An Eco-Friendly, Autonomous Beach-Cleaning Robot based on path planning

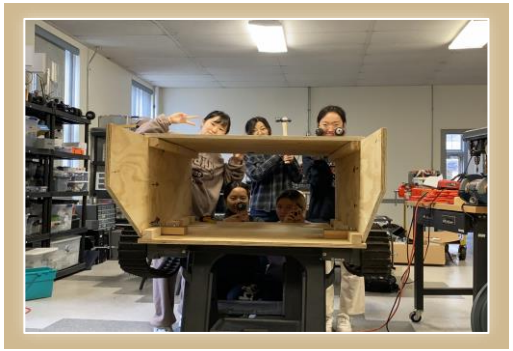
Team C.C

Eunmin Kim, Jeeyoung Oh, Booyong Kim, Seoyeong Lee, Hanbyeol Lee, Caleb Ikallna

Dankook Univ., Chung-ang Univ., Sangmyung Univ., Daegu Catholic Univ., Chung-ang Univ., Purdue Univ.

PURDUE UNIVERSITY | Polytechnic Institute

Team C.C



Team C.C

Eunmin Kim
Dankook Univ.

Seoyeong Lee
Daegu Catholic Univ.

Hanbyeol Lee
Chung-ang Univ.

Jeeyoung Oh
Chung-ang Univ.

Booyong Kim
Sangmyung Univ.

Caleb Ikalina
Purdue Univ.

PURDUE UNIVERSITY | Polytechnic Institute

Contents

1. Introduction	2. Methodology	3. Implementation	4. Conclusion
- Introduce - Expect result	- Structure - Diagram - Development Environment	- Communication - Detection - Driving	- Criteria of Test - Problem - Future Plan

“Autonomous Beach Cleaning roBOT”

Introduction

Why ‘ABC Bot’?

- Environment

  => World will have 710 million metric tons of plastic that will pollute the environment by 2040 [1]

- Injured people

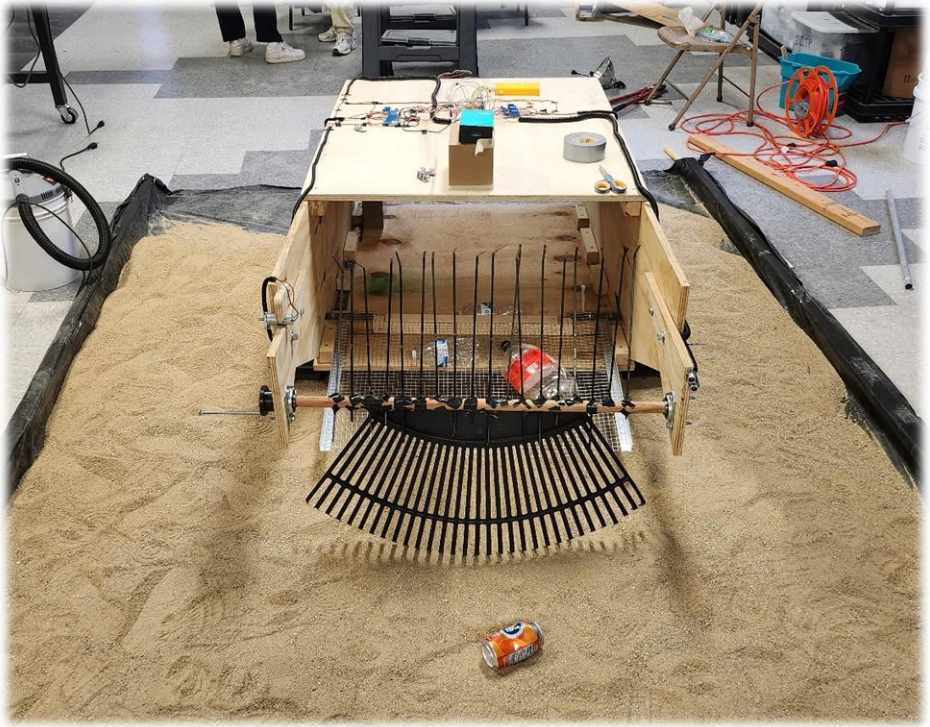
  => Multiple labeled garbage collection dating back 30 years, as well as broken glass and toilet-related waste[2]

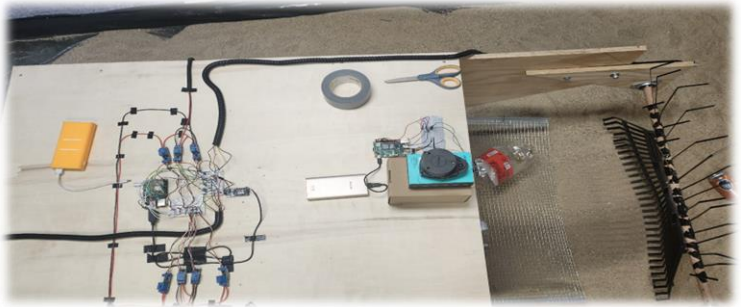
- Tourism

  => Stranded litter may potentially reduce local tourism income by 39.1%, representing losses of up to US$ 8.5 million per year [3]

Introduction

About ‘ABC Bot’





Introduction

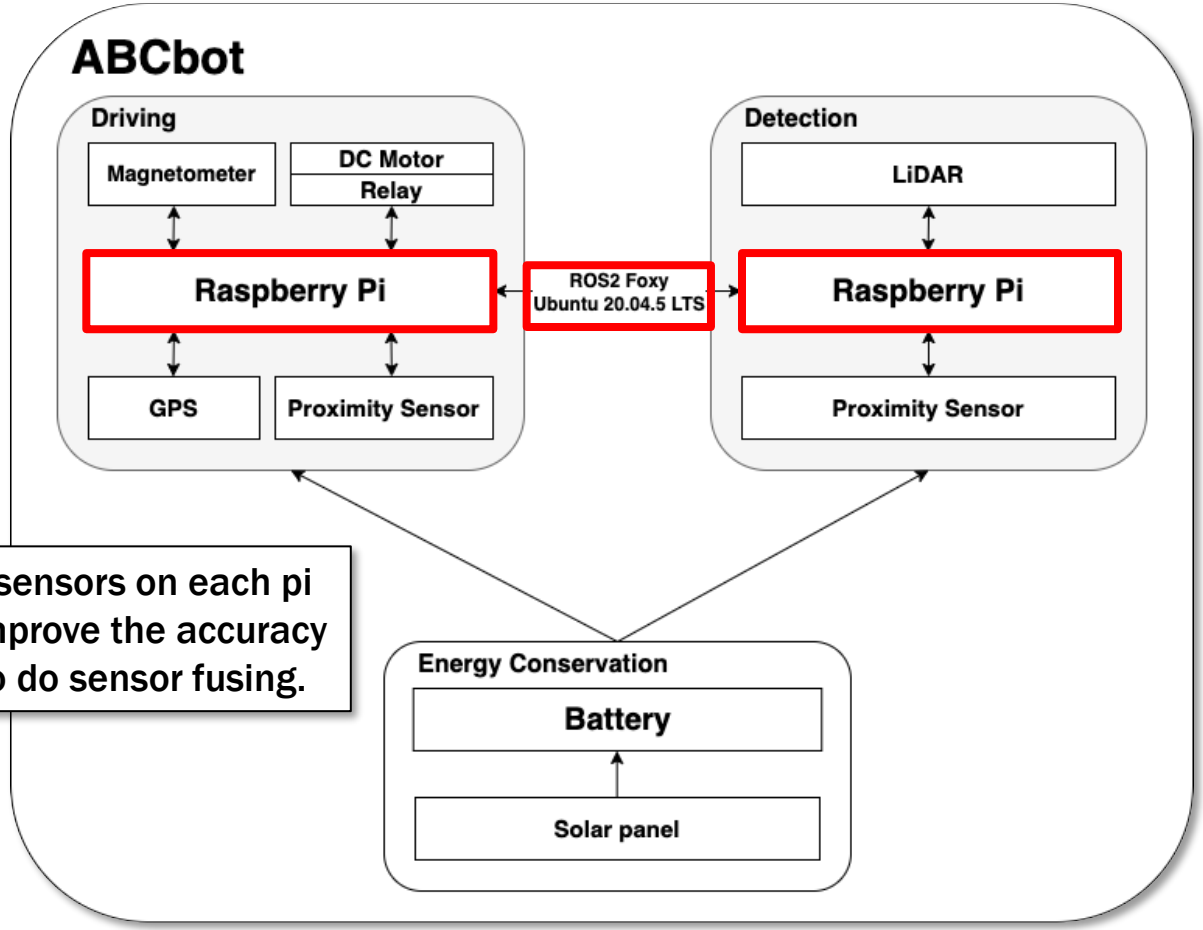
About ‘ABC Bot’

- Drive automatically
- Path planning based on GPS tracking
- Utilizes LiDAR to avoid obstacles
- Detect with proximity sensors and a camera for detection and movement system

