

Multimodal visual understand + Depth Camera Distance Question Answering(Voice Version)

Multimodal visual understand + Depth Camera Distance Question Answering(Voice Version)

1. Course Content
2. Preparation
 - 2.1 Content Description
3. Running the Example
 - 3.1 Starting the Program
 - 3.2 Test Case
 - 3.2.1 Example 1: "Please tell me the distance between the fan in front of you and you."
4. Source Code Parsing
 - 4.1 Example 1


1. Course Content


1. Learn to use the robot's visual feedback to map distance to objects.
2. Analyze newly discovered key source code.


2. Preparation

2.1 Content Description

This lesson uses the Raspberry Pi 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.

 This example uses `model:"qwen/qwen2.5-vl-72b-instruct:free", "qwen-vl-latest"`

 The responses from the large model for the same test command may not be exactly the same and may differ slightly from the screenshots in the tutorial. To increase or decrease the diversity of the large model's responses, refer to the section on configuring the decision-making large model parameters in the **[03.AI Model Basics] -- [5.Configure AI large model]**.

 It is recommended that you first try the previous visual example. This example adds voice functionality to the singleton example. The functionality is largely the same, so I will not elaborate on the debugging details or the results.

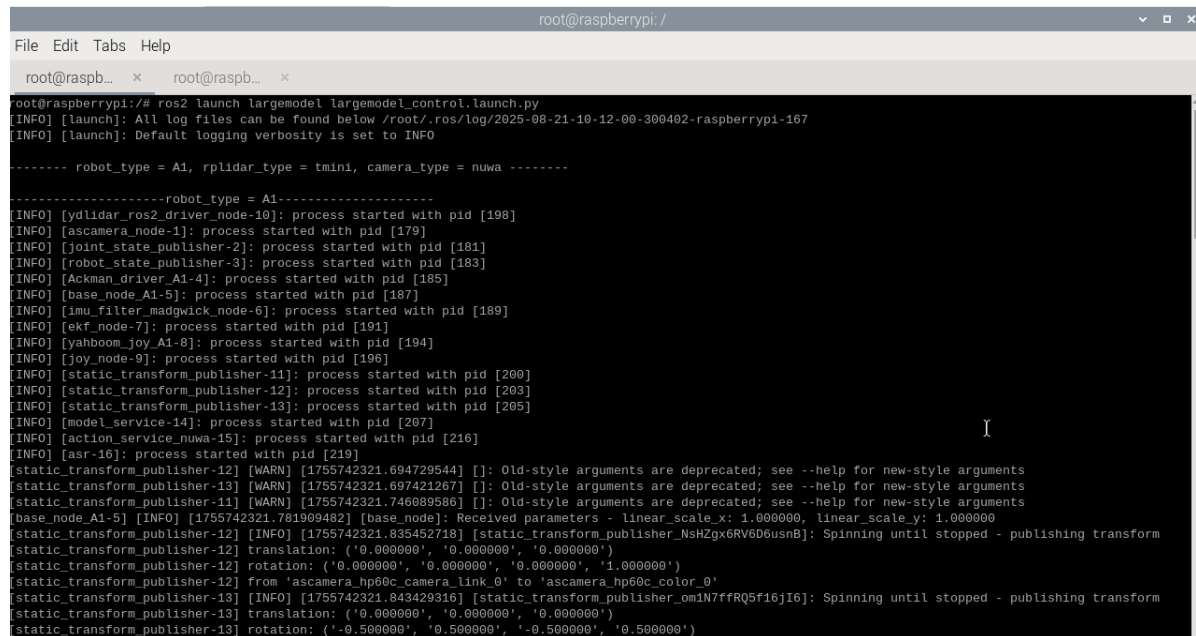
3. Running the Example

3.1 Starting the Program

For Raspberry Pi PI5 and jetson nano, you need to enter the Docker container first. For RDKX5 and Orin main controllers, this is not necessary.

