ESTUN ROBOTICS

ER 系列工业机器人 API 使用手册

V1.5

南京埃斯顿自动化股份有限公司
南京埃斯顿机器人工程有限公司

-本文档符合 ISO 体系要求，并会根据 IPD 体系变化更新-

目录

前言 **1**
读者对象 1
注意事项 1
安全说明 1
第 1 章 库功能介绍 **3**
1.1 本节概要 3
1.2 项目背景 3
1.3 设计说明 3
1.4 功能列表 4
1.5 模式权限说明 9
第 2 章 安装 **12**
2.1 本节概要 12
2.2 安装说明 12
2.2.1 解压库文件 12
2.2.2 配置环境变量 12
2.3 目录结构 13
第 3 章 使用操作 **14**
3.1 本节概要 14
3.2 C++工程搭建 14
3.2.1 环境要求 14
3.2.2 搭建步骤 14
3.1 C#工程搭建 19
3.1.1 环境要求 19
3.1.2 搭建步骤 19
3.2 VB 工程搭建 22
3.2.1 环境要求 22
3.2.2 搭建步骤 23
第 4 章 运行环境 **27**
4.1 本节概要 27
4.2 运行环境 27
4.2.1 软件支持 27
4.2.2 示教器端配置 27
第 5 章 典型应用案例 **29**
5.1 本节概要 29
5.2 上位机加载机器人示教程序并再现运行 29
5.3 上位机规划点位控制机器人运动 31
第 6 章 接口详细参考 **34**
6.1 本节概要 34
6.2 结构化数据 34
6.2.1 内部字符串 34
6.2.2 权限信息 34

6.2.3 坐标类型 35
6.2.4 系统模式 35
6.2.5 运动模式 35
6.2.6 空间坐标形态配置数据.....35
6.2.7 机器人位置 36
6.2.8 运动方向 38
6.2.9 AddDo.....38
6.2.10 AddDO_IO.....38
6.2.11 机器人速度.....39
6.2.12 机器人过渡区域 39
6.2.13 指令信息 40
6.2.14 负载参数 40
6.2.15 用户坐标参数 42
6.2.16 用户坐标类型 42
6.2.17 IO 类型 42
6.2.18 变量类型 43
6.2.19 变量作用域 43
6.2.20 程序信息 43
6.2.21 程序运行状态 43
6.2.22 机器人使能状态 43
6.3 接口列表 44
6.3.1 默认构造函数 44
6.3.2 有参构造函数 44
6.3.3 有参构造函数 44
6.3.4 日志打印开关 44
6.3.5 写本地日志 44
6.3.6 错误日志 45
6.3.7 连接服务器 45
6.3.8 断开连接 45
6.3.9 连接检测 46
6.3.10 获取当前工作状态 46
6.3.11 申请权限.....46
6.3.12 查询权限 46
6.3.13 释放权限 46
6.3.14 使能开关 47
6.3.15 获取使能状态 47
6.3.16 jog 使能开关.....47
6.3.17 获取 jog 使能状态.....48
6.3.18 设置到位精度 48
6.3.19 获取到位精度 48
6.3.20 设置 jog 坐标类型.....48
6.3.21 获取 jog 坐标类型.....49
6.3.22 获取机器人类型 49
6.3.23 获取机器人信息 49

6.3.24 获取机器人在线状态 49
6.3.25 获取控制器状态 50
6.3.26 设置全局速度 50
6.3.27 获取全局速度 50
6.3.28 设置系统模式 50
6.3.29 获取系统模式 51
6.3.30 设置运动模式 51
6.3.31 获取运动模式 51
6.3.32 清除错误 51
6.3.33 清除所有错误报警后获取是否还有错误 52
6.3.34 获取错误号 52
6.3.35 获取错误详情 52
6.3.36 获取错误详情 52
6.3.37 获取关节坐标 53
6.3.38 获取世界坐标 53
6.3.39 运动开始 53
6.3.40 运动暂停 53
6.3.41 运动继续 53
6.3.42 运动停止 54
6.3.43 点动 54
6.3.44 寸动 54
6.3.45 指令内变量说明 55
6.3.46 MovJ 55
6.3.47 MovJ2 56
6.3.48 MovL 56
6.3.49 MovL2 57
6.3.50 MovC 58
6.3.51 MovC2 59
6.3.52 MovArch 59
6.3.53 MovArch2 60
6.3.54 获取当前运行指令号 61
6.3.55 获取当前运动完成状态 61
6.3.56 获取指定运动完成状态 61
6.3.57 获取运动状态 62
6.3.58 设置外部坐标系 62
6.3.59 设置外部坐标系 62
6.3.60 获取工具坐标系 ID 63
6.3.61 新建工具参数 63
6.3.62 修改工具参数 63
6.3.63 选择工具参数 64
6.3.64 获取当前工具 ID 64
6.3.65 获取工具坐标变量名称 64
6.3.66 获取工具参数 65
6.3.67 获取用户坐标系 ID 65

6.3.68 新建用户坐标 66
6.3.69 修改用户坐标 66
6.3.70 选择用户坐标 66
6.3.71 获取当前用户坐标 ID 67
6.3.72 获取用户坐标变量名称 67
6.3.73 获取用户坐标参数 67
6.3.74 获取负载 ID 68
6.3.75 新建负载参数 68
6.3.76 修改负载参数 69
6.3.77 选择负载参数 69
6.3.78 获取当前负载 ID 69
6.3.79 获取负载变量名称 70
6.3.80 获取负载参数 70
6.3.81 获取默认加载的负载参数 70
6.3.82 设置 IO 的值 71
6.3.83 获取 IO 的值 71
6.3.84 设置 DO 的值 72
6.3.85 设置变量 72
6.3.86 获取变量 73
6.3.87 获取整形变量 73
6.3.88 设置整形变量 73
6.3.89 获取实数变量 74
6.3.90 设置实数变量 74
6.3.91 获取字符变量 75
6.3.92 设置字符变量 75
6.3.93 获取位置变量 76
6.3.94 设置位置变量 76
6.3.95 示教点 77
6.3.96 获取当前程序 77
6.3.97 获取程序状态 77
6.3.98 卸载工程 78
6.3.99 卸载程序 78
6.3.100 加载工程 78
6.3.101 加载程序 78
6.3.102 运行程序 79
6.3.103 暂停程序 79
6.3.104 停止程序 79
6.3.105 执行 MovJ2 数组指令 80
6.3.106 执行 MovL2 指令数组 80
6.3.107 获取运动数组执行状态 81
6.3.108 获取运动数组执行状态 81
6.3.109 笛卡尔坐标转换关节坐标 82
6.3.110 关节坐标转换笛卡尔坐标 82
6.3.111 获得机器人当前线速度 82

6.3.112 设置点位数组.....82
6.3.113 获取点位数组.....83
6.3.114 获取力矩的值.....83
6.3.115 设置 PSO 控制字.....84
6.3.116 获取 PSO 状态字的值.....84
6.3.117 设置指定轴的指定 PSO 比较点指令.....84
6.3.118 MovLOffset.....85
6.3.119 MovJOffset.....85
附录一：错误列表.....87

前言

本手册适用于控制系统 RCS2 系列，介绍埃斯顿 ER 系列机器人 API 功能使用方法。

读者对象

本手册仅供埃斯顿机器人相关技术支持人员使用。

注意事项

- 在安装和调试这些组件时，操作人员必须严格遵循本文档的说明和解释。
- 相关负责人员必须确保所述产品的应用或使用满足所有安全要求，包括相关法律、法规、准则和标准。
- 尽管本文档经过精心编制，但由于其中所描述的产品仍处于不断更新换代中，我们可能不会在每次更新后都检查文档中所描述的产品性能数据、标准或其它特性总是与实际产品相一致。
- 本文档中难免会出现一些技术或者编辑错误，我们保留随时对文档信息做出修改之权力，恕不另行通知。对于已经变更的产品，如果本文档中的数据、图表以及文字描述没有修改，我们将不再特别加以声明。
- 任何人不得对软、硬件配置进行文本档中规定之外的修改，ESTUN 公司对因此而造成的一切后果不承担任何责任。
- 本文档中出现图示单位在没有特别标注说明时，默认单位为毫米 mm。

安全说明

⚠ 警告	受伤的危险 不遵守本标志相关的安全说明将危及个人生命和健康安全。
⚠ 注意	对环境和设备有危险 不遵守本标志相关安全说明可能明显危害环境和设备安全。

说明	说明或提示 该标志表示这些信息能够帮助您更好的理解安全说明。

第 1 章 库功能介绍

1.1 本节概要

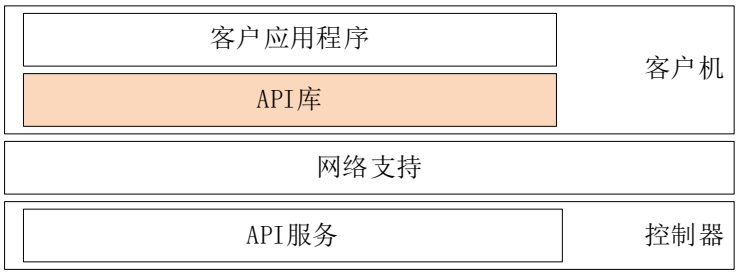
本节主要介绍功能库的基本框架和主要功能。

1.2 项目背景

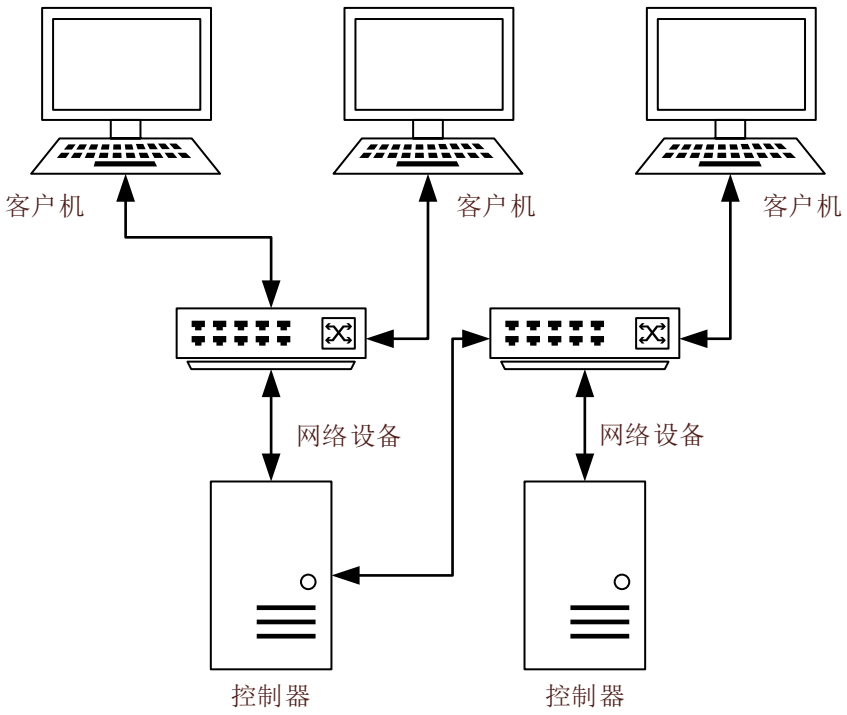
为充分满足客户对机器人运动控制的定制化需求，以及更好的与客户自身控制逻辑或产品相结合，埃斯顿机器人提供了一套完整的接口调用库（以下简称 API 库）供客户进行二次开发来实现对机器人的控制。API 库提供了多种开发语言支持，包括 C++，C#和 VB，适应不同客户的开发需求。

1.3 设计说明

API 库工作于客户上位机，本身不具备单独运行能力，仅提供用户二次开发以及运行依赖支持；用户基于 API 库完成自己的应用程序开发后，运行时会依赖 API 库对象，通过 API 库提供的接口，完成与运行在控制器上的服务端的数据交互，达到驱动机器人的目的。



在功能实现上，API 库提供了接口操作类，通过创建类实例与控制器服务连接，客户程序可以同时创建多个类实例来与多台控制器连接；控制器服务可以同时接受多个客户端连接。这样设计提供了客户程序与机器人的多对多的连接关系，即支持多个客户机程序同时与一台机器人控制器连接，也支持一台客户机程序同时连接多台机器人控制器。



1.4 功能列表

API 库提供了如下的功能接口，用于满足客户对机器人的运动控制以及状态监视。

序号	功能分类	功能名称	详细描述	备注
1	基本功能	日志打印开关	是否输出本地日志	
2		写本地日志	将日志信息写入本地文件	
3		错误日志	获取最近一次错误的信息	
4		连接服务器	连接服务器	
5		断开连接	断开与服务的连接	
6		连接检测	检测是否已连接服务	
7	系统控制	申请权限	申请机器人运动操作权限	
8		查询权限	查询当前机器人运动操作权限	
9		释放权限	释放机器人运动操作权限	
10		使能开关	给机器人上/下励磁	
11		获取使能状态	获取机器人是否上励磁	

12		jog使能开关	设置当前模式下job是否可用	
13		获取jog使能状态	获取当前模式下job是否可用	
14		设置到位精度	设置默认到位精度值，掉电保存	
15		获取到位精度	获取默认到位精度值	
16		设置jog坐标类型	设置点动坐标系类型，轴关节坐标或世界坐标	
17		获取jog坐标类型	获取点动坐标系类型，轴关节坐标或世界坐标	
18		获取机器人类型	获取机器人型号信息	
19		获取机器人信息	获取机器人信息，一些附加备注信息	
20		获取机器人在线状态	获取机器人状态，目前仅有连接状态	
21		获取控制器状态	获取控制器当前状态，有正常，错误和停止三种状态	
22		设置全局速度	设置机器人运行的速度百分比	
23		获取全局速度	获取机器人运行的速度百分比	
24		获取系统模式	获取机器人当前的工作模式	
25		设置运动模式	设置机器人当前的运动模式，单步或连续	
26		获取运动模式	获取机器人当前的运动模式	
27		清除错误	清除下位机报告的机器人错误警告信息	
28		获取错误号	获取错误警告编号	
29		获取错误详情	根据错误号，获取错误详细内容，支持中英文	

30		获取关节坐标	获取机器人当前位置信息，关节坐标系	
31		获取世界坐标	获取机器人当前位置信息，世界坐标系	
32	运动控制	开始	恢复机器人运动，从停止模式转到运行模式	
33		暂停	暂停机器人运动，不会清空运动队列	
34		继续	恢复机器人运动，继续执行运动队列	
35		停止	停止机器人运动，同时清空运动队列	
36		点动	使机器人按照轴关节或空间坐标进行运动，一次运动微量距离	
37		寸动	使机器人按照轴关节或空间坐标方式运动，一次运动指定距离	
38		MovJ	关节运动指令，参数为机器人内部变量名称	
39		MovJ2	关节运动指令，参数为对应类型的对象	
40		MovL	直线运动指令，参数为机器人内部变量名称	
41		MovL2	直线运动指令，参数为对应类型的对象	
42		MovC	圆弧运动指令，参数为机器人内部变量名称	
43		MovC2	圆弧运动指令，参数为对应类型的对象	

44		MovArch	拱形运动指令，参数为机器人内部变量名称	
45		MovArch2	拱形运动指令，参数为对应类型的对象	
46		获取运行指令号	获取最近一次下发成功的运动指令的信息	
47		获取指令状态	获取最近一次下发成功的运动指令的执行状态	
48		获取指令状态	获取指定运动指令执行状态	
49		获取运动状态	获取机器人当前的运动状态：运动/停止	
50	参数设置	设置外部坐标系	根据外部坐标系变量名设置系统当前使用的外部坐标系	
51		设置外部坐标系	根据外部坐标系ID设置系统当前使用的外部坐标系	
52		新建工具参数	在系统内创建指定标号的工具参数	
53		修改工具参数	修改指定标号的工具参数	
54		选择工具参数	设置系统当前使用的工具参数	
55		获取当前工具号	获取当前加载的工具号	
56		获取工具参数	根据工具号获取工具的参数值	
57		新建用户坐标	在系统内创建指定标号的用户坐标	
58		修改用户坐标	修改指定标号的用户坐标	
59		选择用户坐标	设置系统当前使用的用户坐标	
60		获取当前坐标号	获取当前加载的用户坐标号	
61		获取坐标参数	根据坐标号获取坐标的参数值	
62		新建负载参数	在系统内创建指定标号的负载参数	

63		修改负载参数	修改指定标号的负载参数	
64		选择负载参数	设置系统当前使用的负载参数	
65		获取当前负载号	获取当前加载的负载号	
66		获取负载参数	根据负载号获取负载的参数值	
67	IO设置	设置IO的值	设置系统IO的值，包括DI,DO,AI,AO,SimDI,SimDO,SimAI,SimAO	
68		获取IO的值	获取IO的值，包括DI,DO,AI,AO,SimDI,SimDO,SimAI,SimAO	
69		设置DO的值	修改系统DO的值	
70	变量设置	设置变量	设置或增加变量，支持数字，字符，位置等常规类型	
71		获取变量	获取指定变量的值	
72		获取整形变量	获取指定变量的值	
73		设置整形变量	设置或增加变量	
74		获取实数变量	获取指定变量的值	
75		设置实数变量	设置或增加变量	
76		获取字符变量	获取指定变量的值	
77		设置字符变量	设置或增加变量	
78		获取位置变量	获取指定变量的值	
79		设置位置变量	设置或增加变量	
80		示教点	将当前机器人坐标值赋值给位置变量	
81	程序运行	获取当前程序	获取系统当前加载的程序信息，包括工程名、程序名	
82		获取程序状态	获取当前程序运行状态，运行、暂	

			停或停止	
83		卸载工程	卸载工程，连带程序一起卸载，系统恢复到未加载程序的状态	
84		卸载程序	卸载程序，工程依旧加载	
85		加载工程	只加载工程，不会加载工程内程序	
86		加载程序	加载工程和程序	
87		运行程序	机器人执行加载的程序	
88		暂停程序	机器人暂停执行加载的程序	
89		停止程序	机器人停止执行加载的程序	

1.5 模式权限说明

API库在所有模式下都可以调用，但出于安全考虑，在不同模式下，api的功能会由不同限制，具体参考下表：

主控权限

功能名		功能描述	API 模式	非 API 模式		
				远程	自动	手动
			API 接口支持			
工程操作接口		操作用户工程的接口，加载、注销、工程启动、工程暂停、工程停止等	X	X	√	√
变量操作	全局变量	全局变量的操作接口，读、写、创建、删除等操作	√	√	√	√
	工程变量、程序变量	工程、程序变量的操作接口，读、写、创建、删除等操作	X	X	√	√
运动指令		运动指令 MovJ\MovL 等接口函数，以及运动中设置精度，不能限制速度，运动暂停等功能	√	X	X	X

