

EPSON

AX Portal 用户手册

翻译版

©Seiko Epson Corporation 2026

Rev. 2
SCM265S8835M

目录

商标.....	8
使用条款.....	8
1. 简介.....	9
1.1 编程与控制器.....	10
1.2 图形用户界面 (GUI).....	10
1.3 软件模块.....	10
1.3.1 TCP连接模块.....	10
1.3.2 REST连接模块.....	10
1.3.3 ROS连接模块.....	10
1.3.4 Modbus连接模块.....	11
1.3.5 面板界面模块.....	11
1.3.6 Cognex 摄像头模块.....	11
2. 管理员面板 (GUI).....	12
2.1 简介.....	13
2.1.1 界面语言.....	14
2.1.2 仪表板.....	14
2.2 用户管理.....	14
2.2.1 用户等级.....	14
2.2.2 登陆页面.....	14
2.2.3 账户管理.....	15
2.2.4 管理其他用户.....	16
2.3 系统设置.....	16
2.3.1 网络设置.....	16
2.3.2 时间设置.....	18
2.3.3 SSL证书.....	18
2.4 软件更新.....	20
2.5 帮助 & 支持.....	22
2.6 安全认证.....	23
2.6.1 出口 SSL证书.....	23
2.6.2 用户登陆.....	23
2.6.3 连接注册.....	24
3. USB 出口和恢复更新.....	25
3.1 概述.....	26
3.2 USB 出口.....	26
3.3 运行中恢复更新.....	26
4. 桌面界面 (GUI).....	27
4.1 简介.....	28
4.2 主窗口.....	29
4.2.1 菜单概述.....	29
4.2.2 控制台.....	30
4.2.3 活动控制 (单点控制器).....	32
4.2.3.1 活动控制站.....	32

4.2.3.2 被动观察者.....	33
4.2.4 程序状态.....	33
4.2.5 操作模式.....	35
4.2.5.1 自动操作模式.....	35
4.2.5.2 手动操作模式.....	35
4.2.6 精细控制.....	36
4.2.7 运动控制.....	37
4.2.7.1 运动限制.....	37
4.2.7.2 全局参数.....	38
4.2.7.3 全局坐标系.....	39
4.3 构建模块与数据库.....	40
4.4 项目菜单.....	40
4.5 配置菜单.....	42
4.5.1 数据库(下载与上传)	42
4.5.2 I/O 配置.....	42
4.5.3 工具配置.....	46
4.5.4 工具编辑器.....	46
4.5.4.1 单输出模式.....	48
4.5.4.2 双输出模式.....	48
4.5.4.3 分离触发模式.....	48
4.5.5 机器人上电.....	49
4.6 示教菜单.....	50
4.6.1 位姿编辑器.....	52
4.6.2 路径编辑器.....	54
4.6.3 网格编辑器.....	56
4.7 应用程序菜单.....	59
4.7.1 程序编辑器.....	60
4.7.2 应用程序代码.....	60
4.7.3 程序设置.....	61
4.7.4 应用程序变量.....	63
4.7.5 共享全局变量.....	64
4.7.6 文件夹编辑器.....	65
4.7.7 模块编程.....	66
4.7.7.1 使用程序变量和表达式.....	66
4.7.7.2 移动模块.....	67
4.7.7.3 路径块.....	68
4.7.7.4 拾取模块.....	68
4.7.7.5 放置模块.....	68
4.7.7.6 容器模块.....	68
4.7.7.7 下一个网格位姿模块.....	68
4.7.7.8 重置网格位姿块.....	69
4.7.7.9 设置变量模块.....	69
4.7.7.10 设置输出模块.....	69
4.7.7.11 等待模块.....	69

4.7.7.12 循环模块.....	69
4.7.7.13 “退出, 当满足条件时”模块.....	69
4.7.7.14 条件模块.....	69
4.7.7.15 注释模块.....	69
4.7.7.16 退出模块.....	69
4.7.7.17 对话框模块.....	70
4.7.7.18 打印模块.....	70
4.7.7.19 日志模块.....	70
4.7.7.20 通知模块.....	70
4.7.7.21 警告模块.....	70
4.7.7.22 错误模块.....	70
4.7.7.23 Python 代码模块.....	70
4.8 I/O 菜单.....	70
4.9 可视化模型菜单.....	72
4.10 使能控制.....	73
4.11 系统监视器.....	74
4.11.1 机械臂状态.....	74
4.11.2 程序输出.....	74
4.11.3 程序日志.....	74
4.11.4 历史消息.....	74
4.12 帮助菜单.....	75
4.12.1 机器人信息.....	75
4.12.2 快捷键.....	75
4.12.3 代码文档.....	75
4.12.4 开源许可证.....	75
4.12.5 联系方式.....	75
4.13 设置菜单.....	76
4.13.1 界面设置.....	77
4.13.2 自定义导航面板按钮.....	77
4.13.3 启动设置.....	78
4.13.4 运动行为.....	78
4.13.4.1 中断状态行为.....	78
4.13.4.2 中断对话框.....	79
4.14 安全设置菜单.....	81
4.14.1 用户界面.....	81
4.14.2 安全设置.....	83
4.14.2.1 安全限位 (SLP).....	83
4.14.2.2 安全极限关节角度.....	84
4.14.2.3 安全限速 (SLS).....	85
4.14.2.4 安全功率与力限制.....	86
4.14.2.5 安全 I/O 和操作员限制.....	87
4.14.2.6 验证数据.....	88
4.14.2.7 安全输出功能.....	88
4.14.2.8 连接可选设备.....	88

5. 面板界面 (GUI).....	89
5.1 简介.....	89
5.2 配置面板界面的应用程序.....	91
5.3 控制台与设置.....	93
5.4 程序控制.....	94
6. 运动学定义.....	95
6.1 坐标.....	96
6.1.1 航海角.....	96
6.1.2 坐标系.....	97
6.2 配置.....	98
6.3 位姿.....	98
6.4 点.....	98
6.5 路径.....	98
6.6 路径段.....	98
6.7 网格.....	99
7. 连接模块.....	100
7.1 连接、身份验证与活动控制.....	101
7.1.1 连接通道.....	101
7.1.2 身份验证.....	101
7.1.3 活动控制站.....	102
7.2 TCP连接模块.....	102
7.2.1 建立连接.....	102
7.2.2 发送命令.....	103
7.2.3 设置.....	104
7.2.3.1 编码标准.....	104
7.2.3.2 结束字符.....	104
7.2.3.3 发送心跳包.....	104
7.2.4 消息协议 - 接收命令.....	105
7.2.4.1 状态信息.....	105
7.2.4.2 执行返回值.....	106
7.2.4.3 错误.....	106
7.2.4.4 历史消息.....	106
7.2.4.5 状态群组定义.....	106
7.2.4.6 状态群组数据.....	107
7.2.4.7 打开对话框.....	108
7.2.4.8 脚本发送.....	108
7.2.5 消息协议 - 发送命令.....	109
7.2.5.1 令牌登陆.....	109
7.2.5.2 请求活动控制.....	109
7.2.5.3 释放活动控制.....	110
7.2.5.4 请求数据.....	110
7.2.5.5 上电.....	110
7.2.5.6 休眠.....	110
7.2.5.7 暂停.....	111

7.2.5.8 继续运行.....	111
7.2.5.9 释放使能控制模式 (HGC).....	112
7.2.5.10 保持使能控制 (HGC).....	112
7.2.5.11 停止.....	113
7.2.5.12 切换项目.....	113
7.2.5.13 执行脚本.....	114
7.2.5.14 测试脚本.....	114
7.2.5.15 脚本接收.....	115
7.2.5.16 运动位姿.....	115
7.2.5.17 执行路径.....	116
7.2.5.18 切换工具.....	116
7.2.5.19 写入数字与模拟输出.....	117
7.2.5.20 对话框响应.....	118
7.2.5.21 设置状态变量.....	118
7.2.5.22 设置全局变量.....	119
7.2.5.23 获取全局变量.....	119
7.2.5.24 设置共享变量.....	120
7.2.5.25 获取共享变量.....	120
7.3 REST连接模块.....	121
7.3.1 建立连接.....	121
7.3.2 发送命令.....	122
7.3.3 设置.....	122
7.4 ROS 处理程序模块.....	123
7.4.1 安装.....	123
7.4.2 配置.....	124
7.4.3 使用 Cyclone DDS 进行网络发现.....	124
7.4.3.1 ROS 网络接口.....	124
7.4.3.2 发现方法.....	124
7.4.4 建立连接.....	126
7.4.5 通过 ROS 访问机器人数据.....	127
7.4.6 脚本中的 ROS 2 功能.....	127
7.4.7 执行操作.....	127
7.4.8 验证设置.....	128
7.5 Modbus连接模块.....	128
7.5.1 概述.....	128
7.5.2 设置.....	128
7.5.3 写入寄存器(简单描述).....	130
7.5.3.1 命令寄存器.....	130
7.5.3.2 属性寄存器.....	130
7.5.3.3 用户写入寄存器.....	131
7.5.4 写入寄存器(详细描述).....	132
7.5.4.1 不执行任何操作.....	132
7.5.4.2 通电.....	132
7.5.4.3 休眠.....	132

7.5.4.4 停止.....	132
7.5.4.5 暂停.....	132
7.5.4.6 继续运行.....	133
7.5.4.7 校准.....	133
7.5.4.8 释放使能控制模式 (HGC).....	133
7.5.4.9 保持使能控制 (HGC).....	133
7.5.4.10 切换项目.....	133
7.5.4.11 开始脚本.....	134
7.5.4.12 移动到位姿.....	134
7.5.4.13 执行路径.....	134
7.5.4.14 工具放置.....	134
7.5.4.15 工具拾取.....	134
7.5.4.16 写入数字输出.....	135
7.5.4.17 设置脚本事件.....	135
7.5.4.18 确认错误.....	135
7.5.4.19 读取位置.....	135
7.5.4.20 请求活动控制.....	135
7.5.4.21 释放活动控制.....	136
7.5.4.22 切换到手动模式.....	136
7.5.4.23 切换到自动模式.....	136
7.5.4.24 安全重置.....	136
7.5.4.25 开始运动中断.....	136
7.5.5 读取寄存器.....	137
7.5.5.1 状态寄存器.....	137
7.5.5.2 用户读取寄存器.....	140
7.5.5.3 位置寄存器(第一关节至第六关节).....	140
7.5.5.4 坐标寄存器(x, y, z, X轴旋转, Y轴旋转, 俯仰).....	140
7.5.6 错误寄存器映射.....	141
7.5.6.1 示例.....	141
7.5.7 Python 代码示例.....	142
7.5.7.1 示例 1.....	142
7.5.7.2 示例 2.....	142
7.5.7.3 示例 3.....	142
7.5.7.4 示例 4.....	143
8. Cognex 摄像头模块.....	144
8.1 概述.....	145
8.2 摄像头设置.....	145
8.3 摄像头程序.....	145

商标

Cognex® 和 In-Sight Explorer® 是 Cognex Corporation 在美国和其他国家/地区的商标或注册商标。

Google Chrome™ 和 Chrome™ 是 Google LLC 的商标。

Linux® 是 Linus Torvalds 在美国和其他国家/地区的注册商标。

Apple®, macOS® 和 Safari® 是 Apple Inc. 在美国和其他国家及地区的注册商标。

Microsoft®, Windows®, Microsoft Edge® 和 Microsoft Visual C++® 是微软公司和/或其附属公司在美国和其他国家的商标(或注册商标)。

Mozilla 和 Firefox® 是 Mozilla Foundation 在美国和其他国家/地区的商标。

一般声明:本文中使用的其他产品名称仅用于识别目的,可能是其各自所有者的商标Epson 不主张对这些商标拥有任何权利

所有屏幕截图仅用于说明和培训目的,无意亦不暗示与任何商标持有人存在关联或获得其认可

使用条款

未经明确书面许可,不得以任何形式复制或翻印本说明手册的任何部分

本档中的信息如有更改,恕不另行通知

如果您发现本档中存在任何错误,或对其中的信息有任何疑问,请与我们联系

1. 简介

1.1 编程与控制器

AX Portal 软件框架为 Epson AX 系列机器人的用户提供了一种非常直观的与该机器人通信的方式。使用 AX Portal, 用户可以完全访问并操作和控制机器人, 以及使用 **Python 3.13** 编程语言编写程序应用程序。有关使用 Python 编程的基本功能, 可在线查阅。不过, 内置代码文档中也提供了一些非常基础的代码示例。

除了种类繁多的基础 Python 函数外, AX Portal 程序还支持大量专门为控制、维护与 AX 机器人通信而设计的函数。AX Portal 支持的脚本函数完整列表可在内置代码文档中查阅。

1.2 图形用户界面 (GUI)

该 GUI 设计为 Web 应用程序, 可在几乎任何设备的常用 Web 浏览器上运行。官方支持以下浏览器:

- Google Chrome (版本 ≥ 147)
- Mozilla Firefox (版本 ≥ 150)
- Apple Safari (版本 ≥ 26)
- Microsoft Edge (版本 ≥ 147)

因此, 只要机器人与设备处于同一网络或共享直接通信通道, 您无需为控制设备下载特定的软件模块。除了主要针对台式机和笔记本电脑编程优化的 AX Portal 默认桌面界面外, AX Portal 还提供面板界面, 该界面仅包含操作已预编程机器人的必要功能。您可以在本文档下文中找到有关这些图形用户界面及其所有支持功能的更多信息。

1.3 软件模块

除了本文档中详细说明了 AX Portal 基础功能外, AX 机器人还提供了一些具有额外控制功能的内置软件模块。请注意, 下文提到的所有连接模块都提供了远程连接 AX 机器人的功能, 从而实现了机器人的控制和监控。

1.3.1 TCP连接模块

TCP连接模块可作为与机器人通信的替代方案。通常, 用户通过网页浏览器中的 GUI 控制机器人。但是, 如果需要通过自定义界面或另一台机器控制机器人, 它会直接接受 TCP 命令。借助 TCP连接模块, 浏览器能实现的所有功能, 均可通过 TCP 实现。

1.3.2 REST连接模块

REST连接模块可作为与机器人通信的另一种方式。通常, 用户通过网站浏览器中的图形用户界面(GUI)控制机器人。但是, 如果机器人需要由自定义界面或另一台机器控制, 它将直接接受 REST 命令。借助 REST连接模块, 浏览器能实现的所有功能, 均可通过 REST 实现。

1.3.3 ROS连接模块

ROS连接模块提供了将机器人集成到 ROS2 环境中的功能。该机器人允许 ROS2 网络中的其他客户端发送操作指令, 以控制和监控机器人。

1.3.4 Modbus连接模块

Modbus连接模块可作为与机器人通信的替代方案。它采用 Modbus TCP 协议, 允许用户读写标准 Modbus 寄存器。使用此模块可通过 PLC 直接控制机器人。诸如给机器人供电、运行程序和触发输出等最重要功能均可通过 Modbus 模块实现。

1.3.5 面板界面模块

面板界面模块为机器人操作员(而非程序员)提供了一个功能有限的图形用户界面。该界面主要用于启动和监控应用程序, 其内容可由程序员根据具体应用场景进行修改。面板界面支持触摸屏操作, 其设计借鉴了现代移动设备中广为人知的应用程序式结构, 即使对于没有技术背景的用户来说也直观易用。

1.3.6 Cognex 摄像头模块

康耐视相机模块为康耐视相机与AX Portal之间提供通信接口。相机通常集成在末端执行器中, 或固定安装以观察机器人作业场景。该模块非常适合可定制的视觉检测, 支持康耐视相机的全部功能范围。

2. 管理员面板 (GUI)

2.1 简介

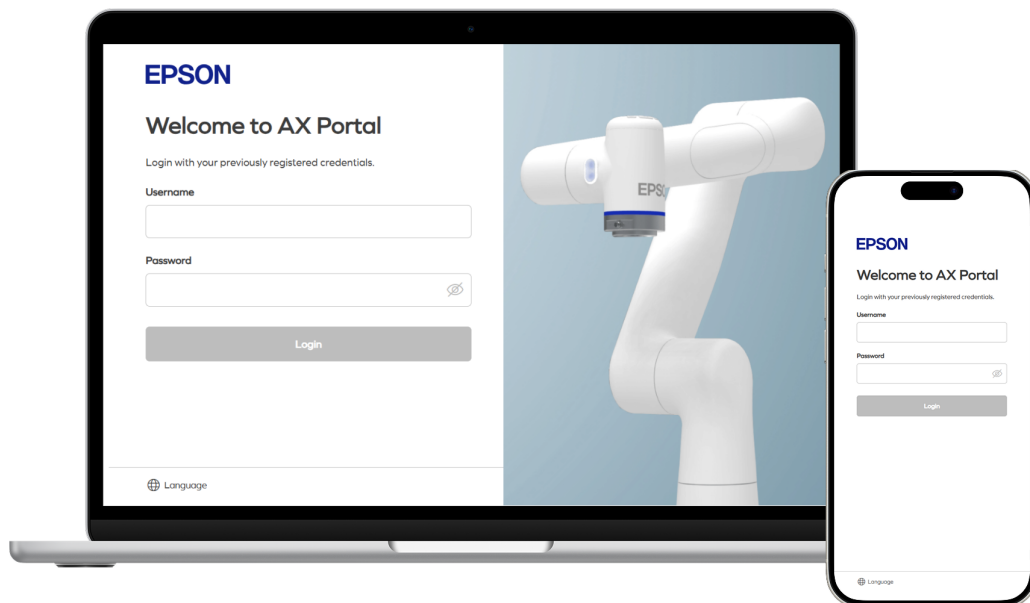


图 2.1.a: 管理员面板的示例截图

AX Portal 的管理员面板是一个图形用户界面 (GUI)，允许用户管理用户账户，并更改与机器人运行操作系统相关的初始设置 (如网络设置、时间设置等)。这些设置通常在机器人安装时仅需设置一次，甚至在程序员开始编写机器人控制应用程序之前就已完成。可通过在网页浏览器 (如 Google Chrome) 的地址栏中输入机器人的 IP 地址访问该界面，系统将自动跳转至 `https://<IP-ADDRESS>/admin` (正确的 IP 地址请参阅《机器人控制器 RC-A101 手册》)。

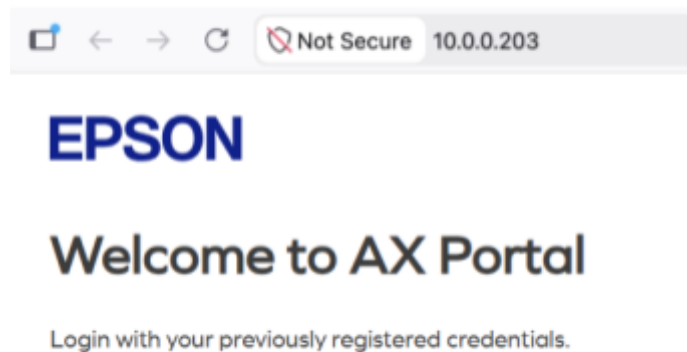


图 2.1.b: 通过 URL 连接到管理员面板

本章将详细介绍管理员面板的全部菜单和元素。

2.1.1 界面语言

要更改界面语言，请点击界面左下角的“语言”下拉菜单，并选择您首选的语言。您的浏览器会记住您在不同会话和界面中的语言设置，从而为您带来便利。

2.1.2 仪表板

登录后，管理员面板的仪表盘可让您访问不同的子菜单、导航至 AX Portal 的其他界面，以及注销。请注意，仪表盘仅显示您当前可访问的选项，具体取决于您的用户等级（管理员、安全主管、程序员、操作员）。若无法导航至其他界面，则表明 AX Portal 当前运行异常，例如在软件更新中即会出现此情况。

2.2 用户管理

2.2.1 用户等级

为防止未经授权的用户配置、编程或操作机器人，我们引入了不同的认证级别。任何用户的用户等级均由管理员定义。可用的用户等级如下（按认证级别从高到低排序）：

- 管理员可以全面管理用户、机器人设置以及所有机器人接口，并访问操作系统的关键系统设置。
- 程序员可访问所有机器人控制界面和功能，因此能够为操作员准备并编程应用程序以便后续执行。在管理员面板中，程序员基本上仅能访问其自身账户详情，以及在各界面之间进行导航。
- 安全主管是具有修改机器人安全设置权限的程序员（参见上文）。其他用户仅可查看安全设置。
- 操作员仅可访问基础操作界面，该界面允许操作机器人，但无法更改任何设置或程序。在管理员面板中，操作员基本上仅能访问其账户详情。

2.2.2 登陆页面

登录页面包含两种不同的表单，具体取决于是否已创建管理员账户。

- 如果已创建管理员，则显示常规登陆页面，您需要输入用户名和密码进行登陆。输入密码时，密码会隐为星号，除非您点击密码输入框内的“显示密码”按钮。
- 如果尚未创建管理员，系统会直接提示您进行创建，且必须先创建管理员账户才能继续操作。详情请参阅《六轴协作机器人 AX6: 快速入门指南 / 安装手册》。此处仅可配置绝对必要的数据；若需查看并设置账户的所有选项，请访问“账户管理”菜单。

每次注销后，系统将自动跳转至登陆页面。

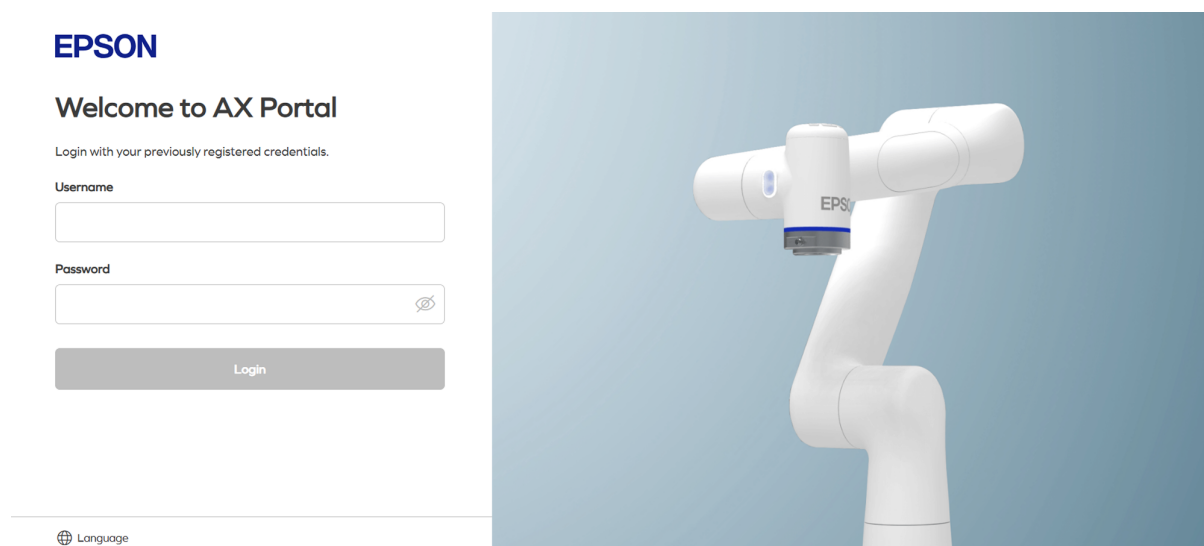


图 2.2.2: 登陆页面

2.2.3 账户管理

在“账户管理”菜单中，您可以修改个人资料，具体包括您的姓名、密码、头像以及初始界面：

- “全名”是界面和消息中用于称呼您的姓名。这是一个必填字段，用于填写您的姓名和姓氏，但您也可以定义自己的命名规则。
- “用户名”字段在用户创建后无法更改，该用户名由管理员在用户创建时设定。
- “初始界面”字段用于指定登录后希望跳转到的界面。不同用户等级预设了不同的默认初始界面，您可以在此处进行更改。
- “更改密码”按钮允许您修改账户密码。系统会要求您重新输入当前密码，并两次输入新密码，以最大限度地减少设置密码时的意外错误。如果您忘记密码，任何管理员也可以为您更改。密码绝不会以明文形式保存，因此您或其他任何用户都无法提取您的密码。
- 更换“个人头像”为可选操作。若未定义个人头像，系统将显示默认用户图标。

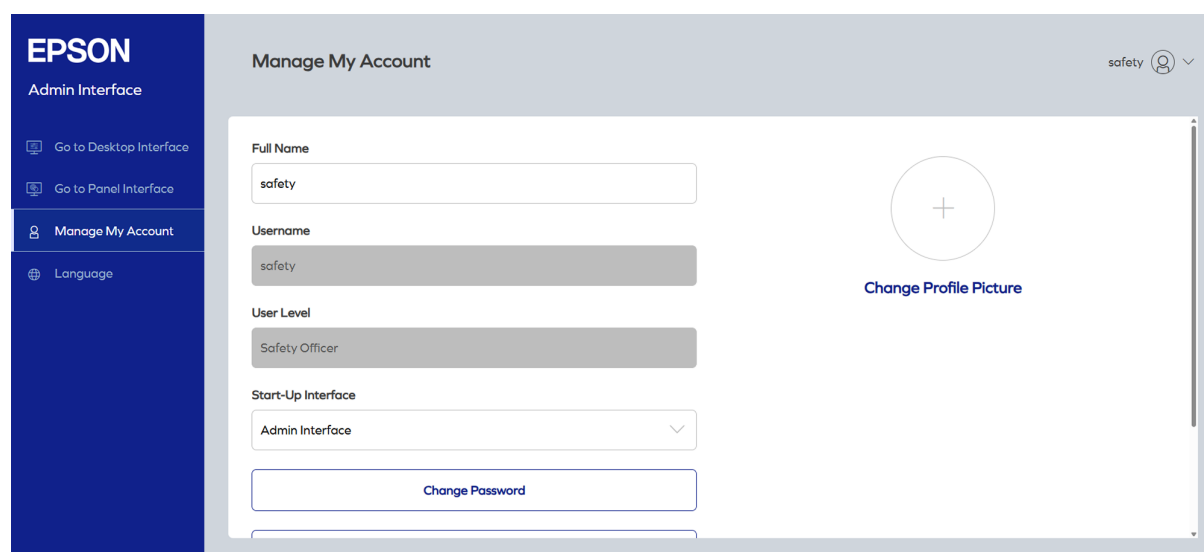


图 2.2.3: 账户管理页面

2.2.4 管理其他用户

在“管理其他用户”菜单中，您将看到当前所有可用用户的概览。此处提供以下选项：

- 添加新用户可创建具有预设用户等级和预设初始密码的用户。任何用户首次登陆后都应立即更改初始密码。
- 修改用户可更改该用户的多数属性，但用户名除外，该字段完全不可更改。
- 删除用户将永久从数据库中移除该用户。此用户将无法再触发机器人控制功能。

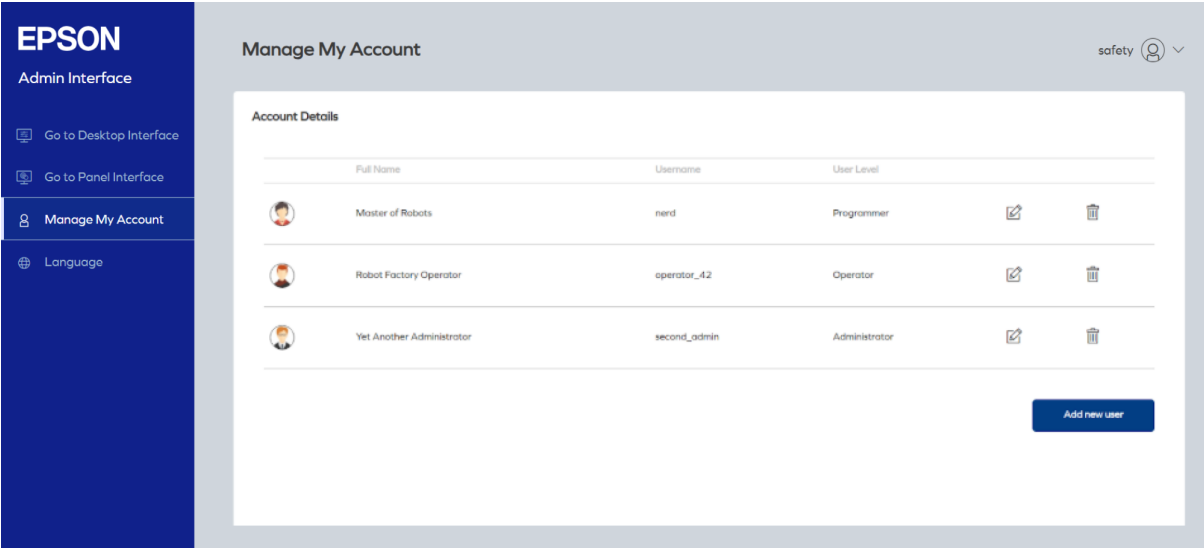


图 2.2.4: 管理其他用户页面

2.3 系统设置

2.3.1 网络设置

“网络设置”菜单包含定义机器人以太网连接的所有相关设置。请注意，一旦您更改了当前活动连接的设置(例如，您通过以太网连接，并更改了以太网 IP地址)，您将失去连接。如果您不小心输入了无效设置，之后可能无法再连接到机器人。 因此强烈建议您在更改唯一可用连接的设置前，先确保有两个不同的连接均可正常工作。

“以太网设置”类别可让您查看并决定机器人使用动态IP地址还是手动IP地址。如果机器人配置为动态获取IP地址，您将在此处看到已选择的IP地址。如果机器人配置为手动操作模式，您需要输入 IP地址、子网掩码(必填)，此外还可以定义网关和最多两个 DNS 服务器(可选)。

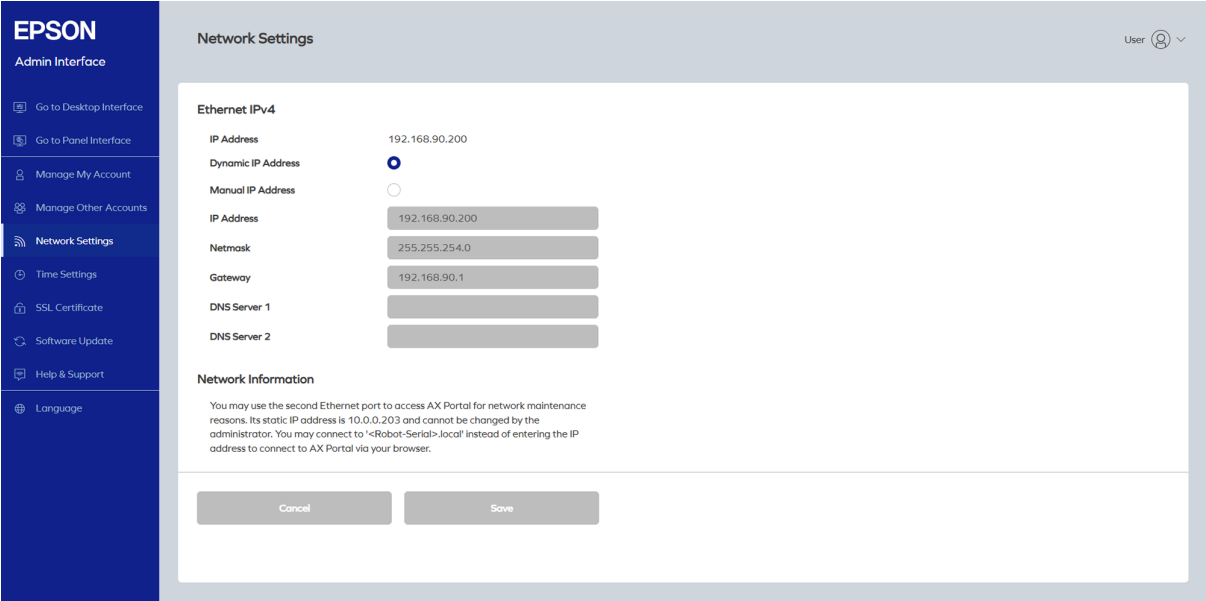


图 2.3.1:网络设置页面

2.3.2 时间设置

“时间设置”菜单显示机器人操作系统的当前日期、时间和时区。请确保输入正确的时间和日期。设置正确的时间后，系统可用于日志分析和故障排除，且基于时间的功能也能正常运行。若手动更改时间，点击保存按钮后，新时间将立即生效。若需更改时区，可通过大洲（例如“Europe”）、国家代码（例如“GB”）、主要城市（例如“London”）或这些要素的组合（例如“eur lon”→ Europe/London）对系统提供的可用时区列表进行筛选。

注意：如果机器人具有网络访问权限，您可以启用 NTP（网络时间协议）同步。

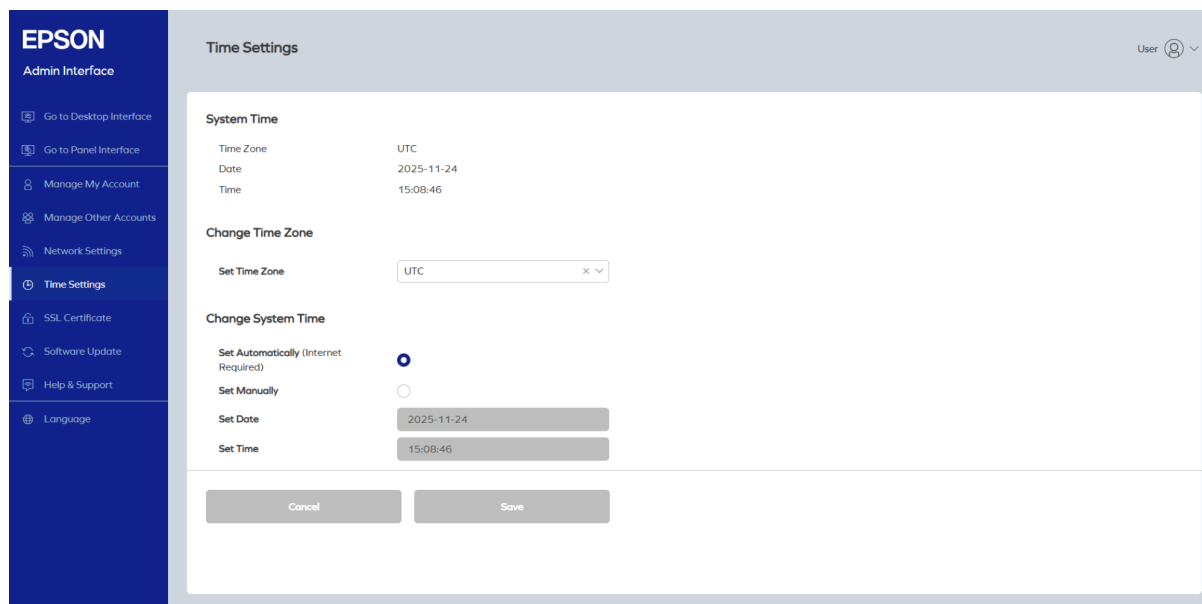


图 2.3.2: 时间设置页面

2.3.3 SSL证书

“SSL证书”菜单向您显示有关当前加载的SSL证书的一些详细信息，并允许您将此证书替换为您自己的私有或企业证书。SSL（安全套接层）证书是一种数字证书，用于验证网站的身份并建立加密连接。该证书可确保您在 Web 浏览器中的在线交易和客户信息保持私密和安全。默认情况下，系统安装的是自签名证书，建议将其替换为适合用户环境且由受信任的权威机构签名的证书。

