

ROKAE 珞石

图形化编程 用户手册



人类生产生活的得力伙伴

图形化编程

用户手册

文档编号：202304131424

文档版本：V3.2

©版权所有 2015-2023 ROKAE 保留所有权利

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

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

本公司不可能预见所有的危险和后果，因此本手册不能警告用户所有可能的危险。

禁止擅自复印或转载本手册的部分或全部内容。

如您发现本手册的内容有误或需要改进抑或补充之处，请不吝指正。

本手册的原始语言为中文，所有其他语言版本均翻译自中文版本。

©版权所有 2015-2023 ROKAE 保留所有权利
珞石（山东）智能科技有限公司
中国·山东

目录

目录..... III

1 界面布局1

1.1 主界面布局1

1.2 导航栏2

1.3 模块工具箱2

1.4 程序工具栏2

1.5 编辑区3

1.5.1 程序树编辑区3

1.5.2 列表编辑区4

1.6 操作工具栏4

1.7 配置面板5

2 指令模块介绍6

2.1 基础模块6

2.1.1 运动6

2.1.2 赋值8

2.1.3 IO10

2.1.4 工具组11

2.1.5 等待12

2.1.6 暂停13

2.1.7 消息13

2.1.8 注释14

2.2 高级模块14

2.2.1 条件15

2.2.2 循环18

2.2.3 脚本19

2.3 工艺模块19

2.3.1 料盘19

2.3.2 码垛21

3 程序操作 24

3.1 基本使用 24

3.1.1 基本概念 24

3.1.2 基本操作 24

3.2 列表操作 25

3.2.1 通用操作 25

3.2.2 路点 25

3.2.3 变量 26

3.2.4 工具组 26

3.3 模块操作 30

3.3.1 属性 30

3.3.2 记录 30

3.4 本地运行 31

3.4.1 运行逻辑 31

3.4.2 运行操作 32

3.5 自动运行 32

3.5.1 生成脚本 32

3.5.2 RL 调用 33

4 示例场景 34

4.1 料盘上下料检查 34

修订记录 35

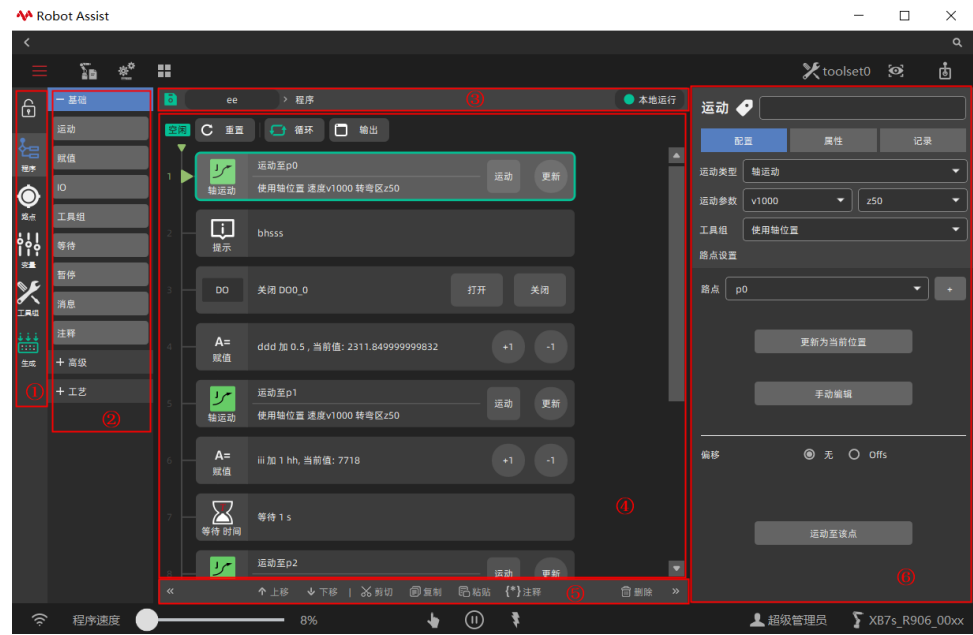
1 界面布局

本手册介绍了 Robot Assist 上用于机器人图形化编程的用户界面和操作说明。它为用户提供了一种可视化的交互式编程方式，用户可以通过简单地组合程序模块和设定预定义的参数对珞石机器人进行编程。

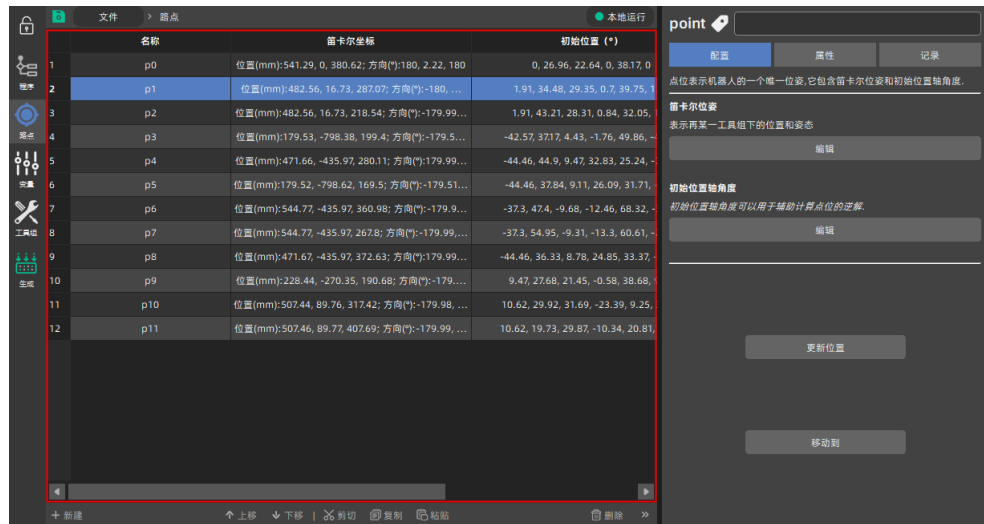
适合不同基础用户的编程平台，用户仅需少量的编程基础即可快速完成应用编程，大大提高工作效率。具有模块划分清晰、用户友好、性能高、运行稳定等特点。

1.1 主界面布局

点击顶部状态栏“菜单”按钮，点击“图形化编程”按钮进入图形化编程界面。



图形化编程主界面布局



列表编辑区

图形化编程界面由以下主要部分构成：

序号	选项	描述
1	导航栏	主要包含程序编辑器、点位管理列表、变量管理列表、工具组管理列表、RL 程序生成器。

2	模块工具箱	编程模块根据功能分为基础、高级、工艺。每个类型包含一些列功能模块，点击标签可展开分类；点击功能模块可以智能添加到程序树中，拖动功能模块可以拖放至任意有效位置。
3	程序工具栏	显示当前加载的程序以及页面名称，本地运行功能展开可以随时对图形化程序模块展开运行、单步、暂停、停止等控制功能。
4	编辑区	包括程序树编辑区以及列表编辑区。
5	操作工具栏	使用底部的操作工具栏可以修改程序树以及各变量列表。
6	配置面板	该区域包含模块特性设置、通用属性、日志记录功能。

1.2 导航栏

主界面最左侧的导航栏方便用户快速在常用功能之间进行切换，并提供程序锁、一键生成脚本等操作，如下图所示：



程序锁：
该功能提供程序、模块操作的防误触保护，点击后界面其他位置均处于不可点状态，且该按钮



呈现为上锁状态。

程序导航：
快捷切换至程序树操作界面。

路点导航：
快捷切换至路点列表编辑界面。

变量导航：
快捷切换至变量列表编辑界面。

工具组导航：
快捷切换至工具组列表编辑界面。

生成脚本：
图形化程序中机器人相关模块可以生成 RL 脚本，多用于自动模式脱离示教器长期运行。有 RL 编程相关经验的开发人员也可以选择通过该功能转换为脚本进行后续编辑。
每个图形化程序对应一个机器人工程，最终在运动任务中会生成名为 xcustom0.mod 的后置文件供原 RL 工程调用，有关 RL 工程使用方法参考机器人使用手册。

1.3 模块工具箱

该区域提供模块选择以及拖拽添加功能。直接点击模块标签可以在程序树中当前选中模块下方快速添加同级相应模块，如果当前未选中任何模块则会添加至程序树末尾，如果选择多个模块时则在最下面模块添加同级相应模块。模块标签同时支持拖拽，可以直接拖拽至指定位置和层级。
模块添加后自动处于选中状态，右侧配置面板显示该模块配置信息。

基础： 运动、赋值、IO、工具组、等待、暂停、消息、注释
高级： 条件、循环、脚本
工艺： 料盘、码垛

1.4 程序工具栏

该区域主要提供程序基本保存、本地运行功能，并显示当前编辑状态：程序、路点、变量、工具组，如下图所示：



保存按钮保存当前程序所有内容，包括程序树、路点列表、变量列表、工具组列表，并同步至远端控制器。
程序文件名与当前加载工程保持一致，即每个机器人工程对应一个图形化程序。
本地运行功能是通过本机设备直接运行图形化程序，对于指针跳转、单步运行等调试功能十分方便，且运行性能很高。点击展开功能如下图所示：



开始：重置程序树运行状态，指针跳转到第一行开始顺序执行程序。
单步：从当前指针位置开始执行一个模块，当无指针存在时重置程序书并从第一行开始执行。
暂停：打断当前指针执行模块，指针保持不变。
停止：打断当前指针执行模块，重置程序树。
*需要注意一些功能模块本地执行与机器人端执行可能会有一定的区别。

1.5 编辑区

编辑区域负责对程序内容进行编辑，出于用户操作方便以及兼容性等原因，图形化程序模块操作与变量操作分别位于程序树编辑区以及相应列表编辑区。不同区域针对内容提供相应便捷操作。

1.5.1 程序树编辑区

该区域以树形结构显示、编辑图形化功能更模块。每个功能模块通过水平连接线连接至任务线，顺序为从上至下依次执行；不同层级通过虚线进行区分，层级为从左至右依次递增。

任务工具栏：
程序树中所有模块均挂在在同一条任务线上，运行顺序按照任务线从上至下执行。任务线的顶端提供任务相关工具，如下图所示：



重置功能作用为重置程序树运行指针至首行。当程序树有改动时，下次执行本地运行、生成等程序相关操作均会自动重置。
循环切换按钮切换程序树是否循环运行，默认为单次运行。程序运行中也可动态切换循环状态。
输出按钮可以弹出或关闭输出框，用于显示功能模块输出，输出结果随程序文件保存而保存。

