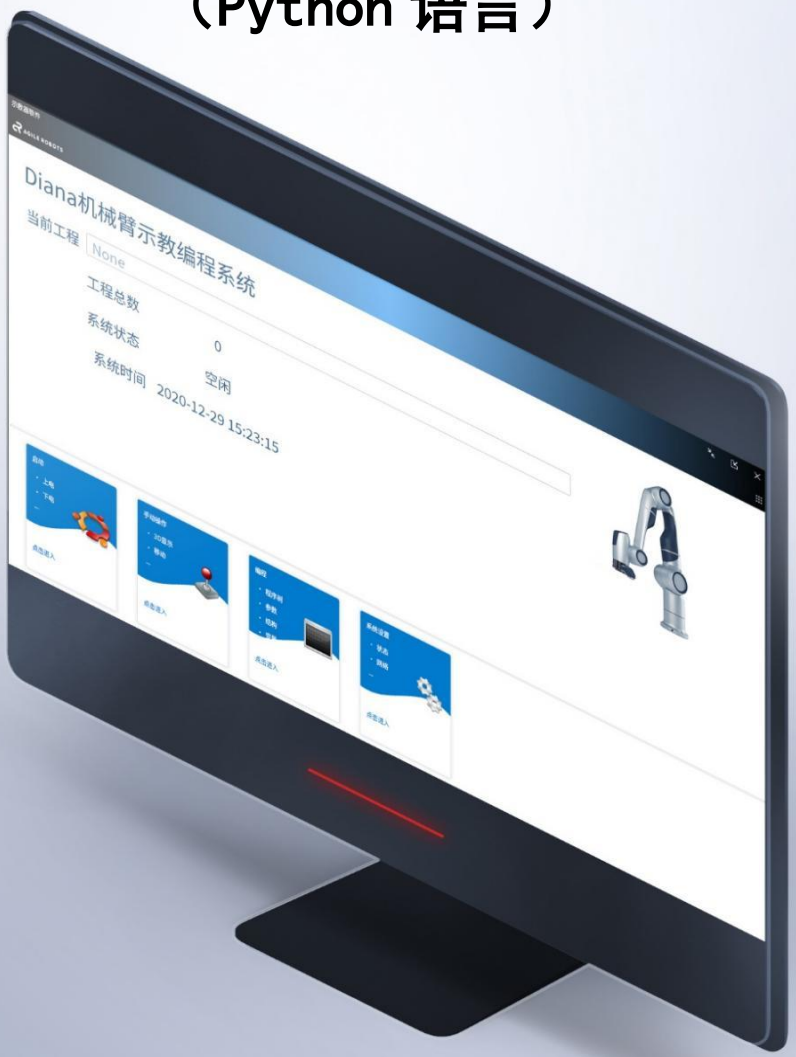
AGILE ROBOTS

北京思灵机器人科技有限责任公司

医疗版远程 API 说明文档

（Python 语言）



软件版本 2.14 | 文档版本 3.32

目录

修订历史

文档版本	修改内容	软件版本	修订时间
V1.0	创建	-	
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	-	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	-	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	-	2020-11-04
V2.3	1. 新增接口 getDH、getOriginalJointTorque、getJacobiMatrix 2. 修改函数 getJointTorque 含义	-	2020-11-27
V2.4	1. 新增 resetDH	-	2020-12-10
V2.5	1. 新增 setPushPeriod、initDHCali、getDHCaliResult、setDH、setWrd2BasRT、setFla2TcpRT、enableTorqueReceiver、sendTorque_rt、dumpToUDisk、dumpToUDiskEx、enterForceMode_ex	-	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	-	2021-06-01
v2.7	1. 支持同时控制多台机械臂	-	2021-07-

	2. 调整 API 顺序与 C 语言一致		15
v2.8	1. 限制 leaveForceMode 输入参数	-	2021-07-20
v2.9	1. 调整 API 顺序与 C 语言一致 2. 补齐 API	-	2021-09-17
V3.0	1. 修改 sendTorque_rt 函数的 torque 参数数组个数为 7，zeroSpaceManualMove 函数的速度和加速度参数的数组个数为 7	-	2021-12-03
V3.8	1. 添加 15 个接口说明和 附件 B，其中接口分别为：setEndKeyEnableState、inverseClosedIdeal、getInverseClosedIdealResultSize、getInverseClosedIdealJoints、destoryInverseClosedIdealItems、getSixAxisZeroOffset、setSixAxisZeroOffset、setDefaultActiveWorkpiece、setDefaultActiveWorkpiecePose、getDefaultActiveWorkpiece、getDefaultActiveWorkpiecePose、setControllerFeatureCode、testControllerFeature、enableSixAxis、isSixAxisEnabled。	-	2021-12-24
V3.9	1. 添加 18 个接口说明：getMechanicalJointsPositionRange、getMechanicalMaxJointsVel、getMechanicalMaxJointsAcc、getMechanicalMaxCartVelAcc、getJointsPositionRange、getMaxJointsVel、getMaxJointsAcc、getMaxCartTranslationVel、getMaxCartRotationVel、getMaxCartTranslationAcc、getMaxCartRotationAcc、setJointsPositionRange、setMaxJointsVel、setMaxJointsAcc、setMaxCartTranslationVel、setMaxCartRotationVel、setMaxCartTranslationAcc、setMaxCartRotationAcc	-	2022-3-16
V3.10	1. 添加 3 个接口说明：freeDriving_ex、freeDrivingWithLockedJoint、freeDrivingWithLockedJoint_ex	V2.7	2022-3-23
V3.11	1. 新增接口 freeDrivingForCurrentLoop 和 getCurrentLoopFreeDrivingState	V2.7	2022-03-26
V3.12	1. 更新错误码	V2.7	2022-03-29
V3.13	1. 新增接口 calcRobotArmAngle、enterSafetyIdle 和 leaveSafetyIdle 2. 更新错误码	V2.7	2022-05-18
V3.14	1. 新增接口 identifyTcpPayload	V2.8	2022-05-19
V3.15	1. 删除接口 getCartImpedance、setCartImpedance、getJointImpedance、setJointImpedance 2. 新增接口 getCartImpeda、setCartImpeda、getJointImpeda、setJointImpeda 3. 删除接口 updateForce_ex，修订 updateForce 接口传参	V2.8	2022-07-07

V3.16	1. 修正 readAI 接口的介绍 2. 修正 getInverseClosedJoints 返回值 3. 补充 setEndKeyEnableState 接口信息 4. 新增接口 getRobotNetworkInfo、setRobotNetworkInfo、getRgbLedState 和 setRgbLedState 5. 新增接口 getFunctionOptI4、setFunctionOptI4	V2.9	2022-07-29
V3.18	新增接口 getGravInfo、setGravInfo、getSixAxisInstallation、initSixAxisInstallation、getCalcSixAxisWaypoints、identifyTcpPayloadWithSixAxis 和 getFixedSixAxisForce	V2.9	2022-08-12
V3.20	1. 修改 MoveJToPose 和 MoveLToPose 函数声明。	V2.10	2022-09-27
V3.21	1. 修改 initSrv 描述。 2. 新增接口 getLastWarning。	V2.10	2022-10-18
V3.22	1. Thor3 六轴机械臂适配	V2.11	2022-11-23
V3.23	1. 更新接口 setFunctionOptI4 描述 2. 更新接口 getFunctionOptI4 描述	V2.11	2023-01-04
V3.24	1. 新增接口 getTcpPayloadWithSixAxis、setTcpPayloadWithSixAxis、getThresholdTorque、setThresholdTorque	V2.12	2023-01-13
V3.25	1. 新增接口 setStaticFriction、getStaticFriciton	V2.12	2023-01-20
V3.26	1. 新增接口 setPredictMoveMode、getPredictMoveMode	V2.12	2023-02-03
V3.27	1. 新增接口 setSafetySpeedLimit、getSafetySpeedLimit	V2.13	2023-03-07
V3.28	2. 新增接口 getEncoderBatteryLevel、getWarningList、formatWarning、cleanWarning，移除 getLastWarning	V2.13	2023-03-10
V3.29	1. 新增接口 setSafetyCartZone、getSafetyCartZone	V2.13	2023-03-17
V3.29	1. 新增接口 calibrateSixAxisSensor	V2.13	2023-03-27
V3.31	1. 新增错误码 -2232：ERROR_CODE_VEL_OR_ACC_PARAMETER_OUT_OF_RANGE 2. 修改 setStaticFriction 的 torque 参数范围 3. 删除附录 E（线程安全函数列表）	V2.13	2023-04-06
V3.32	1. 新增接口 setFunctionOptR8、getFunctionOptR8 和 updateRealtimeVirtualWall	V2.14	2023-05-25

该操作库函数的所有输入输出参数，均采用国际单位，即力（N），扭矩（N.m），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）。如无特殊说明，所有输入输出参数均为轴角或轴角转换的齐次矩阵。另外，文档中涉及关节个数的位置均用 JOINT_NUM 表示，针对 Diana，JOINT_NUM=7，针对 Thor，JOINT_NUM=6。

该操作库所有函数均为线程安全函数。

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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **注：从 2.10 版本开始，系统加入磁盘阵列状态检查。磁盘阵列存在三种状态：可用、部分可用和不可用。initSrv 过程会受到影响，当磁盘阵列中磁盘全部不可用时会自动调用 destroySrv 释放连接，由于释放连接后，lastError 也将被释放，此时本动态库会打印一条 log（"<ip address> "s initSrv failed, RAID state = 2"）提示磁盘阵列中磁盘全部不可用导致 initSrv 失败（返回 False），同时错误码-1018（ERROR_CODE_RAID_INVALID）将传递给 fnError 回调函数,但是该错误码不可以通过 getLastError 获取；当磁盘阵列中部分磁盘可用时，系统会告警，但可以被正常使用， getLastError 会返回 0。** **参数：** srv_net_st：输入参数，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。定义形式 FUNC (e)。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 获取错误码对应的字符串提示信息。fnError 函数会于多线程中获取实时反馈，所以尽量不要在函数实现中执行 sleep 等会阻塞线程的操作。 fnState：可选参数，robot state 回调函数。定义形式 FUNC (robotStateInfo)，回调函数参数类型为 StrRobotStateInfo，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
#Joint_NUM 表示关节 数目，使用 Diana 时，赋值为 7，使用 Thor3 时，赋值为 6。
Joint_NUM=7 #臂型为 Diana 时
Joint_NUM=6 #臂型为 Thor3 时
def errorCallback(e):
    print("error code:" + str(e))
def robotStateCallback(stateInfo):
    for i in range(0, Joint_NUM):
        print(stateInfo.contents.jointPos[i])
    for i in range(0,Joint_NUM):
        print(stateInfo.contents.jointAngularVel[i])
fnError = DianaApi.FNCERRORCALLBACK(errorCallback)
fnState = DianaApi.FNCSTATECALLBACK(robotStateCallback)
netInfo=('192.168.10.75', 0, 0, 0, 0, 0)
ret=DianaApi.initSrv(netInfo, fnError, fnState)
if ret == False:
    print('initSrv failed! Return value ='+ret,'lastError='+str(DianaApi.getLastError()))
else:
    print("inintSrv succeeded!")
DianaApi.destroySrv()
```

调用示例 2(不含回调函数):

```
netInfo=('192.168.10.75', 0, 0, 0, 0, 0)
ret=DianaApi.initSrv(netInfo)
if ret == False:
    print('initSrv failed! Return value ='+ret,'lastError='+str(DianaApi.getLastError()))
else:
    print("inintSrv succeeded!")
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~~另外，如果在 initSrv 中设置了状态或错误回调函数，而未显示调用该函数就退出系统，则会导致 Python 的 GIL 锁与线程之间的冲突导致 Python 主线程卡住的问题，因此在 Python 程序执行完后要显示调用 destroySrv 来结束调用 API。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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

3 setPushPeriod

```
def setPushPeriod(period,ipAddress = "")
```

设置指定 IP 地址机械臂的数据推送周期

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret=DianaApi.setPushPeriod(10, ipAddress)
```

```
if ret:
  print("setPushPeriod succeeded！")
else:
  print("setPushPeriod failed!")
```

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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

d：移动方向，值需要为 DianaApi 中枚举 tcp_direction_e 的枚举值。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：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

d：旋转方向，值需要为 DianaApi 中枚举 tcp_direction_e 的枚举值。枚举值及其含义如下：

- T_MOVE_X_UP 表示沿 x 轴正向旋转；
- T_MOVE_X_DOWN 表示沿 x 轴负向旋转；
- T_MOVE_Y_UP 表示沿 y 轴正向旋转；
- T_MOVE_Y_DOWN 表示沿 y 轴负向旋转；
- T_MOVE_Z_UP 表示沿 z 轴正向旋转；
- T_MOVE_Z_DOWN 表示沿 z 轴负向旋转。

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,0,0,0,0,0))
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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

d：关节移动方向,值需要为 DianaApi 中枚举 joint_direction_e 的枚举值。，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
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
direction = DianaApi.joint_direction_e.T_MOVE_UP
index = 5
vel = 0.8
acc = 0.8
ipAddress = "192.168.10.75"
DianaApi.moveJoint(direction, index, vel, acc, ipAddress)
time.sleep(2)
DianaApi.stop(ipAddress)
```

7 moveJToTarget

```
def moveJToTarget(joints, v, a, ipAddress = "")
```

控制指定 IP 地址的机械臂执行以 JOINT_NUM 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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

joints：包含 JOINT_NUM 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
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
joints= [0]*JOINT_NUM
vel = 0.2
acc = 0.4
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.moveJToTarget(joints, vel, acc, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
```

8 moveJToPose

def moveJToPose(pose, v, a, ipAddress = "", constrain_type:constrain_type_e = constrain_type_e.E_CONSTRAIN_TYPE_NONE, constrain_value = 0.0)

控制指定 IP 地址的机械臂执行以工具中心点位姿为终点的 moveJ，可选择为该移动添加约束。该函数会立即返回，停止运动需要调用 stop 函数。如果移动失败可以通过 getLastError 查询失败原因。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

pose：终点位姿元组，长度为 6。保存 TCP（工具中心点）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。

constrain_type：约束类型，其值及含义如下，

- E_CONSTRAIN_TYPE_NONE 表示无约束 ，为该变量的默认值；
- E_CONSTRAIN_TYPE_NONE 表示无约束；

- E_CONSTRAIN_LOCKED_J3 表示系统会锁定三轴求逆解并以 moveJToTarget 的方式移动到目标位置；
- E_CONSTRAIN_LOCKED_J7 表示系统会锁定七轴求逆解并以 moveJToTarget 的方式移动到目标位置。

constrain_value：约束值。当 constrain_type 为 E_CONSTRAIN_LOCKED_J3 或 E_CONSTRAIN_LOCKED_J7 时该值有效，表示对应类型下被锁轴的角度，单位：rad。该变量的默认值为 0.0。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
import math
def to_degree(x):
   return x*180/math.pi
poses = (0.65, 0, 0.65, 0,0,0)
vel = 0.2
acc = 0.4
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.moveJToPose(poses, vel, acc, ipAddress,
DianaApi.constrain_type_e.E_CONSTRAIN_LOCKED_J3, -math.pi/3)
print("movejToPose lock joint3" + str(to_degree(-math.pi/3)) + "return " + str(ret) + ",
errorcode =" + str(DianaApi.getLastError()))
DianaApi.wait_move()
DianaApi.stop(ipAddress)
```

9 moveLToTarget

```
def moveLToTarget(joints, v, a, ipAddress = "")
```

控制指定 IP 地址的机械臂执行以 JOINT_NUM 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

参数：

joints：包含 JOINT_NUM 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
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
joints = [0.0]* JOINT_NUM
tcp_pose=[0]*6
vel = 0.2
acc = 0.4
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getTcpPos(tcp_pose)
tcp_pose[2]-=0.06
DianaApi.inverse(tcp_pose,joints)
print(tcp_pose)
print(joints)
DianaApi.moveLToTarget(joints, vel, acc, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
```

10 moveLToPose

```
def moveLToPose(pose, v, a, ipAddress = "", constrain_type:constrain_type_e = constrain_type_e.E_CONSTRAIN_TYPE_NONE, constrain_value = 0.0)
```

控制指定 IP 地址的机械臂执行以工具中心点位姿为终点的 moveL，可选择为该移动添加约束。该函数会立即返回，停止运动需要调用 stop 函数。如果移动失败可以通过 getLastError 查询失败原因。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：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。

constrain_type：约束类型。其值及含义如下，中

- E_CONSTRAIN_TYPE_NONE 表示无约束，为该变量的默认值；
- E_CONSTRAIN_TYPE_NONE 表示无约束；
- E_CONSTRAIN_LOCKED_J3 表示系统会锁定三轴求逆解并以 moveJToTarget 的方式移动到目标位置；
- E_CONSTRAIN_LOCKED_J7 表示系统会锁定七轴求逆解并以 moveJToTarget 的方式移动到目标位置。

constrain_value：约束值。当 constrain_type 为 E_CONSTRAIN_LOCKED_J3 或 E_CONSTRAIN_LOCKED_J7 时该值有效，表示对应类型下被锁轴的角度，单位：rad。该变量的默认值为 0.0。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
import math
def to_degree(x):
   return x*180/math.pi
poses=[0]*6
vel = 0.1
acc = 0.1
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getTcpPos(poses)
poses[2]-=0.06
ret = DianaApi.moveLToPose(poses, vel, acc, ipAddress,
DianaApi.constrain_type_e.E_CONSTRAIN_LOCKED_J3, -math.pi/3)
print("moveLToPose lock joint3" + str(to_degree(-math.pi/3)) + "return " + str(ret) + "error ="
+ str(DianaApi.getLastError()))
DianaApi.wait_move()
```

```
DianaApi.stop(ipAddress)
DianaApi.cleanErrorInfo()
```

11 speedJ

def speedJ(speed, acc, t = 0.0, ipAddress = "")

速度模式，控制机械臂关节执行关节空间运动。该函数会立即返回。停止运动需要调用 stop 函数。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

speed：包含 JOINT_NUM 个轴关节角速度的元组。单位：rad/s。

acc：加速度，单位：rad/s^2。

t：可选参数，时间，单位：s，默认是 0。该参数的值会对机械臂的运动产生以下影响，

- t=0：运行速度达到目标速度，机械臂立刻停止运行；
- t>0:不管速度是否达到目标速度，运行持续时间达到时间 t，机械臂立刻停止运行。若运行速度达到目标速度时，时间 t 未消耗完，机械臂会以目标速度运动，直到时间 t 消耗完。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
speeds= [0.0]*JOINT_NUM
speeds[0] = 0.2
acc = 0.2
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.speedJ(speeds, acc, 0, ipAddress)
time.sleep(1)
DianaApi.stop(ipAddress)
```

12 speedL

def speedL(speed, acc, t=0.0, 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。该函数会立即返回。停止运动需要调用 stop 函数。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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。 acc：加速度的元组，长度为 2，单位：m/s^2。 t：可选参数，时间，单位：s，默认是 0。该参数的值会对机械臂的运动产生以下影响， ● t=0：运行速度达到目标速度，机械臂立刻停止运行； ● t>0:不管速度是否达到目标速度，运行持续时间达到时间 t，机械臂立刻停止运行。若运行速度达到目标速度时，时间 t 未消耗完，机械臂会以目标速度运动，直到时间 t 消耗完。 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 = (0.3, 0.50) ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.speedL(speeds, acc, 0, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)

13 freeDriving

def freeDriving(enable, ipAddress = "")
实现控制指定 IP 地址的机械臂在正常模式与零力驱动模式之间切换。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** enable：bool 类型变量，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，才可以通过程序控制机械臂运动。 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.initSrv((ipAddress,0,0,0,0,0)) DianaApi.freeDriving(True, ipAddress) time.sleep(10) DianaApi.freeDriving(False, ipAddress)

14 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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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,0,0,0,0,0)) DianaApi.stop(ipAddress)

15 forward

def forward(joints, pose, ipAddress = "")
正解函数，针对指定 IP 地址机械臂，由传入的关节角度计算出对应的 TCP 位姿。从 2.14 版本开始，正解 forward 函数已支持本地化计算，但因工具坐标系推送存在 1 个周期的延

迟(参考 setPushPeriod)，在设定完工具坐标系后(setDefaultActiveTcp 或 setDefaultActiveTcpPose)立即求正解需要至少等待 1 个推送周期。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

joints：输入参数，JOINT_NUM 个轴关节角度组成的元组。单位：rad。

pose：输入输出参数，位姿列表，用于接收由传入关节角度计算出的 TCP 位姿，长度为 6。数据由 TCP 的坐标（x, y, z）和旋转矢量（Rx，Ry，Rz）组成。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
netInfo=(ipAddress, 0, 0, 0,0,0)
DianaApi.initSrv(netInfo)
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
tcpTestJointPosition = [0.0]* JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
tcp1Position = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
DianaApi.forward(tcpTestJointPosition, tcp1Position, ipAddress)
DianaApi.destroySrv()
```

16 inverse

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

逆解函数，针对指定 IP 地址机械臂，参考当前关节角，通过 TCP 位姿计算出最佳逆解关节角度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

pose：输入参数，位姿元组，长度为 6，包括 active_tcp 坐标（x, y, z）和旋转矢量（Rx，Ry，Rz）。

joints：输入输出参数，关节角度的列表，用于接收逆解结果，长度为 JOINT_NUM。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

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)
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
joints= [0.0]*JOINT_NUM
pose= [0.0]*6
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getTcpPos(pose)
DianaApi.inverse(pose, joints)
```

17 getJointPos

```
def getJointPos(jointsPosition, ipAddress = "")
```

获取指定 IP 地址机械臂当前位姿各个关节的角度信息，库初始化后，后台会自动同步机械臂状态信息，因此所有的监测函数都是从本地缓存或取数据。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

jointsPosition：输入输出参数，长度为 JOINT_NUM 的列表。用于接收获取到的关节角度信息。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
joints =[0.0]*JOINT_NUM
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getJointPos(joints, ipAddress)
```

18 getJointAngularVel

def getJointAngularVel(jointAngularVel, ipAddress = "")

获取指定 IP 地址机械臂当前各关节的角速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

jointAngularVel：输入输出参数，长度为 JOINT_NUM 的列表。用于接收获取到的关节角速度。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
dianaJointAngularVel =  [0.0]*JOINT_NUM
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getJointAngularVel(dianaJointAngularVel, ipAddress)
```

19 getJointCurrent

def getJointCurrent(jointCurrent, ipAddress = "")

获取当前各关节电流。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

jointCurrent：输入输出参数，长度为 JOINT_NUM 的列表。用于传递获取到的结个关节电流。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
jointsCurrent =  [0.0]*JOINT_NUM
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi .getJointCurrent(jointsCurrent, ipAddress)
```

