[伶](https://fairino-doc-zhs.readthedocs.io/latest/index.html) / [SDK](https://fairino-doc-zhs.readthedocs.io/latest/SDKManual/index.html) / [Python](https://fairino-doc-zhs.readthedocs.io/latest/SDKManual/python_intro.html) / 10. 机器⼈⼒控



**10.** 机器⼈⼒控

**10.1.** 获取⼒传感器配置











20-ATI 传感器，21-中科⽶点，22-伟航敏芯;device 设备号，坤维 (0-KWR75B) ，航天⼗—院 (0-M

C

**10.1.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( I192.168.58.2I)  company = 17 *#传感器⼚商，17-坤维科技*  device = 0 *#传感器设备号*  error = robot.FT\_SetConfig(company, device) *#配置⼒传感器* print("配置⼒传感器错误码",error)  config = robot.FT\_GetConfig() *#获取⼒传感器配置信息* print( I获取⼒传感器配置信息 I,config)  time.sleep(1)  error = robot.FT\_Activate(0) *#传感器复位* print("传感器复位错误码",error)  time.sleep(1)  error = robot.FT\_Activate(1) *#传感器激活* print("传感器激活错误码",error)  time.sleep(1)  error = robot.SetLoadWeight(0.0) *#末端负载设置为零*  print("末端负载设置为零错误码",error)  time.sleep(1)  error = robot.SetLoadCoord(0.0,0.0,0.0) *#末端负载质⼼设置为零* print("末端质⼼设置为零错误码",error)  time.sleep(1)  error = robot.FT\_SetZero(0) *#传感器去除零点*  print("传感器去除零点错误码",error)  time.sleep(1)  error = robot.FT\_GetForceTorqueOrigin() *#查询传感器原始数据* print("查询传感器原始数据",error)  error = robot.FT\_SetZero(1) print("传感器零点矫正",error) time.sleep(1)  *#传感器零点矫正,注意此时末端不能安装⼯具，* *只有⼒传感器*  latest  error = robot.FT\_GetForceTorqueRCS() *#查询传感器坐标系下数据* print("查询传感器坐标系下数据",error) |



**10.2.** ⼒传感器配置

*在* *Python 版本发⽣变更:* SDK-v2.0.8-3.7.8







-ATI传感器，21-中科⽶点，22-伟航敏芯，23-NBIT ，24-鑫精诚(XJC) ，26-NSR；

6A-200-4) ，ATI(0-AXIA80-M8) ， 中科⽶点(0-MST2010) ，伟航敏芯(0-WHC6L-YB-10A) ，NBIT(0-





**10.3.** ⼒传感器激活

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_Activate(state) | | |
|  |  |  |
| 描述 | ⼒传感器激活 | | | | |
| 必选参数 | . state ：0-复位，1-激活 | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**10.4.** ⼒传感器校零

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_SetZero(state) | | |
|  | | | |
| 描述 | ⼒传感器校零 | | | | |
| 必选参数 |  | |  | ：0-去除零点，1-零点矫正 | |
| state |
|  |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**10.5.** 设置⼒传感器参考坐标系

*在* *python 版本发⽣变更:* SDK-v2.0.5

latest

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 | |  | | --- | | FT\_SetRCS(ref,coord= [0,0,0,0,0,0]) | |  |
| 描述 | 设置⼒传感器参考坐标系 | |

|  |  |
| --- | --- |
| 必选参数 | . ref ：0-⼯具坐标系，1-基坐标系 |
| 默认参数 | . coord ：[x,y,z,rx,ry,rz]⾃定义坐标系值,默认[0,0,0,0,0,0] |
| 返回值 | 错误码 成功-0 失败- errcode |

**10.5.1.** 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—个机器⼈对象* 4  robot = Robot.RPC( I192.168.58.2I)  5  *#负载辨识，* *此时末端安装要辨识的⼯具，* *⼯具安装在⼒传感器下⽅,末端竖直向下*  6  error = robot.FT\_SetRCS(0) *#设置参考坐标系为⼯具坐标系，0-⼯具坐标系，1-基坐标系* 7  print( I设置参考坐标系错误码 I,error)  8  time.sleep(1)  9  tool\_id = 10 *#传感器坐标系编号*  10  tool\_coord = [0.0,0.0,35.0,0.0,0.0,0.0] *# 传感器相对末端法兰位姿* 11  tool\_type = 1 *# 0-⼯具，1-传感器*  12  tool\_install = 0 *# 0-安装末端，1-机器⼈外部*  13  error = robot.SetToolCoord(tool\_id,tool\_coord,tool\_type,tool\_install) *#设置传*  *感器坐标系，* *传感器相对末端法兰位姿*  14  print( I设置传感器坐标系错误码 I,error)  15  time.sleep(1)  16  error = robot.FT\_PdIdenRecord(tool\_id) *#记录辨识数据* 17  print( I记录负载重量错误码 I,error)  18  time.sleep(1)  19  error = robot.FT\_PdIdenRecord() *#计算负载重量，* *单位kg* 20  print( I计算负载重量错误码 I,error)  21  *#负载质⼼辨识，* *机器⼈需要示教三个不同的姿态，* *然后记录辨识数据，* *最后计算负载质⼼* 22  robot.Mode(1)  23  ret = robot.DragTeachSwitch(1) *#机器⼈切⼊拖动示教模式，* *必须在⼿动模式下才能切⼊拖动示教* *模式*  24  time.sleep(5)  25  ret = robot.DragTeachSwitch(0)  26  time.sleep(1)  27  error = robot.FT\_PdCogIdenRecord(tool\_id,1)  28  print( I负载质⼼1错误码 I,error)*#记录辨识数据*  29  ret = robot.DragTeachSwitch(1) *#机器⼈切⼊拖动示教模式，* *必须在⼿动模式下才能切⼊拖动示教* *模式*  30  time.sleep(5)  31  ret = robot.DragTeachSwitch(0)  32  time.sleep(1)  33  error = robot.FT\_PdCogIdenRecord(tool\_id,2)  34  print( I负载质⼼2错误码 I,error)  35  ret = robot.DragTeachSwitch(1) *#机器⼈切⼊拖动示教模式，* *必须在⼿动模式下才能切⼊拖动示教* *模式*  36  time.sleep(5)  37  ret = robot.DragTeachSwitch(0)  38  time.sleep(1)  39  error = robot.FT\_PdCogIdenRecord(tool\_id,3)  40  print( I负载质⼼3错误码 I,error)  41  time.sleep(1)  42  error = robot.FT\_PdCogIdenCompute()  43  print( I负载质⼼计算错误码 I,error)  latest |