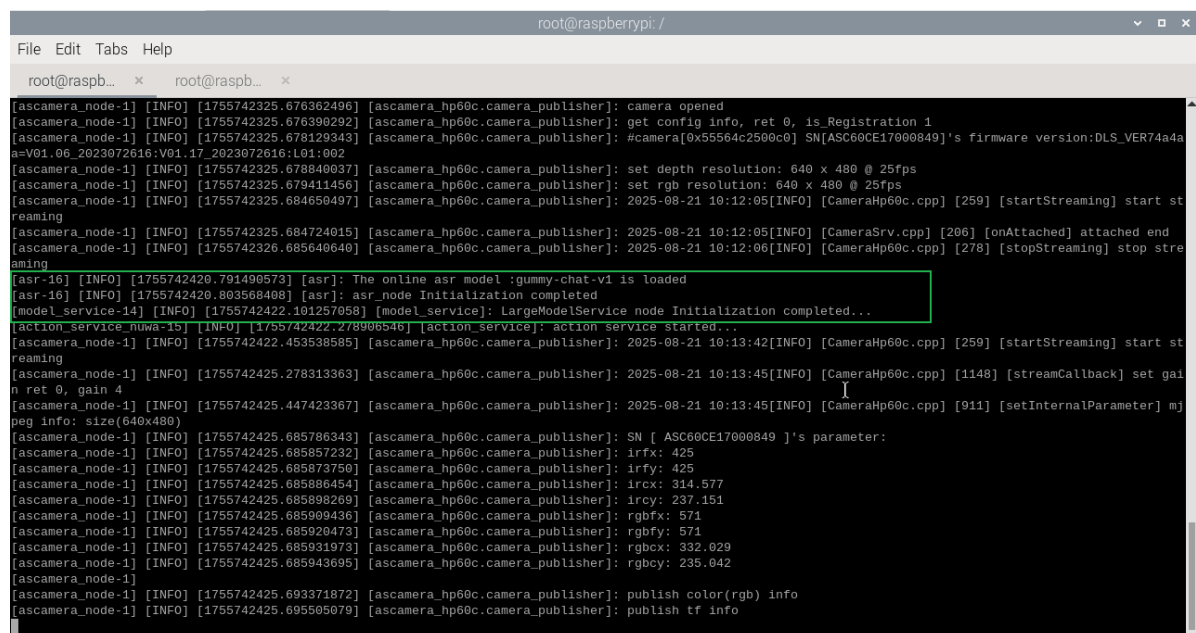
Open a terminal in Docker and enter the following command:

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



```
root@raspberrypi: /
File Edit Tabs Help
root@raspb... x root@raspb... x
root@raspberrypi: /# ros2 launch largemodel largemodel_control.launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2025-08-21-10-12-00-300402-raspberrypi-167
[INFO] [launch]: Default logging verbosity is set to INFO
----- robot_type = A1, rplidar_type = tmini, camera_type = nuwa -----
-----robot_type = A1-----
[INFO] [ydlidar_ros2_driver_node-10]: process started with pid [198]
[INFO] [ascamera_node-1]: process started with pid [179]
[INFO] [joint_state_publisher-2]: process started with pid [181]
[INFO] [robot_state_publisher-3]: process started with pid [183]
[INFO] [Ackman_driver_A1-4]: process started with pid [185]
[INFO] [base_node_A1-5]: process started with pid [187]
[INFO] [imu_filter_madgwick_node-6]: process started with pid [189]
[INFO] [ekf_node-7]: process started with pid [191]
[INFO] [yahboom_joy_A1-8]: process started with pid [194]
[INFO] [joy_node-9]: process started with pid [196]
[INFO] [static_transform_publisher-11]: process started with pid [200]
[INFO] [static_transform_publisher-12]: process started with pid [203]
[INFO] [static_transform_publisher-13]: process started with pid [205]
[INFO] [model_service-14]: process started with pid [207]
[INFO] [action_service_nuwa-15]: process started with pid [216]
[INFO] [asr-16]: process started with pid [219]
[static_transform_publisher-12] [WARN] [1755742321.694729544] [: Old-style arguments are deprecated; see --help for new-style arguments
[static_transform_publisher-13] [WARN] [1755742321.697421267] [: Old-style arguments are deprecated; see --help for new-style arguments
[static_transform_publisher-11] [WARN] [1755742321.746089586] [: Old-style arguments are deprecated; see --help for new-style arguments
[base_node_A1-5] [INFO] [1755742321.781909482] [base_node]: Received parameters - linear_scale_x: 1.000000, linear_scale_y: 1.000000
[static_transform_publisher-12] [INFO] [1755742321.835452718] [static_transform_publisher_NsHZGx6RV6D6usnB]: Spinning until stopped - publishing transform
[static_transform_publisher-12] translation: ('0.000000', '0.000000', '0.000000')
[static_transform_publisher-12] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-12] from 'ascamera_hp60c_camera_link_0' to 'ascamera_hp60c_color_0'
[static_transform_publisher-13] [INFO] [1755742321.843429316] [static_transform_publisher_om1N7ffRQ5f16jI6]: Spinning until stopped - publishing transform
[static_transform_publisher-13] translation: ('0.000000', '0.000000', '0.000000')
[static_transform_publisher-13] rotation: ('-0.500000', '0.500000', '-0.500000', '0.500000')
```

