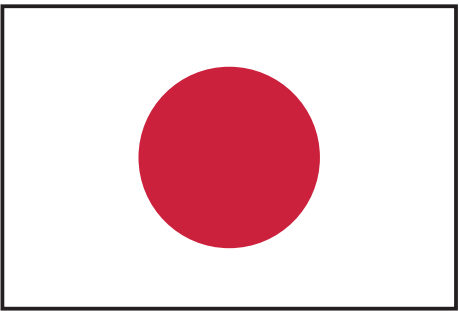


TANOROBO!

League:
Rescue Maze

Country:
Japan



About Us

We are "TANOROBO!", a Japanese robotics team, based at the National Institute of Technology, Gifu College. Our team was formed in October 2023.

Our team's name comes from the word "TANO-SHI-I" (たのしい , 楽しい) which means exciting in Japanese and our logo is designed based on a samurai and the Kanji character, " 楽 ".



MIYAZATO Takaki

Hardware and Software Integrator, Localization

"I integrated the software and the hardware developed by the other members. I also planned the development schedule."



SUMI Minagi

Navigation Algorithms, Embedded System

"I'm responsible for optimizing the robot's navigation. I built our robot's system using RTOS."



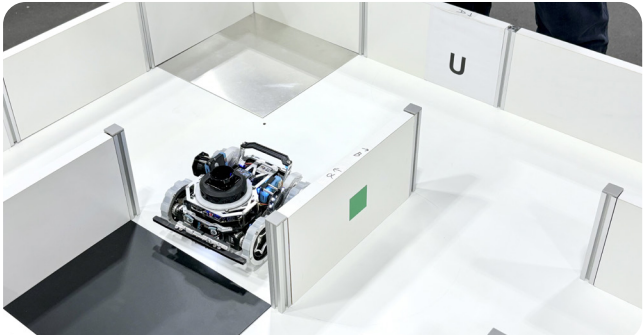
TAKAI Kyoshiro

Hardware Engineer, Fabricator of Robot Parts

"I developed our innovative wheels and fabricated all the mechanical parts of our robot."

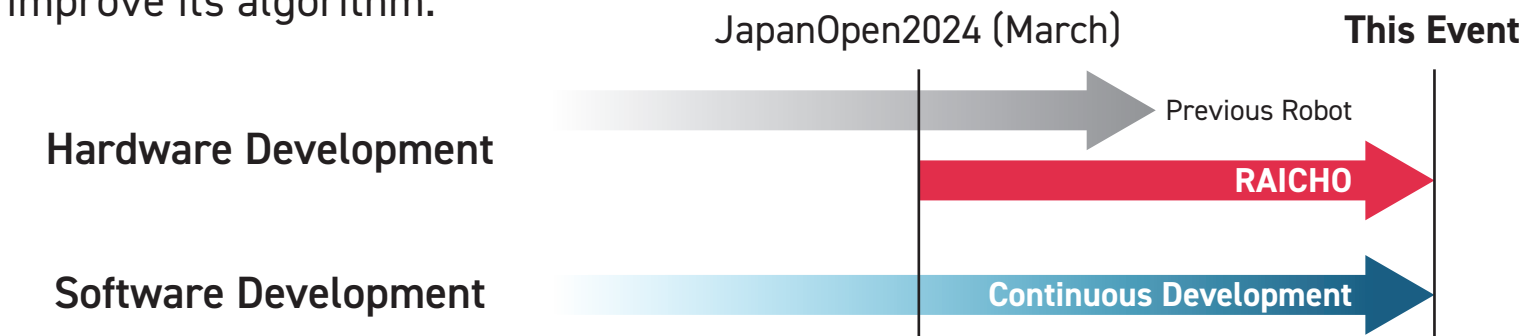
The Achievements at JapanOpen2024 were:

- 1st Place
- Best Poster Award



Development Plan

From the beginning, we aimed to build a rescue robot that is stable in all situations to secure successful missions. Our new robot, "RAICHO", is compatible with our previous one for JapanOpen2024, so we could continuously improve its algorithm.



For More Information

We use GitHub not only for efficient development but also for public access.

You can access all of our source code, schematics, PCB layouts, and mechanical CAD data by visiting <https://github.com/TanoRoboRCJ> or simply scanning this QR code.

We believe sharing discoveries and knowledge with each other is the most important thing at RoboCupJunior. That's why we have made our repositories available to the public.



