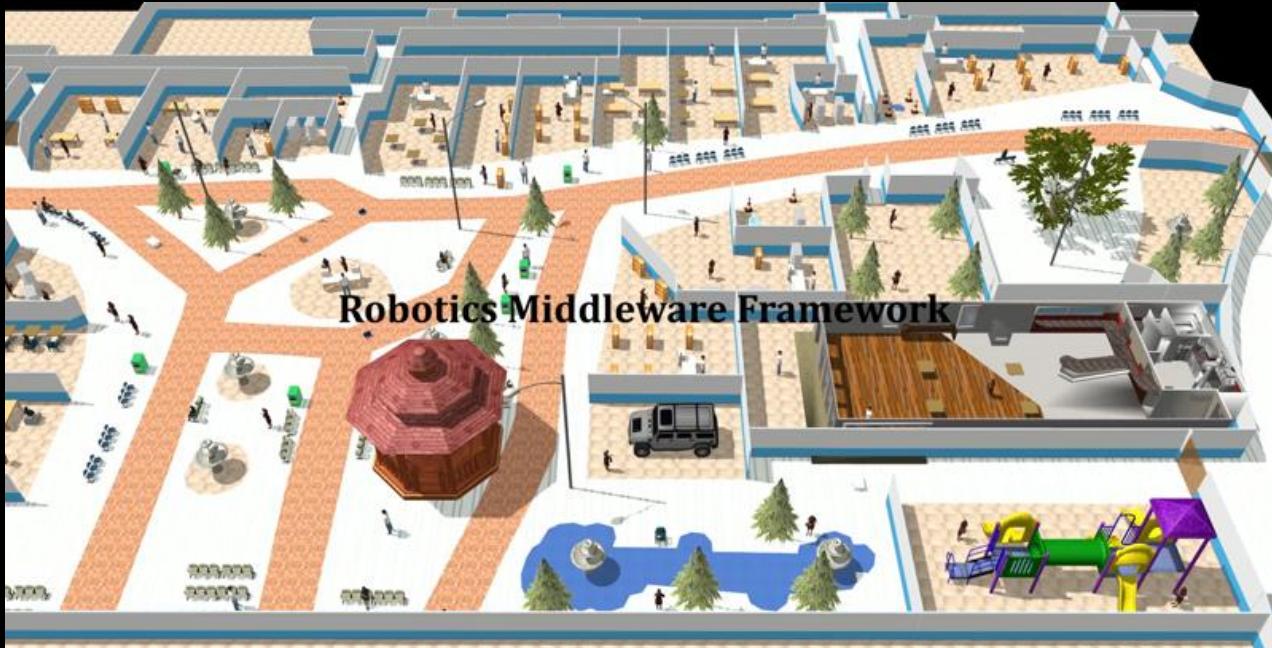


# Open-RMF

A Common Language for Robot Interoperability



## Lecture 2

정은빈

# Contents

01 RMF Demo 소개

02 Office world  
Demo

03 RMF Panel

# **RMF Demo 소개**

# RMF Demos worlds

## » Demos Worlds

```
cd ~/rmf_ws/src/rmf_demos/rmf_demos_maps/maps
```



airport\_terminal



battle\_royale



campus



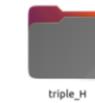
clinic



hotel



office



triple\_H

# RMF Demos worlds

## » Demos Worlds

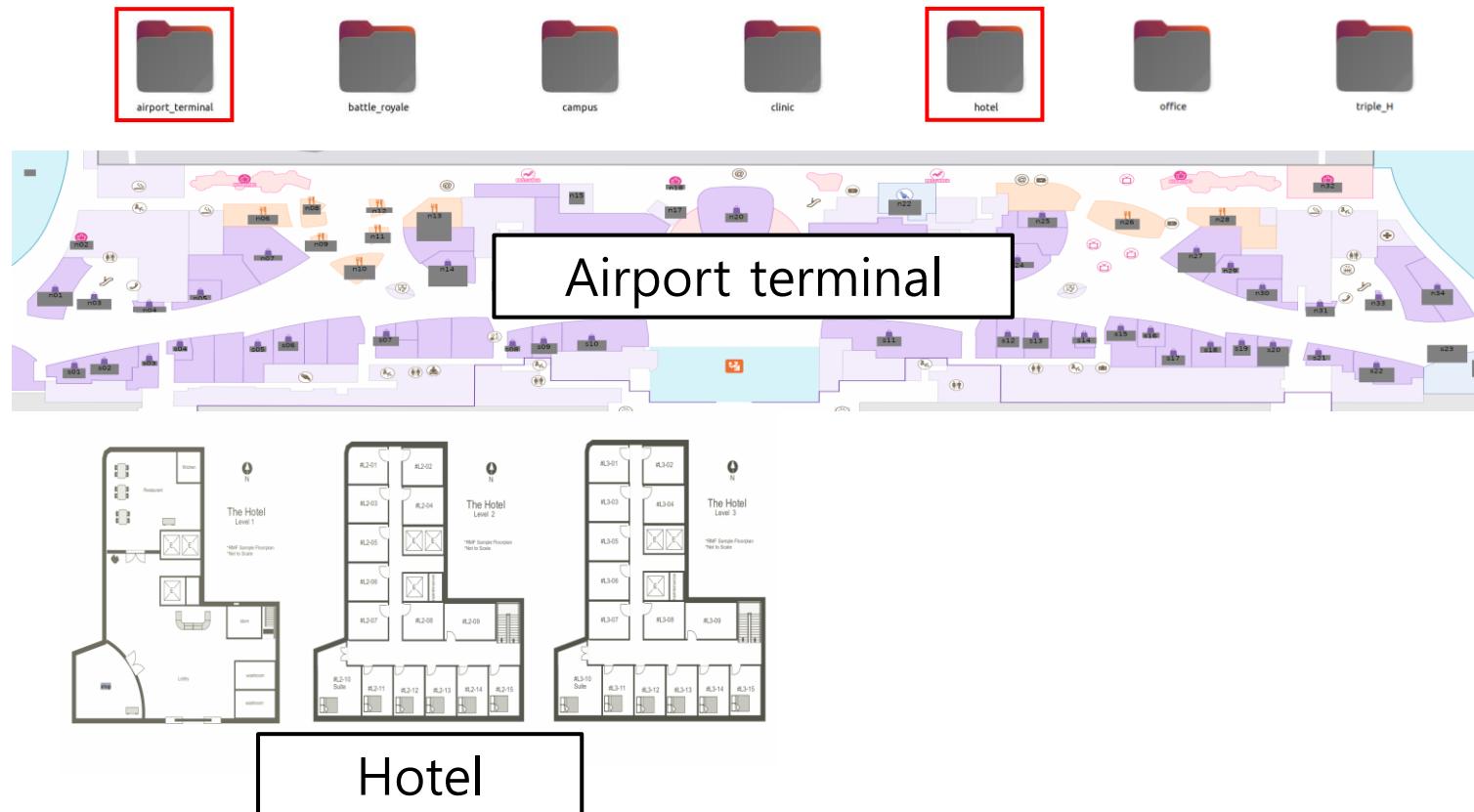
```
cd ~/rmf_ws/src/rmf_demos/rmf_demos_maps/maps
```



# RMF Demos worlds

## » Demos Worlds

```
cd ~/rmf_ws/src/rmf_demos/rmf_demos_maps/maps
```



# RMF Demos worlds

## » Demos Worlds

```
cd ~/rmf_ws/src/rmf_demos/rmf_demos_maps/maps
```



airport\_terminal



battle\_royale



campus



clinic



hotel



office



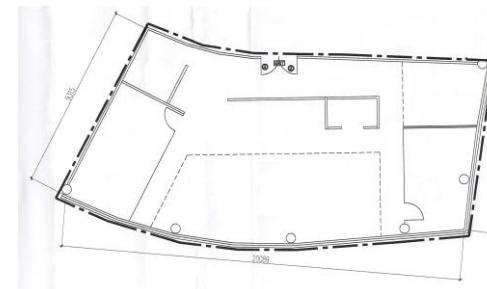
triple\_H



Airport terminal



Hotel



Office

# RMF Demos worlds

## » Demos Task

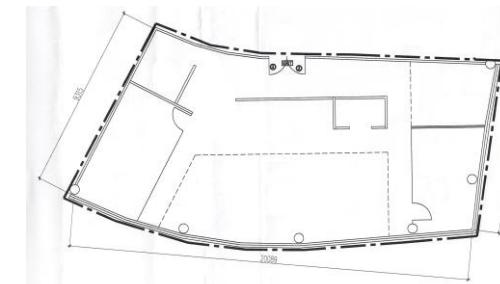
Task: World 내에서 로봇이 수행하는 업무

### | Task 종류

- Delivery
- Patrol
- Clean



Hotel



Office

# **Office World Demo**

# Office World Demo

---

## » Office world 실행

### | 환경 불러오기

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

### | Classic Gazebo로 office world 실행

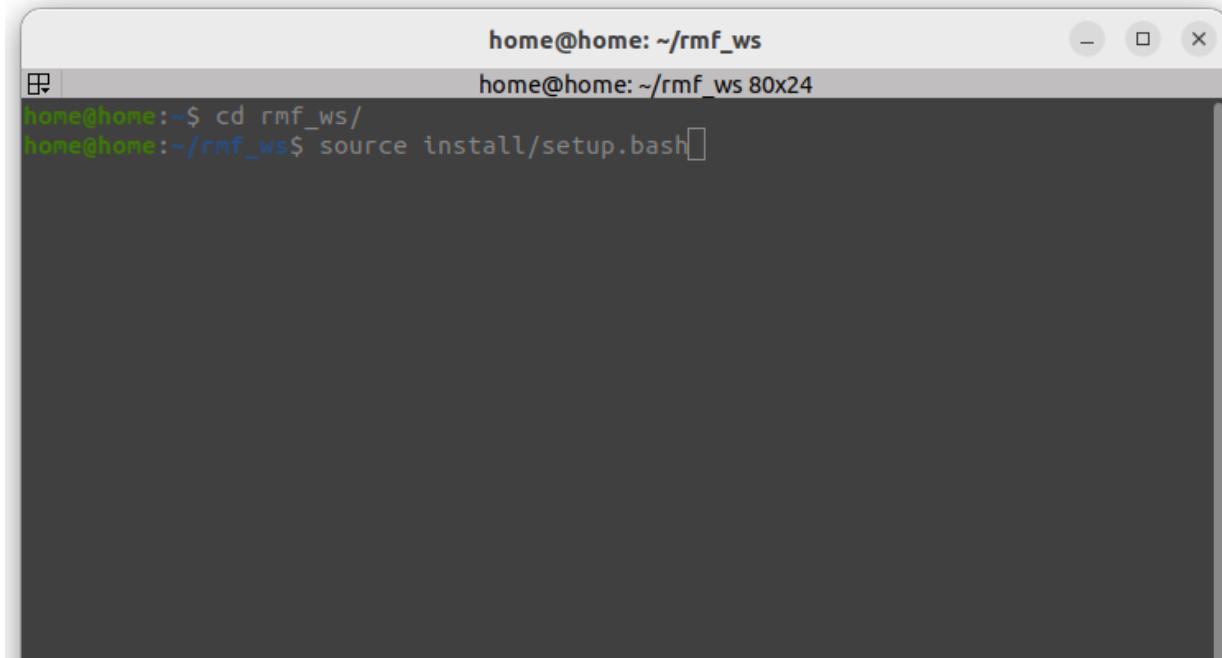
```
ros2 launch rmf_demos_gz_classic office.launch.xml
```

# Office World Demo

## » Office world 실행

### | 환경 불러오기

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



A screenshot of a terminal window titled "home@home: ~/rmf\_ws". The window shows the command "source install/setup.bash" being typed at the prompt. The terminal has a dark background with light-colored text.

