



HAN'S ROBOT  
大族机器人

# HansRobot V5

## 控制通信协议

(Elfin 系列机器人通用)



## 修订记录

文件编号	HR-TCP-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 SetToolMotion 接口

## 目录

<b>第一章 概述 .....</b>	<b>1</b>
1.1. 编写目的 .....	1
1.2. 使用须知 .....	1
1.3. 数据类型 .....	1
1.4. 名词解析 .....	2
<b>第二章 消息格式 .....</b>	<b>3</b>
2.1. 通信协议格式 .....	3
2.2. 返回格式 .....	3
2.2.1. 成功返回 .....	3
2.2.2. 失败返回 .....	3
<b>第三章 服务端 .....</b>	<b>4</b>
3.1. 连接方式 .....	4
3.1.1. 作为客户端 .....	4
3.1.2. 作为服务器 .....	4
3.1.3. 配置文件 .....	5
3.1.4. 连接示例 .....	5
3.2. 服务端口表 .....	7
<b>第四章 通信协议接口 .....</b>	<b>8</b>
4.1. 初始化 .....	8
4.1.1. OSCmd .....	8
4.1.2. ConnectToBox .....	9
4.1.3. Electrify .....	10
4.1.4. BlackOut .....	11
4.1.5. StartMaster .....	12
4.1.6. CloseMaster .....	13
4.1.7. IsSimulation .....	14
4.1.8. ReadControllerState .....	15
4.1.9. ReadRobotModel .....	16
4.2. 轴组控制指令 .....	17
4.2.1. GrpEnable .....	17

4.2.2. GrpDisable .....	18
4.2.3. GrpReset .....	19
4.2.4. GrpStop .....	20
4.2.5. GrpInterrupt .....	21
4.2.6. GrpContinue .....	22
4.2.7. GrpCloseFreeDriver .....	23
4.2.8. GrpOpenFreeDriver .....	24
4.3. 脚本控制指令 .....	25
4.3.1. RunFunc .....	25
4.3.2. StartScript .....	26
4.3.3. StopScript .....	27
4.3.4. PauseScript .....	28
4.3.5. ContinueScript .....	29
4.4. 电箱控制指令 .....	30
4.4.1. ReadBoxInfo .....	30
4.4.2. ReadBoxCI .....	32
4.4.3. ReadBoxDI .....	33
4.4.4. ReadBoxCO .....	34
4.4.5. ReadBoxDO .....	35
4.4.6. ReadBoxAI .....	36
4.4.7. ReadBoxAO .....	37
4.4.8. SetBoxCO .....	38
4.4.9. SetBoxDO .....	39
4.4.10. SetBoxAOMode .....	40
4.4.11. SetBoxAO .....	41
4.4.12. SetEndDO .....	42
4.4.13. ReadEI .....	43
4.4.14. ReadEO .....	44
4.4.15. ReadEAI .....	45
4.5. 状态读取与设置指令 .....	46
4.5.1. SetOverride .....	46
4.5.2. SetToolMotion .....	47
4.5.3. SetPayload .....	48
4.5.4. SetJointMaxVel .....	49
4.5.5. SetJointMaxAcc .....	50
4.5.6. SetLinearMaxVel .....	51
4.5.7. SetLinearMaxAcc .....	52
4.5.8. SetMaxAccsRange .....	53
4.5.9. SetMaxPcsRange .....	55
4.5.10. ReadJointMaxVel .....	56

4.5.11. ReadJointMaxAcc .....	57
4.5.12. ReadJointMaxJerk .....	58
4.5.13. ReadLinearMaxVel .....	59
4.5.14. ReadEmergencyInfo .....	60
4.5.15. ReadRobotState .....	61
4.5.16. ReadCurWayPointID .....	63
4.5.17. ReadAxisErrorCode .....	64
4.5.18. ReadCurFSM .....	65
4.6. 位置、速度、电流读取指令 .....	66
4.6.1. ReadActPos .....	66
4.6.2. ReadCmdPos .....	68
4.6.3. ReadCmdJointVel .....	70
4.6.4. ReadActJointVel .....	71
4.6.5. ReadCmdTcpVel .....	72
4.6.6. ReadActTcpVel .....	73
4.6.7. ReadCmdJointCur .....	74
4.6.8. ReadActJointCur .....	75
4.6.9. ReadTcpVelocity .....	76
4.7. 坐标转换计算指令 .....	77
4.7.1. Quaternion2RPY .....	77
4.7.2. RPY2Quaternion .....	78
4.7.3. GetInverseKin(PCS2ACS) .....	79
4.7.4. GetForwardKin(ACS2PCS) .....	81
4.7.5. Base2UcsTcp .....	83
4.7.6. UcsTcp2Base .....	85
4.7.7. PoseAdd .....	87
4.7.8. PoseSub .....	89
4.7.9. PoseTrans .....	91
4.7.10. PoseInverse .....	93
4.7.11. PoseDist .....	95
4.7.12. PoseInterpolate .....	97
4.7.13. PoseDefdFrame .....	99
4.8. 工具坐标与用户坐标读写指令 .....	101
4.8.1. SetCurTCP .....	101
4.8.2. SetCurUCS .....	102
4.8.3. ReadCurTCP .....	103
4.8.4. ReadCurUCS .....	104
4.8.5. SetTCPByName .....	105
4.8.6. SetUCSByName .....	106
4.8.7. ReadTCPByName .....	107

---

4.8.8. ReadUCSByName .....	108
4.9. 力控控制指令 .....	109
4.9.1. SetForceControlState .....	109
4.9.2. ReadFTControlState .....	110
4.9.3. SetForceToolCoordinateMotion.....	111
4.9.4. GrpFCInterrupt .....	112
4.9.5. GrpFCContinue.....	113
4.9.6. SetForceZero.....	114
4.9.7. HRSetMaxSearchVelocities .....	115
4.9.8. HRSetControlFreedom .....	116
4.9.9. HRSetForceControlStrategy .....	117
4.9.10. SetFTPPosition.....	118
4.9.11. HRSetPIDControlParams.....	119
4.9.12. HRSetMassParams.....	120
4.9.13. HRSetDampParams .....	121
4.9.14. HRSetStiffParams .....	122
4.9.15. HRSetControlGoal.....	123
4.9.16. HRSetForceDataLimit .....	124
4.9.17. HRSetForceDistanceLimit.....	126
4.9.18. SetForceFreeDriveMode.....	127
4.9.19. ReadFTCabData.....	128
4.10. 通用运动类控制指令.....	129
4.10.1. MoveRelJ.....	129
4.10.2. MoveRelL .....	130
4.10.3. WayPointRel .....	131
4.10.4. WayPointEx .....	134
4.10.5. WayPoint.....	137
4.10.6. WayPoint2.....	140
4.10.7. MoveJ(弃用).....	143
4.10.8. MoveL(弃用).....	146
4.10.9. MoveC .....	149
4.10.10. MoveZ.....	152
4.10.11. MoveE.....	155
4.10.12. MoveS.....	158
4.11. 连续轨迹运动类控制指令.....	159
4.11.1. StartPushMovePath.....	159
4.11.2. PushMovePathJ.....	160
4.11.3. EndPushMovePath.....	161
4.11.4. MovePath .....	162

---

4.11.5. ReadMovePathState .....	163
4.11.6. UpdateMovePathName .....	164
4.11.7. DelMovePath .....	165
4.11.8. ReadSoftMotionProcess .....	166
4.11.9. InitMovePathL .....	167
4.11.10. PushMovePathL .....	168
4.11.11. PushMovePaths .....	169
4.11.12. MovePathL .....	170
4.11.13. SetMovePathOverride .....	171
4.12. Servo 运动类控制指令 .....	172
4.12.1. StartServo .....	172
4.12.2. PushServoJ .....	173
4.12.3. PushServoP .....	174
4.13. 相对跟踪运动类控制指令 .....	176
4.13.1. SetMoveTraceParams .....	176
4.13.2. SetMoveTraceInitParams .....	177
4.13.3. SetMoveTraceUcs .....	178
4.13.4. SetTrackingState .....	179
4.14. 其他指令 .....	180
4.14.1. HRAppCmd .....	180
4.14.2. WriteHoldingRegisters .....	181
4.14.3. ReadHoldingRegisters .....	183
<b>第五章 Datasheet 服务 .....</b>	<b>185</b>
5.1. 内容 .....	185
5.2. 包头 .....	188
<b>第六章 控制机器人运动流程—示例 .....</b>	<b>189</b>
6.1. 运动流程图 .....	189
6.2. 示例代码 .....	190
<b>第七章 附录 .....</b>	<b>192</b>
7.1. 参考文件 .....	192

## 第一章 概述

### 1.1.编写目的

为使用大族机器人通信协议进行二次开发的人员提供接口使用说明。

### 1.2.使用须知

1. 大族机器人控制通信协议接口主要提供大族机器人通信协议标准，使用人员依照此协议内容通过 TCP/IP 发送指定消息到大族机器人控制系统（以下简称控制系统），控制系统依据不同的消息内容分别处理，并将处理结果返回发送方。
2. 对于运动类型的消息，控制系统会立即返回命令下发结果（一般返回结果为 0，表示命令成功下发，其他为错误码），然后运动消息发送方需要不断发送 ReadRobotState 消息来获取运动是否完成，一般情况下，只有机器人将当前运动完成，才能够进行下一次运动。
3. 其他类型的消息，控制系统会返回请求结果，具体参见通信协议。
4. 通信控制系统每一次只会处理一个完整的消息，如一次发送多条消息，控制系统只会处理第一条满足协议格式的消息，其他丢弃。如果发送的消息不完整，控制系统会返回命令无效的结果。

### 1.3.数据类型

数据类型	长度	说明
unsigned 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



## 1.4.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

## 第二章 消息格式

### 2.1.通信协议格式

- 消息名称, Param1,Param2,Param3.....Paramn,;
- 消息格式如上所示, 由一个消息名称, 后接参数组成, 每一个字段之间以英文逗号 “,” 相隔, 一个完整的消息以英文逗号和分号 “;” 结束。
- 消息命令与消息应答都是 ASCII 码格式 (字符串形式) 。
- 没有用到的参数赋值为 0.

### 2.2.返回格式

#### 2.2.1.成功返回

- 消息名称,OK,Param1,Param2,Param3.....Paramn,;
- 消息格式如上所示, 由一个消息名称, 成功标志 OK, 后接参数组成, 每一个字段之间以英文逗号 “,” 相隔, 一个完整的消息以英文逗号和分号 “;” 结束。

#### 2.2.2.失败返回

- 消息名称,Fail,ErrorCode,;
- 消息格式如上所示, 由一个消息名称, 失败标志 Fail, 后接错误码, 每一个字段之间以英文逗号 “,” 相隔, 一个完整的消息以英文逗号和分号 “;” 结束。
- 错误码是 int 类型。
- 具体的错误码, 请参阅 “HansRobot\_ErrorCode.docx”。

## 第三章 服务端

协议采用标准的 C/S 架构，其中大族电机提供接收并处理机器人控制消息的客户端或服务端，整个通信过程采用 TCP/IP 方式进行。



图 5-1 系统架构

### 3.1.连接方式

#### 3.1.1.作为客户端

大族机器人控制系统作为通信客户端，会不断监控上位机服务端发出的连接服务器信号（以 IO 的形式触发），当接收到连接 IO 信号之后，客户端会根据配置文件配置好的 IP 地址和端口号，采用 TCP/IP 通信协议主动连接上位机服务器。如果客户端已经连接上服务器，服务器再次发送连接信号时，客户端会先关闭当前的连接，然后再连接服务器。



图 5-2 连接方式

#### 3.1.2.作为服务器

大族机器人控制系统作为通信服务端，启动时，会不断监听指定端口，等待上位机客户端的连接。其中，服务端监控的端口号可通过配置文件设置。

### 3.1.3.配置文件

通过通信协议修改参数，不会保存至配置文件，即单次有效，重新启动控制器，则恢复默认参数。

在 HansRobot 示教器设置参数会保存至配置文件。

### 3.1.4.连接示例

#### 3.1.4.1. 同一台工控机进行通信

1.控制器作为服务端，客户软件作为客户端，以 NetAssist.exe 为例子：

NetAssist 作为客户端，IP 地址为本机地址，端口 10003：

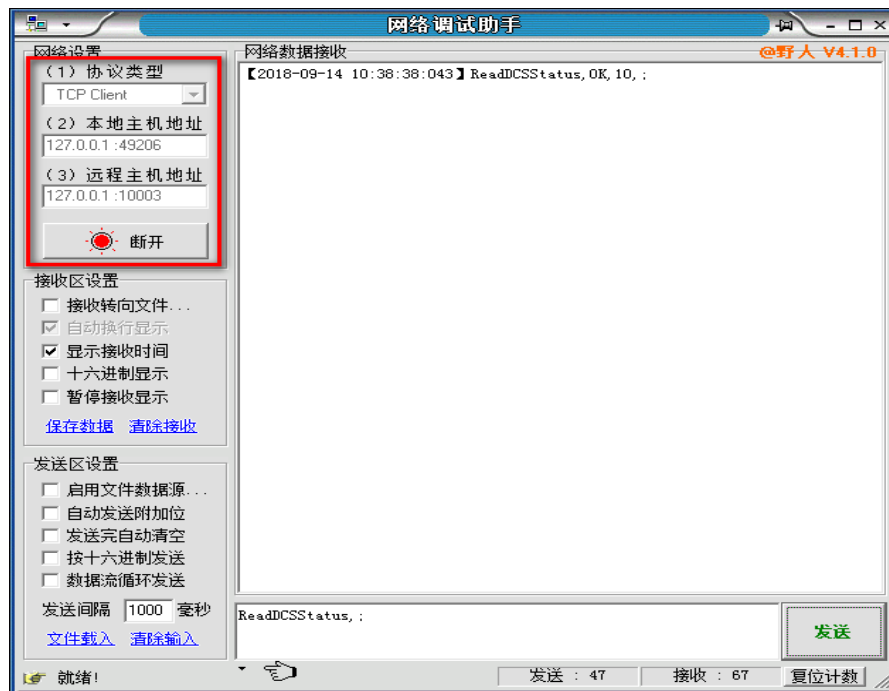


图 5-3 连接示例

#### 3.1.4.2. 工控机与笔记本通信

- 1.使用网线将工控机和笔记本连接；
- 2.修改工控机和笔记本 IP 地址，使其在同一个域。
  - a) 进入管理-网络，修改工控机 IP:

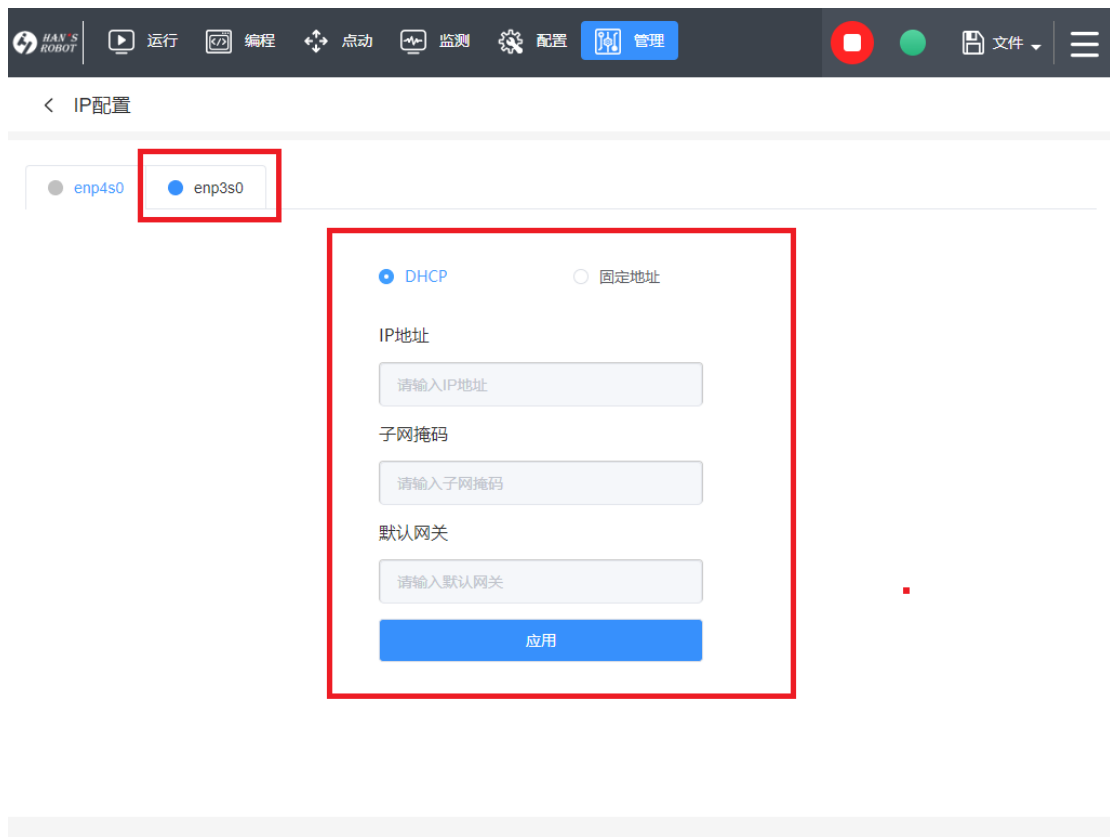


图 5-4 修改工控机 IP

b) 修改笔记本 IP:

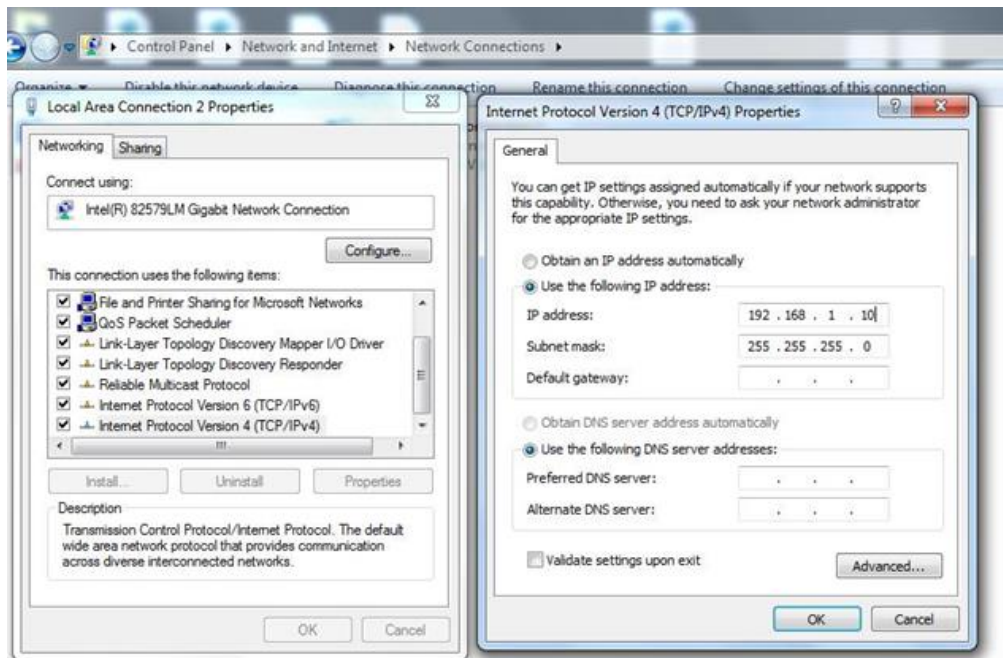


图 5-5 修改笔记本 IP

3. 控制器作为服务端，客户软件作为客户端，以 NetAssist.exe 为例子：

NetAssist 作为客户端，IP 地址为工控机地址，端口 10003：

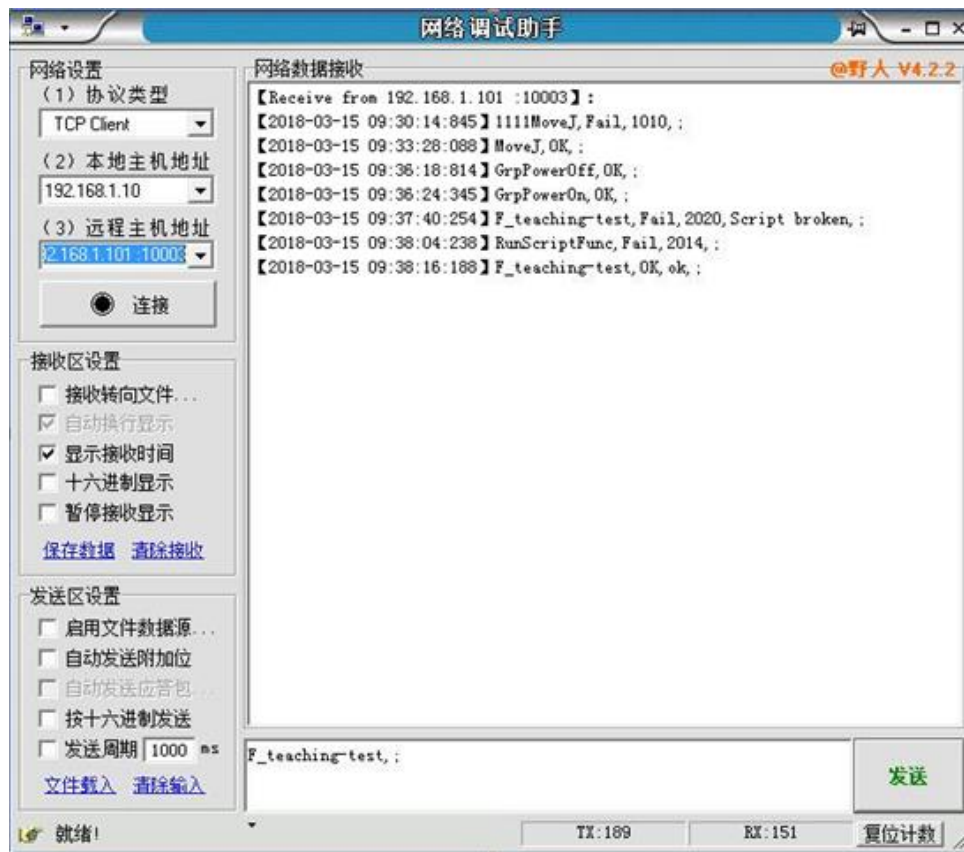


图 5-6 连接示例

## 3.2.服务端口表

服务端口	说明
10003	IF 指令服务端口
10004	Datasheet 服务端口, 使用 Json 格式推送 Datasheet 给客户端, 频率 20Hz
10005	Datasheet 服务端口, 使用 Json 格式推送 Datasheet 给客户端, 频率 10Hz
10006	Datasheet 服务端口, 使用 Json 格式推送 Datasheet 给客户端, 频率 5Hz
10014	Datasheet 服务端口, 使用 Struct 内存推送 Datasheet 给客户端, 频率 20Hz
10015	Datasheet 服务端口, 使用 Struct 内存推送 Datasheet 给客户端, 频率 10Hz
10016	Datasheet 服务端口, 使用 Struct 内存推送 Datasheet 给客户端, 频率 5Hz
10502	ModbusTCP 服务端, 0~99 的 holding 寄存器使用
30003	日志服务器, 产生日志时向客户端发送日志

## 第四章 通信协议接口

### 4.1.初始化

#### 4.1.1.OSCmd

4.1.1.1. 描述：控制器断电（断开机器人供电，系统关机）。

✓ 发送消息：OSCmd,nCmdType,;

参数	名称	数据类型	有效范围	内容
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

✓ 发送成功返回消息：OSCmd,OK,;

✓ 发送失败返回消息：OSCmd,Fail,ErrorCode,;

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

#### 4.1.1.2. 示例

// 控制器断电

OSCmd,1;

// 成功返回

OSCmd,OK,;

// 失败返回

OSCmd,Fail,20018,;

## 4.1.2.ConnectToBox

4.1.2.1. 描述：连接控制器电箱。

- ✓ 发送消息：ConnectToBox,;
- ✓ 发送成功返回消息：ConnectToBox,OK,;
- ✓ 发送失败返回消息：ConnectToBox,Fail,ErrorCode,;

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

4.1.2.2. 示例

// 连接控制器电箱

ConnectToBox,0,;

// 成功返回

ConnectToBox,OK,;

// 失败返回

ConnectToBox,Fail,20018,;



### 4.1.3.Electrify

#### 4.1.3.1. 描述：机器人上电。

- ✓ 发送消息：Electrify,;
- ✓ 发送成功返回消息：Electrify,OK,;
- ✓ 发送失败返回消息：Electrify,Fail,ErrorCode,;

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

#### 4.1.3.2. 示例

// 机器人上电

Electrify,;

// 成功返回

Electrify,OK,;

// 失败返回

Electrify,Fail,20018,;

#### 4.1.4.BlackOut

4.1.4.1. 描述：机器人断电。

- ✓ 发送消息：BlackOut,;
- ✓ 发送成功返回消息：BlackOut,OK,;
- ✓ 发送失败返回消息：BlackOut,Fail,ErrorCode,;

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

4.1.4.2. 示例

// 机器人断电

BlackOut,;

// 成功返回

BlackOut,OK,;

// 失败返回

BlackOut,Fail,20018,;

## 4.1.5.StartMaster

### 4.1.5.1. 描述：连接控制器。

- ✓ 发送消息：StartMaster,;
- ✓ 发送成功返回消息：StartMaster,OK,;
- ✓ 发送失败返回消息：StartMaster,Fail,ErrorCode,;

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

### 4.1.5.2. 示例

// 连接机器人控制器

StartMaster,;

// 成功返回

StartMaster,OK,;

// 失败返回

StartMaster,Fail,20018,;

## 4.1.6.CloseMaster

4.1.6.1. 描述：断开控制器连接，断开连接过程中会关闭主站。

- ✓ 发送消息：CloseMaster,;
- ✓ 发送成功返回消息：CloseMaster,OK,;
- ✓ 发送失败返回消息：CloseMaster,Fail,ErrorCode,;

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

### 4.1.6.2. 示例

// 断开连接机器人控制器

CloseMaster,;

// 成功返回

CloseMaster,OK,;

// 失败返回

CloseMaster,Fail,20018,;

### 4.1.7.IsSimulation

4.1.7.1. 描述：获取机器人是否为模拟模式。

✓ 发送消息：IsSimulation,;

✓ 发送成功返回消息：IsSimulation,OK,nSimulateRobot,;

参数	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

✓ 发送失败返回消息：IsSimulation,Fail,ErrorCode,;

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

4.1.7.2. 示例

// 判断是否为模拟机器人

IsSimulation,;

// 成功返回

IsSimulation,OK,1,;

// 失败返回

IsSimulation,Fail,20018,;

## 4.1.8.ReadControllerState

4.1.8.1. 描述：控制器是否启动完成。

✓ 发送消息：ReadControllerState,;

✓ 发送成功返回消息：ReadControllerState,OK,nStarted,;

参数	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

✓ 发送失败返回消息：ReadControllerState,Fail,ErrorCode,;

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

4.1.8.2. 示例

// 判断控制器是否启动完成

ReadControllerState,;

// 成功返回

ReadControllerState,OK,1,;

// 失败返回

ReadControllerState,Fail,20018,;

## 4.1.9.ReadRobotModel

### 4.1.9.1. 描述：读取机器人型号。

✓ 发送消息：ReadRobotModel,;

✓ 发送成功返回消息：ReadRobotModel,OK,strModel,;

参数	名称	数据类型	有效范围	内容
strModel	机器人型号	string	-	机器人型号

✓ 发送失败返回消息：ReadRobotModel,Fail,ErrorCode,;

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

### 4.1.9.2. 示例

// 读取机器人型号

ReadRobotModel,;

// 成功返回

ReadRobotModel,OK,5KG,;

// 失败返回

ReadRobotModel,Fail,20018,;

## 4.2.轴组控制指令

### 4.2.1.GrpEnable

4.2.1.1. 描述：机器人上使能命令。

✓ 发送消息：GrpEnable,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

✓ 发送成功返回消息：GrpEnable,OK,;

✓ 发送失败返回消息：GrpEnable,Fail,ErrorCode,;

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

4.2.1.2. 示例

// 机器人使能

GrpEnable,0,;

// 成功返回

GrpEnable,OK,;

// 失败返回

GrpEnable,Fail,20018,;



## 4.2.2.GrpDisable

4.2.2.1. 描述：机器人去使能命令。

✓ 发送消息：GrpDisable,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

✓ 发送成功返回消息：GrpDisable,OK,;

✓ 发送失败返回消息：GrpDisable,Fail,ErrorCode,;

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

4.2.2.2. 示例

// 机器人去使能

GrpDisable,0,;

// 成功返回

GrpDisable,OK,;

// 失败返回

GrpDisable,Fail,20018,;

### 4.2.3.GrpReset

4.2.3.1. 描述：机器人复位命令。

✓ 发送消息：GrpReset,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

✓ 发送成功返回消息：GrpReset,OK,;

✓ 发送失败返回消息：GrpReset,Fail,ErrorCode,;

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

4.2.3.2. 示例

// 机器人复位

GrpReset,0,;

// 成功返回

GrpReset,OK,;

// 失败返回

GrpReset,Fail,20018,;

## 4.2.4.GrpStop

4.2.4.1. 描述：机器人停止运动命令。

✓ 发送消息：GrpStop,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

✓ 发送成功返回消息：GrpStop,OK,;

✓ 发送失败返回消息：GrpStop,Fail,ErrorCode,;

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

4.2.4.2. 示例

// 机器人停止运动

GrpStop,0,;

// 成功返回

GrpStop,OK,;

// 失败返回

GrpStop,Fail,20018,;

## 4.2.5.GrpInterrupt

4.2.5.1. 描述：机器人暂停运动命令。

✓ 发送消息：GrpInterrupt,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

✓ 发送成功返回消息：GrpInterrupt,OK,;

✓ 发送失败返回消息：GrpInterrupt,Fail,ErrorCode,;

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

4.2.5.2. 示例

// 机器人暂停运动

GrpInterrupt,0,;

// 成功返回

GrpInterrupt,OK,;

// 失败返回

GrpInterrupt,Fail,20018,;

## 4.2.6.GrpContinue

4.2.6.1. 描述：机器人继续运动命令。

✓ 发送消息：GrpContinue,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

✓ 发送成功返回消息：GrpContinue,OK,;

✓ 发送失败返回消息：GrpContinue,Fail,ErrorCode,;

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

4.2.6.2. 示例

// 机器人继续运动

GrpContinue,0,;

// 成功返回

GrpContinue,OK,;

// 失败返回

GrpContinue,Fail,20018,;

## 4.2.7.GrpCloseFreeDriver

### 4.2.7.1. 描述：关闭自由驱动。

- ✓ 发送消息：GrpCloseFreeDriver,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

- ✓ 发送成功返回消息：GrpCloseFreeDriver,OK,;

- ✓ 发送失败返回消息：GrpCloseFreeDriver,Fail,ErrorCode,;

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

### 4.2.7.2. 示例

// 机器人关闭自由驱动

GrpCloseFreeDriver,0,;

// 成功返回

GrpCloseFreeDriver,OK,;

// 失败返回

GrpCloseFreeDriver,Fail,20018,;

## 4.2.8.GrpOpenFreeDriver

### 4.2.8.1. 描述：打开自由驱动。

- ✓ 发送消息：GrpOpenFreeDriver,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

- ✓ 发送成功返回消息：GrpOpenFreeDriver,OK,;

- ✓ 发送失败返回消息：GrpOpenFreeDriver,Fail,ErrorCode,;

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

### 4.2.8.2. 示例

// 机器人打开自由驱动

GrpOpenFreeDriver,0,;

// 成功返回

GrpOpenFreeDriver,OK,;

// 失败返回

GrpOpenFreeDriver,Fail,20018,;

## 4.3.脚本控制指令

### 4.3.1.RunFunc

4.3.1.1. 描述：运行指定脚本函数。

✓ 发送消息：RunFunc,strFuncName, sParams1,...,sParamsn,;

参数	名称	数据类型	有效范围	内容
strFuncName	函数名称	string	-	指定脚本函数名称，对应示教器界面的函数
sParams	参数列表	string	-	参数列表

✓ 发送成功返回消息：RunFunc,OK,strName,;

返回值	名称	数据类型	有效范围	内容
strName	函数名称	string	-	指令运行的函数名称

✓ 发送失败返回消息：RunFunc,Fail,ErrorCode,;

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

4.3.1.2. 示例

// 运行函数 Func\_1

RunFunc,Func\_1,55,;

// 成功返回

RunFunc,OK,Func\_1,;

// 失败返回

RunFunc,Fail,20018,;



## 4.3.2.StartScript

4.3.2.1. 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。

- ✓ 发送消息：StartScript,;
- ✓ 发送成功返回消息：StartScript,OK,;
- ✓ 发送失败返回消息：StartScript,Fail,ErrorCode,;

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

### 4.3.2.2. 示例

// 运行脚本 Func\_main 函数

StartScript,;

// 成功返回

StartScript,OK,;

// 失败返回

StartScript,Fail,20018,;

### 4.3.3.StopScript

4.3.3.1. 描述：停止脚本，调用后停止示教器页面正在执行脚本文件。

- ✓ 发送消息：StopScript,;
- ✓ 发送成功返回消息：StopScript,OK,;
- ✓ 发送失败返回消息：StopScript,Fail,ErrorCode,;

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

4.3.3.2. 示例

// 停止正在运行的脚本函数

StopScript,;

// 成功返回

StopScript,OK,;

// 失败返回

StopScript,Fail,20018,;

### 4.3.4.PauseScript

4.3.4.1. 描述：暂停脚本，调用后暂停示教器页面正在执行脚本文件。

- ✓ 发送消息：PauseScript,;
- ✓ 发送成功返回消息：PauseScript,OK,;
- ✓ 发送失败返回消息：PauseScript,Fail,ErrorCode,;

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

4.3.4.2. 示例

// 暂停正在运行的脚本函数

PauseScript,;

// 成功返回

PauseScript,OK,;

// 失败返回

PauseScript,Fail,20018,;

### 4.3.5.ContinueScript

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

- ✓ 发送消息: ContinueScript,;
- ✓ 发送成功返回消息: ContinueScript,OK,;
- ✓ 发送失败返回消息: ContinueScript,Fail,ErrorCode,;

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

#### 4.3.5.2. 示例

// 继续运动暂停的脚本函数

ContinueScript,;

// 成功返回

ContinueScript,OK,;

// 失败返回

ContinueScript,Fail,20018,;

## 4.4.电箱控制指令

### 4.4.1.ReadBoxInfo

4.4.1.1. 描述：读取电箱信息。

✓ 发送消息：ReadBoxInfo,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0

✓ 发送成功返回消息：ReadBoxInfo,OK,nConnected,n48V\_ON,db48OUT\_Voltag,db48OUT\_Current,nRemoteBTN,nThreeStageBTN,;

返回值	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

✓ 发送失败返回消息：ReadBoxInfo,Fail,ErrorCode,;

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

4.4.1.2. 示例

// 读取电箱信息

ReadBoxInfo,0,;

// 成功返回

ReadBoxInfo,OK,1,1,48.000,0.000,0,0,;

// 失败返回

---

ReadBoxInfo,Fail,20018,;

## 4.4.2.ReadBoxCI

4.4.2.1. 描述：读取电箱控制数字输入状态。

✓ 发送消息：ReadBoxCI,nBit0,nBit1,nBit2,nBit3,nBit4,nBit5,nBit6,nBit7,;

参数	名称	数据类型	有效范围	内容
nBit0~nBit7	控制数字输入位	int	0~7	控制数字输入位索引

✓ 发送成功返回消息：ReadBoxCI,OK,nVal0,nVal1,nVal2,nVal3,nVal4,nVal5,nVal6,nVal7,;

返回值	名称	数据类型	有效范围	内容
nVal0~nBit7	控制数字输入对应位状态	int	0/1	0: 低电平 1: 高电平

✓ 发送失败返回消息：ReadBoxCI,Fail,ErrorCode,;

