



HAN'S ROBOT  
大族机器人

# HansRobot Library

## Python

(Elfin 系列机器人通用)



## 修订记录

文件状态	[ ]草稿 [√] 正式发布 [ ]正在修改		
当前版本	V1.0.10.0		
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	A 新增接口
V1.0.0.2	2022.08.05	20220720.207	A 新增接口
V1.0.0.3	2022.08.28	20220825.214	A 新增 HRIF_SetToolMotion 接口
V1.0.0.4	2023.02.13	HR5.50.3	A 新增 HRIF_ReadOverride 接口
V1.0.0.5	2023.02.13	HR5.50.3	A 新增 HRIF_ReadPointByName 接口
V1.0.0.6	2023.02.23	HR5.50.4	M 修改 HRIF_WayPointRel 接口参数 nIsUseJoint 的解释
V1.0.1.0	2023.02.23	HR5.50.3	A 新增 HRIF_GetErrorCodeStr 接口
V1.0.2.0	2023.02.23	HR5.50.3	A 新增位置跟踪接口 HRIF_SetPoseTrackingMaxMotionLimit HRIF_SetPoseTrackingPIDParams HRIF_SetPoseTrackingTargetPos HRIF_SetPoseTrackingState HRIF_SetUpdateTrackingPose
V1.0.2.1	2023.03.21	HR5.51.0	M 修改接口 HRIF_IsBlendingDone 接口 M 修改错别字
V1.0.2.2	2023.08.22	HR5.50.3	ReadHoldingRegisters 和 WriteHoldingRegisters 的参数 nSlaveID 的取值范围改为 1~255
V1.0.3.0	2024.04.15	HR6.3.7	A 新增接口 HRIF_SetSimulation HRIF_OpenBrake HRIF_CloseBrake HRIF_ReadBrakeStatus HRIF_MoveToSS HRIF_SetCollideLevel HRIF_ReadMaxPayload HRIF_ReadPayload HRIF_CalUcsPlane HRIF_CalUcsLine HRIF_CalTcp3P

			HRIF_CalTcp4P HRIF_ConfigTCP HRIF_ConfigUCS HRIF_HRSetMaxSearchDistance M 修改 HRIF_HRApp 接口
V1.0.4.0	2024.08.27	HR6.3.9	A 新增接口 HRIF_EnableEndBTN HRIF_ReadPointList HRIF_ReadTCPList HRIF_ReadUSCList HRIF_SetFTFreeDriveSpeedMode HRIF_ReadtFTFreeDriveSpeedMode HRIF_ReadFTCabData HRIF_SetFreeDriveMotionFreedom HRIF_SetFTFreeFactor HRIF_SetFreeDriveCompensateForce HRIF_ReadFTMotionFreedom HRIF_SetSteadyContactDeviationRange HRIF_InitPath HRIF_PushPathPoints HRIF_EndPushPathPoints HRIF_DelPath HRIF_ReadPathList HRIF_ReadPathInfo HRIF_UpdataPathName HRIF_ReadPathState HRIF_MoveLinearWeave HRIF_MoveCircularWeave HRIF_ReadServoEsJState M 修改接口 HRIF_WayPoint2 HRIF_SetScriptForceControlState HRIF_MoveS
V1.0.5.0	2024.09.02	HR6.3.9	A 新增接口 HRIF_MovePathJOL HRIF_GetMovePathJOLIndex HRIF_cdsSetIO HRIF_SetPoseTrackingStopTimeOut HRIF_EnterSafeGuard HRIF_AddSafePlane HRIF_DelSafePlane HRIF_ReadSafePlaneList HRIF_ReadSafePlane HRIF_UpdateSafePlane HRIF_SetDepthThresholdForDampingArea
		HR6.4SpeedL.8 HR6.5.1	HRIF_SpeedJ HRIF_SpeedL HRIF_ReadTriStageSwicth HRIF_SetTriStageSwicth
V1.0.5.1	2024.10.10	HR6.3.9	M 修改接口 HRIF_ReadCurFSM HRIF_MoveLinearWeave HRIF_MoveCircularWeave
V1.0.6.0	2024.10.11	HR6.4.12	A 新增接口 HRIF_SwitchScript HRIF_ReadDefaultScript
		HR7.1.5	A 新增 nJoint 相关接口
V1.0.6.1	2024.10.29	HR5.49.7	M 修改接口 HRIF_MoveE 修改参数说明

V1.0.6.2	2024.11.19	-	M 修改接口文档 HRIF_ReadBrakeStatus HRIF_StopScript HRIF_SetToolMotion
V1.0.6.3	2024.12.10	HR6.3.4	M HRIF_SetForceDataLimit 修改力限制范围接口
V1.0.7.0	2024.12.27	HR6.4.10	A 新增接口 HRIF_MoveAlignToZ HRIF_SetBaseInstallingAngle HRIF_GetBaseInstallingAngle
V1.0.8.0	2025.02.05	HR6.3.9	A 新增接口 HRIF_SetLastCalibParams HRIF_GetLastCalibParams
V1.0.8.1	2025.02.24	HR6.5.10	M 修改力控相关接口参数范围
V1.0.9.0	2025.03.05	HR6.4.13	A 新增接口 HRIF_GetLoadIdentifyResult HRIF_LoadIdentify HRIF_SetFTCalibration HRIF_ReadForceData HRIF_SetInitializeForceSensor HRIF_GetLastCalibParams
V1.0.9.1	2025.03.06	-	M 修改 HRIF_CalTCP4P 接口参数
V1.0.10.0	2025.03.25	6.3.9	A 新增接口 HRIF_ShortJogJ HRIF_ShortJogL HRIF_LongJogJ HRIF_LongJogL HRIF_LongMoveEvent

## 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_SetSimulation .....	13
3.1.10. HRIF_IsSimulateRobot .....	14
3.1.11. HRIF_IsControllerStarted .....	15

---

3.1.12. HRIF_GetErrorCodeStr .....	16
3.1.13. HRIF_ReadVersion .....	17
3.1.14. HRIF_ReadRobotModel .....	19
<b>3.2. 轴组控制指令 .....</b>	<b>20</b>
3.2.1. HRIF_GrpEnable .....	20
3.2.2. HRIF_GrpDisable .....	21
3.2.3. HRIF_GrpReset .....	22
3.2.4. HRIF_GrpStop .....	23
3.2.5. HRIF_GrpInterrupt .....	24
3.2.6. HRIF_GrpContinue .....	25
3.2.7. HRIF_GrpCloseFreeDriver .....	26
3.2.8. HRIF_GrpOpenFreeDriver .....	27
3.2.9. HRIF_OpenBrake .....	28
3.2.10. HRIF_CloseBrake .....	29
3.2.11. HRIF_ReadBrakeStatus .....	30
3.2.12. HRIF_MoveToSS .....	31
3.2.13. HRIF_EnterSafeGuard .....	32
3.2.14. HRIF_ReadTriStageSwitch .....	33
3.2.15. HRIF_SetTriStageSwitch .....	34
3.2.16. HRIF_MoveAlignToZ .....	35
3.2.17. HRIF_LoadIdentify .....	37
3.2.18. HRIF_GetLoadIdentifyResult .....	38
<b>3.3. 脚本控制指令 .....</b>	<b>40</b>
3.3.1. HRIF_RunFunc .....	40
3.3.2. HRIF_StartScript .....	41
3.3.3. HRIF_StopScript .....	42
3.3.4. HRIF_PauseScript .....	43
3.3.5. HRIF_ContinueScript .....	44
3.3.6. HRIF_SwitchScript .....	45
3.3.7. HRIF_ReadDefaultScript .....	46
<b>3.4. 电箱控制指令 .....</b>	<b>47</b>
3.4.1. HRIF_ReadBoxInfo .....	47
3.4.2. HRIF_ReadBoxCI .....	49
3.4.3. HRIF_ReadBoxDI .....	50
3.4.4. HRIF_ReadBoxCO .....	51
3.4.5. HRIF_ReadBoxDO .....	52
3.4.6. HRIF_ReadBoxAI .....	53
3.4.7. HRIF_ReadBoxAO .....	54
3.4.8. HRIF_SetBoxCO .....	55
3.4.9. HRIF_SetBoxDO .....	56

---

3.4.10. HRIF_SetBoxAOMode .....	57
3.4.11. HRIF_SetBoxAOVal .....	58
3.4.12. HRIF_SetEndDO .....	59
3.4.13. HRIF_SetMoveParamsAO .....	60
3.4.14. HRIF_ReadEndDI .....	61
3.4.15. HRIF_ReadEndDO .....	62
3.4.16. HRIF_ReadEndAI .....	63
3.4.17. HRIF_ReadEndBTN .....	64
3.4.18. HRIF_EnableEndBTN .....	65
3.4.19. HRIF_cdsSetIO .....	66
<b>3.5. 状态读取与设置指令 .....</b>	<b>69</b>
3.5.1. HRIF_SetOverride .....	69
3.5.2. HRIF_SetToolMotion .....	70
3.5.3. HRIF_SetPayload .....	71
3.5.4. HRIF_SetJointMaxVel .....	72
3.5.5. HRIF_SetJointMaxVel_nJ .....	73
3.5.6. HRIF_SetJointMaxAcc .....	74
3.5.7. HRIF_SetJointMaxAcc_nJ .....	75
3.5.8. HRIF_SetLinearMaxVel .....	76
3.5.9. HRIF_SetLinearMaxAcc .....	77
3.5.10. HRIF_SetMaxAcsRange .....	78
3.5.11. HRIF_SetMaxAcsRange_nJ .....	80
3.5.12. HRIF_SetMaxPcsRange .....	81
3.5.13. HRIF_SetCollideLevel .....	82
3.5.14. HRIF_ReadMaxPayload .....	83
3.5.15. HRIF_ReadPayload .....	84
3.5.16. HRIF_ReadOverride .....	85
3.5.17. HRIF_ReadJointMaxVel .....	86
3.5.18. HRIF_ReadJointMaxVel_nJ .....	87
3.5.19. HRIF_ReadJointMaxAcc .....	88
3.5.20. HRIF_ReadJointMaxAcc_nJ .....	89
3.5.21. HRIF_ReadJointMaxJerk .....	90
3.5.22. HRIF_ReadJointMaxJerk_nJ .....	91
3.5.23. HRIF_ReadLinearMaxSpeed .....	92
3.5.24. HRIF_ReadEmergencyInfo .....	93
3.5.25. HRIF_ReadRobotState .....	94
3.5.26. HRIF_ReadCurWaypointID .....	96
3.5.27. HRIF_ReadAxisErrorCode .....	97
3.5.28. HRIF_ReadAxisErrorCode_nJ .....	98
3.5.29. HRIF_ReadCurFSM .....	99
3.5.30. HRIF_ReadPointByName .....	100

---

3.5.31. HRIF_ReadPointByName_nJ .....	102
3.5.32. HRIF_ReadCurFSMFromCPS .....	104
3.5.33. HRIF_ReadRobotFlags .....	105
3.5.34. HRIF_ReadPointList .....	107
3.6. 位置、速度、电流读取指令 .....	108
3.6.1. HRIF_ReadActPos .....	108
3.6.2. HRIF_ReadActPos_nJ .....	110
3.6.3. HRIF_ReadCmdJointPos .....	112
3.6.4. HRIF_ReadCmdJointPos_nJ .....	114
3.6.5. HRIF_ReadActJointPos .....	115
3.6.6. HRIF_ReadActJointPos_nJ .....	116
3.6.7. HRIF_ReadCmdTcpPos .....	117
3.6.8. HRIF_ReadActTcpPos .....	118
3.6.9. HRIF_ReadCmdJointVel .....	119
3.6.10. HRIF_ReadCmdJointVel_nJ .....	120
3.6.11. HRIF_ReadActJointVel .....	121
3.6.12. HRIF_ReadActJointVel_nJ .....	122
3.6.13. HRIF_ReadCmdTcpVel .....	123
3.6.14. HRIF_ReadActTcpVel .....	124
3.6.15. HRIF_ReadCmdJointCur .....	125
3.6.16. HRIF_ReadCmdJointCur_nJ .....	126
3.6.17. HRIF_ReadActJointCur .....	127
3.6.18. HRIF_ReadActJointCur_nJ .....	128
3.6.19. HRIF_ReadTcpVelocity .....	129
3.7. 坐标转换计算指令 .....	130
3.7.1. HRIFQuaternion2RPY .....	130
3.7.2. HRIF_RPY2Quaternion .....	132
3.7.3. HRIF_GetInverseKin .....	133
3.7.4. HRIF_GetInverseKin_nJ .....	136
3.7.5. HRIF_GetForwardKin .....	138
3.7.6. HRIF_GetForwardKin_nJ .....	140
3.7.7. HRIF_Base2UcsTcp .....	142
3.7.8. HRIF_UcsTcp2Base .....	144
3.7.9. HRIF_PoseAdd .....	146
3.7.10. HRIF_PoseSub .....	148
3.7.11. HRIF_PoseTrans .....	150
3.7.12. HRIF_PoseInverse .....	152
3.7.13. HRIF_PoseDist .....	154
3.7.14. HRIF_PoseInterpolate .....	156
3.7.15. HRIF_PoseDefdFrame .....	158

---

3.7.16. HRIF_CalUcsPlane .....	161
3.7.17. HRIF_CalUcsLine .....	163
3.7.18. HRIF_CalTcp3P .....	165
3.7.19. HRIF_CalTcp4P .....	167
3.8. 工具坐标与用户坐标读写指令 .....	170
3.8.1. HRIF_SetTCP .....	170
3.8.2. HRIF_SetUCS .....	171
3.8.3. HRIF_ReadCurTCP .....	172
3.8.4. HRIF_ReadCurUCS .....	173
3.8.5. HRIF_SetTCPByName .....	174
3.8.6. HRIF_SetUCSByName .....	175
3.8.7. HRIF_ReadTCPByName .....	176
3.8.8. HRIF_ReadUCSByName .....	178
3.8.9. HRIF_ConfigTCP .....	180
3.8.10. HRIF_ConfigUCS .....	181
3.8.11. HRIF_ReadTCPList .....	182
3.8.12. HRIF_ReadUCSList .....	183
3.8.13. HRIF_SetBaseInstallingAngle .....	184
3.8.14. HRIF_GetBaseInstallingAngle .....	185
3.9. 力控控制指令 .....	186
3.9.1. HRIF_SetForceControlState .....	186
3.9.2. HRIF_ReadForceControlState .....	187
3.9.3. HRIF_SetForceToolCoordinateMotion .....	188
3.9.4. HRIF_ForceControlInterrupt .....	189
3.9.5. HRIF_ForceControlContinue .....	190
3.9.6. HRIF_SetForceZero .....	191
3.9.7. HRIF_SetMaxSearchVelocities .....	192
3.9.8. HRIF_SetControlFreedom .....	193
3.9.9. HRIF_SetForceControlStrategy .....	194
3.9.10. HRIF_SetFreeDrivePositionAndOrientation .....	195
3.9.11. HRIF_SetPIDControlParams .....	196
3.9.12. HRIF_SetMassParams .....	197
3.9.13. HRIF_SetDampParams .....	198
3.9.14. HRIF_SetStiffParams .....	199
3.9.15. HRIF_SetForceControlGoal .....	200
3.9.16. HRIF_SetControlGoal .....	201
3.9.17. HRIF_SetForceDataLimit .....	203
3.9.18. HRIF_SetForceDistanceLimit .....	205
3.9.19. HRIF_SetForceFreeDriveMode .....	206
3.9.20. HRIF_SetFTFreeDriveSpeedMode .....	207

---

3.9.21. HRIF_ReadFTFreeDriveSpeedMode .....	208
3.9.22. HRIF_ReadFTCabData .....	209
3.9.23. HRIF_SetFreeDriveMotionFreedom .....	210
3.9.24. HRIF_SetFTFreeFactor .....	211
3.9.25. HRIF_SetTangentForceBounds .....	212
3.9.26. HRIF_SetFreeDriveCompensateForce .....	213
3.9.27. HRIF_SetFTWrenchThresholds .....	214
3.9.28. HRIF_SetMaxFreeDriveVel .....	215
3.9.29. HRIF_ReadFTMotionFreedom .....	216
3.9.30. HRIF_SetMaxSearchDistance .....	217
3.9.31. HRIF_SetSteadyContactDeviationRange .....	218
3.9.32. HRIF_SetDepthThresholdForDampingArea .....	220
3.9.33. HRIF_AddSafePlane .....	221
3.9.34. HRIF_UpdateSafePlane .....	223
3.9.35. HRIF_DelSafePlane .....	225
3.9.36. HRIF_ReadSafePlaneList .....	226
3.9.37. HRIF_ReadSafePlane .....	227
3.9.38. HRIF_SetFTCalibration .....	228
3.9.39. HRIF_ReadForceData .....	229
3.9.40. HRIF_SetLastCalibParams .....	230
3.9.41. HRIF_GetLastCalibParams .....	231
3.10. 通用运动类控制指令 .....	232
3.10.1. HRIF_MoveRelJ .....	232
3.10.2. HRIF_MoveRelL .....	233
3.10.3. HRIF_WayPointRel .....	234
3.10.4. HRIF_WayPointRel_nJ .....	238
3.10.5. HRIF_IsMotionDone .....	242
3.10.6. HRIF_IsBlendingDone .....	243
3.10.7. HRIF_WayPointEx .....	244
3.10.8. HRIF_WayPointEx_nJ .....	248
3.10.9. HRIF_WayPoint .....	252
3.10.10. HRIF_WayPoint_nJ .....	256
3.10.11. HRIF_WayPoint2 .....	259
3.10.12. HRIF_WayPoint2_nJ .....	263
3.10.13. HRIF_MoveJ .....	267
3.10.14. HRIF_MoveJ_nJ .....	270
3.10.15. HRIF_MoveL .....	273
3.10.16. HRIF_MoveL_nJ .....	276
3.10.17. HRIF_MoveC .....	279
3.10.18. HRIF_MoveC_nJ .....	282
3.10.19. HRIF_MoveZ .....	286

