HansRobot Library

C#

(Elfin 系列机器人通用)

修订记录

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

A - 增加 M - 修订 D - 删除

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

第一章 概述

1.1. 编写目的

1. 为使用 Hans C# 接口的人员提供接口使用说明；
2. 为研发人员进行接口编写和维护提供参考文件。

1.2. 使用须知

文档中所有代码示例默认在机器人的工作空间内没有任何干涉。

1. SDK 使用：

   将 libHR_Pro.dll 和 HR_Pro.h 拷贝到对应的工程目录文件夹下。

2. SDK 的正常使用需要包含动态库文件，动态库版本号查询：

   右键 libHR_Pro.dll 文件，选择属性，在“详细信息”中查看版本信息。

1.3. 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

1.4. 数据类型

数据类型	长度	说明
char	1 byte	字符型
bool	1 byte	布尔型

int	4 bytes	无符号整型
int	4 bytes	整型
double	8 bytes	双精度浮点型
string	-	字符串
List	-	列表

第二章 环境搭建

2.1. 创建项目

打开 Visual Stdio 2019，创建 C# 控制台应用程序：

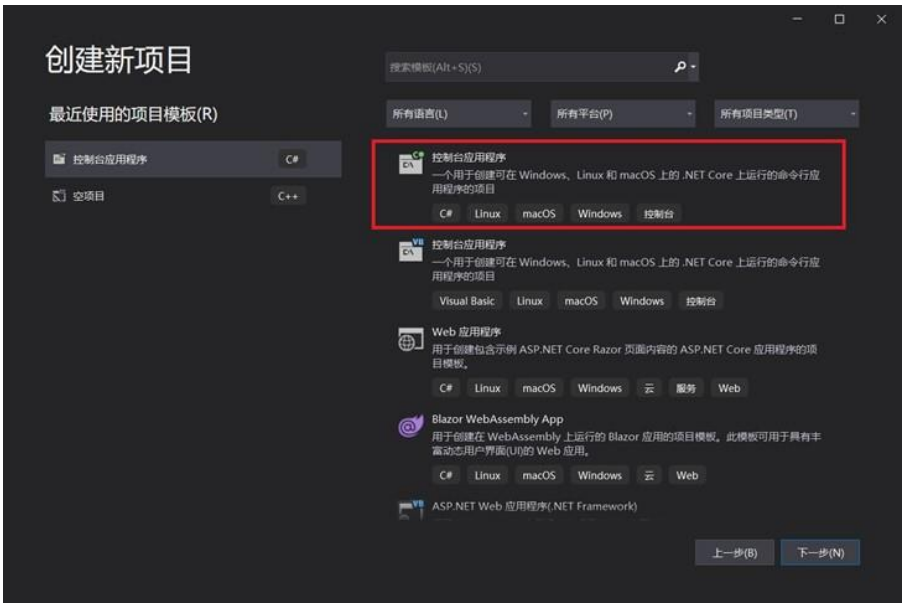

图 2-1：创建项目

2.2. 添加引用

在解决方案下依赖项中右击添加项目引用。找到 HansRobotLibrarys.dll 文件和 Newtonsoft.Json.dll 文件并导入。

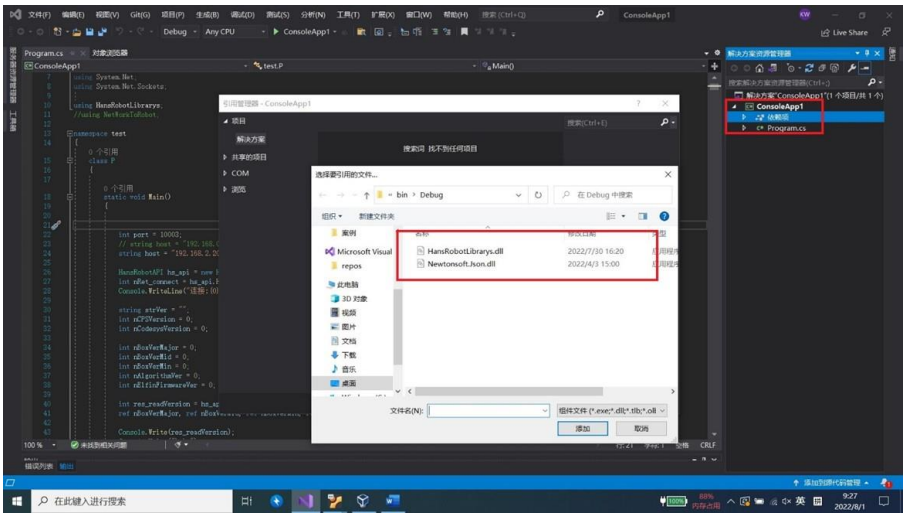

图 2-2：添加引用

2.3. 添加库文件

导入文件，确保文件导入到依赖项下属程序集中。

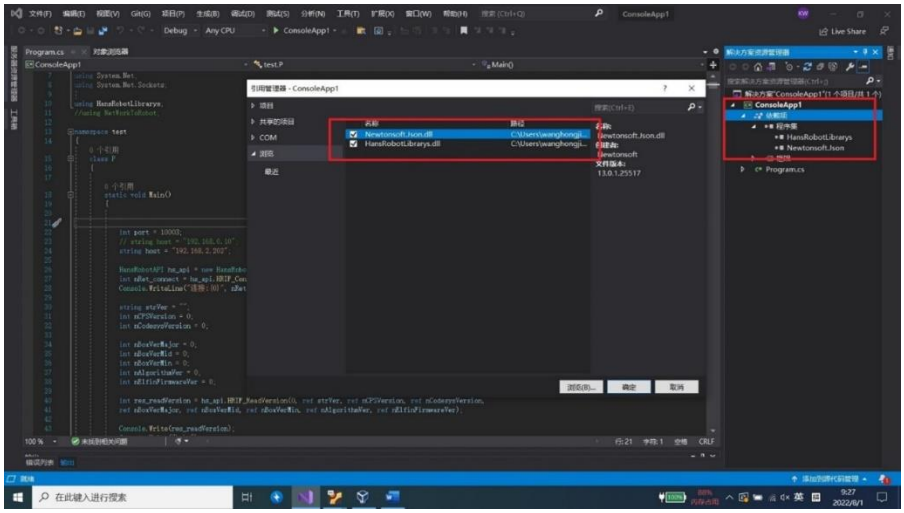

图 2-3：添加库文件

2.4. 初始化

在新建项目中添加 using HansRobotLibrarys; 通过 HansRobotAPI hs_api = new HansRobotApi(); 初始化对象。通过 hs_api （名字可自定义）调用接口。

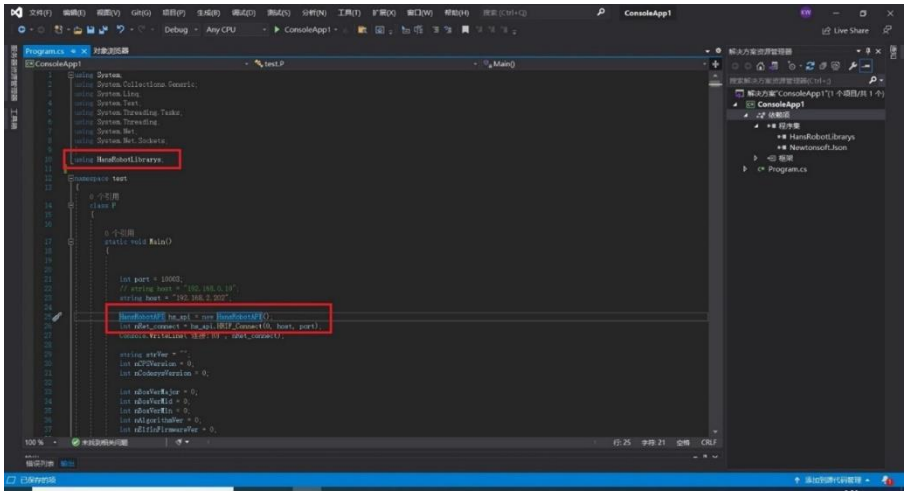

图 2-4：初始化

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	unsigned int	0~5	电箱 ID 号，默认值= 0
hostName	控制器 IP 地址	char*	-	控制器 IP 地址， 根据实际设置的 IP 地址定义
nPort	端口号	unsigned short	10003	控制器端口号，默认值= 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: 返回函数调用成功 nRet>0: 返回调用失败的错误码

3.1.1.2. 示例

```
// 定义控制器 IP 地址
string hostname = "192.168.0.10";
// 定义控制器端口号
int nPort = 10003;
// 连接机器人服务器
int nRet= hr_api.HRIF_Connect(0, hostname, 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	unsigned 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
int nRet= hr_api.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	unsigned 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
bool bRet = hr_api.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	unsigned int	0~5	电箱 ID 号，默认值= 0
nCmdType	断电类型	int	1~2	1：停止运动-去使能-断 48V 供电 2：停止运动-去使能-断 48V 供电-重启操作系统

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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

```
//断电类型
int nCmdType = 1;
// 控制器断电
int nRet = hr_api.HRIF_ShutdownRobot(0,nCmdType);
```

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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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
int nRet= hr_api.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	unsigned 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
int nRet= hr_api.HRIF_BlackOut(0);
```

3.1.8. HRIF_Connect2Controller

3.1.8.1. 描述：连接控制器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
int nRet= hr_api.HRIF_Connect2Controller(0);
```

3.1.9. HRIF_IsSimulateRobot

3.1.9.1. 描述：是否为模拟机器人。

✓ 输入变量

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

✓ 输出变量

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

✓ 返回值

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

3.1.9.2. 示例

```
// 定义是否为模拟机器人
int nSimulateRobot = 0;
// 判断是否为模拟机器人
int nRet= hr_api.HRIF_IsSimulateRobot(0, ref nSimulateRobot);
```

3.1.10. HRIF_IsControllerStarted

3.1.10.1. 描述：控制器是否启动完成。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nStarted	是否启动完成	int	0/1	0: 未启动 1: 已启动

✓ 返回值

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

3.1.10.2. 示例

```
// 定义是否启动完成变量
int nStarted = 0;
// 判断机器人是否启动完成
int nRet= hr_api.HRIF_IsControllerStarted(0, ref nStarted);
```

3.1.11. HRIF_ReadVersion

3.1.11.1. 描述：读取控制器版本号。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
strVer	整体版本号	string	-	版本号的格式以“.”分隔，数字的意义如下： CPSVer.controlVer BoxVerMajor BoxVerM ID. BoxVerMin AlgorithmVer ElfinFirmwareVer
nCPSVersion	CPS 版本	int	-	CPS 版本号
nCodesysVersion	控制器版本	int	-	控制器版本号
nBoxVerMajor	电箱版本	int	0~4	电箱版本号: 0：模拟电箱 1~4：电箱版本号
nBoxVerM ID	控制板固件版本	int	-	控制板固件版本
nBoxVerMin	控制板固件版本	int	-	控制板固件版本
nAlgorithmVer	算法版本	int	-	算法版本
nElfinFirmwareVer	固件版本	int	-	固件版本

✓ 返回值

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

3.1.11.2. 示例

```
// 获取整体版本号
```

```
string strVer = " ";
// 获取 CPS 版本号
int nCPSVersion = 0;
// 获取控制器版本号
int nCodesysVersion = 0;
// 获取电箱版本
int nBoxVerMajor = 0;
// 获取电箱固件版本
int nBoxVerMid = 0;
// 获取电箱固件版本
int nBoxVerMin = 0;
// 获取算法版本
int nAlgorithmVer = 0;
// 获取驱动固件版本
int nElfinFirmwareVer = 0;
// 获取版本号
int nRet= hr_api.HRIF_ReadVersion(0,0, ref strVer, ref nCPSVersion, ref nCodesysVersion, ref nBoxVerMajor, ref nBoxVerMid, ref nBoxVerMin, ref nAlgorithmVer, ref nElfinFirmwareVer);
```

3.1.12. HRIF_ReadRobotModel

3.1.12.1. 描述：读取机器人类型。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
strModel	机器人类型	string	-	机器人类型

✓ 返回值

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

3.1.12.2. 示例

```
// 获取机器人类型
string strModel = " ";
// 读取机器人类型
int nRet= hr_api.HRIF_ReadRobotModel (0, ref strModel);
```

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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api. HRIF_GrpOpenFreeDriver(0,0);
```

3.3. 脚本控制指令

3.3.1. HRIF_RunFunc

3.3.1.1. 描述：运行指定脚本函数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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	vector<string>	-	参数列表

✓ 返回值

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

3.3.1.2. 示例

```
// 指定运行函数 Func_1
string strFuncName = "Func_1";
// 定义输入参数
List<string> param = new List<string>();
// 运行函数 Func_1
int nRet= hr_api.HRIF_RunFunc(0, strFuncName, param);
```

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	unsigned 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
int nRet= hr_api.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	unsigned 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
int nRet= hr_api.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	unsigned 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
int nRet= hr_api.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。

✓ 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.HRIF_ContinueScript(0);
```

3.4. 电箱控制指令

3.4.1. HRIF_ReadBoxInfo

3.4.1.1. 描述：读取电箱信息。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nConnected	电箱连接状态	int	0/1	0：未连接 1：已连接
n48V_ON	48V 电压状态	int	0/1	0：电压<36V 1：电压>36V
db48OUT_Voltag	48V 输出电压值	double	0~53	48V 输出电压值
db48OUT_Current	48V 输出电流值	double	-	48V 输出电流值
nRemoteBTN	远程关机按钮状态	int	0/1	0：无信号 1：有信号
nThreeStageBTN	三段按钮状态	int	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

