers and an IMU sensor used by the robots. To ensure secure and reliable connections, we opted for JST-XH connectors.

4 UPPER CHASSIS

This year, we adopted a closed chassis design with the goal of making the structure as compact and durable as possible. All chassis components were designed using SolidWorks, made possible through a student license granted to our team.
During development, we experimented with various materials, including PLA and PLA Aero. Ultimately, we chose standard PLA due to its ease of use and sufficient durability. It proved capable of withstanding the demands of multiple competitions without structural failure, making it a practical and reliable choice for our design. This year we made barrier sensor witch is integrated into the upper chassis.

5 KICKER PCB

The robots are equipped with a dedicated kicker PCB, which is connected to the main PCB. Two of the PCB's most important components are 74LVC245 bidirectional level shifter and a MOSFET transistor. This PCB works by reciveing a 3.3V signal from Teensy to level shifter which then sends a 5V to the MOSFET which lets 12V from the battery to the solenoid. This setup enables precise control over when the kicker system is activated, ensuring accurate and timely responses during gameplay.

6 MOTOR AND MOTOR DRIVER

Each of our robots is equipped with four Joinmax JMP-BE-3561 motors, selected for their optimal balance of power and speed, perfectly suited to meet our robots' performance requirements. Each motor is controlled by an ML-R Motor Driver 1x50A, which includes reverse voltage protection, ensuring safe and reliable operation without the risk of damage from reverse voltage.

