Engineering journal

Bio Pages

https://github.com/TanoRoboRCJ

This is our GitHub Organization page. We use Github for efficient development.

As it is important to share discoveries with each other, we are making all the codes and electrical circuits available to the public.

We don’t have to hold meetings because we are living in the same student dormitory. We can meet everyday and develop together.

YouTube Channel

https://www.youtube.com/channel/UC5DJXAC5c4HjL22OScChD-g

This is our YouTube channel. Some videos are uploaded and you can watch our games in JapanOpen2024.

Abstract

This is a journal of the development from October 2023 up to now. Specific development details are described below.

Our Respective Roles

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.

Project Plans

LiDAR-based Localization:

● Description: Developed a self-positioning algorithm using LiDAR data.
● Team Member: Miyazato
● Deadline: October 24

Suspention 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:

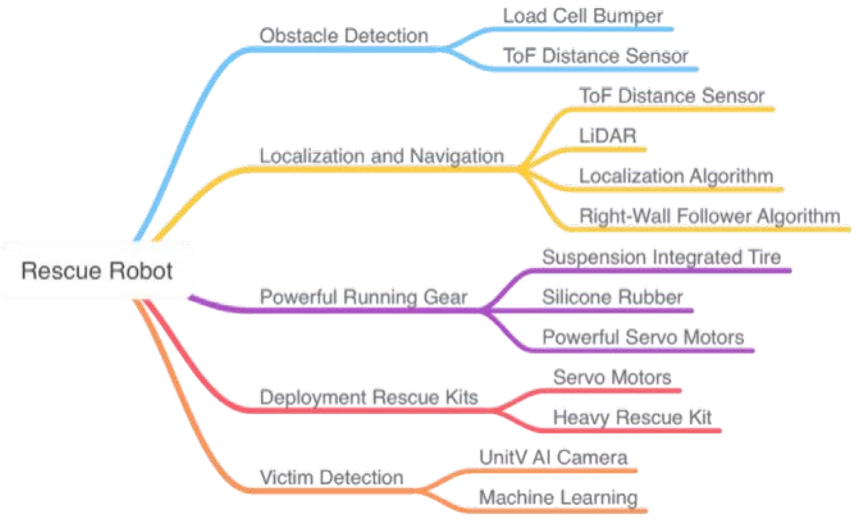
● Description: Enabled machine learning model training on Docker containers, enhancing versatility in data processing.
● Team Members: Miyazato, Takai
● Deadline: February 1

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



Progress Log

October

4th

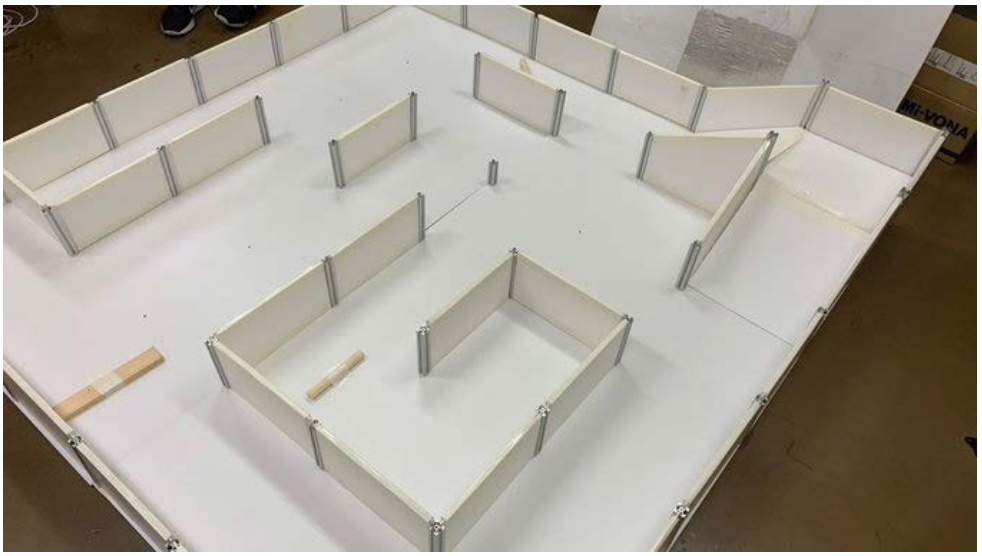
Our team, TANOROBO! (たのロボ！) was formed. We swore to win 1st place in **JapanOpen2024** and **RoboCup2024**.

We set up the following development rules:

- Clarify the development flow.
- Visualize the progress.
- Enjoy building robots!

5th

We made the maze like Japan Open and RoboCupJunior 2023's one. By using aluminum pipes, It is easy to build maze.



6th

We created a software repository on GitHub. And we decided to use RTOS.

What is RTOS?

Maintaining a real-time performance is critical, since victims and obstacles can appear at any time. Therefore, we use RTOS to define processing priorities and ensure that sensor reading intervals are constant.

RTOS allows us to easily start and pause tasks, which are a series of operations. Multiple tasks can be executed seemingly and simultaneously, eliminating the need to assign the current state to complex variables one by one.

→ This made it easy to turn on/off functions that are used or not used during the program development, thereby increasing efficiency.

Good Point	caution⚠
delay function	need accurate debugging
several tasks	disorderly processing
start / stop of tasks easily	void loop() ← don't work
make apps about20	Consumes a lot of memory

9th

We tried ROS and Rviz to evaluate a LiDAR. We succeeded getting point clouds, but still we didn't have enough knowledge of Linux.

