
PRODUCT MANUAL

思灵机器人产品手册

工业版 API 手册（Python 版）

软件版本 V2.16.0

文档版本 V2.16.0

www.agile-robots.cn



目录

概述.....	1
兼容性说明.....	1
函数说明.....	1
1 initSrv.....	1
2 destroySrv	2
3 setPushPeriod	2
4 wait_move.....	3
5 moveTCP	3
6 rotationTCP.....	4
7 moveJoint.....	5
8 moveJToTarget.....	5
9 moveJToPose	6
10 moveJ	7
11 moveL	8
12 moveLToTarget	8
13 moveLToPose.....	9
14 speedJ	10
15 speedL	10
16 freeDriving.....	11
17 stop.....	11
18 forward.....	12
19 inverse	12
20 getJointPos	13
21 getJointAngularVel	13
22 getJointCurrent.....	13
23 getJointTorque.....	14
24 getTcpPos	14
25 getTcpExternalForce	14
26 releaseBrake	15
27 holdBrake	15
28 changeControlMode.....	15
29 getLibraryVersion	16
30 formatError	16
31 getLastError	16
32 setLastError.....	17
33 setDefaultActiveTcp.....	17
34 getLinkState	17
35 getTcpForce.....	18
36 getJointForce.....	18
37 isCollision	18
38 initDHCali.....	19
39 getDHCaliResult	19

40	setDH	20
41	setWrd2BasRT	20
42	setFla2TcpRT	21
43	getRobotState	21
44	resume	22
45	setJointCollision	23
46	setCartCollision	23
47	enterForceMode	23
48	leaveForceMode	24
49	setDefaultActiveTcpPose	24
50	setResultantCollision	25
51	setJointImpeda	25
52	getJointImpeda	25
53	setCartImpeda	26
54	getCartImpeda	26
55	zeroSpaceFreeDriving	27
56	createPath	27
57	addMoveL	28
58	addMoveJ	28
59	runPath	29
60	destroyPath	29
61	rpy2Axis	29
62	axis2RPY	30
63	homogeneous2Pose	30
64	pose2Homogeneous	30
65	enableTorqueReceiver	31
66	sendTorque_rt	31
67	enableCollisionDetection	31
68	setActiveTcpPayload	32
69	servoJ	32
70	servoL	33
71	servoJ_ex	33
72	servoL_ex	34
73	speedJ_ex	35
74	speedL_ex	36
75	dumpToUDisk	36
76	inverse_ext	37
77	getJointLinkPos	37
78	createComplexPath	37
79	addMoveLSegmentByTarget	38
80	addMoveLSegmentByPose	40
81	addMoveJSegmentByTarget	41
82	addMoveJSegmentByPose	42
83	addMoveCSegmentByTarget	43

84	addMoveCSegmentByPose.....	44
85	runComplexPath	45
86	destroyComplexPath	46
87	saveEnvironment.....	46
88	enterForceMode_ex	46
89	readDI	47
90	readDO	48
91	readAI	48
92	readAO	48
93	setAIMode.....	49
94	writeDO.....	49
95	writeAO.....	50
96	readBusCurrent	50
97	readBusVoltage	51
98	getDH.....	51
99	getOriginalJointTorque	52
100	getJacobiMatrix.....	52
101	resetDH	52
102	runProgram	53
103	stopProgram	53
104	getVariableValue	53
105	setVariableValue.....	54
106	isTaskRunning.....	54
107	pauseProgram.....	55
108	resumeProgram	55
109	stopAllProgram	55
110	isAnyTaskRunning.....	56
111	cleanErrorInfo	56
112	setCollisionLevel	56
113	getJointCount	57
114	getWayPoint	57
115	setWayPoint	58
116	addWayPoint	58
117	deleteWayPoint	59
118	getDefaultActiveTcp	59
119	getDefaultActiveTcpPose.....	60
120	getActiveTcpPayload	60
121	zeroSpaceManualMove.....	60
122	moveTcp_ex	61
123	setExternalAppendTorCutoffFreq.....	62
124	poseTransform	62
125	updateForce.....	63
126	inverseClosedFull	64
127	getInverseClosedResultSize.....	65

128	getInverseClosedJoints	65
129	destoryInverseClosedItems	65
130	getGravInfo	65
131	setGravInfo	66
132	getGravAxis	66
133	setGravAxis.....	66
134	speedLOnTcp	67
135	getTcpForceInToolCoordinate	67
136	calculateJacobi	68
137	calculateJacobiTF	68
138	getMechanicalJointsPositionRange	69
139	getMechanicalMaxJointsVel	69
140	getMechanicalMaxJointsAcc	70
141	getMechanicalMaxCartVelAcc	70
142	getJointsPositionRange	71
143	getMaxJointsVel.....	71
144	getMaxJointsAcc.....	71
145	getMaxCartTranslationVel	72
146	getMaxCartRotationVel	72
147	getMaxCartTranslationAcc	72
148	getMaxCartRotationAcc	73
149	setJointsPositionRange	73
150	setMaxJointsVel	73
151	setMaxJointsAcc	74
152	setMaxCartTranslationVel	74
153	setMaxCartRotationVel.....	74
154	setMaxCartTranslationAcc	75
155	setMaxCartRotationAcc.....	75
156	requireHandlingError	76
157	getJointsSoftLimitRange.....	76
158	setJointsSoftLimitRange	76
159	getFunctionOptI4	77
160	setFunctionOptI4.....	78
161	enterRescueMode.....	78
162	leaveRescueMode	78
163	getCartImpedanceCoordinateType.....	79
164	setCartImpedanceCoordinateType	79
165	setJointLockedInCartImpedanceMode	80
166	getJointLockedInCartImpedanceMode	80
167	setThresholdTorque.....	81
168	getThresholdTorque	81
169	setHeartbeatParam	82
170	getHeartbeatParam	82
171	customRobotState	83

172	getCustomRobotState	83
173	getTcpPosByTcpName	84
174	getTcpPoseByWorkPieceName	84
175	getPayLoadByTcpName	84
176	setDefaultToolTcpCoordinate	85
177	setDefaultWorkPieceTcpCoordinate	85
178	getDefaultTcpCoordinate	85
179	getDefaultWorkPieceCoordinate	86
180	setVelocityPercentValue	86
181	switchRescueMode	86
	附件 A: DianaApi 接口错误码	88
	附录 B: 如何确保运动学逆解唯一	93

修订历史

修改内容	修订人	软件版本	修订时间
创建	石国庆	-	
实现接口 forward、inverse、speedJ、speedL、enterForceMode、leaveForceMode、freeDriving、holdBrake、releaseBrake、stop、getRobotState、initSrv、destroySrv、getJointPos、getTcpPos、getJointAngularVel、getTcpForce、getJointForce、createPath、addMoveL、addMoveJ、runPath、destroyPath、moveJToTarget、moveJToPose、moveLToTarget、moveLToPose、getJointCurrent、getJointTorque、setDefaultActiveTcp、setDefaultActiveTcpPose	石国庆	-	2020-09-09
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 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 修改 initSrv 函数，使其兼容支持回调函数作为传参	石国庆	-	2020-09-18
修改文档版式 新增接口 servoJ、servoL、speedJ_ex、speedL_ex、servoJ_ex、servoL_ex、enableCollisionDetection、createComplexPath、addMoveLSegmentByPose、addMoveLSegmentByTarget、addMoveJSegmentByPose、addMoveJSegmentByTarget、	石国庆	-	2020-11-04

addMoveCSegmentByPose、 addMoveCSegmentByTarget、runComplexPath、 destroyComplexPath、saveEnvironment			
新增接口 getDH、getOriginalJointTorque、 getJacobiMatrix 修改函数 getJointTorque 含义	石国庆	-	2020-11-27
新增 resetDH	孟庆婷	-	2020-12-10
1. 新增 setPushPeriod、initDHCali、 getDHCaliResult、setDH、setWrd2BasRT、 setFla2TcpRT、enableTorqueReceiver、 sendTorque_rt、dumpToUDisk、dumpToUDiskEx、 enterForceMode_ex	石国庆	-	2020-12-17
调整 python 的 API 顺序与已有 C 库一致 新增接口 setWayPoint、getWayPoint、 addWayPoint、deleteWayPoint、 getDafaultActiveTcp、getDafaultActiveTcpPose、 getActiveTcpPose、zeroSpaceManualMove、 moveTcp_ex	石国庆	-	2021-06-01
支持同时控制多台机械臂 调整 API 顺序与 C 语言一致	石国庆	-	2021-07-15
限制 leaveForceMode 输入参数	石国庆	-	2021-07-20
调整 API 顺序与 C 语言一致 补齐 API	石国庆	-	2021-09-17
文档中关节个数统一用 JOINT_NUM 表示，对于 Diana 机器人 JOINT_NUM=7，对于 Thor 机器人 JOINT_NUM=6	邹艳艳	-	2021-12-03
新增 readDO 函数	杨冬	-	2021-12-11
添加 18 个接口说明： getMechanicalJointsPositionRange、 getMechanicalMaxJointsVel、 getMechanicalMaxJointsAcc、 getMechanicalMaxCartVelAcc、 getJointsPositionRange、getMaxJointsVel、	孟庆婷	v2.7	2022-3-24

getMaxJointsAcc、getMaxCartTranslationVel、 getMaxCartRotationVel、getMaxCartTranslationAcc、 getMaxCartRotationAcc、setJointsPositionRange、 setMaxJointsVel、setMaxJointsAcc、 setMaxCartTranslationVel、setMaxCartRotationVel、 setMaxCartTranslationAcc、setMaxCartRotationAcc			
更新 freeDriving 函数传参 添加 getJointsSoftPositionRange setJointsSoftPositionRange getFunctionOptI4 setFunctionOptI4 enterSafetyIdle leaveSafetyIdle getCartImpedanceCoordinateType setCartImpedanceCoordinateType setJointLockedInCartImpedanceMode getJointLockedInCartImpedanceMode	杨冬	v2.8	2022-04-25
修复文档中的若干错误，添加 API 线程安全对照表，阻抗参数设置函数调整为阻尼比	杜东方	v2.9	2022-07-26
updateForce 函数去掉力方向参数； 去除 updateForce_ex 函数； 修改 setCollisionLevel 函数支持组合检测级别	杨冬	v2.10	2022-08-16
修改部分专业术语	杜东方	V2.10	2022-09-30
添加新接口 getThresholdTorque,setThresholdTorque	王伊	V2.11	2022-12-19
添加新接口： setHeartbeatParam getHeartbeatParam customRobotState getCustomRobotState getTcpPosByTcpName getTcpPoseByWorkPieceName getPayLoadByTcpName setDefaultToolTcpCoordinate setDefaultWorkPieceTcpCoordinate getDefaultTcpCoordinate getDefaultWorkPieceCoordinate	田涛	V2.12	2023-02-08

更新以下接口参数： moveJToTarget moveJToPose moveJ moveL moveLToTarget moveLToPose addMoveLSegmentByTarget addMoveLSegmentByPose addMoveJSegmentByTarget addMoveJSegmentByPose addMoveCSegmentByTarget addMoveCSegmentByPose 添加整形功能参数： zv_shaper_order zv_shaper_frequency zv_shaper_damping_ratio	王伊	V2.12	2023-02-10
开放输入整形功能，修改整形参数 zv_shaper_order 的取值范围，影响函数同上 V3.7	王伊	V2.13	2023-03-15
更新以下接口函数： setVelocityPercentValue	王伊	V2.14	2023-06-12
删除 dumpToUDiskEx，保留 dumpToUDisk，原 dumpToUDiskEx 的实现逻辑移到 dumpToUDisk 内	齐洪	V2.14	2023-09-02
添加接口： switchSafetyHandlingMode 修改接口名： enterSafetyIdle 改为 enterRescueMode leaveSafetyIdle 改为 leaveRescueMode	田涛	V2.16	2023-11-21

概述

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（Nm），电流（A），长度（m），线速度（m/s），线加速度（m/s²），角度（rad），角速度（rad/s），角加速度（rad/s²），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。

兼容性说明

操作系统	支持情况	验证用的 Python 版本
Ubuntu 18.04 x64	推荐☆	Python 3.8.0
Ubuntu 20.04 x64	支持	Python 3.8.0
Ubuntu 22.04 x64	支持	Python 3.8.0
Windows 10 x64	推荐☆	Python 3.8.0
Windows 11 x64	支持	Python 3.8.0

函数说明

1 initSrv

<div>def initSrv(srv_net_st, fnError = None,fnState = None)</div> <div>初始化 API，完成其他功能函数使用前的初始化准备工作。</div> <div>参数：</div> <div>pinfo: srv_net_st 为元组，用于配置本地连接服务器、心跳服务和状态反馈服务的端口号信息及服务器 IP，其中 IP 地址需要传入字符串，端口号如果传 0 则由系统自动分配。</div> <div>fnError: 可选参数，错误处理回调函数。其中 e 为 int 类型的错误码（包含通信错误例如版本不匹配，链路错误例如网络断开，硬件故障例如编码器错误等），可调用 formatError 获取字符串提示信息。fnError 函数会用于多线程中实时反馈，所以尽量不要在函数实现中使用 sleep 函数之类会阻塞线程的操作。</div> <div>fnState: 可选参数，robot state 回调函数。回调函数参数为类 StrRobotStateInfo 的 POINTER，包含以下数据：</div> <div><ul style="list-style-type: none">● 关节角度元组（jointPos）● 关节角速度元组（jointAngularVel）● 关节电流元组（jointCurrent）● 关节扭矩元组（jointTorque）</div>
--

- 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)
- 轴空间外部力元组 (jointForce)

返回值:

True: 成功

False: 失败

调用示例 1(含回调函数):

```
import DianaApi
def errorCallback(e):
    print("error code" + str(e))
def robotStateCallback(stateInfo):
    for i in range(0,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)
DianaApi.initSrv(netInfo, fnError, fnState)
DianaApi.destroySrv()
```

调用示例 2(不含回调函数):

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

2 destroySrv

```
def destroySrv(ipAddress="")
```

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

参数:

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

返回值:

True: 成功

False: 失败

调用示例:

```
netInfo=('192.168.10.75', 0, 0, 0)
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

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: ipAddress = '192.168.10.75' setPushPeriod(10, ipAddress)

4 wait_move

def wait_move()
<p>等待运动指令执行完成后调用 stop 停止机器人，如果机器人不处于运动状态（如错误状态，具体状态参考 getRobotState）机器人也会立即停止，如需对异常状态进行处理，请参考下方异常处理示例自定义实现 user_wait_move()</p> <p>参数: 无</p> <p>返回值: 无</p>
<p>调用示例: #以 7 轴机器人为例 joints = [0,PI/6,0,PI/2,0,-PI/3,0] DianaApi.moveJ(joints,0.2,0.2,ipAddress) DianaApi.wait_move()</p> <p>user_wait_move 示例: #碰撞后自动恢复运行 def user_wait_move(): time.sleep(0.02) while True: state = getRobotState() if state == 6 and isCollision(): time.sleep(1) #此等待时间内外力须撤销，可以根据实际情况调整 cleanErrorInfo() time.sleep(1) resume() time.sleep(0.01) #此等待时间需要大于数据推送周期，可以根据实际情况调整 continue if state != 0: break else: time.sleep(0.001) stop()</p>

5 moveTCP

def moveTCP(d, v, a, ipAddress="")
手动移动指定 IP 地址的机械臂工具中心点（TCP）。该函数会立即返回，停

<p>止运动需要调用 <code>stop</code> 函数。</p> <p>参数：</p> <p>d: 表示移动方向的枚举类型。<code>DianaApi</code> 中存在枚举 <code>tcp_direction_e</code>，以下为枚举值及其含义：</p> <ul style="list-style-type: none">● <code>T_MOVE_X_UP</code> 表示沿 x 轴正向● <code>T_MOVE_X_DOWN</code> 表示沿 x 轴负向● <code>T_MOVE_Y_UP</code> 表示沿 y 轴正向● <code>T_MOVE_Y_DOWN</code> 表示沿 y 轴负向● <code>T_MOVE_Z_UP</code> 表示沿 z 轴正向● <code>T_MOVE_Z_DOWN</code> 表示沿 z 轴负向 <p>v: 速度，单位：m/s</p> <p>a: 加速度，单位：m/s²</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>	<p>调用示例：</p> <pre>import DianaApi import time dtype = DianaApi.tcp_direction_e.T_MOVE_X_UP vel = 0.1 acc = 0.2 ipAddress = '192.168.10.75' DianaApi.moveTCP(dtype, vel, acc, ipAddress) time.sleep(1) DianaApi.stop('192.168.10.75')</pre>
--	---

6 rotationTCP

<p>def rotationTCP(d, v, a, ipAddress="")</p> <p>使指定的 IP 地址的机械臂绕工具中心点变换姿态。该函数会立即返回，停止运动需要调用 <code>stop</code> 函数。</p> <p>参数：</p> <p>d: 表示旋转方向的枚举类型。具体介绍参见 <code>MoveTcp</code> 第一个参数。</p> <p>v: 速度，单位：rad/s。</p> <p>a: 加速度，单位：rad/s²。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>	<p>调用示例：</p> <pre>import DianaApi ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) type = DianaApi.tcp_direction_e.T_MOVE_X_UP</pre>
--	--

