

RoboCup 2019 Sydney - Australia  
Team Description Paper (TDP)

RCJ Soccer Open

# Epsilon

Russia

[https://www.instagram.com/epsilon\\_team/](https://www.instagram.com/epsilon_team/)

**Participants Name:**

Chuprikov Sergey, slon4444@yandex.ru

Goncharuk Nikolay, goncharuknikolay@gmail.com

Romantsov Aleksei, romantsov@ardumakers.ru

Dubinin Gleb, dubining10@gmail.com

**Mentor:**

Mustafin Sergey, muss@bk.ru

Ogarko Anna, anyaogarko@gmail.com

**Institution:**

Center for Teacher Excellence

## Structure

<b>Introduction</b>	3
<b>1. Locomotion System:</b>	5
1.1. Mechanics of locomotion system:	5
1.1.1. Body previous version	5
1.1.2. Motors and wheels	7
1.2. Circuits	8
1.3. Kicker	8
1.4. Dribbler	9
1.5. Power	10
1.6 Omnivision	10
<b>2. Ball Detection</b>	14
2.1 Algorithms for processing data from the camera	15
2.2 Calculating the angle between the front of the robot and the object	15
2.3 Basic ball handling algorithm	16
<b>3. Goal and Direction Detection</b>	17
<b>4. Out Line Detection</b>	19
<b>6. Board manufacturing process</b>	20
<b>7. List of Components</b>	21
<b>8. Strategy</b>	22
<b>9. Gratitude</b>	22
<b>10. Final robot</b>	23

## Introduction

The Epsilon team has participated in the Robocup competition for the third year. The team members developed robot designs, their software and algorithms, using past participation experience, as well as materials provided by other Robocup participants from different countries.

In this document we will explain the features of our technical and software solutions, as well as describe the future development plans.

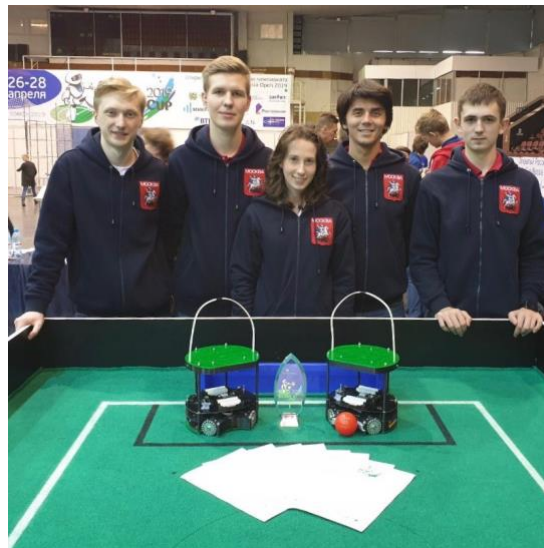


For three years, the team three times participated in the national stage of Robocup, each time improving their robots, their software and the result of participation. So, in 2017 she took the third place. And 2018 is the second, due to which it got a chance to represent Russia in the Pacific Championship in Iran in December 2018. There, the guys were able to acquire new knowledge, experience, and friends.

In April 2019, the team successfully performed at the Robocup Russia Open and went to the international final, taking first place.

This result was obtained due to the tremendous efforts of the participants and their mentors. So, Chuprikov Sergey was engaged in prototyping robots, developing their technical component. Romantsov Alexey was responsible for modeling and

creating boards. Dubinin Gleb and Goncharuk Nikolay worked on the algorithms of the behavior of robots on the field and their software implementation.



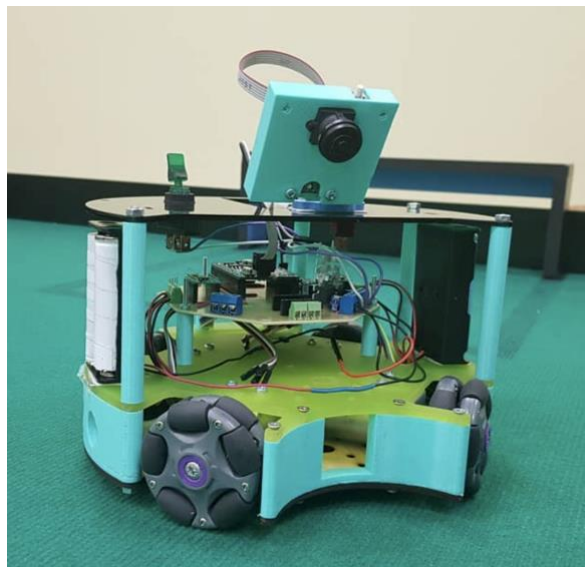
This victory gave us the opportunity to perform at the international finals in Australia this year. This will be our second international competition.