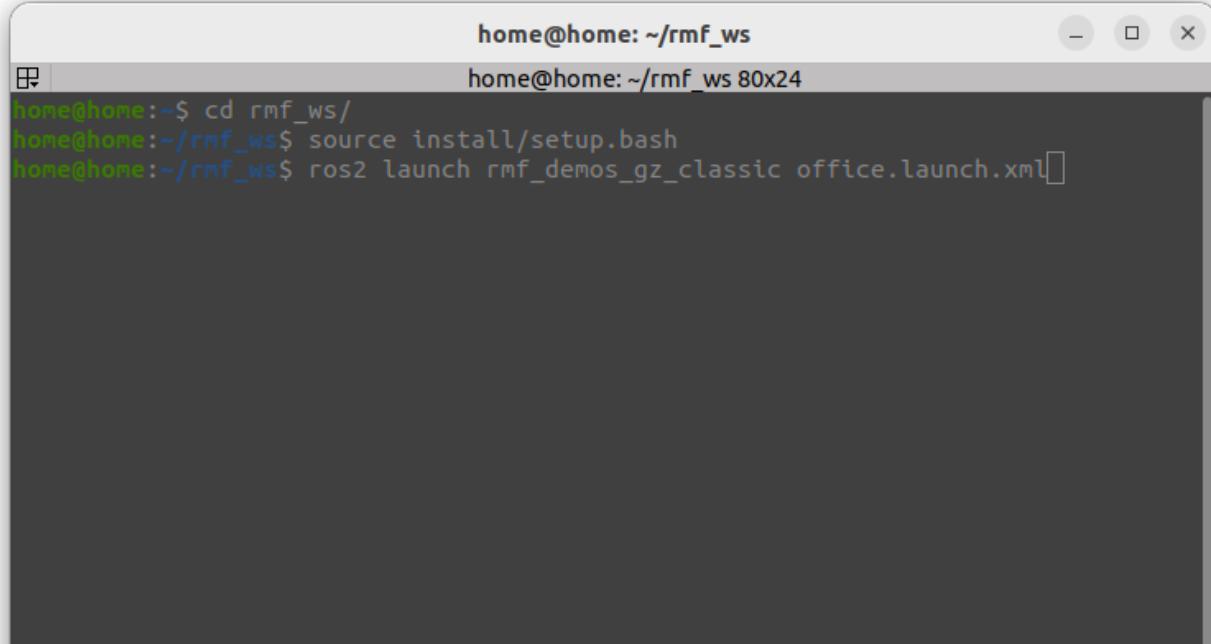
```
home@home: ~/rmf_ws
home@home: ~/rmf_ws 80x24
home@home:~$ cd rmf_ws/
home@home:~/rmf_ws$ source install/setup.bash
```

# Office World Demo

## ➤ Office world 실행

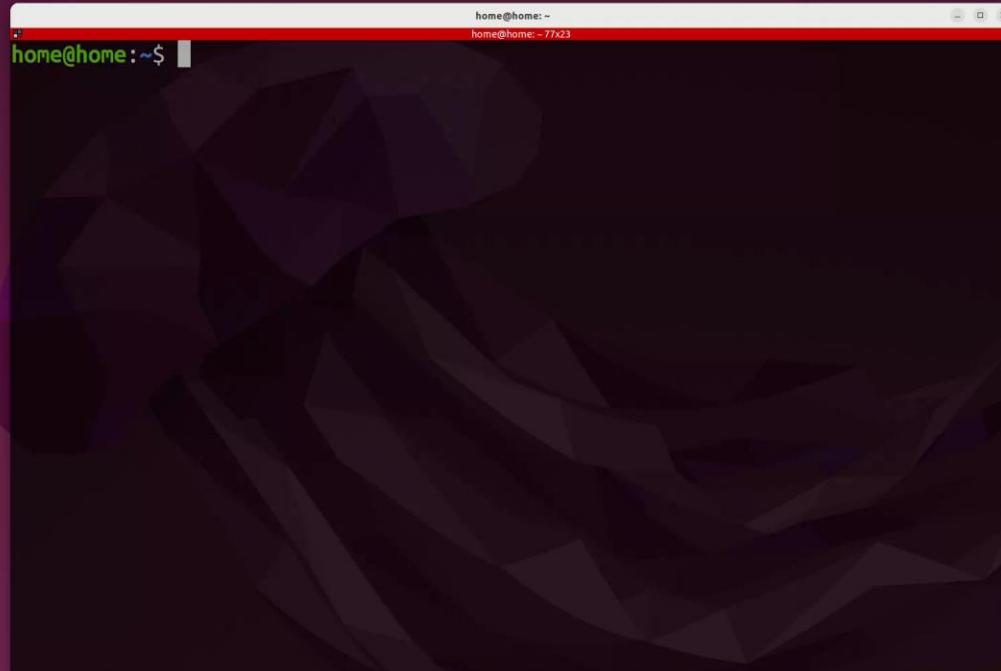
### | Classic Gazebo로 office world 실행

```
ros2 launch rmf_demos_gz_classic office.launch.xml
```



A screenshot of a terminal window titled "home@home: ~/rmf\_ws". The window shows a command-line session where the user has navigated to their workspace directory, run the setup script, and then executed a ROS 2 launch command to start the "office.launch.xml" file from the "rmf\_demos\_gz\_classic" package.

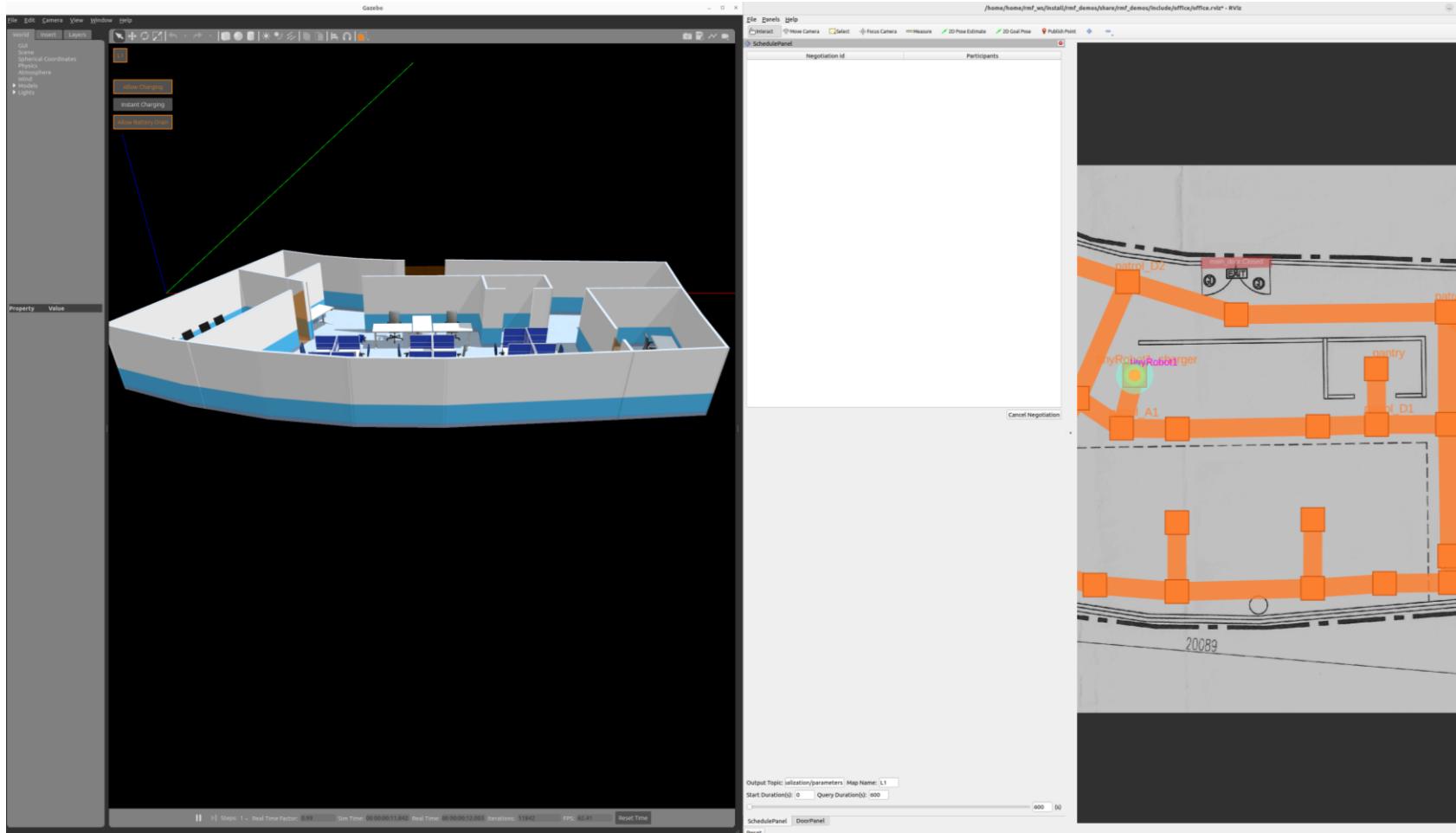
```
home@home:~/rmf_ws
home@home:~/rmf_ws 80x24
home@home:~$ cd rmf_ws/
home@home:~/rmf_ws$ source install/setup.bash
home@home:~/rmf_ws$ ros2 launch rmf_demos_gz_classic office.launch.xml
```



