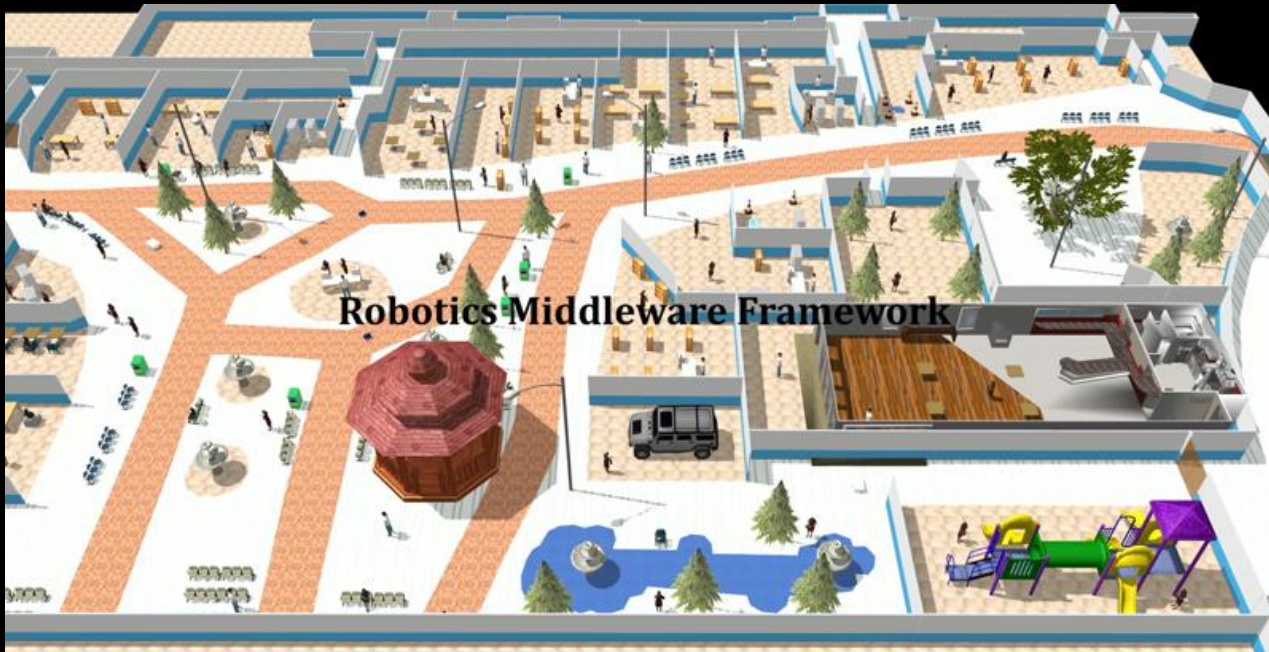




A Common Language for Robot Interoperability



Lecture 3

정은빈

Contents

01 **Hotel world Demo**

02 **RMF Panel**

Hotel world Demo

Hotel world Demo

▶ Hotel world 실행

┆ 환경 불러오기

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

┆ Classic Gazebo로 Hotel world 실행

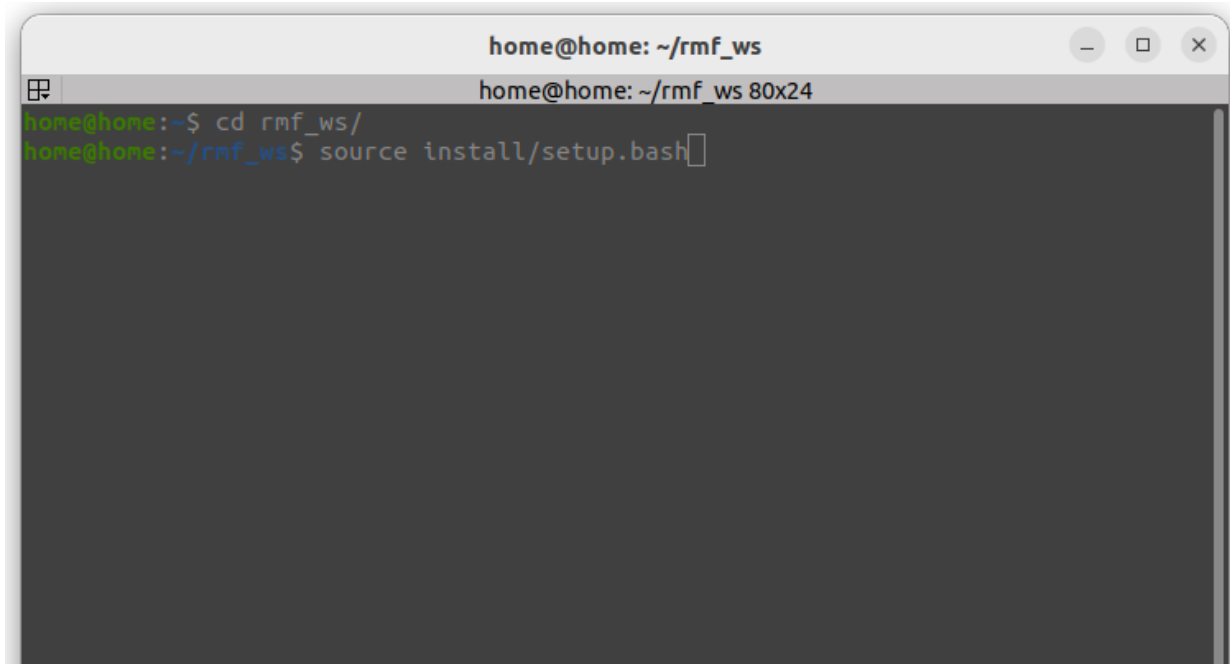
```
ros2 launch rmf_demos_gz_classic hotel.launch.xml
```

Hotel world Demo

🕒 Hotel world 실행

I 환경 불러오기

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

A terminal window titled 'home@home: ~/rmf_ws' with standard window controls. The terminal shows the command 'cd rmf_ws/' being executed, followed by 'source install/setup.bash' which is currently being typed at the prompt. The terminal background is dark gray, and the text is in a monospaced font with green and blue highlights for the prompt and command respectively.

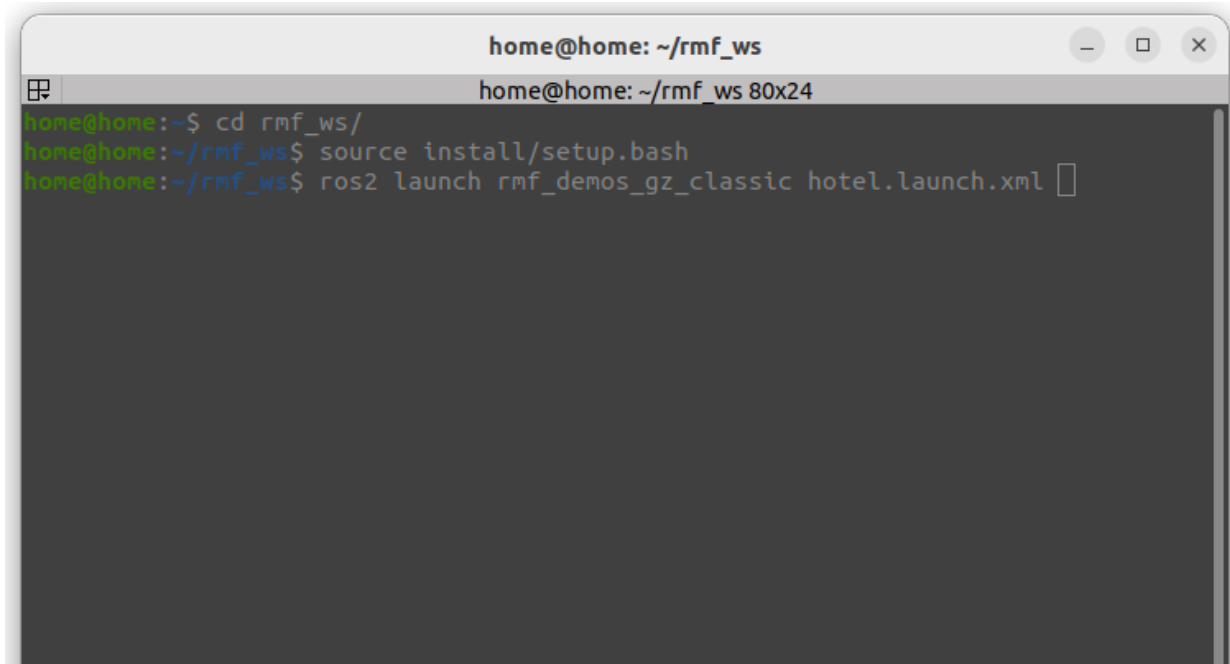
```
home@home: ~/rmf_ws
home@home: ~/rmf_ws 80x24
home@home:~$ cd rmf_ws/
home@home:~/rmf_ws$ source install/setup.bash
```

Hotel world Demo

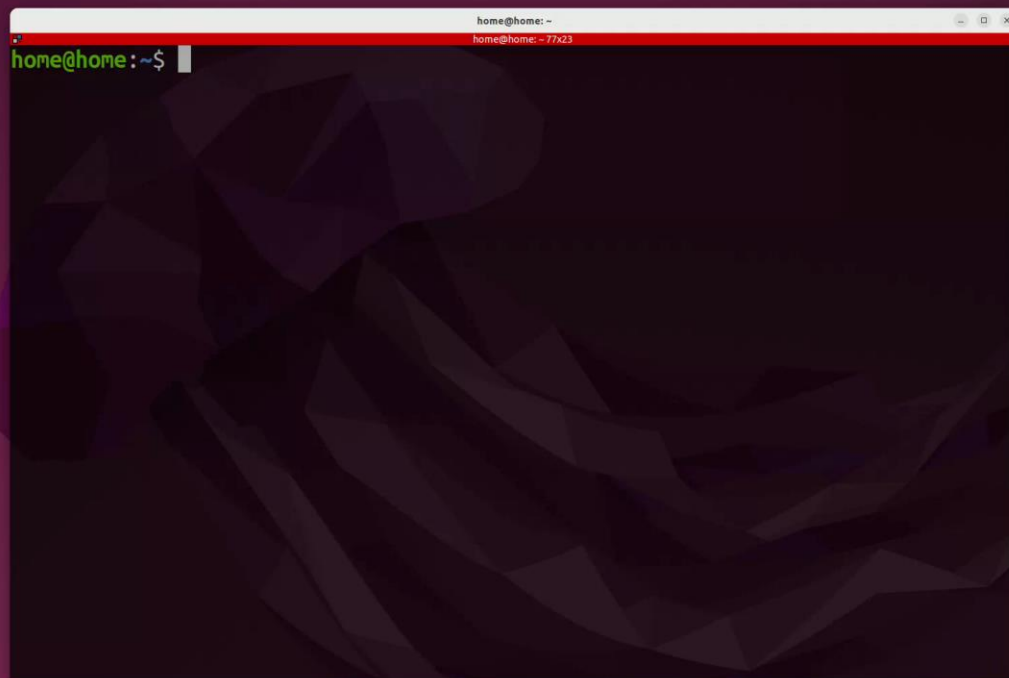
▶ Hotel world 실행

┆ Classic Gazebo로 Hotel world 실행

```
ros2 launch rmf_demos_gz_classic hotel.launch.xml
```

A terminal window titled 'home@home: ~/rmf_ws' with standard window controls. The terminal shows the following commands and prompts:
home@home:~\$ cd rmf_ws/
home@home:~/rmf_ws\$ source install/setup.bash
home@home:~/rmf_ws\$ ros2 launch rmf_demos_gz_classic hotel.launch.xml
The terminal background is dark gray, and the text is light gray. The prompt character is a green dollar sign.

