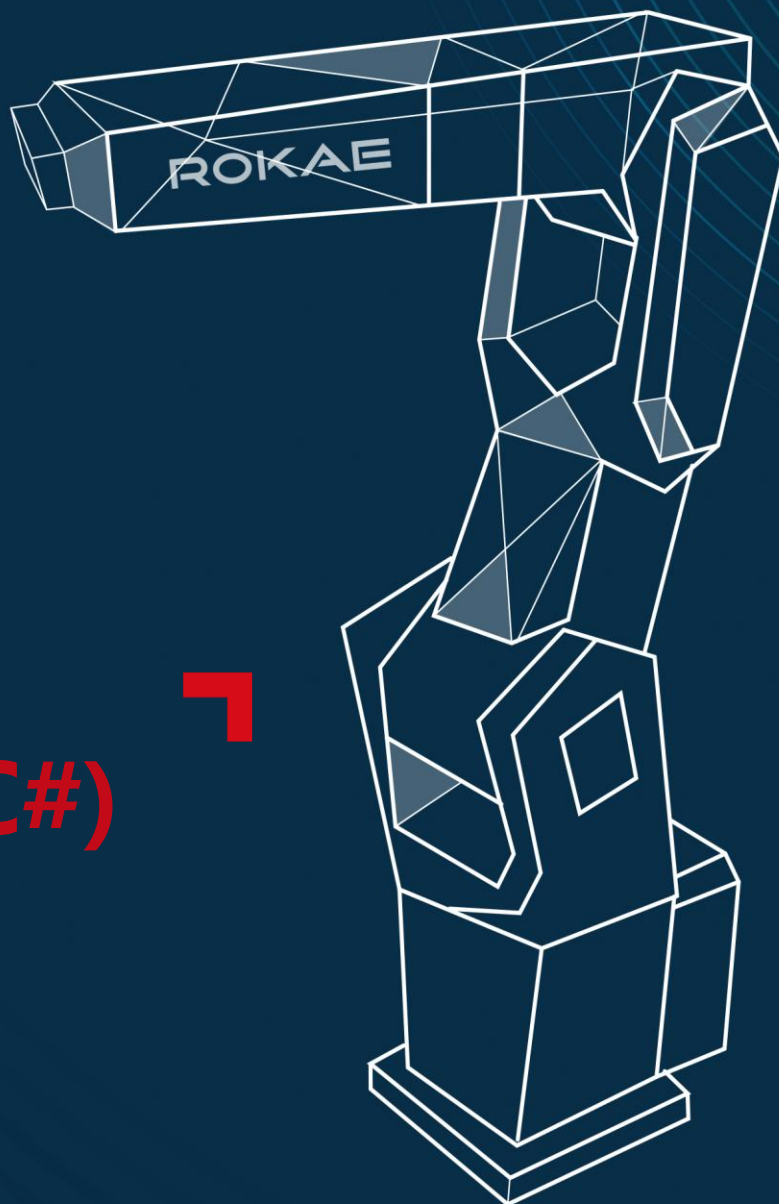


ROKAE 珞石
轻型机器人专家



xCore SDK(C#)

使用手册

让智造更高效

xCore SDK(C#) 使用手册

[类别]

[备注]

控制系统版本：V3.0.1

文档版本：A

本手册中记载的内容如有变更，恕不事先通告。本公司对手册中可能出现的错误均不承担任何责任。

本公司对因使用本手册及其中所述产品而引起的意外或间接伤害均不承担任何责任，敬请谅解。

本公司不可能预见所有的危险和后果，因此本手册不能警告用户所有可能的危险。

禁止擅自复印或转载本手册的部分或全部内容。

如您发现本手册的内容有误或需要改进抑或补充之处，请不吝指正。

本手册的原始语言为中文，所有其他语言版本均翻译自中文版本。

©版权所有 2015-2024 ROKAE 保留所有权利

珞石（北京）智能科技有限公司

中国.北京

目录

1 手册概述	5
1.1 关于本手册	5
1.2 手册对象	5
1.3 如何阅读产品手册	5
1.4 版本记录	5
2 概述	7
2.1 兼容性	7
2.1.1 控制器版本和机器人型号	7
2.1.2 编译	7
3 使用指南	8
3.1 设置	8
3.1.1 硬件设置	8
3.1.2 网络配置	8
3.1.3 机器人功能设置	8
3.2 xCore SDK 使用方法	8
3.2.1 工程包说明	8
3.2.2 使用方法	8
3.3 运行日志	8
3.4 语言	9
4 接口说明	10
4.1 实例化机器人类	10
4.2 机器人基本操作及信息查询	10
4.3 运动控制	11
4.4 通信相关	12
4.5 RL 工程	13
4.6 协作相关	13
4.7 力控指令	14
4.8 错误码和异常	15
4.8.1 错误码	15
4.8.2 异常	20
5 注意事项	22
5.1 与 RobotAssist 同时使用	22
6 使用示例	23

示例一：机器人基本操作，信息查询，Jog，拖动等.....	23
示例二：运动指令	24
7 反馈与勘误.....	25
8 附录 A – C# API.....	26
8.1 枚举类型.....	26
8.1.1 机器人工作状态 rokae.clr.OperationState 机器人工作状态.....	26
8.1.2 机器人操作模式 rokae.clr.OperateMode.....	26
8.1.3 机器人上下电及急停状态 rokae.clr.PowerState	26
8.1.4 位姿坐标系类型 rokae.clr.CoordinateType	26
8.1.5 运动控制模式 rokae.clr.MotionControlMode.....	26
8.1.6 机器人停止运动等级 rokae.clr.StopLevel	26
8.1.7 机器人拖动模式参数 rokae.clr.DragOpt.....	26
8.1.8 坐标系类型 rokae.clr.FrameType.....	27
8.1.9 Jog 选项 - 坐标系 rokae.clr.JogSpace.....	27
8.1.10 奇异规避方式 rokae.clr.AvoidSingularityMethod	27
8.1.11 运动指令笛卡尔点位偏移类型 rokae.clr.CartesianPosition.Offset.Type	27
8.1.12 运动指令类型 rokae.clr.MoveCommand.Type.....	27
8.1.13 xPanel 配置: 对外供电模式 rokae.clr.xPanelOpt.Vout	27
8.1.14 事件类型 rokae.clr.Event.....	28
8.2 数据结构.....	28
8.2.1 机器人基本信息 rokae.clr.Info	28
8.2.2 坐标系 rokae.clr.Frame.....	28
8.2.3 笛卡尔点位 rokae.clr.CartesianPosition	28
8.2.4 运动指令笛卡尔点位偏移量 rokae.clr.CartesianPosition.Offset.....	28
8.2.5 关节点位 rokae.clr.JointPosition	28
8.2.6 运动指令 rokae.clr.MoveCommand	28
8.2.7 负载信息 rokae.clr.Load.....	29
8.2.8 工具工件组信息 rokae.clr.Toolset.....	29
8.2.9 坐标系标定结果 rokae.clr.FrameCalibrationResult.....	29
8.2.10 RL 工程信息 rokae.clr.RLProjectInfo	29
8.2.11 工具/工件信息 rokae.clr.WorkToolInfo	29
8.2.12 控制器日志信息 rokae.clr.LogInfo	29
8.2.13 错误码 rokae.clr.ErrorCode.....	29
8.3 方法.....	29
8.3.1 机器人基本操作及信息查询.....	30
8.3.2 运动控制.....	39

8.3.3 通信相关.....44

8.3.4 RL 工程48

8.3.5 协作相关.....49

8.3.6 力控指令.....51

1 手册概述

1.1 关于本手册

感谢您购买本公司的机器人系统。

本手册记载了正确安装使用机器人的以下说明：

- 机器人二次开发接口 SDK(C#)的使用。

安装使用该机器人系统前，请仔细阅读本手册与其他相关手册。

阅读之后，请妥善保管，以便随时取阅。

1.2 手册对象

本手册面向：

- 机器人应用开发工程师。

请务必保证以上人员具备基础的机器人操作、C#编程等所需的知识，并已接受本公司的相关培训。

1.3 如何阅读产品手册

本手册包含单独的安全章节，必须在阅读安全章节后，才能进行安装或维护作业。

1.4 版本记录

版本编号	日期	说明
V1.0	2023.11	第一版，适配 xCore 版本 v2.1
V2.0	2024.04	增加接口若干，适配 xCore 版本 v2.2
V3.0	2024.12	增加接口若干，适配 xCore 版本 v3.0.1+

2 概述

xCore SDK 编程接口库是珞石机器人提供给客户用于二次开发的软件产品，通过编程接口库，客户可以对配套了 xCore 系统的机器人进行一系列控制和操作，包括运动控制，机器人通信相关的读写操作，查询及运行 RL 工程，等等。该使用说明书主要介绍编程接口库的使用方法，以及各接口函数的功能。用户可编写自己的应用程序，集成到外部软硬件模块中。

2.1 兼容性

2.1.1 控制器版本和机器人型号

- 控制器版本：xCore v3.0.1 及以后。
- 机器人型号：支持控制所有机型，根据协作和工业机器人支持的功能不同，可调用的接口有所差别。

2.1.2 编译

- 支持/clr 编译选项，不支持/clr:netcore

3 使用指南

本章介绍如何配置并运行一个 xCore SDK C#程序。

3.1 设置

3.1.1 硬件设置

关于机器人本体和控制柜等硬件的设置，请参考《xCore 控制系统使用手册 V3.0.1》。除了网络配置外，使用 xCore SDK 无需其他额外的硬件设置。

3.1.2 网络配置

xCore SDK 通过以太网 (TCP/IP) 连接机器人，C#只包含非实时接口，对网络的要求并不高，通过有线或无线连接皆可，使工作站 PC 和机器人连接同一局域网。

3.1.3 机器人功能设置

无需通过 RobotAssist 进行任何设置，用户可直接用 xCore SDK 控制机器人。

3.2 xCore SDK 使用方法

3.2.1 工程包说明

```
xCoreSDK_CSharp
├─ xCoreSDK_CSharp
│   └─ xcore.lib: DLL 文件及 XML 文档
├─ Program.cs, xxxDemo.cs: 示例程序
└─ xCoreSDK_CSharp.sln: VS solution
```

3.2.2 使用方法

使用 Microsoft Visual Studio 打开 xCoreSDK_CSharp.sln，建议 VS 版本不低于 2017

3.3 运行日志

运行 SDK 程序并连接到机器人之后，SDK 会在调用设置数据、执行操作等接口时打印日志到文件，包含调用参数和返回结果等信息。可在遇到问题时将日志提供给珞石公司的技术支持人员，以便分析排查。

目前暂不支持用户配置日志，默认的设置如下：

- 文件路径：<当前工作目录>/_rokae_log_/<机器人 uid>/xCoreSDK_<YYYY-MM-DD>.log
- 最多保留 7 天的日志

目前支持用户自定义配置日志的输出位置和保留天数，需要用户在程序运行目录处放置一个名为 user_log_settings.json 的文件，文件结构如下：

```
{
  "_log_storage_directory_": "user_defined_log_storage_directory",
  "_retention_days_": 5
}
```

_log_storage_directory_ 为用户指定的日志存储目录路径，替换默认日志存储目录 _rokae_log_，建议使用相对

路径的方式写入；_retention_days_ 为日志保留天数。

目前也提供了全局日志，全局日志的输出位置固定在运行程序所在目录下名为 logs 的文件夹，最多保留 7 天的全局日志。

3.4 语言

xCore SDK 的错误码信息和异常信息支持中文和英文，根据用户 PC 设置的系统语言而定。设置显示中文则返回中文错误信息，非中文则返回英文错误信息

4 接口说明

本章列出 xCore SDK C#版本所支持的接口和功能简述。

4.1 实例化机器人类

根据机器人构型和轴数不同，SDK 提供了下列几个可供实例化的 Robot 类，连接到机器人时会检查所选构型和轴数是否和连接的机器人匹配：

类名	适用机型
xMateRobot	协作 6 轴
xMateErProRobot	协作 7 轴
StandardRobot	工业 6 轴
xMateCr5Robot	协作 5 轴
PCB4Robot	工业 4 轴
PCB3Robot	工业 3 轴

