Soccer Open – Team France

LUDOVATECH

Abstract

LudovaTech is a robotics team composed of four students from Louis-le-Grand High School in Paris, France. We take part for the first time in the RoboCup Junior Open League.

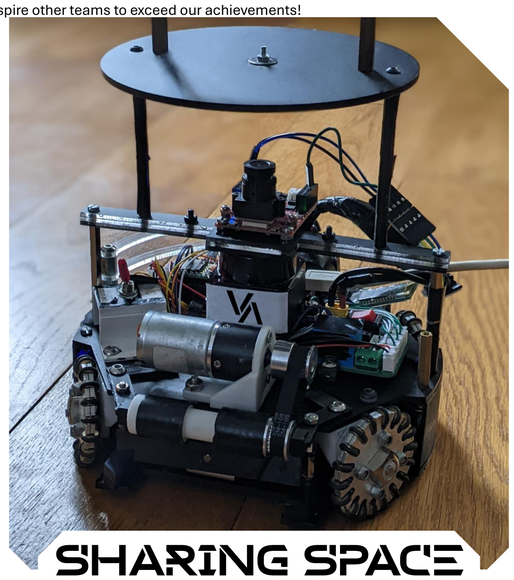
Since October 2023, we have built the robot from scratch, participating in both regional and national competitions. Comparing our robot with those of other competitors has allowed us to enhance our own design, leading to the current version. Each of us has dedicated hundreds of hours designing, building and programming two fully autonomous robots.

Our robot's functioning is composed of four primary processes. Initially, it gathers crucial environmental data using a 360° Lidar sensor LD-19 and an OpenMV RT1062 camera. The Lidar sensor is chosen for its practicality, reliability, and compact size, while the camera employs a Python-based algorithm specifically designed to detect the ball and goals.
Secondly, the collected data is processed by a Teensy board, which interprets the information and determines the appropriate actions using programs written in C++.
Next, smooth communication among all components is ensured by two self-designed printed circuit boards and homemade cables, ensuring seamless data transfer.
Finally, the robot executes the selected actions through its mechanical components, such as the kicker and brushless motors with integrated drivers for the wheels and dribbler.

Nearly all elements were self-designed to meet our specific requirements. Building these components by our own brought us enjoyment, enabling us to participate without relying on a large company and within the budget of a high school club. Through this hands-on approach, we have learned invaluable lessons. A significant amount of time was dedicated to trial and error, particularly in debugging C++ code. We also built Python tool apps to analyze data and logs and speed up the process.

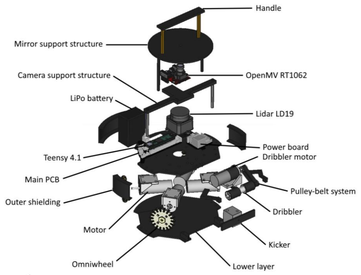
All structural robot components were self-manufactured with a laser cutter and 3D printer. The structure is centered around a single laser-cut hard plastic plate with the camera and mirror placed on elevated supports. The bottom cover plate of the robot can be easily removed, enabling quick access to any component. This design facilitates efficient maintenance and accessibility.

This poster details our background in creating these robots, sharing our discoveries and decisions as openly as possible. Our source code is available under an open-source license, and we have created informative tutorials featuring useful tips for fellow teams. Our objective is to inspire other teams to exceed our achievements!



Method, Production and Design

Component selection & bill of materials



Main PCB
- Teensy 4.1 microcontroller
- HC-05 Bluetooth communication module
- Custom connectors to other electronic components of the robot (mostly JST PH 3pins)

Dribbler & kicker
- GP24-BL2430 brushless motor 1500RPM
- Pulley-belt transmission system
- 3D printed dribbler with custom metal shaft
- JF-0826B solenoid and SRD-03VDC-SL-C relay for kicker

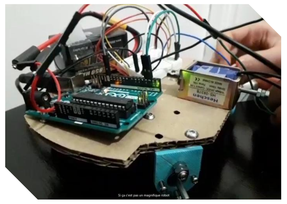
Powertrain
- 4x GP24-BL2430 brushless motors 375 RPM with integrated drivers
- 4x GTF Robots 50mm wide omniwheels
- 4x 3D printed custom motor mounts

