12. Synchronous action of two robotic arms

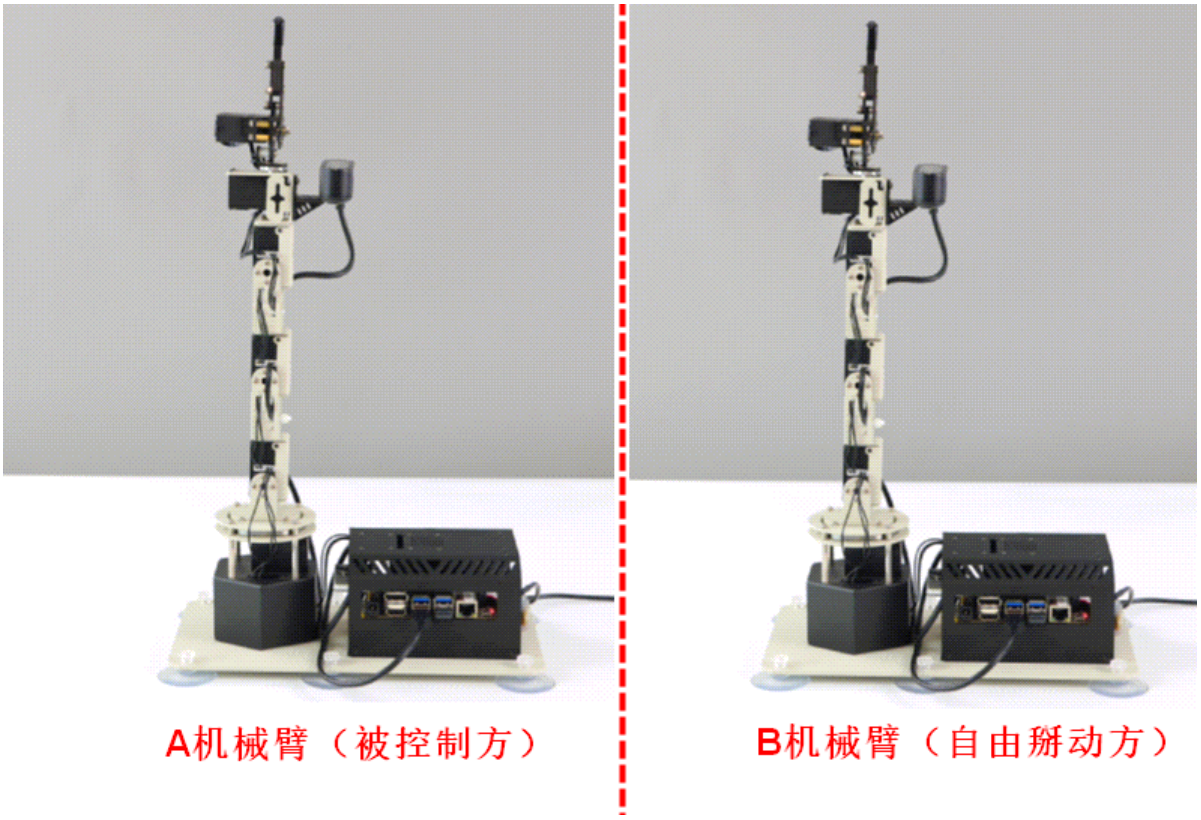
Note: This experiment requires two robotic arms.

12.1. Communication principle

The principle of this experiment is to use one of the robotic arms as a server, open the TCP network service, receive data, and perform operations; and the other robotic arm as a client, that is, the operator, reads the angle values of all servos in sequence, and then immediately sends the angles of all servos to the server through the TCP service; at this time, the client has been quickly reading the angle values of the servos and sending them to the server, and the server receives and sets to immediately update the angles of all servos, in this way to achieve the function of remote synchronous action of the two robotic arms.