4.2 机器人基本操作及信息查询

简述	接口	参数	返回
连接机器人	connectToRobot()		
	connectToRobot(remoteIP)	remoteIP - 机器人 IP 地址	
	connectToRobot(remoteIP, localIP)	remoteIP - 机器人 IP 地址 localIP - 本机地址	
断开连接	disconnectFromRobot()		
查询机器人基本信息	robotInfo()		控制器版本, 机型, 轴数
查询上电状态	powerState()		on/off/Estop/Gstop
机器人上下电	setPowerState(state)	state - on/off	
查询当前操作模式	operateMode()		auto/manual
切换手自动模式	setOperateMode(mode)	mode - auto/manual	
查询机器人运行状态	operationState()		idle/jog/RLprogram/moving 等状态
获取当前末端/法兰位姿	posture()		[X, Y, Z, Rx, Ry, Rz]
获取当前末端/法兰位姿	cartPosture()		[X, Y, Z, Rx, Ry, Rz] 及轴配置参数
获取当前关节角度	jointPos()		各轴角度 rad
获取当前关节速度	jointVel()		各轴速率 rad/s
获取关节力矩	jointTorque()		各轴力矩 Nm
以最高 1KHz 读取机器人状态数据	getStateDate(name, data)	name - 数据名 data - 数据值	读取成功与否
停止接收实时状态数据	stopReceiveRobotState()		
接收一次机器人状态数据	updateRobotState(timeout)	timeout - 超时时间[ms]	接收到的数据长度
查询基坐标系	baseFrame()		[X, Y, Z, Rx, Ry, Rz]
设置基坐标系	setBaseFrame(frame)	frame - 坐标系	
查询当前工具工件组	toolset()		末端坐标系, 参考坐标系, 负载信息
设置工具工件组	setToolset(toolset)	toolset - 工具工件组信息	

	setToolset(toolName, wobjName)	toolName - 工具名字 wobjName - 工件名字	
计算逆解	calclk(posture) calclk(posture, toolset)	posture - 末端相对于外部参考坐标系位姿 toolset - 工具工件组信息	关节角度
计算正解	calcFk(joints) calcFk(joints, toolset)	joints - 关节角度 toolset - 工具工件组信息	末端相对于外部参考坐标系位姿
清除伺服报警	clearServoAlarm()		
查询控制器日志	queryControllerLog(count, level)	count - 查询个数 level - 日志等级	
设置碰撞检测相关参数, 打开碰撞检测功能	enableCollisionDetection(sensitivity, behaviour, fallback)	sensitivity - 灵敏度 behaviour - 碰撞后行为 fallback - 回退距离	
关闭碰撞检测功能	disableCollisionDetection()		
坐标系标定	calibrateFrame(type, points, is_held, base_aux)	type - 坐标系类型 points - 标定轴角度列表 is_held - 手持/外部工具工件 base_aux - 基坐标系标定辅助点	标定结果: 坐标系和偏差
获取当前软限位数值	getSoftLimit(limits)	limits - 各轴软限位	已打开/已关闭
设置软限位	setSoftLimit(enable, limits)	enable - 打开/关闭 limits - 各轴软限位	
恢复状态	recoverState(item)	item - 恢复选项 1: 急停恢复	
设置导轨参数	setRailParameter(name, value)	name - 参数名 value - 参数值	
读取导轨参数	getRailParameter(name, value)	name - 参数名 value - 参数值	
配置 NTP	configNtp(server_ip)	server_ip - NTP 服务端 IP	
手动同步一次 NTP 时间	syncTimeWithServer()		
查询 SDK 版本号	sdkVersion()		版本号

4.3 运动控制

简述	接口	参数	返回
设置运动控制模式	setMotionControlMode(mode)	mode - NRT/RT/RL 工程	
开始/继续运动	moveStart()		
重置运动缓存	moveReset()		
停止机器人运动	stop()		
添加运动指令	moveAppend(type, command, id)	type - 指令类型 command - 一条或多条同类型指令 id - 指令 ID, 用于执行信息反馈	
设置默认运动速度	setDefaultSpeed(speed)	speed - 末端最大线速度	

设置默认转弯区	setDefaultZone(zone)	zone - 转弯区半径	
设置是否使用 conf	setDefaultConfOpt(forced)	forced - 是/否使用	
设置最大缓存指令个数	setMaxCacheSize(number)	number - 个数	
开始 Jog 机器人	startJog(space, rate, step, index, direction)	space - 参考坐标系 rate - 速率 step - 步长 index - XYZABC/J1-7 direction - 方向	
动态调整机器人运动速率	adjustSpeedOnline(scale)	scale - 速率	
设置接收事件的回调函数	setEventWatcher(eventType, callback)	eventType - 事件类型 callback - 处理事件的回调函数	
查询事件信息	queryEventInfo(eventType)	eventType - 事件类型	事件信息
执行运动指令	executeCommand(type, command)	type - 指令类型 command - 一条或多条同类型指令	
读取当前加速度	getAcceleration(acc, jerk)	acc - 加速度 jerk - 加加速度	
设置运动加速度	adjustAcceleration(acc, jerk)	acc - 加速度 jerk - 加加速度	
打开奇异规避功能	setAvoidSingularity(method, enable, threshold)	method - 奇异规避方式 enable - 打开/关闭 threshold - 阈值参数	
查询是否打开规避奇异功能	getAvoidSingularity(method)	method - 奇异规避方式	已打开/已关闭
检验笛卡尔轨迹是否可达	checkPath(start, start_joint, target)	start - 起始点 start_joint - 起始关节角度 target - 目标点	计算出的目标轴角

4.4 通信相关

简述	接口	参数	返回值
查询 DI 信号值	getDI(board, port)	board - IO 板序号 port - 信号端口号	on off
设置 DI 信号值	setDI(board, port, state)	board - IO 板序号 port - 信号端口号 state - 信号值	
查询 DO 信号值	getDO(board, port)	board - IO 板序号 port - 信号端口号	on off
设置 DO 信号值	setDO(board, port, state)	board - IO 板序号 port - 信号端口号 state - 信号值	
查询 AI 信号值	getAI(board, port)	board - IO 板序号 port - 信号端口号	信号值
设置 AO 信号	setAO(board, port, value)	board - IO 板序号 port - 信号端口号 value - 信号值	
设置输入仿真模式	setSimulationMode(state)	state - 打开/关闭	

读取寄存器值	readRegister(name, value) readRegister(name, index, value)	name - 寄存器名称 index - 寄存器数组索引 value - 读取的数组/数值	
写入寄存器值	writeRegister(name, index, value)	name - 寄存器名称 index - 寄存器数组索引 value - 写入的数值	
设置 xPanel 对外供电模式	setxPanelVout(opt)	opt - 模式	
获取末端按键状态	getKeypadState()		末端按键的状态
末端工具 485 通信开关	setxPanelRS485(Vopt, if_rs485)	Vopt - 输出电压 if_rs485 - 是否打开 485 通信	
485 通信读写寄存器	XPRWModbusRTUReg(slave_addr, fun_cmd, reg_addr data_type, num, data_array, lif_crc_reverse)	slave_addr - 从站地址 fun_cmd - 功能码 reg_addr - 寄存器地址 data_type - 数据类型 num - 数据长度 data_array - 数据数组 if_crc_reverse - 是否需要 CRC 反转	
485 通信读写线圈或离散输入	XPRWModbusRTUCoil(slave_addr, fun_cmd, coil_addr, int num, data_array, if_crc_reverse)	slave_addr - 从站地址 fun_cmd - 功能码 coil_addr - 线圈地址 num - 数据长度 data_array - 数据数组 if_crc_reverse - 是否需要 CRC 反转	
485 通信裸传数据	XPRS485SendData(send_byte, rev_byte, send_data, rev_data,)	send_byte - 发送数据长度 rev_byte - 接收数据长度 send_data - 发送数据 rev_data - 接收数据	

4.5 RL 工程

控制器中需要有已创建好的 RL 工程，支持查询工程信息和运行。

简述	接口	参数	返回值
查询 RL 工程列表	projectInfo()		工程名称和任务名
加载工程	loadProject(name, tasks)	name - 工程名称 tasks - 任务列表	
pp-to-main	ppToMain()		
开始运行工程	runProject()		
暂停运行工程	pauseProject()		
设置运行速率和循环模式	setProjectRunningOpt(rate, loop)	rate - 运行速率 loop - 循环/单次	
查询工具信息	toolsInfo()		工具名称, 位姿, 负载等信息
查询工件信息	wobjsInfo()		工件名称, 位姿, 负载等信息

4.6 协作相关

包括拖动示教和路径录制相关功能。

简述	接口	参数	返回值
打开拖动	enableDrag(space, type, enable_drag_button)	space - 拖动空间 type - 拖动类型 enable_drag_button - 无需按键拖动	
关闭拖动	disableDrag()		
开始录制路径	startRecordPath(duration)	duration - 录制时长	
停止录制路径	stopRecordPath()		
取消录制	cancelRecordPath()		
保存路径	saveRecordPath(name, saveAs)	name - 路径名称 saveAs - 重命名为	
路径回放	replayPath(name, rate)	name - 路径名称 rate - 回放速率	
删除保存的路径	removePath(name, all)	name - 路径名称 all - 是否删除所有路径	
查询路径列表	queryPathLists()		路径名称列表
力传感器标定	calibrateForceSensor(all_axes, axis_index)	all_axes - 标定所有轴 axis_index - 单轴标定下标	

4.7 力控指令

简述	接口	参数	返回值
获取当前力矩信息	getEndTorque(ref_type, joint, external, cart_torque, cart_force)	ref_type - 力矩相对的参考系 joint - 各轴测量 external - 各轴外部力 cart_torque - 笛卡尔空间力矩 cart_force - 笛卡尔空间力	
力控初始化	fcInit(frame_type)	frame_type - 力控坐标系	
开始力控	fcStart()		
停止力控	fcStop()		
设置阻抗控制类型	setControlType(type)	type - 阻抗类型	
设置力控模块使用的负载	setLoad(load)	load - 负载	
设置关节阻抗刚度	setJointStiffness(stiffness)	stiffness - 刚度	
设置笛卡尔阻抗刚度	setCartesianStiffness(stiffness)	stiffness - 刚度	
设置笛卡尔零空间阻抗刚度	setCartesianNullspaceStiffness(stiffness)	stiffness - 刚度	
设置关节期望力矩	setJointDesiredTorque(torque)	torque - 力矩值	
设置笛卡尔期望力/力矩	setCartesianDesiredForce(value)	value - 期望力/力矩	
设置绕单轴旋转的正弦搜索运动	setSineOverlay(line_dir, amplify, frequency, phase, bias)	line_dir - 参考轴 amplify - 幅值 frequency - 频率 phase - 相位	

		bias - 偏置	
设置平面内的莉萨如搜索运动	setLissajousOverlay (int plane, double amplify_one, double frequency_one, double amplify_two, double frequency_two, double phase_diff, error_code &ec)	plane - 参考平面 amplify_one - 一方向幅值 frequency_one - 一方向频率 amplify_two - 二方向幅值 frequency_two - 二方向频率 phase_diff 相位偏差	
开启搜索运动	startOverlay()		
停止搜索运动	stopOverlay()		
暂停搜索运动	pauseOverlay()		
重新开启暂停的搜索运动	restartOverlay()		
设置与接触力有关的终止条件	setForceCondition (range, isInside, timeout)	range - 力限制 isInside - 超出/符合限制条件时停止等待 timeout - 超时时间	
设置与接触力矩有关的终止条件	setTorqueCondition (range, isInside, timeout)	range - 力矩限制 isInside - 超出/符合限制条件时停止等待 timeout - 超时时间	
设置与接触位置有关的终止条件	setPoseBoxCondition(supervising_frame, box, isInside, timeout)	supervising_frame - 长方体所在的参考坐标系 box - 长方体 isInside - 超出/符合限制条件时停止等待 timeout - 超时时间	
激活设置的终止条件并等待	waitCondition()		
启动/关闭力控模块保护监控	fcMonitor(enable)	enable - 打开 关闭	
设置力控模式下的轴最大速度	setJointMaxVel(velocity)	velocity - 轴速度	
设置机械臂末端相对基坐标系的轴最大速度	setCartesianMaxVel(velocity)	velocity - 末端速度	
设置力控模式下轴最大动量	setJointMaxMomentum(momentum)	momentum - 动量	
设置力控模式下轴最大动能	setJointMaxEnergy(energy)	energy - 动能	

4.8 错误码和异常

4.8.1 错误码

接口的调用结果通过错误码反馈，每个接口都会传入一个错误码 `rokae.clr.ErrorCode`，可通过 `message` 属性来获取错误码对应的信息。

数值	错误信息	原因及处理方法
0	操作成功完成	无
-1	发生错误	可能由于未识别的错误码，请反馈技术支持人员
-3	机器人急停按钮被按下，请先恢复	恢复急停状态
-16	该操作不允许在当前模式下执行(手动/自动)，请切换到另一个模式	切换手动、自动模式
-17	该操作不允许在当前上下电状态下执行	切换上下电状态

-18	该操作不允许在机器人当前运行状态下执行	机器人非空闲, 可能处于拖动/实时模式控制/辨识中等
-19	该操作不允许在当前控制模式(位置/力控)下执行	切换位置控制/力控
-20	机器人运动中	停止机器人运动
-32	计算逆解错误	传入正确位姿
-33	逆解是奇异点	规避奇异点
-34	逆解超出机器人软限位	检查软限位设置时候合适, 传入限位内位姿
-35	目标点超出运动范围	传入可达点位
-36	目标点超出运动范围	传入可达点位
-37	单步距离过大, 无法计算逆解	传入可达点位
-38	配置数据 cfx 错误逆解无解	检查 conf data
-39	输入数据错误, 无法计算逆解	传入可达点位
-40	逆解参考点是奇异点	传入可达点位
-41	算法失效, 无法计算逆解	可能由于控制器计算问题, 请反馈技术支持人员
-257	通信协议解析失败	请反馈技术支持人员
-258	未找到匹配的数据名称	可能由于控制器和 SDK 版本不匹配
-259	未找到匹配的指令名称	可能由于控制器和 SDK 版本不匹配
-260	数据值错误, 可能是通信协议不匹配	可能由于控制器和 SDK 版本不匹配
-272	未找到要查询的数据名称	可能由于控制器和 SDK 版本不匹配
-273	数据不可读	可能由于控制器和 SDK 版本不匹配
-288	数据不可写	可能由于控制器和 SDK 版本不匹配
-289	设置数据失败	设置的数据不合理
-290	设置数据失败	设置的数据不合理
-291	监控数据失败	不会发生
-292	数据已被监控, 不支持重复监控	不会发生
-320	指令执行未完成	不会发生
-337	录制的路径过短, 数据点不足, 请重新录制	增长录制时间重新录制
-338	录制的路径中存在超过关节软限位的情况。请重新录制	在软限位内拖动
-339	录制的路径中存在关节速度过快的情况, 请重新录制	放慢拖动动作
-340	未从机器人静止状态开始录制路径	机器人静止时再开始录制
-341	停止录制路径时机器人未停止运动	机器人静止时再停止录制
-342	机器人处于下电状态, 保存路径失败	机器人上电
-352	缓冲区不存在有效的路径, 请录制	先录制路径再保存
-353	保存路径到磁盘失败	重新录制拖动轨迹, 或重启机器人
-513	切换手动/自动操作模式失败	- 停止机器人运动 - 恢复急停 - 关闭拖动
-514	上下电失败	- 恢复急停 - 清除伺服报警
-515	打开/关闭拖动失败, 请检查机器人是否处于下电	机器人手动模式下电时打开拖动
-10005	标定中, 或标定时重置负载失败	- 等待标定完成 - 正确设置负载
-10030	未打开仿真模式或信号不存在	打开仿真模式后再设置 DI/AI
-10040	开始拖动失败, 正确设置负载并标定力矩传感器	正确地设置工具负载质量和质心; 设置完成后执行力矩传感器标定
-10051	软限位超出机械限位	设置机械限位内的软限位
-10065	同步 NTP 时间失败	- 安装好 NTP 功能 确认服务端 NTP 功能可用, 确认 IP 地址正确
-10101	IP 地址非法	检查是否符合 IPv4 地址格式

