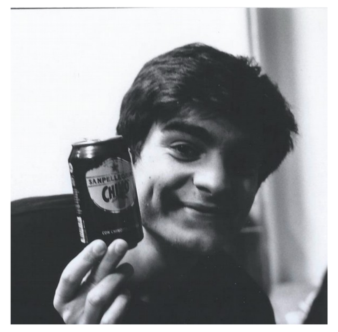
OUR TEAM

SPQR is a project born several years ago in ITIS G. Galilei, Rome. After winning for the tenth year in a row the italian soccer lightweight, we worked to completely renew our robots and show the world our true capability.

EMANUELE LATINO AGE: 18

ROLE: SOFTWARE DEVELOPER, TEAM LEADER

HOBBIES: FILM PHOTOGRAPHY, MUSIC



ACHILLE GHEZZI AGE: 19

ROLE: HARDWARE, CHASSIS DESIGNER, MECHANIC

HOBBIES: CLIMBING

DARIO CASAGRANDE AGE: 19

ROLE: HARDWARE, PCB DESIGNER

HOBBIES: PHOTOGRAPHY, HI-FI ENTHUSIAST

EMANUELE COLETTA AGE: 16

ROLE: SOFTWARE DEVELOPER

HOBBIES: GAME DEVELOPING, READING

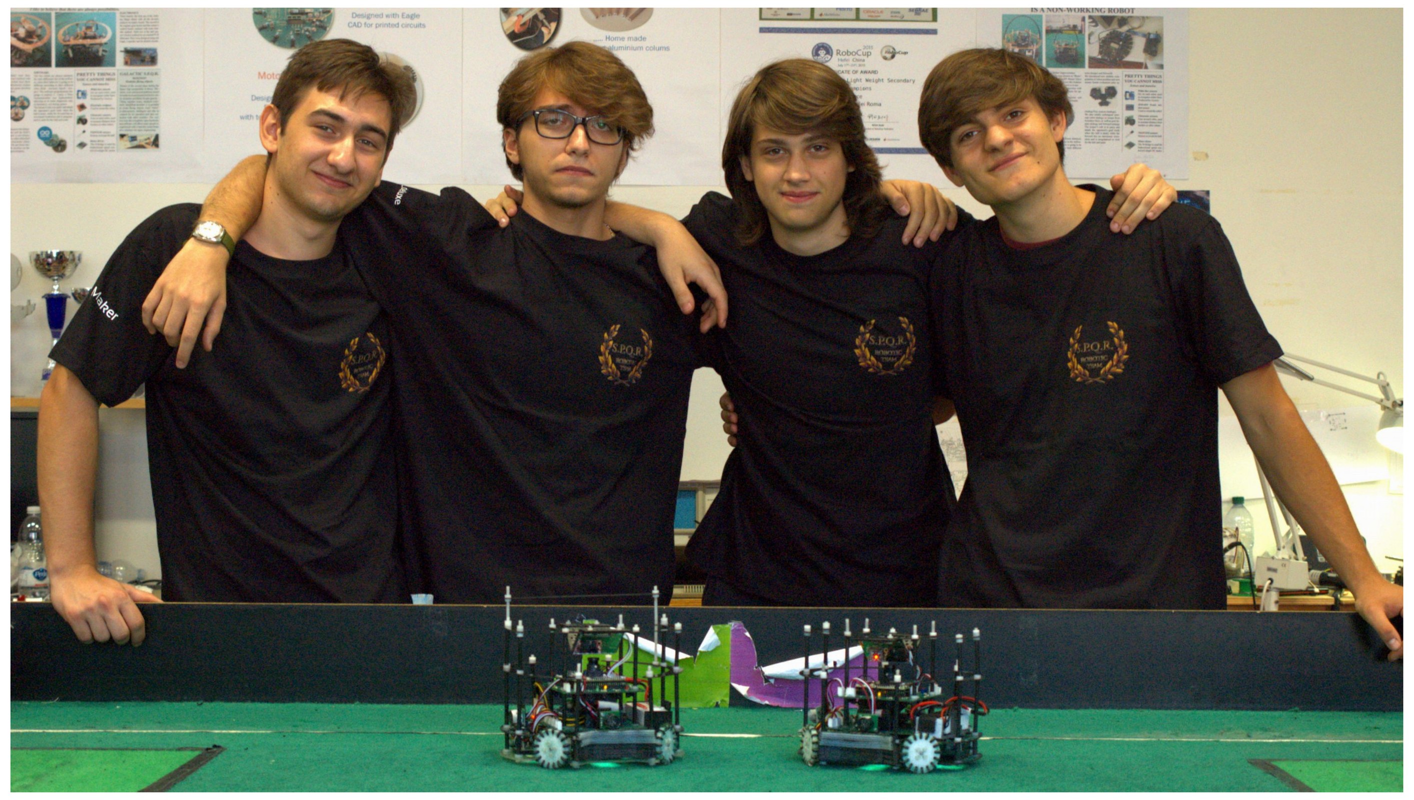
S.P.Q.R.

