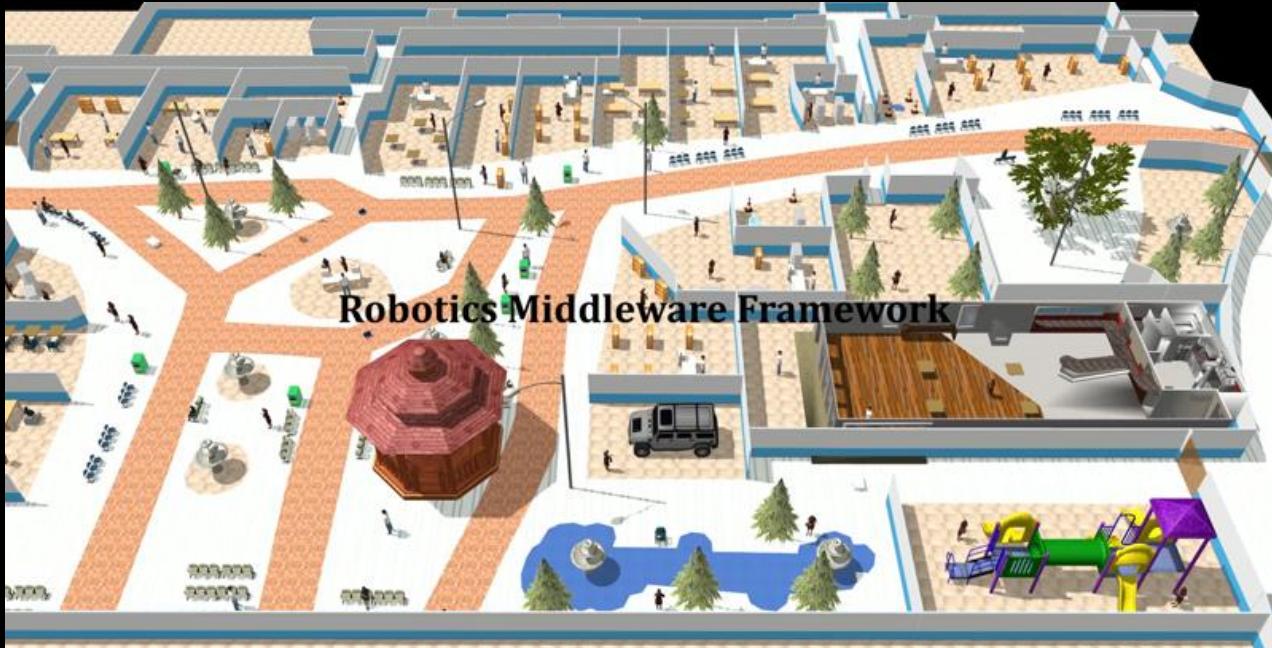


Open-RMF

A Common Language for Robot Interoperability



Lecture 7

정은빈

Contents

01 **RMF Panel**

02 **RMF-Web**

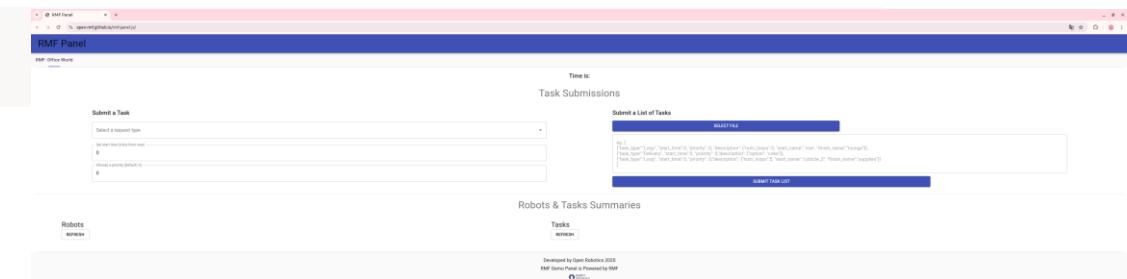
RMF Panel

RMF Panel

◐ RMF Panel이란?

- | 시각화 도구: 사용자에게 직관적으로 사용하기 쉬운 인터페이스 제공
- | 로봇 운영의 기본적인 상호작용 및 모니터링에 작업에 중점
- | 로봇의 현재 위치, 상태 예정된 경로 등 기본적인 정보 확인
- | RMF Panel 경로