-10079	力控模块处于错误状态	触发了力控保护, 请检查力控模式下机器人状态是否正常, 并设置合理的力控保护参数
-10141	负载质量超过了机器人额定负载	使用并设置额定负载范围内的负载
-14010	已绑定为系统 IO	- 取消绑定改信号为系统 IO - 使用其它信号
-14501	DO 信号不存在或为系统输出	检查 DO 信号是否已创建
-17001	读寄存器错误, 寄存器不存在或数据类型不符	- 检查寄存器是否创建 - 是否超数组下标 - 用匹配的数据类型读取
-17002	写寄存器错误, 寄存器不存在或数据类型不符	- 检查寄存器是否创建 - 是否超数组下标 - 或用匹配的数据类型写入
-17320	执行过 RL 程序, 需要重置运动缓存后再开始运动	调用 moveReset 重置
-17407	开启导轨需关闭安全区域	关闭安全区域功能
-28672	实时模式网络错误	- 本机 IP 地址不能为 localhost 或机器人地址 - 存在端口占用情况, 请检查 RCI 设置-端口
-28688	切换运动控制模式失败	- 停止机器人运动 - 重启控制器
-28689	该频率不支持	状态数据发送频率支持 1kHz, 500Hz, 250Hz, 125Hz
-28705	已经开始运动	停止运动后再开始
-28706	无法执行该操作, 可能由于机器人未处于空闲状态	停止机器人运动
-28707	起始位置为奇异点	运动机器人到非奇异点
-28708	参数错误	重新设置数据
-28709	机器人处于碰撞停止	上电恢复碰撞状态
-28710	机器人处于急停状态	恢复急停
-28711	请求被拒绝	阻抗控制时多由于力控模型偏差, 重新标定力矩传感器并设置正确的负载
-41400	尚有力控指令未完成	等待前序力控指令执行完毕
-41419	TCP 长度超限, 力控初始化失败	末端工具长度限制为 0.3m
-41420	未进行力控初始化; 或未正确设置负载; 或力控模型偏差较大	- 标定零点、力矩传感器; - 设置正确的负载质量和质心; - 检查基坐标系是否正确设置; - 开启拖动时机器人没有收到外力
-41421	停止力控失败, 当前不处于力控运行状态	机器人不处于力控状态
-41422	重新开启力控失败	先暂停或停止力控, 再重新开启
-41425	非阻抗控制模式, 或搜索运动在运行中	- 保证当前处于阻抗模式下, 且搜索运动处于停止状态 - 请检查参数设置
-41426	非阻抗控制模式, 或搜索运动在运行中	- 保证当前处于阻抗模式下, 且搜索运动处于停止状态 - 请检查参数设置
-41427	当前不处于笛卡尔阻抗控制模式或未设置搜索运动	检查控制模式指令, 设置搜索运动参数后再尝试
-41428	当前不处于笛卡尔阻抗控制模式或未开始搜索运动	请保证搜索运动已开启且处于笛卡尔阻抗运行状态
-41429	当前未处于笛卡尔阻抗模式或搜索运动未暂停	暂停当前的力控任务并检查控制模式, 切换到笛卡尔阻抗控制模式后再尝试
-41430	不支持的阻抗控制类型或力控坐标系	停止当前的力控任务并重新参数初始化
-41431	当前未处于关节阻抗控制模式或未初始化	运行力控前设置轴空间阻抗控制模式
-41432	当前未处于笛卡尔阻抗控制模式或未初始化	运行力控前设置笛卡尔阻抗控制模式
-41433	当前未处于笛卡尔阻抗控制模式或未初始化	检查当前的阻抗控制模式后再进行尝试。确保力值输入正确, 且设定笛卡尔阻抗控制模式

-41434	关节期望力超过限制	检查当前的阻抗控制模式后再进行尝试。确保期望力值输入正确，且设定轴空间阻抗控制模式
-41435	笛卡尔空间期望力超过限制	检查当前的阻抗控制模式后再进行尝试。确保期望力值输入正确，且设定笛卡尔阻抗控制模式
-41442	机器人不满足软限位要求，开启力控失败	请确认机器人开启阻抗时机器人在软限位内
-41444	阻抗刚度设置失败，数值不合理或当前状态不能设置刚度	请重新设置阻抗刚度数值在合理范围内
-41448	力控过程中下电，初始化失败	重新上电并执行力控初始化
-41449	力控指令尚未执行完毕，初始化失败	力控指令发送太频繁，请增加退出力控到重新开启之间的时间
-41457	执行过力控停止，需要重新开始	执行力控初始化
-41459	处于力控模式，不允许执行的操作	停止力控后再回放路径
-50000	两条轨迹夹角过小或长度过短，转弯区无法生成	- 增大前后两条轨迹夹角; - 增大前后两条轨迹长度; - 增大转弯区数值
-50001	控制器状态错误，无法生成轨迹	调用 moveReset()重置
-50002	目标点超出运动范围，或为奇异点	- 检查目标点位置; - 用关节方式移动机器人 - 检查 confdata 配置
-50003	相邻两个目标点过近	检查相邻两条运动指令的目标点是否是同一个点
-50004	无法生成圆弧，起始点和目标点距离过近	调整圆弧起点或终点点位之间的距离
-50005	无法生成圆弧，起始点和辅助点距离过近	调整圆弧起点或辅助点点位之间的距离
-50006	无法生成圆弧，辅助点和目标点距离过近	调整圆弧终点或辅助点点位之间的距离
-50007	无法生成圆弧，起始点/辅助点/目标点距离过近	调整圆弧起点、辅助点、终点点位之间的距离
-50008	无法生成圆弧，点位在一条直线	调整圆弧起点、辅助点、终点点位之间的距离
-50009	无法生成圆弧，半径过小	调整圆弧起点、辅助点、终点点位之间的距离
-50010	无法生成圆弧	调整圆弧起点、辅助点、终点点位之间的距离或方位
-50019	生成轨迹失败	重新调整目标点的位置、姿态和臂角(仅 7 轴机器人需要考虑臂角)
-50021	指定 conf 参数下目标点无解，请检查数值或不设置 confData	- 调用 setDefaultConfOpt(false) 取消 confdata - 重新示教点位，传入正确的 confdata
-50027	轨迹短于最小转弯区半径，自动拼接路径	- 该轨迹需要前后都衔接转弯区，但是轨迹长度小于最小转弯区半径的两倍; - 该轨迹设定衔接一条转弯区，但是轨迹长度小于最小转弯区半径。 该功能可以使得运动更平滑，如想关闭该功能，可将最小转弯区半径设为 0
-50033	机器人处于锁轴状态，目标点锁轴角度发生了偏离，请调整目标点或关闭锁轴状态	调整轨迹目标点或者关闭锁轴指令，锁轴状态下四轴角度要求为 0 度或 180 度或 -180 度
-50034	锁轴状态下或 5 轴机型无法到达目标点	6 轴机型在锁轴状态下或 5 轴机型无法到达目标点，请更改点位或关闭锁轴
-50101	轴角度超出运动范围，尝试取消软限位后恢复各轴到允许的范围内	- 取消软限位 - 手动将机器人各轴移动到正常的工作范围内
-50102	存在穿越奇异点的轨迹	- 请规避奇异点 - 重新示教点位，更换目标点; - 将笛卡尔空间运动指令改为关节空间运动指令
-50103	笛卡尔路径终点不符合给定的 ConfData	- 调用 setDefaultConfOpt(false)不使用 confdata - 改为 MoveJ 或 MoveAbsJ - 更改目标点 confdata

-50104	关节力矩超限	- 检查负载数值是否符合实际负载情况 - 检查机器人摩擦力系数、电机过载系数、传动过载系数等参数 - 尝试更换指令形式，例如将笛卡尔空间指令更换为关节空间指令
-50112	位姿有误计算逆解失败	- 重新示教位姿 - 如果当前机型为三轴或者四轴机器人,请检查输入位姿与当前机型特征是否相符 - 如果使用了 setAvoidSingularity(lock4axis), 请检查目标点是否符合锁轴要求
-50113	改变的姿态超过设定的阈值	- 请规避奇异点 - 重新设置奇异规避姿态改变的阈值 - 重新示教点位,更换目标点 - 使用其他方式的奇异规避
-50114	前瞻中轴运动超出运动范围, 提前停止运动	- 取消软限位 - 手动将机器人各轴移动到正常的工作范围内
-50115	轨迹前瞻过程中遇到奇异点	- 请规避奇异点 - 重新示教点位,更换目标点 - 更换 Jog 方式, 尝试轴空间 Jog - 将笛卡尔空间运动指令改为关节空间运动指令
-50118	改变姿态后求解的终点与所需要终点不一致	- 请规避奇异点 - 重新示教点位,更换目标点 - 更换奇异规避方式
-50120	奇异规避下搜索路径终点角度失败	- 请规避奇异点 - 重新示教点位,更换目标点 - 更换奇异规避方式 - 如果 Conf 开启, 可以关闭 Conf
-50121	奇异规避下搜索路径终点角度失败	- 请规避奇异点 - 重新示教点位,更换目标点 - 更换奇异规避方式
-50204	规划过程中采样点不足, 请重新运行	工控机状态不稳定, 重新运行或尝试重启
-50205	相邻位置指令相差过大	- 重新示教轨迹目标点位 - 尝试更改指令形式, 例如 MoveL 改为 MoveJ
-50208	内部轨迹错误	- 请调用 moveReset()重置运动指令缓存 - 增大或减小转弯区
-50401	检测到碰撞	- 请检查机器人运行环境, 确认人员、设备安全后, 重新上电, 再恢复运行 - 检查当前工具设置是否与实际一致, 设定的工具质量质心是否合理
-50501	工具工件坐标系设置失败, 可能由于工具或工件不存在	设置已创建的工具工件, 且分别为手持和外部
-50512	轴数不匹配	轴空间 Jog 的下标参数不能超出轴数+外部轴数
-50513	速度设置无效, 机器人无法运动	传入 0.01~1 范围内的速度参数
-50514	步长设置无效, 机器人无法运动	传入大于 0 的步长参数
-50515	参考坐标系设置无效, 机器人无法运动	传入支持的坐标系类型参数
-50516	运动轴设置无效, 机器人无法运动	按照说明传入 index 参数
-50518	运动失败, 目标点可能是当前点	目标点位可能是当前点
-50519	生成轨迹失败, 目标点位可能超出机器人工作范围	请在机器人正常工作范围内, 重新示教点位
-50525	该机型不支持奇异规避模式运动, 或机器人锁轴失败, 4 轴角度不为 0 不运行运动, 请调整 4 轴角度	调整四轴角度至 0 或者正负 180 度

-60005	RL 工程或指定任务不存在	运行已创建的工程和任务
-60014	控制器当前状态不允许开始运动	确保机器人上电并处于空闲中
-60200	负载信息错误, 设置失败	- 负载信息错误, 请检查负载重量是否超过额定负载, 质心不超过 0.3m - 力控不支持外部工具或手持工件
-60511	路径不存在	使用已保存的回放路径
-60611	正在生成诊断数据不能运动, 请等待	等待 10 秒后再次尝试启动
-60702	当前四轴角度不为 0, 不允许切换到四轴固定模式, 或机型不支持	请检查第四轴当前角度是否为 0° 或 180°
-60704	不满足牺牲姿态奇异规避的开启条件	请检查当前状态是否满足牺牲姿态奇异规避开启状态
-60706	不满足轴空间插补奇异规避的开启条件	请检查当前状态是否满足轴空间插补奇异规避开启状态
255	未初始化连接机器人	一般不会发生, 正常构造 Robot 类实例即可
256	网络连接错误	- 工控机未开机 - 机器人地址错误, 或未处于同一局域网 - 实例类型错误或 SDK 未授权, 连接被拒绝
257	无法解析机器人消息, 可能由于 SDK 版本与控制器版本不匹配	可能由于控制器和 SDK 版本不匹配
258	参数错误, 数值超出范围	按照函数说明检查参数数值范围
259	参数错误, 参数类型或个数错误	按照函数说明检查参数类型或数组长度
260	不是合法的变换矩阵	检查传入参数是否符合其次变换矩阵要求
261	数组元素个数与机器人轴数不符	传入和机器人轴数一致的数组长度
262	运动控制模式错误, 请切换到正确的模式	根据实际情况调用 setMotionControlMode 切换模式
263	超时前未收到机器人回复, 可能由于网络通信问题	检查网络连接状态, 或反馈技术支持人员
264	重复操作	碰撞检测已打开, 需要先关闭
265	通过 UDP 端口接收数据失败, 请检查网络及防火墙配置	- 检查本机地址设置是否正确 - 检查防火墙设置, 是否允许 UDP 连接
266	客户端校验失败, 请检查控制器版本, 授权状态和机器人型号	- 按照手册或 README 将控制器升级到匹配版本 - 创建类型匹配的机器人实例 - 联系技术支持人员授权 SDK 功能
272	事件未监听	先调用 setEventWatcher 开始监听数据
273	点位距离过近, 坐标系标定失败	参考《xCore 控制系统使用手册》工具工件标定方法, 重新标定
512	IP 地址或端口设置错误, 或端口被占用	- 本机 IP 地址不能为 localhost 或机器人地址 - 存在端口占用情况, 请检查 RCI 设置-端口
513	设置了不支持的字段, 或总长度超出限制	支持的字段见 RtSupportedFields, 总长度限制为 1024 字节
768	没有可执行的运动指令	先调用 moveAppend 下发运动指令, 再开始运动
769	机器人停止中或当前状态无法暂停	可能由于调用 stop()前刚刚手动模式下电, 或者按下急停, 控制器正在响应停止中, 请等待

