English version of the README -> please click here

越疆 TCP/IP

控制协议文档

文档版本：V4.0

发布日期：2023-06-20

深圳市越疆科技股份有限公司

免责申明

在法律允许的最大范围内，本手册所描述的产品（含其硬件、软件、固件等）均“按照现状”提供，可能存在瑕疵、错误或故障，越疆不提供任何形式的明示或默示保证，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证；亦不对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害进行赔偿。

在使用本产品前详细阅读本使用手册及网上发布的相关技术文档并了解相关信息，确保在充分了解机械臂及其相关知识的前提下使用机械臂。越疆建议您在专业人员的指导下使用本手册。该手册所包含的所有安全方面的信息都不得视为Dobot的保证，即便遵循本手册及相关说明，使用过程中造成的危害或损失依然有可能发生。

本产品的使用者有责任确保遵循相关国家的切实可行的法律法规，确保在越疆机械臂的使用中不存在任何重大危险。

深圳市越疆科技股份有限公司

地址：深圳市南山区留仙大道3370号南山智园崇文区2号楼1003

网址：http://www.dobot.cn/

0.Changelog

- V4.5.0-2023/08/23：增加命令ServoJ、ServoP
- V4.3.0-2023/06/08：增加实时数据CurrentCommandId,VibrationDisZ
- V4.2.9-2023/05/05：删除StopRobot命令
- V4.2.8-2023/04/10：新增GetInputBool、GetInputInt、GetInputFloat、GetOutputBool、GetOutputInt、GetOutputFloat、SetOutputBool、SetOutputInt、SetOutputFloat命令

- V4.2.7-2023/03/29：
  1）用户/工具坐标系相关参数类型自定义模式去除
  对应指令：User、Tool、MovJIO、MovLIO、RelJointMovJ、RelMovLUser、RelMovLTool、RelMovJUser、RelMovJTool、Circle、Arc、MovL、MovJ、StartPath、PositiveKin、InverseKin
  2）坐标系索引范围对应更新到0~50
- V4.2.6-2023/03/21：运动相关指令可选参数vt参数名称更改为speed
- V4.2.5-2023/03/20：更新AO()、AOInstant()指令valuse值参数范围
- V4.2.4-2023/03/16：更新TCP实时反馈数据并注明去除或不实现的字段
- V4.2.3-2023/03/18：TCP新增Pause，Continue，Stop、SetBackDistance、SetPostCollisionMode。删除PauseScript，ContinueScript，StopScript
- V4.2.1-2023/03/14：TCP新增修改用户坐标系、修改工具坐标系、计算用户坐标系、计算 工具坐标系指令:SetUser()、SetTool()、CalcUser()、CalcTool()
- V4.2.0-2023/03/7：TCP新增LUA已有指令：SetToolPower()、SetToolMode()
- V4.1.9-2023/02/28：TCP新增指令文档补充：GetDO()、GetAO ()、GetToolDO ()、GetDOGroup ()、SetTool485()、SetSafeWallEnable ()
- V4.1.8-2023/02/21：运动相关指令新增可选参数vt
- V4.1.7-2023/02/15：调整GetPose()指令参数个数限定，仅允许0/2个参数
- V4.1.6-2023/01/18：新增关于中文字符发送的注意事项：需将发送端的编码方式选择为UTF-8格式，否则会导致中文字符接收异常
- V4.1.5-2023/01/04：EnableRobot()新增负载检查可选参数，SetPayload()新增可根据负载名称进行设置参数类型
- V4.1.4-2022/12/22：TCP实时反馈数据中新增：FA、J4、J5、J6接近暂停状态
- V4.1.3-2022/12/14：RobotMode()新增检测到碰撞状态:ROBOT_MODE_COLLISION；实时反馈数据中新增：检测到碰撞状态RobotCollision；ClearError()指令可清除碰撞状态
- V4.1.2-2022/12/13：安全皮肤的相关指令新增：EnableSafeSkin(status) 、SetSafeSkin(part, status)
- V4.1.1-2022/11/25：轨迹复现指令名称修正：StartPath()
- V4.1.0-2022/11/23: RobotMode()新增初始化状态：ROBOT_MODE_INIT，控制柜启动过程中状态返回ROBOT_MODE_INIT，启动完成且上电成功返回 ROBOT_MODE_DISABLED
- V4.0.9-2022/11/9: 去除MovJ、MovJIO、RelMovJTool、RelMovJUser指令的可选参数r，关节运动不支持r参数的设置。
- V4.0.8-2022/10/24:
  1）DO指令新增固定时间后取反功能。
  2）新增开/关安全皮肤指令 ： SetSafeSkin()
  3）新增进入/退出推拽，设置推拽灵敏度指令：StartDrag()、StopDrag()、DragSensivity()
- V4.0.7-2022/10/08: 补充说明带名称参数格式/范围错误的错误码返回
- V4.0.6-2022/09/22: 补充说明运动指令的坐标系使用规则
- V4.0.5-2022/09/21: 对应错误状态新增可执行指令，处理规则：

1）错误状态可执行指令：错误清除、错误查询、急停、机器人状态；

2）急停状态可执行指令：错误清除、错误查询、急停松开、机器人状态

3）下电状态可执行指令：错误清除、错误查询、上电、机器人状态、急停松开

- V4.0.4-2022/09/19：新增错误码类型，并对所有指令添加错误状态防护，包括：当前为错误状态，当前为下电状态，当前为急停拍下状态，需先解除这些状态再进行指令的实际处理。
- V4.0.3-2022/09/13:

1）增加查找算法队列指令： GetCurrentCommandID();

2）算法指令添加对应算法id返回值，包括：DO()、ToolDO()、AO()、MovJ()、MovL()、MovJIO()、Circle()、RelMovJTool()、RelMovLTool()、 RelMovLUser()、RelJointMovJ()；

- V4.0.2-2022/09/2: 新增错误码类型列表；
- V4.0.1-2022/08/23:

1）TCP对应指令名称更新，包括：DOInstant()、ToolDOInstant()、AOInstant()、VelJ()、VelL()、GetStartPose()、StartTrace()、 PositiveKin()、StopRobot()、SetPayload()；

2）TCP部分指令去除，包括：ResetRobot()、HandleTrajPoints()、GetSixForceData()、StartDrag()、StopDrag()、 SetCollideDrag()、SetTerminalKeys()、SetTerminal485()、Arch()、LoadSwitch()、SetArmOrientation()、ServoJ()、ServoP()、StartPath()、StartFCTrace()、Sync()、JointMovJ()、SetSafeSkin();

3） TCP部分指令参数格式更新,包括：User()、Tool()、PositiveKin()、InverseKin()、MovJ()、MovL()、MovLIO()、MovJIO()、RelMovJTool()、RelMovJUser()、RelMovLUser()、RelJointMovJ()、Arc()、Circle()；

4）RobotMode()，机器人状态对应含义更新；

1. 综述

越疆协作机器人支持TCP/IP二次开发方式：需先在上位机将机器人控制方式设置为"**TCP/IP二次开发模式**"，后连接机器人TCP服务端进行指令发送及对应数据监

控，**TCP/IP控制模式**；具体控制方式详见《Dobot Pro软件使用说明->设置->远程控制章节中；

描述机器人有几种控制方式，此方式针对**远程控制机器人**；由于基于TCP/IP的通讯具有成本低、可靠性高、实用性强、性能高等特点；许多工业自动化项目对支持TCP/IP协议控制机器人需求广泛，因此CR机器人将设计在TCP/IP协议的基础上，提供了丰富的接口用于与外部设备的交互；

根据设计，CR机器人会开启**29999**、**30004**、**30005**以及**30006**服务器端口；

29999服务器端口(以下简称Dashboard端口)通过一发一收的方式负责接收一些设置以及运动控制相关的指令，即**控制端口接收到客户端约定消息格式后会将结**

果反馈客户端；

30004服务器端口(以下简称实时反馈端口)**每8ms反馈机器人的信息；**

30005服务器端口**每200ms反馈机器人的信息**，

30006端口为**可配置**的反馈机器人信息端口(默认为每**50ms**反馈)；

2.消息格式

TCPIP远程控制命令**不区分大小写格式**；如ENABLEROBOT()/enablerobot()/eNabLErobOt()，控制器都会按照**使能**的命令执行；

消息命令与消息应答都是 ASCII 码格式(字符串形式)。

2.1 命令格式

`命令名称(Param1,Param2,Param3,……Paramn)`

命令格式如上所示，由一个命令名称，括号内由参数组成， 每一个参数之间以英文逗号 ”,” 相隔，一个完整的命令以右括号结束。

命令区分为队列指令和立即指令，队列指令和立即指令的区别，详见指令列表中的指令类型

- 队列指令：指令下发后不可立即执行，需发送给算法队列，由算法规划执行执行
- 立即指令：指令下发后立即执行

  注：队列/立即指令详见下表指令类型

2.2 返回格式

2.2.1 返回:

返回命令格式： "指令检查值,{命令返回值},原始命令;"

`"ErrorID,{value,...,valuen},命令名称(Param1,Param2,Param3……Paramn);"`

命令格式如上所示:

- ErrorID为0时表示命令接收成功；返回非0则代表命令有错误，具体的错误描述见第五章节;
- {value1,value2,value3,...,valuen}表示返回值，没有返回值则返回{}；
  - 队列指令value返回值指队列ID:commandID
  - 立即指令valude返回值为要返回的数据内容
- 命令名称(Param1,Param2,Param3……Paramn)指下发的指令内容。

例:

MovL(pose={-500,100,200,150,0,90})

- 返回：0,{1},MovL(pose={-500,100,200,150,0,90}); //0表示接收成功 {1}：commandID

  Mov(-500,100,200,150,0,90) //下发不存在的指令

- 报警： -10000,{},Mov(-500,100,200,150,0,90); //-10000表示命令错误

  GetPose(user = 1, tool = 0)

- 返回：0,{-473.0,-141.0,469.0,-180.0,0.0,90.0},GetPose(user = 1, tool = 0);

3.控制端口命令

上位机可以通过29999端口发送命令给机器人，这些命令被称为控制命令。

控制端口命令列表如下:

指令	描述	支持产品	指令类型

指令	描述	支持产品	指令类型
EnableRobot	使能机器人	CR/Nova/ED6	立即指令
DisableRobot	下使能机器人	CR/Nova/ED6	立即指令
ClearError	复位，用于清除错误	CR/Nova/ED6	立即指令
SpeedFactor	设置全局速率比	CR/Nova/ED6	立即指令
User	选择已标定的用户坐标系 （笛卡尔空间显示值 实际生效根据点）	CR/Nova/ED6	队列指令
Tool	选择已标定的工具坐标系	CR/Nova/ED6	队列指令
RobotMode	机器人模式	CR/Nova/ED6	立即指令
SetPayload	设置负载	CR/Nova/ED6	队列指令
DO	设置数字量输出端口状态	CR/Nova/ED6	队列指令

指令	描述	支持产品	指令类型
DOInstant	设置数字量输出端口状态	CR/Nova/ED6	立即指令
ToolDO	设置末端数字量输出端口状态	CR/Nova/ED6	队列指令
ToolDOInstant	设置末端数字量输出端口状态	CR/Nova/ED6	立即指令
AO	设置模拟量输出端口状态	CR/Nova/ED6	队列指令
AOInstant	设置模拟量输出端口状态	CR/Nova/ED6	立即指令
AccJ	设置关节加速度比例。	CR/Nova/ED6	立即指令
AccL	设置笛卡尔加速度比例。	CR/Nova/ED6	立即指令
VelJ	设置关节速度比例。	CR/Nova/ED6	立即指令
VelL	设置笛卡尔速度比例。	CR/Nova/ED6	立即指令

指令	描述	支持产品	指令类型
CP	运动时设置平滑过渡	CR/Nova/ED6	立即指令
PowerOn	机器人上电	CR/Nova/ED6	立即指令
RunScript	运行脚本	CR/Nova/ED6	立即指令
Stop	停止	CR/Nova/ED6	立即指令
Pause	暂停	CR/Nova/ED6	立即指令
Continue	继续	CR/Nova/ED6	立即指令
EnableSafeSkin	设置安全皮肤开关状态	CR/Nova/ED6	队列指令
SetSafeSkin	设置安全皮肤灵敏度	CR/Nova/ED6	队列指令
GetStartPose	获取轨迹的第一个点位	CR/Nova/ED6	立即指令

