



# ROBOCUPJUNIOR RESCUE MAZE 2024

## TEAM DESCRIPTION PAPER

*TANOROBO!*

### Abstract

We're TANOROBO! from Japan. Our new robot for this competition, named "RAICHO" is a stable rescue robot in all situations to secure successful missions. During JapanOpen2024, we did not experience any LoPs and misidentification.

This robot is entirely self-made. No kits, LEGO, or off-the-shelf microcontroller boards are not included in our robot. The aluminum frame of our robot was machined with CNC and it is so durable. We also designed all the electric components.

Our robot has a lot of innovative components. For localization and navigation, we use not only ToF sensors but also a LiDAR from this year, enabling the robot to be able to get an exit bonus more easily. Furthermore, the silicone grips of the wheels and the inner suspensions were also developed and fabricated by us.

For the victim detection, we trained a machine learning model with TensorFlow library and MobileNet V1. We also validated the model with Grad-CAM method after the training. The training environment that is on a Docker container is developed by us as well.

We use GitHub and Autodesk Fusion Cloud for efficient development and the public access. Anybody can get all our codes, PCB layouts, mechanical CAD data, technical articles, and other resources.

### 1. Introduction

#### a. Team

- MIYAZATO Takaki - Captain, Hardware and Software Integrator

MIYAZATO Takaki designed all of the electric components and the most of our robot and he is managing the development schedule. He also developed the environment for machine learning training. Additionally, he is responsible for ensuring the integration of hardware and software, optimizing performance, and troubleshooting technical issues. His role includes coordinating with team members to align project goals and timelines.

- SUMI Minagi – Software Developer

SUMI Minagi coded the algorithm for the robot to explore the field. He developed the right-wall follower algorithm and the user interface of our robot. Furthermore, he built the RTOS embedded system and implemented real-time task management. He is also responsible for optimizing the robot's navigation and exiting.

- TAKAI Kyoshiro – Hardware Engineer, Fabricator of Robot Parts  
TAKAI Kyoshiro fabricated all the mechanical parts of our robot. He also developed our innovative wheel, suspension integrated tire, enabling the robot to avoid getting stuck over any bumps. It is thanks to him that the team did not receive any LoP penalties during JapanOpen2024.

## 2. Project Planning

### a. Overall Project Plan

We think the most important thing of rescue robots is working stably because we believe rescue robots should not be rescued. To avoid getting LoPs, the robot must be so durable, and its software must have no bugs.

We couldn't earn the exiting bonus in RoboCup2023, so we wanted to earn it at this competition. At JapanOpen2024, we could earn the exit bonus in every game. Fields of international competition are larger than JapanOpen's, so we thought we need to improve its moves faster.

- **Milestones and Project Timeline:**

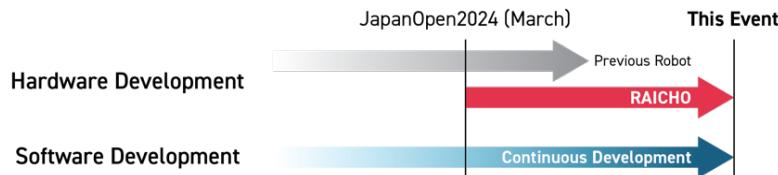
- **LiDAR-based Localization:**
  - **Description:** Developed a self-positioning algorithm using LiDAR data.
  - **Team Member:** Miyazato
  - **Deadline:** October 24
- **Suspension Integrated Tire Prototype:**
  - **Description:** Created airless tires using TPU and PLA for improved cushioning.
  - **Team Member:** Takai
  - **Deadline:** November 21
- **Right-Hand Rule Algorithm:**
  - **Description:** Implemented an algorithm for continuous exploration even if localization drift occurs.
  - **Team Member:** Sumi
  - **Deadline:** January 19
- **Building Machine Learning Environment:**
  - **Description:** Enabled machine learning model training on Docker containers, enhancing versatility in data processing.
  - **Team Members:** Miyazato, Takai
  - **Deadline:** February 1
- **Improving Wheel Grip with Silicone Molding:**
  - **Description:** Enhanced grip capabilities of wheels using molded silicone.
  - **Team Members:** Takai
  - **Deadline:** February 14
- **Dijkstra Algorithm for Exiting:**
  - **Description:** Implemented Dijkstra's algorithm for efficient return path planning.
  - **Team Members:** Sumi
  - **Deadline:** March 15

