

13. 扩展轴

13.1. 设置485扩展轴参数

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoSetParam(servoId,servoCompany,servoModel,servoSoftVersion,servoResolution,axisMechTransRatio)</code>
描述	设置485扩展轴参数
必选参数	<ul style="list-style-type: none"><code>servoId</code> : 伺服驱动器ID, 范围[1-15],对应从站ID;<code>servoCompany</code> : 伺服驱动器厂商, 1-戴纳泰克;<code>servoModel</code> : 伺服驱动器型号, 1-FD100-750C;<code>servoSoftVersion</code> : 伺服驱动器软件版本, 1-V1.0;<code>servoResolution</code> : 编码器分辨率;<code>axisMechTransRatio</code> : 机械传动比;
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.1.1. 代码示例

```
1  from time import sleep
2  from fairino import Robot
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  ret = robot = Robot.RPC('192.168.58.2')
5
6  ret =robot.AuxServoSetParam(1,1,1,1,131072,15.45)#设置485扩展轴参数
7  print("AuxServoSetParam",ret)
8  sleep(1)
9
10 ret =robot.AuxServoGetParam(1)#获取485扩展轴配置参数
11 print("AuxServoGetParam",ret)
12 sleep(1)
13
14 ret =robot.AuxServoGetStatus(1)#查询状态
15 print("AuxServoGetStatus",ret)
16 sleep(1)
17
18 ret =robot.AuxServoClearError(1)#清除错误
19 print("AuxServoClearError",ret)
20 sleep(1)
```



13.2. 获取485扩展轴配置参数

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoGetParam(servoId)</code>
描述	获取485扩展轴配置参数
必选参数	<ul style="list-style-type: none"><code>servoId</code>: 伺服驱动器ID, 范围[1-15],对应从站ID;
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode;<code>servoCompany</code>: 伺服驱动器厂商, 1-戴纳泰克;<code>servoModel</code>: 伺服驱动器型号, 1-FD100-750C;<code>servoSoftVersion</code>: 伺服驱动器软件版本, 1-V1.0;<code>servoResolution</code>: 编码器分辨率;<code>axisMechTransRatio</code>: 机械传动比;

13.2.1. 代码示例

参考设置485扩展轴参数的代码示例

13.3. 设置485扩展轴使能/去使能

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoEnable(servoId,status)</code>
描述	设置485扩展轴使能/去使能
必选参数	<ul style="list-style-type: none"><code>servoId</code>: 伺服驱动器ID, 范围[1-15],对应从站ID;<code>status</code>: 使能状态, 0-去使能, 1-使能;
默认参数	无
返回值	错误码 成功-0 失败- errcode



latest



13.3.1. 代码示例

```
1  from time import sleep
2  from fairino import Robot
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  ret = robot = Robot.RPC('192.168.58.2')
5  ret =robot.AuxServoEnable(1,0)#修改控制模式前需去使能
6  print("AuxServoEnable(0)",ret)
7  sleep(3)
8
9  ret =robot.AuxServoSetControlMode(1,0)#设置为位置模式
10 print("AuxServoSetControlMode",ret)
11 sleep(3)
12
13 ret =robot.AuxServoEnable(1,1)#修改控制模式后需使能
14 print("AuxServoEnable(1)",ret)
15 sleep(3)
16
17 ret =robot.AuxServoHoming(1,1,10,10)#回零
18 print("AuxServoHoming",ret)
19 sleep(5)
20
21 ret =robot.AuxServoGetStatus(1)#查询状态
22 print("AuxServoGetStatus",ret)
23 sleep(1)
24 i=1
25 while(i<5):
26     ret =robot.AuxServoSetTargetPos(1,300*i,30)#位置模式运动, 速度30
27     print("AuxServoSetTargetPos",ret)
28     sleep(11)
29     ret =robot.AuxServoGetStatus(1)#查询状态
30     print("AuxServoGetStatus",ret)
31     sleep(1)
32     i=i+1
```

13.4. 设置485扩展轴控制模式

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoSetControlMode(servoId,mode)</code>
描述	设置485扩展轴控制模式
必选参数	<ul style="list-style-type: none"><code>servoId</code>: 伺服驱动器ID, 范围[1-15],对应从站ID;<code>mode</code>: 控制模式, 0-位置模式, 1-速度模式;
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.4.1. 代码示例

参考设置485扩展轴使能/去使能的代码示例



latest



13.5. 设置485扩展轴目标位置(位置模式)

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoSetTargetPos(servoId,pos,speed)</code>
描述	设置485扩展轴目标位置(位置模式)
必选参数	<ul style="list-style-type: none"><code>servoId</code>: 伺服驱动器ID, 范围[1-15],对应从站ID;<code>pos</code>: 目标位置, mm或°;<code>speed</code>: 目标速度, mm/s或°/s;
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.5.1. 代码示例

参考设置485扩展轴使能/去使能的代码示例

13.6. 设置485扩展轴目标速度(速度模式)