Power
- DOGCOM 6s 850mAh LiPo battery for motors and kicker
- XL4005 5V regulator for main PCB
- Custom designed power board with self-made connectors to motors

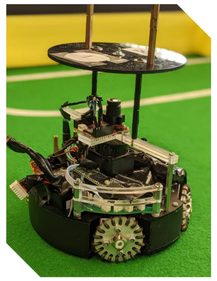
Sensing
- OpenMV RT1062 camera
- Custom mirror with 3D printed mold and self-applied reflective sheet
- Rotating LD19 LiDAR

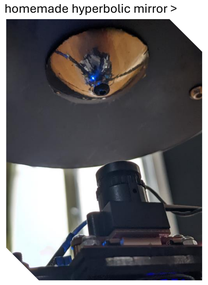
Evolution

At the beginning of the year, we embarked on developing our robots, making steady improvements along the way. Initially, we equipped the robots with omnidirectional wheels, motors, and a Teensy microcontroller, allowing us to create basic prototypes like the three below.
Subsequently, we focused on developing the camera and lidar systems separately before assembling the entire robot for the regional competition. Despite our efforts, this initial attempt was not successful.
For the national competition, we rebuilt the PCBs and other components, resulting in a perfectly functioning robot. In preparation for the international competition, we added advanced features such as a kicker and a dribbler.
Throughout the year, we have consistently enhanced our robot step by step, achieving continuous improvement. In all, we have spent in average of 800 hours of work and 1500 euros for all three models constructed in our high school workshop.



< our robot through time, at the start of the year, before the regional competition and at the national competition >

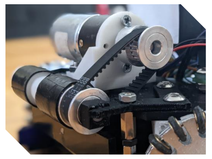




< our camera and its homemade hyperbolic mirror >

Design

Camera : The camera helps us pinpoint three key elements: allied goal, opponent's goal, and the ball. We built ourselves a hyperbolic mirror, by heating a reflective film on a 3D-printed and polished support, as you can see on the picture above on the right. This way, the camera continuously faces the mirror and simulates the functionality of a 360° camera.

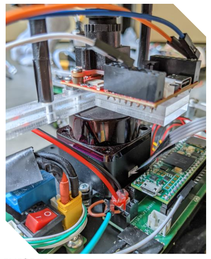


< our final homemade dribbler>

Dribbler : Our dribbler mechanism is quite straightforward. It features a brushless motor, identical to the ones used for our wheels, which powers a belt drive system. To enhance the grip, we needed an adhesive material to wrap around the bar. After some experimentation, we discovered that an old bicycle pump pipe worked exceptionally well for this purpose.

Kicker : Like many teams, we use a powerful solenoid to kick the ball into the goals. Our homemade PCB circuit includes all the components needed to control the kicker from the board, especially the relay, which allows us to safely deliver high voltage to the kicker.

Lidar : The LD-19 360° lidar sensor provides precise localization for our robot on the field. Despite its high accuracy, we encountered significant challenges: there were no tutorials available, and the existing documentation was sparse, incomplete, and sometimes incorrect. This is unfortunate because the LD-19 is one of the few small, low-cost lidar systems capable of operating over short distances with high accuracy. After several days of intensive effort, we finally succeeded in accessing and reading its data. Believing in the value of shared knowledge, we have created a comprehensive documentation-tutorial for this lidar. You can access it by scanning the QR code at the bottom of this poster. We hope this encourages more use of this technology by other teams.

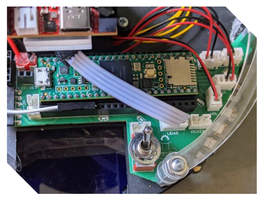


< our lidar inside the robot, on the right the Teensy and its board, on top our camera and on the left the "power board" >

Programming

We have divided our logic across two different boards.
The camera uses OpenMV libraries and Python, allowing us to enhance and speed up our color-based detection system.
The Teensy board serves as the heart and brain of our robot. It receives encoded information from all sensors, including the camera, decodes it, and interprets it to determine the most appropriate strategy. We use C++ for its speed and flexibility, adhering to a fully object-oriented programming paradigm. Inspired by the Rust programming language, we have developed our own system of utility classes from scratch. This system provides null-safety and fast error-handling within the Arduino-C++ environment, enabling us to develop code in a secure, lightweight manner, leading to improved productivity and fewer runtime bugs.