```

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

```

7 moveJoint

```
def moveJoint(d, i, v, a, ipAddress="")
```

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

参数：

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

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

i: 关节索引号。

v: 速度，单位：rad/s。

a: 加速度，单位：rad/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
type = DianaApi.joint_direction_e.MOVE_UP
index = 6
vel = 0.8
acc = 0.8
ipAddress = '192.168.10.75'
DianaApi.moveJoint(type, index, vel, acc, ipAddress)
time.sleep(3)
DianaApi.stop(ipAddress)

```

8 moveJToTarget

```
def moveJToTarget(joints, v, a, zv_shaper_order=0, zv_shaper_frequency=0,
zv_shaper_damping_ratio=0, ipAddress="")
```

控制指定 IP 地址的机械臂以 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。

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/s²。

zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功

<p>能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。</p> <p>zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order， zv_shaper_frequency， zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。</p> <p>zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。</p> <p>zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi import time joints= [0]*JOINT_NUM vel = 0.2 acc = 0.4 zv_shaper_order = 0 zv_shaper_frequency = 0 zv_shaper_damping_ratio = 0 ipAddress = '192.168.10.75' DianaApi.moveJToTarget(joints, vel, acc, zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)</pre>

9 moveJToPose

<pre>def moveJToPose(pose, v, a, zv_shaper_order=0, zv_shaper_frequency=0, zv_shaper_damping_ratio=0, ipAddress="")</pre>
<p>控制指定 IP 地址的机械臂以工具中心点位姿为终点的 moveJ。该函数会立即返回，停止运动需要调用 stop 函数。</p> <p>参数:</p> <p>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。</p> <p>v: 速度，单位：rad/s。</p> <p>a: 加速度，单位：rad/s²。</p> <p>zv_shaper_order， zv_shaper_frequency， zv_shaper_damping_ratio 为整形功能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。</p> <p>zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order， zv_shaper_frequency， zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。</p> <p>zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。</p> <p>zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。</p> <p>返回值:</p> <p>True: 成功。</p>

False: 失败。
调用示例: <pre>import DianaApi import time poses = (0.087,0.0,1.0827,0.0,0.0,0.0) vel = 0.2 acc = 0.4 zv_shaper_order = 0 zv_shaper_frequency = 0 zv_shaper_damping_ratio = 0 ipAddress = '192.168.10.75' DianaApi.moveJToPose(poses, vel, acc, zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)</pre>

10 moveJ

<pre>def moveJ (joints, v, a, zv_shaper_order=0, zv_shaper_frequency=0, zv_shaper_damping_ratio=0, ipAddress="")</pre>
<p>同 moveJToTarget</p> <p>参数:</p> <p>joints: 包含 JOINT_NUM 个终点关节角度的元组</p> <p>v: 速度, 单位: rad/s。</p> <p>a: 加速度, 单位: rad/s²。</p> <p>zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功能相关参数, 函数中默认值都为 0, 此时使用机械臂默认配置参数。</p> <p>zv_shaper_order: 阶次, 范围[-1,2], 其中取值-1 时, 关闭整形功能, 当 zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 任意一值不为 0, 则使用函数传入的阶次值。</p> <p>zv_shaper_frequency: 频率, 单位: Hz。支持取值范围[0, 1000], 取值为 0 时, 使用机械臂默认配置参数。</p> <p>zv_shaper_damping_ratio: 阻尼比, 支持取值范围[0,1], 取值为 0 时, 使用机械臂默认配置参数。</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi import time joints= [0]*JOINT_NUM vel = 0.2 acc = 0.4 zv_shaper_order = 0 zv_shaper_frequency = 0 zv_shaper_damping_ratio = 0 ipAddress = '192.168.10.75' DianaApi.moveJ (joints, vel, acc, zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress) time.sleep(1)</pre>

DianaApi.stop(ipAddress)

11 moveL

<pre>def moveL(pose, v, a, zv_shaper_order=0, zv_shaper_frequency=0, zv_shaper_damping_ratio=0, ipAddress="")</pre>
<p>同 moveLToPose</p> <p>参数:</p> <p>pose: 包含 6 个终点位姿的元组, 保存 TCP 坐标 (x, y, z) 和轴角 (rx, ry, rz) 组合数据。</p> <p>v: 速度, 单位: m/s。</p> <p>a: 加速度, 单位: m/s²。</p> <p>zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功能相关参数, 函数中默认值都为 0, 此时使用机械臂默认配置参数。</p> <p>zv_shaper_order: 阶次, 范围[-1,2], 其中取值-1 时, 关闭整形功能, 当 zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 任意一值不为 0, 则使用函数传入的阶次值。</p> <p>zv_shaper_frequency: 频率, 单位: Hz。支持取值范围[0, 1000], 取值为 0 时, 使用机械臂默认配置参数。</p> <p>zv_shaper_damping_ratio: 阻尼比, 支持取值范围[0,1], 取值为 0 时, 使用机械臂默认配置参数。</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi import time poses= (0.087,0.0,1.0827,0.0,0.0,0.0) vel = 0.2 acc = 0.4 zv_shaper_order = 0 zv_shaper_frequency = 0 zv_shaper_damping_ratio = 0 ipAddress = '192.168.10.75' DianaApi.moveL (poses, vel, acc, zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)</pre>

12 moveLToTarget

<pre>def moveLToTarget(joints, v, a, zv_shaper_order=0, zv_shaper_frequency=0, zv_shaper_damping_ratio=0, ipAddress="")</pre>
<p>控制指定 IP 地址的机械臂以七个关节角度为终点的 moveL。该函数会立即返回, 停止运动需要调用 stop 函数。</p> <p>参数:</p> <p>joints: 包含 JOINT_NUM 个终点关节角度的元组。</p> <p>v: 速度, 单位: m/s。</p> <p>a: 加速度, 单位: m/s²。</p> <p>zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功</p>

<p>能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。</p> <p>zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order， zv_shaper_frequency， zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。</p> <p>zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。</p> <p>zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi import time joints = [0.0]*JOINT_NUM vel = 0.2 acc = 0.4 zv_shaper_order = 0 zv_shaper_frequency = 0 zv_shaper_damping_ratio = 0 ipAddress = '192.168.10.75' DianaApi.moveLToTarget(joints, vel, acc, zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)</pre>

13 moveLToPose

<pre>def moveLToPose(pose, v, a, zv_shaper_order=0, zv_shaper_frequency=0, zv_shaper_damping_ratio=0, ipAddress="")</pre>
<p>以工具中心点位姿为终点的 moveL。该函数会立即返回，停止运动需要调用 stop 函数。</p> <p>参数:</p> <p>pose: 包含 6 个终点位姿的元组，保存 TCP 坐标 (x, y, z) 和轴角 (rx, ry, rz) 组合数据。</p> <p>v: 速度，单位：m/s。</p> <p>a: 加速度，单位：m/s²。</p> <p>zv_shaper_order， zv_shaper_frequency， zv_shaper_damping_ratio 为整形功能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。</p> <p>zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order， zv_shaper_frequency， zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。</p> <p>zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。</p> <p>zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p>

返回值: True: 成功。 False: 失败。
调用示例: <pre>import DianaApi import time poses= (0.087,0.0,1.0827,0.0,0.0,0.0) vel = 0.2 acc = 0.4 zv_shaper_order = 0 zv_shaper_frequency = 0 zv_shaper_damping_ratio = 0 ipAddress = '192.168.10.75' DianaApi.moveToPose(poses, vel, acc, zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)</pre>

14 speedJ

def speedJ(speed, a, t, ipAddress="")
<p>速度模式，关节空间运动。时间 t 为非零时，函数将在 t 时间后返回不管目标速度是否已经达到。如果 t 为 0，函数将在达到目标速度时返回。停止运动需要调用 <code>stop</code> 函数。</p> <p>该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。</p> <p>参数:</p> <p>speed: 包含 JOINT_NUM 个轴关节角速度的元组。单位: rad/s。</p> <p>a: 加速度，单位: rad/s^2。</p> <p>t: 时间，单位: s。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
调用示例: <pre>import DianaApi import time speeds= [0.0]*JOINT_NUM speeds[1]=0.2 acc = 0.4 ipAddress = '192.168.10.75' DianaApi.speedJ(speeds, acc, 0, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)</pre>

15 speedL

def speedL(speed, a, t, ipAddress="")
<p>控制指定 IP 地址的机械臂进入速度模式，笛卡尔空间下直线运动。时间 t 为非零时，函数将在 t 时间后返回不管目标速度是否已经达到。如果 t 为 0，函数将在达到目标速度时返回。停止运动需要调用 <code>stop</code> 函数。</p> <p>该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住</p>

<p>程序，如果使用了传送带功能请注意规避。</p> <p>参数： speed: 工具空间速度元组，长度为 6,其中前 3 个单位为 m/s，后 3 个单位为 rad/s。 a: 加速度的元组，长度为 2，单位：m/s²，rad/s²。 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。</p> <p>返回值： True: 成功。 False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi import time speeds = (0.5,0.0,0.0,0.0,0.0,0.0) acc = (0.1, 0.5) ipAddress = '192.168.10.75' DianaApi.speedL(speeds, acc, 0, ipAddress) time.sleep(1) DianaApi.stop(ipAddress)</pre>

16 freeDriving

<p>def freeDriving(mode, ipAddress="")</p> <p>实现控制指定 IP 地址的机械臂正常模式与零力驱动模式之间的切换。</p> <p>参数： mode: int 变量，描述零力驱动工作模式，取值为 freedriving_mode_e，E_DISABLE_FREEDRIVING(0)表明退出零力驱动，E_NORMAL_FREEDRIVING(1)为进入正常零力驱动模式。E_FORCE_FREEDRIVING(2)为进入安全零力驱动模式。安全零力驱动模式进入之前必须先利用 enterRescueMode 进入安全处理，再使用 switchRescueMode(12)激活安全零力功能，见示例。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值： True: 成功。 False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi import time ipAddress = '192.168.10.75' DianaApi.freeDriving(DianaApi. freedriving_mode_e. E_NORMAL_FREEDRIVING, ipAddress) time.sleep(10) DianaApi.freeDriving(DianaApi. freedriving_mode_e. E_DISABLE_FREEDRIVING, ipAddress)</pre>

17 stop

<p>def stop(ipAddress="")</p> <p>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。</p> <p>参数： 无。</p> <p>返回值：</p>
--

True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' DianaApi.stop(ipAddress)

18 forward

def forward(joints, pose, ipAddress="") 正解函数，针对指定 IP 地址机器人，由传入的关节角都计算出的正解 TCP 位姿。 参数: joints: 传入参数，JOINT_NUM 个轴关节角度的元组。单位: rad。 pose: 输入输出参数，位姿列表，长度为 6。数据为包含默认的工具坐标系坐标 (x, y, z) 和旋转矢量 (轴角坐标) 组合。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' netInfo=(ipAddress, 0, 0, 0) DianaApi.initSrv(netInfo) DianaApi.releaseBrake(ipAddress) tool1 = (0, 0, 0.1, 0, 0, 0) tcpTestJointPosition = [0.0]*JOINT_NUM tcp1Position = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0] DianaApi.setDefaultActiveTcpPose(tool1, ipAddress) DianaApi.forward(tcpTestJointPosition, tcp1Position, ipAddress) print(tcp1Position) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv()

19 inverse

def inverse(pose, joints, ipAddress="") 逆解函数，针对指定 IP 地址机器人，通过 TCP 位姿计算出最佳逆解关节角度。 参数: pose: 输入参数，位姿元组，长度为 6，数据为包含 active_tcp 坐标 (x, y, z) 和旋转矢量 (轴角坐标) 组合。 joints: 输入输出参数，关节角度的列表，长度为 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。 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, 0.0)

```
joints= [0.0]*JOINT_NUM
ipAddress = '192.168.10.75'
DianaApi.inverse(pose, joints, ipAddress)
print(joints)
```

20 getJointPos

def getJointPos(joints, ipAddress='')
<p>获取指定 IP 地址机械臂各个关节角度的位置，库初始化后，后台会自动同步机器人状态信息，因此所有的监测函数都是从本地缓存取数。</p> <p>参数：</p> <p>joints: 输入输出参数，关节角的列表，元组大小为 JOINT_NUM。用于传递获取到的结果。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi joints =[0.0]*JOINT_NUM ipAddress='192.168.10.75' DianaApi.getJointPos(joints, ipAddress) print(joints)</pre>

21 getJointAngularVel

def getJointAngularVel(vels, ipAddress='')
<p>获取指定 IP 地址机械臂当前各关节的角速度。</p> <p>参数：</p> <p>vels: 输入输出参数，传入空的列表，输出关节角速度，长度为 JOINT_NUM。用于传递获取到的结果。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi jointAngularVel= [0.0]*JOINT_NUM ipAddress='192.168.10.75' DianaApi.getJointAngularVel(jointAngularVel, ipAddress) print(jointAngularVel)</pre>

22 getJointCurrent

def getJointCurrent(joints, ipAddress='')
<p>获取当前关节电流。</p> <p>参数：</p> <p>joints: 输入输出参数，传入空的列表，输出关节电流，长度为 JOINT_NUM。用于传递转获取到的结果。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p>

False: 失败。
调用示例: import DianaApi jointsCurrent = [0.0] * JOINT_NUM ipAddress='192.168.10.75' DianaApi.getJointCurrent(jointsCurrent, ipAddress) print(jointsCurrent)

23 getJointTorque

def getJointTorque(torques, ipAddress="")
获取指定 IP 地址机械臂各关节真实扭矩数据，即减去零偏的扭矩值。 参数: torques: 输入输出参数，传入空的列表，输出真实的关节扭矩，长度为 JOINT_NUM。用于传递获取到的结果。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi jointTorques = [0.0]*JOINT_NUM ipAddress = '192.168.10.75' DianaApi.getJointTorque(jointTorques , ipAddress) print(jointTorques)

24 getTcpPos

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

25 getTcpExternalForce

def getTcpExternalForce(ipAddress="")
获取指定 IP 地址机械臂 TCP 实际感受到的力合大小，TCP 位姿可被 setDefaultActiveTcp 函数改变。 参数:

<p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值: 返回力的大小。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' force = DianaApi.getTcpExternalForce(ipAddress) print(force)</pre>

26 releaseBrake

<p>def releaseBrake(ipAddress="")</p> <p>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。</p> <p>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。</p> <p>返回值: True: 成功。 False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' DianaApi.releaseBrake(ipAddress) DianaApi.holdBrake(ipAddress)</pre>

27 holdBrake

<p>def holdBrake(ipAddress="")</p> <p>关闭指定 IP 地址机械臂的抱闸，停止机械臂。</p> <p>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。</p> <p>返回值: True: 成功。 False: 失败。</p>
<p>调用示例: 见 releaseBrake()用例</p>

28 changeControlMode

<p>def changeControlMode(m, ipAddress="")</p> <p>控制指定 IP 地址机械臂的模式切换。</p> <p>参数:</p> <p>m: 枚举类型 mode_e。枚举及其含义如下</p> <ul style="list-style-type: none"> ● T_MODE_INVALID 无意义 ● T_MODE_POSITION 位置模式 ● T_MODE_JOINT_IMPEDANCE 关节空间阻抗模式 ● T_MODE_CART_IMPEDANCE 笛卡尔空间阻抗模式 <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接</p>
--

<p>一台机械臂时生效。</p> <p>返回值：</p> <p>True：成功。</p> <p>False：失败。</p>
<p>调用示例：</p> <pre>import DianaApi ipAddress = '192.168.10.75' DianaApi.changeControlMode(DianaApi.mode_e.T_MODE_POSITION, ipAddress)</pre>

29 getLibraryVersion

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

30 formatError

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

31 getLastError

<p>def getLastError(ipAddress="")</p> <p>返回指定 IP 地址机械臂最近发生的错误码。该错误码会一直保存，确保可以查询得到，直至库卸载，因此，当库函数调用失败后，如果想知道具体的错误原因，应该调用该函数获取错误码。</p> <p>参数：</p> <p>ipAddress:可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p>

返回值: 0: 没有错误。 其它值: 具体错误码。
调用示例: <pre>import DianaApi ipAddress = '192.168.10.75' e = DianaApi.getLastError(ipAddress) print(e)</pre>