ROBOCUP JUNIOR SOCCER LIGHTWEIGHT

Polo Tecnico Professionale Galileo

I.T.I.S. Galileo Galilei

Il mondo della tecnologia



THEY PLAY SOCCER!

WHAT SHOULD THEY DO?

spqrrobotics.tk

@spqr_robotics

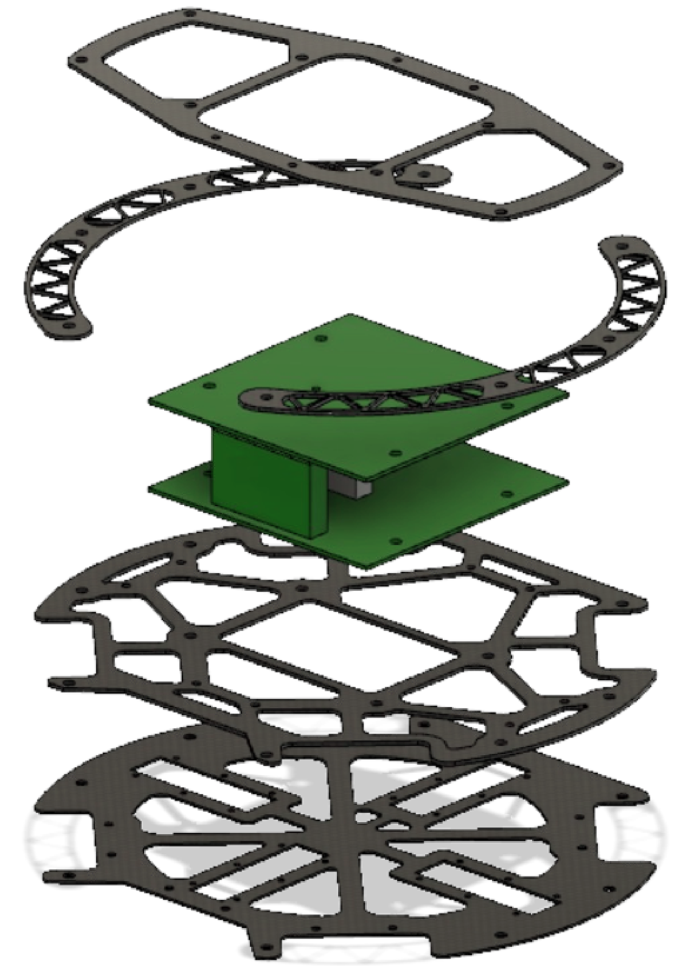
@latino_deluxe

github.com/latino-deluxe

RoboCup2019
2-8 July 2019
Sydney | Australia

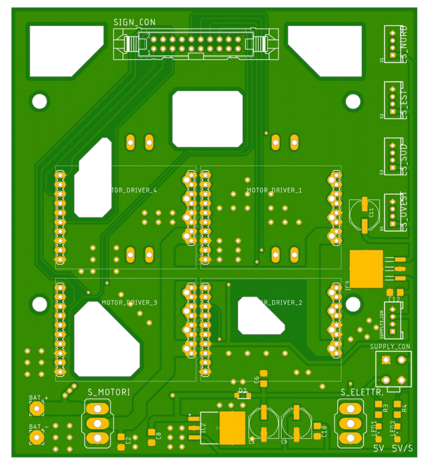
THE CHASSIS

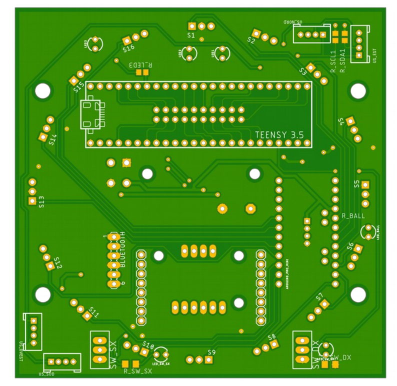
Every part of the structure was carefully designed to be as light as possible without losing the strenght we've always relied on.
Camera support, crowns and PCBs support are made of 1mm carbon fibre: these parts don't need to be extremely solid, so we kept them thin and elastic.
The lowest board works the opposite way: it keeps everything together and absorbs much of the shock the robot copes with, so it must be rock solid; the Goalie has a 1,5mm carbon fiber motor support, which makes it elastic, agile and more reactive. The Keeper needs to be as stable and accurate as possible, so it is equipped with a 2mm fiberglass board.