# Office World Demo

## ⦿ Office world 실행

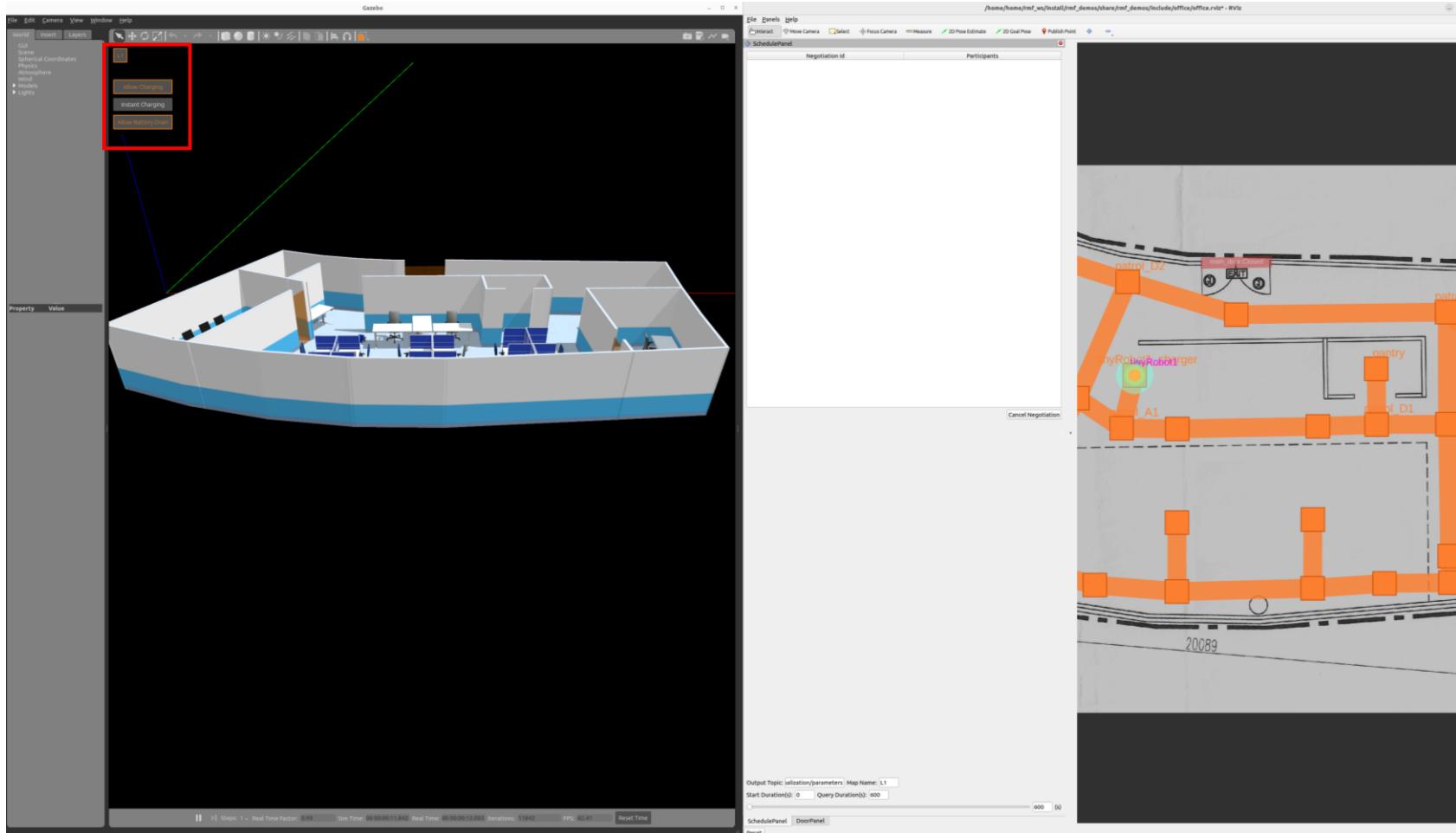
### | Classic Gazebo로 office world 실행



# Office World Demo

## ➤ Office world 실행

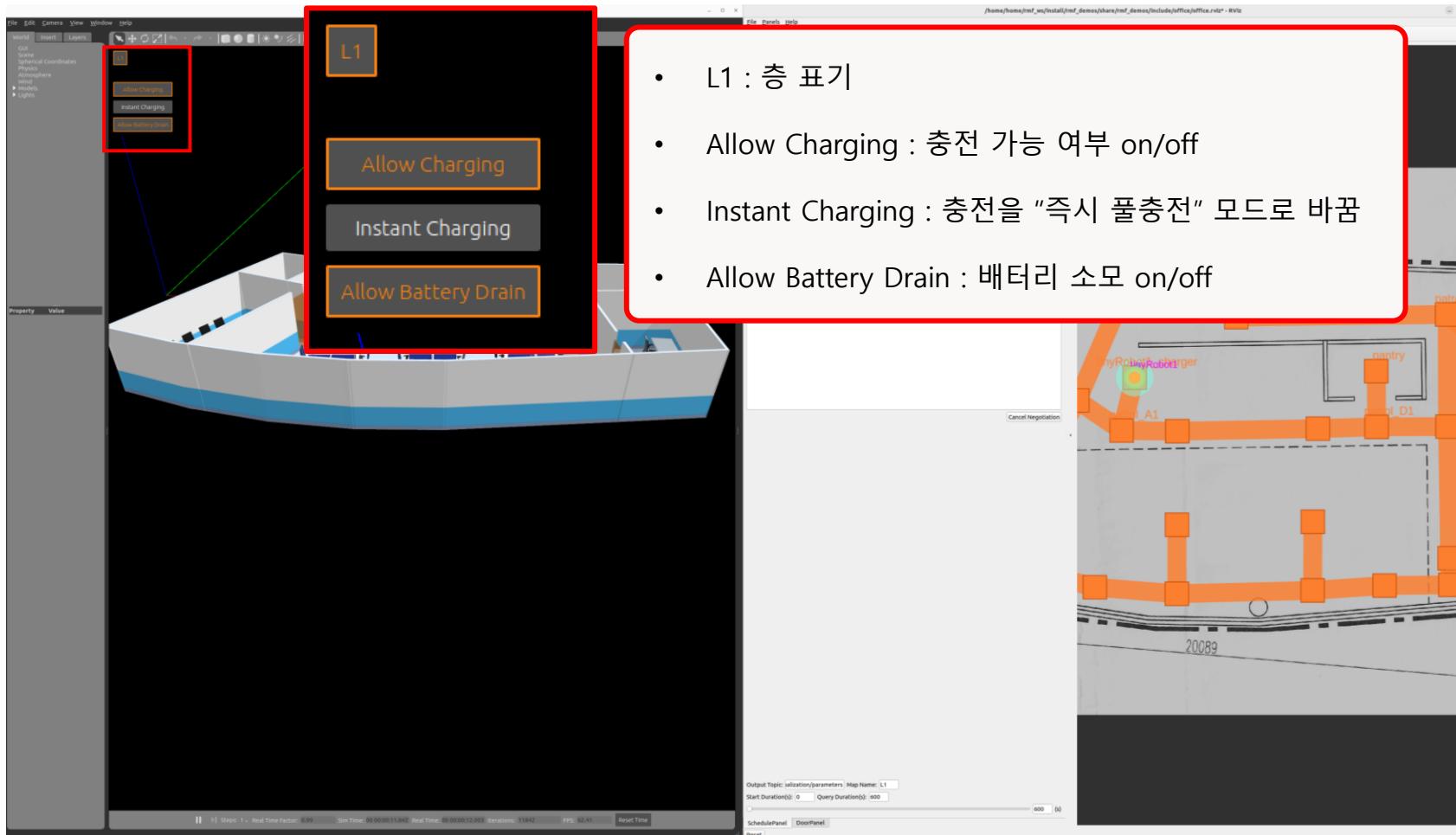
### | Classic Gazebo로 office world 실행



# Office World Demo

## ➤ Office world 실행

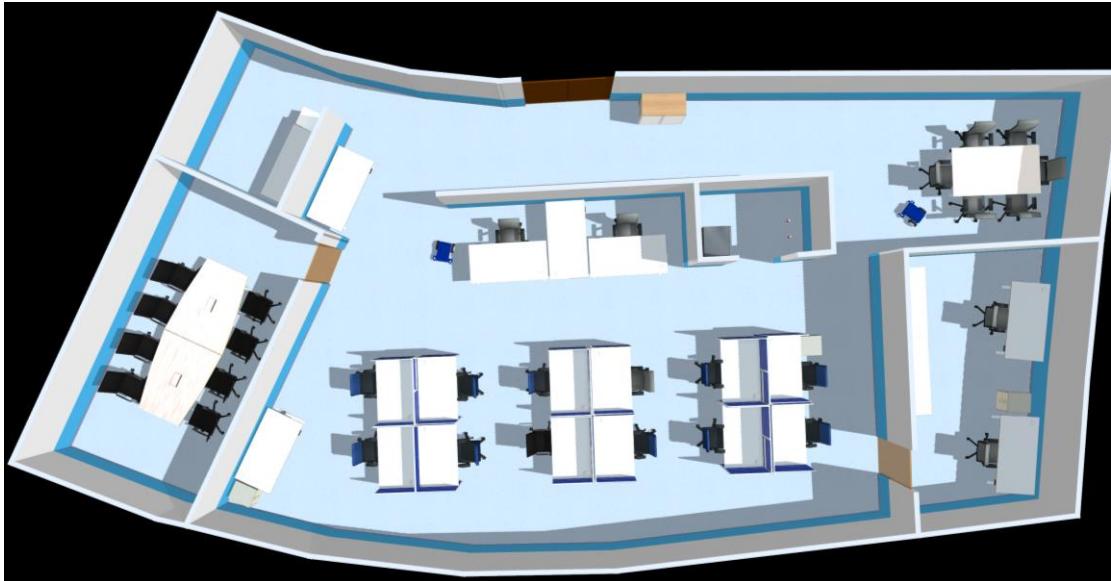
### | Classic Gazebo로 office world 실행



# Office World Demo

## ⦿ Office world 실행

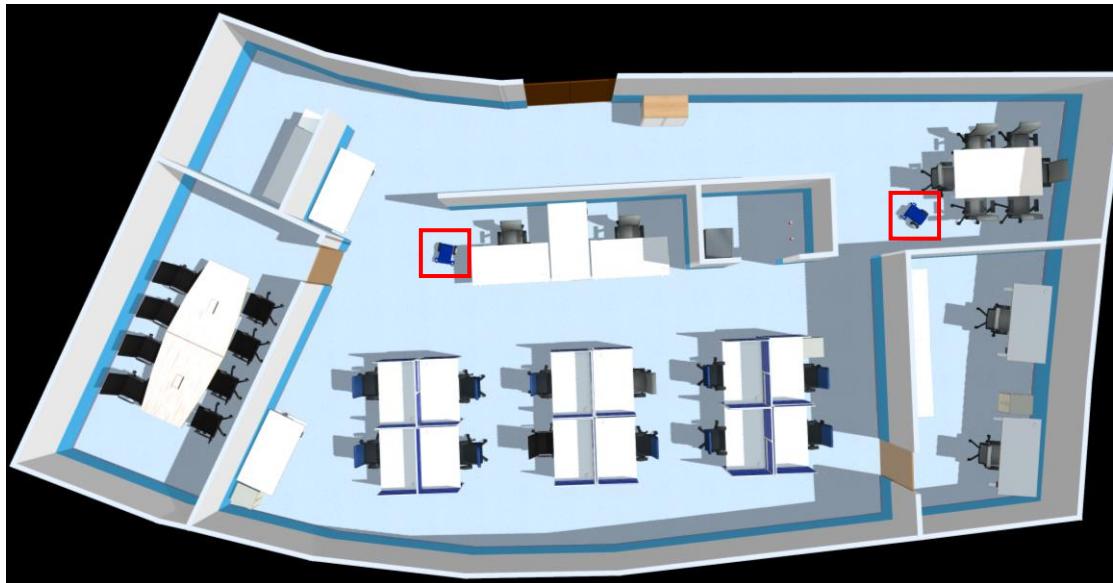
### | Office world 설명



# Office World Demo

## ⦿ Office world 실행

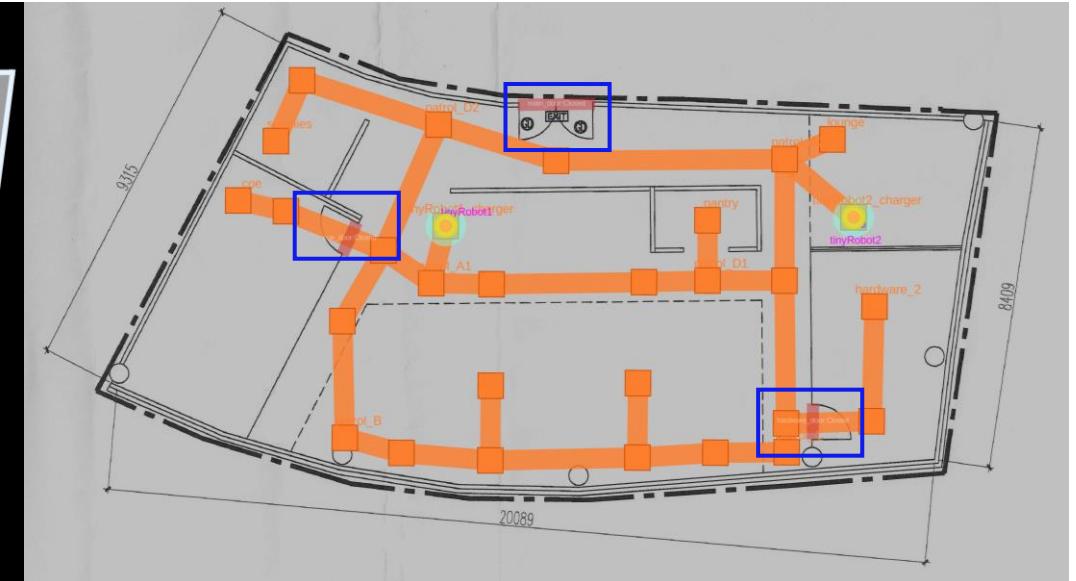
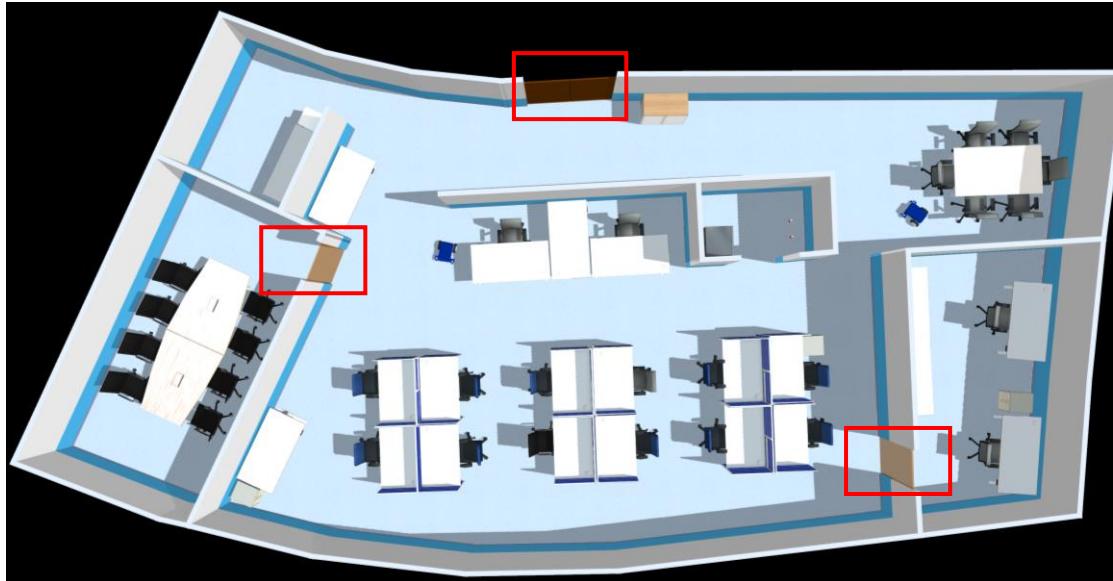
### | Office world 설명



