

# HansRobot Library

## Python

(Elfin 系列机器人通用)



## 修订记录

文件编号	HR-LIB-V1.0.0.3		
文件状态	[ ]草稿 [√] 正式发布 [ ]正在修改		
当前版本	V1.0.0.3		
拟 制	大族机器人系统软件组	日期	2022 年 05 月 12 日
审 核	大族机器人系统测试组	日期	2022 年 05 月 12 日
批 准	大族机器人	日期	2022 年 05 月 12 日
发布日期	2022 年 05 月 12 日		
生效日期	2022 年 05 月 12 日		

A - 增加 M - 修订 D - 删除

版本编号	版本日期	支持的控制器版本	更新说明
V1.0.0.0	2022.05.12	20220314.180	A 创建文件
V1.0.0.1	2022.07.07	20220624.203	M 新增接口
V1.0.0.2	2022.08.05	20220720.207	M 新增接口
V1.0.0.3	2022.08.28	20220825.214	M 新增 HRIF_SetToolMotion 接口

## 目录

<b>第一章 概述 .....</b>	<b>1</b>
1.1. 编写目的 .....	1
1.2. 名词解析 .....	1
1.3. 数据类型 .....	1
<b>第二章 环境搭建 .....</b>	<b>2</b>
2.1. Windows 环境搭建 .....	2
2.1.1. 创建项目 .....	2
2.1.2. 引用 SDK .....	2
2.1.3. 版本选择 .....	2
2.1.4. 运行程序 .....	3
2.2. Linux 环境搭建 .....	3
2.2.1. 上传 SDK .....	3
2.2.2. 引用 SDK .....	3
2.2.3. 版本选择 .....	4
2.2.4. 运行程序 .....	4
<b>第三章 可使用接口 .....</b>	<b>5</b>
3.1. 初始化 .....	5
3.1.1. HRIF_Connect .....	5
3.1.2. HRIF_DisConnect .....	6
3.1.3. HRIF_IsConnected .....	7
3.1.4. HRIF_ShutdownRobot .....	8
3.1.5. HRIF_Connect2Box .....	9
3.1.6. HRIF_Electrify .....	10
3.1.7. HRIF_Blackout .....	11
3.1.8. HRIF_Connect2Controller .....	12
3.1.9. HRIF_IsSimulateRobot .....	13
3.1.10. HRIF_IsControllerStarted .....	14
3.1.11. HRIF_ReadVersion .....	15

3.1.12.	HRIF_ReadRobotModel .....	16
3.2.	轴组控制指令 .....	17
3.2.1.	HRIF_GrpEnable .....	17
3.2.2.	HRIF_GrpDisable .....	18
3.2.3.	HRIF_GrpReset .....	19
3.2.4.	HRIF_GrpStop .....	20
3.2.5.	HRIF_Grpinterrupt .....	21
3.2.6.	HRIF_GrpContinue .....	22
3.2.7.	HRIF_GrpCloseFreeDriver .....	23
3.2.8.	HRIF_GrpOpenFreeDriver .....	24
3.3.	脚本控制指令 .....	25
3.3.1.	HRIF_RunFunc .....	25
3.3.2.	HRIF_StartScript .....	26
3.3.3.	HRIF_StopScript .....	27
3.3.4.	HRIF_PauseScript .....	28
3.3.5.	HRIF_ContinueScript .....	29
3.4.	电箱控制指令 .....	30
3.4.1.	HRIF_ReadBoxInfo .....	30
3.4.2.	HRIF_ReadBoxCI .....	32
3.4.3.	HRIF_ReadBoxDI .....	33
3.4.4.	HRIF_ReadBoxCO .....	34
3.4.5.	HRIF_ReadBoxDO .....	35
3.4.6.	HRIF_ReadBoxAI .....	36
3.4.7.	HRIF_ReadBoxAO .....	37
3.4.8.	HRIF_SetBoxCO .....	38
3.4.9.	HRIF_SetBoxDO .....	39
3.4.10.	HRIF_SetBoxAOMode .....	40
3.4.11.	HRIF_SetBoxAOVal .....	41
3.4.12.	HRIF_SetEndDO .....	42
3.4.13.	HRIF_ReadEndDI .....	43
3.4.14.	HRIF_ReadEndDO .....	44
3.4.15.	HRIF_ReadEndAI .....	45
3.4.16.	HRIF_ReadEndBTN .....	46
3.5.	状态读取与设置指令 .....	47
3.5.1.	HRIF_SetOverride .....	47
3.5.2.	HRIF_SetToolMotion .....	48
3.5.3.	HRIF_SetPayload .....	49
3.5.4.	HRIF_SetJointMaxVel .....	50
3.5.5.	HRIF_SetJointMaxAcc .....	51

3.5.6.	HRIF_SetLinearMaxVel .....	52
3.5.7.	HRIF_SetLinearMaxAcc .....	53
3.5.8.	HRIF_SetMaxAcsRange.....	54
3.5.9.	HRIF_SetMaxPcsRange .....	56
3.5.10.	HRIF_ReadJointMaxVel.....	57
3.5.11.	HRIF_ReadJointMaxAcc.....	58
3.5.12.	HRIF_ReadJointMaxJerk .....	59
3.5.13.	HRIF_ReadLinearMaxSpeed.....	60
3.5.14.	HRIF_ReadEmergencyInfo .....	61
3.5.15.	HRIF_ReadRobotState .....	62
3.5.16.	HRIF_ReadRobotFlags.....	64
3.5.17.	HRIF_ReadCurWaypointID.....	66
3.5.18.	HRIF_ReadAxisErrorCode.....	67
3.5.19.	HRIF_ReadCurFSM .....	68
3.5.20.	HRIF_ReadCurFSMFromCPS.....	69
3.6.	位置、速度、电流读取指令 .....	70
3.6.1.	HRIF_ReadActPos.....	70
3.6.2.	HRIF_ReadCmdJointPos .....	72
3.6.3.	HRIF_ReadActJointPos .....	73
3.6.4.	HRIF_ReadCmdTcpPos.....	74
3.6.5.	HRIF_ReadActTcpPos.....	75
3.6.6.	HRIF_ReadCmdJointVel .....	76
3.6.7.	HRIF_ReadActJointVel .....	77
3.6.8.	HRIF_ReadCmdTcpVel .....	78
3.6.9.	HRIF_ReadActTcpVel .....	79
3.6.10.	HRIF_ReadCmdJointCur.....	80
3.6.11.	HRIF_ReadActJointCur.....	81
3.6.12.	HRIF_ReadTcpVelocity .....	82
3.7.	坐标转换计算指令 .....	83
3.7.1.	HRIF_Quaternion2RPY .....	83
3.7.2.	HRIF_RPY2Quaternion.....	85
3.7.3.	HRIF_GetInverseKin.....	87
3.7.4.	HRIF_GetForwardKin .....	90
3.7.5.	HRIF_Base2UcsTcp .....	92
3.7.6.	HRIF_UcsTcp2Base .....	94
3.7.7.	HRIF_PoseAdd.....	96
3.7.8.	HRIF_PoseSub.....	98
3.7.9.	HRIF_PoseTrans.....	100
3.7.10.	HRIF_PoseInverse .....	102
3.7.11.	HRIF_PoseDist .....	104

3.7.12.	HRIF_Poseinterpolate .....	106
3.7.13.	HRIF_PoseDefdFrame .....	108
3.8.	工具坐标与用户坐标读写指令 .....	111
3.8.1.	HRIF_SetTCP .....	111
3.8.2.	HRIF_SetUCS .....	112
3.8.3.	HRIF_ReadCurTCP .....	113
3.8.4.	HRIF_ReadCurUCS .....	114
3.8.5.	HRIF_SetTCPByName .....	115
3.8.6.	HRIF_SetUCSByName .....	116
3.8.7.	HRIF_ReadTCPByName .....	117
3.8.8.	HRIF_ReadUCSByName .....	119
3.9.	力控控制指令 .....	121
3.9.1.	HRIF_SetForceControlState .....	121
3.9.2.	HRIF_ReadForceControlState .....	122
3.9.3.	HRIF_SetForceToolCoordinateMotion .....	123
3.9.4.	HRIF_ForceControlinterrupt .....	124
3.9.5.	HRIF_ForceControlContinue .....	125
3.9.6.	HRIF_SetForceZero .....	126
3.9.7.	HRIF_SetMaxSearchVelocities .....	127
3.9.8.	HRIF_SetControlFreedom .....	128
3.9.9.	HRIF_SetForceControlStrategy .....	129
3.9.10.	HRIF_SetFreeDrivePositionAndOrientation .....	130
3.9.11.	HRIF_SetPIDControlParams .....	131
3.9.12.	HRIF_SetMassParams .....	132
3.9.13.	HRIF_SetDampParams .....	133
3.9.14.	HRIF_SetStiffParams .....	134
3.9.15.	HRIF_SetForceControlGoal .....	135
3.9.16.	HRIF_SetControlGoal .....	136
3.9.17.	HRIF_SetForceDataLimit .....	138
3.9.18.	HRIF_SetForceDistanceLimit .....	140
3.9.19.	HRIF_SetForceFreeDriveMode .....	141
3.9.20.	HRIF_ReadFTCabData .....	142
3.10.	通用运动类控制指令 .....	143
3.10.1.	HRIF_MoveRelJ .....	143
3.10.2.	HRIF_MoveRelL .....	144
3.10.3.	HRIF_WayPointRel .....	145
3.10.4.	HRIF_IsMotionDone .....	149
3.10.5.	HRIF_IsBlendingDone .....	150
3.10.6.	HRIF_WayPointEx .....	151
3.10.7.	HRIF_WayPoint .....	155

3.10.8.	HRIF_WayPoint2.....	159
3.10.9.	HRIF_MoveJ .....	163
3.10.10.	HRIF_MoveL .....	166
3.10.11.	HRIF_MoveC .....	169
3.10.12.	HRIF_MoveZ .....	172
3.10.13.	HRIF_MoveE .....	176
3.10.14.	HRIF_MoveS.....	180
3.11.	连续轨迹运动类控制指令 .....	182
3.11.1.	HRIF_StartPushMovePathJ .....	182
3.11.2.	HRIF_PushMovePathJ.....	183
3.11.3.	HRIF_EndPushMovePathJ .....	184
3.11.4.	HRIF_MovePathJ .....	185
3.11.5.	HRIF_ReadMovePathJState .....	186
3.11.6.	HRIF_UpdateMovePathJName .....	188
3.11.7.	HRIF_DelMovePathJ.....	189
3.11.8.	HRIF_ReadTrackProcess.....	190
3.11.9.	HRIF_InitMovePathL .....	191
3.11.10.	HRIF_PushMovePathL.....	193
3.11.11.	HRIF_PushMovePaths .....	194
3.11.12.	HRIF_MovePathL .....	196
3.11.13.	HRIF_SetMovePathOverride.....	197
3.12.	Servo 运动类控制指令.....	198
3.12.1.	HRIF_StartServo .....	198
3.12.2.	HRIF_PushServoJ.....	199
3.12.3.	HRIF_PushServoP .....	200
3.13.	相对跟踪运动类控制指令 .....	202
3.13.1.	HRIF_SetMoveTraceParams .....	202
3.13.2.	HRIF_SetMoveTraceInitParams.....	203
3.13.3.	HRIF_SetMoveTraceUcs.....	204
3.13.4.	HRIF_SetTrackingState .....	205
3.14.	其他指令 .....	206
3.14.1.	HRIF_HRAppCmd .....	206
3.14.2.	HRIF_WriteEndHoldingRegisters .....	207
3.14.3.	HRIF_ReadEndHoldingRegisters.....	209
<b>第四章 附录 .....</b>		<b>211</b>
4.1.	参考文件.....	211

## 第一章 概述

### 1.1. 编写目的

- 1.1. 为使用 Hans Python 接口的人员提供接口使用说明
- 1.2. 为研发人员进行接口编写和维护提供参考文件。

### 1.2. 名词解析

专用名称	解释
ACS	关节坐标, 单位为度(°)
PCS	空间坐标, 单位为毫米( mm )、度(°)
TCP	系统默认的工具坐标系, 初始值为 0,0,0,0,0,0
UCS	用户坐标系
Base	系统默认的用户坐标系, 初始值为 0,0,0,0,0,0
Tool	系统默认的工具坐标系

### 1.3. 数据类型

数据类型	长度	说明
bool	-	布尔型
int	-	整型
float	-	浮点型
string	-	字符串
list		数组

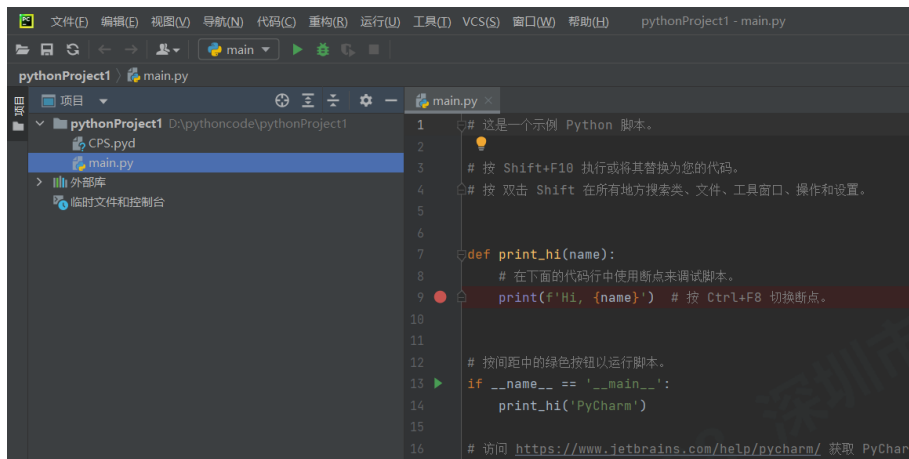


## 第二章 环境搭建

### 2.1. Windows 环境搭建

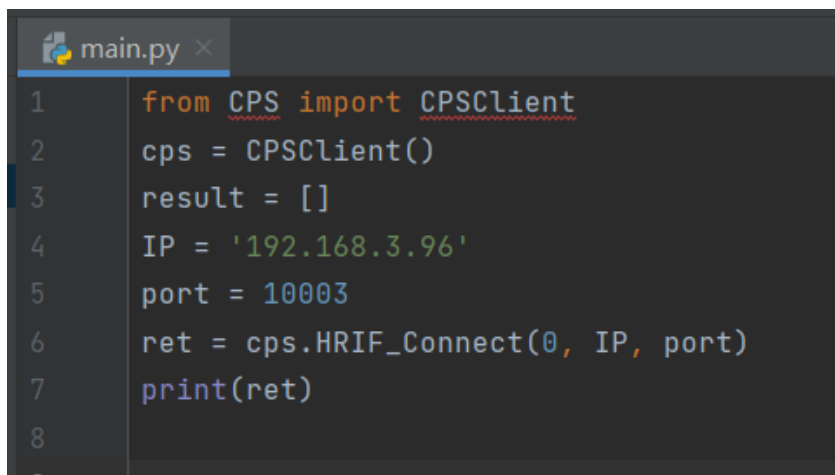
#### 2.1.1. 创建项目

将包体更名为 CPS.pyd，复制到一个空目录下，创建一个 Python 文件 main.py。（以 PyCharm 为例）



#### 2.1.2. 引用 SDK

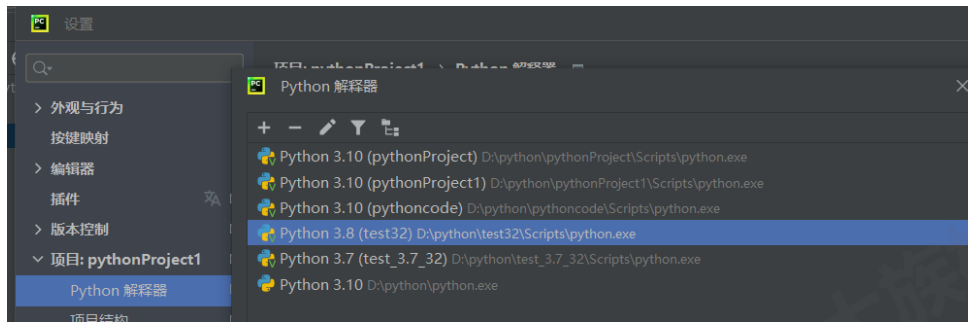
在 main.py 内引入 CPSCClient 对象，尝试连接。



#### 2.1.3. 版本选择

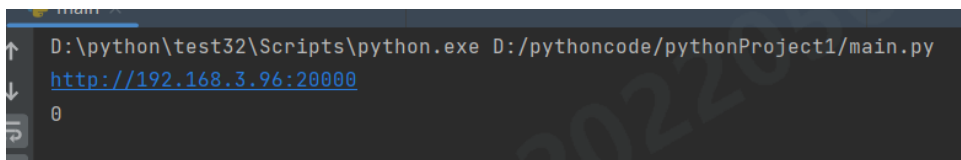
根据所用包体，选择对应版本的编译器，如原包体名为 CPS\_3.8.9\_win\_32.pyd 即表示应选择 32 位 Python

3.8 版本。



## 2.1.4. 运行程序

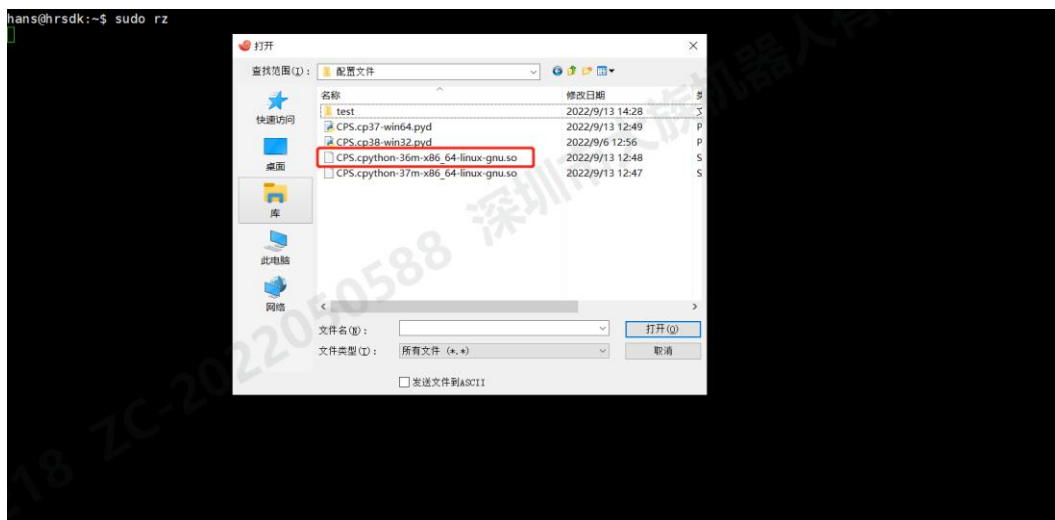
右键单击工作区，选择 run code，查看下方终端打印



## 2.2. Linux 环境搭建

### 2.2.1. 上传 SDK

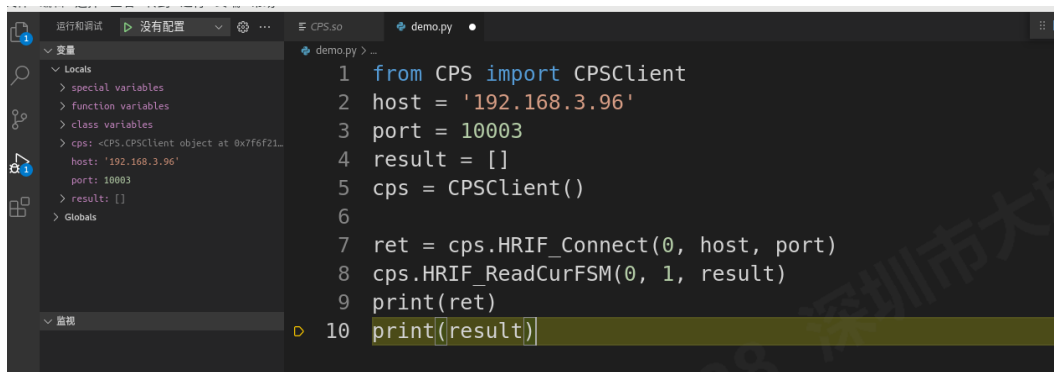
上传 SDK 文件至系统中（以 vscode 为例）



### 2.2.2. 引用 SDK

新建一个文件夹，将 so 文件放置于其中，通过 vscode 打开该文件夹。在其中新建一个 demo.py 文件，引

入 CPSClnt 对象，尝试连接。



```

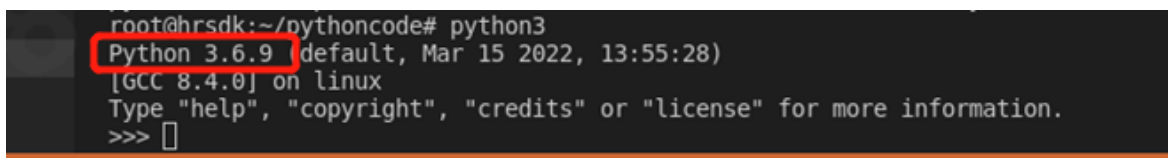
1 from CPS import CPSClnt
2 host = '192.168.3.96'
3 port = 10003
4 result = []
5 cps = CPSClnt()
6
7 ret = cps.HRIF_Connect(0, host, port)
8 cps.HRIF_ReadCurFSM(0, 1, result)
9 print(ret)
10 print(result)
  