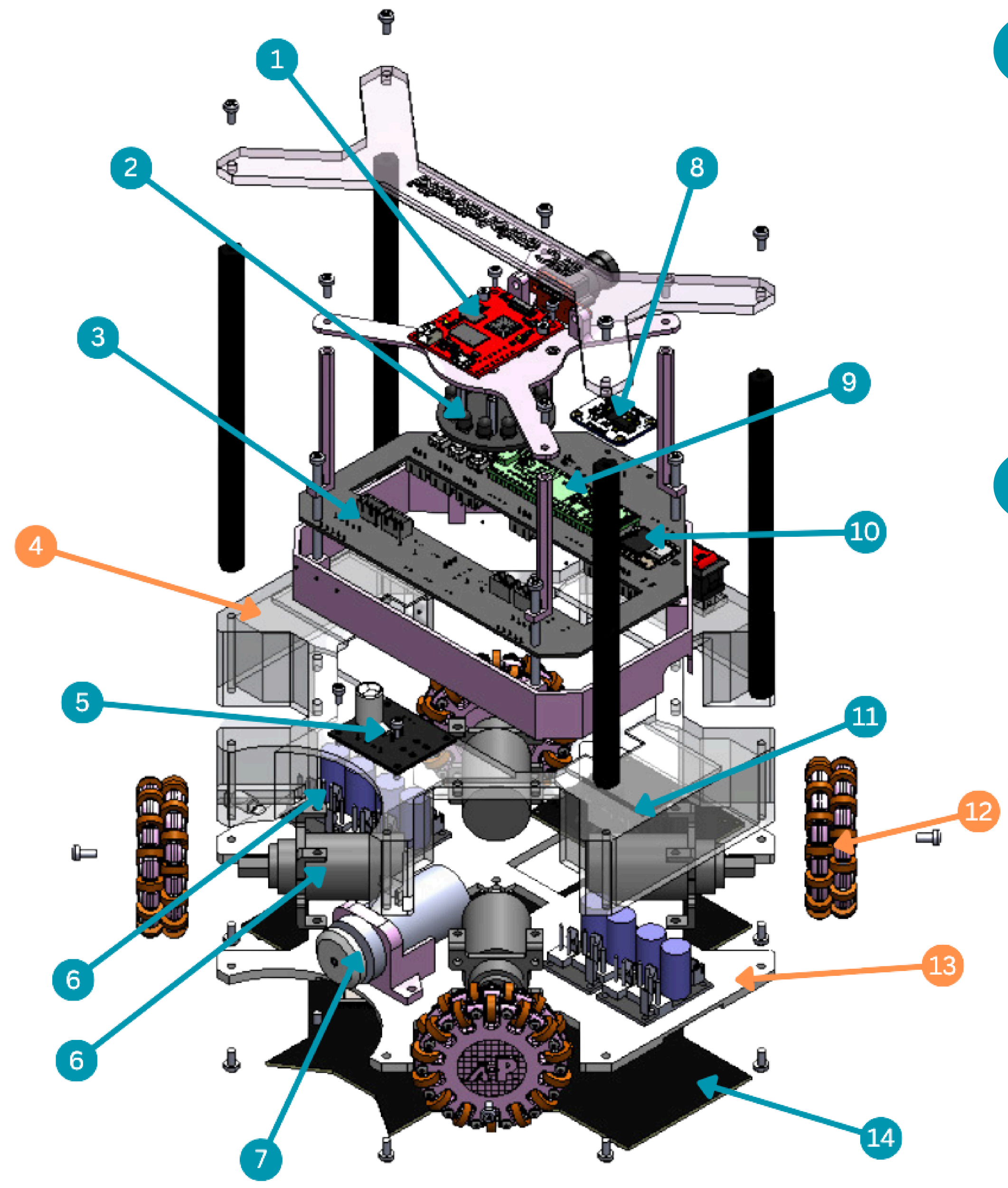


Image 1. Exploded view of our robot "Bubble Blast"

7 KICKER

The kicker system of our robot consists of two main components. The first is the ZMF-2551 solenoid, mounted on the lower part of the chassis and oriented toward the ball-capturing zone. This particular solenoid was selected for its power to size ratio, allowing it to fit within the limited space of our chassis while still providing sufficient force to deliver a powerful kick.

8 COMPASS

For reteining a stable angle, we selected the BNO055. We use this sensor to ensure that the robot is heading straight while moving or to rotate the robot in a specific direction, which is especially important in the goalkeeper's program. Initially, we experienced sensor stability issues with this sensor, particularly during collisions with other robots, which often led to BNO055 crashes. To address this, we made several hardware improvements to the PCB on which the sensor is mounted. These included adding capacitors and pull-up resistors to the I2C lines and widening the PCB traces to improve signal integrity. These changes significantly enhanced the sensor's reliability.

9 TEENSY 4.1

As the main microcontroller we selected the Teensy 4.1, the same microcontroller we used last year. It offers a sufficient number of pins to meet the requirements of our robot and provides excellent performance, capable of handling the calculations required by our software. Additionally, the Teensy 4.1 features onboard EEPROM memory, which is ideal for storing important values such as calibrated line thresholds.

10 XIAO ESP32 C6

One of the key innovations we introduced to our robots is the integration of the XIAO ESP32 microcontroller, which we used to enable communication between the two robots. This microcontroller is mounted on the main PCB and communicates with the main microcontroller (Teensy 4.1) via the UART protocol. Communication between the two robots is established using the ESP-NOW protocol. We were able to utalize this by developing verious tactics between two robots to enhance the overall team performance.

11 POWER SUPPLY PCB

Each robot includes a Power Supply PCB, which connects to a TATTU 2300mAh 11.1V 75C 3S1P LiPo battery which we found to be great for our robots neads. This board is linked to two step-down (buck) voltage regulators that provide a stable 3.3V and 5V power supply to the other components.

12 WHEELS

Over the years, we have tested various types of wheels, and we now believe we have developed an ideal wheel design for our robots which we can see in image 2. Some of the challenges we faced included instability caused by large spacing between rubber rings, as well as grass getting stuck in the wheels during matches. This season, our wheel design performed exceptionally well thanks to the decision to use dual wheels. The dual-wheel setup significantly improved both the robot's stability and overall movement. Additionally, since we built a new and lighter chassis, we were able to support slightly heavier wheels without compromising performance. To achieve this, we designed and ordered smaller polyurethane rubber rings from a local supplier, specifically tailored to our needs.

