

캡스톤 디자인(2)

진행 상황

Motus+er

목차

1

시스템 구조도

2

통신 및 네비게이션

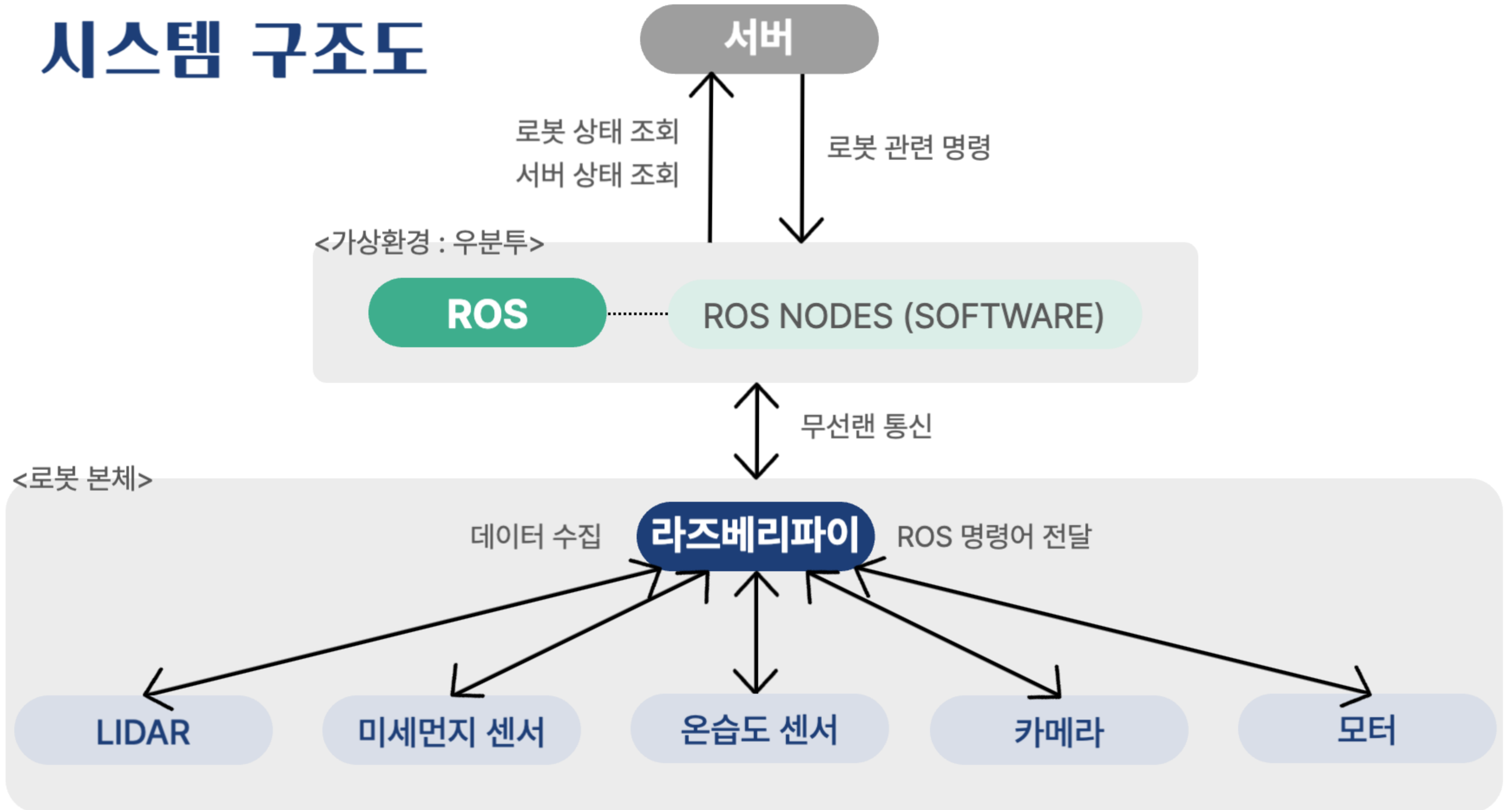
3

공기 데이터

4

서버 통신 및 UI

시스템 구조도



통신 및 네비게이션

문제점



통신 문제

보안 기능을 해제하여 아이폰으로
핫스팟을 사용하여,
라즈베리파이와 우분투가 통신이
되지 않는 문제

해결



- 라즈베리파이의 모든 패키지를 다시 설치하여 오류를 해결
- 아이폰 핫스팟과 연결하지 않기로 함

다른 문제점



로봇에 값을 입력하는 문제
(URDF등을 이용한 RVIZ 시각화 과정)