模块选中：



用户可对选中的当前模块进行编辑以及配合操作工具栏实现上移下移等操作，选中的模块高光显示，PC 端操作时按住 **shift** 可以框选多个模块，或按住 **ctrl** 键依次选中。

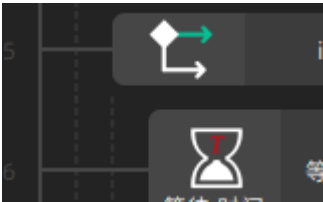
运行指针：



当程序运行时绿色三角形运行指针指向当前运行模块。重置按钮可以重新加载程序并将指针指

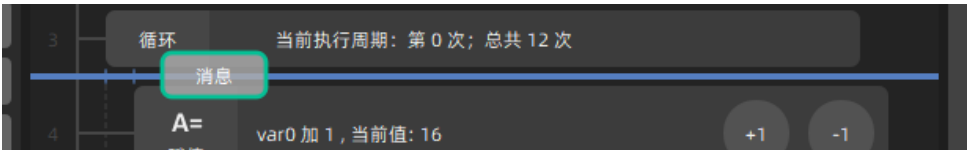
向首个模块：单步按钮仅执行指针指向模块；开始按钮即从指针指向模块开始向下执行。当模块执行出现错误时指针停留在错误模块；当没有运行指针存在时自动从首个模块开始执行。

模块层级：（虚线部分）



对于循环、分支等逻辑模块，其内部模块包含层级概念，通过调整模块层级可以表明模块所属逻辑区域，例如循环模块会多次执行层级大于循环模块所处层级的子模块。

拖动选择：



由模块工具箱直接通过拖拽交互添加的模块会在程序树编辑区显示其即将插入的位置以及对层级。拖拽至非程序树编辑区即可取消新建模块。

1.5.2 列表编辑区

该区域分别用于编辑路点、变量、工具组列表。

路点列表：

包含路点名称、笛卡尔坐标、初始位置。每个路点的笛卡尔坐标用于运动时均会随着当前工具组进行坐标系转换，其相关逆解机器人关节位置取距离初始位置最接近解；每个路点的初始位置同样可以直接用于轴空间运动，此时当前工具组仅规定末端负载。有关路点更多介绍请参考后续章节。

变量列表：

包含变量名称、类型、值。每个变量用于保存程序对应，且所有运行时对于变量的修改均会实时保存。有关变量更多介绍请参考后续章节。

工具组列表：

包含工具组名称、参考坐标系、末端坐标系、负载。每个工具组保留一套完备的机器人运动参考坐标系，同时保留工具、工件等信息以保证末端负载、机器人手持工具、外部工件等所有参考系相关信息的一致性，避免出现由于提示不够明显而带来的工具工件不配套导致的异常轨迹。有关工具组更多介绍请参考后续章节。

1.6 操作工具栏

该区域提供程序树的模块编辑以及表格编辑操作，如下图所示：



左侧折叠：处于程序树编辑界面时折叠或展开模块工具箱。

新建：处于表格编辑界面时点击新建按钮可以直接新建相应内容。

上移：向上移动选中模块或行，程序树与表格编辑均可。

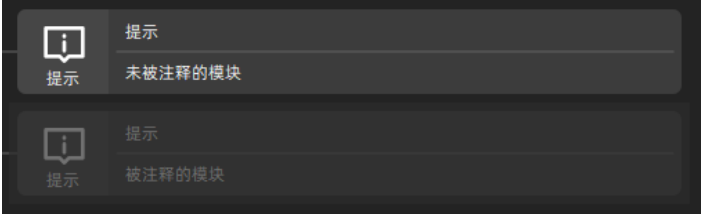
下移： 向下移动选中模块或行，程序树与表格编辑均可。

剪切： 暂不支持。

复制： 复制当前选中行，仅支持表格编辑。

粘贴： 粘贴选中行并自动重命名，仅支持表格编辑，支持多选。

注释： 注释当前选中行，被注释的模块将不参与程序的本地运行及 RL 构建。仅支持程序树编辑界面，支持多选。支持反注释。界面如下图所示：



删除： 删除当前选中内容，程序树与表格编辑均可，支持多选。

右侧折叠： 折叠或展开配置面板。

1.7 配置面板

该区域位于编辑区域右侧，主要用于程序树模块以及各列表的自定义配置、共有属性、操作记录。

配置：
该页面主要显示当前选中模块或当前列表选中行的具体特性配置，每个模块配置不尽相同，具体模块具体配置参考模块配置介绍。

属性：
模块或者列表的通用属性设置，属于高级功能。模块属性主要包括层级、生成脚本、自定义颜色等功能；列表属性则为列表内容逐项手动编辑，用以应对更通用的高级操作。

记录：
图形化编程目前提供了较为详细的运行日志记录，属于高级功能。通常情况下需要过滤具体模块，记录内容主要包括操作记录、反馈以及错误等。

2 指令模块介绍

2.1 基础模块

该分类为用户机器人最基础的操作，例如运动、更新位置、IO、调整等待时间调整时序等。建议现场工人可以进行学习以能够根据现场动态调整、换料换产。

2.1.1 运动

运动模块提供机器人运动指令，完整的运动需要配合路点、工具组等信息共同使用。同时模块界面还提供了运动至该点、更新位置的快捷操作。

模块界面如下图所示：



模块配置：

运动模块配置信息分为运动轨迹设置与路点设置。运动轨迹用于描述机器人运动至目标路点的轨迹特征，主要分为轴运动、直线运动、弧线运动，每种运动类型下还需要规定运动轨迹的速度、转弯区以及工具组。



轴运动：

该运动形式为机器人轴关节空间运动，是机器人最快速的运动形式，常用于回原点、过渡位姿等无高精度运动姿态要求的运动，允许直接使用点位初始位置以固定关节角度作为目标位置，保证在不同工具工件下相同的机器人关节位置。

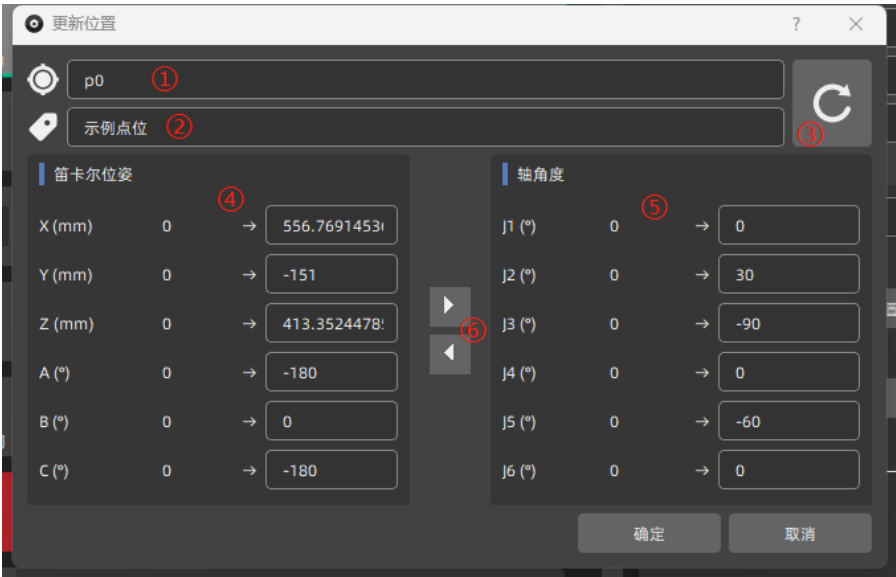


直线运动：

该运动形式为机器人末端相对参考坐标系的直线运动，坐标系信息统一通过工具组指定。通常情况下使用全局工具组以适配不同工具组，也可以指定固定工具组作为直线运动方式。

路点设置：

运动轨迹从当前位置开始，轨迹经过的点位称为路点。路点可以通过路点列表编辑界面进行详细设置，也可以在运动模块直接新建、运动、更新、手动编辑路点。路点编辑界面如下图所示：



具体功能为：

序号	名称	描述
1	路点名称	路点脚本中的英语名称，无重复
2	路点描述	路点中文描述，可缺省
3	更新位置	获取当前机器人点位信息
4	笛卡尔位姿	当前路点笛卡尔位姿
5	轴角度	当前路点轴角度
6	正/逆解	通过笛卡尔位姿/轴角度求出另一组解

路点具体值编辑确定后仍然可以在运动模块配置中进行偏移微调，该调整只针对于笛卡尔位置。

Offs: 路点笛卡尔位姿进行 xyz 方向调整。

Reltool: 路点笛卡尔位姿绕末端坐标系转动。暂未提供支持。

选定偏移方式后点击“运动至该点”可以手动运动以验证运动效果，出于安全等因素考虑，点击该按钮并不会立即开始运动，而是弹出统一的运动窗口进一步确认，如下图所示：



在运动窗口确认前还可以修改运动类型以及速度，例如一条高速直线运动轨迹的路点进行调试时，并不必按照原运动参数到达该路点，并且仍然可以通过该窗口内的偏移进行反复调整保证

位置准确。点击确认按钮后所有运动参数将不再能修改，如下图所示：



处于该界面后，开始运动按钮通过红色方框提示点击该按钮即将开始运动，长按该按钮开始运动，松开立即停止运动。此时如果需要调整位置可以点击后退按钮重新修改参数。

转弯区：

运动模块将对程序树进行前瞻操作，以提供转弯区功能。当前瞻过程中，变量模块、IO 模块、注释模块都不会打断前瞻，其他模块、工具组切换、层级改变时将打断前瞻，不与下一条的运动模块构成转弯区。

快捷操作：

模块界面上自带运动与更新按钮，分别对应路点设置的“更新为当前位置”，以及“运动至该点”两个常用功能，点击后弹出统一窗口进行进一步操作确认，并不会直接修改点位或者运动机器人，减少用户误触带来的风险。
新建运动模块时自动新建默认非重复路点名称，如果希望直接新建运动至当前示教位置可以直接点击“+”进行点位新建。

操作案例：

- 第一步：处于程序树界面时，拖拽或点击模块工具箱中的“运动模块”至程序树合理位置。
- 第二步：在配置界面编辑运动类型、参数、工具组。
- 第三步：使用已有路点或者点击“+”进入点位编辑窗口更新为当前示教点。
- 第四步：需要偏移可以进行设置并尝试运动。
完成

