

5.Voice control robotic arm garbage sorting

5.1 Function description

Through the interaction with the voice module, rosmaster can identify and classify the garbage pictures on the squares and broadcast the results of the recognition and classification; then the robotic arm will grab the recognized squares, navigate to the set area autonomously, and put it down. , and finally return to the set origin.

5.2 Start

Note: Due to the processing speed of the NANO motherboard, when loading the model data, the image processing will be a little stuck, but it can still run successfully, and the NX motherboard will be better.

5.2.1、 ROS package path

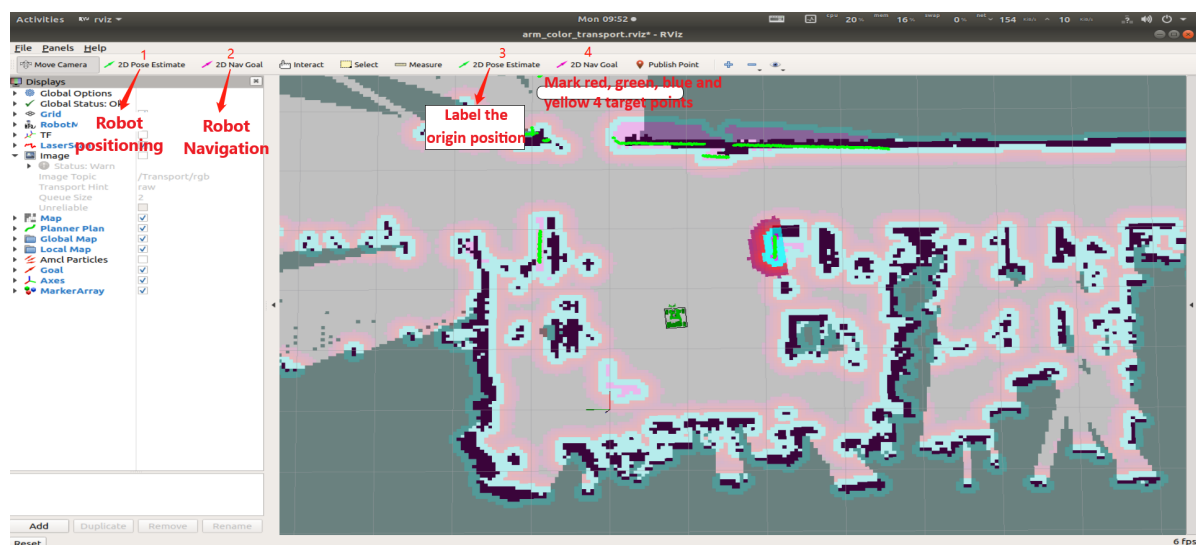
```
~/yahboomcar_ws/src/garbage_identify_yolov5/
```

5.2.2 Start

Note: Multi-machine communication needs to be implemented between rosmaster X3Plus and the virtual machine , you can view the " **06.Linux operating system\4. Multi-machine communication configuration** " tutorial for configuration

```
#rosmaster X3Plus running
roslaunch yahboomcar_voice_ctrl voice_transport_base.launch
python3 ~/yahboomcar_ws/src/garbage_identify_yolov5/garbage_identify_yolov5.py
# virtual machine running
roslaunch arm_color_transport transport_rviz.launch
```

Each tool annotation on the virtual machine rviz is shown in the following figure,