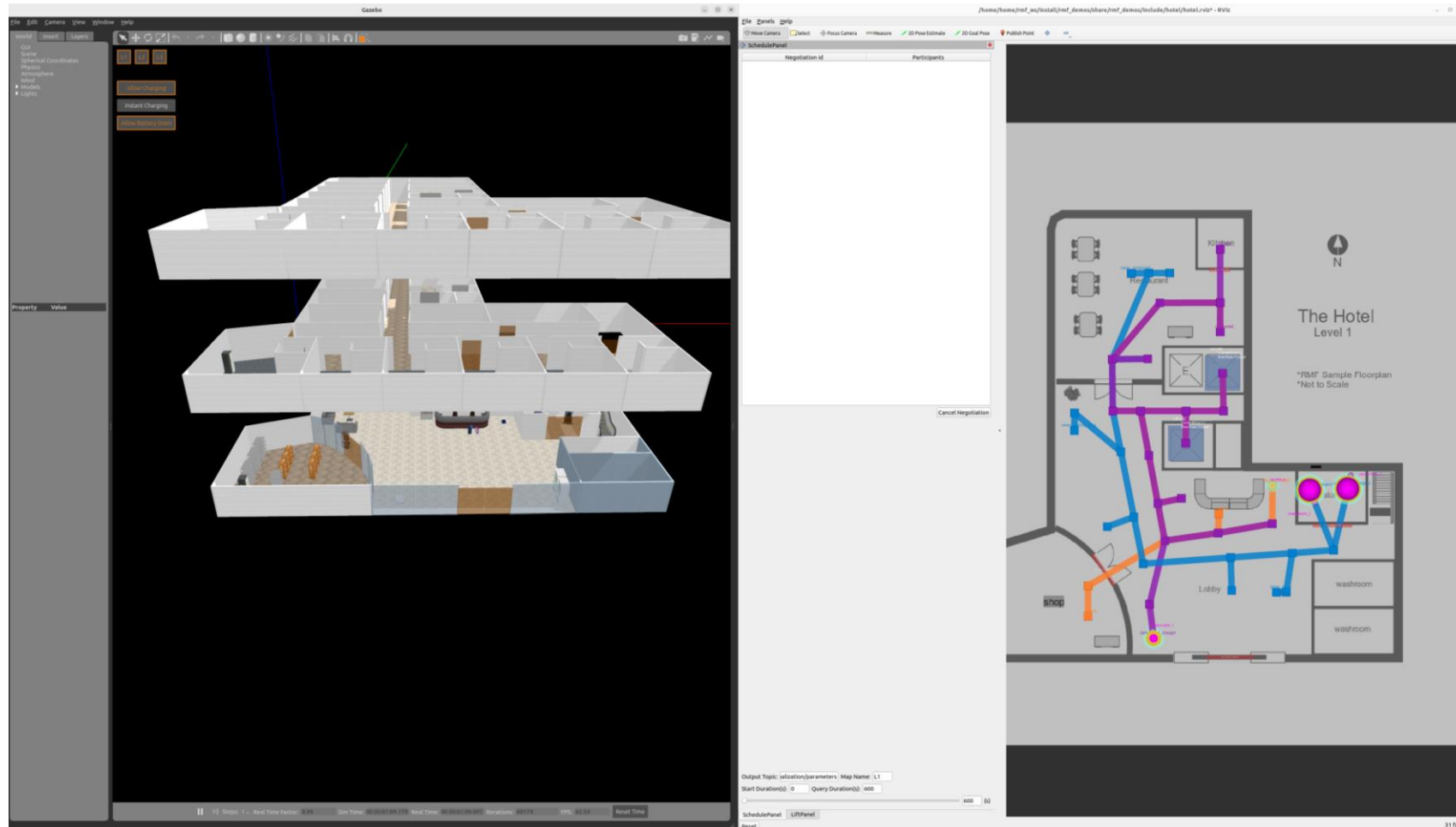
```
home@home: ~/rmf_ws  
home@home:~$ cd rmf_ws/  
home@home:~/rmf_ws$ source install/setup.bash  
home@home:~/rmf_ws$ ros2 launch rmf_demos_gz_classic hotel.launch.xml
```



