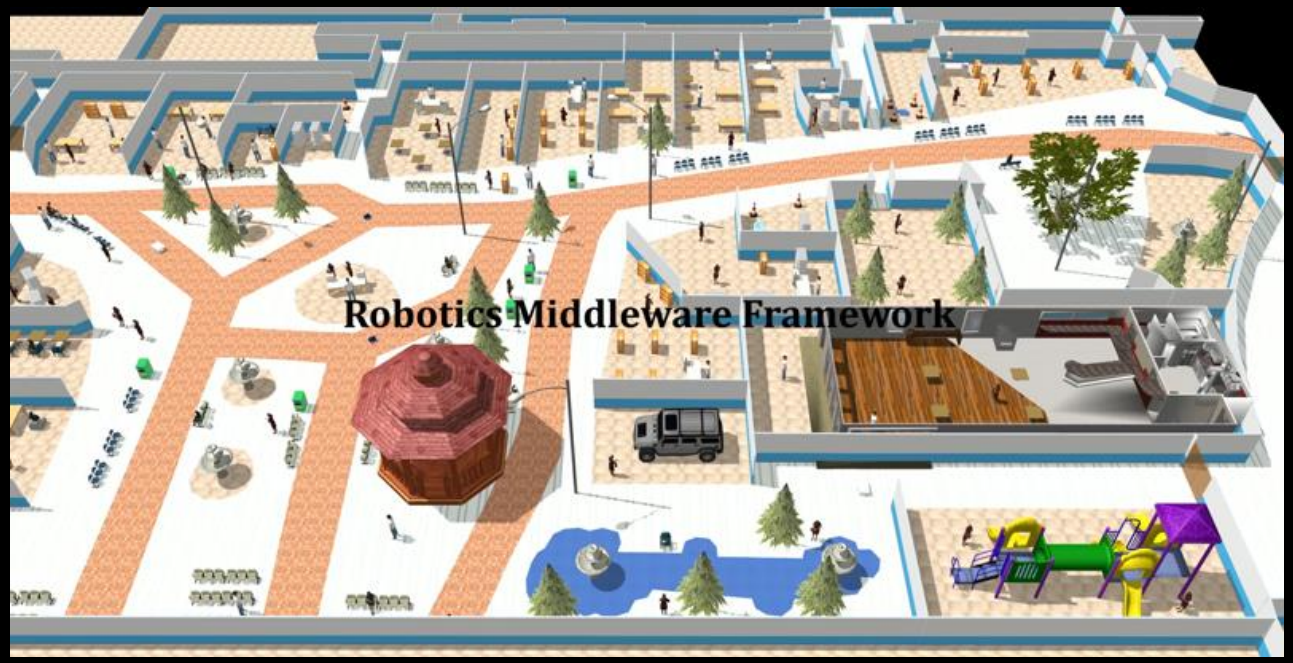
# Open-RMF

A Common Language for Robot Interoperability




# Lecture 6

정은빈

# Contents

# Custom world Task 구성

01 Clean Task

02 Patrol Task

03 Delivery Task

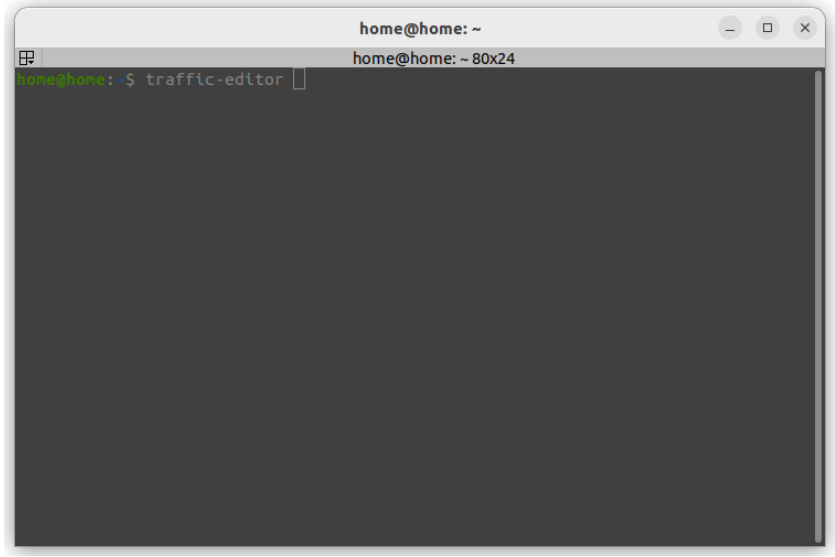
# Clean Task

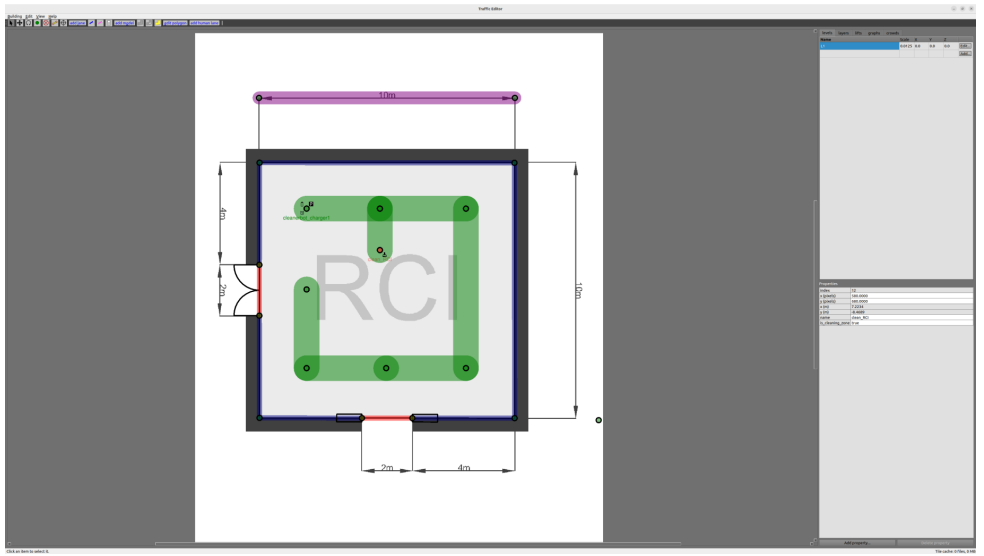
# Custom world Task 구성: Clean

## Clean Task 구성

### Traffic-Editor 실행 명령어 입력

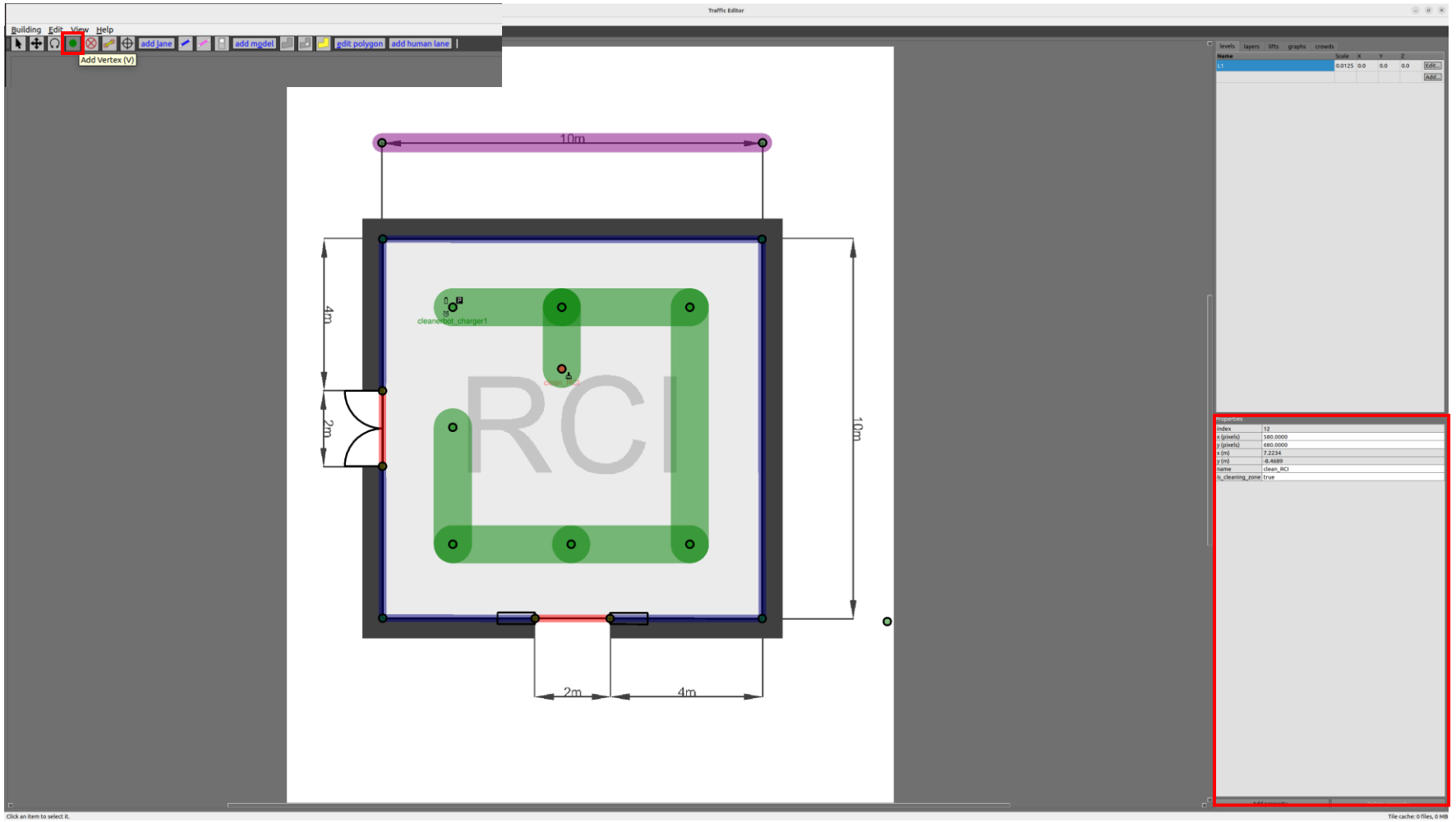
Traffic-editor



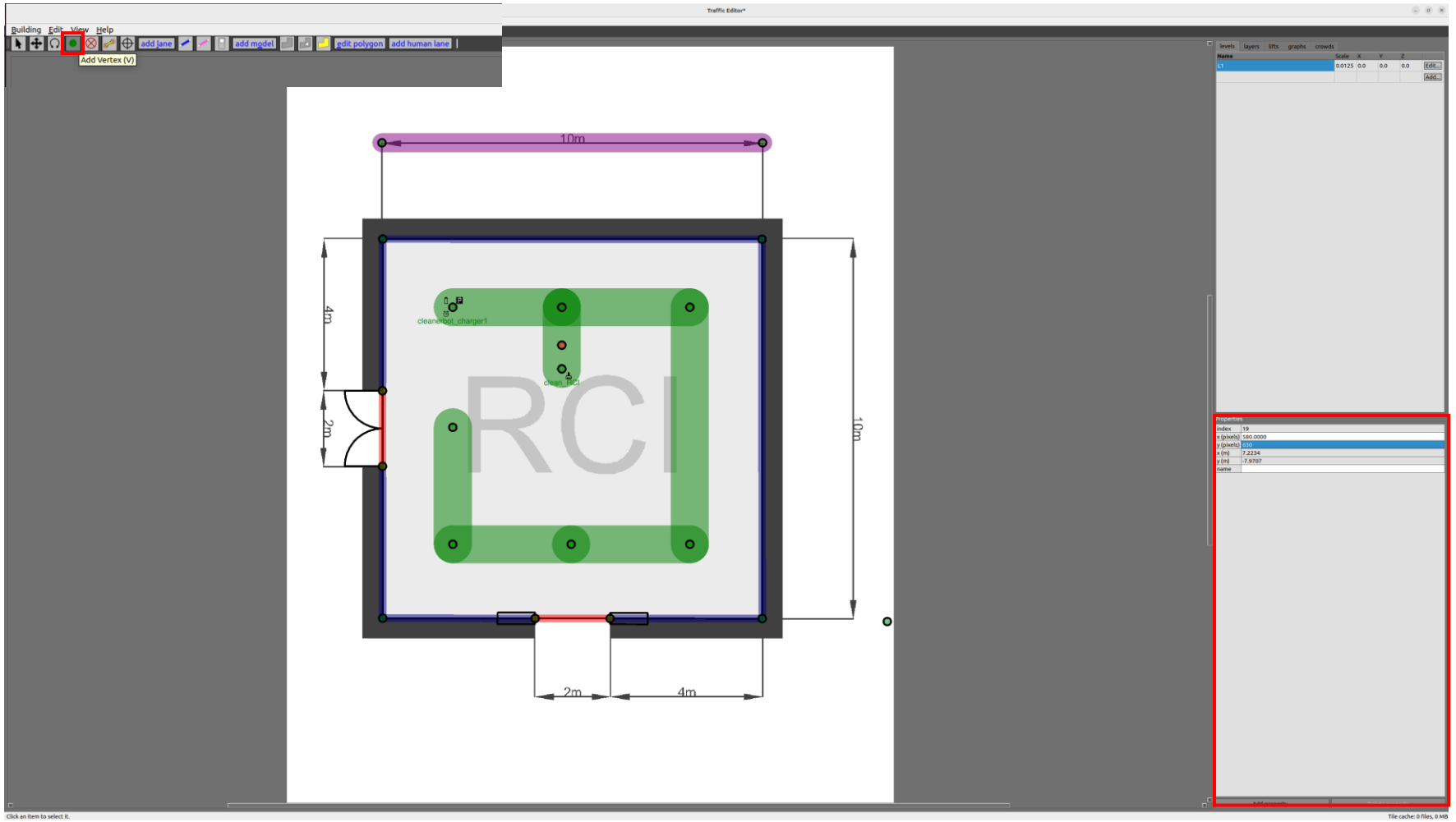
# Custom world Task 구성: Clean

- Clean Task 구성
  - Clean path 생성



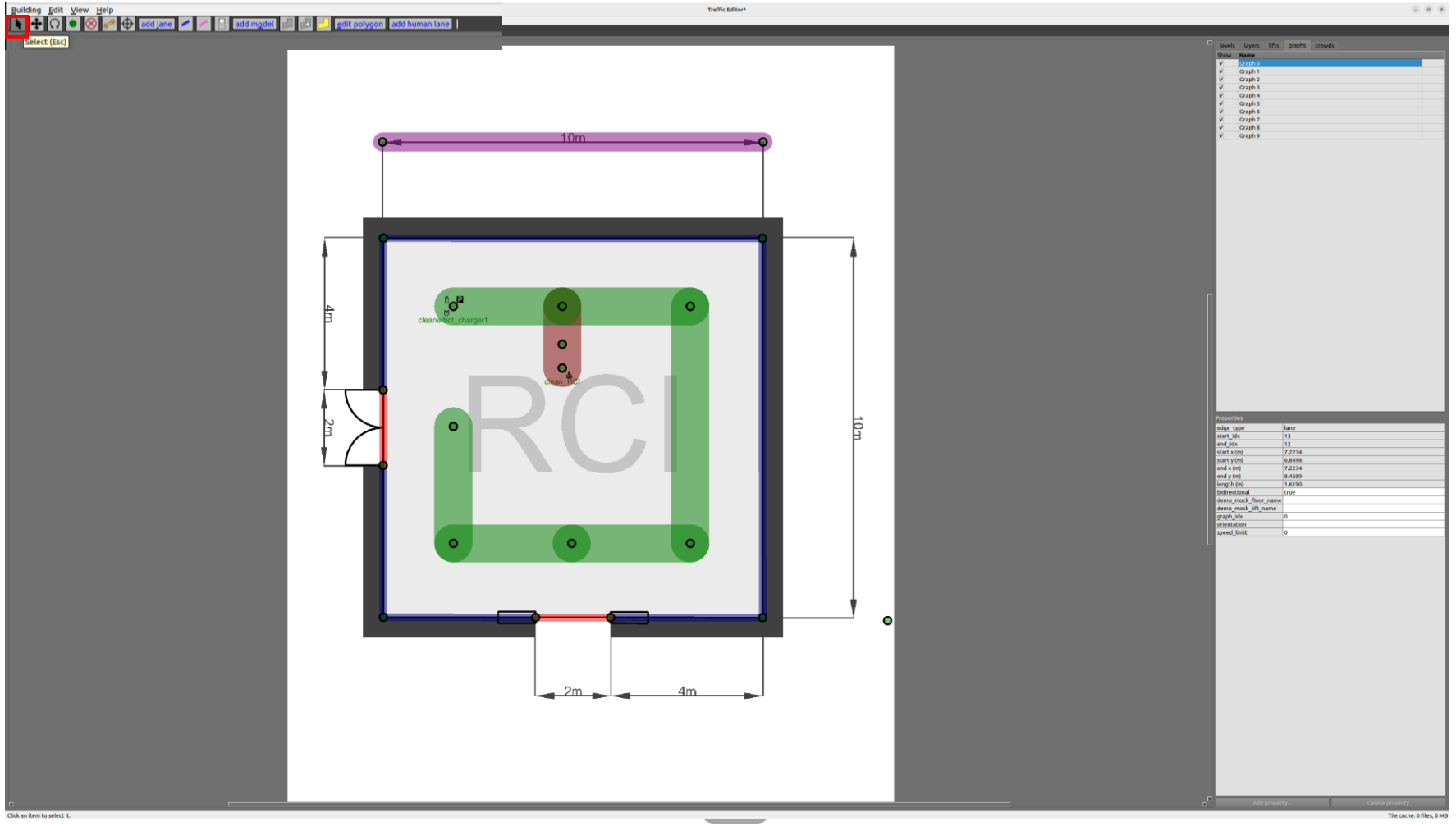
# Custom world Task 구성: Clean

- Clean Task 구성
  - Clean path 생성



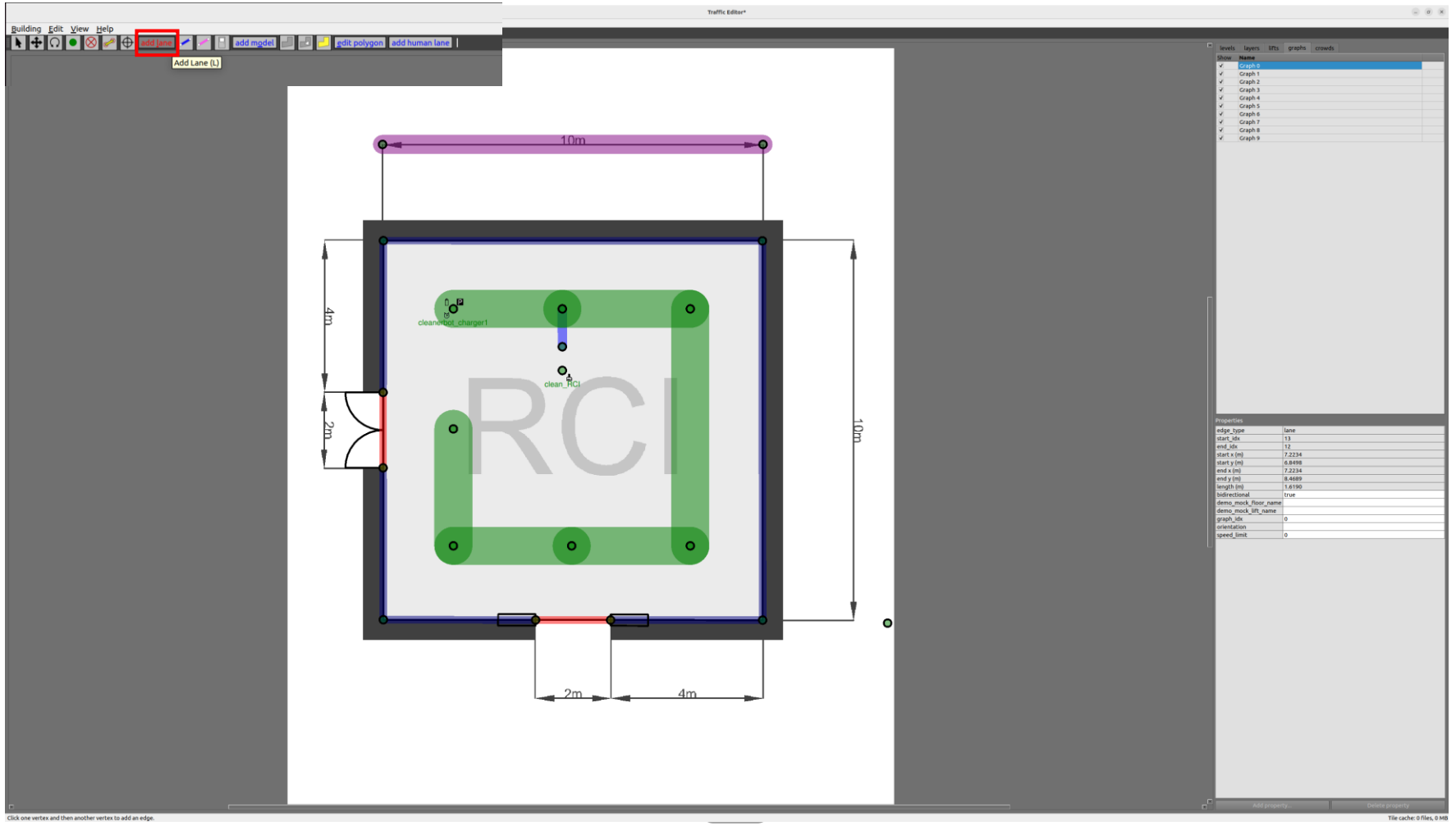
# Custom world Task 구성: Clean

- Clean Task 구성
  - Clean path 생성



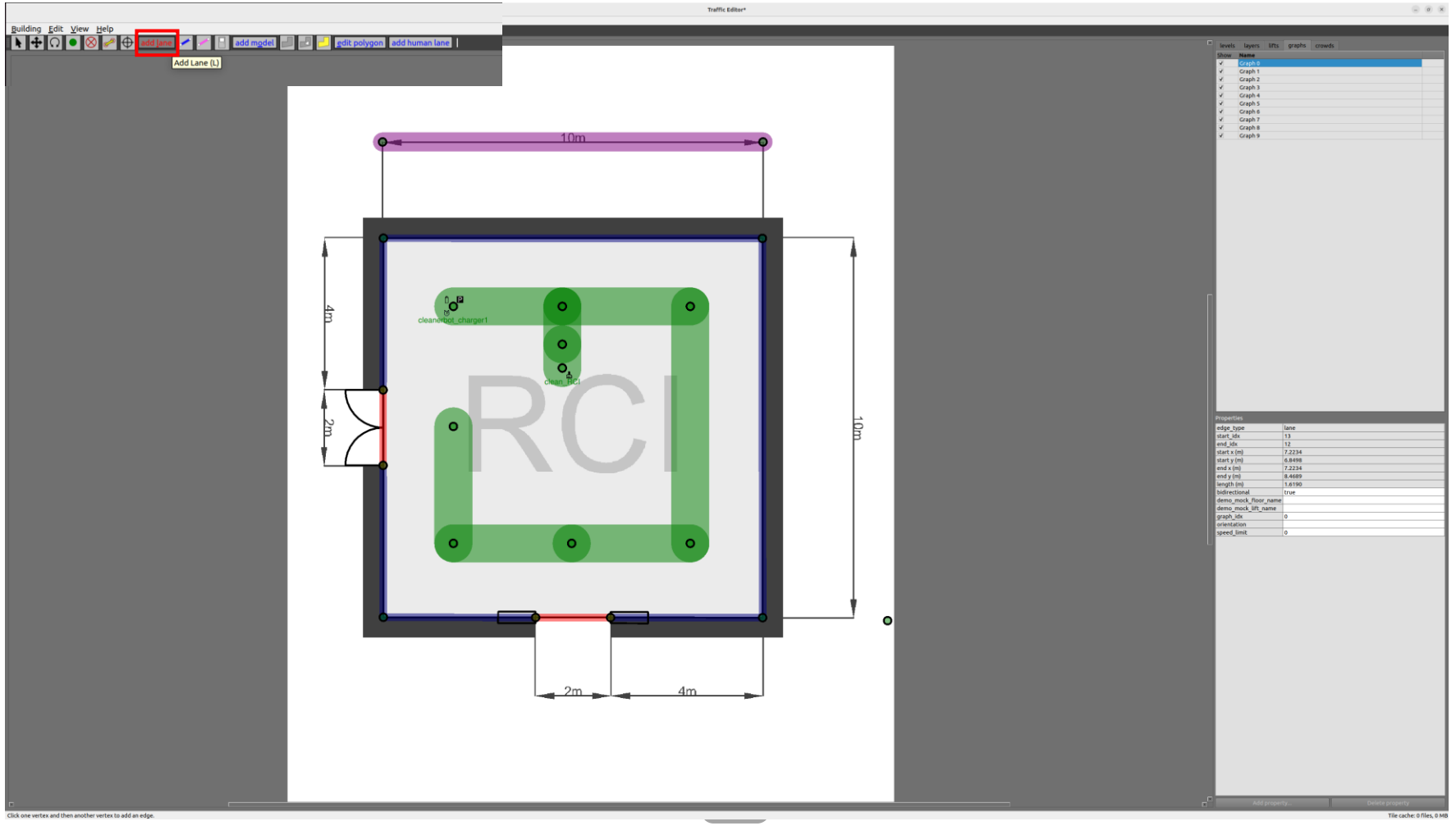
# Custom world Task 구성: Clean

- Clean Task 구성
  - Clean path 생성



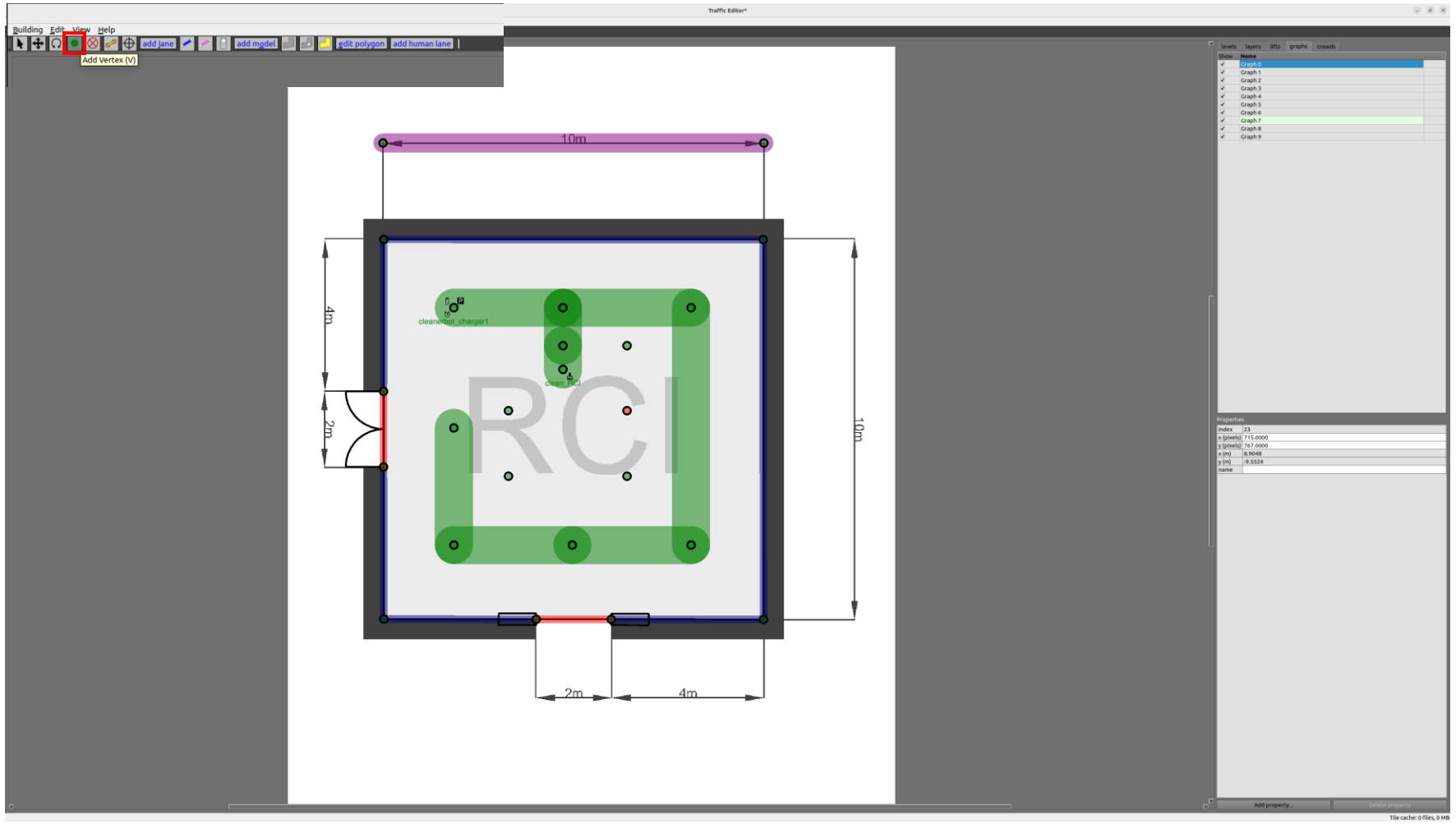
# Custom world Task 구성: Clean

- Clean Task 구성
  - Clean path 생성



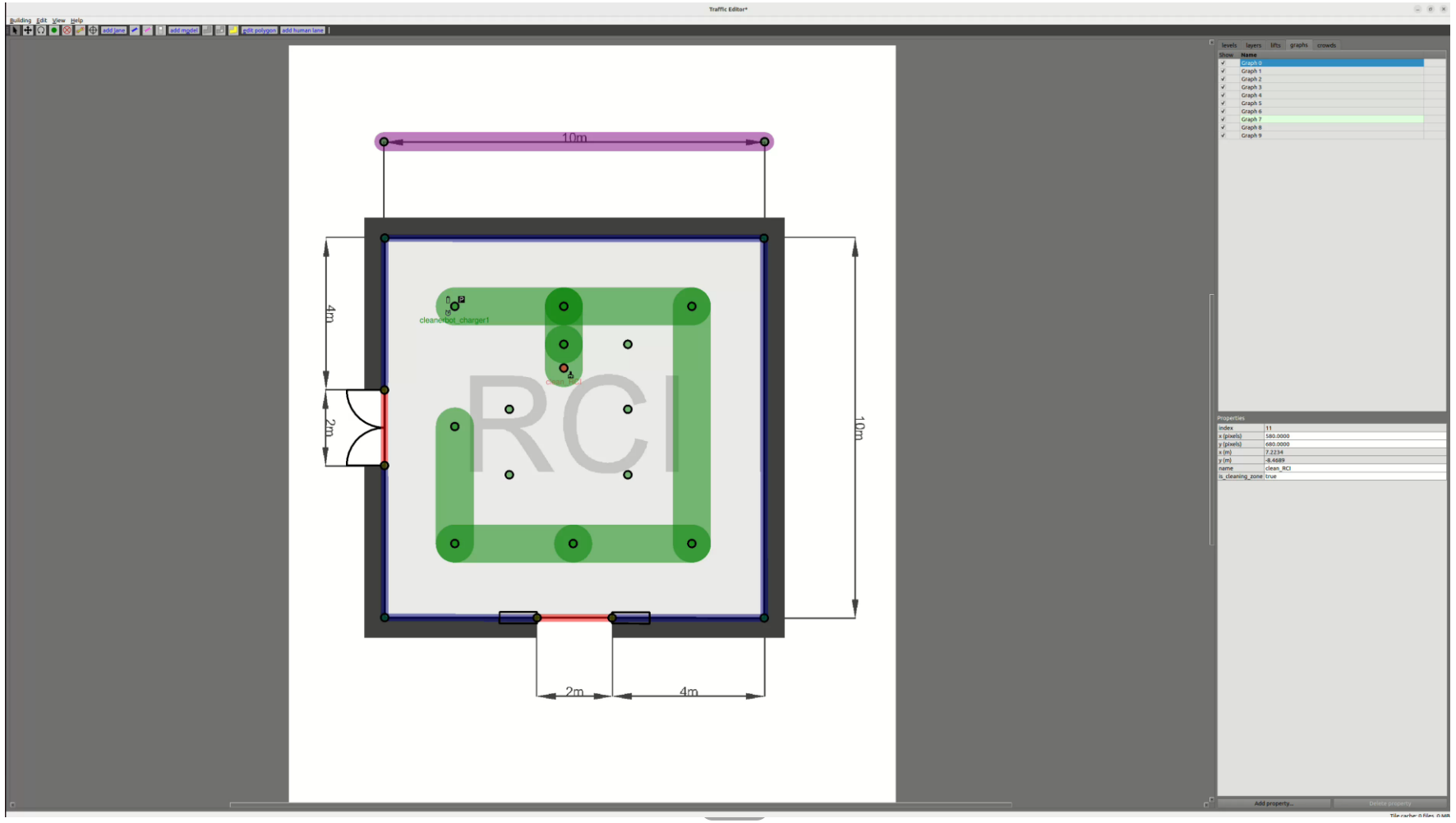
# Custom world Task 구성: Clean

- Clean Task 구성
  - Clean path 생성



# Custom world Task 구성: Clean

- Clean Task 구성
  - Clean path 생성



# Custom world Task 구성: Clean

## Clean Task 구성

### Docking 이란?

일반: 로봇청소기가 충전스테이션으로 복귀하는 것

RMF: 청소 루틴을 시작하고 완료하는 과정

→ 로봇이 청소 경로를 따라 이동하고, 지정된 영역에서 청소 작업을 수행한 뒤, 출발점으로 돌아와 다음 작업을 대기하거나 충전하는 방식

# Custom world Task 구성: Clean

## Clean Task 구성

### Docking 파일(rcilab_cleaner_config.yaml ) 구성 및 적용

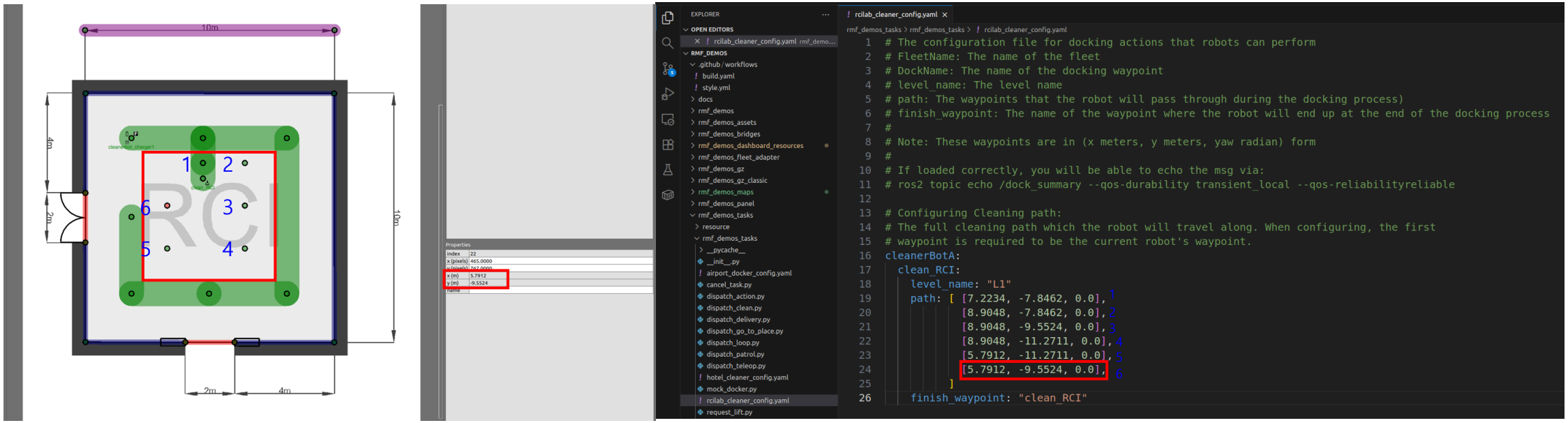
~/rmf_ws/src/rmf_demos/rmf_demos_tasks/rmf_demos_tasks/rci_cleaner_config.yaml