Everything is kept together by carbon fiber 4mm columns.
The overall setup for both the robots is pretty stiff and with a low center of mass, thus allowing us to move on the field with great accuracy.



With 4 motors,we had to do some real weight management. What if carbonf fiber and a new structure aren't enough?

PCBs



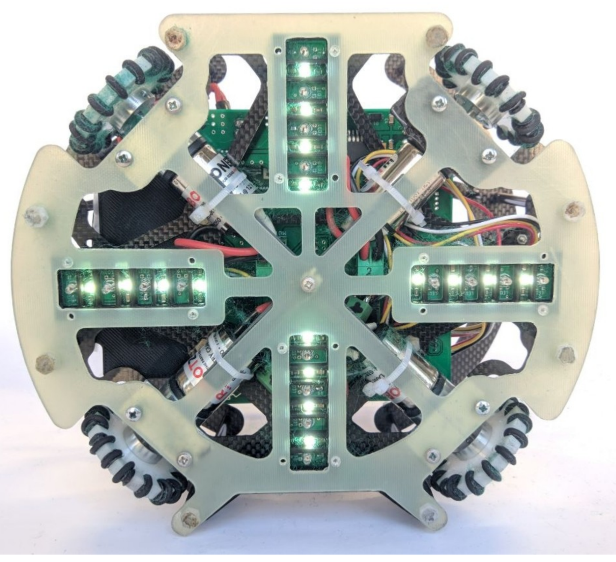


AUTODESK® EAGLE

We made our smallest PCBs ever: 10x10cm and 11x12cm. They're also extremely light, 70 grams less than last year's boards.
Carefully chosing the right SMD package for almost every component, we designed a 4x2cm buck converter capable of 1.5A without any heat dissipation.

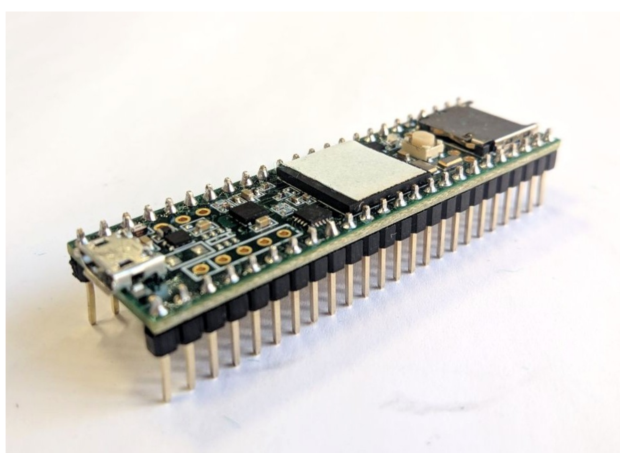
WHAT'S NEW

We improved so much during 2019 that's almost impossible to describe all the new components. We came out of our comfort zone and left behind everything to create a new robot really capable to fight for the first places. About six months and 2000€ later we had the final version of our robots.
Only the competition will tell us if we did right.