如果出现以下情况，证书将被浏览器标记为不安全：

- 证书中注册的域名/IP 与访问机器人所用的域名/IP 不匹配，或
- 证书已过期，或
- 证书为自签名证书。

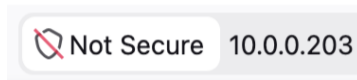


图 2.3.3.a: 浏览器提示 SSL 证书不安全或缺失的示例

如需更换 SSL 证书，您需要上传私钥和公钥这对文件。

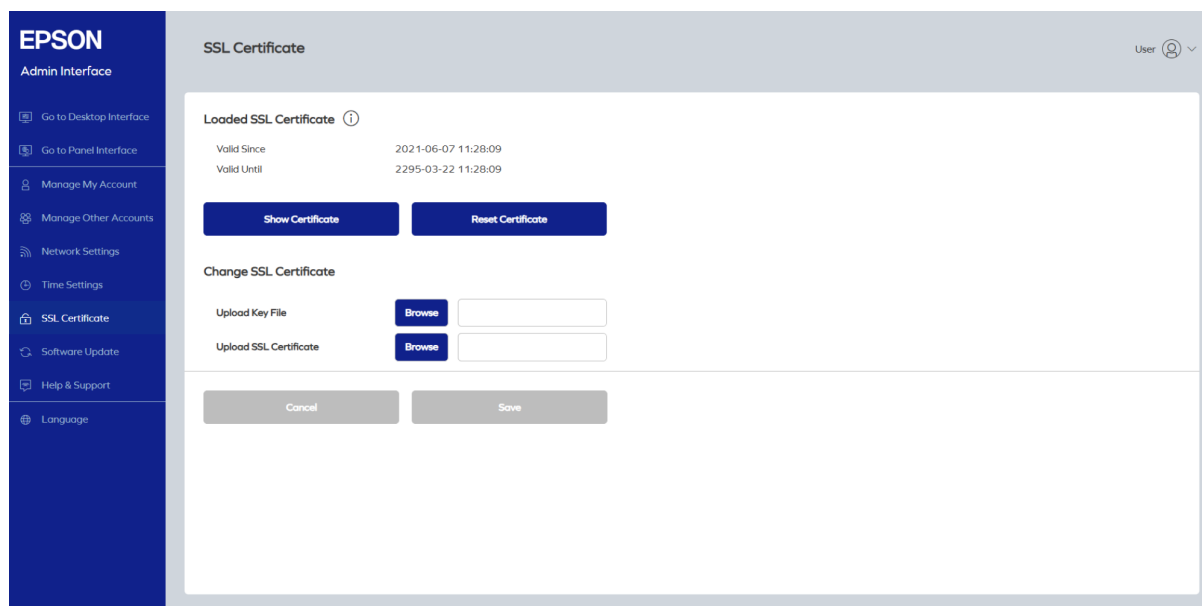


图 2.3.3.b: SSL 证书设置页面

2.4 软件更新

在“软件更新”菜单中，您可以查看当前运行中的 **AX Portal** 版本。您可以在这里上传及安装新软件版本。软件更新可能是为了修复特定问题，或是获取最新功能和错误修复。

无论何种情况，在更新软件版本前，请务必先仔细阅读发布说明，其中包含关于更新至特定版本可能带来的风险的重要信息。改进有时会以弃用旧功能或特性为代价。我们建议您在触发更新前，通过桌面备份数据库（其中包含所有机器人应用程序和设置）。

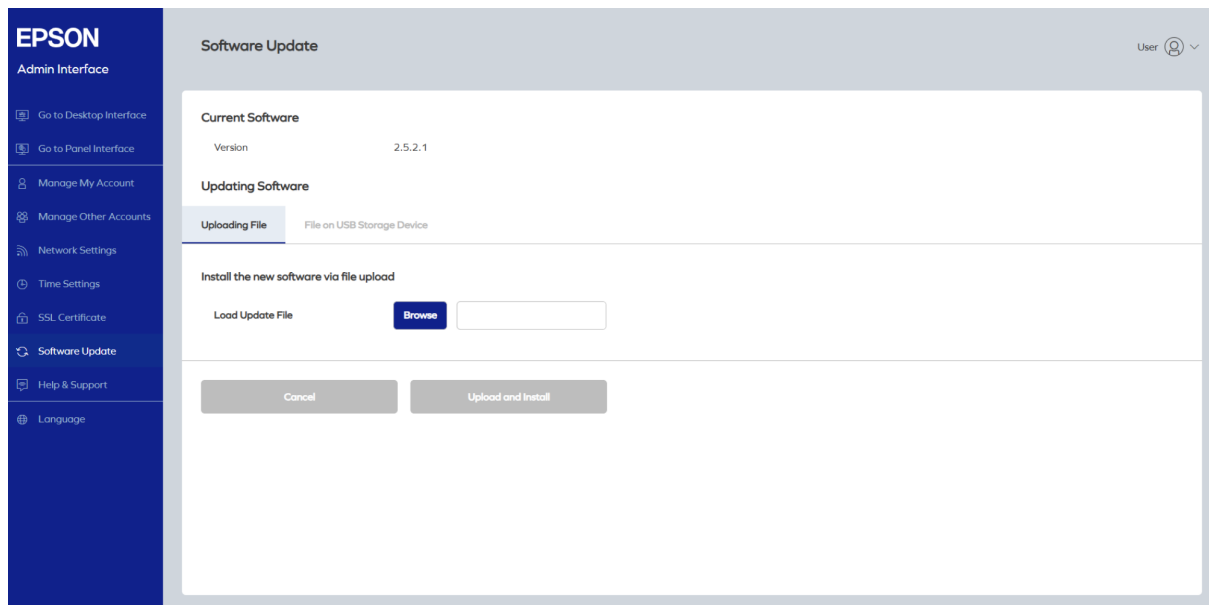


图 2.4.a: 软件更新页面

要安装软件更新，您首先需要获取包含所需 **AX Portal** 版本更新的 **RBIN** 文件。安装此更新有三种方法：

- 1) 选择第一个选项卡“上传文件”。在此您可以从设备上传更新文件。浏览文件系统并选择更新用的 **RBIN** 文件，然后继续安装更新。
- 2) 选择第二个选项卡“文件在 USB 存储设备上”。在此，您可以将 **RBIN** 文件放入 U 盘并将其挂载到机器人的控制器上。从 U 盘目录中选择更新 **RBIN** 文件，然后继续安装更新。
注意：目前仅支持 **exFAT** 和 **FAT32** 文件系统的 USB 闪存盘。
- 3) 如果选项 1) 和 2) 不适用（例如管理员面板在您的机器人上无法正常运行），您始终可以选择运行恢复更新。

在更新过程中，您可以查看当前更新的进行状态、已成功完成的步骤，以及发生错误时具体在哪个步骤出错，并会显示相应的错误信息。根据正在更新的软件包不同，整个更新过程最多可能需要 **15** 分钟。因此请耐心等待，切勿通过切断机器人电源等方式中断更新。最坏的情况下，这可能会导致您后续无法再访问该机器人。

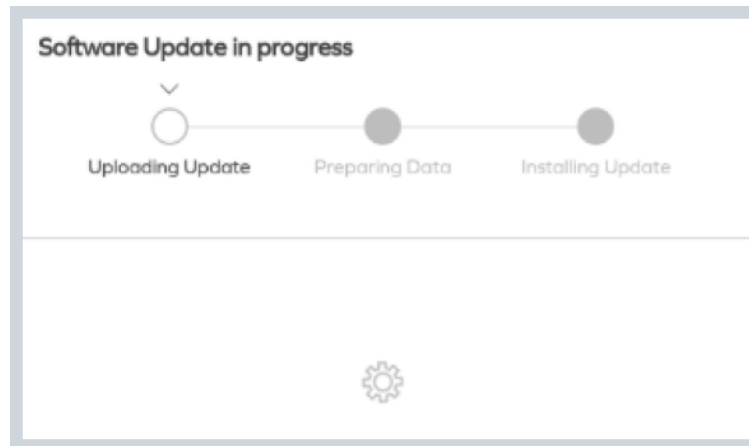


图 2.4.b: 更新进行中窗口

更新成功完成后，必须确认更新。在确认更新之前，红色消息栏将持续显示提示。若未确认更新，重启控制器将重置更新。一旦确认，将无法回滚至较早版本。

2.5 帮助 & 支持

“帮助 & 支持”菜单提供了您在需要帮助时联系爱普生的联系方式。请务必先仔细阅读文档，再通过电子邮件或电话寻求支持。

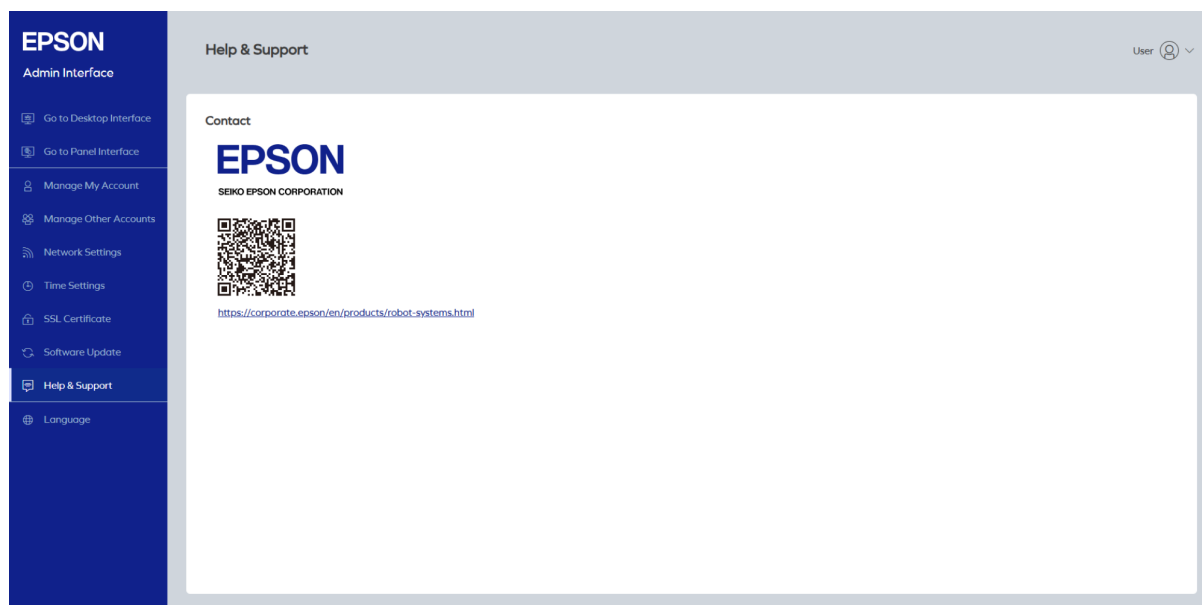


图 2.5:帮助 & 支持页面

2.6 安全认证

如果您使用连接模块访问 AX Portal 来控制机器人(不使用 GUI), 在实际控制机器人之前, 您需要通过安全的分两步验证流程进行身份验证。

2.6.1 出口 SSL证书

第一步, 您需要向内置身份验证服务器发送登陆请求。若需下载当前加载的 SSL证书, 可通过浏览器导出证书。下图展示了如何使用 Google Chrome 完成此操作。其他浏览器也支持类似功能。

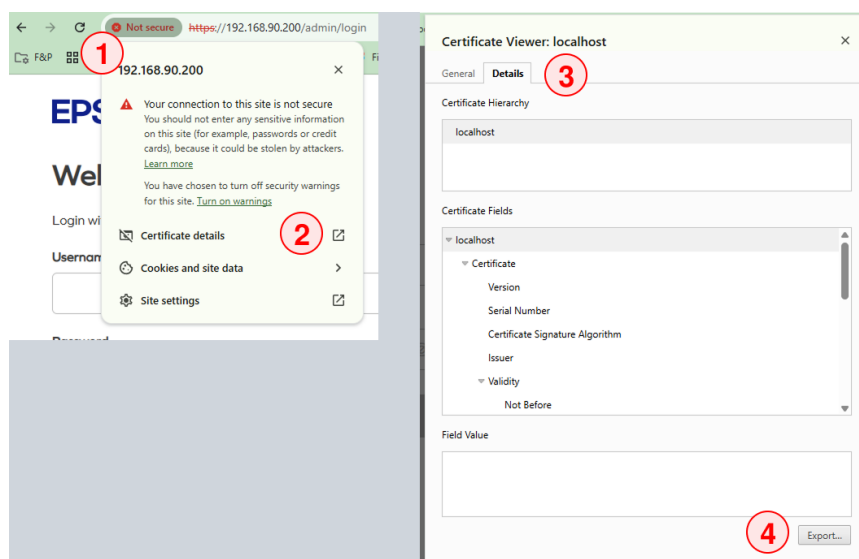


图 2.6.1: 出口 SSL证书

1. 查看网站信息, 2. 转至证书详细信息, 3. 在证书查看器中查看详细信息, 4. 出口证书

2.6.2 用户登陆

第一步, 您需要向内置身份验证服务器发送登陆请求, 该服务器可通过 **TCP 5001** 端口访问。为此, 您需要知道机器人的 **IP地址**、您的登陆凭据(用户名和密码), 并且必须使用与机器人相同的 **SSL证书**。以下是一个通过 TCP 套接字发送登陆请求并返回访问令牌的 Python 示例。您也可以使用其他编程语言实现此功能。

注意: 目前仅支持通过TCP登陆。若您希望通过TCP以外的连接模块(如ROS/REST/Modbus)控制机器人, 仍需使用 TCP 获取访问令牌。或者, 您也可以在 AX Portal 的设置中关闭这些连接模块的认证。 注意: 若禁用身份验证, 任何客户端均可通过 ROS/REST 控制机器人, 无需事先验证!

```
import json, socket, ssl

# define the robot ip address, login credentials, and path to SSL certificate
ip_address = "10.0.0.203"
username = "user_xyz"
password = "my_fancy_password"
session_id = "my_session_id"
ssl_cert = "/home/user/robot-ssl/cert.pem"

# create request to send
request = {"request": "authentication", "action": "login", "data":
           {"username": username, "password": password, "session_id": session_id}}

# prepare SSL context to use with the connection
ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
ssl_context.load_verify_locations(ssl_cert)
ssl_context.check_hostname = False
ssl_context.verify_mode = ssl.CERT_NONE

with socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0) as sock:
    with ssl_context.wrap_socket(sock, server_hostname=ip_address) as ssock:
        # connect to the authentication socket
        ssock.connect((ip_address, 5001))
        # send request (and end character)
        message = json.dumps(request) + "\x03"
        ssock.send(message.encode("UTF-8"))
        # receive answer (before end character)
        message = ssock.recv(1024).decode("UTF-8")
        answer = json.loads(message.split("\x03")[0])

token = answer["response"]["data"]["token"]
session_id = answer["response"]["data"]["session_id"]
```

2.6.3 连接注册

一旦获得有效的访问令牌，您就可以在“token_login”函数中使用它，该函数的使用方法在连接模块的专门章节中有详细描述。

3. USB 出口和恢复更新

3.1 概述

如果控制器正在运行中但无法访问管理员面板，您可以尝试使用恢复更新方法来恢复控制器。

3.2 USB 出口

无需访问网站界面，即可使用 USB 闪存盘导出系统日志和 AXPortal 数据库：

1. 将 USB 闪存盘格式化为 exFAT、FAT32 或 NTFS，并确保其文件系统标签中不包含日志或 RECOVERY
2. 将其插入控制器并等待几分钟。
3. 拔出 USB 闪存盘。您应在闪存盘的根目录中找到一个 zip 文件，其中包含数据库和相关的系统日志。

3.3 运行中恢复更新

要运行恢复软件更新，请按照以下步骤操作：

1. 获取更新文件，该文件通常为RBIN 格式。若您没有更新文件，请联系供应商的联系方式。
2. 请确保手边有一支可用于更新的U盘。该U盘应格式化为exFAT、FAT32或NTFS格式，且第一个分区的文件系统必须命名为“RECOVERY”。
3. 加载更新文件，将其复制到U盘的根目录下。
4. 关闭机器人电源并重新启动。请等待至少 1 分钟，确保机器人正常启动。
5. 将准备好的U盘插入控制器任意USB接口。
6. 等待更新完成。请注意，此过程可能需要长达 15 分钟。尝试访问管理员面板，检查新版本是否已安装。
7. 重要提示：请从控制盒中安全移除 USB 闪存盘。若遗漏此步骤，重启后更新程序将再次运行，这并非您所期望的。

4. 桌面界面 (GUI)

4.1 简介

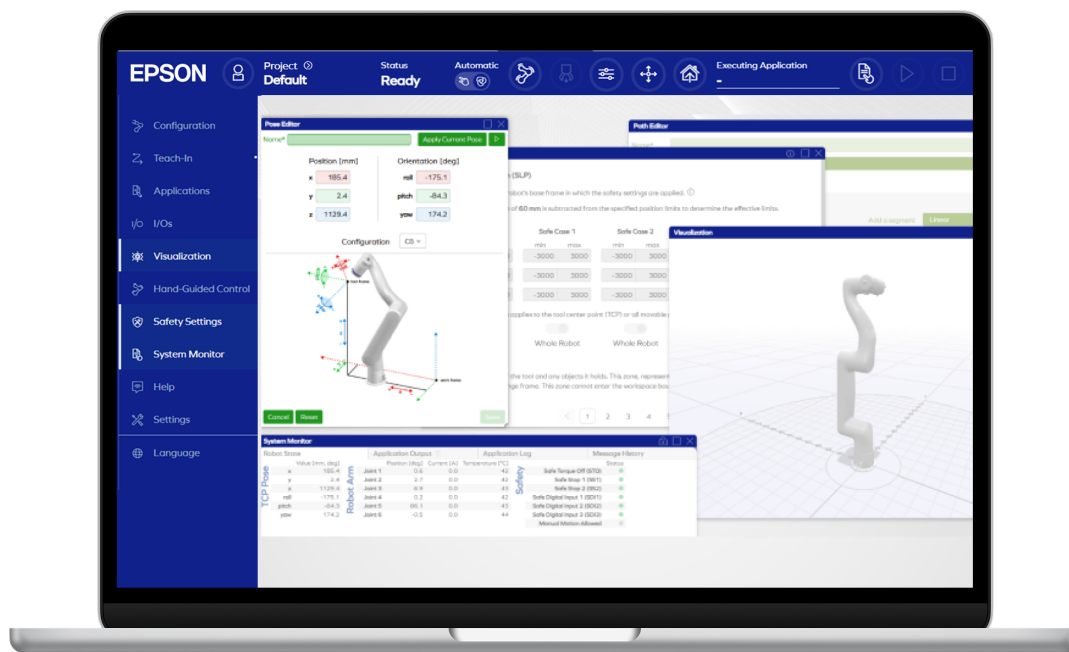


图 4.1.a: 桌面界面的示例截图

AX Portal的桌面界面是主要的图形用户界面(GUI)，允许用户控制机器人、编程应用程序、操作数据库元素以及可视化当前机器人状态。该GUI设计直观且基本一目了然，即使对机器人控制经验不足的用户也是如此。您可以通过管理员面板登录后，点击“导航到桌面”按钮访问此界面。

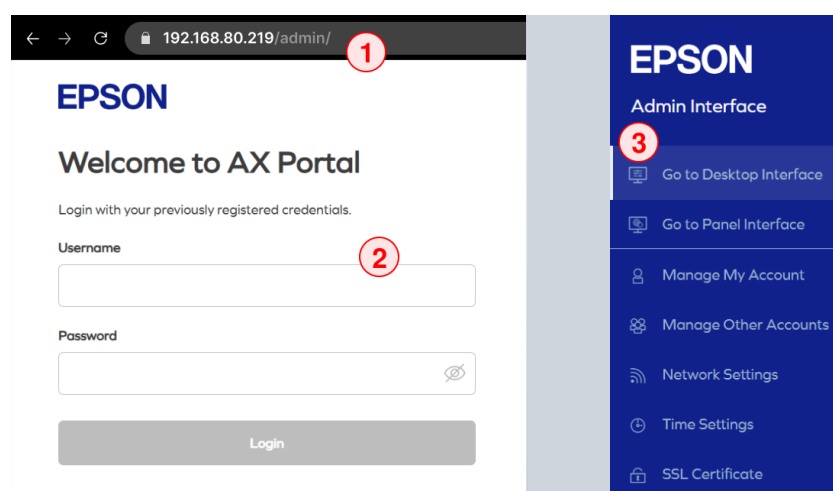


图 4.1.b: 访问桌面

1. 在浏览器地址栏中输入 IP 地址，2. 输入凭据登录，3. 导航至桌面界面

桌面界面包含机器人所有可用的控制功能，以及用于观察和修改其状态的附加功能。若仅需执行预编程的程序，且无需修改任何程序或设置，建议改用面板界面。本章将详细说明桌面界面的所有窗口、菜单及元素。

在某些浏览器中（特别是在 Windows 系统上），显示缩放比例默认设置较高（例如 150%），这可能会导致界面部分内容（如驾驶舱或徽标）显示不完整。为了获得最佳体验，建议将显示缩放比例设置为 100%（甚至更低）。

4.2 主窗口

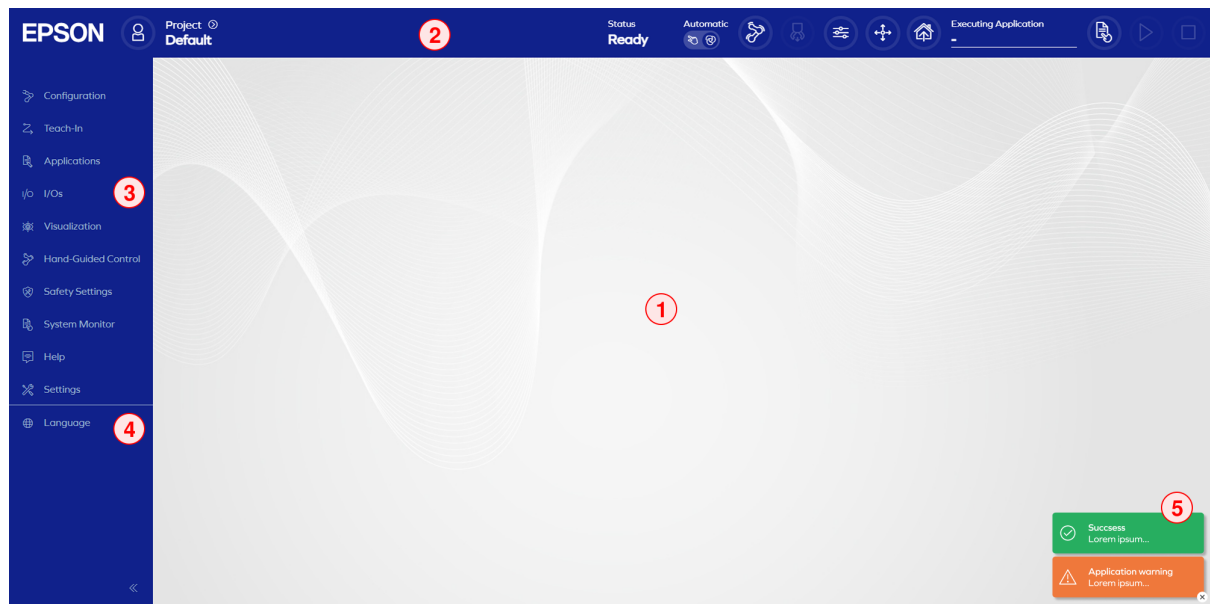


图 4.2: 桌面界面的元素

1. 工作区, 2. 控制台, 3. 菜单, 4. 切换语言, 5. 弹出消息

当您成功将计算机或设备连接到桌面后，主窗口将显示在您的网页浏览器中。

该主窗口包含以下内容：

- 窗口最上方的控制台，用于显示当前程序状态、消息以及快速访问机器人控制功能；
- 窗口左侧的菜单按钮，点击可导航到可用菜单（本章后文将详细说明），
- 屏幕右侧用于显示消息弹窗的区域，以及
- 主工作区，用户可以在其中以类似资源管理器的样式重新排列菜单。

4.2.1 菜单概述

在主窗口的左侧，您会看到进入 AX Portal 不同菜单的按钮。每个可用菜单的具体用途和详细信息将在后续章节中描述。以下是对可用菜单的简要概述：

	“配置”菜单允许用户对机器人进行配置和上电。
	“示教”菜单允许用户向机器人示教位姿、路径和网格。
	“应用程序”菜单允许用户编写、修改和测试自定义应用程序。
	“I/O”菜单允许用户监视I/O状态并更改输出信号。
	“可视化模型”菜单用于显示机器人的当前运动状态。
	“使能控制”菜单允许用户手动移动机器人的关节。
	“安全设置”菜单允许安全主管修改安全设置。
	“系统监视器”菜单允许用户查看系统输出。
	“帮助”菜单包含有关程序和系统的有用信息。
	“设置”菜单允许用户选择一些基础设置。
	“语言”按钮允许用户切换界面的显示语言。

4.2.2 控制台

AX Portal 的控制面板始终显示在 GUI 顶部，其中包含最重要的信息和控件。让我们从左到右详细说明控制面板中的所有元素：

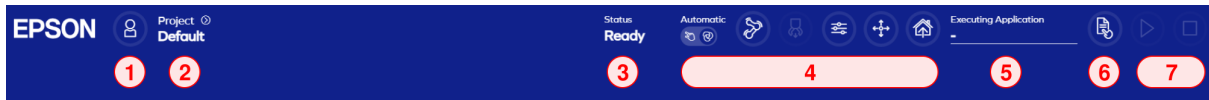


图 4.2.2: 控制台

1. 用户账户选项, 2. 项目选择, 3. 机器人状态, 4. 机器人控制, 5. 运行中或选定的应用, 6. 应用程序选择, 7. 主控件

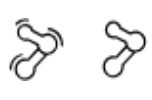
 点击“用户”按钮可更改账户设置或登出。这些选项将引导您进入管理员面板。

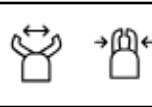
按下“项目”按钮可进入项目菜单，您可在其中切换项目并查看项目概览。该按钮下方显示当前在活动中使用的项目名称。

“请求控制”按钮仅在当前界面并非机器人的活动控制站时显示。有关此安全功能的更多信息，请参阅“活动控制”部分。

状态消息栏显示当前的程序状态。程序状态基本决定了命令或操作（如按键）是否可执行。

“操作模式”滑块用于在手动操作模式和自动操作模式之间切换机器人。有关此安全功能的更多信息，请参阅“操作模式”部分。

 “启用/禁用 HGC”按钮允许用户切换至手动引导控制模式。启用后，状态将切换为“HGC”，用户可手动引导机器人。只有当程序状态为“就绪”时，机器人才能切换至 HGC 模式。禁用 HGC 将使状态恢复为“就绪”。

 “工具”按钮可在已安装工具的“拾取”和“放置”功能之间切换。在工具编辑器中，用户可将这些功能与数字输入关联。只有在行为已定义且程序状态为“就绪”或“HGC”时，才能触发该工具。

 “精细控制”按钮将打开精细控制界面。在此，用户可调整机器人的单个坐标或关节角度。

 “运动”按钮打开运动控制面板。在此，用户可调节机器人速度、修改全局速度和加速度限制，以及切换坐标模式。

 “归位”按钮允许用户将机器人移动回其回零位置。此功能旨在作为快捷方式，例如在运行程序前移动到初始位置，或在断电中移动到休眠位置。回零位置可在“设置”菜单中更改。

“当前选定”区域显示当前与控制按钮（播放、暂停、继续、停止）关联的元素（应用程序、路径、位姿等）。每次点击播放按钮或选择应用程序时，此信息都会更新。

 “应用程序选择”菜单允许用户选择之前使用程序编辑器创建的应用程序。

 “播放”按钮用于执行所选应用程序。程序状态必须为“就绪”才能播放应用程序。“暂停”按钮用于暂停正在运行的应用程序。若在机器人移动过程中暂停应用程序，机器人将平稳地停止。“恢复”按钮用于恢复已暂停的应用程序。若应用程序是在移动过程中被暂停的，该移动将平稳地继续。“停止/中止”按钮将停止运行中或暂停的应用程序。



若应用程序在移动过程中被停止，机器人将立即停止并保持当前位置。此按钮将触发自动恢复程序，将状态切换回“就绪”。



4.2.3 活动控制 (单点控制器)

为确保机器人任何时候都只能由单一控制站进行控制，接口通过“活动控制”功能受到保护。如果您是首位登录机器人的用户，您将自动获得控制权。我们将此称为“活动控制站”。如果其他人(或您自己)随后尝试使用不同的连接通道(也可能是另一个浏览器窗口)，该连接将无法获得控制权，仅能访问机器人监控和观察功能(只读访问)。我们将此称为“被动观察者”。

4.2.3.1 活动控制站

主动控制站可以释放其控制权限。在界面中，通过点击“用户”按钮并选择“释放控制权限”即可实现。如果您通常在有多个用户或界面的机器人上工作，建议始终释放控制权限和/或关闭连接通道。

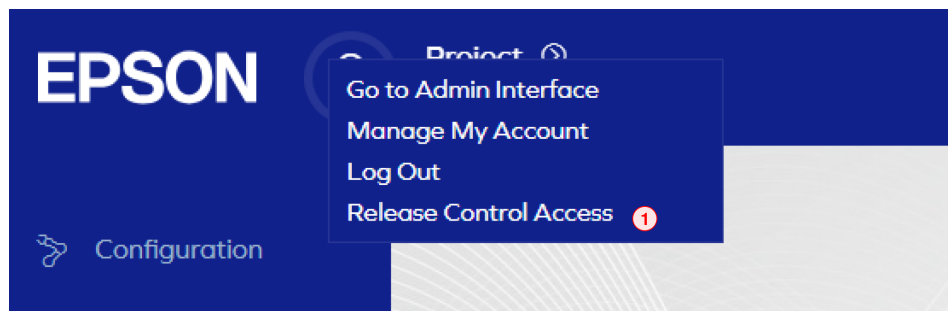


图 4.2.3.1 释放控制权限

1. 释放控制权限

4.2.3.2 被动观察者

被动观察者可以请求控制权。在界面中，通过点击“用户”按钮并选择“申请控制权限”，或直接点击驾驶舱中出现的大号“请求控制”按钮即可实现。申请控制后，主动控制站和被动观察者双方都会弹出对话框，询问是否允许访问。被动观察者只能等待控制权限被授予，而主动控制站可以接受或拒绝该请求。



图 4.2.3.2 申请控制权限
1. 申请控制权限, 2. 请求控制对话框

4.2.4 程序状态

当前程序和/或机器人的状态可在驾驶舱中查看。该状态反馈了AX Portal和/或机器人当前的运行情况，并基本决定了当前可执行的操作或命令。每种状态都有其独特的属性，某些功能仅在特定的程序状态下可用。以下是各种状态及其说明的列表：

未上电	电机未通电，设备未连接。此状态用于上电前配置机器人。使用上电滑块会将状态切换为“上电中”。其他操作可能会将状态切换为“加载中”，特别是当操作不应被中断且需要一定时间完成时。
加载中	机器人当前正在处理某些数据，例如加载数据库或某些设置。数据加载完成后，状态将恢复为“未上电”。
上电中	机器人当前正在上电并连接设备。连接成功后，机器人将开始主动控制电机，状态将变为“就绪”。
就绪	程序已就绪，准备好接收指令和/或执行操作，例如执行程序、拾取和放置物体等。这是机器人的空闲状态。
运行中	当前正在运行应用程序/运动。可通过相应的“暂停”或“停止”按钮分别暂停或停止该应用程序/运动。当应用程序运行至程序脚本末尾或当前运动结束时，状态将恢复为“就绪”。
暂停	当前应用程序/运动已暂停。按下“继续运行”按钮可恢复运行，状态将恢复为“运行中”；按下“停止”按钮可中止运行，状态将恢复为“就绪”。
使能控制	手动引导控制(HGC)已启用，适用于机器人的一处或多处关节。在此模式下，您可通过施加力量于关节，手动引导机器人达到任意姿态。同时，机器人会通过主动补偿重力来保持自身平衡。请注意，您需要按下 HGC 按钮或激活使能按钮才能移动机器人。再次禁用手动引导控制，状态将恢复为“就绪”。
处理中	当前正在进行耗时的操作，例如数据保存/加载或轨迹计算。该操作处理成功后，状态将恢复为之前的数值。
停止中	通常情况下，“停止中”状态会立即自动恢复为“就绪”。如果未恢复，则说明 AX Portal 的某些部分未能正确完成。遗憾的是，此时只能通过重启机器人来恢复。

急停 正常停止 保护停止 碰撞 安全防护 位置限制	这些状态对应不同的中断。如果触发中断，将显示相应的状态，并打开“中断窗口”以处理该中断。一旦中断得到解决，状态通常会变为“继续”。
继续	中断解决后，状态将变为“继续”。这要求用户手动确认可以安全地继续执行机器人应用程序或行为。
错误	发生了无法自动解决或需要用户特别关注的错误。请查看错误信息，并在可能的情况下尝试解决该错误。
断电中	机器人当前正在断电中，关闭电机电源、断开设备连接，并结束所有运行中的模块。此后状态将变为“未上电”。

4.2.5 操作模式

AX系列机器人提供两种操作模式:手动操作模式与自动操作模式。可通过驾驶舱内的专用滑块切换操作模式。

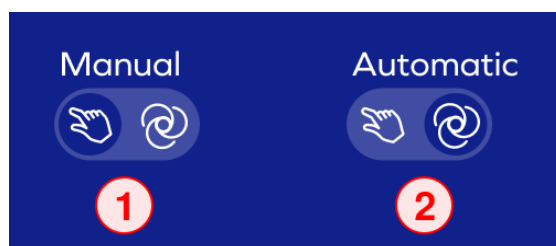


图 4.2.5 操作模式切换按钮

 警告	仅当系统配备必要的安全功能、已通过测试且其安全性经风险评估确认后，方可使用自动操作模式。在其他情况下使用自动操作模式非常危险。更多信息请参阅《AX6 & RC-A101 安全手册》。
---	---

4.2.5.1 自动操作模式

自动操作模式通常用于正常操作，即在程序已验证可正常运行时。此模式允许机器人使用全速范围。

4.2.5.2 手动操作模式

手动操作模式是一项安全功能，通常用于修改机器人程序时，特别是在教授新姿势或测试程序变更时。在手动模式下，机器人以减速状态移动（最大 250 mm/s），且仅在使能按钮被触发时才会移动。使能按钮是一个三位置安全开关。如果使能按钮未被触发，系统将通过警告提示用户，表明不允许移动。



图 4.2.5.2 使能按钮示例

4.2.6 精细控制

按下驾驶舱内的精细控制按钮可打开“精细控制”窗口。通过精细控制操作，可将机器人调整至0.1 [mm]/[deg]的精度。该窗口包含两个选项卡。第一个选项卡允许用户修改工具中心点(TCP)的当前欧几里得坐标，该坐标即以工件坐标系表示的当前工具坐标系。第二个选项卡允许用户修改机器人当前的关节角度。

这些选项卡具有以下功能：

输入字段	在输入字段中输入数值可更改单个坐标或关节值。
滑块	滑块既定义了“增量步长”(即短按控制按钮时机器人移动的距离)，也定义了“最大运动速度”(即使用这些控制时机器人的移动速度)。
控制按钮	短按控制按钮将沿按钮所示方向应用选择的增量。而长按控制按钮则将沿按钮所示方向进行连续运动。
配置	一个特定的目标坐标点(TCP)可以通过多达八组关节角度来达到。这些关节角度组被称为该位姿的配置。通过在下拉菜单中更改“配置”值，用户可以为当前位姿选择特定的配置。通过点击“下一个配置”按钮，用户可以遍历当前位姿的可用配置。请注意，更改配置并非微调步骤，这可能会移动机器人的所有关节。
保存为姿势	点击“保存为姿势”按钮，该姿势将以一个新的唯一名称保存。所有可用姿势均列于“示教”菜单中。