### rcilab_cleaner_config.yaml 내용 작성

```
# The configuration file for docking actions that robots can perform
# FleetName: The name of the fleet
# DockName: The name of the docking waypoint# level_name: The level name
# path: The waypoints that the robot will pass through during the docking process)
# finish_waypoint: The name of the waypoint where the robot will end up at the end of the docking process
# Note: These waypoints are in (x meters, y meters, yaw radian) form
# If loaded correctly, you will be able to echo the msg via:
# ros2 topic echo /dock_summary --qos-durability transient_local --qos-reliabilityreliable
# Configuring Cleaning path:
# The full cleaning path which the robot will travel along. When configuring, the first
# waypoint is required to be the current robot's waypoint.
cleanerBotA:
  clean_RCI:
    level_name: "L1"
    path: [ ]
    finish_waypoint: "clean_RCI"
```

# Custom world Task 구성: Clean

- Clean Task 구성
  - rcilab_cleaner_config.yaml 내용 작성

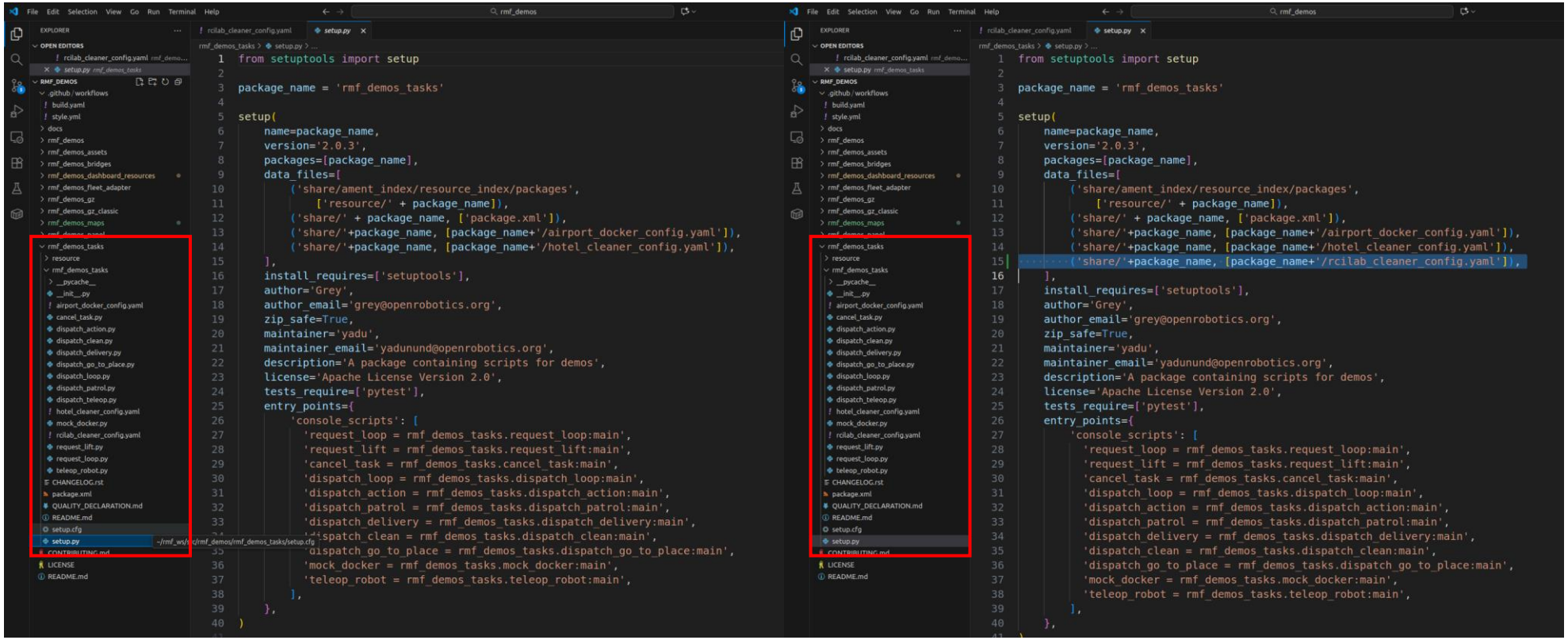


# Custom world Task 구성: Clean

## Clean Task 구성

### Setup.py 내용 수정

('share/'+package_name, [package_name]+'/rcilab_cleaner_config.yaml')



```
from setuptools import setup

package_name = 'rmf_demos_tasks'

setup(
    name=package_name,
    version='2.0.3',
    packages=[package_name],
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
        ('share/'+package_name, [package_name+'/airport_docker_config.yaml']),
        ('share/'+package_name, [package_name+'/hotel_cleaner_config.yaml']),
    ],
    install_requires=['setuptools'],
