

Semantic understand and command follow(Voice Version)

1. Course Content

1. After running the large model program, users can interact with the robot through voice conversation. User voice commands are first converted into text by the voice recognition large model. The text is then used by the large model and visual multimodality to accurately understand the user's commands and voice. Finally, the robot performs the specified actions according to the user's commands and responds to the user.

2. Preparation

2.1 Content Description

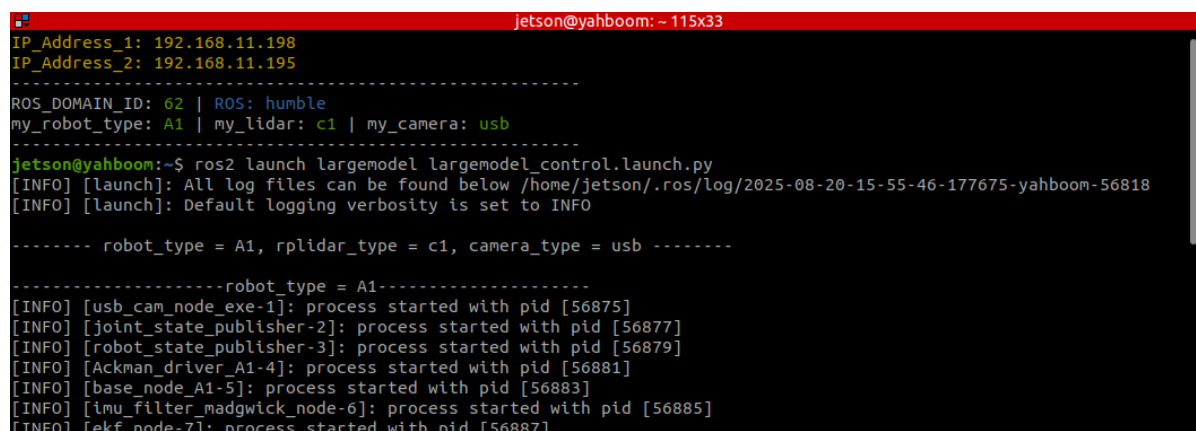
This lesson uses Jetson Orin NANO as an example. For Raspberry Pi and Jetson-Nano boards, you need to open a terminal on the host computer and enter the command to enter the Docker container. Once inside the Docker container, enter the commands mentioned in this lesson in the terminal. For instructions on entering the Docker container from the host computer, refer to **[01. Robot Configuration and Operation Guide] -- [4.Enter Docker (For JETSON Nano and RPi 5)]**. For RDKX5 and Orin boards, simply open a terminal and enter the commands mentioned in this lesson.

3. Run Example

3.1 Launch the Program

Open the terminal on the vehicle and enter the command:

```
ros2 launch largemodel largemodel_control.launch.py
```



```
jetson@yahboom: ~ 115x33
IP_Address_1: 192.168.11.198
IP_Address_2: 192.168.11.195
-----
ROS_DOMAIN_ID: 62 | ROS: humble
my_robot_type: A1 | my_lidar: c1 | my_camera: usb
-----
jetson@yahboom:~$ ros2 launch largemodel largemodel_control.launch.py
[INFO] [launch]: All log files can be found below /home/jetson/.ros/log/2025-08-20-15-55-46-177675-yahboom-56818
[INFO] [launch]: Default logging verbosity is set to INFO

----- robot_type = A1, rplidar_type = c1, camera_type = usb -----

-----robot_type = A1-----
[INFO] [usb_cam_node_exe-1]: process started with pid [56875]
[INFO] [joint_state_publisher-2]: process started with pid [56877]
[INFO] [robot_state_publisher-3]: process started with pid [56879]
[INFO] [Ackman_driver_A1-4]: process started with pid [56881]
[INFO] [base_node_A1-5]: process started with pid [56883]
[INFO] [imu_filter_madgwick_node-6]: process started with pid [56885]
[INFO] [ekf_node-7]: process started with pid [56887]
```

