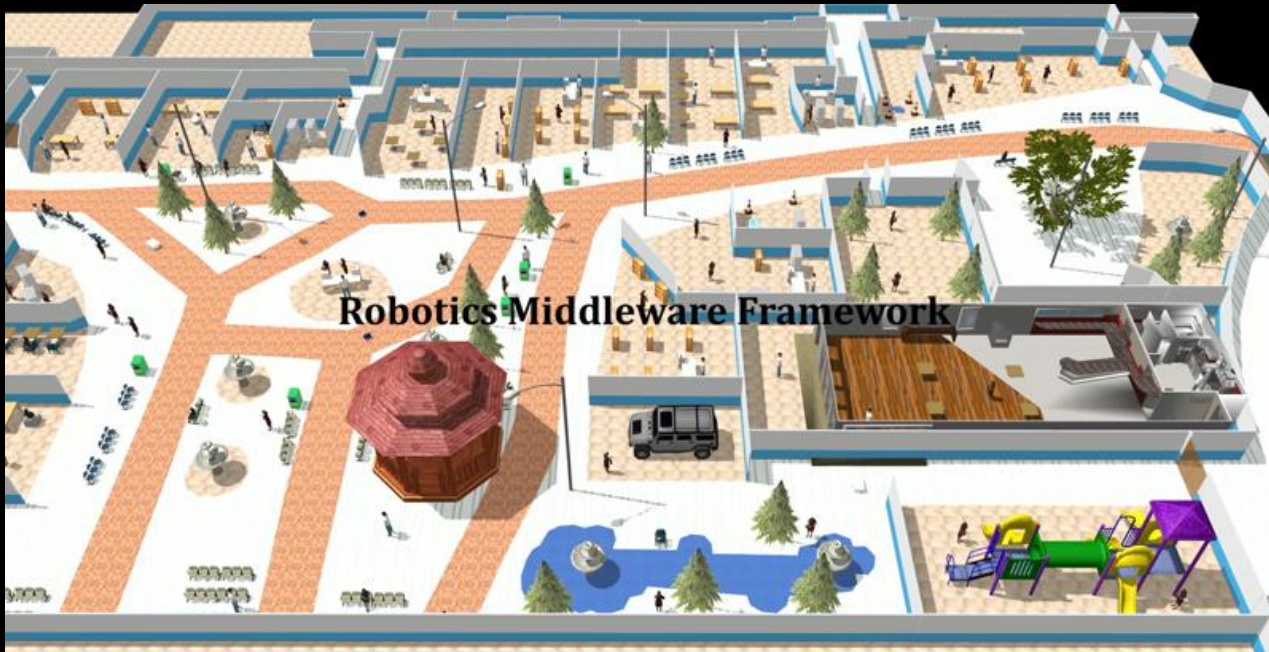




A Common Language for Robot Interoperability



Lecture 7

정은빈

Contents

01 **RMF Panel**

02 **RMF-Web**

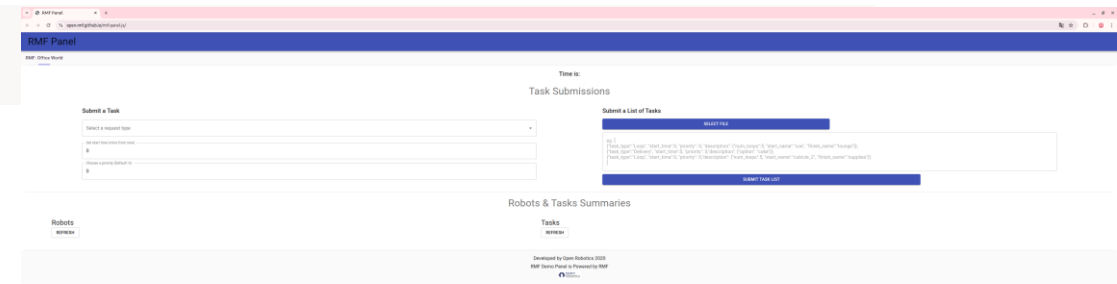
RMF Panel

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

