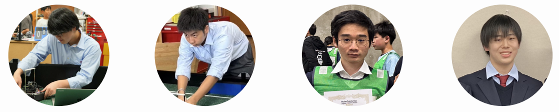
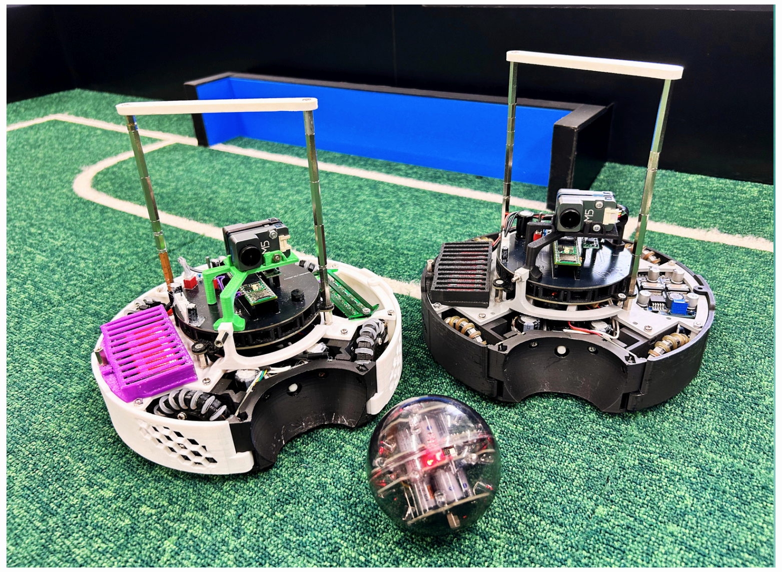
JAPAN SOCCER LWL



Captain, Electric: Uchida Asashi
Software: Kanou Kousuke
Hardware: Matsui Sora
Hardware: Nakazawa Kengo

Github
Blog
X(Twitter)



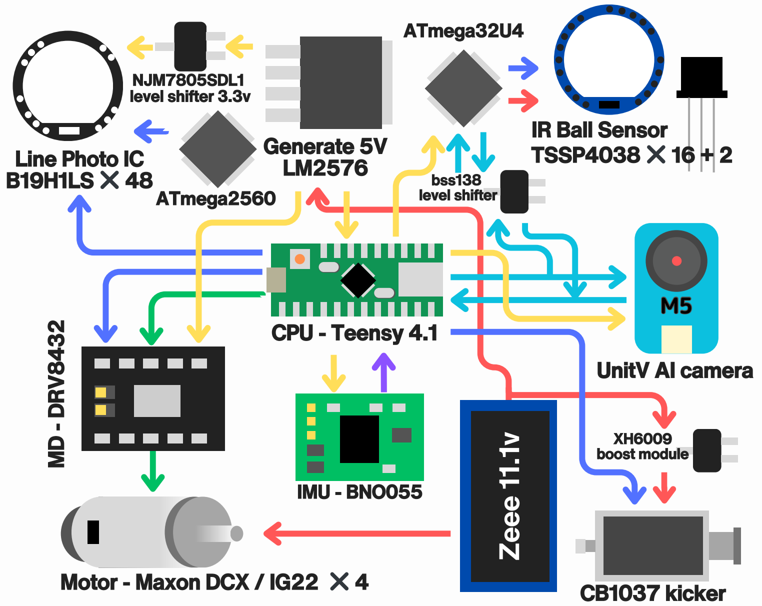
ABSTRACT

We are AIR, a Japanese high school team participating in Soccer LWL. We aim to achieve overwhelming victories with a robot that combines precise control and high mobility.

In the past, we developed algorithms that allowed the robot to advance toward the goal without relying on cameras, focusing on technologies that do not depend entirely on hardware limitations. However, this approach alone did not lead to success or good results in the Japan tournament.