20 getJointTorque

```
def getJointTorque(jointTorque, ipAddress = "")
```

获取指定 IP 地址机械臂当前各关节的真实扭矩数据，即减去零偏的扭矩值。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

jointTorque：输入输出参数，长度为 JOINT_NUM 的列表。用于接收获取到的真实的关节扭矩。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 的 JOINT_NUM 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
dianaJointTorques =[0.0]*JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
```

DianaApi .getJointTorque(dianaJointTorques, ipAddress)

21 getTcpPos

def getTcpPos(tcpPose, ipAddress = "")
获取指定 IP 地址机械臂当前 TCP 位姿数据。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** tcpPose：输入输出参数，长度为 6 的列表，元组大小为 6。用于接收获取到的当前 TCP 位姿数据。 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.initSrv((ipAddress,0,0,0,0,0)) DianaApi .getTcpPos(poses, ipAddress)

22 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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi .getTcpExternalForce(ipAddress)

23 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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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,0,0,0,0,0)) DianaApi.releaseBrake(ipAddress) DianaApi.holdBrake(ipAddress)

24 holdBrake

def holdBrake(ipAddress = "")
关闭指定 IP 地址机械臂的抱闸，停止机械臂。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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

25 changeControlMode

def changeControlMode(m, ipAddress = "")
控制指定 IP 地址机械臂的模式切换。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** m：枚举类型 mode_e。枚举值及其含义如下

- T_MODE_INVALID 表示无意义；
- T_MODE_POSITION 表示位置模式；
- T_MODE_JOINT_IMPEDANCE 表示关节空间阻抗模式；
- T_MODE_CART_IMPEDANCE 表示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.initSrv((ipAddress,0,0,0,0,0))
DianaApi.changeControlMode(DianaApi.mode_e.T_MODE_POSITION, ipAddress)
```

26 getLibraryVersion

```
def getLibraryVersion()
```

获取当前库的版本号。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
uVersion = DianaApi.getLibraryVersion()
```

27 formatError

```
def formatError(e, ipAddress = "")
```

获取指定 IP 地址机械臂的错误码 e 对应的字符串描述，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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
print(DianaApi.formatError(e, ipAddress))
```

28 getLastError

```
def getLastError(ipAddress = "")
```

返回指定 IP 地址机械臂最近产生的错误码。为确保可以查询到最新错误码，该错误码会被一直保存，直至库卸载。当库函数调用失败后，如果想知道具体的错误原因，可以调用该函数获取错误码。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
e = DianaApi.getLastError(ipAddress)
print(e)
```

29 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。该操作完成后会直接影响 getLastError 的调用结果。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
e = DianaApi .setLastError(0, ipAddress)
```

30 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。从 2.14 版本开始，正解 forward 函数已支持本地化计算，但因工具坐标系推送存在 1 个周期的延迟(参考 setPushPeriod)，在设定完工具坐标系后(setDefaultActiveTcp 或 setDefaultActiveTcpPose)立即求正解需要至少等待 1 个推送周期。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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
ipAddress="192.168.10.75"
matrix = (1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)
DianaApi.initSrv((ipAddress,0,0,0,0,0))
```

DianaApi.setDefaultActiveTcp (matrix,ipAddress)

31 getLinkState

def getLinkState(ipAddress = "")
获取当前设备与指定 IP 地址机械臂间的链路状态。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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：链路正常。 False：链路断开。
调用示例： import DianaApi ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.getLinkState(ipAddress)

32 getTcpForce

def getTcpForce(tcpForce, ipAddress = "")
获取指定 IP 地址机械臂的 TCP 末端六维力数据。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** tcpForce：输入输出参数，长度为 6 的列表。用于接收获取到的 TCP 末端六维力数据。 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,0,0,0,0,0)) 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]))

33 getJointForce

def getJointForce(jointForce, ipAddress = "")
获取指定 IP 地址机械臂的轴空间 JOINT_NUM 个关节所受力。 适配臂型：通用医疗臂、定制医疗臂，Thor3。 注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，。 **参数：** jointForce：输入输出参数，长度为 JOINT_NUM 的列表。用于接收获取到的各关节所受力矩。 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
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
forces = [0.0] * JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getJointForce (forces, ipAddress)

34 isCollision

def isCollision(ipAddress = "")
实时检测指定 IP 地址机械臂是否处于碰撞状态。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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：机械臂处于碰撞状态。 False：机械臂未处于碰撞状态。
调用示例： import DianApi ipAddress = "192.168.10.75"

```
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.isCollision(ipAddress)
```

35 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：m。

jntPosMeas：输入参数，关节角位置元组， 元组大小为 JOINT_NUM * nrSets。每组数据为各关节角位置信息[q1~q7]，共 nrSets 组。单位：rad。

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：

True：成功

False：失败

调用示例：

```
import DianaApi
#以下参数仅供 API 展示，无实际含义
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
rowNo_refData = 32
tcpMeas = [0.0] * 96
jntMeas = [0.0] * 32 * JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initDHCali(tcpMeas, jntMeas, rowNo_refData, ipAddress)
```

36 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

rDH：输出参数。机械臂各关节 DH 参数组成的元组，元组大小为 4 * JOINT_NUM。每 JOINT_NUM 个数为一组，共四组数据[a，alpha，d，theta]。单位：rad、m。

wRT：输出参数。机械臂基坐标系相对于世界坐标系下的位姿，大小为 6 的元组。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，大小为 3 的元组。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，大小为 2 的元组。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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功

False：失败

调用示例：

```
import DianaApi
#以下参数仅供 API 展示，无实际含义
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
rDH = [0.0] * 4 * JOINT_NUM
wRT=[0.0] * 6
tRT=[0.0] * 3
confid=[0,0]
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getDHCaliResult(rDH, wRT, tRT, ipAddress)
```

37 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**可能引起机械臂损坏，需谨慎设置！

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

a：输入参数。各关节的 a 参数组成的元组，元组大小为 JOINT_NUM。

alpha：输入参数。各关节的 alpha 参数组成的元组，元组大小为 JOINT_NUM。

d：输入参数。各关节的 d 参数组成的元组，元组大小为 JOINT_NUM。

theta：输入参数。各关节的 theta 参数组成的元组，元组大小为 JOINT_NUM。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功

False：失败。

调用示例：

```
import DianaApi
#以下参数仅供 API 展示，无实际含义
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
a=[0.0] * JOINT_NUM
alpha=[0.0] * JOINT_NUM
d=[0.0] * JOINT_NUM
theta=[0.0] * JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setDH(a, alpha, d, theta, ipAddress)
```

38 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，若不提供此参数，DH 参数标定功能依旧可以使用。如果调用此函数则使用自定义的位姿。特别注意，此功能每次移动机械臂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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

RTw2b：输入参数。世界坐标系到机械臂坐标系的平移和旋转位姿元组，元组大小为 6。单位：m、rad。

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，无实际含义
wRT = [0.0] * 6
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setWrd2BasRT(wRT, ipAddress)
```

39 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，若不提供此参数，DH 参数标定功能依旧可以使用。如果调用此函数则使用自定义的位姿。特别注意，此功能每次移动机械臂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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：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
#以下参数仅供 API 展示，无实际含义
Fla = [0.0] * 3
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setFla2TcpRT (Fla, ipAddress)
```

40 getRobotState

```
def getRobotState(ipAddress = "")
```

获取指定 IP 地址机械臂当前工作状态。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：null-space-free-driving。

5：hold_brake。

6：error。

7：error_handling。

8：torque_receiver。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
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")
elif state == 5:
   print("\t[robot state]: hold-brake \n")
elif state == 6:
   print("\t[robot state]: error \n")
elif state == 7:
   print("\t[robot state]: error_handling \n")
else:
   print("\t[robot state]: unknown state \n")
```

41 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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=0.1
acc=0.1
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.moveJToTarget(target,vel,acc, ipAddress)
while True:
    if DianaApi.isCollision(ipAddress):
        DianaApi.cleanErrorInfo()
        time.sleep(1)
        DianaApi.resume(ipAddress)
        DianaApi.wait_move()
        break
    else:
        time.sleep(0.001)
```

42 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

collision：输入参数。JOINT_NUM 关节轴力矩阈值元组，元组大小为 JOINT_NUM，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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：设置成功。

False：设置失败。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
collision = (200, 200, 200, 200, 200, 200, 200)
#以 7 轴机械臂为例
DianaApi.setJointCollision(collision, ipAddress)
```

43 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

```
import DianaApi
collision = (200, 200, 200, 200, 200, 200)
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setCartCollision (collision, ipAddress)
```

44 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。

适配臂型：通用医疗臂、定制医疗臂。

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。单位：N。

max_approach_velocity：最大接近速度。单位：m/s。

max_allow_tcp_offset：允许的最大偏移距离。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.initSrv((ipAddress,0,0,0,0,0))
DianaApi.enterForceMode(frame_type,frame_matrix,force_direction,force_value,max_approach_velocity,max_allow_tcp_offset,ipAddress)
DianaApi.leaveForceMode(DianaApi.mode_e.T_MODE_POSITION)
```

45 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。

适配臂型：通用医疗臂、定制医疗臂。

参数：

mode：控制模式。为 mode_e 类型的枚举。枚举值及其含义如下：

- T_MODE_INVALID：无效模式
- 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：

见 enterForceMode 调用示例

46 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。从 2.14 版本开始，正解 forward 函数已支持本地化计算，但因工具坐标系推送存在 1 个周期的延迟(参考 setPushPeriod)，在设定完工具坐标系后(setDefaultActiveTcp 或 setDefaultActiveTcpPose)立即求正解需要至少等待 1 个推送周期。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrPose：输入参数。TCP 相对于末端法兰盘的平移旋转向量元组，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：

True：成功。

False：失败。

调用示例：

```
import DianaApi
pose = (0.1,0.1,0.1,0, 0, 0)
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setDefaultActiveTcpPose (pose,ipAddress)
```

47 setResultantCollision

def setResultantCollision (force,ipAddress = "")

设置指定 IP 地址机械臂 TCP 的合力碰撞检测的力矩阈值。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** force：TCP 合力碰撞检测的力距阈值。 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 force = 8.9 ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.setResultantCollision (force, ipAddress)

48 -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。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** enable：输入参数，开启或者退出零空间自由驱动。 ● 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" DianaApi.initSrv((ipAddress,0,0,0,0,0)) if DianaApi.zeroSpaceFreeDriving (True, ipAddress):   time.sleep(10)

```
DianaApi.zeroSpaceFreeDriving(False, ipAddress)
```

49 createPath

```
def createPath (id_type, ipAddress = "")
```

为指定 IP 地址机械臂创建一个路段。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

id_type：输入参数，路段类型。

- 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：

True：成功。

False：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
#以 7 轴机械臂为例
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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
path_id=DianaApi.createPath(1,ipAddress)[1]
DianaApi.addMoveJ(path_id, firstPosition, 0.1, 0.1, 0.3,ipAddress)
DianaApi.addMoveJ(path_id, secondPosition, 0.1, 0.1, 0.3,ipAddress)
DianaApi.addMoveJ(path_id, thirdPosition, 0.1, 0.1, 0.3,ipAddress)
DianaApi.runPath(path_id,ipAddress)
DianaApi.wait_move()
DianaApi.destroyPath(path_id,ipAddress)
```

50 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

id_path：输入参数。要添加路点路段的 ID。

joints：输入参数。要添加的路点，即该路点的各关节角度组成的元组，长度为 JOINT_NUM。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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import copy as cp
#以 7 轴机械臂为例
pose=[0]*6
firstPosition=[0]*7
secondPosition=[0]*7
print('start test moveL path.')
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getTcpPos(pose)
pose_1=cp.deepcopy(pose)
pose_1[1]-=0.01
pose_2=cp.deepcopy(pose_1)
pose_2[1]+=0.01
DianaApi.inverse(pose_1,firstPosition)
DianaApi.inverse(pose_2,secondPosition)
path_id=DianaApi.createPath(2,ipAddress)[1]
DianaApi.addMoveL(path_id, firstPosition, 0.1, 0.1, 0.3,ipAddress)
DianaApi.addMoveL(path_id, secondPosition, 0.1, 0.1, 0.3,ipAddress)
DianaApi.runPath(path_id,ipAddress)
DianaApi.wait_move()
DianaApi.destroyPath(path_id,ipAddress)
```

51 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

id_path：输入参数。要添加路点路段的 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 JOINT_NUM。

单位：rad

vel_percent：moveJ 移动到目标路点的速度，单位：rad/s。

acc_percent：moveJ 移动到目标路点的加速度,单位：rad/s^2。

blendradius_percent：交融半径。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

见 createPath 例子

52 runPath

def runPath (id_path, ipAddress = "")

为指定 IP 地址机械臂启动运行设置好的路段。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

id_path：输入参数。要运行的路段 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 createPath 例子

53 destroyPath

def destroyPath (id_path, ipAddress = "")

销毁指定 IP 地址机械臂的某个路段。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

id_path：输入参数。要销毁的路段 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 createPath 例子

54 rpy2Axis

def rpy2Axis (arr)
rpy 角转轴角。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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： import DianaApi ipAddress="192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) arr = [0.5,0.6,0.7] DianaApi.rpy2Axis(arr) print(arr) DianaApi.axis2RPY(arr) print(arr)

55 axis2RPY

def axis2RPY (arr)
轴角转 rpy 角。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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

56 homogeneous2Pose

def homogeneous2Pose (matrix, pose)

位姿矩阵转位姿向量。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

matrix：输入参数。位姿矩阵，长度为 16 的元组。

pose:输出参数，位姿向量，长度为 6 的列表。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
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)
```

57 pose2Homogeneous

def pose2Homogeneous (pose, matrix)

位姿向量转位姿矩阵。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

pose:输入参数，位姿向量，长度为 6 的元组。

matrix：输出参数。位姿矩阵，长度为 16 的列表。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
```

```
import math
PI=math.pi
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
pose = (-0.231, 0.155, 0.934, PI, PI/3, PI/6)
matrix = [0] * 16
DianaApi. pose2Homogeneous(pose, matrix)matrix = [0] * 16
DianaApi. pose2Homogeneous(pose, matrix)
```

58 enableTorqueReceiver

def enableTorqueReceiver(bEnable, ipAddress = "")

使指定 IP 地址机械臂开启实时扭矩接收。

适配臂型：通用医疗臂、定制医疗臂。

参数：

bEnable：输入参数。是否开启实时扭矩接收。

- True：开启实时扭矩接收；
- False：关闭实时扭矩接收。

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,0,0,0,0,0))
DianaApi.enableTorqueReceiver(True, ipAddress)
DianaApi.enableTorqueReceiver(False, ipAddress)
DianaApi.destroySrv()
```

59 sendTorque_rt

def sendTorque_rt(torque, t, ipAddress = "")

向指定 IP 地址机械臂发送实时扭矩。

适配臂型：通用医疗臂、定制医疗臂。

参数：

torque：输入参数。扭矩值，大小为 JOINT_NUM 的元组。

t：持续时间，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。

注意：

1.JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

2.开启实时扭矩发送之前需要先开启实时扭矩接收，扭矩发送完成需要关闭实时扭矩接收

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
torque = [0.0]*JOINT_NUM
t = 1
DianaApi.enableTorqueReceiver(True, ipAddress)
DianaApi.sendTorque_rt(torque,t, ipAddress)
DianaApi.enableTorqueReceiver(False, ipAddress)
DianaApi.destroySrv()
```

60 enableCollisionDetection

def enableCollisionDetection (enable, ipAddress = "")

开启指定 IP 地址机械臂关节空间碰撞检测。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

enable：bool 变量，是否开启关节空间碰撞检测模式，True 表明开启关节空间碰撞检测，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.initSrv((ipAddress,0,0,0,0,0))
DianaApi.enableCollisionDetection (True, ipAddress)
```

61 setActiveTcpPayload

```
def setActiveTcpPayload(payload,ipAddress = "")
```

设置指定 IP 地址机械臂的负载信息。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
payload = [0] * 10
ret = DianaApi.setActiveTcpPayload (payload, ipAddress)
if ret < 0:
   print("setActiveTcpPayload failed! Return value = %d"%(ret))
else:
   print("setActiveTcpPayload succeed!")
```

62 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械臂，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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

joints_pos：目标关节角位置元组，大小为 JOINT_NUM。

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）。

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,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
#以 7 轴机械臂为例
joints=[0]*7
DianaApi.getJointPos(joints)
for i in range(10):
 joints[6] += 0.01
 ret = DianaApi.servoJ_ex(joints, 0.01, 0.1, 300,False,ipAddress)
 if not ret:
    break
 time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

63 servoL

def servoL(tcp_pose, t=0.01, ah_t=0.03, gain=300, scale=1.0, 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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。

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,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
pose=[0]*6
DianaApi.getTcpPos(pose)
for i in range(10):
  pose[5] += 0.005
  ret = DianaApi.servoL(pose,1,0.03,300,1.0,ipAddress)
  if not ret:
     break
  time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

64 servoJ_ex

def servoJ_ex (joints_pos, t=0.01, ah_t=0.03, gain=300, reliable = 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械臂，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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

joints_pos：目标关节角列表，长度为 JOINT_NUM。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)。

reliable：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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

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,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
PI=3.141592653
#以 7 轴机械臂为例
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 not ret:
     break
  time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

65 servoL_ex

```
def servoL_ex(tcp_pose, t=0.01, ah_t=0.03, gain=300, scale = 1.0, reliable = 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

tcp_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）。

reliable：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,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
pose=[0]*6
DianaApi.getTcpPos(pose)
for i in range(10):
  pose[5] += 0.005
  ret = DianaApi.servoL_ex(pose,1,0.03,300,1.0,False,ipAddress)
  if not ret:
     break
  time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

66 speedJ_ex

```
def speedJ_ex(speed, a, t=0.0, reliable=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，机械臂将在 t 时间后减速。如果没有提供时间 t 值，机械臂将

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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

speed：关节角速度列表，长度为 JOINT_NUM。单位：rad/s。

a：加速度，单位：rad/s^2。

t：时间，单位：s。

reliable：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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
import DianaApi
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake()
speeds = [0.0]*JOINT_NUM
speeds[5]=0.2
acc = 0.2
ret = DianaApi.speedJ_ex(speeds, acc, 0, True,ipAddress)
if ret == False:
   print('speedJ_ex failed! Return value =' + str(ret))
time.sleep(2)
DianaApi.stop()
DianaApi.holdBrake()
DianaApi.destroySrv()
```

67 speedL_ex

def speedL_ex(speed, a, t=0.0, reliable=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，机械臂将在 t 时间后减速。如果没有提供时间 t 值，机械臂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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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。

reliable：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,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
speeds = [0.1,0.0,0.0,0.0,0.0,0.0]
acc = [0.1, 0.1]
DianaApi.speedL_ex(speeds, acc, 0, True, ipAddress)
time.sleep(0.1)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

68 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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

69 inverse_ext

def inverse_ext(ref_joints, pose, joints, ipAddress = "")
针对指定 IP 地址机械臂，给定一个参考关节角，算出欧式距离最近的逆解。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** ref_joints：参考的关节角，大小为 JOINT_NUM 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。 注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。 **返回值：** True：成功。 False：失败。

调用示例：
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
import DianaApi
ref_joints = [0.0] * JOINT_NUM
pose = [0.0]*6
joints = [0.0] * JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getTcpPos(pose)
DianaApi.getJointPos(ref_joints)
ret = DianaApi.inverse_ext(ref_joints,pose, joints, ipAddress)
print(ret)
if not ret:
  print("inverse_ext failed! Return value = {}".format(ret))
else:

```
print("inverse_ext succeed!")
```

70 getJointLinkPos

def getJointLinkPos(joints, ipAddress = "")

获取指定 IP 地址机械臂当前低速侧关节角。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

joints：输出参数。低速侧关节角,大小为 JOINT_NUM 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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
joints = [0,0,0,0,0,0,0]
DianaApi.getJointLinkPos(joints, ipAddress)
print(joints)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

71 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

path_type：枚举类型 complex_path_type。枚举值及其含义如下，

- 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：

True：成功。

False：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
import copy as cp
pose=[0.0]*6
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake()
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_POSE_PATH,
ipAddress)
if ret[0] == 0:
    DianaApi.getTcpPos(pose)
    pose1=cp.deepcopy(pose)
    pose1[2]-=0.01
    pose2=cp.deepcopy(pose1)
    pose2[2]+=0.01
    DianaApi.addMoveLSegmentByPose(ret[1],pose1,0.1,0.1,0.1, ipAddress)
    DianaApi.addMoveLSegmentByPose(ret[1],pose2,0.1,0.1,0.1, ipAddress)
    DianaApi.runComplexPath(ret[1] , ipAddress)
    time.sleep(15)
    DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

72 addMoveLSegmentByTarget

```
def addMoveLSegmentByTarget(complex  path  id, joints, vel, acc, blendradius, ipAddress =
"")
```

在指定 IP 地址机械臂上，往已经创建的路段中插入路点，支持 MoveL 或 MoveP，需要传入点的关节角。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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 JOINT_NUM

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：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：

#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6

```
#以 7 轴机械臂为例
import DianaApi
import time
JOINT_NUM=7
joint1,joint2=[0]*JOINT_NUM,[0]*JOINT_NUM
pose1=[0]*6
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_JOINT_PATH,
ipAddress)
if ret[0] == 0:
    DianaApi.getJointPos(joint1)
    DianaApi.forward(joint1,pose1)
    pose1[1]+=0.01
    DianaApi.inverse(pose1,joint1)
    pose1[1]-=0.01
    DianaApi.inverse(pose1,joint2)
    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)
    DianaApi.wait_move()
    DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

73 addMoveLSegmentByPose

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

在指定 IP 地址机械臂上，往路段中插入路点，支持 MoveL 或 MoveP，需要传入 TCP 位姿。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：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
import copy as cp
pose1=[0.0]*6
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_POSE_PATH,
ipAddress)
if ret[0] == 0:
    DianaApi.getTcpPos(pose1)
```

```
    pose2=cp.deepcopy(pose1)
    pose2[1]+=0.01
    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()
```