## 1. Locomotion System:

### 1.1. Mechanics of locomotion system:

#### 1.1.1. Body previous version

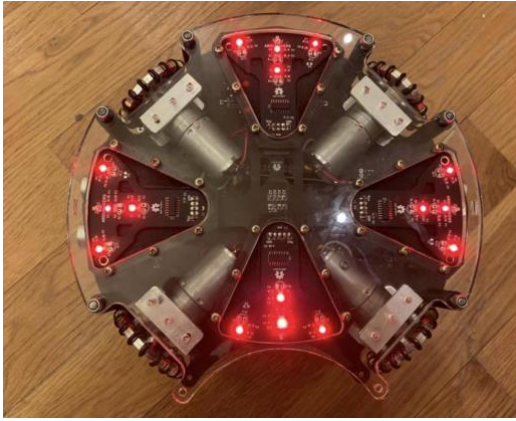
In the previous version, our robot consisted of three layers made of Plexiglas. The motherboard was located in the center, a large number of connections were made by wires. We also used wide wheels, which took a lot of area. There was not omnivision system. In the present model, all the flaws have been corrected. We use Autodesk Fusion 360 to design the details of the robot. We use Corel to work with 2D drawings and prepare them for laser cutting. For the production of parts used laser CNC machine GCC 100 W, and 3D PrintBox printer.



#### 1.1.2. Body new version

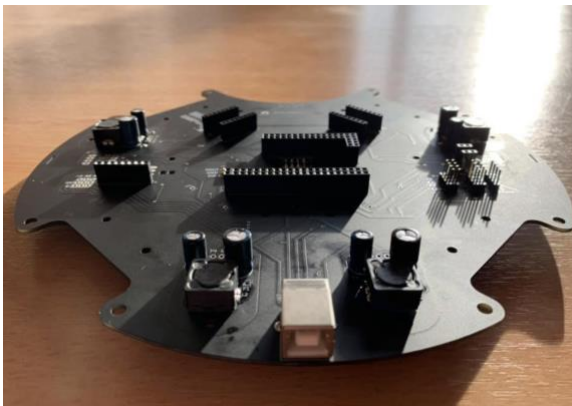
At the moment, our robots also consist of several layers, but each layer is a board and is made of textolite.

1 layer:



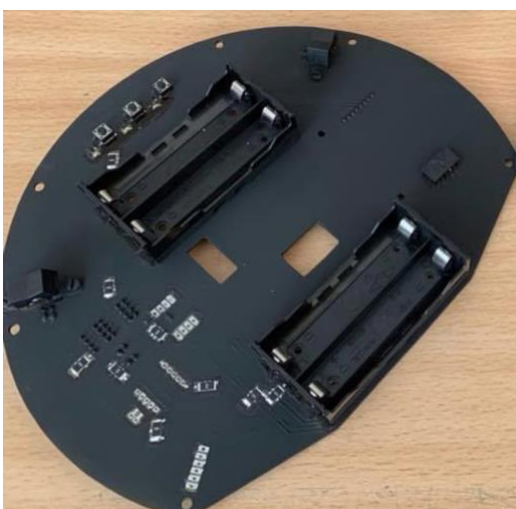
1. 4 line sensors, each consisting of 4 LEDs and 4 photoresistors
2. 12v solenoid for strike

2 слой:



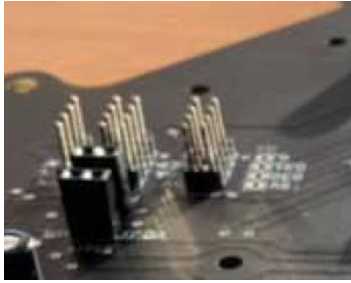
1. block for arduino mega 2560
2. 4 DC-DC converters, of which 2 by 5v for powering the Arduino and sensors and 2 by 12v for powering the engines
3. motor drivers (polulu 2301)
4. USB connector

3 слой:



1. 3 buttons
2. 3 smd LEDs
3. 2 battery compartments 18650
4. pixy camera mount 2
5. gyro (mpu 6250)
6. laser range finder (vl53lox)
7. two power switches

Boards are connected via inter-contact contacts.