After initialization is complete, the following content will be displayed, and you can use voice control.



```
root@raspberrypi: /
File Edit Tabs Help
root@raspb... x root@raspb... x
[ascamera_node-1] [INFO] [1755742325.676362496] [ascamera_hp60c.camera_publisher]: camera opened
[ascamera_node-1] [INFO] [1755742325.676390292] [ascamera_hp60c.camera_publisher]: get config info, ret 0, is_Registration 1
[ascamera_node-1] [INFO] [1755742325.678129343] [ascamera_hp60c.camera_publisher]: #camera[0x55564c2500c9] SN[ASC60CE17000849]'s firmware version:DLS_VER74a4a
a=V01.06_2023072616:V01.17_2023072616:L01:002
[ascamera_node-1] [INFO] [1755742325.678840037] [ascamera_hp60c.camera_publisher]: set depth resolution: 640 x 480 @ 25fps
[ascamera_node-1] [INFO] [1755742325.679411456] [ascamera_hp60c.camera_publisher]: set rgb resolution: 640 x 480 @ 25fps
[ascamera_node-1] [INFO] [1755742325.684650497] [ascamera_hp60c.camera_publisher]: 2025-08-21 10:12:05[INFO] [CameraHp60c.cpp] [259] [startStreaming] start st
reaming
[ascamera_node-1] [INFO] [1755742325.684724015] [ascamera_hp60c.camera_publisher]: 2025-08-21 10:12:05[INFO] [CameraSrv.cpp] [206] [onAttached] attached end
[ascamera_node-1] [INFO] [1755742326.685640640] [ascamera_hp60c.camera_publisher]: 2025-08-21 10:12:06[INFO] [CameraHp60c.cpp] [278] [stopStreaming] stop stre
aming
[asr-16] [INFO] [1755742420.791490573] [asr]: The online asr model :gummy-chat-v1 is loaded
[asr-16] [INFO] [1755742420.803568408] [asr]: asr_node Initialization completed
[model_service-14] [INFO] [1755742422.101257058] [model_service]: LargeModelService node Initialization completed...
[action_service_nuwa-15] [INFO] [1755742422.278906546] [action_service]: action service started...
[ascamera_node-1] [INFO] [1755742422.453538585] [ascamera_hp60c.camera_publisher]: 2025-08-21 10:13:42[INFO] [CameraHp60c.cpp] [259] [startStreaming] start st
reaming
[ascamera_node-1] [INFO] [1755742425.278313363] [ascamera_hp60c.camera_publisher]: 2025-08-21 10:13:45[INFO] [CameraHp60c.cpp] [1148] [streamCallback] set gai
n ret 0, gain 4
[ascamera_node-1] [INFO] [1755742425.447423367] [ascamera_hp60c.camera_publisher]: 2025-08-21 10:13:45[INFO] [CameraHp60c.cpp] [911] [setInternalParameter] mj
peg info: size(640x480)
[ascamera_node-1] [INFO] [1755742425.685786343] [ascamera_hp60c.camera_publisher]: SN [ ASC60CE17000849 ]'s parameter:
[ascamera_node-1] [INFO] [1755742425.685857232] [ascamera_hp60c.camera_publisher]: irfx: 425
[ascamera_node-1] [INFO] [1755742425.685873750] [ascamera_hp60c.camera_publisher]: irfy: 425
[ascamera_node-1] [INFO] [1755742425.685886454] [ascamera_hp60c.camera_publisher]: ircx: 314.577
[ascamera_node-1] [INFO] [1755742425.685898269] [ascamera_hp60c.camera_publisher]: ircy: 237.151
[ascamera_node-1] [INFO] [1755742425.685909436] [ascamera_hp60c.camera_publisher]: rgbfx: 571
[ascamera_node-1] [INFO] [1755742425.685920473] [ascamera_hp60c.camera_publisher]: rgbfy: 571
[ascamera_node-1] [INFO] [1755742425.685931973] [ascamera_hp60c.camera_publisher]: rgbcx: 332.029
[ascamera_node-1] [INFO] [1755742425.685943695] [ascamera_hp60c.camera_publisher]: rgbcy: 235.042
[ascamera_node-1] [INFO] [1755742425.693371872] [ascamera_hp60c.camera_publisher]: publish color(rgb) info
[ascamera_node-1] [INFO] [1755742425.695505079] [ascamera_hp60c.camera_publisher]: publish tr info
```