# Office World Demo

## ⦿ Office world 실행

### | Office world 설명



# Office World Demo

## ④ Topics

### | Robot: 로봇의 위치, 배터리 상태, 작업 진행 상태

- Topics: /fleets\_states

### | Environment:

- Door: /door\_requests, /door\_states, /door\_supervisor\_heartbeat
- Lift: /lift\_requests, /lift\_states

### | RMF Task: RMF 시스템 관련 작업 관리

- Topics: /rmf\_task/bid\_notice, /rmf\_task/dispatch\_request

### | RMF Traffic: 로봇들 사이의 원활한 트래픽 흐름과 경로 계획 관리

- Topics: /rmf\_traffic/blockade\_, /rmf\_traffic/itinerary\_,  
/rmf\_traffic/negotiation\_

```
home@home:~$ ros2 topic list
/action_execution_notice
/adapter_door_requests
/adapter_lift_requests
/building_systems_markers
/clicked_point
/clock
/closed_lanes
/dispatch_states
/dispenser_requests
/dispenser_results
/dispenser_states
/dock_summary
/door_requests
/door_states
/door_supervisor_heartbeat
/fire_alarm_trigger
/fleet_markers
/fleet_states
/floorplan
/floorplan_updates
/goal_pose
/ingestor_requests
/ingestor_results
/ingestor_states
/initialpose
/lane_closure_requests
/lane_states
/lift_requests
/lift_states
/map
/map_markers
/nav_graphs
/parameter_events
/performance_metrics
/rmf_obstacles
/rmf_task/bid_notice
/rmf_task/bid_response
/rmf_task/dispatch_ack
/rmf_task/dispatch_request
/rmf_traffic/blockade_cancel
/rmf_traffic/blockade_heartbeat
/rmf_traffic/blockade_reached
/rmf_traffic/blockade_ready
/rmf_traffic/blockade_release
/rmf_traffic/blockade_set
/rmf_traffic/hearbeat
/rmf_traffic/itinerary_clear
/rmf_traffic/itinerary_delay
/rmf_traffic/itinerary_extend
/rmf_traffic/itinerary_reached
/rmf_traffic/itinerary_set
/rmf_traffic/negotiation_ack
/rmf_traffic/negotiation_conclusion
/rmf_traffic/negotiation_forfeit
/rmf_traffic/negotiation_notice
/rmf_traffic/negotiation_proposal
/rmf_traffic/negotiation_refusal
/rmf_traffic/negotiation_rejection
/rmf_traffic/negotiation_repeat
/rmf_traffic/negotiation_states
/rmf_traffic/negotiation_statuses
/rmf_traffic/participants
/rmf_traffic/query_update_1
/rmf_traffic/registered_queries
/rmf_traffic/schedule_inconsistency
/rmf_traffic/schedule_startup
/rmf_visualization/parameters
/robot_mode_requests
/robot_path_requests
/robot_pause_requests
/robot_state
/rosout
/schedule_markers
/site_map
/task_apl_requests
/task_apl_responses
/task_summaries
/tf
/tf_static
/home@home:~$
```

# Office World Demo

## ⦿ Delivery Task 실행

### | 환경 불러오기

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

### | Delivery Task 명령

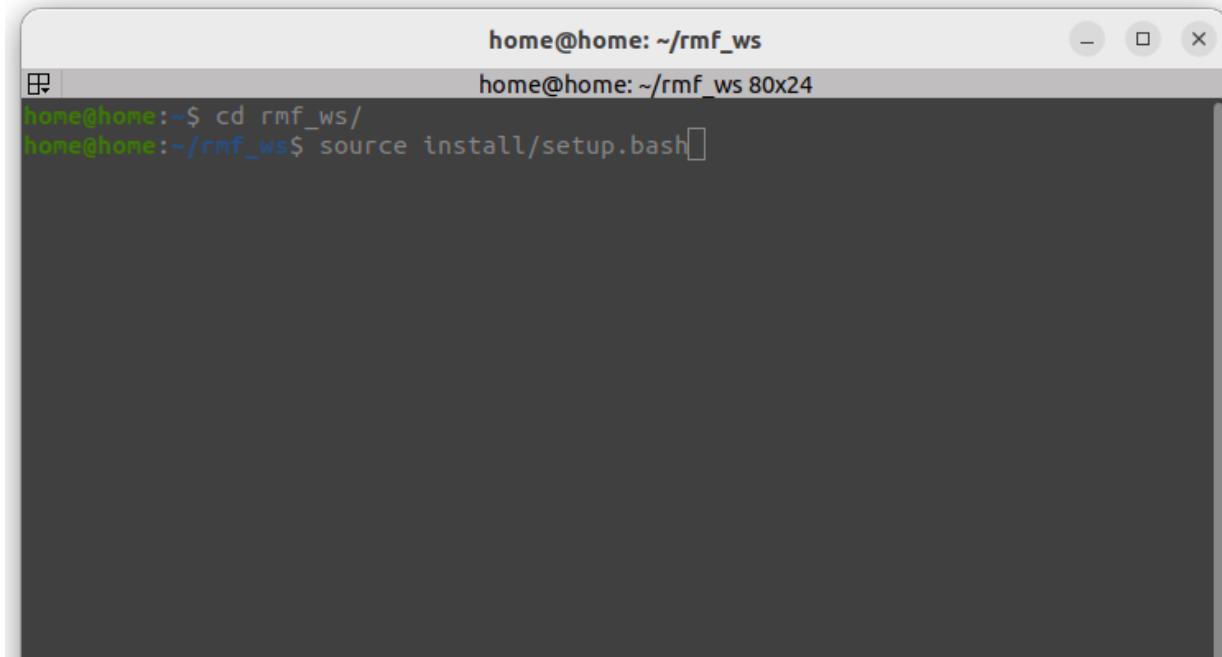
```
ros2 run rmf_demos_tasks dispatch_delivery -p pantry -ph coke_dispenser -d hardware_2 -dh coke_ingestor --use_sim_time
```

# Office World Demo

## ⦿ Delivery Task 실행

### | 환경 불러오기

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



A screenshot of a terminal window titled "home@home: ~/rmf\_ws". The window shows the command "cd ~/rmf\_ws" followed by "source install/setup.bash" being entered at the prompt. The terminal has a dark background and light-colored text.

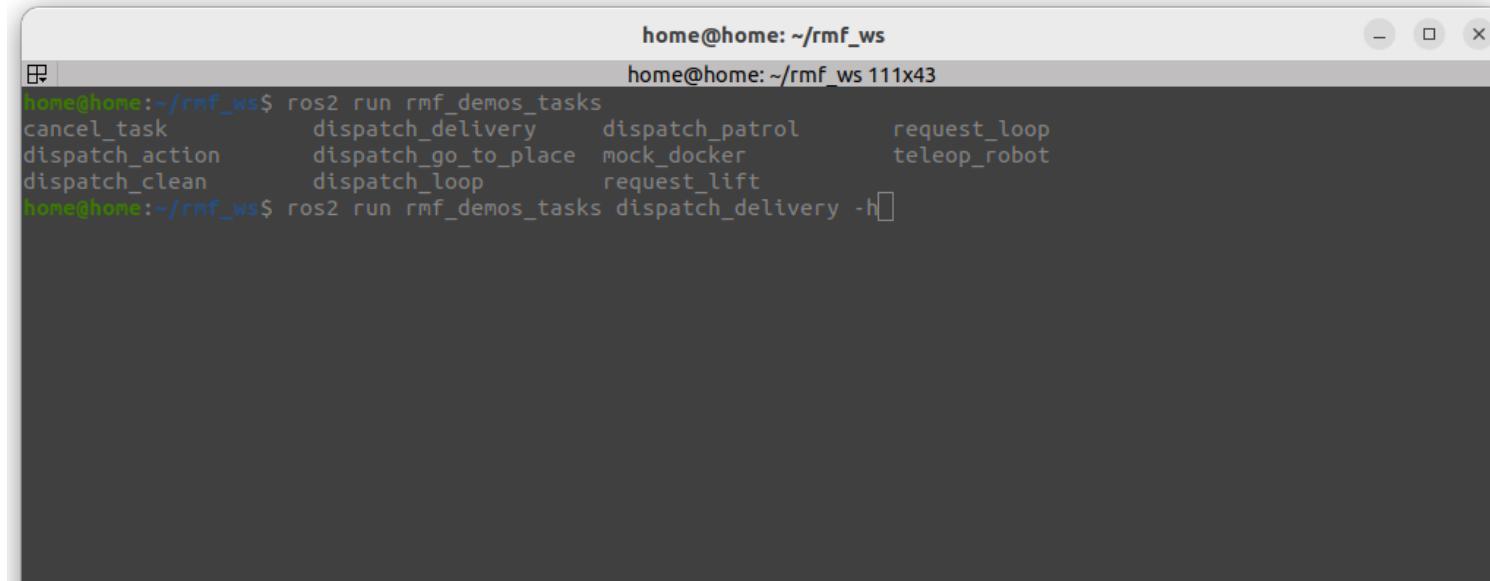
```
home@home: ~/rmf_ws
home@home: ~/rmf_ws 80x24
home@home:~$ cd rmf_ws/
home@home:~/rmf_ws$ source install/setup.bash
```

# Office World Demo

## ⦿ Delivery Task 실행

### | Delivery Task 명령

```
ros2 run rmf_demos_tasks dispatch_delivery -p pantry -ph coke_dispenser -d hardware_2 -dh coke_ingestor --use_sim_time
```



A screenshot of a terminal window titled "home@home: ~/rmf\_ws". The window shows a command-line interface with the following text:

```
home@home:~/rmf_ws$ ros2 run rmf_demos_tasks
cancel_task      dispatch_delivery    dispatch_patrol      request_loop
dispatch_action   dispatch_go_to_place  mock_docker        teleop_robot
dispatch_clean    dispatch_loop       request_lift
home@home:~/rmf_ws$ ros2 run rmf_demos_tasks dispatch_delivery -h
```

# Office World Demo

## ⦿ Delivery Task 실행

### I Delivery Task 명령

```
ros2 run rmf_demos_tasks dispatch_delivery -p pantry -ph coke_dispenser -d hardware_2 -dh coke_ingestor --use_sim_time
```

```
home@home:~/rmf_ws$ ros2 run rmf_demos_tasks dispatch_delivery -h
usage: dispatch_delivery [-h] -p PICKUPS [PICKUPS ...] -d DROPOFFS [DROPOFFS ...] -ph PICKUP_HANDLERS
                           [PICKUP_HANDLERS ...] -dh DROPOFF_HANDLERS [DROPOFF_HANDLERS ...]
                           [-pp PICKUP_PAYLOADS [PICKUP_PAYLOADS ...]]
                           [-dp DROPOFF_PAYLOADS [DROPOFF_PAYLOADS ...]] [-F FLEET] [-R ROBOT] [-st START_TIME]
                           [-pt PRIORITY] [--use_sim_time]

options:
  -h, --help            show this help message and exit
  -p PICKUPS [PICKUPS ...], --pickups PICKUPS [PICKUPS ...]
                        Pickup names
  -d DROPOFFS [DROPOFFS ...], --dropoffs DROPOFFS [DROPOFFS ...]
                        Dropoff names
  -ph PICKUP_HANDLERS [PICKUP_HANDLERS ...], --pickup_handlers PICKUP_HANDLERS [PICKUP_HANDLERS ...]
                        Pickup handler names
  -dh DROPOFF_HANDLERS [DROPOFF_HANDLERS ...], --dropoff_handlers DROPOFF_HANDLERS [DROPOFF_HANDLERS ...]
                        Dropoffs handler names
  -pp PICKUP_PAYLOADS [PICKUP_PAYLOADS ...], --pickup_payloads PICKUP_PAYLOADS [PICKUP_PAYLOADS ...]
                        Pickup payload [sku,quantity sku2,qty...]
  -dp DROPOFF_PAYLOADS [DROPOFF_PAYLOADS ...], --dropoff_payloads DROPOFF_PAYLOADS [DROPOFF_PAYLOADS ...]
                        Dropoff payload [sku,quantity sku2,qty...]
  -F FLEET, --fleet FLEET
                        Fleet name, should define tgt with robot
  -R ROBOT, --robot ROBOT
                        Robot name, should define tgt with fleet
  -st START_TIME, --start_time START_TIME
                        Start time from now in secs, default: 0
  -pt PRIORITY, --priority PRIORITY
                        Priority value for this request
  --use_sim_time        Use sim time, default: false
home@home:~/rmf_ws$
```