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoSetTargetSpeed(servoId,speed)</code>
描述	设置485扩展轴目标速度(速度模式)
必选参数	<ul style="list-style-type: none"><code>servoId</code>: 伺服驱动器ID, 范围[1-15],对应从站ID;<code>speed</code>: 目标速度, mm/s或°/s;
默认参数	无
返回值	错误码 成功-0 失败- errcode



latest



13.6.1. 代码示例

```
1  from time import sleep
2  from fairino import Robot
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  ret = robot = Robot.RPC('192.168.58.2')
5  ret = robot.AuxServoEnable(1,0)#修改控制模式前需去使能
6  print("AuxServoEnable(0)",ret)
7  sleep(3)
8
9  ret = robot.AuxServoSetControlMode(1, 1) # 设置为速度模式
10 print("AuxServoSetControlMode",ret)
11 sleep(3)
12
13 ret = robot.AuxServoEnable(1,1)#修改控制模式后需使能
14 print("AuxServoEnable(1)",ret)
15 sleep(3)
16
17 ret = robot.AuxServoHoming(1,1,10,10)#回零
18 print("AuxServoHoming",ret)
19 sleep(5)
20
21 ret = robot.AuxServoGetStatus(1)#查询状态
22 print("AuxServoGetStatus",ret)
23 sleep(1)
24
25 ret = robot.AuxServoSetTargetSpeed(1, 30) # 速度模式运动, 速度30
26 print("AuxServoSetTargetSpeed", ret)
27 sleep(10)
28
29 ret = robot.AuxServoGetStatus(1) # 查询状态
30 print("AuxServoGetStatus", ret)
31 sleep(1)
32
33 ret = robot.AuxServoSetTargetSpeed(1, 60) # 速度模式运动, 速度60
34 print("AuxServoSetTargetSpeed", ret)
35 sleep(10)
36 ret = robot.AuxServoGetStatus(1) # 查询状态
37 print("AuxServoGetStatus", ret)
38 sleep(1)
39
40 ret = robot.AuxServoSetTargetSpeed(1, 0) # 结束速度模式运动前应当把速度设为0
41 print("AuxServoSetTargetSpeed", ret)
42 sleep(3)
43 ret = robot.AuxServoGetStatus(1) # 查询状态
44 print("AuxServoGetStatus", ret)
45 sleep(1)
```

13.7. 设置485扩展轴目标转矩(力矩模式)-暂未开放

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoSetTargetTorque(servoId,torque)</code>
描述	设置485扩展轴目标转矩(力矩模式)



latest



必选参数	<ul style="list-style-type: none"><code>servoId</code>：伺服驱动器ID，范围[1-15],对应从站ID；<code>torque</code>：目标力矩，Nm；
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.8. 设置485扩展轴回零

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoHoming(servoId,mode,searchVel,latchVel)</code>
描述	设置485扩展轴回零
必选参数	<ul style="list-style-type: none"><code>servoId</code>：伺服驱动器ID，范围[1-15],对应从站ID；<code>mode</code>：回零模式，1-当前位置回零；2-负限位回零；3-正限位回零；<code>searchVel</code>：回零速度，mm/s或°/s；<code>latchVel</code>：箴位速度，mm/s或°/s；
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.8.1. 代码示例

参考设置485扩展轴使能/去使能的代码示例

13.9. 清除485扩展轴错误信息

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServoClearError(servoId)</code>
描述	清除485扩展轴错误信息
必选参数	<ul style="list-style-type: none"><code>servoId</code>：伺服驱动器ID，范围[1-15],对应从站ID；
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.9.1. 代码示例

参考设置485扩展轴参数的代码示例



latest



13.10. 获取485扩展轴伺服状态

在 python 版本加入: SDK-v2.0.3

bit1:0-未运动；1-正在运动; bit2 0-正限位未触发；1-正限位触发； bit3 0-负限位未触发；1-负限位触发

13.10.1. 代码示例

参考设置485扩展轴使能/去使能的代码示例

13.11. 设置状态反馈中485扩展轴数据轴号-暂未开放

在 python 版本加入: SDK-v2.0.3

原型	<code>AuxServosetStatusID(servoId)</code>
描述	设置状态反馈中485扩展轴数据轴号
必选参数	<ul style="list-style-type: none"><code>servoId</code>： 伺服驱动器ID， 范围[1-15],对应从站ID；
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.12. 设置485扩展轴运动加减速速度

在 python 版本加入: SDK-v2.0.5

原型	<code>AuxServoSetAcc(acc, dec)</code>
描述	设置485扩展轴运动加减速速度
必选参数	<ul style="list-style-type: none"><code>acc</code>： 485扩展轴运动加速度<code>dec</code>： 485扩展轴运动减速度

默认参数	无
返回值	错误码 成功-0 失败- errcode

13.13. 设置485扩展轴急停加减速速度

在 python 版本加入: SDK-v2.0.5

原型	<code>AuxServoSetEmergencyStopAcc(acc, dec)</code>
描述	设置485扩展轴急停加减速速度
必选参数	<ul style="list-style-type: none"><code>acc</code> : 485扩展轴急停加速度<code>dec</code> : 485扩展轴急停减速度
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.14. 获取485扩展轴急停加减速速度

在 python 版本加入: SDK-v2.0.5

原型	<code>AuxServoGetEmergencyStopAcc()</code>
描述	获取485扩展轴急停加减速速度
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode<code>acc</code> : 485扩展轴急停加速度<code>dec</code> : 485扩展轴急停减速度

13.15. 获取485扩展轴运动加减速速度

在 python 版本加入: SDK-v2.0.5

原型	<code>AuxServoGetAcc()</code>
描述	获取485扩展轴运动加减速速度
必选参数	无
默认参数	无

返回值	<ul style="list-style-type: none">• 错误码 成功-0 失败- errcode• <code>acc</code> : 485扩展轴运动加速度• <code>dec</code> : 485扩展轴运动减速度
-----	--