3.2 Test Case

Here are two reference test cases. Users can create their own dialogue commands.

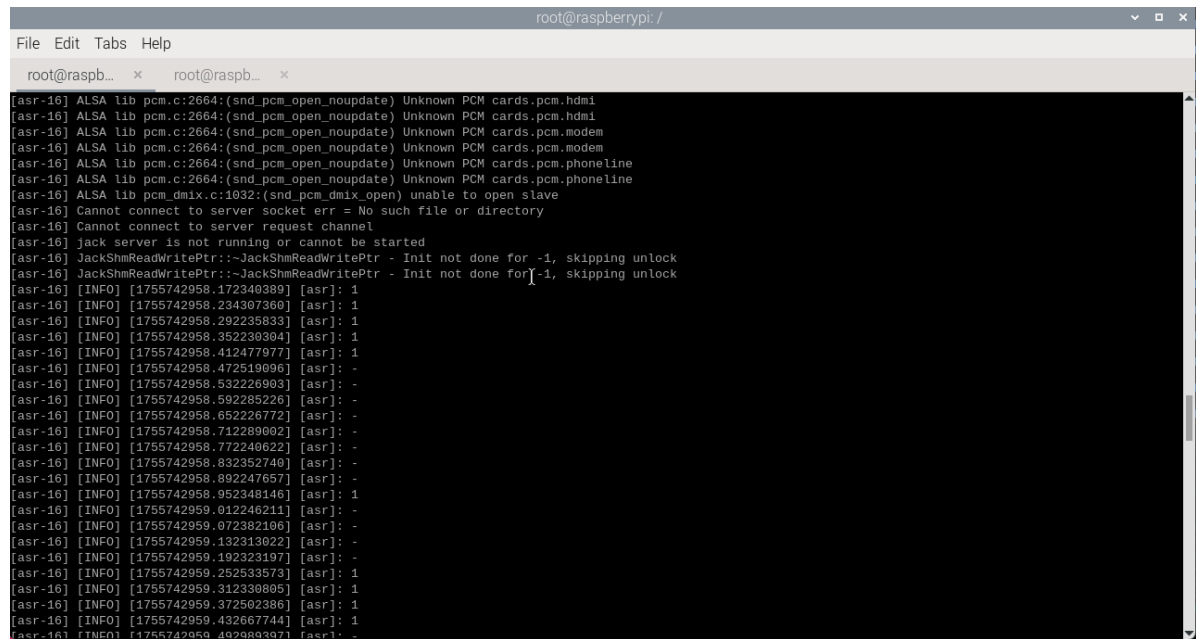
- Please tell me the distance between the object in front of you and you.

Example: Please tell me the distance between the fan in front of you and you.