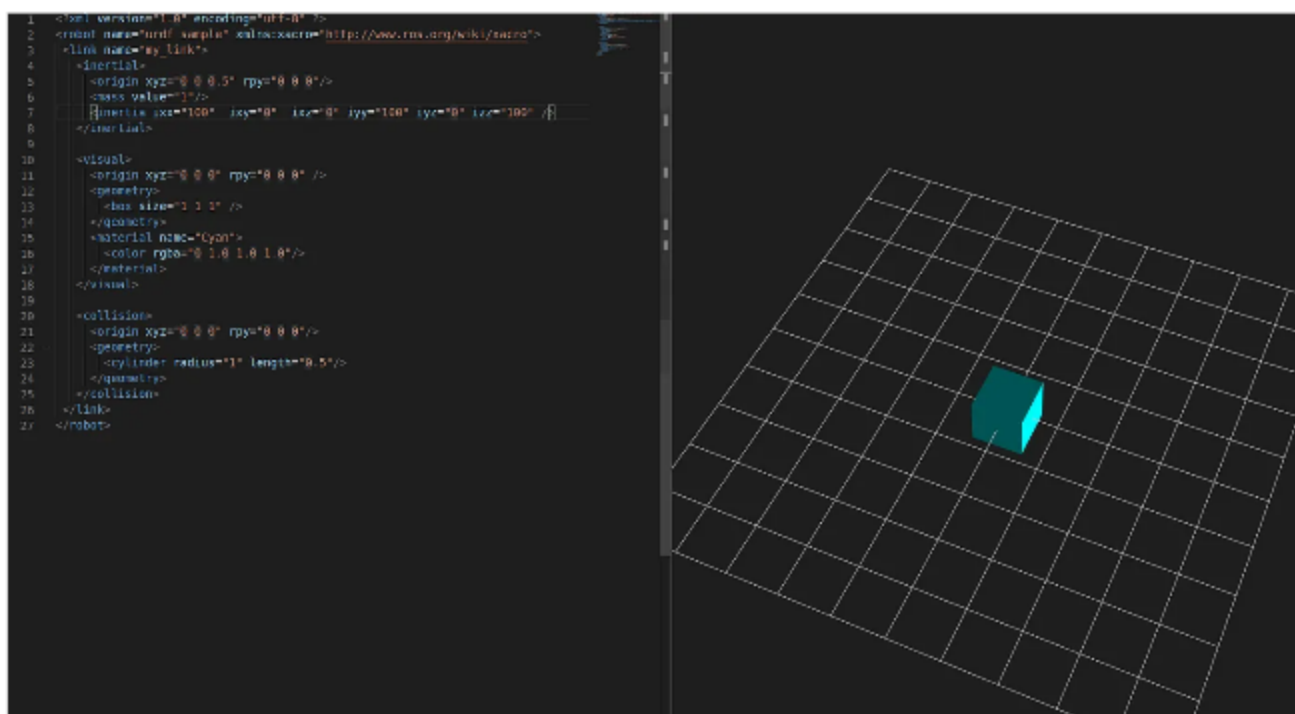
진행 상황

기존에는 모두 라이다 센서를 통한 값으로 매핑 및 시각화가 이루어졌다.
로봇의 움직임과 네비게이션을 진행하기 위해서 다음과 같은 파일과 패키지가 필요하다.