4.8.2 异常

连接到机器人时会检查授权状态、机型是否匹配等, 如果检查不通过会抛出异常 (rokae.clr.ApiException)。

异常信息	原因及处理方法
xCore 最低版本要求	按照提示信息升级控制器版本
未授权 SDK 功能	联系技术支持人员授权 SDK 功能
机器人实例类型与连接的机型不符	按照手册说明创建对应的机器人实例

5 注意事项

5.1 与 RobotAssist 同时使用

目前如果使用 xCore SDK 控制机器人，并不会限制通过 RobotAssist 的控制。机器人的一些状态，通过 xCore SDK 更改后也会体现在 Robot Assist 界面上；一些工程运行，运动控制则是分离的。大致总结如下：

会同步更新的组件	● 底部状态栏：手自动，机器人状态，上下电； ● 状态监控窗口； ● 日志上报；
双方可控，即通过 RobotAssist 修改会生效的组件	● RL 工程运行速率和循环/单次模式； ● 运动的暂停； ● 工具工件组：更改 RobotAssist 右上角选择的工具工件会改变 toolset；通过 setToolset()设置了工具工件，右上角会显示“toolx”，“wobjx”
不会同步更新的组件，包括但不限于	● 下发的运动指令无法通过点击开始按钮让机器人开始执行； ● 加载的 RL 工程，以及前瞻指针、运动指针等，不会同步显示； ● 大部分机器人设置界面显示的功能打开状态和设定值等；

建议的控制方式是单一控制源，避免混淆。

6 使用示例

本章展示一些示例程序。

示例一：机器人基本操作，信息查询，Jog，拖动等

```
// 协作六轴机型
// 方式一：创建对象并连接到机器人
// xMateRobot robot = new xMateRobot("192.168.0.160");

// 方式二：default constructor，然后调用connectToRobot(remoteIP)
xMateRobot robot = new xMateRobot();

/// <summary>
/// 示例：其它SDK接口调用示例
/// </summary>
public void Example_OtherOperations()
{
    ErrorCode ec;

    // xCore-SDK版本
    robot.sdkVersion();

    // 查询上下电状态/操作模式
    Console.WriteLine("上电状态:" + robot.powerState(out ec).ToString() +
        " | 操作模式:" + robot.operateMode(out ec).ToString());

    // 读取基坐标系、计算正逆解
    Console.WriteLine("\n基坐标系 - [ ");
    Array.ForEach(robot.baseFrame(out ec), PrintHelper.printArray);
    Console.WriteLine("]\n当前轴角度 [ ");
    var jntpos = robot.jointPos(out ec);
    Array.ForEach(jntpos, PrintHelper.printArray);
    Console.WriteLine("]\n计算正解 [ ");
    CartesianPosition fkPos = robot.calcFk(jntpos, out ec);
    Array.ForEach(fkPos.trans, PrintHelper.printArray);
    Array.ForEach(fkPos.rpy, PrintHelper.printArray);

    Console.WriteLine("]\n计算逆解 [ ");
    Array.ForEach(robot.calcIk(fkPos, out ec), PrintHelper.printArray);
    Console.WriteLine("]");

    // 读取末端力矩信息
    double[] joint_torque = new double[6], external_torque = new double[6];
    double[] cart_torque = new double[3], cart_force = new double[3];
    robot.getEndTorque(FrameType.flange, ref joint_torque, ref external_torque,
        ref cart_torque, ref cart_force, out ec);
    Console.WriteLine("轴力矩 [ ");
    Array.ForEach(joint_torque, PrintHelper.printArray);
    Console.WriteLine("]\n外力矩 [ ");
    Array.ForEach(external_torque, PrintHelper.printArray);
    Console.WriteLine("]\n笛卡尔空间力矩 [ ");
    Array.ForEach(cart_torque, PrintHelper.printArray);
    Console.WriteLine("]\n笛卡尔空间力 [ ");
    Array.ForEach(cart_force, PrintHelper.printArray);
    Console.WriteLine("]");

    // 查询控制器日志
```

```
// 查询最近5条"error"级别日志
List<LogInfo> logs = robot.queryControllerLog(5, LogInfo.Level.error, out
ec);
foreach (var log in logs)
{
    Console.WriteLine("Log ID" + log.id + " Content " + log.content + "\n");
}
}
```

示例二：运动指令

```
/// <summary>
/// 示例：使用moveAppend() + moveStart()运动指令；
/// 以及设置轴配置数据的示例
/// </summary>
public void Example_Move2()
{
    ErrorCode ec;
    Move_Preset(100, 10);

    string cmdid = "";
    MoveCommand p1 = new MoveCommand(), p2 = new MoveCommand(), p3 = new
MoveCommand(), p4 = new MoveCommand();
    p1.cartTarget.trans = new double[] { 0.2434, -0.314, 0.591 };
    p1.cartTarget.rpy = new double[] { 1.5456, 0.314, 2.173 };
    p2.cartTarget.trans = p3.cartTarget.trans = p1.cartTarget.trans;
    p2.cartTarget.rpy = p3.cartTarget.rpy = p1.cartTarget.rpy;

    // 3个目标点，末端姿态相同，轴角度不同
    p1.cartTarget.confData = new List<int> { -67, 100, 88, -79, 90, -120 };
    p2.cartTarget.confData = new List<int> { -76, 8, -133, -106, 103, 108 };
    p3.cartTarget.confData = new List<int> { -70, 8, -88, 90, -105, -25 };

    // 设置一下速度
    p1.speed = 200;
    p2.speed = 1000;
    p3.speed = 400;

    var cmds = new List<MoveCommand> { p1, p2, p3 };
    robot.moveAppend(MoveCommand.Type.MoveJ, cmds, ref cmdid, out ec);
    robot.moveStart(out ec);
    waitForFinish(cmdid, cmds.Count - 1);
}
}
```

7 反馈与勘误

文档中若出现不准确的描述或者错误，恳请读者指正批评。如果您在阅读过程中发现任何问题或者有想提出的意见，可以发送邮件到 xuqiu@rokae.com, 我们的同事会尽量一一回复。

8 附录 A – C# API

8.1 枚举类型

8.1.1 机器人工作状态 rokae.clr.OperationState 机器人工作状态

idle	机器人静止
jog	Jog 状态(未运动)
rtControlling	实时模式控制中
drag	拖动已开启
rlProgram	RL 工程运行中
demo	Demo 演示中
dynamicIdentify	动力学辨识中
frictionIdentify	摩擦力辨识中
loadIdentify	负载辨识中
moving	机器人运动中
jogging	Jog 运动中
unknown	未知

8.1.2 机器人操作模式 rokae.clr.OperateMode

manual	手动
automatic	自动
unknown	未知(发生异常)

8.1.3 机器人上下电及急停状态 rokae.clr.PowerState

on	上电
off	下电
estop	急停被按下
gstop	安全门打开
unknown	未知(发生异常)

8.1.4 位姿坐标系类型 rokae.clr.CoordinateType

flangeInBase	法兰相对于基坐标系
endInRef	末端相对于外部坐标系

8.1.5 运动控制模式 rokae.clr.MotionControlMode

Idle	空闲
NrtCommand	非实时模式执行运动指令
NrtRLTask	非实时模式运行 RL 工程
RtCommand	实时模式控制

8.1.6 机器人停止运动等级 rokae.clr.StopLevel

stop0	快速停止机器人运动后断电
stop1	规划停止机器人运动后断电, 停在原始路径上
stop2	规划停止机器人运动后不断电, 停在原始路径上
suppleStop	柔顺停止, 仅适用于协作机型

8.1.7 机器人拖动模式参数 rokae.clr.DragOpt

Space:jointSpace	轴空间拖动
------------------	-------

Space::cartesianSpace	笛卡尔空间拖动
Type::translationOnly	仅平移
Type::rotationOnly	仅旋转
Type::freely	自由拖拽

8.1.8 坐标系类型 rokae.clr.FrameType

world	世界坐标系
baseFrame	基坐标系
flange	法兰坐标系
tool	工具坐标系
wobj	工件坐标系
path	路径坐标系, 力控任务坐标系需要跟踪轨迹变化的过程
rail	导轨基坐标系

8.1.9 Jog 选项 - 坐标系 rokae.clr.JogSpace

world	世界坐标系
flange	法兰坐标系
baseFrame	基坐标系
toolFrame	工具坐标系
wobjFrame	工件坐标系
jointSpace	轴空间
singularityAvoidMode	奇异规避模式, 适用于工业六轴, xMateCR 和 xMateSR 六轴机型
baseParallelMode	平行基座模式, 适用于工业六轴, xMateCR 和 xMateSR 六轴机型

8.1.10 奇异规避方式 rokae.clr.AvoidSingularityMethod

lockAxis4	四轴锁定
wrist	牺牲姿态
jointWay	轴空间短轨迹插补

8.1.11 运动指令笛卡尔点位偏移类型 rokae.clr.CartesianPosition.Offset.Type

none	无偏移
offs	相对工件坐标系偏移
relTool	相对工具坐标系偏移

8.1.12 运动指令类型 rokae.clr.MoveCommand.Type

MoveL	末端直线轨迹, 目标点是 cartTarget
MoveJ	轴运动, 目标点是 cartTarget
MoveAbsJ	轴运动, 目标点是 jointTarget
MoveC	圆弧轨迹 MoveC, 目标点是 cartTarget, 经过辅助点 auxPoint
MoveCF	全圆轨迹 MoveCF, 目标点是 cartTarget, 经过辅助点 auxPoint
MoveSP	螺旋线轨迹 MoveSP, 目标点是 cartTarget
MoveWait	运动停留指令

8.1.13 xPanel 配置: 对外供电模式 rokae.clr.xPanelOpt.Vout

off	不输出
reserve	保留

supply12v	输出 12V
supply24v	输出 24V

8.1.14 事件类型 rokae.clr.Event

moveExecution	非实时运动指令执行信息
safety	安全 (是否碰撞)

8.2 数据结构

本节列出数据结构的公有属性。

8.2.1 机器人基本信息 rokae.clr.Info

string id	机器人 uid, 可用于区分连接的机器人
string version	控制器版本
string type	机器人机型名称
int joint_num	轴数

8.2.2 坐标系 rokae.clr.Frame

double[] trans	平移量, [X, Y, Z], 单位:米
double[] rpy	XYZ 欧拉角, [Rx, Ry, Rz], 单位:弧度

8.2.3 笛卡尔点位 rokae.clr.CartesianPosition

double[] trans	平移量, [X, Y, Z], 单位:米
double[] rpy	旋转量, [Rx, Ry, Rz], 单位:弧度
double elbow	臂角, 适用于 7 轴机器人, 单位: 弧度
List<int> confData	轴配置数据, 元素个数应和机器人轴数一致
List<double> external	外部关节数值 单位:弧度 米。导轨单位米

8.2.4 运动指令笛卡尔点位偏移量 rokae.clr.CartesianPosition.Offset

Type type	类型: offs/relTool
Frame frame	偏移坐标

8.2.5 关节点位 rokae.clr.JointPosition

List< double > joints	关节角度值, 单位:弧度
List< double > external	外部关节数值 单位:弧度 米。导轨单位米

8.2.6 运动指令 rokae.clr.MoveCommand

CartesianPosition cartTarget	目标笛卡尔点位
JointPosition jointTarget	目标关节点位
CartesianPosition auxPoint;	辅助点
CartesianPosition::Offset cartTargetOffset	笛卡尔目标点偏移选项
CartesianPosition::Offset auxPointOffset	辅助点偏移选项
int speed	末端线速度, 单位 mm/s, 关节速度根据末端线速度大小划分几个区间, 详见 setDefaultSpeed()
double jointSpeed	节速度百分比, 用于关节类型指令, 范围[0, 1]。大于等于 0 时生效; 小于 0 时仍使用 speed 计算出的关节速度
double rotSpeed	空间旋转速度, 用于末端类型指令, 单位 rad/s。大于等于 0 时生效; 小于 0 时旋转速度默认为 200°/s
int zone	转弯区大小, 单位 mm
List<double> args;	运动指令其它参数