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



RMF Panel

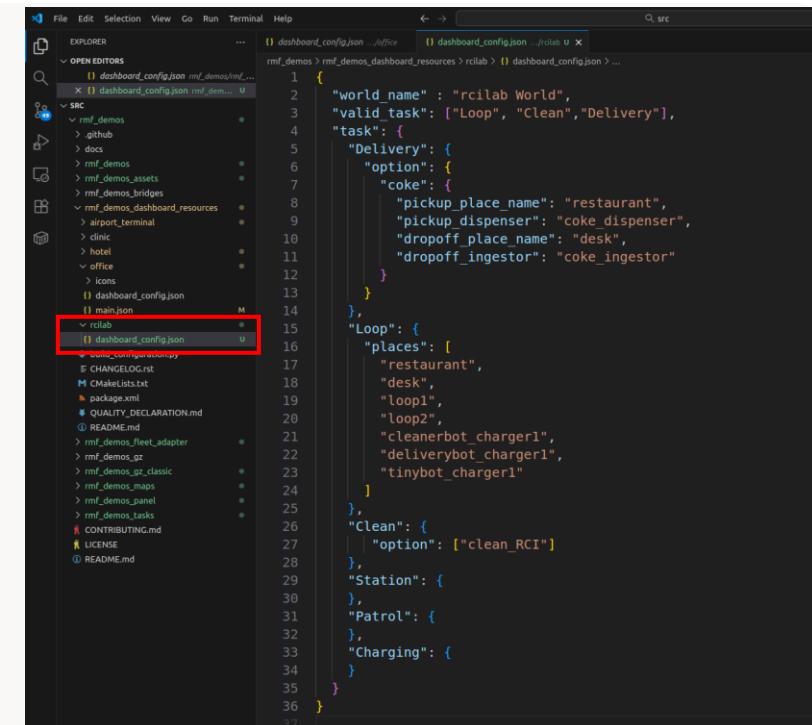
▶ Dashboard 설정

| rcilab 폴더 생성 및 dashboard_config.json 파일 생성

`~/rmf_ws/src/rmf_demos/rmf_demos_dashboard_resources/rcilab/dashboard_config.json`

dashboard_config.json 내용 작성

```
{
    "world_name": "rcilab World",
    "valid_task": ["Loop", "Clean", "Delivery"],
    "task": {
        "Delivery": {
            "option": {
                "coke": {
                    "pickup_place_name": "restaurant",
                    "pickup_dispenser": "coke_dispenser",
                    "dropoff_place_name": "desk",
                    "dropoff_ingestor": "coke_ingestor"
                }
            }
        },
        "Loop": {
            "places": [
                "restaurant",
                "desk",
                "loop1",
                "loop2",
                "cleanerbot_charger1",
                "deliverybot_charger1",
                "tinybot_charger1"
            ]
        },
        "Clean": {
            "option": ["clean_RCI"]
        },
        "Station": {
        },
        "Patrol": {
        },
        "Charging": {
        }
    }
}
```



RMF Panel

» Dashboard 설정

| CMakeLists.txt 내용 수정

/rmf_ws/src/rmf_demos/rmf_demos_dashboard_resources/CMakeLists.txt

```
cmake_minimum_required(VERSION 3.5)
project(rmf_demos_dashboard_resources LANGUAGES NONE)
find_package(ament_cmake REQUIRED)
ament_package()
set(outputs
${CMAKE_CURRENT_SOURCE_DIR}/airport_terminal/main.json
${CMAKE_CURRENT_SOURCE_DIR}/office/main.json)
if(${CMAKE_VERSION} VERSION_GREATER_EQUAL 3.12)
file(GLOB_RECURSE deps CONFIGURE_DEPENDS *.json)
else()
file(GLOB_RECURSE deps *.json)
endif()
list(FILTER deps EXCLUDE_REGEX main.json)
add_custom_command(OUTPUT ${outputs}
COMMAND python3 build_configuration.py
DEPENDS ${deps} build_configuration.py
WORKING_DIRECTORY ${CMAKE_CURRENT_SOURCE_DIR})
add_custom_target(build ALL DEPENDS ${outputs})
install(DIRECTORY
airport_terminal
office
hotel
clinic
rcilab ←추가
DESTINATION share/${PROJECT_NAME})
```

RMF Panel

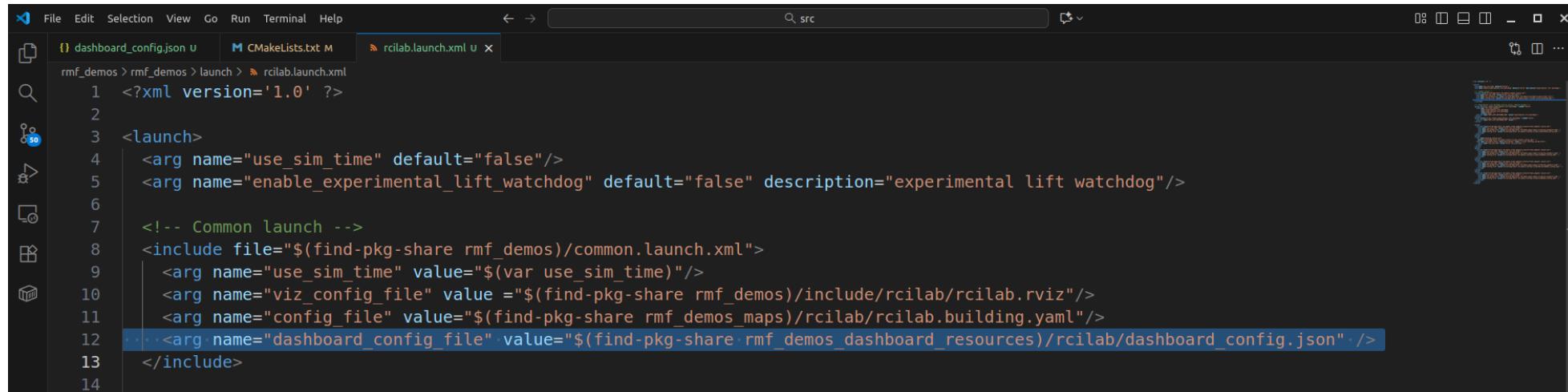
» Dashboard 설정

| rcilab.launch.xml 파일 수정

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

| rcilab.launch.xml 추가 내용

```
<arg name="dashboard_config_file" value="$(find-pkg-share rmf_demos_dashboard_resources)/rcilab/dashboard_config.json" />
```



```
1 <?xml version='1.0' ?>
2
3 <launch>
4   <arg name="use_sim_time" default="false"/>
5   <arg name="enable_experimental_lift_watchdog" default="false" description="experimental lift watchdog"/>
6
7   <!-- Common launch -->
8   <include file="$(find-pkg-share rmf_demos)/common.launch.xml">
9     <arg name="use_sim_time" value="$(var use_sim_time)"/>
10    <arg name="viz_config_file" value ="$(find-pkg-share rmf_demos)/include/rcilab/rcilab.rviz"/>
11    <arg name="config_file" value ="$(find-pkg-share rmf_demos_maps)/rcilab/rcilab.building.yaml"/>
12    <arg name="dashboard_config_file" value ="$(find-pkg-share rmf_demos_dashboard_resources)/rcilab/dashboard_config.json" />
13  </include>
14
```