- Step 1

In rviz, use the 1 tool to calibrate and position the rosmaster

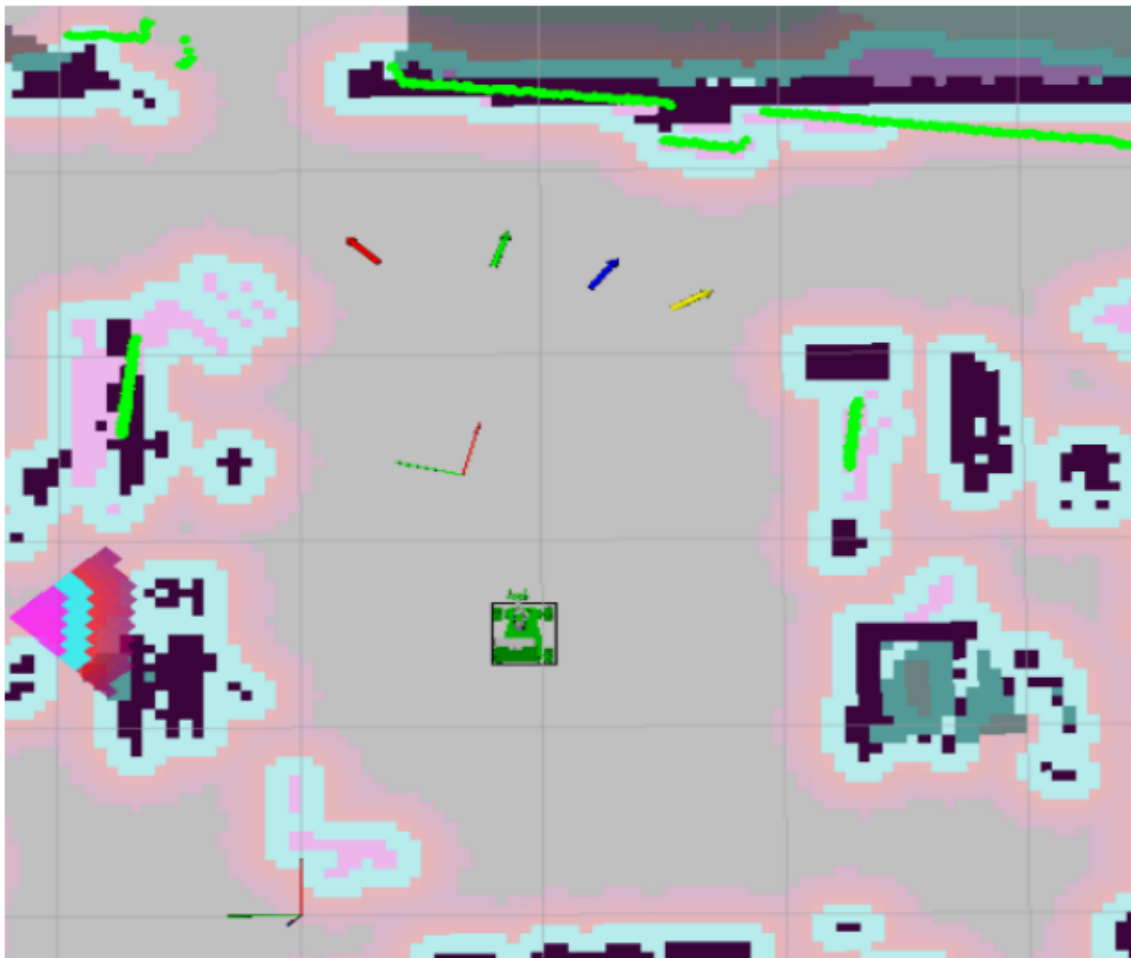
- Step 2

In rviz, use 3 tools to label the start position for rosmaster, the start position is the position that the car will automatically return to after putting down the color blocks

- Step 3

In rviz, use 4 tools to mark the red, green, blue and yellow destinations on the map in turn. After the rosmaster's robotic arm grips the color block, it will navigate to the destination corresponding to the color block.

Here it is best to put four The points are marked in a row, and the distance between the front and rear cannot be too far apart. Otherwise, when planning the path, you may hit the color blocks.



Note: **toxic waste** corresponds to **red** marker point; **wet waste** corresponds to **green** marker point; **recyclable** corresponds to **blue** marker point; **point D** corresponds to **dry waste** marker point;

- Step 4

Say "Hi Yahboom" to Rosmaster, wake up the voice module, place the color block about 20cm away from the camera on the robotic arm, wait for about 8s and other objects to appear steadily on the screen, and say "what garbage garbage' to Rosmaster is this?", it will answer the recognized garbage name and type, and after the second "di", hand the block to the gripper of the robotic arm, which will clamp the block.

After the Rosmaster backs up and adjusts its attitude, it will autonomously navigate to the previously set location. After arriving, it will lower the claw and put the block down, and the module will say "It has been in its place", then, after adjusting the position backward, navigate back to the set origin, and wait for the next recognition.

Note: After running the program, this screen will appear. This is because the program is loading the model data and it will take a while. When the screen becomes colored, it means that the model is loaded successfully.



5.2.3 Core code: garbage_identify_yolov5.py

- code path

```
~/yahboomcar_ws/src/garbage_identify_yolov5
```

- Core code analysis

1) import the corresponding library file

```
from garbage_identify import garbage_identify    #Garbage Recognition Library
from Speech_Lib import Speech    #speech Recognition Library
from garbage_library import GarbageTransport    #Transport Navigation
Library
```

2) create an object

```
self.target = garbage_identify()    #Create garbage identification object
spe = Speech()    #Create speech recognition object
self.garbage_transbot = GarbageTransport()    #Create Transport navigation
object
```

3) recognition library: garbage_identify.py

```
Model path: /home/jetson/software/yolov5-5.0/model0.pt
important function:
garbage_run (img): Execute the garbage identification function, pass in the
original image, return the recognized image, and identify the name
get_pos ():Get identification information, return name
```

The program terminal will also print the recognition result. When garbage is recognized, the recognized screen will be stuck, because other programs are executed at this time, and the program will be blocked. After other programs are executed, it will return to normal Recognized screen.

4) `garbage_identify_yolov5.py` important function: `process`

In this function, it is mainly to judge the return result obtained by

`self.target.garbage_run(img)`, which is the name of the garbage, and then enter the `self.garbage_transbot.process()` function, which is a function with Parameter, the meaning of the parameter is to go to the point marked on RVIZ.

5) `garbage_transbot.process()` function

After entering this function, the current mode state will be judged first. If it is the default Grip state, the incoming parameter value will be judged.

1 means red mark point, 2 means green mark point, 3 means blue mark point, 4 means yellow mark point.

For other states, you can view the source code of the program. It can be seen that each time the program in this state is executed, the mode state will be changed to the next state that is expected to be executed.

5.3 program flow chart