32 setLastError

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

33 setDefaultActiveTcp

def setDefaultActiveTcp(default_tcp, ipAddress="") 设置指定 IP 地址字符串的默认工具坐标系。在没有调用该函数时，默认工具中心点为法兰盘中心，调用该函数后，默认的工具坐标系将被改变。该函数将会改变 moveTCP, rotationTCP, moveJToPos, moveLToPose, speedJ, speedL, forward, inverse, getTcpPos, getTcpExternalForce 的默认行为。 参数: default_tcp: 输入参数，TCP 相对于末端法兰盘的 4*4 齐次变换矩阵的元组，元组大小为 16。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功 False: 失败
调用示例: <pre>import DianaApi matrix = (1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1) DianaApi.setDefaultActiveTcp (matrix, '192.168.10.75')</pre>

34 getLinkState

def getLinkState(ipAddress="") 获取与指定 IP 地址机械臂间的链路状态。
参数: 无。
返回值:

True: 链路正常。 False: 链路断开。
调用示例: <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.getLinkState(ipAddress) if ret == True: print("链路正常") else: print("链路断开")</pre>

35 getTcpForce

def getTcpForce(forces, ipAddress="") 获取指定 IP 地址机械臂的 TCP 所受外部六维力，TCP 位姿可被 setDefaultActiveTcp 函数改变。 参数: forces: 输入输出参数，传入空列表，输出工具中心点处六维力，长度为 6。用于传递获取到的结果。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: <pre>import DianaApi ipAddress = '192.168.10.75' tcpForce = [0.0] * 6 DianaApi.getTcpForce(tcpForce, ipAddress) print('tcpForce={%f, %f, %f, %f, %f, %f}'%(tcpForce [0], tcpForce [1], \ tcpForce [2], tcpForce [3], tcpForce [4], tcpForce [5]))</pre>

36 getJointForce

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

37 isCollision

def isCollision(ipAddress="")

<p>从轴空间判断指定 IP 地址机械臂是否发生碰撞。</p> <p>参数:</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 机器人发生碰撞。</p> <p>False: 机器人未发生碰撞。</p>
<p>调用示例:</p> <pre>import DianApi ipAddress = '192.168.10.75' ret = DianaApi.isCollision(ipAddress) if ret == True: print("collisioned") else: print("not collisioned")</pre>

38 initDHCali

<p>def initDHCali(tcpMeas, jntPosMeas, nrSets, ipAddress="")</p> <p>根据输入的关节角以及 TCP 位置元组计算指定 IP 地址机械臂的 DH 参数。</p> <p>参数:</p> <p>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。</p> <p>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, 共 nrSets 组。单位: rad。</p> <p>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。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi #以下参数仅供 API 展示, 无实际含义 rowNo_refData = 32 tcpMeas = [0.0] * 96 jntMeas = [0.0] * 32*JOINT_NUM ipAddress = '192.168.10.75' DianaApi.initDHCali(tcpMeas, jntMeas, rowNo_refData, ipAddress)</pre>

39 getDHCaliResult

<p>def getDHCaliResult(rDH, wRT, tRT, confid, ipAddress="")</p> <p>获取指定 IP 地址机械臂 DH 参数的计算结果。</p> <p>参数:</p> <p>rDH: 输出参数。机器人各关节 DH 参数的元组, 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。</p> <p>wRT: 输出参数。机器人基坐标系相对于世界坐标系下的位姿的元组, 元组大小为 6。位姿数据[x, y, z, Rx, Ry, Rz]。单位: rad、m。</p> <p>tRT: 输出参数。靶球在法兰坐标系下的位置描述的元组, 元组大小为 3。元组为靶球位置坐标[x,y,z]。单位: m。</p> <p>confid: 输出参数。绝对定位精度参考值的元组, 元组大小为 2。其中, 第</p>
--

<p>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。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>该 API 接口一般不能直接调用，需要配合 initDHCali 接口使用，直接调用会失败。</p> <p>返回值:</p> <p>0: 获取成功。</p> <p>1: 获取结果精度可能够较低。</p> <p>-1: 获取失败，发生异常。</p>
<p>调用示例:</p> <pre>import DianaApi #以下参数仅供 API 展示，无实际含义 rDH = [0.0] * 4*JOINT_NUM wRT = [0.0] * 6 tRT = [0.0] * 3 confid = [0.0] * 2 ipAddress = '192.168.10.75' DianaApi.getDHCaliResult(rDH, wRT, tRT, confid, ipAddress)</pre>

40 setDH

<pre>def setDH(a, alpha, d, theta, ipAddress="")</pre> <p>设置指定 IP 地址机械臂当前 DH 参数。特别注意，错误的参数设置可能引起机器人损坏，需谨慎设置！</p> <p>参数:</p> <p>a: 输入参数。各关节的 a 参数的元组，元组大小为 JOINT_NUM。</p> <p>alpha: 输入参数。各关节的 alpha 参数的元组，元组大小为 JOINT_NUM。</p> <p>d: 输入参数。各关节的 d 参数的元组，元组大小为 JOINT_NUM。</p> <p>theta: 输入参数。各关节的 theta 参数的元组，元组大小为 JOINT_NUM。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi #以下参数仅供 API 展示，无实际含义 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.setDH(a, alpha, d, theta, ipAddress)</pre>

41 setWrd2BasRT

<pre>def setWrd2BasRT(RTw2b, ipAddress="")</pre> <p>初始化世界坐标系到指定 IP 地址机械臂坐标系的平移和旋转位姿。用于 DH 参数标定前设置，若用户不能提供此参数，DH 参数标定功能依旧可以使用。如果调用此函数则使用用户自定义的位姿。特别注意，此功能每次移动机器人与激光跟踪仪都需要重新计算，使用错误的参数可能引起 DH 参数计算不准确或标定异常。</p>

<p>参数: RTw2b: 输入参数。世界坐标系到机器人坐标系的平移和旋转位姿的元组, 元组大小为 6。单位: 米和弧度。 ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值: True: 成功。 False: 失败。</p>
<p>调用示例: import DianaApi #以下参数仅供 API 展示, 无实际含义 wRT = [0.0] * 6 ipAddress = '192.168.10.75' DianaApi.setWrd2BasRT(wRT, ipAddress)</p>

42 setFla2TcpRT

<p>def setFla2TcpRT(RTf2t, ipAddress="")</p> <p>初始化指定 IP 地址机械臂法兰坐标系到工具坐标系的平移位置。用于 DH 参数标定前设置, 若用户不能提供此参数, DH 参数标定功能依旧可以使用。如果调用此函数则使用用户自定义的位姿。特别注意, 此功能每次移动机器人与激光跟踪仪都需要重新计算, 使用错误的参数可能引起 DH 参数计算不准确或标定异常。</p> <p>参数: RTf2t: 输入参数。初始化法兰坐标系到工具坐标系的平移位置元组, 元组大小为 3, 位置信息数据[x,y,z]。单位: 米。 ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值: True: 成功。 False: 失败。</p>
<p>调用示例: import DianaApi #以下参数仅供 API 展示, 无实际含义 Fla = [0.0] * 3 ipAddress = '192.168.10.75' DianaApi.setFla2TcpRT(Fla, ipAddress)</p>

43 getRobotState

<p>def getRobotState(ipAddress="")</p> <p>获取指定 IP 地址机械臂当前工作状态。</p> <p>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。</p> <p>返回值: 0: running (机械臂正在运动或者示教器程序正在运行) 1: paused (暂停正在运动的机械臂或者暂停正在运行的示教器程序) 2: idle (空闲状态) 3: free-driving (进入正常零力驱动模式) 4: zero-space-free-driving (进入零空间零力驱动模式) 5: hold-brake (机械臂关闭抱闸)</p>

6: error (机械臂发生错误) 7: handling-error (机械臂的关节超出极限位置) 8: rescuemode (安全处理模式)
调用示例: <pre> import DianaApi ipAddress = '192.168.10.75' state = DianaApi.getRobotState(ipAddress) if state == 0: print("\t[robot state]:running\n") elif state == 1: print("\t[robot state]:paused\n") elif state == 2: print("\t[robot state]:idle\n") elif state == 3: print("\t[robot state]: free-driving \n") elif state == 4: print("\t[robot state]: zero-space-free-driving \n") 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]: handling-error \n") elif state == 8: print("\t[robot state]: rescuemode \n") else: print("\t[robot state]: unknown state \n") </pre>

44 resume

def resume(ipAddress="") 当指定 IP 地址机械臂发生碰撞或其他原因暂停后，恢复运行时使用。 参数: ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: <pre> import DianaApi import time target =[0,0,0,3.141592653/2,0,0,0] vel = 0.2 acc = 0.2 ipAddress = '192.168.10.75' DianaApi.moveJToTarget(target,vel,acc, ipAddress) while True: state = DianaApi.getRobotState(ipAddress) if state == 0: time.sleep(0.01) elif state == 1 and DianaApi.isCollision(ipAddress): DianaApi.resume(ipAddress) time.sleep(1) </pre>


```

else:
    break
DianaApi.stop(ipAddress)

```

45 setJointCollision

def setJointCollision(collision,ipAddress="")
<p>设置指定 IP 地址机械臂关节空间碰撞检测的力矩阈值。</p> <p>参数：</p> <p>collision: 输入参数。JOINT_NUM 个关节轴力矩阈值元组，元组大小为 JOINT_NUM，单位 Nm</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 设置成功。</p> <p>False: 设置失败。</p>
<p>调用示例：</p> <pre> import DianaApi ipAddress = '192.168.10.75' collision = (200, 200, 200, 200, 200, 200, 200)#以 7 轴机器人为例 DianaApi.setJointCollision(collision, ipAddress) </pre>

46 setCartCollision

def setCartCollision(collision,ipAddress="")
<p>设置指定 IP 地址机械臂笛卡尔空间碰撞检测的六维力阈值。</p> <p>参数：</p> <p>collision: 输入参数。在基坐标系下的六维力元组，长度为 6，前三维单位 N，后三维单位 N·M</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre> collision = (200, 200, 200, 200, 200, 200) ipAddress = '192.168.10.75' DianaApi.setCartCollision (collision, ipAddress) </pre>

47 enterForceMode

def enterForceMode(frame_type, frame_matrix, force_direction, force_value, max_approach_velocity, max_allow_tcp_offset,ipAddress="")
<p>使指定 IP 地址机械臂进入力控模式。</p> <p>参数：</p> <p>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）。</p> <p>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。</p> <p>force_direction: 力指令的方向的元组，大小为 3。</p> <p>force_value: 力指令的大小，长度为 3 的元组。单位：N。</p> <p>max_approach_velocity: 最大接近速度。单位：m/s。</p> <p>max_allow_tcp_offset: 允许的最大偏移。单位：m。</p>

<p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功</p> <p>False: 失败</p>
<p>调用示例:</p> <pre>import DianaApi frame_type = 1 frame_matrix = (1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1) force_direction =(0,0,-1) force_value = 2.0 max_approach_velocity = 0.1 max_allow_tcp_offset = 0.2 ipAddress = '192.168.10.75' DianaApi.enterForceMode(frame_type, frame_matrix, force_direction, force_value, max_approach_velocity, max_allow_tcp_offset,ipAddress)</pre>

48 leaveForceMode

<p>def leaveForceMode (mode,ipAddress="")</p> <p>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。</p> <p>参数:</p> <p>mode: 控制模式。为 mode_e 类型的枚举。枚举声明及其含义如下:</p> <ul style="list-style-type: none">● T_MODE_INVALID: 代表无效模式● T_MODE_POSITION: 代表位置模式● T_MODE_JOINT_IMPEDANCE: 代表关节空间阻抗模式● T_MODE_CART_IMPEDANCE: 代表笛卡尔空间阻抗模式 <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' DianaApi.leaveForceMode(DianaApi.mode_e.T_MODE_POSITION, ipAddress)</pre>

49 setDefaultActiveTcpPose

<p>def setDefaultActiveTcpPose (arrPose,ipAddress="")</p> <p>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。</p> <p>参数:</p> <p>arrPose: 输入参数。TCP 相对于末端法兰盘的位姿向量的元组, 元组大小为 6</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接</p>

一台机械臂时生效。 返回值： True：成功。 False：失败。
调用示例： <pre>import DianaApi pose = (0.1,0.1,0.1,0, 0, 0) ipAddress = '192.168.10.75' DianaApi.setDefaultActiveTcpPose (pose,ipAddress)</pre>

50 setResultantCollision

def setResultantCollision (force,ipAddress="") 设置指定 IP 地址机械臂笛卡尔空间碰撞检测 TCP 的合力阈值。
参数： force：合力值。 ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值： True：成功。 False：失败。
调用示例： <pre>import DianaApi force = 8.9 ipAddress = '192.168.10.75' DianaApi.setResultantCollision(force, ipAddress)</pre>

51 setJointImpeda

def setJointImpeda (arrStiff, dblDamp, ipAddress="") 设置指定 IP 地址机械臂各关节阻抗参数，包含刚度 Stiffness 和阻尼比 DampingRatio 的数据。
参数： arrStiff：表示各关节刚度 Stiffness 的元组，长度为 JOINT_NUM。 dblDamp：表示关节阻尼比 DampingRatio。 ipAddress：可选参数，需要控制机械臂的 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： <pre>import DianaApi arrStiff = (3000, 3000, 3000, 1000, 500, 1000, 1000) dblDamp= 0 ipAddress = '192.168.10.75' DianaApi.setJointImpeda (arrStiff, dblDamp, ipAddress)</pre>

52 getJointImpeda

def getJointImpeda (arrStiff, dblDamp,ipAddress="") 获取指定 IP 地址机械臂各关节阻抗参数，包含刚度 Stiffness 和阻尼比 DampingRatio 的数据。
参数：

<p>arrStiff: 表示各关节刚度 Stiffness 的列表，长度为 JOINT_NUM，用于接收获取到的值。</p> <p>dblDamp: 表示关节阻尼比 DampingRatio。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi arrStiff = [0] * JOINT_NUM dblDamp = [0] * JOINT_NUM ipAddress = '192.168.10.75' DianaApi.getJointImpeda (arrStiff, dblDamp, ipAddress) print('Stiff: %f, %f, %f, %f, %f, %f, %f\nDamp: %f \n' %(arrStiff[0], arrStiff[1], arrStiff[2], arrStiff[3], arrStiff[4], arrStiff[5], arrStiff[6],\ dblDamp))</pre>

53 setCartImpeda

<p>def setCartImpeda (arrStiff, dblDamp, ipAddress="")</p> <p>设置指定 IP 地址机械臂笛卡尔空间阻抗参数。</p> <p>参数:</p> <p>arrStiff: 表示笛卡尔空间，各维度刚度 Stiffness 的元组，长度为 6。</p> <p>dblDamp: 表示笛卡尔空间阻尼比 DampingRatio。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi arrStiff = (1000, 1000, 1000, 500, 500, 500) dblDamp = 0 ipAddress = '192.168.10.75' DianaApi.setCartImpeda(arrStiff, dblDamp, ipAddress)</pre>

54 getCartImpeda

<p>def getCartImpeda (arrStiff, dblDamp, ipAddress="")</p> <p>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 Stiffness 和阻尼比 DampingRatio 的数据。</p> <p>参数:</p> <p>arrStiff: 表示笛卡尔空间，各维度刚度 Stiffness 的列表，长度为 6，用于接收获取到的值。</p> <p>dblDamp: 表示笛卡尔空间阻尼比 DampingRatio。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi arrStiff = [0] * 6</pre>

```

dblDamp = [0] * 6
ipAddress = '192.168.10.75'
DianaApi.getCartImpedance (arrStiff, arrDamp, ipAddress)
print('Stiff: %f, %f, %f, %f, %f, %f\nDamp: %f \n'
      %(arrStiff[0], arrStiff[1], arrStiff[2], arrStiff[3], arrStiff[4], arrStiff[5],\
        dblDamp))

```

55 zeroSpaceFreeDriving

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

56 createPath

<pre>def createPath (type, ipAddress="")</pre>
<p>为指定 IP 地址机械臂创建一个路段。</p> <p>参数:</p> <p>type: 输入参数。1:表示 moveJ, 2: 表示 moveL。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>返回带两个参数的元组</p> <p>参数:</p> <p>True: 成功。</p> <p>False: 失败。</p> <p>参数 1:</p> <p>id_path: 输出参数。用于保存新创建 Path 的 ID。</p>
<p>调用示例:</p> <pre> import DianaApi import time def to_rad(x): return x*math.pi / 180.0 #以 7 轴机器人为例 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)) </pre>

```

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