74 addMoveJSegmentByTarget

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

在指定 IP 地址机械臂上，往已经创建的路段中插入路点，支持 MoveJ，需要传入点的关节角

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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 JOINT_NUM

vel_percent：移动到目标路点的速度。

acc_percent：移动到目标路点的加速度。

blendradius_percent：交融半径。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
import time
#以 7 轴机械臂为例
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake()
ret=DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_JOINT_PATH,
```

```
ipAddress)
print(ret[0])
if ret[0] == 0:
    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.destroyPath(ret[1] , ipAddress)
```

75 addMoveJSegmentByPose

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

在指定 IP 地址机械臂上，往路段中插入路点，支持 MoveJ，需要传入 TCP 位姿。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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 TCP pose 路点，即该路点的位姿列表，长度为 6。

vel_percent：移动到目标路点的速度。

acc_percent：移动到目标路点的加速度。

blendradius_percent：交融半径。

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
import copy as cp
pose1,pose2=[0]*6,[0]*6
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
```

```
time.sleep(2)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_POSE_PATH,
ipAddress)
if ret[0] == 0:
    DianaApi.getTcpPos(pose1)
    pose2=cp.deepcopy(pose1)
    pose2[1]+=0.01
    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)
    DianaApi.wait_move()
    DianaApi.destroyComplexPath(ret[1],ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

76 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，往路段中插入一段圆弧，支持 MoveC 或 MoveP，需要传入关节角。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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。

acc：移动到目标路点的加速度。

blendradius：交融半径。

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：

```
#以 7 轴机械臂为例
import DianaApi
import time
pose=[0]*6
joints1,joints2=[0]*7,[0]*7
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.MOVEP_JOINT_PATH,
ipAddress)
if ret[0] == 0:
    DianaApi.getTcpPos(pose)
    pose[2]-=0.01
    pose[1]-=0.01
    DianaApi.inverse(pose,joints1)
    pose[2]-=0.01
    pose[1]+=0.01
    DianaApi.inverse(pose,joints2)
    DianaApi.addMoveCSegmentByTarget(ret[1],joints1,joints2,0.1,0.1,0,True, ipAddress)
    DianaApi.runComplexPath(ret[1] , ipAddress)
    DianaApi.wait_move()
    DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

77 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，往路段中插入一段圆弧，支持 MoveC 或 MoveP，需要传入 TCP 位姿。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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。

acc：移动到目标路点的加速度。

blendradius：交融半径。

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
import copy as cp
pose,pose1=[0]*6,[0]*6
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
ret = DianaApi.createComplexPath(DianaApi.complex_path_type.MOVEP_POSE_PATH,
ipAddress)
if ret[0] == 0:
    DianaApi.getTcpPos(pose)
    pose[2]-=0.01
    pose[1]-=0.01
    pose1=cp.deepcopy(pose)
    pose1[2]-=0.01
    pose1[1]+=0.01
    DianaApi.addMoveCSegmentByPose(ret[1],pose,pose1,0.1,0.1,0,True, ipAddress)
    DianaApi.runComplexPath(ret[1] , ipAddress)
    DianaApi.wait_move()
    DianaApi.destroyComplexPath(ret[1] , ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()
```

78 runComplexPath

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

在指定 IP 地址机械臂上，运行已创建的复杂路段。

适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** complex_path_id：输入参数。要运行路段的 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

79 destroyComplexPath

def destroyComplexPath(complex_path_id, ipAddress = "")
在指定 IP 地址机械臂上，销毁已创建的复杂路段。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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

80 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，用于重启机械臂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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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,0,0,0,0,0))
DianaApi.saveEnvironment(ipAddress)
```

81 dumpToUDiskEx

```
def dumpToUDiskEx(timeout_second, 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

timeout_second: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

82 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。

适配臂型：通用医疗臂、定制医疗臂。

参数：

forceDirection：表达力的方向的元组，大小为 3。

forceValue：力大小，单位：N。

maxApproachVelocity：最大接近速度。单位：m/s。

maxAllowTcpOffset：允许的最大偏移距离。单位：m。

active_tcp：用户设定的坐标系的元组，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：

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 = 0.1
active_tcp=[0,0,0,0,0,0]
ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.enterForceMode_ex(force_direction,force_value,max_approach_velocity,max_allow_tcp_offset,active_tcp,ipAddress)
DianaApi.leaveForceMode(DianaApi.mode_e.T_MODE_POSITION)
```

83 readDI

```
def readDI (group_name,io_name, ipAddress = "")
```

读取指定 IP 地址机械臂一个数字输入的值。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

group_name：数字输入的分组，例如，"board","plc"；

io_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：

0：成功。

-1：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"
```

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

84 readAI

```
def readAI (group_name, io_name, ipAddress = "")
```

读取指定 IP 地址机械臂一个模拟输入的值和模式。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

group_name：模拟输入的分组，例如，"board","plc"；

io_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，共计三个元素，第一个元素表示函数调用是否成功：

0：成功。

-1：失败。

第二个元素表示读取的模拟输入的类型，1 表示电流，单位 mA，2 表示电压，单位 V

第三个元素表示模拟输入的值。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.readAI('board','ai0', ipAddress)
if ret[0] == 0:
   print(ret[1])
   print(ret[2])
else:
   print("cannot read ai")
```

85 setAIMode

```
def setAIMode (group_name, io_name,mode, ipAddress = "")
```

设置指定 IP 地址机械臂模拟输入的模式。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

group_name：模拟输入的分组，例如，"board","plc"；

io_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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.setAIMode('board','ai0',mode, ipAddress)
if ret== True:
   print(ret)
else:
   print("cannot set ai mode")
```

86 writeDO

def writeDO (group_name, io_name,value, ipAddress = "")

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

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

group_name：数字输出的分组，例如，"board","plc"；

io_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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
value=1
ret = DianaApi.writeDO('board','do0',value, ipAddress)
if ret == True:
   print(ret)
else:
   print("cannot write DO")
```

87 writeAO

def writeAO (group_name, io_name, mode, value, ipAddress = "")

设置指定 IP 地址机械臂模拟输出的值和模式。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

group_name：模拟输出的分组，例如，"board","plc"；

io_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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.writeAO('board','do0',mode,value, ipAddress)
if ret== True:
   print(ret)
else:
   print("cannot write AO")
```

88 readBusCurrent

def readBusCurrent(ipAddress = "")

读取指定 IP 地址机械臂总线电流。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

0：成功。

-1：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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.readBusCurrent(ipAddress)
if ret[0] == 0:
   print(ret[1])
else:
   print("cannot read bus current")
```

89 readBusVoltage

def readBusVoltage(ipAddress = "")

读取指定 IP 地址机械臂总线电压。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

0：成功。

-1：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"
```

```
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.readBusVoltage(ipAddress)
if ret[0] == 0:
   print(ret[1])
else:
   print("cannot read bus voltage")
```

90 getDH

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

获取指定 IP 地址机械臂的 DH 参数。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

aDH：输入输出参数。连杆长度，长度为 JOINT_NUM 列表

alphaDH:输入输出参数，连杆转角，长度为 JOINT_NUM 的列表

dDH:输入输出参数，连杆偏距，长度为 JOINT_NUM 的列表

thetaDH:输入输出参数，连杆的关节角，长度为 JOINT_NUM 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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
a = [0] * JOINT_NUM
alpha =  [0] * JOINT_NUM
d =  [0] * JOINT_NUM
theta = [0] * JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi. getDH(a,alpha,d,theta, ipAddress)
DianaApi.destroySrv()
```

91 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。

适配臂型：通用医疗臂、定制医疗臂。

参数：

torques：输入输出参数。反馈的扭矩值，长度为 JOINT_NUM 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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
torques = [0] * JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi. getOriginalJointTorque(torques,ipAddress)
DianaApi.destroySrv()
```

92 getJacobiMatrix

```
def getJacobiMatrix (matrix_jacobi, ipAddress = "")
```

获取指定 IP 地址机械臂的雅各比矩阵。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

matrix_jacobi：输入输出参数。雅可比矩阵，长度为 6 * JOINT_NUM 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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
```

```
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
matrix_jacobi =[0] * 6 * JOINT_NUM
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi. getJacobiMatrix (matrix_jacobi,ipAddress)
DianaApi.destroySrv()
```

93 resetDH

def resetDH(ipAddress = "")

重置指定 IP 地址机械臂用户自定义 DH 参数。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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,0,0,0,0,0))
DianaApi. resetDH(ipAddress)
DianaApi.destroySrv()
```

94 runProgram

def runProgram(name, ipAddress = "")

运行指定 IP 地址机械臂某个示教器程序。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.runProgram("AgileRobots", ipAddress)
if ret == False:
   print("run Program failed!")
else:
   print("run program succeed!")
```

95 stopProgram

```
def stopProgram(name, ipAddress = "")
```

停止指定 IP 地址机械臂某个示教器程序。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.stopProgram("AgileRobots", ipAddress)
if ret == False:
   print("stop Program failed!")
else:
   print("stop program succeed!")
```

96 getVariableValue

```
def getVariableValue(name, ipAddress = "")
```

获取指定 IP 地址机械臂某个全局变量的值。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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。

第二个元素表示获取的全局变量的值。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
msg, value = DianaApi.getVariableValue("GLOBAL", ipAddress)
```

97 setVariableValue

```
def setVariableValue(name,value, ipAddress = "")
```

设置指定 IP 地址机械臂某个全局变量的值。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setVariableValue("GLOBAL",1, ipAddress)
```

98 isTaskRunning

```
def isTaskRunning(name, ipAddress = "")
```

判断指定 IP 地址机械臂某个示教器程序是否在运行。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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.initSrv((ipAddress,0,0,0,0,0))
DianaApi.isTaskRunning("AgileRobots", ipAddress)
```

99 pauseProgram

```
def pauseProgram(ipAddress = "")
```

暂停指定 IP 地址机械臂所有示教器程序。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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,0,0,0,0,0))
DianaApi.pauseProgram(ipAddress)
```

100 resumeProgram

```
def resumeProgram(ipAddress = "")
```

恢复运行指定 IP 地址机械臂已经暂停的示教器程序。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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,0,0,0,0,0)) DianaApi.resumeProgram(ipAddress)

101 stopAllProgram

def stopAllProgram(ipAddress = "")
停止指定 IP 地址机械臂所有示教器程序。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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,0,0,0,0,0)) DianaApi.stopAllProgram(ipAddress)

102 isAnyTaskRunning

def isAnyTaskRunning(ipAddress = "")
判断指定 IP 地址机械臂是否有示教器程序在运行。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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,0,0,0,0,0))
DianaApi.isAnyTaskRunng(ipAddress)
```

103 cleanErrorInfo

```
def cleanErrorInfo(ipAddress = "")
```

清除指定 IP 地址机械臂的错误信息。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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,0,0,0,0,0))
DianaApi.cleanErrorInfo(ipAddress)
```

104 setCollisionLevel

```
def setCollisionLevel(level, ipAddress = "")
```

设置指定 IP 地址机械臂的碰撞检测类型。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

level：level：碰撞等级，整数，值为 0 或者下面几种值的组合（0 表示只考虑后台最大保护阈值）：

- 1：关节空间检测
- 2：笛卡尔空间检测
- 4： 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.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setCollisionLevel(7, ipAddress)
```

105 zeroSpaceManualMove

def zeroSpaceManualMove (direction, jointsVel, jointsAcc, ipAddress = "")

控制指定 IP 地址机械臂进入零空间手动驱动模式。

适配臂型：通用医疗臂、定制医疗臂。

参数：

direction：输入参数，控制零空间手动运动的方向，须传入 DianaApi 中的枚举 zero_space_move_direction，枚举值和含义如下，

- E_FORWARD 表示顺时针运动；
- E_BACKWARD 表示逆时针运动。

jointsVel：输入参数，各关节运动的速度，大小为 JOINT_NUM 的列表，单位 rad/s

jointsAcc：输入参数，各关节运动的加速度，大小为 JOINT_NUM 的列表，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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
jointVel=[0.1] * JOINT_NUM
jointAcc=[0.1] * JOINT_NUM
ret = DianaApi.zeroSpaceManualMove(DianaApi.zero_space_move_direction.E_BACKWARD,
jointVel, jointAcc, ipAddress)
print(ret)
if  not ret :
   print("Diana API zeroSpaceManualMove failed!\n")
else:
   time.sleep(0.001)
   DianaApi.stop()
```

106 getHomingState

def getHomingState(ipAddress = "")

针对指定 IP 地址的定制医疗臂 V1.0，在其控制箱每次上电后，第七关节需要进行零位状态查询。并根据返回状态决定是否要进行寻零操作。

适配臂型：定制医疗臂 1.0。

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：

3：当前不在零位（需要寻零）。

2：寻零过程中出错。

1：寻零进行中。

0：当前已在零位（寻零成功）。

-1：调用失败。

调用示例：

```
import DianaApi

ipAddress="192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.releaseBrake()
time.sleep(2)
if ret < 0:
  error_code = DianaApi.getLastError()
  msg = DianaApi.formatError(error_code)
  print("release brake failed(%s)\n" %(msg))
  state = DianaApi.getHomingState()
  print("getHomingState = %d\n"%(state))
  if state!=0:
    ret=DianaApi.startHoming()
    print("startHoming = %d\n"%(ret))
    cnt = 0
    while state != 0:
      time.sleep(0.1)
      state = DianaApi.getHomingState()
      print("getHomingState(%d) = %d\n"%(cnt, state))
      if cnt == 60000:
        break
      else:
        cnt = cnt + 1
    ret = DianaApi.stop()
    print("stop = %d\n"%(ret))
    state = DianaApi.getHomingState()
```

```
    print("getHomingState(%d) = %d\n"%(cnt, state))
ret = DianaApi.releaseBrake()
print("releaseBrake again = %d\n"%(ret))
time.sleep(2)
```

107 startHoming

def startHoming(ipAddress = "")

对于指定 IP 的地址的定制机械臂 V1.0，在每次其控制箱上电后，第七关节需要进行寻零操作时调用该函数启动寻零程序。需要与 stop 函数配对使用。当寻零结束或失败后，调用 stop 函数。当七关节不在零位时，调用 *releaseBrake* 开报闸会失败，*getLastError* 会返回 *ERROR_CODE_HOME_POSITION_ERROR*（*-2305*）

适配臂型：定制医疗臂 1.0。

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：

参照 getHomingState 中示例。

108 setEndKeyEnableState

def setEndKeyEnableState(enable,ipAddress = " ")

在指定 IP 地址的机械臂上，使能或禁用末端按键与进入或退出自由驱动的绑定。

适配臂型：通用医疗臂、Thor3。

参数：

enable：bool 类型，使能或禁用末端按键进入自由驱动的绑定。

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,0,0,0,0,0))
ret = DianaApi.setEndKeyEnableState (False, ipAddress)
if not ret:
   print("setEndKeyEnableState failed!\n")
```

109 updateForce

def updateForce(forceValue, ipAddress = "")

针对指定 IP 地址机械臂上，在力控模式下，实时改变力值的大小。

适配臂型：通用医疗臂、定制医疗臂。

参数：

forceValue：输入参数，力的大小。

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 = 0
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
#以 7 轴机械臂为例
joints = [0,PI/6,0,PI/2,0,-PI/3,0]
DianaApi.moveJToTarget(joints,0.2,0.2,ipAddress)
DianaApi.wait_move()
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
    DianaApi.updateForce(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)
```

110 getCartImpedanceCoordinateType

```
def getCartImpedanceCoordinateType(ipAddress = "")
```

在指定 IP 地址机械臂上，获取设置笛卡尔阻抗时选用的坐标系种类。

适配臂型：通用医疗臂、定制医疗臂。

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：

坐标系的类型，0：基坐标系， 1：工具坐标系

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.getCartImpedanceCoordinateType(ipAddress)
if ret == 0:
    print("it's base coordinate\n")
else:
    print("it's tool coordinate\n")
```

111 setCartImpedanceCoordinateType

```
def setCartImpedanceCoordinateType(intCoordinateType, ipAddress = "")
```

在指定 IP 地址机械臂上，选择设置笛卡尔阻抗时使用的坐标系种类。

适配臂型：通用医疗臂、定制医疗臂。

参数：

intCoordinateType：坐标系的类型，0：基坐标系， 1：工具坐标系。

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
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setCartImpedanceCoordinateType(0,ipAddress)
```

112 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。

适配臂型：通用医疗臂、定制医疗臂。

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。

lock_joint_index：输入参数，被约束的关节索引。

lock_joint_position：输入参数，被约束关节的角度，单位:rad。

ref_joints：参考的关节角，大小为 JOINT_NUM 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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7。

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'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#以 7 轴机械臂为例
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_tcp,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]))
    ret = DianaApi.destoryInverseClosedItems(id)
else:
    print("cannot inverse.\n")
DianaApi.destroySrv(ipAddress)
```

113 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。

适配臂型：通用医疗臂、定制医疗臂。

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

114 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。 适配臂型：通用医疗臂、定制医疗臂。 **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：** True：成功； False：失败。
调用示例： 见 inverseClosedFull 示例

115 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。 适配臂型：通用医疗臂、定制医疗臂。 **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：** True：成功。 False：失败。
调用示例：

见 inverseClosedFull 示例

116 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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** matrixJacobi：输出参数，雅各比矩阵，大小为 6*JOINT_NUM 的列表。 joints：输入参数，用于计算雅各比矩阵的关节角（该关节角与机械臂当前反馈的位姿无关），大小为 JOINT_NUM 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。 注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。 **返回值：** True：成功。 False：失败。

调用示例：
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
jointCount = JOINT_NUM
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")

117 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 。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

matrixJacobi：输出参数，雅各比矩阵，大小为 6*JOINT_NUM 的列表。

joints：输入参数，用于计算雅各比矩阵的关节角（该关节角与机械臂当前反馈的位姿无关），大小为 JOINT_NUM 的列表。

toolMatrix: 工具坐标系对应的位姿矢量，长度为 6 的列表，前三个元素表示工具坐标系的平移矢量，后三个元素表示工具坐标系的旋转矢量。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
jointCount = JOINT_NUM
tcpCount = 6
jacobiMatrix = [0.0] *(jointCount*tcpCount)
jointPosition = [0.0] * jointCount
toolMatrix=[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")
```

118 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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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：** True：成功。 False：失败。

调用示例：
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
forces = [0.0]*6
ret = DianaApi.getTcpForceInToolCoordinate(forces,ipAddress)
print(ret)
if not ret:
  print("getTcpForceInToolCoordinate failed! Return value = %d"%(ret))
else:
  print(forces)

119 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，机械臂将在 t 时间后减速。如果没有提供时间 t 值，机械臂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。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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。 acc：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 import time ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.releaseBrake(ipAddress) speeds = [0.1,0.0,0.0,0.0,0.0,0.0] acc = [0.1, 0.50] DianaApi.speedLOnTcp(speeds, acc, 0, ipAddress) time.sleep(1) DianaApi.stop() DianaApi.destroySrv()

120 inverseClosedIdeal

def inverseClosedIdeal(pose, lock_robot_arm_angle, ref_joints, active_tcp, ipAddress='')
根据末端位姿，在固定臂角的情况下，求更接近某组关节位置的逆解集合。现支持在不同工具坐标系下计算，由传入的 active_tcp 决定。 适配臂型：定制医疗臂。 **参数：** pose：输入末端位姿，元组大小为 6。包含 TCP 坐标（x, y, z）和轴角（rx, ry, rz）组合的矢量数据。pose 可以相对于不同工具坐标系，由传入的 active_tcp 决定。 lock_robot_arm_angle：输入参数，臂角，单位：rad。 ref_joints：输入参数，逆解集合参照的关节角，元组大小为 JOINT_NUM，单位：rad。 active_tcp：工具坐标系对应的位姿向量，大小为 6 的元组，为空时，将使用默认的工具坐标系 default_tcp。 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：生成逆解集合对应的 ID。 -1：失败。
调用示例： def to_rad(x): return x*math.pi / 180.0 import DianaApi

```
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
#&&&&&&&&&&
start_joints = (to_rad(0), to_rad(30), to_rad(0), to_rad(90), to_rad(0), to_rad(10), to_rad(0))
tool = (0.1,0.1,0.1,0,0,0)
DianaApi.moveJToTarget(start_joints, 0.2,0.4, ipAddress)
DianaApi.wait_move()
DianaApi.stop(ipAddress)
pose = [0,0,0,0,0,0]
DianaApi.forward(start_joints, pose, ipAddress)
print("forward:")
[print(j) for j in pose]
id = DianaApi.inverseClosedIdeal(pose,to_rad(30), start_joints,tool, ipAddress)
print("inverseClosedIdeal:")
print(id)
size = DianaApi.getInverseClosedIdealResultSize(id, ipAddress)
print("getInverseClosedIdealResultSize:")
print(size)
for i in range(0, size):
  retJoints = [0,0,0,0,0,0,0]
  ret= DianaApi.getInverseClosedIdealJoints(id, i, retJoints, ipAddress)
  print("getInverseClosedIdealJoints return:", ret)
  [print(j) for j in retJoints]
  retPose = [0,0,0,0,0,0]
  DianaApi.forward(retJoints, retPose, ipAddress)
  print("forward:")
  [print(j) for j in retPose]
print("destoryInverseClosedIdealItems:")
ret = DianaApi.destoryInverseClosedIdealItems(id, ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

121 getInverseClosedIdealResultSize

def getInverseClosedIdealResultSize(id, ipAddress='')
在指定 IP 地址的机械臂上，根据 ID 获取约束臂角求逆解结果的组数。 适配臂型：定制医疗臂。

参数：

id：输入参数，所求逆解集合对应的 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 inverseClosedIdeal 示例。

122 getInverseClosedIdealJoints

def getInverseClosedIdealJoints(id, index, joints, ipAddress='')

在指定 IP 地址机械臂上，根据 ID 按索引获取约束臂角求逆解集合中对应关节角。

适配臂型：定制医疗臂。

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，单位：rad。

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 inverseClosedIdeal 示例。。

123 destoryInverseClosedIdealItems

def destoryInverseClosedIdealItems(id, ipAddress='')

在指定 IP 地址机械臂上，根据 ID 删除约束臂角求逆解的结果数据集。

适配臂型：定制医疗臂。

参数：

id：输入参数，逆解集合对应的 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：

True：成功 False：失败。
调用示例： 见 inverseClosedIdeal 示例。

124 getWayPoint

def getWayPoint(waypointName, dblTcppos, dblJoints, strIpAddress='')
获取指定 IP 地址机械臂上的路点变量信息。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** waypointName：路点变量名称 。 dblTcppos：位姿信息，元组大小为 6，其中，后三个角度为轴角。 dblJoints：关节角信息。 strIpAddress：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,0,0,0,0,0))
dblTcppos= [0] * 6
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
dblJoints=[0] * JOINT_NUM
ret=DianaApi.getWayPoint("api_1",dblTcppos,dblJoints,ipAddress)
print(ret)
if not ret:
  print("getWayPoint failed!Return value = %d\n" %(ret))