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

4.4.2.2. 示例

// 读取电箱 CI 状态

ReadBoxCI,0,1,2,3,4,5,6,7,;

// 成功返回

ReadBoxCI,OK,0,1,1,1,1,1,1,1,;

// 失败返回

ReadBoxCI,Fail,20018,;

\* 注：如果仅需读取其中的 1 位或者多位则按照实际需要读取的数量下发，例如：

// 读取电箱 CI 第 0 位状态

ReadBoxCI,0,;

ReadBoxCI,OK,0,;

// 读取电箱 CI 第 5 位和第 6 位状态

ReadBoxCI,5,6,;

ReadBoxCI,OK,0,1,;

### 4.4.3.ReadBoxDI

4.4.3.1. 描述：读取电箱通用数字输入状态。

✓ 发送消息：ReadBoxDI,nBit0,nBit1,nBit2,nBit3,nBit4,nBit5,nBit6,nBit7,;

参数	名称	数据类型	有效范围	内容
nBit0~nBit7	通用数字输入位	int	0~7	控制数字输入位索引

✓ 发送成功返回消息：ReadBoxDI,OK,nVal0,nVal1,nVal2,nVal3,nVal4,nVal5,nVal6,nVal7,;

返回值	名称	数据类型	有效范围	内容
nVal0~nBit7	通用数字输入对应位状态	int	0/1	0: 低电平 1: 高电平

✓ 发送失败返回消息：ReadBoxDI,Fail,ErrorCode,;

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

4.4.3.2. 示例

// 读取电箱 DI 状态

ReadBoxDI,0,1,2,3,4,5,6,7,;

// 成功返回

ReadBoxDI,OK,0,1,1,1,1,1,1,1,;

// 失败返回

ReadBoxDI,Fail,20018,;

// 读取电箱 DI 第 0 位状态

ReadBoxDI,0,;

ReadBoxDI,OK,0,;

// 读取电箱 DI 第 5 位和第 6 位状态

ReadBoxDI,5,6,;

ReadBoxDI,OK,0,1,;



#### 4.4.4.ReadBoxCO

4.4.4.1. 描述：读取电箱控制数字输出状态。

✓ 发送消息：ReadBoxCO,nBit0,nBit1,nBit2,nBit3,nBit4,nBit5,nBit6,nBit7,;

参数	名称	数据类型	有效范围	内容
nBit0~nBit7	控制数字输出位	int	0~7	控制数字输出位索引

✓ 发送成功返回消息：ReadBoxCO,OK,nVal0,nVal1,nVal2,nVal3,nVal4,nVal5,nVal6,nVal7,;

返回值	名称	数据类型	有效范围	内容
nVal0~nVal7	控制数字输出对应位状态	int	0/1	0: 低电平 1: 高电平

✓ 发送失败返回消息：ReadBoxCO,Fail,ErrorCode,;

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

4.4.4.2. 示例

// 读取电箱 CO 状态

ReadBoxCO,0,1,2,3,4,5,6,7,;

// 成功返回

ReadBoxCO,OK,0,1,1,1,1,1,1,1,;

// 失败返回

ReadBoxCO,Fail,20018,;

// 读取电箱 CO 第 0 位状态

ReadBoxCO,0,;

ReadBoxCO,OK,0,;

// 读取电箱 CO 第 5 位和第 6 位状态

ReadBoxCO,5,6,;

ReadBoxCO,OK,0,1,;

## 4.4.5.ReadBoxDO

4.4.5.1. 描述：读取电箱通用数字输出状态。

✓ 发送消息：ReadBoxDO,nBit0,nBit1,nBit2,nBit3,nBit4,nBit5,nBit6,nBit7,;

参数	名称	数据类型	有效范围	内容
nBit0~nBit7	通用数字输出位	int	0~7	通用数字输出位索引

✓ 发送成功返回消息：ReadBoxDO,OK,nVal0,nVal1,nVal2,nVal3,nVal4,nVal5,nVal6,nVal7,;

返回值	名称	数据类型	有效范围	内容
nVal0~nVal7	通用数字输出对应位状态	int	0/1	0: 低电平 1: 高电平

✓ 发送失败返回消息：ReadBoxDO,Fail,ErrorCode,;

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

4.4.5.2. 示例

// 读取电箱 DO 状态

ReadBoxDO,0,1,2,3,4,5,6,7,;

// 成功返回

ReadBoxDO,OK,0,1,1,1,1,1,1,;

// 失败返回

ReadBoxDO,Fail,20018,;

// 读取电箱 DO 第 0 位状态

ReadBoxDO,0,;

ReadBoxDO,OK,0,;

// 读取电箱 DO 第 5 位和第 6 位状态

ReadBoxDO,5,6,;

ReadBoxDO,OK,0,1,;

## 4.4.6.ReadBoxAI

### 4.4.6.1. 描述：读取电箱模拟量输入值。

✓ 发送消息：ReadBoxAI,nBit0,nBit1,;

参数	名称	数据类型	有效范围	内容
nBit0~nBit1	模拟量输入 通道	int	0/1	0: 模拟量通道 0 1: 模拟量通道 1

✓ 发送成功返回消息：ReadBoxAI,OK,dVal0,dVal1,;

返回值	名称	数据类型	有效范围	内容
dVal0~dVal1	对应的模拟 量通道值	double	0~20	电流模式:4-20mA 电压模式:0-10V

✓ 发送失败返回消息：ReadBoxAI,Fail,ErrorCode,;

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

### 4.4.6.2. 示例

// 读取电箱模拟量输入值

ReadBoxAI,0,1,;

// 成功返回

ReadBoxAI,OK,5.5,3.8,;

// 失败返回

ReadBoxAI,Fail,20018,;

\* 注：如果仅需读取其中的 1 位，例如：

// 读取电箱 AI 通道 0 数据

ReadBoxAI,0,;

ReadBoxAI,OK,5.5,;

## 4.4.7.ReadBoxAO

### 4.4.7.1. 描述：读取电箱模拟量输出值。

✓ 发送消息：ReadBoxAO,nRbtID,nBit,;

参数	名称	数据类型	有效范围	内容
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

✓ 发送成功返回消息：ReadBoxAO,OK,dVal,;

返回值	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

✓ 发送失败返回消息：ReadBoxAO,Fail,ErrorCode,;

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

### 4.4.7.2. 示例

// 读取电箱模拟量输出值

ReadBoxAO,0,1,;

// 成功返回

ReadBoxAO,OK,5.5,3.8,;

// 失败返回

ReadBoxAO,Fail,20018,;

\* 注：如果仅需读取其中的 1 位，例如：

// 读取电箱 AO 通道 0 数据

ReadBoxAO,0,;

ReadBoxAO,OK,5.5,;

## 4.4.8.SetBoxCO

4.4.8.1. 描述：设置电箱控制数字输出状态。

✓ 发送消息：SetBoxCO,nBit,nVal,;

参数	名称	数据类型	有效范围	内容
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

✓ 发送成功返回消息：SetBoxCO,OK,;

✓ 发送失败返回消息：SetBoxCO,Fail,ErrorCode,;

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

4.4.8.2. 示例

// 设置电箱 CO5=1

SetBoxCO,5,1,;

// 成功返回

SetBoxCO,OK,;

// 失败返回

SetBoxCO,Fail,20018,;

## 4.4.9.SetBoxDO

4.4.9.1. 描述：设置电箱通用数字输出状态。

✓ 发送消息：SetBoxDO,nBit,nVal,;

参数	名称	数据类型	有效范围	内容
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

✓ 发送成功返回消息：SetBoxDO,OK,;

✓ 发送失败返回消息：SetBoxDO,Fail,ErrorCode,;

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

4.4.9.2. 示例

// 设置电箱 DO5=1

SetBoxDO,5,1,;

// 成功返回

SetBoxDO,OK,;

// 失败返回

SetBoxDO,Fail,20018,;

#### 4.4.10.SetBoxAOMode

4.4.10.1. 描述：设置电箱模拟量输出模式。

✓ 发送消息：SetBoxAOMode,nBit,nMode,;

参数	名称	数据类型	有效范围	内容
nBit	模拟量通道	int	0/1	模拟量通道
nMode	模式	int	1/2	1: 电压模式 2: 电流模式

✓ 发送成功返回消息：SetBoxAOMode,OK,;

✓ 发送失败返回消息：SetBoxAOMode,Fail,ErrorCode,;

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

4.4.10.2. 示例

// 设置电箱 AO 通道 1 模式为电压模式

SetBoxAOMode,0,1,;

// 设置电箱 AO 通道 2 模式为电流模式

SetBoxAOMode,1,2,;

// 成功返回

SetBoxAOMode,OK,;

// 失败返回

SetBoxAOMode,Fail,20018,;

## 4.4.11.SetBoxAO

4.4.11.1. 描述：设置电箱模拟量输出值。

✓ 发送消息：SetBoxAO, nBit,dVal,nMode,;

参数	名称	数据类型	有效范围	内容
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

✓ 发送成功返回消息：SetBoxAO,OK,;

✓ 发送失败返回消息：SetBoxAO,Fail,ErrorCode,;

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

4.4.11.2. 示例

*// 设置电箱 AO 通道 1 模式为电压模式, 输出电压为 5.5V*

SetBoxAO,0,5.5,1,;

*// 设置电箱 AO 通道 2 模式为电流模式, 输出电流为 8.8mA*

SetBoxAO,1,8.8,2,;

*// 成功返回*

SetBoxAO,OK,;

*// 失败返回*

SetBoxAO,Fail,20018,;



## 4.4.12.SetEndDO

### 4.4.12.1. 描述：设置末端数字输出状态。

✓ 发送消息：SetEndDO,nRbtID,nBit,nVal,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
nBit	末端数字输出位	int	0~3	末端通用数字输出位索引
nVal	末端数字输出状态	int	0/1	0：低电平 1：高电平

✓ 发送成功返回消息：SetEndDO,OK,;

✓ 发送失败返回消息：SetEndDO,Fail,ErrorCode,;

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

### 4.4.12.2. 示例

// 设置末端 DOI=1

SetEndDO,0,1,1,;

// 成功返回

SetEndDO,OK,;

// 失败返回

SetEndDO,Fail,20018,;

### 4.4.13.ReadEI

4.4.13.1. 描述：读取末端数字输入状态，新末端 IO 只有三个输入和输出。

✓ 发送消息：ReadEI,nRbtID,nBit0,nBit1,nBit2,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0
nBit0~nBit2	末端数字输入位	int	0~2	末端数字输入对应位状态

✓ 发送成功返回消息：ReadEI,OK,nVal0,nVal1,nVal2,;

返回值	名称	数据类型	有效范围	内容
nVal0~nVal2	末端数字输入 对应位状态	int	0/1	0: 低电平 1: 高电平

✓ 发送失败返回消息：ReadEI,Fail,ErrorCode,;

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

#### 4.4.13.2. 示例

// 读取末端 DI 状态

ReadEI,0,0,1,2,;

// 成功返回

ReadEI,OK,0,1,1,;

// 失败返回

ReadEI,Fail,20018,;

// 读取末端 DI 第 0 位状态

ReadEI,0,0,;

ReadEI,OK,0,;

#### 4.4.14.ReadEO

4.4.14.1. 描述：读取末端数字输出状态，新末端 IO 只有三个输入和输出。

✓ 发送消息：ReadEO,nRbtID,nBit0,nBit1,nBit2,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	int	0~5	机器 ID 号，默认=0
nBit0~nBit2	末端数字输出位	int	0~2	末端数字输出对应位索引

✓ 发送成功返回消息：ReadEO,OK,nVal0,nVal1,nVal2,;

返回值	名称	数据类型	有效范围	内容
nVal0~nVal2	末端数字输出对应位状态	int	0/1	0: 低电平 1: 高电平

✓ 发送失败返回消息：ReadEO,Fail,ErrorCode,;

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

#### 4.4.14.2. 示例

// 读取末端 DO 状态

ReadEO,0,0,1,2,;

// 成功返回

ReadEO,OK,0,1,1,;

// 失败返回

ReadEO,Fail,20018,;

// 读取末端 DO 第 0 位状态

ReadEO,0,0,;

ReadEO,OK,0,;

// 读取末端 DO 第 1 位和第 2 位状态

ReadEO,0,1,2,;

ReadEO,OK,0,1,;

## 4.4.15.ReadEAI

4.4.15.1. 描述：读取末端模拟量输入值。

✓ 发送消息：ReadEAI,nRbtID,nBit,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：ReadEAI,OK,dVal,;

返回值	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

✓ 发送失败返回消息：ReadEAI,Fail,ErrorCode,;

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

4.4.15.2. 示例

// 读取末端模拟量输入值

ReadEAI,0,1,;

// 成功返回

ReadEAI,OK,5.5,;

// 失败返回

ReadEAI,Fail,20018,;

## 4.5.状态读取与设置指令

### 4.5.1.SetOverride

4.5.1.1. 描述：设置速度比，运动状态不能修改速度比。

✓ 发送消息：SetOverride,nRbtID,dOverride,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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(0.01~1)

✓ 发送成功返回消息：SetOverride,OK,;

✓ 发送失败返回消息：SetOverride,Fail,ErrorCode,;

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

4.5.1.2. 示例

// 设置当前速度比为 50%

SetOverride,0,0.5,;

// 成功返回

SetOverride,OK,;

// 失败返回

SetOverride,Fail,20018,;

## 4.5.2.SetToolMotion

4.5.2.1. 描述：开启或关闭 Tool 坐标系运动模式。

✓ 发送消息：SetToolMotion,nRbtID,nState,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
nState	状态	int	0/1	0：关闭 1：开启

✓ 发送成功返回消息：SetToolMotion,OK,;

✓ 发送失败返回消息：SetToolMotion,Fail,ErrorCode,;

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

4.5.2.2. 示例

// 设置开启 Tool 坐标运动模式

SetToolMotion,0,1,;

// 成功返回

SetToolMotion,OK,;

// 失败返回

SetToolMotion,Fail,20018,;

### 4.5.3.SetPayload

#### 4.5.3.1. 描述：设置当前负载参数。

✓ 发送消息：SetPayload,nRbtID,dMass,dX,dY,dZ,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]

✓ 发送成功返回消息：SetPayload,OK,;

✓ 发送失败返回消息：SetPayload,Fail,ErrorCode,;

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

#### 4.5.3.2. 示例

// 设置质量 3kg, 质心为 10,20,30

SetPayload,0,3,10,20,30,;

// 成功返回

SetPayload,OK,;

// 失败返回

SetPayload,Fail,20018,;

## 4.5.4.SetJointMaxVel

4.5.4.1. 描述：设置关节最大运动速度。

✓ 发送消息：SetJointMaxVel,nRbtID,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]

\* 注：关节速度有效范围需要参考具体机型。

✓ 发送成功返回消息：SetJointMaxVel,OK,;

✓ 发送失败返回消息：SetJointMaxVel,Fail,ErrorCode,;

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

4.5.4.2. 示例

// 设置各关节最大速度, J1-J4=180°/s, J5-J6=200°/s

SetJointMaxVel,0,180,180,180,180,200,200,;

// 成功返回

SetJointMaxVel,OK,;

// 失败返回

SetJointMaxVel,Fail,20018,;



## 4.5.5.SetJointMaxAcc

4.5.5.1. 描述：设置关节最大运动加速度。

✓ 发送消息：SetJointMaxAcc,nRbtID,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°/s <sup>2</sup> ]
dJ2	J2 轴最大加速度	double	0~720	J2 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ3	J3 轴最大加速度	double	0~720	J3 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ4	J4 轴最大加速度	double	0~720	J4 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ5	J5 轴最大加速度	double	0~720	J5 轴最大加速度，单位[°/s <sup>2</sup> ]
dJ6	J6 轴最大加速度	double	0~720	J6 轴最大加速度，单位[°/s <sup>2</sup> ]

\* 注：关节加速度有效范围需要参考具体机型。

✓ 发送成功返回消息：SetJointMaxAcc,OK,;

✓ 发送失败返回消息：SetJointMaxAcc,Fail,ErrorCode,;

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

4.5.5.2. 示例

// 设置各关节最大加速度 360 °/s<sup>2</sup>

SetJointMaxAcc,0,360, 360, 360, 360, 360, 360,;

// 成功返回

SetJointMaxAcc,OK,;

// 失败返回

SetJointMaxAcc,Fail,20018,;

## 4.5.6.SetLinearMaxVel

4.5.6.1. 描述：设置直线运动最大速度。

✓ 发送消息：SetLinearMaxVel,nRbtID,dMaxVel,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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：最大直线速度有效范围需要参考具体机型。

✓ 发送成功返回消息：SetLinearMaxVel,OK,;

✓ 发送失败返回消息：SetLinearMaxVel,Fail,ErrorCode,;

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

4.5.6.2. 示例

// 设置直线最大运动速度为 1000mm/s

SetLinearMaxVel,0,1000,;

// 成功返回

SetLinearMaxVel,OK,;

// 失败返回

SetLinearMaxVel,Fail,20018,;

## 4.5.7.SetLinearMaxAcc

4.5.7.1. 描述：设置直线运动最大加速度。

✓ 发送消息：SetLinearMaxAcc,nRbtID,dMaxVel,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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 <sup>2</sup> ]

\* 注：直线运动加速度有效范围需要参考具体机型。

✓ 发送成功返回消息：SetLinearMaxAcc,OK,;

✓ 发送失败返回消息：SetLinearMaxAcc,Fail,ErrorCode,;

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

4.5.7.2. 示例

// 设置直线最大运动加速度为 2500mm/s<sup>2</sup>

SetLinearMaxAcc,0,2500,;

// 成功返回

SetLinearMaxAcc,OK,;

// 失败返回

SetLinearMaxAcc,Fail,20018,;

## 4.5.8.SetMaxAcsRange

### 4.5.8.1. 描述：设置关节最大运动范围。

- ✓ 发送消息：SetMaxAcsRange,nRbtID,dMaxJ1, dMaxJ2, dMaxJ3, dMaxJ4, dMaxJ5, dMaxJ6,dMinJ1, dMinJ2, dMinJ3, dMinJ4, dMinJ5, dMaxJ6,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