---

3.10.20. HRIF_MoveE .....	290
3.10.21. HRIF_MoveS .....	294
3.10.22. HRIF_MoveLinearWeave .....	296
3.10.23. HRIF_MoveCircularWeave .....	299
3.11. 连续轨迹运动类控制指令 .....	308
3.11.1. HRIF_StartPushMovePathJ .....	308
3.11.2. HRIF_PushMovePathJ .....	309
3.11.3. HRIF_EndPushMovePathJ .....	310
3.11.4. HRIF_MovePathJ .....	311
3.11.5. HRIF_ReadMovePathJState .....	312
3.11.6. HRIF_UpdateMovePathJName .....	314
3.11.7. HRIF_DelMovePathJ .....	315
3.11.8. HRIF_ReadTrackProcess .....	316
3.11.9. HRIF_InitMovePathL .....	317
3.11.10. HRIF_PushMovePathL .....	319
3.11.11. HRIF_PushMovePaths .....	320
3.11.12. HRIF_MovePathL .....	322
3.11.13. HRIF_MovePathJOL .....	323
3.11.14. HRIF_GetMovePathJOLIndex .....	326
3.11.15. HRIF_SetMovePathOverride .....	327
3.11.16. HRIF_InitPath .....	328
3.11.17. HRIF_PushPathPoints .....	331
3.11.18. HRIF_EndPushPathPoints .....	333
3.11.19. HRIF_DelPath .....	334
3.11.20. HRIF_ReadPathList .....	335
3.11.21. HRIF_ReadPathInfo .....	336
3.11.22. HRIF_UpdatePathName .....	338
3.11.23. HRIF_ReadPathState .....	339
3.12. Servo 运动类控制指令 .....	341
3.12.1. HRIF_StartServo .....	341
3.12.2. HRIF_PushServoJ .....	342
3.12.3. HRIF_PushServoP .....	343
3.12.4. HRIF_SpeedJ .....	345
3.12.5. HRIF_SpeedL .....	347
3.13. 相对跟踪运动类控制指令 .....	349
3.13.1. HRIF_SetMoveTraceParams .....	349
3.13.2. HRIF_SetMoveTraceInitParams .....	350
3.13.3. HRIF_SetMoveTraceUcs .....	351
3.13.4. HRIF_SetTrackingState .....	352

---

3.14. 位置跟随运动类控制指令 .....	353
3.14.1. HRIF_SetPoseTrackingMaxMotionLimit .....	353
3.14.2. HRIF_SetPoseTrackingStopTimeOut .....	354
3.14.3. HRIF_SetPoseTrackingPIDParams .....	355
3.14.4. HRIF_SetPoseTrackingTargetPos .....	356
3.14.5. HRIF_SetPoseTrackingState .....	357
3.14.6. HRIF_SetUpdateTrackingPose .....	358
3.15. 其他指令 .....	359
3.15.1. HRIF_HRApp .....	359
3.15.2. HRIF_HRAppCmd .....	360
3.15.3. HRIF_WriteEndHoldingRegisters .....	361
3.15.4. HRIF_ReadEndHoldingRegisters .....	363
<b>第四章 附录 .....</b>	<b>365</b>
4.1. 参考文件 .....	365

## 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

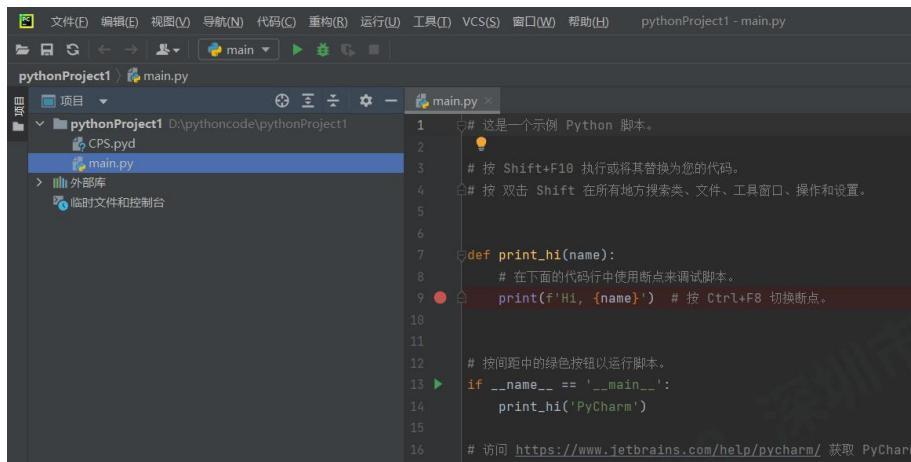
数据类型	长度	说明
bool	-	布尔型
int	-	整型
float	-	浮点型
string	-	字符串
list		数组

## 第二章 环境搭建

### 2.1. Windows 环境搭建

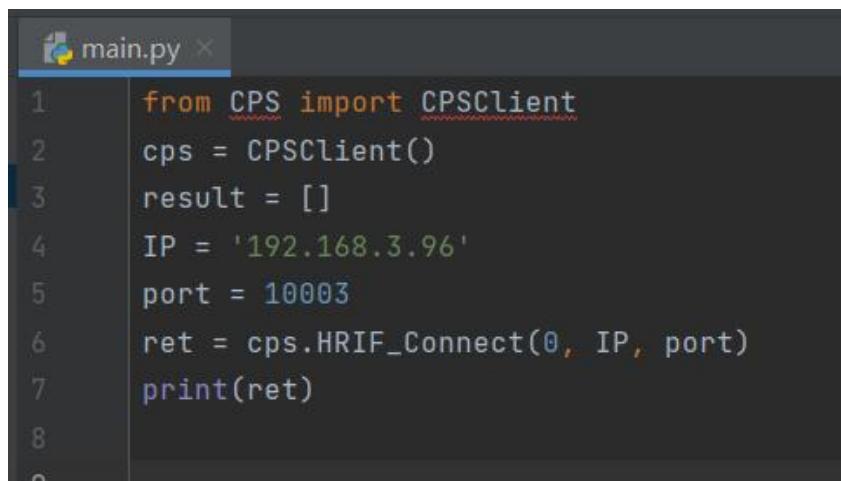
#### 2.1.1. 创建项目

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 CPSClient 对象，尝试连接。

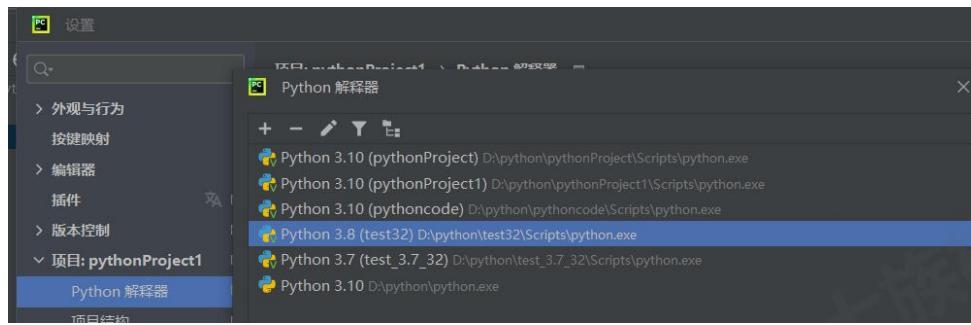


```
from CPS import CPSClient
cps = CPSClient()
result = []
IP = '192.168.3.96'
port = 10003
ret = cps.HRIF_Connect(0, IP, port)
print(ret)
```

#### 2.1.3. 版本选择

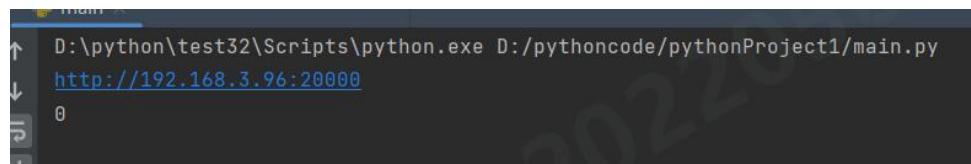
根据所用包体，选择对应版本的编译器，如原包体名为 CPS\_3.8.9\_win\_32.pyd 即表示应选择 32 位 Python 3.8

版本。



## 2.1.4. 运行程序

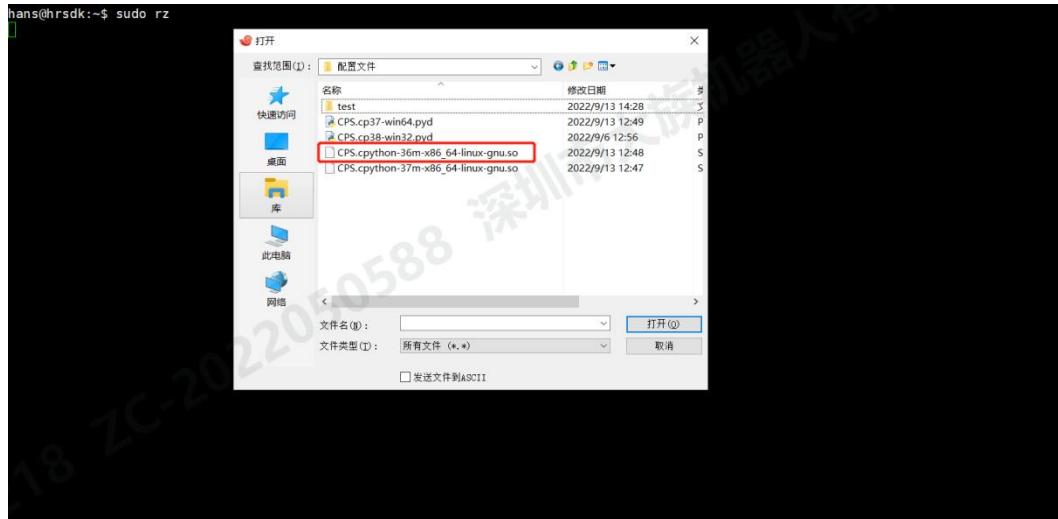
右键单击工作区，选择 run code，查看下方终端打印



## 2.2. Linux 环境搭建

### 2.2.1. 上传 SDK

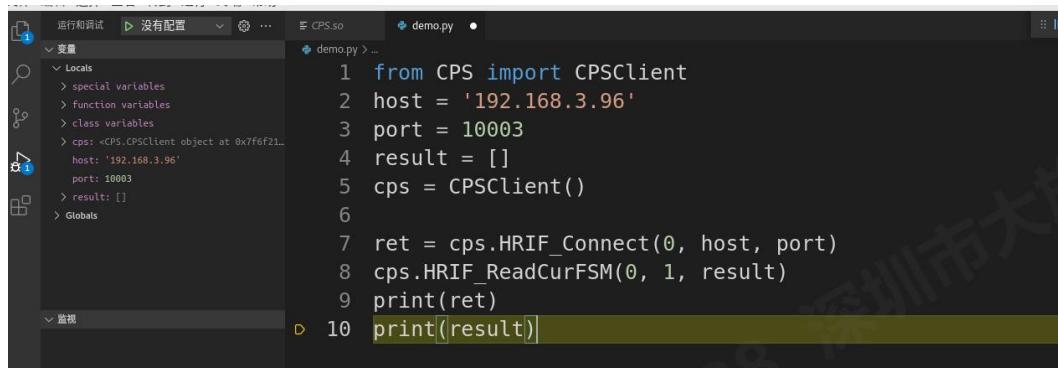
上传 SDK 文件至系统中（以 vscode 为例）



### 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 CPSClient 对象，尝试连接。



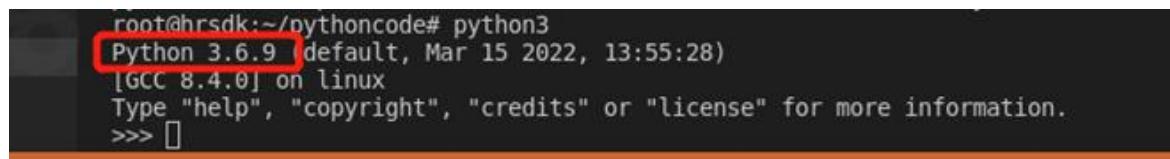
```

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

```

### 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。



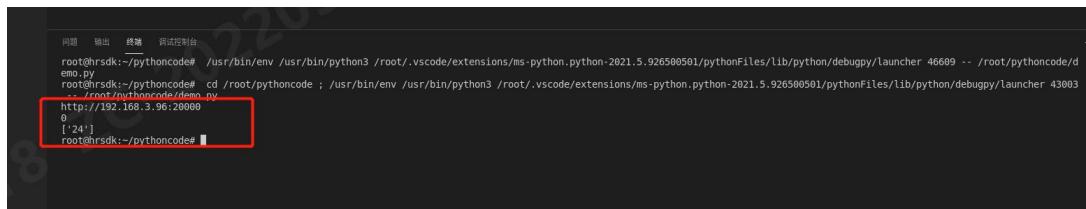
```

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-2021.5.926500501/pythonFiles/lib/python/debugpy/launcher 46609 -- /root/pythoncode/demo.py
root@hrsdk:~/pythoncode# cd /root/pythoncode ; /usr/bin/env /usr/bin/python3 /root/.vscode/extensions/ms-python.python-2021.5.926500501/pythonFiles/lib/python/debugpy/launcher 43003
http://192.168.3.96:20000
[24]
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\_SetSimulation

3.1.9.1. 描述：设置机器人的模拟状态。

✓ 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
nSimulation	是否为模拟 机器人	int	0/1	0: 真实机器人 1: 模拟机器人

注：该指令需要在机器人断电模式下才能正常调用，否则返回 20018 错误。

✓ 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

# 设置当前机器人为模拟模式

```
nRet = cps.HRIF_SetSimulation(0, 1)
```

### 3.1.10. HRIF\_IsSimulateRobot

3.1.10.1. 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

3.1.10.2. 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)
```

### 3.1.11. HRIF\_IsControllerStarted

3.1.11.1. 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

3.1.11.2. 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)
```

### 3.1.12. HRIF\_GetErrorCodeStr

3.1.12.1. 描述：获取错误码解释。

✓ 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
nErrorCode	错误码	int	-	错误码

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
strErrorMsg	错误码描述	StringData	-	错误码的具体描述说明

✓ 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

```
# 错误码
nErrorCode = 20018

# 定义返回值列表
result = []

# 调用接口
nRet = cps.HRIF_GetErrorCodeStr(0,nErrorCode,result)

# 获取错误码描述
```



### 3.1.13. HRIF\_ReadVersion

3.1.13.1. 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
result[7]	软件版本	string	-	软件版本

✓ 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

3.1.13.2. 示例

# 定义返回值空列表

result = [ ]



---

# 获取版本信息

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

### 3.1.14. HRIF\_ReadRobotModel

3.1.14.1. 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

3.1.14.2. 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)
```

### 3.2.9. HRIF\_OpenBrake

3.2.9.1. 描述：松闸。

✓ 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
nAxisID	轴 ID	int	0-5	需要松闸的目标轴 ID, 对应关节 J1-J6

✓ 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

3.2.9.2. 示例

# 松闸 (关节 6)

```
nRet = cps.HRIF_OpenBrake(0,5)
```

### 3.2.10. HRIF\_CloseBrake

3.2.10.1. 描述：抱闸。

✓ 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
nAxisID	轴 ID	int	0-5	需要抱闸的目标轴 ID, 对应关节 J1-J6

✓ 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

3.2.10.2. 示例

# 抱闸 (关节 6)

```
nRet = cps.HRIF_CloseBrake(0,5)
```

### 3.2.11. HRIF\_ReadBrakeStatus

3.2.11.1. 描述：读取各关节松/抱闸状态。

✓ 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 = [ ]

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
result[1~6]	松/抱闸状态	string	0/1	0: 松闸 1: 抱闸

✓ 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

3.2.11.2. 示例

```
result = []
# 读取各关节松/抱闸状态
nRet = cps.HRIF_ReadBrakeStatus(0,result)
```

### 3.2.12. HRIF\_MoveToSS

3.2.12.1. 描述：移动到安全位置。

✓ 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

✓ 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

3.2.12.2. 示例

# 移动到安全位置

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

### 3.2.13. HRIF\_EnterSafeGuard

3.2.13.1. 描述：强制进入安全光幕（软急停）。

✓ 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
bFlag	状态标识	int	0/1	0: 退出软急停 1: 进入软急停

\* 注：该指令用强制进入/退出安全光幕的方式进入/退出软急停。进入软急停后，示教器或者前端界面上看到的现象是进入了安全光幕。用本指令退出软急停后，需要在示教器或者前端界面点击“光幕恢复”。

