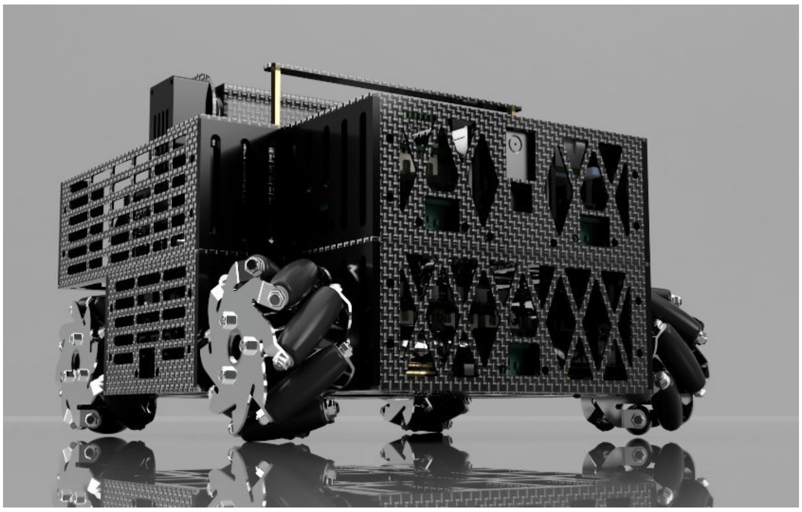
ROBOCUP ASIA-PACIFIC 2021
TEAM DESCRIPTION PAPER

League Name:	RCJ Rescue Line
Age Group:	Secondary
Team Name:	Hidden Eagle Talon
Team Website:	https://github.com/kamiokannde
Participants and Technical Roles:	Mitsuyoshi Sugaya : mechanical,software and electric circuit design. Ryota Obara : software design and debug.
Team Photo	
Mentor Name:	Fumio Yamamoto
Institution:	Kaiyo Academy (海陽学園海陽中等教育学校)
Region:	Gamagori Aichi Japan
Contact Person:	Mitsuyoshi Sugaya
Contact Email:	kamiokannde829@gmail.com
Date:	August 15

ROBOCUP ASIA-PACIFIC 2021

TEAM DESCRIPTION PAPER

RCJ Rescue Line

Hidden Eagle Talon

Mitsuyoshi Sugaya,Ryota Obara

(Gamagori Aichi Japan)

Abstract

Our team believes that the most important thing to complete all the tasks of the rescue line is to do a fast and accurate line trace. For that purpose, we think that it is necessary to predict the future position of the line and move the robot accurately to that position, and detect the future position of the line with a vector by 25 line sensors and cameras to determine the future position of the line. You can predict. This allows you to trace the line accurately by substituting a vector into the equation of motion of the Mecanum wheel. In the rescue zone, in order to efficiently collect the victims, eight laser distance sensors can be used to create a three-dimensional map and collect the balls in a short time. And to make it light and sturdy, the frame is a robot using carbon fiber reinforced plastic (CFRP).

In this way we were able to complete all the tasks within the time limit and create a robot with few failures.

1. Introduction

a. Team

- Team background

  Our team is made up of 5th graders from the Kaiyo Academy Robot Club Rescue Department , and has participated in the RoboCup Junior World League Tokai Tournament twice so far. Our team doesn't have a website, but I've attached a Github link.

- Team member

  Mitsuyoshi Sugaya : mechanical,software and electric circuit design.

  Ryouta Obara : software design and debug.

2. Project Planning

a. Overall project plan

- Purpose of the contest.

  The reason why our team participates in this contest is that we have learned so far by incorporating the achievements of robot control algorithms, designs, electronic circuits, etc. that we have learned in the past year into robots in this contest. This is to make it better and connect it to the future. It is also to deepen the knowledge of our team by learning the good points of other teams and to pass on their skills to the juniors of club activities.

● Overall project plan.