Navigation Algorithms

Exploring:

We use the right-wall following algorithm for exploring. This algorithm repeats the following during the exploring:

1. Check the surrounding walls to know which tile it can move to.
2. Assign numbers to each surrounding tile counterclockwise starting from the right, excluding the tiles it cannot move to.
3. Retrieve the number of times it has reached each tile from its map data, then add five times that number to the variables created in Step 2.
4. Move towards the tile with the lowest number and update the map.

The map data is stored in a multi-dimensional array, enabling our robot to support three-dimensional fields as well.

Exiting:

Our robot starts exiting after six minutes from the start of the game. We use Dijkstra's algorithm and the map data created during the exploring to find the best path to return to the starting point.

Victim Detection

Our robot is equipped with two UnitV AI Cameras, one on each side. In addition, wide-angle lenses are mounted on them because the space of the camera and the wall is narrow.

For Colored Victims:

When the camera detects a colored victim using `find_blobs()` function, it immediately returns the detected color.

For Letter Victims:

We use a machine learning model trained with TensorFlow library and MobileNet V1. The training environment was set up on a Docker container.

We took photos of not only victims but also walls and obstacles to prevent misidentifying them. As a result, we did not experience any misidentification during JapanOpen2024. Furthermore, we used binary images to make identification easier.

After the model training, we validated it with Grad-CAM, a method for visualizing the regions of the image that the CNN model focuses on. The validation results are illustrated as the heatmaps shown on the right.

Localization

Localization is required to explore fields and receive an exit bonus. We combined three methods for accurate localization.

Wheel Odometry:

Our robot has rotary encoders mounted on its wheels, which can measure the distance the wheels have traveled. They allow the robot to calculate its position in the field.