14th

We named our robot ONAMAZU, and began to develop its hardware.

20th

We filled our inventory on our Notion database.

NA-105	MIL 8pin リリーフ	BOX-E 電子部品	4	コネクタ
NA-104	Mini Spoxハウジング 2pin	BOX-E 電子部品	8	コネクタ
NA-103	Mini Spoxハウジング 4pin	BOX-E 電子部品	8	コネクタ
NA-102	Mini Spoxポスト 4pin 直	BOX-E 電子部品	8	コネクタ
NA-101	Mini Spoxポスト 3pin 直	BOX-E 電子部品	10	コネクタ
NA-100	Mini Spoxポスト 2pin L	BOX-E 電子部品	6	コネクタ
NA-99	Mini Spoxポスト 4pin L	BOX-E 電子部品	6	コネクタ
NA-98	MG92B サーボ	BOX-E 電子部品	4	アクチュエータ

23rd

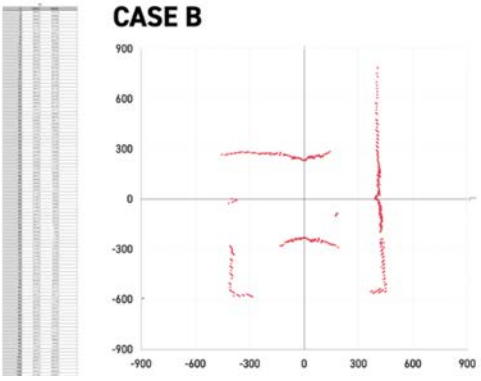
We began to develop the exploring algorithm with our previous robot (from RoboCup 2023) because the new robot was still being built.

We completed the localization program with absolute direction. We could create a map based on the cardinal directions (north, south, east, and west).

We also started using a wireless debugger, eliminating cable interference with sensors. We now receive information and status updates through the "LightBlue" smartphone app.

24th

We tried localization with LiDAR and succeeded to get point clouds. Its result and the calculation method are described below:



First, calculate averages and find the sum of the deviations.

$$\mu_d = \frac{1}{n}\sum_{k=1}^{n} data_k$$
$$\mu_r = \frac{1}{n}\sum_{k=1}^{n} ref_k$$

Then, we can find the covariance value.

$$S_{dr} = \sum_{k=1}^{n}(data_k - \mu_d)(ref_k - \mu_r)$$

$$\begin{aligned} Cov. &= \frac{S_{dr}}{n} \\ &= \frac{1}{n}\sum_{k=1}^{n}(data_k - \mu_d)(ref_k - \mu_r) \end{aligned}$$

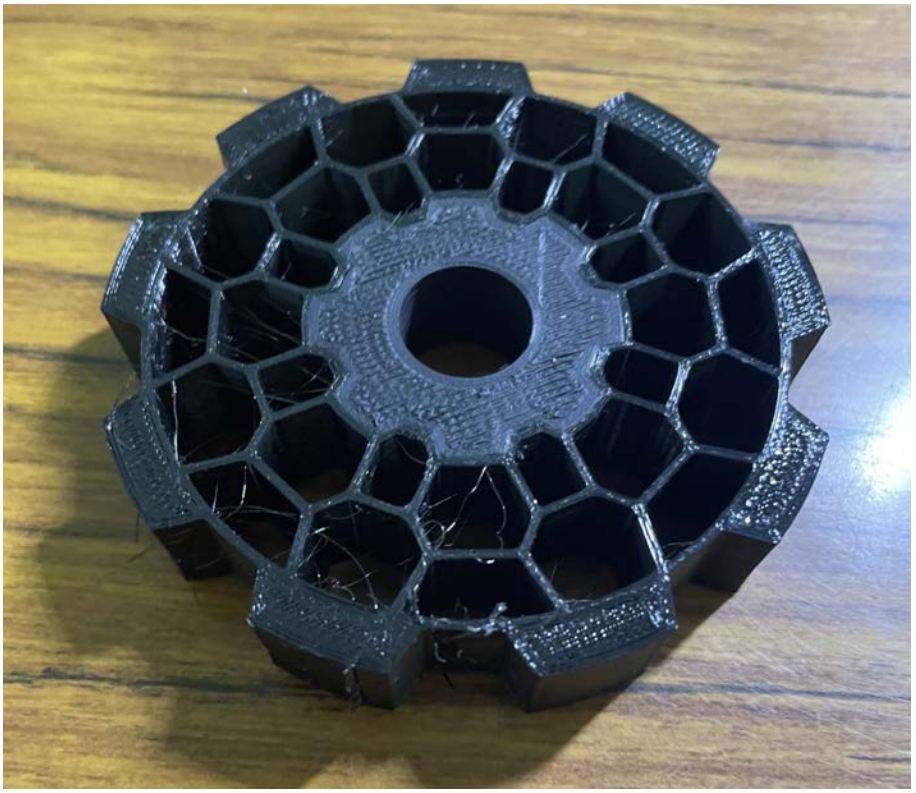
26th

We made a 3D-printed PLA tire with an inner suspension, but it was too rigid to deform properly. We decided to try printing a new one using TPU filament.

November

21st

We developed a cushioned tire composed of TPU and PLA. By adopting a honeycomb structure, the tire is both sturdy and flexible.