Jog 指令		Jog 机器人的接口（设置 Jog 坐标系，点动、寸动等 jog 机器人用到的接口)	√	X	X	√
机器人状态		机器人速度、使能、坐标系、当前系统状态、当前错误等等机器人的各种状态（好的话能都开放）	√	√	√	√
IO 指令	实际 IO	实际 IO 读写操作接口	√	√	√	√
	虚拟 IO	虚拟 IO 读写操作接口	√	√	√	√
注意：在停止状态，非 API 模式下，才可以切换到 API 模式						

监控权限

功能名		功能描述	API 模式	非 API 模式		
				远程	自动	手动
			API 接口支持			
工程操作接口		操作用户工程的接口，加载、注销、工程启动、工程暂停、工程停止等	X	X	√	√
变量操作	全局变量	全局变量的操作接口，读、写、创建、删除等操作	√	√	√	√
	工程变量、程序变量	工程、程序变量的操作接口，读、写、创建、删除等操作	X	X	√	√
运动指令		运动指令 MovJ\MovL 等接口函数，以及运动中设置精度，不能限制速度，运动暂停等功能	X	X	X	X

Jog 指令		Jog 机器人的接口（设置 Jog 坐标系，点动、寸动等 jog 机器人用到的接口)	√	X	X	√
机器人状态		机器人速度、使能、坐标系、当前系统状态、当前错误等等机器人的各种状态（好的话能都开放）	√	√	√	√
IO 指令	实际 IO	实际 IO 读写操作接口	√	√	√	√
	虚拟 IO	虚拟 IO 读写操作接口	√	√	√	√
注意：在停止状态，非 API 模式下，才可以切换到 API 模式						

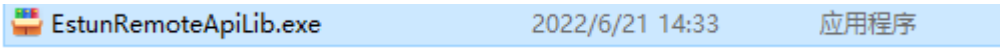
第 2 章 安装

2.1 本节概要

本节介绍了 API 功能库的安装步骤，以及文件目录结构。

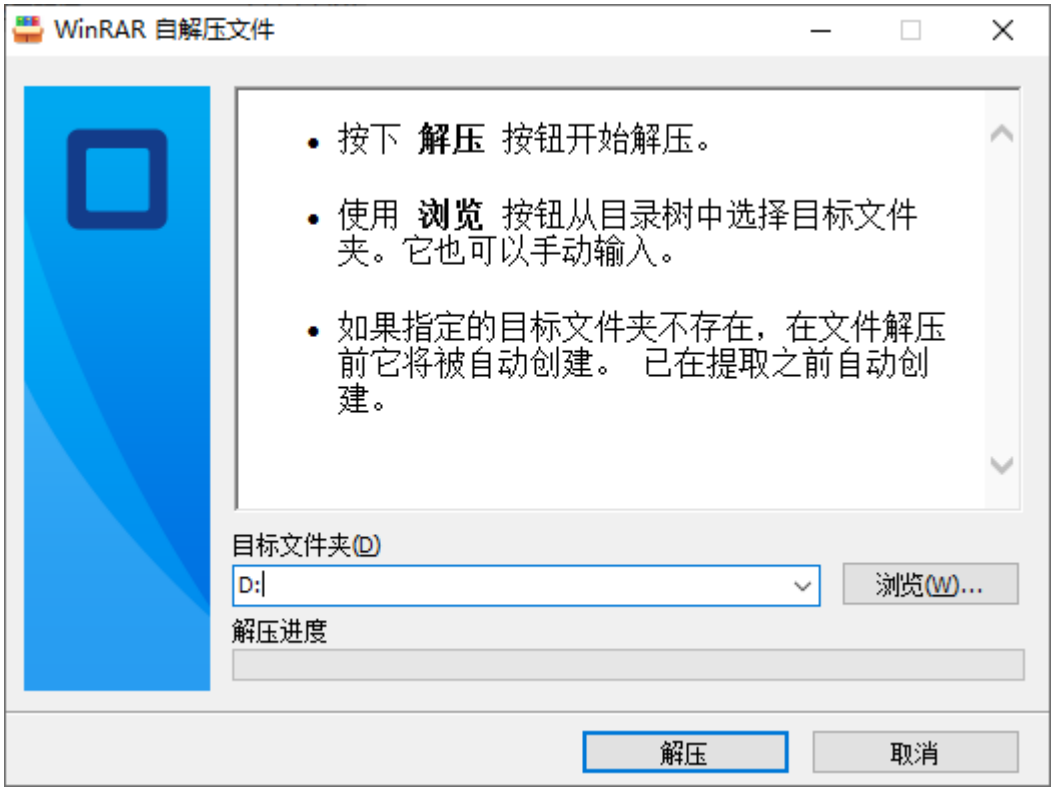
2.2 安装说明

2.2.1 解压库文件

执行 

，

按自己的需求选择库的安装目录，这里直接放到 D 盘根目录，然后点击解压



等待解压完成。

2.2.2 配置环境变量

进入库的安装目录，进入 EstunRemoteApiLib 目录，根据自己的需求选择 32 位或 64 位的库目录（32 位程序对应 x32 目录，64 位程序对应 x64 目录）



进入对应目录，执行 path.bat，完成环境变量的配置。

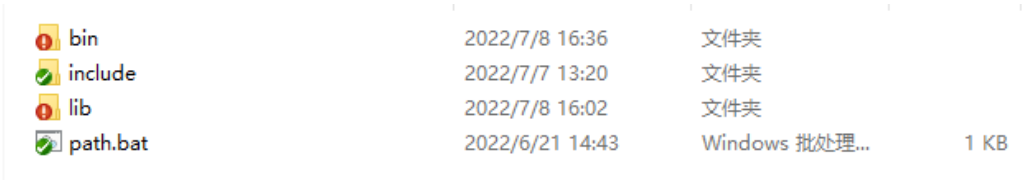
2.3 目录结构

安装完成后，在指定目录会生成三个目录，如下图所示，



文件名	用途	备注
help	代码示例工程	一个 vs 解决方案，包含 c++，c#和 vb 工程
x32	32 位 API 库	包含头文、lib 库、运行依赖库以及一个测试程序
x64	64 位 API 库	同上

库文件目录详细说明，x32 和 x64 结构一致，以 x32 举例



文件名	用途
bin/EstunApiFroPublic-vc140-x32.dll	.Net 项目引用库
bin/EstunRemoteApiLib-vc140-x32.dll	C++/.Net 程序运行依赖库
Include/EstunRemoteApiLib	C++项目需要包含的头文件
lib/EstunRemoteApiLib-vc140-x32.lib	C++项目的附加依赖库

第 3 章 使用操作

3.1 本节概要

本节介绍了如何基于API库创建自己的工程，来实现对机器人的控制。

3.2 C++工程搭建

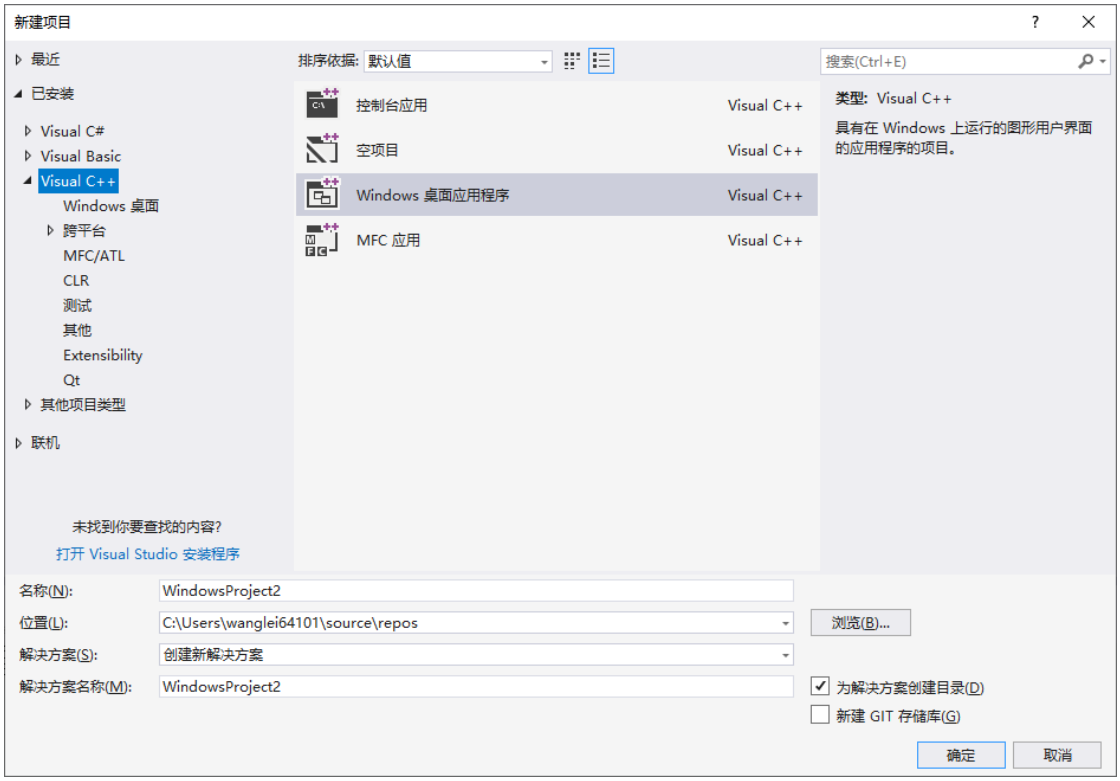
3.2.1 环境要求

配置项	最低要求	推荐配置	备注
系统版本	Win7	Win10	
内存大小	4GB	8GB	
VisualStudio 版本	2015	2017	
QT 版本	4.8.5	5.11	开发基于 QtGui 的程序时需要

3.2.2 搭建步骤

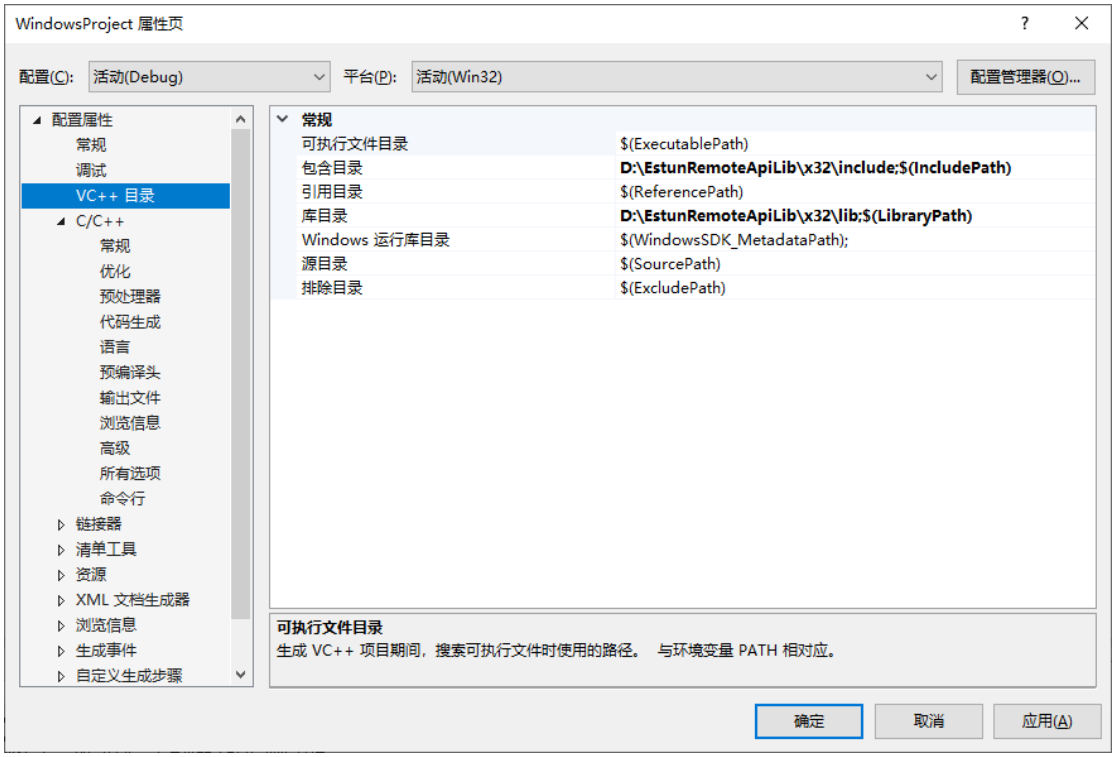
步骤 1 新建工程

打开 vs2015，点击左上角【文件】-【新建】-【项目】，选择【Visual C++】-【Windows 桌面应用工程】



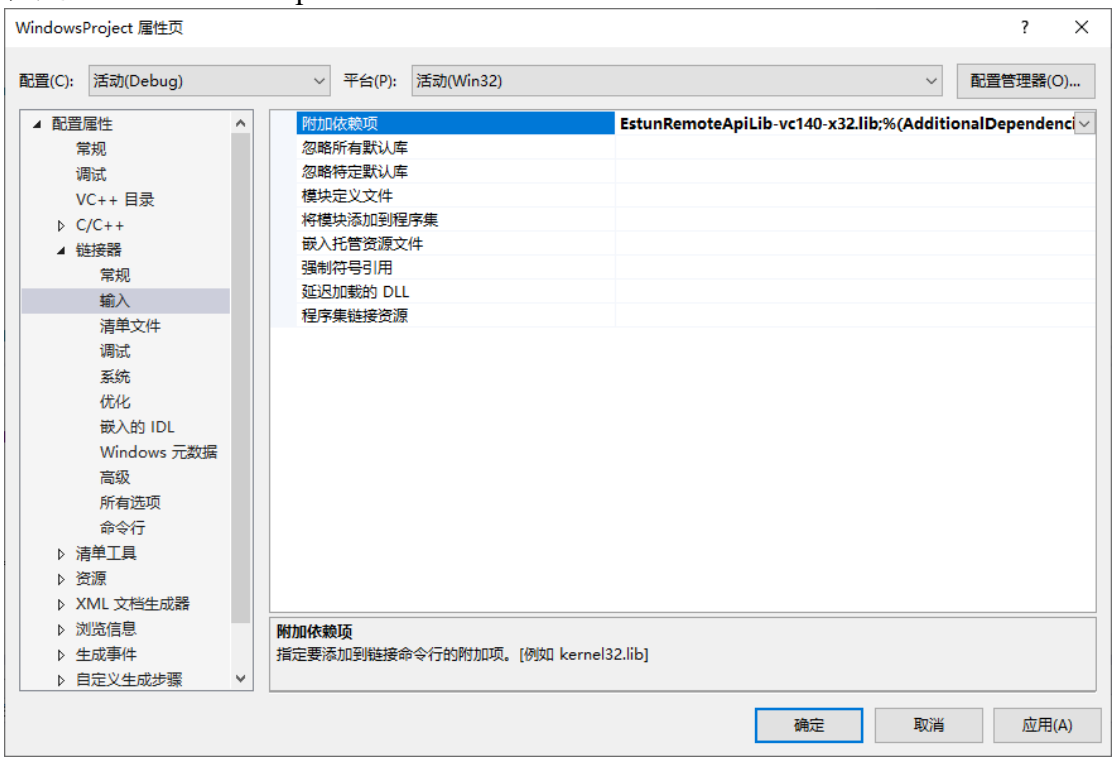
步骤 2　　配置头文件路径和库路径

右键工程，点击【属性】，点击【配置属性】-【VC++目录】，在【包含目录】中添加 API 库的 include 目录，在【库目录】中添加 API 库的 lib 目录



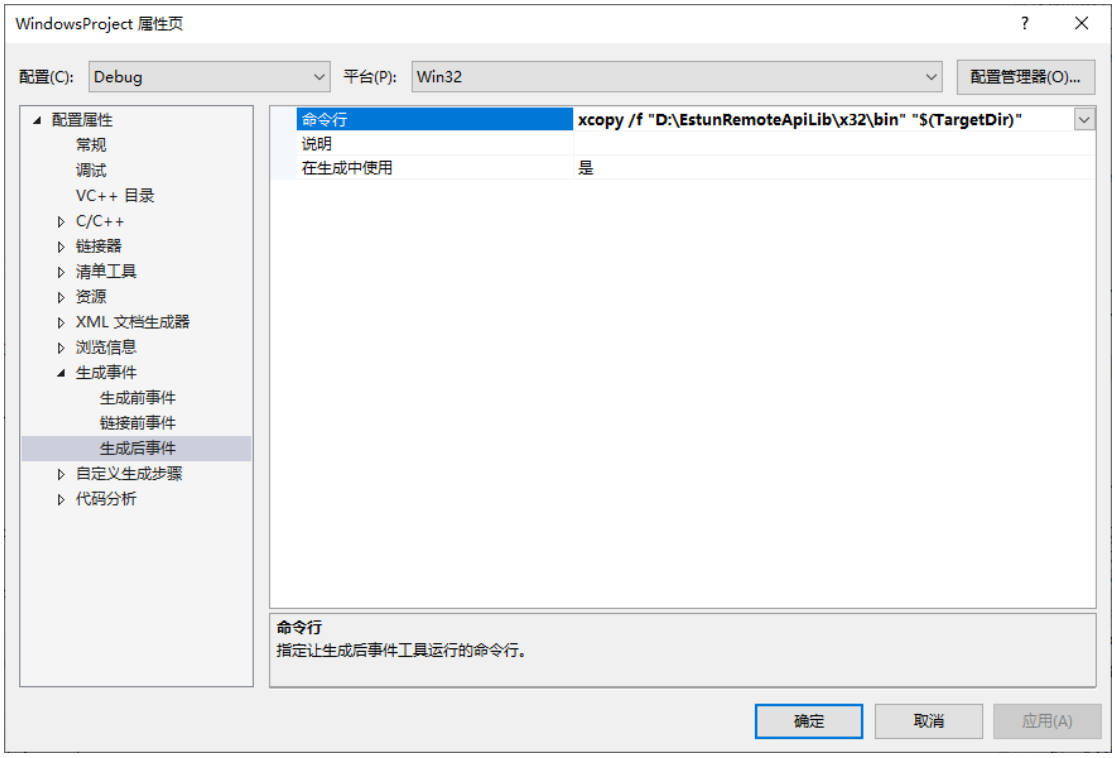
步骤 3 配置外部依赖库

右键工程，点击【属性】，点击【配置属性】-【链接器】-【输入】，在【附加依赖项】中添加 API 库的 lib，32 位是“EstunRemoteApiLib-vc140-x32.lib”，64 位是“EstunRemoteApiLib-vc140-x64.lib”