	- Type.MoveCF: 接受长度为 2 参数, 依次为: 全圆执行角度[弧度]; 旋转姿态类型, 0 不变姿态 1 动轴旋转 2 定轴旋转 - Type.MoveSP: 接受长度为 4 参数, 依次为: 初始半径[米]; 每旋转单位角度半径的变化[米/弧度]; 合计旋转角度[弧度]; 旋转方向, 1 顺时针 0 逆时针 - Type.MoveWait: 接受长度为 1 参数, 含义为: 停留时长[秒], 最小有效时长 1ms
String customInfo	自定义信息, 可在运动信息反馈中返回出来

8.2.7 负载信息 rokae.clr.Load

double mass	负载质量, 单位: 千克
double[] cog	质心 [x, y, z], 单位: 米
double[] inertia	惯量 [ix, iy, iz], 单位: 千克·平方米

8.2.8 工具工件组信息 rokae.clr.Toolset

根据一对工具工件的坐标、负载、机器人手持设置计算得出。

Load load	机器人末端手持负载
Frame end	机器人末端坐标系相对法兰坐标系转换
Frame reference	机器人参考坐标系相对世界坐标系转换

8.2.9 坐标系标定结果 rokae.clr.FrameCalibrationResult

Frame frame	标定结果
double[] errors	样本点与 TCP 标定值的偏差, 依次为最小值, 平均值, 最大值, 单位 m

8.2.10 RL 工程信息 rokae.clr.RLProjectInfo

string name	工程名称
List< string > taskList	任务名称列表

8.2.11 工具/工件信息 rokae.clr.WorkToolInfo

string name	名称
string alias	别名, 暂未使用
bool robotHeld	是否机器人手持
Frame pos	位姿。工件的坐标系已相对其用户坐标系变换
Load load	负载

8.2.12 控制器日志信息 rokae.clr.LogInfo

const int id	日志 ID 号
string timestamp	日期及时间
string content	日志内容
string repair	修复办法

8.2.13 错误码 rokae.clr.ErrorCode

string message	错误信息
int value	错误码数值

8.3 方法

8.3.1 机器人基本操作及信息查询

baseFrame()
double[] rokae.clr.BaseRobot.baseFrame (out ErrorCode ec) 读取基坐标系, 相对于世界坐标系
参数 ec 错误码
返回 [X, Y, Z, Rx, Ry, Rz], 其中平移量单位为米, 旋转量单位为弧度

calcFk() [1/2]
CartesianPosition rokae.clr.BaseRobot.calcFk (double[] joints, out ErrorCode ec) 根据轴角度计算正解
参数 joints 轴角度, 单位: 弧度 ec 错误码
返回 机器人末端位姿, 相对于外部参考坐标系

calcFk() [2/2]
CartesianPosition rokae.clr.BaseRobot.calcFk (double[] joints, Toolset toolset, out ErrorCode ec) 根据轴角度计算正解
参数 joints 轴角度, 单位: 弧度 toolset 工具工件组信息 ec 错误码
返回 机器人末端位姿, 相对于外部参考坐标系

calcIk() [1/2]
double[] rokae.clr.BaseRobot.calcIk (CartesianPosition posture, out ErrorCode ec) 根据位姿计算逆解
参数 posture 机器人末端位姿, 相对于外部参考坐标系 ec 错误码
返回 轴角度, 单位: 弧度

calcIk() [2/2]
double[] rokae.clr.BaseRobot.calcIk (CartesianPosition posture, Toolset toolset, out ErrorCode ec) 根据位姿计算逆解
参数 posture 机器人末端位姿, 相对于外部参考坐标系 toolset 工具工件组信息 ec 错误码

返回

轴角度, 单位:弧度

calibrateFrame()

FrameCalibrationResult rokae.clr.BaseRobot.calibrateFrame (FrameType type, List<double[]> points, bool is_held, double[] base_aux, out ErrorCode ec)

坐标系标定 (N 点标定)。

各坐标系类型支持的标定方法及注意事项:

1. 工具坐标系: 三点/四点/六点标定法
2. 工件坐标系: 三点标定。标定结果不会相对用户坐标系做变换, 即, 若为外部工件, 返回的结果是相对于基坐标系的。
3. 基坐标系: 六点标定。标定前请确保动力学约束和前馈已关闭。
4. 导轨坐标系: 三点标定。若标定成功(无错误码), 控制器会自动保存标定结果, 重启控制器后生效。

若标定成功(无错误码), 控制器会自动保存标定结果, 重启控制器后生效。

参数

type	坐标系类型, 支持工具 (FrameType::tool), 工件 (FrameType::wobj), 基坐标系 (FrameType::base)
points	轴角度列表, 列表长度为 N。例如, 使用三点法标定工具坐标系, 应传入 3 组轴角度。轴角度的单位是弧度。
is_held	true - 机器人手持 false - 外部。仅影响工具/工件的标定
base_aux	基坐标系标定时用到的辅助点, 单位[米]
ec	错误码

connectToRobot() [1/3]

void rokae.clr.BaseRobot.connectToRobot (out ErrorCode ec)

建立与机器人的连接。机器人地址为创建 robot 实例时传入的

参数

ec	错误码
----	-----

connectToRobot() [2/3]

void rokae.clr.BaseRobot.connectToRobot (string remoteIP)

建立与机器人的连接

参数

remoteIP	机器人 IP 地址
----------	-----------

异常

rokae.clr.ApiException 网络连接异常;控制器版本不兼容;机型不匹配等

connectToRobot() [3/3]

void rokae.clr.BaseRobot.connectToRobot (string remoteIP, string localIP)

建立与机器人的连接

参数

remoteIP	机器人 IP 地址
localIP	本机地址

异常

rokae.clr.ApiException 网络连接异常;控制器版本不兼容;机型不匹配等

cartPosture ()

CartesianPosition rokae.clr.BaseRobot.cartPosture(CoordinateType ct, out ErrorCode ec) 获取机器人法兰或末端的当前位姿
参数 ct 坐标系类型 ec 错误码 返回 获取机器人法兰或末端的当前位姿
clearServoAlarm() void rokae.clr.BaseRobot.clearServoAlarm(out ErrorCode ec) 清除伺服报警
参数 ec 错误码, 当有伺服报警且清除失败的情况下错误码置为-1
disableCollisionDetection () void rokae.clr.Cobot.disableCollisionDetection (out ErrorCode ec) 关闭碰撞检测功能
参数 ec 错误码
disconnectFromRobot() void rokae.clr.BaseRobot.disconnectFromRobot (out ErrorCode ec) 断开与机器人连接。断开前会停止机器人运动, 请注意安全
参数 ec 错误码
enableCollisionDetection() void rokae.clr.Cobot.enableCollisionDetection (double[] sensitivity, StopLevel behaviour, double fallback_compliance, out ErrorCode ec) 设置碰撞检测相关参数, 打开碰撞检测功能
参数 sensitivity 碰撞检测灵敏度, 范围 0.01-2.0 behaviour 碰撞后机器人行为, 支持 stop1(安全停止, stop0 和 stop1 处理方式相同)和 stop2(触发暂停), suppleStop(柔顺停止) fallback_compliance 1) 碰撞后行为是安全停止或触发暂停时, 该参数含义是碰撞后回退距离, 单位: 米 2) 碰撞后行为是柔顺停止时, 该参数含义是柔顺度, 范围 [0.0, 1.0] ec 错误码
getSoftLimit() bool rokae.clr.BaseRobot.getSoftLimit (ref double[,] limits, out ErrorCode ec) 获取当前软限位数值
参数 limits 各轴软限位 [下限, 上限], 单位: 弧度 ec 错误码 返回 true - 已打开 false - 已关闭

jointPos()
double[] rokae.clr.BaseRobot.jointPos (out ErrorCode ec) 机器人当前轴角度, 机器人本体+外部轴, 单位: 弧度, 外部轴导轨单位米
参数
ec 错误码
返回
关节弧度

jointTorque()
double[] rokae.clr.BaseRobot.jointTorque (out ErrorCode ec) 关节力传感器数值, 单位: Nm
参数
ec 错误码
返回
力矩值

jointVel()
double[] rokae.clr.BaseRobot.jointVel (out ErrorCode ec) 机器人当前关节速度, 机器人本体+外部轴, 单位: 弧度/秒, 外部轴单位米/秒
参数
ec 错误码
返回
关节速度

getStateDate() [1/3]
int rokae.clr.BaseRobot.getStateData(string name, ref double data) 读取机器人状态数据, 注意传入的 data 类型要和数据类型一致。 支持的数据名称 – • 臂角[rad] – "psi_m"
参数
name 数据名 data 数值
返回
无该数据名; 或未通过 startReceiveRobotState() 设置为要接收的数据; 或该数据类型和 R 不符, 返回-1。成功读取返回 0。
异常
rokae.clr.ApiException: 网络错误

getStateDate() [2/3]
int rokae.clr.BaseRobot.getStateData(string name, ref ulong data) 读取机器人状态数据, 注意传入的 data 类型要和数据类型一致。 支持的数据名称 – • 时间戳, 精确到微秒 – "ts"
参数
name 数据名 data 数值
返回

无该数据名；或未通过 `startReceiveRobotState()` 设置为要接收的数据；或该数据类型和 R 不符，返回-1。成功读取返回 0。

异常

`rokae.clr.ApiException`: 网络错误

getStateData() [3/3]

`int rokae.clr.BaseRobot.getStateData(string name, ref double[] data)`

读取机器人状态数据,注意传入的 data 类型要和数据类型一致。

支持的数据名称及数组长度 –

- 关节角度[rad] –"q_m" | DoF
- 关节速度[rad/s] –"dq_m" | DoF
- 关节力矩[Nm] –"tau_m" | DoF
- 电机位置 –"theta_m" | DoF
- 电机位置微分 –"theta_vel_m" | DoF
- 电机转矩 –"motor_tau" | DoF
- 末端位姿, 相对于外部坐标系 [X,Y,Z,Rx,Ry,Rz] –"pos_abc_m" | 6
- 末端位姿, 相对于外部坐标系, 行优先齐次变换矩阵 –"pos_m" | 16
- 外部轴角度[rad] 导轨[m] –"ex_q_m" | 6
- 外部轴速度[rad/s] 导轨[m/s] –"ex_dq_m" | 6
- 外部轴电机位置 –"ex_motor_m" | 6

参数

name 数据名
data 数值

返回

无该数据名；或未通过 `startReceiveRobotState()` 设置为要接收的数据；或该数据类型和 R 不符，返回-1。成功读取返回 0。

异常

`rokae.clr.ApiException`: 网络错误

startReceiveRobotState()

`void rokae.clr.BaseRobot.startReceiveRobotState(int interval, string[] fields)`

让机器人控制器开始发送实时状态数据。阻塞等待收到第一帧消息，超时时间为 3 秒

参数

interval 控制器发送状态数据的间隔，单位 ms，允许的时长：1ms/2ms/4ms/8ms/1s
fields 接收的机器人状态数据, 最大总长度为 1024 个字节。支持的数据及名称见 `getStateData()`

异常

`rokae.clr.ApiException`: 设置了不支持的状态数据；或机器人无法开始发送数据；或总长度超过 1024；或已经开始发送数据；或超时后仍未收到第一帧数据
`System.NullReferenceException`: Robot 类未构造

stopReceiveRobotState ()

`void rokae.clr.BaseRobot.stopReceiveRobotState()`

停止接收实时状态数据，同时控制器停止发送。可用于重新设置要接收的状态数据

updateRobotState()

`uint rokae.clr.BaseRobot.updateRobotState(int timeout_ms)`

停止接收实时状态数据，同时控制器停止发送。可用于重新设置要接收的状态数据

参数

timeout_ms	超时时间[ms]
返回	
接收到的数据长度。如果超时前没有收到数据，那么返回 0。	
异常	
无法收到数据；或收到的数据有错误导致无法解析	

operateMode()	
OperateMode rokae.clr.BaseRobot.operateMode (out ErrorCode ec)	
查询机器人当前操作模式	
参数	
ec	错误码
返回	
手动 自动	

operationState()	
OperationState rokae.clr.BaseRobot.operationState (out ErrorCode ec)	
查询机器人当前运行状态 (空闲,运动中, 拖动开启等)	
参数	
ec	错误码
返回	
运行状态枚举类	

posture()	
double[] rokae.clr.BaseRobot.posture (CoordinateType ct, out ErrorCode ec)	
获取机器人法兰或末端的当前位姿	
参数	
ct	坐标系类型 1. flangeInBase: 法兰相对于基坐标系; 2. endInRef: 末端相对于外部参考坐标系。例如,当设置了手持工具及外部工件后, 该坐标系类型返回的是工具相对于工件坐标系的坐标。
ec	错误码
返回	
double 数组, [X, Y, Z, Rx, Ry, Rz], 其中平移量单位为米旋转量单位为弧度	

powerState()	
PowerState rokae.clr.BaseRobot.powerState (out ErrorCode ec)	
查询机器人上下电以及急停状态	
参数	
ec	错误码
返回	
on-上电 off-下电 estop-急停 gstop-安全门打开	

queryControllerLog()	
List<LogInfo> rokae.clr.BaseRobot.queryControllerLog (uint count, LogInfo.Level level, out ErrorCode ec)	
查询控制器最新的日志	
参数	

count	查询个数, 上限是 10 条
level	指定日志等级
ec	错误码
返回	
日志信息	

robotInfo()	
Info rokae.clr.BaseRobot.robotInfo (out ErrorCode ec)	
查询机器人基本信息	
参数	
ec	错误码
返回	
机器人基本信息: 控制器版本, 机型, 轴数	

sdkVersion()	
string rokae.clr.BaseRobot.sdkVersion ()	
查询 xCore-SDK 版本	
返回	
版本号	

setBaseFrame()	
void rokae.clr.BaseRobot.setToolset (Frame frame, out ErrorCode ec)	
设置基坐标系, 设置后仅保存数值, 重启控制器后生效	
参数	
frame	坐标系, 默认使用自定义安装方式
ec	错误码

setOperateMode()	
void rokae.clr.BaseRobot.setOperateMode (OperateMode mode, out ErrorCode ec)	
切换手动模式	
参数	
mode	手动/自动
ec	错误码

setPowerState()	
void rokae.clr.BaseRobot.setPowerState (bool on, out ErrorCode ec)	
机器人上下电。注: 只有无外接使能开关或示教器的机器人才能手动模式上电。	
参数	
on	true-上电 false-下电
ec	错误码

setSoftLimit()	
void rokae.clr.BaseRobot.setSoftLimit (bool enable, double[,] limits, out ErrorCode ec)	
设置软限位。软限位设定要求: 1) 打开软限位时, 机械臂应下电且处于手动模式; 2) 软限位不能超过机械硬限位 3) 机械臂当前各轴角度应在设定的限位范围内	
参数	

enable	true - 打开 false - 关闭 各轴[下限, 上限], 单位: 弧度。
limits	- 当 limits 为默认值时, 视为仅打开软限位不修改数值; 不为默认值时, 先修改软限位再打开 - 关闭软限位时不会修改限位数值
ec	错误码

setToolset()

void rokae.clr.BaseRobot.setToolset (string toolName, string wobjName, out ErrorCode ec)

