

Multimodal Large Model+robotic arm handling (Voice Version)

Before running the function, you need to close the App and large programs. For the closing method, refer to [4. Preparation] - [1. Manage APP control services].

1. Function Description

After the program runs, wake up the voice module and input color block handling commands by voice. The large model will plan the actions to complete the color block handling task, then the program will control the robotic arm to handle the color blocks in sequence.

2. Startup

Users with Jetson-Nano mainboard version need to enter the docker container first and then input the following command. Users with Orin mainboard can directly open the terminal and input the following command:

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

After waking up the module, input the command by voice:

```
Put the green block on top of the red block
```

The block can be placed in one of the five positions: front, back, left, right, or top.

The program will first take a photo to get the positions of the green block and red block; then it will control the robotic arm to grip the green block and place it on top of the red block.

3. Task Planning

1. Call `seewhat()` to observe the environment and get the positions of the green block and red block on the desktop.
2. Call `change_pose(x1, y1, x2, y2, x3, y3, x4, y4, 'green', 'red', 5)` function to put the green block on top of the red block. Where `(x1, y1, x2, y2)` are the top outer bounding box coordinates of the green block, `(x3, y3, x4, y4)` are the top outer bounding box coordinates of the red block, `5` represents top.

4. Core Code Analysis

You can refer to the content in **4. Core Code Analysis** from tutorial [17. AI Model - Text Version] - [Multimodal Large Model+robotic arm handling]. The voice version and text version have the same action functions, only the task command input method is different.

5. Advanced Gameplay

We can have the robotic arm complete a more complex task: remember the current position of the color block, then place the color block on another block, and finally put this block back to its original position. After waking up the module, input by voice:

Remember the current position of the blue block, then put the blue block on top of the green block, and finally put the blue block back to its original position

The program will first take a photo to find the blue block and calculate its current position, then take another photo to find the blue block and green block, then control the robotic arm to put the blue block on top of the green block. Then grip the blue block and finally put the blue block back to its original position.

5.1. Task Planning

1. Call `seewhat()` to observe the environment;
2. Call `compute_pose(x1, y1, x2, y2)` function to record the current position of the blue block;
3. Call `seewhat()` to observe the environment and check the positions of the blue block and green block;
4. Call `change_pose(x1, y1, x2, y2, x3, y3, x4, y4, 'blue', 'green', side)` function to put the blue block on top of the green block;
5. Call `grasp_from_down_list('blue')` to find the current position of the blue block in the placement list;
6. Call `return_to_orin('blue')` function to put the blue block back to its original position

5.2. Core Code Analysis

You can refer to the content in **5.2. Core Code Analysis** from tutorial [17. AI Model - Text Version] - [Multimodal Large Model+robotic arm handling]. The voice version and text version have the same action functions, only the task command input method is different.