Robocup2024EINDHOVEN
Presentation Poster

About as

We are a team from the Shibaura Institute of Technology Junior and Senior High School's Dengiken Club, competing in RoboCup Junior Soccer. We have built a remarkable record of achievements, culminating in winning the overall championship at the 2023 World Tournament. After undergoing a change in members, our team is now comprised of four high school students.

Oi_DENGIKEN

Shibaura Institute of Technology Junior and Senior High School

Team Member

Aoi Higaki |Electric
Taiyo Kishi |Software
Hiyu Ami |Software
Wakana Masada |Hardware
Kenta Matsumoto |Mentor

Team HP

http://dengiken.jp/oi-dengiken/

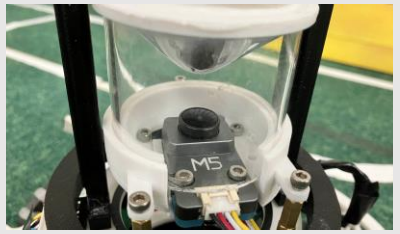
Sponsor

JLCPCB

HARDWARE

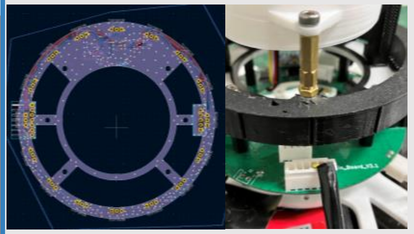
TOP

Omnidirectional Camera



Switching the camera from OpenMV to UnitV has improved the code extensibility of the system. Coupled with tailored firmware, the change has improved. With the introduction of the new camera system, offensive units have an easier time shooting at the goal, and defensive units find it easier to maintain their position, enhancing the quality of control.

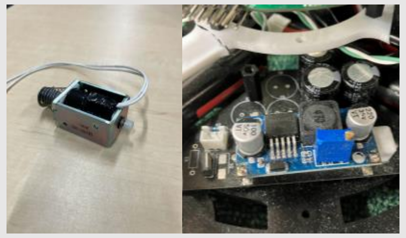
Ball Sensor Substrate



This platform is equipped with 16 high-precision ball sensors, capable of accurately detecting balls from all 360 degrees. Additionally, by adopting the STM32F446RE as a sub-CPU, it achieves high-speed data processing. The design of the sensor cover narrows the window opening while ensuring that balls from the intended direction are reliably captured.

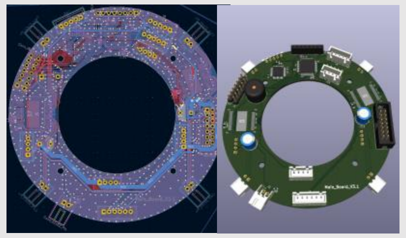
MIDDLE

New Gear [Kicker]



The ability to score goals has improved. By changing the resistor connected to the capacitor in the boost circuit that supplies power to the solenoid to a cement resistor, thermal resistance has been strengthened, allowing for more efficient rapid firing. In terms of body design, placing the solenoid in the most effective position has increased the shooting speed towards the goal.

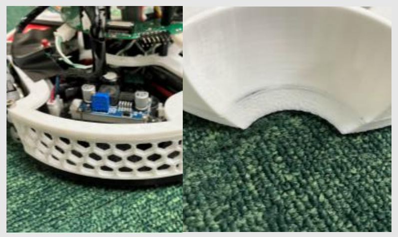
Main Substrate



We have made the airframe compact and easy to maintain by consolidating all wiring onto a main board. This main board uses the STM32F446RE, which is capable of high-speed processing, as the main CPU. Furthermore, the main CPU is able to concentrate on processing by leaving sensor reading to other microcontrollers, thus achieving high speed processing.