13.16. UDP扩展轴通讯参数配置

在 python 版本加入: SDK-v2.0.4

原型	<code>ExtDevSetUDPComParam(ip, port, period, lossPkgTime, lossPkgNum, disconnectTime, reconnectEnable, reconnectPeriod, reconnectNum)</code>
描述	UDP扩展轴通讯参数配置
必选参数	<ul style="list-style-type: none">• <code>ip</code> : PLC IP地址;• <code>port</code> : 端口号;• <code>period</code> : 通讯周期(ms, 暂不开放);• <code>lossPkgTime</code> : 丢包检测时间(ms);• <code>lossPkgNum</code> : 丢包次数;• <code>disconnectTime</code> : 通讯断开确认时长;• <code>reconnectEnable</code> : 通讯断开自动重连使能 0-不使能 1-使能;• <code>reconnectPeriod</code> : 重连周期间隔(ms);• <code>reconnectNum</code> : 重连次数
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.16.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  #UDP扩展轴通讯参数配置
6  error = robot.ExtDevSetUDPComParam('192.168.58.88',2021,2,50,5,50,1,2,5)
7  print("ExtDevSetUDPComParam return:",error)
8  #UDP扩展轴通讯参数配置
9  error = robot.ExtDevGetUDPComParam()
10 print("ExtDevGetUDPComParam return:",error)
```

13.17. 获取UDP扩展轴通讯参数

在 python 版本加入: SDK-v2.0.4

原型	<code>ExtDevGetUDPComParam()</code>
----	-------------------------------------

描述	获取UDP扩展轴通讯参数
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">• 错误码 成功-0 失败- errcode;• <code>ip</code> : PLC IP地址;• <code>port</code> : 端口号;• <code>period</code> : 通讯周期(ms, 暂不开放);• <code>lossPkgTime</code> : 丢包检测时间(ms);• <code>lossPkgNum</code> : 丢包次数;• <code>disconnectTime</code> : 通讯断开确认时长;• <code>reconnectEnable</code> : 通讯断开自动重连使能 0-不使能 1-使能;• <code>reconnectPeriod</code> : 重连周期间隔(ms);• <code>reconnectNum</code> : 重连次数

13.18. 加载UDP通信

在 *python* 版本加入: SDK-v2.0.4

原型	<code>ExtDevLoadUDPDriver()</code>
描述	加载UDP通信
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.18.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  #加载UDP通信
6  error = robot.ExtDevLoadUDPDriver()
7  print("ExtDevLoadUDPDriver return:",error)
8  #卸载UDP通信
9  error = robot.ExtDevUnloadUDPDriver()
10 print("ExtDevUnloadUDPDriver return:",error)
```

13.19. 卸载UDP通信

在 *python* 版本加入: SDK-v2.0.4

原型	<code>ExtDevUnloadUDPDriver()</code>
描述	卸载UDP通信
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.20. UDP扩展轴通信异常断开后恢复连接

在 *python* 版本加入: SDK-v2.0.4

原型	<code>ExtDevUDPClientComReset()</code>
描述	UDP扩展轴通信异常断开后恢复连接
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.20.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  #UDP扩展轴通信异常断开后恢复连接
6  error = robot.ExtDevUDPClientComReset()
7  print("ExtDevUDPClientComReset return:",error)
8  #UDP扩展轴通信异常断开后关闭通讯
9  error = robot.ExtDevUDPClientComClose()
10 print("ExtDevUDPClientComClose return:",error)
```

13.21. UDP扩展轴通信异常断开后关闭通讯

在 *python* 版本加入: SDK-v2.0.4

原型	<code>ExtDevUDPClientComClose()</code>
描述	UDP扩展轴通信异常断开后关闭通讯
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.22. 设置扩展机器人相对扩展轴位置

在 python 版本加入: SDK-v2.0.4

原型	<code>SetRobotPosToAxis(installType)</code>
描述	设置扩展机器人相对扩展轴位置
必选参数	<ul style="list-style-type: none"><code>installType</code>: 0-机器人安装在外部轴上, 1-机器人安装在外部轴外;
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.22.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  #设置扩展机器人相对扩展轴位置
6  error = robot.SetRobotPosToAxis(1)
7  print("SetRobotPosToAxis return:",error)
8  #设置扩展轴系统DH参数配置
9  error = robot.SetAxisDHParaConfig(4,128.5,206.4,0,0,0,0,0,0,)
10 print("SetAxisDHParaConfig return:",error)
11 #UDP扩展轴参数配置
12 error = robot.ExtAxisParamConfig(1,1,0,1000,-1000,1000,1000,1.905,262144,
200,1,1,0)
13 print("ExtAxisParamConfig return:",error)
```

13.23. 设置扩展轴系统DH参数配置

在 python 版本加入: SDK-v2.0.4

`xisDHParaConfig(axisConfig,axisDHd1,axisDHd2,axisDHd3,axisDHd4,axisDHa1, axisDHa2,axisDHa3,axisDHa4)`

扩展轴系统DH参数配置

`xisConfig`: 外部轴构型, 0-单自由度直线滑轨, 1-两自由度L型变位机, 2-三自由度, 3-四自由度, 4

`xisDHd1`: 外部轴DH参数d1 mm;

`xisDHd2`: 外部轴DH参数d2 mm;

`xisDHd3`: 外部轴DH参数d3 mm;

`xisDHd4`: 外部轴DH参数d4 mm;

`xisDHa1`: 外部轴DH参数a1 mm;

`xisDHa2`: 外部轴DH参数a2 mm;

`xisDHa3`: 外部轴DH参数a3 mm;

`xisDHa4`: 外部轴DH参数a4 mm;

  latest 

马 成功-0 失败- errcode

13.24. UDP扩展轴参数配置

在 python 版本加入: SDK-v2.0.4

ie, axisDirection, axisMax, axisMin, axisVel, axisAcc,axisLead, encResolution, axis0ffect, axisCompar