➢ Keeping the beaches clean
without labor & saving the financial costs

Methodology

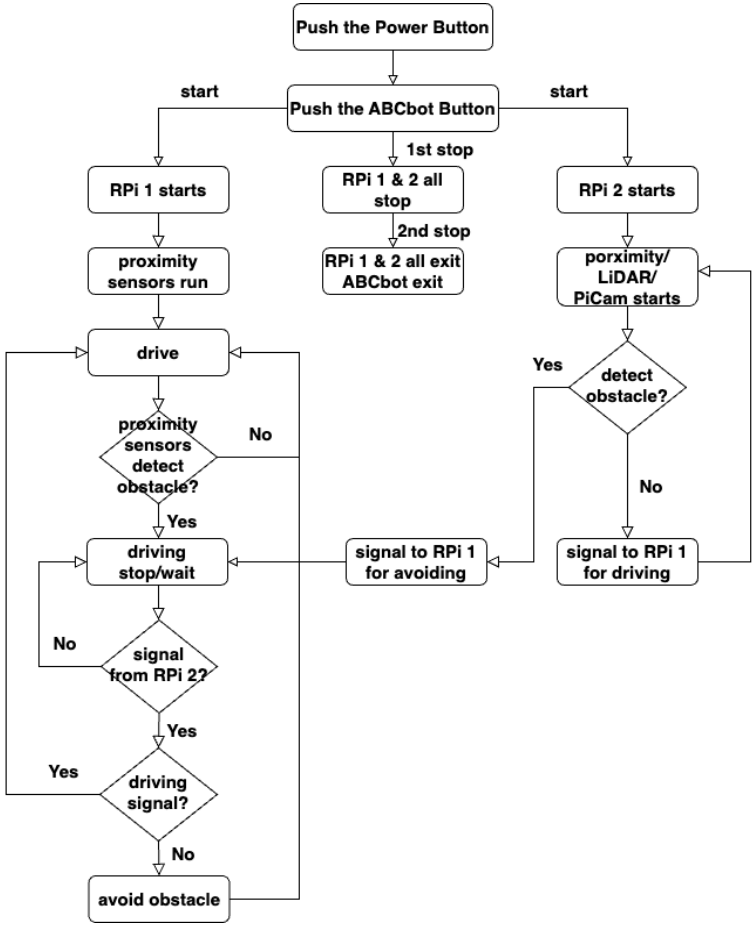
Structure



There are lots of sensors on each pi and we want to improve the accuracy => we decided to do sensor fusing.

Methodology

Flow Diagram



Methodology

Development Environment

	Version
Raspberry Pi OS	Ubuntu Server 20.04.5 LTS (64-bit)
ROS	ROS 2 foxy
Python version	3.8.10

Implementation - Communication

Why ROS2?

- ✓ Investigating various ways for communication
  => Decided to use the robot software platform

1. Highly reusable
   => Considering ROS can be managed as a package,
   can focus on what we want to develop
   and use the package for the rest

2. Communication may be easily performed through
   the node package

Implementation - Communication

Why ROS2?

- Famous and generally used in robotics

  => there are lots of references

- Real-time processing

Implementation - Communication

How? [4]

- Workspace > two packages & one interface file
- Pacakge > publisher/subscriber nodes
  => operate through a 'Topic' for message delivery

  Used as an ROS communication channel for messages between nodes
- Multiple nodes
  - rplidar_node
  - obstacles_detect_node
  - motor_subscriber node

  => all connected via sensor fusing

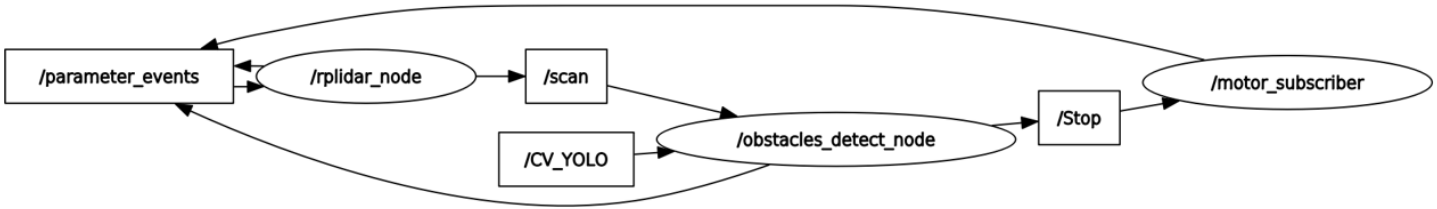
Implementation - Communication

How?

- Combines the data obtained from each sensor
  => supplement the parts that cannot be achieved with a single sensor

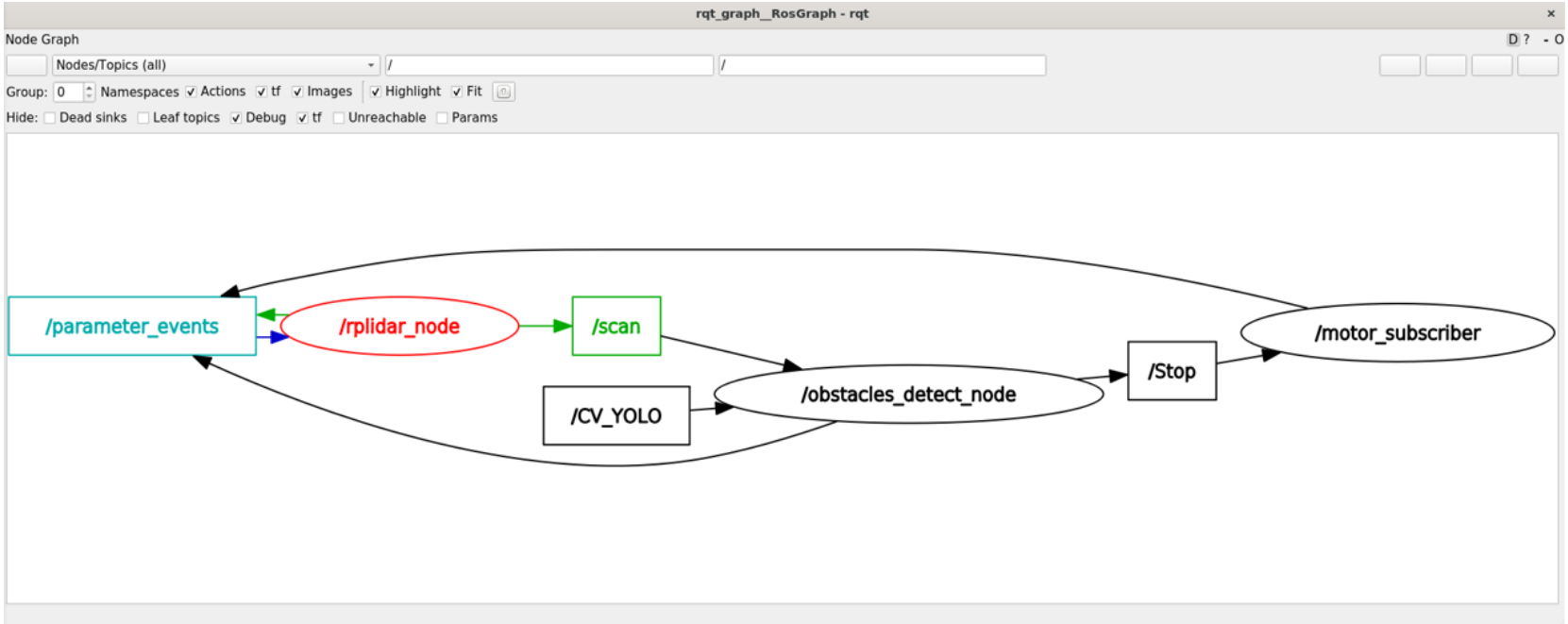
➢ Perform a complex and autonomous function and reduce errors of existing sensors

➢ Sensors were fused by sending information to the motor to control the direction of wheels



Implementation - Communication

How?



▲ rqt_graph

Implementation - Driving

Driving

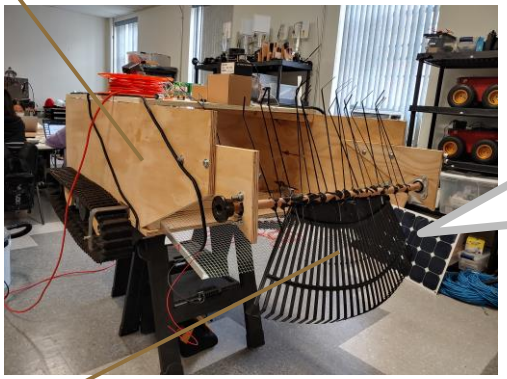
- What the self-driving robots need
  - Controlling movement
  - Localization
  - Path planning
  - Getting surrounding information

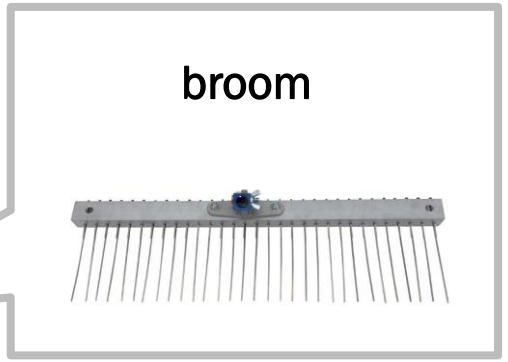
    detection, sensor fusing, computer vision and etc.