Learning from this, we have inherited these innovative software ideas while improving the hardware's completeness, enabling us to build a robot that can compete on the world stage. To increase shooting success even from difficult angles or when defenders block the goal, we implemented a camera-based obstacle avoidance algorithm. We also optimized the robot's movement with custom-designed omni-wheels and a compact frame structure. To improve maintenance efficiency, we adopted detachable side panels and a modular internal layout, allowing quick part replacement. Furthermore, we improved the line detection system by minimizing both physical and processing delays through a combination of vector-based control and sensor configuration. Our electronics were modularized by separating functional blocks, and we focused on stabilizing the power supply to enhance in-game reliability. This poster summarizes our improvements.

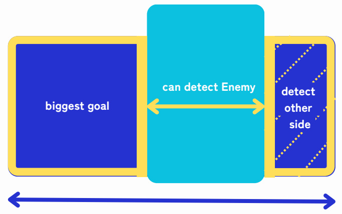
ROBOT STRUCTURE



SHOOTING ACCURACY AND STABILIZATION

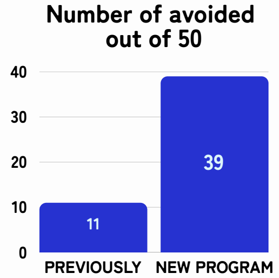
As strong defenses have become more common, the key to winning in this environment is to improve shooting accuracy and make it more consistent. Therefore, we created an avoidance program using a camera.

1. Detect the largest block with the target color in the image
2. From the top third row, scan pixels left and right to check the target color
3. If the target color appears outside the largest block, detect its width as a goal
4. Measure the widths on the left and right sides, and shoot toward the wider side



When the program was installed on the actual robot, it occasionally failed to accurately predict the opponent's movements. One cause of this was that the wide shooting area fluctuated and wobbled left and right. This issue was resolved by fixing the direction of movement immediately upon detecting the ball, which greatly improved stability. However, continuously advancing toward the initially wider area sometimes caused that area to narrow. To address this, we developed a feint-based strategy. The field was divided into two regions, and initially, the robot aimed to shoot toward the fixed wider area. If that side became blocked, it would switch to the opposite side, thereby luring opponents and creating an open space for a successful shot.

To support this strategy, we adopted a ball detection sensor as well as a high-performance motor and driver to achieve high precision and stable operation. Here is the QR code from last year's video of the world's number one team, Edge, evading the goalkeeper.

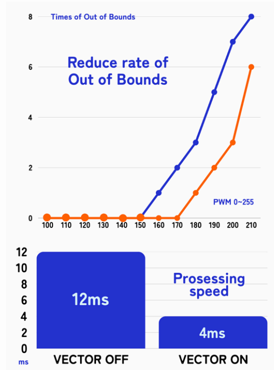


LINE DETECTION SPEED

In our previous setup, we used 24 phototransistors arranged in a circle and read their values using analog input. This method caused delays of several milliseconds due to the slow processing of analog signals. To solve this, we used LM393 comparators to convert the analog signals into digital, reducing the processing time. We also added a sub-microcontroller dedicated to handling the line sensor input. These changes brought the detection speed down to within 2 ms, allowing for much faster reactions. Additionally, we placed sensors not only in the center but also around the outer perimeter of the robot. This improved line detection coverage and made it easier to recognize opponents near the edges of the field. We increased the speed by 10 and examined the number of lineouts, and obtained the results shown on the right, demonstrating an improvement in performance.

For control, we implemented vector control optimized for circular sensor input. Unlike basic threshold-based methods, this approach calculates the ideal movement direction based on the distribution of sensor signals. As a result, the response time from detection to action was reduced to 12ms to 4ms.

Finally, we optimized the mechanical setup by using lightweight and responsive Maxon DCX motors, and by reducing the wheel diameter. These adjustments improved the robot's performance on slopes and significantly increased its tracking accuracy, especially during high-speed maneuvers.

