雷赛智能
Leadshine

2025/7

LV Studio 开发平台使用手册

- 非常感谢您本次购买雷赛产品
- 使用前请详细阅读此说明书，正确使用产品
- 请妥善保管此说明书

www.leisai.com

©Copyright 2025 Leadshine Technology Co.，Ltd.

All Rights Reserved.

前　言

版权说明

本手册版权归深圳市雷赛控制技术有限公司所有，未经本公司书面许可，任何人不得翻印、翻译和抄袭本手册中的任何内容。

本手册中的信息资料仅供参考。由于产品改进和功能升级等原因，雷赛公司保留对本资料的最终解释权，内容如有更改，恕不另行通知。

版本变更记录

修订日期	发布版本	变更内容
2024/12	V1.0	初版发行
2025/7	V1.5	新增运动控制功能

目　录

前　言 1

1. 产品简介 4

1.1 软件概述 4

1.2 运行环境 4

1.3 软件安装 4

2. 软件操作 9

2.1 软件 UI 概述 9

2.2 菜单栏 10

2.3 快捷工具条 18

2.4 工具箱 19

2.5 结果显示 21

3. 项目搭建-硬件资源配置 24

3.1 控制器 24

3.2 相机 31

3.3 通信设备 34

4. 项目搭建-流程配置 43

4.1 添加流程 43

4.2 配置流程 44

4.3 编辑流程 45

4.4 全局变量 50

5. 项目搭建-用户界面组态 51

5.1 用户界面介绍 51

5.2 用户界面组态 53

5.3 预览组态界面 54

5.4 控件工具箱 55

6. 工具使用说明 83

6.1 运动控制 83

6.2 图像采集 94

6.3 标定 95

6.4 定位 105

6.5 测量 137

6.6 识别 159

6.7 图像处理 161

6.8 颜色处理 170

6.9 系统工具 173

6.10 逻辑控制 213

6.11 通信 221

7. 案例搭建 229

7.1 上下料定位抓取 229

1. 产品简介

1.1 软件概述

LV Studio 是一款易于使用的低代码视控一体化开发平台，使用 LV Studio，您无需高级语言编程知识，LV Studio 可满足您对机器视觉和运动控制软件的所有要求，LV Studio 以前所未有的方式简化了构建视觉运控集于一体的应用程序。以往使用 API 编写运动控制和视觉应用程序，是一项专业且耗时的工作，为了让设备执行一项任务，往往需要数天甚至数周的编程和调试，且需要具备专业的知识，造成成本高、系统复杂、维护困难，这样显然难以应对制造业中常见的复杂多变的应用场景，使用 LV Studio 以上问题将迎刃而解。

LV Studio 综合视觉系统、运动控制系统和用户界面系统，设备商可以快速、高效且低成本地完成视控一体项目任务，成为人人可用的好帮手。相比于传统的 API 编程编程调试方式，使用门槛极大的降低，且无需编程和专业知识。LV Studio 可适配 DMC、PAC 等多种硬件平台，从而具有更广泛的适应性，完成多种多样的自动化任务。

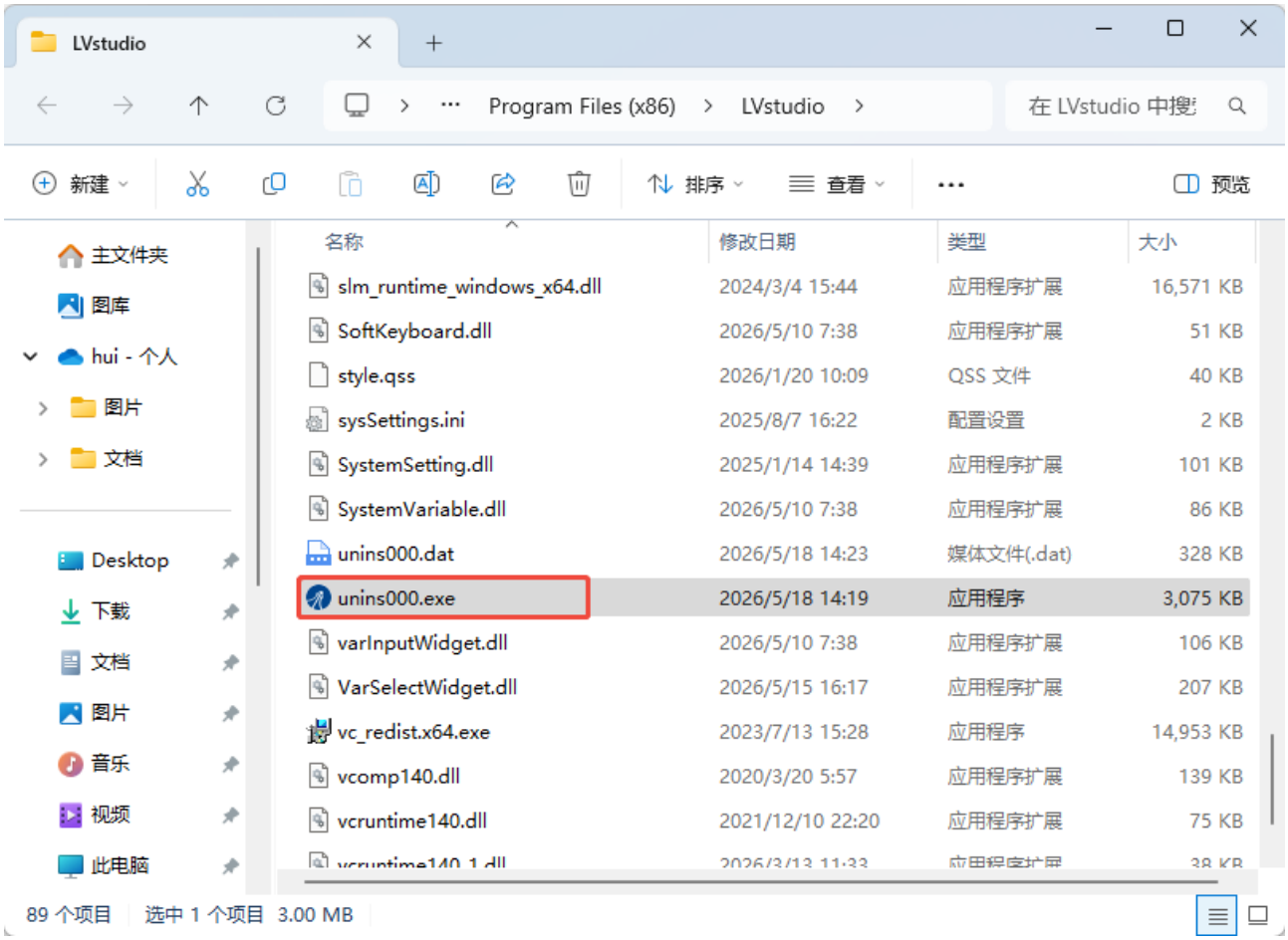
1.2 运行环境

为确保软件正常安装及运行，推荐使用以下 PC 配置。

项目	推荐配置	最低配置
CPU	多核 Intel Core i5 2GHz 或以上	Intel 赛扬 J1900
内存	8GB 或以上	4GB
硬盘	SSD 128GB 或以上	SSD 64GB
显卡	显存 1GB 或以上	
网卡	Intel i210 系列或以上	
USB 接口	USB3.0 接口（如需使用 USB3.0 相机）	
操作系统	Win10 64 位	Windows7

1.3 软件安装

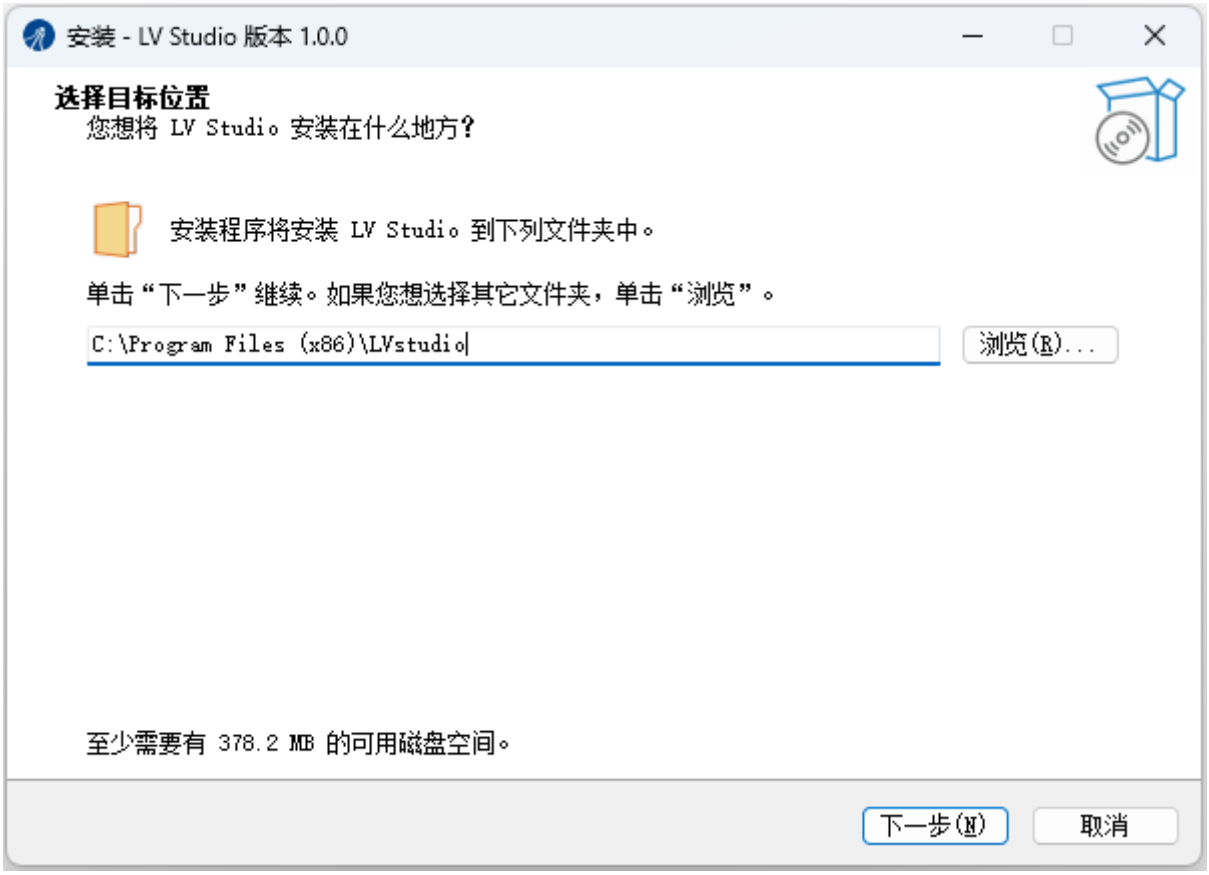
在安装 LV Studio 之前，若已安装之前版本，需先手动卸载老版本，再运行安装包。卸载文件 unins000.exe，默认在软件运行目录下，如下图 C:\Program Files (x86)\LV Studio。



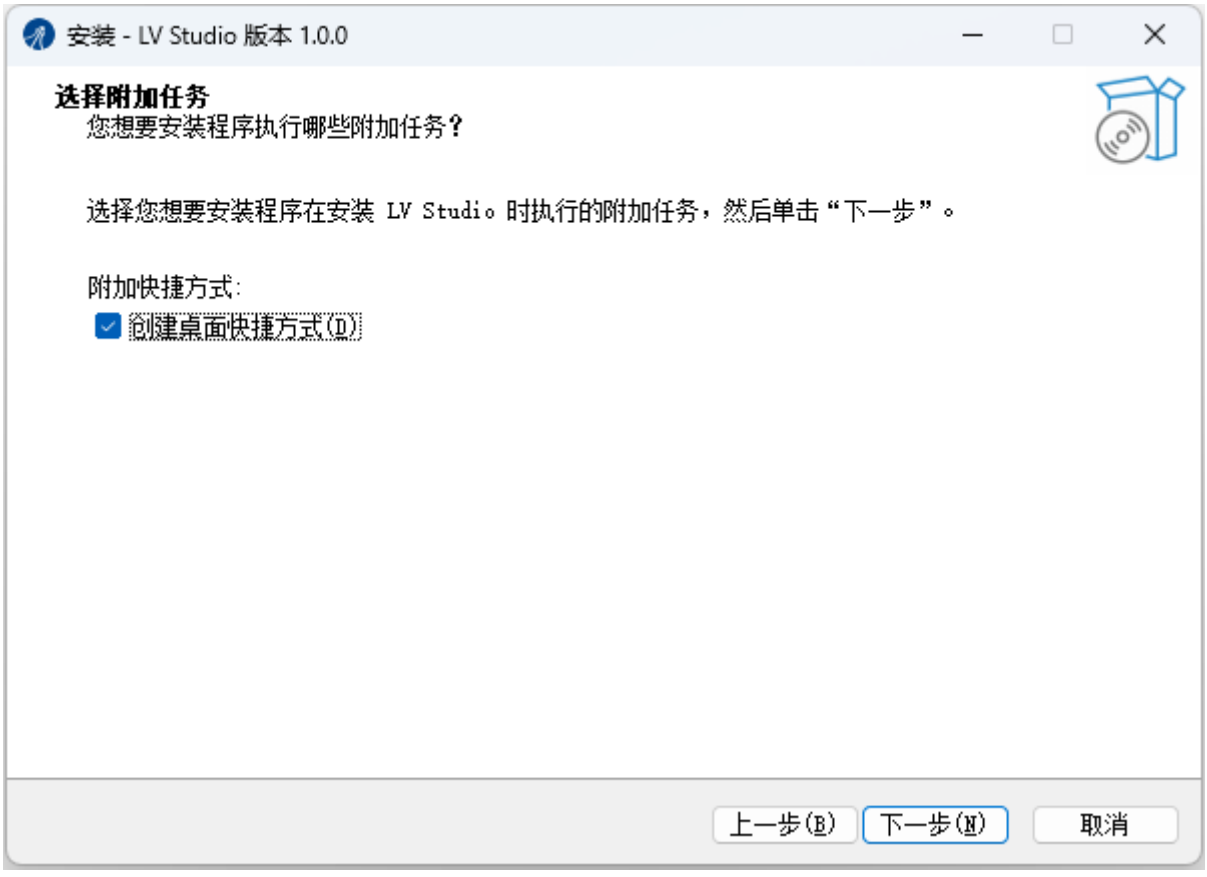
LV Studio 提供 64 位安装包，安装步骤如下：

（1）双击 LV Studio.exe 安装包进行安装。

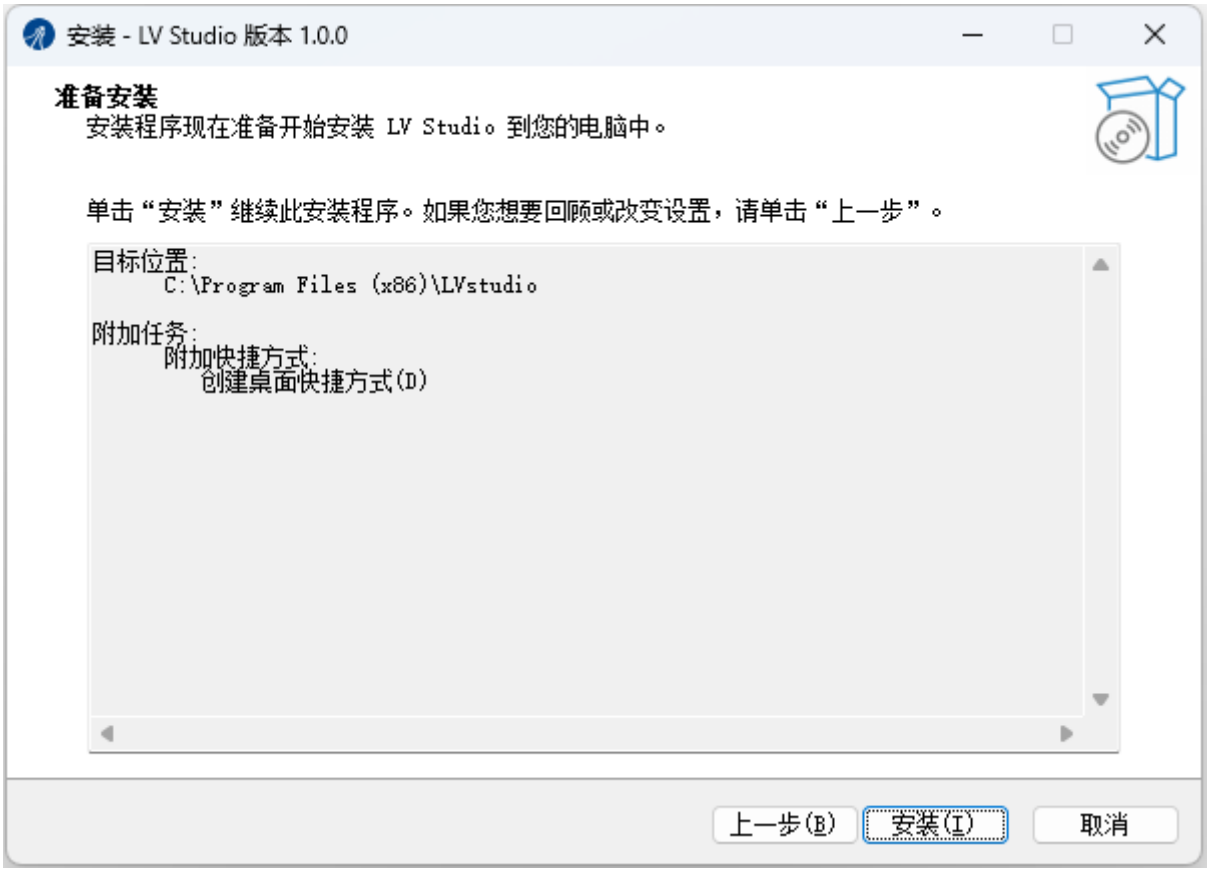
（2）设置软件安装路径。



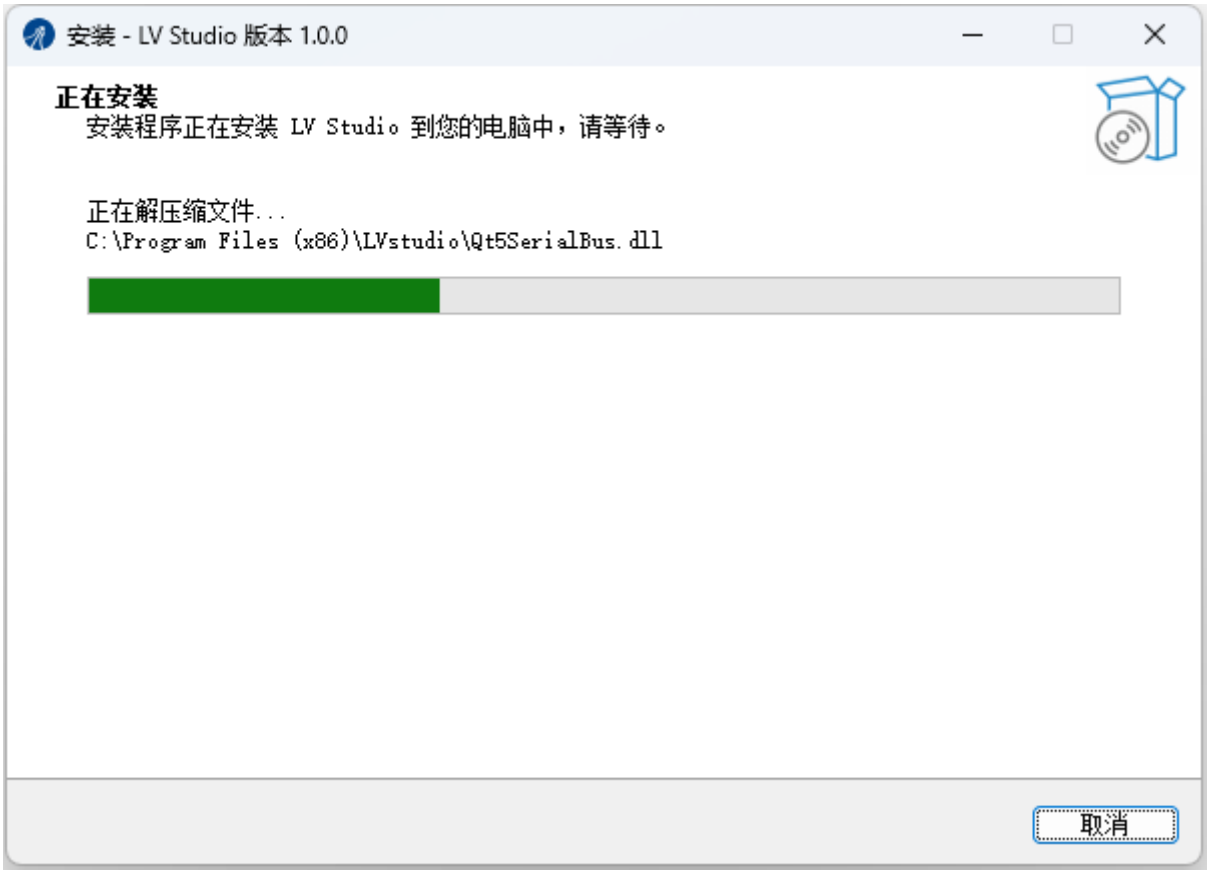
（3）创建快捷方式。



（4）单击“安装”



（5）等待安装完成。



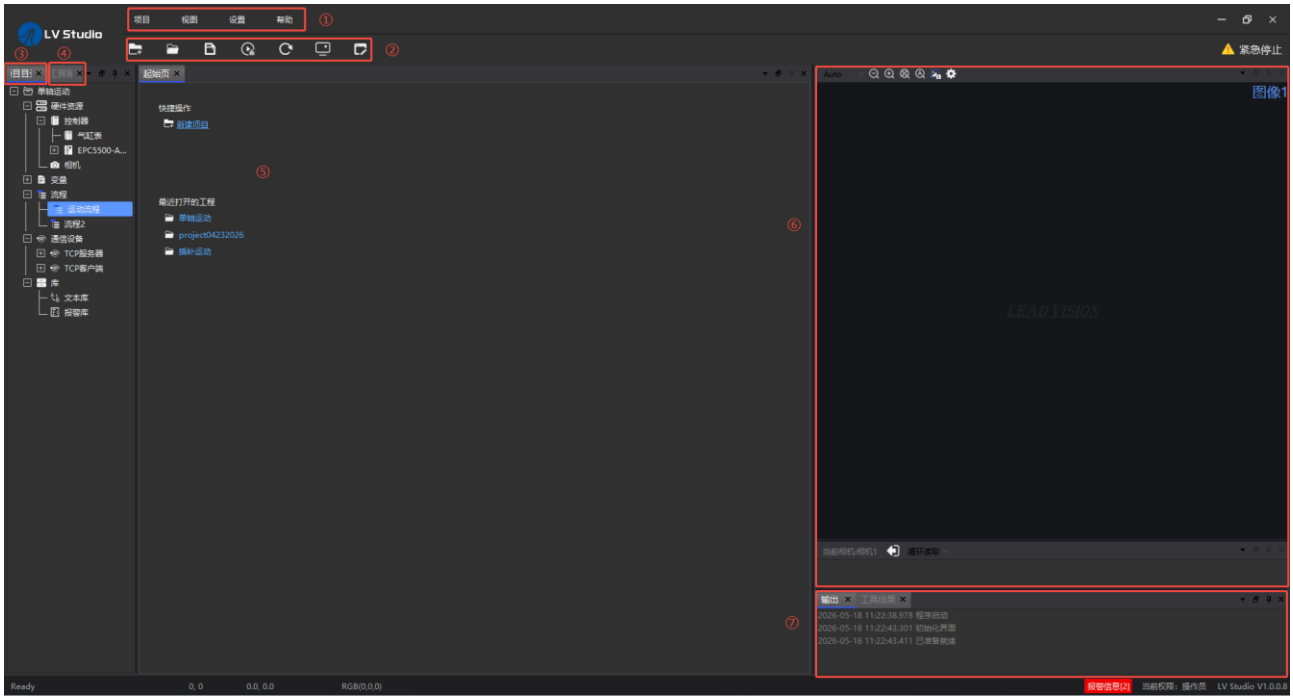
（6）完成安装。


安装 - LV Studio 版本 1.0.0
LV Studio 安装向导完成
安装程序已在您的电脑中安装了 LV Studio。此应用程序可以通过选择安装的快捷方式运行。
单击“完成”退出安装程序。
运行 LV Studio
完成(F)

2. 软件操作

2.1 软件 UI 概述

LV Studio 开发平台主 UI 界面如下图所示，主要包括菜单栏、快捷工具条、项目树、工具箱、起始页、图像显示及结果显示等。



① 菜单栏——项目、视图、设置、帮助。

② 快捷工具条——新建项目、打开项目、保存项目、调试运行、复位、运行模式、编辑模式的快捷操作。

③ 项目树——管理项目的硬件资源、通信设备、流程创建、变量创建。

④ 工具箱——通过列表形式显示所有运动控制工具、视觉算子工具，通过拖拽操作可添加到流程编辑区。

⑤ 起始页——快捷方式创建新项目及显示最近打开的项目

⑥ 图像显示区

a) 图像放大、缩小、原始分辨率显示。
b) 单图像窗口、多图像窗口切换。
c) 显示当前图像的放大倍率、图像分辨率、图像窗口名称。
d) 预览图像文件夹

⑦ 信息显示区

a) 显示工具的输出变量及值。

b) 显示软件运行状态信息。

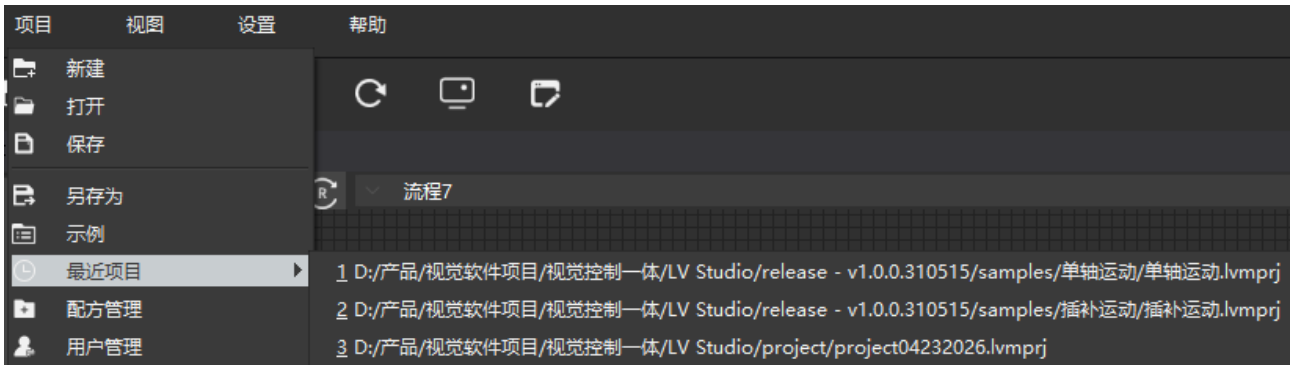
2.2 菜单栏

主界面中最上方显示软件的菜单栏。包含了项目、视图、设置和帮助选项。

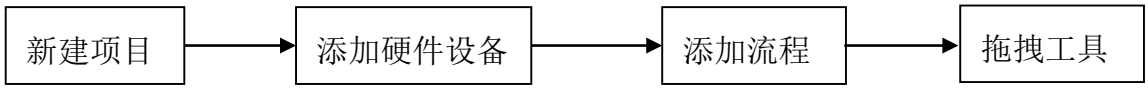


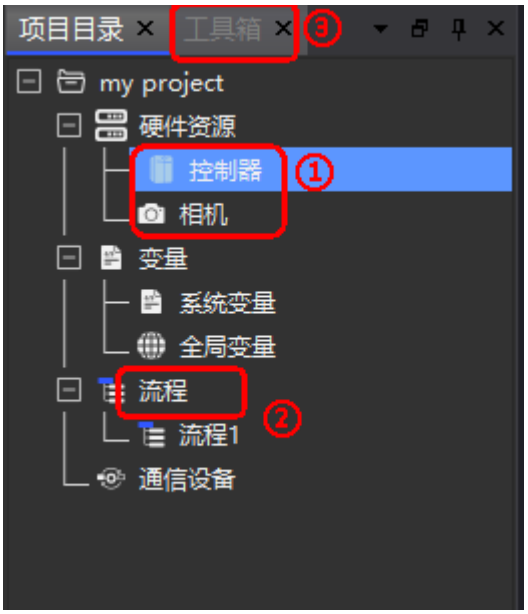
2.2.1 项目

主要用于新建、打开、保存项目，也可用于查看示例项目，配方切换及用户登录等操作选项。



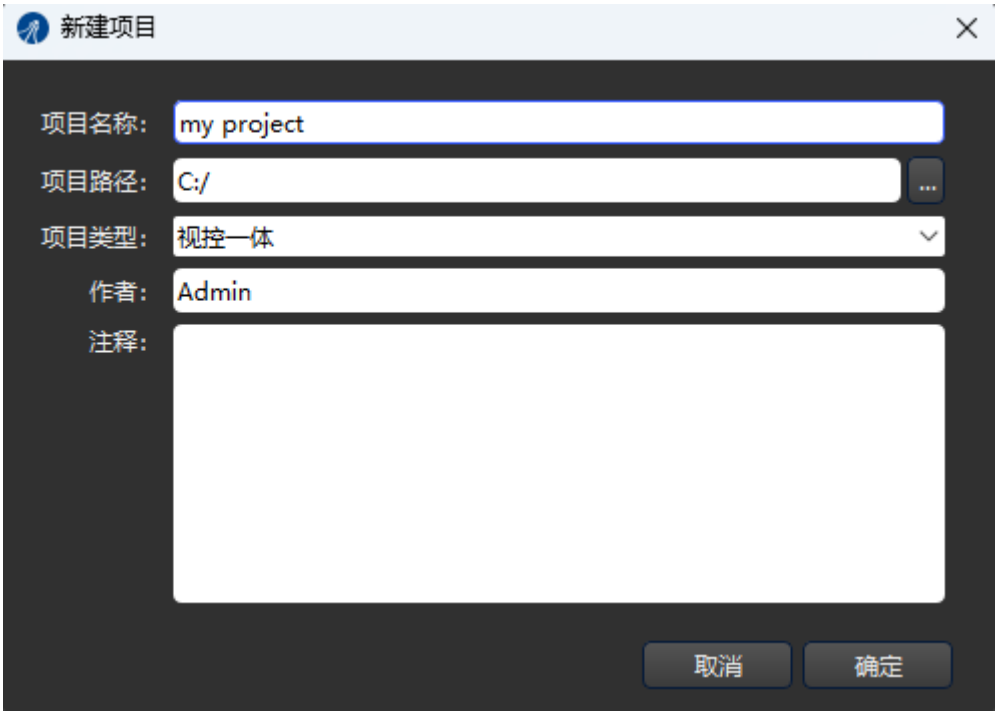
项目搭建一般流程：新建项目->选择【项目目录】树的【控制器】和【相机】，右键添加控制器及相机等硬件设备->选择【项目目录】树的【流程】，右键添加任务流程->最后在任务流程中根据应用需要，从【工具箱】中左键拖拽添加对应的工具块，完成流程搭建。





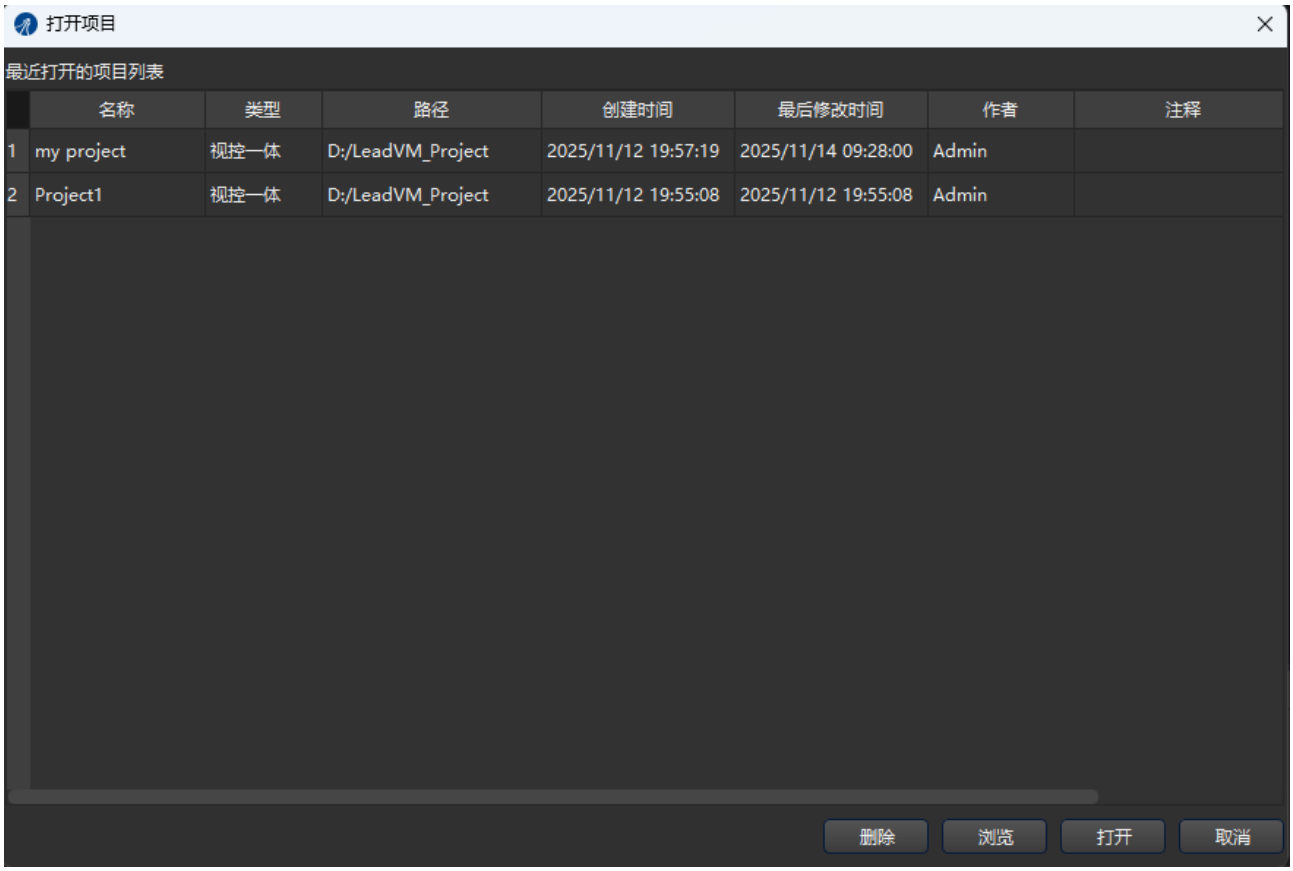
2.2.1.1 新建

可创建新项目。创建新项目时，需设置当前项目名称、项目类型及保存路径。项目类型分为视控一体项目、纯运动控制项目和纯视觉项目三种类型。



2.2.1.2 打开

加载之前保存的项目，最近打开的项目以列表形式显示详细信息。选中列表中的项目，点击【打开】按钮，即可加载所选项目；用户也可通过【浏览】按钮选择项目路径打开。

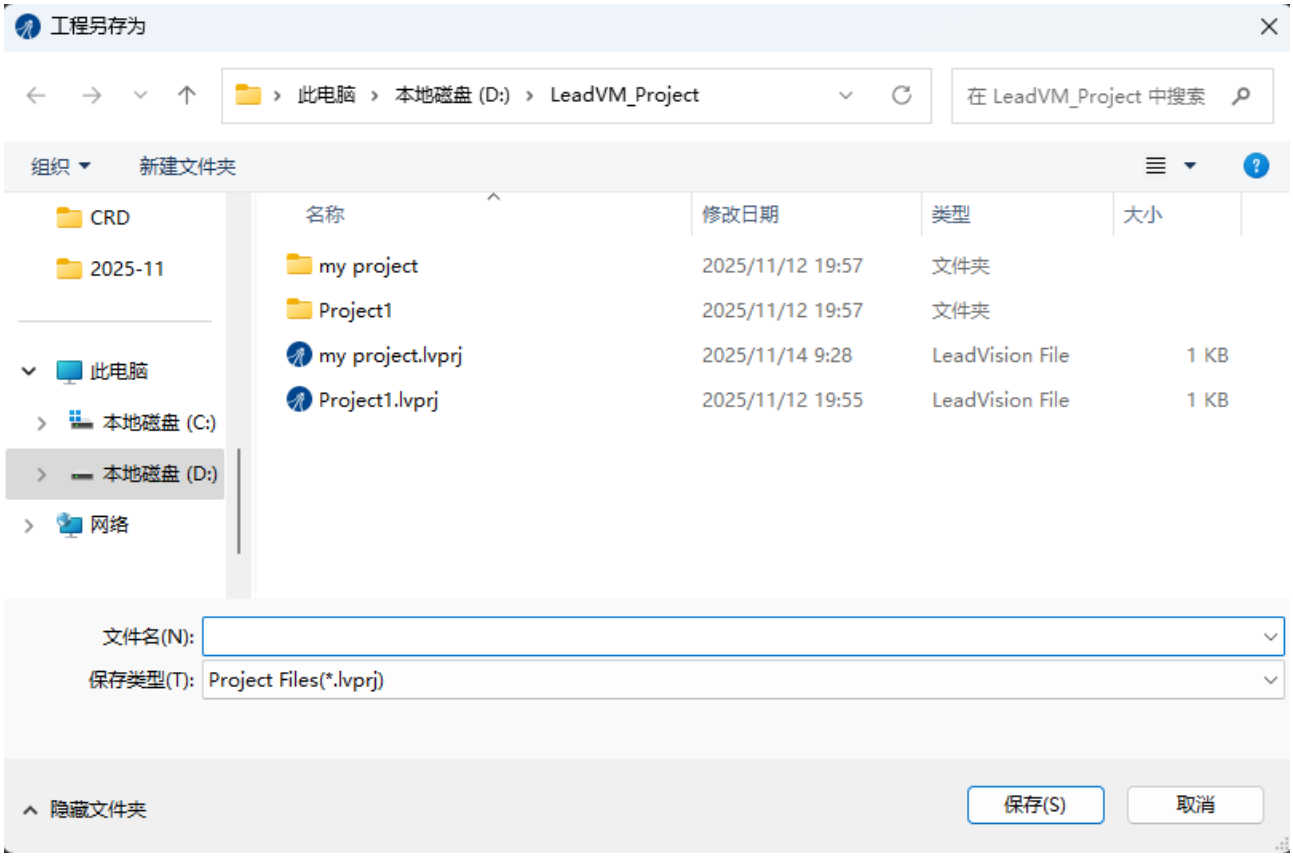


2.2.1.3 保存

保存当前项目到创建项目的路径下。

2.2.1.4 另存为

将当前项目另存为到指定的路径。



2.2.1.5 示例

加载示例工程。

2.2.1.5 最近项目

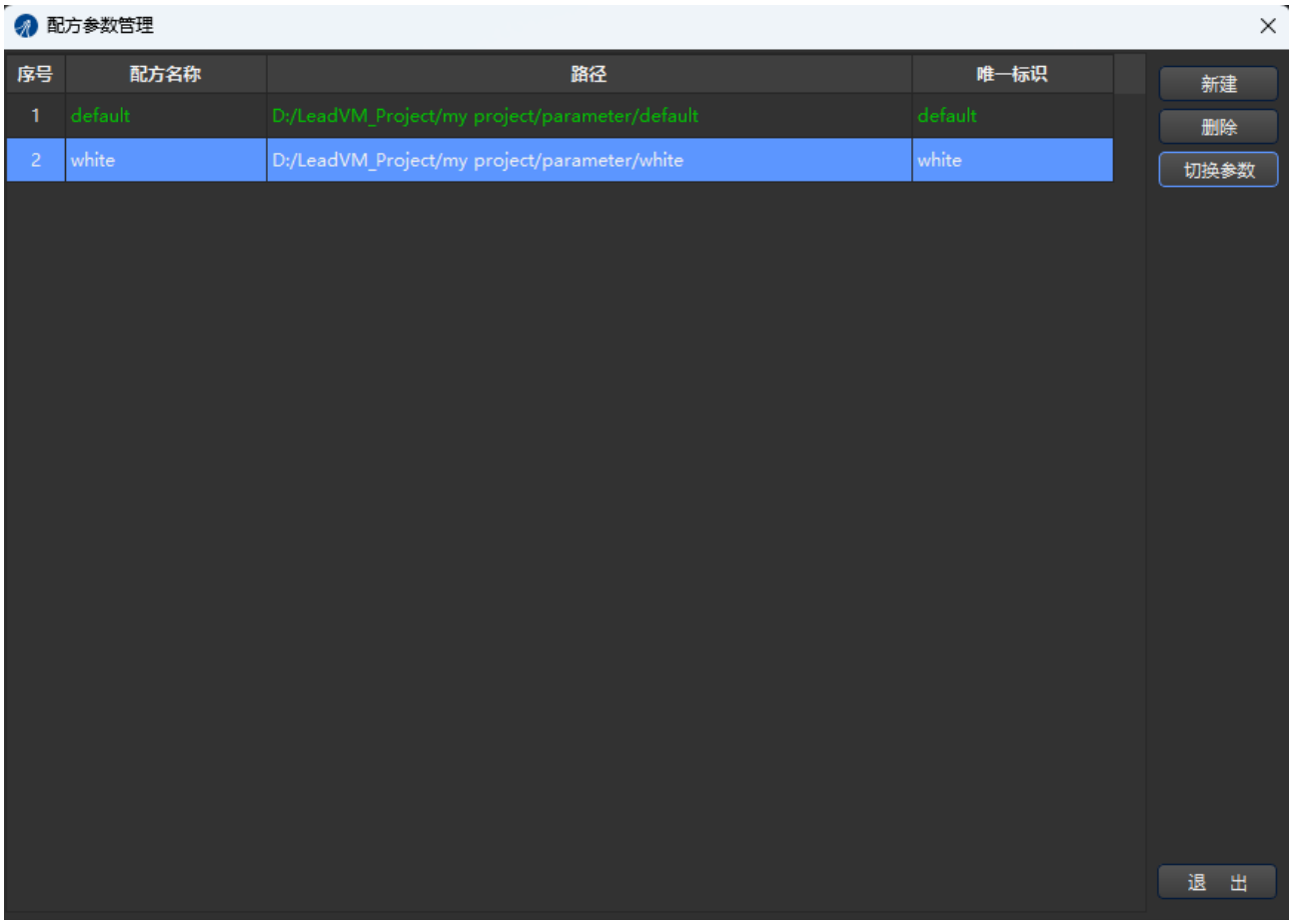
可访问最近创建的 6 个项目，点击选中项目即可打开该项目。



2.2.1.6 配方管理

同一项目中的工具块可能需要设置不同参数，来定位或检测同一款但颜色不同的产品，此时可通过配方实现，项目中的流程完全相同，只是运行时工具块的参数不同。

配方管理用于新建、删除、切换配方。项目中默认使用 default 配方，用户可以修改该配方名称，也可基于现有配方新建一组配方，或切换不同配方。



2.2.1.7 用户管理

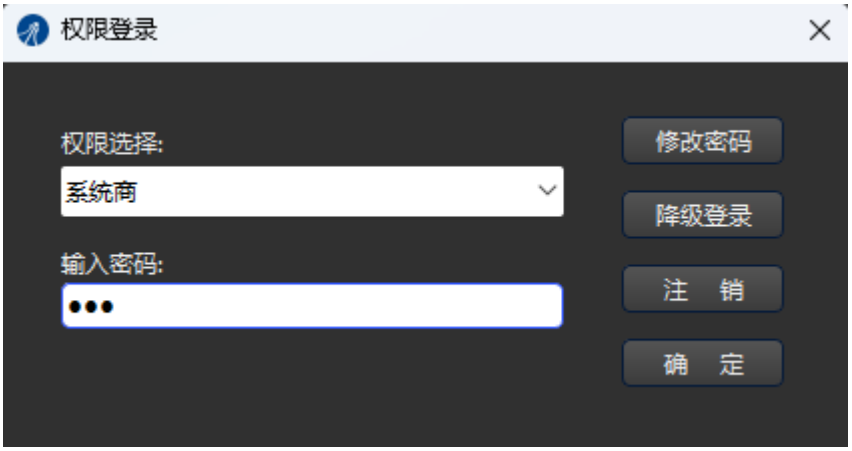
LV Studio 开发平台处于编辑流程的模式下，所有操作不设权限，当进入到运行模式后，再切回编辑模式时，需要相应权限，该权限在菜单栏【设置】->【系统设置】中配置。用户界面组态的每个控件，都可单独配置访问权限。

验证登录权限的密码进行登录确定，权限等级分为三级权限，可逐级授权每个层级的使用权限。初始状态为操作员权限。

系统商(Developer)——最高权限，可以获得软件的所有操作权限，默认密码为 123。

技术员(Technician)——中等权限，默认没有密码，首次登录输入的密码即为登录密码，可以访问设置为技术员和操作员权限的控件。

操作员(Operator)——最低权限，没有密码。



2.2.1.8 退出

退出 LV Studio 软件。

2.2.2 视图

调整 LV Studio 开发平台的布局分布，勾选状态的窗口会显示到主窗口区，调整完成后可保存当前视图，后续打开 LV Studio 开发平台时，将显示该调整后的视图。如果用户想恢复到初始状态，请点击恢复默认视图。



2.2.3 设置

用于配置软件开机自启动、流程模式及图标更换。



- 系统设置

  可进行开机启动设置、项目加载设置及其他设置。



开机软件自启动：该选项启用后，可设置 PC 开机后 LV Studio 开发平台自动打开。

默认启动运行界面：该选项启用后，当打开 LV Studio 开发平台时，自动切换到运行界面。

自动加载指定项目：该选项启用后，当打开 LV Studio 开发平台时，将根据设置自动加载该项目。

自动加载最近项目：该选项启用后，当打开 LV Studio 开发平台时，将自动加载最后一次操作的项目。

启动状态：配置加载项目后，是否自动执行。

运行界面切换到编辑界面权限：可配置切换至编辑界面的操作权限，权限包括操作员、技术员和系统商，默认操作员。

最大撤销次数：15 次。

最大允许打开窗口数：30 次。

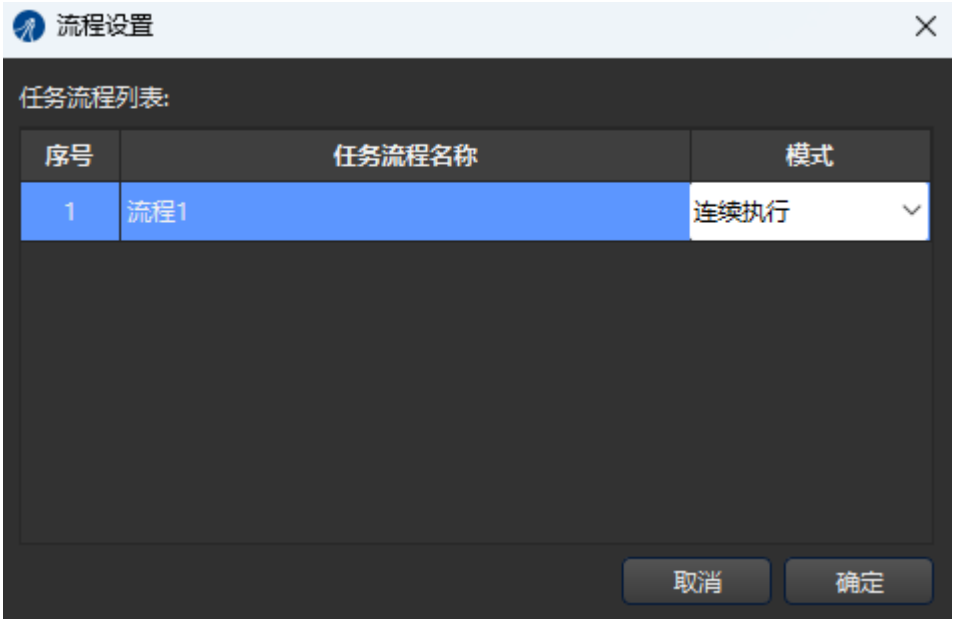
- 流程设置

配置每个流程任务的执行模式，包括连续执行、触发执行和自启动执行。

连续执行：流程启动后循环执行。

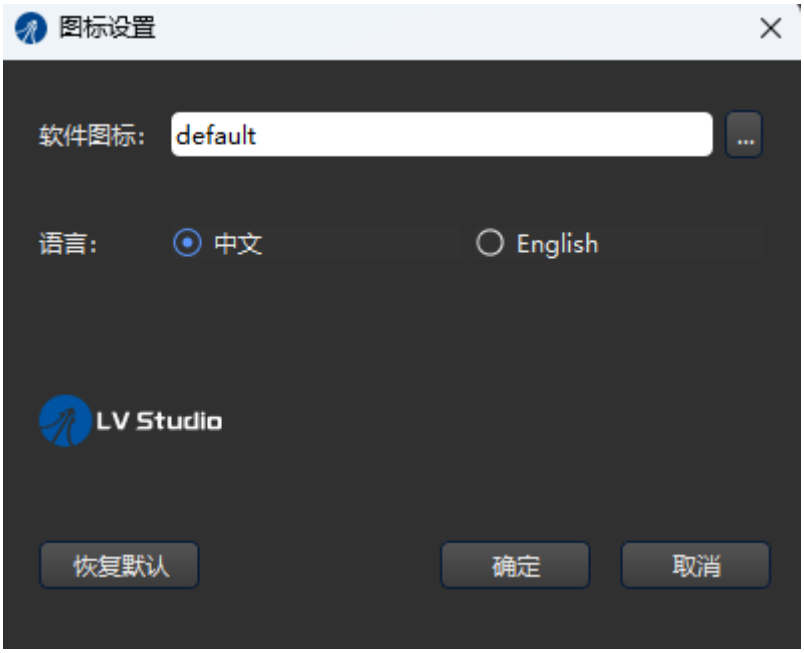
触发执行：流程被触发后执行一次。

自启动执行：流程进入用户界面后自动执行，可用于监控 IO 状态、轴状态等信息。



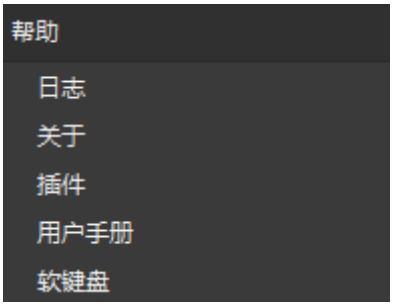
- 图标设置

可修改 LV Studio 开发平台的显示图标，用户可按需更换为本公司的 logo。



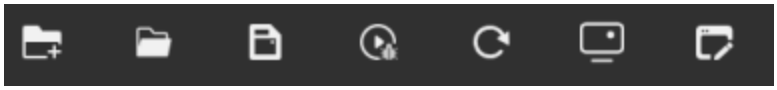
2.2.4 帮助

查看平台运行日志、版本信息、加载的插件信息、使用说明手册及软键盘调用。



2.3 快捷工具条

快捷工具条位于菜单栏下方，可以快速方便地对平台进行相应的操作。



（1）新建项目

单击后进入新建项目窗口。

（2）打开项目

单击后进入打开项目窗口，可浏览或打开指定项目。

（3）保存项目

单击后保存当前项目到创建项目的路径下。

（4）调试运行

在编辑模式下连续运行整个项目流程。

（5）复位所有流程

单击后复位整个平台的运行。

（6）运行模式

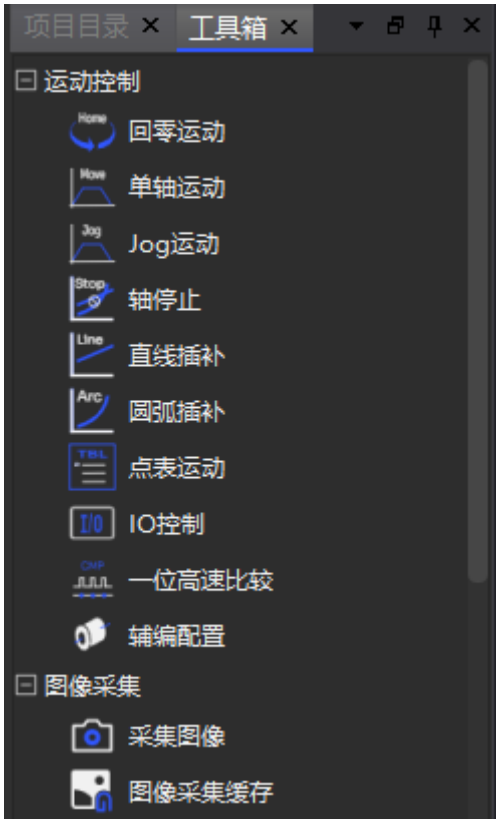
只切换到运行界面，不会运行项目。运行界面由用户界面组态，默认为空界面。

（7）编辑模式

单击后进入用户界面，在此组态运行界面。

2.4 工具箱

2.4.1 工具简介



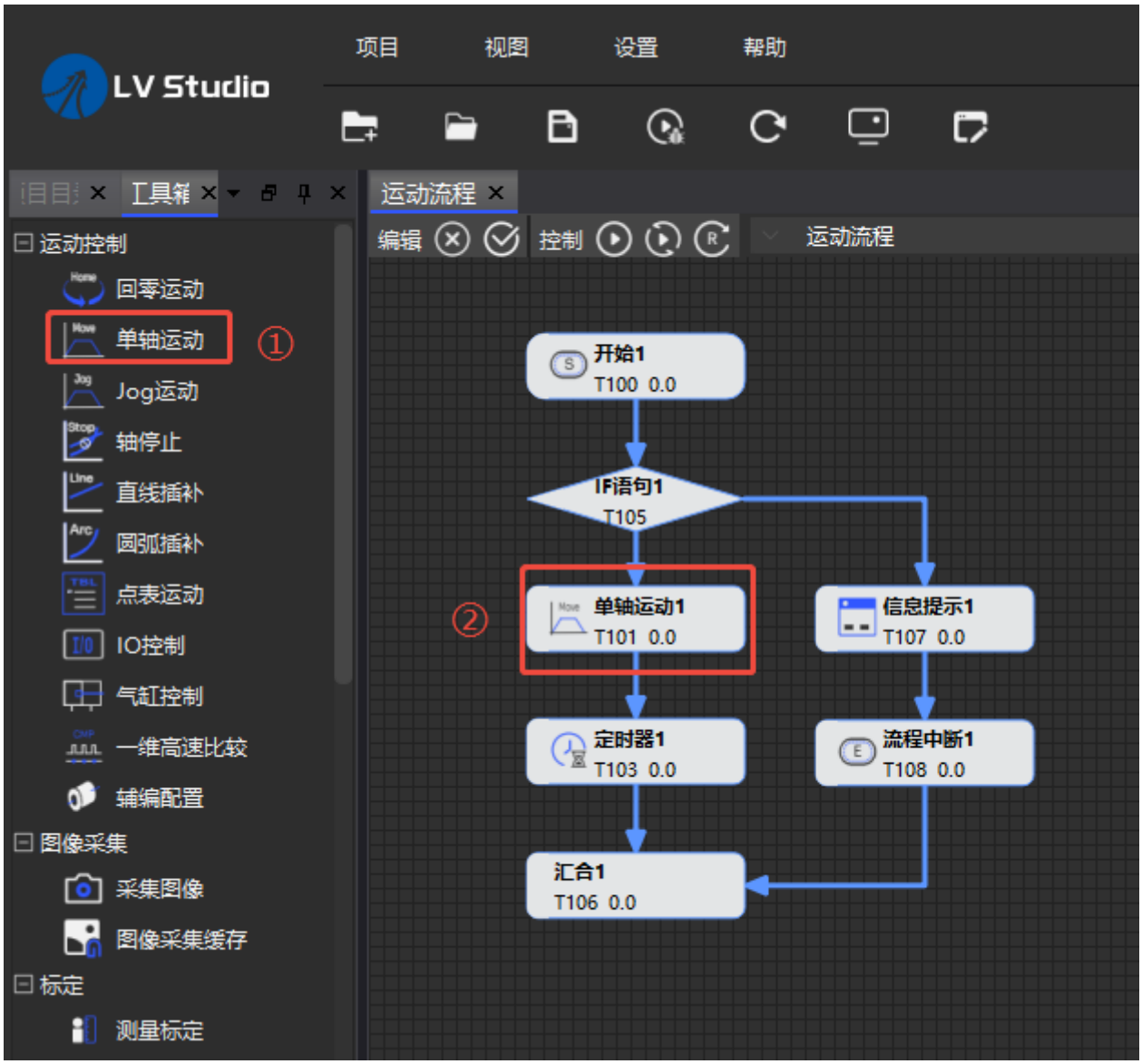
工具箱的工具主要包括运动控制、图像采集、标定、定位、测量、识别、图像处理、颜色处理、系统工具、逻辑控制和通信等模块。

- 运动控制：回零、单轴、Jog、插补等运动功能。

- 图像采集：包括从相机中采集图像和缓存图像。
- 标定：N 点标定、测量标定等。
- 定位：模板匹配等特征提取工具，标识特征点用于定位。
- 测量：测量距离、夹角等基本测量功能。
- 识别：包含一维码、二维码识别等。
- 图像处理：图像预处理类算子工具，如膨胀、腐蚀等形态学处理。
- 系统工具：变量计算、变量读写、脚本计算等。
- 颜色处理：彩色图像的颜色提取、颜色识别、颜色测量等。
- 逻辑控制：视觉流程数据的逻辑处理。
- 通信：常用的工业通信协议，包括 TCP/IP、串口两种通信方式。

2.4.2 工具操作

鼠标停留在工具箱导航树，单击选中要使用的工具，如①“单轴运动”拖拉至②流程编辑区域，按照项目逻辑需求合理放置工具先后顺序并连线，双击配置参数即可，流程执行顺序为从上至下依次执行，如下图所示。

