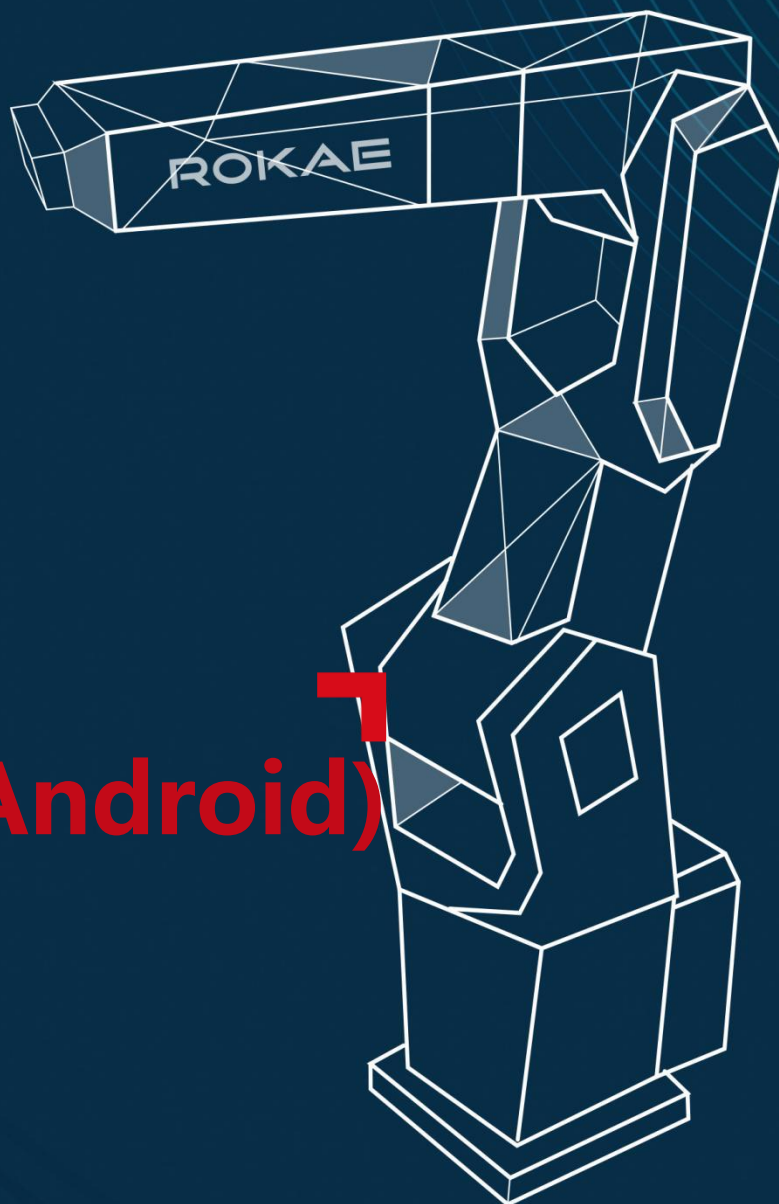


ROKAE 珞石
轻 型 机 器 人 专 家



xCore SDK(Android)

使用手册

让智造更高效

xCore SDK(Android)使用手册

[类别]

[备注]

控制系统版本: V2.2

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 2015-2023 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

目录.....	1
1 手册概述.....	3
1.1 关于本手册.....	3
1.2 手册对象.....	3
1.3 如何阅读产品手册	3
1.4 版本记录.....	3
2 概述.....	5
2.1 兼容性.....	5
2.1.1 控制器版本和机器人型号	5
2.1.2 编译平台及语言	5
2.2 非实时控制.....	5
2.3 实时控制.....	5
3 使用指南.....	6
3.1 硬件设置.....	6
3.2 网络配置.....	6
3.3 机器人功能设置	6
3.4 xCore SDK Android 工程包说明.....	6
3.5 Android 环境搭建	6
3.5.1 安装 Android Studio 软件和 SDK 库安装	6
3.5.2 安装 Java JDK 环境	7
4 接口说明.....	8
4.1 API 支持	8
4.2 机器人枚举类型	8
4.3 机器人基本操作及信息查询	8
4.4 运动控制	9
4.5 实时运动控制	10
4.5.1 参数设置	10
4.5.2 发送运动指令	10
4.5.3 错误异常	10
4.6 通信相关.....	11
4.7 RL 工程.....	12
4.8 协作相关.....	12
4.9 Android SDK 错误码和异常	13
5 注意事项与问题排查.....	14
5.1 与 RobotAssist 同时使用	14
5.2 兼容 RCI 客户端.....	14
5.2.1 首次使用	14