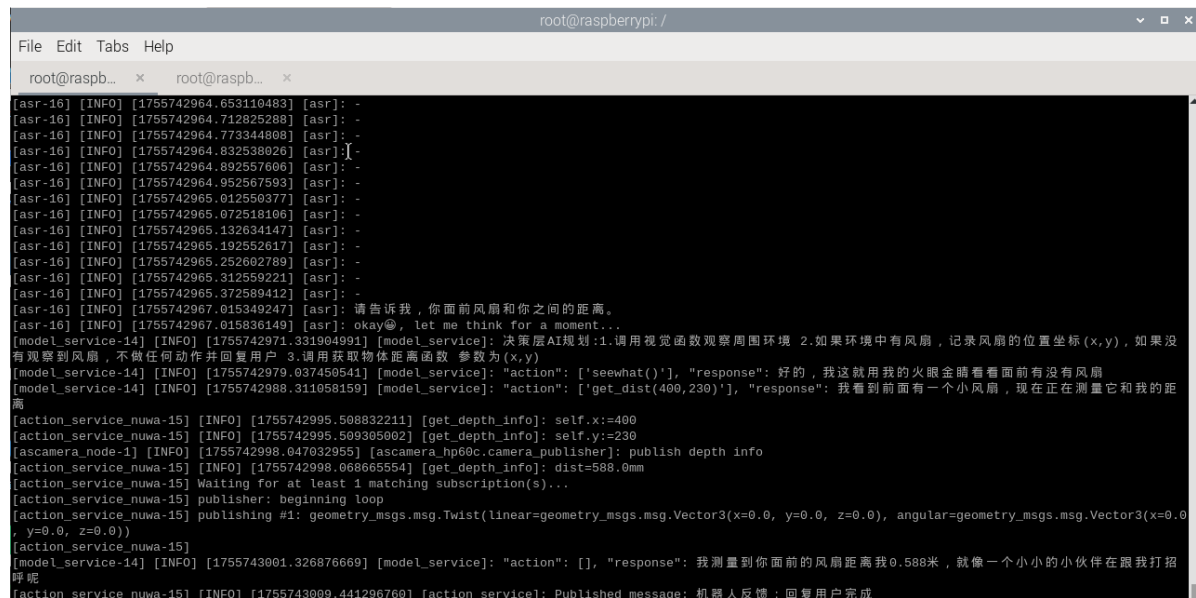
3.2.1 Example 1: "Please tell me the distance between the fan in front of you and you."

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

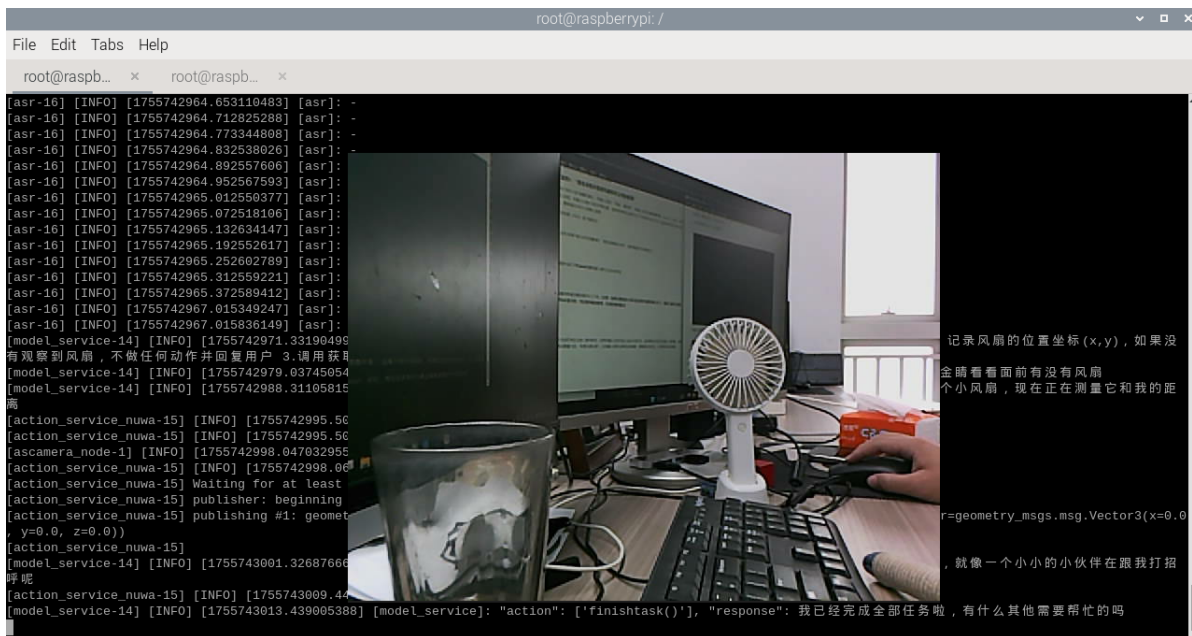
The following image shows Voice Active Detection (VAD):

