Voice Control Stacking

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].

Orin board users can directly open the terminal and enter the tutorial commands to run. Jetson-Nano board users need to enter the docker container first, then enter the tutorial commands in the docker to start the program.

1. Function Description

Through voice commands specifying the current color block to be grabbed, the robotic arm will grab the corresponding color block and stack them layer by layer.

2. Startup and Operation

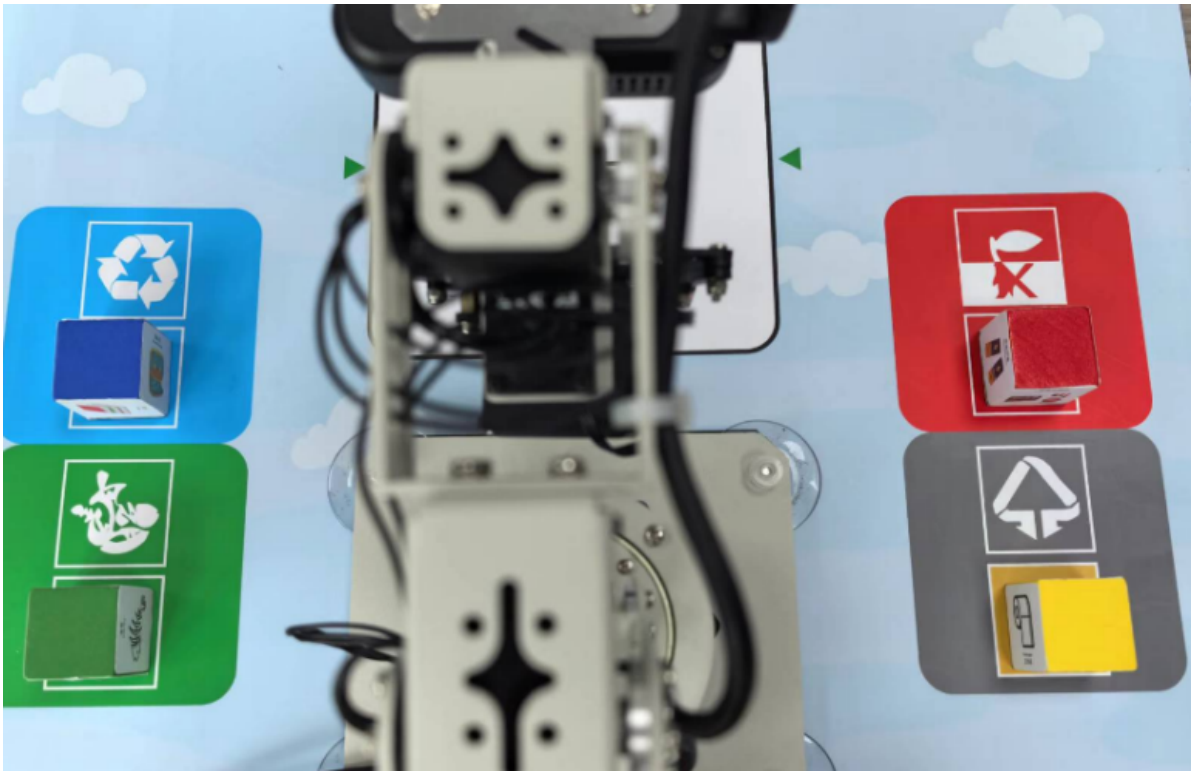
2.1. Startup

Open the terminal and enter the following command to start:

```
python3 ~/dofbot_voice/scripts/voice_control_stack.py
```

2.2. Operation Steps

After the program runs successfully, place blocks according to the following arrangement: yellow blocks in the yellow area, red blocks in the red area, green blocks in the green area, blue blocks in the blue area, as shown in the figure below:



2.2.1. Stacking

Say "Hello, yahboom" to the voice module. The voice module will reply "here" to indicate successful wake-up. Then say "Stack the yellow block/Stack the red block/Stack the green block/Stack the blue block" to the voice module. The voice module will reply and broadcast "ok", and the robotic arm will go to the corresponding color block area, grab the block, and place it in the designated area. The voice module will reply and broadcast "ok". Each time you use voice commands to grab the target color block and place it, you will eventually complete the stacking.