- **Schedule/Plan Agreement Process:**

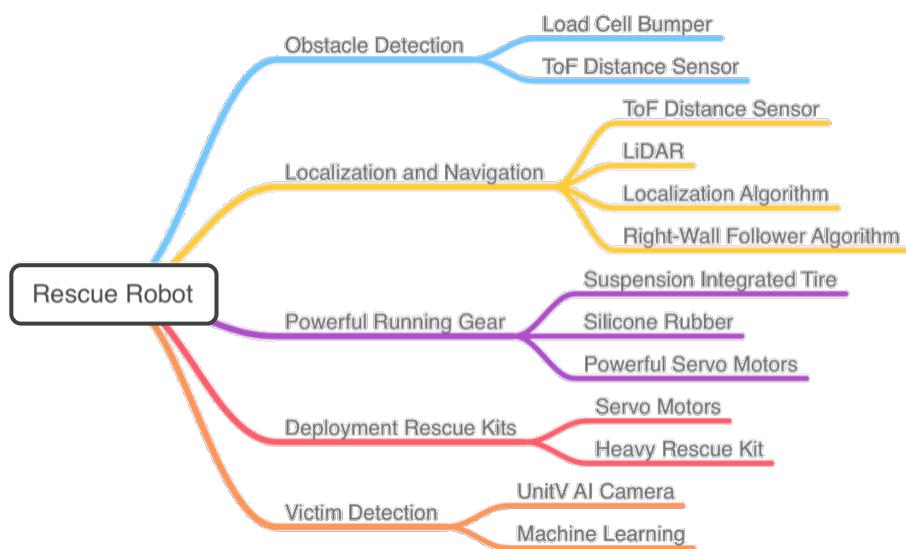
Our Captain decided on the development schedule described above, and we kept in close communication about the progress via Discord.

## b. Integration Plan

We have been ensuring compatibility between our current and previous robot models, so we could integrate hardware and software development concurrently. Most of our new robot, RAICHO is the same as the old version. The software developer didn't have to wait for the new robot to be completed because it could run with the same program.



We made large wheels to go up ramps and get over bumps easily but minimized the robot's size for smooth algorithm operation. For fast victim detection, we opted for a UnitV AI camera instead of Raspberry Pi, processing on a microcontroller. LiDAR-based localization requires heavy computation, so we added another microcontroller. All components communicate via UART for stability.



## 3. Hardware

### a. Mechanical Design and Manufacturing

- **Main Structure:**

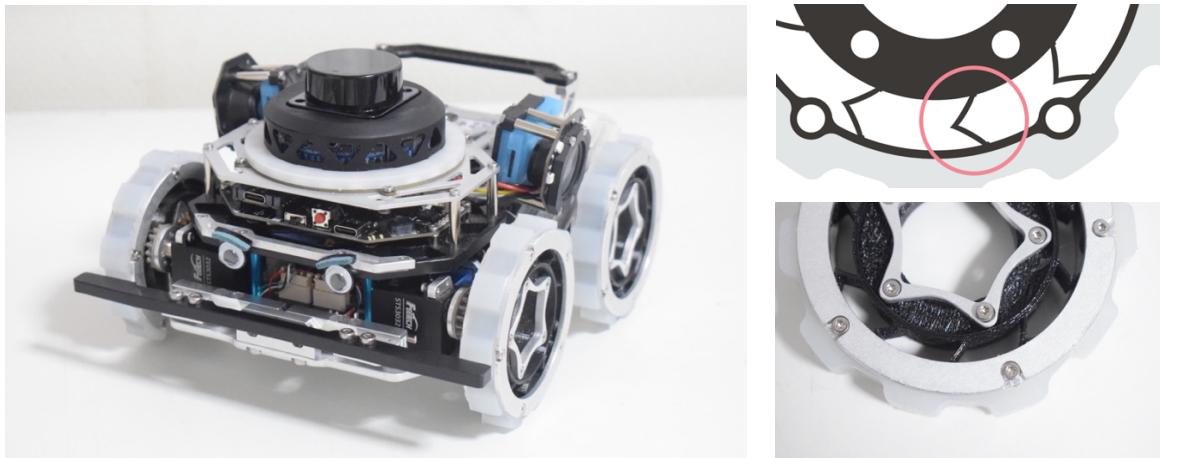
This is a photo of our robot. Our robot is now being improved, so it may be a little different from a robot competing in this competition.