✓ 返回值

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

### 3.2.13.2. 示例

# 指定状态为进入软急停

nFlag=1

# 机器人进入软急停

nRet=cps.HRIF\_EnterSafeGuard(0,0,nFlag)

### 3.2.14. HRIF\_ReadTriStageSwitch

3.2.14.1. 描述：读取三段式按钮的开关以及模式。

✓ 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

✓ 输出变量

参数	名称	数据类型	有效范围	内容
result[0]	三段式开关	string	0/1	0: 关闭三段式按钮 1: 开启三段式
result[1]	三段式模式	string	0/1	0: 零力示教 1: 使能

✓ 返回值

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

3.2.14.2. 示例

# 三段式按钮开关与模式

```
result = []
# 读取三段式按钮开关和模式
nRet=cps.HRIF_ReadTriStageSwitch(0,0,result)
```

### 3.2.15. HRIF\_SetTriStageSwitch

3.2.15.1. 描述：设置三段式按钮的开关以及模式。

✓ 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
nThreeStageEnable	三段式开关	int	0/1	0: 关闭三段式按钮 1: 开启三段式
nThreeStageMode	三段式模式	int	0/1	0: 零力示教 1: 使能

✓ 返回值

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

3.2.15.2. 示例

# 打开三段式按钮开关

nThreeStageEnable=1

# 设置三段式按钮模式为零力示教

nThreeStageMode=0

# 读取三段式按钮开关和模式

nRet=cps.HRIF\_SetTriStageSwitch(0,0,nThreeStageEnable,nThreeStageMode)

### 3.2.16. HRIF\_MoveAlignToZ

3.2.16.1. 描述：z 轴对齐。

✓ 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
sUcsName	用户坐标名称	string	-	需要设置的用户坐标名称
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]	到位标志	string	0/1	1: z 轴已经对齐 0: z 轴尚未对齐
result[1-6]	关节坐标	string	-	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[°]

✓ 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

3.2.16.2. 示例

# 定义工具坐标变量

sTcpName = "TCP"

# 定义用户坐标变量

sUcsName = "Base"

---

# 定义返回值空列表

result = [ ]

IsReached = 0

# 循环调用

while not IsReached:

nRet= HRIF\_MoveAlignToZ(0,0,sTcpName ,sUcsName ,result)

IsReached = int(result[0])

### 3.2.17. HRIF\_LoadIdentify

3.2.17.1. 描述：开启负载辨识。

✓ 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

3.2.17.2. 示例

# 开启负载辨识

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



### 3.2.18. HRIF\_GetLoadIdentifyResult

3.2.18.1. 描述：获取负载辨识结果。

✓ 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]	负载辨识状态	string	0~7	0, 成功 1, 第一条轨迹错误 2, 第二条轨迹错误 3, 第三条轨迹错误 4, 负载辨识运行中 5, 意外停止 6, 意外退出 7, 初始化, 负载辨识未开始
result[1]	进度状态	string	0~100	当前负载识别进度
result[2]	负载质量	string	-	识别出的负载质量 (单位 kg)
result[3]	质心 X 坐标	string	-	负载质心 X 坐标
result[4]	质心 Y 坐标	string	-	负载质心 Y 坐标
result[5]	质心 Z 坐标	string	-	负载质心 Z 坐标

✓ 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

3.2.18.2. 示例

# 初始化



---

```
result = [ ]  
# 循环调用获取结果  
nRet= cps.HRIF_GetLoadIdentifyResult(0, 0, result)
```

### 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 = [ ]

✓ 输出变量

输入变量	名称	数据类型	有效范围	内容
result	返回值	list		输出值

✓ 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

3.3.1.2. 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, 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

✓ 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)
```

### 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)
```

### 3.3.6. HRIF\_SwitchScript

3.3.6.1. 描述：应用指定的脚本程序。

✓ 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
sScriptName	脚本程序名称	string		以.json 为扩展名

✓ 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

3.3.6.2. 示例

```
# 脚本名称
sScriptName = "abcd.json"

# 应用指定脚本
nRet = cps.HRIF_SwitchScript(0, 0, sScriptName)
```

### 3.3.7. HRIF\_ReadDefaultScript

3.3.7.1. 描述：读取系统当前应用的脚本程序的名字。

✓ 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]	脚本程序名	string	-	当前正在应用的脚本程序名字, 以.json 为扩展名

✓ 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

3.3.7.2. 示例

```
result = []
# 读取脚本名称
nRet = cps.HRIF_ReadDefaultScript(0, 0, result)
sScriptName = result[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,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	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

3.4.4.2. 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	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

3.4.5.2. 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	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

3.4.6.2. 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
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

3.4.10.2. 示例

```
# 定义设置值
nMode = 1

# 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 , nMode)
```

### 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 = 1

# 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\_SetMoveParamsAO

3.4.13.1. 描述：设置焊接模拟量输出电压。

✓ 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
nState	通道开关状态	int	0/1	0: 关闭 1: 开启
nIndex	电压模拟量通道	int	0/1	0: 通道 AO0 1: 通道 AO1
dInitAO	初始化电压	float	0-10V	初始化电压, 单位: V
dWeldingAO	焊接电压	float	0-10V	焊接电压, 单位: V

✓ 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

# 定义通道开关状态

nState = 0

# 定义电压模拟量通道

nIndex = 0

# 定义初始化电压

dInitAO = 0

# 定义实际焊接电压

dWeldingAO = 0

# 设置焊接模拟量输出电压

nRet = cps.HRIF\_SetMoveParamsAO(0, nState,nIndex,dInitAO,dWeldingAO)

### 3.4.14. HRIF\_ReadEndDI

3.4.14.1. 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

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

### 3.4.15. HRIF\_ReadEndDO

3.4.15.1. 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

3.4.15.2. 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 位 DO 状态
nRet = cps.HRIF_ReadEndDO(0,0, nBit, result)
```

### 3.4.16. HRIF\_ReadEndAI

3.4.16.1. 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

3.4.16.2. 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)
```

### 3.4.17. HRIF\_ReadEndBTN

3.4.17.1. 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

3.4.17.2. 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)
```

### 3.4.18. HRIF\_EnableEndBTN

3.4.18.1. 描述：启用/禁用末端按键。

✓ 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
state	模块按钮状态	int	0/1	0: 禁用 1: 启用

✓ 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

3.4.18.2. 示例

# 定义需要设置的状态

state = 0

# 启用末端按钮

nRet = cps.HRIF\_EnableEndBTN(0,0,state)



### 3.4.19. HRIF\_cdsSetIO

3.4.19.1. 描述：在运动指令到达目标点位前设置 IO

✓ 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
nEndDOMask	需要更改的 EndDO 按 bit 标识	int	-	EndDO0~EndDO7 对应到 nEndDOMask 的 bit0~bit7, 如果某个 bit 是 1, 则其对应的 EndDO 的值需要更改, 否则无需更改。
nEndDOVal	各个需要更改的 EndDO 的目标状态	int	-	EndDO0~EndDO7 对应到 nEndDOVal 的 bit0~bit7。 0: 低电平 1: 高电平
nBoxDOMask	需要更改的 BoxDO 按 bit 标识	int	-	BoxDO0~BoxDO7 对应到 nBoxDOMask 的 bit0~bit7, 如果某个 bit 是 1, 则其对应的 BoxDO 的值需要更改, 否则无需更改。
nBoxDOVal	各个需要更改的 BoxDO 的目标状态	int	-	BoxDO0~BoxDO7 对应到 nBoxDOVal 的 bit0~bit7。 0: 低电平 1: 高电平
nBoxCOMask	需要更改的 BoxCO 按 bit 标识	int	-	BoxCO~BoxCO 对应到 nBoxCOMask 的 bit0~bit7, 如果某个 bit 是 1, 则其对应的 BoxCO 的值需要更改, 否则无需更改。
nBoxCOVal	各个需要更改的 BoxCO 的目标状态	int	-	BoxCO~BoxCO 对应到 nBoxCOVal 的 bit0~bit7。 0: 低电平 1: 高电平
nBoxAOCH0_Mask	BoxAOCH0 是否需要更改的标识	int	0/1	0: 无需更改 1: 需要更改
nBoxAOCH0_Mode	模式	int	1/2	1: 电压模式 2: 电流模式
nBoxAOCH1_Mask	BoxAOCH1 是否需要更改的标识	int	0/1	0: 无需更改 1: 需要更改

nBoxAOCH1_Mode	模式	int	1/2	1: 电压模式 2: 电流模式
dbBoxAOCH0_Val	对应模拟量值	float	0~20	电流模式: 4-20mA 电压模式: 0-10V
dbBoxAOCH1_Val	对应模拟量值	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	错误码	int	>0 的整型值	返回调用失败的错误码

#### 3.4.19.2. 示例

```
# 需要更改的 EndDO 按 bit 标识
nEndDOMask=1

# 各个需要更改的 EndDO 的目标状态
nEndDOVal=0

# 需要更改的 BoxDO 按 bit 标识
nBoxDOMask=0

# 各个需要更改的 BoxDO 的目标状态
nBoxDOVal=0

# 需要更改的 BoxCO 按 bit 标识
nBoxCOMask=0

# 各个需要更改的 BoxCO 的目标状态
nBoxCOVal=0

# BoxAOCH0 是否需要更改的标识
nBoxAOCH0_Mask=0

# 模式
nBoxAOCH0_Mode=0

# BoxAOCH1 是否需要更改的标识
nBoxAOCH1_Mask=0

# 模式
nBoxAOCH1_Mode=0
```

---

# 对应模拟量值

dbBoxAOCH0\_Val=0

# 对应模拟量值

dbBoxAOCH1\_Val=0

# 在运动指令到达目标点位前设置 IO

nRet=cps.HRIF\_cdsSetIO(0,0,nEndDOMask, nEndDOVal, nBoxDOMask, nBoxDOVal, nBoxCOMask, nBoxCOVal,  
nBoxAOCH0\_Mask, nBoxAOCH0\_Mode, nBoxAOCH1\_Mask, nBoxAOCH1\_Mode, dbBoxAOCH0\_Val,  
dbBoxAOCH1\_Val)

### 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: 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

3.5.2.2. 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\_SetJointMaxVel\_nJ

3.5.5.1. 描述：设置关节最大运动速度。**注意：这是扩展轴版本所用接口**

✓ 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-dJn	轴最大速度	float	0~200	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

3.5.5.2. 示例

# 定义需要设置关节最大速度

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

# 设置最大关节速度

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

### 3.5.6. HRIF\_SetJointMaxAcc

3.5.6.1. 描述：设置关节最大运动加速度，加速度需比速度大。

✓ 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

#### 3.5.6.2. 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)

### 3.5.7. HRIF\_SetJointMaxAcc\_nJ

3.5.7.1. 描述：设置关节最大运动加速度，加速度需比速度大。**注意：这是扩展轴版本所用接口**

✓ 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-dJn	最大加速度	float	0~720	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

3.5.7.2. 示例

# 定义需要设置的关节最大加速度

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

# 设置最大关节加速度

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

### 3.5.8. HRIF\_SetLinearMaxVel

3.5.8.1. 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

3.5.8.2. 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)
```

### 3.5.9. HRIF\_SetLinearMaxAcc

3.5.9.1. 描述：设置直线运动最大加速度，加速度需比速度大。

✓ 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

3.5.9.2. 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)
```

### 3.5.10. HRIF\_SetMaxAcsRange

3.5.10.1. 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

3.5.10.2. 示例

# 定义需要设置的关节最大运动范围

dMax = [360, 135, 153, 360, 180, 360]

# 定义需要设置的关节最小运动范围

dMin = [-360, -135, -153, -360, -180, -360]

# 设置关节运动范围



---

```
nRet = cps.HRIF_SetMaxAcsRange (0,0,dMax, dMin)
```

### 3.5.11. HRIF\_SetMaxAcsRange\_nJ

3.5.11.1. 描述：设置关节最大运动范围。**注意：这是扩展轴版本所用接口**

✓ 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- dMaxJn	关节最大范围	float	-	dMaxJn:关节 n 最大运动范围, 单位[°]
dMinJ1- dMinJn	关节最小范围	float	-	dMinJn:关节 n 最小运动范围, 单位[°]

✓ 返回

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

```
# 定义需要设置的关节最大运动范围
dMax = [360, 135, 153, 360, 180, 360, 360, 360]

# 定义需要设置的关节最小运动范围
dMin = [-360, -135, -153, -360, -180, -360, -360, -360]

# 设置关节运动范围
nRet = cps.HRIF_SetMaxAcsRange_nJ (0,0,dMax, dMin)
```

### 3.5.12. HRIF\_SetMaxPcsRange

3.5.12.1. 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

3.5.12.2. 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)

### 3.5.13. HRIF\_SetCollideLevel

3.5.13.1. 描述：设置安全风险等级。

✓ 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
nSafeLevel	安全风险等级	int	-	安全风险等级 (0-5) 0: 安全风险低 5: 安全风险高

✓ 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

# 定义安全风险等级

nSafeLevel = 3

# 设置安全风险等级

nRet= cps.HRIF\_SetCollideLevel(0, nSafeLevel)

### 3.5.14. HRIF\_ReadMaxPayload

3.5.14.1. 描述：读取末端最大负载。

✓ 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]	最大负载	string	-	末端最大负载, 单位: kg

✓ 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 = [ ]

# 读取末端最大负载

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

### 3.5.15. HRIF\_ReadPayload

3.5.15.1. 描述：读取当前负载参数。

✓ 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
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]	质量	string	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]
result[1]	质心 X 方向偏移	string	>0	质心 X 方向偏移, 单位[mm]
result[2]	质心 Y 方向偏移	string	>0	质心 Y 方向偏移, 单位[mm]
result[3]	质心 Z 方向偏移	string	>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

3.5.15.2. 示例

```
# 定义返回值空列表
```

```
result = [ ]
```

```
# 读取当前负载参数
```

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

### 3.5.16. HRIF\_ReadOverride

3.5.16.1. 描述：读取速度比。

✓ 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
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]	速度比	string	0.01~1	当前系统的速度比(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

3.5.16.2. 示例

# 定义返回值空列表

result = [ ]

# 读取速度比

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

### 3.5.17. HRIF\_ReadJointMaxVel

3.5.17.1. 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

3.5.17.2. 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)
```

### 3.5.18. HRIF\_ReadJointMaxVel\_nJ

3.5.18.1. 描述：读取关节最大运动速度。**注意：这是扩展轴版本所用接口**

✓ 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~n-1]	Jn 轴最大速度	string	0~200	Jn 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

3.5.18.2. 示例

```
# 定义返回值空列表
```

```
result = []
# 读取最大关节速度
nRet = cps.HRIF_ReadJointMaxVel_nJ(0,0, result)
```

### 3.5.19. HRIF\_ReadJointMaxAcc

3.5.19.1. 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

3.5.19.2. 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)
```

### 3.5.20. HRIF\_ReadJointMaxAcc\_nJ

3.5.20.1. 描述：读取关节最大运动加速度。**注意：这是扩展轴版本所用接口**

✓ 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~n-1]	Jn 轴最大加速度	string	0~720	Jn 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

3.5.20.2. 示例

```
# 定义返回值空列表
result = []

# 读取最大关节加速度
nRet = cps.HRIF_ReadJointMaxAcc_nJ(0,0, result)
```

### 3.5.21. HRIF\_ReadJointMaxJerk

3.5.21.1. 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[°/s <sup>3</sup> ]
result[1]	J2 轴最大加加速度	string	0~720	J2 轴最大加加速度, 单位[°/s <sup>3</sup> ]
result[2]	J3 轴最大加加速度	string	0~720	J3 轴最大加加速度, 单位[°/s <sup>3</sup> ]
result[3]	J4 轴最大加加速度	string	0~720	J4 轴最大加加速度, 单位[°/s <sup>3</sup> ]
result[4]	J5 轴最大加加速度	string	0~720	J5 轴最大加加速度, 单位[°/s <sup>3</sup> ]
result[5]	J6 轴最大加加速度	string	0~720	J6 轴最大加加速度, 单位[°/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

3.5.21.2. 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)
```

### 3.5.22. HRIF\_ReadJointMaxJerk\_nJ

3.5.22.1. 描述：读取关节最大运动加加速度。**注意：这是扩展轴版本所用接口**

✓ 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~n-1]	Jn 轴最大加加速度	string	0~720	Jn 轴最大加加速度, 单位[°/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

3.5.22.2. 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_nJ(0,0, result)
```

### 3.5.23. HRIF\_ReadLinearMaxSpeed

3.5.23.1. 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

3.5.23.2. 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)
```

### 3.5.24. HRIF\_ReadEmergencyInfo

3.5.24.1. 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, 当两路信号不相同时, 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, 当两路信号不相同时, 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

3.5.24.2. 示例

```
# 定义返回值空列表
```

```
result = []
```

```
# 读取急停信息
```

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

### 3.5.25. HRIF\_ReadRobotState

3.5.25.1. 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

### 3.5.25.2. 示例

# 定义返回值空列表

