北京思灵机器人科技有限责任公司

Diana API 说明文档

（Python 语言）

此页为空白页

目录

修订历史

版本	修改内容	修订人	修订时间
V1.0	创建	石国庆	
V2.0	1. 实现接口 forward、inverse、speedJ、speedL、enterForceMode、leaveForceMode、freeDriving、holdBrake、releaseBrake、stop、getRobotState、initSrv、destroySrv、getJointPos、getTcpPos、getJointAngularVel、getTcpForce、getJointForce、createPath、addMoveL、addMoveJ、runPath、destroyPath、moveJToTarget、moveJToPose、moveLToTarget、moveLToPose、getJointCurrent、getJointTorque、setDefaultActiveTcp、setDefaultActiveTcpPose	石国庆	2020-09-09
V2.1	1. 修订前版本大部分函数无返回值的情况，返回值改为 True:成功，False:失败 2. 新增接口 MoveTCP、rotationTCP、moveJoint、getTcpExternalForce、changeControlMode、getLibraryVersion、formatError、getLastError、setLastError、getLinkState、isCollision、resume、setJointCollision、setCartCollision、setResultantCollision、setJointImpedance、getJointImpedance、setCartImpedance、getCartImpedance、zeroSpaceFreeDriving、rpy2Axis、axis2RPY、homogeneous2Pose、pose2Homogeneous 3. 修改 initSrv 函数，使其兼容支持回调函数作为传参	石国庆	2020-09-18
V2.2	1. 修改文档版式 2. 新增接口 servoJ、servoL、speedJ_ex、speedL_ex、servoJ_ex、servoL_ex、enableCollisionDetection、createComplexPath、addMoveLSegmentByPose、addMoveLSegmentByTarget、addMoveJSegmentByPose、addMoveJSegmentByTarget、addMoveCSegmentByPose、addMoveCSegmentByTarget、runComplexPath、destroyComplexPath、saveEnvironment	石国庆	2020-11-04
V2.3	1. 新增接口 getDH、getOriginalJointTorque、getJacobiMatrix 2. 修改函数 getJointTorque 含义	石国庆	2020-11-27
V2.4	1. 新增 resetDH	孟庆婷	2020-12-10
V2.5	1. 新增 setPushPeriod、initDHCali、getDHCaliResult、setDH、setWrd2BasRT、setFla2TcpRT、enableTorqueReceiver、sendTorque_rt、dumpToUDisk、dumpToUDiskEx、enterForceMode_ex	石国庆	2020-12-17

V2.6	1. 调整 python 的 API 顺序与已有 C 库一致 2. 新增接口 setWayPoint、getWayPoint、addWayPoint、deleteWayPoint、getDafaultActiveTcp、getDafaultActiveTcpPose、getActiveTcpPose、zeroSpaceManualMove、moveTcp_ex	石国庆	2021-06-01
v2.7	1. 支持同时控制多台机械臂 2. 调整 API 顺序与 C 语言一致	石国庆	2021-07-15
v2.8	1. 限制 leaveForceMode 输入参数	石国庆	2021-07-20
v2.9	1. 调整 API 顺序与 C 语言一致 2. 补齐 API	石国庆	2021-09-17

该操作库函数的所有输入输出参数，均采用国际量纲，即力（N），扭矩（Nm），电流（A），长度（m），线速度（m/s），线加速度（m/s^2），角度（rad），角速度（rad/s），角加速度（rad/s^2），时间（s）。

1 initSrv

def initSrv(srv_net_st, fnError = None,fnState = None)
初始化 API，完成其他功能函数使用前的初始化准备工作。 **参数：** pinfo：srv_net_st 为元组，用于配置本地连接服务器、心跳服务和状态反馈服务的端口号信息及服务器 IP，其中 IP 地址需要传入字符串，端口号如果传 0 则由系统自动分配。 fnError：可选参数，错误处理回调函数。其中 e 为 int 类型的错误码（包含通信错误例如版本不匹配，链路错误例如网络断开，硬件故障例如编码器错误等），可调用 formatError 获取字符串提示信息。fnError 函数会用于多线程中实时反馈，所以尽量不要在函数实现中使用 sleep 函数之类会阻塞线程的操作。 fnState：可选参数，robot state 回调函数。回调函数参数为类 StrRobotStateInfo 的 POINTER，包含以下数据： ● 关节角度元组（jointPos） ● 关节角速度元组（jointAngularVel） ● 关节角电流当前值元组（jointCurrent） ● 关节角扭矩元组（jointTorque） ● TCP 位姿向量（tcpPos） ● TCP 外部力（tcpExternalForce） ● 是否发生碰撞标志（bCollision） ● TCP 外部力是否有效标志（bTcpForceValid） ● TCP 六维力元组（tcpForce） ● 轴空间力元组（jointForce） **返回值：** True：成功 False：失败

调用示例 1(含回调函数)：
import DianaApi
def errorCallback(e):
print("error code" + str(e))
def robotStateCallback(stateInfo):

```
        for i in range(0,7):
            print(stateInfo.contents.jointPos[i])
    for i in range(0,7):
            print(stateInfo.contents. jointAngularVel [i])
fnError = DianaApi.FNCERRORCALLBACK(errorCallback)
fnState = DianaApi.FNCSTATECALLBACK(robotStateCallback)
netInfo=('192.168.10.75', 0, 0, 0, 0, 0)
DianaApi.initSrv(netInfo, fnError ,fnState)
DianaApi.destroySrv()
```

调用示例 2(不含回调函数):

```
netInfo=('192.168.10.75', 0, 0, 0, 0, 0)
netInfo1=('192.168.10.70', 0, 0, 0, 0, 0)
DianaApi.initSrv(netInfo)
DianaApi.initSrv(netInfo1)
DianaApi.destroySrv()
```

2 destroySrv

def destroySrv(ipAddress=‘‘)

结束调用 API，用于结束时释放指定 IP 地址机械臂的资源。如果该函数未被调用就退出系统（例如客户端程序在运行期间崩溃），服务端将因为检测不到心跳而认为客户端异常掉线，直至客户端再次运行，重新连接。除此之外不会引起严重后果。

参数：

ipAddress: 可选参数，需要释放服务资源的机械臂的 IP 地址字符串，如果为空，则会释放全部已经成功 initSrv 的机械臂的资源。

返回值：

True：成功

False：失败

调用示例：

```
netInfo=('192.168.10.75', 0, 0, 0)
netInfo1=('192.168.10.70', 0, 0, 0)
DianaApi.initSrv(netInfo)
DianaApi.initSrv(netInfo1)
```

DianaApi.destroySrv('192.168.10.75') DianaApi.destroySrv('192.168.10.70')

3 setPushPeriod

def setPushPeriod(period,ipAddress = ‘‘)
设置指定 IP 地址机械臂的数据推送周期 **参数：** period：输入参数。推送周期，单位为 ms。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：成功。 -1：失败。
调用示例： ipAddress = ‘192.168.10.75’ setPushPeriod(10, ipAddress)

4 moveTCP

def moveTCP(d, v, a, ipAddress=‘‘)
手动移动指定 IP 地址的机械臂末端中心点。该函数会立即返回，停止运动需要调用 stop 函数。 **参数：** d：表示移动方向的枚举类型。DianaApi 中存在枚举 tcp_direction_e，以下为枚举值及其含义： ● T_MOVE_X_UP 表示沿 x 轴正向 ● T_MOVE_X_DOWN 表示沿 x 轴负向 ● T_MOVE_Y_UP 表示沿 y 轴正向 ● T_MOVE_Y_DOWN 表示沿 y 轴负向 ● T_MOVE_Z_UP 表示沿 z 轴正向 ● T_MOVE_Z_DOWN 表示沿 z 轴负向 v：速度，单位：m/s a：加速度，单位：m/s^2 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值： 0：成功。 -1：失败。
调用示例： import DianaApi import time dtype = DianaApi.tcp_direction_e .T_MOVE_X_UP vel = 0.1 acc = 0.2 ipAddress = '192.168.10.75' DianaApi .moveTCP(dtype, vel, acc, ipAddress) time.sleep(1) DianaApi .stop(’192.168.10.75’)

5 rotationTCP

def rotationTCP(d, v, a, ipAddress=“)
使指定的 IP 地址的机械臂绕末端中心点变换位姿。该函数会立即返回，停止运动需要调用 stop 函数。 **参数：** d：表示移动方向的枚举类型。具体介绍参见 MoveTcp 第一个参数。 v：速度，单位：rad/s。 a：加速度，单位：rad/s^2。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) type = DianaApi.tcp_direction_e.T_MOVE_X_UP

```
vel = 0.1
acc = 0.2
DianaApi.rotationTCP(type, vel, acc, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
DianaApi.destroySrv()
```

6 moveJoint

def moveJoint(d, i, v, a, ipAddress=‘‘)

手动控制指定 IP 地址的机械臂关节移动。该函数会立即返回，停止运动需要调用 stop 函数。

参数：

d：表示关节移动方向的枚举类型。DianaApi 中存在枚举 joint_direction_e ，以下为枚举值及其含义：

- T_MOVE_ UP 表示关节沿正向旋转
- T_MOVE_DOWN 表示关节沿负向旋转。

i：关节索引号。

v：速度，单位：rad/s。

a：加速度，单位：rad/s^2。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
type = DianaApi.joint_direction_e.MOVE_UP
index = 3
vel = 0.8
acc = 0.8
ipAddress = ‘192.168.10.75’
DianaApi.moveJoint(type, index, vel, acc, ipAddress)
time.sleep(3)
```

DianaApi.stop(ipAddress)

7 moveJToTarget

def moveJToTarget(joints, v, a, ipAddress=‘‘)
控制指定 IP 地址的机械臂以七个关节角度为终点的 moveJ。该函数会立即返回，停止运动需要调用 stop 函数。 **参数：** joints：包含 7 个终点关节角度的元组 v：速度，单位：rad/s。 a：加速度，单位：rad/s^2。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi import time DianaApi.joints= (0.0,0.0,0.0,0.0,0.0,0.0,0.0) vel = 0.2 acc = 0.4 ipAddress = ‘192.168.10.75’ DianaApi.moveJToTarget(joints, vel, acc, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)

8 moveJToPose

def moveJToPose(pose, v, a, ipAddress=‘‘)
控制指定 IP 地址的机械臂以工具中心点位姿为终点的 moveJ。该函数会立即返回，停止运动需要调用 stop 函数。 **参数：** pose：终点位姿元组，长度为 6。保存 TCP 坐标（x, y, z）和轴角（rx, ry, rz）组合的矢量数据。

v：速度，单位：rad/s。 a：加速度，单位：rad/s^2。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi import time poses = (0.087,0.0,1.0827,0.0,0.0,0.0) vel = 0.2 acc = 0.4 ipAddress = ‘192.168.10.75’ DianaApi .moveJToPose(poses, vel, acc, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)

9 moveJ

def moveJ (joints, v, a, ipAddress=‘‘)
同 moveJToTarget **参数：** joints：包含 7 个终点关节角度的元组 v：速度，单位：rad/s。 a：加速度，单位：rad/s^2。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi import time DianaApi.joints= (0.0,0.0,0.0,0.0,0.0,0.0,0.0)

```
vel = 0.2
acc = 0.4
ipAddress = ‘192.168.10.75’
DianaApi.moveJ (joints, vel, acc, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
```

10 moveL

def moveL(pose, v, a, ipAddress=‘‘)

同 moveLToPose

参数：

pose：包含 6 个终点位姿的元组，保存 TCP 坐标（x, y, z）和轴角（rx, ry, rz）组合数据。

v：速度，单位：m/s。

a：加速度，单位：m/s^2。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
poses= (0.087,0.0,1.0827,0.0,0.0,0.0)
vel = 0.2
acc = 0.4
ipAddress = ‘192.168.10.75’
DianaApi moveL (poses, vel, acc, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
```

11 moveLToTarget

def moveLToTarget(joints, v, a, ipAddress=‘‘)

控制指定 IP 地址的机械臂以七个关节角度为终点的 moveL。该函数会立即返回，停止运动需要调用 stop 函数。 **参数：** joints：包含 7 个终点关节角度的元组。 v：速度，单位：m/s。 a：加速度，单位：m/s^2。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi import time joints = (0,0.0,0.0,0.0,0.0,0.0,0.0) vel = 0.2 acc = 0.4 ipAddress = '192.168.10.75' DianaApi.moveLToTarget(joints, vel, acc, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)