```
// 定义电箱是否连接
int nConnected = 0;
// 48V 上电状态
```

```
int n48V_ON = 0;
// 48V 电压值
double db48OUT_Voltag = 0;
// 48V 电流值
double db48OUT_Current = 0;
// 远程按钮状态
int nRemoteBTN = 0;
// 三段按钮状态
int nThreeStageBTN = 0;
// 读取电箱信息
int nRet= hr_api.HRIF_ReadBoxInfo(0,0, ref nConnected, ref n48V_ON, ref db48OUT_Voltag,
ref db48OUT_Current, ref nRemoteBTN, ref nThreeStageBTN);
```

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	unsigned int	0~5	电箱 ID 号，默认值= 0
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
nVal	控制数字输入对应位状态	int	0/1	0：低电平 1：高电平

✓ 返回值

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

3.4.2.2. 示例

```
// 读取结果
int nVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取电箱 nBit 位 CI 状态
int nRet= hr_api.HRIF_ReadBoxCI (0, nBit, ref nVal);
```

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	unsigned 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
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
nVal	通用数字输入对应位状态	int	0/1	0：低电平 1：高电平

✓ 返回值

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

3.4.3.2. 示例

```
// 读取结果
int nVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取电箱 nBit 位 DI 状态
int nRet= hr_api.HRIF_ReadBoxDI (0,0, nBit, ref nVal);
```

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	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.4.4.2. 示例

```
// 读取结果
int nVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取电箱 nBit 位 CO 状态
int nRet= hr_api.HRIF_ReadBoxCO (0, nBit, ref nVal);
```

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	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.4.5.2. 示例

```
// 读取结果
int nVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取电箱 nBit 位 DO 状态
int nRet= hr_api.HRIF_ReadBoxDO (0, nBit, ref nVal);
```

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	unsigned int	0~5	电箱 ID 号，默认值= 0
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
dVal	对应的模拟量通道值	double	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

```
// 读取结果
double dVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取电箱 nBit 通道 AI 值
int nRet= hr_api.HRIF_ReadBoxAI(0, nBit, ref dVal);
```

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	unsigned int	0~5	电箱 ID 号，默认值= 0
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
nMode	对应的模拟量通道模式	int	0/1	0：电压模式 1：电流模式:
dVal	对应的模拟量通道值	double	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

```
// 读取结果
double dVal = 0;
// 读取的当前模式
int nMode = 0;
// 定义需要读取的位
int nBit = 0;
// 读取电箱 nBit 通道 AO 值
int nRet= hr_api.HRIF_ReadBoxAO(0,0, nBit, ref nMode, ref dVal);
```

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	unsigned 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
int nVal = 0;
// 定义需要设置的位
int nBit = 0;
// 设置电箱第 nBit 位 CO 状态为 nVal
int nRet= hr_api.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	unsigned 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
int nVal = 0;
// 定义需要设置的位
int nBit = 0;
// 设置电箱第 nBit 位 DO 状态为 nVal
int nRet= hr_api.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	unsigned int	0~5	电箱 ID 号，默认值= 0
nBit	模拟量通道	int	0/1	模拟量通道
nVal	模式	int	1/2	1：电压模式 2：电流模式

✓ 返回值

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

3.4.10.2. 示例

```
// 定义设置值
int nVal = 1;
// 定义需要设置的通道
int nBit = 1;
// 设置电箱第 nBit 通道 AO 模式为 nVal
int nRet= hr_api.HRIF_SetBoxAOMode (0, nBit , nVal);
```

3.4.11. HRIF_SetBoxAOVal

3.4.11.1. 描述：设置电箱模拟量输出值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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	double	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
double dVal = 0;
// 定义需要设置的通道
int nBit = 0;
//设置模式
int nMode = 1;
// 设置电箱第 nBit 通道 AO 值为 dVal
int nRet= hr_api.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	unsigned 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
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
int nVal = 0;
// 定义需要设置的位
int nBit = 0;
// 设置末端 nBit 位 DO 状态
int nRet= hr_api.HRIF_SetEndDO(0,0, nBit, nVal);
```

3.4.13. HRIF_ReadEndDI

3.4.13.1. 描述：读取末端数字输入状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nBit	末端数字输入位	int	0~2	末端数字输入对应位状态

✓ 输出变量

输入变量	名称	数据类型	有效范围	内容
nVal	末端数字输入对应位状态	int	0/1	0：低电平 1：高电平

✓ 返回值：

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

3.4.13.2. 示例

```
// 读取结果
int nVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取末端 nBit 位 DI 状态
int nRet= hr_api.HRIF_ReadEndDI(0,0, nBit, ref nVal);
```

3.4.14. HRIF_ReadEndDO

3.4.14.1. 描述：读取末端数字输出状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nBit	末端数字输出位	int	0~2	末端数字输出对应位索引
nVal	末端数字输出对应位状态	int	0/1	0：低电平 1：高电平

✓ 输出变量

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

✓ 返回值：

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

3.4.14.2. 示例

```
// 读取结果
int nVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取末端 nBit 位 DO 状态
int nRet= hr_api.HRIF_ReadEndDO(0,0, nBit, ref nVal);
```

3.4.15. HRIF_ReadEndAI

3.4.15.1. 描述：读取末端模拟量输入值。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
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
dVal	对应的模拟量通道值	double	0~20	电流模式:4~20mA 电压模式:0-10V

✓ 返回值

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

3.4.15.2. 示例

```
// 读取结果
double dVal = 0;
// 定义需要读取的位
int nBit = 0;
// 读取末端 nBit 通道 AI 值
int nRet= hr_api.HRIF_ReadEndAI(0,0, nBit, ref dVal);
```

3.4.16. HRIF_ReadEndBTN

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nBit1	按键 1 状态	int	0/1	0：松开 1：按下
nBit2	按键 2 状态	int	0/1	0：松开 1：按下
nBit3	按键 3 状态	int	0/1	0：松开 1：按下
nBit4	按键 4 状态	int	0/1	0：松开 1：按下

✓ 返回值

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

3.4.16.2. 示例

```
// 定义需要读取的位
int nBit1 = 0; int nBit2 = 0;
int nBit3 = 0; int nBit4 = 0;
// 读取末端 nBit 通道 AI 值
int nRet= hr_api.HRIF_ReadEndBTN(0,0, ref nBit1, ref nBit2, ref nBit3, ref nBit4);
```

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	unsigned 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
dOverride	速度比	double	0.01~1	需要设置的速度比(0,0.01~1)

✓ 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
double dOverride = 0.5;
// 设置当前速度比
int nRet= hr_api.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	unsigned 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
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
int nState = 1;
// 设置 Tool 运动状态
int nRet= hr_api.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	unsigned 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
dMass	质量	double	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	double	>0	质心 X 方向偏移，单位[mm]
dY	质心 Y 方向偏移	double	>0	质心 Y 方向偏移，单位[mm]
dZ	质心 Z 方向偏移	double	>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
double dMass = 0.5;
// 需要设置负载质量
double dX= 0.5;
// 需要设置负载质量
double dY= 0.5;
// 需要设置负载质量
double dZ= 0.5;
// 设置当前负载
int nRet= hr_api.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	unsigned 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
dJ1	J1 轴最大速度	double	0~200	J1 轴最大速度，单位[°/s]
dJ2	J2 轴最大速度	double	0~200	J2 轴最大速度，单位[°/s]
dJ3	J3 轴最大速度	double	0~200	J3 轴最大速度，单位[°/s]
dJ4	J4 轴最大速度	double	0~200	J4 轴最大速度，单位[°/s]
dJ5	J5 轴最大速度	double	0~200	J5 轴最大速度，单位[°/s]
dJ6	J6 轴最大速度	double	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

* 注：关于错误码请参阅 “HansRobot_ErrorCode.docx”。

3.5.4.2. 示例

```
// 定义需要设置关节最大速度
double dJ1 = 180; double dJ2 = 180; double dJ3 = 180;
double dJ4 = 180; double dJ5 = 180; double dJ6 = 180;
// 设置最大关节速度
int nRet= hr_api.HRIF_SetJointMaxVel(0,0, dJ1, dJ2, dJ3, dJ4, dJ5, dJ6);
```

3.5.5. HRIF_SetJointMaxAcc

3.5.5.1. 描述：设置关节最大运动加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dJ1	J1 轴最大加速度	double	0~720	J1 轴最大加速度，单位[°/s²]
dJ2	J2 轴最大加速度	double	0~720	J2 轴最大加速度，单位[°/s²]
dJ3	J3 轴最大加速度	double	0~720	J3 轴最大加速度，单位[°/s²]
dJ4	J4 轴最大加速度	double	0~720	J4 轴最大加速度，单位[°/s²]
dJ5	J5 轴最大加速度	double	0~720	J5 轴最大加速度，单位[°/s²]
dJ6	J6 轴最大加速度	double	0~720	J6 轴最大加速度，单位[°/s²]

* 注：关节加速度有效范围需要参考具体机型。

✓ 返回值

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

3.5.5.2. 示例

```
// 定义需要设置关节最大加速度
double dJ1 = 360; double dJ2 = 360; double dJ3 = 360;
double dJ4 = 360; double dJ5 = 360; double dJ6 = 360;
// 设置最大关节加速度
int nRet= hr_api.HRIF_SetJointMaxAcc(0,0, dJ1, dJ2, dJ3, dJ4, dJ5, dJ6);
```

3.5.6. HRIF_SetLinearMaxVel

3.5.6.1. 描述：设置直线运动最大速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dMaxVel	最大直线速度	double	0~3000	最大直线速度，默认[2000]，单位[mm/s]

* 注：直线速度有效范围需要参考具体机型。

✓ 返回值

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

3.5.6.2. 示例

```
// 定义需要设置直线最大速度
double dMaxVel = 100;
// 设置最大直线速度
int nRet= hr_api.HRIF_SetLinearMaxVel(0,0, dMaxVel);
```

3.5.7. HRIF_SetLinearMaxAcc

3.5.7.1. 描述：设置直线运动最大加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dMaxAcc	最大直线加速度	double	0~2500	最大直线加速度，默认[2500]，单位[mm/s^2]

* 注：直线加速度有效范围需要参考具体机型。

✓ 返回值

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

3.5.7.2. 示例

```
// 定义需要设置直线最大速度
double dMaxAcc = 2500;
// 设置最大直线速度
int nRet= hr_api.HRIF_SetLinearMaxAcc(0,0, dMaxAcc);
```

3.5.8. HRIF_SetMaxAcsRange

3.5.8.1. 描述：设置关节最大运动范围。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dMaxJ1-dMaxJ6	关节最大范围	double	-	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	double	-	dMinJ1:关节 1 最小运动范围，单位[°] dMinJ2:关节 2 最小运动范围，单位[°] dMinJ3:关节 3 最小运动范围，单位[°] dMinJ4:关节 4 最小运动范围，单位[°] dMinJ5:关节 5 最小运动范围，单位[°] dMinJ6:关节 6 最小运动范围，单位[°]

✓ 返回值

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

3.5.8.2. 示例

```
// 定义需要设置的关节最大运动范围
double dMaxJ1 = 360; double dMaxJ2 = 360; double dMaxJ3 = 360;
double dMaxJ4 = 360; double dMaxJ5 = 360; double dMaxJ6 = 360;
// 定义需要设置的关节最小运动范围
double dMinJ1 = -360; double dMinJ2 = -360; double dMinJ3 = -360;
double dMinJ4 = -360; double dMinJ5 = -360; double dMinJ6 = -360;
```

```
// 设置关节运动范围
int nRet= hr_api.HRIF_SetMaxAcsRange (0,0, dMaxJ1, dMaxJ2, dMaxJ3, dMaxJ4, dMaxJ5, dMaxJ6,
dMinJ1, dMinJ2, dMinJ3, dMinJ4, dMinJ5, dMinJ6);
```

3.5.9. HRIF_SetMaxPcsRange

3.5.9.1. 描述：设置空间最大运动范围。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dMaxX-dMaxRz	X-Rz 最大范围	double	-	XYZ 最大范围，单位[mm] RxRyRz 最大范围，单位[°]，未启用，默认 为 0
dMinX- dMinRz	X-Rz 最小范围	double	-	XYZ 最小范围，单位[mm] RxRyRz 最小范围，单位[°]，未启用，默认 为 0
dUcs_X-dUcs_Rz	基于用户坐标	double	-	dX：X 坐标，单位[mm] dY：Y 坐标，单位[mm] dZ：Z 坐标，单位[mm] dRx：Rx 坐标，单位[°] dRy：Ry 坐标，单位[°] dRz：Rz 坐标，单位[°]

✓ 返回值

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

3.5.9.2. 示例

```
// 定义需要设置的空间最大运动范围
double dMaxX = 1200; double dMaxY = 1200; double dMaxZ = 1200;
double dMaxRx = 180; double dMaxRy = 180; double dMaxRz = 180;
// 定义需要设置的空间最小运动范围
double dMinX = -1200; double dMinY = -1200; double dMinZ = -1200;
double dMinRx = -180; double dMinRy = -180; double dMinRz = -180;
// 定义用户坐标变量
```

```
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 设置关节运动范围
int nRet= hr_api.HRIF_SetMaxPcsRange(0,0, dMaxX, dMaxY, dMaxZ, dMaxRx,dMaxRy,dMaxRz,dMinX, dMinY, dMinZ,dMinRx,dMinRy,dMinRz, dUcs_X, dUcs_Y, dUcs_Z, dUcs_Rx, dUcs_Ry, dUcs_Rz);
```

3.5.10. HRIF_ReadJointMaxVel

3.5.10.1. 描述：读取关节最大运动速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dJ1	J1 轴最大速度	double	0~200	J1 轴最大速度，单位[°/s]
dJ2	J2 轴最大速度	double	0~200	J2 轴最大速度，单位[°/s]
dJ3	J3 轴最大速度	double	0~200	J3 轴最大速度，单位[°/s]
dJ4	J4 轴最大速度	double	0~200	J4 轴最大速度，单位[°/s]
dJ5	J5 轴最大速度	double	0~200	J5 轴最大速度，单位[°/s]
dJ6	J6 轴最大速度	double	0~200	J6 轴最大速度，单位[°/s]

