

English version of the README -> please [click here](#)



工业机器人 TCP/IP 远程控制方案

文档版本: V 2.8

发布日期: 2022-02-16

深圳市越疆科技有限公司

注意事项

版权所有 © 越疆科技有限公司2021。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

免责声明

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。

越疆科技有限公司

地址：深圳市南山区留仙大道3370号南山智园崇文区2号楼9-10楼

网址：cn.dobot.cc

0.Changelog

- V3.2-2023/06/16：针对开会评审意见修改文档格式以及排版。
- V3.1-2023/06/15：针对设置指令以及运动指令添加命令注释说明。
- V3.0-2023/06/14：基于原有TCP/IP文档进行梳理重构，输出初版。

1. 综述

越疆工业机器人现支持两种远程控制方式：**远程I/O模式**、**远程Modbus模式**；具体控制方式详见《Dobot-CR-Series-Robot-APP-User-Guide文档中软件使用说明->设置->远程控制章节中；

以上两种方式主要针对**远程运行脚本的控制**；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MG400/M1Pro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；

关于TCP/IP协议的支持，MG400/M1Pro机器人的控制器版本需V1.5.5.0及以上。

2.消息格式

根据设计，MG400/M1Pro机器人会开启29999、30003、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**Dashboard端口接收到客户端约定消息格式后会将结果反馈客户端**；

30003服务器端口仅**接收客户端的约定消息格式，不做反馈**；(注意：30003端口的拆分成30003和30004端口预计在控制器3.5.2版本实现，3.5.1暂时还是一个30003端口进行实时反馈以及下发运动相关指令；)

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端口为**可配置**的反馈机器人信息端口(默认为每**200ms**反馈)；，30006端口为**可配置**的反馈机器人信息端口(默认为每**50ms**反馈)；

说明：TCP/IP远程控制指令不区分大小写格式；如ENABLEROBOT()指令或者enablerobot()指令或者eNabLErObOt()指令控制器都会按照使能的指令执行；

2.1 下发格式

消息名称(Param1,Param2,Param3.....Paramn)

消息格式如上所示，由一个消息名称，括号内由参数组成，每一个参数之间以英文逗号“,”相隔，一个完整的消息以右括号结束。

消息命令与消息应答都是 ASCII 码格式(字符串形式)。

2.2 返回格式

2.2.1 返回：

"ErrorID,{value,...,valuen},消息名称(Param1,Param2,Param3.....Paramn);"

消息格式如上所示，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六章；{value1,value2,value3,...,valuen}表示返回值，没有返回值则返回{}；消息名称(Param1,Param2,Param3.....Paramn)指下发的内容。

例：

MovL(300,30,-50,0)

返回：0,{},MovL(300,30,-50,0); //0表示接收成功 没有返回值返回{}

Mov(-500,100,200,150)

报警：-10000,{},Mov(-500,100,200,150); //-10000表示命令错误 没有返回值返回{}

3.通信协议---Dashboard端口

上位机可以通过29999端口直接发送一些设置相关的指令给机器人，这些命令被称为控制指令，这些功能被称为Dashboard。如表是Dashboard的指令列表。可以通过Dashboard的指令实现对机器人使能/下使能、复位等控制；

机器人设置相关指令如下：

指令	支持产品	描述
EnableRobot	MG\M1Pro	使能机器人

指令	支持产品	描述
DisableRobot	MG\M1Pro	下使能机器人
ClearError	MG\M1Pro	复位，用于清除错误
ResetRobot	MG\M1Pro	机器人停止当前动作，重新接收使能，规划停
SpeedFactor	MG\M1Pro	设置全局速率比
User	MG\M1Pro	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）
Tool	MG\M1Pro	选择已标定的工具坐标系
RobotMode	MG\M1Pro	机器人模式
PayLoad	MG\M1Pro	设置负载
DO	MG\M1Pro	设置数字量输出端口状态
DOExecute	MG\M1Pr	设置数字量输出端口状态（立即指令）
ToolDO	MG\M1Pr	设置末端数字量输出端口状态
ToolDOExecute	MG\M1Pr	设置末端数字量输出端口状态（立即指令）
AccJ	MG\M1Pro	设置关节加速度比例。该指令仅对MovJ、MovJIO、MovJR、JointMovJ指令有效
AccL	MG\M1Pro	设置笛卡尔加速度比例。该指令仅对MovL、MovLIO、MovLR、Jump、Arc、Circle3指令有效。
SpeedJ	MG\M1Pro	设置关节速度比例。该指令仅对MovJ、MovJIO、MovJR、JointMovJ指令有效。
SpeedL	MG\M1Pro	设置笛卡尔速度比例。该指令仅对MovL、MovLIO、MovLR、Jump、Arc、Circle3指令有效。
Arch	MG\M1Pro	设置Jump门型参数索引（起始点抬升高度、最大抬升高度、结束点下降高度）
CP	MG\M1Pro	运动时设置平滑过渡
SetArmOrientation	M1Pro	设置手系
RunScript	MG\M1Pro	运行lua脚本
StopScript	MG\M1Pro	停止lua脚本
PauseScript	MG\M1Pro	暂停lua脚本
ContinueScript	MG\M1Pro	继续lua脚本
SetPayload	MG\M1Pro	设置负载
PositiveSolution	MG\M1Pro	正解
InverseSolution	MG\M1Pro	逆解

指令	支持产品	描述
SetCollisionLevel	MG\M1Pro	设置碰撞等级
GetAngle	MG\M1Pro	获取关节坐标系下机械臂的实时位姿
GetPose	MG\M1Pro	获取笛卡尔坐标系下机械臂的实时位姿
EmergencyStop	MG\M1Pro	急停
ModbusCreate	MG\M1Pro	创建Modbus主站，并和从站建立连接
ModbusClose	MG\M1Pro	和Modbus从站断开连接
GetInBits	MG\M1Pro	读离散输入功能
GetInRegs	MG\M1Pro	读输入寄存器
GetCoils	MG\M1Pro	读线圈功能
SetCoils	MG\M1Pro	写线圈功能
GetHoldRegs	MG\M1Pro	读保存寄存器
SetHoldRegs	MG\M1Pro	写保存寄存器
GetErrorID	MG\M1Pro	获取错误ID
DI	MG\M1Pro	获取数字量输入端口状态
ToolDI	MG\M1Pro	获取末端数字量输入端口状态
DOGroup	MG\M1Pro	设置数字输出组端口状态
BrakeControl	MG\M1Pro	抱闸控制
StartDrag	MG\M1Pro	进入拖拽
StopDrag	MG\M1Pro	退出拖拽
LoadSwitch	MG\M1Pro	控制负载设置状态

3.1 EnableRobot

- 功能：使能机器人
- 格式：EnableRobot()
- 支持端口：29999
- 可选参数详解：

参数名	类型	含义
load	double	负载重量kg。MG400负载范围：0~0.5Kg；M1 Pro负载范围：0~1.5Kg；

参数名	类型	含义
centerX	double	X方向偏心距离mm，取值范围：-500mm~500mm；
centerY	double	Y方向偏心距离mm，取值范围：-500mm~500mm；
centerZ	double	Z方向偏心距离mm，取值范围：-500mm~500mm；

- 返回：
ErrorID,{},EnableRobot();
- 示例：
EnableRobot()
- 说明：**可选参数数量：0/1/4**（不填参数，正常接收ErrorID返回0；填一个参数默认为负载重量参数，ErrorID返回0；填四个参数分别表示负载重量、X方向偏心距、Y方向偏心距以及Z方向偏心距，ErrorID返回0；失败返回错误码，参考第六章；）

3.2 DisableRobot

- 功能：下使能机器人
- 格式：DisableRobot()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},DisableRobot();
- 示例：
DisableRobot()

3.3 ClearError

- 功能：清错机器人
- 格式：ClearError()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},ClearError();
- 示例：
ClearError()
- 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)；清除报警后需要重新发送EnableRobot()指令方可发送运动指令