RMF Panel

☞ rcilab world RMF Panel 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 server_uri:="ws://localhost:7878"
```

| RMF Panel 접속

```
https://open-rmf.github.io/rmf-panel-js/
```

RMF Panel

rcilab world RMF Panel 확인

The screenshot shows the RMF Panel web application. At the top, there's a navigation bar with tabs: 'RMF Panel' (selected), 'RMF: rcilab World' (highlighted with a red box), and others like 'RMF: open-robotics' and 'RMF: demo'. Below the navigation is a search bar with placeholder text 'Time is:'. The main content area is divided into several sections:

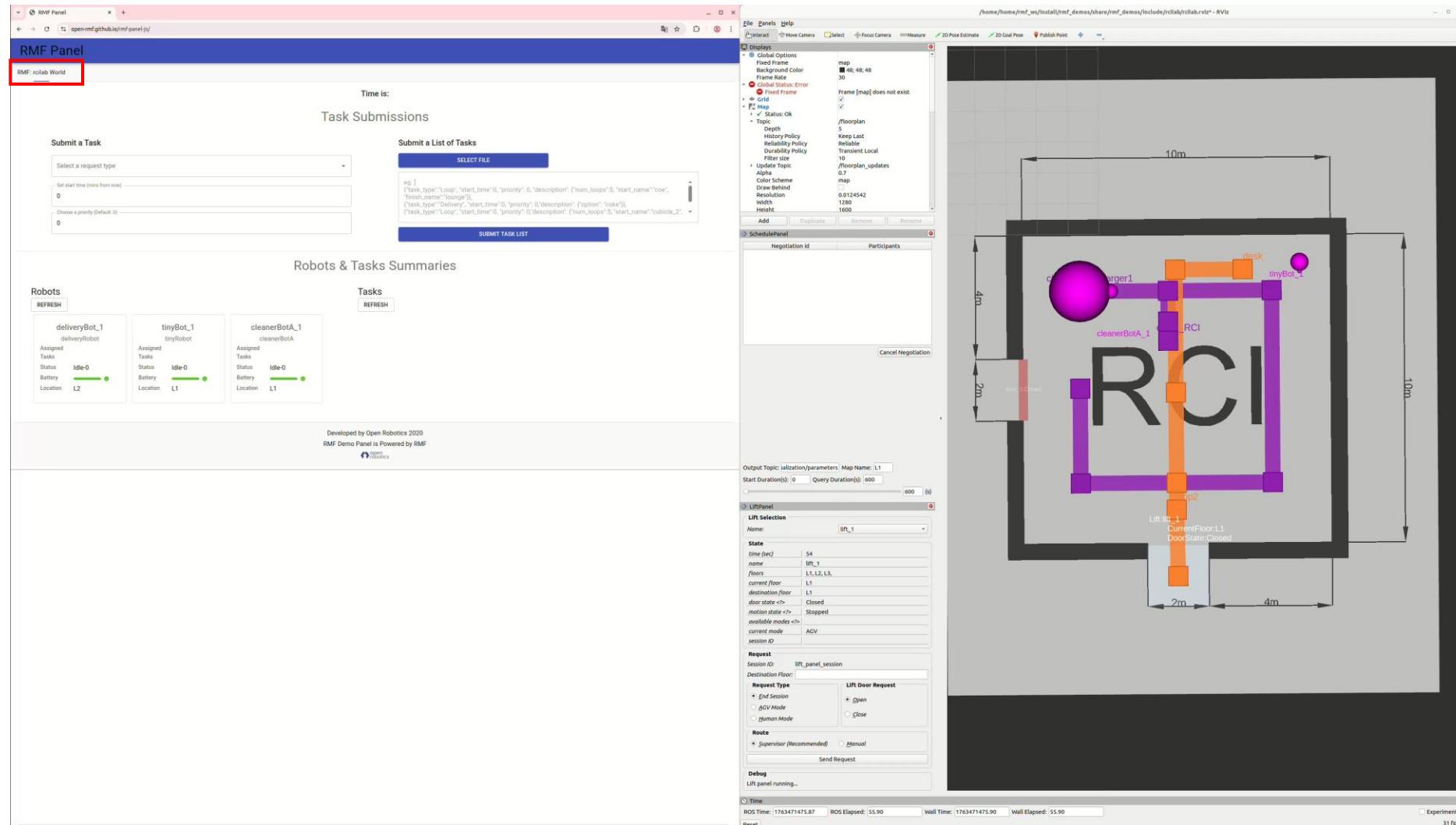
- Task Submissions:** Contains two forms:
 - Submit a Task:** A form for creating a single task. It includes dropdowns for 'Select a request type', input fields for 'Set start time (now from now)' (with value '0'), and 'Choose a priority (Default: 0)' (with value '0').
 - Submit a List of Tasks:** A form for uploading a file of tasks. It has a 'SELECT FILE' button and a text area containing sample JSON code:

```
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": 5, "start_name": "cubicle_2", "finish_name": "supplies" } ]
```

A 'SUBMIT TASK LIST' button is at the bottom.
- Robots & Tasks Summaries:** Contains two sections:
 - Robots:** A table with one row labeled 'REFRESH'.
 - Tasks:** A table with one row labeled 'REFRESH'.
- Footer:** Includes the text 'Developed by Open Robotics 2020' and 'RMF Demo Panel is Powered by RMF' with the Open Robotics logo.

RMF Panel

rcilab world RMF Panel에서 Clean Task 명령 생성



RMF-Web

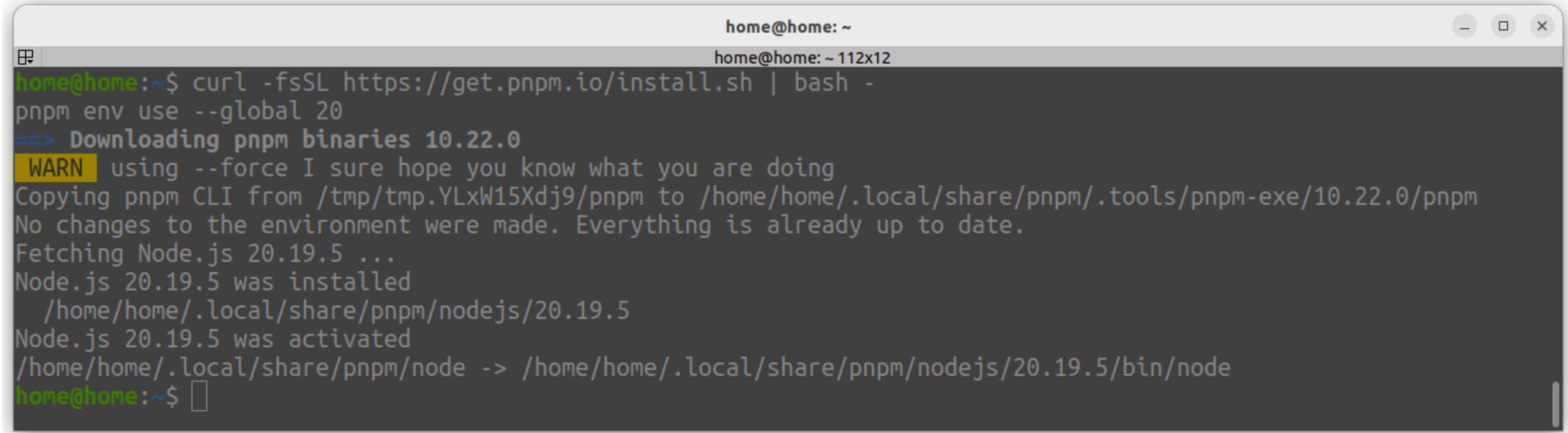
◐ RMF-Web이란?

- | 웹 기반 인터페이스: Open RFM의 모든 측면 시각화
- | RMF Panel보다 많은 기능을 제공(상위 호환)



⦿ RMF-Web 설치

```
curl -fsSL https://get.pnpm.io/install.sh | bash -  
pnpm env use --global 20
```



The screenshot shows a terminal window titled "home@home: ~". The user has run the command to install pnpm and Node.js. The output shows the download of pnpm binaries (version 10.22.0), a warning about using --force, the copying of pnpm CLI to the local share directory, and the activation of Node.js 20.19.5. The terminal window has a dark background with light-colored text.

```
home@home:~$ curl -fsSL https://get.pnpm.io/install.sh | bash -  
pnpm env use --global 20  
==> Downloading pnpm binaries 10.22.0  
WARN using --force I sure hope you know what you are doing  
Copying pnpm CLI from /tmp/tmp.YLxW15Xdj9/pnpm to /home/home/.local/share/pnpm/.tools/pnpm-exe/10.22.0/pnpm  
No changes to the environment were made. Everything is already up to date.  
Fetching Node.js 20.19.5 ...  
Node.js 20.19.5 was installed  
/home/home/.local/share/pnpm/nodejs/20.19.5  
Node.js 20.19.5 was activated  
/home/home/.local/share/pnpm/node -> /home/home/.local/share/pnpm/nodejs/20.19.5/bin/node  
home@home:~$
```

④ RMF-Web 설치

```
curl -fsSL https://get.pnpm.io/install.sh | bash -  
pnpm env use --global 20
```