12 moveLToPose

def moveLToPose(pose, v, a, ipAddress='')
以工具中心点位姿为终点的 moveL。该函数会立即返回，停止运动需要调用 stop 函数。 **参数：** pose：包含 6 个终点位姿的元组，保存 TCP 坐标（x, y, z）和轴角（rx, ry, rz）组合数据。 v：速度，单位：m/s。 a：加速度，单位：m/s^2。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值： True：成功。 False：失败。
调用示例： import DianaApi import time poses= (0.087,0.0,1.0827,0.0,0.0,0.0) vel = 0.2 acc = 0.4 ipAddress = ‘192.168.10.75’ DianaApi moveLToPose(poses, vel, acc, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)

13 speedJ

def speedJ(speed, a, t, ipAddress=‘‘)
速度模式，关节空间运动。时间 t 为可选项，时间 t 是可选项，如果提供了 t 值，函数将在 t 时间后返回不管目标速度是否已经达到。如果没有提供时间 t 值，函数将在达到目标速度时返回。停止运动需要调用 stop 函数。 **参数：** speed：包含 7 个轴关节角速度的元组。单位：rad/s。 a：加速度，单位：rad/s^2。 t：时间，单位：s。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi import time speeds= (0.0,0.0,0.0,0.0,0.0,0.0,0.5)

```
acc = 0.4
ipAddress = ‘192.168.10.75’
DianaApi speedJ(speeds, acc, 0, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
```

14 speedL

def speedL(speed, a, t, ipAddress=‘‘)

控制指定 IP 地址的机械臂进入速度模式，笛卡尔空间下直线运动。时间 t 是可选项，如果提供了 t 值，函数将在 t 时间后返回不管目标速度是否已经达到。如果没有提供时间 t 值，函数将在达到目标速度时返回。停止运动需要调用 stop 函数。

参数：

speed：工具空间速度元组，长度为 6,其中前 3 个单位为 m/s，后 3 个单位为 rad/s。

a：加速度的元组，长度为 2，单位：m/s^2。

t：时间，单位：s。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
speeds = (0.1,0.0,0.0,0.0,0.0,0.0)
acc = (30, 0.50)
ipAddress = ‘192.168.10.75’
DianaApi speedL(speeds, acc, 0, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
```

15 freeDriving

def freeDriving(enable, ipAddress=‘‘)

实现控制指定 IP 地址的机械臂正常模式与零力驱动模式之间的切换。

参数： enable：bool 变量，是否进入零力驱动模式，True 表明进入零力驱动，False 为退出零力驱动进入正常模式。只有在正常模式下，才可以控制机器人运动。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi import time ipAddress = ‘192.168.10.75’ DianaApi freeDriving(True, ipAddress) time.sleep(10) DianaApi.freeDriving(False, ipAddress)

16 stop

def stop(ipAddress=‘‘)
控制指定 IP 地址的机械臂停止当前执行的任务。将会以最大加速度停止。对应于 UR 的 stopL，stopJ 指令。 **参数：** 无。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.stop(ipAddress)

17 forward

def forward(joints, pose, ipAddress=‘‘)
正解函数，针对指定 IP 地址机器人，由传入的关节角都计算出的正解 TCP 位姿。

参数：

joints：传入参数，7 个轴关节角度的元组。单位：rad。

pose：输入输出参数，位姿列表，长度为 6。数据为包含默认的工具坐标系坐标（x, y, z）和旋转矢量（轴角坐标）组合。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
netInfo=(ipAddress, 0, 0, 0)
DianaApi.initSrv(netInfo)
DianaApi.releaseBrake(ipAddress)
tool1 = (0, 0, 0.1, 0, 0, 0)
tcpTestJointPosition = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
tcp1Position = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
DianaApi.setDefaultActiveTcpPose(tool1, ipAddress)
DianaApi.forward(tcpTestJointPosition, tcp1Position, ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

18 inverse

```
def inverse(pose, joints, ipAddress=‘‘)
```

逆解函数，针对指定 IP 地址机器人，通过 TCP 位姿计算出最佳逆解关节角度。

参数：

pose：输入参数，位姿元组，长度为 6，数据为包含 active_tcp 坐标（x, y, z）和旋转矢量（轴角坐标）组合。

joints：输入输出参数，关节角度的列表，长度为 7。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
pose= (0.64221, 0.0, 0.9403, 0.0, 0.0)
joints= [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
ipAddress = '192.168.10.75'
DianaApi.inverse(pose, joints, ipAddress)
```

19 getJointPos

```
def getJointPos(joints, ipAddress='')
```

获取指定 IP 地址机械臂各个关节角度的位置，库初始化后，后台会自动同步机器人状态信息，因此所有的监测函数都是从本地缓存取数。

参数：

joints：输入输出参数，关节角的列表，元组大小为 7。用于传递获取到的结果。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
joints =[0, 0, 0, 0, 0, 0, 0]
ipAddress='192.168.10.75'
DianaApi.getJointPos(joints, ipAddress)
```

20 getJointAngularVel

```
def getJointAngularVel(vels, ipAddress='')
```

获取指定 IP 地址机械臂当前各关节的角速度。

参数：

vels：输入输出参数，传入空的列表，输出关节角速度，长度为 7。用于传递获取到的结果。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值： True：成功。 False：失败。
调用示例： import DianaApi dianaJointAngularVel = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0] ipAddress=’192.168.10.75’ DianaApi.getJointAngularVel(dianaJointAngularVel, ipAddress)

21 getJointCurrent

def getJointCurrent(current, ipAddress=‘‘)
获取当前关节电流。 **参数：** current：输入输出参数，传入空的列表，输出关节电流，长度为 7。用于传递转获取到的结果。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi jointsCurrent = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0] ipAddress=’192.168.10.75’ DianaApi .getJointCurrent(jointsCurrent, ipAddress)

22 getJointTorque

def getJointTorque(torques, ipAddress=‘‘)
获取指定 IP 地址机械臂各关节真实扭矩数据，即减去零偏的扭矩值。 **参数：** torques：输入输出参数，传入空的列表，输出真实的关节扭矩，长度为 7。用于传递获取到的结果。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi dianaJointTorques = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0] ipAddress = ‘192.168.10.75’ DianaApi .getJointTorque(dianaJointTorques, ipAddress)

23 getTcpPos

def getTcpPos(pose, ipAddress=‘‘)
获取指定 IP 地址机械臂当前 TCP 位姿数据，末端可被 setDefaultActiveTcp 函数改变。 **参数：** pose：输入输出参数，传入空的列表，输出关节 TCP 位姿元组，元组大小为 6。用于传递获取到的结果。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi poses = [0.0] * 6 ipAddress = ‘192.168.10.75’ DianaApi .getTcpPos(poses, ipAddress)

24 getTcpExternalForce

def getTcpExternalForce(ipAddress=‘‘)
获取指定 IP 地址机械臂末端实际感受到的力大小，末端可被 setDefaultActiveTcp 函数改变。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。 **返回值：** 返回力的大小。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ force = DianaApi .getTcpExternalForce(ipAddress)

25 releaseBrake

def releaseBrake(ipAddress=‘‘)
打开指定 IP 地址机械臂的抱闸，启动机械臂。调用该接口后，需要调用者延时 2s 后再做其他操作。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.releaseBrake(ipAddress) DianaApi.holdBrake(ipAddress)

26 holdBrake

def holdBrake(ipAddress=‘‘)
关闭指定 IP 地址机械臂的抱闸，停止机械臂。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。

调用示例： 见 releaseBrake()用例

27 changeControlMode

def changeControlMode(m, ipAddress=‘‘)
控制指定 IP 地址机械臂的模式切换。 **参数：** m：枚举类型 mode_e。枚举及其含义如下 ● T_MODE_INVALID 无意义 ● T_MODE_POSITION 位置模式 ● T_MODE_JOINT_IMPEDANCE 关节空间阻抗模式 ● T_MODE_CART_IMPEDANCE 笛卡尔空间阻抗模式 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.changeControlMode(DianaApi.mode_e.T_MODE_POSITION, ipAddress)

28 getLibraryVersion

def getLibraryVersion()
获取当前库的版本号。 **参数：** 无。 **返回值：** 当前版本号,高 8 位为主版本号，低 8 位为次版本号。
调用示例： import DianaApi uVersion = DianaApi.getLibraryVersion()

29 formatError

def formatError(e, ipAddress=‘‘)

获取指定 IP 地址机械臂的错误码 e 的字符串描述，该错误码在初始化指定的回调函数中会作为形参传入，也可以在函数调用失败后查询得到。对于错误码为-2001 的硬件错误，会延时回馈，一般建议对此类错误延时 100 毫秒后调用 formatError 函数获取具体硬件错误提示信息，否则将提示“refresh later ...”而看不到具体内容。 **参数：** e：错误码。 ipAddress:可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 错误描述信息。
调用示例： import DianaApi e = -1003 ipAddress = ‘192.168.10.75’ print(DianaApi.formatError(e, ipAddress))

30 getLastError

def getLastError(ipAddress=‘‘)
返回指定 IP 地址机械臂最近发生的错误码。该错误码会一直保存，确保可以查询得到，直至库卸载，因此，当库函数调用失败后，如果想知道具体的错误原因，应该调用该函数获取错误码。 **参数：** ipAddress:可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：没有错误。 其它值：具体错误码。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ e = DianaApi.getLastError(ipAddress) print(e)

31 setLastError

def setLastError(e, ipAddress=‘‘)
重置指定 IP 地址机械臂错误码。将系统中记录的错误码重置为 e，通常用于清除错误。 **参数：** e：错误码。 ipAddress:可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 错误码。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ e = DianaApi .setLastError(0, ipAddress)

32 setDefaultActiveTcp

def setDefaultActiveTcp(default_tcp, ipAddress=‘‘)
设置指定 IP 地址字符串的默认工具坐标系。在没有调用该函数时，默认工具中心点为法兰盘中心，调用该函数后，默认的工具坐标系将被改变。该函数将会改变 moveTCP，rotationTCP，moveJToPos，moveLToPose，speedJ，speedL，forward，inverse，getTcpPos，getTcpExternalForce 的默认行为。 **参数：** default_tcp：输入参数，TCP 相对于末端法兰盘的 4*4 齐次变换矩阵的元组，元组大小为 16。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True:成功 False:失败
调用示例： import DianaApi matrix = (1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)

DianaApi.setDefaultActiveTcp (matrix,’ 192.168.10.75’)

33 getLinkState

def getLinkState(ipAddress=‘‘)
获取与指定 IP 地址机械臂间的链路状态。 **参数：** 无。 **返回值：** True：链路正常。 False：链路断开。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.getLinkState(ipAddress)

34 getTcpForce

def getTcpForce(forces, ipAddress=‘‘)
获取指定 IP 地址机械臂的 TCP 末端六维力，末端可被 setDefaultActiveTcp 函数改变。 **参数：** forces：输入输出参数，传入空列表，输出工具中心点处六维力，长度为 6。用于传递获取到的结果。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ tcpForce = [0.0] * 6 DianaApi.getTcpForce(tcpForce, ipAddress) print(‘tcpForce={%f, %f, %f, %f, %f, %f}’%(tcpForce [0], tcpForce [1], \ tcpForce [2], tcpForce [3], tcpForce [4], tcpForce [5]))

35 getJointForce

def getJointForce(forces, ipAddress=‘‘)
获取指定 IP 地址机械臂的轴空间七个关节所受力矩。 **参数：** forces：输入输出参数，传入空列表，输出七关节轴力矩，长度为 7。用于传递获取到的结果。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi forces = [0.0] * 7 ipAddress = ‘192.168.10.75’ DianaApi getJointForce (forces, ipAddress)

36 isCollision

def isCollision(ipAddress=‘‘)
从轴空间判断指定 IP 地址机械臂是否发生碰撞。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：机器人发生碰撞。 False：机器人未发生碰撞。
调用示例： import DianApi ipAddress = ‘192.168.10.75’ DianaApi.isCollision(ipAddress)

37 initDHCali

def initDHCali(tcpMeas, jntPosMeas, nrSets, ipAddress=‘‘)
根据输入的关节角以及末端位置元组计算指定 IP 地址机械臂的 DH 参数。 **参数：** tcpMeas：输入参数。TCP 位置数据元组，元组大小为 3 * nrSets。每组数据为[x,y,z]，共 nrSets 组。单位：米。 jntPosMeas：输入参数。关节角位置元组，, 元组大小为 7 * nrSets，每组数据为各关节角位置信息[q1~q7]，共 nrSets 组。单位：弧度。 nrSets：输入参数。测量样本数量，最少 32 组，至少保证大于或等于需要辨识的参数。ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：正常。 -1：失败。

调用示例：
import DianaApi
#以下参数仅供 API 展示，无实际含义
rowNo_refData = 32
tcpMeas = [0.0] * 96
jntMeas = [0.0] * 224
ipAddress = ‘192.168.10.75’
DianaApi.initDHCali(tcpMeas, jntMeas, rowNo_refData, ipAddress)

38 getDHCaliResult

def getDHCaliResult(rDH, wRT, tRT, confid, ipAddress=‘‘)
获取指定 IP 地址机械臂 DH 参数的计算结果。 **参数：** rDH：输出参数。机器人各关节 DH 参数元组的元组，元组大小为 28。每七个数为一组，共四组数据[a，alpha，d，theta]。单位：rad、m。 wRT：输出参数。机器人基坐标系相对于世界坐标系下的位姿元组的元组，元组大小为 6。位姿数据[x, y, z, Rx，Ry，Rz]。单位：rad、m。 tRT： 输出参数。靶球在法兰坐标系下的位置描述元组的元组，元组大小为 3。元组为靶球位置坐标[x,y,z]。单位：m。

confid：输出参数。绝对定位精度参考值元组的元组，元组大小为 2。其中，第一个值为标定前绝对定位精度，第二个值为标定后绝对定位精度。单位：m。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：获取成功。

1：获取结果精度可能够较低。

-1：获取失败，发生异常。

调用示例：