- ✓ 发送成功返回消息：SetMaxAcsRange,OK,;
- ✓ 发送失败返回消息：SetMaxAcsRange,Fail,ErrorCode,;

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

### 4.5.8.2. 示例

// 设置关节最大运动范围

SetMaxAcsRange,0,360,135,153,360,180,360,-360,-135,-153,-360,-180,-360,;

// 成功返回

SetMaxAcsRange,OK,;

// 失败返回

---

SetMaxAcsRange,Fail,20018,;

## 4.5.9.SetMaxPcsRange

### 4.5.9.1. 描述：设置空间最大运动范围。

- ✓ 发送消息：SetMaxPcsRange,nRbtID,dMaxX, dMaxY, dMaxZ, dMaxRx, dMaxRy, dMaxRz, dMinX, dMinY, dMinZ, dMinRx, dMinRy, dMinRz,dUcs\_X, dUcs\_Y, dUcs\_Z, dUcs\_Rx, dUcs\_Ry, dUcs\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

- ✓ 发送成功返回消息：SetMaxPcsRange,OK,;

- ✓ 发送失败返回消息：SetMaxPcsRange,Fail,ErrorCode,;

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

### 4.5.9.2. 示例

// 设置空间最大运动范围

SetMaxPcsRange,0,1200, 1200, 1200, 0, 0, 0, -1200, -1200, -1200, -0, -0, -0,0,0,0,0,0,;

// 成功返回

SetMaxPcsRange,OK,;

// 失败返回

SetMaxPcsRange,Fail,20018,;

## 4.5.10.ReadJointMaxVel

4.5.10.1. 描述：读取关节最大运动速度。

✓ 发送消息：ReadJointMaxVel,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadJointMaxVel,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	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]

✓ 发送失败返回消息：ReadJointMaxVel,Fail,ErrorCode,;

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

4.5.10.2. 示例

// 读取当前关节最大运动速度

ReadJointMaxVel,0,;

// 成功返回

ReadJointMaxVel,OK,170.000,170.000,170.000,170.000,180.000,180.000,;

// 失败返回

ReadJointMaxVel,Fail,20018,;

## 4.5.11.ReadJointMaxAcc

4.5.11.1. 描述：读取关节最大运动加速度。

✓ 发送消息：ReadJointMaxAcc,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadJointMaxAcc,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	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]

✓ 发送失败返回消息：ReadJointMaxAcc,Fail,ErrorCode,;

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

4.5.11.2. 示例

// 读取当前关节最大运动加速度

ReadJointMaxAcc,0,;

// 成功返回

ReadJointMaxAcc,OK,360.000,360.000,360.000,360.000,360.000,360.000,;

// 失败返回

ReadJointMaxAcc,Fail,20018,;



## 4.5.12.ReadJointMaxJerk

4.5.12.1. 描述：读取关节最大运动加加速度。

✓ 发送消息：ReadJointMaxJerk,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadJointMaxJerk,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	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[°/s³]
dJ2	J2 轴最大加加速度	double	0~3600	J2 轴最大加加速度，单位[°/s³]
dJ3	J3 轴最大加加速度	double	0~3600	J3 轴最大加加速度，单位[°/s³]
dJ4	J4 轴最大加加速度	double	0~3600	J4 轴最大加加速度，单位[°/s³]
dJ5	J5 轴最大加加速度	double	0~3600	J5 轴最大加加速度，单位[°/s³]
dJ6	J6 轴最大加加速度	double	0~3600	J6 轴最大加加速度，单位[°/s³]

✓ 发送失败返回消息：ReadJointMaxJerk,Fail,ErrorCode,;

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

4.5.12.2. 示例

// 读取当前关节最大运动加加速度

ReadJointMaxJerk,0,;

// 成功返回

ReadJointMaxJerk,OK,3600.000,3600.000,3600.000,3600.000,3600.000,3600.000,;

// 失败返回

ReadJointMaxJerk,Fail,20018,;

### 4.5.13.ReadLinearMaxVel

4.5.13.1. 描述：读取直线运动最大速度参数。

✓ 发送消息：ReadLinearMaxVel,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadLinearMaxVel,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	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 <sup>2</sup> ]
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 <sup>3</sup> ]

✓ 发送失败返回消息：ReadLinearMaxVel,Fail,ErrorCode,;

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

4.5.13.2. 示例

// 读取当前空间运动最大速度参数

ReadLinearMaxVel,0,;

// 成功返回

ReadLinearMaxVel,OK,1000.000,2500.000,10000000.000,;

// 失败返回

ReadLinearMaxVel,Fail,20018,;

## 4.5.14.ReadEmergencyInfo

### 4.5.14.1. 描述：读取急停信息。

✓ 发送消息：ReadEmergencyInfo,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadEmergencyInfo,OK,nESTO\_Error, nESTO,nSafetyGuard\_Error,nSafetyGuard,;

返回值	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，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，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

✓ 发送失败返回消息：ReadEmergencyInfo,Fail,ErrorCode,;

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

### 4.5.14.2. 示例

// 读取当前急停信息

ReadEmergencyInfo,0,;

// 成功返回

ReadEmergencyInfo,OK,0,0,0,0,;

// 失败返回

ReadEmergencyInfo,Fail,20018,;

## 4.5.15.ReadRobotState

4.5.15.1. 描述：读取当前机器人状态标志。

✓ 发送消息：ReadRobotState,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadRobotState,OK,nMovingState,nEnableState,nErrorState,nErrorCode,  
nErrorAxis, nBreaking, nPause, nEmergencyStop, nSaftyGuard, nElectrify, nIsConnectToBox,  
nBlendingDone, nInPos,;

返回值	名称	数据类型	有效范围	内容
nMovingState	运动状态	int	0/1	0: 机器人未处于运动状态 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	错误轴索引
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)

✓ 发送失败返回消息: ReadRobotState,Fail,ErrorCode,;

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

#### 4.5.15.2. 示例

// 读取当前机器人状态信息

ReadRobotState,0,;

// 成功返回

ReadRobotState,OK,1,1,0,0,0,0,0,0,0,1,1,0,0,;

// 失败返回

ReadRobotState,Fail,20018,;

## 4.5.16.ReadCurWayPointID

4.5.16.1.描述：读取 WayPoint 当前运动 ID 号。

✓ 发送消息：ReadCurWayPointID,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCurWayPointID,OK, strCurWaypointID,;

返回值	名称	数据类型	有效范围	内容
strCurWaypointID	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

✓ 发送失败返回消息：ReadCurWayPointID,Fail,ErrorCode,;

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

4.5.16.2. 示例

// 读取当前机器人路点运动 ID

ReadCurWayPointID,0,;

// 成功返回

ReadCurWayPointID,OK,64BDE6A1BCF9AEAC06670EC8D5431093,;

// 失败返回

ReadCurWayPointID,Fail,20018,;

## 4.5.17.ReadAxisErrorCode

### 4.5.17.1. 描述：读取轴错误码。

✓ 发送消息：ReadAxisErrorCode,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadAxisErrorCode,OK,nErrorCode,nJ1,nJ2,nJ3,nJ4,nJ5,nJ6,;

返回值	名称	数据类型	有效范围	内容
nErrorCode	当前错误码	int	>0	轴组错误码，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

✓ 发送失败返回消息：ReadAxisErrorCode,Fail,ErrorCode,;

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

### 4.5.17.2. 示例

// 读取当前错误信息

ReadAxisErrorCode,0,;

// 成功返回

ReadAxisErrorCode,OK,0,0,0,0,0,0,0,;

// 失败返回

ReadAxisErrorCode,Fail,20018,;

## 4.5.18.ReadCurFSM

### 4.5.18.1. 描述：读取当前状态机。

✓ 发送消息：ReadCurFSM,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCurFSM,OK,nCurFSM,;

返回值	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

\* 注：关于状态机请参阅“HansRobot\_ErrorCode.docx”。

✓ 发送失败返回消息：ReadCurFSM,Fail,ErrorCode,;

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

### 4.5.18.2. 示例

// 读取状态机值，具体读取结果参考状态机表

ReadCurFSM,0,;

// 成功返回

ReadCurFSM,OK,34,;

// 失败返回

ReadCurFSM,Fail,20018,;



## 4.6.位置、速度、电流读取指令

### 4.6.1.ReadActPos

4.6.1.1. 描述：读取实际关节位置，空间位置，工具坐标，用户坐标。

✓ 发送消息：ReadActPos,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadActPos,OK, dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,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,;

返回值	名称	数据类型	有效范围	内容
dJ1-dJ6	实际关节坐标	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[°]
dX-dRz	实际空间坐标	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[°]
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[°]
----------------	--------	--------	---	---

✓ 发送失败返回消息: ReadActPos,Fail,ErrorCode,;

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

#### 4.6.1.2. 示例

// 读取实际关节位置, 空间位置, 工具坐标, 用户坐标

ReadActPos,0,;

// 成功返回

ReadActPos,OK,10.759,0.000,37.828,0.000,37.828,0.000,400.581,76.115,970.143,0.000,75.655,10.759,0.000,  
0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,;

// 失败返回

ReadActPos,Fail,20018,;

## 4.6.2.ReadCmdPos

4.6.2.1. 描述：读取关节和空间命令位置。

✓ 发送消息：ReadCmdPos,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCmdPos,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,dX,dY,dZ,dRx,dRy,dRz,;

返回值	名称	数据类型	有效范围	内容
dJ1-dJ6	关节命令坐标	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[°]
dX-dRz	命令空间坐标	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[°]

✓ 发送失败返回消息：ReadCmdPos,Fail,ErrorCode,;

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

4.6.2.2. 示例

// 读取关节和空间命令位置

ReadCmdPos,0,;

// 成功返回

ReadCmdPos,OK,10.759,0.000,37.828,0.000,37.828,0.000,400.581,76.115,970.143,0.000,75.655,10.759,;

// 失败返回

---

ReadActPos,Fail,20018,;

### 4.6.3.ReadCmdJointVel

4.6.3.1. 描述：读取关节命令速度。

✓ 发送消息：ReadCmdJointVel,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCmdJointVel,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	名称	数据类型	有效范围	内容
dJ1-dJ6	关节命令速度	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]

✓ 发送失败返回消息：ReadCmdJointVel,Fail,ErrorCode,;

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

4.6.3.2. 示例

// 读取关节命令速度

ReadCmdJointVel,0,;

// 成功返回

ReadCmdJointVel,OK,0.000,0.000,50.000,0.000,50.000,0.000,;

// 失败返回

ReadCmdJointVel,Fail,20018,;

## 4.6.4.ReadActJointVel

4.6.4.1. 描述：读取关节实际速度。

✓ 发送消息：ReadActJointVel,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadActJointVel,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	名称	数据类型	有效范围	内容
dJ1-dJ6	实际关节速度	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]

✓ 发送失败返回消息：ReadActJointVel,Fail,ErrorCode,;

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

4.6.4.2. 示例

// 读取实际关节速度

ReadActJointVel,0,;

// 成功返回

ReadActJointVel,OK,0.000,0.000,50.000,0.000,50.000,0.000,;

// 失败返回

ReadActJointVel,Fail,20018,;

## 4.6.5.ReadCmdTcpVel

### 4.6.5.1. 描述：读取命令 TCP 速度。

✓ 发送消息：ReadCmdTcpVel,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCmdTcpVel,OK,dX,dY,dZ,dRx,dRy,dRz,;

返回值	名称	数据类型	有效范围	内容
dX-dRz	命令 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]

✓ 发送失败返回消息：ReadCmdTcpVel,Fail,ErrorCode,;

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

### 4.6.5.2. 示例

// 读取命令 TCP 速度

ReadCmdTcpVel,0,;

// 成功返回

ReadCmdTcpVel,OK,0.000,500.000,-0.000,-0.000,0.000,-0.000,;

// 失败返回

ReadCmdTcpVel,Fail,20018,;

## 4.6.6.ReadActTcpVel

4.6.6.1. 描述：读取实际 TCP 速度。

✓ 发送消息：ReadActTcpVel,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadActTcpVel,OK,dX,dY,dZ,dRx,dRy,dRz,;

返回值	名称	数据类型	有效范围	内容
dX-dRz	实际 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]

✓ 发送失败返回消息：ReadActTcpVel,Fail,ErrorCode,;

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

4.6.6.2. 示例

// 读取实际 TCP 速度

ReadActTcpVel,0,;

// 成功返回

ReadActTcpVel,OK,-1.181,498.099,-0.024,-0.000,0.000,-0.000,;

// 失败返回

ReadActTcpVel,Fail,20018,;



## 4.6.7.ReadCmdJointCur

4.6.7.1. 描述：读取关节命令电流。

✓ 发送消息：ReadCmdJointCur,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCmdJointCur,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	名称	数据类型	有效范围	内容
dJ1-dJ6	关节命令电流	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]

✓ 发送失败返回消息：ReadCmdJointCur,Fail,ErrorCode,;

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

4.6.7.2. 示例

// 读取关节命令电流

ReadCmdJointCur,0,;

// 成功返回

ReadCmdJointCur,OK,-0.000,-0.000,-0.000,-0.000,-0.000,-0.000,;

// 失败返回

ReadCmdJointCur,Fail,20018,;

## 4.6.8.ReadActJointCur

4.6.8.1. 描述：读取实际关节电流。

✓ 发送消息：ReadActJointCur,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadActJointCur,OK,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

返回值	名称	数据类型	有效范围	内容
dJ1-dJ6	实际关节电流	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]

✓ 发送失败返回消息：ReadActJointCur,Fail,ErrorCode,;

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

4.6.8.2. 示例

// 读取实际关节电流

ReadActJointCur,0,;

// 成功返回

ReadActJointCur,OK,-0.000,-0.000,-0.000,-0.000,-0.000,-0.000,;

// 失败返回

ReadActJointCur,Fail,20018,;

## 4.6.9.ReadTcpVelocity

### 4.6.9.1. 描述：读取 TCP 末端速度。

✓ 发送消息：ReadTcpVelocity,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

\* 注：本接口读取的速度为末端 TCP 的 XYZ 综合速度。

✓ 发送成功返回消息：ReadTcpVelocity,OK,dCmdVel,dActVel,;

返回值	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]

✓ 发送失败返回消息：ReadTcpVelocity,Fail,ErrorCode,;

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

### 4.6.9.2. 示例

// 读取 TCP 末端速度

ReadTcpVelocity,0,;

// 成功返回

ReadTcpVelocity,OK,0.000,0.000,;

// 失败返回

ReadTcpVelocity,Fail,20018,;

## 4.7.坐标转换计算指令

### 4.7.1.Quaternion2RPY

4.7.1.1. 描述：四元素转欧拉角。

✓ 发送消息：Quaternion2RPY,nRbtID,dQuaW,dQuaX,dQuaY,dQuaZ,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：Quaternion2RPY,OK,dRx,dRy,dRz,;

返回值	名称	数据类型	有效范围	内容
dRx	欧拉角 Rx	double	$\geq -180, \leq 180$	欧拉角 Rx
dRy	欧拉角 Ry	double	$\geq -180, \leq 180$	欧拉角 Ry
dRz	欧拉角 Rz	double	$\geq -180, \leq 180$	欧拉角 Rz

✓ 发送失败返回消息：Quaternion2RPY,Fail,ErrorCode,;

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

4.7.1.2. 示例

// 四元素转欧拉角

Quaternion2RPY,0,0,0,0,;

// 成功返回

Quaternion2RPY,OK,180.000,0.000,180.000,;

// 失败返回

Quaternion2RPY,Fail,20018,;

## 4.7.2.RPY2Quaternion

### 4.7.2.1. 描述：欧拉角转四元素。

✓ 发送消息：RPY2Quaternion,nRbtID,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
Rx	欧拉角 Rx	double	$\geq -180, < 180$	欧拉角 Rx
Ry	欧拉角 Ry	double	$\geq -180, < 180$	欧拉角 Ry
Rz	欧拉角 Rz	double	$\geq -180, < 180$	欧拉角 Rz

✓ 发送成功返回消息：RPY2Quaternion,OK,dQuaW,dQuaX,dQuaY,dQuaZ,;

返回值	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

✓ 发送失败返回消息：RPY2Quaternion,Fail,ErrorCode,;

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

### 4.7.2.2. 示例

// 欧拉角转四元素

RPY2Quaternion,0,180,0,180,;

// 成功返回

RPY2Quaternion,OK,180.000,0.000,180.000,0,;

// 失败返回

RPY2Quaternion,Fail,20018,;

### 4.7.3.GetInverseKin(PCS2ACS)

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

- ✓ 发送消息：GetInverseKin,nRbtID,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
dX- dRz	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： 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	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[°]
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):

✓ 发送成功返回消息: GetInverseKin,OK,dTargetJ1,dTargetJ2,dTargetJ3,dTargetJ4,dTargetJ5,dTargetJ6,;

✓ 发送失败返回消息: GetInverseKin,Fail,ErrorCode,;

#### 4.7.3.2. 示例

GetInverseKin,Fail,20018,;

## 4.7.4.GetForwardKin(ACS2PCS)

4.7.4.1. 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。

- ✓ 发送消息：GetForwardKin,nRbtID, 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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

- ✓ 发送成功返回消息：GetForwardKin,OK,dX,dY,dZ,dRx,dRy,dRz,;



返回值	名称	数据类型	有效范围	内容
dX- dRz	计算结果	double	-	计算正解的目标迪卡尔位置： 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[°]

✓ 发送失败返回消息: GetForwardKin,Fail,ErrorCode,;

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

#### 4.7.4.2. 示例

// 计算 0,0,90,0,90,0 的正解结果, 工具坐标和用户坐标均为默认 0,

GetForwardKin,0,0,0,90,0,90,0,0,0,0,0,0,0,0,0,0,0,;

// 成功返回, 计算结果为 420,0,445,-180,0,180

GetForwardKin,OK,420.000,-0.000,445.000,-180.000,0.000,180.000,;

// 失败返回

GetForwardKin,Fail,20018,;

## 4.7.5.Base2UcsTcp

4.7.5.1. 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。

- ✓ 发送消息：Base2UcsTcp,nRbtID,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
dX- dRz	基于基座坐标系的迪卡尔坐标位置	double	-	需要转换的迪卡尔位置： 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[°]