指令	描述	支持产品	指令类型
StartPath	轨迹复现	CR/Nova/ED6	队列指令
PositiveKin	正解	CR/Nova/ED6	立即指令
InverseSolution	逆解	CR/Nova/ED6	立即指令
SetCollisionLevel	设置碰撞等级	CR/Nova/ED6	队列指令
GetAngle	获取关节坐标系下机械臂的实时位姿	CR/Nova/ED6	立即指令
GetPose	获取笛卡尔坐标系下机械臂的实时位姿	CR/Nova/ED6	立即指令
EmergencyStop	急停	CR/Nova/ED6	立即指令
ModbusRTUCreate	创建ModbusRTU主站，并和从站建立连接	CR/Nova/ED6	立即指令
ModbusCreate	创建Modbus主站，并和从站建立连接	CR/Nova/ED6	立即指令

指令	描述	支持产品	指令类型
ModbusClose	和Modbus从站断开连接	CR/Nova/ED6	立即指令
GetInBits	读离散输入功能	CR/Nova/ED6	立即指令
GetInRegs	读输入寄存器	CR/Nova/ED6	立即指令
GetCoils	读线圈功能	CR/Nova/ED6	立即指令
SetCoils	写线圈功能	CR/Nova/ED6	立即指令
GetHoldRegs	读保存寄存器	CR/Nova/ED6	立即指令
SetHoldRegs	写保存寄存器	CR/Nova/ED6	立即指令
GetErrorID	获取错误ID	CR/Nova/ED6	立即指令
DI	获取数字量输入端口状态	CR/Nova/ED6	立即指令

指令	描述	支持产品	指令类型
ToolDI	获取末端数字量输入端口状态	CR/Nova/ED6	立即指令
AI	获取模拟量输入端口电压值	CR/Nova/ED6	立即指令
ToolAI	获取末端模拟量输入端口电压值	CR/Nova/ED6	立即指令
DIGroup	获取输入组端口状态	CR/Nova/ED6	立即指令
DOGroup	设置数字输出组端口状态	CR/Nova/ED6	队列指令
BrakeControl	抱闸控制	CR/Nova/ED6	立即指令
StartDrag	进入拖拽	CR/Nova/ED6	立即指令
StopDrag	退出拖拽	CR/Nova/ED6	立即指令
DragSensivity	拖拽灵敏度设置	CR/Nova/ED6	立即指令

指令	描述	支持产品	指令类型
GetDO	获取数字输出端口状态	CR/Nova/ED6	立即指令
GetAO	获取模拟量输出端口状态	CR/Nova/ED6	立即指令
GetDOGroup	获取组数字输出端口状态	CR/Nova/ED6	立即指令
SetTool485	设置末端485通讯参数	CR/Nova/ED6	立即指令
SetSafeWallEnable	设置安全墙开关	CR/Nova/ED6	队列指令
SetToolPower	设置末端电源开关	CR/Nova/ED6	立即指令
SetToolMode	设置末端模式	CR/Nova/ED6	立即指令
SetBackDistance	设置碰撞回退距离	CR/Nova/ED6	队列指令
SetPostCollisionMode	设置选择碰撞后是进入状态	CR/Nova/ED6	队列指令

指令	描述	支持产品	指令类型
SetUser()	设置用户坐标系	CR/Nova/ED6	立即指令
SetTool()	设置工具坐标系	CR/Nova/ED6	立即指令
CalcUser()	计算用户坐标系	CR/Nova/ED6	立即指令
CalcTool()	计算工具坐标系	CR/Nova/ED6	立即指令
GetInputBool	获取输入寄存器bool数值	CR/Nova/ED6	立即指令
GetInputInt	获取输入寄存器int数值	CR/Nova/ED6	立即指令
GetInputFloat	获取输入寄存器float数值	CR/Nova/ED6	立即指令
GetOutputBool	获取输出寄存器bool数值	CR/Nova/ED6	立即指令
GetOutputInt	获取输出寄存器int数值	CR/Nova/ED6	立即指令

指令	描述	支持产品	指令类型
GetOutputFloat	获取输出寄存器float数值	CR/Nova/ED6	立即指令
SetOutputBool	设置输出寄存器bool数值	CR/Nova/ED6	立即指令
SetOutputInt	设置输出寄存器int数值	CR/Nova/ED6	立即指令
SetOutputFloat	设置输出寄存器float数值	CR/Nova/ED6	立即指令
MovJ	点到点运动，目标点位为笛卡尔点位	CR/Nova/ED6	队列指令
MovL	直线运动，目标点位为笛卡尔点位	CR/Nova/ED6	队列指令
MovLIO	直线运动过程中并行设置数字输出端口的状态，可设置多组	CR/Nova/ED6	队列指令
MovJIO	点到点运动过程中并行设置数字输出端口的状态，可设置多组	CR/Nova/ED6	队列指令
Arc	圆弧运动。需结合其他运动指令完成圆弧运动	CR/Nova/ED6	队列指令

指令	描述	支持产品	指令类型
Circle	整圆运动	CR/Nova/ED6	队列指令
MoveJog	关节点动	CR/Nova/ED6	队列指令
StartPath	轨迹复现	CR/Nova/ED6	队列指令
RelMovJTool	沿工具坐标系进行相对运动，末端运动方式为关节运动	CR/Nova/ED6	队列指令
RelMovLTool	沿工具坐标系进行相对运动指令，末端运动方式为直线运动	CR/Nova/ED6	队列指令
RelMovJUser	沿用户坐标系进行相对运动指令，末端运动方式为关节运动	CR/Nova/ED6	队列指令
RelMovLUser	沿用户坐标系进行相对运动指令，末端运动方式为直线运动	CR/Nova/ED6	队列指令
RelJointMovJ	沿各轴关节坐标系进行相对运动指令，末端运动方式为关节运动	CR/Nova/ED6	队列指令
GetCurrentCommandId	获取当前运行指令的队列ID	CR/Nova/ED6	立即指令

3.1 EnableRobot

- 功能：使能机器人
- 格式：

  EnableRobot()
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值
load	double	负载重量kg	否	上一次全局指令设置值
centerX	double	X方向偏心距离mm	否	上一次全局指令设置值
centerY	double	Y方向偏心距离mm	否	上一次全局指令设置值
centerZ	double	Z方向偏心距离mm	否	上一次全局指令设置值
isCheck	int	是否进行负载检查：1：检查，0：不检查	否	0

- 返回：

  ErrorID,{},EnableRobot();
- 示例：

  EnableRobot() //使能

  EnableRobot(2) //使能，设置负载值为2，偏心值为之前设置的值

  EnableRobot(2,1,2,3)//使能，设置负载值为2，偏心值为1，2，3

  EnableRobot(2,1,2,3,1)//使能，设置负载值为2，偏心值为1，2，3，进行负载检查
- 说明：**可选参数数量：0/1/4/5**

机型	load范围(KG)	centerX范围(mm)	centerY范围(mm)	centerZ范围(mm)	isCheck范围
CR3/CR3A	[0,3]	[-999,999]	[-999,999]	[-999,999]	0/1
CR5/CR5A	[0,5]	[-999,999]	[-999,999]	[-999,999]	0/1
CR7/CR7A	[0,7]	[-999,999]	[-999,999]	[-999,999]	0/1
CR10/CR10A	[0,10]	[-999,999]	[-999,999]	[-999,999]	0/1
CR12/CR12A	[0,12]	[-999,999]	[-999,999]	[-999,999]	0/1
CR16/CR16A	[0,16]	[-999,999]	[-999,999]	[-999,999]	0/1
CR20A	[0,20]	[-999,999]	[-999,999]	[-999,999]	0/1
Nova2	[0,2]	[-999,999]	[-999,999]	[-999,999]	0/1

机型	load范围(KG)	centerX范围(mm)	centerY范围(mm)	centerZ范围(mm)	isCheck范围
Nova5	[0,5]	[-999,999]	[-999,999]	[-999,999]	0/1
Magician E6	[0,0.75]	[-999,999]	[-999,999]	[-999,999]	0/1

3.2 DisableRobot

- 功能：下使能机器人
- 格式：

  DisableRobot()
- 支持端口：29999
- 返回：

  ErrorID,{},DisableRobot();
- 示例：

  DisableRobot()//下使能

3.3 ClearError

- 功能：清错机器人
- 格式：

  ClearError()
- 支持端口：29999
- 返回：

  ErrorID,{},ClearError();
- 示例：

  ClearError()//机器人清错
- 说明：清除报警后，用户可以根据RobotMode来判断机器人是否还处于报警状态；对于清除不掉的报警需要重启控制柜解决；(详见GetErrorID说明)

3.5 SpeedFactor

- 功能：设置全局速度比例。
- 格式：

  SpeedFactor(ratio)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围

参数名	类型	含义	是否必填	取值范围
ratio	int	运动速度比例	是	[1,100]

- 返回：

  ErrorID,{},SpeedFactor(ratio);

- 示例：

  SpeedFactor(80)//设置全局速度比例为80%

3.6 User

- 功能：选择已标定的用户坐标系
- 格式：

  User(index)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
index	int	选择已标定的用户坐标系	是	[0,50]

- 返回：

  ErrorID,{},User(index);

- 示例：

  User(1)//设置全局用户坐标系为1

3.7 Tool

- 功能：选择已标定的工具坐标系
- 格式：

  Tool(index)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
index	int	选择已标定的工具坐标系	是	[0,50]

- 返回：

  ErrorID,{},Tool(index);

- 示例：

  Tool(1)//设置全局工具坐标系为1

3.8 RobotMode

- 功能：机器人状态。
- 格式

  RobotMode()
- 支持端口：29999
- 返回值：

模式	描述	备注
1	ROBOT_MODE_INIT	初始化状态
2	ROBOT_MODE_BRAKE_OPEN	抱闸松开
3	ROBOT_MODE_POWEROFF	本体下电状态
4	ROBOT_MODE_DISABLED	未使能(抱闸未松开)
5	ROBOT_MODE_ENABLE	使能(空闲)
6	ROBOT_MODE_BACKDRIVE	拖拽
7	ROBOT_MODE_RUNNING	运行状态（含脚本和TCP队列运行）
8	ROBOT_MODE_SINGLE_MOVE	单次运动状态(点动)
9	ROBOT_MODE_ERROR	错误状态
10	ROBOT_MODE_PAUSE	暂停状态
11	ROBOT_MODE_COLLISION	碰撞状态

- 返回：

  ErrorID,{Value},RobotMode();　//Value为返回模式值
- 示例：

  RobotMode()//获取当前机器人状态

  注：

  开启抱闸，状态为2；

  本体下使能，状态为4；

  使能成功后，则状态为5；

  机器人进入拖拽模式(使能状态)，状态为6；

  机器人运行，状态为7；运行状态包括：轨迹复现/拟合中、机器人运动(movj())以及脚本运行；

  机器人在点动，状态为8；

  机器人暂停，状态为10；

  其中报警优先级最高，其他状态同时存在时，若有报警，先将状态置

3.9 SetPayload

- 功能：设置当前的负载
- 支持端口：29999
- 格式（1）：

  SetPayload(load,x,y,z)
- 参数详解：

参数名	类型	含义	是否必填	默认值
load	double	负载重量 kg	是	
x	double	偏心坐标X mm	否	上一次全局指令设置值
y	double	偏心坐标Y mm	否	上一次全局指令设置值
z	double	偏心坐标Z mm	否	上一次全局指令设置值

- 返回：

  ErrorID,{},SetPayload(load,x,y,z);
- 格式（2）：

  SetPayload(name)
- 参数详解：

参数名	类型	含义	是否必填
name	string	负载名称	是

- 返回：

  ErrorID,{},SetPayload(name);
- 示例：

  SetPayload(3) //设置负载值为3，偏心距离为默认值

  SetPayload(3,1,2,3) //设置负载值为3，偏心距离为1，2，3

  SetPayload("Load1"); //设置负载为"Load1"