30th

We began developing a camera system for victim detection and addressed the following items:

- Fixed development environment-specific bugs (MaixPy)
- Adjusted white balance and exposure settings
- Accelerated processing by building custom firmware (achieving 33fps)

2024

January

12th

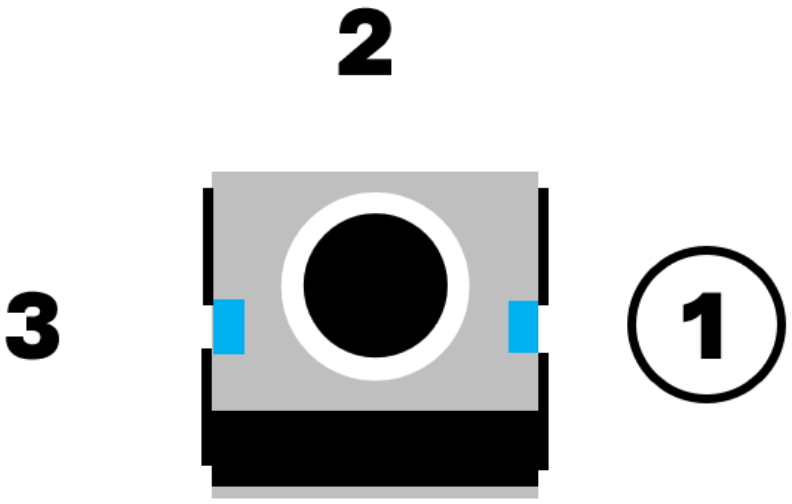
The second cushioned tire was developed, featuring L-shaped spokes that flex to provide suspension.

19th

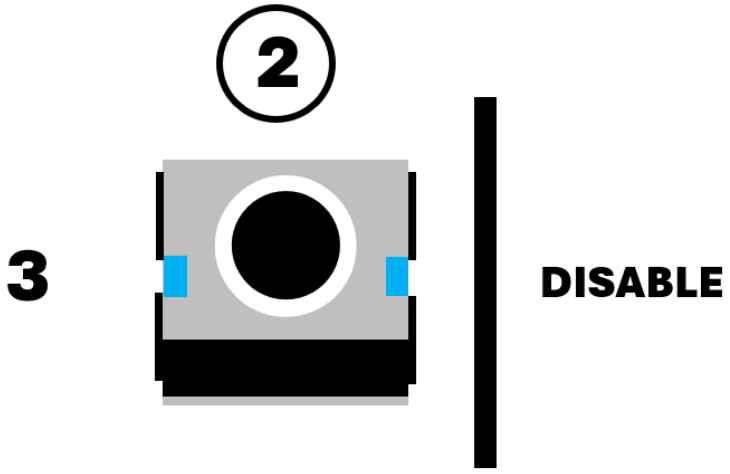
We decided to use the right-hand rule for exploration. While it can go to floating tiles, it may not be the most efficient exploration method.

The algorithm we developed is not heavily reliant on mapping. Even if the robot incorrectly records map information, it can continue exploring.

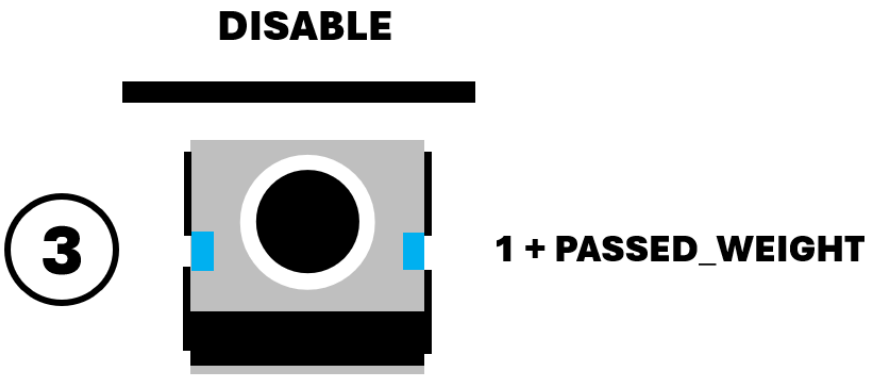
The abstract of the algorithm is described below:



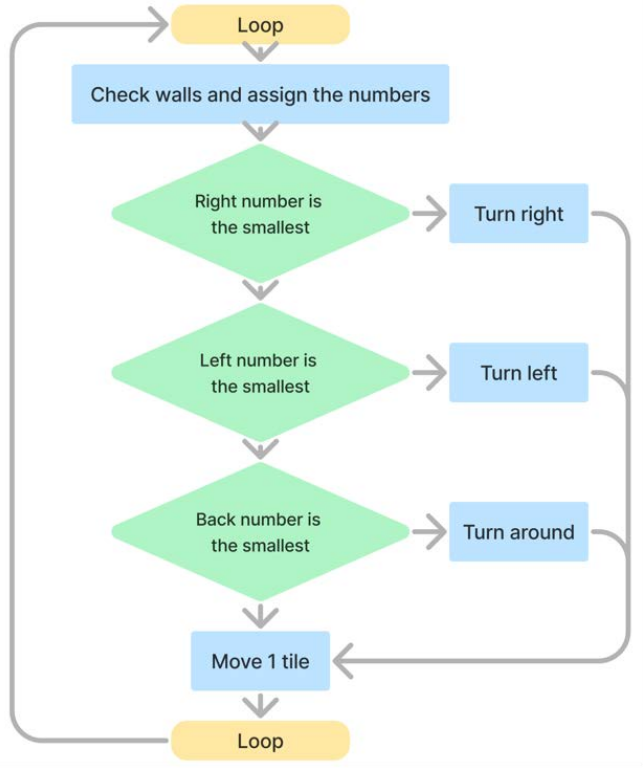
First, check the surrounding walls, on the right, left, and in front of the robot. If there are no walls, assign a weight to each direction as shown in the figure. The robot then moves in the direction with the lowest weight.