2.2.2. Push Down

Say "Hello, yahboom" to the voice module. The voice module will reply "here" to indicate successful wake-up. Then say "Push down" to the voice module. The voice module will reply and broadcast "ok", then the robotic arm will execute the action to push down the stacked blocks.

3. Core Code Analysis

Jetson-Nano users need to enter the docker container to view

`Source code path: ~/dofbot_voice/scripts/simple_voice_ctrl.py`

```
    while True:
        #Get voice recognition result
        res = mySpeech.speech_read()
        time.sleep(0.01)
        #Grab yellow block
        if res == 77:
            if yellow_grabbed == 0:
                mySpeech.void_write(77)
                Arm.Arm_Buzzer_On(1)
                time.sleep(.1)
                start_move_arm(1)
                time.sleep(1)
                # global yellow_grabbed
                yellow_grabbed = 1
        #Grab red block
        elif res == 78:
            if red_grabbed == 0:
                mySpeech.void_write(77)
                Arm.Arm_Buzzer_On(1)
                time.sleep(.1)
                start_move_arm(2)
                # global red_grabbed
                red_grabbed = 1
        #Grab green block
        elif res == 79:
            if green_grabbed == 0:
                mySpeech.void_write(77)
                Arm.Arm_Buzzer_On(1)
                time.sleep(.1)
                start_move_arm(3)
                # global green_grabbed
                green_grabbed = 1
        #Grab blue block
        elif res == 80:
            if blue_grabbed == 0:
                mySpeech.void_write(77)
```

```
                Arm.Arm_Buzzer_On(1)
                time.sleep(.1)
                start_move_arm(4)
                # global blue_grabbed
                blue_grabbed = 1
        elif res == 82: #Push down
            mySpeech.void_write(82)
            Arm.Arm_Buzzer_On(1)
            time.sleep(.1)
            start_move_arm(5)
            yellow_grabbed = 0
            red_grabbed = 0
            green_grabbed = 0
            blue_grabbed = 0


#Number function definition
def number_action(index):
    if index == 1:
        # Grab yellow building block
        arm_move(p_top, 1000)
        arm_move(p_Yellow, 1000)
        arm_clamp_block(1)
        Arm.Arm_serial_servo_write(2, 90, 1000)
        time.sleep(1)
        arm_move(p_top, 1000)
    elif index == 2:
        # Grab red building block
        arm_move(p_top, 1000)
        arm_move(p_Red, 1000)
        arm_clamp_block(1)
        Arm.Arm_serial_servo_write(2, 90, 1000)
        time.sleep(1)
        arm_move(p_top, 1000)
    elif index == 3:
        # Grab green building block
        arm_move(p_top, 1000)
        arm_move(p_Green, 1000)
        arm_clamp_block(1)
        Arm.Arm_serial_servo_write(2, 90, 1000)
        time.sleep(1)
        arm_move(p_top, 1000)
    elif index == 4:
        # Grab blue building block
        arm_move(p_top, 1000)
        arm_move(p_Blue, 1000)
        arm_clamp_block(1)
        Arm.Arm_serial_servo_write(2, 90, 1000)
        time.sleep(1)
        arm_move(p_top, 1000)

#Place stacked blocks, layer indicates which layer to place on
def put_down_block(layer):
    if layer == 1:
        arm_move(p_layer_1, 1000)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)
    elif layer == 2:
```

```
        arm_move(p_layer_2, 1000)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)
    elif layer == 3:
        arm_move(p_layer_3, 1000)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)
    elif layer == 4:
        arm_move(p_layer_4, 1000)
        time.sleep(.1)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)
    #Voice broadcast, placement complete
    mySpeech.void_write(81)
```