3.4 ResetRobot

- 功能：机器人停止
- 格式：ResetRobot()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},ResetRobot();
- 示例：
ResetRobot()
- 说明：该指令用于停止运动指令，用户发送该指令后机器人会立刻停止，不再执行未完成的运动指令。

3.5 SpeedFactor

- 功能：设置全局速度比例。
- 格式：SpeedFactor(ratio)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
ratio	int	运动速度比例，取值范围：1~100

- 返回：
ErrorID,{},SpeedFactor(ratio);
- 示例：

示例	返回	说明
SpeedFactor(80)	0,{},SpeedFactor(80)	立即指令--设置全局速度比例为80

3.6 User（队列指令）

- 功能：选择已标定的用户坐标系。
- 格式：User(index)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
-----	----	----

参数名	类型	含义
index	int	选择已标定的用户坐标系，取值范围：0~9

- 返回：
ErrorID,{},User(index);
若ErrorID返回-1，表示设置的用户坐标索引不存在；
- 示例：

示例	返回	说明
User(1)	0,{},User(1)	队列指令--设置当前用户坐标系为1

3.7 Tool（队列指令）

- 功能：选择已标定的工具坐标系。
- 格式：Tool(index)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
index	int	选择已标定的工具坐标系，取值范围：0~9

- 返回：
ErrorID,{},Tool(index);
若ErrorID返回-1，表示设置的工具坐标索引不存在；
- 示例：

示例	返回	说明
Tool(1)	0,{},Tool(1)	队列指令--设置当前工具坐标系为

3.8 RobotMode

- 功能：机器人状态。
- 格式：RobotMode()
- 参数数量：0
- 支持端口：29999
- 返回值：

模式	描述	备注
1	ROBOT_MODE_INIT	初始化
2	ROBOT_MODE_BRAKE_OPEN	抱闸松开
3	ROBOT_MODE_POWER_STATUS	本体掉电
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
8	ROBOT_MODE_RECORDING	拖拽录制
9	ROBOT_MODE_ERROR	报警
10	ROBOT_MODE_PAUSE	暂停状态
11	ROBOT_MODE_JOG	点动

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

示例	返回	说明
RobotMode()	0,{5},RobotMode()	机器人处于使能状态

- 说明：为保持与控制器3.5.1版本兼容性，之前关键机器人状态返回值没有做修改；如：空闲、拖拽、运行、报警状态；新增抱闸松开、轨迹录制、暂停以及点动等；
- 其中运行状态包含：轨迹复现/拟合中、机器人运行状态以及脚本运行状态；

3.9 PayLoad（队列指令）

- 功能：设置当前的负载
- 格式：PayLoad(weight,inertia)
- 参数数量：2
- 支持端口：29999
- 参数详解：2

参数名	类型	含义
weight	double	负载重量 kg。MG400负载范围：0~0.5Kg； M1 Pro负载范围：0~1.5Kg；
inertia	double	负载惯量 kgm ²

- 返回：

ErrorID,{},PayLoad(weight,inertia);

- 示例:

示例	返回	说明
PayLoad(3,0.4)	0,{},PayLoad(3,0.4); ;	队列指令--负载设置3kg, 负载动态0.4kg

说明: 为了兼容Lua的LoadSet, tcp指令支持LoadSet, 使用LoadSet等同于调用PayLoad, 另外要和LoadSwitch指令一起使用

3.10 DO (队列指令)

- 功能: 设置数字输出端口状态 (队列指令)
- 格式: DO(index,status)
- 参数数量: 2
- 支持端口: 29999
- 参数详解: 2

参数名	类型	含义
index	int	数字输出索引, 取值范围: 1~16或100~1000
status	int	数字输出端口状态, 1: 高电平; 0: 低电平

- 返回:
ErrorID,{},DO(index,status);
- 示例:

示例	返回	说明
DO(1,1)	0,{},DO(1,1)	队列指令--设置数字输出端口1输出高电平

- 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 DOExecute

- 功能: 设置数字输出端口状态 (立即指令)
- 格式: DOExecute(index,status)
- 参数数量: 2
- 支持端口: 29999
- 参数详解: 2

参数名	类型	含义
index	int	数字输出索引, 取值范围: 1~16或100~1000

参数名	类型	含义
status	int	数字输出端口状态，1：高电平；0：低电平

- 返回：
ErrorID,{},DOExecute(index,status);

- 示例：

示例	返回	说明
DOExecute(1,1)	0,{},DOExecute(1,1)	立即指令--DO[1]输出高电平

- 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（队列指令）
- 格式：ToolDO(index,status)
- 参数数量：2
- 支持端口：29999
- 参数详解：2

参数名	类型	含义
index	int	数字输出索引，取值范围：1/2
status	int	数字输出端口状态，1：高电平；0：低电平

- 返回：
ErrorID,{},ToolDO(index,status);

- 示例：

示例	返回	说明
ToolDO(1,1)	0,{},ToolDO(1,1)	队列指令--末端DO[1]输出高电平

3.13 ToolDOExecute

- 功能：设置末端数字输出端口状态（立即指令）
- 格式：ToolDOExecute(index,status)
- 参数数量：2
- 支持端口：29999
- 参数详解：2

参数名	类型	含义
-----	----	----

参数名	类型	含义
index	int	数字输出索引，取值范围：1/2
status	int	数字输出端口状态，1：高电平；0：低电平

- 返回：
ErrorID,{},ToolDOExecute(index,status);

- 示例：

示例	返回	说明
ToolDOExecute(1,1)	0,{},ToolDOExecute(1,1)	立即指令--末端DO[1]输出高电平

3.14 AccJ（队列指令）

- 功能：设置关节加速度比例。该指令仅对MovJ、MovJIO、MovJR、JointMovJ指令有效
- 格式：AccJ(R)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
R	int	关节加速度百分比，取值范围：1~100

- 返回：
ErrorID,{},AccJ(R);

- 示例：

示例	返回	说明
AccJ(50)	0,{},AccJ(50)	队列指令--关节加速度比例设置为50

3.15 AccL（队列指令）

- 功能：设置笛卡尔加速度比例。该指令仅对MovL、MovLIO、MovLR、Jump、Arc、Circle3指令有效。
- 格式：AccL(R)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
-----	----	----

参数名	类型	含义
R	int	笛卡尔加速度比例，取值范围：1~100

- 返回：
ErrorID,{},AccL(R);
- 示例：

示例	返回	说明
AccL(50)	0,{},AccL(50)	队列指令--笛卡尔加速度比例设置为50

3.16 SpeedJ（队列指令）

- 功能：设置关节速度比例。该指令仅对MovJ、MovJIO、MovJR、JointMovJ指令有效。
- 格式：SpeedJ(R)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
R	int	关节速度比例，取值范围：1~100

- 返回：
ErrorID,{},SpeedJ(R);
- 示例：

示例	返回	说明
SpeedJ(50)	0,{},SpeedJ(50)	队列指令--关节速度比例设置为50

3.17 SpeedL（队列指令）

- 功能：设置笛卡尔速度比例。该指令仅对MovL、MovLIO、MovLR、Jump、Arc、Circle3指令有效。
- 格式：SpeedL(R)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
R	int	笛卡尔速度比例，取值范围：1~100

- 返回：
ErrorID,{},SpeedL(R);
- 示例：

示例	返回	说明
SpeedL(50)	0,{},SpeedL(50)	队列指令--笛卡尔速度比例设置为50

3.18 Arch（队列指令）

- 功能：设置Jump门型参数索引（起始点抬升高度、最大抬升高度、结束点下降高度）。
- 格式：Arch(Index,CP=R)
//其中参数CP:平滑过渡比例，取值范围：0~100 不填按照上一次设置的值。
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
Index	int	门型参数索引，取值范围：0~9