5.2.2 切换到使用 RCI 客户端	14
5.3 实时指令	14
5.4 问题排查	14
5.4.1 Windows – 网络连接问题	14
6 使用示例	15
6.1 非实时接口	15
6.1.1 示例一：信息查询, Jog, 拖动等	15
6.1.2 示例二：机器人 RL 工程执行和 joint 运动	15
6.2 实时运动控制	16
7 反馈与勘误	17
8 附录 A-Android API	18
8.1 数据类型	18
8.1.1 坐标类 Frame	18
8.1.2 负载信息 Load	18
8.1.3 机器人关节点位 JointPosition	18
8.1.4 机器人笛卡尔点位 CartesianPosition	18
8.1.5 机器人工具工件信息 ToolSet	18
8.1.6 机器人运动指令 MoveAbsJCommand	18
8.1.7 机器人运动指令 MoveCCommand	18
8.1.8 机器人运动指令 MoveJCommand	18
8.1.8 机器人运动指令 MoveLCommand	19
8.1.9 机器人运动指令 MoveSPCommand	19
8.1.10 机器人回调信息 RobotResult	19
8.2 枚举类型	19
8.2.1 机器人类型 RobotType	19
8.2.2 机型类别 RobotT.WorkType	19
8.2.3 机器人操作模式 RobotT.RobotOperateState	19
8.2.4 机器人上下电及急停状态 RobotT.PowerState	19
8.1.5 运动控制模式 RobotT.MotionControlMode	20
8.1.6 机器人运动空间参数 RobotT. JogMotionSpace	20
8.1.7 机器人拖动模式参数 RobotT.DragParameter	20
8.1.8 机器人拖动坐标系 RobotT.DragParameterSpace	20
8.1.9 机器人运动状态 RobotT. RobotOperateState	20
8.1.10 机器人位姿坐标系类型 RobotT. CoordinateType	21
8.3 方法	21
8.3.1 机器人基本操作及信息查询	21
8.3.2 Command 运动指令（非实时模式）	27
8.3.3 通信相关	27
8.3.4 协作相关	29
8.3.5 RL 工程运动相关	31

8.3.6 力控指令 32

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 (Android) 使用说明。

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、java 语言编程以及 Android 技术开发等所需的知识，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
V1.5	2022.6	初始版本；仅支持 c++；
V1.6	2022.09	Rokae SDK 初版，适配 xCore 版本 v1.6.1；仅支持 C++和 Python；
V1.7	2023.02	Rokae SDK 正式版本，适配 xCore 版本 v1.7；支持 Android 版本；
V3.0.X	2023.10	Rokae SDK 正式版本，适配 xCore 版本 v2.0；支持 Android 版本；
V3.0.4	2024.02.28	Rokae SDK 正式版本，适配 xCore 版本 v2.2；支持 Android 版本；

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，包括实时和非实时的运动控制，机器人通信相关的读写操作，查询及运行 RL 工程等；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 v2.2 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 编译平台及语言

xCore SDK 安卓版本使用 java 和 c++ 语言编写，目前支持 armeabi-v7a, arm64-v8a, x86, x86_64 等平台；Android SDK 最低使用不得低于 Android 8.0。

操作系统	编译器	平台	语言
Ubuntu 18.04/20.04/22.04	build-essential	X86_64	C++, Python
Windows 10	MSVC 14.1+	X86_64	C++, Python
Android	Android studio SDK API 31 NDK 22	armeabi-v7a, arm64-v8a, x86, x86_64	Java

2.2 非实时控制

xCore SDK 提供对机器人的非实时控制，主要通过给机器人发送运动指令，使用控制器内部的轨迹规划，完成路径规划和运动执行。非实时模式提供的操作有：

- 轴空间运动 (MoveAbsJ)
- 笛卡尔空间运动 (MoveL, MoveJ, MoveC)
- 机器人通信：数字量和模拟量 I/O，寄存器读写
- RL 工程的查询与执行
- 拖动与路径回放（只针对 xMate 协作机器人）
- 其他操作：清除报警，查询控制器日志等等

2.3 实时控制

Android xCore SDK 暂时不支持实时控制，后续版本会迭代开发；

实时控制包含了一系列底层控制接口，科研或二次开发用户可以使用该软件包实现最高达 1KHz 的实时控制，用于算法验证以及新应用的开发（暂时只支持 xMate 协作机器人）。目前支持 5 种控制模式：

- 轴空间位置控制
- 笛卡尔空间位置控制
- 轴空间阻抗控制
- 笛卡尔空间阻抗控制
- 直接力矩控制

3 使用指南

本章介绍如何配置并运行一个 xCore SDK Android 程序；其他语言版本（Python、C++、C#）请参考其他手册。

3.1 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 机器人控制系统使用手册 V1.7.0》。除网络配置外，使用 xCore SDK 无需其他额外的硬件设置。

3.2 网络配置

xCore SDK 通过以太网（TCP/IP）连接机器人。通过有线或无线连接皆可，使用户 PC 和机器人连接同一局域网。如果只使用非实时控制，对于网络性能要求不高，可以通过无线连接。

使用实时控制的话推荐通过有线直连到机器人。机器人配置有 2 个网口，一个是外网口，一个是直连网口。直连网口默认静态 IP 地址是 192.168.0.160。连接机器人有两种方式：

- 连接方式 1：机器人与用户 PC 采用网线直连的方式连接。如果用户工控机与机器人不处于同一个网段，需要配置用户 PC 的 IP 使其与机器人静态 IP 地址处于同一个网段，例如 192.168.0.22。
- 连接方式 2：机器人外网口连接路由器或者交换机，用户 PC 也连接路由器或者交换机，两者处于同一局域网。

注：推荐使用方式 1 进行连接，连接方式 2 网络通信质量差时可能会造成机器人运动不稳定现象。

注：使用 xCore SDK Android 版本使用方 1 连接，需要 Android 客户机支持网线插槽。

3.3 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。

切换到实时模式后，机器人重启后会保持打开状态，并自动切换成自动模式。

3.4 xCore SDK Android 工程包说明

Android_Sdk_Api_v3.0.x.zip	Android SDK Api 文档，里面有 SDK 相关函数以及参数说明。
SDK_Example	Android SDK 示例程序，example 里面包含 SDK 函数使用实例。
RokaeSdk_v3.0.3_20230928.aar	SDK 库文件，里面包含 SDK 函数。
xCore Android SDK 集成文档_v3.0.pdf	Android SDK 集成文件，以及基础使用方式。