```

57 addMoveL

<pre>def addMoveL (id_path, joints, vel, acc, blendradius, ipAddress="")</pre>
<p>向指定 IP 地址机械臂已创建的路段添加 MoveL 路点。</p> <p>参数：</p> <p>id_path: 输入参数。要添加路点的路径 ID。</p> <p>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。</p> <p>vel: moveL 移动到目标路点的速度。单位：m/s。</p> <p>acc: moveL 移动到目标路点的加速度。单位：m/s²。</p> <p>blendradius: 交融半径。单位：m。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <p>见 createPath 例子</p>

58 addMoveJ

<pre>def addMoveJ (id_path, joints, vel_percent, acc_percent, blendradius_percent, ipAddress="")</pre>
<p>向指定 IP 地址机械臂已创建的路段添加 MoveJ 路点。</p> <p>参数：</p> <p>id_path: 输入参数。要添加路点的路径 ID。</p> <p>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</p> <p>vel_percent: moveJ 移动到目标路点的速度百分比。</p> <p>acc_percent: moveJ 移动到目标路点的加速度百分比。</p> <p>blendradius_percent: 交融半径百分比。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre> import DianaApi import time def to_rad(x): return x*math.pi / 180.0 #以 7 轴机器人为例 </pre>

```

firstPosition = (to_rad(0), to_rad(20), to_rad(0), to_rad(90), \
to_rad(0), to_rad(120), to_rad(0))
secondPosition = (to_rad(0), to_rad(-20), to_rad(0), to_rad(45), \
to_rad(0), to_rad(-120), to_rad(0))
thirdPosition = (to_rad(0), to_rad(0), to_rad(0), to_rad(0), \
to_rad(0), to_rad(0), to_rad(0))
print('start test moveJ path.')
print('start test moveJ path.')
ipAddress = '192.168.10.75'
path_id=DianaApi.createPath(1, ipAddress)[1]
DianaApi.addMoveJ(path_id, firstPosition, 0.2, 0.2, 0.3, ipAddress)
DianaApi.addMoveJ(path_id, secondPosition, 0.2, 0.2, 0.3, ipAddress)
DianaApi.addMoveJ(path_id, thirdPosition, 0.2, 0.2, 0.3, ipAddress)
DianaApi.runPath(path_id, ipAddress)
DianaApi.destroyPath(path_id, ipAddress)
time.sleep(10)

```

59 runPath

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

60 destroyPath

def destroyPath (id_path, ipAddress="")
<p>销毁指定 IP 地址机械臂某个路段。</p> <p>参数：</p> <p>id_path: 输入参数。要销毁的路径 ID。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <p>详见 addMoveJ 或 addMoveL 调用示例。</p>

61 rpy2Axis

def rpy2Axis (arr)
<p>rpy 角转轴角。</p> <p>参数：</p> <p>arr: 输入输出参数。rpy 角的列表，长度为 3</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>

```
调用示例：
arr = [0.5,0.6,0.7]
DianaApi.rpy2Axis(arr)
print(arr)
DianaApi.axis2RPY(arr)
print(arr)
```

62 axis2RPY

def axis2RPY (arr)
轴角转 rpy 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

63 homogeneous2Pose

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

64 pose2Homogeneous

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

65 enableTorqueReceiver

def enableTorqueReceiver(bEnable, ipAddress="")
使指定 IP 地址机械臂开启实时扭矩接收 参数: bEnable: 输入参数。是否开启扭矩实时接收 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress='192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.enableTorqueReceiver(True, ipAddress) DianaApi.destroySrv()

66 sendTorque_rt

def sendTorque_rt(torque,t, ipAddress="")
对指定 IP 地址机械臂，用户发送实时扭矩 参数: torque: 输入参数。用户传入的扭矩值，大小为 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。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress='192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) torque =[0.0]*JOINT_NUM t = 1000 DianaApi.sendTorque_rt(torque,t, ipAddress) DianaApi.destroySrv()

67 enableCollisionDetection

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

```
import DianaApi
ipAddress='192.168.10.75'
DianaApi.enableCollisionDetection (True, ipAddress)
```

68 setActiveTcpPayload

def setActiveTcpPayload(payload,ipAddress = "")
<p>设置指定 IP 地址机械臂的负载信息</p> <p>参数:</p> <p>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</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>ipAddress = '192.168.10.75' payload = [0] * 10 ret = DianaApi.setActiveTcpPayload (payload, ipAddress) if ret == False: print('setActiveTcpPayload failed! Return value = %d' %(ret))</pre>

69 servoJ

def servoJ(joints_pos, t=0.01, ah_t=0.03, gain=300, ipAddress="")
<p>关节空间内，伺服指定 IP 地址机械臂到指定关节角位置。servoJ 函数用于在线控制机器人，ah_t 时间和 gain 能够调整轨迹是否平滑或尖锐。注意：太高的 gain 或太短的 ah_t 时间可能会导致不稳定。由于该函数主要用于以较短位移为目标点的多次频繁调用，建议在实时系统环境下使用。该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。</p> <p>参数:</p> <p>joints_pos: 目标关节角位置元组，大小为 JOINT_NUM</p> <p>t: 运动时间</p> <p>ah_t: 时间 (s)，范围 (0.03-0.2) 用这个参数使轨迹更平滑</p> <p>gain: 目标位置的比例放大器，范围 (100,2000)</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi import time ipAddress='192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake(ipAddress) PI=3.141592653 #以 7 轴机器人为例 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(target, 0.01, 0.1, 300,ipAddress) if ret == False:</pre>

```

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

```

70 servoL

```
def servoL(tcp_pose, t=0.01, ah_t=0.03, gain=300, scale=1, ipAddress='')

```

笛卡尔空间内，伺服指定 IP 地址机械臂到指定位姿。由于该函数主要用于以较短位移为目标点的多次频繁调用，建议在实时系统环境下使用。该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。

参数：

tcp_pose: 目标位姿列表，元组大小为 6。前三个元素单位：m；后三个元素单位：rad，注意，角度需要用轴角表示

t: 运动时间。单位：s。

time: 运动时间。单位：s。

ah_t: 时间（s），范围（0.03-0.2）用这个参数使轨迹更平滑。

gain: 目标位置的比例放大器，范围（100,2000）。

scale: 平滑比例系数。范围（0.0~1.0）。

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

返回值：

True: 成功。

False: 失败。

调用示例：

```

import DianaApi
import time
ipAddress='192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
pi=3.141592653
target = [0.319912,0,0.867999,0,pi/4,0]
for i in range(10):
    target[3] = target[3]+0.005
    ret = DianaApi.servoL(target, ipAddress)
    if ret < 0:
        break
    time.sleep(0.1)
DianaApi.stop(ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv(ipAddress)

```

71 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器人，**ah_t** 时间和 **gain** 能够调整轨迹是否平滑或尖锐。注意：太高的 **gain** 或太短的 **ah_t** 时间可能会导致不稳定。由于该函数主要用于以较短位移为目标点的多次频繁调用，建议在实时系统环境下使用。

<p>该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。</p> <p>参数：</p> <p>joints: 目标关节角列表，长度为 JOINT_NUM。单位：rad。</p> <p>t: 运动时间。单位：s。</p> <p>ah_t: 时间，范围（0.03-0.2）用这个参数使轨迹更平滑。单位：s。</p> <p>gain: 目标位置的比例放大器，范围(100,2000)。</p> <p>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。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>	<p>调用示例：</p> <pre>import DianaApi import time ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake(ipAddress) PI=3.141592653 #以 7 轴机器人为例 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 ret == False: break time.sleep(0.1) DianaApi.stop(ipAddress) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv()</pre>
---	---

72 servoL_ex

<pre>def servoL_ex(tcp_pose, t=0.01, ah_t=0.03, gain=300, scale = 1, reliable = False, ipAddress="")</pre>	<p>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。</p> <p>该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。</p> <p>参数：</p> <p>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</p> <p>t: 运动时间。单位：s。</p> <p>ah_t: 范围（0.03-0.2）用这个参数使轨迹更平滑。单位：s。</p> <p>gain: 目标位置的比例放大器，范围（100,2000）。</p> <p>scale: 平滑比例系数。范围（0.0~1.0）。</p> <p>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。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接</p>
--	--

<p>一台机械臂时生效。</p> <p>返回值： True: 成功。 False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi import time ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake(ipAddress) pi=3.141592653 target = [0.319912,0,0.867999,0,pi/4,0] for i in range(10): target[3] = target[3]+0.005 ret = DianaApi.servoL_ex(target,ipAddress) if ret == False: break time.sleep(0.1) DianaApi.stop(ipAddress) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv(ipAddress)</pre>

73 speedJ_ex

<p>def speedJ_ex(speed, acc, t=0.0, realiable=False, ipAddress=")</p> <p>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 为非零时，机器人将在 t 时间后减速。如果 t 为 0，机器人将在达到目标速度时减速。该函数调用后立即返回。停止运动需要调用 stop 函数。该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。</p> <p>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/s²。 t: 时间，单位：s。 realiable: bool 型变量，值为 True 需要 socket 反馈通信状态，行为等同 speedJ；值为 False 则无需反馈直接返回。</p> <p>返回值： True: 成功。 False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi import time DianaApi.initSrv(('192.168.10.75',0,0,0)) DianaApi.releaseBrake() speeds = [0.0]*JOINT_NUM speeds[1]=0.2 acc = 0.40 ret = DianaApi.speedJ_ex(speeds, acc, 0, True) if ret == False: print('speedJ_ex failed! Return value =' + str(ret)) time.sleep(2) DianaApi.holdBrake() DianaApi.destroySrv()</pre>

74 speedL_ex

def speedL_ex(speed, acc, 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 为非零 shi ，机器人将在 t 时间后减速。如果 t 为 0，机器人将在达到目标速度时减速。该函数调用后立即返回。停止运动需要调用 stop 函数。 该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。 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: 加速度的元组，长度为 2，单位：m/s ² ，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 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： <pre>import DianaApi import time ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake(ipAddress) speeds = [0.1,0.0,0.0,0.0,0.0,0.0] acc = [0.30, 0.50] DianaApi.speedL_ex(speeds, acc, 0, True, ipAddress) time.sleep(2) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv()</pre>

75dumpToUDisk

def dumpToUDisk(time_out, ipAddress=") 导出指定 IP 地址机械臂的日志文件到 u 盘。控制箱中的系统日志文件（主要包含 ControllerLog.txt 和 DianaServerLog.txt）会自动复制到 u 盘。需要注意的是目前控制箱仅支持 FAT32 格式 u 盘，调用 dumpToUDisk 函数前需先插好 u 盘，如果系统日志拷贝失败将不会提示。 参数： time_out: 单位 秒，设置超时时间，一般需要大于 3 秒,-1 表示设置超时时间无穷大。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值： True: 成功。 False: 失败。
调用示例： <ol style="list-style-type: none"> 1. 系统开机 2. 插入 u 盘到控制箱

3. 调用 Api 函数 dumpToUDisk(-1,'192.168.10.75')
4. 拔下 u 盘查看

76 inverse_ext

<pre>def inverse_ext(ref_joints, pose, joints, ipAddress = "")</pre>
<p>针对指定 IP 地址机器人，逆解函数，给定一个参考关节角，算出欧式距离最近的逆解。</p> <p>参数：</p> <p>ref_joints: 参考的关节角，大小为 JOINT_NUM 的列表</p> <p>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。</p> <p>joints: 输出参数，关节角度列表。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi ref_joints = [0.0] * JOINT_NUM pose = [0.4221, 0.0, 0.9403, 0.0, 0.0, 0.0] joints = [0.0] * JOINT_NUM ipAddress = "192.168.10.75" ret = DianaApi.inverse_ext(ref_joints, pose, joints, ipAddress) if ret < 0: print("inverse_ext failed! Return value = %d\n" %(ret))</pre>

77 getJointLinkPos

<pre>def getJointLinkPos(joints, ipAddress="")</pre>
<p>获取指定 IP 地址机械臂当前低速侧关节角</p> <p>参数：</p> <p>joints: 输出参数。低速侧关节角,大小为 JOINT_NUM 列表</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi ipAddress='192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake(ipAddress) joints = [0.0]*JOINT_NUM DianaApi.getJointLinkPos(joints, ipAddress) print(joints) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv()</pre>

78 createComplexPath

<pre>def createComplexPath (path_type, ipAddress="")</pre>
--

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

参数：

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

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

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

返回值：

返回带两个参数的元组

参数 0：

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：(MoveP 直线运动)

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

79 addMoveLSegmentByTarget

```
def addMoveLSegmentByTarget(complex_path_id, joints, vel, acc, blendradius,
zv_shaper_order=0, zv_shaper_frequency=0, zv_shaper_damping_ratio=0,
ipAddress="")
```

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

参数:

id_path: 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 `MOVE_L` 类型 id, 而传入 `MOVE_R_JOINT_PATH` 枚举生成 `MOVE_R` 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

acc: 移动到目标路点的加速度

blendradius: 交融半径

zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功能相关参数, 函数中默认值都为 0, 此时使用机械臂默认配置参数。

zv_shaper_order: 阶次, 范围[-1,2], 其中取值-1 时, 关闭整形功能, 当 **zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio** 任意一值不为 0, 则使用函数传入的阶次值。

zv_shaper_frequency: 频率, 单位: Hz。支持取值范围[0, 1000], 取值为 0 时, 使用机械臂默认配置参数。

zv_shaper_damping_ratio: 阻尼比, 支持取值范围[0,1], 取值为 0 时, 使用机械臂默认配置参数。

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
import time
pi = 3.141592653
def to_rad(deg):
    return deg/180*pi
zv_shaper_order = 0
zv_shaper_frequency = 0
zv_shaper_damping_ratio = 0
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
#以 7 轴机器人为例
ret =
DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_JOINT_PATH, ipAddress)
if ret[0] == 0:
    joint1 = [0,to_rad(-17.193),0,to_rad(83.132),0,to_rad(65.939),0]
    joint2 = [0,to_rad(-25.992),0,to_rad(128.898),0,to_rad(-102.906),0]
    DianaApi.addMoveLSegmentByTarget(ret[1],joint1,0.2,0.2,0.1,
zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress)
    DianaApi.addMoveLSegmentByTarget(ret[1],joint2,0.2,0.2,0.1,
zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress)
    DianaApi.runComplexPath(ret[1], ipAddress)
    time.sleep(15)
    DianaApi.destroyComplexPath(ret[1], ipAddress)
DianaApi.holdBrake(ipAddress)
```

DianaApi.destroySrv()

80 addMoveLSegmentByPose

```
def addMoveLSegmentByPose(complex_path_id, pose, vel, acc, blendradius,
zv_shaper_order=0,zv_shaper_frequency=0,zv_shaper_damping_ratio=0,
ipAddress="")
```

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

参数：

id_path：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 MOVE_L 类型 id，而传入 MOVE_P_POSE_PATH 枚举生成 MOVE_P 类型 id

pose：输入参数。要添加的路点，为该路点的 TCP 位姿列表，长度为 6

vel：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

zv_shaper_order，**zv_shaper_frequency**，**zv_shaper_damping_ratio** 为整形功能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。

zv_shaper_order：阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 **zv_shaper_order**，**zv_shaper_frequency**，**zv_shaper_damping_ratio** 任意一值不为 0，则使用函数传入的阶次值。

zv_shaper_frequency：频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。

zv_shaper_damping_ratio：阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。

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：（MoveL 直线运动）

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

```

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

```

81 addMoveJSegmentByTarget

```

def addMoveJSegmentByTarget(complex_path_id, vel_percent, acc_percent,
blendradius_percent, zv_shaper_order=0, zv_shaper_frequency=0, zv_shaper_damping_ratio=0, ipAddress="")

```

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

参数：

id_path: 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百分比

acc_percent: 移动到目标路点的加速度百分比

blendradius_percent: 交融半径百分比

zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。

zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。

zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。

zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。

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
import time
zv_shaper_order = 0
zv_shaper_frequency = 0
zv_shaper_damping_ratio = 0
pi = 3.141592653
def to_rad(degree):
    return degree/180 * pi
ret=[False,-1]
ipAddress = '192.168.10.75'
#以 7 轴机器人为例
ret=DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_JOINT_PATH, ipAddress)
if ret[0] == 0:
    joint1 = [0,0,0,to_rad(90),0,0,0]
    joint2 = [0,0,0,to_rad(120),0,0,0]
    DianaApi.addMoveJSegmentByTarget(ret[1],joint1,0.2,0.2,0.1,zv_shaper_order,

```

```

zv_shaper_frequency,zv_shaper_damping_ratio,ipAddress)
    DianaApi.
addMoveJSegmentByTarget(ret[1],joint2,0.2,0.2,0.1,zv_shaper_order,
zv_shaper_frequency,zv_shaper_damping_ratio,ipAddress)
DianaApi.runComplexPath(ret[1] , ipAddress)
time.sleep(15)
DianaApi.destroyComplexPath(ret[1] , ipAddress)