```

Locals

- > special variables
- > function variables
- > class variables
- > cps: <CPS.CPSClnt object at 0x7f6f21...
- host: '192.168.3.96'
- port: 10003
- > result: []
- > Globals

### 2.2.3. 版本选择

根据所用 SDK，选择对应版本的编译器，如 SDK 名为 CPS\_python3\_Linux.so 即表示应选择 Python 3 任一版本。

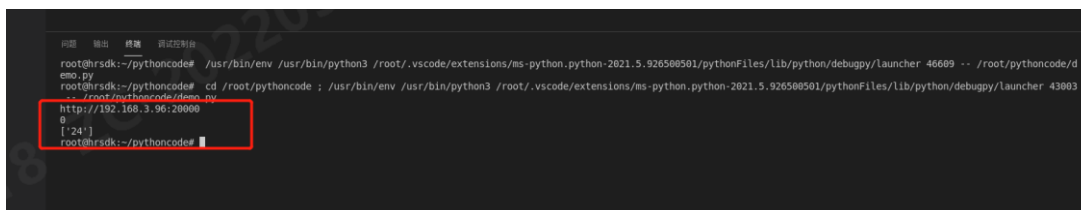


```

root@hrsdk:~/pythoncode# python3
Python 3.6.9 (default, Mar 15 2022, 13:55:28)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
  
```

### 2.2.4. 运行程序

运行程序，查看终端打印。



```

root@hrsdk:~/pythoncode# /usr/bin/env /usr/bin/python3 /root/.vscode/extensions/ms-python.python-2021.5.926580501/pythonFiles/lib/python/debugpy/launcher 46689 .. /root/pythoncode/d
emo.py
root@hrsdk:~/pythoncode# cd /root/pythoncode ; /usr/bin/env /usr/bin/python3 /root/.vscode/extensions/ms-python.python-2021.5.926580501/pythonFiles/lib/python/debugpy/launcher 43803
.. /root/pythoncode/demo.py
http://192.168.3.96:20000
[]
root@hrsdk:~/pythoncode#
  
```

## 第三章 可使用接口

### 3.1. 初始化

#### 3.1.1. HRIF\_Connect

3.1.1.1. 描述：连接机器人服务器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
IP	控制器 IP 地址	string	-	控制器 IP 地址， 根据实际设置的 IP 地址定义
nPort	控制器端口	int		控制器端口，一般为 10003

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：函数调用成功 nRet>0: 返回调用失败的错误码

3.1.1.2. 示例

# 导入 Python SDK

```
from CPS import CPSClient
```

# 定义 IP, 端口

```
IP = '192.168.0.10'
```

```
nPort = 10003
```

# 创建对象

```
cps = CPSClient()
```

# 连接控制器

```
nRet = cps.HRIF_Connect(0,IP,nPort)
```

### 3.1.2. HRIF\_DisConnect

3.1.2.1. 描述：断开连接机器人服务器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0 : 返回函数调用成功 nRet>0 : 返回调用失败的错误码

3.1.2.2. 示例

# 断开连接机器人服务器

```
nRet = cps.HRIF_DisConnect(0)
```

### 3.1.3. HRIF\_IsConnected

3.1.3.1. 描述：判断控制器是否连接。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
bRet	返回值	bool	False,True	False : 控制器未连接 True : 控制器已连接

3.1.3.2. 示例

# 控制器是否连接

```
nRet = cps.HRIF_IsConnected(0)
```

### 3.1.4. HRIF\_ShutdownRobot

3.1.4.1. 描述：控制器断电（断开机器人供电，系统关机）。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.4.2. 示例

# 控制器断电

```
nRet = cps.HRIF_ShutdownRobot(0)
```

### 3.1.5. HRIF\_Connect2Box

3.1.5.1. 描述：连接控制器电箱。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.5.2. 示例

# 连接控制器电箱

```
nRet = cps.HRIF_Connect2Box(0)
```



### 3.1.6. HRIF\_Electrify

3.1.6.1. 描述：机器人上电。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.6.2. 示例

# 机器人上电

```
nRet = cps.HRIF_Electrify(0)
```

### 3.1.7. HRIF\_Blackout

3.1.7.1. 描述：机器人断电。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.7.2. 示例

# 机器人断电

```
nRet = cps.HRIF_Blackout(0)
```

### 3.1.8. HRIF\_Connect2Controller

3.1.8.1. 描述：连接控制器，连接过程中会启动主站，初始化从站，配置参数，检查配置，完成后跳转到去使能状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.8.2. 示例

# 连接机器人

```
nRet = cps.HRIF_Connect2Controller(0)
```

### 3.1.9. HRIF\_IsSimulateRobot

3.1.9.1. 描述：是否为模拟机器人。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nSimulateRobot	是否模拟机器人	int	0/1	0：真实机器人 1：模拟机器人

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.9.2. 示例

# 定义是否是模拟机器人

nSimulateRobot = 0

# 判断是否是模拟机器人

nRet = cps.HRIF\_IsSimulateRobot(0, nSimulateRobot)

### 3.1.10. HRIF\_IsControllerStarted

3.1.10.1. 描述：控制器是否启动完成。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	是否启动完成	string	0/1	0: 未启动 1: 已启动

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.10.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 判断机器人是否启动完成

```
nRet = cps.HRIF_IsControllerStarted(0, result)
```

### 3.1.11. HRIF\_ReadVersion

3.1.11.1. 描述：读取控制器版本号。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	CPS 版本	string	-	CPS 版本号
result[1]	控制器版本	string	-	控制器版本号
result[2]	电箱版本	string	0~4	电箱版本号: 0: 模拟电箱 1~4: 电箱版本号
result[3]	控制板固件版本	string	-	控制板固件版本
result[4]	控制板固件版本	string	-	控制板固件版本
result[5]	算法版本	string	-	算法版本
result[6]	固件版本	string	-	固件版本

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.1.11.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 获取版本信息

```
nRet = cps.HRIF_ReadVersion(0,0, result)
```

### 3.1.12. HRIF\_ReadRobotModel

3.1.12.1. 描述：读取机器人类型。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	机器人类型	string	-	机器人类型

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.1.12.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取机器人类型

```
nRet = cps.HRIF_ReadRobotModel (0,0, result)
```

## 3.2. 轴组控制指令

### 3.2.1. HRIF\_GrpEnable

3.2.1.1. 描述：机器人使能命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.1.2. 示例

# 机器人使能

```
nRet = cps.HRIF_GrpEnable(0,0)
```



### 3.2.2. HRIF\_GrpDisable

3.2.2.1. 描述：机器人去使能命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.2.2. 示例

# 机器人去使能

```
nRet = cps.HRIF_GrpDisable(0,0)
```

### 3.2.3. HRIF\_GrpReset

3.2.3.1. 描述：机器人复位命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.3.2. 示例

# 机器人复位

```
nRet = cps.HRIF_GrpReset(0,0)
```

### 3.2.4. HRIF\_GrpStop

3.2.4.1. 描述：机器人停止运动命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.4.2. 示例

# 机器人停止运动

```
nRet = cps.HRIF_GrpStop(0,0)
```

### 3.2.5. HRIF\_Grpinterrupt

3.2.5.1. 描述：机器人暂停运动命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.5.2. 示例

# 机器人暂停运动

```
nRet = cps.HRIF_Grpinterrupt(0,0)
```

### 3.2.6. HRIF\_GrpContinue

3.2.6.1. 描述：机器人继续运动命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.6.2. 示例

# 机器人继续运动

```
nRet = cps.HRIF_GrpContinue(0,0)
```

### 3.2.7. HRIF\_GrpCloseFreeDriver

3.2.7.1. 描述：关闭自由驱动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.7.2. 示例

# 机器人关闭自由驱动

```
nRet = cps.HRIF_GrpCloseFreeDriver(0,0)
```

### 3.2.8. HRIF\_GrpOpenFreeDriver

3.2.8.1. 描述：打开自由驱动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.2.8.2. 示例

# 机器人打开自由驱动

```
nRet = cps.HRIF_GrpOpenFreeDriver(0,0)
```

### 3.3. 脚本控制指令

#### 3.3.1. HRIF\_RunFunc

3.3.1.1. 描述：运行指定脚本函数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
strFuncName	函数名称	string	-	指定脚本函数名称，对应示教器界面的函数
param	参数表	string	-	参数列表
result	返回值	list		传入空列表，result = [ ]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.3.1.2. 示例

```
# 指定运行函数 Func_1

strFuncName = "Func_1"

# 定义输入参数

param = [1,2,3]

# 定义返回值空列表

result = [ ]

# 运行函数 Func_1

nRet = cps.HRIF_RunFunc(0,0, strFuncName, param,result)
```



### 3.3.2. HRIF\_StartScript

3.3.2.1. 描述：执行脚本 Main 函数，调用后执行示教器页面编译好的脚本文件 Func\_main 函数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.3.2.2. 示例

# 运行脚本 Main 函数

```
nRet = cps.HRIF_StartScript(0)
```

### 3.3.3. HRIF\_StopScript

3.3.3.1. 描述：停止脚本，调用后停止示教器页面正在执行脚本文件。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.3.3.2. 示例

# 停止当前正在运行的脚本

```
nRet = cps.HRIF_StopScript(0,0)
```

### 3.3.4. HRIF\_PauseScript

3.3.4.1. 描述：暂停脚本，调用后暂停示教器页面正在执行脚本文件。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.3.4.2. 示例

# 暂停当前运行的脚本

```
nRet = cps.HRIF_PauseScript(0)
```

### 3.3.5. HRIF\_ContinueScript

3.3.5.1. 描述: 继续脚本, 调用后继续运行示教器页面已暂停脚本文件, 不处于暂停状态则返回 20018 错误。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.3.5.2. 示例

# 继续运行当前暂停的脚本

```
nRet = cps.HRIF_ContinueScript(0)
```

### 3.4. 电箱控制指令

#### 3.4.1. HRIF\_ReadBoxInfo

3.4.1.1. 描述：读取电箱信息。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	电箱连接状态	string	0/1	0：未连接 1：已连接
result[1]	48V 电压状态	string	0/1	0：电压<36V 1：电压>36V
result[2]	48V 输出电压值	string	0~53	48V 输出电压值
result[3]	48V 输出电流值	string	-	48V 输出电流值
result[4]	远程急停状态	string	0/1	0：无信号 1：有信号
result[5]	三段按钮状态	string	0/1	0：无信号 1：有信号

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.1.2. 示例

# 定义返回值空列表

```
result = [ ]
```

---

# 读取电箱信息

nRet = cps.HRIF\_ReadBoxInfo(0,0, result)

### 3.4.2. HRIF\_ReadBoxCI

3.4.2.1. 描述：读取电箱控制数字输入状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	控制数字输入位	int	0~7	控制数字输入位索引

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	控制数字输入对应位状态	string	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.2.2. 示例

# 定义需要读取的位

nBit = 0

# 定义返回值空列表

result = [ ]

# 读取电箱 nBit 位 CI 状态

nRet = cps.HRIF\_ReadBoxCI (0,nBit,result)

### 3.4.3. HRIF\_ReadBoxDI

3.4.3.1. 描述：读取电箱通用数字输入状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	通用数字输入位	int	0~7	通用数字输入位索引

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	通用数字输入对应位状态	string	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.3.2. 示例

# 定义需要读取的位

nBit = 0

# 定义返回值空列表

result = [ ]

# 读取电箱 nBit 位 DI 状态

nRet = cps.HRIF\_ReadBoxDI (0, nBit,result)



### 3.4.4. HRIF\_ReadBoxCO

3.4.4.1. 描述：读取电箱控制数字输出状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	控制数字输出位	int	0~7	控制数字输出位索引

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	控制数字输出对应位状态	int	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.4.2. 示例

# 定义需要读取的位

nBit = 0

# 定义返回值空列表

result = [ ]

# 读取电箱 nBit 位 CO 状态

nRet = cps.HRIF\_ReadBoxCO (0, nBit, result)

### 3.4.5. HRIF\_ReadBoxDO

3.4.5.1. 描述：读取电箱通用数字输出状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	通用数字输出位	int	0~7	通用数字输出位索引

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	通用数字输出对应位状态	int	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.5.2. 示例

# 定义需要读取的位

nBit = 0

# 定义返回值空列表

result = [ ]

# 读取电箱 nBit 位 DO 状态

nRet = cps.HRIF\_ReadBoxDO (0, nBit, result)

### 3.4.6. HRIF\_ReadBoxAI

3.4.6.1. 描述：读取电箱模拟量输入值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	模拟量输入通道	int	0/1	0：模拟量通道 0 1：模拟量通道 1

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	对应的模拟量 通道值	float	0~20	电流模式:4-20mA 电压模式:0-10V

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.6.2. 示例

# 定义需要读取的位

nBit = 0

# 定义返回值空列表

result = [ ]

# 读取电箱 nBit 通道 AI 值

nRet = cps.HRIF\_ReadBoxAI (0, nBit, result)

### 3.4.7. HRIF\_ReadBoxAO

3.4.7.1. 描述：读取电箱模拟量输出值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	模拟量输出通道	int	0/1	0：模拟量通道 0 1：模拟量通道 1

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	对应的模拟量通道模式	string	0/1	0：电压模式 1：电流模式:
result[1]	对应的模拟量通道值	string	0~20	电流模式:4-20mA 电压模式:0-10V

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.7.2. 示例

# 定义需要读取的位

nBit = 0

# 定义返回值空列表

result = [ ]

# 读取电箱 nBit 通道 AO 值

nRet = cps.HRIF\_ReadBoxAO(0, nBit, result)

### 3.4.8. HRIF\_SetBoxCO

3.4.8.1. 描述：设置电箱控制数字输出状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
nBit	控制数字输出位	int	0~7	控制数字输出位索引
nVal	控制数字输出状态	int	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.8.2. 示例

# 定义设置状态

nVal = 0

# 定义需要设置的位

nBit = 0

# 设置电箱第 *nBit* 位 CO 状态为 *nVal*

nRet = cps.HRIF\_SetBoxCO (0, nBit, nVal)

### 3.4.9. HRIF\_SetBoxDO

3.4.9.1. 描述：设置电箱通用数字输出状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
nBit	通用数字输出位	int	0~7	通用数字输出位索引
nVal	通用数字输出状态	int	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.9.2. 示例

# 定义设置状态

nVal = 0

# 定义需要设置的位

nBit = 0

# 设置电箱第 nBit 位 DO 状态为 nVal

nRet = cps.HRIF\_SetBoxDO(0, nBit, nVal)

### 3.4.10. HRIF\_SetBoxAOMode

3.4.10.1. 描述：设置电箱模拟量输出模式。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
nBit	模拟量通道	int	0/1	模拟量通道
nVal	模式	int	1/2	1：电压模式 2：电流模式

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.10.2. 示例

# 定义设置值

nVal = 0

# 定义需要设置的通道

nBit = 0

# 设置电箱第 nBit 通道 AO 模式为 nVal

nRet = cps.HRIF\_SetBoxAOMode (0,nBit , nVal)

### 3.4.11. HRIF\_SetBoxAOVal

3.4.11.1. 描述：设置电箱模拟量输出值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
nBit	模拟量通道	int	0/1	模拟量通道
dVal	对应模拟量值	float	0~20	电流模式:4-20mA 电压模式:0-10V
nMode	模式	int	1/2	1：电压模式 2：电流模式

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.11.2. 示例

# 定义设置值

dVal = 0

# 定义需要设置的通道

nBit = 0

# 读取的当前模式

nMode = 0

# 设置电箱第 nBit 通道 AO 值为 dVal

nRet = cps.HRIF\_SetBoxAOVal(0,nBit, dVal, nMode)



### 3.4.12. HRIF\_SetEndDO

3.4.12.1. 描述：设置末端数字输出状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nBit	末端数字输出位	int	0~3	通用数字输出位索引，末端只有 4 个 DO
nVal	末端数字输出状态	int	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.12.2. 示例

# 定义设置值

nVal = 0

# 定义需要设置的位

nBit = 0

# 设置末端 nBit 位 DO 状态

nRet = cps.HRIF\_SetEndDO(0,0,nBit, nVal)

### 3.4.13. HRIF\_ReadEndDI

3.4.13.1. 描述：读取末端数字输入状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nBit	末端数字输入位	int	0~2	末端数字输入对应位状态
result	返回值	list		传入空列表，result = []

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
reuslt[0]	末端数字输入对应位状态	string	0/1	0：低电平 1：高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.13.2. 示例

# 定义返回值空列表

```
result = []
```

# 定义需要读取的位

```
nBit = 0
```

# 读取末端 nBit 位 DI 状态

```
nRet = cps.HRIF_ReadEndDI(0,0, nBit, result)
```

### 3.4.14. HRIF\_ReadEndDO

3.4.14.1. 描述：读取末端数字输出状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	末端数字输出位	int	0~2	末端数字输出对应位索引

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	末端数字输出对应位状态	string	0/1	0: 低电平 1: 高电平

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.14.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要读取的位

```
nBit = 0
```

# 读取末端 nBit 位 DO 状态

```
nRet = cps.HRIF_ReadEndDO(0,0, nBit, result)
```

### 3.4.15. HRIF\_ReadEndAI

3.4.15.1. 描述：读取末端模拟量输入值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nBit	模拟量输入通道	int	0/1	0：模拟量通道 0 1：模拟量通道 1

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	对应的模拟量通道值	string	0~20	电流模式:4~20mA 电压模式:0-10V

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.15.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要读取的位

```
nBit = 0
```

# 读取末端 nBit 通道 AI 值

```
nRet = cps.HRIF_ReadEndAI(0,0, nBit, result)
```

### 3.4.16. HRIF\_ReadEndBTN

3.4.16.1. 描述：读取末端按键状态，根据搭载的末端类型，各状态表示含义会有区别。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	按键 1 状态	string	0/1	0: 松开 1: 按下
result[1]	按键 2 状态	string	0/1	0: 松开 1: 按下
result[2]	按键 3 状态	string	0/1	0: 松开 1: 按下
result[3]	按键 4 状态	string	0/1	0: 松开 1: 按下

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.4.16.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取末端 nBit 通道 AI 值

```
nRet = cps.HRIF_ReadEndBTN(0,0, result)
```

## 3.5. 状态读取与设置指令

### 3.5.1. HRIF\_SetOverride

3.5.1.1. 描述：设置速度比。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dOverride	速度比	float	0.01~1	需要设置的速度比(0.01~1)

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.1.2. 示例

# 需要设置的速度比

dOverride = 0.5

# 设置当前速度比

nRet = cps.HRIF\_SetOverride(0,0, dOverride)

### 3.5.2. HRIF\_SetToolMotion

3.5.2.1. 描述：开启或关闭 Tool 坐标系运动模式。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nState	状态	int	0/1	0：开启 1：关闭

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.2.2. 示例

# 需要设置的 Tool 运动状态

nState = 1

# 设置 Tool 运动状态

nRet = cps.HRIF\_SetToolMotion(0,0,nState)

### 3.5.3. HRIF\_SetPayload

3.5.3.1. 描述：设置当前负载参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值=0
dMass	质量	float	0~允许最大负载	质量, 范围为 0 到当前允许最大负载, 单位[kg]
dX	质心 X 方向偏移	float	>0	质心 X 方向偏移,单位[mm]
dY	质心 Y 方向偏移	float	>0	质心 Y 方向偏移, 单位[mm]
dZ	质心 Z 方向偏移	float	>0	质心 Z 方向偏移单位[mm]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0 : 返回函数调用成功 nRet>0 : 返回调用失败的错误码

3.5.3.2. 示例

# 需要设置负载质量

dMass = 0.5

# 需要设置负载质心 X 方向偏移量

dX= 0.5

# 需要设置负载质心 Y 方向偏移量

dY= 0.5

# 需要设置负载质心 Z 方向偏移量

dZ= 0.5

# 设置当前负载

nRet = cps.HRIF\_SetPayload(0,0,dMass, dX, dY, dZ)



### 3.5.4. HRIF\_SetJointMaxVel

3.5.4.1. 描述：设置关节最大运动速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dJ1	J1 轴最大速度	float	0~200	J1 轴最大速度，单位[°/s]
dJ2	J2 轴最大速度	float	0~200	J2 轴最大速度，单位[°/s]
dJ3	J3 轴最大速度	float	0~200	J3 轴最大速度，单位[°/s]
dJ4	J4 轴最大速度	float	0~200	J4 轴最大速度，单位[°/s]
dJ5	J5 轴最大速度	float	0~200	J5 轴最大速度，单位[°/s]
dJ6	J6 轴最大速度	float	0~200	J6 轴最大速度，单位[°/s]