Our team has three project plans. The first is a hard plan. Our team has been competing in the LEGO Mindstorms EV3 so far, but we decided to design from scratch because we thought it would be difficult to do all the tasks with the EV3 due to changes in the rules. As a design requirement, we designed with the three goals of "does not break even when subjected to a large impact, is easy to maintain, and is light". The second is electronic circuit planning. Until now, I have only participated in RCJ with EV3, so I had nothing to do with electronic circuits, but now I have to design robots from scratch and design electronic circuits. At the beginning, we used jumper wires to wire the universal board, but from this robot, we can put a highly reliable electronic board on the robot by designing the printed circuit board by ourselves and ordering it from a company. It came to be. The third is software planning. I had to use Arduino as the main microcontroller because I shifted from EV3. Unlike EV3, Arduino had to be programmed in C, so I had to learn C first. However, by migrating to Arduino, I was able to write relatively free programs, and I was able to reduce program malfunctions and bugs.

● Milestone

1. Determination of design requirements and system requirements and selection of various sensors and parts.

2. Structural design.

3. Circuit design.

4. Calculation of the equation of motion of the Mecanum wheel.

5. Understand the functions of the camera.

6. Software design.

7. Fixing defects in the robot and circuits and adjusting each variable in the program.

8. Final adjustment .

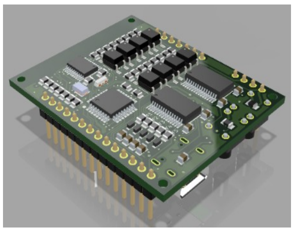
● Analysis of tasks and constraints and impact on projects.

Due to the change from the 2018 and 2019 rules to the 2020 rules, seesaws, rescue kits, etc. have been added to the tasks, and the rescue zone has also been modified. As a result, our team thought that it would be very difficult to complete all the tasks with EV3, so we designed the robot from EV3 with 3DCad, changed the microcomputer to Arduino mega, Arduino nano, esp32, and shifted the electronic circuit from EV3. Designed with KiCad, the printed circuit board is mounted on the robot. Also, I thought it would be difficult to distinguish the rescue kit with a distance sensor, color sensor, etc., so I decided to install a camera. For modification of the rescue zone, a three-dimensional map is created by installing eight laser distance sensors.

b. Integration plan

In our robot, 3 microcomputers, 37 various sensors, 9 motors, and 1 camera are collectively controlled by Arduino mega, which is the main microcomputer among the 3 microcomputers. The microcomputers communicate data by I2C communication, all sensor values are read by the main microcomputer, and the camera data is sent to the main microcomputer after picking up only the necessary information with the other microcomputer (Arduino nano). doing. In this way, in our robot, each microcomputer and sensor does not move separately, but moves in cooperation.

c. Testing



In addition to this robot, our team is developing a Bluetooth two-way serial communication system. By installing this system in our robot, we can check the values of all sensors, camera data, and motor movements on a personal computer. By analyzing the driving, sensor, motor, and camera data obtained in this way, it is possible to accurately calculate the error correction values for variables and motors in the program.

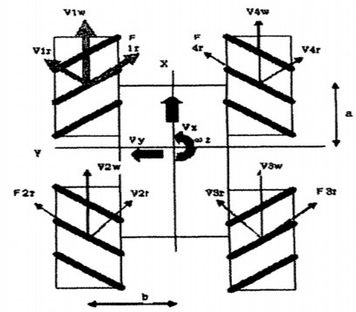
3. Software

a. General software architecture

The software installed in our robots is divided into three main parts. The first is a line trace program. Until now, only line sensors or color sensors were used for line tracing, but since this robot is equipped with a camera, it is possible to process line sensor and camera data at the same time. .. The second is a program for collecting rescue kits. A camera can be used to identify the rescue kit, and a gyro sensor can be used to collect the rescue kit carefully while constantly monitoring the angle of the arm. The third is a program for collecting evacuees. Eight laser distance sensors and cameras are constantly moving in the rescue zone to keep track of the robot's position. In addition, by generating a three-dimensional map, we always know where and what kind of ball is, so we can efficiently recover the victims and dead.

b. Innovative solutions

- Motor control function that applies the equation of motion of the Mecanum wheel.



