



## 睿尔曼机械臂 JSON 通信协议 V3.6.0



---

睿尔曼智能科技（北京）有限公司



## 文件修订记录：

版本号	时间	备注
V1.0	2020-05-01	拟制
V1.1	2020-05-10	修订
V1.2	2020-05-15	修订（通用化修订）
V1.3	2020-05-17	简化部分协议返回值
V1.4	2020-05-20	修改其中部分标点符号
V1.4.1	2020-05-25	修改部分格式
V1.4.2	2020-06-05	修改 WIFI 配置流程
V1.4.3	2020-06-18	修改 arm_all_state 返回帧
V1.4.4	2020-06-29	修改通信错误提示
V1.4.5	2020-06-29	添加 IO 部分协议
V1.4.6	2020-07-03	修改部分指令名称
V1.4.7	2020-07-28	加入拖动示教部分
V1.4.8	2020-08-02	加入末端接口部分
V1.5	2021-03-12	加入末端质心、路径点缓存等功能
V1.6	2021-05-20	加入 Movej_P 指令、PWM 设置和一维力设置
V1.7	2021-07-26	增加动力学标定参数下载指令
V1.8	2021-08-18	增加控制器和末端接口板的 Modbus 协议配置
V1.9	2021-09-26	可显示多个内核的软件版本
V2.0	2021-09-27	加入防碰撞等级设置
V2.1	2021-11-20	加入关节标定设置
V2.2	2022-01-12	加入在线编程设置
V2.3	2022-01-28	加入一维力设置
V2.4	2022-02-12	明确 RS485 使用介绍
V2.5	2022-02-15	加入一键设置关节限位
V2.6	2022-02-19	勘误
V2.7	2022-03-07	加入透传力位混合补偿，修改角度、位姿透传
V2.8	2022-04-08	加入高速网口的控制
V2.9	2022-04-14	修改升降机构返回状态
V3.0	2022-05-06	屏蔽测试指令
V3.1	2022-05-13	文档优化
V3.2	2022-7-1	Modbus 协议部分增加多圈和多寄存器操作
V3.3	2022-9-23	修改力位混合控制指令、安装方式参数、位姿透传返回值等
V3.4	2022-12-29	删除系统状态自动回传指令



		添加 Modbus 读多个线圈和读多个保存寄存器指令 修改手爪协议格式，添加阻塞模式 删除机械臂关节标定
3.5.0	2023-02-15	增加 I 系列新增协议支持 标注 I 系列与基础系列接口区别 工作坐标与工具坐标系描述错误修正 部分内容勘误 增加基础系列机械臂高速网口 IP 配置与查询接口
3.5.1	2023-03-16	添加必要的说明 增加基础系列高速网口配置示例
3.5.2	2023-05-25	查询六维力数据接口的返回值增加系统外力数据反馈 查询力位混合透传补偿接口返回值增加六维力各个方向上的力 Modbus 协议优化 到位状态协议优化，区分不同的到位状态 新增清除系统错误接口，用于手动清除系统错误 关节错误码反馈接口增加抱闸状态 关节错误码修正 TCP 默认参数配置更新 部分文本勘误
3.5.3	2023-06-07	位置步进举例勘误
3.5.4	2023-07-18	部分协议内容勘误 I 系列机械臂增加 UDP 主动上报端口 系统错误码补充说明
3.5.5	2023-08-19	新增 I 系列机器人在线编程存储相关接口 新增通用扩展轴相关扩展接口 新增主动上报功能配置查询接口 其他内容勘误
3.5.6	2023-09-08	优化 UDP 主动上报接口和力数据查询接口 增加力传感器在不同坐标系下的数据 增加 UDP 主动上报目标地址逻辑和配置参数
3.5.7	2023-10-11	增加机械臂缓停接口 增加查询控制器 RS485 模式和查询工具端 RS485 模式接口



		优化无线网卡网络信息查询接口 增加关闭 wifi 功能接口 增加查询指定坐标系返回失败的协议
3.5.8	2023-11-21	新增电子围栏相关接口 增加 modbus-TCP 主站和 modbus-RTU 从站接口 新增设置示教参考坐标系与查询示教参考坐标系接口 新增自碰撞安全检测相关接口 系统错误码补充说明 内容勘误
3.5.9	2023-12-15	新增读取软件信息接口 新增关节驱动器转速、加速度以及最大最小限位设置查询接口 扩充优化开始复合模式拖动示教、movej、movei、movec、movej_p、一键设置关节限位等接口 新增查询夹爪信息接口 错误码增加超软限位及末端接口板相关错误
3.6.0	2023-02-01	新增设置六维力拖动示教阈值参数 新增查询六维力拖动示教阈值参数 新增初始化六维力拖动示教阈值参数



# 目录

一、 关节配置 .....	20
1.1 关节配置命令集 .....	20
(1) 设置关节最大转速 .....	21
(2) 设置关节最大加速度 .....	21
(3) 设置关节最小限位 .....	22
(4) 设置关节最大限位 .....	22
(5) 设置关节最大转速（驱动器） .....	22
(6) 设置关节最大加速度（驱动器） .....	23
(7) 设置关节最小限位（驱动器） .....	23
(8) 设置关节最大限位（驱动器） .....	24
(9) 设置关节使能状态 .....	24
(10) 设置关节零位 .....	24
(11) 清除关节错误代码 .....	25
(12) 一键设置关节限位 .....	25
1.2 关节配置查询集 .....	25
(1) 查询关节最大速度 .....	25
(2) 查询关节最大加速度 .....	26



(3) 查询关节最小限位 .....	26
(4) 查询关节最大限位 .....	26
(5) 查询关节最大速度（驱动器） .....	27
(6) 查询关节最大加速度（驱动器） .....	27
(7) 查询关节最小限位（驱动器） .....	27
(8) 查询关节最大限位（驱动器） .....	27
(9) 查询关节使能状态 .....	28
(10) 查询关节错误代码 .....	28
1.3 关节配置反馈集 .....	28
(1) 反馈关节最大速度 .....	28
(2) 反馈关节最大加速度 .....	29
(3) 反馈关节最小限位 .....	29
(4) 反馈关节最大限位 .....	30
(5) 反馈关节最大速度（驱动器） .....	30
(6) 反馈关节最大加速度（驱动器） .....	30
(7) 反馈关节最小限位（驱动器） .....	31
(8) 反馈关节最大限位（驱动器） .....	31
(9) 反馈关节使能状态 .....	32
(10) 反馈关节错误代码 .....	32



二、机械臂配置 .....	33
2.1 机械臂配置-运动参数命令集 .....	33
(1) 设置机械臂末端最大线速度 .....	33
(2) 设置机械臂末端最大线加速度 .....	33
(3) 设置机械臂末端最大角速度 .....	34
(4) 设置机械臂末端最大角加速度 .....	34
(5) 初始化机械臂参数 .....	35
(6) 设置碰撞防护等级 .....	35
(7) 查询碰撞防护等级 .....	35
(8) 重新设置机械臂 DH 参数 .....	36
(9) 查询机械臂 DH 参数 .....	36
(10) 重新设置关节零位补偿角度 .....	37
2.2 机械臂配置-运动参数查询集 .....	37
(1) 查询机械臂末端最大线速度 .....	37
(2) 查询机械臂末端最大线加速度 .....	37
(3) 查询机械臂末端最大角速度 .....	38
(4) 查询机械臂末端最大角加速度 .....	38
2.3 机械臂配置-运动参数反馈集 .....	38
(1) 反馈机械臂末端最大线速度 .....	38



(2) 反馈机械臂末端最大线加速度 .....	39
(3) 反馈机械臂末端最大角速度 .....	39
(4) 反馈机械臂末端最大角加速度 .....	39
2.4 机械臂配置-工具坐标系命令集 .....	39
(1) 自动计算工具坐标系(标定参考点) .....	40
(2) 自动计算工具坐标系(自动计算生成工具) .....	40
(3) 手动输入工具坐标系 .....	41
(4) 切换当前工具坐标系 .....	41
(5) 删除工具坐标系 .....	42
2.5 机械臂配置-工作坐标系命令集 .....	42
(1) 自动设置工作坐标系 .....	42
(2) 手动输入工作坐标系 .....	43
(3) 切换当前工作坐标系 .....	43
(4) 删除工作坐标系 .....	44
2.6 机械臂配置-坐标系查询集 .....	44
(1) 查询当前工具 .....	44
(2) 查询已有所有工具名称 .....	44
(3) 查询指定工具信息 .....	45
(4) 查询当前工作坐标系 .....	45





(5) 查询已有所有工作坐标系名称 .....	45
(6) 查询指定工作坐标系 .....	45
2.7 机械臂配置-坐标系返回集 .....	46
(1) 返回当前工具信息 .....	46
(2) 返回所有工具名称 .....	46
(3) 返回指定工具信息 .....	47
(4) 返回当前工作坐标系信息 .....	47
(5) 返回所有工作坐标系名称 .....	47
(6) 返回指定坐标系信息 .....	48
2.8 机械臂配置-状态查询集 .....	48
(1) 查询机械臂状态 .....	48
(2) 查询关节温度 .....	49
(3) 查询关节当前电流 .....	49
(4) 查询关节当前电压 .....	49
2.9 机械臂配置-状态反馈帧 .....	49
(1) 反馈机械臂状态 .....	49
(2) 反馈关节温度 .....	50
(3) 反馈关节当前电流 .....	51
(4) 反馈关节当前电压 .....	51



2.10 机械臂配置-初始状态 .....	52
(1) 设置初始状态 .....	52
(2) 查询初始位置 .....	52
(3) 反馈初始位置 .....	53
三、 运动配置 .....	53
3.1 运动配置-轨迹指令类 .....	53
(1) MoveJ：关节运动 .....	53
(2) MoveL：直线运动 .....	54
(3) MoveC：圆弧运动 .....	55
(4) 角度透传 .....	56
(5) 位姿透传 .....	57
(6) MoveJ_P：关节空间规划到目标位姿 .....	58
3.2 运动配置-步进指令类 .....	59
(1) 关节步进 .....	59
(2) 位置步进 .....	60
(3) 姿态步进 .....	60
3.3 运动配置-运动指令类 .....	61
(1) 轨迹急停 .....	61
(2) 轨迹缓停 .....	61



(3) 轨迹暂停 .....	61
(4) 轨迹暂停后恢复 .....	62
(5) 清除当前轨迹 .....	62
(6) 清除所有轨迹 .....	62
(7) 查询当前规划类型 .....	63
3.4 运动配置-示教指令类 .....	63
(1) 关节示教 .....	63
(2) 位置示教 .....	63
(3) 姿态示教 .....	64
(4) 示教停止 .....	64
(5) 设置示教参考坐标系 .....	64
(6) 获取示教参考坐标系 .....	65
3.5 运动配置-轨迹反馈集 .....	65
(1) 返回当前正在运行的轨迹 .....	65
(2) 当前轨迹结束返回标志 .....	66
四、 系统配置 .....	67
4.1 系统配置-系统查询 .....	67
(1) 查询控制器状态 .....	67
4.2 系统配置-系统反馈 .....	67



(1) 反馈控制器状态 .....	67
4.3 系统配置-系统指令 .....	67
(1) 控制机械臂上电、断电 .....	67
(2) 读取机械臂电源状态 .....	68
(3) 读取软件版本号 .....	68
(4) 读取控制器的累计运行时间 .....	68
(5) 清零控制器的累计运行时间 .....	69
(6) 读取关节的累计转动角度 .....	69
(7) 清零关节的累计转动角度 .....	70
(8) 高速网口的配置（基础系列） .....	70
(9) 参数保存（基础系列） .....	71
(10) 配置有线网 IP 地址（I 系列） .....	71
(11) 清除系统错误 .....	71
(12) 读取软件信息 .....	72
4.4 系统配置-配置通讯内容 .....	72
(1) 配置 wifiAP 内容 .....	73
(2) 配置 wifiSTA 内容 .....	73
(3) 配置 USB 内容（基础系列） .....	73
(4) 配置 RS485 内容 .....	74



(5) 配置高速网口 IP (基础系列) .....	74
(6) 查询高速网口 IP (基础系列) .....	74
(7) 查询有线网卡网络信息 (I 系列) .....	75
(8) 查询无线网卡网络信息 (I 系列) .....	75
(9) 恢复网络设置 (I 系列) .....	76
(10) 关闭 wifi 功能 (I 系列) .....	76
4.5 查询机械臂状态信息 .....	77
(1) 查询机械臂关节角度 .....	77
(2) 反馈机械臂关节角度 .....	77
(3) 一次性查询机械臂所有状态信息 .....	77
(4) 反馈所有状态信息 .....	78
(5) 查询规划计数 .....	78
(6) 反馈机械臂轨迹规划计数 .....	79
(7) 查询控制器 RS485 模式 (I 系列) .....	79
(8) 查询工具端 RS485 模式 (I 系列) .....	79
4.6 控制器 IO 配置及获取 .....	80
(1) 设置数字 IO 模式 (I 系列) .....	80
(2) 设置数字 IO 输出状态 .....	81
(3) 获取数字 IO 状态 (I 系列) .....	81



(4) 获取数字 IO 输出状态（基础系列） .....	82
(5) 获取数字 IO 输入状态（基础系列） .....	82
(6) 设置模拟 IO 输出状态（基础系列） .....	82
(7) 获取模拟 IO 输出状态（基础系列） .....	83
(8) 获取模拟 IO 输入状态（基础系列） .....	83
(9) 获取所有 IO 输入状态 .....	83
(10) 获取所有 IO 输出状态 .....	84
(11) 设置电源输出（I 系列） .....	84
(12) 获取电源输出（I 系列） .....	84
4.7 末端工具 IO 控制 .....	85
(1) 设置工具端数字 IO 输出状态 .....	85
(2) 设置工具端数字 IO 模式 .....	85
(3) 获取工具端数字 IO 状态 .....	86
(4) 设置工具端电源输出 .....	86
(5) 获取工具端电源输出 .....	86
4.8 末端工具—手爪控制（选配） .....	87
(1) 设置手爪行程 .....	87
(2) 松开手爪 .....	87
(3) 手爪力控夹取 .....	88



(4) 手爪持续力控夹取 .....	88
(5) 手爪到达指定位置 .....	89
(6) 查询夹爪状态 .....	89
4.9 末端工具—六维力（选配） .....	90
(1) 查询六维力数据 .....	91
(2) 六维力数据清零 .....	91
(3) 自动设置六维力重心参数 .....	92
(4) 手动标定六维力数据 .....	92
(5) 停止标定力传感器重心 .....	93
(6) 设置六维力拖动示教阈值参数 .....	94
(7) 查询六维力拖动示教阈值参数 .....	94
(8) 初始化六维力拖动示教阈值参数 .....	95
4.10 末端传感器—一维力（选配） .....	95
(1) 查询末端一维力数据 .....	96
(2) 清零末端一维力数据 .....	97
(3) 自动标定一维力数据 .....	97
(4) 手动标定一维力数据 .....	97
(5) 停止标定力传感器重心 .....	98
4.11 拖动示教 .....	98



(1) 拖动示教开始 .....	98
(2) 拖动示教结束 .....	99
(3) 开始复合模式拖动示教 .....	99
(4) 轨迹复现开始 .....	100
(5) 轨迹复现暂停 .....	100
(6) 轨迹复现继续 .....	100
(7) 轨迹复现停止 .....	101
(8) 运动到轨迹起点 .....	101
(9) 力位混合控制 .....	101
(10) 结束力位混合控制 .....	102
4.12 五指灵巧手（选配） .....	102
(1) 设置灵巧手手势 .....	102
(2) 设置灵巧手动作序列 .....	103
(3) 设置灵巧手各自由度角度 .....	103
(4) 设置灵巧手速度 .....	103
(5) 设置灵巧手力阈值 .....	104
4.13 Modbus 配置 .....	104
(1) 配置通讯端口 ModbusRTU 模式 .....	105
(2) 关闭通讯端口 ModbusRTU 模式 .....	105





(3) 配置连接 ModbusTCP 从站 (I 系列) .....	106
(4) 配置关闭 ModbusTCP 从站 (I 系列) .....	106
(5) 读线圈 .....	106
(6) 读离散量输入 .....	107
(7) 读保持寄存器 .....	107
(8) 读输入寄存器 .....	108
(9) 写单圈数据 .....	108
(10) 写单个寄存器 .....	109
(11) 写多个寄存器 .....	109
(12) 写多圈数据 .....	110
(13) 读多圈数据 .....	110
(14) 读多个保持寄存器 .....	111
4.14 系统安装方式及关节版本信息 .....	112
(1) 设置安装方式参数 .....	112
(2) 查询关节软件版本号 .....	112
(3) 查询末端接口板软件版本号 .....	112
4.15 透传力位混合控制补偿 (选配) .....	113
(1) 开启透传力位混合控制补偿模式 .....	113
(2) Force_Position_Move: 透传力位混合补偿 .....	113



(3) 关闭透传力位混合控制补偿模式 .....	116
五、升降机构（选配） .....	116
(1) 速度开环控制 .....	116
(2) 位置闭环控制 .....	116
(3) 获取升降机构状态 .....	117
六、通用扩展关节（选配） .....	117
(1) 扩展关节状态获取 .....	117
(2) 关节速度环控制 .....	118
(3) 关节位置环控制 .....	118
七、在线编程 .....	119
7.1 文件传输 .....	119
(1) 下发前准备 .....	119
(2) 下发前准备(I 系列) .....	119
(3) 发送过程中（从第二次开始） .....	120
(4) 发送校验返回 .....	120
(5) 规划过程中改变速度系数 .....	120
7.2 提示框 .....	121
(1) 弹窗提示 .....	121
7.3 拖动示教 .....	121



(1) 获取拖动示教轨迹 .....	121
7.4 在线编程存储列表 (I 系列) .....	121
(1) 查询在线编程程序列表 .....	121
(2) 查询在线编程运行状态 .....	122
(3) 开始运行指定编号轨迹 .....	123
(4) 删除指定编号的轨迹 .....	123
八、UDP 主动上报 (I 系列) .....	124
(1) UDP 机械臂状态主动上报接口 .....	124
(2) 查询 UDP 机械臂状态主动上报配置 .....	125
(3) 设置 UDP 机械臂状态主动上报配置 .....	126
九、电子围栏 (I 系列) .....	127
(1) 设置电子围栏使能状态 .....	127
(2) 获取电子围栏使能状态 .....	127
(3) 设置当前电子围栏参数 .....	128
(4) 获取当前电子围栏参数 .....	129
(5) 增加电子围栏 .....	130
(6) 更新电子围栏参数 .....	131
(7) 删除指定电子围栏 .....	132
(8) 查询所有电子围栏名称 .....	132



(9) 查询指定电子围栏参数 .....	132
(10) 查询所有电子围栏信息 .....	133
十、自碰撞安全检测 (I 系列) .....	134
(1) 设置自碰撞安全检测使能状态 .....	134
(2) 获取自碰撞安全检测使能状态 .....	135
示例：基础系列高速网口配置步骤 .....	136
1. 开启高速网口 .....	136
2. 配置高速网口 IP .....	136
3. 保存参数 .....	136
4. 更换端口，重启控制器 .....	137
附录：错误代码 .....	138
1. 系统错误代码 .....	138
2. 关节错误代码 .....	139



## 前言

睿尔曼机械臂对外采用统一的 JSON 格式进行数据通信，用户可使用 WIFI（AP 或者 STA）、以太网口通过标准的 TCP/IP 通信协议，按照要求的 JSON 格式发送数据，控制机械臂。

另外，用户也可通过机械臂控制器的 RS485 接口或者 RS232-USB（基础系列支持）接口发送 JSON 格式的字符串对机械臂进行控制。这两个接口默认波特率都为 460800BPS，停止位 1，数据位 8，无检验位。

以上所有控制模式无需用户手动切换，机械臂会自动识别，用户通过任一模式发送规定格式的指令后，机械臂会以相同的方式返回指令。

### 注意：

(1) 所有数据必须以发送新行的形式发出，即指令以“\r\n”的形式结束，否则机械臂不响应。

(2) 以上通讯模式不可同时使用，避免指令相互干扰。同时用户在使用 JSON 协议进行开发和测试时，请不要连接示教器，以防指令冲突。

(3) 基础系列机械臂除关节配置外的其他参数配置指令，如需保存，必须调用保存接口，否则重启后丢失，在保存过程中，切勿断电。

**备注：**睿尔曼基础系列机械臂配有高速网口，用户可通过指令打开高速网口，I 系列机械臂配备千兆网口，满足实时性要求较高的任务开发。

## 一、关节配置

### 1.1 关节配置命令集

对机械臂的关节参数进行设置，如果关节发生错误，则无法修改关节参数，必须先清除关节错误代码。另外设置关节之前，必须先将关节掉使能，否则会设



置不成功。

**注意：**关节所有参数在修改完成后，会自动保存到关节 Flash，立即生效，之后关节处于掉使能状态，修改完参数后必须发送指令控制关节上使能。

**注意：**睿尔曼机械臂在出厂前所有参数都已经配置到最佳状态，一般不建议用户修改关节的底层参数。若用户确需修改，首先应使机械臂处于非使能状态，然后再发送修改参数指令，参数设置成功后，发送关节恢复使能指令。需要注意的是，关节恢复使能时，用户需要保证关节处于静止状态，以免上使能过程中关节发生报错。关节正常上使能后，用户方可控制关节运动。

### (1) 设置关节最大转速

功能描述	设置关节最大转速
参数说明	set_joint_max_speed：设置关节最大转速指令 joint_max_speed：关节序号和最大转速，单位：RPM 建议使用默认最大转速，如需更改，设置的关节最大加速度与最大转速的比值需要 $\geq 1.5$ ，否则可能出现运动异常
命令格式	{s:s,s:[i,i]}
示例	{"command":"set_joint_max_speed","joint_max_speed":[2,3000 0]}
	说明：设置关节 2，最大转速 30RPM，转速分辨率 0.001RPM
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"set_joint_max_speed","joint_max_speed":true}

### (2) 设置关节最大加速度

功能描述	设置关节最大加速度
参数说明	set_joint_max_acc：设置关节最大加速度 joint_max_acc：关节序号和最大加速度，单位：RPM/s 建议使用默认关节最大加速度，如需更改，设置的关节最大加速度与最大转速的比值需要 $\geq 1.5$ ，否则可能出现运动异常



命令格式	{s:s,s:[i,i]}
示例	<pre>{"command":"set_joint_max_acc","joint_max_acc":[2,500000]}</pre> <p>说明：设置关节 2，最大加速度 500RPM/s，加速度分辨率 0.001 RPM/s</p>
返回值	<p>格式：{s:s,s:b}，true-设置成功，false-设置失败</p> <pre>{"command":"set_joint_max_acc","joint_max_acc":true}</pre>

### (3) 设置关节最小限位

功能描述	设置关节最小限位
参数说明	<p>set_joint_min_pos：设置关节最小限位</p> <p>joint_min_pos：关节序号和最小限位度数，单位：度</p>
命令格式	{s:s,s:[i,i]}
示例	<pre>{"command":"set_joint_min_pos","joint_min_pos":[1,-170000]}</pre> <p>说明：设置关节 1，最小限位度数-170°，分辨率 0.001°</p>
返回值	<p>格式：{s:s,s:b}，true-设置成功，false-设置失败</p> <pre>{"command":"set_joint_min_pos","joint_min_pos":true}</pre>

### (4) 设置关节最大限位

功能描述	设置关节最大限位
参数说明	<p>set_joint_max_pos：设置关节最大限位</p> <p>joint_max_pos：关节序号和最大限位度数，单位：度</p>
命令格式	{s:s,s:[i,i]}
示例	<pre>{"command":"set_joint_max_pos","joint_max_pos":[1,170000]}</pre> <p>说明：设置关节 1，最大限位度数 170°，分辨率 0.001°</p>
返回值	<p>格式：{s:s,s:b}，true-设置成功，false-设置失败</p> <pre>{"command":"set_joint_max_pos","joint_max_pos":true}</pre>

### (5) 设置关节最大转速（驱动器）

功能描述	设置关节最大转速（驱动器）
参数说明	set_joint_drive_max_speed：设置关节最大转速指令



	joint_max_speed: 关节序号和最大转速, 单位: RPM 建议使用默认最大转速, 如需更改, 设置的关节最大加速度与最大转速的比值需要 $\geq 1.5$ , 否则可能出现运动异常
命令格式	{s:s,s:[i,i]}
示例	{"command":"set_joint_drive_max_speed","joint_max_speed": [2,30000]}
	说明: 设置关节 2, 最大转速 30RPM, 转速分辨率 0.001RPM
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"set_joint_drive_max_speed","joint_max_speed":true}