\* 注：关节速度有效范围需要参考具体机型。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.4.2. 示例

# 定义需要设置关节最大速度

```
Joint = [180, 180, 180, 180, 180, 180]
```

# 设置最大关节速度

```
nRet = cps.HRIF_SetJointMaxVel(0,0,Joint)
```

### 3.5.5. HRIF\_SetJointMaxAcc

3.5.5.1. 描述：设置关节最大运动加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dJ1	J1 轴最大加速度	float	0~720	J1 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ2	J2 轴最大加速度	float	0~720	J2 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ3	J3 轴最大加速度	float	0~720	J3 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ4	J4 轴最大加速度	float	0~720	J4 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ5	J5 轴最大加速度	float	0~720	J5 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ6	J6 轴最大加速度	float	0~720	J6 轴最大加速度，单位[°/s <sup>2</sup> ]

\* 注：关节加速度有效范围需要参考具体机型。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.5.2. 示例

# 定义需要设置的关节最大加速度

```
JointAcc = [20, 20, 20, 20, 20, 20,]
```

# 设置最大关节加速度

```
nRet = cps.HRIF_SetJointMaxAcc(0,0,JointAcc)
```

### 3.5.6. HRIF\_SetLinearMaxVel

3.5.6.1. 描述：设置直线运动最大速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dMaxVel	最大直线速度	float	0~3000	最大直线速度，默认[500]，单位[mm/s]

\* 注：直线速度有效范围需要参考具体机型。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.6.2. 示例

# 定义需要设置直线最大速度

dMaxVel = 100

# 设置最大直线速度

nRet = cps.HRIF\_SetLinearMaxVel(0,0,dMaxVel)

### 3.5.7. HRIF\_SetLinearMaxAcc

3.5.7.1. 描述：设置直线运动最大加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dMaxAcc	最大直线加速度	float	0~2500	最大直线速度，默认[2500]，单位[mm/s <sup>2</sup> ]

\* 注：直线加速度有效范围需要参考具体机型。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.7.2. 示例

# 定义需要设置直线最大速度

dMaxAcc = 2500

# 设置最大直线速度

nRet = cps.HRIF\_SetLinearMaxAcc(0,0,dMaxAcc)

### 3.5.8. HRIF\_SetMaxAcsRange

3.5.8.1. 描述：设置关节最大运动范围。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dMaxJ1- dMaxJ6	关节最大范围	float	-	dMaxJ1:关节 1 最大运动范围，单位[°] dMaxJ2:关节 2 最大运动范围，单位[°] dMaxJ3:关节 3 最大运动范围，单位[°] dMaxJ4:关节 4 最大运动范围，单位[°] dMaxJ5:关节 5 最大运动范围，单位[°] dMaxJ6:关节 6 最大运动范围，单位[°]
dMinJ1- dMinJ6	关节最小范围	float	-	dMinJ1:关节 1 最小运动范围，单位[°] dMinJ2:关节 2 最小运动范围，单位[°] dMinJ3:关节 3 最小运动范围，单位[°] dMinJ4:关节 4 最小运动范围，单位[°] dMinJ5:关节 5 最小运动范围，单位[°] dMinJ6:关节 6 最小运动范围，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.8.2. 示例

# 定义需要设置的关节最大运动范围

dMax = [360, 360, 360, 360, 360, 360]

# 定义需要设置的关节最小运动范围

dMin = [-360, -360, -360, -360, -360, -360]

# 设置关节运动范围

---

nRet = cps.HRIF\_SetMaxAcsRange (0,0,dMax, dMin)

### 3.5.9. HRIF\_SetMaxPcsRange

3.5.9.1. 描述：设置空间最大运动范围。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dMax	XYZ 最大范围	float	-	XYZ 最大范围，单位[mm]
dMin	XYZ 最小范围	float	-	XYZ 最小范围，单位[mm]
dUcs	基于用户坐标	float	-	dX: X 坐标，单位[mm] dY: Y 坐标，单位[mm] dZ: Z 坐标，单位[mm] dRx: Rx 坐标，单位[°] dRy: Ry 坐标，单位[°] dRz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.9.2. 示例

# 定义需要设置的空间最大运动范围

dMax = [360, 360, 360]

# 定义需要设置的空间最小运动范围

dMin = [-360, -360, -360]

# 定义用户坐标变量

dUcs = [0, 0, 0, 0, 0, 0]

# 设置关节运动范围

nRet = cps.HRIF\_SetMaxPcsRange (0,0,dMax, dMin, dUcs)

### 3.5.10. HRIF\_ReadJointMaxVel

3.5.10.1. 描述：读取关节最大运动速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量：

输出变量	名称	数据类型	有效范围	内容
result[0]	J1 轴最大速度	string	0~200	J1 轴最大速度，单位[°/s]
result[1]	J2 轴最大速度	string	0~200	J2 轴最大速度，单位[°/s]
result[2]	J3 轴最大速度	string	0~200	J3 轴最大速度，单位[°/s]
result[3]	J4 轴最大速度	string	0~200	J4 轴最大速度，单位[°/s]
result[4]	J5 轴最大速度	string	0~200	J5 轴最大速度，单位[°/s]
result[5]	J6 轴最大速度	string	0~200	J6 轴最大速度，单位[°/s]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.10.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取最大关节速度

```
nRet = cps.HRIF_ReadJointMaxVel(0,0, result)
```



### 3.5.11. HRIF\_ReadJointMaxAcc

3.5.11.1. 描述：读取关节最大运动加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量：

输出变量	名称	数据类型	有效范围	内容
result[0]	J1 轴最大加速度	string	0~720	J1 轴最大加速度，单位[°/s <sup>2</sup> ]
result[1]	J2 轴最大加速度	string	0~720	J2 轴最大加速度，单位[°/s <sup>2</sup> ]
result[2]	J3 轴最大加速度	string	0~720	J3 轴最大加速度，单位[°/s <sup>2</sup> ]
result[3]	J4 轴最大加速度	string	0~720	J4 轴最大加速度，单位[°/s <sup>2</sup> ]
result[4]	J5 轴最大加速度	string	0~720	J5 轴最大加速度，单位[°/s <sup>2</sup> ]
result[5]	J6 轴最大加速度	string	0~720	J6 轴最大加速度，单位[°/s <sup>2</sup> ]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.11.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取最大关节加速度

```
nRet = cps.HRIF_ReadJointMaxAcc(0,0, result)
```

### 3.5.12. HRIF\_ReadJointMaxJerk

3.5.12.1. 描述：读取关节最大运动加加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量：

输出变量	名称	数据类型	有效范围	内容
result[0]	J1 轴最大加加速度	string	0~720	J1 轴最大加加速度，单位[°/s³]
result[1]	J2 轴最大加加速度	string	0~720	J2 轴最大加加速度，单位[°/s³]
result[2]	J3 轴最大加加速度	string	0~720	J3 轴最大加加速度，单位[°/s³]
result[3]	J4 轴最大加加速度	string	0~720	J4 轴最大加加速度，单位[°/s³]
result[4]	J5 轴最大加加速度	string	0~720	J5 轴最大加加速度，单位[°/s³]
result[5]	J6 轴最大加加速度	string	0~720	J6 轴最大加加速度，单位[°/s³]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.12.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取最大关节加加速度

```
nRet = cps.HRIF_ReadJointMaxJerk(0,0, result)
```

### 3.5.13. HRIF\_ReadLinearMaxSpeed

3.5.13.1. 描述：读取直线运动最大速度参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	最大直线速度	string	0~3000	最大直线速度，默认[500]，单位[mm/s]
result[1]	最大直线加速度	string	0~2500	最大直线速度，默认[2500]，单位[mm/s <sup>2</sup> ]
result[2]	最大直线加加速度	string	0~100000	最大直线速度，默认[100000]，单位[mm/s <sup>3</sup> ]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.13.2. 示例

# 定义返回值空列表

result = [ ]

# 读取最大直线速度

nRet = cps.HRIF\_ReadLinearMaxSpeed(0,0, result)

### 3.5.14. HRIF\_ReadEmergencyInfo

3.5.14.1. 描述：读取急停信息。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	急停错误	string	0/1	急停回路有两路，当两路信号不相同，则认为急停回路有错误，值为 1。
result[1]	急停信号	string	0/1	急停信号，发生急停时，会断 48V 输出到本体的供电，但是不会断 220V 到 48V 的供电。
result[2]	安全光幕错误	string	0/1	安全光幕回路有两路，当两路信号不相同，则认为安全光幕回路有错误，值为 1。
result[3]	安全光幕信号	string	0/1	安全光幕信号，发生安全光幕时，会停止机器人运动，并且不再接受运动指令，不会断本体供电。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.14.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取最大直线速度

```
nRet = cps.HRIF_ReadEmergencyInfo(0, result)
```

### 3.5.15. HRIF\_ReadRobotState

3.5.15.1. 描述：读取当前机器人状态标志。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	运动中状态	string	0/1	0: 机器人不处于运动状态 1: 机器人运动中
result[1]	已使能状态	string	0/1	0: 机器人未使能 1: 机器人已使能
result[2]	错误状态	string	0/1	0: 未发生错误 1: 有错误发生
result[3]	错误码	string	>=0	报错错误码
result[4]	错误轴 ID	string	0~5	错误轴 ID
result[5]	抱闸状态	string	0/1	抱闸状态（松闸后受重力作用可能导致轴掉落） 0: 抱闸 1: 松闸
result[6]	暂停状态	string	0/1	0: 未处于暂停状态 1: 已处于暂停状态
result[7]	急停状态	string	0/1	0: 未处于急停状态 1: 已处于急停状态
result[8]	安全光幕状态	string	0/1	0: 未处于安全光幕状态 1: 已处于安全光幕状态
result[9]	已上电状态	string	0/1	0: 未上电状态 1: 已上电状态
result[10]	电箱连接状态	string	0/1	0: 电箱未连接

				1: 电箱已连接
result[11]	路点运动状态	string	0/1	0: 路点运动中 1: 路点运动完成
result[12]	到位状态	string	0/1	0: 机器人实际位置还没有运动到命令位置 1: 运动到位(命令与实际位置基本没有误差)

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

### 3.5.15.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取状态

```
nRet = cps.HRIF_ReadRobotState(0,0, result)
```

### 3.5.16. HRIF\_ReadRobotFlags

3.5.16.1. 描述：读取当前机器人状态标志。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	运动中状态	string	0/1	0: 机器人不处于运动状态 1: 机器人运动中
result[1]	已使能状态	string	0/1	0: 机器人未使能 1: 机器人已使能
result[2]	错误状态	string	0/1	0: 未发生错误 1: 有错误发生
result[3]	错误码	string	>=0	报错错误码
result[4]	错误轴 ID	string	0~5	错误轴 ID
result[5]	抱闸状态	string	0/1	抱闸状态（松闸后受重力作用可能导致轴掉落） 0: 抱闸 1: 松闸
result[6]	暂停状态	string	0/1	0: 未处于暂停状态 1: 已处于暂停状态
result[7]	路点运动状态	string	0/1	0: 路点运动中 1: 路点运动完成

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

---

### 3.5.16.2. 示例

# 定义返回值空列表

result = [ ]

# 读取状态

nRet = cps.HRIF\_ReadRobotFlags(0,0, result)



### 3.5.17. HRIF\_ReadCurWaypointID

3.5.17.1. 描述：读取 WayPoint 当前运动 ID 号。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	当前 ID	string	-	路点当前运动 ID，与 WayPoint,MoveJ, MoveL,MoveC 里设置的路点 ID 一致

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.5.17.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取路点 ID

```
nRet = cps.HRIF_ReadCurWaypointID(0,0, result)
```

### 3.5.18. HRIF\_ReadAxisErrorCode

3.5.18.1. 描述：读取错误码。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	当前错误码	string	$\geq 0$	当前错误码，没有错误码时=0
result[1]	J1 轴错误码	string	$\geq 0$	J1 轴错误码，没有错误码时=0
result[2]	J2 轴错误码	string	$\geq 0$	J2 轴错误码，没有错误码时=0
result[3]	J3 轴错误码	string	$\geq 0$	J3 轴错误码，没有错误码时=0
result[4]	J4 轴错误码	string	$\geq 0$	J4 轴错误码，没有错误码时=0
result[5]	J5 轴错误码	string	$\geq 0$	J5 轴错误码，没有错误码时=0
result[6]	J6 轴错误码	string	$\geq 0$	J6 轴错误码，没有错误码时=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	$> 0$ 的整型值	nRet=0：返回函数调用成功 nRet $> 0$ ：返回调用失败的错误码

3.5.18.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取错误码

```
nRet = cps.HRIF_ReadAxisErrorCode(0,0, result)
```

### 3.5.19. HRIF\_ReadCurFSM

3.5.19.1. 描述：读取当前状态机。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	当前状态机	string	$\geq 0$	状态机，可以参考状态机列表

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	$> 0$ 的整型值	nRet=0：返回函数调用成功 nRet $> 0$ ：返回调用失败的错误码

3.5.19.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取当前状态机

```
nRet = cps.HRIF_ReadCurFSM(0,0, result)
```

### 3.5.20. HRIF\_ReadCurFSMFromCPS

3.5.20.1. 描述：读取当前状态机。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	当前状态机	int	$\geq 0$	状态机，可以参考状态机列表

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	$> 0$ 的整型值	nRet=0：返回函数调用成功 nRet $> 0$ ：返回调用失败的错误码

3.5.20.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取当前状态机

```
nRet = cps.HRIF_ReadCurFSMFromCPS(0,0, result)
```

## 3.6. 位置、速度、电流读取指令

### 3.6.1. HRIF\_ReadActPos

3.6.1.1. 描述：读取当前实际位置信息。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	迪卡尔坐标	float	-	dX: X 坐标, 单位[mm] dY: Y 坐标, 单位[mm] dZ: Z 坐标, 单位[mm] dRx: Rx 坐标, 单位[°] dRy: Ry 坐标, 单位[°] dRz: Rz 坐标, 单位[°]
result[6]- result[11]	关节坐标	float	-	dJ1: 关节 1 坐标, 单位[°] dJ2: 关节 2 坐标, 单位[°] dJ3: 关节 3 坐标, 单位[°] dJ4: 关节 4 坐标, 单位[°] dJ5: 关节 5 坐标, 单位[°] dJ6: 关节 6 坐标, 单位[°]
result[12]- result[17]	当前工具坐标	float	-	dX: X 坐标, 单位[mm] dY: Y 坐标, 单位[mm] dZ: Z 坐标, 单位[mm] dRx: Rx 坐标, 单位[°] dRy: Ry 坐标, 单位[°]

				dRz: Rz 坐标, 单位[°]
result[18]- result[23]	当前用户坐标	float	-	dX: X 坐标, 单位[mm] dY: Y 坐标, 单位[mm] dZ: Z 坐标, 单位[mm] dRx: Rx 坐标, 单位[°] dRy: Ry 坐标, 单位[°] dRz: Rz 坐标, 单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

### 3.6.1.2. 示例

#### # 定义返回值空列表

```
result = [ ]
```

#### # 读取当前实际位置信息

```
nRet = cps.HRIF_ReadActPos(0,0, result)
```

#### # 读取关节位置变量

```
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])  
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
```

#### # 读取空间位置变量

```
dX = float(result[6]) dY = float(result[7]) dZ = float(result[8])  
dRx = float(result[9]) dRy = float(result[10]) dRz = float(result[11])
```

#### # 读取工具坐标变量

```
dTcp_X = float(result[12]) dTcp_Y = float(result[13]) dTcp_Z = float(result[14])  
dTcp_Rx = float(result[15]) dTcp_Ry = float(result[16]) dTcp_Rz = float(result[17])
```

#### # 读取用户坐标变量

```
dUcs_X = float(result[18]) dUcs_Y = float(result[19]) dUcs_Z = float(result[20])  
dUcs_Rx = float(result[21]) dUcs_Ry = float(result[22]) dUcs_Rz = float(result[23])
```

### 3.6.2. HRIF\_ReadCmdJointPos

3.6.2.1. 描述：读取关节命令位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	关节坐标	float	-	dJ1: 关节 1 命令位置，单位[°] dJ2: 关节 2 命令位置，单位[°] dJ3: 关节 3 命令位置，单位[°] dJ4: 关节 4 命令位置，单位[°] dJ5: 关节 5 命令位置，单位[°] dJ6: 关节 6 命令位置，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.6.2.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取关节命令位置

```
nRet = cps.HRIF_ReadCmdJointPos(0,0, result)
```

# 读取关节命令位置变量

```
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
```

```
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
```

### 3.6.3. HRIF\_ReadActJointPos

3.6.3.1. 描述：读取关节实际位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	关节坐标	float	-	dJ1：关节 1 实际位置，单位[°] dJ2：关节 2 实际位置，单位[°] dJ3：关节 3 实际位置，单位[°] dJ4：关节 4 实际位置，单位[°] dJ5：关节 5 实际位置，单位[°] dJ6：关节 6 实际位置，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.6.3.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取关节实际位置

```
nRet = cps.HRIF_ReadActJointPos(0,0, result)
```

# 读取关节实际位置变量

```
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
```

```
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
```



### 3.6.4. HRIF\_ReadCmdTcpPos

3.6.4.1. 描述：读取命令 TCP 位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	TCP 坐标	float	-	dX: X 坐标, 单位[mm] dY: Y 坐标, 单位[mm] dZ: Z 坐标, 单位[mm] dRx: Rx 坐标, 单位[°] dRy: Ry 坐标, 单位[°] dRz: Rz 坐标, 单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.6.4.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取 TCP 命令位置

```
nRet = cps.HRIF_ReadCmdTcpPos(0,0,result)
```

# 读取 TCP 命令位置

```
dX = float(result[0]) dY = float(result[1]) dZ = float(result[2])
```

```
dRx = float(result[3]) dRy = float(result[4]) dRz = float(result[5])
```

### 3.6.5. HRIF\_ReadActTcpPos

3.6.5.1. 描述：读取实际 TCP 位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	TCP 坐标	float	-	dX: X 坐标, 单位[mm] dY: Y 坐标, 单位[mm] dZ: Z 坐标, 单位[mm] dRx: Rx 坐标, 单位[°] dRy: Ry 坐标, 单位[°] dRz: Rz 坐标, 单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.6.5.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取 TCP 实际位置

```
nRet = cps.HRIF_ReadActTcpPos(0,0,result)
```

# 读取 TCP 实际位置变量

```
dX = float(result[0]) dY = float(result[1]) dZ = float(result[2])
```

```
dRx = float(result[3]) dRy = float(result[4]) dRz = float(result[5])
```

### 3.6.6. HRIF\_ReadCmdJointVel

3.6.6.1. 描述：读取关节命令速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	关节速度	float	-	dJ1: 关节 1 命令速度，单位[°/s] dJ2: 关节 2 命令速度，单位[°/s] dJ3: 关节 3 命令速度，单位[°/s] dJ4: 关节 4 命令速度，单位[°/s] dJ5: 关节 5 命令速度，单位[°/s] dJ6: 关节 6 命令速度，单位[°/s]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.6.6.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取关节命令速度

```
nRet = cps.HRIF_ReadCmdJointVel(0,0,result)
```

# 读取关节命令速度变量

```
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
```

```
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
```

### 3.6.7. HRIF\_ReadActJointVel

3.6.7.1. 描述：读取关节实际速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	关节速度	float	-	dJ1：关节 1 实际速度，单位[°/s] dJ2：关节 2 实际速度，单位[°/s] dJ3：关节 3 实际速度，单位[°/s] dJ4：关节 4 实际速度，单位[°/s] dJ5：关节 5 实际速度，单位[°/s] dJ6：关节 6 实际速度，单位[°/s]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.6.7.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取关节实际速度

```
nRet = cps.HRIF_ReadActJointVel(0,0,result)
```

# 读取关节实际速度变量

```
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
```

```
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
```

### 3.6.8. HRIF\_ReadCmdTcpVel

3.6.8.1. 描述：读取命令 TCP 速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	TCP 速度	float	-	dX: X 速度, 单位[mm/s] dY: Y 速度, 单位[mm/s] dZ: Z 速度, 单位[mm/s] dRx: Rx 速度, 单位[°/s] dRy: Ry 速度, 单位[°/s] dRz: Rz 速度, 单位[°/s]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.6.8.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取 TCP 命令速度

```
nRet = cps.HRIF_ReadCmdTcpVel(0,0,result)
```

# 读取 TCP 命令速度变量

```
dX = float(result[0]) dY = float(result[1]) dZ = float(result[2])
```

```
dRx = float(result[3]) dRy = float(result[4]) dRz = float(result[5])
```