**10.6.** 负载重量辨识计算

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_PdIdenCompute() | | |
|  | | | |
| 描述 | 负载重量辨识计算 | | | | |
| 必选参数 | ⽆ | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 | | | 成功-0 失败- errcode ：负载重量，单位 kg | |
| weight | | |

**10.7.** 负载重量辨识记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_PdIdenRecord(tool\_id) | | |
|  |  |  |
| 描述 | 负载重量辨识记录 | | | | |
| 必选参数 | . tool\_id ：传感器坐标系编号，范围[0~14] | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**10.8.** 负载质⼼辨识计算

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_PdCogIdenCompute() | | |
|  |  |  |
| 描述 | 负载质⼼辨识计算 | | | | |
| 必选参数 | ⽆ | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |
|  | | cog= | | |
| [cogx,cogy,cogz]  ：负载质⼼ ，单位 mm | | | | |

**10.9.** 负载质⼼辨识记录

 latest 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |  |
| FT\_PdCogIdenRecord(tool\_id,index) | | | |
|  |  |  |  |
| 描述 | 负载质⼼辨识记录 | | | | | |
| 必选参数 |  | |  | | ：传感器坐标系编号，范围[0~14]; 点编号[1~3] | |
| tool\_id | |
| index | : |
|  |
| 默认参数 | ⽆ | | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | | |

**10.10.** 获取参考坐标系下⼒**/**扭矩数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| FT\_GetForceTorqueRCS() | |
|  |  |
| 描述 | 获取参考坐标系下⼒/扭矩数据 | | | |
| 必选参数 | ⽆ | | | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |
|  | | |  | | --- | | data= | | |
| [fx,fy,fz,tx,ty,tz] ：参考坐标系下⼒/扭矩数据 | | | |

**10.10.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5 | **import frrpc**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = frrpc.RPC( I192.168.58.2I)  rcs = robot.FT\_GetForceTorqueRCS() *#查询传感器坐标系下数据* print(rcs) |

**10.11.** 获取⼒传感器原始⼒**/**扭矩数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_GetForceTorqueOrigin() | | |
|  |  |  |
| 描述 | 获取⼒传感器原始⼒/扭矩数据 | | | | |
| 必选参数 | ⽆ | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败 | | | - errcode  ：⼒传感器原始⼒/扭矩数据 | |
|  | | |  | | --- | | data= | |
| [fx,fy,fz,tx,ty,tz] | | |

**10.11.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5 | **import frrpc**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = frrpc.RPC( I192.168.58.2I)  origin = robot.FT\_GetForceTorqueOrigin() *#查询传感器原始数据* print(origin) |

**10.12.** 碰撞守护

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_Guard(flag,sensor\_num,select,force\_torque,max\_threshold,min\_threshold) | | |
|  |  |  |
| 描述 | 碰撞守护 | | | | |
| 必选参数 | . flag ：0-关闭碰撞守护，1-开启碰撞守护；  . sensor\_num ：⼒传感器编号；  . select ：六个⾃由度是否检测碰撞[fx,fy,fz,mx,my,mz] ，0-不⽣效，1-⽣效；  . force\_torque ：碰撞检测⼒/⼒矩，单位N或Nm；  . max\_threshold ：最⼤阈值；  . min\_threshold ：最⼩阈值；  ⼒/⼒矩检测范围:(force\_torque-min\_threshold,force\_torque+max\_threshold) | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**10.12.1.** 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—个机器⼈对象*  3  robot = Robot.RPC( I192.168.58.2I)  4  *#碰撞守护*  5  actFlag = 1 *#开启标志，0-关闭碰撞守护，1-开启碰撞守护* 6  sensor\_num = 1 *#⼒传感器编号*  7  is\_select = [1,1,1,1,1,1] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效*  8  force\_torque = [0.0,0.0,0.0,0.0,0.0,0.0] *#碰撞检测⼒和⼒矩，* *检测范围（force\_torque- min\_threshold,force\_torque+max\_threshold）*  9  max\_threshold = [10.0,10.0,10.0,10.0,10.0,10.0] *#最⼤阈值* 10  min\_threshold = [5.0,5.0,5.0,5.0,5.0,5.0] *#最⼩阈值*  11  P1= [-160.619,-586.138,384.988,-170.166,-44.782,169.295]  12  P2= [-87.615,-606.209,556.119,-102.495,10.118,178.985]  13  P3= [41.479,-557.243,484.407,-125.174,46.995,-132.165]  14  error = robot.FT\_Guard(actFlag, sensor\_num, is\_select, force\_torque, max\_threshold, min\_threshold) *#开启碰撞守护*  15  print("开启碰撞守护错误码",error)  16  error = robot.MoveL(P1,1,0) *#笛卡尔空间直线运动*  17  print("笛卡尔空间直线运动错误码",error)  18  error = robot.MoveL(P2,1,0)  19  print("笛卡尔空间直线运动错误码",error)  20  error = robot.MoveL(P3,1,0)  21  print("笛卡尔空间直线运动错误码",error) 22  actFlag = 0  23  error = robot.FT\_Guard(actFlag, sensor\_num, is\_select, force\_torque, max\_threshold, min\_threshold) *#关闭碰撞守护*  24  print("关闭碰撞守护错误码",error) |