Hotel world Demo

▶ Hotel world 실행

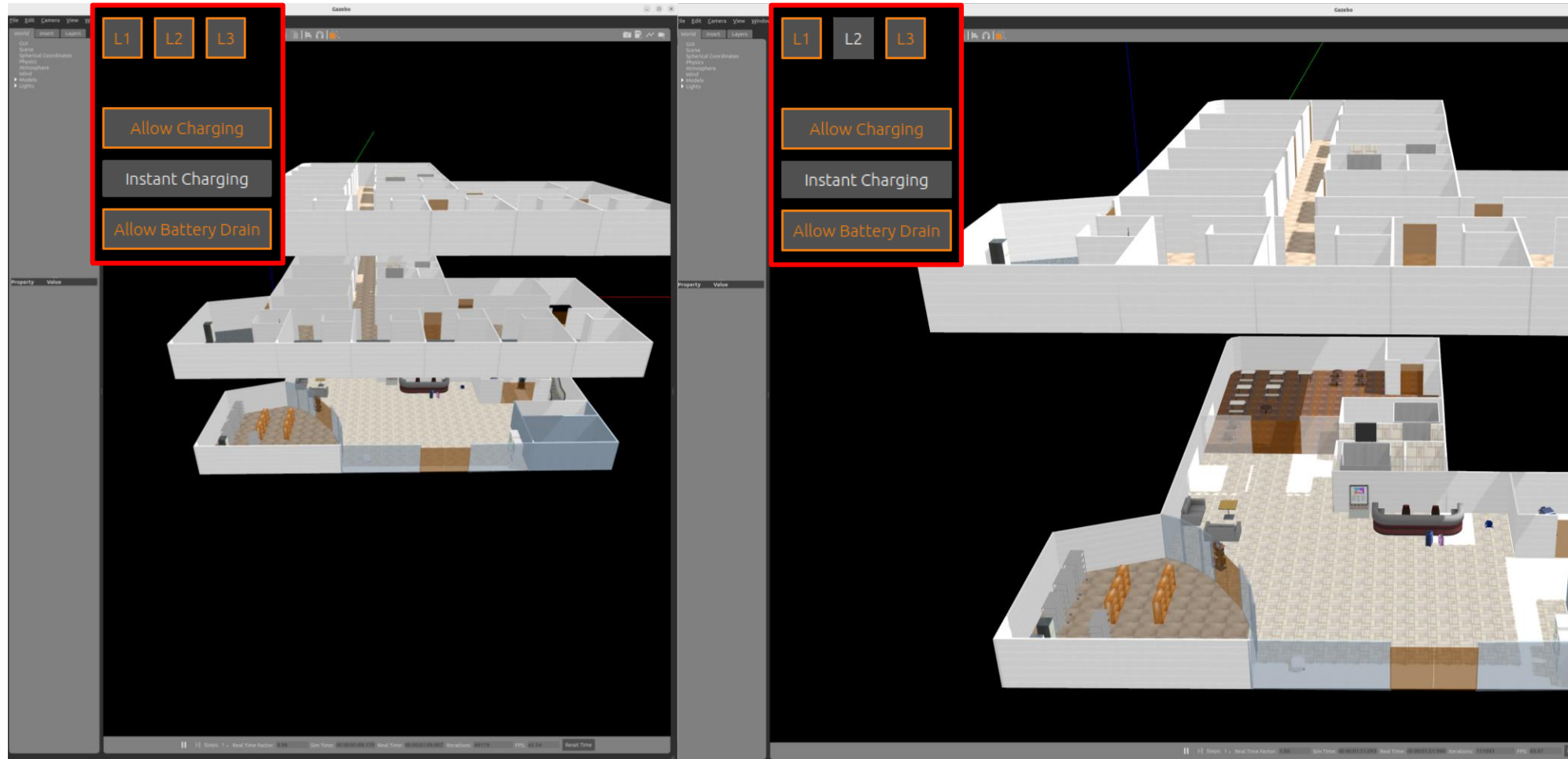
┆ Classic Gazebo로 Hotel world 실행



Hotel world Demo

▶ Hotel world 실행

┆ Classic Gazebo로 Hotel world 실행



Hotel world Demo

Hotel world 실행

Classic Gazebo로 Hotel world 실행



Hotel world Demo

Hotel world 실행

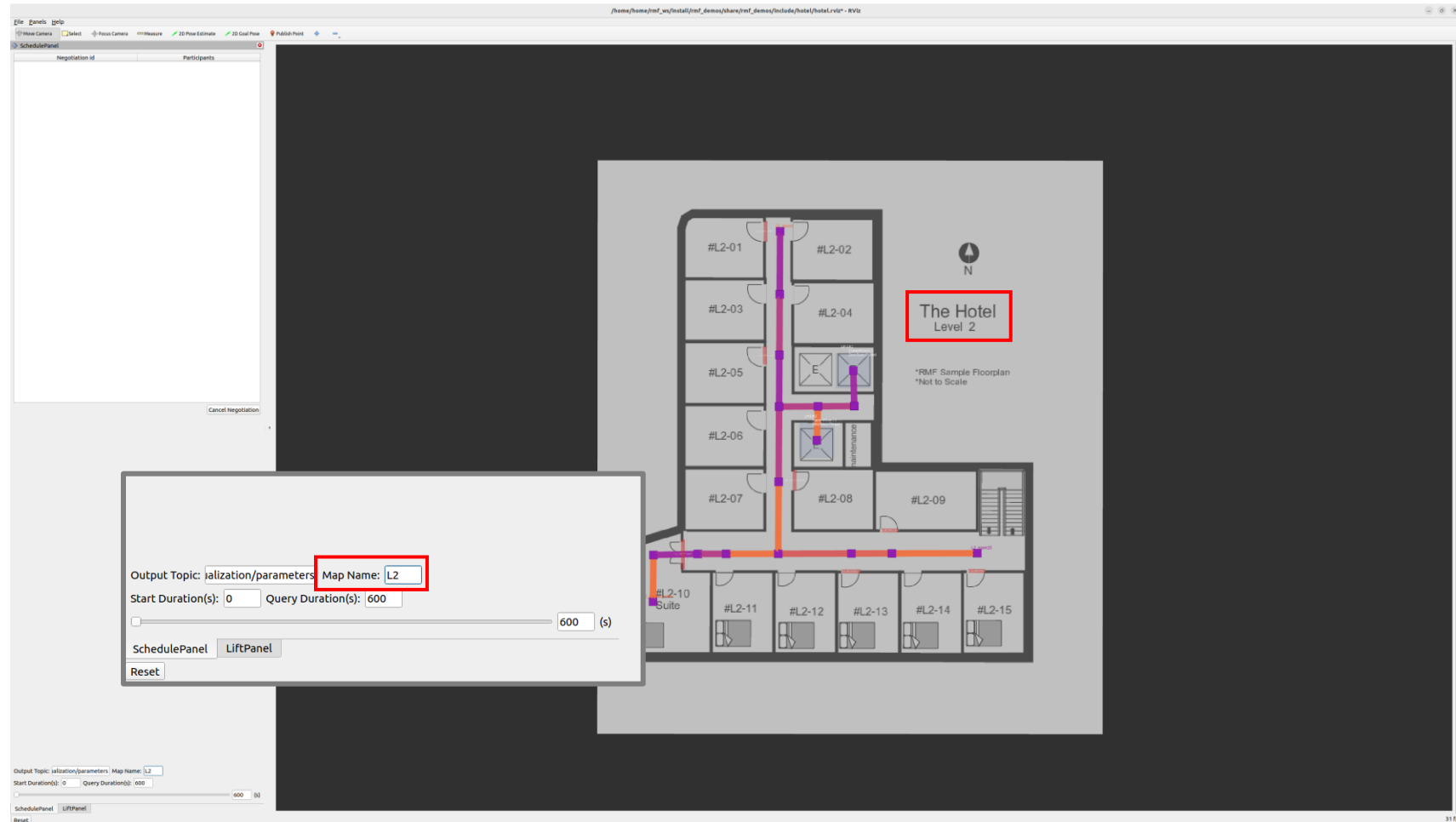
Classic Gazebo로 Hotel world 실행



Hotel world Demo

🕒 Hotel world 실행

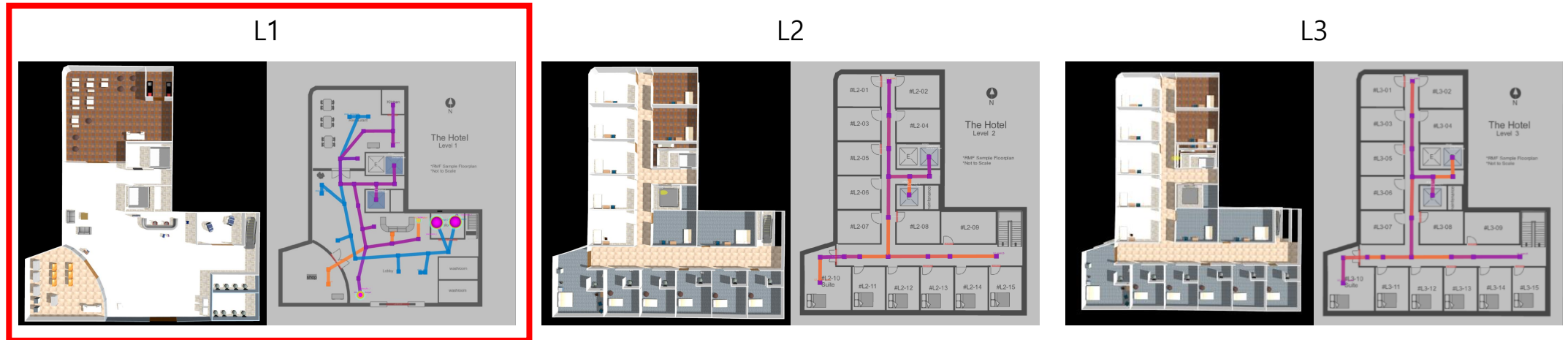
Classic Gazebo로 Hotel world 실행



Hotel world Demo

🕒 Hotel world 실행

