Development of an Open-source Educational and Research Platform for Autonomous Cars

1st Naohiro Nakamoto
Graduate School of Robotics & Design
Osaka Institute of Technology
Osaka, Japan
m1m19r26@oit.ac.jp

2nd Hiroyuki Kobayashi
Graduate School of Robotics & Design
Osaka Institute of Technology
Osaka, Japan
hiroyuki.kobayashi@oit.ac.jp

***Abstract*—Technological innovation in recent years is called "The Fourth Industrial Revolution." As a typical example, the development of autonomous driving is thriving in the world. However, if cars are automated, there is a problem that it is necessary to secure people who can technically deal with autonomous cars in order for us to coexist fully with the autonomous cars. Furthermore, in Japan, it is difficult to provide the cars continuingly, because the shortage of IT personnel is a serious problem. In this study, the authors developed a hardware platform for research and education usage to assist autonomous driving technologies. The platform is a one-tenth size model car that mainly consists of ordinary consumer electronics such as Raspberry Pi and is also an open source. It is not only affordable but also easy-to-modify by everyone. Therefore, the hurdles to learn autonomous driving technologies are very low. In addition, the platform isn't a general robot like LEGO MINDSTORMS, because it is a model very similar to a car and is a robot specialized in autonomous driving technologies. We made three experiments to verify functions of sensors and actuators attached to the model car. As the results, we confirmed that the car can be used as the platform without any big errors.**

***Index Terms*—autonomous driving technologies, open source, educational robot**

I. Introduction

In recent years, by the rapid development of Information and Communication Technology (ICT) around the world, it has been possible to share information among various devices. In addition, the rapid ICT development has created new economic values such as improvement of work efficiency. This disruptive technological changes can be called "The Fourth Industrial Revolution," which is one of the political themes of the Cabinet Office in Japan [1]. The revolution is expected to have an impact on the get rid of shortage of manpower and the support of people's lives. A symbolic example is the development of autonomous driving that is being promoted for practical use in various countries around the world. It uses technologies such as AI and IoT, which are the main technological advances of the Fourth Industrial Revolution. It has received a lot of attention among advanced countries in the world.

However, if cars in the world are automated, it will have many problems. As one of it, in order for people to fully coexist with the autonomous cars that are in the rapid technological innovation of modern times, it is necessary to secure the people who can deal with autonomous cars technically in the future. In Japan, it is easy to estimate the shortage of autonomous driving engineers, because the shortage of IT personnel is a serious problem [2]. Therefore, as one tool to solve such problems, various open source platforms such as Autoware [3] developed by Nagoya University are provided for research and development of autonomous driving technologies. However, the platforms cannot be handled easily because they require a real car to which expensive sensors are attached and a roadway on which the car runs. It is difficult to train autonomous driving engineers, so the society in which people and autonomous cars can coexist will become a dream, because the hurdle to learn autonomous driving is so high.

In this study, the authors will develop a platform to assist in an education and research of autonomous driving technologies. We focus on a small-size and low-price car, considering locations (such as roadway) and costs to solve the problem. In addition, the platform will not be a differential 2 wheel robot like Raspberry Pi Mouse or LEGO MINDSTORMS, because it will be a model very similar to a car. Therefore, it will be a robot specialized in autonomous driving technologies. We will also introduce the development of an educational and research model car for assisting a new era's technologies.

II. Configuration

In this section, the authors will explain the system configuration (hardware and software) of our one-tenth size model car created by them. In this study, in order to give versatility to future applications, we installed the Robot Operating System [4] (ROS) as a framework on the model car. ROS can control various sensors, actuators, and so on, as separate nodes, simultaneously. In addition, ROS is versatile in education, because it is compatible with programming languages such as C++, python and java.

Figure 1 shows the system configuration of the car. A Raspberry Pi 3 model B is used for it. Because sources of various "programs and examples" using a Raspberry Pi are published on the internet. In addition, Ubuntu which can be installed as Raspberry Pi's OS officially supports ROS. Also, essential sensors for autonomous driving such as Light Detected And Ranging (LiDAR).