**10.13.** 恒⼒控制

 latest 

|  |  |  |
| --- | --- | --- |
| 原型 |  |  |
| FT\_Control(flag,sensor\_num,select,force\_torque,gain,adj\_sign,ILC\_sign,max\_dis,max\_ang) |
|  |

|  |  |
| --- | --- |
| 描述 | 恒⼒控制 |
| 必选参数 | . flag ：恒⼒控制开启标志，0-关，1-开；  . sensor\_num ：⼒传感器编号；  . select ：六个⾃由度是否检测 [fx,fy,fz,mx,my,mz] ，0-不⽣效，1-⽣效；  . force\_torque ：检测⼒/⼒矩，单位N或Nm；  . gain ：[f\_p,f\_i,f\_d,m\_p,m\_i,m\_d],⼒PID参数，⼒矩PID参数；  . adj\_sign ： ⾃适应启停状态，0-关闭，1-开启；  . ILC\_sign : ILC控制启停状态，0-停⽌ ，1-训练，2-实操；  . max\_dis ：最⼤调整距离；  . max\_ang ：最⼤调整角度； |
| 默认参数 | ⽆ |
| 返回值 | 错误码 成功-0 失败- errcode |

**10.13.1.** 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—个机器⼈对象* 3  robot = Robot.RPC( I192.168.58.2I)  4  *#恒⼒控制*  5  status = 1 *#恒⼒控制开启标志，0-关，1-开* 6  sensor\_num = 1 *#⼒传感器编号*  7  is\_select = [0,0,1,0,0,0] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效* 8  force\_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]  9  gain = [0.0005,0.0,0.0,0.0,0.0,0.0] *#⼒PID参数，* *⼒矩PID参数* 10  adj\_sign = 0 *#⾃适应启停状态，0-关闭，1-开启*  11  ILC\_sign = 0 *#ILC控制启停状态，0-停⽌* *，1-训练，2-实操*  12  max\_dis = 100.0 *#最⼤调整距离* 13  max\_ang = 0.0 *#最⼤调整角度*  14  J1= [70.395, -46.976, 90.712, -133.442, -87.076, -27.138]  15  P2= [-123.978, -674.129, 44.308, -178.921, 2.734, -172.449]  16  P3= [123.978, -674.129, 42.308, -178.921, 2.734, -172.449]  17  error = robot.MoveJ(J1,1,0)  18  print("关节空间运动指令错误码",error)  19  error = robot.MoveL(P2,1,0)  20  print("笛卡尔空间直线运动指令错误码",error)  21  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  22  print("恒⼒控制开启错误码",error)  23  error = robot.MoveL(P3,1,0) *#笛卡尔空间直线运动* 24  print("笛卡尔空间直线运动指令错误码",error)  25  status = 0  26  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  27  print("恒⼒控制结束错误码",error) |

**10.14.** 螺旋线探索

latest 

|  |  |  |
| --- | --- | --- |
| 原型 |  | FT\_SpiralSearch(rcs, ft, dr=0.7, max\_t\_ms=60000 max\_vel=5) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 描述 | 螺旋线探索 | | | | |
| 必选参数 | . rcs ：参考坐标系，0-⼯具坐标系，1-基坐标系  . ft ：⼒或⼒矩阈值 (0~100) ，单位 N 或 Nm; | | | | |
| 默认参数 |  | dr ：每圈半径进给量，单位 mm 默认0.7;  max\_t\_ms ： 最⼤探索时间，单位 ms 默认 60000; | | | |
| max\_vel | | | ：线速度最⼤值，单位 mm/s 默认 5 |
|  |  |  |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**10.14.1.** 代码示例

|  |
| --- |
| P = [36.794,-675.119, 65.379, -176.938, 2.535, -179.829]  **from fairino import** Robot  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( I192.168.58.2I)  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  *#恒⼒参数*  status = 1 *#恒⼒控制开启标志，0-关，1-开*  sensor\_num = 1 *#⼒传感器编号*  is\_select = [0,0,1,0,0,0] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效* force\_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]  gain = [0.0001,0.0,0.0,0.0,0.0,0.0] *#⼒PID参数，* *⼒矩PID参数* adj\_sign = 0 *#⾃适应启停状态，0-关闭，1-开启*  ILC\_sign = 0 *#ILC控制启停状态，0-停⽌* *，1-训练，2-实操* max\_dis = 100.0 *#最⼤调整距离*  max\_ang = 5.0 *#最⼤调整角度*  *#螺旋线探索参数*  rcs = 0 *#参考坐标系，0-⼯具坐标系，1-基坐标系* fFinish = 10 *#⼒或⼒矩阈值（0~100），* *单位N或Nm*  error = robot.MoveL(P,1,0) *#笛卡尔空间直线运动⾄初始点* print("笛卡尔空间直线运动⾄初始点",error)  is\_select = [0,0,1,1,1,0] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效*  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign, max\_dis,max\_ang)  print("恒⼒控制开启错误码",error)  22  23  24  25  26  error = robot.FT\_SpiralSearch(rcs,fFinish,max\_vel=3)  print("螺旋线探索错误码",error)  status = 0  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign, max\_dis,max\_ang)  27  print("恒⼒控制关闭错误码",error) |