If there is a wall, assign weights as shown in the figure. “DISABLE” represents large value. For example, 1000.



If the robot has already passed the location, add “PASSED_WEIGHT” to the direction’s weight. The PASSED_WEIGHT value is determined by the number of times the robot reaches a tile.

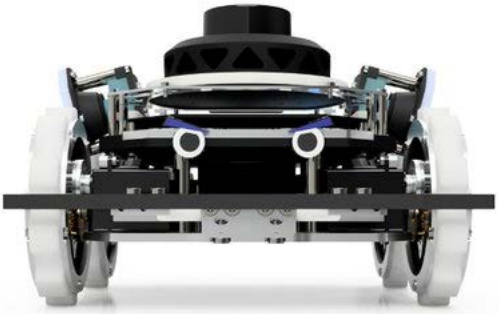


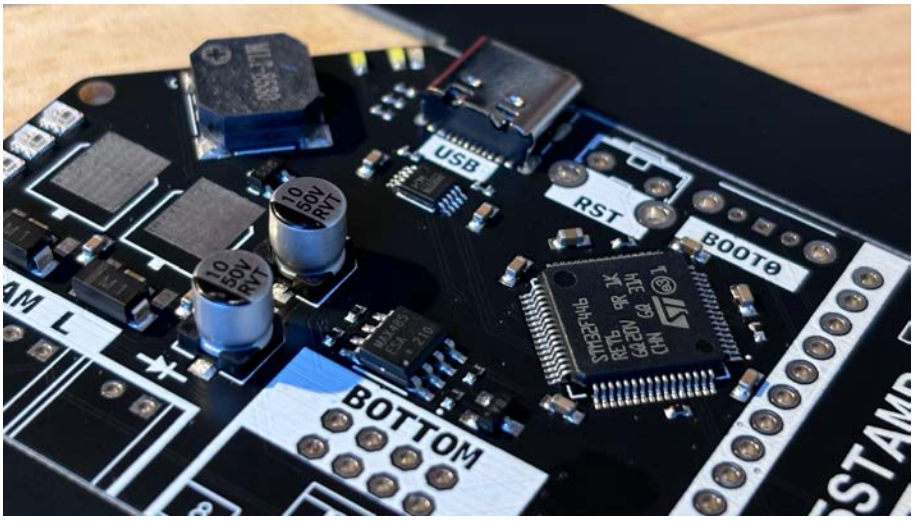
25th

We performed code refactoring, formatting not only the source code itself but also reorganizing the directory structure for improved maintainability.

29th

We finished designing our robot for the JapanOpen and began cutting parts and ordering PCBs. We named our new robot ONAMAZU (大鯰).





February

1st

We succeeded in implementing a exiting function using the right-hand rule algorithm, but it proved to be inefficient.

5th

We were able to perform machine learning on a Docker container. This allowed us to freely adjust the number of training images and epochs, which is expected to improve the model's accuracy.

7th

We tried to cut silicone with CNC for wheels, but it was not good. We determined to try silicone molding next week.

10th

We went to Ise Jingu shrine and wished for the fulfillment of our heart's desires.

14th

We experimented with silicone molding to create tire grips, resulting in improved grip compared to previous versions. We began 3D printing wheel cushion parts, producing a total of eight, including spares. We assembled the cushion parts and paddle wheel parts to complete the wheels.

We also physically destroyed some of silicone parts to inspect their internal components.

March

4th

We debugged the software and began transitioning to the new robot (ONAMAZU).

5日

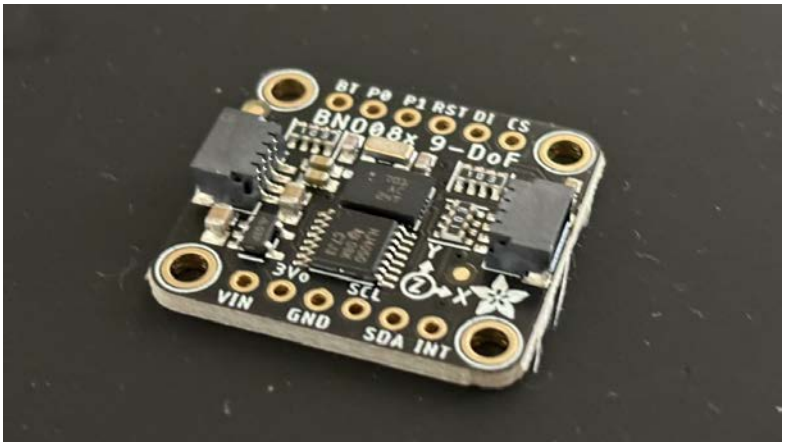
We set up several bumps and obstacles, but the robot traversed them easily. This confirmed the robustness of the robot's suspension system.

8th

Long-Term Reliability Tests:

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. Causing it to malfunction and leading to inaccurate angle and coordinate readings. We promptly ordered a different type of one to fix the issue.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.