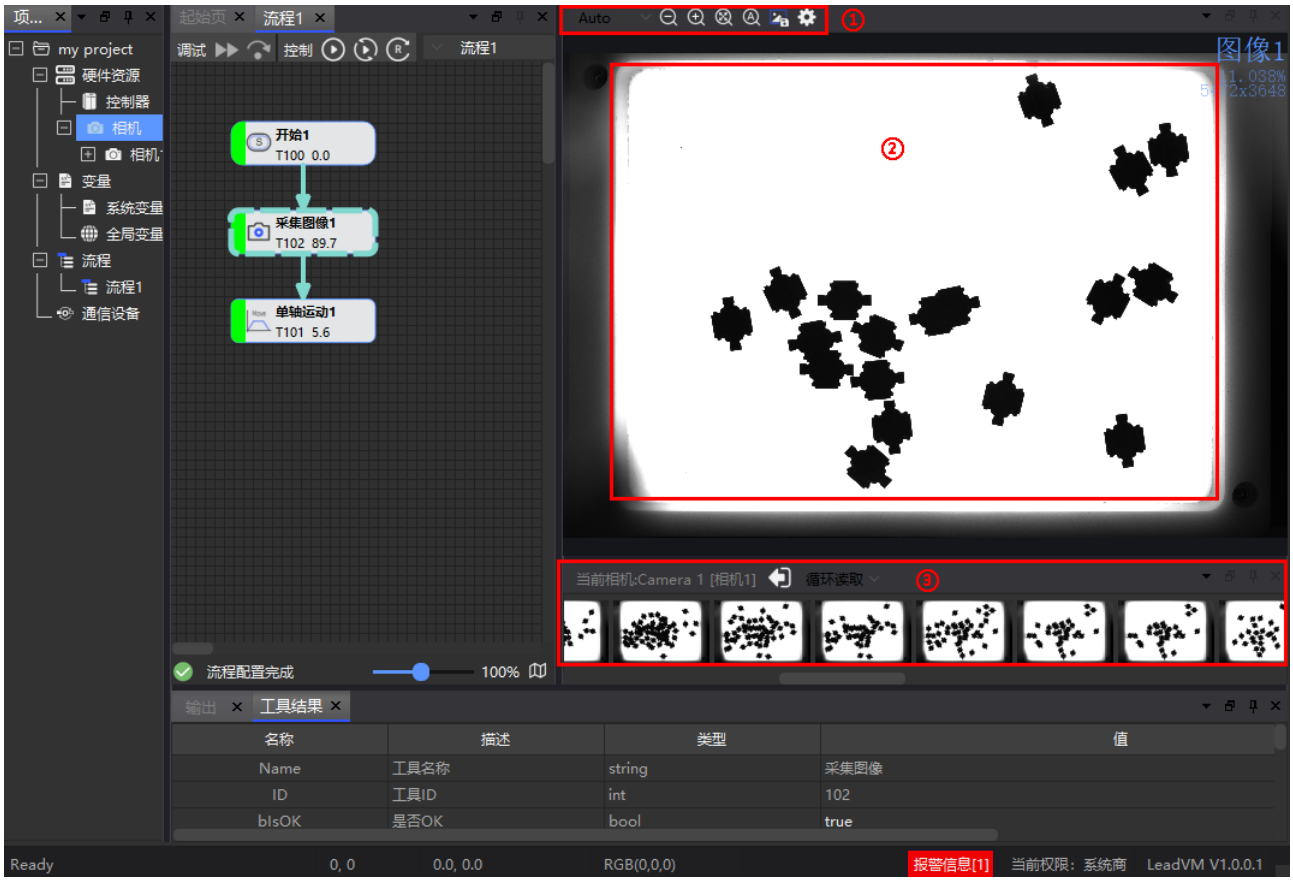


2.5 结果显示

编辑模式下通过主界面的图像显示区和信息显示区，可查看图像结果、工具结果和开发平台的操作输出信息。

2.5.1 图像显示区

显示流程中当前选中工具的图像信息及配置窗口数量。



①工具栏：可对预览窗口的图像进行简单操作，包括缩小、放大、自适应、所有图像窗口自适应、保存图像、图像窗口设置。

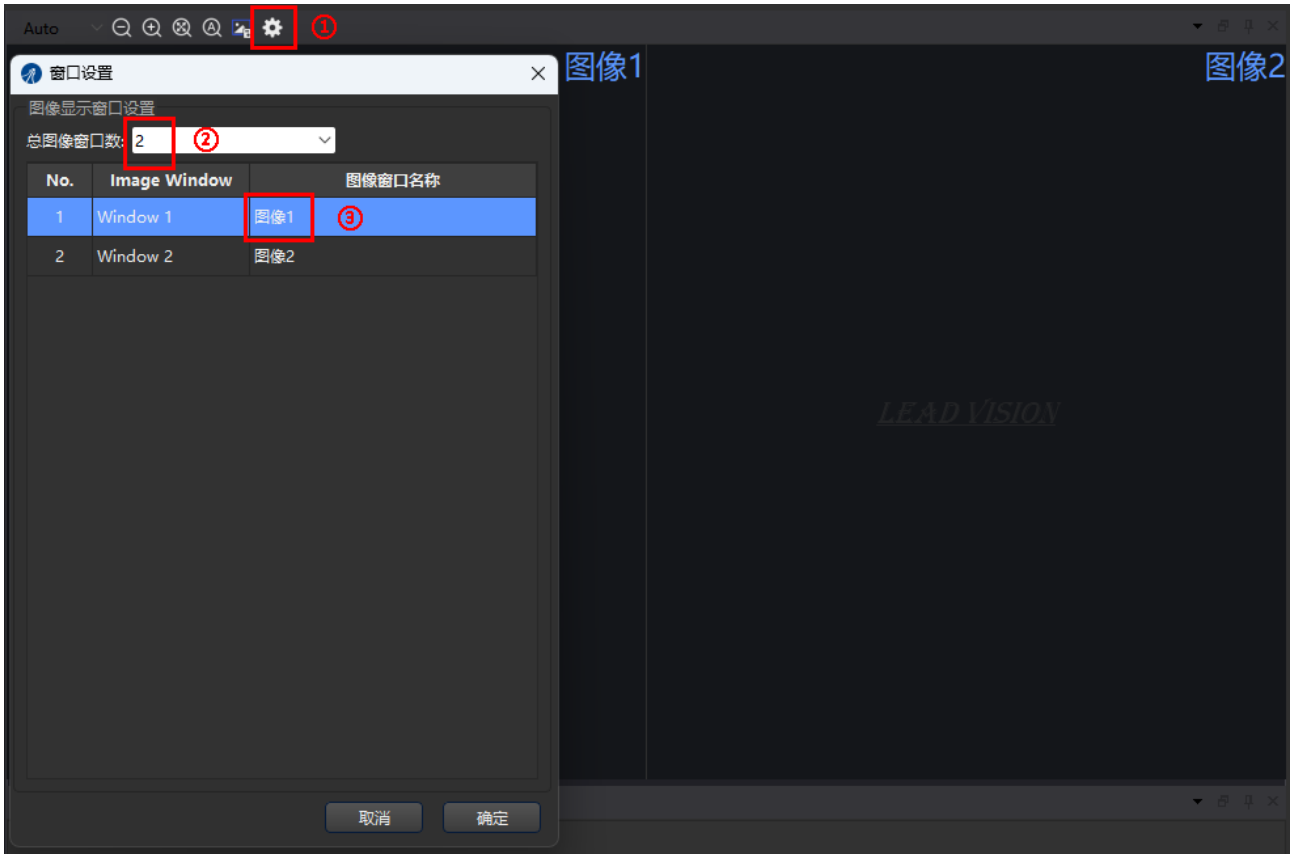
②图像显示：显示流程中当前选中工具的图像。

③图像列表：当流程中的图像源为本地图像时，可通过该区域预览本地加载的图像并进行相关设置。

- 返回到第一张：复位图像到第一张。
- 循环读取：循环切换当前已加载的图像。
- 遍历一次：仅运行一遍已加载的图像
- 单张循环：仅运行勾选的图像。

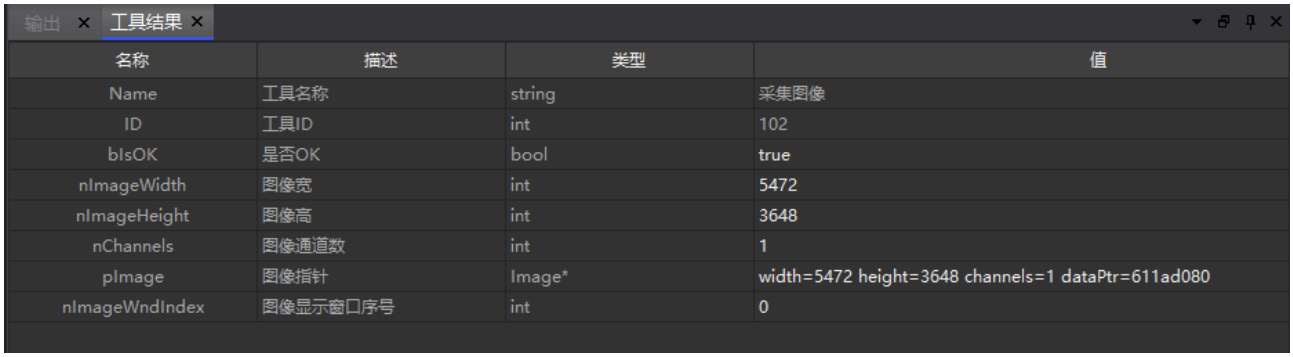
图像窗口设置

如果需要多个图像显示窗口，可通过点击“图像窗口设置按钮”，在弹窗中设置窗口数量及窗口名称，如下图所示。



2.5.2 工具结果显示区

工具的当前输出结果，便于用户分析查看。不同工具的输出结果有所差别，具体结果数据信息以各工具实际显示为准。



3. 项目搭建-硬件资源配置

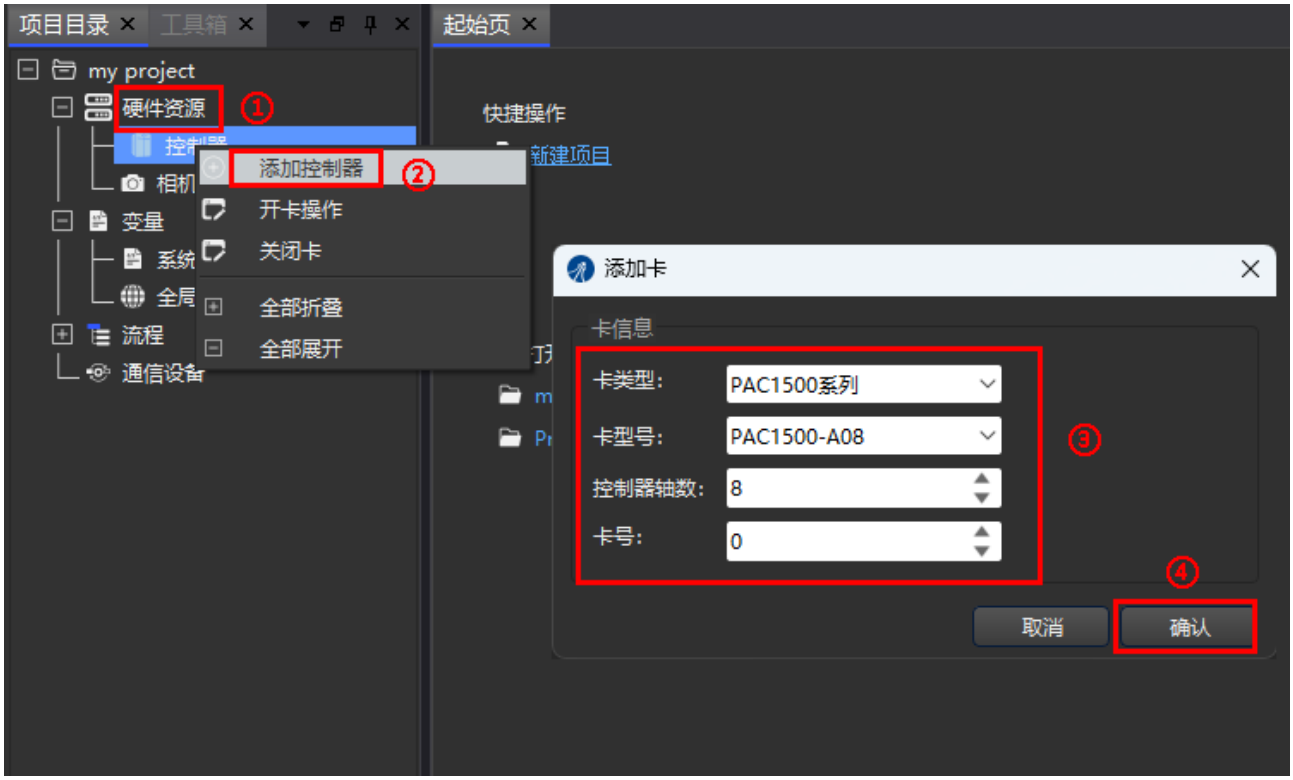
在视觉控制一体化项目中，用户需添加相机、控制器、模块等硬件资源，并对其进行相应配置，然后在流程中进行关联，最后在用户界面进行组态设计以输出运行界面，从而完成一个视控一体项目的构建。

3.1 控制器

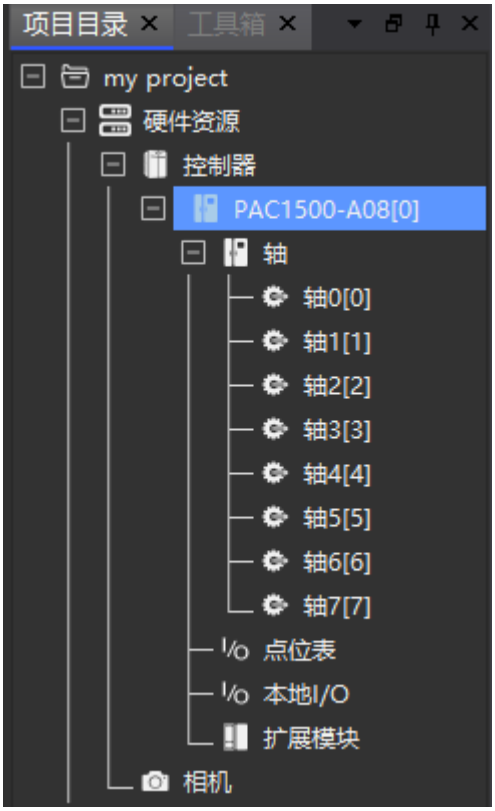
3.1.1 添加控制器

LV Studio 可添加 PAC 系列、EPC 系列和 DMC-E 系列控制器，作为全局资源在整个项目中访问。

双击项目目录树中的①“硬件资源”，展开“硬件资源”项目树下的子项，右击“控制器”，选择②“添加控制器”，在弹出的③“添加卡”对话框中，选择卡类型、型号，输入卡号（根据实际卡号拨码输入），最后点击④“确认”，完成控制器添加，如下图所示。

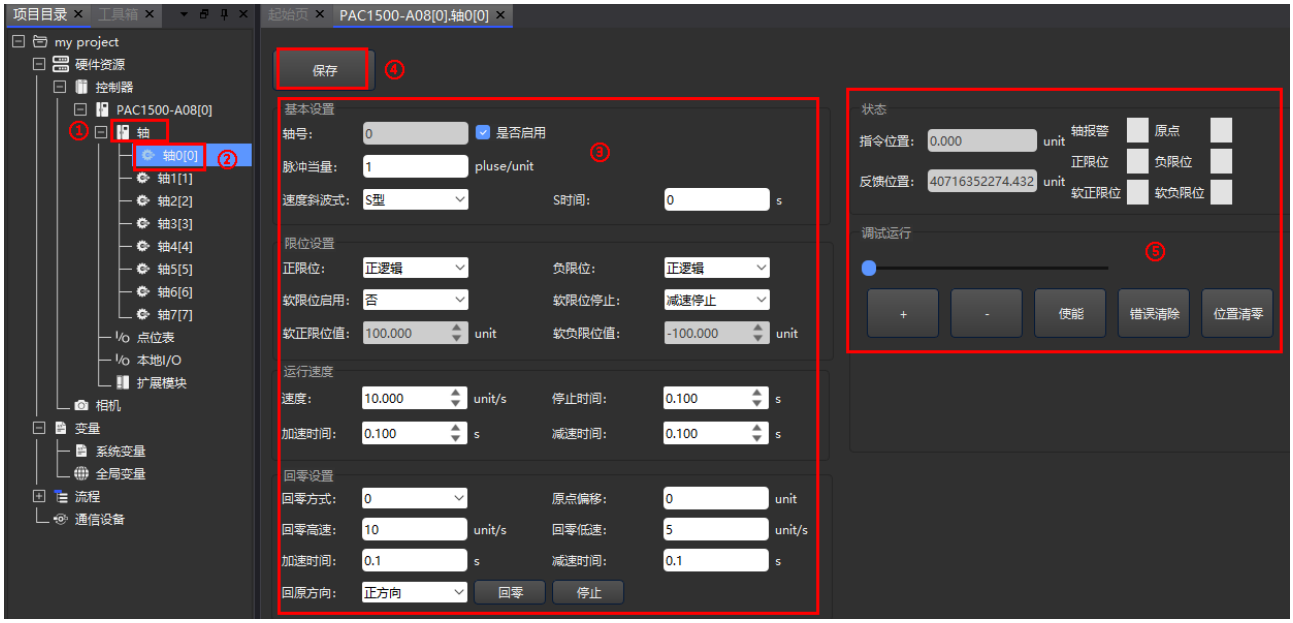


控制器添加完成后，如下图所以。下一步需根据实际设备硬件结构，对轴、IO 表及扩展模块进行配置。如果需要使用点位表运动功能，需配合工具箱的点位表运动工具块一起使用。



3.1.2 配置轴

双击项目目录树中的①“轴”，展开轴列表子项，可对每个轴参数进行配置。以轴 0 为例，其他轴参数配置方式与其相同。双击②“轴 0[0]”，弹出③轴 0 配置窗口，可配置轴回零、脉冲当量、限位、运行速度等参数，按设备实际状况配置轴参数，配置完成后，点击④“保存”参数，如下图所示。配置完成所有轴参数后，后续可以通过项目目录树中的“控制器”，右击选择“开卡操作”，打开卡成功后，对轴进行⑤使能、Jog 运动测试。



轴参数说明如下：

保存：将参数设置到控制器。

轴号：显示轴号，不可设置，从 0 开始自动排序，为 API 函数的轴号，用户无需关注。

是否启用：勾选后，启用该轴，控制器开卡后，会使能该轴。

脉冲当量：每个用户单位的脉冲数，如电机每转 10000pulse，丝杆螺距 10mm，减速比 1：1，则此处设置 10000/10=1000pulse/mm。

速度斜波式：设置轴速度曲线为 T 型和 S 型，选择 S 型后，需配置 S 时间，单位秒。

正负限位：正硬限位、负硬限位的有效电平，正逻辑和负逻辑。

软限位：软限位启用是/否、正限位值设置（用户单位）、负限位值设置（用户单位）、停止模式包括减速停止和立即停止。

运行参数：单轴运动工具块的速度、加减速时间参数设置为-1 时，使用该处设置的运行速度、加速时间、减速时间。Jog 调试按钮的速度比按该处速度值乘以百分比计算，停止时间设置的是限位信号被触发时的停止时间。

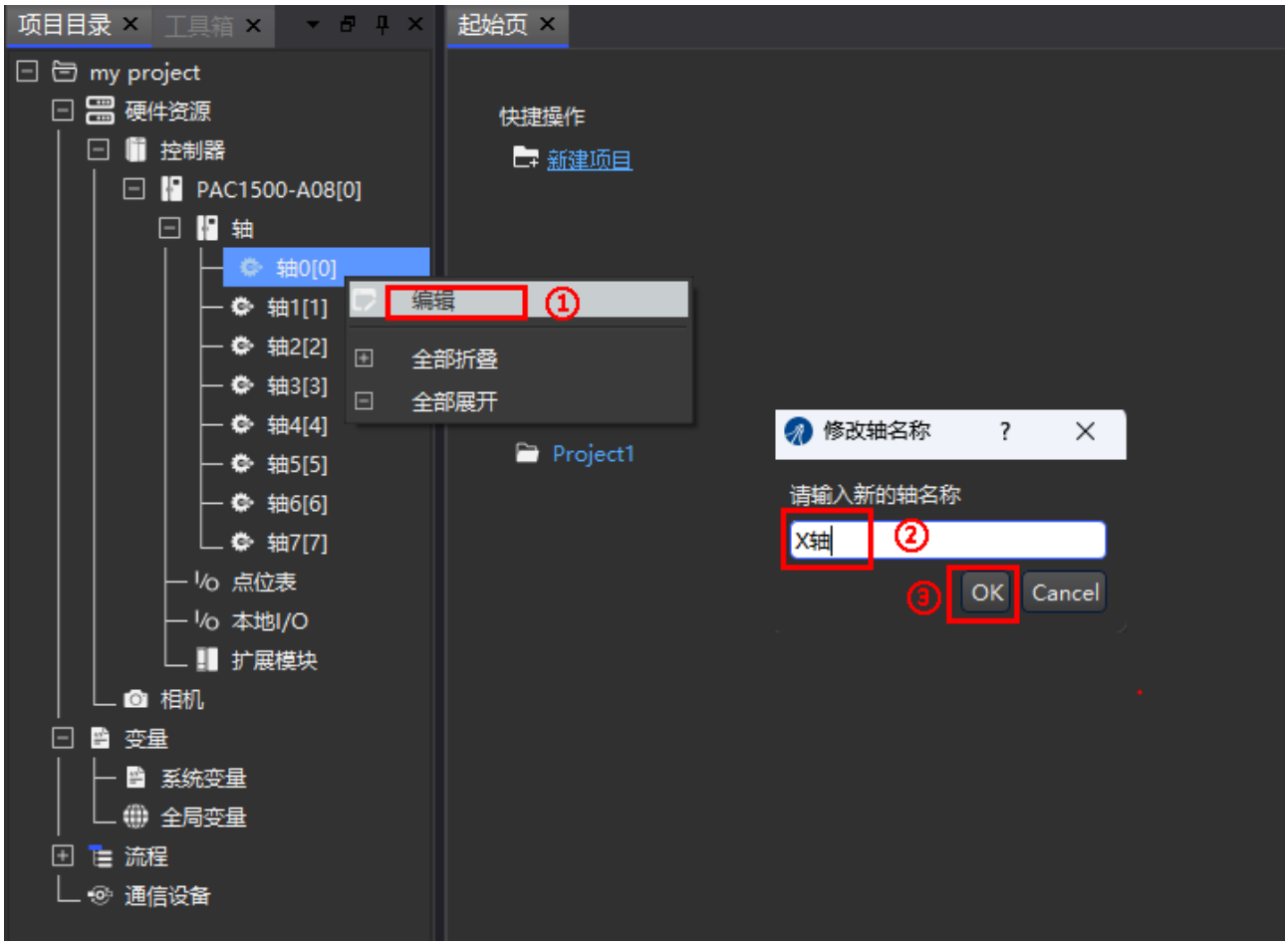
回零参数：回零模式（详见控制器手册）、高速（高速找零时的速度）、低速（返回原点时的速度）、回零偏移（回零完成后的偏移距离）。

调试按钮：Jog+运动、Jog-运动、使能/失能轴、轴错误清除、轴位置清零。

轴状态监控：指令位置、反馈位置、限位、原点、轴报警。其中限位/原点/轴报警绿色表示正常，红色表示异常。

说明：工具箱中的轴运动相关工具块，如果轴运行参数及回零参数设置为-1 时，即使用该界面的运动参数。

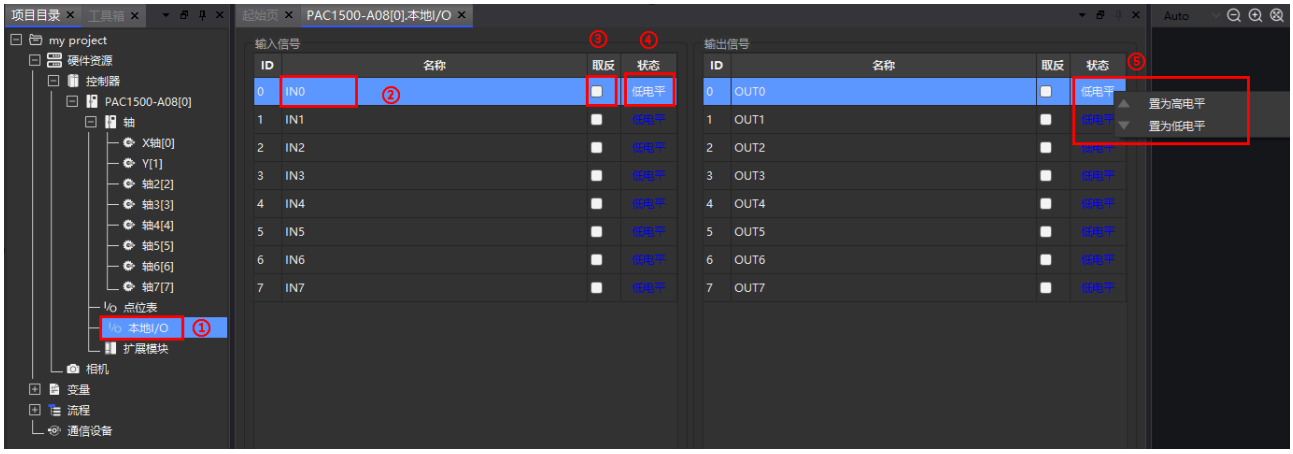
轴名称修改：右击轴，选择①“编辑”，在弹出的编辑对话框中，输入②新名称，如 X 轴，点击③“OK”，完成轴名称的修改。轴名称修改是为了方便用户管理实际设备。



3.1.3 配置本地 IO

本地 IO 为控制器本体 IO，可修改 IO 名称、监控 IO 口的状态，也配置 IO 口的电平取反及操作输出口输出，方便用户调试使用。

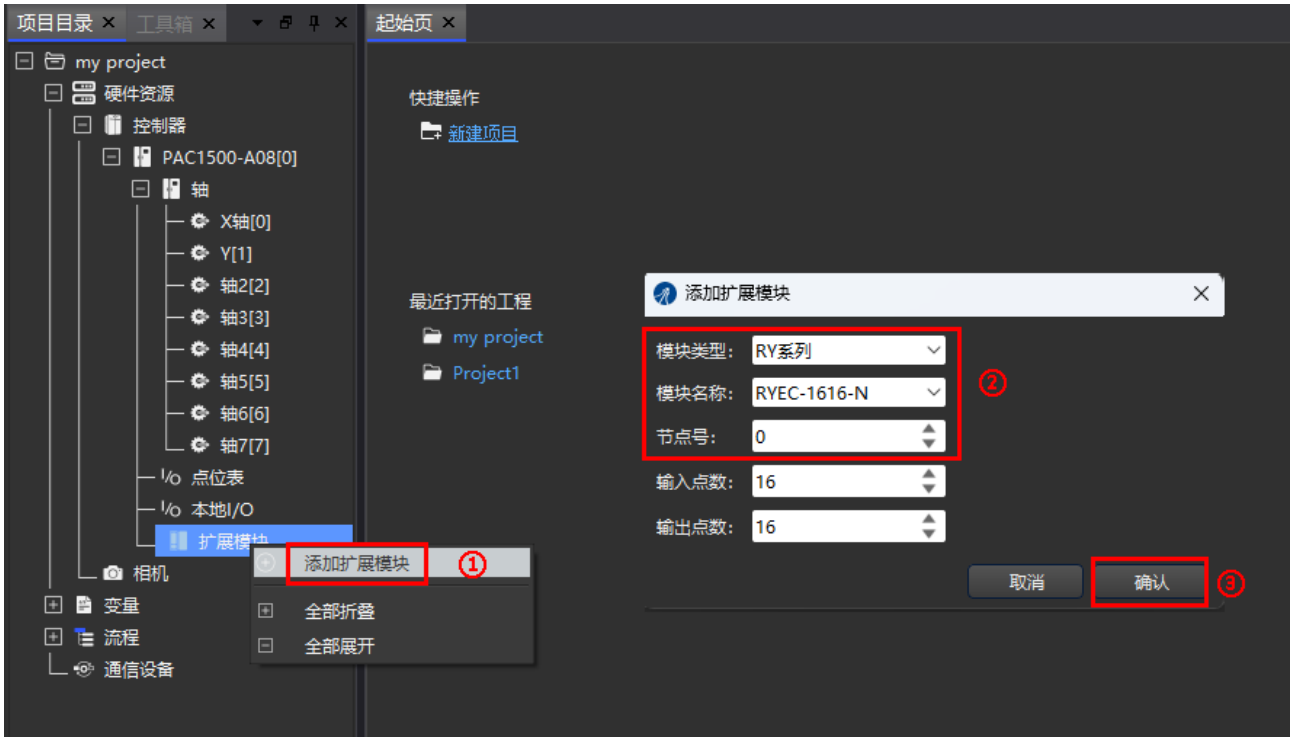
配置方法：双击①“本地 I/O”，打开本地 I/O 窗口。如果需要修改 I/O 口名称，双击②IN0 名称，可直接修改输入名称，输出名称修改方法与之相同。如果 I/O 口高低有限电平需要取反，勾选③“取反”。I/O 口的电平状态可通过④“状态”查看，输出口还可右击当前状态⑤“低电平”，置 ON 或置 OFF 输出口。



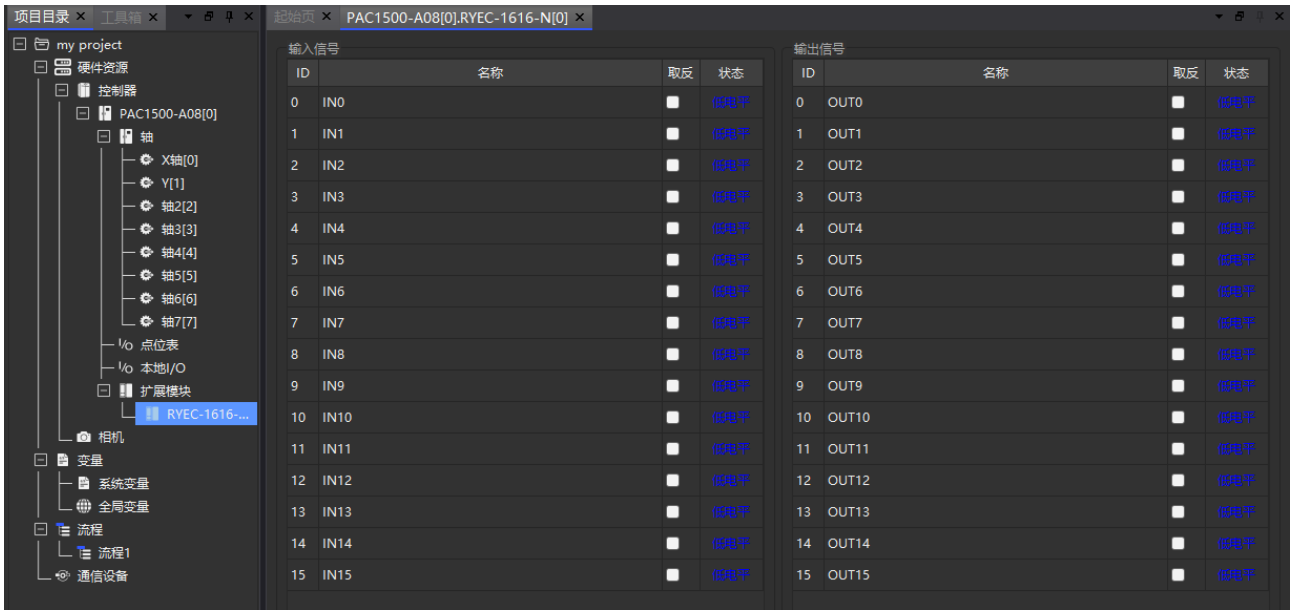
3.1.4 配置扩展模块

当控制器本体资源不够时，用户需要扩展模块，按实际硬件组态添加对应的扩展模块。扩展模块支持扩展 EtherCAT 总线模块，当前支持扩展数字量 I/O 模块，其他模块类型，暂时不支持。

添加扩展模块：右击项目目录树中的扩展模块，选择①“添加扩展模块”，在弹出的“添加扩展模块”对话框中，选择②模块类型，模块名称，输入节点号（第一个节点为 0，依次类推），输入点数和输出点数默认即可，最后点击③“确认”，完成模块添加。



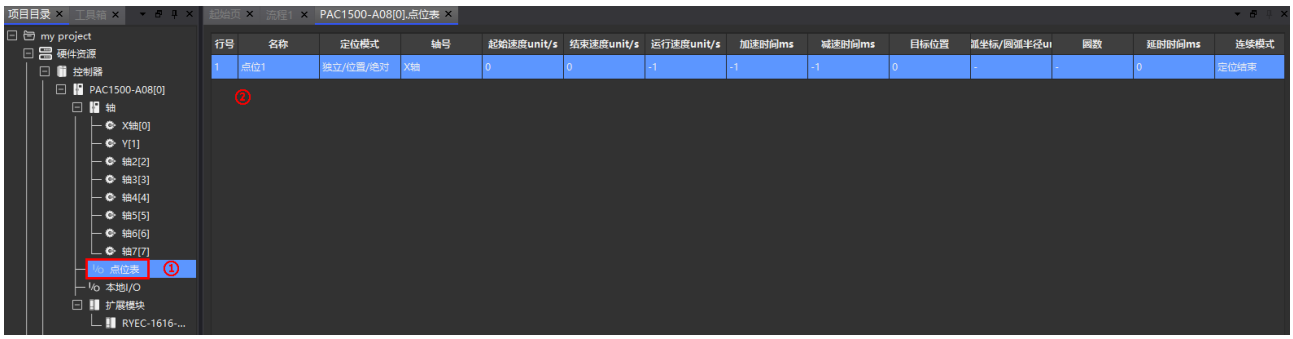
I/O 模块添加完成后，如下图所示。I/O 扩展模块的配置同章节 3.1.1.2 配置本地 IO。



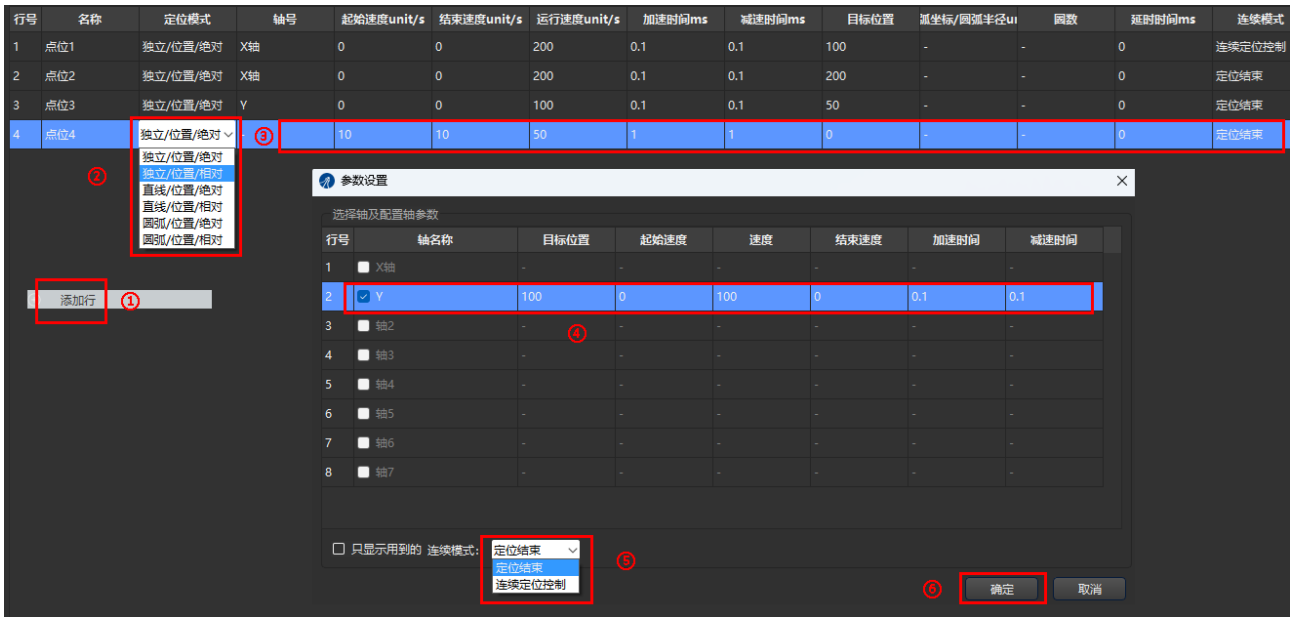
3.1.5 配置点位表

点位表为可选配置，常用于简单重复的运动形式，使用时需配合点表运动工具块，选择的点表运动行启动后，直至遇到连续模式为“定位结束”的行，该点表运动结束。可添加多行，如果每一行的连续模式设为“定位结束”，则为多个独立的定位运动，需多次调用点表运动工具块；如果需要连续运行多行数据，需将连续模式设为“连续定位控制”，最后一行的连续模式设为“定位结束”。

双击“项目目录”树中①“点位表”，打开②点位表配置窗口，如下图所示。



添加点位数据步骤：在点位表窗口任意位置右击，选择①“添加行”新增一行，在②定位模式下拉项中，根据实际动作需求，选择位置/直线/圆弧插补模式，双击③框中的任意位置，弹出参数设置对话框，勾选④轴名称、设置运动目标位置、速度等参数，选择⑤连续模式，最后点击⑥“确定”，完成一行点位数据添加。如添加单轴定位运动，如下图所示。



直线/圆弧插补数据的添加步骤同单轴定位运动。当选择直线插补运动时，步骤④参数设置弹窗的配置界面如下图所示。



当选择圆弧插补运动时，步骤④参数设置弹窗的配置界面如下图所示。



参数说明如下：

延时时间：执行运动完成后，停止指定的时间，再执行后续动作。

定位结束：执行完成后，不再执行后续动作。

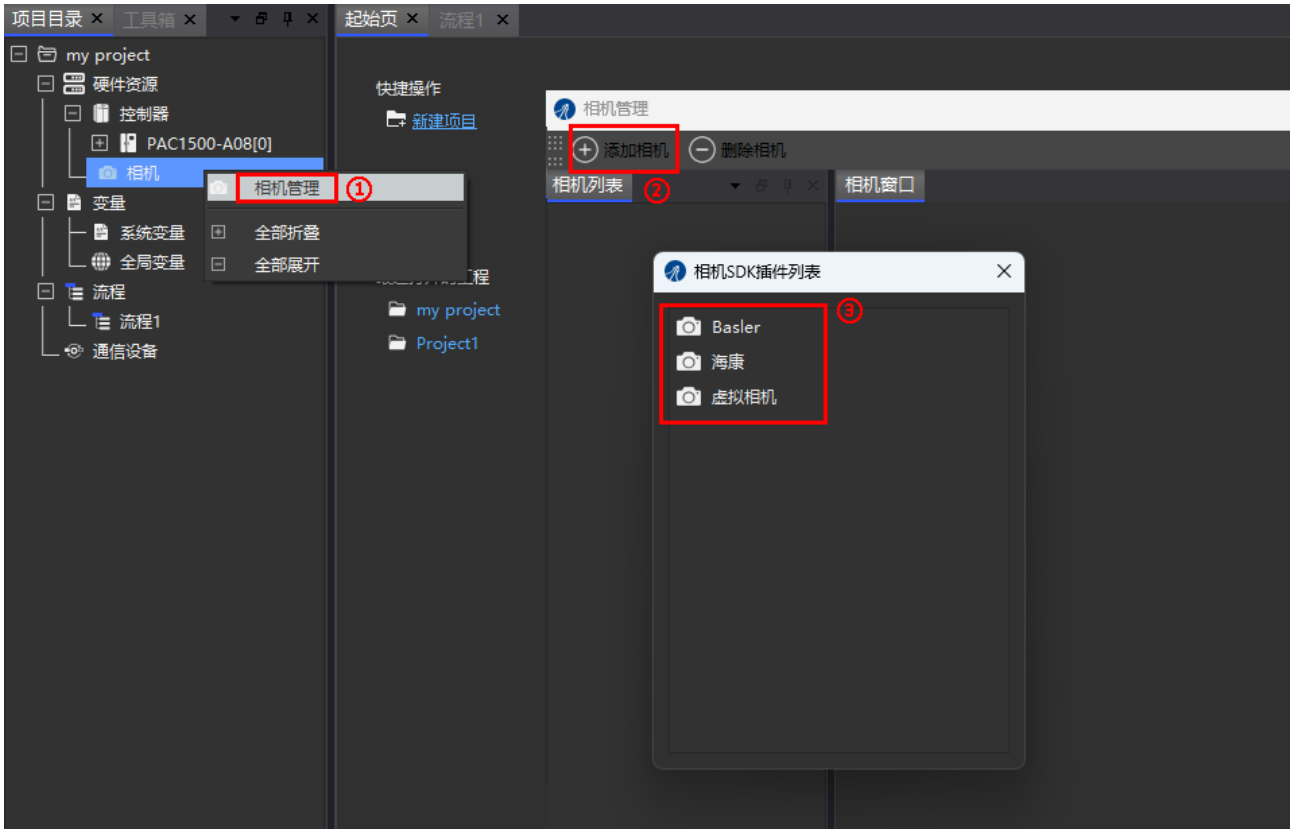
连续定位控制：执行完成后继续执行后续动作。

其他运动参数请参见对应控制器的用户使用手册。

3.2 相机

3.2.1 添加相机

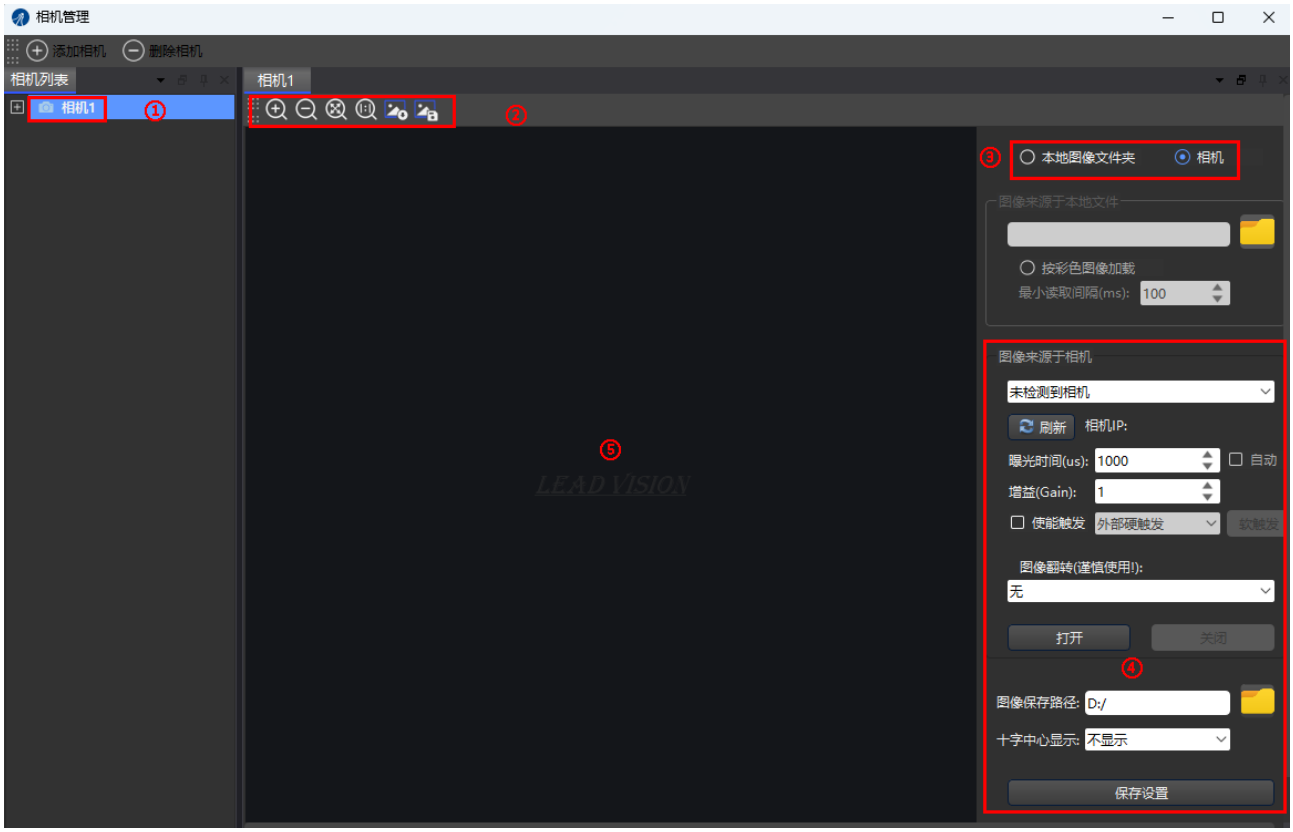
右击项目目录树中的“相机”，在弹窗中点击①“相机管理”，打开“相机管理”窗口，点击②“添加相机”，弹出③“相机插件列表”窗口，根据实际需求选择相机品牌，完成相机的添加。



3.2.2 配置相机

配置相机主要是配置图像源或相机参数，以便后续流程编辑区，调用图像相关工具块使用。如果未连接实际相机，需将图像源选择本地图像文件夹；如果连接实际相机，需将图像源选择相机，请确保相机已上电并处于可用状态，同时确保相机 IP 配置正确。

配置相机步骤：在相机管理窗口的“相机列表”树中，双击①“相机 1”，弹出“相机 1”窗口，如下图所示，完成参数设置后，点击“保存设置”按钮。



相机各参数说明如下：

① 相机列表

显示工程使用的相机数，可切换相机并配置该相机的图像来源及参数。

② 工具栏

对图像进行放大、缩小、保存等操作。从左往右依次为放大图像、缩小图像、使图像适合窗口、显示原始图像、保存图像、图像另存为。

③ 图像源选择

可选择本地图像文件夹或相机图像。

图像来源于本地图像文件。

- 本地图像文件夹：从本地文件夹中读取图片，可通过打开文件对话框，选择图片存放的绝对路径；也可输入相对路径，如./samples/images。
- 最小读取间隔：两张图片读取的间隔时间，ms 单位。

图像来源于相机。

- 刷新：刷新当前连接的相机，在下拉框中选中当前使用的相机。请确保相机已上电并处于可用状态，同时确保相机 IP 配置正确。
- 相机 IP：当前使用的相机 IP 地址。

- 曝光时间：设置相机手动或自动曝光时间，单位 us。
- 增益：设置当前相机的增益
- 图像翻转：对采集的图像旋转处理，包括垂直翻转、水平翻转、旋转 180 度、旋转 90 度、旋转-90 度、垂直翻转+旋转 90 度、水平翻转+旋转 90 度。
- 使能触发：勾选时，相机为外触发模式，包括软触发、外部硬件触发；不勾选时，相机为内触发模式的连续采集模式，即相机自动地按帧率进行曝光输出图像。

④ 保存

图像保存路径：设置保存图像时的存储路径，保存图像的名称为 image+当前时间，如 image2023-12-06_11-51-03_255。

十字中心显示：可选择显示在图像坐标系、显示在窗口坐标系和不显示。

保存设置：保存当前配置的设置选项。

⑤ 显示区

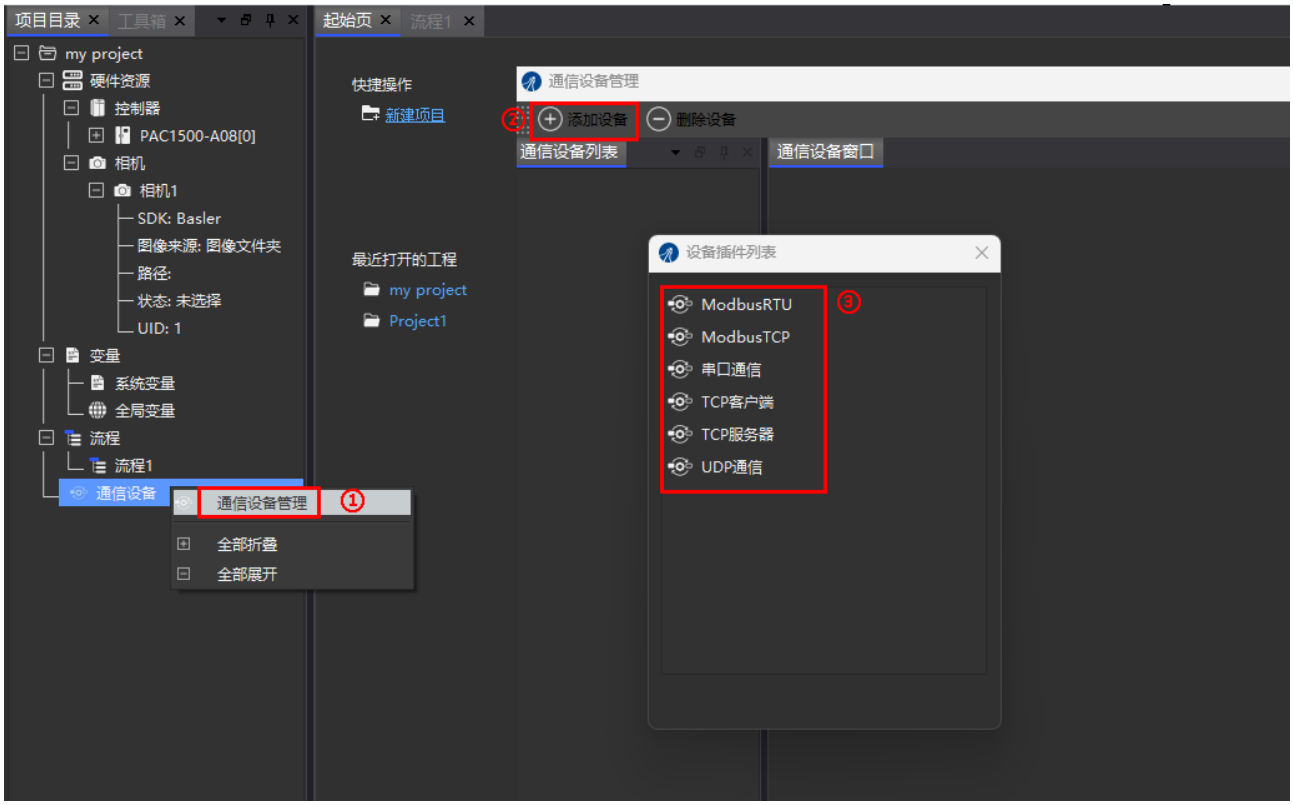
显示图像信息

3.3 通信设备

集中管理项目中所使用的通信设备，支持 TCP、UDP、Modbus、串口等通信协议。用于 LV Studio 开发平台与外部数据的交互，当通信构建后，既可以把处理结果发送给外部，也可以通过外部发送字符触发 LV Studio 开发平台运行或相机拍照。

3.3.1 添加通信设备

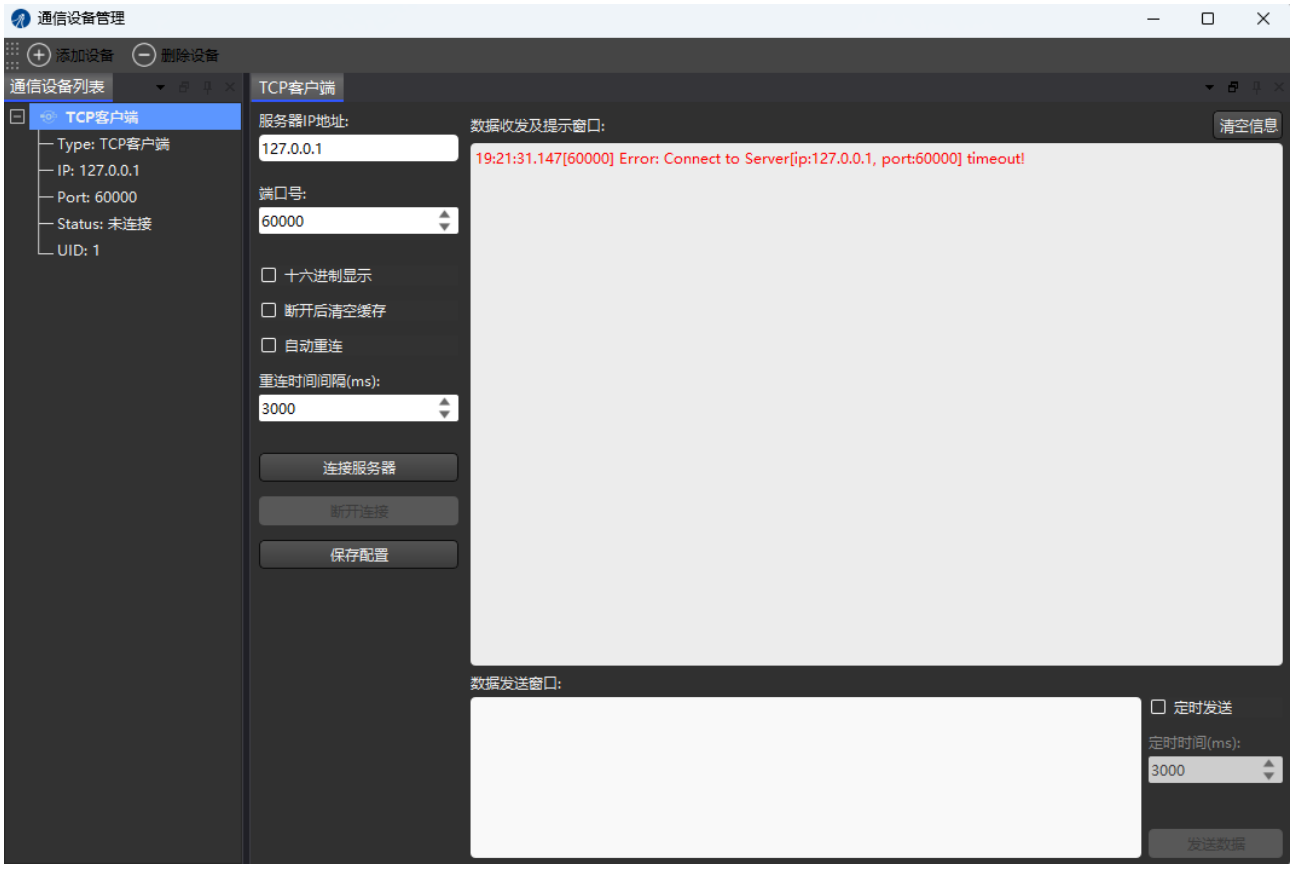
右击项目目录树中的“通信设备”，在弹窗中点击①“通信设备管理”，打开“通信设备管理”窗口，点击②“添加设备”，弹出③“设备插件列表”窗口，根据实际需求选择通信协议，完成通信设备的添加。



3.3.2 TCP 通信

TCP 通信支持客户端和服务器，配置完成后，配合工具箱中的通信相关工具块使用。

- TCP 客户端

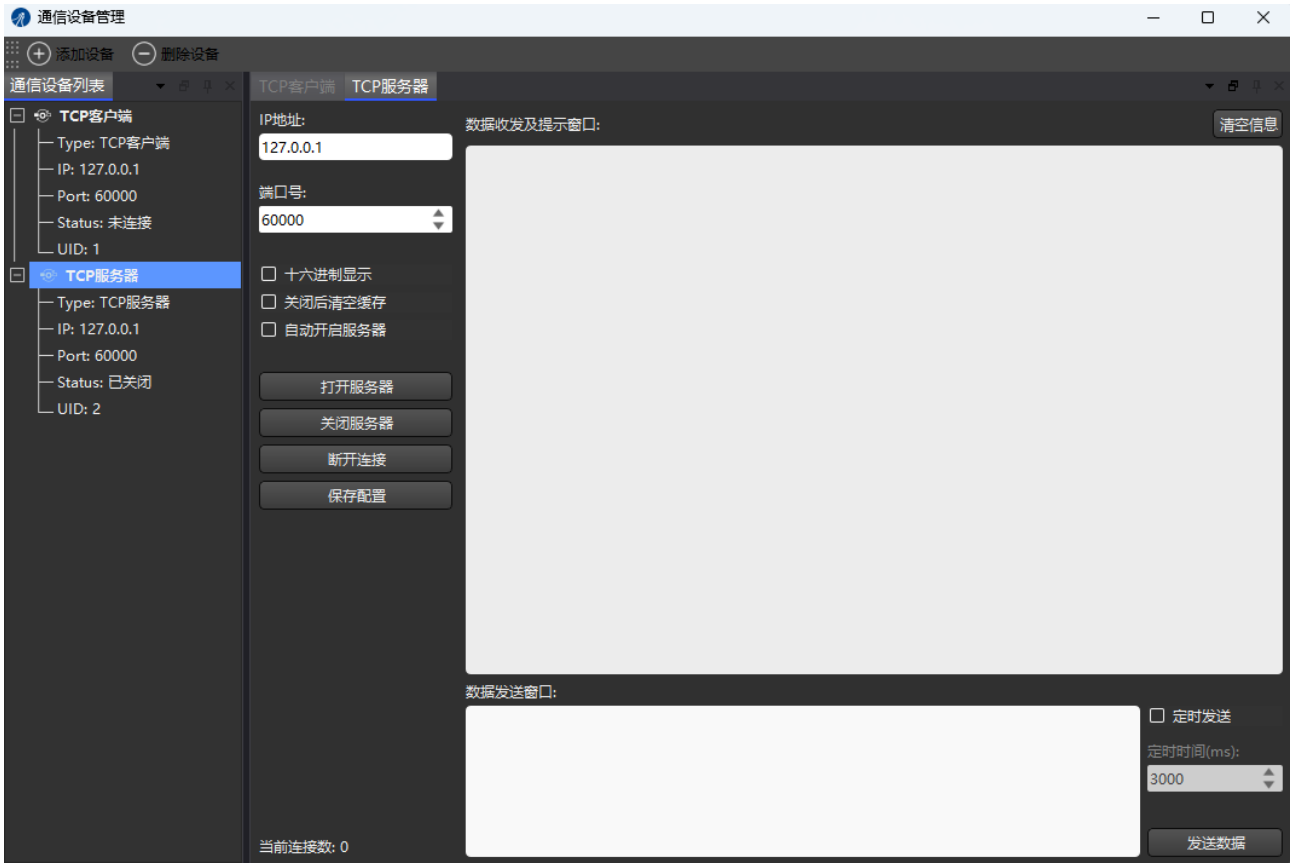


可自定义通信设备名称。操作方法为右击“通信设备列表”树中的“TCP 客户端”，选择“编辑名称”。

- 服务器 IP 地址：连接的目标设备的 IP 地址。
- 端口号：连接的目标设备的端口号。
- 十六进制显示：勾选后，接收数据以十六进制显示。
- 断开后清空缓存：勾选后，连接断开时，清空缓存区的数据。
- 自动重连：勾选后，与服务器断开时会自动重连。
- 重连时间间隔：设置重连时间间隔，单位 ms。
- 连接服务器：与服务器建立连接。
- 断开连接：与服务器断开连接。
- 保存配置：保存当前配置的参数。
- 定时发送：勾选后，按固定时间发送数据，仅供测试使用。
- 定时时间：两帧发送数据之间的间隔，单位 ms。
- 数据收发及提示窗口：显示接收数据及连接信息。
- 数据发送窗口：发送数据输入区。