Implementation - Driving

Control

driving





cleaning tool

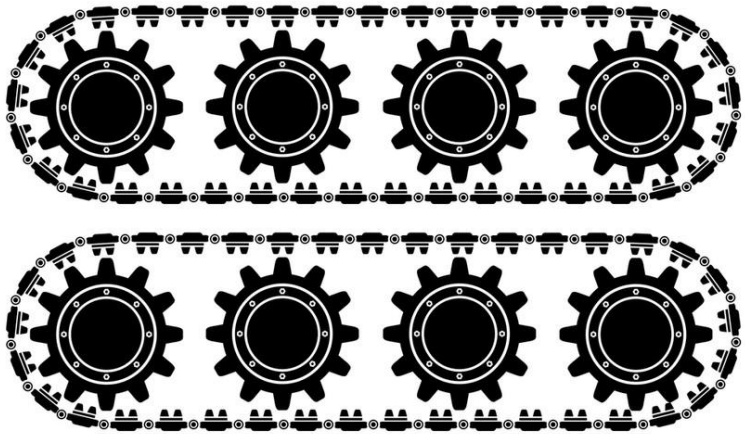
Implementation - Driving

Broom Control

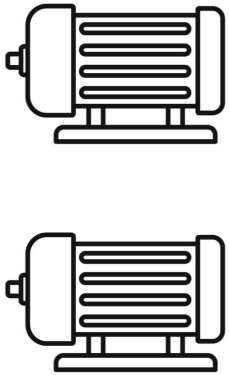
Implementation - Driving

Control

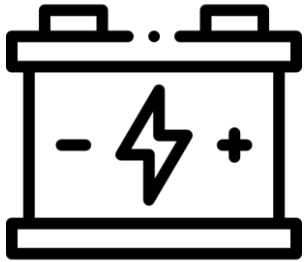
caterpillar wheel



12V DC motor

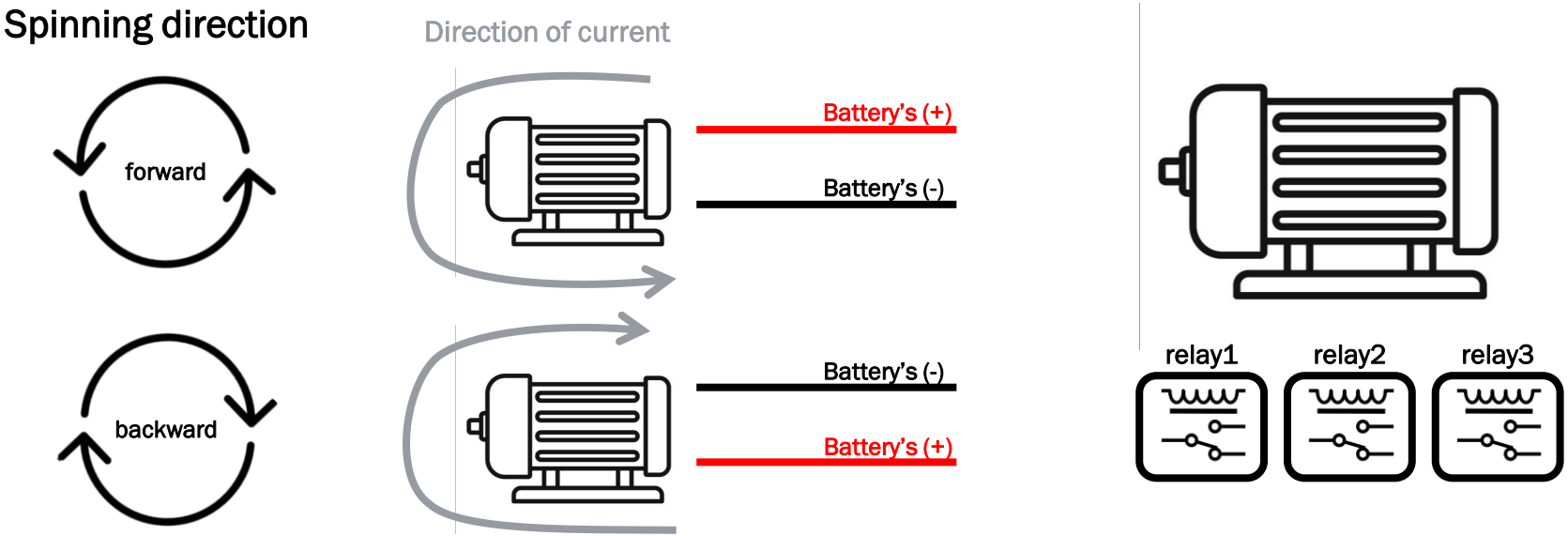


12V battery



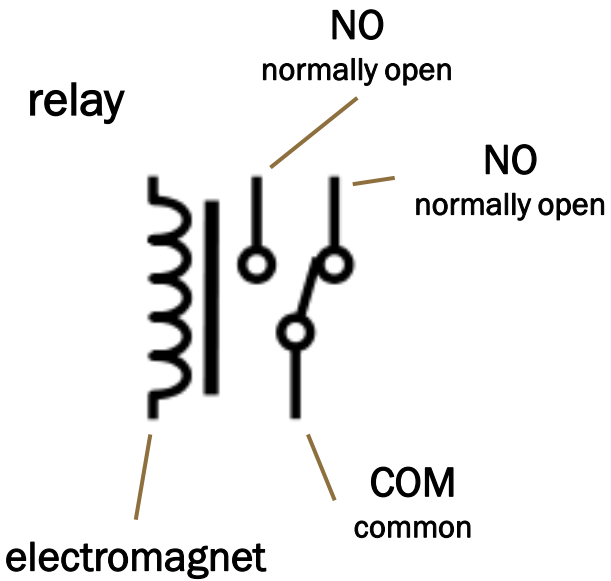
Implementation - Driving

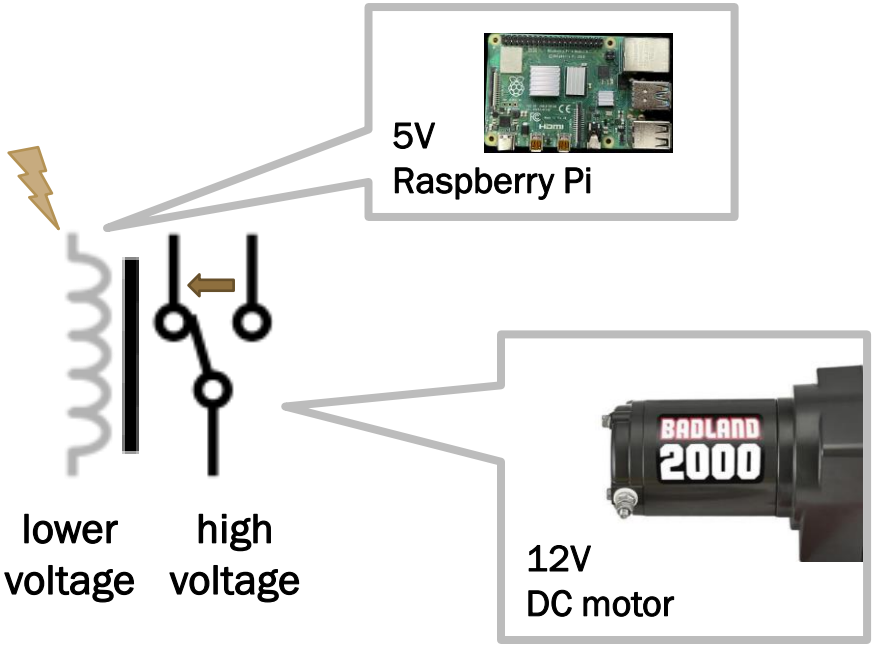
Control



Implementation - Driving

Control

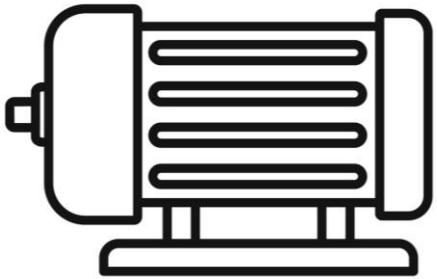


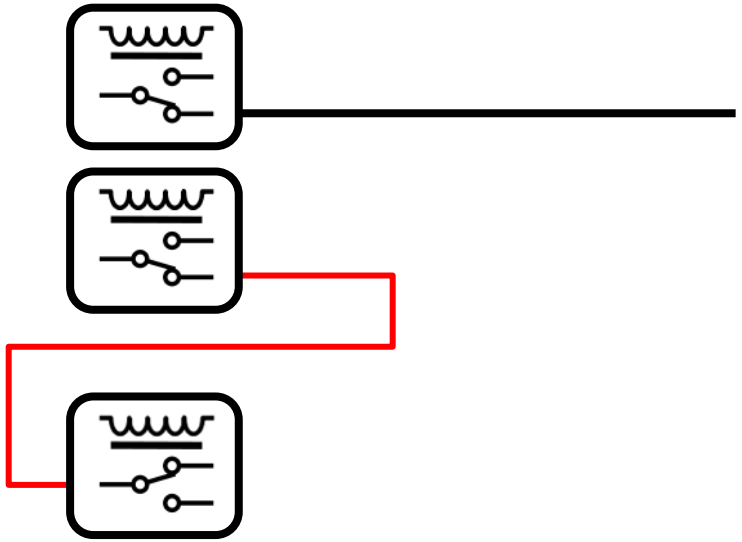


Implementation - Driving

Control

wheel motor

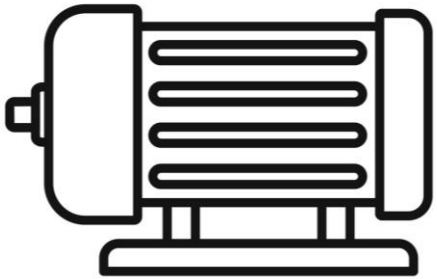


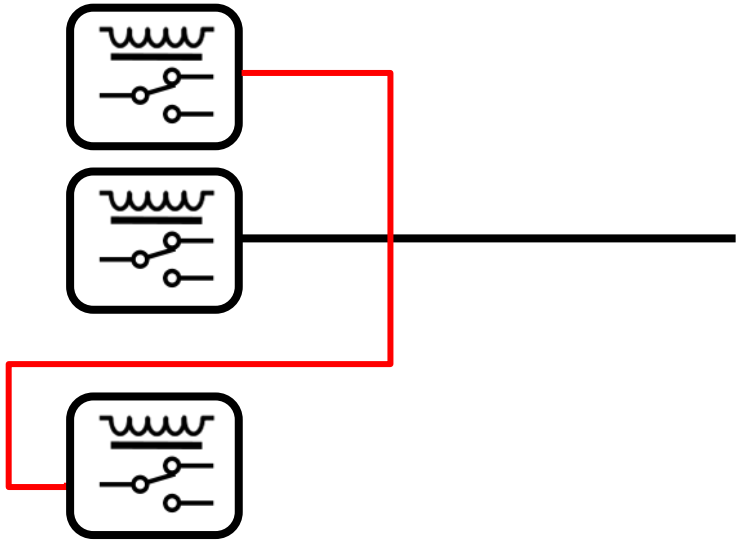


Implementation - Driving

Control

wheel motor

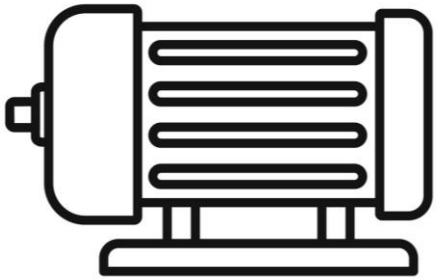