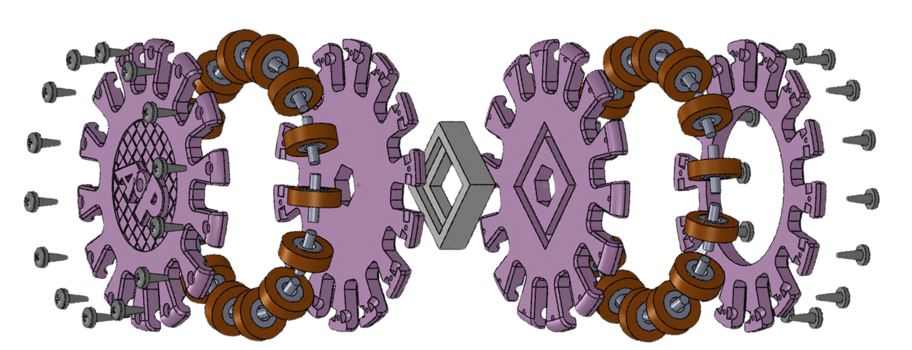


Image 2. Exploded view of our wheels

13 BOTTOM CHASSIS

Bottom part of our robots chassis is where a lot of different components and PCBs are located. Four motors and their motor drivers along with the kicker and power-suply PCB are all located on the same side of this part of the chassis and the line PCB is mounted on the opposite side of this chassis part. All the components are mounted on the chassis either using metal M3 or plastic M2 screws.

14 LINE PCB

The PCB we are most proud of is the Line PCB, which we designed to position the sensors in a Y-shaped layout. It contains 48 ALS-PT19-315C/L177/TR8 reflectance sensors, connected via three multiplexers to the main PCB. To improve sensor accuracy, each reflectance sensor is paired with an LED that ensures adequate illumination, allowing the robots to operate reliably in varying lighting conditions.

Electrical Design

Our robot's electrical system operates on three voltage levels: 12V, 5V, and 3.3V. It supports four communication and connection protocols: analog/digital signals, I2C, UART, and PWM. A 12V battery supplies power to the power supply PCB, kicker system and the main motor system. The power supply PCB also includes two step-down voltage regulators that convert voltage from 12V to 5V and 3.3V, supplying power to all other electronic components. On the Main PCB, 5V is used to power the Teensy 4.1 and XIAO ESP32 microcontrollers, while 3.3V powers the BNO055 IMU. Teensy communicates with ESP32 via UART. Both the BNO055 and SparkFun OTOS are connected to the Teensy through the I2C bus. Additionally, to the Main PCB a ball sensor and a barrier sensor are connected, both powered at 3.3V and connected to the analog pins of the Teensy. A camera is also connected to the Main PCB; it operates on 5V and communicates with the Teensy via UART.
All components on the line PCB operate at 3.3V. This PCB houses 48 reflectance sensors, which are connected to three analog multiplexers that send signals to Teensy. Each reflectance sensor is paired with an SMD LED to enhance accuracy and ensure reliable performance under varying lighting conditions.

To activate the kicker system, a digital signal is sent from the Main PCB to the Kicker PCB. On the kicker PCB there is a level shifter that shifts 3.3V signal to 5V to activate the transistor on the PCB. Upon receiving the signal, the transistor enables the flow of 12V from the battery to the solenoid motor. The main motor system consists of four JoinMax JMP-BE-3561 motors, each controlled by its own MRMS ML-R motor driver. These drivers are connected to the Main PCB via PWM and the motors are powered by 12V from the battery.