- 发送数据：将数据发送窗口的数据发送给目标，仅供测试使用。
- 清空信息：清空接收窗口的数据。

- TCP 服务器



- IP 地址：本机服务器 IP 地址。
- 端口号：本机服务器端口号。
- 自动开启服务器：勾选后，重新打开软件时，自动开启服务器。
- 开启服务器：开启本机 TCP 服务器。
- 关闭服务器：断开所有已连接的客户端连接，不再接受任何客户端的连接。
- 断开连接：断开已经建立连接的客户端， 客户端可再次操作建立连接。
- 当前连接数：显示当前已连接的客户端数量。
- 其他参数配置与 TCP 客户端基本一致，可参考 TCP 客户端章节的参数介绍。

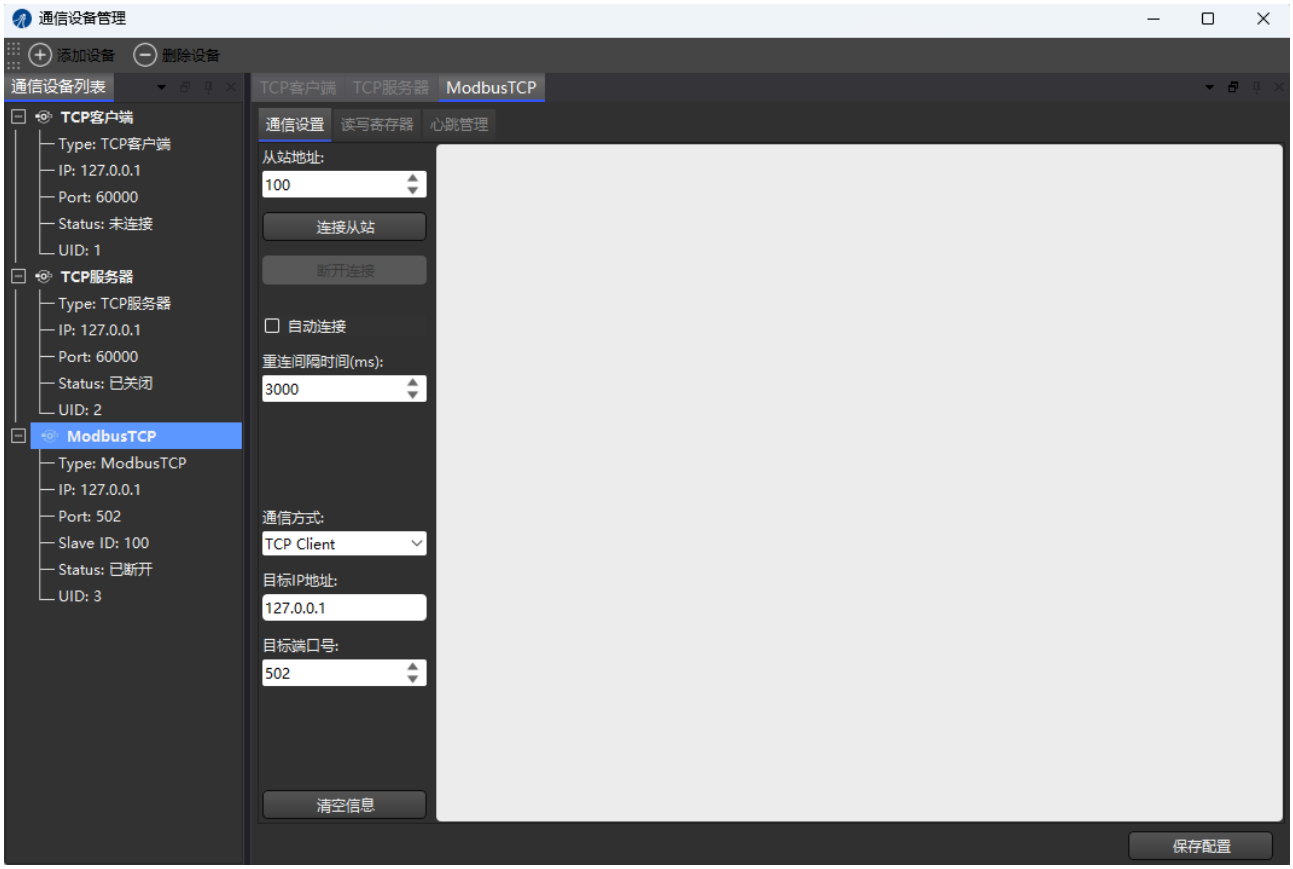
3.3.3 Modbus 通信

Modbus 协议是一个主从架构的协议。有一个节点是主站，其他使用 Modbus 协议参与通信的节点是从站，目前 LV Studio 开发平台仅支持做主站，从下位机（从站）中读取寄存器的值。支持的功能码如下：

功能	功能码
01 线圈状态(0x)	0x01：读线圈状态 0x05：写单个线圈 0x0f：写多个线圈
02 输入状态(1x)	0x02：读输入状态
03 保持寄存器(4x)	0x03：读保持寄存器 0x06：写单个寄存器 0x10：写多个寄存器
04 输入寄存器(3x)	04：读输入寄存器

使用前需先在通信设备中添加 Modbus 通信设备，可添加 ModbusRTU（串口）和 ModbusTCP（网口）。

- ModbusTCP



可自定义通信设备名称。操作方法为右击“通信设备列表”树中的“ModbusTCP”，选择“编辑名称”。

◆ 通信设置选项页

➢ 从站地址：ModbusTCP 从站的协议站地址。

➢ 连接从站：与从站建立 Modbus 通信。

➢ 断开连接：与从站断开 Modbus 通信。

➢ 自动连接：勾选后，检测到通信断开，会自动重连从站设备。

➢ 重连间隔时间：重连的时间间隔，单位 ms。

➢ 通信方式：仅支持客户端模式，即主站模式。

➢ 目标 IP 地址：从站的 IP 地址。

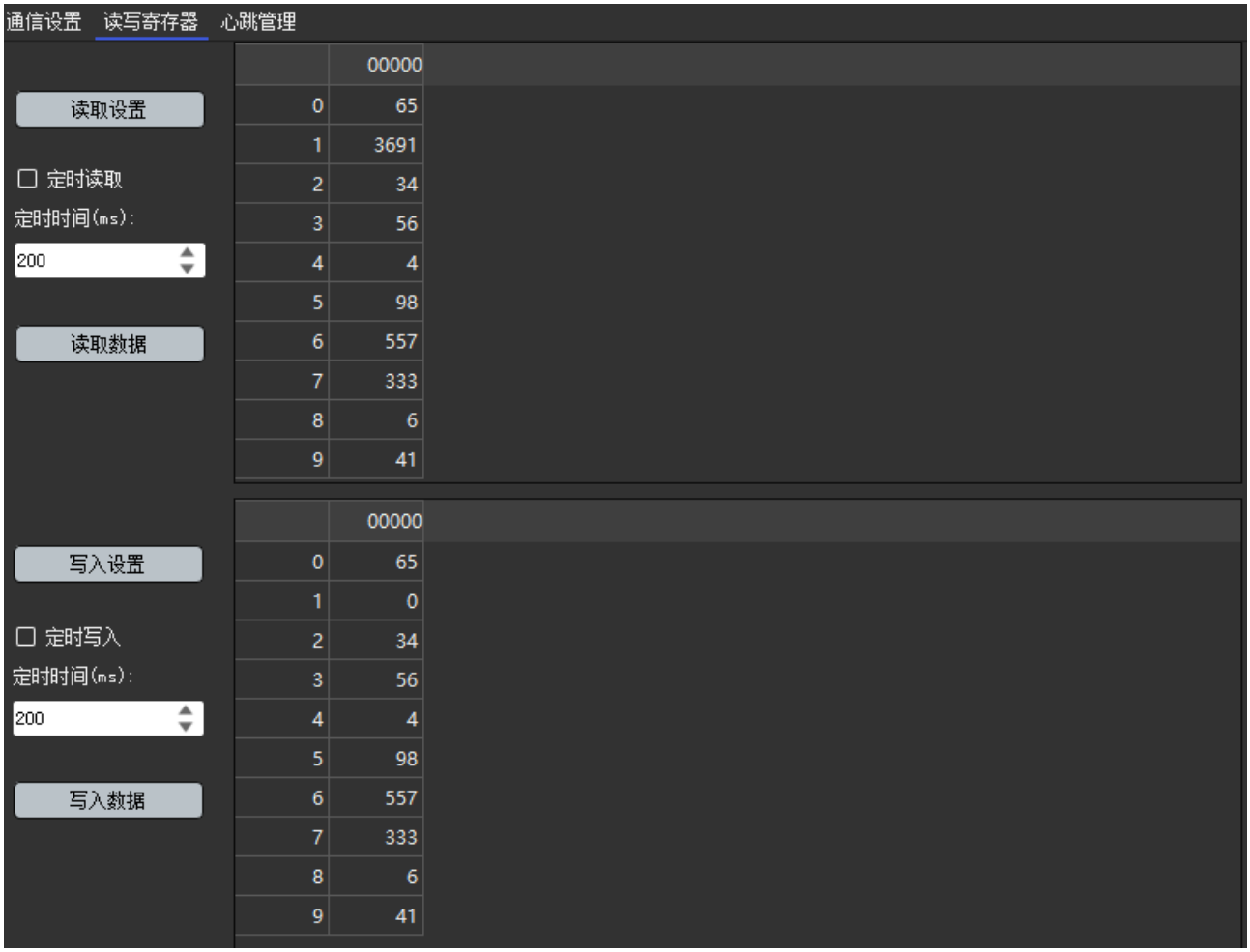
➢ 目标端口号：从站的端口号。

➢ 清空信息：清空消息区的信息。

➢ 保存配置：保存设置的通信参数。

◆ 读写寄存器选项页

仅用于 Modbus 通信测试。



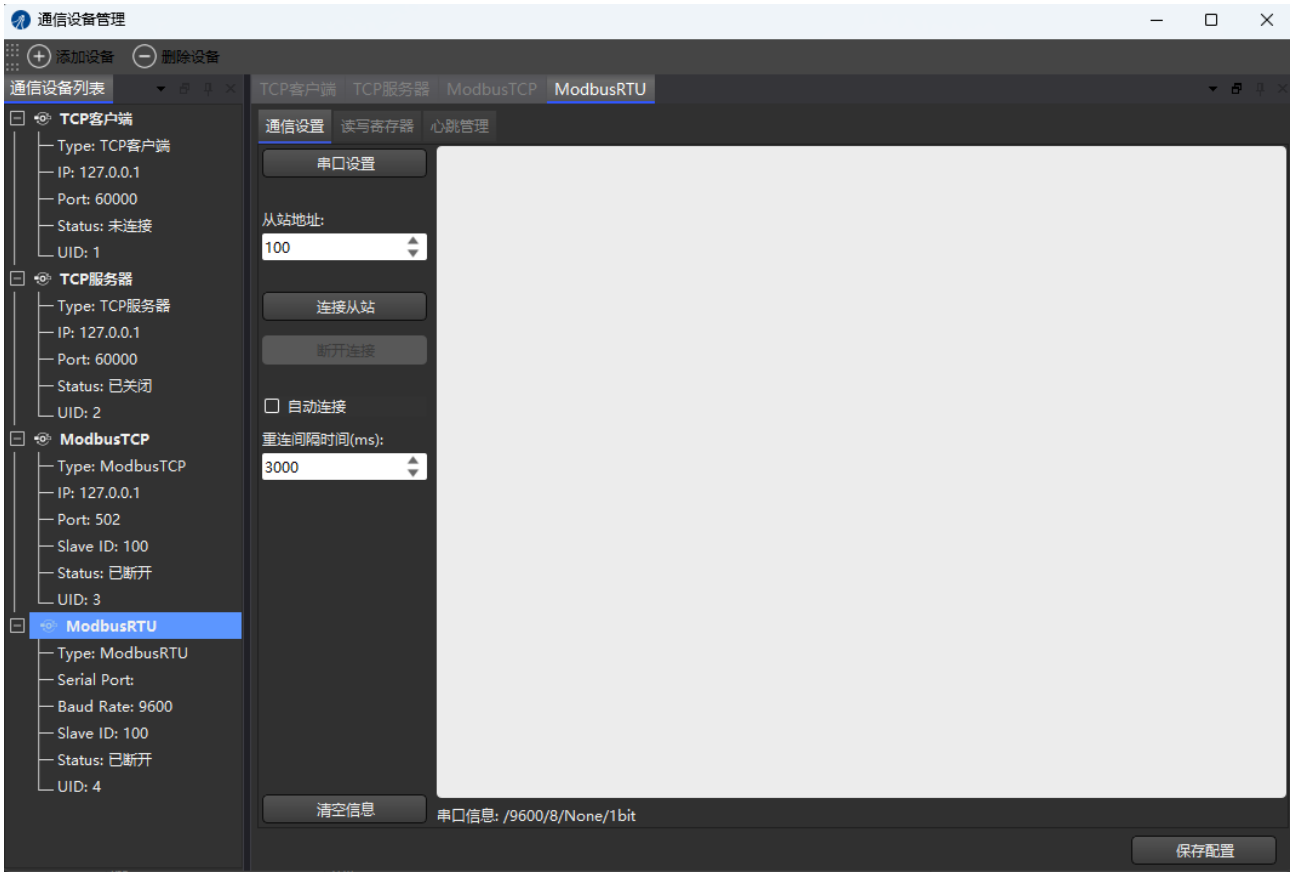
- 读取设置：配置读取的功能码、起始地址、寄存器数量。
- 定时读取：勾选后，按固定时间读取通信数据。
- 定时时间：设置定时读取的时间，单位 ms。
- 读取数据：点击时读取一次数据。
- 写入设置：配置写入的功能码、起始地址、寄存器数量。
- 定时写入：勾选后，按固定时间写入通信数据。
- 定时时间：设置定时写入的时间，单位 ms。
- 写入数据：点击时写入一次数据。

◆ 心跳管理选项页



- 开启心跳功能：勾选后，启用心跳包功能，定时写入数据。
- 心跳间隔时间：两帧数据的间隔时间，单位 ms。
- 寄存器功能类型：配置 Modbus 功能码。
- 写入地址：设置 Modbus 写入数据的地址。
- 写入值：设置 Modbus 写入的数据值。

- ModbusRTU



可自定义通信设备名称。操作方法为右击“通信设备列表”树中的“ModbusRTU”，选择“编辑名称”。

◆ 通信设置选项页

串口设置：配置串口号、波特率、数据位、校验位、停止位等参数。

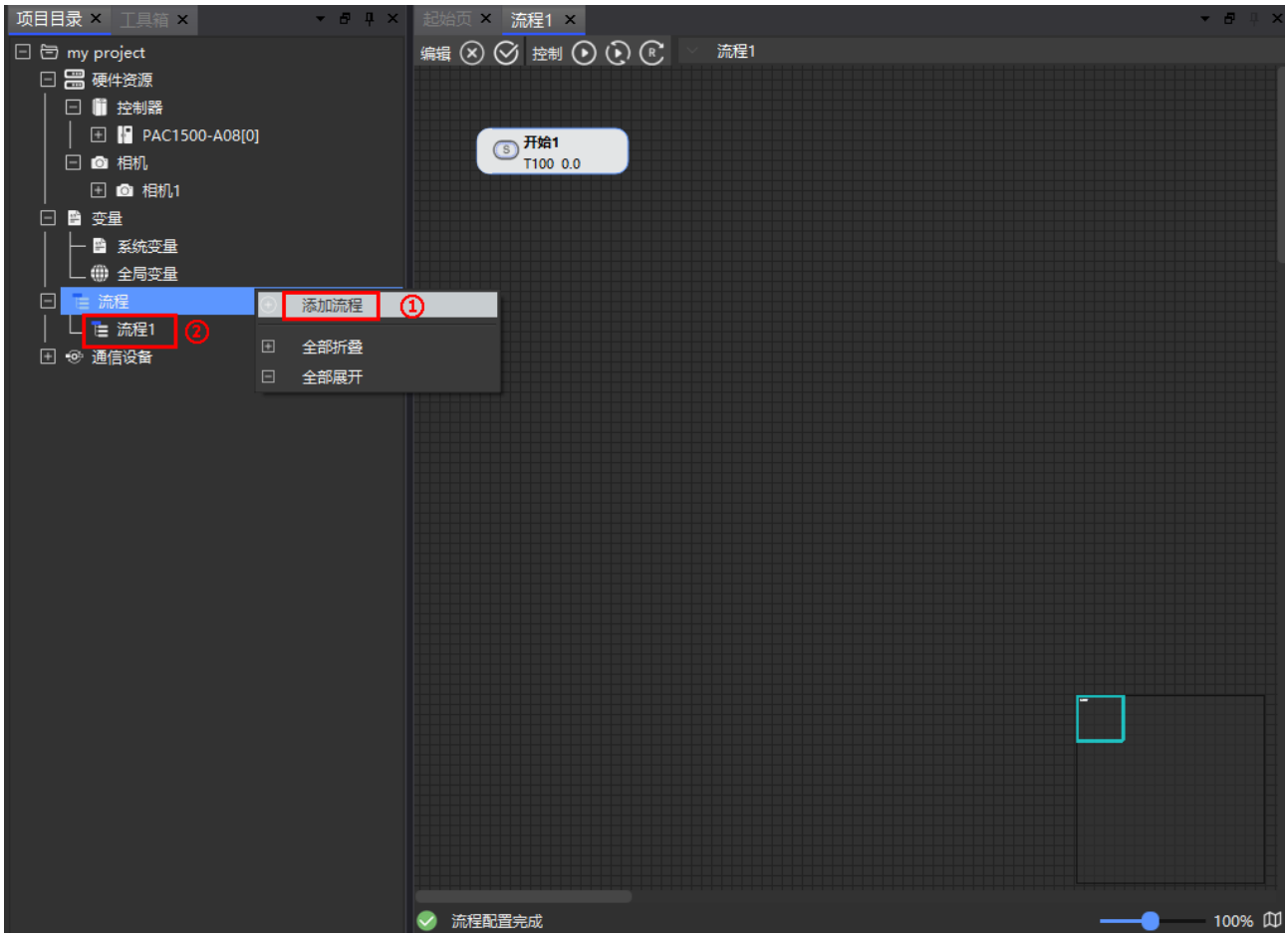


- 从站地址：ModbusRTU 从站的协议站地址。
- 其他参数配置以及读写寄存器选项页、心跳包选项页配置，与 ModbusTCP 基本一致，可参考 TCP 通信章节的参数介绍。

4. 项目搭建-流程配置

4.1 添加流程

右击项目目录树中的“流程”，在弹窗中点击①“添加流程”，自动生成并打开②“流程 1”窗口，流程名称默认为流程 1、流程 2…依次往后排序，如下图所示。用户可在“流程 1”的编辑区，根据实际项目需求，编辑项目流程。



流程名称修改：右击项目目录树中的“流程 1”，选择①“编辑名称”，在弹出的编辑对话框中，输入②新名称，点击“OK”，完成流程名称的修改。流程名称修改是为了利于用户阅读理解程序。

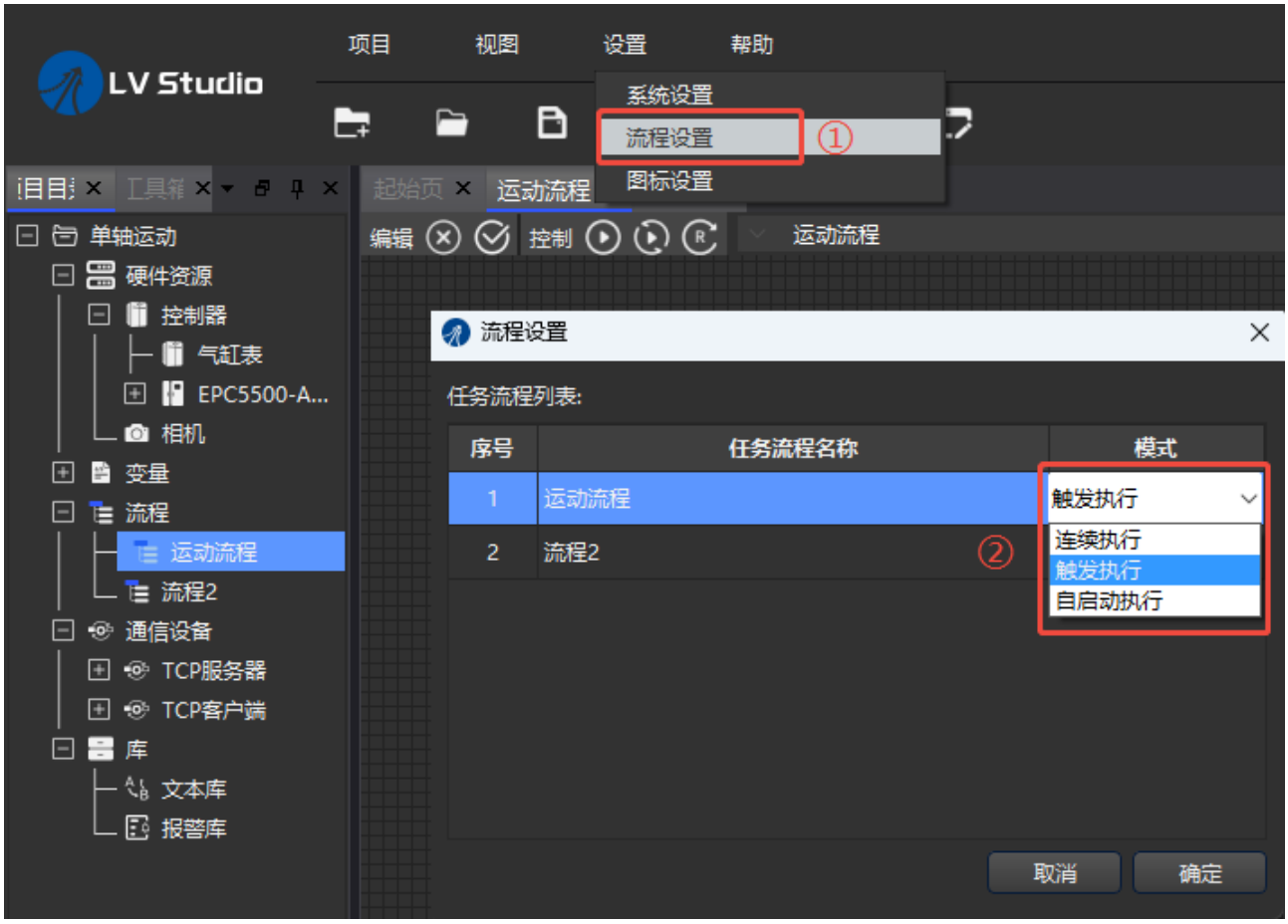


4.2 配置流程

配置流程是指配置流程的执行模式，添加完流程后，用户还需配置每个流程的执行模式，包括连续执行、触发执行和自启动执行，默认为连续执行模式。

- 连续执行：流程启动后循环执行。
- 触发执行：流程被触发后执行一次。
- 自启动执行：流程进入用户界面后自动执行，可用于监控 IO 状态、轴状态等信息。

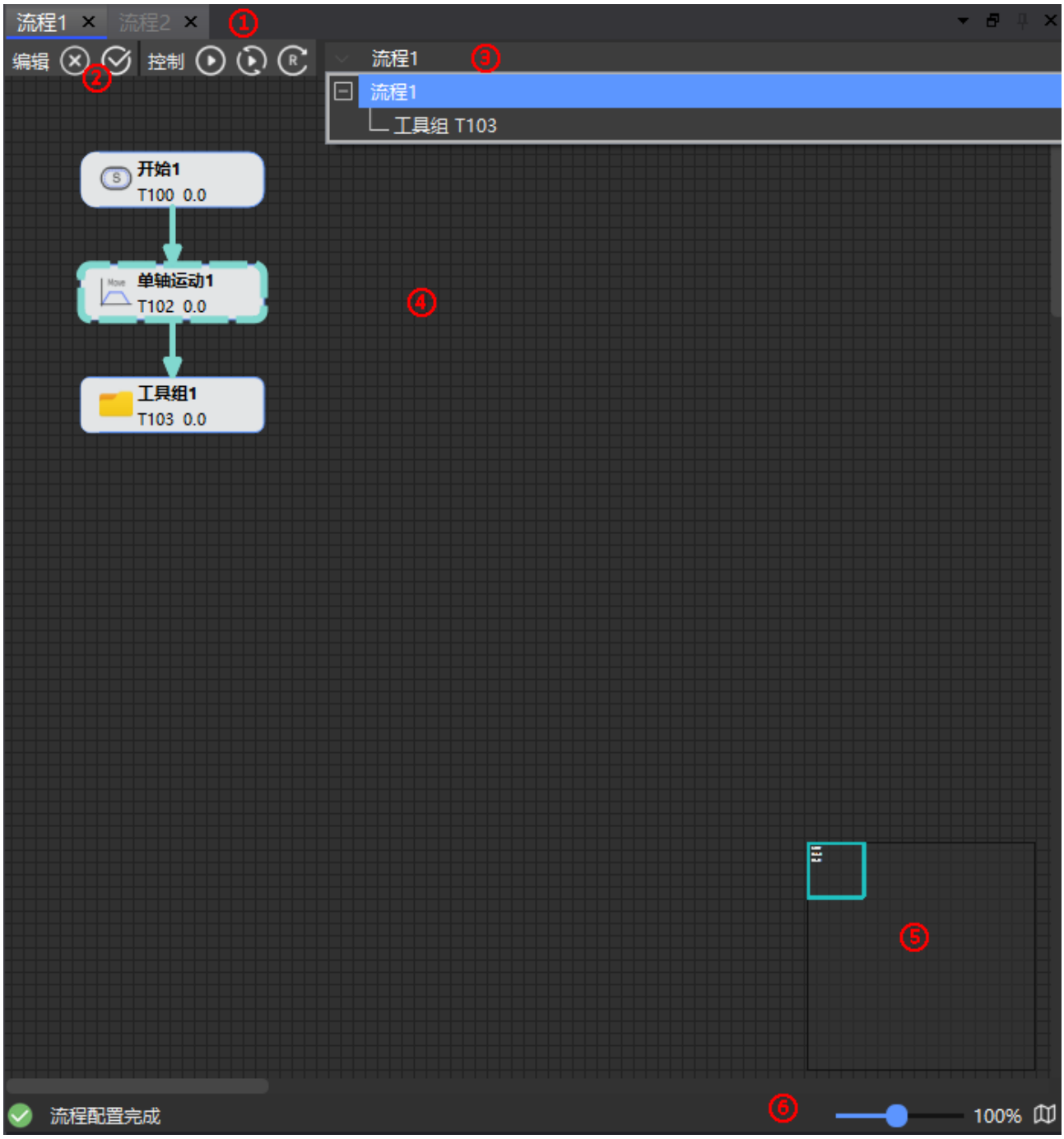
配置流程执行模式的操作步骤：选择“设置”菜单下的子菜单①“流程设置”，打开“流程设置”窗口，在②“模式”下拉框中配置指定流程的执行模式，最后点击确认，完成配置。



4.3 编辑流程

4.3.1 编辑区介绍

流程编辑用于编辑视觉和运动控制项目的执行流程，可分成六部分，包括①流程选项卡、②操作工具栏、③流程下拉选单、④流程编辑区、⑤鹰眼导航窗、⑥状态栏。



①流程选项卡

用于流程切换，点击不同的选项卡可切换到不同的任务流程。

②操作工具栏

操作工具栏分为编辑工具栏、调试工具栏和控制工具栏。

- 编辑工具栏



用于删除选中的工具块及流程的编译检查。该工具栏较少使用，一般框选工具块实现全部删除；工具块拖动到流程编辑区时，也会自动执行检测。

删除选中的工具

校验程序流程

- 调试工具栏



当在流程中设置断点后，运行流程时编辑工具栏将切换为调试工具栏。

继续运行至下一个断点

单步调试运行

- 控制工具栏



控制单个流程的运行、连续运行、复位等操作，调试流程使用。

运行一次本流程，点击运行后，如果流程还处于运行中，再次点击实现暂停流程运行。

连续运行本流程，点击运行后，如果流程还处于运行中，再次点击实现暂停流程运行。

复位本流程，复位到流程的首行，重新执行流程时使用。

③流程下拉选单

当流程中有工具组时，用于切换工具组中的流程和返回本流程。

④流程编辑区

编辑区域主要用于编辑视觉和运动控制项目的执行流程，也可查看工具块的运行状态、耗时、工具块唯一 ID。点击可以选中工具块，双击弹出工具块的属性配置框，每个工具的属性说明，请参见工具块使用说明章节。右击弹出选单，如下图所示。



流程右键菜单功能：

删除：删除当前选中的工具。

编辑名称：修改当前选中工具的名称。

添加备注信息：给工具添加注释说明信息。

删除断点：删除前选中工具块的断点。

断点：给当前选中的工具块设置断点。

删除所有断点：清除当前流程的全部断点。

禁用：禁用选中的工具块。

⑤鹰眼导航窗

通过滑动导航窗，有助于迅速浏览流程整体布局。

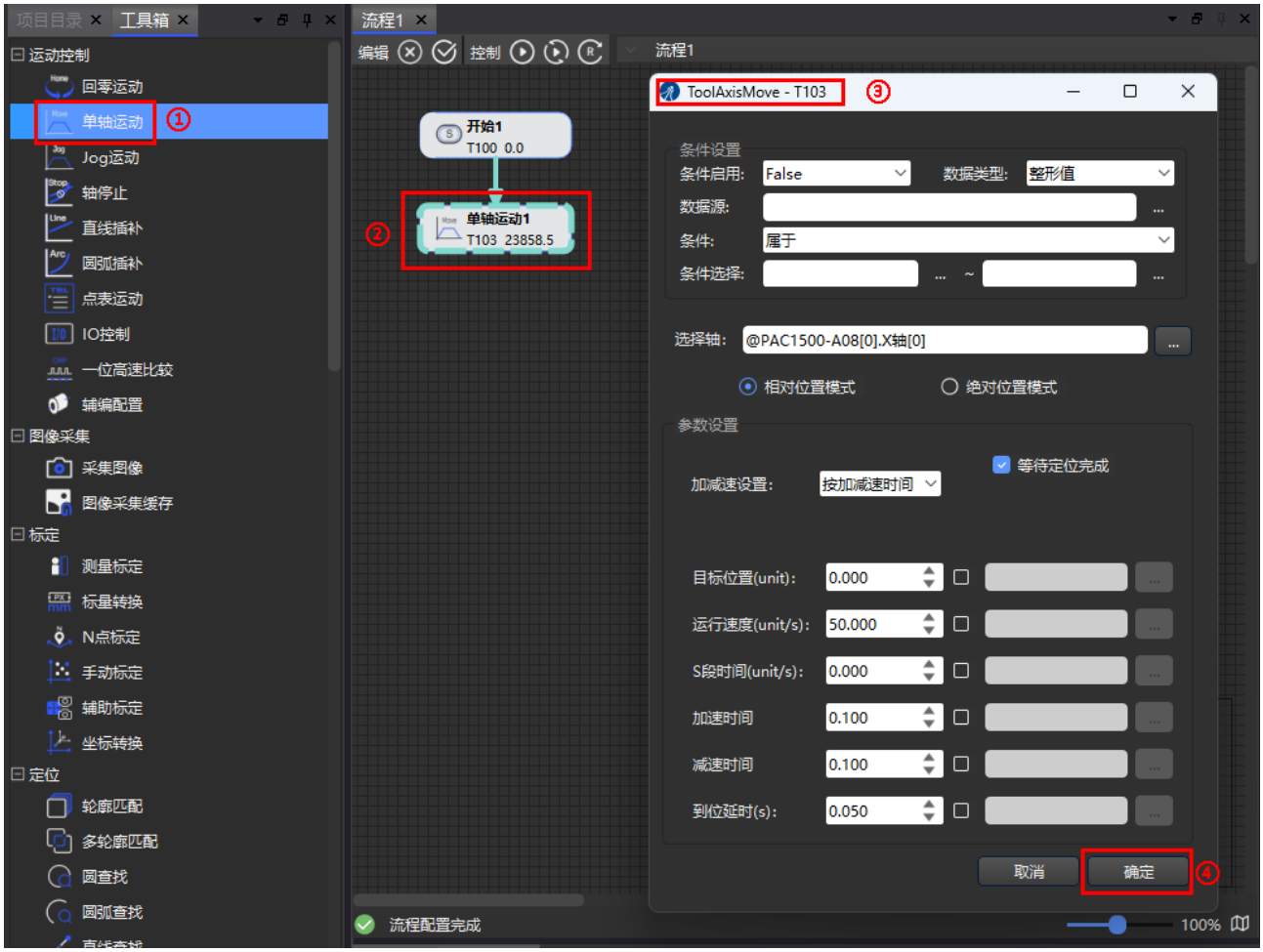
⑥状态栏

显示流程配置状态、流程放大缩小操作（10%~200%）及鹰眼导航窗的折叠。

4.3.2 编辑流程

从工具箱中拖动①工具块“单轴运动“，到②流程编辑区，释放鼠标后，将该工具块与开始块连线，连线完成后，双击工具块，打开该工具块的③属性窗口，如配置单轴运动的轴

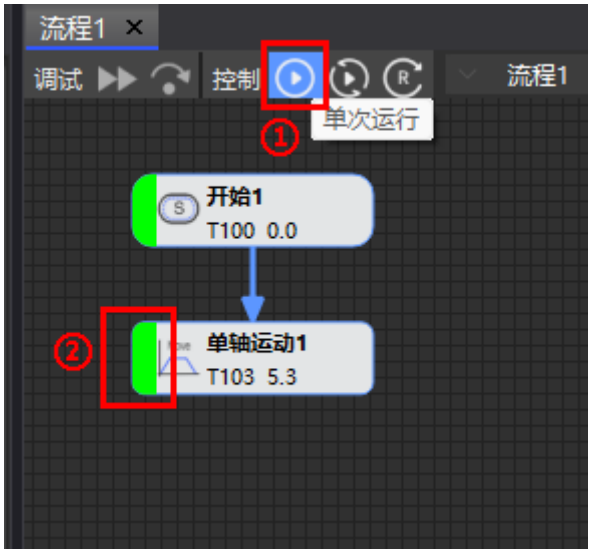
号、运动模式、目标位置、加减速时间等，点击④“确认”保存关闭该工具的属性窗口，至此即完成一个单轴运动的流程编辑。



4.3.3 运行流程

每个流程都有一个唯一的开始块，流程从开始块开始执行。流程执行时，工具块会以不同颜色显示工具块的运行状态，灰白色表示未运行，黄色表示运行中，绿色表示运行成功，红色表示运行失败。

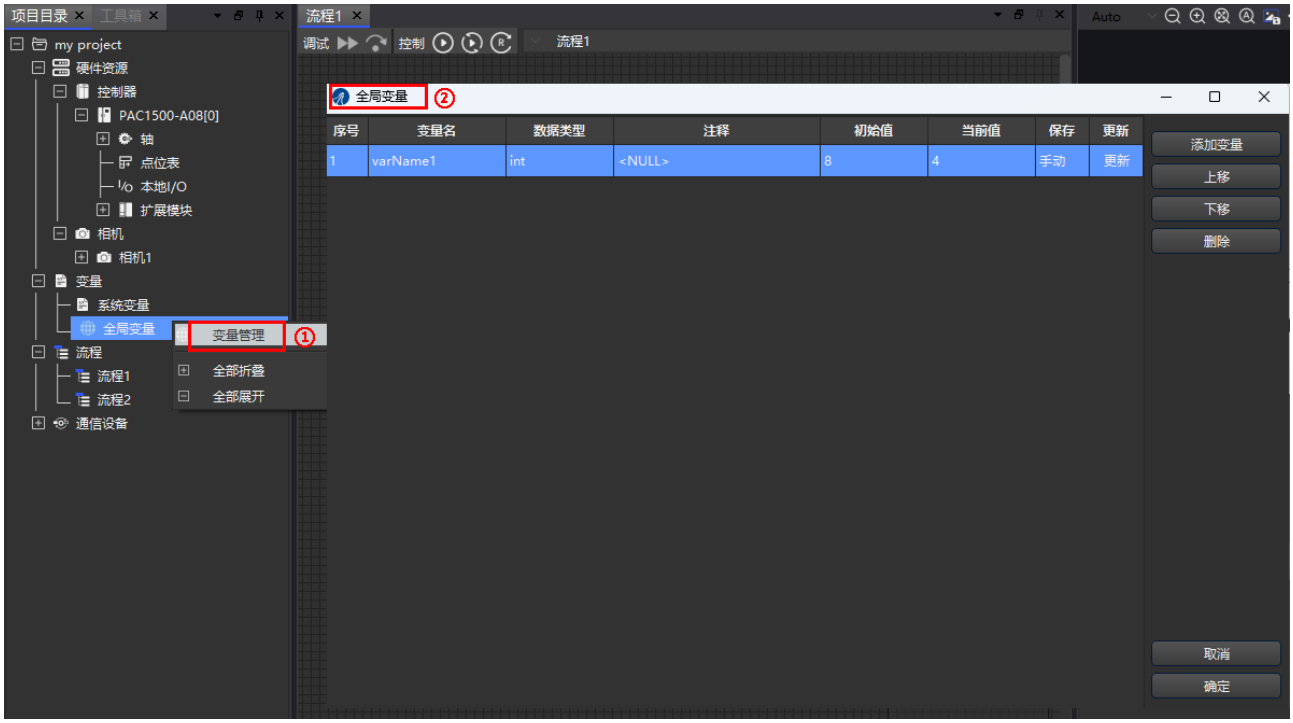
运行流程操作步骤：点击①单次运行，等待流程执行完成，执行完成后，工具块的②运行状态如下图所示。



4.4 全局变量

全局变量是所有视觉/运动工具块外部定义的变量，它在整个项目内都有效，可以被项目内的所有流程调用或修改，可自定义变量名称、类型和初始值，支持 bool、int、float 和 string 类型的变量。

右击项目目录树中的“全局变量”，在弹窗中选择①“变量管理”，打开②全局变量窗口，在此窗口进行全局变量的增删改配置。



全局变量窗口的参数说明：

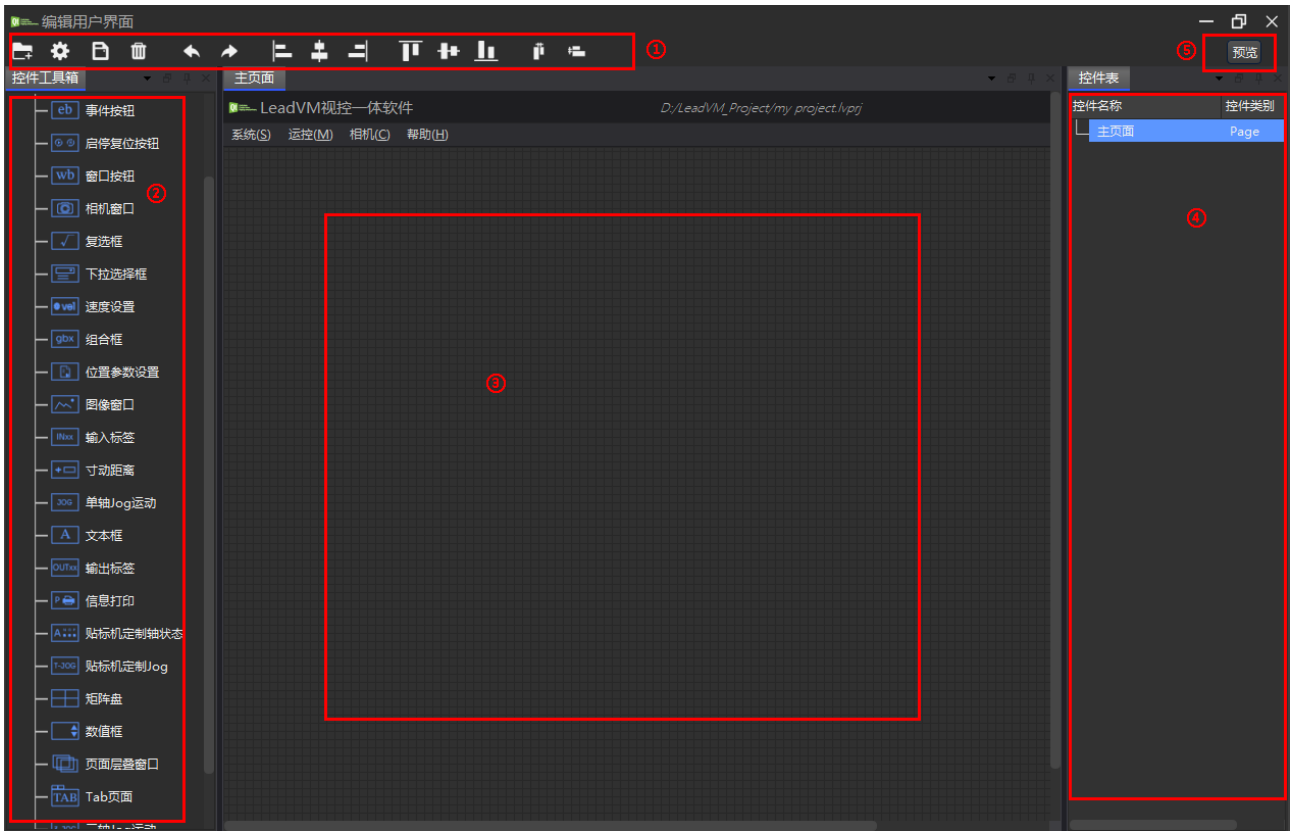
- 变量名：输入添加的全局变量名称。
- 数据类型：包括 bool、int、float 和 string 类型。
- 初始值：全局变量初始值。
- 当前值：读取的全局变量当前值。
- 保存：包括手动和自动选项，选择自动，将自动将当前值同步给初始值；选择手动，需点击更新才会将当前值同步给初始值。
- 添加变量：单击后可新增全局变量。
- 上移/下移：可对变量的位置进行上下调整。
- 删除：可删除选中的变量。
- 取消：取消添加的全局变量。
- 确定：保存添加的全局变量。

5. 项目搭建-用户界面组态

5.1 用户界面介绍

用户界面用于直观操控和监控项目运行，终端现场用户通过用户界面操控项目运行。用户界面组态窗口用于自定义设计运行界面，下文为您介绍用户界面组态窗口各区域功能及说明。

单击 LV Studio 开发平台主界面工具条中的可进入用户界面组态窗口。用户界面整体区域划分为四个区域和一个预览键，分别是①页面工具条、②控件工具箱、③页面编辑区、④控件管理区和⑤预览界面按键。如下图所示。



- 页面工具条

  页面工具条，可设置以下功能，从左往右依次说明。

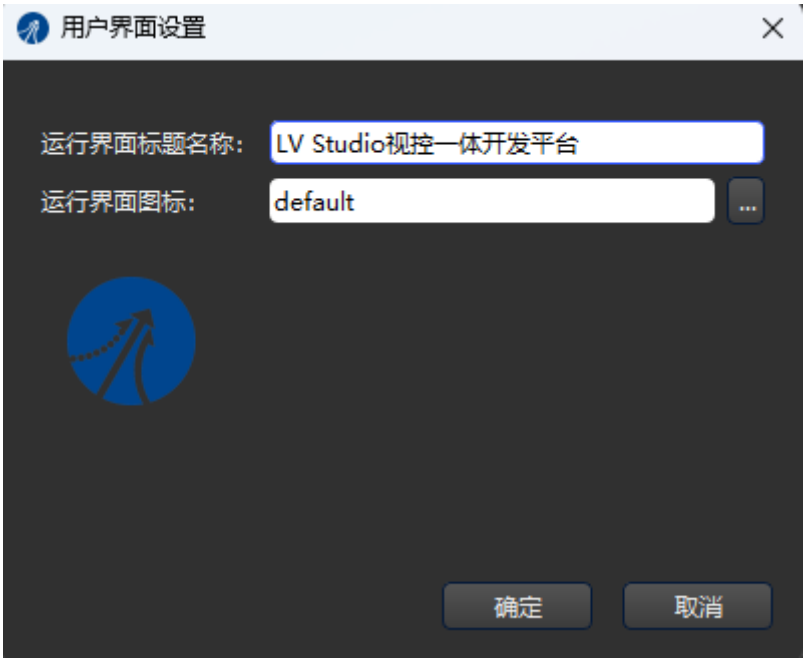


➢ 创建新页面

组态多个页面时，使用该按钮创建多个页面。

➢ 用户页面设置

设置运行界面的标题及图标。



- 保存

  保存当前组态的用户界面。

- 删除

  删除选中的控件。

- 撤销和重做

  用户控件的移动、添加、删除等操作后的撤销和恢复。

- 控件对齐操作

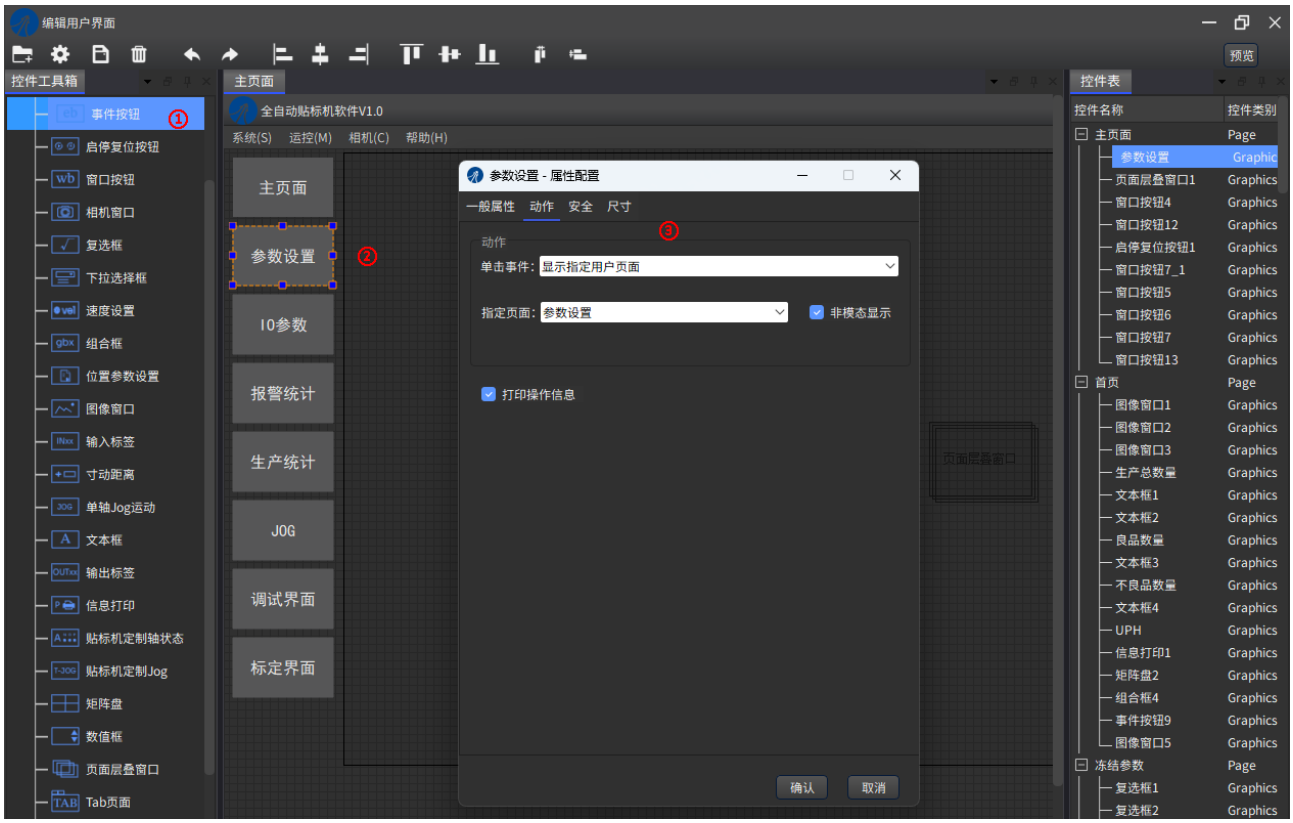
  页面编辑区域选中多个控件时，以选中控件组成的区域为基准对选中控件的对齐方式进行调整。可选左对齐、垂直居中、右对齐、顶部对齐、水平居中、底部对齐。

- 控件布局操作

  页面编辑区域选中多个控件时，以选中控件组成的区域为基准对选中控件的分布方式进行调整。可选水平均匀布局和垂直均匀布局。

5.2 用户界面组态

根据项目需要，从控件工具箱中左击拖拽①按钮、文本等可视化控件到编辑区并释放，双击②打开控件③属性窗口，为每个添加的控件订阅数据源，设置控件文本颜色、字体、背景色、尺寸、安全等功能，完成用户界面组态。



5.3 预览组态界面

组态完成后，点击用户界面右上角的“预览”按钮，可预览及调试组态效果。



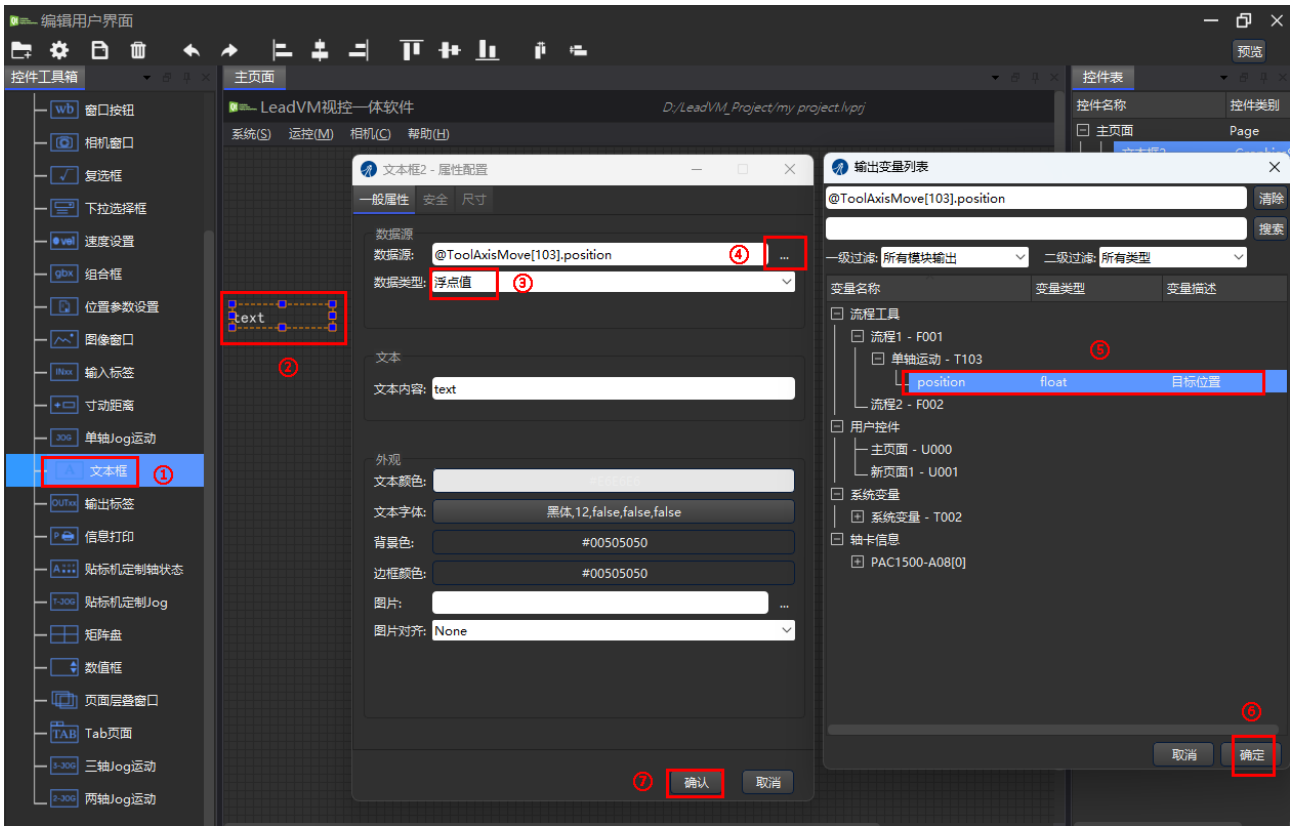
5.4 控件工具箱

控件工具箱主要包括一些常用的标签、按钮、数值框、组合框、窗口等控件。

5.4.1 文本显示框

用于显示固定文本和订阅的文本值。数据源支持订阅项目中的布尔/整型/浮点/字符串类型的数据。数据源可以不订阅，此时显示文本内容里面的固定文本。文本框也可设置文本颜色、字体、背景色、图片、尺寸等功能。

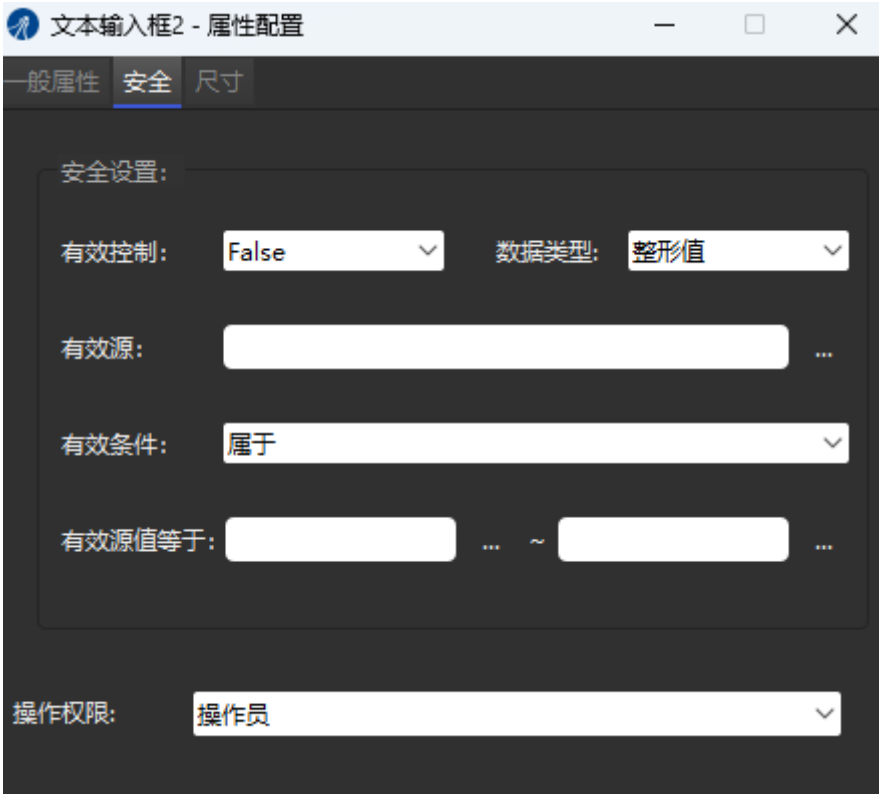
以文本框订阅数据源为例，操作步骤如下：在控件工具箱中，找到“文本框”，左击将其拖拽到页面编辑区并释放，文本框即添加到页面中。双击②“文本框”打开属性窗口，在属性窗口中下拉③选择“数据类型”，如浮点值，再点击④“数据源”，打开数据源订阅窗口，选择浮点型变量值，如订阅⑤“T103 单轴运动”工具块的目标位置，完成后点击⑥“确定”，关闭订阅窗口，最后点击⑦“确认”，完成数据源的订阅。如下图所示。



5.4.2 文本输入框

用于输入字符串文本值，输入的文本内容为字符串类型，流程中可直接访问该文本框输入的字符串内容。文本输入框也可设置文本颜色、字体、背景色、尺寸等功能。

文本输入框支持安全功能，可通过有效源控制是否允许输入文本，当前启用有效控制后，需满足设定的条件，才允许操作该控件。也支持权限安全设置，满足相应权限才允许操作该控件。如下图所示。



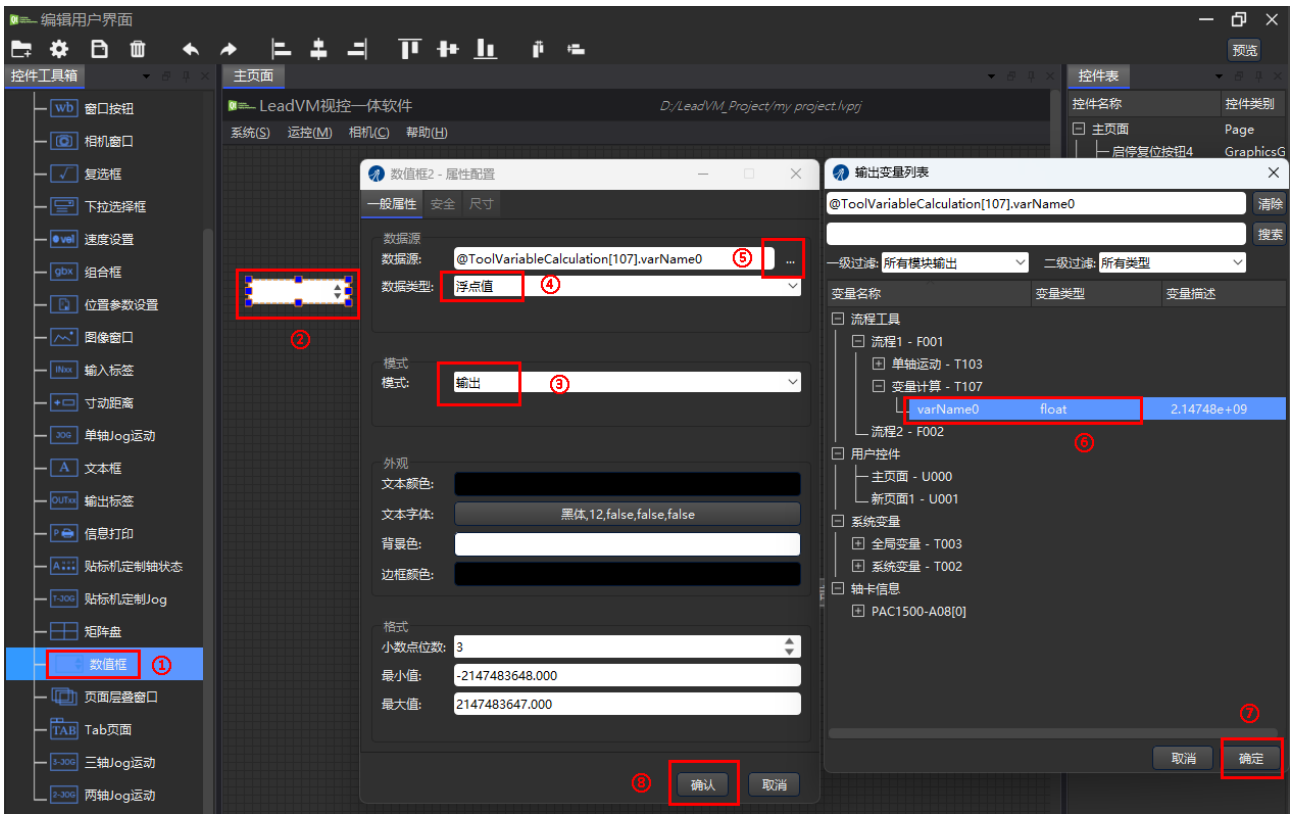
5.4.3 数值框

用于显示和写入变量的值，变量通过数据源进行订阅，可以订阅流程中 int 和 float 类型的数据。通过配置模式，可定义数值框为输出和输入两种模式，当模式设置“输出”时为显示数值控件，需订阅数据源，此时不可编辑数值；当模式设置“输入”时为输入数值控件，此时可编辑数值，同时需要输入初始值，无需订阅数据源。