125 setWayPoint

def setWayPoint(waypointName, dblTcppos, dblJoints, strIpAddress='')
修改指定 IP 地址机械臂上路点变量的值。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** waypointName：路点变量名称 。

dblTcppos：位姿信息，元组大小为 6，其中，后三个角度为轴角。

dblJoints：关节角信息。

strIpAddress：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：

```
def to_rad(x):
    return x * math.pi / 180.0
import DianaApi
#以 7 轴机械臂为例
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
rpy=[to_rad(19.3), to_rad(-0.118), to_rad(10)]
DianaApi.rpy2Axis(rpy)
dblTcppos1=[0.643,0.4607,0.38862,rpy[0], rpy[1], rpy[2]]
#以 7 轴机械臂为例
dblJoint1=[to_rad(-22.67),to_rad(37.27),to_rad(-12.11),to_rad(77.43),to_rad(-10.88),to_rad(-56.03),to_rad(-26.19)]
ret=DianaApi.setWayPoint('api_1',dblTcppos1,dblJoint1,ipAddress)
if not ret :
    print("setWayPoint failed!Return value = %d\n" %(ret))
```

126 addWayPoint

```
def addWayPoint(waypointName, dblTcppos, dblJoints, ipAddress='')
```

在指定 IP 地址机械臂上新增路点变量。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

waypointName：路点变量名称 。

dblTcppos：位姿信息，元组大小为 6，其中，后三个角度为轴角。

dblJoints：输入参数，关节角信息，大小为 JOINT_NUM 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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
def to_rad(x):
    return x*math.pi / 180.0
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
rpy=[to_rad(19.3), to_rad(-0.118), to_rad(0.03)]
DianaApi.rpy2Axis(rpy)
dblTcppos2=[0.472, 0.027, 0.293,rpy[0], rpy[1], rpy[2]]
#以 7 轴机械臂为例
dblJoints2=[to_rad(2),to_rad(0.7),to_rad(-5.64),to_rad(142.98),to_rad(-31.29),to_rad(-39.27),to_rad(21.38)]
ret=DianaApi.addWayPoint('api_1',dblTcppos2, dblJoints2,ipAddress )
print(ret)
if not ret:
    print("addWayPoint failed!Return value = %d\n" %(ret))
```

127 deleteWayPoint

```
def deleteWayPoint(waypointName, ipAddress='')
```

删除指定 IP 地址机械臂上的某个路点变量。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret=DianaApi.deleteWayPoint('api_1',ipAddress )
```

```
if not ret :
    print("deleteWayPoint failed!Return value = %d\n" %(ret))
```

128 getSixAxisZeroOffset

```
def getSixAxisZeroOffset(offset, ipAddress='')
```

获取指定 IP 地址机械臂上六维力传感器的零偏数据。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

offset：输出参数，用于接收六维力传感器零偏数据，元组大小为 10。零偏数据的含义依次为：m、mx、my、mz、x、y、z、rx、ry、rz。其中：

m 为六维力传感器承载的且能感知到的负载质量，单位：Kg；

mx、my、mz 为六维力负载的质心坐标（在六维力自身坐标系下的表达），单位：m；

x、y、z、rx、ry、rz 为六维力在各个维度的零偏值，其中 x、y、z 的单位为 N，rx、ry、rz 的单位为 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：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
offset = [0.0]*10
DianaApi.getSixAxisZeroOffset(offset, ipAddress)
print("getSixAxisZeroOffset:"+str(offset))
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

129 setSixAxisZeroOffset

```
def setSixAxisZeroOffset(offset, ipAddress='')
```

设置指定 IP 地址机械臂上六维力传感器的零偏数据。设置后需调用 saveEnvironment 函数才能使设置永久生效。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

offset：输入参数，用于存放零偏数据，元组大小为 10。零偏数据的含义依次为：m、mx、my、mz、x、y、z、rx、ry、rz。其中：

m 为六维力传感器承载的且能感知到的负载质量，单位：Kg；

mx、my、mz 为六维力负载的质心坐标（在六维力自身坐标系下的表达），单位：m；

x、y、z、rx、ry、rz 为六维力在各个维度的零偏值，其中 x、y、z 的单位为 N，rx、ry、rz 的单位为 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：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
offset = (0.47, 0.0,0.006,0.036,-6.644,-11.771,-1.627,-0.564,0.616,0.007)
DianaApi.setSixAxisZeroOffset(offset, ipAddress)
DianaApi.saveEnvironment(ipAddress)
print("setSixAxisZeroOffset:"+str(offset))
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

130 setDefaultActiveWorkpiece

```
def setDefaultActiveWorkpiece(defaultWorkpiece, ipAddress='')
```

设置指定 IP 地址机械臂的默认工件坐标系。设置后需调用 saveEnvironment 函数才能使设置永久生效。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

defaultWorkpiece：输入参数，工件坐标系相对于基坐标系的 4*4 齐次变换矩阵，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
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
matrix = (1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)
ret=DianaApi.setDefaultActiveWorkpiece(matrix, ipAddress)
print(ret)
if not ret:
   print("setDefaultActiveWorkpiece failed!Return value = %d\n" %(ret))
DianaApi.saveEnvironment(ipAddress)
```

131 setDefaultActiveWorkpiecePose

```
def setDefaultActiveWorkpiecePose(defaultWorkpiecePose, ipAddress = '')
```

设置指定 IP 地址机械臂的默认工件坐标系。设置后需调用 saveEnvironment 函数才能使设置永久生效。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

defaultWorkpiecePose：输入参数，工件坐标系相对于基坐标系位姿向量，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：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
tool = (0.1,0.1,0.1,0,0,0)
```

```
ret=DianaApi.setDefaultActiveWorkpiecePose(tool, ipAddress)
print(ret)
if not ret:
   print("setDefaultActiveWorkpiecePose failed!Return value = %d\n" %(ret))
```

132 getDefaultActiveWorkpiece

def getDefaultActiveWorkpiece(defaultWorkpiece, ipAddress='')

获取指定 IP 地址机械臂上默认工件坐标系相对于基坐标系的 4*4 齐次变换矩阵。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

defaultWorkpiece：输出参数，工件坐标系相对于基坐标系的 4*4 齐次变换矩阵，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
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
matrix = [0.0]*16
DianaApi.getDefaultActiveWorkpiece(matrix, ipAddress)
print("getDefaultActiveWorkpiece:"+str(matrix))
```

133 getDefaultActiveWorkpiecePose

def getDefaultActiveWorkpiecePose(defaultWorkpiecePose, ipAddress = '')

获取指定 IP 地址机械臂上默认工件坐标系相对于基坐标系的转换位姿向量。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

defaultWorkpiecePose：输出参数，工件坐标系相对于基坐标系位姿向量，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：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
tool = [0.0]*6
DianaApi.getDefaultActiveWorkpiecePose(tool, ipAddress)
print("getDefaultActiveWorkpiecePose:"+str(tool))
```

134 setControllerFeatureCode

```
def setControllerFeatureCode(feature_code, enable, ipAddress='')
```

在指定 IP 地址机械臂上禁用或使能某项控制器的特性。设置后需调用 saveEnvironment 函数才能使设置永久生效。

目前不支持该功能。

参数：

feature_code：输入参数，controller_feature_e 枚举类型变量。目前仅支持 NONE_FEATURE（int 值为 0）代表无特性；AUTO_SWITCH_TO_IMPEDANCE_MODE_WHEN_COLLISION_DETECTED （int 值为 1）碰撞后退出到阻抗模式特性。

enable：输入参数，bool 型变量，使能或禁用 intFeatureCode 代表的特性。

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,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
DianaApi.setControllerFeatureCode(DianaApi.controller_feature_e.AUTO_SWITCH_TO_IMPEDANCE_MODE_WHEN_COLLISION_DETECTED, True, ipAddress)
```

```
print("setControllerFeatureCode
AUTO_SWITCH_TO_IMPEDANCE_MODE_WHEN_COLLISION_DETECTED")
DianaApi.saveEnvironment(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

135 testControllerFeature

```
def testControllerFeature(feature_code, ipAddress='')
```

验证指定 IP 地址机械臂是否具备某项特性。

目前不支持该功能。

参数：

feature_code：输入参数，controller_feature_e 枚举类型变量。目前仅支持 NONE_FEATURE（int 值为 0）代表无特性；AUTO_SWITCH_TO_IMPEDANCE_MODE_WHEN_COLLISION_DETECTED （int 值为 1）碰撞后退出到阻抗模式特性。

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：

要查询特性的使能状态

True：使能。

False：禁用。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
bEnabled =
DianaApi.testControllerFeature(DianaApi.controller_feature_e.AUTO_SWITCH_TO_IMPED
ANCE_MODE_WHEN_COLLISION_DETECTED, ipAddress)
print("AUTO_SWITCH_TO_IMPEDANCE_MODE_WHEN_COLLISION_DETECTED
enabled = "+str(bEnabled))
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

136 enableSixAxis

```
def enableSixAxis(six_axis_in_mode, enable, ipAddress='')
```

是否使能指定 IP 地址机械臂上六维力零偏参数。设置后需调用 saveEnvironment 函数才

能使设置永久生效。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

six_axis_in_mode：输入参数，six_axis_type_e 枚举变量。目前仅支持 SIX_AXIS_IN_FREE_DRIVING （零力驱动模式下使能或禁用六维力参数）；未来可能会支持 SIX_AXIS_IN_FORCE_MODE（力控模式下使能或禁用六维力参数）、SIX_AXIS_IN_JOINT_IMPEDANCE（关节空间阻抗中使能或禁用六维力参数）和 SIX_AXIS_IN_CART_IMPEDANCE（笛卡尔空间阻抗中使能或禁用六维力参数）。

enable：既是输入又是输出参数，bool 型变量，使能或禁用六维力参数在 type 模式中的使用，同时在函数调用后返回当前系统状态，可用于判断是否设置成功。

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：

当前系统状态，可用于判断是否设置成功。

True：使能。

False：禁用。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
bRet =
DianaApi.enableSixAxis(DianaApi.six_axis_type_e.SIX_AXIS_IN_FORCE_MODE,True,
ipAddress)
print("enableSixAxis true => return " + str(bRet))
DianaApi.saveEnvironment(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

137 isSixAxisEnabled

```
def isSixAxisEnabled(six_axis_in_mode, ipAddress='')
```

判断指定 IP 地址机械臂上六维力零偏参数是否处于使能状态。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

six_axis_in_mode：输入参数，six_axis_type_e 型枚举变量。目前仅支持 SIX_AXIS_IN_FREE_DRIVING（零力驱动模式下使能或禁用六维力参数）；未来可能会支持 SIX_AXIS_IN_FORCE_MODE（力控模式下使能或禁用六维力参数）、SIX_AXIS_IN_JOINT_IMPEDANCE（关节空间阻抗中使能或禁用六维力参数）和 SIX_AXIS_IN_CART_IMPEDANCE（笛卡尔空间阻抗中使能或禁用六维力参数）。

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：使能。

False：禁用。

调用示例：

```
import DianaApi
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
bRet =
DianaApi.isSixAxisEnabled(DianaApi.six_axis_type_e.SIX_AXIS_IN_FORCE_MODE,
ipAddress)
print("isSixAxisEnabled " + str(bRet))
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)
```

138 getMechanicalJointsPositionRange

```
def getMechanicalJointsPositionRange(arrMin, arrMax, ipAddress = '')
```

获取指定 IP 地址机械臂各关节角的机械限位。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrMin：输出参数，JOINT_NUM 个轴关节角度的最小机械限位元组。单位：rad。

arrMax：输出参数，JOINT_NUM 个轴关节角度的最大机械限位元组。单位：rad。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。 False：失败。

调用示例：
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
minPos = [0.0]* JOINT_NUM
maxPos = [0.0]* JOINT_NUM
DianaApi.getMechanicalJointsPositionRange(minPos, maxPos, ipAddress)
print("getMechanicalJointsPositionRange::minPos="+str(minPos))
print("getMechanicalJointsPositionRange::maxPos="+str(maxPos))

139 getMechanicalMaxJointsVel

def getMechanicalMaxJointsVel(arrVel, ipAddress = '')
获取指定 IP 地址机械臂各关节最大机械速度。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** arrVel：输出参数，JOINT_NUM 个轴关节角度的最大机械速度元组。单位：rad/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。 注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。 **返回值：** True：成功。 False：失败。

调用示例：
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
maxVel = [0.0]* JOINT_NUM
DianaApi.getMechanicalMaxJointsVel(maxVel, ipAddress)
print("getMechanicalMaxJointsVel::maxVel="+str(maxVel))

140 getMechanicalMaxJointsAcc

def getMechanicalMaxJointsAcc(arrAcc, ipAddress = '')
获取指定 IP 地址机械臂各关节最大机械加速度。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** arrAcc：输出参数，JOINT_NUM 个轴关节角度的最大机械加速度元组。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。 注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。 **返回值：** True：成功。 False：失败。

调用示例：
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
maxAcc = [0.0]* JOINT_NUM
DianaApi.getMechanicalMaxJointsAcc(maxAcc, ipAddress)
print("getMechanicalMaxJointsAcc::maxAcc="+str(maxAcc))

141 getMechanicalMaxCartVelAcc

def getMechanicalMaxCartVelAcc(ipAddress = '')
获取指定 IP 地址笛卡尔空间最大机械平移速度、最大机械旋转速度、最大机械平移加速度和最大机械旋转加速度。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **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，共计五个元素 （1）函数调用是否成功： True：成功。 False：失败。 （2）笛卡尔空间最大机械平移速度，double 型变量，单位：m/s。

（3）笛卡尔空间最大机械旋转速度，double 型变量，单位：rad/s。

（4）笛卡尔空间最大机械平移加速度，double 型变量，单位：m/s^2。

（5）笛卡尔空间最大机械旋转加速度，double 型变量，单位：rad/s^2。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret,dblTranslationVel, dblRotationVel, dblTranslationAcc, dblRotationAcc =
DianaApi.getMechanicalMaxCartVelAcc(ipAddress)
print("getMechanicalCartMaxVelAcc::dblTranslationVel="+str(dblTranslationVel)+"dblRotati
onVel="+str(dblRotationVel)+"dblTranslationAcc="+str(dblTranslationAcc)+"dblRotationAcc
="+str(dblRotationAcc))
```

142 getJointsPositionRange

```
def getJointsPositionRange(arrMin, arrMax, ipAddress = '')
```

获取指定 IP 地址机械臂各关节的极限位置。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrMin：输出参数，JOINT_NUM 个轴关节角度的极限位置下限组成的元组。单位：rad。

arrMax：输出参数，JOINT_NUM 个轴关节角度的极限位置上限组成的元组。单位：rad。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
minPos = [0.0]* JOINT_NUM
maxPos = [0.0]* JOINT_NUM
DianaApi.getJointsPositionRange(minPos, maxPos, ipAddress)
print("getJointsPositionRange::minPos="+str(minPos))
print("getJointsPositionRange::maxPos="+str(maxPos))
```

143 getMaxJointsVel

```
def getMaxJointsVel(arrVel, ipAddress = '')
```

获取指定 IP 地址机械臂各关节最大速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrVel：输出参数，JOINT_NUM 个轴关节角度的最大速度组成的元组。单位：rad/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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
maxVel = [0.0]* JOINT_NUM
DianaApi.getMaxJointsVel(maxVel, ipAddress)
print("getMaxJointsVel::maxVel="+str(maxVel))
```

144 getMaxJointsAcc

```
def getMaxJointsAcc(arrAcc, ipAddress = '')
```

获取指定 IP 地址机械臂各关节最大加速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrAcc：输出参数，JOINT_NUM 个轴关节角度的最大加速度元组。单位：rad/s2。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))

#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
maxAcc = [0.0]* JOINT_NUM
DianaApi.getMaxJointsAcc(maxAcc, ipAddress)
print("getMaxJointsAcc::maxVel="+str(maxAcc))
```

145 getMaxCartTranslationVel

```
def getMaxCartTranslationVel(ipAddress = '')
```

获取指定 IP 地址机械臂笛卡尔空间最大平移速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

（1）函数调用是否成功：

True：成功。

False：失败

（2）笛卡尔空间最大平移速度，double 型变量，单位：m/s。

调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
```

```
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret, dblTranslationVel = DianaApi.getMaxCartTranslationVel(ipAddress)
print("getMaxCartRotationAcc::dblTranslationVel="+str(dblTranslationVel))
```

146 getMaxCartRotationVel

```
def getMaxCartRotationVel(ipAddress = '')
```

获取指定 IP 地址机械臂笛卡尔空间最大旋转速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

（1）函数调用是否成功：

True：成功。

False：失败

（2）笛卡尔空间最大旋转速度，double 型变量，单位：rad/s。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret, dblRotationVel = DianaApi.getMaxCartRotationVel(ipAddress)
print("getMaxCartRotationVel::dblRotationVel="+str(dblRotationVel))
```

147 getMaxCartTranslationAcc

```
def getMaxCartTranslationAcc(ipAddress = '')
```

获取指定 IP 地址机械臂笛卡尔空间最大平移加速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

（1）函数调用是否成功：

True：成功。

False：失败

（2）笛卡尔空间最大平移加速度，double 型变量，单位：m/s^2。

调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))

ret, dblTranslationAcc = DianaApi.getMaxCartTranslationAcc( ipAddress)
print("getMaxCartTranslationAcc::dblTranslationAcc="+str(dblTranslationAcc))
```

148 getMaxCartRotationAcc

```
def getMaxCartRotationAcc(ipAddress = '')
```

在指定 IP 地址机械臂笛卡尔空间最大旋转加速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

（1）函数调用是否成功：

True：成功。

False：失败。

（2）笛卡尔空间最大旋转加速度，double 型变量，单位：rad/s^2。

调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))

ret, dblRotationAcc = DianaApi.getMaxCartRotationAcc( ipAddress)
print("getMaxCartRotationAcc::dblRotationAcc="+str(dblRotationAcc))
```

149 setJointsPositionRange

```
def setJointsPositionRange(arrMinPos, arrMaxPos, ipAddress = '')
```

设置指定 IP 地址机械臂各关节极限位置。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrMin：输出参数，JOINT_NUM 个轴关节角度的极限位置下限组成的元组。单位：rad。

arrMax：输出参数，JOINT_NUM 个轴关节角度的极限位置上限组成的元组。单位：rad。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
minPos = [0.0]* JOINT_NUM
maxPos = [0.0]* JOINT_NUM
DianaApi.getJointsPositionRange(minPos, maxPos, ipAddress)
for i in range(0, JOINT_NUM):
    minPos[i] = minPos[i] + 0.01
    maxPos[i] = maxPos[i] - 0.01
DianaApi.setJointsPositionRange(minPos, maxPos, ipAddress)
DianaApi.saveEnvironment(ipAddress)
```

150 setMaxJointsVel

```
def setMaxJointsVel(arrVel, ipAddress = '')
```

设置指定 IP 地址机械臂各关节的最大速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrVel：输入参数，JOINT_NUM 个轴关节角度的最大速度元组。单位：rad/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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
maxVel = [0.0]* JOINT_NUM
DianaApi.getMaxJointsVel(maxVel, ipAddress)
```

```
for i in range(0, JOINT_NUM):
      maxVel[i] = maxVel[i]-2.0
DianaApi.setMaxJointsVel(maxVel, ipAddress)
DianaApi.saveEnvironment(ipAddress)
```

151 setMaxJointsAcc

def setMaxJointsAcc(arrAcc, ipAddress = '')

设置指定 IP 地址机械臂各关节的最大加速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

arrAcc：输入参数，JOINT_NUM 个轴关节角度的最大加速度元组。单位：rad/s2。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
maxAcc = [0.0]* JOINT_NUM
DianaApi.getMaxJointsAcc(maxAcc, ipAddress)
for i in range(0, JOINT_NUM):
   maxAcc[i] = maxAcc[i] - 2.0
DianaApi.setMaxJointsAcc(maxAcc, ipAddress)
DianaApi.saveEnvironment(ipAddress)
```

152 setMaxCartTranslationVel

def setMaxCartTranslationVel(TranslationVel, ipAddress = '')

设置指定 IP 地址机械臂笛卡尔空间最大平移速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

TranslationVel：输入参数，笛卡尔空间最大平移速度，double 型变量，单位：m/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： ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0,0,0)) cartTransVel = 1.0 DianaApi.setMaxCartTranslationVel(cartTransVel, ipAddress) DianaApi.saveEnvironment(ipAddress)

153 setMaxCartRotationVel

def setMaxCartRotationVel(RotationVel, ipAddress = '')
设置指定 IP 地址机械臂笛卡尔空间最大旋转速度。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** RotationVel：输入参数，笛卡尔空间最大旋转速度，double 型变量，单位：rad/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,0,0,0,0,0)) cartRotVel = 1.0 DianaApi.setMaxCartRotationVel(cartRotVel, ipAddress) DianaApi.saveEnvironment(ipAddress)

154 setMaxCartTranslationAcc

def setMaxCartTranslationAcc(lineAcc, ipAddress = '')
设置指定 IP 地址机械臂笛卡尔空间最大平移加速度。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** lineAcc：输入参数，笛卡尔空间最大平移加速度，double 型变量，单位：m/s2。 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,0,0,0,0,0))

cartTransAcc = 1.0
DianaApi.setMaxCartTranslationAcc(cartTransAcc, ipAddress)
DianaApi.saveEnvironment(ipAddress)
```

155 setMaxCartRotationAcc

def setMaxCartRotationAcc(rotationAcc, ipAddress = '')