To facilitate teamwork and programming, we utilize tools like Git and GitHub. GitHub Projects helps us organize tasks agilely, while GitHub Actions allows for continuous testing and integration.
We believe in the power of mutual inspiration to create better software. That's why we've made our entire source code available under an open-source license on GitHub. You can access it using the QR code at the bottom of this poster.



< A top-down view of the Teensy and its board. This is where all the information needed for decision-making is concentrated. >

Data, Results and Discussion

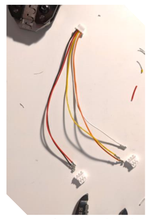
PCB and wires : Voltage Discussion

During the regional competition, our newly finished robots exhibited unexpected bugs, causing erratic behavior and even burning some components. This experience was traumatizing and refocused our efforts on making the robots more reliable for the national competition. We implemented numerous hardware changes to achieve this goal.

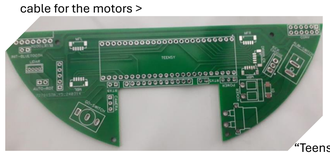
At the regional competition, we used a single custom PCB, designed with EasyEDA, that powered the motors with 24V and the other components with 5V. We suspect there was some unwanted interaction between these voltage circuits. To prevent a similar issue, we took two key steps:

PCB Redesign : We created two separate PCBs for each voltage. The only connection between the boards is a cable linking their ground. Additionally, we used a relay for the kicker to ensure the 24V and 5V circuits stayed isolated. One "power board" powers the motors and the kicker with 24V, while a "Teensy board" hosts the Teensy microcontroller and connects to all components using 3.3V and 5V. We designed these PCBs with foresight, allowing us to integrate the kicker and dribbler without needing new components.

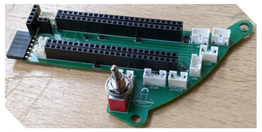
Simplified Connections : We needed a way to easily connect components while accommodating unique circuits, such as the motor circuit where two of the five motor pins go to the power board and three to the Teensy board. We chose 3-pin JST PH connectors for all connections to facilitate easy repairs and assembled and crimped our own cables.
This new system worked perfectly: we did not burn any component during the national competition.

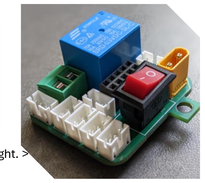


< one of our handmade cable for the motors >



< our first self-designed PCB >



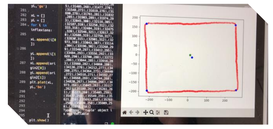


< the second generation of our PCBs, with the "Teensy board" on top and the "power board" on the right. >

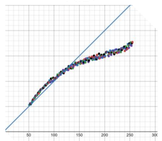
Lidar Data Analysis

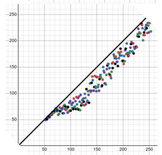
Once we receive data from the LD19 lidar, we process it on the Teensy microcontroller by using the Hough Transform algorithm on the point cloud in order to find the walls and consequently the corners as well. This precisely determines the robot's position on the field, enhancing our strategies and eliminating the need for numerous line sensors.

The robust algorithm can accurately detect the robot's true position even with significant obstacles. In rare failures, it can still measure the distance to the nearest wall to prevent the robot from leaving the field. The lidar's reliability has led us to prioritize it over the camera as our main source of information for the strategies.