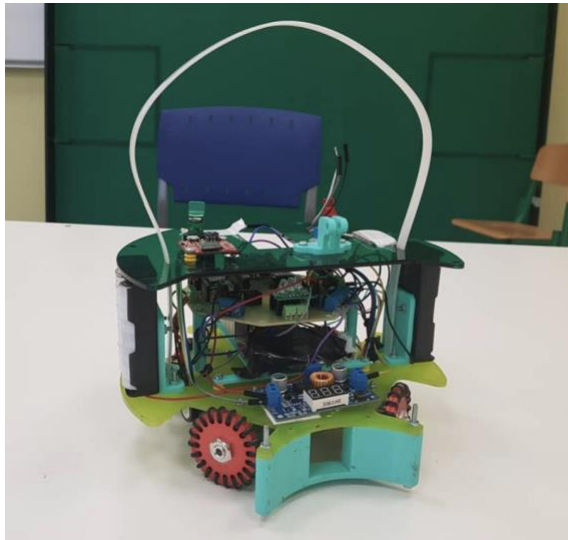
### 1.1.2. Motors and wheels

Our robots use 4 motors for movement. So this design is very maneuverable and stable. We use Polulu 20.4: 1 Metal Gearmotor 25Dx50L mm HP 12V as motors. This gearmotor consists of a high-power, 12 V brushed DC motor combined with a 20.4: 1 metal spur gearbox. Since in the Open category the maximum allowed voltage is 15V, these motors are suitable for us. The gearmotor is cylindrical, with a diameter just under 25 mm, and the D-shaped. 500 RPM and 300 mA (max) free-run, 85 oz-in (6 kg-cm).

For quadrobots, omni wheels are needed. Previously, we used the 58 mm Plastic Omni wheel.



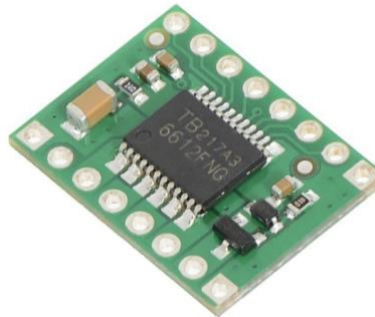
Due to participation in the Pacific Championship, we discovered where we can buy these wheels at a low price. These wheels have less weight than previous ones, and also take up much less space, which allows you to place more components on the bottom layer.



The first test wheels.

## 1.2. Circuits

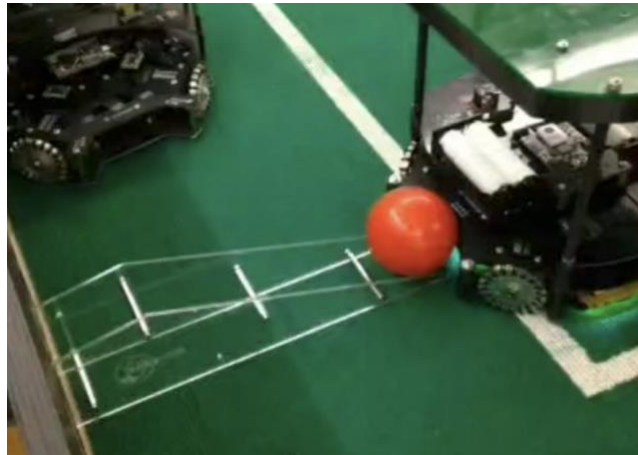
We used TB6612FNG Dual Motor Driver Carrier as motor drivers. This is a bipolar stepper motor. It is recommended to use the motor for low-power motors.



## 1.3. Kicker

To hit the ball, we use a solenoid. To power it, we use a DC-DC converter. We are raising them from 14.8 volts to 100-150 volts. Also in this circuit is a 200 volt capacitor. We charge it and when we need to hit a solenoid, we close the circuit with a 200 volt relay. And then the capacitor is allowed on the solenoid, thereby activating the anchor of the solenoid. The resistance of our solenoid is 30 ohms.