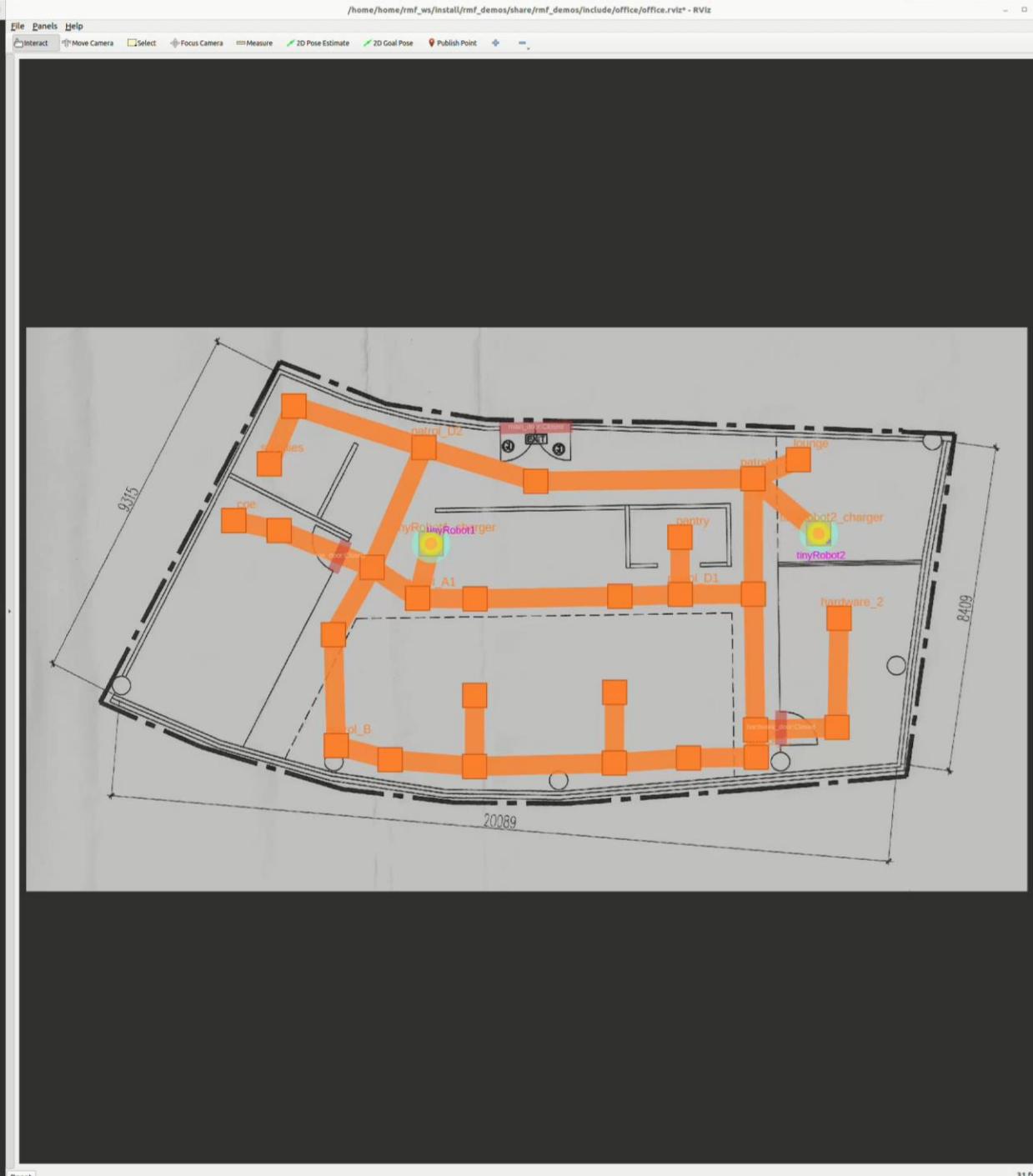
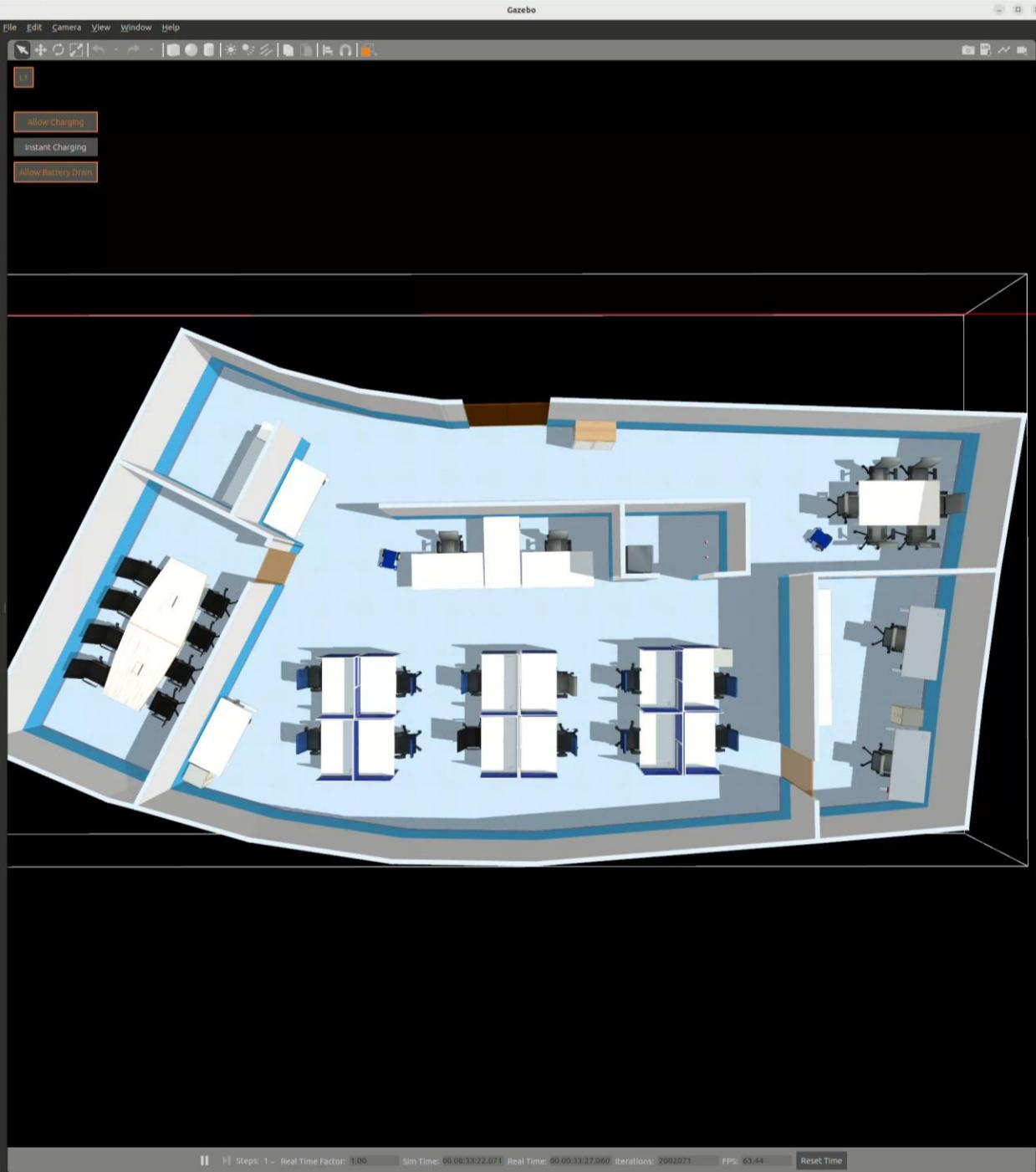
# Office World Demo

## ⦿ Delivery Task 실행

### I Delivery Task 명령

```
ros2 run rmf_demos_tasks dispatch_delivery -p pantry -ph coke_dispenser -d hardware_2 -dh coke_ingestor --use_sim_time
```

```
home@home:~/rmf_ws$ ros2 run rmf_demos_tasks dispatch_delivery -p pantry -ph coke_dispenser -d hardware_2 -dh coke_ingestor --use_sim_time
[INFO] [1763006475.234663088] [task_requester]: Using Sim Time
[INFO] [1763006475.235463970] [task_requester]: Using 'dispatch_task_request'
Json msg payload:
{
    "type": "dispatch_task_request",
    "request": {
        "unix_millis_earliest_start_time": 0,
        "category": "delivery",
        "description": {
            "pickup": {
                "place": "pantry",
                "handler": "coke_dispenser",
                "payload": []
            },
            "dropoff": {
                "place": "hardware_2",
                "handler": "coke_ingestor",
                "payload": []
            }
        }
    }
}
Got response:
{'state': {'booking': {'id': 'delivery.dispatch-4', 'unix_millis_earliest_start_time': 0}, 'category': 'delivery', 'detail': {'dropoff': {'handler': 'coke_ingestor', 'payload': [], 'place': 'hardware_2'}, 'pickup': {'handler': 'coke_dispenser', 'payload': [], 'place': 'pantry'}}, 'dispatch': {'errors': [], 'status': 'queued'}, 'status': 'queued', 'unix_millis_start_time': 0}, 'success': True}
home@home:~/rmf_ws$
```



# Office World Demo

## ⦿ Delivery Task 실행 log 확인

### | Task 할당 log

```
[fleet_adapter-16] [INFO] [1763010267.371189441] [tinyRobot_command_handle]: Robot tinyRobot2 has successfully navigated along requested path.  
[fleet_adapter-16] [INFO] [1763010267.834962847] [tinyRobot_command_handle]: Robot [tinyRobot1] has reached the destination for cmd_id 156  
[fleet_adapter-16] [INFO] [1763010267.848651099] [tinyRobot_command_handle]: Robot tinyRobot1 has successfully navigated along requested path.  
[rmf_task_dispatcher-13] [INFO] [1763010273.742820221] [rmf_dispatcher_node]: Add Task [delivery.dispatch-0] to a bidding queue  
[rmf_task_dispatcher-13] [INFO] [1763010273.821562558] [rmf_dispatcher_node]: - Start new bidding task: delivery.dispatch-0  
[fleet_adapter-16] [INFO] [1763010273.822128310] [tinyRobot_fleet_adapter]: [Bidder] Received Bidding notice for task_id [delivery.dispatch-0]  
[fleet_adapter-16] [INFO] [1763010273.822634538] [tinyRobot_fleet_adapter]: Planning for [2] robot(s) and [1] request(s)  
[fleet_adapter-16] [INFO] [1763010273.827292464] [tinyRobot_fleet_adapter]: Submitted BidProposal to accommodate task [delivery.dispatch-0] by robot [tinyRobot2] with new cost [2445.315016]  
[rmf_task_dispatcher-13] [INFO] [1763010275.821638018] [rmf_dispatcher_node]: Determined winning Fleet Adapter: [tinyRobot], from 1 responses  
[rmf_task_dispatcher-13] [INFO] [1763010275.821727887] [rmf_dispatcher_node]: Dispatcher Bidding Result: task [delivery.dispatch-0] is awarded to fleet adapter [tinyRobot], with expected robot [tinyRobot2].  
[fleet_adapter-16] [INFO] [1763010275.822171381] [tinyRobot_fleet_adapter]: Bid for task_id [delivery.dispatch-0] awarded to fleet [tinyRobot]. Processing request...  
[fleet_adapter-16] [INFO] [1763010275.825748707] [tinyRobot_fleet_adapter]: Assignments updated for robots in fleet [tinyRobot] to accommodate task_id [delivery.dispatch-0]  
[fleet_adapter-16] [INFO] [1763010275.826474474] [tinyRobot_fleet_adapter]: Beginning new task [delivery.dispatch-0] for [tinyRobot/tinyRobot2]. Remaining queue size: 1  
[fleet_adapter-16] [INFO] [1763010275.826924106] [tinyRobot_command_handle]: Requesting tinyRobot2 to stop...  
[fleet_adapter-16] [INFO] [1763010275.840819137] [tinyRobot_command_handle]: Received new path for tinyRobot2
```