```
result = []
# 读取状态
nRet = cps.HRIF_ReadRobotState(0,0, result)
```

### 3.5.26. HRIF\_ReadCurWaypointID

3.5.26.1. 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

3.5.26.2. 示例

# 定义返回值空列表

result = [ ]

# 读取路点 ID

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

### 3.5.27. HRIF\_ReadAxisErrorCode

3.5.27.1. 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	>=0	当前错误码, 没有错误码时=0
result[1]	J1 轴错误码	string	>=0	J1 轴错误码, 没有错误码时=0
result[2]	J2 轴错误码	string	>=0	J2 轴错误码, 没有错误码时=0
result[3]	J3 轴错误码	string	>=0	J3 轴错误码, 没有错误码时=0
result[4]	J4 轴错误码	string	>=0	J4 轴错误码, 没有错误码时=0
result[5]	J5 轴错误码	string	>=0	J5 轴错误码, 没有错误码时=0
result[6]	J6 轴错误码	string	>=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

3.5.27.2. 示例

```
# 定义返回值空列表
```

```
result = [ ]
```

```
# 读取错误码
```

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

### 3.5.28. HRIF\_ReadAxisErrorCode\_nJ

3.5.28.1. 描述：读取错误码。**注意：这是扩展轴版本所用接口**

✓ 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	>=0	当前错误码, 没有错误码时=0
result[n]	Jn 轴错误码	string	>=0	Jn 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

3.5.28.2. 示例

```
# 定义返回值空列表
result = []

# 读取错误码
nRet = cps.HRIF_ReadAxisErrorCode_nJ(0,0, result)
```

### 3.5.29. HRIF\_ReadCurFSM

3.5.29.1. 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	>=0	状态机, 可以参考状态机列表
result[1]	状态描述	string	-	状态机描述

✓ 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

3.5.29.2. 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)

### 3.5.30. HRIF\_ReadPointByName

3.5.30.1. 描述：根据点位名称读取点位信息。

✓ 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
pointName	点位名称	string	-	点位名称
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	string	-	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[6]-result[11]	迪卡尔坐标	string	-	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[12]-result[17]	当前工具坐标	string	-	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	string	-	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

### 3.5.30.2. 示例

```

# 定义返回值空列表
result = []

# 读取当前实际位置信息
nRet = cps.HRIIF_ReadPointByName(0,0,'Point_1', 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])

```

### 3.5.31. HRIF\_ReadPointByName\_nJ

3.5.31.1. 描述：根据点位名称读取点位信息。**注意：这是扩展轴版本所用接口**

✓ 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
pointName	点位名称	string	-	点位名称
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[n-1]	关节坐标	string	-	dJn: 关节 n 坐标, 单位[°]
result[n]-result[n+5]	迪卡尔坐标	string	-	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[n+6]-result[n+11]	当前工具坐标	string	-	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[n+12]-result[n+17]	当前用户坐标	string	-	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

### 3.5.31.2. 示例

```
# 定义返回值空列表
result = []
# 读取当前实际位置信息
nRet = cps.HRIF_ReadPointByName_nJ(0,0,'Point_1', result)
```

### 3.5.32. HRIF\_ReadCurFSMFromCPS

3.5.32.1. 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	>=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

3.5.32.2. 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)
```

### 3.5.33. HRIF\_ReadRobotFlags

3.5.33.1. 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

## 3.5.33.2. 示例

```
# 定义返回值空列表
result = [ ]
# 读取状态
nRet = cps.HRIF_ReadRobotFlags(0,0, result)
```

### 3.5.34. HRIF\_ReadPointList

3.5.34.1. 描述：读取系统中保存的点位名称列表。

✓ 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	点位名称列表	list	-	保存读取到的所有点位名称

✓ 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

3.5.34.2. 示例

# 定义保存点位名称列表的变量

result = []

# 读取点位名称列表

nRet = HRIF\_ReadPointList(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]	关节坐标	string	-	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[6]-result[11]	迪卡尔坐标	string	-	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[12]-result[17]	当前工具坐标	string	-	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	string	-	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\_ReadActPos\_nJ

3.6.2.1. 描述：读取当前实际位置信息。**注意：这是扩展轴版本所用接口**

✓ 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[n-1]	关节坐标	string	-	dJn: 关节 n 坐标, 单位[°]
result[n]-result[n+5]	迪卡尔坐标	string	-	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[n+6]-result[n+11]	当前工具坐标	string	-	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[n+12]-result[n+17]	当前用户坐标	string	-	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

## 3.6.2.2. 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_nJ(0,0, result)
```



### 3.6.3. HRIF\_ReadCmdJointPos

3.6.3.1. 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	string	-	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[°]
result[6]-result[11]	空间坐标	string	-	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

3.6.3.2. 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])
```

# 读取空间命令位置变量

```
dX = float(result[6]) dY = float(result[7]) dZ = float(result[8])
```

```
dRX = float(result[9]) dRY = float(result[10]) dRZ = float(result[11])
```

### 3.6.4. HRIF\_ReadCmdJointPos\_nJ

3.6.4.1. 描述：读取关节命令位置。**注意：这是扩展轴版本所用接口**

✓ 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[n-1]	关节坐标	string	-	dJn: 关节 n 命令位置, 单位[°]
result[n]-result[n+5]	空间坐标	string	-	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 = []
```

```
# 读取关节命令位置
```

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

### 3.6.5. HRIF\_ReadActJointPos

3.6.5.1. 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	string	-	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

3.6.5.2. 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])
```

### 3.6.6. HRIF\_ReadActJointPos\_nJ

3.6.6.1. 描述：读取关节实际位置。**注意：这是扩展轴版本所用接口**

✓ 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[n-1]	关节坐标	string	-	dJn: 关节 n 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

3.6.6.2. 示例

```
# 定义返回值空列表
result = []

# 读取关节实际位置
nRet = cps.HRIF_ReadActJointPos_nJ(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])
dJ7 = float(result[6]) dJ8 = float(result[7]) dJ9 = float(result[8])
```

### 3.6.7. HRIF\_ReadCmdTcpPos

3.6.7.1. 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	string	-	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

3.6.7.2. 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])
```

### 3.6.8. HRIF\_ReadActTcpPos

3.6.8.1. 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	string	-	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

3.6.8.2. 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])
```

### 3.6.9. HRIF\_ReadCmdJointVel

3.6.9.1. 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	string	-	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

3.6.9.2. 示例

```
# 定义返回值空列表
```

```
result = [ ]
```

```
# 读取关节关节命令速度
```

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

```
# 读取关节关节命令速度变量
```

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

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

### 3.6.10. HRIF\_ReadCmdJointVel\_nJ

3.6.10.1. 描述：读取关节命令速度。**注意：这是扩展轴版本所用接口**

✓ 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[n-1]	关节速度	string	-	dJn: 关节 n 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

3.6.10.2. 示例

```
# 定义返回值空列表
result = []

# 读取关节关节命令速度
nRet = cps.HRIF_ReadCmdJointVel_nJ(0,0,result)

# 读取关节关节命令速度变量
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
dJ7 = float(result[6]) dJ8 = float(result[7]) dJ9 = float(result[8])
```

### 3.6.11. HRIF\_ReadActJointVel

3.6.11.1. 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	string	-	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

3.6.11.2. 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])
```

### 3.6.12. HRIF\_ReadActJointVel\_nJ

3.6.12.1. 描述：读取关节实际速度。**注意：这是扩展轴版本所用接口**

✓ 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[n-1]	关节速度	string	-	dJn: 关节 n 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

#### 3.6.12.2. 示例

```
# 定义返回值空列表
result = []

# 读取关节实际速度
nRet = cps.HRIF_ReadActJointVel_nJ(0,0,result)

# 读取关节关节实际速度变量
dJ1 = float(result[0]) dJ2 = float(result[1]) dJ3 = float(result[2])
dJ4 = float(result[3]) dJ5 = float(result[4]) dJ6 = float(result[5])
dJ7 = float(result[6]) dJ8 = float(result[7]) dJ9 = float(result[8])
```

### 3.6.13. HRIF\_ReadCmdTcpVel

3.6.13.1. 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	string	-	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

3.6.13.2. 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])
```

### 3.6.14. HRIF\_ReadActTcpVel

3.6.14.1. 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	string	-	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

3.6.14.2. 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])
```

### 3.6.15. HRIF\_ReadCmdJointCur

3.6.15.1. 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	string	-	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

3.6.15.2. 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])
```



### 3.6.16. HRIF\_ReadCmdJointCur\_nJ

3.6.16.1. 描述：读取关节命令电流。**注意：这是扩展轴版本所用接口**

✓ 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[n-1]	关节电流	string	-	dJn: 关节 n 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

3.6.16.2. 示例

```
# 定义返回值空列表
result = []

# 读取关节命令电流
nRet = cps.HRIF_ReadCmdJointCur_nJ(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])
dJ7 = float(result[6]) dJ8 = float(result[7]) dJ9 = float(result[8])
```

### 3.6.17. HRIF\_ReadActJointCur

3.6.17.1. 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	string	-	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

3.6.17.2. 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])
```

### 3.6.18. HRIF\_ReadActJointCur\_nJ

3.6.18.1. 描述：读取关节实际电流。**注意：这是扩展轴版本所用接口**

✓ 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[n-1]	关节电流	string	-	dJn: 关节 n 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

3.6.18.2. 示例

```
# 定义返回值空列表
result = []

# 读取关节实际电流
nRet = cps.HRIF_ReadActJointCur_nJ(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])
dJ7 = float(result[6]) dJ8 = float(result[7]) dJ9 = float(result[8])
```

### 3.6.19. HRIF\_ReadTcpVelocity

3.6.19.1. 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	string	-	命令速度, 单位[mm/s]
result[1]	实际速度	string	-	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

3.6.19.2. 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
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	string	>=-180,=<180	欧拉角 Rx
result[1]	欧拉角 Ry	string	>=-180,=<180	欧拉角 Ry
result[2]	欧拉角 Rz	string	>=-180,=<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,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
result	返回值	list		传入空列表, result = [ ]
Rx	欧拉角 Rx	float	>=-180,=<180	欧拉角 Rx
Ry	欧拉角 Ry	float	>=-180,=<180	欧拉角 Ry
Rz	欧拉角 Rz	float	>=-180,=<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	string	-	W
result[1]	Xi	string	-	Xi
result[2]	Yj	string	-	Yj
result[3]	Zk	string	-	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,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	float	-	dJ1: 关节 1 坐标, 单位[°]



	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			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	string	-	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
dTTargetJ1 = float(result[0]) dTTargetJ2 = float(result[1]) dTTargetJ3 = float(result[2])
dTTargetJ4 = float(result[3]) dTTargetJ5 = float(result[4]) dTTargetJ6 = float(result[5])
```



### 3.7.4. HRIF\_GetInverseKin\_nJ

3.7.4.1. 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。

**注意：这是扩展轴版本所用接口**

✓ 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-dJn	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	-	dJn: 关节 n 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[n-1]	关节坐标	string	-	dTargetJn: 关节 n 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

## 3.7.4.2. 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, 0, 0, 0, 0]

# 求逆解
nRet = cps.HRIF_GetInverseKin_nJ(0,0,dCoord, rawACS, Tcp, Ucs, result)
```

### 3.7.5. HRIF\_GetForwardKin

3.7.5.1. 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	string	-	<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将所有的工具坐标和用户坐标置 0:</p> <p>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[°]</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	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
dTargt_X = float(result[0]) dTargt_Y = float(result[1]) dTargt_Z = float(result[2])
dTargt_Rx = float(result[3]) dTargt_Ry = float(result[4]) dTargt_Rz = float(result[5])

```



### 3.7.6. HRIF\_GetForwardKin\_nJ

3.7.6.1. 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。

**注意：这是扩展轴版本所用接口**

✓ 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-dJn	需要计算正解的关节坐标	float	-	dJn: 关节 n 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[°]
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]	目标迪 卡尔坐 标	string	-	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 将所有的工具坐标和用户坐标置 0： 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 = []
```

# 定义需要转换的关节位置变量

```
rawACS = [0, 0, 0, 0, 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_nJ(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])
```



### 3.7.7. HRIF\_Base2UcsTcp

3.7.7.1. 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
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	string	-	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

### 3.7.7.2. 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 = [420, 0, 445, 180, 0, 180]

# 定义工具坐标变量
Tcp = [0, 0, 10, 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,dCoord, Tcp, Ucs, result)

# 定义转换后的空间位置结果
dTargt_X = float(result[0]) dTargt_Y = float(result[1]) dTargt_Z = float(result[2])
dTargt_Rx = float(result[3]) dTargt_Ry = float(result[4]) dTargt_Rz = float(result[5])

```

### 3.7.8. HRIF\_UcsTcp2Base

3.7.8.1. 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_R z	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_R z	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	string	-	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

### 3.7.8.2. 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 = [420, 0, 445, 180, 0, 180]

# 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, 10, 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
dTTarget_X = float(result[0]) dTTarget_Y = float(result[1]) dTTarget_Z = float(result[2])
dTTarget_Rx = float(result[3]) dTTarget_Ry = float(result[4]) dTTarget_Rz = float(result[5])

```

### 3.7.9. HRIF\_PoseAdd

3.7.9.1. 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	string	-	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\_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])

### 3.7.10. HRIF\_PoseSub

3.7.10.1. 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
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	string	-	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

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,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])

### 3.7.11. HRIF\_PoseTrans

3.7.11.1. 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
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	string	-	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

## 3.7.11.2. 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\_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])

### 3.7.12. HRIF\_PoseInverse

3.7.12.1. 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
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	string	-	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 I  
Pose = [420, 0, 445, 180, 0, 180]  
# 计算结果  
nRet = cps.HRIF_PoseInverse(0,Pose, 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\_PoseDist

3.7.13.1. 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
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

## 3.7.13.2. 示例

```
# 定义返回值空列表
result = [ ]
# 定义需要计算的空间坐标 1
Pose1 = [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,Pose1, Pose2, result)
```



### 3.7.14. HRIF\_PoseInterpolate

3.7.14.1. 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
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	string	-	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

### 3.7.14.2. 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,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])
```

### 3.7.15. HRIF\_PoseDefdFrame

3.7.15.1. 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
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	string	-	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

## 3.7.15.2. 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]

# 定义需要计算的空间坐标2
Pose2 = [420, 50, 445]

# 定义需要计算的空间坐标3
Pose3 = [470, 50, 445]

# 定义需要计算的空间坐标4
Pose4 = [420, 0, 445]

# 定义需要计算的空间坐标5
Pose5 = [420, 0, 495]
```



---

# 定义需要计算的空间坐标 6

Pose6 = [420,40, 495]

# 计算结果

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

### 3.7.16. HRIF\_CalUcsPlane

3.7.16.1. 描述：通过三点平面法计算 UCS。

✓ 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
dPose1_X- dPose1_Z	空间坐标 1	float	-	点 1 在 Base 坐标系下系统默认 TCP 的位置： 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	-	点 2 在 Base 坐标系下系统默认 TCP 的位置： 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	-	点 3 在 Base 坐标系下系统默认 TCP 的位置： dPose3_X: X 坐标, 单位[mm] dPose3_Y: Y 坐标, 单位[mm] dPose3_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	string	-	计算得出的 UCS 位姿： dRetPose_X: X 坐标, 单位[mm] dRetPose_Y: Y 坐标, 单位[mm] dRetPose_Z: Z 坐标, 单位[mm] dRetPose_Rx: Rx 坐标, 单位[°] dRetPose_Ry: Ry 坐标, 单位[°] dRetPose_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
------	-----	-----	---------	--

### 3.7.16.2. 示例

```
# 定义返回值空列表
```

```
result = [ ]
```

```
# 点1在Base坐标系下系统默认TCP的位置
```

```
Pose1 = [10, 0, 0]
```

```
# 点2在Base坐标系下系统默认TCP的位置
```

```
Pose2 = [0, 10, 0]
```

```
# 点3在Base坐标系下系统默认TCP的位置
```

```
Pose3 = [0, 0, 10]
```

```
# 获取计算结果
```

```
nRet= cps.HRIF_CalUcsPlane(0,Pose1,Pose2,Pose3,result)
```

### 3.7.17. HRIF\_CalUcsLine

3.7.17.1. 描述：通过两点直线法计算 UCS。

✓ 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
dPose1_X- dPose1_Rz	空间坐标 1	float	-	点 1 在 Base 坐标系下系统默认 TCP 的位姿： 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	-	点 2 在 Base 坐标系下系统默认 TCP 的位姿： 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	string	-	计算得出的 UCS 位姿： dRetPose_X: X 坐标, 单位[mm] dRetPose_Y: Y 坐标, 单位[mm] dRetPose_Z: Z 坐标, 单位[mm] dRetPose_Rx: Rx 坐标, 单位[°] dRetPose_Ry: Ry 坐标, 单位[°] dRetPose_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

### 3.7.17.2. 示例

# 定义返回值空列表

```
result = []
```

# 点1在Base坐标系下系统默认TCP的位姿

```
Pose1 = [10, 0, 0, 10, 0, 0]
```

# 点2在Base坐标系下系统默认TCP的位姿

```
Pose2 = [0, 10, 0, 0, 10, 0]
```

# 获取计算结果

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

### 3.7.18. HRIF\_CalTcp3P

3.7.18.1. 描述：通过三点平面法计算 TCP。

✓ 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
dPose1_X- dPose1_Rz	空间坐标 1	float	-	点 1 在 Base 坐标系下系统默认 TCP 的位姿： 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	-	点 2 在 Base 坐标系下系统默认 TCP 的位姿： 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[°]
dPose3_X- dPose3_Rz	空间坐标 3	float	-	点 3 在 Base 坐标系下系统默认 TCP 的位姿： 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[°]

✓ 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	string	-	计算得出的 UCS 位姿： dRetPose_X: X 坐标, 单位[mm] dRetPose_Y: Y 坐标, 单位[mm] dRetPose_Z: Z 坐标, 单位[mm] dRetPose_Rx: Rx 坐标, 单位[°], 一般是 0 dRetPose_Ry: Ry 坐标, 单位[°], 一般是 0 dRetPose_Rz: Rz 坐标, 单位[°], 一般是 0
result[6]	结果质量	string	0/1/2	0: 良好 1: 差 (计算结果最好不用) 2: 异常

✓ **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

### 3.7.18.2. 示例

