

Lesson 3 Move Up and Down

1. Project Logic

In this lesson, you can learn how to control the end effector to move up and down by setting the coordinate of end effector.

Inverse kinematics is mainly used to calculate the joint angle according to the coordinate of the set target position. Then the joint angle is converted into the pulse width of the servo to realize that the end effector is controlled to move to the target position.

The path of the program source code:

/home/pi/ArmPi_mini/ArmIK/ArmMoveIK_demo.py

```

3  import sys
4  import time
5  sys.path.append('/home/pi/ArmPi_mini/')
6  from ArmIK.ArmMoveIK import *
7
8  if sys.version_info.major == 2:
9      print('Please run this program with python3!')
10     sys.exit(0)
11
12     print("""
13     *****
14     ***** 功能:逆运动学上下移动例程 *****
15     *****
16
17     Official website: http://www.hiwonder.com
18     Online mail: https://huaner.tmall.com/
19
20     Tips:
21     * 按下 Ctrl+C 可关闭此次程序运行，若失败请多次尝试！
22
23     """)
24
25     # 实例化逆运动学库
26     AK = ArmIK()
27
28     if __name__ == "__main__":
29         """
30         AK.setPitchRangeMoving(coordinate_data, alpha, alpha1, alpha2, movetime):
31         给定坐标coordinate_data和俯仰角alpha,以及俯仰角范围的范围alpha1,
32         alpha2, 自动寻找最接近给定俯仰角的解,并转到目标位置
33         若无解返回False,否则返回舵机角度、俯仰角、运行时间
34         坐标单位cm, 以元组形式传入,例如(0, 5, 10)
35         alpha: 为给定俯仰角
36         alpha1和alpha2: 为俯仰角的取值范围
37         movetime:为舵机转动时间,单位ms,如果不给出时间,则自动计算
38         """
39         AK.setPitchRangeMoving((0, 6, 18), 0, -90, 90, 1500) # 设置机械臂初始位置(x:0, y:6, z:18),运行时间:1500毫秒
40         time.sleep(1.5) # 延时1.5秒
41
42         for i in range(2): # for循环运行2次
43             AK.setPitchRangeMoving((0, 6, 22), 0, -90, 90, 1000) # 设置机械臂上移到位置(x:0, y:6, z:22),运行时间:1000毫秒
44             time.sleep(1.2) # 延时1.2秒
45
46             AK.setPitchRangeMoving((0, 6, 18), 0, -90, 90, 1000) # 设置机械臂下移到初始位置,运行时间:1000毫秒
47             time.sleep(1.2) # 延时1.2秒

```

2. Operation Steps

Note: The input command should be case sensitive, and “Tab” can be used to complement keywords.

1) Turn on ArmPi mini, and connect it to Raspberry Pi system desktop via VNC viewer.



2) Click  in the upper left corner (as the figure shown below), or press “Ctrl+Alt+T” to open LX terminal.

3) Enter command “cd ArmPi_mini/ArmIK/” and press “Enter” to enter the directory of game programs.

```
pi@raspberrypi:~ $ cd ArmPi_mini/ArmIK/
```

4) Enter command “sudo python3 ArmMoveIK_demo.py” and press “Enter” to start game.

```
pi@raspberrypi:~/ArmPi_mini/ArmIK $ sudo python3 ArmMoveIK_demo.py
```

5) If want to exit the game, you can press “Ctrl+C”. If fail to close, please try multiple times.

3. Project Outcome

After starting the game, the end effector of robot arm will move up first, and then move down.

4. Function Extension

4.1 Modify Movement Direction

The end effector can also be controlled to move along x and y axes. Here take x axis as example. The specific operation steps are as follow:

- 1) Click  icon and press “Ctrl+Alt+T” to open the LX terminal.
- 2) Enter command “cd ArmPi_mini/ArmIK/” and press “Enter” to enter the directory of game programs.

```
pi@raspberrypi:~ $ cd ArmPi_mini/ArmIK/
```

- 3) Enter command “sudo vim ArmMoveIK_demo.py” and press “Enter” to open the program file.

```
pi@raspberrypi:~/ArmPi_mini/ArmIK $ sudo vim ArmMoveIK_demo.py
```