The wheels of our robots use Mecanum wheels. Details of the Mecanum wheel and the reasons for its adoption will be explained in "Hardware". This Mecanum wheel can move sideways while facing forward, and has the characteristic of being able to move like a tank, such as being able to make a super-credit turn, and by applying this, it can move in all directions. increase. To take advantage of this feature, we calculated the equation of motion for the Mecanum wheel. When each variable of the equation of motion is calculated as shown in the figure on the right, the formula is as follows. By substituting the vector of the direction you want to move and the angle you want to turn into this equation, you can make a movement such as rotating 90 degrees while moving on a straight line.

$$\begin{bmatrix} \phi_{1w} \\ \phi_{2w} \\ \phi_{3w} \\ \phi_{4w} \end{bmatrix} = \frac{1}{R_w} \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \\ 1 & 1 & (a+b) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix}$$

- Line reading with line sensor and camera.

Until now, the only sensor used in line tracing was the line sensor. Our team used PID control to accurately perform line tracing with the line sensor, but the gain value adjustment etc. did not go well, and even if the line tracing was done properly, the accuracy is very good. It wasn't a thing. In the case of PID control, it is difficult to predict the future position of the line unless the gain value is adjusted accurately, so instead of predicting the future position of the line using a camera, we will use the future position in the future. I thought it would be better to prevent it, that is, to detect the line to be traced by the camera before the robot traces the line. This way you don't have to move the motor with inaccurate data.

As shown in the previous section, in order to control the Mecanum wheel, the direction vector you want to move must be substituted into the equation of motion. Therefore, analyzing the data of each of the 25 line sensors requires a very complicated algorithm to generate the direction vector, so the 25 data are converted into a linear function using the least squares method. , I am fixing it to the direction vector. However, when the least squares method gives an accurate value, it means that the robot is not located on an intersection or an acute-angled line, and since the least squares method is actually an average of numerical values, multiple lines are detected. If it is, an incorrect value will be output. The camera can also identify lines that are divided into two or three lines, and each can be output as a direction vector, so while the robot is not located on a straight or curved line, the motor uses the direction vector obtained from the camera data. Control. In this way, accurate line tracing can be performed by using the line sensor and camera when the line is straight or curved, and only the camera when there are many lines.

- Efficient recovery of victims by laser distance sensor.

Our robot is equipped with eight laser distance sensors. After entering the rescue zone, the robot makes one revolution. This is because the positions of the evacuation zone, the victim, and the robot are plotted on a three-dimensional map using polar coordinates with eight distance sensors by making one rotation. However, when the victim and the victim overlap, they are regarded as the same target, so I try to plot the target on the map by rotating it at least twice. When collecting a target, the robot cannot collect it without hitting the ball against the wall, so it is important to know the position of the robot itself. The resolution of this distance sensor is 1 mm, but if you make a map in 1 mm units, it will overflow, so the position of the robot is managed with a map in 1 cm units. Also, the question of which ball should be collected from is the most efficient, because the number of balls is not so large, by calculating the travel distance of all routes and analyzing the travel time of each route. , I am trying to collect it through the shortest route. In this way, the victims can be recovered efficiently.

- Minimize program execution time.

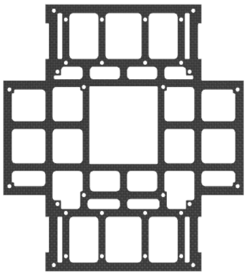
When programming an Arduino series microcomputer, the functions specified by the Arduino IDE, that is, the input / output functions such as IO pins, are not designed to shorten the program execution time as far as the Arduino API is seen. To move the line sensor one cycle, you need to use the IO pin input / output function more than 10 times, but this input / output function does not move as fast as we expect, so you can move it several cycles per hour. Our team thought it wasn't very suitable for line sensor programs where it was important to be able to do it. Therefore, by directly manipulating the Arduino registers to input and output IO pins, the program execution time can be shortened compared to existing functions, and the number of programs that can be executed per hour can be increased. However, since the pin numbers of the registers differ depending on the type of chip mounted on the microcomputer, there is also the disadvantage that programs that directly operate the registers on other microcomputers cannot run. By directly manipulating the registers of the microcomputer in this way, the program execution time can be minimized.

