2.Servo calibration

1. API Introduction

The API corresponding to turning on or off the torque of the bus servo is:

Arm_serial_set_torque(enable)

Parameter explanation:

enable:

enable=0 turns off the torque, the robotic arm can manually adjust the angle of the servo, and does not receive level signals;

enable=1 turns on the torque, the servo receives the level signal to maintain the current angle, and the angle can be changed only by sending a command.

Return value: None.

Arm_serial_servo_write_offset_switch(id)

Parameter explanation:

id:

When id=0, clear the center adjustment of all servos and restore to default;

id=1~6, corresponding to the ID numbers of six servos.

After receiving this command, the underlying MCU will read the angle data of the servo with the corresponding ID. If it is within a reasonable range, it will be saved. If it exceeds the range or the servo ID cannot be found, it will not be saved.

Return value: None.

Arm_serial_servo_write_offset_state()

Parameter explanation:

Return value: Returns the state of setting the median deviation.

state=0 means that the servo has not been detected;

state=1 means that the setting of the center position is successful;

state=2 means that the range value of the setting center position is exceeded.

2.Code

Code path： /home/jetson/Dofbot/2.sys_settings/2.Offset/offset.ipynb

This code must be executed step by step so that the servo can be adjusted correctly. The result of the adjustment is to keep the robotic arm in an upright position.

This code can only be used when the center position needs to be adjusted, and cannot be used casually, otherwise it will cause the center position to be inaccurate and affect the grasping effect of the robotic arm.

```
#!/usr/bin/env   python3
#coding=utf-8
import   time
from   Arm_Lib import Arm_Device

#   Create a robotic arm object
Arm   = Arm_Device()
time.sleep(.1)
```

```
#Return the servo to the upright position
Arm.Arm_serial_servo_write6(90,   90, 90, 90, 90, 180, 1000)
time.sleep(2)
```

```
#   Turn off the torque. At this time, you can adjust the angle of the servo by
hand.
Arm.Arm_serial_set_torque(0)
```

```
#   Separately set the center deviation of a certain servo
id   = 6
Arm.Arm_serial_servo_write_offset_switch(id)
time.sleep(.1)
state   = Arm.Arm_serial_servo_write_offset_state()
if   state == 1:
    print("set offset ok!")
elif   state == 2:
    print("error! set offset overrun   !")
elif   state == 0:
    print("error! set offset error   !")
```

```
#   Set the center deviation of all servos (No. 1-6) at one time
for   i in range(6):
    id = i + 1
      Arm.Arm_serial_servo_write_offset_switch(id)
    time.sleep(.1)
    state =   Arm.Arm_serial_servo_write_offset_state()
    if state == 1:
        print("id:%d set offset   ok!" % id)
    elif state == 2:
        print("error!id:%d set offset   overrun !" % id)
    elif state == 0:
        print("error!id:%d set offset   error !" % id)
```

```
#   After the adjustment is complete, turn on the torque
Arm.Arm_serial_set_torque(1)
```

```
# Clear the median deviation of all servo settings and restore to the default
state.
# If you need to clear the median deviation of all servos, please delete the #
symbol below and run this unit again
#Arm.Arm_serial_servo_write_offset_switch(0)
```

```
del  Arm  # Release the Arm object
```