```
pip3 install pipenv
```

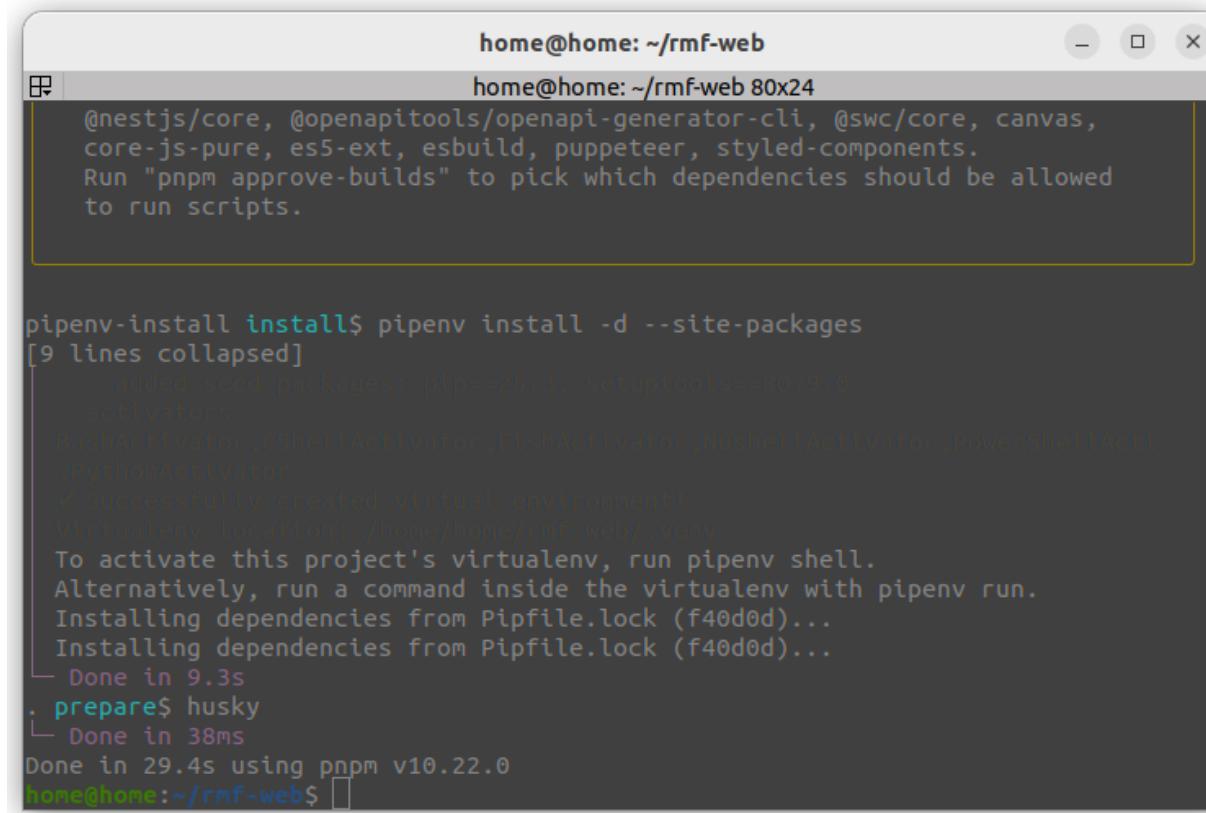
```
sudo apt install python3-venv
```

```
cd ~  
git clone https://github.com/open-rmf/rmf-web.git -b humble
```

```
cd rmf-web/  
pnpm install
```

④ RMF-Web 설치

```
cd rmf-web/  
pnpm install
```



The screenshot shows a terminal window titled "home@home: ~/rmf-web". The terminal displays the following output:

```
home@home: ~/rmf-web  
home@home: ~/rmf-web 80x24  
@nestjs/core, @openapitools/openapi-generator-cli, @swc/core, canvas,  
core-js-pure, es5-ext, esbuild, puppeteer, styled-components.  
Run "pnpm approve-builds" to pick which dependencies should be allowed  
to run scripts.  
  
pipenv-install install$ pipenv install -d --site-packages  
[9 lines collapsed]  
    added seed packages: plpy==25.3, setuptools==30.9.0  
    activators  
    BashActivator, CshActivator, FishActivator, HushActivator, PowerShellActivator,  
    PythonActivator  
✓ Successfully created virtual environment!  
Virtualenv location: /home/home/rmf-web/.venv  
To activate this project's virtualenv, run pipenv shell.  
Alternatively, run a command inside the virtualenv with pipenv run.  
Installing dependencies from Pipfile.lock (f40d0d)...  
Installing dependencies from Pipfile.lock (f40d0d)...  
└─ Done in 9.3s  
· prepare$ husky  
└─ Done in 38ms  
Done in 29.4s using pnpm v10.22.0  
home@home:~/rmf-web$
```

④ RMF-Web server

```
source /opt/ros/humble/setup.bash  
source ~/rmf_ws/install/setup.bash
```

```
cd ~/rmf-web/packages/dashboard
```

```
pnpm start
```

<http://localhost:3000>로 자동 접속



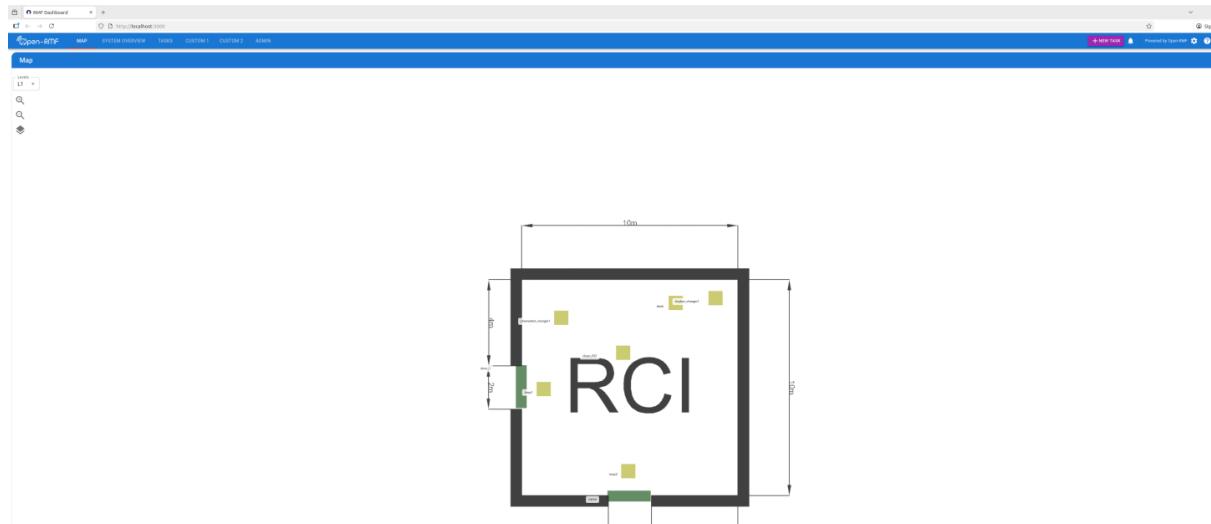
RMF-Web

⦿ RMF-Web server

| rcilab.launch.xml 실행

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