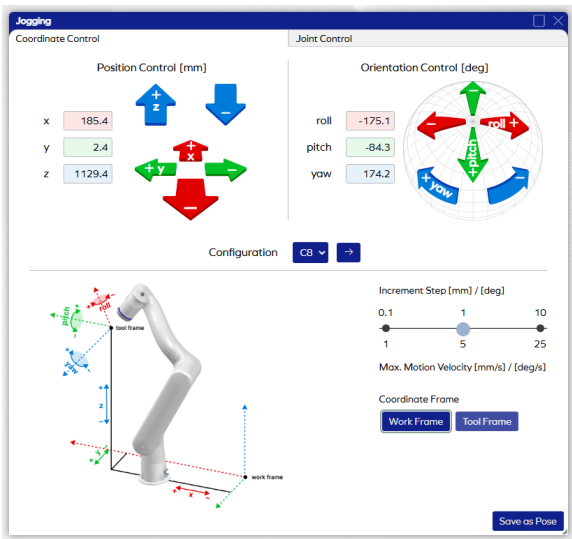


图 4.2.6.a:精细控制——坐标控制

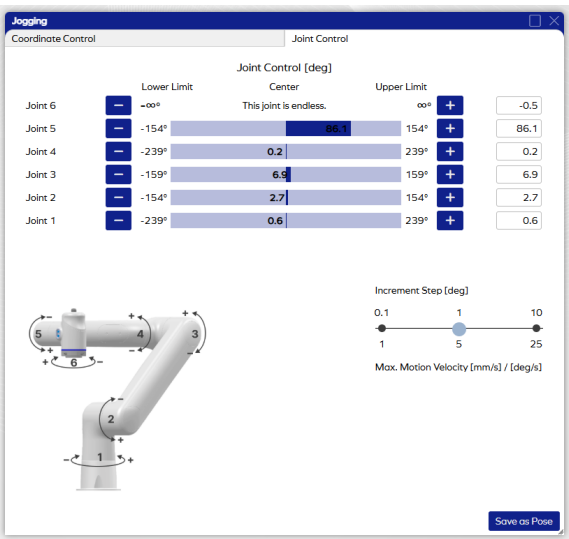


图 4.2.6.b:精细控制 – 关节控制

4.2.7 运动控制

按下驾驶舱中的“运动控制”按钮，即可打开“运动控制”窗口。该窗口包含以下三个选项卡：

4.2.7.1 运动限制

“运动限制”选项卡包含用于限制机器人最大速度的设置，以及通过缩放因子降低所有机器人运动速度的设置。这些控制功能在设置机器人应用程序时非常有用，例如在验证新应用程序或更改现有应用程序的运动时限制速度。

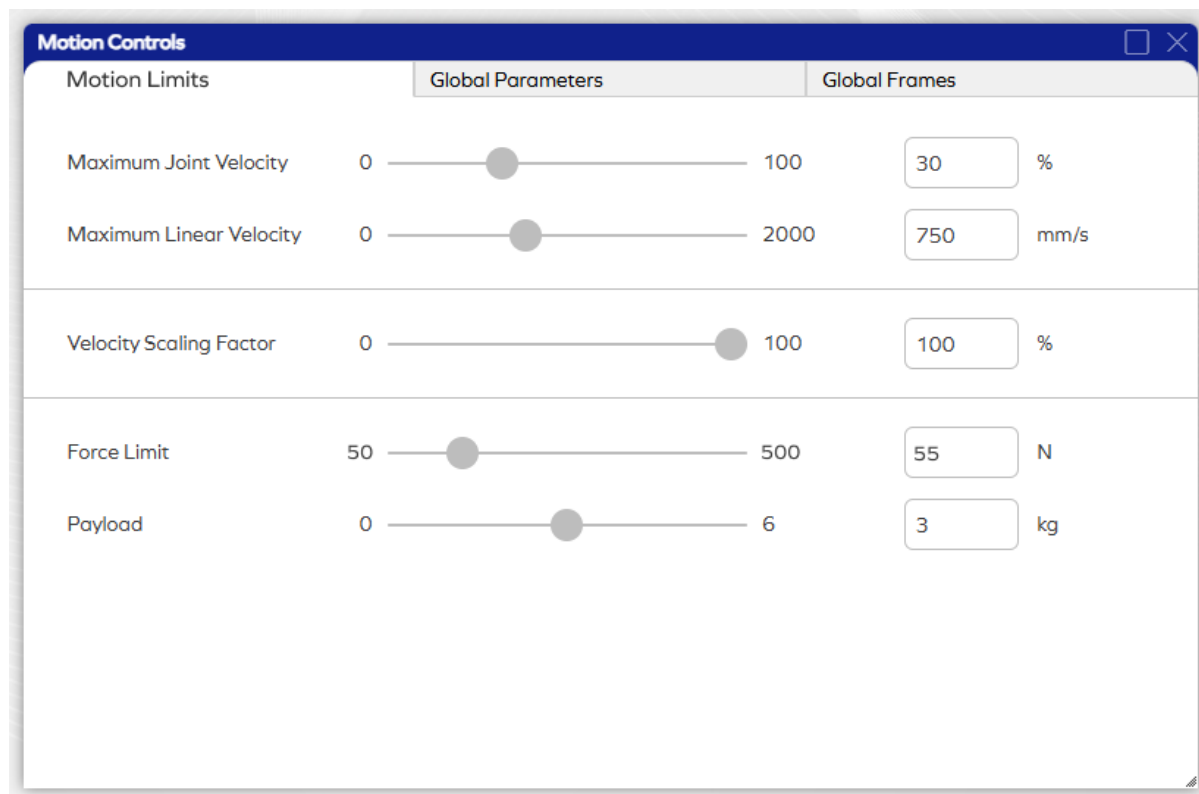


图 4.2.7.1: 运动控制 – 运动限制选项卡

另一方面，力限制和负载用于调整机器人的碰撞灵敏度。力限制定义了作用在工具上的允许外部力的上限，而负载则定义了机器人所承载的总重量(包括其工具)。

4.2.7.2 全局参数

全局参数选项卡包含用于定义各种运动命令默认速度和加速度的设置。运动命令分为以下几类：

- 用于以时间最优方式移动机器人的命令使用“joint_velocity”和“joint_acceleration”参数，其默认值在此处设置为全局关节速度和全局关节加速度。
- 用于使机器人沿特定方向或轨迹（直线、圆弧运动）移动的命令使用“linear_velocity”和“linear_acceleration”参数，其默认值在此处被设置为全局线速度和全局线加速度。
- 用于移动机器人，使工具中心点（TCP）保持在同一位置，仅改变其方向的命令，使用“angular_velocity”和“angular_acceleration”参数，其默认值在此处设置为全局角速度和全局角加速度。
- 用于沿包含多个运动段的路径移动的命令使用“blend_radius”参数，其默认值在此处设置为全局混合半径。该参数用于平滑路径段之间的插值。

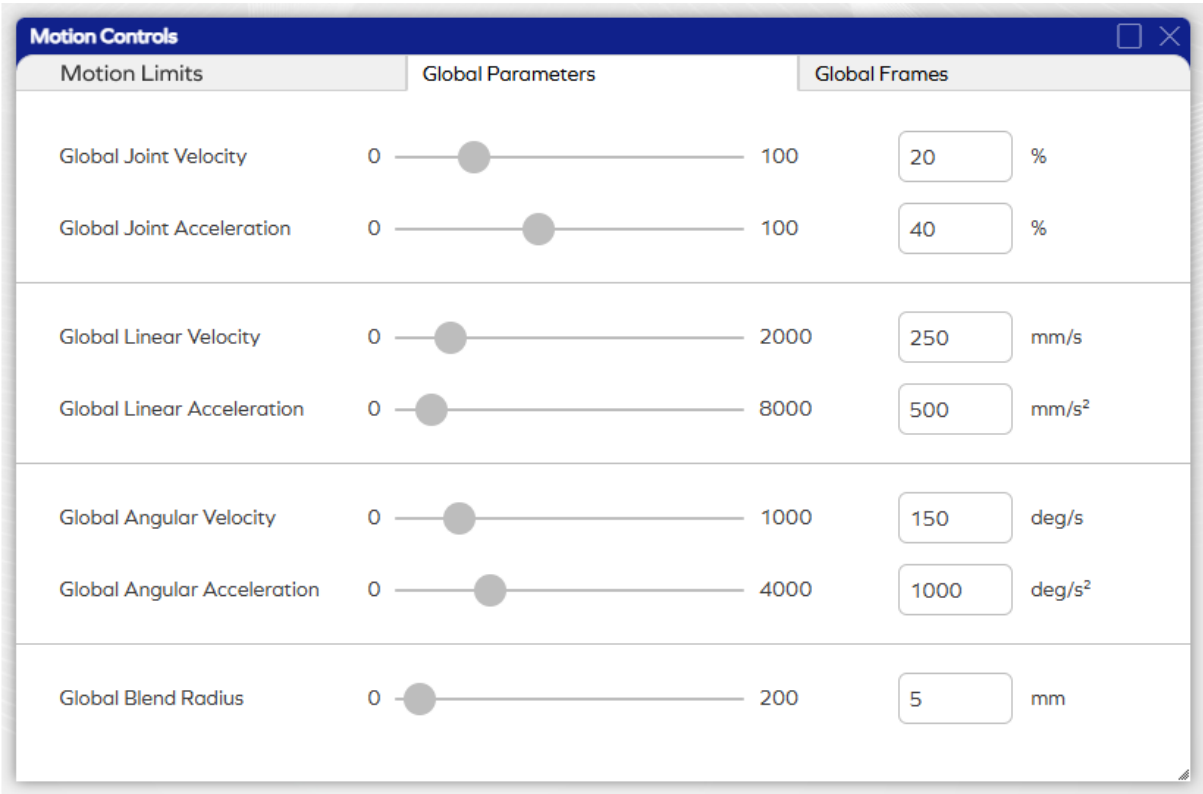


图 4.2.7.2: 运动控制 – “全局参数”选项卡

4.2.7.3 全局坐标系

全局坐标系选项卡包含用于定义机器人默认坐标系的设置。虽然基坐标系和法兰坐标系是固定连接在机器人上的，但工件坐标系和工具坐标系可以相对于这些固定坐标系进行定义：

- 基坐标系固定在机器人底部安装面的中心，此时 z 轴指向机械臂内部， x 轴指向远离已连接电缆的方向。
- 法兰坐标系固定在机器人顶部安装板的中心。当机器人所有关节角度均为零度时，其方向与基坐标系对齐。
- 工作坐标系是用户在程序中表示工具中心点 (TCP) 坐标的参考坐标系。默认情况下，工作坐标系与机器人的基坐标系一致。许多移动命令使用“`work_frame`”参数，其默认值在此处设置为全局工件坐标系。
- 工具坐标系定义了法兰坐标系中表达的工具中心点 (TCP) 的位置和方向。默认情况下，工具坐标系与机器人的法兰坐标系一致。许多运动命令使用“`tool_frame`”参数，其默认值在此处设置为全局工具坐标系。

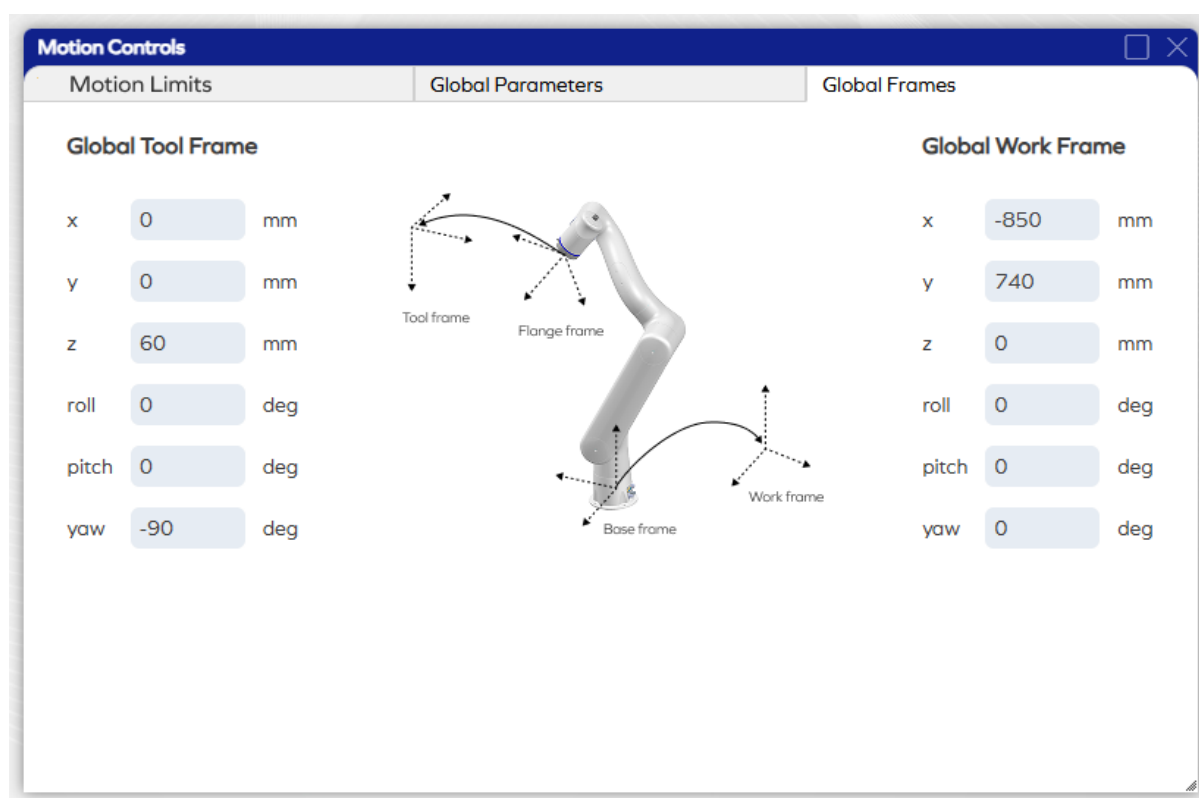


图 4.2.7.3: 运动控制 – 全局坐标系选项卡

4.3 构建模块与数据库

为了实现简单直观的应用程序编程, AX Portal 提供了一个包含不同构建块的面向对象结构。这些构建块(以下简称“元素”)可以在不同的编辑器中单独创建, 这些编辑器可通过桌面界面访问。AX Portal 允许用户智能地将这些元素相互关联, 以面向对象的方式创建应用程序, 从而生成代码简洁、易于阅读且通常不包含大量数字和坐标的应用程序代码。

以下是提供的元素列表:

	应用程序 主要包含应用程序代码(和/或代码模块)以及应用程序的特定设置。程序代码支持 Python 3.13 , 除基本的 Python 函数外, 还包含许多来自内部机器人控制 API(应用程序编程接口)的函数。应用程序通过程序编辑器进行创建和修改。
	动作 是指具有特定方向的工具中心点(TCP)位置, 由一组关节角度唯一定义。动作可通过“动作编辑器”进行创建和修改。
	路径 是一系列位姿序列, 用于描述TCP的平滑运动, 从而描述整个机器人的运动。路径可通过路径编辑器进行创建和修改。
	网格 是可自定义的、最多为三维的点阵, 用于定义预定义托盘的位置等。网格点会被提取出来, 并在程序中用作迭代器。网格可通过“网格编辑器”进行创建和修改。
	I/O 分为数字信号和模拟信号。可通过 I/O 配置菜单进行配置, I/O 的值可在 I/O 菜单中查看。I/O 用于连接外部设备并与其通信。

所有这些元素均存储在AX Portal的数据库中。您可随时通过“配置”菜单下载机器人数据库(以备份完整设置), 或上传新数据库(覆盖当前数据库及其所有元素)。此外, “项目”菜单提供了数据库中所有项目和元素的清晰概览。

4.4 项目菜单

点击控制台中当前选定项目的名称, 即可进入项目菜单。该菜单包含可用元素的概览, 例如姿态、路径、网格和程序。这些元素被分组到多个项目中。任何时候仅能选定一个项目, 且只有选定项目的元素在 AX Portal 的其他菜单中可见、可编辑并可用。

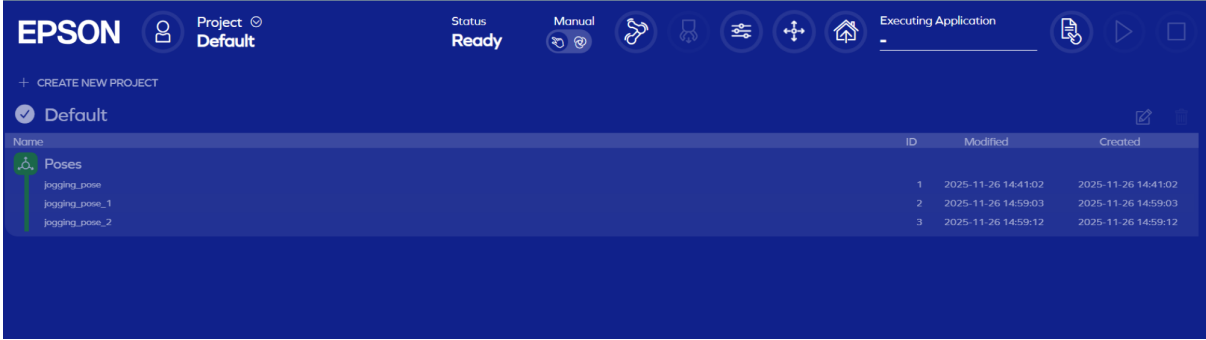


图 4.4: 项目菜单

-
-  点击“创建新项目”按钮可创建一个新的空项目。
 -  点击项目名称左侧的圆圈以选择该项目。
 -  点击项目元素的“复制”按钮, 可在同一项目内创建该元素的精确副本。复制的元素名称会稍作修改, 并在末尾添加“(副本)”字样。
 -  点击项目名称右侧的“编辑”按钮, 可更改该项目的名称。点击项目元素的“编辑”按钮, 可打开相应的编辑器进行查看或修改。
 -  点击项目名称右侧的“删除”按钮, 可删除该项目及其所有元素。点击项目元素的“删除”按钮, 可从项目中移除该特定元素。
-
- 点击某个元素图标 (例如“位姿”、“路径”等) 可显示或隐藏该特定部分中的元素列表。
-
- 通过鼠标将元素从源项目拖拽至目标项目, 即可将其从一个项目移动到另一个项目。拖拽元素时按住 **CTRL** 键, 可在目标项目中复制该元素。
-

4.5 配置菜单

点击主窗口上的“配置”按钮可进入配置菜单。在此菜单中，您可以配置数据库和硬件设置（I/O 和工具）。如果您操作的是用于特定应用的静态机器人，只需进行一次配置，之后仅需通过此窗口启动机器人。您甚至可以通过“设置”菜单中的相应选项，将启动和运行应用程序的步骤自动化。但如果您将机器人用于多种用途，则可能需要更频繁地访问此菜单，并根据不同的项目和目的使用不同的工具甚至不同的数据库。

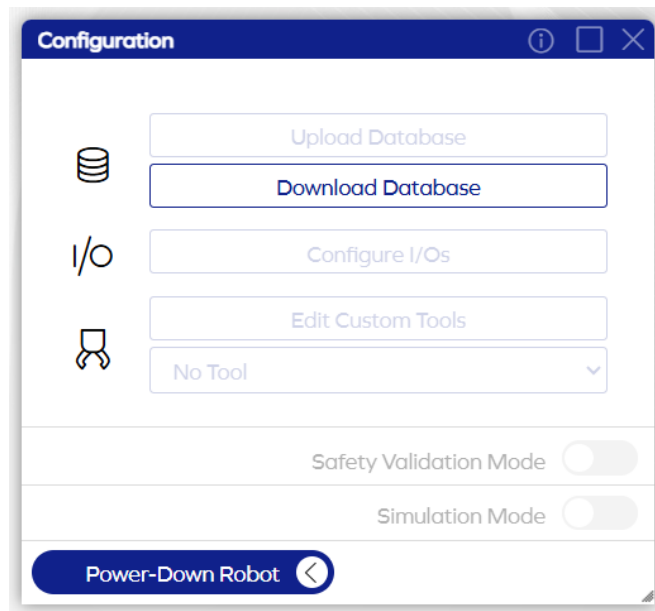


图 4.5: 配置菜单

4.5.1 数据库(下载与上传)

在数据库部分，您可以下载应用程序数据库的当前版本，或上传(并覆盖)另一个兼容版本的数据库。下载数据库将在您的设备上创建数据库 ZIP 文件的本地副本。建议定期备份数据库，特别是当您拥有大量应用程序，且在编写和调试应用程序上投入了大量时间时。上传数据库时，浏览器会提示您从设备中选择 ZIP 文件。选择数据库文件将覆盖当前数据库，并用上传的版本替换它。

通过在一台机器人上下载数据库并将其上传到另一台机器人，您可以将所有程序和元素在机器人之间进行复制。建议定期备份数据库。

另请注意，数据库具有版本控制，并非所有数据库都与 AX Portal 的每个版本兼容。旧版数据库可以导入到新版软件中，但新版数据库无法加载到旧版软件中。换言之，数据库不具备向后兼容性。

4.5.2 I/O 配置

点击“设置 I/Os”按钮以打开“I/Os 配置”窗口。在此处，您可以配置所有 I/O，为数字输入分配预定义功能，并为 I/O 指定自定义名称。在应用程序脚本中使用自定义名称可提高脚本的可读性。在脚本中，您可以直接用 I/O 名称(字符串)替换整数 I/O 标识符(int)。

I/Os Configuration

Main I/Os

Tool I/Os

ID	Configuration	Custom Name	Function	Trigger
DI 01	Input with Pull-Down		Start Execution	Rising Edge
DI 02	Input with Pull-Down		Pause Application	Rising Edge
DI 03	Input with Pull-Down		Stop Application	Falling Edge
DI 04	Input with Pull-Up		No Function	N.A.
DO 01	Output		N.A.	N.A.
DO 02	Output	tool_place	N.A.	N.A.
DO 03	Output	tool_pick	N.A.	N.A.
DO 04	Output		N.A.	N.A.
AIO 01	Output (0-10 V)	pressure	N.A.	N.A.
AIO 02	Output (0-10 V)		N.A.	N.A.
AIO 03	Input (0-10 V)		N.A.	N.A.
AIO 04	Output (0-20 mA)		N.A.	N.A.

ID: 5

ID: 1

ID: 3

ID: 2

ID: 4

Icon	Custom Name	Function	Trigger
①	custom_1	No Function	N.A.
②	custom_2	No Function	N.A.
🔧		Pick/Place Object	Rising Edge
📐		Record Pose	Rising Edge
🔗		Toggling Hand-Guided Control (HGC)	Active While Pressed

Reset

Load Defaults

Save

图 4.5.2.a:I/O 配置菜单

AX系列在最顶端的关节处配备了一个按钮接口。将功能分配给这两个自定义按钮的方式，与在此菜单中将功能分配给其他数字输入的方式完全相同。

请注意，当通过数字输入触发任何功能时，当前控制站的控制权将被撤销。

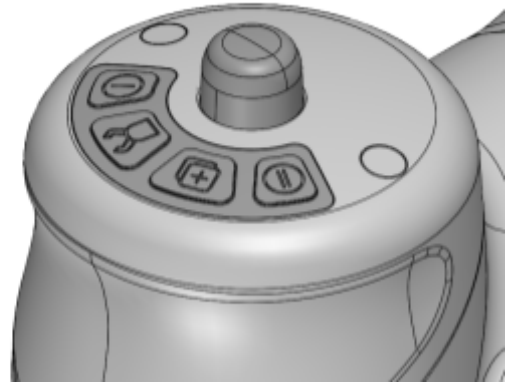


图 4.5.2.b:AX 按钮接口

以下是可用的数字输入预定义功能，您可以将其设置为在上升沿（默认）或下降沿触发：

无功能	没有触发任何操作。此为默认设置。
暂停程序	暂停当前运行中的程序。 这类似于按下驾驶舱上的“暂停”按钮。
中止程序	停止当前运行中或已暂停的程序。 这类似于按下驾驶舱上的“停止”按钮。
开始执行	运行当前选定的应用程序。 这类似于按下驾驶舱上的“运行”按钮。 如果机器人从中断中恢复，此功能将机器人返回中断时的位姿，并继续之前的运动。 这类似于在中断发生后，在 GUI 中弹出的对话框中按下“继续”和“继续运行”按钮。
重置安全	在中断后重置安全。 这类似于在中断发生后，通过 GUI 中弹出的对话框重置安全。
启用 HGC	为所有关节释放使能控制 (HGC) (状态为“就绪”时)。
禁用 HGC	保持所有关节的手动引导控制 (HGC) (状态为手动引导控制时)。
启用/禁用 HGC	在启用和禁用所有关节的 HGC 之间切换。触发的功能取决于当前程序状态。
抓取物体	触发工具拾取功能 (预定义或与数字输出关联)。这类似于按下驾驶舱上的“拾取”按钮。
放置物体	触发工具放置功能 (预定义或关联至数字输出)。这类似于按下驾驶舱上的“放置”按钮。
张开/关闭夹具	根据当前工具状态，在工具的拾取和放置功能之间切换。
已记录的动作	录制当前机器人位姿并将其保存为“button_pose” (或类似名称)。
手动操作模式	将机器人切换到手动操作模式。 这类似于使用驾驶舱内的开关切换模式。
自动操作模式	将机器人切换至自动操作模式。 这类似于使用驾驶舱中的开关切换模式。

4.5.3 工具配置

在“工具”部分，您可以配置机器人的工具。如果您准备集成自定义工具（例如来自第三方公司的工具，或您自己为特定应用开发的专用工具），可以使用“编辑用户工具”选项。这将打开已存用户工具的列表，其中包含以下选项：

	点击“创建新工具”按钮，打开工具编辑器，用于定义工具的运动学和动态性能以及拾取和放置触发条件。
	点击“搜索”按钮，可按名称筛选自定义工具列表。输入三个或更多字符即可启用搜索过滤功能。
	点击“编辑”按钮（或在元素任意位置双击），将在单独窗口中打开工具编辑器，以便查看或修改。
	点击“删除”按钮将不可逆地从数据库中移除该工具。

4.5.4 工具编辑器

工具编辑器分为四个部分（见下图）。要填写这些值，您可能需要研究即将集成到 AX Portal 中的工具的数据表，其中可能详细指定了该工具的物理和电气特性。

在第一部分中，您需要定义工具中心点（TCP），即工具在空间中进行交互时的位姿。对于用于抓取物体的工具而言，这意味着TCP定义了抓取过程中工具被视为所在的位置。TCP 是在附着于顶部连接板正中心的坐标模式中定义的，该坐标模式的方向与基坐标系相同（假设所有关节角度均为零度）。如果未定义 TCP，则该坐标模式即为默认工具坐标系（x、y、z、X轴旋转、Y轴旋转和Z轴旋转均为零）。在此基础上，您可以平移和旋转 TCP，使其与自定义工具的交互点相匹配。

下一步，需要调整自定义工具的动态属性。重力补偿和碰撞检测等功能的质量取决于机器人动态模型的精确度，因此测量并定义工具的这些属性非常值得。您需要定义工具的总质量以及质量中心，质量中心（与 TCP 一样）在机器人的“默认工具坐标系”中定义。惯性旋转瞬间可以手动输入（如果已知），也可以自动近似（AX Portal 无法识别工具的形状和质量分布）。

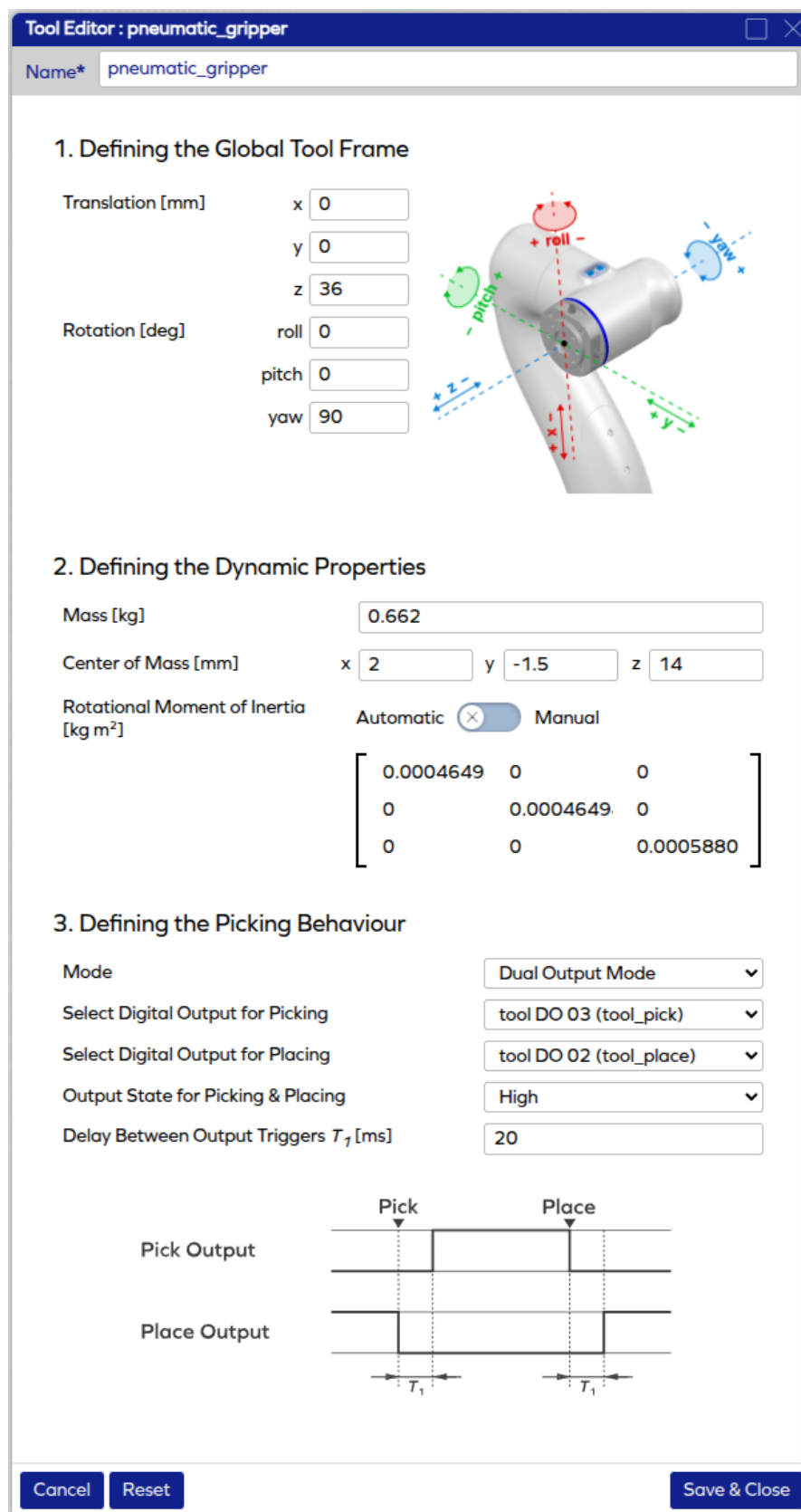


图 4.5.4: 工具编辑器

最后, 如果您的工具配备了拾取物体的机构, 您可以在此处定义哪些数字输出由“工具拾取”和“工具放置”功能触发, 这些功能可通过控制台和脚本函数访问。可用的拾取行为如下:

4.5.4.1 单输出模式

这是所有拾取行为中最简单的模式。您只需定义一个用于在拾取和放置之间切换的数字输出, 并可设定拾取操作由 HIGH 信号还是 LOW 信号触发。



图 4.5.4.1: 单输出模式示例 (高电平用于拾取, 低电平用于放置)

4.5.4.2 双输出模式

此拾取行为允许您为拾取和放置分别定义独立的数字输出。拾取/放置切换后, 两个输出将以 (最小) 时间延迟 T_1 (单位为毫秒) 依次触发。此外, 您还可以定义活动状态对应 HIGH 信号还是 LOW 信号。

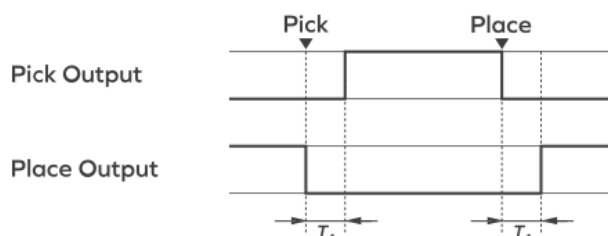


图 4.5.4.2: 双输出模式示例, 其中 HIGH 作为活动触发状态

4.5.4.3 分离触发模式

在分离触发模式下, 您可以定义两个数字输出: 一个用于定义方向 (拾取或放置), 另一个用于定义触发 (是否触发所选方向)。对于方向输出, 您可以选择使用 HIGH 信号还是 LOW 信号来执行拾取操作 (另一个则用于放置)。对于触发输出, 您同样可以选择使用 HIGH 还是 LOW 信号进行触发。此外, 您还可以定义从选择方向到触发所选方向之间的 (最小) 时间延迟 T_1 (以毫秒为单位), 以及定义触发保持激活状态的 (最小) 持续时间 T_2 (以毫秒为单位)。

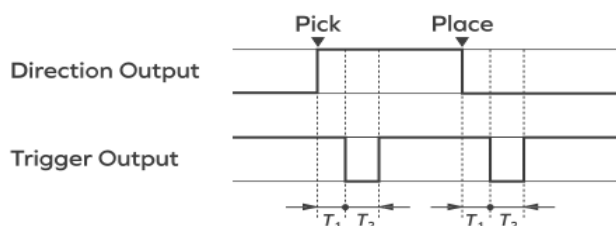


图 4.5.4.3: 分离触发模式示例, 其中 HIGH 作为拾取方向, LOW 作为触发

4.5.5 机器人上电

在加载数据库并设置 I/Os 和工具后,即可启动机器人。通常您无需更改机器人的初始设置,只需直接将“启动机器人”滑块完全向右拖动,即可触发初始化例程。

如果您不想启动硬件,而是仅修改数据库内容(程序、位姿等)且不使用任何硬件(例如不移动机器人、不读写I/O),可以先启用“仿真模式”开关,然后将“启动机器人”滑块拖动到右侧。请注意,在仿真模式下,所有动作均以实时方式模拟。

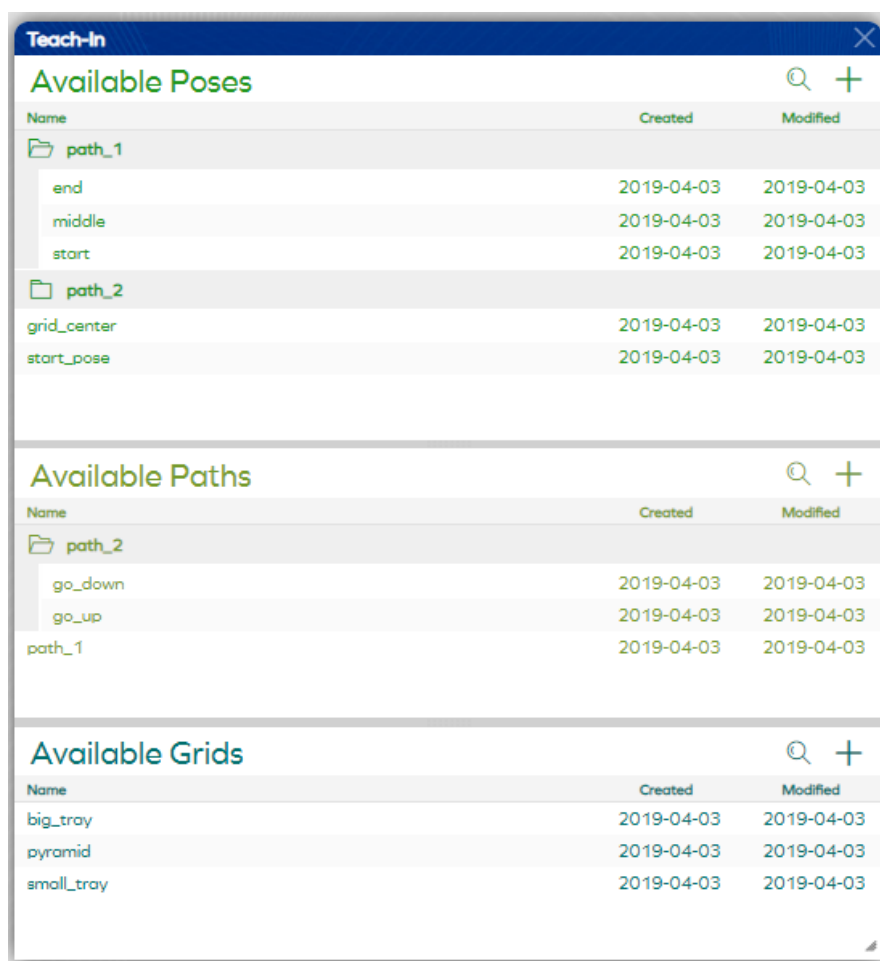
在机器人上电期间,您将无法更改机器人的任何初始配置。您只能下载数据库或选择“休眠机器人”,如果机器人处于空闲或待机状态,建议选择此操作。

如果您以安全主管身份登录,则可在此处访问“安全验证模式”开关。启用此模式会指示软件忽略位置、速度和力限制值,可用于验证在限值被违反时安全装置是否确实触发。请注意,在此模式下使用任何运动控制功能仍会覆盖限值。

4.6 示教菜单

点击主窗口上的“示教”按钮可进入示教菜单。该菜单包含当前所选项目中所有可用动作、路径和网络的概览。

姿态和路径可以创建、编辑、复制、删除，甚至直接执行。另一方面，网络是整理重复且等间距坐标的有用工具。请注意，网络并非直接用于移动机器人，而是将一组坐标值打包并存储起来，以保持程序脚本的结构化，并避免您手动输入坐标。可以从网络中提取网格点列表以满足各种需求，不过通常网络的设置使其点直接代表TCP(工具中心点)坐标，这些坐标随后将作为机器人运动相关函数的输入值。



The screenshot shows the 'Teach-In' window with three sections: 'Available Poses', 'Available Paths', and 'Available Grids'. Each section has a search icon and a plus icon for adding new items. The 'Available Poses' section lists poses under 'path_1' and 'path_2'. The 'Available Paths' section lists paths under 'path_2' and 'path_1'. The 'Available Grids' section lists grids: 'big_tray', 'pyramid', and 'small_tray'.