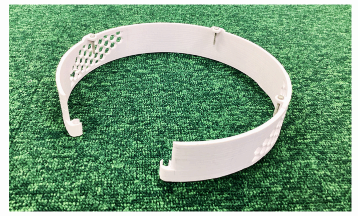


PRODUCTION & DESIGN

— DETACHABLE SIDE PANELS

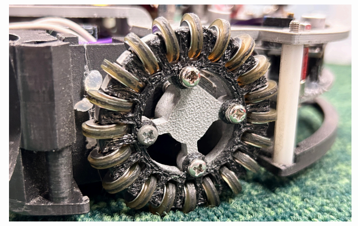
We created all of the side panels of the robot using 3D printing. Instead of enclosing the wheel in this panel and fastening screws to multiple points on the frame, we used a hook-shaped structure that significantly reduced the time required for attachment and detachment.

Additionally, this structure helps protect the wheels, which often get damaged during matches, and also reduces contact with opposing robots.



— ADVANCED OMNI WHEEL

In this year's world competition, we used Maxon DCX16 motors. DCX motors are known for their extremely fast response and high rotational speed. We used urethane with high grip and excellent abrasion resistance, and designed the side wheels with parallel pins inserted in their centers to improve rotational force. Additionally, by incorporating a frame made from a circuit board in part of the tires, we achieved a more durable and damage-resistant structure.



— IMPROVED MAINTAINABILITY

The line sensors are designed to be removable from the bottom. We also designed the structure so that the motors can be detached without disassembling the entire robot, improving maintainability. To keep the wiring clean and organized, we used FFC cables.

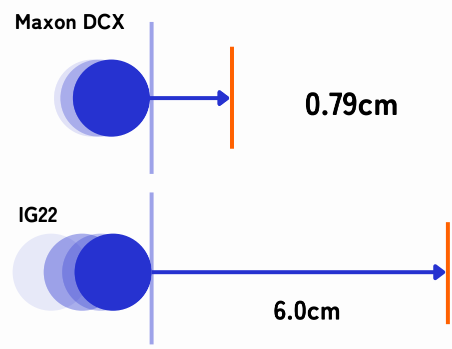
Thanks to 3D printing, we were able to easily implement these flexible and modular design features.

OPTIMIZING MOTOR & CONFIRM INPROVING

Selecting the optimal motor driver is key to maximizing motor performance. We firstly upgraded from the DRV8774 to the higher-spec DRV8432, which offers higher frequency and current capacity, resulting in better speed, torque, and control.

Using the DRV8432, we compared two motors: the IG22 and Maxon DCX. By attaching white tape to the shafts and recording at 120fps slow motion, we measured rotation time and responsiveness. IG22 took 11 frames per rotation and 6 frames to reverse direction, while Maxon DCX completed a rotation in 7 frames and responded almost immediately.

When calculating the distance traveled during response delay without considering friction, the Maxon DCX moved approximately 0.79 cm assuming a 0.5 frame delay, while the IG22 moved approximately 6.05 cm during a 6 frame delay. These results highlight the superior responsiveness of the Maxon DCX, which is essential for fast movements such as line detection in robotics competitions.

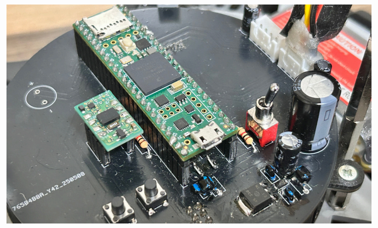


PARTS DIAGRAM

MAINFRAME

Parts : Teensy4.1, Switches, BNO055

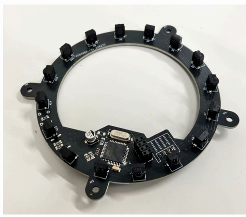
We chose the Teensy 4.1 as the main microcontroller. This is because it has very fast processing speed, simple debugging, and it is easy to use on our robot. Compared to the ATmega2560 that we used before, the Teensy 4.1 is about 38 times faster. This high speed helps improve the robot's response time a lot.