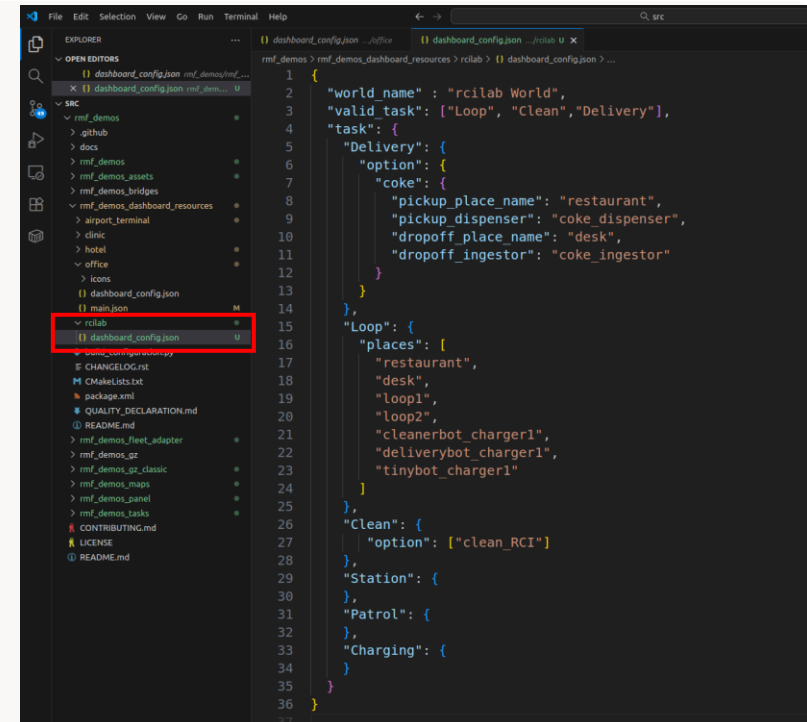
Dashboard 설정

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": {
    }
  }
}
```

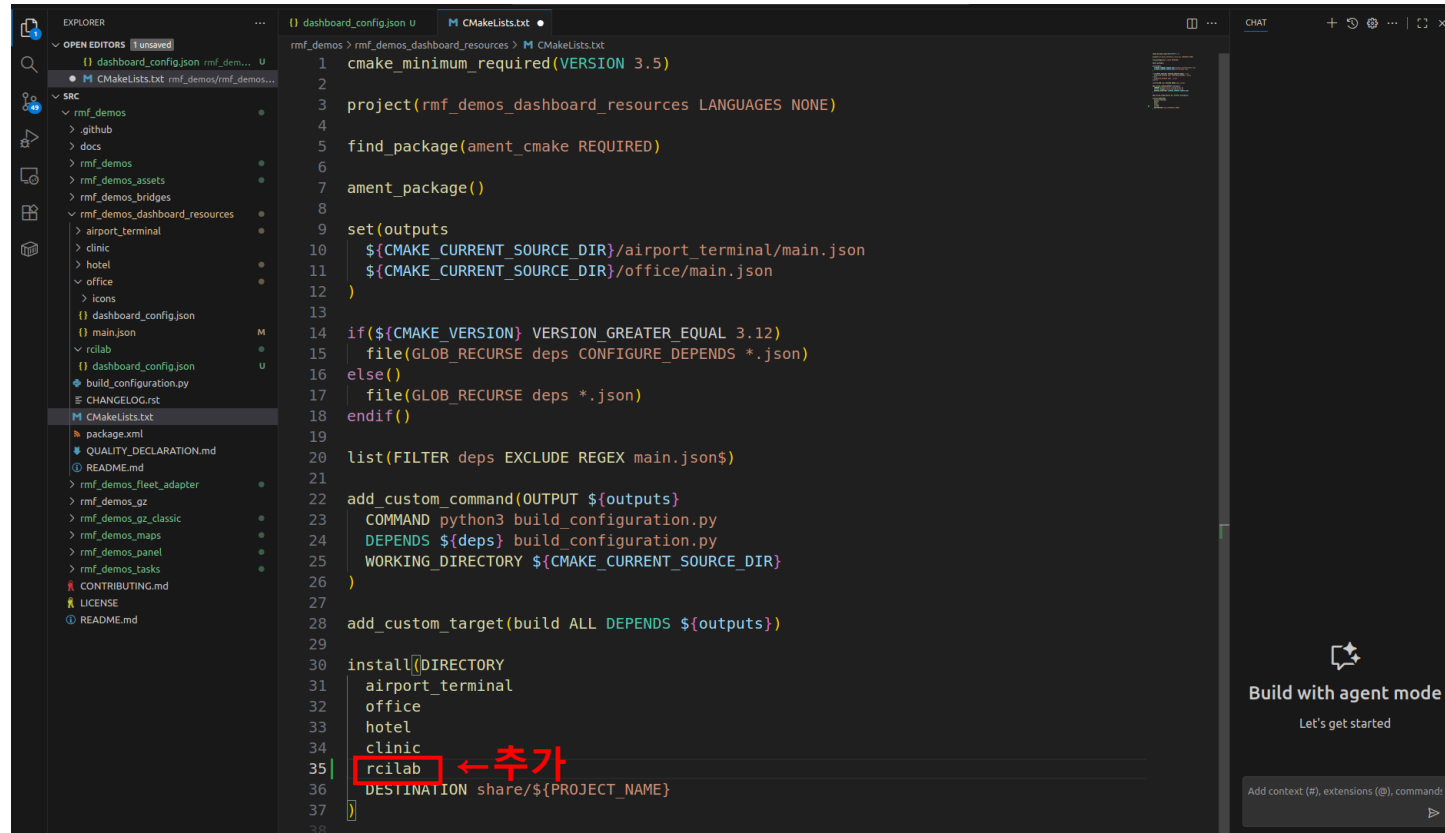


RMF Panel

Dashboard 설정

CMakeLists.txt 내용 수정

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