A terminal window titled 'root@raspberrypi: /' showing a series of log messages. The logs start with ALSA PCM device initialization for 'pcm.c:2664:(snd_pcm_open_noupdate)' for various cards (hdm1, modem, phonenumber). This is followed by a 'Cannot connect to server socket err = No such file or directory' and 'Cannot connect to server request channel' error. Then, 'jack server is not running or cannot be started' is reported. The logs then show 'JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock' and a series of '[INFO] [1755742958.172340309] [asr]: 1' messages, indicating successful voice activity detection. The logs end with '[asr-16] [INFO] [1755742959.492989397] [asr]: -'.

The robot will first communicate with the user, then respond to the user's instructions. The following information will be printed on the terminal:

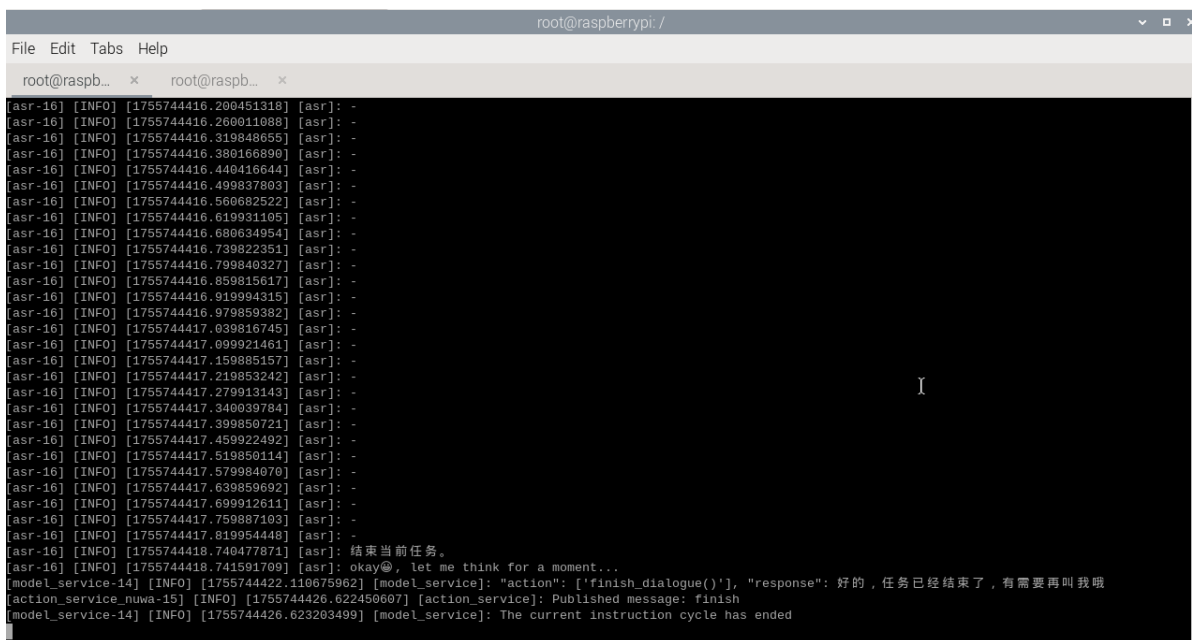
A terminal window titled 'root@raspberrypi: /' showing a series of log messages. The logs start with a series of '[INFO] [1755742964.653110483] [asr]: -' messages. This is followed by a series of '[INFO] [1755742965.012550377] [asr]: -' messages. Then, a '[INFO] [1755742967.015349247] [asr]: 请告诉我，你面前风扇和你之间的距离。' message is shown. This is followed by a '[INFO] [1755742967.015349247] [asr]: okay@, let me think for a moment...' message. Then, a '[model_service-14] [INFO] [1755742971.331904991] [model_service]: 决策层AI规划:1.调用视觉函数观察周围环境 2.如果环境中有风扇，记录风扇的位置坐标(x,y)，如果没有观察到风扇，不做任何动作并回复用户 3.调用获取物体距离函数 参数为(x,y)' message is shown. This is followed by a '[model_service-14] [INFO] [1755742979.037450541] [model_service]: "action": ['seewhat()'], "response": "好的，我就用我的火眼金睛看看面前有没有风扇' message. This is followed by a '[model_service-14] [INFO] [1755742988.311058159] [model_service]: "action": ['get_dist(400,230)'], "response": "我看到前面有一个小风扇，现在正在测量它和我的距离' message. Then, a series of '[action_service_nuwa-15] [INFO] [1755742995.508832211] [get_depth_info]: self.x:=400' and '[action_service_nuwa-15] [INFO] [1755742995.509305002] [get_depth_info]: self.y:=230' messages are shown. This is followed by a '[ascamera_node-1] [INFO] [1755742998.047032955] [ascamera_hp60c.camera_publisher]: publish depth info' message. Then, a '[action_service_nuwa-15] [INFO] [1755742998.068665554] [get_depth_info]: dist=588.0mm' message is shown. This is followed by a '[action_service_nuwa-15] Waiting for at