Available Poses		
Name	Created	Modified
path_1		
end	2019-04-03	2019-04-03
middle	2019-04-03	2019-04-03
start	2019-04-03	2019-04-03
path_2		
grid_center	2019-04-03	2019-04-03
start_pose	2019-04-03	2019-04-03

Available Paths		
Name	Created	Modified
path_2		
go_down	2019-04-03	2019-04-03
go_up	2019-04-03	2019-04-03
path_1	2019-04-03	2019-04-03

Available Grids		
Name	Created	Modified
big_tray	2019-04-03	2019-04-03
pyramid	2019-04-03	2019-04-03
small_tray	2019-04-03	2019-04-03

图 4.6: 示教菜单

	点击“+/添加”按钮，可在新窗口中打开位姿、路径或网格编辑器。本节后文将详细介绍所有这些编辑器。
	点击“搜索”按钮，可按名称筛选姿态、路径或网格列表。输入三个或更多字符即可启用搜索过滤功能。
	点击“播放”按钮可将机器人移动至指定位姿或执行路径。请注意，若机器人尚未处于该路径的起始位姿，执行路径时会先将其移动至该起始位姿。
	点击“复制”按钮可创建该位姿或路径的精确副本。复制的元素名称会稍作修改，并在末尾添加“(副本)”后缀。
	点击“编辑”按钮（或在元素任意位置双击），可在单独窗口中打开该位姿、路径或网格以进行查看或修改。
	点击“删除”按钮可将位姿、路径或网格从数据库中永久删除。
	点击文件夹可显示或隐藏该文件夹内的姿势、路径或网格。只有名称中包含一个或多个“/”字符（例如“folder/subfolder/name”）的元素才会显示在文件夹（及其子文件夹）中。

4.6.1 位姿编辑器

位姿编辑器允许用户创建新位姿或修改现有位姿。请注意，更改位姿的坐标或配置不会自动移动机器人。若需移动机器人，应改用精细控制。

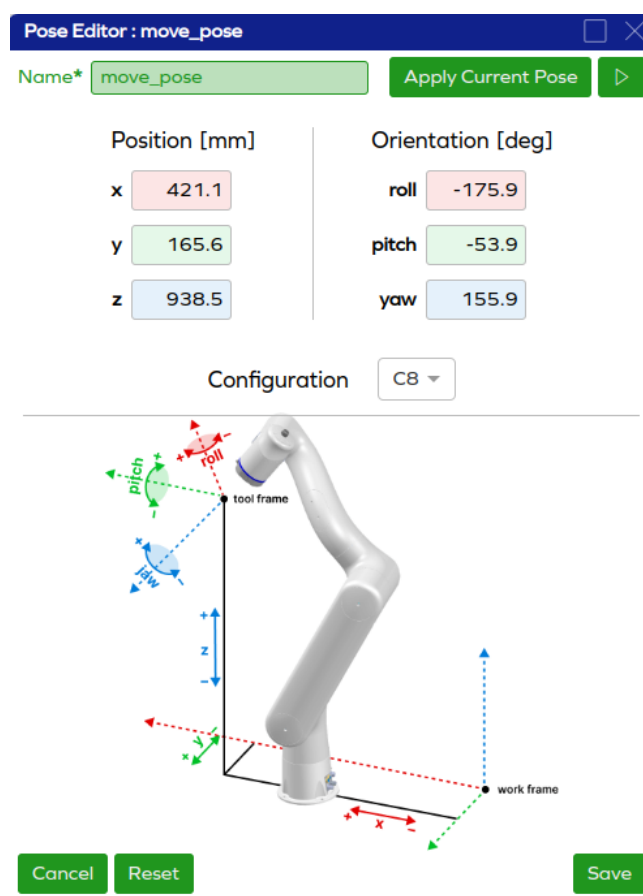


图 4.6.1:位姿编辑器

名称:

指定位姿在数据库中的存储名称。可在名称中使用“/”字符，将位姿按文件夹式结构进行分组。

位置:

表示相对于全局工件坐标系的工具中心点 (TCP) 位置(单位为毫米)。请注意，可以输入无效的位置值(例如机械臂无法到达的位置)。不过，这些值会以红色高亮显示，且点击演示按钮将导致错误。允许这样做是因为当使用不同的坐标模式和偏移量时，该位姿可能是有效的。

方向:

表示相对于全局工件坐标系的工具中心点(TCP)方向，单位为度。方向采用航海角表示。与位置值类似，无效的方向值也会以红色高亮显示。

配置:

指定机械臂的配置。用户可通过下拉菜单从所有可用配置(C1–C8)中进行选择。请注意，修改配置不会改变TCP的位置或方向，它仅影响用于实现指定位姿的关节角度。

位姿编辑器中的其他操作

保存	点击“保存”按钮将当前位姿以指定名称保存。若名称无效, 则无法保存该位姿。
取消	点击“取消”按钮将丢弃编辑器中所有未保存的更改并关闭窗口。
重置	点击“重置”按钮将撤销所有未保存的更改, 并将位姿恢复到上次保存的状态。
 演示	点击窗口顶部的“演示”按钮, 将命令机器人移动到定义的位姿。
应用当前位置	点击“应用当前位置”按钮, 将更新所有字段以反映机器人的当前位姿数值。

4.6.2 路径编辑器

路径编辑器允许用户创建新路径或修改现有路径。它分为三个可编辑部分：名称、初始位姿和该路径的路径段。

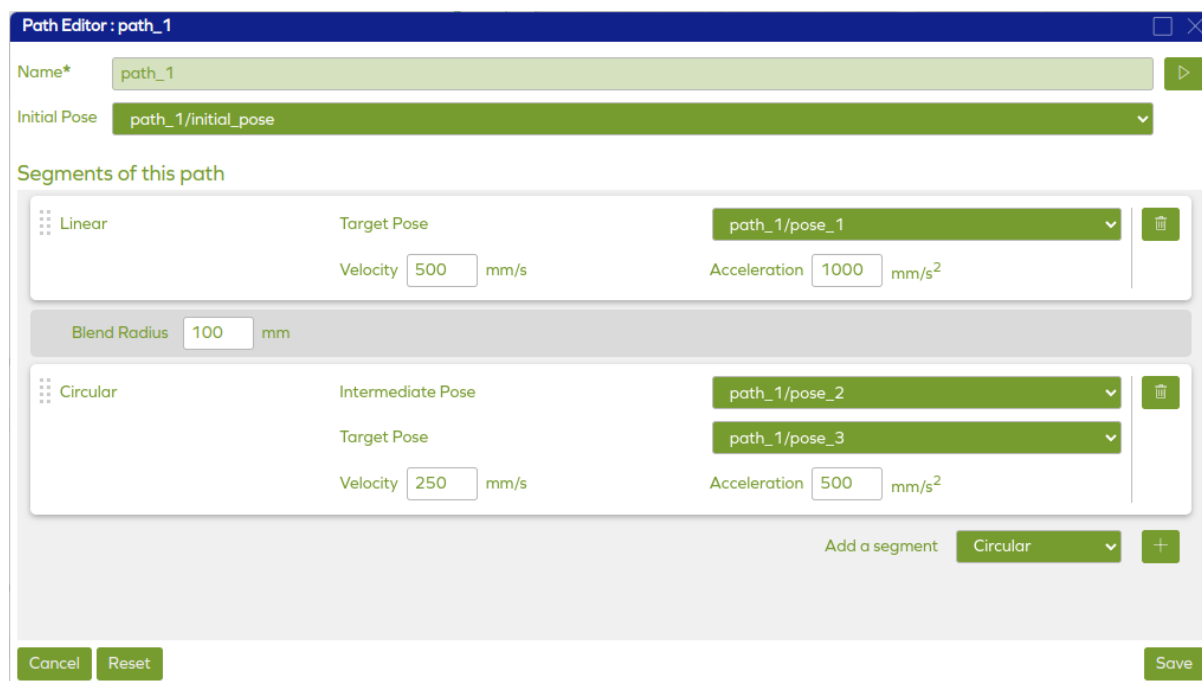


图 4.6.2: 路径编辑器

名称:

指定路径在数据库中的存储名称。可在名称中使用“/”字符，将路径按文件夹结构进行分组。

初始位姿:

定义路径的起始位姿。机器人将首先使用默认的速度和加速度设置移动到该位姿，然后依次执行定义的路径分段。

此路径的各部分:

列出并定义构成路径的各个线段。可用的线段类型如下:

- 线性段
- 圆弧运动段
- 坐标段
- 位姿分段
- 方向段

关于每个线段的更详细描述，请参见“线段”部分。

要添加新线段，请从右下角的下拉菜单中选择所需的分段类型，然后点击加号 (+) 按钮。添加线段后，可通过拖放功能自由调整其顺序。

每个分段都需要指定一个目标姿势，该姿势定义了该分段的最终姿势。目标位姿必须在“位姿编辑器”中预先定义，然后可通过下拉菜单进行选择。圆形分段还需额外指定一个中间位姿，该位姿也必须从预定义位姿列表中选择。

每个线段均可设置速度和加速度参数。若未指定，执行时将使用全局默认值。

可配置混合半径以控制段与段之间的过渡平滑度和效率。它定义了当前段起始位姿相对于前一段结束位姿的最大允许偏差(单位为毫米)。

路径编辑器中的其他操作

路径编辑器还提供其他功能，用于有效管理、编辑和组织路径。

保存	单击“保存”按钮将当前路径以指定名称保存。若未输入有效名称，则无法保存路径。
取消	单击“取消”按钮将放弃编辑器中所有未保存的更改并关闭窗口。
重置	单击“重置”按钮将撤销所有未保存的更改，并将路径恢复到上次保存的状态。
 演示	单击窗口顶部的“播放”按钮，将命令机器人执行已定义的路径。如果机器人的当前位姿与路径指定的初始姿势不匹配，它将先移动到该初始姿势，然后才开始执行。
 删除	单击某段路径旁边的“删除”按钮，将从路径中移除该段。

4.6.3 网格编辑器

创建新网格时，将打开网格编辑器。该编辑器会引导用户逐步完成所有可用的网格选项设置。使用“下一个”和“返回”按钮在向导中导航，直至按需定义出所需的网格点有序列表。若编辑已存在的网格，网格编辑器将直接跳转至手动操作模式。

第一步，用户必须选择网格维度。通常，网格被解释为二维数组，但它也可以是一个不依赖第二维度的简单点列表（例如从传送带上抓取物体），甚至可以是三维结构（例如将物体堆叠在箱子中）。网格初始化时包含完全对齐的点，但请注意，单个网格点在后续操作中始终可以被修改、添加或删除。

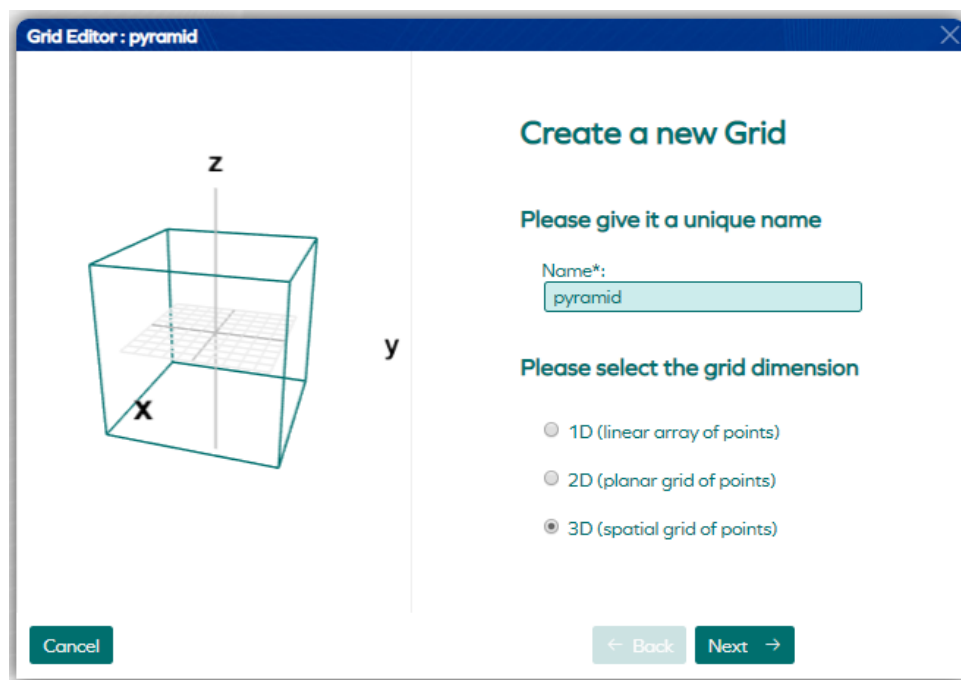


图 4.6.3.a: 网格编辑器 (第一步)

在第二步中，用户必须选择所选网格的角点。不同维度的网格具有不同的角点数量，对于二维网格，角点通常标记为“A”到“D”。用户可直接在输入框中输入角点的坐标值（以机器人的基坐标系为单位），或直接引导末端执行器的工具中心点（TCP）移动至网格的角点，并按下相应的“示教”按钮以保存坐标。

一旦教导的角点足够定义网格（例如，二维网格至少需要 3 个点），用户即可点击“网格校正”按钮来评估给定的角点，并自动填充缺失的数值，使其符合平行四边形（二维）或长方体（三维）的形状。如果启用了额外的“自动对齐”标志，评估过程将考虑横轴和垂直轴及平面，并尝试将生成的网格沿这些轴和平面进行对齐（如果足够接近）。

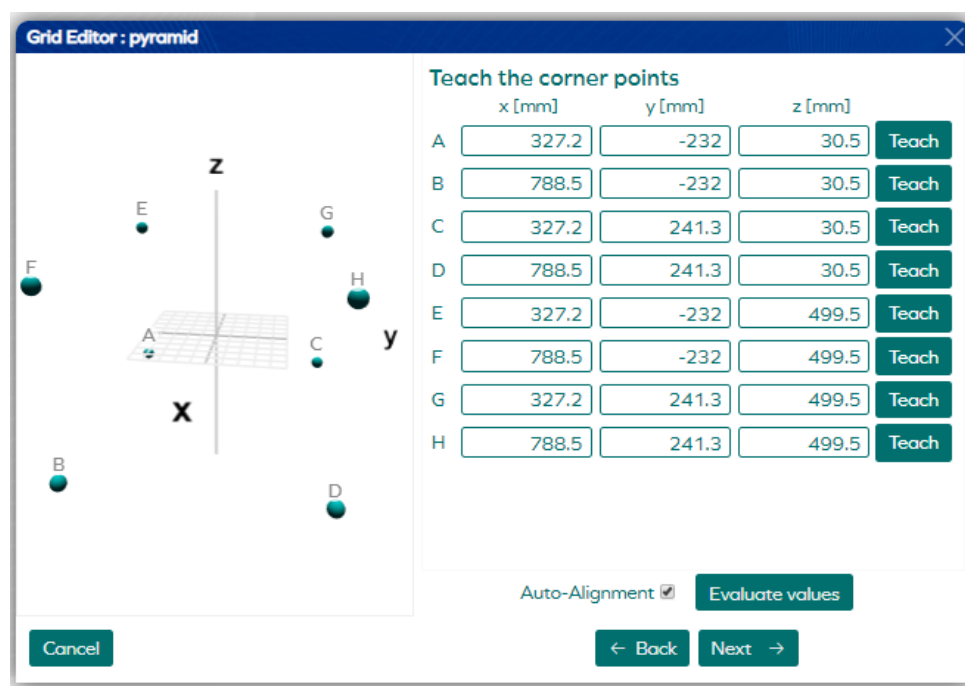


图 4.6.3.b: 网格编辑器 (第二步)

在第三步中, 系统会要求用户选择点数、特殊扩展选项和迭代方向。让我们详细看看这些可用的选项:

- “点数”可针对每个维度单独设置。若难以区分各维度 (例如行和列), 请参考前文所述的关键点示教标签。
- 对于二维网格, 存在一种扩展选项, 用于在每组四个相邻点之间添加中间点。对于三维网格, 该选项不仅适用于每个网格平面, 也适用于空间中每组八个相邻点。
- 迭代方向 (即点被返回并作为编程循环中的迭代器使用的方向) 由一个起始点和每个方向上的一个向量确定。最后一个方向始终由先前选定的选项决定, 仅供参考和可视化验证。下拉菜单中的字母与该网格的角点标签相对应。

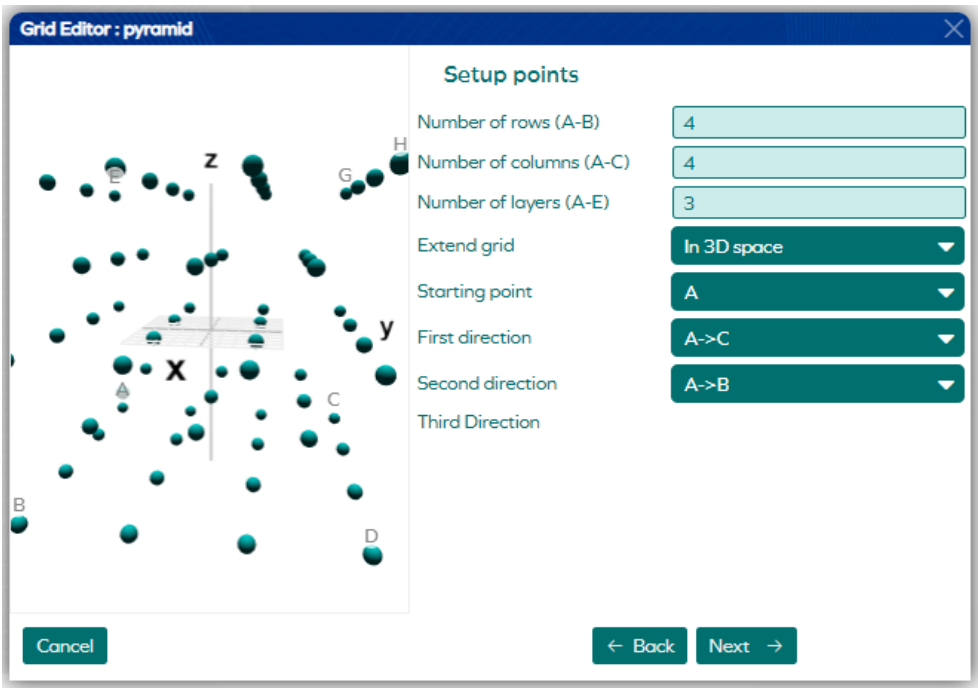


图 4.6.3.c: 网格编辑器 (第 3 步)

最后，“网格点列表”已就绪，可供手动编辑。在此，可以复制、删除、移动单个点，并手动更改坐标值。同时，可视化模型显示了网格点的顺序及其相对坐标。

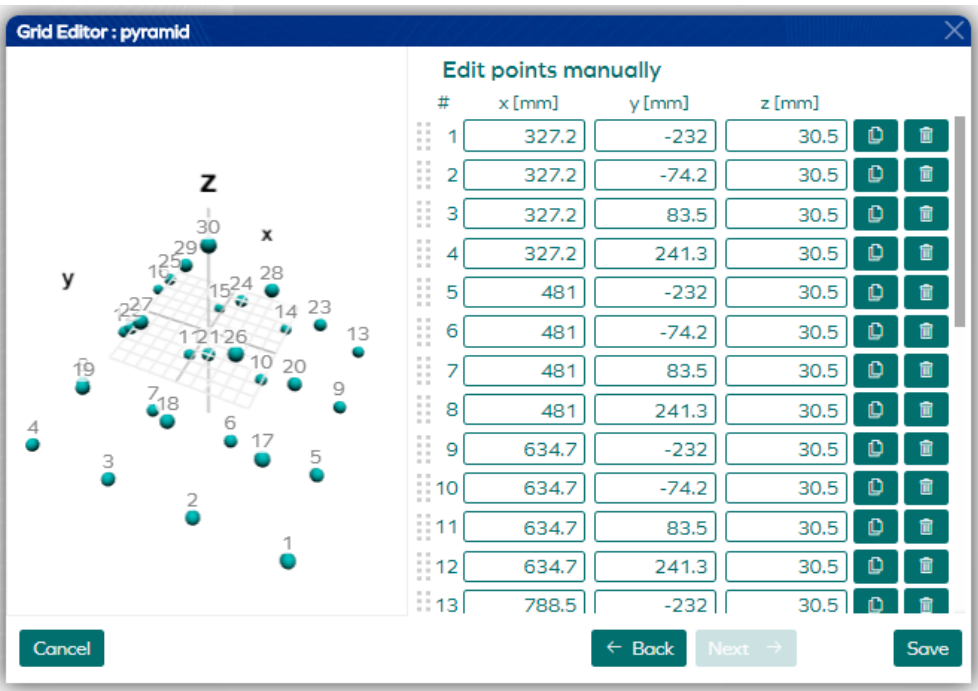


图 4.6.3.d: 网格编辑器 (第 4 步)

编辑过程中，您可以随时返回查看先前设置的选项和数值。您也可以更改这些选项，但这将重置后续窗口中的所有更改。

4.7 应用程序菜单

点击主窗口上的“应用程序”按钮可访问应用程序菜单。该菜单包含所有已保存应用程序的概览以及上次执行的应用程序的输出结果。用户还可通过此菜单进入“程序编辑器”窗口，以创建新程序或编辑现有程序。

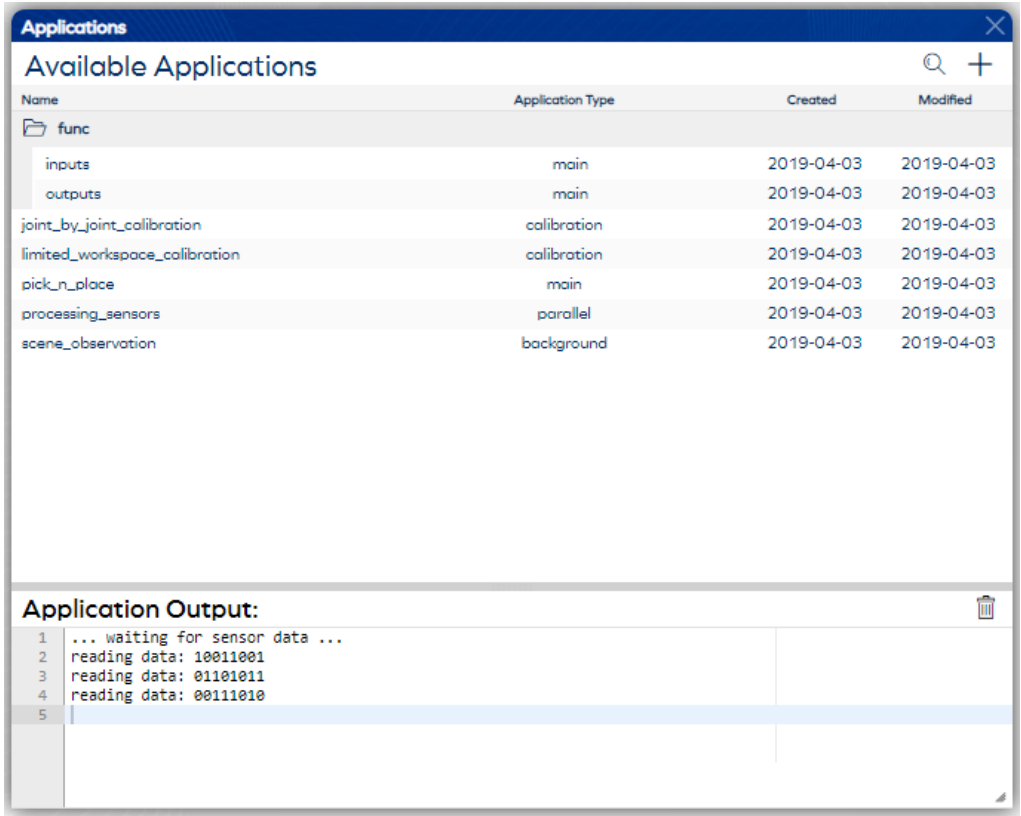


图 4.7: 程序列表菜单

	点击“+/添加”按钮可创建新程序。这将打开程序编辑器，其功能将在本节后文详细说明。
	点击“搜索”按钮可按名称筛选程序列表。输入三个或更多字符即可启用搜索过滤功能。
	点击“播放”按钮可运行程序。请注意，仅“主”程序可通过此方式可执行，因为驾驶舱中仅可选择“主”程序。
	点击“复制”按钮可创建该应用程序的精确副本。复制的元素名称会稍作修改，并在末尾添加“(副本)”后缀。
	点击“编辑”按钮（或在元素任意位置双击），可在单独窗口中打开该程序以进行查看或修改。
	点击“删除”按钮可将应用程序从数据库中永久删除。这将同时删除程序脚本以及应用程序的特定设置。
	点击一个文件夹以显示或隐藏该文件夹内的程序。只有当元素名称包含一个或多个“/”字符时（例如“folder/subfolder/name”），它们才会显示在文件夹（及其子文件夹）中。您可以点击出现的“编辑”按钮进入文件夹编辑器。

4.7.1 程序编辑器

“程序”菜单中的“已存程序”部分列出了所有先前保存的程序。您可以通过点击“编辑”按钮修改现有程序，或通过点击“创建新程序”按钮创建新程序。无论哪种情况，都会弹出“程序编辑器”窗口（ ）。其功能在两种情况下均相似。在深入探讨细节之前，让我们快速浏览一下程序编辑器的主要按钮：

保存	点击“保存”按钮，程序将以指定名称保存。程序必须命名才能保存。在名称中使用“/”字符可将程序按文件夹结构分组。
取消	点击“取消”按钮，将拒绝编辑器中所有未保存的更改，并关闭编辑器。
▶ 测试	点击窗口顶部的“测试应用程序”按钮，应用程序将以测试模式运行。测试模式意味着程序脚本会执行，但不会阻止您继续编辑该应用程序。在测试应用程序后添加的更改将不会被执行。

4.7.2 应用程序代码

程序编辑器的“代码”选项卡是三个子窗口中的主窗口。该部分显示应用程序代码，并可选地显示包含 AX Portal 函数和元素的库，以便将其插入到应用程序代码中。内置的编程语言为 **Python 3.13**，包含其大部分基础功能，包括一些预装的 Python 包。此外，AX Portal 还提供了一个庞大的函数库，旨在简化机器人程序的编写（例如移动机器人、读取传感器数据、与用户通信等）。有关这些函数的详细信息，请参阅代码文档。

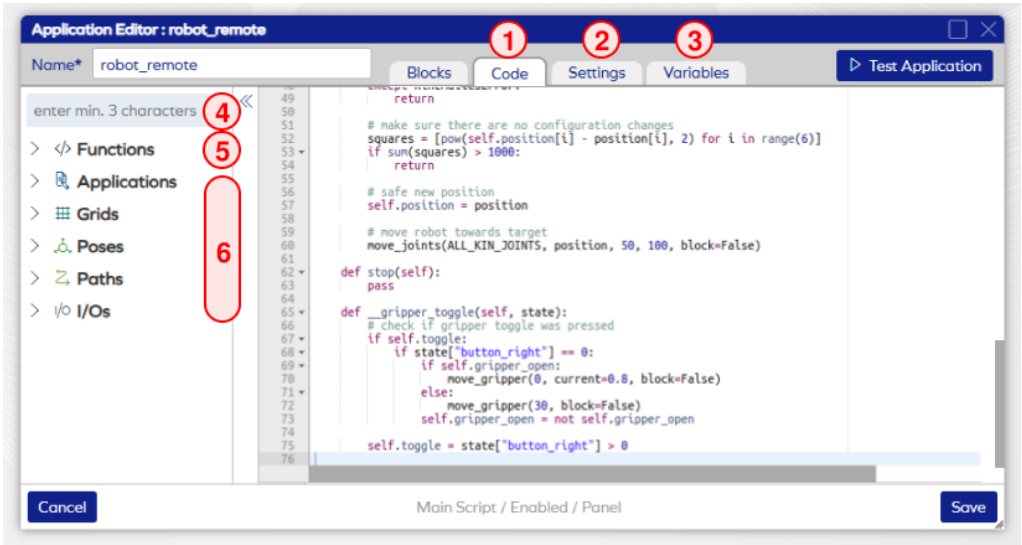


图 4.7.2: 程序编辑器 (程序脚本部分)

- 1. 在此处编写 Python 程序代码，2. 修改应用程序特定设置，3. 定义和修改应用程序变量，4. 搜索过滤器，5. 函数库，6. 可用元素

点击编辑器左侧的某个功能图标或元素图标，即可打开元素面板。在此处，您将看到所有可用功能的完整列表。此外，您还可以通过双击所有元素将其集成到程序脚本中，因此无需记住元素的名称。

提示：通过移除双击函数时插入代码中的可选参数，可提高 Python 代码的可读性。例如，您可以编写

```
move_joints(joints, joint_position)
```

，而非

```
move_joints(joints, joint_position, joint_velocity=None, joint_acceleration=None,
relative=False, block=True)
```

4.7.3 程序设置

“程序设置”选项卡包含一些程序的附加设置，包括

- 一个可禁用的“可执行”标志，例如，如果您的应用程序仍在构建中且不应被执行，
- 一个“在面板界面中显示此程序”的标志，
- 上传和替换在项目面板中显示的图片的功能，
- 应用程序标签，用于将程序名称翻译为面板界面当前选定的语言，
- 以及最重要的“程序类型”，相关说明见下一节：



图 4.7.3.a: 程序编辑器(设置部分)

“程序类型”指定了应用程序应在何种上下文中运行，因此决定了应用程序的执行方式以及该应用程序中可使用的功能：

- 主要脚本包含待执行的主应用程序。它们旨在使用所有可用的脚本函数，并直接控制机器人及其外部设备。通常，一个简单的应用程序可以写在一个脚本中，但对于长或复杂的应用程序，或者为了提高可读性，将应用程序拆分为多个按顺序运行的子脚本会很有用。不过，多个主要脚本之间从不并行运行，若需并行处理，则需使用“并行脚本”。您还可以使用“import”语句在主要脚本之间导入函数。请注意，控制台中仅可选择主要脚本。
- 另一方面，并行脚本在独立线程中运行，因此专为与主应用程序并行运行而设计。这在某些场景下非常有用，例如持续读取传感器数据，或利用机器人移动的空闲时间进行耗费计算资源的运算。并行脚本不允许使用所有脚本函数（例如控制机器人），因为如果不加注意，这可能会严重破坏程序流程或引发意外的运动行为。当主应用程序暂停时，所有并行脚本都会暂停；一旦主应用程序停止运行，它们便会终止。
- 后台脚本是在 AX Portal 后台运行的脚本，其运行状态与 AX Portal 本身无关，因此适用于持续观察、检查或记录数据，或为项目定义初始全局变量。每个项目只能包含一个后台脚本。若已定义该脚本，则在加载相应项目时（例如启动项目或切换项目时）会自动执行。后台脚本不受状态限制的约束，且若包含循环，可能会一直运行直至机器人断电。每当后台脚本开始或结束运行时，系统都会显示相应通知。您可通过控制台启动、重启或停止后台脚本。



图 4.7.3.b: 后台脚本控制器

4.7.4 应用程序变量

“应用程序变量”选项卡提供了一个拖放界面，用于向该应用程序添加不同类型的状态变量。状态变量仅可在主要脚本中使用。它们在面板界面中发挥作用，例如，非常适用于监控应用程序的状态和进度，或为应用程序指定特定的输入值。 请分别使用 `set_state_variable(...)` 和 `get_state_variable(...)` 函数接口来写入和读取状态变量。

以下是变量类型的概述及其在面板界面中的显示方式：

布尔值型	布尔值以简单的开/关滑块形式表示，状态为启用或禁用。它们的读写操作采用 Python 的 bool 类型，其中启用对应 True ，禁用对应 False 。
整数	整数以允许输入数字的文本字段形式呈现，编辑模式下包含用于增减数值的 +/- 按钮。它们的读写操作采用 Python 的 int 类型。
浮点数	浮点数以文本字段的形式呈现，在编辑模式下允许输入数值。它们的读写操作采用 Python 的 float 类型。
滑块	百分比变量在编辑模式下以滑块形式呈现，具有强制性的最小值和最大值。它们的读写操作均采用 Python 的浮点数类型。
文本	文本变量在编辑模式下以文本字段的形式呈现。它们的读写操作均采用 Python 的 str 类型。
下拉菜单	下拉列表变量在编辑模式下以下拉列表的形式呈现。它们的读写操作均采用 Python 的 str 类型。