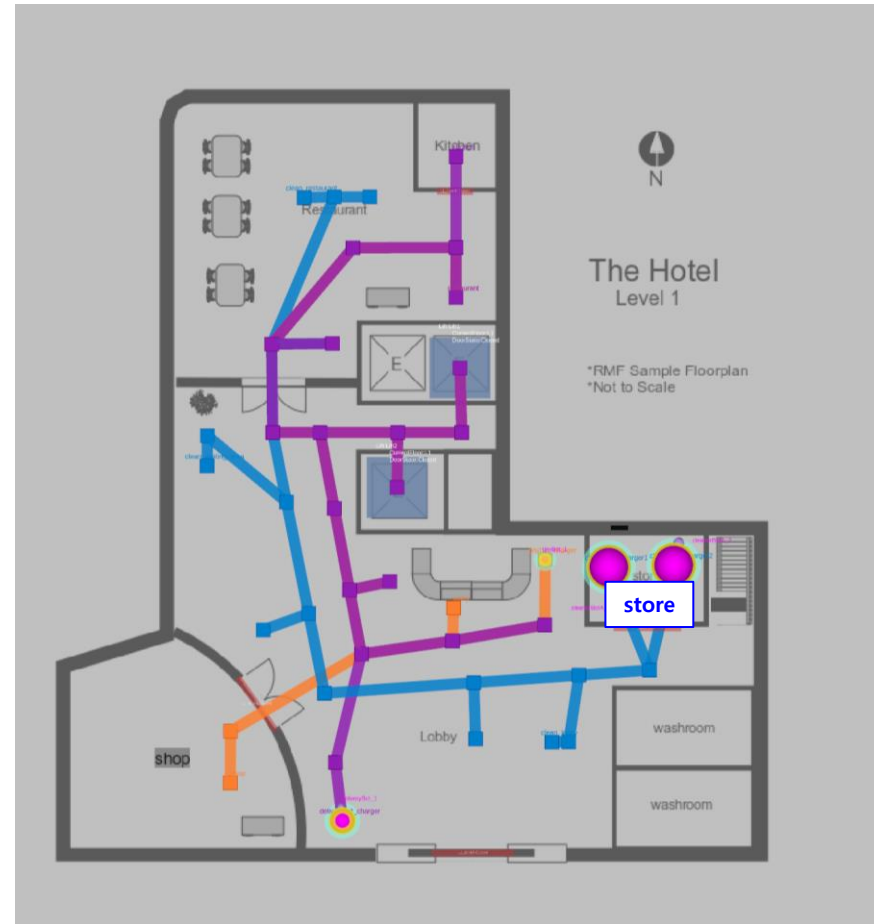
📌 Hotel world 설명



Hotel world Demo

🕒 Hotel world 실행

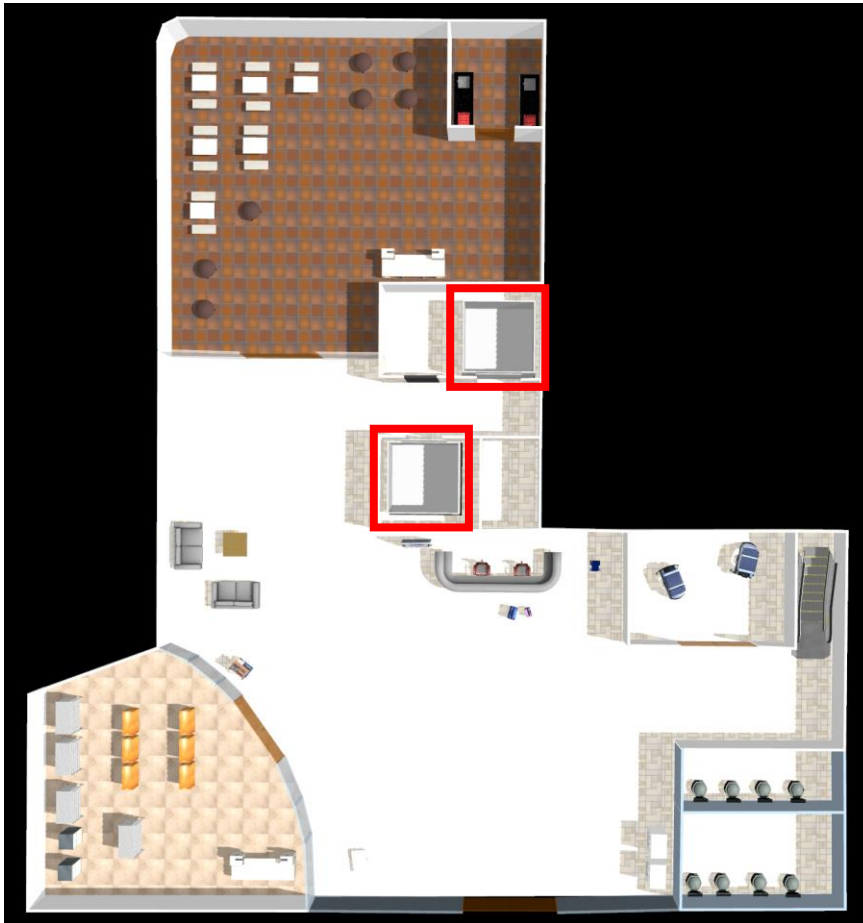
┆ Hotel world 설명 L1



Hotel world Demo

🕒 Hotel world 실행

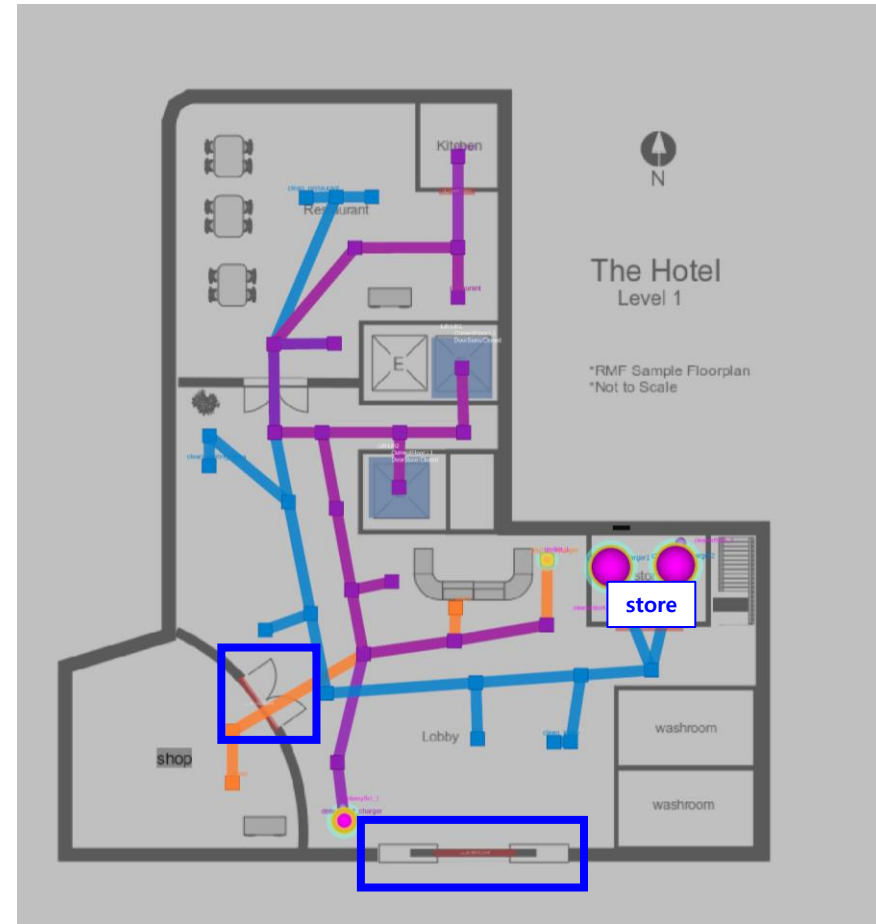
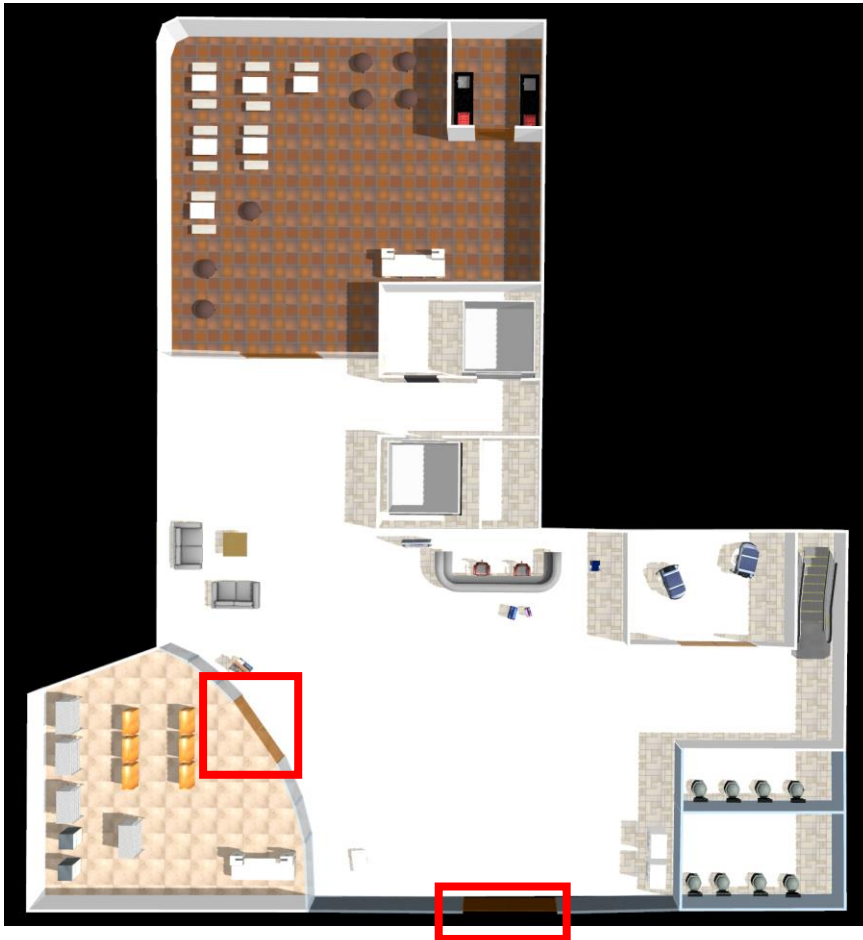
┆ Hotel world 설명 L1



Hotel world Demo

🔗 Hotel world 실행

┆ Hotel world 설명 L1



Hotel world Demo

🔗 Hotel world 실행

┆ Hotel world 설명

L1



L2



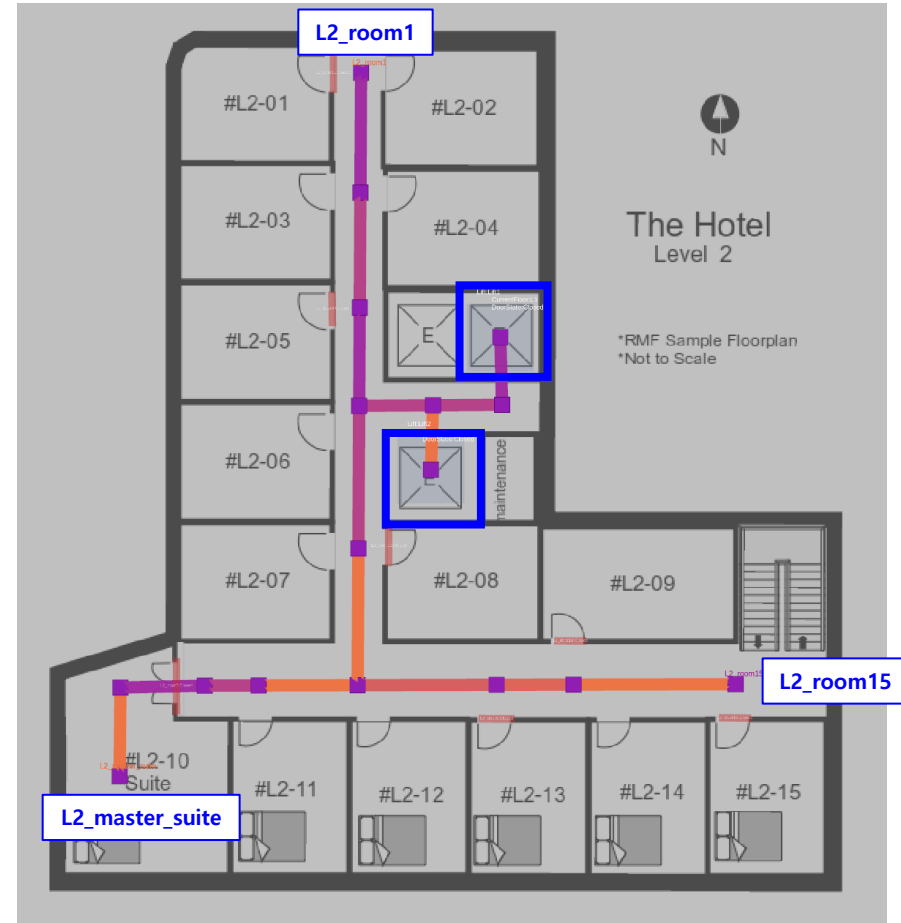
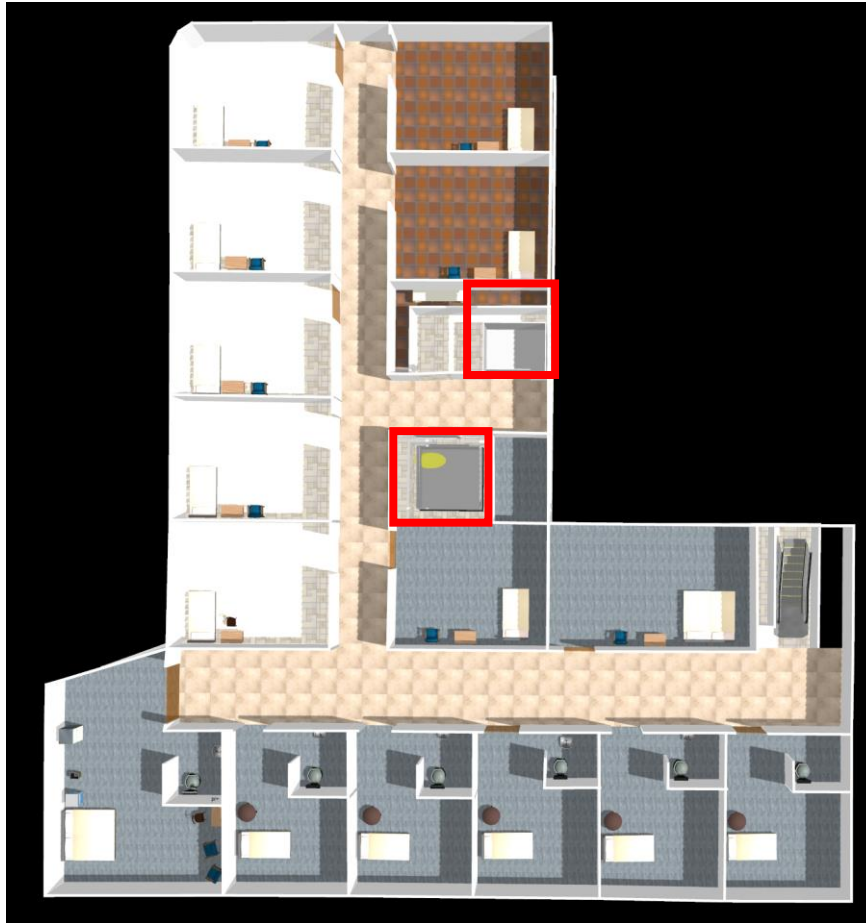
L3