1

URDF

로봇 자체를 RVIZ에 시각하기 위해 필요



로봇 모델에 대한 정보들을 명세해 놓은 규격서

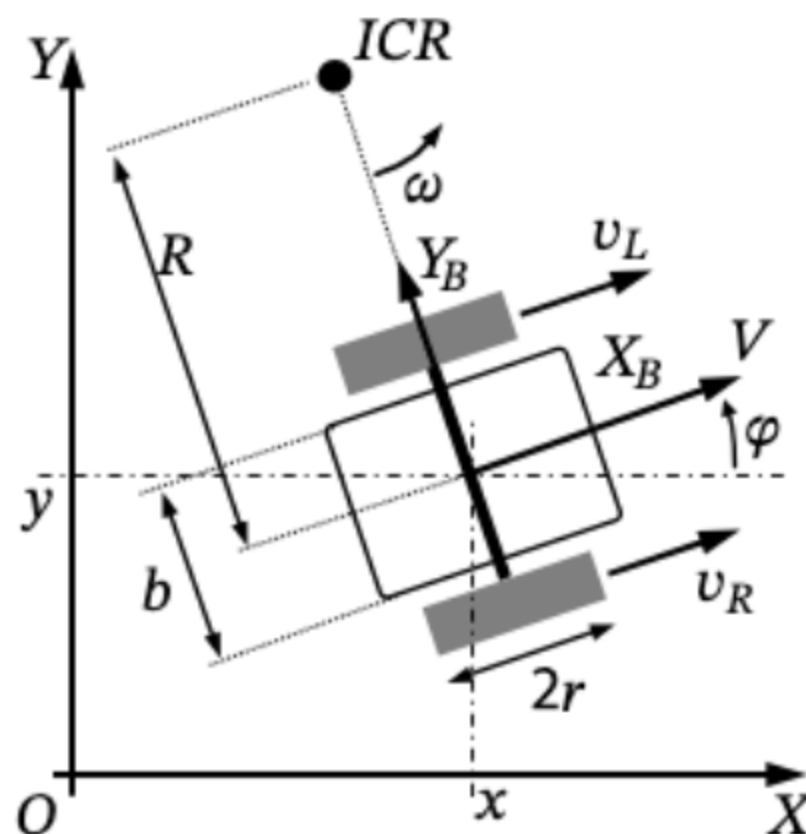
- 하나의 구성 요소인 링크와 이를 연결하는 조인트를 통해 로봇의 형태를 URDF로 표현