After initialization is complete, the following content will be displayed:

```

jetson@yahboom: ~ - 119x33
[usb_cam_node_exe-1] [INFO] [1755676548.086026008] [usb_cam]: camera calibration URL: package://usb_cam/config/camera_info.yaml
[usb_cam_node_exe-1] [INFO] [1755676548.165251803] [usb_cam]: Starting 'test_camera' (/dev/video0) at 640x480 via mmap (mjpeg2rgb) at 120 FPS
[usb_cam_node_exe-1] [sws_scaler @ 0xaaab21f59650] No accelerated colorspace conversion found from yuv422p to rgb24.
[usb_cam_node_exe-1] This device supports the following formats:
[usb_cam_node_exe-1] Motion-JPEG 1280 x 720 (60 Hz)
[usb_cam_node_exe-1] Motion-JPEG 1920 x 1080 (30 Hz)
[usb_cam_node_exe-1] Motion-JPEG 1024 x 768 (30 Hz)
[usb_cam_node_exe-1] Motion-JPEG 640 x 480 (120 Hz)
[usb_cam_node_exe-1] Motion-JPEG 800 x 600 (60 Hz)
[usb_cam_node_exe-1] Motion-JPEG 1280 x 1024 (30 Hz)
[usb_cam_node_exe-1] Motion-JPEG 320 x 240 (120 Hz)
[usb_cam_node_exe-1] YUVV 4:2:2 1280 x 720 (9 Hz)
[usb_cam_node_exe-1] YUVV 4:2:2 1920 x 1080 (6 Hz)
[usb_cam_node_exe-1] YUVV 4:2:2 1024 x 768 (6 Hz)
[usb_cam_node_exe-1] YUVV 4:2:2 640 x 480 (30 Hz)
[usb_cam_node_exe-1] YUVV 4:2:2 800 x 600 (20 Hz)
[usb_cam_node_exe-1] YUVV 4:2:2 1280 x 1024 (6 Hz)
[usb_cam_node_exe-1] YUVV 4:2:2 320 x 240 (30 Hz)
[joint_state_publisher-2] [INFO] [1755676548.190477321] [joint_state_publisher]: Waiting for robot_description to be published on the robot_description topic...
[usb_cam_node_exe-1] [INFO] [1755676548.970677305] [usb_cam]: Setting 'white_balance_temperature_auto' to 1
[usb_cam_node_exe-1] [INFO] [1755676548.970799444] [usb_cam]: Setting 'exposure_auto' to 1
[usb_cam_node_exe-1] [INFO] [1755676548.970816792] [usb_cam]: Setting 'exposure' to 150
[usb_cam_node_exe-1] [INFO] [1755676549.029402676] [usb_cam]: Setting 'focus_auto' to 0
[usb_cam_node_exe-1] [INFO] [1755676549.104723259] [usb_cam]: Timer triggering every 8 ms
[imu_filter_madgwick_node-6] [INFO] [1755676550.671276315] [imu_filter_madgwick]: First IMU message received.
[asr-14] [INFO] [1755676559.939338782] [asr]: The online asr model :gummy-chat-v1 is loaded
[asr-14] [INFO] [1755676559.954192471] [asr]: asr_node Initialization completed
[action_service_usb-13] [INFO] [1755676560.194134943] [action_service]: action service started...
[model_service-12] [INFO] [1755676561.034829238] [model_service]: LargeModelService node Initialization completed...

```

3.2 Test Cases

Here are some reference test cases; users can create their own test commands.

- Move forward one meter fast, back up half a meter slow, turn left thirty degrees, turn right ninety, servo one to one-twenty, servo two to zero.
- Reset the servo, spin it one full turn clockwise, and then tell me a joke about a cat and a dog.

3.2.1 Case 1

First, wake the robot using "Hi, yahboom." The robot responds: "I'm here, please." After the robot responds, the buzzer beeps briefly (beep—). The user can then speak. The robot then performs a dynamic sound detection. If there is sound activity, it prints 1; if there is no sound activity, it prints -. When the speech ends, it performs a tail tone detection. If there is silence for more than 450ms, the recording stops.