数值框支持设置最大值和最小值限制，保证位置安全，也支持格式化小数位数，设置文本颜色、字体、背景色、尺寸、安全等功能。

以数值框设置“输出”模式为例，操作步骤如下：

在控件工具箱中，找到“数值框”，左击将其拖拽到页面编辑区并释放，数值框即添加到页面中。双击②“数值框”打开属性窗口，在属性窗口中选择③“模式”为“输出”，④“数据类型”为浮点值，再点击⑤“数据源”，打开数据源订阅窗口，选择浮点型变量值，如订阅⑥“T107 变量计算”工具块的输出值，完成后点击⑦“确定”，关闭订阅窗口，最后点击⑧“确认”，完成数据源的订阅。如下图所示。



在数值框属性窗口的安全页面，可设置操作该控件的有效控制变量及访问权限等安全，用于防止控件被意外操作。有效控制开启后，订阅的数据源满足指定条件，才允许操作该控件，否则禁止操作，True 开启，False 关闭，默认 False。除此之外，用户还可设置控件的操作权限，只允许满足登录权限的用户操作。

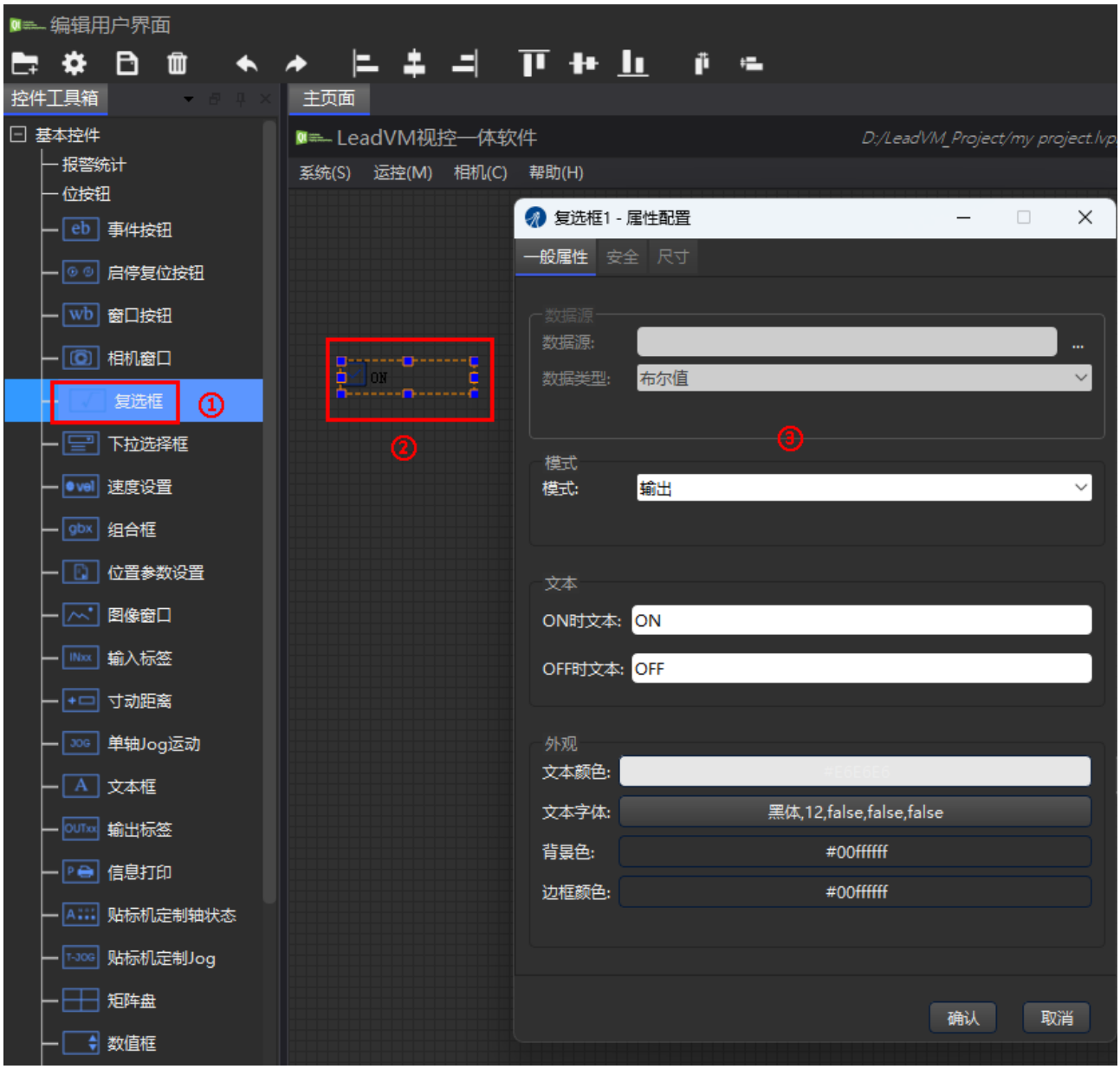


5.4.4 复选框

复选框用来表示是否选取了某个选项条件，常用于为用户提供具有是/否或真/假值的选项。复选框可以订阅流程中布尔和整型类型的数据。通过配置模式，可定义复选框为输出和输入两种模式。当模式设置“输出”时为显示控件，此时不可编辑；当模式设置“输入”时为输入控件，此时可编辑状态，同时需要配置初始值；

复选框支持设置文本颜色、字体、背景色、尺寸、安全等功能。

在控件工具箱中，找到“复选框”，拖拽添加其中的“复选框”。双击打开“复选框”的属性窗口，之后订阅流程中的 bool 变量，配置模式、显示文本等属性。其配置过程可参考 “数值框”控件。

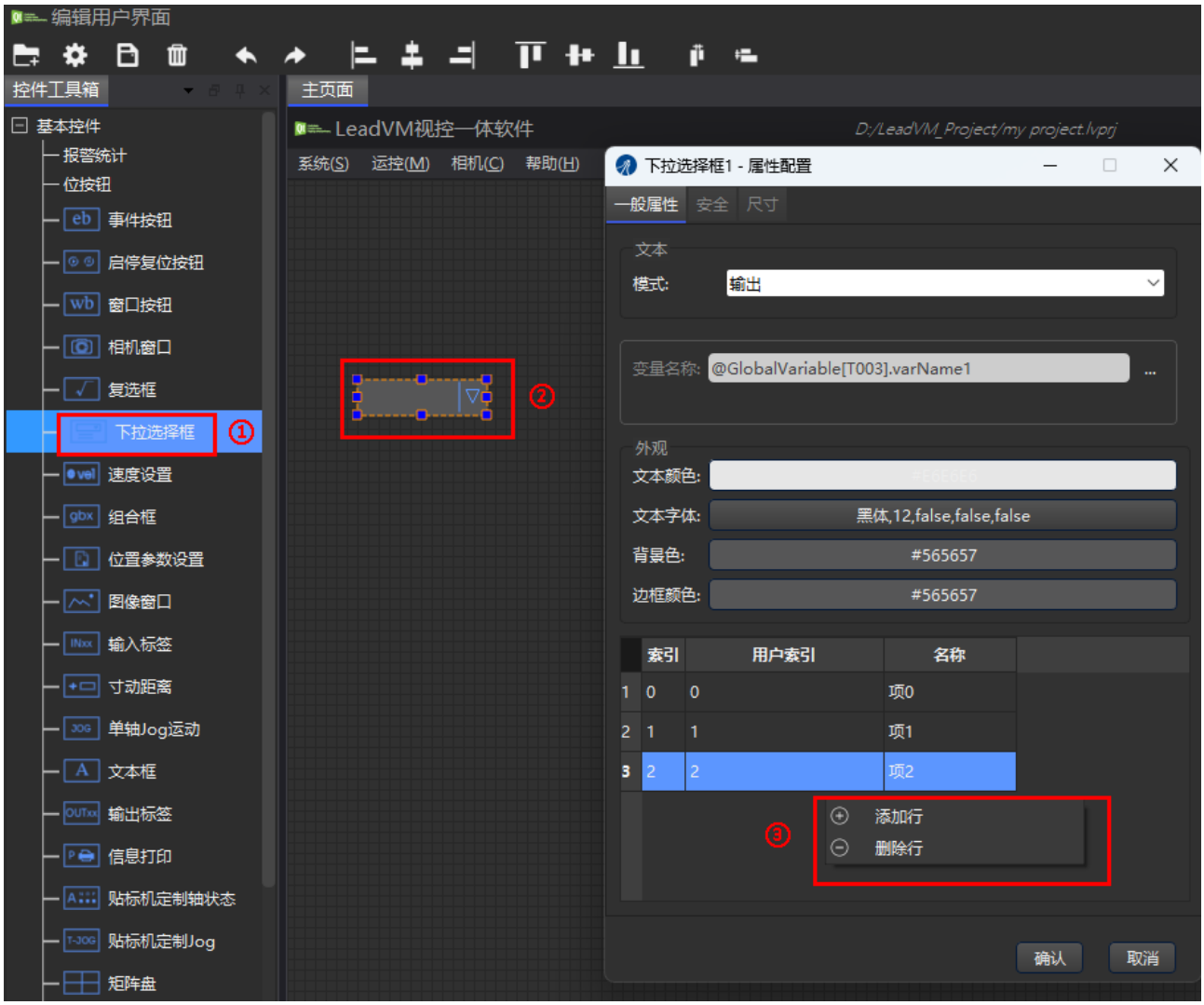


5.4.5 下拉选择框

允许用户根据下拉菜单选择想要改变的整型数据值。可定义下拉选单为输出和输入两种模式，当模式设置“输出”时为显示控件，此时不可编辑；当模式设置“输入”时为输入控件，此时可编辑，同时需设置初始值；

下拉选择框支持设置文本颜色、字体、背景色、尺寸、安全等功能。

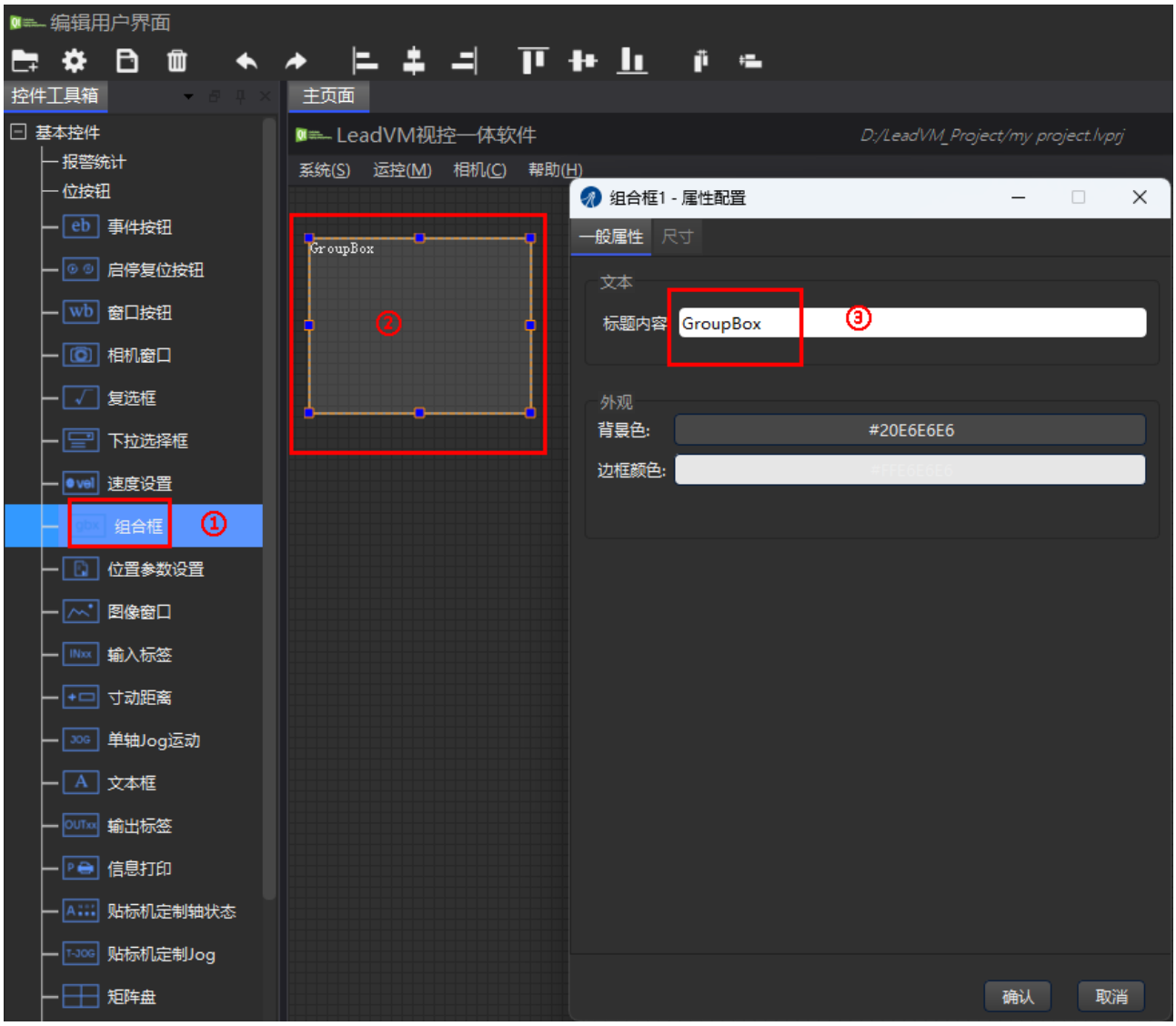
在控件工具箱中，找到“下拉选择框”，拖拽添加其中的“下拉选择框”。双击打开“下拉选择框”的属性窗口，之后订阅流程中的 int 变量，配置模式、添加显示文本等属性。其配置过程可参考 “数值框”控件。



5.4.6 组合框

组合框是一种容器，可以将多个控件放入组合框，实现整体布局及移动。

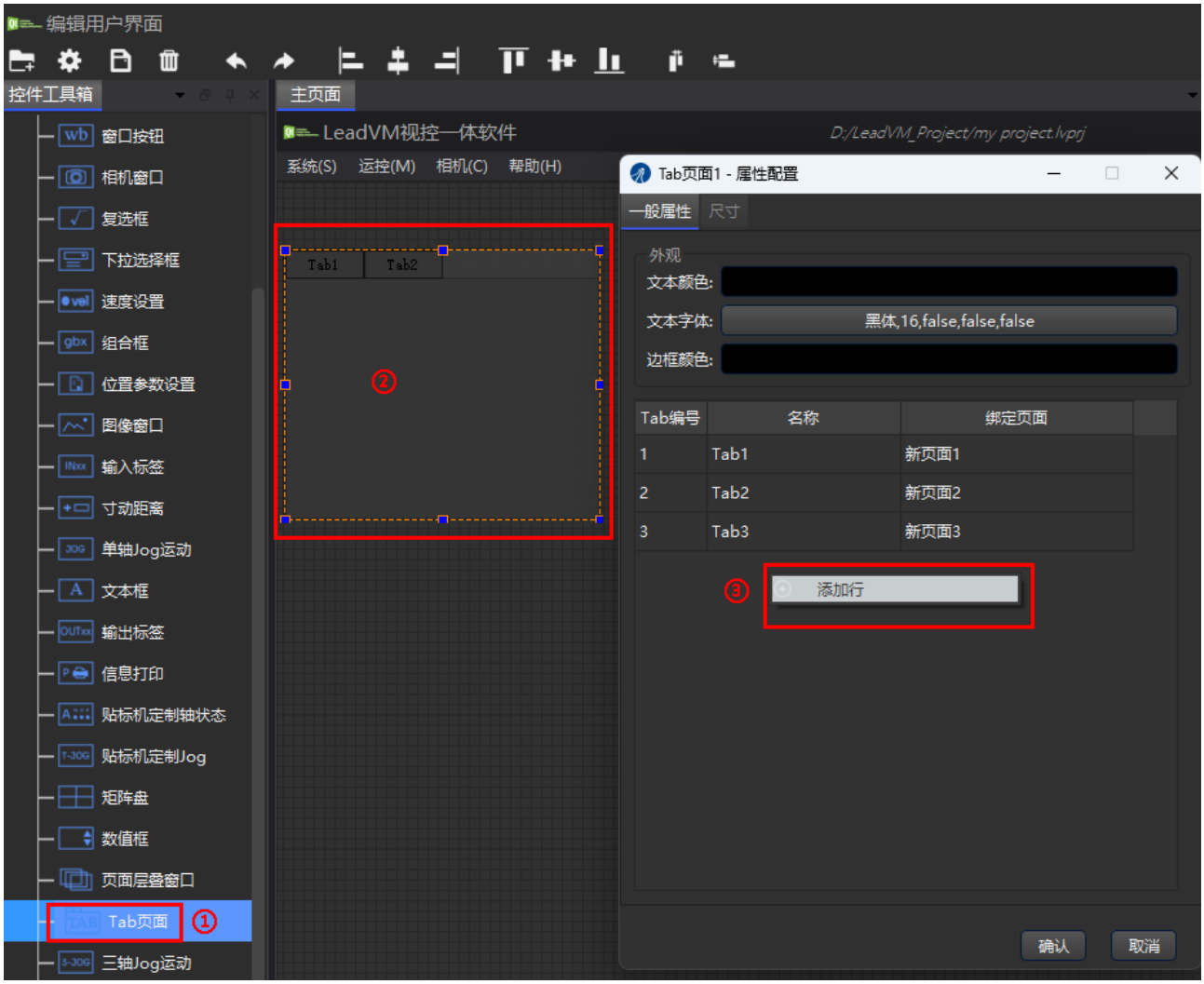
在控件工具箱中，找到“组合框”，拖拽添加其中的“组合框”。双击打开“组合框”的属性窗口，在一般属性页面可配置“组合框”的名称、背景色等功能。



5.4.7 Tab 页面

Tab 页面提供了一个选项卡窗口，可以用于创建多个选项卡，并在每个选项卡中显示不同的内容。

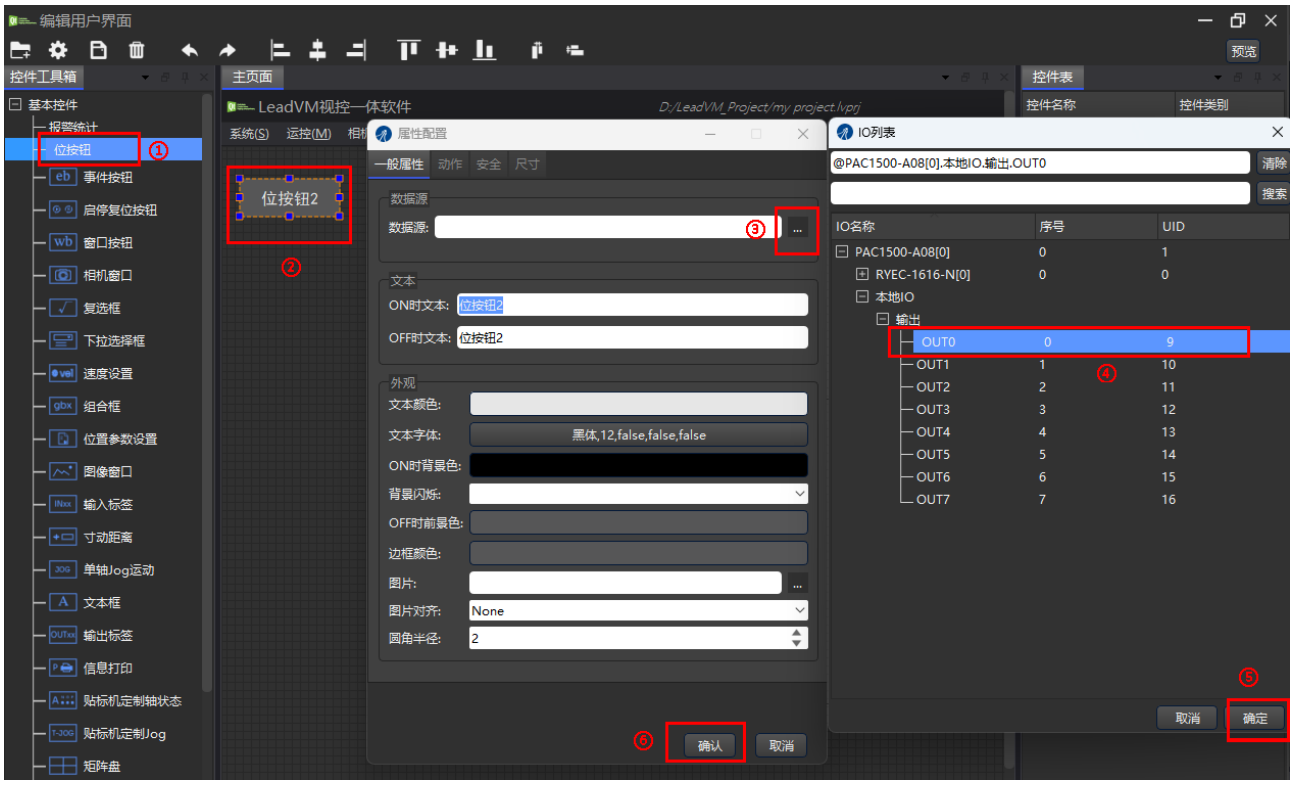
在控件工具箱中，找到①“Tab 页面”，拖拽添加其中的“Tab 页面”。双击打开②“Tab 页面”的属性窗口，在一般属性页面可为其添加多个选项卡及绑定显示的页面，右击③“添加行”，配置新增 Tab 选项卡的名称及页面。如下图所示。



5.4.8 位按钮

位按钮允许用户通过单击来执行操作，用于控制输出口的状态。在按钮控件被单击时，可以通过控件的动作属性选择其触发行为，可实现位反转、自复位功能。当使用自复位功能时，自复位时间由用户设定，范围[1,10000]ms。位按钮支持设置文本颜色、字体、背景色、尺寸、安全等功能。

在控件工具箱中，找到①“位按钮”，左击将其拖拽到页面编辑区并释放，位按钮即添加到页面中。双击②“位按钮”打开属性窗口，在属性窗口中选择③“数据源”，打开数据源订阅窗口，选择一个输出口，如订阅④“PAC1500-A08”的本地 IO 的 OUT0，完成后点击⑤“确定”，关闭订阅窗口，最后点击⑥“确认”，完成数据源的订阅。如下图所示。



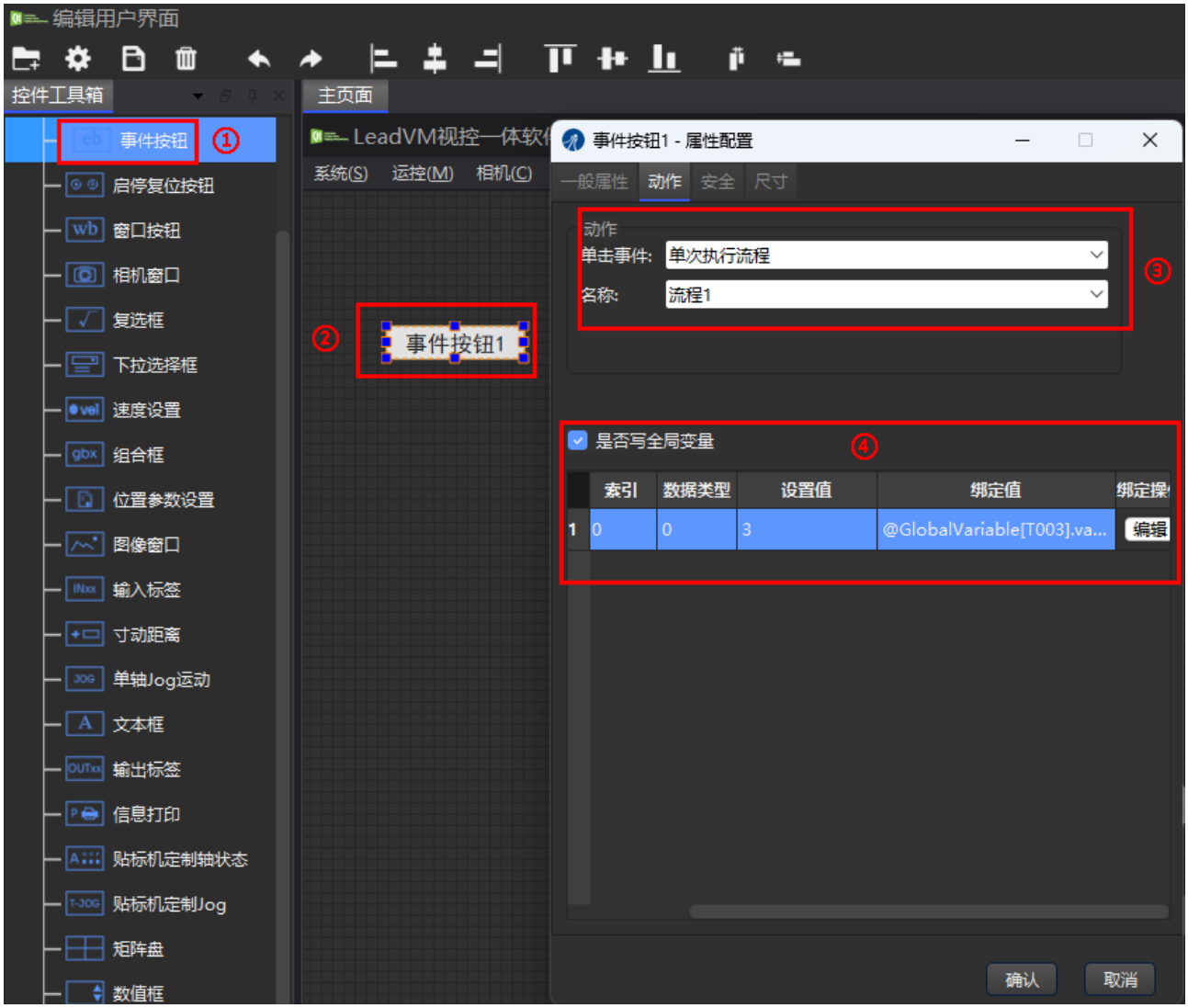
在“位按钮”属性窗口的动作页面，用户还需配置“位按钮”的动作行为，如下图所示。包括位取反（输出 ON/OFF 切换）、自复位（输出 ON，延时指定时间后，输出 OFF，单位 ms，范围[1,10000]）。



5.4.9 事件按钮

事件按钮允许用户通过单击来执行操作，在按钮控件被单击时，可以通过控件的动作属性选择其触发行为，可实现执行流程、执行工具等功能。事件按钮支持设置文本颜色、字体、背景色、尺寸、安全等功能。

在控件工具箱中，找到“事件按钮”，拖拽添加其中的“事件按钮”。双击打开“事件按钮”的属性窗口，在动作页面可配置“事件按钮”的动作行为及是否写入全局变量的值。



事件按钮的动作行为如下：

（1）执行指定工具：可执行一次指定的工具，工具正处于执行状态时，执行失败。

（2）重置工具：复位工具值，重新执行，如计数器工具的当前值。

（3）单次执行流程：执行一次指定的流程或全部流程。

（4）连续执行流程：连续执行指定的流程或全部流程。

（5）停止流程：停止执行指定的流程或全部流程。

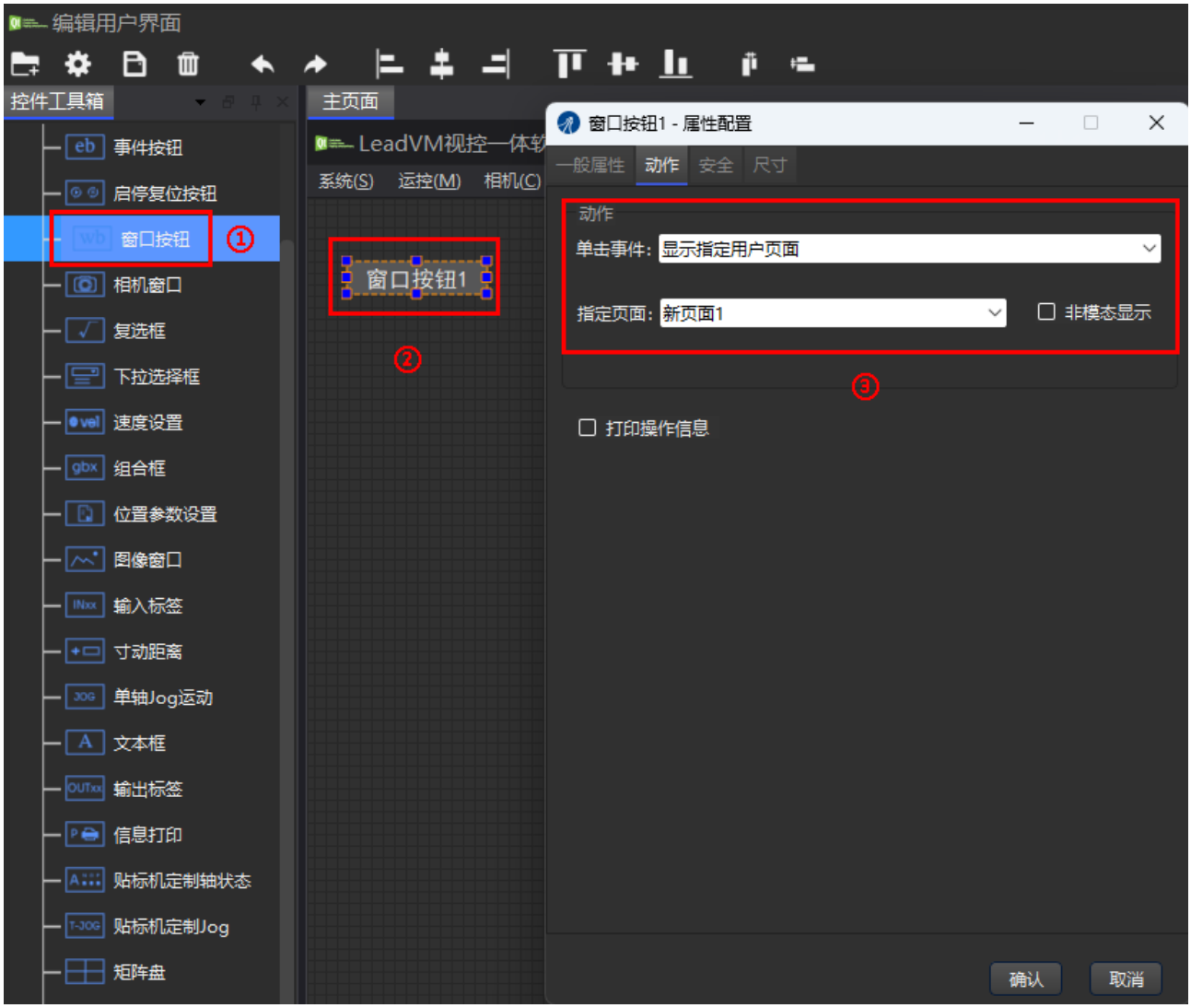
（6）继续流程：继续执行指定的流程或全部流程。

（7）复位流程：复位指定的流程或全部流程。

5.4.10 窗口按钮

窗口按钮允许用户通过单击来执行操作，在按钮控件被单击时，可以通过控件的动作属性选择其触发行为，可实现打开指定页面、打开工具页面等功能。窗口按钮支持设置文本颜色、字体、背景色、尺寸、安全等功能。

在控件工具箱中，找到“窗口按钮”，拖拽添加其中的“窗口按钮”。双击打开“窗口按钮”的属性窗口，在动作页面可配置“窗口按钮”的动作行为。



窗口按钮的动作行为如下：

（1）打开权限登入画面：用于登入权限，该画面打开后处于垄断状态。

（2）打开相机管理画面：用于修改相机参数，非垄断状态，可同时操作底部画面和打开的画面。

（3）打开全局变量画面：用于查看全局变量的当前值及修改初始值，非垄断状态，可同时操作底部画面和打开的画面。

（4）打开指定工具对话框：用于修改流程中的工具块参数，非垄断状态，可同时操作底部画面和打开的对话框。

（5）显示指定用户画面：用户界面组态多个页面时，通过该功能切换组态的页面，勾选非模态显示，即为垄断状态。



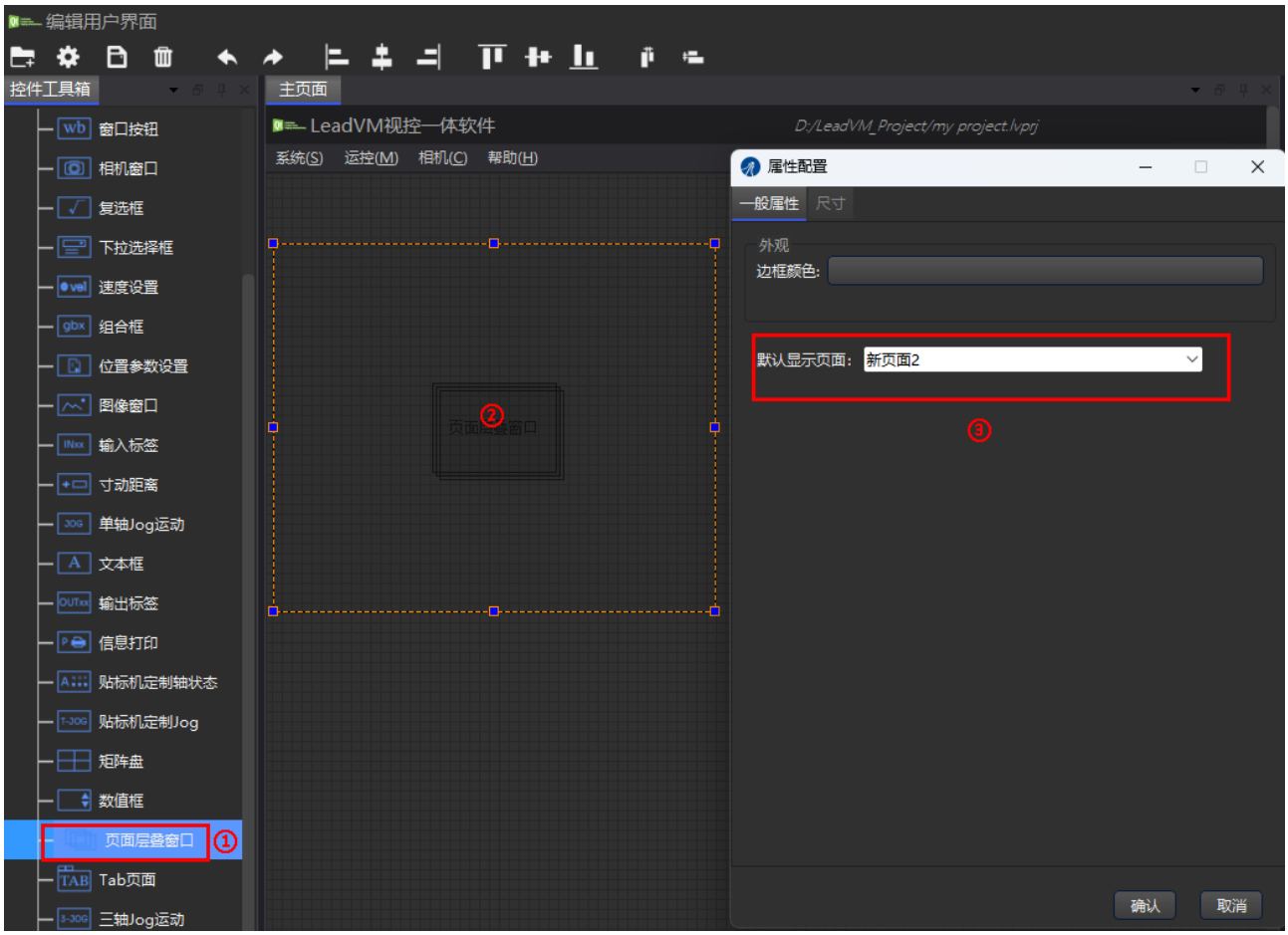
（6）切换层叠窗口画面：在用户界面的共有区域，切换显示组态画面，需配合控件“页面层叠窗口”使用。



5.4.11 页面层叠窗口

用于在一个窗口的公共区域显示多个页面的内容。层叠窗口需配置一个默认显示页面，在层叠窗口显示的页面，其尺寸不能超过层叠窗口的尺寸，否则不能完整显示。切换显示页面时，需配合窗口按钮控件的切换层叠窗口功能。

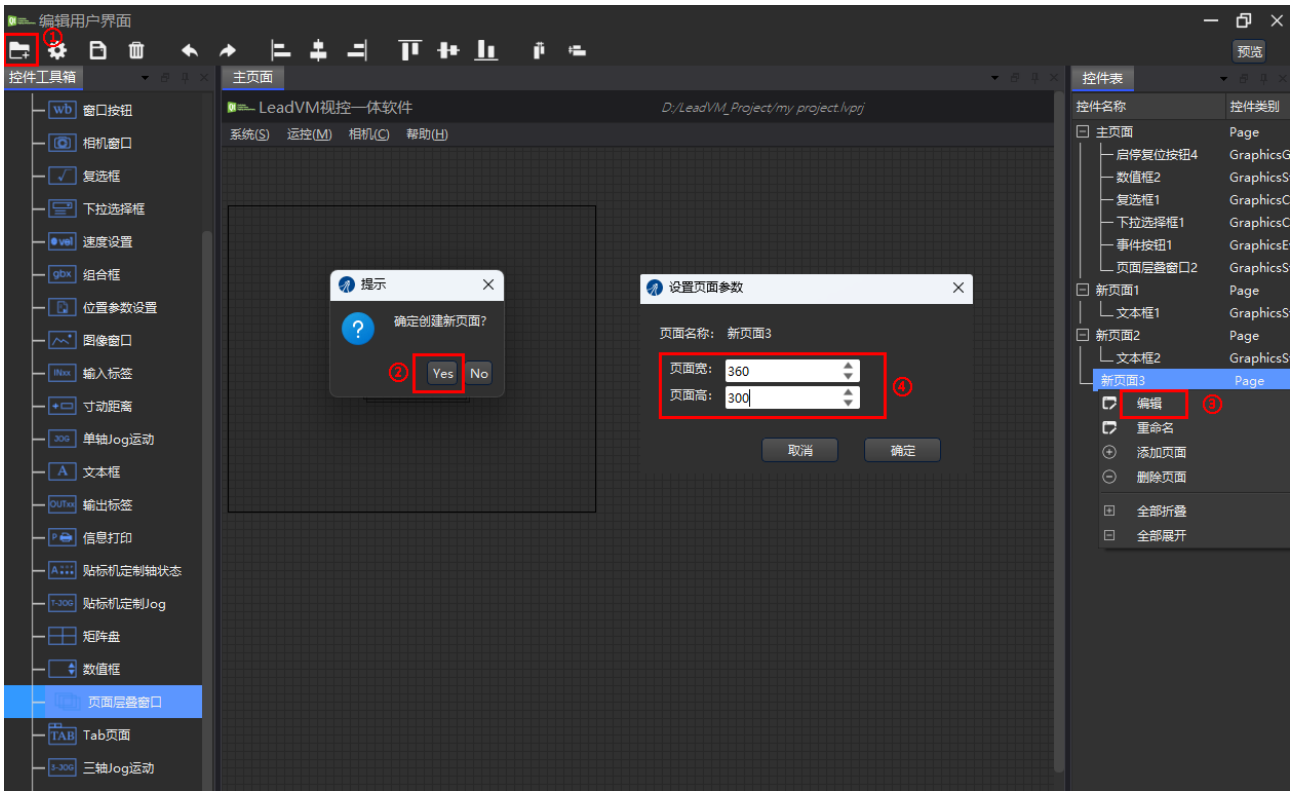
在控件工具箱中，找到“页面层叠窗口”，拖拽添加其中的“页面层叠窗口”。双击打开“页面层叠窗口”的属性窗口，在一般属性页面可配置“页面层叠窗口”的默认显示页面，如新页面 2。在尺寸页面配置“页面层叠窗口”的尺寸，如宽度 360、高度 300。如下图所示。



下文继续说明在页面层叠窗口中，如何实现从默认页面的新页面 2 切换到新页面 3。

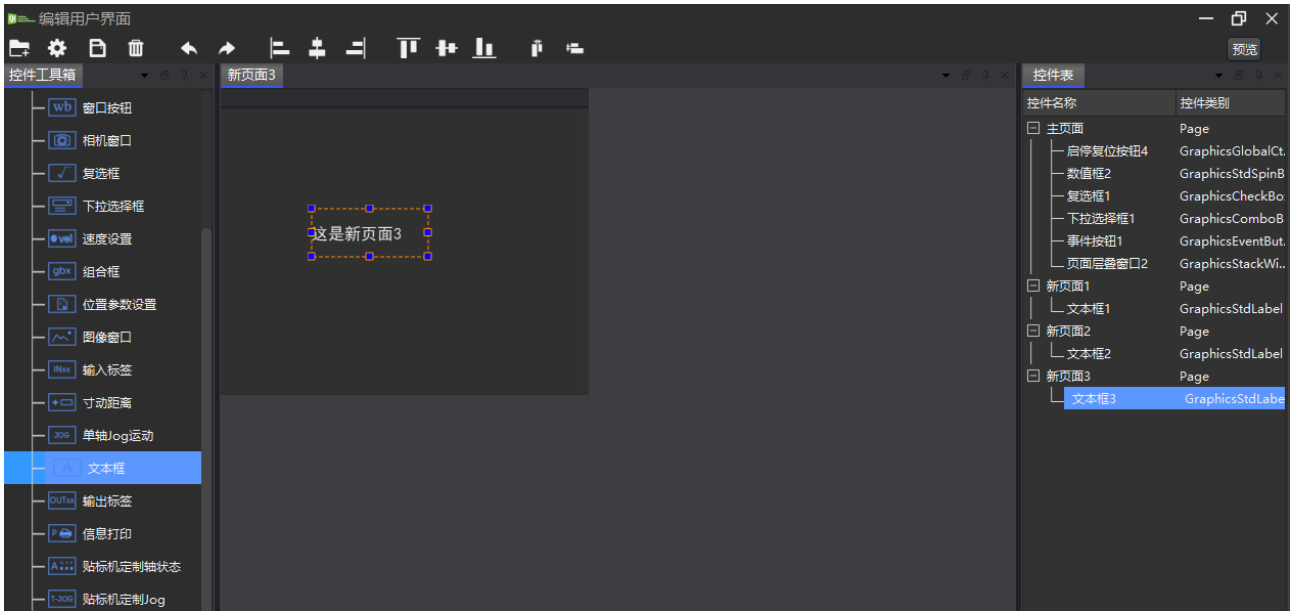
步骤 1：新建页面 3

在用户界面的工具栏点击①“创建新页面”，在弹窗中点击②“OK”，在右侧控件列表中可看找到新页面 3，右击选择③“编辑”打开④“设置页面参数”窗口，设置新页面 3 的宽 360、高 300，与“页面层叠窗口”的尺寸保持一致。点击“确认”后完成新页面 3 的创建。



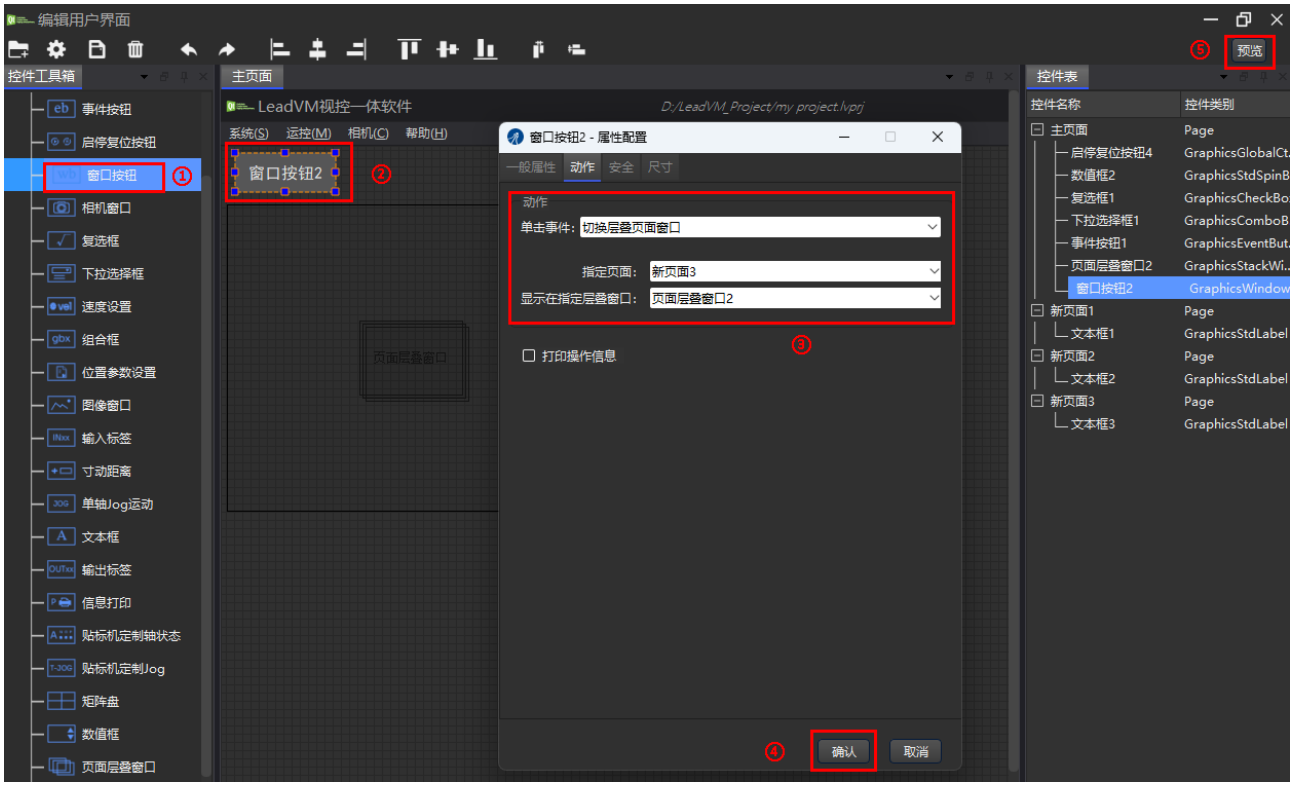
步骤 2：在新页面中添加控件

为了显示切换的效果，新页面 2 与新页面 3 分别添加文本控件，文本内容为“这是新页面 2”和“这是新页面 3”。如下图所示，在新页面 3 中添加文本控件。



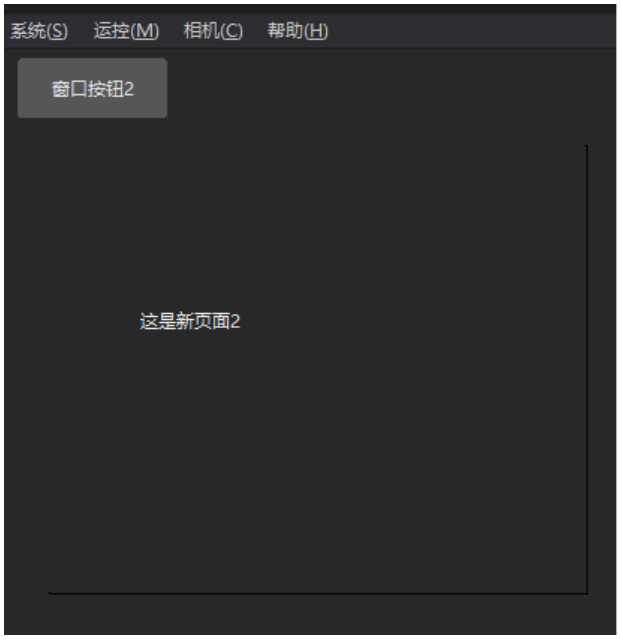
步骤 3：主页面添加并配置窗口按钮

返回添加了“页面层叠窗口”的主页面，在控件工具箱中，找到“窗口按钮”，拖拽添加其中的“窗口按钮”。双击打开“窗口按钮”的属性窗口，在动作页面配置“窗口按钮”的动作如下图所示。完成后点击⑤“预览”切换到运行界面的预览窗口。



步骤 4：在预览窗口操作层叠页面切换

在预览窗口点击“窗口按钮 2”，实现新页面 2 切换到新页面 3 的效果，即从“这是新页面 2”切换为“这是新页面 3”。实现效果如下图所示。





5.4.12 运行控制

用于控制项目全部流程的运行，支持复位、启动、暂停、停止流程，其含义如下：

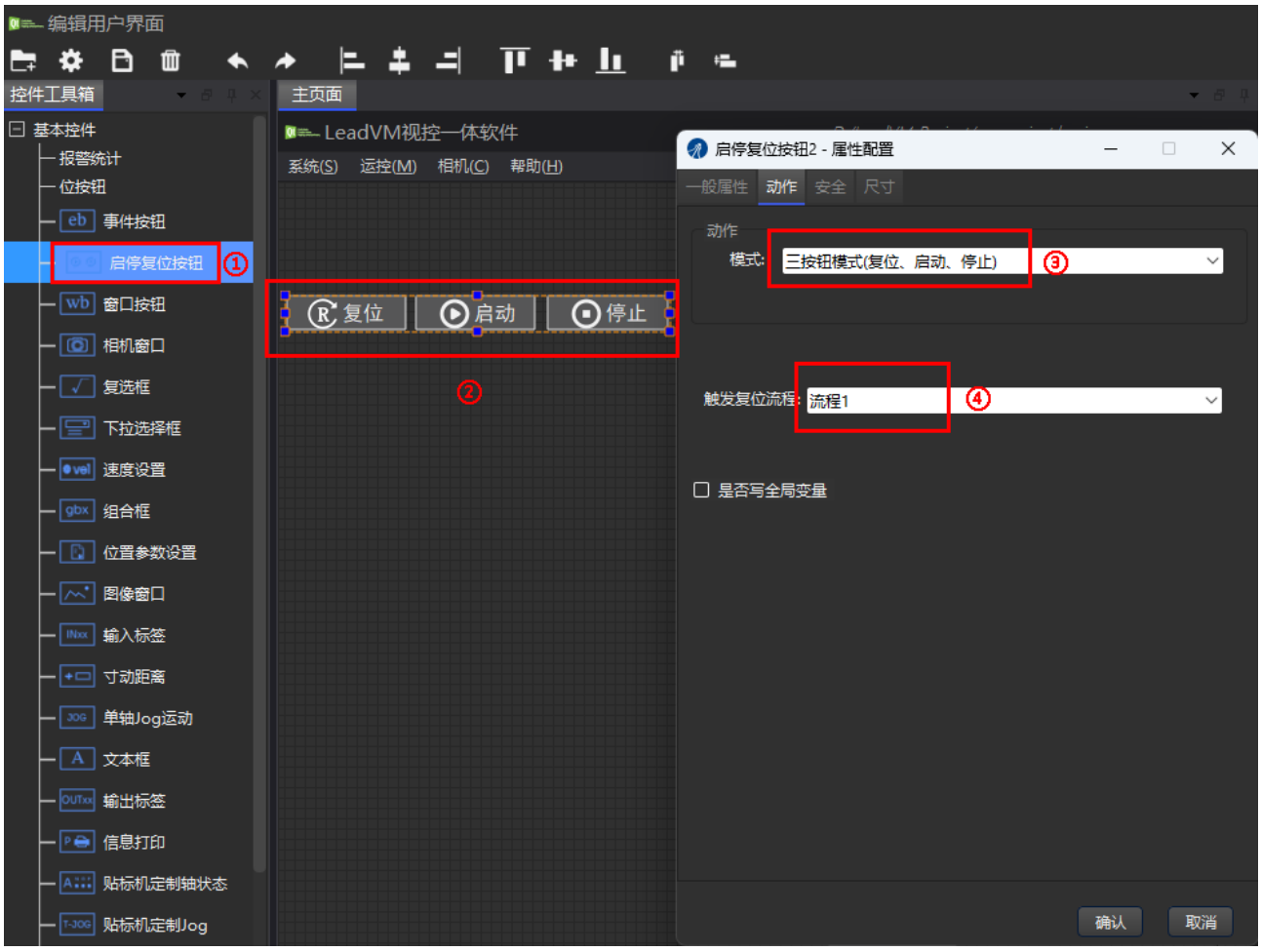
复位：先停止所有流程（包括循环流程和触发流程），再将所有流程复位到初始状态，最后执行用户指定的复位触发流程（一般用户编写机台复位动作）。

启动：启动所有循环执行的流程，包括循环流程和处于暂停状态的流程。

暂停：暂停所有流程，包括循环流程和触发执行的流程。如果暂停了启动流程，可通过启动按钮继续执行，也可通过复位按钮，执行复位流程。如果暂停了复位流程，可通过复位按钮继续执行，但是不能执行启动流程。

停止：停止所有流程，包括循环流程和触发执行的流程。停止后，必须重新执行复位流程。

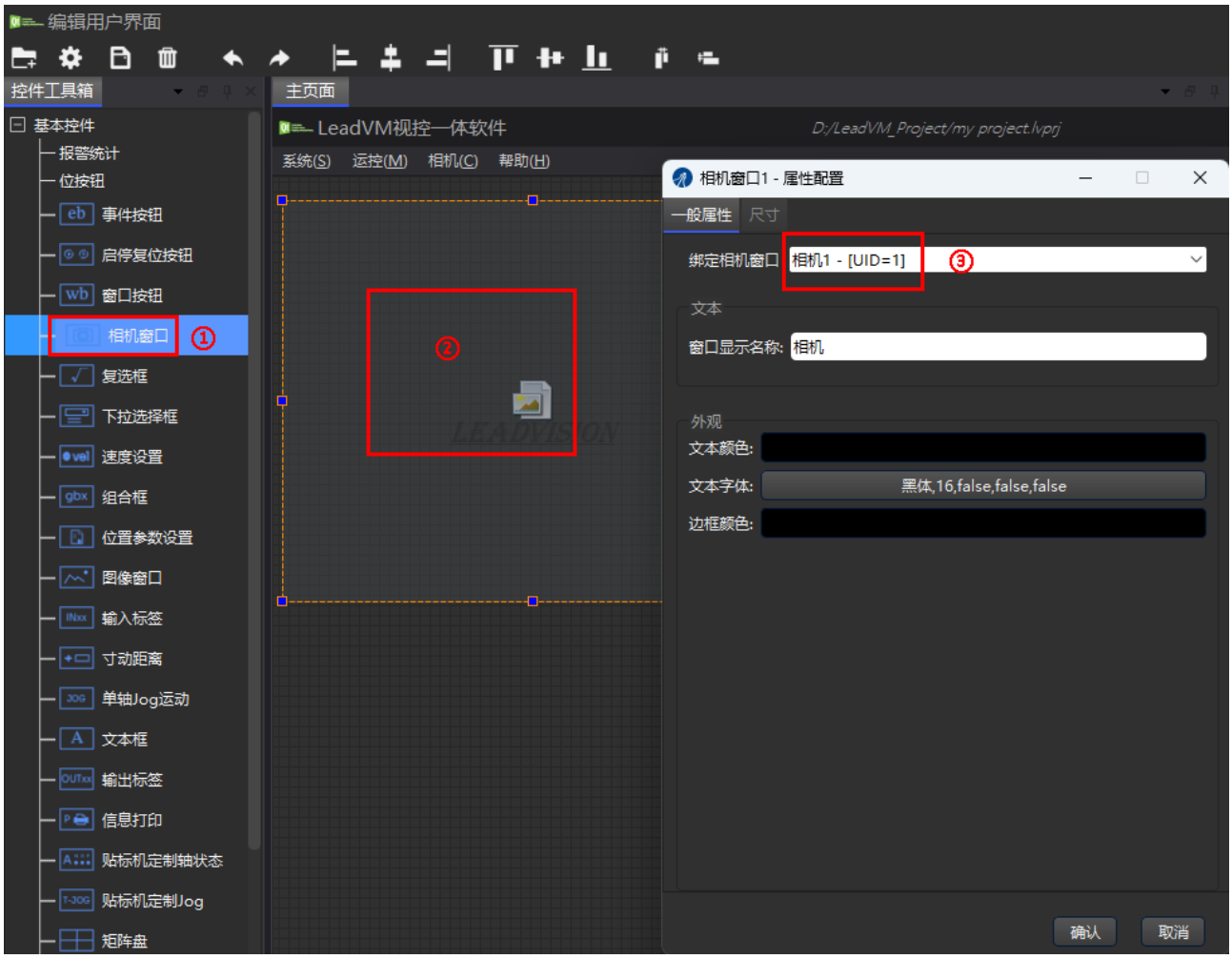
在控件工具箱中，找到“运行控制”，拖拽添加其中的“运行控制”。双击打开“运行控制”的属性窗口，在动作页面可配置“运行控制”的动作行为及是否写入全局变量的值。