    author='Grey',
    author_email='grey@openrobotics.org',
    zip_safe=True,
    maintainer='yadu',
    maintainer_email='yadunund@openrobotics.org',
    description='A package containing scripts for demos',
    license='Apache License Version 2.0',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
          'request_loop = rmf_demos_tasks.request_loop:main',
          'request_lift = rmf_demos_tasks.request_lift:main',
          'cancel_task = rmf_demos_tasks.cancel_task:main',
          'dispatch_loop = rmf_demos_tasks.dispatch_loop:main',
          'dispatch_action = rmf_demos_tasks.dispatch_action:main',
          'dispatch_patrol = rmf_demos_tasks.dispatch_patrol:main',
          'dispatch_delivery = rmf_demos_tasks.dispatch_delivery:main',
          'dispatch_clean = rmf_demos_tasks.dispatch_clean:main',
          'dispatch_go_to_place = rmf_demos_tasks.dispatch_go_to_place:main',
          'mock_docker = rmf_demos_tasks.mock_docker:main',
          'teleop_robot = rmf_demos_tasks.teleop_robot:main',
        ],
    },
)
```

# Custom world Task 구성: Clean

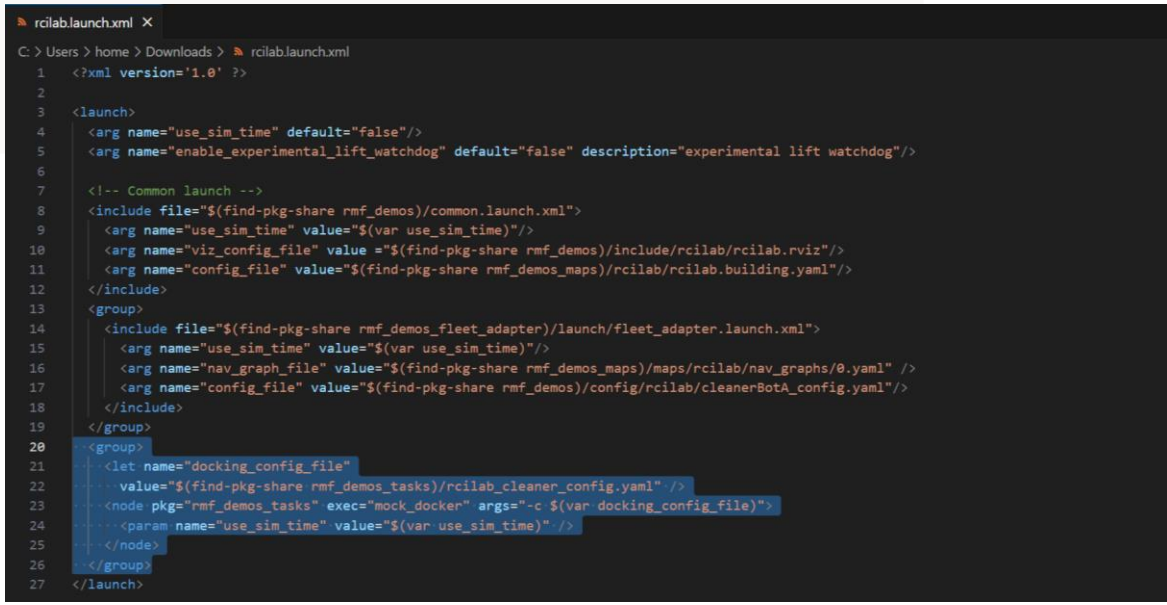
## Clean Task 구성

### rcilab.launch.xml 파일 수정

~/rmf_ws/src/rmf_demos/rmf_demos/launch/rcilab.launch.xml

### rcilab.launch.xml 추가 내용

```
<group>
  <let name="docking_config_file" value="$(find-pkg-share rmf_demos_tasks)/rcilab_cleaner_config.yaml" />
  <node pkg="rmf_demos_tasks" exec="mock_docker" args="-c $(var docking_config_file)">
    <param name="use_sim_time" value="$(var use_sim_time)" />
  </node>
</group>
```