c. Source code

- https://github.com/kamiokannde/Rescue_project

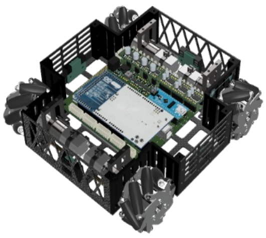
4. Hardware

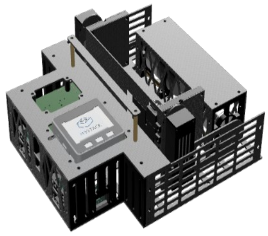
- Hardware design overview.



As shown in "Project Planning", the hardware design of our robot is designed with the goal of "does not break even if a large impact is applied, is easy to maintain, and is light". But the first and third goals are inconsistent, so we needed to study the structural materials that support the robot's main weight. Therefore, we focused on reinforced plastic (FRP) as a light and strong material. There are many types of reinforced plastics, but we decided to use carbon fiber reinforced plastics (CFRP), which boasts lightness and high load capacity, for robots. Because CFRP has the property of being lighter than other FRPs. However, CFRP has the disadvantage of being more expensive than other FRPs. In order to solve this problem, by designing with 3DCad(Fusion 360), we can know the position of the center of gravity of the entire robot, where and what kind of force is applied, etc. I can. This can reduce the weight of CFRP used, which leads to cost reduction. The photo on the right is the CFRP board at the bottom of the robot. Regarding the second goal, by designing with 3D Cad, each part can be made with a 3D printer, so it is possible to make a large space for loading batteries etc. and maintainability is better than when using commercially available parts. It is improving. In this way, our robot is structured according to three goals.

- Combination of each function.





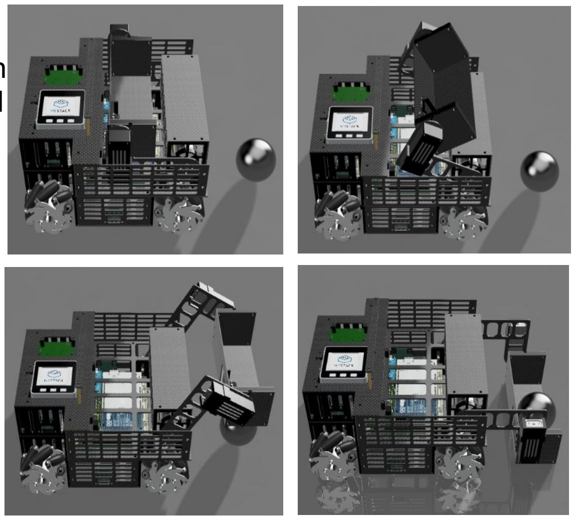
Also, as shown in the following picture, our robot is mainly divided into two layers. The first level consists of four motors with wheels to support the robot, four distance sensors, and the main electronic board. The second level consists of a module for collecting victims and rescue kits, a sub electronic board, four distance sensors, three gyro sensors, and a camera. The base and motor of each layer are designed by 3D Cad, so they do not interfere with other parts. In addition, since the frames other than the parts created by CFRP are created by the 3D printer, the structural weight can be distributed and the weight can be reduced.

a. Mechanical design and manufacturing

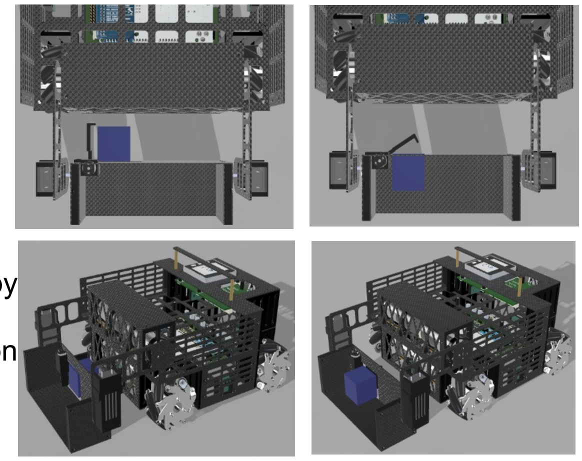
- Use of carbon fiber reinforced plastic and 3D printer.