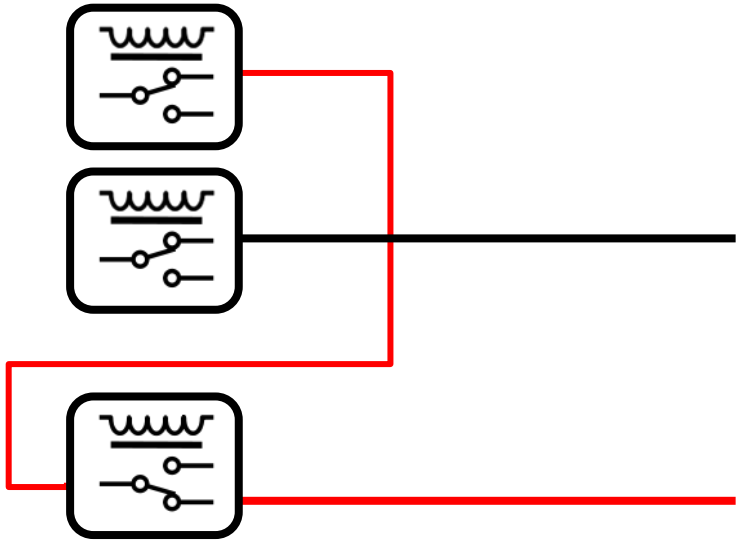


Implementation - Driving

Control

wheel motor

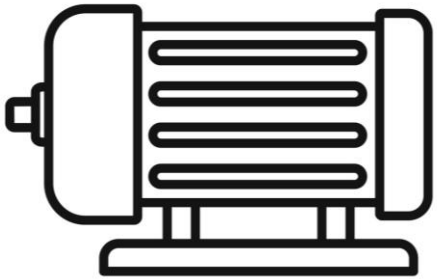




Implementation - Driving

Control

broom motor



Implementation - Driving

Control

DC motor left			DC motor right			broom motor	left caterpillar wheel	right caterpillar wheel	driving direction	broom spin
relay 1	relay 2	relay 3	relay 1	relay 2	relay 3	relay				
C	C	C	C	C	C		forward	forward	**forward**	
0	0	C	0	0	C		backward	backward	**backward**	
0	0	C	C	C	C		backward	forward	**turn right**	
C	C	C	0	0	C		forward	backward	**turn left**	
		0			0		stop	stop	**stop**	
						0				**stop**
						C				**run**

Implementation - Driving

Control

- How the robot chooses its action
  - Path
  - Interruptions by the detection unit
  - Surrounding obstacles
  - Amount of the collected trash