```
# 定义返回值空列表
result = []

# 点1在Base坐标系下系统默认TCP的位置
Pose1 = [10, 0, 0, 10, 0, 0]

# 点2在Base坐标系下系统默认TCP的位置
Pose2 = [0, 10, 0, 0, 10, 0]

# 点3在Base坐标系下系统默认TCP的位置
Pose3 = [0, 0, 10, 0, 0, 10]

# 获取计算结果
nRet= cps.HRIF_CalTcp3P(0,Pose1,Pose2,Pose3,result)
```



### 3.7.19. HRIF\_CalTcp4P

3.7.19.1. 描述：通过四点平面法计算 TCP。

✓ 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
dPose1_X- dPose1_Rz	空间坐标 1	float	-	点 1 在 Base 坐标系下系统默认 TCP 的位姿： 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	-	点 2 在 Base 坐标系下系统默认 TCP 的位姿： 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[°]
dPose3_X- dPose3_Rz	空间坐标 3	float	-	点 3 在 Base 坐标系下系统默认 TCP 的位置： 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[°]
dPose4_X- dPose4_Rz	空间坐标 3	float	-	点 4 在 Base 坐标系下系统默认 TCP 的位置： dPose4_X: X 坐标, 单位[mm] dPose4_Y: Y 坐标, 单位[mm]

				dPose4_Z: Z 坐标, 单位[mm] dPose4_Rx: Rx 坐标, 单位[°] dPose4_Ry: Ry 坐标, 单位[°] dPose4_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	string	-	计算得出的 UCS 位姿：  dRetPose_X: X 坐标, 单位[mm] dRetPose_Y: Y 坐标, 单位[mm] dRetPose_Z: Z 坐标, 单位[mm] dRetPose_Rx: Rx 坐标, 单位[°], 一般是 0 dRetPose_Ry: Ry 坐标, 单位[°], 一般是 0 dRetPose_Rz: Rz 坐标, 单位[°], 一般是 0
result[6]	结果质量	string	0/1/2	0: 良好 1: 差 (计算结果最好不用) 2: 异常
result[7]-result[10]	源点的错误指示	string	0/1	0: 正常 1: 异常

✓ **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

### 3.7.19.2. 示例

```
# 定义返回值空列表
```

```
result = []
```

```
# 点1在Base坐标系下系统默认TCP的位置
```

```
Pose1 = [10, 0, 0, 10, 0, 0]
```

```
# 点2在Base坐标系下系统默认TCP的位置
```

```
Pose2 = [0, 10, 0, 0, 10, 0]
```

---

# 点3在Base坐标系下系统默认TCP的位置

Pose3 = [0, 0, 10, 0, 0, 10]

# 点4在Base坐标系下系统默认TCP的位置

Pose4 = [0, 0, 0, 0, 0, 0]

# 获取计算结果

nRet= cps.HRIF\_CalTcp4P(0,Pose1,Pose2,Pose3,Pose4,result)

## 3.8. 工具坐标与用户坐标读写指令

### 3.8.1. HRIIF\_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
rbtID	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.HRIIF_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	string	-	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	string	-	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
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	string	-	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	string	-	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
```

### 3.8.9. HRIF\_ConfigTCP

3.8.9.1. 描述：新建指定名称的 TCP 和值。

✓ 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
sTcpName	工具坐标名称	string	-	需要新建的工具坐标名称
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

3.8.9.2. 示例

```
# 需要下发的用户坐标名称
sTcpName = "TCP"

# 设置工具坐标
Pose=[0, 0, 10, 0, 0, 0]

# 新建 Tcp
nRet = cps.HRIF_ConfigTCP(0,sTcpName,Pose)
```

### 3.8.10. HRIF\_ConfigUCS

3.8.10.1. 描述：新建指定名称的 UCS 和值。

✓ 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
sUcsName	用户坐标名称	string	-	需要新建的用户坐标名称
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

3.8.10.2. 示例

```
# 需要下发的用户坐标名称
sUcsName = "UCS"

# 设置工具坐标
Pose=[0, 0, 10, 0, 0, 0]

# 新建 Ucs
nRet = cps.HRIF_ConfigUCS(0,sUcsName,Pose)
```

### 3.8.11. HRIF\_ReadTCPList

3.8.11.1. 描述：读取系统中保存的 TCP 名称列表。

✓ 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	TCP 名称列表	list	-	保存读取到的所有 TCP 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

3.8.11.2. 示例

# 定义保存 TCP 列表的变量

result = []

# 读取用户坐标系名称列表

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

### 3.8.12. HRIF\_ReadUCSList

3.8.12.1. 描述：读取系统中保存的 UCS 名称列表。

✓ 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	UCS 名称列表	list	-	保存读取到的所有 UCS 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

3.8.12.2. 示例

# 定义保存 UCS 列表的变量

result = []

# 读取用户坐标系名称列表

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

### 3.8.13. HRIF\_SetBaseInstallingAngle

3.8.13.1. 描述：设定机座安装角度。

✓ 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
nRotation	机座旋转角度	int	-360~360	设定的机座旋转角度
nTilt	机座倾斜角度	int	-360~360	设定的机座倾斜角度

✓ 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

3.8.13.2. 示例

# 定义机座旋转角度

nRotation= 0

# 定义机座倾斜角度

nTilt= 0

# 设定机座安装角度

nRet = cps.HRIF\_SetBaseInstallingAngle(0,0, nRotation,nTilt)

### 3.8.14. HRIF\_GetBaseInstallingAngle

3.8.14.1. 描述：读取机座安装角度。

✓ 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]	机座旋转角度	string	-360~360	读取的机座旋转角度
result[1]	机座倾斜角度	string	-360~360	读取的机座倾斜角度

✓ 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

3.8.14.2. 示例

# 定义保存角度的变量

result= []

# 读取机座安装角度

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

## 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	string	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
mode = 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。 (此接口功能已屏蔽)

✓ 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。 (此接口功能已屏蔽)

✓ 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	1~50	探寻的直线最大速度, 单位[mm/s]
dMaxAngularVelocity	姿态角速度	float	1~10	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 = 10

# 设置力控探寻姿态角速度

dMaxAngularVelocity = 5

# 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
freedom[0-5]	各方向自由度	list	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
freedom = [0, 0, 0, 0, 0, 0]
# 设置力控自由度状态
nRet = cps.HRIF_SetControlFreedom (0,0,freedom)
```

### 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_z	正方向力阈值	float	1~1000	正方向力最大阈值： 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-dMax_Rz	正方向力矩阈值	float	0.2~50	正方向力矩最大阈值： 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_z	负方向力阈值	float	1~1000	负方向力最大阈值： 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-dMin_Rz	负方向力矩阈值	float	0.2~50	负方向力矩最大阈值： 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

# 设置正方向力最大阈值

---

```
dMax = [500, 500, 500, 50, 50, 50]
# 设置负方向力最大阈值

dMin = [300, 300, 300, 30, 30, 30]
# 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。 (此接口功能已弃用)

✓ 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
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
nMode	是否开启	int	0/1	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

3.9.19.2. 示例

```
# 设置开启力控自由驱动
nMode = 1
# 设置开启力控自由驱动
nRet = cps.HRIF_SetForceFreeDriveMode(0,0,nMode )
```

### 3.9.20. HRIF\_SetFTFreeDriveSpeedMode

3.9.20.1. 描述：设置自由驱动的速度模式。

✓ 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	速度模式	int	0~3	速度模式, 默认为 0 0: 正常速度模式 1: 慢速模式 2: 快速模式 3: 焊接模式

✓ 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

3.9.20.2. 示例

nMode = 3

# 设置速度模式为焊接模式

```
nRet = cps.HRIF_SetFTFreeDriveSpeedMode(0,0,nMode)
```

### 3.9.21. HRIF\_ReadFTFreeDriveSpeedMode

3.9.21.1. 描述：读取设定后的自由驱动速度模式。

✓ 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
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]	速度模式	string	0~3	速度模式, 默认为 0 0: 正常速度模式 1: 慢速模式 2: 快速模式 3: 焊接模式

✓ 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

3.9.21.2. 示例

# 定义返回值空列表

result = []

# 读取设定的自由驱动速度模式

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

mode = int(result[0])

### 3.9.22. HRIF\_ReadFTCabData

3.9.22.1. 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	string	-	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

3.9.22.2. 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])
```

### 3.9.23. HRIF\_SetFreeDriveMotionFreedom

3.9.23.1. 描述：设置力控自由驱动末端自由度。

✓ 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
nX-nRz	各方向自由度	int	0/1	各方向探寻自由度开关： 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

3.9.23.2. 示例

```
# 定义力控自由驱动自由度状态,nX-nRz
df=[0, 0, 1, 0, 0, 0]
# 设置力控自由驱动自由度状态
nRet = cps.HRIF_SetFreeDriveMotionFreedom(0,df)
```

### 3.9.24. HRIF\_SetFTFreeFactor

3.9.24.1. 描述：设置平移柔顺度和旋转柔顺度。

✓ 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
dLinear	平移柔顺度	float	0~100	力控自由驱动平移柔顺度
dAngular	旋转柔顺度	float	0~100	力控自由驱动旋转柔顺度

✓ 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

3.9.24.2. 示例

# 定义平移柔顺度

dLinear = 50

# 定义旋转柔顺度

dAngular = 50

# 设置力控自由驱动平移柔顺度和旋转柔顺度

nRet = cps.HRIF\_SetFTFreeFactor(0, dLinear, dAngular)

### 3.9.25. HRIF\_SetTangentForceBounds

3.9.25.1. 描述：设置 X/Y 方向切向力最大值、最小值和最大上抬速度。

✓ 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
dMax	最小越障力	float	1~100	X/Y 方向触发越障最小力, 单位: N
dMin	已弃用	float	1~100	已弃用参数
dVel	抬升速度	float	5~50	最大抬升速度, 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

3.9.25.2. 示例

# 定义 X/Y 方向触发越障最小力

dMax = 25

# 定义弃用参数

dMin = 1

# 定义越障上抬最大速度

dVel = 30

# 设置 X/Y 方向切向力最大值、最小值和上抬最大速度

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

### 3.9.26. HRIF\_SetFreeDriveCompensateForce

3.9.26.1. 描述：设置 FreeDrive 模式下的定向补偿力大小及矢量方向。

✓ 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
dForce	补偿力	float	0~500	补偿力大小, 单位: N
dX-dZ	力方向向量	float	-	补偿力的矢量方向[x,y,z]

✓ 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

3.9.26.2. 示例

# 定义补偿力大小

dForce = 10

# 定义补偿力在基坐标系下的矢量方向

dX = 0

dY = 0

dZ = 0

# 设置 FreeDrive 模式下的定向补偿力大小和矢量方向[x,y,z]

nRet = cps.HRIF\_SetFreeDriveCompensateForce(0,0,dForce,dX,dY,dZ)

### 3.9.27. HRIF\_SetFTWrenchThresholds

3.9.27.1. 描述：设置力控自由驱动启动阈值（力与力矩）。

✓ 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
dForceThreshold	力	float	0.1~100	自由驱动力启动阈值, 单位[N]
dTorqueThreshold	力矩	float	0.1~10	自由驱动力矩启动阈值, 单位[N.m]

✓ 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

3.9.27.2. 示例

# 定义力阈值

dForceThreshold=10

# 定义力矩阈值

dTorqueThreshold=10

# 设置力控自由驱动启动阈值（力与力矩）

nRet = cps.HRIF\_SetFTWrenchThresholds(0,0,dForceThreshold,dTorqueThreshold)

### 3.9.28. HRIF\_SetMaxFreeDriveVel

3.9.28.1. 描述：设置力控自由驱动最大直线速度及姿态角速度。

✓ 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	1~500	自由驱动的最大直线速度, 单位[mm/s]
dMaxAngularVelocity	姿态角速度	float	1~180	自由驱动的最大姿态角速度, 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

3.9.28.2. 示例

# 定义自由驱动最大直线速度

dMaxLinearVelocity=100

# 定义自由驱动最大角速度

dMaxAngularVelocity=10

# 设置力控自由驱动最大直线速度及姿态角速度

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

### 3.9.29. HRIF\_ReadFTMotionFreedom

3.9.29.1. 描述：读取力控自由驱动的末端自由度。

✓ 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
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[0-5]	各方向自由度	string	0/1	各方向探寻自由度开关： 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

3.9.29.2. 示例

```
result = []
```

// 读取力控自由驱动的末端自由度

```
nRet = cps.HRIF_ReadFTMotionFreedom(0,0,result)
nx = int(result[0])
ny = int(result[1])
nz = int(result[2])
nRx = int(result[3])
nRy = int(result[4])
nRz = int(result[5])
```

### 3.9.30. HRIF\_SetMaxSearchDistance

3.9.30.1. 描述：设置各自由度力控探寻最大距离。

✓ 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
Dis_X	X 方向探寻边界	float	10~1000	X 方向探寻边界, 单位: mm
Dis_Y	Y 方向探寻边界	float	10~1000	Y 方向探寻边界, 单位: mm
Dis_Z	Z 方向探寻边界	float	10~1000	Z 方向探寻边界, 单位: mm
Dis_RX	RX 方向探寻边界	float	1~360	RX 方向探寻边界, 单位: °
Dis_RY	RY 方向探寻边界	float	1~360	RY 方向探寻边界, 单位: °
Dis_RZ	RZ 方向探寻边界	float	1~360	RZ 方向探寻边界, 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

3.9.30.2. 示例

# 定义各自由度力控探寻最大距离

```
Dis_X = 300
Dis_Y = 300
Dis_Z = 300
Dis_RX = 20
Dis_RY = 20
Dis_RZ = 20
```

# 设置各自由度力控探寻最大距离

```
nRet = cps.HRIF_SetMaxSearchDistance(0,0,Dis_X,Dis_Y,Dis_Z,Dis_RX,Dis_RY,Dis_RZ)
```

### 3.9.31. HRIF\_SetSteadyContactDeviationRange

3.9.31.1. 描述：设置恒力控稳定阶段边界。

✓ 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
Pos_X~Pos_Z	正方向边界	float	10~1000	正方向边界 (与探寻方向一致), 单位: mm
Pos_RX~Pos_RZ	正方向边界	float	1~360	正方向边界 (与探寻方向一致), 单位: °
Neg_X ~ Neg_Z	负方向边界	float	-1000~-10	负方向边界 (与探寻方向相反), 单位: mm
Neg_RX ~ Neg_RZ	负方向边界	float	-360~-1	负方向边界 (与探寻方向相反), 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

3.9.31.2. 示例

```
Pos_X = 100
Pos_Y = 100
Pos_Z = 100
Pos_RX = 20
Pos_RY = 20
Pos_RZ = 20
```

```
Neg_X = -100
Neg_Y = -100
Neg_Z = -100
Neg_RX = -20
Neg_RY = -20
Neg_RZ = -20
```

# 设置恒力控稳定阶段边界

```
nRet = cps.HRIF_SetSteadyContactDeviationRange(0,0,Pos_X,Pos_Y,Pos_Z,Pos_RX,Pos_RY,Pos_RZ,
```

---

Neg\_X,Neg\_Y,Neg\_Z,Neg\_RX,Neg\_RY,Neg\_RZ)

### 3.9.32. HRIF\_SetDepthThresholdForDampingArea

3.9.32.1. 描述：设置虚拟墙开始产生阻尼时的距离阈值。

✓ 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
dDepth	阈值大小	float	-	触发虚拟墙的阈值大小, 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

3.9.32.2. 示例

# 定义阈值大小

dDepth = 100

# 设置虚拟墙开始产生阻尼时的距离阈值

nRet = cps.HRIF\_SetDepthThresholdForDampingArea(0, 0, dDepth)

### 3.9.33. HRIIF\_AddSafePlane

3.9.33.1. 描述：添加虚拟墙平面；虚拟墙平面即安全平面，只是相应的功能不同，如果添加的安全平面名称已经存在，会报 20006 错误；最后一个参数为是否激活，如果不激活，不会检查 UCS 是否存在；

✓ 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
Name	平面名称	string	——	要添加的平面名称
UcsName	用户坐标名称	string	——	所使用的用户坐标系
Mode	安全模式	int	0/1	正常/缩减模式
Display	显示	int	0/1	是否显示目标平面
Switch	启用	int	0/1	是否启用目标平面

✓ 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

### 3.9.33.2. 示例

# 定义平面名称

Name = "plane1"

# 定义使用的用户坐标名称

UcsName = "Plane\_1"

# 定义安全模式

Mode = 0

# 定义是否显示

Display = 0

# 定义是否启用

Switch = 1

# 添加虚拟墙平面



---