We designed this robot with Autodesk Fusion. The size of the robot is 190 [mm] × 172 [mm]. To lower the robot's center of gravity, an aluminum frame is used for the lower structure and CFRP chassis is used for the upper structure.

- **Actuators and Power Train:**

We use four STS3032 serial servo motors to drive our robot. It is so powerful that the robot can go ramps up easily and get over bumps.

Starting this year, we have been developing innovative airless tires that incorporate a suspension system. First, the cushion part is created using a 3D printer, with TPU (thermoplastic polyurethane elastomer) as the material. This cushion part provides the tire's elasticity and absorbs shocks, stabilizing the robot. This allows the robot to move smoothly even on rough terrain. It is V-shaped suspension in theee pictures, but we're now developing O-shaped one for this competition.



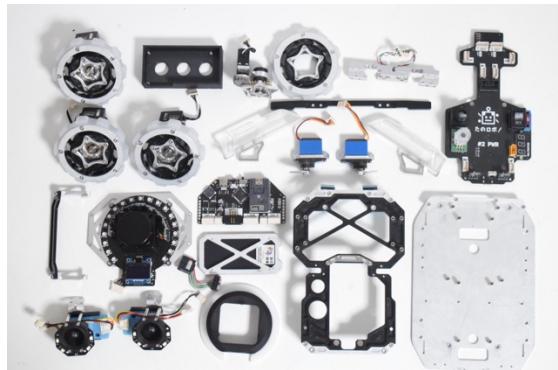
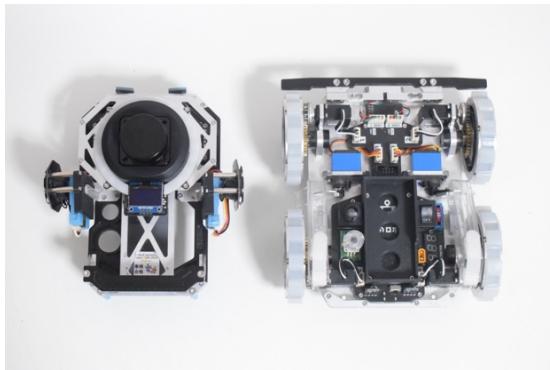
Next, the outer ring part is created by pouring silicone into a mold and hardening it. This mold was precisely cut using a CNC. The shape of the outer ring provides good grip on steps. This allows the robot to move smoothly even on terrain with height differences and complicated bumps.



Particularly noteworthy is the weight of the machine can be reduced that by incorporating suspension functionality into the tire. Since there is no need to install a separate suspension system, the overall structure of the robot is simplified, making design and manufacturing easier. In this way, airless tires are an innovative structure that significantly improves the robot's mobility and stability.

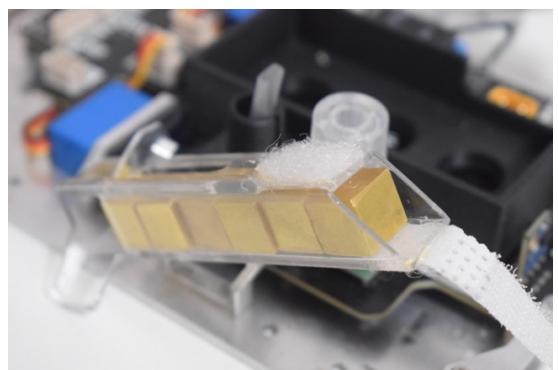
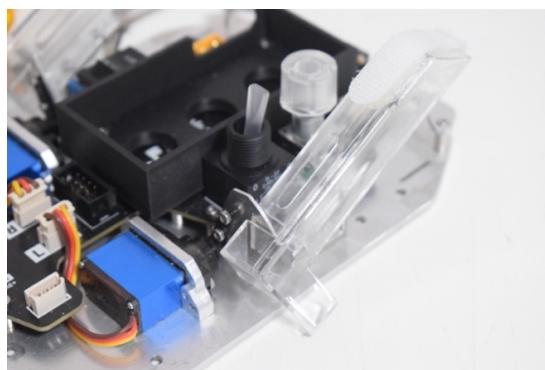
- **Other Subassemblies/modules:**

Our robot is designed with a thorough modular approach. This robot is designed to be easily separated into upper and lower by removing four screws on the bottom. The upper and lower wiring can be separated by a single 8-wires flat cable connector. The upper half has the main board, LiDAR and distance sensors, and victim detection cameras. The lower half has the drive wheels, obstacle detection bumper, rescue kit deployment mechanism, battery, and floor color sensors. The drive wheels modules are easily removable to accommodate wear and tear on the outer silicone, improving maintainability.