步骤 4 配置生成事件

右键工程，点击【属性】，点击【配置属性】-【生成事件】-【后期生成事件】，在【命令行】中添加 xcopy /f "D:\EstunRemoteApiLib\x32\bin" "$(TargetDir)"，（D:\EstunRemoteApiLib 替换为自己安装 API 库的目录）



步骤 5　添加程序逻辑，调用 API 接口

包含头文件

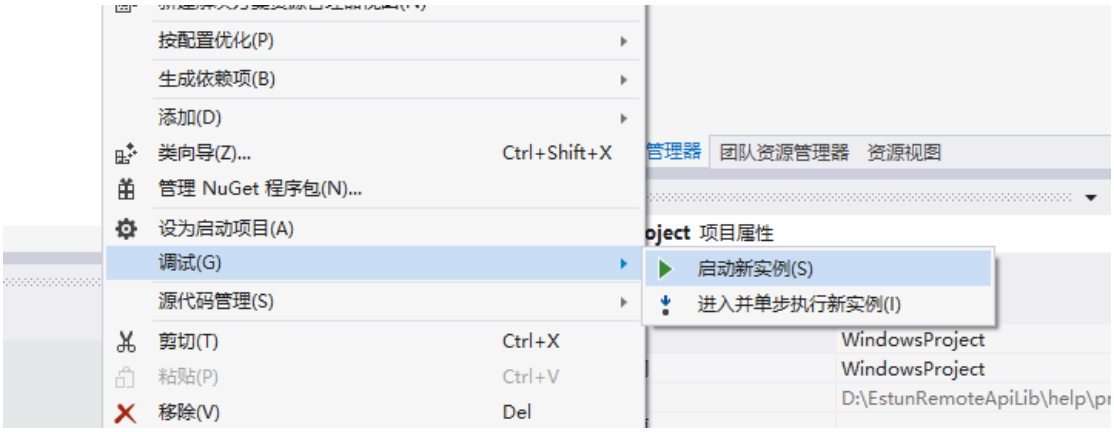
```
#include <EstunRemoteApiLib/EstunRemoteApiLib.h>
```

实现调用逻辑

```
//创建API功能库对象
EstunRemoteApiLib *api = new EstunRemoteApiLib;

//连接机器人服务器
api->connectToServer("127.0.0.1");

//申请运动控制权限
E_ROB_PERMIT permit = api->E_AcquirePermit();

//机器人上励磁
api->E_SetMot(true);

//执行MovJ
E_ROB_JOBID jobId = api->E_MovJ("t_g.P0");

//查询当前机器人位置坐标
E_ROB_POS pos = api->E_GetCurJPos();

//释放权限
api->E_ReleasePermit(permit);

//断开连接
api->disConnect();

//释放API功能库对象
delete api;
```

步骤 6　编译，运行

右键项目，启动实例，如果提示找不到 xxx.dll，参考 2.2.2



3.1 C#工程搭建

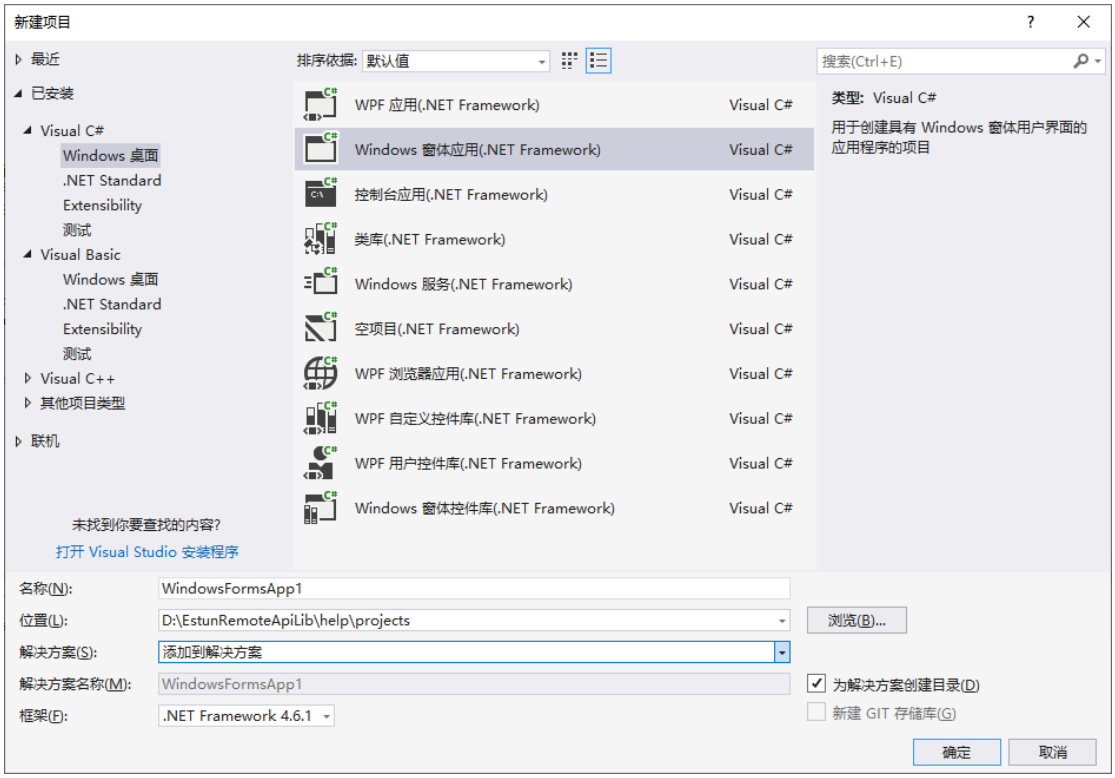
3.1.1 环境要求

配置项	最低要求	推荐配置	备注
系统版本	Win7	Win10	
内存大小	4GB	8GB	
VisualStudio 版本	2015	2017	
.Net 版本	4.0	4.6	

3.1.2 搭建步骤

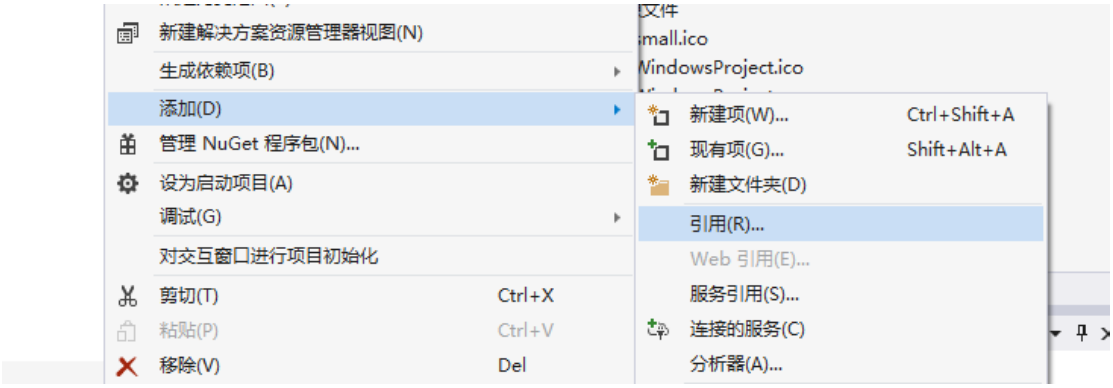
步骤 1 **新建工程**

打开 vs2015，点击左上角【文件】-【新建】-【项目】，选择【Visual C#】-【Windows 窗体应用】

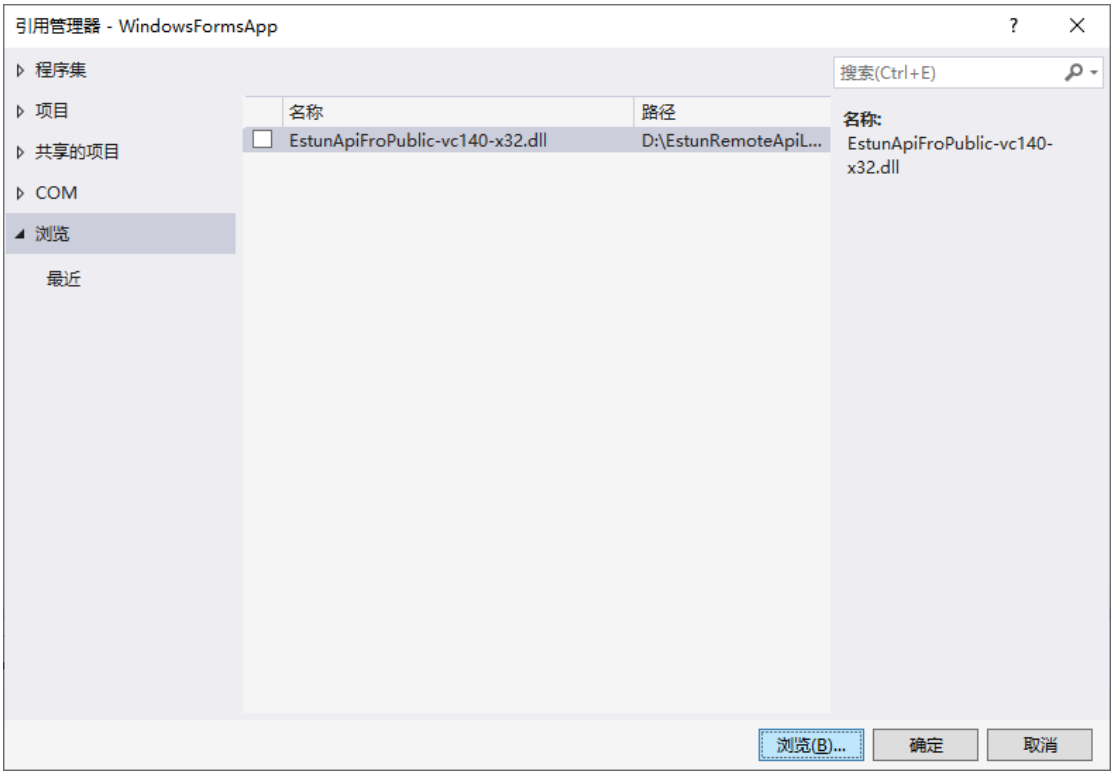


步骤 2 **添加引用**

右键工程，点击【添加】-【引用】

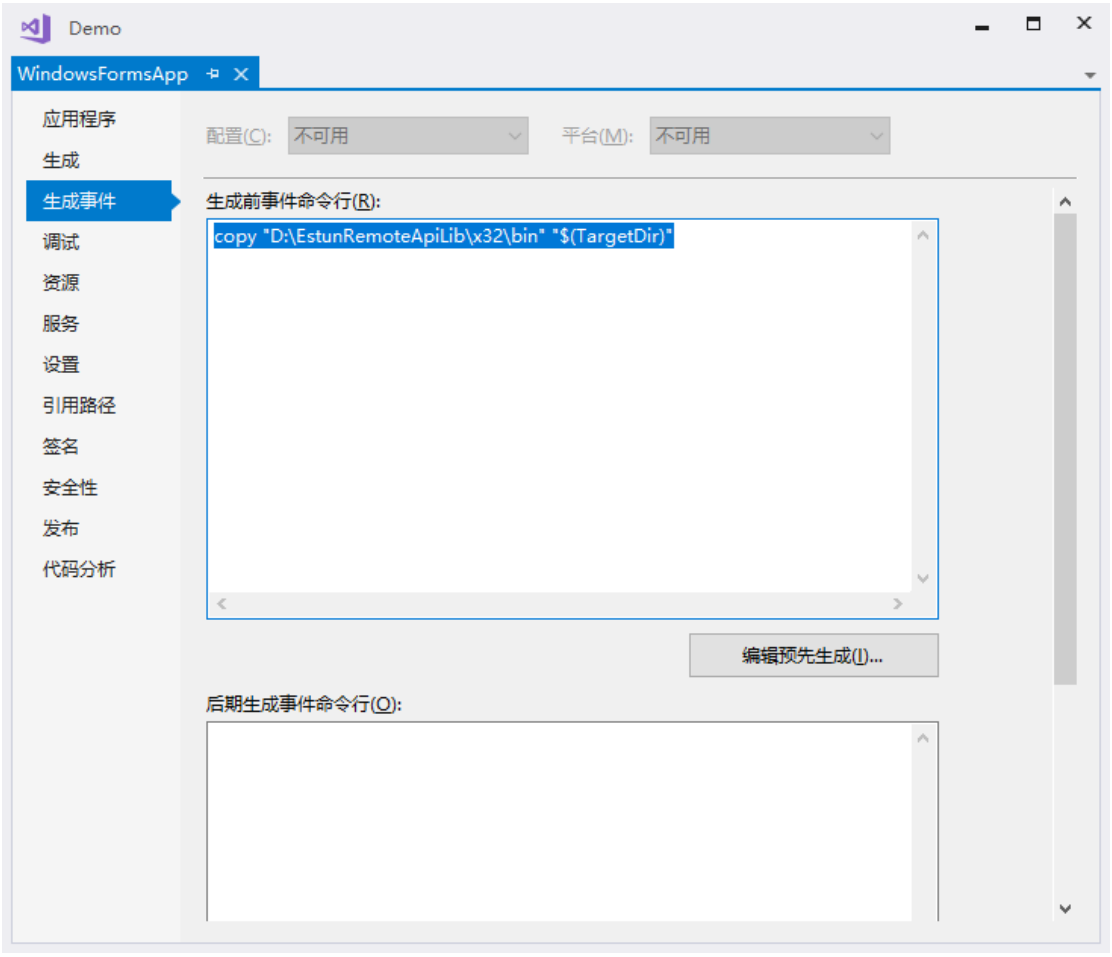


点击右下角【浏览】，添加 API 库的依赖；此处以 32 位项目举例，添加 EstunRemoteApiLib\x32\bin\ EstunApiFroPublic-vc140-x32.dll，点击【确定】



步骤 3　配置生成事件

右键工程，点击【属性】，点击【生成事件】，在【命令行】中添加 copy "D:\EstunRemoteApiLib\x32\bin" "$(TargetDir)"，（D:\EstunRemoteApiLib 替换为自己安装 API 库的目录）



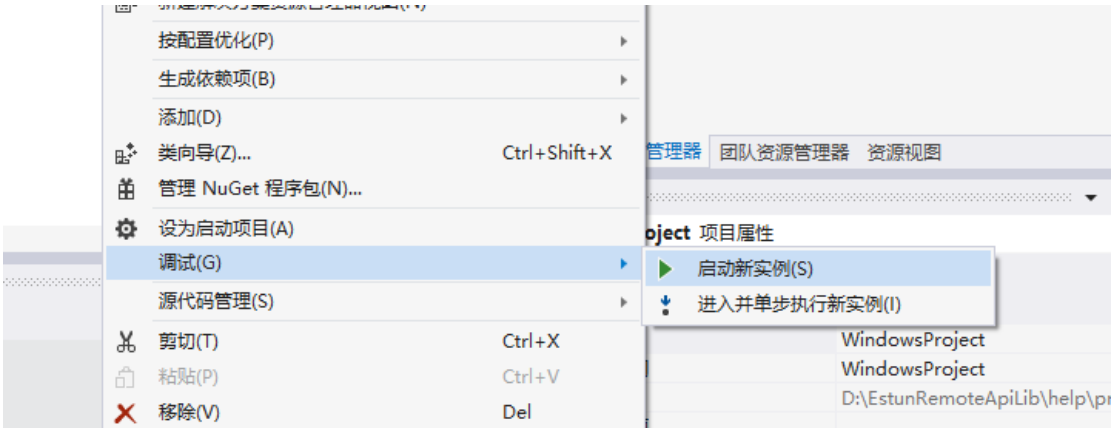
步骤 4　　添加程序逻辑，调用 API 接口

实现调用逻辑

```
//创建API功能库对象
EstunApiFroPublic.EstunApi_CLI api = new EstunApiFroPublic.EstunApi_CLI();

//连接机器人服务器
api.connectToServer("127.0.0.1", true);

//申请运动控制权限
EstunApiStruct_CLI.E_ROB_PERMIT_CLI permit = api.E_AcquirePermit();

//机器人上励磁
api.E_SetMot(true);

//执行MovJ
EstunApiStruct_CLI.E_ROB_JOBID_CLI jobId = api.E_MovJ("t_g.P0");

//查询当前机器人位置坐标
EstunApiStruct_CLI.E_ROB_POS_CLI pos = api.E_GetCurJPos();

//释放权限
api.E_ReleasePermit(permit);

//断开连接
api.disConnect();

//释放API功能库对象
api.Dispose();
```