; 1-旋转;
-正向; 1-反向;
1;
1;

;
曲偏移量;
; 川; 2-汇川; 3-松下;
|-SV-XD3EA040L-E, 2-禾川-SV-X2EA150A-A, 1-汇川-SV620PT5R4I, 1-松下-MADLN15SG, ;
量; 1-绝对值;

13.25. 设置扩展轴坐标系参考点-四点法

在 python 版本加入: SDK-v2.0.4

原型	ExtAxisSetRefPoint(pointNum)
描述	设置扩展轴坐标系参考点-四点法
必选参数	<ul style="list-style-type: none">pointNum : 点编号[1-4];
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.25.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 #设置扩展轴坐标系参考点-四点法
6 error = robot.ExtAxisSetRefPoint(1)
7 print("ExtAxisComputeECoordSys(1) return:",error)
```

13.26. 计算扩展轴坐标系-四点法

在 python 版本加入: SDK-v2.0.4

原型	<code>ExtAxisComputeECoordSys()</code>
描述	计算扩展轴坐标系-四点法
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">• 错误码 成功-0 失败- errcode;• <code>coord</code>: 扩展轴坐标系值[x,y,z,rx,ry,rz];

13.26.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 #计算扩展轴坐标系-四点法
6 error,coord = robot.ExtAxisComputeECoordSys()
7 print("ExtAxisComputeECoordSys() return:",error,coord)
8 #应用扩展轴坐标系
9 error = robot.ExtAxisActiveECoordSys(1,1,coord,1)
10 print("ExtAxisActiveECoordSys() return:",error)
```

13.27. 应用扩展轴坐标系

在 python 版本加入: SDK-v2.0.4

`ExtAxisActiveECoordSys(applyAxisId,axisCoordNum,coord,calibFlag)`

应用扩展轴坐标系



latest



- `applyAxisId`: 扩展轴编号 bit0-bit3对应扩展轴编号1-4，如应用扩展轴1和3，则是 0b 0000 010
- `axisCoordNum`: 扩展轴坐标系编号；
- `coord`: 坐标系值[x,y,z,rx,ry,rz]；
- `calibFlag`: 标定标志 0-否，1-是；

无

错误码 成功-0 失败- errcode

13.28. 设置标定参考点在变位机末端坐标系下位姿

在 python 版本加入: SDK-v2.0.4

原型	<code>SetRefPointInExAxisEnd(pos)</code>
描述	设置标定参考点在变位机末端坐标系下位姿
必选参数	<ul style="list-style-type: none"> • <code>pos</code>: 位姿值[x,y,z,rx,ry,rz]；
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.28.1. 代码示例

```

1  from fairino import Robot
2  # 与机器人控制器建立连接， 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  #设置标定参考点在变位机末端坐标系下位姿
5  error = robot.SetRefPointInExAxisEnd(desc_pos)
6  print("SetRefPointInExAxisEnd(1) return:",error)

```

13.29. 变位机坐标系参考点设置-四点法

在 python 版本加入: SDK-v2.0.4

原型	<code>PositionorSetRefPoint(pointNum)</code>
描述	变位机坐标系参考点设置-四点法
必选参数	<ul style="list-style-type: none"> • <code>pointNum</code>: 点编号[1-4]；
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.29.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 #变位机坐标系参考点设置-四点法
6 error = robot.SetRefPointInExAxisEnd(desc_pos)
7 print("SetRefPointInExAxisEnd(1) return:",error)
```

13.30. 变位机坐标系计算-四点法

在 python 版本加入: SDK-v2.0.4

原型	<code>PositionorComputeECoordSys()</code>
描述	变位机坐标系计算-四点法
必选参数	无
默认参数	无
返回值	<ul style="list-style-type: none">• 错误码 成功-0 失败- errcode;• <code>coord</code>: 变位机坐标系值[x,y,z,rx,ry,rz];

13.30.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 #变位机坐标系计算-四点法
6 error,coord = robot.PositionorComputeECoordSys()
7 print("PositionorComputeECoordSys() return:",error,coord)
```

13.31. 末端传感器寄存器写入

在 python 版本加入: SDK-v2.0.5

原型	<code>AxleSensorRegWrite(devAddr, regHAddr, regLAddr, regNum, data1, data2, isNoBlock)</code>
描述	末端传感器寄存器写入

必选参数	<ul style="list-style-type: none"><code>devAddr</code>：设备地址编号 0-255<code>regHAddr</code>：寄存器地址高8位<code>regLAddr</code>：寄存器地址低8位<code>regNum</code>：寄存器个数 0-255<code>data1</code>：写入寄存器数值1<code>data2</code>：写入寄存器数值2<code>isNoBlock</code>：0-阻塞；1-非阻塞
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- <code>errcode</code>；

13.32. UDP扩展轴使能

在 *python* 版本加入: SDK-v2.0.4

原型	<code>ExtAxisServo0n(axisID, status)</code>
描述	UDP扩展轴使能
必选参数	<ul style="list-style-type: none"><code>axisID</code>：轴号[1-4]；<code>status</code>：0-去使能；1-使能；
默认参数	无
返回值	错误码 成功-0 失败- <code>errcode</code>