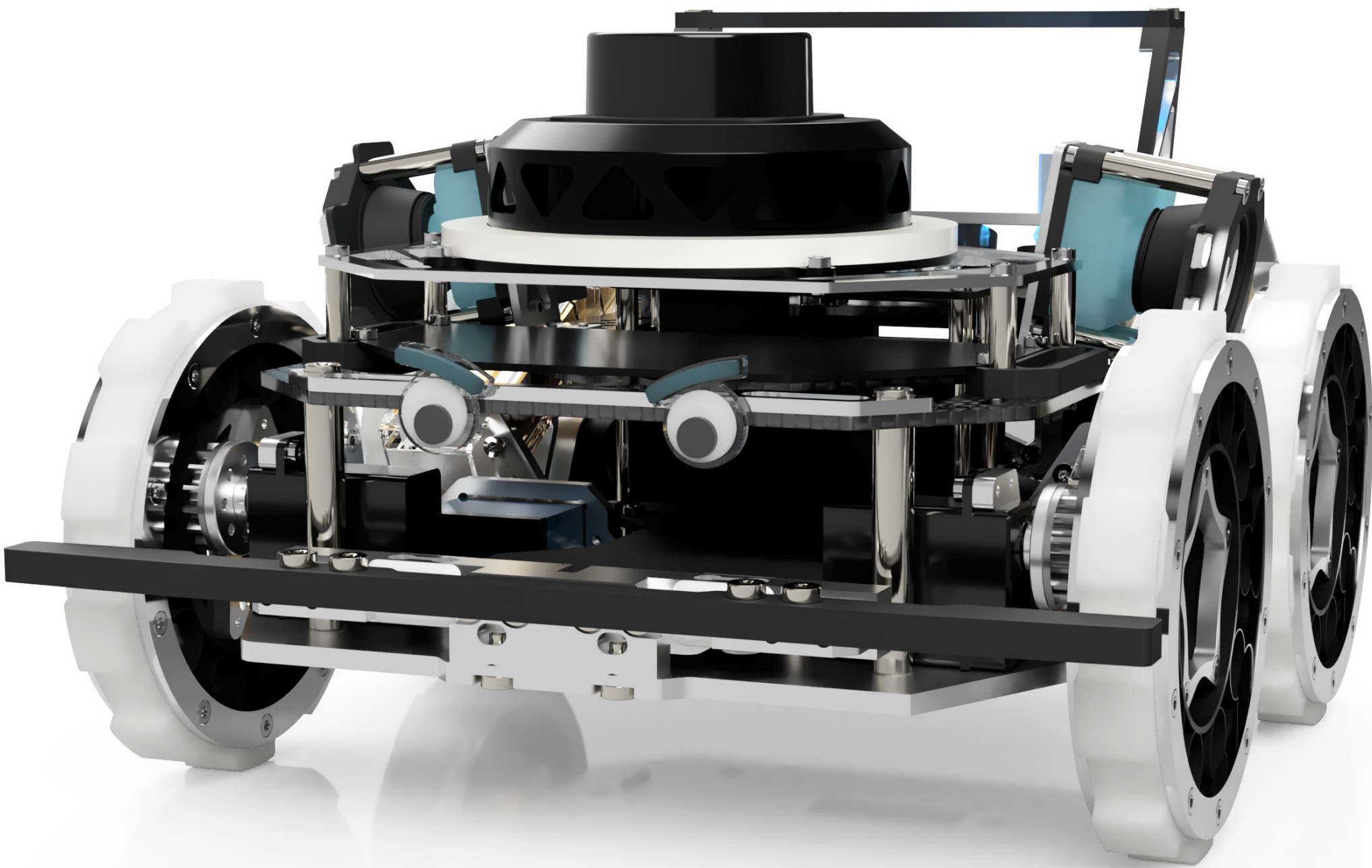
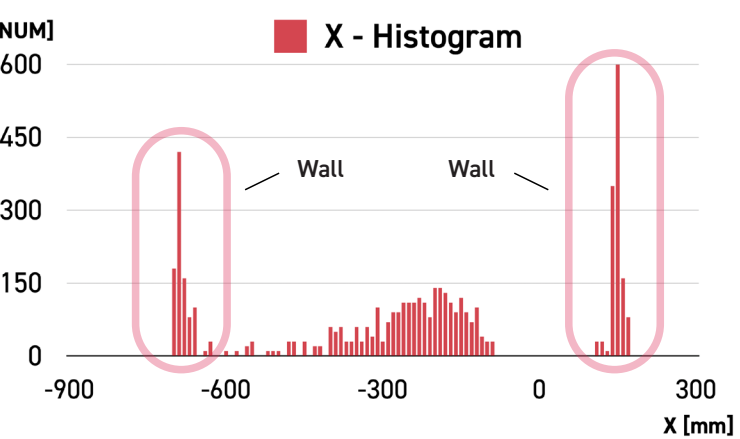
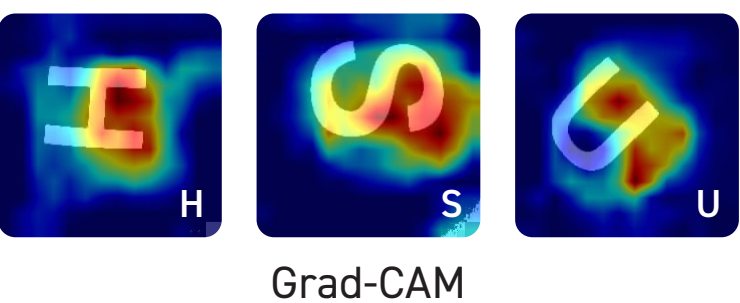