步骤 5　　编译，运行

右键项目，启动实例，如果提示找不到 xxx.dll，参考 2.2.2



3.2 VB 工程搭建

3.2.1 环境要求

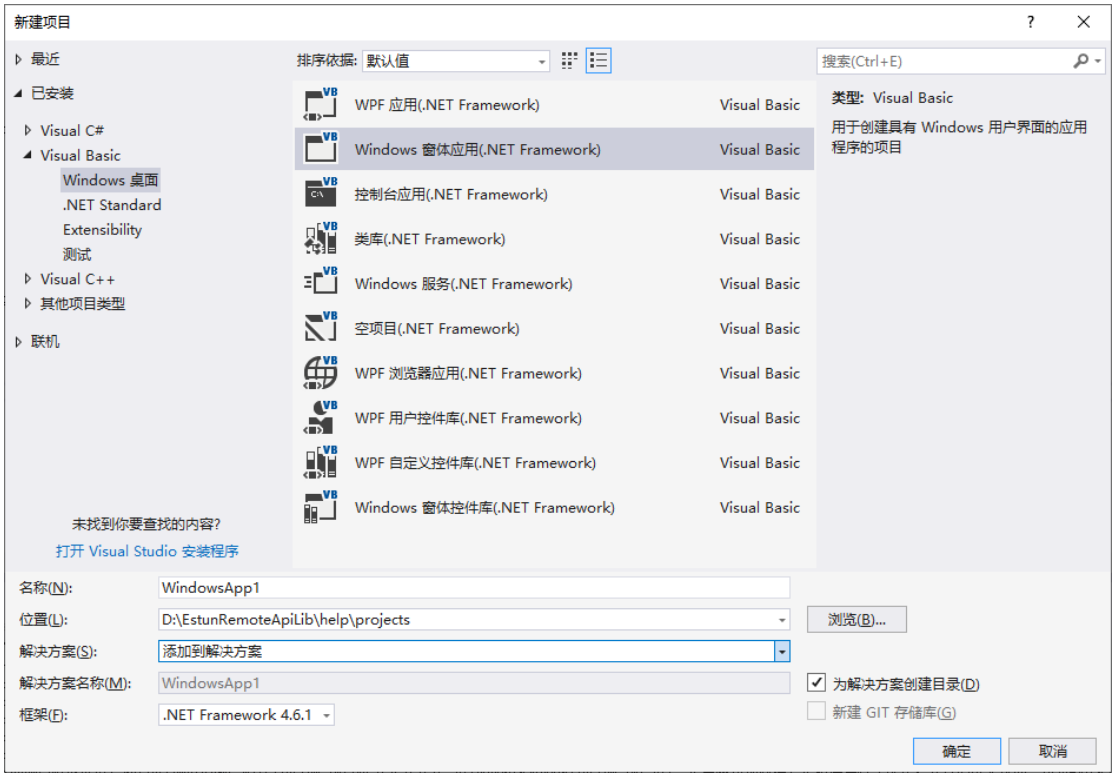
配置项	最低要求	推荐配置	备注
系统版本	Win7	Win10	

内存大小	4GB	8GB	
VisualStudio 版本	2015	2017	
.Net 版本	4.0	4.6	

3.2.2 搭建步骤

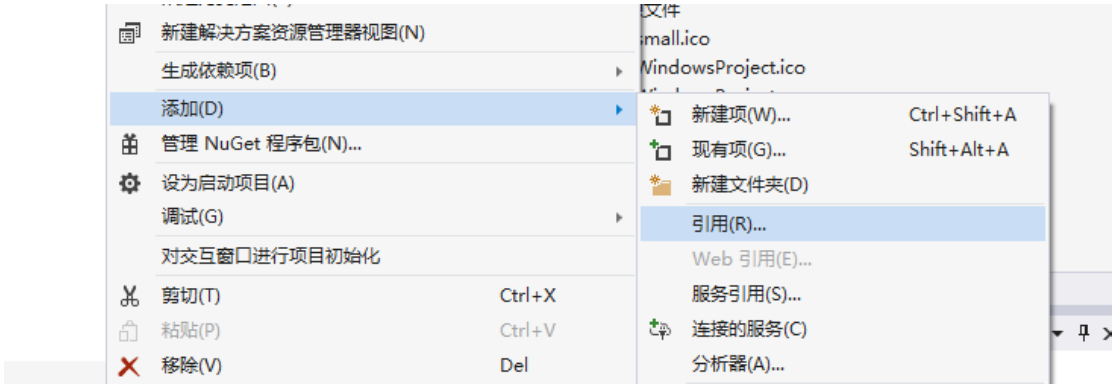
步骤 1　　新建工程

打开 vs2015，点击左上角【文件】-【新建】-【项目】，选择【Visual Basic】-【Windows 窗体应用】

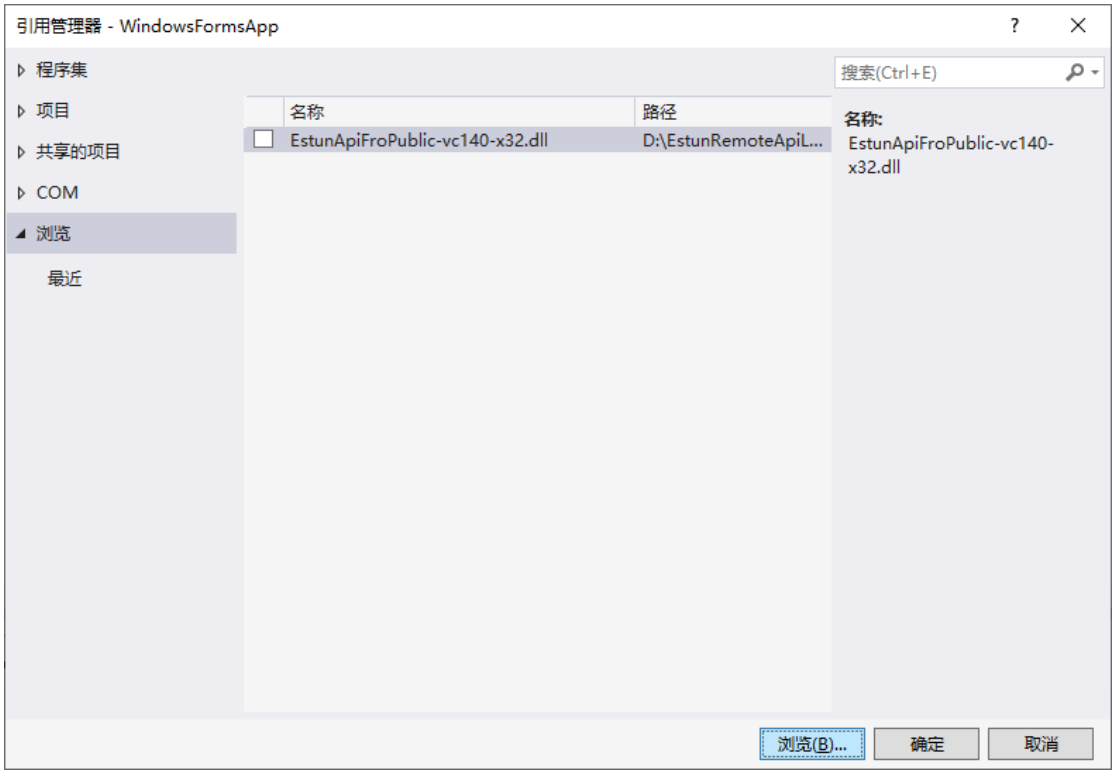


步骤 2　　添加引用

右键工程，点击【添加】-【引用】

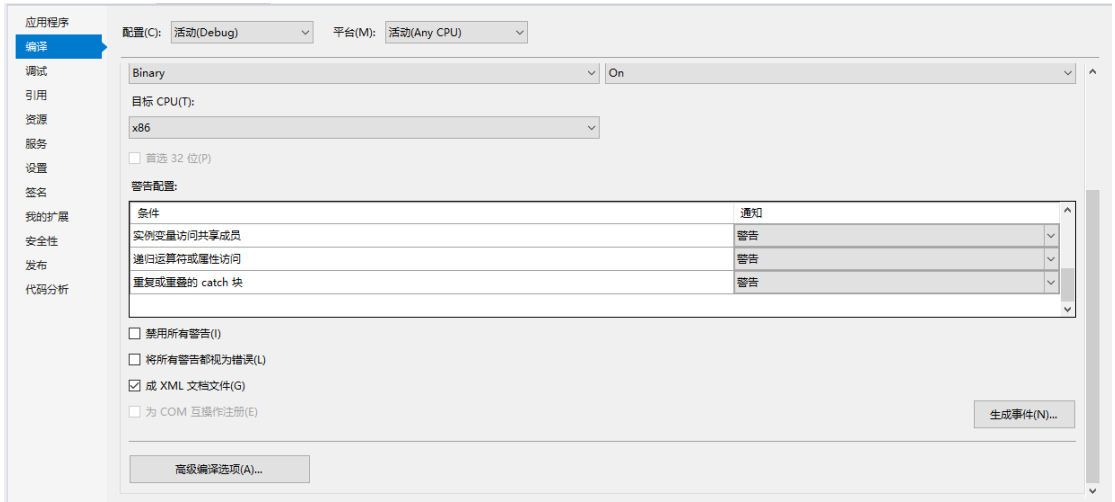


点击右下角【浏览】，添加 API 库的依赖；此处以 32 位项目举例，添加 EstunRemoteApiLib\x32\bin\ EstunApiFroPublic-vc140-x32.dll，点击【确定】

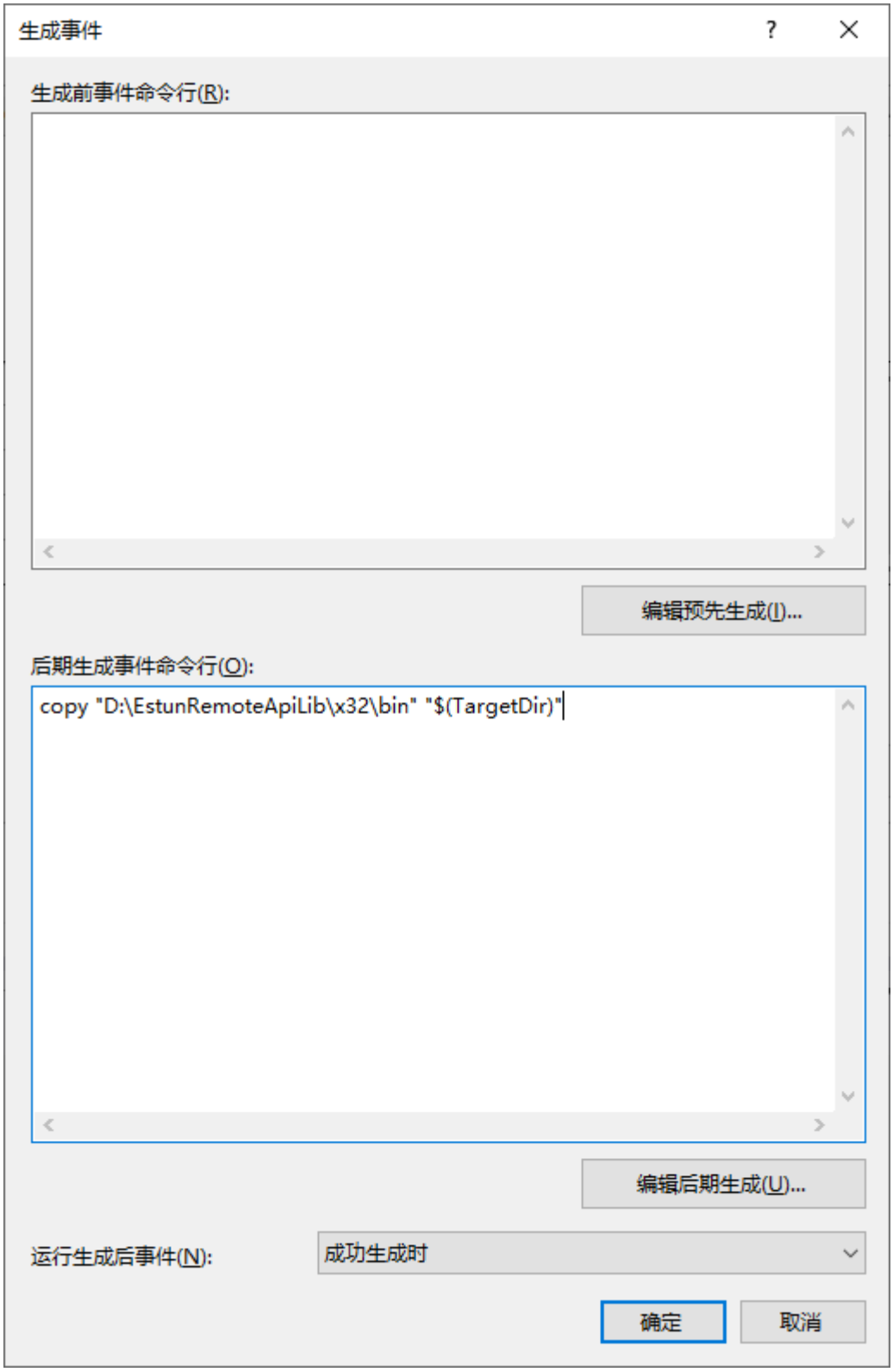


步骤 3　　配置生成事件

右键工程，点击【属性】，点击【编译】选项卡，点击右下角【生成事件】



在后期生成事件命令行中添加 copy "D:\EstunRemoteApiLib\x32\bin" "$(TargetDir)"，（D:\EstunRemoteApiLib 替换为自己安装 API 库的目录）



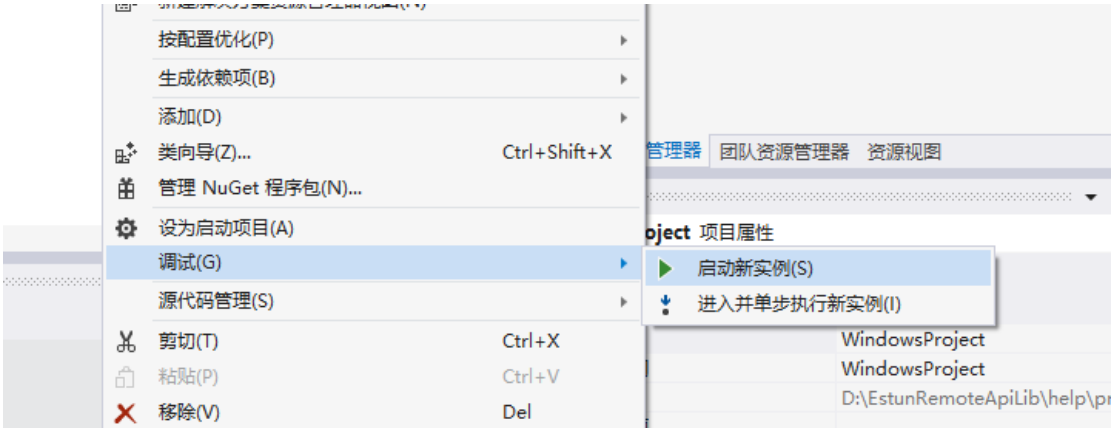
步骤 4　　添加程序逻辑，调用 API 接口

实现调用逻辑

```
'创建API功能库对象
Dim api As New EstunApiFroPublic.EstunApi_CLI
'连接机器人服务器
api.connectToServer("127.0.0.1", True)

'申请运动控制权限
Dim permit = api.E_AcquirePermit()

'机器人上励磁
api.E_SetMot(True)

'执行MovJ
Dim jobId = api.E_MovJ("t_g.P0")

'查询当前机器人位置坐标
Dim pos = api.E_GetCurJPos()

'释放权限
api.E_ReleasePermit(permit)

'断开连接
api.disConnect()

'释放API功能库对象
api.Dispose()
```

步骤 5　　编译，运行

右键项目，启动实例，如果提示找不到 xxx.dll，参考 2.2.2



第 4 章 运行环境

4.1 本节概要

本节主要介绍基于 API 库上开发的上位机程序如何与实际机器人连接使用。

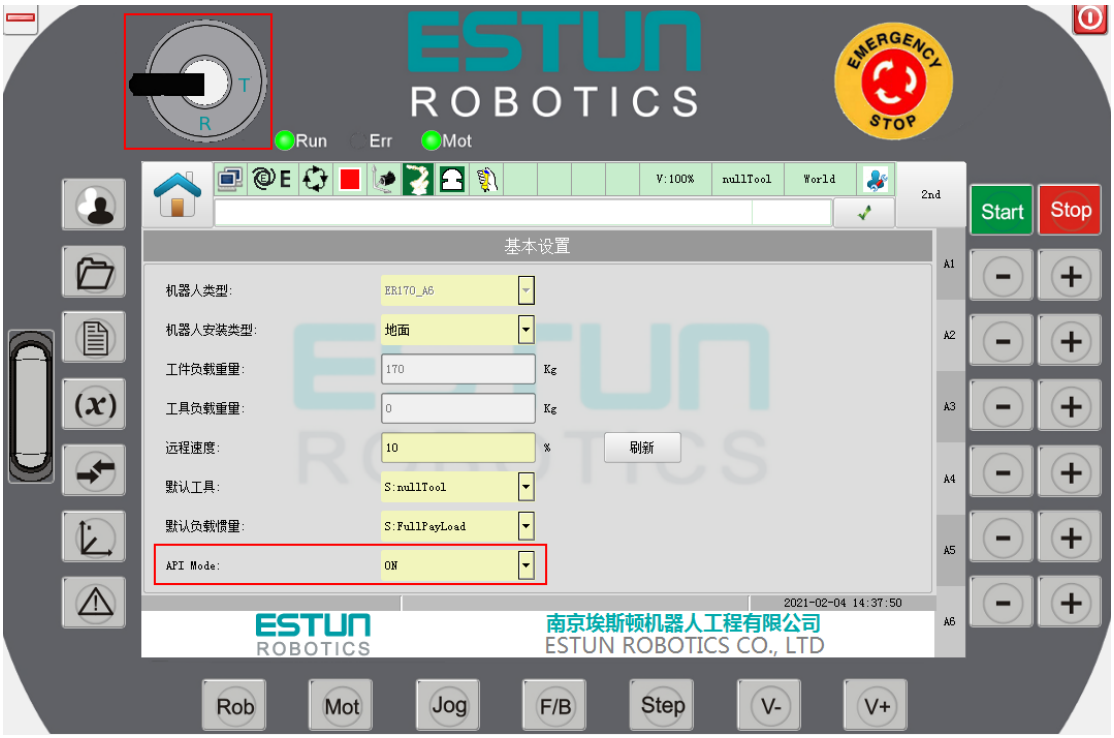
4.2 运行环境

4.2.1 软件支持

控制系统需 RCS2 V1.28.00 及以上版本，在目标控制器上运行 APIServer.exe 程序。

4.2.2 示教器端配置

使用 API 功能控制机器人运动时，需将 API Mode 使能，同时将示教器上的模式开关切换至远程模式。

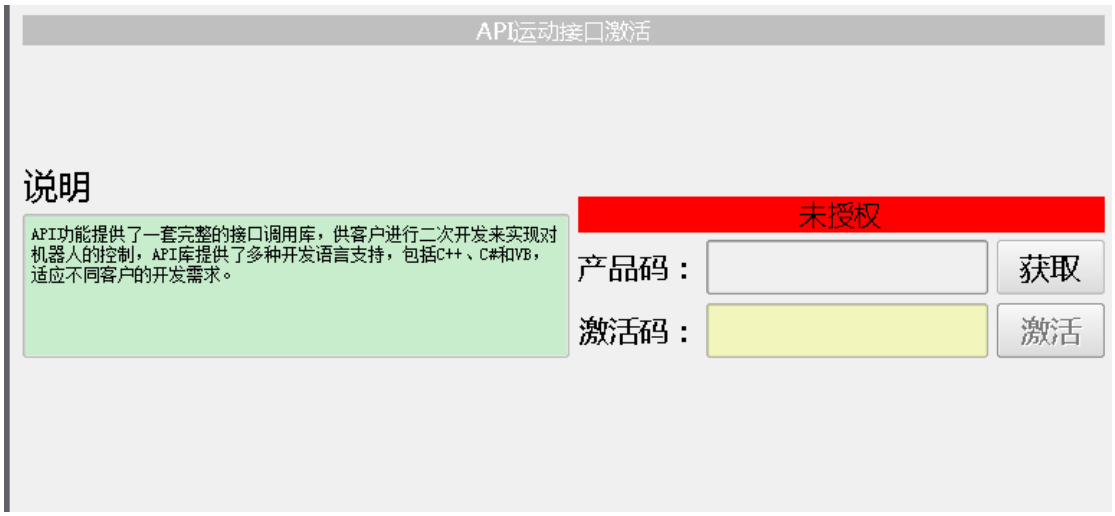


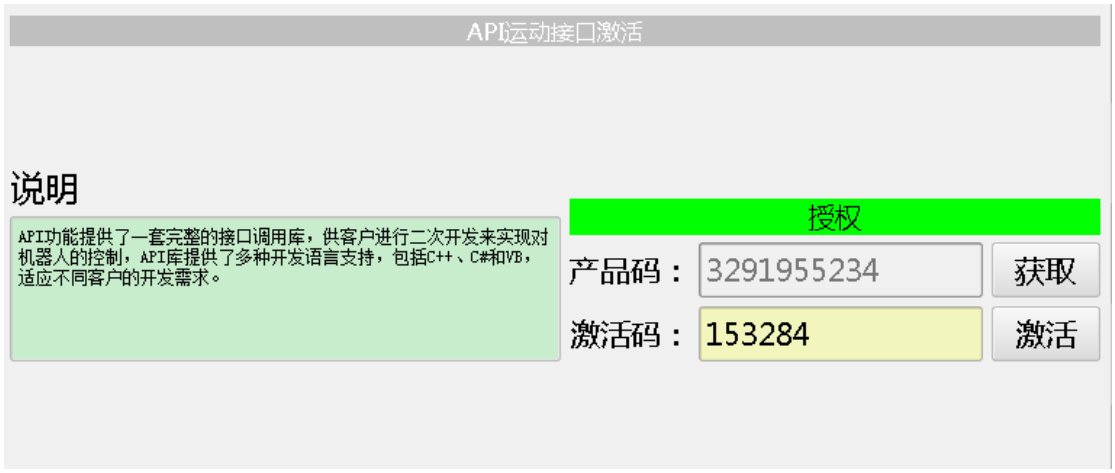
如需使用 API 功能的运动接口，需要 API 功能进行授权激活。可通过以下步骤进行激活：