3.10 DO（队列指令）

- 功能：设置数字输出端口状态
- 格式：

  DO(index,status,time)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	取值范围

参数名	类型	含义	是否必填	默认值	取值范围
index	int	数字输出索引，取值范围：1~16或100~1000	是		[1,16] [100,1000]
status	int	数字输出端口状态，1：打开；0：关闭	是		0/1
time	int	持续输出时间，单位ms	否	0	[25,60000]

- 返回：ResultID:算法队列ID

  ErrorID,{ResultID},DO(index,status);

- 示例：

  DO(1,1,2000)//设置DO1为打开状态，2000ms后关闭

- 说明：使用取值范围100-1000需要有拓展IO模块的硬件支持；

3.11 DOInstant

- 功能：设置数字输出端口状态
- 格式：

  DOInstant(index,status)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
index	int	数字输出索引	是	[1,16] [100,1000]
status	int	数字输出端口状态，1：高电平；0：低电平	是	0/1

- 返回：

  ErrorID,{},DOInstant(index,status);

- 示例：

  DOInstant(1,1)//设置DO1为打开状态

- 说明：使用取值范围100-1000需要有拓展IO模块的硬件支持；

3.12 ToolDO（队列指令）

- 功能：设置末端数字输出端口状态
- 格式：

  ToolDO(index,status)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
index	int	数字输出索引	是	[1,2]
status	int	数字输出端口状态，1：打开；0：关闭	是	0/1

- 返回：ResultID:算法队列ID

  ErrorID,{ResultID},ToolDO(index,status);

- 示例：

  ToolDO(1,1) //设置末端DO1为打开状态

3.13 ToolDOInstant

- 功能：设置末端数字输出端口状态
- 格式：

  ToolDOInstant(index,status)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
index	int	数字输出索引	是	[1,2]
status	int	数字输出端口状态，1：打开；0：关闭	是	0/1

- 返回：

  ErrorID,{},ToolDOInstant(index,status);

- 示例：

  ToolDOInstant(1,1)//设置末端DO1为打开状态

3.14 AO（队列指令）

- 功能：设置控制柜模拟输出端口的模拟值
- 格式：

  AO(index,value)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
index	int	模拟输出索引	是	1/2

参数名	类型	含义	是否必填	取值范围
value	double	对应index的模拟值	是	电压取值范围0~10V，电流取值范围为4~20

- 返回：ResultID:算法队列ID

  ErrorID,{ResultID},AO(index,value);

- 示例：

  AO(1,2)//设置模拟输出端口1的模拟值为2

3.15 AOInstant

- 功能：设置控制柜模拟输出端口的模拟值
- 格式：

  AOInstant(index,value)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
index	int	模拟输出索引	是	1/2
value	double	对应index的模拟值	是	电压取值范围0~10V，电流取值范围为4~20

- 返回：

  ErrorID,{},AOInstant(index,value);

- 示例：

  AOInstant(1,2)//设置模拟输出端口1的模拟值为2

3.16 AccJ

- 功能：设置关节加速度比例。
- 格式：

  AccJ(R)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
R	int	关节加速度百分比	是	[1,100]

- 返回：

ErrorID,{},AccJ(R);

- 示例：

  AccJ(50) //设置关节加速度比例为50%

3.17 AccL

- 功能：设置笛卡尔加速度比例。
- 格式：

  AccL(R)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
R	int	笛卡尔加速度比例	是	[1,100]

- 返回：

  ErrorID,{},AccL(R);

- 示例：

  AccL(50)//设置笛卡尔加速度比例为50%

3.18 VelJ

- 功能：设置关节速度比例。
- 格式：

  VelJ(R)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
R	int	关节速度比例	是	[1,100]

- 返回：

  ErrorID,{},VelJ(R);

- 示例：

  VelJ(50)//设置关节速度比例为50%

3.19 VelL

- 功能：设置笛卡尔速度比例。
- 格式：

  VelL(R)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
R	int	笛卡尔速度比例	是	[1,100]

- 返回：

  ErrorID,{},VelL(R);

- 示例：

  VelL(50)//设置笛卡尔速度比例为50%

3.20 CP

- 功能：设置CP比例。CP即平滑过渡，机械臂从起始点经过中间点到达终点时，经过中间点是以直角方式过渡还是以曲线方式过渡。
- 格式：

  CP(R)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
R	int	平滑过渡比例	是	[0,100]

- 返回：

  ErrorID,{},CP(R);

- 示例：

  CP(50)//设置平滑过渡比例为50%

3.21 PowerOn

- 功能：机器人上电。
- 格式：

  PowerOn()

- 支持端口：29999
- 返回：

  ErrorID,{},PowerOn();

- 示例：

  PowerOn()//上电

- 说明：机器人上电到完成，需要等待大概10秒钟的时间接口才返回

3.22 RunScript

- 功能：运行脚本。
- 格式：

  RunScript(projectName)
- 支持端口：29999
- 参数详解：

参数名	类型	含义
projectName	string	脚本名称

- 返回：

  ErrorID,{},RunScript(projectName);
- 示例：

  RunScript("123")

  注：需将发送端的编码方式选择为UTF-8格式，否则会导致中文字符接收异常

  若为纯数字工程名称，则需要添加""，参数格式为字符串类型

3.23 Stop

- 功能：停止运动(或脚本)。
- 格式：

  Stop()
- 支持端口：29999
- 返回：

  ErrorID,{},Stop();
- 示例：

  Stop()

  注：可以停止点动、脚本运行、关节运动等一系列运动

3.24 Pause

- 功能：暂停运动(或脚本)。
- 格式：

  Pause()
- 支持端口：29999
- 返回：

  ErrorID,{},Pause();
- 示例：

  Pause()

注：可以暂停movj()等一系列运动，使机器人处于暂停状态，点动不可暂停

3.25 Continue

- 功能：继续运动(或脚本)。
- 格式：

  Continue()
- 支持端口：29999
- 返回：

  ErrorID,{},Continue();
- 示例：

  Continue()

  注：继续运动，暂停状态下的算法队列指令可继续运动，机器人处于运行状态

3.26 PositiveKin

- 功能：正解。（给定机器人各关节的角度，计算出机器人末端的空间位置）
- 格式：

  PositiveKin(J1,J2,J3,J4,J5,J6,user=1,tool=1)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	取值范围
J1	double	J1 轴位置，单位：度	是		任意值
J2	double	J2 轴位置，单位：度	是		任意值
J3	double	J3 轴位置，单位：度	是		任意值
J4	double	J4 轴位置，单位：度	是		任意值
J5	double	J5 轴位置，单位：度	是		任意值
J6	double	J6 轴位置，单位：度	是		任意值
user=1	string	选择已标定的用户坐标系	否	上一次全局指令设置值	[0,50]
tool=1	string	选择已标定的工具坐标系	否	上一次全局指令设置值	[0,50]

- 返回：

  ErrorID,{x,y,z,a,b,c},PositiveKin(J1,J2,J3,J4,J5,J6,user=1,tool=1);

  {x,y,z,a,b,c}指返回的空间位置

- 示例：

  PositiveKin(0,0,-90,0,90,0,user=1,tool=1）//下发关节角度返回当前的机器人末端的空间位置

  返回：

  0,
  {473.000000,-141.000000,469.000000,-180.000000,-0.000000,-90.000000},PositiveKin(0,0,-90,0,90,0,,user=1,tool=1);

3.27 InverseKin

- 功能：逆解。（已知机器人末端的位置和姿态，计算机器人各关节的角度值）
- 格式：

  InverseKin(X,Y,Z,Rx,Ry,Rz,User,Tool,useJointNear,JointNear)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	取值范围
X	double	J1 轴位置，单位：度	是		任意值
Y	double	J2 轴位置，单位：度	是		任意值
Z	double	J3 轴位置，单位：度	是		任意值
Rx	double	J4 轴位置，单位：度	是		任意值
Ry	double	J5 轴位置，单位：度	是		任意值
Rz	double	J6 轴位置，单位：度	是		任意值
useJointNear=1	string	是否角度选解 1:jointNear数据有效; 0:jointNear数据无效	否	0	0/1
jointNear={j1,j2,j3,j4,j5,j6}	string	选解六个关节角度值	否	{0,0,0,0,0,0}	任意值
user=1	string	选择已标定的用户坐标系	否	上一次全局指令设置值	[0,50]
tool=1	string	选择已标定的工具坐标系	否	上一次全局指令设置值	[0,50]

- 返回值：

  ErrorID,{J1,J2,J3,J4,J5,J6},InverseKin(X,Y,Z,RX,RY,RZ,useJointNear =1,jointNear={J1,J2,J3,J4,J5,J6},user=1,tool=1);

  {J1,J2,J3,J4,J5,J6}:逆解的关节值

- 示例：

  InverseKin(473.000000,-141.000000,469.000000,-180.000000,0.000,-90.000,useJointNear =1,jointNear={0,0,-90,0,90,0})

  返回：

  0,{0,0,-90,0,90,0},InverseKin(473.000000,-141.000000,469.000000,-180.000000,0.000,-90.000,useJointNear =1,jointNear={0,0,-90,0,90,0})

  注：若jointNear参数未带，仅带useJointNear参数，则useJointNear参数无效

3.28 SetCollisionLevel（队列指令）

- 功能：设置碰撞等级。
- 格式：

  SetCollisionLevel(level)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
level	int	level: 碰撞等级 0：关闭碰撞检测 1~5：等级越高越灵敏	是	[0,5]

- 返回：

  ErrorID,{},SetCollisionLevel(level);

- 示例：

  SetCollisionLevel(1)//设置碰撞等级为1

3.29 GetAngle

- 功能：获取关节坐标系下机械臂的实时位姿
- 格式：

  GetAngle()

- 支持端口：29999

  返回：

  ErrorID,{J1,J2,J3,J4,J5,J6},GetAngle();

  {J1,J2,J3,J4,J5,J6}表示当前位置的关节坐标值；

- 示例：

GetAngle()//获取当前位置的关节坐标值

返回：

0,{0.0,0.0,90.0,0.0,-90.0,0.0},GetAngle();

3.30 GetPose

- 功能：获取笛卡尔坐标系下机械臂的实时位姿(如果设置了用户坐标系或工具坐标系，则获取的位姿为设置坐标系下的位姿)
- 格式：

  GetPose()
- 支持端口：29999
- 参数详解：0/2
-

参数名	类型	含义	是否必填	默认值	取值范围
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50]
tool=1	string	工具坐标系索引，取值范围：0-50，若不带则为之前设置的全局工具坐标系	否	上一次全局指令设置值	[0,50]

- 返回：

  ErrorID,{X,Y,Z,Rx,Ry,Rz},GetPose();

  {X,Y,Z,Rx,Ry,Rz}表示当前位置的笛卡尔坐标值；
- 示例：

  GetPose() //获取当前用户、工具坐标系下的位姿值

  返回：

  0,{-473.0,141.0,469.0,-180.0,0.0,90.0},GetPose();
- 示例：

  GetPose(user = 1, tool = 0)//获取用户坐标系1，工具坐标系0下的位姿值

  返回：

  0,{-473.0,-141.0,469.0,-180.0,0.0,90.0},GetPose(user = 1, tool = 0);

  注：可选参数可全不带，也可全带，不可只带其中一个

3.31 EmergencyStop

- 功能：急停
- 格式：

EmergencyStop(1)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	取值范围
value	int	1:按下急停　0：松开急停	是	0/1

- 返回：

  ErrorID,{},EmergencyStop(1);

- 示例：

  EmergencyStop(1)//按下急停

3.32 ModbusRTUCreate

- 功能：创建modbusRTU主站。
- 格式：

  ModbusRTUCreate(slave_id,baud,parity,data_bit,stop_bit)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	取值范围
slave_id	int	从站id；	是		
baud	int	波特率	是		
parity	string	校验位, "N": 无校验, "O": 奇校验, "E": 偶校验	否	E	N/O/E
data_bit	int	数据位	否	8	
stop_bit	int	停止位	否	1	