✓ 发送成功返回消息：Base2UcsTcp,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	名称	数据类型	有效范围	内容
dTargetX- dTargetRz	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[°]

✓ 发送失败返回消息：Base2UcsTcp,Fail,ErrorCode,;

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

#### 4.7.5.2. 示例

// 计算基座坐标 420,0,445,180,0,180 在用户坐标 420,100,445,180,0,180 下的坐标值

Base2UcsTcp,0,420,0,445,180,0,180,0,0,0,0,0,420,100,445,180,0,180,;

// 成功返回,计算结果为 0,-100,0,0,0,0

Base2UcsTcp,OK,0.000,-100.000,0.000,0.000,-0.000,0.000,;

// 失败返回

Base2UcsTcp,Fail,20018,;

## 4.7.6.UcsTcp2Base

4.7.6.1. 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。

- ✓ 发送消息：UcsTcp2Base,nRbtID,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
dX- dRz	基于用户坐标系的迪卡尔坐标位置	double	-	需要转换的迪卡尔位置： 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[°]

✓ 发送成功返回消息：UcsTcp2Base,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	名称	数据类型	有效范围	内容
dTargetX- dTargetRz	目标迪卡尔 坐标位置	double	-	指定用户坐标系和工具坐标系下的迪卡尔 坐标位置：  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[°]

✓ 发送失败返回消息：Base2UcsTcp,Fail,ErrorCode,;

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

#### 4.7.6.2. 示例

// 计算基座坐标 0,-100,0,0,0,0 在用户坐标 420,100,445,180,0,180 下的坐标值

UcsTcp2Base,0,0,-100,0,0,0,0,0,0,0,0,420,100,445,180,0,180,;

// 成功返回

UcsTcp2Base,OK,420.000,0.000,445.000,180.000,-0.000,180.000,;

// 失败返回

UcsTcp2Base,Fail,20007,;

## 4.7.7.PoseAdd

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

- ✓ 发送消息：PoseAdd,nRbtID,dPose1\_X,dPose1\_Y,dPose1\_Z,dPose1\_Rx,dPose1\_Ry,dPose1\_Rz,  
dPose2\_X,dPose2\_Y,dPose2\_Z,dPose2\_Rx,dPose2\_Ry,dPose2\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

- ✓ 发送成功返回消息：PoseAdd,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	名称	数据类型	有效范围	内容
dTargetX- dTargetRz	目标迪卡尔 坐标位置	double	-	指定用户坐标系和工具坐标系下的迪卡尔 坐标位置：  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[°]
--	--	--	--	-------------------------

✓ 发送失败返回消息: PoseAdd,Fail,ErrorCode,;

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

#### 4.7.7.2. 示例

```
// 计算基座坐标 Pose1 和 Pose2 坐标相加
PoseAdd,0,420,100,445,180,0,180,420,100,445,180,0,180,;

// 成功返回,计算结果为 840.000,200.000,890.000,0.000,-0.000,0.000
PoseAdd,OK,840.000,200.000,890.000,0.000,-0.000,0.000,;

// 失败返回
PoseAdd,Fail,20007,;
```

## 4.7.8.PoseSub

4.7.8.1. 描述：点位减法计算，以第二个点为参考点。

- ✓ 发送消息：PoseSub,nRbtID,dPose1\_X,dPose1\_Y,dPose1\_Z,dPose1\_Rx,dPose1\_Ry,dPose1\_Rz,  
dPose2\_X,dPose2\_Y,dPose2\_Z,dPose2\_Rx,dPose2\_Ry,dPose2\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

- ✓ 发送成功返回消息：PoseSub,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	名称	数据类型	有效范围	内容
dTargetX- dTargetRz	目标迪卡尔 坐标位置	double	-	指定用户坐标系和工具坐标系下的迪卡尔 坐标位置：  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[°]
--	--	--	--	-------------------------

✓ 发送失败返回消息: PoseSub,Fail,ErrorCode,;

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

#### 4.7.8.2. 示例

// 计算基座坐标 Pose1 和 Pose2 坐标相减

PoseSub,0,420,0,445,180,0,180,420,100,445,180,0,180,;

// 成功返回, 计算结果为 0.000,-100.000,0.000,0.000,-0.000,0.000

PoseSub,OK,0.000,-100.000,0.000,0.000,-0.000,0.000,;

// 失败返回

PoseSub,Fail,20007,;

## 4.7.9.PoseTrans

### 4.7.9.1. 描述：坐标变换。

- ✓ 发送消息：PoseTrans,nRbtID,dPose1\_X,dPose1\_Y,dPose1\_Z,dPose1\_Rx,dPose1\_Ry,dPose1\_Rz,  
dPose2\_X,dPose2\_Y,dPose2\_Z,dPose2\_Rx,dPose2\_Ry,dPose2\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

- ✓ 发送成功返回消息：PoseTrans,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	名称	数据类型	有效范围	内容
dTargetX- dTargetRz	目标迪卡尔 坐标位置	double	-	指定用户坐标系和工具坐标系下的迪卡尔坐标位置：  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[°]

✓ 发送失败返回消息: PoseTrans,Fail,ErrorCode,;

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

#### 4.7.9.2. 示例

// 计算基座坐标 Pose1 和 Pose2 坐标转换

PoseTrans,0,420,0,445,180,0,180,420,100,445,180,0,180,;

// 成功返回,计算结果为 0.000,100.000,0.000,0.000,-0.000,0.000

PoseTrans,OK,0.000,100.000,0.000,0.000,-0.000,0.000,;

// 失败返回

PoseTrans,Fail,20007,;

## 4.7.10.PoseInverse

### 4.7.10.1. 描述：坐标逆变换。

✓ 发送消息：PoseInverse,nRbtID,dPose1\_X,dPose1\_Y,dPose1\_Z,dPose1\_Rx,dPose1\_Ry,dPose1\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

✓ 发送成功返回消息：PoseInverse,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	名称	数据类型	有效范围	内容
dTargetX- dTargetRz	目标迪卡尔 坐标位置	double	-	指定用户坐标系和工具坐标系下的迪卡尔 坐标位置：  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[°]

✓ 发送失败返回消息：PoseInverse,Fail,ErrorCode,;

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

### 4.7.10.2. 示例

// 计算基座坐标 Pose1 坐标逆转换

PoseInverse,0,420,0,445,180,0,180,;

---

// 成功返回,计算结果为 420,0,445,180,0,180

PoseInverse,OK,420.000,-0.000,445.000,180.000,-0.000,180.000,;

// 失败返回

PoseInverse,Fail,20007,;

## 4.7.11.PoseDist

### 4.7.11.1. 描述：计算点位距离。

- ✓ 发送消息：PoseDist,nRbtID,dPose1\_X,dPose1\_Y,dPose1\_Z,dPose1\_Rx,dPose1\_Ry,dPose1\_Rz,  
dPose2\_X,dPose2\_Y,dPose2\_Z,dPose2\_Rx,dPose2\_Ry,dPose2\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

- ✓ 发送成功返回消息：PoseDist,OK,dDistance,dAngle,;

返回值	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	姿态偏差	double	-	姿态偏差，单位[°]

- ✓ 发送失败返回消息：PoseDist,Fail,ErrorCode,;

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

---

4.7.11.2. 示例

// 计算基座坐标 Pose1 和 Pose2 坐标距离

PoseDist,0,420,0,445,180,0,180,420,100,445,180,0,180,;

// 成功返回,计算结果为 100.000,0

PoseDist,OK,100.000,0,;

// 失败返回

PoseDist,Fail,20007,;

## 4.7.12.PoseInterpolate

### 4.7.12.1. 描述：空间位置直线插补计算。

- ✓ 发送消息：PoseInterpolate,nRbtID,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

- ✓ 发送成功返回消息：PoseInterpolate,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	名称	数据类型	有效范围	内容
dTargetX- dTargetRz	目标迪卡尔 坐标位置	double	-	指定用户坐标系和工具坐标系下的迪卡尔 坐标位置：  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[°]

✓ 发送失败返回消息: PoseInterpolate,Fail,ErrorCode,;

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

#### 4.7.12.2. 示例

// 计算插补

PoseInterpolate,0,420,0,445,0,0,180,420,100,445,180,0,180,0.5,;

// 成功返回

PoseInterpolate,OK,420.000,50.000,445.000,90.000,-0.000,180.000,;

// 失败返回

PoseInterpolate,Fail,20007,;

### 4.7.13.PoseDefdFrame

4.7.13.1. 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。

- ✓ 发送消息：PoseDefdFrame,nRbtID,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]
--	--	--	--	--

✓ 发送成功返回消息: PoseDefdFrame,OK,dTargetX,dTargetY,dTargetZ,dTargetRx,dTargetRy,dTargetRz,;

返回值	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[°]

✓ 发送失败返回消息: PoseDefdFrame,Fail,ErrorCode,;

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

#### 4.7.13.2. 示例

// 计算旋转用户坐标

PoseDefdFrame,0,420,0,445,420,50,445,470,50,445,420,0,445,420,0,495,420,50,495,;

// 成功返回

PoseDefdFrame,OK,420.000,-445.000,25.000,0.000,-90.000,-90.000,;

// 失败返回

PoseDefdFrame,Fail,20007,;

## 4.8.工具坐标与用户坐标读写指令

### 4.8.1.SetCurTCP

4.8.1.1. 描述：设置当前工具坐标系，不写入配置文件，重启后失效。

✓ 发送消息：SetCurTCP,nRbtID,dTcp\_X,dTcp\_Y,dTcp\_Z,dTcp\_Rx,dTcp\_Ry,dTcp\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

✓ 发送成功返回消息：SetCurTCP,OK,;

✓ 发送失败返回消息：SetCurTCP,Fail,ErrorCode,;

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

4.8.1.2. 示例

// 设置当前 TCP 值

```
SetCurTCP,0,60,80,120,50,0,0,;
```

// 成功返回

```
SetCurTCP,OK,;
```

// 失败返回

```
SetCurTCP,Fail,20007,;
```

## 4.8.2.SetCurUCS

4.8.2.1. 描述：设置当前用户坐标系，不写入配置文件，重启后失效。

✓ 发送消息：SetCurUCS,nRbtID,dUcs\_X,dUcs\_Y,dUcs\_Z,dUcs\_Rx,dUcs\_Ry,dUcs\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

✓ 发送成功返回消息：SetCurUCS,OK,;

✓ 发送失败返回消息：SetCurUCS,Fail,ErrorCode,;

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

### 4.8.2.2. 示例

// 设置当前 UCS 值

```
SetCurUCS,0,60,80,120,50,0,0,;
```

// 成功返回

```
SetCurUCS,OK,;
```

// 失败返回

```
SetCurUCS,Fail,20007,;
```

### 4.8.3.ReadCurTCP

4.8.3.1. 描述：读取当前设置的工具坐标系值。

✓ 发送消息：ReadCurTCP,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCurTCP,OK,dTcp\_X,dTcp\_Y,dTcp\_Z,dTcp\_Rx,dTcp\_Ry,dTcp\_Rz,;

返回值	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[°]

✓ 发送失败返回消息：ReadCurTCP,Fail,ErrorCode,;

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

4.8.3.2. 示例

// 设置当前 TCP 值

ReadCurTCP,0,;

// 成功返回

ReadCurTCP,OK,60.000,80.000,120.000,50.000,0.000,0.000,;

// 失败返回

ReadCurTCP,Fail,20007,;

## 4.8.4.ReadCurUCS

4.8.4.1. 描述：读取当前设置的用户坐标系值。

✓ 发送消息：ReadCurUCS,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadCurUCS,OK,dUcs\_X,dUcs\_Y,dUcs\_Z,dUcs\_Rx,dUcs\_Ry,dUcs\_Rz,;

返回值	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[°]

✓ 发送失败返回消息：ReadCurUCS,Fail,ErrorCode,;

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

4.8.4.2. 示例

// 读取当前 UCS 值

ReadCurUCS,0,;

// 成功返回

ReadCurUCS,OK,60.000,80.000,120.000,50.000,0.000,0.000,;

// 失败返回

ReadCurUCS,Fail,20007,;

## 4.8.5.SetTCPByName

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

✓ 发送消息：SetTCPByName,nRbtID,sTcpName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：SetTCPByName,OK,;

✓ 发送失败返回消息：SetTCPByName,Fail,ErrorCode,;

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

### 4.8.5.2. 示例

// 设置 TCP 坐标

SetTCPByName,0,TCP\_1,;

// 成功返回

SetTCPByName,OK,;

// 失败返回, 指定的 TCP 不存在

SetTCPByName,Fail,20029,;



## 4.8.6.SetUCSByName

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

✓ 发送消息：SetUCSByName,nRbtID,sUcsName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：SetUCSByName,OK,;

✓ 发送失败返回消息：SetUCSByName,Fail,ErrorCode,;

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

### 4.8.6.2. 示例

// 设置 UCS 坐标

SetUCSByName,0,Base,;

// 成功返回

SetUCSByName,OK,;

// 失败返回, 指定的 UCS 不存在

SetUCSByName,Fail,20030,;

## 4.8.7.ReadTCPByName

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

✓ 发送消息：ReadTCPByName,nRbtID,sTcpName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：ReadTCPByName,OK,dTcp\_X,dTcp\_Y,dTcp\_Z,dTcp\_Rx,dTcp\_Ry,dTcp\_Rz,;

返回值	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[°]

✓ 发送失败返回消息：ReadTCPByName,Fail,ErrorCode,;

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

### 4.8.7.2. 示例

// 读取指定名称的 TCP 值

```
ReadTCPByName,0,TCP_1,;
```

// 成功返回

```
ReadTCPByName,OK,60.000,80.000,120.000,50.000,0.000,0.000,;
```

// 失败返回

```
ReadTCPByName,Fail,20029,;
```

### 4.8.8.ReadUCSByName

4.8.8.1. 描述:通过名称读取指定用户坐标系值,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。

✓ 发送消息: ReadUCSByName,nRbtID,sUcsName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息: ReadUCSByName,OK,dUcs\_X,dUcs\_Y,dUcs\_Z,dUcs\_Rx,dUcs\_Ry,dUcs\_Rz,;

返回值	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[°]

✓ 发送失败返回消息: ReadUCSByName,Fail,ErrorCode,;

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

#### 4.8.8.2. 示例

// 读取指定名称的 UCS 值

```
ReadUCSByName,0,Point_1,;
```

// 成功返回

```
ReadUCSByName,OK,420.000, 0.000,445.000,180.000,0.000,180.000,;
```

// 失败返回

```
ReadUCSByName,Fail,20030,;
```

## 4.9.力控控制指令

### 4.9.1.SetForceControlState

4.9.1.1. 描述：设置力控状态，执行命令后机器人跳转到运动状态。

✓ 发送消息：SetForceControlState,nRbtID,nState,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：SetForceControlState,OK,;

✓ 发送失败返回消息：SetForceControlState,Fail,ErrorCode,;

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

#### 4.9.1.2. 示例

// 设置开启力控

SetForceControlState,0,1,;

// 成功返回

SetForceControlState,OK,;

// 失败返回

SetForceControlState,Fail,40020,;

## 4.9.2.ReadFTControlState

4.9.2.1. 描述：读取当前力控状态。

✓ 发送消息：ReadFTControlState,nRbtID,nState,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadFTControlState,OK,nState,;

返回值	名称	数据类型	有效范围	内容
nState	力控状态	int	0~3	读取力控状态： 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

✓ 发送失败返回消息：ReadFTControlState,Fail,ErrorCode,;

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

4.9.2.2. 示例

// 读取力控状态

ReadFTControlState,0,;

// 成功返回

ReadFTControlState,OK,1,;

// 失败返回

ReadFTControlState,Fail,40020,;

### 4.9.3.SetForceToolCoordinateMotion

4.9.3.1. 描述：设置力控坐标系方向为 Tool 坐标方向模式。

✓ 发送消息：SetForceToolCoordinateMotion,nRbtID,nMode,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：SetForceToolCoordinateMotion,OK,;

✓ 发送失败返回消息：SetForceToolCoordinateMotion,Fail,ErrorCode,;

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

4.9.3.2. 示例

// 设置力控坐标系控制方向为 Tool 坐标系方向

SetForceToolCoordinateMotion,0,1,;

// 成功返回

SetForceToolCoordinateMotion,OK,;

// 失败返回

SetForceToolCoordinateMotion,Fail,40020,;

## 4.9.4.GrpFCInterrupt

4.9.4.1. 描述：暂停力控运动，仅暂停力控功能，不暂停运动和脚本。

✓ 发送消息：GrpFCInterrupt,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：GrpFCInterrupt,OK,;

✓ 发送失败返回消息：GrpFCInterrupt,Fail,ErrorCode,;

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

4.9.4.2. 示例

// 设置力控暂停

GrpFCInterrupt,0,;

// 成功返回

GrpFCInterrupt,OK,;

// 失败返回

GrpFCInterrupt,Fail,20018,;

## 4.9.5.GrpFCContinue

4.9.5.1. 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。

✓ 发送消息：GrpFCContinue,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：GrpFCContinue,OK,;

✓ 发送失败返回消息：GrpFCContinue,Fail,ErrorCode,;

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

4.9.5.2. 示例

// 设置力控继续

GrpFCContinue,0,;

// 成功返回

GrpFCContinue,OK,;

// 失败返回

GrpFCContinue,Fail,20018,;



## 4.9.6.SetForceZero

4.9.6.1. 描述：力控清零，在原有参数的基础上重新标定力传感器。

✓ 发送消息：SetForceZero,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：SetForceZero,OK,;

✓ 发送失败返回消息：SetForceZero,Fail,ErrorCode,;

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

4.9.6.2. 示例

// 清零力传感器数据

SetForceZero,0,;

// 成功返回

SetForceZero,OK,;

// 失败返回

SetForceZero,Fail,20018,;

## 4.9.7.HRSetMaxSearchVelocities

4.9.7.1. 描述：设置力控探寻的最大速度。

- ✓ 发送消息：HRSetMaxSearchVelocities,nRbtID,dMaxLinearVelocity,dMaxAngularVelocity,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]