```
ros2 launch rmf_demos_gz_classic rcilab.launch.xml server_uri:="http://localhost:8000/_internal"
```



» RMF-Web server

| System overview

The screenshot displays the RMF-Web system overview dashboard. The top navigation bar includes links for Open-RMF, Map, SYSTEM OVERVIEW (highlighted with a red box), TASKS, CUSTOM 1, CUSTOM 2, and ADMIN. The dashboard is divided into several sections:

- Robots:** A table showing three robots: cleanerBot_1 (cleanerBot, L1, 100%, IDLE), deliveryBot_1 (deliveryRobot, L2, 100%, IDLE), and tinyBot_1 (tinyRobot, L1, 100%, IDLE).
- Doors:** A table showing two doors: door_1 (ONLINE, L1, Double Swing, CLOSED) and L2_door_1 (ONLINE, L2, Double Swing, CLOSED). Buttons for OPEN and CLOSE are present.
- Lifts:** A table showing one lift: lift_1 (ONLINE, L1, L1, L1, CLOSED, REQUEST).
- Map:** A floor plan of a building with dimensions 10m by 10m. It shows various rooms and objects, including a large central area labeled "RCI" and several green circular spots labeled "medium_1 spot" and "large_1 spot". A green rectangle at the bottom is labeled "NORMA" and "L1".