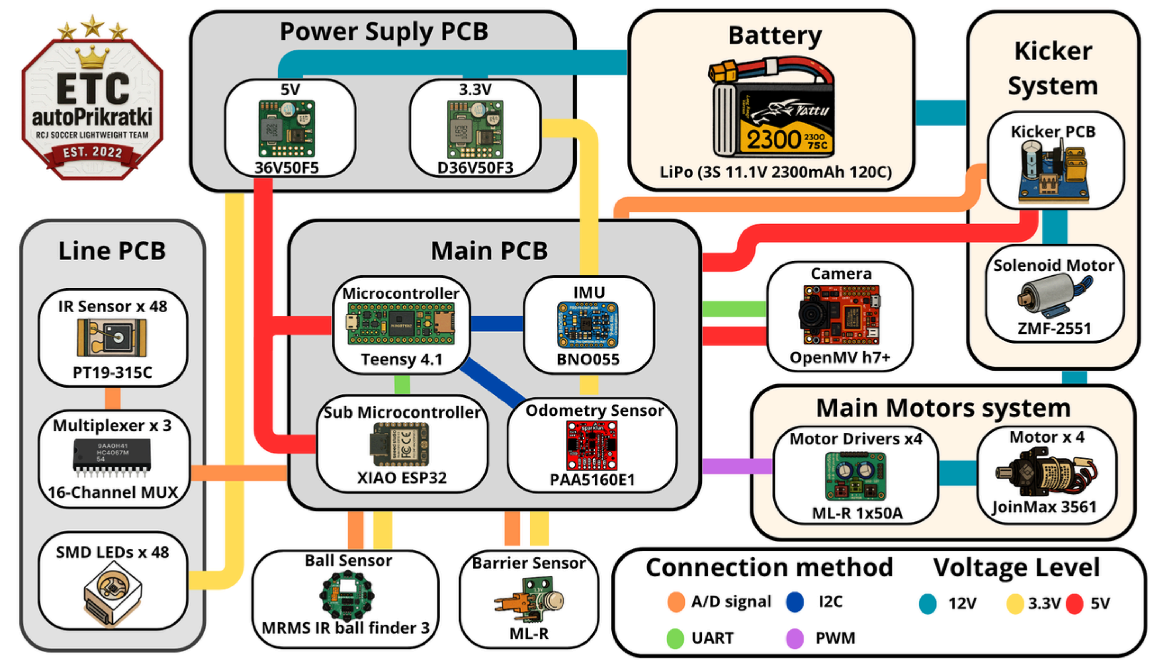


Diagram 1. Electrical design of our robots

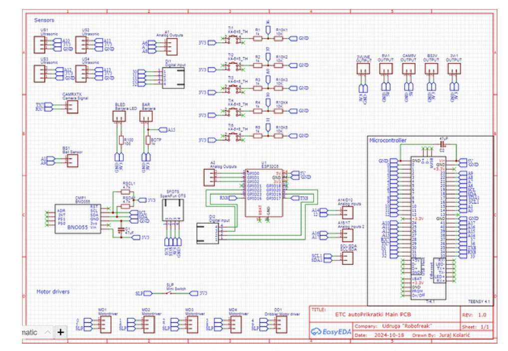


Image 3. Main PCB schematic in EasyEda

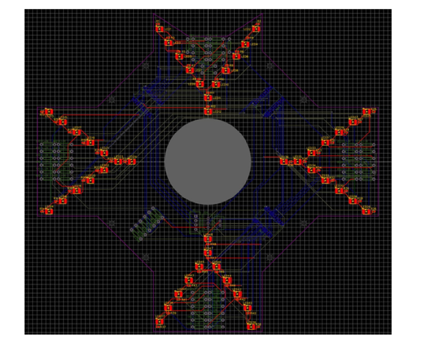
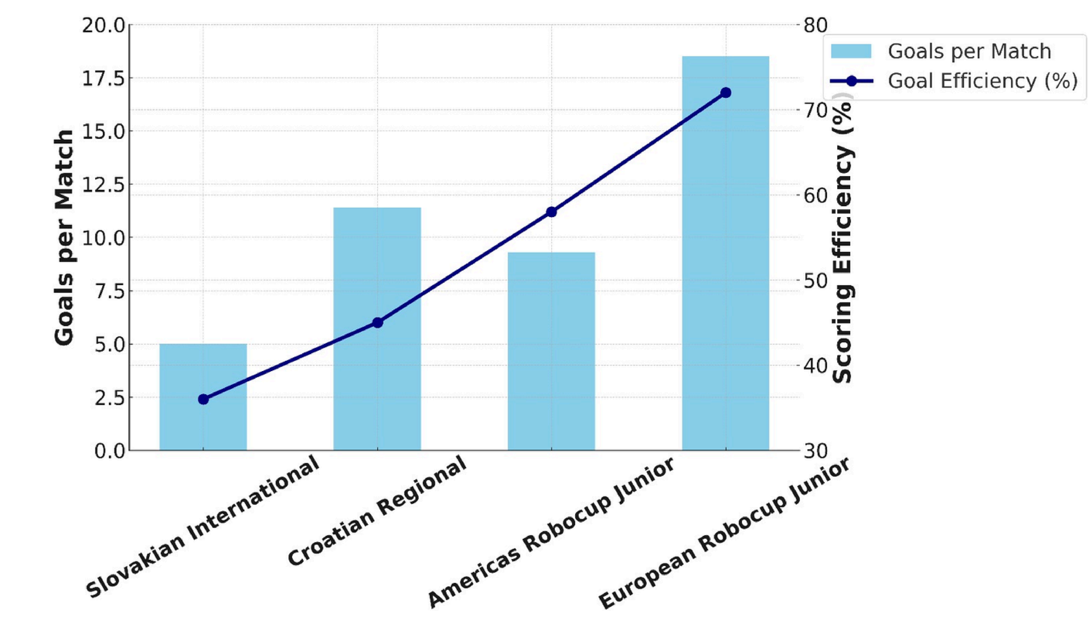


Image 4. Line PCB in EasyEda

Data, Results & Discussion



Graph 1. Goals per match and scoring efficiency by competition

Goal Scoring Efficiency Analysis

To improve our robots' goal efficiency, we analyzed performance across four competitions throughout this season. At the Slovakian International, we competed without any targeting system, which resulted in matches with very few goals and an efficiency of only 35.71%. For the National Championship, we implemented basic odometry-based targeting, slightly increasing efficiency to 43.59%, but we were still far from satisfied. Ahead of the Americas Superregional, we developed a simpler solution by introducing a "light overshooting" method in our ball-following function, which allowed the robot to approach the ball from the side. Combined with our octagonal robot shape, this caused the ball to be pushed into the goal using the front edges, improving our efficiency to 57.14%. After further refinements, the robot reliably scored from either front corner in 90% of attempts, boosting efficiency to 72.47% and securing second place at the European Championship in Italy. For the World Championship in Brazil, we went a step further by integrating a camera-based targeting system that enables near-perfect autonomous scoring from almost any position, and we estimate this could increase our efficiency by another 10% or more.

Gravity-Based Line Avoidance Algorithm

To avoid going out of bounds, our robots utilize the SparkFun Optical Tracking Odometry Sensor. Upon detecting a line, the robot looks at its current coordinates and calculates the appropriate movement angle using a mathematical formula. This angle directs the robot toward the center of the field, which is calibrated to the coordinate origin (0,0). By consistently redirecting itself toward the center of the field when a line is detected, the robot minimizes the risk of unintentionally crossing the line.