5.4.13 相机窗口

用于显示相机实时画面，需绑定相机源。

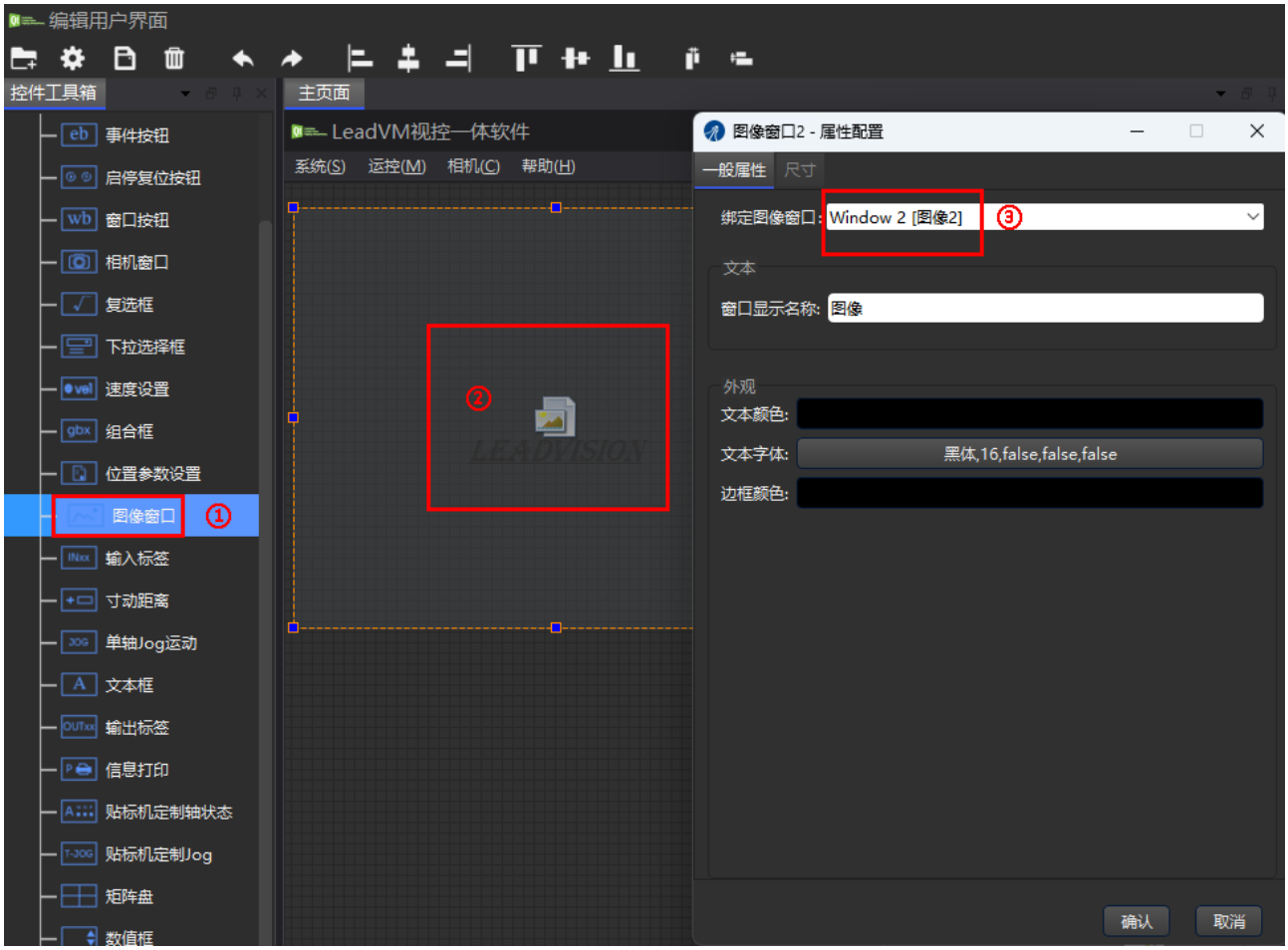
在控件工具箱中，找到“相机窗口”控件，拖拽添加其中的“相机窗口”。双击打开“相机窗口”的属性窗口，在一般属性页面可配置“相机窗口”的相机源、窗口名称、文本颜色、字体等功能。



5.4.14 图像窗口

用于显示流程工具块中的图像画面及图像中的 ROI，需绑定图像窗口源。

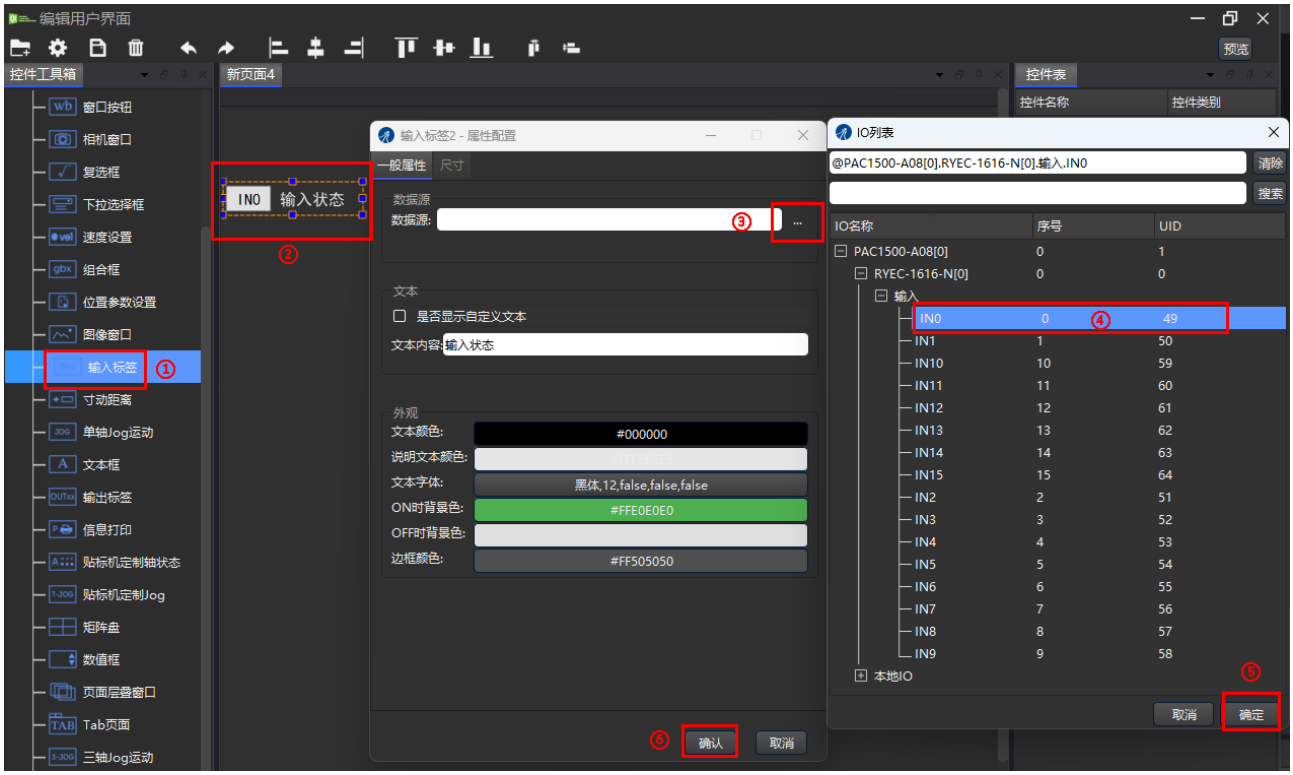
在控件工具箱中，找到“图像窗口”控件，拖拽添加其中的“图像窗口”。双击打开“图像窗口”的属性窗口，在一般属性页面可配置“图像窗口”的图像源、窗口名称、文本颜色、字体等功能。



5.4.15 输入标签

用于输入口的状态监控。

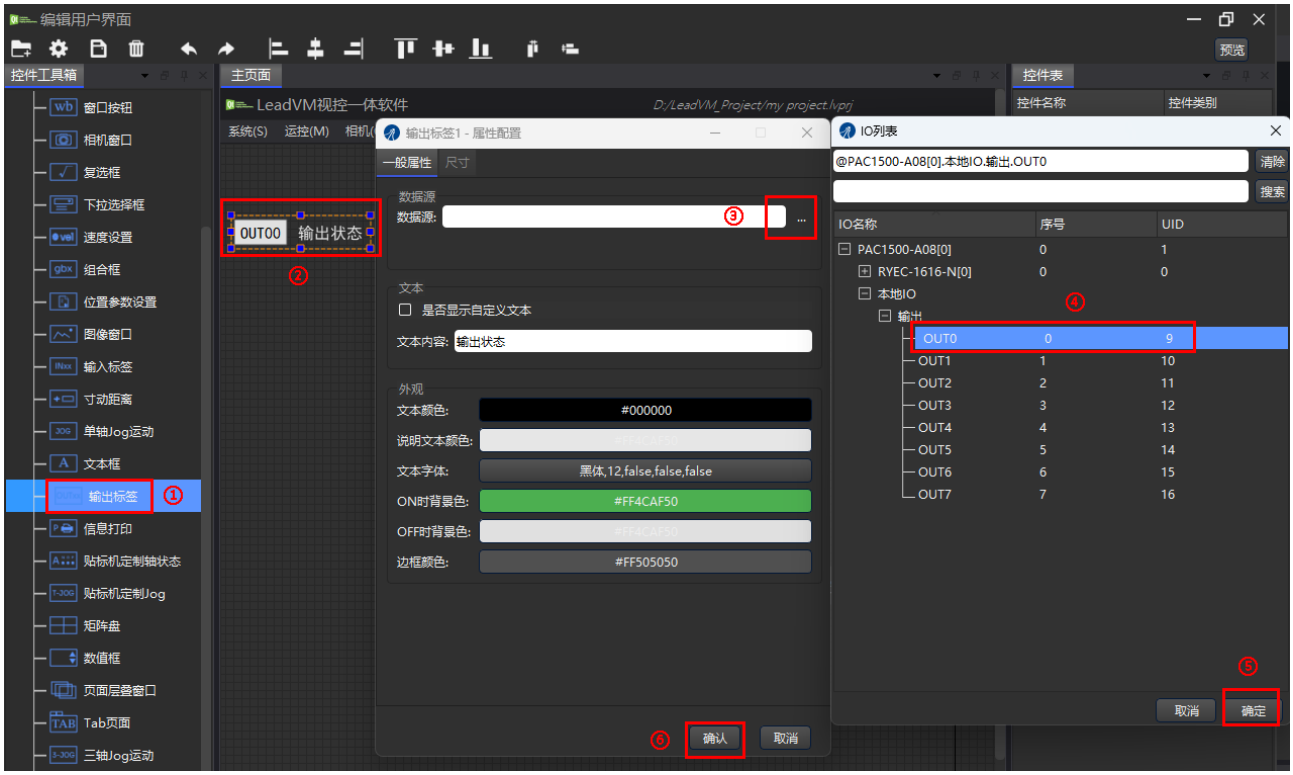
在控件工具箱中，找到“输入标签”，左击将其拖拽到页面编辑区并释放，输入标签即添加到页面中。双击②“输入标签”打开属性窗口，在属性窗口中选择③“数据源”，打开数据源订阅窗口，如订阅④“RYEC 模块的 IN0”输入口，完成后点击⑤“确定”，关闭订阅窗口，最后点击⑥“确认”，完成数据源的订阅。如下图所示。



5.4.16 输出标签

用于输出口的控制及状态监控。

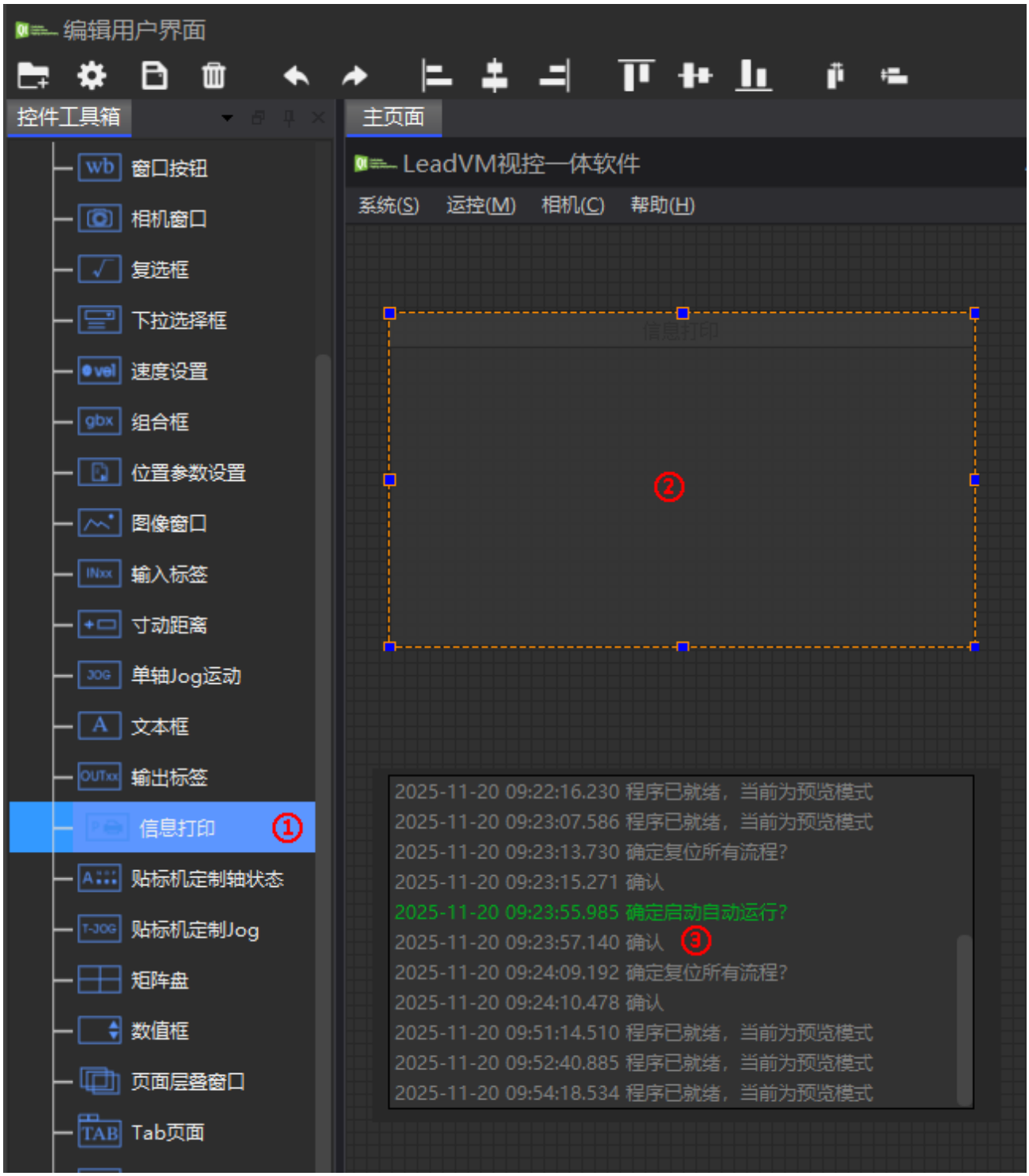
在控件工具箱中，找到“输出标签”，左击将其拖拽到页面编辑区并释放，输出标签即添加到页面中。双击②“输出标签”打开属性窗口，在属性窗口中选择③“数据源”，打开数据源订阅窗口，如订阅④“PAC1500-A08”的 OUT0 口，完成后点击⑤“确定”，关闭订阅窗口，最后点击⑥“确认”，完成数据源的订阅。如下图所示。



5.4.17 信息打印

打印用户操作的信息、报警、错误、及流程工具“信息提示”push 出来的信息，无需配置参数。

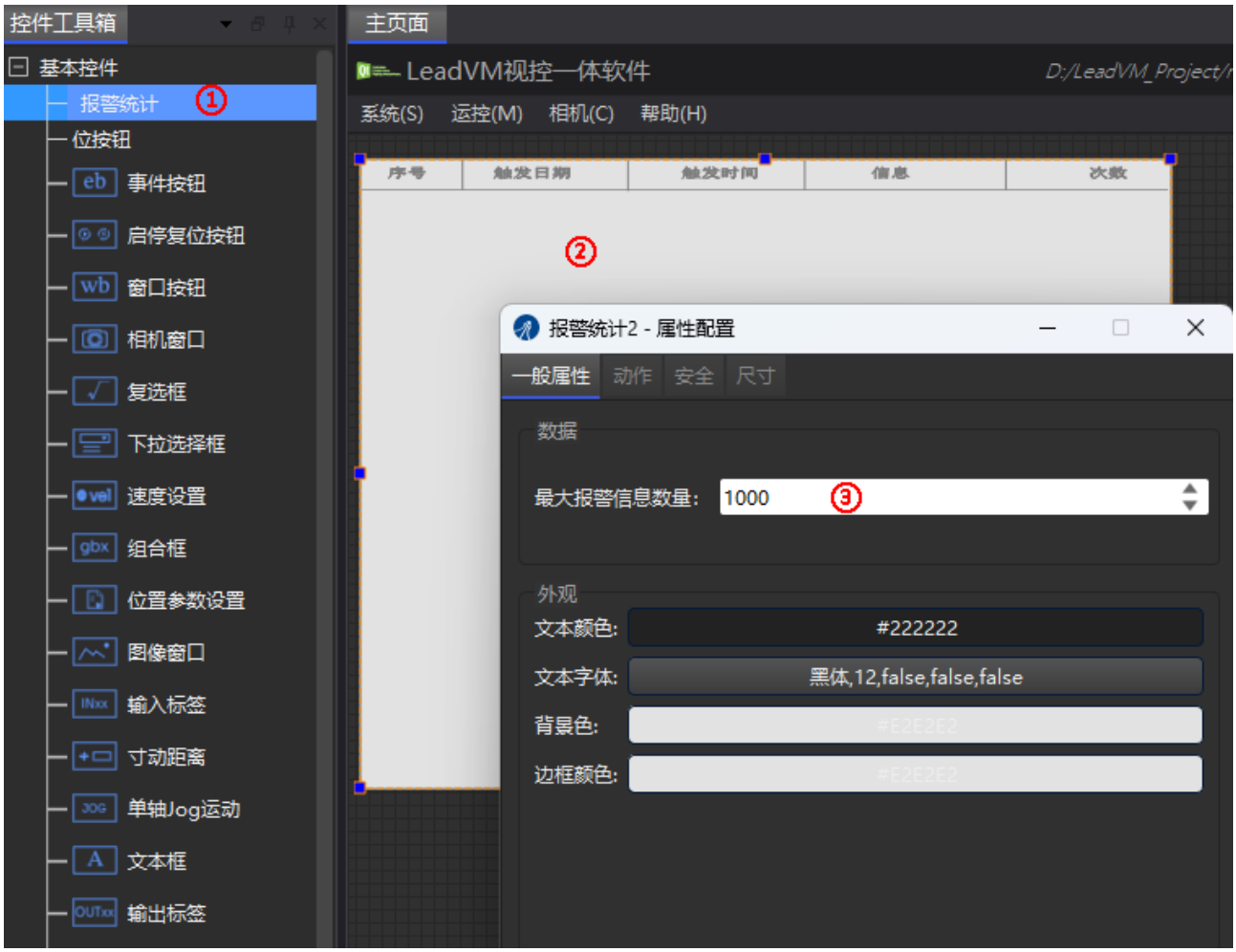
在控件工具箱中，找到①“信息打印”，左击将其拖拽到页面编辑区并释放，②信息打印即添加到页面中，调整到合适尺寸后预览，③为预览效果，如下图所示。



5.4.18 报警统计

记录流程工具“信息提示”push 出来的报警信息，每条报警产生的时间、信息、次数。并可导出报警记录。最大报警信息 1000 条，用户可设置 1~1000 条内的报警显示，设置后按设置的条数统计，大于该值后重新统计。也可手动清除报警统计。

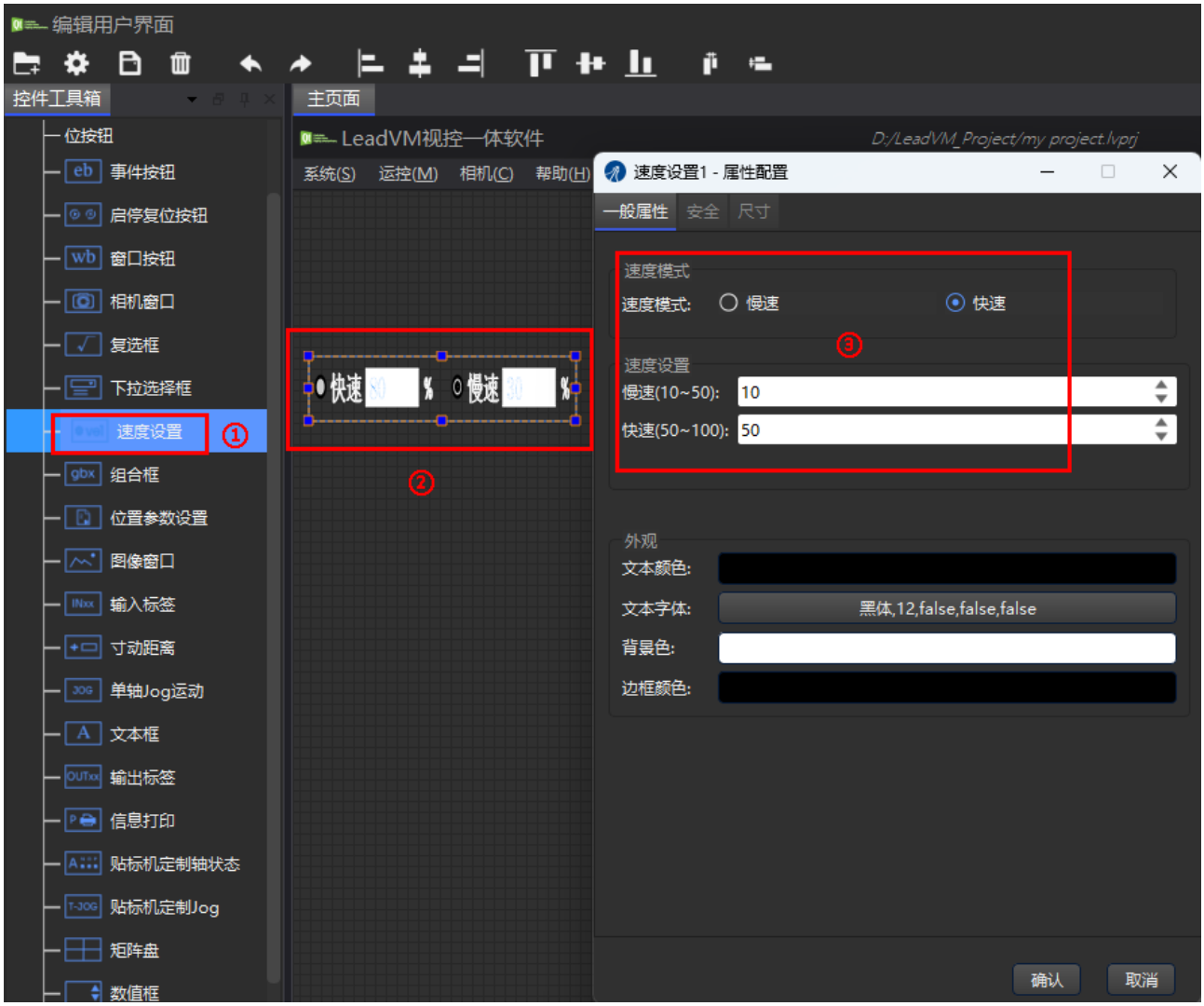
在控件工具箱中，找到①“报警统计”，左击将其拖拽到页面编辑区并释放，②报警统计即添加到页面中，双击打开属性页面，在一般属性页配置最大报警信息即完成配置，如下图所示。



5.4.19 速度设置

用于 Jog 运动速度设置，速度百分比为章节 3.1.2 配置轴页面设置的速度的百分比，慢速速度百分比范围 0.01%~20%，快速速度百分比范围 0.01%~50%。本控件需配合单轴 Jog 运动、两轴 Jog 运动、三轴 Jog 运动控件使用。

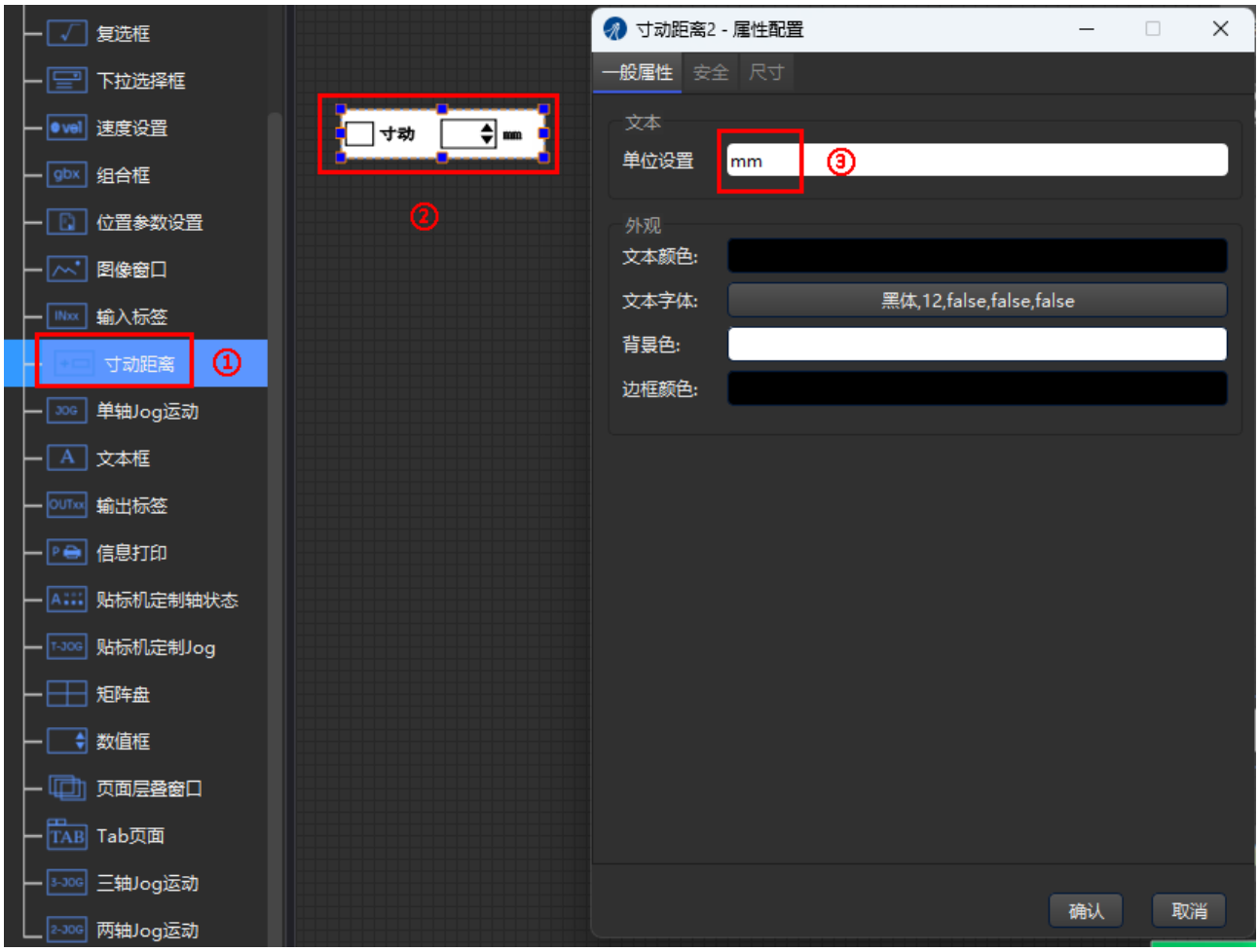
在控件工具箱中，找到①“速度设置”，左击将其拖拽到页面编辑区并释放，②速度设置即添加到页面中，在一般属性页面可配置“速度设置”的速度比分比及默认速度模式。如下图所示。



5.4.20 寸动距离

用于 Jog 运动位置设置，本控件需配合单轴 Jog 运动、两轴 Jog 运动、三轴 Jog 运动控件使用。

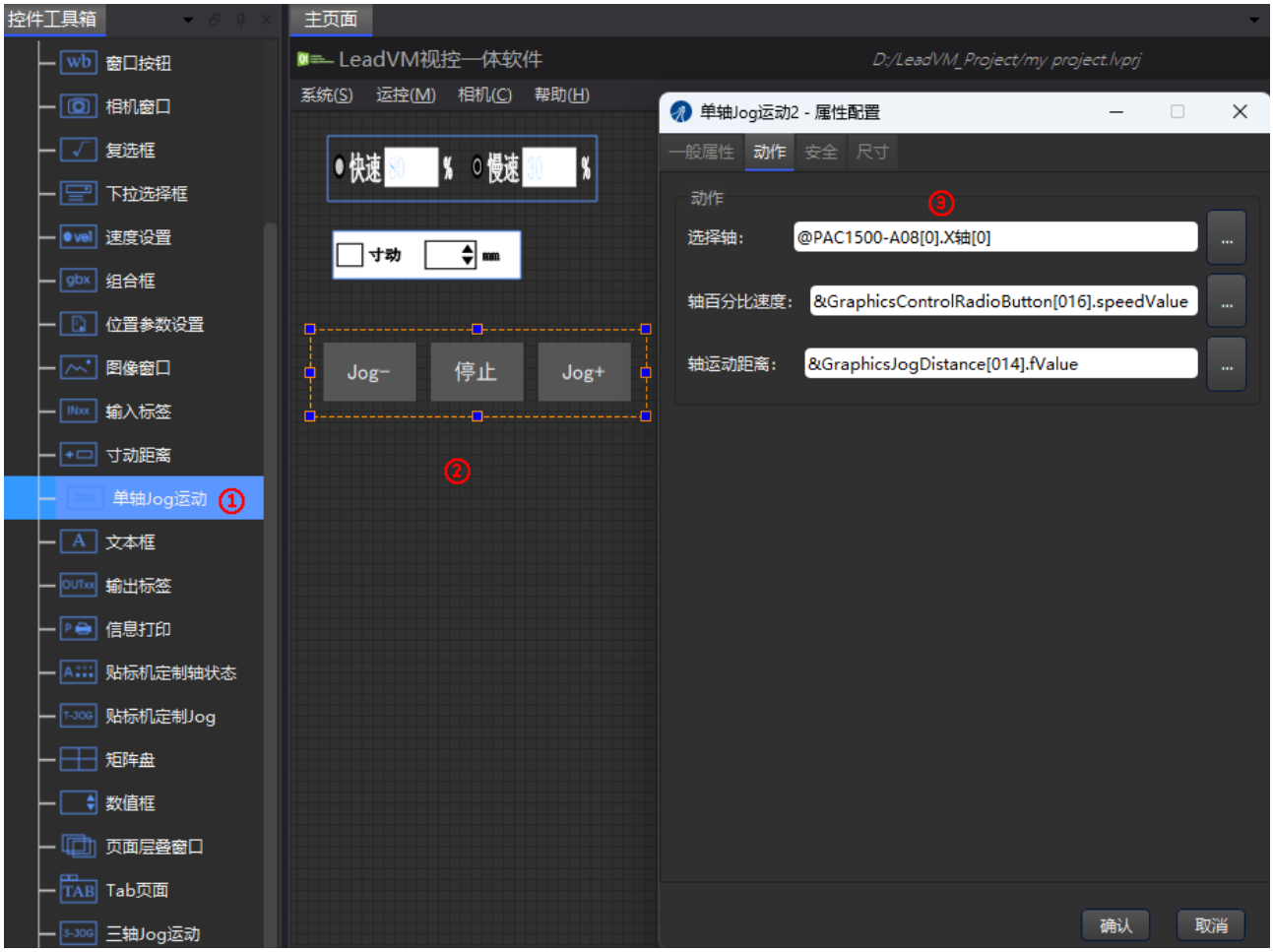
在控件工具箱中，找到①“寸动距离”，左击将其拖拽到页面编辑区并释放，②寸动距离即添加到页面中，在一般属性页面可配置“寸动距离”的显示单位。如下图所示。



5.4.21 单轴 Jog 运动

用于单轴点动运动操作，一般与速度设置和寸动距离控件配合使用，点动运动的速度及最大寸动距离可选择速度设置和寸动距离控件的输出。

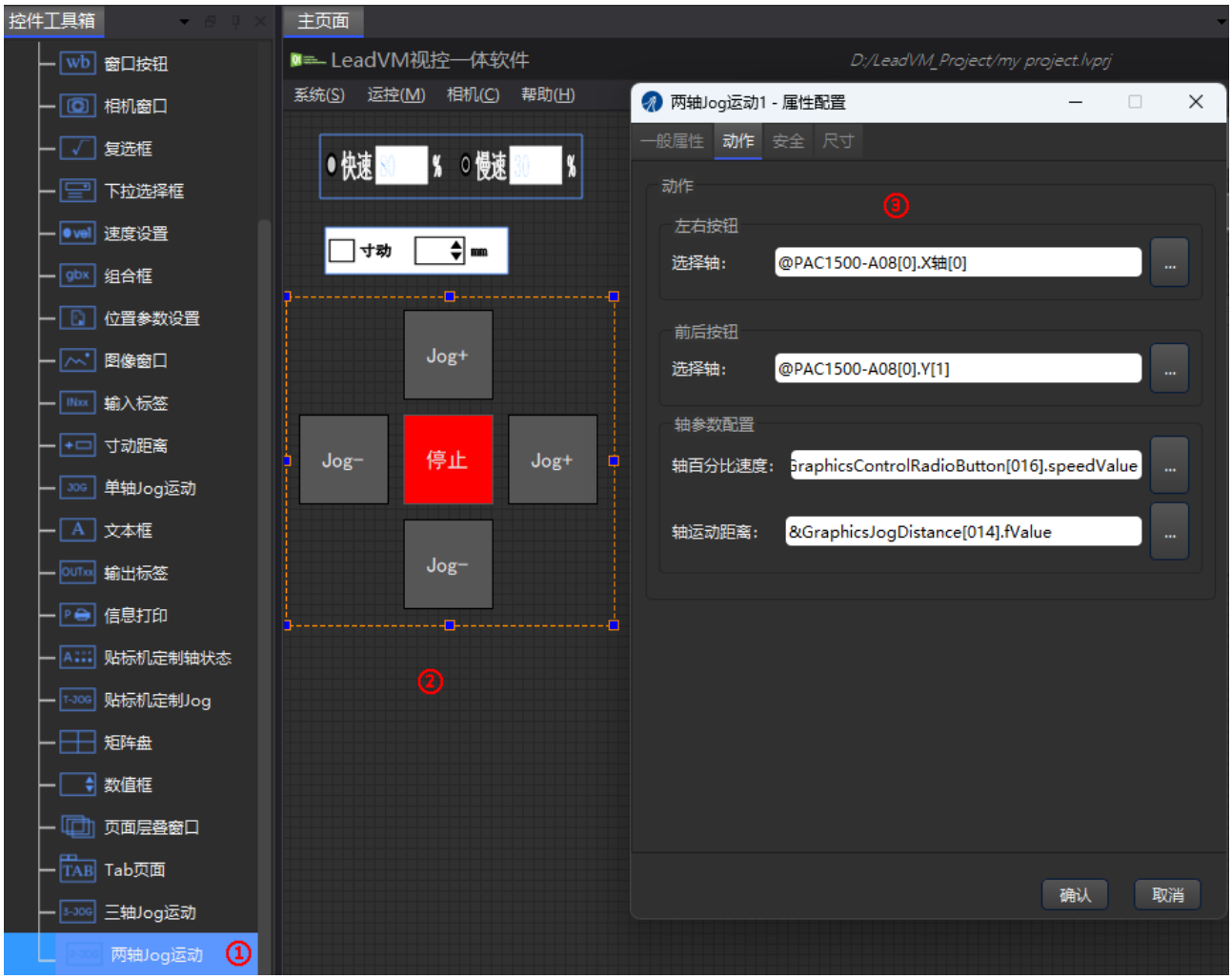
在控件工具箱中，找到①“单轴 Jog 运动”，左击将其拖拽到页面编辑区并释放，②单轴 Jog 运动即添加到页面中，在动作属性页面可订阅 Jog 运动的轴号、速度百分比和寸动距离。如下图所示。



5.4.22 两轴 Jog 运动

用于两轴点动运动操作，一般与速度设置和寸动距离控件配合使用，点动运动的速度及最大寸动距离可选择速度设置和寸动距离控件的输出。

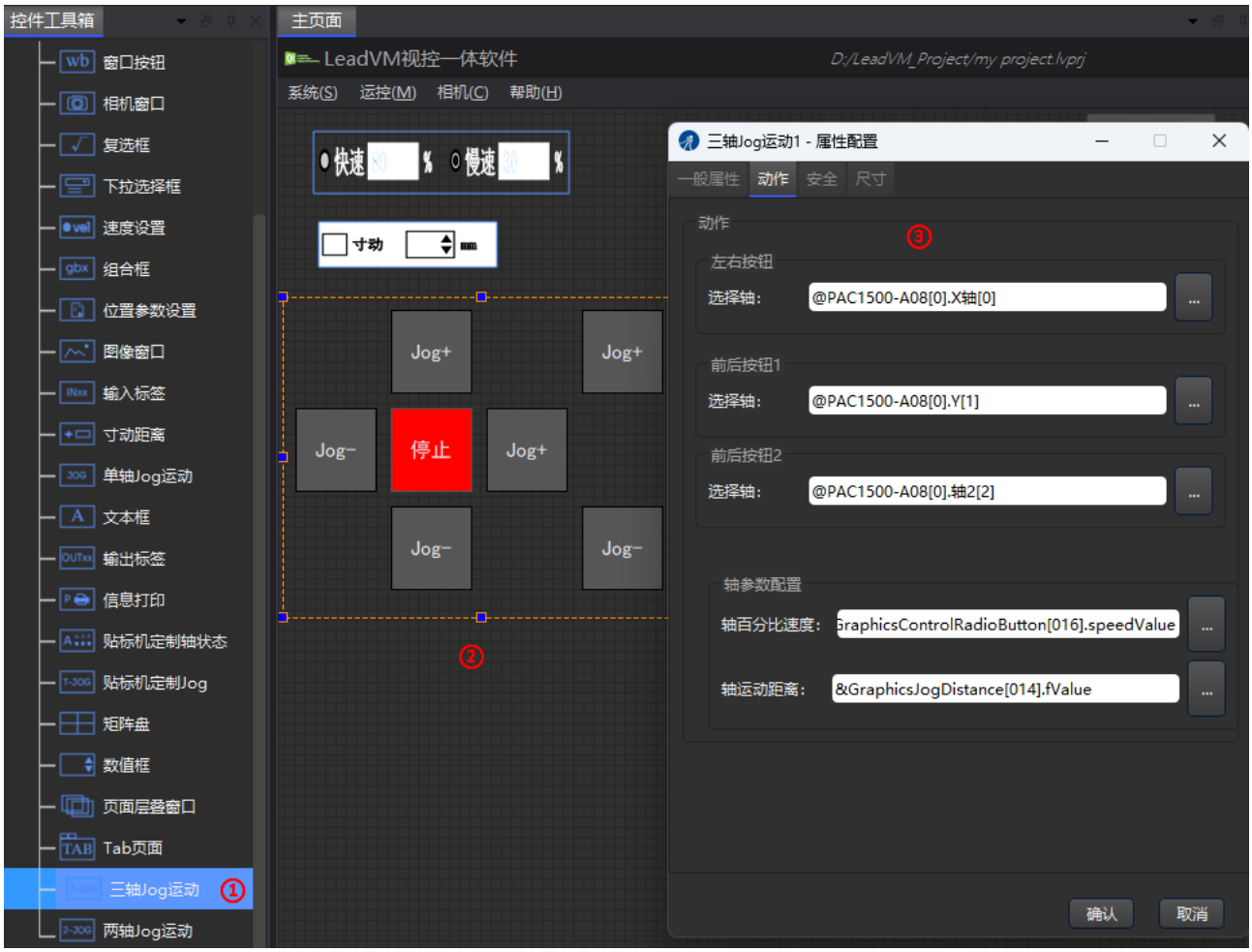
在控件工具箱中，找到①“两轴 Jog 运动”，左击将其拖拽到页面编辑区并释放，②两轴 Jog 运动即添加到页面中，在动作属性页面可订阅 Jog 运动的轴号、速度百分比和寸动距离。如下图所示。



5.4.23 三轴 Jog 运动

用于三轴点动运动操作，一般与速度设置和寸动距离控件配合使用，点动运动的速度及最大寸动距离可选择速度设置和寸动距离控件的输出。

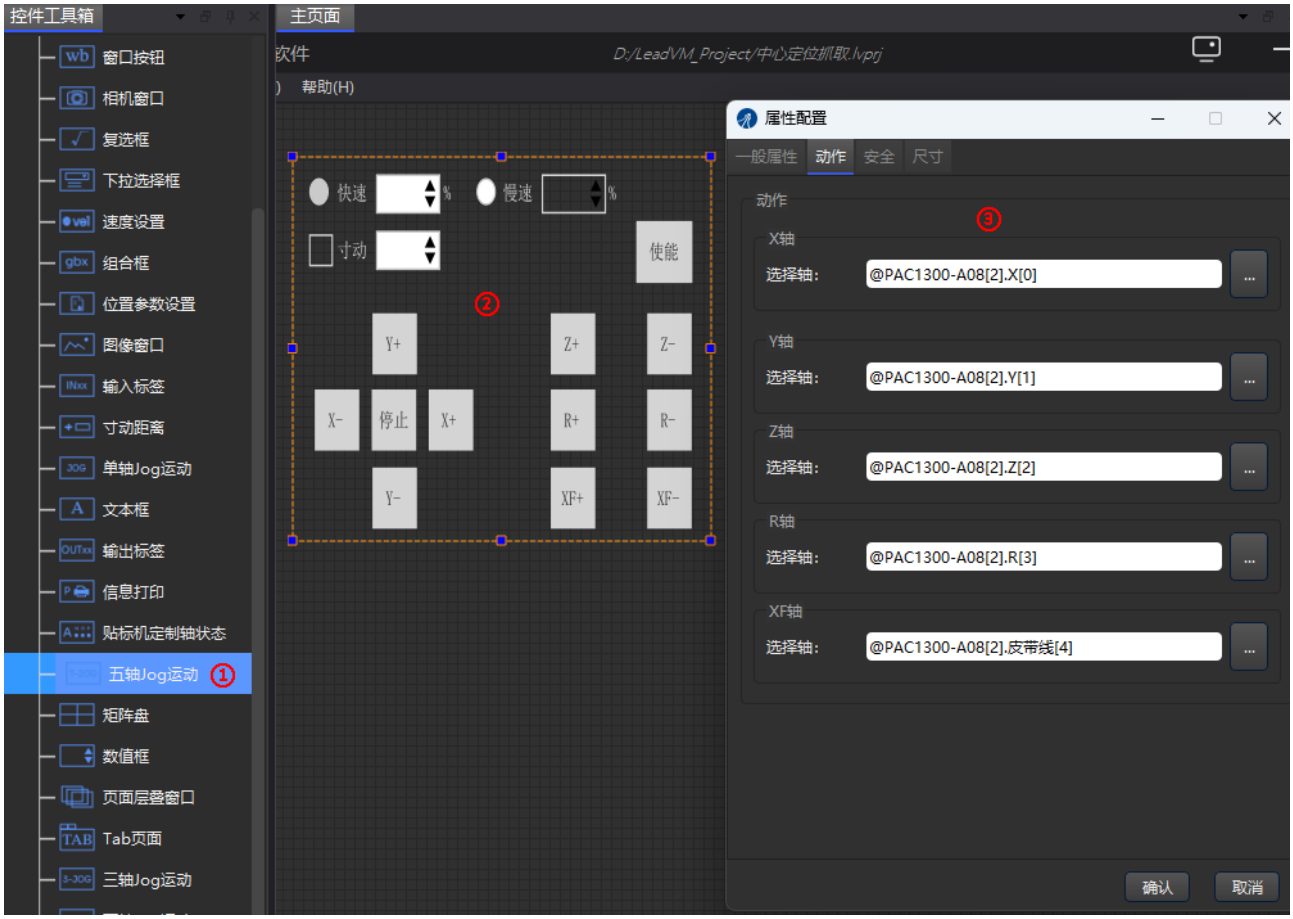
在控件工具箱中，找到①“三轴 Jog 运动”，左击将其拖拽到页面编辑区并释放，②三轴 Jog 运动即添加到页面中，在动作属性页面可订阅 Jog 运动的轴号、速度百分比和寸动距离。如下图所示。



5.4.24 五轴 Jog 运动

用于五轴点动运动操作，集成速度设置控件和寸动距离控件功能，点动运动的速度百分比为章节 3.1.2 配置轴页面设置的速度的百分比。慢速速度百分比范围 0.01%~20%，快速速度百分比范围 0.01%~50%。

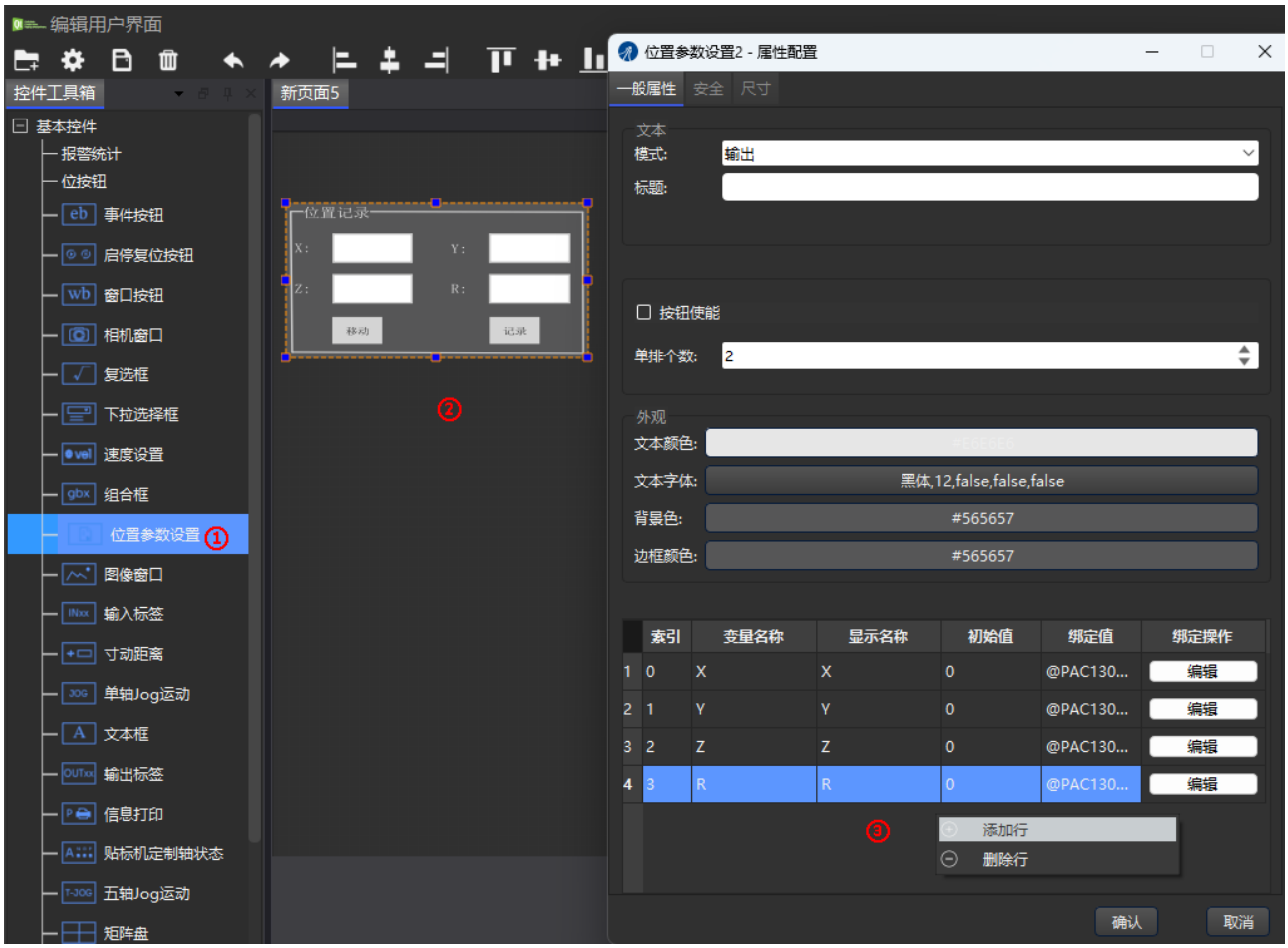
在控件工具箱中，找到①“五轴 Jog 运动”，左击将其拖拽到页面编辑区并释放，②五轴 Jog 运动即添加到页面中，在动作属性页面可订阅 Jog 运动的轴号。如下图所示。



5.4.25 位置记录

用于示教轴的指令位置，可直接通过记录按钮示教轴当前指令位置，也可手动修改位置值，通过移动按钮，可控制各轴移动到示教位置。轴数量通过列表增加，需绑定每个轴号。单排个数用于控制单排显示的轴个数。

在控件工具箱中，找到①“位置记录”，左击将其拖拽到页面编辑区并释放，②位置记录即添加到页面中，在一般属性页面可添加轴、绑定轴号、修改名称。如下图所示。



6. 工具使用说明

6.1 运动控制

6.1.1 单轴运动

用于单轴进行点与点之间的定长运动。用户可设置定长距离、运行速度、加减速时间等参数，双击击单轴运动块，可对该工具块进行设置。

注意：多轴联动运行，需结合并行子流程块+单轴运动块两个工具实现。



单轴运动块的参数设置包括条件设置和参数设置两部分。

- 条件设置

根据设置的条件，决定运动功能是否执行。默认不启用，直接执行单轴运动。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行单轴运动。

- 参数设置

用户根据实际动作需求，订阅执行运动的轴号，配置相对/绝对运动模式，设置加减速时间或加减速度模式，目标位置、运行速度、S 曲线时间、加减速时间、到位延时支持订阅用户组态界面的数值框控件的值。

到位延时：目标位置到达后，延时设定的时间后，该工具执行完成。

等待定位完成：勾选后，等待指令位置到达后，再执行流程后续工具块。

等待编码器到位：该参数在绝对运动模式有效。勾选后，指令位置到达后，一直等待“编码器位置”和“指令位置”的差值在设定的“到位偏差”范围内。

等待编码器到位超时：该参数在绝对运动模式有效。勾选后，指令位置到达开始计时，“编码器位置”和“指令位置”的差值在设定的“到位超时”时间内达到“到位偏差”内，该工具块正常完成，否则异常完成。

6.1.2 回零运动

用于单轴进行回原点运动，设备运行时，一般都需要执行一次回原点运动。用户可设置回零高/低速、回零类型、回零方向、加减速时间等参数，双击回零运动块，可对该工具块进行设置。



回零运动块的参数设置包括条件设置和参数设置两部分。

- 条件设置

根据设置的条件，决定运动功能是否执行。默认不启用，直接执行回零运动。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行回零运动。

- 参数设置

用户根据实际动作需求，订阅执行运动的轴号，配置回零模式、加减速时间、运行速度等参数，速度相关参数支持订阅用户组态界面的数值框控件的值。

回零类型：EtherCAT 总线轴为从站回零模式，脉冲轴为控制器主站回零模式。回零模式的介绍请参见控制器使用手册。

6.1.3 Jog 运动

用于单轴进行连续运动，该轴将按设定的速度一直运行下去，直到遇到轴停止指令。双击 Jog 运动块，可对该功能块进行设置。



Jog 运动块的参数设置包括条件设置和参数设置两部分。

- 条件设置

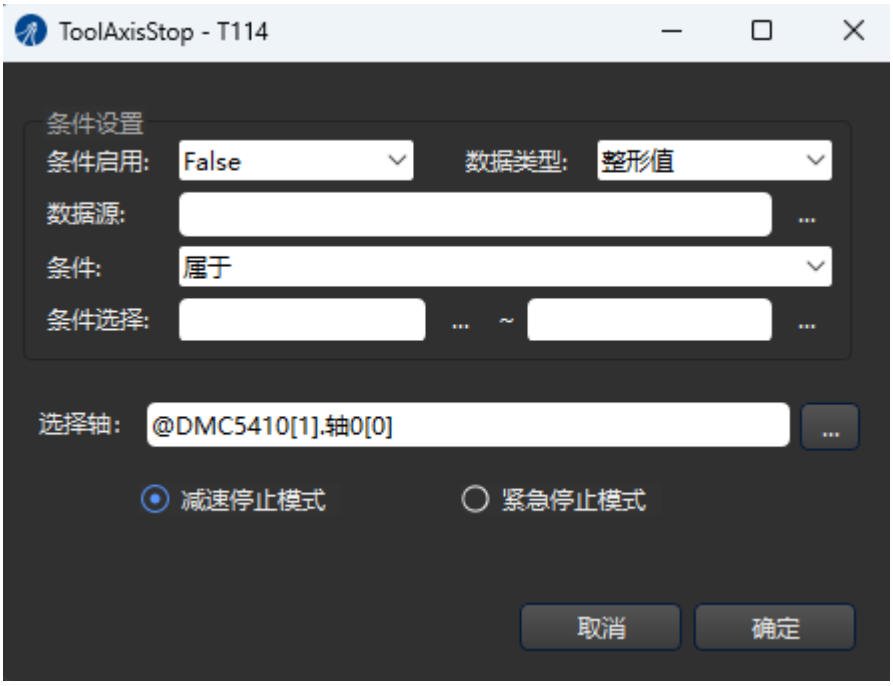
根据设置的条件，决定运动功能是否执行。默认不启用，直接执行 Jog 运动。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行 Jog 运动。

- 参数设置

用户根据实际动作需求，订阅执行运动的轴号，配置运动方向、加减速时间、运行速度等参数，运动参数支持订阅用户组态界面的数值框控件的值。

6.1.4 轴停止

用于控制轴停止运行。双击轴停止块，可对该功能块进行设置。



轴停止块的参数设置包括条件设置和参数设置两部分。

- 条件设置

根据设置的条件，决定运动功能是否执行。默认不启用，直接执行轴停止块。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行轴停止块。

- 参数设置

用户根据实际动作需求，订阅执行运动的轴号，配置停止模式。其中减速停止模式的减速停止时间将使用章节 3.1.2 配置轴页面的运行参数的减速时间。

6.1.5 直线插补

直线插补运动是按照一定的控制策略控制多轴联动，使运动平台用微小直线段精确地逼近轨迹的理论曲线，保证运动平台从起点到终点上的所有轨迹点都控制在允许误差范围内，可以保证运动轨迹的确定性。直线插补运动至少需要两个轴参与，否则会报错。双击直线插补块，可对该功能块进行设置。



直线插补运动块的参数设置包括条件设置和参数设置两部分。

- 条件设置

根据设置的条件，决定运动功能是否执行。默认不启用，直接执行直线插补块。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行直线插补块。

- 参数设置

参与插补运动的轴数量，需手动添加，用户根据实际动作需求，订阅执行运动的轴号、目标位置，配置插补系号、插补速度、加减速时间等参数。

6.1.6 圆弧插补

圆弧插补运动是按照一定的控制策略控制多轴联动，使运动平台用微小线段精确地逼近轨迹的理论曲线，保证运动平台从起点到终点上的所有轨迹点都控制在允许误差范围内，可以保证运动轨迹的确定性。圆弧插补运动至少需要两个轴参与，否则会报错。双击圆弧插补块，可对该功能块进行设置。



圆弧插补运动块的参数设置包括条件设置和参数设置两部分。

- 条件设置

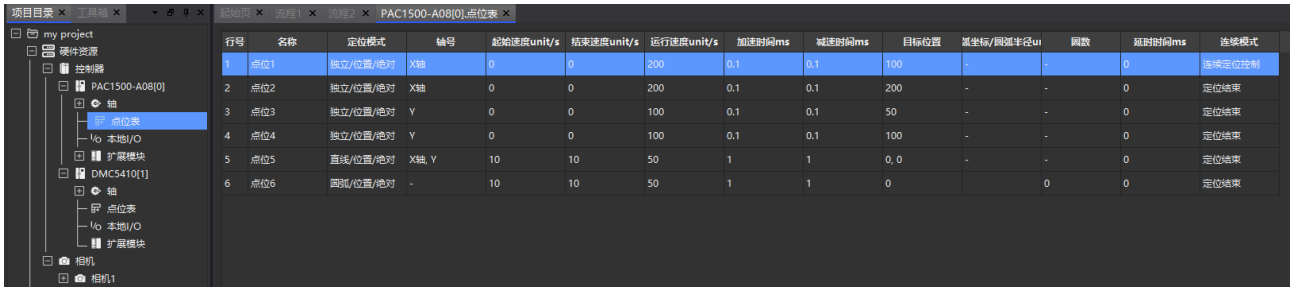
根据设置的条件，决定运动功能是否执行。默认不启用，直接执行圆弧插补块。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行圆弧插补块。

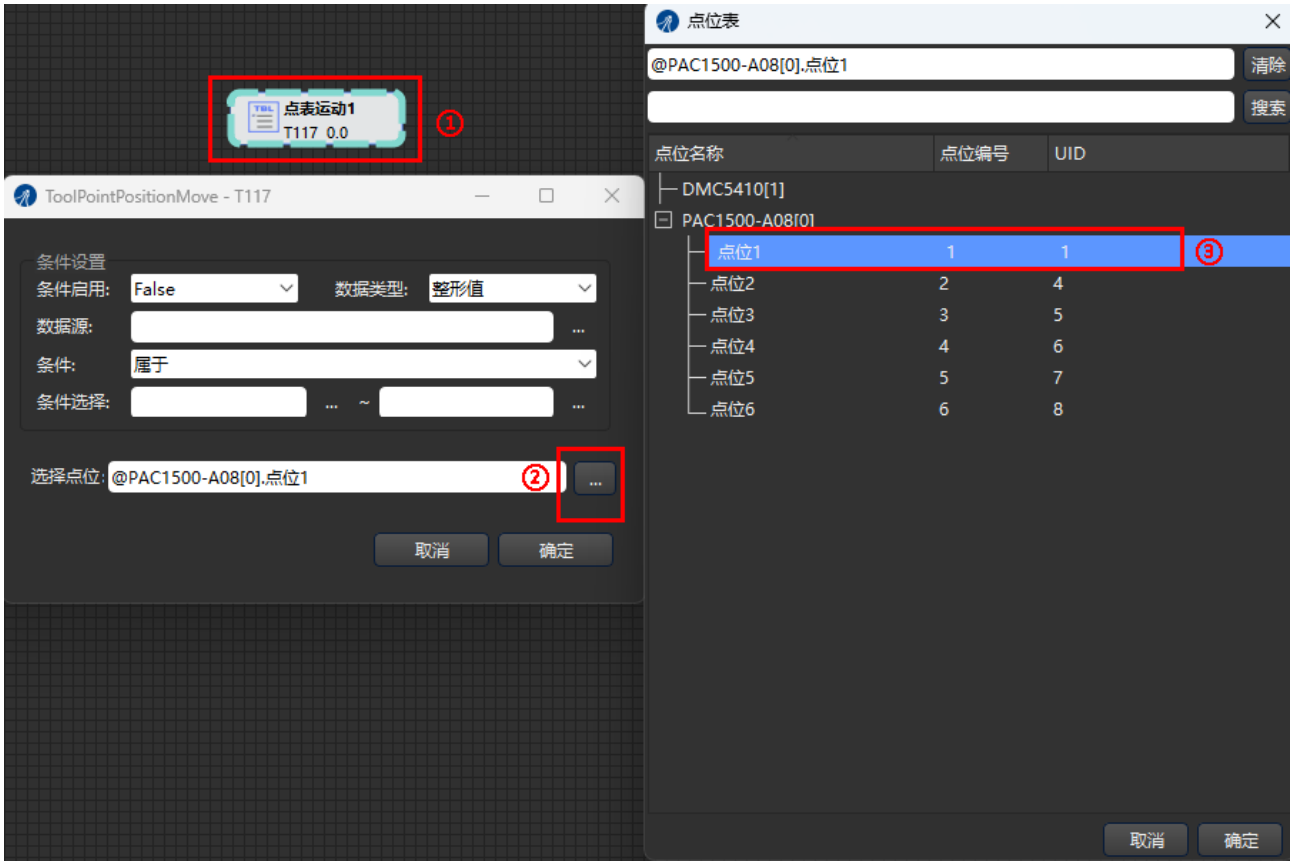
➢ 参数设置

圆弧插补支持三点圆弧、圆心终点圆弧、半径终点圆弧三种模式。参与插补运动的轴数量，需手动添加，用户根据实际动作需求，订阅执行运动的轴号、目标位置，配置插补系号、插补速度、加减速时间等参数。圆弧插补三种模式的介绍请参见控制器使用手册。