3.5 Android 环境搭建

Android 开发环境搭建，需要包含以下环境：

1. Android Studio 开发软件；
2. Java JDK 1.8 以上，推荐 JDK 11；
3. Android SDK，推荐 Api 版本 30；
4. Android NDK（可选）；

3.5.1 安装 Android Studio 软件和 SDK 库安装

参考谷歌官方文档：<https://developer.android.google.cn/studio/install?hl=zh-cn>；

如需在 Windows 上安装 Android Studio，请按以下步骤操作：

- 如果您下载了.exe 文件（推荐），请双击以启动该文件；
- 如果您下载了.zip 文件；

1. 解压缩 zip。
2. 将 android-studio 文件夹复制到 Program Files 文件夹中。
3. 打开 android-studio > bin 文件夹。
4. 启动 studio64.exe (对于 64 位计算机) 或 studio.exe (对于 32 位计算机)。
5. 按照 Android Studio 中设置向导的指示操作，并安装推荐的 SDK 软件包。

如果您需要在 Mac 和 Linux 上安装 Android Studio，请参考上述链接中的文档和视频。

3.5.2 安装 Java JDK 环境

请自行参考 Oracle JDK 官网教程安装；

官网下载地址：<https://www.oracle.com/cn/java/technologies/downloads/>；

4 接口说明

本章列出各版本 xCore SDK 所支持的接口和功能简述。不同开发语言的版本对接口的功能定义基本一致，但是参数、返回值和调用方法会有区别。

本文档记录的接口均为 Android 端已支持的接口，与 C++ 差异化接口，将在后续版本更新。

4.1 API 支持

下表是各语言版本接口支持情况概览。

模块	API 功能	C++	Python	Android
rokae::Robot	基本操作	全部支持	全部支持	全部支持
	非实时运动	全部支持	全部支持	全部支持
	Jog 机器人	全部支持	不支持	不支持
	通信	全部支持	部分支持	部分支持
	RL 工程	全部支持	不支持	不支持
	协作相关	全部支持	部分支持	全部支持
rokae::Model	运动学计算	全部支持	全部支持	不支持
rokae::RtMotionControl	实时模式	全部支持	不支持	不支持
rokae::Planner	上位机路径规划	全部支持	不支持	不支持
rokae::xMateModel	运动学和动力学计算	全部支持 (仅 Linux)	不支持	不支持

4.2 机器人枚举类型

根据机器人构型和轴数不同，Android 版本的 SDK 提供了下列几个可供初始化的枚举值，初始化时会检查所选构型和轴数是否和连接的机器人匹配：

Java 枚举	对应机器人类型	适用机型
ROBOT_XMATE	XMateRobot	协作 6 轴
ROBOT_XMATE_ER_PRO	XMateErProRobot	协作 7 轴
ROBOT_STANDARD	StandardRobot	标准工业 6 轴
ROBOT_PCB3	PCB4Robot	工业 4 轴
ROBOT_PCB4	PCB3Robot	工业 3 轴

4.3 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	connect ()	无	
断开连接	disconnect ()	无	
机器人基本信息	getRobotInfo ()	无	控制器版本, 机型, 轴数
查询上电状态	getPowerState ()	无	on/off/Estop/Gstop
机器人上下电	setPowerState(state)	state - on/off	
查询当前操作模式	getOperateMode ()	无	auto/manual
切换手自动模式	setOperateMode(mode)	RobotT.RobotMode	

查询机器人运行状态	getOperateState ()	无	idle/jog/RLprogram/moving 等状态
获取法兰当前位姿	getFlangePos()	无	[X, Y, Z, A, B, C]
获取当前关节角度	getJointPos()	无	各轴角度 rad
获取当前关节速度	getJointVel()	无	各轴速率 rad/s
获取关节力矩	getJointTorque()	无	各轴力矩 Nm
查询当前工具工件组	getToolset()	无	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	
清除伺服报警	clearServoAlarm()	无	
查询控制器日志	queryControllerLog(count, level)	count - 查询个数 level - 日志等级	
查询 SDK 版本号	sdkVersion()	无	版本号
获取机器人法兰或末端的当前位姿	Posture()	RobotT.CoordinateType	
查询控制器最新的日志	queryControllerLog		
查询事件信息。	queryEventInfo		
设置接收事件的回调函数	setEventWatcher		
设置最大缓存指令个数	setMaxCacheSize	maxNumber	
设置输入仿真模式	setSimulationMode	boolean state	
设置的回调在每次执行时间	updateRobotState	long timeOut	
查询当前加载工程的工件信息	getWobjInfo		
动态调整机器人运动速率	adjustSpeedOnline	double scale	
读取基坐标系	baseFrame		

4.4 运动控制

非实时模式运动控制相关接口。

简述	接口	参数	返回
设置运动控制模式	setMotionControlMode(mode)	mode - NRT/RT/RL 工程	
重置运动缓存	moveReset()		
停止机器人运动	stop()		
执行运动指令	executeCommand(command)	command - 一条或多条 MoveL/MoveJ/MoveAbsJ/MoveC 指令	
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	

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	
查询运动指令执行错误码	lastErrorCode()		错误码

4.5 实时运动控制

Android 版本暂时不支持，计划后期版本迭代；

4.5.1 参数设置

设置开始运动前的参数。所有参数都只用于实时模式运动，与非实时运动、通过 RobotAssist 操作的运动均不互相影响。

4.5.2 发送运动指令

通过 appendCommand()接口发送运动指令，指令的数据类型（关节角度/笛卡尔位姿/关节力矩）应与控制模式匹配。控制器的控制周期是 1ms，所以指令下发的周期也应是 1ms。控制器的监测窗口是 2 秒，根据网络延迟阈值的高低，如果在监测窗口内没有收到足够的指令，将返回“网络不稳定”错误，然后做停止运动并下电处理。