UNDER

Ultra-lightweight Frame



In pursuit of making the body lighter, a honeycomb structure was adopted for the walls, and the top plate was designed to be as minimal as necessary. Furthermore, by angling the hold area, the previous issue of balls bouncing out of the hold area has been resolved. This improvement ensures reliable ball holding, which in turn has increased the success rate of shots on goal.

Line Sensor Substrate

By arranging the luminance sensors in a large circular pattern, white lines can be detected accurately and over a wide area. Additionally, by installing luminance sensors as sub-sensors on the outside circle, the detection speed of white lines is improved. Furthermore, by adopting the STM32F446RE as a subCPU, similar to the ball sensors, highspeed data processing is achieved.

TOOL

Software

cubeIDE mbed MaixpyIDE

To control the STM, we used CubeIDE, and for camera control, we used MaixpyIDE.

Hardware

Kicad Fusion360

We created circuits with KiCad, design frames with Fusion, and print them using a 3D printer.

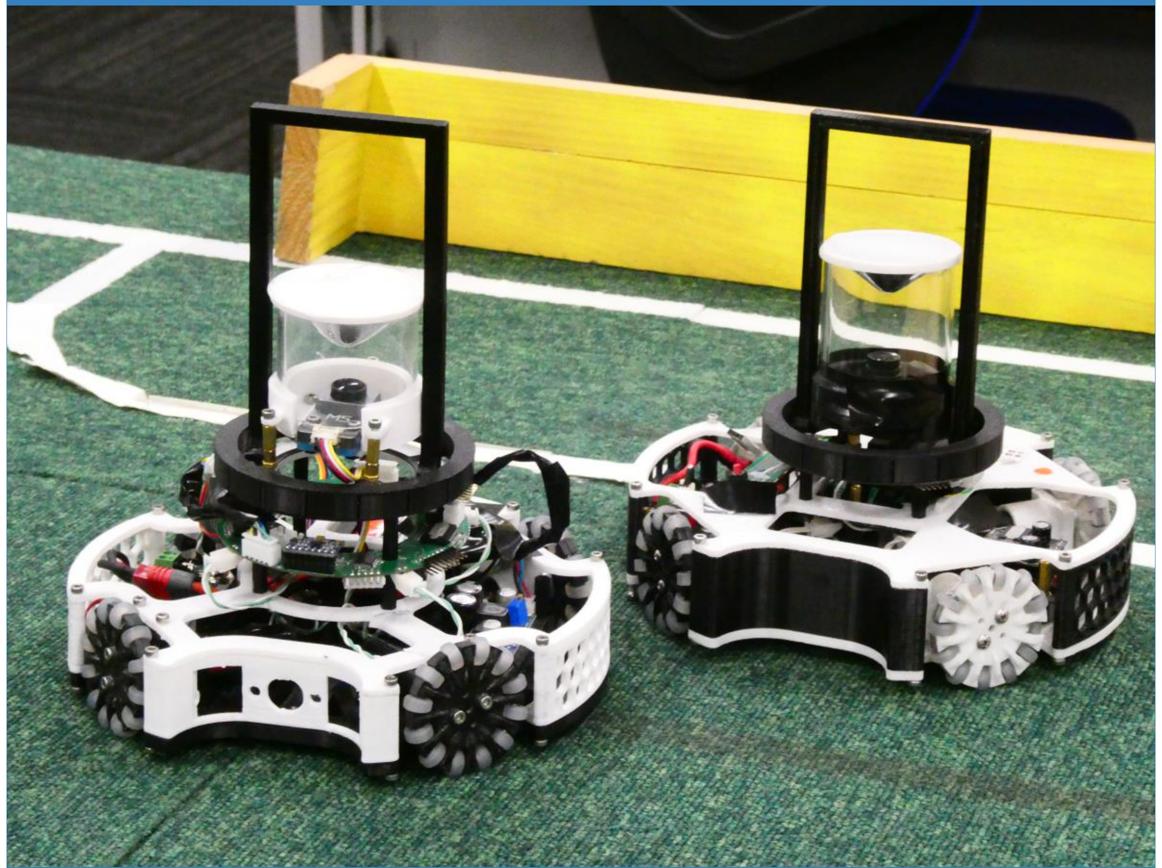