注意事项：

当运动模块出现报错信息（类似“目标点超出运动范围”）时，受限于图形化前瞻操作的影响，需要沿程序树向下寻找真正的报错路点。“单步”运行能更快速定位报错路点。

2.1.2 赋值

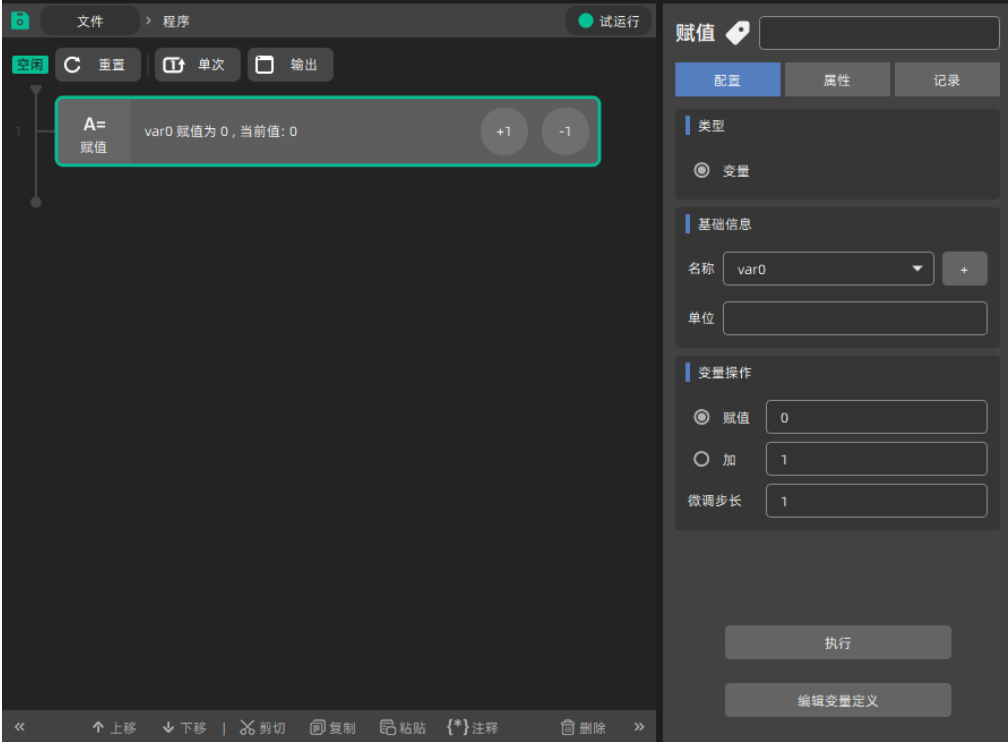
赋值模块主要提供对变量的赋值和修改功能。将赋值模块插入程序树中，模块界面如下图所示：



变量赋值：

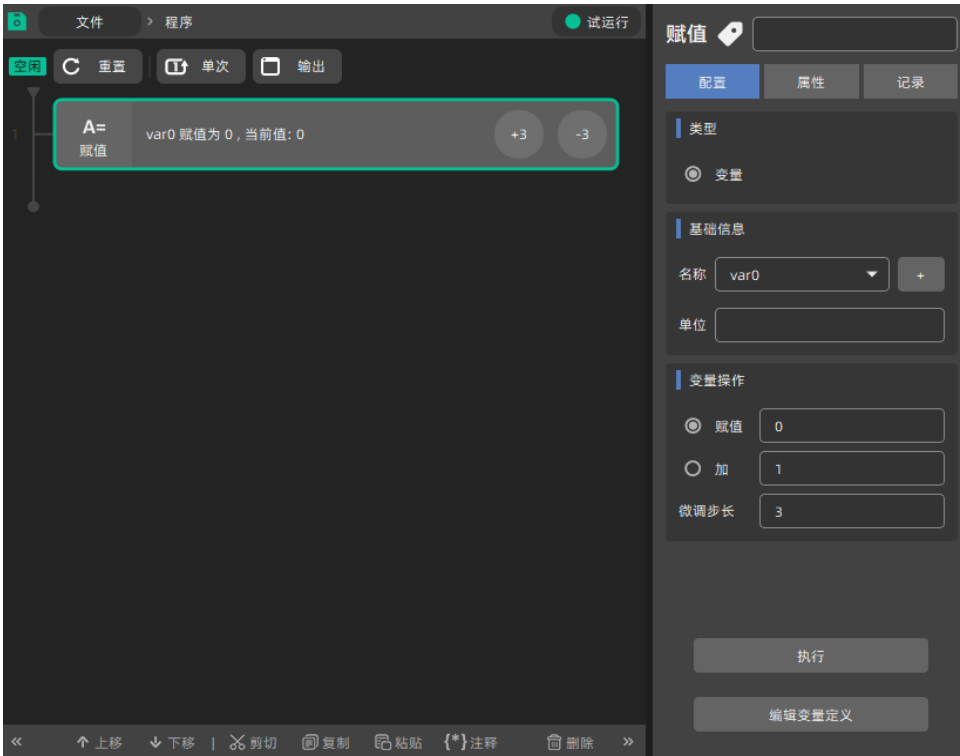
在右侧赋值配置窗口的“类型”组别中选中“变量”，则进入变量赋值模式。其模块显示及配

置窗口如下图所示：



在右侧配置窗口区的“基础信息”组别中通过下拉列表选择要赋值变量的名称，也可通过单击右侧“+”快捷按钮直接新建一个变量。变量的操作支持“赋值”和“加”两种，在“变量操作”组别中选中“赋值”，并在后边对应的编辑框中输入数值，则运行时，变量的值会被修改为当前设置的值；选中“加”，并在后边对应编辑框架中输入递增值，在运行时，会对变量执行加递增值的操作，此操作只支持数值型变量。点击下方的“执行”按钮，可以执行一次当前所设置的“赋值”或“加”操作。可使用配置窗口下方“编辑变量定义”按钮，实现对选中变量的快捷编辑功能。

另外，可以使用模块上的圆形“+1”、“-1”按钮实现的数值型变量的微调功能，默认的调整步长为 1，可以通过设置“微调步长”来改变变量调整的幅度。比如可将“微调步长”设置为 3，如下图所示。



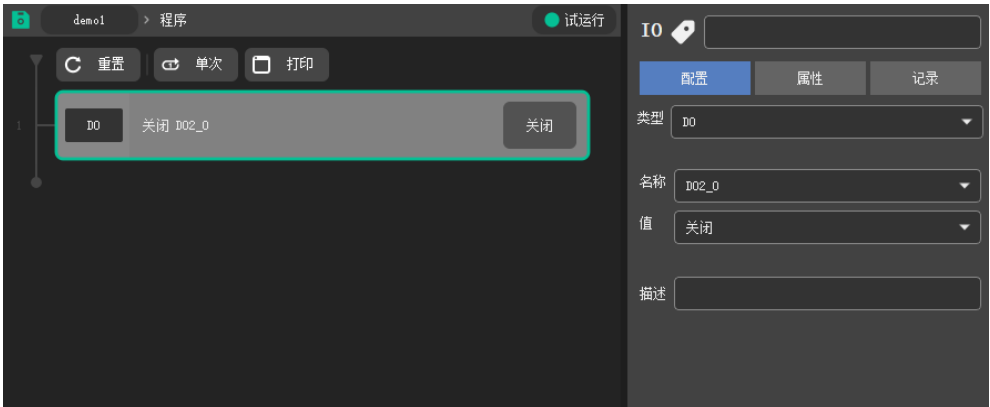
2.1.3 IO

IO 模块主要实现对外部信号的输出功能，支持 DO、AO、寄存器三种类型，默认为 DO，其模块显示如下图所示：



DO 输出：

在右侧配置窗口“类型”下拉列表中选择“DO”，则进入 DO 输出模式。其模块显示及配置窗口如下图所示：



在“名称”下拉列表中选择需要输出的 DO 信号，在“值”下拉列表中选择“关闭”或“打开”，在运行时，会对选择的 DO 信号执行打开或关闭操作。也可通过模块上的“打开”或“关闭”快捷按钮进行 DO 操作。

AO 输出：

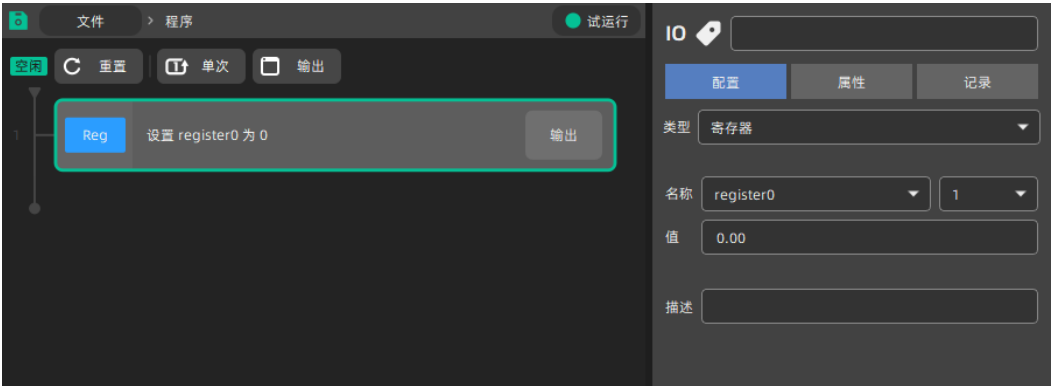
在右侧配置窗口“类型”下拉列表中选择“AO”，则进入 AO 输出模式。其模块显示及配置窗口如下图所示：



在“名称”下拉列表中选择需要输出的 AO 信号，在“值”编辑框中输入需要设置的值，在运行时，会对选择的 AO 信号执行设置操作。也可通过模块上的“输出”快捷按钮对 AO 信号进行设置。

寄存器输出：

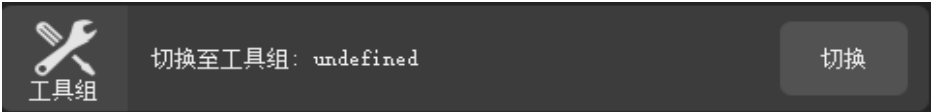
在右侧配置窗口“类型”下拉列表中选择“寄存器”，则进入寄存器输出模式。其模块显示及配置窗口如下图所示：