4.5.3 错误异常

机器人在运动过程中会检测机器人状态，检测到异常会上报给 SDK，用户可以通过以下 20 个错误位判断异常类型。

注：Android 端不会直接抛出以下异常，但是会抛出以下异常对应的错误信息。

错误位	错误名称	错误原因	解决方法
0	kActualJointPositionLimitsViolation	实际轴角度超限	检查规划轨迹是否连续可导，一般规划出来的轨迹需要做到速度和角速度连续可导，机器人运行不平稳或出现异响优先考虑轨迹是否平滑，必要时做额外的滤波处理。
1	kActualCartesianPositionLimitsViolation	实际末端位姿超限	
2	kActualCartesianMotionGeneratorElbowLimitViolation	实际臂角超限	
3	kActualJointVelocityLimitsViolation	实际轴速度超限	
4	kActualCartesianVelocityLimitsViolation	实际末端速度超限	
5	kActualJointAccelerationLimitsViolation	实际轴加速度超限	
6	kActualCartesianAccelerationLimitsViolation	实际末端加速度超限	
7	kCommandJointPositionLimitsViolation	指令轴角度超限	
8	kCommandCartesianPositionLimitsViolation	指令末端位姿超限	

	onLimitsViolation		
9	kCommandCartesianMotionGeneratorElbowLimitViolation	指令臂角超限	
10	kCommandJointVelocityLimitsViolation	指令轴速度超限	
11	kCommandCartesianVelocityLimitsViolation	指令末端速度超限	
12	kCommandJointAccelerationLimitsViolation	指令轴加速度超限	
13	kCommandCartesianAccelerationLimitsViolation	指令末端加速度超限	
14	kCommandJointAccelerationDiscontinuity	指令轴加速度不连续	
17	kCommandTorqueDiscontinuity	指令力矩不连续	
18	kCommandTorqueRangeViolation	指令力矩超限	
15	kCollision	检测到碰撞	若频繁触发碰撞检测，应适当调高碰撞检测阈值，急停触发也有可能触发此报错
16	kCartesianPositionMotionGeneratorInvalidFrame	机器人奇异	笛卡尔空间运动时机器人不应该经过奇异位姿
19	kInstabilityDetection	检测到不稳定	1 检查丢包阈值是否过于低，一般设置为 10~20； 2 伺服报错，通常需要重启机器人； 3 按下了急停开关也会触发此错误；

4.6 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
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	
读取寄存器值	readRegister(name,	name - 寄存器名称	

	index, value)	index - 寄存器数组索引 value - 读取的数值	
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	

4.7 RL 工程

支持 RL 工程相关功能。

简述	接口	参数	返回值
打开拖动	enableDrag(space, type)	space - 拖动空间 type - 拖动类型	
开始运行当前加载的工程	runProject()		
程序指针跳转到 main	ppToMain()		
暂停运行工程	pauseProject()		
开启机器人运动	moveStart()		
停止机器人运动	moveStop()		
重置运动缓存	moveReset()		
添加机器人运动指令	moveAppend		
更改工程的运行速度和循环模式	setProjectRunOption	double rate boolean isLoop	
加载工程	loadProject	projectName tasks	

4.8 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)	space - 拖动空间 type - 拖动类型	
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,	name - 路径名称	

	saveAs)	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

4.9 Android SDK 错误码和异常

非实时接口的调用结果通过错误码反馈,每个接口都会返回一个 msg,可以通过 msg 来确定注意事项与问题排查。

5 注意事项与问题排查

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； ● 非实时模式运动的停止；
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； ● 所有机器人设置界面显示的功能打开状态和设定值等；

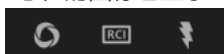
建议的控制方式是单一控制源，避免混淆。对于运动指令，推荐在每次使用 SDK 下发指令之前调用 `moveReset()` 接口来重置运动缓存。

5.2 兼容 RCI 客户端

对于原 RCI 客户端的用户，可以在控制器升级到 v1.7 之后直接使用 RokaeSDK。

5.2.1 首次使用

1. 确保 RobotAssist – 通信 – RCI 设置是关闭的状态，也可通过状态栏中间的机器人状态确认；
2. 调用 `setMotionControlMode(RtCommand)` 接口来打开 RCI 功能，若打开成功，机器人状态变为 RCI，状态栏的图标也显示 RCI：



3. 之后就可通过上述接口或 RobotAssist 打开关闭 RCI 功能。需要注意的是，如果进行了抹除配置操作，那么同样视为首次使用，需要再通过 SDK 的接口打开 RCI。

5.2.2 切换到使用 RCI 客户端

在用过 SDK 后，RCI 客户端就无法同时使用了。如果想切换回去，需调用 `useRciClient(true)`，调用前按照函数说明，确保 RCI 是关闭的状态。然后再通过 RobotAssist 打开关闭 RCI 功能。