✓ 返回值

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

3.5.10.2. 示例

```
// 定义关节最大运动速度变量
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取最大关节速度
int nRet= hr_api.HRIF_ReadJointMaxVel(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.5.11. HRIF_ReadJointMaxAcc

3.5.11.1. 描述：读取关节最大运动加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dJ1	J1 轴最大加速度	double	0~720	J1 轴最大加速度，单位[°/s²]
dJ2	J2 轴最大加速度	double	0~720	J2 轴最大加速度，单位[°/s²]
dJ3	J3 轴最大加速度	double	0~720	J3 轴最大加速度，单位[°/s²]
dJ4	J4 轴最大加速度	double	0~720	J4 轴最大加速度，单位[°/s²]
dJ5	J5 轴最大加速度	double	0~720	J5 轴最大加速度，单位[°/s²]
dJ6	J6 轴最大加速度	double	0~720	J6 轴最大加速度，单位[°/s²]

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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

```
// 定义关节最大运动加速度变量
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取最大关节加速度
int nRet= hr_api.HRIF_ReadJointMaxAcc(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.5.12. HRIF_ReadJointMaxJerk

3.5.12.1. 描述：读取关节最大运动加加速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dJ1	J1 轴最大加加速度	double	0~3600	J1 轴最大加加速度，单位[°/s^3]
dJ2	J2 轴最大加加速度	double	0~3600	J2 轴最大加加速度，单位[°/s^3]
dJ3	J3 轴最大加加速度	double	0~3600	J3 轴最大加加速度，单位[°/s^3]
dJ4	J4 轴最大加加速度	double	0~3600	J4 轴最大加加速度，单位[°/s^3]
dJ5	J5 轴最大加加速度	double	0~3600	J5 轴最大加加速度，单位[°/s^3]
dJ6	J6 轴最大加加速度	double	0~3600	J6 轴最大加加速度，单位[°/s^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

3.5.12.2. 示例

```
// 定义关节最大运动加加速度变量
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取最大关节加加速度
int nRet= hr_api.HRIF_ReadJointMaxJerk(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.5.13. HRIF_ReadLinearMaxSpeed

3.5.13.1. 描述：读取直线运动最大速度参数。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dMaxVel	最大直线速度	double	0~3000	最大直线速度，默认[2000]，单位[mm/ s]
dMaxAcc	最大直线加速度	double	0~2500	最大直线加速度，默认[2500]，单位[mm/s^2]
dMaxJerk	最大直线加加速度	double	0~10000000	最大直线加加速度，默认[10000000]，单位[mm/s^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

3.5.13.2. 示例

```
// 定义直线最大运动速度变量
double dMaxVel = 0; double dMaxAcc = 0; double dMaxJerk = 0;
// 读取最大直线速度
int nRet= hr_api.HRIF_ReadLinearMaxSpeed (0,0, ref dMaxVel , ref dMaxAcc, ref dMaxJerk);
```

3.5.14. HRIF_ReadEmergencyInfo

3.5.14.1. 描述：读取急停信息。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nESTO_Error	急停错误	int	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
nESTO	急停信号	int	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
nSafetyGuard_Error	安全光幕错误	int	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
nSafetyGuard	安全光幕信号	int	0/1	安全光幕信号，发生安全光幕时，会停止机器人运动，并且不再接受运动指令，不会断本体供电

✓ 返回值

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

3.5.14.2. 示例

```
// 定义直线最大运动速度变量
int nESTO_Error = 0; int nESTO = 0; int nSafetyGuard_Error = 0; int nSafetyGuard = 0;
// 读取最大直线速度
int nRet= hr_api.HRIF_ReadEmergencyInfo (0,0, ref nESTO_Error, ref nESTO, ref nSafetyGuard_Error,
ref nSafetyGuard);
```

3.5.15. HRIF_ReadRobotState

3.5.15.1. 描述：读取当前机器人状态标志。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nMovingState	运动状态	int	0/1	0：机器人不处于运动状态 1：机器人运动中
nEnableState	使能状态	int	0/1	0：机器人未使能 1：机器人已使能
nErrorState	错误状态	int	0/1	0：未发生错误 1：有错误发生
nErrorCode	错误码	int	>= 0	报错错误码
nErrorAxis	错误轴 ID	int	0~5	错误轴 ID
nBreaking	抱闸状态	int	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
nPause	暂停状态	int	0/1	0：未处于暂停状态 1：已处于暂停状态
nEmergencyStop	急停状态	int	0/1	0：未处于急停状态 1：已处于急停状态
nSaftyGuard	安全光幕状态	int	0/1	0：未处于安全光幕状态 1：已处于安全光幕状态
nElectrify	上电状态	int	0/1	0：未上电状态 1：已上电状态
nIsConnectToBox	电箱连接状态	int	0/1	0：电箱未连接

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

✓ 返回值

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

3.5.15.2. 示例

```
// 定义需要读取的机器人状态变量
int nMovingState = 0; int nEnableState = 0; int nErrorState = 0; int nErrorCode = 0;
int nErrorAxis = 0; int nBreaking = 0; int nPause = 0; int nEmergencyStop = 0;
int nSaftyGuard = 0; int nElectrify = 0; int nIsConnectToBox = 0; int nBlendingDone = 0; int nInPos = 0;
// 读取状态
int nRet= hr_api.HRIF_ReadRobotState(0,0, ref nMovingState, ref nEnableState, ref nErrorState, ref nErrorCode,
ref nErrorAxis, ref nBreaking, ref nPause, ref nEmergencyStop, ref nSaftyGuard, ref nElectrify,
ref nIsConnectToBox, ref nBlendingDone, ref nInPos);
```

3.5.16. HRIF_ReadRobotFlags

3.5.16.1. 描述：读取当前机器人状态标志。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nMovingState	运动状态	int	0/1	0：机器人不处于运动状态 1：机器人运动中
nEnableState	使能状态	int	0/1	0：机器人未使能 1：机器人已使能
nErrorState	错误状态	int	0/1	0：未发生错误 1：有错误发生
nErrorCode	错误码	int	>= 0	报错错误码
nErrorAxis	错误轴 ID	int	0~5	错误轴 ID
nBreaking	抱闸状态	int	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
nPause	暂停状态	int	0/1	0：未处于暂停状态 1：已处于暂停状态
nBlendingDone	路点运动状态	int	0/1	0：路点运动中 1：路点运动完成

✓ 返回值

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

3.5.16.2. 示例

```
// 定义需要读取的机器人状态变量
int nMovingState = 0; int nEnableState = 0; int nErrorState = 0; int nErrorCode = 0;
int nErrorAxis = 0; int nBreaking = 0; int nPause = 0; int nBlendingDone = 0;
// 读取状态
int nRet= hr_api.HRIF_ReadRobotState(0,0, ref nMovingState, ref nEnableState, ref nErrorState, ref nErrorCode,
ref nErrorAxis, ref nBreaking, ref nPause, ref nBlendingDone);
```

3.5.17. HRIF_ReadCurWaypointID

3.5.17.1. 描述：读取 WayPoint 当前运动 ID 号。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.5.17.2. 示例

```
// 定义路点运动 ID 变量
string strCurWaypointID = " ";
// 读取路点 ID
int nRet= hr_api.HRIF_ReadCurWaypointID(0,0, ref strCurWaypointID);
```

3.5.18. HRIF_ReadAxisErrorCode

3.5.18.1. 描述：读取错误码。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nErrorCode	当前错误码	int	>= 0	当前错误码，没有错误码时= 0
nJ1	J1 轴错误码	int	>= 0	J1 轴错误码，没有错误码时= 0
nJ2	J2 轴错误码	int	>= 0	J2 轴错误码，没有错误码时= 0
nJ3	J3 轴错误码	int	>= 0	J3 轴错误码，没有错误码时= 0
nJ4	J4 轴错误码	int	>= 0	J4 轴错误码，没有错误码时= 0
nJ5	J5 轴错误码	int	>= 0	J5 轴错误码，没有错误码时= 0
nJ6	J6 轴错误码	int	>= 0	J6 轴错误码，没有错误码时= 0

✓ 返回值

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

3.5.18.2. 示例

```
// 定义错误码变量
int nErrorCode = 0;
// 定义各轴错误码变量
int nJ1= 0; int nJ2= 0; int nJ3= 0; int nJ4= 0; int nJ5= 0; int nJ6= 0;
// 读取错误码
int nRet= hr_api.HRIF_ReadAxisErrorCode (0,0,ref nErrorCode, ref nJ1, ref nJ2, ref nJ3, ref nJ4, ref nJ5, ref nJ6);
```

3.5.19. HRIF_ReadCurFSM

3.5.19.1. 描述：读取当前状态机。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nCurFSM	当前状态机	int	>= 0	状态机，可以参考状态机列表
strCurFSM	当前状态机描述	string	-	当前状态机描述

✓ 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

3.5.19.2. 示例

```
// 定义状态机变量
int nCurFSM = 0;
// 定义状态机字段变量
string strCurFSM = " ";
// 读取当前状态机
int nRet= hr_api.HRIF_ReadCurFSM(0,0, ref nCurFSM, ref strCurFSM);
```

3.5.20. HRIF_ReadCurFSMFromCPS

3.5.20.1. 描述：读取当前状态机。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
nCurFSM	当前状态机	int	>= 0	状态机，可以参考状态机列表

✓ 返回值

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

3.5.20.2. 示例

```
// 定义状态机变量
int nCurFSM = 0;
// 读取当前状态机
int nRet= hr_api.HRIF_ReadCurFSMFromCPS(0,0, ref nCurFSM);
```

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	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dX -dRz	迪卡尔坐标	double	-	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	关节坐标	double	-	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	当前工具坐标	double	-	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[°]

dUcs_X- dUcs_Rz	当前用户坐标	double	-	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

```
// 定义空间位置变量
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 定义关节位置变量
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 定义用户坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 读取当前实际位置信息
int nRet= hr_api.HRIF_ReadActPos(0,0, ref dX, ref dY, ref dZ, ref dRx, ref dRy, ref dRz, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6, ref dTcp_X, ref dTcp_Y, ref dTcp_Z, ref dTcp_Rx, ref dTcp_Ry, ref dTcp_Rz, ref dUcs_X, ref dUcs_Y, ref dUcs_Z, ref dUcs_Rx, ref dUcs_Ry, ref dUcs_Rz);
```

3.6.2. HRIF_ReadCmdJointPos

3.6.2.1. 描述：读取关节命令位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.2.2. 示例

```
// 定义关节命令位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取关节命令位置
int nRet= hr_api.HRIF_ReadCmdJointPos(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.6.3. HRIF_ReadActJointPos

3.6.3.1. 描述：读取关节实际位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.3.2. 示例

```
// 定义关节实际位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取关节实际位置
int nRet= hr_api.HRIF_ReadActJointPos(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.6.4. HRIF_ReadCmdTcpPos

3.6.4.1. 描述：读取命令 TCP 位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dX -dRz	TCP 坐标	double	-	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

```
// TCP 命令位置
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 读取 TCP 命令位置
int nRet= hr_api.HRIF_ReadCmdTcpPos(0,0, ref dX, ref dY, ref dZ, ref dRx, ref dRy, ref dRz);
```

3.6.5. HRIF_ReadActTcpPos

3.6.5.1. 描述：读取实际 TCP 位置。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.5.2. 示例

```
// TCP 实际位置
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 读取 TCP 实际位置
int nRet= hr_api.HRIF_ReadActTcpPos(0,0, ref dX, ref dY, ref dZ, ref dRx, ref dRy, ref dRz);
```

3.6.6. HRIF_ReadCmdJointVel

3.6.6.1. 描述：读取关节命令速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.6.2. 示例

```
// 定义关节实际位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取关节实际位置
int nRet= hr_api.HRIF_ReadCmdJointVel(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.6.7. HRIF_ReadActJointVel

3.6.7.1. 描述：读取关节实际速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.7.2. 示例

```
// 定义关节实际速度
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取关节实际速度
int nRet= hr_api.HRIF_ReadActJointVel(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.6.8. HRIF_ReadCmdTcpVel

3.6.8.1. 描述：读取命令 TCP 速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.8.2. 示例

```
// TCP 命令速度
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 读取 TCP 命令速度
int nRet= hr_api.HRIF_ReadCmdTcpVel(0,0, ref dX, ref dY, ref dZ, ref dRx, ref dRy, ref dRz);
```

3.6.9. HRIF_ReadActTcpVel

3.6.9.1. 描述：读取实际 TCP 速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.9.2. 示例

```
// TCP 实际速度
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 读取 TCP 实际速度
int nRet= hr_api.HRIF_ReadActTcpVel(0,0, ref dX, ref dY, ref dZ, ref dRx, ref dRy, ref dRz);
```

3.6.10. HRIF_ReadCmdJointCur

3.6.10.1. 描述：读取关节命令电流。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.10.2. 示例

```
// 定义关节命令电流
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取关节命令电流
int nRet= hr_api.HRIF_ReadCmdJointCur(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.6.11. HRIF_ReadActJointCur

3.6.11.1. 描述：读取关节实际电流。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.6.11.2. 示例

```
// 定义关节实际电流
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 读取关节实际电流
int nRet= hr_api.HRIF_ReadActJointCur(0,0, ref dJ1, ref dJ2, ref dJ3, ref dJ4, ref dJ5, ref dJ6);
```

3.6.12. HRIF_ReadTcpVelocity

3.6.12.1. 描述：读取 TCP 末端速度。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dCmdVel	命令速度	double	-	命令速度，单位[mm/s]
dActVel	实际速度	double	-	实际速度，单位[mm/s]

✓ 返回值

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

3.6.12.2. 示例

```
// 定义 TCP 速度变量
double dCmdVel = 0;
double dActVel = 0;
// 读取当前末端 TCP 速度
int nRet= hr_api.HRIF_ReadTcpVelocity(0,0, ref dCmdVel, ref dActVel);
```

