

# Voice Control Tracking and Grasping AprilTag

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

Voice commands are issued to instruct the robotic arm to track and grasp AprilTags. The program will control the robotic arm to track AprilTags and grasp AprilTag blocks when conditions are met according to the command.

## 2. Startup and Operation

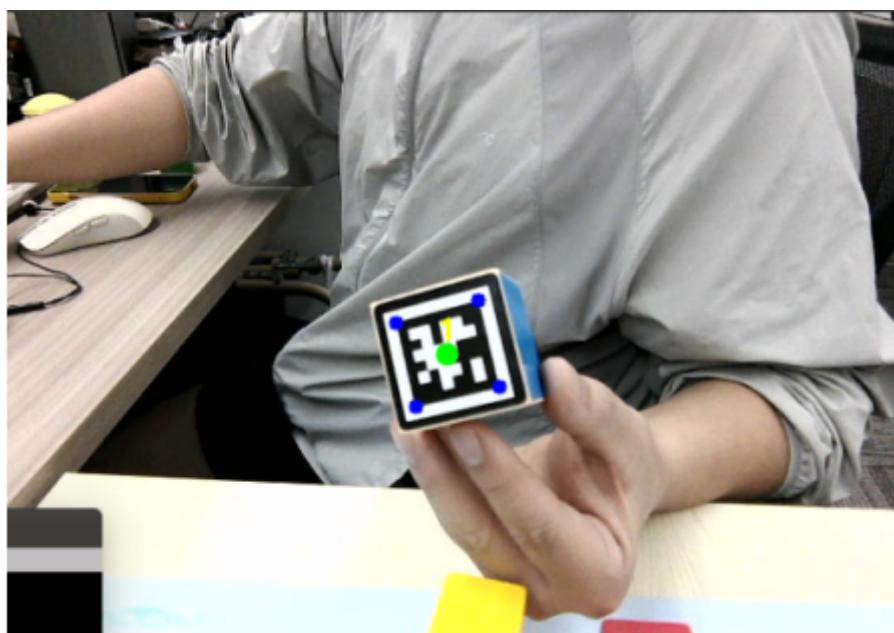
### 2.1. Startup

Users with Jetson-Nano board version need to enter the docker container and input the following commands. Orin board users can directly open the terminal and input the following commands,

```
#Start camera and inverse kinematics  
ros2 launch dofbot_pro_info camera_arm_kin.launch.py  
#Start speech recognition and broadcast  
ros2 launch yahboom_speech speech.launch.py  
#Start tracking and grasping AprilTag  
ros2 run dofbot_pro_voice_ctrl apriltag_follow_vc
```

### 2.2. Operation Steps

After all programs are running, the robotic arm will move to the tracking posture. At this time, hold an AprilTag and place it in the image. Say "Hello, yahboom" to the speech recognition module, and the speaker will broadcast "here". Then say "Track and clip machine code" to the speech module. Slowly move the AprilTag, and the robotic arm will track the AprilTag. Wait for the robotic arm to stop tracking and stabilize, then you will hear a beep sound. The robotic arm will grasp the held AprilTag block, then place it at the designated position, and finally return to the robotic arm's tracking posture.



### 3. Core Code Analysis

#### 3.1. Tracking and Grasping AprilTag Node apriltag\_follow\_VC.py

Mainly explains how to subscribe to and process speech recognition result topics and publish voice broadcast topics.

```
#Create subscriber for speech recognition result topic
self.sub_voice =
    self.create_subscription(Int8,"voice_result",self.getVoiceResultCallBack,1)
#Create publisher for voice broadcast topic
self.pub_playID = self.create_publisher(Int8,"player_id", 1)

#Callback function, processes received speech recognition result topic data. If
the received speech recognition result topic data is 105, it means the command is
"Track and clip machine code", change self.start_flag to true indicating that
AprilTag position information can be published and publish voice broadcast topic
data, broadcasting "OK" audio file.
def getVoiceResultCallBack(self,msg):
    if msg.data == 105:
        self.start_flag = True
        play_id = Int8()
        play_id.data = 45
        self.pub_playID.publish(play_id)
        print("Start tracking and grabbing.")
```