```
1 cmake_minimum_required(VERSION 3.5)
2
3 project(rmf_demos_dashboard_resources LANGUAGES NONE)
4
5 find_package(ament_cmake REQUIRED)
6
7 ament_package()
8
9 set(outputs
10   ${CMAKE_CURRENT_SOURCE_DIR}/airport_terminal/main.json
11   ${CMAKE_CURRENT_SOURCE_DIR}/office/main.json
12 )
13
14 if(${CMAKE_VERSION} VERSION_GREATER_EQUAL 3.12)
15   file(GLOB_RECURSE deps CONFIGURE_DEPENDS *.json)
16 else()
17   file(GLOB_RECURSE deps *.json)
18 endif()
19
20 list(FILTER deps EXCLUDE REGEX main.json$)
21
22 add_custom_command(OUTPUT ${outputs}
23   COMMAND python3 build_configuration.py
24   DEPENDS ${deps} build_configuration.py
25   WORKING_DIRECTORY ${CMAKE_CURRENT_SOURCE_DIR}
26 )
27
28 add_custom_target(build ALL DEPENDS ${outputs})
29
30 install(DIRECTORY
31   airport_terminal
32   office
33   hotel
34   clinic
35   rcilab ← 추가
36   DESTINATION share/${PROJECT_NAME})
37
38
```

RMF Panel

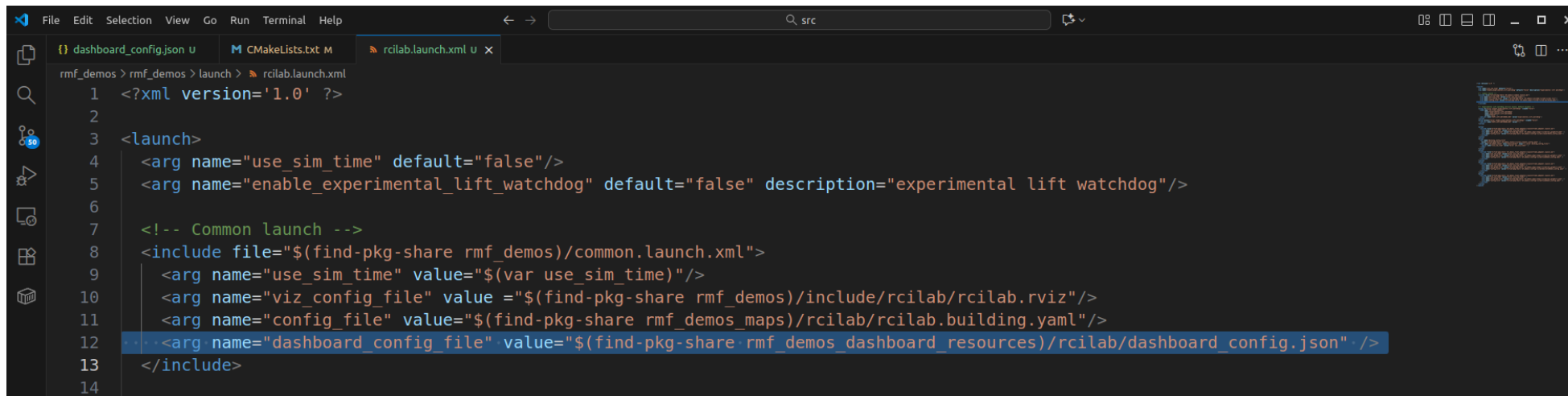
🔗 Dashboard 설정

▮ 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 displays the RMF Panel web interface in a browser window. The address bar shows the URL `open-rmf.github.io/rmf-panel/`. The page has a blue header with the text "RMF Panel" and a sub-header "RMF: rcilab World" which is highlighted with a red box. The main content area is divided into several sections:

- Time is:** A section for time-related information.
- Task Submissions:** A section for submitting tasks, containing two forms:
 - Submit a Task:** Includes a dropdown for "Select a request type", a text input for "Set start time (now from now)" with the value "0", and a text input for "Choose a priority (Default: 0)" with the value "0".
 - Submit a List of Tasks:** Includes a "SELECT FILE" button, a text area for a JSON task list (with an example provided), and a "SUBMIT TASK LIST" button.
- Robots & Tasks Summaries:** A section for viewing summaries, containing two sub-sections:
 - Robots:** Includes a "REFRESH" button.
 - Tasks:** Includes a "REFRESH" button.

The footer of the page contains the text: "Developed by Open Robotics 2020", "RMF Demo Panel is Powered by RMF", and a small logo.

RMF Panel

rcilab world RMF Panel에서 Clean Task 명령 생성

The screenshot displays the RMF Panel interface, which is used for managing robots and tasks in a simulated environment. The interface is divided into several sections:

- RMF Panel Header:** Includes a red box highlighting the "RMF: rcilab World" tab.
- Task Submissions:** Contains a "Submit a Task" section with a dropdown for "Select a request type" (set to "0") and a "Submit a List of Tasks" section with a "SELECT FILE" button and a "SUBMIT TASK LIST" button.
- Robots & Tasks Summaries:** Features a "Robots" section with a "REFRESH" button and three robot status cards: "deliveryBot_1" (Idle-0, L2), "tinyBot_1" (Idle-0, L1), and "cleanerBotA_1" (Idle-0, L1). A "Tasks" section with a "REFRESH" button is also present.
- Developed by Open Robotics 2020:** A footer note indicating the software's origin.
- Right Panel:** Displays a 2D map of the environment with dimensions (10m x 10m). It shows the positions of robots (deliveryBot_1, tinyBot_1, cleanerBotA_1) and a task (task) being assigned. The map also includes a "RCI" (Robot Control Interface) label.
- Bottom Panel:** Contains a "LIFTPanel" section with a "Name" dropdown (set to "L1_1") and a "State" section showing details like "time (sec)", "name", "floors", "current floor", "destination floor", "door state", "motion state", "available modes", and "current mode". It also includes a "Request" section with a "Session ID" dropdown (set to "L1F_panel_session") and a "Destination Floor" dropdown (set to "L1"). The "Request Type" is set to "LIFT Door Request" with "Open" selected. A "Route" section shows "Supervisor Recommended" and "Manual" options. A "Send Request" button is located at the bottom of this section.
- Bottom Status Bar:** Displays system information including "ROS Time: 1763471475.87", "ROS Elapsed: 55.90", "Wall Time: 1763471475.90", "Wall Elapsed: 55.90", and "FPS: 31 fps".

RMF-Web

RMF-Web

🔗 RMF-Web이란?

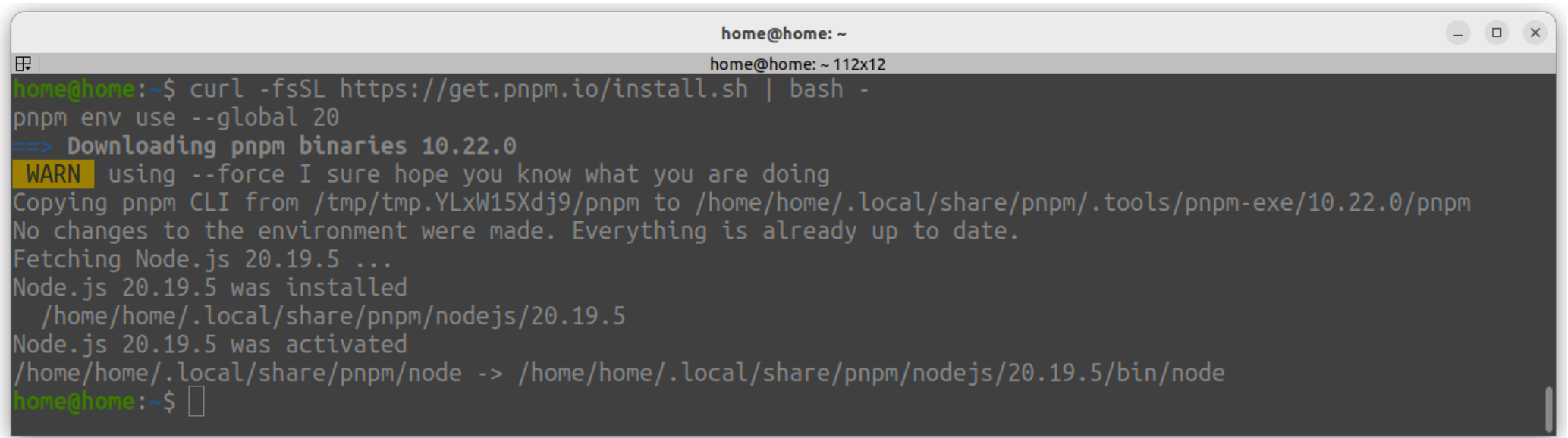
- ▮ 웹 기반 인터페이스: Open RFM의 모든 측면 시각화
- ▮ RMF Panel보다 많은 기능을 제공(상위 호환)



RMF-Web

🔗 RMF-Web 설치

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

A terminal window titled 'home@home: ~' showing the execution of the pnpm installation script. The user runs 'curl -fsSL https://get.pnpm.io/install.sh | bash -' followed by 'pnpm env use --global 20'. The terminal output shows the download of pnpm binaries (10.22.0), a warning about using --force, the copying of the pnpm CLI, and the fetching and installation of Node.js (20.19.5). The final output shows the activation of Node.js and the path to the node binary.