Implementation - Driving

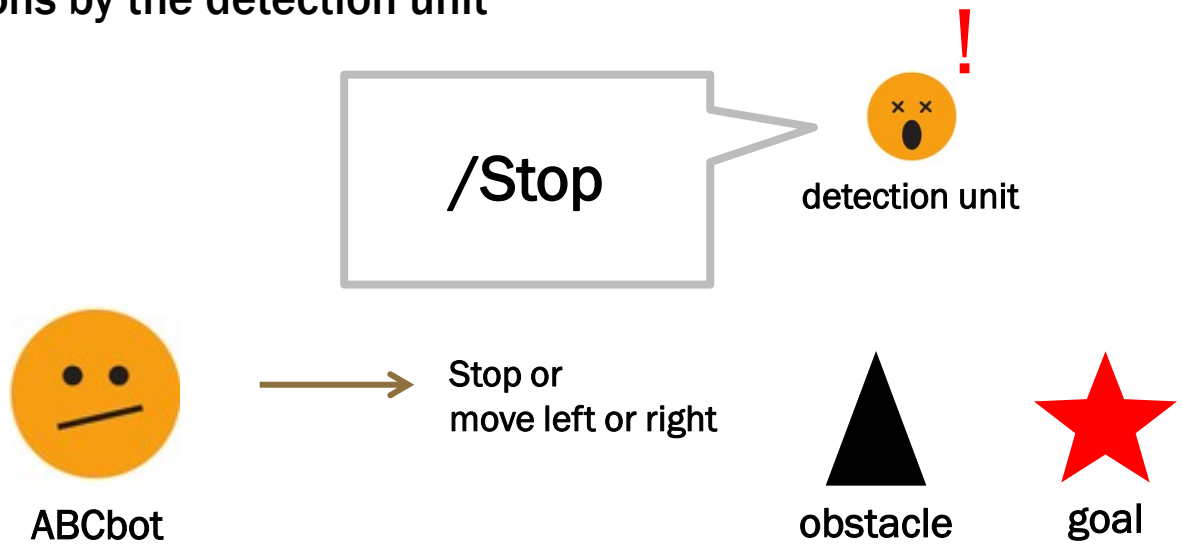
Control

- Path

Implementation - Driving

Control

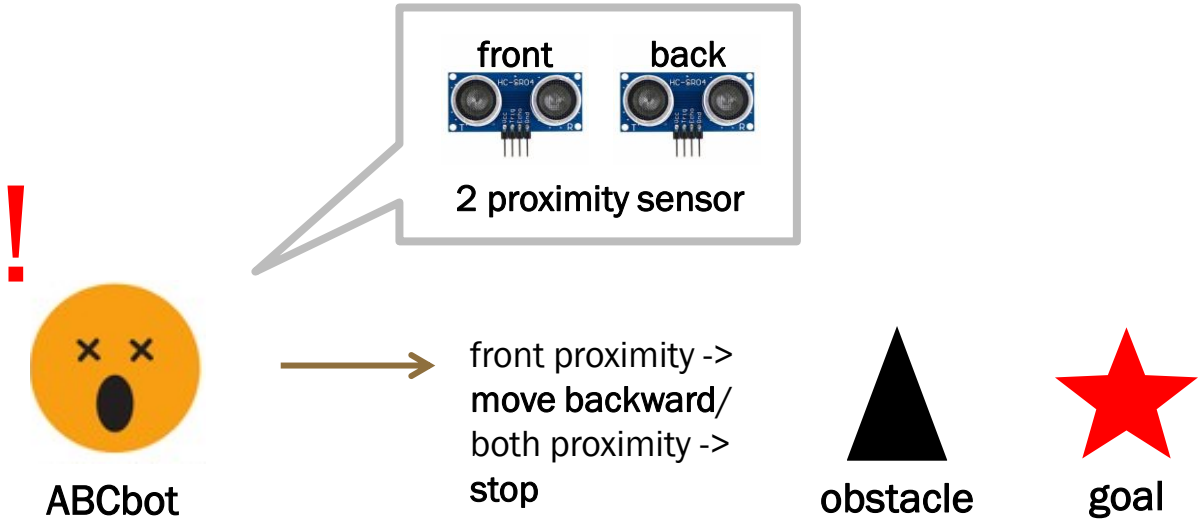
- Interruptions by the detection unit



Implementation - Driving

Control

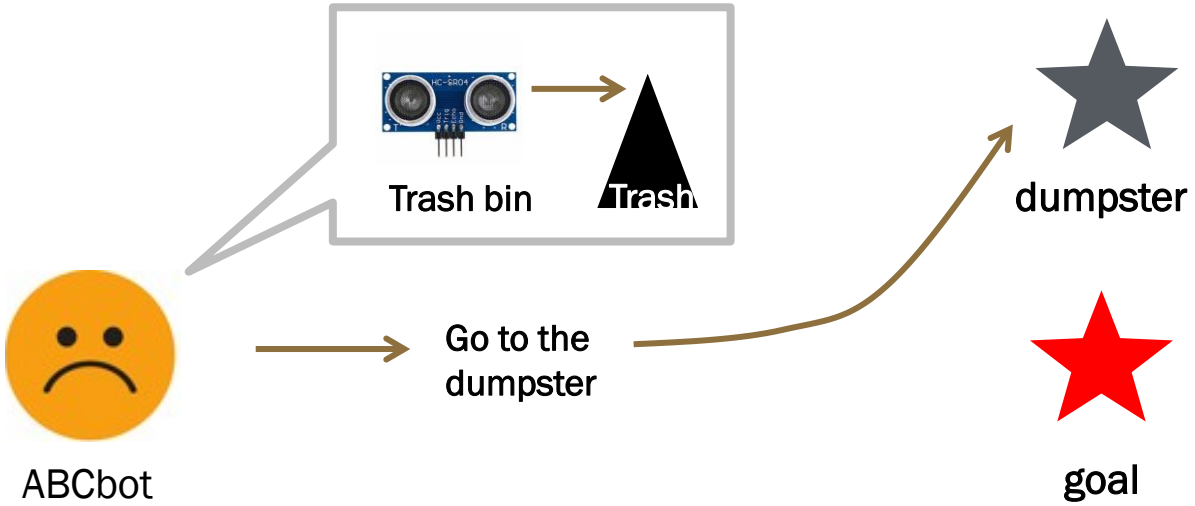
- Surrounding obstacles



Implementation - Driving

Control

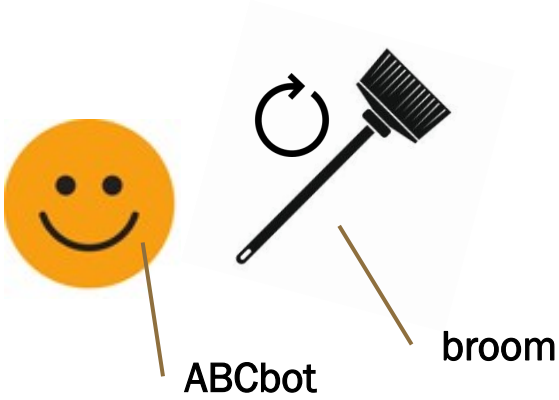
- The amount of collected trash



Implementation - Driving

Control

- Broom



When it is going to a goal

When it is avoiding obstacles or going to a dumpster

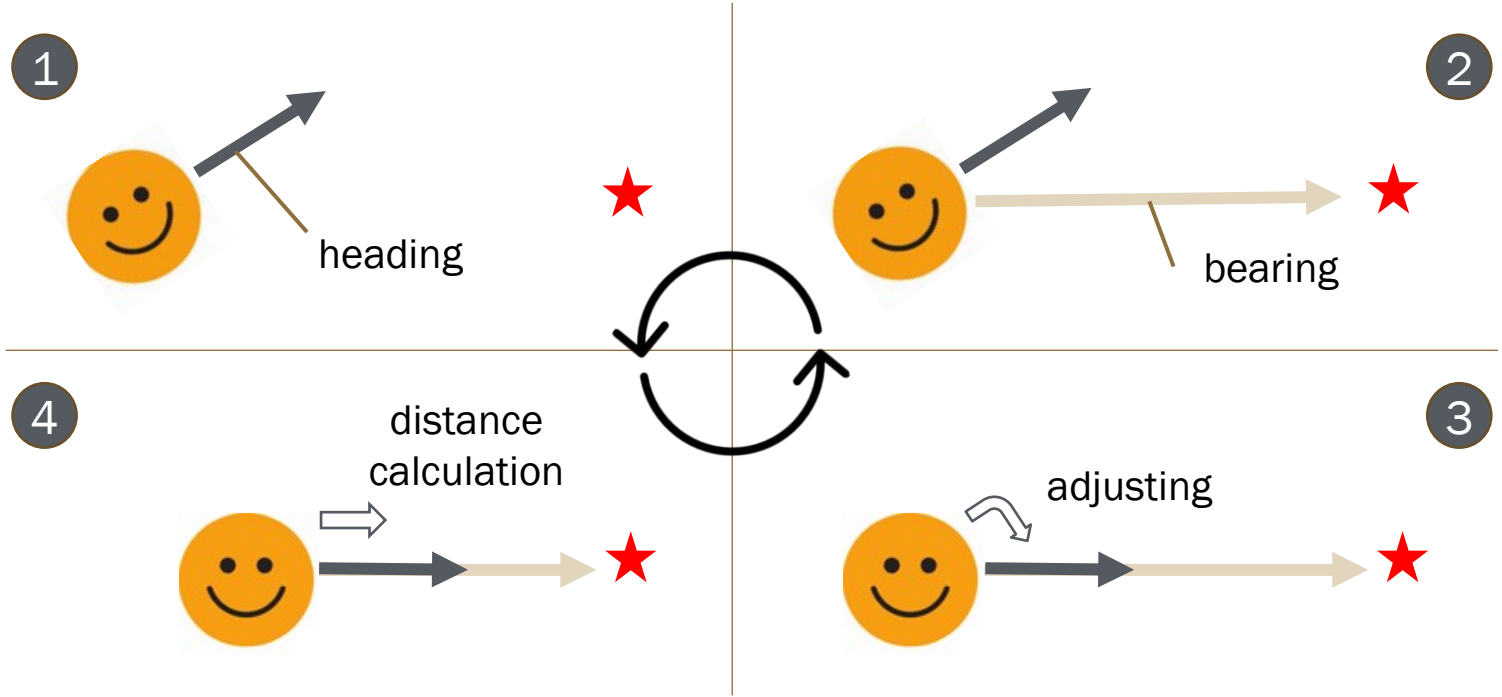
Implementation - Driving

Localization

- GPS(Global Positioning System)
  - global navigation satellite systems
  - The location is denoted by a latitude and a longitude.