978-1-7281-4878-6/19/$31.00 ©2019 IEEE

Figure 2 shows the model car. Th model car is created based on the Tamiya chassis: “58600 TT-02 Type S Chassis 4wd Kit.” This chassis has an Ackerman link [5] implemented as it would be in a real car. By this link mechanism, it is possible to run like a car.

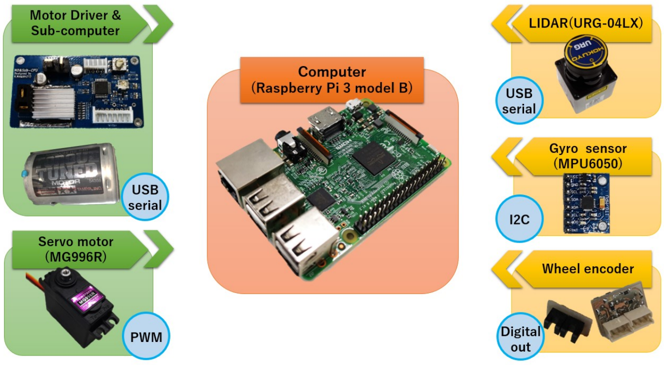

Fig. 1. The system configuration

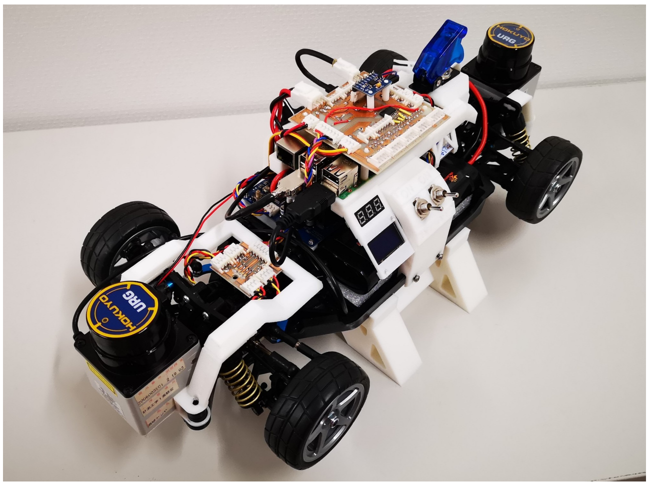

Fig. 2. The one-tenth size model car

A. Circuit boards

There are five types of circuit boards that make up the car. Their boards are created using KiCad that is an open source project. Although this software has different specifications depending on the version, it is excellent in functional extensibility and versatility. The role of each board is as follows.

- Daughter board (Figure 3)
  - The Daughter board was designed to be inserted directly into Raspberry Pi’s pin headers. This board can mainly handle control of input of wheel encoders and an inertial sensor and a servo motor. In addition, in order to have versatility in future applications, we prepared multiple interfaces: I2C, SPI and UART pins and ultrasonic sensors.
- Motor-driver & Sub-computer board (Figure 4)
  - The Motor-driver & Sub-computer board can mainly receive signals from Raspberry Pi and perform motor control. A STMicroelectronics STM32F303K8T6 microcontroller compatible with ROS was attached to this board. As this microcontroller is an ARM microcontroller like Raspberry Pi, the control voltage is equal and, the compatibility is good.
- Power-supply board (Figure 5)
  - The Power-supply board can supply battery power to Daughter board and Motor-driver & Sub-computer board. As over current protection, three poly switches with withstand voltage of 30 V and withstand current of 5 A were implemented.
- Wheel-encoder board (Figure 6)
  - Four Wheel-encoder boards were attached to the chassis front of the car. These boards can detect the number of rotations by reading slits of the original slit plate which is 22 pulses per rotation.
- Wheel-encoder branch board (Figure 7)
  - Chip LEDs were implemented on the Wheel-encoder branch board and can visualize output from the wheel encoders. Also, this board has the role of putting together the wiring of wheel encoders.