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	unsigned 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
dQuaW	W	double	-	W
dQuaX	Xi	double	-	Xi
dQuaY	Yj	double	-	Yj
dQuaZ	Zk	double	-	Zk

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dRx	欧拉角 Rx	double	>=-180,=<180	欧拉角 Rx
dRy	欧拉角 Ry	double	>=-180,=<180	欧拉角 Ry
dRz	欧拉角 Rz	double	>=-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

```
// 需要转换的四元素变量
double dQuaW = 0; double dQuaX = 0; double dQuaY= 0; double dQuaZ =0;
// 转换后的欧拉角结果
double dRx = 0; double dRy = 0; double dRz = 0;
// 转换
int nRet= hr_api.HRIF_Quaternion2RPY(0,0, dQuaW, dQuaX, dQuaY, dQuaZ, ref dRx, ref dRy, ref dRz);
```

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	unsigned 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
Rx	欧拉角 Rx	double	>=-180,=<180	欧拉角 Rx
Ry	欧拉角 Ry	double	>=-180,=<180	欧拉角 Ry
Rz	欧拉角 Rz	double	>=-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
dQuaW	W	double	-	W
dQuaX	Xi	double	-	Xi
dQuaY	Yj	double	-	Yj
dQuaZ	Zk	double	-	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

```
// 需要转换的转换后的欧拉角变量
double dRx = 0; double dRy = 0; double dRz = 0;
// 转换后的四元素变量
double dQuaW = 0; double dQuaX = 0; double dQuaY= 0; double dQuaZ = 0;
// 转换
int nRet= hr_api.HRIF_RPY2Quaternion (0,0, dRx, dRy, dRz, ref dQuaW, ref dQuaX, ref dQuaY, ref dQuaZ);
```

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	unsigned 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
dCoord_X-dCoord_Rz	需要计算逆解的目标迪卡尔位置	double	-	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	double	-	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	double	-	目标位置是否包含用户坐标(不包含用户坐标则所有值= 0): dUcs_X：X 坐标，单位[mm] dUcs_Y：Y 坐标，单位[mm] dUcs_Z：Z 坐标，单位[mm] dUcs_Rx：Rx 坐标，单位[°] dUcs_Ry：Ry 坐标，单位[°] dUcs_Rz：Rz 坐标，单位[°]

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

✓ 输出变量

输出变量	名称	数据类型	有效范围	内容
dTargetJ1-dTargetJ6	关节坐标	double	-	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

```
// 定义需要转换的空间位置变量
double dCoord_X = 420; double dCoord_Y = 0; double dCoord_Z = 445;
double dCoord_Rx = 180; double dCoord_Ry = 0; double dCoord_Rz = 180;
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 定义工具坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 定义参考关节位置变量
```

```
double dJ1 = 0; double dJ2 = 0; double dJ3 = 90;
double dJ4 = 0; double dJ5 = 90; double dJ6 = 0;
// 定义转换结果
double dTargetJ1 = 0; double dTargetJ2 = 0; double dTargetJ3 = 0;
double dTargetJ4 = 0; double dTargetJ5 = 0; double dTargetJ6 = 0;
// 求逆解
int nRet = hr_api.HRIF_GetInverseKin(0,0, dCoord_X, dCoord_Y, dCoord_Z, dCoord_Rx, dCoord_Ry, dCoord_Rz,
dJ1, dJ2, dJ3, dJ4, dJ5, dJ6, dTcp_X, dTcp_Y, dTcp_Z, dTcp_Rx, dTcp_Ry, dTcp_Rz, dUcs_X, dUcs_Y, dUcs_Z,
dUcs_Rx, dUcs_Ry, dUcs_Rz, ref dTargetJ1, ref dTargetJ2, ref dTargetJ3, ref dTargetJ4, ref dTargetJ5,
ref dTargetJ6);
```

3.7.4. HRIF_GetForwardKin

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dJ1 -dJ6	需要计算正解的关节坐标	double	-	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	double	-	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	double	-	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
dTarget_X-dTarget_Rz	目标迪卡尔坐标	double	-	计算正解的目标迪卡尔位置： 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

3.7.4.2. 示例

```
// 定义需要转换的关节位置变量
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 定义用户坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 定义转换后的空间位置结果
double dTarget_X = 0; double dTarget_Y = 0; double dTarget_Z = 0;
double dTarget_Rx = 0; double dTarget_Ry = 0; double dTarget_Rz = 0;
// 求正解
int nRet= hr_api.HRIF_GetForwardKin(0,0, dJ1, dJ2, dJ3, dJ4, dJ5, dJ6, dTcp_X, dTcp_Y, dTcp_Z, dTcp_Rx, dTcp_Ry,
dTcp_Rz,dUcs_X, dUcs_Y, dUcs_Z, dUcs_Rx, dUcs_Ry, dUcs_Rz, ref dTarget_X, ref dTarget_Y,
ref dTarget_Z, ref dTarget_Rx, ref dTarget_Ry, ref dTarget_Rz);
```

3.7.5. HRIF_Base2UcsTcp

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dCoord_X-dCoord_Rz	基于基座坐标系的迪卡尔位置	double	-	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	double	-	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	double	-	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
dTarget_X-dTarget_Rz	目标迪卡尔坐标	double	-	指定用户坐标系和工具坐标系下的迪卡尔坐标位置： dTargetX：X 坐标，单位[mm] dTargetY：Y 坐标，单位[mm] dTargetZ：Z 坐标，单位[mm] dTargetRx：Rx 坐标，单位[°] dTargetRy：Ry 坐标，单位[°] dTargetRz：Rz 坐标，单位[°]

✓ 返回值

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

3.7.5.2. 示例

```
// 定义需要转换的空间位置变量
double dCoord_X = 0; double dCoord_Y = 0; double dCoord_Z = 0;
double dCoord_Rx = 0; double dCoord_Ry = 0; double dCoord_Rz = 0;
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 定义用户坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 定义转换后的空间位置结果
double dTarget_X = 0; double dTarget_Y = 0; double dTarget_Z = 0;
double dTarget_Rx = 0; double dTarget_Ry = 0; double dTarget_Rz = 0;
// 基座坐标转换为用户坐标
int nRet= hr_api.HRIF_Base2UcsTcp(0,0,dCoord_X,dCoord_Y,dCoord_Z,dCoord_Rx,dCoord_Ry,dCoord_Rz,
dTcp_X, dTcp_Y, dTcp_Z, dTcp_Rx, dTcp_Ry, dTcp_Rz, dUcs_X, dUcs_Y, dUcs_Z, dUcs_Rx, dUcs_Ry,
```

dUcs_Rz, ref dTarget_X, ref dTarget_Y, ref dTarget_Z, ref dTarget_Rx, ref dTarget_Ry, ref dTarget_Rz);

3.7.6. HRIF_UcsTcp2Base

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dCoord_X-dCoord_Rz	迪卡尔位置	double	-	指定用户坐标系和工具坐标系下的迪卡尔坐标位置： dCoord_X：X 坐标，单位[mm] dCoord_Y：Y 坐标，单位[mm] dCoord_Z：Z 坐标，单位[mm] dCoord_Rx：Rx 坐标，单位[°] dCoord_Ry：Ry 坐标，单位[°] dCoord_Rz：Rz 坐标，单位[°]
dTcp_X-dTcp_Rz	工具坐标	double	-	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	double	-	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
dTarget_X-dTarget_Rz	目标迪卡尔坐标	double	-	基座坐标系下的迪卡尔坐标位置： dTargetX：X 坐标，单位[mm] dTargetY：Y 坐标，单位[mm] dTargetZ：Z 坐标，单位[mm] dTargetRx：Rx 坐标，单位[°] dTargetRy：Ry 坐标，单位[°] dTargetRz：Rz 坐标，单位[°]

✓ 返回值

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

3.7.6.2. 示例

```
// 定义需要转换的空间位置变量
double dCoord_X = 0; double dCoord_Y = 0; double dCoord_Z = 0;
double dCoord_Rx = 0; double dCoord_Ry = 0; double dCoord_Rz = 0;
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 定义用户坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 定义转换后的空间位置结果
double dTarget_X = 0; double dTarget_Y = 0; double dTarget_Z = 0;
double dTarget_Rx = 0; double dTarget_Ry = 0; double dTarget_Rz = 0;
// 用户坐标转换为基座坐标
int nRet= hr_api.HRIF_UcsTcp2Base(0,0,dCoord_X,dCoord_Y,dCoord_Z,dCoord_Rx,dCoord_Ry,dCoord_Rz,
dTcp_X, dTcp_Y, dTcp_Z, dTcp_Rx, dTcp_Ry, dTcp_Rz, dUcs_X, dUcs_Y, dUcs_Z, dUcs_Rx, dUcs_Ry,
```

dUcs_Rz, ref dTarget_X, ref dTarget_Y, ref dTarget_Z, ref dTarget_Rx, ref dTarget_Ry, ref dTarget_Rz);

3.7.7. HRIF_PoseAdd

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dPose1_X-dPose1_Rz	空间坐标 1	double	-	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	double	-	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
dPose3_X-dPose3_Rz	计算结果	double	-	计算结果： dPose3_X：X 坐标，单位[mm] dPose3_Y：Y 坐标，单位[mm] dPose3_Z：Z 坐标，单位[mm] dPose3_Rx：Rx 坐标，单位[°] dPose3_Ry：Ry 坐标，单位[°] dPose3_Rz：Rz 坐标，单位[°]

✓ 返回值

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

3.7.7.2. 示例

```
// 定义需要计算的空间坐标 1
double dPose1_X = 420; double dPose1_Y = 0; double dPose1_Z = 445;
double dPose1_Rx = 180; double dPose1_Ry = 0; double dPose1_Rz = 180;
// 定义需要计算的空间坐标 2
double dPose2_X = 420; double dPose2_Y = 50; double dPose2_Z = 445;
double dPose2_Rx =18 0; double dPose2_Ry = 0; double dPose2_Rz = 180;
// 计算结果
double dPose3_X = 0; double dPose3_Y = 0; double dPose3_Z = 0;
double dPose3_Rx = 0; double dPose3_Ry = 0; double dPose3_Rz = 0;
// 计算结果
int nRet= hr_api.HRIF_PoseAdd(0,0, dPose1_X, dPose1_Y, dPose1_Z, dPose1_Rx, dPose1_Ry, dPose1_Rz, dPose2_X, dPose2_Y, dPose2_Z, dPose2_Rx, dPose2_Ry, dPose2_Rz, ref dPose3_X, ref dPose3_Y, ref dPose3_Z, ref dPose3_Rx, ref dPose3_Ry, ref dPose3_Rz);
```

3.7.8. HRIF_PoseSub

3.7.8.1. 描述：点位减法计算，以第二个点为参考点。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dPose1_X-dPose1_Rz	空间坐标 1	double	-	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	double	-	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
dPose3_X-dPose3_Rz	计算坐标	double	-	计算结果： dPose3_X：X 坐标，单位[mm] dPose3_Y：Y 坐标，单位[mm] dPose3_Z：Z 坐标，单位[mm] dPose3_Rx：Rx 坐标，单位[°] dPose3_Ry：Ry 坐标，单位[°] dPose3_Rz：Rz 坐标，单位[°]

✓ 返回值

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

3.7.8.2. 示例

```
// 定义需要计算的空间坐标 1
double dPose1_X = 420; double dPose1_Y = 0; double dPose1_Z = 445;
double dPose1_Rx = 180; double dPose1_Ry = 0; double dPose1_Rz = 180;
// 定义需要计算的空间坐标 2
double dPose2_X = 420; double dPose2_Y = 50; double dPose2_Z = 445;
double dPose2_Rx =18 0; double dPose2_Ry = 0; double dPose2_Rz = 180;
// 计算结果
double dPose3_X = 0; double dPose3_Y = 0; double dPose3_Z = 0;
double dPose3_Rx = 0; double dPose3_Ry = 0; double dPose3_Rz = 0;
// 计算结果
int nRet= hr_api.HRIF_PoseSub(0,0, dPose1_X, dPose1_Y, dPose1_Z, dPose1_Rx, dPose1_Ry, dPose1_Rz,
dPose2_X, dPose2_Y, dPose2_Z, dPose2_Rx, dPose2_Ry, dPose2_Rz, ref dPose3_X, ref dPose3_Y,
ref dPose3_Z, ref dPose3_Rx, ref dPose3_Ry, ref dPose3_Rz);
```

3.7.9. HRIF_PoseTrans

3.7.9.1. 描述：坐标变换。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dPose1_X-dPose1_Rz	坐标位置 1	double	-	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	double	-	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
dPose3_X-dPose3_Rz	计算坐标	double	-	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

```
// 定义需要计算的空间坐标 1
double dPose1_X = 420; double dPose1_Y = 0; double dPose1_Z = 445;
double dPose1_Rx = 180; double dPose1_Ry = 0; double dPose1_Rz = 180;
// 定义需要计算的空间坐标 2
double dPose2_X = 420; double dPose2_Y = 50; double dPose2_Z = 445;
double dPose2_Rx =18 0; double dPose2_Ry = 0; double dPose2_Rz = 180;
// 计算结果
double dPose3_X = 0; double dPose3_Y = 0; double dPose3_Z = 0;
double dPose3_Rx = 0; double dPose3_Ry = 0; double dPose3_Rz = 0;
// 计算结果
int nRet= hr_api.HRIF_PoseTrans(0,0, dPose1_X, dPose1_Y, dPose1_Z, dPose1_Rx, dPose1_Ry, dPose1_Rz,
dPose2_X, dPose2_Y, dPose2_Z, dPose2_Rx, dPose2_Ry, dPose2_Rz, ref dPose3_X, ref dPose3_Y,
ref dPose3_Z, ref dPose3_Rx, ref dPose3_Ry, ref dPose3_Rz);
```

