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

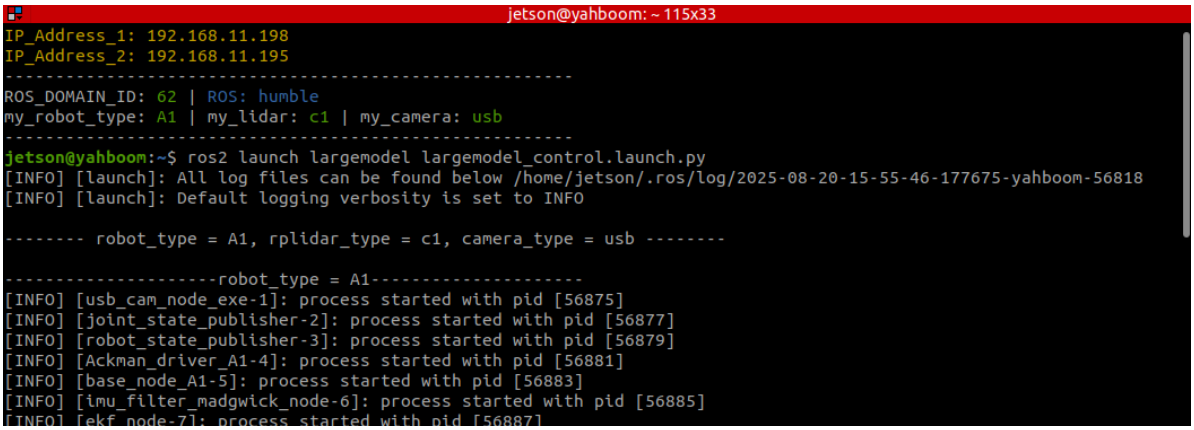
> This course uses the Jetson Orin NANO as an example. This is for users of the gimbal servo USB camera version. For Raspberry Pi and Jetson Nano boards, you need to open a terminal on the host computer and enter the command to enter the Docker container. Once inside the Docker container, enter the commands mentioned in this course in the terminal. For instructions on entering the Docker container on the host computer, refer to **[01. Robot Configuration and Operation Guide] -- [5.Enter Docker (For JETSON Nano and RPi 5)]**. For Orin boards, simply open a terminal and enter the commands mentioned in this course.

3. Run Example

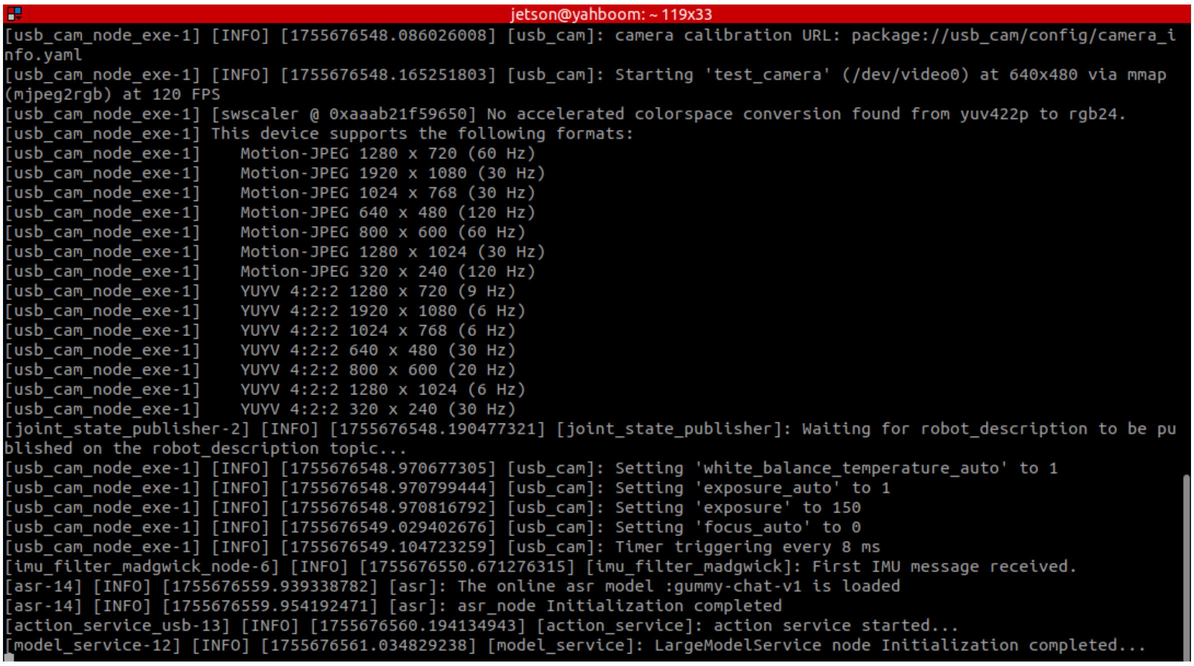
3.1 Launch the Program

Open the terminal on the vehicle and enter the command:

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



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



3.2 Test Cases

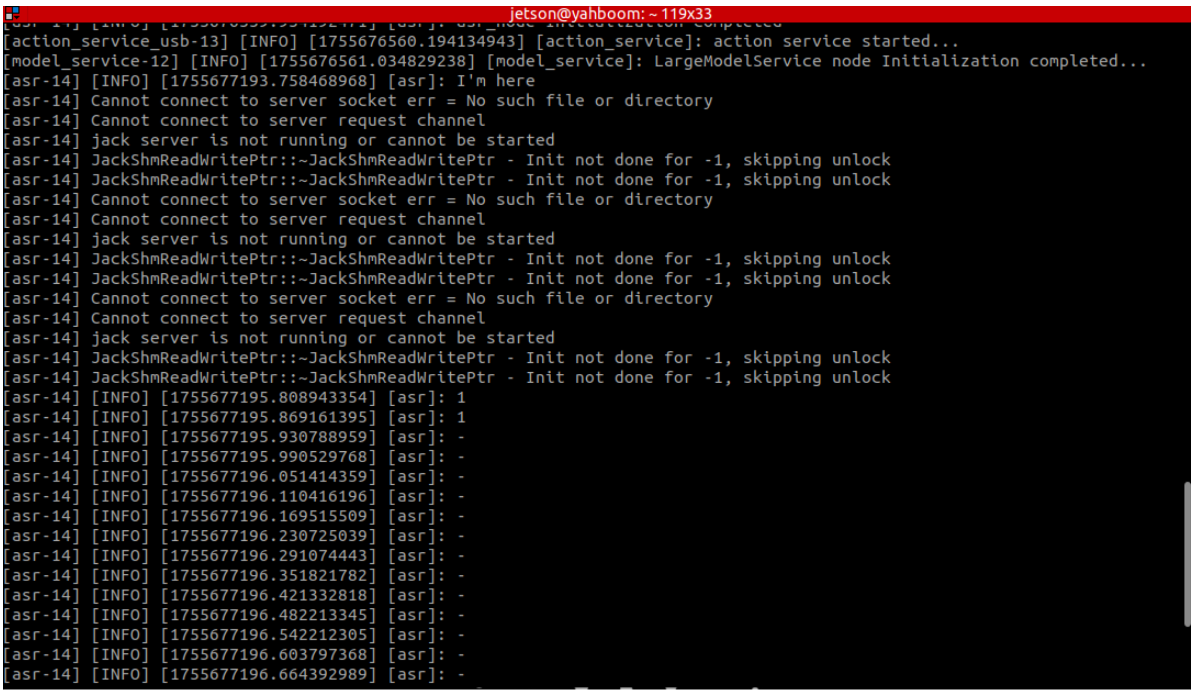
Here are some reference test cases; users can create their own test commands.

- 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:

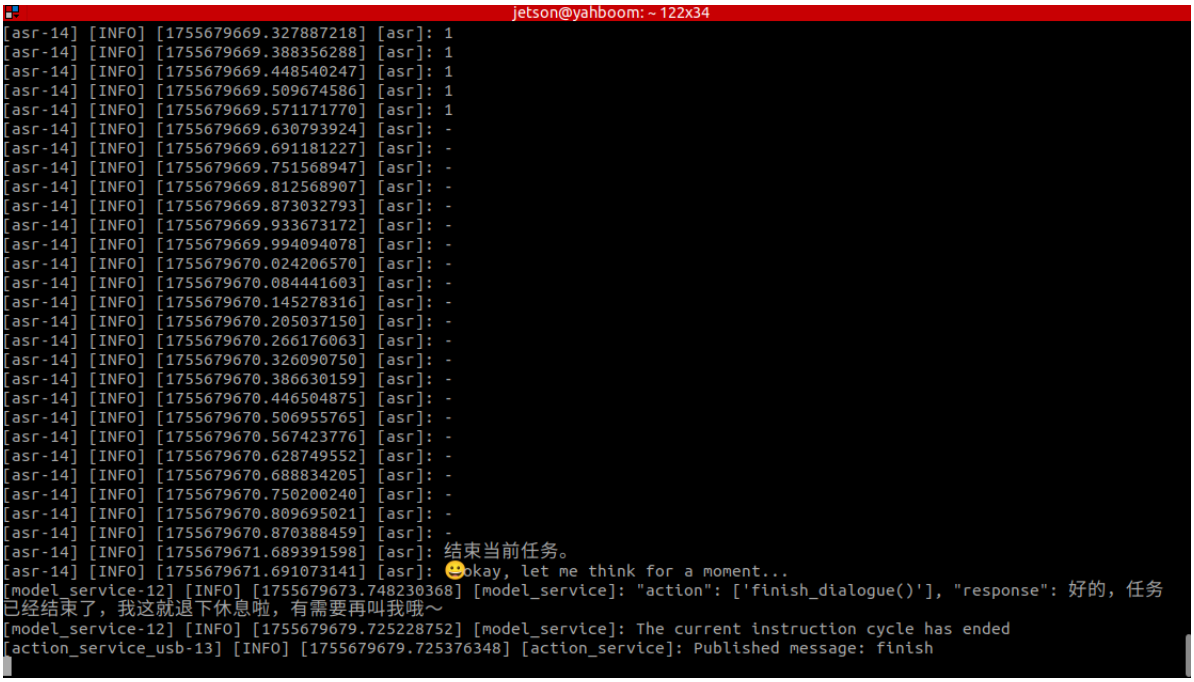


The robot will first communicate with the user, then respond to the user, and then follow the instructions. At the same time, the terminal prints the following information:

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:



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.

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]**.