- **Rescue kit deployment mechanisms:**

Acrylic cases are attached to the left and right sides of the robot, each containing six rescue kits. The rescue kits are deployed by a servo motor. The rescue kit is a brass 11mm square cube. To prevent the robot from accidentally running over deployed rescue kits, they are dropped by throwing them outside the wheels.



- **Reliability Tests and quality assurance, Manufacturing:**

Burrs on the processed aluminum parts were manually removed to ensure a smooth surface finish, then anodizing was applied. The dimensions of each part were precisely checked using measuring instruments to ensure they matched the design drawings. A visual inspection was also conducted to check for surface scratches, chips, or other defects.

For the tire cushion, many prototypes with various materials and thicknesses were 3D-printed. As the result, TPU was chosen for its shock absorption and durability, with a thickness of 0.7mm. These prototype tires were tested on the robot under various conditions. For the outer ring's silicone molding, the silicone was treated in a vacuum container to remove air bubbles before pouring, ensuring a high-quality product without microbubbles. We also physically destroyed some of silicone parts to inspect their internal components.



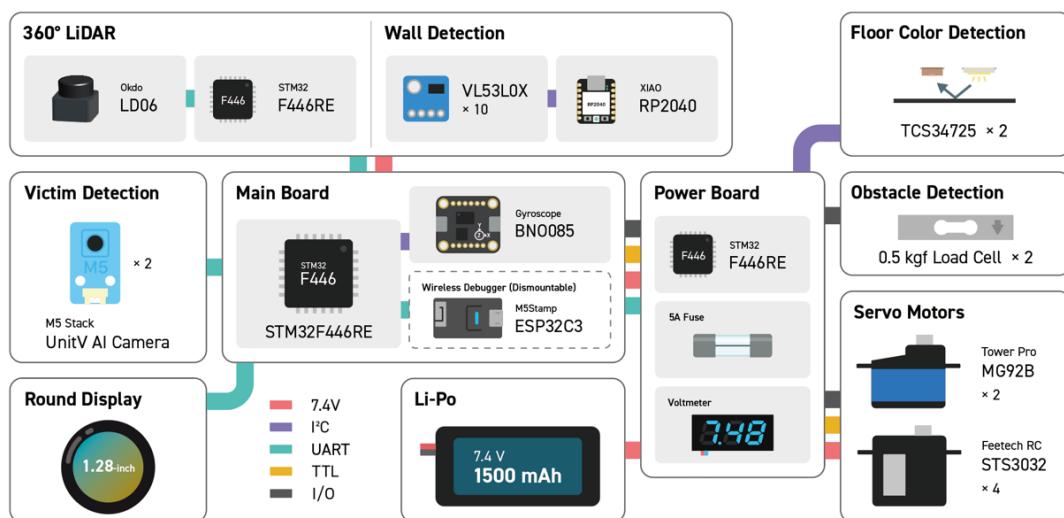
We conducted a test where we had a robot run on the field shown in this photo. There are many bumps in the field, even some not allowed by the rules. The maximum height of bump is 30 [cm]. There is also bumps on a ramps. Through repeated testing on a challenging field, we were able to refine the robot and make it more durable.



## b. Electronic Design and Manufacturing

- **Abstract:**

This is the diagram that explains the electrical components of our robot.



- **Distance Sensors:** VL53LOX  
10 distance sensors are placed on the top of the robot for detecting obstacles and walls.
- **LiDAR:** LD06  
A LiDAR is used to detect grids of tiles. The data from LiDAR is processed by the another microcontroller, because the LiDAR based localization algorithm requires heavy calculations.
- **Floor Color Sensor:** TCS34725  
They are equipped in front and center of the robot to detect the color of the floor. The sensor in front is for detecting black tiles, and another one is for detecting blue tiles.
- **Camera:** M5Stack UnitV AI Camera  
Cameras are equipped one on each side. In addition, wide-angle lenses are mounted on them because the space of the camera and the wall is narrow.
- **Gyroscope:** BNO085  
For checking the direction and accurate localization.