least 1 matching subscription(s)...' message. Then, a '[action_service_nuwa-15] publisher: beginning loop' message is shown. This is followed by a '[action_service_nuwa-15] publishing #1: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0))' message. This is followed by a '[model_service-14] [INFO] [1755743001.326876669] [model_service]: "action": [], "response": "我测量到你面前的风扇距离我0.588米，就像一个小小的小伙伴在跟我打招呼呢' message. Finally, a '[action_service_nuwa-15] [INFO] [1755743009.441296760] [action_service]: Published message: 机器人反馈：回复用户完成' message is shown.

A window titled **frame** will open on the VNC screen and automatically close after 4 seconds.



As you can see, the terminal response distance is 0.588 meters. **(Note: If the feedback is 0.0 meters, it may be because the object being recognized is too small, resulting in inaccurate center coordinates from the large visual model. This can be resolved by replacing the visual model and the recognized object.)**

After completing a task, the robot enters a waiting state, during which it reenters the free conversation state, but all conversation history is retained. At this point, you can wake yahboom up again and select "End Current Task" to terminate the current task cycle, clear the conversation history, and start a new one.



4. Source Code Parsing

Source code located at:

Jetson Orin Nano, Jetson Orin NX Host:


```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_voice_ctrl_depth/yahboomcar_voice_ctrl_depth/voice_get_dist.py
```

```
def get_dist(self,x,y):
    self.get_dist_future = Future() #Reset Future object
    x1 = int(x)
    y1 = int(y)
    process_1 = subprocess.Popen(['ros2', 'run', 'yahboomcar_voice_ctrl_depth',
    'voice_get_dist', '--ros-args', '-p', f'x:={x1}', '-p', f'y:={y1}'])
    while not self.get_dist_future.done():
        if self.interrupt_flag:
            break
        time.sleep(0.1)

    self.kill_process_tree(process_1.pid)
    self.cancel()
```

Wait for the callback function to feedback the distance and send the future.done signal. Then the `while not self.get_dist_future.done()` in the **dist_callback** function will exit the blocking state. Then the **kill_process_tree** method will be called to recursively kill the process tree of the child process. Finally, the status of the execution action will be fed back to the execution layer model.