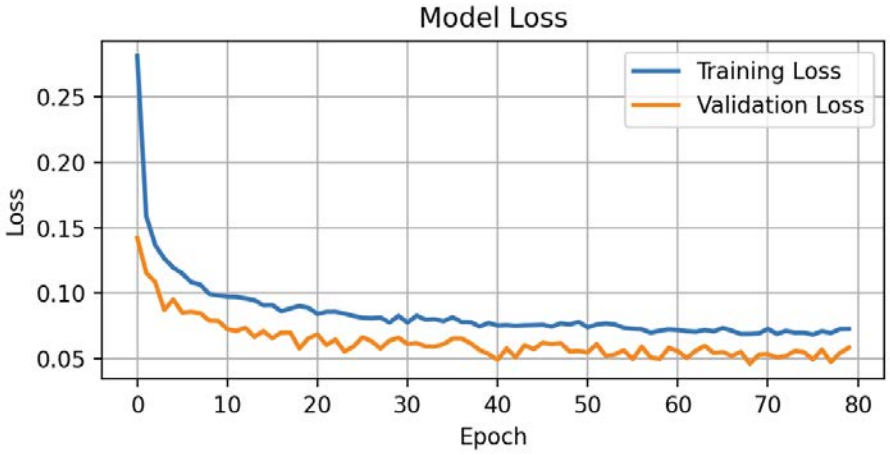


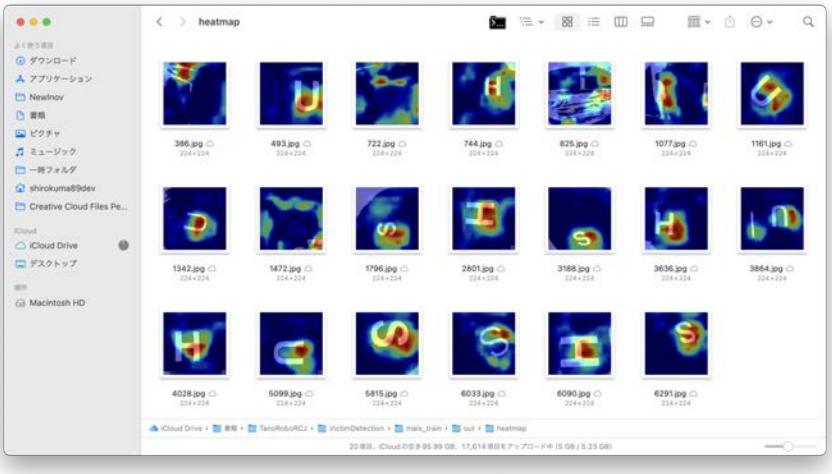
- **Freezing Bug:**

  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 tils in three dimensions (X, Y, and Z) for preventing freezing.

12th

We retrained our machine learning model for victim detection. We expanded the training data to include not only images of "H", "S", and "U" victims but also views of the robot's surroundings during operation. This reduced the likelihood of false positive detections. Additionally, we utilized Grad-CAM to visualize the model's focus areas, helping us understand and refine its learning process.





13th

We created a new machine learning model, but it didn't perform well due to overfitting on the "S" victim class.
We also optimized the return-to-origin program. Previously, the robot would signal completion upon reaching the origin during exploration. Now, the robot can navigate back to the origin using the shortest path.

15th

We found the exiting program had bugs, so we changed the algorithm to Dijkstra's method, which proved successful.

-1	-1	-1	-1	-1	-1	-1	-
-1	2	1	0	1	2	3	-
-1	3	4	3	2	3	4	-
-1	-1	-1	6	5	4	5	-
-1	-1	-1	-1	-1	5	6	-
-1	-1	-1	-1	-1	8	7	-
-1	-1	-1	-1	-1	-1	8	-
-1	-1	-1	-1	-1	-1	-1	-

Fig. example of Dijkstra's map

16th

We ordered 3D-printed parts from JLC3DP, utilizing their stereolithography service for its precision and high-quality finish.

20th

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 6×6 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.

22nd

The robot didn't sometimes successfully navigate over bumps. Therefore, we implemented a novel feature allowing the robot to autonomously detect if it was stuck. This idea was very innovative.

24th

The RoboCupJunior Japan Open 2024 was held in Nagoya, and we won first place, the Best Presentation Award, and a Sponsor Award. Our robot demonstrated a stable performance throughout the competition, without any significant

errors.

Toward RoboCupJunior2024

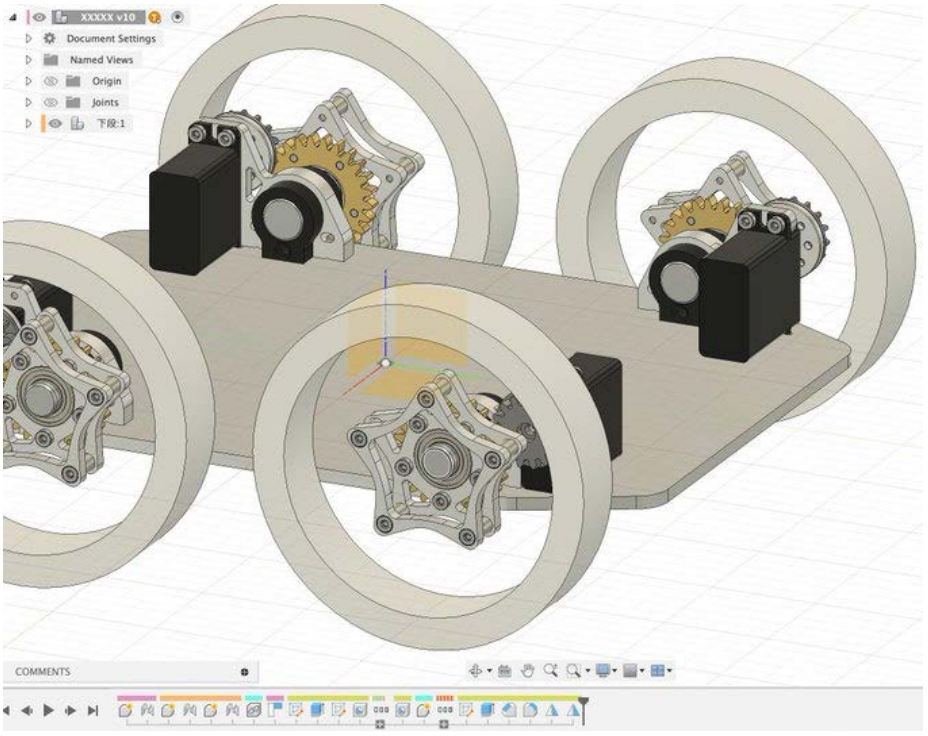
We summarize the advantages and disadvantages of ONAMAZU