```
nRet = cps.HRIF_AddSafePlane(0, 0, Name, UcsName , Mode, Display, Switch)
```

### 3.9.34. HRIF\_UpdateSafePlane

3.9.34.1. 描述：修改更新虚拟墙平面属性；虚拟墙平面即安全平面，只是相应的功能不同；最后一个参数为是否激活，如果不激活，不会检查 UCS 是否存在；

✓ 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
Name	平面名称	string	——	要修改的平面名称
UcsName	用户坐标名称	string	——	所使用的用户坐标系
Mode	安全模式	int	0/1	正常/缩减模式
Display	显示	int	0/1	是否显示目标平面
Switch	启用	int	0/1	是否启用目标平面

✓ 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

### 3.9.34.2. 示例

# 定义平面名称

Name = "plane1"

# 定义使用的用户坐标名称

UcsName = "Plane\_1"

# 定义安全模式

Mode = 0

# 定义是否显示

Display = 0

# 定义是否启用

Switch = 1

# 修改虚拟墙平面



---

```
nRet = cps.HRIF_UpdateSafePlane(0, 0, Name, UcsName , Mode, Display, Switch)
```

### 3.9.35. HRIF\_DelSafePlane

3.9.35.1. 描述：删除虚拟墙平面；虚拟墙平面即安全平面，只是相应的功能不同，如果指定的安全平面名称

不存在，报 20006 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
rbtID	机器人 ID	int	0~5	机器人 ID 号, 默认值= 0
Name	平面名称	string	——	要删除的平面名称

✓ 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

3.9.35.2. 示例

# 定义要删除的平面名称

Name = "planel"

# 删除虚拟墙平面

nRet = cps.HRIF\_DelSafePlane(0, 0, Name )

### 3.9.36. HRIF\_ReadSafePlaneList

3.9.36.1. 描述：返回结果为所有安全平面的名字清单，虚拟墙平面即安全平面，只是相应的功能不同；

✓ 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	所有虚拟墙平面名称	list[string]	—	虚拟墙平面名称

✓ 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

### 3.9.36.2. 示例

# 定义接收所有虚拟墙平面名称的列表

result = []

# 读取当前所有的虚拟墙平面名称

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

### 3.9.37. HRIF\_ReadSafePlane

3.9.37.1. 描述：返回结果为指定安全平面的详细参数；虚拟墙平面即安全平面，只是相应的功能不同，如果指定的安全平面不存在，返回结果为空；

✓ 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
Name	平面名称	string	——	要读取信息的平面名称

✓ 输入变量

输出变量	名称	数据类型	有效范围	内容
result[0]	用户坐标名称	string	——	所使用的用户坐标系
result[1]	安全模式	string	0/1	正常/缩减模式
result[2]	显示	string	0/1	是否显示目标平面
result[3]	启用	string	0/1	是否启用目标平面

✓ 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

### 3.9.37.2. 示例

```

result = []
# 定义要读取的平面名称
BorderName = "planel"
# 读取目标虚拟墙平面的详细信息
nRet = cps.HRIF_ReadSafePlane(0, 0, BorderName,result)

```



### 3.9.38. HRIF\_SetFTCalibration

3.9.38.1. 描述: 计算力控标定结果, 接收 8~16 组标定数据, 每组数据包含 12 个参数 (6 个位置参数 + 6 个力控参数)。

✓ 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
pointnum	标定点位数据	int	8~16	标定点位个数, 8~16 个
Point[n]	标定点位置信息	list	-	标定点位置与标定点力信息, 依次存放 X, Y, Z, Rx, Ry, Rz, Fx, Fy, Fz, Tx, Ty, Tz, 点位数量需与 pointnum 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

### 3.9.38.2. 示例

# 写入力控标定数据

```
pointnum = 8
Point = [
[420.000, -0.001, 445.004, -180.000, 0.001, -180.000, 0.1, 0, 0, 0.001, -0.002, 0.007],
[419.999, -0.002, 445.000, 114.295, 0.000, 180.000, 2.8, -1.3, -1.9, 0.021, 0.037, 0.024],
[420.000, 0.001, 445.005, -98.256, 0.001, -179.999, -3, 1.5, -2.8, -0.013, -0.042, 0.003],
[419.998, 0.002, 445.003, 180.000, -59.664, -180.000, -1, -2.6, -1.6, 0.042, -0.02, -0.001],
[420.001, 0.002, 444.995, 180.000, 53.128, 180.000, 1.3, 2.4, -1.4, -0.027, 0.013, 0.018],
[419.302, 0.000, 435.845, 127.693, -50.301, -179.484, 0.7, -3, -2.1, 0.046, 0.003, 0.019],
[420.000, -0.002, 445.012, -92.510, 1.413, -150.687, -2.8, 1.6, -3.3, -0.012, -0.045, 0.012],
[409.266, -34.590, 421.446, -82.100, 40.722, -127.015, -1.2, 3.1, -3.8, -0.032, -0.019, 0.026] ]
```

# 应用力控标定数据

```
nRet = cps.HRIF_SetFTCalibration(0,0,pointnum,Point)
```

### 3.9.39. HRIF\_ReadForceData

3.9.39.1. 描述：读取力传感器原始数据；

✓ 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[0~5]	原始力数据	list	—	力传感器提供的原始力数据： Pose.dX: x 方向的原始力数据; Pose.dY: y 方向的原始力数据; Pose.dZ: z 方向的原始力数据; Pose.dRX: RX 方向的原始力数据; Pose.dRY: RY 方向的原始力数据; Pose.dRZ: RZ 方向的原始力数据;

✓ 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

3.9.39.2. 示例

```
result = []
# 读取当前力传感器原始数据
nRet=cps.HRIF_ReadForceData(0, 0, result)
```

### 3.9.40. HRIF\_SetInitializeForceSensor

3.9.40.1. 描述：设置力控标定结果。

✓ 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
calibrationForce	x-z 方向标定力	list[float]	-	x-z 方向标定力
torque	x-z 方向力矩	list[float]	-	x-z 方向力矩
gravity	重力	float	-	重力
cmOffset	重心偏移	list[float]	-	x-z 方向重心偏移
installRotationAngel	安装角度	float	-	机器安装角度

✓ 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

3.9.40.2. 示例

# 定义标定力

```
calibrationForce = [1,2,3]
```

# 定义力矩

```
torque = [1,2,3]
```

# 定义重力

```
gravity = 10
```

# 定义重心偏移

```
cmOffset = [1,1,1]
```

# 定义安装角度

```
installRotationAngel = 0
```

# 设置力控标定结果

```
nRet = cps.HRIF_SetInitializeForceSensor(0, 0, calibrationForce, torque, gravity, cmOffset, installRotationAngel)
```

### 3.9.41. HRIF\_GetLastCalibParams

3.9.41.1. 描述：获取力控标定结果。

✓ 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[0~2]	x-z 方向标定力	string	-	x-z 方向标定力
result[3~5]	x-z 方向力矩	string	-	x-z 方向力矩
result[6]	重力	string	-	重力
result[7~9]	重心偏移	string	-	x-z 方向重心偏移
result[10]	安装角度	string	-	机器安装角度

✓ 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

3.9.41.2. 示例

```
result = []
# 设置力控标定结果
nRet = cps.HRIF_GetLastCalibParams(0, 0, result)
```

### 3.10.通用运动类控制指令

#### 3.10.1. HRIIF\_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.HRIIF_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> ] <b>要求大于速度</b>
dRadius	过渡半径	float	-	过渡半径，单位[mm]
nIsUseJoint	是否使用关节角度	int	0	默认为 0(其它值无效)，默认使用关节角度
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
nPontList= 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

nRet = cps.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\_WayPointRel\_nJ

3.10.4.1. 描述：路点相对运动。**注意：这是扩展轴版本所用接口**

✓ 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_Jn: 关节 n 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> ] <b>要求大于速度</b>
dRadius	过渡半径	float	-	过渡半径，单位[mm]
nIsUseJoint	是否使用关节角度	int	0	默认为 0(其它值无效)，默认使用关节角度
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

## 3.10.4.2. 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

nRet = cps.HRIF\_WayPointRel\_nJ(0,0,nType, nPointList, Pos, rawACT, nrelMoveType, nAxisMask, nTarget,  
sTcpName, sUcsName, dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)

### 3.10.5. HRIF\_IsMotionDone

3.10.5.1. 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	bool	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 = []

# 判断机器人是否处于运动状态
nRet = cps.HRIF_IsMotionDone(0,0,result)
```

### 3.10.6. HRIF\_IsBlendingDone

3.10.6.1. 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	bool	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

3.10.6.2. 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)
```

### 3.10.7. HRIF\_WayPointEx

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: 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	-	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:  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	用户坐标值	float	-	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:  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[°]
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[°/s <sup>2</sup> ], 空间运动时 XYZ 单位[mm/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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= [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
strCmdID = "0"

# 执行路点运动

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

### 3.10.8. HRIF\_WayPointEx\_nJ

3.10.8.1. 描述：路点运动。注意：这是扩展轴版本所用接口

✓ 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-dJn	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 dJn: 关节 n 坐标, 单位[°]
dTcp_X-dTcp_Rz	工具坐标值	float	-	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: 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	用户坐标值	float	-	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:  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[°]
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[°/s <sup>2</sup> ], 空间运动时 XYZ 单位[mm/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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

### 3.10.8.2. 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

strCmdID = "0"

# 执行路点运动

nRet = cps.HRIF\_WayPointEx\_nJ(0,0,nMoveType , Point, RawACSPoints, Tcp, Ucs, dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)



### 3.10.9. HRIF\_WayPoint

3.10.9.1. 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[°/s <sup>2</sup> ]，空间运动时 XYZ 单位[mm/s <sup>2</sup> ]，Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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
------	-----	-----	---------	--

### 3.10.9.2. 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
strCmdID = "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)
```

### 3.10.10. HRIIF\_WayPoint\_nJ

3.10.10.1. 描述：路点运动。**注意：这是扩展轴版本所用接口**

✓ 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-dJn	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 dJn: 关节 n 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[°/s <sup>2</sup> ]，空间运动时 XYZ 单位[mm/s <sup>2</sup> ]，Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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

### 3.10.10.2.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

strCmdID = "0"

# 执行路点运动

nRet = cps.HRIF\_WayPoint\_nJ(0,0,nMoveType , Point, rawACS, sTcpName , sUcsName, dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)

### 3.10.11. HRIIF\_WayPoint2

3.10.11.1. 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	0/1/2	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[°]
				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-dJ6	关节目标位置	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[°]
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[°/s <sup>2</sup> ], 空间运动时 XYZ 单位[mm/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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\_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

### 3.10.11.2.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
strCmdID = "0"
# 执行路点运动
nRet = cps.HRIF_WayPoint2(0,0,nMoveType ,EndPos, AuxPos, AcsPose, sTcpName , sUcsName, dVelocity,
dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID)
```

### 3.10.12. HRIIF\_WayPoint2\_nJ

3.10.12.1. 描述：路点运动。**注意：这是扩展轴版本所用接口**

✓ 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	0/1/2	0: 关节运动 1: 直线运动 2: 圆弧运动
dEndPos_X-dEndPos_R_z	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-dJn	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 dJn: 关节 n 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[°/s <sup>2</sup> ], 空间运动时 XYZ 单位[mm/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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\_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

### 3.10.12.2.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

strCmdID = "0"

# 执行路点运动

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

### 3.10.13. HRIIF\_MoveJ

3.10.13.1. 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[°/s <sup>2</sup> ] <b>要求大于速度</b>
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

### 3.10.13.2.示例

# 定义空间目标位置

```
Point = [ 0, 0, 90, 0, 90, 0 ]
```

# 定义关节目标位置

```
RawACSpoints = [ 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

strCmdID = "0"

# 执行路点运动

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

### 3.10.14. HRIIF\_MoveJ\_nJ

3.10.14.1. 描述：关节运动。**注意：这是扩展轴版本所用接口**

✓ 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-dJn	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: dJn: 关节 n 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[°/s <sup>2</sup> ] <b>要求大于速度</b>
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

### 3.10.14.2.示例

# 定义空间目标位置

Point = [ 0, 0, 90, 0, 90, 0 ]

# 定义关节目标位置

RawACSpoints = [ 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

strCmdID = "0"

# 执行路点运动

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

### 3.10.15. HRIIF\_MoveL

3.10.15.1. 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> ] <b>要求大于速度</b>

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

### 3.10.15.2.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

```
RawACSPoints = [ 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

strCmdID = "0"

# 执行路点运动

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



### 3.10.16. HRIIF\_MoveL\_nJ

3.10.16.1. 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-dJn	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： dJn: 关节 n 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> ] <b>要求大于速度</b>
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

### 3.10.16.2.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

```
RawACSPoints = [ 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

---

```
strCmdID = "0"
```

```
# 执行路点运动
```

```
nRet = cps.HRIF_MoveL_nJ(0,0, Point, RawACSpoints, sTcpName, sUcsName, dVelocity, dAcc,  
dRadius,nIsSeek, nIOBit, nIOState, strCmdID)
```



### 3.10.17. HRIIF\_MoveC

3.10.17.1. 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/s <sup>2</sup> ]，Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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

### 3.10.17.2.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

strCmdID = "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)
```



### 3.10.18. HRIIF\_MoveC\_nJ

3.10.18.1. 描述：圆弧轨迹运动。注意：这是扩展轴版本所用接口

✓ 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/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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"
joints	结束点的参考关节角度	list		结束点的参考关节角度，用于判断联动轴是否需要运动 joints[i]:第 i 轴的角度，单位[°]
nIsUseCurRefAC S	是否使用参考的关节角度	int	0/1	0：使用结束点的参考关节角度 1：扩展轴不运动

✓ 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
--	--	--	--	---------------------

---

### 3.10.18.2.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

strCmdID = "0"

# 参考的关节角度

joints = [0,0,0,0,0,0,0,0,0]

# 定义是否使用参考的关节角度

nIsUseCurRefACS= 0



---

# 执行路点运动

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

### 3.10.19. HRIIF\_MoveZ

3.10.19.1. 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> ] <b>要求大于速度</b>
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
--	--	--	--	---------------------

## 3.10.19.2.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

strCmdID = "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)

### 3.10.20. HRIIF\_MoveE

3.10.20.1. 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:不使用固定姿态 1:使用固定姿态
nMoveType	运动类型	int	0/1	0: 椭圆圆弧 1: 整个椭圆
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/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ] <b>要求大于速度</b>
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

## 3.10.20.2.示例

# 示教点1

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

# 示教点2

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

# 示教点3

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

# 示教点4

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

# 示教点5

dP5 = [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  
strCmdID = "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)
```

### 3.10.21. HRIIF\_MoveS

3.10.21.1.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[°/s <sup>2</sup> ] <b>要求大于速度</b>
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

3.10.21.2.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

strCmdID = "0"

# 执行螺旋轨迹运动

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

### 3.10.22. HRIIF\_MoveLinearWeave

3.10.22.1. 描述：直线摆焊运动。

✓ 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
StartPoint	开始点位置	list[float]	-	开始点位置
EndPoint	结束点位置	list[float]	-	结束点位置
dVelocity	速度	float	-	运动最大速度, 单位[mm/s]
dAcc	加速度	float	-	运动最大加速度, 单位[mm/s <sup>2</sup> ] <b>要求大于速度</b>
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
dAmplitude	宽度	float	-	宽度, 摆焊的幅值, 单位[mm]
dInterValDistance	间距	float	-	摆焊的间距, 此值仅供参考, 机器人系统会自动微调该值以确保运动轨迹经过若干个完整周期的摆动后能通过起始点和结束点, 单位[mm]
nWeaveFrameType	选择方式	int	0/1	用于决定摆焊平面和摆焊方向的选择方式。 0: 摆焊平面的+X 方向为从 StartPos 到 EndPos, +Z 方向 (即摆焊平面的法线方向) 为跟刀具坐标系的+Z 相同的方向, +Y 方向由右手定则确定。摆焊启动时的运动方向跟+Y 方向同侧。 1: 摆焊平面的+X 方向为从 StartPos 到 EndPos, Z 方向跟刀具坐标系的 Z 方向平行, +Y 方向跟刀具坐标系的+Y 方向同侧。摆焊启动时的运动方向跟+Y 方向同侧。
dElevation	仰角	float	-	摆焊的仰角, 单位[°]
dAzimuth	方向角	float	-	摆焊的方向角, 即绕摆焊平面法向量的旋转

				角, 单位[°]
dCentreRise	中心隆起量	float	-	摆焊的中心隆起量, 即摆焊中心处焊炬的隆起量, 单位[mm]
nEnableWaitTime	是否等待	int	0/1	是否开启转折点等待时间: 0:不使用 1:使用
nPosiTime	正等待时间	float	-	正向转折点等待时间, 单位[ms]
nNegaTime	负等待时间	float	-	负向转折点等待时间, 单位[ms]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称
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"
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[0~5]	点位	string	-	摆焊结束点计算的位置