5.3 实时指令

Android SDK 暂时不支持实时运动相关功能。

5.4 问题排查

5.4.1 Windows – 网络连接问题

实时运动指令和状态信息都通过 UDP 协议单播发送。如果出现“超时前未收到机器人状态反馈”的实时状态异常，请检查防火墙设置，UDP 是否是允许的状态。

6 使用示例

本章展示一些 C++ 示例程序，更多示例请见软件包中 examples。

6.1 非实时接口

6.1.1 示例一：信息查询，Jog，拖动等

```
BaseRobot robot = new XMateRobot(ip); // 创建机器人实例
RobotResult connect = robot.connect(); // 连接机器人
if (connect.getCode() != 0) return;

robot.setPowerState(false); // 机器人下电
robot.setPowerState(true); // 机器人上电
robot.getPowerState(); // 获取机器人上电状态
robot.getRobotInfo(); // 获取机器人信息
robot.getOperateState(); // 获取机器人运动状态
robot.setOperateMode(RobotT.RobotMode.Operate_automatic); // 设置机器人运动模式
robot.getFlangePos(); // 获取机器人法兰位姿
robot.posture(RobotT.CoordinateType.endInRef); // 获取机器人末端位姿
robot.carPosture(RobotT.CoordinateType.flangeInBase);

robot.setDI(0, 3, true); // 设置 DI
robot.getDI(0, 3); // 获取 DI

robot.setDO(0, 2, 1); // 设置 DO
robot.getDO(0, 2); // 获取 DO

robot.getAI(1, 1); // 获取 AI
robot.setAO(1, 1, 1.0); // 设置 Ao
```

6.1.2 示例二：机器人 RL 工程执行和 joint 运动

```
BaseRobot robot = new XMateRobot(ip);
robot.setLog(true);
RobotResult result = robot.connect();
if (result.getCode() != 0) return;

robot.getRobotInfo();
robot.setPowerState(true);
robot.setMotionControlMode(RobotT.MotionControlMode.NrtRLTask);
robot.setOperateMode(RobotT.RobotMode.Operate_automatic);

result = robot.projectsInfo();
String data = result.getData();
JSONArray objects = JSON.parseArray(data);
if (objects != null && objects.size() > 0) {
    JSONObject o = objects.getJSONObject(0);
```

```
String name = o.getString("name");
JSONArray taskList = o.getJSONArray("taskList");
if (!TextUtils.isEmpty(name) && taskList != null && taskList.size() > 0)
{
    List<String> arrayList = new ArrayList<>();
    for (int i = 0; i < taskList.size(); i++) {
        String taskName = (String) taskList.get(i);
        arrayList.add(taskName);
    }
    robot.loadProject(name, arrayList);
    robot.ppToMain();
    robot.runProject();

    Thread.sleep(50000); // 运动到一定时间暂定 5 秒
    robot.pauseProject();
    robot.runProject();
    reportState(robot);
}

Thread.sleep(3*60000); // 运动 3 分钟后停止工程
robot.moveStop();
robot.disconnect();
```

6.2 实时运动控制

Android SDK 暂不支持。

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-Android API

8.1 数据类型

8.1.1 坐标类 Frame

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

8.1.2 负载信息 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

8.1.3 机器人关节点位 JointPosition

double[] joints	关节角度值, 单位:弧度
double[] external	外部关节角度值, 单位:弧度,外部关节可缺省

8.1.4 机器人笛卡尔点位 CartesianPosition

double[] external	机器人位姿,[X, Y, Z, A,B,C] 单位:rad 和 m
double[] confData	轴配置数据, 长度为 8: [cf1, cf2, cf3, cf4, cf5, cf6, cf7, cf8]

8.1.5 机器人工具工件信息 ToolSet

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 ref	机器人参考坐标系相对世界坐标系转换

8.1.6 机器人运动指令 MoveAbsJCommand

JointPosition target	目标点位关节信息;
int speed	速度;
int zone	运动空间

8.1.7 机器人运动指令 MoveCCommand

CartesianPosition target	目标点位笛卡尔坐标信息;
CartesianPosition aux	辅助点位笛卡尔坐标信息;
int zone	运动空间
int speed	运动速度

8.1.8 机器人运动指令 MoveJCommand

CartesianPosition target	目标点位笛卡尔坐标信息;
int zone	运动空间
int speed	运动速度

8.1.8 机器人运动指令 MoveLCommand

CartesianPosition target	目标点位笛卡尔坐标信息;
int zone	运动空间
int speed	运动速度

8.1.9 机器人运动指令 MoveSPCommand

CartesianPosition target	目标点位笛卡尔坐标信息;
double radius	初始半径
double rStep	半径增长率
double angle	旋转角度
boolean dir	旋转方向, true: 顺时针, false: 逆时针
int speed	运动速度

8.1.10 机器人回调信息 RobotResult

String data	SDK 返回的信息;
String msg	请求 ID
int code	SDK 返回的错误码

8.2 枚举类型

8.2.1 机器人类型 RobotType

ROBOT_PCB3	3 轴工业机器人, PCB3 轴机型;
ROBOT_PCB4	4 轴工业机器人, PCB4 轴机型;
ROBOT_STANDARD	标准工业机器人;
ROBOT_XMATE	6 轴协作机器人, 例如:xMateCR7/12;
ROBOT_XMATE_ER_PRO	7 轴协作机器人, 例如:xMateER3 Pro;

8.2.2 机型类别 RobotT.WorkType

collaborative	协作机器人
industrial	工业机器人

8.2.3 机器人操作模式 RobotT.RobotOperateState

manual	手动
automatic	自动
unknown	未知(发生异常)

8.2.4 机器人上下电及急停状态 RobotT.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)

8.1.5 运动控制模式 RobotT.MotionControlMode

manual	手动
automatic	自动
unknown	未知(发生异常)