3.7.10. HRIF_PoseInverse

3.7.10.1. 描述：坐标逆变换。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dPose1_X-dPose1_Rz	空间坐标 1	double	-	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
dPose3_X-dPose3_Rz	计算坐标	double	-	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

```
// 定义需要计算的空间坐标 1
```

```
double dPose1_X = 420; double dPose1_Y = 0; double dPose1_Z = 445;
double dPose1_Rx = 180; double dPose1_Ry = 0; double dPose1_Rz = 180;
// 计算结果
double dPose3_X = 0; double dPose3_Y = 0; double dPose3_Z = 0;
double dPose3_Rx = 0; double dPose3_Ry = 0; double dPose3_Rz = 0;
// 计算结果
int nRet= hr_api.HRIF_PoseInverse(0,0, dPose1_X, dPose1_Y, dPose1_Z, dPose1_Rx, dPose1_Ry, dPose1_Rz,
ref dPose3_X, ref dPose3_Y, ref dPose3_Z, ref dPose3_Rx, ref dPose3_Ry, ref dPose3_Rz);
```

3.7.11. HRIF_PoseDist

3.7.11.1. 描述：计算点位距离。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dPose1_X-dPose1_Rz	空间坐标 1	double	-	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	double	-	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
dDistance	点位距离	double	-	点位距离，单位[mm]
dAngle	姿态距离	double	-	姿态距离，单位[°]

✓ 返回值

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

3.7.11.2. 示例

```
// 定义需要计算的空间坐标 1
double dPose1_X = 420; double dPose1_Y = 0; double dPose1_Z = 445;
double dPose1_Rx = 180; double dPose1_Ry = 0; double dPose1_Rz = 180;
// 定义需要计算的空间坐标 2
double dPose2_X = 420; double dPose2_Y = 50; double dPose2_Z = 445;
double dPose2_Rx =18 0; double dPose2_Ry = 0; double dPose2_Rz = 180;
// 计算结果
double dDistance = 0;
double dAngle = 0;
// 计算结果
int nRet= hr_api.HRIF_PoseDist(0,0, dPose1_X, dPose1_Y, dPose1_Z, dPose1_Rx, dPose1_Ry, dPose1_Rz, dPose2_X, dPose2_Y, dPose2_Z, dPose2_Rx, dPose2_Ry, dPose2_Rz, ref dDistance, ref dAngle);
```

3.7.12. HRIF_Poseinterpolate

3.7.12.1. 描述：空间位置直线插补计算。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dPose1_X-dPose1_Rz	空间坐标 1	double	-	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	double	-	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	double	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
dPose3_X-dPose3_Rz	计算坐标	double	-	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

```
// 定义需要计算的空间坐标 1
double dPose1_X = 420; double dPose1_Y = 0; double dPose1_Z = 445;
double dPose1_Rx = 180; double dPose1_Ry = 0; double dPose1_Rz = 180;
// 定义需要计算的空间坐标 2
double dPose2_X = 420; double dPose2_Y = 50; double dPose2_Z = 445;
double dPose2_Rx =18 0; double dPose2_Ry = 0; double dPose2_Rz = 180;
// 插补比例
double dAlpha = 0.5;
// 计算结果
double dPose3_X = 0; double dPose3_Y = 0; double dPose3_Z = 0;
double dPose3_Rx = 0; double dPose3_Ry = 0; double dPose3_Rz = 0;
// 计算结果
int nRet= hr_api.HRIF_PoseInterpolate(0,0, dPose1_X, dPose1_Y, dPose1_Z, dPose1_Rx, dPose1_Ry, dPose1_Rz,
dPose2_X, dPose2_Y, dPose2_Z, dPose2_Rx, dPose2_Ry, dPose2_Rz, dAlpha, ref dPose3_X, ref dPose3_Y,
ref dPose3_Z, ref dPose3_Rx, ref dPose3_Ry, ref dPose3_Rz);
```

3.7.13. HRIF_PoseDefdFrame

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dPose1_X-dPose1_Z	空间坐标 1	double	-	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	double	-	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	double	-	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	double	-	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	double	-	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	double	-	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
dUcs_X- dUcs_Rz	计算结果	double	-	计算结果： dUcs_X：X 坐标，单位[mm] dUcs_Y：Y 坐标，单位[mm] dUcs_Z：Z 坐标，单位[mm] dUcs_Rx：Rx 坐标，单位[°] dUcs_Ry：Ry 坐标，单位[°] dUcs_Rz：Rz 坐标，单位[°]

✓ 返回值

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

3.7.13.2. 示例

```
// 定义需要计算的空间坐标 1
double dPose1_X = 0; double dPose1_Y = 0; double dPose1_Z = 0;
// 定义需要计算的空间坐标 2
double dPose2_X = 0; double dPose2_Y = 0; double dPose2_Z = 0;
// 定义需要计算的空间坐标 3
double dPose3_X = 0; double dPose3_Y = 0; double dPose3_Z = 0;
// 定义需要计算的空间坐标 4
double dPose4_X = 0; double dPose4_Y = 0; double dPose4_Z = 0;
// 定义需要计算的空间坐标 5
double dPose5_X = 0; double dPose5_Y = 0; double dPose5_Z = 0;
// 定义需要计算的空间坐标 6
double dPose6_X = 0; double dPose6_Y = 0; double dPose6_Z = 0;
```

```
// 计算结果
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 计算结果
int nRet= hr_api.HRIF_PoseDefdFrame(0,0, dPose1_X, dPose1_Y, dPose1_Z, dPose2_X, dPose2_Y, dPose2_Z, dPose3_X, dPose3_Y, dPose3_Z, dPose4_X, dPose4_Y, dPose4_Z, dPose5_X, dPose5_Y, dPose5_Z, dPose6_X, dPose6_Y, dPose6_Z, ref dUcs_X, ref dUcs_Y, ref dUcs_Z, ref dUcs_Rx, ref dUcs_Ry, ref dUcs_Rz);
```

3.8. 工具坐标与用户坐标读写指令

3.8.1. HRIF_SetTCP

3.8.1.1. 描述：设置当前工具坐标，不写入配置文件，重启后失效。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dTcp_X- dTcp_Rz	工具坐标	double	-	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
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 设置工具坐标
int nRet= hr_api.HRIF_SetTCP(0,0, dTcp_X, dTcp_Y, dTcp_Z, dTcp_Rx, dTcp_Ry, dTcp_Rz);
```

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	unsigned 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
dUcs_X- dUcs_Rz	用户坐标	double	-	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
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 设置用户坐标
int nRet= hr_api.HRIF_SetUCS(0,0, dUcs_X, dUcs_Y, dUcs_Z, dUcs_Rx, dUcs_Ry, dUcs_Rz);
```

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	unsigned 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

✓ 输出变量

输入变量	名称	数据类型	有效范围	内容
dTcp_X- dTcp_Rz	工具坐标	double	-	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

```
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 读取工具坐标
int nRet= hr_api.HRIF_ReadCurTCP(0,0, ref dTcp_X, ref dTcp_Y, ref dTcp_Z, ref dTcp_Rx, ref dTcp_Ry,
ref dTcp_Rz);
```

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	unsigned 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

✓ 输出变量

输入变量	名称	数据类型	有效范围	内容
dUcs_X- dUcs_Rz	用户坐标	double	-	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

```
// 定义用户坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 读取用户坐标
int nRet= hr_api.HRIF_ReadCurUCS(0,0, ref dUcs_X, ref dUcs_Y, ref dUcs_Z, ref dUcs_Rx, ref dUcs_Ry,
ref dUcs_Rz);
```

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	unsigned 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
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
string sTcpName = "TCP";
// 设置工具坐标
int nRet= hr_api.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	unsigned 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
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
string sUcsName = "Base";
// 设置用户坐标
int nRet= hr_api.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	unsigned 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
sTcpName	工具坐标名称	string	-	需要读取的工具坐标名称

✓ 输出变量

输入变量	名称	数据类型	有效范围	内容
dTcp_X- dTcp_Rz	工具坐标	double	-	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

```
// 需要读取的工具坐标名称
string sTcpName = "TCP";
// 定义读取到的工具坐标结果变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 读取工具坐标
int nRet= hr_api.HRIF_ReadTCPByName(0,0,sTcpName, ref dTcp_X, ref dTcp_Y, ref dTcp_Z, ref dTcp_Rx,
```

ref dTcp_Ry, ref dTcp_Rz);

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	unsigned 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
sUcsName	用户坐标名称	string	-	需要读取的用户坐标名称

✓ 输出变量

输入变量	名称	数据类型	有效范围	内容
dUcs_X- dUcs_Rz	用户坐标	double	-	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

```
// 需要读取的用户坐标名称
string sUcsName = "Base";
// 定义读取到的用户坐标变量
double dUcs_X = 420; double dUcs_Y = 0; double dUcs_Z = 445;
double dUcs_Rx = 180; double dUcs_Ry = 0; double dUcs_Rz = 180;
// 读取用户坐标
int nRet= hr_api.HRIF_ReadUCSByName(0,0, sUcsName , ref dUcs_X, ref dUcs_Y, ref dUcs_Z, ref dUcs_Rx,
```

ref dUcs_Ry, ref dUcs_Rz);

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	unsigned 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
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
int nState = 1;
// 设置力控状态
int nRet= hr_api.HRIF_SetForceControlState(0,0,ref 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	unsigned 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

✓ 输出变量

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

✓ 返回值

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

3.9.2.2. 示例

```
// 读取到的力控状态
int nState = 1;
// 读取力控状态
int nRet= hr_api.HRIF_ReadForceControlState(0,0, ref nState);
```

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	unsigned 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
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
int nState = 1;
// 设置力控坐标系状态
int nRet= hr_api.HRIF_SetForceToolCoordinateMotion(0,0, nState);
```

3.9.4. HRIF_ForceControlinterrupt

3.9.4.1. 描述：暂停力控运动，仅暂停力控功能，不暂停运动和脚本。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.HRIF_ForceControlInterrupt(0,0);
```

3.9.5. HRIF_ForceControlContinue

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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

```
// 设置力控暂停状态
int nRet= hr_api.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	unsigned 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

✓ 返回值

返回值	名称	数据类型	有效范围	内容
nRet	返回值	int	>=0 的整型值	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
int nRet= hr_api.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	unsigned 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
dMaxLinearVelocity	直线速度	double	>0	探寻的直线最大速度，单位[mm/s]
dMaxAngularVelocity	姿态角速度	double	>0	探寻的姿态最大速度，单位[°/s]

✓ 返回值

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

3.9.7.2. 示例

```
//设置力控探寻直线速度
double dMaxLinearVelocity = 10;
// 设置力控探寻姿态角速度
double dMaxAngularVelocity = 5;
// 设置力控探寻速度
int nRet= hr_api.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	unsigned 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
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

3.9.8.2. 示例

```
// 定义力控自由度状态
int nX = 0; int nY = 0; int nZ = 1;
int nRx = 0; int nRy = 0; int nRz = 0;
// 设置力控自由度状态
int nRet= hr_api.HRIF_SetControlFreedom(0,0, nX, nY, nZ, nRx, nRy, nRz);
```

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	unsigned 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
nState	控制策略	int	0~2	力控控制策略： 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
int nState = 0;
// 设置力控策略为恒力模式
int nRet= hr_api.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	unsigned 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
dX -dRz	迪卡尔坐标	double	-	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
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 设置力传感器的安装位置和姿态
int nRet= hr_api.HRIF_SetFreeDrivePositionAndOrientation(0,0, dX, dY, dZ, dRx, dRy, dRz);
```

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	unsigned 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
dFp	PID 参数	double	-	PID 参数 Fp
dFi	PID 参数	double	-	PID 参数 Fi
dFd	PID 参数	double	-	PID 参数 Fd
dTp	PID 参数	double	-	PID 参数 Tp
dTi	PID 参数	double	-	PID 参数 Ti
dTd	PID 参数	double	-	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
double dFp = 1.0; double dFi= 0.1; double dFd = 0;
double dTp = 1.0; double dTi = 0.1; double dTd = 0;
// 设置 PID 参数
int nRet= hr_api.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	unsigned 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
dX -dRz	惯量控制参数	double	-	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

```
// 设置质量参数
double dX = 40; double dY = 40; double dZ = 40;
double dRx = 10; double dRy = 10; double dRz = 10;
// 设置质量参数
int nRet= hr_api.HRIF_SetMassParams(0,0, dX, dY, dZ, dRx, dRy, dRz);
```

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	unsigned 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
dX -dRz	阻尼控制参数	double	-	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
double dX = 800; double dY = 800; double dZ = 800;
double dRx = 40; double dRy = 40; double dRz = 40;
// 设置阻尼参数
int nRet= hr_api.HRIF_SetDampParams(0,0, dX, dY, dZ, dRx, dRy, dRz);
```

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	unsigned 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
dX -dRz	刚度控制参数	double	-	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
double dX = 1000; double dY = 1000; double dZ = 1000;
double dRx = 100; double dRy = 100; double dRz = 100;
// 设置刚度参数
int nRet= hr_api.HRIF_SetStiffParams(0,0, dX, dY, dZ, dRx, dRy, dRz);
```

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	unsigned 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
dX -dRz	力控目标力	double	-	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
double dX = 0; double dY = 0; double dZ = 10;
double dRx = 0; double dRy = 0; double dRz = 0;
// 设置力控距离
double dDistance1 = 0; double dDistance2 = 0; double dDistance3 = 0;
double dDistance4 = 0; double dDistance5 = 0; double dDistance6 = 0;
// 设置力控目标力 Z 方向 10N
int nRet= hr_api.HRIF_SetForceControlGoal(0,0, dX, dY, dZ, dRx, dRy, dRz, dDistance1, dDistance2, dDistance3, dDistance4, dDistance5, dDistance6);
```