```
home@home: ~  
home@home: ~ 112x12  
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:~$
```

RMF-Web

🔗 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
```

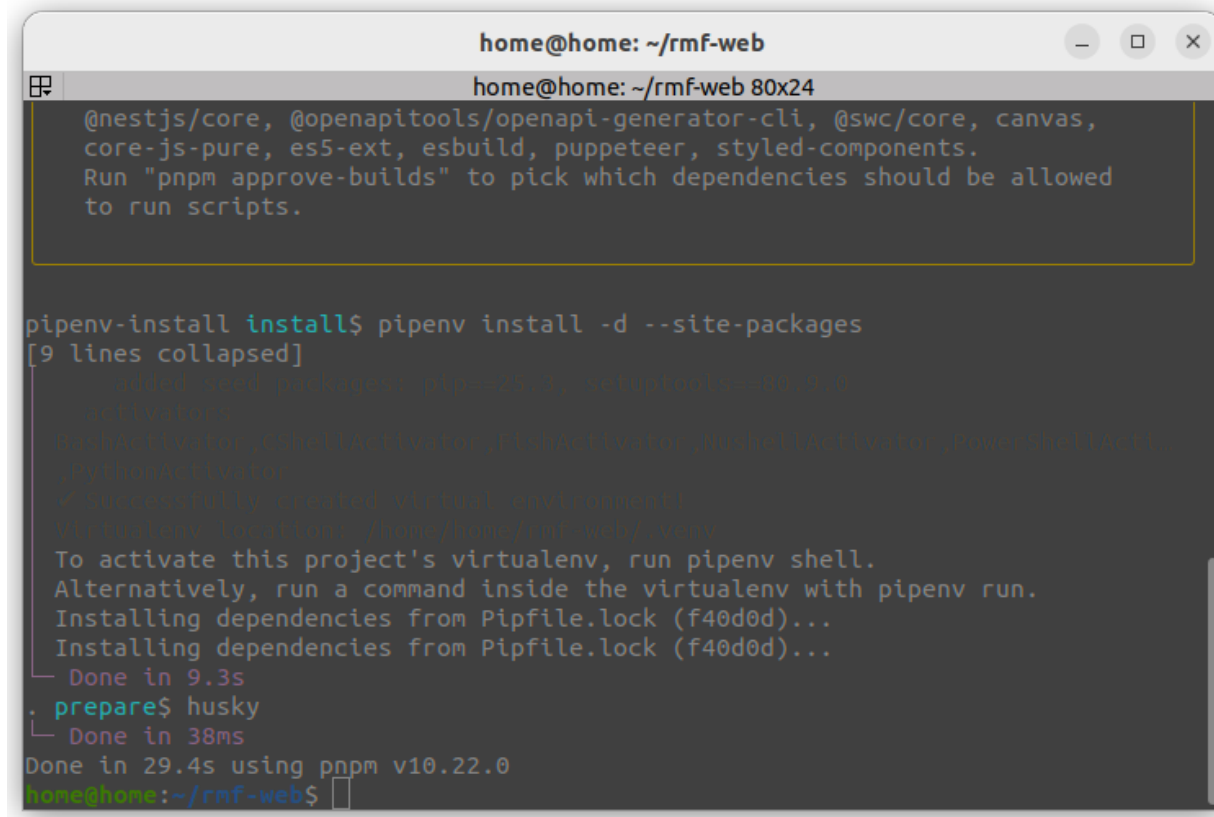
```
cd ~  
git clone https://github.com/open-rmf/rmf-web.git -b humble
```

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

RMF-Web

🕒 RMF-Web 설치

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



```
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]  
  virtualenv-clone, pipenv, pipenv-standalone, pipenv-standalone,  
  activator  
  pipenv-standalone, pipenv-standalone, pipenv-standalone, pipenv-standalone,  
  pipenv-standalone  
  ✓ 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$
```