在“名称”下拉列表中选择需要输出的寄存器信号，如寄存器是数组，在其后的下拉列表中选择序号，在“值”编辑框中输入需要设置的值，在运行时，会对选择的寄存器信号执行输出操作。也可通过模块上的“输出”快捷按钮对寄存器信号进行设置。

2.1.4 工具组

工具组模块主要实现全局工具组的切换，即更新全局的工具、工件坐标系，其模块显示如下图所示：



模块配置：

点击选中工具组模块，配置窗口如下图所示：



在右侧配置窗口“设置工具组”下拉列表中选择需要切换的工具组，下方的说明会对应显示所选工具组的参考坐标系、末端坐标系、负载的数值。在运行时，会把全局工具组切换为当前所选的工具组。也可以通过模块上的“切换”按钮及下方的“立即切换”按钮直接执行工具组切换操作。

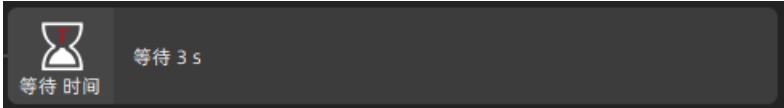
注意：在切换工具组后，右上方会分别显示工具的名称，如图所示



，表明此时使用的是全局工具组，所有的Jog操作及坐标显示都会基于当前生效的全局工具组坐标系。

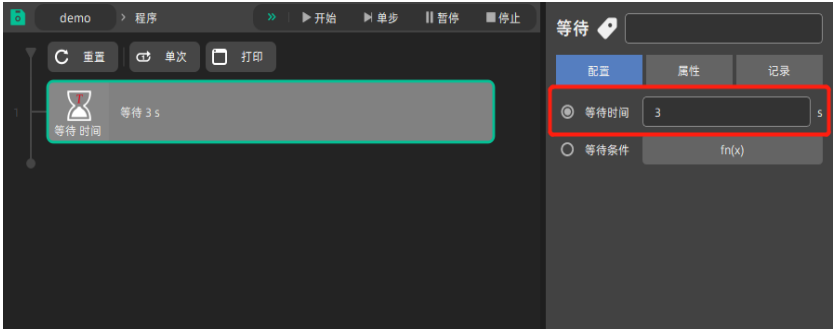
2.1.5 等待

等待模块主要实现程序等待某一条件成立时，继续往下执行，否则程序等待在该模块。该模块提供两种等待选项，一是等待时间，二是等待条件。将等待模块插入程序树中，模块界面如下图所示：



等待时间：

在右侧等待配置窗口中选择“等待时间”，在输入框中填入需要等待的时间。当程序树执行到等待时间模块时，程序会在该模块等待设定的时间，然后进入其模块显示及配置窗口如下图所示：



等待条件：

在右侧等待配置窗口中选择“等待条件”，点击“fn(x)”按钮，在条件选择对话框中选择需要等待的条件，点击“确认”。“fn(x)”按钮的使用方法请参考“条件”模块。

等待条件分为等待 IO 和等待 function 两种类型：



等待 IO：

当条件类型为 DI、DO、AI、AO 的时候，为“等待 IO”；



等待 function：

当条件类型为 Var、Evaluate 的时候，为“等待 function”。

2.1.6 暂停

暂停模块提供了程序暂停的功能。当程序树执行到该模块时，程序暂停，不再执行后续的模块。将暂停模块插入程序树中，暂停模块界面如下图所示：



暂停模块不提供配置界面。

2.1.7 消息

消息模块主要提供消息提示的功能。当程序树执行到该模块时，界面将弹出一个对话框显示配置的消息文本，并打断程序的执行状态。将消息模块插入程序树中，消息模块界面如下图所示：



编辑消息：

可以在右侧配置窗口填写需要提示的文本内容。如下图所示：



执行效果：

执行到该模块时，程序树将弹出对话框来显示消息文本，并点图所示，弹出对话框，提示“夹具未检测到工件”：



2.1.8 注释

注释模块提供对程序树中某一部分内容进行注释说明的功能。该模块不参与程序的运行。需要注意的是，该模块需要和所注释的内容保持一致的层级关系，否则会影响程序的执行逻辑。将注释模块插入程序树中，模块界面如下图所示：



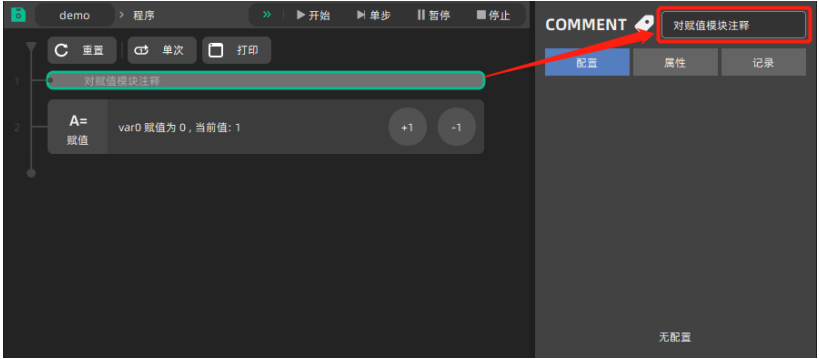
程序注释：

注释模块的右侧配置窗口没有具体的配置信息。如需对程序树内容进行注释时，在下图中红框位置填写文本信息。



示例：

对赋值模块进行注释说明



2.2 高级模块

高级模块面向有一定编程经验的开发人员、技术支持，需要具备一定的逻辑、设计能力，以及对于机器人的理解，目前主要包括条件分支等跳转模块。

2.2.1 条件

条件模块主要提供状态判断的功能，可以根据 IO 的状态或变量的值来改变程序树的执行逻辑。条件模块分为三种类型：**if**、**else if** 和 **else**。

在一个条件模块执行逻辑中必须包含一个**仅且只有一个 if** 模块；

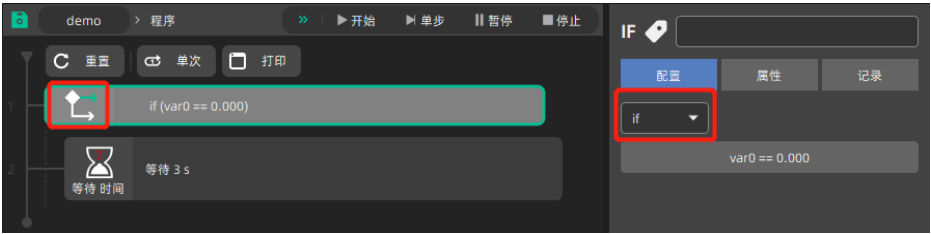
else 模块必须在一个条件模块执行逻辑中的末尾，其在一个条件模块执行逻辑中的个数**不能超过 1 个**；

else if 模块要处于 **if** 模块之后和 **else** 之前。

首次将条件模块插入程序树中时，模块为未初始化状态，模块界面如下图所示：



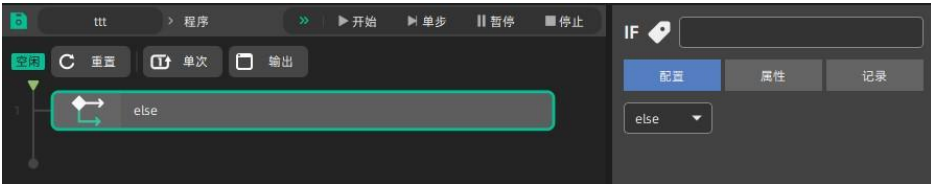
if 模块：



else if 模块：



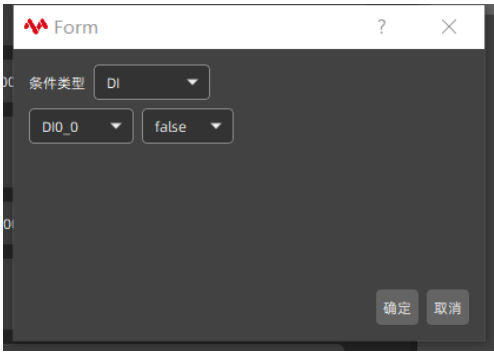
else 模块：



条件选择对话框：

条件选择对话框中提供的判断条件类型有：DI、DO、AI、AO、Var、Reg、Evaluate 等判断条件。

可以根据程序的编写逻辑来选择不同的条件类型以及其需要满足的条件。**Evaluate** 目前实现有限，不建议使用该条件类型。



IO 类型：包含 DI、DO、AI、AO 四种信号类型。当程序树执行到该条件模块时，程序树将判断机器人的信号状态以决定其后续的执行逻辑。

寄存器类型：在运行时，会对选择的寄存器信号执行输出操作

变量类型：包含 Var 类型。当程序树执行到该条件模块时，程序树将判断工程中的变量值以决定其后续的执行逻辑。

未定义状态：



如果条件选择未完成或条件选择错误，模块将显示为未定义状态，如果此时本地运行或构建 RL 工程，程序树将执行失败。

示例 1 单 if 结构：



示例 2 if - else 结构：



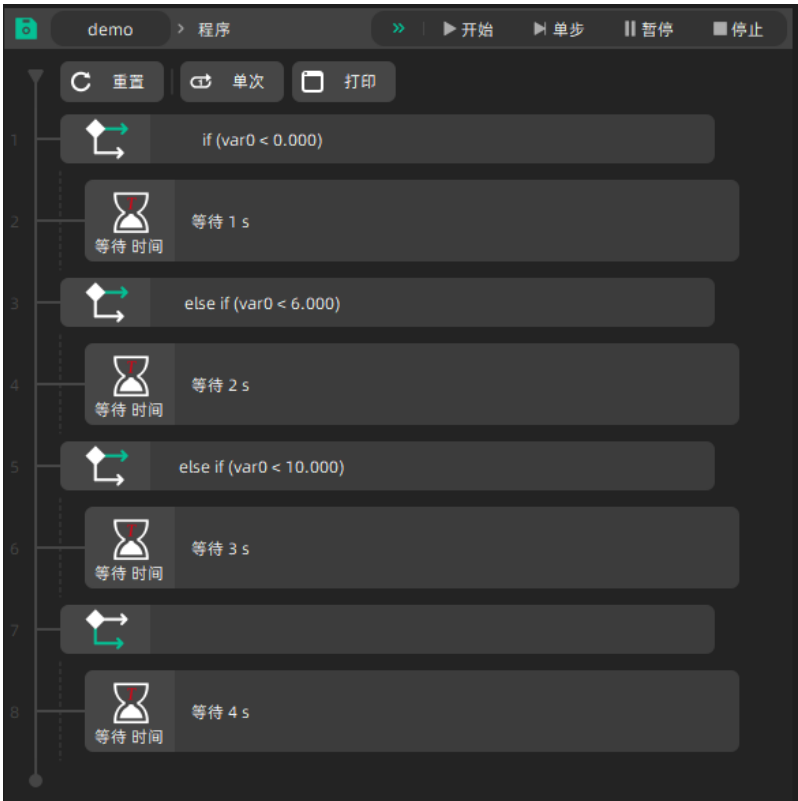
示例 3 if - else if 结构:



示例 4 if - else if - else 结构:



示例 5 if - else if - else if - else 结构:



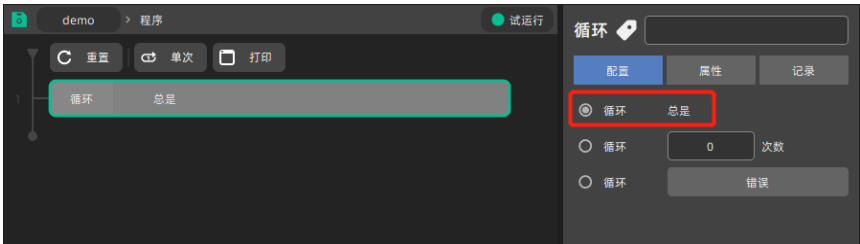
2.2.2 循环

循环模块的功能是重复执行循环模块层级之下的模块。循环模块提供三种循环类型：无限循环，指定次数循环，条件循环。将循环模块插入程序树中，模块界面如下图所示：



无限循环：

在右侧配置界面选择“循环 总是”。程序树将一直在重复执行循环模块所包含的子模块。



指定次数循环：

在右侧配置界面选择“循环 次数”，在输入框中填入需要循环的次数。程序树将重复执行循环模块所包含的子模块指定次数，次数从 1 开始。



条件循环：

在右侧配置界面选择“循环 条件”，点击“条件”按钮，在弹出的条件选择对话框中，选择进入循环的条件，点击“确定”按钮。“条件选择对话框”的使用方法请参考“条件”模块。



2.2.3 脚本

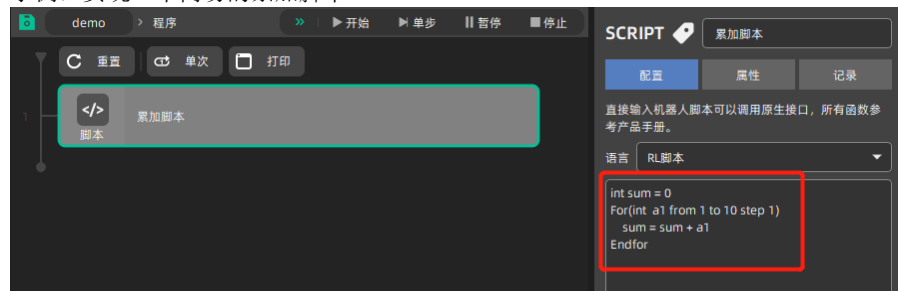
脚本模块提供更高级的用法，经验丰富的编程人员可以编写 RL 脚本进行定制化的开发。该模块对“本地运行”支持有限，对“自动运行”较为完整。将脚本模块插入程序树中，模块界面如下图所示：



RL 脚本：

RL 脚本为珞石机器人系统原生语言，其使用方法请参考产品说明书。

示例：实现一个简易的累加脚本。



2.3 工艺模块

2.3.1 料盘

料盘工艺是为机床上下料解决方案开发的一项料盘功能。用户可以通过使用 xCore 系统已配置好的料盘工艺包，在图形化编程中快速实现料盘上下料的工艺场景。

2.3.1.1 料盘系统变量

创建料盘模板时，为了料盘功能正常使用，若不存在料盘则会自动生成相关的全局系统变量，料盘相关的变量如下：

- 1、料盘工具组，命名为：**TrayToolset**，所有料盘模块共用此工具组；料盘模块执行或更新时，会将料盘工艺包的工具组更新到此工具组变量。
- 2、三个路点，命名为：**TrayApproachPoint**(料盘接近点)、**TrayWobjPoint**(料盘工件点)、**TrayRetractPoint**(料盘回撤点)，所有料盘共用这三个全局路点，料盘在运行时，会动态更新这三个路点的位姿。
- 3、一个整形变量，一般命名为：**tray0_index**(当前工件序号)，一个料盘模块会自动生成一个，用户可通过此变量来控制料盘输出某个工件序号的点位。

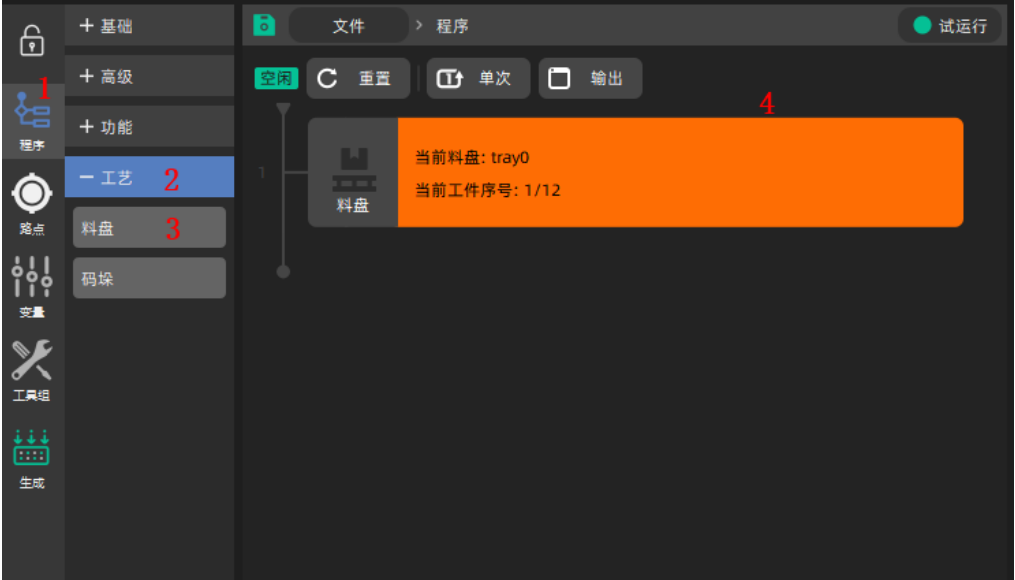
注意：料盘模块自动创建的系统变量，在列表中不可见，用户不能编辑和删除，但可以在程序

模块中选择和使用；在删除料盘模块时，其自动创建的 `tray0_index` 变量会被自动删除。

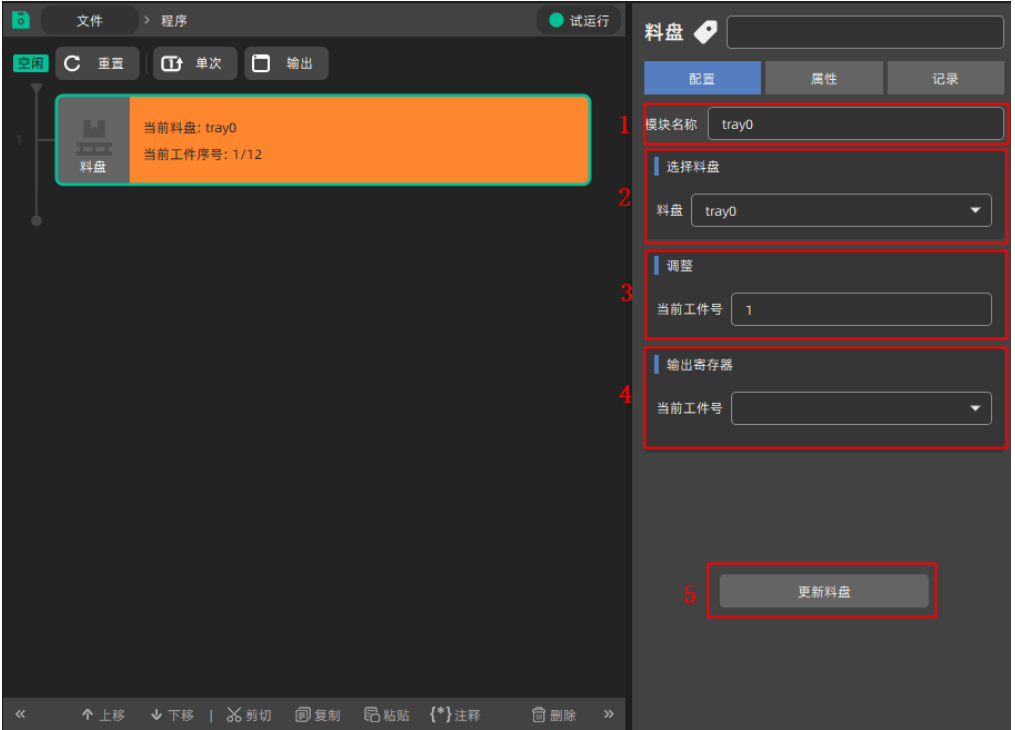
2.3.1.2 创建料盘模板

进入示教器图形化编程主界面，点击“程序”选项卡，进入程序编辑器界面。通过以下两种方式将料盘模块插入到程序树中：

- 点击模块列表“工艺” - “料盘” 按钮，模块自动插入到程序树中指定位置
- 点击按住“料盘” 按钮，拖动模块至程序树中任意位置



配置料盘属性，关联料盘工艺包、调整当前工件序号、设置输出寄存器。



序号	选项	描述
1	模块名称	用于显示模块变量的名称，默认不可修改。
2	选择料盘	可以从下拉列表中选择配置好的料盘工艺。料盘工艺包的配置可详见料盘工艺配置相关说明书。

3	当前工件序号	设定当前码放工件序号，修改该变量值可修改起始码放位置，默认最小起始位置序号为 1。
4	输出寄存器	在执行程序时将相关变量值输出至对应寄存器。将料盘的变量与寄存器提前关联并保存配置。
5	更新料盘	点击更新“更新料盘”，会根据当前工件序号，将当前料盘工艺的工具组、接近点、工件点、回撤点更新至 TrayToolset、TrayApproachPoint、TrayWobjPoint、TrayRetractPoint 系统变量中，可使用此按钮功能进行调试。

