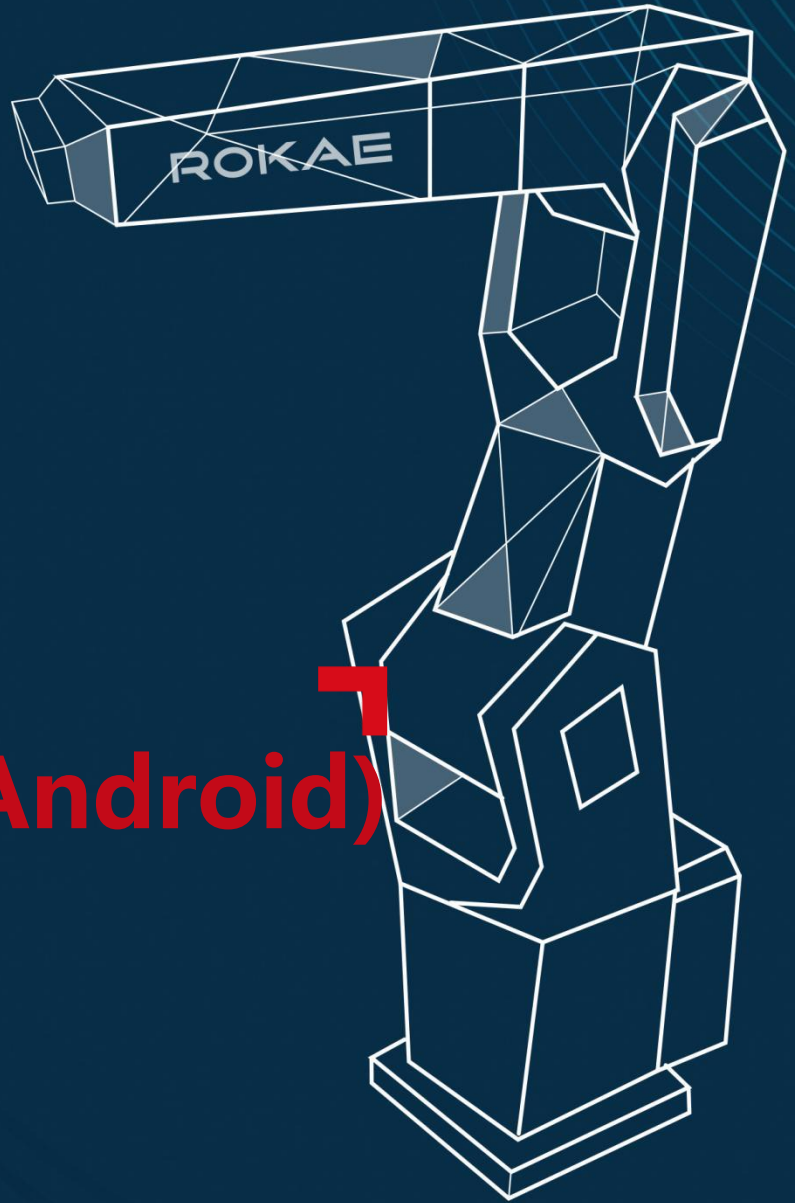


ROKAE 珞石
轻型机器人专家



xCore SDK(Android)

使用手册

让智造更高效

xCore SDK(Android)使用手册

控制系统版本：V1.7

文档版本：B

©版权所有 2015-2023 ROKAE 保留所有权利

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.....	9
4.5.1 参数设置.....	9
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.....	11
4.8 协作相关.....	11
4.9 Android SDK 错误码和异常.....	12
5 注意事项与问题排查.....	13
5.1 与 RobotAssist 同时使用.....	13
5.2 兼容 RCI 客户端.....	13
5.2.1 首次使用.....	13

5.2.2 切换到使用 RCI 客户端	13
5.3 实时指令	13
5.4 问题排查	13
5.4.1 Windows – 网络连接问题	13
6 使用示例	14
6.1 非实时接口	14
6.1.1 示例一：信息查询, Jog, 拖动等	14
6.2 实时运动控制	14
7 反馈与勘误	15
8 附录 A-Android API	16
8.1 枚举类型	16
8.1.1 机器人类型 RobotType	16
8.1.2 机型类别 RobotT.WorkType	16
8.1.3 机器人操作模式 RobotT.RobotOperateState	16
8.1.4 机器人上下电及急停状态 RobotT.PowerState	16
8.1.5 运动控制模式 RobotT.MotionControlMode	16
8.1.6 控制器实时控制模式 RobotT.RtControllerMode	16
8.1.7 机器人拖动模式参数 RobotT.DragParameter	17
8.1.8 机器人拖动坐标系 RobotT.DragParameterSpace	17
8.2 方法	17
8.2.1 机器人基本操作及信息查询	17
8.2.2 运动控制（非实时模式）	19
8.2.3 通信相关	20
8.2.4 协作相关	21