RMF-Web

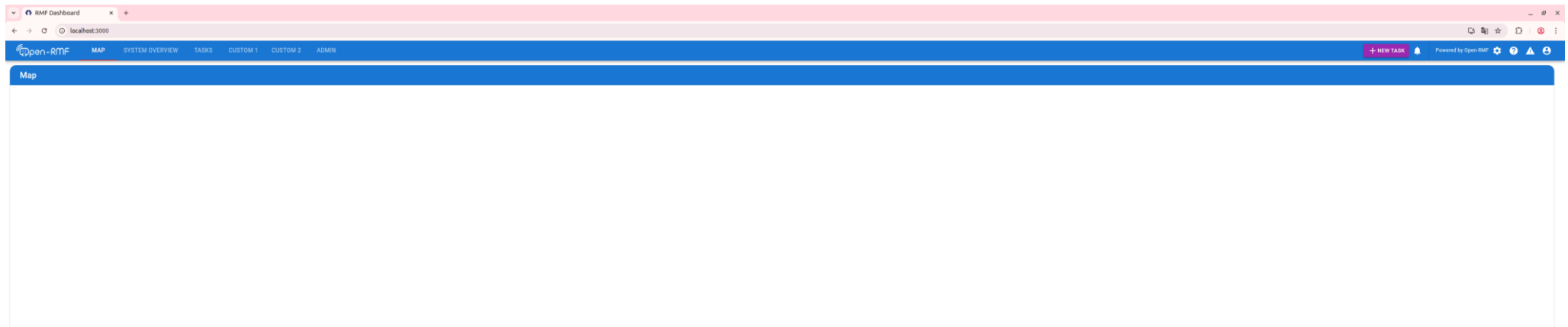
🔗 RMF-Web server

```
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

- RMF-Web server
 - ▮ System overview

The screenshot displays the RMF-Web System Overview interface. The top navigation bar includes links for HOME, SYSTEM OVERVIEW (highlighted with a red box), TASKS, CUSTOM 1, CUSTOM 2, and ADMIN. The main content area is divided into three sections: Robots, Doors, and Lifts, each with a table of system components and their status.

Robots Table:

Name	Fleet	Est. Task Finish Time	Level	Battery	Last Updated	Status
cleanerBot_1	cleanerBotA	-	L1	100%	1/1/1970, 9:01:40 AM	IDLE
deliveryBot_1	deliveryRobot	-	L2	100%	1/1/1970, 9:01:39 AM	IDLE
tinyBot_1	tinyRobot	-	L1	100%	1/1/1970, 9:01:39 AM	IDLE

Doors Table:

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

Lifts Table:

Name	Op. Mode	Current Floor	Destination Floor	Lift State	Actions
IR_1	ONLINE	L1	L1	CLOSED	REQUEST

On the right side of the interface is a 'Map' section showing a floor plan of a square room with dimensions 10m by 10m. The room contains several objects: a red rectangle labeled 'door', a yellow circle labeled 'target', a green square labeled 'NORMAL L1', and a large black 'X' labeled 'RCI'. The map also shows a 'door' label on the left wall and a 'door' label on the right wall.

RMF-Web

▶ RMF-Web server

┆ Clean 명령 생성

The screenshot displays the RMF-Web interface, which is divided into two main sections: a data table on the left and a map on the right.