13.32.1. 代码示例

```
1  from fairino import Robot
2  import time
3  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4  robot = Robot.RPC('192.168.58.2')
5  #UDP扩展轴去使能
6  error = robot.ExtAxisServo0n(1,0)
7  print("ExtAxisServo0n return:",error)
8  #UDP扩展轴使能
9  error = robot.ExtAxisServo0n(1,1)
10 print("ExtAxisServo0n return:",error)
11 #UDP扩展轴回零
12 error = robot.ExtAxisSetHoming(1,0,40,40)
13 print("ExtAxisSetHoming return:",error)
14 time.sleep(1)
15 #UDP扩展轴点动开始
16 error = robot.ExtAxisStartJog(1,1,20,20,20)
17 print("ExtAxisStartJog return:",error)
18 time.sleep(1)
19 #UDP扩展轴点动停止
20 error = robot.ExtAxisStopJog(1)
21 print("ExtAxisStopJog return:",error)
```

13.33. UDP扩展轴回零

在 python 版本加入: SDK-v2.0.4

原型	<code>ExtAxisSetHoming(axisID, mode, searchVel, latchVel)</code>
描述	UDP扩展轴回零
必选参数	<ul style="list-style-type: none"><code>axisID</code>: 轴号[1-4];<code>mode</code>: 回零方式 0当前位置回零, 1负限位回零, 2-正限位回零;<code>searchVel</code>: 寻零速度(mm/s);<code>latchVel</code>: 寻零箍位速度(mm/s);
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.34. UDP扩展轴点动开始

在 python 版本加入: SDK-v2.0.4

原型	<code>ExtAxisStartJog(axisID, direction, vel, acc, maxDistance)</code>
描述	UDP扩展轴点动开始

必选参数	<ul style="list-style-type: none"><code>axisID</code>：轴号[1-4];<code>direction</code>：转动方向 0-反向; 1-正向;<code>vel</code>：速度(mm/s);<code>acc</code>：加速度(mm/s);<code>maxDistance</code>：最大点动距离;
默认参数	无
返回值	错误码 成功-0 失败- <code>errcode</code>

13.35. UDP扩展轴点动停止

在 *python* 版本加入: SDK-v2.0.4

原型	<code>ExtAxisStopJog(axisID)</code>
描述	UDP扩展轴点动停止
必选参数	<ul style="list-style-type: none"><code>axisID</code>：轴号[1-4];
默认参数	无
返回值	错误码 成功-0 失败- <code>errcode</code>

13.36. 设置扩展DO

在 *python* 版本加入: SDK-v2.0.4

原型	<code>SetAuxDO(DONum,bOpen,smooth,block)</code>
描述	设置扩展DO
必选参数	<ul style="list-style-type: none"><code>DONum</code>：DO编号;<code>bOpen</code>：开关 True-开,False-关;<code>smooth</code>：是否平滑 True -是, False -否;<code>block</code>：是否阻塞 True -是, False -否;
默认参数	无
返回值	错误码 成功-0 失败- <code>errcode</code>

13.36.1. 代码示例

```
1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  #设置扩展D0
5  error = robot.SetAuxD0(1,True,False,True)
6  print("GetAuxAI",error)
7  #设置扩展A0
8  error = robot.SetAuxA0(1,60,True)
9  print("SetAuxA0",error)
10 #设置扩展DI输入滤波时间
11 error = robot.SetAuxDIFilterTime(10,False)
12 print("SetAuxDIFilterTime",error)
13 #设置扩展AI输入滤波时间
14 error = robot.SetAuxAIFilterTime(10,True)
15 print("SetAuxAIFilterTime",error)
16 #等待扩展DI输入
17 error = robot.WaitAuxDI(0,False,100,False)
18 print("WaitAuxDI",error)
19 #等待扩展AI输入
20 error = robot.WaitAuxAI(0,0,100,500,False)
21 print("WaitAuxAI",error)
22 #获取扩展AI值
23 error = robot.GetAuxAI(0,False)
24 print("GetAuxAI",error)
25 #获取扩展DI值
26 error = robot.GetAuxDI(0,True)
27 print("GetAuxDI",error)
```

13.37. 设置扩展AO

在 python 版本加入: SDK-v2.0.4

原型	SetAuxA0(A0Num,value,block)
描述	设置扩展AO
必选参数	<ul style="list-style-type: none">A0Num : AO编号;value : 模拟量值[0-4095];block : 是否阻塞 True -是, False -否;
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.38. 设置扩展DI输入滤波时间

在 python 版本加入: SDK-v2.0.4

原型	SetAuxDIFilterTime(filterTime)
描述	设置扩展DI输入滤波时间

必选参数	<ul style="list-style-type: none"><code>filterTime</code>：滤波时间(ms);
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.39. 设置扩展AI输入滤波时间

在 python 版本加入: SDK-v2.0.4

原型	<code>SetAuxAIFilterTime(AINum,filterTime)</code>
描述	设置扩展AI输入滤波时间
必选参数	<ul style="list-style-type: none"><code>AINum</code>：AI编号;<code>filterTime</code>：滤波时间(ms);
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.40. 等待扩展DI输入

在 python 版本加入: SDK-v2.0.4

原型	<code>WaitAuxDI(DINum,bOpen,time,errorAlarm)</code>
描述	等待扩展DI输入
必选参数	<ul style="list-style-type: none"><code>DINum</code>：DI编号;<code>bOpen</code>：开关 True-开,False-关;<code>time</code>：最大等待时间(ms);<code>errorAlarm</code>：是否继续运动 True-是,False-否
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.41. 等待扩展AI输入