Implementation - Driving

Localization

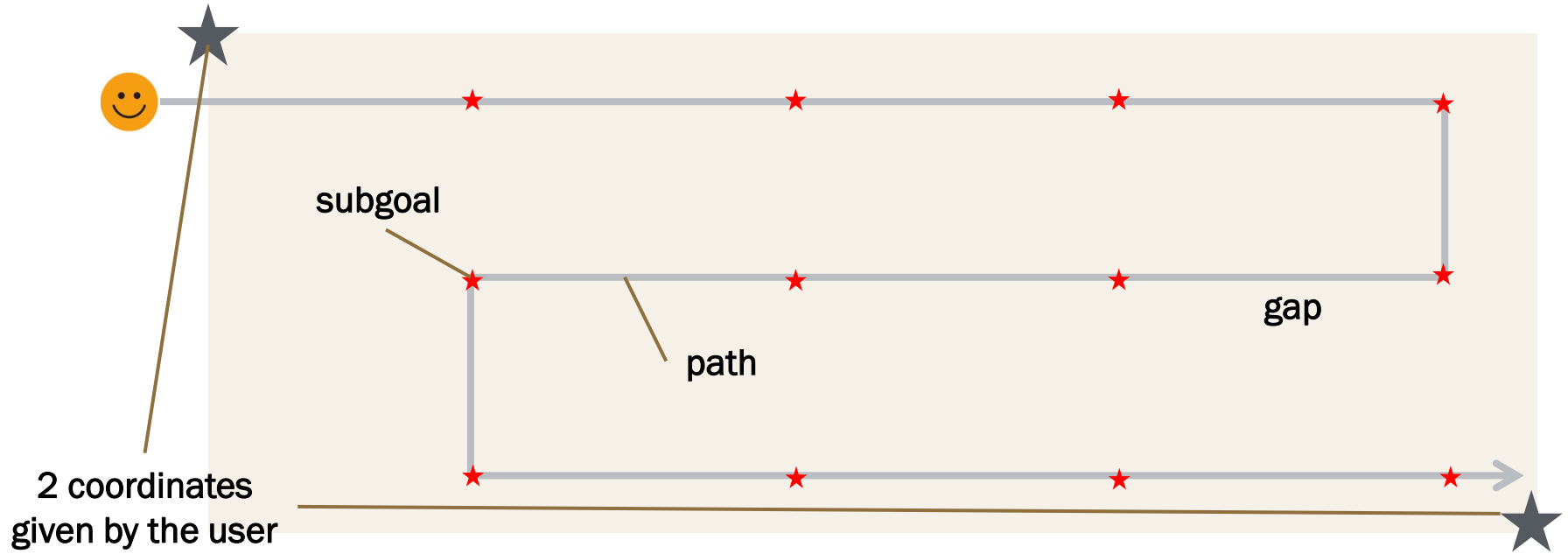


Implementation - Driving

Path planning

Implementation - Driving

Path planning

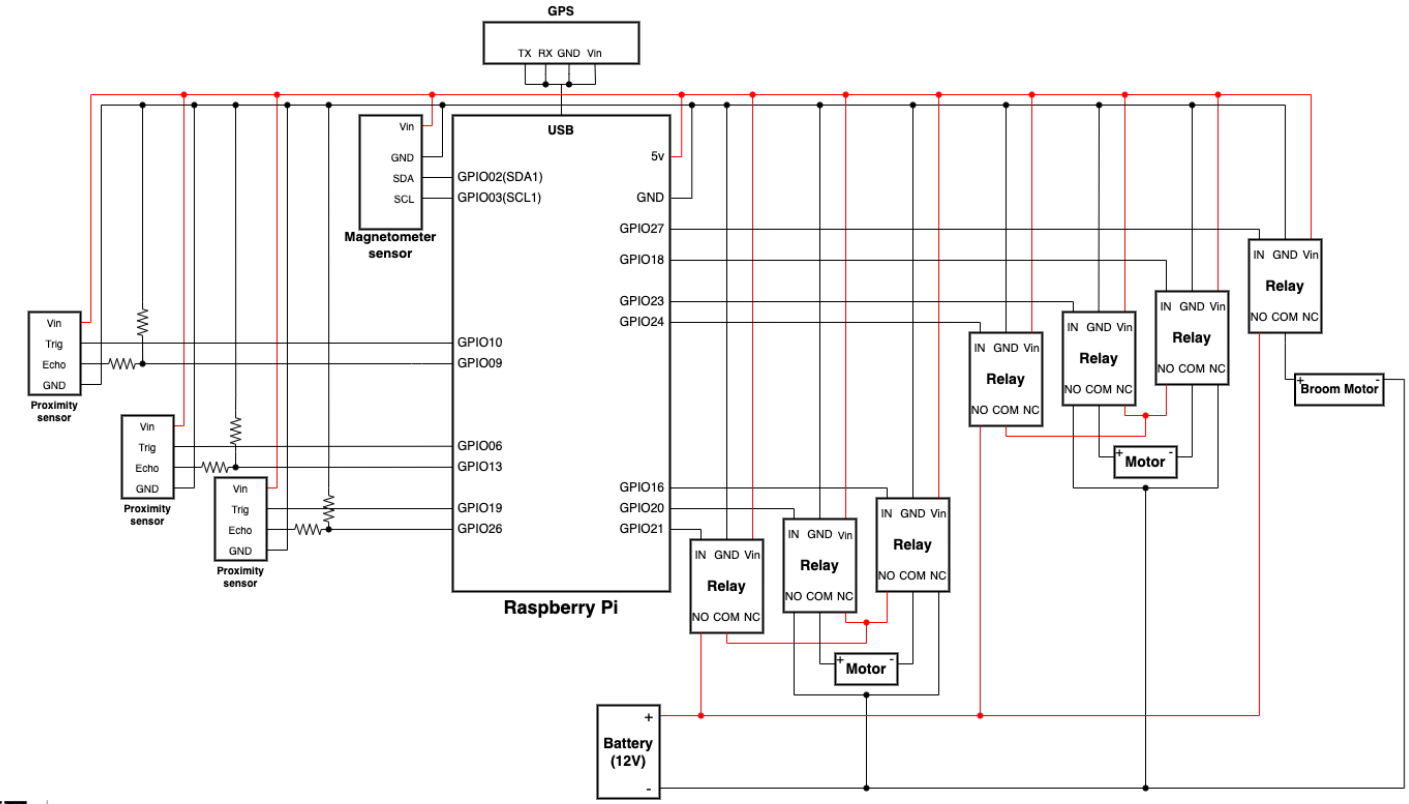


Implementation - Driving

Path planning

Implementation - Driving

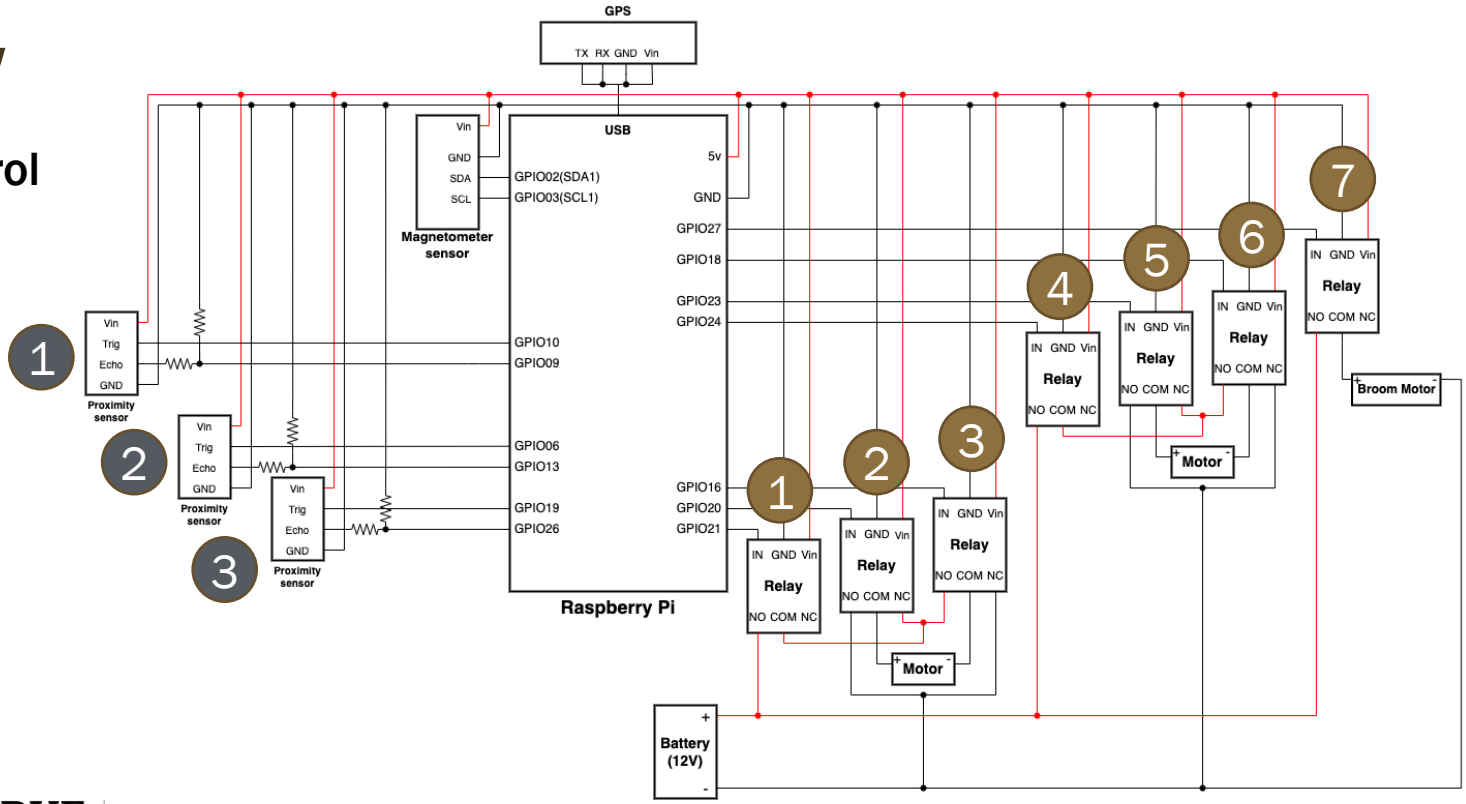
Summary



Implementation - Driving

Summary

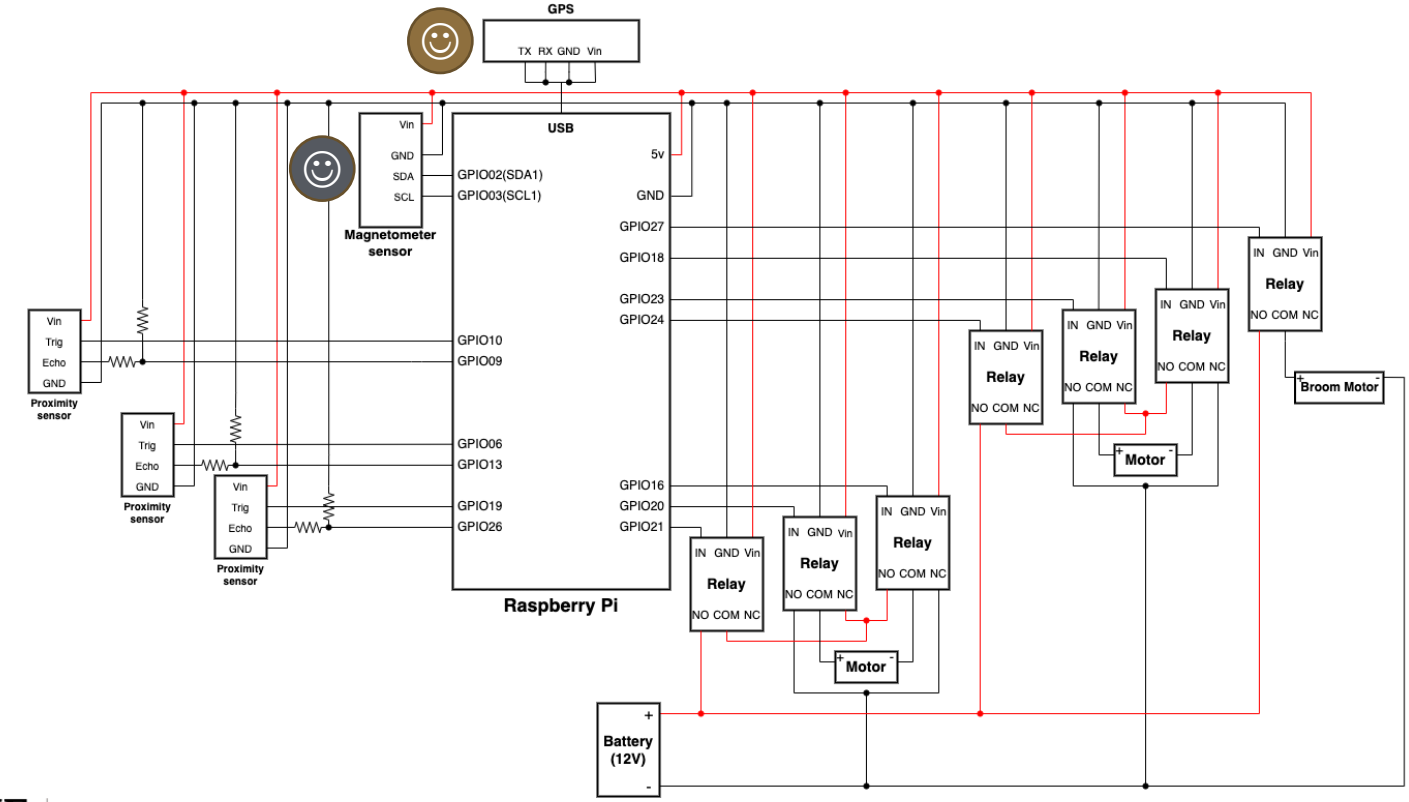
- control



Implementation - Driving

Summary

- path



Implementation - Detection

Implementation - Detection

Detection

Implementation - Detection

Detection

- 2D LiDAR

- Power Supply: 5V
- Detection Range: 12 meters
- Sample Rate: 8,000 Samples/s
- Angular Resolution: 1 degree

- Data are within Full 360-degree using x and y axis cartesian coordinates

Implementation - Detection

Detection

- 2D LiDAR Algorithm

1. Send data from the obstacles around the LiDAR
2. Subscribe /scan topic
3. Preprocessed the range of distance points set (120 degrees)