设置指定 IP 地址机械臂笛卡尔空间最大旋转加速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

rotationAcc：输入参数，笛卡尔空间最大旋转加速度，double 型变量，单位：rad/s2。

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,0,0,0,0,0))

cartRotAcc = 1.0
DianaApi.setMaxCartRotationAcc(cartRotAcc, ipAddress)
DianaApi.saveEnvironment(ipAddress)
```

156 freeDriving_ex

def freeDriving_ex(enable, frame_type, force_direction, ipAddress = '')

实现控制指定 IP 地址的机械臂正常模式与定向零力拖动模式之间的切换。定向零力拖动是指可选择笛卡尔空间可拖动方向的轴空间零力拖动。

适配臂型：通用医疗臂、定制医疗臂。

参数：

enable：输入参数，bool 变量，是否进入定向零力拖动模式，True 表明进入定向零力拖动，False 为退出定向零力拖动进入正常模式。只有在正常模式下，才可以通过程序控制

机械臂运动。

frame_type：输入参数，int 型变量，用于指定在哪个坐标系下选择可拖动方向。

- 0：不限制拖动方向，等同于关节空间虚拟墙；
- 1：在基坐标系下描述可拖动方向；
- 2：在工具坐标系下描述可拖动方向；
- 3：在工件坐标系下描述可拖动方向。

force_direction：输入参数，6 个表达力的方向（三个平移维度和三个旋转维度）的元组。非“0”代表对应维度可拖动，“0”代表对应维度不可拖动。

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

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake()
direction = [0, 0, 1, 0, 0, 0]
ch = input('enter free driving ex')
DianaApi.freeDriving_ex(True, 1, direction, ipAddress)
ch = input('leave free driving ex')
DianaApi.freeDriving_ex(False, 1, direction, ipAddress)
```

157 freeDrivingWithLockedJoint

def freeDrivingWithLockedJoint(enable, indexLockedJoint = -1, ipAddress = '')

实现控制指定 IP 地址的机械臂正常模式与锁轴零力拖动模式之间的切换。

适配臂型：通用医疗臂、定制医疗臂。

参数：

enable：输入参数，bool 变量，是否进入锁轴零力拖动模式，True 表明进入锁轴零力拖动，False 为退出锁轴零力拖动进入正常模式。只有在正常模式下，才可以通过程序控制机械臂运动。

indexLockedJoint：输入参数，int 型变量，锁定轴的索引值，-1 代表不锁轴，0 代表 1 轴。目前仅支持锁定 3 轴，即该值设为 2，其他值视为不锁轴。

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,0,0,0,0,0))
DianaApi.releaseBrake()
ch = input('enter free driving with locked joint 2')
DianaApi.freeDrivingWithLockedJoint(True, 2, ipAddress)
ch = input('leave free driving with locked joint 2')
DianaApi.freeDrivingWithLockedJoint(False, 2, ipAddress)
```

158 freeDrivingWithLockedJoint_ex

```
def freeDrivingWithLockedJoint_ex(enable, frame_type, force_direction, indexLockedJoint = -1, ipAddress = '')
```

实现控制指定 IP 地址的机械臂正常模式与锁轴定向零力拖动模式之间的切换。定向零力拖动是指可选择笛卡尔空间可拖动方向的轴空间零力拖动。

适配臂型：通用医疗臂、定制医疗臂。

参数：

enable：输入参数，bool 变量，是否进入锁轴定向零力拖动模式，true 表明进入锁轴定向零力拖动，false 为退出锁轴定向零力拖动进入正常模式。只有在正常模式下，才可以通过程序控制机械臂运动。

frame_type：输入参数，int 型变量，用于指定在哪个坐标系下选择可拖动方向。

- 0：不限制拖动方向，等同于关节空间虚拟墙；
- 1：在基坐标系下描述可拖动方向；
- 2：在工具坐标系下描述可拖动方向；
- 3：在工件坐标系下描述可拖动方向。

force_direction：输入参数，6 个表达力的方向（三个平移维度和三个旋转维度）的元组。非“0”代表对应维度可拖动，“0”代表对应维度不可拖动。

indexLockedJoint：输入参数，int 型变量，锁定轴的索引值，-1 代表不锁轴，0 代表 1 轴。目前仅支持锁定 3 轴，即该值设为 2，其他值视为不锁轴。

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,0,0,0,0,0)) DianaApi.releaseBrake() direction = [0, 0, 1, 0, 0, 0] ch = input('enter free driving ex with locked joint 2') DianaApi.freeDrivingWithLockedJoint_ex(True, 1, direction, 2, ipAddress) ch = input('leave free driving ex with locked joint 2') DianaApi.freeDrivingWithLockedJoint_ex(False, 1, direction, 2, ipAddress)

159 freeDrivingForCurrentLoop

def freeDrivingForCurrentLoop (enable:bool, ipAddress = '') -> bool
使指定 IP 地址机械臂开启或关闭电流环自由驱动。开启电流环自由驱动的前提是系统已进入安全处理模式。安全处理模式是非正常工作模式，通常用于处理机械臂的异常情况，如关节超极限位置、机械臂发生严重碰撞或发生 JCU 过流等硬件故障。开启电流环自由驱动后，抱闸自动打开，用户可以手动将机械臂关节拖回预期的工作区间。关闭电流环自由驱动后，抱闸自动关闭。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **注：该 API 需要结合 enterSafetyIdle 和 leaveSafetyIdle 进行使用。只有进入安全处理模式以后，才能成功开启电流环自由驱动。当关闭电流环自由驱动后，需要退出安全处理模式。典型的调用过程如下：** **1、调用 enterSafetyIdle 进入安全处理模式；** **2、调用 freeDrivingForCurrentLoop(True)开启电流环自由驱动，将机械臂关节拖回正常工作区间；** **3、调用 freeDrivingForCurrentLoop(False)关闭电流环自由驱动；** **4、调用 leaveSafetyIdle 退出安全处理模式。** **参数：** enable：输入参数，开启或关闭电流环自由驱动，bool 型变量。 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：** False：调用失败。 True：调用成功
调用示例：

```
import DianaApi

ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret = DianaApi.enterSafetyIdle(ipAddress)
if ret == True:
   ret = DianaApi.freeDrivingForCurrentLoop(True, ipAddress)
   if ret == True:
      time.sleep(15)
      ret = DianaApi.freeDrivingForCurrentLoop(False, ipAddress)
      if ret != True:
         print("failed to leave current loop free driving")
      ret = DianaApi.leaveSafetyIdle(ipAddress)
      if ret != True:
         print("failed to leave Safety Idle state")
   else:
      print("failed to enter current loop free driving")
else:
   print("failed to enter Safety Idle state")
```

160 getCurrentLoopFreeDrivingState

def getCurrentLoopFreeDrivingState(ipAddress = '')

查询指定 IP 地址机械臂电流环自由驱动的状态。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

-1：网络调用失败。

0：未处于电流环自由驱动

1：处在电流环自由驱动

调用示例：

```
import DianaApi

ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret=DianaApi.getCurrentLoopFreeDrivingState(ipAddress)
print(ret)
```

161 calcRobotArmAngle

def calcRobotArmAngle(joints,ipAddress = '')

对指定 IP 地址机械臂，计算传入关节角对应的臂角。

适配臂型：定制医疗臂。

参数：

joints：输入参数，机械臂各关节的关节角元组，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：

返回带两个参数的元组

参数 0：

True：成功。

False：失败。

参数 1：

输出参数。计算出的轴角，角度。

调用示例：

```
import DianaApi
PI=3.1415926
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.initSrv([ipAddress,0,0,0,0,0])
#以 7 轴机械臂为例
joints = [0,0,90 * PI /180,90 * PI/180,0,0,0]
ret = DianaApi.calcRobotArmAngle(joints, ipAddress)
if ret[0] == True:
  print(ret[1])
else:
  print("cannot get robot arm angle")
DianaApi.destroySrv()
```

162 enterSafetyIdle

def enterSafetyIdle(ipAddress = '') -> bool

使能指定 IP 地址机械臂进入安全处理模式。安全处理模式是非正常工作模式，通常用于处理机械臂的异常情况，如关节超极限位置、机械臂发生严重碰撞或发生 JCU 过流等硬件故障。进入安全处理模式以后，抱闸会自动关闭，此时可以通过开启电流环自由驱动来将机械臂的关节拖回预期的工作区间。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

参考 freeDrivingForCurrentLoop 的调用示例。

163 leaveSafetyIdle

def leaveSafetyIdle(ipAddress = '') -> bool

使能指定 IP 地址机械臂退出安全处理模式。安全处理模式是非正常工作模式，通常用于处理机械臂的异常情况，如关节超极限位置、机械臂发生严重碰撞或发生 JCU 过流等硬件故障。退出安全处理模式以后，抱闸仍处于关闭状态。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

参考 freeDrivingForCurrentLoop 的调用示例。

164 identifyTcpPayload

def identifyTcpPayload(result, fnIsStop=None, is_mass_valid=False, mass=0.0, waypoints = list(list()), vels=list(), ipAddress='')

对指定 IP 地址的机械臂执行一系列移动指令进行负载辨识。该功能会执行一段时间，移动至每个路点后大概会停留 3 秒左右。请不要使用多线程同时调用其他移动指令或修改系统参数配置。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

result：输出参数，用于接收负载信息，大小为 10 的数组，第 1 位为质量（单位：kg），2~4 位为质心（单位：m），5~10 位为张量（单位：$kg \cdot m^2$）。该结果可以作为输入参数传给 setActiveTcpPayload 使用。目前该函数仅支持辨识质量和质心，即当返回结果有效时

result 数组只有前四个有值。

fnIsStop：可选参数，停止负载辨识功能的回调函数。函数声明形式： def fnIsStop (ipAddress) 。当 fnIsStop 函数返回 True 时，将中止正在执行的负载辨识程序。该参数缺省值为空，如果不传入该参数，用户开启负载辨识程序后将无法干预程序执行。fnIsStop 函数将在负载辨识程序开始执行后，每 200 毫秒检测一次，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。

is_mass_valid：可选参数，bool 型输入参数，标记 mass 参数是否有效。当该值为 False 时，参数 mass 的值将被忽略。缺省值为 False。

mass：可选参数，double 型输入参数，仅当 is_mass_valid 为 True 时，该值有效。缺省为 0.0。

waypoints：可选参数，嵌套链表，用于存放 JOINT_NUM *n(暂时仅支持 n=32)的路点信息，单位：rad。缺省值为空。当该值为空时，使用系统默认的 32 个路点作为负载辨识路点。在负载辨识过程中将以传入路点的第一组值作为初始点，机械臂先移动至初始点，然后从第二个路点开始依次移动至最后一个路点，做为第一轮正向辨识。第一轮结束后，机械臂回到初始点，再进行第二轮反向识别，即回到最后一个点，逆向依次移动至第二个路点后结束整个辨识过程。因此如果路点数过少可能会导致辨识失败、结果不准确等问题。由于整个辨识过程相对独立，部分失败原因会以 identifyTcpPayload 函数返回值的方式通知用户。请配合 getLastError 函数一起判断辨识结果的有效性。

vels：可选参数。与 waypoints 每组路点一一对应的速度百分比（如 30 表示 30%，建议速度百分比不超过 30%）。vels 中元素个数与 waypoints 中关节角组数应保持一致。不一致将返回错误且不会运行负载辨识程序。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

且会打印以下错误码对应的错误信息，

0：成功。

-1：其他错误。

-2：无法读取指定 IP 地址的机械臂的关节扭矩。

-3：初始化失败。

-4：采样失败。

-5：传送辨识数据失败。

-6：结果不可信。

-7：负载辨识失败。

-8：回调函数中断负载辨识程序执行。当用户传入的 fnNeedStop 函数返回 true 时触发此错误。可以不作处理。

-9：指定 IP 地址的机械臂已经在执行负载辨识任务。

-10：开始执行负载辨识任务失败。

-11：结束负载辨识任务失败。

-12：指定 IP 地址的机械臂正在运行示教程序，无法执行负载辨识任务。

-13：传入了非法参数。

-17：未打开抱闸。

调用示例：

```
import DianaApi

ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))

glb_cnt = 0
def stopLoadIdentification(ipAddress):
    global glb_cnt
    glb_cnt = glb_cnt + 1
    print(ipAddress)
    if glb_cnt == 10:
        glb_cnt = 0
        print("stopLoadIdentification:return ")
        return True
    print("stopLoadIdentification:"+str(glb_cnt))
    return False

netInfo=('192.168.10.75', 0, 0, 0, 0)
DianaApi.initSrv(netInfo)
DianaApi.releaseBrake()
time.sleep(2)
#以 7 轴机械臂为例
waypoints=((0.0, 0.0, 0.0, 1.57, -1.047, -1.57, -1.57)
           ,(0.0, 0.0, 0.0, 1.57, -0.979, -1.503, -1.469)
           ,(0.0, 0.0, 0.0, 1.57, -0.912, -1.435, -1.368)
           ,(0.0, 0.0, 0.0, 1.57, -0.844, -1.368, -1.266)
           ,(0.0, 0.0, 0.0, 1.57, -0.776, -1.3, -1.165)
           ,(0.0, 0.0, 0.0, 1.57, -0.709, -1.232, -1.064)
           ,(0.0, 0.0, 0.0, 1.57, -0.641, -1.165, -0.962)
           ,(0.0, 0.0, 0.0, 1.57, -0.574, -1.097, -0.861)
           ,(0.0, 0.0, 0.0, 1.57, -0.506, -1.030, -0.76)
           ,(0.0, 0.0, 0.0, 1.57, -0.439, -0.962, -0.658)
           ,(0.0, 0.0, 0.0, 1.57, -0.371, -0.895, -0.557)
```

```
            ,(0.0, 0.0, 0.0, 1.57, -0.304, -0.827, -0.456)
            ,(0.0, 0.0, 0.0, 1.57, -0.236, -0.76, -0.354)
            ,(0.0, 0.0, 0.0, 1.57, -0.168, -0.692, -0.253)
            ,(0.0, 0.0, 0.0, 1.57, -0.101, -0.624, -0.152)
            ,(0.0, 0.0, 0.0, 1.57, -0.033, -0.557, -0.05)
            ,(0.0, 0.0, 0.0, 1.57, 0.033, -0.489, 0.05)
            ,(0.0, 0.0, 0.0, 1.57, 0.101, -0.422, 0.152)
            ,(0.0, 0.0, 0.0, 1.57, 0.168, -0.354, 0.253)
            ,(0.0, 0.0, 0.0, 1.57, 0.236, -0.287, 0.354)
            ,(0.0, 0.0, 0.0, 1.57, 0.304, -0.219, 0.456)
            ,(0.0, 0.0, 0.0, 1.57, 0.371, -0.152, 0.557)
            ,(0.0, 0.0, 0.0, 1.57, 0.439, -0.084, 0.658)
            ,(0.0, 0.0, 0.0, 1.57, 0.506, -0.016, 0.76)
            ,(0.0, 0.0, 0.0, 1.57, 0.574, 0.050, 0.861) #
            ,(0.0, 0.0, 0.0, 1.57, 0.641, 0.118, 0.962)
            ,(0.0, 0.0, 0.0, 1.57, 0.709, 0.185, 1.064)
            ,(0.0, 0.0, 0.0, 1.57, 0.776, 0.253, 1.165)
            ,(0.0, 0.0, 0.0, 1.57, 0.844, 0.32, 1.266) #
            ,(0.0, 0.0, 0.0, 1.57, 0.912, 0.388, 1.368)
            ,(0.0, 0.0, 0.0, 1.57, 0.979, 0.456, 1.469)
            ,(0.0, 0.0, 0.0, 1.57, 1.047, 0.523, 1.57))
vels=(30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,30,3
0,30,30,30)
result = [0.0]*10
funStopLoadIdentification = DianaApi.FNCSTOP(stopLoadIdentification)
ret = DianaApi.identifyTcpPayload(result, None, True, 1.5, waypoints, vels, "192.168.10.75")
print("ret=",ret)
print("result=",result)
```

165 setJointImpeda

```
def setJointImpeda (arrStiff, arrDampRatio, ipAddress = "")
```

设置指定 IP 地址机械臂各关节阻抗参数，包含刚度 Stiffiness 和阻尼比 DampRatio 的数据。

适配臂型：通用医疗臂、定制医疗臂。

参数：

arrStiff：表示各关节刚度 Stiffiness 的元组，长度为 JOINT_NUM。

arrDampRatio：表示各关节共用的阻尼比 DampRatio 的元组，长度为 1，设置范围在[0.5,1.2]。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7。

返回值：

True：成功。

False：失败。

调用示例：
import DianaApi
#以 7 轴机械臂为例
arrStiff = (3000, 3000, 3000, 1000, 500, 1000, 1000)
arrDampRatio= (0.7,)
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.setJointImpeda(arrStiff, arrDampRatio, ipAddress)

166 getJointImpeda

def getJointImpeda (arrStiff, arrDampRatio, ipAddress = "")
获取指定 IP 地址机械臂各关节阻抗参数，包含刚度 Stiffiness 和统一阻尼比 DampRatio 的数据。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** arrStiff：表示各关节刚度 Stiffiness 的列表，长度为 JOINT_NUM，用于接收获取到的值。 arrDampRatio：表示各关节共用的阻尼比 DampRatio 的列表，长度为 1，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。 注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7。 **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] * JOINT_NUM
arrDampRatio= [0]
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.getJointImpeda (arrStiff, arrDampRatio, ipAddress)

167 setCartImpeda

def setCartImpeda (arrStiff, arrDampRatio, ipAddress = "")
设置指定 IP 地址机械臂笛卡尔空间阻抗参数。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** arrStiff：表示笛卡尔空间，各维度刚度 Stiffiness 的元组，长度为 6。 arrDampRatio：表示笛卡尔空间，各维度共用的阻尼比 DampRatio 的元组，长度为 1，设置范围在[0.5,1.2]。 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 arrStiff = (1000, 1000, 1000, 500, 500, 500) arrDampRatio = (0.7,) ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.setCartImpeda(arrStiff, arrDampRatio, ipAddress)

168 getCartImpeda

def getCartImpeda (arrStiff, arrDampRatio, ipAddress = "")
获取指定 IP 地址机械臂笛卡尔空间各维度阻抗参数，包含刚度 Stiffiness 和阻尼比 DampRatio 的数据。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** arrStiff：表示笛卡尔空间，各维度刚度 Stiffiness 的列表，长度为 6，用于接收获取到的值。 arrDampRatio：表示笛卡尔空间，各维度共用的阻尼比 DampRatio 的列表，长度为 1，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 arrStiff = [0] * 6 arrDampRatio = [0] ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.getCartImpeda (arrStiff, arrDampRatio, ipAddress)

169 getRobotNetworkInfo

def getworkNetworkInfo(robotNetworkInfo:list,ipAddress = "")
获取指定 IP 地址机械臂上的网络配置信息，包含网卡名、IP 地址和子网掩码的数据。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** robotNetworkInfo：表示远端机器网络信息的列表，不同网卡的信息以分号(;)进行分割，网卡名与网络信息以冒号(:)进行分割，IP 地址与子网掩码以正斜杠(/)作为分割。例如:"enp2s0:192.168.10.75/24;enp3s0:192.168.1.75/24" 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
robotNetworkInfo = []
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
if DianaApi.getRobotNetworkInfo(robotNetworkInfo,ipAddress) == False:
  print("Diana API getworkNetworkInfo failed!\n")
else:
  print(robotNetworkInfo)

170 setRobotNetworkInfo

def setworkNetworkInfo(currentNetworkInfo,targetNetworkInfo,ipAddress = "")
设置指定 IP 地址机械臂上的网络配置信息，目前支持设置 IP 地址和子网掩码。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

currentNetworkInfo：表示远端机器期望修改的网络信息的字符串，现支持修改 IP 地址和子网掩码。如果需要修改多个 IP 地址，需要用分号(;)进行分割，比如"192.168.10.75/24;192.168.1.75/24"。如果不需要修改子网掩码，则可以不写，比如上述例子也可以直接写成"192.168.10.75;192.168.1.75"，但是如果携带子网掩码，则必须确保子网掩码为当前 IP 所使用。

targetNetworkInfo：表示远端机器中期望变成的网络信息，现支持修改 IP 地址和子网掩码，如果需要修改多个 IP 地址，需要用分号(;)进行分割，比如"192.168.1.75/24;192.168.10.75/24"。如果不需要修改子网掩码，可以不写。注意，需要确保 targetNetworkInfo 中的 IP 信息的数量与 currentNetworkInfo 中一致。

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（单 IP 修改）：

```
import DianaApi
currentNetworkInfo = "192.168.10.75"
targetNetworkInfo = "192.168.10.88"
ipAddress =  "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
if  DianaApi.setRobotNetworkInfo(currentNetworkInfo,targetNetworkInfo,ipAddress) ==
False:
   print("Diana API setworkNetworkInfo failed!\n")
#调用示例 2（多 IP 修改）：
currentNetworkInfo = "192.168.10.75/24;192.168.1.75/24"
targetNetworkInfo = "192.168.1.75;192.168.10.75"
ipAddress =  "192.168.10.75"
if DianaApi.setRobotNetworkInfo(currentNetworkInfo,targetNetworkInfo,ipAddress) == False:
   print("Diana API setRobotkNetworkInfo failed!\n")
else:
   print("Diana API setRobotNetworkInfo succeed!\n")
```

171 getRgbLedState

```
def getRgbLedState(rgbLedState : list, ipAddress = "")
```

获取指定 IP 地址机械臂各轴灯环的颜色。

适配臂型：定制医疗臂。

参数：

rgbLedState：表示各轴灯环颜色的列表，长度为 JOINT_NUM，其中每个关节使用一个长度为 3 的列表(即 RGB 值)表示颜色。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
#JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7。
color_in = [[10,5,10]] * JOINT_NUM
color_out = []
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
if DianaApi.setRgbLedState(color_in, ipAddress) == False:
   print("Diana API setRgbLedState failed!\n")
else:
   if DianaApi.getRgbLedState(color_out, ipAddress) == True:
      print(color_out)
   else:
      print("Diana API getRgbLedState failed!\n")
```

172 setRgbLedState

def setRgbLedState(rgbLedState : list, ipAddress = "")

设置指定 IP 地址机械臂各轴灯环的颜色。

适配臂型：定制医疗臂。

参数：

rgbLedState：表示各轴灯环颜色的列表，长度为 JOINT_NUM，其中每个关节使用一个长度为 3 的列表(即 RGB 值)表示颜色。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7。

返回值：

True：成功。

False：失败。

调用示例：

见 getRgbLedState

173 getFunctionOptI4

def getFunctionOptI4(function_index:function_index_e, opt_name:function_opt_name_e, ipAddress = '')

获取指定功能的可选整型参数。
适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

intFunctionIndex：指定功能索引，取值为 function_index_e 枚举类型，枚举值及含义如下表所示：

intFunctionIndex	对应枚举	含义
0	function_index_e:: E_FREEDRIVING	自由驱动相关
1	function_index_e::E_CART_IMPEDANCE	力控和笛卡尔阻抗相关

intOptName：功能的可选参数，

当 intFunctionIndex 为 0 时，表示自由驱动相关参数，对应含义如下：

intOptName	对应枚举值	含义
0x10005	function_opt_name_e::T_ENABLE_REALTIME_VIRTUAL_WALL	获取实时虚拟墙功能是否开启的状态
0x10008	function_opt_name_e::T_CONTROL_RULE	获取自由驱动的控制方式