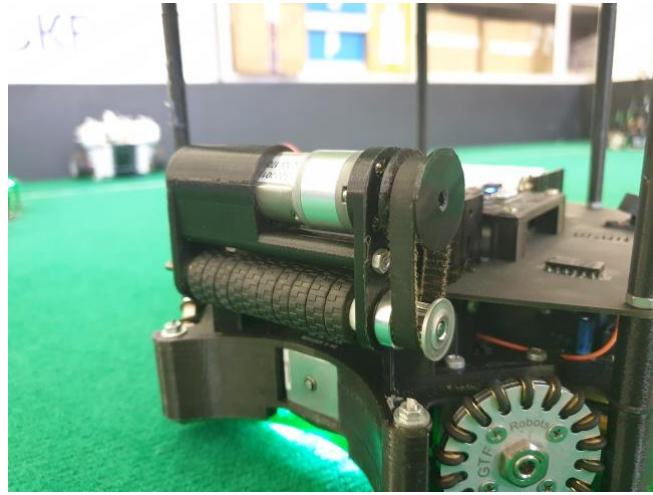
#### 1.4. Dribbler

In the first iteration of dribbling, two brushless motors from a quadcopter were used. Of the minuses, they are too fast and also too weak. Of the benefits, the design of the dribbler was very simple. Also for this dribbler we decided to cast the dribbling shaft out of silicone. It turned out very smooth, but with too much friction coefficient. Therefore, in the second iteration, we will use ready-made rubber wheels with a rather small coefficient of friction.



In the second iteration, we decided to use instead of two brushless motors, one collector motor. To begin, we had to decide on the speed and torque of the motor. Experimentally, we decided that the dribble shaft should spin at a speed of 1500 rev / sec. We picked up the motor with a speed of 780 rev / sec. And they decided that we needed an overdrive. Transmission, we decided to do the belt. Since they did not want to use big gears.

As a result, from the second iteration we got a dribbler, which completely suits us in all respects.



Before we put dribbling on our robots, we determined the presence of a ball in a niche using a camera. But after dribbling was set on the robot. He began to block the ball and the robot stopped seeing the ball in the robot niche with the camera. Because of this, we had to put a distance sensor to determine the ball in the niche.

### 1.5. Power

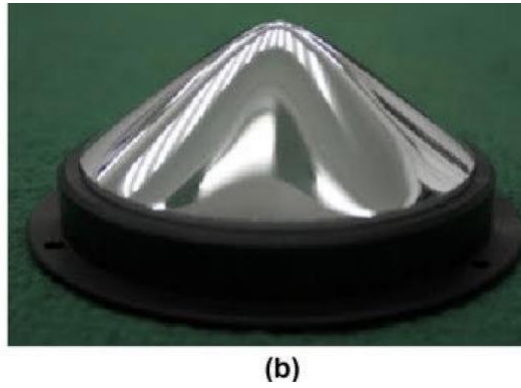
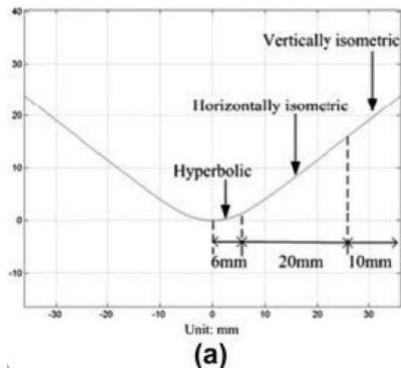
We use four 18650 batteries to power the robot. We chose this type of batteries because of their compactness and mobility. As for balancing the robot, we placed two holders for two batteries in different edges of the robot. In total, four batteries connected in series give us 14.8 volts.

Also on the robot we have three DC-DC converter. First, it lowers the voltage from batteries to 12 volts for motors. Second, it lowers the voltage from the batteries to 5 volts to power the arduino and all peripheral devices. There is also a third converter, it increases the voltage from the batteries to 100 - 150 volts. Such a large voltage is used for the solenoid.

### 1.6 Omnivision

Since last year we used only a camera whose angle of view was limited. This year it was decided to make a spherical mirror.

First of all, it was necessary to calculate the form. To do this, we turned to the experience of other teams. The following chart was found.



The second important task was to search for reflective material. We tried several options.

The first option, perfect for testing the form. He was to apply a mirror membrane to the surface. Since it was difficult to cover the entire sphere with a membrane with a uniform layer, we applied only the sector.