6.1.7 点表运动

点表运动是指各轴将动作类型（定位模式、目标坐标、速度等）或其它的设定（连续模式等）作为点表参数进行设定，在流程中的点表运动块仅需选择指定的点位名称，便可实现定位运动、插补运动等多种运动。





点表运动块的参数设置包括条件设置和参数设置两部分。

➢ 条件设置

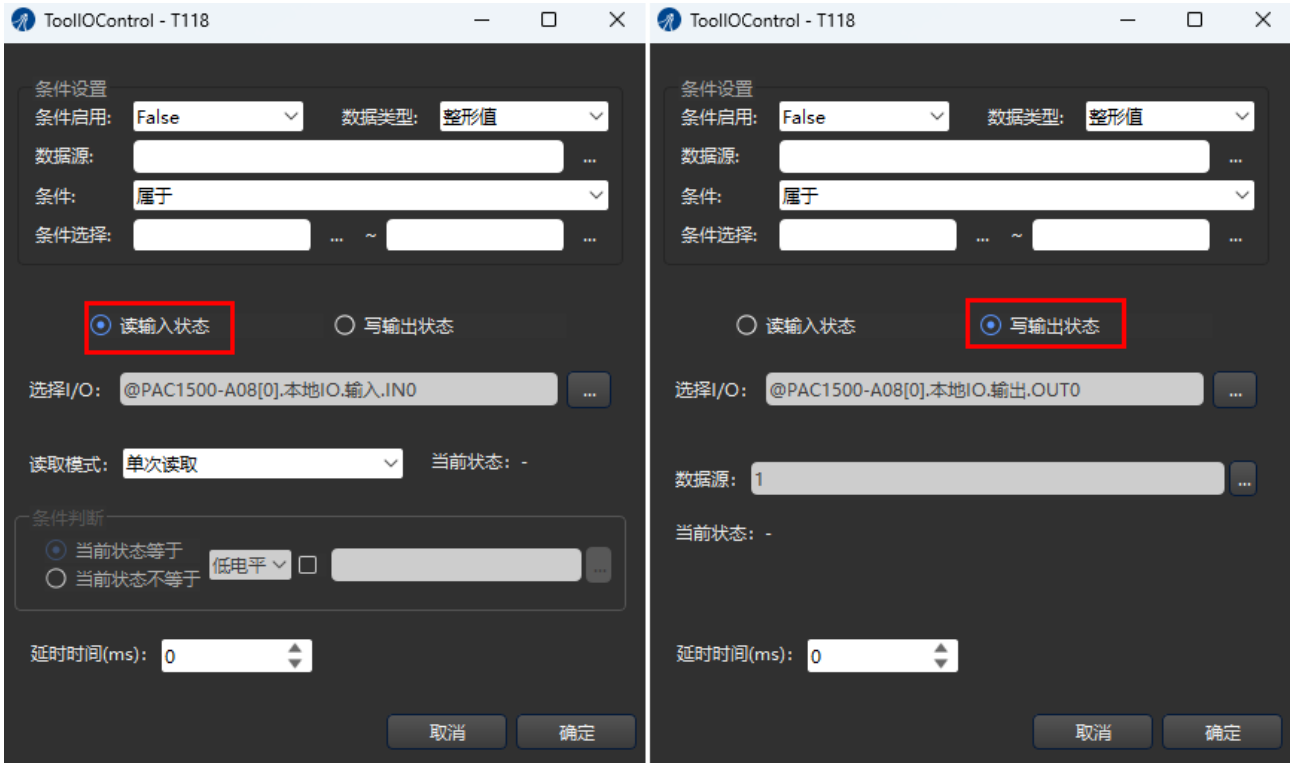
根据设置的条件，决定运动功能是否执行。默认不启用，直接执行点表块。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行点表运动块。

➢ 参数设置

用户根据实际动作需求，在点位表中定义运动路径，在点表运动块中订阅点表名称，即可实现预定的运动轨迹。点位表的配置请参加章节 3.1.5 配置点位表。

6.1.8 IO 控制

IO 控制用于读取输入信号或写入输出信号。



IO 控制的参数设置包括条件设置和参数设置两部分。

➢ 条件设置

根据设置的条件，决定运动功能是否执行。默认不启用，直接执行 IO 控制块。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行 IO 控制块。

➢ 参数设置

包括读取输入和写入输出两种模式。

读取输入：需订阅输入点号，设置读取模式“单次读取”或“连续读取直到条件判断成立”。设置“连续读取直到条件判断成立”模式时，还需设置输入状态为高/低电平。

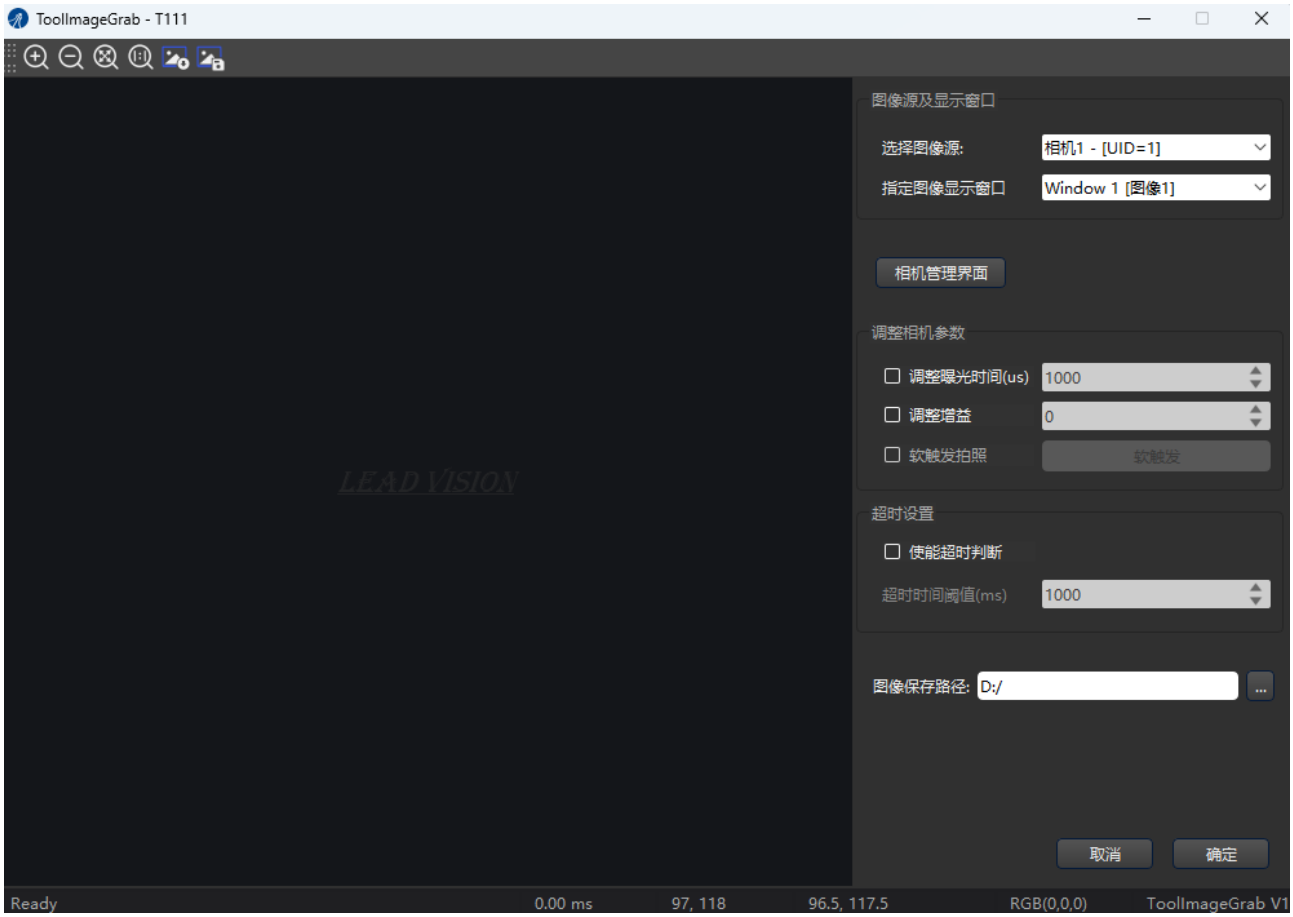
写入输出：需订阅输出点号，输出数据源，输出数据源可订阅，也可设置固定值 0 或 1。

延时时间：读取输入或写入输出后，再延时指定时间。

6.2 图像采集

6.2.1 采集图像

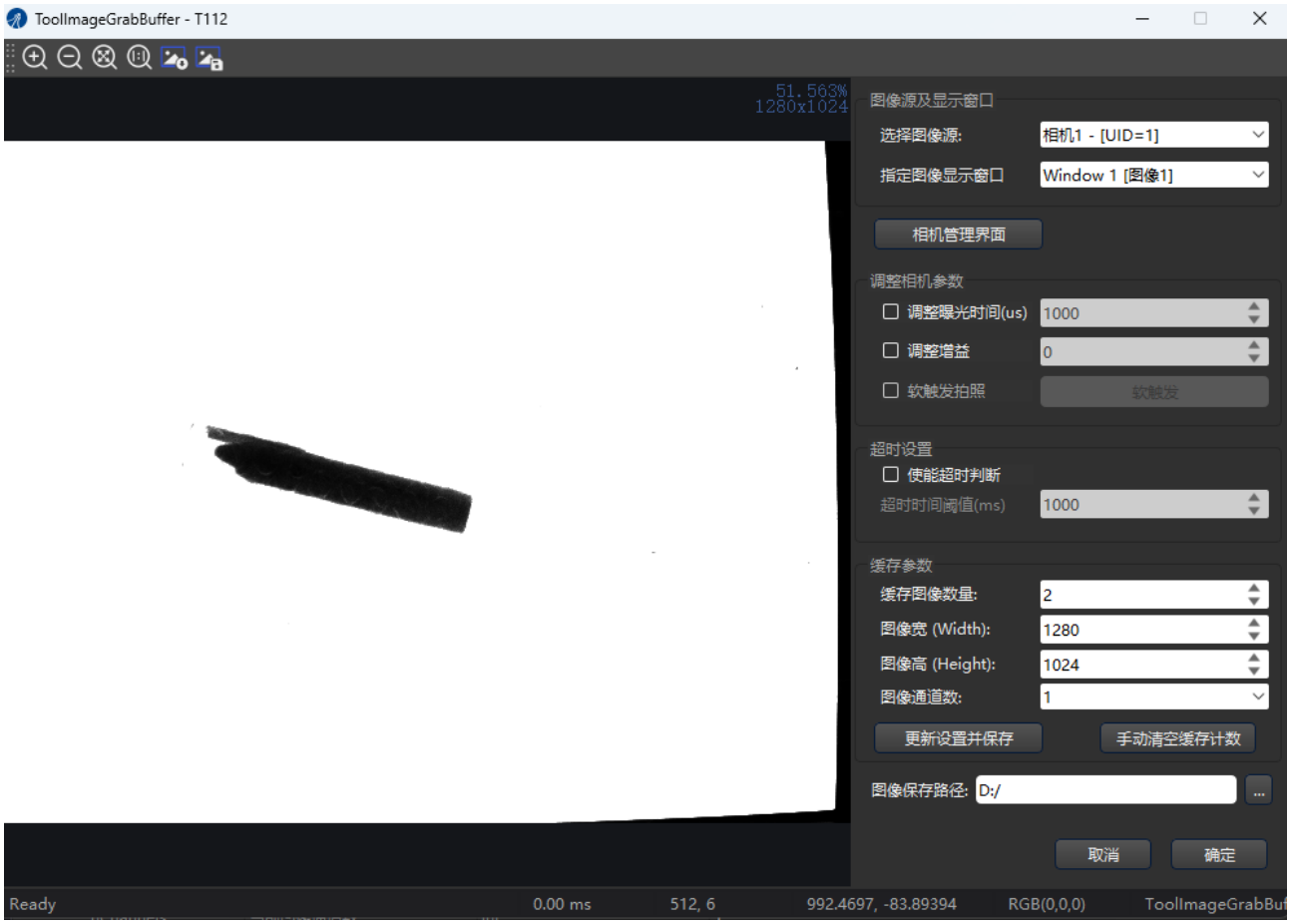
用于采集一张图像。选择图像来源并指定该图像显示的窗口，图像源可以从本地文件夹或者相机获取，在“相机管理界面”配置，当图像来源为相机获取时，可调整相机的曝光时间、增益和软硬触发模式。支持设置相机使能超时，超时后采集图像工具报错。图像保存路径，用于手动点击“保存图像”按钮的存储路径。



6.2.2 图像采集缓存

用于连续采集多张图像，如飞拍应用，最大缓存 64 张图像，通过“数组索引”工具块，可以选取指定的图像用于后续图像处理工具。选择图像来源并指定该图像显示的窗口，图像源可以从本地文件夹或者相机获取，在“相机管理界面”配置，当图像来源为相机获取时，可调整相机的曝光时间、增益和软硬触发模式。支持设置相机使能超时，超时后工具块报错。图像保存路径，用于手动点击“保存图像”按钮的存储路径。

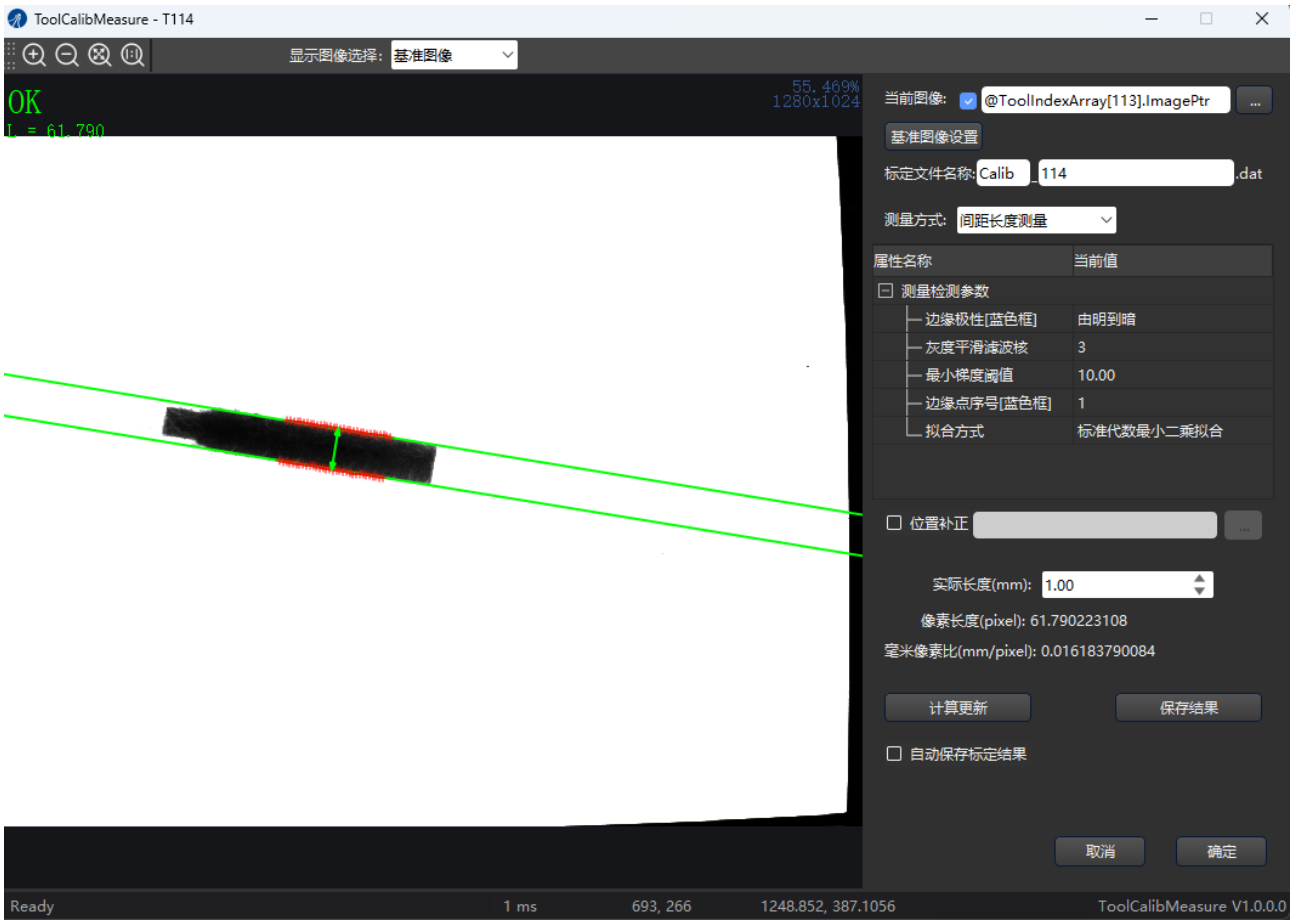
缓存参数用于配置连续采集图像次数，如 2，则执行到该工具后，需采集完 2 张图像后，该工具才执行完成，否则一直等待。图像宽、高需按实际相机或图片的像素设置，黑白图像通道数需设置为 1，彩色图像通道数需设置为 3。



6.3 标定

6.3.1 测量标定

通过测量已知尺寸的标准块，计算出像素单位到物理单位的转换比例，通常作为测量项目的标定。



- 基准图像设置：手动选择一张图像，作为执行测量标定的基准图像，也可运行该算子块加载一张图像作为基准图像。选择基准图像后，会自动添加一个矩形检测 ROI 或圆形 ROI，再设置 ROI 参数。
- 当前图像：算子运行时的当前图像，一般绑定图像采集、图像处理等工具的输出图像。如果未手动选择基准图像，则算子块运行时的第一张图像为基准图像。
- 标定文件名称：设置标定结果保存的文件名称。
- 测量方法：可以选择间距长度测量或圆半径测量。
- 测量方向：圆半径测量方法的参数，指定每个卡尺查找点的方向，有由圆心向圆外和由圆外向圆心两种模式。

- 测量宽度：圆半径测量方法的参数，卡尺的搜索长度，决定卡尺检测区域的纵向长度，即圆环宽度。
- 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。
- 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。
- 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多条边缘时，可修改该序号调整要检测的边缘。
- 拟合方式：拟合方式有标准代数最小二乘和带权重的最小二乘两种。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定轮廓匹配的仿射变换输出，基准图像也需保持和轮廓匹配的基准图像相同。
- 实际长度：设置测量目标的实际尺寸。当测圆形时，实际尺寸为半径。
- 像素长度：被测量目标的像素尺寸。
- 毫米像素比例：被测量目标的实际长度与像素长度的比值。
- 计算更新：执行标定计算出标定结果。
- 保存结果：保存标定结果，也可自动保存标定结果。

6.3.2 标量转换

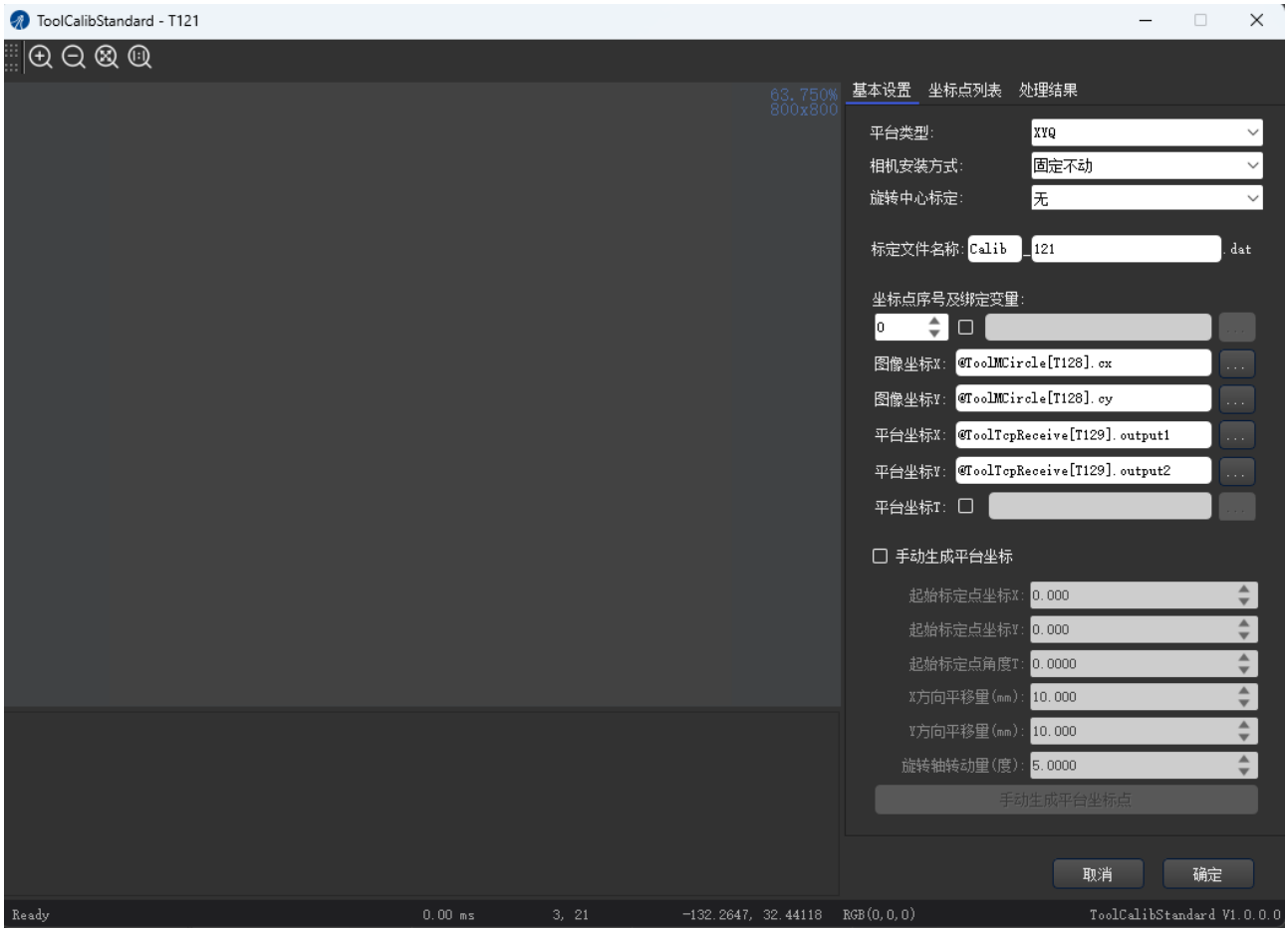
将输入的一组间距数据从图像坐标系转换到世界坐标系，可转换距离、宽度等像素单位到物理单位，可在“工具结果”查看转换后的标量值。



- 标定数据来源：加载“测量标定”工具生成的标定文件。
- 图像坐标系下标量值：设置需要转换的图像间距值，可关联。

6.3.3 N 点标定

根据配置的 N 组坐标点对（图像坐标—世界坐标），计算图像坐标映射到世界坐标的变换矩阵并输出标定文件。N 组数据可包括 9 组平移坐标和 7 组旋转坐标。一般用于一个 Mark 点 N 次拍照的标定。



该算子工具共 3 个设置页，分别为：基本设置、坐标点列表、处理结果。

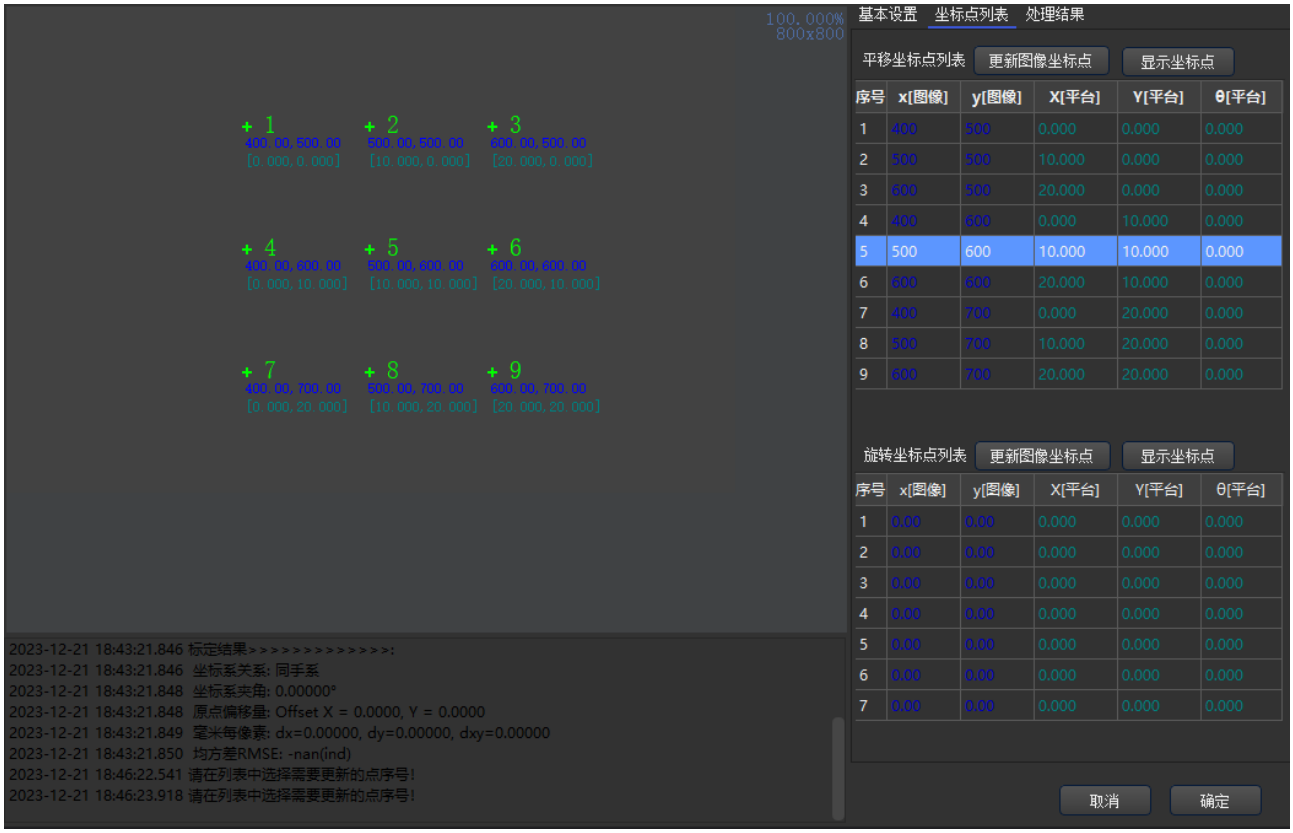
- 基本设置页

  用于配置标定平台类型。

  - 平台类型：支持 XYθ、Xθ+Y、Yθ+X 平台，其中 XYθ 平台包含 XY 平台。
  - 相机安装方式：可选固定不动、随 XY 移动、随 X 移动、随 Y 移动。
  - 旋转中心标定：包括拟合圆法找旋转中心、三角函数法找旋转中心（精确）。当选择“无”时，则 XYθ 平台为 XY 平台。
  - 标定文件名称：设置标定结果保存的文件名称。
  - 坐标点序号及绑定变量：N 点坐标数据的编号，范围[1~16]，可关联，平台移动后可通过通信工具发送此编号。
  - 图像坐标 X/Y：关联图像的 X/Y 坐标。平台每次移动后都需要相机进行一次拍照，并将特征点的图像坐标 X/Y 插入到“坐标点列表”页的图像坐标中。将 X 和 Y 分别关联到对应测量工具可快速插入数据，无需手动输入。
  - 平台坐标 X/Y/T：平台的物理 X/Y 坐标和角度 T（单位度）。平台每次移动后都需要相机进行一次拍照，并将平台物理坐标 X/Y 和角度 T 插入到“坐标点列表”页的平台坐标中，可关联通信接收到的平台物理坐标。
  - 手动生成平台坐标：勾选后使用下方手动设置的平台坐标。
  - 起始标定点坐标 X/Y/T：N 点标定第 5 个点的平台物理 X/Y 坐标和角度 T（单位度）。
  - X/Y 方向平移量：每次平台的移动量。
  - 旋转轴转动量：旋转标定时，每次转动的角度（单位度）。
  - 手动生成平台坐标点：点击后，应用平台设置的坐标数据，将在“坐标点列表”页的平台坐标中，自动生成平台坐标。

- 坐标点列表页

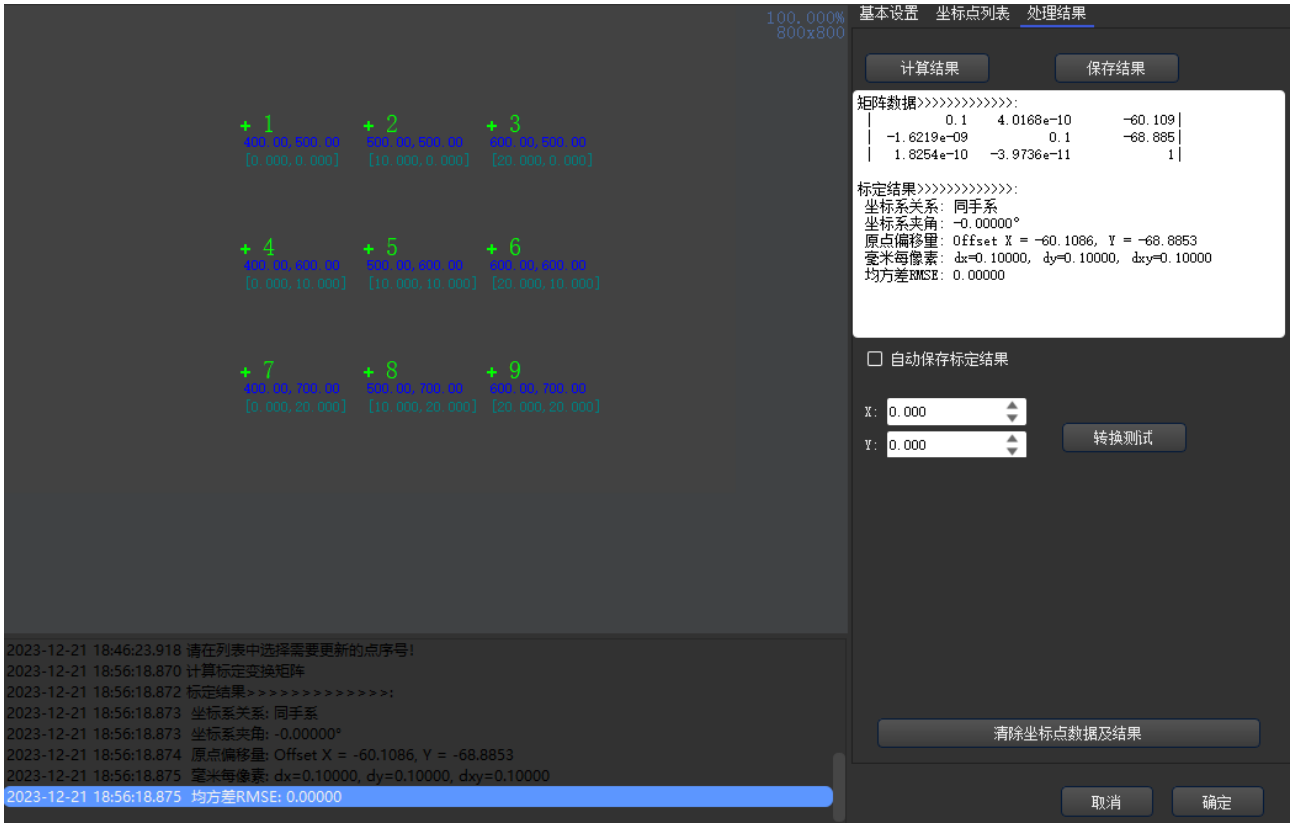
  图像坐标和平台物理坐标的对应点，用于生成坐标系的变换矩阵。



- 更新图像坐标点：手动将获取的图像坐标更新到表格。
- 显示坐标点：在左侧图像显示区显示所有点的坐标。

- 处理结果页

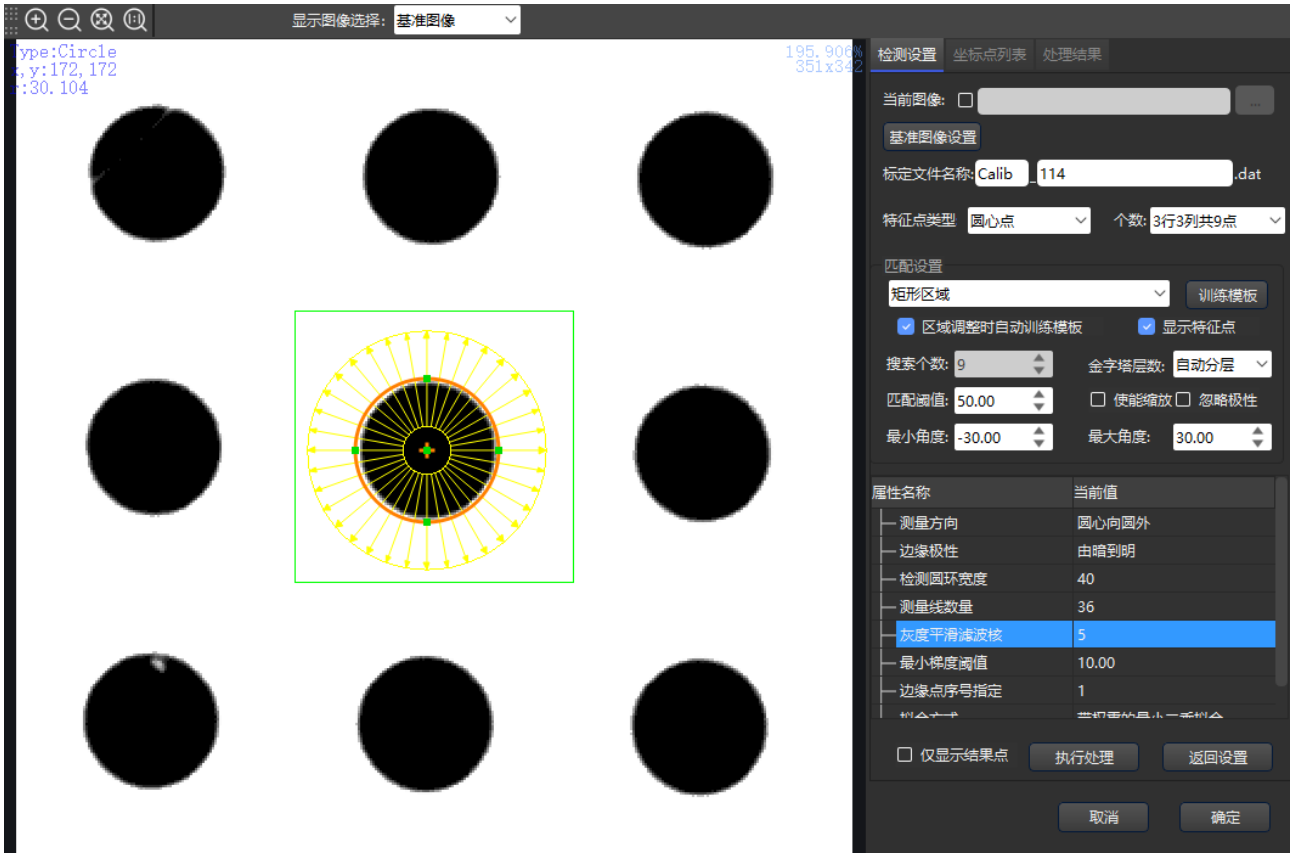
  计算标定结果并保存。



- 计算结果：生成标定矩阵数据。
- 保存结果：保存标定结果，也可勾选自动保存标定结果。
- 转换测试：输入图像坐标点，测试转换的平台物理坐标，验证标定是否正确。
- 清除坐标点数据及结果：清空坐标点及标定矩阵数据。

6.3.4 手动标定

通过构建一系列图像坐标和平台物理坐标的对应点，计算图像坐标系到平台物理坐标系之间的变换矩阵关系，并输出标定文件。一般用于 N 个 Mark 点一次拍照的标定。



该算子工具共 3 个设置页，分别为：检测设置、坐标点列表、处理结果。

- 检测设置页

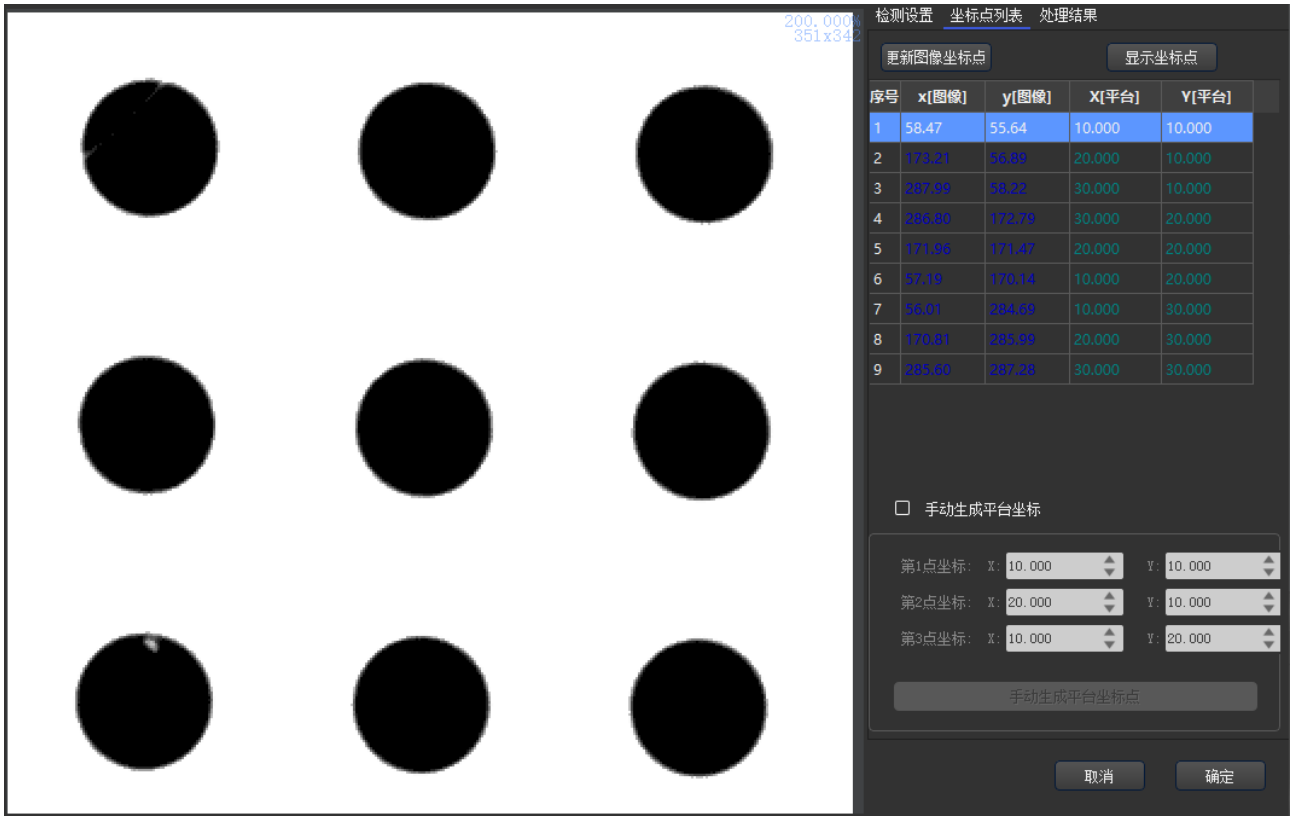
  用于配置检测目标的特征点参数，目标图形可以为圆形格、黑白格或圆形十字架等。

- 当前图像：算子运行时的当前图像，一般绑定图像采集、图像处理等工具的输出图像。
- 基准图像设置：手动选择建立检测特征点的基准图像，也可运行该算子块加载一张图像作为基准图像。用于设置 ROI 参数。
- 标定文件名称：设置标定结果保存的文件名称。
- 特征点类型：可以选择模板匹配对象的中心点、圆心对象的圆心点、黑白格对象的直角点。当选择“圆心点”时，会自动添加圆环 ROI，用于检测圆形边缘点；当选择“直角点”时，会自动添加 2 个矩形 ROI，用于检测两条直线。
- 个数：可以限定检测的目标个数为 9 个或不做限制。
- 匹配设置：选择模板矩形 ROI 形状后，将在图像区自动添加对应的 ROI，将模板区域 ROI 拖动到需要作为模板的图像区域，一般与“特征点类型”选择同一个区域。
- 训练模板：点击时生成一次模板。
- 区域调整时自动训练模板：勾选后，调整 ROI 区域自动生成新的模板。

- 显示特征点：勾选后，显示模板轮廓点。
- 搜索个数、金字塔层级、匹配阈值、最小角度、最大角度、使能缩放、忽略极性等搜索模板参数设置，请参考模板匹配工具介绍章节。
- 检测属性设置：当选择“圆心点”时，请参考圆查找工具介绍章节。当选择“直角点”时，请参考直线查找章节。
- 执行处理：按上述设置的参数，执行一次本工具，得到特征点结果。
- 返回设置：返回到 ROI 设置状态，重新设置检测区域。
- 仅显示结果点：只显示找到的特征点。

- 坐标点列表页

  构建图像坐标和平台物理坐标的对应点，用于生成坐标系的变换关系。

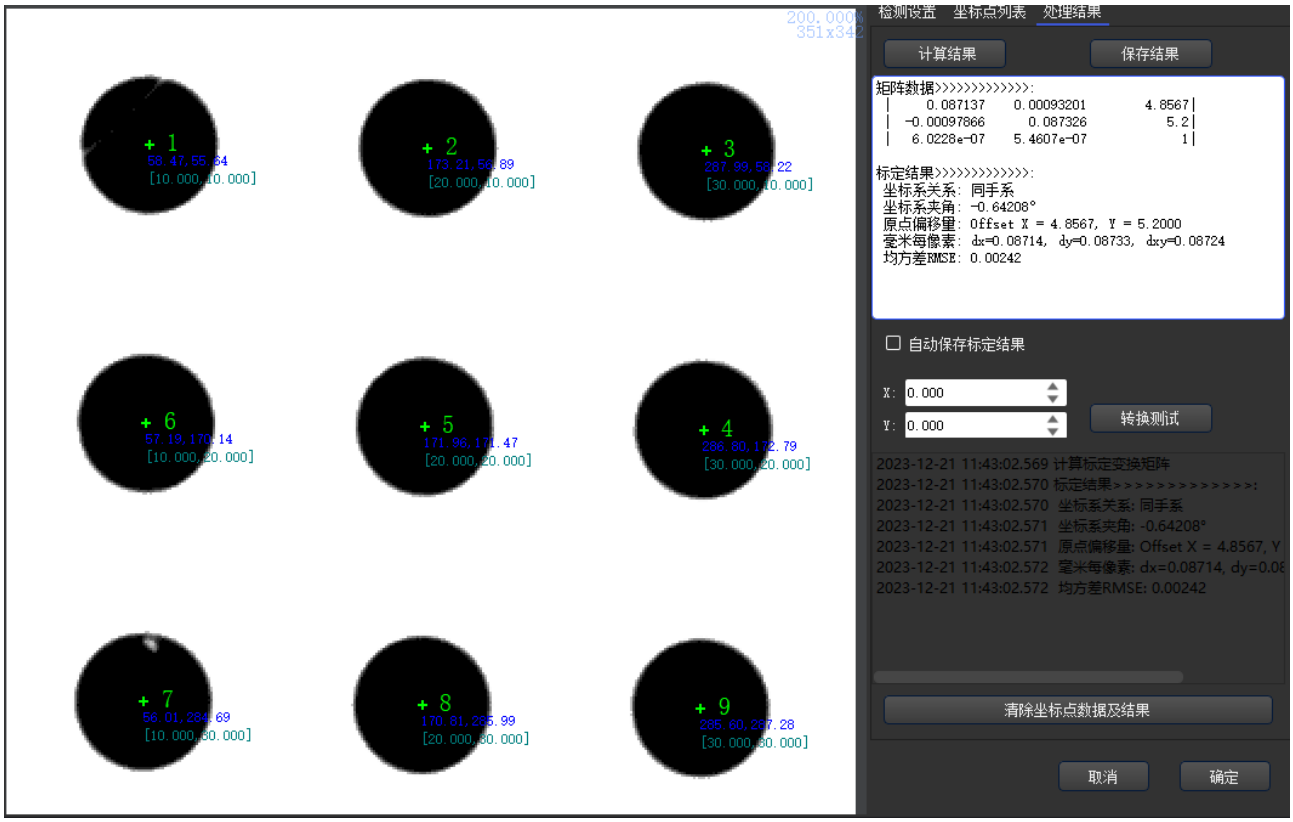


- 更新图像坐标点：将检测到的特征点坐标更新到表格。
- 显示坐标点：在左侧图像显示区显示所有点的坐标。
- 手动生成平台坐标：未勾选时，所有图像上的点均需移动平台对点，对点完成后，手动输入每一个图像坐标对应的平台实际物理坐标。勾选后，左侧图像显示区将标注 3 个特征点，平台移动到对应序号的位置，记录 3 点实际物理坐标，填入下方的第 1 点坐标~第 3

点坐标中，然后点击“手动生成平台坐标点”，剩余平台坐标将自动填充，用户只需对点 3 次即可。

- 处理结果页

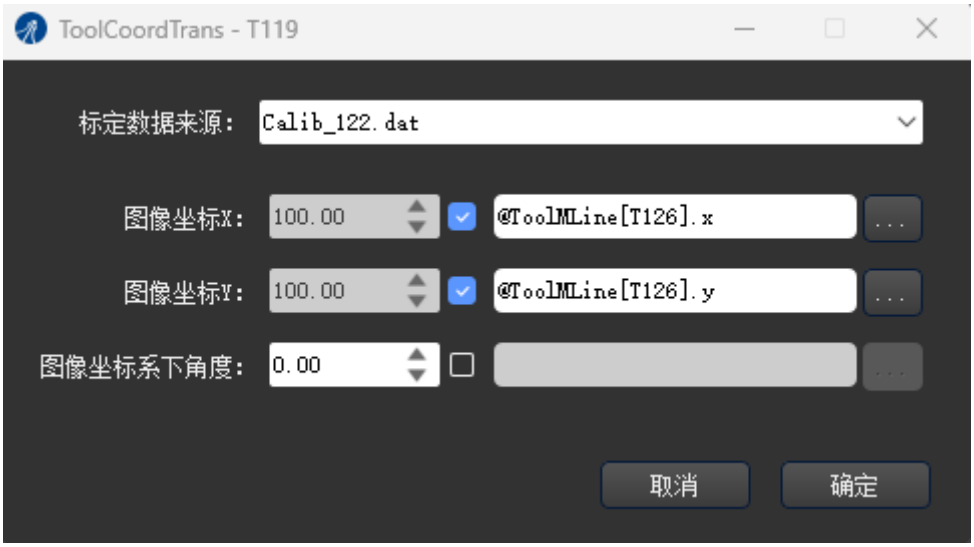
  计算标定结果并保存。



- 计算结果：生成标定矩阵数据。
- 保存结果：保存标定结果，也可勾选自动保存标定结果。
- 转换测试：输入图像坐标点，测试转换的平台物理坐标，验证是否正确。
- 清除坐标点数据及结果：清空坐标点及标定矩阵数据。

6.3.5 坐标转换

将输入的一组坐标数据从图像坐标系转换到世界坐标系，可在“工具结果”查看转换后的坐标值。

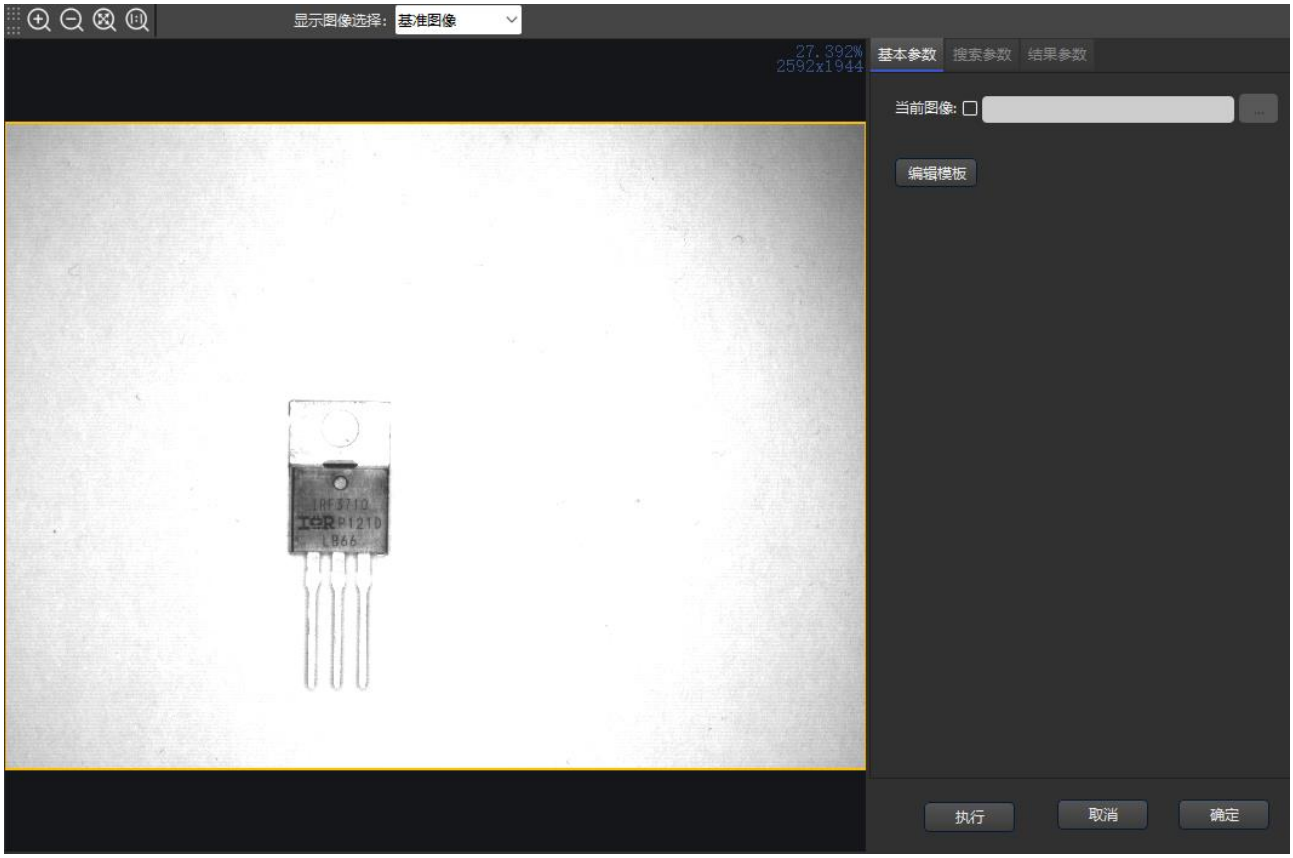


- 标定数据来源：加载标定工具生成的标定文件。
- 图像坐标 X：设置需要转换的坐标 X 轴数据，可关联。
- 图像坐标 Y：设置需要转换的坐标 Y 轴数据，可关联。
- 图像坐标系下角度：设置需要转换的角度数据，可关联。

6.4 定位

6.4.1 轮廓匹配

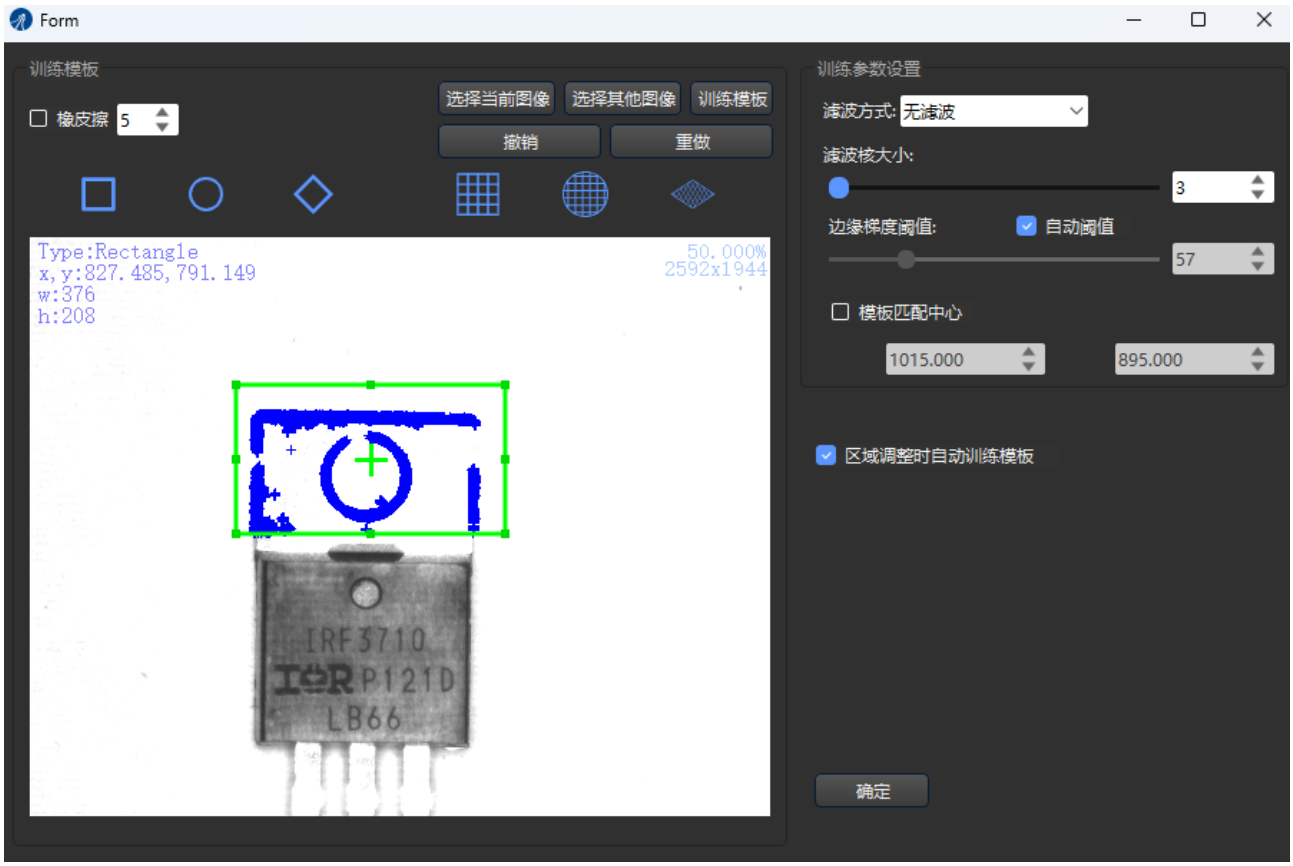
使用图像的边缘轮廓特征作为模板，按照预设的参数确定搜索空间，在图像中搜索与模板相似的目标，允许目标存在旋转、缩放、亮度变化、模糊、遮挡等情况，可用于定位、计数和判断有无等。



该算子工具共 3 个设置页，分别为：基本设置、搜索参数、结果参数。

- 基本设置页

  主要用于配置创建模板时的参数。点击“编辑模板”进入模板设置页面。



- 选择当前图像：算子运行时加载的当前图像，点击后将当前图像作为设置搜索模板参数的图像。
- 选择其他图像：打开加载图像对话框，选择一张本地图像作为设置搜索模板参数的图像。
- 模板区域及掩膜区域：选择模板 ROI 形状（矩形、圆、旋转矩形）后，将在图像区自动添加对应的 ROI，将模板区域 ROI 拖动到需要作为模板的图像区域，如果需要屏蔽干扰区域，将掩膜区域 ROI（带网格的矩形、圆、旋转矩形）拖动到需要屏蔽的图像区域。
- 橡皮擦：勾选后，点击鼠标左键可擦除滑动的区域，与掩膜区域相同的效果。
- 训练模板：点击时生成一次模板。
- 滤波方式：对图像进行滤波（高斯、中值、边缘保持）操作。
- 边缘梯度阈值：可选自动和手动两种模式，建议勾选自动模式，当自动模式不满足要求时再切换到手动模式调整，范围 1~255，只有边缘梯度阈值大于该值的边缘点才能被检测到。数值越大，抗噪声能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除。
- 模板匹配中心：可设置任意位置为模板中心点，而非 ROI 区域的中心。
- 区域调整时自动训练模板：勾选后，调整 ROI 区域自动生成新的模板。

➢ 备注：模板搜索范围默认为全图像搜索，可通过调整图像显示区域最外边的矩形 ROI，将搜索区域限定在设置的 ROI 矩形区域内。

创建模板流程如下：

Step1：点击“编辑模板”，在弹出中选择“当前图像”或“其他图像”作为设置模板参数的基准图像。并在“显示图像选择”下拉列表框中，选中基准图像显示。

Step2：在“模板区域及掩膜区域”中，选择 ROI 形状，并将 ROI 拖动到需要作为模板的图像区域。如果需要屏蔽模板区域的干扰区，可添加掩膜区域，将掩膜 ROI 拖动到需要屏蔽的图像区域。如果需要屏蔽少量的干扰区，可使用橡皮擦擦除干扰点。