当 intFunctionIndex 为 1 时，表示力控和笛卡尔阻抗相关参数，对应含义如下：

intOptName	对应枚举	含义
0x10100	function_opt_name_e:: T_COORDINATE_TYPE	获取力控和笛卡尔阻抗的坐标系类型。
0x10101	function_opt_name_e::T_IS_SINGLE_AXIS_LOCKED	获取力控和笛卡尔阻抗是否锁轴
0x10103	function_opt_name_e::T_CONSTRAIN_TYPE	获取力控和笛卡尔阻抗的零空间约束条件

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：

返回值包含两个元素的元组，第一个元素表示获取是否成功，返回值及含义如下，

True：获取成功，

False：获取失败。

对应不同的 function_opt_name，第二个元素的值及含义有所不同，具体如下表：

intOptName	对应 OptName 枚举值	返回元组第二项含义
0x10005	function_opt_name_e::T_ENABLE_REALTIME_VIRTUAL_WALL	0 表示失败；

		1 表示成功。
0x10008	function_opt_name_e::T_CONTROL_RULE	0 表示基于阻抗的自由驱动； 1 表示基于导纳的自由驱动。
0x10100	function_opt_name_e:: T_COORDINATE_TYPE	0 表示基坐标系 1 表示工具坐标系 2 表示工件坐标系
0x10101	function_opt_name_e::T_IS_SINGLE_AXIS_LOCKED	1 表示锁 3 轴
0x10103	function_opt_name_e::T_CONSTRAIN_TYPE	0 表示零空间无约束； 1 表示以 3 轴趋向正负 10 度作为零空间的约束条件。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
ret=DianaApi.getFunctionOptI4(DianaApi.function_index_e.E_CART_IMPEDANCE,
DianaApi.function_opt_name_e.E_CONSTRAIN_TYPE)
if ret[0] == True:
   print('Constrain type for cart impedance:' + str(ret[1]))
```

174 setFunctionOptI4

```
def setFunctionOptI4(function_index:function_index_e, opt_name:function_opt_name_e,
opt_value, ipAddress = '')
```

设置指定功能的可选整型参数。
适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

function_index：指定功能，取值为 function_index_e 枚举类型，枚举值及含义如下，

function_index	对应枚举	含义
0	function_index_e::T_FREEDRIVING	自由驱动相关
1	function_index_e::T_CART_IMPEDANCE	力控和笛卡尔阻抗相关

opt_name:指定功能的属性，取值为 function_opt_name_e 的枚举类型，枚举值及含义如下：

当 function_index 为 0 时，代表自由驱动相关参数，对应含义如下：

opt_name	对应枚举	含义
0x10005	function_opt_name_e::T_ENABLE_REALTIME_VIRTUAL_WALL	使能实时虚拟墙功能
0x10008	function_opt_name_e::T_CONTROL_RULE	设置自由驱动的控制方式

opt_value：用于存放参数的值，随 opt_name 的值变化：

opt_name	对应 opt_name 枚举值	opt_value 值及其含义
0x10005	function_opt_name_e::T_ENABLE_REALTIME_VIRTUAL_WALL	0 表示 disable； 1 表示 enable。
0x10008	function_opt_name_e::T_CONTROL_RULE	0 表示基于阻抗的自由

		驱动； 1 表示基于导纳的自由驱动。

当 function_index 为 1 时，代表力控和笛卡尔阻抗相关参数，对应含义如下：

intOptName	对应枚举	含义
0x10100	function_opt_name_e:: T_COORDINATE_TYPE	设置力控和笛卡尔阻抗的坐标系类型。
0x10102	function_opt_name_e::T_LOCKED_SINGLE_AXIS	设置力控和笛卡尔阻抗是否锁轴
0x10103	function_opt_name_e::T_CONSTRAIN_TYPE	设置力控和笛卡尔阻抗的零空间约束条件

opt_value：用于存放参数的值，随 opt_name 的值变化：

opt_name	对应 opt_name 枚举值	opt_value 值及其含义
0x10100	function_opt_name_e:: T_COORDINATE_TYPE	0 表示基坐标系 1 表示工具坐标系 2 表示工件坐标系
0x10102	function_opt_name_e::T_LOCKED_SINGLE_AXIS	1 表示锁 3 轴；
0x10103	function_opt_name_e::T_CONSTRAIN_TYPE	0 表示零空间无约束； 1 表示以 3 轴趋向正负 10 度作为零空间的约束条件。

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,0,0,0,0,0))
ret=DianaApi.setFunctionOptI4(DianaApi.function_index_e.E_CART_IMPEDANCE,
DianaApi.function_opt_name_e.E_CONSTRAIN_TYPE, 1)
```

175 setGravInfo

```
def setGravInfo(gravity, ipAddress= " ")
```

针对指定 IP 地址的机械臂，设置其重力矢量。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

gravity：输入参数，重力矢量的列表，长度为 3。

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,0,0,0,0,0)) grav = (0,0,9.8) ret = DianaApi.setGravInfo(grav, ipAddress) print("setGravInfo ret=",ret) print("setGravInfo result=",grav)

176 getGravInfo

def getGravInfo(gravity, ipAddress = " ")
针对指定 IP 地址的机械臂，获取其安装信息的重力矢量。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** gravity：输出参数，重力矢量的列表，长度为 3。 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,0,0,0,0,0)) grav = [0.0]*3 ret = DianaApi.getGravInfo(grav, ipAddress) print("getGravInfo ret=",ret) print("getGravInfo result=",grav)

177 getSixAxisInstallation

def getSixAxisInstallation(center_of_mass, pose_of_six_axis_in_falan, gravity, ipAddress = " ")
针对指定 IP 地址的机械臂，获取系统中存储的六维力传感器和转接盘安装信息。安装信息包括：质量、质心、六维力坐标系相对法兰的位姿和表达机械臂安装姿态的重力矢

量。常用于配合 initSixAxisInstallation 函数一起使用。

适配臂型：通用医疗臂、Thor3。

参数：

center_of_mass：输出参数，系统中存储的六维力传感器和转接盘的质心向量的列表，长度为 3。单位：m。

pose_of_six_axis_in_falan：输出参数，系统中存储的六维力坐标系相对法兰的位姿向量的列表，长度为 16。

gravity：输出参数，重力加速度的矢量在机械臂基座标系中的表达，长度为 3。

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：

（1）函数调用是否成功：

True：成功。

False：失败。

（2）系统中存储的六维力传感器和转接盘的质量。单位：kg。

调用示例：

详见 initSixAxisInstallation 的调用示例。

178 initSixAxisInstallation

```
def initSixAxisInstallation(mass, center_of_mass, pose_of_six_axis_in_falan, gravity, ipAddress = "")
```

针对指定 IP 地址的机械臂，初始化六维力传感器和转接盘的安装信息。安装信息包括：质量、质心、六维力坐标系相对法兰的位姿和表达机械臂安装姿态的重力矢量。

在使用跟六维力传感器相关功能时应首先调用此函数。为了方便用户使用可以通过 getSixAxisInstallation 获取系统中存储的安装信息，确认与 robot 上安装的传感器信息匹配后作为初始化参数传入。该参数传入系统后除了设置给六维力相关功能，还会保存用于下次 getSixAxisInstallation 调用。

适配臂型：通用医疗臂、Thor3。

参数：

mass：输入参数，六维力传感器和转接盘的质量。单位：kg。

center_of_mass：输入参数，六维力传感器和转接盘的质心向量的列表，长度为 3。单位：m。

pose_of_six_axis_in_falan：输入参数，六维力坐标系相对法兰的位姿向量的列表，长度

为 16。

gravity：输入参数，重力加速度的矢量在机械臂基座标系中的列表，长度为 3。

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,0,0,0,0,0))
glb_cnt = 0
def stopLoadIdentification(ipAddress):
  global glb_cnt
  glb_cnt = glb_cnt + 1
  print(ipAddress)
  if glb_cnt == 500:
    glb_cnt = 0
    print("stopLoadIdentification:return ")
    return True
  print("stopLoadIdentification:"+str(glb_cnt))
  return False

center = [0.0]*3
falan = [0.0]*16
grav = [0.0]*3
ret, mass = DianaApi.getSixAxisInstallation(center, falan, grav)
print("getSixAxisInstallation ret=",ret)
print("getSixAxisInstallation mass=",mass)
print("getSixAxisInstallation center=",center)
print("getSixAxisInstallation falan=",falan)
print("getSixAxisInstallation grav=",grav)

dblmass = 0.185
ret = DianaApi.initSixAxisInstallation(dblmass, center, falan, grav)
print("initSixAxisInstallation ret=",ret)

home = [0.0]* JOINT_NUM
wp = [0.0]*36
vel = [0.0]*6
acc = [0.0]*6
ret = DianaApi.getCalcSixAxisWaypoints(home, wp, vel, acc)
print("getCalcSixAxisWaypoints ret=",ret)
print("getCalcSixAxisWaypoints home =", home)
print("getCalcSixAxisWaypoints wp=",wp)
```

```
print("getCalcSixAxisWaypoints vel=",vel)
print("getCalcSixAxisWaypoints acc=",acc)

payload = [0.0]*10
funStopIdentification = DianaApi.FNCSTOP(stopLoadIdentification)
ret = DianaApi.identifyTcpPayloadWithSixAxis(payload, home, wp, vel, acc, funStopIdentification)
print("identifyTcpPayloadWithSixAxis ret=",ret)
print("identifyTcpPayloadWithSixAxis payload=",payload)
fixed = [0.0]*6
for i in range(0, 100):
  ret = DianaApi.getFixedSixAxisForce(fixed)
  print("getFixedSixAxisForce ret=",ret)
  print("getFixedSixAxisForce fixed=",fixed)
```

179 getCalcSixAxisWaypoints

def getCalcSixAxisWaypoints((homepoint, waypoints, vels, accs, ipAddress = " ")

针对指定 IP 地址的机械臂，获取六维力传感器标定过程推荐的点位。每次调用 initSixAxisInstallation 后，该点位信息会根据传入的参数重新计算，并通过本函数获取。如果没有调用 initSixAxisInstallation，直接调用本函数，将得到上次成功标定六维力时使用的信息。此处获取到的点位可能受限于 robot 实际安装位姿，需谨慎使用。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

homepoint：输出参数，六维力传感器标定过程推荐的 Home 点位向量的列表，长度为 JOINT_NUM 的关节角信息。单位：rad。

waypoints：输出参数，六维力传感器标定过程推荐的点位向量的列表，长度为 6*6 的 pose 信息，每组点位后三个量为轴角。

vels：输出参数，六维力传感器标定过程推荐的点位对应的速度向量的列表，长度为 6。单位：m/s。

accs：输出参数，六维力传感器标定过程推荐的点位对应的加速度向量的列表，长度为 6。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：

元组，共计三个元素：

（1）函数调用是否成功：

True：成功。

False：失败。

（2）系统推荐的 Home 点移动速度。

（3）系统推荐的 Home 点移动加速度。

调用示例：

详见 initSixAxisInstallation 的调用示例。

180 identifyTcpPayloadWithSixAxis

def identifyTcpPayloadWithSixAxis (payload, homepoint, vel_home, acc_home, waypoints, vels, accs, fnIsStop=None, ipAddress = " ")

针对指定 IP 地址的机械臂，通过执行一系列移动指令来标定六维力传感器，并返回包含六维力传感器和转接盘的负载信息。六维力传感器零偏参数和计算所得的质量质心结果会直接存入后台。该功能会执行一段时间，移动至每个路点后大概会停留 10 秒左右。请不要使用多线程同时调用其他移动指令或修改系统参数配置。

适配臂型：通用医疗臂、Thor3。

参数：

payload：输出参数，用于接收负载信息，大小为 10 的列表，第 1 位为质量（单位：kg），2~4 位为质心（单位：m），5~10 位为张量（单位：$kg•m^2$）。该结果可以作为输入参数传给 setActiveTcpPayload 使用。目前该函数仅支持辨识质量和质心，即当返回结果有效时 dblResult 数组只有前四个有值。

homepoint：输入参数，六维力传感器标定过程的 Home 点位向量的列表，长度为 JOINT_NUM 的关节角信息。单位：rad。

vel_home：输入参数，移动至 Home 点的速度。

acc_home：输入参数，移动至 Home 点的加速度。

waypoints：输入参数，六维力传感器标定过程的点位向量的列表，长度为 6*6 的 pose 信息，每组点位后三个量为轴角。该路点信息可以通过 getCalcSixAxisWaypoints 获取。

vels：输入参数，六维力传感器标定过程推荐的点位对应的速度向量的列表，长度为 6。单位：m/s。

accs：输入参数，六维力传感器标定过程推荐的点位对应的加速度向量的列表，长度为 6。单位：m/s^2。

fnIsStop：可选参数，停止标定功能的回调函数。函数声明形式：bool fnNeedStop (const char *strIpAddress)。当 fnNeedStop 函数返回 true 时，将中止正在执行的负载辨识程序。该参数缺省值为空，如果不传入该参数，用户开启负载辨识程序后将无法干预程序执行。fnNeedStop 函数将在负载辨识程序开始执行后，每 200 毫秒检测一次，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。

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。

且会打印以下错误码对应的错误信息，

0：成功。

2：正在运行六维力标定。

-1：其他错误

-2：六维力标定相应 callback 函数停止。

-3：六维力标定路点数量不匹配。

-4：六维力标定设置路点或 Home 点失败。

-5：开始六维力标定失败。

-6：停止六维力标定失败。

-7：获取六维力标定结果失败。

-8：六维力标定路段规划失败。

-9：六维力标定 move 失败。

-11：非位置模式无法进行六维力标定

-12：系统正在执行其他程序无法进行六维力标定。

-13：负载辨识中无法进行六维力标定。

-14：力控模式无法进行六维力标定。

-15：系统处于非空闲态，无法进行六维力标定。

-16：路点存储失败。

-17：未打开抱闸。

-18：参数非法。

调用示例：

详见 initSixAxisInstallation 的调用示例。

181 getFixedSixAxisForce

def getFixedSixAxisForce(fixed_force, ipAddress = " ")

获取指定 IP 地址的机械臂（重新标定六维力传感器后）修正后的六维力值。

适配臂型：通用医疗臂、Thor3。

参数：

fixed_force：输出参数，由修正后的六维力传感器感应到的力向量组成的列表，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：

详见 initSixAxisInstallation 的调用示例。

182 setSixAxisSensorAbsAccuracy

def setSixAxisSensorAbsAccuracy(accuracy, ipAddress = " ")

设置指定 IP 地址的机械臂六维力传感器的绝对测量精度。其中平移力分量的有效值范围为[0.001, 2.0]，扭矩分量的有效值范围为[0.0001, 0.1]。其中，平移分量的默认值都为 1.0，扭矩分量的默认值为 0.02。

适配臂型：通用医疗臂、Thor3。

参数：

accuracy：输入参数，绝对测量精度的列表，长度为 6，依次对应平移力 x 分量、平移力 y 分量、平移力 z 分量、扭矩 x 分量、扭矩 y 分量和扭矩 z 分量。

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
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
accuracy=(1.0,1.0,1.0,0.02, 0.02, 0.02)
ret = DianaApi.setSixAxisSensorAbsAccuracy(accuracy, ipAddress)
print(accuracy)
```

183 getSixAxisSensorAbsAccuracy

def getSixAxisSensorAbsAccuracy (accuracy, ipAddress = " ")

获取指定 IP 地址的机械臂六维力传感器的绝对测量精度。

适配臂型：通用医疗臂、Thor3。

参数：

accuracy：输出参数，绝对测量精度的列表，长度为 6，依次对应平移力 x 分量、平移力 y 分量、平移力 z 分量、扭矩 x 分量、扭矩 y 分量和扭矩 z 分量。

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
ipAddress = "192.168.10.75"
DianaApi.initSrv((ipAddress,0,0,0,0,0))
accuracy=[0]*6
ret = DianaApi.getSixAxisSensorAbsAccuracy(accuracy, ipAddress)
print(accuracy)
```

184 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。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

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：

int 型数值，表示关节数量。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
joint_cnt = DianaApi.getJointCount(ipAddress)
print("getJointCount return : ", joint_cnt)
```

185 setExternalAppendTorCutoffFreq

```
def setExternalAppendTorCutoffFreq(freq, ipAddress = '')
```

在指定 IP 地址机械臂上，设置各关节附加力矩的滤波截止频率。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

freq：设置各关节附加力矩的滤波截止频率。

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,0,0,0,0,0))
dblFreq = 1.0
ret = DianaApi.setExternalAppendTorCutoffFreq(dblFreq, ipAddress)
print("setExternalAppendTorCutoffFreq return : ", ret)
```

186 getInertiaMatrix

```
def getInertiaMatrix(motorPosition, inerMatrix, ipAddress = '')
```

在指定 IP 地址机械臂上，获取惯性参数矩阵。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

motorPosition：输入参数，机械臂的关节角，大小为 JOINT_NUM 的列表。

inerMatrix：输出参数，惯性矩阵,大小为 JOINT_NUM * JOINT_NUM 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
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
joints = [0]*7
matrix =[0] * 16
ret = DianaApi.getInertiaMatrix(joints, matrix, ipAddress)
print("getInertiaMatrix matrix:",matrix," return ", ret)
```

187 getSixAxiaForce

def getSixAxiaForce(force, ipAddress = '')
在指定 IP 地址机械臂上，获取六维力传感器的读数。 适配臂型：通用医疗臂、Thor3。 **参数：** force：输出参数，六维力传感器的读数，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：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0,0,0)) force=[0]*6 ret = DianaApi.getSixAxiaForce(force, ipAddress) print("getSixAxiaForce force:",force," return ", ret)

188 getJointsSoftLimitRange

def getJointsSoftLimitRange(minPos, maxPos, ipAddress = '')
获取指定 IP 地址机械臂各关节的软限位。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** minPos：输出参数，JOINT_NUM 个轴关节的软限位下限组成的元组。单位：rad。 maxPos：输出参数，JOINT_NUM 个轴关节的软限位上限组成的元组。单位：rad。 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。 注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。 **返回值：** True：成功。 False：失败。
调用示例： import DianaApi ipAddress = '192.168.10.75'

```
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
minPos = [0.0]* JOINT_NUM
maxPos = [0.0]* JOINT_NUM
DianaApi. getJointsSoftLimitRange (minPos, maxPos, ipAddress)
print("getJointsSoftLimitRange::minPos="+str(minPos))
print("getJointsSoftLimitRange::maxPos="+str(maxPos))
```

189 setJointsSoftLimitRange

```
def setJointsSoftLimitRange(minPos, maxPos, ipAddress = '')
```

设置指定 IP 地址机械臂各关节软限位。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

minPos：输出参数，JOINT_NUM 个轴关节的软限位下限组成的元组。单位：rad。

maxPos：输出参数，JOINT_NUM 个轴关节的软限位上限组成的元组。单位：rad。

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。

注：JOINT_NUM 表示机械臂的关节个数，Diana 机械臂的 JOINT_NUM 为 7，Thor3 为 6。

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
#如果臂型是 Diana，JOINT_NUM=7;如果臂型是 Thor3，JOINT_NUM=6
minPos = [0.0]* JOINT_NUM
maxPos = [0.0]* JOINT_NUM
DianaApi.getJointsSoftLimitRange (minPos, maxPos, ipAddress)
for i in range(0, JOINT_NUM):
   minPos[i] = minPos[i] + 0.02
   maxPos[i] = maxPos[i] - 0.02
DianaApi.setJointsSoftLimitRange(minPos, maxPos, ipAddress)
DianaApi.saveEnvironment(ipAddress)
```

190 setJointLockedInCartImpedanceMode

```
def setJointLockedInCartImpedanceMode(lock, locked_joint_index, ipAddress = '')
```

设置指定 IP 地址机械臂在笛卡尔阻抗和/或力控模式下锁定/解锁某轴（当前版本仅支持 3

轴）。该设置将在机械臂进入笛卡尔阻抗或者力控模式时正式生效。

适配臂型：通用医疗臂、定制医疗臂。

参数：

lock：输入参数，如果该值为 true，表示机械臂在笛卡尔阻抗和/或力控模式下将锁定某轴（当前版本仅支持锁定 3 轴，即 intLockedJointIdex 必须为 2，否则锁定无效）；如果该值为 false，则不论 intLockedJointIdex 取值多少，均表示机械臂在笛卡尔阻抗和/或力控模式下解锁某轴（当前版本仅支持解锁 3 轴）。

locked_joint_index：输入可选参数，表示轴的索引值（索引从 0 开始），缺省值为 2（即 3 轴）。

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：

```
#设置锁定 3 轴
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
isLocked = True
index = 2
ret = DianaApi.setJointLockedInCartImpedanceMode(isLocked, index, ipAddress)
print("setJointLockedInCartImpedanceMode return ", ret)
#设置解锁 3 轴
ipAddress = '192.168.10.75'
isLocked = False
index = -1
ret = DianaApi.setJointLockedInCartImpedanceMode(isLocked, index, ipAddress)
print("setJointLockedInCartImpedanceMode return ", ret)
```

191 getJointLockedInCartImpedanceMode

def getJointLockedInCartImpedanceMode(ipAddress = '')

查询指定 IP 地址机械臂在笛卡尔阻抗和/或力控模式下某轴（当前版本仅支持 3 轴）是否属于锁定/解锁状态。

适配臂型：通用医疗臂、定制医疗臂。

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：

元组，共计两个元素：

（1）bool 型变量，是否为锁轴的状态。如果该值为 true，表示机械臂在笛卡尔阻抗和/或力控模式下某轴（当前版本仅支持 3 轴）处于锁定状态；如果该值为 false，则表示机械臂在笛卡尔阻抗和/或力控模式下某轴（当前版本仅支持 3 轴）处于解锁状态。

（2）函数调用是否成功：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
isLocked, ret = DianaApi.getJointLockedInCartImpedanceMode(ipAddress)
print(ret, "getJointLockedInCartImpedanceMode mode", isLocked)
```

192 getDefaultActiveTcp

```
def getDefaultActiveTcp(default_tcp, ipAddress = '')
```