**10.15.** 旋转插⼊

 latest

|  |  |
| --- | --- |
| 原型 | FT\_RotInsertion(rcs, ft, orn, angVelRot=3, angleMax=45, angAccmax=0, rotorn=1) |
| 描述 | 旋转插⼊ |
| 必选参数 | . rcs ：参考坐标系，0-⼯具坐标系，1-基坐标系；  . ft ：⼒或⼒矩阈值 (0~100) ，单位 N 或 Nm;  . orn ：⼒/扭矩⽅向，1-沿z轴⽅向，2-绕z轴⽅向; |

|  |  |
| --- | --- |
| 默认参数 | . angVelRot ：旋转角速度，单位°/s,默认 3;  . angleMax ：最⼤旋转角度，单位 ° 默认 45;  . angAccmax ：最⼤旋转加速度，单位 °/s^2 ，暂不使⽤ 默认0;  . rotorn ：旋转⽅向，1-顺时针，2-逆时针 默认1 |
| 返回值 | 错误码 成功-0 失败- errcode |

**10.15.1.** 代码示例

|  |
| --- |
| P = [36.794,-675.119, 65.379, -176.938, 2.535, -179.829]  **from fairino import** Robot  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( I192.168.58.2I)  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  *#恒⼒参数*  status = 1 *#恒⼒控制开启标志，0-关，1-开*  sensor\_num = 1 *#⼒传感器编号*  is\_select = [0,0,1,0,0,0] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效* force\_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]  gain = [0.0001,0.0,0.0,0.0,0.0,0.0] *#⼒PID参数，* *⼒矩PID参数* adj\_sign = 0 *#⾃适应启停状态，0-关闭，1-开启*  ILC\_sign = 0 *#ILC控制启停状态，0-停⽌* *，1-训练，2-实操* max\_dis = 100.0 *#最⼤调整距离*  max\_ang = 5.0 *#最⼤调整角度*  *#旋转插⼊参数*  rcs = 0 *#参考坐标系，0-⼯具坐标系，1-基坐标系*  forceInsertion = 2.0 *#⼒或⼒矩阈值（0~100），* *单位N或Nm* orn = 1 *#⼒的⽅向，1-fz,2-mz*  *#默认参数* *angVelRot：* *旋转角速度，* *单位* *°/s 默认* *3*  *#默认参数* *angleMax：* *最⼤旋转角度，* *单位* *° 默认* *5*  *#默认参数* *angAccmax：* *最⼤旋转加速度，* *单位* *°/s^2，* *暂不使⽤* *默认0*  *#默认参数* *rotorn ：旋转⽅向，1-顺时针，2-逆时针* *默认1*  error = robot.MoveL(P,1,0) *#笛卡尔空间直线运动⾄初始点* print("笛卡尔空间直线运动⾄初始点",error)  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  print("恒⼒控制开启错误码",error)  26  27  28  29  30  error = robot.FT\_RotInsertion(rcs,1,orn)  print("旋转插⼊错误码",error)  status = 0  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  31  print("恒⼒控制关闭错误码",error) |

**10.16.** 直线插⼊

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 | FT\_LinInsertion(rcs, ft, disMax, linorn, lin\_v=1.0, lin\_a=1.0) | | |
| 描述 | 直线插⼊ | | |
| 必选参数 |  | rcs ：参考坐标系，0-⼯具坐标系，1-基坐标系；  latest  ft ：⼒或⼒矩阈值 (0~100) ，单位 N 或 Nm; |  |
| disMax ：最⼤插⼊距离，单位 mm;  linorn ：插⼊⽅向:0-负⽅向，1-正⽅向 | |

|  |  |  |
| --- | --- | --- |
| 默认参数 |  | lin\_v ：直线速度，单位 mm/s 默认1;  lin\_a ：直线加速度，单位 mm/s^2 ，暂不使⽤ 默认1 |
| 返回值 | 错误码 成功-0 失败- errcode | |

**10.16.1.** 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—个机器⼈对象* 3  robot = Robot.RPC( I192.168.58.2I)  4  P = [36.794,-675.119, 65.379, -176.938, 2.535, -179.829] 5  *#恒⼒参数*  6  status = 1 *#恒⼒控制开启标志，0-关，1-开* 7  sensor\_num = 1 *#⼒传感器编号*  8  is\_select = [0,0,1,0,0,0] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效* 9  force\_torque = [0.0,0.0,-10.0,0.0,0.0,0.0]  10  gain = [0.0001,0.0,0.0,0.0,0.0,0.0] *#⼒PID参数，* *⼒矩PID参数* 11  adj\_sign = 0 *#⾃适应启停状态，0-关闭，1-开启*  12  ILC\_sign = 0 *#ILC控制启停状态，0-停⽌* *，1-训练，2-实操*  13  max\_dis = 100.0 *#最⼤调整距离* 14  max\_ang = 5.0 *#最⼤调整角度* 15  *#直线插⼊参数*  16  rcs = 0 *#参考坐标系，0-⼯具坐标系，1-基坐标系*  17  force\_goal = 10.0 *#⼒或⼒矩阈值（0~100），* *单位N或Nm* 18  disMax = 100.0 *#最⼤插⼊距离，* *单位mm*  19  linorn = 1 *#插⼊⽅向，1-正⽅向，2-负⽅向* 20  *#默认参数* *lin\_v ：直线速度，* *单位* *mm/s 默认1*  21  *#默认参数* *lin\_a ：直线加速度，* *单位* *mm/s^2，* *暂不使⽤* *默认0* 22  error = robot.MoveL(P,1,0) *#笛卡尔空间直线运动⾄初始点* 23  print("笛卡尔空间直线运动⾄初始点",error)  24  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  25  print("恒⼒控制开启错误码",error)  26  error = robot.FT\_LinInsertion(rcs,force\_goal,disMax,linorn)  27  print("直线插⼊错误码",error)  28  status = 0  29  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  30  print("恒⼒控制关闭错误码",error) |