- 返回：
ErrorID,{},Arch(Index,CP=R);
- 示例：

示例	返回	说明
Arch(1)	0,{},Arch(1)	队列指令--设置Jump门型参数索引为1

3.19 CP（队列指令）

- 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。该指令对Jump指令无效。当R等于0时，表示关闭。
- 格式：CP(R)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
R	unsigned int	平滑过渡比例，取值范围：0~100

- 返回：
ErrorID,{},CP(R);

- 示例：

示例	返回	说明
CP(50)	0,{},CP(50)	队列指令--设置CP比例为50

3.20 SetArmOrientation（队列指令）

- 功能：设置手系指令。
- 格式：SetArmOrientation(LorR,UorD,ForN,Config6)
- 可选参数数量：4
- 支持端口：29999
- 可选参数详解：4

参数名	类型	含义	
LorR	int	臂方向向前/向后(1/-1) 1：向前（其中对于M1Pro产品需要设置0表示左手系） -1：向后（其中对于M1Pro产品需要设置1表示右手系）	

- 返回：
ErrorID,{},SetArmOrientation(LorR,UorD,ForN,Config6);
- M1Pro产品手势设置左手示例：
SetArmOrientation(0)
- 此条指令为M1Pro机器人特有；参数数量：1;填一个参数默认为M1Pro机型，表示左右手系；其他参数数量填写将会返回报错；

3.21 RunScript

- 功能：运行lua脚本。
- 格式：RunScript(projectName)
- 参数数量：1
- 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：

示例	返回	说明
----	----	----

示例	返回	说明
RunScript(abcd)	0,{},RunScript(abcd);	运行名称为abcd的脚本

3.22 StopScript

- 功能：停止lua脚本。
- 格式：StopScript()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},StopScript();
- 示例：

示例	返回	说明
StopScript()	0,{},StopScript();	停止当前运行脚本

3.23 PauseScript

- 功能：暂停lua脚本。
- 格式：PauseScript()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},PauseScript();
- 示例：

示例	返回	说明
PauseScript()	0,{},PauseScript();	暂停当前运行脚本

3.24 ContinueScript

- 功能：继续lua脚本。
- 格式：ContinueScript()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},ContinueScript();

- 示例：

示例	返回	说明
ContinueScript()	0,{},ContinueScript();	继续运行当前被暂停的脚本

3.25 SetPayload

- 功能：设置负载
- 格式：SetPayload(load,inertia)
- 参数数量：2
- 支持端口：29999
- 必选参数详解：1

参数名	类型	含义
load	float	load：单位kg；MG400负载范围：0~0.5Kg；M1 Pro负载范围：0~1.5Kg；

- 可选参数详解：1

参数名	类型	含义
inertia	float	inertia：设置惯量值；

- 返回：
ErrorID,{},SetPayload(load);
ErrorID,{},SetPayload(inertia);
- 示例：
SetPayload(0.3) // 设置负载

3.26 PositiveSolution

- 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：PositiveSolution(J1,J2,J3,J4,User,Tool)
- 参数数量：6
- 支持端口：29999
- 参数详解：6

参数名	类型	含义
J1	double	J1 轴位置，单位：度
J2	double	J2 轴位置，单位：度

参数名	类型	含义
J3	double	J3 轴位置，单位：度
J4	double	J4 轴位置，单位：度
User	int	选择已标定的用户坐标系
Tool	int	选择已标定的工具坐标系

- 返回：

ErrorID,{x,y,z,r},PositiveSolution(J1,J2,J3,J4,User,Tool); // {x,y,z,r}指返回的空间位置

- 示例：下发关节角度返回当前的机器人末端的空间位置

PositiveSolution(0,0,-90,0,1,1)

返回：

0,{473.000000,-141.000000,469.000000,-180.000000},PositiveSolution(0,0,-90,0,0,0);

- 说明，需要已知：
 - 机器人的臂方向SetArmOrientation

3.27 InverseSolution（队列指令）

- 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：InverseSolution(X,Y,Z,R,User,Tool,isJointNear,JointNear)
//其中isJointNear以及JointNear为可选设置参数;
- 参数数量：8
- 支持端口：29999
- 必选参数详解：6

参数名	含义	类型
X	X 轴位置，单位：毫米	double
Y	Y 轴位置，单位：毫米	double
Z	Z 轴位置，单位：毫米	double
R	R 轴位置，单位：毫米	double
User	选择已标定的用户坐标系	int
Tool	选择已标定的工具坐标系	int

- 可选参数详解：2

参数名	含义	类型
isJointNear	是否角度选解（值为1:JointNear数据有效，值为0:JointNear数据无效，算法根据当前角度进行选解；不填默认值为0）	int

参数名	含义	类型
JointNear	选解四个关节角度值	string

- 返回：
`ErrorID,{j1,j2,j3,j4},InverseSolution(X,Y,Z,R,User,Tool,isJointNear,JointNear);` //{j1,j2,j3,j4}指返回的关节值;isJointNear,JointNear若有下发则返回，没有下发则不返回;
- 示例：
下发不带选解关节角度的笛卡尔坐标值返回机器人关节角度值：
`InverseSolution(473.000000,-141.000000,469.000000,-180.000000,0,0)`
返回：`0,{0,0,-90,0},InverseSolution(473.000000,-141.000000,469.000000,0.00,0,0);`
下发带选解关节角度的笛卡尔坐标值返回机器人关节角度值：
`InverseSolution(473.0000,-141.0000,40.0000,0.000,0,0,1,{0,0,-90,0})`
返回：`0,{0,0,-90,0},InverseSolution(0,-247,1050,-90,0,0,1,{0,0,-90,0});`

3.28 SetCollisionLevel（队列指令）

- 功能：设置碰撞等级。
- 格式：`SetCollisionLevel(level)`
- 参数数量：1
- 支持端口：29999
- 参数详解：

参数名	类型	含义
level	int	level: 碰撞等级 0：关闭碰撞检测 1~5：等级越高越灵敏

- 返回：
`ErrorID,{},SetCollisionLevel(level);`
- 示例：

示例	返回	说明
<code>SetCollisionLevel(1)</code>	<code>0,{},SetCollisionLevel(1);</code>	队列指令--设置碰撞等级 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()`
- 参数数量：0
- 支持端口：29999

- 返回：
ErrorID,{j1,j2,j3,j4},GetAngle(); // {j1,j2,j3,j4}表示当前位置的关节坐标值；
- 示例：

示例	返回	说明
GetAngle()	0, {0.0,0.0,90.0,0.0},GetAngle();	立即指令--获取关节坐标系下机械臂的实时位姿

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，则获取的位姿为当前坐标系下的位姿)
- 格式：GetPose(User=0,Tool=0)
- 参数数量：0/2
- 支持端口：29999
- 返回：
ErrorID,{X,Y,Z,R},GetPose(); // {X,Y,Z,Rx,Ry,Rz}表示当前位置的笛卡尔坐标值；
- 参数详解：

参数名	类型	含义
User	int	用户坐标系索引号
Tool	int	工具坐标系索引号

- 示例：
默认参数返回上位机选择坐标系的位姿，同时传入user和tool索引值返回指定坐标系下的位姿

示例	返回	说明
GetPose()	0,{-473.0,-141.0,469.0,-180.0},GetPose();	立即指令--获取笛卡尔坐标系下机械臂的实时位姿
GetPose(User=1,Tool=0)	0, {0.0,-246.0,847.0,-90.0},GetPose(User=1,Tool=0);	立即指令--获取 用户坐标系索引号1，工具坐标系索引号0 下笛卡尔坐标系机械臂的实时位姿

示例	返回	说明

3.31 EmergencyStop

- 功能：急停
- 格式：EmergencyStop()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},EmergencyStop();
- 示例：

示例	返回	说明
EmergencyStop()	0,{},EmergencyStop();	立即指令--控制器急停 机器人下电

3.32 ModbusCreate

- 功能：创建modbus主站。
- 格式：ModbusCreate(ip,port,slave_id,isRTU)
- 参数数量：4
- 支持端口：29999
- 参数详解：