1）在管理员权限下进入用户设置的 API 激活界面，通过点击界面中的获取按钮获取产品码，将此产品码发送给厂家，由厂家处获取激活码（与产品码一一对应）；（如果进入界面时，激活状态为已激活，则不需要此激

活配置过程）

2）通过将由厂家获取的激活码输入到激活码编辑框中，点击激活按钮，如果输入的激活码正确，则状态变为授权；如果输入的激活码不正确，则状态仍为未授权。





第 5 章 典型应用案例

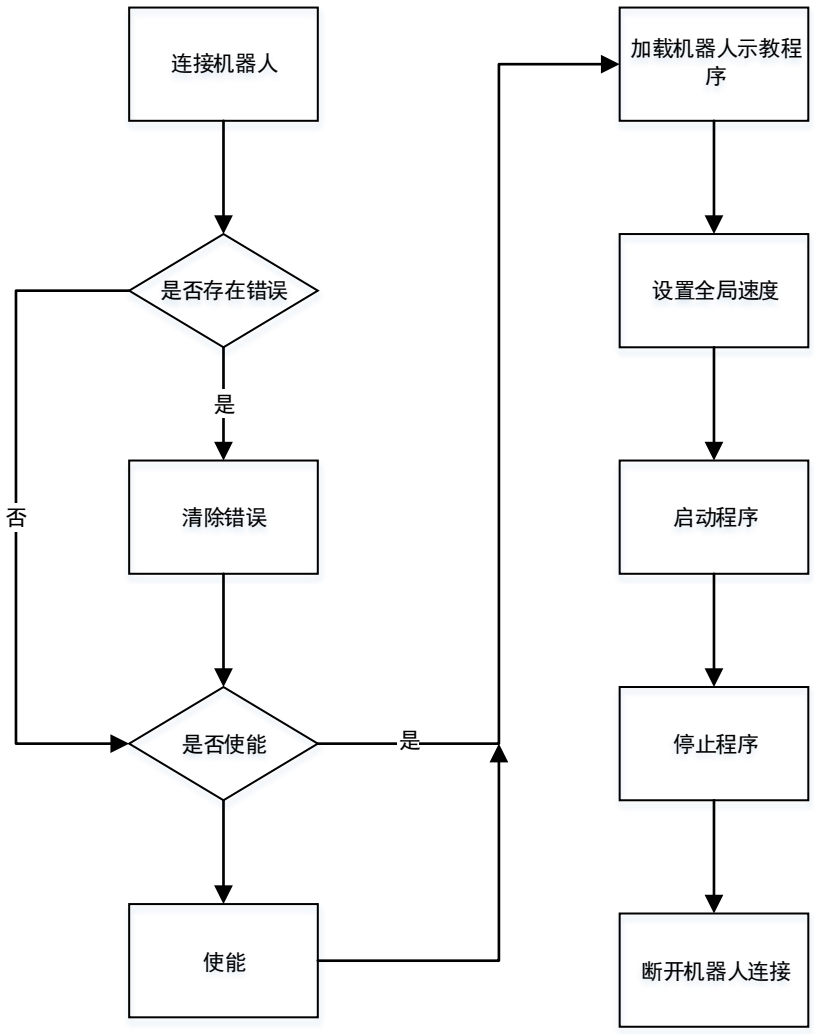
5.1 本节概要

本节主要通过典型案例进一步介绍 API 接口的调用过程。

5.2 上位机加载机器人示教程序并再现运行

在进行该操作前，请确认目标机器人的示教程序可以正常运行，同时将示教器上的模式开关切换至再现模式。

整体流程：



示例代码如下：

```
bool ret = false;
/*创建API功能库对象*/
```

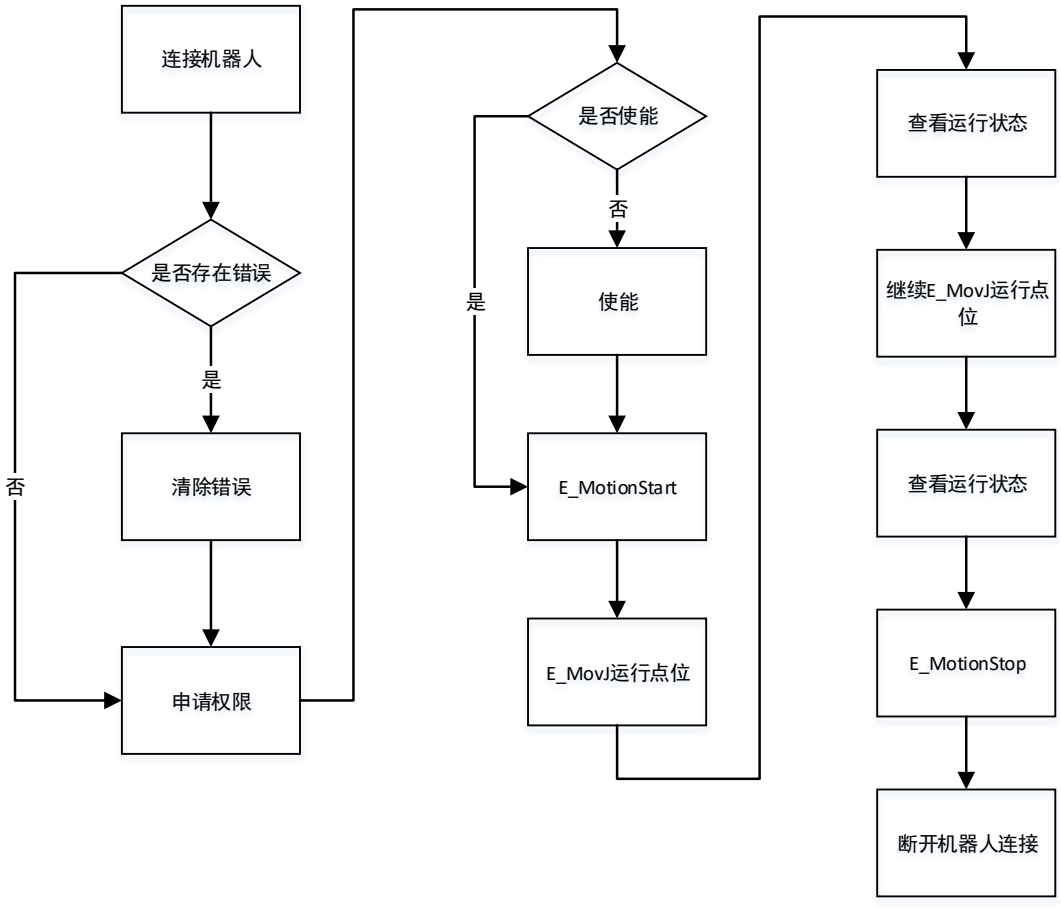
```
EstunRemoteApiLib *api = new EstunRemoteApiLib;
/*连接机器人服务器*/
ret = api->connectToServer("192.168.6.63");
if (!ret)
{
    //此处加入异常处理代码
    return;
}
/*检查是否存在错误*/
if (0 != api->E_GetErrorId())
{
    /*清除错误*/
    api->E_ClearError();
}
/*机器人上使能*/
ret = api->E_SetServoOn(true);
if (!ret)
{
    //此处加入异常处理代码
    return;
}
/*加载示教器程序*/
ret = api->E_LoadProgame("Estun","test");
if (!ret)
{
    //此处加入异常处理代码
    return;
}
/*设置全局速度*/
ret = api->E_SetGlobalSpeed(20);
if (!ret)
{
    //此处加入异常处理代码
    return;
}
/*启动程序*/
ret = api->E_ProgRun();
if (!ret)
{
    //此处加入异常处理代码
    return;
}
ret = api->E_ProgStop();
```

```
if (!ret)
{
    //此处加入异常处理代码
    return;
}
/*断开连接*/
api->disConnect();
/*释放API功能库*/
delete api;
```

5.3 上位机规划点位控制机器人运动

在进行该操作前，请确认上位机规划的点位目标机器人可以安全到达，同时将示教器上的模式开关切换至远程模式，API Mode 使能。

整体流程：



示例代码如下：

```
bool ret = false;
/*创建API功能库对象*/
EstunRemoteApiLib *api = new EstunRemoteApiLib;
```

```
/*连接机器人服务器*/
ret = api->connectToServer("192.168.6.63");
if (!ret)
{
    //此处加入异常处理代码
    return;
}
/*检查是否存在错误*/
if (0 != api->E_GetErrorId())
{
    /*清除错误*/
    api->E_ClearError();
}
/*申请权限*/
E_ROB_PERMIT permit = api->E_AcquirePermit();
if (permit.m_mainctrlcode <= 1 )
{
    //此处加入异常处理代码
    return;
}
/*机器人上使能*/
ret = api->E_SetServoOn(true);
if (!ret)
{
    //此处加入异常处理代码
    return;
}
/*启动机器人运动*/
ret = api->E_MotionStart();
if (!ret)
{
    //此处加入异常处理代码
    return;
}
E_ROB_POS pos = E_ROB_POS(0,0,0,0,30,0);
/*MovJ运动*/
E_ROB_JOBID jobid = api->E_MovJ2(pos);
/*检查运动完成状态*/
int status = api->E_GetCurMovCmdSts();
if (status == 0)
{
    //运动未完成，自定义操作，可循环查询
}
```

```
std::vector<double>  tmpApos;
tmpApos.clear();
tmpApos.push_back(90);
tmpApos.push_back(0);
tmpApos.push_back(0);
tmpApos.push_back(0);
tmpApos.push_back(90);
tmpApos.push_back(0);
pos.setAPosByArray(tmpApos);
/*MovJ运动下一点*/
jobid = api->E_MovJ2(pos);
/*检查运动完成状态*/
status = api->E_GetMovCmdSts(jobid);
if (status == 0)
{
    //运动未完成，自定义操作，可循环查询
}
ret = api->E_MotionStop();
if (!ret)
{
    //此处加入异常处理代码
    return;
}

/*断开连接*/
api->disConnect();
/*释放API功能库*/
delete api;
```

第 6 章 接口详细参考

6.1 本节概要

本节详细说明了 API 库提供的接口功能以及参数说明。

6.2 结构化数据

6.2.1 内部字符串

Class EstunRobot::ESString

内部字符串类型.

构造函数说明

EstunRobot::ESString::ESString ()

默认构造函数，构造一个空字符串.

EstunRobot::ESString::ESString (const char * c_str)

构造函数，构造一个内容为 c_str 的字符串.

参数

c_str	字符串内容

EstunRobot::ESString::ESString (const ESString & t)

拷贝构造函数.

参数

t	内部字符串类型

6.2.2 权限信息

Struct EstunRobot::E_ROB_PERMIT

权限信息.

结构体成员变量说明

变量名	类型	说明
m_mainctrlcode	uint16_t	权限唯一码，大于 1 的整数表示有主控运动操作权限，小于等于 1 是监控权限，否则没有（见 1.5 节权限说明）
m_timestamp	uint16_t	时间戳，时间戳大于 0 表示正确获取到权限，否

		则获取权限失败（见 1.5 节权限说明）

6.2.3 坐标类型

（含义见《ER 系列机器人操作手册》 8.1 节机器人坐标系）

enum EstunRobot::E_CoordType

坐标类型 0-关节坐标，1-世界(直角)坐标，2-工具坐标，3-用户坐标，4-错误

6.2.4 系统模式

enum EstunRobot::E_SysModeType

系统模式类型 0-手动，1-自动，2-远程，3-禁止，4-多工位，5-API，6-错误

6.2.5 运动模式

enum EstunRobot::E_RunModeType

运动模式类型 1-单步，2-连续，3-错误

6.2.6 空间坐标形态配置数据

（含义见《ER 系列机器人操作手册》 10.2 节 POSCFG）

Class EstunRobot::E_ROB_POSCFG

空间位置形态配置数据.

构造函数说明

EstunRobot::E_ROB_POSCFG::E_ROB_POSCFG ()

默认构造，值无效.

EstunRobot::E_ROB_POSCFG::E_ROB_POSCFG (int mode, int cf1 = 0, int cf2 = 0, int cf3 = 0, int cf4 = 0, int cf5 = 0, int cf6 = 0)

基本构造函数.

参数

mode	关节位置模式，[0,7]
cf1	关节 1 号轴角度所在的象限值
cf2	关节 2 号轴角度所在的象限值
cf3	关节 3 号轴角度所在的象限值
cf4	关节 4 号轴角度所在的象限值

cf5	关节 5 号轴角度所在的象限值
cf6	关节 6 号轴角度所在的象限值

EstunRobot::E_ROB_POSCFG::E_ROB_POSCFG (const std::vector< int > & c_array)

数组参数构造函数.

参数

c_array	形态配置数据，{mode,cf1,cf2,cf3,cf4,cf5,cf6}

EstunRobot::E_ROB_POSCFG::E_ROB_POSCFG (const E_ROB_POSCFG & t)

拷贝构造函数

参数

t	空间位置形态配置数据源

6.2.7 机器人位置

（含义见《ER 系列机器人操作手册》 10.2 节 APOS 和 CPOS）

Class EstunRobot::E_ROB_POS

机器人位置类.

成员枚举类型说明

enum EstunRobot::E_ROB_POS::PosType

坐标类型. 0-轴关节坐标，1-空间坐标

构造函数说明

EstunRobot::E_ROB_POS::E_ROB_POS ()

默认构造，值无效.

EstunRobot::E_ROB_POS::E_ROB_POS (double j1, double j2, double j3, double j4, double j5, double j6, double j7 = 0, double j8 = 0, double j9 = 0, double j10 = 0, double j11 = 0, double j12 = 0, double j13 = 0, double j14 = 0, double j15 = 0, double j16 = 0)

轴关节坐标构造函数.

参数

j1	关节 1 号轴的坐标
j2	关节 2 号轴的坐标
j3	关节 3 号轴的坐标
j4	关节 4 号轴的坐标
j5	关节 5 号轴的坐标
j6	关节 6 号轴的坐标
j7	关节 7 号轴的坐标，选填，默认为 0

j8	关节 8 号轴的坐标，选填，默认为 0
j9	关节 9 号轴的坐标，选填，默认为 0
j10	关节 10 号轴的坐标，选填，默认为 0
j11	关节 11 号轴的坐标，选填，默认为 0
j12	关节 12 号轴的坐标，选填，默认为 0
j13	关节 13 号轴的坐标，选填，默认为 0
j14	关节 14 号轴的坐标，选填，默认为 0
j15	关节 15 号轴的坐标，选填，默认为 0
j16	关节 16 号轴的坐标，选填，默认为 0

EstunRobot::E_ROB_POS::E_ROB_POS (const std::vector< double > & apos_Array)

轴关节坐标构造函数.

参数

apos_Array	坐标数组，{j1,j2,j3...j16}

EstunRobot::E_ROB_POS::E_ROB_POS (const E_ROB_POSCFG & cfg, double x, double y, double z, double a, double b, double c, double a7 = 0, double a8 = 0, double a9 = 0, double a10 = 0, double a11 = 0, double a12 = 0, double a13 = 0, double a14 = 0, double a15 = 0, double a16 = 0)

空间坐标构造函数.

参数

cfg	空间位置形态配置数据，类型说明详见 6.2.6
x	TCP 点在参考坐标系上 x 方向的坐标
y	TCP 点在参考坐标系上 y 方向的坐标
z	TCP 点在参考坐标系上 z 方向的坐标
a	TCP 点相对于参考坐标系 z 轴旋转的欧拉角
b	TCP 点相对于参考坐标系 y 轴旋转的欧拉角
c	TCP 点相对于参考坐标系 x 轴旋转的欧拉角
a7	附加轴关节 7 号轴的坐标，选填，默认为 0
a8	附加轴关节 8 号轴的坐标，选填，默认为 0
a9	附加轴关节 9 号轴的坐标，选填，默认为 0
a10	附加轴关节 10 号轴的坐标，选填，默认为 0
a11	附加轴关节 11 号轴的坐标，选填，默认为 0
a12	附加轴关节 12 号轴的坐标，选填，默认为 0
a13	附加轴关节 13 号轴的坐标，选填，默认为 0
a14	附加轴关节 14 号轴的坐标，选填，默认为 0
a15	附加轴关节 15 号轴的坐标，选填，默认为 0
a16	附加轴关节 16 号轴的坐标，选填，默认为 0

EstunRobot::E_ROB_POS::E_ROB_POS (const std::vector< int > & cfg_Array, const std::vector< double > & cpos_Array)

空间坐标构造函数.