- ✓ 发送成功返回消息：HRSetMaxSearchVelocities,OK,;

- ✓ 发送失败返回消息：HRSetMaxSearchVelocities,Fail,ErrorCode,;

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

4.9.7.2. 示例

// 设置力控探寻的最大速度

HRSetMaxSearchVelocities,0,5,5,;

// 成功返回

HRSetMaxSearchVelocities,OK,;

// 失败返回

HRSetMaxSearchVelocities,Fail,20018,;

## 4.9.8.HRSetControlFreedom

4.9.8.1. 描述：设置力控探寻自由度。

✓ 发送消息：HRSetControlFreedom,nRbtID,nX,nY,nZ,nRx,nRy,nRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：HRSetControlFreedom,OK,;

✓ 发送失败返回消息：HRSetControlFreedom,Fail,ErrorCode,;

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

4.9.8.2. 示例

// 设置力控探寻控制自由度

HRSetControlFreedom,0,0,0,1,0,0,0,;

// 成功返回

HRSetControlFreedom,OK,;

// 失败返回

HRSetControlFreedom,Fail,20018,;

## 4.9.9.HRSetForceControlStrategy

### 4.9.9.1. 描述：设置力控控制策略。

✓ 发送消息：HRSetForceControlStrategy,nRbtID,nState,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：HRSetForceControlStrategy,OK,;

✓ 发送失败返回消息：HRSetForceControlStrategy,Fail,ErrorCode,;

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

### 4.9.9.2. 示例

// 设置力控控制策略

HRSetForceControlStrategy,0,1,;

// 成功返回

HRSetForceControlStrategy,OK,;

// 失败返回

HRSetForceControlStrategy,Fail,20018,;

## 4.9.10.SetFTPPosition

4.9.10.1. 描述：设置力传感器中心相对于法兰盘的安装位置和姿态。

✓ 发送消息：SetFTPPosition,nRbtID,dX,dY,dZ,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

✓ 发送成功返回消息：SetFTPPosition,OK,;

✓ 发送失败返回消息：SetFTPPosition,Fail,ErrorCode,;

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

### 4.9.10.2. 示例

// 设置力传感器安装位置和姿态

SetFTPPosition,0,0,0,110,0,0,60,;

// 成功返回

SetFTPPosition,OK,;

// 失败返回

SetFTPPosition,Fail,20018,;

## 4.9.11.HRSetPIDControlParams

4.9.11.1. 描述：设置力控探寻 PID 参数。

✓ 发送消息：HRSetPIDControlParams,nRbtID,dFp,dFi,dFd,dTp,dTi,dTd,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：HRSetPIDControlParams,OK,;

✓ 发送失败返回消息：HRSetPIDControlParams,Fail,ErrorCode,;

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

4.9.11.2. 示例

// 设置 PID 参数

HRSetPIDControlParams,0,1.0,0.1,0,1.0,0.1,0,;

// 成功返回

HRSetPIDControlParams,OK,;

// 失败返回

HRSetPIDControlParams,Fail,20018,;

## 4.9.12.HRSetMassParams

4.9.12.1. 描述：设置惯量控制参数。

✓ 发送消息：HRSetMassParams,nRbtID,dX,dY,dZ,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：HRSetMassParams,OK,;

✓ 发送失败返回消息：HRSetMassParams,Fail,ErrorCode,;

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

4.9.12.2. 示例

// 设置惯量控制参数

HRSetMassParams,0,40,40,40,10,10,10,;

// 成功返回

HRSetMassParams,OK,;

// 失败返回

HRSetMassParams,Fail,20007,;

### 4.9.13.HRSetDampParams

4.9.13.1. 描述：设置阻尼(b)控制参数。

✓ 发送消息：HRSetDampParams,nRbtID,dX,dY,dZ,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：HRSetDampParams,OK,;

✓ 发送失败返回消息：HRSetDampParams,Fail,ErrorCode,;

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

4.9.13.2. 示例

// 设置阻尼控制器参数

HRSetDampParams,0,800,800,800,40,40,40,;

// 成功返回

HRSetDampParams,OK,;

// 失败返回

HRSetDampParams,Fail,20007,;



## 4.9.14.HRSetStiffParams

4.9.14.1. 描述：设置刚度(k)控制参数。

✓ 发送消息：HRSetStiffParams,nRbtID,dX,dY,dZ,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：HRSetStiffParams,OK,;

✓ 发送失败返回消息：HRSetStiffParams,Fail,ErrorCode,;

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

4.9.14.2. 示例

// 设置刚度控制参数

HRSetStiffParams,0,1000,1000,1000,100,100,100,;

// 成功返回

HRSetStiffParams,OK,;

// 失败返回

HRSetStiffParams,Fail,20007,;

## 4.9.15.HRSetControlGoal

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

- ✓ 发送消息：HRSetControlGoal,nRbtID,dX,dY,dZ,dRx,dRy,dRz,dDistance1,dDistance2,dDistance3,dDistance4,dDistance5, dDistance6,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]
dDistance1_ dDistance6	力控探寻距离	double	-	暂未启用，默认：0

- ✓ 发送成功返回消息：HRSetControlGoal,OK,;

- ✓ 发送失败返回消息：HRSetControlGoal,Fail,ErrorCode,;

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

4.9.15.2. 示例

// 设置控制目标力和目标距离

HRSetControlGoal,0,0,0,10,0,0,0,0,0,0,0,0,;

// 成功返回

HRSetControlGoal,OK,;

// 失败返回

HRSetControlGoal,Fail,20007,;

## 4.9.16.HRSetForceDataLimit

4.9.16.1. 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。

- ✓ 发送消息：HRSetForceDataLimit,nRbtID,dMax\_X,dMax\_Y,dMax\_Z,dMax\_Rx,dMax\_Ry,dMax\_Rz,  
dMin\_X,dMin\_Y,dMin\_Z,dMin\_Rx,dMin\_Ry,dMin\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]

- ✓ 发送成功返回消息：HRSetForceDataLimit,OK,;

- ✓ 发送失败返回消息：HRSetForceDataLimit,Fail,ErrorCode,;

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

4.9.16.2. 示例

// 设置力控限制范围

HRSetForceDataLimit,0,500,500,500,500,500,-500,-500,-500,-500,-500,;

// 成功返回

---

HRSetForceDataLimit,OK,;

// 失败返回

HRSetForceDataLimit,Fail,20007,;

## 4.9.17.HRSetForceDistanceLimit

4.9.17.1. 描述：设置力控形变范围。

✓ 发送消息：HRSetForceDistanceLimit,nRbtID,dAllowDistance,dStrengthLevel,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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；

✓ 发送成功返回消息：HRSetForceDistanceLimit,OK,;

✓ 发送失败返回消息：HRSetForceDistanceLimit,Fail,ErrorCode,;

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

4.9.17.2. 示例

// 设置力控形变范围

HRSetForceDistanceLimit,0,10,2,;

// 成功返回

HRSetForceDistanceLimit,OK,;

// 失败返回

HRSetForceDistanceLimit,Fail,20007,;

## 4.9.18.SetForceFreeDriveMode

4.9.18.1. 描述：设置开启或者关闭力控自由驱动模式。

✓ 发送消息：SetForceFreeDriveMode,nRbtID,nMode,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
nMode	是否开启	int	0/1	0：关闭 1：开启

✓ 发送成功返回消息：SetForceFreeDriveMode,OK,;

✓ 发送失败返回消息：SetForceFreeDriveMode,Fail,ErrorCode,;

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

4.9.18.2. 示例

// 设置开启力控自由驱动

SetForceFreeDriveMode,0,1,;

// 成功返回

SetForceFreeDriveMode,OK,;

// 失败返回

SetForceFreeDriveMode,Fail,20007,;

## 4.9.19.ReadFTCabData

4.9.19.1. 描述：读取标定后的力传感器数据。

✓ 发送消息：ReadFTCabData,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0

✓ 发送成功返回消息：ReadFTCabData,OK,dX,dY,dZ,dRx,dRy,dRz,;

返回值	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]

✓ 发送失败返回消息：ReadFTCabData,Fail,ErrorCode,;

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

4.9.19.2. 示例

// 读取标定后的力传感器值

ReadFTCabData,0,;

// 成功返回

ReadFTCabData,OK,0.000,0.000,0.000,0.000,0.000,0.000,;

// 失败返回

ReadFTCabData,Fail,20007,;

## 4.10.通用运动类控制指令

### 4.10.1.MoveRelJ

4.10.1.1.描述：关节相对运动。

✓ 发送消息：MoveRelJ,nRbtID,nAxisId,nDirection,dDistance;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0
nAxisId	轴 id	int	0~5	运动的目标轴 ID, 对应关节 J1-J6
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

✓ 发送成功返回消息：MoveRelJ,OK,;

✓ 发送失败返回消息：MoveRelJ,Fail,ErrorCode,;

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

4.10.1.2. 示例

// 关节2 正方向相对运动 10°

MoveRelJ,0,1,1,10,;

// 成功返回

MoveRelJ,OK,;

// 失败返回

MoveRelJ,Fail,20018,;



## 4.10.2.MoveRelL

### 4.10.2.1. 空间相对运动。

✓ 发送消息：MoveRelL,nRbtID,nAxisId,nDirection,dDistance, nToolMotion,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0
nAxisId	轴 id	int	0~5	运动的目标轴 ID, 对应空间坐标 X-Rz
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

\* 注: nToolMotion 未设置时按照点动页面选择的用户坐标运动。

✓ 发送成功返回消息：MoveRelL,OK,;

✓ 发送失败返回消息：MoveRelL,Fail,ErrorCode,;

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

### 4.10.2.2. 示例

// Y 轴正方向相对运动 10mm, 使用 Tool 坐标

```
MoveRelL,0,1,1,10,1,;
```

// 成功返回

```
MoveRelL,OK,;
```

// 失败返回

```
MoveRelL,Fail,20018,;
```

// Y 轴正方向相对运动 10mm, 使用页面选择的坐标

```
MoveRelL,0,1,1,10,;
```

### 4.10.3.WayPointRel

#### 4.10.3.1. 描述：路点相对运动。

- ✓ 发送消息：WayPointRel,nRbtID,nType,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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- nAxisMask_6	是否运动	int	0/1	各轴/各方向是否运动 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[°/s²], 空间运动时 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]
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”.
----------	-------	--------	---	---------------------------------------

✓ 发送成功返回消息: WayPointRel,OK,;

✓ 发送失败返回消息: WayPointRel,Fail,ErrorCode,;

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

#### 4.10.3.2. 示例

// 空间相对运动-不使用点位列表点位, +X 叠加运动 10mm,-Y 叠加运动 10mm

WayPointRel,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,10,-10,0,0,0,0,TCP,Base,500.0,2500.0,50.0,0,0,0,0,0,;

// 成功返回

WayPointRel,OK,;

// 失败返回

WayPointRel,Fail,20007,;

// 关节相对运动-不使用点位列表点位, J1 轴叠加运动 10°

WayPointRel,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,10,-10,0,0,0,0,TCP,Base,500.0,2500.0,50.0,0,0,0,0,0,;

// 关节相对运动-不使用点位列表点位, J1 轴运动到 10°

WayPointRel,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,10,-10,0,0,0,0,TCP,Base,500.0,2500.0,50.0,0,0,0,0,0,;

// 关节相对运动-使用点位列表点位, J1 轴叠加运动 10°

WayPointRel,0,0,1,553.646,-0.055,325.096,-179.990,-0.004,-89.995,-0.004,-21.894,79.364,-0.004,78.732,  
-90.009,1,1,0,0,0,0,0,10,-10,0,0,0,0,TCP,Base,500.0,2500.0,50.0,0,0,0,0,0,;

// 关节相对运动-使用点位列表点位, J1 轴运动到 10°

WayPointRel,0,0,1,553.646,-0.055,325.096,-179.990,-0.004,-89.995,-0.004,-21.894,79.364,-0.004,78.732, -  
90.009,0,1,0,0,0,0,0,10,-10,0,0,0,0,TCP,Base,500.0,2500.0,50.0,0,0,0,0,0,;

## 4.10.4.WayPointEx

### 4.10.4.1. 描述：路点运动。

- ✓ 发送消息：WayPointEx,nRbtID,dX,dY,dZ,dRx,dRy,dRz,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,dUcs\_X,dUcs\_Y,dUcs\_Z,dUcs\_Rx,dUcs\_Ry,dUcs\_Rz,dTcp\_X,dTcp\_Y,dTcp\_Z,dTcp\_Rx,dTcp\_Ry,dTcp\_Rz,dVelocity,dAcc,dRadius,nMoveType,nIsUseJoint,nIsSeek, nIOBit, nIOState, strCmdID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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	-	<p>目标空间坐标所处的工具坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可设置为 0:</p> <p>dTcp_X: X 坐标, 单位[mm]</p> <p>dTcp_Y: Y 坐标, 单位[mm]</p> <p>dTcp_Z: Z 坐标, 单位[mm]</p> <p>dTcp_Rx: Rx 坐标, 单位[°]</p> <p>dTcp_Ry: Ry 坐标, 单位[°]</p> <p>dTcp_Rz: Rz 坐标, 单位[°]</p>
dUcs_X- dUcs_Rz	用户坐标值	double	-	<p>目标空间坐标所处的用户坐标系名称, 与示教器页面的名称对应, 当 nIsUseJoint=1 时无效, 可设置为 0:</p> <p>dUcs_X: X 坐标, 单位[mm]</p> <p>dUcs_Y: Y 坐标, 单位[mm]</p> <p>dUcs_Z: Z 坐标, 单位[mm]</p> <p>dUcs_Rx: Rx 坐标, 单位[°]</p> <p>dUcs_Ry: Ry 坐标, 单位[°]</p> <p>dUcs_Rz: Rz 坐标, 单位[°]</p>
dVelocity	速度	double	-	<p>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]</p>
dAcc	加速度	double	-	<p>运动最大加速度, 关节运动时单位[°/s²], 空间运动时 X, Y, Z 单位[mm/s²], Rx, Ry, Rz 单位[°/s²]</p>
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	<p>是否使用关节角度作为目标点, 如果 nMoveType=0 时, 则 nIsUseJoint 有效:</p> <p>0: 不使用关节角度</p>

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

✓ 发送成功返回消息: WayPointEx,OK,;

✓ 发送失败返回消息: WayPointEx,Fail,ErrorCode,;

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

#### 4.10.4.2. 示例

// 关节运动-不使用关节点位

WayPointEx,0,420,0,445,180,0,180,0,0,90,0,90,0,0,0,0,0,0,0,0,0,0,50,360,0,0,0,0,0,0,;

// 成功返回

WayPointEx,OK,;

// 失败返回

WayPointEx,Fail,20007,;

// 关节运动-使用关节点位

WayPointEx,0,420,0,445,180,0,180,0,0,90,0,90,0,0,0,0,0,0,0,0,0,0,50,360,0,0,1,0,0,0,0,;

// 直线运动

WayPointEx,0,420,0,445,180,0,180,0,0,90,0,90,0,0,0,0,0,0,0,0,0,0,50,360,0,1,0,0,0,0,0,;

## 4.10.5.WayPoint

### 4.10.5.1. 描述：路点运动。

- ✓ 发送消息：WayPoint,nRbtID,dX,dY,dZ,dRx,dRy,dRz,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,sTcpName,sUcsName,dVelocity,dAcc,dRadius,nMoveType,nIsUseJoint,nIsSeek,nIOBit,nIOState,strCmdID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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 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[°/s²], 空间运动时 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]
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: WayPointEx 需要设置工具坐标与用户坐标具体的值; WayPoint 使用示教器示教的工具坐标与用户坐标名称。

✓ 发送成功返回消息: WayPoint,OK,;

✓ 发送失败返回消息: WayPoint,Fail,ErrorCode,;

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

## 4.10.5.2. 示例

// 关节运动-不使用关节点位

WayPoint,0,420,0,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,0,0,0,0,0,;

// 成功返回

WayPoint,OK,;

// 失败返回

WayPoint,Fail,20007,;

// 关节运动-使用关节点位

WayPoint,0,420,0,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,0,1,0,0,0,0,;

// 直线运动

WayPoint,0,420,0,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,1,0,0,0,0,0,;

## 4.10.6.WayPoint2

### 4.10.6.1. 描述：路点运动。

- ✓ 发送消息：WayPoint2,nRbtID,dEndPos\_X,dEndPos\_Y,dEndPos\_Z,dEndPos\_Rx,dEndPos\_Ry,dEndPos\_Rz,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,sTcpName,sUcsName,dVelocity,dAcc,dRadius,nMoveType,nIsUseJoint,nIsSeek,nIOBit,nIOState,dAuxPos\_X,dAuxPos\_Y,dAuxPos\_Z,dAuxPos\_Rx,dAuxPos\_Ry,dAuxPos\_Rz,strCmdID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
nMoveType	运动类型	int	0/2	0：关节运动 1：直线运动 2：圆弧运动
dEndPos_X- dEndPos_Rz	空间目标位置	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：作为圆弧的目标位置 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：作为圆弧的经过位置 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 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[°/s²], 空间运动时 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]
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: WayPoint2 支持直线与圆弧过渡不减速功能; WayPoint 仅支持直线与直线过渡不减速。

✓ 发送成功返回消息: WayPoint2,OK,;

✓ 发送失败返回消息: WayPoint2,Fail,ErrorCode,;

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

#### 4.10.6.2. 示例

// 关节运动-不使用关节点位

WayPoint2,0,470,50,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,0,0,0,0,0,420,50,445,180,0,180,;

// 成功返回

WayPoint2,OK,;

// 失败返回

WayPoint2,Fail,20007,;

// 直线运动

WayPoint2,0,470,50,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,1,0,0,0,0,420,50,445,180,0,180,;

// 圆弧运动

WayPoint2,0,470,50,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,2,0,0,0,0,420,50,445,180,0,180,;

#### 4.10.7.MoveJ(弃用)

##### 4.10.7.1.描述：关节运动。

- ✓ 发送消息：MoveJ,nRbtID,dX,dY,dZ,dRx,dRy,dRz,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,sUcsName,sTcpName,dVelocity,dAcc,dRadius,nIsUseJoint,nIsSeek, nIOBit, nIOState, strCmdID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
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 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[°/s²]，空间运动时 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]
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”.