#### (6) 设置关节最大加速度 (驱动器)

功能描述	设置关节最大加速度 (驱动器)
参数说明	set_joint_drive_max_acc: 设置关节最大加速度 joint_max_acc: 关节序号和最大加速度, 单位: RPM/s 建议使用默认关节最大加速度, 如需更改, 设置的关节最大加速度与最大转速的比值需要 $\geq 1.5$ , 否则可能出现运动异常
命令格式	{s:s,s:[i,i]}
示例	{"command":"set_joint_drive_max_acc","joint_max_acc":[2,500 000]}
	说明: 设置关节 2, 最大加速度 500RPM/s, 加速度分辨率 0.001 RPM/s
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"set_joint_drive_max_acc","joint_max_acc":true}

#### (7) 设置关节最小限位 (驱动器)

功能描述	设置关节最小限位 (驱动器)
参数说明	set_joint_drive_min_pos: 设置关节最小限位 joint_min_pos: 关节序号和最小限位度数, 单位: 度
命令格式	{s:s,s:[i,i]}





示例	<pre>{"command":"set_joint_drive_min_pos","joint_min_pos":[1,-170 000]}</pre> <p>说明：设置关节 1，最小限位度数-170 °，分辨率 0.001 °</p>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败 <pre>{"command":"set_joint_drive_min_pos","joint_min_pos":true}</pre>

#### (8) 设置关节最大限位（驱动器）

功能描述	设置关节最大限位（驱动器）
参数说明	set_joint_drive_max_pos：设置关节最大限位 joint_max_pos：关节序号和最大限位度数，单位：度
命令格式	{s:s,s:[i,i]}
示例	<pre>{"command":"set_joint_drive_max_pos","joint_max_pos":[1,170 000]}</pre> <p>说明：设置关节 1，最大限位度数 170 °，分辨率 0.001 °</p>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败 <pre>{"command":"set_joint_drive_max_pos","joint_max_pos":true}</pre>

#### (9) 设置关节使能状态

功能描述	设置关节使能状态
参数说明	set_joint_en_state：设置关节使能状态 joint_en_state：关节序号和使能状态，1：上使能 0：掉使能
命令格式	{s:s,s:[i,i]}
示例	<pre>{"command":"set_joint_en_state","joint_en_state":[6,1]}</pre> <p>说明：设置关节 6 上使能</p>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败 <pre>{"command":"set_joint_en_state","joint_en_state":true}</pre>

#### (10) 设置关节零位

功能描述	设置关节零位
参数说明	set_joint_zero_pos：设置关节零位



	joint_zero_pos: 关节序号
命令格式	{s:s,s:i}
示例	{"command":"set_joint_zero_pos","joint_zero_pos":3} 说明: 设置关节 3 位置为零位
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败 {"command":"set_joint_zero_pos","joint_zero_pos":true}

### (11) 清除关节错误代码

功能描述	清除关节错误代码
参数说明	set_joint_clear_err: 清除关节错误代码 joint_clear_err: 关节序号
命令格式	{s:s,s:i}
示例	{"command":"set_joint_clear_err","joint_clear_err":2} 说明: 清除关节 2 错误代码
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败 {"command":"set_joint_clear_err","joint_clear_err":true}

### (12) 一键设置关节限位

功能描述	一键设置关节限位
参数说明	auto_set_joint_limit: 一键设置关节限位
命令格式	{s:s,s:i}
示例	{"command":"auto_set_joint_limit","limit_mode":1} 说明: limit_mode: 限位设置的模式 1-正式模式, 各关节限位为规格参数中的软限位和硬件限位
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败 {"command":"auto_set_joint_limit","limit_mode":true}

## 1.2 关节配置查询集

### (1) 查询关节最大速度



功能描述	查询关节最大速度
参数说明	get_joint_max_speed: 查询关节最大速度
命令格式	{s:s}
示例	{"command":"get_joint_max_speed"}
	说明: 查询关节最大速度
返回值	见表 1.3- (1)

### (2) 查询关节最大加速度

功能描述	查询关节最大加速度
参数说明	get_joint_max_acc: 查询关节最大加速度
命令格式	{s:s}
示例	{"command":"get_joint_max_acc"}
	说明: 查询关节最大加速度
返回值	见表 1.3- (2)

### (3) 查询关节最小限位

功能描述	查询关节最小限位
参数说明	get_joint_min_pos: 查询关节最小限位
命令格式	{s:s}
示例	{"command":"get_joint_min_pos"}
	说明: 查询关节最小限位
返回值	见表 1.3- (3)

### (4) 查询关节最大限位

功能描述	查询关节最大限位
参数说明	get_joint_max_pos: 查询关节最大限位
命令格式	{s:s}
示例	{"command":"get_joint_max_pos"}
	说明: 查询关节最大限位



返回值	见表 1.3- (4)
-----	-------------

#### (5) 查询关节最大速度 (驱动器)

功能描述	查询关节最大速度 (驱动器)
参数说明	get_joint_drive_max_speed: 查询关节最大速度
命令格式	{s:s}
示例	{"command":"get_joint_drive_max_speed"}
	说明: 查询关节最大速度
返回值	见表 1.3- (5)

#### (6) 查询关节最大加速度 (驱动器)

功能描述	查询关节最大加速度 (驱动器)
参数说明	get_joint_drive_max_acc: 查询关节最大加速度
命令格式	{s:s}
示例	{"command":"get_joint_drive_max_acc"}
	说明: 查询关节最大加速度
返回值	见表 1.3- (6)

#### (7) 查询关节最小限位 (驱动器)

功能描述	查询关节最小限位 (驱动器)
参数说明	get_joint_drive_min_pos: 查询关节最小限位
命令格式	{s:s}
示例	{"command":"get_joint_drive_min_pos"}
	说明: 查询关节最小限位
返回值	见表 1.3- (7)

#### (8) 查询关节最大限位 (驱动器)

功能描述	查询关节最大限位 (驱动器)
参数说明	get_joint_drive_max_pos: 查询关节最大限位



命令格式	{s:s}
示例	{"command":"get_joint_drive_max_pos"}
	说明：查询关节最大限位
返回值	见表 1.3- (8)

### (9) 查询关节使能状态

功能描述	查询关节使能状态
参数说明	get_joint_en_state：查询关节使能状态
命令格式	{s:s}
示例	{"command":"get_joint_en_state"}
	说明：查询关节使能状态
返回值	见表 1.3- (9)

### (10) 查询关节错误代码

功能描述	查询关节错误代码
参数说明	get_joint_err_flag：查询关节错误代码
命令格式	{s:s}
示例	{"command":"get_joint_err_flag"}
	说明：查询关节错误代码
返回值	见表 1.3- (10)

## 1.3 关节配置反馈集

### (1) 反馈关节最大速度

功能描述	反馈所有关节最大速度
参数说明	joint_max_speed：反馈关节最大速度
命令格式	六自由度：{s:s,s:[i,i,i,i,i,i]}，七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	六自由度： {"state":"joint_max_speed","joint_speed":[30,30,30,30,30,30]}
	七自由度：



	<pre>{"state":"joint_max_speed","joint_speed":[30,30,30,30,30,30,30]}</pre>
	说明：依次反馈关节最大转速均为 0.03RPM，单位 RPM，分辨率：0.001RPM

## (2) 反馈关节最大加速度

功能描述	反馈关节最大加速度
参数说明	joint_max_acc：反馈关节最大加速度
命令格式	六自由度：{s:s,s:[i,i,i,i,i,i]}，七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	六自由度： <pre>{"state":"joint_max_acc","joint_acc":[500,500,500,500,500,500]}</pre>
	七自由度： <pre>{"state":"joint_max_acc","joint_acc":[500,500,500,500,500,500,500]}</pre>
	说明：依次反馈关节最大加速度均为 0.5RPM/s,单位 RPM/s，分辨率：0.001RPM/s

## (3) 反馈关节最小限位

功能描述	反馈关节最小限位
参数说明	joint_min_pos：反馈关节最小限位
命令格式	六自由度：{s:s,s:[i,i,i,i,i,i]}，七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	六自由度 <pre>{"state":"joint_min_pos","min_pos":[-170000,-110000,-170000,-110000,-170000,-110000]}</pre>
	七自由度 <pre>{"state":"joint_min_pos","min_pos":[-170000,-110000,-170000,-170000,-110000,-170000,-110000]}</pre>
	说明：反馈关节最小限位，六自由度的关节 1、3、5 最小位置-170°，关节 2、4、6 最小位置-110°，七自由度的关节 1、3、4、6 最小位置-170°，关节 2、5、7 最小位置-110° 关节单位：度，分辨率：0.001°



#### (4) 反馈关节最大限位

功能描述	反馈关节最大限位
参数说明	joint_max_pos: 反馈关节最大限位
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i]}, 七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	<p>六自由度</p> <pre>{"state":"joint_max_pos","max_pos":[170000,110000,170000,110000,170000,110000]}</pre> <p>七自由度:</p> <pre>{"state":"joint_max_pos","max_pos":[170000,110000,170000,170000,110000,170000,110000]}</pre> <p>说明: 反馈关节最大限位, 六自由度的关节 1、3、5 最小位置 170°, 关节 2、4、6 最小位置 110°, 七自由度的关节 1、3、4、6 最小位置 170°, 关节 2、5、7 最小位置 110° 关节单位: 度, 分辨率: 0.001°</p>

#### (5) 反馈关节最大速度 (驱动器)

功能描述	反馈所有关节最大速度 (驱动器)
参数说明	get_joint_drive_max_speed: 反馈关节最大速度
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i]}, 七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	<p>六自由度:</p> <pre>{"command":"get_joint_drive_max_speed","joint_speed":[30,30,30,30,30,30]}</pre> <p>七自由度:</p> <pre>{"command":"get_joint_drive_max_speed","joint_speed":[30,30,30,30,30,30,30]}</pre> <p>说明: 依次反馈关节最大转速均为 0.03RPM, 单位 RPM, 分辨率: 0.001RPM</p>

#### (6) 反馈关节最大加速度 (驱动器)

功能描述	反馈关节最大加速度 (驱动器)
参数说明	get_joint_drive_max_acc: 反馈关节最大加速度



命令格式	六自由度：{s:s,s:[i,i,i,i,i,i]}，七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	<p>六自由度：</p> <pre>{"command":"get_joint_drive_max_acc","joint_acc":[500,500,500,500,500,500]}</pre> <p>七自由度：</p> <pre>{"command":"get_joint_drive_max_acc","joint_acc":[500,500,500,500,500,500,500]}</pre> <p>说明：依次反馈关节最大加速度均为 0.5RPM/s,单位 RPM/s，分辨率：0.001RPM/s</p>

### (7) 反馈关节最小限位（驱动器）

功能描述	反馈关节最小限位（驱动器）
参数说明	get_joint_drive_min_pos：反馈关节最小限位
命令格式	六自由度：{s:s,s:[i,i,i,i,i,i]}，七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	<p>六自由度</p> <pre>{"command":"get_joint_drive_min_pos","min_pos":[-170000,-110000,-170000,-110000,-170000,-110000]}</pre> <p>七自由度</p> <pre>{"command":"get_joint_drive_min_pos","min_pos":[-170000,-110000,-170000,-110000,-170000,-110000,-170000]}</pre> <p>说明：反馈关节最小限位，六自由度的关节 1、3、5 最小位置-170°，关节 2、4、6 最小位置-110°，七自由度的关节 1、3、4、6 最小位置-170°，关节 2、5、7 最小位置-110° 关节单位：度，分辨率：0.001°</p>

### (8) 反馈关节最大限位（驱动器）

功能描述	反馈关节最大限位（驱动器）
参数说明	get_joint_drive_max_pos：反馈关节最大限位
命令格式	六自由度：{s:s,s:[i,i,i,i,i,i]}，七自由度{s:s,s:[i,i,i,i,i,i,i]}
示例	<p>六自由度</p> <pre>{"command":"get_joint_drive_max_pos","max_pos":[170000,110000,170000,110000,170000,110000]}</pre>





	<p>七自由度：</p> <pre>{"command":"get_joint_drive_max_pos","max_pos":[170000,110000,170000,170000,110000,170000,110000]}</pre>
	<p>说明：反馈关节最大限位，六自由度的关节 1、3、5 最小位置 170°，关节 2、4、6 最小位置 110°，七自由度的关节 1、3、4、6 最小位置 170°，关节 2、5、7 最小位置 110° 关节单位：度，分辨率：0.001°</p>

### (9) 反馈关节使能状态

功能描述	反馈关节使能状态
参数说明	joint_en_state：反馈关节使能状态
命令格式	六自由度：{s:s,s:[i,i,i,i,i,i]}，七自由度{s:s,s:[i,i,i,i,i,i]}
示例	<p>六自由度：</p> <pre>{"state":"joint_en_state","en_state":[1,1,1,1,1,0]}</pre> <p>七自由度：</p> <pre>{"state":"joint_en_state","en_state":[1,1,1,1,1,1,0]}</pre> <p>说明：反馈关节使能状态，1-上使能状态，0-掉使能状态</p>

### (10) 反馈关节错误代码

功能描述	反馈关节错误代码
参数说明	joint_err_flag：反馈关节错误代码
命令格式	六自由度：{s:s,s:[i,i,i,i,i,i],s:[i,i,i,i,i,i]}
	七自由度：{s:s,s:[i,i,i,i,i,i],s:[i,i,i,i,i,i]}
示例	<p>六自由度：</p> <pre>{"state":"joint_err_flag","err_flag":[0,0,0,0,0,1],"brake_state":[0,0,0,0,0,1]}</pre> <p>七自由度：</p> <pre>{"state":"joint_err_flag","err_flag":[0,0,0,0,0,0,1],"brake_state":[0,0,0,0,0,0,1]}</pre> <p>说明：</p> <p>err_flag：反馈关节错误代码，错误代码为整型</p>



	brake_state: 反馈关节抱闸状态, 1 代表抱闸未打开, 0 代表抱闸已打开
--	---

## 二、机械臂配置

### 2.1 机械臂配置-运动参数命令集

机械臂末端参数设置完毕后, 基础系列需要调用 4.3- (9) 手动保存到控制器 Flash, 否则断电之后不会保存。

#### (1) 设置机械臂末端最大线速度

功能描述	设置机械臂末端最大线速度
参数说明	set_arm_max_line_speed: 设置机械臂末端最大线速度 arm_line_speed: 目标线速度单位: m/s 建议使用默认最大线速度, 如需更改, 设置的机械臂末端最大线加速度与最大线速度的比值需要 $\geq 3$ , 否则可能出现运动异常
命令格式	{s:s,s:i}
示例	<pre>{"command": "set_arm_max_line_speed", "arm_line_speed": 500}</pre> <p>说明: 设置机械臂末端最大线速度 0.5m/s, 分辨率 0.001m/s</p>
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败 <pre>{"command": "set_arm_max_line_speed", "arm_line_speed": true}</pre>

#### (2) 设置机械臂末端最大线加速度

功能描述	设置机械臂末端最大线加速度
参数说明	set_arm_max_line_acc: 设置机械臂末端最大线加速度 arm_line_acc: 目标线加速度单位: $m/s^2$ 建议使用默认最大线加速度, 如需更改, 设置的机械臂末端最大线加速度与最大线速度的比值需要 $\geq 3$ , 否则可能出现运动异常
命令格式	{s:s,s:i}
示例	<pre>{"command": "set_arm_max_line_acc", "arm_line_acc": 2000}</pre>



	说明：设置机械臂末端最大线加速度 $2\text{m/s}^2$ ，分辨率 $0.001\text{m/s}^2$
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"set_arm_max_line_acc","arm_line_acc":true}

### (3) 设置机械臂末端最大角速度

功能描述	设置机械臂末端最大角速度
参数说明	set_arm_max_angular_speed：设置机械臂末端最大角速度 arm_angular_speed：目标角速度单位：rad/s 建议使用默认最大角速度，如需更改，设置的机械臂末端最大角加速度与最大角速度的比值需要 $\geq 3$ ，否则可能出现运动异常
命令格式	{s:s,s:i}
示例	<div>{"command":"set_arm_max_angular_speed","arm_angular_speed":200}</div> <div>说明：设置机械臂末端最大角速度 <math>0.2\text{rad/s}</math>，分辨率 <math>0.001\text{rad/s}</math></div>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"set_arm_max_angular_speed","arm_angular_speed":true}

### (4) 设置机械臂末端最大角加速度

功能描述	设置机械臂末端最大角加速度
参数说明	set_arm_max_angular_acc：设置机械臂末端最大角加速度 arm_angular_acc：目标角加速度单位： $\text{rad/s}^2$ 建议使用默认最大角加速度，如需更改，设置的机械臂末端最大角加速度与最大角速度的比值需要 $\geq 3$ ，否则可能出现运动异常
命令格式	{s:s,s:i}
示例	<div>{"command":"set_arm_max_angular_acc","arm_angular_acc":4000}</div> <div>说明：设置机械臂末端最大角加速度 <math>4\text{rad/s}^2</math>，分辨率 <math>0.001\text{rad/s}^2</math></div>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败



	<pre>{"command":"set_arm_max_angular_acc","arm_angular_acc":true}</pre>
--	---

### (5) 初始化机械臂参数

功能描述	初始化机械臂末端参数
参数说明	set_arm_init: 初始化机械臂末端参数
命令格式	{s:s}
示例	<pre>{"command":"set_arm_init"}</pre>
	说明: 初始化机械臂参数, 机械臂的末端参数恢复到默认值。其中 末端线速度: 0.25m/s 末端线加速度: 1.6m/s <sup>2</sup> 末端角速度: 0.6rad/s 末端角加速度: 4rad/s <sup>2</sup>
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	<pre>{"command":"set_arm_init","arm_init":true}</pre>

### (6) 设置碰撞防护等级

功能描述	设置机械臂碰撞防护等级
参数说明	set_collision_stage: 设置机械臂碰撞防护等级 collision_stage: 等级, 范围: 0~8
命令格式	{s:s,s:i}
示例	<pre>{"command":"set_collision_stage","collision_stage":1}</pre>
	说明: 设置机械臂碰撞防护等级, 等级越高, 检测越灵敏
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	<pre>{"command":"set_collision_state","collision_state":true}</pre>

### (7) 查询碰撞防护等级

功能描述	设置机械臂碰撞防护等级
参数说明	get_collision_stage: 设置机械臂碰撞防护等级 collision_stage: 等级, 范围: 0~8
命令格式	{s:s}



示例	<code>{"command":"get_collision_stage"}</code>
	说明：设置机械臂碰撞防护等级，等级越高，检测越灵敏
返回值	格式： <code>{s:s,s:i}</code>
	<code>{"state":"get_collision_stage","collision_stage":5}</code>

#### (8) 重新设置机械臂 DH 参数

功能描述	重新设置机械臂 DH 参数
参数说明	set_DH_data：设置机械臂 DH 参数
命令格式	<code>{s:s,s:[i,i,i,i,i]}</code>
示例	<code>{"command":"set_DH_data","data":[2405,2560,2100,1440,0]}</code>
	说明：设置机械臂 DH 参数，精度：0.1mm，上述指令内容如下： lsb:240.5mm lse:256mm lew:210mm lwt:144mm d3:0mm
返回值	格式： <code>{s:s,s:b}</code> ，true-设置成功，false-设置失败
	<code>{"command":"set_DH_data","set_state":true}</code>
备注	该指令用户不可自行使用，必须配合测量设备进行绝对精度补偿时方可使用，否则会导致机械臂参数错误！

#### (9) 查询机械臂 DH 参数

功能描述	查询机械臂 DH 参数
参数说明	get_DH_data：查询机械臂 DH 参数
命令格式	<code>{s:s,s:i}</code>
示例	<code>{"command":"get_DH_data"}</code>
	说明：查询机械臂 DH 参数
返回值	格式： <code>{s:s,s:[i,i,i,i,i]}</code>
	<code>{"state":"DH_data","data":[2405,2560,2100,1440,0]}</code>



	<p>说明：设置机械臂 DH 参数，精度：0.1mm，上述指令内容如下：</p> <p>lsb:240.5mm</p> <p>lse:256mm</p> <p>lew:210mm</p> <p>lwt:144mm</p> <p>d3:0mm</p>
--	--

### (10) 重新设置关节零位补偿角度

功能描述	重新设置关节零位补偿角度，用于校正绝对定位精度
参数说明	set_joint_zero_offset
命令格式	{s:s,s:[i,i,i,i,i,i]}
示例	{"command":"set_joint_zero_offset","offset":[1000,-2000,3000,-4000,5000,-6000]}
	<p>说明：设置关节零位偏移，精度：0.001 °</p> <p>关节 1~6 的零位补偿角度：1 °，-2 °，3 °，-4 °，5 °，-6 °</p>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"set_joint_zero_offset","set_state":true}
备注	该指令用户不可自行使用，必须配合测量设备进行绝对精度补偿时方可使用，否则会导致机械臂参数错误！

## 2.2 机械臂配置-运动参数查询集

### (1) 查询机械臂末端最大线速度

功能描述	查询机械臂末端最大线速度
参数说明	get_arm_max_line_speed：查询机械臂末端最大线速度
命令格式	{s:s}
示例	{"command":"get_arm_max_line_speed"}
	说明：查询机械臂末端最大线速度
返回值	见表 2.3- (1)

### (2) 查询机械臂末端最大线加速度



功能描述	查询机械臂末端最大线加速度
参数说明	get_arm_max_line_acc：查询机械臂末端最大线加速度
命令格式	{s:s}
示例	{"command":"get_arm_max_line_acc"}
	说明：查询机械臂末端最大线加速度
返回值	见表 2.3-（2）

### （3）查询机械臂末端最大角速度

功能描述	查询机械臂末端最大角速度
参数说明	get_arm_max_angular_speed：查询机械臂末端最大角速度
命令格式	{s:s}
示例	{"command":"get_arm_max_angular_speed"}
	说明：查询机械臂末端最大角速度
返回值	见表 2.3-（3）

### （4）查询机械臂末端最大角加速度

功能描述	查询机械臂末端最大角加速度
参数说明	get_arm_max_angular_acc：查询机械臂末端最大角加速度
命令格式	{s:s}
示例	{"command":"get_arm_max_angular_acc"}
	说明：查询机械臂末端最大线角加速度
返回值	见表 2.3-（4）

## 2.3 机械臂配置-运动参数反馈集

### （1）反馈机械臂末端最大线速度

功能描述	反馈机械臂末端最大线速度
------	--------------



参数说明	arm_max_line_speed: 反馈机械臂末端最大线速度
命令格式	{s:s,s:i}
示例	{"state":"arm_max_line_speed","arm_line_speed":500}
	说明: 反馈机械臂末端最大线速度, 0.5m/s, 分辨率: 0.001m/s

### (2) 反馈机械臂末端最大线加速度

功能描述	反馈机械臂末端最大线加速度
参数说明	arm_max_line_acc: 反馈机械臂末端最大线加速度
命令格式	{s:s,s:i}
示例	{"state":"arm_max_line_acc","arm_line_acc":200}
	说明: 反馈机械臂末端最大线加速度, 0.2m/s <sup>2</sup> , 分辨率: 0.001 m/s <sup>2</sup>

### (3) 反馈机械臂末端最大角速度

功能描述	反馈机械臂末端最大角速度
参数说明	arm_max_angular_speed: 反馈机械臂末端最大角速度
命令格式	{s:s,s:i}
示例	{"state":"arm_max_angular_speed","arm_angular_speed":1000}
	说明: 反馈机械臂末端最大角速度, 1rad/s, 分辨率: 0.001rad/s

### (4) 反馈机械臂末端最大角加速度

功能描述	反馈机械臂末端最大角加速度
参数说明	arm_max_angular_acc: 反馈机械臂末端最大角加速度
命令格式	{s:s,s:i}
示例	{"state":"arm_max_angular_acc","arm_angular_acc":10000}
	说明: 反馈机械臂末端最大角加速度, 10rad/s <sup>2</sup> , 分辨率: 0.001 rad/s <sup>2</sup>

## 2.4 机械臂配置-工具坐标系命令集





### (1) 自动计算工具坐标系(标定参考点)

功能描述	自动计算工具坐标系（六点法）：机械臂只能存储 10 个工具坐标系，若超过 10 个，则新建立工具不成功。
参数说明	set_auto_tool_frame：自动计算工具坐标系 point_num：1~6 为标定参考点
命令格式	{s:s,s:i}
示例	<pre>{"command":"set_auto_tool_frame","point_num":1} {"command":"set_auto_tool_frame","point_num":2} {"command":"set_auto_tool_frame","point_num":3} {"command":"set_auto_tool_frame","point_num":4} {"command":"set_auto_tool_frame","point_num":5} {"command":"set_auto_tool_frame","point_num":6}</pre> <p>说明：自动计算工具坐标系，标定当前位置为参考点 6 备注：机械臂上电初始化后，默认无负载</p>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败 <pre>{"command":"set_auto_tool_frame","auto_tool_frame":true}</pre>