```

82 addMoveJSegmentByPose

```

def addMoveJSegmentByPose(complex_path_id, pose, vel, acc, blendradius,
zv_shaper_order=0,zv_shaper_frequency=0,zv_shaper_damping_ratio=0,
ipAddress='')

```

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

参数：

id_path: 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 MOVEJ 类型 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。单位：rad。

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。

zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。

zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。

zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。

zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。

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
import time
zv_shaper_order = 0
zv_shaper_frequency = 0
zv_shaper_damping_ratio = 0
pi = 3.141592653
def to_rad(deg):
    return deg/180*pi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
time.sleep(2)
ret =
DianaApi.createComplexPath(DianaApi.complex_path_type.NORMAL_POSE_PA

```

```

TH, ipAddress)
if ret[0] == 0:
    pose1 = [0.40286,0,0.871044,0,0,0]
    pose2 = [0.40286,0,0.571044,0,0,0]
    DianaApi.addMoveJSegmentByPose(ret[1],pose1,0.2,0.2,0.1,
zv_shaper_order, zv_shaper_frequency,zv_shaper_damping_ratio,ipAddress)
    DianaApi.addMoveJSegmentByPose(ret[1],pose2,0.2,0.2,0.1,
zv_shaper_order, zv_shaper_frequency,zv_shaper_damping_ratio,ipAddress)
    DianaApi.runComplexPath(ret[1], ipAddress)
    time.sleep(15)
    DianaApi.destroyComplexPath(ret[1],ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()

```

83 addMoveCSegmentByTarget

```

def
addMoveCSegmentByPose(complex_path_id,pass_joints,target_joints,vel,acc, b
lendlradius,ignore_rotation,zv_shaper_order=0,zv_shaper_frequency=0,zv_shap
er_damping_ratio=0, ipAddress='')

```

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

参数：

id_path: 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

zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。

zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。

zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。

zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。

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
import time
zv_shaper_order = 0
zv_shaper_frequency = 0
zv_shaper_damping_ratio = 0

```

```

pi = 3.141592653
def to_rad(deg):
    return deg/180*pi
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0))
DianaApi.releaseBrake(ipAddress)
#以 7 轴机器人为例
ret =
DianaApi.createComplexPath(DianaApi.complex_path_type.MOVEP_JOINT_PATH, ipAddress)
if ret[0] == 0:
    joint0 = [to_rad(0),to_rad(0),to_rad(0),to_rad(120),to_rad(0),to_rad(-60),to_rad(0)]
    joint1 = [to_rad(30),to_rad(0),to_rad(0),to_rad(120),to_rad(0),to_rad(-60),to_rad(0)]
    joint2 = [to_rad(60),to_rad(0),to_rad(0),to_rad(120),to_rad(0),to_rad(-60),to_rad(0)]

    DianaApi.addMoveLSegmentByTarget(ret[1],joint0,0.2,0.2,0.1,zv_shaper_order,
    zv_shaper_frequency,zv_shaper_damping_ratio,ipAddress)
    DianaApi.addMoveCSegmentByTarget(ret[1],joint1,joint2,0.2,0.4,0,True,
    zv_shaper_order, zv_shaper_frequency,zv_shaper_damping_ratio,ipAddress)
    DianaApi.runComplexPath(ret[1], ipAddress)
    time.sleep(15)
    DianaApi.destroyComplexPath(ret[1], ipAddress)
DianaApi.holdBrake(ipAddress)
DianaApi.destroySrv()

```

84 addMoveCSegmentByPose

```

def
addMoveCSegmentByPose(complex_path_id, pass_pose, target_pose,vel, acc, blendradius,zv_shaper_order=0,zv_shaper_frequency=0,zv_shaper_damping_ratio=0, ipAddress='')

```

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

参数：

id_path: 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

zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 为整形功能相关参数，函数中默认值都为 0，此时使用机械臂默认配置参数。

zv_shaper_order: 阶次，范围[-1,2]，其中取值-1 时，关闭整形功能，当 zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio 任意一值不为 0，则使用函数传入的阶次值。

zv_shaper_frequency: 频率，单位：Hz。支持取值范围[0, 1000]，取值为 0 时，使用机械臂默认配置参数。

<p>zv_shaper_damping_ratio: 阻尼比，支持取值范围[0,1]，取值为 0 时，使用机械臂默认配置参数。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值: True: 成功。 False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi import time zv_shaper_order = 0 zv_shaper_frequency = 0 zv_shaper_damping_ratio = 0 pi = 3.141592653 def to_rad(deg): return deg/180*pi ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake() ret = DianaApi.createComplexPath(DianaApi.complex_path_type.MOVEP_POSE_PATH, ipAddress) if ret[0] == 0: pose0 = [0.285369,0.088991,0.633264,to_rad(20.379),to_rad(9.576),to_rad(28.775)] pose1 = [0.20282,-0.227599,0.423234,to_rad(-0.282),to_rad(59.506),to_rad(-0.952)] pose2 = [0.267353,0.241704,0.399786,to_rad(1.057),to_rad(54.307),to_rad(0.907)] DianaApi.addMoveLSegmentByPose(ret[1],pose0,0.2,0.2,0.1,zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio, ipAddress) DianaApi.addMoveCSegmentByPose(ret[1],pose1,pose2,0.05,0.05,0,False,zv_shaper_order, zv_shaper_frequency, zv_shaper_damping_ratio,ipAddress) DianaApi.runComplexPath(ret[1],ipAddress) print(DianaApi.getLastError(ipAddress)) time.sleep(10) DianaApi.destroyComplexPath(ret[1], ipAddress) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv()</pre>

85 runComplexPath

<pre>def runComplexPath(complex_path_id, ipAddress="")</pre>
<p>在指定 IP 地址机械臂上，运行 id 的路段</p> <p>参数: id_path: 输入参数。要运行路段 ID。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值: True: 成功。</p>

False: 失败。
调用示例:
见 createComplexPath 示例

86 destroyComplexPath

def destroyComplexPath(complex_path_id, ipAddress="")
在指定 IP 地址机械臂上，销毁 id 的路段
参数:
id_path: 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

87 saveEnvironment

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

88 enterForceMode_ex

def enterForceMode_ex(forceDirection, forceValue, maxApproachVelocity, maxAllowTcpOffset, active_tcp, ipAddress="")
使指定 IP 地址机械臂进入力控模式，支持用户自定义的坐标系
参数:
forceDirection: 力指令的方向的元组，大小为 3。
forceValue: 力指令的大小，长度为 3 的元组。单位: N。
maxApproachVelocity: 最大接近速度。单位: m/s。
maxAllowTcpOffset: 允许的最大偏移。单位: m。
active_tcp: 用户设定的坐标系的元组，大小为 6。
ipAddress: 可选参数，需要控制机械臂的 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。

<p>调用示例：</p> <pre>import DianaApi force_direction =(0,0,-1) force_value = 2.0 max_approach_velocity = 0.1 max_allow_tcp_offset = 0.2 active_tcp=[0,0,0,0,0,0] ipAddress='192.168.10.75' DianaApi.enterForceMode_ex(force_direction, force_value, max_approach_velocity, max_allow_tcp_offset,active_tcp, ipAddress)</pre>
--

89 readDI

<p>def readDI (group_name,di_name, ipAddress='')</p> <p>读取指定 IP 地址机械臂一个数字输入的值。</p> <p>参数：</p> <p>group_name: 数字输入的分组，例如， 'board','plc';</p> <p>name: 数字输入的信号名，例如， 'di0';</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>元组，共计两个元素，第一个元素表示函数调用是否成功：</p> <p>0: 成功。</p> <p>-1: 失败。</p> <p>第二个元素表示读取的数字输入的值。</p>
<p>调用示例：</p> <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.readDI('board','di0', ipAddress) if ret[0] == 0: print(ret[1]) else: print("cannot read di")</pre>

90readDO

def readDO (group_name,di_name, ipAddress="")
<p>读取指定 IP 地址机械臂一个数字输出的值。</p> <p>参数：</p> <p>group_name: 数字输出的分组，例如，'board','plc'；</p> <p>name: 数字输出的信号名，例如，'do0'；</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>元组，共计两个元素，第一个元素表示函数调用是否成功：</p> <p>0: 成功。</p> <p>-1: 失败。</p> <p>第二个元素表示读取的数字输出的值。</p>
<p>调用示例：</p> <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.readDO('board','do0', ipAddress) if ret[0] == 0: print(ret[1]) else: print("cannot read do")</pre>

91readAI

def readAI (group_name,name, ipAddress="")
<p>读取指定 IP 地址机械臂一个模拟输入的值。</p> <p>参数：</p> <p>group_name: 模拟输入的分组，例如，'board','plc'；</p> <p>name: 模拟输入的信号名，例如，'ai0'；</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>元组，共计两个元素，第一个元素表示函数调用是否成功：</p> <p>0: 成功。</p> <p>-1: 失败。</p> <p>第二个元素表示读取的模拟输出的模式，1 代表电流，2 代表电压。</p> <p>第三个元素表示读取的模拟输出的值。</p>
<p>调用示例：</p> <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.readAI('board','ai0', ipAddress) if ret[0] == 0: print(ret[1]) else: print("cannot read ai")</pre>

92 readAO

def readAO (group_name,name, ipAddress="")
<p>读取指定 IP 地址机械臂一个模拟输出的值。</p>

参数:

group_name: 模拟输出的分组, 例如, 'board','plc';

name: 模拟输出的信号名, 例如, 'ao0';

ipAddress: 可选参数, 需要控制机械臂的 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模拟输出的模式, 1 代表电流, 2 代表电压。

第三个元素表示读取的模拟输出的值。

调用示例:

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.readAO('board','ao0', ipAddress)
if ret[0] == 0:
    print(ret[1])
else:
    print("cannot read ao")
```

93setAIMode

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

设置指定 IP 地址机械臂模拟输入的模式。

参数:

group_name: 模拟输入的分组, 例如, 'board','plc';

name: 模拟输入的信号名, 例如, 'ai0';

mode: 模拟输入模式, 1 代表电流, 2 代表电压。

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

返回值:

True: 成功。

False: 失败。

调用示例:

```
import DianaApi
mode = 1
ipAddress = '192.168.10.75'
ret = DianaApi.setAIMode('board','ai0',mode, ipAddress)
if ret== True:
    print("setAIMode succeed")
else:
    print("setAIMode failed")
```

94 writeDO

```
def writeDO (group_name,do_name,value, ipAddress="")
```

设置指定 IP 地址机械臂一个数字输出的值。

参数:

group_name: 数字输出的分组, 例如, 'board','plc';

name: 数字输出的信号名, 例如, 'do0';

<p>value: 设置的值。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' value = 1 ret = DianaApi.writeDO('board','do0',value, ipAddress) if ret == True: print("writeDO succeed") else: print("writeDO failed")</pre>

95 writeAO

<p>def writeAO (group_name,do_name,value, ipAddress=")</p> <p>设置指定 IP 地址机械臂一个模拟输出的值和模式。</p> <p>参数:</p> <p>group_name: 模拟输出的分组，例如，'board','plc'；</p> <p>name: 模拟输出的信号名，例如：'ao0'；</p> <p>mode: 当前模拟输出模式，1 代表电流，2 代表电压。</p> <p>value: 设置输出的值。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi mode=1 value=8.8 ipAddress = '192.168.10.75' ret = DianaApi.writeAO('board','do0',mode,value, ipAddress) if ret == True: print("writeAO succeed") else: print("writeAO failed")</pre>

96 readBusCurrent

<p>def readBusCurrent(ipAddress=")</p> <p>读取指定 IP 地址机械臂总线电流。</p> <p>参数:</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>元组，共计两个元素，第一个元素表示函数调用是否成功：</p> <p>True: 成功。</p> <p>False: 失败。</p> <p>第二个元素表示读取的总线电流的值。</p>

调用示例:

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

97 readBusVoltage

```
def readBusVoltage(ipAddress="")
```

读取指定 IP 地址机械臂总线电压。

参数:

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

返回值:

元组, 共计两个元素, 第一个元素表示函数调用是否成功:

True: 成功。

False: 失败。

第二个元素表示读取的总线电压的值。

调用示例:

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.readBusVoltage(ipAddress)
if ret[0] == 0:
    print(ret[1])
else:
    print("cannot read bus voltage")
```

98 getDH

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

获取指定 IP 地址机械臂的 DH 参数。

参数:

aDH: 输入输出参数。连杆长度, 长度为 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。

返回值:

True: 成功。

False: 失败。

调用示例:

```
import DianaApi
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))
DianaApi.getDH(a,alpha,d,theta, ipAddress)
```

```
print(a)
print(alpha)
print(d)
print(theta)
DianaApi.destroySrv()
```

99 getOriginalJointTorque

def getOriginalJointTorque (torques, ipAddress="")
<p>获取指定 IP 地址机械臂传感器反馈的扭矩值，未减去零偏。</p> <p>参数：</p> <p>torques: 输入输出参数。反馈的扭矩值，长度为 JOINT_NUM 列表</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi torques = [0] * JOINT_NUM ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.getOriginalJointTorque(torques,ipAddress) print(torques) DianaApi.destroySrv()</pre>

100 getJacobiMatrix

def getJacobiMatrix (matrix_jacobi, ipAddress="")
<p>获取指定 IP 地址机械臂的雅各比矩阵。</p> <p>参数：</p> <p>matrix_jacobi: 输入输出参数。雅可比矩阵，长度为 6*JOINT_NUM 列表</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi matrix_jacobi =[0] * 6 *JOINT_NUM ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.getJacobiMatrix (matrix_jacobi,ipAddress) print(matrix_jacobi) DianaApi.destroySrv()</pre>

101 resetDH

def resetDH(ipAddress="")
<p>重置指定 IP 地址机械臂用户自定义 DH 参数。</p> <p>参数：</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p>

True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.resetDH(ipAddress) DianaApi.destroySrv()

102 runProgram

def runProgram(name, ipAddress="")
运行指定 IP 地址机械臂某个程序。 参数: name: 程序的名字, 最长 127 个英文字符。 ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.runProgram('AgileRobots', ipAddress) if ret == False: print('run Program failed!')

103 stopProgram

def stopProgram(name, ipAddress="")
停止指定 IP 地址机械臂某个程序。 参数: name: 程序的名字, 最长 127 个英文字符。 ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.stopProgram('AgileRobots', ipAddress) if ret == False: print('stop Program failed!')