### 3.6.9. HRIF\_ReadActTcpVel

3.6.9.1. 描述：读取实际 TCP 速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	TCP 速度	float	-	dX: X 速度, 单位[mm/s] dY: Y 速度, 单位[mm/s] dZ: Z 速度, 单位[mm/s] dRx: Rx 速度, 单位[°/s] dRy: Ry 速度, 单位[°/s] dRz: Rz 速度, 单位[°/s]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.6.9.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取 TCP 实际速度

```
nRet = cps.HRIF_ReadActTcpVel(0,0,result)
```

# 读取 TCP 实际速度变量

```
dX = float(result[0]) dY = float(result[1]) dZ = float(result[2])
```

```
dRx = float(result[3]) dRy = float(result[4]) dRz = float(result[5])
```

### 3.6.10. HRIF\_ReadCmdJointCur

3.6.10.1. 描述：读取关节命令电流。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	关节电流	float	-	dJ1：关节 1 命令电流，单位[A] dJ2：关节 2 命令电流，单位[A] dJ3：关节 3 命令电流，单位[A] dJ4：关节 4 命令电流，单位[A] dJ5：关节 5 命令电流，单位[A] dJ6：关节 6 命令电流，单位[A]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.6.10.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取关节命令电流

```
nRet = cps.HRIF_ReadCmdJointCur(0,0,result)
```

# 读取关节命令电流变量

```
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
```

```
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
```

### 3.6.11. HRIF\_ReadActJointCur

3.6.11.1. 描述：读取关节实际电流。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	关节电流	float	-	dJ1：关节 1 实际电流，单位[A] dJ2：关节 2 实际电流，单位[A] dJ3：关节 3 实际电流，单位[A] dJ4：关节 4 实际电流，单位[A] dJ5：关节 5 实际电流，单位[A] dJ6：关节 6 实际电流，单位[A]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.6.11.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取关节实际电流

```
nRet = cps.HRIF_ReadActJointCur(0,0,result)
```

# 读取关节实际电流变量

```
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
```

```
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
```



### 3.6.12. HRIF\_ReadTcpVelocity

3.6.12.1. 描述：读取 TCP 末端速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	命令速度	float	-	命令速度，单位[mm/s]
result[1]	实际速度	float	-	实际速度，单位[mm/s]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.6.12.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取当前末端 TCP 速度

```
nRet = cps.HRIF_ReadTcpVelocity(0,0,result)
```

### 3.7. 坐标转换计算指令

#### 3.7.1. HRIF\_Quaternion2RPY

3.7.1.1. 描述：四元素转欧拉角。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dQuaW	W	float	-	W
dQuaX	Xi	float	-	Xi
dQuaY	Yj	float	-	Yj
dQuaZ	Zk	float	-	Zk

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	欧拉角 Rx	float	$\geq -180, \leq 180$	欧拉角 Rx
result[1]	欧拉角 Ry	float	$\geq -180, \leq 180$	欧拉角 Ry
result[2]	欧拉角 Rz	float	$\geq -180, \leq 180$	欧拉角 Rz

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.7.1.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 需要转换的四元素变量

```
dQuaW = 0 dQuaX = 0 dQuaY = 0 dQuaZ = 0
```

# 转换

---

```
nRet = cps.HRIF_Quaternion2RPY(0,0,dQuaW, dQuaX, dQuaY, dQuaZ, result)
```

```
# 转换后的欧拉角结果
```

```
dRx = float(result[0]) dRy = float(result[1]) dRz = float(result[2])
```

### 3.7.2. HRIF\_RPY2Quaternion

3.7.2.1. 描述：欧拉角转四元素。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
Rx	欧拉角 Rx	float	$\geq -180, \leq 180$	欧拉角 Rx
Ry	欧拉角 Ry	float	$\geq -180, \leq 180$	欧拉角 Ry
Rz	欧拉角 Rz	float	$\geq -180, \leq 180$	欧拉角 Rz

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	W	float	-	W
result[1]	Xi	float	-	Xi
result[2]	Yj	float	-	Yj
result[3]	Zk	float	-	Zk

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.7.2.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 需要转换的转换后的欧拉角变量

```
dRx = 0 dRy = 0 dRz = 0
```

# 转换

```
nRet = cps.HRIF_RPY2Quaternion (0,0,dRx, dRy, dRz, result)
```

# 转换后的四元素变量

---

dQuaW = float(result[0]) dQuaX = float(result[1])

dQuaY= float(result[2]) dQuaZ = float(result[3])

### 3.7.3. HRIF\_GetInverseKin

3.7.3.1. 描述: 运动学逆解, 由指定用户坐标系位置和工具坐标系下的迪卡尔坐标计算对应的关节坐标位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值=0
result	返回值	list		传入空列表, result = [ ]
dCoord_X- dCoord_Rz	需要计算 逆解的目 标迪卡尔 位置	float	-	需要计算逆解的目标迪卡尔位置:  dCoord_X: X 坐标, 单位[mm]  dCoord_Y: Y 坐标, 单位[mm]  dCoord_Z: Z 坐标, 单位[mm]  dCoord_Rx: Rx 坐标, 单位[°]  dCoord_Ry: Ry 坐标, 单位[°]  dCoord_Rz: Rz 坐标, 单位[°]
dTcp_X- dTcp_Rz	工具坐标	float	-	目标位置是否包含工具坐标(不包含工具坐标则所有值=0):  dTcp_X: X 坐标, 单位[mm]  dTcp_Y: Y 坐标, 单位[mm]  dTcp_Z: Z 坐标, 单位[mm]  dTcp_Rx: Rx 坐标, 单位[°]  dTcp_Ry: Ry 坐标, 单位[°]  dTcp_Rz: Rz 坐标, 单位[°]
dUcs_X- dUcs_Rz	用户坐标	float	-	目标位置是否包含用户坐标(不包含用户坐标则所有值=0):  dUcs_X: X 坐标, 单位[mm]  dUcs_Y: Y 坐标, 单位[mm]  dUcs_Z: Z 坐标, 单位[mm]  dUcs_Rx: Rx 坐标, 单位[°]  dUcs_Ry: Ry 坐标, 单位[°]  dUcs_Rz: Rz 坐标, 单位[°]

dJ1-dJ6	参考关节坐标，逆解出现多个解时需要根据参考关节坐标选取最终解	float	-	dJ1: 关节 1 坐标, 单位[°] dJ2: 关节 2 坐标, 单位[°] dJ3: 关节 3 坐标, 单位[°] dJ4: 关节 4 坐标, 单位[°] dJ5: 关节 5 坐标, 单位[°] dJ6: 关节 6 坐标, 单位[°]
---------	--------------------------------	-------	---	--

**✓ 输出变量**

输出变量	名称	数据类型	有效范围	内容
result[0]-result[5]	关节坐标	float	-	dTargetJ1: 关节 1 坐标, 单位[°] dTargetJ2: 关节 2 坐标, 单位[°] dTargetJ3: 关节 3 坐标, 单位[°] dTargetJ4: 关节 4 坐标, 单位[°] dTargetJ5: 关节 5 坐标, 单位[°] dTargetJ6: 关节 6 坐标, 单位[°]

**✓ 返回值**

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

**3.7.3.2. 示例**

*# 定义返回值空列表*

```
result = [ ]
```

*# 定义需要转换的空间位置变量*

```
dCoord = [0, 0, 0, 0, 0, 0]
```

*# 定义工具坐标变量*

```
Tcp = [0, 0, 0, 0, 0, 0]
```

*# 定义工具坐标变量*

```
Ucs = [0, 0, 0, 0, 0, 0]
```

---

# 定义参考关节位置变量

rawACS = [0, 0, 0, 0, 0, 0]

# 求逆解

nRet = cps.HRIF\_GetInverseKin(0,0,dCoord, rawACS, Tcp, Ucs, result)

# 读取转换结果

dTargetJ1 = float(result[0]) dTargetJ2 = float(result[1]) dTargetJ3 = float(result[2])

dTargetJ4 = float(result[3]) dTargetJ5 = float(result[4]) dTargetJ6 = float(result[5])



### 3.7.4. HRIF\_GetForwardKin

3.7.4.1. 描述：运动学正解，由关节坐标位置计算指定用户坐标系和工具坐标系下的迪卡尔坐标位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dJ1-dJ6	需要计算正解的关节坐标	float	-	dJ1: 关节 1 坐标, 单位[°] dJ2: 关节 2 坐标, 单位[°] dJ3: 关节 3 坐标, 单位[°] dJ4: 关节 4 坐标, 单位[°] dJ5: 关节 5 坐标, 单位[°] dJ6: 关节 6 坐标, 单位[°]
dTcp_X- dTcp_Rz	工具坐标	float	-	目标位置是否包含工具坐标(不包含工具坐标则所有值=0): dTcp_X: X 坐标, 单位[mm] dTcp_Y: Y 坐标, 单位[mm] dTcp_Z: Z 坐标, 单位[mm] dTcp_Rx: Rx 坐标, 单位[°] dTcp_Ry: Ry 坐标, 单位[°] dTcp_Rz: Rz 坐标, 单位[°]
dUcs_X- dUcs_Rz	用户坐标	float	-	目标位置是否包含用户坐标(不包含用户坐标则所有值=0): dUcs_X: X 坐标, 单位[mm] dUcs_Y: Y 坐标, 单位[mm] dUcs_Z: Z 坐标, 单位[mm] dUcs_Rx: Rx 坐标, 单位[°] dUcs_Ry: Ry 坐标, 单位[°] dUcs_Rz: Rz 坐标, 单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
------	----	------	------	----

result[0]- result[5]	目标迪卡尔坐标	float	-	<p>如果正解得到的迪卡尔坐标不需要带工具坐标和用户坐标则</p> <p>将所有的工具坐标和用户坐标置 0:</p> <p>dTargetX: X 坐标, 单位[mm]</p> <p>dTargetY: Y 坐标, 单位[mm]</p> <p>dTargetZ: Z 坐标, 单位[mm]</p> <p>dTargetRx: Rx 坐标, 单位[°]</p> <p>dTargetRy: Ry 坐标, 单位[°]</p> <p>dTargetRz: Rz 坐标, 单位[°]</p>
-------------------------	---------	-------	---	---

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	<p>nRet=0: 返回函数调用成功</p> <p>nRet&gt;0: 返回调用失败的错误码</p>

### 3.7.4.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要转换的关节位置变量

```
rawACS = [0, 0, 0, 0, 0, 0]
```

# 定义工具坐标变量

```
Tcp = [0, 0, 0, 0, 0, 0]
```

# 定义用户坐标变量

```
Ucs = [0, 0, 0, 0, 0, 0]
```

# 求正解

```
nRet = cps.HRIF_GetForwardKin(0,0,rawACS, Tcp, Ucs, result)
```

# 获取转换后的空间位置结果

```
dTarget_X = float(result[0]) dTarget_Y = float(result[1]) dTarget_Z = float(result[2])
```

```
dTarget_Rx = float(result[3]) dTarget_Ry = float(result[4]) dTarget_Rz = float(result[5])
```

### 3.7.5. HRIF\_Base2UcsTcp

3.7.5.1. 描述：由基坐标系下的坐标位置计算指定用户坐标系和工具坐标系下的迪卡尔坐标位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dCoord_X- dCoord_Rz	基于基座坐 标系的迪卡 尔位置	float	-	需要转换的迪卡尔位置：  dCoord_X: X 坐标，单位[mm]  dCoord_Y: Y 坐标，单位[mm]  dCoord_Z: Z 坐标，单位[mm]  dCoord_Rx: Rx 坐标，单位[°]  dCoord_Ry: Ry 坐标，单位[°]  dCoord_Rz: Rz 坐标，单位[°]
dTcp_X- dTcp_Rz	工具坐标	float	-	目标位置是否包含工具坐标(不包含工具坐标则所有值=0):  dTcp_X: X 坐标，单位[mm]  dTcp_Y: Y 坐标，单位[mm]  dTcp_Z: Z 坐标，单位[mm]  dTcp_Rx: Rx 坐标，单位[°]  dTcp_Ry: Ry 坐标，单位[°]  dTcp_Rz: Rz 坐标，单位[°]
dUcs_X- dUcs_Rz	用户坐标	float	-	目标位置是否包含用户坐标(不包含用户坐标则所有值=0):  dUcs_X: X 坐标，单位[mm]  dUcs_Y: Y 坐标，单位[mm]  dUcs_Z: Z 坐标，单位[mm]  dUcs_Rx: Rx 坐标，单位[°]  dUcs_Ry: Ry 坐标，单位[°]  dUcs_Rz: Rz 坐标，单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	目标迪卡尔 坐标	float	-	指定用户坐标系和工具坐标系下的迪卡尔坐标位置：  dTargetX: X 坐标, 单位[mm]  dTargetY: Y 坐标, 单位[mm]  dTargetZ: Z 坐标, 单位[mm]  dTargetRx: Rx 坐标, 单位[°]  dTargetRy: Ry 坐标, 单位[°]  dTargetRz: Rz 坐标, 单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

### 3.7.5.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要转换的空间位置变量

```
dCoord = [0, 0, 0, 0, 0, 0]
```

# 定义工具坐标变量

```
Tcp = [0, 0, 0, 0, 0, 0]
```

# 定义用户坐标变量

```
Ucs = [0, 0, 0, 0, 0, 0]
```

# 基座坐标转换为用户坐标

```
nRet = cps.HRIF_Base2UcsTcp(0,0,dCoord, Tcp, Ucs, result)
```

# 定义转换后的空间位置结果

```
dTarget_X = float(result[0]) dTarget_Y = float(result[1]) dTarget_Z = float(result[2])
```

```
dTarget_Rx = float(result[3]) dTarget_Ry = float(result[4]) dTarget_Rz = float(result[5])
```

### 3.7.6. HRIF\_UcsTcp2Base

3.7.6.1. 描述：由指定用户坐标系和工具坐标系下的迪卡尔坐标位置计算基坐标系下的坐标位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
result	返回值	list		传入空列表, result = [ ]
dCoord_X- dCoord_Rz	迪卡尔位置	float	-	指定用户坐标系和工具坐标系下的迪卡尔坐标位置:  dCoord_X: X 坐标, 单位[mm]  dCoord_Y: Y 坐标, 单位[mm]  dCoord_Z: Z 坐标, 单位[mm]  dCoord_Rx: Rx 坐标, 单位[°]  dCoord_Ry: Ry 坐标, 单位[°]  dCoord_Rz: Rz 坐标, 单位[°]
dTcp_X- dTcp_Rz	工具坐标	float	-	目标位置是否包含工具坐标(不包含工具坐标则所有值=0):  dTcp_X: X 坐标, 单位[mm]  dTcp_Y: Y 坐标, 单位[mm]  dTcp_Z: Z 坐标, 单位[mm]  dTcp_Rx: Rx 坐标, 单位[°]  dTcp_Ry: Ry 坐标, 单位[°]  dTcp_Rz: Rz 坐标, 单位[°]
dUcs_X- dUcs_Rz	用户坐标	float	-	目标位置是否包含用户坐标(不包含用户坐标则所有值=0):  dUcs_X: X 坐标, 单位[mm]  dUcs_Y: Y 坐标, 单位[mm]  dUcs_Z: Z 坐标, 单位[mm]  dUcs_Rx: Rx 坐标, 单位[°]  dUcs_Ry: Ry 坐标, 单位[°]  dUcs_Rz: Rz 坐标, 单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	目标迪卡尔 坐标	float	-	基座坐标系下的迪卡尔坐标位置：  dTargetX: X 坐标, 单位[mm]  dTargetY: Y 坐标, 单位[mm]  dTargetZ: Z 坐标, 单位[mm]  dTargetRx: Rx 坐标, 单位[°]  dTargetRy: Ry 坐标, 单位[°]  dTargetRz: Rz 坐标, 单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功  nRet>0: 返回调用失败的错误码

### 3.7.6.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要转换的空间位置变量

```
dCoord = [0, 0, 0, 0, 0, 0]
```

# 定义工具坐标变量

```
Tcp = [0, 0, 0, 0, 0, 0]
```

# 定义用户坐标变量

```
Ucs = [0, 0, 0, 0, 0, 0]
```

# 用户坐标转换为基座坐标

```
nRet = cps.HRIF_UcsTcp2Base(0,dCoord, Tcp, Ucs, result)
```

# 读取转换后的空间位置结果

```
dTarget_X = float(result[0]) dTarget_Y = float(result[1]) dTarget_Z = float(result[2])
```

```
dTarget_Rx = float(result[3]) dTarget_Ry = float(result[4]) dTarget_Rz = float(result[5])
```

### 3.7.7. HRIF\_PoseAdd

3.7.7.1. 描述：点位加法计算，使用矩阵左乘运算（第二个点左乘第一个点）。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dPose1_X- dPose1_Rz	空间坐标 1	float	-	需要计算的空间坐标 1：  dPose1_X: X 坐标，单位[mm] dPose1_Y: Y 坐标，单位[mm] dPose1_Z: Z 坐标，单位[mm] dPose1_Rx: Rx 坐标，单位[°] dPose1_Ry: Ry 坐标，单位[°] dPose1_Rz: Rz 坐标，单位[°]
dPose2_X- dPose2_Rz	空间坐标 2	float	-	需要计算的空间坐标 2：  dPose2_X: X 坐标，单位[mm] dPose2_Y: Y 坐标，单位[mm] dPose2_Z: Z 坐标，单位[mm] dPose2_Rx: Rx 坐标，单位[°] dPose2_Ry: Ry 坐标，单位[°] dPose2_Rz: Rz 坐标，单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	计算结果	float	-	计算结果：  dPose3_X: X 坐标，单位[mm] dPose3_Y: Y 坐标，单位[mm] dPose3_Z: Z 坐标，单位[mm] dPose3_Rx: Rx 坐标，单位[°] dPose3_Ry: Ry 坐标，单位[°] dPose3_Rz: Rz 坐标，单位[°]

## ✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

## 3.7.7.2. 示例

# 定义返回值空列表

result = [ ]

# 定义需要计算的空间坐标 1

Pose1 = [420, 0, 445, 180, 0, 180]

# 定义需要计算的空间坐标 2

Pose2 = [420, 50, 445, 180, 0, 180]

# 计算结果

nRet = cps.HRIF\_PoseAdd(0,Pose1,Pose2, result)

# 计算结果

dPose3\_X = float(result[0]) dPose3\_Y = float(result[1]) dPose3\_Z = float(result[2])

dPose3\_Rx = float(result[3]) dPose3\_Ry = float(result[4]) dPose3\_Rz = float(result[5])



### 3.7.8. HRIF\_PoseSub

3.7.8.1. 描述：点位减法计算，以第二个点为参考点。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dPose1_X- dPose1_Rz	空间坐标 1	float	-	需要计算的空间坐标 1：  dPose1_X: X 坐标，单位[mm] dPose1_Y: Y 坐标，单位[mm] dPose1_Z: Z 坐标，单位[mm] dPose1_Rx: Rx 坐标，单位[°] dPose1_Ry: Ry 坐标，单位[°] dPose1_Rz: Rz 坐标，单位[°]
dPose2_X- dPose2_Rz	空间坐标 2	float	-	需要计算的空间坐标 2：  dPose2_X: X 坐标，单位[mm] dPose2_Y: Y 坐标，单位[mm] dPose2_Z: Z 坐标，单位[mm] dPose2_Rx: Rx 坐标，单位[°] dPose2_Ry: Ry 坐标，单位[°] dPose2_Rz: Rz 坐标，单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	计算坐标	float	-	计算结果：  dPose3_X: X 坐标，单位[mm] dPose3_Y: Y 坐标，单位[mm] dPose3_Z: Z 坐标，单位[mm] dPose3_Rx: Rx 坐标，单位[°]

				dPose3_Ry: Ry 坐标, 单位[°] dPose3_Rz: Rz 坐标, 单位[°]
--	--	--	--	--

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

### 3.7.8.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要计算的空间坐标 1

```
Pose1 = [420, 0, 445, 180, 0, 180]
```

# 定义需要计算的空间坐标 2

```
Pose2 = [420, 50, 445, 180, 0, 180]
```

# 计算结果

```
nRet = cps.HRIF_PoseSub(0,0,Pose1, Pose2, result)
```

# 计算结果

```
dPose3_X = float(result[0]) dPose3_Y = float(result[1]) dPose3_Z = float(result[2])
```

```
dPose3_Rx = float(result[3]) dPose3_Ry = float(result[4]) dPose3_Rz = float(result[5])
```