**10.17.** 计算中间平⾯位置开始

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 |  |  |  |
| FT\_CalCenterStart() |
|  | |
| 描述 | 计算中间平⾯位置开始 | | |
| 必选参数 | ⽆ | | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |

**10.18.** 计算中间平⾯位置结束

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |
| FT\_CalCenterEnd() | |
|  |  |
| 描述 | 计算中间平⾯位置结束 | | | |
| 必选参数 | ⽆ | | | |
| 默认参数 | ⽆ | | | |
| 返回值 | 错误码 成功-0 失败- errcode -  pos= [x,y,z,rx,ry,rz] ： 中间平⾯位置 | | | |

**10.19.** 表⾯定位

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 原型 | FT\_FindSurface (rcs, dir, axis, disMax, ft, lin\_v=3.0, lin\_a=0.0) | | | |
| 描述 | 表⾯定位 | | | |
| 必选参数 |  |  | | ： 参考坐标系，0-⼯具坐标系，1-基坐标系； ：移动⽅向，1-正⽅向，2-负⽅向； |
| rcs | |
|  |  |
| dir | |
|  |  |
| axis ：移动轴，1-x ，2-y ，3-z；  disMax ：最⼤探索距离，单位 mm;  ft ：动作终⽌⼒阈值，单位N； | | |
| 默认参数 | . lin\_v ：探索直线速度，单位mm/s 默认3;  . lin\_a ：探索直线加速度，单位mm/s^2 默认0; | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | |



**10.19.1.** 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—个机器⼈对象* 4  robot = Robot.RPC( I192.168.58.2I)  5  *#恒⼒控制*  6  status = 1 *#恒⼒控制开启标志，0-关，1-开* 7  sensor\_num = 1 *#⼒传感器编号*  8  is\_select = [1,0,0,0,0,0] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效* 9  force\_torque = [-2.0,0.0,0.0,0.0,0.0,0.0]  10  gain = [0.0002,0.0,0.0,0.0,0.0,0.0] *#⼒PID参数，* *⼒矩PID参数* 11  adj\_sign = 0 *#⾃适应启停状态，0-关闭，1-开启*  12  ILC\_sign = 0 *#ILC控制启停状态，0-停⽌* *，1-训练，2-实操*  13  max\_dis = 15.0 *#最⼤调整距离* 14  max\_ang = 0.0 *#最⼤调整⻆度* 15  *#表⾯定位参数*  16  rcs = 0 *#参考坐标系，0-⼯具坐标系，1-基坐标系* 17  direction = 1 *#移动⽅向，1-正⽅向，2-负⽅向* 18  axis = 1 *#移动轴，1-X,2-Y,3-Z*  19  lin\_v = 3.0 *#探索直线速度，* *单位mm/s*  20  lin\_a = 0.0 *#探索直线加速度，* *单位mm/s^2* 21  disMax = 50.0 *#最⼤探索距离，* *单位mm*  22  force\_goal = 2.0 *#动作终⽌⼒阈值，* *单位N*  23  P1= [-77.24,-679.599,58.328,179.373,-0.028,-167.849] 24  Robot.MoveCart(P1,1,0) *#关节空间点到点运动*  25  *#x⽅向寻找中⼼* 26  *#第1个表⾯*  27  error = robot.FT\_CalCenterStart()  28  print("计算中间平⾯开始错误码",error)  29  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  30  print("恒⼒控制开始错误码",error)  31  error = robot.FT\_FindSurface(rcs,direction,axis,disMax,force\_goal)  32  print("寻找X+表⾯错误码",error)  33  status = 0  34  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  35  print("恒⼒控制结束错误码",error)  36  time.sleep(2)  37  error = robot.MoveCart(P1,1,0) *#关节空间点到点运动*  38  print("关节空间点到点运动错误码",error)  39  time.sleep(5)  40  *#第2个表⾯*  41  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  42  print("恒⼒控制开始错误码",error)  43  direction = 2 *#移动⽅向，1-正⽅向，2-负⽅向*  44  error = robot.FT\_FindSurface(rcs,direction,axis,disMax,force\_goal)  45  print("寻找X—表⾯错误码",error)  46  status = 0  47  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  48  print("恒⼒控制结束错误码",error) 49  *#计算x ⽅向中⼼位置*  50  error,xcenter = robot.FT\_CalCenterEnd()  51  print("计算X⽅向中间平⾯结束错误码",xcenter)  52  error = robot.MoveCart(xcenter,1,0)  53  print("关节空间点到点运动错误码",error)  latest  54  time.sleep(1)  55  *#y⽅向寻找中⼼* 56  *#第1个表⾯* |