2.3.1.3 料盘模块运行

料盘模块运行时会执行以下操作：

- 1、会根据当前工件序号，将当前料盘工艺的工具组、接近点、工件点、回撤点更新至 TrayToolset、TrayApproachPoint、TrayWobjPoint、TrayRetractPoint 系统变量中，用户可以使用系统变量进行工具组切换和运动。
- 2、对相应的当前工件序号执行加 1 操作，比如：tray0_index(当前工件序号)执行加 1 操作。

2.3.2 码垛

用户可以通过使用 xCore 控制系统中已经配置好的码垛工艺，在图形化编程中快速实现码垛搬运场景。

2.3.2.1 码垛系统变量

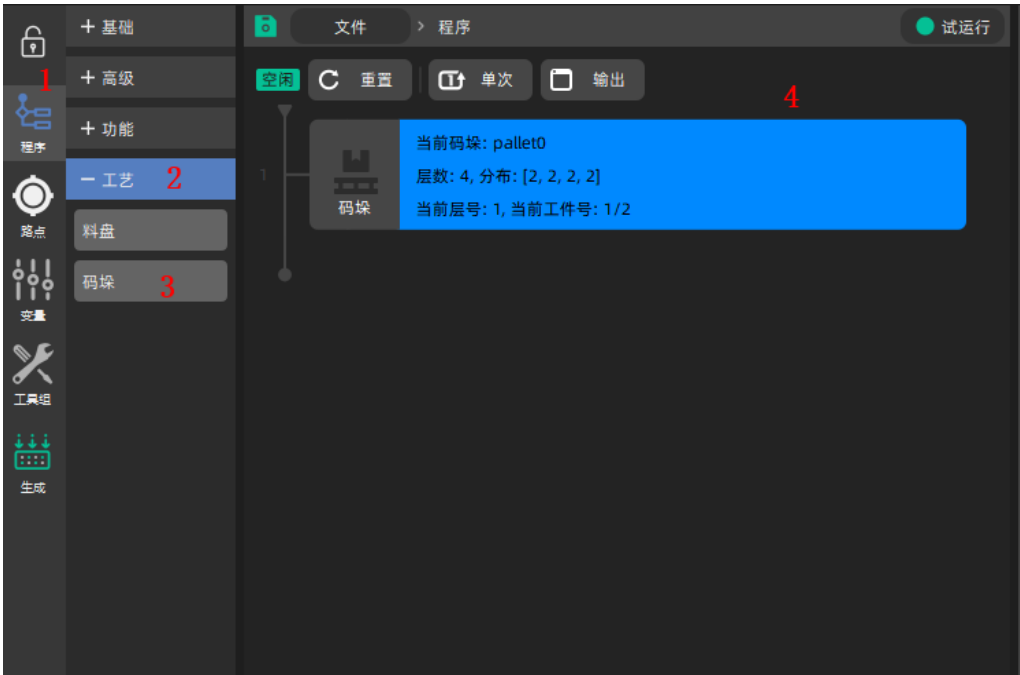
创建码垛模板时，为了码垛功能正常使用，若不存在码垛，则会自动生成相关的全局系统变量，码垛相关的变量如下：

- 1、码垛工具组，命名为：PalletToolset，所有码垛模块共用此工具组；码垛模块执行或更新时，会将码垛工艺的工具组更新到此工具组变量。
- 2、六个路点，命名为：PalletEnterPoint(进入点)、PalletReadyPoint(准备点)、PalletApproachPoint(接近点)、PalletWobjPoint(工件点)、PalletRetractPoint(回撤点)、PalletExitPoint(离开点)，所有码垛共用这六个全局路点，码垛在运行时，会动态更新这六个路点的位姿。
- 3、两个整形变量，一般命名为：pallet0_layer(当前层序号)、pallet0_wobj(当前层工件序号)，每个码垛模块会自动生成两个，用户可通过这两个变量来确定唯一的码垛工件点。
注意：码垛模块自动创建的系统变量，在列表中不可见，用户不能编辑和删除，但可以在程序模块中选择和使用；在删除码垛模块时，其自动创建的 pallet0_layer(当前层序号)、pallet0_wobj(当前层工件序号)变量会被自动删除。

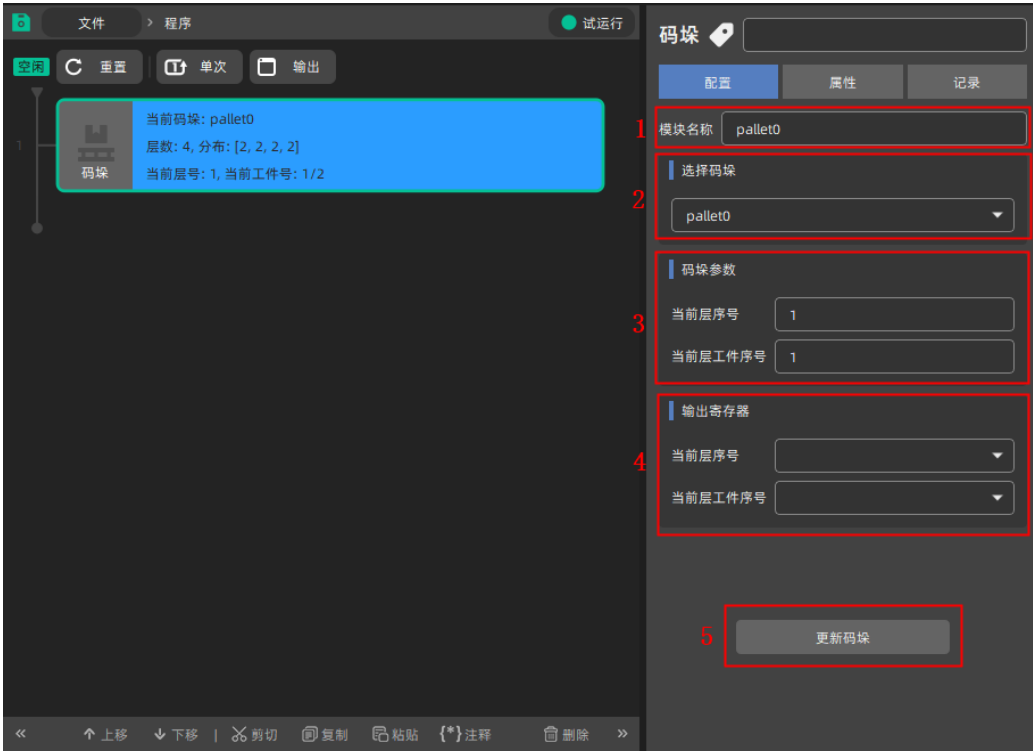
2.3.2.2 创建码垛模板

进入示教器图形化编程主界面，点击“程序”选项卡，进入程序编辑器界面。通过以下两种方式将码垛模块插入到程序树中：

- 点击模块列表“工艺” - “码垛”按钮，模块自动插入到程序树中指定位置
- 点击按住“码垛”按钮，拖动模块至程序树中任意位置



配置码垛属性，关联码垛工艺包、调整当前工件序号、设置输出寄存器。



序号	选项	描述
1	模块名称	用于显示模块变量的名称，默认不可修改。
2	选择码垛	可以从下拉列表中选择配置好的码垛工艺。码垛工艺包的配置可详见码垛工艺配置相关说明书。
3	码垛参数	当前层序号：设定当前码放层序号，修改该变量值可修改起始码放层位置，默认最小起始位置序号为 1。 当前层工件序号：设置当前层内的工件序号，修改此变量可以修改当前层内工件序号；
4	输出寄存器	在执行程序时将相关变量值输出至对应寄存器。将料盘

		的变量与寄存器提前关联并保存配置。
5	更新码垛	点击更新“更新码垛”，会根据当前层序号、当前层工件序号，将当前码垛工艺的工具组、进入点、准备点接近点、工件点、回撤点、离开点更新至 PalletToolset 、 PalletEnterPoint 、 PalletReadyPoint 、 PalletApproachPoint 、 PalletWobjPoint 、 PalletRetractPoint 、 PalletExitPoint 系统变量中，可使用此按钮功能进行调试。

2.3.2.3 码垛模块运行

- 码垛模块运行时会执行以下操作：
- 1、会根据当前层序号、当前层工件序号，工具组、进入点、准备点接近点、工件点、回撤点、离开点更新至 **PalletToolset**、**PalletEnterPoint**、**PalletReadyPoint**、**PalletApproachPoint**、**PalletWobjPoint**、**PalletRetractPoint**、**PalletExitPoint** 系统变量中，用户可以使用系统变量进行工具组切换和运动。
 - 2、会对相应的当前层工件序号，比如：**pallet0_wobj**(当前层工件序号)执行加 1 操作，若 **pallet0_wobj** 大于当前层的总工件数，则 **pallet0_layer**(当前层序号)加 1，并将 **pallet0_wobj** 赋值为 1。

3 程序操作

3.1 基本使用

图形化编程致力于通过友好的界面交互、多元化的功能模块、自定义的定制配置实现更加综合的应用场景。每个功能模块不仅仅是单一机器人功能，还可能是外部设备、工艺包、快捷操作、状态展示等，不同模块复杂程度也不尽相同，通过更有针对性的设计降低不同行业应用客户的使用成本。

3.1.1 基本概念

由图形化功能模块组成的程序树执行按照脚本运行逻辑顺序自上而下执行，对于连接机器人的运行设备示教器或台式机，可以直接在本地运行；对于希望机器人长期自动运行的工厂生产需要点击生成翻译成后置脚本发送至机器人。有关运行的更多基本概念如下：