### 3.7.9. HRIF\_PoseTrans

3.7.9.1. 描述：坐标变换。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dPose1_X- dPose1_Rz	坐标位置 1	float	-	坐标位置 1：  dPose1_X: X 坐标，单位[mm]  dPose1_Y: Y 坐标，单位[mm]  dPose1_Z: Z 坐标，单位[mm]  dPose1_Rx: Rx 坐标，单位[°]  dPose1_Ry: Ry 坐标，单位[°]  dPose1_Rz: Rz 坐标，单位[°]
dPose2_X- dPose2_Rz	坐标位置 2	float	-	坐标位置 2：  dPose2_X: X 坐标，单位[mm]  dPose2_Y: Y 坐标，单位[mm]  dPose2_Z: Z 坐标，单位[mm]  dPose2_Rx: Rx 坐标，单位[°]  dPose2_Ry: Ry 坐标，单位[°]  dPose2_Rz: Rz 坐标，单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	计算坐标	float	-	计算结果：  dPose3_X: X 坐标，单位[mm]  dPose3_Y: Y 坐标，单位[mm]  dPose3_Z: Z 坐标，单位[mm]  dPose3_Rx: Rx 坐标，单位[°]

				dPose3_Ry: Ry 坐标, 单位[°] dPose3_Rz: Rz 坐标, 单位[°]
--	--	--	--	--

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

### 3.7.9.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要计算的空间坐标 1

```
Pose1 = [420, 0, 445, 180, 0, 180]
```

# 定义需要计算的空间坐标 2

```
Pose2 = [420, 50, 445, 180, 0, 180]
```

# 计算结果

```
nRet = cps.HRIF_PoseTrans(0,0,Pose1, Pose2, result)
```

# 计算结果

```
dPose3_X = float(result[0]) dPose3_Y = float(result[1]) dPose3_Z = float(result[2])
```

```
dPose3_Rx = float(result[3]) dPose3_Ry = float(result[4]) dPose3_Rz = float(result[5])
```

### 3.7.10. HRIF\_PoseInverse

3.7.10.1. 描述：坐标逆变换。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dPose1_X- dPose1_Rz	空间坐标 1	float	-	需要计算的空间坐标 1：  dPose1_X: X 坐标，单位[mm] dPose1_Y: Y 坐标，单位[mm] dPose1_Z: Z 坐标，单位[mm] dPose1_Rx: Rx 坐标，单位[°] dPose1_Ry: Ry 坐标，单位[°] dPose1_Rz: Rz 坐标，单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	计算坐标	float	-	计算结果：  dPose3_X: X 坐标，单位[mm] dPose3_Y: Y 坐标，单位[mm] dPose3_Z: Z 坐标，单位[mm] dPose3_Rx: Rx 坐标，单位[°] dPose3_Ry: Ry 坐标，单位[°] dPose3_Rz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

## 3.7.10.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要计算的空间坐标1

```
Pose = [420, 0, 445, 180, 0, 180]
```

# 计算结果

```
nRet = cps.HRIF_PoseInverse(0,0,Pose1, result)
```

# 计算结果

```
dPose3_X = float(result[0]) dPose3_Y = float(result[1]) dPose3_Z = float(result[2])
```

```
dPose3_Rx = float(result[3]) dPose3_Ry = float(result[4]) dPose3_Rz = float(result[5])
```

### 3.7.11. HRIF\_PoseDist

3.7.11.1. 描述：计算点位距离。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dPose1_X- dPose1_Rz	空间坐标 1	float	-	需要计算的空间坐标 1：  dPose1_X: X 坐标，单位[mm]  dPose1_Y: Y 坐标，单位[mm]  dPose1_Z: Z 坐标，单位[mm]  dPose1_Rx: Rx 坐标，单位[°]  dPose1_Ry: Ry 坐标，单位[°]  dPose1_Rz: Rz 坐标，单位[°]
dPose2_X- dPose2_Rz	空间坐标 2	float	-	需要计算的空间坐标 2：  dPose2_X: X 坐标，单位[mm]  dPose2_Y: Y 坐标，单位[mm]  dPose2_Z: Z 坐标，单位[mm]  dPose2_Rx: Rx 坐标，单位[°]  dPose2_Ry: Ry 坐标，单位[°]  dPose2_Rz: Rz 坐标，单位[°]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	点位距离	float	-	点位距离，单位[mm]
result[1]	姿态距离	float	-	姿态距离，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功

				nRet>0：返回调用失败的错误码
--	--	--	--	-------------------

## 3.7.11.2. 示例

# 定义返回值空列表

result = [ ]

# 定义需要计算的空间坐标 1

Pose1 = [420, 0, 445, 180, 0, 180]

# 定义需要计算的空间坐标 2

Pose2 = [420, 50, 445, 180, 0, 180]

# 计算结果

nRet = cps.HRIF\_PoseDist(0,0,Pose1, Pose2, result)



### 3.7.12. HRIF\_Poseinterpolate

3.7.12.1. 描述：空间位置直线插补计算。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dPose1_X- dPose1_Rz	空间坐标 1	float	-	需要计算的空间坐标 1：  dPose1_X：X 坐标，单位[mm] dPose1_Y：Y 坐标，单位[mm] dPose1_Z：Z 坐标，单位[mm] dPose1_Rx：Rx 坐标，单位[°] dPose1_Ry：Ry 坐标，单位[°] dPose1_Rz：Rz 坐标，单位[°]
dPose2_X- dPose2_Rz	空间坐标 2	float	-	需要计算的空间坐标 2：  dPose2_X：X 坐标，单位[mm] dPose2_Y：Y 坐标，单位[mm] dPose2_Z：Z 坐标，单位[mm] dPose2_Rx：Rx 坐标，单位[°] dPose2_Ry：Ry 坐标，单位[°] dPose2_Rz：Rz 坐标，单位[°]
dAlpha	插补比例	float	0-1	dAlpha=0：dPose3=dPose1 dAlpha=1：dPose3=dPose2 0-1：按照 dPose1 到 dPose2 的位置取比例为 dAlpha 的位置返回 dPose3

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	计算坐标	float	-	计算结果：

				dPose3_X: X 坐标, 单位[mm]
				dPose3_Y: Y 坐标, 单位[mm]
				dPose3_Z: Z 坐标, 单位[mm]
				dPose3_Rx: Rx 坐标, 单位[°]
				dPose3_Ry: Ry 坐标, 单位[°]
				dPose3_Rz: Rz 坐标, 单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

### 3.7.12.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 定义需要计算的空间坐标 1

```
Pose1 = [420, 0, 445, 180, 0, 180]
```

# 定义需要计算的空间坐标 2

```
Pose2 = [420, 50, 445, 180, 0, 180]
```

# 插补比例

```
dAlpha = 0.5
```

# 计算结果

```
nRet = cps.HRIF_Poseinterpolate(0,0,Pose1, Pose2, dAlpha, result)
```

# 计算结果

```
dPose3_X = float(result[0]) dPose3_Y = float(result[1]) dPose3_Z = float(result[2])
```

```
dPose3_Rx = float(result[3]) dPose3_Ry = float(result[4]) dPose3_Rz = float(result[5])
```

### 3.7.13. HRIF\_PoseDefdFrame

3.7.13.1. 描述：以轨迹中心旋转计算，p1,p2,p3 为旋转前选取的轨迹的特征点，p4,p5,p6 为旋转后选取的轨迹的特征点，计算结果表示为旋转特征的用户坐标系。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbotID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
dPose1_X- dPose1_Z	空间坐标 1	float	-	需要计算的空间坐标 1： dPose1_X: X 坐标，单位[mm] dPose1_Y: Y 坐标，单位[mm] dPose1_Z: Z 坐标，单位[mm]
dPose2_X- dPose2_Z	空间坐标 2	float	-	需要计算的空间坐标 2： dPose2_X: X 坐标，单位[mm] dPose2_Y: Y 坐标，单位[mm] dPose2_Z: Z 坐标，单位[mm]
dPose3_X- dPose3_Z	空间坐标 3	float	-	需要计算的空间坐标 3： dPose3_X: X 坐标，单位[mm] dPose3_Y: Y 坐标，单位[mm] dPose3_Z: Z 坐标，单位[mm]
dPose4_X- dPose4_Z	空间坐标 4	float	-	需要计算的空间坐标 4： dPose4_X: X 坐标，单位[mm] dPose4_Y: Y 坐标，单位[mm] dPose4_Z: Z 坐标，单位[mm]
dPose5_X- dPose5_Z	空间坐标 5	float	-	需要计算的空间坐标 5： dPose5_X: X 坐标，单位[mm] dPose5_Y: Y 坐标，单位[mm] dPose5_Z: Z 坐标，单位[mm]

dPose6_X- dPose6_Z	空间坐标 6	float	-	需要计算的空间坐标 6:  dPose6_X: X 坐标, 单位[mm]  dPose6_Y: Y 坐标, 单位[mm]  dPose2_Z: Z 坐标, 单位[mm]
-----------------------	--------	-------	---	--

**✓ 输出变量**

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	计算结果	float	-	计算结果:  dUcs_X: X 坐标, 单位[mm]  dUcs_Y: Y 坐标, 单位[mm]  dUcs_Z: Z 坐标, 单位[mm]  dUcs_Rx: Rx 坐标, 单位[°]  dUcs_Ry: Ry 坐标, 单位[°]  dUcs_Rz: Rz 坐标, 单位[°]

**✓ 返回值**

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功  nRet>0: 返回调用失败的错误码

**3.7.13.2. 示例**

```

# 定义返回值空列表
result = [ ]

# 定义需要计算的空间坐标 1
Pose1 = [0, 0, 0]

# 定义需要计算的空间坐标 2
Pose2 = [0, 0, 0]

# 定义需要计算的空间坐标 3
Pose3 = [0, 0, 0]

# 定义需要计算的空间坐标 4
Pose4 = [0, 0, 0]

# 定义需要计算的空间坐标 5

```

---

Pose5 = [0, 0, 0]

# 定义需要计算的空间坐标6

Pose6 = [0, 0, 0]

# 计算结果

nRet = cps.HRIF\_PoseDefdFrame(0,0,Pose1, Pose2, Pose3, Pose4, Pose5, Pose6, result)

## 3.8. 工具坐标与用户坐标读写指令

### 3.8.1. HRIF\_SetTCP

3.8.1.1. 描述：设置当前工具坐标，不写入配置文件，重启后失效。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rblID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dTcp_X-dTcp_Rz	工具坐标	float	-	需要设置的工具坐标：  dTcp_X: X 坐标，单位[mm] dTcp_Y: Y 坐标，单位[mm] dTcp_Z: Z 坐标，单位[mm] dTcp_Rx: Rx 坐标，单位[°] dTcp_Ry: Ry 坐标，单位[°] dTcp_Rz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.8.1.2. 示例

# 定义工具坐标变量

Tcp = [0, 0, 0, 0, 0, 0]

# 设置工具坐标

nRet = cps.HRIF\_SetTCP(0,0,Tcp)

### 3.8.2. HRIF\_SetUCS

3.8.2.1. 描述：设置当前用户坐标，不写入配置文件，重启后失效。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dUcs_X-dUcs_Rz	用户坐标	float	-	需要设置的用户坐标： dUcs_X：X 坐标，单位[mm] dUcs_Y：Y 坐标，单位[mm] dUcs_Z：Z 坐标，单位[mm] dUcs_Rx：Rx 坐标，单位[°] dUcs_Ry：Ry 坐标，单位[°] dUcs_Rz：Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.8.2.2. 示例

# 定义用户坐标变量

Ucs = [0, 0, 0, 0, 0, 0]

# 设置用户坐标

nRet = cps.HRIF\_SetUCS(0,0,Ucs)

### 3.8.3. HRIF\_ReadCurTCP

3.8.3.1. 描述：读取当前设置的工具坐标值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	工具坐标	float	-	读取到的工具坐标： dTcp_X: X 坐标，单位[mm] dTcp_Y: Y 坐标，单位[mm] dTcp_Z: Z 坐标，单位[mm] dTcp_Rx: Rx 坐标，单位[°] dTcp_Ry: Ry 坐标，单位[°] dTcp_Rz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.8.3.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取工具坐标

```
nRet = cps.HRIF_ReadCurTCP(0,0,result)
```

# 读取工具坐标变量

```
dTcp_X = float(result[0]) dTcp_Y = float(result[1]) dTcp_Z = float(result[2])
```

```
dTcp_Rx = float(result[3]) dTcp_Ry = float(result[4]) dTcp_Rz = float(result[5])
```



### 3.8.4. HRIF\_ReadCurUCS

3.8.4.1. 描述：读取当前设置的用户坐标值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	用户坐标	float	-	读取到的用户坐标： dUcs_X: X 坐标，单位[mm] dUcs_Y: Y 坐标，单位[mm] dUcs_Z: Z 坐标，单位[mm] dUcs_Rx: Rx 坐标，单位[°] dUcs_Ry: Ry 坐标，单位[°] dUcs_Rz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.8.4.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取用户坐标

```
nRet = cps.HRIF_ReadCurUCS(0,0, result)
```

# 读取用户坐标变量

```
dUcs_X = float(result[0]) dUcs_Y = float(result[1]) dUcs_Z = float(result[2])
```

```
dUcs_Rx = float(result[3]) dUcs_Ry = float(result[4]) dUcs_Rz = float(result[5])
```

### 3.8.5. HRIF\_SetTCPByName

3.8.5.1. 描述: 通过名称设置工具坐标列表中的值为当前工具坐标, 对应名称为示教器配置页面 TCP 示教的工具名称。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值=0
sTcpName	工具坐标名称	string	-	需要设置的工具坐标名称

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.8.5.2. 示例

# 需要下发的工具坐标名称

sTcpName = "TCP"

# 设置工具坐标

nRet = cps.HRIF\_SetTCPByName(0,0,sTcpName)

### 3.8.6. HRIF\_SetUCSByName

3.8.6.1. 描述：通过名称设置用户坐标列表中的值为当前用户坐标，对应名称为示教器配置页面用户坐标示教名称。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sUcsName	用户坐标名称	string	-	需要设置的用户坐标名称

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.8.6.2. 示例

# 需要下发的用户坐标名称

sUcsName = "UCS"

# 设置用户坐标

nRet = cps.HRIF\_SetUCSByName(0,0,sUcsName)

### 3.8.7. HRIF\_ReadTCPByName

3.8.7.1. 描述：通过名称读取指定 TCP 坐标，对应名称为示教器配置页面 TCP 示教的工具名称。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
sTcpName	工具坐标名称	string	-	需要读取的工具坐标名称

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	工具坐标	float	-	读取到的工具坐标： dTcp_X: X 坐标，单位[mm] dTcp_Y: Y 坐标，单位[mm] dTcp_Z: Z 坐标，单位[mm] dTcp_Rx: Rx 坐标，单位[°] dTcp_Ry: Ry 坐标，单位[°] dTcp_Rz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.8.7.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 需要读取的工具坐标名称

```
sTcpName = "TCP"
```

# 读取工具坐标

```
nRet = cps.HRIF_ReadTCPByName(0,0, sTcpName,result)
```

---

# 读取工具坐标结果变量

dTcp\_X = float(result[0]) dTcp\_Y = float(result[1]) dTcp\_Z = float(result[2])

dTcp\_Rx = float(result[3]) dTcp\_Ry = float(result[4]) dTcp\_Rz = float(result[5])

### 3.8.8. HRIF\_ReadUCSByName

3.8.8.1. 描述：通过名称读取指定 UCS 坐标，对应名称为示教器配置页面用户坐标示教的用户坐标名称。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
sUcsName	用户坐标名称	string	-	需要读取的用户坐标名称

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	用户坐标	float	-	读取到的用户坐标： dUcs_X: X 坐标，单位[mm] dUcs_Y: Y 坐标，单位[mm] dUcs_Z: Z 坐标，单位[mm] dUcs_Rx: Rx 坐标，单位[°] dUcs_Ry: Ry 坐标，单位[°] dUcs_Rz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.8.8.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 需要读取的用户坐标名称

```
sUcsName = "UCS"
```

# 读取用户坐标

```
nRet = cps.HRIF_ReadUCSByName(0,0,sUcsName , result)
```

---

# 读取用户坐标变量

dUcs\_X = 0 dUcs\_Y = 0 dUcs\_Z = 0

dUcs\_Rx = 0 dUcs\_Ry = 0 dUcs\_Rz = 0

## 3.9. 力控控制指令

### 3.9.1. HRIF\_SetForceControlState

3.9.1.1. 描述：设置力控状态，执行命令后机器人跳转到运动状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nState	力控状态	int	0/1-	力控状态： 0：关闭力控 1：开启力控

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.1.2. 示例

# 定义设置状态

nState = 1

# 设置力控状态

nRet = cps.HRIF\_SetForceControlState(0,0,nState)



### 3.9.2. HRIF\_ReadForceControlState

3.9.2.1. 描述：读取当前力控状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	力控状态	int	0~3	力控状态： 0：关闭状态 1：开力控探寻状态 2：力控探寻完成状态 3：力控自由驱动状态

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.2.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取力控状态

```
nRet = cps.HRIF_ReadForceControlState(0,0,result)
```

# 读取到的力控状态

```
nState = int(result[0])
```

### 3.9.3. HRIF\_SetForceToolCoordinateMotion

3.9.3.1. 描述：设置力控坐标系方向为 Tool 坐标方向模式。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nMode	模式	int	0/1	设置力控坐标系为 Tool 方向： 0：关闭 1：开启

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.3.2. 示例

# 定义设置状态

nState = 1

# 设置力控坐标系状态

nRet = cps.HRIF\_SetForceToolCoordinateMotion(0,0,nState)

### 3.9.4. HRIF\_ForceControlinterrupt

3.9.4.1. 描述：暂停力控运动，仅暂停力控功能，不暂停运动和脚本。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.4.2. 示例

# 设置力控暂停状态

```
nRet = cps.HRIF_ForceControlinterrupt(0,0)
```

### 3.9.5. HRIF\_ForceControlContinue

3.9.5.1. 描述：继续力控运动，仅继续力控运动功能，不继续运动和脚本。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.5.2. 示例

# 设置力控继续运动

```
nRet = cps.HRIF_ForceControlContinue(0,0)
```

### 3.9.6. HRIF\_SetForceZero

3.9.6.1. 描述：力控清零，在原有数据的基础上重新标定力传感器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.6.2. 示例

# 清零力控数据

```
nRet = cps.HRIF_SetForceZero(0,0)
```

### 3.9.7. HRIF\_SetMaxSearchVelocities

3.9.7.1. 描述：设置力控探寻的最大速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值=0
dMaxLinearVelocity	直线速度	float	>0	探寻的直线最大速度, 单位[mm/s]
dMaxAngularVelocity	姿态角速度	float	>0	探寻的姿态最大速度, 单位[°/s]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0 : 返回函数调用成功 nRet>0 : 返回调用失败的错误码

3.9.7.2. 示例

*#设置力控探寻直线速度*

dMaxLinearVelocity = 100

*# 设置力控探寻姿态角速度*

dMaxAngularVelocity = 50

*# 设置力控探寻速度*

nRet = cps.HRIF\_SetMaxSearchVelocities(0,0,dMaxLinearVelocity, dMaxAngularVelocity)

### 3.9.8. HRIF\_SetControlFreedom

3.9.8.1. 描述：设置力控探寻自由度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nX-nRz	各方向自由度	int	0/1	各轴探寻自由度开关： 0：关闭 1：开启

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.8.2. 示例

# 定义力控自由度状态

dWrench = [0, 0, 0, 0, 0, 0]

# 设置力控自由度状态

nRet = cps.HRIF\_SetControlGoal (0,0,dWrench)

### 3.9.9. HRIF\_SetForceControlStrategy

3.9.9.1. 描述：设置力控控制策略。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nState	控制策略	int	0~2	各轴探寻自由度开关： 0：恒力模式 1：柔顺模式 2：柔顺越障模式

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.9.2. 示例

# 定义力控策略

nState = 1

# 设置力控策略为恒力模式

nRet = cps.HRIF\_SetForceControlStrategy(0,0,nState)



### 3.9.10. HRIF\_SetFreeDrivePositionAndOrientation

3.9.10.1. 描述：设置力传感器中心相对于法兰盘的安装位置和姿态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dX-dRz	迪卡尔坐标	float	-	力传感器相对于法兰盘安装位置和姿态： dX: X 坐标，单位[mm] dY: Y 坐标，单位[mm] dZ: Z 坐标，单位[mm] dRx: Rx 坐标，单位[°] dRy: Ry 坐标，单位[°] dRz: Rz 坐标，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.10.2. 示例

# 定义力传感器安装位置和姿态

```
dPCS = [0, 0, 0, 0, 0, 0]
```

# 设置力传感器的安装位置和姿态

```
nRet = cps.HRIF_SetFreeDrivePositionAndOrientation(0,0,dPCS)
```

### 3.9.11. HRIF\_SetPIDControlParams

3.9.11.1. 描述：设置力控探寻 PID 参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dFp	PID 参数	float	-	PID 参数 fP
dFi	PID 参数	float	-	PID 参数 fI
dFd	PID 参数	float	-	PID 参数 fD
dTp	PID 参数	float	-	PID 参数 tP
dTi	PID 参数	float	-	PID 参数 tI
dTd	PID 参数	float	-	PID 参数 tD

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.11.2. 示例

# 设置 PID 参数

dFp = 1.0 dFi = 0.1 dFd = 0

dTp = 1.0 dTi = 0.1 dTd = 0

# 设置 PID 参数

nRet = cps.HRIF\_SetPIDControlParams(0,0,dFp, dFi, dFd, dTp, dTi, dTd)