12.2. Preparation before the experiment

1. This experiment involves two robotic arms with the same hardware. Except for the different running programs, everything else can be the same. If the robotic arm uses the factory-provided system image, please end the factory-provided large program that starts automatically at boot time before conducting this experiment.
2. Ensure that both robotic arms are connected to the same LAN, that is, connected to the same router (wired or wireless connection is OK), and can communicate within the LAN.
3. Since there are two robotic arms involved here, in order to easily distinguish them, we call the robotic arm that serves as the server the A robotic arm (the controlled party), and the robotic arm that serves as the controller the B robotic arm (the free bending party). Since the movement of the A robotic arm depends on the movement of the B robotic arm, in order to avoid the A robotic arm's movement range being too large when the program is just started, it is best to adjust both the A robotic arm and the B robotic arm to a vertical upward state before starting the program.



12.3, Program code

[Server code: A robot arm]

For specific code, please see the content of the following path:

```
~/dofbot_pro/dofbot_ctrl/scripts/12.sync_movement/A_arm_follow.py
```

[Client code: B robot arm]

For specific code, please see the content of the following path:

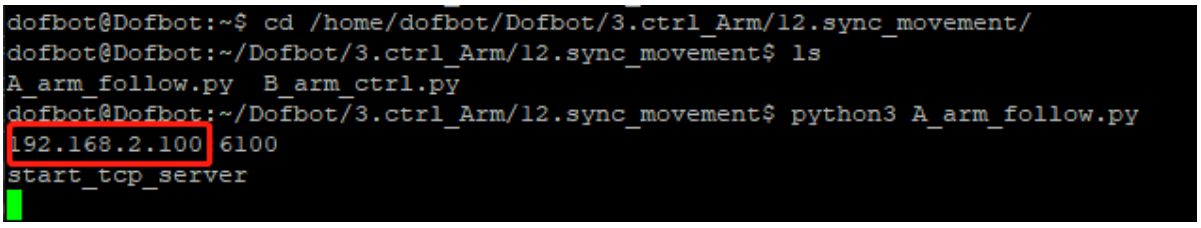
```
~/dofbot_pro/dofbot_ctrl/scripts/12.sync_movement/B_arm_ctrl.py
```

12.4, Operation experiment and experimental purpose

Experimental purpose: Use B robot arm to remotely control A robot arm.

1. Open the terminal of robot arm A and run the A_arm_follow.py program. Then write down the printed IP address.

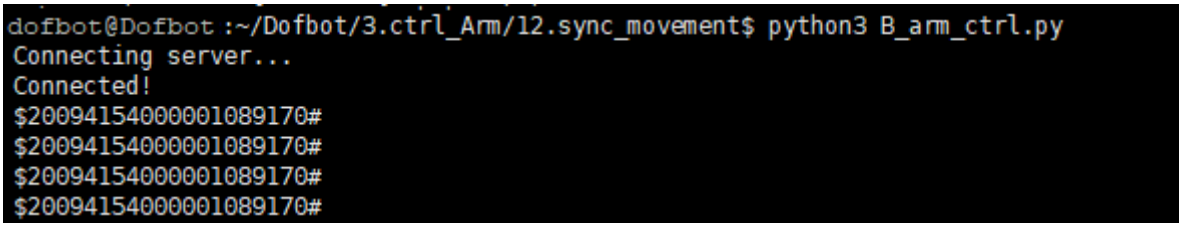
```
cd ~/dofbot_pro/dofbot_ctrl/scripts/12.sync_movement/
python3 A_arm_follow.py
```



2. Open the terminal of robot arm B and replace the IP address written down in the previous step with the IP address position in B_arm_ ctrl.py.

3. After the modification is completed, save and exit, and then run the B_arm_ ctrl.py program.

```
cd ~/dofbot_pro/dofbot_ctrl/scripts/12.sync_movement/
python3 B_arm_ctrl.py
```



Note that you must run the A_arm_follow.py program of the A robot first, and then run the B_arm_ctrl.py program of the B robot. The B robot terminal will print the connection information and the sent data, and the communication between the two robots will officially start.

At this point, you can see that the state of Arm A is the same as that of Arm B, and you can modify the posture of Arm A by moving Arm B. By transmitting commands via TCP, you can achieve the effect of synchronized movement of the two arms. Arm A is controlled with an angle limit, while Arm B moves automatically without an angle limit. If the No. 1 servo of Arm B is moved more than 180 degrees, Arm A will stay at 180 degrees until the angle of Arm B is adjusted back to a reasonable range before it can continue to move.