⦿ RMF-Web server

| Clean 명령 생성

RMF Dashboard - http://localhost:3000/robots

Open - RMF SYSTEM OVERVIEW TASKS CUSTOM 1 CUSTOM 2 ADMIN

Robots

Name	Fleet	Est. Task Finish Time	Level	Battery	Last Updated	Status
cleanerBotA_1	cleanerBotA	-	L1	100%	1/1/1970, 9:00:37 AM	IDLE
deliveryBot_1	deliveryRobot	-	L2	100%	1/1/1970, 9:00:37 AM	IDLE
tinyBot_1	tinyRobot	-	L1	100%	1/1/1970, 9:00:37 AM	IDLE

Doors

Name	Op. Mode	Current Floor	Type	Door State	OPEN CLOSE
door_1	ONLINE	L1	Double Swing	CLOSED	OPEN CLOSE
L2_door_1	ONLINE	L2	Double Swing	CLOSED	OPEN CLOSE

Lifts

Name	Op. Mode	Current Floor	Destination Floor	Lift State	REQUEST
lift_1	ONLINE	L1	L1	CLOSED	REQUEST

Map

RMF-Web

⦿ RMF-Web server

| Task 확인

The screenshot displays the RMF Dashboard interface. The top navigation bar includes links for Open-RMF, MAP, SYSTEM OVERVIEW, **TASKS** (which is highlighted with a red box), CUSTOM 1, CUSTOM 2, and ADMIN. Below the navigation is a header with '+ NEW TASK', a bell icon, and 'Powered by Open-RMF' along with other system status indicators.

Tasks section:

Date	Requester	ID	Category	Assignee	Start Time	End Time	State