104 getVariableValue

def getVariableValue(name, ipAddress="")
获取指定 IP 地址机械臂某个全局变量的值。 参数: name: 全局变量的名字 ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。

返回值: 元组，共计两个元素，第一个元素表示函数调用是否成功： True: 成功。 False: 失败。 第二个元素表示获取的全局变量的值。
调用示例: <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.getVariableValue('GLOBAL', ipAddress) if ret[0] == True: print(ret[1]) else: print("getVariableValue failed")</pre>

105 setVariableValue

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

106 isTaskRunning

def isTaskRunning(name, ipAddress="") 判断指定 IP 地址机械臂某个程序是否在运行。
参数: name: 程序名称，最长 127 个英文字符。 ipAddress: 可选参数，需要控制机械臂的 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: <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.isTaskRunning('AgileRobots', ipAddress) if ret == True: print("AgileRobots is running") else:</pre>


```
print("AgileRobots is not running")
```

107 pauseProgram

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

暂停指定 IP 地址机械臂所有程序。

注意：该指令会暂停所有程序，且内部会保留暂停标记，调用过此指令后必须调用 `resumeProgram` 或者 `stopAllProgram` 清除暂停标记，否则下次运行程序会直接进入暂停态。

返回值：

True：成功。

False：失败。

参数：

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

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.pauseProgram(ipAddress)
if ret == True:
    print("AgileRobots is paused")
else:
    print("AgileRobots is not paused")
```

108 resumeProgram

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

恢复运行指定 IP 地址机械臂已经暂停的程序。

参数：

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

返回值：

True：成功。

False：失败。

调用示例：

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.resumeProgram(ipAddress)
if ret == True:
    print("resumeProgram succeed")
else:
    print("resumeProgram failed")
```

109 stopAllProgram

```
def stopAllProgram(ipAddress="")
```

停止指定 IP 地址机械臂所有程序。

参数：

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

返回值：

True：成功。

False：失败。

调用示例:

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.stopAllProgram(ipAddress)
if ret == True:
    print("stopAllProgram succeed")
else:
    print("stopAllProgram failed")
```

110 isAnyTaskRunning

```
def isAnyTaskRunning(ipAddress="")
```

判断指定 IP 地址机械臂是否有程序在运行。

参数:

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

返回值:

True: 成功。

False: 失败。

调用示例:

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.isAnyTaskRunning(ipAddress)
if ret == True:
    print("stopAllProgram succeed")
else:
    print("stopAllProgram failed")
```

111 cleanErrorInfo

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

清除指定 IP 地址机械臂的错误信息。

参数:

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

返回值:

True: 成功。

False: 失败。

调用示例:

```
import DianaApi
ipAddress = '192.168.10.75'
ret = DianaApi.cleanErrorInfo(ipAddress)
if ret == True:
    print("stopAllProgram succeed")
else:
    print("stopAllProgram failed")
```

112 setCollisionLevel

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

设置指定 IP 地址机械臂的碰撞检测类型

参数:

level: 碰撞等级, 值为 E_NO_COLLISION_DETECTION 或者下面几种值的组合

<p>(取值 <code>E_NO_COLLISION_DETECTION</code> 表示只考虑后台最大保护阈值, 取值以下值的组合表示使能对应级别的碰撞检测):</p> <p><code>E_JOINT_SPACE_DETECTION</code>: 关节空间碰撞检测</p> <p><code>E_CART_SPACE_DETECTION</code>: 笛卡尔空间碰撞检测</p> <p><code>E_TCP_RESULTANT_DETECTION</code>: Tcp 合力检测</p> <p><code>ipAddress</code>: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.setCollisionLevel(DianaApi.collision_level.E_NO_COLLISION_DETECTION, ipAddress) if ret == True: print("setCollisionLevel succeed") else: print("setCollisionLevel failed")</pre>

113 getJointCount

<pre>def getJointCount(ipAddress = "")</pre>
<p>在指定 IP 地址机械臂上, 获取其机械臂的关节数目</p> <p>参数:</p> <p><code>ipAddress</code>: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>机械臂的关节数目。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' count = DianaApi.getJointCount(ipAddress) print("Joint Count = %d" %(count))</pre>

114 getWayPoint

<pre>def getWayPoint (waypointName, tcpPos, joints, toolName, workpieceName, ipAddress="")</pre>
<p>获取指定 IP 地址机械臂上路点变量信息。</p> <p>参数:</p> <p><code>waypointName</code>: 路点变量名称。</p> <p><code>tcpPos</code>: 输出参数, 位姿信息, 大小为 6 的列表, 注意, 角度需要用轴角表示</p> <p><code>joints</code>: 输出参数, 关节角信息</p> <p><code>toolName</code>: 输出参数, 路点关联工具坐标系名称</p> <p><code>workpieceName</code>: 输出参数, 路点关联工件坐标系名称</p> <p><code>ipAddress</code>: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>

调用示例:

```

import DianaApi
wayPointName = 'point'
Tcpos= [0] * 6
Joint=[0] * JOINT_NUM
ipAddress = '192.168.10.75'
toolName = []
workName = []
ret = DianaApi.getWayPoint(wayPointName,Tcpos,Joint, toolName,
workName ,ipAddress)
if ret == True:
    print(pos)
    print(toolName)
    print(workName)
else:
    print('cannot get waypoint')

```

115 setWayPoint

```
def setWayPoint(waypointName, tcpos, joints, toolName, workpieceName,
ipAddress="")
```

修改指定 IP 地址机械臂上路点变量的值

参数:

strWayPointName: 路点变量名称。

tcpos: 位姿信息，大小为 6 的数组，注意，角度需要用轴角表示

joints: 关节角信息

toolName: 输入参数，路点关联工具坐标系名称

workpieceName: 输入参数，路点关联工件坐标系名称

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
#以 7 轴机器人为例
pos = [-0.087, 0, 1.316, 0, 3.142, 0]
joints = [-1.518, 0, 22.977, 3.142, 0, 3.142, 0.0]
ipAddress = '192.168.10.75'
toolName = 'tool1'
workName = 'work1'
ret = DianaApi.setWayPoint('test', pos, joints, toolName, workName ,ipAddress)
if ret == True:
    print(pos)
    print(joints)
else:
    print('cannot set waypoint')

```

116 addWayPoint

```
def addWayPoint(waypointName, tcpos, joints, toolName, workpieceName,
ipAddress="")
```

在指定 IP 地址机械臂上新增路点变量。

<p>参数: wayPointName: 输入参数, 路点变量名称。 tcpPos: 输入参数, 位姿信息。 大小为 6 的列表, 注意, 角度需要用轴角表示 joints: 输入参数, 关节角信息, 大小为 JOINT_NUM 的列表。 toolName: 输入参数, 路点关联工具坐标系名称 workpieceName: 输入参数, 路点关联工件坐标系名称 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。</p> <p>返回值: True: 成功。 False: 失败。</p>
<p>调用示例: import DianaApi #以 7 轴机器人为例 pos = [-0.087, 0, 1.316, 0, 3.142, 0] joints = [-1.518, 0, 22.977, 3.142, 0, 3.142, 0.0] ipAddress = '192.168.10.75' toolName = 'tool1' workName = 'work1' ret = DianaApi.addWayPoint('test', pos, joints, toolName, workName, ipAddress) if ret == True: print(pos) print(joints) else: print('cannot add waypoint')</p>

117 deleteWayPoint

<p>def deleteWayPoint(waypointName, ipAddress="")</p> <p>在指定 IP 地址机械臂上删除路点变量。</p> <p>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。</p> <p>返回值: True: 成功。 False: 失败。</p>
<p>调用示例: import DianaApi wayPointName = 'point' ipAddress = '192.168.10.75' ret = DianaApi.deleteWayPoint(wayPointName, ipAddress) if ret == True: print('deleteWayPoint succeed') else: print('deleteWayPoint failed')</p>

118 getDefaultActiveTcp

<p>def getDefaultActiveTcp(default_tcp, ipAddress="")</p>
<p>获取指定 IP 地址机械臂的默认工具坐标系</p>

参数: default_tcp: 当前工具坐标系的矩阵, 为一个长度 16 的列表。 ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: <pre>import DianaApi tcp=[0] * 16 ipAddress='192.168.10.75' DianaApi.getDefaultActiveTcp(tcp, ipAddress) print(tcp)</pre>

119 **getDefaultActiveTcpPose**

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

120 **getActiveTcpPayload**

def getActiveTcpPayload(payload, ipAddress="") 获取指定 IP 地址机械臂的负载信息 参数: arrPose: 负载信息, 第 1 位为质量, 2~4 位为质心, 5~10 位为张量 ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: <pre>import DianaApi load=[0] * 10 ipAddress='192.168.10.75' DianaApi.getActiveTcpPayload(load, ipAddress) print(load)</pre>

121 **zeroSpaceManualMove**

<div>def zeroSpaceManualMove(direction,jointsVel,jointAcc, ipAddress="")</div> <div>使指定 IP 地址机械臂启动零空间手动移动</div> <div>参数:</div> <div>direction: 零空间的方向, 为 zero_space_move_direction 类型的枚举, 枚举及其含义如下:</div> <div><div><div>● E_FORWARD:正向移动</div><div>● E_BACKWARD:反向移动</div></div></div> <div>jointsVel:各关节的速度, 大小为 JOINT_NUM 的列表</div> <div>jointAcc:各关节的加速度, 大小为 JOINT_NUM 的列表</div> <div>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</div> <div>返回值:</div> <div>True: 成功。</div> <div>False: 失败。</div>
<div>调用示例:</div> <div>import DianaApi</div> <div>jointVel=[0.1] * JOINT_NUM</div> <div>jointAcc=[0.1] * JOINT_NUM</div> <div>ipAddress='192.168.10.75'</div> <div>DianaApi.zeroSpaceManualMove (DianaApi.zero_space_move_direction.</div> <div>E_FORWARD, jointVel, jointAcc, ipAddress)</div> <div>time.sleep(3)</div>

122 moveTcp_ex

<div>def moveTcp_ex(coordinate,direction,velocity,accleration, ipAddress="")</div> <div>使指定 IP 地址机械臂进行多种坐标系下的直线移动</div> <div>参数:</div> <div>coordinate:坐标系类型, 应当为枚举类型 coordinate_e, 枚举与其含义如下:</div> <div><div><div>● E_BASE_COORDINATE:基坐标系</div><div>● E_TOOL_COORDINATE:工具坐标系</div><div>● E_WORK_PIECE_COORDINATE:工件坐标系</div><div>● E_VIEW_COORDINATE:视角坐标系</div></div></div> <div>direction:移动方向, 需要为 tcp_direction_e 的枚举类型。枚举值及其含义为:</div> <div><div><div>● T_MOVE_X_UP 表示沿 x 轴正向</div><div>● T_MOVE_X_DOWN 表示沿 x 轴负向</div><div>● T_MOVE_Y_UP 表示沿 y 轴正向</div><div>● T_MOVE_Y_DOWN 表示沿 y 轴负向</div><div>● T_MOVE_Z_UP 表示沿 z 轴正向</div><div>● T_MOVE_Z_DOWN 表示沿 z 轴负向</div></div></div> <div>velocity: 速度, 单位: m/s</div>
--

<p>accleration: 加速度, 单位: m/s^2</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress='192.168.10.75' DianaApi.moveTcp_ex(DianaApi.coordinate_e.E_BASE_COORDINATE, DianaApi.tcp_direction_e.T_MOVE_X_UP,0.1,0.2, ipAddress) time.sleep(2) DianaApi.stop(ipAddress)</pre>

123 setExternalAppendTorCutoffFreq

<p>def setExternalAppendTorCutoffFreq(dblFreq, ipAddress = "")</p> <p>针对指定 IP 地址机械臂, 设置各关节附加力矩的滤波截止频率</p> <p>参数:</p> <p>dblFreq: 输入参数, 各关节附加力矩的滤波截止频率, 需要提供一个正值。</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' ret = DianaApi.setExternalAppendTorCutoffFreq(1.0,ipAddress) if ret == False: print("setExternalAppendTorCutoffFreq failed! Return value = %d" %(ret))</pre>

124 poseTransform

<p>def poseTransform(srcPose,srcMatrixPose,dstMatrixPose,dstPose)</p> <p>把源坐标系下机械臂位姿转换成目的坐标系下位姿</p> <p>参数:</p> <p>srcPose: 输入参数, 源坐标系下的位姿, 大小为 6 的元组, 均为国际单位制(m 和 rad), 注意, 角度需要用轴角表示</p> <p>srcMatrixPose: 输入参数, 源坐标系对应的位姿向量, 大小为 6 的列表</p> <p>dstMatrixPose: 输入参数, 目标坐标系对应的位姿向量, 大小为 6 的列表</p> <p>dstPose: 输出参数, 机械臂在目标坐标系下的位姿, 大小为 6 的列表, 均为国际单位制(m 和 rad), 注意, 角度用轴角表示</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi PI = 3.141592653</pre>


```
srcPose = [-0.087,0,1.3165,PI,0,PI]
srcMatrixPose = [0] * 6
dstMatrixPose = [0.1,0.2,0.3,0,0,0]
dstPose = [0] * 6
DianaApi.poseTransform(srcPose,srcMatrixPose,dstMatrixPose,dstPose)
print(dstPose)
```

125 updateForce

```
def updateForce(forceValue, ipAddress="")
```

针对指定 IP 地址机械臂上，在力控模式下，实时改变力指令的大小

参数：

forceValue: 输入参数，力的大小，需要是一个大于 0 的数

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

返回值：

True: 成功。

False: 失败。

调用示例：

