12. Mediapipe arm attitude control robotic arm

12.1. Use

After the program is started, the camera captures the image and lifts the right arm. After the buzzer is not heard, the robotic arm will synchronously imitate the movement of the forearm (straighten, bend and close the palm).

Note: [R2] on the remote control handle has the [pause/start] function for this gameplay.

Function package path: ~/transbot_ws/src/arm_mediapipe/

jetson motherboard/Raspberry Pi 4B

Robot side

```
roslaunch arm_mediapipe mediaArm.launch # robot
```

Virtual machine side (recommended)

```
rosrun arm_mediapipe PoseArm.py
```

Raspberry Pi 5

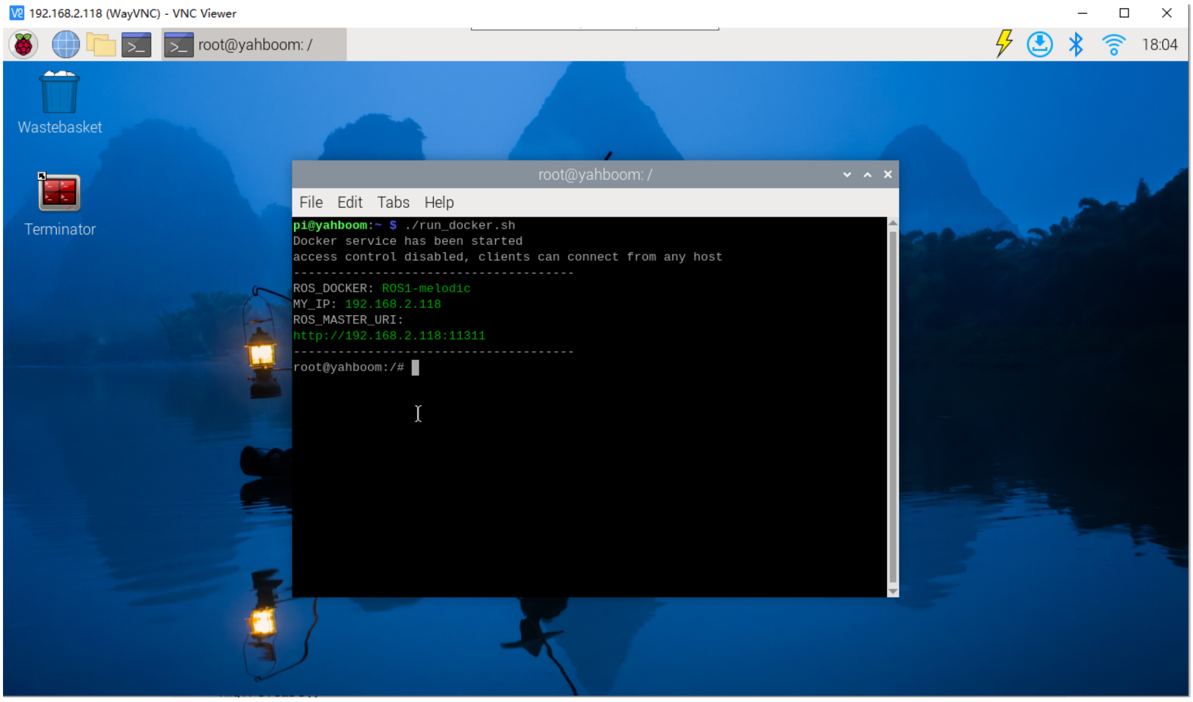
Before running, please confirm that the large program has been permanently closed

Enter docker

Note: If there is a terminal that automatically starts docker, or there is a docker terminal that has been opened, you can directly enter the docker terminal to run the command, and there is no need to manually start docker

Start docker manually

```
./run_docker.sh
```



```
pi@yahboom:~ $ ./run_docker.sh
Docker service has been started
access control disabled, clients can connect from any host
--------------------------------------
ROS_DOCKER: ROS1-melodic
MY_IP: 192.168.2.118
ROS_MASTER_URI:
http://192.168.2.118:11311
--------------------------------------
root@yahboom:/# 
```

```
roslaunch arm_mediapipe mediaArm.launch
```

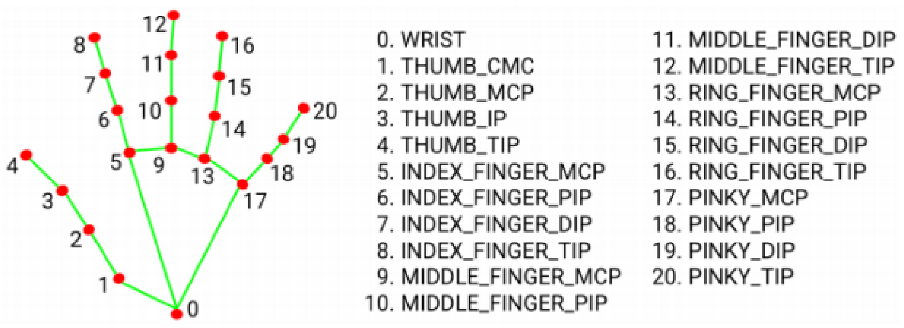
Virtual machine side (recommended)

```
rosrun arm_mediapipe PoseArm.py
```