	advantages	disadvantages	solutions
hardware	High maintainability	Can’t turn on big bumps	Redesign a cushion tire
software	Redundant algorithms	Not compatible with three-dimensional mazes.	Add a Z-axis to self- positioning

May

10th

We began to design a new robot for RoboCupJunior 2024.



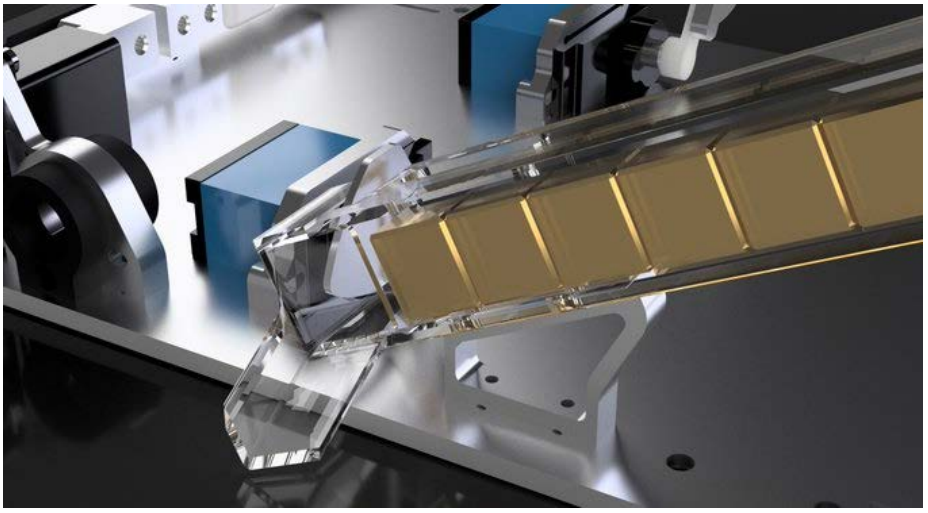
16th

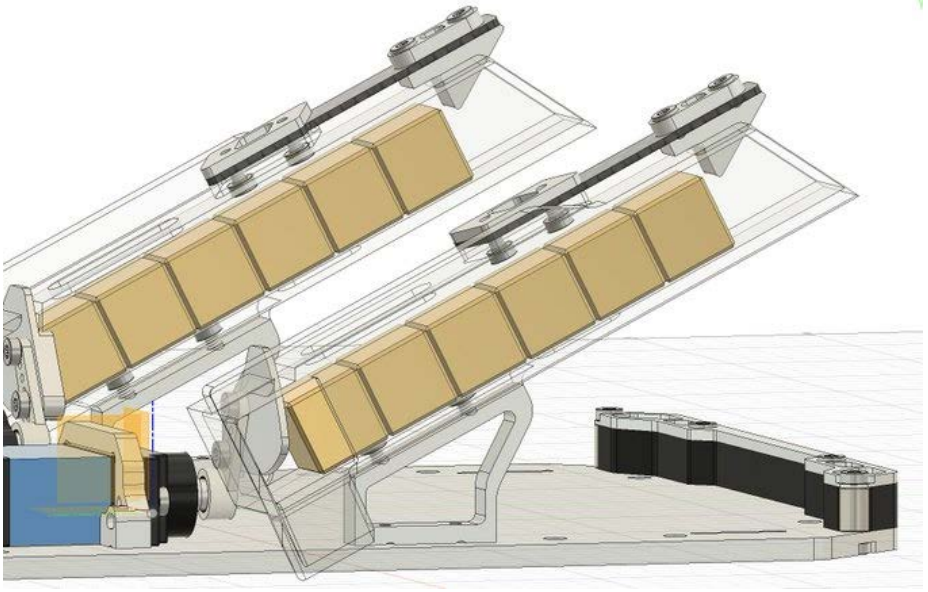
We designed new wheel cushion parts, adopting an airless tire structure similar to the one used on the robot in the national competition.

19th

We completed the rescue kit discharge mechanism, enabling it to reliably eject a predetermined number of kits in a specific direction.