- 返回：

  ErrorID,{index},ModbusRTUCreate((slave_id,baud,parity,data_bit,stop_bit);

  {index}: 返回的主站索引，最多支持5个设备,取值范围(0~4)；

- 示例：

  ModbusRTUCreate(1, 115200, "E", 8, 1)

3.32 ModbusCreate

- 功能：创建modbus主站。
- 格式：

  ModbusCreate(ip,port,slave_id,isRTU)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	取值范围
ip	string	从站ip地址；	是		
port	int	从站端口；	是		
slave_id	int	从站ID)	是		大于等于0的整数
isRTU	int	建立modbus TCP/RTU通信 0：建立modbusTCP通信； 1：建立modbusRTU通信；	否	0	0/1

- 返回：

  ErrorID,{index},ModbusCreate(ip,port,slave_id,isRTU);

  {index}: 返回的主站索引，最多支持5个设备,取值范围0~4；

- 示例：建立RTU通信主站(60000末端透传端口)

  ModbusCreate("127.0.0.1",60000,1,0)

3.33 ModbusClose

- 功能：和Modbus从站断开连接,释放主站。
- 格式：

  ModbusClose(index)

- 支持端口：29999
- 参数详解：

参数名	类型	含义
index	int	返回的主站索引；

- 返回：

  ErrorID,{},ModbusClose(index);

- 示例：

  ModbusClose(0)

3.34 GetInBits

- 功能：读离散输入功能。

- 格式：

  GetInBits(index,addr,count)

- 支持端口：29999
- 参数详解：

参数名	类型	含义
index	int	返回的主站索引；
addr	int	视从站配置而定；
count	int	个数，取值范围1~16；

- 返回：

  ErrorID,{value1,value2,...,valuen},GetInBits(index,addr,count);

  {value1,value2...,valuen}：按位获取结果

- 示例：

  GetInBits(0,3000,5)

  返回：

  0,{1,0,1,1,0},GetInBits(0,3000,5);

3.35 GetInRegs

- 功能：读输入寄存器。
- 格式：

  GetInRegs(index,addr,count,valType)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
index	int	返回的主站索引；	是		
addr	int	视从站配置而定；	是		
count	int	个数，取值范围：1 -4	是		[1,4]

参数名	类型	含义	是否必填	默认值	参数范围
valType	string	数据格式 U16: 读取16位无符号数（2个字节，占用1个寄存器） U32: 读取32位无符号数（4个字节，占用2个寄存器） F32: 读取32位浮点数（4个字节，占用2个寄存器） F64: 读取64位浮点数（8个字节，占用4个寄存器）	否	U16	U16/U32/F32/F64

- 返回：

  ErrorID,{value1,value2,...,valuen},GetInRegs(index,addr,count,valType);

  {value1,value2...,valuen}：按变量类型返回

- 示例：

  GetInRegs(0,4000,3)

  正常返回：

  0,{5,18,12},GetInRegs(0,4000,3);

  错误返回：

  -1,{},GetInRegs(0,4000,3);

3.36 GetCoils

- 功能：读线圈功能。
- 格式：

  GetCoils(index,addr,count)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	返回的主站索引；	是	
addr	int	视从站配置而定；	是	
count	int	个数	是	[1,16]

- 返回：

  ErrorID,{value1,value2,...,valuen},GetCoils(index,addr,count);

  {value1,value2...,valuen}：按变量类型返回

- 示例：

GetCoils(0,1000,3)

正常返回：

0,{1,1,0},GetCoils(0,1000,3);

错误返回：

-1,{},GetCoils(0,1000,3);

3.37 SetCoils

- 功能：写线圈功能。
- 格式：

  SetCoils(index,addr,count,valTab)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	返回的主站索引；	是	
addr	int	视从站配置而定；	是	
count	int	个数	是	[1,16]
valTab	string	写线圈地址值	是	

- 返回：

  ErrorID,{},SetCoils(index,addr,count,valTab);
- 示例：

  SetCoils(0,1000,3,{1,0,1})

  正常返回：

  0,{},SetCoils(0,1000,3,{1,0,1});

  错误返回：

  -1,{},SetCoils(0,1000,3,{1,0,1});

3.38 GetHoldRegs

- 功能：读保持寄存器。
- 格式：

  GetHoldRegs(index,*addr*, *count*,valType)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认参数	参数范围
index	int	index,返回的主站索引，最多支持5个设备	是		[0,4]
addr	int	保持寄存器的起始地址	是		视从站配置而定
count	int	读取指定数量type类型的数据	是		
valType	string	数据类型： U16：读取16位无符号整数（2个字节，占用1个寄存器） U32：读取32位无符号整数（4个字节，占用2个寄存器） F32：读取32位单精度浮点数（4个字节，占用2个寄存器） F64：读取64位双精度浮点数（8个字节，占用4个寄存器）	否	U16	U16/U32/F32/F64

- 返回：

  ErrorID,{value1,value2,...,valuen},GetHoldRegs(index,*addr*, *count*,valType);

  {value1,value2...,valuen}：按变量类型返回

- 示例：从地址3095开始读取一个16位无符号整数

  GetHoldRegs(0,3095,1)

  正常返回：

  0,{13},GetHoldRegs(0,3095,1);

  错误返回：

  -1,{},GetHoldRegs(0,3095,1);

3.39 SetHoldRegs

- 功能：写保存寄存器。
- 格式：

  SetHoldRegs(index,*addr*, *count*,*valTab*,valType)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认参数	参数范围
index	int	index,返回的主站索引，最多支持5个设备,取值范围(0~4)	是		[0,4]
addr	int	保持寄存器的起始地址。视从站配置而定；	是		
count	int	写入指定数量type类型的数据。取值范围：1~4	是		
valTab	string	保持寄存器地址的值	是		
valType	string	数据类型 U16：读取16位无符号整数（2个字节，占用1个寄存器） U32：读取32位无符号整数（4个字节，占用2个寄存器） F32：读取32位单精度浮点数（4个字节，占用2个寄存器） F64：读取64位双精度浮点数（8个字节，占用4个寄存器）	否	U16	U16/U32/F32/F64

- 返回：

  ErrorID,{},SetHoldRegs(index,*addr*, *count*,*valTab*,valType);

- 示例：从地址3095开始，写入两个16位无符号整数值6000，300

  SetHoldRegs(0,3095,2,{6000,300}, U16)

  正常返回：

  0,{},SetHoldRegs(0,3095,2,{6000,300}, U16);

  错误返回：

  -1,{},SetHoldRegs(0,3095,2,{6000,300}, U16);

3.40 GetErrorID

- 功能：获取机器人错误码
- 格式：

  GetErrorID()

- 支持端口：29999
- 返回：

ErrorID,{[[id,...,id], [id], [id], [id], [id], [id], [id]]},GetErrorID();

[id, ..., id]为控制器以及算法报警信息，后面六个[id]分别表示六个伺服的报警信息；

- 示例：

  GetErrorID()

  返回：

  0,{[[],[],[],[],[],[],[]],GetErrorId();

- 说明：对于错误码对应的错误内容请参考控制器错误描述文件alarm_controller.json以及伺服错误描述alarm_servo.json；

3.41 DI

- 功能：获取数字量输入端口状态
- 格式：

  DI(index)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	数字输入索引	是	[1,32] [100,1000]

- 返回：

  ErrorID,{value},DI(index);

  {value}：index的值状态，取值范围0/1；

- 示例：

  DI(1)//返回数字输入端口1为低电平：

  返回：

  0,{0},DI(1);

- 说明：使用取值范围100-1000需要有拓展IO模块的硬件支持；

3.42 ToolDI

- 功能：获取末端数字量输入端口状态
- 格式：

  ToolDI(index)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	末端数字量输入索引	是	1/2（根据机型限定）

- 返回：

  ErrorID,{value},ToolDI(index);

- 示例：

  ToolDI(2)

  返回末端数字输入端口2为高电平：

  0,{1},ToolDI(2);

3.43 AI

- 功能：获取模拟量输入端口模拟值（立即指令）
- 格式：

  AI(index)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	控制柜模拟输入索引	是	1/2

- 返回：

  ErrorID,{value},AI(index);

  {value}：index的值状态，取值范围0/1;

- 示例：

  AI(2)

  返回：模拟输入端口2的模拟值为3.5：

  0,{3.5},AI(2);

3.44 ToolAI

- 功能：获取末端模拟量输入端口模拟值
- 格式：

  ToolAI(index)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	末端模拟输入索引	是	1/2（根据机型限定）

- 返回：

  ErrorID,{value},ToolAI(index);

  {value}：index的值状态，取值范围0/1;

- 示例：

  ToolAI(1)

  返回：模拟输入端口1的电压值为1.5V：

  0,{1.5},ToolAI(1);

3.45 DIGroup

- 功能：获取输入组端口状态
- 格式：

  DIGroup(index1,index2,...,indexn)
- 参数数量：不固定(最大支持64个)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	参数范围
index1	int	数字输入索引	[1,32] [100,1000]
...	...	...	...
indexn	int	数字输入索引	[1,32] [100,1000]

- 返回：

  ErrorID,{value1,value2,...,valuen},DIGroup(index1,index2,...,indexn);

  {value1,...valuen,}：返回当前index1到indexn的电压值;
- 示例：

  DIGroup(4,6,2,7)

  返回：获取的输入4、6、2、7端口的电平分别为1，0，1，1

  0,{1,0,1,1},DIGroup(4,6,2,7);
- 说明：使用取值范围100-1000需要有拓展IO模块的硬件支持；

3.46 DOGroup（队列指令）

- 功能：设置输出组端口状态
- 格式：

  DOGroup(index1,value1,index2,value2,...,indexn,valuen)
- 参数数量：不固定(最大支持64个)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	参数范围
index1	int	设置数字输出索引	[1,16] [100,1000]

参数名	类型	含义	参数范围
value1	int	设置数字输出端口状态	0/1
…	…	…	…
indexn	int	设置数字输出索引	[1,16] [100,1000]
valuen	int	设置数字输出端口状态	0/1

- 返回：

  ErrorID,{},DOGroup(index1,value1,index2,value2,…,indexn,valuen);

- 示例：

  DOGroup(4,1,6,0,2,1,7,0)

  返回：分别设置输出端口4、6、2、7分别为1、0、1、0

  0,{ResultID},DOGroup(4,1,6,0,2,1,7,0);

- 说明：使用取值范围100-1000需要有拓展IO模块的硬件支持；

3.47 BrakeControl

- 功能：开关抱闸
- 格式：

  BrakeControl(axisID,value)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
axisID	int	关节轴号	是	[1,6]
value	int	设置抱闸状态； 0:关闭抱闸 1：打开抱闸	是	0/1

- 返回：

  ErrorID,{},BrakeControl(axisID,value);

- 示例：打开关节1抱闸

  BrakeControl(1,1)

- 返回：

  0,{},BrakeControl(1,1);

  注：抱闸的控制需要机器人在下使能的条件下进行；否则机器人错误返回-1；

3.48 StartDrag

- 功能：进入拖拽(在报错状态下，不可进入拖拽)
- 格式：

StartDrag()

- 支持端口：29999
- 返回：

  ErrorID,{},StartDrag();
- 示例：

  StartDrag()//进入拖拽

3.49 StopDrag

- 功能：退出拖拽（在报错状态下，可以退出拖拽）
- 格式：

  StopDrag()
- 支持端口：29999
- 返回：

  ErrorID,{},StopDrag();
- 示例：

  StopDrag()//退出拖拽

3.50 DragSensivity

- 功能：设置拖拽灵敏度
- 格式：

  DragSensivity(index,value)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	轴号 0代表1到6轴均设置为此灵敏度	是	[0,6]
value	int	轴拖拽灵敏度	是	[1,90]

- 返回：

  ErrorID,{},DragSensivity(1,20);
- 示例：

  DragSensivity(1,20)//设置轴1的拖拽灵敏度为20

3.51 GetDO

- 功能：获取数字输出端口状态
- 格式：

  GetDO(index)
- 参数详解：
参数名	类型	含义	是否必填	参数范围
index	int	输出端口索引：1~16，100~1000	是	[1,16] [100,1000]
- 返回：