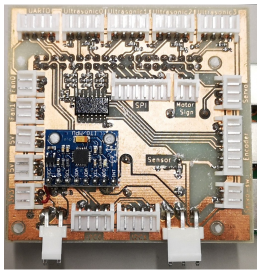

Fig. 3. The Daughter board

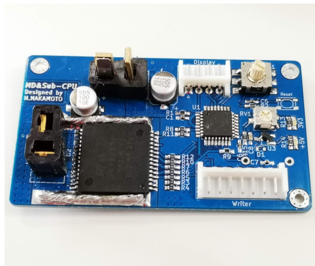

Fig. 4. The Motor-driver & Subcomputer board

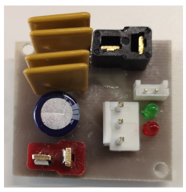

Fig. 5. The Poer-supply board

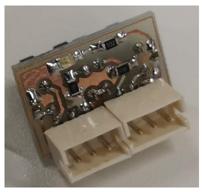

Fig. 6. The Wheel-encoder board

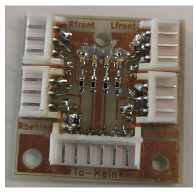

Fig. 7. The Wheel-encoder branch board

B. Open-source using GitHub

Five kinds of self-made circuit boards were made open, and the open source is managed by the public repository of GitHub. Anyone can be downloaded from the following URL. https://github.com/Auto-Naohiro/one-tenth-size-autonomous-car-circuit

III. Evaluation experiment

The purpose of this study is to develop the hardware platform. Therefore, the authors needed proofs that sensors and actuators attached to the small model car can work on ROS. We made three experiments to verify them. We will explain three experiments using an Ackerman link, two actuators and two sensors: URG-04LX (LiDAR) and MPU6050 (inertial sensor) attached to the car.

A. Operation of an Ackerman link

We experimented whether an Ackerman link implemented in the car functioned well. The angle at which the tire was turned to maximum was defined as the maximum turning angle. In that state, we shot an image from directly above the tire. From this image, with the pixel at the center of the tire width as coordinates, normal vector was determined from $P_{\rm t1}$ and $P_{\rm t2}$ of the two points at both ends of the tire. In addition, the normal vector, which is the vertical reference of the inclination, was also obtained from $P_{\rm b1}$ and $P_{\rm b2}$ of the two vertical parts of the car (Figure 8).

In order to obtain tire movement plots, we created the field as shown in Figure 9. The radius of the circumference circle drawn on the experimental field was 590 mm, which is equal to the minimum turning radius of the car. That is, this circle is the ideal trajectory of the outer tire of the car at the time of maximum turning. In addition, scales were placed on this field as shown in Figure 10 at equal intervals. The scale used in the experiments graduated in millimeters from zero to seven. Finally, a smartphone camera was placed just above the outer tire of the car as shown in Figure 11. Then, plot points were determined from a captured video clip.

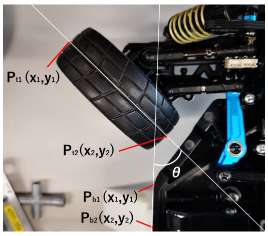

Fig. 8. Measurement of turning angle

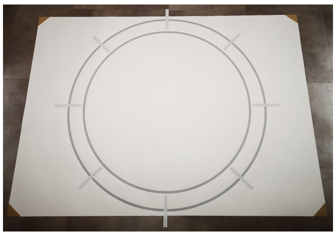

Fig. 9. Ideal trajectory circle

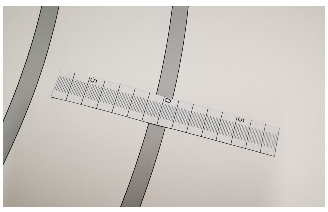

Fig. 10. Scale using measurement

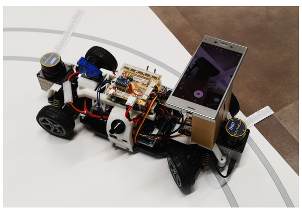

Fig. 11. Measuring method

B. Wall recognition using LiDAR: URG-04LX

The relationship of nodes used for wall recognition is shown in Figure 12. The role of each node is as follows.