设置前提:

1) 使用 RL 工程中创建的工具工件: 需要先加载对应 RL 工程;

2) 全局工具工件: 无需加载工程, 直接调用即可。e.g. setToolset("g_tool_0", "g_wobj_0")

一组工具工件无法同时为手持或外部; 如果有冲突, 以工具的位置为准, 例如工具工件同时为手持, 不会返回错误, 但是工件的坐标系变成了外部

参数

toolName 工具名称

wobjName 工件名称

ec 错误码

setToolset()

void rokae.clr.BaseRobot.setToolset (Toolset toolset, out ErrorCode ec)

设置工具工件组信息。此工具工件组仅为 SDK 运动控制使用, 不与 RL 工程相关。设置后 RobotAssist 右上角会显示 "toolx", "wobjx", 状态监控显示的末端坐标也会变化。除此接口外, 如果通过 RobotAssist 更改默认工具工件(右上角的选项), 该工具工件组也会相应更改。

参数

toolset 工具工件组信息

ec 错误码

toolset()

Toolset rokae.clr.BaseRobot.toolset (out ErrorCode ec)

查询当前工具工件组信息。此工具工件组仅为 SDK 运动控制使用, 不与 RL 工程相关

参数

ec 错误码

返回

工具工件组

recoverState()

void rokae.clr.BaseRobot.recoverState (int item, out ErrorCode ec)

根据选项恢复机器人状态

参数

item 恢复选项, 1: 急停恢复

ec 错误码

setRailParameter() [1/6]

void rokae.clr.setRailParameter(String name, rokae.clr.Frame value, out ErrorCode ec)

设置导轨基坐标系

参数

name 参数名, "baseFrame"

value	基坐标系
ec	错误码

setRailParameter() [2/6]	
void rokae.clr.setRailParameter(String name, String value, out ErrorCode ec) 设置导轨名称	
参数	
name	参数名, "name"
value	导轨名称
ec	错误码

setRailParameter() [3/6]	
void rokae.clr.setRailParameter(String name, double value, out ErrorCode ec) 设置导轨参数	
参数	
name	参数名, 见 value 说明
value	减速比 - reductionRatio
	最大速度(m/s) - maxSpeed
	最大加速度 (m/s^2) - maxAcc
	最大加加速度(m/s^3) - maxJerk
ec	错误码

setRailParameter() [4/6]	
void rokae.clr.setRailParameter(String name, bool value, out ErrorCode ec) 打开关闭导轨	
参数	
name	参数名, enable
value	true – 打开
ec	错误码

setRailParameter() [5/6]	
void rokae.clr.setRailParameter(String name, double[] value, out ErrorCode ec) 设置导轨参数	
参数	
name	参数名, 见 value 说明
value	软限位(m), [下限,上限] - softLimit
	运动范围(m), [下限,上限] - range
ec	错误码

setRailParameter() [6/6]	
void rokae.clr.setRailParameter(String name, int value, out ErrorCode ec) 设置导轨参数	
参数	
name	参数名, 见 value 说明
value	编码器分辨率 - encoderResolution
	电机最大转速(rpm) - motorSpeed
ec	错误码

getRailParameter()	
void rokae.clr.getRailParameter(string name, ref T value, out ErrorCode ec)	
读取导轨参数	
类型参数	T 读取的导轨参数类型
参数	
name	参数名, 见 setRailParameter()
value	参数数值, 见 setRailParameter()
ec	错误码 参数名不存在或数据类型不匹配返回错误码

configNtp()	
void rokae.clr.configNtp(string server_ip, out ErrorCode ec)	
配置 NTP。非标配功能, 需要额外安装	
参数	
server_ip	NTP 服务端 IP
ec	错误码, NTP 未正确安装, 或 IP 地址格式错误

syncTimeWithServer()	
void rokae.clr.syncTimeWithServer(out ErrorCode ec)	
手动同步一次时间, 远端 IP 是通过 configNtp 配置的。耗时几秒钟, 阻塞等待同步完成, 接口预设的超时时间是 12 秒	
参数	
ec	错误码。NTP 服务未正确安装, 或无法和服务端同步

8.3.2 运动控制

adjustAcceleration()	
void rokae.clr.BaseRobot.adjustAcceleration(double acc, double jerk, out ErrorCode ec)	
调节运动加/减速度和加加速度。如果在机器人运动中调用, 当前正在执行的指令不生效, 下一条指令生效	
参数	
acc	系统预设加速度的百分比, 范围[0.2, 1.5], 超出范围不会报错, 自动改为上限或下限值
jerk	系统预设的加加速度的百分比, 范围[0.1, 2], 超出范围不会报错, 自动改为上限或下限值
ec	错误码

adjustSpeedOnline()	
void rokae.clr.BaseRobot.adjustSpeedOnline (double scale, out ErrorCode ec)	
动态调整机器人运动速率, 效果和 RobotAssist 界面下方调整程序速率滑条一样。	
参数	
scale	运动指令的速度的比例, 范围 0.01 - 1。当设置 scale 为 1 时, 机器人将以路径原本速度运动
ec	错误码

executeCommand()	
void rokae.clr.BaseRobot.executeCommand (MoveCommand.Type type, List<MoveCommand> cmd, out ErrorCode ec)	
执行单条或多条运动指令, 调用后机器人立刻开始运动	
参数	
type	指令类型
cmd	指令列表, 允许的个数为 1-100, 需为同类型指令

ec	错误码, 仅反馈执行前的错误, 包括:
	1. 网络连接问题;
	2. 指令个数或类型不符

getAcceleration()

void rokae.clr.BaseRobot.getAcceleration(ref double acc, ref double jerk, out ErrorCode ec)

读取当前加/减速度和加加速度

参数

acc	系统预设加速度的百分比
jerk	系统预设的加加速度的百分比
ec	错误码

getAvoidSingularity()

bool rokae.clr.xMateRobot.getAvoidSingularity (AvoidSingularityMethod method, out ErrorCode ec)

查询是否处于规避奇异点的状态

参数

method	奇异规避方式
ec	错误码

返回

true - 已打开 | false - 已关闭

moveAppend() [1/2]

void rokae.clr.BaseRobot.moveAppend (MoveCommand.Type type, List<MoveCommand> cmds, string cmdID, out ErrorCode ec)

添加单条或多条运动指令, 添加后调用 moveStart()开始运动

参数

type	指令类型
cmds	指令列表, 允许的个数为 1-100, 须为同类型的指令
cmdID	本条指令的 ID, 可用于查询指令执行信息
ec	错误码, 仅反馈指令发送前的错误, 包括: 1) 网络连接问题; 2) 指令个数不符;

moveAppend() [2/2]

void rokae.clr.BaseRobot.moveAppend (MoveCommand.Type type, MoveCommand cmd, string cmdID, out ErrorCode ec)

添加单条, 添加后调用 moveStart()开始运动

参数

type	指令类型
cmds	指令
cmdID	本条指令的 ID, 可用于查询指令执行信息。MoveWait 指令没有信息反馈
ec	错误码, 仅反馈指令发送前的错误, 包括: 1) 网络连接问题; 2) 指令个数不符;

moveReset()

void rokae.clr.BaseRobot.moveReset (out ErrorCode ec)

重置运动缓存, 清空已发送的运动指令, 清除执行信息。

次程序开始运行并第一次执行运动指令之前, 调用该函数来重置运动缓存

参数

ec	错误码
----	-----

moveStart()

void rokae.clr.BaseRobot.moveStart (out ErrorCode ec)

开始/继续运动

参数

ec 错误码

queryEventInfo()

Dictionary<string, object> rokae.clr.BaseRobot.queryEventInfo (Event eventType, out ErrorCode ec)

查询事件信息

参数

eventType 事件类型

ec 错误码

返回

事件信息

setAvoidSingularity()

void rokae.clr.xMateRobot.setAvoidSingularity(AvoidSingularityMethod method, bool enable, double limit, out ErrorCode ec)

打开/关闭奇异点规避功能。只适用于部分机型:

- 1) 四轴锁定: 支持工业六轴, xMateCR 和 xMateSR 六轴机型;
- 2) 牺牲姿态: 支持所有六轴机型;
- 3) 轴空间插补: 支持工业六轴机型

参数

method 奇异规避方式

enable true - 打开功能 | false - 关闭。对于四轴锁定方式, 打开之前要确保 4 轴处于零位。

limit 不同的规避方式, 该参数含义分别为:
 1) 牺牲姿态: 允许的姿态误差, 范围 (0, $\pi \times 2$], 单位弧度
 2) 轴空间插补: 规避半径, 范围 [0.005, 10], 单位米
 3) 四轴锁定: 无参数

ec 错误码

setDefaultConfOpt()

void setDefaultConfOpt (bool forced, out ErrorCode ec)

设置是否严格遵循笛卡尔点位 Conf 检查。初始值为 false (不严格遵循)。

对于 xMateCR 和 xMateSR 系列机型, 由于构型特殊, 当运动指令设置了 confData 时, 默认使用 confData 进行逆解计算。

参数

forced true - 严格遵循, 此时会用运动指令的 confData 计算笛卡尔点位逆解, 如计算失败则返回错误;
 false - 不严格遵循, 逆解时会选取机械臂当前轴角度的最近解

ec 错误码

setDefaultSpeed()

void rokae.clr.BaseRobot.setDefaultSpeed (int speed, out ErrorCode ec)

设定默认运动速度, 初始值为 100。

该数值表示末端最大线速度(单位 mm/s), 自动计算对应末端旋转速度及轴速度

参数

speed	该接口不对参数进行范围限制。末端线速度的实际有效范围分别是 5-4000(协作), 5-7000(工业)。 关节速度百分比划分为 5 个的范围: < 100 10%; 100 ~ 200 30%; 200 ~ 500 50%; 500 ~ 800 80%; > 800 100%
ec	错误码

setDefaultZone()	
void rokae.clr.BaseRobot.setDefaultZone (int zone, out ErrorCode ec) 设定默认转弯区。 该数值表示运动最大转弯区半径(单位:mm), 自动计算转弯百分比. 若不设置, 则为 0 (fine, 无转弯区)	
参数	
zone	该接口不对参数进行范围限制。转弯区半径大小实际有效范围是 0-200。 转弯百分比划分 4 个范围: < 1 : 0 (fine) ; 1 ~ 20 : 10% ; 20 ~ 60 : 30% ; > 60 : 100%
ec	错误码

setEventWatcher()	
void rokae.clr.BaseRobot.setEventWatcher (Event eventType, EventCallbackDelegate callback, out ErrorCode ec) 设置接收事件的回调函数	
参数	
eventType	事件类型
callback	处理事件的回调函数 - Event::moveExecution, 回调函数在同一个线程执行, 请避免函数中有执行时间较长的操作; - Event::safety 则每次独立线程回调, 没有执行时间的限制
ec	错误码

setMaxCacheSize()	
void rokae.clr.BaseRobot.setMaxCacheSize (int number, out ErrorCode ec) 设置最大缓存指令个数, 指发送到控制器待规划的路径点个数, 允许的范围[1,300], 初始值为 30。 如果轨迹多为短轨迹, 可以调大这个数值, 避免因指令发送不及时导致机器人停止运动(停止后如果有未执行的指令, 可 moveStart()继续);	
参数	
number	个数
ec	错误码

setMotionControlMode()	
void rokae.clr.BaseRobot.setMotionControlMode (MotionControlMode mode, out ErrorCode ec) 设置运动控制模式。 在调用各运动控制接口之前, 必须设置对应的控制模式。	
参数	
mode	模式。C#不支持 RtCommand
ec	错误码

startJog()	
void rokae.clr.BaseRobot.startJog (JogSpace space, double rate, double step, uint index, bool direction, out ErrorCode ec) 开始 jog 机器人, 需要切换到手动操作模式。	

