

# Gesture control of the car's basic movements

## 1. Program Functionality

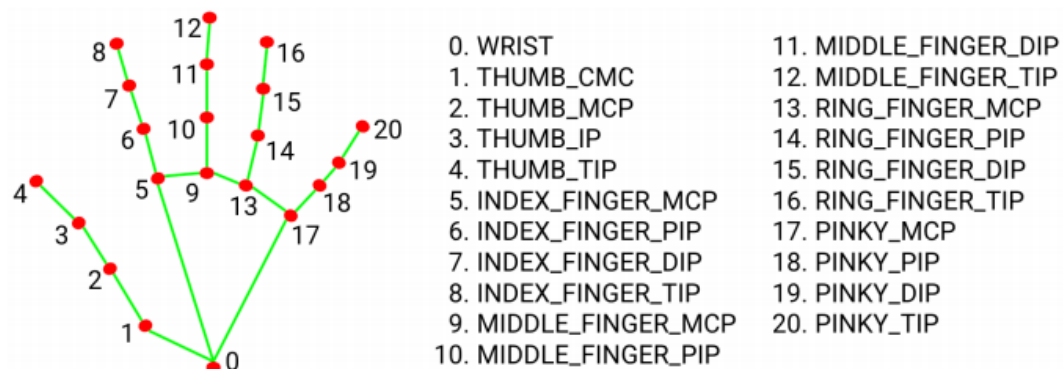
For Raspberry Pi and Jetson-Nano boards, open a terminal on the host computer and enter the command to enter the Docker container. Once inside the Docker container, enter the commands mentioned in this lesson in the terminal. For instructions on entering the Docker container from the host computer, refer to the section [Entering the Robot Dock (Jetson-Nano and Raspberry Pi 5 users, see here)] in [01. Robot Robot Configuration and Operation Guide] of this product tutorial.

For Orin boards, simply open a terminal and enter the commands mentioned in this lesson.

Once the function is enabled, the camera captures images and recognizes gestures to control the robot's movement.

Gesture	Action
Gesture "5"	Car moves forward
Gesture "1"	Car moves right
Gesture "2"	Car stops
Gesture "3"	Car moves backward
Gesture "4"	Car moves left

**MediaPipe Hands infers the 3D coordinates of 21 hand joints from a single frame**



## 2. Program Code Reference Path

**After entering the Docker container, the source code for this function is located at:**

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/yahboomcar_astra
```

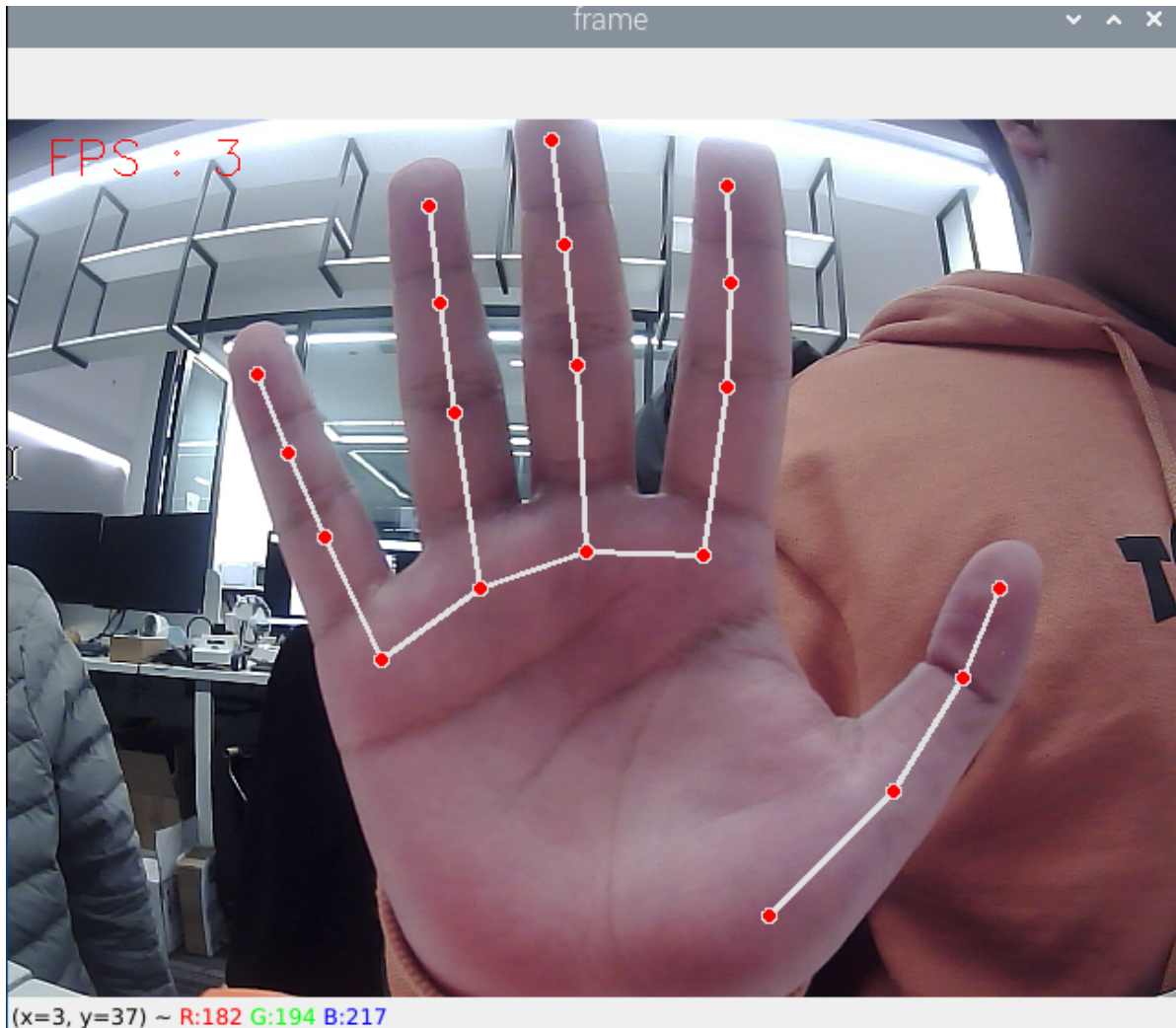
## 3. Program Startup

### 3.1. Startup Command

After entering the Docker container, enter the following command in the terminal:

```
ros2 run yahboomcar_astra HandCtrl
```

Enable this function and place your hand in front of the camera. The image will depict the shape of your finger. Once the program recognizes the gesture, it will send the speed to the chassis, thereby controlling the car's movement.



## 4. Core Code

### 4.1. HandCtrl.py

- Importing Key Libraries

```
from yahboomcar_mediapipe.media_library import * #This library contains  
functions for detecting hands and obtaining gestures
```

- Getting Finger Data

```
frame, lmList, _ = self.hand_detector.findHands(frame)
fingers = self.hand_detector.fingersUp(lmList)
sum(fingers)
fingers[]
```

As you can see, the hand is first detected, the value of `lmList` is obtained, and then passed to the `fingersUp` function. The `fingersUp` function is used to detect which fingers are extended. The value of an extended finger is 1. The specific code here can also be seen in the `media_library.py` function, which has a detailed explanation. In fact, it is actually to determine the `xy` value of the finger joint to determine when it is extended. The `sum(fingers)` function is used to count the number of extended fingers. `fingers[]` can be used to enumerate fingers. For example, the index finger is represented by `fingers[1]`.

- Release speed to chassis

```
self.media_ros.pub_vel(x,y,z)    #This function is also in media_library.py
```