OI_Dengiken RoboCup 2024 Design Document

Aoi Higaki,Taiyo Kishi,Hiyu Ami,Wakana Masada
Mentor:Kenta Matsumoto

Shibaura Institute of Technology Junior and Senior High School,Japan

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

Abstract

This paper describes the innovative hardware and software developed by OI_Dengiken for participation in the RoboCup 2024 Junior Soccer Lightweights. Our robot incorporates new technologies such as dribblers and kickers and has significantly improved the performance of its line and ball sensors. On the software side, many innovations have been made to significantly improve both processing speed and accuracy. The process of developing these technologies and the experimental procedures will be described in detail.

1. Introduction

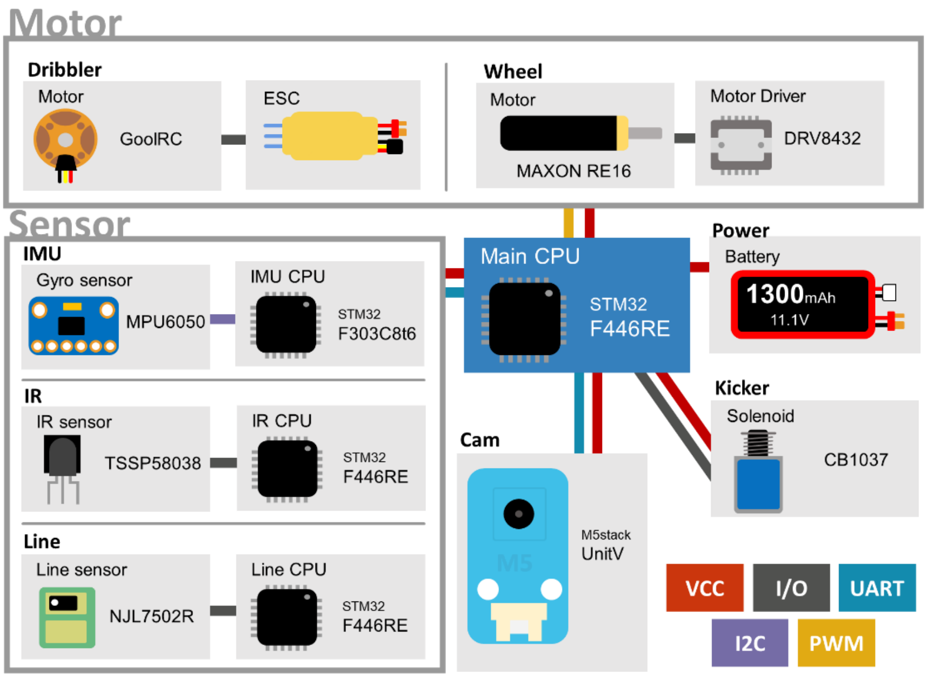
Our team was formed by members of the Electronic Technology Club of the junior high school and high school attached to the Shibaura Institute of Technology with the aim of winning the Japan RoboCup Junior competition in 2021, and we have made great efforts to realise this dream. They then participated in the national competition in 2022 and 2021 and successfully won the overall championship at the world competition in 2023. After a change of members, the team now consists of four high school students.

Aoi Higaki	Taiyo Kishi	Hiyu Ami	Wakana Masada
Circuit	Program	Program	3D Design

We have different roles, as shown in the diagram, in order to maintain a high level of technical competence in the team as a whole, with each of us focusing on a narrow area of expertise in depth. Our goal is to build stable robots. To achieve this, we hold regular meetings to check our progress. We use a task management tool so that we can check the progress and immediately start working on tasks. We have also worked on robot development with an emphasis on stability. We have solved many problems in robot production through the process of [hypothesis and verification], in which we formulate a hypothesis, list possible causes and verify them one by one when a problem arises. In addition, the risk of robot stoppages has been minimised through thorough risk management and by using a combination of highly reliable existing electronic boards and experimental boards with new functions on a trial basis. Furthermore, code management utilising Git prevents the robot from malfunctioning and ensures stable operation. The airframe design is designed to be easy to dismantle, so that if any defects occur, the structure is designed to fix them immediately and return to competition.