|  |
| --- |
| 57  error =robot.FT\_CalCenterStart()  58  print("计算中间平⾯开始错误码",error)  59  error =robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  60  print("恒⼒控制开始错误码",error)  61  direction = 1 *#移动⽅向，1-正⽅向，2-负⽅向* 62  axis = 2 *#移动轴，1-X,2-Y,3-Z*  63  disMax = 150.0 *#最⼤探索距离，* *单位mm* 64  lin\_v = 6.0 *#探索直线速度，* *单位mm/s*  65  error =robot.FT\_FindSurface(rcs,direction,axis,disMax,force\_goal)  66  print("寻找表⾯Y+错误码",error)  67  status = 0  68  error =robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  69  print("恒⼒控制结束错误码",error)  70  error =robot.MoveCart(P1,1,0) *#关节空间点到点运动*  71  print("关节空间点到点运动错误码",error)  72  Robot.WaitMs(1000)  73  *#第2个表⾯*  74  error =robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  75  print("恒⼒控制开始错误码",error)  76  direction = 2 *#移动⽅向，1-正⽅向，2-负⽅向*  77  error =robot.FT\_FindSurface(rcs,direction,axis,disMax,force\_goal)  78  print("寻找表⾯Y-错误码",error)  79  status = 0  80  error =robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  81  print("恒⼒控制结束错误码",error) 82  *#计算y⽅向中⼼位置*  83  error,ycenter=robot.FT\_CalCenterEnd()  84  print("计算中间平⾯Y⽅向结束错误码",ycenter)  85  error =robot.MoveCart(ycenter,1,0)  86  print("关节空间点到点运动错误码",error) |

**10.20.** 柔顺控制关闭

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 |  |  |  |
| FT\_ComplianceStop() |
|  |
| 描述 | 柔顺控制关闭 | | |
| 必选参数 | ⽆ | | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |

**10.21.** 柔顺控制开启

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| FT\_ComplianceStart(p, force) | | |
|  |  |  |
| 描述 | 柔顺控制开启 | | | | |
| 必选参数 | . p : 位置调节系数或柔顺系数  . force ：柔顺开启⼒阈值，单位N | | | | |
| 默认参数 | ⽆ | | | | |



 latest 

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

**10.21.1.** 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—个机器⼈对象* 4  robot = Robot.RPC( I192.168.58.2I)  5  J1= [75.005,-46.434,90.687,-133.708,-90.315,-27.139] 6  P2= [-77.24,-679.599,38.328,179.373,-0.028,-167.849] 7  P3= [77.24,-679.599,38.328,179.373,-0.028,-167.849] 8  *#恒⼒控制参数*  9  status = 1 *#恒⼒控制开启标志，0-关，1-开* 10  sensor\_num = 1 *#⼒传感器编号*  11  is\_select = [1,1,1,0,0,0] *#六个⾃由度选择[fx,fy,fz,mx,my,mz]，0-不⽣效，1-⽣效* 12  force\_torque = [-10.0,-10.0,-10.0,0.0,0.0,0.0]  13  gain = [0.0005,0.0,0.0,0.0,0.0,0.0] *#⼒PID参数，* *⼒矩PID参数* 14  adj\_sign = 0 *#⾃适应启停状态，0-关闭，1-开启*  15  ILC\_sign = 0 *#ILC控制启停状态，0-停⽌* *，1-训练，2-实操*  16  max\_dis = 1000.0 *#最⼤调整距离* 17  max\_ang = 0.0 *#最⼤调整角度*  18  error = robot.MoveJ(J1,1,0)  19  print("关节空间运动到点1错误码",error) 20  *#柔顺控制*  21  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  22  print("恒⼒控制开始错误码",error)  23  p = 0.00005 *#位置调节系数或柔顺系数* 24  force = 30.0 *#柔顺开启⼒阈值，* *单位N*  25  error = robot.FT\_ComplianceStart(p,force)  26  print("柔顺控制开始错误码",error)  27  error = robot.MoveL(P2,1,0,vel =10) *#笛卡尔空间直线运动*  28  print("笛卡尔空间直线运动到点2错误码", error)  29  error = robot.MoveL(P3,1,0,vel =10)  30  print("笛卡尔空间直线运动到点3错误码", error)  31  time.sleep(1)  32  error = robot.FT\_ComplianceStop()  33  print("柔顺控制结束错误码",error)  34  status = 0  35  error = robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,  ILC\_sign,max\_dis,max\_ang)  36  print("恒⼒控制关闭错误码",error) |

**10.22.** 负载辨识滤波初始化

*在* *python 版本加⼊:* SDK-v2.0.1

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 |  |  |  |
| LoadIdentifyDynFilterInit() |
|  |
| 描述 | 负载辨识滤波初始化 | | |
| 必选参数 | ⽆ | | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |



 latest 

**10.22.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | **from fairino import** Robot  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( '192.168.58.2')  *#负载辨识滤波初始化*  error = robot.LoadIdentifyDynFilterInit()  print("LoadIdentifyDynFilterInit:",error)  *#负载辨识变量初始化*  error = robot.LoadIdentifyDynVarInit()  print("LoadIdentifyDynVarInit:",error)  joint\_torque= [0,0,0,0,0,0]  joint\_pos= [0,0,0,0,0,0]  gain= [0,0.05,0,0,0,0,0,0.02,0,0,0,0] t =10  error,joint\_pos=robot.GetActualJointPosDegree(1)  joint\_pos [1] = joint\_pos [1]+10  error,joint\_torque=robot.GetJointTorques(1)  joint\_torque[1] = joint\_torque[1]+2  *#负载辨识主程序*  error = robot.LoadIdentifyMain(joint\_torque, joint\_pos, t)  print("LoadIdentifyMain:",error)  *#获取负载辨识结果*  error = robot.LoadIdentifyGetResult(gain)  print("LoadIdentifyGetResult:",error) |

**10.23.** 负载辨识变量初始化

*在* *python 版本加⼊:* SDK-v2.0.1

|  |  |  |  |
| --- | --- | --- | --- |
| 原型 |  |  |  |
| LoadIdentifyDynVarInit() |
|  |
| 描述 | 负载辨识变量初始化 | | |
| 必选参数 | ⽆ | | |
| 默认参数 | ⽆ | | |
| 返回值 | 错误码 成功-0 失败- errcode | | |