### 3.9.12. HRIF\_SetMassParams

3.9.12.1. 描述：设置惯量控制参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dX-dRz	惯量控制参数	float	-	惯量控制参数： dX: X 方向 dY: Y 方向 dZ: Z 方向 dRx: Rx 方向 dRy: Ry 方向 dRz: Rz 方向

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.12.2. 示例

# 设置惯量控制参数

```
Mass = [0, 0, 0, 0, 0, 0]
```

```
nRet = cps.HRIF_SetMassParams(0,0,Mass)
```

### 3.9.13. HRIF\_SetDampParams

3.9.13.1. 描述：设置阻尼(b)控制参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dX-dRz	阻尼控制参数	float	-	阻尼控制参数： dX: X 方向 dY: Y 方向 dZ: Z 方向 dRx: Rx 方向 dRy: Ry 方向 dRz: Rz 方向

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.13.2. 示例

# 设置阻尼参数

Damp = [800, 800, 800, 40, 40, 40]

# 设置阻尼参数

nRet = cps.HRIF\_SetDampParams(0,0,Damp)

### 3.9.14. HRIF\_SetStiffParams

3.9.14.1. 描述：设置刚度(k)控制参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dX-dRz	刚度控制参数	float	-	刚度控制参数： dX: X 方向 dY: Y 方向 dZ: Z 方向 dRx: Rx 方向 dRy: Ry 方向 dRz: Rz 方向

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.14.2. 示例

# 设置刚度参数

```
Stiff = [1000, 1000, 1000, 100, 100, 100]
```

# 设置刚度参数

```
nRet = cps.HRIF_SetStiffParams(0,0,Stiff)
```

### 3.9.15. HRIF\_SetForceControlGoal

3.9.15.1. 描述：设置力控目标力。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dX-dRz	力控目标力	float	-	力控目标力： dX: X 方向，单位[N] dY: Y 方向，单位[N] dZ: Z 方向，单位[N] dRx: Rx 方向，单位[NM] dRy: Ry 方向，单位[NM] dRz: Rz 方向，单位[NM]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.15.2. 示例

# 设置力控目标力

Goal = [0, 0, 10, 0, 0, 0]

# 设置力控目标力 Z 方向 10N

nRet = cps.HRIF\_SetForceControlGoal(0,0,Goal)

### 3.9.16. HRIF\_SetControlGoal

3.9.16.1. 描述：设置力控目标力和目标距离(0,0,力控目标距离暂未启用)。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dWrench_X- dWrench_Rz	力控目标力	float	-	力控目标力：  dWrenchX：X 方向，单位[N] dWrenchY：Y 方向，单位[N] dWrenchZ：Z 方向，单位[N] dWrenchRx：Rx 方向，单位[NM] dWrenchRy：Ry 方向，单位[NM] dWrenchRz：Rz 方向，单位[NM]
dDistance_X- dDistance_Rz	力控目标距离	float	-	力控目标距离：  dDistanceX：X 方向，单位[N] dDistanceY：Y 方向，单位[N] dDistanceZ：Z 方向，单位[N] dDistanceRx：Rx 方向，单位[NM] dDistanceRy：Ry 方向，单位[NM] dDistanceRz：Rz 方向，单位[NM]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.16.2. 示例

# 设置力控目标力

Wrench = [0, 0, 10, 0, 0, 0]

# 设置力控目标距离

---

Distance = [0, 0, 0, 0, 0, 0]

# 设置力控目标力Z 方向 10N

nRet = cps.HRIF\_SetControlGoal (0,0,Wrench, Distance)



### 3.9.17. HRIF\_SetForceDataLimit

3.9.17.1. 描述：设置力控限制范围，力传感器超过此范围后控制器断电。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dMax_X-dMax_Rz	力最大范围	float	-	力最大范围： dMax_X: X 方向，单位[N] dMax_Y: Y 方向，单位[N] dMax_Z: Z 方向，单位[N] dMax_Rx: Rx 方向，单位[NM] dMax_Ry: Ry 方向，单位[NM] dMax_Rz: Rz 方向，单位[NM]
dMin_X-dMin_Rz	力最小范围	float	-	力最小范围： dMin_X: X 方向，单位[N] dMin_Y: Y 方向，单位[N] dMin_Z: Z 方向，单位[N] dMin_Rx: Rx 方向，单位[NM] dMin_Ry: Ry 方向，单位[NM] dMin_Rz: Rz 方向，单位[NM]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.17.2. 示例

# 设置力最大值

dMax = [500, 500, 500, 500, 500, 500]

# 设置力最小值

---

dMin = [-500, -500, -500, -500, -500, -500]

# 设置力传感器数据限制范围

nRet = cps.HRIF\_SetForceDataLimit(0,0, dMax, dMin)

### 3.9.18. HRIF\_SetForceDistanceLimit

3.9.18.1. 描述：设置力控形变范围。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dAllowDistance	允许最大距离	float		允许最大形变距离，单位[mm]
dStrengthLevel	位置与边界设置偏离距离的幂次项	float	2/3	位置与边界设置偏离距离的幂次项： 2：阻力与偏离边界的平方项成比例； 3：阻力与偏离边界的立方项成比例

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.18.2. 示例

# 设置允许最大距离

dAllowDistance = 10

# 设置位置与边界设置偏离距离的幂次项

dStrengthLevel = 2

# 设置力控形变范围

nRet = cps.HRIF\_SetForceDistanceLimit(0,0, dAllowDistance, dStrengthLevel)

### 3.9.19. HRIF\_SetForceFreeDriveMode

3.9.19.1. 描述：设置开启或者关闭力控自由驱动模式。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
bEnable	是否开启	bool	False/True	False：关闭 True：开启

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.9.19.2. 示例

*# 设置开启力控自由驱动*

bool bEnable = True

*# 设置开启力控自由驱动*

nRet = cps.HRIF\_SetForceFreeDriveMode(0,0,bEnable)

### 3.9.20. HRIF\_ReadFTCabData

3.9.20.1. 描述：读取标定后的力传感器数据。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]- result[5]	标定后的力 传感器数据	float	-	dX: X 坐标, 单位[N] dY: Y 坐标, 单位[N] dZ: Z 坐标, 单位[N] dRx: Rx 坐标, 单位[NM] dRy: Ry 坐标, 单位[NM] dRz: Rz 坐标, 单位[NM]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.9.20.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取力传感器数据

```
nRet = cps.HRIF_ReadFTCabData(0,0,result)
```

# 读取到力传感器数据

```
dX = float(result[0]) dY = float(result[1]) dZ = float(result[2])
```

```
dRx = float(result[3]) dRy = float(result[4]) dRz = float(result[5])
```

## 3.10.通用运动类控制指令

### 3.10.1. HRIF\_MoveRelJ

3.10.1.1. 描述：关节相对运动。

✓ 输入变量

参数	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nAxis	轴 ID	int	0~5	运动的目标轴 ID，对应关节 J1-J6
nDirection	方向	int	0/1	0：负方向 1：正方向
dDistance	运动距离	float	-	相对运动距离

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.10.1.2. 示例

# 定义轴ID

nAxis = 1

# 定义运动方向

nDirection = 1

# 定义运动距离

nDistance = 1

# 执行相对关节运动

nRet = cps.HRIF\_MoveRelJ(0,0, nAxis, nDirection, nDistance)

### 3.10.2. HRIF\_MoveRelL

3.10.2.1. 描述：空间相对运动。

✓ 输入变量

参数	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nAxis	轴 ID	int	0~5	运动的目标轴 ID，对应关节 J1-J6
nDirection	方向	int	0/1	0：负方向 1：正方向
dDistance	运动距离	float	-	相对运动距离
nToolMotion	运动坐标类型	int	0/1	0：按当前选择的用户坐标运动 1：按 Tool 坐标运动

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.10.2.2. 示例

# 定义轴ID

nAxis = 1

# 定义运动方向

nDirection = 1

# 定义运动距离

nDistance= 1

# 定义运动坐标类型

nToolMotion = 1

# 执行相对空间运动

nRet = cps.HRIF\_MoveRelL(0,0, nAxis, nDirection, nDistance, nToolMotion)

### 3.10.3. HRIF\_WayPointRel

3.10.3.1. 描述：路点相对运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值=0
nType	运动类型	int	>0	0: 关节相对运动 1: 直线相对运动
nPointList	是否使用列表点位	int	>1	0: 不使用点位列表中点位 1: 使用点位列表中点位
Pos	空间位置	list	-	nPointList= 1: 点位空间位置  dPos_X: X 坐标, 单位[mm] dPos_Y: Y 坐标, 单位[mm] dPos_Z: Z 坐标, 单位[mm] dPos_Rx: Rx 坐标, 单位[°] dPos_Ry: Ry 坐标, 单位[°] dPos_Rz: Rz 坐标, 单位[°]  nPointList= 0, 均为 0
rawACT	关节位置	list	-	nPointList= 1: 点位关节位置  dPos_J1: 关节 1 坐标, 单位[°] dPos_J2: 关节 2 坐标, 单位[°] dPos_J3: 关节 3 坐标, 单位[°] dPos_J4: 关节 4 坐标, 单位[°] dPos_J5: 关节 5 坐标, 单位[°] dPos_J6: 关节 6 坐标, 单位[°]  nPointList= 0, 均为 0
nrelMoveType	相对运动类型	int	-	0: 绝对值 1: 叠加值
nAxisMask	各轴是否运动	list	-	各轴/各方向是否运动



				0: 不运动 1: 运动
dTarget	运动距离	float	-	nType= 0 并 nAxisMask= 1: 该方向运动绝对距离或叠加距离 nType= 1 并 nAxisMask= 1: 该轴运动绝对距离或叠加距离 nAxisMask= 0: 无效
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint= 1 时无效, 可使用默认名称 “TCP”
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint= 1 时无效, 可使用默认名称 “Base”
dVelocity	速度	float	-	运动最大速度, 关节运动时单位[°/s], 空间运动时 X, Y, Z 单位[mm/s], Rx, Ry, Rz 单位[°/s]
dAcc	加速度	float	-	运动最大加速度, 关节运动时单位[°/s <sup>2</sup> ], 空间运动时 X, Y, Z 单位[mm/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ]
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	是否使用关节角度作为目标点, 如果 nMoveType= 0 时, 则 nIsUseJoint 有效: 0: 不使用关节角度 1: 使用关节角度
nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1, 则开启检测 DI 停止, 路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态= nIOState 时, 机器人停止运动, 否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引, nIsSeek= 0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态, , nIsSeek= 0 时无效

strCmdID	命令 ID	string	-	当前路点 ID，可以自定义，也可以按顺序设置为“1”，“2”，“3”。
----------	-------	--------	---	-------------------------------------

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

### 3.10.3.2. 示例

```

# 定义运动类型
nType = 0;

# 定义是否使用点位列表的点位
nPointList= 0;

# 定义空间目标位置
Pos = [0, 0, 0, 0, 0, 0]

# 定义关节目标位置
rawACT = [0, 0, 0, 0, 0, 0]

# 定义相对运动类型
nrelMoveType= 1;

# 定义各轴各方向是否运动
nAxisMask = [1, 1, 0, 0, 0, 0]

# 定义运动距离
nTarget = [10, -10, 0, 0, 0, 0]

# 定义工具坐标变量
sTcpName = "TCP";

# 定义用户坐标变量
sUcsName = "Base";

# 定义运动速度
dVelocity = 50;

# 定义运动加速度
dAcc = 50;

# 定义过渡半径

```

---

```
dRadius = 50;
# 定义是否使用关节角度

nIsUseJoint = 0;
# 定义是否使用检测 DI 停止

nIsSeek = 0;
# 定义检测的 DI 索引

nIOBit = 0;
# 定义检测的 DI 状态

nIOState = 0;
# 定义路点 ID

strCmdID = "0";
# 路点相对运动

int nRet = HRIF_WayPointRel(0,0,nType, nPointList, Pos, rawACT, nrelMoveType, nAxisMask, nTarget,
sTcpName, sUcsName, dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)
```

### 3.10.4. HRIF\_IsMotionDone

3.10.4.1. 描述：判断机器人是否处于运动状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	返回值	string	False/True	False：运动未完成，处于运动状态 True：运动完成，处于准备就绪状态

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.10.4.2. 示例

*# 定义返回值空列表*

result = [ ]

*# 判断机器人是否处于运动状态*

nRet = cps.HRIF\_IsMotionDone(0,0,result)

*# 机器人处于运动状态*

result[0] = 'False'

*# 机器人不处于运动状态*

Result[0] = 'True'

### 3.10.5. HRIF\_IsBlendingDone

3.10.5.1. 描述：判断路点是否运动完成。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

\* 注：此接口仅判断路点是否运动完成，返回 True 后，机器人可能还处于运动状态；如：开启跟随、力控、相对跟踪后，机器人处于运动状态，在没有调用路点时，此接口返回 True.

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	返回值	string	False/True	False：运动未完成，处于运动状态 True：运动完成，处于准备就绪状态

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.10.5.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 判断路点是否运动完成

```
nRet = cps.HRIF_IsBlendingDone(0,0,result)
```

# 路点运动未完成

```
result[0] = 'False'
```

# 路点运动完成

```
Result[0] = 'True'
```

### 3.10.6. HRIF\_WayPointEx

3.10.6.1. 描述：路点运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nMoveType	运动类型	int	0/1	0：关节运动 1：空间运动
dX-dRz	空间目标位置	float	-	nMoveType=0 并 nIsUseJoint=1：无效， nMoveType=0 并 nIsUseJoint=0：用此空间坐标作为目标位置，通过逆解计算得到关节坐标为目标关节坐标 nMoveType=1：目标空间位置 dX：X 坐标，单位[mm] dY：Y 坐标，单位[mm] dZ：Z 坐标，单位[mm] dRx：Rx 坐标，单位[°] dRy：Ry 坐标，单位[°] dRz：Rz 坐标，单位[°]
dJ1-dJ6	关节目标位置	float	-	nMoveType=0 并 nIsUseJoint=1：使用此关节坐标作为目标关节坐标 nMoveType=0 并 nIsUseJoint=0：此关节坐标仅作为计算逆解时选解的参考关节坐标 nMoveType=1：无效 dJ1：关节 1 坐标，单位[°] dJ2：关节 2 坐标，单位[°] dJ3：关节 3 坐标，单位[°] dJ4：关节 4 坐标，单位[°] dJ5：关节 5 坐标，单位[°]

				dJ6: 关节 6 坐标, 单位[°]
dTcp_X- dTcp_Rz	工具坐标值	float	-	<p>目标空间坐标所处的工具坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可设置为 0:</p> <p>dTcp_X: X 坐标, 单位[mm]</p> <p>dTcp_Y: Y 坐标, 单位[mm]</p> <p>dTcp_Z: Z 坐标, 单位[mm]</p> <p>dTcp_Rx: Rx 坐标, 单位[°]</p> <p>dTcp_Ry: Ry 坐标, 单位[°]</p> <p>dTcp_Rz: Rz 坐标, 单位[°]</p>
dUcs_X- dUcs_Rz	用户坐标值	float	-	<p>目标空间坐标所处的用户坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可设置为 0:</p> <p>dUcs_X: X 坐标, 单位[mm]</p> <p>dUcs_Y: Y 坐标, 单位[mm]</p> <p>dUcs_Z: Z 坐标, 单位[mm]</p> <p>dUcs_Rx: Rx 坐标, 单位[°]</p> <p>dUcs_Ry: Ry 坐标, 单位[°]</p> <p>dUcs_Rz: Rz 坐标, 单位[°]</p>
dVelocity	速度	float	-	运动最大速度, 关节运动时单位[°/s], 空间运动时 XYZ 单位[mm/s], Rx, Ry, Rz 单位[°/s]
dAcc	加速度	float	-	运动最大加速度, 关节运动时单位[°/s²], 空间运动时 XYZ 单位[mm/s²], Rx, Ry, Rz 单位[°/s²]
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	<p>是否使用关节角度作为目标点, 如果 nMoveType=0 时, 则 nIsJoint 有效:</p> <p>0: 不使用关节角度</p> <p>1: 使用关节角度</p>

nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1，则开启检测 DI 停止，路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态=nIOState 时，机器人停止运动，否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0-7	电箱对应 DI 索引，nIsSeek=0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态，nIsSeek=0 时无效
strCmdID	命令 ID	string	-	当前路点 ID，可以自定义，也可以按顺序设置为"1"，"2"，"3"

\* 注：HRIF\_WayPointEx 需要设置工具坐标与用户坐标具体的值；HRIF\_WayPoint 使用示教器示教的工具坐标与用户坐标名称。

#### ✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

#### 3.10.6.2. 示例

##### # 定义运动类型

```
nMoveType = 0
```

##### # 定义空间目标位置

```
Point= [500, 500, 500, 500, 500, 500]
```

##### # 定义关节目标位置

```
RawACSpoints = [0, 0, 0, 0, 0, 0]
```

##### # 定义工具坐标变量

```
Tcp = [0, 0, 0, 0, 0, 0]
```

##### # 定义用户坐标变量

```
Ucs = [0, 0, 0, 0, 0, 0]
```

##### # 定义运动速度

```
dVelocity = 50
```

##### # 定义运动加速度

```
dAcc = 50
```

##### # 定义过渡半径



---

```
dRadius = 50
# 定义是否使用关节角度

nIsUseJoint = 1
# 定义是否使用检测DI 停止

nIsSeek = 0
# 定义检测的DI 索引

nIOBit = 0
# 定义检测的DI 状态

nIOState = 0
# 定义路点ID

stdCmdID = "0"
# 执行路点运动

nRet = cps.HRIF_WayPointEx(0,0,nMoveType , Point, rawACS,Tcp, Ucs, ,dVelocity, dAcc, dRadius,
nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)
```

### 3.10.7. HRIF\_WayPoint

3.10.7.1. 描述：路点运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nMoveType	运动类型	int	0/1	0：关节运动 1：直线运动
dX-dRz	空间目标位置	float	-	nMoveType=0 并 nIsUseJoint=1：无效， nMoveType=0 并 nIsUseJoint=0：用此空间坐标作为目标位置，通过逆解计算得到关节坐标为目标关节坐标 nMoveType=1：目标空间位置 dX：X 坐标，单位[mm] dY：Y 坐标，单位[mm] dZ：Z 坐标，单位[mm] dRx：Rx 坐标，单位[°] dRy：Ry 坐标，单位[°] dRz：Rz 坐标，单位[°]
dJ1-dJ6	关节目标位置	float	-	nMoveType=0 并 nIsUseJoint=1：使用此关节坐标作为目标关节坐标 nMoveType=0 并 nIsUseJoint=0：此关节坐标仅作为计算逆解时选解的参考关节坐标 nMoveType=1：无效 dJ1：关节 1 坐标，单位[°] dJ2：关节 2 坐标，单位[°] dJ3：关节 3 坐标，单位[°] dJ4：关节 4 坐标，单位[°] dJ5：关节 5 坐标，单位[°]

				dJ6: 关节 6 坐标, 单位[°]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"Base"
dVelocity	速度	float	-	运动最大速度, 关节运动时单位[°/s], 空间运动时 XYZ 单位[mm/s], Rx, Ry, Rz 单位[°/s]
dAcc	加速度	float	-	运动最大加速度, 关节运动时单位[°/s²], 空间运动时 XYZ 单位[mm/s²], Rx, Ry, Rz 单位[°/s²]
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	是否使用关节角度作为目标点, 如果 nMoveType=0 时, 则 nIsJoint 有效: 0: 不使用关节角度 1: 使用关节角度
nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1, 则开启检测 DI 停止, 路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态=nIOState 时, 机器人停止运动, 否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引, nIsSeek=0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态, nIsSeek=0 时无效
strCmdID	命令 ID	string	-	当前路点 ID, 可以自定义, 也可以按顺序设置为"1", "2", "3"

\* 注: HRIF\_WayPointEx 需要设置工具坐标与用户坐标具体的值; HRIF\_WayPoint 使用示教器示教的工具坐标与用户坐标名称。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
-----	----	------	------	----

nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码
------	-----	-----	---------	--------------------------------------

### 3.10.7.2. 示例

```

# 定义运动类型
nMoveType = 0
# 定义空间目标位置
Point = [0, 0, 0, 0, 0, 0]
# 定义关节目标位置
rawACS = [0, 0, 0, 0, 0, 0]
# 定义工具坐标变量
sTcpName = "TCP"
# 定义用户坐标变量
sUcsName = "Base"
# 定义运动速度
dVelocity = 50
# 定义运动加速度
dAcc = 50
# 定义过渡半径
dRadius = 50
# 定义是否使用关节角度
nIsUseJoint= 1
# 定义是否使用检测DI 停止
nIsSeek = 0
# 定义检测的DI 索引
nIOBit = 0
# 定义检测的DI 状态
nIOState = 0
# 定义路点ID
stdCmdID = "0"
# 执行路点运动

```

---

```
nRet = cps.HRIF_WayPoint(0,0,nMoveType , Point, rawACS, sTcpName , sUcsName, dVelocity, dAcc, dRadius,  
nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)
```