### (2) 自动计算工具坐标系(自动计算生成工具)

功能描述	自动计算工具坐标系（六点法）：机械臂只能存储 10 个工具坐标系，若超过 10 个，则新建立工具不成功。
参数说明	set_auto_tool_frame：自动计算工具坐标系 tool_name：工具坐标系名称，不能超过 10 个字符 payload:单位：g，最高不超过 5000g； position：质心位置，单位：mm，精度 0.001mm
命令格式	{s:s,s:s,s:i,s:[i,i,i]}
示例	<pre>{"command":"generate_auto_tool_frame","tool_name":"tool_frame","payload":5000,"position":[1000,2000,3000]}</pre> <p>说明：自动计算工具坐标系，名称 tool2_frame，标定当前位置为参考点 6 末端负载 5000g，质心位置：x-1mm,y-2mm,z-3mm</p>



	备注：机械臂上电初始化后，默认无负载
返回值	格式：{s:s,s:b}, true-设置成功, false-设置失败
	{"command":"set_auto_tool_frame","auto_tool_frame":true}

### (3) 手动输入工具坐标系

功能描述	手动输入工具坐标系：机械臂只能存储 10 个工具坐标系，若超过 10 个，则新建立工具不成功。
参数说明	set_manual_tool_frame：手动输入工具坐标系参数 tool_name：工具坐标系名称，不能超过 10 个字符 tool_pose：工具相对机械臂末端法兰中心位姿
命令格式	{s:s,s:s,s:[i,i,i,i,i],s:i,s:[i,i,i]}
示例	<p>{"command":"set_manual_tool_frame","tool_name":"tool_frame","tool_pose":[100000,200000,30000,400,500,600],"payload":5000,"position":[1000,2000,3000]}</p> <p>说明：手动输入工具坐标系，名称 tool2_frame， 工具位置：x：0.1m，y：0.2m，z：0.03m，位置精度：0.001mm 工具姿态：rx：0.4rad，ry：0.5rad，rz：0.6rad，姿态精度：0.001rad 末端负载 5000g，质心位置：x-1mm,y-2mm,z-3mm payload：单位：g，最高不超过 5000g； position：质心位置，单位：mm，精度 0.001mm 备注：机械臂上电初始化后，默认无负载</p>
返回值	<p>格式：{s:s,s:b},true-设置成功, false-设置失败</p> <p>{"command":"set_manual_tool_frame","manual_tool_frame":true}</p>

### (4) 切换当前工具坐标系

功能描述	切换当前工具坐标系
参数说明	set_change_tool_frame：切换当前工具坐标系 tool_name：工具坐标系名称
命令格式	{s:s,s:s}



示例	<pre>{"command":"set_change_tool_frame","tool_name":"tool2_frame"}</pre>
	说明：切换当前工具坐标系，名称 tool2_frame
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	<pre>{"command":"set_change_tool_frame","change_tool_frame":true}</pre>

### (5) 删除工具坐标系

功能描述	删除工具坐标系
参数说明	set_delete_tool_frame：删除工具坐标系 tool_name：工具坐标系名称
命令格式	{s:s,s:s}
示例	<pre>{"command":"set_delete_tool_frame","tool_name":"tool2_frame"}</pre>
	说明：删除工具坐标系，名称 tool2_frame
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	<pre>{"command":"set_delete_tool_frame","delete_tool_frame":true}</pre>

## 2.5 机械臂配置-工作坐标系命令集

### (1) 自动设置工作坐标系

功能描述	设置工作坐标系：机械臂只能存储 10 个工作坐标系，若超过 10 个，则新建立坐标系不成功。
参数说明	set_work_frame：设置工作坐标系 frame_name：工作坐标系名称，不能超过 10 个字符 point_num：参考点 1~3 代表工作坐标系原点、X 轴一点和 Y 轴上一点，4 代表根据前三个标定点计算工作坐标系。
命令格式	{s:s,s:s,s:i}
示例	<pre>{"command":"set_auto_work_frame","frame_name":"work_frame", "point_num":3}</pre>
	说明：设置工作坐标系，名称 work2_frame，将当前位置标定为点 3



返回值	格式: {s:s,s:b},true-设置成功, false-设置失败
	{"command":"set_auto_work_frame","auto_work_frame":true}

## (2) 手动输入工作坐标系

功能描述	手动输入工作坐标系: 机械臂只能存储 10 个工作坐标系, 若超过 10 个, 则新建立坐标系不成功。
参数说明	set_manual_work_frame: 手动输入工作坐标系 frame_name: 工作坐标系名称, 不能超过 10 个字符 frame_pose: 工作位置
命令格式	{s:s,s:s,s:[i,i,i,i,i]}
示例	{"command":"set_manual_work_frame","frame_name":"work_frame","frame_pose":[100000,200000,30000,400,500,600]}
	说明: 手动输入工作坐标系, 名称 work2_frame, 坐标系位置: x: 0.1m, y:0.2m, z: 0.03m, 位置精度: 0.001m 坐标系姿态: rx: 0.4rad, ry: 0.5rad, rz: 0.6rad, 姿态精度: 0.001rad
返回值	格式: {s:s,s:b},true-设置成功, false-设置失败
	{"command":"set_manual_work_frame","manual_work_frame":true}

## (3) 切换当前工作坐标系

功能描述	切换当前工作坐标系
参数说明	set_change_work_frame: 切换当前工作坐标系 frame_name: 工具坐标系名称
命令格式	{s:s,s:s}
示例	{"command":"set_change_work_frame","frame_name":"work2_frame"}
	说明: 切换当前工作坐标系, 名称 work2_frame
返回值	格式: {s:s,s:b},true-设置成功, false-设置失败
	{"command":"set_change_work_frame","change_work_frame":true}



	ue}
--	-----

#### (4) 删除工作坐标系

功能描述	删除工作坐标系
参数说明	set_delete_work_frame: 删除工作坐标系 frame_name: 工作坐标系名称
命令格式	{s:s,s:s}
示例	<pre>{"command":"set_delete_work_frame","frame_name":"work2_frame"}</pre> <p>说明: 删除工作坐标系, 名称 work2_frame</p>
返回值	<p>格式: {s:s,s:b},true-设置成功, false-设置失败</p> <pre>{"command":"set_delete_work_frame","delete_work_frame":true}</pre>

## 2.6 机械臂配置-坐标系查询集

#### (1) 查询当前工具

功能描述	查询当前工具
参数说明	get_current_tool_frame: 查询当前工具
命令格式	{s:s}
示例	<pre>{"command":"get_current_tool_frame"}</pre> <p>说明: 查询当前工具</p>
返回值	见表 2.7- (1)

#### (2) 查询已有所有工具名称

功能描述	查询已有所有工具名称
参数说明	get_total_tool_frame: 查询所有工具名称
命令格式	{s:s}
示例	<pre>{"command":"get_total_tool_frame"}</pre> <p>说明: 查询所有工具名称</p>



返回值	见表 2.7- (2)
-----	-------------

### (3) 查询指定工具信息

功能描述	查询指定工具信息
参数说明	get_tool_frame: 查询指定工具信息 tool_name: 工具名称
命令格式	{s:s,s:s}
示例	{"command":"get_tool_frame","tool_name":"tool"}
	说明: 查询指定工具信息, 工具名称: tool
返回值	见表 2.7- (3)

### (4) 查询当前工作坐标系

功能描述	查询当前工作坐标系
参数说明	get_current_work_frame: 查询当前工作坐标系
命令格式	{s:s}
示例	{"command":"get_current_work_frame"}
	说明: 查询当前工作坐标系
返回值	见表 2.7- (4)

### (5) 查询已有所有工作坐标系名称

功能描述	查询已有所有工作坐标系名称
参数说明	get_total_work_frame: 查询所有工作坐标系名称
命令格式	{s:s}
示例	{"command":"get_total_work_frame"}
	说明: 查询所有工作坐标系名称
返回值	见表 2.7- (5)

### (6) 查询指定工作坐标系

功能描述	查询指定工作坐标系
------	-----------



参数说明	get_work_frame: 查询指定工作坐标系 frame_name: 坐标系名称
命令格式	{s:s,s:s}
示例	<pre>{"command":"get_work_frame","frame_name":"work1"}</pre> <p>说明: 查询指定工作坐标系, 坐标系名称: work1</p>
返回值	见表 2.7- (6)

## 2.7 机械臂配置-坐标系返回集

### (1) 返回当前工具信息

功能描述	返回当前工具信息
参数说明	current_tool_frame: 返回当前工具信息
命令格式	{s:s,s:s,s:[i,i,i,i,i],s:i,s:[i,i,i]}
示例	<pre>{"state":"current_tool_frame","tool_name":"tool2_frame","pose": [100000,200000,30000,400,500,600],"payload":5000,"position": [1000,2000,3000]}</pre> <p>说明: 返回当前工具信息, 工具名称: tool2_frame 工具位置: x: 0.1m, y:0.2m, z: 0.03m, 位置精度: 0.001mm 工具姿态: rx: 0.4rad, ry: 0.5rad, rz: 0.6rad, 姿态精度: 0.001rad 重量: payload: 5kg 精度 0.001kg 质心: position: 1mm 精度 0.001mm</p>

### (2) 返回所有工具名称

功能描述	返回所有工具名称, NULL 为空
参数说明	total_tool_frame: 返回所有工具名称
命令格式	{s:s,s:s,s:[s,s,...,s]}
示例	<pre>{"state":"total_tool_frame","tool_names":["base_tool1","base_to ol2"...,"NULL"]}</pre> <p>说明: 返回所有工具名称, 共 10 个, 工具名称: base_tool1, base_tool2, 其中 "NULL" 为空坐标系, 未建立</p>



### (3) 返回指定工具信息

功能描述	返回指定工具信息
参数说明	given_tool_frame: 返回指定工具信息
命令格式	{s:s,s:s,s:[i,i,i,i,i,i],s:i,s:[i,i,i]}
示例	成功: <pre>{"state":"given_tool_frame","tool_name":"tool2_frame","pose":[100000,200000,30000,400,500,600],"payload":5000,"position":[1000,2000,3000]}</pre>
	失败: <pre>{"command":"get_tool_frame","get_state":false}</pre>
	说明: 返回指定工具信息, 工具名称: tool2_frame, 工具位置: x: 0.1m, y:0.2m, z: 0.03m, 位置精度: 0.001mm 工具姿态: rx: 0.4rad, ry: 0.5rad, rz: 0.6rad, 姿态精度: 0.001rad 重量: payload: 5kg 精度 0.001kg 质心: position: 1mm 精度 0.001mm

### (4) 返回当前工作坐标系信息

功能描述	返回当前工作坐标系信息
参数说明	current_work_frame: 返回当前工作坐标系信息
命令格式	{s:s,s:s,s:[i,i,i,i,i,i]}
示例	<pre>{"state":"current_work_frame","frame_name":"work2_frame","pose":[100000,200000,30000,400,500,600]}</pre>
	说明: 返回当前工作坐标系信息, 坐标系名称: work2_frame, 坐标系位置: x: 0.1m, y:0.2m, z: 0.03m, 位置精度: 0.001mm 坐标系姿态: rx: 0.4rad, ry: 0.5rad, rz: 0.6rad, 姿态精度: 0.001rad

### (5) 返回所有工作坐标系名称

功能描述	返回所有工作坐标系名称,总共 10 个, NULL 为空, 表示未建立
------	-------------------------------------





参数说明	total_work_frame: 返回所有工作坐标系名称
命令格式	{s:s,s:s,s:[s,s,...,s]}
示例	{"state":"total_work_frame","frame_names":["work1","work2"...,"NULL"]}
	说明: 返回所有工作坐标系名称, 坐标系名称: work1,work2,...

## (6) 返回指定坐标系信息

功能描述	返回指定坐标系信息
参数说明	given_work_frame: 返回指定坐标系信息
命令格式	{s:s,s:s,s:[i,i,i,i,i]}
示例	<p>成功:</p> <pre>{"state":"given_work_frame","frame_name":"work2_frame","pose":[100000,200000,30000,400,500,600]}</pre> <p>失败:</p> <pre>{"command":"get_work_frame","get_state":false}</pre>
	<p>说明: 返回指定坐标系信息, 坐标系名称: work2_frame, 坐标系位置: x: 0.1m, y:0.2m, z: 0.03m, 位置精度: 0.001m 坐标系姿态: rx: 0.4rad, ry: 0.5rad, rz: 0.6rad, 姿态精度: 0.001rad</p>

## 2.8 机械臂配置-状态查询集

### (1) 查询机械臂状态

功能描述	查询机械臂状态
参数说明	get_current_arm_state: 查询机械臂状态
命令格式	{s:s}
示例	{"command":"get_current_arm_state"}
	说明: 查询机械臂状态



返回值	见表 2.9- (1)
-----	-------------

## (2) 查询关节温度

功能描述	查询关节温度
参数说明	get_current_joint_temperature: 查询关节温度
命令格式	{s:s}
示例	{"command":"get_current_joint_temperature"}
	说明: 查询关节温度
返回值	见表 2.9- (2)

## (3) 查询关节当前电流

功能描述	查询关节当前电流
参数说明	get_current_joint_current: 查询关节当前电流
命令格式	{s:s}
示例	{"command":"get_current_joint_current"}
	说明: 查询关节当前电流
返回值	见表 2.9- (3)

## (4) 查询关节当前电压

功能描述	查询关节当前电压
参数说明	get_current_joint_voltage: 查询关节当前电压
命令格式	{s:s}
示例	{"command":"get_current_joint_voltage"}
	说明: 查询关节当前电压
返回值	见表 2.9- (4)

# 2.9 机械臂配置-状态反馈帧

## (1) 反馈机械臂状态

功能描述	反馈机械臂状态: 包括关节角度、末端位姿、机械臂错误代码、
------	-------------------------------



	控制器错误代码
参数说明	current_arm_state: 反馈机械臂状态 joint: 关节角度 pose: 末端位姿 arm_err: 机械臂错误代码 sys_err: 控制器错误代码
命令格式	六自由度: {s:s,s:{s:[i,i,i,i,i,i],s:[i,i,i,i,i,i],s:i,s:i}} 七自由度: {s:s,s:{s:[i,i,i,i,i,i,i],s:[i,i,i,i,i,i,i],s:i,s:i,i}}
示例	<p>六自由度:</p> <pre>{"state":"current_arm_state","arm_state":{"joint":[100,200,300,400,500,600],"pose":[100000,200000,30000,400,500,600],"arm_err":0,"sys_err":0}}</pre> <p>七自由度:</p> <pre>{"state":"current_arm_state","arm_state":{"joint":[100,200,300,400,500,600,700],"pose":[100000,200000,30000,400,500,600],"arm_err":0,"sys_err":0}}</pre> <p>说明: 反馈机械臂状态,</p> <p>六自由度机械臂关节 1~6 角度依次为: <math>0.1^{\circ}</math>, <math>0.2^{\circ}</math>, <math>0.3^{\circ}</math>, <math>0.4^{\circ}</math>, <math>0.5^{\circ}</math>, <math>0.6^{\circ}</math>, 精度: <math>0.001^{\circ}</math></p> <p>七自由度机械臂关节 1~7 角度依次为: <math>0.1^{\circ}</math>, <math>0.2^{\circ}</math>, <math>0.3^{\circ}</math>, <math>0.4^{\circ}</math>, <math>0.5^{\circ}</math>, <math>0.6^{\circ}</math>, <math>0.7^{\circ}</math>, 精度: <math>0.001^{\circ}</math></p> <p>位置: x: 0.1m, y:0.2m, z: 0.03m, 位置精度: 0.001mm</p> <p>姿态: rx: 0.4rad, ry: 0.5rad, rz: 0.6rad, 姿态精度: 0.001rad</p> <p>arm_err: 机械臂错误代码, 指机械臂运行规划中的软件错误: 0</p> <p>sys_err: 控制器错误代码, 指控制器运行过程中的硬件错误: 0</p>

## (2) 反馈关节温度

功能描述	反馈关节温度
参数说明	current_joint_temperature: 反馈关节温度单位: $^{\circ}\text{C}$
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i]}



	七自由度: {s:s,[i,i,i,i,i,i]}
示例	<p>六自由度:</p> <pre>{"state":"current_joint_temperature","joint_temperature":[27500,28000,26800,26800,28900,30100]}</pre> <p>七自由度:</p> <pre>{"state":"current_joint_temperature","joint_temperature":[27500,28000,26800,26800,28900,30100,31100]}</pre> <p>说明: 反馈关节温度, 六自由度机械臂关节温度[27.5,28.0,26.8,26.8,28.9,30.1], 七自由度机械臂关节温度[27.5,28.0,26.8,26.8,28.9,30.1,31.1], 单位: °C, 精度: 0.001°C</p>

### (3) 反馈关节当前电流

功能描述	反馈关节当前电流
参数说明	current_joint_current: 反馈关节当前电流单位: mA, 精度: 0.001mA
命令格式	<p>六自由度: {s:s,[i,i,i,i,i,i]}</p> <p>七自由度: {s:s,[i,i,i,i,i,i]}</p>
示例	<p>六自由度:</p> <pre>{"state":"current_joint_current","joint_current":[65,-200,170,200,-300,168]}</pre> <p>七自由度:</p> <pre>{"state":"current_joint_current","joint_current":[65,-200,170,200,-300,168,178]}</pre> <p>说明: 反馈关节当前电流, 六自由度机械臂关节 1~6 电流依次为: 0.065mA, -0.2mA, 0.17mA, 0.2mA, -0.3mA, 0.168mA。七自由度机械臂关节 1~7 电流依次为: 0.065mA, -0.2mA, 0.17mA, 0.2mA, -0.3mA, 0.168mA, 0.178mA</p>

### (4) 反馈关节当前电压

功能描述	反馈关节当前电压
参数说明	current_joint_voltage: 反馈关节当前电压单位: V, 精度: 0.001V



命令格式	六自由度: {s:s,s:[i,i,i,i,i,i]} 七自由度: {s:s,s:[i,i,i,i,i,i,i]}
示例	六自由度: <pre>{"state":"current_joint_voltage","joint_voltage":[27500,28000,26800,26800,28900,30100]}</pre> 七自由度: <pre>{"state":"current_joint_voltage","joint_voltage":[27500,28000,26800,26800,28900,30100,31100]}</pre> 说明: 反馈关节当前电压, 六自由度机械臂关节 1~6 电压依次为 27.5V,28.0V,26.8V,26.8V,28.9V,30.1V, 七自由度机械臂关节 1~7 电压依次为 27.5V,28.0V,26.8V,26.8V,28.9V,30.1V,31.1V

## 2.10 机械臂配置-初始状态

### (1) 设置初始状态

功能描述	设置初始状态
参数说明	set_init_pose: 设置初始状态 init_pose: 初始状态位置, 精度 0.001 °
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i]} 七自由度: {s:s,s:[i,i,i,i,i,i,i]}
示例	六自由度: <pre>{"command":"set_init_pose","init_pose":[10000,0,20000,30000,0,20000]}</pre> 七自由度: <pre>{"command":"set_init_pose","init_pose":[10000,0,20000,30000,0,20000,20000]}</pre> 说明: 设置初始状态, 六自由度系机械臂初始状态位置[10°, 0°, 20°, 30°, 0°, 20°], 七自由度系机械臂初始状态位置[10°, 0°, 20°, 30°, 0°, 20°, 20°]
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败 <pre>{"command":"set_init_pose","init_pose":true}</pre>

### (2) 查询初始位置



功能描述	查询初始位置
参数说明	get_init_pose: 查询初始位置
命令格式	{s:s}
示例	{"command":"get_init_pose"}
	说明: 查询初始位置
返回值	见下表 (3)

### (3) 反馈初始位置

功能描述	反馈初始位置
参数说明	init_pose: 反馈初始位置 init_pose: 精度 0.001 °
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i]} 七自由度: {s:s,s:[i,i,i,i,i,i,i]}
示例	六自由度: {"state":"arm_init_pose","init_pose":[10000,0,20000,30000,0,20000]}
	七自由度: {"state":"arm_init_pose","init_pose":[10000,0,20000,30000,0,20000,20000]}
	说明: 反馈初始位置, 六自由度初始状态位置[10 ° ,0 ° ,20 ° ,30 ° ,0 ° ,20 ° ], 七自由度初始状态位置[10 ° ,0 ° ,20 ° ,30 ° ,0 ° ,20 ° ,20 ° ]

## 三、运动配置

### 3.1 运动配置-轨迹指令类

#### (1) MoveJ: 关节运动

功能描述	MoveJ: 关节运动
参数说明	movej: 关节运动 joint: 目标关节角度, 精度 0.001 ° v: 速度百分比例系数, 0~100



	<p>r: 交融半径, 精度 0.001m, 暂不支持交融, 默认 0</p> <p>trajectory_connect: 可选参数, 代表是否和下一条运动一起规划, 0 代表立即规划, 1 代表和下一条轨迹一起规划, 当为 1 时, 轨迹不会立即执行</p>
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i],s:i,s:i,s:i} , 七自由度: {s:s,s:[i,i,i,i,i,i],s:i,s:i,s:i}
示例	<p>六自由度:</p> <pre>{"command":"movej","joint":[10100,200,20300,30400,500,20600],"v":50,"r":0,"trajectory_connect":0}</pre> <p>七自由度:</p> <pre>{"command":"movej","joint":[10100,200,20300,30400,500,20600,20600],"v":50,"r":0,"trajectory_connect":0}</pre> <p>说明: 关节运动, 六自由度关节角度[10.1 °,0.2 °,20.3 °,30.4 °,0.5 °,20.6 °], 七自由度关节角度[10.1 °,0.2 °,20.3 °,30.4 °,0.5 °,20.6 °,20.6 °], 速度系数 50%, 交融半径: 0</p>
返回值	<p>格式: {s:s,s:b,s:i},true-到达目标位置, false-规划失败; 0:机械臂运动轨迹到位</p> <pre>{"state":"current_trajectory_state","trajectory_state":true,"device":0}</pre>

## (2) MoveL: 直线运动

功能描述	MoveL: 直线运动
参数说明	<p>movel: 直线运动</p> <p>pose: 目标位姿, 位置精度: 0.001mm, 姿态精度: 0.001rad</p> <p>v: 速度百分比例系数, 0~100</p> <p>r: 交融半径, 精度 0.001m, 暂不支持交融, 默认 0</p> <p>trajectory_connect: 可选参数, 代表是否和下一条运动一起规划, 0 代表立即规划, 1 代表和下一条轨迹一起规划, 当为 1 时, 轨迹不会立即执行</p>
命令格式	{s:s,s:[i,i,i,i,i,i],s:i,s:i,s:i}
示例	<pre>{"command":"movel","pose":[100000,200000,30000,400,500,600],"v":50,"r":0,"trajectory_connect":0}</pre>



	<p>说明：直线运动，</p> <p>目标位置：x: 0.1m, y:0.2m, z: 0.03m</p> <p>目标姿态：rx:0.4rad, ry:0.5rad, rz:0.6rad</p> <p>速度系数：50%，</p> <p>不交融</p>
返回值	<p>格式：{s:s,s:b,s:i},true-到达目标位置，false-规划失败；0:机械臂运动轨迹到位</p> <p>{"state":"current_trajectory_state","trajectory_state":true,"device":0}</p>
备注：	MOVL 指令也适用于目标位置不变，姿态变化

### (3) MoveC：圆弧运动

功能描述	MoveC：圆弧运动
参数说明	<p>movec：圆弧运动</p> <p>pose：位姿</p> <p>pose_via：中间点位姿，位置精度 0.001mm，姿态精度 0.001rad</p> <p>pose_to：目标位姿，位置精度 0.001mm，姿态精度 0.001rad</p> <p>v：速度百分比例系数，0~100</p> <p>r：交融半径，暂不支持轨迹交融，默认 0</p> <p>loop：循环圈数，默认 0</p> <p>trajectory_connect：可选参数，代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹不会立即执行</p>
命令格式	{s:s,s:{s:[i,i,i,i,i,i],s:[i,i,i,i,i,i]},s:i,s:i,s:i,s:i}
示例	<p>{"command":"movec","pose":{"pose_via":[100000,200000,30000,400,500,600],"pose_to":[200000,300000,30000,400,500,600]},"v":50,"r":0,"loop":0,"trajectory_connect":0}</p> <p>说明：圆弧运动，</p> <p>中间点位置：x: 0.1m, y:0.2m, z: 0.03m</p> <p>中间点姿态：rx:0.4rad, ry:0.5rad, rz:0.6rad</p> <p>终点位置：x: 0.2m, y:0.3m, z: 0.03m</p>





	终点姿态: rx:0.4rad, ry:0.5rad, rz:0.6rad 速度系数 50%, 不交融 不循环
返回值	格式: {s:s,s:b,s:i},true-到达目标位置, false-规划失败; 0:机械臂运动轨迹到位  {"state":"current_trajectory_state","trajectory_state":true,"device":0}