在 python 版本加入: SDK-v2.0.4

原型	<code>WaitAuxAI(,AINum,sign,value,time,errorAlarm)</code>
描述	等待扩展AI输入



latest



必选参数	<ul style="list-style-type: none"><code>AINum</code>： AI编号；<code>sign</code>： 0-大于； 1-小于；<code>value</code>： AI值；<code>time</code>： 最大等待时间(ms)；<code>errorAlarm</code>： 是否继续运动 True-是,False-否
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.42. 获取扩展DI值

在 python 版本加入: SDK-v2.0.4

原型	<code>GetAuxDI(DINum,isNoBlock)</code>
描述	获取扩展DI值
必选参数	<ul style="list-style-type: none"><code>DINum</code>： DI编号；<code>isNoBlock</code>： 是否阻塞 True-阻塞 false-非阻塞；
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode；<code>isOpen</code>： 0-关； 1-开；

13.43. 获取扩展AI值

在 python 版本加入: SDK-v2.0.4

原型	<code>GetAuxAI(AINum,isNoBlock)</code>
描述	获取扩展AI值
必选参数	<ul style="list-style-type: none"><code>AINum</code>： AI编号；<code>isNoBlock</code>： 是否阻塞 True-阻塞 False-非阻塞
默认参数	无
返回值	<ul style="list-style-type: none">错误码 成功-0 失败- errcode；<code>value</code>： 输入值；

13.44. UDP扩展轴运动

在 python 版本加入: SDK-v2.0.4


原型	<code>ExtAxisMove(pos,ovl)</code>
描述	UDP扩展轴运动
必选参数	<ul style="list-style-type: none"><code>pos=</code> <code>[exaxis[0],exaxis[1],exaxis[2],exaxis[3]]</code> : 目标位置 轴1位置~轴4位置;<code>ovl</code> : 速度百分比
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.44.1. 代码示例

```
1 from fairino import Robot
2 import time
3 # 与机器人控制器建立连接, 连接成功返回一个机器人对象
4 robot = Robot.RPC('192.168.58.2')
5 error,joint_pos = robot.GetActualJointPosDegree()
6 print("GetActualJointPosDegree",error,joint_pos)
7 e_pos =[-10,0,0,0]
8 joint_pos[0] = joint_pos[0]+30
9 #UDP扩展轴异步运动
10 error = robot.ExtAxisMove(e_pos,30)
11 print("ExtAxisMove",error)
12 print("joint_pos",joint_pos)
13 error = robot.MoveJ(joint_pos,0,0,exaxis_pos=e_pos)
14 print("MoveJ",error)
```

13.45. UDP扩展轴与机器人关节运动同步运动

在 python 版本加入: SDK-v2.0.4

原型	<code>ExtAxisSyncMoveJ(joint_pos,desc_pos,tool,user,exaxis_pos, vel=20.0, acc=0.0, ovl= 100.0, blendT=-1.0, offset_flag=0, offset_pos=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0])</code>
描述	UDP扩展轴与机器人关节运动同步运动
选参数	<ul style="list-style-type: none"><code>joint_pos</code> : 目标关节位置, 单位 [°];<code>desc_pos</code> : 目标笛卡尔位姿, 单位 [mm][°]<code>tool</code> : 工具号, [0~14]<code>user</code> : 工件号, [0~14]<code>exaxis_pos</code> : 外部轴 1 位置 ~ 外部轴 4 位
认参数	<ul style="list-style-type: none"><code>vel</code> : 速度百分比, [0~100] 默认20.0;<code>acc</code> : 加速度百分比, [0~100] 暂不开放,默认0.0 ;<code>ovl</code> : 速度缩放因子, [0~100] 默认100.0 ;<code>blendT</code> : [-1.0]-运动到位 (阻塞), [0~500.0]-平滑时间 (非阻塞), 单位 [ms] 默认 10<code>offset_flag</code> : [0]-不偏移, [1]-工件/基坐标系下偏移, [2]-工具坐  latest ▼<code>offset_pos</code> : 位姿偏移量, 单位 [mm][°] 默认[0.0,0.0,0.0,0.0,0.0,0.0] ,