✓ 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

### 3.10.22.2.示例

# 定义开始点位置

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

# 定义结束点位置

```
EndPoint = [470, 0, 445, 180, 0, 180]
```

# 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
# 定义宽度

dAmplitude = 10
# 定义间距

dInterValDistance = 10
# 定义方式

nWeaveFrameType = 0
# 定义仰角

dElevation = 10
# 定义方向角

dAzimuth = 10
# 定义中心隆起量

dCentreRise = 5
# 定义是否等待

nEnableWaitTime = 0
# 定义正向转折点等待时间

nPosiTime = 10
# 定义负向转折点等待时间

nNegaTime = 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

sCmdID = "0"
# 执行直线摆焊运动

nRet=cps.HRIF_MoveLinearWeave(0, 0, StartPoint, EndPoint, dVelocity, dAcc, dRadius, dAmplitude,
dInterValDistance, nWeaveFrameType, dElevation, dAzimuth, dCentreRise, nEnableWaitTime, nPosiTime,
nNegaTime, sTcpName, sUcsName, sCmdID)
```



### 3.10.23. HRIIF\_MoveCircularWeave

3.10.23.1. 描述：圆弧摆焊运动。

✓ 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
StartPoint	开始点位置	list[float]	-	开始点位置
AuxPoint	经过点位置	list[float]	-	经过点位置
EndPoint	结束点位置	list[float]	-	结束点位置
dVelocity	速度	float	-	运动最大速度, 单位[mm/s]
dAcc	加速度	float	-	运动最大加速度, 单位[mm/s <sup>2</sup> ]要求大于速度
dRadius	过渡半径	float	-	过渡半径, 单位[mm]
nOrientMode	是否使用固定姿态	int	0/1	是否使用固定姿态, 0: 不使用固定姿态 1: 使用固定姿态
nMoveWhole	轨迹	int	0/1	0: 整圆轨迹 1: 圆弧轨迹
nMoveWholeLen	圈数	int	-	当使用圆弧运动时该参数无效, 通过计算三个点位来确定圆弧路径及弧度; 当使用整圆运动时表示整圆的圈数
dAmplitude	宽度	float	-	宽度, 摆焊的幅值, 单位[mm]
dInterValDistance	间距	float	-	摆焊的间距, 此值仅供参考, 机器人系统会自动微调该值以确保运动轨迹经过若干个完整周期的摆动后能通过起始点和结束点, 单位[mm]
nWeaveFrameType	选择方式	int	0/1	用于决定摆焊平面和摆焊方向的选择方式。 0: 摆焊平面的+X 方向为从 StartPos 到 EndPos, +Z 方向 (即摆焊平面的法线方向)



				为跟刀具坐标系的+Z 相同的方向, +Y 方向由右手定则确定。摆焊启动时的运动方向跟+Y 方向同侧。  1: 摆焊平面的+X 方向为从 StartPos 到 EndPos, Z 方向跟刀具坐标系的 Z 方向平行, +Y 方向跟刀具坐标系的+Y 方向同侧。摆焊启动时的运动方向跟+Y 方向同侧。
dElevation	仰角	float	-	摆焊的仰角, 单位[°]
dAzimuth	方向角	float	-	摆焊的方向角, 即绕摆焊平面法向量的旋转角, 单位[°]
dCentreRise	中心隆起量	float	-	摆焊的中心隆起量, 即摆焊中心处焊炬的隆起量, 单位[mm]
nEnableWaitTime	是否等待	int	0/1	是否开启转折点等待时间: 0:不使用 1:使用
nPosiTime	正等待时间	float	-	正向转折点等待时间, 单位[ms]
nNegaTime	负等待时间	float	-	负向转折点等待时间, 单位[ms]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称
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"
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[0~5]	点位	string	-	摆焊结束点计算的位置

✓ 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

## 3.10.23.2.示例

```
result = []  
  
# 定义开始点位置  
StartPoint = [420, 0, 445, 180, 0, 180]  
  
# 定义经过点位置  
AuxPoint = [420, 50, 445, 180, 0, 180]  
  
# 定义结束点位置  
EndPoint = [470, 0, 445, 180, 0, 180]  
  
# 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  
  
# 定义固定姿态  
nOrientMode = 0  
  
# 定义轨迹  
nMoveWhole = 0  
  
# 定义圈数  
nMoveWholeLen = 10  
  
# 定义宽度  
dAmplitude = 10  
  
# 定义间距  
dInterValDistance = 10  
  
# 定义方式  
nWeaveFrameType = 0  
  
# 定义仰角  
dElevation = 10  
  
# 定义方向角  
dAzimuth = 10  
  
# 定义中心隆起量
```

---

```
dCentreRise = 5
# 定义是否等待
nEnableWaitTime = 0
# 定义正向转折点等待时间
nPosiTime = 10
# 定义负向转折点等待时间
nNegaTime = 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
sCmdID = "0"
# 执行圆弧摆焊运动
nRet=cps.HRIF_MoveCircularWeave(0, 0, StartPoint, AuxPoint, EndPoint, dVelocity, dAcc, dRadius,
nOrientMode, nMoveWhole, nMoveWholeLen, dAmplitude, dInterValDistance, nWeaveFrameType, dElevation,
dAzimuth, dCentreRise, nEnableWaitTime, nPosiTime, nNegaTime, sTcpName, sUcsName, sCmdID, result)
```

### 3.10.24. HRIF\_ShortJogJ

3.10.24.1.描述：关节短点动，运动距离 2°，最大速度不超过 10°/s。

✓ 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
nAxisID	关节轴 ID	int	0~5	运动的目标轴 ID , 对应关节 J1-J6
nDerection	运动方向	int	0/1	0: 负方向 1: 正方向

✓ 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

3.10.24.2.示例

# 关节轴 ID

nAxisID = 0

# 运动方向

nDerection = 0

# 关节短点动

nRet= cps.HRIF\_ShortJogJ(0,0, nAxisID , nDerection)

### 3.10.25. HRIF\_ShortJogL

3.10.25.1.描述：空间坐标短点动，运动距离 2mm，最大速度不超过 10mm/s.

✓ 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
nAxisID	坐标系轴 ID	int	0~5	运动的目标轴 ID , 对应坐标 X-Rz
nDerection	方向	int	0/1	0: 负方向 1: 正方向

✓ 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

3.10.25.2.示例

# 坐标系轴 ID

nAxisID = 0

# 运动方向

nDerection = 0

# 空间短点动

nRet= cps.HRIF\_ShortJogL(0,0, nAxisID , nDerection)

### 3.10.26. HRIIF\_LongJogJ

3.10.26.1. 描述：关节长点动，最大运动速度不超过 10°/s.

✓ 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
nAxisID	轴 ID	int	0~5	运动的目标轴 ID , 对应关节 J1-J6
nDerection	方向	int	0/1	0: 负方向 1: 正方向
nState	状态	int	0/1	0: 关闭 1: 开启

\* 注. 须与 HRIIF\_LongMoveEvent 配合使用。

✓ 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

3.10.26.2. 示例

```
# 坐标系轴 ID
nAxisID = 0

# 运动方向
nDerection = 0

# 设置状态
nState = 1

# 关节长点动
nRet= cps.HRIIF_LongJogJ(0,0, nAxisID , nDerection, nState)
```

### 3.10.27. HRIF\_LongJogL

3.10.27.1.描述：空间长点动，最大运动速度不超过 30mm/s.

✓ 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
nAxisID	轴 ID	int	0~5	运动的目标轴 ID , 对应空间坐标 X-Rz
nDerection	方向	int	0/1	0: 负方向 1: 正方向
nState	状态	int	0/1	0: 关闭 1: 开启

\* 注. 须与 HRIF\_LongMoveEvent 配合使用。

✓ 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

3.10.27.2.示例

```
# 坐标系轴 ID
nAxisID = 0

# 运动方向
nDerection = 0

# 设置状态
nState = 1

# 空间长点动
nRet= cps.HRIF_LongJogL(0,0, nAxisID , nDerection, nState)
```

### 3.10.28. HRIIF\_LongMoveEvent

3.10.28.1.描述: 长点动继续指令, 当开始长点动之后, 要按 200 毫秒或更短时间为时间周期发送一次该指令, 否则长点动会停止。

✓ 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

#### 3.10.28.2.示例

# 长点动继续指令

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

## 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	>=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	string	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	string	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	string	>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_EMovingPathL 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> ] <b>要求大于速度</b>
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. HRIIF\_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 HRIIF_MovePathJ 共用 1: 下发空间点位, 与 HRIIF_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.HRIIF_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 = [0,0,0,0,0,0,0,0,5,0,5,0,0,0,10,0,10,0,
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. HRIIF\_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 HRIIF\_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.HRIIF_MovePathL(0,0,sTrackName)
```



### 3.11.13. HRIIF\_MovePathJOL

3.11.13.1.描述：启动在线实施规划的 MovePathJ。

✓ 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 ID 号, 默认值=0
dVel	关节速度	float	-	单位[°/s]
dAcc	关节加速度	float	-	单位[°/s^2]
dTol	过渡参数	float	>=0.1	单位[°]
RawACSpoints	关节目标位置	list[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[°]
nIsSetIO	各点是否设置 IO	list[int]	0,1	1:该点位运动结束后时要设置 IO; 0:该点位运动结束后时不设置 IO;
nEndDOMask	各个需要更改的 EndDO 按 bit 标识	list[int]	-	EndDO0~EndDO7 对应到 nEndDOMask 的 bit0~bit7, 如果某个 bit 是 1, 则其对应的 EndDO 的值需要更改, 否则无需更改。
nEndDOVal	各个需要更改的 EndDO 的目标状态	list[int]	-	EndDO0~EndDO7 对应到 nEndDOVal 的 bit0~bit7。 0: 低电平 1: 高电平
nBoxDOMask	各个需要更改的 BoxDO 按 bit 标识	list[int]	-	BoxDO0~BoxDO7 对应到 nBoxDOMask 的 bit0~bit7, 如果某个 bit 是 1, 则其对应的 BoxDO 的值需要更改, 否则无需更改。
nBoxDOVal	各个需要更改的 BoxDO 的目标状态	list[int]	-	BoxDO0~BoxDO7 对应到 nBoxDOVal 的 bit0~bit7。 0: 低电平 1: 高电平
nBoxCOMask	需要更改的 BoxCO 按 bit 标识	list[int]	-	BoxCO0~BoxCO7 对应到 nBoxCOMask 的 bit0~bit7, 如果某个 bit 是 1, 则其对应的 BoxCO 的值需要更改, 否则无需更改。

nBoxCOVal	各个需要更改的 BoxCO 的目标状态	list[int]	-	BoxCO0~BoxCO7 对应到 nBoxCOVal 的 bit0~bit7。 0: 低电平 1: 高电平
nBoxAOCH0_Mask	BoxAOCH0 是否需要更改的标识	list[int]	0/1	0: 无需更改 1: 需要更改
nBoxAOCH0_Mode	模式	list[int]	1/2	1: 电压模式 2: 电流模式
nBoxAOCH1_Mask	BoxAOCH1 是否需要更改的标识	list[int]	0/1	0: 无需更改 1: 需要更改
nBoxAOCH1_Mode	模式	list[int]	1/2	1: 电压模式 2: 电流模式
dbBoxAOCH0_Val	各点对应模拟量值	list[float]	0~20	电流模式: 4-20mA 电压模式: 0-10V
dbBoxAOCH1_Val	各点对应模拟量值	list[float]	0~20	电流模式: 4-20mA 电压模式: 0-10V

\* 注: 以上参数中, 从 RawACSPoints 到 dBoxAOCh1\_Val 依次为一个点的完整数据, 均为必选。一次下发的指令, 最少要包含 2 个点的数据, 最多可包含 5000 个点的数据。

✓ 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

### 3.11.13.2.示例

# 定义速度

dVel = 15

# 定义加速度

dAcc = 20

# 定义过渡参数

dTol = 2

# 定义各点关节目标位置, 6 个一组

RawACSPoints = [0,0,90,0,90,0, 0,0,91,0,90,0, 0,0,92,0,90,0]

# 各点是否设置 IO

---

```
nIsSetIO = [1,1,1]
# 需要更改的各点 EndDO

nEndDOMask = [7,7,7]
# 需要更改的各点 EndDO 的目标状态

nEndDOVal = [2,2,2]
# 需要更改的各点 BoxDO

nBoxDOMask = [86,86,86]
# 需要更改的各点 BoxDO 的目标状态

nBoxDOVal = [255,255,255]
# 需要更改的各点 BoxCO

nBoxCOMask = [255,255,255]
# 需要更改的各点 BoxCO 的目标状态

nBoxCOVal = [169,169,169]
# 各点 BoxAOCH0 是否需要更改的标识

nBoxAOCH0_Mask = [1,1,1]
# 模式

nBoxAOCH0_Mode = [2,2,2]
# 各点 BoxAOCH1 是否需要更改的标识

nBoxAOCH1_Mask = [1,1,1]
# 模式

nBoxAOCH1_Mode = [1,1,1]
# 各点对应模拟量值

dbBoxAOCH0_Val = [6.66,6.66,6.66]
# 各点对应模拟量值

dbBoxAOCH1_Val = [9.99,9.99,9.99]
# 开始运动

nRet = cps.HRIF_MovePathJOL(0,0,dVel, dAcc, dTol, RawACSPoints, nIsSetIO, nEndDOMask, nEndDOVal,
nBoxDOMask, nBoxDOVal, nBoxCOMask, nBoxCOVal, nBoxAOCH0_Mask, nBoxAOCH0_Mode,
nBoxAOCH1_Mask, nBoxAOCH1_Mode, dbBoxAOCH0_Val, dbBoxAOCH1_Val)
```

### 3.11.14. HRIIF\_GetMovePathJOLIndex

3.11.14.1.描述：获取 MovePathJOL 运动当前的点位索引号及轨迹运动所有点总数。

✓ 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]	点位索引号	string	-	MovePathJOL 运动当前的点位索引号
result[1]	点总数	string	-	轨迹运动所有点总数

✓ 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

3.11.14.2.示例

```
result = []
# 读取点位索引号与总数
nRet = cps.HRIIF_GetMovePathJOLIndex(0,0,result)
```

### 3.11.15. HRIF\_SetMovePathOverride

3.11.15.1.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

3.11.15.2.示例

# 速度比

MovePathOverride= 0.01

#设置 MovePath 速度比

```
nRet = cps.HRIF_SetMovePathOverride(0,0,MovePathOverride)
```



### 3.11.16. HRIIF\_InitPath

3.11.16.1.描述：初始化轨迹。

✓ 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
nRawDataType	原始点位类型	int	0/1	0: 关节点位 1: 空间点位
sPathName	轨迹名称	string	-	轨迹名称, 由下划线, 英文字母或者数字组成, 最长 29 个字符。
dSpeedRatio	轨迹运动速度比	float	0.01~1.00	轨迹运动速度比。计算 PathJ 时需要用到此参数。
dRadius	过渡半径	float	>0	过渡半径,单位[mm]。计算 PathJ 时需要用到此参数。
dVelocity	轨迹运动速度	float	-	轨迹运动速度, 单位[mm/s]。计算 PathL 时需要用到此参数。当 nRawDataType 取值为 1 时, 此参数必给; nRawDataType 取值为 0 时, 此参数不需要, 系统自动使用最大值计算 PathL。
dAcc	轨迹运动加速度	float	-	轨迹运动加速度, 单位[mm/s <sup>2</sup> ]。计算 PathL 时需要用到此参数。当 nRawDataType 取值为 1 时, 此参数必给; nRawDataType 取值为 0 时, 此参数不需要, 系统自动使用最大值计算 PathL。 <b>要求大于速度</b>
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> ]。计算 PathL 时需要用到此参数。当 nRawDataType 取值为 1 时, 此参数必给; nRawDataType 取值为 0 时, 此参数不需要, 系统自动使用最大值计算 PathL。

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”。计算 PathL 时需要用到此参数。当 nRawDataType 取值为 1 时，此参数必给；nRawDataType 取值为 0 时，此参数不需要，系统自动使用 “Base” 计算 PathL。
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”。计算 PathL 时需要用到此参数。当 nRawDataType 取值为 1 时，此参数必给；nRawDataType 取值为 0 时，此参数不需要，系统自动使用 “TCP” 计算 PathL。

\* 注：

此指令执行时，系统会先检查同名轨迹是否已经存在，如果存在，系统会先将其删除。

轨迹新建成功后，其状态为“示教中”。

本指令可以替换 HRIF\_StartPushMovePathJ 和 HRIF\_InitMovePathL。

✓ 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

### 3.11.16.2.示例

#### # 原始点位类型

```
nRawDataType = 1
```

#### # 轨迹名称

```
trajectName = "Path_01"
```

#### # 轨迹运动速度比

```
dSpeedRatio = 0.3
```

#### # 过渡半径

```
dRadius = 20
```

#### # 轨迹运动速度

---

```
dVelocity = 100
# 轨迹运动加速度

dAcc = 500
# 轨迹运动加加速度

dJerk = 10000
# 用户坐标名称

sUcsName = "Base"
# 工具坐标名称

sTcpName = "TCP"
# 初始化直线运动轨迹

nRet = cps.HRIF_InitPath(0,0,nRawDataType,trajectName,dSpeedRatio,dRadius,dVelocity,dAcc, dJerk, sUcsName,
sTcpName)
```

### 3.11.17. HRIF\_PushPathPoints

3.11.17.1.描述：向轨迹中批量推送原始点位（可多次调用）。

✓ 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
sPathName	轨迹名称	string	-	轨迹名称, 适用 MovePathJ 和 MovePathL
sPoints	点位数据	list	-	点位数据, 每个数据以逗号分隔。每个点位 6 个坐标值, 因此此字段值个数必须是 6 的倍数。系统自动计算出点位个数。