Hotel world Demo

🕒 Hotel world 실행

📌 Hotel world 설명 L2, L3



Hotel world Demo

▶ Patrol Task 실행

┆ 환경 불러오기

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

┆ Patrol Task 명령

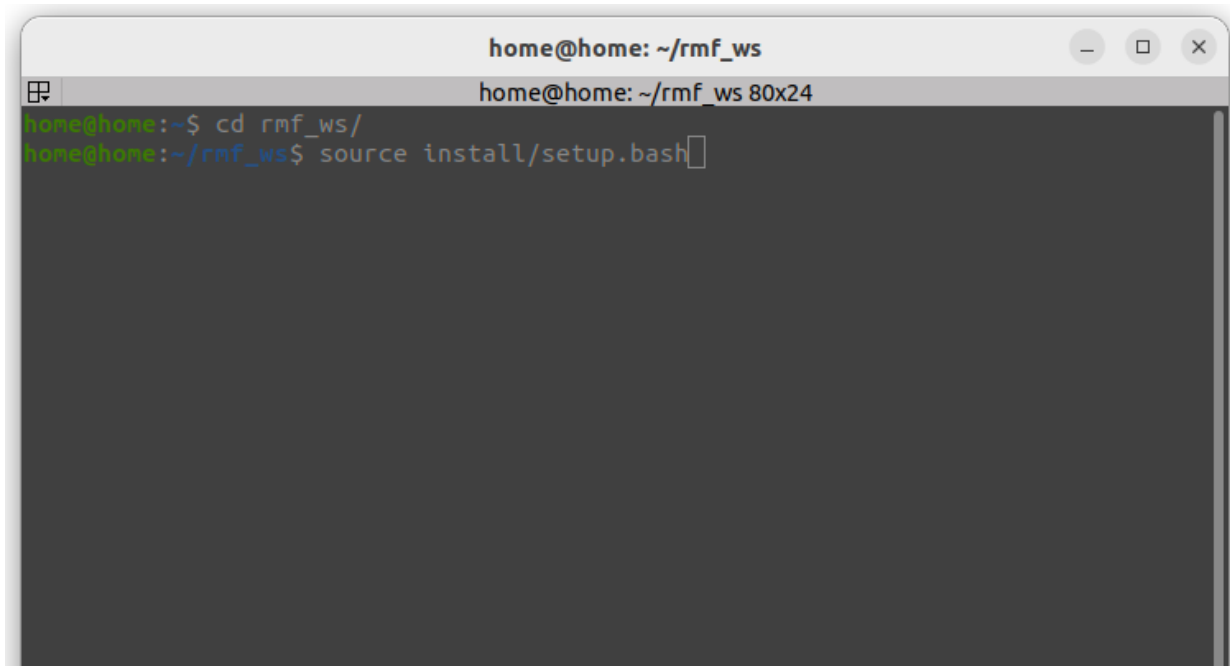
```
ros2 run rmf_demos_tasks dispatch_patrol -p L3_room1 L3_room15 -n 1 --use_sim_time
```

Hotel world Demo

🕒 Patrol Task 실행

I 환경 불러오기

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



A terminal window titled "home@home: ~/rmf_ws" with standard window controls. The terminal shows the following commands and prompts:

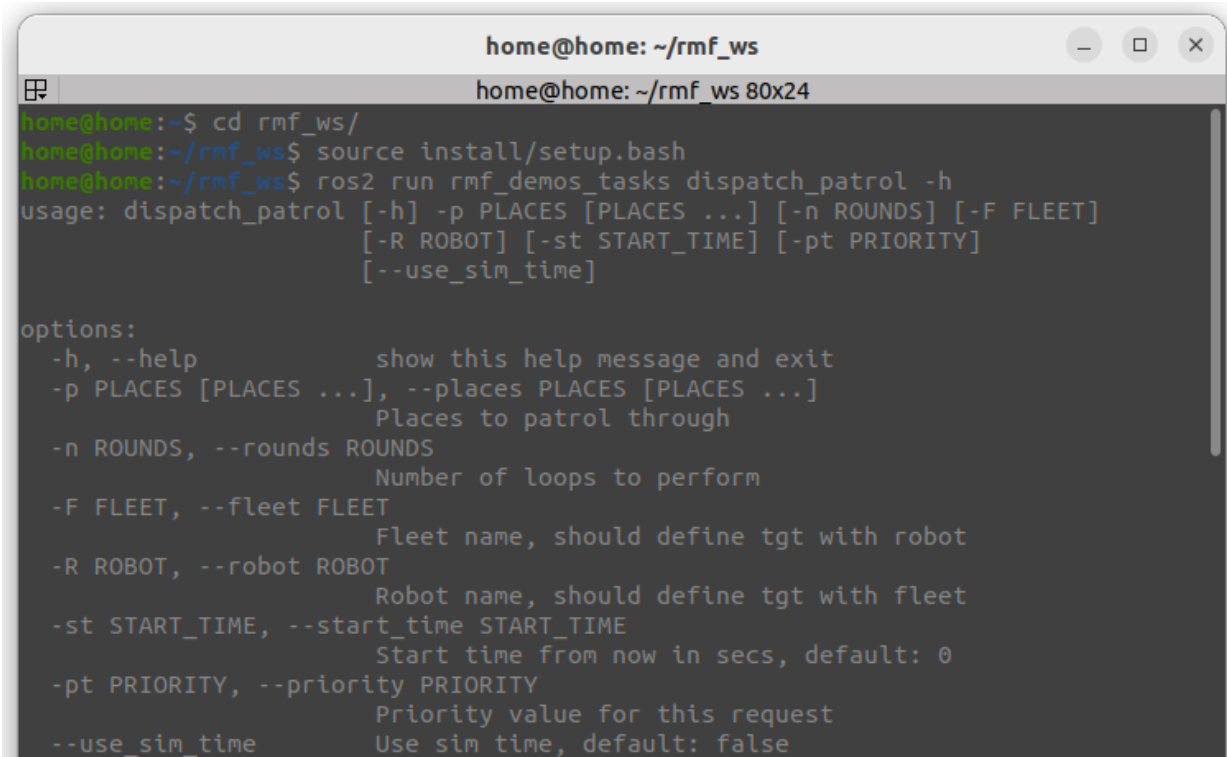
```
home@home:~$ cd rmf_ws/  
home@home:~/rmf_ws$ source install/setup.bash
```

Hotel world Demo

🕒 Patrol Task 실행

▮ Patrol Task 명령

```
ros2 run rmf_demos_tasks dispatch_patrol -p L3_room1 L3_room15 -n 1 --use_sim_time
```



```
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 run rmf_demos_tasks dispatch_patrol -h
usage: dispatch_patrol [-h] -p PLACES [PLACES ...] [-n ROUNDS] [-F FLEET]
                        [-R ROBOT] [-st START_TIME] [-pt PRIORITY]
                        [--use_sim_time]

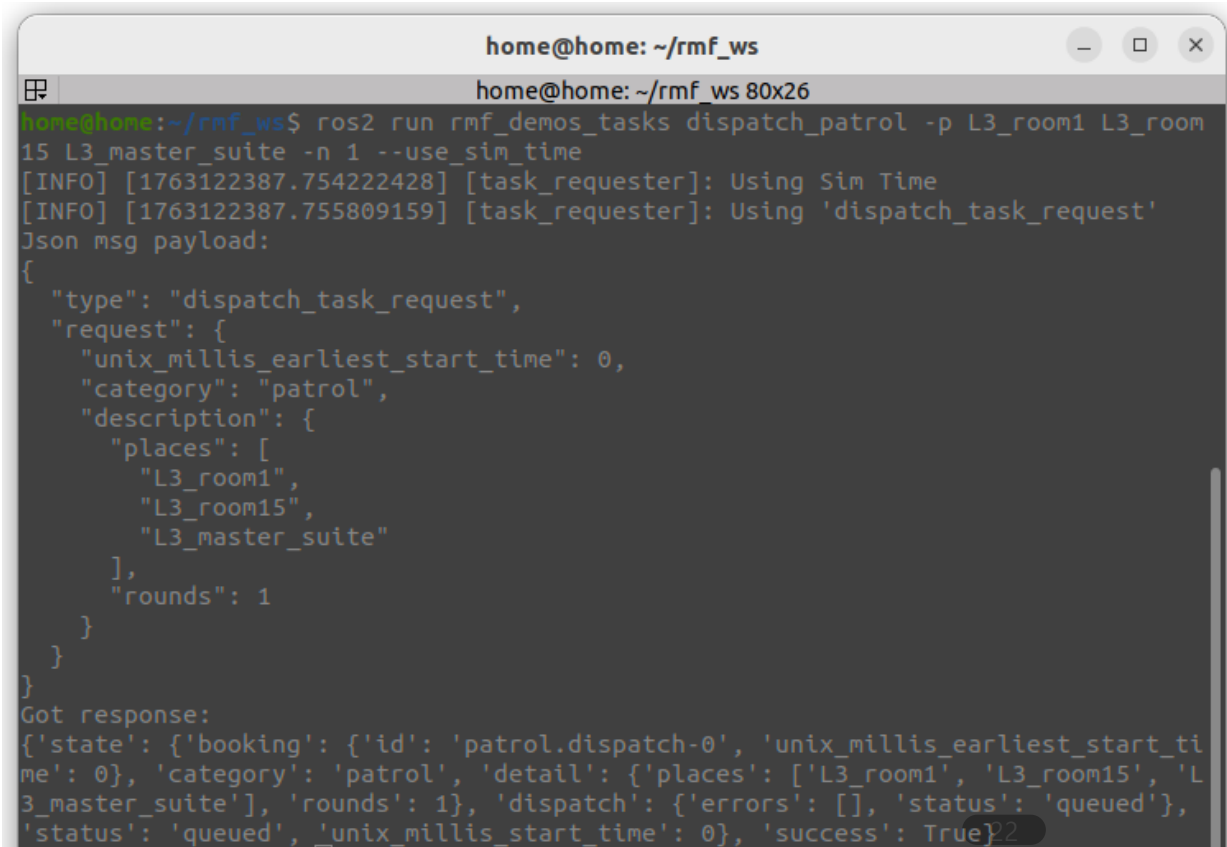
options:
  -h, --help            show this help message and exit
  -p PLACES [PLACES ...], --places PLACES [PLACES ...]
                        Places to patrol through
  -n ROUNDS, --rounds ROUNDS
                        Number of loops to perform
  -F FLEET, --fleet FLEET
                        Fleet 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
```