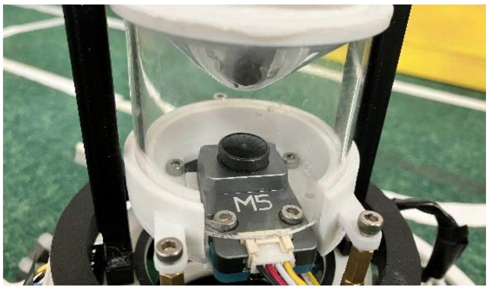
2. Hardware

a. System Overview



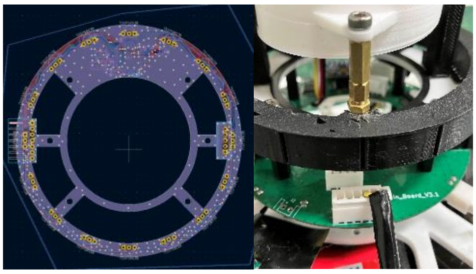
We are encouraging improvements in the hardware in order to create high-level software. We are trying to distribute the processing as much as possible in the robot system, so that each process is very fast and can be done quickly.

b. Omnidirectional Camera



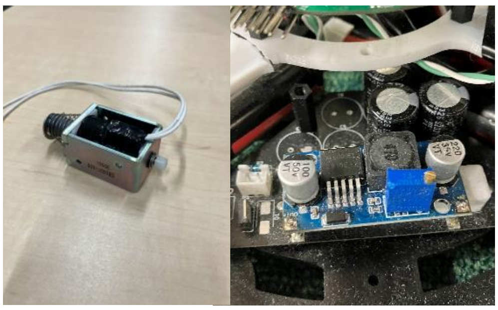
Switching the camera from OpenMV to UnitV has improved the code extensibility of the system. Coupled with tailored firmware, **the change has improved.** With the introduction of the new camera system, offensive units have an easier time shooting at the goal, and defensive units find it easier to maintain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

c. 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.**

d. New Gear [Kicker]

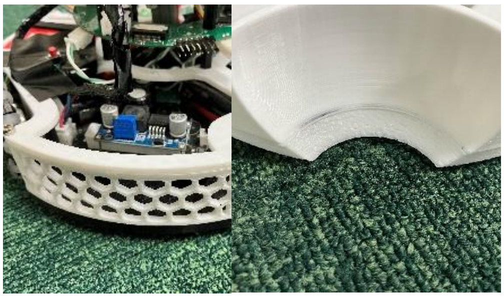


The ability to score goals has improved. By changing the resistor connected to the capacitor in the boost circuit that supplies power to the solenoid to a cement resistor, thermal resistance has been strengthened, **allowing for more efficient rapid firing.** In terms of body design, placing the solenoid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

e. New Gear [Dribbler]

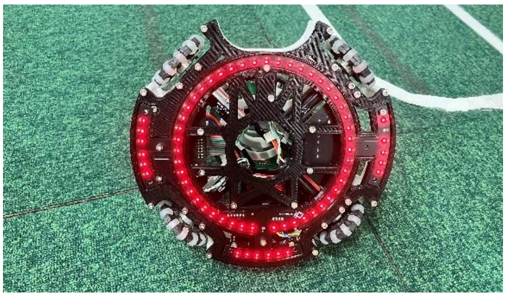
The introduction of the dribbler has **significantly improved ball retention.** The dribbler is designed to convey maximum holding power to the ball, **allowing for a hold and transition to shooting while avoiding stacks on the goal side.** Furthermore, the use of brushless motors has enabled precise control of the motor, **expanding the options for control.**