#### (4) 角度透传

功能描述	movej_canfd: 角度通过 CANFD 透传给机械臂, 不需控制器规划
参数说明	movej_canfd: 角度透传到 CANFD, 若指令正确, 机械臂立即执行  joint: 关节角度, 精度 0.001 °  备注: 透传效果和周期、轨迹是否平滑有关, 周期要求稳定, 防止出现较大波动, 基础系列 WIFI 和网口模式透传周期最快 20ms, USB 和 RS485 模式透传周期最快 10ms。高速网口的透传周期最快也可到 10ms, 不过在使用该高速网口前, 需要使用指令打开配置。另外 I 系列有线网口周期最快可达 2ms。
备注	用户使用该指令时请做好轨迹规划, 轨迹规划的平滑程度决定了机械臂的运行状态, 帧与帧之间关节的角度最大不能超过 10°, 并保证关节规划的速度不要超过 180°/s, 否则关节会不响应。
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i],s:b} 七自由度: {s:s,s:[i,i,i,i,i,i,i],s:b}
示例	六自由度: <pre>{"command":"movej_canfd","joint":[1000,0,20000,30000,0,20000],"follow":true,"expand":1000}</pre> 七自由度: <pre>{"command":"movej_canfd","joint":[1000,0,20000,30000,0,20000,20000],"follow":true,"expand":1000}</pre> 说明: 角度透传到 CANFD, 六自由度机械臂目标关节角度: [1°, 0°, 20°, 30°, 0°, 20°], 七自由度机械臂目标关节角度: [1°, 0°, 20°, 30°, 0°, 20°, 20°]。  follow 表示驱动器的运动跟随效果, true 为高跟随, false 为低跟随。



	<p>若使用高跟随，透传周期要求不超过 10ms。</p> <p>expand 为下发可选字段，如果存在通用扩展口轴，并需要进行透传，可使用此字段进行透传发送。</p>
返回值	<p>格式：{s:s,s: [i,i,i,i,i,i],s:i}</p> <p>六自由度：{"state":"joint_state","joint":[10,20,30,40,50,60], "arm_err":0}</p> <p>七自由度：{"state":"joint_state","joint":[10,20,30,40,50,60,70], "arm_err":0}</p> <p>关节精度：0.001 °</p> <p>arm_err：若为 0，则代表系统正常，指令正常运行；若为其他错误，则反馈相应错误代码，指令不执行。</p> <p>需要注意的是 I 系列机械臂不再提供返回值，通过 UDP 状态主动上报接口采集机械臂实时状态。</p>

## (5) 位姿透传

功能描述	movep_canfd：目标位姿透传给机械臂，不需控制器规划
参数说明	<p>movep_canfd：目标位姿透传到机械臂，控制器进行逆解后，若逆解存在并且逆解出的各角度与当前角度未有较大差值，则直接下发给关节执行，不再进行轨迹规划。</p> <p>适用于用户需要对位姿进行周期性调整的场景，如视觉伺服等。</p> <p>备注：透传效果和周期、轨迹是否平滑有关，周期要求稳定，防止出现较大波动，用户使用该指令时请做好轨迹规划，轨迹规划的平滑程度决定了机械臂的运行状态。基础系列 WIFI 和网口模式透传周期最快 20ms，USB 和 RS485 模式透传周期最快 10ms。高速网口的透传周期最快也可到 10ms，不过在使用该高速网口前，需要使用指令打开配置。另外 I 系列有线网口周期最快可达 2ms。</p>
命令格式	<p>位姿透传姿态欧拉角方式{s:s,s:[i,i,i,i,i,i],s:b}</p> <p>位姿透传姿态四元数方式{s:s,s:[i,i,i,i,i,i],s:b}</p>
示例	<p>位姿透传欧拉角方式</p> <pre>{"command":"movep_canfd","pose":[100000,200000,30000,400,500,600],"follow":true}</pre> <p>位姿透传四元数方式</p>



	<pre>{"command":"movep_canfd","pose_quat":[100000,200000,30000,400000,500000,600000,700000],"follow":true}</pre> <p>说明：pose：目标位姿，位置精度：0.001mm，姿态精度：0.001rad</p> <p>目标位置：x：0.1m，y：0.2m，z：0.03m</p> <p>目标姿态（欧拉角）：rx：0.4rad，ry：0.5rad，rz：0.6rad</p> <p>目标姿态（四元数）：w：0.4，x：0.5，y：0.6，z：0.7</p> <p>目标位姿为当前工具在当前工作和工具坐标系下的数值。</p> <p>follow 表示驱动器的运动跟随效果，true 为高跟随，false 为低跟随。</p> <p>若使用高跟随，透传周期要求不超过 10ms。</p>
返回值	<p>格式：{s:s,s: [i,i,i,i,i,i],s: [i,i,i,i,i,i],s:i}</p> <p>六自由度：</p> <pre>{"state":"pose_state","pose":[10,20,30,40,50,60],"joint":[10,20,30,40,50,60], "arm_err":0}</pre> <p>七自由度：</p> <pre>{"state":"pose_state","pose":[10,20,30,40,50,60],"joint":[10,20,30,40,50,60,70], "arm_err":0}</pre> <p>pose：当前位姿，位置精度：0.001mm，姿态精度：0.001rad；joint：当前关节角度，关节精度：0.001°；arm_err：若为 0，则代表系统正常，指令正常运行；若为其他错误，则反馈相应错误代码，指令不执行。</p> <p>需要注意的是 I 系列机械臂不再提供返回值，可通过 UDP 状态主动上报接口采集机械臂实时状态。</p>

## (6) MoveJ\_P：关节空间规划到目标位姿

功能描述	MoveJ_P：关节空间规划到目标位姿
参数说明	<p>movej_p：关节空间规划到目标位姿</p> <p>pose：目标位姿，位置精度：0.001mm，姿态精度：0.001rad</p> <p>v：速度百分比例系数，0~100</p> <p>r：交融半径，精度 0.001m，暂不支持交融，默认 0</p> <p>trajectory_connect：可选参数，代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹</p>



	不会立即执行
命令格式	{s:s,s:[i,i,i,i,i],s:i,s:i}
示例	<pre>{"command":"movej_p","pose":[100000,200000,30000,400,500,600],"v":50,"r":0,"trajectory_connect":0}</pre>
	说明：直线运动， 目标位置：x: 0.1m, y:0.2m, z: 0.03m 目标姿态：rx:0.4rad, ry:0.5rad, rz:0.6rad 速度系数 50%， 不交融
返回值	格式：{s:s,s:b,s:i},true-到达目标位置，false-规划失败；0:机械臂运动轨迹到位
	<pre>{"state":"current_trajectory_state","trajectory_state":true,"device":0}</pre>
备注：	目标位姿必须是机械臂当前工具坐标系相对于当前工作坐标系的位姿，用户在使用该指令前务必确保，否则目标位姿会出错！

## 3.2 运动配置-步进指令类

### (1) 关节步进

功能描述	关节步进
参数说明	set_joint_step：关节步进 joint_step：（1）步进关节号；（2）关节步进角度，单位：°，精度：0.001° v：速度百分比例系数，0~100
命令格式	{s:s,s:[i,i]}
示例	<pre>{"command":"set_joint_step","joint_step":[1,-10000],"v":30}</pre>
	说明：关节步进，关节 1 反方向步进 10 度，速度系数 30%
返回值	格式：{s:s,s:b,s:i},true-到达目标位置，false-规划失败；0:机械臂运动轨迹到位
	<pre>{"state":"current_trajectory_state","trajectory_state":true,"device":0}</pre>



## (2) 位置步进

功能描述	位置步进
参数说明	<p>set_pos_step: 位置步进</p> <p>step_type: 步进类型, x_step 为 X 轴方向, y_step 为 Y 轴方向, z_step 为 Z 轴方向。</p> <p>step: 步进距离单位: m, 精度: 0.001mm, 即 0.000001m</p> <p>v: 速度百分比例系数, 0~100</p>
命令格式	{s:s,s:s,s:i,s:i}
示例	<pre>{"command":"set_pos_step","step_type":"x_step","step":-500000,"v":30}</pre>
	说明: 位置步进, x 轴负方向步进 0.5m, 速度 30%
返回值	格式: {s:s,s:b,s:i}, true-到达目标位置, false-规划失败; 0:机械臂运动轨迹到位
	<pre>{"state":"current_trajectory_state","trajectory_state":true,"device":0}</pre>

## (3) 姿态步进

功能描述	姿态步进
参数说明	<p>set_ort_step: 姿态步进</p> <p>step_type: 步进方向, rx_step: 绕 X 轴旋转, ry_step: 绕 Y 旋转, rz_step: 绕 Z 轴旋转</p> <p>step: 步进弧度, 单位: rad, 精度 0.001rad</p> <p>v: 速度百分比例系数, 0~100</p>
命令格式	{s:s,s:s,s:i,s:i}
示例	<pre>{"command":"set_ort_step","step_type":"rx_step","step":-500,"v":30}</pre>
	说明: 姿态步进, x 轴负方向旋转 0.5rad, 速度 30%
返回值	格式: {s:s,s:b,s:i}, true-到达目标位置, false-规划失败; 0:机械臂运动轨迹到位
	<pre>{"state":"current_trajectory_state","trajectory_state":true,"device":0}</pre>



	0}
--	----

### 3.3 运动配置-运动指令类

#### (1) 轨迹急停

功能描述	轨迹急停
参数说明	set_arm_stop: 轨迹急停, 关节最快速度停止, 轨迹不可恢复
命令格式	{s:s}
示例	{"command":"set_arm_stop"}
	说明: 轨迹急停
返回值	格式: {s:s,s:b},true-设置成功, false-设置失败
	{"command":"set_arm_stop","arm_stop":true}

#### (2) 轨迹缓停

功能描述	轨迹缓停
参数说明	set_arm_slow_stop: 轨迹缓停, 在当前正在运行的轨迹上停止
命令格式	{s:s}
示例	{"command":"set_arm_slow_stop"}
	说明: 轨迹缓停
返回值	格式: {s:s,s:b},true-设置成功, false-设置失败
	{"command":"set_arm_slow_stop","arm_slow_stop":true}

#### (3) 轨迹暂停

功能描述	轨迹暂停
参数说明	set_arm_pause: 轨迹暂停, 停在轨迹上, 轨迹可恢复
命令格式	{s:s}
示例	{"command":"set_arm_pause"}
	说明: 轨迹暂停
返回值	格式: {s:s,s:b},true-设置成功, false-设置失败



	<code>{"command":"set_arm_pause","arm_pause":true}</code>
--	---

#### (4) 轨迹暂停后恢复

功能描述	轨迹暂停后恢复
参数说明	set_arm_continue：轨迹暂停后恢复
命令格式	{s:s}
示例	<code>{"command":"set_arm_continue"}</code>
	说明：轨迹暂停后恢复
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	<code>{"command":"set_arm_continue","arm_continue":true}</code>

#### (5) 清除当前轨迹

功能描述	清除当前轨迹，必须在暂停后使用！
参数说明	set_delete_current_trajectory：清除当前轨迹
命令格式	{s:s}
示例	<code>{"command":"set_delete_current_trajectory"}</code>
	说明：清除当前轨迹
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	<code>{"command":"set_arm_delete_current_trajectory","delete_current_trajectory":true}</code>

#### (6) 清除所有轨迹

功能描述	清除所有轨迹，必须在暂停后使用！
参数说明	set_arm_delete_trajectory：清除所有轨迹
命令格式	{s:s}
示例	<code>{"command":"set_arm_delete_trajectory"}</code>
	说明：清除所有轨迹
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	<code>{"command":"set_arm_delete_trajectory","arm_delete_trajectory":true}</code>



### (7) 查询当前规划类型

功能描述	查询当前规划类型
参数说明	get_arm_current_trajectory: 查询当前轨迹
命令格式	{s:s}
示例	{"command":"get_arm_current_trajectory"}
	说明: 查询当前轨迹
返回值	见表 3.5- (1)

## 3.4 运动配置-示教指令类

### (1) 关节示教

功能描述	关节示教
参数说明	set_joint_teach: 关节示教 teach_joint: 关节序号 direction: 方向, "pos": 正方向, "neg": 反方向 v: 速度系数
命令格式	{s:s,s:i,s:s,s:i}
示例	{"command":"set_joint_teach","teach_joint":1,"direction":"pos","v":50}
	说明: 关节 1 示教, 正方向, 速度 50%
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"set_joint_teach","joint_teach":true}

### (2) 位置示教

功能描述	位置示教
参数说明	set_pos_teach: 位置示教 teach_type: 坐标轴, "x", "y", "z" direction: 方向, "pos": 正方向, "neg": 反方向 v: 速度系数





命令格式	{s:s,s:s,s:s,s:i}
示例	{"command":"set_pos_teach","teach_type":"x","direction":"neg","v":50}
	说明：位置示教，x 轴负方向，速度 50%
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"set_pos_teach","pos_teach":true}

### (3) 姿态示教

功能描述	姿态示教
参数说明	set_ort_teach：姿态示教 teach_type：旋转所绕坐标轴，"rx"，"ry"，"rz" direction：方向，"pos"：正方向，"neg"：反方向 v：速度系数
命令格式	{s:s,s:s,s:s,s:i}
示例	{"command":"set_ort_teach","teach_type":"rx","direction":"neg","v":50}
	说明：姿态示教，rx 轴负方向，速度 50%
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"set_ort_teach","ort_teach":true}

### (4) 示教停止

功能描述	示教停止
参数说明	set_stop_teach：示教停止
命令格式	{s:s}
示例	{"command":"set_stop_teach"}
	说明：示教停止
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"set_stop_teach","stop_teach":true}

### (5) 设置示教参考坐标系



功能描述	设置示教参考坐标系
参数说明	frame_type: 0 代表工作坐标系, 1 代表工具坐标系
命令格式	{s:s,s:i}
示例	{"command":"set_teach_frame","frame_type":0}
	说明: 设置示教参考坐标系为工作坐标系
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"set_teach_frame","set_state":true}

#### (6) 获取示教参考坐标系

功能描述	获取示教参考坐标系
参数说明	get_teach_frame: 获取示教参考坐标系
命令格式	{s:s,s:i}
示例	{"command":"get_teach_frame"}
	说明: 获取示教运动参考坐标系
返回值	{"command":"get_teach_frame","frame_type":0}
	说明: frame_type: 0 代表工作坐标系, 1 代表工具坐标系

### 3.5 运动配置-轨迹反馈集

#### (1) 返回当前正在运行的轨迹

功能描述	返回当前正在运行的轨迹
参数说明	arm_current_trajectory: 返回当前正在运行的轨迹
命令格式	六自由度: {s:s,s:s,s:[i,i,i,i,i,i]} 七自由度: {s:s,s:s,s:[i,i,i,i,i,i]}
示例	六自由度: { "state":"arm_current_trajectory", "type":"movj", "data":[0,0,0,0,0,0,0]}
	七自由度: { "state":"arm_current_trajectory", "type":"movj", "data":[0,0,0,0,0,0,0]}



	说明：当前正在运行关节规划，数组内为当前关节角度，精度 0.001 °
	<pre>{"state":"arm_current_trajectory","type":"movl","data":[0,0,0,0,0,0]}</pre>
	说明：当前正在运行直线规划，数组内为当前末端位姿，位置精度：0.001mm，姿态精度：0.001rad
	<pre>{"state":"arm_current_trajectory","type":"movc","data":[0,0,0,0,0,0]}</pre>
	说明：当前正在运行圆弧规划，数组内为当前末端位姿，位置精度：0.001mm，姿态精度：0.001rad
	六自由度： <pre>{"state":"arm_current_trajectory","type":"movj","data":[0,0,0,0,0,0]}</pre>
	七自由度： <pre>{"state":"arm_current_trajectory","type":"movj","data":[0,0,0,0,0,0,0]}</pre>
	说明：当前无规划，数组内为当前关节角度，精度 0.001 °

## (2) 当前轨迹结束返回标志

功能描述	当前轨迹结束返回标志
参数说明	current_trajectory_state：当前轨迹结束返回标志 device: 0：关节 1：夹爪 2：灵巧手 3：升降机构 4：扩展关节 其他：保留
命令格式	{s:s,s:b,s:i}
示例	<pre>{"state":"current_trajectory_state","trajectory_state":true,"device":0}</pre>
	说明：当前轨迹到达目标



## 四、系统配置

### 4.1 系统配置-系统查询

#### (1) 查询控制器状态

功能描述	查询控制器状态
参数说明	get_controller_state: 查询控制器状态
命令格式	{s:s}
示例	{"command":"get_controller_state"}
	说明: 查询控制器状态
返回值	见表 4.2- (1)

### 4.2 系统配置-系统反馈

#### (1) 反馈控制器状态

功能描述	反馈控制器状态
参数说明	controller_state: 反馈控制器状态
命令格式	{s:s,s:i,s:i,s:i}
示例	{"state":"controller_state","voltage":24000,"current":1500,"temperature":42000,"err_flag":0}
	说明: 反馈控制器状态,电压: 24v, 电流: 1.5A,温度: 42℃, 控制器错误标志 0, 电压、电流和温度的精度均为 0.001

### 4.3 系统配置-系统指令

#### (1) 控制机械臂上电、断电

功能描述	控制机械臂上电、断电
参数说明	set_arm_power: 控制机械臂上电、断电 arm_power: 上电状态 1-上电 0-断电
命令格式	{s:s,s:i}
示例	{"command":"set_arm_power","arm_power":1}



	说明：控制机械臂上电
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	{"command":"set_arm_power","arm_power":true}

## (2) 读取机械臂电源状态

功能描述	读取机械臂电源状态
参数说明	get_arm_power_state：读取机械臂电源状态
命令格式	{s:s}
示例	{"command":"get_arm_power_state"} 说明：读取机械臂电源状态
返回值	格式：{s:s,s:i},1-上电状态，0 断电状态 {"state":"arm_power_state","power_state":1}

## (3) 读取软件版本号

功能描述	读取机械臂软件版本
参数说明	get_arm_software_version：读取机械臂软件版本
命令格式	{s:s}
示例	{"command":"get_arm_software_version"} 说明：读取机械臂软件版本
返回值	格式：{s:s,s:i,s:i,s:i,s:i,s:i,s:i},true-设置成功，false-设置失败 {"Real-time_Kernal2":515,"state":"arm_software_version","Ctrl_version": 7014147,"Plan_version":7013891,"ctrlM4_build_info":"202212282001","Real-time_Kernal1":515,"ctrlM7_build_info":"202212282001", "Plan_build_info": "202212282001", "Product_version": "RM65-BI"} 7013129：转换成 uint32_t 类型的 16 进制数据，0x6B0309 6B-代表 RM65-B 型号 6D-代表 RM65-ZF 型号 6F-代表 RM65-SF 型号 0309-代表软件版本号为 V3.0.9 Product_version：机械臂型号，仅 I 系列机械臂支持

## (4) 读取控制器的累计运行时间



功能描述	读取控制器自出厂以来，累计的运行时间
参数说明	get_system_runtime：读取控制器累计的运行时间
命令格式	{s:s}
示例	{"command":"get_system_runtime"}
	说明：读取系统运行时间
返回值	格式：{s:s,s:i,s:i,s:i,s:i},
	{"command":"get_system_runtime","day":0,"hour":0,"min":0,"sec":0}若系统正常，则返回运行时间

#### (5) 清零控制器的累计运行时间

功能描述	清零控制器自出厂以来，累计的运行时间
参数说明	clear_system_runtime：清零控制器累计的运行时间
命令格式	{s:s}
示例	{"command":"clear_system_runtime"}
	说明：清零系统运行时间
返回值	格式：{s:s,s:b}
	{"command":"clear_system_runtime","clear_state":true} true-清除成功，false-清除失败

#### (6) 读取关节的累计转动角度

功能描述	读取各关节自出厂以来，累计的转动角度
参数说明	get_joint_odom：读取各关节的累计转动角度
命令格式	{s:s}
示例	{"command":"get_joint_odom"}
	说明：读取关节的累计转动角度
返回值	格式：六自由度：{s:s,s:[i,i,i,i,i,i]}, 七自由度：{s:s,s:[i,i,i,i,i,i,i]}
	六自由度 {"command":"get_joint_odom","odom":[1000,2000,3000,4000,5000,6000]}



	<p>七自由度</p> <pre>{"command":"get_joint_odom","odom":[1000,2000,3000,4000,5000,6000,7000]}</pre> <p>若指令正确，返回各关节累计的转动角度</p>
--	---

#### (7) 清零关节的累计转动角度

功能描述	清零各关节自出厂以来，累计的转动角度
参数说明	clear_joint_odom：清零关节累计转动的角度
命令格式	{s:s}
示例	<pre>{"command":"clear_joint_odom"}</pre>
	说明：清零关节累计转动的角度
返回值	格式：{s:s,s:b}
	<pre>{"command":"clear_joint_odom","clear_state":true}</pre> <p>true-清除成功，false-清除失败</p>

#### (8) 高速网口的配置（基础系列）

功能描述	<p>控制器面板有 2 个网口，左侧为高速网口，默认为关闭状态，需要通过指令打开；右侧靠近面板边缘的为普通网口，用户无需配置，可直接使用。</p> <p>备注：高速网口的 IP 地址为：192.168.1.18，端口号为：8080，可修改。</p>
参数说明	set_high_speed_eth：配置高速网口
命令格式	{s:s,s:i}
示例	<pre>{"command":"set_high_speed_eth", "mode":0}</pre>
	<p>说明：mode</p> <p>关闭高速网口</p> <p>打开高速网口，设置成功后，机械臂控制器蜂鸣器会提示，然后用户将网线插在高速网口上，重启控制器，初始化成功后用户即可使用。该配置信息会保存在控制器内，重启后不会丢失。</p> <p>备注：控制器启动后，会有自动查询上位机 MAC 地址，会一直等待上位机与控制网口的物理连接。</p>



返回值	格式: {s:s,s:b}
	{"command":"set_high_speed_eth","set_state":true} true-设置成功, false-设置失败

### (9) 参数保存 (基础系列)

功能描述	保存所有参数
参数说明	save_device_info_all: 保存所有参数
命令格式	{s:s}
示例	{"command":"save_device_info_all"}
	说明: 保存所有参数
返回值	格式: {s:s,s:i} true:保存成功 false:保存失败
	{"command":"save_device_info_all","model":true} true-设置成功, false-设置失败

### (10) 配置有线网 IP 地址 (I 系列)

功能描述	配置有线网卡 IP 地址
参数说明	set_NetIP:配置有线网口 IP 地址
命令格式	{s:s, s:s}
示例	{"command":"set_NetIP","ip":"192.168.1.20"}
说明	说明: 配置有线网口 IP 地址为 192.168.1.20, 指令下发后, 若成功设置机械臂蜂鸣器会响一声, 然后手动重启机械臂
返回值	{"command":"set_NetIP","status":true} //IP 地址设置成功
	{"command":"set_NetIP","status":false} //IP 地址设置失败

### (11) 清除系统错误

功能描述	清除系统错误
参数说明	clear_system_err 清除系统错误
命令格式	{s:s}





示例	<code>{"command":"clear_system_err"}</code>
说明	用于手动清除系统错误，如果不手动清除，错误将一直保留，直到新的运动指令下发则会自动清除。
返回值	<code>{"command":"clear_system_err","clear_state":true}</code> //清除系统错误成功 <code>{"command":"clear_system_err","clear_state":false}</code> //清除系统错误失败

## (12) 读取软件信息

功能描述	读取机械臂软件信息
参数说明	get_arm_software_info：读取机械臂软件信息
命令格式	<code>{s:s}</code>
示例	<code>{"command":"get_arm_software_info"}</code> 说明：读取机械臂软件信息
返回值	<code>{"Product_version":"RM65-BI","algorithm_info":{"version":"1.2.3"},"command":"arm_software_info","ctrl_info":{"build_time":"2023/12/13 20:58:10","version":"V1.4.6"},"dynamic_info":{"model_version":"1"},"plan_info":{"build_time":"2023/12/13 20:58:24","version":"V1.4.6"}}</code> 参数说明： plan_info：plan 层软件信息，version 代表版本号、build_time 代表编译时间 ctrl_info：ctrl 层软件信息，version 代表版本号、build_time 代表编译时间 algorithm_info：算法库信息，version 代表版本号 Product_version：机械臂型号，仅 I 系列机械臂支持 dynamic_info：动力学版本，model_version 动力学模型版本号

## 4.4 系统配置-配置通讯内容

机械臂控制器可通过网口、WIFI、RS232-USB 接口和 RS485 接口与用户通信，用户使用时无需切换，可使用上述任一接口，控制器收到指令后，若指令



格式正确，则会通过相同的接口反馈数据。

### (1) 配置 wifiAP 内容

功能描述	配置 wifiAP 内容，无返回，设置成功后蜂鸣器响，手动重启控制器进入 WIFIAP 模式。
参数说明	set_wifi_ap：配置 wifiAP 内容
命令格式	{s:s,s:s,s:s}
示例	<pre>{"command":"set_wifi_ap","wifi_name":"robot","password":"12345678"}</pre> <p>说明：配置 wifiAP 内容，wifi 名称：robot，连接密码：12345678</p>