3.9.16. HRIF_SetControlGoal

3.9.16.1. 描述：设置力控目标力和目标距离(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	unsigned 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
dWrench_X-dWrench_Rz	力控目标力	double	-	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	double	-	暂未启用，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

3.9.16.2. 示例

// 设置力控目标力

double dWrench_X = 0; double dWrench_Y = 0; double dWrench_Z = 10;
double dWrench_Rx = 0; double dWrench_Ry = 0; double dWrench_Rz = 0;

// 设置力控目标距离

double dDistance_X = 0; double dDistance_Y = 0; double dDistance_Z = 10;
double dDistance_Rx = 0; double dDistance_Ry = 0; double dDistance_Rz = 0;

// 设置力控目标力 Z 方向 10N

int nRet= hr_api.HRIF_SetControlGoal (0,0, dWrench_X, dWrench_Y, dWrench_Z, dWrench_Rx, dWrench_Ry, dWrench_Rz, dDistance_X, dDistance_Y, dDistance_Z, dDistance_Rx, dDistance _Ry, dDistance_Rz);

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	unsigned 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
dMax_X-dMax_Rz	力最大范围	double	-	力最大范围： dMax_X：X 方向，单位[N] dMax_Y：Y 方向，单位[N] dMax_Z：Z 方向，单位[N] dMax_Rx：Rx 方向，单位[NM] dMax_Ry：Ry 方向，单位[NM] dMax_Rz：Rz 方向，单位[NM]
dMin_X-dMin_Rz	力最小范围	double	-	力最小范围： dMin_X：X 方向，单位[N] dMin_Y：Y 方向，单位[N] dMin_Z：Z 方向，单位[N] dMin_Rx：Rx 方向，单位[NM] dMin_Ry：Ry 方向，单位[NM] dMin_Rz：Rz 方向，单位[NM]

✓ 返回值

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

3.9.17.2. 示例

```
// 设置力最大值
double dMax_X = 500; double dMax_Y = 500; double dMax_Z = 500;
double dMax_Rx = 500; double dMax_Ry = 500; double dMax_Rz = 500;
```

```
// 设置力最小值
double dMin_X = -500; double dMin_Y = -500; double dMin_Z = -500;
double dMin_Rx = -500; double dMin_Ry = -500; double dMin_Rz = -500;
// 设置力传感器数据限制范围
int nRet= hr_api.HRIF_SetForceDataLimit(0,0, dMax_X, dMax_Y, dMax_Z, dMax_Rx, dMax_Ry, dMax_Rz, dMin_X, dMin_Y, dMin_Z, dMin_Rx, dMin_Ry, dMin_Rz);
```

3.9.18. HRIF_SetForceDistanceLimit

3.9.18.1. 描述：设置力控形变范围。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dAllowDistance	允许最大距离	double		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	double	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
double dAllowDistance = 10;
// 设置位置与边界设置偏离距离的幂次项
double dStrengthLevel = 2;
// 设置力控形变范围
int nRet= hr_api.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	unsigned 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
bEnable	是否开启	bool	false/true	false：关闭 true：开启

✓ 返回值

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

3.9.19.2. 示例

```
// 设置开启力控自由驱动
bool bEnable = true;
// 设置开启力控自由驱动
int nRet= hr_api.HRIF_SetForceFreeDriveMode(0,0, bEnable);
```

3.9.20. HRIF_ReadFTCabData

3.9.20.1. 描述：读取标定后的力传感器数据。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

输入变量	名称	数据类型	有效范围	内容
dX -dRz	标定后的力传感器数据	double	-	dX：X 坐标，单位[N] dY：Y 坐标，单位[N] dZ：Z 坐标，单位[N] dRx：Rx 坐标，单位[NM] dRy：Ry 坐标，单位[NM] dRz：Rz 坐标，单位[NM]

✓ 返回值

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

3.9.20.2. 示例

```
// 读取到力传感器数据
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 读取力传感器数据
int nRet= hr_api.HRIF_ReadFTCabData(0,0, ref dX, ref dY, dZ, ref dRx, ref dRy, ref dRz);
```

3.10. 通用运动类控制指令

3.10.1. HRIF_MoveRelJ

3.10.1.1. 描述：关节相对运动。

✓ 输入变量

参数	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nAxis ID	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	double	-	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

MoveRelL

3.10.1.2. 示例

```
//运动的目标轴 ID，对应关节 J1-J6
int nAxisId = 2;
//方向
int nDirection = 1;
//运动距离
double dDistance = 10;
//关节相对运动
int nRet= hr_api.HRIF_MoveRelJ(0,0, nAxisId, nDirection, dDistance);
```

3.10.2. HRIF_MoveRelL

3.10.2.1. 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	unsigned 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
nAxis ID	轴 ID	int	0~5	运动的目标轴 ID ，对应空间坐标 X-Rz
nDirection	方向	int	0/1	0：负方向 1：正方向
dDistance	运动距离	double	-	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

```
//运动的目标轴 ID，对应空间坐标 X-Rz
int nAxisId = 2;
//方向
int nDirection = 1;
//运动距离
double dDistance = 10;
//运动坐标类型
int nToolMotion = 0;
//空间相对运动
int nRet= hr_api.HRIF_MoveRelL(0,0, nAxisId, nDirection, dDistance, nToolMotion);
```

3.10.3. HRIF_WayPointRel

3.10.3.1. 描述：路点相对运动。

✓ 输入变量

参数	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nType	运动类型	int	0/1	0：关节相对运动 1：直线相对运动
nPointList	是否使用点位列表点位	int	0/1	0：不使用点位列表中点位 1：使用点位列表中点位
dPos_X-dPos_Rz	空间位置	double		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
dPos_J1-dPos_J6	关节位置	double		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	0/1	0：绝对值 1：叠加值
nAxisMask_1-	是否运动	int	0/1	各轴/各方向是否运动

nAxisMask_6				0：不运动 1：运动
dTarget_1- dTarget_6	运动距离	double	-	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	double	-	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	double	-	运动最大加速度，关节运动时单位[°/s^2]，空间运动时 X，Y，Z 单位[mm/s^2]，Rx，Ry，Rz 单位[°/s^2]
dRadius	过渡半径	double	-	过渡半径，单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	是否使用关节角度作为目标点，如果 nMoveType= 0 时，则 nIsUseJoint 有效： 0：不使用关节角度 1：使用关节角度
nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1，则开启检测 DI 停止，路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态= nIOState 时，机器人停止运动，否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引，nIsSeek= 0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态，，nIsSeek= 0 时无效

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

✓ 返回值

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

3.10.3.2. 示例

```
//运动类型
int nMoveType = 0;
//是否使用点位列表点位
int nPointList = 1;
//空间位置
double dPos_X = 420; double dPos_Y = 0; double dPos_Z = 445;
double dPos_Rx = 180; double dPos_Ry = 0; double dPos_Rz = 180;
//关节位置
double dPos_J1 = 0; double dPos_J2 = 0; double dPos_J3 = 90;
double dPos_J4 = 0; double dPos_J5 = 90; double dPos_J6 = 0;
//相对运动类型
int nrelMoveType = 0;
//是否运动
int nAxisMask_1 = 0; int nAxisMask_2 = 0; int nAxisMask_3 = 1;
int nAxisMask_4 = 0; int nAxisMask_5 = 1; int nAxisMask_6 = 0;
//运动距离
double dTarget_1 = 0; double dTarget_2 = 0; double dTarget_3 = 0;
double dTarget_4 = 0; double dTarget_5 = 0; double dTarget_6 = 0;
//工具坐标名称
string sTcpName = "TCP";
//用户坐标名称
string sUcsName = "Base";
//速度
```

```
double dVelocity = 5;
//加速度
double dAcc = 360;
//过度半径
double dRadius = 50;
//是否使用关节坐标
int nIsUseJoint = 1;
//是否检测 DI 停止
int nIsSeek = 1;
//检测的 DI 索引
int nIOBit = 0;
//检测的 DI 状态
int nIOState = 1;
//命令 ID
string strCmdID = "1";
int nRet= hr_api.HRIF_WayPointRel(0,0, nMoveType, nPointList, dPos_X, dPos_Y, dPos_Z, dPos_Rx, dPos_Ry, dPos_Rz, dPos_J1, dPos_J2, dPos_J3, dPos_J4, dPos_J5, dPos_J6, nrelMoveType, nAxisMask_1, nAxisMask_2, nAxisMask_3, nAxisMask_4, nAxisMask_5, nAxisMask_6, dTarget_1, dTarget_2, dTarget_3, dTarget_4, dTarget_5, dTarget_6, sTcpName, sUcsName, dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID);
```

3.10.4. HRIF_IsMotionDone

3.10.4.1. 描述：判断机器人是否处于运动状态。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

✓ 输出变量

返回值	名称	数据类型	有效范围	内容
bDone	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

3.10.4.2. 示例

```
// 判断机器人是否处于运动状态

bool bDone = false;
int nRet= hr_api.HRIF_IsMotionDone(0,0,ref bDone);
```

3.10.5. HRIF_IsBlendingDone

3.10.5.1. 描述：判断路点是否运动完成

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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

* 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

返回值	名称	数据类型	有效范围	内容
bDone	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

```
// 判断路点是否运动完成
bool bDone = false;
int nRet= hr_api.HRIF_IsBlendingDone(0,0, ref bDone);
```

3.10.6. HRIF_WayPointEx

3.10.6.1. 描述：路点运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
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	double	-	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	double	-	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	double	-	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	double	-	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	double	-	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	double	-	运动最大加速度，关节运动时单位[°/s^2]，空间运动时 XYZ 单位[mm/s^2]，Rx，Ry，Rz 单位[°/s^2]
dRadius	过渡半径	double	-	过渡半径，单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	是否使用关节角度作为目标点，如果 nMoveType= 0 时，则 nIsUseJoint 有效： 0：不使用关节角度 1：使用关节角度

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

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

✓ 返回值

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

3.10.6.2. 示例

```
// 定义运动类型
int nMoveType = 1;
// 定义空间目标位置
double dX = 420; double dY = 0; double dZ = 445;
double dRx = 180; double dRy = 0; double dRz = 180;
// 定义关节目标位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 70;
double dJ4 = 0; double dJ5 = 90; double dJ6 = 0;
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 定义用户坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 定义运动速度
```

```
double dVelocity = 50;
// 定义运动加速度
double dAcc = 50;
// 定义过渡半径
double dRadius = 50;
// 定义是否使用关节角度
int nIsUseJoint = 0;
// 定义是否使用检测 DI 停止
int nIsSeek = 0;
// 定义检测的 DI 索引
int nIOBit = 0;
// 定义检测的 DI 状态
int nIOState = 0;
// 定义路点 ID
string strCmdID = "0";
// 执行路点运动
int nRet= hr_api.HRIF_WayPointEx(0,0, nMoveType , dX, dY, dZ, dRx, dRy, dRz,
dJ1, dJ2, dJ3, dJ4, dJ5, dJ6,dTcp_X, dTcp_Y, dTcp_Z, dTcp_Rx, dTcp_Ry, dTcp_Rz,
dUcs_X, dUcs_Y, dUcs_Z, dUcs_Rx, dUcs_Ry, dUcs_Rz,dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit,
nIOState, strCmdID);
```

3.10.7. HRIF_WayPoint

3.10.7.1. 描述：路点运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
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	double	-	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	double	-	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	double	-	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	double	-	运动最大加速度，关节运动时单位[°/s²]，空间运动时 XYZ 单位[mm/s²]，Rx，Ry，Rz 单位[°/s²]
dRadius	过渡半径	double	-	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
int nMoveType = 0;
// 定义空间目标位置
double dX = 420; double dY = 0; double dZ = 445;
double dRx = 180; double dRy = 0; double dRz = 180;
// 定义关节目标位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 80;
double dJ4 = 0; double dJ5 = 90; double dJ6 = 0;
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 定义运动速度
double dVelocity = 50;
// 定义运动加速度
double dAcc = 50;
// 定义过渡半径
double dRadius = 50;
// 定义是否使用关节角度
int nIsUseJoint = 0;
// 定义是否使用检测 DI 停止
int nIsSeek = 0;
// 定义检测的 DI 索引
int nIOBit = 0;
// 定义检测的 DI 状态
int nIOState = 0;
// 定义路点 ID
string strCmdID = "0";
```

```
// 执行路点运动
int nRet= hr_api.HRIF_WayPoint(0,0, nMoveType , dX, dY, dZ, dRx, dRy, dRz, dJ1, dJ2, dJ3, dJ4, dJ5, dJ6,
sTcpName , sUcsName, dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID);
```

3.10.8. HRIF_WayPoint2

3.10.8.1. 描述：路点运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nMoveType	运动类型	int	0/1	0：关节运动 1：直线运动 2：圆弧运动
dEndPos_X-dEndPos_Rz	空间目标位置	double	-	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	double	-	nMoveType= 0：无效 nMoveType= 1：无效 nMoveType= 2：做为圆弧的经过位置 dAuxPosX_X：X 坐标，单位[mm] dAuxPosX _Y：Y 坐标，单位[mm] dAuxPosX _Z：Z 坐标，单位[mm] dAuxPosX _Rx：Rx 坐标，单位[°] dAuxPosX _Ry：Ry 坐标，单位[°]

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

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

* 注： HRIF_WayPoint2 支持直线与圆弧过渡不减速功能； HRIF_WayPoint 仅支持直线与直线过渡不减速。

✓ 返回值

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

3.10.8.2. 示例

```
// 定义运动类型
int nMoveType = 0;
// 定义空间目标位置
double dEndPos_X = 420; double dEndPos_Y = 0; double dEndPos_Z = 445;
double dEndPos_Rx = 180; double dEndPos_Ry = 0; double dEndPos_Rz = 180;
// 定义空间目标位置
double dAuxPos_X = 420; double dAuxPos_Y = 0; double dAuxPos_Z = 445;
double dAuxPos_Rx = 180; double dAuxPos_Ry = 0; double dAuxPos_Rz = 180;
// 定义关节目标位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 90;
double dJ4 = 0; double dJ5 = 90; double dJ6 = 0;
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 定义运动速度
```