CFRP is used for the main frame of our robot. This is to make the robot lighter and more rigid. CFRP is a carcinogen that produces dust during CNC processing. To prevent this dust from scattering into the air during processing, the CFRP plate is submerged in liquid for processing to prevent dust from being generated. In addition, parts other than CFRP, that is, motor mounts, etc., are created using a 3D printer. The advantage of using a 3D printer is that we can quickly create parts made with 3D Cad. In this way, the frames, various joints, parts, etc. of our robot are all created within our team.

● Rescue kit deployment and rescue mechanism.



Our robot uses five motors to rescue the victims. To rescue the victims, deploy the arms in the order of upper left, upper right, lower left, and lower right. After deploying the arm, the robot uses a distance sensor to measure the distance to the wall and pushes the victim against the wall to push the victim into the cage. After recovering the victim, the arm moves in the opposite direction and moves back to its original position. The robot then goes to rescue other victims. Since the gyro sensor constantly reads and monitors data such as the position and angle of the arm when the arm is returned to its original position, the basket attached to the arm is programmed to always be parallel to the ground so as not to drop the victim. increase. When releasing the victim, you can move the arm to the state shown in the lower left photo and then tilt the basket to put the victim into the evacuation zone. Robots can also carry up to three victims. In this way, the victims can be evacuated efficiently in a short time.



To collect the rescue kit, first grasp the positional relationship between the rescue kit and the robot with a camera. The robot then uses the distance sensor to move to a position where the rescue kit can be retrieved. When recovering, extend the arm in the same way as when rescuing the victim. However, the rescue kit is square and there is no wall that can be pushed around, so you can put it in the basket by hooking it with the claws attached to the basket as shown in the two pictures above. Like a spoon when eating ice cream. (The two photos below are side views of the photo above.) The deployment is the same as putting the victims in the evacuation zone. In this way, our robot collects and deploys the rescue kit.

● Omnidirectional movement with Mecanum wheels.

We thought that ordinary tires like before would not give us the exact line traces we were aiming for. So we decided to use Mecanum wheels. Mecanum wheels are wheels designed to move in any direction. Rollers are mounted on the wheel circumference at a 45 degree angle. By transmitting the driving force, in addition to the same movement as the conventional wheel, it can move in the direction of 45 degrees with the free roller. By adjusting the rotation direction and speed of the four motors, it is possible to move in all directions by rotating the wheels and moving the rollers. In a typical four-wheel configuration, each wheel applies force approximately at right angles to the diagonal of the wheelbase to which the wheels are mounted. By changing the rotation speed and rotation direction of each wheel, it can move and rotate in any direction. If you move all four motors in the same direction, they will move back and forth. Moving one wheel in the opposite direction to the opposite wheel creates a rotational motion, and moving one wheel in the opposite direction to the other diagonal wheel moves laterally. The combination of these wheel movements allows for movement in all directions, with or without rotation of the robot.

b. Electronic design and manufacturing

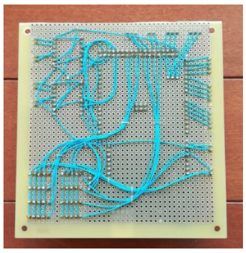
- Use of camera.

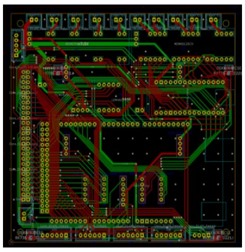
We use a camera called Pixy2 to do many tasks. Pixy2 uses a color-based filtering algorithm for object detection. This method is popular because it is fast, efficient, and relatively stable. Calculate the hue and saturation of each RGB pixel through the image sensor and use them as the primary filtering parameters. Changing the normal lighting and exposure does not significantly change the hue of the object, but it can have a serious impact on the color filtering algorithm. However, Pixy2's filtering algorithm is highly stable and resistant to changes in lighting and exposure. In this way we can use Pixy2 properly.