**10.24.** 负载辨识主程序

*在* *python*

*版本加⼊:* SDK-v2.0.1

 latest 



|  |  |
| --- | --- |
| 原型 | LoadIdentifyMain(joint\_torque, joint\_pos, t) |
| 描述 | 负载辨识主程序 |

|  |  |
| --- | --- |
| 必选参数 | . joint\_torque ： 关节扭矩 j1-j6；  . joint\_pos ：关节位置 j1-j6  . t ：采样周期 |
| 默认参数 | ⽆ |
| 返回值 | 错误码 成功-0 失败- errcode |

**10.25.** 获取负载辨识结果

*在* *python 版本加⼊:* SDK-v2.0.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |  |  |
| LoadIdentifyGetResult(gain) | | | | |
|  |  |  |  |  |
| 描述 | 获取负载辨识结果 | | | | | | |
| 必选参数 | . gain ：重⼒项系数double[6] ，离⼼项系数double[6] | | | | | | |
| 默认参数 | ⽆ | | | | | | |
| 返回值 | 错误码 成功-0 失败- errcode . weight ：负载重量  . cog= [x,y,z] ：负载质⼼坐标 | | | | | | |

**10.26.** ⼒传感器辅助拖动

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |
| --- | --- |
| 原型 | ForceAndJointImpedanceStartStop(status, impedanceFlag, lamdeDain, KGain, BGain, dragMaxTcpVel, dragMaxTcpOriVel) |
| 描述 | ⼒传感器辅助拖动 |
| 必选参数 | . status ：控制状态，0-关闭；1-开启  . impedanceFlag ： 阻抗开启标志，0-关闭；1-开启  . lamdeDain ：[D1,D2,D3,D4,D5, D6] 拖动增益  . KGain ：[K1,K2,K3,K4,K5,K6]刚度增益  . BGain ：[B1,B2,B3,B4,B5,B]阻尼增益  . dragMaxTcpVel ：拖动末端最⼤线速度限制  . dragMaxTcpOriVel ：拖动末端最⼤角速度限制 |
| 默认参数 | ⽆ |
| 返回值 | 错误码 成功-0 失败- errcode |

**10.26.1.** 代码示例

|  |
| --- |
| robot = Robot.RPC( I192.168.58.2I)  **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  status = 1 *#控制状态，0-关闭；1-开启*  asaptiveFlag = 1 *#⾃适应开启标志，0-关闭；1-开启*  interfereDragFlag = 1 *#⼲涉区拖动标志，0-关闭；1-开启*  ingularityConstraintsFlag = 0 *#奇异点策略，0-规避；1-穿越*  M = [15, 15, 15, 0.5, 0.5, 0.1] *#惯性系数* B = [150, 150, 150, 5, 5, 1] *#阻尼系数*  K = [0, 0, 0, 0, 0, 0] *#刚度系数*  F = [5, 5, 5, 1, 1, 1] *#拖动六维⼒阈值* Fmax = 50 *#最⼤拖动⼒限制*  Vmax = 1810 *#最⼤关节速度限制*  error = robot.EndForceDragControl(status, asaptiveFlag, interfereDragFlag,  ingularityConstraintsFlag, M, B, K, F, Fmax, Vmax)  print("EndForceDragControl return:",error)  19  20  21  22  23  time.sleep(10)  status=0  error = robot.EndForceDragControl(status, asaptiveFlag, interfereDragFlag,  ingularityConstraintsFlag, M, B, K, F, Fmax, Vmax)  24  print("EndForceDragControl return:",error) |

**10.27.** 报错清除后⼒传感器⾃动开启

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| SetForceSensorDragAutoFlag(status) | | |
|  |  |  |
| 描述 | 报错清除后⼒传感器⾃动开启 | | | | |
| 必选参数 |  | |  | ：控制状态，0-关闭；1-开启 | |
| status |
|  |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**10.27.1.** 代码示例

 latest

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( I192.168.58.2I)  error = robot. SetForceSensorDragAutoFlag (1)  print("SetForceSensorDragAutoFlag return:",error) |

**10.28.** 设置六维⼒和关节阻抗混合拖动开关及参数

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |
| --- | --- |
| 原型 | EndForceDragControl(status, asaptiveFlag, interfereDragFlag, ingularityConstraintsFlag, M, B, K, F, Fmax, Vmax) |
| 描述 | 设置六维⼒和关节阻抗混合拖动开关及参数 |
| 必选参数 | . status ：控制状态，0-关闭；1-开启  . asaptiveFlag ： ⾃适应开启标志，0-关闭；1-开启  . interfereDragFlag ：⼲涉区拖动标志，0-关闭；1-开启  . ingularityConstraintsFlag ：奇异点策略，0-规避；1-穿越 . M= [m1,m2,m3,m4,m5,m6] ：惯性系数  . B= [b1,b2,b3,b4,b5,b6] ： 阻尼系数  . K= [k1,k2,k3,k4,k5,k6] ： 刚度系数  . F= [f1,f2,f3,f4,f5,f6] ：拖动六维⼒阈值  . Fmax ：最⼤拖动⼒限制  . Vmax ：最⼤关节速度限制 |
| 默认参数 | ⽆ |
| 返回值 | 错误码 成功-0 失败- errcode |