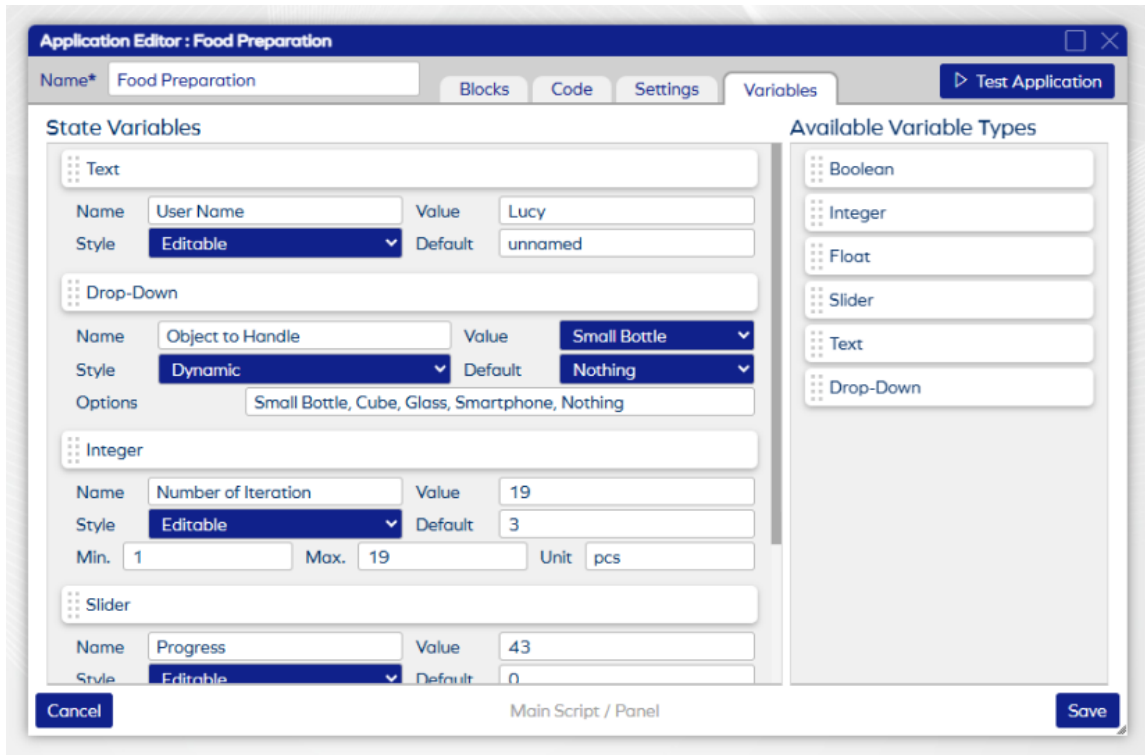


图 4.7.4: 程序编辑器 (变量部分)

以下是对变量选项的简要概述，这些选项会根据变量类型而有所不同：

姓名	为变量指定一个合适的名称，后续可通过该名称以字符串形式访问该变量。
----	-----------------------------------

数值	变量的值包含该变量的当前设置值，该值将以上文所述的 Python 对应类型进行读写。
默认	变量的默认值是指在面板界面中使用相应按钮重置变量时，该变量所采用的数值。
样式	变量的样式定义了该变量是否、何时以及如何可编辑。
最低值	数值类型允许为该数值设置最小边界。若输入的值低于最小值，则将应用最小值。边界对百分比变量是强制要求的，而对整数/浮点数则是可选的。
最大	数值类型允许为该数值设置最大限制。若输入的值高于最大值，则将应用最大值。对于百分比变量，限制是强制性的；对于整数/浮点数，限制是可选的。
单位	对于数值类型，您可以为数字添加可选的单位字符串。
选项	下拉列表变量的选项可以以逗号分隔的字符串形式输入。

以下是对变量样式(定义变量使用方式)的简要概述：

- 可编辑变量允许在运行程序之前更改其数值，但运行中不可更改。
- 受保护变量是可编辑的，作为保护层，您必须先点击编辑按钮修改值，再点击保存按钮接受值的变化。
- 动态变量可编辑，甚至可以在运行时动态更改其数值。
- 动态和保护变量兼具受保护变量和动态变量的特性，因此可在运行时通过点击编辑按钮进行修改。
- 只读变量不可编辑，其值只能由应用程序更改。它们通常用于监控程序的状态或进度。
- 隐藏变量不会在项目面板中显示，但可以像其他状态变量一样进行读写。

4.7.5 共享全局变量

除了上述特定于应用程序的状态变量外，还可以定义共享全局变量和共享变量，以便在并行运行的多个脚本或同一应用程序的多个会话之间共享数据：

- 共享变量仅在单个应用程序的运行期间存在。使用 `set_shared_variable(...)` 和 `get_shared_variable(...)` 函数，可在该应用程序中并行或顺序运行的多个脚本之间共享值。
- 全局变量不仅在应用程序的脚本之间共享值，而且在所有应用程序之间共享，即使在机器人重启后仍然存在。因此，请使用 `set_global_variable(...)` 和 `get_global_variable(...)` 函数。

4.7.6 文件夹编辑器

如果您的程序名称中包含字符“/”，该程序将自动被归类到文件夹结构中。您可以通过将鼠标悬停在文件夹上时出现的编辑按钮来编辑该文件夹。这将打开文件夹编辑器。该编辑器仅包含用于翻译文件夹名称的标签（类似于程序编辑器）。文件夹标签和程序标签都会显示在面板界面中，这样面板界面就可以轻松设置为支持多种语言。

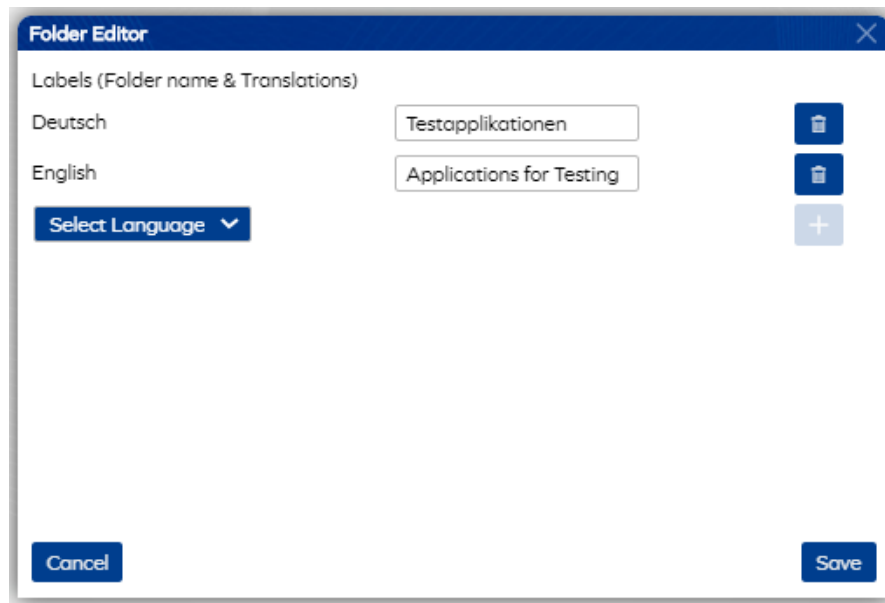


图 4.7.6: 文件夹编辑器

4.7.7 模块编程

程序编辑器的“模块”选项卡包含一个图形化编程界面。通过将编程模块拖放至程序流程区域，用户可以创建简单直观的程序流程。

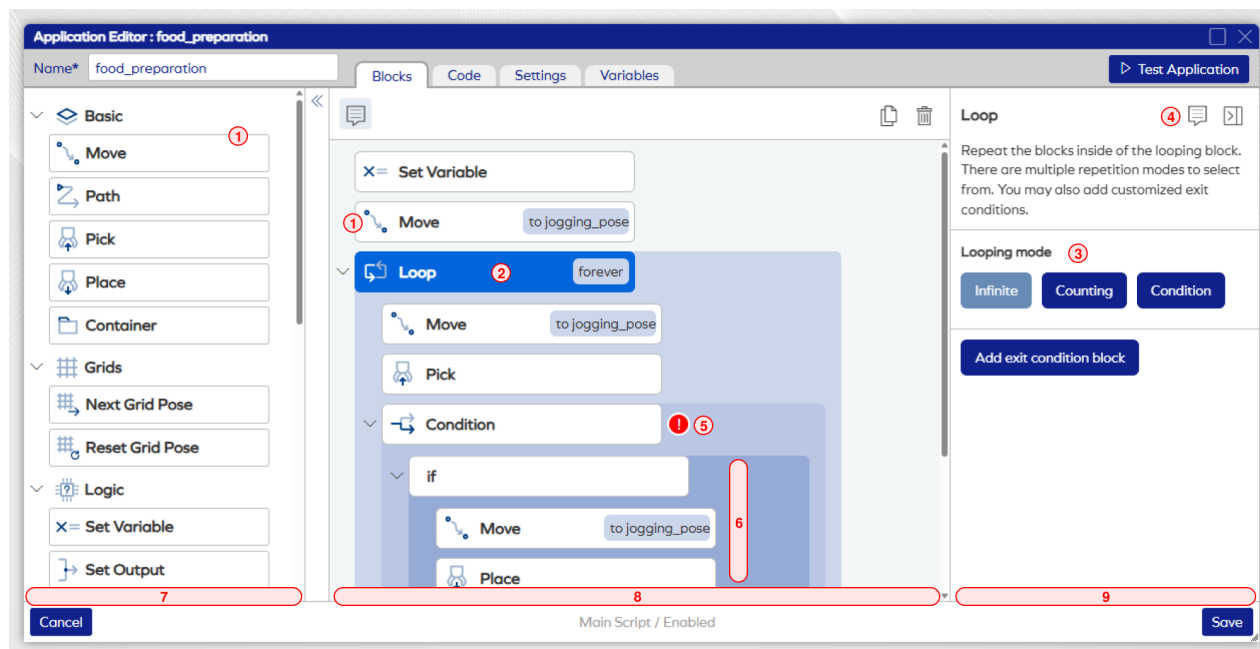


图 4.7.7: 程序编辑器 (模块编程)

1. 将模块拖入程序区域, 2. 选择要修改的模块, 3. 修改模块详情, 4. 注释, 5. 错误指示器, 6. 模块容器, 7. 可用模块, 8. 程序流程, 9. 模块详情

在程序流程中选择一个程序模块即可查看其详细信息。某些程序模块的详细信息必须填写才能使程序可执行 (例如, 对于运动程序模块, 您需要提供机器人应移动到的位置信息)。其他程序模块的详细信息则是可选的 (例如, 若未提供则使用默认值), 或可用于提高程序的可读性 (例如注释)。如果某个模块定义不正确 (例如未定义必填参数, 或提供了无效数值), 程序流程中将显示错误指示符, 选择该模块时, 模块详情中将显示错误原因。

某些模块在添加时会在程序流程中生成彩色容器。这是图形化的帮助提示, 表明这些模块被分组在一起, 且始终按顺序 (自上而下) 执行。循环和条件语句便是此类模块的典型代表, 它们分别负责例如多次执行一组模块, 或仅在满足特定条件时执行该组模块。

此外, 选择的模块可以被删除, 或复制 (并拖动到程序中的目标位置)。

4.7.7.1 使用程序变量和表达式

在详细讲解所有块编程模块之前, 请注意: 在模块详情中的大多数文本字段中, 均可使用数学和逻辑表达式。这些字段还支持插入预定义的应用程序变量值, 以及数字输入和模拟输入的数值。

通过一个示例来解释这一点最为直观。假设我们定义了以下变量:

State Variables

Boolean

Name: Value: ☐

Style: Default: ☐

Integer

Name: Value:

Style: Default:

Min: Max: Unit:

图 4.7.7.1.a:变量示例

有效数值表达式的示例：

有效逻辑表达式的示例：

value (number)

42

value (number)

-7 * (8.2 - 3)

value (number)

{counter} + 1

value (number)

{Tool AIO 02}

value (bool)

FALSE

value (bool)

NOT {flag}

value (bool)

{flag} AND ({counter} > 5)

value (bool)

{Main DIO 09}

图 4.7.7.1.b:数值与逻辑表达式示例

变量和输入的数值也可以注入到普通文本字段中：

value (text)

The counter is set to {counter}.

图 4.7.7.1.c:文本表达式示例

4.7.7.2 移动模块

“移动”模块执行移动函数，将机器人从当前位置移动到目标位置。用户必须选择运动风格
.....

- 点到点运动 → 以时间最优方式移动至目标，对轨迹无限制
- 直线运动 → 沿直线轨迹运动至目标，方向进行线性插值
- 圆弧运动 → 沿圆弧轨迹移动至目标，方向进行直线运动，且用户必须在圆弧轨迹上指定一个中间位置

.....以及目标格式。

- 位姿名称 → 选择(先前保存的)目标姿势名称
- 网格名称 → 移动到(先前保存的)网格的点上，初始时将移动到该网格的第一个点，然后使用“下一个网格位姿”模块来增加网格点计数器
- 关节角度 → 指定目标姿势的关节角度
- 坐标 → 指定目标姿势的坐标

此外，用户还可修改以下可选设置。请注意，若未指定，将使用默认/全局值。

- 速度设置
 - 关节速度/加速度 → 指定点到点运动的速度
 - 线速度/加速度 → 指定直线运动/圆弧运动的速度
 - 角速度/加速度 → 指定仅改变 TCP 方向(位置保持不变)的运动速度
- 坐标系设置
 - 工具偏移 → 相对于法兰坐标系(机械臂顶部)移动TCP
 - 工具坐标系 → 相对于法兰坐标系(机械臂顶部)调整 TCP
 - 工件偏移 → 相对于基坐标系(机械臂底部)移动坐标参考系
 - 工件坐标系 → 相对于基坐标系(机械臂底部)调整坐标参考系
- 其他设置
 - 相对坐标 → 如果启用, 坐标值将加到当前坐标上
 - 阻塞运动 → 如果启用, 程序流程仅在运动完成后继续

4.7.7.3 路径块

“路径”模块会触发一个预先定义的路径。用户选择要运行的路径。运行该模块时, 机器人会先移动到路径的初始姿势, 然后以平滑的动作沿路径各段运行。

4.7.7.4 拾取模块

“拾取”模块向连接的工具发送拾取命令。对于角式或平行式夹具, 这意味着关闭夹具以将物体夹持在内部; 对于气动夹具, 这意味着启用真空泵以将物体固定在原位。该模块不带参数。

4.7.7.5 放置模块

放置模块向已连接的工具发送放置命令。对于角式或平行式夹具, 这意味着张开夹具以释放内部物体。对于气动夹具, 这意味着禁用真空泵以将物体放置到位。该模块不带参数。

4.7.7.6 容器模块

容器模块仅用于将多个模块群组。它本身不具备任何特殊功能。

4.7.7.7 下一个网格位姿模块

“下一个网格位姿”模块会增加特定网格的内部位姿计数器。需先选择目标网格。每次使用“移动”模块向网格移动时, 其目标均为该计数器当前指向的位姿。若要实际移动至网格的所有点, 可在循环结构中同时使用“移动”和“下一个网格位姿”模块, 如下所示:

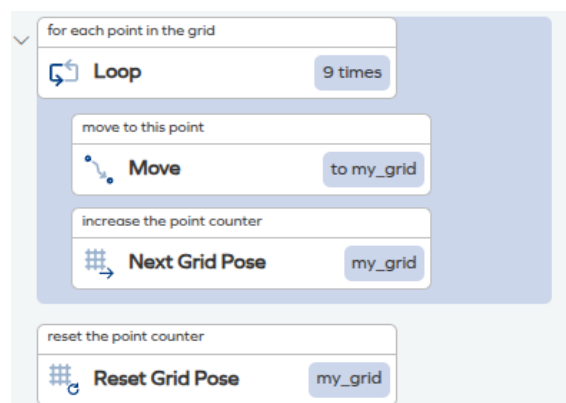


图 4.7.7.7: 网格模块使用示例

4.7.7.8 重置网格位姿块

“重置网格位姿”模块将特定网格的内部位姿计数器重置为零(初始值)。需要先选择目标网格。

4.7.7.9 设置变量模块

“设置变量”模块用于设置预定义应用程序变量的值。必须先选择一个变量, 然后选择数值。允许使用数值和/或逻辑表达式。

4.7.7.10 设置输出模块

“设置输出”模块用于设置数字输出或模拟输出的值。用户首先需要选择要写入的输出, 并根据输出类型(数字输出或模拟输出)分别设置布尔值或要设置的数值。

4.7.7.11 等待模块

“等待”模块会阻塞程序流程, 直到发生特定事件。有三种不同的等待模式可供选择:

- 时间 → 等待特定时间(允许使用数值表达式)
- 运动 → 等待正在运行中的运动命令停止
- 输入 → 等待数字输入的数值等于特定数值, 或模拟输入的数值超过某个阈值

4.7.7.12 循环模块

循环模块创建一组将被反复调用的模块。有三种不同的循环模式可供选择:

- 无限循环 → 永远循环
- 计数 → 循环特定次数(允许使用数值表达式)
- 条件 → 循环直至满足特定条件(允许使用逻辑表达式)

用户可在循环中添加“退出, 当满足条件时”模块, 详见下一个节。

4.7.7.13 “退出, 当满足条件时”模块

“退出, 当”块与其“循环”块绑定, 如果满足指定的条件(允许使用逻辑表达式), 则退出该循环。

4.7.7.14 条件模块

条件模块会创建一个包含“if”和“else”两个群组的容器。条件(允许使用逻辑表达式)用于将程序流程分支为“if”群组(如果条件成立)或“else”群组(如果条件不成立)。

4.7.7.15 注释模块

注释模块仅用于在程序流程中添加说明性注释。请注意, 其他每个模块也支持添加注释, 这些注释随后会在程序区域中显示。不过, 有时创建一个不与特定模块绑定的独立注释也是有意义的。

4.7.7.16 退出模块

退出模块将停止当前运行中的程序。剩余的编程模块将不会被执行。

4.7.7.17 对话框模块

“对话框”模块会创建一个包含多个群组的容器。执行时，程序流程将中断并弹出对话框。用户必须对对话框作出响应，程序流程才会继续。内部的群组(默认是“确定”和“取消”)与对话框中的按钮相对应。根据按下的按钮，程序流程将分支并继续执行所选的群组。对话框消息(允许变量注入)、按钮名称和按钮数量均可在块详情中进行修改。

4.7.7.18 打印模块

Print 模块将自定义消息打印到程序输出中。

4.7.7.19 日志模块

日志块将自定义消息记录到自定义日志文件中。日志文件可通过桌面下载，例如使用快捷键 CTRL + SHIFT + L。

4.7.7.20 通知模块

“通知”模块向图形用户界面(GUI)发送自定义通知消息。该通知不会阻塞程序流程。

4.7.7.21 警告模块

警告模块向 GUI 发送自定义警告消息。该警告不会阻塞程序流程。

4.7.7.22 错误模块

“错误”模块会向 GUI 发送自定义错误消息，并停止当前正在运行的程序。

4.7.7.23 Python 代码模块

Python 代码模块是一个高级模块，允许用户调用几行独立的 Python 代码。您可能需要使用此模块来调用那些通过块编程无法直接访问的函数。该模块内可使用完整的应用程序函数集。

4.8 I/O 菜单

点击主窗口上的“**I/O**”按钮可访问 I/O 菜单。该菜单显示所有数字和模拟 I/O，以及先前在 I/O 配置窗口中定义的属性(包括配置、分配的功能和自定义名称)，同时显示所有 I/O 的当前值。输入端口为只读，因此这些值仅供在此处直观查看。而输出端口则可通过此窗口进行设置。

数字 **I/O** 表示布尔值，其中 **True**、**1** 或“绿色”表示“通电”，**False**、**0** 或“灰色”表示“未通电”。数字输入在圆形指示器中显示该值，而数字输出可通过按下方形指示器按钮来更改其状态。

模拟 **I/O** 以 0.1 为步长显示为浮点数值。模拟 I/O 的单位由该 I/O 的配置定义(V 表示电压，mA 表示电流)。模拟输入以可观察的数值显示，而模拟输出可通过该输出当前值旁边的文本框进行修改。

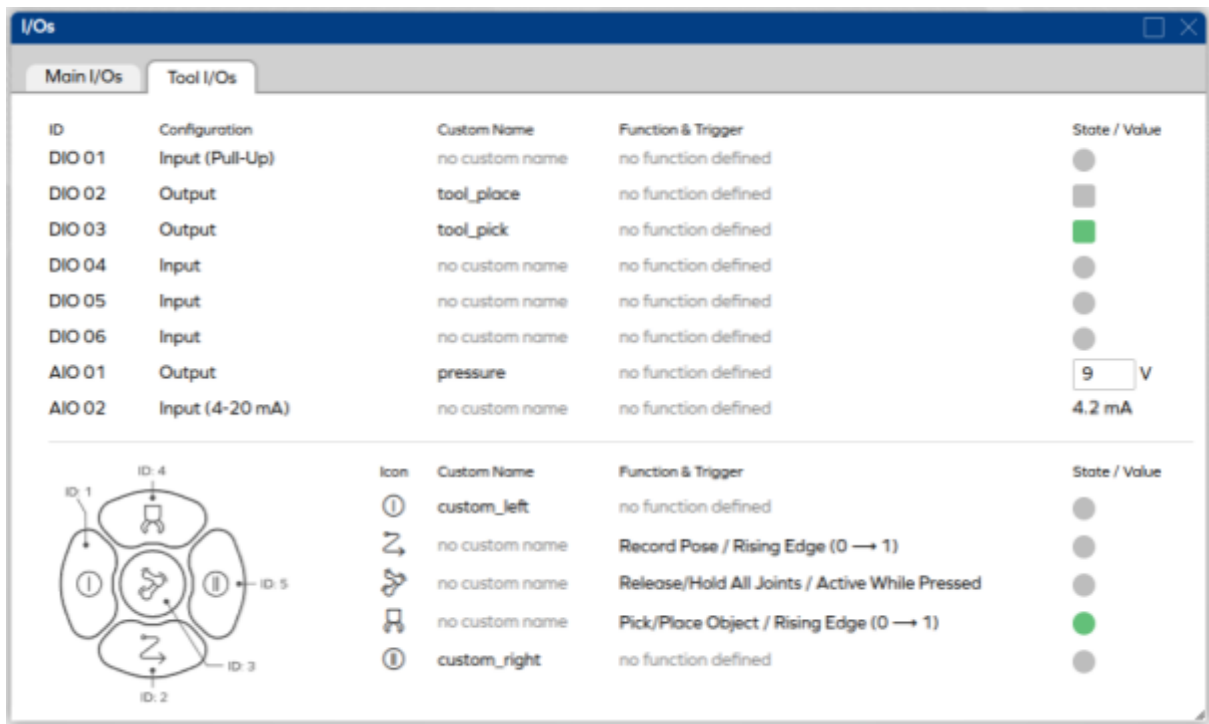


图 4.8:I/O 菜单

4.9 可视化模型菜单

点击主窗口上的“可视化模型”按钮可进入可视化模型窗口。该窗口仅显示机器人当前位姿，这在远程操作机器人且无法获得直接视觉反馈时尤为有用。您还可以在 3D 可视化模型中查看之前保存的位姿。可视化模型功能是交互式的，并支持以下操作：

- 在可视化模型窗口内单击鼠标（左键）并拖动，即可旋转图像，同时也可左右或上下拖动图像。
- 通过向上滚动鼠标滚轮进行放大，向下滚动则进行缩小。
- 使用键盘上的方向键可将摄像头左右及上下移动，或者右键单击鼠标并左右拖动。

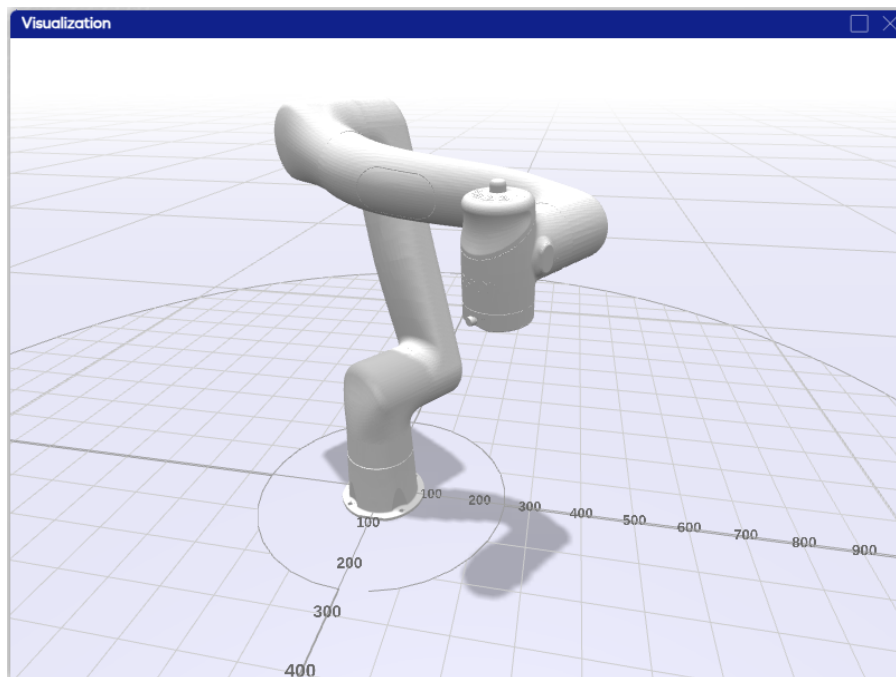


图 4.9: 可视化模型窗口

4.10 使能控制

点击主窗口中的“使能控制”按钮可打开使能控制菜单。该菜单允许用户指定在 HGC 模式下允许移动的机器人关节。默认情况下，按下机器人的 HGC 按钮（以及驾驶舱中的相关按钮）可让用户移动所有关节。通过点击此菜单中的各个关节指示器，用户可以启用或禁用特定关节，使其不受 HGC 影响。

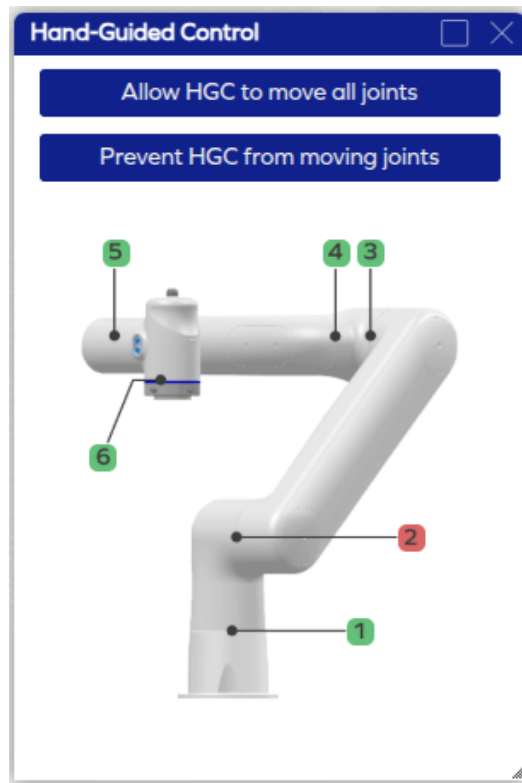


图 4.10:使能控制菜单

4.11 系统监视器

点击主窗口上的“系统监视器”按钮可进入系统监视器菜单。该菜单包含以下供用户观察的系统输出信息：

4.11.1 机械臂状态

“机械臂状态”选项卡显示机械臂的当前状态。其中包括相对于机器人基坐标系的工具中心点 (TCP) 坐标，以及所有电机的位置、电流和温度值。此外，所有安全输入的状态在此处均可查看。

4.11.2 程序输出

每当应用程序使用 `print(...)` 函数返回文本时，该文本都会显示在此窗口的“程序输出”选项卡中。请注意，如果多个应用程序并行运行，它们都可能向程序输出中打印文本。

4.11.3 程序日志

每当应用程序使用 `log_message(...)` 函数记录数据时，该数据会显示在此窗口的“程序日志”选项卡中。此处可对记录的数据进行筛选和搜索。若使用快捷键 **CTRL + SHIFT + L**，这些自定义日志将与内置日志文件一同下载。

4.11.4 历史消息

每当界面中弹出消息时，该消息也会列在此窗口的“历史消息”选项卡中。消息分为四种类型，适用于不同的场景：



信息消息向用户提供有用的提示、说明或建议。



成功消息提供反馈，告知用户某条命令或操作执行成功。



警告向用户提供重要信息，在继续操作前必须仔细阅读。警告从不属于关键性错误，也不会中断程序流程。



错误类型多种多样，从轻微的操作错误（例如错误的操作或功能参数）到由机器人损坏等原因引起的严重错误。

4.12 帮助菜单

“帮助”菜单包含有关当前运行中的系统和模块、用户可用的快捷键以及在出现问题或疑问时可使用的联系方式等有用的信息。

4.12.1 机器人信息

帮助菜单中的“机器人信息”选项卡包含有关 AX Portal 环境以及已连接硬件组件(如有连接)的详细信息。在机器人信息的“模块”部分,您将看到已安装模块的列表,以及它们是否已正确加载。只有当状态显示为“OK”时,您才能使用该模块提供的功能和菜单。如需寻求支持,请务必提供 AX Portal 及已连接硬件的详细机器人信息。您只需点击“复制到剪贴板”按钮,然后将其(最好使用等宽字体)粘贴在您的支持请求中即可。

4.12.2 快捷键

“快捷键”选项卡包含可用快捷键的概览。快捷键分为两类:常规键盘快捷键(例如用于操作控制台中的控制按钮)和应用程序编辑器快捷方式(用于编程 AX Portal 应用程序)。请注意,在 macOS 上,键盘布局与其他操作系统(Windows、Linux)有显著差异,因此快捷键列表也会有所不同。

4.12.3 代码文档

“代码文档”选项卡包含所有可用于应用程序编程的函数文档。该文档分为始终可用的核心函数,以及指向特定附加安装模块的扩展函数。通过选项卡标题栏中的“下载”按钮,可将代码文档下载为 ZIP 文件。若需离线查看代码文档,请在浏览器中打开其“index.html”文件。

4.12.4 开源许可证

“开源许可证”选项卡包含有关 AX Portal 所使用的开源库和模块的所有必要信息。

4.12.5 联系方式

“联系”选项卡显示了技术支持的联系方式。在提交支持请求前,请务必先查阅手册。提交支持请求时,请务必下载日志文件并随请求一并发送。

4.13 设置菜单

在“设置”菜单中，用户可以永久更改 AX Portal 的一些基本设置。点击“保存”以接受所有更改，点击“取消”以拒绝更改，或点击“重置”将所有设置恢复为最新保存的值。以下将对所有设置类别和选项进行说明：

The screenshot displays the 'Settings' window with the following sections and options:

- GUI Settings**
 - Live Auto-Completion: ☒
 - Show Optional Function Arguments: ☒
 - Message Fading Level: Warning (dropdown)
 - Message Fading Time: 5 (slider)
- Customize Cockpit Buttons**
 - Show 'Hand-Guided Control' button: ☒
 - Show 'Gripper' button: ☒
 - Show 'Jogging' button: ☒
 - Show 'Motion Controls' button: ☒
 - Show 'Homing' button: ☒
 - Homing Position
 - Joint 1: 0
 - Joint 2: 0
 - Joint 3: 0
 - Joint 4: 0
 - Joint 5: 0
 - Joint 6: 0
- Start-Up Settings**
 - Start-Up Project: Last Selected (dropdown)
 - Start-Up Application: Last Selected (dropdown)
 - Triggering Events: No Trigger (dropdown)
- Motion Behaviors**
 - Interrupt Behavior: Pause and Approve (dropdown)
 - Collision Resume Attempts: 3 (slider)
 - Collision Resume Timeout: 20 (slider)
 - Temperature Speed Reduction: ☒
- Authentication Settings**
 - Disable Authentication for ROS: ☒
 - Disable Authentication for TCP: ☒
 - Disable Authentication for REST: ☒
 - Disable Authentication for Modbus: ☒
- Default TCP Communication Settings**
 - Encoding Standard: UTF-8
 - End Character: \x03
 - Send Heartbeat: ☒
- ROS Function Settings**
 - Discovery Mode: Peer to Peer (dropdown)
 - IP Adresses of ROS 2 Peers: 127.0.0.1 (text input with add/remove buttons)
 - ROS Domain ID: 10 (dropdown)

Buttons at the bottom: Cancel, Reset, Save.