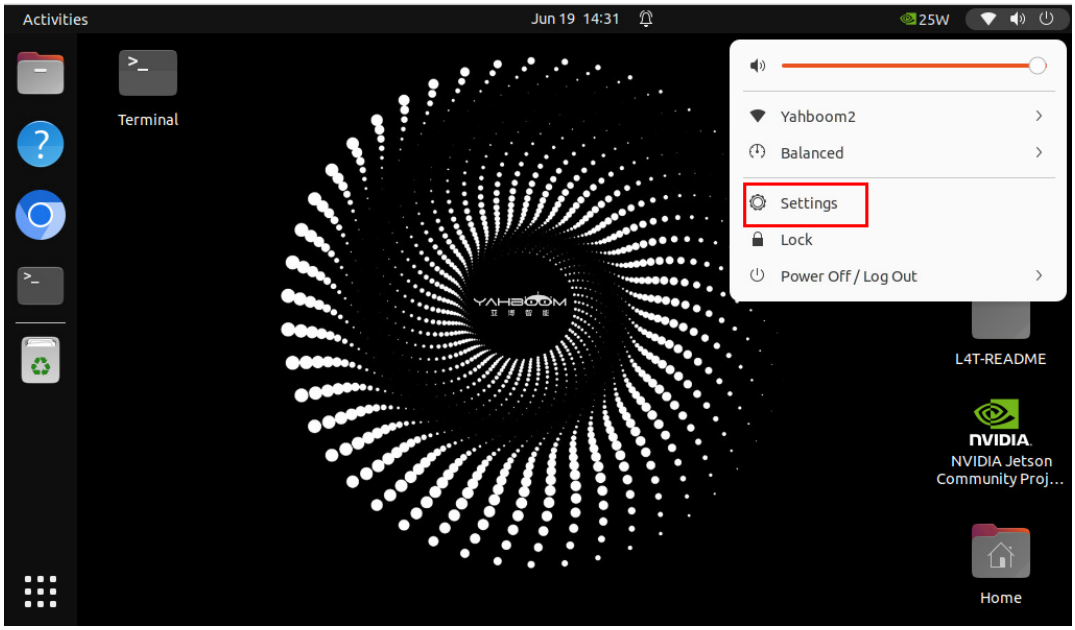
5. Common Problem Solutions

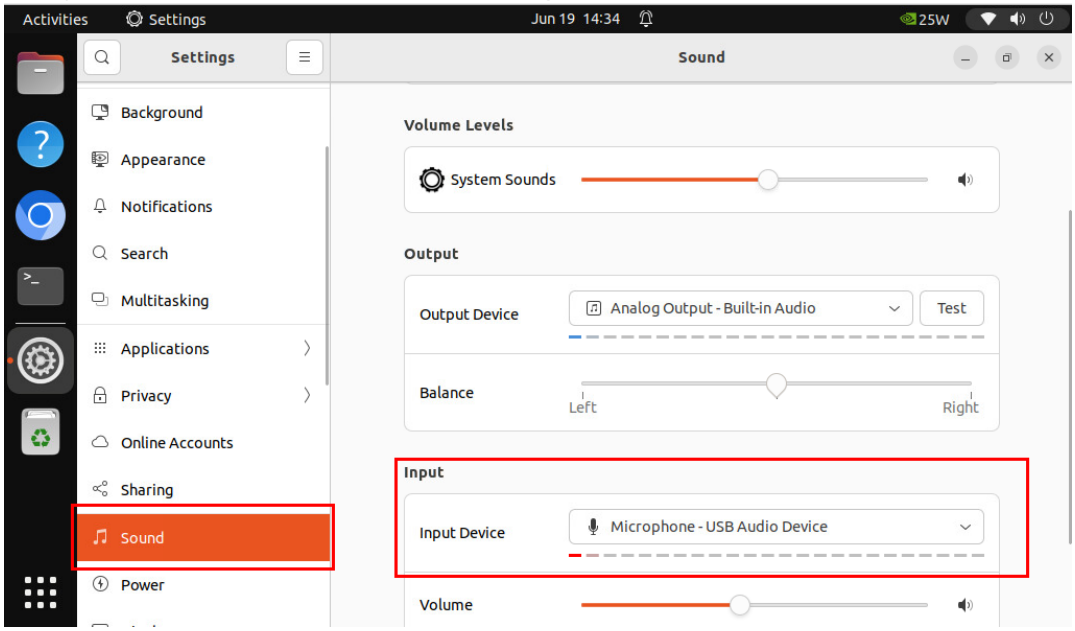
5.1 Microphone Recording Too Sensitive

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.

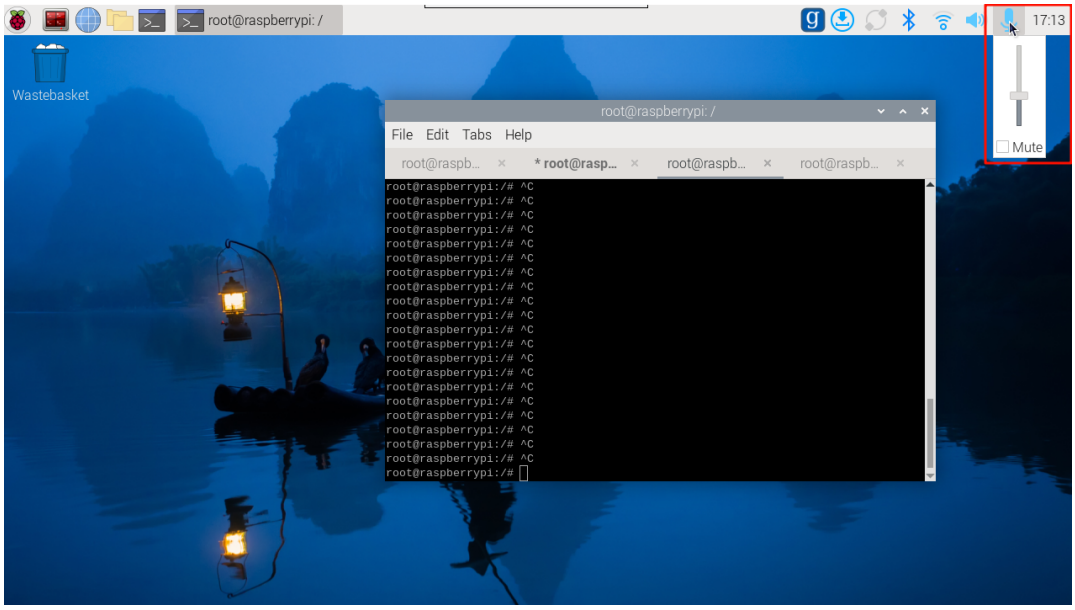
First, connect to the robot car computer screen via VNC.

- 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.



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.