参数

cfg_Array	形态配置数据，{mode,cf1,cf2,cf3,cf4,cf5,cf6}
cpos_Array	空间坐标数组，{x,y,z,a,b,c,a7...a16}

EstunRobot::E_ROB_POS::E_ROB_POS (const E_ROB_POS & t)

拷贝构造函数.

参数

t	轴关节坐标/空间坐标

6.2.8 运动方向

enum EstunRobot::E_DistentionType

方向类型 1-为正方向，0-负方向

6.2.9 AddDo

Struct EstunRobot::E_ROB_APIBUFF

附加指令，具体内容由其他结构参数转换得到，需要调用 AddDoTypeConvert 相关接口

结构体成员变量说明

变量名	类型	说明
m_type	uint8_t	附加指令类型
m_data	uint8_t[]	指令内容二进制数组

6.2.10 AddDO_IO

Struct EstunRobot::E_ROB_ADDDO_IO

附加指令结构一.

结构体成员变量说明

变量名	类型	说明
ioType	E_IOType	IO 类型，类型说明详见 6.2.17
ioindex	int	IO 序号
iovalue	double	IO 值
trigType	int	触发类型（1:路径百分比；2：距离）
exTimes	int	触发延时,单位毫秒，负数为提前触发
exPercent	int	当前运动执行触发百分比
exDistance	double	当前运动执行触发距离

6.2.11 机器人速度

（含义见《ER 系列机器人操作手册》 10.3 节 SPEED）

Class EstunRobot::E_ROB_SPEED

机器人运动速度类.

构造函数说明

EstunRobot::E_ROB_SPEED::E_ROB_SPEED ()

默认构造，值无效.

EstunRobot::E_ROB_SPEED::E_ROB_SPEED (double m_Per, double m_Tcp, double m_ori, double m_exj_l, double m_exj_r)

基本构造函数.
参数

m_Per	关节速度百分比 ，定义机器人执行关节运动时的关节转动速度，取值范围 1~100（百分比），适用于 MovJ\2
m_Tcp	TCP 线速度，定义机器人执行圆弧运动时末端点的速度 ，单位 mm/s，适用于 MovL\2、MovC\2、MovArch\2
m_ori	空间旋转速度，定义机器人末端点的姿态旋转速度
m_exj_l	外部轴线速度，定义外部直线轴的运动速度
m_exj_r	外部轴角速度，定义外部旋转轴的运动速度

EstunRobot::E_ROB_SPEED::E_ROB_SPEED (const E_ROB_SPEED & t)

拷贝构造函数.
参数

t	机器人运动速度

6.2.12 机器人过渡区域

（含义见《ER 系列机器人操作手册》 10.3 节 ZONE）

Class EstunRobot::E_ROB_ZONE

机器人转弯区类.

构造函数说明

EstunRobot::E_ROB_ZONE::E_ROB_ZONE ()

默认构造，值无效.

EstunRobot::E_ROB_ZONE::E_ROB_ZONE (int8_t m_ZoneType, double m_Per, double m_dis, int m_vConst)

基本构造函数.

参数

m_ZoneType	过渡类型(0-无过渡/1-绝对过渡/2-相对过渡);
m_Per	关节轨迹转弯百分比，适用于 MovJ\2 等关节运动指令，表示距离目标角度还有多远时开始转弯
m_dis	笛卡尔空间转弯区大小，用于 MovL\2，MovC\2 等直线圆弧运动指令，定义笛卡尔空间轨迹的转弯区大小，即当机器人运动到距离目标点还有 dis 毫米的地方时，开始转向往下一个目标点运动，单位是 mm
m_vConst	恒速过渡使能状态，过渡段线速度是否恒定不变的使能状态。 为 1 表示过渡恒定不变，为 0 表示跟随过渡参数

EstunRobot::E_ROB_ZONE::E_ROB_ZONE (const E_ROB_ZONE & t)

拷贝构造函数.

参数

t	机器人转弯区

6.2.13 指令信息

Struct EstunRobot::E_ROB_JOBID

指令标识信息.

结构体成员变量说明

变量名	类型	说明
m_jobID	int	指令 ID，大于 0 表示指令下发成功，否则表示指令下发失败
m_timeStamp	uint32_t	运动指令下发成功的当前时间戳

6.2.14 负载参数

（含义见《ER 系列机器人操作手册》 10.3 节 CenterPos、InertiaTensor、LoadDyn）

Class EstunRobot::E_ROB_PAYLOAD

机器人负载类.

构造函数说明

EstunRobot::E_ROB_PAYLOAD::E_ROB_PAYLOAD ()

默认构造，值无效.

EstunRobot::E_ROB_PAYLOAD::E_ROB_PAYLOAD (double m_M, double m_Mx, double m_My, double m_Mz, double m_Ixx = 0, double m_Iyy = 0, double m_Izz = 0, double m_Ixy = 0, double m_Ixz = 0, double m_Iyz = 0)

m_M	工具&负载的重量， 单位是 kg
m_Mx	安装的工具或装夹的负载在坐标系 OTool-XYZ 的 X 方向上的偏移量，单位是 mm
m_My	安装的工具或装夹的负载在坐标系 OTool-XYZ 的 Y 方向上的偏移量，单位是 mm
m_Mz	安装的工具或装夹的负载在坐标系 OTool-XYZ 的 Z 方向上的偏移量，单位是 mm
m_Ixx	安装的工具或装夹的负载在重心位臵的 X 方向回转的惯量矩，单位是 kg· ○ mm2
m_Iyy	安装的工具或装夹的负载在重心位臵的 Y 方向回转的惯量矩，单位是 kg· ○ mm2
m_Izz	安装的工具或装夹的负载在重心位臵的 Z 方向回转的惯量矩，单位是 kg· ○ mm2
m_Ixy	安装的工具或装夹的负载在重心位臵的 XY 交叉方向的惯量积，单位是 kg· ○ mm2
m_Ixz	安装的工具或装夹的负载在重心位臵的 XZ 交叉方向的惯量积，单位是 kg· ○ mm2
m_Iyz	安装的工具或装夹的负载在重心位臵的 YZ 交叉方向的惯量积，单位是 kg· ○ mm2

EstunRobot::E_ROB_PAYLOAD::E_ROB_PAYLOAD (const std::vector< double > & array)

array	负载数据数组，{m_M,m_Mx,m_My,m_Mz,m_Ixx,m_Iyy,m_Izz,m_Ixy,m_Ixz,m_Iyz}

EstunRobot::E_ROB_PAYLOAD::E_ROB_PAYLOAD (const E_ROB_PAYLOAD & t)

拷贝构造函数.

t	机器人负载

6.2.15 用户坐标参数

（含义见《ER 系列机器人操作手册》 10.3 节 USERCOOR）

Class EstunRobot::E_ROB_COORD

用户坐标系类.

构造函数说明

EstunRobot::E_ROB_COORD::E_ROB_COORD()

默认构造，值无效.

EstunRobot::E_ROB_COORD::E_ROB_COORD (double x, double y, double z, double a, double b, double c)

x	TCP 相对于世界（基） 坐标系在 x 方向的位移偏移量，单位是 mm
y	TCP 相对于世界（基） 坐标系在 y 方向的位移偏移量，单位是 mm
z	TCP 相对于世界（基） 坐标系在 z 方向的位移偏移量，单位是 mm
a	TCP 相对于世界（基） 坐标系 z 轴旋转的欧拉角，单位是 deg
b	TCP 相对于世界（基） 坐标系 y’轴旋转的欧拉角，单位是 deg
c	TCP 相对于世界（基） 坐标系 x’’轴旋转的欧拉角，单位是 deg

EstunRobot::E_ROB_COORD::E_ROB_COORD (const std::vector< double > & array)

array	坐标系值数组，{x,y,z,a,b,c}

EstunRobot::E_ROB_COORD::E_ROB_COORD (const E_ROB_COORD & t)

t	用户坐标系

6.2.16 用户坐标类型

enum EstunRobot::E_UserCoordType

用户坐标系类型 0-用户坐标，1-外部坐标系

6.2.17 IO 类型

（含义见《ER 系列机器人操作手册》 10.4 节 IO 数据类型）

enum EstunRobot::E_IOType

IO 类型. 0-DI，1-DO，2-AO，3-AI，11-SIM_DI，12-SIM_DO，13-SIM_AI，14-SIM_AO

6.2.18 变量类型

（含义见《ER 系列机器人操作手册》 10.1 节 基本数据类型）

enum EstunRobot::E_VarType

数据类型. 1-整型，2-实数，3-关节位置，4-空间位置，5-字符串，6-实数数组，7-工具参数，8-用户坐标参数

6.2.19 变量作用域

（含义见《ER 系列机器人操作手册》 9.3 节 程序数据）

enum EstunRobot::E_ScopeType

变量域类型. 0-系统，1-全局，2-工程，3-程序本地

6.2.20 程序信息

Struct EstunRobot::E_ROB_PROGNAME

工程程序名.

结构体成员变量说明

变量名	类型	说明
prjname	ESString	当前加载的工程名，类型说明详见 6.2.1
programname	ESString	当前加载的程序名，类型说明详见 6.2.1
exeprogramName	ESString	当前运行的程序名，类型说明详见 6.2.1

6.2.21 程序运行状态

enum EstunRobot::E_RunStatusType

运动状态类型. 1-运行，2-暂停，3-停止，4-错误

6.2.22 机器人使能状态

enum EstunRobot::E_MotStatusType

mot 使能状态. 0-未使能, 1-使能, 3-错误

6.3 接口列表

6.3.1 默认构造函数

EstunRemoteApiLib::EstunRemoteApiLib()
类的默认构造接口，默认为当前目录下的 Config 文件夹.

6.3.2 有参构造函数

EstunRemoteApiLib::EstunRemoteApiLib(ESString configDir)
类的构造接口.

参数

configDir	配置文件以及错误信息文件所在文件夹

6.3.3 有参构造函数

void EstunRemoteApiLib::setLogConfig(ESString logDir, unsigned int size);

设置本地日志配置.

参数

logDir	志文件目录，不指定则在当前目录生成
size	日志文件大小

6.3.4 日志打印开关

void EstunRemoteApiLib::setLogEnable (bool enable)

是否输出本地日志.

参数

enable	为 true 写本地日志，false 则不写日志

6.3.5 写本地日志

void EstunRemoteApiLib::writeLog (ESString str)

写本地日志.

参数

str	日志信息，类型说明详见 6.2.1

6.3.6 错误日志

ESString EstunRemoteApiLib::getLastErr () const

获取最近一次错误的信息.

返回

错误信息字符串

6.3.7 连接服务器

bool EstunRemoteApiLib::connectToServer (ESString ipAddr, bool isAutoReConnect = `false`)

连接服务器.

参数

ipAddr	服务 IP 地址，IPV4 格式，类型说明详见 6.2.1
isAutoReConnect	是否自动重连，如果为 True，客户端在异常掉线时会自动重连，否则不会

返回

成功返回 true，失败返回 false

6.3.8 断开连接

bool EstunRemoteApiLib::disConnect ()

断开与服务的连接.

返回

成功返回 true，失败返回 false

注

调用该接口与插拔网线效果相同，控制器不会报错并且将未完成的指令完成。

6.3.9 连接检测

bool EstunRemoteApiLib::isConnected ()

检测是否已连接服务.

返回

连接返回 true，断开返回 false

6.3.10 获取当前工作状态

int EstunRemoteApiLib::E_GetAPIStatus();

获取当前模式的工作状态

返回

-1 表示失败，0 表示正常，1 表示机器人错误，2 表示机器人处于停止状态，需要调用 start 接口启动

6.3.11 申请权限

E_ROB_PERMIT EstunRemoteApiLib::E_AcquirePermit ()

申请机器人运动操作权限.

返回

权限信息，类型说明详见 6.2.2

6.3.12 查询权限

E_ROB_PERMIT EstunRemoteApiLib::E_CurrentPermit ()

查询当前机器人运动操作权限.

返回

权限信息，类型说明详见 6.2.2

6.3.13 释放权限

bool EstunRemoteApiLib::E_ReleasePermit (E_ROB_PERMIT permit)

释放机器人运动操作权限.

参数

permit	之前获取的权限，类型说明详见 6.2.2

返回

成功返回 true，失败返回 false

6.3.14 使能开关

bool EstunRemoteApiLib::E_SetServoOn (bool isOn)

设置机器人伺服使能.

参数

isOn	为 true 上使能，为 false 掉使能

返回

成功返回 true，失败返回 false

6.3.15 获取使能状态

E_ServoStatusType EstunRemoteApiLib::E_GetServoOn ()

获取机器人伺服是否使能.

返回

使能标志，类型说明详见 6.2.22

6.3.16 jog 使能开关

bool EstunRemoteApiLib::E_SetJogEnable (bool isEnable)

设置当前模式下 job 是否可用.

参数

isEnable	为 true，jog 可用，为 false，jog 不可用

返回

成功返回 true，失败返回 false

6.3.17 获取 jog 使能状态

bool EstunRemoteApiLib::E_GetJogEnable ()

获取当前模式下 job 是否可用.

返回

可用返回 true，不可用返回 false

6.3.18 设置到位精度

bool EstunRemoteApiLib::E_SetFine (double value)

设置默认到位精度值，掉电保存.

参数

value	到位精度值

返回

成功返回 true，失败返回 false

6.3.19 获取到位精度

bool EstunRemoteApiLib::E_GetFine (double & value)

获取默认到位精度值.

参数

value	到位精度值，引用类型

返回

成功返回 true，失败返回 false

6.3.20 设置 jog 坐标类型

（含义见《ER 系列机器人操作手册》 8.1 节机器人坐标系）

bool EstunRemoteApiLib::E_SetJogCoord (E_CoordType coordType)

设置点动坐标系类型.

参数

coordType	坐标系类型，类型说明详见 6.2.3

返回

成功返回 true，失败返回 false

6.3.21 获取 jog 坐标类型

（含义见《ER 系列机器人操作手册》 8.1 节机器人坐标系）

E_CoordType EstunRemoteApiLib::E_GetJogCoord ()

获取点动坐标系类型.

返回

坐标系类型，类型说明详见 6.2.3

6.3.22 获取机器人类型

bool EstunRemoteApiLib::E_GetRobotType (ESString & value)

获取机器人类型.

参数

value	机器人型号，类型说明详见 6.2.1

返回

成功返回 true，失败返回 false

6.3.23 获取机器人信息

bool EstunRemoteApiLib::E_GetRobotInfo (ESString & value)

获取机器人信息.

参数

value	机器人信息，类型说明详见 6.2.1

返回

成功返回 true，失败返回 false

6.3.24 获取机器人在线状态

int EstunRemoteApiLib::E_GetRobotStatus ()

获取机器人状态.

返回

12345 表示机器人在线，其余为机器人不在线

6.3.25 获取控制器状态

int EstunRemoteApiLib::E_GetAPIStatus ()

获取当前模式的工作状态.

返回

-1 表示失败，0 表示正常，1 表示机器人错误，2 表示机器人处于停止状态，需要调用 start 接口启动

6.3.26 设置全局速度

bool EstunRemoteApiLib::E_SetGlobalSpeed (int value)

设置全局速度.

参数

value	速度值[0-100]

返回

成功返回 true，失败返回 false

6.3.27 获取全局速度

int EstunRemoteApiLib::E_GetGlobalSpeed ()

获取全局速度.

返回

速度值

6.3.28 设置系统模式

bool E_SetSysMode(E_SysModeType sysMode)

设置系统模式.

参数

sysMode	系统模式，类型说明详见 6.2.4

返回

return 成功返回 true，失败返回 false

6.3.29 获取系统模式

E_SysModeType EstunRemoteApiLib::E_GetSysMode ()

获取系统模式.

返回

系统模式，类型说明详见 6.2.4

6.3.30 设置运动模式

bool EstunRemoteApiLib::E_SetRunMode (E_RunModeType mode)

设置运动模式.

参数

mode	运动模式，类型说明详见 6.2.5

返回

成功返回 true，失败返回 false

6.3.31 获取运动模式

E_RunModeType EstunRemoteApiLib::E_GetCurRunMode ()

获取运动模式.

返回

运动模式，类型说明详见 6.2.5

6.3.32 清除错误

bool EstunRemoteApiLib::E_ClearError ()

清除错误报警.

返回

成功返回 true，失败返回 false

6.3.33 清除所有错误报警后获取是否还有错误

bool E_CheckClearError()

清除所有错误报警后获取是否还有错误.

返回

没有错误返回 true，有错误返回 false

6.3.34 获取错误号

int EstunRemoteApiLib::E_GetErrorId ()

获取错误号.

返回

错误号

6.3.35 获取错误详情

ESString EstunRemoteApiLib::E_GetErrorString (int errorId)

获取错误内容.

参数

errorId	错误号

返回

错误详情，信息的语言根据 ServerConfig.xml 的 Language 配置项决定，目前仅支持中文和英文

6.3.36 获取错误详情

ESString EstunRemoteApiLib::E_GetErrorString (int errorId，ESString language)

获取错误内容.

参数

errorId	错误号
language	语言，ch 表示获取中文信息，en 表示获取英文信息

返回

错误详情

6.3.37 获取关节坐标

（含义见《ER 系列机器人操作手册》 10.2 节位置数据类型）

E_ROB_POS EstunRemoteApiLib::E_GetCurJPos ()

获取当前位置信息，关节坐标系.

返回

位置信息，APos，类型说明详见 6.2.7

6.3.38 获取世界坐标

E_ROB_POS EstunRemoteApiLib::E_GetCurWPos ()

获取当前位置信息，世界坐标系.

返回

位置信息，CPos，类型说明详见 6.2.7

6.3.39 运动开始

bool EstunRemoteApiLib::E_MotionStart ()

唤醒机器人运动，从 stop 模式恢复.

返回

成功返回 true，失败返回 false

6.3.40 运动暂停

bool EstunRemoteApiLib::E_MotionPause ()

暂停机器人运动，不会清空运动队列.