**10.28.1.** 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—个机器⼈对象*  4  5  robot = Robot.RPC( '192.168.58.2')  6  7  status = 1 *#控制状态，0-关闭；1-开启*  8  impedanceFlag = 1 *#阻抗开启标志，0-关闭；1-开启*  9  lamdeDain = [ 3.0, 2.0, 2.0, 2.0, 2.0, 3.0] *# 拖动增益* 10  KGain = [0.00, 0.00, 0.00, 0.00, 0.00, 0.00] *# 刚度增益* 11  BGain = [150, 150, 150, 5.0, 5.0, 1.0] *# 阻尼增益*  12  dragMaxTcpVel = 1000 *#拖动末端最⼤线速度限制* 13  dragMaxTcpOriVel = 180 *#拖动末端最⼤角速度限制*  14  15  error = robot.DragTeachSwitch(1)  16  print("DragTeachSwitch 1 return:",error)  17  18  error = robot.ForceAndJointImpedanceStartStop(status, impedanceFlag, lamdeDain,  KGain, BGain,dragMaxTcpVel,dragMaxTcpOriVel)  19  print("ForceAndJointImpedanceStartStop return:",error)  20  21  error = robot.GetForceAndTorqueDragState()  22  print("GetForceAndTorqueDragState return:",error)  23  24  time.sleep(10)  25  26  status = 0 *#控制状态，0-关闭；1-开启*  27  impedanceFlag = 0 *#阻抗开启标志，0-关闭；1-开启*  28  error = robot.ForceAndJointImpedanceStartStop(status, impedanceFlag, lamdeDain,  KGain, BGain,dragMaxTcpVel,dragMaxTcpOriVel)  29  print("ForceAndJointImpedanceStartStop return:",error)  30  31  error = robot.GetForceAndTorqueDragState()  32  print("GetForceAndTorqueDragState return:",error)  33  34  error = robot.DragTeachSwitch(0)  35  print("DragTeachSwitch 0 return:",error) |

**10.29.** 获取⼒传感器拖动开关状态

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |  | |
| GetForceAndTorqueDragState() | | | |
|  |  |  |  |
| 描述 | 获取⼒传感器拖动开关状态 | | | | | | |
| 必选参数 | ⽆ | | | | | | |
| 默认参数 | ⽆ | | | | | | |
| 返回值 | 错误码 成功-0 失败- errcode  . dragState ：⼒传感器辅助拖动控制状态，0-关闭；1-开启 . sixDimensionalDragState ：六维⼒辅助拖动控制状态，0-关  t | | | | | | lates |

**10.30.** 设置⼒传感器下负载重量

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| SetForceSensorPayload(weight) | | |
|  |  |  |
| 描述 | 设置⼒传感器下负载重量 | | | | |
| 必选参数 | . weight ：负载重量 kg | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |

**10.30.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | **from fairino import** Robot  **import time**  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( '192.168.58.2')  error = robot.SetForceSensorPayload(0.8)  print("SetForceSensorPayload return:",error)  error = robot.SetForceSensorPayloadCog(0.5,0.6,12.5)  print("SetForceSensorPayLoadCog return:",error)  error = robot.GetForceSensorPayload()  print("GetForceSensorPayLoad return:",error)  error = robot.GetForceSensorPayloadCog()  print("GetForceSensorPayLoadCog return:",error) |

**10.31.** 设置⼒传感器下负载质⼼

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| SetForceSensorPayloadCog(x,y,z) | | |
|  |  |  |
| 描述 | 设置⼒传感器下负载质⼼ | | | | |
| 必选参数 | · x ：负载质⼼x mm  . y ：负载质⼼y mm  . z ：负载质⼼z mm | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode | | | | |



 latest 

**10.32.** 获取⼒传感器下负载重量

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| GetForceSensorPayload() | | |
|  |  |  |
| 描述 | 获取⼒传感器下负载重量 | | | | |
| 必选参数 | ⽆ | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode . weight ：负载重量 kg | | | | |

**10.33.** 获取⼒传感器下负载质⼼

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| GetForceSensorPayloadCog() | | |
|  | | | |
| 描述 | 获取⼒传感器下负载质⼼ | | | | |
| 必选参数 | ⽆ | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败- errcode . x ：负载质⼼x mm  . y ：负载质⼼y mm  . z ：负载质⼼z mm | | | | |

**10.34.** ⼒传感器⾃动校零

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| ForceSensorAutoComputeLoad() | | |
|  |  |  |
| 描述 | ⼒传感器⾃动校零 | | | | |
| 必选参数 | ⽆ | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 |  | | 错误码 成功-0 失败- errcode weight ：传感器质量 kg  pos= [x,y,z] ：传感器质⼼ mm | | |



 latest 

**10.34.1.** 代码示例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | **from fairino import** Robot  *# 与机器⼈控制器建⽴连接，* *连接成功返回—个机器⼈对象*  robot = Robot.RPC( '192.168.58.2')  error = robot.SetForceSensorPayload(0)  print("SetForceSensorPayload return:",error)  error = robot.SetForceSensorPayloadCog(0,0,0)  print("SetForceSensorPayLoadCog return:",error)  error = robot.ForceSensorAutoComputeLoad()  print("ForceSensorAutoComputeLoad return:",error) |

**10.35.** 传感器⾃动校零数据记录

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |
| ForceSensorSetSaveDataFlag(recordCount) | | |
|  |  |  |
| 描述 | 传感器⾃动校零数据记录 | | | | |
| 必选参数 | . recordCount ：记录数据个数 1-3 | | | | |
| 默认参数 | ⽆ | | | | |
| 返回值 | 错误码 成功-0 失败-errcode | | | | |

**10.36.** 传感器⾃动校零计算

*在* *python 版本加⼊:* SDK-v2.0.5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 原型 |  |  |  |  |  |  |
| ForceSensorComputeLoad() | | | |
|  |  |  |  |
| 描述 | 传感器⾃动校零数据记录 | | | | | |
| 必选参数 | ⽆ | | | | | |
| 默认参数 | ⽆ | | | | | |
| 返回值 | 错误码 成功-0 失败-errcode . weight ：传感器质量 kg  . pos= [x,y,z] ：传感器质⼼ mm | | | | | |