# Custom world Task 구성: Clean

- rcilab world Clean Task 확인
  - Build

```
cd ~/rmf_ws
colcon build
```

  - rcilab.launch.xml 실행

```
cd ~/rmf_ws && source install/setup.bash
```

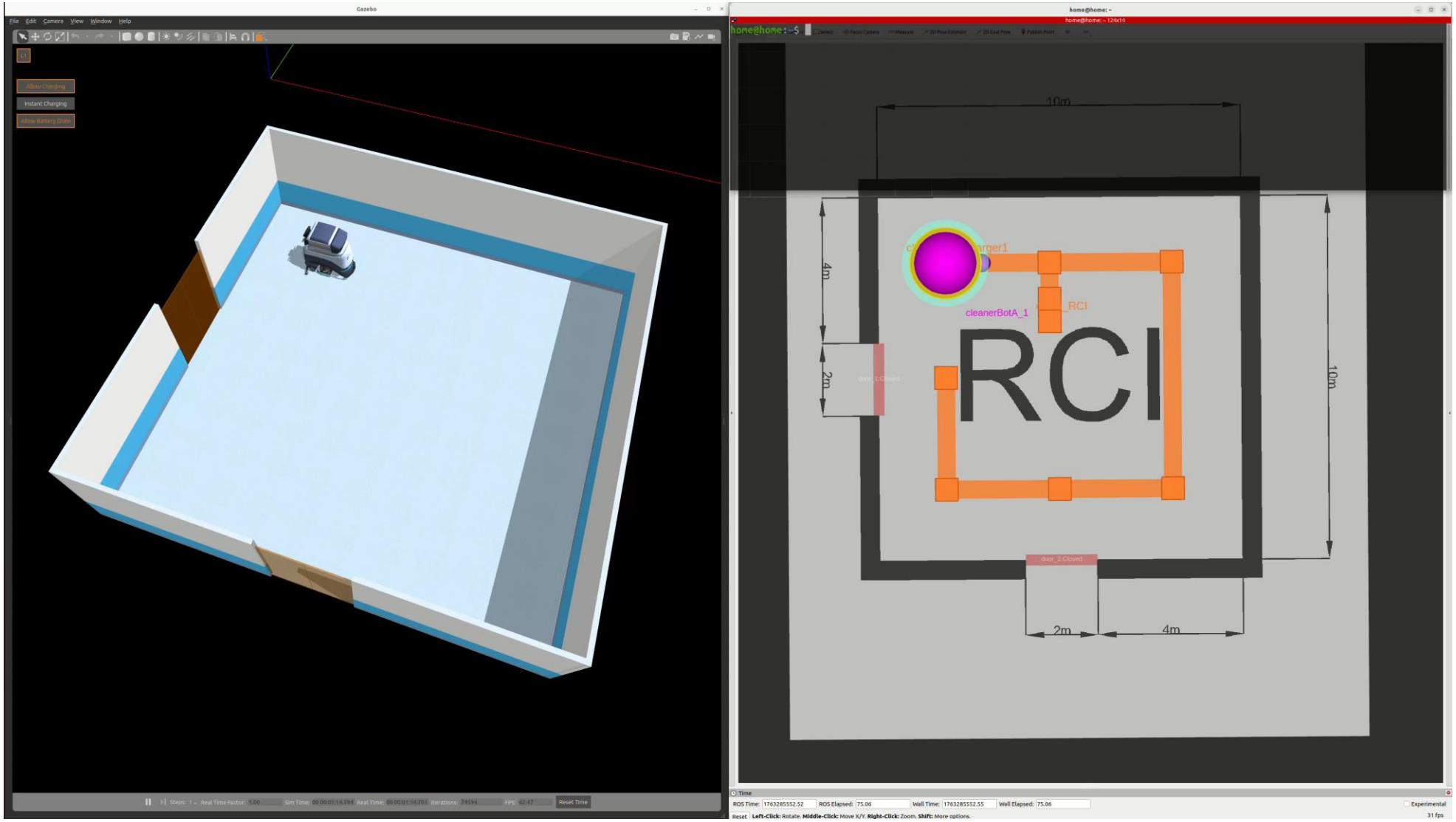
```
ros2 launch rmf_demos_gz_classic rcilab.launch.xml
```

  - Clean Task 명령

```
ros2 run rmf_demos_tasks dispatch_clean –cs clean_RCI --use_sim_time
```