图 4.13: 设置菜单

4.13.1 界面设置

此类别包含若干与用户界面相关的设置。

实时自动补全	启用此选项可以在应用程序脚本编辑器中键入代码时实时完成函数的自动补全;如果禁用,您仍然可以使用快捷键 CTRL+SPACE 进行自动补全
显示可用函数参数	在代码中插入函数片段时,选择是否显示或者隐藏函数参数。为保持程序脚本简洁,建议禁用此选项。但如果您尚未熟悉内置函数,建议启用此选项。
消息显示级别	选择消息显示级别。若希望所有消息都消失,请尽可能选择高的级别;若希望所有消息都消失,请尽可能选择低的级别。
消息显示时间	指定消息的显示时间(01 - 60 秒)

4.13.2 自定义导航面板按钮

这些设置仅允许用户启用或禁用图形用户界面(GUI)中驾驶舱内显示的控制按钮,适用于桌面界面和面板界面。 这些按钮的详细功能在“控制台”章节中有说明。如果用户不想或无法使用某个按钮(例如,如果未连接交互式工具,则无法使用夹具按钮),或者用户不希望操作员通过面板界面使用该按钮,禁用该按钮会很方便。

显示 * 按钮	启用以显示此按钮,禁用以隐藏它。
回零位置	指定回零位置的角度,该位置与驾驶舱中可访问的“回零”按钮相关联。为每个关节角度输入一个值。

4.13.3 启动设置

在此类别中，所有设置均涉及 AX Portal 的启动流程。修改这些设置有助于避免启动时的重复操作。对于某些启动设置，请选择列表项“上一次选择”（默认选项），以记住并重新加载上次会话。

启动项目	选择启动时加载的项目。所有已定义的项目均会显示在此列表中。
启动程序	若已选择项目，请在此选择启动时加载的程序。该项目的所有已定义主程序均会显示在此列表中。
要触发的事件	选择此选项可让机器人启动时自动上电，甚至执行程序。默认情况下，此选项禁用。为防止用户意外执行非预期的程序，在选择触发器之前，必须先选定上述部分启动设置。

4.13.4 运动行为

在此类别中，所有设置均涉及机器人移动过程中的行为。

中断状态行为	选择中断时的运动行为。有关不同中断状态行为及如何处理中断的更多详情，请参阅以下子章节“中断状态行为”和“中断对话框”。
碰撞后的恢复尝试次数与超时	设置发生碰撞后尝试继续运行的最大次数。如果在设定的超时时间内发生过多次碰撞，机器人将停止继续运行。
温度速度降低	选择在驱动器温度过高时是否自动降低机器人速度。此设置默认启用，当温度超过 70 °C 时，运动速度会自动减慢以防止过热。若检测到温度超过 90 °C，将触发严重错误以避免因过热导致的损坏。有关此功能的更多详细信息，请参阅《六轴协作机器人 AX6：手册》中的“最大性能限制”章节。

4.13.4.1 中断状态行为

中断可能由多种原因引发，例如：

- 触发了 SS1 停止事件
- 触发了 SS2 停止事件
- 机器人与人员、物体或环境发生碰撞
- 安全防护中断被触发

当机器人关节的电流偏离其动态模型超过预定义阈值时，系统将检测到碰撞。碰撞检测是一项安全功能，旨在提供额外的保护层；切勿将其作为应用程序的组成部分有意使用，因为反复碰撞会增加机器人关节的负荷并加速磨损。

请注意，为确保碰撞检测正常工作，程序员在拾取或放置物体时，必须将机器人的负载调整为正确数值。

除了物理碰撞外，还可以将光栅等外部安全装置或类似的安全防护装置连接到机器人上。这些触发器在发生物理接触之前就发挥预防作用，但系统会以与碰撞事件相同的方式处理它们。“中断状态行为”定义了系统如何处理由碰撞和安全防护装置触发的中断。

目前，在中断状态行为处理中，建议的流程如下：

暂停&等待指令	会导致正在运行的程序暂停，并等待用户通过对话框批准继续或停止程序。
---------	-----------------------------------

4.13.4.2 中断对话框

当机器人正在运行过程中发生中断条件时，将显示中断对话框。典型的中断源包括检测到的碰撞、激活急停、安全防护装置触发或其他与安全相关的事件。该对话框将引导用户以安全且受控的方式解决该条件。

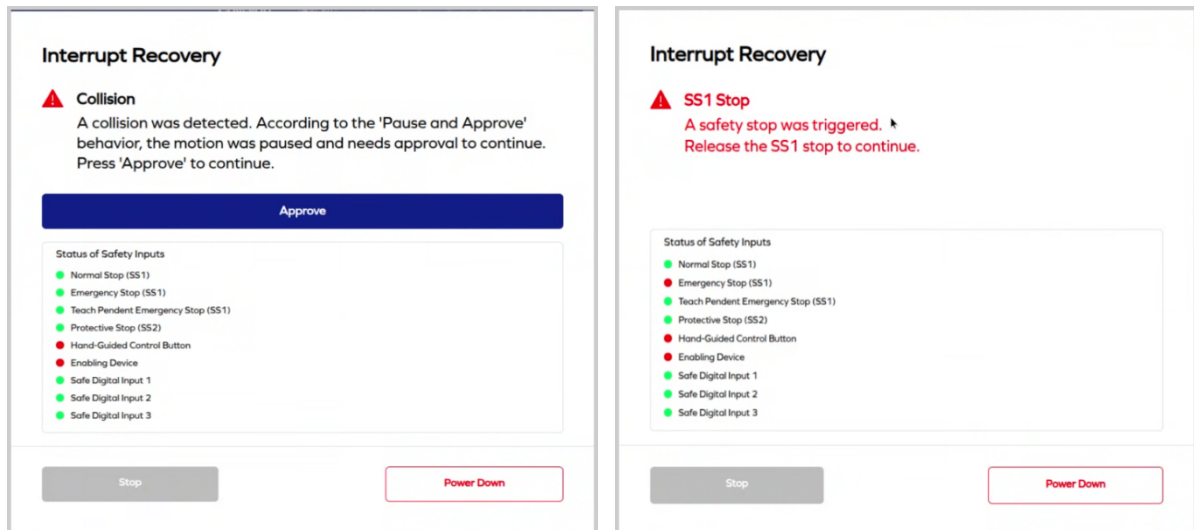


图 4.13.4.2: 中断恢复

该对话框分为两个主要部分：

上部区域——中断信息与处理：该区域描述当前检测到的中断条件，并提供明确的处理指导。示例包括：

- 碰撞后清理机器人路径。
- 释放急停或其他安全装置。
- 在必要时重置安全条件。

根据中断类型，会显示“批准”、“继续”或“重置”等操作按钮。用户必须按照说明操作并按下指定按钮才能继续运行。

下半部分 – 安全输入状态：对话框下半部分显示所有相关安全输入的实时状态。这使用户能够快速识别当前处于激活状态的安全装置，例如：

- 急停（包括示教器紧急停止）
- 保护性停止
- 使能按钮
- 手导控制按钮
- 安全数字输入

绿色指示灯表示正常状态，红色指示灯表示活动或阻塞条件（或按钮未被按下）。

停止选项

当用户需要在继续执行先前应用程序/运动或停止之间进行选择时，可使用“停止”按钮。按下该按钮将停止机器人并使其返回“就绪”状态，此时机器人已准备好执行新动作。

休眠选项

“休眠”按钮在“中断对话框”中始终可用。它允许随时对机器人进行休眠操作。请注意，休眠并不会自动解除中断条件；在重新上电继续运行之前，可能仍需清除根本原因（例如按下的急停按钮）。

典型工作流程

1. 发生中断时，对话框会自动显示。
2. 阅读上部区域中检测到的条件描述。
3. 检查下方的安全输入状态以确定当前的触发原因。
4. 通过物理操作解除该条件（例如：松开急停按钮、移除障碍物）。
5. 按下指定的操作按钮（“批准”、“继续”或“重置”）以继续。
6. 如有必要，请重复该操作，直到机器人继续运行。

SLP 违规

中断恢复的一个特例是 SLP（安全限位）违规，当机器人试图移动到安全设置中指定的位置限制之外时，就会发生这种情况。如果发生这种情况，中断对话框会引导用户切换到手动操作模式，在该模式下，用户可以手动移动到位置限制之外。除了手动移动外，当机器人处于 SLP 违规状态时，不允许进行任何移动。如果机器人已休眠，或者在启动时存在有效的 SLP 违规，请执行以下操作：

1. 切换到手动操作模式
2. 重启机器人
3. 进入精细控制菜单
4. 通过精细控制将机器人移出位置限制范围

4.14 安全设置菜单

“安全设置”菜单是进入 AX 系列机器人安全系统的网关。该界面专为安全主管设计。其他用户仅可查看和提取安全参数，但无法对其进行修改。

4.14.1 用户界面

进入安全设置菜单后，该菜单将展示机器人安全系统中当前设置的所有安全参数。所有用户均可查看这些设置，但只有安全主管可以修改它们。该菜单分为多个子菜单，您可以通过窗口底部的页码或前后按钮进行导航。

出口	点击“导出”按钮，安全设置将以 *.yaml 文件格式导出。该文件既可用于备份安全设置，也可导入到其他机器人中以复制安全配置。
编辑	安全主管可点击“编辑”按钮（并输入用户密码）进入编辑模式。

进入编辑模式后，所有子菜单的设置均可编辑。安全主管现在可以根据机器人的安装设置和使用场景更改安全限制。为了更好地了解哪些安全设置已被更改，已更改的字段会以蓝色标记。

进口	点击“导入”按钮，即可加载先前导出的安全设置文件。这将覆盖所有子菜单中手动修改的数值。
申请	点击“应用”按钮（并输入用户密码）后，所有安全设置将发送至安全系统，由其验证标志和限制条件。若验证失败，菜单中无效的条目将以红色显示，便于您快速识别。若验证成功，菜单将切换至验证模式。请注意，此时值已保存但尚未经过验证。必须验证设置后才能启动机器人。
取消	点击“取消”按钮，安全设置将不予应用，所有更改均被拒绝。随后菜单将切换回（默认）观察模式。

当安全员完成安全设置的编辑，且所有数值均已成功应用后，该菜单将进入验证模式。此前已应用但尚未通过验证的设置将以黄色标记。

验证	按下“验证页面”按钮，安全主管将正式确认当前可见子菜单中的设置已通过，并选择下一个子菜单。按下最后一个子菜单的“完成验证”按钮是保存设置的最后一步。
----	--

验证新数值后，必须对安全系统的行为进行验证。可通过在“安全验证模式”下上电来实现，该模式允许软件在验证过程中忽略所有安全限值，以测试安全系统是否按预期触发。

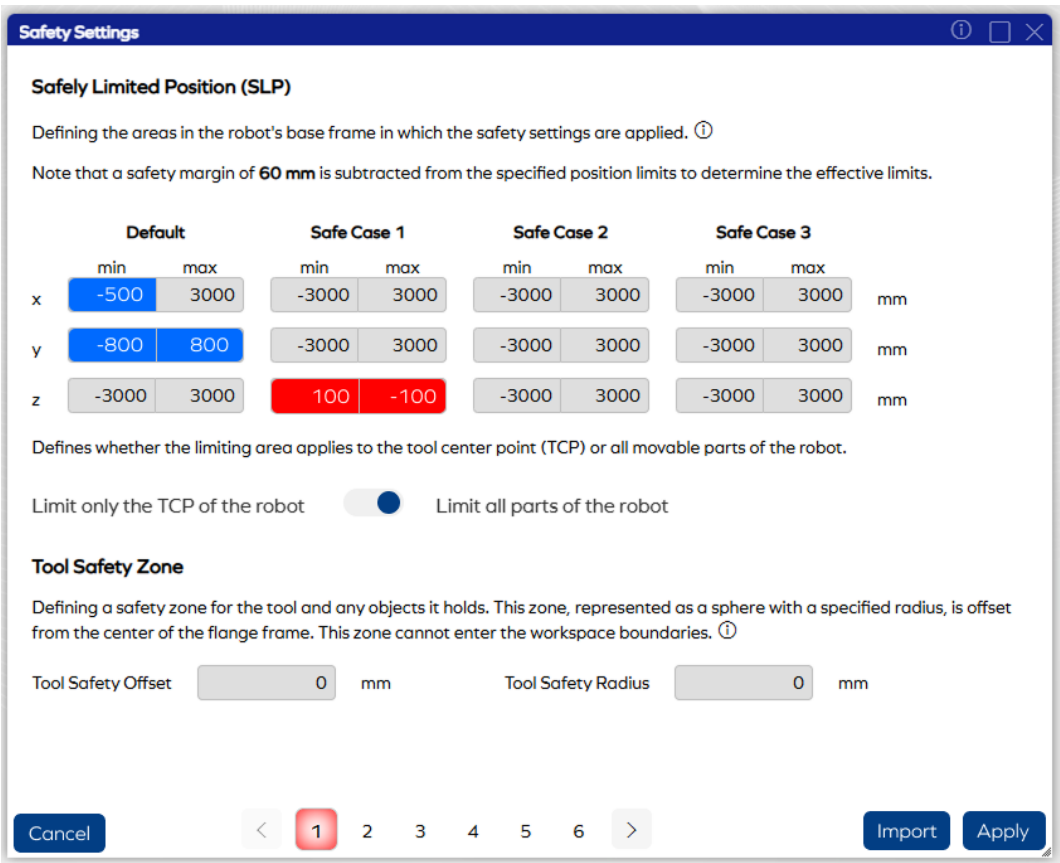


图 4.14.1:编辑设置时的“安全设置”菜单

4.14.2 安全设置

大多数安全设置(涵盖各子菜单)分为 4 列。这些列定义了机器人的安全工况。当相应的安全数字输入(SDI 1-3)被触发时,相应的安全工况即被触发。无论安全数字输入状态如何,默认参数始终处于激活状态。

安全数字输出(SDO)不可配置,其电气规格详见《机器人控制器 RC-A101 手册》。

对于某些设置,安全裕度是硬编码的。该数值指定了软件可在安全系统之前触发操作的可接受范围,以防止发生安全违规。

4.14.2.1 安全限位 (SLP)

安全设置菜单的第一个子菜单包含“安全限位位置”。在此,可以限制机器人在 x、y 和 z 方向上的位置,以机器人的基坐标系表示。如果坐标模式不明确,点击信息图标将显示带有坐标的图像。您可以限制机器人所有部件(包括底座和连杆)的位置,也可以仅限制机器人的工具中心点(TCP)位置。

Safety Settings

Safely Limited Position (SLP)

Defining the areas in the robot's base frame in which the safety settings are applied. ⓘ

Note that a safety margin of 60 mm is subtracted from the specified position limits to determine the effective limits.

	Default		Safe Case 1		Safe Case 2		Safe Case 3		
	min	max	min	max	min	max	min	max	
x	-3000	3000	-3000	3000	-3000	3000	-3000	3000	mm
y	-3000	3000	-3000	3000	-3000	3000	-3000	3000	mm
z	-3000	3000	-3000	3000	-3000	3000	-3000	3000	mm

Defines whether the limiting area applies to the tool center point (TCP) or all movable parts of the robot.

Limit only the TCP of the robot ☐ Limit all parts of the robot ☒

Tool Safety Zone

Defining a safety zone for the tool and any objects it holds. This zone, represented as a sphere with a specified radius, is offset from the center of the flange frame. This zone cannot enter the workspace boundaries. ⓘ

Tool Safety Offset mm Tool Safety Radius mm

Export < 1 2 3 4 5 6 > Edit Start Verification

图 4.14.2.1: SLP 设置示例

4.14.2.2 安全极限关节角度

安全设置菜单的第二个子菜单包含“安全极限关节角度”选项。默认情况下，“安全极限关节角度”处于禁用状态。启用安全限位后，即可监控关节的角位置。

然而，在安全模式下，关节角度的监控范围被限制在 $[-180^\circ, 180^\circ]$ 之间。因此，与机器人的实际物理极限相比，其工作范围可能会受到限制。为避免这种限制，可以分别禁用第一关节、第四关节和第六关节的监控功能。

“安全极限关节角度”功能仅在下限和上限均已定义时才生效。若两限均未定义，则该关节的监控功能将被禁用。

Safety Settings

Safely Limited Joint Angles

Defining the position limits of each joint of the robot. ⓘ

The default position limits are determined by the physical constraints of the joints and cannot be modified.

Note that some joints can move beyond **180 deg**, but safe cases do not support these ranges. Disabling the limits allows the full joint range to be used.

Note that a safety margin of **5 deg** is subtracted from the specified position limits to determine the effective limits.

	Default		Safe Case 1		Safe Case 2		Safe Case 3		
	min	max	min	max	min	max	min	max	
Joint 6	Disabled		Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	deg
Joint 5	Disabled		-160	160	-160	160	-160	160	deg
Joint 4	Disabled		Dis.	Dis.	-100	100	Dis.	Dis.	deg
Joint 3	Disabled		-165	165	-165	165	-165	165	deg
Joint 2	Disabled		-160	160	-160	160	-160	160	deg
Joint 1	Disabled		Dis.	Dis.	Dis.	Dis.	-179.9	179.9	deg

Export

< 1 2 3 4 5 6 >

Edit

Start Verification

图 4.14.2.2: 安全极限关节角度的示例设置

4.14.2.3 安全限速 (SLS)

安全设置菜单的第三个子菜单包含安全限速功能。在此，可以单独限制机器人每个关节的速度，以及机器人所有部件（包括工具、底座和连杆）的运动速度。

Safety Settings

Safely Limited Speed (SLS)

Defining the speed limits of each joint and all movable parts of the robot.

Note that a safety margin of **7 deg/s** or **100 mm/s** is subtracted from the specified speed limits to determine the effective limits.

	Default	Safe Case 1	Safe Case 2	Safe Case 3	
Joint Velocity 6	247	247	247	247	deg/s
Joint Velocity 5	247	247	247	247	deg/s
Joint Velocity 4	247	247	247	247	deg/s
Joint Velocity 3	187	187	187	187	deg/s
Joint Velocity 2	187	187	187	187	deg/s
Joint Velocity 1	187	187	187	187	deg/s
Speed	2000	2000	2000	2000	mm/s

Export

< 1 2 3 4 5 6 >

Edit

Start Verification

图 4.14.2.3:SLS 的示例设置

4.14.2.4 安全功率与力限制

安全设置菜单的第四个子菜单包含与功率和力量相关的设置：

- 对于每种安全工况，用户可定义不同的力限制值，该值决定了机器人对外部力（如与人员或物体的碰撞）的敏感程度。力限制值作用于工具中心点（TCP）。
- 设置平均负载（以及运行期间的适当负载值）对于提高力限制对特定应用的敏感度至关重要，该值定义为工具重量加上抓取物体的重量。
- 机器人的重力矢量必须与机器人的安装方式相匹配。如果机器人安装在地面上，重力矢量为 $[0, 0, -9.81]$ 。但如果机器人安装在墙壁、倾斜表面或天花板上，则必须先计算重力矢量，并在此处正确输入。

The screenshot shows the 'Safety Settings' window with the 'Safe Power and Force Limit (PFL)' section selected. The window has a blue header bar with the title 'Safety Settings' and standard window controls. The main content area is white with a blue border. The 'Safe Power and Force Limit (PFL)' section has a title bar and a descriptive paragraph. Below the text are four input fields for 'Force Limit' with values 250, 250, 250, and 250, and a unit 'N'. The 'Average Payload' section has a title bar and a descriptive paragraph, followed by an input field for 'Average Payload' with a value of 0 and a unit 'kg'. The 'Gravity Vector' section has a title bar and a descriptive paragraph, followed by three input fields for 'Gravity Vector' with values 0, 0, and -9.81, and a unit 'm/s²'. At the bottom of the window is a navigation bar with an 'Export' button, a set of numbered tabs (1, 2, 3, 4, 5, 6) with tab 4 selected, an 'Edit' button, and a 'Start Verification' button.

	Default	Safe Case 1	Safe Case 2	Safe Case 3	
Force Limit	250	250	250	250	N

Average Payload: 0 kg

Gravity Vector: x=0, y=0, z=-9.81 m/s²

图 4.14.2.4:安全功率和力限制设置

4.14.2.5 安全 I/O 和操作员限制

安全设置菜单的第五个子菜单用于配置与安全相关的I/O行为和操作员权限。用户可定义保护停止的处理方式、在手动模式下是否监测输入信号、复位条件、模式切换权限，以及操作变更所需的密码要求。

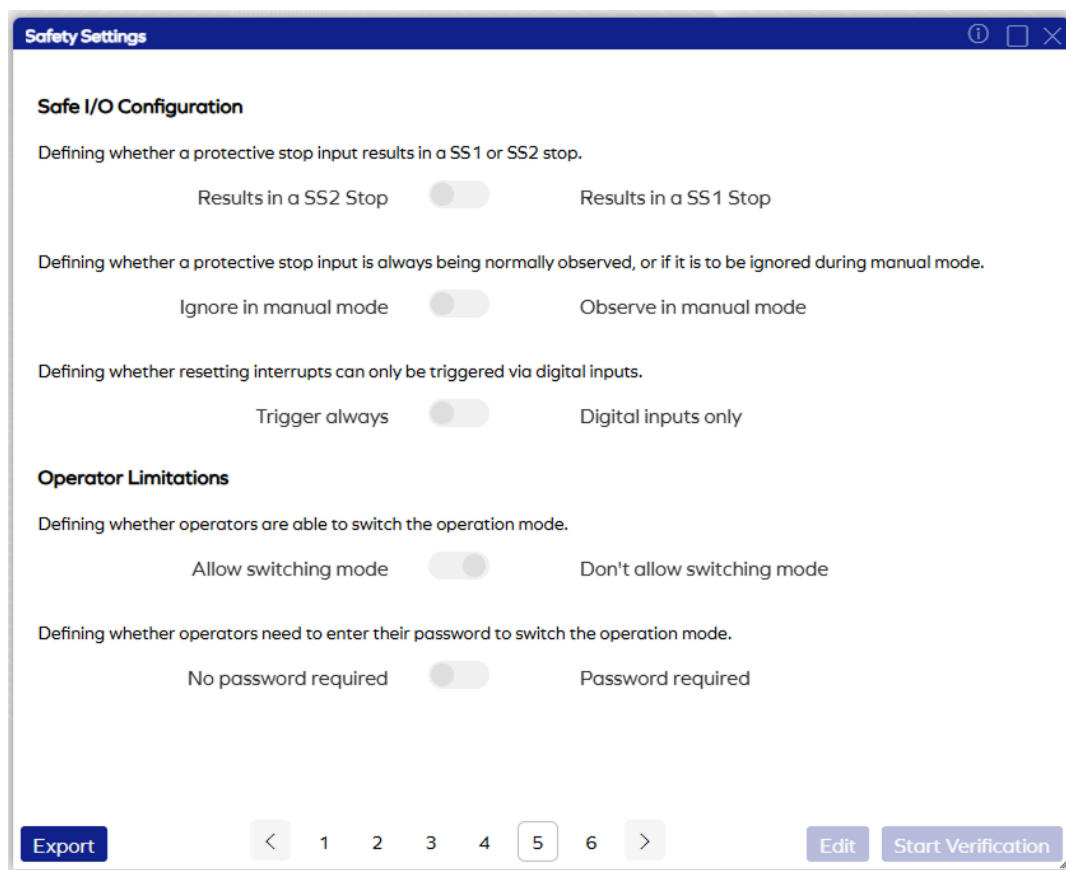


图 4.14.2.5:安全 I/Os 配置设置

4.14.2.6 验证数据

安全设置的最后一个子菜单包含有关安全设备的信息：

- 从安全设备读取的校验和(用于故障排除)
- 传输计数器(记录安全参数的写入次数)
- 安全参数的最后更新时间
- 安全参数是否已验证
- 安全设备的序列号和固件版本

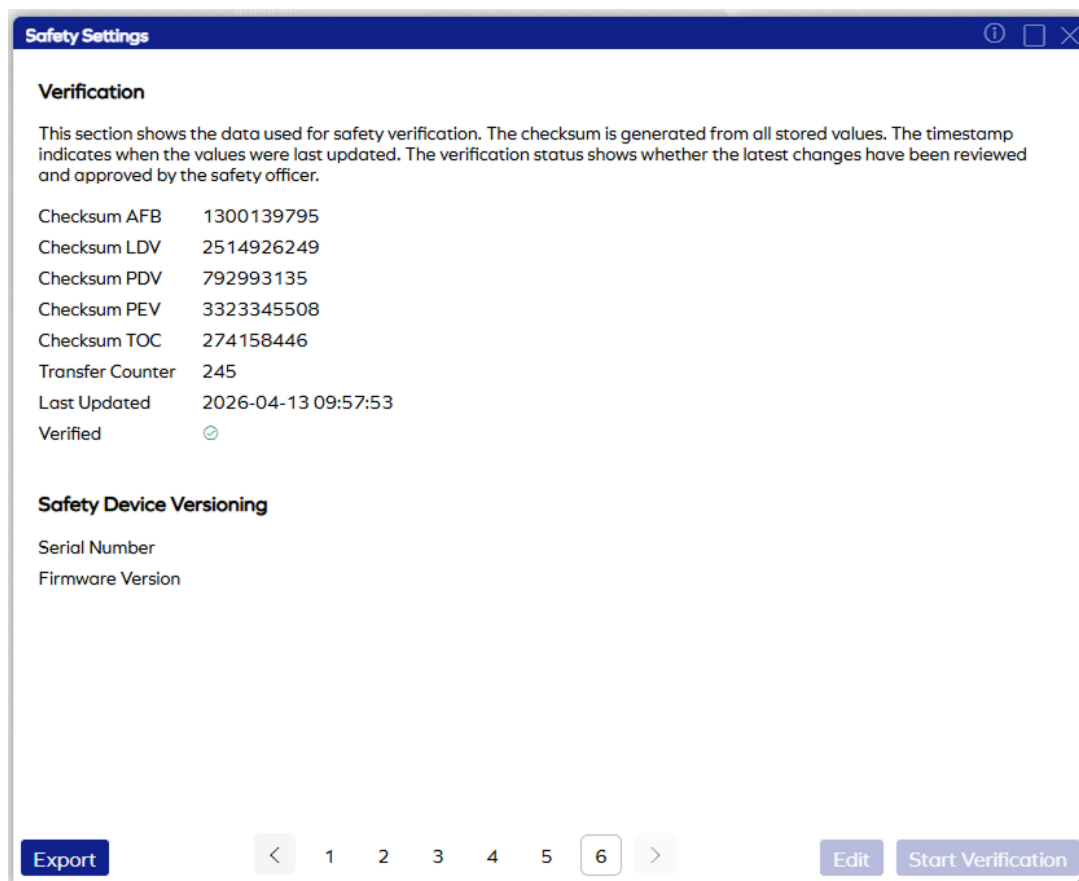


图 4.14.2.6: 验证设置

4.14.2.7 安全输出功能

特定安全功能的逻辑状态与机器人控制器的安全输出相连。有关详细信息，请参阅《机器人控制器 RC-A101 手册》。

4.14.2.8 连接可选设备

有关连接平板支架或使能按钮等可选设备的说明，请参阅《机器人控制器 RC-A101 手册》。

5. 面板界面 (GUI)

5.1 简介

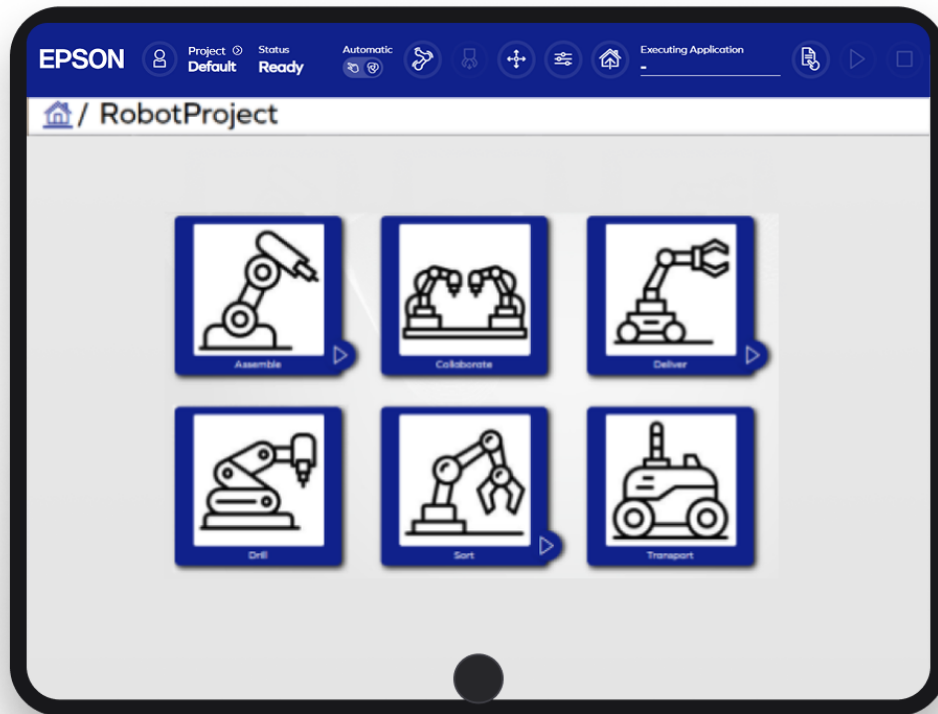


图 5.1.a: 面板界面的示例屏幕截图

面板界面是桌面界面的替代方案，可快速访问您的主要应用程序。该界面设计完全适应客户需求，特别是仅显示程序（及其状态信息）。面板界面支持触摸屏操作，其设计借鉴了现代移动设备中广为人知的应用程序式结构，即使对没有技术背景的用户而言也直观易用。此外，使用面板界面时，用户无需关注编程或应用程序背后的程序脚本。只需点击相应的应用程序图标，该应用程序即开始运行。可通过管理员面板登录后，点击“导航到控制面板”按钮访问此界面。

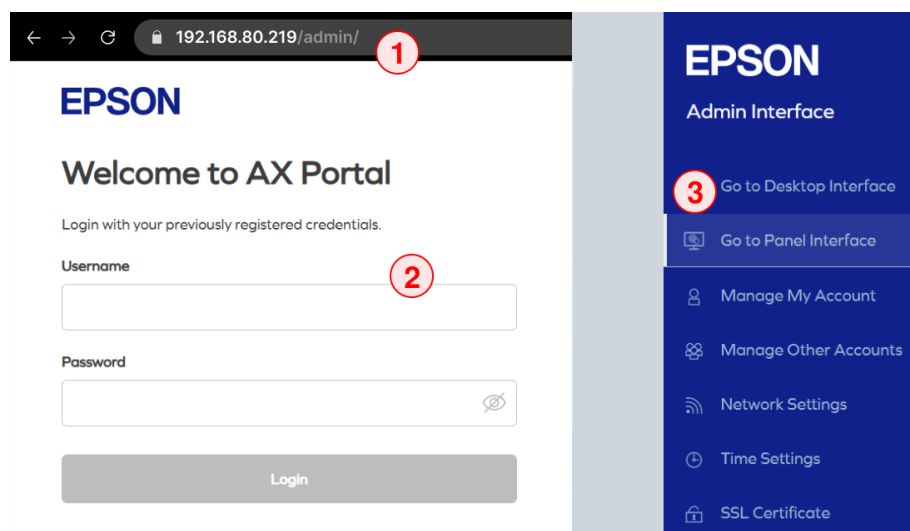


图 5.1.b: 访问面板界面

在某些浏览器中(尤其是在 Windows 系统上), 显示缩放比例默认设置为高数值(例如 150%), 这可能会导致界面部分内容(如控制面板或徽标) 显示不完整。为了获得最佳体验, 我们建议将显示缩放比例设置为 100%(甚至更低)。

5.2 配置面板界面的应用程序

默认情况下, 面板界面不会显示任何程序。用户首先需要指定要在该界面显示的程序, 以便操作人员快速执行。要配置面板程序, 请按照以下步骤操作:

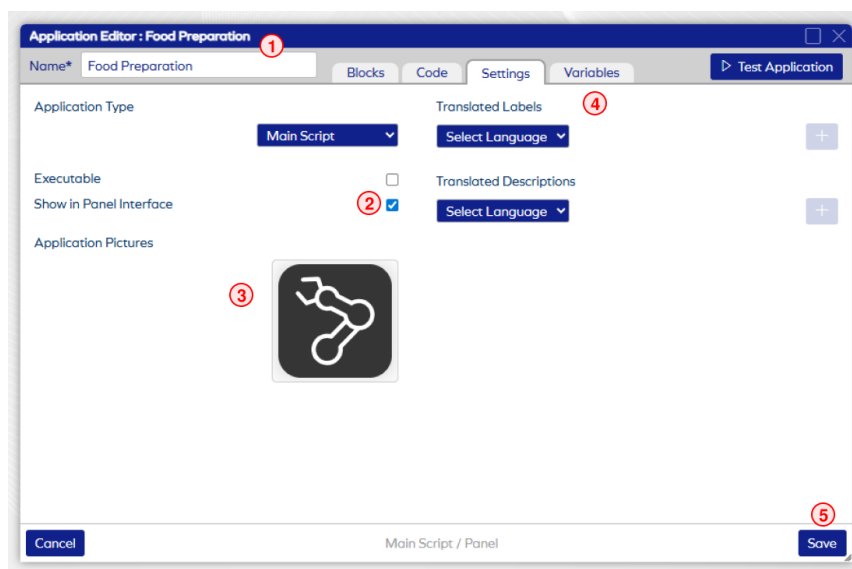


图 5.2.a: 程序编辑器中的面板界面相关设置(桌面界面)
与面板界面相关的设置: 2、3、4

1. 导航到桌面界面，编辑程序，进入“程序设置”，
2. 启用“在项目面板中显示”选项，
3. 可选上传合适的程序图片(最佳尺寸比例为 5:4)，该图片随后将与程序名称一同显示在面板界面的主菜单中，
4. 可选:导航至“应用程序变量”选项卡添加状态变量，这允许您通过面板界面监控和/或修改变量，
5. 最后保存程序。

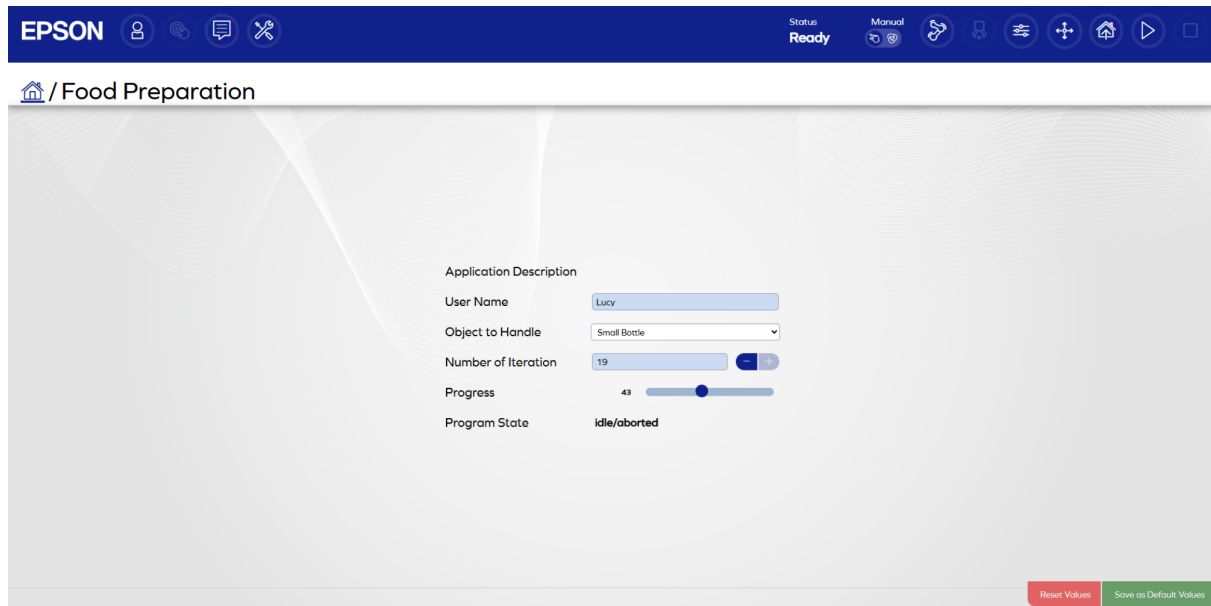


图 5.2.b:带状态变量的面板应用程序

5.3 控制台与设置

面板界面的控制台显示的按钮与桌面界面的控制台大致相同。有关所有元素的详细说明，请参阅“控制台”一节。除了这些元素外，您还会在左侧看到一些不同的按钮：



“面板接口”按钮将打开主面板页面，显示可供选择和运行的程序列表。



“消息”按钮将打开历史消息，其中所有过往消息（包括警告和错误）均按时间顺序显示。



“设置”按钮将引导您进入设置窗口，该窗口仅包含最关键的快速访问设置，即

- 选择已连接的工具、
- 启动和关闭机器人，
- 下载日志文件。

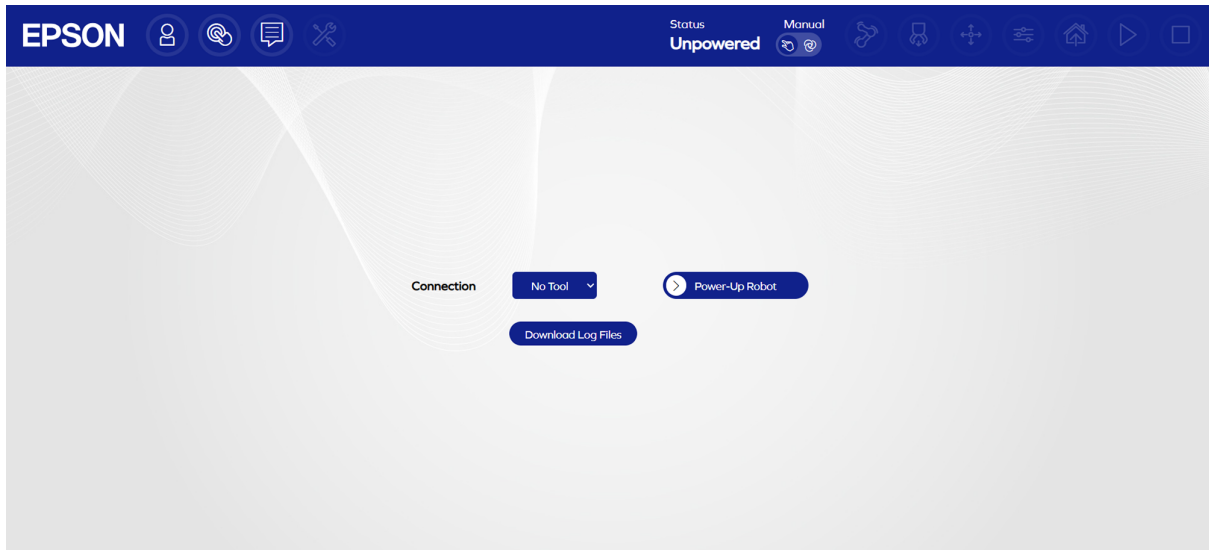


图 5.3: 面板接口设置窗口

5.4 程序控制

在面板界面的主窗口中，所有程序均以类似程序的结构列出，显示其名称和程序图片。面板将程序分为两类：

- 不包含状态变量的程序会显示一个播放按钮。点击程序按钮即可直接运行该程序。
- 包含状态变量的程序不显示播放按钮。点击程序按钮可进入程序详情视图，该视图显示所有状态变量。在此窗口中，您可以修改可编辑的状态变量，并点击播放按钮来执行该程序。点击返回按钮可返回主面板窗口。