### 3.10.8. HRIF\_WayPoint2

3.10.8.1. 描述：路点运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nMoveType	运动类型	int	0/1	0: 关节运动 1: 直线运动 2: 圆弧运动
dEndPos_X- dEndPos_Rz	空间目标位置	float	-	nMoveType=0 并 nIsUseJoint=1: 无效, nMoveType=0 并 nIsUseJoint=0: 用此空间 坐标作为目标位置, 通过逆解计算得到关 节坐标为目标关节坐标  nMoveType=1: 目标空间位置 nMoveType=2: 做为圆弧的目标位置  dEndPos _X: X 坐标, 单位[mm] dEndPos _Y: Y 坐标, 单位[mm] dEndPos _Z: Z 坐标, 单位[mm] dEndPos _Rx: Rx 坐标, 单位[°] dEndPos _Ry: Ry 坐标, 单位[°] dEndPos _Rz: Rz 坐标, 单位[°]
dAuxPos_X- dAuxPos_Rz	空间目标位置	float	-	nMoveType=0: 无效 nMoveType=1: 无效 nMoveType=2: 做为圆弧的经过位置  dAuxPosX_X: X 坐标, 单位[mm] dAuxPosX _Y: Y 坐标, 单位[mm] dAuxPosX _Z: Z 坐标, 单位[mm] dAuxPosX _Rx: Rx 坐标, 单位[°] dAuxPosX _Ry: Ry 坐标, 单位[°]

				dAuxPosX_Rz: Rz 坐标, 单位[°]
dJ1-dJ6	关节目标位置	float	-	nMoveType=0 并 nIsUseJoint=1: 使用此关节坐标作为目标关节坐标 nMoveType=0 并 nIsUseJoint=0: 此关节坐标仅作为计算逆解时选解的参考关节坐标 nMoveType=1: 无效 nMoveType=2: 无效 dJ1: 关节 1 坐标, 单位[°] dJ2: 关节 2 坐标, 单位[°] dJ3: 关节 3 坐标, 单位[°] dJ4: 关节 4 坐标, 单位[°] dJ5: 关节 5 坐标, 单位[°] dJ6: 关节 6 坐标, 单位[°]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"Base"
dVelocity	速度	float	-	运动最大速度, 关节运动时单位[°/s], 空间运动时 XYZ 单位[mm/s], Rx, Ry, Rz 单位[°/s]
dAcc	加速度	float	-	运动最大加速度, 关节运动时单位[°/s²], 空间运动时 XYZ 单位[mm/s²], Rx, Ry, Rz 单位[°/s²]
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	是否使用关节角度作为目标点, 如果 nMoveType=0 时, 则 nIsUseJoint 有效: 0: 不使用关节角度 1: 使用关节角度

nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1，则开启检测 DI 停止，路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态=nIOState 时，机器人停止运动，否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引，nIsSeek=0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态，nIsSeek=0 时无效
strCmdID	命令 ID	string	-	当前路点 ID，可以自定义，也可以按顺序设置为"1"，"2"，"3"

\* 注： HRIF\_WayPoint2 支持直线与圆弧过渡不减速功能； HRIF\_WayPoint 仅支持直线与直线过渡不减速。

#### ✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

#### 3.10.8.2. 示例

# 定义运动类型

nMoveType = 0

# 定义空间目标位置

EndPos = [420, 0, 445, 180, 0, 0, 180]

# 定义空间目标位置

AuxPos = [420, 0, 445, 180, 0, 0, 180]

# 定义关节目标位置

AcsPose = [0, 0, 90, 0, 90, 0]

# 定义工具坐标变量

sTcpName = "TCP"

# 定义用户坐标变量

sUcsName = "Base"

# 定义运动速度

dVelocity = 50

# 定义运动加速度

dAcc = 50



---

# 定义过渡半径

dRadius = 50

# 定义是否使用关节角度

nIsUseJoint= 1

# 定义是否使用检测DI 停止

nIsSeek = 0

# 定义检测的DI 索引

nIOBit = 0

# 定义检测的DI 状态

nIOState = 0

# 定义路点ID

stdCmdID = "0"

# 执行路点运动

nRet = cps.HRIF\_WayPoint2(0,0,nMoveType ,EndPose, AuxPos, AcsPose, sTcpName , sUcsName, dVelocity,  
dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)

### 3.10.9. HRIF\_MoveJ

3.10.9.1. 描述：关节运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dX-dRz	空间目标位置	float	-	nIsUseJoint=1：无效 nIsUseJoint=0：用此空间坐标作为目标位置，通过逆解计算得到关节坐标为目标关节坐标： dTcp_X：X 坐标，单位[mm] dTcp_Y：Y 坐标，单位[mm] dTcp_Z：Z 坐标，单位[mm] dTcp_Rx：Rx 坐标，单位[°] dTcp_Ry：Ry 坐标，单位[°] dTcp_Rz：Rz 坐标，单位[°]
dJ1-dJ6	关节目标位置	float	-	nIsUseJoint=1：使用此关节坐标作为目标关节坐标 nIsUseJoint=0：此关节坐标仅作为计算逆解时选解的参考关节坐标： dJ1：关节 1 坐标，单位[°] dJ2：关节 2 坐标，单位[°] dJ3：关节 3 坐标，单位[°] dJ4：关节 4 坐标，单位[°] dJ5：关节 5 坐标，单位[°] dJ6：关节 6 坐标，单位[°]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，当 nIsUseJoint=1 时无效，可使用默认名称"TCP"

sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应-当 nIsUseJoint=1 时无效，可使用默认名称"Base"
dVelocity	速度	float	-	运动最大速度，单位[°/s]
dAcc	加速度	float	-	运动最大加速度，单位[°/s²]
dRadius	过渡半径	float	-	过渡半径，单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	是否使用关节角度作为目标点，如果 nMoveType=0 时，则 nIsJoint 有效： 0：不使用关节角度 1：使用关节角度
nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1，则开启检测 DI 停止，路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态=nIOState 时，机器人停止运动，否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引，nIsSeek=0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态，nIsSeek=0 时无效
strCmdID	命令 ID	string	-	当前路点 ID，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

### 3.10.9.2. 示例

# 定义空间目标位置

```
Points = [ 0, 0, 90, 0, 90, 0]
```

# 定义关节目标位置

```
RawACSpnts = [ 0, 0, 90, 0, 90, 0]
```

# 定义工具坐标变量

```
sTcpName = "TCP"
```

---

# 定义用户坐标变量

sUcsName = "Base"

# 定义运动速度

dVelocity = 50

# 定义运动加速度

dAcc = 50

# 定义过渡半径

dRadius = 50

# 定义是否使用关节角度

nIsUseJoint= 1

# 定义是否使用检测DI 停止

nIsSeek = 0

# 定义检测的DI 索引

nIOBit = 0

# 定义检测的DI 状态

nIOState = 0

# 定义路点ID

stdCmdID = "0"

# 执行路点运动

nRet = cps.HRIF\_MoveJ(0,0, Point, RawACSpots, sTcpName , sUcsName, dVelocity, dAcc,  
dRadius,nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)

### 3.10.10. HRIF\_MoveL

3.10.10.1.描述：直线轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dX-dRz	空间目标位置	float	-	目标空间位置： dTcp_X: X 坐标，单位[mm] dTcp_Y: Y 坐标，单位[mm] dTcp_Z: Z 坐标，单位[mm] dTcp_Rx: Rx 坐标，单位[°] dTcp_Ry: Ry 坐标，单位[°] dTcp_Rz: Rz 坐标，单位[°]
dJ1-dJ6	关节参考位置	float	-	关节参考位置-建议使用目标迪卡尔坐标附近的关节坐标值： dJ1: 关节 1 坐标，单位[°] dJ2: 关节 2 坐标，单位[°] dJ3: 关节 3 坐标，单位[°] dJ4: 关节 4 坐标，单位[°] dJ5: 关节 5 坐标，单位[°] dJ6: 关节 6 坐标，单位[°]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，可使用默认名称 "TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应，可使用默认名称 "Base"
dVelocity	速度	float	-	运动最大速度，X, Y, Z 单位[mm/s], Rx, Ry, Rz 单位[°/s]

dAcc	加速度	float	-	运动最大加速度, X, Y, Z 单位[mm/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ]
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1, 则开启检测 DI 停止, 路 点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态=nIOState 时, 机器人停止运动, 否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引, nIsSeek=0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态, , nIsSeek=0 时无效
strCmdID	命令 ID	string	-	当前路点 ID, 可以自定义, 也可以按顺序 设置为"1", "2", "3"

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功 nRet>0: 返回调用失败的错误码

### 3.10.10.2.示例

#### # 定义空间目标位置

```
Point = [ 420, 0, 445, 180, 0, 180]
```

#### # 定义关节目标位置

```
RawACSpots = [ 0, 0, 90, 0, 90, 0]
```

#### # 定义工具坐标变量

```
sTcpName = "TCP"
```

#### # 定义用户坐标变量

```
sUcsName = "Base"
```

#### # 定义运动速度

```
dVelocity = 50
```

#### # 定义运动加速度

```
dAcc = 50
```

#### # 定义过渡半径

```
dRadius = 50
```

---

# 定义是否使用检测DI 停止

nIsSeek = 0

# 定义检测的DI 索引

nIOBit = 0

# 定义检测的DI 状态

nIOState = 0

# 定义路点ID

stdCmdID = "0"

# 执行路点运动

nRet = cps.HRIF\_MoveL(0,0, Point, RawACSpoints sTcpName, sUcsName, dVelocity, dAcc, dRadius,nIsSeek,  
nIOBit, nIOState, strCmdID)

### 3.10.11. HRIF\_MoveC

3.10.11.1.描述：圆弧轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dStartPoint	圆弧起始点位置	list[float]	-	圆弧起始点位置： dStartPos_X: X 坐标，单位[mm] dStartPos_Y: Y 坐标，单位[mm] dStartPos_Z: Z 坐标，单位[mm] dStartPos_Rx: Rx 坐标，单位[°] dStartPos_Ry: Ry 坐标，单位[°] dStartPos_Rz: Rz 坐标，单位[°]
dAuxPoint	圆弧经过点位置	list[float]	-	圆弧经过点位置： dAuxPos_X: X 坐标，单位[mm] dAuxPos_Y: Y 坐标，单位[mm] dAuxPos_Z: Z 坐标，单位[mm] dAuxPos_Rx: Rx 坐标，单位[°] dAuxPos_Ry: Ry 坐标，单位[°] dAuxPos_Rz: Rz 坐标，单位[°]
dEndPoint	圆弧结束点位置	list[float]	-	圆弧结束点位置，如果是用整圆时，结速点也是圆上的一个经过点，整圆跑完后才停止： dEndPos_X: X 坐标，单位[mm] dEndPos_Y: Y 坐标，单位[mm] dEndPos_Z: Z 坐标，单位[mm] dEndPos_Rx: Rx 坐标，单位[°] dEndPos_Ry: Ry 坐标，单位[°] dEndPos_Rz: Rz 坐标，单位[°]



nFixedPosure	是否固定姿态	int	0/1	圆弧整个运动过程中是否保持姿态不变： 0：固定姿态 1：不固定姿态
nMoveCType	圆弧类型	int	0/1	0：整圆 1：圆弧
dRadLen	弧长	float	-	当使用圆弧运动时无效，通过三个点位确定圆弧路径，当使用整圆运动时表示整圆的圈数，小数部分无效。
dVelocity	速度	float	-	运动最大速度，X，Y，Z 单位[mm/s]， Rx，Ry，Rz 单位[°/s]
dAcc	加速度	float	-	运动最大加速度，X，Y，Z 单位[mm/s²]， Rx，Ry，Rz 单位[°/s²]
dRadius	过渡半径	float	-	过渡半径，单位[mm]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，当 nIsUseJoint=1 时无效，可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应，当 nIsUseJoint=1 时无效，可使用默认名称"Base"
strCmdID	命令 ID	string	-	当前路点 ID，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

### 3.10.11.2.示例

#### # 圆弧起始点位置

```
dStartPoint = [420, 0, 445, 180, 0, 180]
```

#### # 圆弧经过点位置

---

```
dAuxPoint = [420, 50, 445, 180, 0, 180]
# 圆弧结束点位置

dEndPoint = [470, 0, 445, 180, 0, 180]
# 是否固定姿态

nFixedPosure = 0
# 圆弧类型

nMoveCType = 0
# 整圆圈数

dRadLen = 1
# 定义运动速度

dVelocity = 50
# 定义运动加速度

dAcc = 50
# 定义过渡半径

dRadius = 50
# 定义工具坐标变量

sTcpName = "TCP"
# 定义用户坐标变量

sUcsName = "Base"
# 定义路点ID

stdCmdID = "0"
# 执行路点运动

nRet = cps.HRIF_MoveC(0,0,dStartPoint , dAuxPoint, dEndPoint,
nFixedPosure, nMoveCType, dRadLen, dVelocity, dAcc, dRadius,sTcpName , sUcsName, strCmdID)
```

### 3.10.12. HRIF\_MoveZ

3.10.12.1.描述: Z 型轨迹运动

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值=0
dStartPos	Z 型起始点位置	float	-	Z 型起始点位置: dStartPos_X: X 坐标, 单位[mm] dStartPos_Y: Y 坐标, 单位[mm] dStartPos_Z: Z 坐标, 单位[mm] dStartPos_Rx: Rx 坐标, 单位[°] dStartPos_Ry: Ry 坐标, 单位[°] dStartPos_Rz: Rz 坐标, 单位[°]
dEndPos	Z 型结束点位置	float	-	Z 型结束点位置: dEndPos_X: X 坐标, 单位[mm] dEndPos_Y: Y 坐标, 单位[mm] dEndPos_Z: Z 坐标, 单位[mm] dEndPos_Rx: Rx 坐标, 单位[°] dEndPos_Ry: Ry 坐标, 单位[°] dEndPos_Rz: Rz 坐标, 单位[°]
dPlanePos	轨迹确定平面点位置	float	-	Z 型轨迹平面点位置: dPlanePosPos_X: X 坐标, 单位[mm] dPlanePosPos_Y: Y 坐标, 单位[mm] dPlanePosPos_Z: Z 坐标, 单位[mm] dPlanePosPos_Rx: Rx 坐标, 单位[°] dPlanePosPos_Ry: Ry 坐标, 单位[°] dPlanePosPos_Rz: Rz 坐标, 单位[°]
dVelocity	速度	float	-	运动最大速度, X, Y, Z 单位[mm/s], Rx, Ry, Rz 单位[°/s]

dAcc	加速度	float	-	运动最大加速度, X, Y, Z 单位 [mm/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ]
dWidth	宽度	float	-	轨迹宽度
dDensity	密度	float	-	轨迹密度, 当不使用密度时根据速度自动 计算密度
nEnableDensity	是否使用密度	int	-	是否使用密度: 0: 不使用 1: 使用
nEnablePlane	是否使用平面点	int	-	是否使用平面点, 不使用时根据选择的用 户坐标确定 XYZ 平面: 0: 不使用 1: 使用
nEnableWaiTime	是否开启转折点等待 时间	int	-	是否开启转折点等待时间: 0: 不使用 1: 使用
nPosiTime	正向转折点等待时间	int	-	正向转折点等待时间, 单位[ms]
nNegaTime	负向转折点等待时间	int	-	负向转折点等待时间, 单位[ms]
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称, 与 示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称, 与 示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"Base"
strCmdID	命令 ID	string	-	当前路点 ID, 可以自定义, 也可以按顺序 设置为"1", "2", "3"

**✓ 返回值**

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0: 返回函数调用成功

				nRet>0：返回调用失败的错误码
--	--	--	--	-------------------

### 3.10.12.2.示例

```

# 起始点位置

dStartPos = [420, 0, 445, 180, 0, 180]

# 结束点位置

dEndPos = [420, 100, 445, 180, 0, 180]

# 确定轨迹平面点位置

dPlanePos = [470, 50, 445, 180, 0, 180]

# 定义运动速度

dVelocity = 50

# 定义运动加速度

dAcc = 2500

# 宽度

dWidth = 50

# 密度

dDensity = 10

# 使用密度

nEnableDensity = 1

# 使用平面点

nEnablePlane = 1

# 是否在转折点等待-不等待

nEnableWaiTime = 0

# 正向转折点等待时间

nPosiTime = 0

# 负向转折点等待时间

nNegaTime = 0

# 定义过渡半径

dRadius = 5

# 定义工具坐标变量

sTcpName = "TCP "

```

---

# 定义用户坐标变量

sUcsName = "Base"

# 定义路点ID

stdCmdID = "0"

# 执行路点运动

nRet = cps.HRIF\_MoveZ(0,0,dStartPos, dEndPos, dPlanePos, dVelocity, dAcc, dWidth, dDensity,  
nEnableDensity, nEnablePlane, nEnableWaiTime, nPosiTime, nNegaTime, dRadius, sTcpName, sUcsName,  
strCmdID)

### 3.10.13. HRIF\_MoveE

3.10.13.1.描述：椭圆型轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dP1	示教位置 1	list[float]	-	示教位置 1: dP1_X: X 坐标, 单位[mm] dP1_Y: Y 坐标, 单位[mm] dP1_Z: Z 坐标, 单位[mm] dP1_Rx: Rx 坐标, 单位[°] dP1_Ry: Ry 坐标, 单位[°] dP1_Rz: Rz 坐标, 单位[°]
dP2	示教位置 2	list[float]	-	示教位置 2: dP2_X: X 坐标, 单位[mm] dP2_Y: Y 坐标, 单位[mm] dP2_Z: Z 坐标, 单位[mm] dP2_Rx: Rx 坐标, 单位[°] dP2_Ry: Ry 坐标, 单位[°] dP2_Rz: Rz 坐标, 单位[°]
dP3	示教位置 3	list[float]	-	示教位置 3: dP3_X: X 坐标, 单位[mm] dP3_Y: Y 坐标, 单位[mm] dP3_Z: Z 坐标, 单位[mm] dP3_Rx: Rx 坐标, 单位[°] dP3_Ry: Ry 坐标, 单位[°] dP3_Rz: Rz 坐标, 单位[°]
dP4	示教位置 4	list[float]	-	示教位置 4: dP4_X: X 坐标, 单位[mm]

				dP4_Y: Y 坐标, 单位[mm] dP4_Z: Z 坐标, 单位[mm] dP4_Rx: Rx 坐标, 单位[°] dP4_Ry: Ry 坐标, 单位[°] dP4_Rz: Rz 坐标, 单位[°]
dP5	示教位置 5	list[float]	-	示教位置 5: dP5_X: X 坐标, 单位[mm] dP5_Y: Y 坐标, 单位[mm] dP5_Z: Z 坐标, 单位[mm] dP5_Rx: Rx 坐标, 单位[°] dP5_Ry: Ry 坐标, 单位[°] dP5_Rz: Rz 坐标, 单位[°]
nOrientMode	运动模式	int	0/1	0: 椭圆圆弧 1: 整个椭圆
nMoveType	运动类型	int	0/1	0: 不使用固定姿态 1: 使用固定姿态
dArcLength	弧长	float	-	弧长, 单位[°]
dVelocity	速度	float	-	运动最大速度, X, Y, Z 单位[mm/s], Rx, Ry, Rz 单位[°/s]
dAcc	加速度	float	-	运动最大加速度, X, Y, Z 单位[mm/s²], Rx, Ry, Rz 单位[°/s²]
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可使用默认名称"Base"
strCmdID	命令 ID	string	-	当前路点 ID, 可以自定义, 也可以按顺序设置为"1", "2", "3"



**✓ 返回值**

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

**3.10.13.2.示例**
**# 示教点1**

dP1[6] = [420,0,445,180,0,180]

**# 示教点2**

dP2[6] = [460,0,445,180,0,180]

**# 示教点3**

dP3[6] = [480,10,445,180,0,180]

**# 示教点4**

dP4[6] = [460,20,445,180,0,180]

**# 示教点5**

dP5[6] = [420,20,445,180,0,180]

**# 运动模式**

nOrientMode = 0

**# 运动类型**

nMoveType = 1

**# 弧长**

dArcLength = 360

**# 定义运动速度**

dVelocity = 50

**# 定义运动加速度**

dAcc = 2500

**# 定义过渡半径**

dRadius = 5

**# 定义工具坐标变量**

sTcpName = "TCP"

**# 定义用户坐标变量**

---

```
sUcsName = "Base"
```

```
# 定义路点ID
```

```
stdCmdID = "0"
```

```
# 执行椭圆运动
```

```
nRet = cps.HRIF_MoveE(0,0,dP1, dP2, dP3,dP4, dP5,  
nOrientMode,nMoveType,dArcLength,dVelocity,dAcc,dRadius, sTcpName, sUcsName, strCmdID)
```