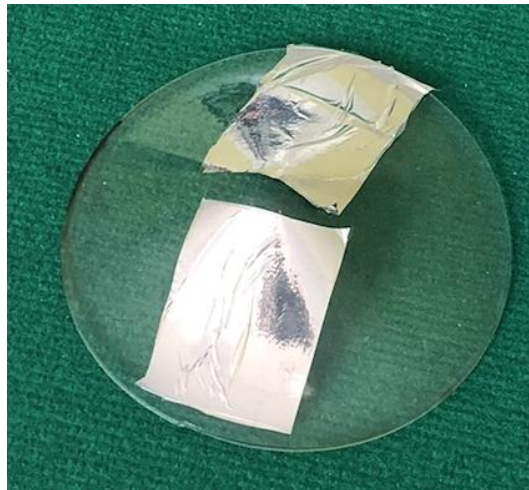


The first experience. The cone is made of paper and covered with a mirror film on top.

Then we turned to chemical metallization. But for this, a perfectly smooth surface was needed, and our blanks printed on a 3D printer did not fit. We tried to grind them, but that was not enough. Therefore, we tried to squeeze out the form of plexiglass.



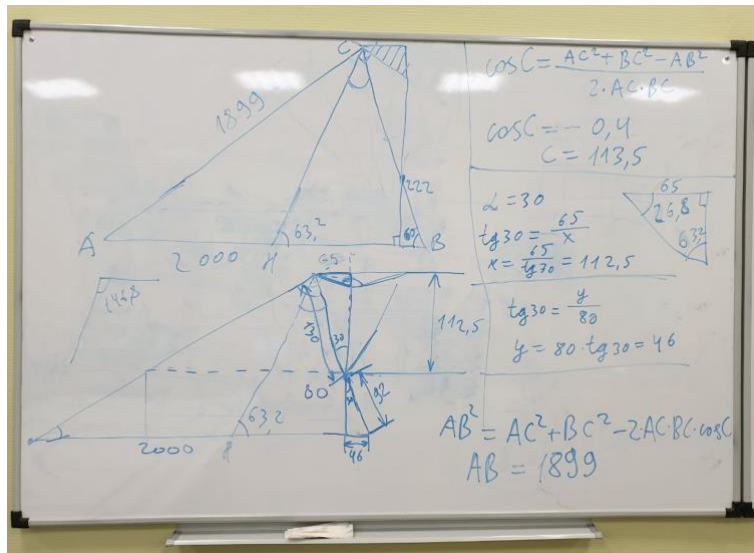
We are faced with the problem of uneven stretching. As a result, the image distortion was unacceptable.



We also tried to use ready-made mirrors for the side mirrors of cars. But their angle was not enough.



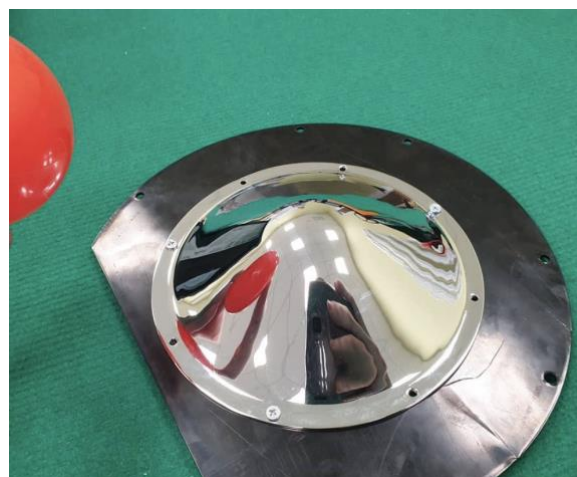
The mirror formula we found did not suit us. The image distortion was too large. And we tried to calculate the angle for a mirror with a large radius at the base.



One of the ideas was to make a mirror of Duralumin 16t. This alloy has a high hardness that allows it to grind to a mirror shine.

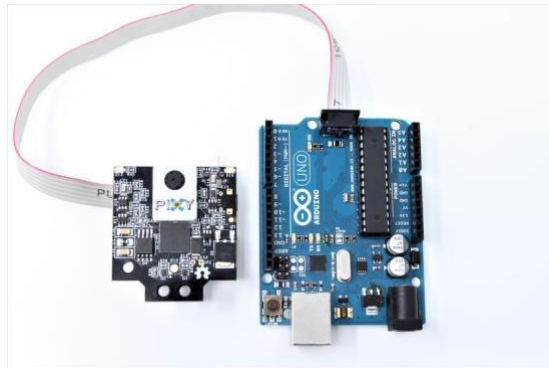


As a result, we printed a model on a polymer printer. and using an electroplating process, NiCr coating was applied.