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 初始版本；

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 v1.7 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，一个是直联网口。直联网口默认静态 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

```
Release_Android SDK
├── Android_Sdk_Api_1.0.0.zip: Api 文档
├── SDK_Example: 示例程序
├── RokaeSdk_v1.0.0_20230106.aar: SDK 库文件
└── Rokae 机器人 Android SDK 接入手册.pdf
```

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 C++ 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	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	connectToRobot()		
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
获取法兰当前位姿	getFlangePos()		[X, Y, Z, A, B, C]
获取当前关节角度	getJointPos()		各轴角度 rad
获取当前关节速度	getJointVel()		各轴速率 rad/s
获取关节力矩	getJointTorque()		各轴力矩 Nm
查询当前工具工件组	getToolset()		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()		
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()		版本号

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	
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, index, value)	name - 寄存器名称 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

Android 版本 SDK 不支持 RL 工程相关功能。

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,	space - 拖动空间	

	type)	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, 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

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	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● 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；

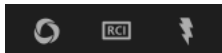
建议的控制方式是单一控制源，避免混淆。对于运动指令，推荐在每次使用 SDK 下发指令之前调用 `moveReset()` 接口来重置运动缓存。

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

```
RobotManager.getInstance().connect("192.168.0.160", RobotType.ROBOT_XMATE, new
HelperCallback(CODE_ROBOT_CONNECT)); // 连接机器人;

RobotManager.getInstance().setOperateMode(RobotT.RobotMode.Operate_automatic, new
HelperCallback(CODE_ROBOT_SET_OPERATE_MODE)); // 设置操作模式;

RobotManager.getInstance().setMotionControlMode(RobotT.MotionControlMode.NrtCommand, new
HelperCallback(CODE_ROBOT_SET_MOTION_MODE)); // 设置运动参数

RobotManager.getInstance().moveReset(new HelperCallback(CODE_ROBOT_DEFAULT)); //清理运动指令
RobotManager.getInstance().setPowerState(true, new HelperCallback(CODE_ROBOT_SET_POWER)); // 上电;

RobotManager.getInstance().getFlangePos(new HelperCallback(CODE_ROBOT_GET_JOINTS)); // 获取机器人当前位置

// 设置默认的转弯区和速率。
RobotManager.getInstance().setDefaultZone(0, new HelperCallback(CODE_ROBOT_DEFAULT));
RobotManager.getInstance().setDefaultSpeed(2000, new HelperCallback(CODE_ROBOT_DEFAULT));
```

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 枚举类型

8.1.1 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;

8.1.2 机型类别 RobotT.WorkType

collaborative	协作机器人
industrial	工业机器人

8.1.3 机器人操作模式 RobotT.RobotOperateState

manual	手动
automatic	自动
unknown	未知(发生异常)

8.1.4 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 控制器实时控制模式 RobotT.RtControllerMode

jointPosition	实时轴空间位置控制
cartesianPosition	实时笛卡尔空间位置控制
jointImpedance	实时轴空间阻抗控制
cartesianImpedance	实时笛卡尔空间阻抗控制
torque	实时力矩控制

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

8.2 方法

8.2.1 机器人基本操作及信息查询

connect ()	
public void connect(java.lang.String ip, RobotType type, JNIcallback callback)	
建立与机器人的连接。	
参数	ip - ip 地址 type - 机器人类型 callback - 回调函数

getRobotInfo ()	
public void getRobotInfo(JNIcallback callback)	
获取机器人信息。	
参数	callback - 回调函数

disconnect ()	
public void getRobotInfo(JNIcallback callback)	
断开机器人连接。	
参数	callback - 回调函数

getPowerState ()	
public void getRobotInfo(JNIcallback callback)	
获取机器人上电状态。	
参数	callback - 回调函数

setPowerState ()	
public void setPowerState(boolean isON, JNIcallback callback)	
设置机器人上电状态。	
参数	isON - 是否上电; callback - 回调函数

setPowerState ()	
public void setOperateMode(RobotT.RobotMode mode, JNIcallback callback) 获取机器人上电状态。	
参数	callback - 回调函数

setOperateMode ()	
public void setOperateMode(RobotT.RobotMode mode, JNIcallback callback) 设置机器人运动模式。	
参数	callback - 回调函数

getOperateMode ()	
public void getOperateMode(RobotT.RobotMode mode, JNIcallback callback) 获取机器人运动模式。	
参数	callback - 回调函数

getOperateState ()	
public void getOperateState (RobotT.RobotMode mode, JNIcallback callback) 获取机器人运动状态, 状态参考 RobotOperateState 类。	
参数	callback - 回调函数

getFlangePos ()	
public void getFlangePos(JNIcallback callback) 获取法兰相对基坐标系的位置信息。	
参数	callback - 回调函数