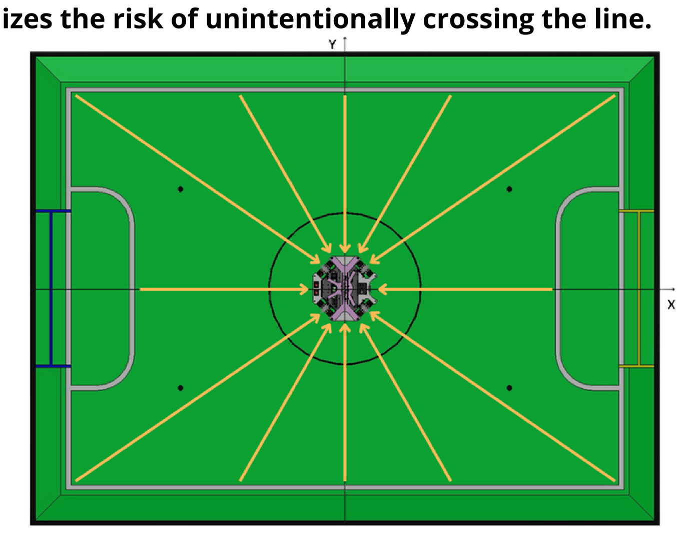


Illustration 1. Robot position and vector of movment

Angle = arctan2(xtarget − xcurrent, ytarget − ycurrent) · π/180

Example 1.

- x_current = 5
- y_current = 5
- x_target = 0
- y_target = 0

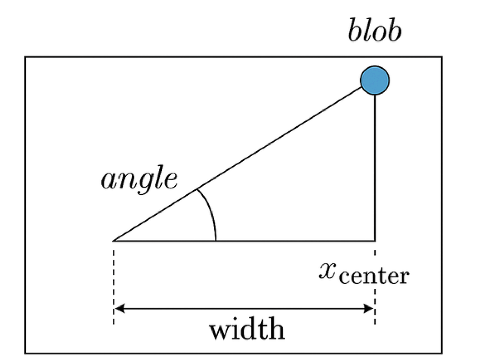
Angle = arctan2(0 - 5, 0 - 5) * 180 / PI

Angle = arctan2(-5, -5) * 180 / PI = -135°

Angle conversion: radians to degrees

Camera Vision Algorithm

To determine the angle at which the robot should rotate to face the goal, we use the horizontal position of the detected goal (or blob) in the camera image. The camera has a known horizontal field of view (FOV), and the image has a known width in pixels.
We first measure the difference between the x-coordinate of the detected blob (x_blob) and the center of the image (x_center). This pixel difference tells us how far the blob is offset from the camera's forward direction.
We then multiply this difference by the number of degrees per pixel. This gives us the angular offset of the goal relative to the robot's heading, allowing us to align the robot toward the goal with precision.



$$angle = (x_{blob} - x_{center}) \frac{FOV}{width}$$

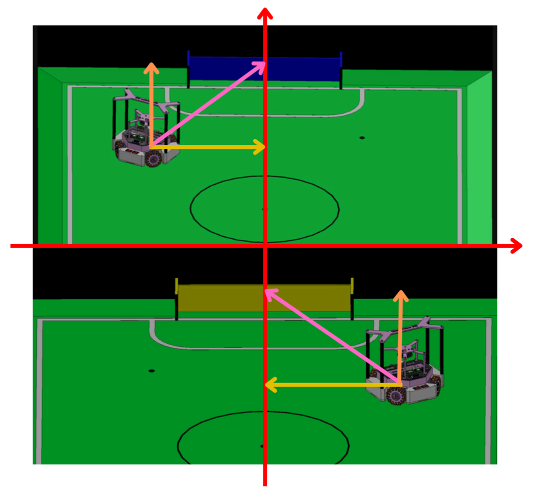


Illustration 2. Robot targeting system

Illustration 2 shows how the robot calculates the angle toward the opponent's goal using the horizontal position of the goal (blob). It represents how far the goal is from the center of the image in degrees, allowing the robot to rotate and face the goal accurately.

Made in CROATIA

Our sponsors: AC PRIKRATKI foi Udruga "Robofreak"

We're a friendly team from Croatia — feel free to reach out to us at *ETCautoPrikratkiCROTeam@gmail.com*