URDF를 통해 로봇을 시각화할 예정

2

diff_drive_controller

로봇을 ROS 명령어로 제어하기 위해 필요



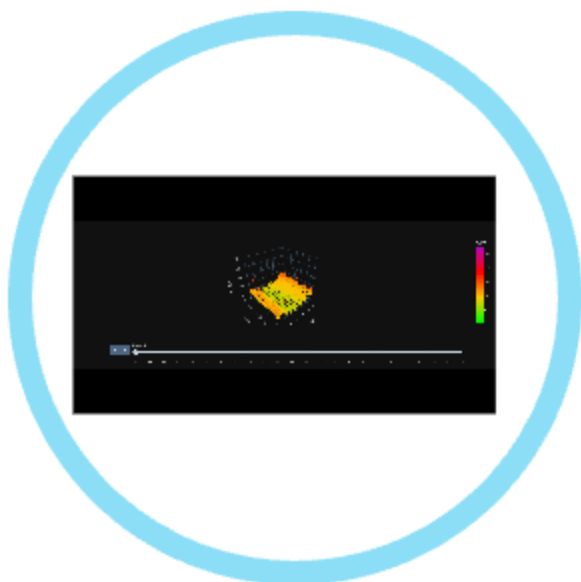
차량 구동 휠 시스템용 컨트롤러

- 제어: 속도 명령의 형태로 이루어짐
- 속도 명령 : 분할된 후 로봇의 구동 휠 베이스의 두 바퀴로 전송됨

현재 라즈베리파이에서 로봇을 컨트롤 하였지만,
이를 ROS에서 컨트롤 할 수 있도록 할 예정

공기 데이터

완료



- LSTM 이용한 예측 모델 1차 완성
- 깃허브에 푸시 완료

문제점



라즈베리파이와 우분투 통신의 어려움

진행 중




라즈베리의 센서 데이터를 받을 수 있는
ROS 명령어를 찾는 중

서버 통신 및 UI

☰

CareBuddy




Hello, mybuddy

공간별 케어도

현재 케어 공간

Map status



필터 교체 까지 : 2주

```
def stop_and_save(self, map_name):
    map_name = unquote(map_name)
    try:
        # 지도 저장 : subprocess.run 사용
        save_cmd = ['rosrun', 'map_server', 'map_saver', '-f', f'/home/haley/catkin_ws/saveMap/{map_name}']
        subprocess.run(save_cmd, check=True) # 명령어 완료까지 대기
        response = f"Map saved as {map_name}."

        # hector_slam 뷰트러얼 먼저 종료
        subprocess.call(['pkill', '-f', 'roslaunch hector_slam_launch tutorial.launch'])
        # 지도가 저장되었으므로 hector_slam 관련 모든 명령어 종료
        subprocess.call(['kill', '-f', 'hector_slam_launch'])

    except FileNotFoundError as e:
        response = f"Error saving map or stopping hector_slam: {e}"
    except subprocess.CalledProcessError as e:
        response = f"Error during command execution: {e}"

    self.send_response(200)
    self.send_header('Content-type', 'text/plain')
    self.end_headers()
    self.wfile.write(bytes(response, "utf8"))

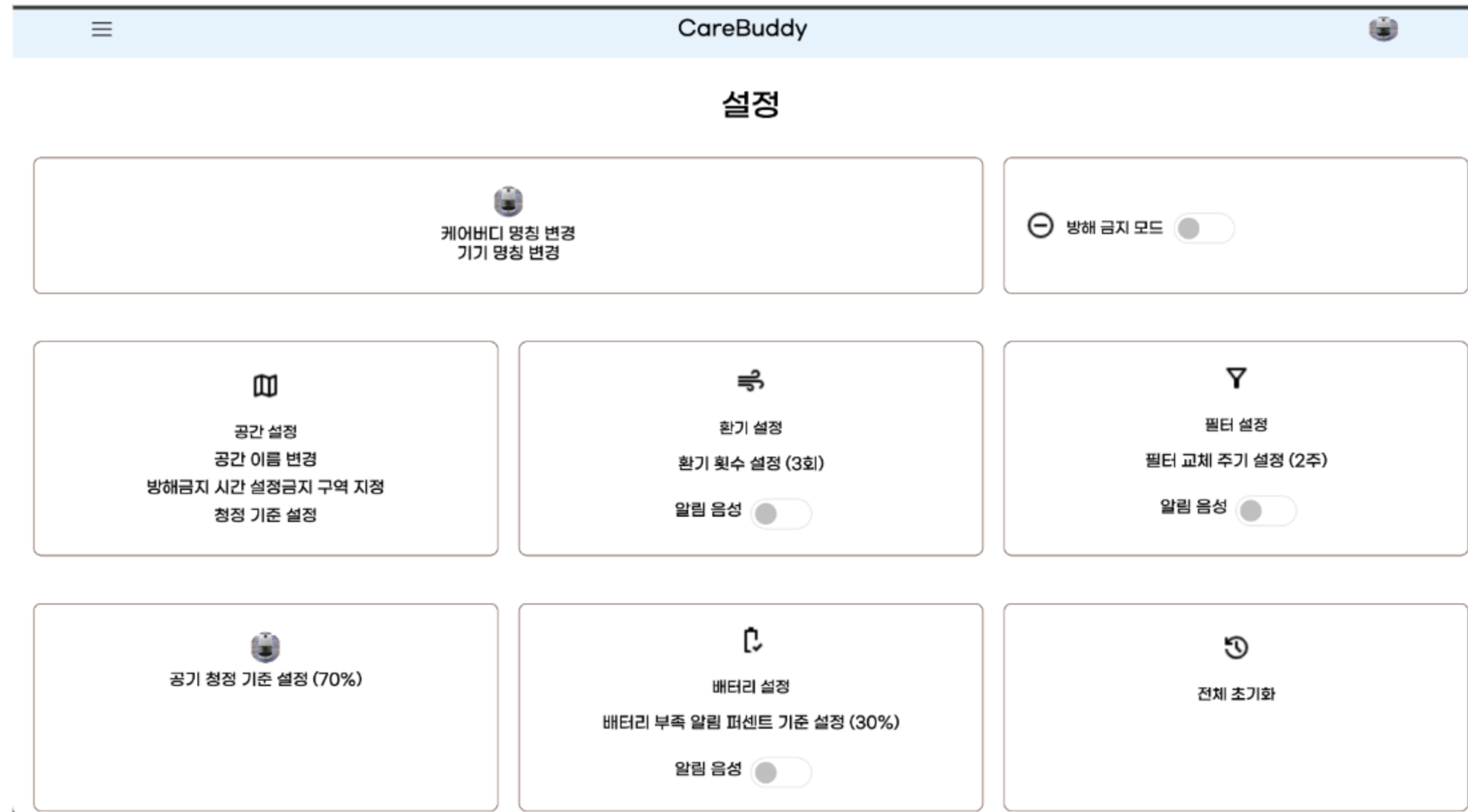
def load_map(self, map_name):
    map_path = f"/home/haley/catkin_ws/saveMap/{map_name}"
    try:
        print(f"Loading map: {map_path}")
        # 지도 불러오는 ros 명령어
        subprocess.Popen(['rosrun', 'map_server', 'map_server', f'{map_path}.yaml'])
        response = "Map loaded successfully."
    except Exception as e:
        response = f"Error loading map: {e}"
    self.send_response(200)
    self.send_header('Content-type', 'text/plain')
    self.end_headers()
    self.wfile.write(bytes(response, "utf8"))
```

매핑 후 기본 설정 Testcase

매핑 성공시 매핑 정보 출력할 수 있는가?

파이썬 코드와 ROS 명령어를 통하여, 우분투 터미널에서 직접적으로 명령어를 입력하지 않아도
"매핑 시작", "매핑 종료", "지도 저장", "지도 불러오기"를 동작하도록 하였다.