8.1.6 机器人运动空间参数 RobotT.JogMotionSpace

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

8.1.7 机器人拖动模式参数 RobotT.DragParameter

translationOnly	仅平移
rotationOnly	仅旋转
freely	自由拖拽

8.1.8 机器人拖动坐标系 RobotT.DragParameterSpace

cartesianSpace	世界坐标系
flange	法兰坐标系
jointSpace	轴空间

8.1.9 机器人运动状态 RobotT.RobotOperateState

Idle	机器人静止
jog	jog
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
unknown	未知

8.1.10 机器人位姿坐标系类型 RobotT. CoordinateType

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

8.3 方法

8.3.1 机器人基本操作及信息查询

connect ()	
RobotResult connect() 建立与机器人的连接。	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

disconnect()	
RobotResult disconnect() 断开机器人连接	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

getRobotInfo()	
RobotResult getRobotInfo() 获取机器人基本信息;	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

setPowerState()	
RobotResult setPowerState() 设置机器人上电;	
参数:	isON - 是否上电, true: 上电, false: 下电;
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

getPowerState()	
RobotResult getPowerState() 获取机器人上电信息;	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

setOperateMode()	
RobotResult setOperateMode() 设置机器人运动模式;	

参数:	mode - RobotMode 运动模式
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

getOperateMode()

RobotResult getOperateMode()

获取运动模式, 参考枚举 RobotMode 类;

参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

getOperateState()

RobotResult getOperateState()

获取机器人运动状态, 状态参考枚举 RobotOperateState 类

参数:	mode - RobotMode 运动模式
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json. 0: 机器人静止 1: jog 2: RCI 控制中 3: 拖动已开启 4: RL 工程运行中 5: Demo 演示中 6: 动力学辨识中 7: 摩擦力辨识中 8: 负载辨识中 9: 机器人运动中

adjustSpeedOnline()

RobotResult adjustSpeedOnline(double scale);

动态调整机器人运动速率; 动态调整机器人运动速率, 非实时模式时生效。

参数:	double scale: 运动速率
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

baseFrame()

RobotResult baseFrame()

读取基坐标系, 相对于世界坐标系

参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

carPosture

RobotResult carPosture(RobotT.CoordinateType ct)

获取机器人法兰或末端的当前位姿 ;

参数:	Ct 位姿坐标系类型
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

clearServoAlarm	
RobotResult clearServoAlarm()	
获取法兰相对基坐标系的位置信息	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

getFlangePos()	
RobotResult getFlangePos()	
获取法兰相对基坐标系的位置信息	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.
getJointPos()	
RobotResult getJointPos()	
机器人当前轴角度, 单位: [rad];	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

getJointTorque ()	
RobotResult getJointTorque()	
机器人当前关节速度, 单位: [rad/8];	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

getJointVel ()	
RobotResult getJointVel()	
关节力传感器数值, 单位: [Nm];	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

clearServoAlarm ()	
RobotResult clearServoAlarm ()	
清除伺服报警, 当有伺服报警且清除失败的情况下错误码置为-1;	
参数:	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

queryControllerLog()	
RobotResult queryControllerLog ()	
查询控制器最新的日志.	
参数:	count: 查询个数, 上限是 10 条. Levels: 指定日志等级, 空集合代表不指定 . info:0, warning:1, error:2.
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

moveReset ()	
RobotResult moveReset();	
重置运动缓存;	

注意：清空已发送的运动指令，默认速度重置为 100(v100)，默认转弯区重置为 0(fine)。每次程序开始运行并第一次执行运动指令之前，必须调用该函数来重置运动缓存，否则控制器可能会报错；

参数： -
返回值： RobotResult: 里面包含错误 Code 和正确的 Json.

moveStart ()

RobotResult moveStart ()

开启机器人运动；

参数： -
返回值： RobotResult: 里面包含错误 Code 和正确的 Json.

moveStop()

RobotResult moveStop()

停止机器人运动；

注意：目前支持 stop2 停止类型，规划停止不断电，参见 StopLevel。调用此接口后，已经下发的运动指令会被清除，不再执行。；

参数： -
返回值： RobotResult: 里面包含错误 Code 和正确的 Json.

startJog()

RobotResult startJog()

开启机器人 Jog 运动；

参数： space - jog 参考坐标系。工具/工件坐标系使用的是 RobotAssist 右上角设定的工具/工件。
 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；
 jogIndex - 笛卡尔空间 - 0~5 分别对应 XYZABC | 轴空间 - 关节序号，从 0 开始计数；
 direction - 方向，true - 正向 | false - 负向
返回值： RobotResult: 里面包含错误 Code 和正确的 Json.

lastErrorCode()

RobotResult lastErrorCode()

获取最新的错误码；

参数： -
返回值： RobotResult: 里面包含错误 Code 和正确的 Json.

setEventWatcher()

RobotResult setEventWatcher()

设置接收事件的回调函数；

参数： -
返回值： RobotResult: 里面包含错误 Code 和正确的 Json.

queryEventInfo ()

RobotResult queryEventInfo()

查询事件信息。与 setEventWatcher()回调时提供的信息相同，区别是这个接口是主动查询的方式；

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

queryEventInfo ()

RobotResult queryEventInfo()

查询事件信息。与 setEventWatcher()回调时提供的信息相同，区别是这个接口是主动查询的方式；

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

stopReceiveRobotState()

RobotResult 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。调用此函数后，实时运动的错误信息也会停止接收，建议在运动停止时调用；

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

updateRobotState()

RobotResult updateRobotState()

通过 setControlLoop()设置的回调在每次执行时会更新一次状态数据，因此不需要在回调函数中调用此接口；

参数 timeOut - 超时时间

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

getStateData()