获取指定 IP 地址机械臂默认的工具坐标系。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

default_tcp：输出参数，Tcp 相对于末端法兰盘的 4*4 齐次变换矩阵的首地址，列表长度为 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
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
default_tcp = [0.0] * 16
ret = DianaApi.getDefaultActiveTcp(default_tcp, ipAddress)
print("getDefaultActiveTcp failed! Return value = ", ret)
```

193 getTcpPayloadWithSixAxis

def getTcpPayloadWithSixAxis(mass, mass_center, ipAddress = '')
获取指定 IP 地址机械臂上六维力传感器感知到的负载质量和质心。 适配臂型：通用医疗臂、Thor3。 **参数：** mass：输出参数，用于接收六维力传感器质量数据。 mass_center：输出参数，用于接收六维力传感器质心数据，元组大小为 3。质心数据的含义依次为：mx、my、mz，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 ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.releaseBrake(ipAddress) time.sleep(2) center= [0.0]*3 mass= 0.0 DianaApi. getTcpPayloadWithSixAxis (mass, center, ipAddress) print("getTcpPayloadWithSixAxis:"+ str(mass) + " : " + str(center)) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv(ipAddress)

194 setTcpPayloadWithSixAxis

def setTcpPayloadWithSixAxis(mass, mass_center, ipAddress = '')
设置指定 IP 地址机械臂上六维力传感器感知到的负载质量和质心。设置后需调用 saveEnvironment 函数才能使设置永久生效。 适配臂型：通用医疗臂、Thor3。 **参数：** mass：输入参数，用于接收六维力传感器质量数据。 mass_center：输入参数，用于接收六维力传感器质心数据，元组大小为 3。质心数据的含义依次为：mx、my、mz，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 ipAddress = "192.168.10.75" DianaApi.initSrv((ipAddress,0,0,0,0,0)) DianaApi.releaseBrake(ipAddress) time.sleep(2) center = (0.47, 0.0, 0.036) mass = 2.0 DianaApi. setTcpPayloadWithSixAxis (mass, center, ipAddress) DianaApi.saveEnvironment(ipAddress) print("setTcpPayloadWithSixAxis:"+str(mass) + " : " +str(center)) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv(ipAddress)

195 setThresholdTorque

def setThresholdTorque (arrThreshold, ipAddress='')
设置指定 IP 地址机械臂各关节传感器检测阈值，此阈值在切换至阻抗模式时用于检测是否允许切换至阻抗模式。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** arrThreshold：表示各关节阈值 arrThreshold 的数组的首地址，数组长度为 JOINT_NUM，单位（Nm)。各关节允许的最大值分别为[18,18,15,15,6,6,6]，最小值皆为 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 arrThreshold = (6, 6, 5, 5, 2, 2, 2) ipAddress = '192.168.10.75' DianaApi. setThresholdTorque (arrThreshold, ipAddress)

196 getThresholdTorque

def getThresholdTorque (arrThreshold, ipAddress='')
获取指定 IP 地址机械臂各关节传感器检测阈值，此阈值在切换至阻抗模式时用于检测是否允许切换至阻抗模式。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** arrThreshold：表示各关节阈值 arrThreshold 的数组的首地址，数组长度为 JOINT_NUM，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： import DianaApi arrThreshold = (0, 0, 0, 0, 0, 0, 0) ipAddress = '192.168.10.75' DianaApi. getThresholdTorque (arrThreshold, ipAddress) import DianaApi arrThreshold = [0] * JOINT_NUM ipAddress = '192.168.10.75' DianaApi. getThresholdTorque (arrThreshold, ipAddress) print('arrThreshold: %f, %f, %f, %f, %f, %f, %f\n' %(arrThreshold[0], arrThreshold[1], arrThreshold[2], arrThreshold[3], arrThreshold[4], arrThreshold[5], arrThreshold[6]))

197 setStaticFriction

def setStaticFriction (enable, velocity, force, torque ,ipAddress = '')
设置指定 IP 地址机械臂的静摩擦力，静摩擦力可以在自由驱动时用于弥补负载不匹配导致的机械臂漂移。 适配臂型：通用医疗臂、定制医疗臂。 **参数：** enable：输入参数，表示静摩擦力的开启与关闭，bool 类型。 velocity：输入参数，当任意关节的速度超过该值时，静摩擦力停止生效，为浮点类型，单位 rad/s，输入范围[0.01,0.5]。 force：输入参数，表示当静摩擦力生效时，在平移方向上，末端所允许补偿的最大力值，double 类型，单位 N，输入范围[0,10]。 torque：输入参数，表示当静摩擦力生效时，在旋转方向上，末端所允许补偿的最大力

矩值，double 类型，单位为 N.m，输入范围[0,2]。

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：

```
velocity = 0.1
force = 3
torque = 0.5;
ipAddress = "192.168.10.75"
if  setStaticFriction(True, velocity, force, torque ,ipAddress) == True:
   print("Diana API setStaticFriction failed!\n")
```

198 getStaticFriction

```
def getStaticFriction(static_friction_param , ipAddress = "")
```

获取指定 IP 地址机械臂的静摩擦力状态，静摩擦力可以在自由驱动时用于弥补负载不匹配导致的机械臂漂移。

适配臂型：通用医疗臂、定制医疗臂。

参数：

static_friction_param ：输入输出参数，表示静摩擦力相关的参数的列表，包含四个参数:

参数 1 表示静摩擦力的开启与关闭，bool 类型；

参数 2 表示关节速度阈值，，单位 rad/s；

参数 3 表示表示平移方向上，末端所允许的阈值，单位 N；

参数 4 表示旋转方向上，末端所允许的阈值，单位为 N.m

strIpAddress：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：

```
staticFrictionParam = []
ipAddress = "192.168.10.75"
if getStaticFriction(staticFrictionParam, strIpAddress) == False:
   print("Diana API getStaticFriction failed!\n")
else:
```

```
print(staticFrictionParam)
```

199 setPredictMoveMode

```
def setPredictMoveMode(enable, samples = 1000, ipAddress = '')
```

设置当前系统是否处于运动预测模式。

a. 当系统处于运动预测模式时，所有可以使机器人移动的功能都将被禁用。

b. 设置系统模式为运动预测模式或普通模式（即非运动预测模式）。仅当当前系统为位置模式时才可以进入运动预测模式。如果当前为非位置模式，进入运动预测模式会失败，但退出运动预测模式会返回成功，但对系统无任何操作。

c. 当系统被设置为预测模式后，如需修改每个路段插补个数，无需退出运动预测模式，再此调用进入该模式即可更新。

注 1：仅当系统处于位置模式时可以切换到运动预测模式。在运动预测模式下，仅支持 moveL、moveJ、moveC、runPath、moveJToPose、moveJToTarget、moveLToPose、moveLToTarget、runComplexPath 几个指令的预测。并且在该模式下除 initSrv、destroySrv、releaseBrake、holdBrake、enterSafetyIdle、leaveSafetyIdle、freeDrivingForCurrentLoop、getCurrentLoopFreeDrivingState、cleanErrorInfo、定制臂 1.0 的寻零和所有 getXXX 系列 API 可以正常使用外，其他 API 均返回失败（返回值 False），但不报错误码。

注 2：预测模式下，当前仅支持一条 move 指令（包括：moveL、moveJ、moveC、moveJToPose、moveJToTarget、moveLToPose、moveLToTarget）的预测，如需要多路段预测请使用 runPath 或 runComplexPath。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数说明：

enable：输入参数，bool 型变量。当 enable= True，代表进入预测模式，此时 samples 的值有效；当 enable= False，代表退出预测模式，此时 samples 的值会被忽略。

samples：输入参数，int 型变量。代表每个路段插补个数，仅当 enable 为 True 时有效。该参数的取值范围为[10, 1000]。默认值为 1000。

strIpAddress：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：

```
ipAddress = "192.168.10.75"
ret = DianaApi.setPredictMoveMode(True, 948, ipAddress)
print("setPredictMoveMode return ", ret)
```

```
ret = DianaApi.getPredictMoveMode(ipAddress)
print("getPredictMoveMode return ", ret)
joints1 = [to_rad(0),to_rad(0),to_rad(0),to_rad(90),to_rad(0),to_rad(-90),to_rad(0)]
ret = DianaApi.moveJToTarget(joints1, 0.2,0.2)
print("predict mode: moveJToTarget 1 return ", ret)
ret = DianaApi.setPredictMoveMode(False, -1, ipAddress)
print("setPredictMoveMode return ", ret)
ret = DianaApi.getPredictMoveMode(ipAddress)
print("getPredictMoveMode return ", ret);
```

200 getPredictMoveMode

def getPredictMoveMode(ipAddress = '')

获取当前系统是否处于运动预测模式。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数说明：

strIpAddress：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：当前系统为运动预测模式。

False：当前系统为普通模式（即非运动预测模式）。

调用示例：

详见 setPredictMoveMode

201 setSafetySpeedLimit

def setSafetySpeedLimit(tcpSafetySpeedLimit, jointsSafetySpeedLimit,ipAddress = '')

设置指定 IP 地址机械臂在全模式下所允许的最大速度，当满足某个限制条件时，机械臂会自动进入位置模式，并且关闭抱闸。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数说明：

tcpSafetySpeedLimit 输入参数，大小为 6 的元组，指定笛卡尔空间允许的最大速度。其中前三个参数表示平移方向的速度限值（单位 m/s），设定的值应不小于 0.05 且不大于机械限定的笛卡尔空间平移速度的最大值(参考 getMechanicalMaxCartVelAcc)，当超过该值时，会自动修改设定的值为该最大值。后三个参数表示旋转方向的速度限值（单位 rad/s），设定的值应不小于 0.2 且不大于机械限定的笛卡尔空间旋转速度的最大值(参考 getMechanicalMaxCartVelAcc)，当超过该值时，会自动修改设定的值为该最大值。如果将某方向设为 0，则默认为该方向上的速度不做限制。

jointsSafetySpeedLimit：输入参数，大小为关节数量的元组，指定关节空间允许的最大速度（单位 rad/s），设定的值应不小于 0.2 且不大于机械限定的关节空间转动速度的最大值(参考 getMechanicalMaxJointsVel)，当超过该值时，会自动修改设定的值为该最大值。如

果将某关节设为 0，则默认为该关节上的速度不做限制。

strIpAddress：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：

```
ipAddress = "192.168.10.75"
tcpSafetySpeedLimit=(0.5,0.5,0.5,1.57,1.57,1.57)
jointsSafetySpeedLimit=(0.5,0.5,0.5,0.6,0.5,0.5,0.5)
ret = DianaApi.setSafetySpeedLimit(tcpSafetySpeedLimit, jointsSafetySpeedLimit, ipAddress)
print("setSafetySpeedLimit return %d", %(ret))
tcpSafetySpeedLimit1=[0.0]*6
jointsSafetySpeedLimit1=[0.0] * 7
ret =
DianaApi.getSafetySpeedLimit(tcpSafetySpeedLimit1,jointsSafetySpeedLimit1,ipAddress)
print("getSafetySpeedLimit return %d", %(ret))
print(tcpSafetySpeedLimit1)
print(jointsSafetySpeedLimit1)
```

202 getSafetySpeedLimit

def getSafetySpeedLimit(tcpSafetySpeedLimit, jointsSafetySpeedLimit, ipAddress = '')

获取指定 IP 地址机械臂在全模式下所允许的最大速度。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数说明：

tcpSafetySpeedLimit：输出参数，大小为 6 的列表，指定笛卡尔空间允许的最大速度，其中前三个参数单位为 m/s，后三个参数单位为 rad/s。

jointsSafetySpeedLimit：输出参数，大小为关节数量的列表，指定关节空间允许的最大速度，单位 rad/s。

strIpAddress：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 setSafetySpeedLimit

203 setSafetyCartZone

def setSafetyCartZone(tcpPose, cartFreeZone,ipAddress = '')

设置指定 IP 地址机械臂的安全运动空间，该空间是一个以指定 TCP 位置为中心点，指定

长、宽、高而形成的立方体。该安全空间在非安全处理模式下均生效。当机械臂在运动过程中 TCP 超出该安全空间后，机械臂会立即停止、关闭抱闸，并切换至位置模式。此时用户可以通过安全处理方式将机械臂驱回安全运动空间。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数说明：

tcpPose：输入参数，大小为 3 的元组，表示安全立方体的中心点，为基坐标系下指定 TCP 的位置，单位为 m。

cartFreeZone：输入参数，大小为 3 的元组，表示安全立方体的长、宽、高，单位为 m。当设置为 0 时，则表示该方向不做限制。

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：

```
ipAddress = "192.168.10.75"
tcpPose=(0.5994,0.005,0.7045)
cartFreeZone=(0.6,0.6,0.6)
ret = DianaApi.setSafetyCartZone(tcpPose, cartFreeZone, ipAddress)
print("setSafetyCartZone return %d", %(ret))
tcpPose1=[0.0]*6
cartFreeZone1=[0.0] * 7
ret = DianaApi.getSafetyCartZone (tcpPose1, cartFreeZone1,ipAddress)
print("getSafetyCartZone return %d", %(ret))
print(tcpPose1)
print(cartFreeZone1)
```

204 getSafetyCartZone

def getSafetyCartZone(tcpPose, cartFreeZone, ipAddress = '')

获取指定 IP 地址机械臂 TCP 的安全运动空间，该空间是一个以指定 TCP 位置为中心点，指定长、宽、高而形成的立方体。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数说明：

tcpPose：输出参数，大小为 3 的列表，表示安全立方体的中心点，为基坐标系下指定 TCP 的位置，单位为 m。

cartFreeZone：输出参数，大小为 3 的列表，表示安全立方体的长、宽、高，单位为 m。

strIpAddress：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 setSafetyCartZone

205 getEncoderBatteryLevel

def getEncoderBatteryLevel(battery_level : list,ipAddress = '')
获取机械臂关节编码器电源电量等级。 适用臂型： Thor3。 **参数：** battery_level：输出参数，大小为关节数的列表。1 代表电源电量正常；0 代表电源电量低；-1 代表编码器不可用，电量耗尽。 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： bat = [0]*7 ret = DianaApi.getEncoderBatteryLevel(bat) print("getEncoderBatteryLevel return (", ret, "):", bat)

206 getWarningList

def getWarningList(warning_list : list,ipAddress = '')
获取指定 IP 地址机械臂的系统告警码列表。 适用臂型：Diana、 Thor3 **参数：** warning_list：输出参数，机械臂系统告警码列表，最大长度为 MAX_WARNING_LIST_SIZE（暂定为 10）。列表用于接收系统当前所有告警码（WARNING_CODE）。 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：当前系统告警数量。 -1：失败。
调用示例： lstWarning = [0]*10 ret = DianaApi.getWarningList(lstWarning) print("getWarningList return (", ret, "):", lstWarning)

```
ret = DianaApi.cleanWarning()
print("cleanWarning ret (", ret, ")")


lstWarning = [0]*10
ret = DianaApi.getWarningList(lstWarning)
print("getWarningList return (", ret, "):", lstWarning)
```

207 formatWarning

```
def formatWarning(w, ipAddress = '')
```

获取指定 IP 地址机械臂的告警码 w 的字符串描述，该告警码可以通过调用 getWarningList 得到。

适配臂型：通用医疗臂、定制医疗臂、Thor3。

参数：

w：告警码。

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：

告警码对应的描述信息。

调用示例：

详见 getWarningList 。

208 cleanWarning

```
def cleanWarning(ipAddress = '')
```

清除指定 IP 地址机械臂的所有普通告警。

适用臂型：Diana、Thor3

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：

详见 getWarningList 。

209 calibrateSixAxisSensor

```
def calibrateSixAxisSensor(ipAddress = '')
```

标定指定 IP 地址机械臂的零偏，使用本功能需要六维力传感器已进行过一次标定。标定结果不会存储到配置文件，如希望下次开机后结果依然生效，则应当调用 saveEnvironment。本功能仅支持位置模式下使用。 适配臂型：通用医疗臂、Thor3。 **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：
DianaApi.enableSixAxis(DianaApi.six_axis_type_e.SIX_AXIS_IN_FREE_DRIVING,True)
ret = DianaApi.calibrateSixAxisSensor()
if ret == 0:
print("calibrateSixAxisSensor succeeded\n")
DianaApi.saveEnvironment()
else:
print("calibrateSixAxisSensor failed\n")

210 calcInstallAngle

def calcInstallAngle(tcpPose1, tcpPose2, tcpPose3, angle, ipAddress='')
用户通过自由驱动 (freeDriving) 在水平面上示教三个点，获取 (getTcpPos) 其位姿，通过这三个点的位姿计算基坐标系安装角度。 示教的三个输入点位的要求： 1. 三点需要可以构成有效平面，即三点不重合且不共线。 2. 三点在水平面按照逆时针分布 (俯视图)。 适配臂型：通用医疗臂、定制医疗臂、Thor3。 **参数：** dblPose1：输入参数，大小为 6 的列表，用户示教的第一个点的位姿，其中前三个参数单位为 m，后三个参数单位为 rad，后三个角度是轴角。 dblPose2：输入参数，大小为 6 的列表，用户示教的第二个点的位姿，其中前三个参数单位为 m，后三个参数单位为 rad，后三个角度是轴角。 dblPose3：输入参数，大小为 6 的列表，用户示教的第三个点的位姿，其中前三个参数单位为 m，后三个参数单位为 rad，后三个角度是轴角。 angle：输出参数，大小为 3 的列表，机械臂基坐标的安装角度，单位为 rad，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：

```
tcpPose1 = [0.511716, -0.17627, 0.546576, to_rad(-0.763316), to_rad(0.567477), to_rad(-37.009011)]
tcpPose2 = [0.511716, -0.17627, 0.593378, to_rad(0.320060), to_rad(0.528476), to_rad(-21.828699)]
tcpPose3 = [0.511716, -0.15027, 0.593378, to_rad(0.3833975), to_rad(0.716449), to_rad(-24.986168)]

print("请在平面上逆时针示教三个点:")
freeDriving(True)
print("示教第一个点，示教到点位后按任意键：")
sys.stdin.read(1)
getTcpPos(tcpPose1)

print("示教第二个点，示教到点位后按任意键：")
sys.stdin.read(1)
getTcpPos(tcpPose2)

print("示教第三个点，示教到点位后按任意键：")
sys.stdin.read(1)
getTcpPos(tcpPose3)

angle = [0, 0, 0]
calcInstallAngle(tcpPose1, tcpPose2, tcpPose3, angle, strIpAddr)
print("angle:" + str(to_degree(angle[0])) + ", " + str(to_degree(angle[1])) + ", " + str(to_degree(angle[2])))
```

211 getFunctionOptR8

```
def getFunctionOptR8(function_index:function_index_e, opt_name:function_opt_name_e, ipAddress = '')
```

获取指定功能的高精度浮点型参数。

适用臂型：通用医疗臂

参数：

function_index：指定功能的名称，取值为 function_index_e 枚举类型，枚举值及含义如下:

function_index	对应枚举	含义
0	function_index_e::E_FREEDRIVING	自由驱动相关
1	function_index_e::E_CART_IMPEDANCE	力控和笛卡尔阻抗相关

opt_name：取值为 function_opt_name_e 类型，用于指定功能的属性，枚举值及含义如下:
当指定功能为自由驱动时，

opt_name	对应枚举	含义
0x10006	function_opt_name_e:: E_MAX_TRANSLATION_DECELERATION	获取实时虚拟墙内减速区域的最大平移速度
0x10009	function_opt_name_e::E_MAX_ALLOWED_MOVE_ZONE	获取导纳自由驱动的最大活动距离。
0x1000A	function_opt_name_e::E_MAX_ALLOWED_MOVE_SPEED	获取导纳自由驱动的极限速度。

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：

包含两个元素的元组，第一个元素表示获取是否成功，

True：获取成功，

False：获取失败。

第二个元素表示获取属性的结果，opt_name 输入参数不同，获取到的结果的含义也不同，结果代表的含义如下。

opt_name	含义
function_opt_name_e:: E_MAX_TRANSLATION_DECELERATION	获取实时虚拟墙内减速区域的最大平移速度
function_opt_name_e::E_MAX_ALLOWED_MOVE_ZONE	设置导纳自由驱动的最大活动距离。如果当前为阻抗自由驱动，则此参数设置无效。
function_opt_name_e::E_MAX_ALLOWED_MOVE_SPEED	设置导纳自由驱动的极限速度。如果当前为阻抗自由驱动，则此参数设置无效。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.initSrv((ipAddress,0,0,0,0,0))
if ret == True:
    time.sleep(2)
    DianaApi.releaseBrake()
    ret = DianaApi.setFunctionOptI4(function_index_e.E_FREEDRIVING, function_opt_name_e.E_ADMITTANCE_CONTROL, 1)
    if  ret == True:
        print("succeed to enable admittance control!\n")
        ret = DianaApi.setFunctionOptI4(function_index_e.E_FREEDRIVING, function_opt_name_e.E_ENABLE_REALTIME_VIRTUAL_WALL,1)
        if ret == True:
            print("succeed to enable realtime virtual wall!\n")
            ret = DianaApi.getFunctionOptI4(function_index_e.E_FREEDRIVING, function_opt_name_e.E_ENABLE_REALTIME_VIRTUAL_WALL)
            if  ret[0] == True:
```

```
            if ret[1] == 1 :
                print("realtime virtual wall is enable!\n")
                ret = DianaApi.setFunctionOptR8(function_index_e.E_FREEDRIVING,
function_opt_name_e.E_MAX_TRANSLATION_DECELERATION, 0.1)
                if ret == True:
                    print("succeed to set max translation deceleration in realtime virtual wall!\n")
                    ret = DianaApi.getFunctionOptR8(function_index_e.E_FREEDRIVING,
function_opt_name_e.E_MAX_TRANSLATION_DECELERATION)
                    if ret[0] == True:
                        print("current max translation deceleration is %lf in realtime virtual wall!\n" % (ret[1]))
                        DianaApi.freeDrivingWithLockedJoint_ex(True,0,[1,0,0,0,0,0],0)
                        enteredDecelerationBoundary = False
                        distance2HazardousBoundary = 1.0
                        safetyDirection = [1, 0, 0]
                        for i in range(0,1000):
                            if i < 500:
                                enteredDecelerationBoundary = False
                            else:
                                enteredDecelerationBoundary = True
                            distance2HazardousBoundary = 1 - i * 0.0005
                            DianaApi.updateRealtimeVirtualWall(enteredDecelerationBoundary,distance2HazardousBoundary,safetyDirection)
                            time.sleep(0.001)
                        DianaApi.freeDrivingWithLockedJoint_ex(False,0,[1,0,0,0,0,0],0)
    ret = DianaApi.setFunctionOptI4(function_index_e.E_FREEDRIVING, function_opt_name_e.E_ENABLE_REALTIME_VIRTUAL_WALL,1)
    if ret == True:
        print("succeed to disable realtime virtual wall!\n")
    ret = DianaApi.setFunctionOptI4(function_index_e.E_FREEDRIVING, function_opt_name_e.E_ADMITTANCE_CONTROL, 0)
    if ret == True:
        print("succeed to disable admittance control!\n")
DianaApi.holdBrake()
DianaApi.destroySrv()
```