Others

Discord GitHub Notion

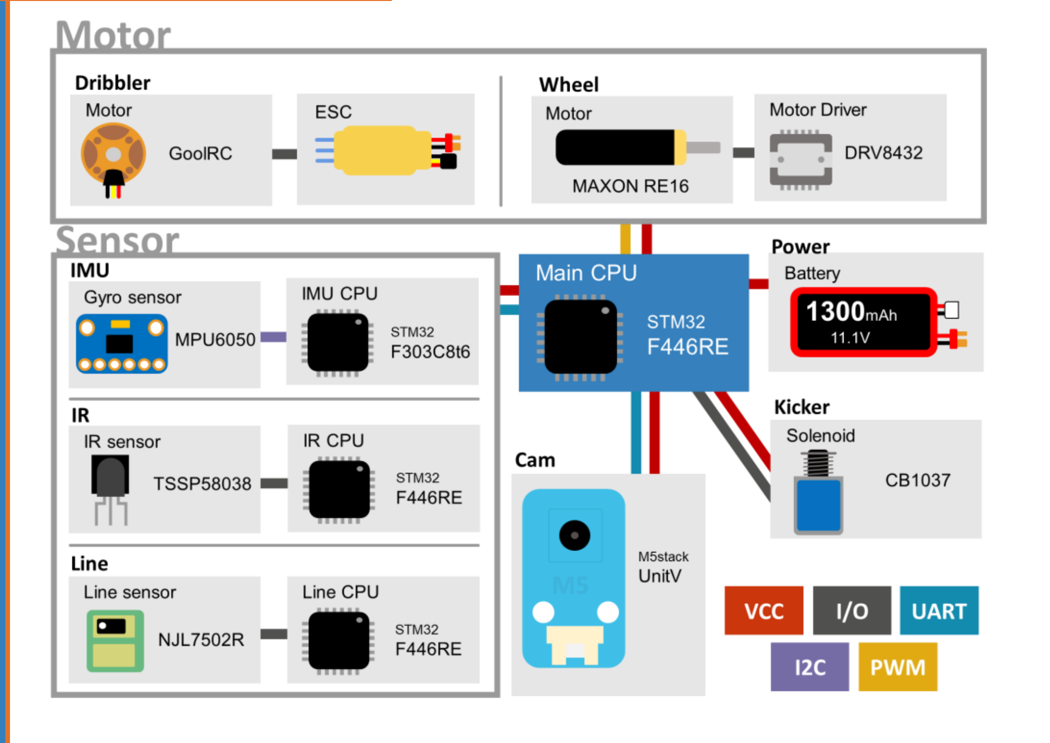
We communicate via Discord, manage files on GitHub, and handle task through Notion.

Development Concept

Our team is dedicated to developing robots with a focus on stability. We approach challenges by formulating hypotheses, identifying potential causes, and methodically verifying each one. This systematic process has enabled us to overcome numerous obstacles in robot construction. Moreover, we meticulously manage risks by utilizing reliable existing electronic boards alongside experimental boards with new features, thereby minimizing the likelihood of robot malfunctions. Additionally, our use of Git for code management further prevents errors and guarantees consistent operation. The robot's design facilitates easy disassembly, ensuring that any issues can be swiftly addressed, allowing for immediate resumption of collaborative work. Our commitment to these principles is driven by our aspiration to secure the world championship. We are determined to achieve outstanding results.