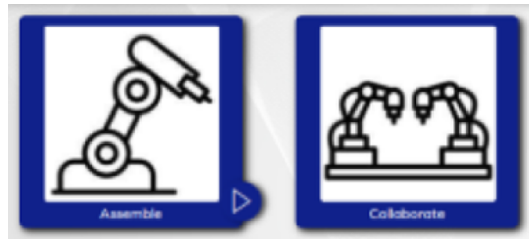


图 5.4:应用程序按钮

6. 运动学定义

机器人运动学描述了机器人的运动方式及其运动的表示方法。要开始使用运动学，用户必须了解可用的运动学参数和物体的不同定义：

6.1 坐标

坐标定义了机器人的绝对位置和方向。位置是一个包含三个数值 $[x, y, z]$ 的向量(单位为毫米)，方向也是一个包含三个数值 $[\text{roll}, \text{pitch}, \text{Z轴旋转}]$ (航海角)的向量(单位为度)。

初始化可按 ([600, 0, 300, 180, -90, 0]) 方式进行：

```
coord = Coordinates([600, 0, 300, 180, -90, 0])
# ... this is similar to ...
coord = Coordinates({"x": 600, "y": 0, "z": 300, "roll": 180, "pitch": -90, "yaw": 0})
```

访问元素可以如下进行：

```
coord = Coordinates([600, 0, 300, 180, -90, 0])

# accessing individual elements (x, y, z, roll, pitch, yaw)
x = coord.x
x = coord["x"]
x = coord[0]

# accessing position or orientation
position = coord.position
orientation = coord.orientation
```

6.1.1 航海角

航海角是定义坐标系方向分量(X轴旋转、Y轴旋转、Z轴旋转)的多种方法之一。航海角常用于飞机的方向表示。使用航海角时，三维旋转定义为

- 1st 绕负Z轴在Z轴旋转 (ψ) 方向上的旋转，
- 2nd 绕负Y轴在俯仰 (θ) 方向上的旋转，以及
- 3 绕正X轴在横滚 (ϕ) 方向上的旋转。

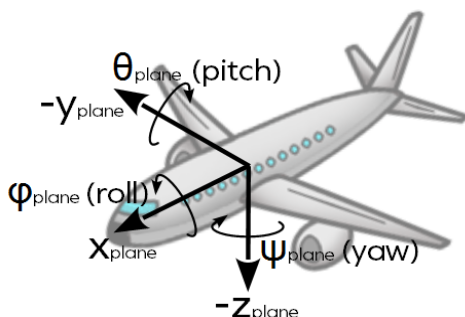


图 6.1.1.a: 航海坐标系

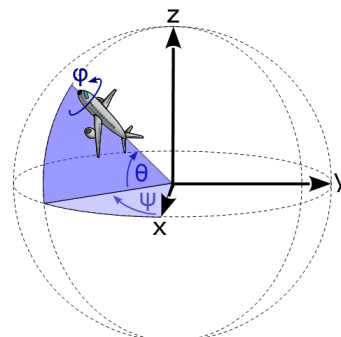


图 6.1.1.b: 航海旋转顺序

提示: 每当一个函数要求你处理末端执行器的三维方向时，想象一个平面附着在末端执行器上会很有帮助，这样末端执行器和该平面的坐标系就重合了。这样，就更容易提取出导致这种特定方向的角度(X轴旋转、Y轴旋转和Z轴旋转)。

6.1.2 坐标系

对于 AX 系列机器人的操作, 以下坐标系对用户至关重要:

- 机器人的基坐标系定义为: 其 z 轴指向远离底板的方向, x 轴与底板平行并指向远离底板面板接口连接器的方向。随后, 众所周知的右手定则定义了 y 轴的方向。
- 机器人的工件坐标系是相对于基坐标系的用户定义坐标系, 除非另有说明, 否则其与基坐标系一致。坐标值将以此坐标系表示。用户可利用该坐标系为机器人定义自定义工作空间。
- 机器人的法兰坐标系因连接的工具类型而异:
 - 若无任何工具安装, 该坐标系的原点位于机器人顶部法兰(连接板)的中心。当所有关节角度均为零时, 其方向与基坐标系的方向一致。
 - 如果安装了自定义工具(并通过工具编辑器定义), 则该坐标系将根据该工具指定的位置和方向进行偏移。
- 机器人的“工具坐标系”是相对于法兰坐标系的用户定义坐标系, 除非另有说明, 否则其与法兰坐标系一致。坐标值将以该坐标系的原点为基准。用户可利用该坐标系定义工具中心点。

关于这些坐标系的示例, 请参见下图:

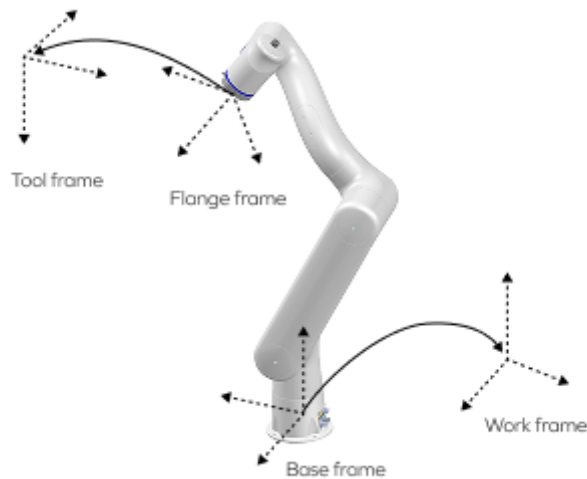


图 6.1.2: 机器人坐标模式

6.2 配置

对于给定的机器人位置和方向，反向运动学计算可能产生多种不同的关节配置。这些配置在关节的弯曲或旋转方式上存在差异。配置设置用于从这些可能的选项中选择一种特定的关节配置。对于典型的六轴机器人，即使末端执行器的位置和方向相同，肩部、肘部和腕部的不同方向组合也可能产生多达八种不同的配置。为区分这些配置，它们被命名为“c1”至“c8”。

请注意，配置不适用于比较两个位姿或检查路径是否连续。配置更改并不一定导致路径不连续。同样，不连续的路径也不一定包含配置更改。

6.3 位姿

位姿定义了机器人的坐标和配置。它可以通过数据库中预定义位姿的ID或名称来定义。或者，可以通过指定坐标和配置并使用“创建位姿”函数，或通过指定关节角度并使用正向运动学函数来创建。

初始化操作如下：

```
# read predefined pose from database by index
pose = Pose(1)
# read predefined pose from database by name
pose = Pose("fancy_pose")
```

6.4 点

点是路径中的一个离散元素。除了机器人的位姿外，它还指定了机器人的速度以及自上一个点以来经过的时间间隔。这被称为 PVT(位置、速度、时间)，通常保存为关节角度向量、关节速度向量和时间间隔浮点数。请注意，点的扩展版本可能还会详细说明机器人的加速度。

6.5 路径

路径定义了机器人如何在两个或多个位姿之间移动。连续路径仅在工具坐标系中通过多个线性或圆弧运动的路径段定义。随后，该路径被离散化并转换为关节坐标系。如果路径的离散解不需要机器人重新定位(即不进行配置更改)，则该路径被称为无缝路径；如果沿路径的加速度是连续的，则该路径被称为连续路径。

路径始于一个指定的位姿(而非坐标)。路径的各部分通常定义坐标，因为路径的配置由初始位姿决定。通过使用逆运动学函数对每个前一离散点的最近解，路径将在可能的情况下自动成为无缝路径。请注意，尽管从技术上讲路径并不包含速度信息，但我们仍将“路径”和“轨迹”这两个词互换使用。

初始化可按以下方式进行：

```
# read predefined path from database by name
path = Path("fancy_path")
```

6.6 路径段

路径段代表路径中的一个独立部分，用于定义 TCP 如何向目标移动。

可用的分段类型包括：

- 线性:TCP 进行直线运动, 到达该路径段定义的目标坐标。
- 圆弧运动:TCP 从当前位姿出发, 沿圆弧路径经过定义的中间位置, 最终到达该路径段的目标坐标。请注意, 此运动过程中的 TCP 方向由算法自动确定。
- 坐标:TCP 通过最优路径移动至目标坐标。这对应于关节空间中的直线运动。
- 位姿:TCP 通过最优路径移动至目标姿势。这对应于关节空间中的直线运动。
- 方向:TCP在保持当前位置的同时, 执行纯旋转运动以达到指定的目标方向。

对于每个段, 可以定义速度和加速度参数:

- 对于“方向”段, 这些参数分别对应角速度(单位:[deg/s])和角加速度(单位:[deg/s²])
- 对于所有其他段类型, 它们分别对应线速度(单位:[mm/s])和线性加速度(单位:[mm/s²])

此外, 还可以指定一个混合半径, 该参数定义了过渡期间, 当前段起始位姿相对于前一节段结束位姿的最大允许偏差(单位为毫米)。混合半径为 0 毫米时, 会强制 TCP 精确通过定义的位姿;而较大的值则允许路径在节段之间弯曲, 从而实现更平滑、更自然的过渡。

6.7 网格

网格是可自定义的、最多为三维的点数组, 用于定义预定义托盘的位置等, 并可用于打包和存储一组坐标值。通常, 网格被解释为二维数组, 但它也可以是一个不依赖第二维度的简单点列表(例如从传送带上拾取物体), 甚至可以是三维结构(例如将物体堆叠在箱子中)。

7. 连接模块

除了用于控制和监控机器人的用户友好且直观的图形用户界面 (GUI) 外, AX Portal 还提供了一套连接模块, 支持通过 TCP、REST、ROS 和 Modbus 等多种通信方式对机器人进行控制和监控。每种通信方式都适用于不同的应用场景:

- **TCP** 非常适合需要快速、持续数据交换的实时通信场景。它通常用于直接控制应用、自定义客户端实现, 或在与需要持久套接字连接的系统集成时。
- **REST** 最适合与基于网站的服务或程序集成。当易用性、可扩展性以及基于标准 HTTP 系统的兼容性比实时性能更为重要时, 通常会使用它。
- **ROS** (机器人操作系统) 专为机器人研发设计。当将机器人集成到复杂的机器人系统中时, 它特别有用, 可实现传感器融合、自主行为以及与其他支持 ROS 的设备交互等功能。
- **Modbus** 常用于工业自动化环境。它非常适合与 PLC (可编程逻辑控制器) 及其他工业设备进行通信, 在这些场景中, 可靠性和标准化至关重要。

本章将详细说明这些模块的使用方法。

7.1 连接、身份验证与活动控制

在使用任何连接模块之前, 您必须了解这些模块如何连接到客户端、身份验证如何运作, 以及活动控制在 AX Portal 中的工作原理。

7.1.1 连接通道

设备与机器人之间的连接始终通过特定的连接通道进行。不同连接模块对这些连接通道的实现方式略有差异。

TCP 模块支持多通道。这意味着每个想要连接到机器人的设备都会创建自己的连接通道, 并单独登录。关闭其中一个通道将断开连接, 并释放活动控制 (见下文)。

 警告	REST、ROS 和 Modbus 模块仅提供单一通道。这意味着一旦与这些模块中的任意一个建立连接并完成身份验证, 任何设备均可向机器人发送命令。这些通道在 AX Portal 启动时创建, 且无法由已连接的设备关闭。
---	---

7.1.2 身份验证

AX Portal 采用安全的两步验证机制。首先, 请按照“安全认证”章节中描述的远程登陆流程获取有效的访问令牌。此步骤必须始终通过独立的 **TCP** 连接完成。成功登陆并创建访问令牌后, 您可以通过选定的连接通道注册该令牌, 并使用 **token_login** 命令通过该令牌进行用户认证。

 注意	若在设置菜单中禁用了通过 TCP 的身份验证, 则可跳过此身份验证步骤。 请注意, Modbus 模块不支持发送 token_login 命令, 因此必须在“设置”菜单中禁用该模块的身份验证功能, 才能使用该模块。
---	---

7.1.3 活动控制站

AX Portal 具有活动控制站保护功能，其他通道无法同时控制机器人。如果没有此保护，可能会导致危险情况。

连接通道使用 **request_active_control** 命令成为活动控制站。一旦该命令执行成功，该控制站将拥有永久控制权，直至通道关闭或机器人断电。如果另一个控制站正在控制，该命令将阻塞，直到该控制站接受控制请求。

当前活动控制站使用 **release_active_control** 命令来释放当前控制权。建议在通道不再被积极使用时始终释放控制权。否则，该通道将永久锁定当前控制权。

7.2 TCP连接模块

7.2.1 建立连接

TCP连接模块的服务器端基于 Python 实现，并使用了内置的 **socket** 库。默认情况下，可通过端口 **38152** 访问服务器。消息采用 **UTF-8** 编码标准进行编码。消息之间以文本结束字符 **0x03** (Python 中为 **"\x03"**) 分隔。请使用以下示例脚本连接到您的机器人并打印所有接收到的响应。关于接收消息的含义，将在下文的“接收命令”部分中进一步说明。

```
import socket
import time

# define the robot ip address
ip_address = "10.0.0.203"

# create the tcp socket
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client_socket:
    # connect to the server
    client_socket.connect((ip_address, 38152))
    # prepare message buffer
    message_buffer = ""
    while True:
        # read data and add it to message buffer
        try:
            message_buffer += client_socket.recv(1024).decode("UTF-8")
        except socket.timeout:
            time.sleep(0.1)
            continue
        # check for complete messages in buffer
        if "\x03" in message_buffer:
            # extract complete messages
            message_split = message_buffer.split("\x03")
            for message in message_split[:-1]:
                print(message)
            # incomplete message (if any) remains in buffer
            message_buffer = message_split[-1]
```

7.2.2 发送命令

在向机器人发送命令之前，请确保您已理解“连接、身份验证和活动控制”一节中阐述的原则。以下示例发送初始的 **token_login** 和 **request_active_control** 命令。

```
import json
import socket
import time

# define the robot ip address, access token, and session identifier
ip_address = "10.0.0.203"
token = "MY_ACCESS_TOKEN"
session_id = "MY_SESSION_ID"

# create the tcp socket
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client_socket:
    # connect to the server
    client_socket.connect((ip_address, 38152))

    # use the authentication token to log in (needed if authentication is not disabled)
    request = {"action": "token_login", "bridge": "core",
               "arguments": {"token": token, "session_id": session_id}}
    message = json.dumps(request) + "\x03"
    client_socket.sendall(message.encode("UTF-8"))

    time.sleep(0.5)

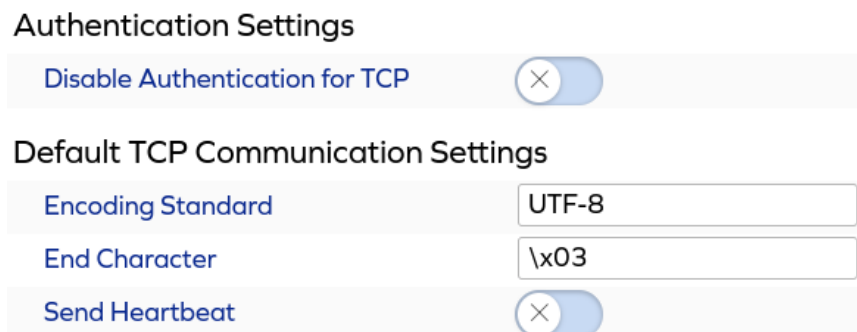
    # make sure to acquire active control, which makes the TCP connection the active
    # control station of the robot (only one control station can be active at any time)
    request = {"action": "request_active_control", "bridge": "core",
               "arguments": {}}
    message = json.dumps(request) + "\x03"
    client_socket.sendall(message.encode("UTF-8"))

    time.sleep(0.5)

# add more robot control commands here
```

7.2.3 设置

上述示例使用了以下默认 TCP 设置, 这些设置可在桌面界面的“设置”菜单中进行修改。在此处, 您可以永久关闭 TCP 认证, 并更改 TCP 消息的编码和结束字符。此外, 您还可以启用心跳功能, 以持续检查连接是否仍然有效。



The image shows a settings window for the TCP module. It is divided into two main sections. The first section, 'Authentication Settings', contains a single toggle switch labeled 'Disable Authentication for TCP', which is currently turned on (indicated by a blue circle with an 'X'). The second section, 'Default TCP Communication Settings', contains three items: 'Encoding Standard' with a text input field showing 'UTF-8', 'End Character' with a text input field showing '\x03', and 'Send Heartbeat' with a toggle switch that is also turned on (blue circle with an 'X').

图 7.2.3:TCP 模块设置

7.2.3.1 编码标准

在此处可定义通过 TCP 连接发送的消息编码。支持常见的 Python 编码标准。常用的标准包括默认的 UTF-8 或 ASCII。

7.2.3.2 结束字符

在此处, 您可以更改通过 TCP 连接发送的消息的结束字符。结束字符标志着消息的结束。当接收消息时, 系统会自动从消息中截取结束字符。

7.2.3.3 发送心跳包

在此您可以选择是否让 TCP 服务器自动发送心跳包。心跳将每 50 毫秒发送一次, 内容为一条空消息 (仅发送结束字符)。

7.2.4 消息协议 - 接收命令

所有发送和接收的命令均为 JSON 字符串。如上所述，消息采用 **UTF-8** 编码标准，并以文本结束字符 **0x03** (Python 中为 “\x03”) 分隔。接收到的命令是一个 JSON 编码的字典，包含一个键和一个值。键定义了消息的类型。值包含消息的具体信息。您可以在“建立连接”章节中找到一个从机器人接收这些消息的程序脚本。

请注意，此列表并非完整列表，但包含了最重要的命令。

7.2.4.1 状态信息

机器人的状态会在发生变化时立即发送。

键	数值	类型
status_info	根据以下列表显示的当前机器人状态： ● 0 → 未上电 ● 1 → 上电中 ● 2 → 加载中 ● 3 → 就绪 ● 4 → 处理中 ● 5 → 校准中 ● 6 → 运行中 ● 7 → 自主模式 ● 8 → 暂停 ● 9 → 继续 ● 10 → 保护停止 ● 11 → 安全防护 ● 12 → 位置限制 ● 13 → 使能控制 ● 14 → 碰撞 ● 15 → 急停 ● 16 → 正常停止 ● 17 → 错误 ● 18 → 断电中 ● 19 → 停止中	整数

```
{"status_info": 0}
```

7.2.4.2 执行返回值

执行操作后，将发送包含该操作命令的返回值。如果操作没有返回值，则发送 `null`。只有在函数执行过程中未发生错误时，才视为成功。最后，序列号是一个可选参数，仅在用户发送命令时指定了序列号时才会添加。

键	数值	类型
<code>action_return_value</code>	返回操作的名称、返回值以及操作是否成功。如果发生错误，则操作不成功。	字典

```
{"action_return_value": {"action": "release", "value": null,
                        "sequence": 100, "success": true}}
```

7.2.4.3 错误

如果发生错误，将发送一条包含更详细信息的错误消息。

键	数值	类型
错误	错误信息。	字典

```
{"error": {"text": "The argument 'variable_value' is missing for the
              'set_state_variable' action.",
          "title": "Minor Error",
          "color": 3,
          "code": "e0010035",
          "time": 1554907874.0436678}}
```

7.2.4.4 历史消息

发送到用户界面的历史消息列表。

键	数值	类型
<code>message_history</code>	消息包含文本、标题、程序脚本和时间戳。	列表

```
{"message_history": [{"text": "The specified bridge does not exist.",
                        "title": "Minor Error", "color": 3,
                        "code": "e0010019",
                        "time": 1554897027.5446029}]}
```

7.2.4.5 状态群组定义

定义机器人的位置数据。状态群组包含一个 ID，用于指定所包含数据的类型。这些定义可用于识别状态群组数据。

键	数值	类型
---	----	----

state_group_define	状态群组定义。	字典
--------------------	---------	----

```
{"state_group_define": {"id": "pose",
  "label": "tcp_pose",
  "members": [{"id": "x", "label": "x"},
    {"id": "y", "label": "y"},
    {"id": "z", "label": "z"},
    {"id": "roll", "label": "roll"},
    {"id": "pitch", "label": "pitch"},
    {"id": "yaw", "label": "yaw"}],
  "cols": [{"id": "value", "unit": "mm, deg"}]}}
```

7.2.4.6 状态群组数据

状态群组数据包含 TCP 或电机数据。使用“request_data”接收此数据。

键	数值	类型
state_group_data	该数值包含机器人的 TCP 数据(id = “pose”)、电机数据(id = “robot”)或安全标志(id = “safety”)。	词典

```
{"state_group_data": {"id": "pose",
  "data": {"x": {"value": "0.0"},
    "y": {"value": "0.0"},
    "z": {"value": "1378.5"},
    "roll": {"value": "0.0"},
    "pitch": {"value": "90.0"},
    "yaw": {"value": "180.0"}}}}
```

7.2.4.7 打开对话框

当用户界面中的对话框打开时，系统会接收到“open_dialog”命令。随后可通过“answer_dialog”执行操作对该对话框进行回复。

键	数值	类型
open_dialog	该对话框接受以下值： ● id → 唯一的对话框 ID (用于响应该对话框) ● kind ○ 0 → 确认对话框 (包含确认/拒绝按钮) ○ 1 → 输入对话框 (文本输入) ○ 2 → 文件对话框 (文件上传) ○ 3 → 下拉对话框 (需要选项列表) ○ 4 → 选择对话框 (需要选项列表) ● 默认 → 若未收到回答时返回的值 ● 颜色 ○ 0 → 信息 ○ 1 → 成功 ○ 2 → 警告 ○ 3 → 错误 ● title → 要显示的标题 ● text → 显示文本 ● options → 下拉/选择对话框的选项列表	字典

```
{"open_dialog": {"id": "5cee85a8-9ab2-4de8-b1ea-52ded8cc68c0",  
  "kind": 0, "default": true, "color": 2,  
  "title": "Movement Warning",  
  "text": "In order to Calibrate, the robot will move.  
  Do you want to proceed?"}}
```

7.2.4.8 脚本发送

接收由脚本发送的消息。

键	数值	类型
script_send	脚本发送的数值	多种

```
{"script_send": 123}
```

7.2.5 消息协议 – 发送命令

所有发送和接收的命令均为 JSON 字符串。如上所述，消息采用 **UTF-8** 编码标准，并以文本结束字符 0x03 (Python 中为“\x03”) 分隔。发送的命令必须包含一个 JSON 编码的字典，其中包含三个必填键（以及一个可选键）。您可以在“发送命令”部分找到向机器人发送消息的程序脚本。

执行	定义要执行的命令	str
bridge	定义包含该命令的模块，通常为“core”	str
参数	定义命令参数，请参阅下文关于参数的详细描述	字典
序列	<可选> 最终在 <code>action_result_value</code> 中返回的命令标识符	int

请注意，此列表并非完整列表，但包含了最重要的命令。

7.2.5.1 令牌登陆

为了使用活动连接，必须首先发送一个登陆调用。请按照“安全认证”部分中描述的远程登陆程序获取有效的访问令牌。

执行	token_login	
Bridge	核心	
参数键	token	session_id
参数数值	由登陆命令提供的个性化登陆令牌	由登陆命令提供的唯一会话 ID
参数类型	字符串	字符串

```
{
  "action": "token_login",
  "bridge": "core",
  "arguments": {
    "token": "MY_ACCESS_TOKEN",
    "session_id": "MY_SESSION_ID"
  }
}
```

7.2.5.2 请求活动控制

获取活动控制权，使 TCP 连接成为机器人的活动控制站。请注意，这会向活动控制站发送一个请求，该请求可能被接受或拒绝。

执行	request_active_control
桥接	核心

```
{
  "action": "request_active_control",
  "bridge": "core",
  "arguments": {}
}
```

7.2.5.3 释放活动控制

释放活动控制，使得该 TCP 连接不再是机器人的活动控制站。

执行	release_active_control
Bridge	核心

```
{"action": "release_active_control",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.4 请求数据

此命令可作为心跳检测使用。接收不同类型的数据。

执行	request_data
Bridge	核心

```
{"action": "request_data",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.5 上电

让机器人启动其电机。

执行	power_up
Bridge	核心
参数键	工具ID
参数数值	自定义工具的 ID, 或 0 以自动检测工具
参数类型	整数或字符串

```
{"action": "power_up",  
  "bridge": "core",  
  "arguments": {"tool_id": 0}}
```

7.2.5.6 休眠

终止连接，使机器人关闭其电机。

执行	power_down
Bridge	核心

```
{"action": "power_down",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.7 暂停

暂停所有正在运行的移动函数和脚本。

执行	暂停
Bridge	核心

```
{"action": "pause",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.8 继续运行

如果机器人处于暂停状态, 则继续运行。

执行	resume
Bridge	核心

```
{"action": "resume",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.9 释放使能控制模式 (HGC)

释放机器人关节，使其进入重力补偿模式并可手动移动。状态需为“就绪”，并将切换至使能控制模式。

执行	release
Bridge	核心
参数键	joint_ids
参数数值	<可选> 要启用手控的关节 ID 列表。默认情况下，所有关节均启用手控。
参数类型	列表

```
{"action": "release",  
  "bridge": "core",  
  "arguments": {"joint_ids": ["1", "2", "5", "6"]}}
```

7.2.5.10 保持使能控制 (HGC)

将机器人关节保持在位置控制模式下，使其无法再通过手动移动。状态需为“使能控制”，并将切换为“就绪”。

执行	hold
Bridge	核心
参数键	joint_ids
参数数值	<可选> 要保留的关节 ID 列表。默认保留所有关节。
参数类型	列表

```
{"action": "hold",  
  "bridge": "core",  
  "arguments": {"joint_ids": ["1", "2", "5", "6"]}}
```

7.2.5.11 停止

中断所有正在运行的运动函数和脚本。

执行	停止
Bridge	核心

```
{"action": "stop",  
  "bridge": "core",  
  "arguments": {}}
```

7.2.5.12 切换项目

将当前数据库项目更改为具有特定 ID 的项目。该项目的状态必须为“就绪”。

执行	change_project
Bridge	核心
参数键	project_id
参数数值	要切换到的项目的 ID。默认项目的 ID 为 1。
参数类型	整数

```
{"action": "change_project",  
  "bridge": "core",  
  "arguments": {"project_id": 1}}
```

7.2.5.13 执行脚本

根据名称或 ID 运行应用程序。状态需为“就绪”，并将切换为“运行中”。脚本运行期间，机器人可以暂停。脚本完成后，状态将切换回“就绪”。

执行	execute_script	
Bridge	核心	
参数键	script_name	script_id
参数数值	要运行的脚本名称。	要运行的脚本的 ID。
参数类型	字符串	整数

```
{
  "action": "execute_script",
  "bridge": "core",
  "arguments": {
    "script_id": 3
  }
}
```

```
{
  "action": "execute_script",
  "bridge": "core",
  "arguments": {
    "script_name": "my_application"
  }
}
```

7.2.5.14 测试脚本

运行应用程序的 Python 脚本。状态需为“就绪”，随后将切换为“运行中”。脚本运行期间，机器人可以暂停。脚本完成后，状态将切换回“就绪”。

执行	test_script		
Bridge	核心		
参数键	脚本代码	script_type	script_id
参数数值	待测试应用程序的 Python 代码。	脚本的类型： “main” “parallel” “background”	待测试脚本的 ID。如果该脚本在数据库中不存在，则 ID 为 0。
参数类型	字符串	字符串	整数

```
{
  "action": "test_script",
  "bridge": "core",
  "arguments": {
    "script_code": "print('Hello World')",
    "script_type": "main",
    "script_id": 0
  }
}
```

7.2.5.15 脚本接收

使用此功能可从连接通道向应用程序脚本发送消息。该消息可通过脚本的 **receive_message** 函数接收。

执行	script_receive
桥接	核心
参数键	消息
参数数值	脚本将接收的消息。
参数类型	多种

```
{"action": "script_receive",  
  "bridge": "core",  
  "arguments": {"message": 123}}
```

7.2.5.16 运动位姿

将机器人移动到具有特定 ID 或特定名称的位姿。状态需为“就绪”，并将切换为“运行中”。机器人移动期间，其运动可以暂停。一旦移动完成，状态将切换回“就绪”。

执行	move_pose
Bridge	核心
参数键	pose_id
参数数值	要移动到的位姿的 ID 或名称。
参数类型	整数或字符串

```
{"action": "move_pose",  
  "bridge": "core",  
  "arguments": {"pose_id": 2}}
```

```
{"action": "move_pose",  
  "bridge": "core",  
  "arguments": {"pose_id": "start_pose"}}
```

7.2.5.17 执行路径

执行具有特定 ID 或特定名称的路径。状态需为“就绪”，并将切换为“运行中”。机器人移动期间，其运动可以暂停。一旦移动完成，状态将切换回“就绪”。

执行	播放路径
桥	核心
参数键	path_id
参数数值	要执行的路径的 ID 或名称。
参数类型	整数或字符串

```
{"action": "play_path",  
  "bridge": "core",  
  "arguments": {"path_id": 2}}
```

```
{"action": "play_path",  
  "bridge": "core",  
  "arguments": {"path_id": "p1"}}
```

7.2.5.18 切换工具

切换工具功能，通常指拾取和放置操作。状态需为“就绪”，并将切换为“运行中”。一旦功能完成，状态将切换回“就绪”。

执行	toggle_tool
Bridge	核心
参数键	选择
参数数值	true 表示拾取物体，false 表示放置物体。这由拾取行为定义。
参数类型	布尔值

```
{"action": "toggle_tool",  
  "bridge": "core",  
  "arguments": {"pick": false}}
```

7.2.5.19 写入数字与模拟输出

写入主控制器或工具控制器(从主控制器传入的信号线)的一个或多个输出(数字或模拟)。对于每种组合,使用不同的函数(参数相同)。所有 I/O 的列表可在 I/O 配置菜单或 I/O 菜单中找到。

执行	write_digital_main_output write_analog_main_output write_digital_tool_output write_analog_tool_io	
桥接	核心	
参数键	标识符	值
参数数值	识别单个 I/O 的方法有很多,可以通过 ● 数字(整数) ● 客户名(字符串) 如果同时写入多个 I/O,您可以选择将编号或客户名组合在列表中。	根据您的选择的“标识符”类型,数值可以是 ● true/false(布尔值)用于数字 I/O ● 数字(浮点数)用于模拟 I/O 或者,如果同时写入多个 I/O,则为这些数值的列表。
参数类型	整数 / 字符串 / 列表	布尔值 / 浮点数 / 列表

```
{
  "action": "write_digital_tool_io",
  "bridge": "core",
  "arguments": {
    "identifiers": 3,
    "values": true
  }
}
```

此示例写入数字工具输出 3 并将其设置为 true。

```
{
  "action": "write_analog_tool_io",
  "bridge": "core",
  "arguments": {
    "identifiers": "my_analog_output",
    "values": 9.5
  }
}
```

此示例写入名为“my_analog_output”的模拟工具输出,并将其设置为 9.5V。

```
{
  "action": "write_digital_main_io",
  "bridge": "core",
  "arguments": {
    "identifiers": [1, 2],
    "values": [true, false]
  }
}
```

此示例写入数字主输出 1 和 2,并将 1 设置为 true,2 设置为 false。

7.2.5.20 对话框响应

响应一个已打开的对话框窗口。对话框的响应取决于对话框的类型。

执行	answer_dialog	
Bridge	核心	
参数键	对话框ID	dialog_answer
参数数值	要发送答案的对话框的唯一标识符。该 ID 是传入的 open_dialog 响应的一部分。	要发送给该对话框的回答。
参数类型	字符串	布尔值 / 整数型 / 字符串

```
{
  "action": "answer_dialog",
  "bridge": "core",
  "arguments": {
    "dialog_id": "5cee85a8-9ab2-4de8-b1ea-52ded8cc68c0",
    "dialog_answer": true
  }
}
```

7.2.5.21 设置状态变量

将状态变量“variable_name”的值设置为“variable_value”。该变量必须已在ID为“script_id”的脚本中存在，且该脚本需已保存至数据库。此命令可在脚本运行中执行。

执行	set_state_variable		
Bridge	核心		
参数键	script_id	variable_name	variable_value
参数数值	包含待修改状态变量的脚本ID。	要修改的状态变量的名称。	新值。
参数类型	整数	字符串	各种

```
{
  "action": "set_state_variable",
  "bridge": "core",
  "arguments": {
    "script_id": 2,
    "variable_name": "integer_1",
    "variable_value": 42
  }
}
```

7.2.5.22 设置全局变量

将全局变量“name”的值设置为“value”。全局变量在脚本运行结束后仍会被保存, 即使机器人重启后也会保留。

执行	set_global_variable	
Bridge	核心	
参数键	姓名	值
参数数值	全局变量的名称	全局变量的数值
参数类型	字符串	多种

```
{"action": "set_global_variable",  
  "bridge": "core",  
  "arguments": {"name": "my_variable", "value": 3}}
```

7.2.5.23 获取全局变量

获取全局变量“name”的值。全局变量在脚本运行结束后仍会被保存, 即使机器人重启后也会保留。

执行	get_global_variable	
Bridge	核心	
参数键	姓名	
参数数值	全局变量的名称	
参数类型	字符串	

```
{"action": "get_global_variable",  
  "bridge": "core",  
  "arguments": {"name": "my_variable"}}
```

7.2.5.24 设置共享变量

将共享变量“name”的值设置为“value”。共享变量仅在脚本运行期间可用。脚本执行完毕或机器人停止后，这些变量将被删除。

执行	set_shared_variable	
Bridge	核心	
参数键	姓名	值
参数数值	共享变量的名称	共享变量的数值
参数类型	字符串	各种

```
{
  "action": "set_shared_variable",
  "bridge": "core",
  "arguments": {
    "name": "my_variable",
    "value": 3
  }
}
```

7.2.5.25 获取共享变量

获取共享变量“name”的值。共享变量仅在脚本运行期间可用。脚本执行完毕或机器人停止后，这些变量将被删除。

执行	get_shared_variable	
Bridge	核心	
参数键	姓名	
参数数值	共享变量的名称	
参数类型	字符串	