✓ 发送成功返回消息：MoveJ,OK,;

✓ 发送失败返回消息：MoveJ,Fail,ErrorCode,;

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

---

ErrorCode	错误码	int	>0 的整型值	返回调用失败的错误码
-----------	-----	-----	---------	------------

---

## 4.10.7.2. 示例

// 关节运动-不使用关节点位

MoveJ,0,420,0,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,0,0,0,0,0,;

// 成功返回

MoveJ,OK,;

// 失败返回

MoveJ,Fail,20007,;

// 关节运动-使用关节点位

MoveJ,0,420,0,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,0,1,0,0,0,0,;



#### 4.10.8.MoveL(弃用)

4.10.8.1.描述：直线轨迹运动。

- ✓ 发送消息：MoveL,nRbtID,dX,dY,dZ,dRx,dRy,dRz,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,sUcsName,sTcpName,  
dVelocity,dAcc,dRadius,nIsSeek, nIOBit, nIOState, strCmdID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
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 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[°/s²]，空间运动时 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”。

✓ 发送成功返回消息：MoveL,OK,;

✓ 发送失败返回消息：MoveL,Fail,ErrorCode,;

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

#### 4.10.8.2. 示例

// 直线运动

MoveL,0,420,0,445,180,0,180,0,0,90,0,90,0,TCP,Base,50,360,0,1,0,0,0,;

---

// 成功返回

MoveL,OK,;

// 失败返回

MoveL,Fail,20007,;

## 4.10.9.MoveC

### 4.10.9.1. 描述：圆弧轨迹运动。

- ✓ 发送消息：MoveC,nRbtID,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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，结束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[°]
nFixedPosture	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/s²], Rx, Ry, Rz 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, 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”
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”。

✓ 发送成功返回消息: MoveC,OK,;

✓ 发送失败返回消息: MoveC,Fail,ErrorCode,;

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

#### 4.10.9.2. 示例

// 圆弧运动

---

MoveC,0,420,0,445,180,0,180,420,50,445,180,0,180,470,50,445,180,0,180,0,0,1,100,2500,1,TCP,Base,0,;

// 成功返回

MoveC,OK,;

// 失败返回

MoveC,Fail,20007,;

## 4.10.10.MoveZ

### 4.10.10.1. 描述：Z 型轨迹运动。

- ✓ 发送消息：MoveZ,nRbtID,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,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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 型轨迹确定平面位置： 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/s <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ]
dWidth	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计算 密度
nEnableDensity	是否使用密度	int	-	是否使用密度: 0: 不使用 1: 使用
nEnablePlane	是否使用平面点	int	-	是否使用平面点, 不使用时根据选择的用 户坐标确定 XYZ 平面: 0: 不使用 1: 使用
nEnableWaiTime	是否开启转折点 等待时间	int	-	是否开启转折点等待时间: 0: 不使用 1: 使用
nPosiTime	正向转折点等待 时间	int	-	正向转折点等待时间, 单位[ms]
nNegaTime	负向转折点等待 时间	int	-	负向转折点等待时间, 单位[ms]
dRadius	过渡半径	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, 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”
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”.

✓ 发送成功返回消息: MoveZ,OK,;



✓ 发送失败返回消息：MoveZ,Fail,ErrorCode,;

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

#### 4.10.10.2. 示例

// Z 型轨迹运动

MoveZ,0,420,0,445,180,0,180,420,100,445,180,0,180,470,50,445,180,0,180,500.000,2500.000,20.000,1.000,0,0,0,0,0,1.000,TCP,Base,0,;

// 成功返回

MoveZ,OK,;

// 失败返回

MoveZ,Fail,20007,;

## 4.10.11.MoveE

4.10.11.1. 描述：椭圆轨迹运动。

- ✓ 发送消息：MoveE,nRbtID,dP1\_X,dP1\_Y,dP1\_Z,dP1\_Rx,dP1\_Ry,dP1\_Rz, dP2\_X, dP2\_Y, dP2\_Z, dP2\_Rx,dP2\_Ry,dP2\_Rz,dP3\_X,dP3\_Y,dP3\_Z,dP3\_Rx,dP3\_Ry,dP3\_Rz,dP4\_X,dP4\_Y,dP4\_Z, dP4\_Rx, dP4\_Ry,dP4\_Rz,dP5\_X,dP5\_Y,dP5\_Z,dP5\_Rx,dP5\_Ry,dP5\_Rz,nOrientMode,nMoveType,dArcLength,dVelocity,dAcc,dRadius,sTcpName,sUcsName,strCmdID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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 <sup>2</sup> ], Rx, Ry, Rz 单位[°/s <sup>2</sup> ]
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, 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”

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”。
----------	-------	--------	---	--------------------------------------

✓ 发送成功返回消息：MoveE,OK,;

✓ 发送失败返回消息：MoveE,Fail,ErrorCode,;

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

#### 4.10.11.2. 示例

// 椭圆轨迹运动

MoveE,0,420,0,445,180,0,180,460,0,445,180,0,180,480,10,445,180,0,180,460,20,445,180,0,180,420,20,445,  
180,0,180,0,1,360,100,2500,0,TCP,Base,1,;

// 成功返回

MoveE,OK,;

// 失败返回

MoveE,Fail,20007,;

## 4.10.12.MoveS

4.10.12.1. 描述：阿基米德螺旋线运动，初始半径为固定 1mm。

✓ 发送消息：

MoveS,nRbtID,dSpiralIncrement,dSpiralDiameter,sTcpName,sUcsName,dVelocity,dAcc,dRadius,strCmdID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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，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”
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”。

✓ 发送成功返回消息：MoveS,OK,;

✓ 发送失败返回消息：MoveS,Fail,ErrorCode,;

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

4.10.12.2. 示例

// 螺旋轨迹运动

MoveS,0,1,5,30,30,1,TCP,Base,1,;

// 成功返回

MoveS,OK,;

// 失败返回

MoveS,Fail,20007,;

## 4.11.连续轨迹运动类控制指令

### 4.11.1.StartPushMovePath

4.11.1.1. 描述：初始化关节连续轨迹运动。

✓ 发送消息：StartPushMovePath,nRbtID,sTrackName,dSpeedRatio,dRadius,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]

✓ 发送成功返回消息：StartPushMovePath,OK,;

✓ 发送失败返回消息：StartPushMovePath,Fail,ErrorCode,;

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

4.11.1.2. 示例

// 初始化关节连续运动

StartPushMovePath,0,Path1,0.5,2,;

// 成功返回

StartPushMovePath,OK,;

// 失败返回

StartPushMovePath,Fail,20018,;

## 4.11.2.PushMovePathJ

4.11.2.1. 描述：下发关节连续运动轨迹点位。

✓ 发送消息：PushMovePathJ,nRbtID,sTrackName,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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 StartPushMovePath 后，可多次调用此函数，一般情况下点位数量需要大于 4。

✓ 发送成功返回消息：PushMovePathJ,OK,;

✓ 发送失败返回消息：PushMovePathJ,Fail,ErrorCode,;

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

4.11.2.2. 示例

// 下发关节连续运动轨迹点

PushMovePathJ,0,Path1,0,0,0,0,0,0,;

// 成功返回

PushMovePathJ,OK,;

// 失败返回

PushMovePathJ,Fail,20007,;

### 4.11.3.EndPushMovePath

4.11.3.1. 描述：轨迹下发完成并开始计算轨迹。

✓ 发送消息：EndPushMovePath,nRbtID,sTrackName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

✓ 发送成功返回消息：EndPushMovePath,OK,;

✓ 发送失败返回消息：EndPushMovePath,Fail,ErrorCode,;

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

4.11.3.2. 示例

// 结束下发关节连续运动轨迹点并开始计算

EndPushMovePath,0,Path1,;

// 成功返回

EndPushMovePath,OK,;

// 失败返回

EndPushMovePath,Fail,20007,;



## 4.11.4.MovePath

### 4.11.4.1. 描述：运动指定的轨迹。

✓ 发送消息：MovePath,nRbtID,sTrackName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

✓ 发送成功返回消息：MovePath,OK,;

✓ 发送失败返回消息：MovePath,Fail,ErrorCode,;

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

### 4.11.4.2. 示例

// 关节连续运动轨迹开始执行运动

MovePath,0,Path1,;

// 成功返回

MovePath,OK,;

// 失败返回

MovePath,Fail,20042,;

## 4.11.5.ReadMovePathState

### 4.11.5.1. 描述：读取当前的轨迹状态。

✓ 发送消息：ReadMovePathState,nRbtID,sTrackName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

✓ 发送成功返回消息：ReadMovePathState,OK,nState,;

返回值	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 StartPushMovePath 后切换到此状态 1: 轨迹示教中，调用 PushMovePathJ 后切换到此状态 2: 轨迹计算中，调用 EndPushMovePath 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。

✓ 发送失败返回消息：ReadMovePathState,Fail,ErrorCode,;

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

### 4.11.5.2. 示例

// 读取关节连续运动轨迹状态

ReadMovePathState,0,Path1,;

// 成功返回

ReadMovePathState,OK,5,;

// 失败返回

ReadMovePathState,Fail,20007,;

## 4.11.6.UpdateMovePathName

4.11.6.1. 描述：更新指定轨迹的名称。

✓ 发送消息：UpdateMovePathName,nRbtID,sTrackName,sTrackNewName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：UpdateMovePathName,OK,;

✓ 发送失败返回消息：UpdateMovePathName,Fail,ErrorCode,;

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

4.11.6.2. 示例

// 更新轨迹 Path1 名称为 NewPath1

UpdateMovePathName,0,Path1,NewPath1,;

// 成功返回

UpdateMovePathName,OK,;

// 失败返回

UpdateMovePathName,Fail,20007,;

## 4.11.7.DelMovePath

### 4.11.7.1. 描述：删除指定轨迹。

✓ 发送消息：DelMovePath,nRbtID,sTrackName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0
sTrackName	轨迹原名称	string	-	轨迹原名称

✓ 发送成功返回消息：DelMovePath,OK,;

✓ 发送失败返回消息：DelMovePath,Fail,ErrorCode,;

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

### 4.11.7.2. 示例

// 删除轨迹 Path1

DelMovePath,0,Path1,;

// 成功返回

DelMovePath,OK,;

// 失败返回

DelMovePath,Fail,20007,;

## 4.11.8.ReadSoftMotionProcess

4.11.8.1. 描述：读取当前的轨迹运动进度。

✓ 发送消息：ReadSoftMotionProgress,nRbtID,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0

\* 注：此接口仅对 MovePathL 有效, 对 MovePath 无效。

✓ 发送成功返回消息：ReadSoftMotionProgress,OK,dProcess,nIndex,;

返回值	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(0-1), >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

✓ 发送失败返回消息：ReadSoftMotionProgress,Fail,ErrorCode,;

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

4.11.8.2. 示例

// 读取 MovePathL 轨迹运动进度

ReadSoftMotionProgress,0,;

// 成功返回

ReadSoftMotionProgress,OK,0.88,100,;

// 失败返回

ReadSoftMotionProgress,Fail,20007,;

## 4.11.9.InitMovePathL

### 4.11.9.1. 描述：初始化空间轨迹运动。

✓ 发送消息：InitMovePathL,nRbtID,sTrackName,dVelocity,dAcc,dJerk,sUcsName,sTcpName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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，同一轨迹运行需共同执行 InitMovePathL, PushMovePaths, EndPushMovePath 调用 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 <sup>2</sup> ]
dJerk	轨迹运动加加速度	double	-	轨迹运动加加速度，单位[mm/s <sup>3</sup> ]
sTcpName	工具坐标名称	string	-	目标空间坐标所处的工具坐标系名称，与示教器页面的名称对应，可使用默认名称“TCP”
sUcsName	用户坐标名称	string	-	目标空间坐标所处的用户坐标系名称，与示教器页面的名称对应-可使用默认名称“Base”

✓ 发送成功返回消息：InitMovePathL,OK,;

✓ 发送失败返回消息：InitMovePathL,Fail,ErrorCode,;

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

### 4.11.9.2. 示例

// 初始化空间轨迹运动

```
InitMovePathL,0,Path1,100,2500,1000000,Base,TCP,;
```

// 成功返回

```
InitMovePathL,OK,;
```

// 失败返回

```
InitMovePathL,Fail,20007,;
```

## 4.11.10.PushMovePathL

4.11.10.1. 描述：下发运动轨迹点位。

✓ 发送消息：PushMovePathL,nRbtID,sTrackName,dX,dY,dZ,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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: Ry 坐标，单位[°] dRz: Rz 坐标，单位[°]

\* 注：调用 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。

✓ 发送成功返回消息：PushMovePathL,OK,;

✓ 发送失败返回消息：PushMovePathL,Fail,ErrorCode,;

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

4.11.10.2. 示例

// 下发运动轨迹点位

```
PushMovePathL,0,Path1,420,0,445,180,0,180,;
```

// 成功返回

```
PushMovePathL,OK,;
```

// 失败返回

```
PushMovePathL,Fail,20007,;
```

## 4.11.11.PushMovePaths

4.11.11.1. 描述：批量下发轨迹点位，调用一次可下发多个点位数据。

✓ 发送消息：PushMovePaths,nRbtID,sTrackName,dX,dY,dZ,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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 MovePath 共用 1：下发空间点位，与 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

✓ 发送成功返回消息：PushMovePaths,OK,;

✓ 发送失败返回消息：PushMovePaths,Fail,ErrorCode,;

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

4.11.11.2. 示例

// 下发空间运动轨迹点位

```
PushMovePaths,0,Path1,1,6,420,0,445,180,0,180,420,10,445,180,0,180,420,20,445,180,0,180,
420,30,445,180,0,180,420,40,445,180,0,180,420,50,445,180,0,180,;
```

// 成功返回

```
PushMovePaths,OK,;
```

// 失败返回

```
PushMovePaths,Fail,20007,;
```

\* 注：关节连续轨迹点位可以共用此接口，但 InitMovePathL, PushMovePathL, MovePathL 不可共用。

// 下发关节运动轨迹点位

```
PushMovePaths,0,Path1,0,6,420,0,445,180,0,180,420,10,445,180,0,180,420,20,445,180,0,180,
420,30,445,180,0,180,420,40,445,180,0,180,420,50,445,180,0,180,;
```



## 4.11.12.MovePathL

4.11.12.1. 描述：执行空间坐标轨迹运动。

✓ 发送消息：MovePathL,nRbtID,sTrackName,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号, 默认值=0
sTrackName	轨迹名称	string	-	轨迹名称

\* 注：调用 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。

✓ 发送成功返回消息：MovePathL,OK,;

✓ 发送失败返回消息：MovePathL,Fail,ErrorCode,;

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

4.11.12.2. 示例

// 执行空间连续轨迹运动

MovePathL,0,Path1,;

// 成功返回

MovePathL,OK,;

// 失败返回

MovePathL,Fail,20007,;

### 4.11.13.SetMovePathOverride

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

✓ 发送消息：SetMovePathOverride,nRbtID,dOverride,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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(0.01~1)

✓ 发送成功返回消息：SetMovePathOverride,OK,;

✓ 发送失败返回消息：SetMovePathOverride,Fail,ErrorCode,;

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

4.11.13.2. 示例

// 设置当前速度比为 50%

SetMovePathOverride,0,0.5,;

// 成功返回

SetMovePathOverride,OK,;

// 失败返回

SetMovePathOverride,Fail,20018,;

## 4.12.Servo 运动类控制指令

### 4.12.1.StartServo

4.12.1.1. 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。

✓ 发送消息：StartServo,nRbtID,dServoTime,dLookaheadTime,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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]

✓ 发送成功返回消息：StartServo,OK,;

✓ 发送失败返回消息：StartServo,Fail,ErrorCode,;

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

#### 4.12.1.2. 示例

// 开启机器人在线控制

StartServo,0,0.2,0.02,;

// 成功返回

StartServo,OK,;

// 失败返回

StartServo,Fail,20007,;

## 4.12.2.PushServoJ

4.12.2.1. 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。

✓ 发送消息：PushServoJ,nRbtID,dJ1,dJ2,dJ3,dJ4,dJ5,dJ6,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

✓ 发送成功返回消息：PushServoJ,OK,;

✓ 发送失败返回消息：PushServoJ,Fail,ErrorCode,;

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

### 4.12.2.2. 示例

// 下发在线关节命令位置控制

PushServoJ,0,0,0,0,0,0,0,;

// 成功返回

PushServoJ,OK,;

// 失败返回

PushServoJ,Fail,40014,;

### 4.12.3.PushServoP

4.12.3.1. 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。

- ✓ 发送消息：PushServoP,nRbtID,dX,dY,dZ,dRx,dRy,dRz,dUcs\_X,dUcs\_Y,dUcs\_Z,dUcs\_Rx,dUcs\_Ry,dUcs\_Rz,dTcp\_X,dTcp\_Y,dTcp\_Z,dTcp\_Rx,dTcp\_Ry,dTcp\_Rz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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: Rz 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[°]

- ✓ 发送成功返回消息：PushServoP,OK,;

✓ 发送失败返回消息：PushServoP,Fail,ErrorCode,;

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

#### 4.12.3.2. 示例

// 下发在线末端TCP 命令位置控制

PushServoP,0,420,0,445,180,0,180,0,0,0,0,0,0,0,0,0,0,0,;

// 成功返回

PushServoP,OK,;

// 失败返回

PushServoP,Fail,40014,;

## 4.13.相对跟踪运动类控制指令

### 4.13.1.SetMoveTraceParams

4.13.1.1. 描述：设置相对跟踪运动控制参数。

✓ 发送消息：SetMoveTraceParams,nRbtID,nState,dDistance,dAwayVelocity,dBackVelocity,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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	相对跟踪运动的靠近探寻速度

✓ 发送成功返回消息：SetMoveTraceParams,OK,;

✓ 发送失败返回消息：SetMoveTraceParams,Fail,ErrorCode,;