参数名	类型	含义
ip	string	从站ip地址;
port	int	从站端口;
slave_id	int	从站ID (取值范围大于0的整数)
isRTU	int	可选参数 ，取值范围0/1： 如果为空或者值为0，建立modbusTCP通信； 如果为1，建立modbusRTU通信；

- 返回：
ErrorID,{index},ModbusCreate(ip,port,slave_id,isRTU); //ErrorID:0表示创建Modbus主站成功，-1表示创建Modbus主站失败，其他值参考错误码描述；index: 返回的主站索引，最多支持5个设备,取值范围(0~4);

示例	返回	说明
----	----	----

示例	返回	说明
ModbusCreate(127.0.0.1,60000,1,1)	0, {1},ModbusCreate(127.0.0.1,60000,1,1);	立即指令--建立RTU通信主站,60000末端透传端口

3.33 ModbusClose

- 功能：和Modbus从站断开连接,释放主站。
- 格式：ModbusClose(index)
- 参数数量：1
- 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：

示例	返回	说明
ModbusClose(0)	0,{},ModbusClose(0);	立即指令--释放索引为0的主站

3.34 GetInBits

- 功能：读离散输入功能。
- 格式：GetInBits(index,addr,count)
- 参数数量：3
- 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); //table, 按位获取结果
{value1,value2...,valuen}

- 示例:

示例	返回	说明
GetInBits(0,3000,5)	0, {1,0,1,1,0},GetInBits(0,3000,5);	立即指令--从离散输入地址3000 开始 连续读取5个数据 成功
GetInBits(0,3000,5)	-1,{},GetInBits(0,3000,5);	立即指令--从离散输入地址3000 开始 连续读取5个数据 失败

3.35 GetInRegs

- 功能: 读输入寄存器。
- 格式: GetInRegs(index,addr,count,valType)
- 参数数量: 4
- 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 -4
valType	string	可选参数, 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)

- 返回:
ErrorID,{value1,value2,...,valuen},GetInRegs(index,addr,count,valType); //ErrorID为0表示正
常, 为-1表示没有获取成功;table, 按变量类型返回{value1,value2...,valuen}
- 示例:

示例	返回	说明
GetInRegs(0,4000,3)	0, {5,18,12},GetInRegs(0,4000,3);	立即指令--从地址4000开始 连 续读取3个16位无符号整数 成 功
GetInRegs(0,4000,3)	-1,{},GetInRegs(0,4000,3);	立即指令--从地址4000开始 连 续读取3个16位无符号整数 失 败

3.36 GetCoils

- 功能：读线圈功能。
- 格式：GetCoils(index,addr,count)
- 参数数量：3
- 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},GetCoils(index,addr,count); //ErrorID为0表示正常，为-1表示没有获取成功;table，按变量类型返回{value1,value2...,valuen}
- 示例：

示例	返回	说明
GetCoils(0,1000,3)	0, {1,1,0},GetCoils(0,1000,3);	立即指令--从地址1000开始连续读取3个数据 成功
GetCoils(0,1000,3)	-1,{},GetCoils(0,1000,3);	立即指令--从地址1000开始连续读取3个数据 成功

3.37 SetCoils

- 功能：写线圈功能。
- 格式：SetCoils(index,addr,count,valTab)
- 参数数量：4
- 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
valTab	string	写线圈地址值；

- 返回：
ErrorID,{},SetCoils(index,addr,count,valTab); //ErrorID为0表示正常，为-1表示没有设置成功;
- 示例：

示例	返回	说明
SetCoils(0,1000,3,{1,0,1})	0,{},SetCoils(0,1000,3,{1,0,1});	立即指令--从地址1000开始连续写3个数据 成功
SetCoils(0,1000,3,{1,0,1})	-1,{},SetCoils(0,1000,3,{1,0,1});	立即指令--从地址1000开始连续写3个数据 成功

3.38 GetHoldRegs

- 功能：读保持寄存器。
- 格式：GetHoldRegs(index,addr, count,valType)
- 参数数量：4
- 支持端口：29999
- 参数详解：

参数名	类型	含义
index	int	index,返回的主站索引，最多支持5个设备,取值范围(0~4);
addr	int	保持寄存器的起始地址。视从站配置而定；
count	int	读取指定数量type类型的数据。取值范围：1~4
valType	string	数据类型： 如果为空，默认读取16位无符号整数（2个字节，占用1个寄存器） 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）

- 返回：
ErrorID,{value1,value2,...,valuen},GetHoldRegs(index,addr, count,valType); //ErrorID为0表示正常，为-1表示没有获取成功;table, 按变量类型返回{value1,value2...,valuen}
- 示例：

示例	返回	说明
GetHoldRegs(0,3095,1)	0,{13},GetHoldRegs(0,3095,1);	立即指令--从地址3095开始读取一个16位无符号整数 成功
GetHoldRegs(0,3095,1)	-1,{},GetHoldRegs(0,3095,1);	立即指令--从地址3095开始读取一个16位无符号整数 失败

3.39 SetHoldRegs

- 功能：写保存寄存器。
- 格式：SetHoldRegs(index,addr, count,valTab,valType)
- 参数数量：5
- 支持端口：29999
- 参数详解：

参数名	类型	含义
index	int	index,返回的主站索引，最多支持5个设备,取值范围(0~4)
addr	int	保持寄存器的起始地址。视从站配置而定；
count	int	写入指定数量type类型的数据。取值范围：1~4
valTab	string	保持寄存器地址的值
valType	string	数据类型 如果为空，默认读取16位无符号整数（2个字节，占用1个寄存器） 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）

- 返回：
ErrorID,{},SetHoldRegs(index,addr, count,valTab,valType); //ErrorID为0表示正常，为-1表示没有设置成功;
- 示例：

示例	返回	说明
SetHoldRegs(0,3095,2,{6000,300}, U16)	0, {},SetHoldRegs(0,3095,2,{6000,300}, U16);	立即指令--从地址3095开始连续写两个16位无符号整数6000，300成功
SetHoldRegs(0,3095,2,{6000,300}, U16)	-1, {},SetHoldRegs(0,3095,2,{6000,300}, U16);	立即指令--从地址3095开始连续写两个16位无符号整数6000，300失败

3.40 GetErrorID

- 功能：获取机器人错误码
- 格式：GetErrorID()
- 参数数量：0
- 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，其中碰撞检测值为-2,电子皮肤碰撞检测值-3；后面六个[id]分别表示六个伺服的报警信息；

- 示例：

GetErrorID()

返回：

0,{{{[-2],[],[],[],[],[]}},GetErrorID());

示例	返回	说明
GetErrorID()	0,{}, 0,{{{[1000],[],[],[],[],[]}},GetErrorID());	立即指令--获取报警id-12288（紧急停止按键按下）

- 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；

文件位置 properties/alarm_controller.json properties/alarm_servo.json

3.41 DI

- 功能：获取数字量输入端口状态
- 格式：DI(index)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
index	int	数字输入索引，取值范围：1~32或100~1000

- 返回：

ErrorID,{value},DI(index); //value:返回当前index的值状态，取值范围0/1；

- 示例：

示例	返回	说明
DI(1)	0,{0},DI(1)	立即指令--数字输入端口1为低电平

- 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)
- 参数数量：1
- 支持端口：29999

- 参数详解：1

参数名	类型	含义
index	int	末端数字量输入索引，取值范围：1/2

- 返回：
ErrorID,{value},ToolDI(index); //value:返回当前index的值状态，取值范围0/1;
- 示例：

示例	返回	说明
ToolDI(2)	0,{1},ToolDI(2)	立即指令--末端数字输入端口2为高电平

3.43 DOGroup

- 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：不固定

参数名	类型	含义
index1	int	设置数字输出索引，取值范围：1-16或100~1000
value1	int	设置数字输出端口状态，取值0/1
...
indexn	int	设置数字输出索引，取值范围：1-16或100~1000
valuen	int	设置数字输出端口状态，取值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：