- urg_node (https://github.com/ros-drivers/rosserial)
  - "Urg_node" was created by Open Robotics [6] which has official support for ROS, and was published on GitHub. This node can easily connect LiDARs manufactured by Hokuyo Automatic Co., Ltd. on ROS. In addition, it publishes to the number of laser points and a measurement distance at each point on the topic named "/scan."
- urg_data_node (Self-made node)
  - "Urg_data_node" subscribes to "/scan" and performs wall recognition and wall angle measurement. This node publishes measurement results to "serial_node" and "servo_node" on the topic named "urg_msg."
- serial_node (https://github.com/ros-drivers/urg node)
  - "Serial_node" is a package in "rosserial" created by Open Robotics and was published on GitHub. This node can subscribe to the topic published from "urg_data_node" to STM32F303K8T6. It has a role to link Raspberry Pi and STM32F303K8T6.
- mbed_node (Self-made node)
  - STM32F303K8T6 that controls a DC motor can be programmed on a web browser using mbed OS 5 of Arm Limited Co. Binary files can be downloaded by compiling programs. In this study, we used a library called "ros_lib_kinetic" [7] that can develop ROS in the mbed OS environment. By introducing this library, STM32F303K8T6 can be treated as a ROS node. This node subscribes to the topic of "serial_node" and controls the DC motor.
- servo_node (Self-made node)
  - "Servo_node" subscribes to "/urg_msg" and uses it to control a servo motor. After defining the upper and lower limits of the servo motor, a subscribed angle is used as the output of the servo motor.

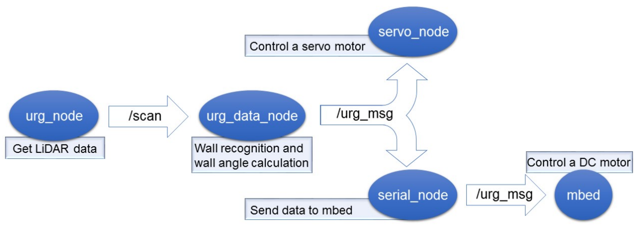

Fig. 12. Nodes of Wall recognition

In this study, we experimented runs parallelly along a wall. As shown in Figure 13, the initial location of the car was set 0.5 m away from a wall with reference to URG-04LX. When the car is horizontal to a wall, the angle of a wall calculated from data of URG-04LX is -90°.

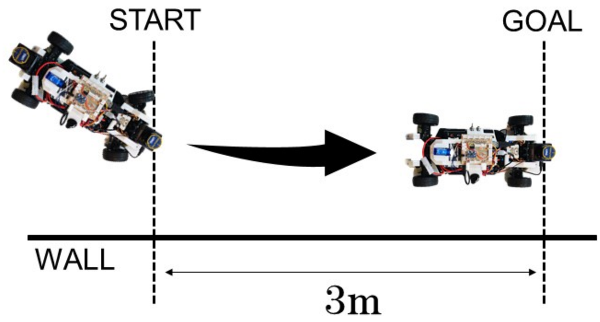

Fig. 13. Horizontal driving

C. Angle detection of the small model car using six-axis inertial sensor: MPU6050

The relationship of nodes used for angle detection is shown in Figure 14, and the role of each node is as follows.

- mpu6050_node (Self-made node & https://github.com/richardghirst/PiBits/tree/master/MPU6050-Pi-Demo)
  - "Mpu6050_node" can distribute the output of the MPU6050 YAW axis as a topic. In order to control a MPU 6050, we used the Digital Motion Processor [8] (DMP) mounted on it. DMP makes it possible to obtain attitude estimation values from acceleration and gyroscope data, and can reduce the load on a host processor. A library for using DMP is released on GitHub, and it is downloaded to this node and introduced to ROS. This node distributes the topic named "/mpu6050_msg" to "servo_node."
- servo_node (Self-made node)
  - "Servo_node" subscribes to the topic that is the output of an "mpu_6050_node" YAW axis, and the servo motor is controlled using it.

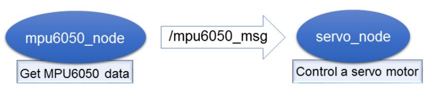

Fig. 14. The system configuration

In this experiment, a library group for using a MPU6050 DMP was made available to ROS. The purpose was to place the car on a rotating table and to measure angles using DMP. The experimental field used in this experiment is shown in Figure 15. In order to be able to measure angles using this field, a plate for positioning was attached at the top of the car. After that, a rotating table was aligned with this field and placed. At this time, as shown by the circle in Figure 16, the plate and this field were aligned vertically. This was the initial position of the car. After that, the car was measured by rotating 90 $^\circ$ to the left and right.