Step3：根据实际效果，设置滤波方式、边缘梯度阈值。

Step4：点击训练模板，完成模板创建并观察模板效果，建议勾选区域调整时自动训练模板和显示特征点，便于实时观察 ROI 区域拖动中提取的模板特征。

Step4：点击确认按钮，保存模板。

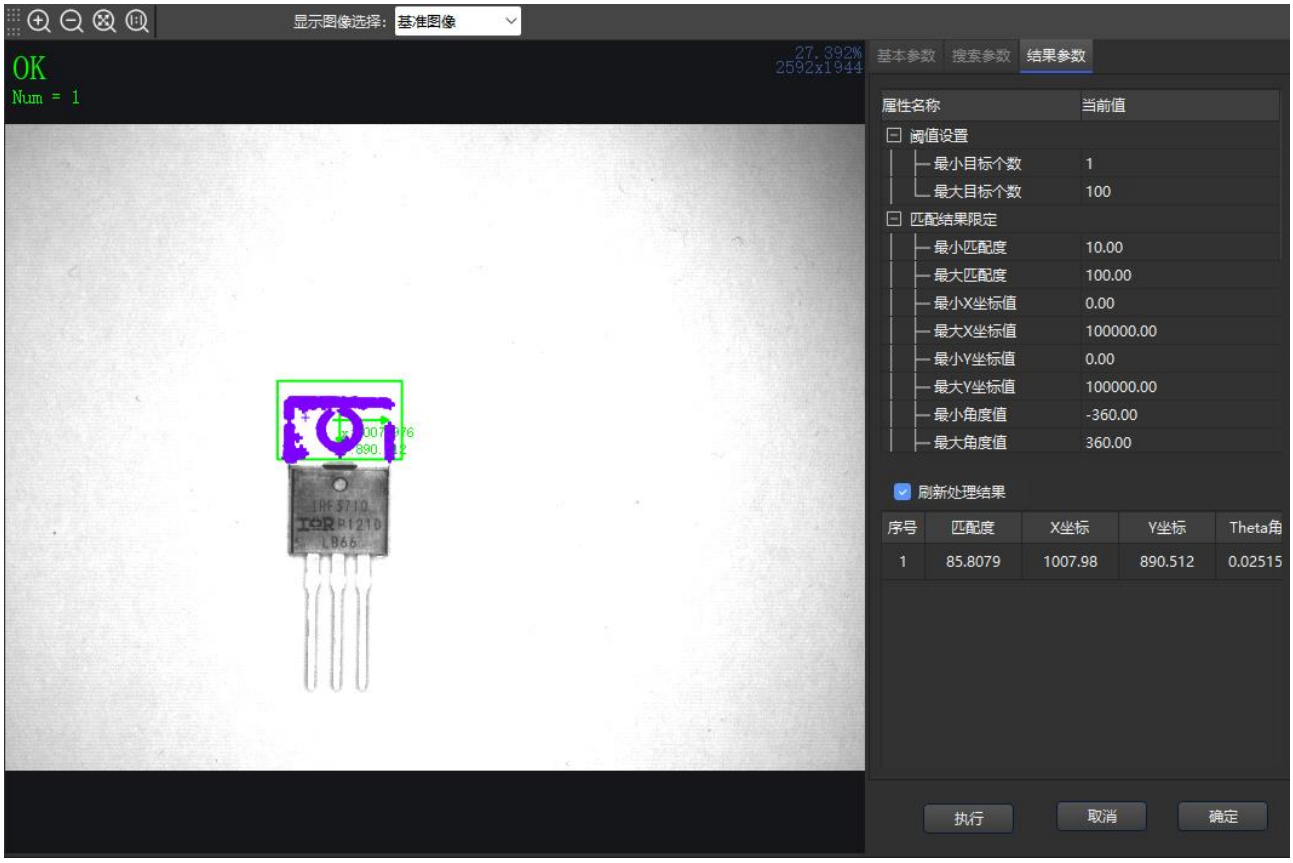
● 搜索参数页



可以修改运行时的匹配参数，以提高匹配成功率。比如，当搜索图像与模板相似度低时，可调低匹配阈值，提高匹配机会，但这也会增加错误匹配的机会。再比如，当匹配速度慢时，可将金字塔级别提高，以加快形状轮廓检测速度。

- 搜索个数：设置允许查找的最大目标个数，识别结果的数量将小于或等于该数量。
- 匹配阈值：模板与搜索图像中目标的相似程度，即相似度阈值，搜索时在相似度达到该阈值时才会被确认为目标，最大是 100，表示完全契合。
- 允许遮挡率：当搜索的目标存在遮挡时，匹配对象所允许的最大遮挡比例，遮挡率小于该值的对象将被正常匹配，范围 0~0.75。
- 金字塔层级：用于加速匹配过程，在目标面积较大时可以适当调高，级别越高，速度越快，但可能会降低识别能力，级别越小精度越高。
- 搜索速度：设置匹配过程中的匹配速度等级。搜索速度值越大速度越快，但可能会导致识别率降低，需要根据实际情况设置，在保证能够识别到目标的情况下再调大。
- 使能加速：通过减小匹配精细度，提升匹配速度。如模板的轮廓特征太多，勾选后将会自动减少特征点从而提高匹配速度。
- 使能缩放：使能搜索过程中最大最小缩放比例。
- 忽略极性：不勾选，待识别目标的边缘方向跟模板一致；勾选后，待识别目标的边缘方向跟模板一致或相反都可以，可用于目标和模板之间存在局部光照变化的情况，该模式会花费更多的时间。
- 最小角度：待识别目标相对于已创建模板的最小旋转角度，取值范围 0°~360°或-180°~180°。
- 最大角度：待识别目标相对于已创建模板的最大旋转角度，取值范围-0°~360°或-180°~180°。
- 最小比例：待识别目标尺寸相对于模板尺寸的最小缩放比例，取值范围 0.8~1。
- 最大比例：待识别目标尺寸相对于模板尺寸的最大缩放比例，取值范围 1~1.2。
- 角度输出范围：限定模板输出的角度范围在 0~360°或-180°~180°。
- 排序方式：当存在多个相同目标图形时，用于设置匹配结果的排序方式。
- 参数如果调整自动执行匹配：当修改参数时自动执行一次匹配。

- 结果参数页

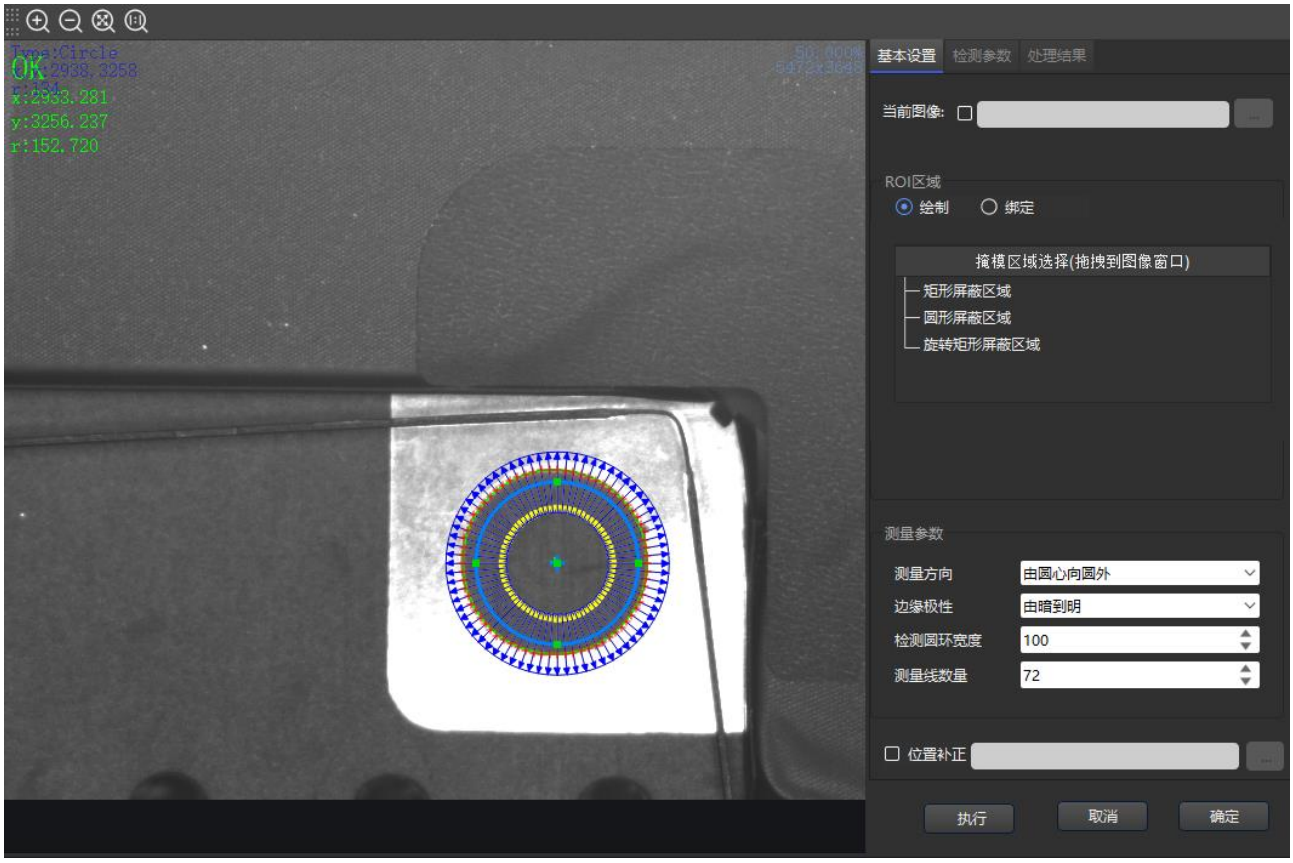


查看匹配结果及设置结果阈值判定范围。

- 阈值设置：当匹配目标个数、匹配度、X 坐标范围、Y 坐标范围、角度范围、缩放比范围均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：设置目标矩形框颜色、匹配目标特征点颜色、搜索区域显示颜色及线宽。
- 匹配度：匹配目标与模板的相似度。
- 坐标：匹配目标相对于当前图像的 XY 坐标位置。
- Theta 角度：匹配目标相对于模板的旋转角度。
- 缩放比：匹配目标相对于模板的缩放比例。

6.4.2 圆查找

从设定圆环形范围内检测符合条件的边界点，利用所有被检测的点拟合出一个圆，得到圆心坐标和半径。可用于圆的定位和测量。



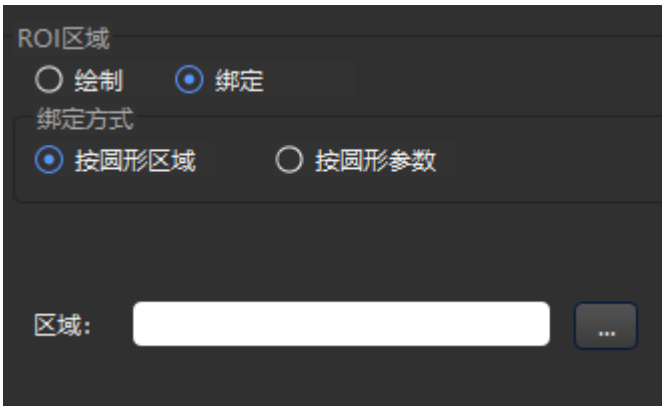
该算子工具共 3 个设置页，分别为：基本设置、检测参数、处理结果。

- 基本设置页

  主要用于设置圆查找的测量参数。

➢ 当前图像：算子运行时检测圆形的图像，一般绑定图像采集、图像处理等工具的输出图像。运行时会自动添加一个圆形检测 ROI。

➢ ROI 区域：包括绘制和绑定两种方式。当选择绘制时，在图像区域绘制圆形 ROI 大小及位置；当选择绑定时，需绑定其他算子块的圆形输出，如圆拟合算子块的输出。

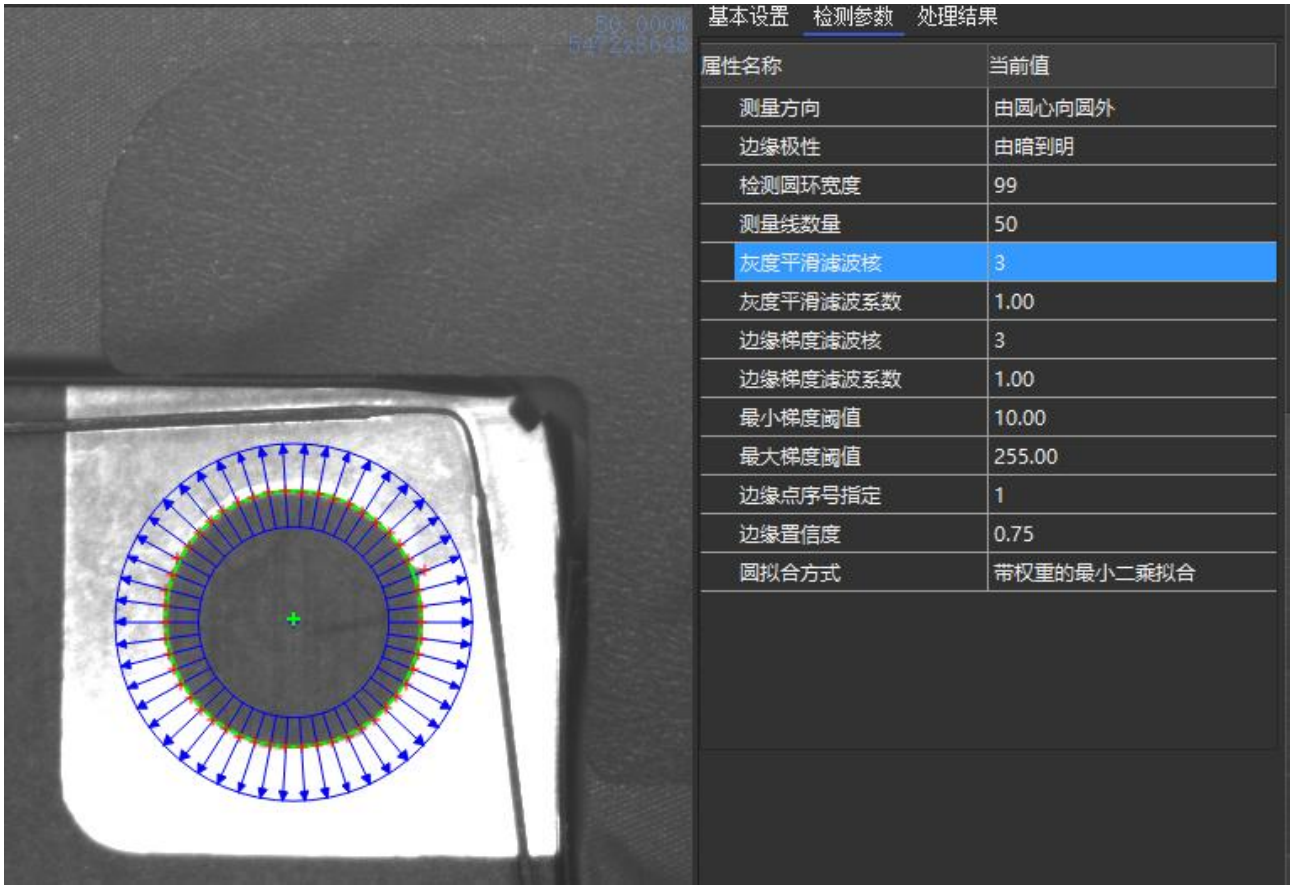


➢ 测量方向：指定每个卡尺查找点的方向，有由圆心向圆外和由圆外向圆心两种模式。

- 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。
- 检测圆环宽度：卡尺的搜索长度，决定卡尺检测区域的纵向长度，取值范围 5~99。
- 测量线数量：即卡尺的个数，决定找点的区域个数，每个卡尺里找到一个满足的点。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的圆形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

- 检测参数页

  圆检测工具运行时的参数配置。



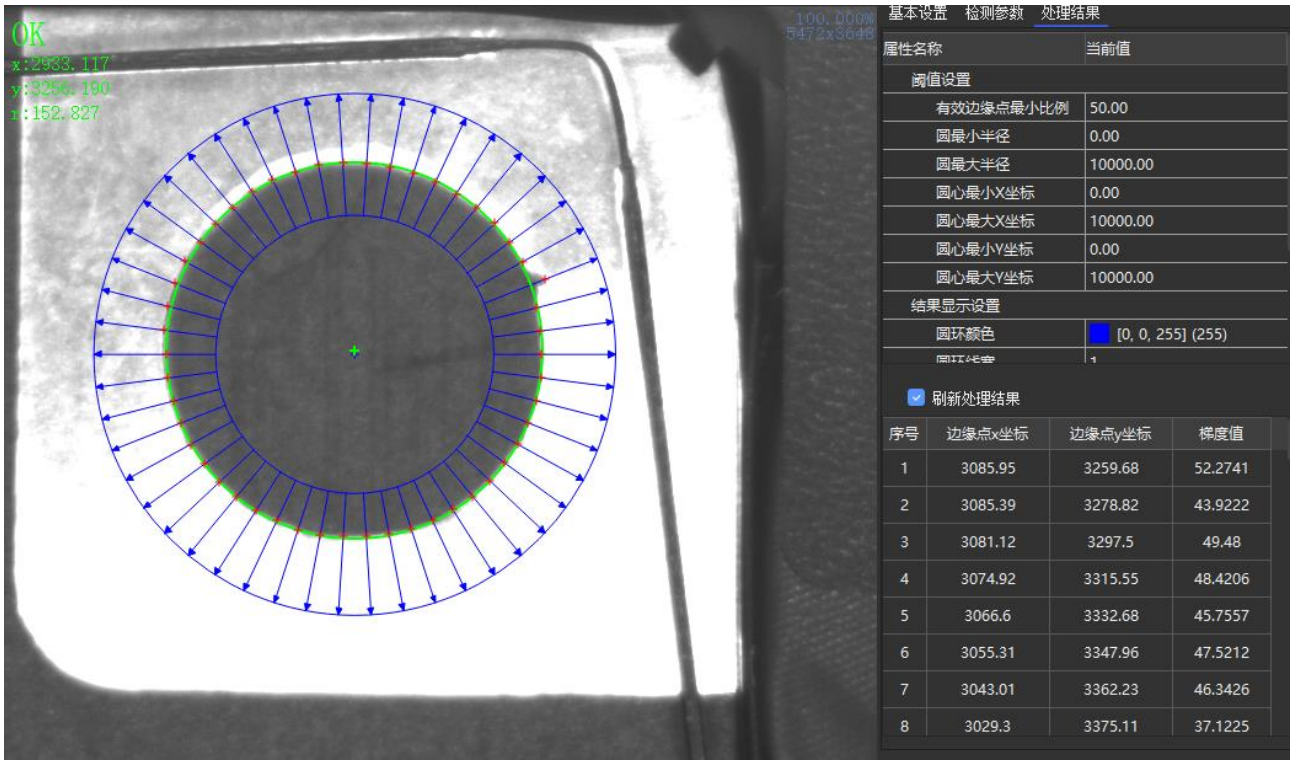
- 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。
- 灰度平滑滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。

边缘梯度滤波核：卡尺测量线方向上的滤波核大小，用于增强边缘和抑制噪音干扰，最小值为 1，当边缘模糊不清晰或有噪音干扰时可以增大滤波核，这样可以使得检测结果更加稳定。

- 边缘梯度滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。
- 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。
- 最大梯度阈值：设定检测边缘梯度阈值的上限，只有边缘梯度阈值小于该值的边缘点才被检测到，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多个圆时，可修改该序号调整要检测的圆。
- 边缘置信度：当边缘点序号设置为 0 时有效，边缘点梯度的权重占该卡尺区域梯度最大值的比重。
- 圆拟合方式：拟合方式有标准代数最小二乘、基于正交距离最小二乘和带权重的最小二乘三种。

- 处理结果页

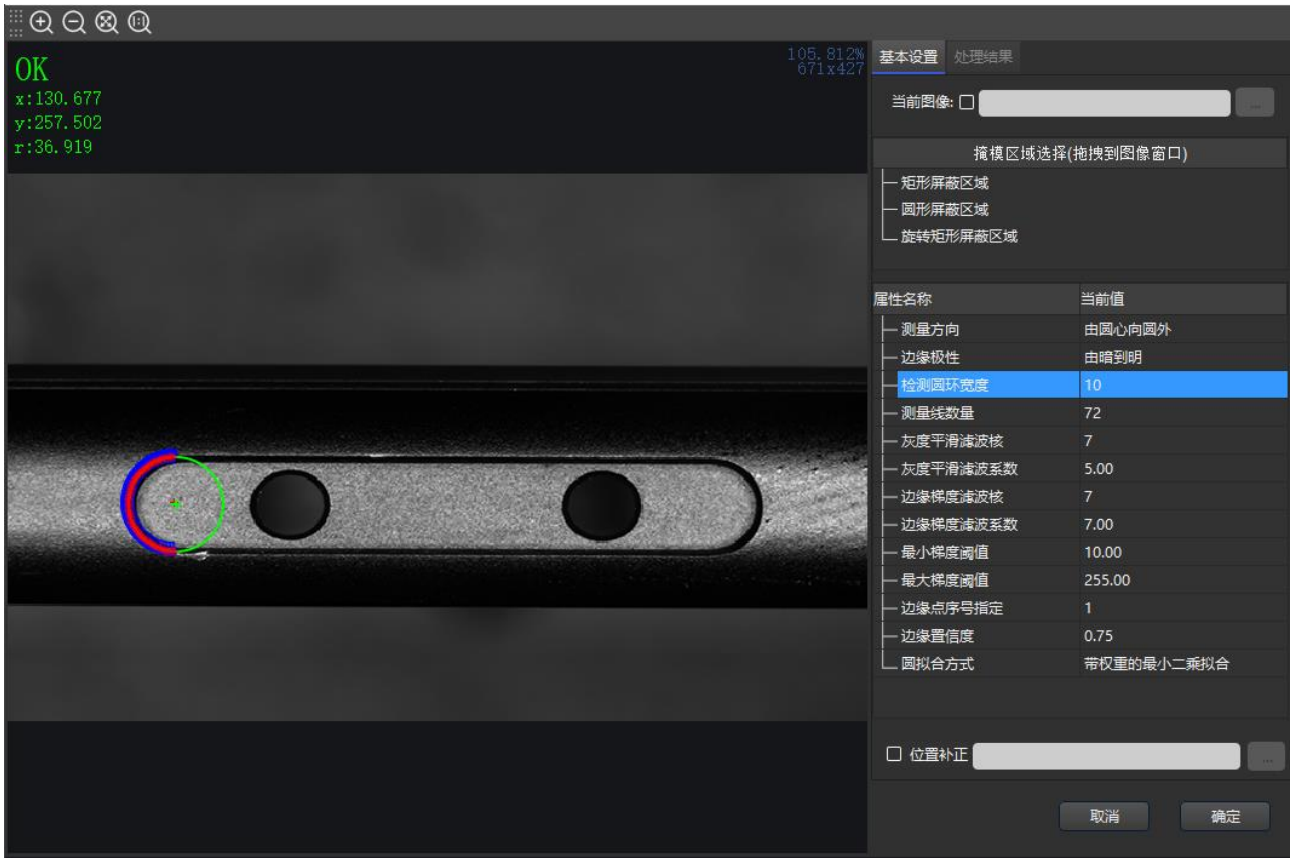
查看圆查找结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的结果、拟合圆的圆心坐标和半径。



- 阈值设置：当有效边缘点最小比例，圆最小最大半径、圆形最小最大 XY 坐标均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：圆环颜色、拟合的圆颜色、边缘点颜色及线宽设置。
- 序号：边缘点编号。
- 边缘点坐标：所有检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：所有检测到的边缘点的边缘梯度阈值。

6.4.3 圆弧查找

从设定圆弧范围内检测符合条件的边界点，利用所有被检测的点拟合出一个圆，得到圆弧圆心坐标和半径，可用于圆弧定位和测量。



该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置圆弧查找的测量参数。

➢ 当前图像：算子运行时查找圆弧的图像，一般绑定图像采集、图像处理等工具的输出图像。运行时会自动添加一个圆弧查找 ROI。

➢ 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。

➢ 测量线数量：即卡尺的个数，决定找点的区域个数，每个卡尺里找到一个满足的点。

➢ 检测圆环宽度：卡尺的搜索长度，决定卡尺检测区域的纵向长度。

➢ 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。

➢ 灰度平滑滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。

- 边缘梯度滤波核：卡尺测量线方向上的滤波核大小，用于增强边缘和抑制噪音干扰，最小值为 1，当边缘模糊不清晰或有噪音干扰时可以增大滤波核，这样可以使得检测结果更加稳定。
- 边缘梯度滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。
- 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。
- 最大梯度阈值：设定检测边缘梯度阈值的上限，只有边缘梯度阈值小于该值的边缘点才被检测到，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多条直线时，可修改该序号调整要检测的直线。
- 边缘置信度：当边缘点序号设置为 0 时有效，边缘点梯度的权重占该卡尺区域梯度最大值的比重。
- 圆拟合方式：拟合方式有标准代数最小二乘、基于正交距离最小二乘和带权重的最小二乘三种。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

- 处理结果页

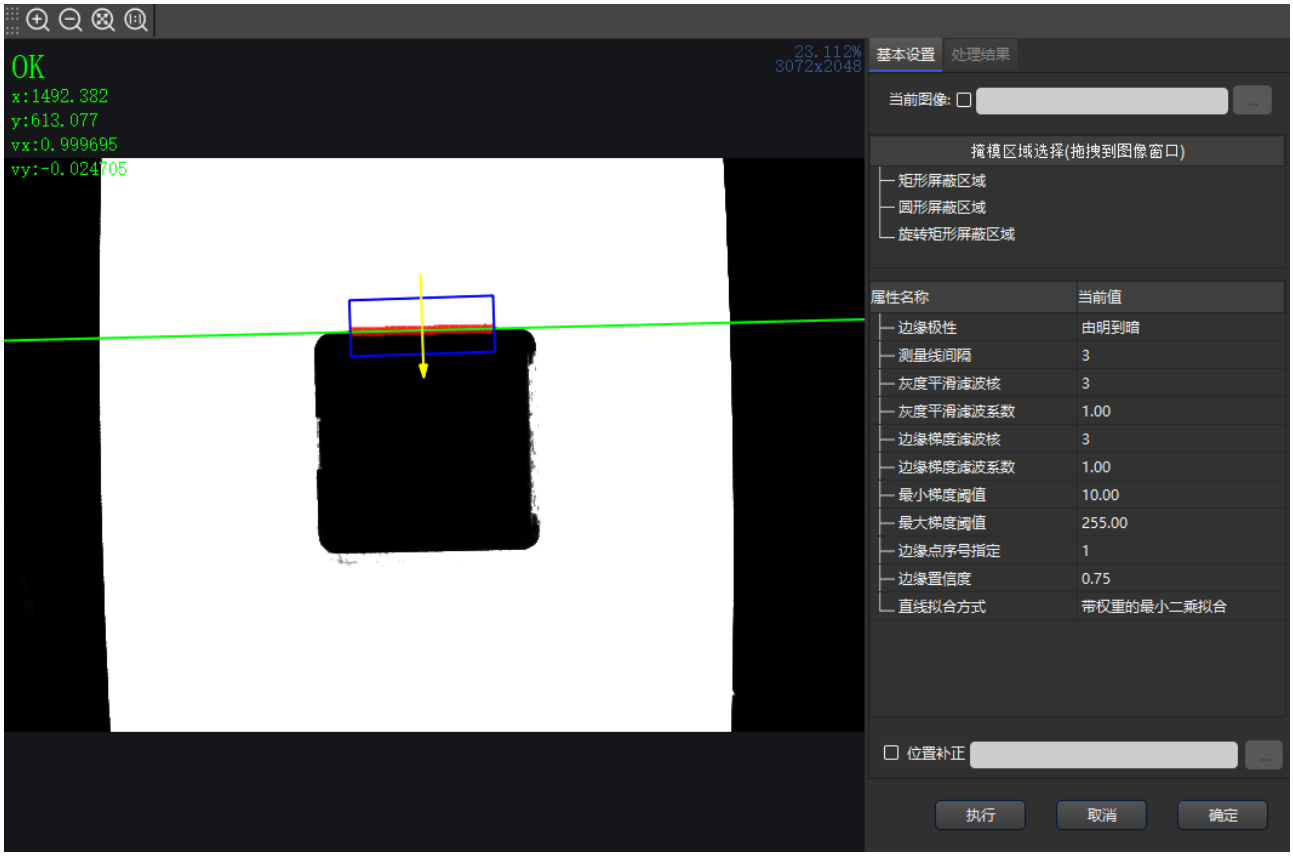
查看圆弧查找结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的结果、拟合圆弧的圆心坐标和半径。



- 阈值设置：当有效边缘点最小比例，圆弧最小最大半径、圆弧最小最大 XY 坐标均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：圆弧圆环颜色、拟合的圆弧颜色、边缘点颜色及线宽设置。
- 序号：边缘点编号。
- 边缘点坐标：所有检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：所有检测到的边缘点的边缘梯度阈值。

6.4.4 直线查找

从设定矩形范围内检测符合条件的边界点，利用所有被检测的点拟合出一条直线，得到矩形框中心点到该直线的垂足坐标和直线的方向向量，可用于直线定位和测量。



该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置直线查找的测量参数。

➢ 当前图像：算子运行时查找直线的图像，一般绑定图像采集、图像处理等工具的输出图像。运行时会自动添加一个矩形检测 ROI。

➢ 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。

➢ 测量线数量：即卡尺的个数，决定找点的区域个数，每个卡尺里找到一个满足的点。

➢ 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。

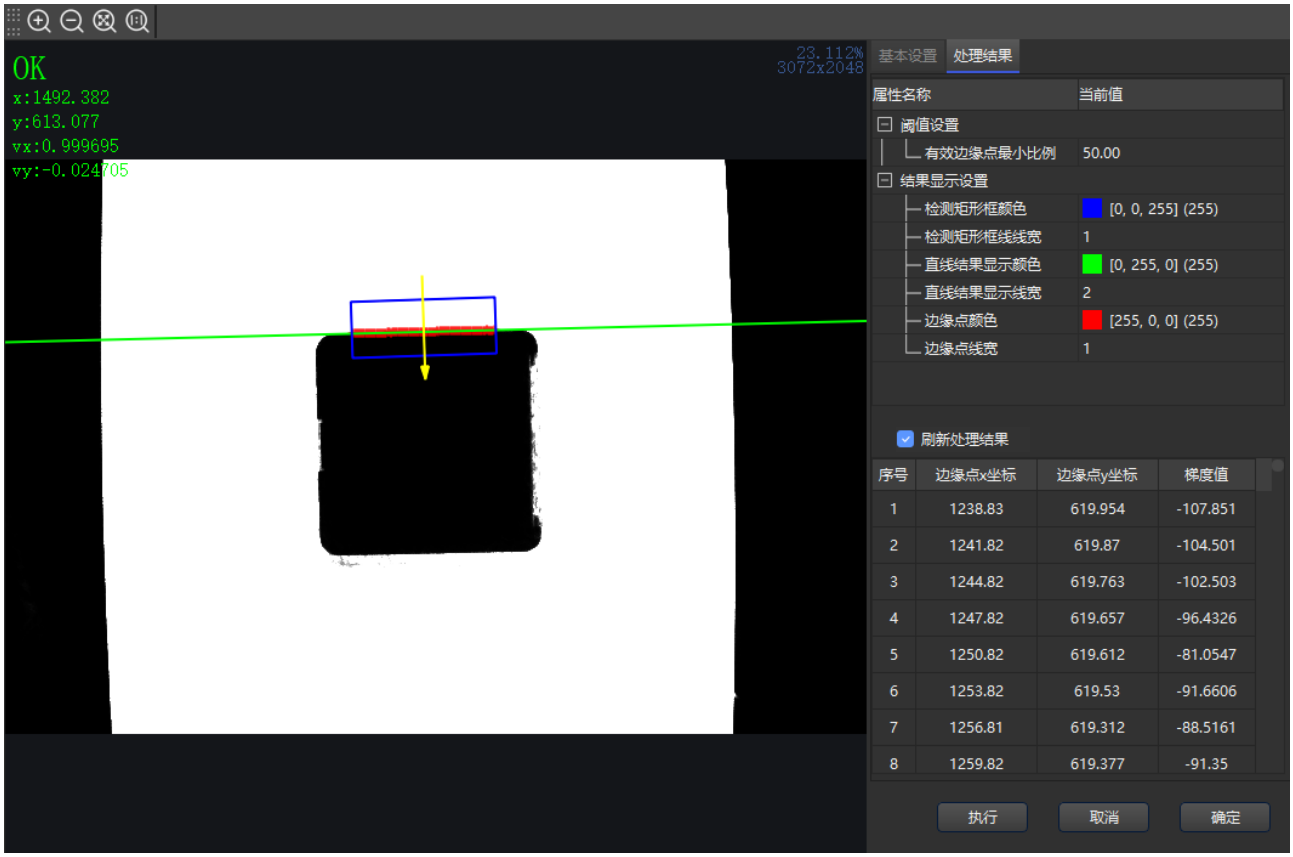
➢ 灰度平滑滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。

➢ 边缘梯度滤波核：卡尺测量线方向上的滤波核大小，用于增强边缘和抑制噪音干扰，最小值为 1，当边缘模糊不清晰或有噪音干扰时可以增大滤波核，这样可以使得检测结果更加稳定。

- 边缘梯度滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。
- 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。
- 最大梯度阈值：设定检测边缘梯度阈值的上限，只有边缘梯度阈值小于该值的边缘点才被检测到，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多条直线时，可修改该序号调整要检测的直线。
- 边缘置信度：当边缘点序号设置为 0 时有效，边缘点梯度的权重占该卡尺区域梯度最大值的比重。
- 直线拟合方式：拟合方式有标准代数最小二乘和带权重的最小二乘两种。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

- 处理结果页

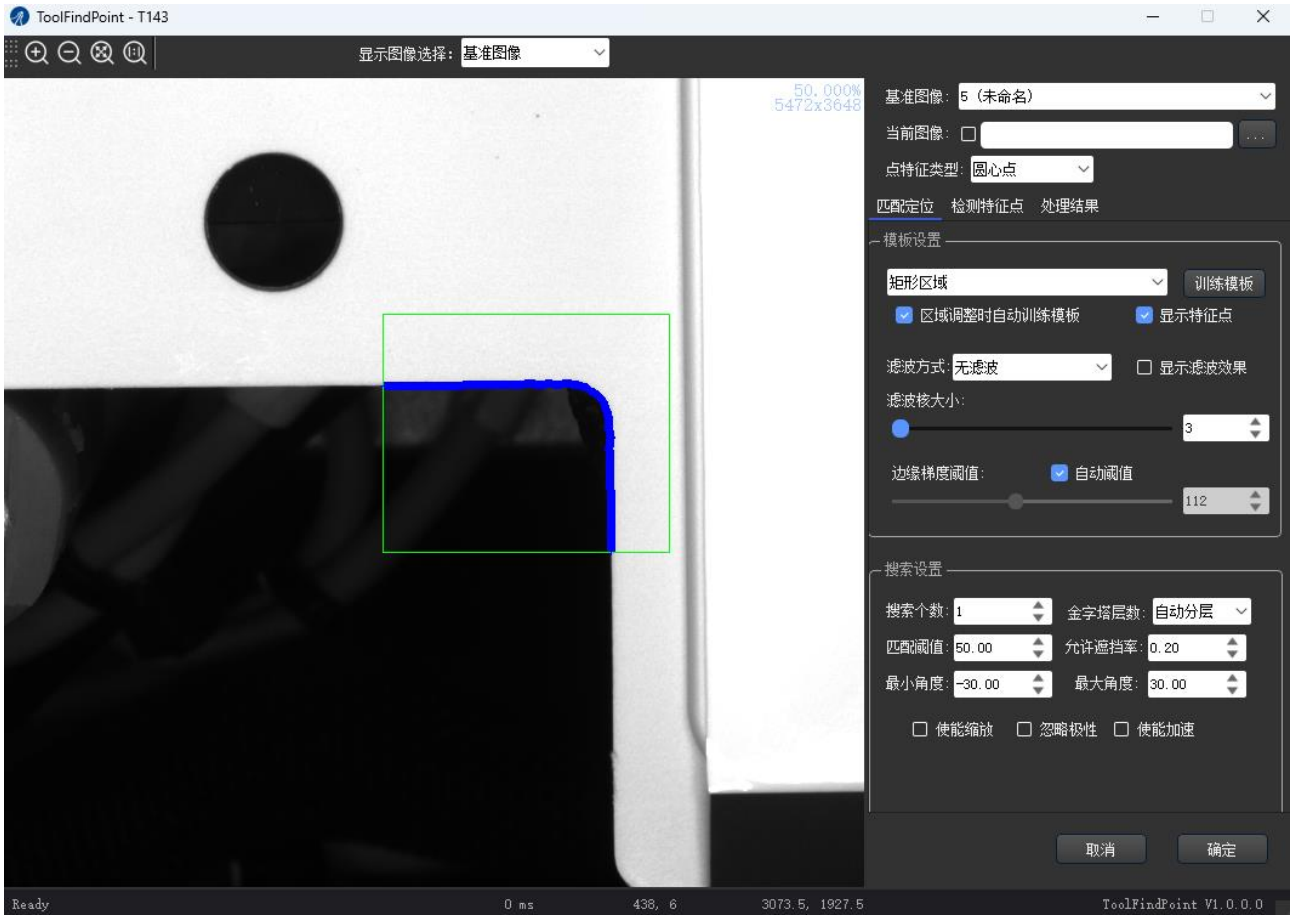
查看直线查找结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果、矩形框中心点到该直线的垂足坐标和该直线的方向向量。



- 阈值设置：当有效边缘点最小比例在设定的范围内时，工具状态结果为 True。
- 结果显示设置：检测矩形框颜色、拟合的直线颜色、边缘点颜色及线宽设置。
- 序号：边缘点编号。
- 边缘点坐标：所有检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：所有检测到的边缘点的边缘梯度阈值。

6.4.5 特征点定位

基于模板匹配查找图片中的圆心点或直角点并输出该特征点的坐标。



该算子工具共 3 个设置页，分别为：匹配定位、检测特征点、处理结果。

- 匹配定位页

  主要用于创建模板及设置匹配参数，参数设置请参考轮廓匹配章节。

- 检测特征点页

  设置检测的特征点类型、参数及 ROI 区域。

点特征类型：可选择圆心点和直角点。选择圆心点时，参数设置请参考圆查找章节；选择直角点时，参数设置请参考直线查找章节。

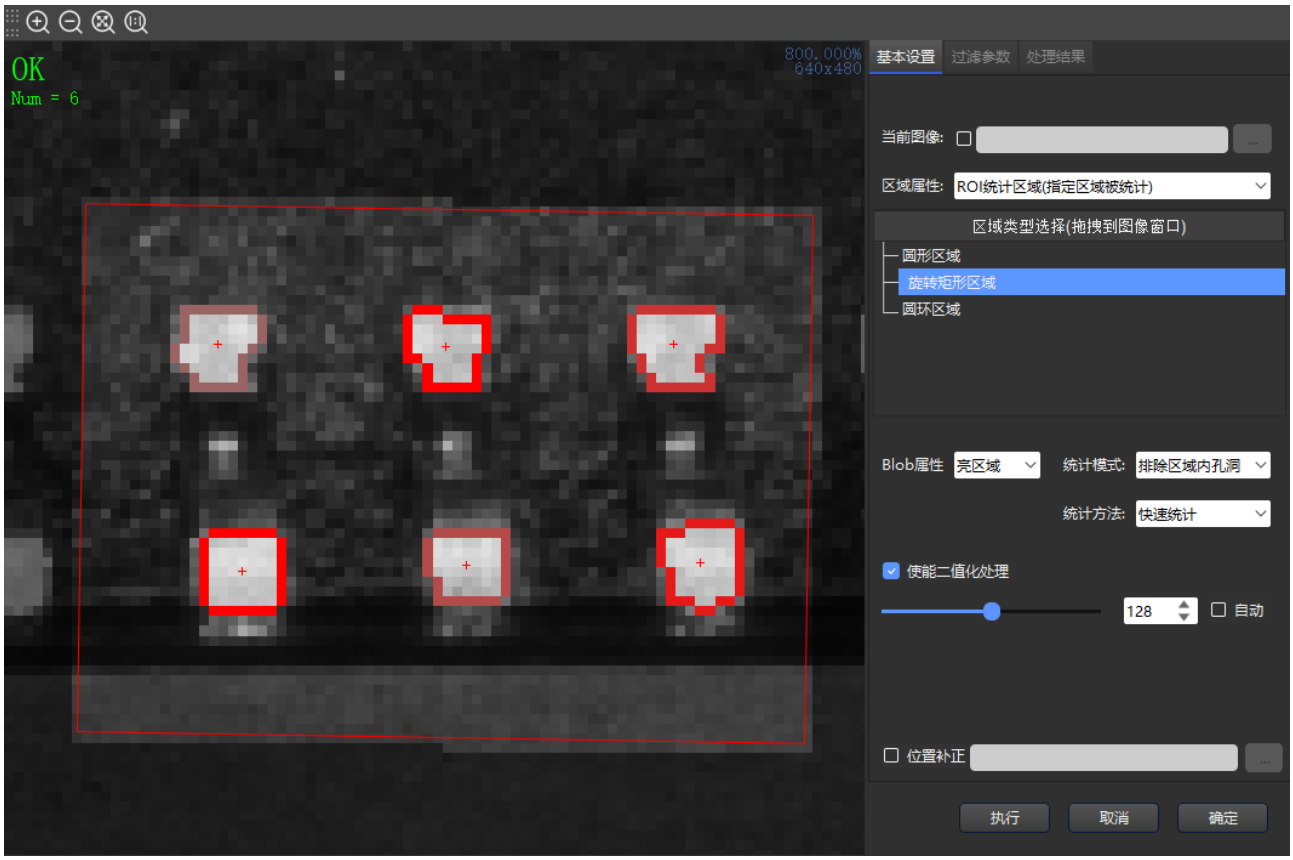
- 处理结果页

  查看特征点结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的结果及特征点坐标。

特征点为圆心点时，参数说明请参考圆查找章节，特征点为直角点时，参数说明请参考线线交点章节。

6.4.6 BLOB 分析

将图像指定区域二值化处理后统计分析亮区域或暗区域的信息，如面积、轮廓周长、重心坐标、外接矩形、最小外接圆等。一般用于对图像中的缺陷进行检测或定位。



该算子工具共 3 个设置页，分别为：基本设置、过滤参数和处理结果。

- 基本设置页

  主要用于设置 BLOB 分析的 ROI 区域。

➢ 当前图像：算子运行时进行 BLOB 分析的图像，一般绑定图像采集、图像处理等工具的输出图像。

➢ 区域属性：设定要检测的 Blob 区域为统计对象或掩膜对象。

➢ 区域类型选择：选择 ROI 形状（圆、旋转矩形、圆环）。

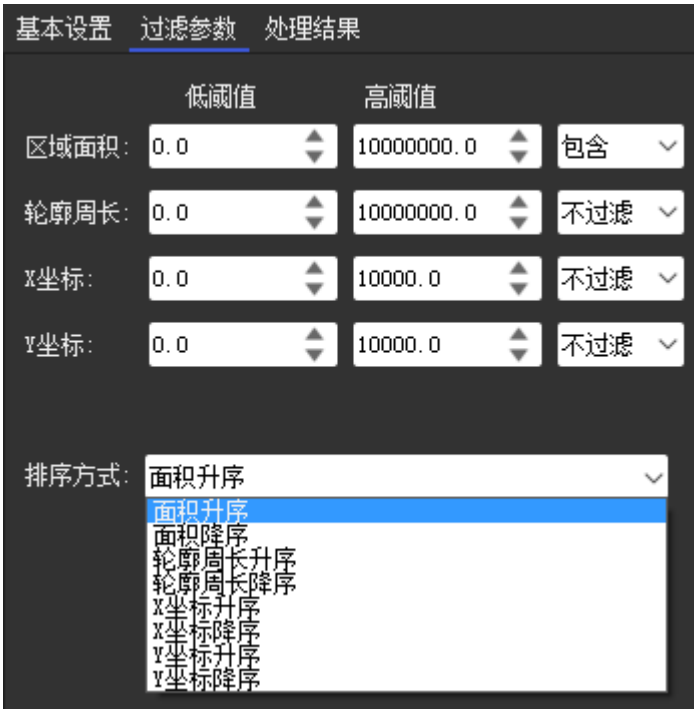
➢ Blob 属性：可以设置当前需要提取的目标为亮区域（白色）或暗区域（黑色）。

➢ 统计模式：计算区域面积时，是否计算区域内的孔洞，分为排除区域内孔洞和包含区域内孔洞。

- 统计方法：快速统计和逐像素精确统计两种方式。快速统计方式速度快但有一定偏差，逐像素精确统计方式速度稍慢但精准。
- 使能二值化处理：设置二值图像的分割阈值，当像素灰度值大于等于该值为白色（亮区域），否则为黑色（暗区域）。
- 自动：根据直方图分布自动计算出最佳分割阈值。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

- 过滤参数页

设置目标每种特征的最小值和最大值，当某个目标的特征不在该范围值内会被移除还是保留。

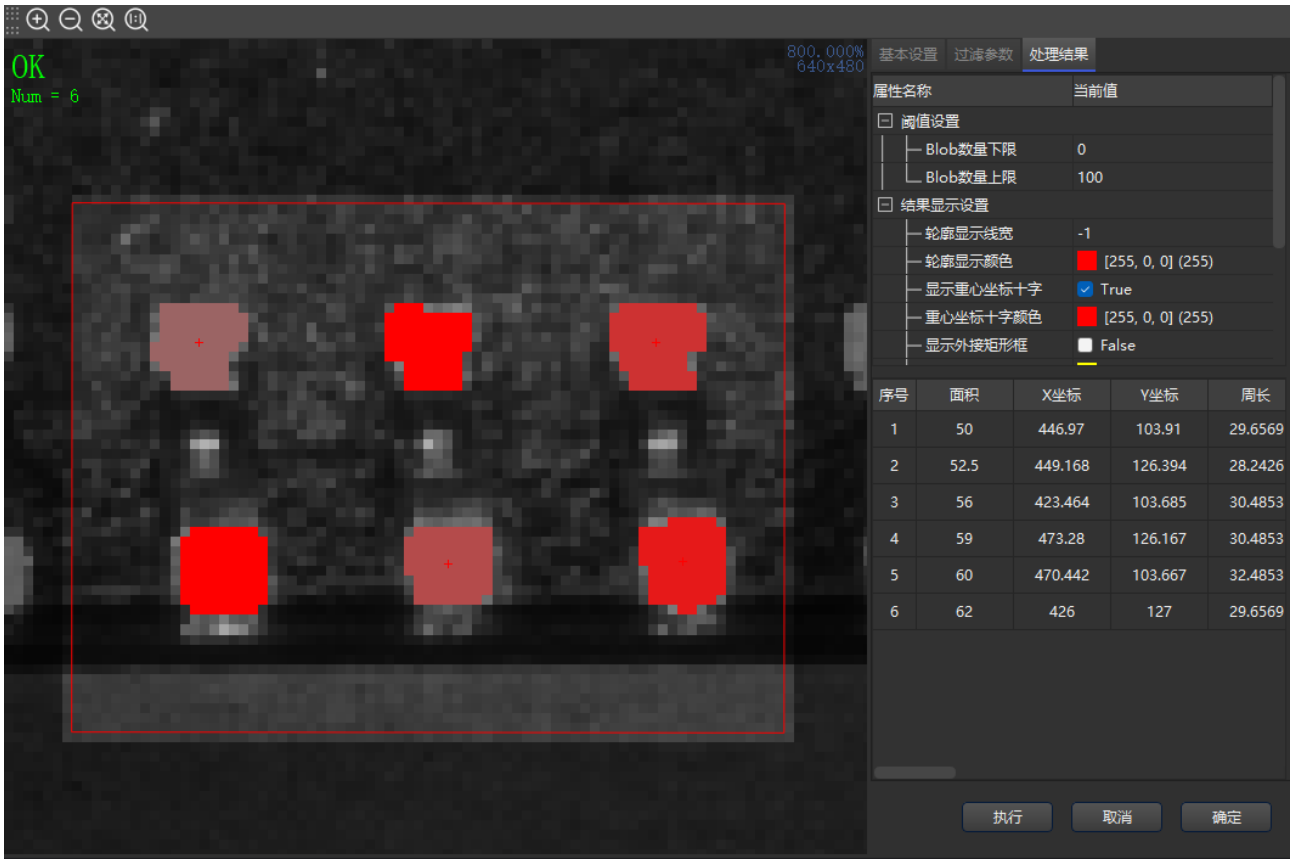


- 区域面积：设置目标面积过滤参数，不在低阈值~高阈值范围内的目标将被排除或保留，被排除的目标将不会再参与特征的计算。
- 轮廓周长：设置目标轮廓周长过滤范围。
- X 坐标：设置目标 X 坐标过滤范围。
- Y 坐标：设置目标 Y 坐标过滤范围。

- 排序方式：设置提取到的目标按面积、轮廓、坐标的升序或降序排列。

- 处理结果页

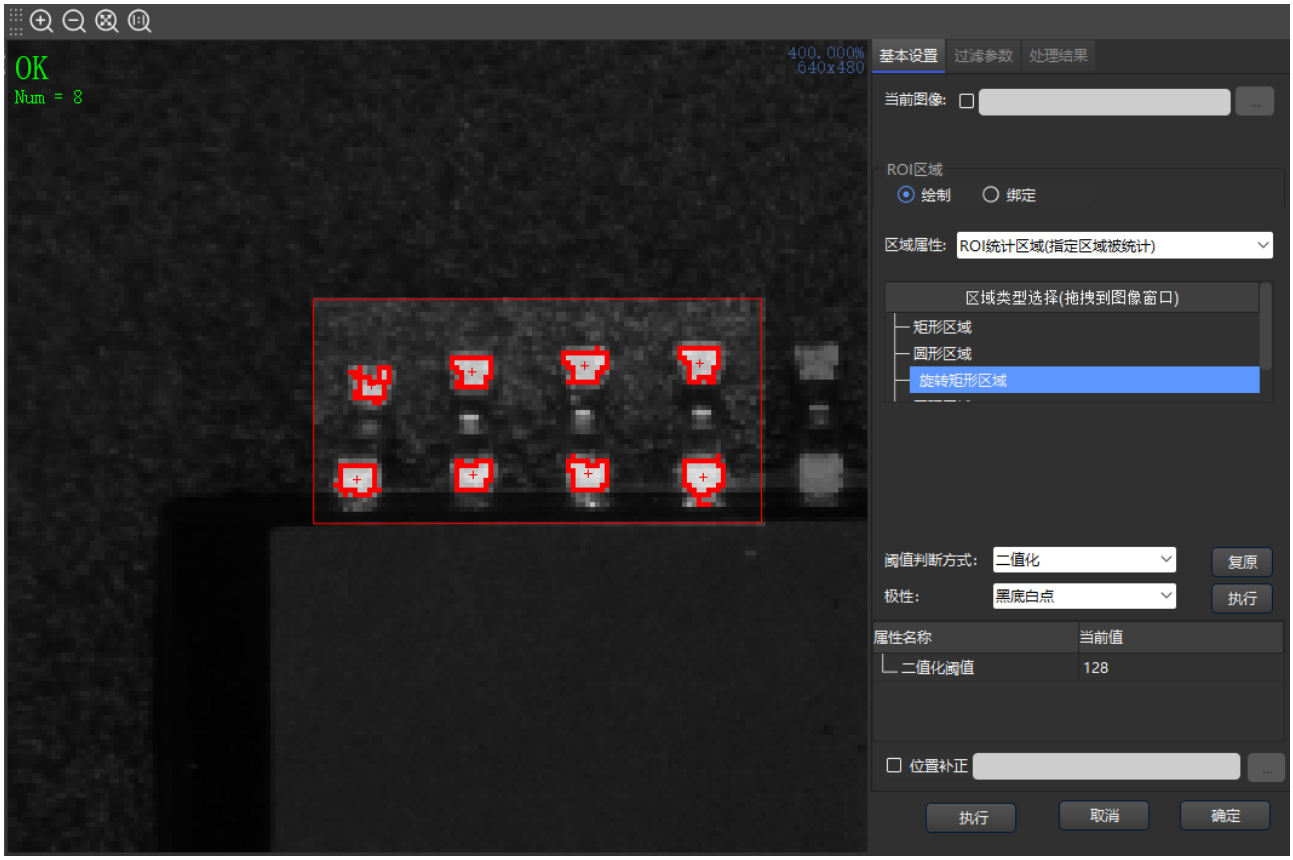
查看 BLOB 分析结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果。



- 阈值设置：当 Blob 数量的上下限在设定的范围内时，工具状态结果为 True。
- 结果显示设置：轮廓颜色、重心十字标颜色、最小外接矩形框颜色及最小外接矩形圆颜色的设置。
- 序号：目标编号。
- XY 坐标：重心坐标，即目标对象所有像素坐标的平均值。
- 周长：目标对象的周长。
- 最小外接矩形：目标对象的最小外接矩形。
- 最小外接圆：目标对象的最小外接圆。

6.4.7 连通域分析

连通域即图像上互相连通的像素点构成的区域。连通域分析是将图像指定区域二值化处理后统计分析亮区域或暗区域的信息，如面积、轮廓周长、重心坐标、外接矩形、最小外接圆等。一般用于对图像中的缺陷进行检测或定位。图像二值化指将图像中的像素点转换为纯黑（灰度值 0）或纯白（灰度值 255）。转换后，图像被分割为黑白两部分。转换过程中，灰度值转换为 0 和 255 的像素点具体由预设的像素阈值决定。



该算子工具共 3 个设置页，分别为：基本设置、过滤参数和处理结果。

- 基本设置页

  主要用于设置连通域分析的 ROI 区域。

  - 当前图像：算子运行时进行连通域分析的图像，一般绑定图像采集、图像处理等工具的输出图像。
  - ROI 区域属性：绘制模式，绘制 ROI 设定要检测的区域为统计对象或掩膜对象；绑定模式，绑定其他工具输出的圆、矩形作为统计对象。
  - 区域类型选择：绘制模式时，选择 ROI 形状（圆、旋转矩形、圆环）。

- 阈值判断方式：二值化、自动阈值和双阈值三种方式。

二值化

属于单阈值二值化，输入图像像素点的灰度值大于阈值时，则为目标（灰度值转换为 255），否则则为背景（灰度值转换为 0）。

自动阈值

算法会遍历所有可能阈值，从而找到最佳的阈值 T，并根据该阈值进行图像二值化。输入图像中灰度值大于 T 的像素点为目标，小于的为背景。该方式无预设参数。

双阈值

根据预设的低阈值和高阈值进行图像二值化。输入图像像素点的灰度值大于低阈值或小于高阈值时，则为目标（灰度值转换为 255），否则则为背景（灰度值转换为 0）。

- 极性：可以设置当前需要提取的目标为白底黑点和黑底白点。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

- 过滤参数页

设置目标每种特征的最小值和最大值，当某个目标的特征不在该范围值内会被移除还是保留。



- 区域面积：设置目标面积过滤参数，不在低阈值~高阈值范围内的目标将被排除或保留，被排除的目标将不会再参与特征的计算。
- 轮廓周长：设置目标轮廓周长过滤范围。
- X 坐标：设置目标 X 坐标过滤范围。
- Y 坐标：设置目标 Y 坐标过滤范围。
- 排序方式：设置提取到的目标按面积、轮廓、坐标的升序或降序排列。
- 连通性：4 连通和 8 连通。

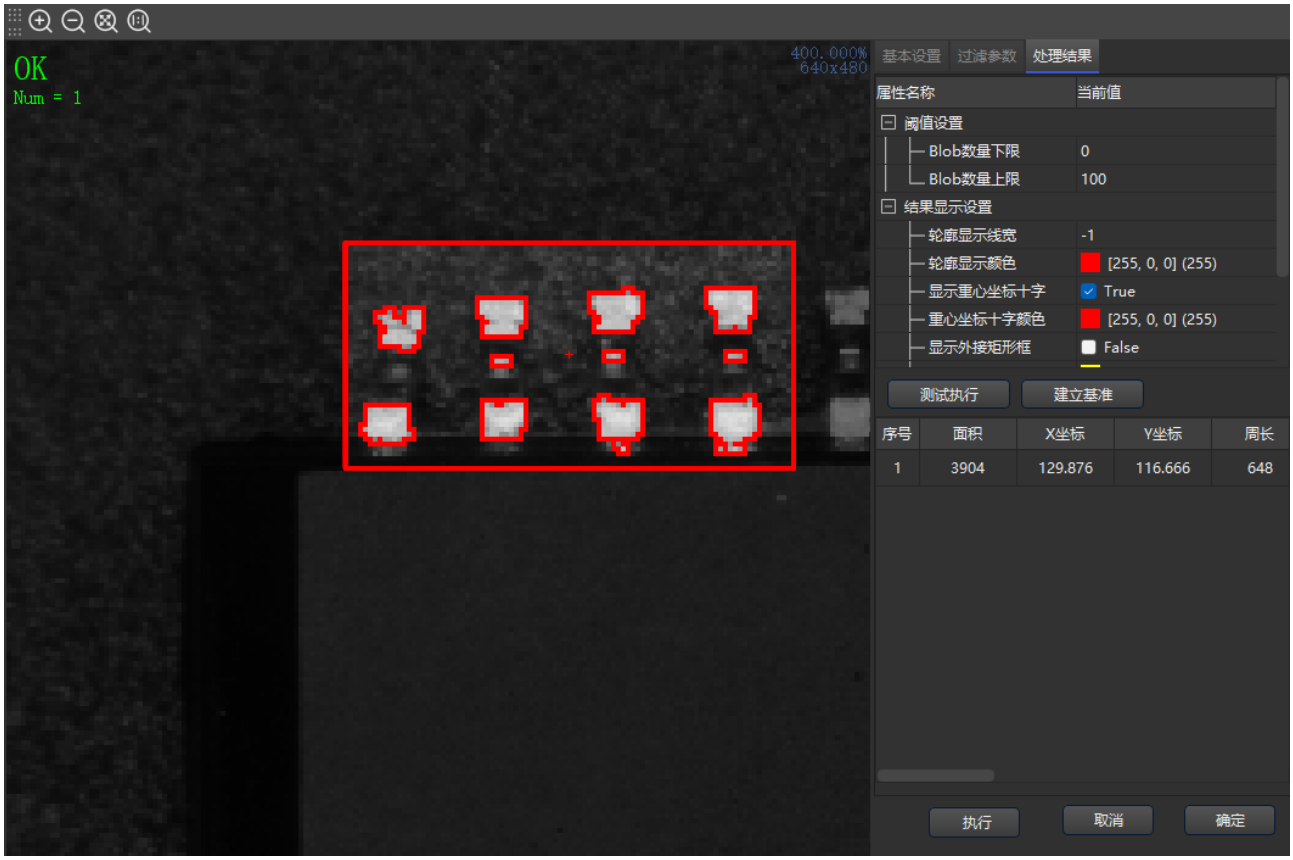
  4 连通：两个像素点有共同的边缘，即一个像素点在另一个像素的上方、下方、左侧或右侧。

8 连通：在 4 连通的基础上，将对角线上的相邻像素也纳入为“互相连通”的像素。

- 查找数量：设置最终输出的连通域分析的最大数量。
- 圆度：检测出的各个目标的圆形度。
- 矩形度：各个目标的面积与其相对应的最小外接矩形面积的比值。
- Box 宽：ROI 检测区域的宽度。
- Box 高：ROI 检测区域的高度。
- Box 角度：ROI 检测区域相对于水平方向的旋转角度。顺时针旋转，角度为正；逆时针旋转，角度为负。
- 外接矩形宽：外接矩形的宽度。
- 外接矩形高：外接矩形的高度。