示例	返回	说明
DOGroup(4,1,6,0,2,1,7,0)	0, {},DOGroup(4,1,6,0,2,1,7,0)	立即指令--设置数字输出端口4、6、2、7端口的电平分别为1, 0, 1, 0

- 说明：使用取值范围100-1000需要有拓展IO模块的硬件支持；

3.44 BrakeControl

- 功能：开关抱闸
- 格式：BrakeControl(axisID,value)
- 必填参数数量：2
- 注意：**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；
- 支持端口：29999
- 必填参数详解：2

参数名	类型	含义
axisID	int	关节轴号
value	int	设置抱闸状态；取值0/1 0:关闭抱闸 1：打开抱闸

- 返回：
ErrorID,{},BrakeControl(axisID,value);
-
- 示例：打开关节1抱闸
BrakeControl(1,1)

示例	返回	说明
BrakeControl(1,1)	0,{},BrakeControl(1,1)	立即指令--设置关节1打开抱闸

3.45 StartDrag

- 功能：进入拖拽(在报错状态下，不可进入拖拽)
- 格式：StartDrag()
- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},StartDrag();
- 示例：

示例	返回	说明
StartDrag()	0,{},StartDrag()	立即指令--机器人进入拖拽

3.46 StopDrag

- 功能：退出拖拽
- 格式：StopDrag()

- 参数数量：0
- 支持端口：29999
- 返回：
ErrorID,{},StopDrag();
- 示例：

示例	返回	说明
StopDrag()	0,{},StopDrag()	立即指令--机器人退出拖拽

3.47 LoadSwitch（队列指令）

- 功能：控制负载设置状态
- 格式：LoadSwitch(status)
- 参数数量：1
- 支持端口：29999
- 参数详解：1

参数名	类型	含义
status	int	status：设置负载设置状态，0：关闭；1：开启，开启负载设置会提高碰撞灵敏度

- 返回：
ErrorID,{},LoadSwitch(status);
- 示例：

示例	返回	说明
LoadSwitch(1)	0,{},LoadSwitch(1)	队列指令--开启负载设置

4.通信协议----实时反馈端口

30004端口即实时反馈端口(30004、30005以及30006端口将在控制器3.5.2+版本支持)，客户端**每8ms**能收到一次机器人如下表所示的信息。30005端口为**可配置**的反馈机器人信息端口(默认为每**200ms**反馈)，30006端口为**可配置**的反馈机器人信息端口(默认为每**50ms**反馈)；通过实时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。下表是字节排列的顺序表。(30006端口的实时数据的配置更新可以在线修改后，实时生效；)

意义/Meaning	数据类型/Type	值的数目/Number of values	字节大小/Size in bytes	字节位置值/Byte position value	描述/Notes	支持产品
------------	-----------	-----------------------	--------------------	---------------------------	----------	------

意义/Meaning	数据类型/Type	值的数目/Number of values	字节大小/Size in bytes	字节位置值/Byte position value	描述/Notes	支持产品
MessageSize	unsigned short	1	2	0000 ~ 0001	消息字节总长度/Total message length in bytes	MG\M1Pro
	unsigned short	3	6	0002 ~ 0007	保留位	MG\M1Pro
DigitalInputs	uint64	1	8	0008 ~ 0015	数字输入/Current state of the digital inputs.	MG\M1Pro
DigitalOutputs	uint64	1	8	0016 ~ 0023	数字输出	MG\M1Pro
RobotMode	uint64	1	8	0024 ~ 0031	机器人模式/Robot mode	MG\M1Pro
TimeStamp	uint64	1	8	0032 ~ 0039	时间戳 (单位 ms)	CR
	uint64	1	8	0040 ~ 0047	保留位	MG\M1Pro
TestValue	uint64	1	8	0048 ~ 0055	内存结构测试标准值 0x0123 4567 89AB CDEF	MG\M1Pro
	double	1	8	0056 ~ 0063	保留位	MG\M1Pro
SpeedScaling	double	1	8	0064 ~ 0071	速度比例/Speed scaling of the trajectory limiter	MG
LinearMomentumNorm	double	1	8	0072 ~ 0079	机器人当前动量/Norm of Cartesian linear momentum(需要特定硬件版本)	CR
VMain	double	1	8	0080 ~ 0087	控制板电压/Masterboard: Main voltage	CR
VRobot	double	1	8	0088 ~ 0095	机器人电压/Masterboard: Robot voltage (48V)	CR
IRobot	double	1	8	0096 ~ 0103	机器人电流/Masterboard: Robot current	CR
	double	1	8	0104 ~ 0111	保留位	MG\M1Pro
	double	1	8	0112 ~ 0119	保留位	MG\M1Pro

意义/Meaning	数据类型/Type	值的数目/Number of values	字节大小/Size in bytes	字节位置值/Byte position value	描述/Notes	支持产品
ToolAccelerometer	double	3	24	0120 ~ 0143	TCP加速度/Tool x,y and z accelerometer values(需要特定硬件版本)	CR
ElbowPosition	double	3	24	0144 ~ 0167	肘位置/Elbow position(需要特定硬件版本)	CR
ElbowVelocity	double	3	24	0168 ~ 0191	肘速度/Elbow velocity(需要特定硬件版本)	CR
QTarget	double	6	48	0192 ~ 0239	目标关节位置/Target joint positions	MG
QDTarget	double	6	48	0240 ~ 0287	目标关节速度/Target joint velocities	MG
QDDTarget	double	6	48	0288 ~ 0335	目标关节加速度/Target joint accelerations	MG
ITarget	double	6	48	0336 ~ 0383	目标关节电流/Target joint currents	MG
MTarget	double	6	48	0384 ~ 0431	目标关节扭矩/Target joint moments (torques)	MG
QActual	double	6	48	0432 ~ 0479	实际关节位置/Actual joint positions	MG
QDActual	double	6	48	0480 ~ 0527	实际关节速度/Actual joint velocities	MG
IActual	double	6	48	0528 ~ 0575	实际关节电流/Actual joint currents	MG
ActualTCPForce	double	6	48	0576 ~ 0623	TCP传感器力值(通过六维力计算)	MG
ToolVectorActual	double	6	48	0624 ~ 0671	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	MG

意义/Meaning	数据类型/Type	值的数目/Number of values	字节大小/Size in bytes	字节位置值/Byte position value	描述/Notes	支持产品
TCPSpeedActual	double	6	48	0672 ~ 0719	TCP笛卡尔实际速度值/Actual speed of the tool given in Cartesian coordinates	MG
TCPForce	double	6	48	0720 ~ 0767	TCP力值（通过关节电流计算）	CR
ToolVectorTarget	double	6	48	0768 ~ 0815	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	MG
TCPSpeedTarget	double	6	48	0816 ~ 0863	TCP笛卡尔目标速度值/Target speed of the tool given in Cartesian coordinates	MG
MotorTemperatures	double	6	48	0864 ~ 0911	关节温度/Temperature of each joint in degrees celsius	CR
JointModes	double	6	48	0912 ~ 0959	关节控制模式/Joint control modes	CR
VActual	double	6	48	960 ~ 1007	关节电压/Actual joint voltages	CR
HandType	char	4	4	1008 ~ 1011	手系	M1Pro
User	char	1	1	1012	用户坐标	CR
Tool	char	1	1	1013	工具坐标	CR
RunQueuedCmd	char	1	1	1014	算法队列运行标志	CR
PauseCmdFlag	char	1	1	1015	算法队列暂停标志	CR
VelocityRatio	char	1	1	1016	关节速度比例 (0~100)	CR
AccelerationRatio	char	1	1	1017	关节加速度比例 (0~100)	CR
JerkRatio	char	1	1	1018	关节加加速度比例 (0~100)	CR