### (2) 配置 wifiSTA 内容

功能描述	配置 wifiSTA 内容，无返回，设置成功后蜂鸣器响，手动重启控制器进入 WIFISTA 模式。
参数说明	set_wifi_sta：配置 wifiSTA 内容
命令格式	{s:s,s:s,s:s}
示例	<pre>{"command":"set_wifi_sta","router_name":"robot","password":"12345678"}</pre> <p>说明：配置 wifiSTA 内容，目标路由器名称：robot，路由器密码：12345678</p>

### (3) 配置 USB 内容（基础系列）

功能描述	配置 UART-USB 波特率，无返回
参数说明	set_usb：配置 USB 波特率，最高 460800
命令格式	{s:s,s:i}
示例	<pre>{"command":"set_usb","baudrate":460800}</pre> <p>说明：配置 USB 波特率为 460800</p> <p>波特率可选范围：9600，19200，38400，115200 和 460800，若用户设置其他数据，控制器会默认按照 460800 处理。</p> <p>该指令下发后控制器会记录当前波特率，断电重启后仍会使用该波特率对外通信。</p>



#### (4) 配置 RS485 内容

功能描述	配置 RS485 接口波特率，无返回
参数说明	set_RS485:配置 RS485 接口波特率，最高 460800
命令格式	{s:s, s:i}
示例	{"command":"set_RS485","baudrate":460800}
说明	<p>说明：配置 RS485 波特率为 460800</p> <p>波特率可选范围：9600，19200，38400，115200 和 460800，若用户设置其他数据，控制器会默认按照 460800 处理。</p> <p>该指令下发后，若 Modbus 模式为打开状态，则会自动关闭，同时控制器会记录当前波特率，断电重启后仍会使用该波特率对外通信。</p>

#### (5) 配置高速网口 IP（基础系列）

功能描述	设置高速网口网络配置
参数说明	set_high_ethernet 配置高速网口网络 IP
命令格式	{s:s,s:s,s:s,s:ss}
示例	<pre>{"command":"set_high_ethernet","ip":"192.168.1.18","mask":"255.255.255.0","gateway":"192.168.1.1"}</pre> <p>说明：</p> <p>ip：网络地址</p> <p>mask：子网掩码</p> <p>gateway：网关</p>
返回值	<p>格式：{s:s,s:b}</p> <pre>{"command": "set_high_ethernet", "state": true}</pre> <p>true-设置成功，false-设置失败</p>

#### (6) 查询高速网口 IP（基础系列）

功能描述	获取高速网口网络配置
参数说明	get_high_ethernet 获取高速网口网络配置



命令格式	{s:s}
示例	{"command":"get_high_ethernet"}
返回值	<p><b>格式：</b> {s:s,s:s,s:s,s:s,s}</p> <pre>{"command": "get_high_ethernet","mask": "255.255.255.0","ip": "192.168.1.20", "gateway": "192.168.1.1","mac": "11:22:33:44:55:66"}</pre> <p>ip: 网络地址 mask: 子网掩码 gateway: 网关 mac:mac 地址</p>

#### (7) 查询有线网卡网络信息 (I 系列)

功能描述	查询有线网卡网络信息
参数说明	get_wired_net: 获取有线网卡信息, 未连接有线网卡则会返回无效数据
命令格式	{s:s}
示例	{"command":"get_wired_net"}
说明	说明: 查询有线网卡网络信息
返回值	<pre>{"command":"get_wired_net","mask": "255.255.255.0","ip": "192.168.1.18","mac": "11:22:33:44:55:66"}</pre> <p>ip: 网络地址 mask: 子网掩码 mac: mac 地址</p>

#### (8) 查询无线网卡网络信息 (I 系列)

功能描述	查询无线网卡网络信息
参数说明	get_wifi_net: 查询无线网卡网络信息
命令格式	{s:s}
示例	{"command":"get_wifi_net"}
说明	说明: 查询无线网卡网络信息



返回值	<pre>{"command": "get_wifi_net", "mode": "ap", "mask": "255.255.255.0", "ip": "192.168.1.18", "mac": "11:22:33:44:55:66", "ssid": "realman_ap", "password": "12345678", "channel": 6}</pre> <p>ip: 网络地址</p> <p>mask: 子网掩码</p> <p>mac: mac 地址</p> <p>mode: ap 代表热点模式, sta 代表联网模式</p> <p>ssid: 设置或默认的 wifi 名称</p> <p>password: 设置或默认的 wifi 密码</p> <p>channel: 如果是 AP 模式, 则存在此字段, 标识 wifi 热点的物理信道号</p>
-----	---

### (9) 恢复网络设置 (I 系列)

功能描述	恢复网络出厂设置
参数说明	set_net_default: 设置网络为出厂设置
命令格式	{s:s}
示例	<pre>{"command": "set_net_default"}</pre>
返回值	<p>配置成功:</p> <pre>{"command": "set_net_default", "net_default_state": true}</pre> <p>配置失败:</p> <pre>{"command": "set_net_default", "net_default_state": false}</pre>
说明	<p>说明: 网络恢复默认设置</p> <p>设置成功后, 手动重新启动后生效</p>

### (10) 关闭 wifi 功能 (I 系列)

功能描述	配置关闭 wifi 功能, 需要重启后生效
参数说明	set_wifi_close: 配置关闭 wifi 功能
命令格式	{s:s}
示例	<pre>{"command": "set_wifi_close"}</pre>
返回值	配置成功:



	<pre>{"command": "set_wifi_close", "wifi_close_state": true}</pre> <p>配置失败:</p> <pre>{"command": "set_wifi_close", "wifi_close_state": false}</pre>
--	---

## 4.5 查询机械臂状态信息

### (1) 查询机械臂关节角度

功能描述	查询机械臂关节角度
参数说明	get_joint_degree: 获取机械臂角度信息
命令格式	{s:s}
示例	<pre>{"command": "get_joint_degree"}</pre>
	说明: 查询机械臂关节角度
返回值	如下表 4.5- (2) 所示

### (2) 反馈机械臂关节角度

功能描述	反馈机械臂关节角度
参数说明	joint_degree: 机械臂关节角度
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i]} 七自由度: {s:s,s:[i,i,i,i,i,i,i]}
示例	六自由度: <pre>{"state": "joint_degree", "joint": [10,20,30,40,50,60]}</pre> 七自由度: <pre>{"state": "joint_degree", "joint": [10,20,30,40,50,60,70]}</pre>
	<pre>{"state": "joint_degree", "joint": [10,20,30,40,50,60]}</pre> 关节精度: 0.001 °

### (3) 一次性查询机械臂所有状态信息

功能描述	一次性查询机械臂所有信息
参数说明	get_arm_all_state: 获取机械臂所有信息
命令格式	{s:s}
示例	<pre>{"command": "get_arm_all_state"}</pre>
	说明: 查询所有状态信息



返回值	如下表 4.5-（4）所示
-----	---------------

#### （4）反馈所有状态信息

功能描述	反馈机械臂所有信息
参数说明	all_state：机械臂所有信息
命令格式	六自由度： {s:s,s:{s:[i,i,i,i,i,i],s:[i,i,i,i,i,i],s:[i,i,i,i,i,i],s:[i,i,i,i,i,i],s:[i,i,i,i,i,i],s:i,s:i}} 七自由度： {s:s,s:{s:[i,i,i,i,i,i,i],s:[i,i,i,i,i,i,i],s:[i,i,i,i,i,i,i],s:[i,i,i,i,i,i,i],s:[i,i,i,i,i,i,i],s:i,s:i}} 温度精度：0.001℃ 电流精度：0.001mA 电压精度：0.001V err_flag:关节错误代码 sys_err:机械臂错误代码
示例	<pre>{"command":"get_arm_all_state"}</pre> <pre>六自由度：{"state":"arm_all_state","all_state":{"temperature": [21, 22,23,24,25,26],"current":[11,12,13,14,15,16],"voltage":[31,32,33,34,35,36],"err_flag":[1,2,3,4,5,6],"en_flag":[1,1,1,1,1,1],"sys_err":0}}</pre> <pre>七自由度：{"state":"arm_all_state","all_state":{"temperature": [21, 22,23,24,25,26,27],"current":[11,12,13,14,15,16,17],"voltage":[31,32,33,34,35,36,37],"err_flag":[1,2,3,4,5,6,7],"en_flag":[1,1,1,1,1,1,1],"sys_err":0}}</pre>

#### （5）查询规划计数

功能描述	查询轨迹规划计数
参数说明	get_arm_plan_num：获取机械臂轨迹规划计数
命令格式	{s:s}
示例	<pre>{"command":"get_arm_plan_num"}</pre> 说明：查询机械臂轨迹规划计数
返回值	如下表 4.5-（6）所示



## (6) 反馈机械臂轨迹规划计数

功能描述	反馈机械臂轨迹规划计数
参数说明	plan_num: 机械臂轨迹规划计数 loop_num: 存在循环指令的行数, 没有循环指令则返回空 loop_cont: 循环指令行数对应的运行次数, 没有循环指令则返回空
命令格式	{s:s,s:i,s:[i,i...],s:[i,i...]}
示例	<pre>{"command":"get_arm_plan_num"} {"loop_cont":[],"loop_num":[],"plan_num":3,"state":"arm_plan_num"}</pre> <p>规划完成第三段轨迹, 轨迹规划中不包含循环指令</p>

## (7) 查询控制器 RS485 模式 (I 系列)

功能描述	查询控制器 RS485 模式
参数说明	get_controller_RS485_mode: 查询控制器 RS485 模式
命令格式	{s:s}
示例	<pre>{"command":"get_controller_RS485_mode"}</pre> <p>说明: 查询控制器 RS485 模式</p>
返回值	<p>格式: {s:s,s:i,s:i,s:i}</p> <p>controller_RS485_mode: 0 代表默认 RS485 串行通讯, 1 代表 modbus-RTU 主站模式, 2-代表 modbus-RTU 从站模式;</p> <p>baudrate: 波特率;</p> <p>modbus_timeout: modbus 协议超时时间, 单位 100ms, 仅在 modbus-RTU 模式下提供此字段</p> <p>示例</p> <pre>{"command":"get_controller_RS485_mode","controller_RS485_mode":1,"baudrate":460800,"modbus_timeout":1}</pre>

## (8) 查询工具端 RS485 模式 (I 系列)

功能描述	查询工具端 RS485 模式
------	----------------





参数说明	get_tool_RS485_mode: 查询工具端 RS485 模式
命令格式	{s:s}
示例	{"command":"get_tool_RS485_mode"}
	说明: 查询工具端 RS485 模式
返回值	格式: {s:s,s:i,s:i,s:i}
	格式: {s:s,s:b},tool_RS485_mode: 0 代表默认 RS485 串行通讯 1 代表 modbus-RTU 主站模式; baudrate: 波特率; modbus_timeout: modbus 协议超时时间, 单位 100ms, 仅在 modbus-RTU 模式下提供此字段
	示例 {"command":"get_tool_RS485_mode","tool_RS485_mode":0,"baudrate":460800,"modbus_timeout":1}

## 4.6 控制器 IO 配置及获取

机械臂具有 IO 端口, 基础系列数量和分类如下所示:

数字输出: DO	4 路, 可配置为 0~12V
数字输入: DI	3 路, 可配置为 0~12V
模拟输出: AO	4 路, 输出电压 0~10V
模拟输入: AI	4 路, 输入电压 0~10V

I 系列数量和分类如下所示:

数字 IO: DO/DI 复用	4 路, 可配置为 0~24V
-----------------	-----------------

### (1) 设置数字 IO 模式 (I 系列)

功能描述	设置数字 IO 模式
参数说明	set_IO_mode: 设置数字 IO 模式
命令格式	{s:s,s:i,s:i}
示例	{"command":"set_IO_mode","IO_Num":1,"IO_mode":0}
	说明: "IO_Num": IO 端口号, 范围: 1~4



	"IO_mode": 模式，0-通用输入模式，1-通用输出模式、2-输入开始功能复用模式、3-输入暂停功能复用模式、4-输入继续功能复用模式、5-输入急停功能复用模式、6-输入进入电流环拖动复用模式、7-输入进入力只动位置拖动模式（六维力版本可配置）、8-输入进入力只动姿态拖动模式（六维力版本可配置）、9-输入进入力位姿结合拖动复用模式（六维力版本可配置）、10-输入外部轴最大软限位复用模式（外部轴模式可配置）、11-输入外部轴最小软限位复用模式（外部轴模式可配置）。
返回值	{"command":"set_IO_mode","set_state":true}配置成功 {"command":"set_IO_mode","set_state":false}配置失败

### (2) 设置数字 IO 输出状态

功能描述	设置数字 IO 输出
参数说明	set_DO_state: 设置数字 IO 输出
命令格式	{s:s,s:i,s:i}
示例	{"command":"set_DO_state","IO_Num":1,"state":1} 说明: " IO_Num" : IO 端口号, 范围: 1~4 " state " : IO 状态, 1-输出高, 0-输出低
返回值	{"command":"set_DO_state","set_state":true}配置成功 {"command":"set_DO_state","set_state":false}配置失败

### (3) 获取数字 IO 状态 (I 系列)

功能描述	获取数字 IO 输出状态
参数说明	get_IO_state: 获取数字 IO 输出
命令格式	{s:s,s:i}
示例	{"command":"get_IO_state","IO_Num":1} 说明: " IO_Num" : IO 端口号, 范围: 1~4
返回值	{"state":"IO_state","IO_Num":1,"IO_Mode":1,"IO_state":1} "state" : IO 状态 "IO_Mode" : 0-通用输入模式, 1-通用输出模式、2-输入开始功能复用模式、3-输入暂停功能复用模式、4-输入继续功能复用模式、



	5-输入急停功能复用模式、6-输入进入电流环拖动复用模式、7-输入进入力只动位置拖动模式、8-输入进入力只动态拖动模式、9-输入进入力位姿结合拖动复用模式、10-输入外部轴最大软限位复用模式、11-输入外部轴最小软限位复用模式
--	---

#### (4) 获取数字 IO 输出状态 (基础系列)

功能描述	获取数字 IO 输出状态
参数说明	get_DO_state: 获取数字 IO 输出
命令格式	{s:s,s:i}
示例	{"command":"get_DO_state","IO_Num":1}
	说明: “IO_Num”: IO 端口号, 范围: 1~4
返回值	{"state":"DO_state","IO_Num":1,"IO_state":1} “state”: IO 状态, 1-输出高, 0-输出低

#### (5) 获取数字 IO 输入状态 (基础系列)

功能描述	获取数字 IO 输入
参数说明	get_DI_state: 获取数字 IO 输入状态
命令格式	{s:s,s:i}
示例	{"command":"get_DI_state","IO_Num":1}
	说明: “IO_Num”: IO 端口号, 范围: 1~3
返回值	{"state":"DI_state","IO_Num":1, “IO_state ”:1} “state”: IO 状态, 1-输入高, 0-输入低

#### (6) 设置模拟 IO 输出状态 (基础系列)

功能描述	设置模拟 IO 输出
参数说明	set_AO_state: 设置模拟 IO 输出
命令格式	{s:s,s:i,s:i}
示例	{"command":"set_AO_state","IO_Num":1,"voltage":1000}
	说明: “IO_Num”: IO 端口号, 范围: 1~4 “voltage”: IO 输出电压, 分辨率 0.001V, 范围: 0~10000, 代



	表输出电压 0v~10v
返回值	<pre>{"command":"set_AO_state","state":true}</pre> 配置成功 <pre>{"command":"set_AO_state","state":false}</pre> 配置失败

#### (7) 获取模拟 IO 输出状态（基础系列）

功能描述	获取模拟 IO 输出
参数说明	get_AO_state：获取模拟 IO 输出
命令格式	{s:s,s:i}
示例	<pre>{"command":"get_AO_state","IO_Num":1}</pre> 说明：“IO_Num”：IO 端口号，范围：1~4
返回值	<pre>{"state":"AO_state", "IO_Num":1,"voltage":1000}</pre> “voltage”：IO 输出电压，分辨率 0.001V，范围：0~10000，代表输出电压 0v~10v

#### (8) 获取模拟 IO 输入状态（基础系列）

功能描述	获取模拟 IO 输入
参数说明	get_AI_state：获取模拟 IO 输入状态
命令格式	{s:s,s:i,s:i}
示例	<pre>{"command":"get_AI_state","IO_Num":1}</pre> 说明：“IO_Num”：IO 端口号，范围：1~4
返回值	<pre>{"state":"AI_state","IO_Num":1,"voltage":1000}</pre> “voltage”：IO 输入电压，分辨率 0.001V，范围：0~10000，代表输入电压 0v~10v

#### (9) 获取所有 IO 输入状态

功能描述	获取所有 IO 输入
参数说明	get_IO_input：获取所有 IO 输入状态
命令格式	{s:s}
示例	<pre>{"command":"get_IO_input"}</pre>
返回值	<pre>{"state":"IO_input_state","DI":[1,1,1],"AI":[1000,2000,3000,4000]}</pre>



	0}} “DI”：数字输入状态，1-高，0-低 “AI”：模拟输入电压，精度 0.001V，如：1000，代表 1V
--	--

#### (10) 获取所有 IO 输出状态

功能描述	获取所有 IO 输出
参数说明	get_io_output：获取所有 IO 输出状态
命令格式	{s:s}
示例	{"command":"get_io_output"}
返回值	{"state":"IO_output_state","DO":[1,1,1,1],"AO":[1000,2000,3000,3000]} “DO”：数字输出状态，1-高，0-低 “AO”：模拟输出电压，精度 0.001V，如：1000，代表 1V

#### (11) 设置电源输出 (I 系列)

功能描述	设置电源输出
参数说明	set_voltage：设置电源输出
命令格式	{s:s,s:i}
示例	{"command":"set_voltage","voltage_type":2}
说明：	“voltage_type”：电源输出类型，范围：0~3 0-0V，2-12V，3-24V
返回值	{"command":"set_voltage","state":true}配置成功 {"command":"set_voltage","state":false}配置失败

#### (12) 获取电源输出 (I 系列)

功能描述	获取控制器电源输出类型
参数说明	get_voltage：获取电源输出
命令格式	{s:s}
示例	{"command":"get_voltage"}
返回值	{"command":"voltage_state", "voltage_type":2}



	“voltage_type”：电源输出类型，范围：0~3 0-0V，2-12V，3-24V
--	--

## 4.7 末端工具 IO 控制

机械臂末端工具端具有 IO 端口，数量和分类如下所示：

电源输出	1 路，可配置为 0V/5V/12V/24V
数字 IO	2 路，输入输出可配置 输入：参考电平 12V~24V 输出：5~24V，与输出电压一致
通讯接口	1 路，可配置为 RS485

### (1) 设置工具端数字 IO 输出状态

功能描述	设置工具端数字 IO 输出
参数说明	set_tool_DO_state：设置数字 IO 输出
命令格式	{s:s,s:i,s:i}
示例	<pre>{"command":"set_tool_DO_state","IO_Num":1,"state":1}</pre> <p>说明：“IO_Num”：IO 端口号，范围：1~2 “state”：IO 状态，1-输出高，0-输出低</p>
返回值	<pre>{"command":"set_tool_DO_state","set_state":true}</pre> 配置成功 <pre>{"command":"set_tool_DO_state","set_state":false}</pre> 配置失败
备注	数字 IO 为输入输出可复用模式，当下发输出指令后，末端数字 IO 会自动变为输出模式

### (2) 设置工具端数字 IO 模式

功能描述	设置数字 IO 模式
参数说明	set_tool_IO_mode：设置数字 IO 模式
命令格式	{s:s,s:i,s:i}
示例	<pre>{"command":"set_tool_IO_mode","IO_Num":1,"state":0}</pre> <p>说明：“IO_Num”：IO 端口号，范围：1~2</p>



	“state”：模式，0-输入状态，1-输出状态
返回值	<pre>{"command":"set_tool_IO_mode","set_state":true}</pre> 配置成功 <pre>{"command":"set_tool_IO_mode","set_state":false}</pre> 配置失败

### (3) 获取工具端数字 IO 状态

功能描述	获取数字 IO 状态
参数说明	get_tool_IO_state：获取数字 IO 状态
命令格式	{s:s}
示例	<pre>{"command":"get_tool_IO_state"}</pre>
返回值	<pre>{"state":"tool_IO_state","IO_Mode":[0,1],"IO_State":[0,1]}</pre> 说明：“IO_Mode”：0-输入模式，1-输出模式 “IO_State”：0-低，1-高

### (4) 设置工具端电源输出

功能描述	设置电源输出
参数说明	set_tool_voltage：设置电源输出
命令格式	{s:s,s:i}
示例	<pre>{"command":"set_tool_voltage","voltage_type":1}</pre>
说明：	“voltage_type”：电源输出类型，范围：0~3 0-0V，1-5V，2-12V，3-24V，电源输出设置为 5V 时，工具端的 IO 暂不支持输入输出功能
返回值	<pre>{"command":"set_tool_voltage","state":true}</pre> 配置成功 <pre>{"command":"set_tool_voltage","state":false}</pre> 配置失败

### (5) 获取工具端电源输出

功能描述	获取电源输出类型
参数说明	get_tool_voltage：获取电源输出
命令格式	{s:s}
示例	<pre>{"command":"get_tool_voltage"}</pre>



返回值	<pre>{"state":"tool_voltage_state", "voltage_type":1}</pre> <p>“voltage_type”：电源输出类型，范围：0~3 0-0V，1-5V，2-12V，3-24V</p>
-----	---

## 4.8 末端工具—手爪控制（选配）

睿尔曼机械臂末端配备了因时机器人公司的 EG2-4C2 手爪，为了便于用户操作手爪，机械臂控制器对用户开放了手爪的控制协议（手爪控制协议与末端 modbus 功能互斥），如下所示：

### （1）设置手爪行程

功能描述	设置手爪行程，即手爪开口的最大值和最小值，设置成功后会自动保存，手爪断电不丢失
参数说明	set_gripper_route：设置手爪行程
命令格式	{s:s,s:i,s:i}
示例	<pre>{"command":"set_gripper_route","min":70,"max":500}</pre>
说明	“min”：手爪开口最小值，范围：0~1000，无单位量纲 “max”：手爪开口最大值，范围：0~1000，无单位量纲
返回值	<pre>{"command":"set_gripper_route","state":true}</pre> 配置成功 <pre>{"command":"set_gripper_route","state":false}</pre> 配置失败

### （2）松开手爪

功能描述	松开手爪，即手爪以指定的速度运动到开口最大处
参数说明	set_gripper_release：设置手爪松开
命令格式	{s:s,s:i,s:b}
示例	<pre>{"command":"set_gripper_release","speed":500,"block":true}</pre>
说明	“speed”：手爪松开速度，范围 1~1000，无单位量纲 “block”：true 表示阻塞模式，false 表示非阻塞模式
返回值	该指令不论是否为阻塞模式均会返回 <pre>{"command":"set_gripper","state":true}</pre> 手爪松开成功





	<pre>{"command":"set_gripper","state":false}</pre> 手爪松开失败 该指令为阻塞模式下，运动到指定位置的上报信息 <pre>{"state":"current_trajectory_state","trajectory_state":true,"device":1}</pre>
--	---

### (3) 手爪力控夹取

功能描述	手爪力控夹取，手爪以设定的速度和力夹取，当夹持力超过设定的力阈值后，停止夹取
参数说明	set_gripper_pick：设置手爪力矩夹取
命令格式	{s:s,s:i,s:i,s:b}
示例	<pre>{"command":"set_gripper_pick","speed":500,"force":200,"block":true}</pre>
说明	“speed”：手爪夹取速度，范围 1~1000，无单位量纲 “force”：力控阈值，范围：50~1000，无单位量纲 “block”：true 表示阻塞模式，false 表示非阻塞模式
返回值	该指令不论是否为阻塞模式均会返回 <pre>{"command":"set_gripper","state":true}</pre> 手爪松开成功 <pre>{"command":"set_gripper","state":false}</pre> 手爪松开失败 该指令为阻塞模式下，运动到指定位置的上报信息 <pre>{"state":"current_trajectory_state","trajectory_state":true,"device":1}</pre>

### (4) 手爪持续力控夹取

功能描述	手爪力控夹取，手爪以设定的速度和力夹取，当夹持力超过设定的力阈值后，停止夹取；当夹持力再次小于力矩阈值时，手爪再次夹取，直至夹持力超过力控阈值。
参数说明	set_gripper_pick_on：设置手爪力控夹取
命令格式	{s:s,s:i,s:i,s:b}
示例	<pre>{"command":"set_gripper_pick_on","speed":500,"force":200,"block":true}</pre>
说明	“speed”：手爪夹取速度，范围 1~1000，无单位量纲



	<p>“force” :力控阈值，范围：50~1000，无单位量纲</p> <p>“block” :true 表示阻塞模式，false 表示非阻塞模式</p>
返回值	<p>该指令不论是否为阻塞模式均会返回</p> <p>{“command”:“set_gripper”,“state”:true}手爪松开成功</p> <p>{“command”:“set_gripper”,“state”:false}手爪松开失败</p> <p>该指令为阻塞模式下，运动到指定位置的上报信息</p> <p>{“state”:“current_trajectory_state”,“trajectory_state”:true,“device”:1}</p>