Robots Table:

Name	Fleet	Est. Task Finish Time	Level	Battery	Last Updated	Status
cleanerBot_1	cleanerBot	-	L1	100%	1/1/1970, 9:00:37 AM	IDLE
deliveryBot_1	deliveryRobot	-	L2	100%	1/1/1970, 9:00:37 AM	IDLE
trayBot_1	trayRobot	-	L1	100%	1/1/1970, 9:00:37 AM	IDLE

Doors Table:

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

Lifts Table:

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

Map:

The map shows a floor plan with dimensions 10m by 10m. It includes a central area labeled 'RCI' and a bottom section labeled 'NORMAL L1'. Various colored shapes (yellow circles, green squares, red rectangles) represent robots and other elements on the floor.

RMF-Web

RMF-Web server

Task 확인

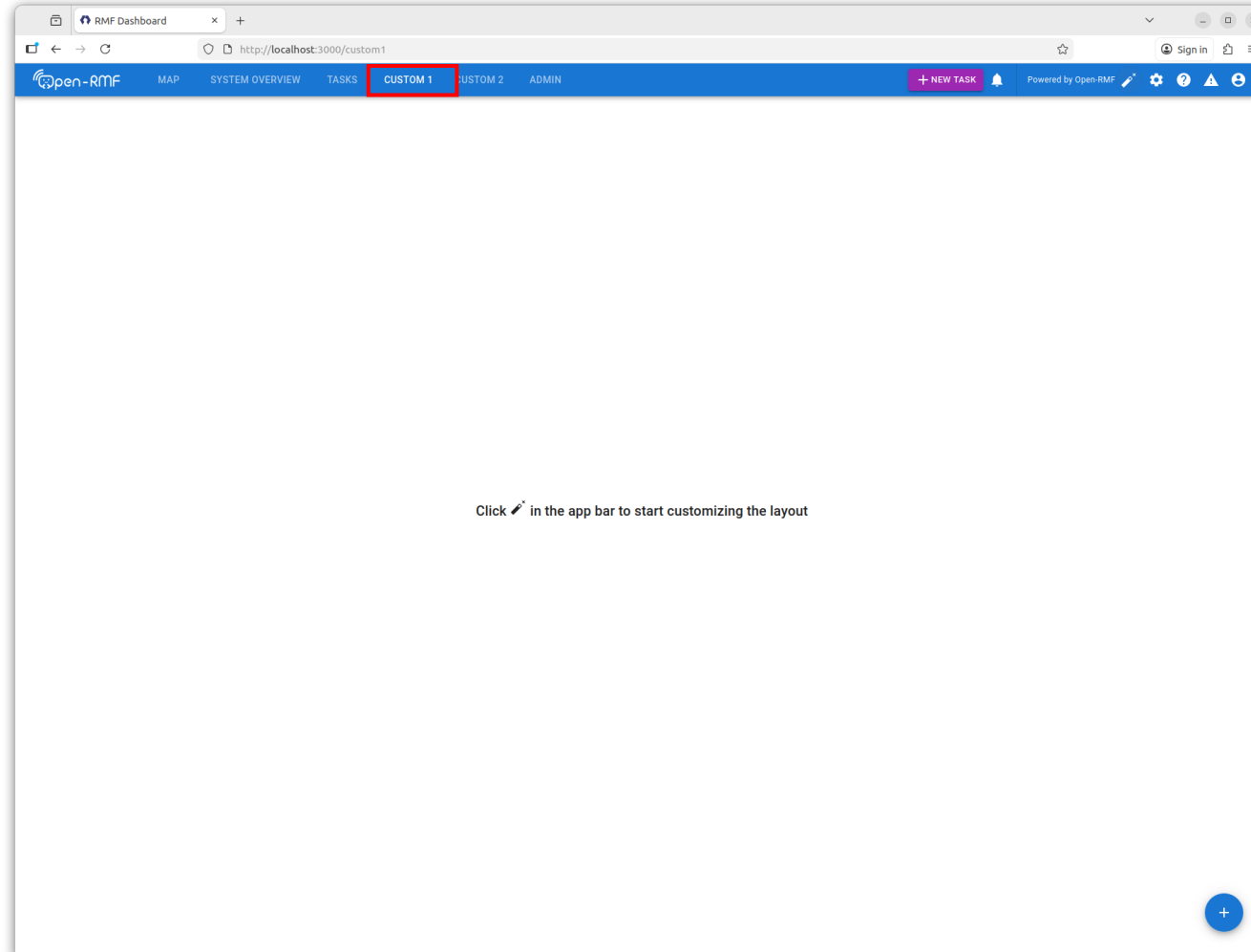
The screenshot displays the RMF-Web interface. The top navigation bar includes 'open-RMF', 'MAP', 'SYSTEM OVERVIEW', 'TASKS' (highlighted with a red box), 'CUSTOM 1', 'CUSTOM 2', and 'ADMIN'. A '+ NEW TASK' button is visible on the right. The 'Tasks' section is active, showing a table with task details. The table has columns for Date, Requester, ID, Category, Assignee, Start Time, End Time, and State. Two tasks are listed, both with a 'completed' state. The 'Map' section on the right shows a top-down view of a square environment with dimensions 10m by 10m. It features a central 'RCI' label, a 'cleanerBotA_1' robot, a 'cleanerBot_charger1' station, and a 'tinyBot_L1_viper1' robot. A 'loop1' and 'loop2' are also indicated. A 'NORMAL L1' label is visible at the bottom of the map.