返回

成功返回 true，失败返回 false

6.3.41 运动继续

bool EstunRemoteApiLib::E_MotionContinue ()

恢复机器人运动，继续执行运动队列.

返回

成功返回 true，失败返回 false

6.3.42 运动停止

bool EstunRemoteApiLib::E_MotionStop ()

停止机器人运动，同时清空运动队列.

返回

成功返回 true，失败返回 false

6.3.43 点动

bool EstunRemoteApiLib::E_JogMotion (int axisId, E_DistentionType distention, bool mov)

点动.

参数

axisId	值[0,5]，对应关节坐标系下为轴 1-6，世界坐标系下为 xyzabc
distention	运动方向，类型说明详见 6.2.8
mov	为 true 运动，为 false 停止

返回

成功返回 true，失败返回 false

6.3.44 寸动

bool EstunRemoteApiLib::E_JogMicroMotion (int axisId, E_DistentionType distention, double dist, bool mov)

寸动.

参数

axisId	值[0,5]，对应关节坐标系下为轴 1-6，世界坐标系下为 xyzabc
distention	运动方向，类型说明详见 6.2.8
dist	寸动距离，单位毫米
mov	为 true 运动，为 false 停止

返回

成功返回 true，失败返回 false

6.3.45 指令内变量说明

MovJ、MovL、MovC、MovArch 等指令内变量名，由作用域和变量名构成，例："t_g.p1"，代表的是系统内的名为 p1 的全局变量，需要提前在系统内创建；t_s 表示系统变量，t_g 表示全局变量，t_p 表示工程变量， t_l 表示程序变量

6.3.46 MovJ

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovJ）

E_ROB_JOBID EstunRemoteApiLib::E_MovJ (ESString Pname, ESString velName = "", ESString blendType = "", ESString zame = "", ESString tname = "", ESString coorname = "", ESString payloadName = "", const E_ROB_APIBUFF & addDo = `E_ROB_APIBUFF()`, double valfine = `0.0`, bool IsWaitFinish = `false`)

执行 MovJ 指令.

参数

Pname	目标位置点变量名，例："t_g.p1"，类型说明详见 6.2.1
velName	运动速度变量名， 例："t_g.SPEED0"，类型说明详见 6.2.1
blendType	过渡类型，"","Rel","Abs"，三选一，其中：“ ”表示无过渡；“Rel”表示相对过渡；“Abs”表示绝对过渡；类型说明详见 6.2.1
zame	过渡参数变量名， 例："t_g.ZONE0"，类型说明详见 6.2.1
tname	工具变量名， 例："t_g.TOOL1"，类型说明详见 6.2.1
coorname	坐标系变量名， 例："t_g.USER0"，类型说明详见 6.2.1
payloadName	负载变量名， 例："t_g. PAYLOAD0"，类型说明详见 6.2.1
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

附加说明

API 暂时不支持“速率类型”参数设置。“速率类型”参数含义可参考《ER 系列机器人操作手册》 11.1 节运动指令 MovJ。

6.3.47 MovJ2

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovJ）

E_ROB_JOBID EstunRemoteApiLib::E_MovJ2 (const E_ROB_POS & dstPoint, const E_ROB_SPEED & velo = `E_ROB_SPEED()`, const E_ROB_ZONE & zone = `E_ROB_ZONE()`, int payloadId = `0`, int toolId = `0`, int coordId = `0`, const E_ROB_APIBUFF & addDo = `E_ROB_APIBUFF()`, double valfine = `0.0`, bool IsWaitFinish = `false`)

执行 MovJ2 指令.

参数

dstPoint	目标位置点，类型说明详见 6.2.7
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12
payloadId	负载号
toolId	工具号
coordId	用户坐标号
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

附加说明

API 暂时不支持“速率类型”参数设置。“速率类型”参数含义可参考《ER 系列机器人操作手册》 11.1 节运动指令 MovJ。

6.3.48 MovL

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovL）

E_ROB_JOBID EstunRemoteApiLib::E_MovL (ESString Pname, ESString velName = `""`, ESString blendType = `""`, ESString zname = `""`, ESString tname = `""`, ESString coorname = `""`, ESString payloadName = `""`, ESString overrideType = `"GON"`, const E_ROB_APIBUFF & addDo = `E_ROB_APIBUFF()`, double valfine = `0.0`, bool IsWaitFinish = `false`)

执行 MovL 指令.

参数

Pname	目标位置点变量名， 例："t_g.p1"，类型说明详见 6.2.1
velName	运动速度变量名， 例："t_g.SPEED0"，类型说明详见 6.2.1
blendType	过渡类型，"","Rel","Abs"，三选一，类型说明详见 6.2.1
zame	过渡参数变量名， 例："t_g.ZONE0"，类型说明详见 6.2.1
tname	工具变量名， 例："t_g.TOOL1"，类型说明详见 6.2.1
coorname	坐标系变量名， 例："t_g.USER0"，类型说明详见 6.2.1
payloadName	负载变量名， 例："t_g. PAYLOAD0"，类型说明详见 6.2.1
overrideType	速率类型，"","GOFF","GON"，三选一，其中：“ ”表示无过渡；“Rel”表示相对过渡；“Abs”表示绝对过渡；类型说明详见 6.2.1
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

6.3.49 MovL2

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovL）

E_ROB_JOBID EstunRemoteApiLib::E_MovL2 (const E_ROB_POS & dstPoint, const E_ROB_SPEED & velo = `E_ROB_SPEED()`, const E_ROB_ZONE & zone = `E_ROB_ZONE()`, int payloadId = `0`, int toolId = `0`, int coordId = `0`, int overrideType = `1`, const E_ROB_APIBUFF & addDo = `E_ROB_APIBUFF()`, double valfine = `0.0`, bool IsWaitFinish = `false`)

执行 MovL2 指令.

参数

dstPoint	目标位置点，类型说明详见 6.2.7
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12
payloadId	负载号
toolId	工具号
coordId	用户坐标号
overrideType	速率类型, 99:GOVROFF（全局速度不生效）;非 99:GOVRON（全局速度生效）

addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

6.3.50 MovC

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovC）

E_ROB_JOBID EstunRemoteApiLib::E_MovC (ESString PAuxname, ESString PDstname, ESString velName = "", ESString blendType = "", ESString zame = "", ESString tname = "", ESString coorname = "", ESString payloadName = "", ESString overrideType = "GON", const E_ROB_APIBUFF & addDo = E_ROB_APIBUFF(), double valfine = 0.0, bool IsWaitFinish = false)

执行 MovC 指令.

参数

PAuxname	辅助位置点变量名，例："t_g.p0"，类型说明详见 6.2.1
PDstname	目标位置点变量名，例："t_g.p1"，类型说明详见 6.2.1
velName	运动速度变量名，例："t_g.SPEED0"，类型说明详见 6.2.1
blendType	过渡类型，"","Rel","Abs"，三选一，其中：“”表示无过渡；“Rel”表示相对过渡；“Abs”表示绝对过渡；类型说明详见 6.2.1
zame	过渡参数变量名，例："t_g.ZONE0"，类型说明详见 6.2.1
tname	工具变量名，例："t_g.TOOL1"，类型说明详见 6.2.1
coorname	坐标系变量名，例："t_g.USER0"，类型说明详见 6.2.1
payloadName	负载变量名，例："t_g. PAYLOAD0"，类型说明详见 6.2.1
overrideType	速率类型，"","GOFF","GON"，三选一，类型说明详见 6.2.1
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

6.3.51 MovC2

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovC）

E_ROB_JOBID EstunRemoteApiLib::E_MovC2 (const E_ROB_POS & auxP, const E_ROB_POS & dstPoint, const E_ROB_SPEED & velo = `E_ROB_SPEED()`, const E_ROB_ZONE & zone = `E_ROB_ZONE()`, int payloadId = `0`, int toolId = `0`, int coordId = `0`, int overrideType = `1`, const E_ROB_APIBUFF & addDo = `E_ROB_APIBUFF()`, double valfine = `0.0`, bool IsWaitFinish = `false`)

执行 MovC2 指令.

参数

auxP	辅助位置点，类型说明详见 6.2.7
dstPoint	目标位置点，类型说明详见 6.2.7
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12
payloadId	负载号
toolId	工具号
coordId	用户坐标号
overrideType	速率类型, 99:GOVROFF（全局速度不生效）;非 99:GOVRON（全局速度生效）
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

6.3.52 MovArch

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovArch）

E_ROB_JOBID EstunRemoteApiLib::E_MovArch (ESString Pname, ESString velName = "", ESString blendType = "", ESString zame = "", ESString tname = "", ESString coorname = "", ESString payloadName = "", ESString cornerName = "", double height = `0.0`, const E_ROB_APIBUFF & addDo = `E_ROB_APIBUFF()`, double valfine = `0.0`, bool IsWaitFinish = `false`)

执行 MovArch 指令.

参数

Pname	目标位置点变量名， 例："t_g.p1"，类型说明详见 6.2.1
velName	运动速度变量名， 例："t_g.SPEED0"，类型说明详见 6.2.1
blendType	过渡类型，"","Rel","Abs"，三选一，其中：“ ”表示无过渡；“Rel”表示相对过渡；“Abs”表示绝对过渡；类型说明详见 6.2.1
zame	过渡参数变量名， 例："t_g.ZONE0"，类型说明详见 6.2.1
tname	工具变量名， 例："t_g.TOOL1"，类型说明详见 6.2.1
coorname	坐标系变量名， 例："t_g.USER0"，类型说明详见 6.2.1
payloadName	负载变量名， 例："t_g. PAYLOAD0"，类型说明详见 6.2.1
cornerName	拱形运动转角变量名， 例："t_g.ZONE1"，类型说明详见 6.2.1
height	抬升高度，单位毫米
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

6.3.53 MovArch2

（含义见《ER 系列机器人操作手册》 11.1 节运动指令 MovArch）

E_ROB_JOBID EstunRemoteApiLib::E_MovArch2 (const E_ROB_POS & dstPoint, const E_ROB_SPEED & velo = `E_ROB_SPEED()`, const E_ROB_ZONE & zone = `E_ROB_ZONE()`, const E_ROB_ZONE & connerZone = `E_ROB_ZONE()`, int payloadId = `0`, int toolId = `0`, int coordId = `0`, double heigt = `0.0`, const E_ROB_APIBUFF & addDo = `E_ROB_APIBUFF()`, double valfine = `0.0`, bool IsWaitFinish = `false`)

执行 MovArch2 指令.

参数

dstPoint	目标位置点，类型说明详见 6.2.7
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12
connerZone	转角参数，类型说明详见 6.2.12
payloadId	负载号

toolId	工具号
coordId	用户坐标号
heigt	抬升高度，单位毫米
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

6.3.54 获取当前运行指令号

E_ROB_JOBID EstunRemoteApiLib::E_GetCurMovCmdJobId ()

获取最近一次下发成功的运动指令的信息.

返回

运动指令信息，类型说明详见 6.2.13

6.3.55 获取当前运动完成状态

int EstunRemoteApiLib::E_GetCurMovCmdSts ()

获取最近一次下发成功的运动指令的执行状态.

返回

-1-未找到，0-运动未完成，1-运动完成

6.3.56 获取指定运动完成状态

int EstunRemoteApiLib::E_GetMovCmdSts (E_ROB_JOBID jobID)

获取指定运动指令执行状态.

参数

jobID	运动指令信息，类型说明详见 6.2.13

返回

-1-未找到，0-运动未完成，1-运动完成,2-暂停，3-结束/错误，4-(预留类型)

6.3.57 获取运动状态

int EstunRemoteApiLib::E_GetRobotRunStatus ()

获取机器人运动状态.

返回

1 表示机器人在运动，0 表示不在运动

6.3.58 设置外部坐标系

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 EXTTCP）

bool EstunRemoteApiLib::E_SetExternalTCPByName (ESString ecpName)

设置当前外部坐标系.

参数

ecpName	外部坐标系变量名，例：t_g.EXTTCP1

返回

成功返回 true，失败返回 false

注

调用该接口设置外部坐标系时，若 ecpName 不存在则创建名为 ecpName 的外部坐标系，若 ecpName 已存在则修改内存中的值。

6.3.59 设置外部坐标系

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 EXTTCP）

bool EstunRemoteApiLib::E_SetExternalTCPByID (int ecpid)

设置当前外部坐标系.

参数

ecpid	外部坐标系变量 id，如果为 0 表示不加载

返回

成功返回 true，失败返回 false

注

调用该接口设置外部坐标系时，若 ecpid 不存在则创建外部坐标系 id 为 ecpid 的外部坐标系，若 ecpid 已存在则修改内存中的值。

6.3.60 获取工具坐标系 ID

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 TOOL）

bool E_GetToolsID(int& num, int*& idArray)

获取所有工具坐标系 ID.

参数

num	获取到的个数
idArray	ID 数组指针，由接口初始化

返回

成功返回 true，失败返回 false

6.3.61 新建工具参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 TOOL）

bool EstunRemoteApiLib::E_CreateTool (int toolID, const E_ROB_COORD & toolCoord, const E_ROB_PAYLOAD & payloadvalue)

创建工具参数.

参数

toolID	新的工具号，必须大于 0
toolCoord	工具坐标参数，类型说明详见 6.2.15
payloadvalue	工具负载参数，类型说明详见 6.2.14

返回

成功返回 true，失败返回 false

注

调用该接口创建工具坐标系时，若 toolID 不存在则创建工具号为 toolID 的工具坐标系，若 toolID 已存在则修改内存中的值。

6.3.62 修改工具参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 TOOL）

bool EstunRemoteApiLib::E_ModifyTool (int toolID, const E_ROB_COORD & toolCoord, const E_ROB_PAYLOAD & payloadvalue)

修改工具参数.

参数

toolID	要修改的工具号，必须大于 0
toolCoord	工具坐标参数，类型说明详见 6.2.15
payloadvalue	工具负载参数，类型说明详见 6.2.14

返回

成功返回 true，失败返回 false

注

调用该接口修改工具坐标系时只修改内存中的值。

6.3.63 选择工具参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 TOOL）

bool EstunRemoteApiLib::E_LoadTool (int toolID)

加载工具.

参数

toolID	工具号

返回

成功返回 true，失败返回 false

6.3.64 获取当前工具 ID

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 TOOL）

int EstunRemoteApiLib::E_GetCurToolID ()

获取当前工具号.

返回

工具号

6.3.65 获取工具坐标变量名称

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 TOOL）

ESString E_GetToolName(int toolID)

获取工具坐标变量名称.

参数

toolID	工具号

返回

成功返回名称，为空则表示获取失败

6.3.66 获取工具参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 TOOL）

bool EstunRemoteApiLib::E_GetTool (int toolID, E_ROB_COORD & toolCoord, E_ROB_PAYLOAD & toolPayload)

获取工具坐标系变量.

参数

toolID	工具号
toolCoord	工具参数，引用，类型说明详见 6.2.15
toolPayload	负载参数，引用，类型说明详见 6.2.14

返回

成功返回 true，失败返回 false

6.3.67 获取用户坐标系 ID

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 USERCOORD）

bool E_GetUserCoordsID(int& num, int*& idArray)

获取所有用户坐标系 ID.

参数

num	获取到的个数
idArray	ID 数组指针，由接口初始化

返回

成功返回 true，失败返回 false

6.3.68 新建用户坐标

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 USERCOORD）

bool EstunRemoteApiLib::E_CreateUserCoord (int usrID, const E_ROB_COORD & usrData)

创建用户坐标.

参数

usrID	新的坐标号，必须大于 0
usrData	坐标参数，类型说明详见 6.2.15

返回

成功返回 true，失败返回 false

注

调用该接口创建用户坐标系时，若 usrID 不存在则创建坐标号为 usrID 的用户坐标系，若 usrID 已存在则修改内存中的值。

6.3.69 修改用户坐标

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 USERCOORD）

bool EstunRemoteApiLib::E_ModifyUserCoord (int usrID, const E_ROB_COORD & usrData)

修改用户坐标.

参数

usrID	要修改的用户坐标号，必须大于 0
usrData	坐标参数，类型说明详见 6.2.15

返回

成功返回 true，失败返回 false

注

调用该接口修改用户坐标系时只修改内存中的值。

6.3.70 选择用户坐标

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 USERCOORD）

bool EstunRemoteApiLib::E_LoadUserCoord (int usrID)

加载用户坐标.

参数

usrID	用户坐标号

返回

成功返回 true，失败返回 false

6.3.71 获取当前用户坐标 ID

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 USERCOORD）

int EstunRemoteApiLib::E_GetCurUserCoordID (E_UserCoordType & coordtype)

获取当前用户坐标号.

参数

coordtype	用户坐标类型，引用，类型说明详见 6.2.16

返回

用户坐标号

6.3.72 获取用户坐标变量名称

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 USERCOORD）

ESString E_GetUserCoordName(int coordID, E_UserCoordType coordtype)

获取用户坐标变量名称.

参数

coordID	用户坐标 id
coordtype	用户坐标类型，引用，类型说明详见 6.2.16

返回

用户坐标名称，为空则表示获取失败

6.3.73 获取用户坐标参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 USERCOORD）

bool EstunRemoteApiLib::E_GetUserCoord (int coordID, E_UserCoordType coordtype, E_ROB_COORD & coord)

获取用户坐标信息.

参数

coordID	用户坐标 id
coordtype	用户坐标类型，类型说明详见 6.2.16
coord	用户坐标参数，引用，类型说明详见 6.2.15

返回

成功返回 true，失败返回 false

6.3.74 获取负载 ID

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

bool E_GetPayLoadsID(int& num, int*& idArray)

获取所有负载 ID.

参数

num	获取到的个数
idArray	ID 数组指针，由接口初始化

返回

成功返回 true，失败返回 false

6.3.75 新建负载参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

bool EstunRemoteApiLib::E_CreatePayLoad (int payloadID, const E_ROB_PAYLOAD & payloadData)

创建负载.

参数

payloadID	新的负载号，必须大于 0
payloadData	负载参数，类型说明详见 6.2.14

返回

成功返回 true，失败返回 false

注

调用该接口创建负载变量时，若 payloadID 不存在则创建负载号为 payload 负载变量，若 payloadID 已存在则修改内存中的值。

6.3.76 修改负载参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

bool EstunRemoteApiLib::E_ModifyPayLoad (int payloadID, const E_ROB_PAYLOAD & payloadData)

修改负载.