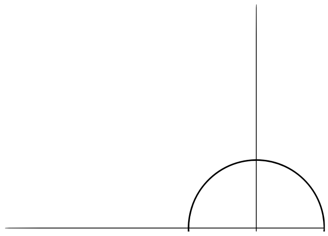

Fig. 15. The experimental field

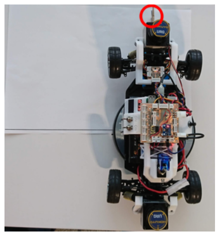

Fig. 16. Initial position of Angle detection

D. Open-source using GitHub

These ROS nodes were made open, and the open source is managed by the public repository of GitHub. Anyone can be downloaded from the following URL.
https://github.com/Auto-Naohiro/one-tenth-size-autonomous-car-program

IV. Results and Discussion

A. Operation of an Ackerman link

At the time of this experiment, the small model car ran five rounds of the experiment field, and during that time the camera took a movie. The experiment was conducted at each of the left and right maximum turning angles, and the average of the scale values when the outer tire passed five times on each scale was plotted as shown in Figure 17 and Figure 18.

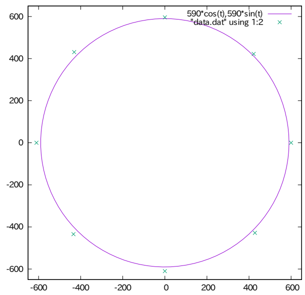

Fig. 17. Eight movement plots when turning right and ideal trajectory of the small model car

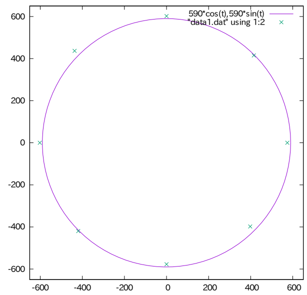

Fig. 18. Eight movement plots when turning left and ideal trajectory of the small model car

According to Figure 17 and Figure 18, the maximum deviation was 27.6 mm outside the 590 mm of the ideal trajectory circle. In addition, it was at least 3.40 mm away from the reference trajectory circle. However, those outliers are acceptable ranges, because the authors assume the car in normal driving control. Therefore, in this study, it can be judged that the basic operation is done.

B. Wall recognition using LiDAR: URG-04LX

The car ran three meters from the initial position, and the calculated wall angles were plotted as shown in Figure 19. The vertical axis of this figure is the wall angle, and the horizontal axis is the number of plots.

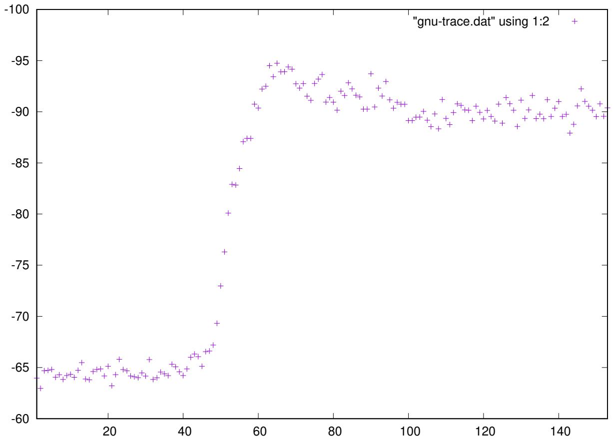

Fig. 19. Plots of calculated wall angle

According to Figure 19, when the number of plots exceeds 60, the angle of the wall is an angle that exceeds the target value of -90 ° . However, the angle of a wall was stable after the number of plots exceeds 100. Therefore, it turned out that it was able to run horizontally with a wall. As a result, it can be judged that the angle value of a wall can be output to a servo motor by the system of the car, and LIDAR can be controlled.

C. Angle detection of the small model car using six-axis inertial sensor: MPU6050

In this experiment, the output generated by MPU6050's drift was treated as the initial angle. The left and right angles measurement is shown in Table I.