```
{
  "action": "get_shared_variable",
  "bridge": "core",
  "arguments": {
    "name": "my_variable"
  }
}
```

7.3 REST连接模块

7.3.1 建立连接

REST连接模块提供资源 **https://<IP地址>/rest/**网关，用于接收来自连接客户端的命令。请注意，仅有一个连接通道。如果多个用户通过 REST 连接，他们将共享此通道。

请使用以下示例脚本通过 REST 连接控制机器人，然后开始向机器人发送命令。请注意，在发送命令之前，您需要已登录，并且拥有活动控制权。因此，建议在不再使用该连接后释放控制权并注销。

```
import json
import requests

IP_ADDRESS = "10.0.0.203"
TOKEN = "MY_ACCESS_TOKEN"
SESSION_ID = "MY_SESSION_ID"

REST_URL = f"https://{IP_ADDRESS}/rest/gateway"
HEADERS = {"Accept": "application/json", "Content-Type": "application/json"}

def request(method, url, headers, data, timeout=None):
    try:
        response = method(url=url, headers=headers, data=data, timeout=timeout,
                           verify=False)
        response_code = response.status_code

        # this is equal to 'response.status_code < 400' (errors have codes 400+)
        if response.ok:
            # try to parse json payload
            try:
                response_payload = response.json()
            except json.decoder.JSONDecodeError:
                # usually this happens because the response payload is empty
                # this happens for example when a request returns "", 200
                response_payload = dict()
                print("response payload json was empty")
            else:
                print("response.ok returned False")

        except requests.Timeout:
            response_payload = dict()
            response_code = 408 # timeout
        except Exception as e:
            # usually connection problems...
            print(f"rest client could not connect to specified endpoint: {e}")
            response_payload = dict()
            response_code = 404 # not found

    return response_code, response_payload

def post_request(data, timeout=None):
    return request(method=requests.post, url=REST_URL, headers=HEADERS, data=data,
                  timeout=timeout)
```

```
# login
data = {"bridge": "core", "action": "token_login", "arguments": {"token": "<TOKEN>",
"session_id": "<SESSION_ID>"}}
response = post_request(data, timeout=10.0)
# acquire the active control before start sending commands to the robot
data = {"bridge": "core", "action": "request_active_control", "arguments": {}}
response = post_request(data, timeout=1.0)

# add your commands here

# when you're done using this connection, give away the active control
data = {"bridge": "core", "action": "release_active_control", "arguments": {}}
response = post_request(data, timeout=1.0)
# when you're done working with this connection, you should logout
data = {"bridge": "core", "action": "logout", "arguments": {}}
response = post_request(data, timeout=1.0)
```

7.3.2 发送命令

除了上文示例中提到的命令外，还有许多其他命令可用。请参考 TCP连接模块中列出的命令列表，因为这些命令也适用于 REST连接。

除了发送“bridge”、“执行”、“arguments”和可选的“sequence”参数外，您还可以发送可选的“超时”参数。若发送超时参数，REST 请求成功发送后不会返回默认响应 (200, {}), 而是该命令将阻塞，您将收到响应 (200, action_return_value), 其中 Action Return Value 包含以下信息：调用的 API 函数、该函数是否成功，以及该函数的实际返回值。

7.3.3 设置

在桌面界面的“设置”菜单中，您可以找到一个用于关闭REST连接模块认证的设置。关闭认证后，将无需登陆和注销。

Authentication Settings

Disable Authentication for REST



图 7.3.3:REST 模块设置

7.4 ROS 处理程序模块

ROS 处理程序模块允许您将机器人集成到 ROS 2 环境(机器人操作系统)中, 并使用 ROS 2 命令与 AX Portal 进行通信。本章中的说明兼容 ROS Jazzy 和 CycloneDDS 作为 ROS 中间件(RMW)。

 警告	ROS 处理程序提供了对某些低级功能的访问权限。例如, 您可以向机器人的电机发送直接控制命令, 这些命令将被立即执行。在图形用户界面(GUI)中, 某些功能可能需要额外的检查或对话框确认, 而这些功能在此处可能不可用。因此, 当您通过 ROS 直接控制机器人时, 请务必格外谨慎。
---	--

注:

- 这是首个支持 ROS 2 的 AX Portal 版本, 未来可能会推出更多功能。
- 目前, ROS 处理程序尚不支持 AX Portal 模拟器, 仅适用于实际机器人。

7.4.1 安装

要与 AX Portal 的 ROS 处理程序进行通信, 您需要在计算机上安装 ROS。官方 ROS 文档提供了多种安装选项, 例如 deb 包、Docker 镜像、源代码安装等。在本示例中, 我们将使用官方 ROS Docker 镜像与机器人进行通信。以下 Docker Compose 文件可在您的计算机上创建一个简单的 ROS 环境:

```
services:
  ros2_development:
    image: osrf/ros:jazzy-desktop
    container_name: "ros2_development"
    command: /bin/bash -c
      "apt update &&
      apt install -y ros-jazzy-rmw-cyclonedds-cpp &&
      sleep infinity"
    environment:
      - RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
      - CYCLONEDDS_URI=/ros/cyclonedds_config.xml

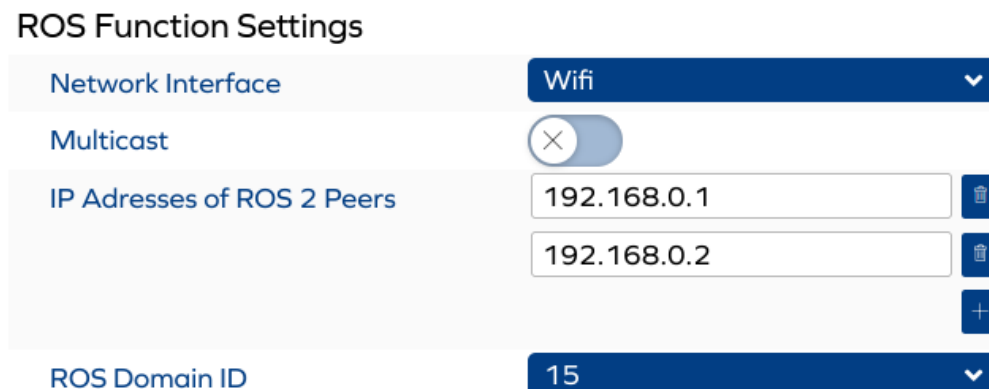
    network_mode: "host"

volumes:
  - "./cyclonedds_config.xml:/ros/cyclonedds_config.xml"
```

请注意, 我们在启动时安装了 `ros-jazzy-rmw-cyclonedds-cpp`, 并将 `RMW_IMPLEMENTATION` 设置为 `rmw_cyclonedds_cpp`, 以告知 ROS 使用 CycloneDDS——这与机器人上使用的 RMW 相同。此外, 我们共享一个 `cyclonedds_config.xml` 文件, 并将 `CYCLONEDDS_URI` 设置为其位置, 以实现自定义的 CycloneDDS 配置。

7.4.2 配置

机器人与计算机之间的 ROS 通信需要在两台机器上进行相应调整。在您的笔记本电脑上，您需要设置一个新的环境变量并提供网络配置文件。在机器人上，您无法直接访问这些元素，但可以通过设置菜单中的 ROS 部分在用户界面中进行编辑：



The screenshot displays the 'ROS Function Settings' window. It contains several configuration options: 'Network Interface' is a dropdown menu currently showing 'Wifi'; 'Multicast' is a toggle switch that is turned off; 'IP Addresses of ROS 2 Peers' is a list with two entries, '192.168.0.1' and '192.168.0.2', each with a delete icon to its right and a plus icon at the bottom right of the list; 'ROS Domain ID' is a dropdown menu currently showing '15'.

图 7.4.2:ROS 处理程序设置

在接下来的章节中，您将了解这些设置各自的含义，并学习在计算机端需要进行哪些相应的调整。

7.4.3 使用 Cyclone DDS 进行网络发现

我们之前多次提到过 Cyclone DDS。它究竟是什么？作为 AX Portal 用户，您需要了解以下内容：

当选择 Cyclone DDS 作为 ROS 中间件时，它将负责通过您的网络分发 ROS 相关的数据包。它负责在您的设备之间建立网络连接，以便 ROS 2 能够通过这些连接发送数据。

7.4.3.1 ROS 网络接口

首先，Cyclone DDS 需要知道 ROS 将使用哪个 ROS 网络接口。默认设置为以太网。

请在“ROS 功能设置”下的“ROS 网络接口”选项中选择所需的接口。

注意：当 AX Portal 启动 ROS 处理程序时，Cyclone DDS 会扫描 ROS 网络接口以查找对等机器的 IP 地址。如果无法访问该接口，ROS 处理程序将无法加载。因此，如果您将 ROS 网络接口设置为以太网，请在启动机器人之前确保网线已插入。

7.4.3.2 发现方法

Cyclone DDS 提供两种发现网络中其他设备的方法：多播和点对点。对于多播，它会将每个网络数据包发送给网络范围内所有可达的客户端。其优点在于简单：您无需费心定义哪些计算机被允许相互通信。在此设置中，ROS_DOMAIN_ID 环境变量（见下文）仍然可用，以防止两台机器人混淆消息。

其缺点在于网络开销:所有数据包都将被每个网络客户端处理。即使设置不同的 ROS_DOMAIN_ID 也无法避免这一点。即使某台计算机未安装 ROS:由于 Cyclone DDS 基于 UDP, 网络中的所有计算机仍会接收并处理这些 UDP 数据包。对于较小的网络和少量数据而言, 这通常不是问题。 但如果一个机器人在大型网络中向每个客户端发送摄像头流, 这会显著降低整个网络的速度。在这种情况下, 更安全的方法是在 Cyclone DDS 配置中明确定义一组 IP 地址, 允许它们通过点对点连接相互通信。

在接下来的两个小节中, 您将分别学习如何配置网络以支持组播或点对点通信。提示:在大多数 Linux 发行版中, ip addr show 命令可用于查询您的 IP 地址及对应的 ROS 网络接口名称。

7.4.3.2.1 点对点

在上述安装说明中, 我们在 Docker Compose 文件中挂载了一个 Cyclone DDS 配置文件。以下是一个 Cyclone DDS 配置示例, 用于将您的机器配置为支持点对点通信:

```
<?xml version='1.0' encoding='utf-8'?>
<CycloneDDS xmlns="https://cdds.io/config"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://cdds.io/config
https://raw.githubusercontent.com/eclipse-cyclonedds/cyclonedds/master/etc/cyclonedds.xsd"
">
  <Domain id="any">
    <General>
      <Interfaces>
        <NetworkInterface name="INTERFACE_NAME" multicast="false" />
      </Interfaces>
      <AllowMulticast>false</AllowMulticast>
      <EnableMulticastLoopback>false</EnableMulticastLoopback>
    </General>
    <Discovery>
      <MaxAutoParticipantIndex>119</MaxAutoParticipantIndex>
      <ParticipantIndex>auto</ParticipantIndex>
      <Peers>
        <Peer address="192.168.1.1" />
        <Peer address="192.168.1.2" />
      </Peers>
    </Discovery>
  </Domain>
</CycloneDDS>
```

首先, 将 INTERFACE_NAME 替换为计算机上用于 ROS 通信的实际 ROS 网络接口名称。示例:enp1s0、wlp2s0 等。

接下来, 调整“Peers”下的列表, 使其反映您网络中那些希望通过 ROS 相互连接的机器的 IP 地址。

与上述文件类似, 您的机器人上也存储着一个类似的文件。您无法直接访问该文件, 但可以通过 AX Portal 设置菜单中的“ROS 功能设置”部分对其内容进行编辑。在那里, “ROS 2 对等节点的 IP 地址”设置提供了一张列表, 您可以输入相应的 IP 地址。

注意:

- 更改生效需重启机器人。
- 若列表中的任何 IP 地址格式不正确, AX Portal 将报错且新列表不会被保存。

- 该列表必须包含机器人自身的 IP 地址。目前，该地址不会被自动读取。
- 当从列表中移除某个 IP 地址并保存新列表时，该 IP 地址也将从 Cyclone DDS 配置中移除。
- 由于防火墙设置，可能需要所有 ROS 参与者位于同一网络子网范围内。

7.4.3.2.2 多播

以下是另一个 Cyclone DDS 配置示例，用于将您的机器配置为支持多播：

```
<?xml version="1.0" encoding="utf-8"?>
<CycloneDDS
  xmlns="https://cdds.io/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://cdds.io/config
https://raw.githubusercontent.com/eclipse-cyclonedds/cyclonedds/master/etc/cyclonedds.xsd
"
>
  <Domain Id="any">
    <General>
      <Interfaces>
        <NetworkInterface name="INTERFACE_NAME" multicast="true" />
      </Interfaces>
      <AllowMulticast>true</AllowMulticast>
      <EnableMulticastLoopback>true</EnableMulticastLoopback>
    </General>
  </Domain>
</CycloneDDS>
```

同样，请将 INTERFACE_NAME 替换为计算机上用于 ROS 通信的实际 ROS 网络接口名称。示例：enp1s0、wlp2s0 等。

若要将机器人切换为多播模式，只需切换“Multicast”设置按钮。同样，需要重启机器人才能使更改生效。

7.4.4 建立连接

现在，要在计算机上启动 ROS 环境，请导航至您的 Docker Compose 文件并运行：

```
docker compose up -d
```

使用以下命令验证 Docker 容器是否正在运行中：

```
docker ps -a
```

该命令将打印类似以下内容的输出：

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
132bdd6fa4de	osrf/ros:jazzy-desktop	"/ros_..."	...	...	...	...

使用以下命令以交互模式进入容器

```
docker exec -it ros2_development bash
```

每次进入容器时，请务必执行以下命令加载 ROS 配置文件：

```
source /opt/ros/jazzy/setup.bash
```

为了在同一网络中运行多台机器人，我们需要为每台机器人设置不同的 `ROS_DOMAIN_ID`。请进入 AX Portal 中的“ROS 功能设置”，并设置所需的 ROS 域 ID。允许的范围为 6-41。此外，请注意此操作完成后必须重启机器人。现在，请返回已打开 ROS Docker 容器的终端，并将 `ROS_DOMAIN_ID` 设置为相同的数值：

```
export ROS_DOMAIN_ID=SELECTED_NUMBER
```

此时，您应能通过以下命令看到名为“机器人”的 AX Portal ROS 节点：

```
ros2 node list
```

7.4.5 通过 ROS 访问机器人数据

看到该节点后，可通过运行以下命令查看可用主题列表：

```
ros2 topic list
```

通过 ROS 2 主题，您可以实时获取机器人的数据。例如，要订阅关节位置、速度等数据，请运行：

```
ros2 topic echo /joint_states
```

7.4.6 脚本中的 ROS 2 功能

您可以在脚本中直接进口 `rclpy`。这允许您创建 ROS 2 物体，如发布者、订阅者等。

ROS 处理程序会创建其专属的 ROS 节点。请使用 `ros_handler.ros_node()` 函数访问该节点。注意：请勿在此节点上运行 `rclpy.spin`，因为该操作已在 ROS 处理程序内部完成。

7.4.7 执行操作

AX Portal ROS 节点会创建一个名为“/robot/execute_function”的操作服务器，用于执行桥接函数。该操作采用来自 `generic_msgs` 包 (<https://github.com/Epson-Robots/epson-ax-ros2>) 的自定义 `ExecuteFunction` 操作，必须安装该包才能获取操作定义。将该包克隆到您的工作区，并使用 `colcon` 进行构建：

```
mkdir -p /colcon_ws/src
cd /colcon_ws/src
git clone https://github.com/Epson-Robots/epson-ax-ros2
cd ..
colcon build --packages-up-to generic_msgs
```

```
source install/setup.bash
```

为了向机器人发送 ROS2 动作，您的客户端需要获取活动控制权。为此，请在终端中运行以下命令，并在 AX Portal 图形界面中确认对话框：

```
ros2 action send_goal --feedback /execute_function generic_msgs/action/ExecuteFunction  
"{action: request_active_control, bridge: core, arguments: '{}'}"
```

现在，您可以通过终端使用以下命令运行桥接函数：

```
ros2 action send_goal --feedback /execute_function generic_msgs/action/ExecuteFunction  
"{action: read_actuator_position, bridge: core, arguments: '{\"actuator_ids\": 6}'}"  
  
ros2 action send_goal /execute_function generic_msgs/action/ExecuteFunction "{action:  
move_joints, bridge: core, arguments: '{\"joints\": 6, \"joint_position\": 100.0}'}"
```

7.4.8 验证设置

当前 ROS 处理程序仅支持无身份验证模式。在桌面界面的“设置”菜单中，您需要手动禁用此特定连接模块的身份验证。

Authentication Settings

Disable Authentication for ROS



图 7.4.8:ROS 验证设置

7.5 Modbus连接模块

7.5.1 概述

Modbus 模块采用的协议是 TCP Modbus。在服务器端，该模块基于 Python 开发，并通过 PyModbus 库实现。有关详细信息，请参阅 PyModbus 文档。该模块分为读取寄存器（模拟输入寄存器，30000-39999）和写入寄存器（模拟输出保持寄存器，40000-49999）。任何 TCP Modbus 客户端均可通过 与服务器通信。所使用的端口为 **5020**。本章最后一部分提供了若干 Python 代码示例，其中部分示例说明了如何使用 Python 程序脚本与服务器建立连接。

7.5.2 设置

目前 Modbus 连接仅支持无身份验证模式。在桌面界面的“设置”菜单中，您需要手动禁用此特定连接模块的身份验证功能。

Authentication Settings

Disable Authentication for Modbus



图 7.5.2:Modbus 模块设置

7.5.3 写入寄存器(简单描述)

Modbus 插件会对命令寄存器和属性寄存器中的任何信息变更作出响应。清空写入寄存器的数据是用户的责任。

7.5.3.1 命令寄存器

向命令寄存器 (30000) 写入单个位以执行命令。某些命令需要在命令数据寄存器 (30001) 中提供参数。若无法同时写入这两个寄存器, 请务必先写入命令数据寄存器。

绝对地址:30000 - 30001 相对地址:0x000 - 0x001		
位	命令寄存器 (30000)	数据 (30001)
0	上电	工具 ID
1	休眠	
2	停止	
3	暂停	
4	继续运行	
5	校准	脚本 ID
6	释放	
7	保持	
8	更改项目	项目 ID
9	开始脚本	脚本 ID
10	移动到位姿	位姿ID
11	执行路径	路径ID
12	工具位置	
13	工具选择	
14 ... 15	保留	

7.5.3.2 属性寄存器

向属性寄存器 (30002) 写入一个比特即可执行命令。某些命令需要在属性数据寄存器 (30003) 中提供参数。如果无法同时写入这两个寄存器, 请务必先写入属性数据寄存器。

绝对地址: 30002 - 30003 相对地址: 0x002 - 0x003		
位	属性寄存器 (30002)	数据 (30003)
0	写入数字输出	I/O 状态
1	设置脚本事件	事件 ID
2	确认错误	
3	读取位置	
4	请求活动控制	
5	释放活动控制	
6	切换到手动模式	
7	切换到自动模式	
8	重置安全	
9	继续运动	
10 ... 15	保留	

7.5.3.3 用户写入寄存器

绝对地址: 30016 相对地址: 0x010		...	绝对地址: 30255 相对地址: 0x0FF	
0 ... 15	自定义数据		0 ... 15	自定义数据

使用用户写入寄存器将信息发送至脚本应用程序。脚本应用程序的 **modbus.write_user_register** 函数可用于读取自定义数据。

7.5.4 写入寄存器(详细描述)

7.5.4.1 不执行任何操作

清空写入寄存器的数据是用户的责任。要清空命令寄存器 (30000) 或属性寄存器 (30002), 只需写入整数 0。

0x000	0x001	0x002	0x003
0		0	

7.5.4.2 通电

在操作机器人之前, 必须先为其电机上电。数字 *n* 指定了要连接的自定义工具的 ID。机器人的状态将从“未上电”(1, 第 0 位) 变为“就绪”(8, 第 3 位)。

0x000	0x001	0x002	0x003
1	<i>n</i>		

7.5.4.3 休眠

工作日结束后, 机器人可以休眠其电机电源。无论处于何种状态, 机器人均可休眠电源。一旦机器人休眠电源, 状态将变为“未上电(1, 位 0)”。

0x000	0x001	0x002	0x003
2			

7.5.4.4 停止

中断所有机器人的运动, 并将状态从“运行中”(64, 第 6 位) 返回“就绪”(8, 第 3 位)。

0x000	0x001	0x002	0x003
4			

7.5.4.5 暂停

在机器人运行中暂停 (64, 第 6 位)。状态将切换为暂停 (128, 第 7 位)。

0x000	0x001	0x002	0x003
8			

7.5.4.6 继续运行

如果机器人处于“暂停”状态(128, 第 7 位), 则继续运行。状态将切换为“运行中”(64, 第 6 位)。

0x000	0x001	0x002	0x003
16			

7.5.4.7 校准

当前校准无效。此位保留。

0x000	0x001	0x002	0x003
32			

7.5.4.8 释放使能控制模式 (HGC)

启用所有机器人关节的手动引导控制, 使其处于重力补偿模式并可手动移动。状态需为“就绪”(8, 第 3 位), 并将切换至“使能控制”(2048, 第 11 位)。

0x000	0x001	0x002	0x003
64			

7.5.4.9 保持使能控制 (HGC)

将所有机器人关节保持在位置控制模式下, 使其无法再通过手动运动。状态需为使能控制(2048, 第 11 位), 并将切换为就绪(8, 第 3 位)。

0x000	0x001	0x002	0x003
128			

7.5.4.10 切换项目

将当前数据库项目更改为 ID 为 n 的项目。状态必须为“就绪”(8, 第 3 位)。

0x000	0x001	0x002	0x003
256	n		

7.5.4.11 开始脚本

运行 ID 为 n 的脚本。状态需为“就绪”(8, 第 3 位), 并将切换为“运行中”(64, 第 6 位)。脚本运行期间, 机器人可以暂停。脚本完成后, 状态将切换回“就绪”(8, 第 3 位)。

0x000	0x001	0x002	0x003
512	n		

7.5.4.12 移动到位姿

将机器人移动到ID为n的位姿。状态需为就绪(8, 第3位), 并将切换为运行中(64, 第6位)。机器人运动过程中, 其运动可以暂停。一旦运动完成, 状态将切换回就绪(8, 第3位)。

0x000	0x001	0x002	0x003
1024	n		

7.5.4.13 执行路径

执行 ID 为 n 的路径。状态需为“就绪”(8, 第 3 位), 并将切换为“运行中”(64, 第 6 位)。机器人移动期间, 其运动可被暂停。一旦移动完成, 状态将切换回“就绪”(8, 第 3 位)。

0x000	0x001	0x002	0x003
2048	n		

7.5.4.14 工具放置

使用工具放置物体(例如张开夹具)。状态需为“就绪”(8, 第3位), 并将切换为“运行中”(64, 第6位)。夹具移动期间, 其运动可暂停。一旦运动完成, 状态将切换回“就绪”(8, 第3位)。

0x000	0x001	0x002	0x003
4096			

7.5.4.15 工具拾取

使用工具拾取物体(例如关闭夹具)。状态需为“就绪”(8, 第3位), 并将切换为“运行中”(64, 第6位)。夹具移动期间, 其运动可被暂停。一旦运动结束, 状态将切换回“就绪”(8, 第3位)。

0x000	0x001	0x002	0x003
8192			

7.5.4.16 写入数字输出

该功能用于写入数字输出。

0x000	0x001	0x002	0x003

		1	d
--	--	---	---

7.5.4.17 设置脚本事件

如果用户应用程序需要等待 Modbus 设备发送的事件，可以使用脚本事件。共有 16 个事件。根据数据 d 的值，事件会被设置或清除。**modbus.wait_for_event** 脚本函数会等待直到特定事件被设置。如果 d 为 0，则清除所有事件。如果 d 为 65535，则设置所有事件。如果 d 为 9，则设置第一个和第四个事件，其余事件被清除。

0x000	0x001	0x002	0x003
		2	d

7.5.4.18 确认错误

错误必须被确认后才能清除。但这仅用于读取错误信息。确认操作不会影响机械臂状态。如果主错误代码和次错误代码 (40004-40005) 均为 0，则无需再确认其他错误。

0x000	0x001	0x002	0x003
		4	

7.5.4.19 读取位置

更新机器人的位置数据。每当请求位置数据时，当前的机器人关节角度和机器人坐标会被写入相应的读取寄存器 (40256-40267)。

0x000	0x001	0x002	0x003
		8	

7.5.4.20 请求活动控制

请求活动控制以向机器人发送命令。用户数据寄存器和读取寄存器可在无活动控制的情况下访问。当前拥有活动控制的监视器将收到通知，并需主动释放控制权。一旦控制权被授予，相应的读取寄存器 (40008) 将被设置。

0x000	0x001	0x002	0x003
		16	

7.5.4.21 释放活动控制

释放活动控制，以便另一个监视器可以向机器人发送命令。在没有活动控制的情况下，仍可访问用户数据寄存器和读取寄存器。相应的读取寄存器 (40008) 将切换为 0。

0x000	0x001	0x002	0x003
		32	

7.5.4.22 切换到手动模式

切换到手动模式。在此模式下，机器人仅在用户按下启用按钮时才会运动。请注意，机器人切换模式时需要执行安全重置。

0x000	0x001	0x002	0x003
		64	

7.5.4.23 切换到自动模式

切换到自动模式。请注意，机器人切换模式时需要进行安全重置。

0x000	0x001	0x002	0x003
		128	

7.5.4.24 安全重置

启动/重启互锁重置。对于某些安全功能，机器人需要用户执行明确的、有意的操作才能继续运行。要检查是否需要此类操作，请使用相应的读取寄存器 (40009)。

0x000	0x001	0x002	0x003
		256	

7.5.4.25 开始运动中断

安全中断发生后，机器人可继续运行并恢复运动。此属性可用于使机器人重新移动。否则，可发送停止命令来终止正在运行的应用程序。要检查是否需要此类操作，请使用相应的读取寄存器 (40009)。

0x000	0x001	0x002	0x003
		512	

7.5.5 读取寄存器

7.5.5.1 状态寄存器

使用状态寄存器读取当前机器人状态。状态以位号的形式提供。

7.5.5.1.1 第一个状态寄存器

绝对地址:40000 相对地址:0x000	
0	未上电
1	上电中
2	加载中
3	就绪
4	处理中
5	校准中
6	运行中
7	自主模式
8	暂停
9	继续
10	保护停止
11	安全防护
12	位置限制
13	使能控制 (HGC)
14	碰撞
15	急停

7.5.5.1.2 第二个状态寄存器

绝对地址:40001 相对地址:0x001	
0	正常停止
1	错误
2	断电中
3	停止中
4.. 15	保留

7.5.5.1.3 第三状态寄存器

绝对地址:40002 相对地址:0x002	
0 ... 15	保留

7.5.5.1.4 第四状态寄存器

绝对地址:40003 相对地址:0x003	
0 ... 15	保留

7.5.5.1.5 主要错误寄存器

绝对地址:40004 相对地址:0x004	
0 ... 15	关于错误类别的信息

主要错误寄存器读取当前错误标识符的前 2 个字节。使用属性寄存器确认错误, 以便读取下一个错误。

有关错误寄存器值的含义的详细信息, 请参阅“错误寄存器映射”。

7.5.5.1.6 次要错误寄存器

绝对地址:40005 相对地址:0x005	
0 ... 15	有关错误类别中具体错误的信息

轻微错误寄存器读取当前错误标识符的最后 2 个字节。使用属性寄存器确认错误，以便读取下一个错误。

有关错误寄存器数值的含义的详细信息，请参阅“错误寄存器映射”。

7.5.5.1.7 程序列表寄存器

绝对地址:40006 相对地址:0x006	
0 ... 15	当前脚本 ID

应用程序寄存器读取当前运行中的脚本的 ID。

7.5.5.1.8 校准寄存器

绝对地址:40007 相对地址:0x007	
0	机器人是否已校正

校准寄存器用于读取机器人是否已校准。0 表示机器人尚未标定。1 表示机器人已校准。请注意，机器人必须连接到其电机，才能提供有意义的校准反馈。

7.5.5.1.9 活动控制寄存器

绝对地址:40008 相对地址:0x008	
0	Modbus监视器是否具有活动控制权

活动控制寄存器用于读取 Modbus 监视器是否具有活动控制权。0 表示 Modbus 没有活动控制权。1 表示 Modbus 具有活动控制权。Modbus 监视器可以使用 0x30002 0x04 请求活动控制来请求活动控制权。

7.5.5.1.10 中断寄存器

中断寄存器指示活动的中断对话框选项。

绝对地址:40009 相对地址:0x009	
0	需要重置安全
1	继续执行运动
2	需切换到手动模式

3.. 15	保留
--------	----

7.5.5.1.11 操作模式寄存器

操作模式寄存器指示机器人处于手动操作模式 (1) 还是自动操作模式 (0)。

绝对地址:40010 相对地址:0x00A	
0	机器人是否处于手动操作模式

7.5.5.2 用户读取寄存器

绝对地址:40016 相对地址:0x010		...	绝对地址:40255 相对地址:0x0FF	
0 ... 15	自定义数据		0 ... 15	用户数据

使用用户读取寄存器从脚本程序接收信息。程序的 **modbus.read_user_register** 函数将读取数据。

7.5.5.3 位置寄存器(第一关节至第六关节)

绝对地址:40256 相对地址:0x100		...	绝对地址:40261 相对地址:0x105	
0 ... 15	第一关节的关节角度		0 ... 15	第六关节的关节角度

读取特定关节的角度值，单位为十分之一度。如果读取值为 101，则该关节的角度为 10.1°。该值仅在属性寄存器中写入“读取位置”标志时才会更新。若读取的值大于 32768(=2¹⁵)，则表示负值。减去 32768 即可得到绝对值。例如，值 32869 对应关节角度 -10.1°。

7.5.5.4 坐标寄存器 (x, y, z, X轴旋转, Y轴旋转, 俯仰)

绝对地址:40262 相对地址:0x106		...	绝对地址:40267 相对地址:0x10B	
0 ... 15	TCP 的 x 坐标		0 ... 15	TCP 的 Z轴旋转坐标

按 x、y、z、X轴旋转、俯仰、Z轴旋转的顺序读取刀具中心点的坐标或方向。空间坐标以十分之一毫米为单位读取。如果某个坐标读数为 314，则数值为 31.4 毫米。旋转坐标以十分之一度为单位读取。如果某个坐标读数为 271，则数值为 27.1 (mm 或 °)。仅当在属性寄存器中写入读取位置标志时，该值才会更新。若读取的数值大于 32768(=2¹⁵)，则表示负值。减去 32768 即可得到绝对值。例如，数值 33039 对应坐标值 -27.1 (mm 或 °)。

7.5.6 错误寄存器映射

错误信息始终显示在两个寄存器中。次要错误寄存器包含实际的错误编号。主要错误寄存器包含以下数值之一，这些数值映射到不同的错误类别如下：

101	附加功能
102	配置
103	数据库
104	控制器
105	编辑
106	桥接
107	连接
108	对话
109	分销商
110	活动
111	运动学
112	套接字
113	核心
114	安全
115	设置
116	信号
117	状态
118	脚本
119	线程

201	模块
202	康耐视
203	控制
204	ROS
205	MODBUS
206	REST
207	TCP

301	AX
302	CFG
303	CMK
304	CMD
305	CTL
306	ECAT
307	LIC
308	数学
309	MCM
310	其他
311	操作系统
312	PPM
313	PYAPI
314	SIG
315	SYS
316	TLC
317	UTIL

7.5.6.1 示例

如果错误寄存器读取值为 [113, 4]，则这对应于机器人触发的错误代码 **CORE_004**。

7.5.7 Python 代码示例

这些示例用于远程连接到机器人的外部系统。

7.5.7.1 示例 1

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # read all status
    address = 0x000
    count = 4
    status = client.read_input_registers(address, count)
    print("STATUS", status.registers[0], status.registers[1],
          status.registers[2], status.registers[3])

if __name__ == "__main__":
    main()
```

连接到服务器。读取机器人状态。

7.5.7.2 示例 2

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # command to connect to motors
    address = 0x000
    values = [1 << 0]
    client.write_registers(address, values, unit=1)

if __name__ == "__main__":
    main()
```

连接到服务器。将机器人连接到其电机。

提示:<< 命令用于位移操作。

位	命令	整数
0	1 << 0	1
5	1 << 5	32

7.5.7.3 示例 3

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # command to run script with id 5
    address = 0x000
    values = [1 << 9, 5]
    client.write_registers(address, values, unit=1)

if __name__ == "__main__":
    main()
```

连接到服务器。运行 ID 为 5 的程序。

7.5.7.4 示例 4

```
from pymodbus.client.sync import ModbusTcpClient

def main():
    # connect to client
    client = ModbusTcpClient("192.168.80.203", port=5020)
    client.connect()

    # read all error
    address = 0x004
    count = 2
    error = client.read_input_registers(address, count)
    print("ERROR", error.registers[0], error.registers[1])

    # acknowledge error
    address = 0x002
    values = [1 << 2]
    client.write_registers(address, values, unit=1)

if __name__ == "__main__":
    main()
```

连接到服务器。读取当前错误消息。确认错误消息。

8. Cognex 摄像头模块

8.1 概述

Cognex 相机模块允许您将预配置的 Cognex 相机连接到 AX Portal。连接后，您可以触发相机上的预配置任务来拍摄并处理图像。处理后的数据随后传输到 AX Portal，您可以在应用程序中读取这些数据。

8.2 摄像头设置

要设置 Cognex 摄像头以兼容 AX Portal，请按照以下步骤操作：

1. 下载并安装 Cognex 的 In-Sight Explorer 软件。
2. 为摄像头供电并将其连接至网络。若摄像头的网络设置为静态 IP 地址，您也可将其直接连接至计算机。
3. 启动 In-Sight Explorer，如系统提示请允许访问，并安装所需组件（例如 Microsoft Visual C++、Microsoft .NET Framework）。
4. 随后，Explorer 的 Easybuilder 视图将自动打开。
5. 导航至“系统”→“将传感器/设备添加到网络”。查找您的摄像头并将其添加到网络中。
6. 请确保摄像头的网络设置与 AX 系列机器人的网络设置兼容。同时请确保 Telnet 端口设置为 3000。
7. 选择您的摄像头并点击“连接”按钮，将其与 In-Sight Explorer 连接。
8. 现在您可以创建任务（例如用于校准摄像头，或识别供机器人拾取的物体），并将它们保存到摄像头中。更多详细信息请参阅康耐视 (Cognex) 文档。

8.3 摄像头程序

要在 AX Portal 应用程序中使用 Cognex 摄像头，

- 机器人和摄像头必须位于同一网络中，
- 摄像头的 Telnet 端口已设置为 3000，
- 且摄像头已设置为在线状态。

以下是一个示例，展示如何使用预配置的任务来检测圆的坐标。

```
# modify these settings to fit to your application
camera_ip = "10.0.0.210"
job_name = "CircleDetection"
event_id = 1
keyword = "coordinates"

# connect the camera to the application
cognex.connect_to_camera(camera_ip)
# select the job to trigger
cognex.change_job(camera_ip, job_name)
# trigger the event and wait for a response
response = cognex.get_camera_message(camera_ip, event_id, keyword)
print(response)
# close the connection
cognex.disconnect_camera(camera_ip)
```