4. Divide into 3 parts: left, front, and right side
5. The distance of the scanning point is added to each of the parts
6. Compare the 3 parts

Implementation - Detection

Detection

- Proximity Sensor

- HC-SR04 Ultrasonic Module Distance Sensor
- HC-SR04 Power Supply : 5V DC; Quiescent
  - Current : <2mA,; Effectual Angle: <15°
  - Detection Distance: 2cm~500 cm

Implementation - Detection

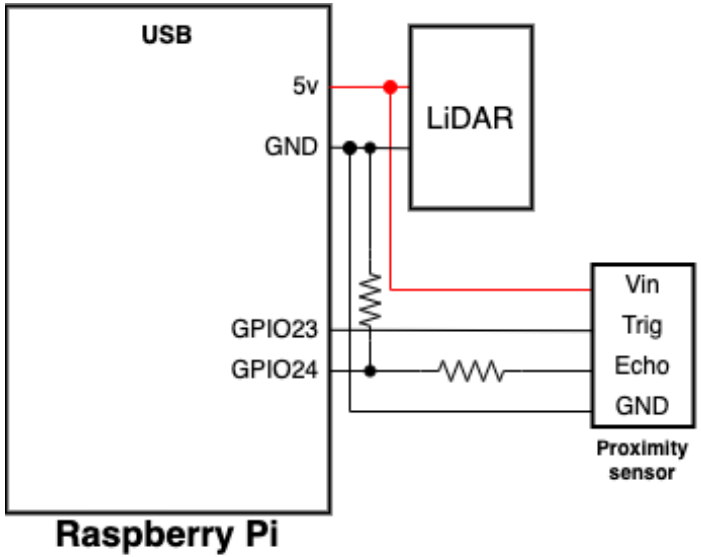
Detection

- Proximity Sensor Algorithm

  1. Execute proximity sensor

     (ultrasonic waves)

  2. Return data to GPIO



Implementation - Detection

Detection

- Raspberry Pi Camera Module V2 8-megapixel sensor
  3280 * 2464 1080p V2.1
- Robot vision to recognize entities on the z-axis that the 2D LIDAR cannot detect

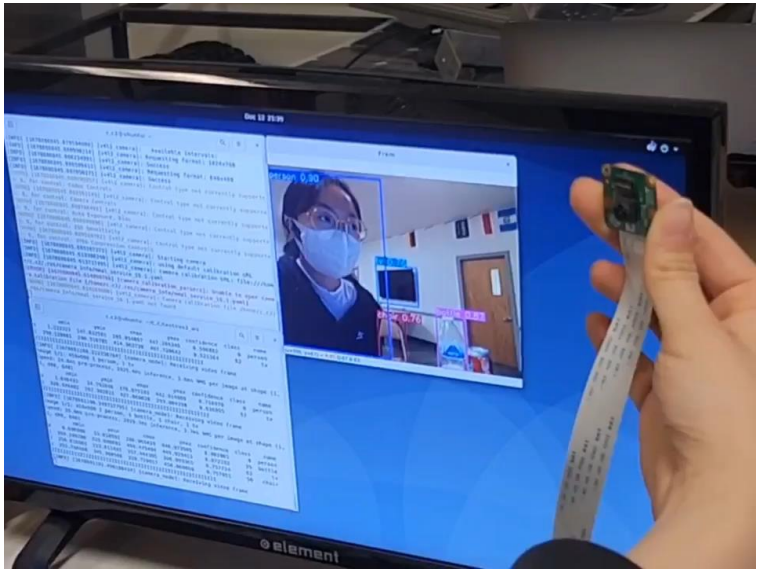
- Image Sent through ROS2

- Pre-trained YOLO v5 and YOLO v7 are used for analyzing images taken by the Pi Camera.