TABLE I
MEASURED VALUE IN ANGLE MEASUREMENT(°)

Direction of rotaion	Init state(0°)	90°	Reverted state(0°)
Right	-1.04	-92.25	-2.61
Left	-1.65	89.78	-3.43

In this experiment, it experimented in the state which noise does not occur easily. According to Table I, the errors at the time of right turn were only 1.34% and 1.74%, and the errors at the time of left turn were only 1.58% and 1.98%. However, these measured values are in an ideal state, and in the case of actual driving, the errors may be large. A six-axis inertial sensor used in this experiment has no axis of attitude angle, so the attitude angle was determined by DMP using six axes. Therefore, in order to improve the accuracy in actual running, it is necessary to perform attitude angle measurement using a 9-axis inertial sensor instead of a 6-axis inertial sensor.

V. CONCLUSIONS

In this study, the authors created an educational and research hardware platform for autonomous cars, and released the designed circuit boards on GitHub. In addition, we experimented operations of sensors and actuators built in a ROS environment, and we evaluated it as a platform. According to our experiments, the small model car was able to perform basic operation. Therefore, a hardware platform imitating an autonomous car was completed.

However, there is no software implementation that actually allows students to operate the platform, because there are only programs used for experiments. In the future developments, it is necessary to develop a support system for autonomous driving technologies so that people can coexist with autonomous cars. Therefore, using the platform, it is necessary to verify and consider what kind of software configuration and how to educate will be conducted in the educational field.

A society that is the shortage of various types of engineers can be imagined easliy. We wonder if autonomous cars run frequently. We wonder if Japanese companies will increase to. If the future will be a society that live together with autonomous cars and people, the economic effects of the Fourth Industrial Revolution are immeasurable. We will develop a tool that assiste to be able to realize the dream of automatic driving and become a rich country.

VI. FUTURE OUTLOOK

A. Open-source using GitHub

Currently, only data of the manufactured circuit boards and ROS nodes are released to GitHub. The reason why the authors have not been able to release other bodies and existing products is that we have not been able to reformat the data. Therefore, the data will be reshaped, and an environment that anyone can use will be constructed.

B. Circuit boards design

There are five types of circuit boards created and it is troublesome when created. Therefore, it is necessary to reduce circuit boards for easily making. In addition, mounting work is difficult without experience in soldering, because chip components mounted on circuit boards are small. Therefore, we think that it is necessary to change into large parts so that anyone can easily mount its. If we prepare a simple lecture to explain in a video, we will design circuit boards that anyone can implement easily.

C. Platform

In order to use the one-tenth size model car as a platform, there are conditions that you can program using C ++ and python and so on. Therefore, generally the authors think that the target that can use it is more than university students. Our ultimate goal is an autonomous driving technologies platform for students of various grades. In the future developments, we are plannning to develop an Android app or Scratch [9] based on ROS. They will be visual programming language to use the platform easily. Therefore, with the platform, beginners can learn simple autonomous driving technologies with the app or scratch only, and experts can learn autonomous driving technologies using C ++, python and java on ROS.

VII. ACKNOWLEDGEMENTS

The authors are grateful to Professor Hiroyuki Kobayashi for this study.

REFERENCES

[1] Cabinet Office:“Responding to new industrial changes” https://www5.cao.go.jp/keizai3/2016/0117nk/n16_2_1.html, February 2018.

[2] Ministry of Economy, Trade and Industry Business Information Policy Bureau, IT Innovation Division and Local Informatization and Human Resource Development Office:“Importance of IT personell resource development and security personnel resources for the Fourth Industrial Revolution”, February 2018.

[3] Nagoya University, Nagasaki University and National Institute of Advanced Industrial Science and Technology, “Autoware” https://www.autoware.ai/, December 2017.

[4] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger and Rob Wheeler Andrew Ng, “ROS: an open-source Robot Operating System”, June 2009.

[5] Hiromichi Nozaki, “Basic automotive engineering”, Tokyo Denki University Press, 2008, pp.13-14

[6] Open Source Robotics Foundation, Inc., ““Open Robotics” https://www.openrobotics.org

[7] G. Servin, “ROSSerial mbed for Kinetic Distribution” https://os.mbed.com/users/garyservin/code/ros_lib_kinetic/, December 2016.

[8] InvenSense Inc., “MPU-6000 and MPU-6050 Product SpecificationRevision 3.4” https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V34.pdf, August 2013.

[9] Guilherme Affonso, “Interfacing ROS with Scratch” http://wiki.ros.org/scratch, June 2019.