## 2. Ball Detection

To define the ball, we use the omnivision system and the Pixy2 camera. Pixy2 can memorize up to 7 signatures (colors), find hundreds of objects in one frame and process up to 60 frames per second, so this camera is very convenient to use in the Open League.



Camera setup and signature training is done in PixyMon utility.

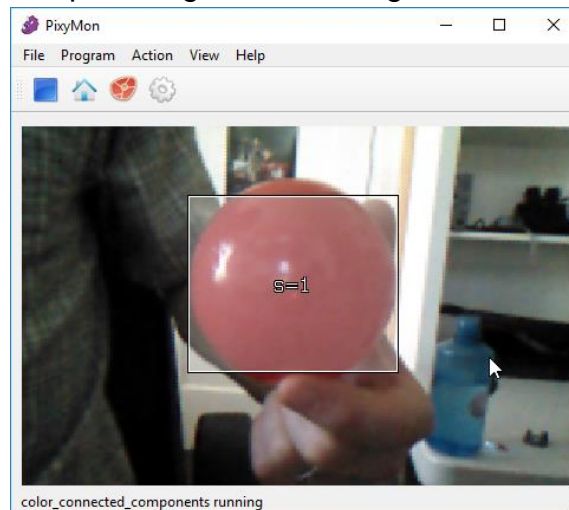
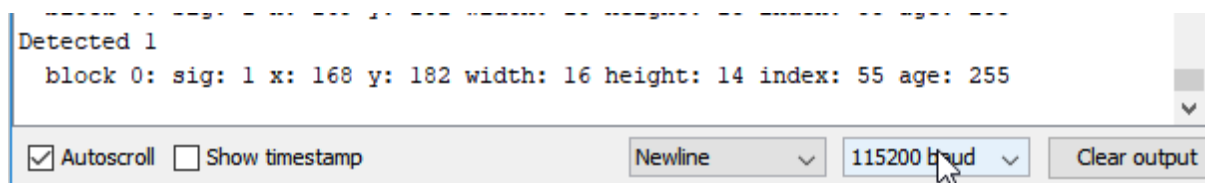


Image processing takes place on the camera's processor and, using the SPI protocol, an array of the found blocks (objects) is sent to the controller.



Each block contains 7 parameters:

**sig** - signature sequence number

**x** - the central pixel of the block along the X axis

**y** - the center pixel of the block along the Y axis

**width** - block size along the X axis

**height** - the block size along the Y axis

**index** - the ordinal number of the block, id

**age** - block age (the number of consecutive frames on which this block was found)

## 2.1 Algorithms for processing data from the camera

As spherical mirrors are installed on our robots, it is necessary to correctly process data from cameras.



## 2.2 Calculating the angle between the front of the robot and the object

We use the angle formula between vectors on the plane.

$$\cos \alpha = \frac{(\vec{a}, \vec{b})}{|\vec{a}| |\vec{b}|}$$

In this case, one of the vectors is directed forward by the robot.

$$\vec{a} = (0, 1)$$

And the second vector has coordinates

$$\vec{b} = (x - x_0, y - y_0)$$

Where

x - the center of the block on OX,  
 x0 - the center of the robot on OX,  
 y is the center of the block by OY,  
 y0- robot center according to OY

Also note that  $\alpha \in (0, \pi)$ , so you need an extra check  $x \geq 0$  or  $x < 0$

The final formula looks like this:

$$\alpha = \arccos\left(\frac{b_y}{\sqrt{b_x^2 + b_y^2}}\right) * \text{sign}(x)$$

Calculate the distance from the robot to the object

We use the formula for the distance between two points on the plane.

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

Where

x - the center of the block on OX,  
 x0 - the center of the robot on OX,  
 y is the center of the block by OY,  
 y0- robot center according to OY

Image distortion is calculated, we work with a non-linear value.