全局数据：

路点、变量、工具组等列表属于全局公共数据，每个模块都可以直接使用这些全局变量。还可以用来作为不同模块之间的数据传递。

程序指针：

指向当前正在或即将运行模块。

任务线：

程序树左侧连接任务模块的连接线，可以显示程序指针、行号、模块层级等信息。

起始任务：

任务线的起点，以三角符号显示，右侧为程序操作区。

终止任务：

任务线的终点，以圆形符号显示，运行到此处程序停止。

重置：

手动重置程序指针至首个模块。

单次、循环：

当程序指针运行到最后一行时的处理，单次即指向终止任务；循环即指向起始任务重新执行。

输出：

弹出或隐藏程序运行输出框。

3.1.2 基本操作

图形化程序目前属于机器人工程的一部分，每个工程对应一个图形化程序。

新建工程：

1. 对于初次使用机器人的用户，根据机器人使用手册，首先要新建工程，点击左上角工程按钮：



2. 点击工程选项：

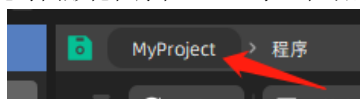


3. 点击右下角新建按钮根据向导新建工程（工程其它介绍详见机器人使用手册）。

4. 切换回图形化编程界面：



5. 此时图形化程序栏已经显示工程名，已经可以正常使用，如添加运动模块等。

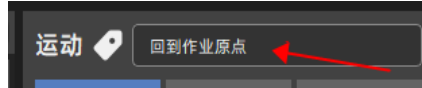


旧工程适配：

对于正在使用无图形化功能的用户升级后，原工程会自动生成工程文件进行加载。

保存工程：

点击保存按钮保存工程、列表、日志信息，并同步至机器人。

修改标签：

每个功能模块、列表变量均可以自定义备注信息用于对重要内容进行标记，输入完成后点击任意位置自动保存。不同模块对于备注标签会有不同显示效果。

***注意：**

修改路点标签后，运动模块默认直接显示“运动至路点标签”，不再显示路点英文名称。同样变量模块也会在模块面板上直接显示用户自定义的标签，便于用户理解。

***注意：**

通过工程管理界面进入导入图形化工程时，需要确认工程名称是否与当前列表中工程重名，如果重名，建议使用“导入选项”中的“自动替换”进行工程的导入操作。

3.2 列表操作

列表用于存储、展示全局数据。图形化编程专门优化列表显示性能，可以支持大量数据显示。

3.2.1 通用操作

对于不同全局数据类型的列表，大部分操作是通用的，每种列表的配置与新建工程是有区别的，将在后面章节介绍。

复制粘贴：

选中单行或多行点击复制或粘贴实现复制功能，自动重命名。

删除：

选中单行或多行点击删除。

属性： 右侧配置区域属性页，属于高级功能

基本属性包括名称与标签，名称用于编程，需要保证唯一性，默认生成的名称会自动进行重名检查且需要符合命名规范；标签用于用户自定义标记，便于最终用户理解。

参数设置包括各功能的详细参数的手动修改，以保证高级用户可以查看修改路点、变量、工具组等实际细节数据。

属性的修改属于高级操作，不会立即生效，需要手动点击下方保存按钮进行确认。

3.2.2 路点

路点是描述机器人唯一位姿的完整信息，通常与工具组、运动结合使用。

笛卡尔位姿：

表示在当前工具组中，机器人末端 TCP 相对参考坐标系的位姿。通常情况下机器人末端手持工具，操作外部工件，笛卡尔位姿即工具相对工件位姿；对于手持工件操作外部工具的情况，该位姿描述的是工件相对工具的位姿。

位姿由平移以及旋转构成：平移为空间 xyz 方向位移，单位为毫米；旋转为相应 rx, ry, rz, 即 RPY（滚动角、俯仰角、偏航角），单位为度。对于包含冗余轴的机型，如 7 轴（含臂角）

机器人，该位姿需要包含臂角角度，以保证信息完备。

初始轴位置：

表示该位置的关节角度。在笛卡尔位姿发生变化，或者工具组发生改变时，该轴位置被用于确定逆解唯一性。例如手动将笛卡尔位姿进行微调，但是并没有更新初始轴位置，实际使用工具组运动至该路点时，通过修改后的笛卡尔位姿以及原初始轴位置确定唯一逆解。

轴位置由机器人所有关节的轴角度构成，单位为度。通常 6 轴机型机器人关节角度为 J1 ~ J6，7 轴及其它轴数机型机器人轴位置的关节数目相应发生变化。

路点配置：

更新位置功能可以快速更新路点值为当前机器人示教位姿，点击刷新按钮可以手动更新。笛卡尔位姿使用当前工具组作为参考坐标系，轴位置即当前关节位置。更新后仍可手动编辑。

选择当前正在运行的运动指令窗口，选择运动参数以及目标位姿进行确认进行运动，在实际运动开始前，仍然可以进行偏移微调。

3.2.3 变量

图形化编程的变量目前包括整数、实数、字符串三种常用类型。每个变量全局使用，可以用于数据记录、信号传递等。程序中如果修改变量的值，则该变量实际保存的数值也会立即变化。如需常数变量保持不变等用法，用户可以自行设计变量读写。目前变量尚无特殊配置，使用属性面板进行通用修改。

3.2.4 工具组

工具组包含了机器人末端以及外部参考的工具工件所有信息。与传统的工具、工件设置不同的是，工具组将所有坐标系信息作为整体一起考虑，以防止误操作导致的工具与工件不匹配，从而损坏精细工具等严重问题。

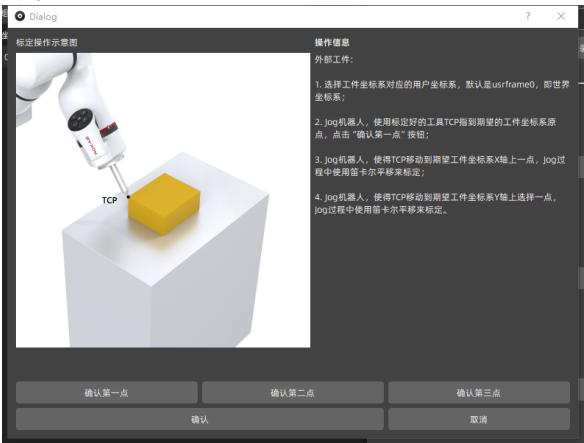
参考坐标系：

通常指外部工件相对世界坐标系的位姿转换，对于手持工件的情况，描述外部工具相对世界的位姿转换。

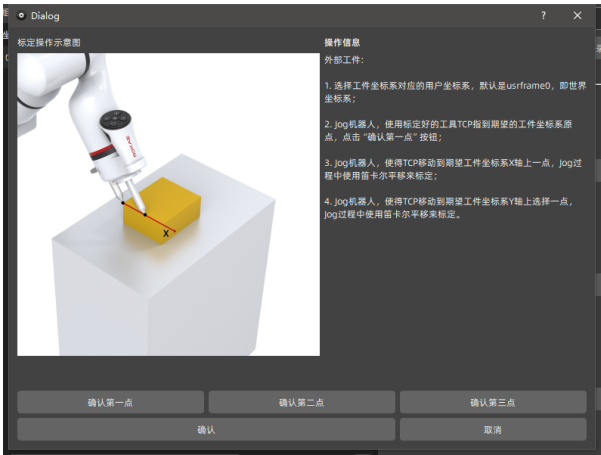
三点法标定参考坐标系（外部工件）：

1 选择当前坐标系并点击标定。

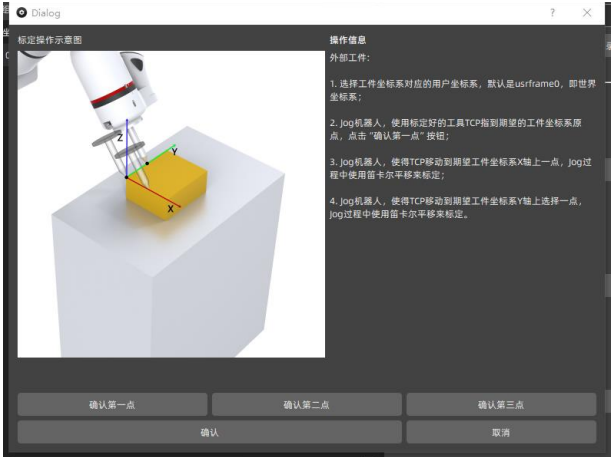
2 手动 jog 机器人，使用标定好的末端坐标系（工具）TCP 指到期望的参考坐标系原点，确认为第一点。



3 手动 jog 机器人，使得 TCP 指到参考坐标系 X 轴正方向上的一点，确认为第二点。



4 手动 jog 机器人，使得 TCP 指到参考坐标系 Y 轴正方向上的一点，确认为第三点。



5 点击确定计算参考坐标系。

末端坐标系（工具）：

通常指手持末端工具相对法兰坐标系的位姿转换，对于外部工具的情况，描述手持工件相对法兰坐标系的位姿转换。

六点法标定末端坐标系（手持工具）：

1 选择当前坐标系并点击标定。

2 手动 jog 机器人，使得选定的 TCP 与外部尖端点接触，确认为第一点。



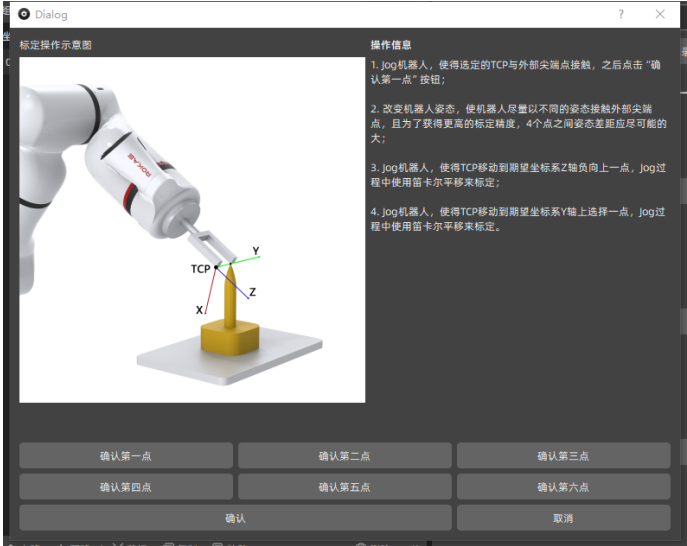
3 改变机器人姿态，使得机器人尽量以不同的另外三个姿态接触外部尖端点，姿态差距尽可能大，确认为第二至第四点。



4 手动 jog 机器人，使得 TCP 移动到期望坐标系 Z 轴负向上一，确认为第五点。



5 手动 jog 机器人，使得 TCP 移动到期望坐标系 Y 轴上任意点，确认为第六点。

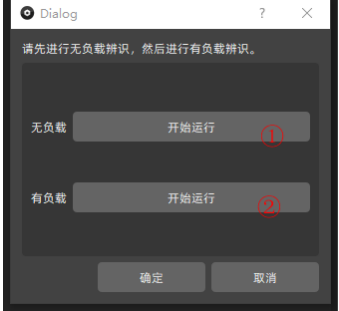


6 点击确定计算末端坐标系。

负载:

描述机器人末端手持负载，通常指手持末端工具的质量分布信息。
负载包括质量、质心坐标、惯量

负载辨识:



1 拆除负载，将机器人切换至自动运行、上电，确认周围无障碍物后运行无负载辨识。

- 2 待机器人运动结束后，下电按装负载，重新上电并切换至自动状态，确认周围无障碍物后运行有负载辨识。
- 3 待系统计算结果后点击确定保存。

3.3 模块操作

图形化程序模块相对独立，每个模块的界面设计以及配置是针对模块功能单独开发设计，通用的模块操作主要包括属性和记录。

3.3.1 属性

模块属性多为高级操作，建议有经验的用户配合高级功能使用。

调整层级：



模块层级表示模块之间互相包含关系，影响包含指针跳转的模块，如分支、循环等。通常情况下可以直接拖拽模块实现模块层级修改，同样可以在此处点击加减号或直接输入数值进行调整。

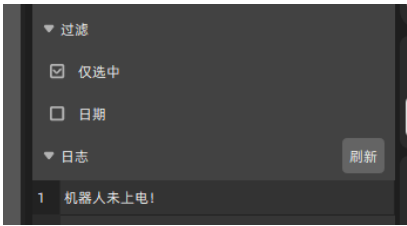
调整图形参数：



图形化模块对于高级用户开放了部分自定义颜色、控件的功能。颜色输入采用国际通用单词和16进制 RGB 格式。对于模块上的快捷按钮可以在此选择每个按钮的显隐。

3.3.2 记录

保存程序运行以及模块报错等日志信息。该区域不仅用来显示模块本身日志记录，还可用于查看全局日志记录，如下图所示：

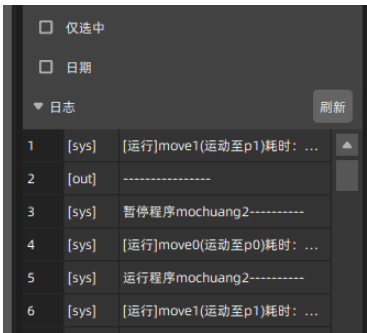


查看模块日志：

勾选“仅选中”选项过滤当前模块日志。通常包括模块运行、生成时产生的自定义提示信息、警告以及错误。勾选“日期”选项可以显示日志时间戳。日志按照添加顺序倒序排列，由于日志量较大，刷新日志需要手动点击刷新按钮。

查看全局日志：

更加详细的日志信息可通过去掉“仅选中”选项查看，例如下图所示：



此时首列会显示模块 ID，第二列是具体内容。全局日志主要包括其它各模块日志以及系统日志，系统日志主要用于记录程序运行时间、运行耗时等，随时记录运行效果。

保存日志：

当点击保存按钮时日志将自动保存至图形化程序文件，日志最多保存 1 万条，超过之后会按照插入顺序删除最早插入的部分。

3.4 本地运行

当图形化编程运行于示教器或 PC 端与机器人连接时，用户可以直接在设备上运行该程序。该功能对于混合多元图形化程序非常有用，例如包括快捷操作、视觉集成、自定义工艺、自定义模块等，包含了所有模块的运行逻辑。

3.4.1 运行逻辑

程序运行顺序为从上至下顺序执行，具体执行模块由程序指针指引。

合法性检查：

当程序树中的模块发生结构性变化时需要重新检查程序，只有程序所有内容合法时才会开始运行。如果某个模块合法性检查失败，显示窗口自动跳转至该行并选中该模块进行报错提示。

指针跳转：

通常情况下模块执行结束程序指针会跳转到下一行，一些高级模块会根据用户配置的条件进行动态指针跳转。使用运行或单步执行到该模块均会发生相同的跳转动作。

手动跳转指针：

在某些调试场景下，运行过程被打断时需要绕过一些运动、IO 等未开发完善的模块。此时可以使用注释功能或者直接点击模块左侧图标手动跳转程序指针。

示例场景：判断如果收到 PLC 发出的某 IO 信号时，机器人执行作业。此时调试过程中 PLC 尚未调试完成，那么可以直接手动跳转至运动模块进行后续机器人作业。该操作在实际应用场景可以极大方便熟悉程序的调试人员在不同模块之间的调试。

重置指针：

当数据需要重新更新，或需要重新从首行开始运行，可以点击重置按钮进行复位。

3.4.2 运行操作

运行图形化程序：

1. 点击本地运行按钮展开运行面板，点击开始按钮。
2. 如果程序模块发生增加或删除，将重新检查程序正确性，此时程序指针将重置，否则不重置。
3. 如果无程序指针指向首行。
4. 从指针指向行开始持续执行，直至程序末尾。如果选中循环模式则会持续运行。
5. 出现错误时运行暂停，指针保留。

单步运行图形化程序：

1. 点击单步按钮。
2. 如果程序模块发生增加或删除，将重新检查程序正确性，此时程序指针将重置，否则不重置。
3. 如果无程序指针指向首行。
4. 从指针指向行开始执行一个模块，正确执行后指针跳转结束执行。
5. 出现错误时运行暂停，指针保留。

暂停图形化程序：

1. 点击暂停按钮停止当前运行模块。
2. 待模块暂停后程序指针保留在当前模块。

停止图形化程序：

1. 点击停止按钮停止当前运行模块。
2. 待模块停止后指针重置。

3.5 自动运行

机器人在脱离示教器或上位机设备长期自动运行时，图形化程序需要生成后置脚本并发送至机器人用于自动生产运行。机器人相关模块在本地运行与自动运行行为需要保持一致。

3.5.1 生成脚本

脚本生成需要机器人工程正确配置，保证存在唯一的运动任务。具体机器人工程配置方法参考机器人使用手册。

生成步骤：

1. 保证机器人连接。
2. 点击左侧生成按钮自动保存、同步图形化程序，并开始后置文件。
3. 自动进行程序模块合法性检查。
4. 自动生成路点、变量、工具组列表后置文件。

- 6. 自动自顶向下生成程序树中模块后置脚本。
- 7. 自动保存文件之当前机器人工程运动任务。

3.5.2 RL 调用

处于兼容性等因素考虑，图形化程序并未删除或替换原 RL 工程，而是生成图形化专有变量、RL 文档。

调用步骤：

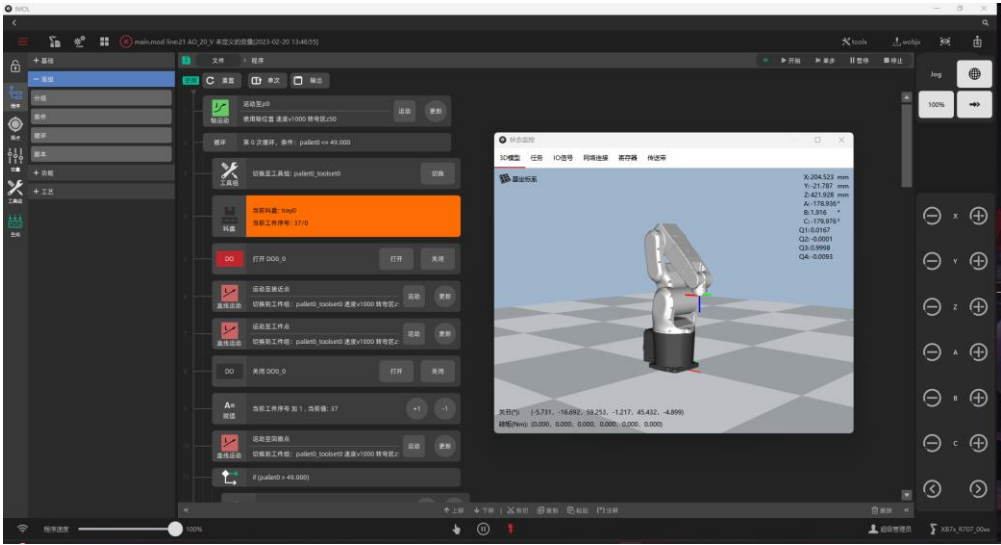


- 1. 进入 RL 程序编辑界面。
- 2. 进入唯一运动任务。
- 3. 检查已经生成 xcustom 脚本文件。
- 4. 在程序中调用 xcustom 函数。

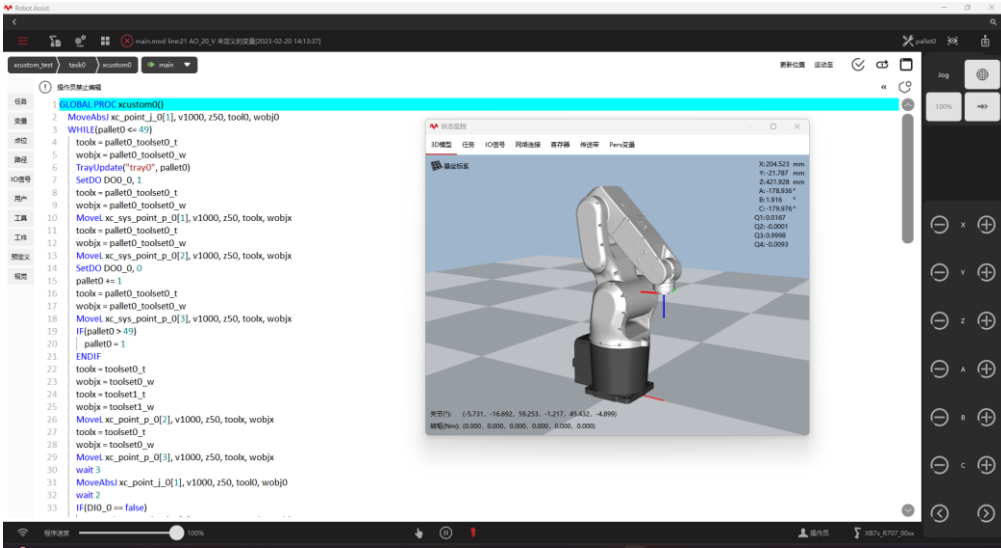
4 示例场景

4.1 料盘上下料检查

图形化界面例



生成脚本程序示例



修订记录

版本	修订内容
V3.0	初版发行
V3.1	完善细节、核对文档
V3.2	核对文档

ROKAE



珞石机器人

400-010-8700
www.rokae.com
sales@rokae.com