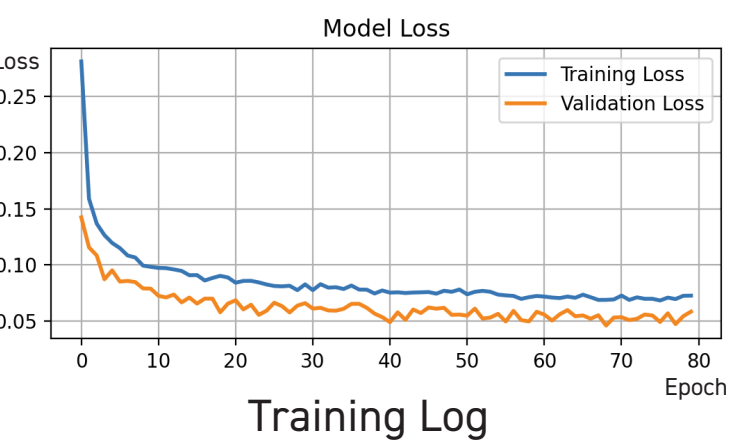
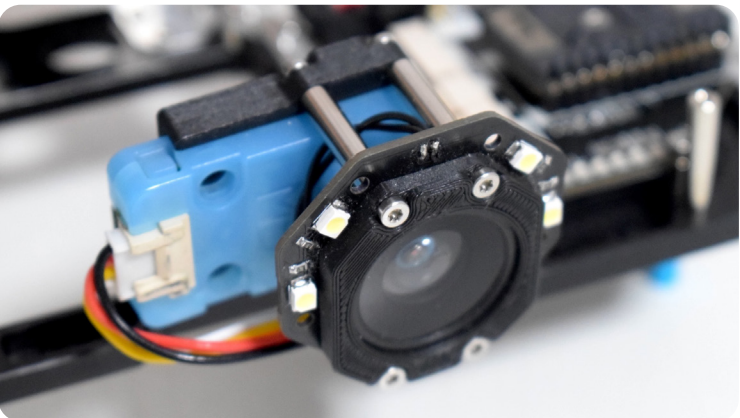
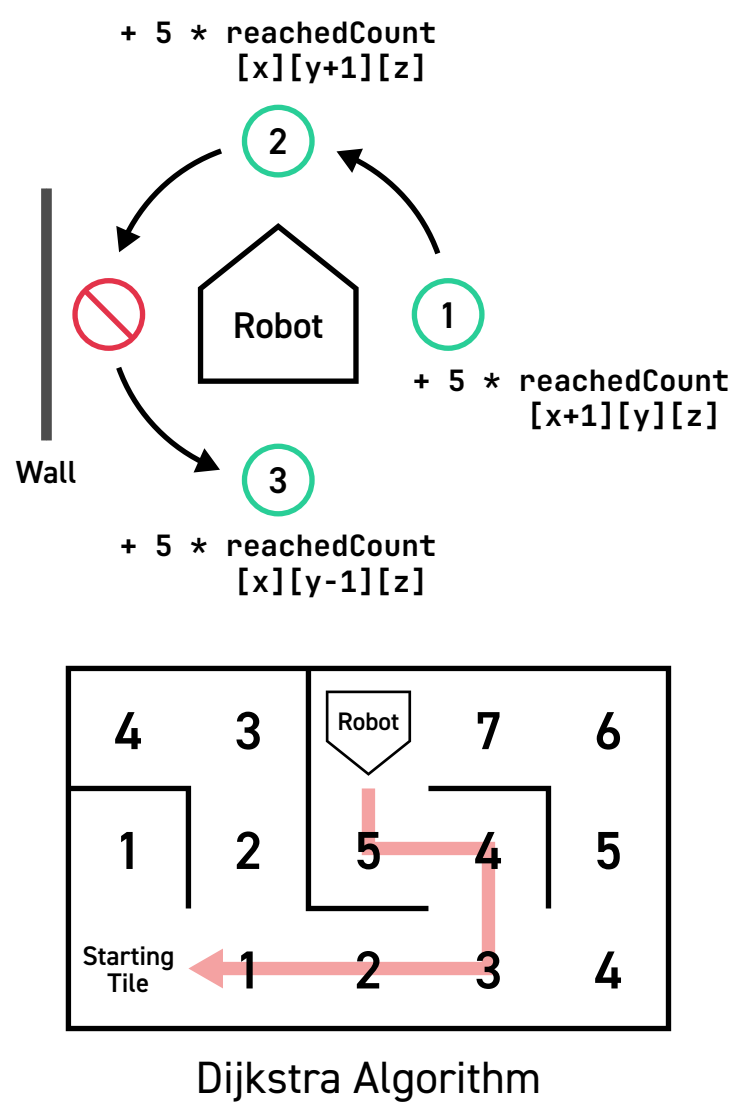
Position Correction with ToF Sensors:

We use ToF distance sensors to adjust the width of the robot and the wall. These sensors are arranged in a circular pattern so that the robot can estimate its position regardless of the direction it is facing

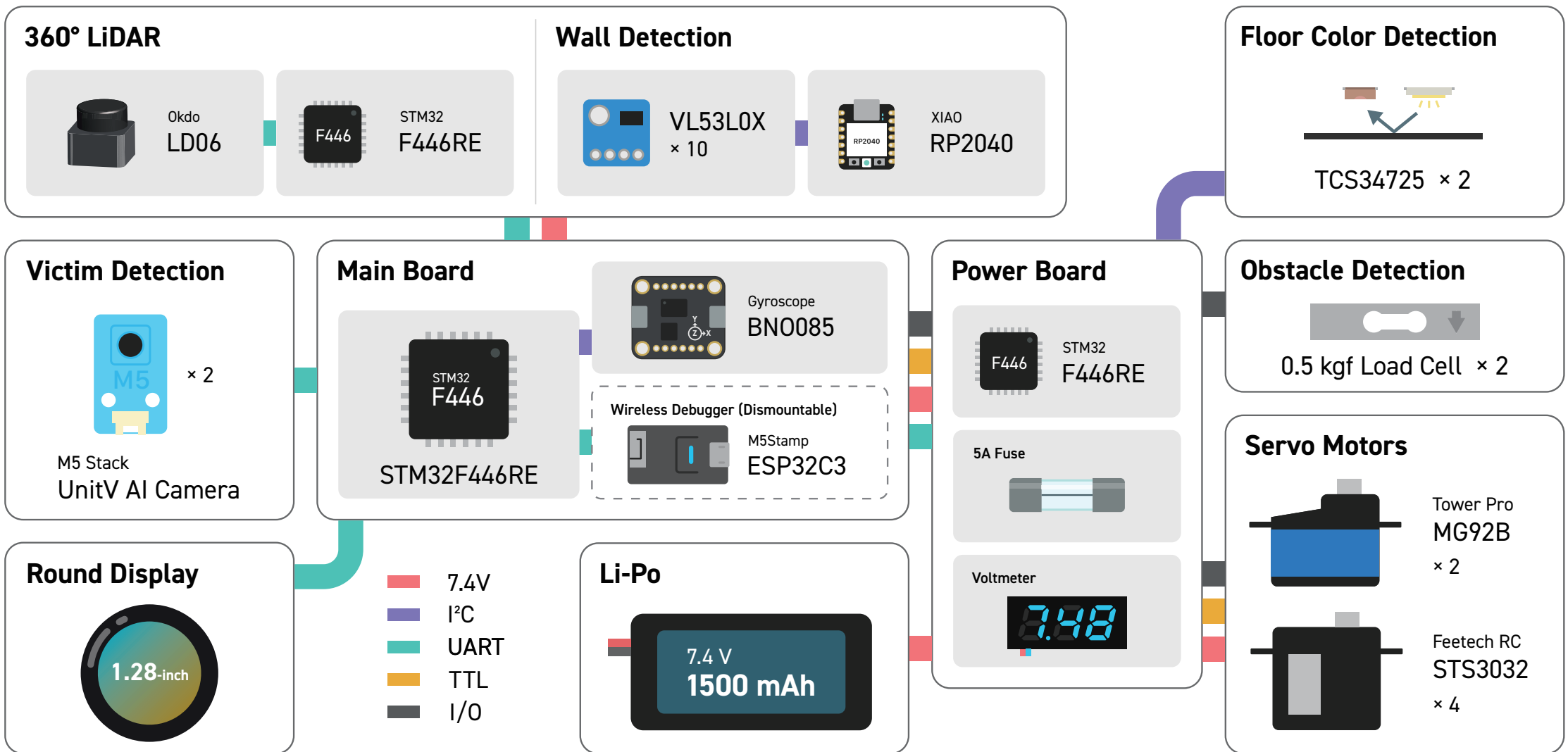
Real-Time LiDAR Scanning:

If the robot is positioned at the center of a tile, the point clouds will be concentrated on the positions of 15 cm, 45 cm, and 75 cm in length and width, as the walls in the Rescue Maze field are placed every 30 cm.

This method is not SLAM. Only histograms of the point clouds and covariances are needed, so an STM32 microcontroller is sufficient for the calculation and this algorithm runs in real time.



Electrical Components



Hardware

Our robot is designed for easy disassembly and can be separated by simply removing four screws at the corners. We used Autodesk Fusion for designing our robot. Our robot is so durable that we did not experience any LoPs during JapanOpen2024.

Motors and Wheels:

Four serial servo motors are used to drive the robot. These motors are so powerful that the robot can get over bumps and go up ramps easily.

For the wheels, we developed 3D-printed suspension integrated tires made of TPU. The outer is made of silicone and the inner is an O-shaped suspension.

Rescue Kit Distribution Mechanism:

Six rescue kits are stored on each side. As shown in the figure on the right, they are simply distributed by servo motors. This mechanism is fast enough to distribute one kit in 0.5 seconds.

Electric Components:

Our PCBs are designed with Autodesk Eagle. Our modular design of the PCBs facilitates efficient maintenance and reduces downtime and all the wires can be easily disconnected from connectors.