Hotel world Demo

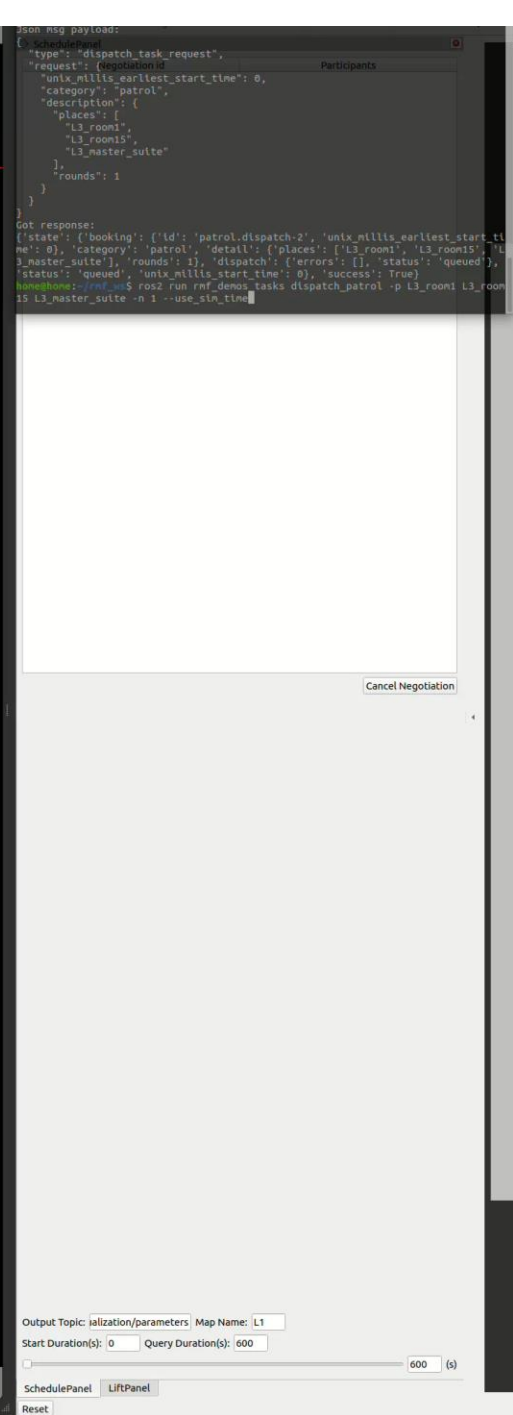
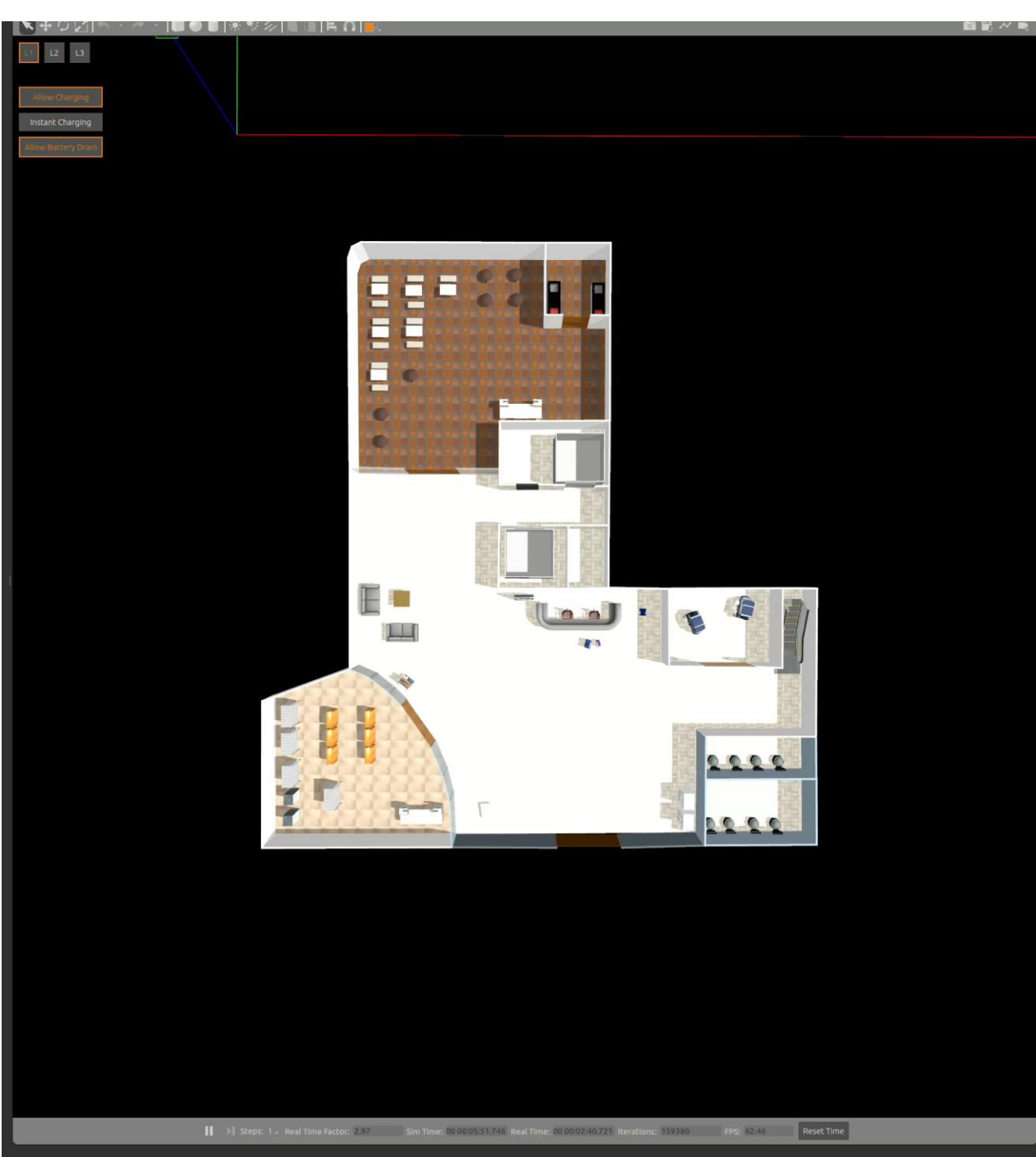
🕒 Patrol Task 실행

▮ Patrol Task 명령

```
ros2 run rmf_demos_tasks dispatch_patrol -p L3_room1 L3_room15 L3_master_suite -n 1 --use_sim_time
```



```
home@home: ~/rmf_ws
home@home: ~/rmf_ws 80x26
home@home:~/rmf_ws$ ros2 run rmf_demos_tasks dispatch_patrol -p L3_room1 L3_room
15 L3_master_suite -n 1 --use_sim_time
[INFO] [1763122387.754222428] [task_requester]: Using Sim Time
[INFO] [1763122387.755809159] [task_requester]: Using 'dispatch_task_request'
Json msg payload:
{
  "type": "dispatch_task_request",
  "request": {
    "unix_millis_earliest_start_time": 0,
    "category": "patrol",
    "description": {
      "places": [
        "L3_room1",
        "L3_room15",
        "L3_master_suite"
      ],
      "rounds": 1
    }
  }
}
Got response:
{'state': {'booking': {'id': 'patrol.dispatch-0', 'unix_millis_earliest_start_ti
me': 0}, 'category': 'patrol', 'detail': {'places': ['L3_room1', 'L3_room15', 'L
3_master_suite'], 'rounds': 1}, 'dispatch': {'errors': [], 'status': 'queued'},
'status': 'queued', 'unix_millis_start_time': 0}, 'success': True}
```



Hotel world Demo

🕒 Patrol Task 실행 log 확인

Task 할당 log

```
[fleet_adapter-18] [INFO] [1763468666.595882108] [tinyRobot_command_handle]: Robot tinyBot_1 has successfully navigated along requested path
[rmf_task_dispatcher-13] [INFO] [1763468674.651599153] [rmf_dispatcher_node]: Add Task [patrol.dispatch-0] to a bidding queue
[rmf_task_dispatcher-13] [INFO] [1763468674.784726274] [rmf_dispatcher_node]: - Start new bidding task: patrol.dispatch-0
[fleet_adapter-16] [INFO] [1763468674.784930587] [cleanerBotA_fleet_adapter]: [Bidder] Received Bidding notice for task_id [patrol.dispatch-0]
[fleet_adapter-18] [INFO] [1763468674.784996538] [tinyRobot_fleet_adapter]: [Bidder] Received Bidding notice for task_id [patrol.dispatch-0]
[fleet_adapter-18] [INFO] [1763468674.785115153] [tinyRobot_fleet_adapter]: Planning for [1] robot(s) and [1] request(s)
[fleet_adapter-18] [INFO] [1763468674.787010220] [tinyRobot_fleet_adapter]: Submitted BidProposal to accommodate task [patrol.dispatch-0] by robot [tinyBot_1] with new cost [263.236785]
[rmf_task_dispatcher-13] [INFO] [1763468676.784736010] [rmf_dispatcher_node]: Determined winning Fleet Adapter: [tinyRobot], from 2 responses
[rmf_task_dispatcher-13] [INFO] [1763468676.786152086] [rmf_dispatcher_node]: Dispatcher Bidding Result: task [patrol.dispatch-0] is awarded to fleet adapter [tinyRobot], with expected robot [tinyBot_1].
[fleet_adapter-18] [INFO] [1763468676.786331114] [tinyRobot_fleet_adapter]: Bid for task_id [patrol.dispatch-0] awarded to fleet [tinyRobot]. Processing request...
[fleet_adapter-18] [INFO] [1763468676.787489681] [tinyRobot_fleet_adapter]: Assignments updated for robots in fleet [tinyRobot] to accommodate task_id [patrol.dispatch-0]
[fleet_adapter-18] [INFO] [1763468676.787691537] [tinyRobot_fleet_adapter]: Beginning new task [patrol.dispatch-0] for [tinyRobot/tinyBot_1]. Remaining queue size: 1
[fleet_adapter-18] [INFO] [1763468676.787909142] [tinyRobot_command_handle]: Requesting tinyBot_1 to stop...
[fleet_adapter-18] [INFO] [1763468676.800169084] [tinyRobot_command_handle]: Received new path for tinyBot_1
[fleet_adapter-18] [INFO] [1763468677.115417086] [tinyRobot_command_handle]: Robot [tinyBot_1] has reached the destination for cmd_id 6
```