  ErrorID,{value},GetDO(1);
- 示例：

  GetDO(1)//获取输出端口1的开关状态

3.52 GetAO

- 功能：获取模拟量输出端口状态
- 格式：

  GetAO(index)
- 支持端口：29999
- 参数详解：
参数名	类型	含义	是否必填	参数范围
index	int	输出端口索引	是	1/2
- 返回：

  ErrorID,{value},GetAO(1);
- 示例：

  GetAO(1)//获取模拟端口1的模拟量

3.53 GetToolDO

- 功能：获取末端数字输出端口状态
- 格式：

  GetDO(index)
- 支持端口：29999
- 参数详解：
参数名	类型	含义	是否必填	参数范围
index	int	输出端口索引	是	1/2

- 返回：

  ErrorID,{value},GetToolDO(1);

- 示例：

  GetToolDO(1)//获取末端输出端口1的状态

3.54 GetDOGroup

- 功能：获取一组数字输出组端口状态
- 格式：

  GetDOGroup(index1, ..., indexn)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	输出端口索引	是	[1,16] [100,1000]

- 返回：

  ErrorID,{value1,value2,value3,...},GetDOGroup(index1,index2,index3,...);

- 示例：

  GetDOGroup(1,2,3)

3.55 SetTool485

- 功能：设置末端485通讯参数
- 格式：

  SetTool485(baudrate)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
baudrate	int	设置的波特率	是		>0
parity	string	设置的校验位 "O":奇校验，"E":偶校验，"N":无校验	否	N	O/E/N
stop	int	设置的停止位	否	1	
identify	int	航插序号1~2	否	1	1/2

- 返回：

  ErrorID,{},SetTool485(115200);

- 示例：

  SetTool485(115200);//设置末端485通讯波特率为115200，无校验位，停止位为1，航插1

3.56 SetSafeWallEnable（队列指令）

- 功能：设置安全墙开关
- 格式：

  SetSafeWallEnable(index,value)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	墙的序号	是	[1,8]
value	int	0：关闭，1：开启	是	0/1

- 返回：

  ErrorID,{},SetSafeWallEnable(1,1);
- 示例：

  SetSafeWallEnable(1,1)//开启安全墙1

3.57 SetToolPower

- 功能：设置末端工具供电状态
- 格式：

  SetToolPower(status)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
status	int	0:关闭，1：开启	是	0/1

- 返回：

  ErrorID,{},SetToolPower(1);
- 示例：

  SetToolPower(1)//开启末端工具电源

3.58 SetToolMode

- 功能：设置末端模式
- 格式：

  SetToolMode(mode, type)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
mode	int	1：485模式，2：采集模式	是		
type	int	采集模式：个位为第一路模式，十位为第二路模式 0：为0-10V电压输入模式 1：为电流采集模式 2：为0-5V电压输入模式	否		>0
identify	int	航插序号1~2，可选参数，不填则为1	否	1	1/2参数名参数名

- 返回：

  ErrorID,{},SetToolMode(1);//设置末端模式为485模式

  ErrorID,{},SetToolMode(2,0);
- 示例：

  SetToolMode(2,0);//设置末端双路模式为0~10V电压输入模式
- 注：

  若mode 为485模式，则仅填一个参数变量

  若mode 为采集模式，则可填2/3个变量

3.59 SetBackDistance（队列指令）

- 功能：设置碰撞回退距离
- 格式：

  SetBackDistance(distance)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
distance	double	碰撞后回退距离	是	[0,50]

- 返回：

  ErrorID,{},SetBackDistance(10);

- 示例：

  SetBackDistance(10);//设置碰撞回退距离为10

3.60 SetPostCollisionMode（队列指令）

- 功能：设置选择碰撞后是进入停止状态还是暂停状态
- 格式：

  SetPostCollisionMode(mode)
- 支持端口：29999
- 参数详情：
参数名	类型	含义	是否必填	参数范围
mode	int	0, 状态后进入停止状态。 1，碰撞后进入暂停状态	是	0/1
- 返回：

  ErrorID,{},SetPostCollisionMode(0);
- 示例：

  SetPostCollisionMode(1);//设置碰撞后进入停止状态

3.61 SetUser

- 功能：设置用户坐标系
- 格式：

  SetUser(index, value)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	坐标系索引	是	[0,50]
value	string	要设置的坐标值{x,y,z,rx,ry,rz}	是	任意值
- 返回：

  ErrorID,{},SetUser(1,{1,2,3,4,5,6});
- 示例：

  SetUser(1,{1,2,3,4,5,6})//设置用户坐标系1的值为{1,2,3,4,5,6}

3.62 SetTool

- 功能：设置工具坐标系
- 格式：

  SetTool(index, value)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	坐标系索引	是	[0,50]
value	string	要设置的坐标值{x,y,z,rx,ry,rz}	是	任意值

- 返回：

  ErrorID,{},SetTool(1,{1,2,3,4,5,6})
- 示例：

  SetTool(1,{1,2,3,4,5,6})//设置工具坐标系1的值为{1,2,3,4,5,6}

3.63 CalcUser

- 功能：计算用户坐标系
- 格式：

  CalcUser(index, matrix,offset)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	坐标系索引	是	[0,50]
matrix	int	计算的方向 1：index对应的用户坐标系左乘{x,y,z,rx,ry,rz} 0：index对应的用户坐标系右乘{x,y,z,rx,ry,rz}	是	0/1
offset	string	偏移的坐标值 {offsetx,offsety,offsetz,offsetrx,offsetry,offsetrz}	是	任意值

- 返回：

  ErrorID,{x,y,z,rx,ry,rz},CalcUser(1,0,{1,2,3,4,5,6});
- 示例：

  CalcUser(1,0,{1,2,3,4,5,6})//计算用户坐标系值为坐标系1左乘{1,2,3,4,5,6}

3.64 CalcTool

- 功能：计算工具坐标系
- 格式：

  CalcTool(index, matrix,offset)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
index	int	坐标系索引	是	[0,50]
matrix	int	计算的方向： 1：index对应的用户坐标系左乘{x,y,z,rx,ry,rz} 0：index对应的用户坐标系右乘{x,y,z,rx,ry,rz}	是	0/1
offset	string	偏移的坐标值 {offsetx,offsety,offsetz,offsetrx,offsetry,offsetrz}	是	任意值

- 返回：

  ErrorID,{x,y,z,rx,ry,rz},CalcTool(1,0,{1,2,3,4,5,6});
- 示例：

  CalcTool(1,0,{1,2,3,4,5,6})//计算工具坐标系值为坐标系1左乘{1,2,3,4,5,6}

3.65 GetInputBool

- 功能：获取输入寄存器bool数值
- 格式：

  GetInputBool(address)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,63]

- 返回：

  ErrorID,{value},GetInputBool(address);
- 示例：

  GetInputBool(0)//获取输入寄存器地址位0的bool值

3.66 GetInputInt

- 功能：获取输入寄存器int数值
- 格式：

  GetInputInt(address)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引：0~23	是	[0,23]

- 返回：

  ErrorID,{value},GetInputInt(address);
- 示例：

  GetInputInt(0))//获取输入寄存器地址位0的int值

3.67 GetInputFloat

- 功能：获取输入寄存器float数值
- 格式：

  GetInputFloat(address)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,23]

- 返回：

  ErrorID,{value},GetInputFloat(address);
- 示例：

  GetInputFloat(0))//获取输入寄存器地址位0的float值

3.68 GetOutputBool

- 功能：获取输出寄存器bool数值
- 格式：

  GetOutputBool(address)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,63]

- 返回：

  ErrorID,{value},GetOutputBool(address);

- 示例：

  GetOutputBool(0))//获取输出寄存器地址位0的bool值

3.69 GetOutputInt

- 功能：获取输出寄存器int数值
- 格式：

  GetOutputInt(address)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,23]

- 返回：

  ErrorID,{value},GetOutputInt(address);

- 示例：

  GetOutputInt(0))//获取输出寄存器地址位0的int值

3.70 GetOutputFloat

- 功能：获取输出寄存器float数值
- 格式：

  GetOutputFloat(address)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,23]

- 返回：

  ErrorID,{value},GetOutputFloat(address);

- 示例：

  GetOutputFloat(0)//获取输出寄存器地址位0的float值

3.71 SetOutputBool

- 功能：设置输出寄存器bool数值
- 格式：

  SetOutputBool(address, value)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,63]
value	int	设置值	是	0/1

- 返回：

  ErrorID,{},SetOutputBool(address, value);
- 示例：

  SetOutputBool(0, 1)//设置输出寄存器地址位0的bool值为1

3.72 SetOutputInt

- 功能：设置输出寄存器int数值
- 格式：

  SetOutputInt(address, value)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,23]
value	int	设置值	是	带符号32位整型

- 返回：

  ErrorID,{},SetOutputInt(address, value);
- 示例：

  SetOutputInt(0, -1)//设置输出寄存器地址位0的bool值为-1

3.73 SetOutputFloat

- 功能：设置输出寄存器float数值
- 格式：

  SetOutputFloat(address, value)
- 支持端口：29999

- 参数详解：

参数名	类型	含义	是否必填	参数范围
address	int	地址索引	是	[0,23]
value	float	设置值	是	单精度浮点

- 返回：

  ErrorID,{},SetOutputFloat(address, value);

- 示例：

  SetOutputFloat(0, 1.23)//设置输出寄存器地址位0的float值为1.23

3.74 MovJ

- 功能：点到点运动，目标点位为笛卡尔点位。
- 格式：

  MovJ(joint = {j1, j2, j3, j4, j5, j6},user = 1, tool = 0, a = 20, v = 50, cp = 100)

  MovJ(pose= {x,y,z,rx,ry,rz},user = 1, tool = 0, a = 20, v = 50, cp = 100)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
joint = {j1, j2, j3, j4, j5, j6} OR pose= {x,y,z,rx,ry,rz}	string	目标点的坐标值	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50]int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50]int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值

- 返回：

  ErrorID,{ResultID},MovJ(pose= {x,y,z,rx,ry,rz},user = 1, tool = 0, a = 20, v = 50, cp = 100);

  ResultID:算法队列ID

- 示例：

  MovJ(joint = {1, 2, 3, 4, 5, 6},user = 1, tool = 0, a = 20, v = 50, cp = 100)

  返回：

  ErrorID,{1},MovJ(joint = {1, 2, 3, 4, 5, 6},user = 1, tool = 0, a = 20, v = 50, cp = 100);

注：关节变量进行关节运动，user/tool参数设置无效

3.75 MovL

- 功能：直线运动，目标点位为笛卡尔点位。
- 格式：

  MovL(joint = {j1, j2, j3, j4, j5, j6},user = 1, tool = 0, a = 20, v = 50, cp = 100)

  MovL(pose= {x,y,z,rx,ry,rz},user = 1, tool = 0, a = 20, v = 50, cp = 100)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
joint = {j1, j2, j3, j4, j5, j6} OR pose= {x,y,z,rx,ry,rz}	string	点的坐标值	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值
speed=1000	string	速度数值	否		>0 int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值
r=20	string	过渡半径，单位mm	否		>0 int值

- 返回：

  ErrorID,{ResultID},MovL(pose= {x,y,z,rx,ry,rz},user = 1, tool = 0, a = 20, v = 50, cp = 100) ;

  ResultID:算法队列ID
- 示例：

MovL(pose= {x,y,z,rx,ry,rz},user = 1, tool = 0, a = 20, v = 50, cp = 100)

返回：

ErrorID,{2},MovL(pose= {x,y,z,rx,ry,rz},user = 1, tool = 0, a = 20, v = 50, cp = 100) ;

注：

运动指令参数cp 和r 同时存在，以r为优先

运动指令参数v 和speed 同时存在，以和speed为优先

3.76 MovLIO

- 功能：在直线运动时并行设置数字输出端口状态，目标点位为笛卡尔点位。
- 格式：

  MovLIO(joint = {j1, j2, j3, j4, j5, j6},{Mode,Distance,Index,Status},..., {Mode,Distance,Index,Status},user = 1, tool = 0, a = 20, v = 50, cp = 100)