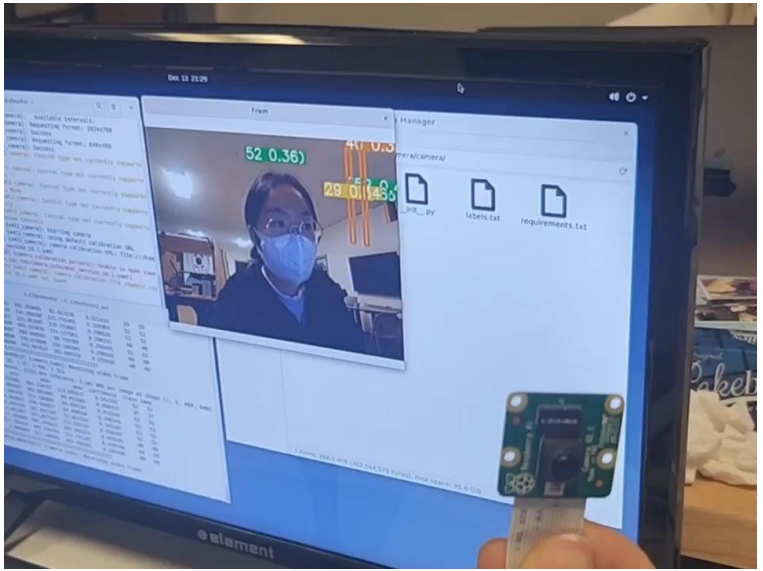
Implementation - Detection

Detection

- Real-time Object Detection Demo



▲ YOLO v5



▲ YOLO v7

Implementation - Detection

Detection Final Algorithm - Sensor fusing and Voting system

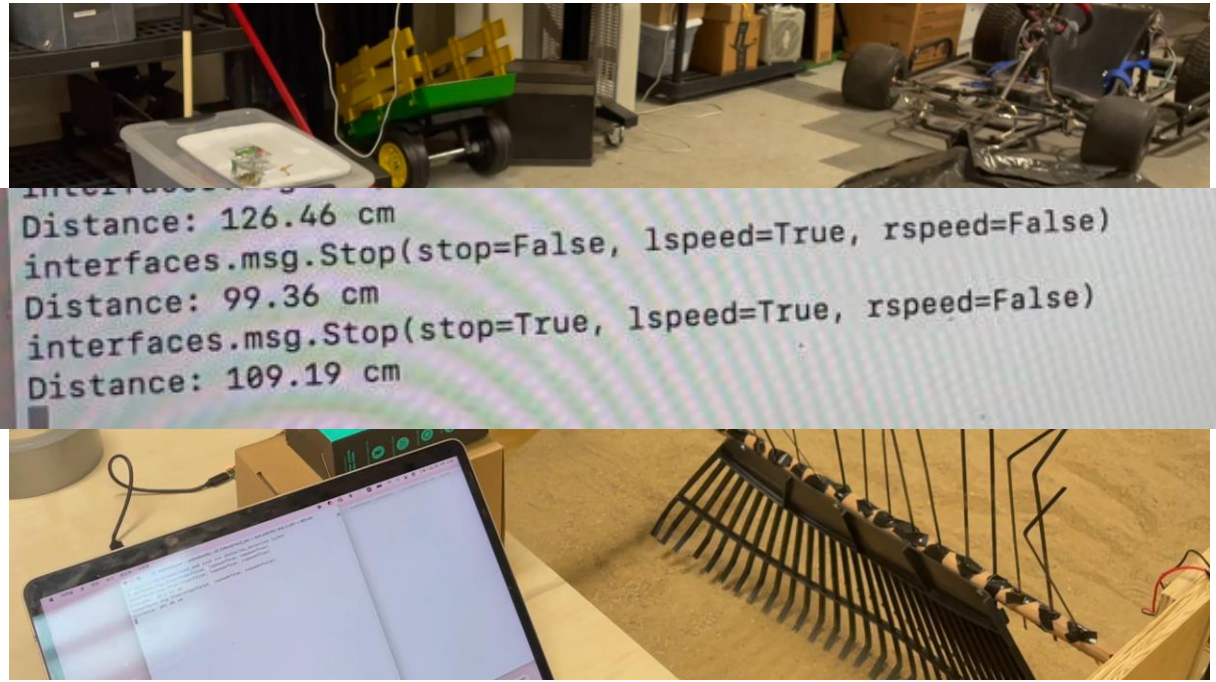
1. Subscribe /scan topic
2. left=20°~60°, front=0~30°, 330~360°, right=300°~340°
3. voting points within three parts

   that do not exceed the threshold distance
4. execute proximity sensor
5. if proximity sensor detects shorten distance

   stop = true

Implementation - Detection

Detection Demo

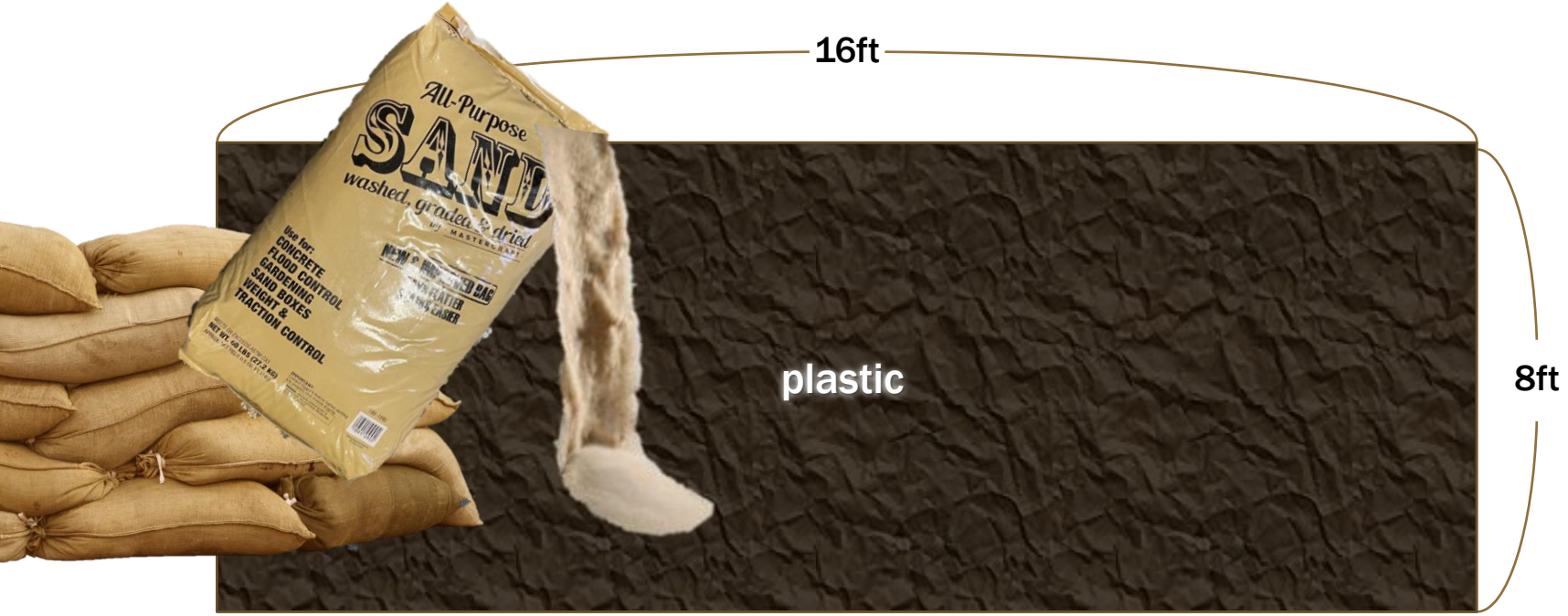


Conclusion

ABCbot

Conclusion

Test Criteria



Conclusion

Demo

Conclusion

Challenge

1. raspberry pi hw(CPU, PiCamera) -> pytorch, torchvision downgrade
   => image transmission was delayed and worked slowly
   => Jetson Nano and High-Quality Camera

2. In a very short time of four months,
   => had to build everything
   from robot basics to software

3. Sensor Quality Problem
   => proximity sensor, motor, PiCamera etc

Model	size	objects	mAP	Jetson Nano 1479 MHz	RPi 4 64-OS 1950 MHz
NanoDet	320x320	80	20.6	26.2 FPS	13.0 FPS
NanoDet Plus	416x416	80	30.4	18.5 FPS	5.0 FPS
YoloFastestV2	352x352	80	24.1	38.4 FPS	18.8 FPS
YoloV2	416x416	20	19.2	10.1 FPS	3.0 FPS
YoloV3	352x352 tiny	20	16.6	17.7 FPS	4.4 FPS
YoloV4	416x416 tiny	80	21.7	16.1 FPS	3.4 FPS
YoloV4	608x608 full	80	45.3	1.3 FPS	0.2 FPS
YoloV5	640x640 small	80	22.5	5.0 FPS	**1.6 FPS**
YoloV6	640x640 nano	80	35.0	10.5 FPS	2.7 FPS
YoloV7	640x640 tiny	80	38.7	8.5 FPS	2.1 FPS
YoloX	416x416 nano	80	25.8	22.6 FPS	7.0 FPS
YoloX	416x416 tiny	80	32.8	11.35 FPS	2.8 FPS
YoloX	640x640 small	80	40.5	3.65 FPS	0.9 FPS

Conclusion

Future Plan

- Improve accuracy in terms of Path planning and Detection
- Enough Indoor & outdoor tests
- If others want stronger hw,

  they can just change the robot base with existing sw.

References

[1] H. Regan. "World will have 710M tons of plastic pollution by 2040 despite efforts to cut waste, study says." CNN.com. \url{https://www.cnn.com/2020/07/23/world/plastic-pollution-2040-study-intl-hnk/index.html} (accessed Sept. 27, 2022).

[2] P. HOARE, "Broken glass proving the scourge of Cork's beaches," Irish Examiner, Aug. 2021. Accessed: Sept. 26, 2022. [Online]. Available: \url{https://www.irishexaminer.com/news/munster/arid-40353995.html}

[3] A. PaulKrelling, A. Thomas, and A. Turra, “Differences in perception and reaction of tourist groups to beach marine debris that can influence a loss of tourism revenue in coastal areas,” Marine Policy, vol. 85, pp. 87-99, Nov. 2017, doi: \url{https://doi.org/10.1016/j.marpol.2017.08.021}

[4] Open Robotics. "Writing a simple publisher and subscriber (Python)" docs.ros.org \url{https://docs.ros.org/en/foxy/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Py-Publisher-And-Subscriber.html} (accessed Oct. 12, 2022).

Q&A | Thank you

Eunmin Kim, Jeeyoung Oh, Seoyeong Lee, Booyong Kim, Hanbyeol Lee, Caleb Ikalina

Team C.C

K-SW 2022 Fall Program

kim4153@purdue.edu, oh310@purdue.edu, lee4490@purdue.edu, kim4162@purdue.edu, lee4487@purdue.edu, c.ikalina@purdue.edu

polytechnic.purdue.edu

/ TechPurdue

PURDUE UNIVERSITY | Polytechnic Institute