### (5) 手爪到达指定位置

功能描述	手爪到达指定位置，当当前开口小于指定开口时，手爪以指定速度松开到指定开口位置；当当前开口大于指定开口时，手爪以指定速度和力矩闭合往指定开口处闭合，当夹持力超过力矩阈值或者达到指定位置后，手爪停止。
参数说明	set_gripper_position：设置手爪达到指定位置
命令格式	{s:s,s:i,s:b}
示例	{“command”:“set_gripper_position”,“position”:500,“block”:true}
说明	<p>“position”：手爪开口位置，范围：1~1000，无单位量纲</p> <p>“block” :true 表示阻塞模式，false 表示非阻塞模式</p>
返回值	<p>该指令不论是否为阻塞模式均会返回</p> <p>{“command”:“set_gripper”,“state”:true}手爪松开成功</p> <p>{“command”:“set_gripper”,“state”:false}手爪松开失败</p> <p>该指令为阻塞模式下，运动到指定位置的上报信息</p> <p>{“state”:“current_trajectory_state”,“trajectory_state”:true,“device”:1}</p>

### (6) 查询夹爪状态

功能描述	查询夹爪状态
参数说明	get_gripper_state：查询夹爪状态
命令格式	{s:s}
示例	{“command”:“get_gripper_state”}



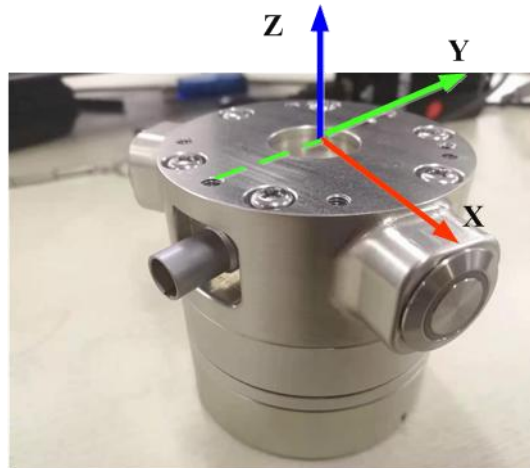
	<p>注意事项说明：</p> <p>此接口默认不更新数据，从首次控制夹爪开始后，使能更新状态，如果此时控制灵巧手或打开末端 modbus 功能，将不再更新数据。</p> <p>另外夹爪需要支持最新的固件，方可支持此功能</p>
返回值	<pre>{"command":"get_gripper_state","enable":1,"status":1,"error":0,"mode":1,"current_force":100,"temperature":40,"actpos":150}</pre> <p>说明：</p> <p>enable：夹爪使能标志，0 表示未使能，1 表示使能</p> <p>status：夹爪在线状态，0 表示离线，1 表示在线</p> <p>error：夹爪错误信息，低 8 位表示夹爪内部的错误信息 bit5-7 保留 bit4 内部通 bit3 驱动器 bit2 过流 bit1 过温 bit0 堵转</p> <p>mode：当前工作状态：1 夹爪张开到最大且空闲，2 夹爪闭合到最小且空闲，3 夹爪停止且空闲，4 夹爪正在闭合，5 夹爪正在张开，6 夹爪闭合过程中遇到力控停止</p> <p>current_force：夹爪当前的压力，单位 g</p> <p>temperature：当前温度单位℃</p> <p>actpos：夹爪开口度</p>

## 4.9 末端工具—六维力（选配）

睿尔曼机械臂六维力版末端配备集成式六维力传感器，无需外部走线，用户可直接通过协议对六维力进行操作，获取六维力数据。

如下图所示，正上方为六维力的 Z 轴，航插反方向为六维力的 Y 轴，坐标系符合右手定则。机械臂位于零位姿态时，工具坐标系与六维力的坐标系方向一致。

另外，六维力额定力 200N，额定力矩 8Nm，过载水平 300%FS，工作温度 5~80℃，准度 0.5%FS。使用过程中注意使用要求，防止损坏六维力传感器。



### (1) 查询六维力数据

功能描述	查询当前六维力传感器得到的力和力矩信息：Fx,Fy,Fz,Mx,My,Mz
参数说明	get_force_data：获取力传感器信息，若要周期获取力数据，基础系列机械臂查询周期不能小于 50ms
命令格式	{s:s}
示例	{"command":"get_force_data"}
说明	无
返回值	<pre>{"command":"get_force_data","force_data":[1000,2000,3000,400,500,600],"zero_force_data":[500,1000,1500,200,250,300],"work_zero_force_data":[500,1000,1500,200,250,300],"tool_zero_force_data":[500,1000,1500,200,250,300]}</pre> <p>数据精度：0.001</p> <p>原始力数据 force_data 依次为 Fx=1N, Fy=2N, Fz=3N, Mx=0.4Nm, My=0.5Nm, Mz=0.6Nm；传感器坐标系下系统受到的外力数据 zero_force_data 依次为 Fx=0.5N, Fy=1N, Fz=1.5N, Mx=0.2Nm, My=0.25Nm, Mz=0.3Nm；当前工作坐标系下系统受到的外力数据 work_zero_force_data 依次为 Fx=0.5N, Fy=1N, Fz=1.5N, Mx=0.2Nm, My=0.25Nm, Mz=0.3Nm；当前工具坐标系下系统受到的外力数据 tool_zero_force_data 依次为 Fx=0.5N, Fy=1N, Fz=1.5N, Mx=0.2Nm, My=0.25Nm, Mz=0.3Nm；</p>

### (2) 六维力数据清零

功能描述	将六维力数据清零，标定当前状态下的零位
------	---------------------



参数说明	clear_force_data: 标定当前状态下的零位
命令格式	{s:s}
示例	{"command":"clear_force_data"}
说明	无
返回值	{"command":"clear_force_data","clear_state":true}清空成功 {"command":"clear_force_data","clear_state":false}清空失败

### (3) 自动设置六维力重心参数

功能描述	<p>设置六维力重心参数，六维力重新安装后，必须重新计算六维力所受到的初始力和重心。分别在不同姿态下，获取六维力的数据，用于计算重心位置。该指令下发后，机械臂以固定的速度运动到各标定点，该过程不可中断，中断后必须重新标定。</p> <p>重要说明：必须保证在机械臂静止状态下标定。</p> <p>以 RM65 机械臂为例，四个标定点的关节角度分别为：</p> <p>位置 1 关节角度：{0,0,-60,0,60,0}</p> <p>位置 2 关节角度：{0,0,-60,0,-30,0}</p> <p>位置 3 关节角度：{0,0,-60,0,-30,180}</p> <p>位置 4 关节角度：{0,0,-60,0,-120,0}</p>
参数说明	set_force_sensor: 设置力传感器指定位置时的数值
命令格式	{s:s}
示例	{"command":"set_force_sensor"}
返回值	{"command":"set_force_sensor","set_state":true}配置成功 {"command":"set_force_sensor","set_state":false}配置失败

### (4) 手动标定六维力数据

功能描述	<p>设置六维力重心参数，六维力重新安装后，必须重新计算六维力所受到的初始力和重心。该手动标定流程，适用于空间狭窄工作区域，以防自动标定过程中机械臂发生碰撞，用户可以手动选取四个位姿下发，当下发完四个点后，机械臂开始自动沿用户设置的目标运动，并在此过程中计算六维力重心。</p>
------	---



参数说明	<p>manual_set_force: 标定传感器重心数据</p> <p>pose1:位置 1 关节角度;</p> <p>pose2:位置 2 关节角度;</p> <p>pose3:位置 3 关节角度;</p> <p>pose4:位置 4 关节角度;</p> <p>上述 4 个位置必须按照顺序依次下发, 当下发完 pose4 后, 机械臂开始自动运行计算重心, 计算完成后返回协议。</p>
命令格式	<p>六自由度: {s:s,s:[i,i,i,i,i,i]}</p> <p>七自由度: {s:s,s:[i,i,i,i,i,i,i]}</p>
示例	<p>六自由度:</p> <pre>{"command":"manual_set_force_pose1", "joint":[0, 0, 0, 0, 90 000, 0]}</pre> <p>七自由度:</p> <pre>{"command":"manual_set_force_pose1", "joint":[0, 0, 0, 0, 0, 90000, 0]}</pre>
说明	<p>joint: 精度 0.001°, 即六自由度机械臂第一个位置关节 1~6 的目标角度依次为 0°, 0°, 0°, 0°, 90°, 0°, 七自由度机械臂第一个位置关节 1~7 的目标角度依次为 0°, 0°, 0°, 0°, 0°, 90°, 0°</p>
返回值	<p>{"command":"set_force_sensor","set_state":true} 标定成功</p> <p>{"command":"set_force_sensor","set_state":false} 标定失败</p>

### (5) 停止标定力传感器重心

功能描述	在标定力传感器过程中, 如果发生意外, 发送该指令, 停止机械臂运动, 退出标定流程
参数说明	stop_set_force_sensor: 停止计算力传感器重心位置
命令格式	{s:s}
示例	{"command":"stop_set_force_sensor"}
说明	
返回值	{"command":"stop_set_force_sensor","stop_state":true}计算成功



	<code>{"command":"stop_set_force_sensor","stop_state":false}</code> 计算失败
--	--

#### (6) 设置六维力拖动示教阈值参数

功能描述	设置六维力拖动示教阈值参数
参数说明	set_force_drag_threshold：设置六维力拖动示教阈值参数
命令格式	<code>{s:s,s:i,s:i}</code>
示例	<code>{"command":"set_force_drag_threshold","force":5000,"torque":500}</code>
说明	<p>force：此为可选参数，代表力方向（平移方向）的拖动阈值，单位 0.001N，参考坐标系为 Base 坐标系，设置范围为 2~20N，用户在机械臂末端施加大于拖动阈值的力才能让末端跟随用户平移。但拖动阈值设置过大会严重影响用户拖动体验。</p> <p>torque：此为可选参数，代表力矩方向（旋转方向）的拖动阈值，单位 0.001Nm，参考坐标系为 Base 坐标系，设置范围为 0.1Nm~1Nm，用户在机械臂末端施加大于拖动阈值的力矩才能让末端跟随用户旋转。拖动阈值设置过大会严重影响用户拖动体验。</p>
返回值	<p><code>{"command":"set_force_drag_threshold","set_state":true}</code>设置成功</p> <p><code>{"command":"set_force_drag_threshold","set_state":false}</code>设置失败</p>
备注	设置完成后，系统自动保存。

#### (7) 查询六维力拖动示教阈值参数

功能描述	查询六维力拖动示教阈值参数
参数说明	get_force_drag_threshold：查询六维力拖动示教阈值参数
命令格式	<code>{s:s}</code>
示例	<code>{"command":"get_force_drag_threshold"}</code>
返回值	<p><code>{"command":"get_force_drag_threshold","force":5000,"torque":500}</code></p> <p>说明：</p>



	<p>force: 力方向（平移方向）的拖动阈值，单位 0.001N，设置范围为 2~20N，参考坐标系为 Base 坐标系，用户在机械臂末端施加大于拖动阈值的力才能让末端跟随用户平移。但拖动阈值设置过大会严重影响用户拖动体验。</p> <p>torque: 力矩方向（旋转方向）的拖动阈值，单位 0.001Nm，参考坐标系为 Base 坐标系，设置范围为 0.1Nm~1Nm，用户在机械臂末端施加大于拖动阈值的力矩才能让末端跟随用户旋转。拖动阈值设置过大会严重影响用户拖动体验。</p>
--	--

#### (8) 初始化六维力拖动示教阈值参数

功能描述	初始化六维力拖动示教阈值参数
参数说明	init_force_drag_threshold: 初始化六维力拖动示教阈值参数
命令格式	{s:s}
示例	{"command":"init_force_drag_threshold"}
说明	
返回值	<p>{"command":"init_force_drag_threshold","init_state",true} 初始化成功</p> <p>{"command":"init_force_drag_threshold","init_state",false} 初始化失败</p>

### 4.10 末端传感器一维力（选配）

睿尔曼机械臂末端接口板集成了一维力传感器，可获取 Z 方向的力，量程 200N，准度 0.5%FS。





### (1) 查询末端一维力数据

功能描述	查询末端一维力数据
参数说明	get_Fz：获取末端一维力数据
命令格式	{s:s}
示例	{"command":"get_Fz"}
说明	<p>备注：基础系列机械臂第一帧指令下发后，开始更新一维力数据，此时返回的数据有滞后性；请从第二帧的数据开始使用。</p> <p>若周期查询 Fz 数据，频率不能高于 40Hz</p>
返回值	<p>{"command":"get_Fz","Fz":12000,"zero_Fz":100,"work_zero_Fz":100,"tool_zero_Fz":100}若成功，返回数据</p> <p>Fz 原始数据，精度：0.001N，示例返回 Fz 为 12N</p> <p>zero_Fz 传感器坐标系下系统外受力数据，精度：0.001N，示例返回 Fz 为 0.1N</p> <p>work_zero_Fz 当前工作坐标系下系统外受力原始数据，精度：0.001N，示例返回 Fz 为 0.1N</p> <p>tool_zero_Fz 当前工具坐标系下系统外受力原始数据，精度：0.001N，示例返回 Fz 为 0.1N</p> <p>{"command":"get_Fz","set_state":false}若失败，返回该指令</p>



## (2) 清零末端一维力数据

功能描述	清零末端一维力数据
参数说明	clear_Fz:清空一维力数据后, 后续所有获取到的数据都是基于当前的偏置。
命令格式	{s:s}
示例	{"command":"clear_Fz"}
说明	无
返回值	{"command":"clear_Fz","set_state":true}设置成功 {"command":"clear_Fz","set_state":false}设置失败

## (3) 自动标定一维力数据

功能描述	设置一维力重心参数, 一维力重新安装后, 必须重新计算一维力所受到的初始力和重心。分别在不同姿态下, 获取一维力的数据, 用于计算重心位置, 该步骤对于基于一维力的力位混合控制操作具有重要意义。
参数说明	auto_set_Fz: 标定传感器零位数据
命令格式	{s:s}
示例	{"command":"auto_set_Fz"}
返回值	{"command":"set_force_sensor","set_state":true}标定成功 {"command":"set_force_sensor","set_state":false}标定失败

## (4) 手动标定一维力数据

功能描述	设置一维力重心参数, 一维力重新安装后, 必须重新计算一维力所受到的初始力和重心。该手动标定流程, 适用于空间狭窄工作区域, 以防自动标定过程中机械臂发生碰撞, 用户可以手动选取 2 个位姿下发, 当下发完后, 机械臂开始自动沿用户设置的目标运动, 并在此过程中计算一维力重心。
参数说明	manual_set_Fz: 标定传感器零位数据 pose1:关节角度; pose2:关节角度;
命令格式	六自由度: {s:s,s:[i,i,i,i,i,i] ,s:[i,i,i,i,i,i]}



	七自由度: {s:s,s:[i,i,i,i,i,i,i] ,s:[i,i,i,i,i,i,i]}
示例	六自由度: <pre>{"command":"manual_set_Fz", "pose1":[0, 0, 0, 0, 0, 0], "pose2":[0, 0, 90000, 0, 90000, 0]}</pre> 七自由度: <pre>{"command":"manual_set_Fz", "pose1":[0, 0, 0, 0, 0, 0], "pose2":[0, 0, 90000, 0, 0, 90000, 0]}</pre>
说明	pose1: 关节角度, 精度 0.001 °, 如 90000 代表 90 °
返回值	<pre>{"command":"set_force_sensor","set_state":true}</pre> 标定成功 <pre>{"command":"set_force_sensor","set_state":false}</pre> 标定失败

#### (5) 停止标定力传感器重心

功能描述	在标定力传感器过程中, 如果发生意外, 发送该指令, 停止机械臂运动, 退出标定流程
参数说明	stop_set_force_sensor: 停止计算力传感器重心位置
命令格式	{s:s}
示例	<pre>{"command":"stop_set_force_sensor"}</pre>
说明	
返回值	<pre>{"command":"stop_set_force_sensor","stop_state":true}</pre> 计算成功 <pre>{"command":"stop_set_force_sensor","stop_state":false}</pre> 计算失败

### 4.11 拖动示教

睿尔曼机械臂在拖动示教过程中, 可记录拖动的轨迹点, 并根据用户的指令对轨迹进行复现。

#### (1) 拖动示教开始

功能描述	拖动示教开始
参数说明	start_drag_teach: 拖动示教开始, trajectory_record: 拖动示教时记录轨迹, 0-不记录, 1-记录轨迹



命令格式	{s:s,s:i}
示例	{"command":"start_drag_teach","trajectory_record":1}
	说明：开始拖动示教，同时记录轨迹
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	{"command":"start_drag_teach","drag_teach":true}

## (2) 拖动示教结束

功能描述	拖动示教结束
参数说明	stop_drag_teach：拖动示教结束
命令格式	{s:s}
示例	{"command":"stop_drag_teach"}
	说明：拖动示教停止
返回值	格式：{s:s,s:b},true-设置成功，false-设置失败
	{"command":"stop_drag_teach","drag_teach":true}

## (3) 开始复合模式拖动示教

功能描述	开始复合模式拖动示教
参数说明	start_multi_drag_teach:开始复合模式拖动示教
命令格式	{s:s,s:i,s:i}
示例	{"command":"start_multi_drag_teach","mode":0,"singular_wall":0}
	说明： mode：拖动示教模式 0-电流环模式，1-使用末端六维力，只动位置，2-使用末端六维力，只动姿态，3-使用末端六维力，位置和姿态同时动 singular_wall：可选参数，仅在六维力模式拖动示教中生效，用于指定是否开启拖动奇异墙，0 表示关闭拖动奇异墙，1 表示开启拖动奇异墙，若无配置参数，默认启动拖动奇异墙
返回值	{"command":"start_multi_drag_teach","set_state":true}设置成功
	{"command":"start_multi_drag_teach","set_state":false}设置失败



备注	失败的可能原因： 当前机械臂非六维力版本（六维力拖动示教） 机械臂当前处于 IO 急停状态 机械臂当前处于仿真模式 输入参数有误 使用六维力模式拖动示教时，当前已处于奇异区
----	---

#### （4）轨迹复现开始

功能描述	复现拖动示教中的轨迹： 备注：必须在拖动示教结束后才能使用，同时保证机械臂位于拖动示教的起点位置
参数说明	run_drag_trajectory：轨迹复现开始
命令格式	{s:s}
示例	{"command":"run_drag_trajectory"}
返回值	{"command":"run_drag_trajectory","run_state":true}复现成功 {"command":"run_drag_trajectory","run_state":false}复现失败

#### （5）轨迹复现暂停

功能描述	轨迹复现过程中暂停
参数说明	pause_drag_trajectory：轨迹复现暂停
命令格式	{s:s}
示例	{"command":"pause_drag_trajectory"}
返回值	{"command":"pause_drag_trajectory","pause_state":true}暂停成功 {"command":"pasuse_drag_trajectory","pause_state":false}暂停失败

#### （6）轨迹复现继续

功能描述	轨迹复现过程中暂停后继续 备注：轨迹继续时，必须保证机械臂位于暂停时的位置，否则会报
------	---



	错，用户只能从开始位置重新复现轨迹。
参数说明	continue_drag_trajectory：轨迹复现继续
命令格式	{s:s}
示例	{"command":"continue_drag_trajectory"}
返回值	<pre>{"command":"continue_drag_trajectory","continue_state":true}</pre> 继续成功 <pre>{"command":"continue_drag_trajectory","continue_state":false}</pre> 继续失败

### (7) 轨迹复现停止

功能描述	轨迹复现停止
参数说明	stop_drag_trajectory：轨迹复现继续
命令格式	{s:s}
示例	{"command":"stop_drag_trajectory"}
返回值	<pre>{"command":"stop_drag_trajectory","stop_state":true}</pre> 停止成功 <pre>{"command":"stop_drag_trajectory","stop_state":false}</pre> 停止失败

### (8) 运动到轨迹起点

功能描述	轨迹复现前，必须控制机械臂运动到轨迹起点，如果设置正确，机械臂将以 20%的速度运动到轨迹起点
参数说明	drag_trajectory_origin：轨迹复现起点
命令格式	{s:s}
示例	{"command":"drag_trajectory_origin"}
返回值	参照 3-5

### (9) 力位混合控制

功能描述	在笛卡尔空间轨迹规划时，使用该功能可保证机械臂末端接触力恒定，使用时力的方向与机械臂运动方向不能在同一方向。开启力位混合控制，执行笛卡尔空间运动，接收到运动完成反馈后，需要等待 2S 后继续下发下一条运动指令。
------	---



参数说明	set_force_position:设置力位混合控制模式
命令格式	{s:s,s:i,s:i,s:i,s:i}
示例	{"command":"set_force_position","sensor":1,"mode":0,"direction":2,"N":10}
说明	Sensor:传感器；0-一维力；1-六维力 Mode：0-工作坐标系力控；1-工具坐标系力控； direction：力控方向；0-沿 X 轴；1-沿 Y 轴；2-沿 Z 轴；3-沿 RX 姿态方向；4-沿 RY 姿态方向；5-沿 RZ 姿态方向 N:力的大小，单位 0.1N
返回值	{"command":"set_force_position","set_state":true}设置成功 {"command":"set_force_position","set_state":false}设置失败

### (10) 结束力位混合控制

功能描述	退出力位混合控制模式
参数说明	stop_force_position:结束力位混合控制模式
命令格式	{s:s}
示例	{"command":"stop_force_position"}
返回值	{"command":"stop_force_position","stop_state":true}结束成功 {"command":"stop_force_position","stop_state":false}结束失败

## 4.12 五指灵巧手（选配）

睿尔曼机械臂末端配置因时的五指灵巧手，可通过协议对灵巧手进行设置。

### (1) 设置灵巧手手势

功能描述	设置灵巧手手势
参数说明	set_hand_posture：设置手势 posture_num:预先保存在灵巧手内的手势序号，范围：1~40
命令格式	{s:s,s:i}
示例	{"command":"set_hand_posture","posture_num":1}
说明	设置灵巧手执行 1 号手势
返回值	{"command":"set_hand_posture","set_state":true}设置成功



	{"command":"set_hand_posture","set_state":false}设置失败
--	--

## (2) 设置灵巧手动作序列

功能描述	设置灵巧手动作序列
参数说明	set_hand_seq: 设置手势 seq_num:预先保存在灵巧手内的序列序号, 范围: 1~40
命令格式	{s:s,s:i}
示例	{"command":"set_hand_seq","seq_num":1}
说明	设置灵巧手执行 1 号动作序列
返回值	{"command":"set_hand_seq","set_state":true}设置成功 {"command":"set_hand_seq","set_state":false}设置失败

## (3) 设置灵巧手各自由度角度

功能描述	设置灵巧手角度, 灵巧手有 6 个自由度, 从 1~6 分别为小拇指, 无名指, 中指, 食指, 大拇指弯曲, 大拇指旋转
参数说明	set_hand_angle: 设置手指角度 hand_angle:手指角度数组, 范围: 0~1000.另外, -1 代表该自由度不执行任何操作, 保持当前状态
命令格式	{s:s,s:[i,i,i,i,i,i]}
示例	{"command":"set_hand_angle","hand_angle":[-1,100,200,300,400,500]}
说明	设置灵巧手各手指动作
返回值	{"command":"set_hand_angle","set_state":true}设置成功 {"command":"set_hand_angle","set_state":false}设置失败

## (4) 设置灵巧手速度

功能描述	设置灵巧手关节速度
参数说明	set_hand_speed: 设置手指角度 hand_speed:手指速度, 范围: 1~1000





命令格式	{s:s,s:i}
示例	{"command":"set_hand_speed","hand_speed":500}
说明	设置灵巧手各手指动作速度
返回值	{"command":"set_hand_speed","set_state":true}设置成功 {"command":"set_hand_speed","set_state":false}设置失败

#### (5) 设置灵巧手力阈值

功能描述	设置灵巧手关节力阈值
参数说明	set_hand_force: 设置手指力阈值 hand_force:手指力, 范围: 1~1000
命令格式	{s:s,s:i}
示例	{"command":"set_hand_force","hand_force":500}
说明	设置灵巧手各手指动作力阈值
返回值	{"command":"set_hand_force","set_state":true}设置成功 {"command":"set_hand_force","set_state":false}设置失败

### 4.13 Modbus 配置

睿尔曼机械臂在控制器的航插和末端接口板航插处, 各有 1 路 RS485 通讯接口, 这两个 RS485 端口可通过 JSON 协议配置为标准的 ModbusRTU 模式。然后通过 JSON 协议对端口连接的外设进行读写操作。

注意: 控制器的 RS485 接口在未配置为 Modbus RTU 模式的情况下, 可用于用户对机械臂进行控制, 这两种模式不可兼容。若要恢复机械臂控制模式, 必须将该端口的 Modbus RTU 模式关闭。Modbus RTU 模式关闭后, 系统会自动切换回机械臂控制模式, 波特率 460800BPS, 停止位 1, 数据位 8, 无检验。