参数

payloadID	要修改的负载号，必须大于 0
payloadData	负载参数，类型说明详见 6.2.14

返回

成功返回 true，失败返回 false

注

调用该接口修改负载变量时只修改内存中的值。

6.3.77 选择负载参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

bool EstunRemoteApiLib::E_LoadPayLoad (int payloadID)

加载负载.

参数

payloadID	负载号

返回

成功返回 true，失败返回 false

6.3.78 获取当前负载 ID

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

int EstunRemoteApiLib::E_GetCurPayLoadID ()

获取当前负载变量 id 号.

返回

负载号

6.3.79 获取负载变量名称

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

ESString E_GetPayLoadName(int payloadID)

获取负载变量名称.

参数

payloadID	负载变量 id 号

返回

变量负载名称，为空则表示获取失败

6.3.80 获取负载参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

bool EstunRemoteApiLib::E_GetPayLoad (int payloadID, E_ROB_PAYLOAD & payload)

获取负载变量.

参数

payloadID	负载变量 id 号
payload	负载参数，引用

返回

成功返回 true，失败返回 false

6.3.81 获取默认加载的负载参数

（含义见《ER 系列机器人操作手册》 10.3 系统数据类型 PAYLOAD）

bool E_GetDefaultPayload(int& payloadID, ESString& name, E_ROB_PAYLOAD& payload)

获取默认加载的负载参数.

参数

payloadID	负载变量 id 号 ， 引用

name	负载名称，引用
payload	负载参数，引用

返回

成功返回 true，失败返回 false

6.3.82 设置 IO 的值

（含义见《ER 系列机器人操作手册》 10.4 IO 数据类型）

bool EstunRemoteApiLib::E_SetIOValue (int IoIndex, E_IOType IoType, double* dataIn, int dataNum)

设置 IO 的值.

参数

IoIndex	开始的 IO 下标
IoType	IO 类型，类型说明详见 6.2.17
dataIn	要修改的 IO 值数组指针
dataNum	要修改的 IO 值数组元素个数

返回

成功返回 true，失败返回 false

注

通过此接口设置 IO 效果为强制设置的效果。

6.3.83 获取 IO 的值

（含义见《ER 系列机器人操作手册》 10.4 IO 数据类型）

bool EstunRemoteApiLib::E_GetIOValue (int IoIndex, E_IOType IoType, int IoNum, double* dataOut)

获取 IO 的值.

参数

IoIndex	开始的 IO 下标
IoType	IO 类型，类型说明详见 6.2.17
IoNum	查询 IO 数量
dataOut	从【IoIndex】开始的【IoNum】个的 IO 的值数组，数组指针

返回

成功返回 true，失败返回 false

6.3.84 设置 DO 的值

（含义见《ER 系列机器人操作手册》 10.4 IO 数据类型）

bool EstunRemoteApiLib::E_SetMultiDO (int* IoValues, int number)

设置 DO 的值.

参数

IoValues	IO 值数组指针，+/-代表 1/0，数值代表 DO 下标,不同 DO 之间使用_分割，例：{-1_2_6_-8}，表示将 DO1、DO8 设置为 0，DO2、DO6 设置为 1
number	IoValues IO 值数组个数

返回

成功返回 true，失败返回 false

注

通过此接口设置 IO 效果为强制设置的效果。

6.3.85 设置变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_SetVarValue (E_VarType varType, ESString varName, ESString varValue, E_ScopeType varScope)

设置变量.

参数

varType	变量类型，类型说明详见 6.2.18
varName	变量名，类型说明详见 6.2.1
varValue	变量值，字符串类型，需要从其他类型转换，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19

返回

成功返回 true，失败返回 false

注

调用该接口设置变量时，若 varScope 中不存在 varType 类型的 varName 变量则在 varScope 中创建变量名为 varName 的 varType 变量，若 varScope 中存在 varType 类型的 varName 变量则修改内存中的值。

6.3.86 获取变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_GetVarValue (E_VarType varType, ESString varName, E_ScopeType varScope, ESString & valueBack)

获取变量.

参数

varType	变量类型，类型说明详见 6.2.18
varName	变量名，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
valueBack	变量值，字符串类型，引用类型，类型说明详见 6.2.1

返回

成功返回 true，失败返回 false

6.3.87 获取整形变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_ReadInt (ESString varName, E_ScopeType varScope, int & valueBack)

读取 Int 型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
valueBack	变量值，引用类型

返回

成功返回 true，失败返回 false

6.3.88 设置整形变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_WriteInt (ESString varName, E_ScopeType varScope, int value)

写入 Int 型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
value	变量值

返回

成功返回 true，失败返回 false

注

调用该接口设置整形变量时，若 varScope 中不存在 varName 变量则在 varScope 中创建变量名为 varName 的变量，若 varScope 中已存在 varName 变量则修改内存中的值。

6.3.89 获取实数变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_ReadReal (ESString varName, E_ScopeType varScope, double & valueBack)

读取 Double 型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
valueBack	变量值，引用类型

返回

成功返回 true，失败返回 false

6.3.90 设置实数变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_WriteReal (ESString varName, E_ScopeType varScope, double value)

写入 Double 型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
value	变量值

返回

成功返回 true，失败返回 false

注

调用该接口设置实数变量时，若 varScope 中不存在 varName 变量则在 varScope 中创建变量名为 varName 的变量，若 varScope 中已存在 varName 变量则修改内存中的值。

6.3.91 获取字符变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_ReadStr (ESString varName, E_ScopeType varScope, ESString & valueBack)

读取 String 型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
valueBack	变量值，引用类型，类型说明详见 6.2.1

返回

成功返回 true，失败返回 false

6.3.92 设置字符变量

（含义见《ER 系列机器人操作手册》 10.1 基本数据类型）

bool EstunRemoteApiLib::E_WriteStr (ESString varName, E_ScopeType varScope, ESString value)

写入 String 型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
value	变量值，类型说明详见 6.2.1

返回

成功返回 true，失败返回 false

注

调用该接口设置字符变量时，若 varScope 中不存在 varName 变量则在 varScope 中创建变量名为 varName 的变量，若 varScope 中已存在 varName 变量则修改内存中的值。

6.3.93 获取位置变量

（含义见《ER 系列机器人操作手册》 10.2 位置数据类型）

bool EstunRemoteApiLib::E_ReadPos (ESString varName, E_ScopeType varScope, E_ROB_POS & valueBack)

读取位置类型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
valueBack	变量值，引用类型，类型说明详见 6.2.7

返回

成功返回 true，失败返回 false

6.3.94 设置位置变量

（含义见《ER 系列机器人操作手册》 10.2 位置数据类型）

bool EstunRemoteApiLib::E_WritePos (ESString varName, E_ScopeType varScope, E_ROB_POS value)

写入位置类型变量值.

参数

varName	变量名称，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
value	变量值，类型说明详见 6.2.7

返回

成功返回 true，失败返回 false

注

调用该接口设置位置变量时，若 varScope 中不存在 varName 变量则在 varScope 中创建变量名为 varName 的变量，若 varScope 中已存在 varName 变量则修改内存中的值。

6.3.95 示教点

（含义见《ER 系列机器人操作手册》 10.2 位置数据类型）

bool EstunRemoteApiLib::E_TeachPoint (ESString pointName, E_ScopeType scope)

示教点.

参数

pointName	位置点变量名，类型说明详见 6.2.1
scope	变量域，不能是系统域，类型说明详见 6.2.19

返回

成功返回 true，失败返回 false

6.3.96 获取当前程序

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_GetCurProgramName (E_ROB_PROGNAME & value)

获取当前加载的程序名.

参数

value	工程程序名，引用类型，类型说明详见 6.2.20

返回

成功返回 true，失败返回 false

6.3.97 获取程序状态

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

E_RunStatusType EstunRemoteApiLib::E_GetCurProgramRunStatus ()

获取当前程序运行状态.

返回

程序运行状态，类型说明详见 6.2.21

6.3.98 卸载工程

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_UnLoadProject ()

卸载工程.

返回

成功返回 true，失败返回 false

6.3.99 卸载程序

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_UnLoadProgram ()

卸载程序.

返回

成功返回 true，失败返回 false

6.3.100 加载工程

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_LoadProject (ESString prjname)

只加载工程.

参数

prjname	工程名，类型说明详见 6.2.1

返回

成功返回 true，失败返回 false

6.3.101 加载程序

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_LoadProgame (ESString prjname, ESString programname)

加载程序.

参数

prjname	工程名，类型说明详见 6.2.1
programname	程序名，类型说明详见 6.2.1

返回

成功返回 true，失败返回 false

6.3.102 运行程序

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_ProgRun ()

运行程序.

返回

成功返回 true，失败返回 false

6.3.103 暂停程序

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_ProgPause ()

暂停程序.

返回

成功返回 true，失败返回 false

6.3.104 停止程序

（含义见《ER 系列机器人操作手册》 第 9 章系统功能中有关工程的介绍）

bool EstunRemoteApiLib::E_ProgStop ()

停止程序.

返回

成功返回 true，失败返回 false

6.3.105 执行 MovJ2 数组指令

E_ROB_JOBID E_MovJ2_Trajectory(E_ROB_POS* dstPointArray, int posNums, const E_ROB_SPEED& velo = E_ROB_SPEED(), const E_ROB_ZONE& zone = E_ROB_ZONE(), int payloadId = 0, int toolId = 0, int coordId = 0, const E_ROB_APIBUFF& addDo = E_ROB_APIBUFF(), double valfine = 0.0, bool IsWaitFinish = false)

执行 MovJ2 数组指令.

参数

dstPointArray	目标位置点,轨迹点数组，类型说明详见 6.2.7
posNums	轨迹点个数
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12
payloadId	负载号
toolId	工具号
coordId	用户坐标号
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败

6.3.106 执行 MovL2 指令数组

E_ROB_JOBID E_MovL2_Trajectory(E_ROB_POS* dstPointArray, int posNums, const E_ROB_SPEED& velo = E_ROB_SPEED(), const E_ROB_ZONE& zone = E_ROB_ZONE(), int payloadId = 0, int toolId = 0, int coordId = 0, int overrideType = 1, const E_ROB_APIBUFF& addDo = E_ROB_APIBUFF(), double valfine = 0.0, bool IsWaitFinish = false)

执行 MovL2 数组指令.

参数

dstPointArray	目标位置点,轨迹点数组，类型说明详见 6.2.7
posNums	轨迹点个数
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12

payloadId	负载号
toolId	工具号
coordId	用户坐标号
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败

6.3.107 获取运动数组执行状态

int E_GetTrajectoryCmdSts(E_ROB_JOBID posArrayJobID)

获取指定运动数组指令执行状态.

参数

posArrayJobID	指定的轨迹 id 信息

返回

-2-错误 -1-未找到，0-运动未完成，1-运动完成,2-暂停，3-结束/运动错误，4-(预留类型)

6.3.108 获取运动数组执行状态

bool E_GetTrajectoryPointsJobId(E_ROB_JOBID posArrayJobID, E_ROB_JOBID* posAllJobIDBuff,int buffNums)

获取指定运动数组指令执行状态.

参数

posArrayJobID	指定的轨迹 id 信息
posAllJobIDBuff	所有当前轨迹中所有点的 JobId 的 buff 缓存
buffNums	缓存的数量

返回

成功返回 true，失败返回 false

6.3.109 笛卡尔坐标转换关节坐标

bool E_CposToAPos(double Cpos[13], double Apos[6])

笛卡尔坐标转关节坐标.

参数

Cpos	笛卡尔坐标，类型说明详见 6.2.7
Apos	关节坐标，类型说明详见 6.2.7

返回

成功返回 true，失败返回 false

6.3.110 关节坐标转换笛卡尔坐标

bool E_AposToCPos(double Apos[6], double Cpos[13]);

关节坐标转笛卡尔坐标.

参数

Cpos	笛卡尔坐标，类型说明详见 6.2.7
Apos	关节坐标，类型说明详见 6.2.7

返回

成功返回 true，失败返回 false

6.3.111 获得机器人当前线速度

int E_GetCurTcpVelo()

获得机器人当前线速度.

返回

机器人线速度

6.3.112 设置点位数组

bool E_SetPosArrayValue(E_VarType varType, ESString varName, E_ScopeType varScope, int startIndex, int nums, E_ROB_POS* dataIn)

设置点位数组.

参数

varType	变量类型 只可以是 CposArray 或 AposArray，类型说明详见 6.2.18
varName	变量名，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
startIndex	指定存储的起始下标
nums	指定存储的个数
dataIn	存储的点位数组

返回

成功返回 true，失败返回 false

注

调用该接口设置点位数组时，若 varScope 中不存在 varType 类型的 varName 变量则在 varScope 中创建 varType 类型的 varName 的变量，若 varScope 中已存在 varType 类型的 varName 变量则修改内存中的值。

6.3.113 获取点位数组

bool E_GetPosArrayValue(E_VarType varType, ESString varName, E_ScopeType varScope, int startIndex, int nums, E_ROB_POS* dataOut)

获取点位数组.

参数

varType	变量类型 只可以是 CposArray 或 AposArray,，类型说明详见 6.2.18
varName	变量名，类型说明详见 6.2.1
varScope	变量域，类型说明详见 6.2.19
startIndex	指定存储的起始下标
nums	指定存储的个数
dataOut	返回的点位数组

返回

成功返回 true，失败返回 false

6.3.114 获取力矩的值

bool E_GetAxeTorqueValue(double* dataOut)

获取力矩的值.

参数

dataOut	力矩值的存储位置，16 个 double 的内存

返回

成功返回 true，失败返回 false

6.3.115 设置 PSO 控制字

bool E_Set_LatchEnable(int AxleID, bool EnableSts);

设置 PSO 控制字使能与禁能指令

参数

AxleID	轴号
EnableSts	使能状态 true 上使能 false 下使能

返回

成功返回 true，失败返回 false

6.3.116 获取 PSO 状态字的值

int E_Get_PSOStatus(int AxleID);

获取 PSO 状态字的值

参数

AxleID	轴号

返回

PSO 状态字的值

6.3.117 设置指定轴的指定 PSO 比较点指令

bool E_Set_LatchPso(int AxleID, int number, double Apos[16]);

设置指定轴的指定 PSO 比较点指令

参数

AxleID	轴号
number	比较点号，（1-8），比较点有效或者无效通过示教器进行 Pn608 参数设置，详情见机器人操作手册
Apos	APOS 的位置值

返回

成功返回 true，失败返回 false

6.3.118 MovLOffset

（含义见《ER 系列机器人操作手册》 6.1 节运动指令 MovLOffset）

E_ROB_JOBID E_MovLOffset(const E_ROB_POS& dstPoint, const E_ROB_POS& relativePoint, int RefSys, const E_ROB_SPEED& velo = E_ROB_SPEED(), const E_ROB_ZONE& zone = E_ROB_ZONE(), int payloadId = 0, int toolId = 0, int coordId = 0, int overrideType = 1, const E_ROB_APIBUFF& addDo = E_ROB_APIBUFF(), double valfine = 0.0, bool IsWaitFinish = false);

执行 MovLOffset 指令.

参数

dstPoint	目标位置点，类型说明详见 6.2.7
relativePoint	相对偏移位置点，类型说明详见 6.2.7
RefSys	坐标系类型 0：coord； 1：tool
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12
payloadId	负载号
toolId	工具号
coordId	用户坐标号
overrideType	速率类型, 99:GOVROFF（全局速度不生效）;非 99:GOVRON（全局速度生效）
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

6.3.119 MovJOffset

（含义见《ER 系列机器人操作手册》 6.1 节运动指令 MovJOffset）

E_ROB_JOBID E_MovJOffset(E_ROB_POS& dstPoint, E_ROB_POS& relativePoint, int RefSys, const E_ROB_SPEED& velo = E_ROB_SPEED(), const E_ROB_ZONE& zone = E_ROB_ZONE(), int payloadId = 0, int toolId = 0, int coordId = 0, const E_ROB_APIBUFF& addDo = E_ROB_APIBUFF(), double valfine = 0.0, bool IsWaitFinish = false);

执行 MovJOffset 指令.

参数

dstPoint	目标位置点，类型说明详见 6.2.7
relativePoint	相对偏移位置点，类型说明详见 6.2.7
RefSys	坐标系类型 0：coord； 1：tool
velo	运行速度，类型说明详见 6.2.11
zone	过度参数，类型说明详见 6.2.12
payloadId	负载号
toolId	工具号
coordId	用户坐标号
addDo	附加指令，内容由其他结构体填充，需要调用 AddDoTypeConvert 的相关接口完成转换，目前仅支持 E_ROB_ADDDO_IO 类型，类型说明详见 6.2.8
valfine	到位精度值
IsWaitFinish	是否等运动执行结束，为 true 则接口在运动执行到位后再返回，为 false 则立即返回

返回

指令标识信息，id 小于 0 为失败，类型说明详见 6.2.13

附录一：错误列表

错误号	错误信息	处理方法
88801	APIServer 断开连接	重新启动 APIServer 程序
88802	执行运动指令中,不允许设置 JOG 使能	先停止当前运动，在设置 JOG 使能
88803	JOG 使能中,不可以执行运动指令	先取消 JOG 使能，然后调用运动接口
88804	执行运动指令中,不允许设置 Mot 使能	先停止当前运动，在设置 Mot 使能
88805	API 组件初始化失败	重新启动控制器
88806	当前 TOOL 加载中,不允许修改	先卸载当前 TOOL（即加载 nulltool（id 为 0)），再进行修改
88807	当前 UserCoord 加载中,不允许修改	先卸载当前 UserCoord（即加载 world（id 为 0)），再进行修改
88808	当前 PayLoad 加载中,不允许修改	先卸载当前 PayLoad（即加载 Full PayLoad（id 为 0)），再进行修改
88809	当前 TOOL 不存在,修改失败	先创建对应 id 的 TOOL 变量，再进行修改
88810	当前 UserCoord 不存在,修改失败	先创建对应 id 的 UserCoord 变量，再进行修改
88811	当前 PayLoad 不存在,修改失败	先创建对应 id 的 PayLoad 变量，再进行修改
88812	当前 TOOL 不存在,加载失败	选择加载存在的 TOOL
88813	当前 UserCoord 不存在,加载失败	选择加载存在的 UserCoord
88814	当前 PayLoad 不存在,加载失败	选择加载存在的 PayLoad
88815	机器人运动中,不允许加载	先停止当前运动，再加载 Tool 或 UserCoord 或 PayLoad
88816	当前机器人不在暂停状态,继续运动失败	重新发送运动指令