212 setFunctionOptR8

def setFunctionOptR8(funtion_index:function_index_e, opt_name:function_opt_name_e, opt_value, ipAddress='')

设置指定功能的高精度浮点型参数。
适配臂型：通用医疗臂

参数：

function_index：指定功能的名称，取值为 function_index_e 枚举类型，枚举值及含义如下，

function_index	对应枚举	含义
0	function_index_e.E_FREEDRIVING	自由驱动相关
1	function_index_e.E_CART_IMPEDANCE	力控和笛卡尔阻抗相关

opt_name：取值为 function_opt_name_e 类型，用于指定功能的属性，枚举值及含义如下，
当指定 function_index 为 function_index_e.E_FREEDRIVING(自由驱动)时，支持参数如下

opt_name	对应枚举	含义
0x10006	function_opt_name_e::T_MAX_TRANSLATION_DECELERATION	设置实时虚拟墙内减速区域的最大平移速度
0x10009	function_opt_name_e::T_MAX_ALLOWED_MOVE_ZONE	设置导纳自由驱动的最大活动距离。如果当前为阻抗自由驱动，则此

		参数设置无效。
0x1000A	function_opt_name_e::T_MAX_ALLOWED_MOVE_SPEED	设置导纳自由驱动的极限速度。如果当前为阻抗自由驱动，则此参数设置无效。

opt_value：输入参数，设置的对应属性的 double 类型值。
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：

参考 getFunctionOptR8

213 updateRealtimeVirtualWall

def updateRealtimeVirtualWall(enteredDeclerationBoundary:bool, distance2HazardousBoundary:float,safetyDirection:tuple, ipAddress='')

实时指令，设置指定 IP 地址机械臂的实时虚拟墙参数。目前仅支持在导纳模式（需保证六维力传感器在线），进行定向自由驱动（需要保证自由度只存在 1~5 个分量）时设置。机械臂会根据收到的实时指令参数实现任意形状的虚拟墙效果。目前虚拟墙的效果如下：机械臂会按照收到的危险边界距离参数和当前自身速度在危险方向上的速度分量判断是否会导致机械臂触碰危险边界，如判断导致碰撞，则开始在危险方向上进行减速，确保机械臂以确定的加速度停止在危险边界上。

适配臂型：通用医疗臂、Thor3。

关联函数：

开启实时虚拟墙请参考函数 setFunctionOptI4。

查询实时虚拟墙的状态请参考 getFunctionOptI4。

设置虚拟墙的减速区的最大平移速度请参考 setFunctionOptR8。

查询虚拟墙的减速区的最大平移速度请参考 getFunctionOptR8。

参数：

enteredDecelerationBoundary：输入参数，bool 变量，表示是否进入减速边界。

distance2HazardousBoundary：输入参数，浮点类型，表示距离危险边界的距离，单位 m，由上位机实时发送。

safetyDirection：输入参数，大小为 3 的浮点类型元组，安全方向矢量[X, Y, Z]，由上位机实时发送。

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。

True：成功。

False：失败。
调用示例： 参考 getFunctionOptR8

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_RELEASE_BRAKE_FAILED	-1011	打开抱闸失败
ERROR_CODE_HOLD_BRAKE_FAILED	-1012	关闭抱闸失败
ERROR_CODE_IP_ADDRESS_NOT_REGISTER	-1013	该 IP 机械臂尚未 initSrv
ERROR_CODE_ROBOTARM_OVERNUMBER	-1014	超过最大支持机械臂数
ERROR_CODE_SOCKET_OTHER_ERROR	-1015	其他 socket 连接错误
ERROR_CODE_SOCKET_CONNECTED_ALREADY	-1016	socket 已建立连接
ERROR_CODE_UNKNOWN_ROBOTSTATE	-1017	未知机械臂状态
ERROR_CODE_JOINT_REGIST_ERROR	-2001	硬件错误
ERROR_CODE_CURRENT_OFFSET_ERROR	-2002	电流偏置错误
ERROR_CODE_EQEP_ENCODER	-2003	高速侧编码器正交编码信号错误
ERROR_CODE_SPIENCODER	-2004	与低速侧编码器通信断开
ERROR_CODE_HALL_SENSOR	-2005	开关 Hall 相序错误
ERROR_CODE_CURRENT_BUS_OVERTIME	-2006	母线电流长时间过载
ERROR_CODE_CURRENT_IQ_OVERTIME	-2007	电机三相电流长时间过载
ERROR_CODE_POSITION_CMD_STEP	-2008	位置指令阶跃错误
ERROR_CODE_TORQUE_SENSOR	-2009	扭矩传感器信号故障 1
ERROR_CODE_EEPROM_READ	-2010	关节读 EEPROM 参数错误
ERROR_CODE_EEPROM_WRITE	-2011	关节写 EEPROM 参数错误

ERROR_CODE_LS_ENCODER_OVERSPEED	-2012	低速侧编码器反馈位置超速
ERROR_CODE_LS_ENCODER_FB_ERROR	-2013	低速侧编码器反馈数据错误
ERROR_CODE_MS_SINGAL_Z_ERROR	-2014	高速侧编码器 Z 信号异常
ERROR_CODE_THREE_PHASE_CURRENT	-2015	电机三相电流瞬时过流
ERROR_CODE_TORQUE_SENSOR_READ_ERROR	-2016	扭矩传感器读取故障
ERROR_CODE_COMMUNICATE_ERROR	-2101	底层通信失败（ln）
ERROR_CODE_LOST_HEART_WITH_DIANAROBOT_ERROR	-2102	与后台服务心跳断开
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
ERROR_CODE_OUT_RANGE_FEEDBACK	-2206	关节角反馈超软限位
ERROR_CODE_EMERGENCY_STOP	-2207	急停已拍下
ERROR_CODE_NO_INIT_PARAMETER	-2208	找不到关节初始参数
ERROR_CODE_NOT_MATCH_LOAD	-2209	负载与理论值不匹配
ERROR_CODE_CANNOT_MOVE_WHILE_FREE_DRIVING	-2210	自由驱动模式不能执行其他运动
ERROR_CODE_CANNOT_MOVE_WHILE_ZERO_SPACE_FREE_DRIVING	-2211	零空间自由驱动模式下不能执行其他运动
ERROR_CODE_PASSTHROUGH_FIFO_OVERFLOW_UP_ERROR	-2212	透传上溢出
ERROR_CODE_PASSTHROUGH_FIFO_OVERFLOW_DOWN_ERROR	-2213	透传下溢出
ERROR_CODE_ROBOT_IN_VIRTUAL_WALL	-2214	有关节在虚拟墙内
ERROR_CODE_CONFLICT_TASK_RUNNING	-2215	运动任务冲突
ERROR_CODE_OUT_OF_PHYSICAL_RANGE_FEEDBACK	-2216	超出物理限位
ERROR_CODE_PARAMETER_POINTER_EQUALS_NULLPTR	-2217	非法参数传递
ERROR_CODE_ PARAMETER_EQUALS_NAN_OR_INF	-2218	输入参数存在 nan 或 inf
ERROR_CODE_UNKNOWN_FIRMWARE_VERSION	-2219	未知的固件版本
ERROR_CODE_INPUT_OUT_OF_EXTREME_POSITION_RANGE	-2220	输入超出关节极限位置
ERROR_CODE_INPUT_OUT_OF_PHYSICAL_POSITION_RANGE	-2221	输入超出物理限位
ERROR_CODE_NOT_IN_SAFETY_HANDLE_MODE	-2222	当前未处于安全处理模式
ERROR_CODE_INCOMPATIBLE_FIRMWARE_VERSION	-2223	电流环自由驱动不兼容

		的固件版本
ERROR_CODE_VEL_OR_ACC_PARAMETER_OUT_OF_RANGE	-2232	速度或加速度超限
ERROR_CODE_POINTS_CANT_FORM_PLANE	-2233	三点无法构成有效平面
ERROR_CODE_PLAN_ERROR	-2301	路段规划失败
ERROR_CODE_INTERPOLATE_POSITION_ERROR	-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_ PLANNER_ERROR	-2305	规划失败
ERROR_CODE_ HOME_POSITION_ERROR	-2306	需要寻零
ERROR_CODE_ FATAL	-2307	严重错误(关节位置超出物理极限)
ERROR_CODE_ POS_LIMIT	-2308	位置超出限制
ERROR_CODE_ FORCE_LIMIT	-2309	关节力矩超出限制
ERROR_CODE_ SPEED_LIMIT	-2310	速度超出限制
ERROR_CODE_ ACC_LIMIT	-2311	加速度超出限制
ERROR_CODE_ JERK_LIMIT	-2312	加加速度超出限制
ERROR_CODE_ MOTION_LIMIT	-2313	位置超出限制
ERROR_CODE_ IK_TRACK	-2314	轨迹跟踪过程逆解求解失败
ERROR_CODE_ IK_GENERAL	-2315	通用位置逆解求解失败
ERROR_CODE_ PLAN_INPUT	-2316	轨迹规划输入错误
ERROR_CODE_ PLAN_MOVJ	-2317	关节空间轨迹规划失败
ERROR_CODE_ PLAN_MOVL	-2318	直线轨迹规划失败
ERROR_CODE_ PLAN_MOVC	-2319	圆弧轨迹规划失败
ERROR_CODE_ PLAN_BLEND	-2320	过渡轨迹规划失败
ERROR_CODE_ PLAN_SPDJ	-2321	SpeedJ 轨迹规划失败
ERROR_CODE_ PLAN_SPDL	-2322	SpeedL 轨迹规划失败
ERROR_CODE_ PLAN_SRVJ	-2323	ServoJ 轨迹规划失败
ERROR_CODE_ PLAN_SRVL	-2324	ServoL 轨迹规划失败
ERROR_CODE_ MOVE_UNKNOWN	-2325	未知运动类型或运动类型不匹配
ERROR_CODE_ MOVE_UNPLAN	-2326	轨迹未规划
ERROR_CODE_ MOVE_INPUT	-2327	轨迹插补输入错误
ERROR_CODE_ MOVE_INTERP	-2328	轨迹插补失败
ERROR_CODE_ PLAN_TRANSLATION	-2329	移动规划失败
ERROR_CODE_ PLAN_ROTATION	-2330	旋转规划失败
ERROR_CODE_ PLAN_JOINTS	-2331	关节规划失败
ERROR_CODE_ UNMATCHED_JOINTS_NUMBER	-2332	零空间自由驱动关节数

		不匹配
ERROR_CODE_ TCPCALI_FUTILE_WPS	-2333	示教点不合理
ERROR_CODE_ TCPCALI_FIT_FAIL	-2334	拟合 TCP 失败
ERROR_CODE_ DHCALI_FIT_WF_FAIL	-2335	DH 参数初始化世界坐标系失败
ERROR_CODE_ DHCALI_FIT_TF_FAIL	-2336	DH 参数初始化工具坐标系失败
ERROR_CODE_DHCALI_FIT_DH_FAIL	-2337	DH 参数拟合失败
ERROR_CODE_ DHCALI_INIT_FAIL	-2338	DH 参数初始化失败
ERROR_CODE_ SLFMOV_SINGULAR	-2339	零空间运动至奇异位置
ERROR_CODE_ SLFMOV_FUTILE	-2340	零空间运动在笛卡尔空间内无效
ERROR_CODE_SLFMOV_JNTLIM	-2341	零空间运动至关节限位
ERROR_CODE_ SLFMOV_SPDLIM	-2342	零空间运动至关节限位
ERROR_CODE_ SLFMOV_FAIL	-2343	零空间运动插补失败
ERROR_CODE_ SLFMOV_FFC_FAIL	-2344	零空间运动前馈补偿错误
ERROR_CODE_ LOADIDENT_INIT_FAIL	-2345	负载辨识初始化失败
ERROR_CODE_ LOADIDENT_UFB_FAIL	-2346	负载辨识更新反馈数据错误
ERROR_CODE_ LOADIDENT_FIT_FAIL	-2347	负载辨识失败
ERROR_CODE_ LOADIDENT_NONLOADED	-2348	未检测到有效负载
ERROR_CODE_RESOURCE_UNAVAILABLE	-3001	参数错误
ERROR_CODE_DUMP_LOG_TIMEOUT	-3002	导出 Log 文件超时
ERROR_CODE_DUMP_LOG_FAILED	-3003	导出 Log 文件失败
ERROR_CODE_RESET_DH_FAILED	-3004	重置 DH 参数失败
ERROR_CODE_SIX_AXIS_DEVICE_OFFLINE	-3005	六维力离线

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。

附件 B：

表 2：Diana API 接口告警码表

系统告警宏定义	告警码	说明
WARNING_CODE_RAID_PARTIAL_INVALID	-1001	部分磁盘阵列不可用
WARNING_CODE_JOINT_1_ENCODER_BATTERY_LOW	-1002	关节 1 编码器电源电量低
WARNING_CODE_JOINT_2_ENCODER_BATTERY_LOW	-1003	关节 2 编码器电源电量低
WARNING_CODE_JOINT_3_ENCODER_BATTERY_LOW	-1004	关节 3 编码器电源电量低
WARNING_CODE_JOINT_4_ENCODER_BATTERY_LOW	-1005	关节 4 编码器电源电量低
WARNING_CODE_JOINT_5_ENCODER_BATTERY_LOW	-1006	关节 5 编码器电源电量低
WARNING_CODE_JOINT_6_ENCODER_BATTERY_LOW	-1007	关节 6 编码器电源电量低
WARNING_CODE_JOINT_7_ENCODER_BATTERY_LOW	-1008	关节 7 编码器电源电量低
WARNING_CODE_PREDICT_CONFLICTED_CALLING	-1009	预测模式下的冲突调用

附件 C：

确保运动学逆解唯一性的解决方案

Diana 机械臂为七自由度机械臂，由于多了一个冗余自由度，理论上存在无数多组逆解(根据末端位姿求解关节角)，在实际应用中，当执行逆解运算或者进行笛卡尔空间运动时，有可能出现逆解不唯一的情况。为了确保逆解的唯一性，可采取如下解决方案。

第一种情况: 已知其中某个路点对应的关节角。

例：已知 A 点关节角 Joints_A 和 B 点位姿 Pose_B，机械臂在两点之间进行往复运动。

解决方案：向目标点 A 运动时，调用 moveJToTarget 或 moveLToTarget 函数。

```
# A 点关节角
Joints_A=(0.000000, 0.523599, 0.000000, 1.570796, 0.000000, -0.872665, 0.000000)
# B 点位姿
pose_B=(0.5, 0.5, 0.5, 0, 0, 0)
# 直线运动的速度与加速度
velL=0.2
accL=0.8
for i in range(0, 10):
    # 调用 moveJToTarget 或 moveLToTarget 函数移动至目标点 A
    DianaApi.moveLToTarget(Joints_A, velL, accL)
    wait_move()
    # 调用 moveJToPose 或 moveLToPose 函数移动至目标点 B
    DianaApi.moveLToPose (pose_B, velL, accL)
    wait_move()
```

第二种情况: 所有路点对应的关节角均未知。

例：已知 A 点位姿 Pose_A 和 B 点位姿 Pose_B，机械臂在两点之间进行往复运动。

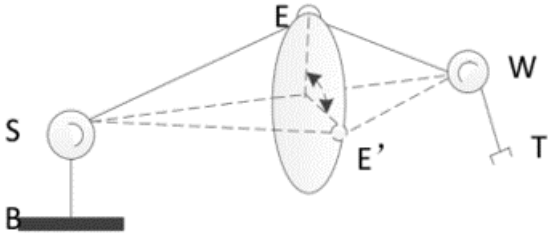
解决方案：首先在 A 点和 B 点附近分别示教一个参考点，并记录下参考点位下的机械臂关节角 q_ref_A 和 q_ref_B，然后利用 inverse_ext 函数，求解目标位姿下相对于参考点关节角距离最近的逆解，最后调用 moveJToTarget 或 moveLToTarget 函数进行运动。

```
#A 点与 B 点的位姿
Pose_A = (0.5, 0.5, 0.5, 0, 0, 0)
Pose_B = (0.4, 0.6, 0.2, 0, 0, 0)
```

```
    # 示教两个参考点位并记录下关节角 q_ref_A, q_ref_B
    q_ref_A = (-0.645772, 0.261799, -0.157080, 1.675516, 0.052360, -1.186824, -0.802851)
    q_ref_B = (-0.645772, 0.261799, -0.157080, 1.675516, 0.052360, -1.186824, -0.802851)
    # 关节空间运动的速度与加速度
    velJ = 0.25
    accJ = 1.0
    # 直线运动的速度与加速度
    velL = 0.1
    accL = 0.4
    for i in range(0, 10):
        # 调用 inverse_ext 函数, 根据 q_ref_A 计算 Pose_A 所对应的关节角 Joints_A
        Joints_A = [0.0] * 7;
        DianaApi.inverse_ext(q_ref_A, Pose_A, Joints_A)
        # 调用 moveJToTarget 或 moveLToTarget 函数移动到目标点 A
        DianaApi.moveJToTarget(Joints_A, velJ, accJ)
        wait_move()
        # 调用 inverse_ext 函数, 根据 q_ref_B 计算 Pose_B 所对应的关节角 Joints_B
        Joints_B = [0.0] * 7
        DianaApi.inverse_ext(q_ref_B, Pose_B, Joints_B)
        # 调用 moveJToTarget 或 moveLToTarget 函数移动到目标点 B
        DianaApi.moveLToTarget(Joints_B, velL, accL)
        wait_move()
```

附件 D：

关于臂角



S，E，W 三点构成平面。当此平面与机械臂安装面垂直且 E 在上方时，臂角为 0 度，如图 1；

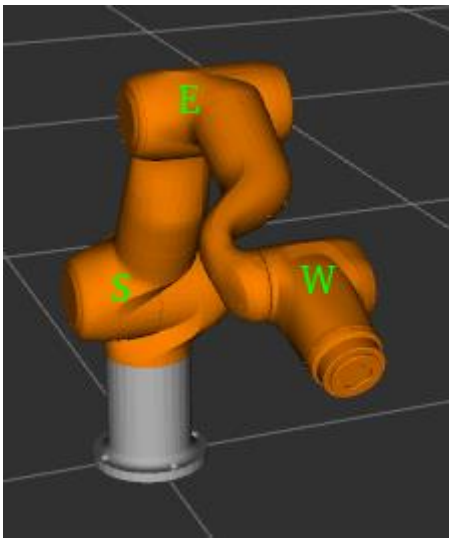


图 1

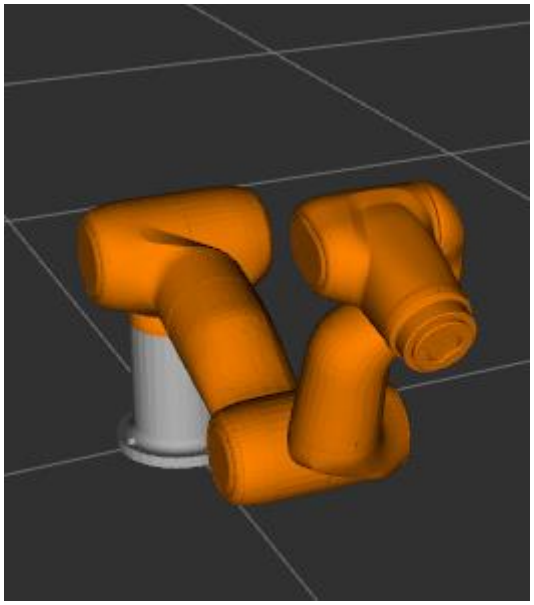


图 2

如图 2，E 在下方，此时臂角为±180 度

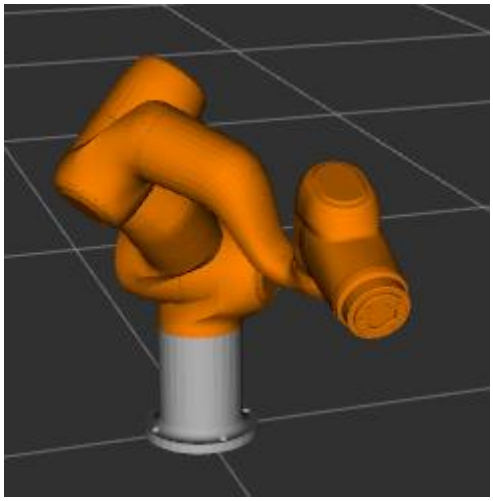


图 3

如图 3，臂角为正 30 度。臂角为±30 度时，添加一句话：此时 S, E, W 三点构成的平面与机械臂安装面夹角为 150 度。

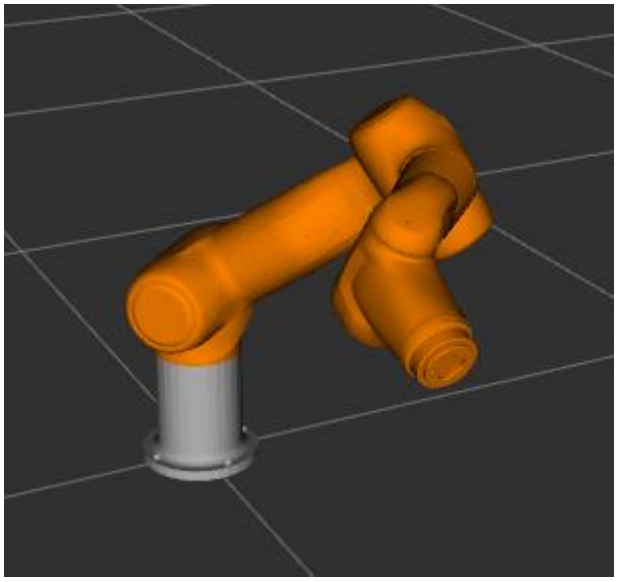


图 4

如图 4，臂角为负 30 度。臂角为±30 度时，添加一句话：此时 S, E, W 三点构成的平面与机械臂安装面夹角为 150 度。