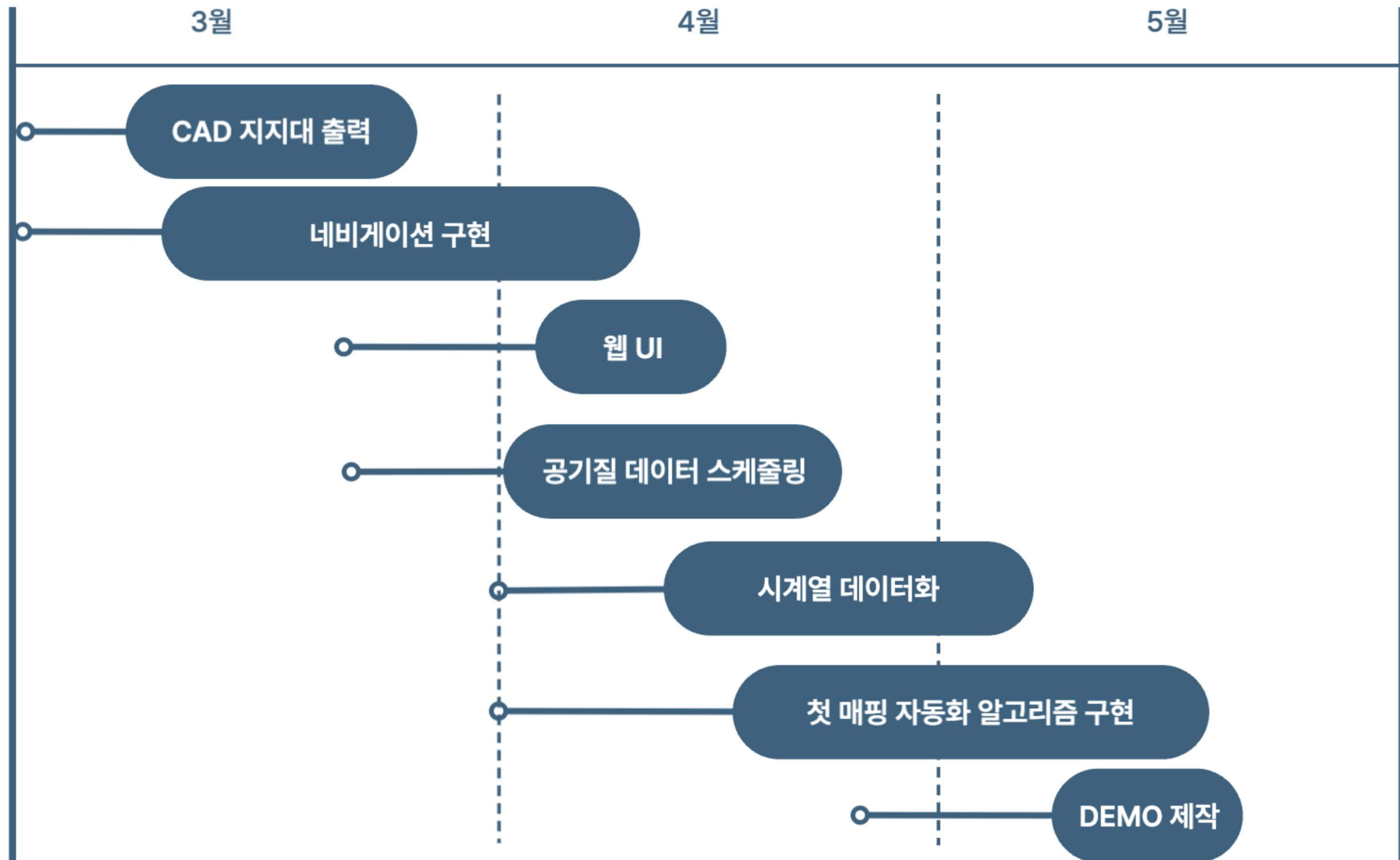
서버 통신 및 UI



설정 Testcase

- 사용자가 설정창에서 케어버디의 명칭을 변경 할 수 있는가?
- 사용자가 공간 설정창에서 공기 청정 기준을 설정할 수 있는가?
- 사용자가 환기 설정창에서 환기 알림 횟수를 설정할 수 있는가?
- 사용자가 필터 설정창에서 필터 교체 알림 주기를 설정할 수 있는가?
- 사용자가 배터리 설정창에서 배터리가 얼마 이하 일때 알림을 할지 설정 할 수 있는가?
- 사용자가 초기화창에서 케어버디의 설정 전체를 초기화 할 수 있는가?

계획



감사합니다.

Motus+er