意义/Meaning	数据类型/Type	值的数目/Number of values	字节大小/Size in bytes	字节位置值/Byte position value	描述/Notes	支持产品
XYZVelocityRatio	char	1	1	1019	笛卡尔位置速度比例(0~100)	CR
RVelocityRatio	char	1	1	1020	笛卡尔姿态速度比例(0~100)	CR
XYZAccelerationRatio	char	1	1	1021	笛卡尔位置加速度比例(0~100)	CR
RAccelerationRatio	char	1	1	1022	笛卡尔姿态加速度比例(0~100)	CR
XYZJerkRatio	char	1	1	1023	笛卡尔位置加加速度比例(0~100)	CR
RJerkRatio	char	1	1	1024	笛卡尔姿态加加速度比例(0~100)	CR
BrakeStatus	char	1	1	1025	机器人抱闸状态	CR
EnableStatus	char	1	1	1026	机器人使能状态	CR
DragStatus	char	1	1	1027	机器人拖拽状态	CR
RunningStatus	char	1	1	1028	机器人运行状态	CR
ErrorStatus	char	1	1	1029	机器人报警状态	CR
JogStatusCR	char	1	1	1030	机器人点动状态	CR
CRRobotType	char	1	1	1031	机器类型	CR
DragButtonSignal	char	1	1	1032	按钮板拖拽信号	CR
EnableButtonSignal	char	1	1	1033	按钮板使能信号	CR
RecordButtonSignal	char	1	1	1034	按钮板录制信号	CR
ReappearButtonSignal	char	1	1	1035	按钮板复现信号	CR
JawButtonSignal	char	1	1	1036	按钮板夹爪控制信号	CR
SixForceOnline	char	1	1	1037	六维力在线状态	CR
Reserve2[82]	char	1	82	1038-1119	保留位	MG\M1Pro
MActual[6]	double	6	48	1120 ~ 1167	实际扭矩	CR
Load	double	1	8	1168-1175	负载重量kg	MG\M1Pro
CenterX	double	1	8	1176-1183	X方向偏心距离mm	MG\M1Pro
CenterY	double	1	8	1184-1191	Y方向偏心距离mm	MG\M1Pro
CenterZ	double	1	8	1192-1199	Z方向偏心距离mm	MG\M1Pro
User[6]	double	6	48	1200-1247	用户坐标值	CR
Tool[6]	double	6	48	1248-1295	工具坐标值	CR

意义/Meaning	数据类型/Type	值的数目/Number of values	字节大小/Size in bytes	字节位置值/Byte position value	描述/Notes	支持产品
TraceIndex	double	1	8	1296-1303	轨迹复现运行索引	CR
SixForceValue[6]	double	6	48	1304-1351	当前六维力数据原始值	CR
TargetQuaternion[4]	double	4	32	1352-1383	[qw,qx,qy,qz] 目标四元数	CR
ActualQuaternion[4]	double	4	32	1384-1415	[qw,qx,qy,qz] 实际四元数	CR
Reserve3[24]	char	1	24	1416 ~ 1440	保留位	MG\M1Pro
TOTAL			1440		1440byte package	

其中Robot Mode返回机器人模式：

模式	描述	备注
1	ROBOT_MODE_INIT	初始化
2	ROBOT_MODE_BRAKE_OPEN	抱闸松开
3	---	保留位
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
8	ROBOT_MODE_RECORDING	拖拽录制
9	ROBOT_MODE_ERROR	报警
10	ROBOT_MODE_PAUSE	暂停状态
11	ROBOT_MODE_JOG	点动

• 说明：

- ☐ 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；
- ☐ 机器人运行，状态为7；
- ☐ 机器人暂停，状态为10；
- ☐ 机器人进入拖拽模式(使能状态)，状态为6；
- ☐ 机器人在拖拽录制，状态为8；

- ☐ 机器人在点动，状态为11；
- ☐ 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9；
- 其中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；
 - ...
 如下位表示抱闸状态：

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		

- 其中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；
- 其中HandType包含1个char 类型的参数，M1Pro产品参数值为0为左手系，值为1表示右手系；详情参考第3.23章节的SetArmOrientation指令；
- 其中RobotType表示机器类型：

RobotType值	代表机型
1	MG400
2	M1Pro

5.通信协议----运动相关端口

上位机可以通过30003端口直接发送一些机器人运动相关指令给机器人，这些指令CR自己定义的；如下表是30003端口的指令列表。可以通过如下指令实现对机器人的运动相关控制；

其中MG400/M1Pro是四轴产品；用户使用MG400/M1Pro产品时需要填写四个坐标值的长度；

指令	支持产品	描述
MovJ	MG\M1Pro	点到点运动，目标点位为笛卡尔点位
MovL	MG\M1Pro	直线运动，目标点位为笛卡尔点位
JointMovJ	MG\M1Pro	点到点运动，目标点位为关节点位
MovLIO	MG\M1Pro	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
MovJIO	MG\M1Pro	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

指令	支持产品	描述
Arc	MG\M1Pro	圆弧运动。需结合其他运动指令完成圆弧运动
MoveJog	MG\M1Pro	点动
Sync	MG\M1Pro	阻塞程序执行队列指令
RelMovJUser	MG\M1Pro	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
RelMovLUser	MG\M1Pro	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
RelJointMovJ	MG\M1Pro	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
MovJExt	M1Pro	滑轨控制指令
SyncAll	M1Pro	阻塞之上所有程序执行队列指令
Circle	MG\M1Pro	整圆运动
wait	MG\M1Pro	队列延时一段时间
pause	MG\M1Pro	暂停非脚本运动指令
continue	MG\M1Pro	继续暂停非脚本运动指令

5.1 MovJ（队列指令）

- 功能：点到点运动，目标点位为笛卡尔点位。
- 格式：MovJ(X,Y,Z,R,User=index,Tool=index,SpeedJ=R,AccJ=R,CP=R)

//其中User=index,Tool=index,SpeedJ=R,AccJ=R为可选设置参数，分别表示设置用户坐标系、工具坐标系、关节速度比例以及加速度比例值；和29999端口设置的SpeedJ、AccJ意义一致；User:用户索引0~9,不填按照上一次设置的值；Tool:工具索引0~9,不填按照上一次设置的值；CP:平滑过渡比例，取值范围：0~100 不填按照上一次设置的值。

- 必填参数数量：4
- 支持端口：30003
- 必填参数详解：4

参数名	类型	含义
X	double	X 轴位置，单位：毫米
Y	double	Y 轴位置，单位：毫米
Z	double	Z 轴位置，单位：毫米
R	double	R 轴位置，单位：毫米

- 返回：
ErrorID,{},MovJ(X,Y,Z,R);
- 示例：

MovJ(-500,100,200,150,AccJ=50,CP=1)

返回：

示例	返回	说明
MovJ(-500,100,0,90,AccJ=50)	0, {},MovJ(-500,100,0,90,AccJ=50)	队列指令--点到点运动，目标点位为笛卡尔点位 (-500,100,200,0,90)，设置关节加速度百分比50

5.2 MovL（队列指令）

- 功能：直线运动，目标点位为笛卡尔点位。
- 格式：MovL(X,Y,Z,R,User=index,Tool=index,SpeedL=R,AccL=R,CP=R)

其中User=index,Tool=index,SpeedL=R,AccL=R为可选设置参数，分别表示设置用户坐标系、工具坐标系、笛卡尔速度比例以及加速度比例值；和29999端口设置的SpeedL、AccL意义一致；User: 用户索引0~9,不填按照上一次设置的值；Tool:工具索引0~9,不填按照上一次设置的值；CP:平滑过渡比例，取值范围：0~100 不填按照上一次设置的值。

- 必填参数数量：4
- 支持端口：30003
- 必填参数详解：4

参数名	类型	含义
X	double	X 轴位置，单位：毫米
Y	double	Y 轴位置，单位：毫米
Z	double	Z 轴位置，单位：毫米
R	double	R 轴位置，单位：毫米

- 返回：
ErrorID,{},MovL(X,Y,Z,R,SpeedL=R,AccL=R,CP=R);