- 4) Find the following code:

```
41     for i in range(2): # for循环运行2次
42         AK.setPitchRangeMoving((0, 6, 22), 0, -90, 90, 1000) # 设置机械臂上移
           到位置(x:0, y:6, z:22),运行时间:1000毫秒
43         time.sleep(1.2) # 延时1.2秒
44
```

- 5) Press “i” key on the keyboard to enter the editing mode.

```
41     for i in range(2): # for循环运行2次
42         AK.setPitchRangeMoving((0, 6, 22), 0, -90, 90, 1000) # 设置机械臂上移
           到位置(x:0, y:6, z:22),运行时间:1000毫秒
43         time.sleep(1.2) # 延时1.2秒
44
45         AK.setPitchRangeMoving((0, 6, 18), 0, -90, 90, 1000) # 设置机械臂下移
           到初始位置,运行时间:1000毫秒
46         time.sleep(1.2) # 延时1.2秒
47
-- 插入 --
```

39,1 底端

- 6) Change “(0, 6, 22)” to “(4, 6, 18)”, as the figure shown below:

```

40
41     for i in range(2): # for循环运行2次
42         AK.setPitchRangeMoving((4, 6, 18), 0, -90, 90, 1000) # 设置机械臂上移
到位置(x:0, y:6, z:22),运行时间:1000毫秒
43         time.sleep(1.2) # 延时1.2秒

```

- 7) After modification, press “Esc” and enter “:wq”, and then press “Enter” to save and close the file.

```

45         AK.setPitchRangeMoving((0, 6, 18), 0, -90, 90, 1000) # 设置机械臂下移
到初始位置,运行时间:1000毫秒
46         time.sleep(1.2) # 延时1.2秒
47
:wq

```

- 8) Enter command “sudo python3 ArmMoveIK_demo.py” and press “Enter” to start game and check the game effect after modification.

```

pi@raspberrypi:~/ArmPi_mini/ArmIK $ sudo python3 ArmMoveIK_demo.py

```

5. Program Analysis

5.1 Import Library

```

3 import sys
4 import time
5 sys.path.append('/home/pi/ArmPi_mini/')
6 from ArmlK.ArmMoveIK import *

```

“sys” is the most commonly used library interacting strongly with the Python interpreter; “time” is the library for processing time; “ArmIK” is the library for processing inverse kinematics.

If want to call a function in library, you can use this format “library name+function name (parameter, parameter)” to use the code. For example,

```

43 time.sleep(1.2) # 延时1.2秒

```

The code above is used to call sleep() in “time” library and this function is for delaying.

There are some built-in libraries in Python, so they can be called directly. For example, “time”, “cv2” and “math”. In addition, you can also customize a new library like “ArmlK.ArmMoveIK” inverse kinematics library.

5.2 Instantiate Function Library

The name of function library is too long to memorize. For calling function easily, the library can be instantiated. For example,

```
25 # 实例化逆运动学库
26 AK = ArmlK()
```

After instantiating, you can directly call the function when inputting the function like “AK.function name (parameter, parameter)”.

5.3 Control Robot Arm

```
38 AK.setPitchRangeMoving((0, 6, 18), 0, -90, 90, 1500) # 设置机械臂初始位置(x:0, y:6, z:18),运行时间:1500毫秒
39 time.sleep(1.5) # 延时1.5秒
40
41 for i in range(2): # for循环运行2次
42     AK.setPitchRangeMoving((0, 6, 22), 0, -90, 90, 1000) # 设置机械臂上移到位置(x:0, y:6, z:22),运行时间:1000毫秒
43     time.sleep(1.2) # 延时1.2秒
44
45     AK.setPitchRangeMoving((0, 6, 18), 0, -90, 90, 1000) # 设置机械臂下移到初始位置,运行时间:1000毫秒
46     time.sleep(1.2) # 延时1.2秒
```

The function AK.setPitchRangeMoving() in inverse kinematics library is used to control robot arm to move. Take “AK.setPitchRangeMoving((0, 6, 18), 0, -90, 90, 1500)” for example.

The first parameter “(0, 6, 18)” is the end-effector coordinate. The three parameters represent the coordinate of x, y and z axes.

The second parameter “0” is the pitch angle.

The third parameter “-90” is the minimum range of the pitch angle.

The fourth parameter “90” is the maximum range of the pitch angle.

The fifth parameter “1500” is the running time and the unit is millisecond.