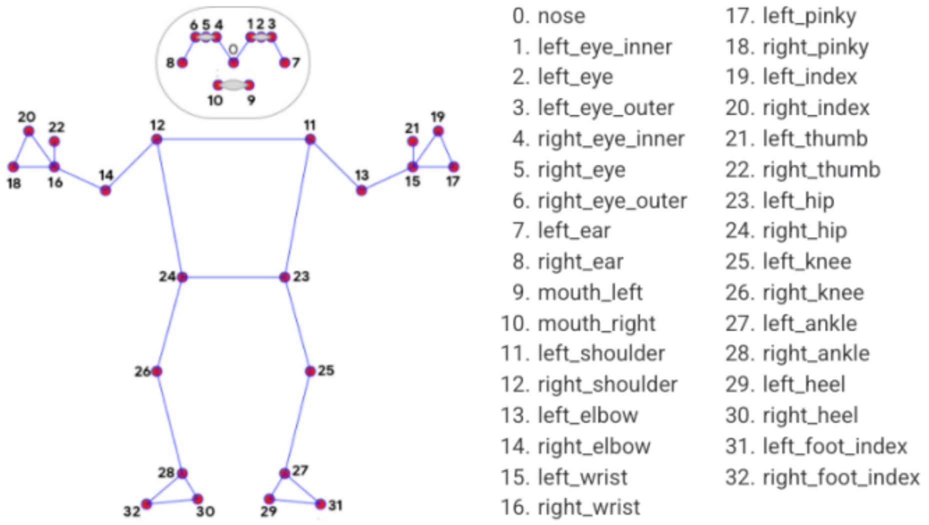
After the program is started, press the R2 button on the handle to turn on the function. Stand in front of the camera so that the entire arm appears in the screen. At this time, the buzzer will sound. This is written in the program to prevent the robotic arm from returning after the arm is put down. Follow the movement, causing confusion in the program. We raise our elbows and wait for the buzzer to stop sounding, and the robotic arm will simulate the movement of the forearm. You can straighten or bend your wrist and open/close your palm to control the robotic arm. Remember not to move the forearm too much.

12.2. Principle of realization

MediaPipe Hands infers 3D coordinates of 21 hand-valued joints from one frame



The landmark model in MediaPipe Pose predicts the location of 33 pose coordinates (see figure below).



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

In this program, all we need is the coordinates of the part of the right arm. By calculating the angle formed by these coordinates, we can calculate how much angle each steering gear of the robotic arm needs to turn.

12.3.Core code analysis PoseArm.py

Code reference path： ~/transbot_ws/src/arm_mediapipe/scripts

- Import critical libraries

```
from media_library import *
```

- Get the information of each joint of the right arm

```
frame, pointArray, lhandptArray, rhandptArray =
self.pose_detector.findHolistic(frame)
#This pose_detector.findHolistic function prototype is in media_library.py
```

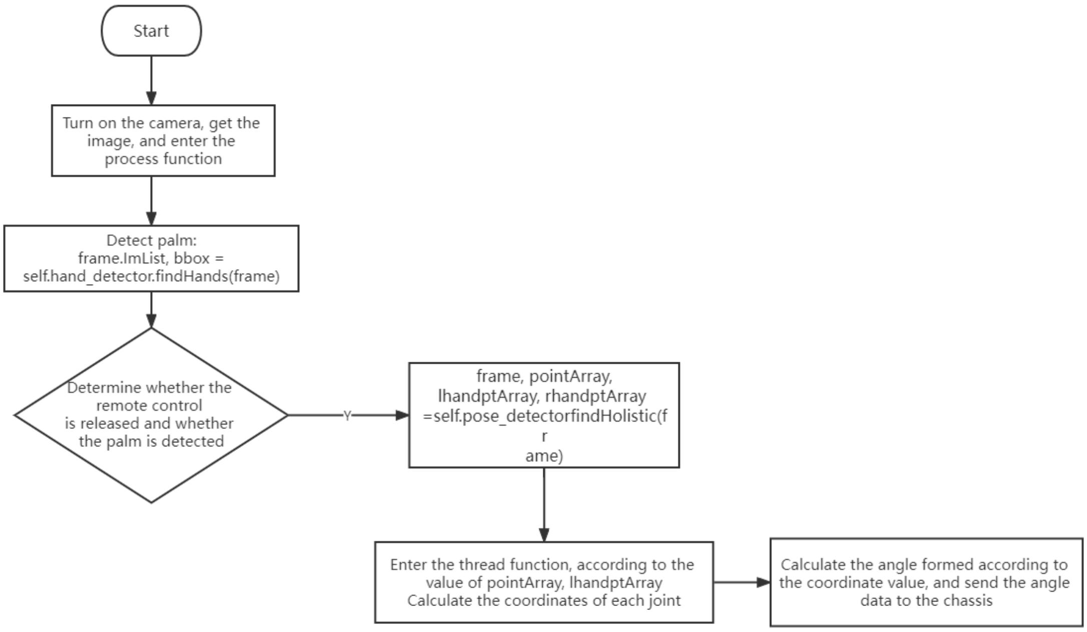
- Here is an explanation of how the angles formed by the elbow joint, wrist joint, palm opening and closing are calculated:

  1）、As can be seen from the above figure, the three points 13-0 (palm wrist joint)-8 (top of index finger) form the wrist joint. Therefore, the coordinate values of these three points can be obtained to calculate the angle of the wrist joint;

  2）、As can be seen from the above figure, the three points of 4 (the top of the thumb)-0 (the palm wrist joint)-8 (the top of the index finger) form the angle of the gripper. Therefore, the coordinate values of these three points can be obtained to calculate the gripper angle at which the claw needs to be clamped;

  The above-mentioned 4 key joint codes are specifically realized,you can look at the program source code.

12.4. Program flow chart


Start
Turn on the camera, get the image, and enter the process function
Detect palm: frame.ImList, bbox = self.hand_detector.findHands(frame)
Determine whether the remote control is released and whether the palm is detected
Y
frame, pointArray, lhandptArray, rhandptArray =self.pose_detectorfindHolistic(fr ame)
Enter the thread function, according to the value of pointArray, lhandptArray Calculate the coordinates of each joint
Calculate the angle formed according to the coordinate value, and send the angle data to the chassis