```
import DianaApi
import time
PI=3.141592653
ipAddress = '192.168.10.75'
DianaApi.initSrv((ipAddress,0,0,0,0,0))
DianaApi.releaseBrake(ipAddress)
frameType = DianaApi.coordinate_e.E_BASE_COORDINATE.value
frameMatrix = [1,0,0,0,0,1,0,0,0,0,1,0,0,0,1]
forceDirection=[0,0,-1]
forceValue = 1.0
maxVel = 0.1
maxOffset = 0.2
#以 7 轴机器人为例
joints = [0,PI/6,0,PI/2,0,-PI/3,0]
DianaApi.moveJ(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)
```

126 inverseClosedFull

<pre>def inverseClosedFull(pose,lock_joint_index, lock_joint_position, ref_joints, active_tcp,ipAddress = "")</pre>
<p>在指定 IP 地址机械臂上，基于工具坐标系，给定一个参考关节角，约束单轴求逆解，注意，工业版 Diana 只能锁定第七轴。</p> <p>参数：</p> <p>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</p> <p>lock_joint_index: 输入参数，被约束的关节号</p> <p>lock_joint_position: 输入参数，被约束关节的角度，单位为弧度</p> <p>ref_joints: 参考的关节角，大小为 JOINT_NUM 的列表。</p> <p>active_tcp: 当前工具坐标系对应的位姿向量，大小为 6 的列表</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>非负数：生成逆解对应的 ID</p> <p>-1：失败。</p>
<p>调用示例：</p> <pre>import DianaApi import time def to_rad(deg): return deg * PI / 180 PI=3.141592653 ipAddress = '192.168.10.75' #以 7 轴机器人为例 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])) DianaApi.destroyInverseClosedItems(id) else: print("cannot inverse.\n") DianaApi.destroySrv(ipAddress)</pre>

127 getInverseClosedResultSize

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

128 getInverseClosedJoints

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

129 destoryInverseClosedItems

def destoryInverseClosedItems(id,ipAddress = "")
在指定 IP 地址机械臂上，根据 ID 删除约束单轴求逆解的结果数据集
参数：
id: 输入参数，逆解对应的 ID
ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。
返回值：
True: 成功。
False: 失败。
调用示例：
见 inverseClosedFull 示例

130 getGravInfo

def getGravInfo(grav,ipAddress = "")
针对指定 IP 地址的机械臂，获取其安装信息的重力矢量
参数：
grav: 输出参数，重力矢量，大小为 3 的列表
ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。
返回值：

True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' grav = [0.0]*3 ret = DianaApi.getGravInfo(grav,ipAddress) if ret == False: print("getGravInfo failed! Return value = %d\n"%(ret)) else: print(grav)

131 setGravInfo

def setGravInfo(grav,ipAddress = "")
针对指定 IP 地址的机械臂，设置其重力矢量 参数： grav: 输入参数，重力矢量，大小为 3 的元组 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' grav = [0.0]*3 ret = DianaApi.setGravInfo(grav,ipAddress) if ret == False: print("setGravInfo failed! Return value = %d\n"%(ret))

132 getGravAxis

def getGravAxis(grav_axis,ipAddress = "")
针对指定 IP 地址的机械臂，获取其安装信息的轴角 参数： grav_axis: 输出参数，安装时的轴角，单位为 rad，大小为 3 的列表 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值: True: 成功。 False: 失败。
调用示例: import DianaApi ipAddress = '192.168.10.75' grav_axis = [0.0]*3 ret = DianaApi.getGravAxis(grav_axis,ipAddress) if ret == False: print("getGravAxis failed! Return value = %d\n"%(ret)) else: print(grav_axis)

133 setGravAxis

<code>def setGravAxis(grav_axis,ipAddress = '')</code>
针对指定 IP 地址的机械臂，设置其安装信息的轴角 参数： grav_axis: 输入参数，安装轴角，单位 rad，大小为 3 的元组 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值： True: 成功。 False: 失败。
调用示例： import DianaApi ipAddress = '192.168.10.75' grav_axis = [0.0]*3 ret = DianaApi.setGravAxis(grav,ipAddress) if ret < 0: print("setGravAxis failed! Return value = %d\n"%(ret))

134 speedLOnTcp

<code>def speedLOnTcp(speed, acc, t=0.0, ipAddress='')</code>
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 为非零时，机器人将在 t 时间后减速。如果 t 为 0，机器人将在达到目标速度时减速。该函数调用后立即返回。停止运动需要调用 stop 函数。 该函数暂不支持传送带跟踪期间使用，目前没有主动报错，使用时会卡住程序，如果使用了传送带功能请注意规避。 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: 加速度元组，元组大小为 2，前一个为平移加速度，单位：m/s ² 。后一个为旋转加速度，单位：rad/s ² t: 时间，单位：s。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。 返回值： True: 成功。 False: 失败。
调用示例： import DianaApi ipAddress = '192.168.10.75' DianaApi.initSrv((ipAddress,0,0,0)) DianaApi.releaseBrake(ipAddress) speeds = [0.1,0.0,0.0,0.0,0.0,0.0] acc = [0.10, 0.50] DianaApi.speedLOnTcp(speeds, acc, 0, ipAddress) time.sleep(3) DianaApi.holdBrake(ipAddress) DianaApi.destroySrv()

135 getTcpForceInToolCoordinate

<code>def getTcpForceInToolCoordinate(forces,ipAddress = '')</code>
在指定 IP 地址机械臂上，获取工具坐标系的 Tcp 外力值 参数：

<p>forces: 输出参数, Tcp 外力, 大小为 6</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' forces = [0.0]*6 ret = DianaApi.getTcpForceInToolCoordinate(forces,ipAddress) if ret == False: print("getTcpForceInToolCoordinate failed! Return value = %d"%(ret)) else: print(forces)</pre>

136 calculateJacobi

<pre>def calculateJacobi(dblJacobiMatrix,dblJointPosition,ipAddress = "")</pre>
<p>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</p> <p>参数:</p> <p>dblJacobiMatrix: 输出参数, 雅各比矩阵, 大小为 6*JOINT_NUM 的列表</p> <p>dblJointPosition: 输入参数, 用于计算雅各比矩阵的关节角 (与当前机械臂反馈关节角无关), 大小为 JOINT_NUM 的列表</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' 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")</pre>

137 calculateJacobiTF

<pre>def calculateJacobiTF(dblJacobiMatrix,dblJointPosition,toolMatrix ,ipAddress = "")</pre>
<p>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</p>

<p>参数:</p> <p>dblJacobiMatrix: 输出参数, 雅各比矩阵, 大小为 6*JOINT_NUM 的列表</p> <p>dblJointPosition: 输入参数, 用于计算雅各比矩阵的关节角 (与当前机械臂反馈关节角无关), 大小为 JOINT_NUM 的列表</p> <p>toolMatrix: 工具位姿信息</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例:</p> <pre>import DianaApi ipAddress = '192.168.10.75' jointCount = JOINT_NUM tcpCount = 6 jacobiMatrix = [0.0] *(jointCount*tcpCount) jointPosition = [0.0] * jointCount toolMatrix = [0.0] * 6 ret = DianaApi.calculateJacobiTF(jacobiMatrix,jointPosition,toolMatrix, ipAddress) if ret == False: 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")</pre>

138 getMechanicalJointsPositionRange

<p>def getMechanicalJointsPositionRange(arrMin, arrMax, ipAddress = "")</p> <p>获取指定 IP 地址机械臂各关节角的机械限位。</p> <p>参数:</p> <p>arrMin: 传出参数, JOINT_NUM 个轴关节角度的最小机械限位元组。单位: rad。</p> <p>arrMax: 传出参数, JOINT_NUM 个轴关节角度的最大机械限位元组。单位: rad。</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>函数执行是否成功。</p>
<p>调用示例:</p> <pre>ipAddress = '192.168.10.75' minPos = [0.0]*7 maxPos = [0.0]*7 DianaApi.getMechanicalJointsPositionRange(minPos, maxPos, ipAddress) print("getMechanicalJointsPositionRange::minPos="+str(minPos)) print("getMechanicalJointsPositionRange::maxPos="+str(maxPos))</pre>

139 getMechanicalMaxJointsVel

<p>def getMechanicalMaxJointsVel(arrVel, ipAddress = "")</p>

<p>获取指定 IP 地址机械臂各关节最大机械速度。</p> <p>参数： arrVel: 传出参数，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。</p> <p>返回值： 函数执行是否成功。</p>
<p>调用示例： ipAddress = '192.168.10.75' maxVel = [0.0]*7 DianaApi.getMechanicalMaxJointsVel(maxVel, ipAddress) print("getMechanicalMaxJointsVel::maxVel="+str(maxVel))</p>

140 getMechanicalMaxJointsAcc

<p>def getMechanicalMaxJointsAcc(arrAcc, ipAddress = '')</p> <p>获取指定 IP 地址机械臂各关节最大机械加速度。</p> <p>参数： arrAcc: 传出参数，JOINT_NUM 个轴关节角度的最大机械加速度元组。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。</p> <p>返回值： 函数执行是否成功。</p>
<p>调用示例： ipAddress = '192.168.10.75' maxAcc = [0.0]*7 DianaApi.getMechanicalMaxJointsAcc(maxAcc, ipAddress) print("getMechanicalMaxJointsAcc::maxAcc="+str(maxAcc))</p>

141 getMechanicalMaxCartVelAcc

<p>def getMechanicalMaxCartVelAcc(ipAddress = '')</p> <p>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。</p> <p>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。</p> <p>返回值： 由查询是否成功和四个速度加速度组成的元组： (1) 查询是否成功。 (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/s²。 (5) 笛卡尔空间最大机械旋转加速度，double 型变量，单位：rad/s²。</p>
<p>调用示例： ipAddress = '192.168.10.75' ret,dblTranslationVel, dblRotationVel, dblTranslationAcc, dblRotationAcc = DianaApi.getMechanicalMaxCartVelAcc(ipAddress) print("getMechanicalCartMaxVelAcc::dblTranslationVel="+str(dblTranslationVel)+" dblRotationVel="+str(dblRotationVel)+"dblTranslationAcc="+str(dblTranslati onAcc)+"dblRotationAcc="+str(dblRotationAcc))</p>

142 getJointsPositionRange

def getJointsPositionRange(arrMin, arrMax, ipAddress = "")
<p>获取指定 IP 地址机械臂各关节的极限位。</p> <p>参数：</p> <p>arrMin: 传出参数，JOINT_NUM 个轴关节角度的最小极限位元组。单位：rad。</p> <p>arrMax: 传出参数，JOINT_NUM 个轴关节角度的最大极限位元组。单位：rad。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>函数执行是否成功。</p>
<p>调用示例：</p> <pre>ipAddress = '192.168.10.75' minPos = [0.0]*7 maxPos = [0.0]*7 DianaApi.getJointsPositionRange(minPos, maxPos, ipAddress) print("getJointsPositionRange::minPos="+str(minPos)) print("getJointsPositionRange::maxPos="+str(maxPos))</pre>

143 getMaxJointsVel

def getMaxJointsVel(arrVel, ipAddress = "")
<p>获取指定 IP 地址机械臂各关节最大软速度。</p> <p>参数：</p> <p>arrVel: 传出参数，JOINT_NUM 个轴关节角度的最大软速度元组。单位：rad/s。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>函数执行是否成功。</p>
<p>调用示例：</p> <pre>ipAddress = '192.168.10.75' maxVel = [0.0]*7 DianaApi.getMaxJointsVel(maxVel, ipAddress) print("getMaxJointsVel::maxVel="+str(maxVel))</pre>

144 getMaxJointsAcc

def getMaxJointsAcc(arrAcc, ipAddress = "")
<p>获取指定 IP 地址机械臂各关节最大软加速度。</p> <p>参数：</p> <p>arrAcc: 传出参数，JOINT_NUM 个轴关节角度的最大软加速度元组。单位：rad/s²。</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>函数执行是否成功。</p>
<p>调用示例：</p> <pre>ipAddress = '192.168.10.75'</pre>

```
maxAcc = [0.0]*7
DianaApi.getMaxJointsAcc(maxAcc, ipAddress)
print("getMaxJointsAcc::maxVel="+str(maxAcc))
```

145 getMaxCartTranslationVel

def getMaxCartTranslationVel(ipAddress = "")
<p>获取指定 IP 地址机械臂笛卡尔空间最大软平移速度。</p> <p>参数:</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>由查询是否成功和笛卡尔空间最大软平移速度组成的元组:</p> <p>(1) 查询是否成功。</p> <p>(2) 笛卡尔空间最大平移速度, double 型变量, 单位: m/s。</p>
<p>调用示例:</p> <pre>ipAddress = '192.168.10.75' ret, dblTranslationVel = DianaApi.getMaxCartTranslationVel(ipAddress) print("getMaxCartRotationAcc::dblTranslationVel="+str(dblTranslationVel))</pre>

146 getMaxCartRotationVel

def getMaxCartRotationVel(ipAddress = "")
<p>获取指定 IP 地址机械臂笛卡尔空间最大软旋转速度。</p> <p>参数:</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>由查询是否成功和笛卡尔空间最大软旋转速度组成的元组:</p> <p>(1) 查询是否成功。</p> <p>(2) 笛卡尔空间最大旋转速度, double 型变量, 单位: rad/s。</p>
<p>调用示例:</p> <pre>ipAddress = '192.168.10.75' ret, dblRotationVel = DianaApi.getMaxCartRotationVel(ipAddress) print("getMaxCartRotationVel::dblRotationVel="+str(dblRotationVel))</pre>

147 getMaxCartTranslationAcc

def getMaxCartTranslationAcc(ipAddress = "")
<p>获取指定 IP 地址机械臂笛卡尔空间最大软平移加速度。</p> <p>参数:</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>由查询是否成功和笛卡尔空间最大平移加速度组成的元组:</p> <p>(1) 查询是否成功。</p> <p>(2) 笛卡尔空间最大平移加速度, double 型变量, 单位: m/s²。</p>
<p>调用示例:</p> <pre>ipAddress = '192.168.10.75'</pre>

```
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软旋转加速度。
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：
由查询是否成功和笛卡尔空间最大旋转加速度组成的元组：
(1) 查询是否成功。
(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：
<pre>ipAddress = '192.168.10.75' ret, dblRotationAcc = DianaApi.getMaxCartRotationAcc(ipAddress) print("getMaxCartRotationAcc::dblRotationAcc="+str(dblRotationAcc))</pre>

149 setJointsPositionRange

def setJointsPositionRange(arrMinPos, arrMaxPos, ipAddress = "")
设置指定 IP 地址机械臂各关节极限位。
参数：
arrMinPos: 传入参数，JOINT_NUM 个轴关节角度的最小极限位元组。单位：rad。
arrMaxPos: 传入参数，JOINT_NUM 个轴关节角度的最大极限位元组。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。
返回值：
函数执行是否成功。
调用示例：
<pre>ipAddress = '192.168.10.75' minPos = [to_rad(-150.0), to_rad(-80.0), to_rad(-150.0), to_rad(0.0), to_rad(-150.0), to_rad(-150.0), to_rad(-150.0)] maxPos = [to_rad(150.0), to_rad(80.0), to_rad(150.0), to_rad(150.0), to_rad(150.0), to_rad(30.0), to_rad(150.0)] DianaApi.setJointsPositionRange(minPos, maxPos, ipAddress) print(minPos) print(maxPos) DianaApi.saveEnvironment(ipAddress)</pre>

150 setMaxJointsVel

def setMaxJointsVel(arrVel, ipAddress = "")
设置指定 IP 地址机械臂各关节的最大软速度。
参数：
arrVel: 传入参数，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。 返回值： 函数执行是否成功。
调用示例： <pre> ipAddress = '192.168.10.75' maxVel = [to_rad(150.0), to_rad(150.0), to_rad(150.0), to_rad(150.0), to_rad(150.0), to_rad(150.0), to_rad(150.0)] DianaApi.setMaxJointsVel(maxVel, ipAddress) print(maxVel) DianaApi.saveEnvironment(ipAddress) </pre>

151 setMaxJointsAcc

<pre>def setMaxJointsAcc(arrAcc, ipAddress = '')</pre>
设置指定 IP 地址机械臂各关节的最大软加速度。 参数： arrAcc: 传入参数，JOINT_NUM 个轴关节角度的最大加速度元组。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。 返回值： 函数执行是否成功。
调用示例： <pre> ipAddress = '192.168.10.75' maxAcc = [0.0]*7 DianaApi.getMaxJointsAcc(maxAcc, ipAddress) for i in range(0,7): maxAcc[i] = maxAcc[i] - 2.0 DianaApi.setMaxJointsAcc(maxAcc, ipAddress) DianaApi.saveEnvironment(ipAddress) </pre>

152 setMaxCartTranslationVel

<pre>def setMaxCartTranslationVel(TranslationVel, ipAddress = '')</pre>
设置指定 IP 地址机械臂笛卡尔空间最大软平移速度。 参数： TranslationVel: 传入参数，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： 函数执行是否成功。
调用示例： <pre> ipAddress = '192.168.10.75' cartTransVel = 1.0 DianaApi.setMaxCartTranslationVel(cartTransVel, ipAddress) DianaApi.saveEnvironment(ipAddress) </pre>

153 setMaxCartRotationVel

def setMaxCartRotationVel(RotationVel, ipAddress = "")
设置指定 IP 地址机械臂笛卡尔空间最大软旋转速度。 参数： RotationVel: 传入参数，笛卡尔空间最大软旋转速度，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： 函数执行是否成功。
调用示例： <pre>ipAddress = '192.168.10.75' cartRotVel = 1.0 DianaApi.setMaxCartRotationVel(cartRotVel, ipAddress) DianaApi.saveEnvironment(ipAddress)</pre>

154 setMaxCartTranslationAcc

def setMaxCartTranslationAcc(lineAcc, ipAddress = "")
设置指定 IP 地址机械臂笛卡尔空间最大软平移加速度。 参数： lineAcc: 传入参数，笛卡尔空间最大软平移加速度，double 型变量，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： 函数执行是否成功。
调用示例： <pre>ipAddress = '192.168.10.75' cartTransAcc = 1.0 DianaApi.setMaxCartTranslationAcc(cartTransAcc, ipAddress) DianaApi.saveEnvironment(ipAddress)</pre>

155 setMaxCartRotationAcc

def setMaxCartRotationAcc(rotationAcc, ipAddress = "")
设置指定 IP 地址机械臂笛卡尔空间最大软旋转加速度。 参数： rotationAcc: 传入参数，笛卡尔空间最大软旋转加速度，double 型变量，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： 函数执行是否成功。
调用示例： <pre>ipAddress = '192.168.10.75' cartRotAcc = 1.0 DianaApi.setMaxCartRotationAcc(cartRotAcc, ipAddress) DianaApi.saveEnvironment(ipAddress)</pre>

156 requireHandlingError

def requireHandlingError(ipAddress = "")
判断当前系统是否有错误待处理。 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(是否存在待处理的错误，错误码)，(True, -2206)表示有错误待处理, 错误码是-2206（有关节超出极限位置）。
调用示例： ipAddress = '192.168.10.75' ret = DianaApi.requireHandlingError (ipAddress) if ret[0] == False: print('No handling error') else: print('Error should be handling:' + str(ret[1]))

157 getJointsSoftLimitRange

def getJointsSoftLimitRange(min, max, ipAddress = "")
获取当前系统的软限位。 参数： min: 用于接收软限位下限，输入时必须为列表，列表大小不能小于关节数 max: 用于接收软限位上限，输入时必须为列表，列表大小不能小于关节数 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： 函数执行是否成功。
调用示例： ipAddress = '192.168.10.75' min=[0, 0, 0, 0, 0, 0, 0] max=[0, 0, 0, 0, 0, 0, 0] ret = DianaApi. getJointsSoftLimitRange (min, max, ipAddress) print('min soft position:' + str(min)) print('max soft position:' + str(max))

158 setJointsSoftLimitRange

def setJointsSoftLimitRange(min, max, ipAddress = "")
设置当前系统的软限位。 参数： min: 新的软限位下限，输入为元组或者列表，元组或列表大小不能小于关节数

<p>max: 新的软限位上限，输入为元组或者列表，元组或列表大小不能小于关节数</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值: 函数执行是否成功。</p>
<p>调用示例:</p> <pre>ipAddress = '192.168.10.75' min=[0, 0, 0, 0, 0, 0, 0] max=[0, 0, 0, 0, 0, 0, 0] DianaApi.getJointsSoftLimitRange (min, max, ipAddress) for i in range(len(min)): min[i] = min[i] + 0.1 max[i] = max[i] - 0.1 DianaApi.setJointsSoftLimitRange (min, max, ipAddress)</pre>

159 getFunctionOptI4

<p>def getFunctionOptI4(function_index, opt_name, ipAddress = "")</p> <p>获取功能可选参数。</p> <p>参数:</p> <p>function_index: 取值为 function_index_e 类型，当获取零力驱动（暂不支持）可选参数时，该值为 E_FREEDRIVING，当获取笛卡尔阻抗可选参数时，该值为 E_CART_IMPEDANCE</p> <p>opt_name: 取值为 function_opt_name_e 类型，用于指定功能的可选参数，当 function_index 为 E_FREEDRIVING 时，当前参数被忽略，当 function_index 为 E_CART_IMPEDANCE 时，当前参数可选择的有：</p> <ul style="list-style-type: none"> a) E_COORDINATE_TYPE: 表示笛卡尔阻抗的参考坐标系； b) E_IS_SINGLE_AXIS_LOCKED: 用于标志当前阻抗控制是否锁轴。 <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值: 元组(是否存在该属性，属性值)，(True, 1)表示有该属性，且值为 1。随 opt_name 的值变化：</p> <ul style="list-style-type: none"> a) E_COORDINATE_TYPE: 0 代表基坐标系，1 代表工具坐标系； b) E_IS_SINGLE_AXIS_LOCKED: 1 代表锁轴。
<p>调用示例:</p> <pre>ipAddress = '192.168.10.75' ret = DianaApi.getFunctionOptI4(DianaApi.function_index_e.E_CART_IMPEDANCE, DianaApi.function_opt_name_e.E_COORDINATE_TYPE) if ret[0] == True: print('Coordinate for cart impedance:' + str(ret[1])) else: print('Do not support this opt')</pre>