- Laser distance sensor.

The eight distance sensors mounted on this robot are laser distance sensors. The reason we use laser distance sensors is that they are more accurate than ultrasonic sensors. Ultrasound diffuses in proportion to distance, making it unsuitable for detecting small targets such as victims and rescue kits, and even if they do, the distance is a bit vague. However, with a laser, it goes straight without diffusing and has an accurate resolution of 1 mm, so even small targets can be detected. The laser distance sensor we use uses STMicro's laser distance sensor VL53L0X, which can measure distances up to 200 mm.

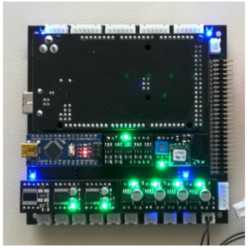
- Improved reliability with printed circuit board.





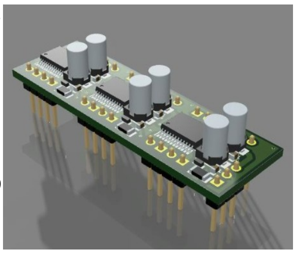
Printed circuit boards are used as electronic boards for our robots. We had to build our own foundation after shifting from EV3 to Arduino. At the beginning, as shown in the photo on the right, the universal board was wired using jumper wires. However, in wiring using jumper wires, many disconnections occurred, such as poor soldering between jumper wires and catching the board somewhere. Also, jumper wire wiring makes it difficult to know where and which wire is connected, and it takes hours to find the unconnected part. In addition, the universal board has the disadvantage that the size of the board becomes large because SMD parts cannot be attached. Therefore, by designing the board on a personal computer and placing an order with the board company, we were able to make a compact electronic board without wiring defects. However, there are also problems caused by using printed circuit boards. The point is that it is difficult to correct the printed circuit board if the wiring is incorrect. However, this problem can also be avoided by dividing the board into motor drivers, power supply circuits, LED drivers, etc. and modularizing them. In this way, we mount a highly reliable electronic board on the robot.

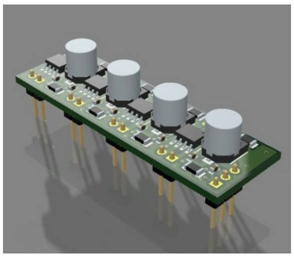
- Main board and each module.



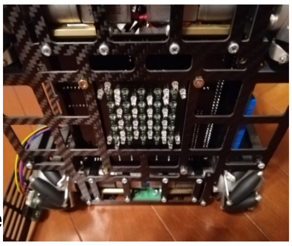
The main board in the photo on the right is mounted on this robot. This foundation mainly consists of 7 modules. Even if something breaks by separating each module, there is an advantage that it can be repaired by replacing only the broken module. And even if the robot is remade, each module can be reused.

First is the motor driver. There are two types of motor drivers mounted on the board, the first type is a module using TB67H450FNG and the second type is a module using TB6612FNG. The reason why we use a motor driver is that the IO pin of the Arduino mega we are using can only output 40mA per pin, but the motor consumes up to 3A of current, so connect the motor directly to the Arduino. If you do, the Arduino may break. In addition, by relaying the motor driver, the speed of rotation of the motor and the direction of rotation can be controlled. The reason for using two types of motor drivers is that the amount of current required differs between the motor connected to the robot wheel and the motor that collects the victims and rescue kits. You can also protect a microcontroller such as Arduino from overcurrent by using a motor driver. The previous robot used a commercially available motor driver, but now that I design the printed board myself, I can make a motor driver module that uses a motor driver that is compatible with the motor.