### | 경로 추종 및 Task 수행 log

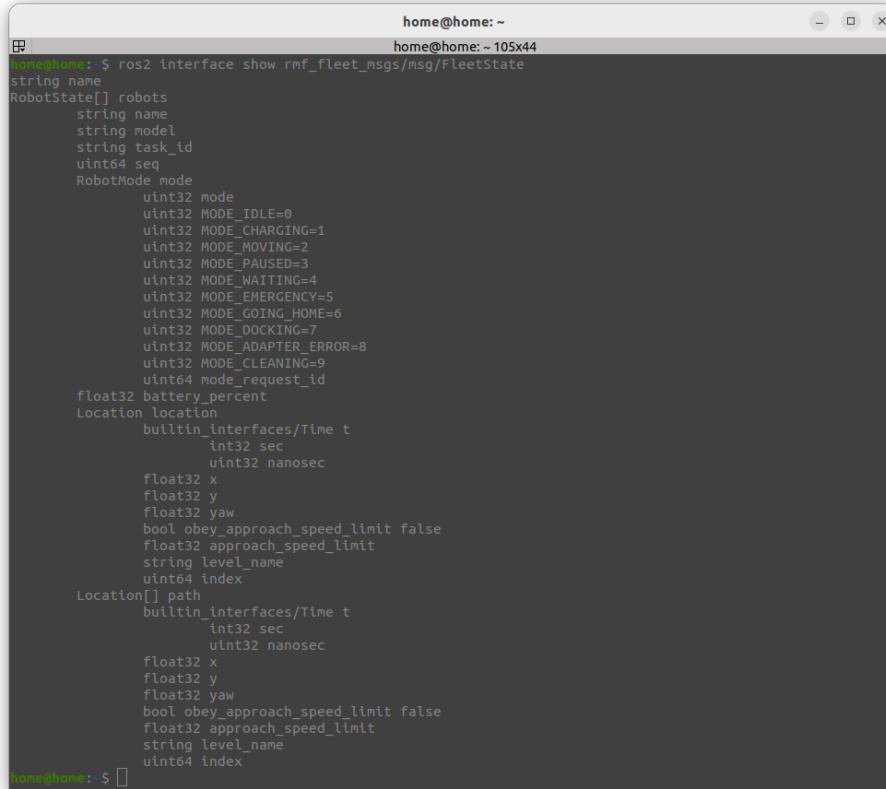
```
[fleet_adapter-16] [INFO] [1763010354.579464275] [tinyRobot_command_handle]: Received new path for tinyRobot1  
[fleet_adapter-16] [INFO] [1763010355.143711315] [tinyRobot_command_handle]: Robot [tinyRobot1] has reached the destination for cmd id 172  
[fleet_adapter-16] [INFO] [1763010356.118913877] [tinyRobot_command_handle]: Robot [tinyRobot2] has reached the destination for cmd_id 173  
[fleet_adapter-16] [INFO] [1763010356.124192956] [tinyRobot_command_handle]: Robot tinyRobot2 has successfully navigated along requested path.  
[fleet_adapter-16] [INFO] [1763010356.125844110] [tinyRobot_command_handle]: Received new path for tinyRobot2  
[fleet_adapter-16] [INFO] [1763010359.630450838] [tinyRobot_command_handle]: Robot [tinyRobot2] has reached the destination for cmd_id 174  
[fleet_adapter-16] [INFO] [1763010360.103722353] [tinyRobot_command_handle]: Robot [tinyRobot1] has reached the destination for cmd_id 173  
[gzserver-17] [WARN] [1763010360.110072947] [coke_dispenser]: No item to dispense: [delivery.dispatch-1]  
[fleet_adapter-16] [INFO] [1763010360.111275200] [tinyRobot_command_handle]: Robot tinyRobot1 has successfully navigated along requested path.  
[gzserver-17] [WARN] [1763010361.109841109] [coke_dispenser]: Request already failed: [delivery.dispatch-1]  
[fleet_adapter-16] [INFO] [1763010361.117741141] [tinyRobot_command_handle]: Received new path for tinyRobot1  
[fleet_adapter-16] [INFO] [1763010366.162596753] [tinyRobot_command_handle]: Robot [tinyRobot1] has reached the destination for cmd_id 174  
[fleet_adapter-16] [INFO] [1763010367.110456160] [tinyRobot_command_handle]: Robot [tinyRobot2] has reached the destination for cmd_id 175  
[gzserver-17] [INFO] [1763010367.116557690] [coke_ingestor_node]: Ingesting item  
[gzserver-17] [INFO] [1763010367.116762904] [coke_ingestor_node]: Success  
[fleet_adapter-16] [INFO] [1763010367.117269296] [tinyRobot_command_handle]: Robot tinyRobot2 has successfully navigated along requested path.  
[fleet_adapter-16] [INFO] [1763010369.120823201] [tinyRobot_fleet_adapter]: Beginning new task [ParkRobotfcf9b5] for [tinyRobot/tinyRobot2]. Remaining queue size: 0  
[fleet_adapter-16] [INFO] [1763010369.144763890] [tinyRobot_command_handle]: Received new path for tinyRobot2
```

# Office World Demo

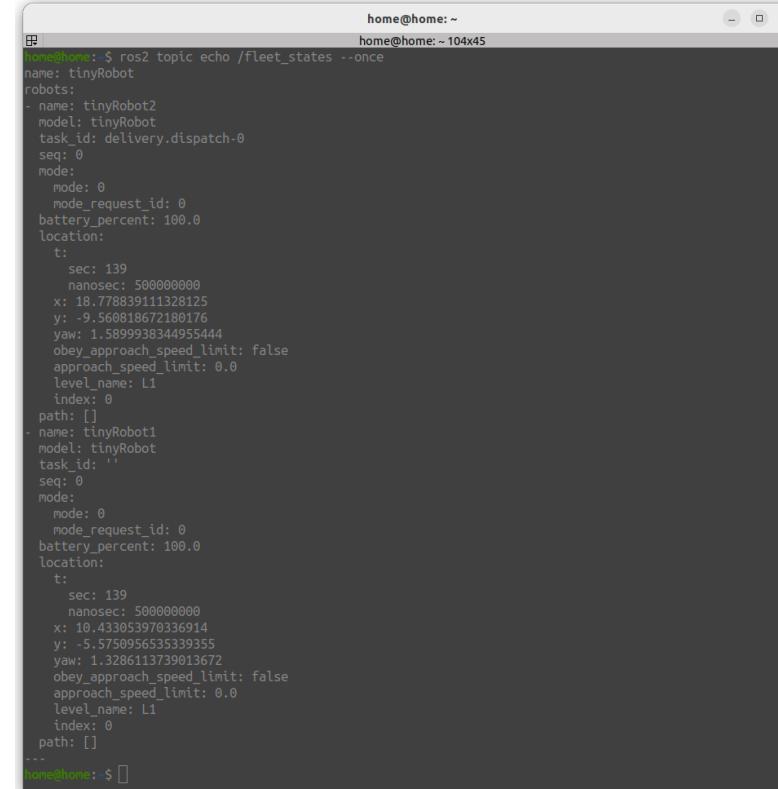
## ⦿ Delivery Task 실행 후 주요 Topic 확인

### | Robot: /fleet\_states

```
ros2 topic echo /fleet_states
```



```
home@home:~$ ros2 interface show rmf_fleet_msgs/msg/FleetState
string name
RobotState[] robots
    string name
    string model
    string task_id
    uint64 seq
    RobotMode mode
        uint32 mode
        uint32 MODE_IDLE=0
        uint32 MODE_CHARGING=1
        uint32 MODE_MOVING=2
        uint32 MODE_PAUSED=3
        uint32 MODE_WAITING=4
        uint32 MODE_EMERGENCY=5
        uint32 MODE_GOING_HOME=6
        uint32 MODE_DOCKING=7
        uint32 MODE_ADAPTER_ERROR=8
        uint32 MODE_CLEANING=9
        uint64 mode_request_id
    float32 battery_percent
    Location location
        builtin_interfaces/Time t
            int32 sec
            uint32 nanosec
        float32 x
        float32 y
        float32 yaw
        bool obey_approach_speed_limit false
        float32 approach_speed_limit
        string level_name
        uint64 index
    Location[] path
        builtin_interfaces/Time t
            int32 sec
            uint32 nanosec
        float32 x
        float32 y
        float32 yaw
        bool obey_approach_speed_limit false
        float32 approach_speed_limit
        string level_name
        uint64 index
home@home:~$
```



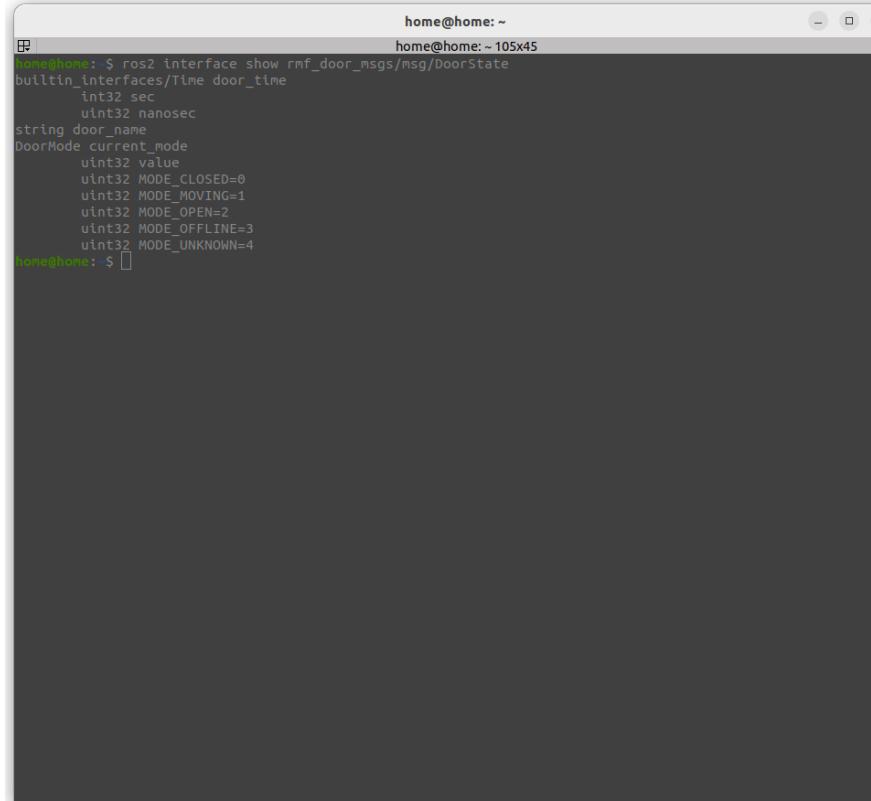
```
home@home:~$ ros2 topic echo /fleet_states --once
name: tinyRobot
robots:
- name: tinyRobot2
  model: tinyRobot
  task_id: delivery.dispatch-0
  seq: 0
  mode:
    mode: 0
    mode_request_id: 0
  battery_percent: 100.0
  location:
    t:
      sec: 139
      nanosec: 500000000
    x: 18.778839111328125
    y: -9.560818672180176
    yaw: 1.589993834495544
    obey_approach_speed_limit: false
    approach_speed_limit: 0.0
    level_name: L1
    index: 0
  path: []
- name: tinyRobot1
  model: tinyRobot
  task_id: ''
  seq: 0
  mode:
    mode: 0
    mode_request_id: 0
  battery_percent: 100.0
  location:
    t:
      sec: 139
      nanosec: 500000000
    x: 10.433053970336914
    y: -5.5750956535339355
    yaw: 1.3286113739013672
    obey_approach_speed_limit: false
    approach_speed_limit: 0.0
    level_name: L1
    index: 0
  path: []
...
home@home:~$
```

# Office World Demo

## ⦿ Delivery Task 실행 후 주요 Topic 확인

### | Door: /door\_states

```
ros2 topic echo /door_states
```



```
home@home:~$ ros2 interface show rmf_door_msgs/msg/DoorState
builtin_interfaces/Time door_time
  int32 sec
  uint32 nanosec
string door_name
DoorMode current_mode
  uint32 value
  uint32 MODE_CLOSED=0
  uint32 MODE_MOVING=1
  uint32 MODE_OPEN=2
  uint32 MODE_OFFLINE=3
  uint32 MODE_UNKNOWN=4
home@home:~$
```

```
---
```

```
door_time:
  sec: 978
  nanosec: 802000000
door_name: hardware_door
current_mode:
  value: 0
---
```

```
---
```

```
door_time:
  sec: 986
  nanosec: 802000000
door_name: hardware_door
current_mode:
  value: 1
---
```