返回值	错误码 成功-0 失败- errcode;
-----	-----------------------

13.45.1. 代码示例

```
1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  #1. 标定并应用机器人工具坐标系, 您可以使用四点法或六点法进行工具坐标系的标定和应用, 涉及工具坐标系
    标定的接口如下:
5  point_num=1
6  id=1
7  coord=[100,200,300,0,0,0,]
8  type=0
9  install=0
10 #1. 设置工具坐标系
11 # robot.SetToolPoint(point_num) #设置工具参考点-六点法
12 # robot.ComputeTool() #计算工具坐标系
13 # robot.SetTcp4RefPoint() #设置工具参考点-四点法
14 # robot.ComputeTcp4() #计算工具坐标系-四点法
15 # robot.SetToolCoord(id, coord,type,install) #设置应用工具坐标系
16 # robot.SetToolList(id, coord,type,install) #设置应用工具坐标系列表
17 #2. 设置UDP通信参数, 并加载UDP通信
18 robot.ExtDevSetUDPComParam("192.168.58.88", 2021, 2, 100, 3, 100, 1, 100, 10);
19 robot.ExtDevLoadUDPDriver();
20 #3. 设置扩展轴参数, 包括扩展轴类型、扩展轴驱动器参数、扩展轴DH参数
21 robot.SetAxisDHParaConfig(4, 200, 200, 0, 0, 0, 0, 0, 0)#单轴变位机及DH参数
22 robot.SetRobotPosToAxis(1); #扩展轴安装位置
23 robot.ExtAxisParamConfig(1, 0, 1, 100, -100, 10, 10, 12, 131072, 0, 1, 0, 0)#伺服
    驱动器参数, 本示例为单轴变位机, 因此只需要设置一个驱动器参数, 若您选择包含多个轴的扩展轴类型, 需要每一个
    轴设置驱动器参数
24 #4. 设置所选的轴使能、回零
25 robot.ExtAxisServoOn(1, 0);
26 robot.ExtAxisSetHoming(1, 0, 20, 3);
27 #5. 进行扩展轴坐标系标定及应用(注意: 变位机和直线滑轨的标定接口不同, 以下时变位机的标定接口)
28 pos = [0,0,0,0,0,0] #输入您的标定坐标
29 robot.SetRefPointInExAxisEnd(pos)
30 robot.PositionorSetRefPoint(1)#您需要通过四个不同位置的点来标定扩展轴, 因此需要调用此接口4
    次才能完成标定
31 error,coord = robot.PositionorComputeECoordSys()#计算扩展轴标定结果
32 robot.ExtAxisActiveECoordSys(1, 1, coord, 1); #将标定结果应用到扩展轴坐标系
33 method=1
34 #6. 在扩展轴上标定工件坐标系, 您需要用到以下接口
35 # robot.SetWObjCoordPoint( point_num)
36 # error,coord=robot.ComputeWObjCoord( method)
37 # robot.SetWObjCoord(id,coord)
38 # robot.SetWObjList(id, coord)
39 #7. 记录您的同步关节运动起始点
40 startdescPose = [0,0,0,0,0,0]#输入您的坐标
41 startjointPos = [0,0,0,0,0,0]#输入您的坐标
42 startexaxisPos = [0,0,0,0,]#输入您的坐标
43 #8. 记录您的同步关节运动终点坐标
44 enddescPose = [0,0,0,0,0,0]#输入您的坐标
45 endjointPos = [0,0,0,0,0,0]#输入您的坐标
46 endexaxisPos = [0,0,0,0,]#输入您的坐标
47 #9. 编写同步运动程序
48 #运动到起始点, 假设应用的工具坐标系、工件坐标系都是1
49 robot.ExtAxisMove(startexaxisPos, 20);
50 robot.MoveJ(startjointPos, 1, 1,
desc_pos=startdescPose,exaxis_pos=startexaxisPos);
51 #开始同步运动
52 robot.ExtAxisSyncMoveJ(endjointPos, enddescPose, 1, 1, endexaxisPc
```

13.46. UDP扩展轴与机器人直线运动同步运动

在 python 版本加入: SDK-v2.0.4


原型	<code>ExtAxisSyncMoveL(self, joint_pos,desc_pos, tool, user, exaxis_pos, vel=20.0, acc=0.0, ovl=100.0, blendR=-1.0, search=0, offset_flag=0, offset_pos=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0])</code>
描述	UDP扩展轴与机器人直线运动同步运动
选参数	<ul style="list-style-type: none"><code>joint_pos</code>: 目标关节位置, 单位 [°];<code>desc_pos</code>: 目标笛卡尔位姿, 单位 [mm][°];<code>tool</code>: 工具号, [0~14];<code>user</code>: 工件号, [0~14];<code>exaxis_pos</code>: 外部轴 1 位置 ~ 外部轴 4 位;
认参数	<ul style="list-style-type: none"><code>vel</code>: 速度百分比, [0~100] 默认20.0;<code>acc</code>: 加速度百分比, [0~100] 暂不开放,默认0.0;<code>ovl</code>: 速度缩放因子, [0~100] 默认100.0;<code>blendR</code>: [-1.0]-运动到位 (阻塞), [0~500.0]-平滑时间 (非阻塞), 单位 [ms] 默认-1.0<code>search</code>: [0]-不焊丝寻位, [1]-焊丝寻位;<code>offset_flag</code>: [0]-不偏移, [1]-工件/基坐标系下偏移, [2]-工具坐标系下偏移 默认 0;<code>offset_pos</code>: 位姿偏移量, 单位 [mm][°] 默认[0.0,0.0,0.0,0.0,0.0,0.0];
返回值	错误码 成功-0 失败- errcode;

13.46.1. 代码示例

```
1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  robot.Mode(0)
5  time.sleep(1)
6  e_pos =[-20,0,0,0]
7  joint_pos0 = [114.089,-85.740, 119.106,-129.884,-91.655, 79.642]
8  desc_pos0= [-87.920,-178.539,-64.513,-175.471,7.664,139.650]
9  #UDP扩展轴与机器人直线运动同步运动
10 error = robot.ExtAxisSyncMoveL(joint_pos0,desc_pos0,1,1,e_pos)
11 print("ExtAxisSyncMoveL",error)
```

13.47. UDP扩展轴与机器人圆弧运动同步运动

在 python 版本加入: SDK-v2.0.4

<code>ExtAxisSyncMoveC(joint_pos_p, desc_pos_p, tool_p, user_p,exaxis_pos_p, joint_</code> <code>user_t,exaxis_pos_t,vel_p=20.0, acc_p=100.0, offset_flag_p=0, offset_pos_p =[</code> <code>0.0], vel_t=20.0, acc_t=100.0, offset_flag_t=0, offset_pos_t=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0], ovl=</code> <code>blendR=-1.0)</code>	 latest ▼
---	--