```
import DianaApi
#以下参数仅供 API 展示，无实际含义
rDH = [0.0] * 28
wRT[0.0] * 6
tRT[0.0] * 3
ipAddress = ‘192.168.10.75’
DianaApi.getDHCaliResult(rDH, wRT, tRT, ipAddress)
```

39 setDH

def setDH(a, alpha, d, theta, ipAddress=‘‘)

设置指定 IP 地址机械臂当前 DH 参数。特别注意，**错误的参数设置**可能引起机器人损坏，需谨慎设置！

参数：

a：输入参数。各关节的 a 参数元组的元组，元组大小为 7。

alpha：输入参数。各关节的 alpha 参数元组的元组，元组大小为 7。

d：输入参数。各关节的 d 参数元组的元组，元组大小为 7。

theta：输入参数。各关节的 theta 参数元组的元组，元组大小为 7。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
#以下参数仅供 API 展示，无实际含义
a=[0.0] * 7
alpha=[0.0] * 7
d=[0.0] * 7
theta=[0.0] * 7
ipAddress = '192.168.10.75'
DianaApi.setDH(a, alpha, d, theta, ipAddress)
```

40 setWrd2BasRT

def setWrd2BasRT(RTw2b, ipAddress='')

初始化世界坐标系到指定 IP 地址机械臂坐标系的平移和旋转位姿。用于 DH 参数标定前设置，若用户不能提供此参数，DH 参数标定功能依旧可以使用。如果调用此函数则使用用户自定义的位姿。特别注意，此功能每次移动机器人与激光跟踪仪都需要重新计算，使用错误的参数可能引起 DH 参数计算不准确或标定异常。

参数：

RTw2b：输入参数。世界坐标系到机器人坐标系的平移和旋转位姿元组的元组，元组大小为 6。单位：米和弧度。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
#以下参数仅供 API 展示，无实际含义
wRT = [0.0] * 6
ipAddress = '192.168.10.75'
DianaApi.setWrd2BasRT(wRT, ipAddress)
```

41 setFla2TcpRT

def setFla2TcpRT(RTf2t, ipAddress='')

初始化指定 IP 地址机械臂法兰坐标系到工具坐标系的平移位置。用于 DH 参数标定前设

置，若用户不能提供此参数，DH 参数标定功能依旧可以使用。如果调用此函数则使用用户自定义的位姿。特别注意，此功能每次移动机器人与激光跟踪仪都需要重新计算，使用错误的参数可能引起 DH 参数计算不准确或标定异常。 **参数：** RTf2t：输入参数。初始化法兰坐标系到工具坐标系的平移位置元组，元组大小为 3，位置信息数据[x,y,z]。单位：米。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：成功。 -1：失败。

调用示例：
import DianaApi
#以下参数仅供 API 展示，无实际含义
Fla = [0.0] * 3
ipAddress = ‘192.168.10.75’
DianaApi.setFla2TcpRT(Fla, ipAddress)

42 getRobotState

def getRobotState(ipAddress=‘‘)
获取指定 IP 地址机械臂当前工作状态。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：running。 1：paused。 2：idle。 3：free-driving。 4：zero-space-free-driving。
调用示例： import DianaApi

```
ipAddress = ‘192.168.10.75’
state = DianaApi .getRobotState(ipAddress)
if state == 0:
    print("\t[robot state]:running\n")
elif state == 1:
    print("\t[robot state]:paused\n")
elif state == 2:
    print("\t[robot state]:idle\n")
elif state = 3:
    print("\t[robot state]: free-driving \n")
elif state = 4:
    print("\t[robot state]: zero-space-free-driving \n")
else:
    print("\t[robot state]: unknown state \n")
```

43 resume

def resume(ipAddress=‘‘)

当指定 IP 地址机械臂发生碰撞或其他原因暂停后，恢复运行时使用。

参数：

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianApi
import time
target =[0,0,0,3.141592653/2,0,0,0]
vel = 0.2
acc = 0.2
ipAddress = ‘192.168.10.75’
DianaApi.moveJToTarget(target,vel,acc, ipAddress)
```

```
while True:
    state = DianaApi.getRobotState(ipAddress)
    if state == 0:
        time.sleep(0.01)
    elif state == 1 and DianaApi.isCollision(ipAddress):
        DianaApi.resume(ipAddress)
        time.sleep(1)
    else:
        break
DianaApi.stop(ipAddress)
```

44 setJointCollision

```
def  setJointCollision(collision,ipAddress=‘‘)
```

设置指定 IP 地址机械臂关节空间碰撞检测的力矩阈值。

参数：

collision：输入参数。七关节轴力矩阈值元组，元组大小为 7，单位 N·M

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：设置成功。

False：设置失败。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
collision = (200, 200, 200, 200, 200, 200, 200)
DianaApi.setJointCollision(collision, ipAddress)
```

45 setCartCollision

```
def  setCartCollision(collision,ipAddress=‘‘)
```

设置指定 IP 地址机械臂笛卡尔空间碰撞检测的力矩阈值。

参数：

collision：输入参数。六维力元组，长度为 6，前三维单位 N，后三维单位 N·M

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。 **返回值：** True：成功。 False：失败。
调用示例： collision = (200, 200, 200, 200, 200, 200) ipAddress = ‘192.168.10.75’ DianaApi.setCartCollision (collision, ipAddress)

46 enterForceMode

def enterForceMode(frame_type, frame_matrix, force_direction, force_value, max_approach_velocity, max_allow_tcp_offset,ipAddress=‘‘)
使指定 IP 地址机械臂进入力控模式。 **参数：** frame_type：参考坐标系类型。0：基坐标系；1：工具坐标系；2：自定义坐标系（暂不支持）。 frame_matrix：自定义坐标系矩阵（暂不支持），使用时传单位矩阵对应的元组即可。 force_direction：表达力的方向的元组，大小为 3。 force_value：力大小，长度为 3 的元组。单位：N。 max_approach_velocity：最大接近速度。单位：m/s。 max_allow_tcp_offset：允许的最大偏移。单位：m。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功 False：失败
调用示例： import DianaApi frame_type = 1 frame_matrix = (1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1) force_direction =(0,0,-1) force_value = 2.0

max_approach_velocity = 0.1

max_allow_tcp_offset = 0.2

ipAddress = ‘192.168.10.75’

DianaApi.enterForceMode(frame_type, frame_matrix, force_direction, force_value, max_appro

ach_velocity, max_allow_tcp_offset,ipAddress)

47 leaveForceMode

def leaveForceMode (mode,ipAddress=‘‘)

设置指定 IP 地址机械臂退出力控模式,并设置退出后机械臂的工作模式。

参数：

mode：控制模式。为 mode_e 类型的枚举。枚举声明及其含义如下：

- T_MODE_INVALID：代表无效模式
- T_MODE_POSITION：代表位置模式
- T_MODE_JOINT_IMPEDANCE：代表关节空间阻抗模式
- T_MODE_CART_IMPEDANCE：代表笛卡尔空间阻抗模式

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

import DianaApi

ipAddress = ‘192.168.10.75’

DianaApi.leaveForceMode(mode_e. T_MODE_POSITION, ipAddress)

48 setDefaultActiveTcpPose

def setDefaultActiveTcpPose (arrPose,ipAddress=‘‘)

设置指定 IP 地址机械臂默认的工具坐标系。在没有调用该函数时，默认工具中心点为法兰盘中心，调用该函数后，默认的工具坐标系将被改变。该函数将会改变 moveTCP，rotationTCP，moveJToPos，moveLToPose，speedJ，speedL，forward，inverse，getTcpPos，getTcpExternalForce 的默认行为。

参数：

arrPose：输入参数。TCP 相对于末端法兰盘的位姿向量的元组，元组大小为 6

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。 **返回值：** 0：成功。 -1：失败。
调用示例： import DianaApi pose = (0.1,0.1,0.1,0, 0, 0) ipAddress = ‘192.168.10.75’ DianaApi.setDefaultActiveTcpPose (pose,ipAddress)

49 setResultantCollision

def setResultantCollision (force,ipAddress=‘‘)
设置指定 IP 地址机械臂笛卡尔空间碰撞检测 TCP 的合力矩阈值。 **参数：** force：合力值。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：成功。 -1：失败。
调用示例： import DianaApi force = 8.9 ipAddress = ‘192.168.10.75’ DianaApi.setResultantCollision (force, ipAddress)

50 setJointImpedance

def setJointImpedance (arrStiff, arrDamp, ipAddress=‘‘)
设置指定 IP 地址机械臂各关节阻抗参数，包含钢度 Stiffiness 和阻尼 Damping 的数据。 **参数：** arrStiff：表示各关节钢度 Stiffiness 的元组，长度为 7。 arrDamp：表示各关节阻尼 Damping 的元组，长度为 7。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi arrStiff = (3000, 3000, 1000, 500, 1000, 1000) arrDamp= (50, 40, 15, 30, 9.88, 3.4, 1.0) ipAddress = ‘192.168.10.75’ DianaApi.setJointImpedance (arrStiff, arrDamp, ipAddress)

51 getJointImpedance

def getJointImpedance (arrStiff, arrDamp,ipAddress=‘‘)
获取指定 IP 地址机械臂各关节阻抗参数，包含钢度 Stiffiness 和阻尼 Damping 的数据。 **参数：** arrStiff：表示各关节钢度 Stiffiness 的列表，长度为 7，用于接收获取到的值。 arrDamp：表示各关节阻尼 Damping 的列表，长度为 7，用于接收获取到的值。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。

调用示例：
import DianaApi
#以下参数仅供 API 展示使用，无实际含义
arrStiff = [0] * 7
arrDamp = [0] * 7
ipAddress = ‘192.168.10.75’
DianaApi.getJointImpedance (arrStiff, arrDamp, ipAddress)

52 setCartImpedance

def setCartImpedance (arrStiff, arrDamp, ipAddress=‘‘)
设置指定 IP 地址机械臂笛卡尔空间阻抗参数。

参数：

arrStiff：表示笛卡尔空间，各维度钢度 Stiffiness 的元组，长度为 6。

arrDamp：表示笛卡尔空间，各维度阻尼 Damping 的元组，长度为 6。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
arrStiff = (1000, 1000, 1000, 500, 500, 500)
arrDamp = (10, 10, 10, 5, 5, 5)
ipAddress = ‘192.168.10.75’
DianaApi.setCartImpedance(arrStiff, arrDamp, ipAddress)
```

53 getCartImpedance

```
def getCartImpedance (arrStiff, arrDamp, ipAddress=‘‘)
```