\* 注：

本指令可反复调用。

对于状态为示教完、计算完成、计算出错、未示教的轨迹，亦可使用本指令。

原始点位最多可以有 10000 个。如果将要推送的点位数量加上已有的点位数量超过 10000，则本批推送的点位全部不加入到点位队列。

使用本指令后，轨迹状态变为“示教中”。

本指令可以替换 HRIF\_PushMovePaths。

✓ 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

3.11.17.2.示例

# 轨迹名称

```
trajectName = "Path_01"
```

# 点位数据

```
sPoints = [420,0,445,180,0,180, 430,10,445,180,0,180, 440,50,445,180,0,180,
           520,100,445,180,0,180, 450,50,445,180,0,180, 430,200,445,180,0,180]
```

# 向轨迹中批量推送原始点位

---

```
nRet = cps.HRIF_PushPathPoints(0,0, trajectName,sPoints)
```

```
# 轨迹名称
```

```
trajectName = "drag_01"
```

```
# 点位数据
```

```
sPoints = [0,0,90,0,90,0, 10,20,90,0,90,0, 20,0,90,0,90,0,  
           40,0,90,0,90,0, 60,0,90,0,90,0, 60,20,90,0,90,0]
```

```
# 向轨迹中批量推送原始点位
```

```
nRet = cps.HRIF_PushPathPoints(0,0, trajectName,sPoints)
```

### 3.11.18. HRIF\_EndPushPathPoints

3.11.18.1.描述：结束向轨迹中推送点位，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
sPathName	轨迹名称	string	-	轨迹名称, 适用 MovePathJ 和 MovePathL

✓ 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

3.11.18.2.示例

# 轨迹名称

```
sPathName= "drag_01"
```

# 下发完成，开始计算轨迹

```
nRet = cps.HRIF_EndPushPathPoints(0,0, sPathName)
```

### 3.11.19. HRIF\_DelPath

3.11.19.1.描述：删除指定名称的轨迹

✓ 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
sPathName	轨迹名称	string	-	轨迹名称, 适用 MovePathJ 和 MovePathL

\* 注：

本指令可以替换 HRIF\_DelMovePathJ。

正在计算目标点位的轨迹，或者正在进行导入后处理的轨迹不能删除。

✓ 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

3.11.19.2.示例

# 轨迹名称

```
sPathName = "drag_03"  
# 删除轨迹  
nRet = cps.HRIF_DelPath(0,0, sPathName)
```

### 3.11.20. HRIF\_ReadPathList

3.11.20.1.描述：读取轨迹列表

✓ 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	轨迹列表	list	-	读取到的所有轨迹名称列表

✓ 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

3.11.20.2.示例

# 轨迹列表

result = [ ]

# 读取轨迹列表

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



### 3.11.21. HRIIF\_ReadPathInfo

3.11.21.1.描述：读取指定名称轨迹的信息

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned int	0~5	电箱 ID 号, 默认值=0
rbtID	机器人 ID	unsigned int	0~5	机器人 ID 号, 默认值=0
sPathName	轨迹名称	string	-	轨迹名称, 适用 MovePathJ 和 MovePathL
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]	原始点位类型	string	-	原始点位类型 0: 关节点位 1: 空间点位
result[1]	MovePathJ 的状态	string	-	8 种状态: 未示教(0)、示教中(1)、示教完(4)、计算中(2)、计算出错(5)、计算完成(3)、已导入(9)、导入后处理(10)
result[2]	错误码	string	-	MovePathJ 的错误码
result[3]	MovePathL 的状态	string	-	8 种状态: 未示教(0)、示教中(1)、示教完(4)、计算中(2)、计算出错(5)、计算完成(3)、已导入(9)、导入后处理(10)
result[4]	错误码	string	-	MovePathL 的错误码
result[5]	轨迹运动速度比	string	-	轨迹运动速度比
result[6]	过渡半径	string	-	过渡半径, 单位: mm
result[7]	轨迹运动速度	string	-	轨迹运动速度, 单位: mm/s
result[8]	轨迹运动加速度	string	-	轨迹运动加速度, 单位: mm/s^2
result[9]	轨迹运动加加速度	string	-	轨迹运动加加速度, 单位: mm/s^3
result[10]	用户坐标	string	-	dUcs_X-dUcs_Rz, 用户坐标
result[11]	工具坐标	string	-	dTcp_X-dTcp_Rz, 工具坐标
result[12]	原始点位个数	string	-	原始点位个数

result[13]	第一个原始点位坐标	string	-	第一个原始点位坐标
------------	-----------	--------	---	-----------

✓ 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

### 3.11.21.2.示例

```
# 轨迹名称
sPathName = "Path_01"

# 轨迹信息
result = [ ]
# 读取指定名称轨迹的信息
nRet = cps.HRIF_ReadPathInfo(0,0, sPathName, result)
```

### 3.11.22. HRIF\_UpdatePathName

3.11.22.1.描述：更新轨迹名称。

✓ 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
sPathName	轨迹原名称	string	-	轨迹原名称, 适用 MovePathJ 和 MovePathL
sPathNewName	新轨迹名称	string	-	新轨迹名称, 适用 MovePathJ 和 MovePathL

✓ 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

3.11.22.2.示例

# 轨迹原名称

```
sPathName = "drag_01"
```

# 新轨迹名称

```
sPathNewName = "drag_02"
```

# 更新轨迹名称

```
nRet = cps.HRIF_UpdatePathName(0,0, sPathName, sPathNewName)
```

### 3.11.23. HRIIF\_ReadPathState

3.11.23.1.描述：读取当前轨迹状态。

✓ 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
sPathName	轨迹名称	string	-	轨迹名称, 适用 MovePathJ 和 MovePathL
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]	MovePathJ 的状态	string	0/1/2/3/4/5/9/10	0: 轨迹尚未示教, 示教中的轨迹结束示教后, 如果没有原始点位, 则进入到本状态; 1: 轨迹正在示教中, InitPath、PushPathPoints 后进入到本状态; 2: 轨迹正在目标点位计算中, 系统检测到状态为示教结束后, 自动进入到本状态, 本状态持续时长视轨迹线路长短和速度比而定, 可能会长达 10 分钟以上。轨迹处于本状态时, 不能删除。 3: 轨迹目标点位计算完成, 系统计算目标轨迹成功完成后自动进入到本状态。 4: 轨迹结束示教, EndPushPathPoints 后, 进入到本状态, 轨迹文件导入后处理结束后也会进入到本状态; 5: 轨迹目标点位计算错误, 系统计算目标轨迹出错后自动进入到本状态; 9: 轨迹已导入, 前端导入某个轨迹后进入

				到本状态; 10: 轨迹导入后处理中, 系统检测到状态为已导入后, 自动进入到本状态。轨迹处于本状态时, 不能删除。
result[1]	MovePathJ 的错误码	string	-	参见错误码总表。当 StateJ 为 5 时, 此参数有意义。
result[2]	MovePathL 的状态	string	0/1/2/3/4/5/9/10	同 StateJ。其中, 状态 2 的持续时长视轨迹线路长短和速度比而定, 可能会长达数十秒。
result[3]	MovePathL 的错误码	string	-	参见错误码总表。当 StateL 为 5 时, 此参数有意义。

✓ **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

### 3.11.23.2.示例

```
# 轨迹名称
sPathName = "drag_01"

# MovePathJ 的状态
nStateJ = 0

# MovePathJ 的错误码
nErrorCodeJ = 0

# MovePathL 的状态
nStateL = 0

# MovePathL 的错误码
nErrorCodeL = 0

# 更新轨迹名称
nRet = cps.HRIF_ReadPathState(0,0, sPathName,nStateJ,nErrorCodeJ,nStateL,nErrorCodeL)
```

## 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)

### 3.12.4. HRIF\_SpeedJ

3.12.4.1. 描述：在线关节运动速度伺服控制，以该指令中指定的各个关节的速度和加速度运动指定的时长。

✓ 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
cmdVel	命令关节速度	list[float]	-	更新的各关节命令关节速度, 单位: °/s
acc	加速度	float	-	关节设定加速度, 单位: °/s^2
runtime	运行时间	float	0.001~86400	运动最长执行时长, 指令运行超过该时间, 运动将会停止, 单位: s

\* 注:

1. 关节速度随不同机型取值范围有所不同
2. 本指令启动的运动可以用 GrpStop 指令提前终止, 可以被 GrpInterrupt/GrpContinue 指令暂停/继续。
3. 如果当前指令的最长执行时长还未到, 又接收到新的成功执行的 SpeedJ 指令, 则前一个 SpeedJ 指令的运动会提前终止, 并按新的 SpeedJ 指令的参数继续运动。
4. 该指令触发的运动结束前, 执行其它路点运动指令会返回失败, 执行 SpeedL 指令机器人会进入错误状态。
5. 该指令触发的运动速度不受 SetOverride 指令影响。

✓ 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

3.12.4.2. 示例

```
# 命令关节速度
```

```
cmdVel = [5,5,5,2,2,2]
```

```
# 定义关节加速度
```

```
acc = 20
```

```
# 定义运行时间
```

---

```
runtime = 2
```

```
# 下发关节命令速度指令
```

```
nRet = cps.HRIF_SpeedJ(0,0,cmdVel, acc, runtime)
```

### 3.12.5. HRIF\_SpeedL

3.12.5.1. 描述：在线空间运动速度伺服控制，以该指令中指定的位姿各个坐标的速度和加速度运动指定的时长。

✓ 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
cmdVel	坐标命令速度	list[float]	-	更新的坐标命令速度 X, 单位: mm/s Y, 单位: mm/s Z, 单位: mm/s RX, 单位: °/s RY, 单位: °/s RZ, 单位: °/s
LinearAcc	X/Y/Z 加速度	float	-	X/Y/Z 加速度, 单位: mm/s^2
AngularAcc	RX/RY/RZ 加速度	float	-	RX/RY/RZ 加速度, 单位: °/s^2
runtime	运行时间	float	0.001~86400	运动最长执行时长, 指令运行超过该时间, 运动将会停止, 单位: s

\* 注:

1. 速度随不同机型取值范围有所不同
2. 本指令启动的运动可以用 GrpStop 指令提前终止, 可以被 GrpInterrupt/GrpContinue 指令暂停/继续。
3. 如果当前指令的最长执行时长还未到, 又接收到新的成功执行的 SpeedL 指令, 则前一个 SpeedL 指令的运动会提前终止, 并按新的 SpeedL 指令的参数继续运动。
4. 该指令触发的运动结束前, 执行其它路点运动指令会返回失败, 执行 SpeedJ 指令机器人会进入错误状态。
5. 该指令触发的运动速度不受 SetOverride 指令影响。

✓ 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
------	-----	-----	---------	--

## 3.12.5.2. 示例

```
# 命令空间速度
cmdVel = [-1,1,1,0,1,2]
# 定义X/Y/Z 加速度
LinearAcc = 100
# 定义RX/RY/RZ 加速度
AngularAcc = 200
# 定义运行时间
runtime = 2
# 下发空间命令速度指令
nRet = cps.HRIF_SpeedL(0,0,cmdVel, LinearAcc , AngularAcc, runtime)
```

### 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.位置跟随运动类控制指令

#### 3.14.1. HRIF\_SetPoseTrackingMaxMotionLimit

3.14.1.1. 描述：设置位置跟随的最大跟随速度

✓ 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
dMaxLineVel	直线最大速度	float	>0	位置跟随直线最大跟随速度, 单位[mm/s]
dMaxOriVel	姿态最大速度	float	>0	位置跟随姿态最大跟随速度, 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

3.14.1.2. 示例

```
# 设置直线最大速度
dMaxLineVel = 100

# 设置姿态最大速度
dMaxOriVel = 1

# 发送信息
nRet = cps.HRIF_SetPoseTrackingMaxMotionLimit(0,0,dMaxLineVel,dMaxOriVel)
```

### 3.14.2. HRIF\_SetPoseTrackingStopTimeOut

3.14.2.1. 描述：设置位置跟踪超时停止时间。

✓ 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
dTime	超时时间	float	>=0	位置跟随超时停止时间, 单位: 秒[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

3.14.2.2. 示例

# 超时时间

dTime = 10

# 设置位置跟踪超时停止时间

nRet = cps.HRIF\_SetPoseTrackingStopTimeOut(0,0,dTime)

### 3.14.3. HRIF\_SetPoseTrackingPIDParams

#### 3.14.3.1. 描述：设置 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
dPosPID1	位置跟随 PID	float	-	位置跟随 PID, 默认=5
dPosPID2	位置跟随 PID	float	-	位置跟随 PID, 默认=0.1
dPosPID3	位置跟随 PID	float	-	位置跟随 PID, 默认=0
dOriPID1	姿态跟随 PID	float	-	姿态跟随 PID, 默认=5
dOriPID2	姿态跟随 PID	float	-	姿态跟随 PID, 默认=0.1
dOriPID3	姿态跟随 PID	float	-	姿态跟随 PID, 默认=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

#### 3.14.3.2. 示例

```
# 设置位置跟随 PID 和姿态跟随 PID
```

```
dPosPID1 = 5
dPosPID2 = 0.1
dPosPID3 = 0
dOriPID1 = 5
dOriPID2 = 0.1
dOriPID3 = 0
```

```
# 发送信息
```

```
nRet = cps.HRIF_SetPoseTrackingPIDParams(0,0,dPosPID1,dPosPID2,dPosPID3,
                                         dOriPID1,dOriPID2,dOriPID3)
```

### 3.14.4. HRIF\_SetPoseTrackingTargetPos

#### 3.14.4.1. 描述：设置位置跟随的目标位置

✓ 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
dX	X 方向保持的距离	float	-	X 方向保持的距离,单位[mm]
dY	Y 方向保持的距离	float	-	Y 方向保持的距离,单位[mm]
dZ	Z 方向保持的距离	float	-	Z 方向保持的距离,单位[mm]
dRx	Rx 方向保持的距离	float	-	Rx 方向保持的距离,单位[°]
dRy	Ry 方向保持的距离	float	-	Ry 方向保持的距离,单位[°]
dRz	Rz 方向保持的距离	float	-	Rz 方向保持的距离,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

#### 3.14.4.2. 示例

# 设置目标位置

dX = 0 dY = 0 dZ = 100 dRx = 0 dRy = 0 dRz = 0

# 发送信息

nRet = cps.HRIF\_SetPoseTrackingTargetPos(0,0,dX,dY,dZ,dRx,dRy,dRz)

### 3.14.5. HRIF\_SetPoseTrackingState

#### 3.14.5.1. 描述：设置位置跟随状态

✓ 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	位置跟随状态	int	0/1	0: 关闭位置跟随 1: 开启位置跟随

✓ 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

#### 3.14.5.2. 示例

```
# 开启位置跟随
```

```
nState = 1
```

```
# 发送信息
```

```
nRet = cps.HRIF_SetPoseTrackingState(0,0,nState)
```

### 3.14.6. HRIF\_SetUpdateTrackingPose

3.14.6.1. 描述：设置实时更新传感器位置信息

✓ 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
dX	检测到的 X 方向的距离	float	-	检测到的 X 方向的距离,单位[mm]
dY	检测到的 Y 方向的距离	float	-	检测到的 Y 方向的距离,单位[mm]
dZ	检测到的 Z 方向的距离	float	-	检测到的 Z 方向的距离,单位[mm]
dRx	检测到的 Rx 方向的距离	float	-	检测到的 Rx 方向的距离,单位[°]
dRy	检测到的 Ry 方向的距离	float	-	检测到的 Ry 方向的距离,单位[°]
dRz	检测到的 Rz 方向的距离	float	-	检测到的 Rz 方向的距离,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

3.14.6.2. 示例

# 设置检测到的位置信息

dX = 0 dY = 0 dZ = 100 dRx = 0 dRy = 0 dRz = 0

# 发送信息

nRet = cps.HRIF\_SetUpdateTrackingPose(0,0,dX,dY,dZ,dRx,dRy,dRz)

### 3.15.其他指令

#### 3.15.1. HRIF\_HRApp

3.15.1.1. 描述：执行插件 App 命令。(推荐使用 HRIF\_HRAppCmd)

✓ 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
sAppName	App 名称	string	-	App 名称
sParams	参数列表	list	-	命令名称与参数列表
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

3.15.1.2. 示例

```
# 定义返回值空列表
result = []

# 插件名称
sAppName = 'FT_Plugin'

# 插件指令与参数
sParams = ['FT_reset']

# 发送插件命令
nRet = cps.HRIF_HRApp(0,0,sAppName, sParams, result)
```

### 3.15.2. HRIF\_HRAppCmd

3.15.2.1. 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
sAppName	App 名称	string	-	App 名称
sParams	参数列表	list	-	命令名称与参数列表
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

3.15.2.2. 示例

# 定义返回值空列表

```
result = [ ]
```

# 插件名称

```
sAppName = 'FT_Plugin'
```

# 插件指令与参数

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

# 发送插件命令

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

### 3.15.3. HRIF\_WriteEndHoldingRegisters

3.15.3.1. 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	1~255	从站 ID
nFunction	功能码	int	>0	功能码: 0x01-读线圈寄存器 0x02-读线离散输入寄存器 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

3.15.3.2. 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)
```



### 3.15.4. HRIF\_ReadEndHoldingRegisters

3.15.4.1. 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	1~255	从站 ID
nFunction	功能码	int	>0	功能码: 0x01-读线圈寄存器 0x02-读线离散输入寄存器 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

## 3.15.4.2. 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, 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++》](#)。