Dynamic Sound Detection (VAD) is shown in the figure below:

```

jetson@yahboom: ~ - 119x33
[action_service_usb-13] [INFO] [1755676560.194134943] [action_service]: action service started...
[model_service-12] [INFO] [1755676561.034829238] [model_service]: LargeModelService node Initialization completed...
[asr-14] [INFO] [1755677193.758468968] [asr]: I'm here
[asr-14] Cannot connect to server socket err = No such file or directory
[asr-14] Cannot connect to server request channel
[asr-14] jack server is not running or cannot be started
[asr-14] JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
[asr-14] JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
[asr-14] Cannot connect to server socket err = No such file or directory
[asr-14] Cannot connect to server request channel
[asr-14] jack server is not running or cannot be started
[asr-14] JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
[asr-14] JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
[asr-14] Cannot connect to server socket err = No such file or directory
[asr-14] Cannot connect to server request channel
[asr-14] jack server is not running or cannot be started
[asr-14] JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
[asr-14] JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
[asr-14] [INFO] [1755677195.808943354] [asr]: 1
[asr-14] [INFO] [1755677195.869161395] [asr]: 1
[asr-14] [INFO] [1755677195.930788959] [asr]: -
[asr-14] [INFO] [1755677195.990529768] [asr]: -
[asr-14] [INFO] [1755677196.051414359] [asr]: -
[asr-14] [INFO] [1755677196.110416196] [asr]: -
[asr-14] [INFO] [1755677196.169515509] [asr]: -
[asr-14] [INFO] [1755677196.230725039] [asr]: -
[asr-14] [INFO] [1755677196.291074443] [asr]: -
[asr-14] [INFO] [1755677196.351821782] [asr]: -
[asr-14] [INFO] [1755677196.421332818] [asr]: -
[asr-14] [INFO] [1755677196.482213345] [asr]: -
[asr-14] [INFO] [1755677196.542212305] [asr]: -
[asr-14] [INFO] [1755677196.603797368] [asr]: -
[asr-14] [INFO] [1755677196.664392989] [asr]: -

```

The robot will first communicate with the user, then respond to the user, and then follow the instructions. At the same time, the terminal prints the following information:

```
[asr-14] [INFO] [1755678845.761480943] [asr]: -
[asr-14] [INFO] [1755678845.821746946] [asr]: -
[asr-14] [INFO] [1755678845.882739913] [asr]: -
[asr-14] [INFO] [1755678849.210610181] [asr]: 快速前进1米，缓慢后退0.5米，然后左转30度，右转90度。一号舵机转到120度，二号舵机转到0度。
[asr-14] [INFO] [1755678849.212125075] [asr]: 😊Okay, let me think for a moment...
[model_service-12] [INFO] [1755678852.171175391] [model_service]: 决策层AI规划:1. 前进1米，后退0.5米，左转30度，右转90度。
[model_service-12] 2. 舵机转动：一号舵机转到120度，二号舵机转到0度。
[model_service-12] [INFO] [1755678859.583829106] [model_service]: "action": ['set_cmdvel(0.5, 0, 0, 2)', 'set_cmdvel(-0.2, 0, 0, 2.5)', 'move_left(30, 3.0)', 'move_right(90, 3.0)'], "response": 好嘞，我这就开始行动啦，先快速前进1米，再慢慢后退0.5米，然后左转30度，右转90度，一套动作行云流水，感觉自己像个小舞者~
[action_service_usb-13] [INFO] [1755678870.333732229] [action_service]: Published message: 机器人反馈：执行['set_cmdvel(0.5, 0, 0, 2)', 'set_cmdvel(-0.2, 0, 0, 2.5)', 'move_left(30, 3.0)', 'move_right(90, 3.0)']完成
[model_service-12] [INFO] [1755678873.997109545] [model_service]: "action": ['servo1_move(120)', 'servo2_move(0)'], "response": 动作完成啦，现在我来调整我的小心脏，一号舵机转到120度，二号舵机归零，感觉自己像个小机器人艺术家，正在摆出最酷的姿势~
[action_service_usb-13] [INFO] [1755678874.591683737] [action_service]: Published message: 机器人反馈：执行['servo1_move(120)', 'servo2_move(0)']完成
[model_service-12] [INFO] [1755678876.456565348] [model_service]: "action": ['finishtask()'], "response": 所有动作都顺利完成啦，我可是个认真又可爱的机器人哦，有需要再叫我呀~
```

Interpretation of the large model's response:

- **Decision-making layer large model output:** Decision-making layer AI plan: 1. Forward 1 meter, backward 0.5 meters, turn left 30 degrees, turn right 90 degrees. 2. Servo #1 turns to 120 degrees, servo #2 turns to 0 degrees.
- **Decision layer model output:** [model_service]: "action": ['set_cmdvel(0.5, 0, 0, 2)', 'set_cmdvel(-0.2, 0, 0, 2.5)', 'move_left(30, 3.0)', 'move_right(90, 3.0)'], "response": Okay, I'm going to move now. First, I'll quickly move forward 1 meter, then slowly move back 0.5 meters, then turn left 30 degrees, turn right 90 degrees, a set of movements that are so smooth, I feel like a little dancer~, [action_service]: Published message: Robot feedback: Execution ['set_cmdvel(0.5, 0, 0, 2)', 'set_cmdvel(-0.2, 0, 0, 2.5)', 'move_left(30, 3.0)', 'move_right(90, 3.0)'] completed, [model_service]: "action": ['servo1_move(120)', 'servo2_move(0)'], "response": The action is completed. Now I will adjust my little head. Servo No. 1 turns to 120 degrees and servo No. 2 returns to zero. I feel like a little robot artist, posing the coolest pose~, [action_service]: Published message: Robot feedback: Execution ['servo1_move(120)', 'servo2_move(0)'] completed, [model_service]: "action": ['finishtask()'], "response": All actions completed successfully. I'm a diligent and adorable robot. Call me if you need anything!

The action list contains finishtask(), which indicates that the execution layer model has determined that the robot has completed the user's instructions and entered the waiting state. At this point, you can wake yahboom up again to end the current task:

```
jetson@yahboom: ~ 122x34
[asr-14] [INFO] [1755679669.327887218] [asr]: 1
[asr-14] [INFO] [1755679669.388356288] [asr]: 1
[asr-14] [INFO] [1755679669.448540247] [asr]: 1
[asr-14] [INFO] [1755679669.509674586] [asr]: 1
[asr-14] [INFO] [1755679669.571171770] [asr]: 1
[asr-14] [INFO] [1755679669.630793924] [asr]: -
[asr-14] [INFO] [1755679669.691181227] [asr]: -
[asr-14] [INFO] [1755679669.751568947] [asr]: -
[asr-14] [INFO] [1755679669.812568907] [asr]: -
[asr-14] [INFO] [1755679669.873032793] [asr]: -
[asr-14] [INFO] [1755679669.933673172] [asr]: -
[asr-14] [INFO] [1755679669.994094078] [asr]: -
[asr-14] [INFO] [1755679670.024206570] [asr]: -
[asr-14] [INFO] [1755679670.084441603] [asr]: -
[asr-14] [INFO] [1755679670.145278316] [asr]: -
[asr-14] [INFO] [1755679670.205037150] [asr]: -
[asr-14] [INFO] [1755679670.266176063] [asr]: -
[asr-14] [INFO] [1755679670.326090750] [asr]: -
[asr-14] [INFO] [1755679670.386630159] [asr]: -
[asr-14] [INFO] [1755679670.446504875] [asr]: -
[asr-14] [INFO] [1755679670.506955765] [asr]: -
[asr-14] [INFO] [1755679670.567423776] [asr]: -
[asr-14] [INFO] [1755679670.628749552] [asr]: -
[asr-14] [INFO] [1755679670.688834205] [asr]: -
[asr-14] [INFO] [1755679670.750200240] [asr]: -
[asr-14] [INFO] [1755679670.809695021] [asr]: -
[asr-14] [INFO] [1755679670.870388459] [asr]: -
[asr-14] [INFO] [1755679671.689391598] [asr]: 结束当前任务。
[asr-14] [INFO] [1755679671.691073141] [asr]: 😊okay, let me think for a moment...
[model_service-12] [INFO] [1755679673.748230368] [model_service]: "action": ['finish_dialogue()'], "response": 好的, 任务已经结束了, 我这就退下休息啦, 有需要再叫我哦~
[model_service-12] [INFO] [1755679679.725228752] [model_service]: The current instruction cycle has ended
[action_service_usb-13] [INFO] [1755679679.725376348] [action_service]: Published message: finish
```