- 示例：

示例	返回	说明
MovL(-500,100,0,90,SpeedL=60)	0, {},MovL(-500,100,0,90,SpeedL=60)	队列指令--点到点运动，目标点位为笛卡尔点位 (-500,100,0,90)，设置笛卡尔速度比例60

返回：

ErrorID,{},MovL(-500,100,200,150,SpeedL=60,CP=1);

5.3 JointMovJ（队列指令）

- 功能：点到点运动，目标点位为关节点位。
- 格式：JointMov(J1,J2,J3,J4,SpeedJ=R,AccJ=R,CP=R)

//其中SpeedJ=R,AccJ=R为可选设置参数，分别表示设置关节速度比例以及加速度比例值； 和29999端口设置的SpeedJ、AccJ意义一致；CP:平滑过渡比例，取值范围：0~100 不填按照上一次设置的值。

- 必填参数数量：4
- 支持端口：30003
- 必填参数详解：4

参数名	类型	含义
J1	double	J1 轴位置，单位：度
J2	double	J2 轴位置，单位：度
J3	double	J3 轴位置，单位：度
J4	double	J4 轴位置，单位：度

- 返回：
ErrorID,{},JointMov(J1,J2,J3,J4,SpeedJ=R,AccJ=R,CP=R);

- 示例：

示例	返回	说明
JointMov(0,0,90,0,SpeedJ=60,AccJ=50)	0, {},JointMov(0,0,90,0,SpeedJ=60,AccJ=50)	队列指令--点到点运动，目标点位为关节点位（0,0,90,0），设置关节速度比例60，加速度比例50

返回：
ErrorID,{},JointMov(0,0,-90,0,SpeedJ=60,AccJ=50,CP=1);

5.4 MovLIO（队列指令）

- 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(X,Y,Z,R,{Mode,Distance,Index,Status},...,{Mode,Distance,Index,Status},User=index,Tool=index,SpeedL=R,AccL=R,CP=R)

//其中SpeedL=R,AccL=R为可选设置参数，分别表示设置用户坐标系、工具坐标系、笛卡尔速度比例以及加速度比例值； 和29999端口设置的SpeedL、AccL意义一致；User:用户索引0~9,不填按照上一次设置的值；Tool:工具索引0~9,不填按照上一次设置的值；

- 参数数量：8
- 支持端口：30003
- 参数详解：8

参数名	类型	含义
X	double	X 轴位置，单位：毫米

参数名	类型	含义
Y	double	Y 轴位置，单位：毫米
Z	double	Z 轴位置，单位：毫米
R	double	R 轴位置，单位：毫米
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
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，取值范围：0~100；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
Index	int	数字输出索引，取值范围：1~24
Status	int	数字输出状态，取值范围：0或1

- 返回：
ErrorID,{},MovLIO(X,Y,Z,R,{Mode,Distance,Index,Status},...,
{Mode,Distance,Index,Status},SpeedL=R,AccL=R,CP=R);
- 示例：

示例	返回	说明
MovLIO(-500,100,0,90, {0,50,1,0})	0, {},MovLIO(-500,100,0,90, {0,50,1,0});	队列指令--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（-500,100,0,90），离起始点距离50，DO[1]设置为低电平

5.5 MovJIO（队列指令）

- 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(X,Y,Z,R,{Mode,Distance,Index,Status},...,
{Mode,Distance,Index,Status},User=index,Tool=index,SpeedJ=R,AccJ=R,CP=R)
//其中SpeedJ=R,AccJ=R为可选设置参数，分别表示设置用户坐标系、工具坐标系、关节速度比例以及加速度比例值；和29999端口设置的SpeedJ、AccJ意义一致；User:用户索引|0~9,不填按照上一次设置的值；Tool:工具索引|0~9,不填按照上一次设置的值；
- 参数数量：不固定
- 支持端口：30003
- 参数详解：8

参数名	类型	含义
X	double	X 轴位置，单位：毫米
Y	double	Y 轴位置，单位：毫米

参数名	类型	含义
Z	double	Z 轴位置，单位：毫米
R	double	R 轴位置，单位：毫米
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
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，取值范围：0~100；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
Index	int	数字输出索引，取值范围：1~24
Status	int	数字输出状态，取值范围：0或1

- 返回：

ErrorID,{},MovJIO(X,Y,Z,R,{Mode,Distance,Index,Status},...,
{Mode,Distance,Index,Status},SpeedJ=R,AccJ=R,CP=R);

- 示例：

示例	返回	说明
MovJIO(-500,100,0,90, {0,50,1,0})	0, {},MovLIO(-500,100,0,90, {0,50,1,0});	队列指令--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（-500,100,0,90），离起始点距离50，DO[1]设置为低电平

5.6 Arc（队列指令）

- 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(X1,Y1,Z1,R1,X2,Y2,Z2,R2,User=index,Tool=index,SpeedL=R,AccL=R,CP=R)

//其中SpeedL=R,AccL=R为可选设置参数，分别表示设置用户坐标系、工具坐标系、笛卡尔速度比例以及加速度比例值；和29999端口设置的SpeedL、AccL意义一致；User:用户索引0~9,不填按照上一次设置的值；Tool:工具索引0~9,不填按照上一次设置的值；CP:平滑过渡比例，取值范围：0~100 不填按照上一次设置的值。

- 必填参数数量：8
- 支持端口：30003
- 必填参数详解：8

参数名	类型	含义
X1	double	表示圆弧中间点X1 轴位置，单位：毫米
Y1	double	表示圆弧中间点Y1 轴位置，单位：毫米
Z1	double	表示圆弧中间点Z1 轴位置，单位：毫米

参数名	类型	含义
R1	double	表示圆弧中间点R1轴位置，单位：毫米
X2	double	表示圆弧结束点X2 轴位置，单位：毫米
Y2	double	表示圆弧结束点Y2 轴位置，单位：毫米
Z2	double	表示圆弧结束点Z2 轴位置，单位：毫米
R2	double	表示圆弧结束点R2 轴位置，单位：毫米

- 返回：
ErrorID,{},Arc(X1,Y1,Z1,R1,X2,Y2,Z2,R2,SpeedL=R,AccL=R,CP=R);
- 示例：
MovL(-300,-150,0,90,SpeedL=100,AccL=100,CP=1)
Arc(-350,-200,0,90,-300,-250,0,90)
队列指令--确定圆弧起始点。
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 |

5.7 MoveJog（队列指令）

- 功能：点动运动，不固定距离运动
- 格式：MoveJog(axisID,CoordType=typeValue,User=index,Tool=index)
//其中CoordType、User以及Tool为可选设置参数，不填按照默认值；CoordType: 0:用户坐标系
1:关节点动 2:工具坐标系，默认值为1； User:用户索引|0~9,默认值为0； Tool:工具索引|0~9,默认值
为0；
- 必填参数数量：1
- 支持端口：30003
- 注意：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；完全停止需要大概消耗100ms时间；
- 必填参数详解：1

参数名	类型	含义
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 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 Z+ 表示Z轴正方向运动 R+ 表示R轴正方向运动 R- 表示Rx轴负方向运动

- 返回：

ErrorID,{},MoveJog(axisID,CoordType=typeValue,User=index,Tool=index);

若ErrorID返回-1，表示设置的用户坐标索引或工具坐标索引不存在；

- 示例：J2负方向运动，再停止点动

MoveJog(j2-)

返回：

0,{},MoveJog(j2-);

MoveJog()

返回：

0,{},MoveJog();

说明：**MG400/M1Pro控制器1.5.6版本及以上版本支持此命令**；其中用户若是再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；

5.8 Sync（队列指令）

- 功能：阻塞程序执行队列指令，待所有队列指令执行完才返回。

- 格式：Sync()

- 参数数量：0

- 支持端口：30003

- 返回：

ErrorID,{},Sync();

- 示例：

Sync()