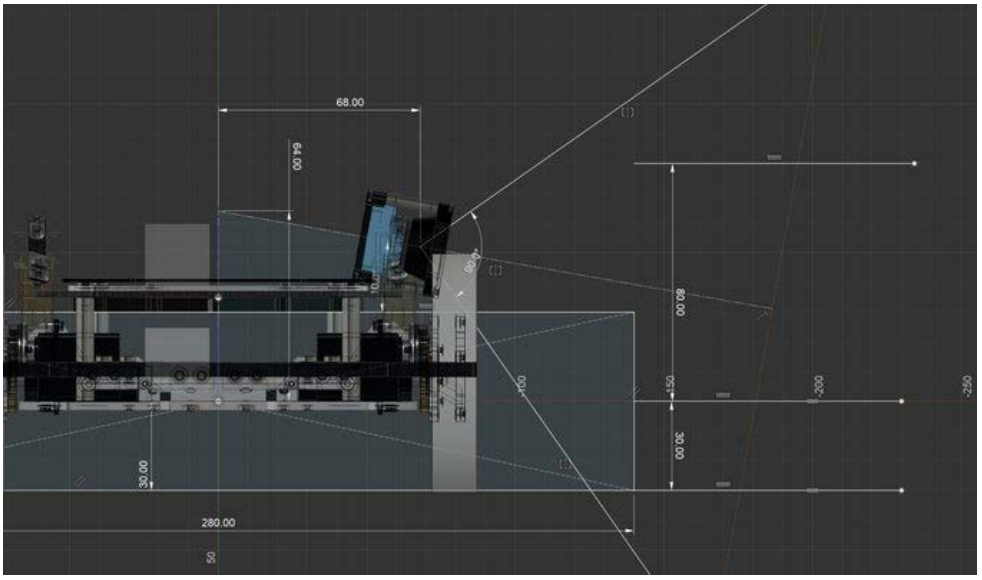
We completed the rescue kit discharge mechanism, enabling it to reliably eject a predetermined number of kits in a specific direction.





26th

We reevaluated the camera's field of view.



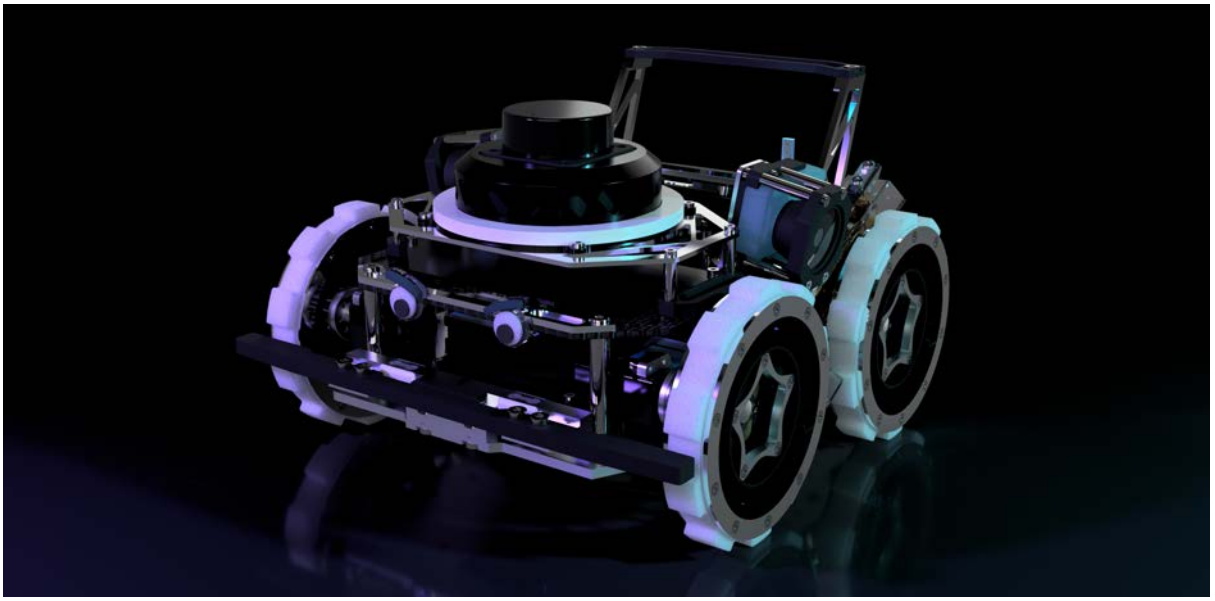
June

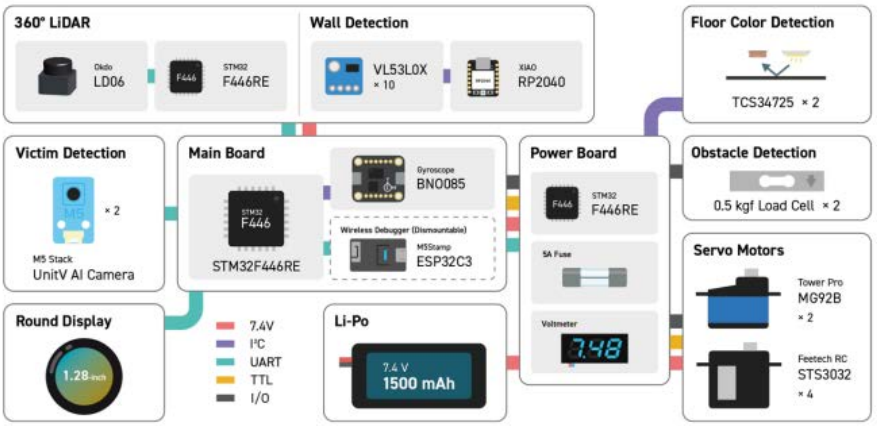
12th

Color Sensor Test: We used an iPad display to verify the linearity of the color sensor values.