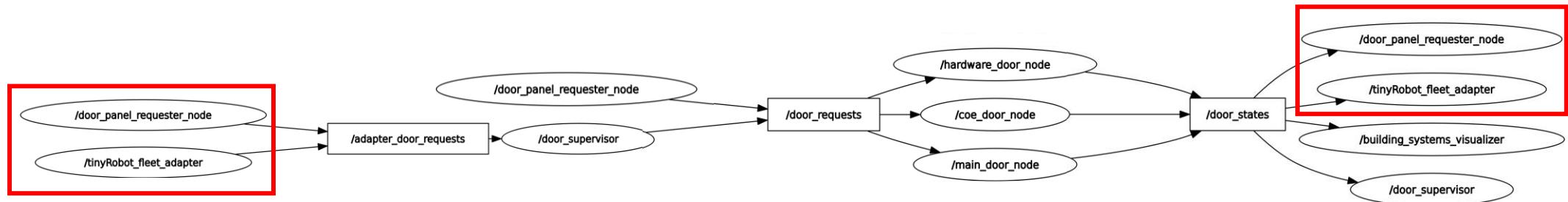
```
---
```

```
door_time:
  sec: 988
  nanosec: 802000000
door_name: hardware_door
current_mode:
  value: 2
---
```

# Office World Demo

- ▶ Delivery Task 실행 후 Door 관련 rosgraph 확인

- | Door 관련 rosgraph



# Office World Demo

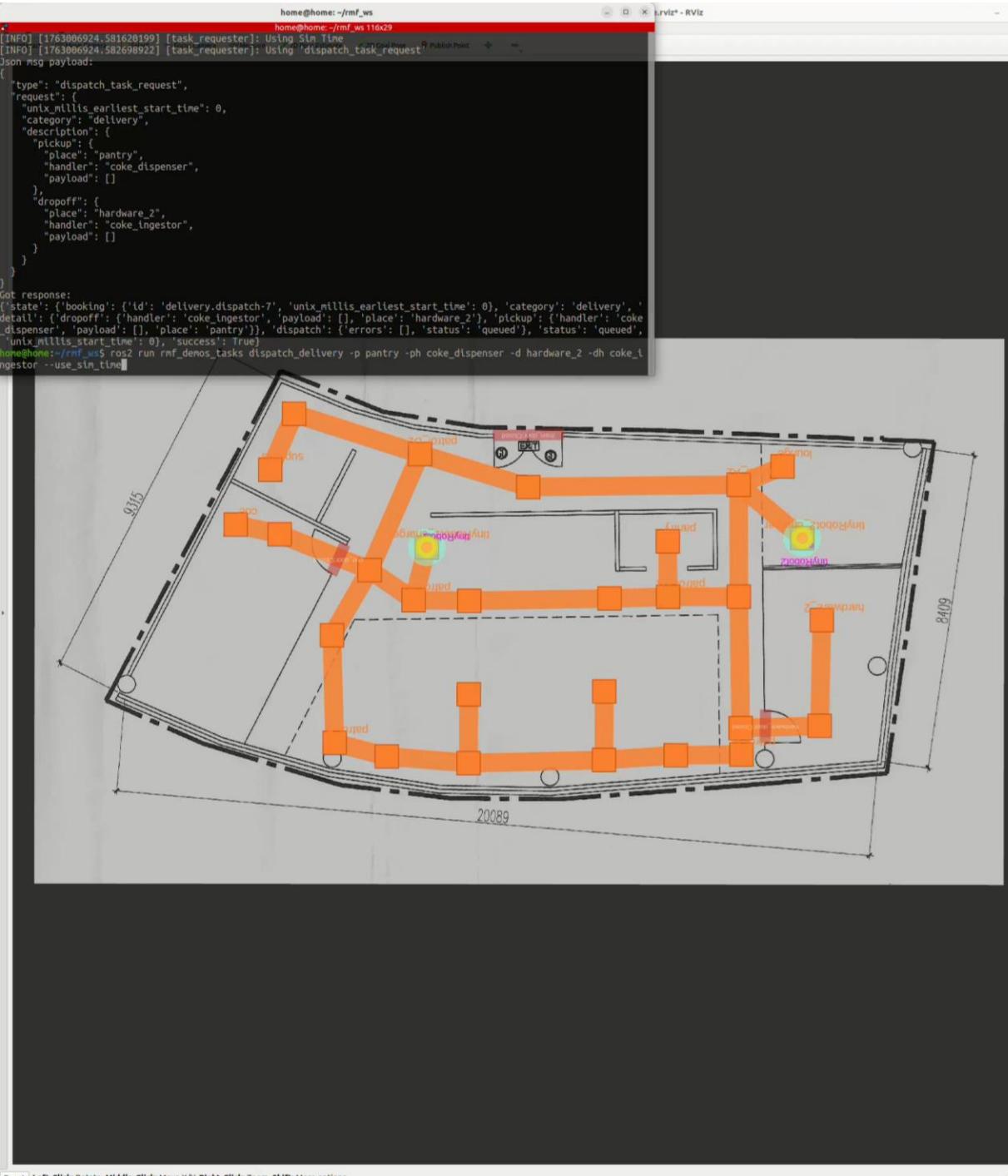
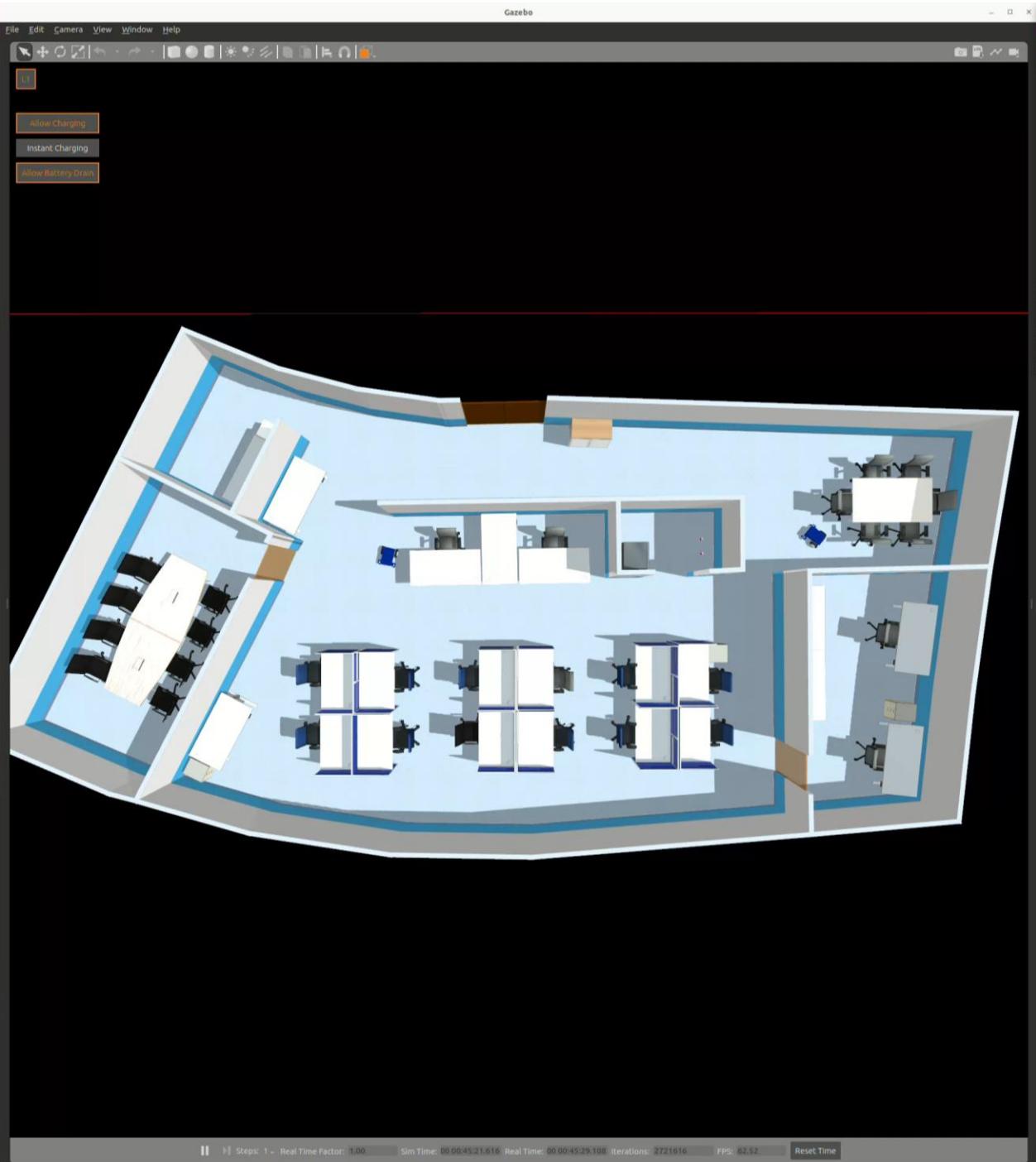
## ⦿ Delivery Task 명령으로 로봇 2대 운용

### | Delivery Task 명령 실행 / 로봇 지정 X

```
ros2 run rmf_demos_tasks dispatch_delivery -p pantry -ph coke_dispenser -d hardware_2 -dh coke_ingestor --use_sim_time
```

### | Delivery Task 명령 실행 / 로봇 지정 O (tinyRobot1)

```
ros2 run rmf_demos_tasks dispatch_delivery -p pantry -ph coke_dispenser -d hardware_2 -dh coke_ingestor -R tinyRobot1 --use_sim_time
```



# Office World Demo

## ⦿ Delivery Task 명령으로 로봇 2대 운용 log 확인

### | Traffic Negotiation

```
[gzserver-17] [INFO] [1763010311.456589757] [coke_dispenser]: Dispensing item
[gzserver-17] [INFO] [1763010311.456861253] [coke_dispenser]: Success
[fleet_adapter-16] [INFO] [1763010311.458254632] [tinyRobot_command_handle]: Received new path for tinyRobot1
[fleet_adapter-16] [INFO] [1763010311.915147300] [tinyRobot_command_handle]: Robot [tinyRobot1] has reached the destination for cmd_id 164
[gzserver-17] [WARN] [1763010312.456250615] [coke_dispenser]: Request already succeeded: [delivery.dispatch-0]
[fleet_adapter-16] [INFO] [1763010312.460333781] [tinyRobot_command_handle]: Received new path for tinyRobot2
[rmf_traffic_schedule-1] Require negotiation for: 5:4
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1] [0] Active negotiation: 5:4
[rmf_traffic_schedule-1]   Current negotiation state
[rmf_traffic_schedule-1]   .. 4:1 [5:0]
[rmf_traffic_schedule-1]   .. [5:0]
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1] [0] Active negotiation: 5:4
[rmf_traffic_schedule-1]   Current negotiation state
[rmf_traffic_schedule-1]   .. {4:1}
[rmf_traffic_schedule-1]   .. [5:0]
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1] [0] Active negotiation: 5:4
[rmf_traffic_schedule-1]   Current negotiation state
[rmf_traffic_schedule-1]   .. {4:1}
[rmf_traffic_schedule-1]   .. 5:1 [4:0]
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1] [0] Active negotiation: 5:4
[rmf_traffic_schedule-1]   Current negotiation state
[rmf_traffic_schedule-1]   .. {4:1}
[rmf_traffic_schedule-1]   .. 5:1 >4:1<
[rmf_traffic_schedule-1]
[rmf_traffic_schedule-1] [INFO] [1763010312.481019046] [rmf_traffic_schedule_primary]: Resolved negotiation [0]: 5:1 4:1
[fleet_adapter-16] [INFO] [1763010312.484005250] [tinyRobot_command_handle]: Received new path for tinyRobot2
[fleet_adapter-16] [INFO] [1763010312.486447055] [tinyRobot_command_handle]: Received new path for tinyRobot1
```