# Custom world Task 구성: Clean

- rcilab world Clean Task 확인



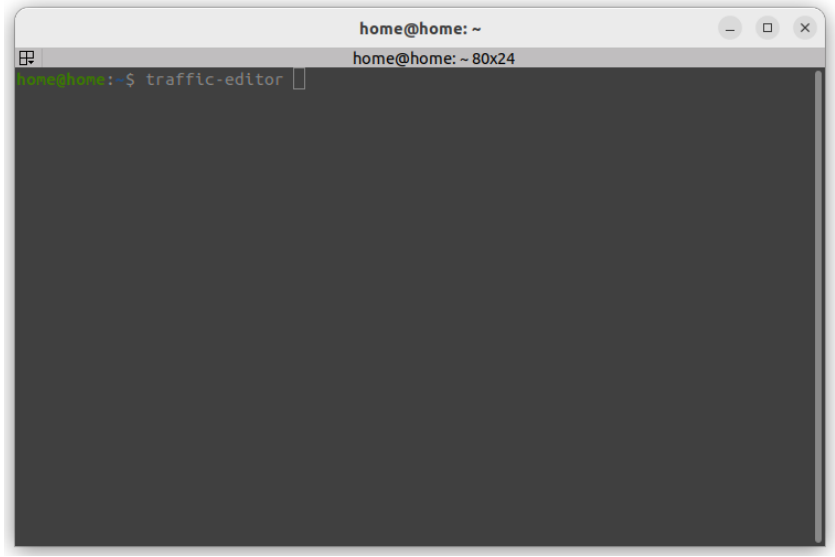
# Patrol Task

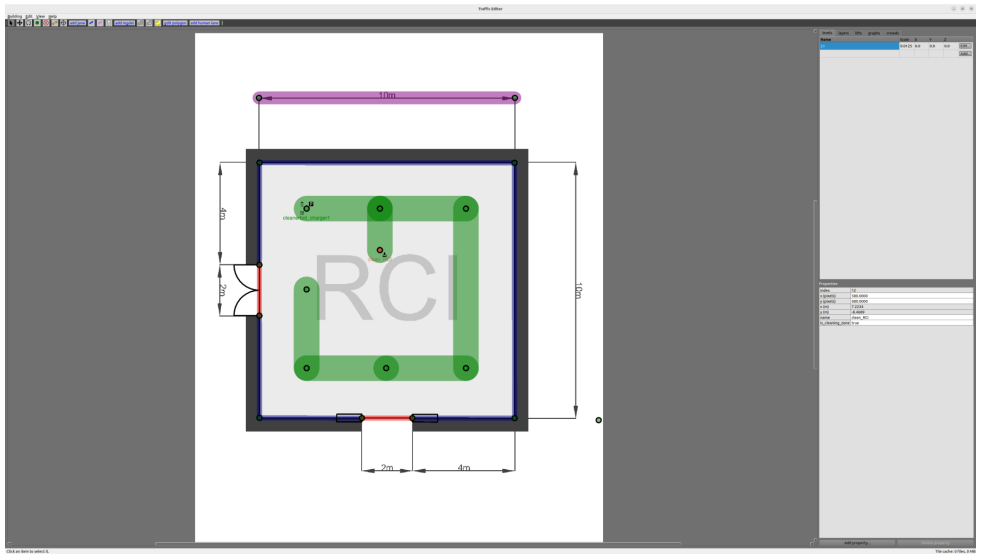
# Custom world Task 구성: Patrol

## Patrol Task 구성

### Traffic-Editor 실행 명령어 입력

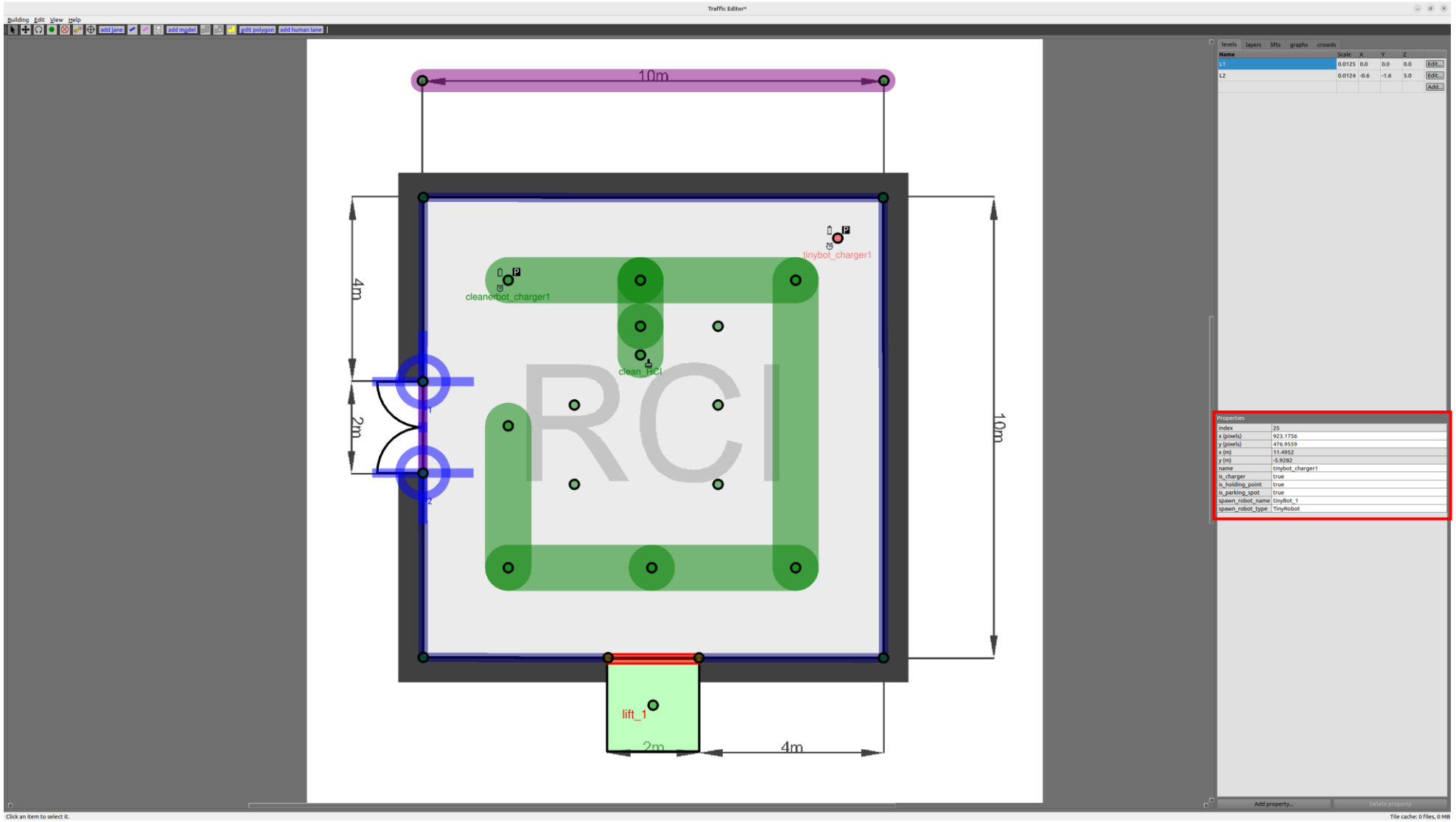
Traffic-editor



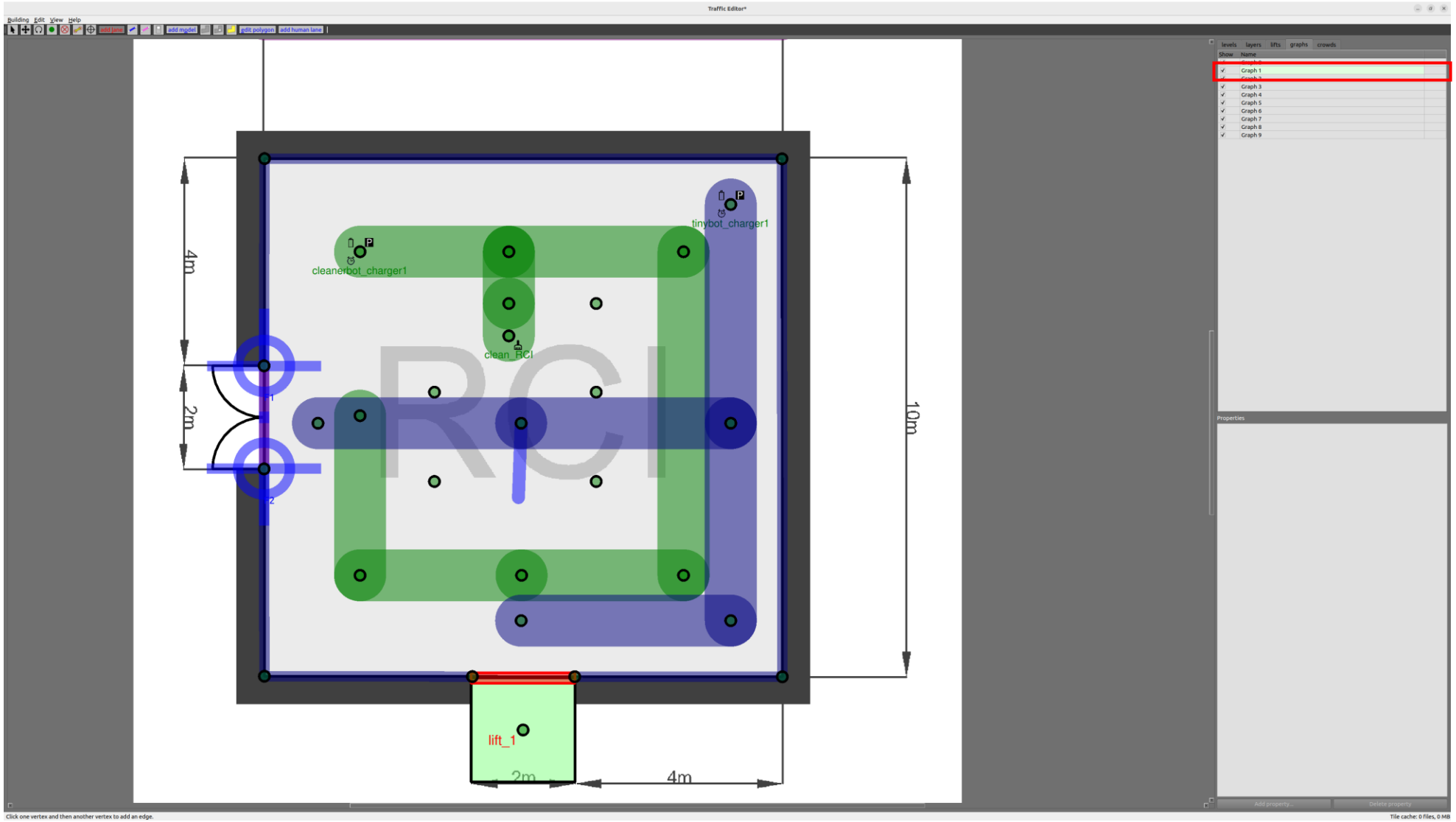
# Custom world Task 구성: Patrol

- rcilab world TinyRobot 추가



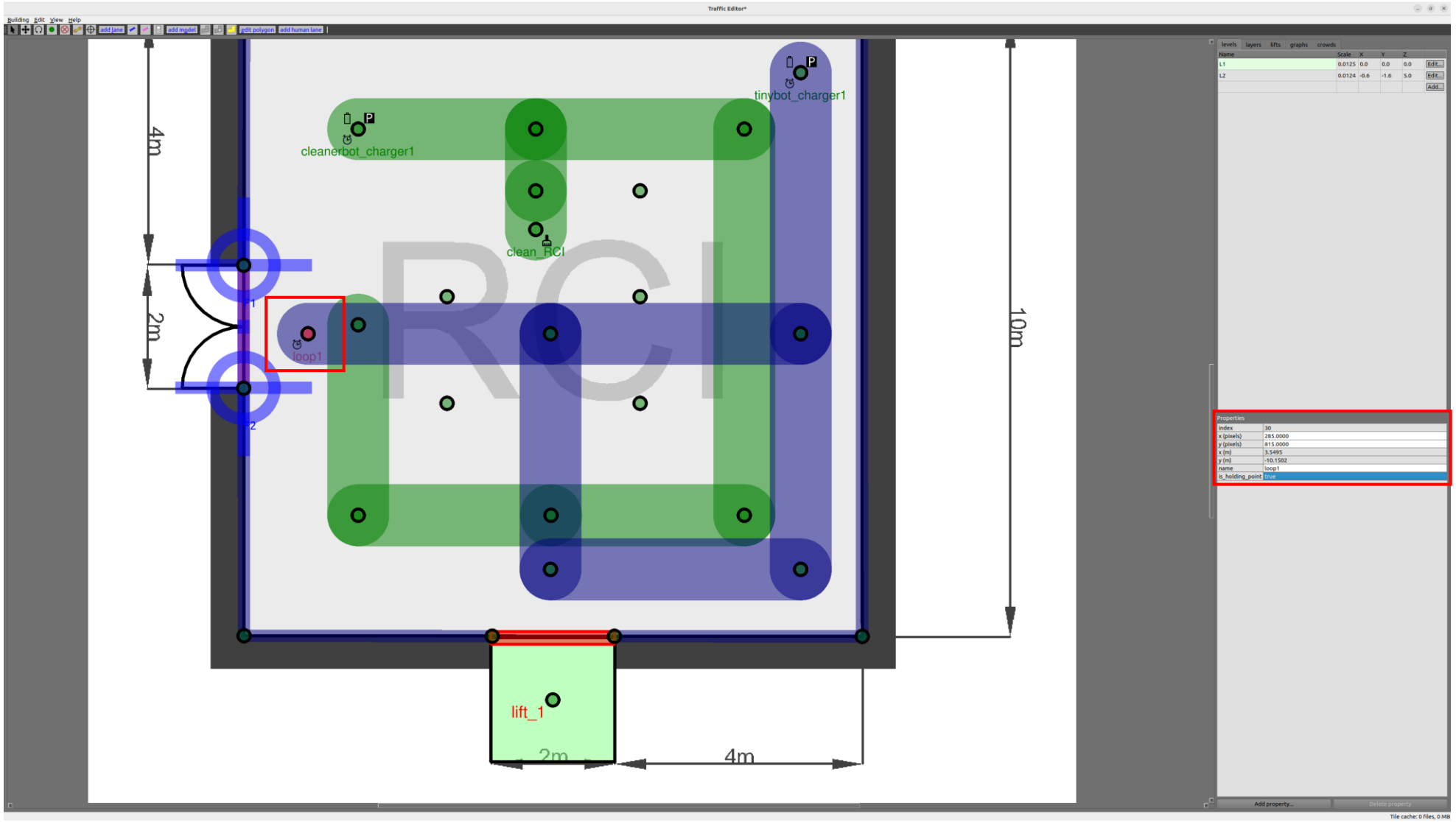
# Custom world Task 구성: Patrol

- rcilab world Patrol 경로 추가



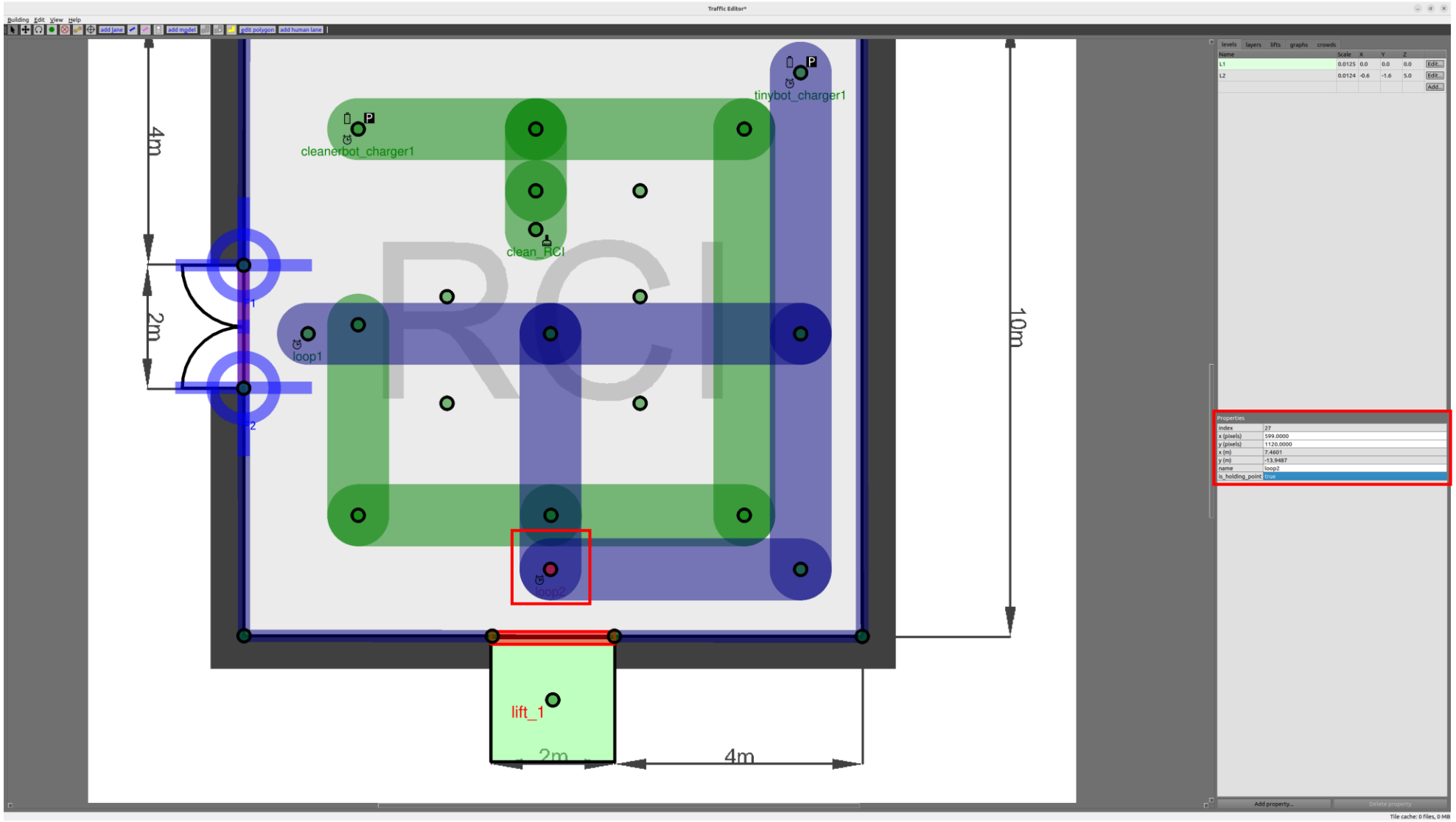
# Custom world Task 구성: Patrol

- rcilab world Patrol 경유지 추가



# Custom world Task 구성: Patrol

- rcilab world Patrol 경유 점 추가



# Custom world Task 구성: Patrol

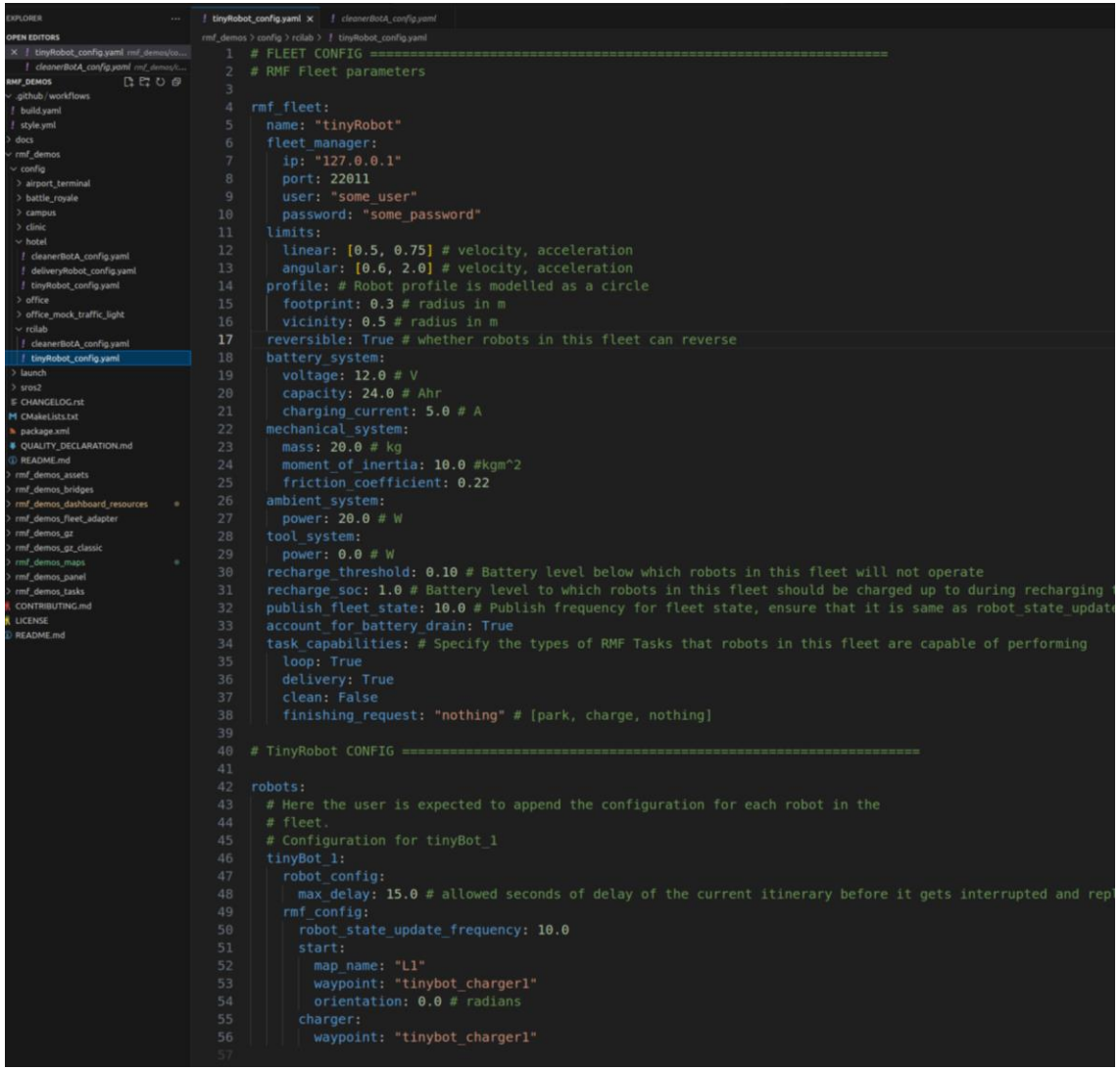
- rcilab tinyRobot 경로 추가 (config 파일 생성)
  - tinyRobot_config.yaml 파일 생성

```
~/rmf_ws/src/rmf_demos/rmf_demos/rcilab/config/tinyRobot_config.yaml
```

# tinyRobot_config.yaml 내용

```yaml
# FLEET CONFIG ================================================================
# RMF Fleet parameters
rmf_fleet:
  name: "tinyRobot"
  fleet_manager:
    ip: "127.0.0.1"
    port: 22011
    user: "some_user"
    password: "some_password"
  limits:
    linear: [0.5, 0.75] # velocity, acceleration
    angular: [0.6, 2.0] # velocity, acceleration
  profile: # Robot profile is modelled as a circle
    footprint: 0.3 # radius in m
    vicinity: 0.5 # radius in m
  reversible: True # whether robots in this fleet can reverse
  battery_system:
    voltage: 12.0 # V
    capacity: 24.0 # Ahr
    charging_current: 5.0 # A
  mechanical_system:
    mass: 20.0 # kg
    moment_of_inertia: 10.0 #kgm^2
    friction_coefficient: 0.22
  ambient_system:
    power: 20.0 # W
  tool_system:
    power: 0.0 # W
  recharge_threshold: 0.10 # Battery level below which robots in this fleet will not operate
  recharge_soc: 1.0 # Battery level to which robots in this fleet should be charged up to during recharging tasks
  publish_fleet_state: 10.0 # Publish frequency for fleet state, ensure that it is same as robot_state_update_frequency
  account_for_battery_drain: True
  task_capabilities: # Specify the types of RMF Tasks that robots in this fleet are capable of performing
    loop: True
    delivery: True
    clean: False
    finishing_request: "park" # [park, charge, nothing]

# TinyRobot CONFIG ============================================================
robots:
  # Here the user is expected to append the configuration for each robot in the
  # fleet.
  # Configuration for tinyBot_1
  tinyBot_1:
    robot_config:
      max_delay: 15.0 # allowed seconds of delay of the current itinerary before it gets interrupted and replanned
    rmf_config:
      robot_state_update_frequency: 10.0
      start:
        map_name: "L1"
        waypoint: "tinybot_charger1"
        orientation: 0.0 # radians
      charger:
        waypoint: "tinybot_charger1"
```