  MovLIO(pose= {x,y,z,rx,ry,rz},{Mode,Distance,Index,Status},..., {Mode,Distance,Index,Status},user = 1, tool = 0, a = 20, v = 50, cp = 100)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
joint = {j1, j2, j3, j4, j5, j6}|pose= {x,y,z,rx,ry,rz}	string	目标点的坐标值	是		
{Mode,Distance,Index,Status}...	string	可设置多组，最少一组数据	是		详见下表
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值

参数名	类型	含义	是否必填	默认值	参数范围
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值
speed=1000	string	速度数值	否		>0 int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值
r=20	string	过渡半径，单位mm	否		>0 int值

-

{Mode,Distance,Index,Status}	类型	含义	是否必填	参数范围
Mode	int	设置Distance模式 0：Distance为距离百分比； 1：Distance为离起始点或目标点的距离	是	0/1
Distance	int	运行指定的距离: 若Mode为0，则Distance表示起始点与目标点之间距离的百分比 若Distance取值为正，则表示离起始点的距离; 若Distance取值为负，则表示离目标点的距离	是	[0,100] OR 任意值
Index	int	数字输出索引	是	[1,16] [100,1000]
Status	int	数字输出状态	是	0/1

- 返回：ResultID:算法队列ID

  ErrorID,{ResultID},MovLIO(pose= {x,y,z,rx,ry,rz},{Mode,Distance,Index,Status},...,{Mode,Distance,Index,Status},user = 1, tool = 0, a = 20, v = 50, cp = 100);

- 示例：

  MovLIO(joint= {1,2,3,4,56},{0,50,1,0})

注：运动指令参数cp 和r 同时存在，以r为优先

运动指令参数v 和speed 同时存在，以和speed为优先

若Mode为0，Distance不在[0,100]范围内，则报参数超限错误

3.77 MovJIO

- 功能：点到点运动时并行设置数字输出端口状态，目标点位为笛卡尔点位。
- 格式：
- MovJIO(joint = {j1, j2, j3, j4, j5, j6},{Mode,Distance,Index,Status},...,{Mode,Distance,Index,Status},user = 1, tool = 0, a = 20, v = 50, cp = 100)
- MovJIO(pose= {x,y,z,rx,ry,rz},{Mode,Distance,Index,Status},...,{Mode,Distance,Index,Status},user = 1, tool = 0, a = 20, v = 50, cp = 100)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
joint = {j1, j2, j3, j4, j5, j6}|pose= {x,y,z,rx,ry,rz}	string	目标点的坐标值	是		
{Mode,Distance,Index,Status}...	string	可设置多组，最少一组数据	是		详见下表
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值

参数名	类型	含义	是否必填	默认值	参数范围
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值

-

{Mode,Distance,Index,Status}	类型	含义	是否必填	参数范围
Mode	int	设置Distance模式 0：Distance为距离百分比; 1：Distance为离起始点或目标点的距离	是	0/1
Distance	int	运行指定的距离: 若Mode为0，则Distance表示起始点与目标点之间距离的百分比 若Distance取值为正，则表示离起始点的距离; 若Distance取值为负，则表示离目标点的距离	是	[0,100] OR 任意值
Index	int	数字输出索引	是	[1,16] [100,1000]
Status	int	数字输出状态	是	0/1

- 返回：ResultID:算法队列ID

  ErrorID,{3},MovJIO(pose= {x,y,z,rx,ry,rz},{Mode,Distance,Index,Status},...,{Mode,Distance,Index,Status},user = 1, tool = 0, a = 20, v = 50, cp = 100);

- 示例：

  MovJIO(joint= {1,2,3,4,5,6},{0,50,1,0})

3.78 Arc

- 功能：：从当前位置以圆弧插补方式移动至笛卡尔坐标系下的目标位置。

  该指令需结合其他运动指令确定圆弧起始点。

- 格式：

  Arc(joint = {j1, j2, j3, j4, j5, j6},joint = {j1, j2, j3, j4, j5, j6},user = 1, tool = 0, a = 20, v = 50, cp = 100, ori_mode=1)

Arc(pose= {x,y,z,rx,ry,rz},pose= {x,y,z,rx,ry,rz},user = 1, tool = 0, a = 20, v = 50, cp = 100 , ori_mode=1)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
joint = {j1, j2, j3, j4, j5, j6} OR pose= {x,y,z,rx,ry,rz}	string	表示圆弧中间点坐标值	是		任意值
joint = {j1, j2, j3, j4, j5, j6} OR pose= {x,y,z,rx,ry,rz}	string	表示圆弧目标点坐标值	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值
speed=1000	string	速度数值	否		>0 int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值
r=20	string	过渡半径，单位mm	否		>0 int值
ori_mode	int	0表示起始按姿态按Slerp插值，目标点姿态可达； 1表示起始姿态按圆弧Z轴旋转，目标点姿态不可达	否	0	0/1

- 返回：ResultID:算法队列ID

ErrorID,{},Arc(joint = {j1, j2, j3, j4, j5, j6},joint = {j1, j2, j3, j4, j5, j6},user = 1, tool = 0, a = 20, v = 50, cp = 100, ori_mode=1);

- 示例：

  Arc(joint = {1, 2, 3, 4, 5, 6},joint = {7, 8, 9, 10, 11, 12},user = 1, tool = 0, a = 20, v = 50, cp = 100, ori_mode=1)

  注：

  运动指令参数cp 和r 同时存在，以r为优先

  运动指令参数v 和speed 同时存在，以和speed为优先

3.79 Circle

- 功能：：整圆运动
- 格式：

  Circle(joint = {j1, j2, j3, j4, j5, j6},joint = {j1, j2, j3, j4, j5, j6},1,user = 1, tool = 0, a = 20, v = 50, cp = 100)

  Circle(pose= {x,y,z,rx,ry,rz},pose= {x,y,z,rx,ry,rz},1,user = 1, tool = 0, a = 20, v = 50, cp = 100)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
joint = {j1, j2, j3, j4, j5, j6} | pose= {x,y,z,rx,ry,rz}	string	表示圆弧中间点坐标值	是		
joint = {j1, j2, j3, j4, j5, j6} | pose= {x,y,z,rx,ry,rz}	string	表示圆弧目标点坐标值	是		
count	int	整圆个数	是		>0
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值

参数名	类型	含义	是否必填	默认值	参数范围
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值
speed=1000	string	速度数值	否		>0 int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值
r=20	string	过渡半径，单位mm	否		>0 int值

返回：

ErrorID,{ResultID},Circle(joint = {j1, j2, j3, j4, j5, j6},joint = {j1, j2, j3, j4, j5, j6},1， user = 1, tool = 0, a = 20, v = 50, cp = 100);

ResultID:算法队列ID

- 示例：

  Circle(joint = {1, 2, 3, 4, 5, 6},joint = {7, 8, 9, 10, 11, 12},1,user = 1, tool = 0, a = 20, v = 50, cp = 100);

  注：运动指令参数cp 和r 同时存在，以r为优先

  运动指令参数v 和speed 同时存在，以和speed为优先

3.80 MoveJog

- 功能：点动运动，不固定距离运动
- 格式：

  MoveJog(axisID,coordtype=typeValue,user=index,tool=index)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围

参数名	类型	含义	是否必填	默认值	参数范围
axisID	string	点动运动轴 J1+ 表示关节1正方向运动 J1- 表示关节1负方向运动 J2+ 表示关节2正方向运动 J2- 表示关节2负方向运动 J3+ 表示关节3正方向运动 J3- 表示关节3负方向运动 J4+ 表示关节4正方向运动 J4- 表示关节4负方向运动 J5+ 表示关节5正方向运动 J5- 表示关节5负方向运动 J6+ 表示关节6正方向运动 J6- 表示关节6负方向运动 X+ 表示X轴正方向运动 X- 表示X轴负方向运动 Y+ 表示Y轴正方向运动 Y- 表示Y轴负方向运动 Z+ 表示Z轴正方向运动 Z- 表示Z轴负方向运动 Rx+ 表示Rx轴正方向运动 Rx- 表示Rx轴负方向运动 Ry+ 表示Ry轴正方向运动 Ry- 表示Ry轴负方向运动 Rz+ 表示Rz轴正方向运动 Rz- 表示Rz轴负方向运动	是		指定字符串值
coordtype=0	string	0:关节点动 1:用户坐标系 2:工具坐标系	否	上一次全局指令设置值	0/1/2
user=1	string	用户索引	否	上一次全局指令设置值	[0,50]
tool=1	string	工具索引	否	上一次全局指令设置值	[0,50]

- 返回：

  ErrorID,{},MoveJog(axisID,coordType=typeValue,user=index,tool=index);

- 示例（1）：

  MoveJog(J2-) //J2负方向运动，再停止点动

- 返回：

  0,{},MoveJog(J2-);

- 示例（2）：

  MoveJog() //停止运动

- 返回：

  0,{},MoveJog();

- 说明：

  其中用户若是在发关节点动运行则会忽略CoordType、User以及Tool这三个可选设置参数；

  命令下发后，须另外下发MoveJog()停止命令控制机器人停止运动；另下发非指定string内容的参数都会导致机器人停止；

3.81 StartPath

- 功能：轨迹复现。

  StartPath(traceName,isConst=0,multi=1,user=0,tool=0)

- 支持端口：29999

- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
traceName	string	轨迹文件名（.csv） 轨迹路径存放在/dobot/userdata/project/process/trajectory/	是		.csv文件
isConst=0	string	1:为匀速复现；0：非匀速复现	否	0	0/1
multi=1	string	复现倍率	否	1	[0.25,2]
user=0	string	用户坐标系索引	否	上一次全局指令设置值	[0,50]
tool=0	string	工具坐标系索引	否	上一次全局指令设置值	[0,50]

- 返回：

  ErrorID,{},StartPath(traceName);

- 示例：

  StartPath(recv_string.csv)

  返回：

  0,{},StartPath(recv_string.csv);

- 注：

  需将发送端的编码方式选择为UTF-8格式，否则会导致中文字符接收异常

  当前轨迹复现指令不会将轴走到首点，需调用获取首点指令后将轴运动到首点，再进行轨迹复现。

用户可以通过获取RobotMode查询机器人运行状态

若在ROBOT_MODE_RUNNING表示机器人在轨迹拟合运行中，达到ROBOT_MODE_ENABLE表示轨迹拟合运行完成，ROBOT_MODE_ERROR表示报警；

3.82 GetStartPose

- 功能： 获取轨迹的第一个点位
- 格式：

  GetStartPose(traceNamel)
- 支持端口：29999
- 参数详解：
-

参数名	类型	含义	是否必填	参数范围
traceName	string	轨迹文件名（.csv） 轨迹路径存放在/dobot/userdata/project/process/trajectory/	是	.csv文件

- 示例：

  GetStartPose(recv_string.csv)//获取名字recv_string的轨迹的首个点的对应信息
- 返回：

  ErrorID,{pointtype,{j1,j2,j3,j4,j5,j6},user,tool,{x,y,z,rx,ry,rz}},GetStartPose(traceNamel);

  {pointtype}:返回点位类型，0:示教常量；1：关节变量；2：位姿变量
- 示例：

  ErrorID,{0,{10,20,20,20,20,20},1,1,{x,y,z,rx,ry,rz}},GetStartPose(“test.csv”);

  ErrorID,{1,{j1,j2,j3,j4,j5,j6}},GetStartPose(traceNamel);

  ErrorID,{2,{x,y,z,rx,ry,rz}},GetStartPose(traceNamel);

3.83 RelMovJTool

- 功能：沿工具坐标系进行相对运动指令，末端运动方式为关节运动。
- 格式：

  RelMovJTool(x, y, z, rx, ry, rz, user = 1, tool = 0, a = 20, v = 50, cp = 100)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
x	double	X轴方向偏移，单位：mm	是		任意值

