1. API introduction

Arm_serial_set_torque(enable)

Parameter explanation:

enable: enable=0 means turn off the torque, we can manually change the angle of all servos of the robotic arm without receiving the level signal;

enable=1 means turn on the torque, we can't manually change the angle of all servos of the robotic arm the servo. The servo must be controlled by sending commands.

Return value: None.

Arm_serial_servo_write_offset_switch(id)

Parameter explanation:

id: id=0, clear the median value of all servos and restore the default;

id=1~6, corresponding to the ID numbers of six servos, the bottom MCU(coprocessor) will read the angle value of the servo corresponding to the ID after receiving this command. If it is within a reasonable range, it will be saved. If it exceeds the specified range or the servo ID cannot be found, it will not be saved.

Return value: None.

Arm_serial_servo_write_offset_state()

Return value: state

state=0 means that the servo is not detected;

state=1 means that the middle position is set successfully;

state=2 means that the set median data exceeds the specified range.

2.About code

Path: /home/dofbot/Dofbot/2.sys_settings/2.Offset/offset.ipynb

This code must be executed step by step to ensure the correct adjustment of the servo. The result of the adjustment is to keep centered vertically.

This program can only be used when the middle position of the servo needs to be adjusted. Do not use it at will, otherwise it will cause the middle position to be inaccurate and affect the effects of all functions.

```
#!/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)
```

```
# Middle servo
Arm.Arm_serial_servo_write6(90, 90, 90, 90, 90, 180, 1000)
time.sleep(2)
```

```
# Turn off the torque, we can manually change the angle of all servos of the robotic arm
Arm.Arm_serial_set_torque(0)
```

```
# After adjusting the angle of a certain servo, you can set the median deviation of a certain servo
separately
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 median deviation of all servos (No.1-6)
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 the default state.
# If you need to clear the median deviation of all servos, please delete the # below and run this unit
#Arm.Arm_serial_servo_write_offset_switch(0)
```

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