11/17/2025	stub	clean.dispatch	Clean	cleanerBotA	1/1/1970 9:03:20 AM	1/1/1970 9:04:23 AM	completed
11/17/2025	stub	clean.dispatch	Clean	cleanerBotA	1/1/1970 9:01:02 AM	1/1/1970 9:02:04 AM	completed

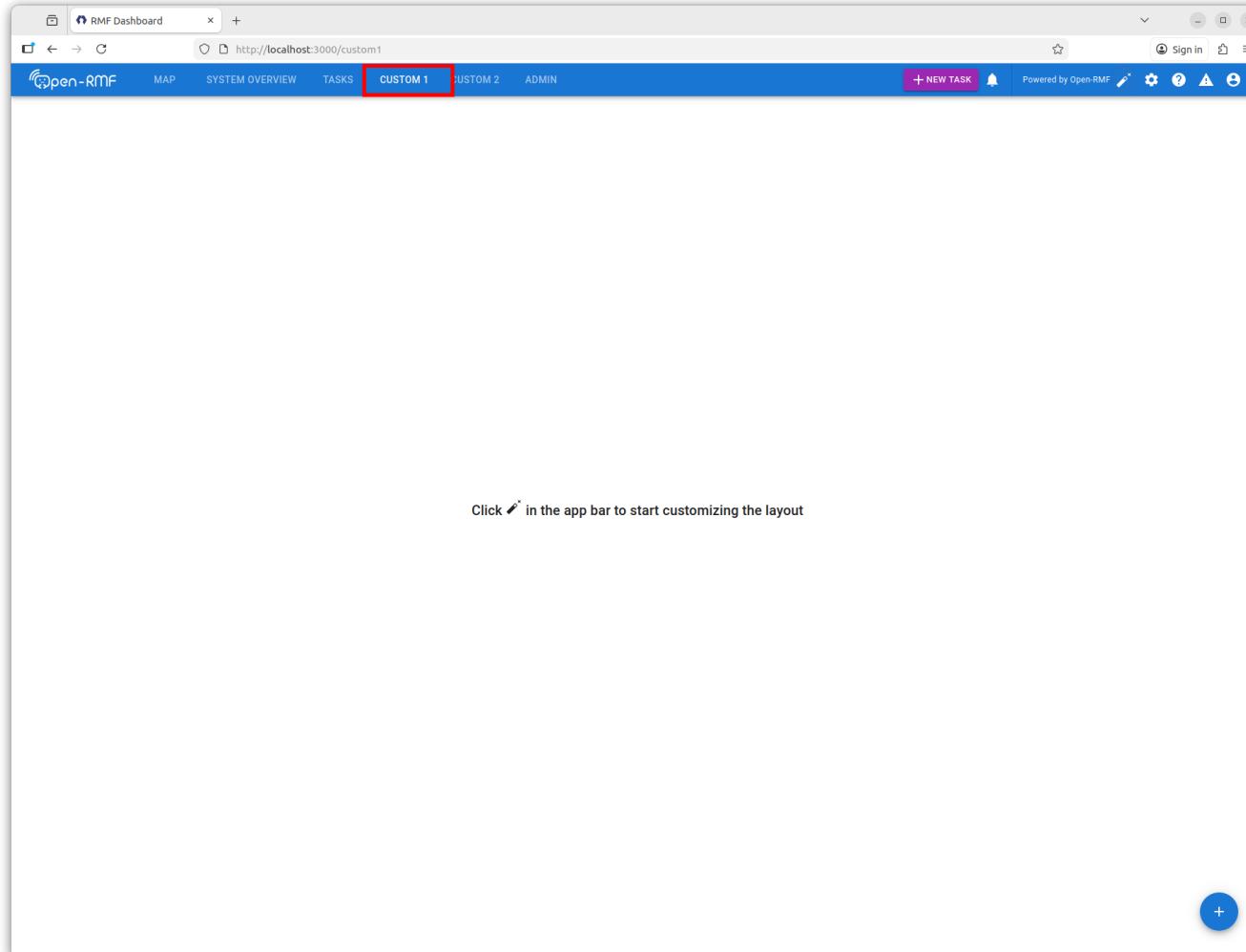
Map section:

The map shows a rectangular room with dimensions of 10m by 10m. Inside the room, there is a large black rectangle containing the letters "RCI". A yellow circle labeled "MyBot_1" is positioned near the top right of the "RCI" area. A green square labeled "charger1" is located on the left side. A red rectangle labeled "loop1" is at the bottom left, and a green square labeled "loop2" is at the bottom center. A small green square labeled "desk" is near the top right. A red rectangle labeled "norma" is at the bottom center. A green rectangle labeled "L1" is at the bottom right. Arrows indicate the boundaries of the room and the paths between objects.

RMF-Web

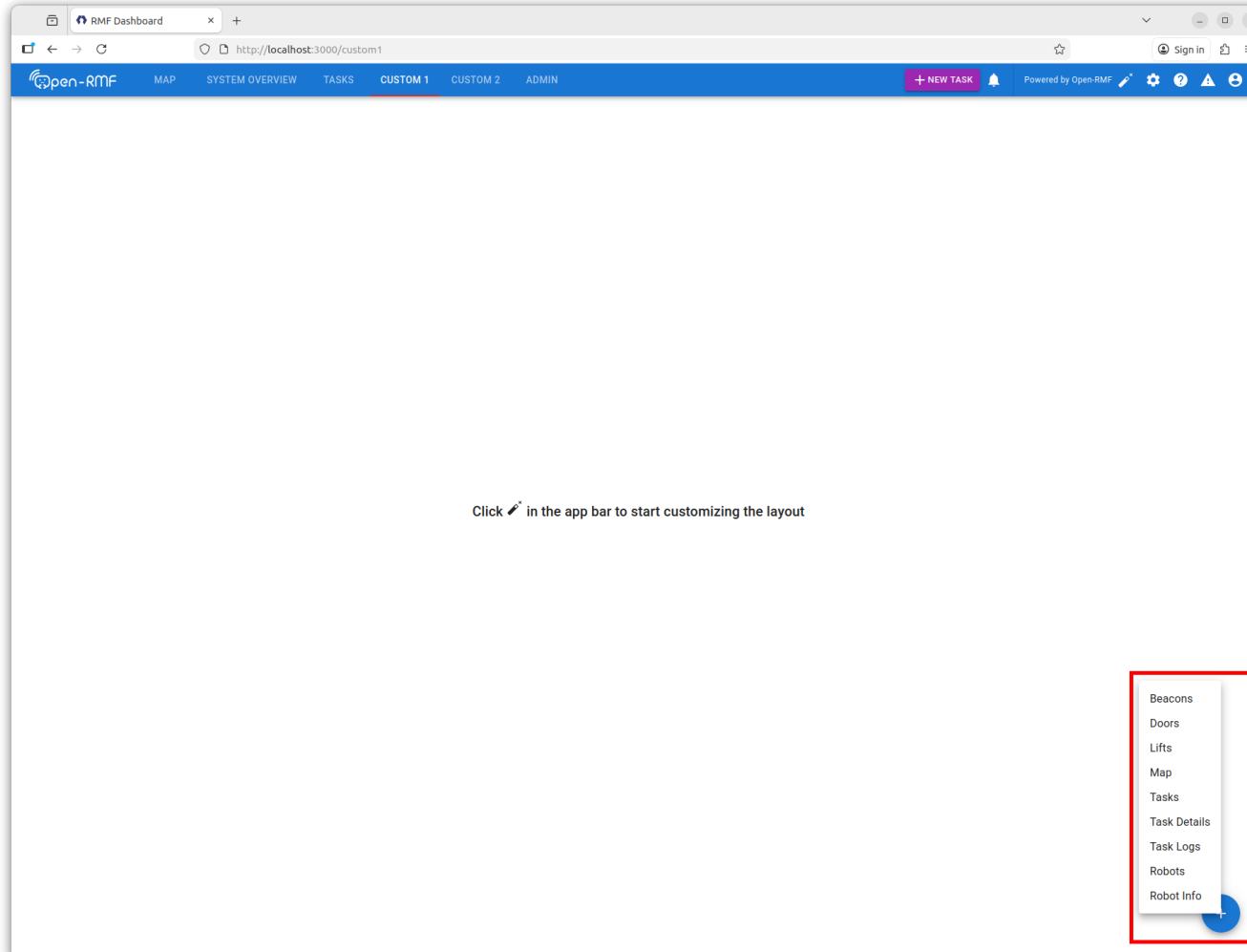
» RMF-Web server

| Custom



» RMF-Web server

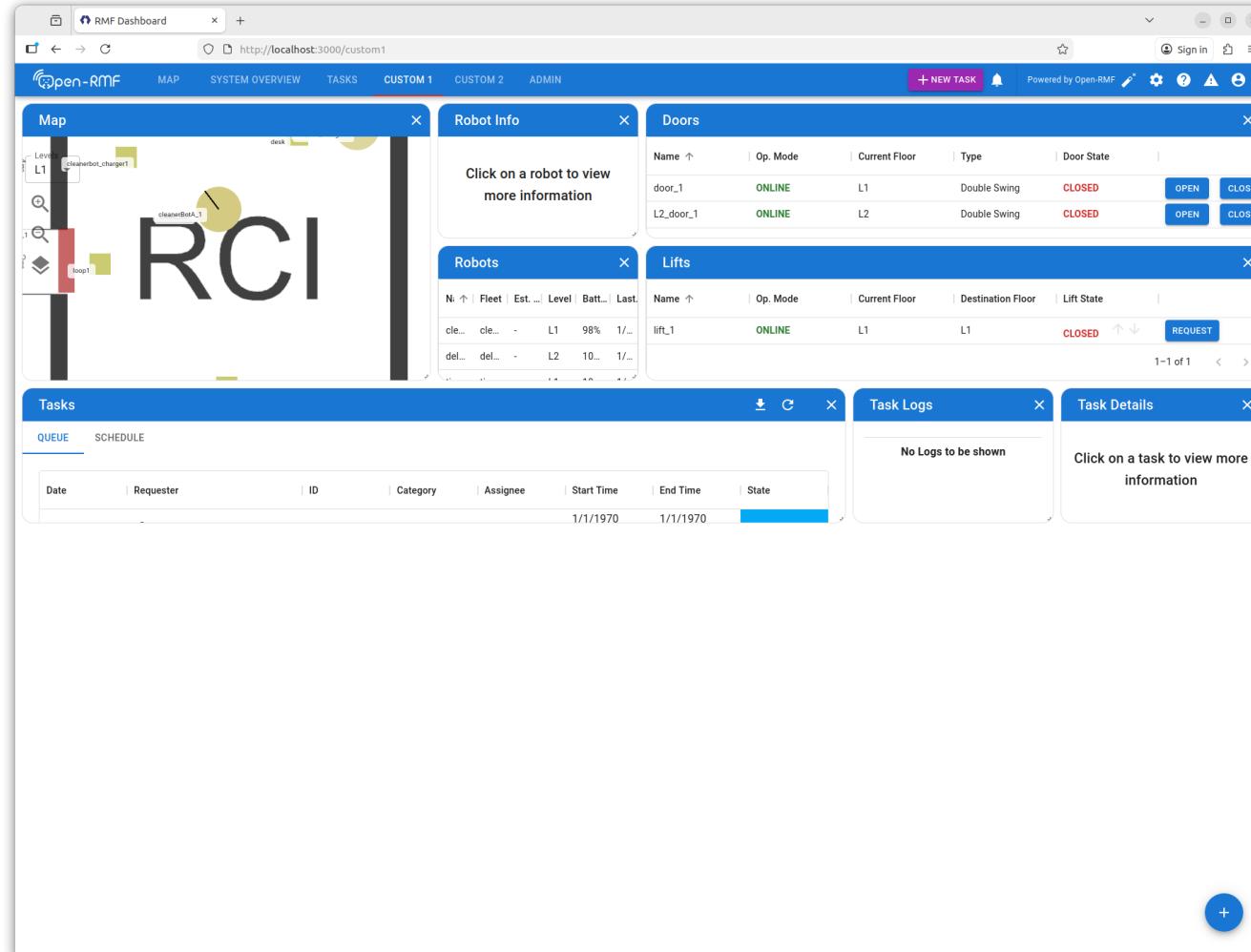
| Custom



RMF-Web

» RMF-Web server

| Custom



감사합니다