Lift log

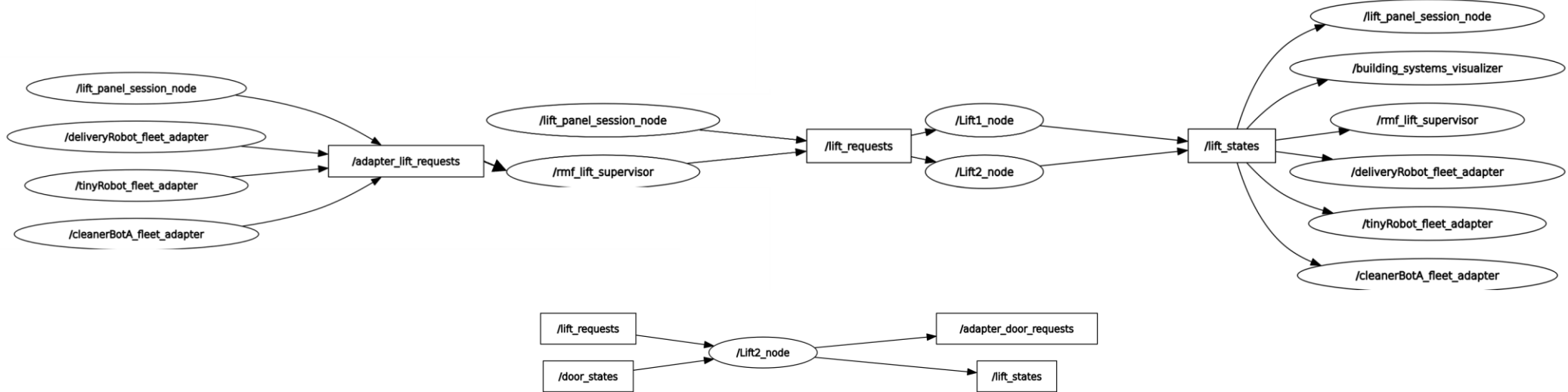
```
[gzserver-22] [INFO] [1763468920.832698976] [lift_Lift2]: Lift [Lift2] requested at level [L3]
[gzserver-22] [INFO] [1763468916.464763240] [lift_Lift2]: Reached floor L3 with doors open
[gzserver-22] [INFO] [1763468922.977644212] [lift_Lift2]: Reached floor L3 with doors closed
```


Hotel world Demo

🕒 Patrol Task 실행 후 Lift 관련 rosgraph 확인

▮ Lift 관련 rosgraph

`/_fleet_adaptor → /adapter_lift_requests → /rmf_lift_supervisor → /lift_requests → /Lift2_node → /lift_states`



Hotel world Demo

🕒 Patrol Task 실행 후 주요 Topic 확인

▮ Lift: /lift_states

```
ros2 topic echo /lift_states
```

```
home@home: ~  
home@home: ~ 105x45  
home@home:~$ ros2 interface show rmf_lift_msgs/msg/LiftState  
# lift time records when the information in this message was generated  
builtin_interfaces/Time lift_time  
  int32 sec  
  uint32 nanosec  
  
string lift_name  
  
string[] available_floors  
string current_floor  
string destination_floor  
  
uint8 door_state  
uint8 DOOR_CLOSED=0  
uint8 DOOR_MOVING=1  
uint8 DOOR_OPEN=2  
  
uint8 motion_state  
uint8 MOTION_STOPPED=0  
uint8 MOTION_UP=1  
uint8 MOTION_DOWN=2  
uint8 MOTION_UNKNOWN=3  
  
# We can only set human or agv mode, but we can read other modes: fire, etc.  
uint8[] available_modes  
uint8 current_mode  
uint8 MODE_UNKNOWN=0  
uint8 MODE_HUMAN=1  
uint8 MODE_AGV=2  
uint8 MODE_FIRE=3  
uint8 MODE_OFFLINE=4  
uint8 MODE_EMERGENCY=5  
# we can add more "read-only" modes as we come across more of them.  
  
# this field records the session_id that has been granted control of the lift  
# until it sends a request with a request_type of REQUEST_END_SESSION  
string session_id  
home@home:~$
```

Lift2

```
---  
lift_time:  
  sec: 1630  
  nanosec: 893000000  
lift_name: Lift2  
available_floors:  
  - L1  
  - L2  
  - L3  
current_floor: L3  
destination_floor: L3  
door_state: 1  
motion_state: 0  
available_modes: []  
current_mode: 2  
session_id: deliveryRobot/deliveryBot_1  
---  
lift_time:  
  sec: 1412  
  nanosec: 673000000  
lift_name: Lift2  
available_floors:  
  - L1  
  - L2  
  - L3  
current_floor: L2  
destination_floor: L3  
door_state: 0  
motion_state: 1  
available_modes: []  
current_mode: 2  
session_id: deliveryRobot/deliveryBot_1  
---  
lift_time:  
  sec: 1654  
  nanosec: 893000000  
lift_name: Lift2  
available_floors:  
  - L1  
  - L2  
  - L3  
current_floor: L3  
destination_floor: L1  
door_state: 0  
motion_state: 2  
available_modes: []  
current_mode: 2  
session_id: deliveryRobot/deliveryBot_1
```

Lift1

```
---  
lift_time:  
  sec: 1630  
  nanosec: 913000000  
lift_name: Lift1  
available_floors:  
  - L1  
  - L2  
  - L3  
current_floor: L1  
destination_floor: L1  
door_state: 0  
motion_state: 0  
available_modes: []  
current_mode: 2  
session_id: ''  
---
```

Hotel world Demo

🕒 Delivery Task 실행 후 주요 Topic 확인

❶ Lift: /lift_states

```
ros2 topic echo /lift_states
```

```
home@home: ~  
home@home: ~ 105x45  
$ ros2 interface show rmf_lift_msgs/msg/LiftState  
# lift_time records when the information in this message was generated  
builtin_interfaces/Time lift_time  
  int32 sec  
  uint32 nanosec  
  
string lift_name  
  
string[] available_floors  
string current_floor  
string destination_floor  
  
uint8 door_state  
uint8 DOOR_CLOSED=0  
uint8 DOOR_MOVING=1  
uint8 DOOR_OPEN=2  
  
uint8 motion_state  
uint8 MOTION_STOPPED=0  
uint8 MOTION_UP=1  
uint8 MOTION_DOWN=2  
uint8 MOTION_UNKNOWN=3  
  
# We can only set human or agv mode, but we can read other modes: fire, etc.  
uint8[] available_modes  
uint8 current_mode  
uint8 MODE_UNKNOWN=0  
uint8 MODE_HUMAN=1  
uint8 MODE_AGV=2  
uint8 MODE_FIRE=3  
uint8 MODE_OFFLINE=4  
uint8 MODE_EMERGENCY=5  
# we can add more "read-only" modes as we come across more of them.  
  
# this field records the session_id that has been granted control of the lift  
# until it sends a request with a request type of REQUEST_END_SESSION  
string session_id
```

Lift2

```
---  
lift_time:  
  sec: 1630  
  nanosec: 893000000  
lift_name: Lift2  
available_floors:  
- L1  
- L2  
- L3  
current_floor: L3  
destination_floor: L3  
door_state: 1  
motion_state: 0  
available_modes: []  
current_mode: 2  
session_id: deliveryRobot/deliveryBot_1  
---  
lift_time:  
  sec: 1412  
  nanosec: 673000000  
lift_name: Lift2  
available_floors:  
- L1  
- L2  
- L3  
current_floor: L2  
destination_floor: L3  
door_state: 0  
motion_state: 1  
available_modes: []  
current_mode: 2  
session_id: deliveryRobot/deliveryBot_1  
---  
lift_time:  
  sec: 1654  
  nanosec: 893000000  
lift_name: Lift2  
available_floors:  
- L1  
- L2  
- L3  
current_floor: L3  
destination_floor: L1  
door_state: 0  
motion_state: 2  
available_modes: []  
current_mode: 2  
session_id: deliveryRobot/deliveryBot_1
```

Lift1

```
---  
lift_time:  
  sec: 1630  
  nanosec: 913000000  
lift_name: Lift1  
available_floors:  
- L1  
- L2  
- L3  
current_floor: L1  
destination_floor: L1  
door_state: 0  
motion_state: 0  
available_modes: []  
current_mode: 0  
session_id: ''
```

RMF Panel

RMF Panel

🔗 RMF Panel으로 Patrol Task 명령 내리기

┆ 환경 불러오기

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

┆ Classic Gazebo로 office world 실행

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

┆ RMF Panel 접속

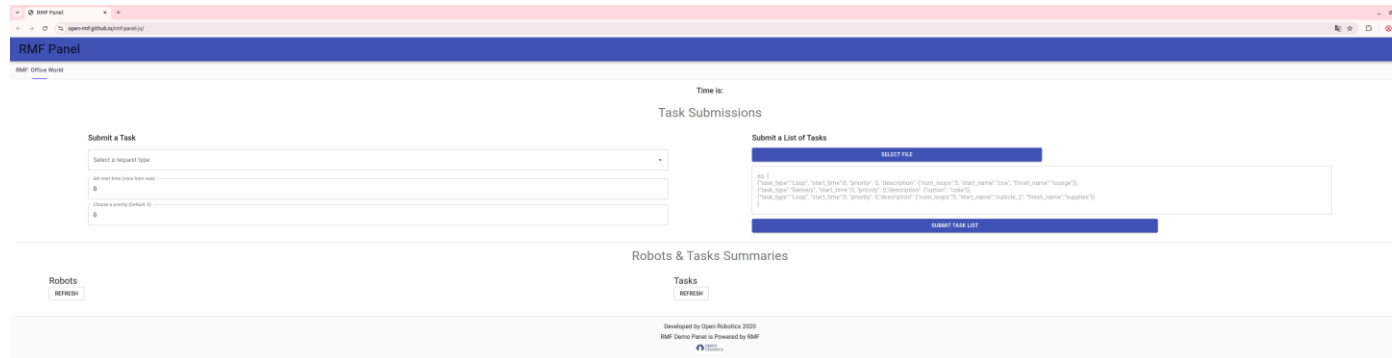
<https://open-rmf.github.io/rmf-panel-js/>

RMF Panel

🕒 RMF Panel으로 Loop Task 명령 내리기

RMF Panel 접속

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



RMF Panel

🔗 RMF Panel으로 Loop Task 명령 내리기

The screenshot displays the RMF Panel web interface in a browser window. The interface is divided into several sections:

- Header:** "RMF Panel" title bar.
- Time is:** A section for "Task Submissions" with a "Time is:" label.
- Submit a Task:** A form with a "Select a request type" dropdown, a "Set start time (now from now)" input field, and a "Choose a priority (Default: 0)" input field.
- Submit a List of Tasks:** A section with a "SELECT FILE" button and a "SUBMIT TASK LIST" button. Below the buttons is a code editor showing a JSON task list example.
- Robots & Tasks Summaries:** A section with a "Robots" tab and a "Tasks" tab. The "Robots" tab shows a "REFRESH" button and four robot status cards: "deliveryBot_1", "tinyBot_1", "cleanerBotA_2", and "cleanerBotA_1". Each card displays "Assigned Tasks", "Status", "Battery", and "Location".

At the bottom of the interface, there is a footer section with the text: "Developed by Open Robotics 2020", "RMF Demo Panel is Powered by RMF", and the Open Robotics logo.