```
double dVelocity = 50;
// 定义运动加速度
double dAcc = 50;
// 定义过渡半径
double dRadius = 50;
// 定义是否使用关节角度
int nIsUseJoint = 1;
// 定义是否使用检测 DI 停止
int nIsSeek = 0;
// 定义检测的 DI 索引
int nIOBit = 0;
// 定义检测的 DI 状态
int nIOState = 0;
// 定义路点 ID
string strCmdID = "0";
// 执行路点运动
int nRet= hr_api.HRIF_WayPoint2(0,0, nMoveType ,dEndPos_X, dEndPos_Y, dEndPos_Z, dEndPos_Rx, dEndPos_Ry, dEndPos_Rz,dAuxPos_X, dAuxPos_Y, dAuxPos_Z, dAuxPos_Rx, dAuxPos_Ry, dAuxPos_Rz,dJ1, dJ2, dJ3, dJ4, dJ5, dJ6,sTcpName , sUcsName, dVelocity, dAcc, dRadius, nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID);
```

3.10.9. HRIF_MoveJ

3.10.9.1. 描述：关节运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dX -dRz	空间目标位置	double	-	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	double	-	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	double	-	运动最大速度，单位[°/s]
dAcc	加速度	double	-	运动最大加速度，单位[°/s²]
dRadius	过渡半径	double	-	过渡半径，单位[mm]
nIsUseJoint	是否使用关节坐标	int	0/1	是否使用关节角度作为目标点，如果 nMoveType= 0 时，则 nIsUseJoint 有效： 0：不使用关节角度 1：使用关节角度
nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1，则开启检测 DI 停止，路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态= nIOState 时，机器人停止运动，否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引，nIsSeek= 0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态，，nIsSeek= 0 时无效
strCmdID	命令 ID	string	-	当前路点 ID ，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

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

3.10.9.2. 示例

```
// 定义空间目标位置
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 0; double dRy = 0; double dRz = 0;
// 定义关节目标位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 90;
double dJ4 = 0; double dJ5 = 90; double dJ6 = 0;
// 定义工具坐标变量
```

```
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 定义运动速度
double dVelocity = 50;
// 定义运动加速度
double dAcc = 50;
// 定义过渡半径
double dRadius = 50;
// 定义是否使用关节角度
int nIsUseJoint = 1;
// 定义是否使用检测 DI 停止
int nIsSeek = 0;
// 定义检测的 DI 索引
int nIOBit = 0;
// 定义检测的 DI 状态
int nIOState = 0;
// 定义路点 ID
string strCmdID = "0";
// 执行路点运动
int nRet= hr_api.HRIF_MoveJ(0,0, dX, dY, dZ, dRx, dRy, dRz,dJ1, dJ2, dJ3, dJ4, dJ5, dJ6,sTcpName,sUcsName, dVelocity,dAcc,dRadius,nIsUseJoint, nIsSeek, nIOBit, nIOState, strCmdID);
```

3.10.10. HRIF_MoveL

3.10.10.1.描述：直线轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dX -dRz	空间目标位置	double	-	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	double	-	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	double	-	运动最大速度，X，Y，Z 单位[mm/s]，Rx，Ry，Rz 单位[°/s]

dAcc	加速度	double	-	运动最大加速度，X，Y，Z 单位[mm/s²]，Rx，Ry，Rz 单位[°/s²]
dRadius	过渡半径	double	-	过渡半径，单位[mm]
nIsSeek	是否检测 DI 停止	int	0/1	如果 nIsSeek 为 1，则开启检测 DI 停止，路点运动过程中如果电箱的 nIOBit 位索引的 DI 的状态= nIOState 时，机器人停止运动，否则运动到目标点完成运动
nIOBit	检测的 DI 索引	int	0~7	电箱对应 DI 索引，nIsSeek= 0 时无效
nIOState	检测的 DI 状态	int	0/1	检测的 DI 状态，，nIsSeek= 0 时无效
strCmdID	命令 ID	string	-	当前路点 ID ，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

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

3.10.10.2.示例

```
// 定义空间目标位置
double dX = 420; double dY = 0; double dZ = 445;
double dRx = 180; double dRy = 0; double dRz = 180;
// 定义关节目标位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 90;
double dJ4 = 0; double dJ5 = 90; double dJ6 = 0;
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 定义运动速度
double dVelocity = 50;
// 定义运动加速度
double dAcc = 50;
```

```
// 定义过渡半径
double dRadius = 50;
// 定义是否使用检测 DI 停止
int nIsSeek = 0;
// 定义检测的 DI 索引
int nIOBit = 0;
// 定义检测的 DI 状态
int nIOState = 0;
// 定义路点 ID
string strCmdID = "0";
// 执行路点运动
int nRet= hr_api.HRIF_MoveL(0,0, dX, dY, dZ, dRx, dRy, dRz,dJ1, dJ2, dJ3, dJ4, dJ5, dJ6,sTcpName , sUcsName, dVelocity, dAcc, dRadius,nIsSeek, nIOBit, nIOState, strCmdID);
```

3.10.11. HRIF_MoveC

3.10.11.1.描述：圆弧轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dStartPos_X-dStartPos_Rz	圆弧起始点位置	double	-	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[°]
dAuxPos_X-dAuxPos_Rz	圆弧经过点位置	double	-	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[°]
dEndPos_X-dEndPos_Rz	圆弧结束点位置	double	-	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：不固定姿态 1：固定姿态
nMoveCType	圆弧类型	int	0/1	0：整圆 1：圆弧
dRadLen	弧长	double	-	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	double	-	运动最大速度，X，Y，Z 单位[mm/s]，Rx，Ry，Rz 单位[°/s]
dAcc	加速度	double	-	运动最大加速度，X，Y，Z 单位[mm/s^2]，Rx，Ry，Rz 单位[$°/s^2$]
dRadius	过渡半径	double	-	过渡半径，单位[mm]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，当 nIsUseJoint= 1 时无效，可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应，当 nIsUseJoint= 1 时无效，可使用默认名称"Base"
strCmdID	命令 ID	string	-	当前路点 ID ，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

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

3.10.11.2.示例

```
// 圆弧起始点位置
double dStartPos_X = 420; double dStartPos_Y = 0; double dStartPos_Z = 445;
double dStartPos_Rx = 180; double dStartPos_Ry = 0; double dStartPos_Rz = 180;
// 圆弧经过点位置
double dAuxPos_X = 420; double dAuxPos_Y = 50; double dAuxPos_Z = 445;
double dAuxPos_Rx = 180; double dAuxPos_Ry = 0; double dAuxPos_Rz = 180;
// 圆弧结束点位置
double dEndPos_X = 470; double dEndPos_Y = 0; double dEndPos_Z = 445;
double dEndPos_Rx = 180; double dEndPos_Ry = 0; double dEndPos_Rz = 180;
// 是否固定姿态
int nFixedPosure = 0;
// 圆弧类型
int nMoveCType = 0;
// 整圆圈数
double dRadLen = 1;
// 定义运动速度
double dVelocity = 50;
// 定义运动加速度
double dAcc = 50;
// 定义过渡半径
double dRadius = 50;
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 定义路点 ID
string strCmdID = "0";
// 执行路点运动
int nRet= hr_api.HRIF_MoveC(0,0,dStartPos_X, dStartPos_Y, dStartPos_Z, dStartPos_Rx, dStartPos_Ry, dStartPos_Rz, dAuxPos_X, dAuxPos_Y, dAuxPos_Z, dAuxPos_Rx, dAuxPos_Ry, dAuxPos_Rz, dEndPos_X, dEndPos_Y, dEndPos_Z, dEndPos_Rx, dEndPos_Ry, dEndPos_Rz, nFixedPosure, nMoveCType, dRadLen, dVelocity,
```

dAcc, dRadius,sTcpName , sUcsName, strCmdID);

3.10.12. HRIF_MoveZ

3.10.12.1.描述：Z 型轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dStartPos_X-dStartPos_Rz	Z 型起始点位置	double	-	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_X-dEndPos_Rz	Z 型结束点位置	double	-	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_X-dPlanePos_Rz	轨迹确定平面点位置	double	-	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	double	-	运动最大速度，X，Y，Z 单位[mm/s]， Rx，Ry，Rz 单位[°/s]

dAcc	加速度	double	-	运动最大加速度，X，Y，Z 单位[mm/s²]，Rx，Ry，Rz 单位[°/s²]
dW ID th	宽度	double	-	轨迹宽度
dDensity	密度	double	-	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 ms	int	-	正向转折点等待时间，单位[ms]
nNegaTime	负向转折点等待时间 ms	int	-	负向转折点等待时间，单位[ms]
dRadius	过渡半径	double	-	过渡半径，单位[mm]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，当 nIsUseJoint= 1 时无效，可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应，当 nIsUseJoint= 1 时无效，可使用默认名称"Base"
strCmdID	命令 ID	string	-	当前路点 ID ，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

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

3.10.12.2.示例

```
// 起始点位置
double dStartPos_X = 420; double dStartPos_Y = 0; double dStartPos_Z = 445;
double dStartPos_Rx = 180; double dStartPos_Ry = 0; double dStartPos_Rz = 180;
// 结束点位置
double dEndPos_X = 420; double dEndPos_Y = 100; double dEndPos_Z = 445;
double dEndPos_Rx = 180; double dEndPos_Ry = 0; double dEndPos_Rz = 180;
// 确定轨迹平面点位置
double dPlanePos_X = 470; double dPlanePos_Y = 50; double dPlanePos_Z = 445;
double dPlanePos_Rx = 180; double dPlanePos_Ry = 0; double dPlanePos_Rz = 180;
// 定义运动速度
double dVelocity = 50;
// 定义运动加速度
double dAcc = 2500;
// 宽度
double dWidth = 50;
// 密度
double dDensity = 10;
// 使用密度
int nEnableDensity = 1;
// 使用平面点
int nEnablePlane = 1;
// 是否在转折点等待.不等待
int nEnableWaiTime = 0;
// 正向转折点等待时间
int nPosiTime = 0;
// 负向转折点等待时间
int nNegaTime = 0;
```

```
// 定义过渡半径
double dRadius = 5;
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 定义路点 ID
string strCmdID = "0";
// 执行路点运动
int nRet= hr_api.HRIF_MoveZ(0,0, dStartPos_X, dStartPos_Y, dStartPos_Z, dStartPos_Rx, dStartPos_Ry, dStartPos_Rz, dEndPos_X, dEndPos_Y, dEndPos_Z, dEndPos_Rx, dEndPos_Ry, dEndPos_Rz, dPlanePos_X, dPlanePos_Y, dPlanePos_Z, dPlanePos_Rx, dPlanePos_Ry, dPlanePos_Rz, dVelocity, dAcc, dWidth, dDensity, nEnableDensity, nEnablePlane, nEnableWaiTime, nPosiTime, nNegaTime, dRadius, sTcpName, sUcsName, strCmdID);
```

3.10.13. HRIF_MoveE

3.10.13.1. 描述：椭圆型轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
dP1_X- dP1_Rz	示教位置 1	double	-	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_X- dP2_Rz	示教位置 2	double	-	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_X- dP1_Rz	示教位置 3	double	-	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_X- dP4_Rz	示教位置 4	double	-	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_X- dP5_Rz	示教位置 5	double	-	示教位置 5： dP5_X：X 坐标，单位[mm] dP5_Y：Y 坐标，单位[mm] dP5_Z：Z 坐标，单位[mm] dP5_Rx：Rx 坐标，单位[°] dP5_Ry：Ry 坐标，单位[°] dP5_Rz：Rz 坐标，单位[°]
nOrientMode	运动模式	int	0/1	0：椭圆圆弧 1：整个椭圆
nMoveType	运动类型	int	0/1	0:不使用固定姿态 1:使用固定姿态
dArcLength	弧长	double	-	弧长，单位[°]
dVelocity	速度	double	-	运动最大速度，X，Y，Z 单位[mm/s]， Rx，Ry，Rz 单位[°/s]
dAcc	加速度	double	-	运动最大加速度，X，Y，Z 单位[mm/s^2]， Rx，Ry，Rz 单位[$°/s^2$]
dRadius	过渡半径	double	-	过渡半径，单位[mm]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，当 nIsUseJoint= 1 时无效，可使用默认名称"TCP"
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应，当 nIsUseJoint= 1 时无效，可使用默认名称"Base"
strCmdID	命令 ID	string	-	当前路点 ID ，可以自定义，也可以按顺序设置为"1"，"2"，"3"

✓ 返回值

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

3.10.13.2.示例

```
// 示教点 1
double[] dP1 = { 420,0,445,180,0,180 };
// 示教点 2
double[] dP2 = { 460,0,445,180,0,180 };
// 示教点 3
double[] dP3 = { 480,10,445,180,0,180 };
// 示教点 4
double[] dP4 = { 460,20,445,180,0,180 };
// 示教点 5
double[] dP5 = { 420,20,445,180,0,180 };
// 运动模式
double nOrientMode = 0;
// 运动类型
double nMoveType = 1;
// 弧长
int dArcLength = 360;
// 定义运动速度
double dVelocity = 50;
// 定义运动加速度
double dAcc = 2500;
// 定义过渡半径
double dRadius = 5;
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
```