# Custom world Task 구성: Patrol

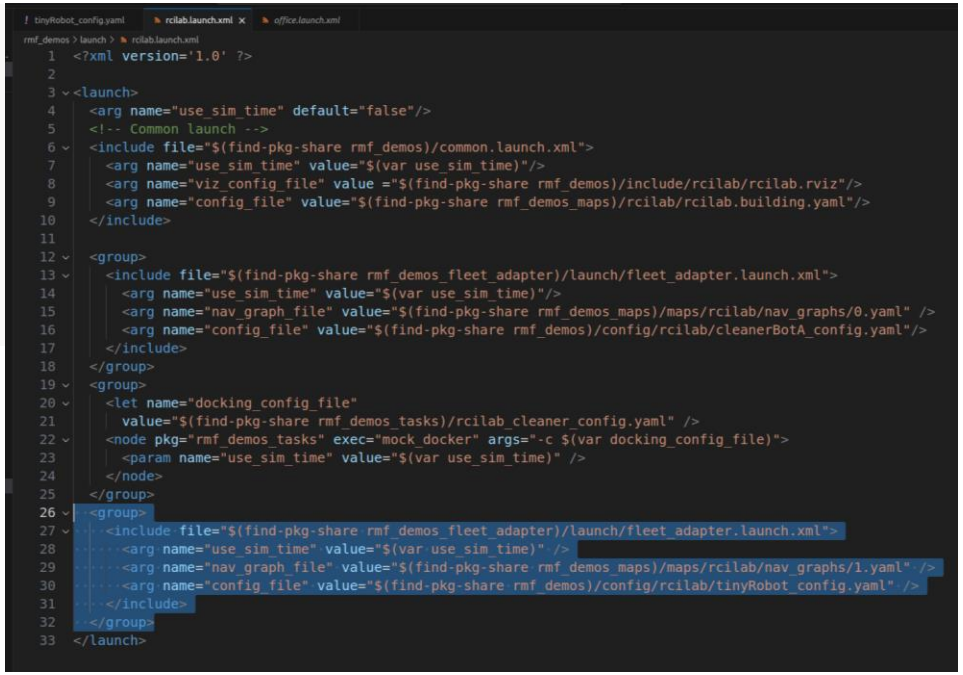
## launch 파일 수정

### rcilab.launch.xml 파일 수정

~/rmf_ws/src/rmf_demos/rmf_demos/launch/rcilab.launch.xml

### rcilab.launch.xml 추가 내용

```
<group>
  <include file="$(find-pkg-share rmf_demos_fleet_adapter)/launch/fleet_adapter.launch.xml">
    <arg name="use_sim_time" value="$(var use_sim_time)"/>
    <arg name="nav_graph_file" value="$(find-pkg-share rmf_demos_maps)/maps/rcilab/nav_graphs/1.yaml" />
    <arg name="config_file" value="$(find-pkg-share rmf_demos)/config/rcilab/tinyRobot_config.yaml"/>
  </include>
</group>
```



# Custom world Task 구성: Patrol

- rcilab world Patrol Task 확인
  - Build

```
cd ~/rmf_ws
colcon build
```

  - rcilab.launch.xml 실행

```
cd ~/rmf_ws && source install/setup.bash
```

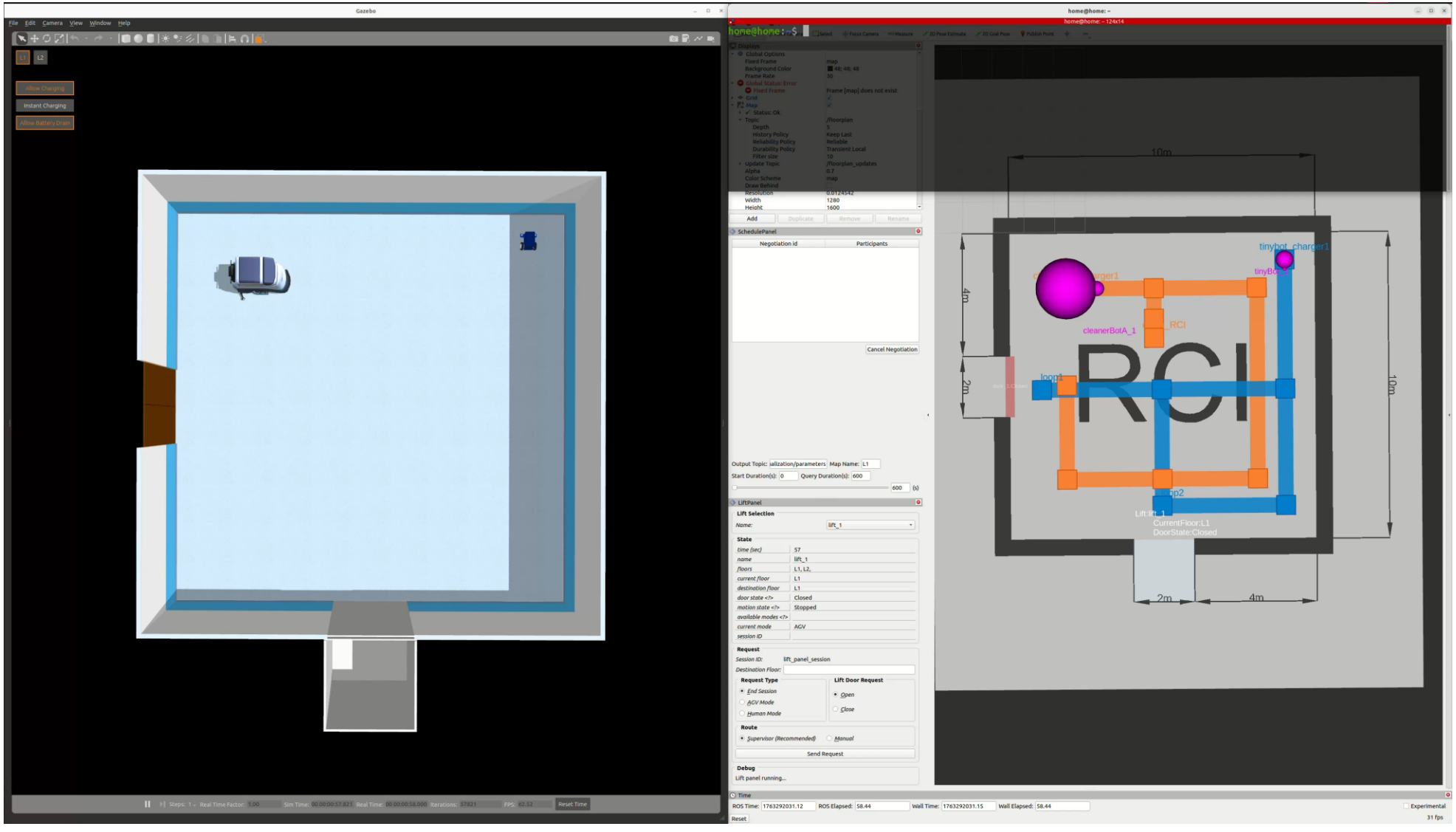
```
ros2 launch rmf_demos_gz_classic rcilab.launch.xml
```

  - Patrol Task 명령

```
ros2 run rmf_demos_tasks dispatch_patrol –p loop1 loop2 -n 1 --use_sim_time
```