160 **setFunctionOptI4**

<div>def setFunctionOptI4(function_index, opt_name, opt_value, ipAddress = "")</div> <div>设置功能可选参数。</div> <div>参数：</div> <div>function_index: 取值为 function_index_e 类型，当设置零力驱动（暂不支持）时，该值为 E_FREEDRIVING,; 当设置笛卡尔阻抗时，该值为 E_CART_IMPEDANCE。</div> <div>opt_name: 取值为 function_opt_name_e 类型，用于指定功能的可选参数，当 function_index 为 E_CART_IMPEDANCE 时，当前参数可选择的有：</div> <div> a) E_COORDINATE_TYPE: 表示笛卡尔阻抗的参考坐标系；</div> <div> b) E_LOCKED_SINGLE_AXIS: 用于指定阻抗控制锁轴的索引。</div> <div>opt_value:可选参数的新值，随 opt_name 的值变化：</div> <div> a) E_COORDINATE_TYPE: 0 代表基坐标系，1 代表工具坐标系；</div> <div> b) E_LOCKED_SINGLE_AXIS: 当前版本仅支持锁第 3 轴（即取值 2），-1 代表解锁。</div> <div>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</div> <div>返回值：</div> <div>函数执行是否成功。</div>	
<div>调用示例：</div> <div>ipAddress = '192.168.10.75'</div> <div>ret = DianaApi.setFunctionOptI4(DianaApi.function_index_e.E_CART_IMPEDANCE, DianaApi.function_opt_name_e.E_COORDINATE_TYPE, 1)</div>	

161 **enterRescueMode**

<div>def enterRescueMode(ipAddress = "")</div> <div>进入安全处理状态，安全处理状态有三种模式，安全零力模式，关节驱动模式和笛卡尔驱动模式，默认进入关节驱动模式，进入安全处理状态后可通过 switchRescueMode 来切换安全处理模式，只有控制器处于 Idle 状态时可以进行该操作。原函数名为 enterSafetyIdle。</div> <div>参数：</div> <div>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</div> <div>返回值：</div> <div>函数执行是否成功。</div>	
<div>调用示例：</div> <div>ipAddress = '192.168.10.75'</div> <div>ret = DianaApi. enterRescueMode()</div>	

162 **leaveRescueMode**

def leaveRescueMode(ipAddress = "")
退出安全处理模式。原函数名为 <code>leaveSafetyIdle</code> 。
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： 函数执行是否成功。
调用示例： <code>ipAddress = '192.168.10.75'</code> <code>ret = DianaApi.leaveRescueMode()</code>

163 getCartImpedanceCoordinateType

def getCartImpedanceCoordinateType(ipAddress = "")
在指定 IP 地址机械臂上，获取设置笛卡尔阻抗模式时坐标系种类。
参数： 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： 坐标系的类型，0：基坐标系，1：工具坐标系
调用示例： <code>import DianaApi</code> <code>ipAddress = "192.168.10.75"</code> <code>ret = DianaApi.getCartImpedanceCoordinateType(ipAddress)</code> <code>if ret == 0:</code> <code>print("it's base coordinate\n")</code> <code>else:</code> <code>print("it's tool coordinate\n")</code>

164 setCartImpedanceCoordinateType

def setCartImpedanceCoordinateType(intCoordinateType, ipAddress = "")
在指定 IP 地址机械臂上，设置笛卡尔阻抗模式时坐标系种类。
参数： intCoordinateType: 坐标系的类型，0：基坐标系，1：工具坐标系 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.setCartImpedanceCoordinateType(0,ipAddress)

165 setJointLockedInCartImpedanceMode

def setJointLockedInCartImpedanceMode (isLocked, lockIndex, 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（当前版本仅支持第 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。（注：该 API 是一个设置选项，默认情况下不锁轴，如果需要在笛卡尔阻抗模式或力控模式下锁定某轴，需要先设置该选项再进入笛卡尔阻抗模式或力控模式） 参数: isLocked: 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第 3 轴，即 intLockedJointIndex 必须为 2，否则锁定无效）；如果该值为 false，则不论 intLockedJointIndex 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第 3 轴）。 lockIndex: 输入可选参数，表示轴的索引值（索引从 0 开始），缺省值为 2（即第 3 轴）。 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: 函数执行是否成功。
调用示例: import DianaApi ipAddress = "192.168.10.75" DianaApi.setJointLockedInCartImpedanceMode (True, 2, ipAddress)

166 getJointLockedInCartImpedanceMode

def getJointLockedInCartImpedanceMode (ipAddress = “)
--

<p>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（当前版本仅支持第 3 轴）是否属于锁定/解锁状态。</p> <p>参数：</p> <p>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。</p> <p>返回值：</p> <p>(查询是否成功, 是否锁轴), 例如(True, True)</p>
<p>调用示例：</p> <pre>import DianaApi ipAddress = "192.168.10.75" ret = DianaApi.getJointLockedInCartImpedanceMode (ipAddress) if ret[0]==True: if ret[1] == True: print('机械臂处于锁轴模式') else: print('机械臂未锁轴') else: print('查询失败')</pre>

167 **setThresholdTorque**

<p>def setThresholdTorque (arrThreshold, ipAddress="")</p> <p>设置指定 IP 地址机械臂各关节传感器检测阈值，此阈值在切换工作模式时用于检测是否允许切换工作模式。</p> <p>参数：</p> <p>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）</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi arrThreshold = (6, 6, 5, 5, 2, 2, 2) ipAddress = '192.168.10.75' DianaApi.setThresholdTorque (arrThreshold, ipAddress)</pre>

168 **getThresholdTorque**

<p>def getThresholdTorque (arrThreshold, ipAddress="")</p> <p>获取指定 IP 地址机械臂各关节传感器检测阈值，此阈值在切换工作模式时</p>
--

<p>用于检测是否允许切换工作模式。</p> <p>参数：</p> <p>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)</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi arrThreshold = [0] * JOINT_NUM ipAddress = '192.168.10.75' DianaApi.getThresholdTorque (arrThreshold, ipAddress) print("arrThreshold:") print(arrThreshold)</pre>

169 setHeartbeatParam

<p>def setHeartbeatParam(disconnectTimeout, stopRobotTimeout, ipAddress="")</p> <p>设置 API 心跳相关的超时时间。</p> <p>参数：</p> <p>disconnectTimeout: API 断连超时时间,单位(毫秒 ms)(-1 时相当于永不断连)</p> <p>stopRobotTimeout: 停止机器人超时时间,单位(毫秒 ms)(-1 时相当于永不停止)</p> <p>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。</p> <p>返回值：</p> <p>0: 成功。</p> <p>-1: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi disconnectTimeout = c_int(-1) stopRobotTimeout = c_int(-1) DianaApi.setHeartbeatParam(disconnectTimeout, stopRobotTimeout) print("ret:" + str(ret))</pre>

170 getHeartbeatParam

<p>def getHeartbeatParam(disconnectTimeout, stopRobotTimeout, ipAddress="")</p> <p>获取 API 心跳相关的超时时间。</p> <p>参数：</p> <p>disconnectTimeout: API 断连超时时间,单位(毫秒 ms)(-1 时相当于永不断连)</p> <p>stopRobotTimeout: 停止机器人超时时间,单位(毫秒 ms)(-1 时相当于永不停止)</p> <p>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。</p>

返回值: 0: 成功。 -1: 失败。
调用示例: <pre>import DianaApi disconnectTimeout = c_int(-1) stopRobotTimeout = c_int(-1) ret = DianaApi.getHeartbeatParam(disconnectTimeout, stopRobotTimeout) print("ret:" + str(ret)) print("set distimeout:" + str(disconnectTimeout) + ", stopTimeout:" + str(stopRobotTimeout))</pre>

171 customRobotState

<pre>def customRobotState(action, customBits, ipAddress="")</pre> <p>定制机器人状态推送信息。</p> <p>参数:</p> <p>action: 取值范围 0-2, 分别定义为:</p> <ul style="list-style-type: none"> 0: 增加 customBits 指示的推送信息。 1: 减少 customBits 指示的推送信息。 2: 重置为 customBits 指示的推送信息。 <p>CustomBits: 定制推送信息比特位。</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>0: 成功。 -1: 失败。</p>
调用示例: <pre>import DianaApi action=1 configBits = c_ulonglong(0x188) ret = DianaApi.customRobotState(action, configBits) print("ret:" + str(ret))</pre>

172 getCustomRobotState

<pre>def getCustomRobotState(customBits, ipAddress="")</pre> <p>获取当前定制的推送信息。</p> <p>参数:</p> <p>CustomBits: 定制推送信息比特位。</p> <p>ipAddress: 可选参数, 需要控制机械臂的 IP 地址字符串, 不填仅当只连接一台机械臂时生效。</p> <p>返回值:</p> <p>0: 成功。 -1: 失败。</p>
调用示例: <pre>import DianaApi configBits = c_ulonglong(0x188) action = 1 ret = DianaApi.customRobotState(action, configBits) ret = DianaApi.getCustomRobotState(configBits) print("ret:%#x" %ret) print("configBits:%#x" %c_ulonglong(configBits.value).value)</pre>

173 getTcpPosByTcpName

<pre>def getTcpPosByTcpName(tcpName, arrPose, ipAddress="")</pre>
<p>根据工具坐标系名称获取工具坐标系位姿。</p> <p>参数：</p> <p>tcpName: 工具坐标系名称</p> <p>arrPose: 工具坐标系的位姿，为一个长度 6 的列表，注意，角度需要用轴角表示</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi arrPose=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0] ret = DianaApi.getTcpPosByTcpName("1", arrPose) print(ret) if (ret): print(str(arrPose[0]) + str(arrPose[1]) + str(arrPose[2]) + str(arrPose[3]) + str(arrPose[4]) + str(arrPose[5]))</pre>

174 getTcpPoseByWorkPieceName

<pre>def getTcpPoseByWorkPieceName(workPieceName, arrPose, ipAddress="")</pre>
<p>根据工件坐标系名称获取工件坐标位姿。</p> <p>参数：</p> <p>workPieceName: 工件坐标系名称</p> <p>arrPose: 工件坐标系的位姿，为一个长度 6 的列表，注意，角度需要用轴角表示</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。</p> <p>返回值：</p> <p>True: 成功。</p> <p>False: 失败。</p>
<p>调用示例：</p> <pre>import DianaApi arrPose=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0] ret = DianaApi.getTcpPoseByWorkPieceName("wp1", arrPose) print(ret) if (ret): print(str(arrPose[0]) + str(arrPose[1]) + str(arrPose[2]) + str(arrPose[3]) + str(arrPose[4]) + str(arrPose[5]))</pre>

175 getPayLoadByTcpName

<pre>def getPayLoadByTcpName(tcpName, payload, ipAddress="")</pre>
<p>根据工具坐标系名称获取工具坐标系负载信息。</p> <p>参数：</p> <p>tcpName: 工具坐标系名称</p> <p>payload: 负载信息，第 1 位为质量，2~4 位为质心，5~10 位为张量</p> <p>ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接</p>

一台机械臂时生效。 返回值： True：成功。 False：失败。
调用示例： <pre>import DianaApi payload=[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0] ret = DianaApi.getPayLoadByTcpName ("1", payload) print(ret) if (ret): print(str(payload[0]) + str(payload[1]) + str(payload[2]) + str(payload[3]) + str(payload[4]) + str(payload[5]) + str(payload[6]) + str(payload[7]) + str(payload[8]) + str(payload[9]))</pre>

176 setDefaultToolTcpCoordinate

<pre>def setDefaultToolTcpCoordinate(tcpName, ipAddress="")</pre>
设置默认工具坐标系。 参数： tcpName：工具坐标系名称 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： <pre>import DianaApi ret = DianaApi.setDefaultToolTcpCoordinate("0") print("set tcpcoor:" + str(ret))</pre>

177 setDefaultWorkPieceTcpCoordinate

<pre>def setDefaultWorkPieceTcpCoordinate(workPieceName, ipAddress="")</pre>
设置默认工件坐标系。 参数： workPieceName：工件坐标系名称 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： <pre>import DianaApi ret = DianaApi.setDefaultWorkPieceTcpCoordinate("workpiececoor") print("set wpcoor:" + str(ret))</pre>

178 getDefaultTcpCoordinate

<pre>def getDefaultTcpCoordinate(ipAddress="")</pre>
获取默认的工具坐标系名称。 参数： ipAddress：可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。

返回值: 元组，共计两个元素，第一个元素表示函数调用是否成功： True: 成功。 False: 失败。 第二个元素表示获取的默认工具坐标系的值。
调用示例: <pre>import DianaApi ret, coordName = DianaApi.getDefaultTcpCoordinate() print("tcp, ret:" + str(ret) + ":" + coordName)</pre>

179 **getDefaultWorkPieceCoordinate**

def getDefaultWorkPieceCoordinate(ipAddress="") 获取默认的工作坐标系名称。
参数: ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。
返回值: 元组，共计两个元素，第一个元素表示函数调用是否成功： True: 成功。 False: 失败。 第二个元素表示获取的默认工作坐标系的值。
调用示例: <pre>import DianaApi ret, coordName = DianaApi.getDefaultWorkPieceCoordinate () print("tcp, ret:" + str(ret) + ":" + coordName)</pre>

180 **setVelocityPercentValue**

def setVelocityPercentValue(value, ipAddress="") 设置速度百分比。
参数: value: 设置的百分比速度值。 ipAddress: 可选参数，需要控制机械臂的 IP 地址字符串，不填仅当只连接一台机械臂时生效。
返回值: 0: 成功。 -1: 失败。
调用示例: <pre>import DianaApi Value = 20 ret = setVelocityPercentValue(value) print("ret:" + str(ret))</pre>

181 **switchRescueMode**

def switchRescueMode(scuWorkMode, ipAddress="")
--

切换安全处理模式，只有在安全处理模式下才能进行切换。

参数：

scuWorkMode: 安全处理模式，取值为：12-安全零力模式，13-关节驱动模式，14-笛卡尔驱动模式。

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.enterRescueMode()

ret = DianaApi.switchRescueMode(14)
```

附件 A: DianaApi 接口错误码

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_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	-	电机三相电流长

	2007	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
ERROR_CODE_LOST_HEART_WITH_DIANAROBOT_ERROR	- 2102	与后台服务心跳断开
ERROR_CODE_CALLING_CONFLICT_ERROR	- 2201	调用冲突
ERROR_CODE_COLLISION_ERROR	- 2202	发生碰撞
ERROR_CODE_NOT_FOLLOW_POSITION_CMD	- 2203	力控模式关节位置与指令发生滞后
ERROR_CODE_NOT_FOLLOW_TCP_CMD	- 2204	力控模式 TCP 位置与指令发生滞后
ERROR_CODE_NOT_ALL_AT_OP_STATE	- 2205	有关节未进入正常状态
ECODE_OUT_RANGE_FEEDBACK	- 2206	关节角反馈超软限位
ECODE_EMERGENCY_STOP	- 2207	急停已拍下
ECODE_NO_INIT_PARAMETER	- 2208	找不到关节初始参数
ECODE_NOT_MATCH_LOAD	- 2209	负载与理论值不匹配
ERROR_CODE_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_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_OUT_SOFT_RANGE_FEEDBACK	- 2217	超出软限位
ERROR_CODE_CONVEYOR_NOT_ONLINE	- 2218	传送带编码器不在线
ERROR_CODE_CONVEYOR_IS_TRACKED	- 2219	传送带正在被跟踪，不能 moveJ
ERROR_CODE_CONVEYOR_CANNOT_TRACK	- 2220	开启跟踪传送带失败
ERROR_CODE_INPUT_OUT_OF_EXTREME_POSITION_RANGE	- 2221	超出关节极限位置
ERROR_CODE_SLOPOVER_VIRTUAWALL	- 2222	关节越过虚拟墙
ERROR_CODE_SLOPOVER_REDUCE_VIRTUAWALL	- 2223	关节越过减速墙
ERROR_CODE_PLAN_ERROR	- 2301	路径规划失败
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	-	零空间运动至关

	2342	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_PARAMETER_POINTER_EQUALS_NULLPTR	- 2901	输入参数为空
ERROR_CODE_PARAMETER_POINTER_EQUALS_NAN_OR_INF	- 2902	输入参数存在 nan 或者 inf
ERROR_CODE_ENTER_FORCE_MODE_ERROR	- 2903	进入力控模式失败
ERROR_CODE_CANNOT_SET_VELOCITY_PERCENT_VALUE	- 2904	设置速度百分比失败
ERROR_CODE_INPUT_OUT_OF_PHYSICAL_POSITION_RANGE	- 2905	输入参数超出物理极限位置
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

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。

附录 B：如何确保运动学逆解唯一

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()
```