20th

We completed the design of a new robot, RAICHO (雷鳥).





Research

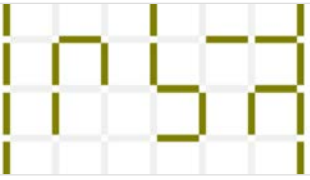
These are the links we referred to. (Some of them are written in Japanese.)

Right-Wall Following Algorithm

Fornax 迷路探索アルゴリズム解説｜さばみそ

こんにちは。Fornaxのハード&迷路探索担当のさばみそです。 この記事ではRCJ2024 名古屋オープンで使う迷路探索アルゴリズムの解説をしていきます。 どんなアルゴリズム？ 一言でいうと、「直進優先の探索」です。 ほとんどのチームでは、拡張左手（もしくは右手）法を採用していると思いま

https://note.com/sabamiso_rrsc/n/n06c5941bd80f



迷路の探索まとめ | たにたにっき

迷路の探索方法まとめ １．左手法 いちばん一般的 左手を壁についてゴールまで向かう 利点： コードが簡単，省メモリ 欠点： ゴールへの道が2本あったときに，たどり着かないことがある...

https://blog.tanitanin.net/?p=536

Ftech 教育教材ロボット 迷路解析の基本２～拡張左手法～

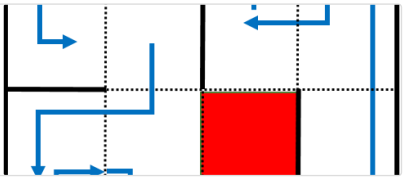
メカトロニクス学習・研修に、次世代エンジニアの創意工夫を支援！！ エフテック教育教材ロボットを紹介するページです。総合学習のツールとしてご利用ください。

https://www.ftech-net.co.jp/robot/howto/micromouse03.html

マッピング、とか - 地面。's Note

ソフトウェアの概要のところで丸投げしたマッピングと移動方向判断についてのことを書いていきます。(マッピングのことはまだです(殴)) ソフトウェア概要のところで書いたように、pallasのロボットは進行方向の判断材料に”各タイルの通った回数”と”優先して進む方向”を用意し、それらを使って拡張右手法を

https://nagumo-s-note.hatenablog.com/entry/2019/01/24/233933



Dijkstra’s Algorithm

ダイクストラ法による単一始点最短経路を求めるアルゴリズム | アルゴリズムロジック

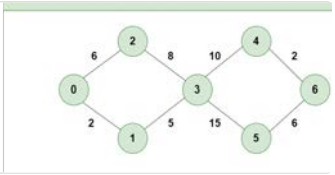
グラフにおける単一始点最短経路問題とは、始点を固定した時に、他のすべての頂点への最短経路を求める問題のことです。 ダイクストラ法は、単一始点最短経路問題を解く時に利用され、利点と...

https://algo-logic.info/dijkstra/

What is Dijkstra’s Algorithm? | Introduction to Dijkstra's Shortest Path Algorithm - GeeksforGeeks

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles, quizzes and practice/competitive programming/company interview Questions.

https://www.geeksforgeeks.org/introduction-to-dijkstras-shortest-path-algorithm/



BNO085 Gyroscope

Adafruit 9-DOF Orientation IMU Fusion Breakout - BNO085 (BNO080)

Here it is, the motion sensor you were looking for: the one that just gives you the directly usable information without requiring you to first consult with a PhD to learn the arcane arts of ...

https://www.adafruit.com/product/4754

cdn.sparkfun.com

https://cdn.sparkfun.com/assets/2/b/9/0/6/DS-14686-BNO080.pdf

TensorFlow / Machine Learning

Local model training - Sipeed Wiki

maixpy local model training

https://wiki.sipeed.com/soft/maixpy/en/course/ai/train/local.html

TensorFlow

An end-to-end open source machine learning platform for everyone. Discover TensorFlow's flexible ecosystem of tools, libraries and community resources.

https://www.tensorflow.org/

LiDAR

www.debix.io

https://www.debix.io/Uploads/Temp/file/20220921/LDrobot_LD06_Datasheet.pdf

共分散の意味と簡単な求め方 | 高校数学の美しい物語

共分散の定義，意味，簡単な求め方など覚えておくべきことを整理しました。

https://manabitimes.jp/math/853

Silicone Molding

カツマタケイコ／強みビジネスコンサルタント：材料と手順付き！オリジナルのシリコンモールドを作ろう♡

ご訪問ありがとうございます！ アイシングクッキーを始めとしたカラフルなデコレーションスイーツの基礎とノウハウが学べる教室の講師、カツマタケイコです。 アイシン...

https://ameblo.jp/sugarpopstudio/entry-12371667143.html

フィギュア複製を徹底解説！シリコンを使ってレジンを流す型を作ってみよう！

こんにちは。柚P（@yzphouse）です。 前回の記事『シリコン型とレジンを使った本格的な複製をしてみた！「パーツの粘土埋め編」』の続きです。 今回は、粘土にパーツを埋めたところに「シリコーンゴム」を流して、本格的なシ ...

https://yzphouse.com/hukusei-silicone/

Future outlook

We will update the field by creating a second floor and multi-level crossings. We will also adapt our software to accommodate the second floor and multi-level crossings.

These were our engineering journals. Now, we’re still developing our robot for RoboCup2024. We’re looking forward to show you the new robot :)