# Custom world Task 구성: Patrol

- rcilab world Patrol Task 확인



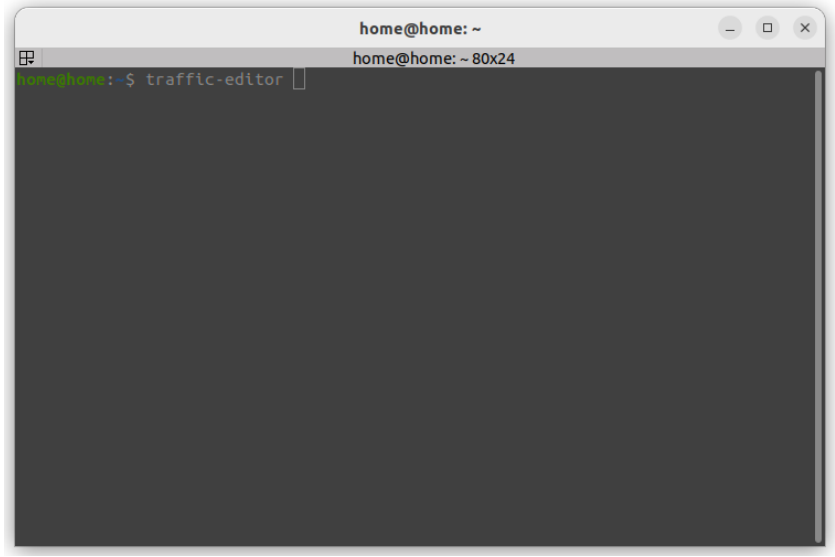
# Delivery **Task**

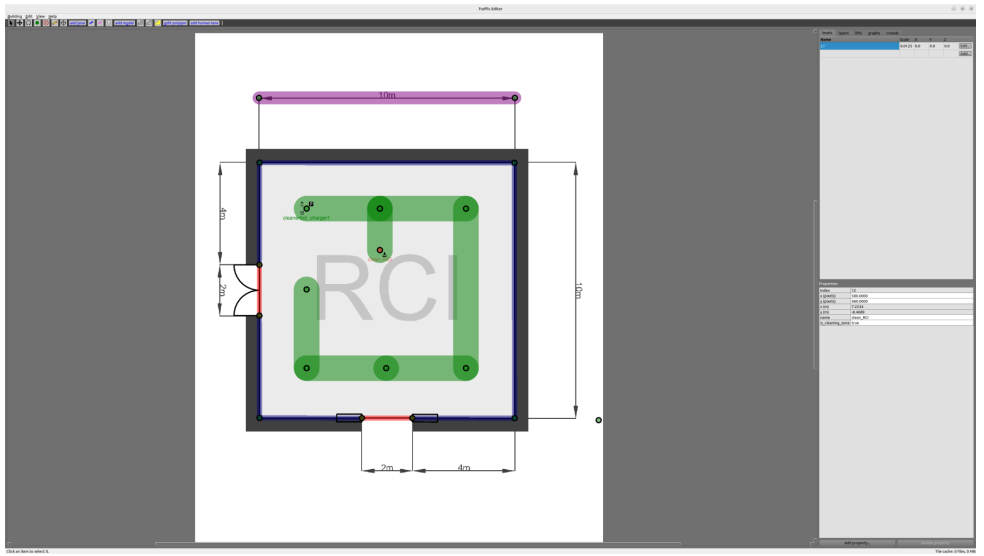
# Custom world Task 구성: Delivery

## Delivery Task 구성

### Traffic-Editor 실행 명령어 입력

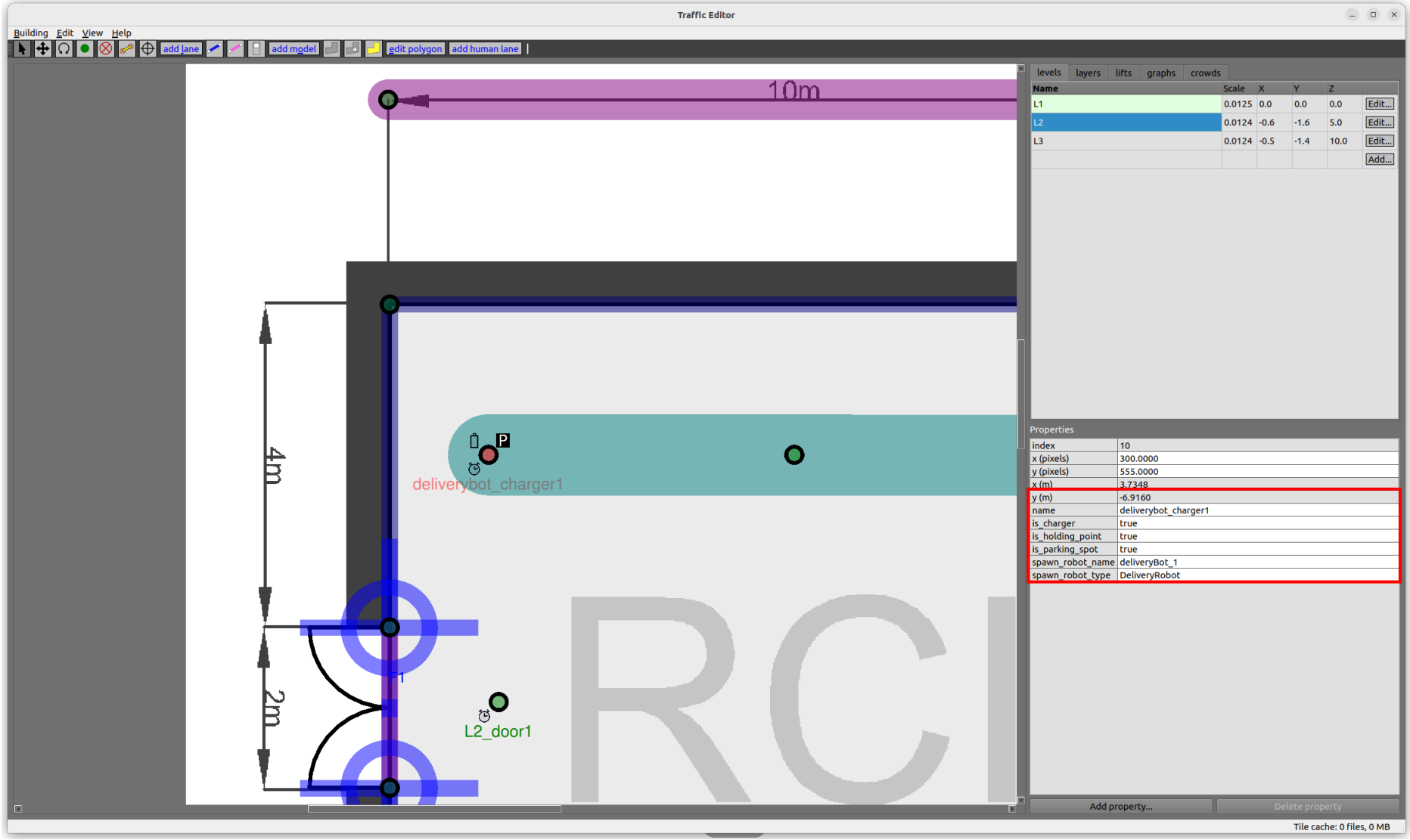
Traffic-editor



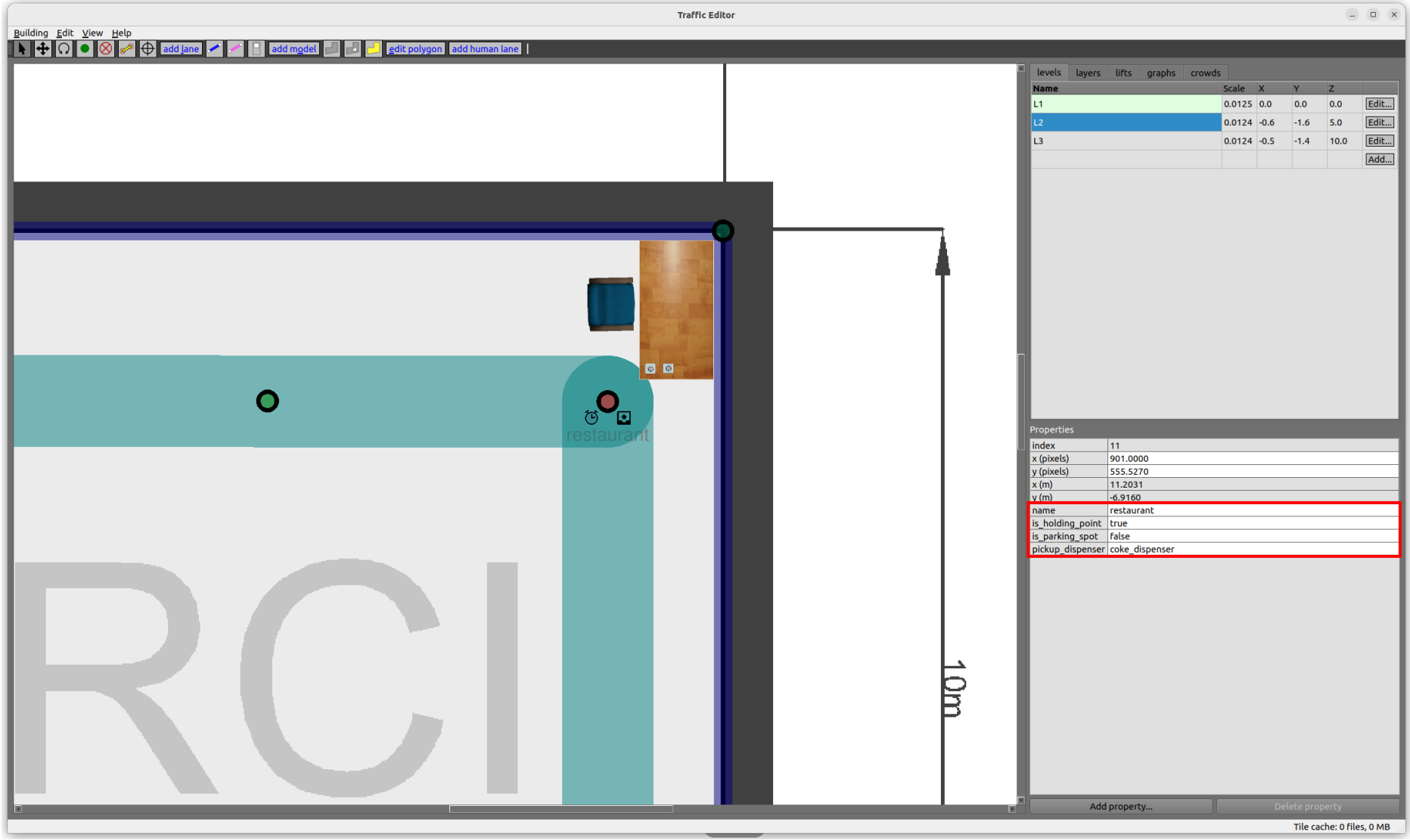
# Custom world Task 구성: Delivery

- rcilab world DeliveryRobot 추가



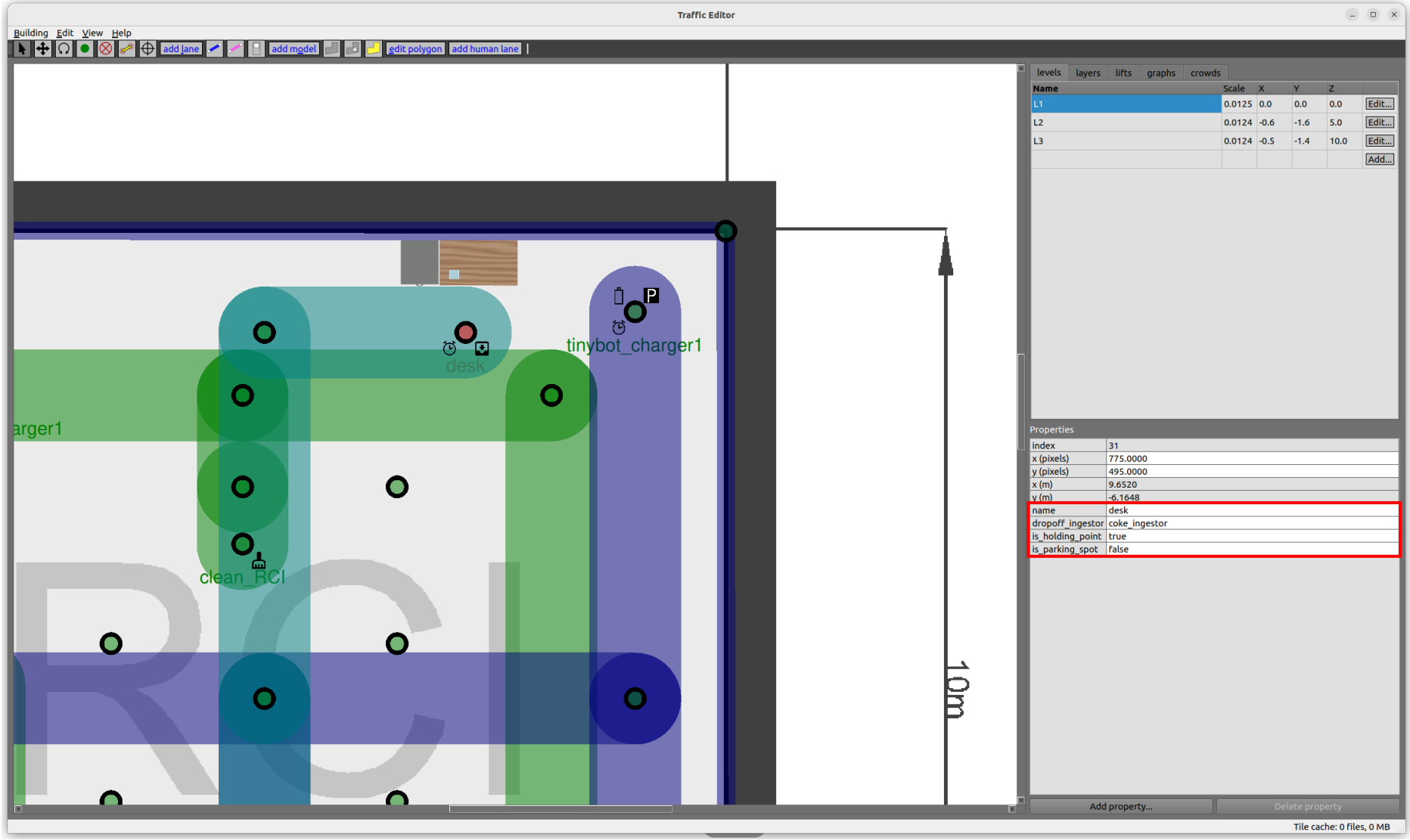
# Custom world Task 구성: Delivery

- rcilab world Delivery pick up 지점 추가



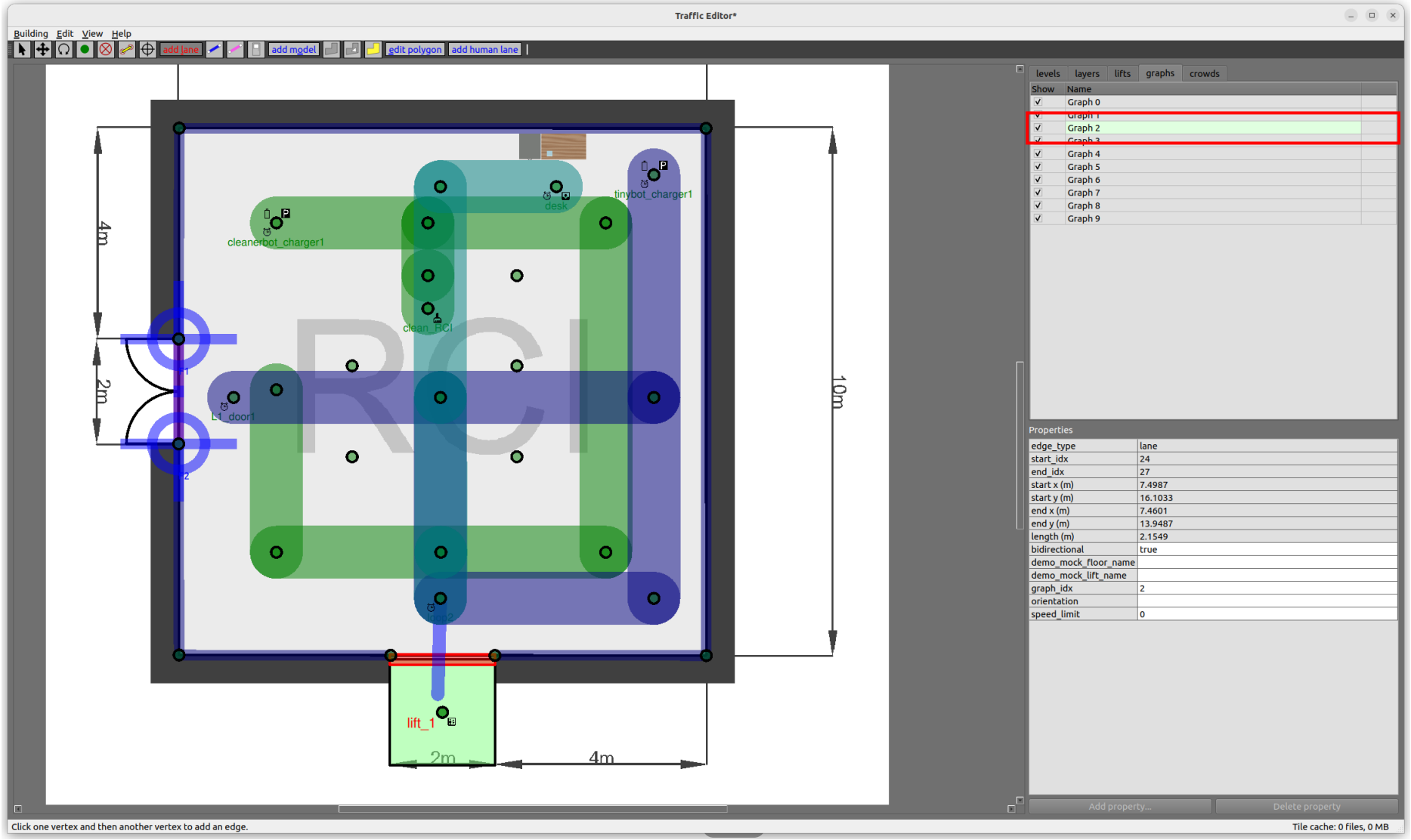
# Custom world Task 구성: Delivery

- rcilab world Delivery drop off 지점 추가



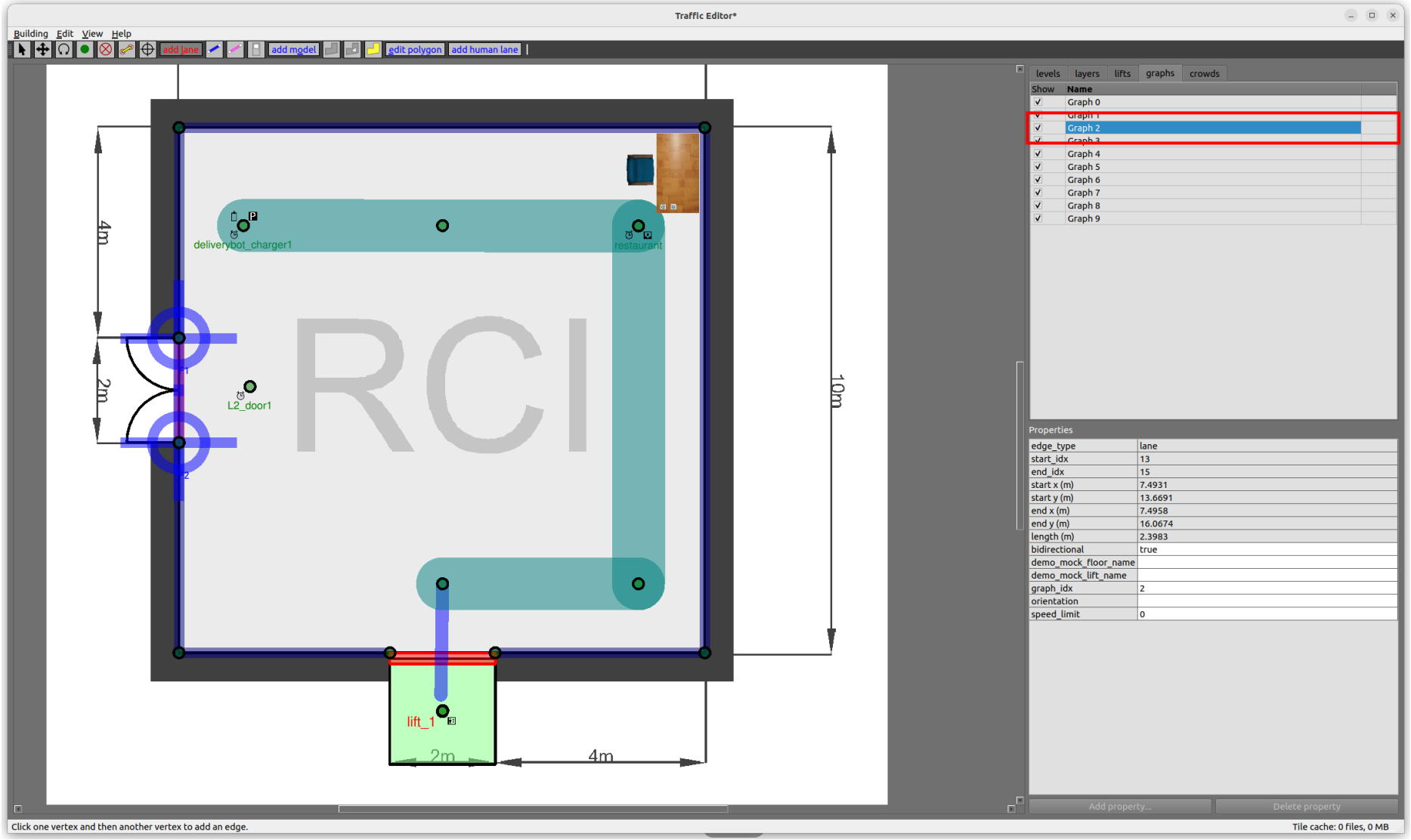
# Custom world Task 구성: Delivery

- rcilab world Delivery L1 경로 추가



# Custom world Task 구성: Delivery

- rcilab world Delivery L2 경로 추가



# Custom world Task 구성: Delivery

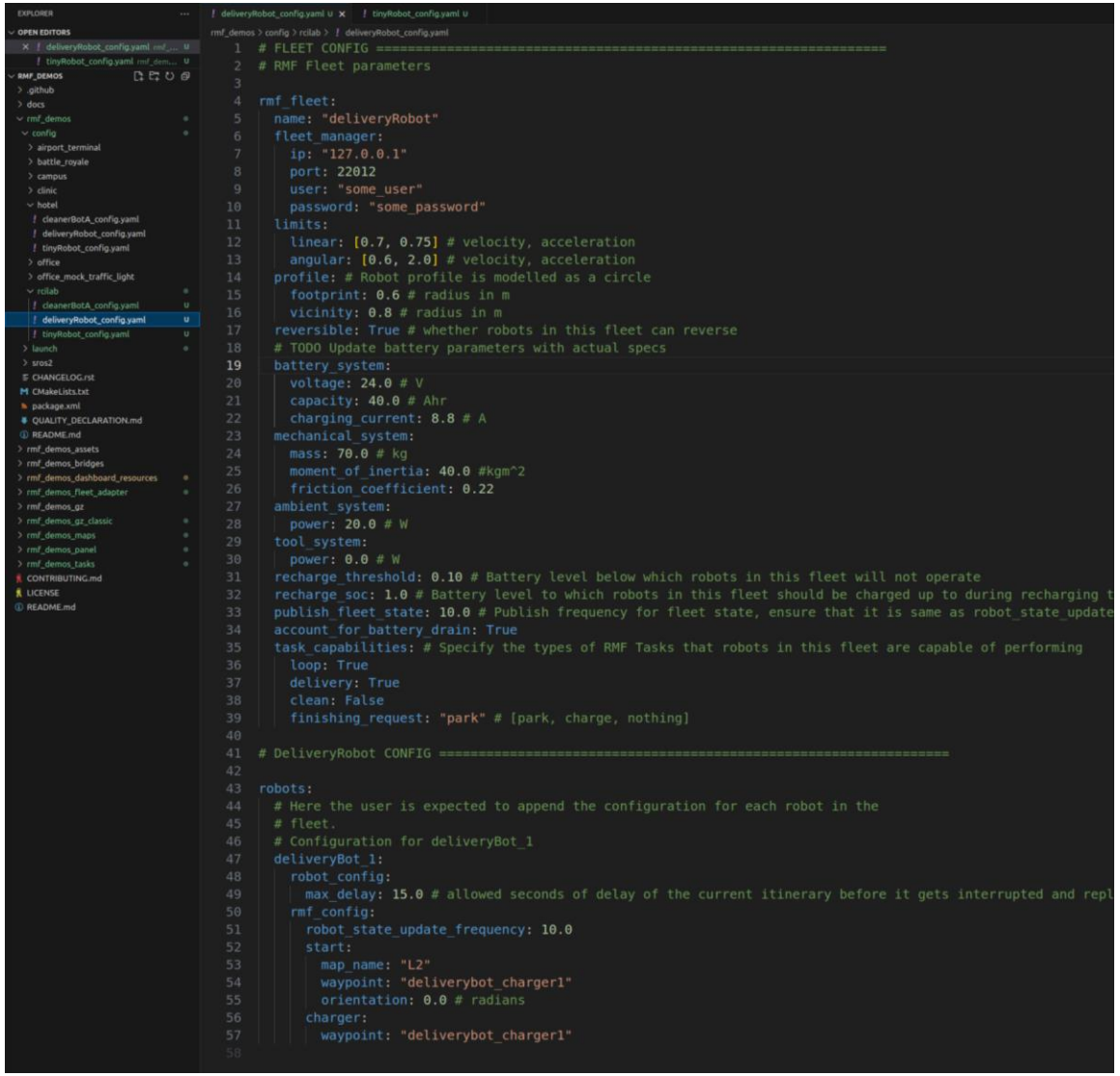
- rcilab deliveryRobot 경로 추가 (config 파일 생성)
  - deliveryRobot_config.yaml 파일 생성

```
~/rmf_ws/src/rmf_demos/rmf_demos/config/deliveryRobot_config.yaml
```

# deliveryRobot_config.yaml 내용

```yaml
# FLEET CONFIG ================================================================
# RMF Fleet parameters
rmf_fleet:
  name: "deliveryRobot"
  fleet_manager:
    ip: "127.0.0.1"
    port: 22012
    user: "some_user"
    password: "some_password"
  limits:
    linear: [0.7, 0.75] # velocity, acceleration
    angular: [0.6, 2.0] # velocity, acceleration
  profile: # Robot profile is modelled as a circle
    footprint: 0.6 # radius in m
    vicinity: 0.8 # radius in m
  reversible: True # whether robots in this fleet can reverse
  # TODO Update battery parameters with actual specs
  battery_system:
    voltage: 24.0 # V
    capacity: 40.0 # Ahr
    charging_current: 8.8 # A
  mechanical_system:
    mass: 70.0 # kg
    moment_of_inertia: 40.0 #kgm^2
    friction_coefficient: 0.22
  ambient_system:
    power: 20.0 # W
  tool_system:
    power: 0.0 # W
  recharge_threshold: 0.10 # Battery level below which robots in this fleet will not operate
  recharge_soc: 1.0 # Battery level to which robots in this fleet should be charged up to during recharging tasks