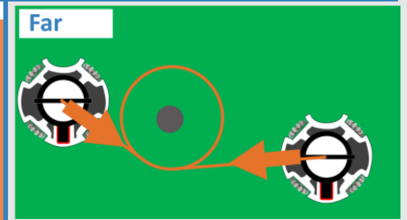
System Overview



SOFTWARE

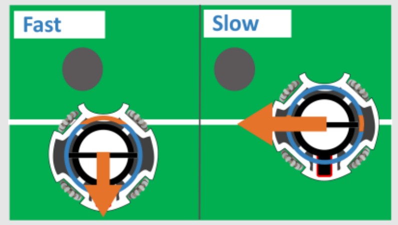
OFFENCE

Quick Wraparound



Flanking is performed using the tangent of a circle. If the distance is far, create a circle of any size centered on the ball and proceed towards its tangent. If the distance is close, flank at a 90-degree angle. Once in front of the ball, move forward towards the goal. This method improves code readability and allows for efficient and accurate shortest distance flanking by simply changing the radius of the circle for adjustments.

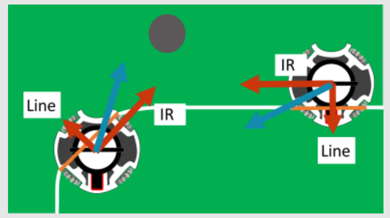
The Secrets of Line Control



We integrated the offense and defense line sensor programs to streamline debugging tasks. Moreover, when the speed of the machine is fast, it moves in the opposite direction of the initially recorded line, ensuring it doesn't cross the line even if it goes over. When the machine is slow, the defensive line handling technique can be applied to chase balls that would normally be out of reach by just barely crossing the line.

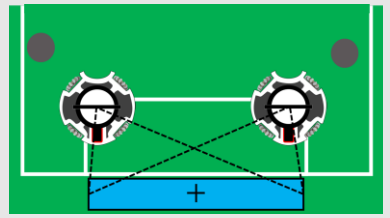
DEFENCE

Smooth Lateral Movement



By integrating the calculations of the IR sensor vector with the Line sensor vector, stacking actions were eliminated, allowing for rapid movement towards the ball. The Line sensor has been improved to accurately capture the angle at which the machine intersects with the line. This enhancement allows for clean line tracing at the goal ends, approaching the ball smoothly, and stopping precisely at the edges, making it easier to block shots.

Camera Defense



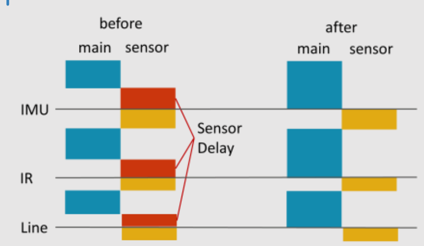
We use cameras to constantly recognize our goal, so we can keep track of where we are. This allows us to precisely calculate Current coordinates on the field. As a result, we no longer get stuck on the edge of the court and can't get back. Furthermore, if we are pushed out of the goal area by an opponent or our own machine, we can return to the goal and efficiently return to our original position. This stabilizes our vehicle.

COMMON

Fast Program Processing

Interrupt

In the previous system, the main program would remain in a waiting state until the MainCPU receive the data from the sensors arrived, resulting in delays in data processing. However, by changing the system to an interrupt-driven communication method, the waiting time of the main program was significantly reduced. When data from the sensors, an interrupt signal is generated, and the main program immediately begins data processing upon receiving this signal. This allows the main loop to continue other tasks while asynchronously waiting for the data to arrive. The system can now read data from the sensors more quickly, which has enhanced the overall processing speed.



Data Compression

In traditional serial communication, each data value was assigned 1 byte for transmission. However, by utilizing bit operators to compress the data, a new method was introduced to transmit with fewer bytes. This improvement reduced time-consuming processes such as UART, thereby enhancing the overall processing speed of the system.

New Development Environment

We transitioned the development environment from mbed to CubeIDE. Moving from the convenient high-level APIs of mbed to the more direct and detailed hardware control provided by CubeIDE has improved real-time performance and significantly increased the processing speed of the system.

Time of The Main Program

20,000μs Mbed timer function

↓

100μs cubeIDE DWT Cycle Counter