5.9 RelMovJUser（队列指令）

- 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(OffsetX,OffsetY,OffsetZ,OffsetR, User,SpeedJ=R, AccJ=R,Tool=Index,CP=R)

//其中SpeedJ=R,AccJ=R,Tool=Index为可选设置参数，分别表示设置用关节速度比例以及加速度比例值以及工具坐标索引；和29999端口设置的SpeedJ、AccJ意义一致；Tool:工具索引0~9,不填按照上一次设置的值；User为必选参数:用户索引0~9；参数CP:平滑过渡比例，取值范围：0~100不填按照上一次设置的值。

- 必填参数数量：5

- 支持端口：30003

- 必填参数详解：5

参数名	类型	含义
OffsetX	double	X轴方向偏移，单位：mm
OffsetY	double	Y轴方向偏移，单位：mm

参数名	类型	含义
OffsetZ	double	Z轴方向偏移，单位：mm
OffsetR	double	R轴方向偏移，单位：mm

| User | int | 选择已标定的用户坐标系，取值范围：0~9 |

• 返回：

ErrorID,{},RelMovJUser(OffsetX,OffsetY,OffsetZ,OffsetR,User,SpeedJ=R,AccJ=R,Tool=Index,CP=R);

• 示例：

示例	返回	说明
RelMovJUser(10,10,0,0,0)	0, {},RelMovJUser(10,10,0,0,0);	队列指令--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 X轴方向偏移10mm Y轴方向偏移10mm Y轴方向偏移10mm Z轴，R轴 用户坐标系为0；

• 说明：MG400/M1Pro控制器1.5.6版本及以上版本支持此命令；

5.10 RelMovLUser（队列指令）

• 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(OffsetX,OffsetY,OffsetZ,OffsetR, User,SpeedL=R, AccL=R,Tool=Index,CP=R)

//其中SpeedL=R,AccL=R,Tool=Index为可选设置参数，分别表示设置用笛卡尔速度比例以及加速度比例值以及工具坐标系索引；和29999端口设置的SpeedL、AccL意义一致；Tool:工具索引0~9,不填按照上一次设置的值；User为必选参数:用户索引0~9；参数CP:平滑过渡比例，取值范围：0~100 不填按照上一次设置的值。

• 必填参数数量：5

• 支持端口：30003

• 必填参数详解：5

参数名	类型	含义
OffsetX	double	X轴方向偏移，单位：mm
OffsetY	double	Y轴方向偏移，单位：mm
OffsetZ	double	Z轴方向偏移，单位：mm
OffsetR	double	R轴方向偏移，单位：mm

| User | int | 选择已标定的用户坐标系，取值范围：0~9 |

- 返回：
ErrorID,{},RelMovLUser(OffsetX,OffsetY,OffsetZ,OffsetR,User,SpeedL=R, AccL=R,Tool=Index,CP=R);
- 示例：

示例	返回	说明
RelMovLUser(10,10,10,0,0,0)	0, {},RelMovLUser(10,10,10,0,0);	队列指令--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 X轴方向偏移10mm Y轴方向偏移10mm Z轴方向偏移10mm R轴无变化 用户坐标系为0；

- 说明：MG400/M1Pro控制器1.5.6版本及以上版本支持此命令；

5.11 RelJointMovJ（队列指令）

- 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,SpeedJ=R, AccJ=R,CP=R)
//其中SpeedJ=R,AccJ=R为可选设置参数，分别表示设置用关节速度比例以及加速度比例值；和29999端口设置的SpeedJ、AccJ意义一致；参数CP:平滑过渡比例，取值范围：0~100 不填按照上一次设置的值。
- 必填参数数量：4
- 支持端口：30003
- 必填参数详解：4

参数名	类型	含义
Offset1	double	关节1的偏移值，单位：度
Offset2	double	关节2的偏移值，单位：度
Offset3	double	关节3的偏移值，单位：度
Offset4	double	关节4的偏移值，单位：度

- 返回：
ErrorID,{},RelJointMovJ(Offset1,Offset2,Offset3,Offset4,SpeedJ=R, AccJ=R,CP=R);
- 示例：

示例	返回	说明
----	----	----

示例	返回	说明
RelJointMovJ(10,10,10,0)	0, {},RelJointMovJ(10,10,10,0);	队列指令--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 关节1的偏移值10度 关节2的偏移值10度 关节3的偏移值10度 R轴无变化；

- 说明：MG400/M1Pro控制器1.5.6版本及以上版本支持此命令；

5.12 MovJExt

- 功能：滑轨运动控制指令
- 格式：MovJExt(角度/距离,SpeedE=50,AccE=50,Sync=1)
- 参数数量：4
- 支持端口：30003
- 必选参数详解：1

参数名	类型	含义
角度/距离	float	要移动的位置值（角度/距离）

- 可选参数详解：3

参数名	类型	含义
SpeedE	int	SpeedE：设置本指令运行速度；
AccE	int	AccE：设置本指令运行加速度；
Sync	int	设置指令运行方式同步/异步；Sync=0 异步运行，发送指令不必等运行结果立即返回；Sync=1 同步运行，发送指令会等运行结果才返回；

- 返回：
ErrorID,{},MovJExt();
- 示例：
MovJExt(300) // 移动到300位置
- 说明：此条指令为MG\M1Pro系列机器人特有；

5.13 SyncAll

- 功能：阻塞本指令之前的程序执行队列指令，待之前所有队列指令执行完才返回。
- 格式：SyncAll()

- 参数数量：0
- 支持端口：30003
- 返回：
ErrorID,{},SyncAll();
- 示例：
SyncAll()
- 说明：此条指令为MG\M1Pro系列机器人特有；

5.14 Circle（队列指令）

- 功能：整圆运动，仅对笛卡尔点位生效。
- 格式：circle({p1.x,p1.y,p1.z,p1.r},{p2.x,p2.y,p2.z,p2.r},count,User=0,Tool=0,SpeedL=R,AccL=R)
- 参数数量：必选3
- 支持端口：30003
- 参数详解：

参数名	类型	含义
P1	table	第一个点
P2	table	第二个点
count	int	大于等于1的整数
User	int	用户坐标系索引号
Tool	int	工具坐标系索引号

- 返回：
ErrorID,{},circle(输入);
- 示例：

示 例	circle({322.3267,-379.0799,545.6118,-171.5755},{-153.785,-473.2296,545.6118,-171.5755},1)
返 回	0,{},circle({322.3267,-379.0799,545.6118,-171.5755},{-153.785,-473.2296,545.6118,-171.5755},1);
说 明	队列指令--整圆运动，进行整圆运动一圈。

5.15 wait

- 功能：队列延时一段时间。
- 格式：wait(time)

- 参数数量：1
- 支持端口：30003

参数名	类型	含义
time	int	延时的时间，单位是ms。范围是(0,3600*1000)

- 返回：
ErrorID,{},wait(time);
- 示例：

示例	返回	说明
MovJ(-500,100,200,150) Wait(1000)	MovJ(-500,100,200,150); 0,{},Wait(1000);	队列指令--机械臂MovJ到位后，延时1000毫秒

5.16 pause

- 功能：暂停非脚本运动指令,不清空运动队列。
- 格式：pause()
- 参数数量：0
- 支持端口：30003

参数名
无参

- 返回：
ErrorID,{},pause();
- 示例：
pause()

5.17 continue

- 功能：与pause指令对应，继续运行pause暂停的非脚本运动指令。或者用于发生碰撞、报警后导致的机器人停止后，重新恢复接收运动指令并运行。
- 格式：continue()
- 参数数量：0
- 支持端口：30003

参数名
无参

- 返回：

ErrorID,{},continue());

- 示例:

continue()

6.错误码描述

错误码	描述	备注
0	无错误	下发成功
-1	没有获取成功	命令接收失败/建立失败
...
-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
-30001	第一个参数的参数类型错误	-30000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	-30000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
...
-40001	第一个参数的参数范围错误	-40000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	-40000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
...