  publish_fleet_state: 10.0 # Publish frequency for fleet state, ensure that it is same as robot_state_update_frequency
  account_for_battery_drain: True
  task_capabilities: # Specify the types of RMF Tasks that robots in this fleet are capable of performing
    loop: True
    delivery: True
    clean: False
    finishing_request: "park" # [park, charge, nothing]

# DeliveryRobot CONFIG
# ================================================================
robots:
  # Here the user is expected to append the configuration for each robot in the
  # fleet.
  # Configuration for deliveryBot_1
  deliveryBot_1:
    robot_config:
      max_delay: 15.0 # allowed seconds of delay of the current itinerary before it gets interrupted and replanned
    rmf_config:
      robot_state_update_frequency: 10.0
      start:
        map_name: "L2"
        waypoint: "deliverybot_charger1"
        orientation: 0.0 # radians
      charger:
        waypoint: "deliverybot_charger1"
```



# Custom world Task 구성: Delivery

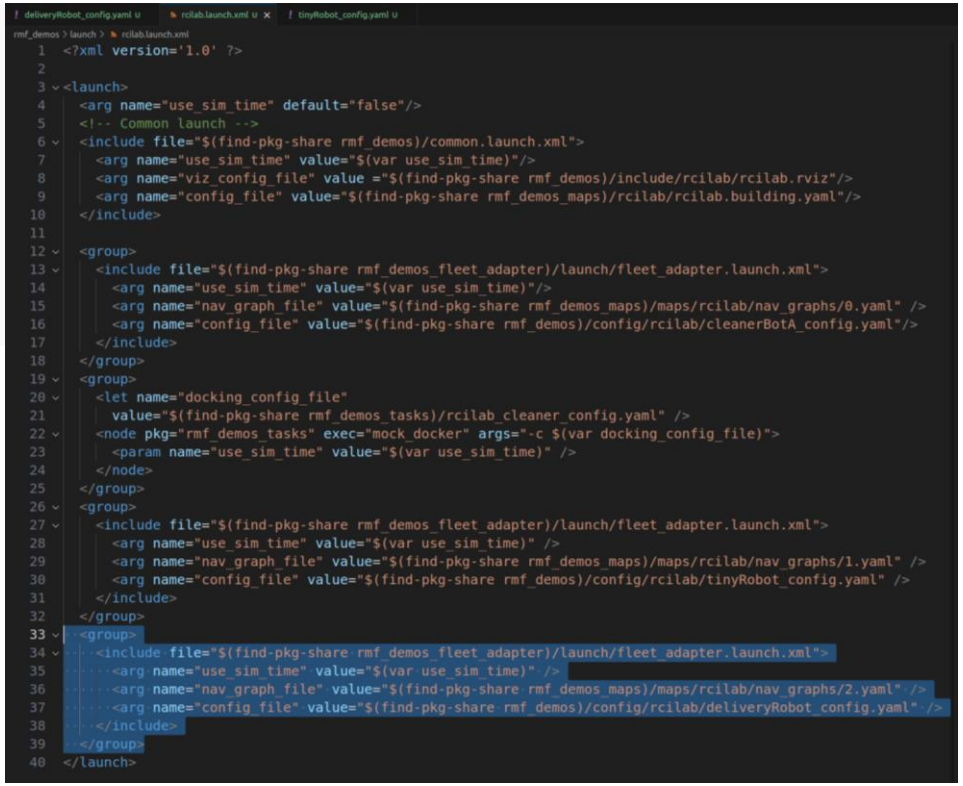
## launch 파일 수정

### rcilab.launch.xml 파일 수정

~/rmf_ws/src/rmf_demos/rmf_demos/launch/rcilab.launch.xml

### rcilab.launch.xml 추가 내용

```
<group>
  <include file="$(find-pkg-share rmf_demos_fleet_adapter)/launch/fleet_adapter.launch.xml">
    <arg name="use_sim_time" value="$(var use_sim_time)"/>
    <arg name="nav_graph_file" value="$(find-pkg-share rmf_demos_maps)/maps/rcilab/nav_graphs/2.yaml" />
    <arg name="config_file" value="$(find-pkg-share rmf_demos)/config/rcilab/deliveryRobot_config.yaml"/>
  </include>
</group>
```



# Custom world Task 구성: Delivery

- rcilab world Delivery Task 확인
  - Build

```
cd ~/rmf_ws
colcon build
```

  - rcilab.launch.xml 실행

```
cd ~/rmf_ws && source install/setup.bash
```

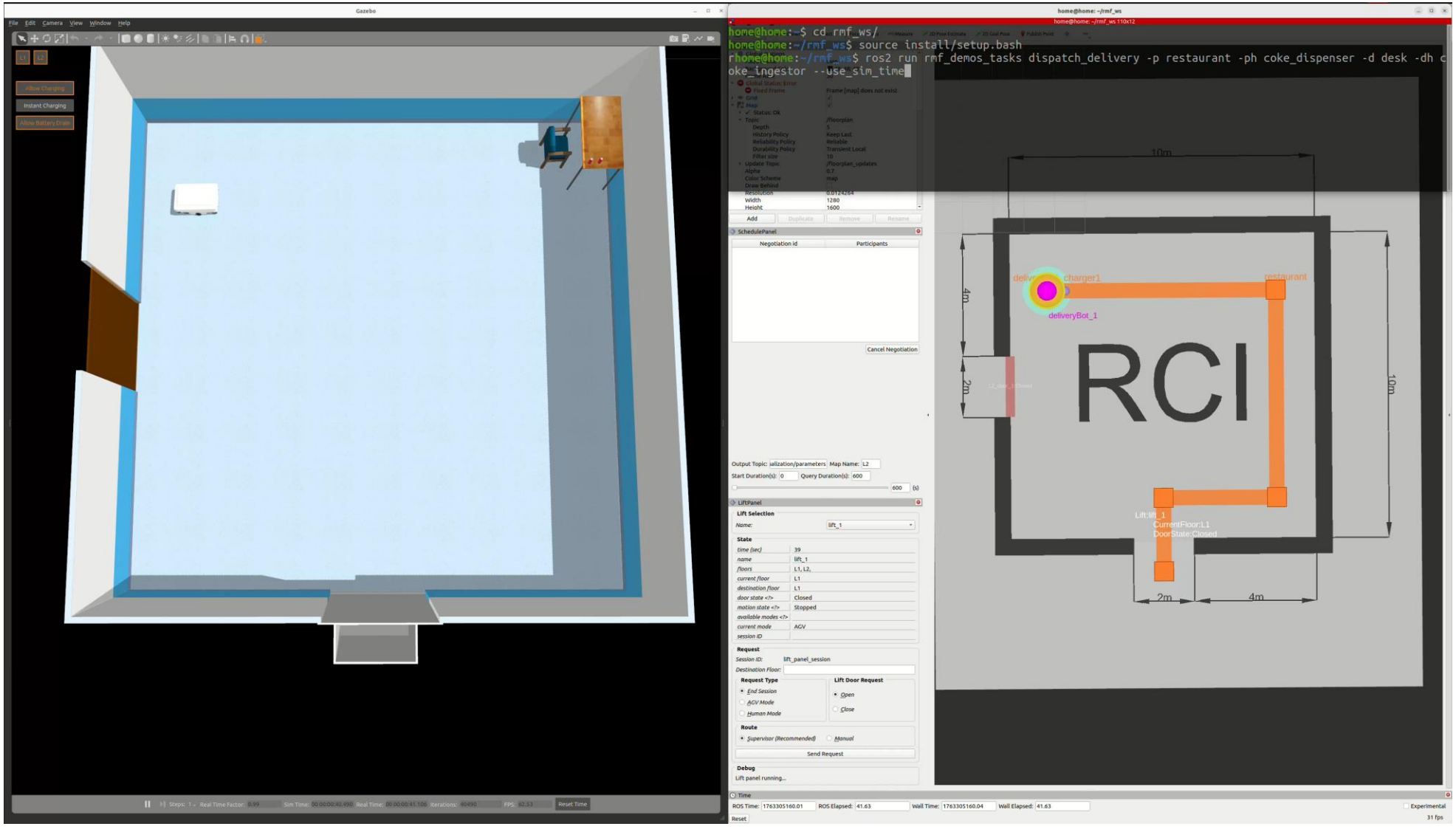
```
ros2 launch rmf_demos_gz_classic rcilab.launch.xml
```

  - Patrol Task 명령

```
ros2 run rmf_demos_tasks dispatch_delivery –p restaurant –ph coke_dispenser –d desk –dh coke_ingestor --use_sim_time
```

# Custom world Task 구성: Delivery

- rcilab world Delivery Task 확인



# 감사합니다