调用此接口并且机器人开始运动后，无论机器人是否已经自行停止，都必须调用 stop()来结束 jog 操作，否则机器人会一直处于 jog 的运行状态。

参数

	jog 参考坐标系。
space	- 工具/工件坐标系使用原则同 setToolset() - 工业六轴机型和 xMateCR 和 xMateSR 六轴机型支持两种奇异规避方式 Jog: Space.singularityAvoidMode, Space.baseParallelMode - CR5 轴机型支持平行基座模式 Jog: Space::baseParallelMode
rate	速率, 范围 0.01 - 1
step	步长。单位: 笛卡尔空间-毫米 轴空间-度。步长大于 0 即可, 不设置上限, 如果机器人无法继续 jog 会自行停止运动。
index	根据不同的 space, 该参数含义如下: 1) 世界坐标系,基坐标系,法兰坐标系,工具工件坐标系: 0~5 分别对应 X, Y, Z, Rx, Ry, Rz a) 6 轴机型: 0~5 分别对应 X, Y, Z, Rx, Ry, Rz。 >5 代表外部轴(若有) b) 7 轴机型 6 代表肘关节, >6 代表外部轴(若有) 2) 轴空间: 关节序号, 从 0 开始计数 3) 奇异规避模式,平行基座模式: a) 6 轴机型: 0~5 分别对应 X, Y, Z, J4(4 轴), Ry, J6(6 轴); b) 5 轴机型: 0~4 分别对应 X, Y, Z, Ry, J5(5 轴) 根据不同的 space 和 index, 该参数含义如下: 1) 奇异规避模式 J4: true - $\pm 180^\circ$ false - 0°; 2) 平行基座模式 J4 & Ry: true - $\pm 180^\circ$ false - 0° 3) 其它, true - 正向 false - 负向
ec	错误码

stop()

void rokae.clr.BaseRobot.stop (out ErrorCode ec)

暂停机器人运动; 暂停后可调用 moveStart()继续运动。若需要完全停止, 不再执行已添加的指令, 可调用 moveReset()。

目前支持 stop2 停止类型, 规划停止不断电, 参见 StopLevel。

参数

ec	错误码
----	-----

checkPath() [1/4]

double[] BaseRobot.checkPath(CartesianPosition start, double[] start_joint, CartesianPosition target, out ErrorCode ec)

检验直线轨迹是否可达。支持导轨, 返回的目标轴角为轴数+外部轴数

参数

start	起始点
start_joint	起始轴角 [弧度]
target	目标点
ec	错误码, 含不可达的错误原因

返回

计算出的目标轴角, 仅当无错误码时有效

checkPath() [2/4]

int BaseRobot. checkPath(double[] start_joint, CartesianPosition[] targets, ref double[] target_joint_calculated, out ErrorCode ec)

校验多个直线轨迹	
参数	
start_joint	起始轴角 [弧度]
targets	笛卡尔点位，至少需要 2 个点，第一个点是起始点
target_joint_calculated	若校验通过。返回计算出的目标轴角
ec	错误码，含不可达的错误原因
返回	
若校验失败，返回 points 中出错目标点的下标。其它情况返回 0	

checkPath() [3/4]	
double[] BaseRobot.checkPath(CartesianPosition start, double[] start_joint, CartesianPosition aux, CartesianPosition target, out ErrorCode ec)	
检验圆弧轨迹是否可达	
参数	
start	起始点
start_joint	起始轴角 [弧度]
aux	辅助点
target	目标点
ec	错误码，含不可达的错误原因
返回	
计算出的目标轴角，仅当无错误码时有效	

checkPath() [4/4]	
double[] BaseRobot.checkPath(CartesianPosition start, double[] start_joint, CartesianPosition aux1, CartesianPosition aux2, double angle, double rot_type, out ErrorCode ec)	
检验全圆轨迹是否可达	
参数	
start	起始点
start_joint	起始轴角 [弧度]
aux1	辅助点 1
aux2	辅助点 2
angle	全圆执行角度，单位弧度，不等于零时代表校验全圆轨迹
rot_type	全圆旋转类型
ec	错误码，含不可达的错误原因
返回	
计算出的目标轴角，仅当无错误码时有效	

8.3.3 通信相关

getAI()	
double rokae.clr.BaseRobot.getAI(uint board, uint port, out ErrorCode ec)	
读取模拟量输入信号值	
参数	
board	IO 板序号
port	信号端口号
ec	错误码

返回

信号值

getDI()

bool rokae.clr.BaseRobot.getDI (uint board, uint port, out ErrorCode ec)

读取数字量输入信号值

参数

board IO 板序号
port 信号端口号
ec 错误码

返回

true-开 | false-关

getDO()

bool rokae.clr.BaseRobot.getDO (uint board, uint port, out ErrorCode ec)

查询数字输出量信号值

参数

board IO 板序号
port 信号端口号
ec 错误码

返回

true-开 | false-关

getKeypadState()

bool[] rokae.clr.Cobot.getKeypadState(out ErrorCode ec)

获取末端按键状态，不支持的机型会返回错误码

参数

ec 错误码

readRegister()

void rokae.clr.BaseRobot.readRegister (string name, ref List<T> value, out ErrorCode ec)

读取寄存器值。可读取单个寄存器，或按索引读取寄存器数组

参数

name 寄存器名称
value 寄存器数值，允许的类型有 bool/int/float
ec 错误码

readRegister()

void rokae.clr.BaseRobot.readRegister (string name, uint index, ref T value, out ErrorCode ec)

读取寄存器值。可读取单个寄存器，或按索引读取寄存器数组

参数

name 寄存器名称
index 按索引读取寄存器数组中元素，从 0 开始。 下列两种情况会报错：1) 索引超出数组长度; 2) 寄存器不是数组但 index 大于 0
value 寄存器数值，允许的类型有 bool/int/float
ec 错误码

setAO()	
void rokae.clr.BaseRobot.setAO (uint board, uint port, double value, out ErrorCode ec)	
设置模拟量输出信号	
参数	
board	IO 板序号
port	信号端口号
value	输出值
ec	错误码

setDI()	
void rokae.clr.BaseRobot.setDI (uint board, uint port, bool state, out ErrorCode ec)	
设置数字量输入信号，仅当输入仿真模式打开时可以设置(见 setSimulationMode())	
参数	
board	IO 板序号
port	信号端口号
state	true-开 false-关
ec	错误码

setDO()	
void rokae.clr.BaseRobot.setDO (uint board, uint port, bool state, out ErrorCode ec)	
设置数字量输出信号值	
参数	
board	IO 板序号
port	信号端口号
state	true-开 false-关
ec	错误码

setSimulationMode()	
void rokae.clr.BaseRobot.setSimulationMode (bool state, out ErrorCode ec)	
设置输入仿真模式	
参数	
state	true - 打开 false - 关闭
ec	错误码

writeRegister()	
void rokae.clr.BaseRobot.writeRegister (string name, uint index, T value, out ErrorCode ec)	
写寄存器值。可写入单个寄存器，寄存器数组，或按索引写入寄存器数组中某一元素，如果要写入整个寄存器数组，value 传入对应类型的 vector，index 值被忽略。	
参数	
name	寄存器名称
index	数组索引，从 0 开始。下列两种情况会报错：1) 索引超出数组长度；2) 寄存器不是数组但 index 大于 0
value	写入的数值，允许的类型有 bool/int/float/bool[]/int[]/float[]
ec	错误码

setxPanelVout()

```
void rokae.clr.Cobot.setxPanelVout (xPanelOpt.Vout opt, out ErrorCode ec)
```

设置 xPanel 对外供电模式。注：仅部分机型支持 xPanel 功能，不支持的机型会返回错误码

参数

opt	模式
ec	错误码

setxPanelRS485 ()

```
void rokae.clr.Cobot.setxPanelRS485(xPanelOpt::Vout opt, bool if_rs485, out ErrorCode ec)
```

使用 CR 和 SR 末端的 485 通信功能，需要修改末端的参数配置，可通过此接口进行参数配置

参数

opt	对外供电模式，0：不输出，1：保留，2：12v，3：24v
if_rs485	接口工作模式，是否打开末端 485 通信
ec	错误码

XPRWModbusRTUReg ()

```
void rokae.clr.Baserobot.XPRWModbusRTUReg (int slave_addr, int fun_cmd, int reg_addr, string data_type, int num, ref int[] data_array, bool if_crc_reverse, out ErrorCode ec)
```

通过 xPanel 末端读写 modbus 寄存器

参数

slave_addr	设备地址 0-65535
fun_cmd	功能码 0x03 0x04 0x06 0x10
reg_addr	寄存器地址 0-65535
data_type	支持的数据类型 int32、int16、uint32、uint16
num	一次连续操作寄存器的个数 0-3，类型为 int16/uint16 时，最大为 3；类型为 int32/uint32、float 时，最大为 1，功能码为 0x06 时，此参数无效
data_array	发送或接收数据的数组，非 const，功能码为 0x06 时，只使用此数组的数据[0]，此时 num 的值无效，除了 0x06 功能码，大小需要与 num 匹配
if_crc_reverse	是否改变 CRC 校验高低位，默认 false，少数厂家末端工具需要反转
ec	错误码

XPRWModbusRTUCoil ()

```
void rokae.clr.Cobot.XPRWModbusRTUCoil (int slave_addr, int fun_cmd, int coil_addr, int num, ref bool[] data_array, bool if_crc_reverse, out ErrorCode ec)
```

通过 xPanel 末端读写 modbus 线圈或离散输入

参数

slave_addr	设备地址 0-65535
fun_cmd	功能码 0x01 0x02 0x05 0x0F
coil_addr	线圈或离散输入寄存器地址 0-65535
num	一次连续读写线圈离散输入的个数 (0-48)，功能码 0x05 时，此值无效
data_array	发送或接收数据的数组，非 const，功能码为 0x05 时，只使用此数组的数据[0]，此时 num 的值无效，除了 0x05 功能码，大小需要与 num 匹配
if_crc_reverse	是否改变 CRC 校验高低位，默认 false，少数厂家末端工具需要反转
ec	错误码

XPRS485SendData ()	
void rokae.clr.Cobot.XPRS485SendData(int send_byte, int rev_byte, Byte[] send_data, ref Byte[] rev_data, out ErrorCode ec)	
通过 xPanel 末端直接传输 RTU 协议裸数据	
参数	
send_byte	发送字节长度 0-16
rev_byte	接收字节长度 0-16
send_data	发送字节数据 数组长度需要和 send_byte 参数一致
rev_data	接收字节数据 数组长度需要和 rev_byte 参数一致
ec	错误码

8.3.4 RL 工程

loadProject()	
void rokae.clr.BaseRobot.loadProject (string name, List<string> tasks, out ErrorCode ec)	
加载工程	
参数	
name	工程名称
tasks	要运行的任务。该参数必须指定，不能为空，否则无法执行工程。
ec	错误码

pauseProject()	
void rokae.clr.BaseRobot.pauseProject (out ErrorCode ec)	
暂停运行工程	
参数	
ec	错误码

ppToMain()	
void rokae.clr.BaseRobot.ppToMain (out ErrorCode ec)	
程序指针跳转到 main。调用后，等待控制器解析完工程后返回，阻塞时间视工程大小而定，超时时间设定为 10 秒。	
参数	
ec	错误码。错误码能提供的信息有限，不能反馈如 RL 语法错误、变量不存在等错误。可通过 queryControllerLog()查询错误日志。

projectsInfo()	
List<RLProjectInfo> rokae.clr.BaseRobot.projectsInfo (out ErrorCode ec)	
查询工控机中 RL 工程名称及任务	
参数	
ec	错误码
返回	
工程信息列表，若没有创建工程则返回空列表	

runProject()	
void rokae.clr.BaseRobot.runProject (out ErrorCode ec)	
开始运行当前加载的工程	

参数

ec 错误码

setProjectRunningOpt()

void rokae.clr.BaseRobot.setProjectRunningOpt (double rate, bool loop, out ErrorCode ec)

更改工程的运行速度和循环模式

参数

rate 运行速率, 范围 0.01 - 1

loop true - 循环执行 | false - 单次执行

ec 错误码

toolsInfo()

List<WorkToolInfo> rokae.clr.BaseRobot.toolsInfo (out ErrorCode ec)

查询当前加载工程的工具信息

参数

ec 错误码

返回

工具信息列表, 若未加载任何工程或没有创建工具, 则返回默认工具 tool0 的信息

wobjInfo()

List<WorkToolInfo> rokae.clr.BaseRobot.wobjInfo (out ErrorCode ec)

查询当前加载工程的工件信息

参数

ec 错误码

返回

工件信息列表, 若未加载任何工程或没有创建工件, 则返回空列表

8.3.5 协作相关

calibrateForceSensor()

void rokae.clr.Cobot.calibrateForceSensor(bool all_axes, int axis_index, out ErrorCode ec)

力传感器标定。标定过程需要约 100ms, 该函数不会阻塞等待标定完成。 标定前需要通过 setToolset()设置正确的负载(toolset.load), 否则会影响标定结果准确性。

参数

all_axes true - 标定所有轴 | false - 单轴标定

axis_index 轴下标, 范围[0, 6), 仅当单轴标定时生效

ec 错误码

cancelRecordPath()

void rokae.clr.Cobot.cancelRecordPath (out ErrorCode ec)