### 2.3 Basic ball handling algorithm



```

void followBall() {
if (ballFound) {
dir = ball.dir; // ball angle relative to the robot's front
if (ball.dist < 70) { // if the distance to the ball is small
if (abs(ball.dir) < PI / 2) { // if the ball is in front of the robot
dir = ball.dir * 2; // approach to the ball by parabola
} else { // if the ball is behind - departure to your goal
dir = home.dir;
}
float ang = constrainAngle(ball.dir - yaw);
if (abs(ang) > 2.2 && ball.dist < 57) // a detour of the ball if it is behind
dir = ang + PI / 2 * sign(ang);
}

if ((ball.x > 190 && ball.x < 200 && ball.y > 121 && ball.y < 140)) { // yspeed if the
ball is in the ball grab zone
dir = target.dir;
speed = BOOST_SPEED;
}
}
}
}

```

### 3. Goal and Direction Detection

The robot distinguishes the enemy's gates and their gates according to the indication of the gyroscope. The robot has a SparkFun 9DoF Razor IMU M0 gyroscope



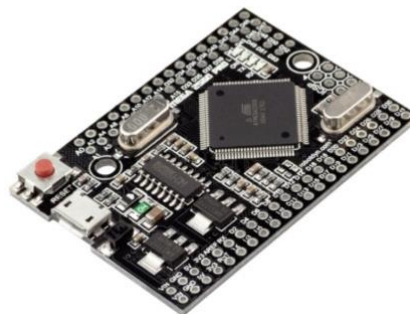
This gyroscope is characterized by high performance and high accuracy. Since the data is processed on board and there are built-in filters.

Robots have 3 ultrasonic distance sensors.



The sensors are located at the bottom of the robot and help the robot to position itself at the gate. One sensor looks back and regulates the distance to the gate. The remaining two are located on the sides and help the robot stay in the center of the gate.

The board we use for processing logic and signaling to peripherals Mega 2560 PRO. This board has a compact body and a sufficient number of ports.

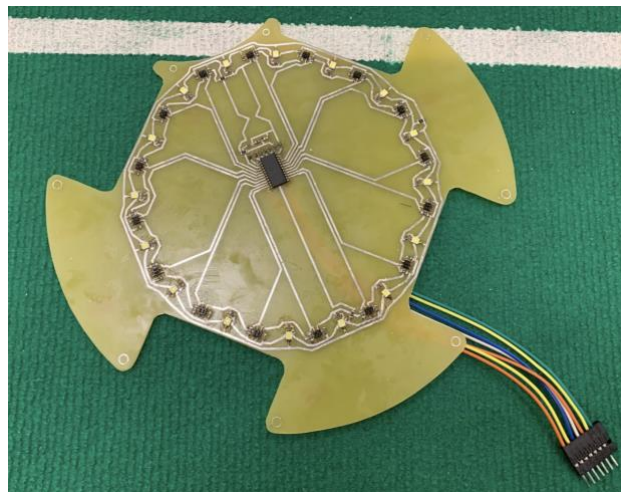


## 4. Out Line Detection

In the first iteration, we use 16 QRD1114 sensors and 16 LEDs to determine the line.



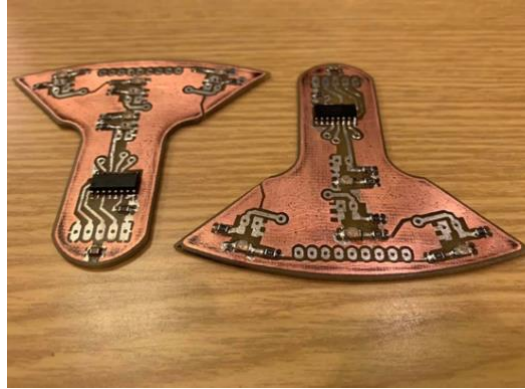
We connected the sensors through a multiplexer. Sensors are arranged in a circle to detect the line as early as possible.



While debugging the sensor, we displayed the readings of the sensors and selected the required height to install the sensor so that the readings on white and green were quite different. We also experienced difficulties in producing such a large sensor.

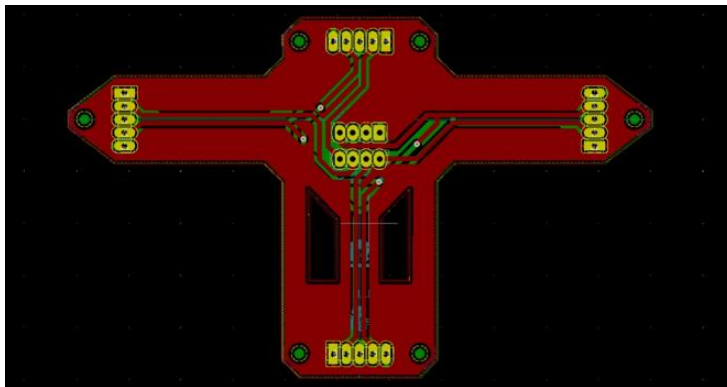
As a result, it was decided to make the system of modules. In total, there are 4 sensors on the robot on each board, there are 4 phototransistors, 4 LEDs (white), a multiplexer and resistors. In the beginning, we used red LEDs, but it turned out that the camera of the robot does not see the difference between the gamma of the sword and the gamma of the LED, so we replaced them with white LEDs.