获取指定 IP 地址机械臂笛卡尔空间各维度阻抗参数，包含钢度 Stiffiness 和阻尼 Damping 的数据。

参数：

arrStiff：表示笛卡尔空间，各维度钢度 Stiffiness 的列表，长度为 6，用于接收获取到的值。

arrDamp：表示笛卡尔空间，各维度阻尼 Damping 的列表，长度为 6，用于接收获取到的值。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
arrStiff =  [0] * 6
arrDamp =  [0] * 6
ipAddress = ‘192.168.10.75’
```

DianaApi.getCartImpedance (arrStiff, arrDamp, ipAddress)

54 zeroSpaceFreeDriving

def zeroSpaceFreeDriving (enable, ipAddress=‘‘)
控制指定 IP 地址机械臂进入或退出零空间自由驱动模式。 **参数：** enable：输入参数。True 为进入零空间自由驱动模式；False 为退出零空间自由驱动模式。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ if DianaApi.zeroSpaceFreeDriving (True, ipAddress): time.sleep(10) DianaApi.zeroSpaceFreeDriving(False, ipAddress)

55 createPath

def createPath (type, ipAddress=‘‘)
为指定 IP 地址机械臂创建一个路段。 **参数：** type：输入参数。1 :表示 moveJ, 2：表示 moveL。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 返回带两个参数的元组 参数 0： 0：成功。 -1：失败。 参数 1：

id_path：输出参数。用于保存新创建 Path 的 ID。

调用示例：

```
import DianaApi
import time
def to_rad(x):
    return x*math.pi / 180.0
firstPosition = (to_rad(0), to_rad(20), to_rad(0), to_rad(90), \
to_rad(0), to_rad(120), to_rad(0))
secondPosition = (to_rad(0), to_rad(-20), to_rad(0), to_rad(45), \
to_rad(0), to_rad(-120), to_rad(0))
thirdPosition = (to_rad(0), to_rad(0), to_rad(0), to_rad(0), \
to_rad(0), to_rad(0), to_rad(0))
print('start test moveJ path.')
ipAddress = '192.168.10.75'
path_id=DianaApi.createPath(1,ipAddress)[1]
DianaApi.addMoveJ(path_id, firstPosition, 0.2, 0.2, 0.3,ipAddress)
DianaApi.addMoveJ(path_id, secondPosition, 0.2, 0.2, 0.3,ipAddress)
DianaApi.addMoveJ(path_id, thirdPosition, 0.2, 0.2, 0.3,ipAddress)
DianaApi.runPath(path_id,ipAddress)
DianaApi.destroyPath(path_id,ipAddress)
time.sleep(10)
```

56 addMoveL

def addMoveL (id_path, joints, vel, acc, blendradius, ipAddress='')

向指定 IP 地址机械臂已创建的路段添加 MoveL 路点。

参数：

id_path：输入参数。要添加路点的路径 ID。

joints：输入参数。要添加的路点，即该路点的各关节角度的元组，长度为 7。单位：rad。

vel：moveL 移动到目标路点的速度。单位：m/s。

acc：moveL 移动到目标路点的加速度。单位：m/s^2。

blendradius：交融半径。单位：m。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： 见 createPath 例子

57 addMoveJ

def addMoveJ (id_path, joints, vel_percent, acc_percent, blendradius_percent, ipAddress=‘‘)
向指定 IP 地址机械臂已创建的路段添加 MoveJ 路点。 **参数：** id_path：输入参数。要添加路点的路径 ID。 joints：输入参数。要添加的路点，即该路点的各关节角度的元组，长度为 7。单位：rad vel_percent：moveJ 移动到目标路点的速度百分比。 acc_percent：moveJ 移动到目标路点的加速度百分比。 blendradius _percent：交融半径百分比。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi import time def to_rad(x):   return x*math.pi / 180.0 firstPosition = (to_rad(0), to_rad(20), to_rad(0), to_rad(90), \ to_rad(0), to_rad(120), to_rad(0)) secondPosition = (to_rad(0), to_rad(-20), to_rad(0), to_rad(45), \ to_rad(0), to_rad(-120), to_rad(0))

```
thirdPosition = (to_rad(0), to_rad(0), to_rad(0), to_rad(0), \
to_rad(0), to_rad(0), to_rad(0))
print('start test moveJ path.')
print('start test moveJ path.')
ipAddress = ‘192.168.10.75’
path_id=DianaApi.createPath(1, ipAddress)[1]
DianaApi.addMoveJ(path_id, firstPosition, 0.2, 0.2, 0.3, ipAddress)
DianaApi.addMoveJ(path_id, secondPosition, 0.2, 0.2, 0.3, ipAddress)
DianaApi.addMoveJ(path_id, thirdPosition, 0.2, 0.2, 0.3, ipAddress)
DianaApi.runPath(path_id, ipAddress)
DianaApi.destroyPath(path_id, ipAddress)
time.sleep(10)
```

58 runPath

def runPath (id_path, ipAddress=‘‘)
指定 IP 地址机械臂启动运行设置好的路段。 **参数：** id_path：输入参数。要运行的路径 ID。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： 详见 addMoveJ 或 addMoveL 调用示例。

59 destroyPath

def destroyPath (id_path, ipAddress=‘‘)
销毁指定 IP 地址机械臂某个路段。 **参数：** id_path：输入参数。要销毁的路径 ID。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值： True：成功。 False：失败。
调用示例： 详见 addMoveJ 或 addMoveL 调用示例。

60 rpy2Axis

def rpy2Axis (arr)
欧拉角转轴角。 **参数：** arr：输入输出参数。rpy 角的列表，长度为 3 **返回值：** True：成功。 False：失败。
调用示例： arr = [0.5,0.6,0.7] DianaApi.rpy2Axis(arr) print(arr) DianaApi.axis2RPY(arr) print(arr)

61 axis2RPY

def axis2RPY (arr)
轴角转欧拉角 **参数：** arr：输入输出参数。轴角的列表，长度为 3 **返回值：** True：成功。 False：失败。
调用示例： 见 rpy2Axis

62 homogeneous2Pose

def homogeneous2Pose (matrix, pose)

位姿矩阵转位姿向量 **参数：** matrix：输入参数。位姿矩阵对应的元组，长度为 16 pose:输出参数，位姿向量对应的列表，长度为 6 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi matrix = (0.433013, 0.250000, -0.866025, 0.000000, 0.500000, -0.866025, -0.000000, 0.000000, -0.750000, -0.433013, -0.500000, 0.000000, -0.231000, 0.155000, 0.934000, 1.000000) pose = [0] * 6 DianaApi.homogeneous2Pose(matrix,pose)

63 pose2Homogeneous

def pose2Homogeneous (pose, matrix)
位姿向量转位姿矩阵 **参数：** pose:输入参数，位姿向量对应的元组，长度为 6 matrix：输出参数。位姿矩阵对应的列表，长度为 16 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi pose = (-0.231, 0.155, 0.934, PI, PI/3, PI/6) matrix = [0] * 16 DianaApi. pose2Homogeneous(pose, matrix)

64 enableTorqueReceiver

def enableTorqueReceiver(bEnable, ipAddress=‘‘)
使指定 IP 地址机械臂开启实时扭矩接收

参数：

bEnable：输入参数。是否开启扭矩实时接收

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress=‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.enableTorqueReceiver(True, ipAddress)
DianaApi.destroySrv()
```

65 sendTorque_rt

```
def sendTorque_rt(torque,t, ipAddress=‘‘)
```

对指定 IP 地址机械臂，用户发送实时扭矩

参数：

torque：输入参数。用户传入的扭矩值，大小为 6 的元组。

t:持续时间，单位 ms

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress=‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
torque =(0,0,0,0,0,0)
t = 1000
DianaApi.sendTorque_rt(torque,t, ipAddress)
```

DianaApi.destroySrv()

66 enableCollisionDetection

def enableCollisionDetection (enable, ipAddress=‘‘)
开启指定 IP 地址机械臂碰撞检测 **参数：** enable：bool 变量，是否开启碰撞检测模式，True 表明开启碰撞检测，False 为关闭碰撞检测。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress=‘192.168.10.75’ DianaApi.enableCollisionDetection (True, ipAddress)

67 setActiveTcpPayload

def setActiveTcpPayload(payload,ipAddress = ‘’)
设置指定 IP 地址机械臂的负载信息 **参数：** payload：负载信息，第 1 位为质量，2~4 位为质心，5~10 位为张量，大小为 10 的数组 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：成功。 -1：失败。
调用示例： ipAddress = ‘192.168.10.75’ payload = [0] * 10 ret = setActiveTcpPayload (payload, ipAddress) if ret < 0:

```
print('setActiveTcpPayload failed! Return value = %d' %(ret))
```

68 servoJ

def servoJ(joints_pos, t=0.01, ah_t=0.03, gain=300, ipAddress='')

关节空间内，伺服指定 IP 地址机械臂到指定关节角位置。 servoJ 函数用于在线控制机器人，ah_t 时间和 gain 能够调整轨迹是否平滑或尖锐。 注意：太高的 gain 或太短的 ah_t 时间可能会导致不稳定。由于该函数主要用于以较短位移为目标点的多次频繁调用，建议在实时系统环境下使用。

参数：

joints_pos：目标关节角位置元组，大小为 7

t：运动时间

ah_t：时间（s），范围（0.03-0.2）用这个参数使轨迹更平滑

gain：目标位置的比例放大器，范围（100,2000）

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
ipAddress='192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
PI=3.141592653
target=[0, PI/6, 0, PI/2, 0, -PI/2, 0]
for i in range(10):
   target[3] = target[3]+PI/20
   ret = DianaApi.servoJ(target, 0.01, 0.1, 300,ipAddress)
   if ret < 0:
        break
   time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
```

DianaApi.destroySrv()

69 servoL

def servoL(tcp_pose, t=0.01, ah_t=0.03, gain=300, scale=1, ipAddress=‘‘)
笛卡尔空间内，伺服指定 IP 地址机械臂到指定位姿。由于该函数主要用于以较短位移为目标点的多次频繁调用，建议在实时系统环境下使用。 **参数：** tcp_pose：目标位姿列表，元组大小为 6。前三个元素单位：m；后三个元素单位：rad，注意，角度需要用轴角表示 t：运动时间。单位：s。 time：运动时间。单位：s。 ah_t：时间（s），范围（0.03-0.2）用这个参数使轨迹更平滑。 gain：目标位置的比例放大器，范围（100,2000）。 scale：平滑比例系数。范围（0.0~1.0）。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。

调用示例：
import DianaApi
import time
ipAddress=‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
pi=3.141592653
target = [0.319912,0,0.867999,0,pi/4,0]
for i in range(10):
  target[3] = target[3]+0.005
  ret = DianaApi.servoL(target, ipAddress)
  if ret < 0:
  break

```
    time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

70 servoJ_ex

def servoJ_ex (joints_pos, t=0.01, ah_t=0.03, gain=300, realiable = False, ipAddress=‘‘)

关节空间内，伺服指定 IP 地址机械臂到指定关节角位置优化版。 servoJ_ex 函数用于在线控制机器人，ah_t 时间和 gain 能够调整轨迹是否平滑或尖锐。 注意：太高的 gain 或太短的 ah_t 时间可能会导致不稳定。由于该函数主要用于以较短位移为目标点的多次频繁调用，建议在实时系统环境下使用。

参数：

joints：目标关节角列表，长度为 7。单位：rad。

t：运动时间。单位：s。

ah_t：时间，范围（0.03-0.2）用这个参数使轨迹更平滑。单位：s。

gain：目标位置的比例放大器，范围(100,2000)。

realiable：bool 型变量，值为 True 需要 socket 反馈通信状态，行为等同 servoJ；值为 False 则无需反馈直接返回。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
ipAddress = ‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
PI=3.141592653
target=[0, PI/6, 0, PI/2, 0, -PI/2, 0]
for i in range(10):
```

```
    target[3] = target[3]+PI/20
    ret = DianaApi.servoJ_ex(target, ipAddress)
    if ret < 0:
        break
    time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

71 servoL_ex