取消录制, 缓存的路径数据将被删除

参数

ec 错误码

disableDrag()	
void rokae.clr.Cobot.disableDrag (out ErrorCode ec)	
关闭拖动	
参数	
ec	错误码

enableDrag() [1/2]	
void rokae.clr.Cobot.enableDrag (DragOpt.Space space, DragOpt.Type type, out ErrorCode ec)	
打开拖动	
参数	
space	拖动空间. 轴空间拖动仅支持自由拖拽类型
type	拖动类型
ec	错误码

enableDrag() [1/2]	
void rokae.clr.Cobot.enableDrag (DragOpt.Space space, DragOpt.Type type, bool enable_drag_button, out ErrorCode ec)	
打开拖动	
参数	
space	拖动空间. 轴空间拖动仅支持自由拖拽类型
type	拖动类型
enable_drag_button	True - 打开拖动功能之后可以直接拖动机器人，不需要按住末端按键
ec	错误码

queryPathLists()	
List<string> rokae.clr.Cobot.queryPathLists (out ErrorCode ec)	
查询已保存的所有路径名称	
参数	
ec	错误码
返回	
名称列表, 若没有路径则返回空列表	

removePath()	
void rokae.clr.Cobot.removePath (string name, bool removeAll, out ErrorCode ec)	
删除已保存的路径	
参数	
name	要删除的路径名称
removeAll	是否删除所有路径
ec	错误码。若路径不存在，错误码不会被置位

replayPath()	
void rokae.clr.Cobot.replayPath (string name, double rate, out ErrorCode ec)	
运动指令-路径回放	
参数	
name	要回放的路径名称

rate	回放速率, 应小于 3.0, 1 为路径原始速率。注意当速率大于 1 时, 可能产生驱动器无法跟随错误
ec	错误码

saveRecordPath()	
void rokae.clr.Cobot.saveRecordPath (string name, out ErrorCode ec, string saveAs) 保存录制好的路径	
参数	
name	路径名称
saveAs	重命名, 可传入空字符串。如果已录制好一条路径但没有保存, 则用该名字保存路径。如果没有未保存的路径, 则将已保存的名为"name"的路径重命名为"saveAs"
ec	错误码

startRecordPath ()	
void rokae.clr.Cobot.startRecordPath (out ErrorCode ec) 开始录制路径。录制的时长需要限制在 30 分钟以内, 此接口不会阻塞等待, 录制完毕调用 stopRecordPath()来停止录制	
参数	
ec	错误码

stopRecordPath()	
void rokae.clr.Cobot.stopRecordPath (out ErrorCode ec) 停止录制路径, 若录制成功(无错误码)则路径数据保存在缓存中	
参数	
ec	错误码

8.3.6 力控指令

fclnit()	
void rokae.clr.ForceControl.fclnit(FrameType type, out ErrorCode ec) 力控初始化	
参数	
type	力控坐标系, 支持 world/wobj/tool/baseFrame/flange。工具工件坐标系使用 setToolset()设置的坐标系
ec	错误码

fcMonitor()	
void rokae.clr.ForceControl.fcMonitor(bool enable, out ErrorCode ec) 启动/关闭力控模块保护监控。设置监控参数(setJointMaxVel, setJointMaxMomentum, setJointMaxEnergy, setCartesianMaxVel)后, 不立即生效, 调用 fcMonitor(true)后开始生效, 并且一直保持, 直到调用 fcMotion(false)后结束。 结束后保护阈值恢复成默认值, 即仍然会有保护效果, 关闭监控后不再是用户设置的参数	
参数	
enable	true - 打开 false - 关闭
ec	错误码

fcStart()
void rokae.clr.ForceControl.fcStart(out ErrorCode ec) 开始力控，fcInit()之后调用。如需在力控模式下执行运动指令，fcStart()之后可执行。注意，如果在 fcStart()之前通过 moveAppend()下发了运动指令但未开始运动，fcStart 之后就会执行这些运动指令。
参数
ec 错误码

fcStop()
void rokae.clr.ForceControl.fcStop(out ErrorCode ec) 停止力控
参数
ec 错误码

forceControl()
ForceControl rokae.clr.Cobot.forceControl() 力控指令类
返回
ForceControl

getEndTorque()
void rokae.clr.ForceControl.getEndTorque(FrameType ref_type, ref double[] joint_torque_measured, ref double[] external_torque_measured, ref double[] cart_torque, ref double[] cart_force, out ErrorCode ec) 获取当前力矩信息
参数
ref_type 力矩相对的参考系 • FrameType::world –末端相对世界坐标系的力矩信息 • FrameType::flange –末端相对于法兰盘的力矩信息 • FrameType::tool –末端相对于 TCP 点的力矩信息
joint_torque_measured 轴空间测量力信息,力传感器测量到的各轴所受力矩, 单位 Nm
external_torque_measured 轴空间外部力信息, 控制器根据机器人模型和测量力计算出的各轴所受力矩信息, 单位 Nm
cart_torque 笛卡尔空间各个方向[X, Y, Z]受到的力矩, 单位 Nm
cart_force 笛卡尔空间各个方向[X, Y, Z]受到的力, 单位 N
ec 错误码

pauseOverlay()
void rokae.clr.ForceControl.pauseOverlay(out ErrorCode ec) 暂停搜索运动。startOverlay()之后调用生效
参数
ec 错误码

restartOverlay()
void rokae.clr.ForceControl.restartOverlay(out ErrorCode ec) 重新开启暂停的搜索运动。pauseOverlay()之后调用生效。
参数
ec 错误码

setCartesianDesiredForce()

void rokae.clr.ForceControl.setCartesianDesiredForce(double[] value, out ErrorCode ec)

设置笛卡尔期望力/力矩。fcStart()之后可调用

参数

value 依次为: X Y Z 方向笛卡尔期望力, 范围[-60,60], 单位 N; X Y Z 方向笛卡尔期望力矩, 范围[-10,10], 单位 Nm
ec 错误码

setCartesianMaxVel()

void rokae.clr.ForceControl.setCartesianMaxVel(double[] velocity, out ErrorCode ec)

设置力控模式下, 机械臂末端相对基坐标系的最大速度

参数

velocity 依次为: X Y Z [m/s], A B C [rad/s], 范围 >=0
ec 错误码

setCartesianNullspaceStiffness()

void rokae.clr.ForceControl.setCartesianNullspaceStiffness(double stiffness, out ErrorCode ec)

设置笛卡尔零空间阻抗刚度。fcInit()之后调用生效

参数

stiffness 范围[0,4], 大于 4 会默认设置为 4, 单位 Nm/rad
ec 错误码

setCartesianStiffness()

void rokae.clr.ForceControl.setCartesianStiffness(double[] stiffness, out ErrorCode ec)

设置笛卡尔阻抗刚度, fcInit()之后调用生效。各机型的最大刚度不同, 请参考《xCore 控制系统手册》SetCartCtrlStiffVec 指令的说明

参数

stiffness 依次为: X Y Z 方向阻抗力刚度[N/m], X Y Z 方向阻抗力矩刚度[Nm/rad]
ec 错误码

setControlType()

void rokae.clr.ForceControl.setControlType(int type, out ErrorCode ec)

设置阻抗控制类型

参数

type 0 - 关节阻抗 | 1 - 笛卡尔阻抗
ec 错误码

setForceCondition()

void rokae.clr.ForceControl.setForceCondition(double[] range, bool isInside, double timeout, out ErrorCode ec)

设置与接触力有关的终止条件

参数

range 各方向上的力限制 { X_min, X_max, Y_min, Y_max, Z_min, Z_max }, 单位 N。设置下限时, 负值表示负方向上的最大值;
设置上限时, 负值表示负方向上的最小值。

isInside	超出限制条件时停止等待; false - 符合限制条件时停止等待
timeout	超时时间, 范围[1, 600], 单位秒
ec	错误码

setJointDesiredTorque()	
void rokae.clr.ForceControl.setJointDesiredTorque(double[] torque, out ErrorCode ec)	
设置关节期望力矩。fcStart()之后可调用	
参数	
torque	力矩值, 范围[-30,30], 单位 Nm
ec	错误码

setJointMaxEnergy()	
void rokae.clr.ForceControl.setJointMaxEnergy(double[] energy, out ErrorCode ec)	
设置力控模式下轴最大动能。 计算方式: $F \cdot v$, 可以理解为功率, F 为力矩传感器读数, v 为关节速度, 如果超过 30 个周期都超过动能阈值则触发保护	
参数	
energy	动能 $[N \cdot \text{rad/s}]$, 范围 ≥ 0
ec	错误码

setJointMaxMomentum()	
void rokae.clr.ForceControl.setJointMaxMomentum(double[] momentum, out ErrorCode ec)	
设置力控模式下轴最大动量。 计算方式: $F \cdot t$, 可以理解为冲量, F 为力矩传感器读数, t 为控制周期, 如果超过 30 个周期都超过动量阈值则触发保护	
参数	
momentum	动量 $[N \cdot s]$, 范围 ≥ 0
ec	错误码

setJointMaxVel()	
void rokae.clr.ForceControl.setJointMaxVel(double[] velocity, out ErrorCode ec)	
设置力控模式下的轴最大速度	
参数	
velocity	轴速度 $[\text{rad/s}]$, 范围 ≥ 0
ec	错误码

setJointStiffness()	
void rokae.clr.ForceControl.setJointStiffness(double[] stiffness, out ErrorCode ec)	
设置关节阻抗刚度, fcInit()之后调用生效. 各机型的最大刚度不同, 请参考《xCore 控制系统手册》SetJntCtrlStiffVec 指令的说明	
参数	
stiffness	各轴刚度
ec	错误码

setLissajousOverlay()	
void rokae.clr.ForceControl.setLissajousOverlay(int plane, double amplify_one, double frequency_one, double amplify_two, double frequency_two, double phase_diff, out ErrorCode ec)	
设置平面内的莉萨如搜索运动。 设置阻抗控制类型为笛卡尔阻抗(即 setControlType(1))之后, startOverlay()之	

前调用生效。

参数

plane	搜索运动参考平面: 0 - XY 1 - XZ 2 - YZ
amplify_one	搜索运动一方向幅值, 范围[0, 20], 单位 Nm
frequency_one	搜索运动一方向频率, 范围[0, 5], 单位 Hz
amplify_two	搜索运动二方向幅值, 范围[0, 20]单位 Nm
frequency_two	搜索运动二方向频率, 范围[0, 5], 单位 Hz
phase_diff	搜索运动两个方向相位偏差, 范围[0, PI], 单位弧度
ec	错误码

setLoad()

void rokae.clr.ForceControl.setLoad(Load load, out ErrorCode ec)

设置力控模块使用的负载信息, fcStart()之后可调用。

参数

load	负载
ec	错误码

setPoseBoxCondition()

void rokae.clr.ForceControl.setPoseBoxCondition(Frame supervising_frame, double[] box, bool isInside, double timeout, out ErrorCode ec)

设置与接触位置有关的终止条件

参数

supervising_frame	长方体所在的参考坐标系, 相对于外部工件坐标系。外部工件坐标系是通过 setToolset() 设置的, toolset.reference
box	定义一个长方体 { X_start, X_end, Y_start, Y_end, Z_start, Z_end }, 单位米
isInside	超出限制条件时停止等待; false - 符合限制条件时停止等待
timeout	超时时间, 范围[1, 600], 单位秒
ec	错误码

setSineOverlay()

void rokae.clr.ForceControl.setSineOverlay(int line_dir, double amplify, double frequency, double phase, double bias, out ErrorCode ec)

设置绕单轴旋转的正弦搜索运动。设置阻抗控制类型为笛卡尔阻抗(即 setControlType(1))之后, startOverlay() 之前调用生效。各机型的搜索运动幅值上限和搜索运动频率上限不同, 请参考《xCore 控制系统手册》SetSineOverlay 指令的说明。

参数

line_dir	搜索运动参考轴: 0 - X 1 - Y 2 - Z
amplify	搜索运动幅值, 单位 Nm
frequency	搜索运动频率, 单位 Hz
phase	搜索运动相位, 范围[0, PI], 单位弧度
bias	搜索运动偏置, 范围[0, 10], 单位 Nm
ec	错误码

setTorqueCondition()

void rokae.clr.ForceControl.setTorqueCondition(double[] range, [MarshalAs(UnmanagedType.U1)] bool isInside, double timeout, out ErrorCode ec)

设置与接触力矩有关的终止条件	
参数	
range	各方向上的力矩限制 { X_min, X_max, Y_min, Y_max, Z_min, Z_max }, 单位 Nm。设置下限时, 负值表示负方向上的最大值; 设置上限时, 负值表示负方向上的最小值。
isInside	超出限制条件时停止等待; false - 符合限制条件时停止等待
timeout	超时时间, 范围[1, 600], 单位秒
ec	错误码

startOverlay()	
void rokae.clr.ForceControl.startOverlay(out ErrorCode ec)	
开启搜索运动, fcStart()之后调用生效。搜索运动为前序设置的 setSineOverlay()或 setLissajousOverlay()的叠加	
参数	
ec	错误码

stopOverlay()	
void rokae.clr.ForceControl.stopOverlay(out ErrorCode ec)	
停止搜索运动	
参数	
ec	错误码

waitCondition()	
void rokae.clr.ForceControl.waitCondition(out ErrorCode ec)	
激活前序设置的终止条件并等待, 直到满足这些条件或者超时	
参数	
ec	错误码

ROKAE 珞石
轻型机器人专家



 **400-010-8700**

北京总部：北京市海淀区农科院西路6号海青大厦A座7层

山东分公司：济宁市邹城市中心店镇机电产业园恒丰路888号

苏州分公司：苏州工业园区星湖街328号创意产业园1-A1F

深圳分公司：深圳市宝安区中粮福安机器人智造产业园10栋1楼