参数名	类型	含义	是否必填	默认值	参数范围
y	double	Y轴方向偏移，单位：mm	是		任意值
z	double	Z轴方向偏移，单位：mm	是		任意值
rx	double	Rx 轴位置，单位：度	是		任意值
ry	double	Ry 轴位置，单位：度	是		任意值
rz	double	Rz 轴位置，单位：度	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int 值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int 值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int 值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int 值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int 值

- 返回：

  ErrorID,{ResultID},RelMovJTool(x, y, z, rx, ry, rz, user = 1, tool = 0, a = 20, v = 50, cp = 100);

  {ResultID}:算法队列ID

- 示例：

  RelMovJTool(10,10,10,0,0,0,0)

3.84 RelMovLTool

- 功能：沿工具坐标系进行相对运动指令，末端运动方式为直线运动。
- 格式：

  RelMovLTool(x, y, z, rx, ry, rz, user = 1, tool = 0, a = 20, v = 50, cp = 100)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围

参数名	类型	含义	是否必填	默认值	参数范围
x	double	X轴方向偏移，单位：mm	是		任意值
y	double	Y轴方向偏移，单位：mm	是		任意值
z	double	Z轴方向偏移，单位：mm	是		任意值
rx	double	Rx 轴位置，单位：度	是		任意值
ry	double	Ry 轴位置，单位：度	是		任意值
rz	double	Rz 轴位置，单位：度	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int 值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int 值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int 值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int 值
speed=1000	string	速度数值	否		>0 int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int 值
r=20	string	过渡半径，单位 mm	否		>0 int值

- 返回：

  ErrorID,{ResultID},RelMovLTool(x, y, z, rx, ry, rz, user = 1, tool = 0, a = 20, v = 50, cp = 100);

  {ResultID}:算法队列ID

- 示例：

  RelMovLTool(10,10,10,0,0,0)

3.85 RelMovJUser

- 功能：沿用户坐标系进行相对运动指令，末端运动方式为关节运动。
- 格式：

RelMovJUser(x, y, z, rx, ry, rz, user = 1, tool = 0, a = 20, v = 50, cp = 100)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
x	double	X轴方向偏移，单位：mm	是		任意值
y	double	Y轴方向偏移，单位：mm	是		任意值
z	double	Z轴方向偏移，单位：mm	是		任意值
rx	double	Rx 轴位置，单位：度	是		任意值
ry	double	Ry 轴位置，单位：度	是		任意值
rz	double	Rz 轴位置，单位：度	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int 值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int 值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int 值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int 值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int 值

- 返回：

  ErrorID,{ResultID},RelMovJUser(x,y,z,rx,ry,rz,user = 1, tool = 0, a = 20, v = 50, cp = 100);

  {ResultID}:算法队列ID

- 示例：

  RelMovJUser(10,10,10,0,0,0,0)

3.86 RelMovLUser

- 功能：沿用户坐标系进行相对运动指令，末端运动方式为直线运动。
- 格式：

  RelMovLUser(x, y, z, rx, ry, rz, user = 1, tool = 0, a = 20, v = 50, cp = 100)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
x	double	X轴方向偏移，单位：mm	是		任意值
y	double	Y轴方向偏移，单位：mm	是		任意值
z	double	Z轴方向偏移，单位：mm	是		任意值
rx	double	Rx 轴位置，单位：度	是		任意值
ry	double	Ry 轴位置，单位：度	是		任意值
rz	double	Rz 轴位置，单位：度	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值
speed=1000	string	速度数值	否		>0 int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值

- 返回：

  ErrorID,{ResultID},RelMovLUser(x,y,z,rx,ry,rz,user = 1, tool = 0, a = 20, v = 50, cp = 100);

  {ResultID}:算法队列ID

- 示例：

  RelMovLUser(10,10,10,0,0,0)

  注：

  运动指令参数cp 和r 同时存在，以r为优先

  运动指令参数v 和speed 同时存在，以和speed为优先

3.87 RelJointMovJ

- 功能：沿各轴关节坐标系进行相对运动指令，末端运动方式为关节运动。
- 格式：

  RelJointMovJ(Offset1, Offset2, Offset3, Offset4, Offset5, Offset6)

  RelJointMovJ(Offset1, Offset2, Offset3, Offset4, Offset5, Offset6, user = 1, tool = 0, a = 20, v = 50, cp = 100)
- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	默认值	参数范围
Offset1	double	关节1的偏移值，单位：度	是		任意值
Offset2	double	关节2的偏移值，单位：度	是		任意值
Offset3	double	关节3的偏移值，单位：度	是		任意值
Offset4	double	关节4的偏移值，单位：度	是		任意值
Offset5	double	关节5的偏移值，单位：度	是		任意值
Offset6	double	关节6的偏移值，单位：度	是		任意值
user=1	string	用户坐标系索引	否	上一次全局指令设置值	[0,50] int值
tool=1	string	工具坐标系索引	否	上一次全局指令设置值	[0,50] int值
a=20	string	加速度百分比	否	上一次全局指令设置值	[1,100]int值
v=50	string	速度百分比	否	上一次全局指令设置值	[1,100]int值
cp=100	string	过渡百分比	否	上一次全局指令设置值	[0,100]int值

- 返回：

  ErrorID,{ResultID},RelJointMovJ(Offset1,Offset2,Offset3,Offset4,Offset5,Offset6,user = 1, tool = 0, a = 20, v = 50, cp = 100);

  {ResultID}:算法队列ID
- 示例：

RelJointMovJ(10,10,10,0,0,0)

3.88 GetCurrentCommandID()

- 功能：返回当前算法运行队列ID
- 格式：

  GetCurrentCommandID()
- 支持端口：29999
- 返回：

  ErrorID,{ResultID},GetCurrentCommandID();
- 示例：

  0,{100},GetCurrentCommandID();
- 示例：队列指令怎么判断执行完毕?

  队列指令为立即返回指令，接口返回成功仅代表发送成功，不代表执行完毕。若判断执行完毕，则需要结合CommandID和RobotMode来综合判断

```
MovJ(P1)
uint64_t p2Id = parseResultId(MovJ(P2));
MovJ(P3)

while(true) {
  uint64_t currentId = parseResultId (GetCurrentCommndID());
  if (currentId > p2Id) { // currentId执行到P2后面的点。
    break; // 退出等待，P2点执行完成
  }
  Sleep(1);
}
```

```
MovJ(P1)
uint64_t p2Id = parseResultId(MovJ(P2));

while(true) {
  uint64_t currentId = parseResultId (GetCurrentCommndID());
  bool isStop = parseResultId (RobotMode ()) == 5; // ROBOT_MODE_ENABLE
  if (currentId == p2Id && isStop ) { // currentId等于p2Id时P2点并执行完成。
    break; // 退出等待，P2点执行完成
  }
  Sleep(1);
}
```

3.89 EnableSafeSkin()

- 功能：启动/停止安全皮肤功能
- 格式：

  EnableSafeSkin(status)
- 支持端口：29999

- 参数详解：

参数名	类型	含义	是否必填	参数范围
status	int	0代表关闭，1代表开启	是	0/1

- 返回：

  ErrorID,{result},EnableSafeSkin();

- 示例：

  EnableSafeSkin(1);

返回值	类型	含义
result	int	有无电子皮肤： 0：无 ， 1：有

3.90 SetSafeSkin()

- 功能：设置安全皮肤的灵敏度
- 格式：

  SetSafeSkin(part, status)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
part	int	皮肤部件 arm（小臂安全皮肤）=3、J4=4、J5=5、J6=6	是	3/4/5/6
status	int	灵敏度阈值 0表示关闭，1表示low,2表示middle，3表示high	是	0/1/2/3

- 返回：

  ErrorID,{},SetSafeSkin(part, status);

- 示例：

  格式：SetSafeSkin(3, 1);

3.91 ServoJ()

- 功能：基于关节空间的动态跟随命令。
- 格式：

  ServoJ(J1,J2,J3,J4,J5,J6, t=0.1, aheadtime=50, gain=500)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
J1	double	点J1 轴位置，单位：度	是	
J2	double	点J2 轴位置，单位：度	是	
J3	double	点J3 轴位置，单位：度	是	
J4	double	点J4 轴位置，单位：度	是	
J5	double	点J5 轴位置，单位：度	是	
J6	double	点J6 轴位置，单位：度	是	
t	float	该点位的运行时间，默认0.1,单位：s	否	[0.02,3600.0]
aheadtime	float	作用类似于PID的D项，默认50，标量，无单位	否	[20.0,100.0]
gain	float	目标位置的比例放大器，作用类似于PID的P项，默认500，标量，无单位	否	[200.0,1000.0]

- 返回：

  ErrorID,{},ServoJ(J1,J2,J3,J4,J5,J6, t=0.1, aheadtime=50, gain=500);

- 示例：

  格式：ServoJ(0,0,-90,0,90,0)

3.92 ServoP()

- 功能：基于笛卡尔空间的动态跟随命令。
- 格式：

  ServoP(X,Y,Z,Rx,Ry,Rz, t=0.1, aheadtime=50, gain=500)

- 支持端口：29999
- 参数详解：

参数名	类型	含义	是否必填	参数范围
X	double	X 轴位置，单位：毫米	是	
Y	double	Y 轴位置，单位：毫米	是	
Z	double	Z 轴位置，单位：毫米	是	
Rx	double	Rx 轴位置，单位：度	是	
Ry	double	Ry 轴位置，单位：度	是	

参数名	类型	含义	是否必填	参数范围
Rz	double	Rz 轴位置，单位：度	是	
t	float	该点位的运行时间，默认0.1,单位：s	否	[0.02,3600.0]
aheadtime	float	作用类似于PID的D项，默认50，标量，无单位	否	[20.0,100.0]
gain	float	目标位置的比例放大器，作用类似于PID的P项，默认500，标量，无单位	否	[200.0,1000.0]

- 返回：

  ErrorID,{},ServoP(X,Y,Z,Rx,Ry,Rz, t=0.1, aheadtime=50, gain=500);

- 示例：

  格式：ServoP(-500,100,200,150,0,90)

4.实时反馈端口

30004服务器端口，**每8ms**能收到一次机器人反馈信息

30005服务器端口**每200ms**能收到一次机器人反馈信息

30006端口为**可配置**的反馈机器人信息端口(默认为每**50ms**反馈)，30006端口的实时数据的配置更新可以在线修改后，实时生效；

通过反馈端口每次收到的数据包有1440个字节，这些字节以标准的格式排列，数据以小端方式储存。

意义/Meaning	值的数目/Number of values	数据类型/Type	描述/Notes	字节大小/Size in bytes	字节位置值/Byte position valueCR	支持产品	是否实现
MessageSize	1	unsigned short	消息字节总长度/Total message length in bytes	2	0000 ~ 0001	CR/Nova/ED6	√
reserved	3	unsigned short	保留位	6	0002 ~ 0007	CR/Nova/ED6	
DigitalInputs	1	uint64	数字输入/Current state of the digital inputs.	8	0008 ~ 0015	CR/Nova/ED6	√
DigitalOutputs	1	uint64	数字输出	8	0016 ~ 0023	CR/Nova/ED6	√
RobotMode	1	uint64	机器人模式/Robot mode	8	0024 ~ 0031	CR/Nova/ED6	√
TimeStamp	1	uint64	机器人系统时间戳	8	0032 ~ 0039	CR/Nova/ED6	√
RunTime	1	uint64	机器人开机运行时间（单位ms）	8	0040 ~ 0047	CR/Nova/ED6	√
TestValue	1	uint64	内存结构测试标准值 0x0123 4567 89AB CDEF	8	0048 ~ 0055	CR/Nova/ED6	√
	1	double	保留位	8	0056 ~ 0063	CR/Nova/ED6	

