Two robotic arms move synchronously

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; while the other robotic arm serves as the client, that is, the operator, and reads the data of all servos in sequence. angle value, and then immediately sends the angles of all servos to the server through the TCP service; at this time, the client has been rapidly reading the angle values of the servos and sending them to the server. The server receives and sets the angles of all servos and immediately updates them. In this way, the function of remote synchronization of two robotic arms is realized.

2. Preparation before 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 robot arm uses the system image that comes with the factory, please end the factory program that starts automatically at boot before performing this experiment.

2. Ensure that both robot arms are connected to the same LAN, that is, connected to the same router (wired or wireless connection is acceptable), and can communicate within the LAN.

3. Since there are two robot arms involved here, in order to easily distinguish them, we call the robot arm that is the server A robot arm (controlled party), and the robot arm that is the controller is called B robot arm (Free to move). Since the movement of robotic arm A depends on the movement of robotic arm B, in order to avoid that the range of movement of robotic arm A is too large when starting the program, it is best to adjust both robotic arm A and robotic arm B to the vertical position before starting the program. Straight upward state.

A机械臂（被控制方）

B机械臂（自由掰动方）

3. Program code

[Server code: A robotic arm]

Please see the content of the following path for the specific code:

```
/root/Dofbot/3.ctrl_Arm/12.sync_movement/A_arm_follow.py
```

[Client code: B robotic arm]

Please see the content of the following path for the specific code:

```
/root/Dofbot/3.ctrl_Arm/12.sync_movement/B_arm_ctrl.py
```

4. Operational experiment and experimental purpose

Experiment purpose: Use B robotic arm to remotely control A robotic arm.

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

```
cd /root/Dofbot/3.ctrl_Arm/12.sync_movement/
python3 A_arm_follow.py
```

```
root@raspberrypi:~/Dofbot/3.ctrl_Arm/12.sync_movement# python3 A_arm_follow.py
192.168.2.85:6100
start_tcp_server
```

2. Open the terminal of robotic arm B and replace the IP address noted in the previous step to the ip address in B_arm_ ctrl.py.

```
if __name__ == '__main__':
    # Modify the following parameters according to the IP address of the server
    # 根据服务器的IP地址修改以下参数
    ip = '192.168.2.85'
    port = 6100
    try:
        connect_tcp_server(ip, port)
    except KeyboardInterrupt:
        waitClose(g_sock)
        print(" Program closed! ")
        pass
```

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

```
python3 B_arm_ctrl.py
```

```
root@raspberrypi:/# cd /root/Dofbot/3.ctrl_Arm/12.sync_movement/
root@raspberrypi:~/Dofbot/3.ctrl_Arm/12.sync_movement# nano B_arm_ctrl.py
root@raspberrypi:~/Dofbot/3.ctrl_Arm/12.sync_movement# python3 B_arm_ctrl.py
Connecting server...
Connected!
$20090083090090090000#
$20090083090090090000#
$20090083090090090000#
$20090083090090090000#
```

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

At this time, you can see that the status of robotic arm A is the same as that of robotic arm B, and you can modify the posture of robotic arm A by moving the posture of robotic arm B. Through such TCP transmission commands, the effect of synchronous movement of two robotic arms can be achieved. Robotic arm A is controlled with an angle limit, while robotic arm B automatically moves without any angle limit. If the No. 1 servo added to robotic arm B moves more than 180 degrees, A will stay at 180 degrees until robotic arm B Only when the angle is adjusted back to a reasonable range can the movement continue.