```
def servoL_ex(tcp_pose, t=0.01, ah_t=0.03, gain=300, scale = 1, realiable = False, ipAddress=‘‘)
```

笛卡尔空间内，伺服指定 IP 地址机械臂到指定位姿优化版。由于该函数主要用于以较短位移为目标点的多次频繁调用，建议在实时系统环境下使用。

参数：

pose：目标位姿列表，长度为 6。前三个元素单位：m；后三个元素单位：rad，注意，角度需要用轴角表示

t：运动时间。单位：s。

ah_t：范围（0.03-0.2）用这个参数使轨迹更平滑。单位：s。

gain：目标位置的比例放大器，范围（100,2000）。

scale：平滑比例系数。范围（0.0~1.0）。

realiable：bool 型变量，值为 True 需要 socket 反馈通信状态，行为等同 servoJ；值为 False 则无需反馈直接返回。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
ipAddress = ‘192.168.10.75’
```

```
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
pi=3.141592653
target = [0.319912,0,0.867999,0,pi/4,0]
for i in range(10):
   target[3] = target[3]+0.005
   ret = DianaApi.servoL_ex(target,ipAddress)
   if ret < 0:
         break
   time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

72 speedJ_ex

def speedJ_ex(speed, acc, t=0.0, realiable=False, ipAddress=‘‘)

速度模式优化版，使指定 IP 地址机械臂进行关节空间运动。时间 t 为可选项，时间 t 是可选项，如果提供了 t 值，机器人将在 t 时间后减速。如果没有提供时间 t 值，机器人将在达到目标速度时减速。该函数调用后立即返回。停止运动需要调用 stop 函数。

参数：

speed：关节角速度列表，长度为 7。单位：rad/s。

a：加速度，单位：rad/s^2。

t：时间，单位：s。

realiable：bool 型变量，值为 True 需要 socket 反馈通信状态，行为等同 speedJ；值为 False 则无需反馈直接返回。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
DianaApi.initSrv((‘192.168.10.75’,0,0,0))
DianaApi.releaseBrake()
```

```
speeds = [0.0,0.0,0.0,0.0,0.0,0.0,0.5]
acc = 0.40
ret = DianaApi.speedJ_ex(speeds, acc, 0, True)
if ret == False:
    print('speedJ_ex failed! Return value =' + str(ret))
DianaApi.holdBrake()
DianaApi.destroySrv()
```

73 speedL_ex

def speedL_ex(speed, acc, t=0.0, realiable=False, ipAddress=‘‘)

速度模式优化版，使指定 IP 地址机械臂笛卡尔空间下直线运动。时间 t 为可选项，时间 t 是可选项，如果提供了 t 值，机器人将在 t 时间后减速。如果没有提供时间 t 值，机器人将在达到目标速度时减速。该函数调用后立即返回。停止运动需要调用 stop 函数。

参数：

speed：工具空间速度，元组大小为 6,其中前 3 个单位为 m/s，后 3 个单位为 rad/s。

a：加速度，单位：m/s^2。

t：时间，单位：s。

realiable：bool 型变量，值为 true 需要 socket 反馈通信状态，行为等同 speedL；值为 false 则无需反馈直接返回。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
import DianaApi
speeds = [0.1,0.0,0.0,0.0,0.0,0.0]
acc = [0.30, 0.50]
```

```
DianaApi.speedL_ex(speeds, acc, 0, True, ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

74 dumpToUDisk

def dumpToUDisk(ipAddress=‘‘)

导出指定 IP 地址机械臂的日志文件到 u 盘。控制箱中的系统日志文件（主要包含 ControllerLog.txt 和 DianaServerLog.txt）会自动复制到 u 盘。需要注意的是目前控制箱仅支持 FAT32 格式 u 盘，调用 dumpToUDiskEx 函数前需先插好 u 盘，如果系统日志拷贝失败将不会提示。

参数：

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

1. 系统开机
2. 插入 u 盘到控制箱
3. 调用 Api 函数 dumpToUDisk(‘192.168.10.75’)
4. 拔下 u 盘查看

75 inverse_ext

def inverse_ext(ref_joints, pose,joints, ipAddress = ‘’)

针对指定 IP 地址机器人，逆解函数，给定一组参考关节角，算出欧式距离最近的逆解。

参数：

ref_joints：参考的关节角，大小为 7 的列表

pose：输入参数，位姿列表，数据为包含坐标（x, y, z）和旋转矢量（轴角坐标）组合。

joints：输出参数，关节角度列表。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。
调用示例： import DianaApi ref_joints = [0.0] * 7 pose = [0.64221, 0.0, 0.9403, 0.0, 0.0, 0.0] joints = [0.0] * 7 ipAddress = "192.168.10.75" ret = DianaApi.inverse_ext(ref_joints,pose, joints, ipAddress) if ret < 0: print("inverse_ext failed! Return value = %d\n" %(ret))

76 getJointLinkPos

def getJointLinkPos(joints, ipAddress=‘‘)
获取指定 IP 地址机械臂当前低速侧关节角 **参数：** joints：输出参数。低速侧关节角,大小为 7 列表 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress=‘192.168.10.75’ DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake(ipAddress) joints = [0,0,0,0,0,0,0] DianaApi.getJointLinkPos(joints, ipAddress) print(joints) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv()

77 createComplexPath

def createComplexPath (path_type, ipAddress=‘‘)

在指定 IP 地址机械臂上创建一个复杂路段，包括 MoveL、MoveJ、MoveC、MoveP 类型。

参数：

path_type：枚举类型 complex_path_type。枚举及其含义如下

- NORMAL_JOINT_PATH：创建 MoveJ、MoveL、MoveC 路段,传入关节角
- MOVEP_JOINT_PATH：创建 MoveP 路段，传入关节角
- NORMAL_POSE_PATH：创建 MoveJ、MoveL、MoveC 路段，传入 TCP 位姿
- MOVEP_POSE_PATH：创建 MoveP 路段，传入 TCP 位姿

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

返回带两个参数的元组

参数 0：

0：成功。

-1：失败。

参数 1：

id_path：输出参数。用于保存新创建 Path 的 ID。

调用示例：（MoveP 直线运动）

```
import DianaApi
import time
ipAddress = ‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake()
time.sleep(2)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.MOVEP_POSE_PATH,
ipAddress)
if ret[0] == 0:
    pose1 = [0.402863,0,0.871044,0,0,0]
    pose2 = [0.402863,0,0.571044,0,0,0]
    DianaApi.addMoveLSegmentByPose(ret[1],pose1,0.2,0.2,0.1, ipAddress)
    DianaApi.addMoveLSegmentByPose(ret[1],pose2,0.2,0.2,0.1, ipAddress)
    DianaApi.runComplexPath(ret[1] , ipAddress)
```

```
    time.sleep(15)
    DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

78 addMoveLSegmentByTarget

```
def addMoveLSegmentByTarget(complex_path_id, joints, vel, acc, blendradius, ipAddress=‘‘)
```

在指定 IP 地址机械臂上，往已经创建的路径中插入一段直线，支持 MoveL 或 MoveP，需要传入点的关节角

参数：

complex_path_id：输入参数。要添加路点的路段 ID，通过 createComplexPath 传入 NORMAL_JOINT_PATH 枚举生成 MOVEL 类型 id，而传入 MOVEP_JOINT_PATH 枚举生成 MOVEP 类型 id

joints：输入参数。要添加的路点，即该路点的各关节角的列表，长度为 7

vel：移动到目标路点的速度。单位：m/s

acc：移动到目标路点的加速度。单位：m/s^2

blendradius：交融半径。单位：米

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
pi = 3.141592653
def to_rad(deg):
   return deg/180*pi
ipAddress = ‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
```

```
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_JOINT_PATH,
ipAddress)
if ret[0] == 0:
    joint1 = [0,to_rad(-17.193),0,to_rad(83.132),0,to_rad(65.939),0]
    joint2 = [0,to_rad(-25.992),0,to_rad(128.898),0,to_rad(102.906),0]
    DianaApi.addMoveLSegmentByTarget(ret[1],joint1,0.2,0.2,0.1, ipAddress)
    DianaApi.addMoveLSegmentByTarget(ret[1],joint2,0.2,0.2,0.1, ipAddress)
    DianaApi.runComplexPath(ret[1] , ipAddress)
    time.sleep(15)
    DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

79 addMoveLSegmentByPose

```
def addMoveLSegmentByPose(complex_path_id, pose, vel, acc, blendradius, ipAddress='')
```

在指定 IP 地址机械臂上，往路径中插入一段直线段，支持 MoveL 或 MoveP，需要传入 TCP 位姿。

参数：

complex_path_id：输入参数。要添加路点的路段 ID，通过 createComplexPath 传入 NORMAL_POSE_PATH 枚举生成 MOVEL 类型 id，而传入 MOVEP_POSE_PATH 枚举生成 MOVEP 类型 id

pose：输入参数。要添加的路点，为该路点的的 TCP 位姿列表，长度为 6

vel：移动到目标路点的速度。单位：m/s

acc：移动到目标路点的加速度。单位：m/s^2

blendradius：交融半径。单位：米

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：（MoveL 直线运动）

```
import DianaApi
```

```
import time
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_POSE_PATH,
ipAddress)
if ret[0] == 0:
    pose1 = [0.402863,0,0.871044,0,0,0]
    pose2 = [0.402863,0,0.571044,0,0,0]
    DianaApi.addMoveLSegmentByPose(ret[1],pose1,0.2,0.2,0.1, ipAddress)
    DianaApi.addMoveLSegmentByPose(ret[1],pose2,0.2,0.2,0.1, ipAddress)
    DianaApi.runComplexPath(ret[1] , ipAddress)
    time.sleep(15)
    DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

80 addMoveJSegmentByTarget

```
def addMoveJSegmentByTarget(complex_path_id,joints, vel_percent, acc_percent
, blendradius_percent, ipAddress='')
```

在指定 IP 地址机械臂上，往已经创建的路径中插入一段直线，支持 MoveJ，需要传入点的关节角

参数：

complex_path_id：输入参数。要添加路点的路段 ID

joints：输入参数。要添加的路点，即该路点的各关节角度的列表，长度为 7

vel_percent：移动到目标路点的速度百分比

acc_percent：移动到目标路点的加速度百分比

blendradius_percent：交融半径百分比

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
pi = 3.141592653
def to_rad(degree):
    return degree/180 * pi
ret=[False,-1]
ipAddress = '192.168.10.75'
ret=DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_JOINT_PATH,
ipAddress)
if ret[0] == True:
joint1 = [0,0,0,0,0,0,0]
joint2 = [0,0,0,to_rad(90),0,0,0]
DianaApi. addMoveJSegmentByTarget(ret[1],joint1,0.2,0.2,0.1, ipAddress)
DianaApi. addMoveJSegmentByTarget(ret[1],joint2,0.2,0.2,0.1, ipAddress)
DianaApi.runComplexPath(ret[1] , ipAddress)
time.sleep(5)
DianaApi.destoryComplexPath(ret[1] , ipAddress)
```

81 addMoveJSegmentByPose

```
def addMoveJSegmentByPose(complex_path_id,pose, vel_percent, acc_percent
, blendradius_percent, ipAddress='')
```

在指定 IP 地址机械臂上，往路径中插入一段直线段，支持 MoveJ，需要传入 TCP 位姿。

参数：

complex_path_id：输入参数。要添加路点的路段 ID，通过 createComplexPath 传入 NORMAL_POSE_PATH 枚举生成 MOVEL 类型 id，

pose：输入参数。要添加的路点，为该路点的 TCP 位姿列表，长度为 6。

vel_percent：移动到目标路点的速度百分比。

acc_percent：移动到目标路点的加速度百分比。

blendradius_percent：交融半径百分比。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
pi = 3.141592653
def to_rad(deg):
    return deg/180*pi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_POSE_PATH,
ipAddress)
if ret[0] == 0:
        pose1 = [0.40286,0,0.871044,0,0,0]
        pose2 = [0.40286,0,0.571044,0,0,0]
        DianaApi.addMoveJSegmentByPose(ret[1],pose1,0.2,0.2,0.1, ipAddress)
        DianaApi.addMoveJSegmentByPose(ret[1],pose2,0.2,0.2,0.1, ipAddress)
        DianaApi.runComplexPath(ret[1] , ipAddress)
        time.sleep(15)
        DianaApi.destroyComplexPath(ret[1],ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

82 addMoveCSegmentByTarget

```
def addMoveCSegmentByPose(complex_path_id,pass_joints,target_joints,vel,acc, blendradius,
ignore_rotation, ipAddress='')
```

在指定 IP 地址机械臂上，往路径中插入一段圆弧，支持 MoveC 或 MoveP，需要传入关

节角。

参数：

complex_path_id：要添加路点的路段ID。通过createComplexPath传入MOVEP_POSE_PATH枚举生成MOVEP类型，传入NORMAL_POSE_PATH枚举生成MOVEC类型

pass_joints：输入参数。圆弧经过的路点，传入该点关节角的列表

target_joints：输入参数。圆弧的终点，传入该点关节角的列表

vel：移动到目标路点的速度。单位：m/s

acc：移动到目标路点的加速度。单位：m/s^2

blendradius：交融半径。单位：米

ignore_rotation：是否为固定姿态

ipAddress：可选参数，需要控制机械臂的IP地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
pi = 3.141592653
def to_rad(deg):
    return deg/180*pi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.MOVEP_JOINT_PATH,
ipAddress)
if ret[0] == 0:
    joint0 = [to_rad(6.666),to_rad(-25.508),to_rad(-13.723),to_rad(116.906),to_rad(-
10.561),to_rad(43.197),to_rad(12.617)]
    joint1 = [to_rad(18.021),to_rad(-
```