### 3.10.14. HRIF\_MoveS

3.10.14.1.描述：阿基米德螺旋线运动，初始半径为固定 1mm 。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dSpiralIncrement	增量半径	float	>0	螺旋运动每圈增量半径，单位[mm]
dSpiralDiameter	结束半径	float	>1	螺旋运动结束半径，单位[mm]
dVelocity	速度	float	-	运动最大速度，单位[°/s]
dAcc	加速度	float	-	运动最大加速度，单位[°/s²]
dRadius	过渡半径	float	-	过渡半径，单位[mm]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，当 nIsUseJoint=1 时无效，可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应，当 nIsUseJoint=1 时无效，可使用默认名称 "Base"
strCmdID	命令 ID	string	-	当前路点 ID，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.10.14.2.示例

# 定义增量半径

dSpiralIncrement = 1

# 定义结束半径

dSpiralDiameter = 5

---

# 定义工具坐标变量

sTcpName = "TCP"

# 定义用户坐标变量

sUcsName = "Base"

# 定义运动速度

dVelocity = 50

# 定义运动加速度

dAcc = 50

# 定义过渡半径

dRadius = 50

# 定义路点ID

stdCmdID = "0"

# 执行螺旋轨迹运动

nRet= cps.HRIF\_MoveS(0,0, dSpiralIncrement, dSpiralDiameter, sTcpName, sUcsName, dVelocity, dAcc,  
dRadius, sCmdID)

## 3.11.连续轨迹运动类控制指令

### 3.11.1. HRIF\_StartPushMovePathJ

3.11.1.1. 描述：初始化关节连续轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值=0
sTrackName	轨迹名称	string	-	轨迹名称
dSpeedRatio	轨迹运动速度比	float	0~1	轨迹运动速度比
dRadius	过渡半径	float	$\geq 0$	过渡半径,单位[mm]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	$>0$ 的整型值	nRet=0 : 返回函数调用成功 nRet $>0$ : 返回调用失败的错误码

3.11.1.2. 示例

# 轨迹名称

sTrackName = "Path1"

# 速度比

dSpeedRatio = 0.5

# 过渡半径

dRadius = 2

# 初始化关节连续轨迹运动

nRet = cps.HRIF\_StartPushMovePathJ(0,0,sTrackName, dSpeedRatio, dRadius)

### 3.11.2. HRIF\_PushMovePathJ

3.11.2.1. 描述：下发运动轨迹点位。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称
paramsJ	关节点位	list[float]	-	目标关节位置： dJ1：关节 1 位置，单位[°] dJ2：关节 2 位置，单位[°] dJ3：关节 3 位置，单位[°] dJ4：关节 4 位置，单位[°] dJ5：关节 5 位置，单位[°] dJ6：关节 6 位置，单位[°]

\* 注：调用 HRIF\_StartPushMovePath 后，可多次调用此函数，一般情况下点位数量需要大于 4。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.2.2. 示例

# 轨迹名称

sTrackName = "Path1"

# 目标关节位置

paramsJ = [0,0,90,0,0,0]

# 下发关节点位

nRet = cps.HRIF\_PushMovePathJ (0,0,sTrackName, paramsJ)

### 3.11.3. HRIF\_EndPushMovePathJ

3.11.3.1. 描述：轨迹下发完成并开始计算轨迹。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.3.2. 示例

# 轨迹名称

sTrackName = "Path1"

# 下发完成, 开始计算轨迹

nRet = cps.HRIF\_EndPushMovePathJ(0,0,sTrackName)

### 3.11.4. HRIF\_MovePathJ

3.11.4.1. 描述：运动指定的轨迹。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.4.2. 示例

# 轨迹名称

sTrackName = "Path1"

# 运动轨迹

nRet = cps.HRIF\_MovePathJ(0,0,sTrackName)



### 3.11.5. HRIF\_ReadMovePathJState

3.11.5.1. 描述：读取当前的轨迹状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
sTrackName	轨迹名称	string	-	轨迹名称

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	轨迹状态	int	0~5	当前轨迹状态： 0：轨迹未开始示教，调用 HRIF_StartPushMovePathJ 后切换到此状态 1：轨迹示教中，调用 HRIF_PushMovePathJ 后切换到此状态 2：轨迹计算中，调用 HRIF_EndPushMovePathJ 后切换到此状态 3：轨迹计算完成，计算完成后切换到此状态，可以开始运动。 4：结束示教 5：计算错误，需要检查点位是否合理，然后修正后重新示教。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.5.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 轨迹名称

```
sTrackName = "Path1"
```

# 读取轨迹状态

---

```
nRet = cps.HRIF_ReadMovePathJState(0,0,sTrackName, result)
```

```
# 读取到的当前轨迹状态
```

```
nState = int(result[0])
```

### 3.11.6. HRIF\_UpdateMovePathJName

3.11.6.1. 描述：更新指定轨迹的名称。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹原名称	string	-	轨迹原名称
sTrackNewName	更新的轨迹名称	string	-	更新的轨迹名称

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.6.2. 示例

# 轨迹名称

sTrackName = "Path1"

# 更新的轨迹名称

sTrackNewName = "Path2"

# 重命名轨迹名称

nRet = cps.HRIF\_UpdateMovePathJName(0,0,sTrackName, sTrackNewName)

### 3.11.7. HRIF\_DelMovePathJ

3.11.7.1. 描述：删除指定轨迹。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.7.2. 示例

# 轨迹名称

sTrackName = "Path1"

# 删除轨迹

nRet = cps.HRIF\_DelMovePathJ(0,0,sTrackName)

### 3.11.8. HRIF\_ReadTrackProcess

3.11.8.1. 描述：读取当前的轨迹运动进度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]

\* 注：此接口仅对 HRIF\_MovePathL 有效，对 HRIF\_MovePathJ 无效。

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[0]	轨迹运行进度	float	0~1	当前轨迹运动进度，当前运动进度(0-1),>0.999999 表示运动完成
result[1]	点位索引	int	>0 的整型值	当前轨迹运动到哪一个点索引

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.8.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 读取轨迹状态

```
nRet = cps.HRIF_ReadTrackProcess(0,0,result)
```

# 轨迹运行进度

```
dProcess = float(result[0])
```

# 点位索引

```
nIndex = int(result[1])
```

### 3.11.9. HRIF\_InitMovePathL

3.11.9.1. 描述：初始化空间轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称，目前空间轨迹运动的轨迹名称没有作用 可以任意定义，同一轨迹运行需要同执行 HRIF_InitMovePathL, HRIF_PushMovePaths, HRIF_EMovePathL 调用 HRIF_MovePathL 后会计算完轨迹后直接开始 运动，计算时间 2-4s 左右，根据实际轨迹大小确定
dVelocity	轨迹运动速度	float	-	轨迹运动速度，单位[mm/s]
dAcc	轨迹运动加速度	float	-	轨迹运动加速度，单位[mm/s <sup>2</sup> ]
dJerk	轨迹运动加加速度	float	-	轨迹运动加加速度，单位[mm/s <sup>3</sup> ]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应-可使用默认名称"Base"

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.9.2. 示例

# 轨迹名称

sTrackName = "Path1"

# 定义运动速度

dVelocity = 100

---

# 定义运动加速度

dAcc = 2500

# 定义运动加加速度

dJerk = 1000000

# 定义工具坐标变量

sTcpName = "TCP"

# 定义用户坐标变量

sUcsName = "Base"

# 初始化关节连续轨迹运动

nRet = cps.HRIF\_InitMovePathL(0,0,sTrackName, dVelocity, dAcc, dJerk, sUcsName, sTcpName)

### 3.11.10. HRIF\_PushMovePathL

3.11.10.1.描述：下发运动轨迹点位。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称
paramPcs	空间点位	list[float]	-	目标空间位置： dX: X 坐标，单位[mm] dY: Y 坐标，单位[mm] dZ: Z 坐标，单位[mm] dRx: Rx 坐标，单位[°] dRy: Ry 坐标，单位[°] dRz: Rz 坐标，单位[°]

\* 注：调用 HRIF\_InitPushMovePathL 后，可多次调用此函数，一般情况下点位数量需要超过 4 个。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.10.2.示例

# 轨迹名称

sTrackName = "Path1"

# 定义空间目标位置

paramPcs = [420, 0, 445, 180, 0, 180]

# 下发空间目标点位

nRet = cps.HRIF\_PushMovePathL(0,0,sTrackName, paramPcs)



### 3.11.11. HRIF\_PushMovePaths

3.11.11.1.描述：批量下发轨迹点位，调用一次可下发多个点位数据。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称
nMoveType	点位类型	int	0/1	0：下发关节点位，与 HRIF_MovePathJ 共用 1：下发空间点位，与 HRIF_MovePathJ 共用
nPointsSize	点位数量	int	-	点位数量，必须与 sPoints 里的数量一致
sPoints	点位数据	list[float]	-	点位数据，每个数据以逗号分隔

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.11.2.示例

# 轨迹名称

sTrackName = "Path1"

# 运动类型 MovePathL

nMoveType = 1

# 点位数量

nPointsSize = 6

sPoints = [420,0,445,180,0,180,420,10,445,180,0,180,420,20,445,180,0,180,  
420,30,445,180,0,180,420,40,445,180,0,180,420,50,445,180,0,180]

# 下发空间目标点位

nRet = cps.HRIF\_PushMovePaths(0,0,sTrackName, nMoveType, nPointsSize, sPoints)

# 轨迹名称

sTrackName = "Path2"

# 运动类型 MovePathJ

---

```
nMoveType = 0
```

```
# 点位数量
```

```
nPointsSize = 9
```

```
sPoints = [0,0,0,0,0,0,0,5,0,5,0,0,10,0,10,0,  
0,0,15,0,15,0,0,0,20,0,20,0,0,0,25,0,25,0,  
0,0,30,0,30,0,0,0,34,0,34,0,0,0,40,0,40,0]
```

```
# 下发关节点位
```

```
nRet = cps.HRIF_PushMovePaths(0,0,sTrackName, nMoveType, nPointsSize, sPoints)
```

### 3.11.12. HRIF\_MovePathL

3.11.12.1.描述：执行空间坐标轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

\* 注：调用 HRIF\_MovePathL 后会计算完轨迹后直接开始运动，计算时间 2-4s 左右，根据实际轨迹大小确定。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.12.2.示例

# 轨迹名称

sTrackName = "Path2"

# 开始空间连续轨迹运动

nRet = cps.HRIF\_MovePathL(0,0,sTrackName)

### 3.11.13. HRIF\_SetMovePathOverride

3.11.13.1.描述：设置 MovePath 速度比，MovePath 运动中设置有效。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
MovePathOverride	设置速度比	Float	0.01~1	需要设置的速度比(0.01~1)

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.11.13.2.示例

# 速度比

MovePathOverride= 0.01

#设置 MovePath 速度比

nRet = cps.HRIF\_SetMovePathOverride(0,0,MovePathOverride)

## 3.12.Servo 运动类控制指令

### 3.12.1. HRIF\_StartServo

3.12.1.1. 描述：启动机器人在线控制（ServoJ 或 ServoP）时，设定位置固定更新的周期和前瞻时间。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dServoTime	更新周期	float	>0	固定更新的周期时间，单位[s]
dLookaheadTime	前瞻时间	float	>0	前瞻时间，单位[s]

\* 注：HRIF\_StartServo/HRIF\_PushServoJ/HRIF\_PushServoP 为一套接口，与其他 Servo 指令不共用。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.12.1.2. 示例

# 周期

dServoTime = 0.02

# 前瞻时间

dLookaheadTime = 0.2

# 启动机器人在线控制

nRet = cps.HRIF\_StartServo(0,0,dServoTime, dLookaheadTime)

### 3.12.2. HRIF\_PushServoJ

3.12.2.1. 描述：在线关节位置命令控制，以 StartServo 设定的固定更新时间发送关节位置，机器人将实时的跟踪关节位置指令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dAcs	关节点位	list[float]	-	目标关节位置： dJ1：关节 1 位置，单位[°] dJ2：关节 2 位置，单位[°] dJ3：关节 3 位置，单位[°] dJ4：关节 4 位置，单位[°] dJ5：关节 5 位置，单位[°] dJ6：关节 6 位置，单位[°]

\* 注：HRIF\_StartServo/HRIF\_PushServoJ/HRIF\_PushServoP 为一套接口，与其他 Servo 指令不共用。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.12.2.2. 示例

# 目标关节位置

dAcs = [0, 0, 0, 0, 0, 0]

# 在线关节位置命令控制

nRet = cps.HRIF\_PushServoJ(0,0, dAcs )

### 3.12.3. HRIF\_PushServoP

3.12.3.1. 描述：在线末端 TCP 位置命令控制，以 StartServo 设定的固定更新时间发送 TCP 位置，机器人将实时的跟踪目标 TCP 位置逆运算转换后的关节位置指令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dPcs	空间点位	list[float]	-	目标空间位置： dPcs[0]: X 坐标，单位[mm] dPcs[1]: Y 坐标，单位[mm] dPcs[2]: Z 坐标，单位[mm] dPcs[3]: Rx 坐标，单位[°] dPcs[4]: Ry 坐标，单位[°] dPcs[5]: Rz 坐标，单位[°]
dTcp	工具坐标	list[float]	-	目标位置对应的工具坐标： dTcp[0]: X 坐标，单位[mm] dTcp[1]: Y 坐标，单位[mm] dTcp[2]: Z 坐标，单位[mm] dTcp[3]: Rx 坐标，单位[°] dTcp[4]: Ry 坐标，单位[°] dTcp[5]: Rz 坐标，单位[°]
dUcs	用户坐标	list[float]	-	目标位置对应的用户坐标： dUcs[0]: X 坐标，单位[mm] dUcs[1]: Y 坐标，单位[mm] dUcs[2]: Z 坐标，单位[mm] dUcs[3]: Rx 坐标，单位[°] dUcs[4]: Ry 坐标，单位[°] dUcs[5]: Rz 坐标，单位[°]

\* 注：HRIF\_StartServo/HRIF\_PushServoJ/HRIF\_PushServoP 为一套接口，与其他 Servo 指令不共用。

## ✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

## 3.12.3.2. 示例

# 空间目标位置

dPcs = [420, 0, 445, 180, 0, 180]

# 定义工具坐标变量

dTcp = [0, 0, 0, 0, 0, 0]

# 定义用户坐标变量

dUcs = [0, 0, 0, 0, 0, 0]

# 在线空间位置命令控制

nRet = cps.HRIF\_PushServoP(0,0,dPcs, dTcp, dUcs)



### 3.13.相对跟踪运动类控制指令

#### 3.13.1. HRIF\_SetMoveTraceParams

3.13.1.1. 描述：设置相对跟踪运动控制参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nState	跟踪状态	int	0/1	0：关闭相对跟踪运动 1：开启相对跟踪运动
dDistance	相对跟踪运动保持的相对距离	float	>0	相对跟踪运动保持的相对距离
dAwayVelocity	相对跟踪的运动的远离探寻速度	float	>0	相对跟踪的运动的远离探寻速度
dBackVelocity	相对跟踪的运动的靠近探寻速度	float	>0	相对跟踪的运动的靠近探寻速度

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.13.1.2. 示例

*# 设置跟踪状态和保持的相对距离*

nState = 1

dDistance = 100

*# 相对跟踪的运动的探寻速度*

dAwayVelocity = 50

dBackVelocity = 50

*# 设置相对跟踪运动控制参数并开启相对跟踪运动*

nRet = cps.HRIF\_SetMoveTraceParams(0,0,nState, dDistance,dAwayVelocity,dBackVelocity)

### 3.13.2. HRIF\_SetMoveTraceInitParams

3.13.2.1. 描述：设置相对跟踪运动初始化参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
dK	传感器计算参数	float	-	计算公式 $y = dK * x + dB$
dB	传感器计算参数	float	-	计算公式 $y = dK * x + dB$
dMaxLimit	激光传感器检测 距离最大值	float	-	激光传感器检测距离最大值
dMinLimit	激光传感器检测 距离最小值	float	-	激光传感器检测距离最小值

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.13.2.2. 示例

# 传感器计算参数

dK= -14

# 传感器计算参数

dB= 135

# 激光传感器检测距离最大值

dMaxLimit = 130

# 激光传感器检测距离最小值

dMinLimit = 65

# 设置跟踪状态初始化参数

nRet = cps.HRIF\_SetMoveTraceInitParams(0,0,dK, dB, dMaxLimit, dMinLimit)

### 3.13.3. HRIF\_SetMoveTraceUcs

3.13.3.1. 描述：设置相对跟踪运动的跟踪探寻方向。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
direction	跟踪探寻方向	list[float]	>-180,<180	目标空间位置： dX： 无效，可设置为 0 dY： 无效，可设置为 0 dZ： 无效，可设置为 0 dRx： Rx 方向，单位[°] dRy： Ry 方向，单位[°] dRz： Rz 方向，单位[°]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.13.3.2. 示例

# 设置跟踪方向

```
direction = [420, 0, 445, 180, 0, 180]
```

# 设置跟踪方向

```
nRet = cps.HRIF_SetMoveTraceUcs(0,0,direction)
```

### 3.13.4. HRIF\_SetTrackingState

3.13.4.1. 描述：设置传送带跟踪运动状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nState	跟踪状态	int	0/1	0：关闭传送带跟踪运动 1：开启传送带跟踪运动

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.13.4.2. 示例

# 设置传送带跟踪开启

nState = 1

# 开启传送带跟踪

nRet = cps.HRIF\_SetTrackingState(0,0,nState)

## 3.14.其他指令

### 3.14.1. HRIF\_HRAppCmd

3.14.1.1. 描述：执行插件 App 命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
sCmdName	命令名称	string	0/1	命令名称
sParams	参数列表	string	-	参数列表

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.14.1.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 插件名称

```
sCmdName = 'FT_Plugin'
```

# 插件指令与参数

```
sParams = ['FT_reset']
```

# 发送插件命令

```
nRet = cps.HRIF_HRAppCmd(0,0,sCmdName, sParams, result)
```

### 3.14.2. HRIF\_WriteEndHoldingRegisters

3.14.2.1. 描述：写末端连接的 Modbus 从站寄存器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
nSlaveID	从站 ID	int	0/1	从站 ID
nFunction	功能码	int	>0	功能码: 0x01-读线圈寄存器 0x02-读线圈离散输入寄存器 0x03-读保持寄存器 0x04-读输入寄存器 0x05-写单个线圈寄存器 0x06-写单个保持寄存器 0x0f-写多个线圈寄存器 0x10-写多个保持寄存器
nRegAddr	寄存器地址	int	>0	寄存器起始地址
nRegCount	寄存器数量	int	>0	寄存器数量
nData	寄存器数据	list[int]	-	寄存器数据

\* 注：末端为 EtherCAT 总线版本 IO 时有效。

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码

3.14.2.2. 示例

# 设置从站ID=1

nSlaveID = 1

# 设置功能码

---

nFunction = 16

# 设置寄存器起始地址

nRegAddr = 100

# 设置寄存器数量

nRegCount = 2

# 设置寄存器数据

nData=[196,34465]

nRet = cps.HRIF\_WriteEndHoldingRegisters(0,0,nSlaveID, nFunction, nRegAddr, nRegCount, nData)

### 3.14.3. HRIF\_ReadEndHoldingRegisters

3.14.3.1. 描述：读末端连接的 Modbus 从站寄存器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	int	0~5	电箱 ID 号，默认值=0
rbtID	机器人 ID	int	0~5	机器人 ID 号，默认值=0
result	返回值	list		传入空列表，result = [ ]
nSlaveID	从站 ID	int	0/1	从站 ID
nFunction	功能码	int	>0	功能码： 0x01-读线圈寄存器 0x02-读线圈离散输入寄存器 0x03-读保持寄存器 0x04-读输入寄存器 0x05-写单个线圈寄存器 0x06-写单个保持寄存器 0x0f-写多个线圈寄存器 0x10-写多个保持寄存器
nRegAddr	寄存器地址	int	>0	寄存器起始地址
nRegCount	寄存器数量	int	>0	寄存器数量

\* 注：末端为 EtherCAT 总线版本 IO 时有效。

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result	寄存器数据	list[int]	-	寄存器数据

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>0 的整型值	nRet=0：返回函数调用成功 nRet>0：返回调用失败的错误码



## 3.14.3.2. 示例

```
# 设置从站ID=1
nSlaveID = 1
# 设置功能码
nFunction = 3
# 设置寄存器起始地址
nRegAddr = 100
# 设置寄存器数量
nRegCount = 4
# 定义返回值空列表
result = [ ]
# 读取寄存器数据
nRet = cps.HRIF_ReadEndHoldingRegisters(0,0,nSlaveID, nFunction, nRegAddr, nRegCount, nRegCount,
result)
for i in range(len(result)):
    print(result[i])
```

---

## 第四章 附录

### 4.1. 参考文件

1. 错误码及状态机描述请参阅 [《HansRobot\\_ErrorCode.docx》](#)。
2. 大族机器人通信协议使用请参阅 [《HanRobotV5 控制通信协议接口》](#)。
3. 大族机器人 C# SDK 使用请参阅 [《HanRobot Library C#》](#)。
4. 大族机器人 Java SDK 使用请参阅 [《HanRobot Library Java》](#)。
5. 大族机器人 C++ SDK 使用请参阅 [《HanRobot Library C++》](#)。