BALL SENSOR SUBSTRATE

Parts : ATmega32U4 / TSSP4038 / Type-C

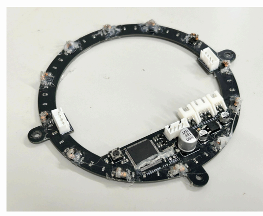
We use the ATmega32U4 as a sub microcontroller. In our robot, the ball sensor leads to the longest processing time and it needs many pins, so we decided to let the ATmega32U4 handle it. The ATmega32U4 was chosen because there are many articles about it on Japanese websites, making it easy to build on our robot. It communicates with the main microcontroller through UART.



LINE SENSOR SUBSTRATE

Parts : ATmega2560 / B19H1LS / LM393

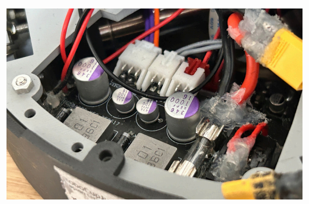
We use the ATmega2560 as a sub microcontroller for the line sensor. Since the line sensor sends a digital signal using comparators, it needs around 30 digital pins. That's why we chose this microcontroller, which has many digital pins. Also we used a CH340 module to connect the ATmega2560 to USB. It communicates with the main microcontroller through UART.



POWER SUPPLY INTEGRATED BOARD

Parts : Fuse 10A / LM2576-5v / LM2576-3.3v / Switch

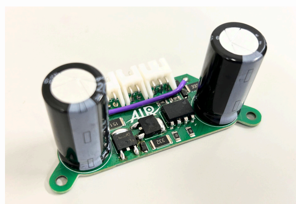
This is the power distribution board. It steps down the LiPo battery voltage to 5V using a DC-DC converter and supplies stable power to each board. To protect the circuit from overcurrent, a fuse rated for up to 10A is installed. Previously used DC-DC converters had the potential to exceed voltage ratings due to ripple or surge. To address this, we selected components with higher voltage tolerance to ensure safer and more reliable operation.



KICKER SUBSTRATE

Parts : Pch MOSFET / AQW212 / 4400uF capacitor

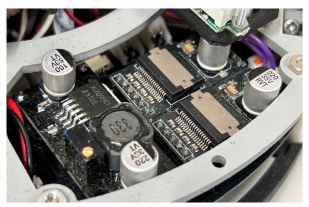
We bring 46V from the MD/Boost board and charge a 4400uF capacitor, then use a P-channel MOSFET to switch the solenoid on and off. To avoid directly connecting this switching signal to the microcontroller pin, which could be dangerous, we use the AQW212 to electrically isolate it. Since the Teensy 4.1 can only provide a small amount of current, this photo coupler is very useful.



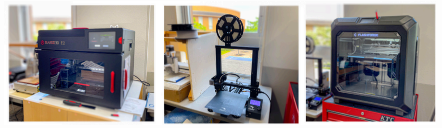
MD / BOOST INTEGRATED BOARD

Parts : DRV8432 Module / XL6009 boost Module

To save space, we combined the motor driver and boost converter into one board. For the motor driver, we chose the DRV8432, which supports the LAP method, can handle high frequencies up to 500kHz, and allows large current flow. The motor driver is modular, so it can be easily replaced if it breaks. We also modified the XL6009 boost module in order to output around 46V.



APPLICATIONS & TOOLS



We used the following applications and devices to build our robot. We developed the software using Visual Studio Code, designed the circuits with KiCad, created the mechanical parts with Fusion 360, and manufactured components using a 3D printer. By using these tools and equipment, we were able to carry out the development smoothly.

COST & SUPPORT

Our team contributes about $700 annually to cover basic activity costs like practice and equipment. However, this amount was not enough to fully fund the robot development and travel expenses to Brazil for the RCJ World Competition.To raise the necessary funds, we launched a crowdfunding campaign and actively promoted it through social media platforms. Many people kindly supported us after learning about our goals and passion for the competition.We also reached out to several companies via email to request sponsorship, clearly explaining the educational value of RCJ and the benefits of supporting our team's activities.

As a result, we raised about $5,000 through crowdfunding and secured sponsorship from the eight companies displayed at the top of this poster. In addition to financial support, we received parts and valuable technical advice, which greatly helped us prepare for the competition.

We are truly grateful for all the support we have received. Thank you very much!