20.483),to_rad(15.462),to_rad(141.904),to_rad(32.691),to_rad(68.061),to_rad(25.780)]

joint2 = [to_rad(11.94),to_rad(-15.738),to_rad(-41.365),to_rad(136.998),to_rad(-30.748), to_rad(72.927),to_rad(-17.738)]

DianaApi.addMoveLSegmentByTarget(ret[1],joint0,0.2,0.2,0.1, ipAddress)

DianaApi.addMoveCSegmentByTarget(ret[1],joint1,joint2,0.2,0.4,0,True, ipAddress)

DianaApi.runComplexPath(ret[1] , ipAddress)

time.sleep(15)

DianaApi.destroyComplexPath(ret[1] , ipAddress)

DianaApi.holdBrake(ipAddress)

DianaApi.destroySrv()

83 addMoveCSegmentByPose

def addMoveCSegmentByPose(complex_path_id, pass_pose, target_pose,vel, acc, blendradius, ignore_rotation,ipAddress=‘‘)

在指定 IP 地址机械臂上，往路径中插入一段圆弧，支持 MoveC 或 MoveP，需要传入 TCP 位姿。

参数：

complex_path_id：要添加路点的路段 ID。通过 createComplexPath 传入 MOVEP_POSE_PATH 枚举生成 MOVEP 类型，传入 NORMAL_POSE_PATH 枚举生成 MOVEC 类型

pass_pose：输入参数。圆弧经过的路点，传入该点 TCP 位姿的列表

target_pose：输入参数。圆弧的终点，传入该点 TCP 位姿的列表

vel：移动到目标路点的速度。单位：m/s

acc：移动到目标路点的加速度。单位：m/s^2

blendradius：交融半径。单位：米

ignore_rotation：是否为固定姿态。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
pi = 3.141592653
def to_rad(deg):
    return deg/180*pi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake()
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.MOVEP_POSE_PATH,
ipAddress)
if ret[0] == 0:
        pose0 = [0.285369,0.088991,0.633264,to_rad(20.379),to_rad(9.576),to_rad(28.775)]
        pose1 = [0.20282,-0.227599,0.423234,to_rad(-0.282),to_rad(59.506),to_rad(-0.952)]
        pose2 = [0.267353,0.241704,0.399786,to_rad(1.057),to_rad(54.307),to_rad(0.907)]
        DianaApi.addMoveLSegmentByPose(ret[1],pose0,0.2,0.2,0.1, ipAddress)
        DianaApi.addMoveCSegmentByPose(ret[1],pose1,pose2,0.05,0.05,0,False,ipAddress)
        DianaApi.runComplexPath(ret[1],ipAddress)
        print(DianaApi.getLastError(ipAddress))
        time.sleep(10)
        DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

84 runComplexPath

```
def runComplexPath(complex_path_id, ipAddress='')
```

在指定 IP 地址机械臂上，运行 id 的路段

参数：

complex_path_id：输入参数。要运行路段 ID。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。
调用示例： 见 createComplexPath 示例

85 destroyComplexPath

def destroyComplexPath(complex_path_id, ipAddress=‘‘)
在指定 IP 地址机械臂上，销毁 id 的路段 **参数：** complex_path_id：输入参数。要销毁路段的 ID。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： 见 createComplexPath 示例

86 saveEnvironment

def saveEnvironment(ipAddress=‘‘)
将指定 IP 地址机械臂的控制器当前参数数据写入配置文件，用于重启机器人时初始化设置各参数，包括碰撞检测阈值、阻抗参数、DH 参数等所用可通过 API 设置的参数数据。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi. SaveEnvironment(ipAddress)

87 dumpToUDiskEx

def dumpToUDiskEx(time_out, ipAddress=‘‘)
导出指定 IP 地址机械臂的日志文件到 u 盘。控制箱中的系统日志文件（主要包含 ControllerLog.txt 和 DianaServerLog.txt）会自动复制到 u 盘。需要注意的是目前控制箱仅支持 FAT32 格式 u 盘，调用 dumpToUDiskEx 函数前需先插好 u 盘，如果系统日志拷贝失败将不会提示。 **参数：** time_out:单位 秒，设置超时时间，一般需要大于 3 秒,-1 表示设置超时时间无穷大。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： 1. 系统开机 2. 插入 u 盘到控制箱 3. 调用 Api 函数 dumpToUDiskEx(-1,’192.168.10.75’) 4. 拔下 u 盘查看

88 enterForceMode_ex

def enterForceMode_ex(forceDirection,forceValue,maxApproachVelocity,maxAllowTcpOffset, active_tcp, ipAddress=‘‘)
使指定 IP 地址机械臂进入力控模式，支持用户自定义的坐标系 **参数：** forceDirection：表达力的方向的元组，大小为 3。 forceValue：力大小，长度为 3 的元组。单位：N。 maxApproachVelocity：最大接近速度。单位：m/s。 maxAllowTcpOffset：允许的最大偏移。单位：m。 active_tcp：用户设定的坐标系的元组，大小为 6，不传则默认为基坐标系。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。

调用示例：

```
import DianaApi
force_direction =(0,0,-1)
force_value = 2.0
max_approach_velocity = 0.1
max_allow_tcp_offset = 0.2
active_tcp=[0,0,0,0,0,0]
ipAddress='192.168.10.75'
DianaApi.enterForceModeEx(force_direction, force_value, max_approach_velocity, max_allo
w_tcp_offset,active_tcp, ipAddress)
```

89 readDI

def readDI (group_name,di_name, ipAddress='')

读取指定 IP 地址机械臂一个数字输入的值。

参数：

group_name：数字输入的分组，例如，’ board’,’plc’；

name：数字输入的信号名，例如，’ di0’；

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

元组，共计两个元素，第一个元素表示函数调用是否成功：

True：成功。

False：失败。

第二个元素表示读取的数字输入的值。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.readDI('board','di0', ipAddress)
if ret[0] == True:
    print(ret[1])
else:
    print("cannot read di")
```

90 readAI

def readAI (group_name,name, ipAddress=‘‘)
读取指定 IP 地址机械臂一个模拟输入的值和模式。 **参数：** group_name：模拟输入的分组，例如，’ board’,’plc’； name：模拟输入的信号名，例如，’ ai0’； ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 元组，共计两个元素，第一个元素表示函数调用是否成功： True：成功。 False：失败。 第二个元素表示读取的模拟输入的值。

调用示例：
import DianaApi
ipAddress = ‘192.168.10.75’
ret = DianaApi.readAI('board','ai0', ipAddress)
if ret[0] == True:
print(ret[1])
else:
print("cannot read ai")

91 setAIMode

def setAIMode (group_name,name,mode, ipAddress=‘‘)
设置指定 IP 地址机械臂模拟输入的模式。 **参数：** group_name：模拟输入的分组，例如，’ board’,’plc’； name：模拟输入的信号名，例如，’ ai0’； mode: 模拟输入模式，1 代表电流，2 代表电压。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
mode = 1
ipAddress = ‘192.168.10.75’
ret = DianaApi.setAIMode('board','ai0',mode, ipAddress)
if ret[0] == True:
    print(ret[1])
else:
    print("cannot set ai mode")
```

92 writeDO

def writeDO (group_name,name,value, ipAddress=‘‘)

设置指定 IP 地址机械臂一个数字输出的值。可用于改变模拟信号的模式，将信号名替换为“aimode”或“aomode”。

参数：

group_name：数字输出的分组，例如，’ board’,’plc’；

name：数字输出的信号名，例如，’ do0’；

value：设置的值。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
ret = DianaApi.writeDO('board','do0',value, ipAddress)
if ret[0] == True:
    print(ret[1])
```

```
else:
    print("cannot write DO")
```

93 writeAO

def writeAO (group_name,name,value, ipAddress=‘‘)

设置指定 IP 地址机械臂一个模拟输出的值和模式。

参数：

group_name：模拟输出的分组，例如，’ board’,’plc’；

name：模拟输出的信号名，例如：’ ao0’；

mode：当前模拟输出模式，1 代表电流，2 代表电压。

value：设置输出的值。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
mode=1
value=8.8
ipAddress = ‘192.168.10.75’
ret = DianaApi.writeAO('board','do0',mode,value, ipAddress)
if ret[0] == True:
    print(ret[1])
else:
    print("cannot write AO")
```

94 readBusCurrent

def readBusCurrent(ipAddress=‘‘)

读取指定 IP 地址机械臂总线电流。

参数：

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

元组，共计两个元素，第一个元素表示函数调用是否成功：

True：成功。

False：失败。

第二个元素表示读取的总线电流的值。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
ret = DianaApi.readBusCurrent(ipAddress)
if ret[0] == True:
    print(ret[1])
else:
    print("cannot read bus current")
```

95 readBusVoltage

def readBusVoltage(ipAddress=‘‘)

读取指定 IP 地址机械臂总线电压。

参数：

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

元组，共计两个元素，第一个元素表示函数调用是否成功：

True：成功。

False：失败。

第二个元素表示读取的总线电压的值。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
ret = DianaApi.readBusVoltage(ipAddress)
if ret == True:
    print(ret[1])
else:
```

```
print("cannot read bus voltage")
```

96 getDH

def getDH (aDH,alphaDH,dDH,thetaDH, ipAddress=‘‘)

获取指定 IP 地址机械臂的 DH 参数。

参数：

aDH：输入输出参数。连杆长度，长度为 7 的列表

alphaDH:输入输出参数，连杆转角，长度为 7 的列表

dDH:输入输出参数，连杆偏距，长度为 7 的列表

thetaDH:输入输出参数，连杆的关节角，长度为 7 的列表

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
a = [0] * 7
alpha =  [0] * 7
d =  [0] * 7
theta = [0] * 7
ipAddress = ‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi. getDH(a,alpha,d,theta, ipAddress)
DianaApi.destroySrv()
```

97 getOriginalJointTorque

def getOriginalJointTorque (torques, ipAddress=‘‘)

获取指定 IP 地址机械臂传感器反馈的扭矩值，未减去零偏。

参数：

torques：输入输出参数。反馈的扭矩值，长度为 7 列表

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值： True：成功。 False：失败。
调用示例： import DianaApi torques = [0] * 7 ipAddress = ‘192.168.10.75’ DianaApi.initSrv((ipAddress,0,0,0)) DianaApi. getOriginalJointTorque(torques,ipAddress) DianaApi.destroySrv()

98 getJacobiMatrix

def getJacobiMatrix (matrix_jacobi, ipAddress=‘‘)
获取指定 IP 地址机械臂的雅各比矩阵。 **参数：** matrix_jacobi：输入输出参数。雅可比矩阵，长度为 42 列表 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi matrix_jacobi =[0] * 42 ipAddress = ‘192.168.10.75’ DianaApi.initSrv((ipAddress,0,0,0)) DianaApi. getJacobiMatrix (matrix_jacobi,ipAddress) DianaApi.destroySrv()

99 resetDH

def resetDH(ipAddress=‘‘)
重置指定 IP 地址机械臂用户自定义 DH 参数。 **参数：**

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.initSrv((ipAddress,0,0,0)) DianaApi. resetDH(ipAddress) DianaApi.destroySrv()

100 runProgram

def runProgram(name, ipAddress=‘‘)
运行指定 IP 地址机械臂某个程序。 **参数：** name： 程序的名字。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ ret = DianaApi.runProgram(‘AgileRobots’, ipAddress) if ret == False: print(‘run Program failed!’)

101 stopProgram

def stopProgram(name, ipAddress=‘‘)
停止指定 IP 地址机械臂某个程序。 **参数：**

name：程序的名字。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ ret = DianaApi.stopProgram(‘AgileRobots’, ipAddress) if ret == False: 　　print(‘stop Program failed!’)

102 getVariableValue

def getVariableValue(name, ipAddress=‘‘)
获取指定 IP 地址机械臂某个全局变量的值。 **参数：** name：全局变量的名字 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 元组，共计两个元素，第一个元素表示函数调用是否成功： True：成功。 False：失败。 第二个元素表示读取的全局变量的值。 True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.getVariableValue(‘GLOBAL’,value, ipAddress)

103 setVariableValue

def setVariableValue(name,value, ipAddress=‘‘)
设置指定 IP 地址机械臂某个全局变量的值。 **参数：** name：全局变量的名字 value: 设置的全局变量的值 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.setVariableValue(‘GLOBAL’,1, ipAddress)

104 isTaskRunning

def isTaskRunning(name, ipAddress=‘‘)
判断指定 IP 地址机械臂某个程序是否在运行。 **参数：** name：程序名称 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.isTaskRunning(‘AgileRobots’, ipAddress)

105 pauseProgram

def pauseProgram(ipAddress=‘‘)
暂停指定 IP 地址机械臂所有程序。

返回值：

True：成功。

False：失败。

参数：

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
DianaApi.pauseProgram(ipAddress)
```