- **Main Controller:** STM32F446RE  
 For high-speed operation of the RTOS, we employed a high-speed microcontroller with an operating frequency of 180 [MHz]. This microcontroller has six UART communication ports, making the modules on this robot can communicate reliably.
  - **Power Subsystem:**  
 This robot uses a 1500mAh Li-Po battery. We are using both 3.3V and 5V for powering the sensors and controllers.
- **Sub-module Design and Reliability Tests:**  
 We designed all PCBs with Autodesk EAGLE. All the wires can be easily disconnected from connectors for easy repairing. The main board has a microcontroller, a gyroscope and a Bluetooth module. This Bluetooth module is used for debugging purposes and can be easily removed. The main board communicates with cameras, distance sensors, servo motors, and LiDAR via UART.
- For reliability testing, we conducted vibration tests. We also performed durability tests by dropping the entire robot to check that the circuits operate normally even in harsh environments. For the color sensor, we used an iPad display to verify the linearity of the color sensor values.

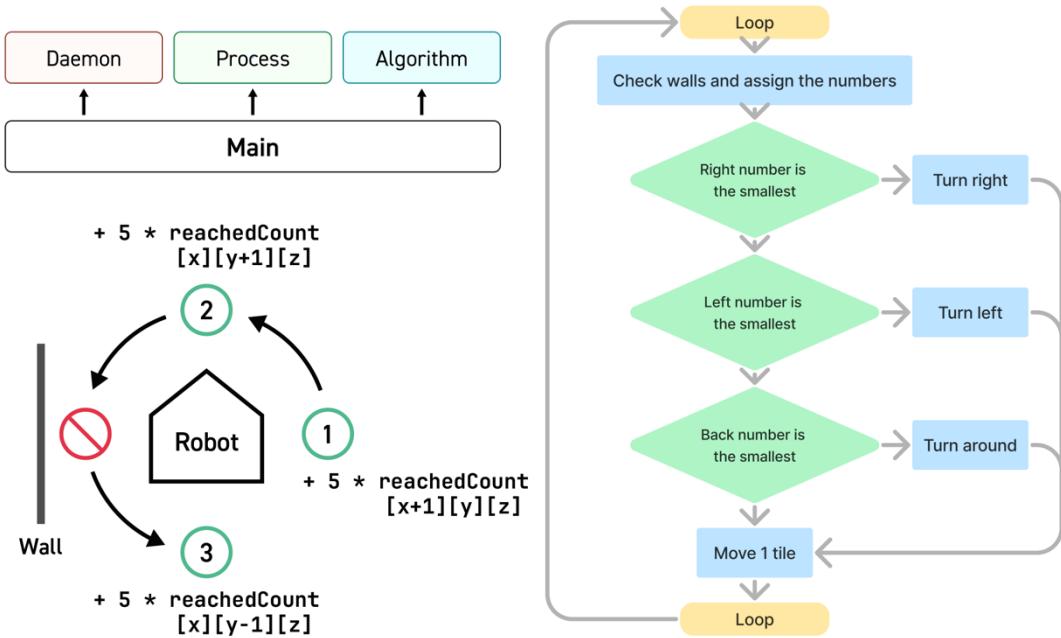


## 4. Software

### a. General software architecture

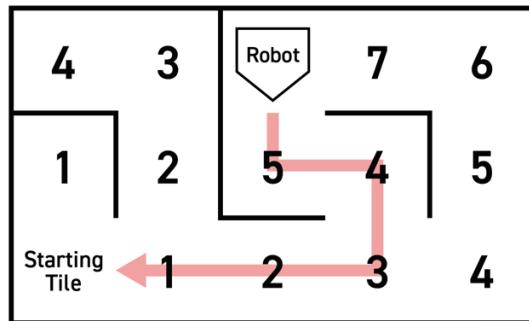
- **RTOS Embedded System:**  
 We utilize RTOS (Real-Time Operating System) embedded system to divide functions into separate tasks and process them concurrently. In a RTOS, each priority of task is clearly defined. The main program consists of four tasks:
  - **Main:** Responsible for starting, pausing, and resuming other tasks.
  - **Daemon:** Continuously updates values from sensors and actuators, such as distance sensors, color sensors, load cell bumpers, gyroscope, camera data, and servo motor rotations.
  - **Process:** Performs calculations based on the values obtained by Daemon.
  - **Algorithm:** Handle the specific algorithms for the robot's navigation and behavior.
- **Right-Wall Following Algorithm:**  
 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.