RMF Panel

🔗 RMF Panel으로 Loop Task 명령 내리기

The screenshot displays the RMF Panel web interface in a browser window. The page has a blue header with the text "RMF Panel" and a sub-header "RMF: Hotel World". Below the header, there is a section titled "Task Submissions" which includes a "Time is:" label. Under this section, there are two main areas: "Submit a Task" and "Submit a List of Tasks". The "Submit a Task" area features a dropdown menu for "Select a request type" with options "Loop" and "Clean", and a text input field with the value "0". The "Submit a List of Tasks" area has a "SELECT FILE" button, a text area containing a JSON array of task objects, and a "SUBMIT TASK LIST" button. Below the "Task Submissions" section is a "Robots & Tasks Summaries" section with two sub-sections: "Robots" and "Tasks", each with a "REFRESH" button. At the bottom of the page, there is a footer that reads "Developed by Open Robotics 2020" and "RMF Demo Panel is Powered by RMF" with the Open Robotics logo.

RMF Panel

RMF: Hotel World

Time is:

Task Submissions

Submit a Task

Select a request type

Loop

Clean

0

Submit a List of Tasks

SELECT FILE

eg. [{ "task_type": "Loop", "start_time": 0, "priority": 0, "description": { "from_loop": 3, "start_name": "cow", "finish_name": "lounge" }, }, { "task_type": "Delivery", "start_time": 0, "priority": 0, "description": { "caption": "cookie" }, }, { "task_type": "Loop", "start_time": 0, "priority": 0, "description": { "from_loop": 3, "start_name": "cubicle_2", "finish_name": "supplies" } }]

SUBMIT TASK LIST

Robots & Tasks Summaries

Robots

REFRESH

Tasks

REFRESH

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

RMF Panel

🔗 RMF Panel으로 Loop Task 명령 내리기

The screenshot displays the RMF Panel web interface. At the top, there's a blue header with 'RMF Panel' and a sub-header 'RMF: Hotel World'. The main content area is divided into several sections:

- Task Submissions:** This section contains two forms. The 'Submit a Task' form has fields for 'Select a request type' (set to 'Loop'), 'Set start time (none from now)' (set to '0'), and 'Choose a priority (Default: 0)' (set to '0'). Below this is the 'Schedule a Loop Request' form with 'Select start location' (set to 'L2_room1'), 'Select end location' (set to 'L3_room15'), and 'Number of Loops' (set to '1'). A 'SUBMIT REQUEST' button is at the bottom. The 'Submit a List of Tasks' form has a 'SELECT FILE' button and a 'SUBMIT TASK LIST' button. A code editor shows a JSON task list example.
- Robots & Tasks Summaries:** This section has a 'Robots' tab with a 'REFRESH' button. It displays four robot cards: 'deliveryBot_1' (deliveryRobot), 'tinyBot_1' (tinyRobot), 'cleanerBotA_2' (cleanerBotA), and 'cleanerBotA_1' (cleanerBotA). Each card shows 'Assigned Tasks', 'Status' (Idle-0), 'Battery' (full), and 'Location' (L1). There is also a 'Tasks' tab with a 'REFRESH' button.
- Footer:** A grey bar at the bottom contains the text 'Developed by Open Robotics 2020' and 'RMF Demo Panel is Powered by RMF' with the Open Robotics logo.

RMF Panel

🔗 RMF Panel으로 Loop Task 명령 내리기

The screenshot displays the RMF Panel web interface. At the top, a blue header bar contains the text "RMF Panel" and a green notification bar stating "Request submitted successfully! Task ID: [demo_a1762b5c-c2ae-47b2-88ff-ae492e7a7a74]". Below the header, the main content area is divided into two columns. The left column contains two sections: "Submit a Task" and "Schedule a Loop Request". The "Submit a Task" section has a dropdown menu for "Select a request type" set to "Loop", a text input for "Set start time (mins from now)" with the value "0", and a text input for "Choose a priority (Default: 0)" with the value "0". The "Schedule a Loop Request" section has a dropdown menu for "Select start location", a dropdown menu for "Select end location", and a text input for "Number of Loops" with the value "1". A blue "SUBMIT REQUEST" button is at the bottom of this section. The right column contains a section titled "Task Submissions" with a "Submit a List of Tasks" button, a "SELECT FILE" button, a text area for a JSON task list, and a "SUBMIT TASK LIST" button. Below these sections, a "Robots & Tasks Summaries" section is visible. It includes a "Robots" subsection with a "REFRESH" button and four robot status cards: "deliveryBot_1", "tinyBot_1", "cleanerBot_2", and "cleanerBot_1". Each card shows "Assigned Tasks", "Status" (Idle-0), "Battery" (full), and "Location" (L1). There is also a "Tasks" subsection with a "REFRESH" button. At the bottom of the interface, a footer bar contains the text "Developed by Open Robotics 2020" and "RMF Demo Panel is Powered by RMF" with the Open Robotics logo.

RMF Panel

Request submitted successfully! Task ID: [demo_a1762b5c-c2ae-47b2-88ff-ae492e7a7a74]

Time is:

Task Submissions

Submit a Task

Select a request type

Loop

Set start time (mins from now)

0

Choose a priority (Default: 0)

0

Schedule a Loop Request

Select start location

Select end location

Number of Loops

1

SUBMIT REQUEST

Submit a List of Tasks

SELECT FILE

eg [

```
{ "task_type": "Loop", "start_time": 0, "priority": 0, "description": { "num_loops": 5, "start_name": "cow", "finish_name": "bongu" },
```

```
{ "task_type": "Delivery", "start_time": 0, "priority": 0, "description": { "option": "coke" },
```

```
{ "task_type": "Loop", "start_time": 0, "priority": 0, "description": { "num_loops": 5, "start_name": "cubicle_2", "finish_name": "supplies" } }
```

SUBMIT TASK LIST

Robots & Tasks Summaries

Robots

REFRESH

deliveryBot_1
deliveryRobot

Assigned Tasks

Status Idle-0

Battery

Location L1

tinyBot_1
tinyRobot

Assigned Tasks

Status Idle-0

Battery

Location L1

cleanerBot_2
cleanerBotA

Assigned Tasks

Status Idle-0

Battery

Location L1

cleanerBot_1
cleanerBotA

Assigned Tasks

Status Idle-0

Battery

Location L1

Tasks

REFRESH

Developed by Open Robotics 2020

RMF Demo Panel is Powered by RMF

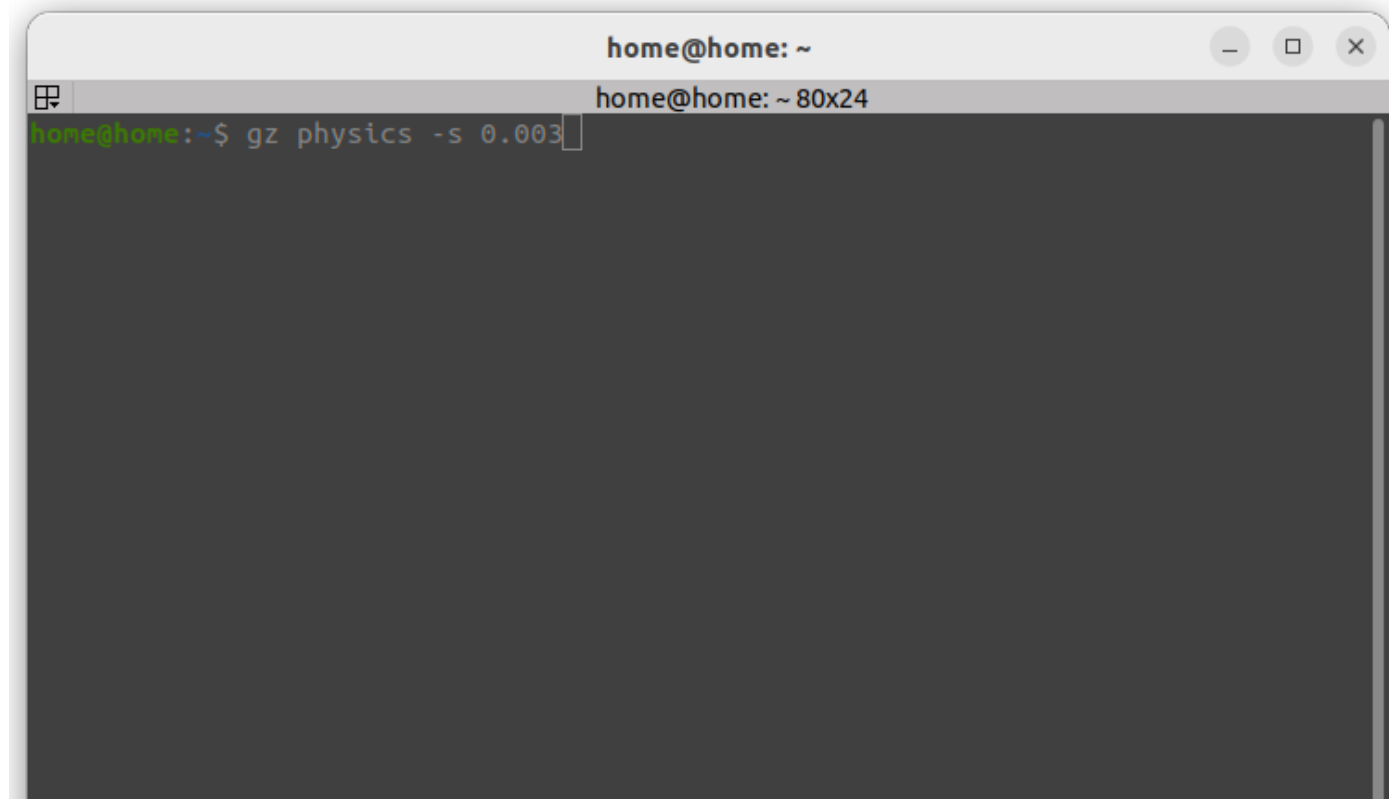
Open Robotics

RMF Panel

▶ RMF Panel으로 Loop Task 명령 내리기

┆ Robot 속도 조절

```
gz physics -s 0.003 (← 조절)
```



Time is:

Task Submissions

Submit a Task

Select a request type

Loop

Set start time (mins from now)

0

Choose a priority (Default: 0)

0

Schedule a Loop Request

Select start location

L2_room1

Select end location

L3_room1

restaurant

kitchen

shop

deliverybot_charger

tinybot_charger

cleanerbot_charger1

cleanerbot_charger2

L2_room1

L2_room15

L2_master_suite

L3_room1

L3_room15

L3_master_suite

Robot 1

Battery Location

L1

Robot 2

Battery Location

L1

Robot 3

Battery Location

L1

Robot 4

Battery Location

L1

Robots

REFRESH

Tasks

REFRESH

Submit a List of Tasks

SELECT FILE

eg. [{"task_type": "Loop", "start_time": 0, "priority": 0, "description": {"num_loops": 5, "start_name": "coe", "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": 5, "start_name": "cubicle_2", "finish_name": "supplies"}}]

SUBMIT TASK LIST

Tasks & Tasks Summaries

감사합니다