106 resumeProgram

def resumeProgram(ipAddress=‘‘)

恢复运行指定 IP 地址机械臂已经暂停的程序。

参数：

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
DianaApi.resumeProgram(ipAddress)
```

107 stopAllProgram

def stopAllProgram(ipAddress=‘‘)

停止指定 IP 地址机械臂所有程序。

参数：

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.stopAllProgram(ipAddress)

108 isAnyTaskRunning

def isAnyTaskRunning(ipAddress=‘‘)
判断指定 IP 地址机械臂是否有程序在运行。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.isAnyTaskRunng(ipAddress)

109 cleanErrorInfo

def cleanErrorInfo(ipAddress=‘‘)
清除指定 IP 地址机械臂的错误信息。 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.cleanErrorInfo(ipAddress)

110 setCollisionLevel

def setCollisionLevel(level, ipAddress=‘‘)
设置指定 IP 地址机械臂的碰撞检测类型 **参数：** level：碰撞等级，必须是枚举类型：collision_level，否则报错。各枚举与其对应含义： E_NO_COLLISION_DETECTION:无碰撞检测 E_JOINT_SPACE_DETECTION:关节空间碰撞检测 E_CART_SPACE_DETECTION:笛卡尔空间碰撞检测 E_TCP_RESULTANT_DETECTION: Tcp 合力检测 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = ‘192.168.10.75’ DianaApi.setCollisionLevel(collision_level.E_NO_COLLISION_DETECTION, ipAddress)

111 getJointCount

def getJointCount(ipAddress = ‘‘)
在指定 IP 地址机械臂上，获取其机械臂的关节数目 **参数：** ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：成功。 -1：失败。
调用示例： import DianaApi ipAddress = '192.168.10.75' count = DianaApi.getJointCount(ipAddress)

```
print("Joint Count = %d" %(count))
```

112 getWayPoint

def setWayPoint(waypointName, tcppos, joints, ipAddress=‘‘)
获取指定 IP 地址机械臂上路点变量信息。 **参数：** waypointName：路点变量名称 。 tcppos：输出参数，位姿信息，大小为 6 的列表，注意，角度需要用轴角表示 joints：输出参数，关节角信息 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。

调用示例：
import DianaApi
wayPointName = ‘point’
Tcppos= [0] * 6
Joint=[0] * 7
ipAddress = ‘192.168.10.75’
DianaApi.getWayPoint(wayPointName,Tcppos,Joint,ipAddress)
if ret == True:
print(pos)
print(joints)
else:
print('cannot get waypoint')

113 setWayPoint

def setWayPoint(waypointName, tcppos, joints, ipAddress=‘‘)
修改指定 IP 地址机械臂上路点变量的值 **参数：** strWayPointName：路点变量名称 。 tcppos：位姿信息，大小为 6 的列表，注意，角度需要用轴角表示

joints：关节角信息，大小为 7 的列表。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
pos = [-0.087, 0, 1.316, 0, 3.142, 0]
joints = [-1.518, 0, 22.977, 3.142, 0, 3.142, 0.0]
ipAddress = ‘192.168.10.75’
ret = DianaApi.setWayPoint('test', pos, joints, ipAddress)
if ret == True:
    print(pos)
    print(joints)
else:
    print('cannot set waypoint')
```

114 addWayPoint

def addWayPoint(waypointName, tcppos, joints, ipAddress=‘‘)

在指定 IP 地址机械臂上新增路点变量。

参数：

wayPointName：输入参数，路点变量名称 。

tcppos：输入参数，位姿信息。 大小为 6 的列表，注意，角度需要用轴角表示

joints：输入参数，关节角信息，大小为 7 的列表。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
```

```
pos = [-0.087, 0, 1.316, 0, 3.142, 0]
joints = [-1.518, 0, 22.977, 3.142, 0, 3.142, 0.0]
ipAddress = ‘192.168.10.75’
ret = DianaApi.addWayPoint('test', pos, joints, ipAddress)
if ret == True:
        print(pos)
        print(joints)
else:
        print('cannot add waypoint')
```

115 deleteWayPoint

def deleteWayPoint(waypointName, ipAddress=‘‘)

在指定 IP 地址机械臂上删除路点变量。

参数：

wayPointName：路点变量名称 。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
wayPointName = ‘point’
ipAddress = ‘192.168.10.75’
DianaApi.deleteWayPoint(wayPointName, ipAddress)
```

116 getDefaultActiveTcp

def getDefaultActiveTcp(default_tcp, ipAddress=‘‘)

获取指定 IP 地址机械臂当前的工具坐标系

参数：

default_tcp：当前工具坐标系的矩阵，为一个长度 16 的列表。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值： True：成功。 False：失败。
调用示例： import DianaApi tcp=[0] * 16 ipAddress=‘192.168.10.75’ DianaApi.getDefaultActiveTcp(tcp, ipAddress) print(tcp)

117 getDefaultActiveTcpPose

def getDefaultActiveTcpPose(arrPose, ipAddress=‘‘)
获取指定 IP 地址机械臂末端工具的位姿 **参数：** arrPose：末端工具的位姿，为一个长度 6 的列表，注意，角度需要用轴角表示 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi pose=[0,0,0,0,0,0] ipAddress=‘192.168.10.75’ DianaApi.getDefaultActiveTcpPose(pose, ipAddress) print(pose)

118 getActiveTcpPayload

def getActiveTcpPayload(payLoad, ipAddress=‘‘)
获取指定 IP 地址机械臂的负载信息 **参数：** arrPose：负载信息，第 1 位为质量，2~4 位为质心，5~10 位为张量 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi load=[0] * 10 ipAddress=‘192.168.10.75’ DianaApi.getActiveTcpPayload(load, ipAddress) print(load)

119 zeroSpaceManualMove

def zeroSpaceManualMove(direction,jointsVel,jointAcc, ipAddress=‘‘)
使指定 IP 地址机械臂启动零空间手动移动 **参数：** direction：零空间的方向，为 zero_space_move_direction 类型的枚举，枚举及其含义如下： ● E_FORWARD:正向移动 ● E_BACKWARD:反向移动 jointsVel:各关节的速度，大小为 7 的列表 jointAcc:各关节的加速度，大小为 7 的列表 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi jointVel=[0.1] * 7 jointAcc=[0.1] * 7 ipAddress=‘192.168.10.75’ DianaApi.zeroSpaceManualMove (zero_space_move_direction. E_FORWARD, jointVel, jointAcc, ipAddress)

120 moveTcp_ex

def moveTcp_ex(coordinate,direction,velocity,accleration, ipAddress=‘‘)
使指定 IP 地址机械臂进行多种坐标系下的直线移动 **参数：** coordinate:坐标系类型，应当为枚举类型 coordinate_e，枚举与其含义如下： ● E_BASE_COORDINATE:基坐标系 ● E_TOOL_COORDINATE:工具坐标系 ● E_WORK_PIECE_COORDINATE:工件坐标系 ● E_VIEW_COORDINATE:视角坐标系 direction:移动方向，需要为 tcp_direction_e 的枚举类型。枚举值及其含义为： ● T_MOVE_X_UP 表示沿 x 轴正向 ● T_MOVE_X_DOWN 表示沿 x 轴负向 ● T_MOVE_Y_UP 表示沿 y 轴正向 ● T_MOVE_Y_DOWN 表示沿 y 轴负向 ● T_MOVE_Z_UP 表示沿 z 轴正向 ● T_MOVE_Z_DOWN 表示沿 z 轴负向 velocity：速度，单位：m/s accleration：加速度，单位：m/s^2 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress=‘192.168.10.75’ DianaApi.moveTcp_ex(coordinate_e.E_BASE_COORDINATE, tcp_direction_e. T_MOVE_X_UP,0.1,0.2, ipAddress) time.sleep(2) DianaApi.stop(ipAddress)

121 setExternalAppendTorCutoffFreq

def setExternalAppendTorCutoffFreq(dblFreq, ipAddress = ‘‘)
针对指定 IP 地址机械臂，设置其附加力矩的滤波截止频率

参数：

dblFreq：输入参数，附加力矩的滤波截止频率，需要提供一个正值。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.setExternalAppendTorCutoffFreq(1.0,ipAddress)
if ret < 0:
    print("setExternalAppendTorCutoffFreq failed! Return value = %d" %(ret))
```

122 poseTransform

```
def poseTransform(srcPose,srcMatrixPose,dstMatrixPose,dstPose)
```

把源坐标系下机械臂位姿转换成目的坐标系下位姿

参数：

srcPose：输入参数，源坐标系下的位姿，大小为 6 的元组，均为国际单位制(m 和 rad)，注意，角度需要用轴角表示

srcMatrixPose：输入参数，源坐标系对应的位姿向量 ，大小为 6 的列表

dstMatrixPose：输入参数，目标坐标系对应的位姿向量 ，大小为 6 的列表

dstPose：输出参数，机械臂在目标坐标系下的位姿，大小为 6 的列表，均为国际单位制(m 和 rad)，注意，角度用轴角表示

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
PI = 3.141592653
srcPose = [-0.087,0,1.3165,PI,0,PI]
srcMatrixPose = [0] * 6
```

```
dstMatrixPose = [-0.087,0,1.3165,PI,0,PI]
dstPose = [0] * 6
DianaApi.poseTransform(srcPose,srcMatrixPose,dstMatrixPose,dstPose)
print(dstPose)
```

123 updateForce

def updateForce(forceDirection,forceValue, ipAddress=‘‘)

针对指定 IP 地址机械臂上，在力控模式下，实时改变力指令的大小与方向

参数：

forceDirection：输入参数，力的方向，大小为三的数组，分别表示下 x,y,z 方向的分量

forceValue：输入参数，力的大小，需要是一个大于 0 的数

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
PI=3.141592653
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
frameType = DianaApi.coordinate_e.E_BASE_COORDINATE.value
frameMatrix = [1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1]
forceDirection=[0,0,-1]
forceValue = 1.0
maxVel = 0.1
maxOffset = 0.2
joints = [0,PI/6,0,PI/2,0,-PI/3,0]
DianaApi.moveJ(joints,0.2,0.2,ipAddress)
DianaApi.wait_move(ipAddress)
```

```
if DianaApi.enterForceMode(frameType, frameMatrix, forceDirection, forceValue, maxVel,maxOffset,ipAddress) < 0:
    print("Diana API enterForceMode failed!\n")
else:
    print("Diana API enterForceMode succeeded!\n")
count = 0
while count < 3000:
    forceValue = forceValue - 0.001
    if forceValue < 0:
        forceDirection[2] = 1
    else:
        forceDirection[2] = -1
    DianaApi.updateForce(forceDirection,abs(forceValue),ipAddress)
    time.sleep(0.001)
    count = count + 1
if DianaApi.leaveForceMode(DianaApi.mode_e.T_MODE_POSITION,ipAddress) < 0:
    print("Diana API leaveForceMode failed!\n")
else:
    print("Diana API leaveForceMode succeeded!\n")
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

124 updateForce_ex

def updateForce(forceDirection,forceValue,active_tcp, ipAddress=‘‘)

针对指定 IP 地址机械臂，在力控模式下，实时改变力指令的大小与方向，相比于 updateForce 增加了对工具坐标系支持

参数：

forceDirection：输入参数，力的方向，为一个长度为 3 的列表，分别表示下 x,y,z 方向的分量

forceValue：输入参数，力的大小，单位 N

active_tcp：可选参数，坐标系对应的位姿向量，大小为 6 的列表，不传则默认为基坐标系

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
PI=3.141592653
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
frameType = DianaApi.coordinate_e.E_BASE_COORDINATE.value
frameMatrix = [1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1]
forceDirection=[0,0,-1]
forceValue = 1.0
maxVel = 0.1
maxOffset = 0.2
joints = [0,PI/6,0,PI/2,0,-PI/3,0]
DianaApi.moveJ(joints,0.2,0.2,ipAddress)
DianaApi.wait_move(ipAddress)
if DianaApi.enterForceMode(frameType, frameMatrix, forceDirection, forceValue, maxVel,maxOffset,ipAddress) < 0:
    print("Diana API enterForceMode failed!\n")
else:
    print("Diana API enterForceMode succeeded!\n")
count = 0
active_tcp=[0]*6
while count < 3000:
    forceValue = forceValue - 0.001
    if forceValue < 0:
```