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

RMF-Web

▶ RMF-Web server

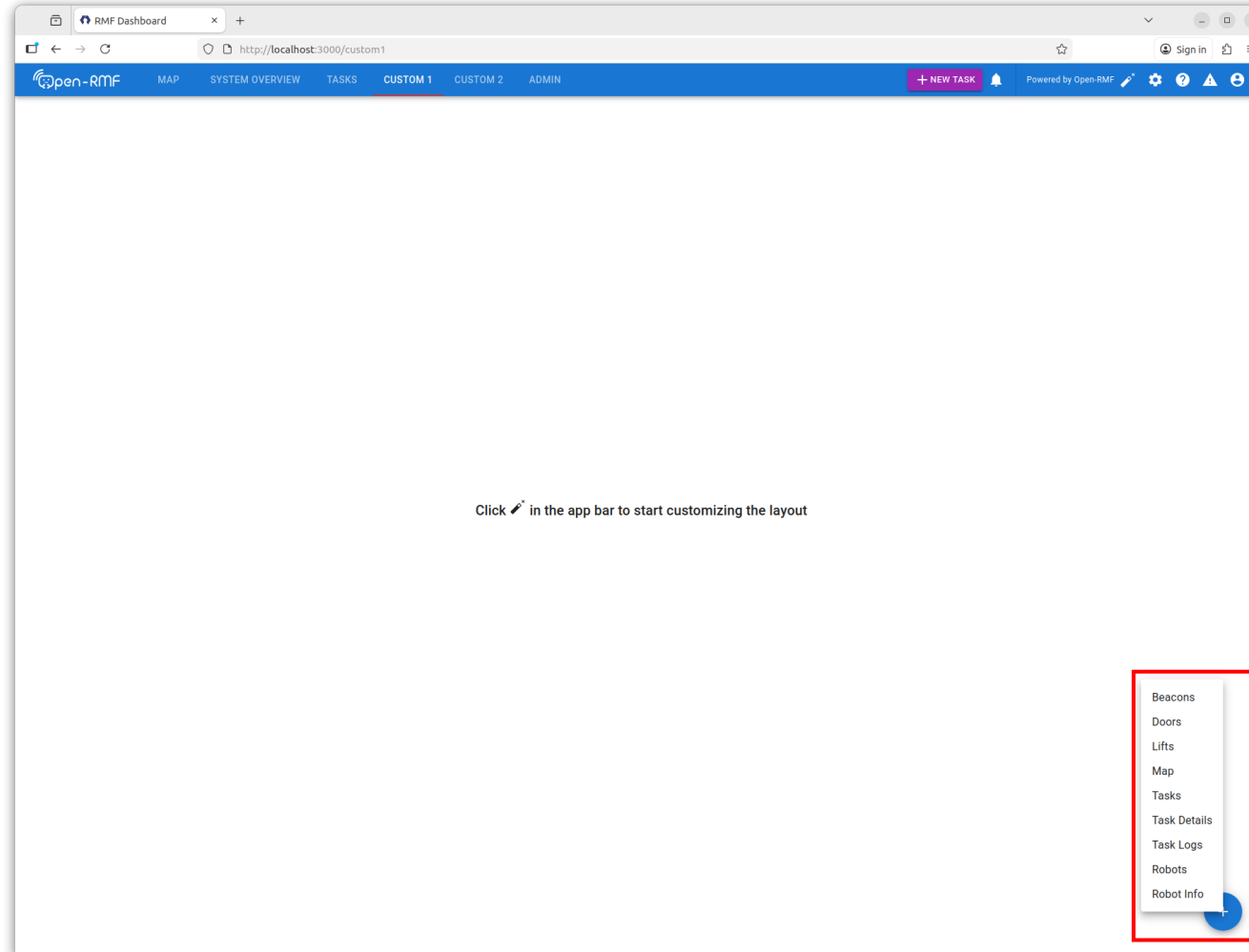
┆ Custom



RMF-Web

➤ RMF-Web server

┆ Custom



RMF-Web

➤ RMF-Web server

Custom

The screenshot displays the RMF-Web dashboard interface. The top navigation bar includes links for MAP, SYSTEM OVERVIEW, TASKS, CUSTOM 1, CUSTOM 2, and ADMIN. A '+ NEW TASK' button and a 'Sign in' link are also present. The main content area is divided into several panels:

- Map:** Shows a floor plan with a robot (yellow circle) and a door (red rectangle). The text 'RCI' is visible on the map.
- Robot Info:** A panel with the text 'Click on a robot to view more information'.
- Doors:** A table listing doors with columns: Name, Op. Mode, Current Floor, Type, and Door State. It includes 'OPEN' and 'CLOSE' buttons for each door.
- Lifts:** A table listing lifts with columns: Name, Op. Mode, Current Floor, Destination Floor, and Lift State. It includes a 'REQUEST' button for each lift.
- Tasks:** A panel with tabs for 'QUEUE' and 'SCHEDULE'. It shows a table with columns: Date, Requester, ID, Category, Assignee, Start Time, End Time, and State.
- Task Logs:** A panel with the text 'No Logs to be shown'.
- Task Details:** A panel with the text 'Click on a task to view more information'.

A blue circular button with a white '+' sign is located in the bottom right corner of the dashboard.

감사합니다