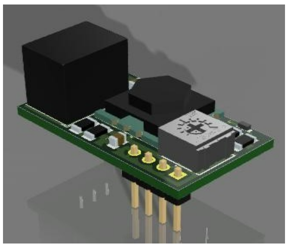




The second is the line sensor. The line sensor consists of 25 phototransistors, 16 white LEDs and 4 RGB LEDs. The line sensor is directly connected to the back side of the main board with a pin header. This is because connecting as many as 25 data and LED signal lines with a cord requires a large number of cords, and the area where the cord connector is attached squeezes the area of the main board. In addition, the LED driver module that controls the LEDs is also mounted on the main board, and 16 LEDs can be operated at once. Twelve of the phototransistors are read in analog and the other 13 are read in digital. The reason for reading data in analog is that 12 of the 25 have both line and color sensor functions. In the case of a line sensor, the threshold value can be determined by a semi-fixed resistance and there is no problem even if it is read digitally, but in the case of a color sensor, it is necessary to be analog because each RGB color is applied in order and the intensity of the reflected light is read. Because there is. In this way, our robot reads lines with 25 line sensors and 12 color sensors.



The third is the power supply circuit. The power circuit consists of a fuse and a DCDC converter. This DCDC converter can carry a voltage of 5V up to 6A. Since it can flow up to 6A, the 28 LEDs of the line sensor can be illuminated to the maximum amount of light, and the line can be read without being affected by the surrounding environment. In addition, since it has an overcurrent prevention function and a backflow prevention circuit, damage can be suppressed even if a short circuit occurs.



The last is the microcontroller. Arduino nano and Arduino mega are mounted on this board. The Arduino maga is the main microcontroller and the Arduino nano is the sub microcontroller. The two microcomputers are connected by I2C communication. In addition, M5stack (esp32) is installed on the sub-board. Since this microcomputer has a liquid crystal, it is used to display data.

In this way, all the electronic boards mounted on our robots are printed circuit boards, and since each function can be replaced as a module, it is a highly reliable board that can be replaced immediately even if it breaks down.

5. Performance Evaluation (Result)

- Our robot is a robot that can perform line tracing accurately and quickly according to the original goal. In addition, the base mounted on the robot is not broken, and I think that the design requirements indicated by the milestones have been fully achieved. However, it is still difficult to complete all the tasks of the rescue line only by line tracing, so I would like to improve the program in the period until the contest.

6. Discussion and Conclusion

- In conclusion of this TDP, our team thinks: In robots, software, hardware, and electronic circuits do not work individually, but work together. If any one of these three elements is applied, the robot will not give the best performance. Each of these three elements supplements, but there is also a range that can be supplemented. In order to complete all the tasks of the rescue line, even if any one element is particularly good, it cannot be completed, but it is completed by all the elements moving uniformly and perfectly. Our team is thinking.

  I think that the hardware design and software algorithms of this robot will have a great impact on the project. Basics Basic hardware design and software algorithms support the bottom of our robot, without which the various parts, electronic boards, modules, and functions mounted on this robot would not be possible. In addition, successful development of hardware design and software algorithms will facilitate project planning along milestones.

  The learning experience of a team can be broadly divided into three. The first is hardware design with 3D Cad. Now that you can design with 3DCad, you can quickly make the parts you envisioned or parts that are not on the market using a 3D printer or the like. In addition, I was able to learn techniques such as the positional relationship of the center of gravity of the robot and weight reduction. The second is board design by KiCad. By shifting from EV3, I had to learn from the basics of circuits. This allowed me to learn the laws and physical phenomena necessary for circuit design. Also, I was able to understand the country of foundation that I had been using. The third is software algorithms. I learned how to operate the registry of a microcomputer to shorten the program execution time and the technique to omit unnecessary programs.

  In the future, I would like to strengthen the areas where the three factors, such as adjusting the variables of each program and increasing the strength of the parts made by 3D printers, are not enough for the contest.

7. Acknowledgements

- We express our sincere thanks to parents of our team members and Dr.Yamamoto,an advisor for club activities.

8. References

- http://www2.lib.yamagata-u.ac.jp/you-campus/sangitan/kiyou-sangitan/11/p81-84.pdf
- https://www.jstage.jst.go.jp/article/ieejias/128/6/128_6_849/_pdf/-char/ja

Appendix

- Programs and 3D data are released on Github in sequence.

  https://github.com/kamiokannde/Rescue_project