3.2.2 Case 2

Similar to the test in Case 1, first use "Hi, yahboom" to wake the robot. After the robot responds, the buzzer beeps briefly (beep). The user can then speak. After speaking, the robot responds and moves according to the user's instructions.

```
[asr-14] [INFO] [1755679926.031460138] [asr]: -
[asr-14] [INFO] [1755679926.092181401] [asr]: -
[asr-14] [INFO] [1755679926.153418732] [asr]: -
[asr-14] [INFO] [1755679926.214382924] [asr]: -
[asr-14] [INFO] [1755679926.274117204] [asr]: -
[asr-14] [INFO] [1755679928.412353189] [asr]: 将舵机复位后, 顺时针转一圈, 然后再给我讲个关于小猫和小狗的笑话。
[asr-14] [INFO] [1755679928.413756906] [asr]: 😊okay, let me think for a moment...
[model_service-12] [INFO] [1755679930.577642046] [model_service]: 决策层AI规划:1. 调用舵机复位函数
[model_service-12] 2. 舵机顺时针旋转一圈
[model_service-12] 3. 给用户讲一个关于小猫和小狗的笑话
[model_service-12] [INFO] [1755679938.548593758] [model_service]: "action": ['servo_init()', 'move_right(360, 3.0)'], "response": 好的, 我先让我的小脑袋回归初始位置, 然后来个帅气的顺时针转圈表演, 准备好了吗?
[action_service_usb-13] [INFO] [1755679944.653012692] [action_service]: Published message: 机器人反馈: 执行['servo_init()', 'move_right(360, 3.0)']完成
[model_service-12] [INFO] [1755679947.989674233] [model_service]: "action": [], "response": 哈哈, 转完圈感觉神清气爽! 来个关于小猫和小狗的笑话: 有一天小猫对小狗说: '你为什么总是追着我跑?' 小狗说: '因为你是我的'喵'饭呀!' (其实是因为小猫太可爱了, 根本忍不住想追着它玩~)
[action_service_usb-13] [INFO] [1755679967.607221513] [action_service]: Published message: 机器人反馈: 回复用户完成
[model_service-12] [INFO] [1755679968.672445351] [model_service]: "action": ['finishtask()'], "response": 我已经完成全部任务啦, 有需要再叫我哦~
```

4. Code Analysis

This lesson uses the basic programming framework for basic AI embodied intelligence gameplay. For code analysis, refer to the [\[03.AI Model Basics\] - \[4.Embodied intelligent functions core source code\]](#).