< Our Python app displaying lidar data and Hough transform results >





< Here is an example of data analysis we had to perform. We had to find the function that properly mapped motor speed measured with the hall sensors to their actual speed. >

Ball and Goals detection

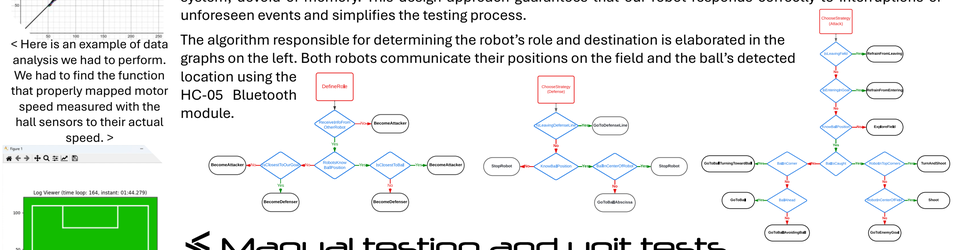
We use our OpenMV RT1062 camera to code in Python an object detection system that relies on identifying colors in the captured images. The pixel coordinates of these objects are then obtained on the hyperbolic mirror.

Additionally, based on the side of the blue or yellow goals, the camera helps us select the correct position from the two provided by the lidar. However, the camera remains the most delicate component of the robot's system: color detection is highly complex and requires recalibration at halftime to adjust for changes in lighting and field side.

Logic and Strategy

After our unsettling experience at the regional competition, we meticulously devised strategies to ensure their straightforward and secure implementation. Our entire system is engineered to function robustly even in the absence of crucial information, leveraging multiple redundancy systems. These strategies operate as a stateless deterministic system, devoid of memory. This design approach guarantees that our robot responds correctly to interruptions or unforeseen events and simplifies the testing process.

The algorithm responsible for determining the robot's role and destination is elaborated in the graphs on the left. Both robots communicate their positions on the field and the ball's detected location using the HC-05 Bluetooth module.



Manual testing and unit tests

At the start of the year, our goal was to develop a virtual testing system. However, implementing it became impractical due to the complexity of generating virtual images with accurate camera simulations. As a result, we pivoted to conducting tests in real-life settings. To facilitate these tests, we constructed a portable testing field. To share our approach, we created a comprehensive guide for building similar setups, complete with essential information and practical tips !
Access the guide via the QR code located at the bottom of this poster. Conducting tests in real environments is time-intensive, so we designed tools to streamline the process. Our Matplotlib Python app automatically reads logs, interprets, creates a virtual replica of the testing field to display data on screen.

< Our homemade transportable field. >

< Our Python app analyzing the robot logs in real time. >

```
// Test: Cas de base avec une séquence complète "b...e"
Run | Debug | Show Log | Show in Test Explorer
TEST(CamReader, ExtractLastCompleteSequenceBasicSequence) {
    String input = "abcde";
    String expected = "bcde";
    EXPECT_EQ(String(extractLastCompleteSequence(input)), String(expected));
}

// Test: Séquence avec plusieurs 'b' et 'e', prendre la dernière séquence complète
Run | Debug | Show Log | Show in Test Explorer
TEST(CamReader, ExtractLastCompleteSequenceMultipleSequences) {
    String input = "abcde fghbijklmne";
    String expected = "bijklmne";
    EXPECT_EQ(extractLastCompleteSequence(input), expected);
}
```

To mitigate the risk of bugs detected solely through real-world tests, we rigorously tested most functions using unit tests. This posed a challenge, as we aimed for these tests to run on computers, at compilation time, rather than the Teensy platform. We successfully crafted a custom CMake file that tricked the code into recognizing one of our files as an Arduino library. This enabled us to execute unit tests using Google Tests not only locally but also in the cloud, ensuring continuous integration checks on GitHub.

< two of our many unit tests >

SHARING SPACE

Lidar LD-19 tutorial
github.com/LudovaTech/lidar-LD19-tutorial

Our source code
github.com/LudovaTech/robot-prog-public

Build your own field
github.com/LudovaTech/build-robocup-junior-open-soccer-field-tutorial

Our Team :

Thomas Diot, 16, software engineer
Antoine Guilmot, 17, software engineer
Yveline Liu, 17, manufacturing and assembly engineer
Rémy Magnon, 17, team captain and hardware engineer

Lycée Louis le Grand PARIS

FFROB FÉDÉRATION FRANÇAISE DE ROBOTIQUE

TFABOT.COM

ROBOT-MAKER.COM