同时, I 系列控制器支持 modbus-TCP 主站配置, 可配置使用 modbus-TCP 主站, 用于连接外部设备的 modbus-TCP 从站。



### (1) 配置通讯端口 ModbusRTU 模式

功能描述	配置通讯端口 ModbusRTU 模式，机械臂启动后，要对通讯端口进行任何操作，必须先启动该指令，否则会返回报错信息。 另外，机械臂会对用户的配置方式进行保存，机械臂重启后会自动恢复到用户断电之前配置的模式。
参数说明	set_modbus_mode：配置端口 ModbusRTU 模式
命令格式	{s:s,s:i,s:i,s:i}
示例	{"command":"set_modbus_mode","port":0,"baudrate":115200,"timeout":1}
说明	port:通讯端口，0-控制器 RS485 端口为 RTU 主站，1-末端接口板 RS485 接口为 RTU 从站，2-控制器 RS485 端口为 RTU 从站 baudrate：波特率，支持 9600,115200,460800 三种常见波特率 Timeout：超时时间，单位百毫秒。对 Modbus 设备所有的读写指令，在规定的超时时间内未返回响应数据，则返回超时报错提醒。 超时时间不能为 0，若设置为 0，则机械臂按 1 进行配置。 其他配置默认为：数据位-8，停止位-1，奇偶校验-无
返回值	{"command":"set_modbus_mode","set_state":true}设置成功 {"command":"set_modbus_mode","set_state":false}设置失败

### (2) 关闭通讯端口 ModbusRTU 模式

功能描述	关闭通讯端口 ModbusRTU 模式
参数说明	close_modbus_mode：关闭端口 ModbusRTU 模式
命令格式	{s:s}
示例	{"command":"close_modbus_mode","port":0}
说明	关闭后，该端口将不会响应任何读写指令的操作 port:通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，2-控制器 RS485 端口为 RTU 从站
返回值	{"command":"close_modbus_mode","set_state":true}设置成功 {"command":"close_modbus_mode","set_state":false}设置失败



### (3) 配置连接 ModbusTCP 从站 (I 系列)

功能描述	控制器作为 ModbusTCP 主站连接外部 ModbusTCP 从站设备
参数说明	set_modbustcp_mode: 连接 ModbusTCP 从站
命令格式	{s:s,s:s,i:s,i}
示例	{"command":"set_modbustcp_mode","ip":"192.168.1.120","port":502,"timeout":2000}
说明	ip: 从机 IP 地址, port: 端口号, timeout: 超时时间, 单位: 秒
返回值	{"command":"set_modbustcp_mode","set_state":true} //连接成功 {"command":"set_modbustcp_mode","set_state":false} //连接失败

### (4) 配置关闭 ModbusTCP 从站 (I 系列)

功能描述	断开控制器连接的 ModbusTCP 从站设备
参数说明	close_modbustcp_mode: 断开 ModbusTCP 从站
命令格式	{s:s}
示例	{"command":"close_modbustcp_mode"}
返回值	{"command":"close_modbustcp_mode","set_state":true} //成功 {"command":"close_modbustcp_mode","set_state":false} //失败

### (5) 读线圈

功能描述	读线圈
参数说明	read_coils: 读线圈
命令格式	{s:s,s:i,s:i,s:i,s:i}
示例	{"command":"read_coils","port":0,"address":10,"num":2,"device":2}
说明	port:通讯端口, 0-控制器 RS485 端口, 1-末端接口板 RS485 接口, 3-控制器 ModbusTCP 设备 address: 线圈起始地址 num: 要读的线圈的数量, 该指令最多一次性支持读 8 个线圈数据,



	即返回的数据不会超过一个字节 device: 外设备地址
返回值	{"command":"read_coils","data":8}读取成功, 返回线圈状态, 数据类型: int8  {"command":"read_coils","read_state":false}读取失败, 超时时间内未获取到数据

#### (6) 读离散量输入

功能描述	读离散量输入
参数说明	read_input_status: 读离散量输入
命令格式	{s:s,s:i,s:i,s:i,s:i}
示例	{"command":"read_input_status","port":0,"address":10,"num":2,"device":2}
说明	port:通讯端口, 0-控制器 RS485 端口, 1-末端接口板 RS485 接口, 3-控制器 ModbusTCP 设备 address: 数据起始地址 num: 要读的数据的数量, 该指令最多一次性支持读 8 个离散量数据, 即返回的数据不会超过一个字节 device: 外设备地址
返回值	{"command":"read_input_status","data":8}读取成功, 返回离散量, 数据类型: int8  {"command":"read_input_status","read_state":false}读取失败, 超时时间内未获取到数据

#### (7) 读保持寄存器

功能描述	读保持寄存器
参数说明	read_holding_registers: 读保存寄存器
命令格式	{s:s,s:i,s:i,s:i}
示例	{"command":"read_holding_registers","port":0,"address":10,"device":2}
说明	port:通讯端口, 0-控制器 RS485 端口, 1-末端接口板 RS485 接口,



	3-控制器 ModbusTCP 设备 address: 数据起始地址, 该指令每次只能读 1 个寄存器, 即 2 个字节的数 据, 不可一次性读取多个寄存器数据 device: 外设备地址
返回值	{"command":"read_holding_registers","data":8}读取成功, 返回寄存器数据, 数据类型: int16  {"command":"read_holding_registers","read_state":false}读取失败, 超时时间内未获取到数据

### (8) 读输入寄存器

功能描述	读输入寄存器
参数说明	read_input_registers: 读输入寄存器
命令格式	{s:s,s:i,s:i,s:i}
示例	{"command":"read_input_registers","port":0,"address":10,"device":2}
说明	port:通讯端口, 0-控制器 RS485 端口, 1-末端接口板 RS485 接口, 3-控制器 ModbusTCP 设备 address: 数据起始地址, 该指令每次只能读 1 个寄存器, 即 2 个字节的数 据, 不可一次性读取多个寄存器数据 device: 外设备地址
返回值	{"command":"read_input_registers","data":8}读取成功, 返回寄存器数据, 数据类型: int16  {"command":"read_input_registers","read_state":false}读取失败, 超时时间内未获取到数据

### (9) 写单圈数据

功能描述	写单圈数据
参数说明	write_single_coil: 写单圈数据
命令格式	{s:s,s:i,s:i,s:i,s:i}
示例	{"command":"write_single_coil","port":0,"address":10,"data":1,"device":2}



说明	port:通讯端口,0-控制器 RS485 端口,1-末端接口板 RS485 接口, 3-控制器 ModbusTCP 设备 address: 线圈起始地址 data: 要写入线圈的数据,数据类型: int16 device: 外设备地址
返回值	{"command":"write_single_coil","write_state":true}写操作成功 {"command":"write_single_coil","write_state":false}写操作失败, 超时时间内未获取到数据,或者指令内容错误

### (10) 写单个寄存器

功能描述	写单个寄存器数据
参数说明	write_single_register: 写单个寄存器
命令格式	{s:s,s:i,s:i,s:i,s:i}
示例	{"command":"write_single_register","port":0,"address":10,"data": 1000,"device":2}
说明	port:通讯端口,0-控制器 RS485 端口,1-末端接口板 RS485 接口, 3-控制器 ModbusTCP 设备 address: 寄存器起始地址 data: 要写入寄存器的数据,数据类型: int16 device: 外设备地址
返回值	{"command":"write_single_register","write_state":true}写操作成功 {"command":"write_single_register","write_state":false}写操作失 败,超时时间内未获取到数据,或者指令内容错误

### (11) 写多个寄存器

功能描述	写多个寄存器数据
参数说明	write_registers: 写多个寄存器
命令格式	{s:s,s:i,s:i,s:i,s:[i,i...],s:i }
示例	{"command":"write_registers","port":0,"address":10,"num":2,"dat a":[15, 20, 25, 30],"device":2}



说明	port:通讯端口, 0-控制器 RS485 端口, 1-末端接口板 RS485 接口, 3-控制器 ModbusTCP 设备 address: 寄存器起始地址 num: 写寄存器个数, 寄存器每次写的数量不超过 10 个 data: 要写入寄存器的数据数组, 类型: byte device: 外设设备地址
返回值	{"command":"write_registers","write_state":true}写操作成功 {"command":"write_registers","write_state":false}写操作失败, 超时时间内未获取到数据, 或者指令内容错误

### (12) 写多圈数据

功能描述	写多圈数据
参数说明	write_coils: 写多个寄存器
命令格式	{s:s,i,s:i,s:i,s:[i,i...],s:i }
示例	{"command":"write_coils","port":0,"address":10,"num":16,"data":[15, 20],"device":2}
说明	port:通讯端口, 0-控制器 RS485 端口, 1-末端接口板 RS485 接口, 3-控制器 ModbusTCP 设备 address: 线圈起始地址 num: 写线圈个数, 每次写的数量不超过 160 个 data: 要写入线圈的数据数组, 类型: byte。若线圈个数不大于 8, 则写入的数据为 1 个字节; 否则, 则为多个数据的数组 device: 外设设备地址
返回值	{"command":"write_coils","write_state":true}写操作成功 {"command":"write_coils","write_state":false}写操作失败, 超时时间内未获取到数据, 或者指令内容错误

### (13) 读多圈数据

功能描述	读多圈数据
参数说明	read_multiple_coils: 读线圈



命令格式	{s:s,s:i,s:i,s:i,s:i}
示例	{"command":"read_multiple_coils","port":0,"address":0,"num":24,"device":2}
说明	port:通讯端口,0-控制器 RS485 端口,1-末端接口板 RS485 接口,3-控制器 ModbusTCP 设备 address: 线圈起始地址 num: $8 < \text{num} \leq 120$ 要读的线圈的数量,该指令最多一次性支持读 120 个线圈数据,即 15 个 byte device: 外设备地址
返回值	{"command":"read_multiple_coils","data":[1,2,3]}读取成功,返回线圈状态,数据类型: int8 {"command":"read_multiple_coils","read_state":false}读取失败,超时时间内未获取到数据

#### (14) 读多个保持寄存器

功能描述	读多个保持寄存器
参数说明	read_multiple_holding_registers: 读多个保持寄存器
命令格式	{s:s,s:i,s:i,s:i,s:i}
示例	{"command":"read_multiple_holding_registers","port":0,"address":0,"num":5,"device":2}
说明	port:通讯端口,0-控制器 RS485 端口,1-末端接口板 RS485 接口,3-控制器 ModbusTCP 设备 address: 寄存器起始地址 num: $2 < \text{num} < 13$ 要读的寄存器的数量,该指令最多一次性支持读 12 个寄存器数据,即 24 个 byte device: 外设备地址
返回值	{"command":"read_multiple_holding_registers","data":[1,2,3,4,5,6,7,8,9,10]}读取成功,返回寄存器数据,数据类型: int8 {"command":"read_multiple_holding_registers","read_state":false}读取失败,超时时间内未获取到数据





#### 4.14 系统安装方式及关节版本信息

睿尔曼机械臂可支持不同形式的安装方式，但是安装方式不同，机器人的动力学模型参数和坐标系的方向也有所差别。

##### (1) 设置安装方式参数

功能描述	设置安装方向参数
参数说明	set_install_pose:设置机械臂基座安装方式
命令格式	{s:s,s:[i,i,i]}
示例	{"command":"set_install_pose","pose":[0,90,45]}
说明	Pose:基座相对水平面的旋转角、俯仰角和方位角，精度：1 度，如上所示，即为旋转角为 0 度，俯仰角为 90 度，方位角为 45 度
返回值	{"command":"set_install_pose","set_state":true}设置成功 {"command":"set_install_pose","set_state":false}设置失败

##### (2) 查询关节软件版本号

功能描述	查询关节软件版本号
参数说明	get_joint_software_version:获取关节软件版本号
命令格式	{s:s}
示例	{"command":"get_joint_software_version"}
说明	无
返回值	{s:s,s:[i,i,i,i,i]}
	{"state":"joint_software_version","version":[531,531,531,531,531,531]}
	531 为 uint16 类型，转化为 16 进制为：0x0213，则当前关节的版本号为 2.13

##### (3) 查询末端接口板软件版本号

功能描述	查询末端接口板软件版本号
参数说明	get_tool_software_version:获取末端接口板软件版本号



命令格式	{s:s}
示例	{"command":"get_tool_software_version"}
说明	无
返回值	{s:s,s:i}
	{"state":"tool_software_version","version":531}
	531 为 uint16 类型，转化为 16 进制为：0x0213，则当前末端接口板的版本号为 2.13

#### 4.15 透传力位混合控制补偿（选配）

针对睿尔曼带一维力和六维力版本的机械臂，用户除了可直接使用示教器调用底层的力位混合控制模块外，还可以将自定义的轨迹以周期性透传的形式结合底层的力位混合控制算法进行补偿。

##### （1）开启透传力位混合控制补偿模式

功能描述	开启底层力位混合控制模块补偿模式。在下发透传轨迹前必须下发该指令开启功能
参数说明	Start_Force_Position_Move
命令格式	{s:s}
示例	{"command":"Start_Force_Position_Move"}
说明	无
返回值	{s:s,s:b}
	{"command":"Start_Force_Position_Move","set_state":true}
	True：设置成功，可进行后续透传。False：设置失败，机械臂有错误，不可进行后续透传

##### （2）Force\_Position\_Move：透传力位混合补偿

功能描述	Force_Position_Move：用户周期性下发目标角度或者目标位姿，使用机械臂底层力位混合控制模块通过一维力传感器或者六维力
------	--



	<p>传感器实现力位补偿。</p> <p>备注 1：该功能只适用于一维力传感器和六维力传感器机械臂版本</p> <p>备注 2：透传效果和周期、轨迹是否平滑有关，周期要求稳定，防止出现较大波动，用户使用该指令时请做好轨迹规划，轨迹规划的平滑程度决定了机械臂的运行状态。基础系列 WIFI 和网口模式透传周期最快 20ms，USB 和 RS485 模式透传周期最快 10ms。高速网口的透传周期最快也可到 10ms，不过在使用该高速网口前，需要使用指令打开配置。另外 I 系列有线网口周期最快可达 2ms。</p>
参数说明	<p>Force_Position_Move：透传力位混合补偿</p> <p>pose：当前坐标系下目标位姿，位置精度：0.001mm，欧拉角表示姿态，姿态精度：0.001rad</p> <p>pose_quat：表示当前坐标系下的目标位姿，位置精度：0.001mm，采用四元数方式表示姿态，姿态精度：0.000001</p> <p>joint：目标关节角度，精度 0.001 °</p> <p>sensor：所使用传感器类型，0-一维力，1-六维力</p> <p>mode：模式，0-沿工作坐标系，1-沿工具端坐标系</p> <p>dir：力控方向，0~5 分别代表 X/Y/Z/Rx/Ry/Rz，其中一维力类型时默认方向为 Z 方向</p> <p>force：力的大小，精度 0.1N 或者 0.1Nm</p> <p>follow：表示驱动器的运动跟随效果，true 为高跟随，false 为低跟随。</p>
命令格式	<p>六自由度机械臂关节角度方式或位姿（姿态欧拉角）方式{s:s,s:[i,i,i,i,i,i],s:i,s:i,s:i,s:i,s:b}</p> <p>七自由度机械臂关节角度方式或位姿（姿态四元数）方式{s:s,s:[i,i,i,i,i,i,i,i],s:i,s:i,s:i,s:i,s:i,s:b}</p>
示例	<p>位姿（姿态欧拉角）方式</p> <pre>{"command":"Force_Position_Move","pose":[100000,200000,30000,400,500,600],"sensor":0,"mode":0,"dir":0,"force":15,"follow":true}</pre> <p>位姿（姿态四元数）方式</p> <pre>{"command":"Force_Position_Move","pose_quat":[100000,200000,30000,400000,500000,600000,700000],"sensor":0,"mode":</pre>



	<p>0,"dir":0, "force":15,"follow":true}</p> <p>说明：透传目标位姿进行力位混合控制补偿</p> <p>目标位置（欧拉角方式）：x: 0.1m, y:0.2m, z: 0.03m, Rx: 0.4rad, Ry: 0.5rad, Rz: 0.6rad</p> <p>目标位置（四元数方式）：x: 0.1m, y:0.2m, z: 0.03m, w: 0.4, x: 0.5, y: 0.6, z: 0.7</p> <p>使用一维力沿工作坐标系进行 Z 方向补偿，力控为 Fz: 1.5N</p> <p>{"command":"Force_Position_Move","joint":[1000,2000,3000,4000,5000,6000],"sensor":0,"mode":0, "dir":0, "force":15}</p> <p>说明：透传目标角度进行力位混合控制补偿</p> <p>关节 1~6 目标角度：1°，2°，3°，4°，5°，6°</p> <p>使用一维力沿工作坐标系进行 Z 方向补偿，力控为 Fz: 1.5N</p>
返回值	<p>规划成功——返回当前各关节角度和所使用力控方式的力或力矩，如果使用的是六维力，则也会返回全部方向上的力和力矩</p> <p>{s:s,s:[i,i,i,i,i,i],s:i, s:i}或{s:s, s:[i,i,i,i,i,i], s:i, s:[i,i,i,i,i,i], s:i}</p> <p>一维力：</p> <p>{"state":"Force_Position_State","joint":[10,20,30,40,50,60],"force":-15, "arm_err":0}当前关节 1~6 角度为 0.01°~0.06°，所受到的力或力矩为-1.5</p> <p>六维力：</p> <p>{"state":"Force_Position_State","joint":[10,20,30,40,50,60],"force":-15, "all_direction_force":[11,21,-15,41,51,61],"arm_err":0}当前关节 1~6 角度为 0.01°~0.06°，力控方向所受到的力或力矩为 -1.5N，所有方向的力或力矩为 X: 1.1N, Y: 2.1N, Z: -1.5N, Rx: 4.1Nm, Ry: 5.1Nm, Rz: 6.1Nm</p> <p>需要注意的是 I 系列机械臂不再提供返回值，可通过 UDP 状态主动上报接口采集机械臂实时状态。</p> <p>规划失败——返回错误提示, {s:s,s:b}</p> <p>{"command":"Force_Position_Move", "set_state":false}</p>
备注：	透传开始的起点务必为机械臂当前位姿，否则可能会力控补偿失败或者机械臂无法运动！



### (3) 关闭透传力位混合控制补偿模式

功能描述	关闭底层力位混合控制模块补偿模式。在完成透传轨迹后必须下发该指令关闭功能
参数说明	Stop_Force_Position_Move
命令格式	{s:s}
示例	{"command":"Stop_Force_Position_Move"}
说明	无
返回值	{s:s,s:b}
	{"command":"Stop_Force_Position_Move","set_state":true}
	True: 设置成功。 False: 设置失败

## 五、升降机构（选配）

### (1) 速度开环控制

功能描述	升降机构速度开环控制
参数说明	set_lift_speed: 设置升降机构速度
命令格式	{s:s,s:i}
示例	{"command":"set_lift_speed","speed":50}
	说明: speed-速度百分比, -100~100。 speed<0:升降机构向下运动 speed>0:升降机构向上运动 speed=0: 升降机构停止运动
返回值	{"command":"set_lift_speed","set_state":true}设置成功 {"command":"set_lift_speed","set_state":false}设置失败

### (2) 位置闭环控制

功能描述	升降机构位置闭环控制
参数说明	set_lift_height: 设置升降机构高度



命令格式	{s:s,s:i}
示例	{"command":"set_lift_height","height":1000,"speed":50}
	说明: height-目标高度, 单位 mm, 范围: 0~2600 speed-速度百分比, 1~100
返回值	{"state":"current_trajectory_state","trajectory_state":true,"device":3}成功到位
	{"state":"current_trajectory_state","trajectory_state":false,"device":3}失败

### (3) 获取升降机构状态

功能描述	获取升降机构状态
参数说明	get_lift_state:获取升降机构状态
命令格式	{s:s}
示例	{"command":"get_lift_state"}
返回值	{"state":"lift_state","height":1000,"current":500,"err_flag":0,"mode":1}
	Height:当前升降机构高度, 单位: mm, 精度: 1mm, 范围: 0~2300 Current:当前升降驱动电流, 单位: mA, 精度: 1mA Err_flag:升降驱动错误代码, 错误代码类型参考关节错误代码 Mode 当前升降状态, 0-空闲, 1-正方向速度运动, 2-正方向位置运动, 3-负方向速度运动, 4-负方向位置运动

## 六、通用扩展关节 (选配)

### (1) 扩展关节状态获取

功能描述	扩展关节状态获取
参数说明	expand_get_state: 扩展关节状态获取
命令格式	{s:s}
示例	{"command":"expand_get_state"}



返回值	<pre>{"state": "expand_state", "pos": 0, "err_flag": 0, "current": 0, "mode": 0}</pre> <p>pos: 扩展关节角度, 单位度, 精度 0.001 °</p> <p>current: 当前升降驱动电流, 单位: mA, 精度: 1mA</p> <p>err_flag: 升降驱动错误代码, 错误代码类型参考关节错误代码</p> <p>mode: 当前升降状态, 0-空闲, 1-正方向速度运动, 2-正方向位置运动, 3-负方向速度运动, 4-负方向位置运动</p>
-----	--

## (2) 关节速度环控制

功能描述	扩展关节速度环控制
参数说明	expand_set_speed: 扩展关节速度环控制
命令格式	{s:s,s:i}
示例	<pre>{"command": "expand_set_speed", "speed": -50}</pre>
	speed: -50 表示最大速度的百分之五十反方向运动
返回值	<pre>{"command": "expand_set_speed", "set_speed_state": true}</pre> 设置成功
	<pre>{"command": "expand_set_speed", "set_speed_state": false}</pre> 设置失败

## (3) 关节位置环控制

功能描述	扩展关节位置环控制
参数说明	expand_set_pos: 扩展关节位置环控制
命令格式	{s:s,s:i,s:i}
示例	<pre>{"command": "expand_set_pos", "pos": 100, "speed": 50}</pre>
	speed: 50 表示最大速度的百分之五十, 且速度必须大于 0 pos: 精度 0.001 °
返回值	<pre>{"command": "expand_set_pos", "set_pos_state": true}</pre> true 表示收到运行指令, 到位后, 会返回到位指令
	<pre>{"state": "current_trajectory_state", "device": 4, "trajectory_state": ...}</pre>



	": true}
--	----------

## 七、在线编程

### 7.1 文件传输

#### (1) 下发前准备

功能描述	开始下发文件前准备
参数说明	project_name:文件名称;file_size:文件大小,plan_speed:轨迹运行速度
命令格式	{s:s,s:s,s:i}
示例	{"command":"run_project","project_name":"XXX","file_size":2048,"plan_speed":50}
	说明: file_size: 文件的大小, 单位 Byte。 plan_speed:规划速度比例系数 文件下发前准备
返回值	{"command":"run_project","project_state":true}准备就绪

#### (2) 下发前准备(I 系列)

功能描述	开始下发文件前准备
参数说明	project_name:文件名称;file_size:文件大小,plan_speed:轨迹运行速度
命令格式	{s:s,s:s,s:i}
示例	{"command":"run_project","project_name":"XXX","file_size":2048,"plan_speed":50,"auto_start":1,"step_flag":0}
	说明: file_size: 文件的大小, 单位 Byte。 plan_speed:规划速度比例系数; auto_start:设置默认在线编程文件 1-设置默认 0-设置非默认 step_flag:设置单步运行方式模式 1-设置单步模式 0-设置正常运动模式





	文件下发前准备
返回值	{"command":"run_project","project_state":true}准备就绪

### (3) 发送过程中（从第二次开始）

功能描述	机械臂每收到 2K 的数据，返回一帧该指令，示教器收到后，继续发送剩余数据
命令格式	{}
示例	文件流（2k）
	说明： 发送剩余文件数据，最大 2k
返回值	{"command":"conduct_project","project_conduct":true}接收成功

### (4) 发送校验返回

功能描述	控制器接收到最后一帧数据后，进行校验，返回成功或者失败状态
参数说明	project_state:校验结果;err_line:错误行数
命令格式	{s:s,s:b}或者{s:s,s:b,s:i}
示例	文件传输完成
	说明：
返回值	{"command":"download_project","project_state":true}校验成功，示教器弹框提示
	{"command":"download_project","project_state":false,"err_line":60}
	校验失败，err_line 为有问题的工程行数，示教器弹框提示，并将该行轨迹标红。若 err_line 为 0，则代表校验数据长度不对

### (5) 规划过程中改变速度系数

功能描述	轨迹规划中改变速度比例系数
参数说明	speed:速度
命令格式	{s:s,s:i}
示例	{"command":"plan_speed","speed":50}