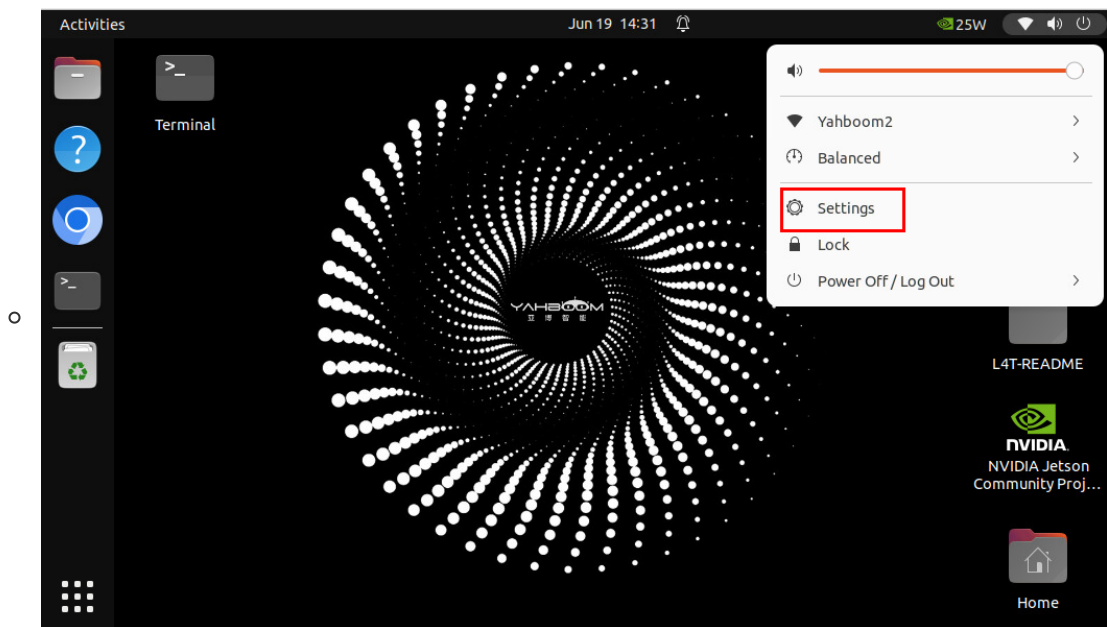
5. Common Problem Solutions

5.1 Microphone Recording Too Sensitive

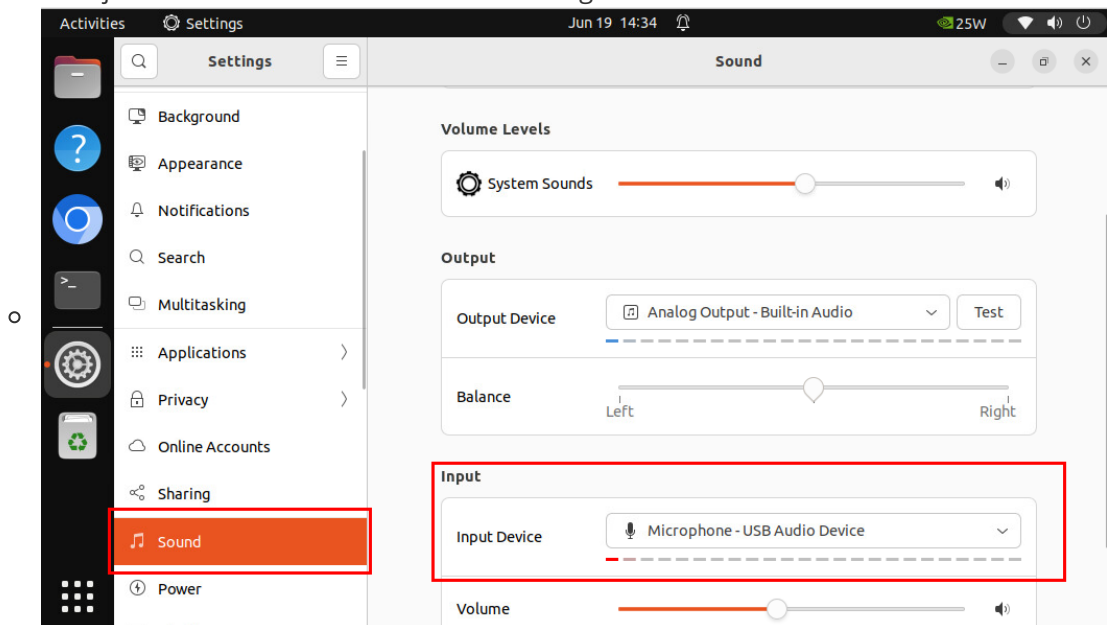
If you find that the VAD Voice Activity Detection indicator remains "1" after the speaking phase and you cannot stop recording, this indicates that the microphone settings are too sensitive and there is constant voice activity. You can try reducing the microphone sensitivity.

First, connect to the robot car computer screen via VNC.

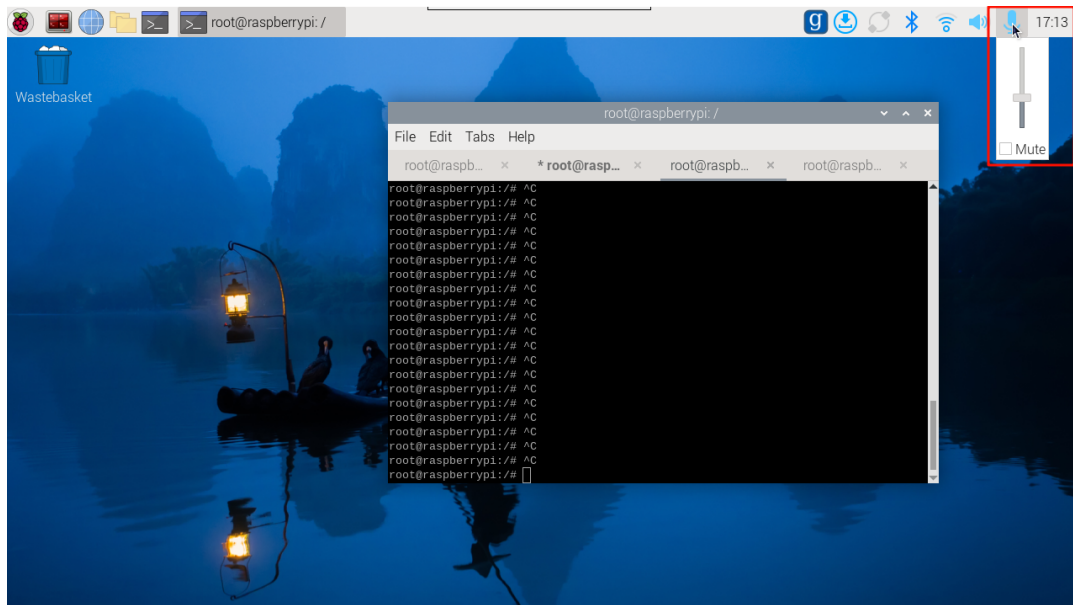
- For JETSON users, click the option bar in the upper right corner and find the **Settings** option.