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

4.13.1.2. 示例

// 设置相对跟踪控制参数并开启相对跟踪运动

SetMoveTraceParams,0,1,120,50,50,;

// 成功返回

SetMoveTraceParams,OK,;

// 失败返回

SetMoveTraceParams,Fail,20007,;

### 4.13.2.SetMoveTraceInitParams

4.13.2.1. 描述：设置相对跟踪运动初始化参数。

✓ 发送消息：SetMoveTraceInitParams,nRbtID,nState,dDistance,dVelocity,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：SetMoveTraceInitParams,OK,;

✓ 发送失败返回消息：SetMoveTraceInitParams,Fail,ErrorCode,;

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

4.13.2.2. 示例

// 设置相对跟踪初始化参数

SetMoveTraceInitParams,0,-14,135,130,65,;

// 成功返回

SetMoveTraceInitParams,OK,;

// 失败返回

SetMoveTraceInitParams,Fail,20007,;



### 4.13.3.SetMoveTraceUcs

4.13.3.1. 描述：设置相对跟踪运动的跟踪探寻方向。

✓ 发送消息：SetMoveTraceUcs,nRbtID,dX,dY,dZ,dRx,dRy,dRz,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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[°]

✓ 发送成功返回消息：SetMoveTraceUcs,OK,;

✓ 发送失败返回消息：SetMoveTraceUcs,Fail,ErrorCode,;

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

4.13.3.2. 示例

// 设置相对跟踪跟踪方向

SetMoveTraceUcs,0,420,0,445,180,0,180,;

// 成功返回

SetMoveTraceUcs,OK,;

// 失败返回

SetMoveTraceUcs,Fail,20007,;

## 4.13.4.SetTrackingState

4.13.4.1. 描述：设置传送带跟踪运动状态。

✓ 发送消息：SetTrackingState,nRbtID,nState,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned 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

✓ 发送成功返回消息：SetTrackingState,OK,;

✓ 发送失败返回消息：SetTrackingState,Fail,ErrorCode,;

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

4.13.4.2. 示例

// 开启传送带跟踪

SetTrackingState,0,1,;

// 成功返回

SetTrackingState,OK,;

// 失败返回

SetTrackingState,Fail,20007,;

## 4.14.其他指令

### 4.14.1.HRAppCmd

4.14.1.1.描述：执行插件 App 命令。

✓ 发送消息：HRAppCmd, HRAppName, HRAppCmdName,sParams1,...,sParamsn,;

参数	名称	数据类型	有效范围	内容
HRAppName	App 名称	string	-	插件 App 名称
HRAppCmdName	命令名称	string	-	插件命令名称
sParams	参数列表	string		参数列表

✓ 发送成功返回消息：HRAppCmd,OK,;

✓ 发送失败返回消息：HRAppCmd,Fail,ErrorCode,;

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

4.14.1.2. 示例

// 执行插件命令

HRAppCmd,0,;

// 成功返回

HRAppCmd,OK,;

// 失败返回

HRAppCmd,Fail,20007,;

## 4.14.2. WriteHoldingRegisters

4.14.2.1. 描述：写末端连接的 Modbus 从站寄存器。

✓ 发送消息：WriteHoldingRegisters,nRbtID,nSlaveID,nFunction,nRegAddr,nRegCount,vecData,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
nSlaveID	从站 ID	int	0/1	从站 ID
nFunction	功能码	int	>0	功能码： 0x01-读线圈寄存器 0x02-读线圈离散输入寄存器 0x03-读保持寄存器 0x04-读输入寄存器 0x05-写单个线圈寄存器 0x06-写单个保持寄存器 0x0f-写多个线圈寄存器 0x10-写多个保持寄存器
nRegAddr	寄存器地址	int	>0	寄存器起始地址
nRegCount	寄存器数量	int	1~11	寄存器数量,单次写寄存器最多支持连续 11 个连续寄存器
vecData	寄存器数据	int	-	寄存器数据，根据寄存器数量确定具体容器大小

\* 注：末端为 EtherCAT 总线版本 IO 时有效。

✓ 发送成功返回消息：WriteHoldingRegisters,OK,;

✓ 发送失败返回消息：WriteHoldingRegisters,Fail,ErrorCode,;

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

4.14.2.2. 示例

// 写寄存器地址 5000 的寄存器值为 88

WriteHoldingRegisters,0,1,16,5000,1,88,;

---

// 成功返回

WriteHoldingRegisters,OK,;

// 失败返回

WriteHoldingRegisters,Fail,40058,;

\* 注：连续写多个寄存器的地址必须是连续的，并且输入参数寄存器值的数量与参数寄存器数量必须一致。

// 写寄存器地址 5000 开始的连续 5 个寄存器值分别为 88,89,90,91,92

WriteHoldingRegisters,0,1,16,5000,5,88,89,90,91,92,;

// 成功返回

WriteHoldingRegisters,OK,;

// 失败返回

WriteHoldingRegisters,Fail,40058,;

\* 注：末端 Modbus 当前最多支持连续写 11 个寄存器

// 写寄存器地址 5000 开始的连续 11 个寄存器值为 88,89,90,91,92,93,94,95,96,97,98

WriteHoldingRegisters,0,1,16,5000,11,88,89,90,91,92,93,94,95,96,97,98,;

// 成功返回

WriteHoldingRegisters,OK,;

// 失败返回

WriteHoldingRegisters,Fail,40058,;

### 4.14.3.ReadHoldingRegisters

4.14.3.1. 描述：读末端连接的 Modbus 从站寄存器。

✓ 发送消息：ReadHoldingRegisters,nRbtID,nSlaveID,nFunction,nRegAddr,nRegCount,;

参数	名称	数据类型	有效范围	内容
nRbtID	机器人 id	unsigned int	0~5	机器 ID 号，默认值=0
nSlaveID	从站 ID	int	0/1	从站 ID
nFunction	功能码	int	>0	功能码： 0x01-读线圈寄存器 0x02-读线圈离散输入寄存器 0x03-读保持寄存器 0x04-读输入寄存器 0x05-写单个线圈寄存器 0x06-写单个保持寄存器 0x0f-写多个线圈寄存器 0x10-写多个保持寄存器
nRegAddr	寄存器地址	int	>0	寄存器起始地址
nRegCount	寄存器数量	int	1~12	寄存器数量, 单次最多允许读取连续 12 个寄存器值

\* 注：末端为 EtherCAT 总线版本 IO 时有效。

✓ 发送成功返回消息：ReadHoldingRegisters,OK,vecData1,...vecData12;

返回值	名称	数据类型	有效范围	内容
vecData1~ vecData12	寄存器数据	int	-	寄存器数据，根据实际读取的寄存器数量返回参数个数

✓ 发送失败返回消息：ReadHoldingRegisters,Fail,ErrorCode,;

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

4.14.3.2. 示例

// 读寄存器地址 5000 的寄存器

---

ReadHoldingRegisters,0,1,3,5000,1,;

// 成功返回

ReadHoldingRegisters,OK,88,;

// 失败返回

ReadHoldingRegisters,Fail,40058,;

// 读寄存器地址 5000 开始的连续 5 个寄存器值

ReadHoldingRegisters,0,1,3,5000,5,;

// 成功返回, 读取结果分别为 88,89,90,91,92

ReadHoldingRegisters,OK,88,89,90,91,92,;

// 失败返回

ReadHoldingRegisters,Fail,40058,;

// 读寄存器地址 5000 开始的连续 11 个寄存器

ReadHoldingRegisters,0,1,3,5000,11,;

// 成功返回, 读取结果分别为 88,89,90,91,92,93,94,95,96,97,98

ReadHoldingRegisters,OK,88,89,90,91,92,93,94,95,96,97,98,;

// 失败返回

ReadHoldingRegisters,Fail,40058,;

## 第五章 Datasheet 服务

### 5.1.内容

// DataSheet 是机器人数据, 里面包含了如位置, 状态等数据, 提供 Json 和 Struct 两种形式, Json 示例如下:

```
{
  "PosAndVel":{"Actual_Position":["0.000","0.000","0.000","0.000","0.000","0.000","0.000","0.000","0.000","0.000",
  "-0.000","0.000"],
  "Actual_PCS_TCP":["0.000","0.000","0.000","0.000","0.000","0.000"],
  "Actual_PCS_Base":["0.000","0.000","0.000","0.000","-0.000","0.000"],
  "Actual_Joint_Current":["-0.000","-0.000","-0.000","-0.000","-0.000","-0.000"],
  "Actual_Joint_Velocity":["0.000","0.000","0.000","0.000","0.000","0.000"],
  "Actual_Joint_Acceleration":["0.000","0.000","0.000","0.000","0.000","0.000"],
  "Actual_Override":"0.050"
},
  "EndIO":{"EndDI":[0,0,0,0],
    "EndDO":[0,0,0,0],
    "EndButton":[1,1,1,1],
    "EnableEndBTN":0,
    "EndAI":[0.00,0.00]
  },
  "ElectricBoxIO":{"BoxCI":[0,0,0,0,0,0,0,0],
    "BoxCO":[0,0,0,0,0,0,0,0],
    "BoxDI":[0,0,0,0,0,0,0,0],
    "BoxDO":[0,0,0,0,0,0,0,0],
    "Conveyor":"0.000",
    "Encode":0
  },
  "ElectricBoxAnalogIO":{"BoxAnalogOutMode_1":0,
    "BoxAnalogOutMode_2":0,
    "BoxAnalogOut_1":"0.00",
    "BoxAnalogOut_2":"0.00",
    "BoxAnalogIn_1":"-12.50",
    "BoxAnalogIn_2":"-12.50"
  },
  "StateAndError":{"robotState":33,
    "robotEnabled": 1,
    "robotPaused": 0,
```




```

    "robotMoving": 0,
    "robotBlendingDone": 1,
    "InPos": 0,
    "Error_AxisID":0,
    "Error_Code":0,
    "BrakeState": [0, 0, 0, 0, 0, 0],
    "nAxisStatus": [3, 3, 3, 3, 3, 3],
    "nAxisErrorCode": [0, 0, 0, 0, 0, 0],
    "nResetSafeSpace":[1],
    "nAxisGroupStatus":[0],
    "nAxisGroupErrorCode":[0],
  },
  "HardLoad":{"EtherCAT_TotalFrame":0,
    "EtherCAT_FramesPerSecond":0,
    "EtherCAT_TotalLostFrame":0,
    "EtherCAT_TxErrorFrame":0,
    "EtherCAT_RxErrorFrame":0,
    "Box48IN_Voltage":48.00,
    "Box48IN_Current":0.00,
    "Box48Out_Voltage":48.00,
    "Box48Out_Current":0.00,
    "Slave_temperature":[0.00,0.00,0.00],
    "Slave_Voltage":[0.00,0.00,0.00]
  },
  "FTData":{"FTControlState": 0,
    "FTData":[0.00,0.00,0.00,0.00,0.00,0.00],
    "FTSrcData":[0.00,0.00,0.00,0.00,0.00,0.00]
  },
  "Script":{"errorCode": "0",
    "cmdid":["", "", "", "", "", ""],
    "GlobalVar":[]},
  "pluginsdata": {
  }
}

```

数据包内容解释:

字段	解释
Actual_Position	关节位置，当前用户坐标和工具坐标下的迪卡尔坐标位置
Actual_PCS_TCP	当前工具坐标下的迪卡尔坐标位置

Actual_PCS_Base	基于基座坐标系下的迪卡尔坐标位置
Actual_PCS_Tool	Tool 坐标系下的迪卡尔坐标位置，新版本已不再提供
Actual_Joint_Current	当前实际关节运行时电流，单位[A]
Actual_Joint_Velocity	实际关节速度，单位[°/s]
Actual_Joint_Acceleration	实际关节加速度，单位[°/s <sup>2</sup> ]
Actual_Override	速度比，[0~1]
EndDI	末端数字输入
EndDO	末端数字输出
EndButton	末端按钮，按钮与下标对应如图 
EndAI	末端模拟输入
BoxAnalogOutMode_1	电箱模拟输出通道 0 的模式，1=电压，2=电流，0=关闭通道
BoxAnalogOutMode_2	电箱模拟输出通道 1 的模式，1=电压，2=电流，0=关闭通道
BoxAnalogOut_1	电箱模拟输出通道 0 的值
BoxAnalogOut_2	电箱模拟输出通道 1 的值
BoxAnalogIn_1	电箱模拟输入通道 0 的值
BoxAnalogIn_2	电箱模拟输入通道 1 的值
robotState	状态机，具体看状态机表
robotEnabled	使能状态，0 未使能；1 使能
robotPaused	暂停状态，0 未处于暂停状态；1 处于暂停状态
robotMoving	运动状态，0 未运动；1 运动中
robotBlendingDone	路点运动状态，0 路点运动中；1 运动运动完成
InPos	到位状态，0 未到位；1 运动到位
Error_AxisID	错误轴号，1~6
Error_Code	错误码
BrakeState	抱闸状态，0 抱闸；1 松闸
nAxisStatus	轴状态
nAxisErrorCode	每个轴的错误码
nResetSafeSpace	在超出安全空间时，是否运动回到安全空间
nAxisGroupStatus	轴组状态
nAxisGroupErrorCode	轴组错误码
EtherCAT_TotalFrame	与本体通信的总次数
EtherCAT_FramesPerSecond	与本体通信频率
EtherCAT_TotalLostFrame	与本体通信的错误总次数

EtherCAT_TxErrorFrame	接收本体信息错误次数
EtherCAT_RxErrorFrame	发送给本体信息错误次数
Box48IN_Voltage	48V 电源电压
Box48IN_Current	48V 电源电流
Box48OUT_Voltage	48V 供给本体电压
Box48OUT_Current	48V 供给本体电流
Slave_temperature	关节温度，共 3 个模组
Slave_Voltage	关节电压，共 3 个模组
FTControlState	力控控制状态，0 关闭力控，1 开启力控
FTData	力传感器标定后数据，X,Y,Z 单位[N],Rx,Ry,Rz 单位[NM]
FTSrcData	力传感器原始数据，X,Y,Z 单位[N],Rx,Ry,Rz 单位[NM]
Script	脚本运行信息
pluginsdata	插件运行信息

## 5.2.包头

Datasheet 数据带如下包头：

4BYTE	总长（包头+数据） 4BYTE	数据长度 4BYTE	Data(json)
RBTL			

## 第六章 控制机器人运动流程—示例

### 6.1.运动流程图

控制机器人运动，在机器人成功使能的情况下，可以通过发送 Move 系列指令以及长短点动指令的方式让机器人运动到目标位置。发送运动指令的过程中有以下几点需要注意：

1. 只有当确保机器人当前运动指令成功完成的时候，才能下发下一次运动指令（通过发送获取机器人运动状态指令 ReadRobotState 来获得当前机器人运动状态）。
2. 当调用获取机器人运动状态指令返回运动状态为运动中时，上位机应该循环调用获取机器人运动状态指令，直到机器人运动状态标识运动完成或运动出错。
3. 当控制系统返回机器人运动报错时，上位机应该调用 GrpReset 指令对控制系统进行清错，只有当清错成功时，才能继续调用下一个运动指令。
4. 如果控制系统返回机器人运动完成，那么可以继续下发下一个运动指令，然后重复 2-4。

具体流程如下图所示：

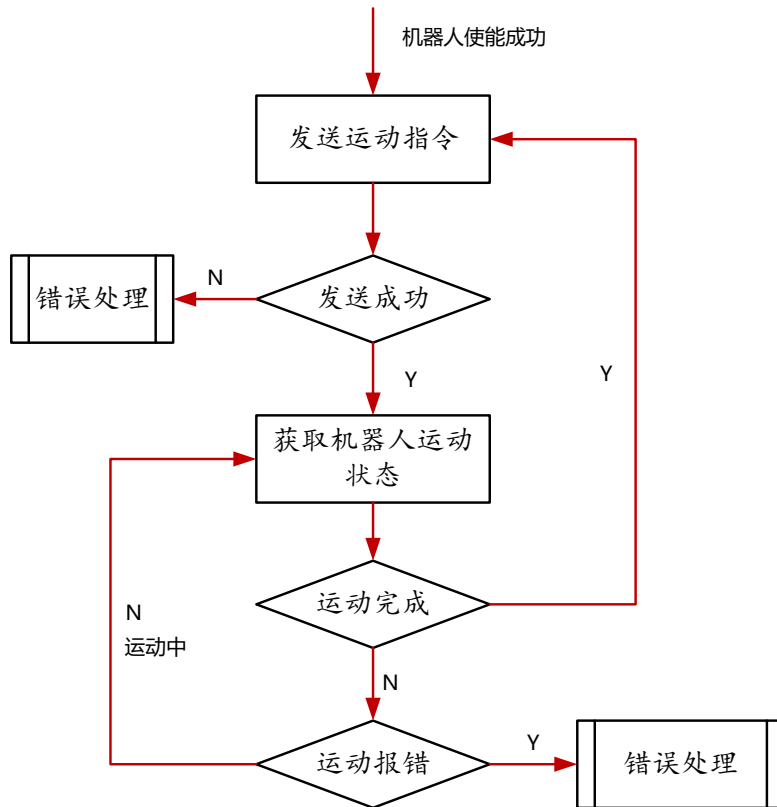


图 6-1 机器人运动流程图

## 6.2.示例代码

伪代码如下，仅供参考：

// 等待运动完成

```

int WaitMotionFinish()
{
    while(1)
    {
        // 1.发送获取机器人运动状态消息:ReadRobotState,;
        // 2.解析返回的运动状态:MoveState

        if(movingState == 1)
        {
            // 处于运动中, 等待10 毫秒, 再次查询
            Sleep(10);
        }
        else if(movingState == 0)
        {

```

---

```
// 运动完成, 直接返回0  
return 0;  
}  
else  
{  
    // 其他错误情况, 分别处理  
    return nErrCode;  
}  
}  
return 0;  
}
```

---

## 第七章 附录

### 7.1.参考文件

1. 错误码及状态机描述请参阅 [《HansRobot\\_ErrorCode.docx》](#)。
2. 大族机器人 C++ SDK 使用请参阅 [《HanRobot Library C++》](#)。
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》](#)。