RobotResult getStateData()

读取机器人状态数据；

参数 fieldName - 数据名

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

startReceiveRobotState()

RobotResult startReceiveRobotState(long interval, java.util.List<java.lang.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，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；

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

model()

RobotResult model()

获取模型类；

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setToolSet ()

RobotResult setToolSet(ToolSet toolSet)

使用已创建的工具和工件，设置工具工件组信息

参数 toolSet - 定义的工具名

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setToolSet ()

RobotResult setToolSet(java.lang.String toolName, java.lang.String wobjName)

使用已创建的工具和工件，设置工具工件组信息

参数 toolName - 已定义的工具名称
wobjName - 已定义的工件名称

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

sdkVersion()

RobotResult sdkVersion()

获取 SDK 版本号

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setDefaultZone()

RobotResult setDefaultZone(int zone)

设定默认转弯区;

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: 0 ~ 1 :
0 (fine) 1 ~ 20 : 10% 20 ~ 60 : 30% ∞ ~ 60 : 100%

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setMaxCacheSize()

RobotResult setMaxCacheSize(int maxNumber)

设定默认转弯区;

参数 maxNumber - 最大个数

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

getToolsInfo()

RobotResult getToolsInfo()

查询当前加载工程的工具信息;

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

getWobjInfo()

RobotResult getWobjInfo()

查询当前加载工程的工件信息;

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setDefaultSpeed()

RobotResult setDefaultSpeed(int speed)

设定默认运动速度;

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: 0 ~ 100 : 10% 100~200 : 30% 200~500 : 50%
500~800 : 80% ∞ ~800 : 100%

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

8.3.2 Command 运动指令（非实时模式）

moveAppend ()	
RobotResult moveAppend(java.util.List<Command> list, java.lang.String cmdId)	
添加机器人运动指令； 添加单条或多条运动指令，添加后调用 moveStart()开始运动；	
参数	list - 列表 cmdId - 运动执行 ID
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

moveAppend ()	
RobotResult moveAppend(Command[] arr, java.lang.String cmdId)	
添加机器人运动指令； 添加单条或多条运动指令，添加后调用 moveStart()开始运动；	
参数	arr - 数组 cmdId - 运动执行 ID
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

executeCommands()	
RobotResult executeCommands(java.util.List<Command> arr)	
执行单条或多条运动指令，调用后机器人立刻开始运动； 注解：须为同类型的指令；运动指令异步执行，如发生执行错误，需通过 lastErrorCode 获取错误码。参数中传入的 ec 仅反馈执行前的错误，如网络通信错误或参数个数不符。	
参数	cmdList - 列表 List，指令列表，允许的个数为 1-1000；
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

executeCommands()	
RobotResult executeCommands(Command[] arr)	
执行单条或多条运动指令，调用后机器人立刻开始运动； 注解：须为同类型的指令；运动指令异步执行，如发生执行错误，需通过 lastErrorCode 获取错误码。参数中传入的 ec 仅反馈执行前的错误，如网络通信错误或参数个数不符。	
参数	cmdList - 数组 Array，指令列表，允许的个数为 1-1000；
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

8.3.3 通信相关

getDI()	
RobotResult getDI(int board, int port)	

获取 DI 信号标量;

参数 board - 设备编号
port - 串口编号

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setDI()

RobotResult setDI(int board, int port, boolean state)

设置 DI 信号标量; 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 - 值

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

getDO()

RobotResult getDO(int board, int port)

获取 DO 信号标量;

参数 board - IO 板序号
port - 信号端口号

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setDO()

RobotResult setDO(int board, int channel, int state)

设置 DO 标量;

参数 board - IO 板序号
channel - 信号端口号
state - true: on | false: off

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setAO()

RobotResult setAO(int board, int port, double value)

设置 AO 标量;

参数 board - IO 板序号
port - 信号端口号
value - true: on | false: off

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setSimulationMode()

RobotResult setSimulationMode(boolean state)

设置输入仿真模式

参数 state - 状态 port - 信号端口号

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

readRegister()

RobotResult readRegister(java.lang.String name, int index, int regType)

读取寄存器值。 可读取单个寄存器, 寄存器数组, 或按索引读取寄存器数组。 如果要读取整个寄存器数组, value 传入对应类型的 vector, index 值被忽略。 更多...

参数 name - 名称
index - 标号

	regType - 1:int,2:List,3:float,4:List
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

writeRegister ()	
RobotResult writeRegister(java.lang.String name, int index, int regType, java.lang.Object value)	
写寄存器值。 可写入单个寄存器, 或按索引写入寄存器数组中某一元素。	
参数	name - 名称 index - 标号 regType - 寄存器类型 value - v
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

8.3.4 协作相关

enableDrag()	
RobotResult enableDrag(RobotT.DragParameterSpace space, RobotT.DragParameterType type)	
机器人开启拖拽模式。	
注解 : 拖拽模式, 只能在手动模式、下电状态打开, 且当 space 为 jog 时, 拖拽方向只能 freely	
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
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

disableDrag()	
RobotResult disableDrag()	
机器人停止拖拽模式。	
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
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

startRecordPath()	
RobotResult startRecordPath(int duration)	
开始录制路径。	
参数	duration - 路径的时长, 单位:秒, 范围 1~1800.此时长只做范围检查用, 到时长后控制器不会停止录制, 需要调用 stopRecordPath()来停止
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

stopRecordPath()	
RobotResult stopRecordPath()	
停止录制路径	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

cancelRecordPath()	
RobotResult cancelRecordPath()	
取消录制路径	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

saveRecordPath()