clearServoAlarm ()	
public void clearServoAlarm (JNIcallback callback) 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	callback - 回调函数

getToolsInfo ()	
public void getToolsInfo(JNIcallback callback) 查询当前加载工程的工具信息。	
参数	callback - 回调函数

getWobjInfo ()	
public void getWobjInfo (JNIcallback callback) 查询当前加载工程的工件信息。	
参数	callback - 回调函数

getJointPos ()	
public void getJointPos (JNICallback callback) 机器人当前轴角度, 单位: [rad]。	
参数	callback - 回调函数

getJointVel ()	
public void getJointVel (JNICallback callback) 关节力传感器数值, 单位: [Nm]。	
参数	callback - 回调函数

getJointTorque ()	
public void getJointTorque (JNICallback callback) 机器人当前关节速度, 单位: [rad/8]。	
参数	callback - 回调函数

8.2.2 运动控制 (非实时模式)

setMotionControlMode ()	
public void setMotionControlMode(RobotT.MotionControlMode mode, JNICallback callback) 设置运动模式。	
参数	Mode 机器人控制模式, 枚举 RobotT.MotionControlMode 类; callback - 回调函数

setDefaultSpeed ()	
public void setDefaultSpeed (int speed, JNICallback callback) 设定默认运动速度 注意: 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. 若不设置, 则为 v100。	
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%; callback - 回调函数

setDefaultZone ()	
public void setDefaultZone (int zone, JNICallback callback) 设定默认转弯区 注解: 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: 0 ~ 1: 0 (fine) 1 ~ 20: 10% 20 ~ 60: 30% ∞ ~ 60: 100%; callback - 回调函数

moveReset ()	
public void moveReset (JNICallback callback) 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	callback - 回调函数

stop ()	
public void stop (JNICallback callback)	
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	callback - 回调函数

append ()	
public void append(java.util.List<Command> cmdList, JNICallback callback)	
将单条或多条运动指令加入到缓存中。	
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;	
Command 运动指令类 : MoveJCommand , MoveAbsJCommand , MoveLCommand , MoveCCommand ;	
参数	cmdList - 列表 List, 指令列表, 允许的个数为 1-1000;
	callback - 回调函数

append ()	
public void append(Command [] cmdList, JNICallback callback)	
将单条或多条运动指令加入到缓存中。	
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;	
Command 运动指令类 : MoveJCommand , MoveAbsJCommand , MoveLCommand , MoveCCommand ;	
参数	cmdList -数组 Array, 指令列表, 指令列表, 允许的个数为 1-1000;
	callback - 回调函数

executeCommands ()	
public void executeCommands (JNICallback callback)	
执行运动指令函数, 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	callback - 回调函数

8.2.3 通信相关

getDI ()	
public void getDI (JNICallback callback)	
获取 DI 信号标量;	
参数	callback - 回调函数

getDO()	
public void getDO (JNICallback callback)	
获取 DO 信号标量;	
参数	callback - 回调函数

setDO()	
public void setDO(int board, int channel, int state, JNICallback callback)	
设置 DO 信号标量;	
参数	board - IO 板序号
	channel - 信号端口号

state - true-on false-off
callback - 回调函数

8.2.4 协作相关

enableDrag()	
public void enableDrag(RobotT.DragParameterSpace space, RobotT.DragParameterType type, JNIcallback callback)	
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 channel - 信号端口号; type - 拖动类型 callback - 回调函数

disableDrag ()	
public void disableDrag(JNIcallback callback)	
机器人停止拖拽模式;	
参数	callback - 回调函数

startRecordPath()	
public void startRecordPath(int duration, JNIcallback callback)	
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, 到时后控制器不会停止录制, 需要调用 stopRecordPath()来停止; callback - 回调函数

stopRecordPath()	
public void stopRecordPath (JNIcallback callback)	
停止录制路径;	
参数	callback - 回调函数

cancelRecordPath()	
public void cancelRecordPath (JNIcallback callback)	
取消录制路径;	
参数	callback - 回调函数

saveRecordPath()	
public void saveRecordPath (String path ,JNIcallback callback)	
保存录制好的路径;	
参数	path - 路径名称 callback - 回调函数

saveRecordPath()	
public void saveRecordPath (String path, String saveAsName ,JNIcallback callback)	
保存录制好的路径;	
参数	path - 路径名称 saveAsName - 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"saveAsName"
callback - 回调函数

replayPath()

public void replayPath (String name, double rate, ,JNICallback callback)

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;
 callback - 回调函数

removePath()

public void removePath(String name, boolean removeAll, ,JNICallback callback)

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;
 callback - 回调函数

queryPathLists()

public void queryPathLists (String name, boolean removeAll, ,JNICallback callback)

查询已保存的所有路径名称;

参数 callback - 回调函数, 名称列表, 若没有路径则返回空列表.

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