f. 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.**

g. Line Sensor Substrate

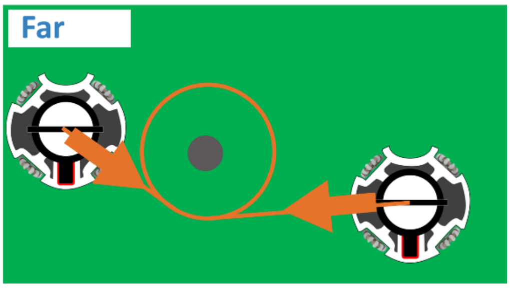


By arranging the luminance sensors in a large circular pattern, white lines **can be detected accurately and over a wide area.** Additionally, by installing luminance sensors as sub-sensors on the outside circle, the detection speed of white lines is improved. Furthermore, by adopting the STM32F446RE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

3. Software

A. Offence

a. Quick Wraparound



Angle>0

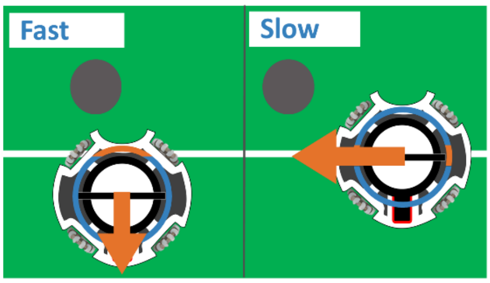
$$Angle + \tan^{-1}(-\frac{r}{\sqrt{a^2 - r^2}}) \times 180 \div \pi$$

Angle<0

$$Angle - \tan^{-1}(-\frac{r}{\sqrt{a^2 - r^2}}) \times 180 \div \pi$$

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.**

b. The Secrets of Line Control

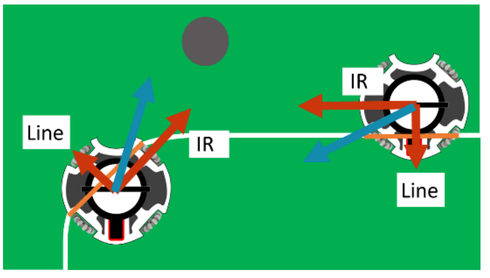


We integrated the offense and defense line sensor programs to **streamline debugging**

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.**

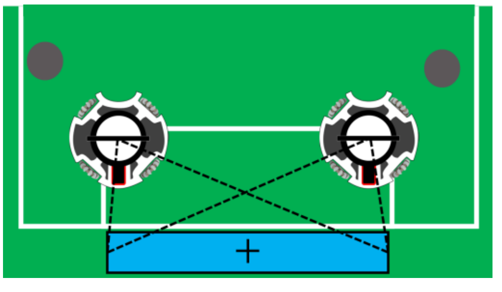
B. Defense

a. Smooth Lateral Movement



By integrating the calculations of the IR sensor vetor 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.**

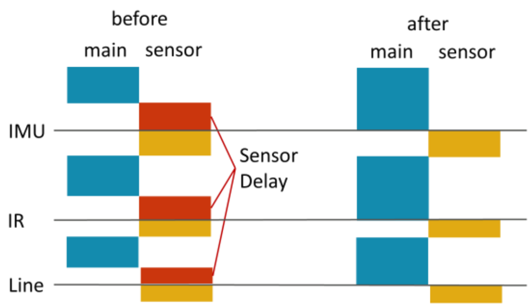
b. Camera Defense



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.

C. Common

a. 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.**

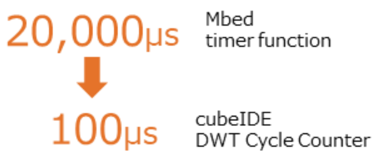
b. Data Compression

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.**

c. 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.**

d. Time of The Main Program



4. Tool

a. **Software**

cubeIDE mbed MaixpyIDE

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

b. **Hardware**

Kicad Fusion360

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

c. **Others**

Discord GitHub Notion

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