- **Dijkstra's Algorithm:**

Dijkstra's algorithm can find the shortest path between the current tile and the starting one. First, assign the number 0 to the starting tile. Then, assign the number 1 to the tiles adjacent to the starting tile. Next, assign the number 2 to the tiles adjacent to the tiles assigned number 1 in the previous step. Continue in this way to create a map of steps from the starting tile. This map helps to find the shortest path for exiting.



Our robot starts exiting after six minutes from the start of the game.

- **Victim Detection:**

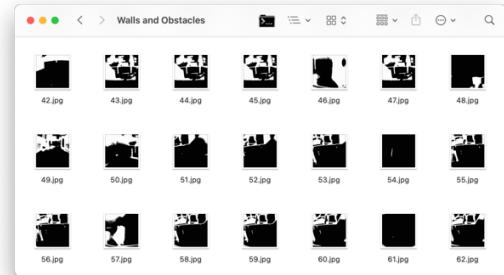
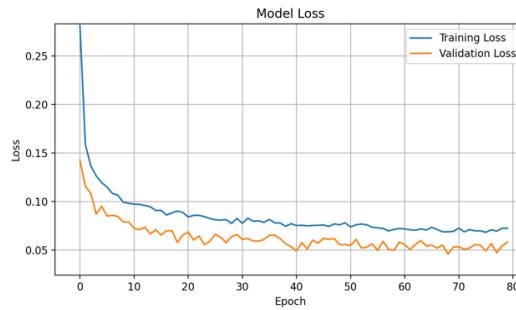
- **Colored Victims:**

We utilize the `find_blobs()` function to rapidly identify colored victims and immediately return the detected color to the main microcontroller.

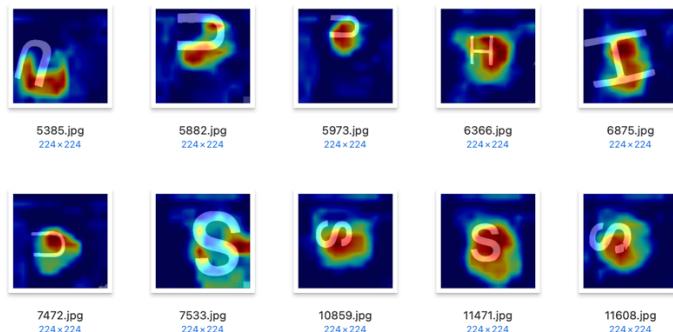
- **Letter Victims:**

Our approach involves a machine learning model trained with TensorFlow and MobileNet V1 in a Docker container environment. For rapid training, we use NVIDIA GPU and TensorFlow-GPU Python library. In addition to the images of the letter victims, we've included images of walls and obstacles in the training data to prevent any misidentifications. This approach

proved successful, with no misidentifications during JapanOpen2024. To further streamline the identification process, we employed binary images. This figure shows the training log.



For testing, we validated the data by using Grad-CAM method, which is visualize method to represent the regions of the image that the CNN model focuses on. This is the validation result. The result shows that the model correctly focuses on the letters in images.