```
            forceDirection[2] = 1
        else:
            forceDirection[2] = -1
        DianaApi.updateForce_ex(forceDirection,abs(forceValue),active_tcp,ipAddress)
        time.sleep(0.001)
        count = count + 1
if DianaApi.leaveForceMode(DianaApi.mode_e.T_MODE_POSITION,ipAddress) < 0:
        print("Diana API leaveForceMode failed!\n")
else:
        print("Diana API leaveForceMode succeeded!\n")
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

125 inverseClosedFull

def inverseClosedFull(pose,lock_joint_index, lock_joint_position, ref_joints, active_tcp,ipAddress = ‘‘)
在指定 IP 地址机械臂上，基于工具坐标系，给定一个参考关节角，约束单轴求逆解，注意，工业臂只能锁定七轴。 **参数：** pose：输入参数，位姿列表，数据为包含 active_tcp 坐标（x, y, z）和旋转矢量（轴角坐标）组合，注意，角度需要用轴角表示。当 active_tcp 不为空时，pose 描述工具中心点在基座标系下的位姿，否则 active_tcp 描述法兰中心点在基座标系下的位姿。 lock_joint_index：输入参数，被约束的关节号。 lock_joint_position：输入参数，被约束关节的角度，单位为弧度。 ref_joints：参考的关节角，大小为 7 的列表。 active_tcp：当前工具坐标系对应的位姿向量，大小为 6 的列表。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 非负数：生成逆解对应的 ID。 -1：失败。
调用示例：

```
import DianaApi
import time
def to_rad(deg):
    return deg * PI / 180
PI=3.141592653
ipAddress = '192.168.10.75'
start_point= [to_rad(0), 0.523599, to_rad(0), 1.570796,to_rad(0), 0.174533, to_rad(0) ]
pose= [0] * 6
lock_joint_index = 6
lock_joint_position = 0
active_tcp=[0] * 6
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.forward(start_point, pose, ipAddress);
id=DianaApi.inverseClosedFull(pose,lock_joint_index,lock_joint_position,start_point,active_tc
p,ipAddress)
if id == -1:
    print("inverseClosedFull failed! Return value = %d\n" %(id))
else:
    size = DianaApi.getInverseClosedResultSize(id)
if size > 0:
    joints=[0] * 7
    for i in range(0,size):
        ret = DianaApi.getInverseClosedJoints(id, i, joints, ipAddress)
        if  ret != -1:
                                                print("%.6f,%.6f,%.6f,%.6f,%.6f,%.6f,%.6f"%
(joints[0],joints[1],joints[2],joints[3],joints[4],joints[5],joints[6]))
    DianaApi.destoryInverseClosedItems(id)
else:
    print("cannot inverse.\n")
DianaApi.destroySrv(ipAddress)
```

126 getInverseClosedResultSize

def getInverseClosedResultSize(id,ipAddress = ‘’)
在指定 IP 地址的机械臂上，根据 ID 获取约束单轴求逆解结果的组数。 **参数：** id：输入参数，所求逆解对应的 ID。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 非负数：ID 对应逆解的组数。 -1：失败。
调用示例： 见 inverseClosedFull 示例。

127 getInverseClosedJoints

def getInverseClosedJoints(id,index,joints,ipAddress = ‘’)
在指定 IP 地址机械臂上，根据 ID 按索引获取对应关节角。 **参数：** id：输入参数，所求逆解对应的 ID。 index：输入参数，对应的多组逆解中的编号。 joints：输出参数，需要求的多组逆解中编号对应的逆解值。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：成功。 -1：失败。
调用示例： 见 inverseClosedFull 示例。

128 destoryInverseClosedItems

def destoryInverseClosedItems(id,ipAddress = ‘’)
在指定 IP 地址机械臂上，根据 ID 删除约束单轴求逆解的结果数据集。 **参数：** id：输入参数，逆解对应的 ID。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。 **返回值：** 0：成功。 -1：失败。
调用示例： 见 inverseClosedFull 示例。

129 getGravInfo

def getGravInfo(grav,ipAddress = ‘‘)
针对指定 IP 地址的机械臂，获取其安装信息的重力矢量。 参数： grav：输出参数，重力矢量，大小为 3 的列表。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：** 0：成功。 -1：失败。
调用示例： import DianaApi ipAddress = '192.168.10.75' grav = [0.0]*3 ret = DianaApi.getGravInfo(grav,ipAddress) if ret < 0: print("getGravInfo failed! Return value = %d\n"%(ret))

130 setGravInfo

def setGravInfo(grav,ipAddress = ‘‘)
针对指定 IP 地址的机械臂，设置其重力矢量。 参数： grav：输入参数，重力矢量，大小为 3 的元组。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 **返回值：**

0：成功。

-1：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
grav = (0.0)*3
ret = DianaApi.setGravInfo(grav,ipAddress)
if ret < 0:
    print("setGravInfo failed! Return value = %d\n"%(ret))
```

131 getGravAxis

def getGravAxis(grav_axis,ipAddress = ‘‘)

针对指定 IP 地址的机械臂，获取其安装信息的轴角。

参数：

grav_axis：输出参数，安装时的轴角，单位为 rad，大小为 3 的列表。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
grav_axis = [0.0]*3
ret = DianaApi.getGravAxis(grav,ipAddress)
if ret < 0:
    print("getGravAxis failed! Return value = %d\n"%(ret))
```

132 setGravAxis

def setGravAxis(grav_axis,ipAddress = ‘‘)

针对指定 IP 地址的机械臂，设置其安装信息的轴角。

参数：

grav_axis：输入参数，安装轴角，单位 rad，大小为 3 的元组。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
grav_axis = (0.0)*3
ret = DianaApi.setGravAxis(grav,ipAddress)
if ret < 0:
    print("setGravAxis failed! Return value = %d\n"%(ret))
```

133 speedLOnTcp

def speedLOnTcp(speed, acc, t=0.0, ipAddress=‘‘)

速度模式优化版，使指定 IP 地址机械臂笛卡尔空间下直线运动。时间 t 为可选项，时间 t 是可选项，如果提供了 t 值，机器人将在 t 时间后减速。如果没有提供时间 t 值，机器人将在达到目标速度时减速。该函数调用后立即返回。停止运动需要调用 stop 函数。

参数：

speed：工具空间速度，元组大小为 6,其中前 3 个单位为 m/s，后 3 个单位为 rad/s。

a：加速度，单位：m/s^2。

t：时间，单位：s。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = ‘192.168.10.75’
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
```

```
speeds = [0.1,0.0,0.0,0.0,0.0,0.0]
acc = [0.30, 0.50]
DianaApi.speedLOnTcp(speeds, acc, 0,  ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

134 getTcpForceInToolCoordinate

def getTcpForceInToolCoordinate(forces,ipAddress = ‘‘)

在指定 IP 地址机械臂上，获取工具坐标系的 Tcp 外力值。

参数：

forces：输出参数，Tcp 外力，大小为 6。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
forces = [0.0]*6
ret = DianaApi.getTcpForceInToolCoordinate(forces,ipAddress)
if ret < 0:
    print("getTcpForceInToolCoordinate failed! Return value = %d"%(ret))
else:
    print(forces)
```

135 calculateJacobi

def calculateJacobi(matrixJacobi,joints,ipAddress = ‘‘)

在指定 IP 地址机械臂上，求解末端法兰中心点坐标系相对于基坐标系的雅各比矩阵。

参数：

matrixJacobi：输出参数，雅各比矩阵，大小为 6*7 的列表。

joints：输入参数，用于计算雅各比矩阵的关节角，大小为 7 的列表。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时

生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
jointCount = 7
tcpCount = 6
jacobiMatrix = [0.0] *(jointCount*tcpCount)
jointPosition = [0.0] * jointCount
ret = DianaApi.calculateJacobi(jacobiMatrix,jointPosition,ipAddress)
if ret == -1:
    print("cannot get jacobi matrix")
else:
    for i in range(0,jointCount):
        for j in range(0,tcpCount):
            print("%lf "%(jacobiMatrix[i* tcpCount + j]))
        print("/n")
```

136 calculateJacobiTF

def calculateJacobiTF(matrixJacobi,joints,toolMatrix ,ipAddress = ‘‘)

在指定 IP 地址机械臂上，求解工具中心点坐标系相对于基坐标系的雅各比矩阵 。

参数：

matrixJacobi：输出参数，雅各比矩阵，大小为 6*7 的列表。

joints：输入参数，用于计算雅各比矩阵的关节角，大小为 7 的列表。

toolMatrix：当前工具坐标系对应的位姿向量，大小为 6 的列表。

ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值：

0：成功。

-1：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
jointCount = 7
tcpCount = 6
jacobiMatrix = [0.0] *(jointCount*tcpCount)
jointPosition = [0.0] * jointCount
toolMatrix= [0.0] * 6
ret = DianaApi.calculateJacobiTF(jacobiMatrix,jointPosition,toolMatrix,ipAddress)
if ret == -1:
    print("cannot get jacobi matrix")
else:
    for i in range(0,jointCount):
        for j in range(0,tcpCount):
            print("%lf "%(jacobiMatrix[i* tcpCount + j]))
        print("/n")
```

137 附件 A：

表 1：Diana API 接口错误码表

系统错误宏定义	错误码	说明
ERROR_CODE_WSASTART_FAIL	-1001	加载 windows 系统 socket 库失败
ERROR_CODE_CREATE_SOCKET_FAIL	-1002	创建 socket 对象失败
ERROR_CODE_BIND_PORT_FAIL	-1003	socket 绑定端口失败
ERROR_CODE_SOCKET_READ_FAIL	-1004	socket 的 select 调用失败
ERROR_CODE_SOCKET_TIMEOUT	-1005	socket 的 select 调用超时
ERROR_CODE_RECVFROM_FAIL	-1006	socket 接收数据失败
ERROR_CODE_SENDTO_FAIL	-1007	socket 发送数据失败
ERROR_CODE_LOST_HEARTBEAT	-1008	服务端的心跳连接丢失
ERROR_CODE_LOST_ROBOTSTATE	-1009	服务端信息反馈丢失
ERROR_CODE_GET_DH_FAILED	-1010	获取 DH 信息失败
ERROR_CODE_JOINT_REGIST_ERROR	-2001	硬件错误
ERROR_CODE_COMMUNICATE_ERROR	-2101	底层通信失败（ln）
ERROR_CODE_CALLING_CONFLICT_ERROR	-2201	暂停状态执行操作失败
ERROR_CODE_COLLISION_ERROR	-2202	发生碰撞
ERROR_CODE_NOT_FOLLOW_POSITION_CMD	-2203	力控模式关节位置与指令发生滞后
ERROR_CODE_NOT_FOLLOW_TCP_CMD	-2204	力控模式 TCP 位置与指令发生滞后
ERROR_CODE_NOT_ALL_AT_OP_STATE	-2205	有关节未进入正常状态
ECODE_OUT_RANGE_FEEDBACK	-2206	关节角反馈超软限位
ECODE_EMERGENCY_STOP	-2207	急停已拍下
ECODE_NO_INIT_PARAMETER	-2208	找不到关节初始参数
ECODE_NOT_MATCH_LOAD	-2209	负载与理论值不匹配
ERROR_CODE_PLAN_ERROR	-2301	路径规划失败
ERROR_CODE_INTERPOLATE_POSITION_ERRO R	-2302	位置模式插补失败
ERROR_CODE_INTERPOLATE_TORQUE_ERROR	-2303	阻抗模式插补失败
ERROR_CODE_SINGULAR_VALUE_ERROR	-2304	奇异位置
ERROR_CODE_RESOURCE_UNAVAILABLE	-3001	参数错误

注：表 1 中 ERROR_CODE_JOINT_REGIST_ERROR（-2001）硬件错误和 ERROR_CODE_NOT_ALL_AT_OP_STATE（-2205）的 OP 状态错误需要通过调用 holdBrake()合抱闸函数或重启硬件来清除错误。