RobotResult saveRecordPath(java.lang.String path)

保存录制好的路径

参数 path - 路径名称**返回值:** RobotResult: 里面包含错误 Code 和正确的 Json.**saveRecordPath()**

RobotResult saveRecordPath(java.lang.String path, java.lang.String name)

保存录制好的路径

参数 path - 路径名称

name - 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, 则已将保存的名为"name"的路径重命名为"saveAsName"

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.**replayPath()**

RobotResult replayPath(java.lang.String name, double rate)

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

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.**removePath()**

RobotResult removePath(java.lang.String name, boolean removeAll)

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, 可选参数, 默认为否

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.**queryPathLists()**

RobotResult queryPathLists()

删除已保存的路径

参数 查询已保存的所有路径名称**返回值:** RobotResult: 里面包含错误 Code 和正确的 Json.**enableCollisionDetection()**

RobotResult enableCollisionDetection(double[] sensitivity, RobotT.StopLevel stopLevel, double fallback)

设置碰撞检测相关参数, 打开碰撞检测功能

参数 sensitivity - 数组, 长度和轴数相同, 碰撞检测灵敏度, 范围 0.01-2.0,

stopLevel - 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)

	fallback - 碰撞后后退距离, 单位: 米
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

disableCollisionDetection()	
RobotResult disableCollisionDetection() 设置碰撞检测相关参数, 关闭碰撞检测功能	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

8.3.5 RL 工程运动相关

projectsInfo()	
RobotResult projectsInfo() 查询工控机中 RL 工程名称及任务	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

loadProject()	
RobotResult loadProject(java.lang.String projectName, java.util.List<java.lang.String> tasks) 加载工程	
参数	projectName - 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。
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

setProjectRunOption()	
RobotResult setProjectRunOption(double rate, boolean isLoop) 更改工程的运行速度和循环模式	
参数	rate - 程序运行速率, 范围 0.01 - 1 isLoop - true - 循环执行 false - 单次执行
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

runProject()	
RobotResult runProject() 开始运行当前加载的工程	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

pauseProject()	
RobotResult pauseProject() 暂停运行工程	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

stopProject()	
RobotResult stopProject() 停止运行工程。	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

8.3.6 力控指令

fcInit()	
RobotResult fcInit(FrameType frame_type) 力控初始化	
参数	frame_type 力控坐标系。工具工件坐标系使用 setToolset()设置的坐标系.
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

fcStart()	
RobotResult fcStart() 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。	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

fcStop ()	
RobotResult fcStop() 停止力控	
参数	-
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

setLoad ()	
RobotResult setLoad(Load load) 设置力控模块使用的负载信息, fcStart()之后可调用。	
参数	load - 负载
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

setSineOverlay ()	
RobotResult setSineOverlay(int line_dir, double amplify, double frequency, double phase, double bias) 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
返回值:	RobotResult: 里面包含错误 Code 和正确的 Json.

setLissajousOverlay ()

RobotResult setLissajousOverlay(int plane, double amplify_one, double frequency_one, double amplify_two, double frequency_two, double phase_diff)

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

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

startOverlay ()

RobotResult startOverlay()

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 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

stopOverlay ()

RobotResult stopOverlay()

停止搜索运动

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

restartOverlay ()

RobotResult restartOverlay ()

重新开启暂停的搜索运动。pauseOverlay()之后调用生效。

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

pauseOverlay ()

RobotResult pauseOverlay()

暂停搜索运动。startOverlay()之后调用生效。

参数 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setForceCondition ()

RobotResult setForceCondition(double[] range, boolean isInside, double timeout)

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 - true - 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

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setTorqueCondition ()

RobotResult setTorqueCondition(double[] range, boolean isInside, double timeout)

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 - true - 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

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setPoseBoxCondition ()

RobotResult setPoseBoxCondition(Frame supervising_frame, double[] box, boolean isInside, double timeout)

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.ref
box - 定义一个长方体 { X_start, X_end, Y_start, Y_end, Z_start, Z_end }, 单位米
isInside - true - 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

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

waitCondition ()

RobotResult waitCondition()

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 -

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setJointStiffness ()

RobotResult setJointStiffness(java.util.List<java.lang.Double> stiffness)

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

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setCartesianStiffness ()

RobotResult setCartesianStiffness(double[] stiffness)

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]

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setCartesianNullSpaceStiffness ()

RobotResult setCartesianNullSpaceStiffness(double stiffness)

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

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

setJointDesiredTorque ()

RobotResult setJointDesiredTorque(double[] torque)

设置关节期望力矩。fcStart()之后可调用。

参数 torque - 力矩值, 范围[0,30], 单位 Nm**返回值:** RobotResult: 里面包含错误 Code 和正确的 Json.**setCartesianDesiredForce ()**

RobotResult setCartesianDesiredForce(double[] value)

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[0,60], 单位 N; X Y Z 方向笛卡尔期望力矩, 范围[0,20], 单位 Nm**返回值:** RobotResult: 里面包含错误 Code 和正确的 Json.**getEndTorque ()**

RobotResult getEndTorque(FrameType ref_type)

获取当前力矩信息

参数 ref_type - 力矩相对的参考系:

- 1) FrameType::world - 末端相对世界坐标系的力矩信息
- 2) FrameType::flange - 末端相对于法兰盘的力矩信息
- 3) FrameType::tool - 末端相对于 TCP 点的力矩信息

返回值: RobotResult: 里面包含错误 Code 和正确的 Json.

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

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