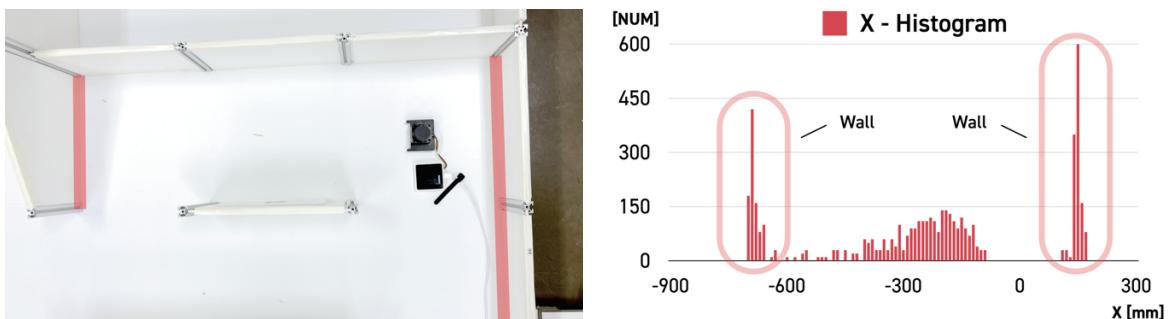


## b. Innovative solutions

- **Localization Using LiDAR:**

From this year, we use real-time lidar scanning method for localization, in addition to the wheel odometry method and position correction with ToF sensors. When the robot is positioned at the center of a tile, the LiDAR point cloud data will exhibit a distinct pattern: concentrations at distances of 15 [cm], 45 [cm], and 75 [cm] in both length and width directions. This is due to the Rescue Maze field's wall placement at 30 [cm] intervals.

Our approach uses this pattern to determine the robot's position relative to the walls. By analyzing histograms of the point cloud data and calculating covariances, we can efficiently and accurately estimate the robot's location. This method just compares to the sine wave with a wavelength of 30 [cm], so an STM32 microcontroller is sufficient for the calculation and this algorithm runs in real time. As a result, there is no longer any confusion between walls and obstacles.





## 5. Performance evaluation

- **High-Speed Running Tests:**

We did high-speed running tests by increasing the motor speed beyond normal operation. This was done to verify the correct functioning of our algorithms under demanding conditions. In these tests, we successfully explored and returned from a 6x6 field in 3 minutes and 2 seconds. We were able to identify the following issue:

- **Getting Stuck on Bumps:**

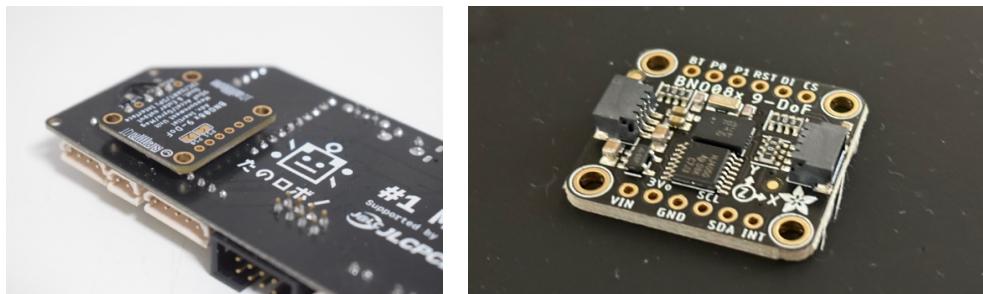
Previously, when the robot got stuck on a bump, we had to wait until it luckily passed through, which was a waste of time. We developed a bump avoidance program. The program works by comparing the desired angle the robot should face after rotating on the bump with the actual angle it is facing. If there is a discrepancy, the robot slightly backs up and rotates again. This method has significantly improved the robot's ability to clear bumps.

- **Long-Term Reliability Tests:**

Although the game time is 8 minutes, we conducted a long-term reliability test. We ran the robot continuously for 20 minutes. This was done to simulate the games of a larger field and to identify any potential bugs. As a result of this test, we were able to identify the following two issues:

- **Gyroscope Drift:**

We found that the gyroscope drifts over time when the robot is operated for extended periods. This can negatively impact accurate localization and mapping. We resolved this issue by replacing the BNO055 sensor with the BNO085, which offers improved stability and reduced drift over extended operation. We also found that accurate calibration was necessary.



- **Freezing Bug:**

We initially developed a program to store the order in which the robot arrived tiles, with a capacity of 200 entries array. However, during a long-term reliability test, we discovered that the array would overflow, causing the robot to freeze. To resolve this issue, we modified the array to represent the tiles in three dimensions (X, Y, and Z) for preventing freezing.

## 6. Conclusion

TANOROBO's new rescue robot, RAICHO, features a durable CNC-machined aluminum frame, custom electrical components, and innovative solutions like LiDAR-based localization and a TensorFlow-trained victim detection model. We are confident we can win this competition.

## Appendix

- Our Blog (written in Japanese): <https://tanorobo.shirokuma89.dev>
- Our Twitter: @TanoRoboRCJ
- Our GitHub Organization: <https://github.com/TanoRoboRCJ>