```
string sUcsName = "Base";
// 定义路点 ID
string strCmdID = "0";
// 执行椭圆运动
int nRet= hr_api.HRIF_MoveE(0,0,dP1[0],dP1[1],dP1[2],dP1[3],dP1[4],dP1[5],
    dP2[0],dP2[1],dP2[2],dP2[3],dP2[4],dP2[5], dP3[0],dP3[1],dP3[2],dP3[3],dP3[4],dP3[5],
    dP4[0],dP4[1],dP4[2],dP4[3],dP4[4],dP4[5], dP5[0],dP5[1],dP5[2],dP5[3],dP5[4],dP5[5],
    nOrientMode,nMoveType,dArcLength,dVelocity,dAcc,dRadius, sTcpName, sUcsName, strCmdID);
```

3.10.14. HRIF_MoveS

3.10.14.1.描述：阿基米德螺旋线运动，初始半径为固定 1mm 。

✓ 输入变量

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

✓ 返回值

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

3.10.14.2.示例

```
// 定义增量半径
double dSpiralIncrement = 1;
// 定义结束半径
double dSpiralDiameter = 5;
```

```
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 定义运动速度
double dVelocity = 50;
// 定义运动加速度
double dAcc = 50;
// 定义过渡半径
double dRadius = 50;
// 定义路点 ID
string strCmdID = "0";
// 执行螺旋轨迹运动
int nRet = hr_api.HRIF_MoveS(0, 0, dSpiralIncrement, dSpiralDiameter, dVelocity, dAcc, dRadius, sTcpName, sUcsName, strCmdID);
```

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	unsigned 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
sTrackName	轨迹名称	string	-	轨迹名称
dSpeedRatio	轨迹运动速度比	double	0~1	轨迹运动速度比
dRadius	过渡半径	double	>= 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
string sTrackName = "Path1";
// 速度比
double dSpeedRatio = 0.5;
// 过渡半径
double dRadius = 2;
// 初始化关节连续轨迹运动
int nRet= hr_api.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	unsigned 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
sTrackName	轨迹名称	string	-	轨迹名称
dJ1 -dJ6	关节点位	double	-	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
string sTrackName = "Path1";
// 目标关节位置
double dJ1 = 0; double dJ2 = 0; double dJ3 = 0;
double dJ4 = 0; double dJ5 = 0; double dJ6 = 0;
// 下发关节点位
int nRet= hr_api.HRIF_PushMovePathJ (0,0, sTrackName, dJ1, dJ2, dJ3, dJ4, dJ5, dJ6);
```

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	unsigned 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
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
string sTrackName = "Path1";
// 下发完成，开始计算轨迹
int nRet= hr_api.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	unsigned 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
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
string sTrackName = "Path1";
// 运动轨迹
int nRet= hr_api.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	unsigned 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
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
nState	轨迹状态	int	0~5	当前轨迹状态： 0：轨迹未开始示教，调用 HRIF_StartPushMovePathJ 后切换到此状态 1：轨迹示教中，调用 HRIF_PushMovePathJ 后切换到此状态 2：轨迹计算中，调用 HRIF_EndPushMovePathJ 后切换到此状态 3：轨迹计算完成，计算完成后切换到此状态，可以开始运动。 4：结束示教 5：计算错误，需要检查点位是否合理，然后修正后重新示教。

✓ 返回值

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

3.11.5.2. 示例

```
// 轨迹名称
string sTrackName = "Path1";
// 读取到的当前轨迹状态
int nState = 0;
// 读取轨迹状态
int nRet= hr_api.HRIF_ReadMovePathJState(0,0, sTrackName, ref nState);
```

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	unsigned 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
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
string sTrackName = "Path1";
// 更新的轨迹名称
string sTrackNewName = "Path2";
// 重命名轨迹名称
int nRet= hr_api.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	unsigned 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
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
string sTrackName = "Path1";
// 删除轨迹
int nRet= hr_api.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	unsigned 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

* 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
dProcess	轨迹运行进度	double	0~1	当前轨迹运动进度，当前运动进度(0,0-1),>0.999999 表示运动完成
nIndex	点位索引	int	>0 的整型值	当前轨迹运动到哪一个点索引

✓ 返回值

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

3.11.8.2. 示例

```
// 轨迹运行进度
double dProcess = 0;
// 点位索引
int nIndex = 0;
// 读取轨迹状态
int nRet= hr_api.HRIF_ReadTrackProcess(0,0, ref dProcess, ref nIndex);
```

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	unsigned 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
sTrackName	轨迹名称	string	-	轨迹名称，目前空间轨迹运动的轨迹名称没有作用可以任意定义，同一轨迹运行需要同执行 HRIF_InitMovePathL，HRIF_PushMovePaths，HRIF_EMovePathL 调用 HRIF_MovePathL 后会计算完轨迹后直接开始运动，计算时间 2-4s 左右，根据实际轨迹大小确定
dVelocity	轨迹运动速度	double	-	轨迹运动速度，单位[mm/s]
dAcc	轨迹运动加速度	double	-	轨迹运动加速度，单位[mm/s^2]
dJerk	轨迹运动加加速度	double	-	轨迹运动加加速度，单位[mm/s^3]
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
```

```
string sTrackName = "Path1";
// 定义运动速度
double dVelocity = 10;
// 定义运动加速度
double dAcc = 2500;
// 定义运动加加速度
double dJerk = 100;
// 定义工具坐标变量
string sTcpName = "TCP";
// 定义用户坐标变量
string sUcsName = "Base";
// 初始化关节连续轨迹运动
int nRet= hr_api.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	unsigned 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
sTrackName	轨迹名称	string	-	轨迹名称
dX -dRz	空间点位	double	-	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： R y 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
string sTrackName = "Path1";
// 定义空间目标位置
double dX = 420; double dY = 0; double dZ = 445;
double dRx = 180; double dRy = 0; double dRz = 180;
// 下发空间目标点位
int nRet= hr_api.HRIF_PushMovePathL(0,0, sTrackName, dX, dY, dZ, dRx, dRy, dRz);
```

3.11.11. HRIF_PushMovePaths

3.11.11.1.描述：批量下发轨迹点位，调用一次可下发多个点位数据。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
sTrackName	轨迹名称	string	-	轨迹名称
nMoveType	点位类型	int	0/1	0：下发关节点位，与 HRIF_MovePathJ 共用 1：下发空间点位，与 HRIF_MovePathL 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	string	-	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
string sTrackName = "Path1";
// 运动类型-MovePathL
int nMoveType = 1;
// 点位数量
int nPointsSize = 6;
string 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
int nRet= hr_api.HRIF_PushMovePaths(0,0, sTrackName, nMoveType, nPointsSize, nPointsSize);
// 轨迹名称
string sTrackName = "Path2";
// 运动类型-MovePathJ
```

```
int nMoveType = 0;
// 点位数量
int nPointsSize = 9;
string sPoints ="420,0,445,180,0,180,420,10,445,180,0,180,420,20,445,180,0,180,420,30,445,180,0,180,420,40,445,180,0,180,420,50,445,180,0,180,420,60,445,180,0,180,420,70,445,180,0,180,420,80,445,180,0,180";
// 下发关节点位
int nRet= hr_api.HRIF_PushMovePaths(0,0, sTrackName, nMoveType, nPointsSize, sPoints);
```

3.11.12. HRIF_MovePathL

3.11.12.1.描述：执行空间坐标轨迹运动。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
sTrackName	轨迹名称	string	-	轨迹名称

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

✓ 返回值

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

3.11.12.2.示例

```
// 轨迹名称
string sTrackName = "Path1";
// 开始空间连续轨迹运动
int nRet= hr_api.HRIF_MovePathL(0,0, sTrackName);
```

3.11.13. HRIF_SetMovePathOverride

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

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
MovePathOverride	设置速度比	double	1~100	需要设置的速度比

✓ 返回值

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

3.11.13.2.示例

```
//速度比
double dOverride = 0.1;
//设置速度比
int nRet= hr_api.HRIF_SetMovePathOverride(0,0, dOverride);
```

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	unsigned 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
dServoTime	更新周期	double	>0	固定更新的周期时间，单位[s]
dLookaheadTime	前瞻时间	double	>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
double dServoTime = 0.2;
// 前瞻时间
double dLookaheadTime = 0.02;
// 启动机器人在线控制
int nRet= hr_api.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	unsigned 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
dJ1 -dJ6	关节点位	double	-	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
double dJ1 = 0; double dJ2 = 0; double dJ3 = 90;
double dJ4 = 0; double dJ5 = 90; double dJ6 = 0;
// 在线关节位置命令控制
int nRet= hr_api.HRIF_PushServoJ(0,0, dJ1, dJ2, dJ3, dJ4, dJ5, dJ6);
```

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	unsigned 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
dX -dRz	空间点位	double	-	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[°]
dTcp_X- dTcp_Rz	工具坐标	double	-	目标位置对应的工具坐标： 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 ： R z 坐标，单位[°]
dUcs_X- dUcs_Rz	用户坐标	double	-	目标位置对应的用户坐标： 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[°]

* 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
double dX = 420; double dY = 0; double dZ = 445;
double dRx = 180; double dRy = 0; double dRz = 180;
// 定义工具坐标变量
double dTcp_X = 0; double dTcp_Y = 0; double dTcp_Z = 0;
double dTcp_Rx = 0; double dTcp_Ry = 0; double dTcp_Rz = 0;
// 定义用户坐标变量
double dUcs_X = 0; double dUcs_Y = 0; double dUcs_Z = 0;
double dUcs_Rx = 0; double dUcs_Ry = 0; double dUcs_Rz = 0;
// 在线空间位置命令控制
int nRet= hr_api.HRIF_PushServoP(0,0, dX, dY, dZ, dRx, dRy, dRz,dTcp_X, dTcp_Y, dTcp_Z, dTcp_Rx, dTcp_Ry, dTcp_Rz,dUcs_X, dUcs_Y, dUcs_Z, dUcs_Rx, dUcs_Ry, dUcs_Rz);
```

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	unsigned 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
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	double	>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	double	>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	double	>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
double nState = 1;
double dDistance = 100;
// 相对跟踪的运动的探寻速度
double dAwayVelocity = 50;
double dBackVelocity = 50;
// 设置相对跟踪运动控制参数并开启相对跟踪运动
```

int nRet= hr_api.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	unsigned 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
dK	传感器计算参数	double	-	计算公式 y = dK * x + dB
dB	传感器计算参数	double	-	计算公式 y = dK * x + dB
dMaxLimit	激光传感器检测距离最大值	double	-	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	double	-	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
double dK= -32;
// 传感器计算参数
double dB= 280;
// 激光传感器检测距离最大值
double dMaxLimit = 130;
// 激光传感器检测距离最小值
double dMinLimit = 65;
// 设置跟踪状态初始化参数
int nRet= hr_api.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	unsigned 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
dX - dRz	跟踪探寻方向	double	>-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
double dX = 0; double dY = 0; double dZ = 0;
double dRx = 180; double dRy = 0; double dRz = 0;
// 设置跟踪方向
int nRet= hr_api.HRIF_SetMoveTraceUcs(0,0, dX, dY, dZ, dRx, dRy, dRz);
```

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	unsigned 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
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
int nState = 1;
// 开启传送带跟踪
int nRet= hr_api.HRIF_SetTrackingState(0,0, nState);
```

3.14. 其他指令

3.14.1. HRIF_HRAppCmd

3.14.1.1. 描述：执行插件 App 命令。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
sCmdName	命令名称	string	0/1	命令名称
sParams	参数列表	string	-	参数列表

✓ 返回值

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

3.14.1.2. 示例

```
//插件名称
string HRAppName = "hr_gri_plugins";
// 设置命令名称
string sCmdName = "";
// 设置参数
string sParams = "GetNumOFGripper";
// 开启传送带跟踪
int nRet = hr_api.HRIF_HRAppCmd(0, 0,HRAppName, HRAppCmdName, sParams);
```

3.14.2. HRIF_WriteEndHoldingRegisters

3.14.2.1. 描述：写末端连接的 Modbus 从站寄存器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nSlave ID	从站 ID	int	0/1	从站 ID
nFunction	功能码	int	>0	功能码： 0x01-读线圈寄存器 0x02-读线离散输入寄存器 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
vecData	寄存器数据	vector<int>	-	寄存器数据

* 注：末端为 EtherCAT 总线版本 IO 时有效。

✓ 返回值

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

3.14.2.2. 示例

```
// 设置从站 ID=1
int nSlaveID = 1;
// 设置功能码
```

```
int nFunction = 3;
// 设置寄存器起始地址
int nRegAddr = 5004;
// 设置寄存器数量
int nRegCount = 2;
// 设置寄存器数据
List<int> vecData = new List<int>();
vecData.Add(5);
vecData.Add(6);
// 写寄存器数据
int nRet= hr_api.HRIF_WriteEndHoldingRegisters(0,0, nSlaveID, nFunction, nRegAddr, nRegCount, vecData);
```

3.14.3. HRIF_ReadEndHoldingRegisters

3.14.3.1. 描述：读末端连接的 Modbus 从站寄存器。

✓ 输入变量

输入变量	名称	数据类型	有效范围	内容
boxID	电箱 ID	unsigned 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
nSlave ID	从站 ID	int	0/1	从站 ID
nFunction	功能码	int	>0	功能码： 0x01-读线圈寄存器 0x02-读线离散输入寄存器 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

输入变量	名称	数据类型	有效范围	内容
vecData	寄存器数据	vector<int>	-	寄存器数据

✓ 返回值

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

3.14.3.2. 示例

```
// 设置从站 ID=1
```

```
int nSlaveID = 1;
// 设置功能码
int nFunction = 3;
// 设置寄存器起始地址
int nRegAddr = 5004;
// 设置寄存器数量
int nRegCount = 2;
// 读取寄存器数据
List<int> vecData = new List<int>();
int nData1 = 0;
int nData2 = 0;
// 写寄存器数据
int nRet= hr_api.HRIF_ReadEndHoldingRegisters(0,0, nSlaveID, nFunction, nRegAddr, nRegCount, ref vecData);
if (nRet != 0)
{
Console.WriteLine(nRet);
}
if (vecData.Count() == 2)
{
nData1 = vecData[0];
nData2 = vecData[1];
}
```

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 Python SDK 使用请参阅《HanRobot Library Python》。