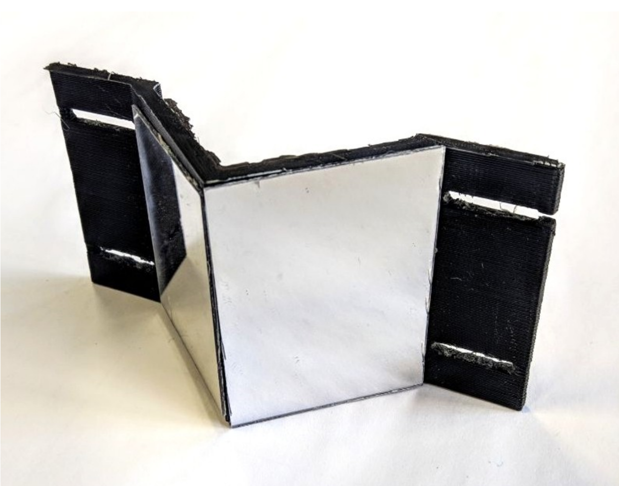
4 MOTORS

Even if that means more weight, 4 motors layout significantly improves robot's grip on the field, which has positive repercussions on movements' precision, speed and robot's ability not to go out of bounds.
The overall performance takes a huge step forward.



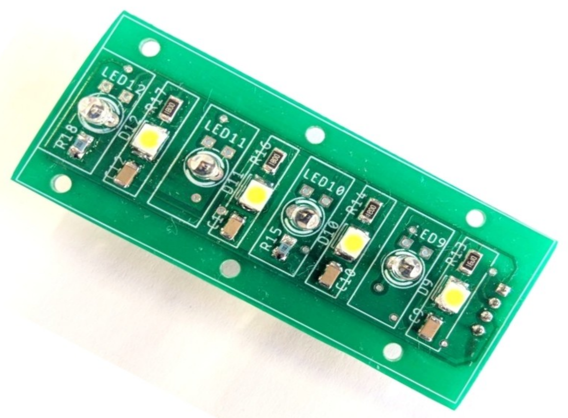
TEENSY 3.5

Teensy's advantages compared to Arduino Mega are infinite.
It is way lighter, has more capability and many I2C buses to separate stuff. We can also interpolate the ball sensors' data, with the loop lasting only 1,2ms.
Teensy 3.5 is also 5V tolerant, making it compatible with all of our sensors.



"V" SHAPED MIRROR

We found the "V" shaped mirror is the best compromise between weight, ease of use and sight. With a simple and light structure made of carbon and plastic we are able to see both the goals, helping us scoring and recentering the robot.



ORIGINAL LINE SENSORS

Instead of the old digital sensors we chose to build our own line sensors. Since the new ones return an analog value, we can adjust the reading via software, ideal when the fields aren't exactly the same color.
Over this, we put them in a cross shape, to undestand better where the robot exactly is.

ABOUT THE SOFTWARE

New microcontroller, new rules.
To take full advantage of the Teensy's capabilities, our programmers had to work hard.

VS CODE & PlatformIO

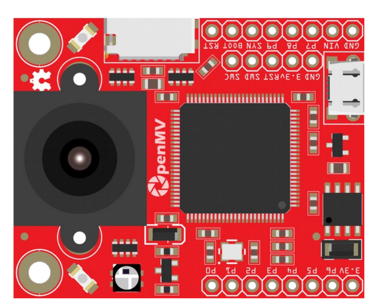
Due to its flexibility and its large community we program with Visual Studio Code paired with PlatformIO, a cross-platform IDE and unified debugger.

What else?
It's lightweight and works fine on Linux.

MODULAR ORGANIZATION

Since the robots share the same code, we need it to be modular.
Many tasks have their own functions, so that we can intervene on small portions of the software, without affecting anything else. We also wrote code in C++, which is more powerful and flexible than a simple Arduino sketch.

openMV



OpenMV M7

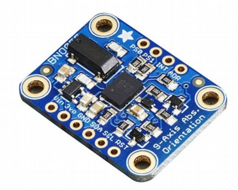
It's programmed in MicroPython through its own IDE, so we needed some time to make it work,but now it's crucial for two reasons: defining the keeper's position in the field and pushing the ball towards the biggest free part of the goal.

WORK TOGETHER

We use GitHub for three reasons:
- working on the code simultaneously
- keeping old, working versions of the software in case of emergency
- keeping in mind what's still to do and what's been done.

SENSORS

IMU



As the last years, we use Adafruit/Bosch BNO055. It's extremely reliable, light and works great upside down, which is great for making a small PCB.

BALL SENSORS

We chose to use 16 Vishay TSSP4038 IR sensors divided in 4 supply blocks.
Interpolating data from every sensor we're able to find the ball with great accuracy and less sensors than last year.

DISTANCE SENSORS

We use 4 US distance sensors (SRF-02) to understand our robots' position.
Being I2C devices we can have as many as we want on a single serial interface.