	说明：speed：当前进度条的速度数据
返回值	{"command":"plan_speed","plan_state":true}设置速度系数成功

## 7.2 提示框

### (1) 弹窗提示

功能描述	文件树弹窗提醒 本指令是控制器发送给示教器，返回值是示教器发送给控制器
参数说明	content：弹窗提示指令所在文件树的位置
命令格式	{s:s,s:i}
示例	{"state":"popup","content":1} 说明；执行文件树第一条弹窗
返回值	{"command":"popup","popup_result":true}继续执行

## 7.3 拖动示教

### (1) 获取拖动示教轨迹

功能描述	获取刚拖动过的轨迹，在拖动示教后调用
参数说明	save_trajectory：保存轨迹
命令格式	{s:s,s:s}
示例	{"command":"save_trajectory", "trajectory_name":"XXX"} 说明： trajectory_name：轨迹要保存的名称（任意）
返回值	{"point":[" 30971, 56885, -3416, 76201, 47121, -4845]} 返回全部拖动示教点位

## 7.4 在线编程存储列表（I 系列）

### (1) 查询在线编程程序列表



功能描述	查询轨迹列表
参数说明	get_program_trajectory_list: 查询已存储的轨迹列表, 支持可分页查询和模糊查询
命令格式	{s:s,s:i,s:i,s:s}
示例	<pre>{"command":"get_program_trajectory_list","page_num":1,"page_size":10,"vague_search":"file"}</pre>
	<p>说明:</p> <p>page_num:页码 (全部查询时不传此参数)</p> <p>page_size:每页大小 (全部查询时不传此参数)</p> <p>vague_search:模糊搜索 (传递此参数可进行模糊查询)</p>
返回值	<pre>{"command":"get_program_trajectory_list","page_num":1,"page_size":2,"total_size":2,"vague_search":"file","list":[{"id":1,"size":2580,"speed":50,"trajectory_name":"1_file1.txt"}, {"id":2,"size":2580,"speed":50,"trajectory_name":"2_file2.txt"}]}</pre>
	返回查询符合的在线编程列表

## (2) 查询在线编程运行状态

功能描述	查询在线编程运行状态
参数说明	get_program_run_state: 查询当前在线编程文件的运行状态
命令格式	{s:s}
示例	<pre>{"command":"get_program_run_state"}</pre>
返回值	<pre>{"command":"get_program_run_state","run_state":1,"id":1,"plan_num":1,"loop_num":[1],"loop_cont":[3],"step_mode":0,"plan_speed":50}</pre>
	<p>参数说明</p> <p>run_state: 0 未开始 1 运行中 2 暂停中</p>



	<p>id:运行轨迹编号，已存储轨迹 的 id，没有存储则为 0，未运行则不返回</p> <p>plan_num：运行到的行数，未运行则不返回</p> <p>loop_num：存在循环指令的行数，未运行则不返回</p> <p>loop_cont：循环指令行数对应的运行次数，未运行则不返回</p> <p>step_mode：单步模式，1 为单步模式，0 为非单步模式</p> <p>plan_speed：全局规划速度比例 1-100</p>
--	--

### (3) 开始运行指定编号轨迹

功能描述	开始运行编号轨迹
参数说明	set_program_id_start：运行指定编号在线编程
命令格式	{s:s,s:i,s:i}
示例	{ "command": "set_program_id_start", "id": 2, "speed": 50 }
	<p>说明：</p> <p>id:运行指定的 ID，1-100，存在轨迹可运行</p> <p>speed:1-100，需要运行轨迹的速度，可不提供速度比例，按照存储的速度运行</p>
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	<p>{ "command": "set_program_id_run", "start_state": true } 开始运行成功</p> <p>{ "command": "set_program_id_run", "start_state": false } 开始运行失败</p> <p>在线编程序结束后，会主动上报结束的 ID</p> <p>{ "state": "program_run_finish", "finish_id": 4 }</p>

### (4) 删除指定编号的轨迹

功能描述	删除指定编号已保存的轨迹
参数说明	delete_program_trajectory：删除指定 ID 的轨迹



命令格式	{s:s,s:i}
示例	{"command":"delete_program_trajectory","id":2}
	说明： id：指定需要删除的轨迹编号，需要在 1-100 之间
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"delete_program_trajectory","delete_state",true} 删除成功 {"command":"delete_program_trajectory","delete_state",false} 删除失败

## 八、UDP 主动上报（I 系列）

### （1）UDP 机械臂状态主动上报接口

功能描述	I 系列机械臂提供 UDP 机械臂状态主动上报接口，使用时，需要和机械臂处于同一局域网络下，通过设置主动上报配置接口的目标 IP 或和机械臂建立 TCP 连接，机械臂即会主动周期性上报机械臂状态数据，广播的默认目标端口为 8089（可配置），使用 UDP 协议监听本机的 8089 端口，即可收到数据，数据周期可配置，默认 5ms。
示例（六自由度）	<pre>{"arm_err":0,"joint_status":{"joint_current":[43000,2085000,102000,1000,257000,-57000],"joint_en_flag":[1,1,1,1,1,1],"joint_err_code":[0,0,0,0,0,0],"joint_position":[13434,-69764,2926,-4742,-45721,-223],"joint_temperature":[33000,35000,37000,36000,3700,39000],"joint_voltage":[22000,22000,22000,22000,22000,22000]},"six_force_sensor":{"force":[-13000,3799,-22393,-216,-408,481],"zero_force":[17476,10415,30827,5,2,2],"coordinate":1},"state":"realtime_arm_joint_state","sys_err":0,"waypoint":{"euler":[2935,2935,2935],"position":[578568,127709,345856],"quat":[-23405,824245,106348,555663]}}</pre>



示例（七自由度）	<pre>{"arm_err":0,"joint_status":{"joint_current":[43000,2085000,102000,1000,257000,-57000,1000],"joint_en_flag":[1,1,1,1,1,1,1],"joint_err_code":[0,0,0,0,0,0,0],"joint_position":[13434,-69764,2926,-4742,-45721,-223,-223],"joint_temperature":[33000,35000,37000,36000,37000,39000,37000],"joint_voltage":[22000,22000,22000,22000,22000,22000,22000]},"six_force_sensor":{"force":[-13000,3799,-22393,-216,-408,481],"zero_force":[17476,10415,30827,5,2,2],"coordinate":1},"state":"realtime_arm_joint_state","sys_err":0,"waypoint":{"euler":[2935,2935,2935],"position":[578568,127709,345856],"quat":[-23405,824245,106348,555663]}}</pre>
参数说明	<p>state: realtime_arm_joint_state 实时机械臂状态上报</p> <p>arm_err: 机械臂错误码</p> <p>sys_err: 系统错误码</p> <p>joint_status: 当前关节状态</p> <p>joint_current: 当前关节电流，精度 0.001mA</p> <p>joint_en_flag: 当前关节使能状态，1 为上使能，0 为掉使能</p> <p>joint_err_code: 当前关节错误码</p> <p>joint_position: 当前关节角度，精度 0.001°</p> <p>joint_temperature: 当前关节温度，精度 0.001℃</p> <p>joint_voltage: 当前关节电压，精度 0.001V</p> <p>waypoint: 当前路点信息</p> <p>position: 当前路点位置，精度 0.000001M</p> <p>euler: 当前路点姿态欧拉角，精度 0.001rad</p> <p>quat: 当前路点四元数，精度 0.000001</p> <p>six_force_sensor: 六维力数据（六维力版本支持）</p> <p>one_force_sensor: 一维力数据（一维力版本支持）</p> <p>force: 当前力传感器原始数据 0.001N 或 0.001Nm</p> <p>zero_force: 当前力传感器系统外受力数据 0.001N 或 0.001Nm</p> <p>coordinate: 系统外受力数据的坐标系，0 为传感器坐标系 1 为当前工作坐标系 2 为当前工具坐标系</p>

## (2) 查询 UDP 机械臂状态主动上报配置



功能描述	查询 UDP 机械臂状态主动上报配置
参数说明	get_realtime_push: 查询 UDP 机械臂状态主动上报配置
命令格式	{s:s}
示例	<pre>{"command":"get_realtime_push"}</pre> <p>说明：用于查询 UDP 机械臂状态主动上报配置</p>
返回值	<p>参数说明：</p> <p>port: 设置广播的端口号</p> <p>cycle: 设置广播周期，为 5ms 的倍数</p> <p>enable: 设置使能，是否使能主动上报</p> <p>force_coordinate: 系统外受力数据的坐标系，0 为传感器坐标系 1 为当前工作坐标系 2 为当前工具坐标系（力传感器版本支持）</p> <p>ip: 自定义的上报目标 IP 地址</p> <pre>{"command":"get_realtime_push","cycle":10,"enable":true,"port":8099,"force_coordinate":2,"ip":"192.168.1.10"}</pre>

### (3) 设置 UDP 机械臂状态主动上报配置

功能描述	设置 UDP 机械臂状态主动上报配置
参数说明	set_realtime_push: 设置 UDP 机械臂状态主动上报配置
命令格式	{s:s,s:i,s:b,s:i}
示例	<pre>{"command":"set_realtime_push","cycle":10,"enable":true,"port":8099,"force_coordinate":2,"ip":"192.168.1.10"}</pre> <p>参数说明：以下参数可分开设置，均为可选字段</p> <p>port: 设置广播的端口号</p> <p>cycle: 设置广播周期，为 5ms 的倍数</p> <p>enable: 设置使能，是否主动上报</p> <p>force_coordinate: 系统外受力数据的坐标系，0 为传感器坐标系 1</p>



	为当前工作坐标系 2 为当前工具坐标系（力传感器版本支持） ip: 自定义的上报目标 IP 地址
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败  {"command":"set_realtime_push","state":true}

## 九、电子围栏 (I 系列)

I 系列机械臂支持管理电子围栏，通过设置电子围栏参数，使机械臂的轨迹规划、示教等运动均只能在规定的电子围栏参数下运动，当以上运动可能超出电子围栏时，系统会报相应错误码，同时中止运动，需要注意的是，以上电子围栏的安全防护功能均目前只在仿真模式下生效，用于进行预演轨迹与轨迹优化。

### (1) 设置电子围栏使能状态

功能描述	设置电子围栏使能状态
参数说明	set_electronic_fencel_enable: 设置电子围栏使能状态
命令格式	{s:s,s:b,s:i,s:i}
示例	<pre>{"command":"set_electronic_fence_enable","set_enable":false,"in_out_side":0,"effective_region":0}</pre> set_enable: true 代表使能, false 代表禁使能 in_out_side: 0-机器人在电子围栏内部, 1-机器人在电子围栏外部 effective_region: 0-针对整臂区域生效
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败  {"command":"set_electronic_fence_enable","set_state":true} 设置成功  {"command":"set_electronic_fence_enable","set_state":false} 设置失败

### (2) 获取电子围栏使能状态





功能描述	获取电子围栏使能状态
参数说明	get_electronic_fencel_enable: 获取电子围栏使能状态
命令格式	{s:s}
示例	<pre>{"command":"get_electronic_fence_enable"}</pre> <p>获取电子围栏使能状态</p>
返回值	<pre>{"command":"get_electronic_fence_enable","enable_state":false,"in_out_side":0,"effective_region":0}</pre> <p>参数说明</p> <p>enable_state: true 代表使能, false 代表禁使能</p> <p>in_out_side: 0-机器人在电子围栏内部, 1-机器人在电子围栏外部</p> <p>effective_region: 0-针对整臂区域生效</p>

### (3) 设置当前电子围栏参数

功能描述	设置当前电子围栏参数配置
参数说明	set_electronic_fence_config: 设置当前电子围栏参数
命令格式	立方体: {s:s,s:i,s:i,s:i,s:i,s:i,s:i,s:i}
示例	<p>立方体:</p> <pre>{"command":"set_electronic_fence_config","form":1,"x_min_limit":500,"x_max_limit":1000,"y_min_limit":500,"y_max_limit":1000,"z_min_limit":500,"z_max_limit":1000}</pre> <p>form: 形状, 1 表示立方体</p> <p>x_min_limit: 立方体基于世界坐标系 X 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>x_max_limit: 立方体基于世界坐标系 X 方向最大位置, 单位 0.001m, 示例中表示 1m</p> <p>y_min_limit: 立方体基于世界坐标系 y 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p>



	<p>y_max_limit: 立方体基于世界坐标系 y 方向最大位置, 单位 0.001 m, 示例中表示 1m</p> <p>z_min_limit: 立方体基于世界坐标系 z 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>z_max_limit: 立方体基于世界坐标系 z 方向最大位置, 单位 0.001 m, 示例中表示 1m</p>
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"set_electronic_fence_config","set_config":false}

#### (4) 获取当前电子围栏参数

功能描述	获取当前电子围栏参数配置
参数说明	get_electronic_fence_config: 获取当前电子围栏参数
命令格式	{s:s}
示例	{"command":"get_electronic_fence_config"}
	获取当前电子围栏参数配置
返回值	<p>立方体:</p> <p>{"command":"get_electronic_fence_config","form":1,"x_min_limit":500,"x_max_limit":1000,"y_min_limit":500,"y_max_limit":1000,"z_min_limit":500,"z_max_limit":1000}</p>
	<p>参数说明</p> <p>form: 形状, 1 表示立方体</p> <p>x_min_limit: 立方体基于世界坐标系 X 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>x_max_limit: 立方体基于世界坐标系 X 方向最大位置, 单位 0.001 m, 示例中表示 1m</p> <p>y_min_limit: 立方体基于世界坐标系 y 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>y_max_limit: 立方体基于世界坐标系 y 方向最大位置, 单位 0.001</p>



	<p>m, 示例中表示 1m</p> <p>z_min_limit: 立方体基于世界坐标系 z 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>z_max_limit: 立方体基于世界坐标系 z 方向最大位置, 单位 0.001m, 示例中表示 1m</p>
--	--

### (5) 增加电子围栏

功能描述	增加当前电子围栏参数 (最多支持 10 个电子围栏)
参数说明	add_electronic_fence_config: 增加当前电子围栏参数
命令格式	立方体: {s:s,s:i,s:s,s:i,s:i,s:i,s:i,s:i}
示例	<p>立方体:</p> <pre>{"command":"add_electronic_fence_config","form":1,"form_name":"my_form","x_min_limit":500,"x_max_limit":1000,"y_min_limit":500,"y_max_limit":1000,"z_min_limit":500,"z_max_limit":1000}</pre> <p>form: 形状, 1 表示立方体</p> <p>form_name: 电子围栏名称, 不超过 10 个字节, 支持字母、数字、下划线</p> <p>x_min_limit: 立方体基于世界坐标系 X 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>x_max_limit: 立方体基于世界坐标系 X 方向最大位置, 单位 0.001m, 示例中表示 1m</p> <p>y_min_limit: 立方体基于世界坐标系 y 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>y_max_limit: 立方体基于世界坐标系 y 方向最大位置, 单位 0.001m, 示例中表示 1m</p> <p>z_min_limit: 立方体基于世界坐标系 z 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>z_max_limit: 立方体基于世界坐标系 z 方向最大位置, 单位 0.001m, 示例中表示 1m</p>



返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"add_electronic_fence_config","add_config":true}

#### (6) 更新电子围栏参数

功能描述	更新当前电子围栏参数 (最多支持 10 个电子围栏)
参数说明	update_electronic_fence_config: 更新当前电子围栏参数
命令格式	立方体: {s:s,s:i,s:s,s:i,s:i,s:i,s:i,s:i}
示例	<p>立方体:</p> <pre>{"command":"update_electronic_fence_config","form":1,"form_name":"my_form","x_min_limit":500,"x_max_limit":1000,"y_min_limit":500,"y_max_limit":1000,"z_min_limit":500,"z_max_limit":1000}</pre>
	<p>form: 形状, 1 表示立方体</p> <p>form_name: 电子围栏名称, 不超过 10 个字节, 支持字母、数字、下划线</p> <p>x_min_limit: 立方体基于世界坐标系 X 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>x_max_limit: 立方体基于世界坐标系 X 方向最大位置, 单位 0.001m, 示例中表示 1m</p> <p>y_min_limit: 立方体基于世界坐标系 y 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>y_max_limit: 立方体基于世界坐标系 y 方向最大位置, 单位 0.001m, 示例中表示 1m</p> <p>z_min_limit: 立方体基于世界坐标系 z 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>z_max_limit: 立方体基于世界坐标系 z 方向最大位置, 单位 0.001m, 示例中表示 1m</p>
返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"update_electronic_fence_config","update_config":true}



### (7) 删除指定电子围栏

功能描述	删除指定电子围栏（最多支持 10 个电子围栏）
参数说明	delete_electronic_fence_config：删除指定电子围栏
命令格式	{s:s,s:s}
示例	{"command":"delete_electronic_fence_config","form_name":"my_form"}
	form_name:电子围栏名称，不超过 10 个字节，支持字母、数字、下划线
返回值	格式：{s:s,s:b}，true-设置成功，false-设置失败
	{"command":"delete_electronic_fence_config","delete_config":true}

### (8) 查询所有电子围栏名称

功能描述	查询所有电子围栏名称（最多支持 10 个电子围栏）
参数说明	get_electronic_fence_list_names：查询所有电子围栏名称
命令格式	{s:s}
示例	{"command":"get_electronic_fence_list_names"}
	查询所有电子围栏名称
返回值	{"command":"get_electronic_fence_list_names","name_list":["a_form","b_form"]}
	<p>参数说明</p> <p>name_list：电子围栏名称列表，长度为实际存在电子围栏数量</p>

### (9) 查询指定电子围栏参数

功能描述	查询指定电子围栏参数（最多支持 10 个电子围栏）
------	---------------------------



参数说明	given_electronic_fence_config: 查询指定电子围栏参数
命令格式	{s:s,s:s}
示例	<pre>{"command":"given_electronic_fence_config","form_name":"my_form"}</pre>
	form_name:电子围栏名称, 不超过 10 个字节, 支持字母、数字、下划线
返回值	<p><b>立方体:</b> <pre>{"command":"given_electronic_fence_config","form":1,"form_name":"my_form","x_min_limit":500,"x_max_limit":1000,"y_min_limit":500,"y_max_limit":1000,"z_min_limit":500,"z_max_limit":1000}</pre></p> <p><b>查询失败:</b> <pre>{"command":"given_electronic_fence_config","given_state":false}</pre></p>
	<p><b>参数说明</b></p> <p>form_name:电子围栏名称, 不超过 10 个字节, 支持字母、数字、下划线</p> <p>form: 形状, 1 表示立方体</p> <p>x_min_limit: 立方体基于世界坐标系 X 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>x_max_limit: 立方体基于世界坐标系 X 方向最大位置, 单位 0.001m, 示例中表示 1m</p> <p>y_min_limit: 立方体基于世界坐标系 y 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>y_max_limit: 立方体基于世界坐标系 y 方向最大位置, 单位 0.001m, 示例中表示 1m</p> <p>z_min_limit: 立方体基于世界坐标系 z 方向最小位置, 单位 0.001m, 示例中表示 0.5m</p> <p>z_max_limit: 立方体基于世界坐标系 z 方向最大位置, 单位 0.001m, 示例中表示 1m</p>

## (10) 查询所有电子围栏信息



功能描述	查询所有电子围栏信息（最多支持 10 个电子围栏）
参数说明	get_electronic_fence_list_infos：查询所有电子围栏信息
命令格式	{s:s}
示例	<pre>{"command":"get_electronic_fence_list_infos"}</pre> <p>获取所有电子围栏信息</p>
返回值	<pre>{"command":"get_electronic_fence_list_infos","info_list":[{"form":1,"form_name":"my_form","x_min_limit":500,"x_max_limit":1000,"y_min_limit":500,"y_max_limit":1000,"z_min_limit":500,"z_max_limit":1000}, {"form":2,"form_name":"my_form","x1":500,"y1":1000,"z1":500,"x2":900,"y2":200,"z2":800,"x3":700,"y3":1200,"z3":300}]}</pre> <p>参数说明</p> <p>info_list：电子围栏信息列表，长度为实际存在电子围栏数量</p>

## 十、自碰撞安全检测（I 系列）

I 系列机械臂支持自碰撞安全检测，自碰撞安全检测使能状态下，可确保在轨迹规划、示教等运动过程中机械臂的各个部分不会相互碰撞，需要注意的是，以上自碰撞安全检测功能目前只在仿真模式下生效，用于进行预演轨迹与轨迹优化。

### （1）设置自碰撞安全检测使能状态

功能描述	设置自碰撞安全检测使能状态，默认不使能
参数说明	set_self_collision_enable：设置自碰撞安全检测使能状态
命令格式	{s:s,s:b}
示例	<pre>{"command":"set_self_collision_enable","set_enable":false}</pre> <p>set_enable：true 代表使能，false 代表禁使能</p>



返回值	格式: {s:s,s:b}, true-设置成功, false-设置失败
	{"command":"set_self_collision_enable","set_state":true}

## (2) 获取自碰撞安全检测使能状态

功能描述	获取自碰撞安全检测使能状态
参数说明	get_self_collision_enable: 获取自碰撞安全检测使能状态
命令格式	{s:s}
示例	{"command":"get_self_collision_enable"}
	获取自碰撞安全检测使能状态
返回值	{"command":"get_self_collision_enable","enable_state":false}
	<p>参数说明</p> <p>enable_state: true 代表使能, false 代表禁使能</p>





## 示例：基础系列高速网口配置步骤

### 1. 开启高速网口

使用 socket 助手通过 TCP 向控制器下发如下 JSON 协议字符串：

```
{"command":"set_high_speed_eth", "mode":1}
```

命令下发成功后，socket 助手会收到控制器反馈内容，如下所示：

```
{"command":"set_high_speed_eth","set_state":true}
```

- 若 set\_state 值返回 true，则设置成功。
- 若 set\_state 值返回 false，则设置失败。

### 2. 配置高速网口 IP

使用 socket 助手通过 TCP 向控制器下发如下 JSON 协议字符串：

```
{"command":"set_high_ethernet","ip":"192.168.1.18","mask":"255.255.255.0",  
"gateway":"192.168.1.1"}
```

命令下发成功后，socket 助手会收到控制器反馈内容，如下所示：

```
{"command": "set_high_ethernet", "state": true}
```

- 若 state 值返回 true，则设置成功。
- 若 state 值返回 false，则设置失败。

### 3. 保存参数

所有配置都需要进行保存参数操作，否则配置失效。



参数保存操作时,使用 socket 助手通过 TCP 向控制器下发如下 JSON 协议字符串:

```
{"command":"save_device_info_all"}
```

命令下发成功后,socket 助手会收到控制器反馈内容,如下所示:

```
{"command":"save_device_info_all","model":true}
```

- 若 model 值返回 true,则设置成功。
- 若 model 值返回 false,则设置失败。

#### 4.更换端口,重启控制器

首先将网线插到高速网口,然后重启控制器。

若在终端执行 ping 192.168.1.18 有握手反馈,则说明设置成功,否则请重复上述步骤或联系售后技术支持。

注意:

- (1) 下发 JSON 协议字符串时,需要添加回车换行。
- (2) 高速网口处于机器人面板中间位置,最快通讯周期是 10ms。



## 附录：错误代码

### 1.系统错误代码

序号	错误代码（16 进制）	错误内容
1	0x0000	系统正常
2	0x1001	关节通信异常
3	0x1002	目标角度超过限位
4	0x1003	该处不可达，为奇异点
5	0x1004	实时内核通信错误
6	0x1005	关节通信总线错误
7	0x1006	规划层内核错误
8	0x1007	关节超速
9	0x1008	末端接口板无法连接
10	0x1009	超速度限制
11	0x100A	超加速度限制
12	0x100B	关节抱闸未打开
13	0x100C	拖动示教时超速
14	0x100D	机械臂发生碰撞
15	0x100E	无该工作坐标系
16	0x100F	无该工具坐标系
17	0x1010	关节发生掉使能错误
18	0x1011	圆弧规划错误
19	0x1012	自碰撞错误



20	0x1013	碰撞到电子围栏错误
21	0x1014	超关节软限位错误
22	0x2001	夹爪异常
23	0x2002	灵巧手异常
24	0x2003	六维力模块异常
25	0x2004	一维力模块异常
26	0x2005	输出电流异常
27	0x5001	预留
28	0x5002	预留
29	0x5003	控制器过温
30	0x5004	预留
31	0x5005	控制器过流
32	0x5006	控制器欠流
33	0x5007	控制器过压
34	0x5008	控制器欠压
35	0x5009	实时内核通讯错误

## 2. 关节错误代码

序号	错误代码 (16 进制)	错误内容
1	0x0000	关节正常
2	0x0001	FOC 错误
3	0x0002	过压
4	0x0004	欠压



5	0x0008	过温
6	0x0010	启动失败
7	0x0020	编码器错误
8	0x0040	过流
9	0x0080	软件错误
10	0x0100	温度传感器错误
11	0x0200	位置超限错误
12	0x0400	关节 ID 非法
13	0x0800	位置跟踪错误
14	0x1000	电流检测错误
15	0x2000	抱闸打开失败
16	0x4000	位置指令阶跃警告
17	0x8000	多圈关节丢圈数
18	0xF000	通信丢帧