- `joint_pos_p` : 路径点关节位置, 单位 [°];
 - `desc_pos_p` : 路径点笛卡尔位姿, 单位 [mm][°];
 - `tool_p` : 路径点工具号, [0~14];
 - `user_p` : 路径点工件号, [0~14];
 - `exaxis_pos_p` : 路径点外部轴 1 位置 ~ 外部轴 4 位置 默认[0.0,0.0,0.0,0.0];
 - `joint_pos_t` : 目标点关节位置, 单位 [°];
 - `desc_pos_t` : 目标点笛卡尔位姿, 单位 [mm][°];
 - `tool_t` : 工具号, [0~14];
 - `user_t` : 工件号, [0~14];
 - `exaxis_pos_t` : 目标点外部轴 1 位置 ~ 外部轴 4 位置 默认[0.0,0.0,0.0,0.0];
-
- `vel_p` : 路径点速度百分比, [0~100] 默认20.0;
 - `acc_p` : 路径点加速度百分比, [0~100] 暂不开放,默认0.0 ;
 - `offset_flag_p` : 路径点是否偏移[0]-不偏移, [1]-工件/基坐标系下偏移, [2]-工具坐标系下偏移
 - `offset_pos_p` : 路径点位姿偏移量, 单位 [mm][°] 默认[0.0,0.0,0.0,0.0,0.0,0.0];
 - `vel_t` : 目标点速度百分比, [0~100] 默认20.0;
 - `acc_t` : 目标点加速度百分比, [0~100] 暂不开放 默认0.0;
 - `offset_flag_t` : 目标点是否偏移[0]-不偏移, [1]-工件/基坐标系下偏移, [2]-工具坐标系下偏移
 - `offset_pos_t` : 目标点位姿偏移量, 单位 [mm][°] 默认[0.0,0.0,0.0,0.0,0.0,0.0];
 - `ovl` : 速度缩放因子, [0~100] 默认100.0;
 - `blendR` : [-1.0]-运动到位 (阻塞), [0~1000]-平滑半径 (非阻塞), 单位 [mm] 默认-1.0;

错误码 成功-0 失败- errcode;

13.47.1. 代码示例

```

1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  robot.Mode(0)
5  time.sleep(1)
6  desc_pos_mid = [-131.2748107910156, -60.21242523193359, -22.55266761779785,
175.9907989501953, 5.92541742324829, 145.5211791992187]
7  desc_pos_end = [-91.3530502319336, -174.5040588378906, -64.93866729736328,
177.1370544433593, 15.96347618103027, 136.1746368408203]
8  joint_pos_mid = [120.9549040841584, -109.8869943146658, 134.1448068146658,
-126.2150709699876, -88.6738087871287, 79.6419593131188]
9  joint_pos_end = [110.1896078279703, -89.01601659189356, 125.5532806698638,
-139.7967831451114, -82.93198387221534, 79.6452225788985]
10 # #UDP扩展轴与机器人圆弧运动同步运动
11 time.sleep(3)
12 error = robot.ExtAxisSyncMoveC(joint_pos_mid,desc_pos_mid,1,1,
[-10,0,0,0],joint_pos_end,desc_pos_end,1,1,[-20,0,0,0])
13 print("ExtAxisSyncMoveC",error)

```

13.48. 可移动装置控制

在 python 版本加入: SDK-v2.0.5

13.48.1. 可移动装置使能

原型	<code>TractorEnable(enable)</code>
描述	可移动装置使能
必选参数	<ul style="list-style-type: none"><code>enable</code> : 使能状态, 0-去使能, 1-使能
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.48.2. 可移动装置回零

原型	<code>TractorHoming()</code>
描述	可移动装置回零
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.48.3. 可移动装置直线运动

原型	<code>TractorMoveL(distance, vel)</code>
描述	可移动装置直线运动
必选参数	<ul style="list-style-type: none"><code>distance</code> : 直线运动距离 (mm)<code>vel</code> : 直线运动速度百分比 (0-100)
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.48.4. 可移动装置圆弧运动

原型	<code>TractorMoveC(radius, angle, vel)</code>
描述	可移动装置圆弧运动
必选参数	<ul style="list-style-type: none"><code>radius</code> : 圆弧运动半径 (mm)<code>angle</code> : 圆弧运动角度 (°)<code>vel</code> : 圆弧运动速度百分比 (0-100)

默认参数	无
返回值	错误码 成功-0 失败- errcode

13.48.5. 可移动装置停止运动

原型	<code>ProgramStop()</code>
描述	可移动装置停止运动
必选参数	无
默认参数	无
返回值	错误码 成功-0 失败- errcode

13.48.6. 代码示例

```

1  from fairino import Robot
2  # 与机器人控制器建立连接, 连接成功返回一个机器人对象
3  robot = Robot.RPC('192.168.58.2')
4  robot.ExtDevSetUDPComParam("192.168.58.2", 2021, 2, 50, 5, 50, 1, 50, 10)
5  robot.ExtDevLoadUDPDriver()
6  robot.ExtAxisParamConfig(1, 0, 0, 50000, -50000, 1000, 1000, 6.280, 16384, 200,
0, 0, 0)
7  robot.ExtAxisParamConfig(2, 0, 0, 50000, -50000, 1000, 1000, 6.280, 16384, 200,
0, 0, 0)
8  robot.SetAxisDHParaConfig(5, 0, 0, 0, 0, 0, 0, 0, 0, 0)
9
10 robot.TractorEnable(False)
11 time.sleep(2)
12 robot.TractorEnable(True)
13 time.sleep(2)
14 robot.TractorHoming()
15 time.sleep(2)
16 robot.TractorMoveL(100, 20)
17 time.sleep(5)
18 robot.TractorMoveL(-100, 20)
19 time.sleep(5)
20 robot.TractorMoveC(300, 90, 20)
21 time.sleep(4)
22 error = robot.TractorStop()
23 print("TractorStop return ", error)

```