意义/Meaning	值的数目/Number of values	数据类型/Type	描述/Notes	字节大小/Size in bytes	字节位置值/Byte position valueCR	支持产品	是否实现
SpeedScaling	1	double	速度比例/Speed scaling of the trajectory limiter	8	0064 ~ 0071	CR/Nova/ED6	√
LinearMomentumNorm	1	double	机器人当前动量/Norm of Cartesian linear momentum(需要特定硬件版本)	8	0072 ~ 0079	CR/Nova/ED6	×
VMain	1	double	控制板电压/Masterboard: Main voltage	8	0080 ~ 0087	CR/Nova/ED6	×
VRobot	1	double	机器人电压/Masterboard: Robot voltage (48V)	8	0088 ~ 0095	CR/Nova/ED6	√
IRobot	1	double	机器人电流/Masterboard: Robot current	8	0096 ~ 0103	CR/Nova/ED6	√
ProgramState	1	double	脚本运行状态	8	0104 ~ 0111	CR/Nova/ED6	√
SafetyStatus	1	double	安全状态	8	0112 ~ 0119	CR/Nova/ED6	×
ToolAcceleroMeter	3	double	TCP加速度/Tool x,y and z accelerometer values(需要特定硬件版本)	24	0120 ~ 0143	CR/Nova/ED6	×
ElbowPosition	3	double	肘位置/Elbow position(需要特定硬件版本)	24	0144 ~ 0167	CR/Nova/ED6	×
ElbowVelocity	3	double	肘速度/Elbow velocity(需要特定硬件版本)	24	0168 ~ 0191	CR/Nova/ED6	×
QTarget	6	double	目标关节位置/Target joint positions	48	0192 ~ 0239	CR/Nova/ED6	√
QDTarget	6	double	目标关节速度/Target joint velocities	48	0240 ~ 0287	CR/Nova/ED6	√
QDDTarget	6	double	目标关节加速度/Target joint accelerations	48	0288 ~ 0335	CR/Nova/ED6	√
ITarget	6	double	目标关节电流/Target joint currents	48	0336 ~ 0383	CR/Nova/ED6	√
MTarget	6	double	目标关节扭矩/Target joint moments (torques)	48	0384 ~ 0431	CR/Nova/ED6	√
QActual	6	double	实际关节位置/Actual joint positions	48	0432 ~ 0479	CR/Nova/ED6	√
QDActual	6	double	实际关节速度/Actual joint velocities	48	0480 ~ 0527	CR/Nova/ED6	√
IActual	6	double	实际关节电流/Actual joint currents	48	0528 ~ 0575	CR/Nova/ED6	√
ActualTCPForce	6	double	TCP传感器力值(通过六维力计算)	48	0576 ~ 0623	CR/Nova/ED6	×
ToolVectorActual	6	double	TCP笛卡尔实际坐标值/Actual Cartesian coordinates of the tool: (x,y,z,rx,ry,rz), where rx, ry and rz is a rotation vector representation of the tool orientation	48	0624 ~ 0671	CR/Nova/ED6	√

意义/Meaning	值的数目/Number of values	数据类型/Type	描述/Notes	字节大小/Size in bytes	字节位置值/Byte position valueCR	支持产品	是否实现
TCPSpeedActual	6	double	TCP笛卡尔实际速度值/Actual speed of the tool given in Cartesian coordinates	48	0672 ~ 0719	CR/Nova/ED6	√
TCPForce	6	double	TCP力值（通过关节电流计算）	48	0720 ~ 0767	CR/Nova/ED6	√
ToolVectorTarget	6	double	TCP笛卡尔目标坐标值/Target Cartesian coordinates of the tool: (x,y,z,rx,ry,rz), where rx, ry and rz is a rotation vector representation of the tool orientation	48	0768 ~ 0815	CR/Nova/ED6	√
TCPSpeedTarget	6	double	TCP笛卡尔目标速度值/Target speed of the tool given in Cartesian coordinates	48	0816 ~ 0863	CR/Nova/ED6	√
MotorTemperatures	6	double	关节温度/Temperature of each joint in degrees celsius	48	0864 ~ 0911	CR/Nova/ED6	√
JointModes	6	double	关节控制模式/Joint control modes	48	0912 ~ 0959	CR/Nova/ED6	√
VActual	6	double	关节电压/Actual joint voltages	48	960 ~ 1007	CR/Nova/ED6	√
HandType	4	char	手系(备用参数)	4	1008 ~ 1011	CR/Nova/ED6	√
User	1	char	全局用户坐标系索引	1	1012	CR/Nova/ED6	√
Tool	1	char	全局工具坐标系索引	1	1013	CR/Nova/ED6	√
RunQueuedCmd	1	char	算法队列运行标志	1	1014	CR/Nova/ED6	√
PauseCmdFlag	1	char	算法队列暂停标志	1	1015	CR/Nova/ED6	√
VelocityRatio	1	char	关节速度比例(0~100)	1	1016	CR/Nova/ED6	√
AccelerationRatio	1	char	关节加速度比例(0~100)	1	1017	CR/Nova/ED6	√
JerkRatio	1	char	关节加加速度比例(0~100)	1	1018	CR/Nova/ED6	×
XYZVelocityRatio	1	char	笛卡尔位置速度比例(0~100)	1	1019	CR/Nova/ED6	√
RVelocityRatio	1	char	笛卡尔姿态速度比例(0~100)	1	1020	CR/Nova/ED6	√
XYZAccelerationRatio	1	char	笛卡尔位置加速度比例(0~100)	1	1021	CR/Nova/ED6	√
RAccelerationRatio	1	char	笛卡尔姿态加速度比例(0~100)	1	1022	CR/Nova/ED6	√
XYZJerkRatio	1	char	笛卡尔位置加加速度比例(0~100)	1	1023	CR/Nova/ED6	×
RJerkRatio	1	char	笛卡尔姿态加加速度比例(0~100)	1	1024	CR/Nova/ED6	×
BrakeStatus	1	char	机器人抱闸状态	1	1025	CR/Nova/ED6	√
EnableStatus	1	char	机器人使能状态	1	1026	CR/Nova/ED6	√
DragStatus	1	char	机器人拖拽状态	1	1027	CR/Nova/ED6	√
RunningStatus	1	char	机器人运行状态	1	1028	CR/Nova/ED6	√
ErrorStatus	1	char	机器人报警状态	1	1029	CR/Nova/ED6	√

意义/Meaning	值的数目/Number of values	数据类型/Type	描述/Notes	字节大小/Size in bytes	字节位置值/Byte position valueCR	支持产品	是否实现
JogStatusCR	1	char	机器人点动状态	1	1030	CR/Nova/ED6	√
CRRobotType	1	char	机器类型	1	1031	CR/Nova/ED6	√
DragButtonSignal	1	char	按钮板拖拽信号	1	1032	CR/Nova/ED6	√
EnableButtonSignal	1	char	按钮板使能信号	1	1033	CR/Nova/ED6	√
RecordButtonSignal	1	char	按钮板录制信号	1	1034	CR/Nova/ED6	√
ReappearButtonSignal	1	char	按钮板复现信号	1	1035	CR/Nova/ED6	√
JawButtonSignal	1	char	按钮板夹爪控制信号	1	1036	CR/Nova/ED6	√
SixForceOnline	1	char	六维力在线状态	1	1037	CR/Nova/ED6	×
CollisionState	1	char	碰撞状态	1	1038	CR/Nova/ED6	√
ArmApproachState	1	char	小臂接近暂停状态	1	1039	CR/Nova/ED6	√
J4ApproachState	1	char	J4接近暂停	1	1040	CR/Nova/ED6	√
J5ApproachState	1	char	J5接近暂停	1	1041	CR/Nova/ED6	√
J6ApproachState	1	char	J6接近暂停	1	1042	CR/Nova/ED6	√
Reserve2[61]	1	char	保留位	61	1043-1103	CR/Nova/ED6	√
VibrationDisZ	1	double	加速度计测量Z轴抖动位移	8	1104~1111	CR/Nova/ED6	√
CurrentCommandId	1	uint64	当前运动队列id	8	1112~1119	CR/Nova/ED6	√
MActual[6]	6	double	实际扭矩	48	1120 ~ 1167	CR/Nova/ED6	√
Load	1	double	负载重量kg	8	1168-1175	CR/Nova/ED6	√
CenterX	1	double	X方向偏心距离mm	8	1176-1183	CR/Nova/ED6	√
CenterY	1	double	Y方向偏心距离mm	8	1184-1191	CR/Nova/ED6	√
CenterZ	1	double	Z方向偏心距离mm	8	1192-1199	CR/Nova/ED6	√
User[6]	6	double	用户坐标值	48	1200-1247	CR/Nova/ED6	√
Tool[6]	6	double	工具坐标值	48	1248-1295	CR/Nova/ED6	√
TraceIndex	1	double	轨迹复现运行索引	8	1296-1303	CR/Nova/ED6	×
SixForceValue[6]	6	double	当前六维力数据原始值	48	1304-1351	CR/Nova/ED6	√
TargetQuaternion[4]	4	double	[qw,qx,qy,qz] 目标四元数	32	1352-1383	CR/Nova/ED6	√
ActualQuaternion[4]	4	double	[qw,qx,qy,qz] 实际四元数	32	1384-1415	CR/Nova/ED6	√
AutoManualMode	1	char	手自动模式	2	1416~1417	CR/Nova/ED6	√
Reserve3[22]	1	char	保留位	22	1418 ~ 1439	CR/Nova/ED6	
TOTAL			1440byte package	1440			

说明：

- RobotMode返回机器人模式：
- BrakeStatus抱闸状态：

  0x01表示第六个轴抱闸打开；

  0x02表示第五个轴抱闸打开；

  0x03表示五六轴抱闸打开；

  0x04表示第四个轴抱闸打开；

...

7	6	5	4	3	2	1	0
保留位	保留位	关节一	关节二	关节三	关节四	关节五	关节六

- JointModes关节控制模式：

  当前值为8表示位置模式；

  当前值为10表示力矩模式；

- RobotType表示机器类型：

RobotType值	代表机型
3	CR3
113	CR3A
5	CR5
115	CR5A
7	CR7
117	CR7A
10	CR10
120	CR10A
12	CR12
122	CR12A
16	CR16
126	CR16A
130	CR20A
101	Nova2
103	Nova5
150	Magician E6

5.错误码描述

错误码	描述	备注
0	无错误	下发成功
-1	没有执行成功	命令执行失败

错误码	描述	备注
-2	当前处于错误状态拒绝执行指令	需要清除错误
-3	当前处于急停拍下状态拒绝执行指令	需要松开急停并清除错误
-4	当前处于下电状态拒绝执行指令	需要上电
。。。	。。。	。。。
-10000	命令错误	不存在下发的命令
-20000	参数数量错误	下发命令中的参数数量错误
-30000	任意带名称必选参数类型格式错误	-30001表示带名称的必选参数格式错误：例如：join={1,2,3,4,5,6}，
-30001	第一个参数的参数类型错误	-3000-表示参数类型错误 最后一位1表示下发第1个参数的参数类型错误
-30002	第二个参数的参数类型错误	-3000-表示参数类型错误 最后一位2表示下发第2个参数的参数类型错误
。。。	。。。	。。。
-40000	第一个参数的参数范围错误	-4000-表示参数范围错误 最后一位1表示下发第1个参数的参数范围错误
-40002	第二个参数的参数范围错误	-4000-表示参数范围错误 最后一位2表示下发第2个参数的参数范围错误
。。。	。。。	。。。
-50001	任意带名称可选参数类型格式错误	-50001表示带名称的可选参数格式错误：例如：use=1
-50001	可选参数第一个参数的参数类型错误	-5000-表示参数类型错误 最后一位1表示下发第1个参数的参数类型错误
-50002	可选参数第二个参数的参数类型错误	-5000-表示参数类型错误 最后一位2表示下发第2个参数的参数类型错误
。。。	。。。	。。。
-60000	任意带名称可选参数的范围错误	-60001表示带名称可选参数的范围错误：例如：user=11
-60001	可选参数第一个参数的参数范围错误	-6000-表示参数范围错误 最后一位1表示下发第1个参数的参数范围错误
-60002	可选参数第二个参数的参数范围错误	-6000-表示参数范围错误 最后一位2表示下发第2个参数的参数范围错误
。。。	。。。	。。。

注：错误状态可执行指令：错误清除、错误查询、急停、机器人状态；

急停状态可执行指令：错误清除、错误查询、急停松开、机器人状态

下电状态可执行指令：错误清除、错误查询、上电、机器人状态、急停松开