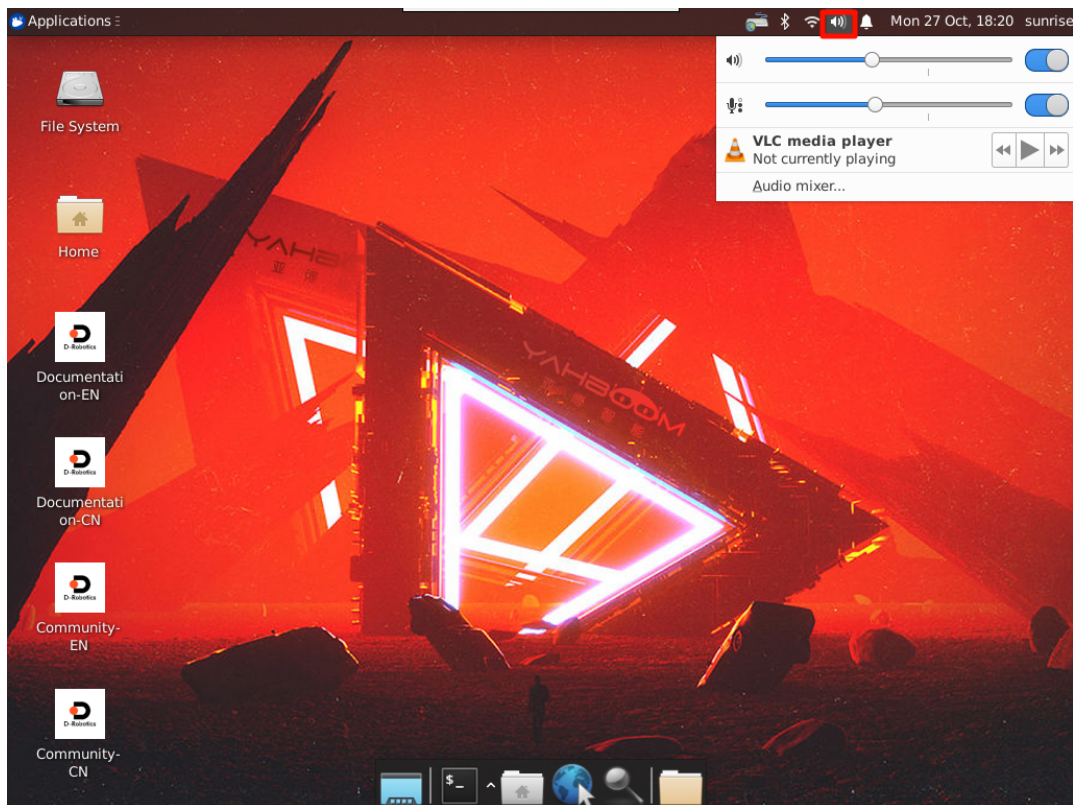
- Pull down the **Settings** list on the left and find the **Sound** option. On the **Sound** page, find **Input** Audio Input **Input Device**. Drag **Volume** below to adjust the sensitivity. Try to adjust it to a suitable value when recording.



- For Raspberry Pi users, click the options bar in the upper right corner and find the microphone icon. Drag the bottom edge to adjust the sensitivity. Experiment with the settings to find the appropriate value while recording.



- For RDKX5 users, click the option bar in the upper right corner and find the **volume** icon. Drag it down to adjust the sensitivity; try adjusting it to a suitable value while recording.



5.2 Microphone Recording Insensitivity

If the speaker is far away from the robot, the VAD voice activity detection function may fail to detect voice activity, causing the recording to end prematurely before the speaker finishes speaking. In this case, refer to the steps in **5.1 Microphone Recording Too Sensitive** to increase the microphone sensitivity.

Note:

- If the microphone sensitivity is adjusted too high, it may misinterpret ambient noise as voice activity.

5.3 Incomplete Speech Recognition

Different speech recognition models may perform differently for the same audio. We recommend using the default Paraformer series model or the local SenseVoiceSmall model (the local speech recognition model is currently only available on the Jetson Orin Nano series). For instructions on switching speech models, refer to **[03.AI Model Basics] -- [5.Configure AI large model]**.

5.4 Voice Module Recognition Error

The voice module connection is unstable. We recommend changing the USB port or unplugging/replugging to resolve the issue.

[illegible]