Handmade Prototype:



This design allows you to win in the speed of the survey of sensors and, accordingly, in the speed of the system as a whole.

The sensor boards are interconnected by an intermediate board.



## 6. Board manufacturing process

After we spread the board, we create a prototype using the LUT method (laser-iron technology) and check its performance and for errors in the wiring.



LUT method steps:

- 1) We print the PCB layout on photo paper
- 2) we clean the copper on the textolyte with sandpaper from dirt and grease marks

- 3) Before drawing the picture, we lower the textolite into the ferric chloride solution to completely remove the fat traces
- 4) Attach the drawing to the future board and iron it with iron
- 5) We bring the board under running water and gradually, gently begin to rub the paper until it disappears completely
- 6) Remove toner from the board using emery paper
- 7) We fix the board
- 8) Making holes

After we have verified that the board is working and that there are no errors in it, you can order it in production.

## 7. List of Components

Name	Model	Link	Feature
Motor	20.4:1 Metal Gearmotor HP 12V	<a href="https://www.pololu.com/product/3203">https://www.pololu.com/product/3203</a>	Fast motor, high torque
Controller	Mega 2560 PRO	<a href="https://robotdyn.ru/catalog/arduino/boards/mega_2560_pro_embed_ch3_40g_atmega2560_16au.html">https://robotdyn.ru/catalog/arduino/boards/mega_2560_pro_embed_ch3_40g_atmega2560_16au.html</a>	A sufficient number of ports in a compact package
Power	4 x 18650	<a href="https://ru.wikipedia.org/wiki/Литий-ионный_аккумулятор">https://ru.wikipedia.org/wiki/Литий-ионный_аккумулятор</a>	compactness, low weight, availability, low cost, high current
Motor controller	TB6612FNG Dual Motor Driver Carrier	<a href="https://www.pololu.com/product/713">https://www.pololu.com/product/713</a>	Light weight, small, inexpensive
Gyroscope	SparkFun 9DoF Razor IMU M0	<a href="https://www.sparkfun.com/products/14001">https://www.sparkfun.com/products/14001</a>	High performance, high accuracy
Wheels	Wheels	<a href="https://gtfrobots.com">https://gtfrobots.com</a>	Light weight, narrow, which saves space in the robot
Camera	Pixy2	<a href="https://pixycam.com/pixy2/">https://pixycam.com/pixy2/</a>	Processing in the camera itself, an intuitive setting

## 8. Strategy

Due to the high pace of the game, and the wide gates, we decided to abandon the tactics of the two forwards and the changing roles of the goalkeeper and the attacker between the robots. Robots are strictly divided by roles: the attacker and the goalkeeper.

The goalkeeper almost never leaves the penalty area and protects the gate. An attacker, on the contrary, usually does not enter the zone of his goalkeeper, so that the robots do not collide, do not interfere with each other and there is no double defense.

The arduino ide is used for programming the goalkeeper. The program uses several library libraries for work, as well as several functions and procedures.

For example, the Pixy2.h library is used to work with the camera, I2Cdev.h for working with a gyroscope and a laser rangefinder. A list of procedures and functions with a description:

updateUMI () receiving data from the gyroscope,

checkLine () checks data from line sensors,

checkButtons () handles buttons.

The robot works according to the following logic, at first it searches for the gate, returns to them for a certain distance specified by a constant in the code, then searches for the ball, if it finds it, then moves to the side towards the ball, if it does not find it, then you are in the center of the goal, if the ball approaches quickly or close to it, it leaves for him, and then returns to the gate, if the robot sees the line, it also returns to the gate.

## 9. Gratitude

At the international competitions RoboCup Asia Pacific Iranian teams have provided us with great help. In particular, the team Sudo and Barcode-Technorobot and many others. The teams told in detail why they use certain components. talked about the strategies and principles of their robots. We also maintain close contact with the Russian MVK team.

Financial support is provided by Center for Teacher Excellence.

10. Final robot

**ta-da!**