# **RMF Panel**

# RMF Panel

## ⦿ RMF Panel

→ RMF 시스템의 건물·로봇·태스크 상태를 브라우저에서 모니터링·제어할 수 있는 웹 기반 대시보드/관리 도구

## ⦿ RMF Panel으로 Delivery Task 명령 내리기

| 환경 불러오기

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

| Classic Gazebo로 office world 실행

```
ros2 launch rmf_demos_gz_classic office.launch.xml server_uri:="ws://localhost:7878"
```

| RMF Panel 접속

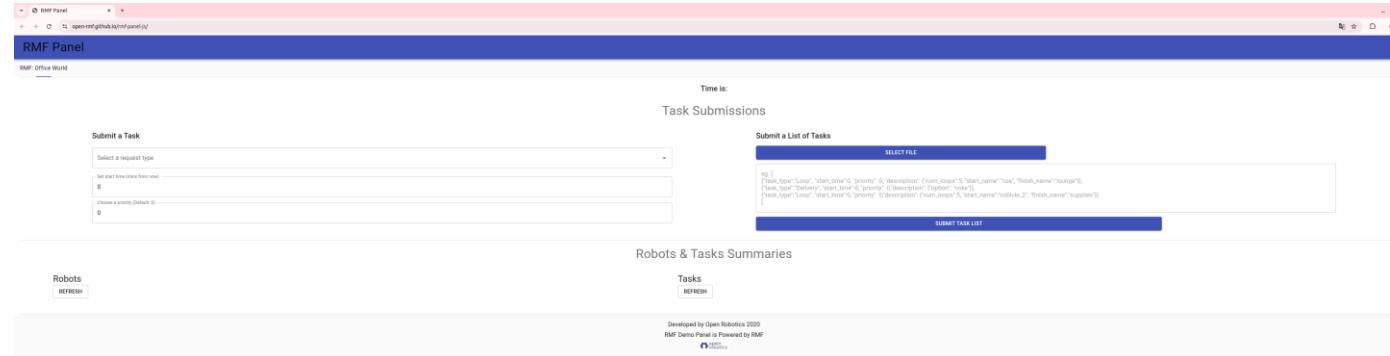
<https://open-rmf.github.io/rmf-panel-js/>

# RMF Panel

## ⦿ RMF Panel으로 Delivery Task 명령 내리기

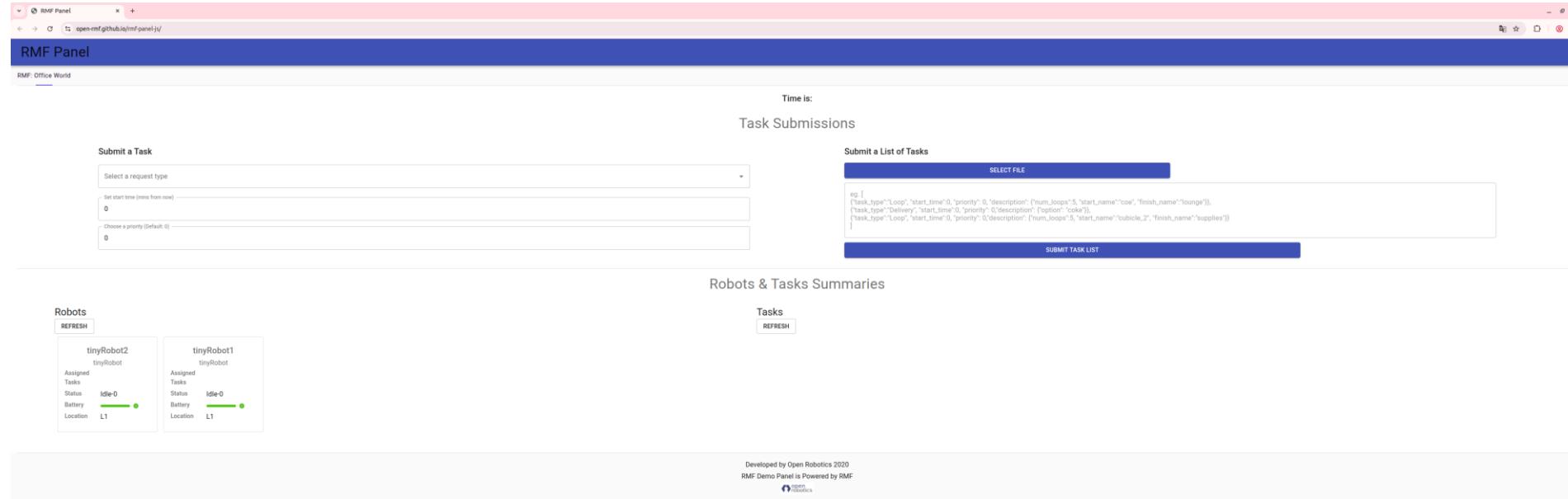
### | RMF Panel 접속

<https://open-rmf.github.io/rmf-panel-js/>



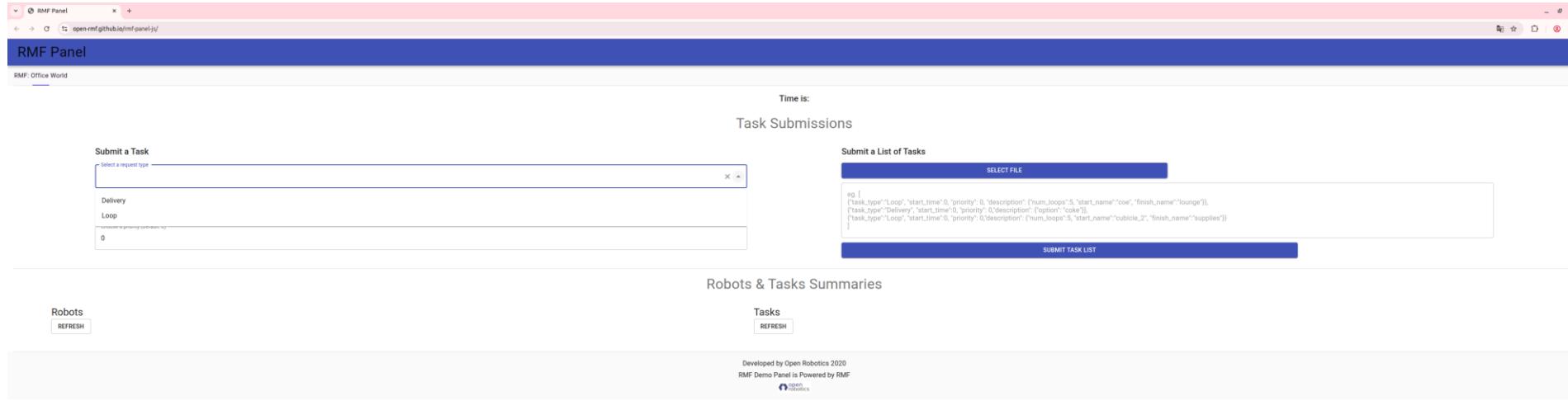
# RMF Panel

## ☞ RMF Panel으로 Delivery Task 명령 내리기



# RMF Panel

## ☞ RMF Panel으로 Delivery Task 명령 내리기



# RMF Panel

## ☞ RMF Panel으로 Delivery Task 명령 내리기

The screenshot displays the RMF Panel web application interface. At the top, there's a navigation bar with tabs like 'Home', 'Tasks', 'Logs', etc., and a search bar. Below the header, the main content area is divided into several sections:

- Task Submissions:** This section contains two forms:
  - Submit a Task:** A form for creating a single task. It includes fields for selecting a request type (Delivery), setting start time (0), choosing priority (0), and scheduling a delivery request for 'coke'.
  - Submit a List of Tasks:** A form for submitting multiple tasks via a file. It has a 'SELECT FILE' button and a 'SUBMIT TASK LIST' button. The input field contains JSON task definitions:

```
eg [ { "task_type": "Loop", "start_time": 0, "priority": 0, "description": { "num_loops": 5, "start_name": "coke", "finish_name": "lounge" } }, { "task_type": "Delivery", "start_time": 0, "priority": 0, "description": { "option": "coke" } }, { "task_type": "Loop", "start_time": 0, priority: 0, description: { "num_loops": 2, "start_name": "cubicle_2", "finish_name": "supplies" } } ]
```
- Robots & Tasks Summaries:** This section is divided into two parts: 'Robots' and 'Tasks'.
  - Robots:** Shows two robots: 'tinyRobot2' and 'tinyRobot1'. Both are listed as 'tinyRobot' and have 0 assigned tasks, 0 status, 0 battery, and are located at L1. They both have green battery bars.
  - Tasks:** Shows the same two robots with identical information: 0 assigned tasks, 0 status, 0 battery, and located at L1.
- Footer:** Includes a copyright notice 'Developed by Open Robotics 2020' and a link 'RMF Demo Panel is Powered by RMF'.

# RMF Panel

## ☞ RMF Panel으로 Delivery Task 명령 내리기

The screenshot shows the RMF Panel web interface. At the top, a message says "Request submitted successfully! Task ID: [demos\_f116aa9c-4397-4b45-#9ae-b226611dd6ab]".

**Task Submissions**

**Submit a Task**

- Select a request type: Delivery
- Set start time (from now): 0
- Choose a priority (Default: 0): 0

**Schedule a Delivery Request**

Select delivery task:

**Submit a List of Tasks**

SELECT FILE

```
eg [
  {"task_type": "Loop", "start_time": 0, "priority": 0, "description": {"num_loops": 5, "start_name": "coff", "finish_name": "lounge"}},
  {"task_type": "Delivery", "start_time": 0, "priority": 0, "description": {"option": "coke"}},
  {"task_type": "Loop", "start_time": 0, "priority": 0, "description": {"num_loops": 3, "start_name": "cubicle_2", "finish_name": "supplies"}}
]
```

**Submit Task List**

**Robots & Tasks Summaries**

**Robots**

REFRESH

<b>tinyRobot2</b> tinyRobot	<b>tinyRobot1</b> tinyRobot
Assigned Tasks	Assigned Tasks
Status	Status
Battery	Battery
Location	Location

idle-0      idle-0

REFRESH

**Tasks**

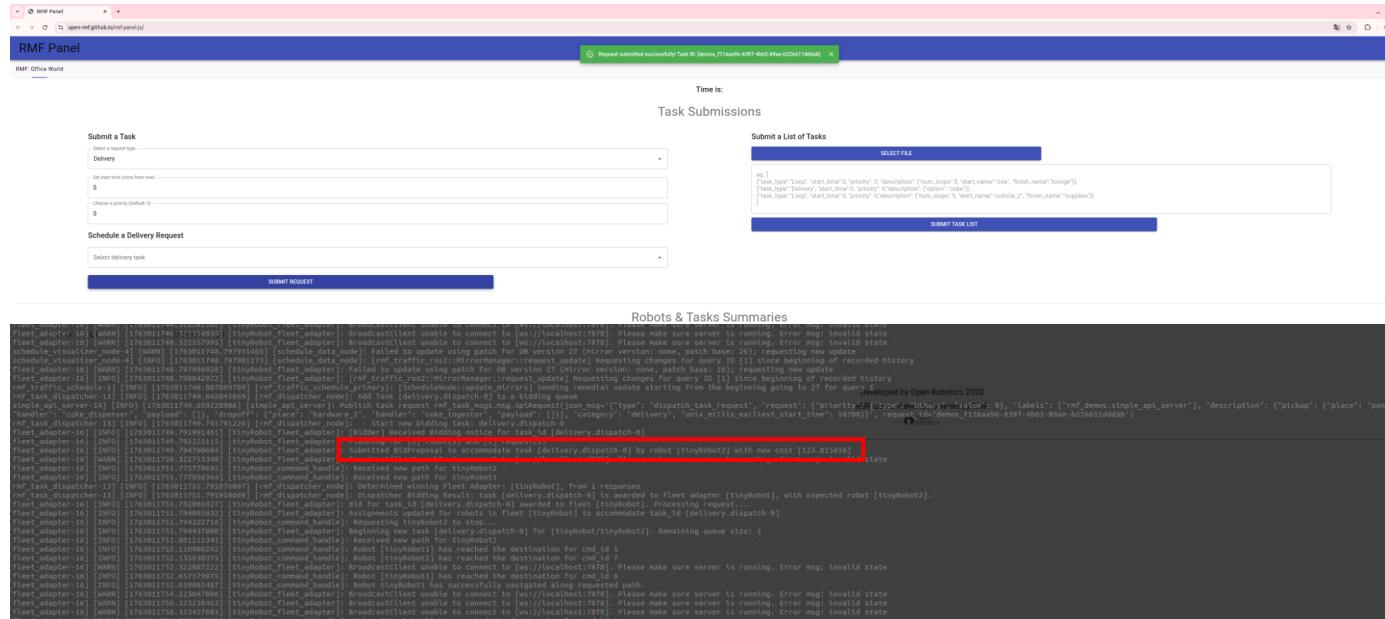
Developed by Open Robotics 2020  
RMF Demo Panel is Powered by RMF

# RMF Panel

## ⦿ RMF Panel으로 Delivery Task 명령 내리기

### | RMF Panel 접속

<https://open-rmf.github.io/rmf-panel-js/>



**감사합니다**