- 处理结果页

查看 BLOB 分析结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果。

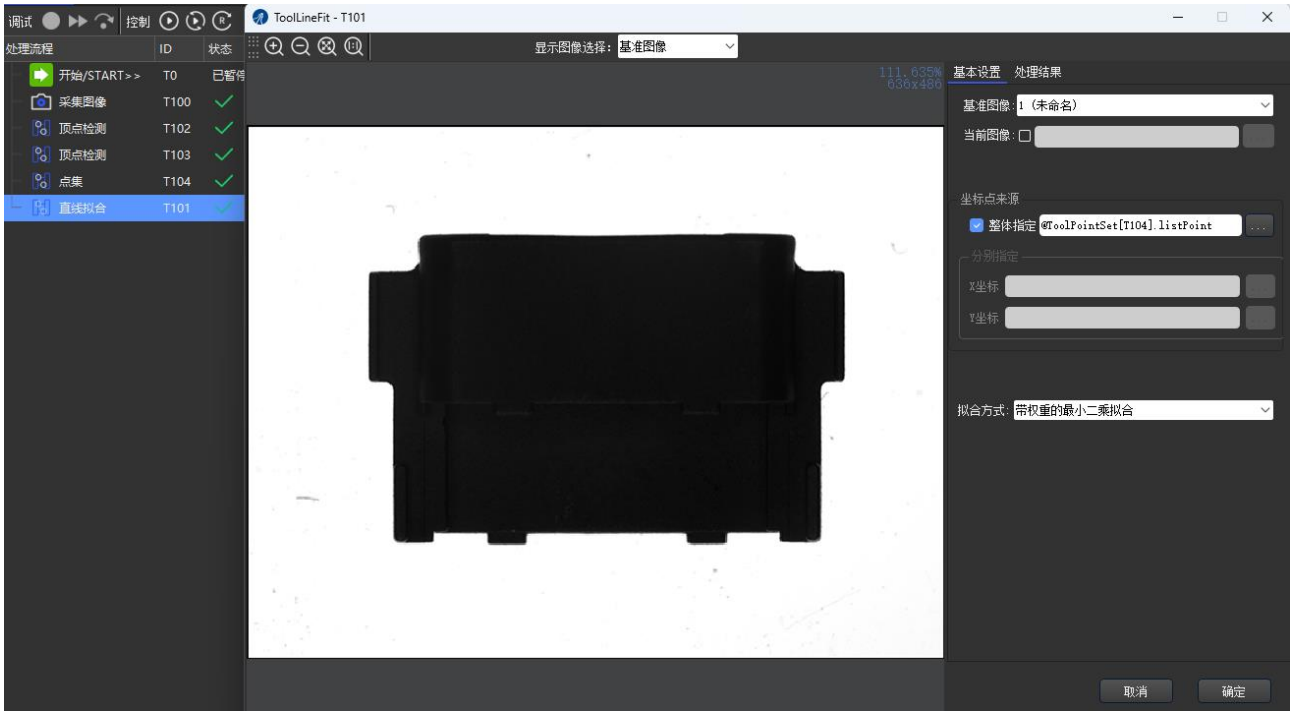


- 阈值设置：当 Blob 数量的上下限在设定的范围内时，工具状态结果为 True。

- 结果显示设置：轮廓颜色、重心十字标颜色、最小外接矩形框颜色及最小外接矩形圆颜色的设置。
- 建立基准：将当前连通域分析的重心作为基准，与位置补正工具配合使用。
- 序号：目标编号。
- XY 坐标：重心坐标，即目标对象所有像素坐标的平均值。
- 周长：目标对象的周长。
- 最小外接矩形：目标对象的最小外接矩形。
- 圆度：目标对象的圆形度。
- 矩形度：目标对象的面积与其相对应的最小外接矩形面积的比值。

6.4.8 直线拟合

检测两个及以上的特征点形成点集，然后拟合成一条直线，输出点集坐标和直线的方向向量，往往用在直线查找或顶点检测提取特征点之后的阶段。



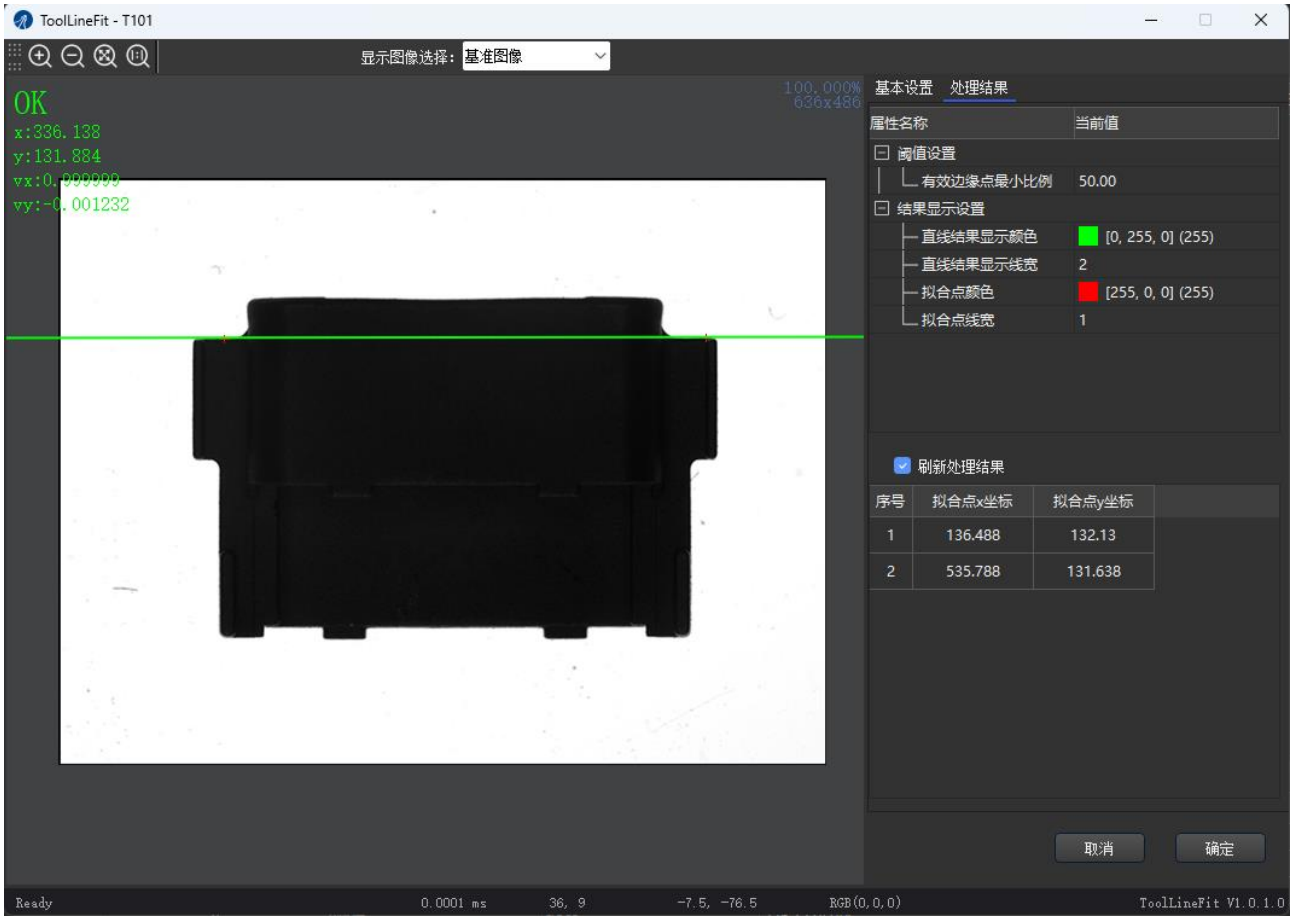
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置拟合直线的点集来源。

- 当前图像：算子运行时拟合直线的图像，一般绑定图像采集、图像处理等工具的输出图像。

- 坐标点来源：设置拟合直线的点集来源，与点集工具或其他输出特征点集合的工具配合使用。整体指定：按点方式，分别指定：按坐标方式。
- 直线拟合方式：拟合方式有标准代数最小二乘和带权重的最小二乘两种。



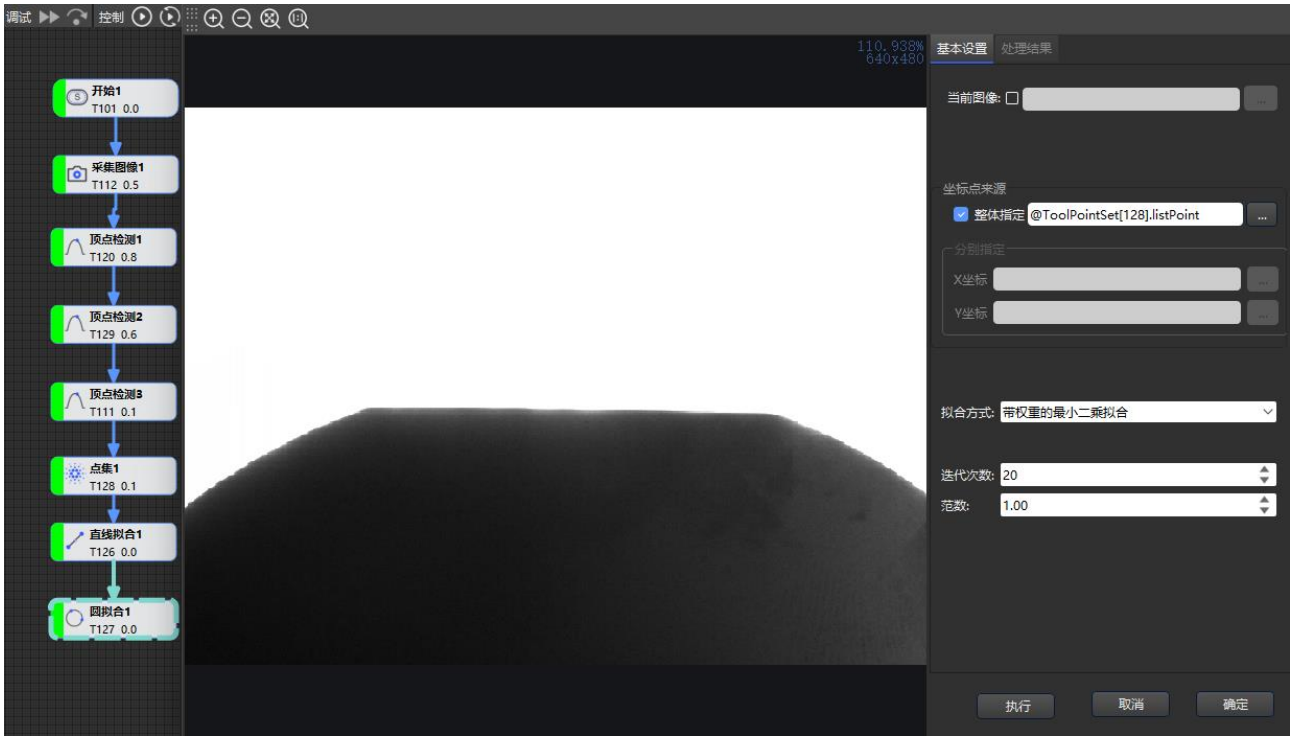
- 处理结果页

查看直线结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果、点集的中点坐标和该直线的方向向量。

- 阈值设置：当有效边缘点最小比例在设定的范围内时，工具状态结果为 True。
- 结果显示设置：点集颜色、大小、拟合的直线颜色及线宽设置。
- 序号：点集编号。
- 拟合点坐标：所有特征点在当前图像的 XY 坐标位置。

6.4.9 圆拟合

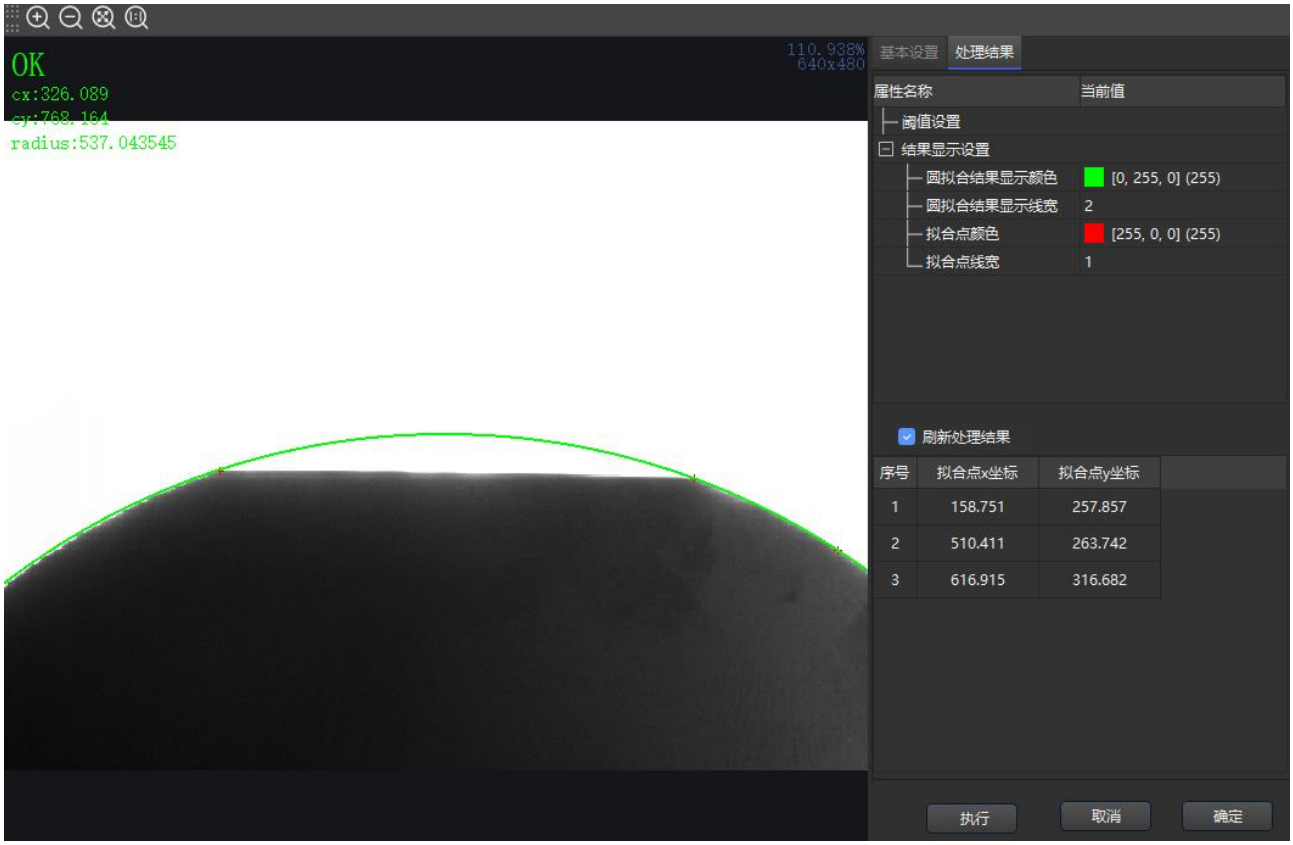
检测 3 个及以上的特征点形成点集，然后拟合成一个圆，输出点集坐标、圆心坐标和半径，往往用在圆查找或顶点检测提取特征点之后的阶段。



该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置拟合圆的点集来源。参数设置请参考直线拟合章节。



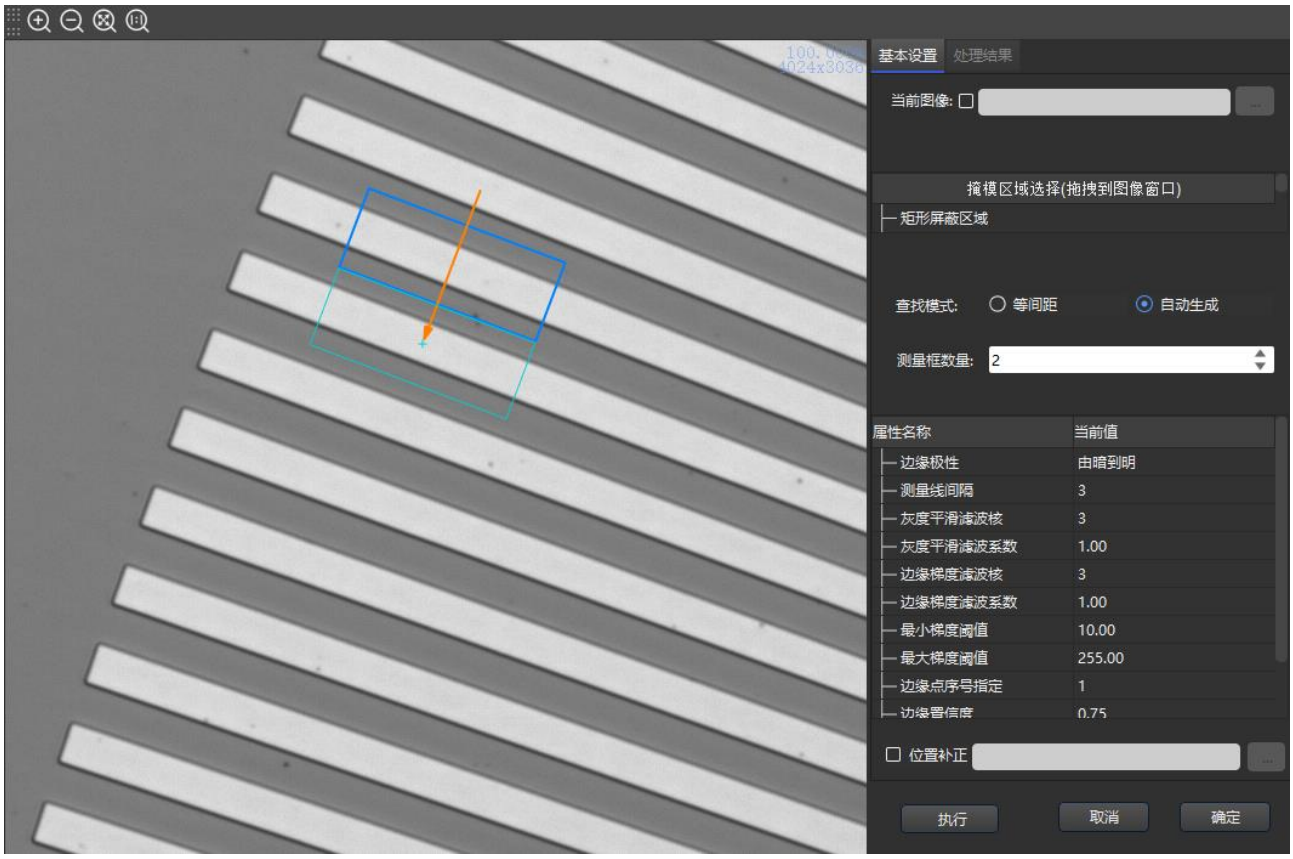
- 处理结果页

查看圆结果及显示结果设置。其中图形显示窗左上角输出该工具的状态结果、拟合圆的圆心坐标和半径。

- 阈值设置：当有效边缘点最小比例在设定的范围内时，工具状态结果为 True。
- 结果显示设置：点集颜色、大小、拟合的圆颜色及线宽设置，当线宽设置为 0 时，不显示。
- 序号：点集编号。
- 拟合点坐标：所有特征点在当前图像的 XY 坐标位置。

6.4.10 多直线查找

同时查找一个 ROI 区域内的多条直线，可用于硅片间距、芯片引脚间距测量。需要在同一区域定位多条直线时，如果使用直线查找工具，需要在流程中放置多个该工具。而如果使用多直线查找工具，仅需要放置单个该工具即可。

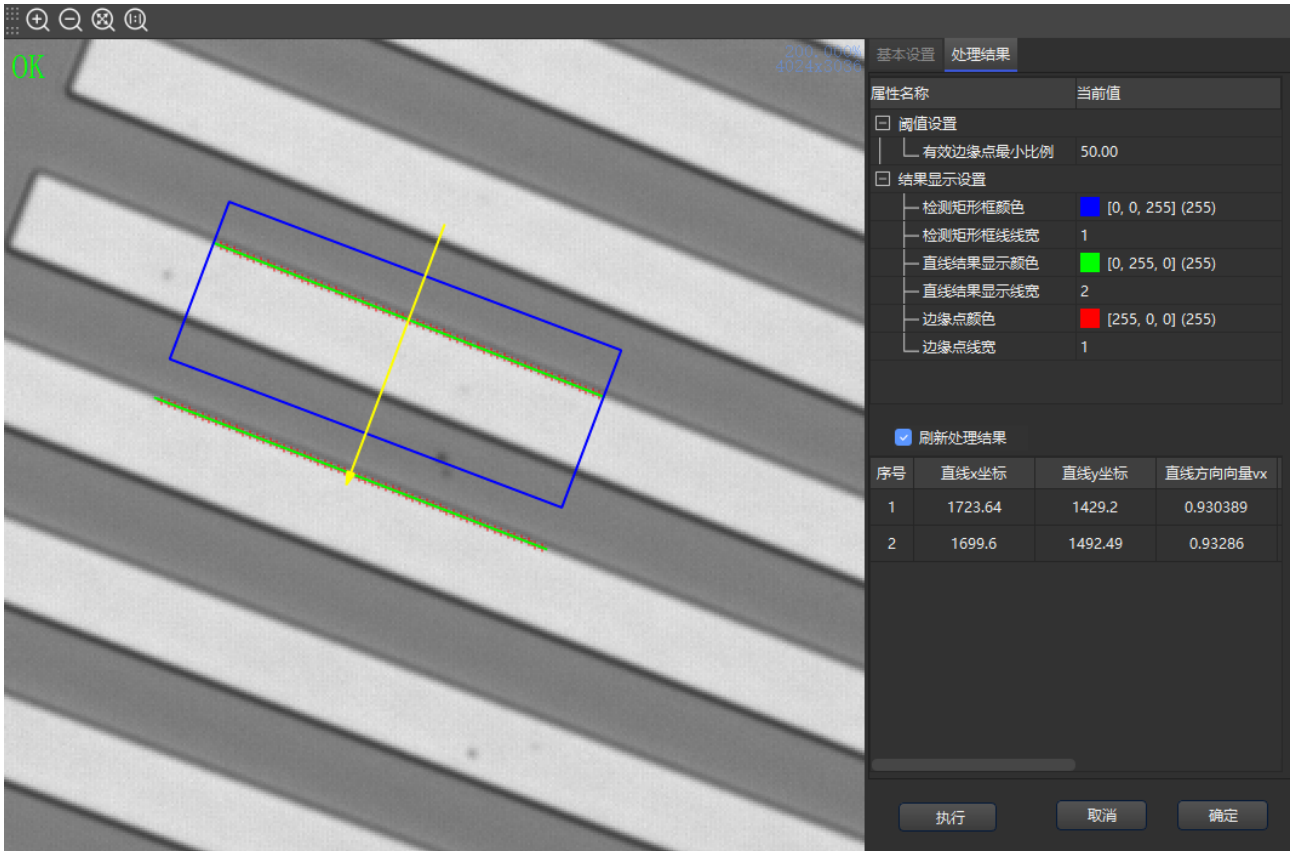


该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置多直线查找的 ROI 区域及测量参数。以下仅说明该工具特有的参数，其他参数设置请参考直线查找章节。

- 查找模式：①等间距，根据绘制的首个 ROI 中线，绘制**测量框数量**预设的 ROI 个数；②自动生成，根据自动生成，根据绘制的首个 ROI 大小自动生成**测量框数量**预设的相同尺寸 ROI 个数。
- 测量框数量：检测区域的数量。



- 处理结果页

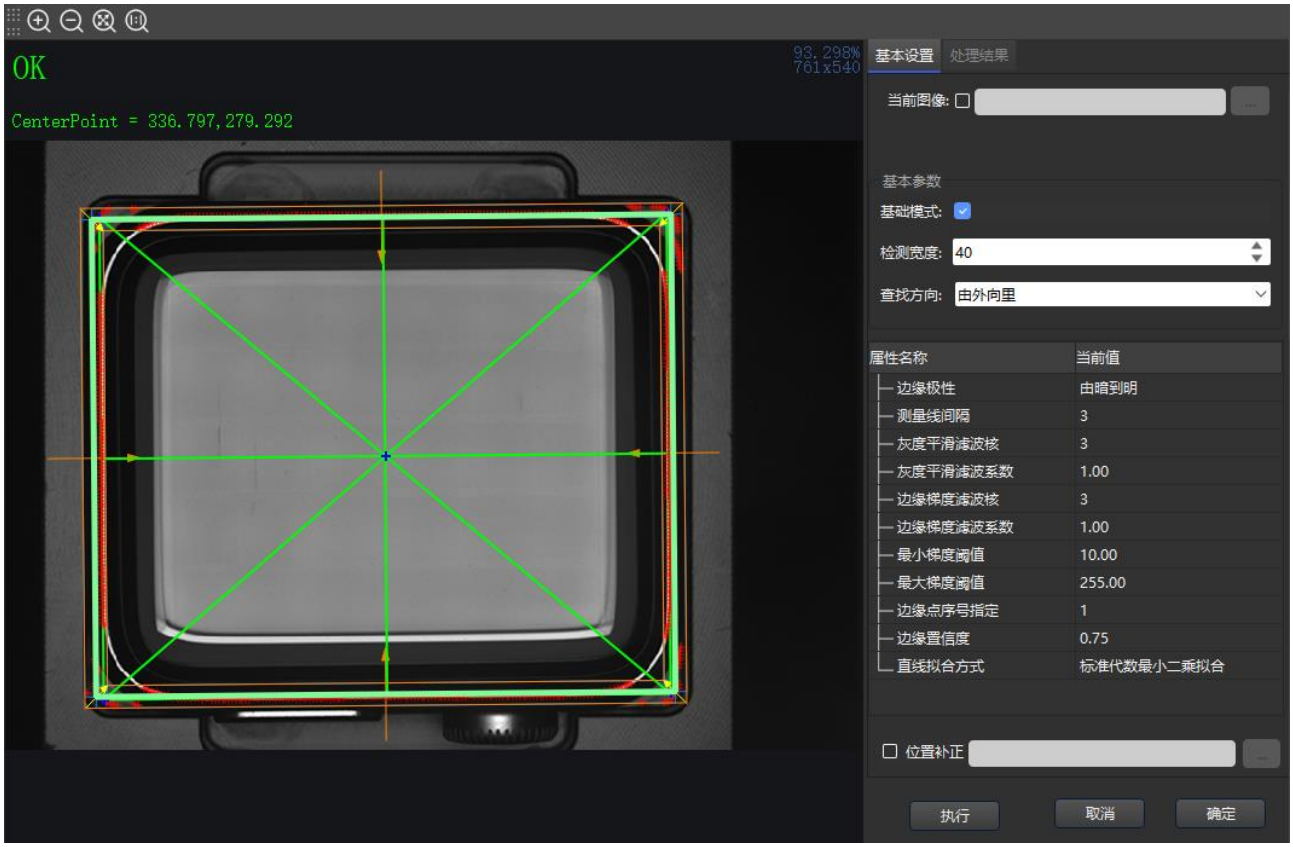
查看多直线查找的结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果。

- 阈值设置：当所有直线的有效边缘点最小比例在设定的范围内时，工具状态结果为 True。
- 结果显示设置：检测框颜色、直线颜色、边缘点颜色及线宽的设置，当线宽设置为 0 时，不显示。
- 序号：点集编号。
- 直线 X/Y 坐标：过直线的点坐标，ROI 边框与直线的两个交点的中点坐标。
- 直线方向向量 vx：直线单位向量 x 方向分量。

- 直线方向向量 vy：直线单位向量 y 方向分量。
- 有效点占比：参与直线拟合的点占总点数的比例。

6.4.11 四边形查找

检测图像中指定区域内是否存在四边形，输出四边形信息，包括四条直线、四边形的对角线交点及中点线交点。

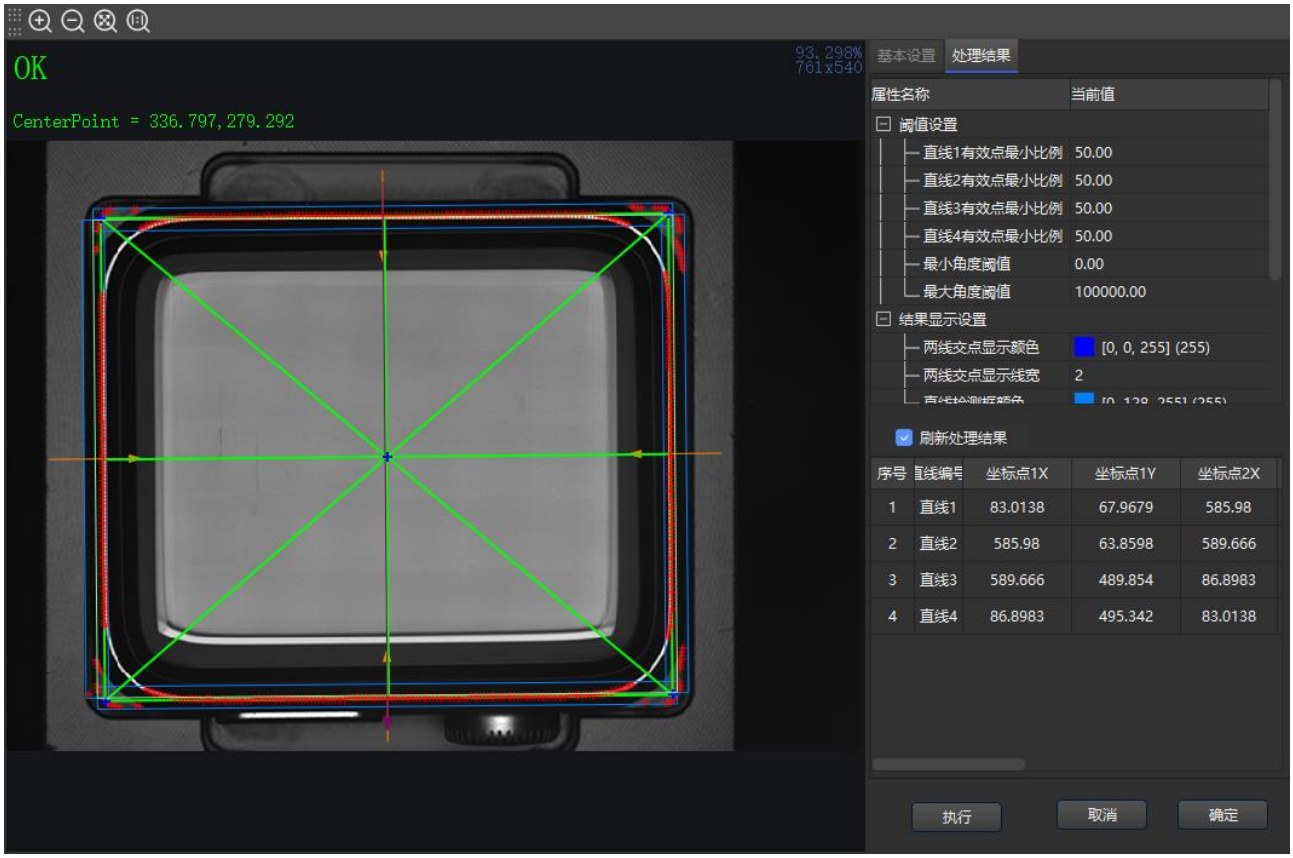


该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置四边形查找的 ROI 区域及测量参数。

- 当前图像：算子运行时搜索四边形的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 基础模式：勾选后 ROI 区域作为整体进行调整；不勾选，可单独调整四个 ROI 区域大小。
- 其他参数设置请参考直线查找章节。



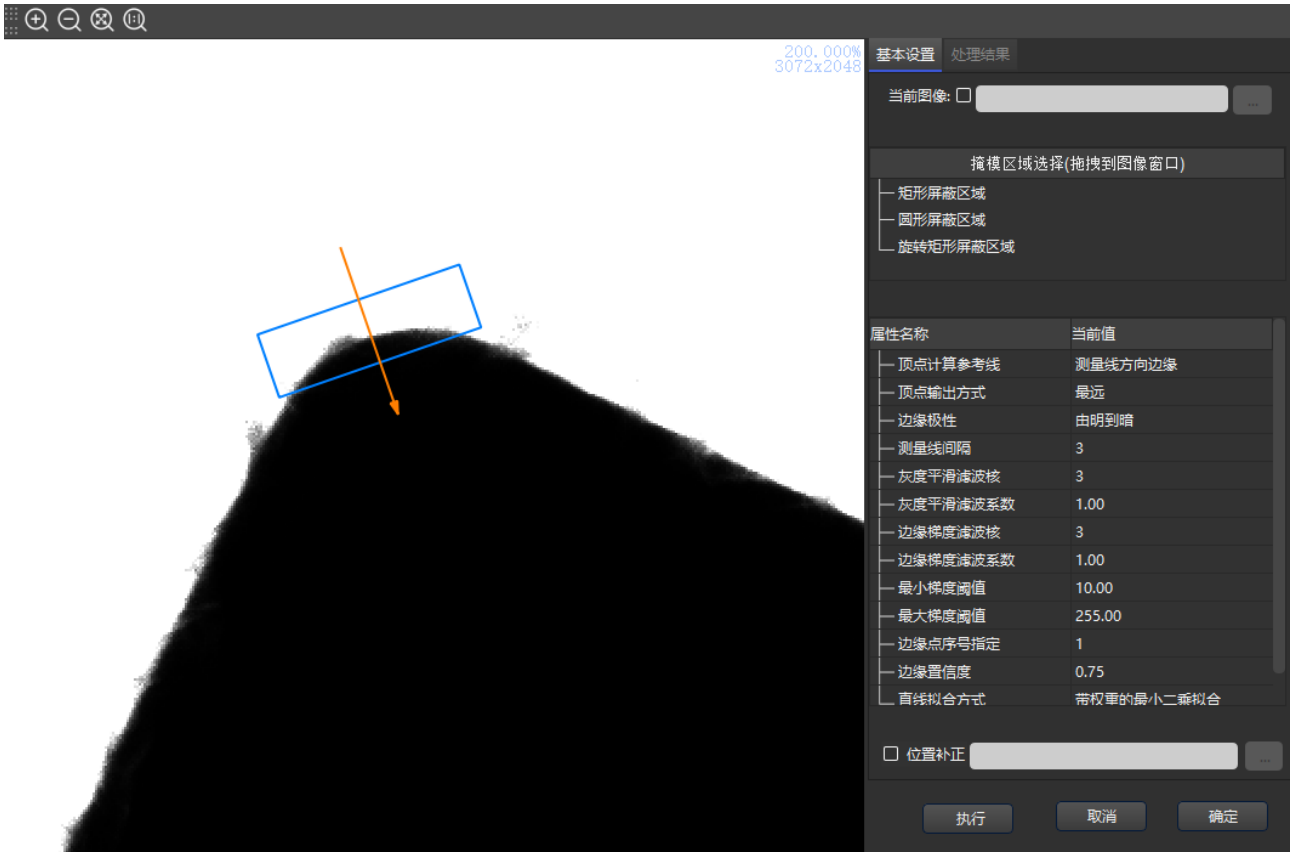
- 处理结果页

查看四条直线边的结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和对角线交点坐标。

- 阈值设置：当四条直线边的有效边缘点最小比例在设定的范围内时，工具状态结果为 True。
- 结果显示设置：两线交点颜色、交点线宽、检测框线宽及边缘点颜色、线宽的设置，当线宽设置为 0 时，不显示。
- 序号：点集编号。
- 坐标点 X/Y：直线的两个端点坐标，即相邻两直线边的交点在当前图像的坐标位置。
- 角度：直线边的角度，单位度。
- 有效点比例：参与直线拟合的点数占该直线的 ROI 检测区域总点数的比例。

6.4.12 顶点检测

按扫描方向检测图像指定 ROI 区域内的顶点，输出边缘点集合、顶点坐标、顶点梯度及顶点距离。

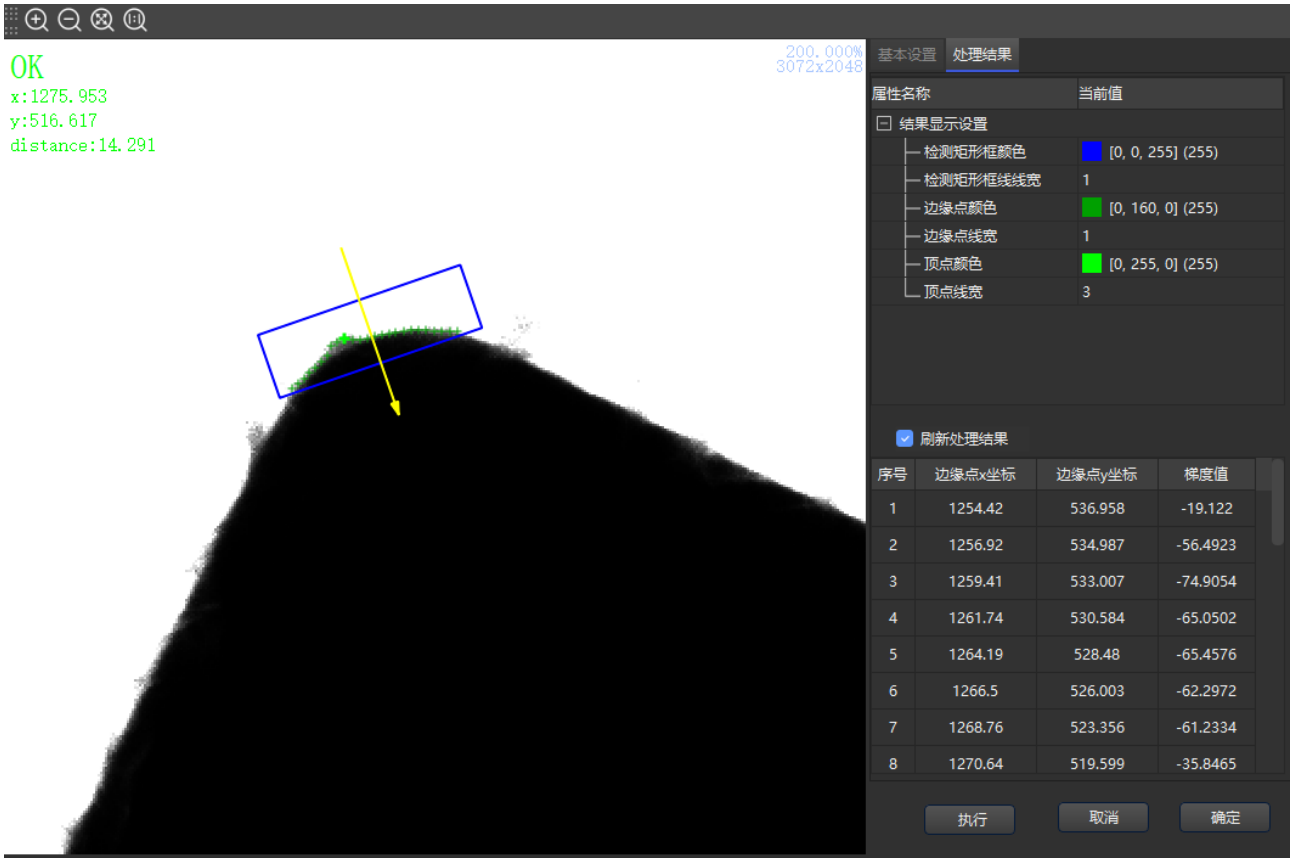


该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置顶点检测的 ROI 区域及测量参数。以下仅说明该工具特有的参数，其他参数设置请参考直线查找章节。

- 顶点计算参考线：①测量线方向边缘，点集沿测量线方向比较，离测量线指向的 ROI 矩形边缘的最大/最小距离的点；②图像 X 轴，点集沿 X 轴的垂线并指向 X 轴的方向比较，离 X 轴最大/最小距离的点；③图像 Y 轴，点集沿 Y 轴的垂线并指向 Y 轴的方向比较，离 Y 轴最大/最小距离的点。
- 顶点输出方式：距离顶点计算参考线的最近/最远点。



- 处理结果页

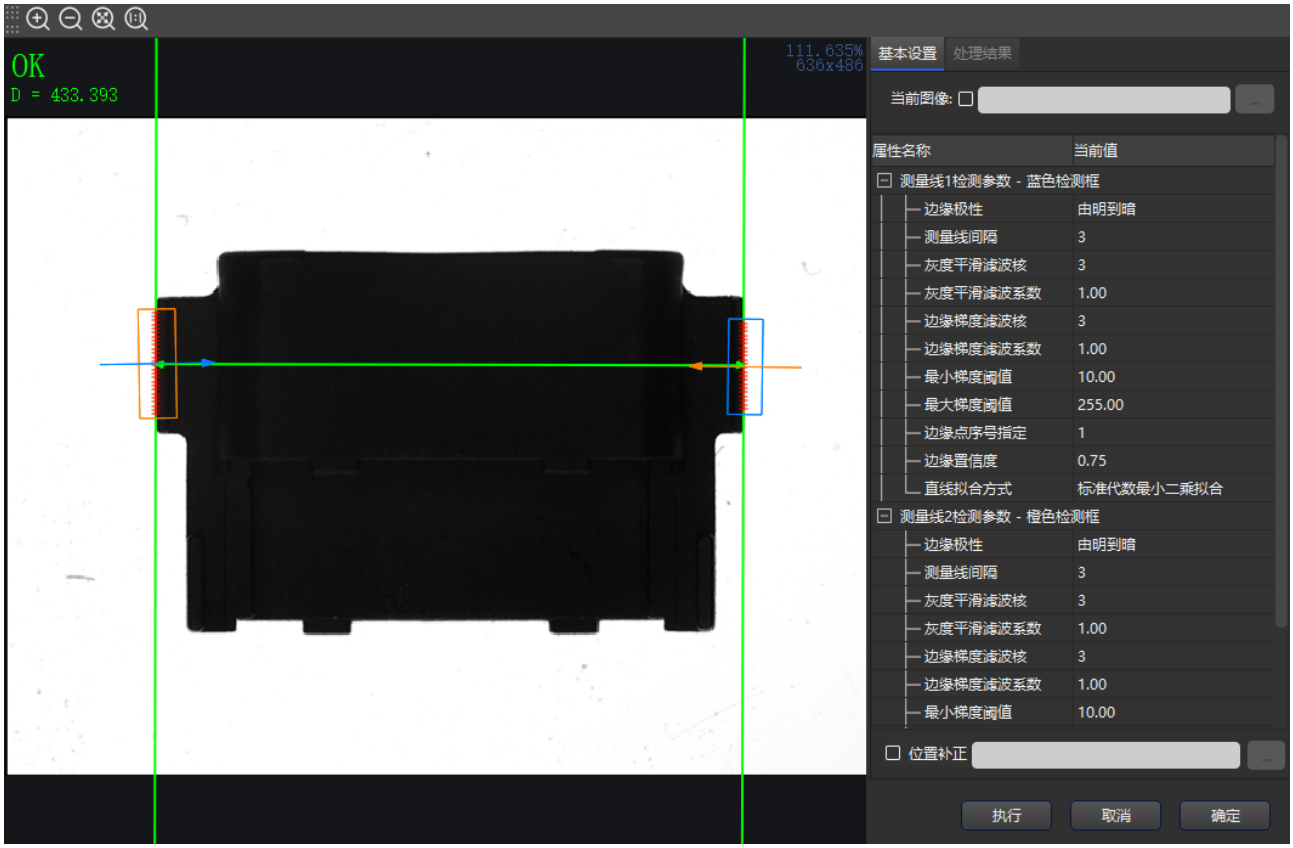
查看边缘点集合、检测的顶点及结果显示设置。其中图形显示窗左上角输出该工具的状态结果、顶点坐标、顶点距离及顶点梯度值。

- 结果显示设置：检测矩形框、边缘点及顶点的颜色、线宽的设置，当线宽设置为 0 时，不显示。
- 序号：点集编号。
- 边缘点坐标：所有检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：所有检测到的边缘点的边缘梯度阈值。

6.5 测量

6.5.1 线线距离

测量两条直线的距离。两条直线一般不会绝对平行，所以线线测量距离按照直线 1 的重心与直线 2 的重心生成的线段的中点，该中点到直线 1 的距离与到直线 2 的距离的总和。



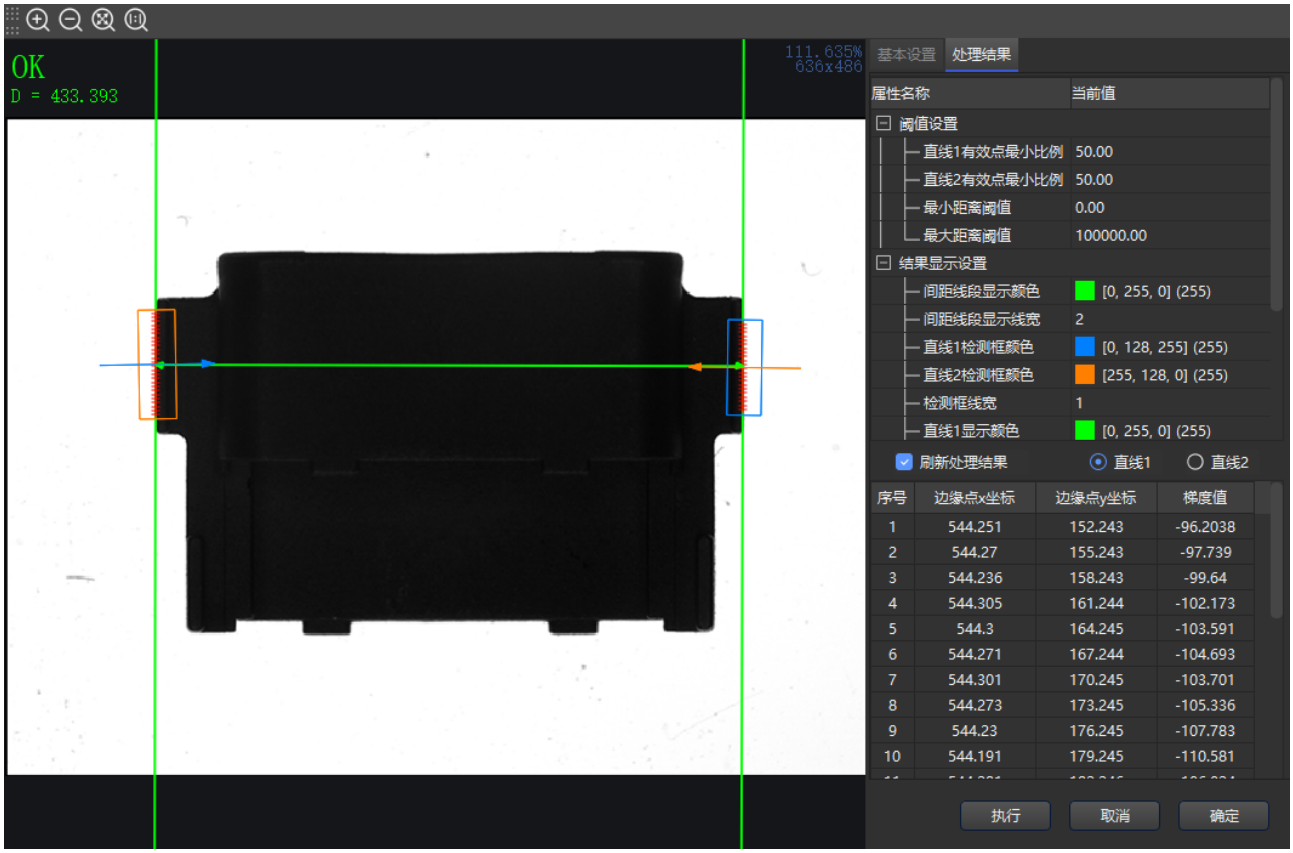
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置两条直线检测的测量参数。参数设置请参考直线查找章节。

- 处理结果页

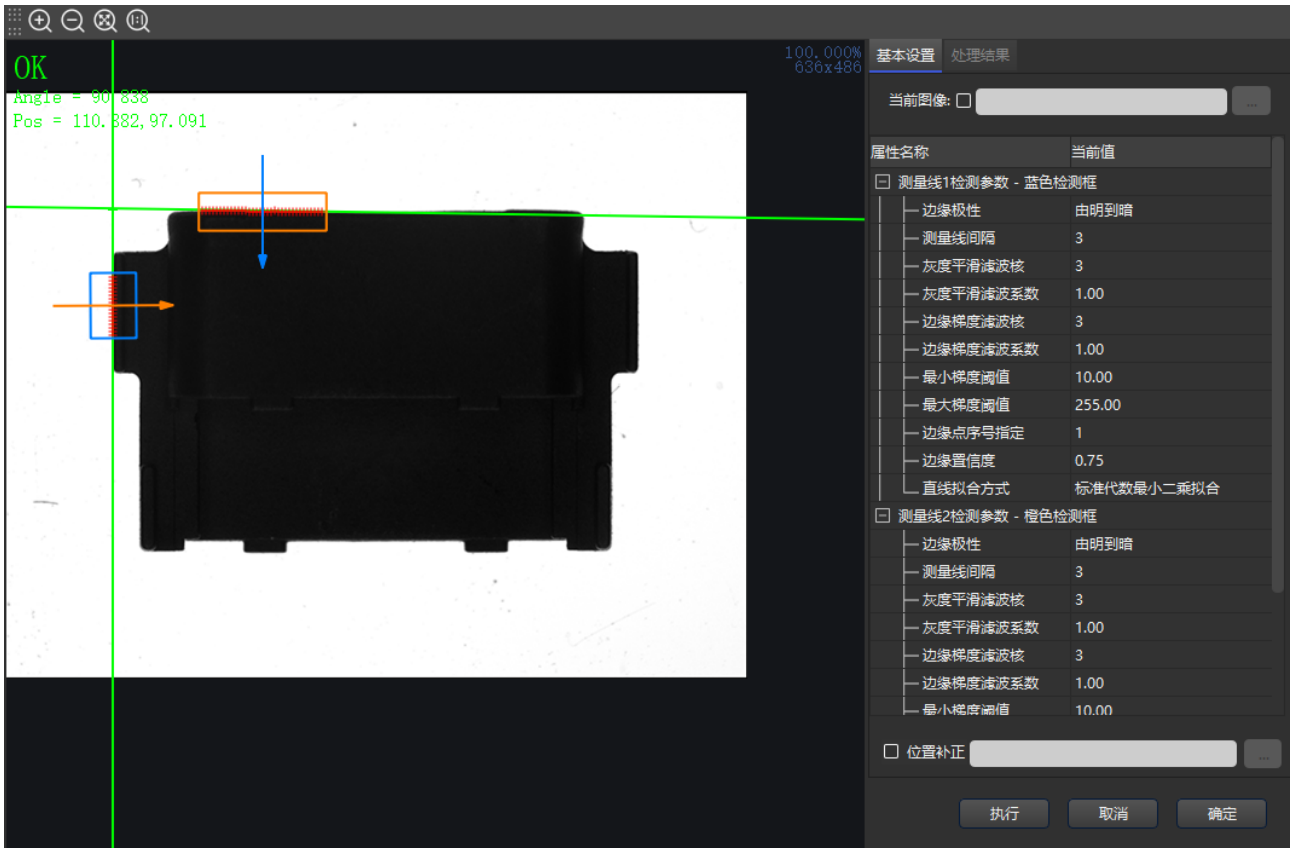
  查看线线测量结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和两直线的距离。



- 阈值设置：当两直线的有效点最小比例、最大最小距离阈值均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：两直线的间距线段颜色、两直线的检测框颜色、拟合的两条直线颜色、两直线边缘点颜色及线宽设置。
- 序号：每条直线的边缘点编号。
- 边缘点坐标：每条直线检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：每条直线检测到的边缘点的边缘梯度阈值。

6.5.2 线线交点

计算两条直线的交点和夹角。



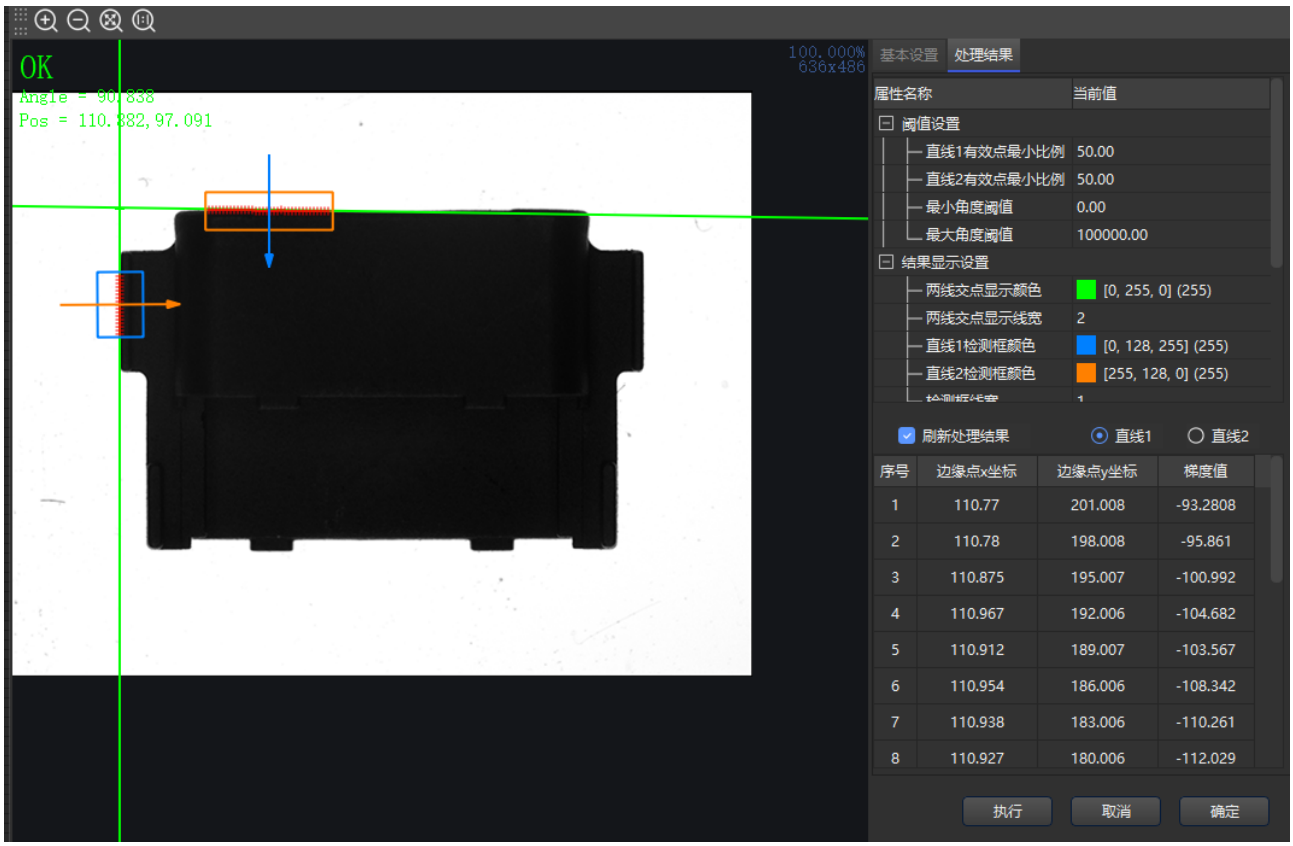
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置两条直线检测的测量参数。参数设置请参考直线查找章节。

- 处理结果页

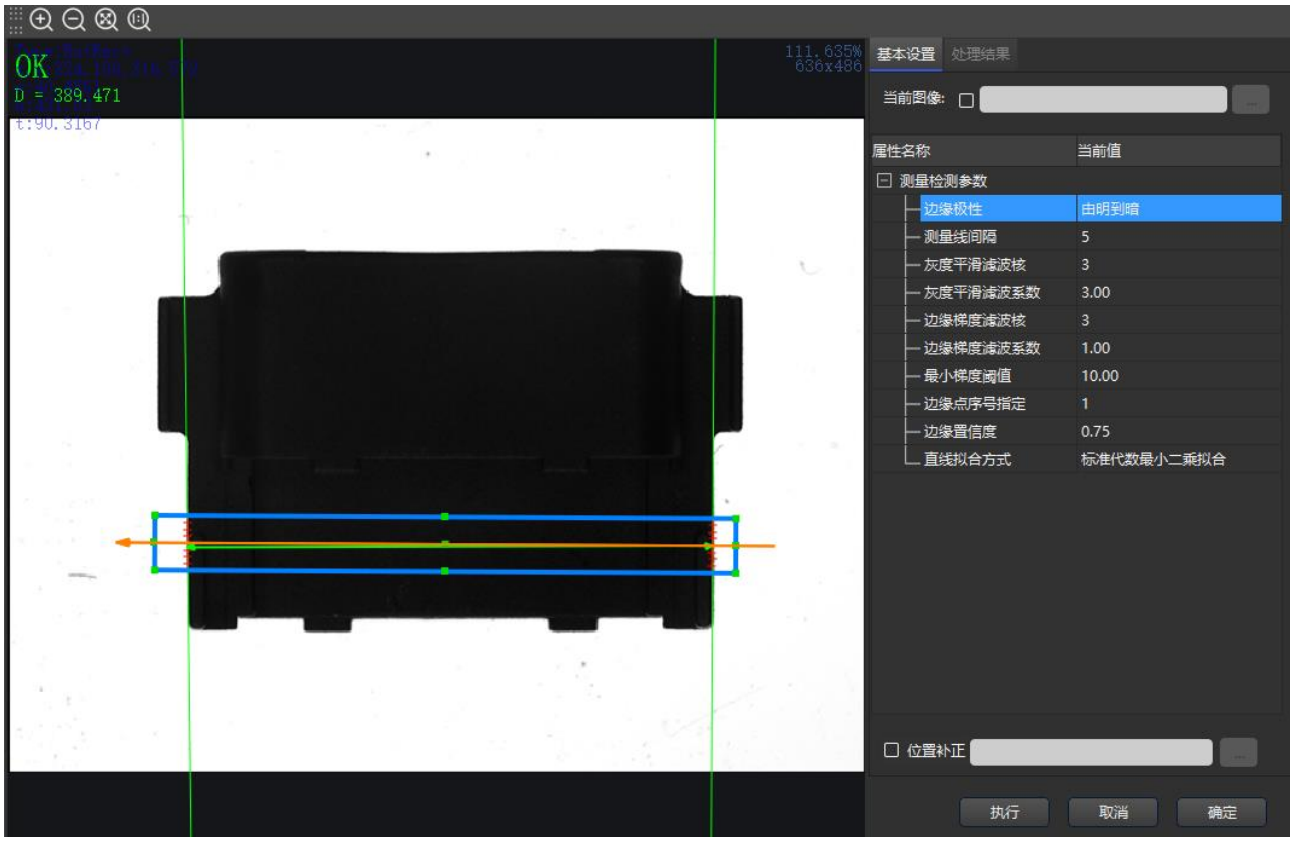
查看线线交点结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和两直线的角度、交点。



- 阈值设置：当两直线的有效点最小比例、最大最小距离阈值均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：两直线的交点颜色、两直线的检测框颜色、拟合的两条直线颜色、两直线边缘点颜色及线宽设置。
- 序号：每条直线的边缘点编号。
- 边缘点坐标：每条直线检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：每条直线检测到的边缘点的边缘梯度阈值。

6.5.3 间距测量

检测指定区域内的边缘对间距，可用于测量长度。



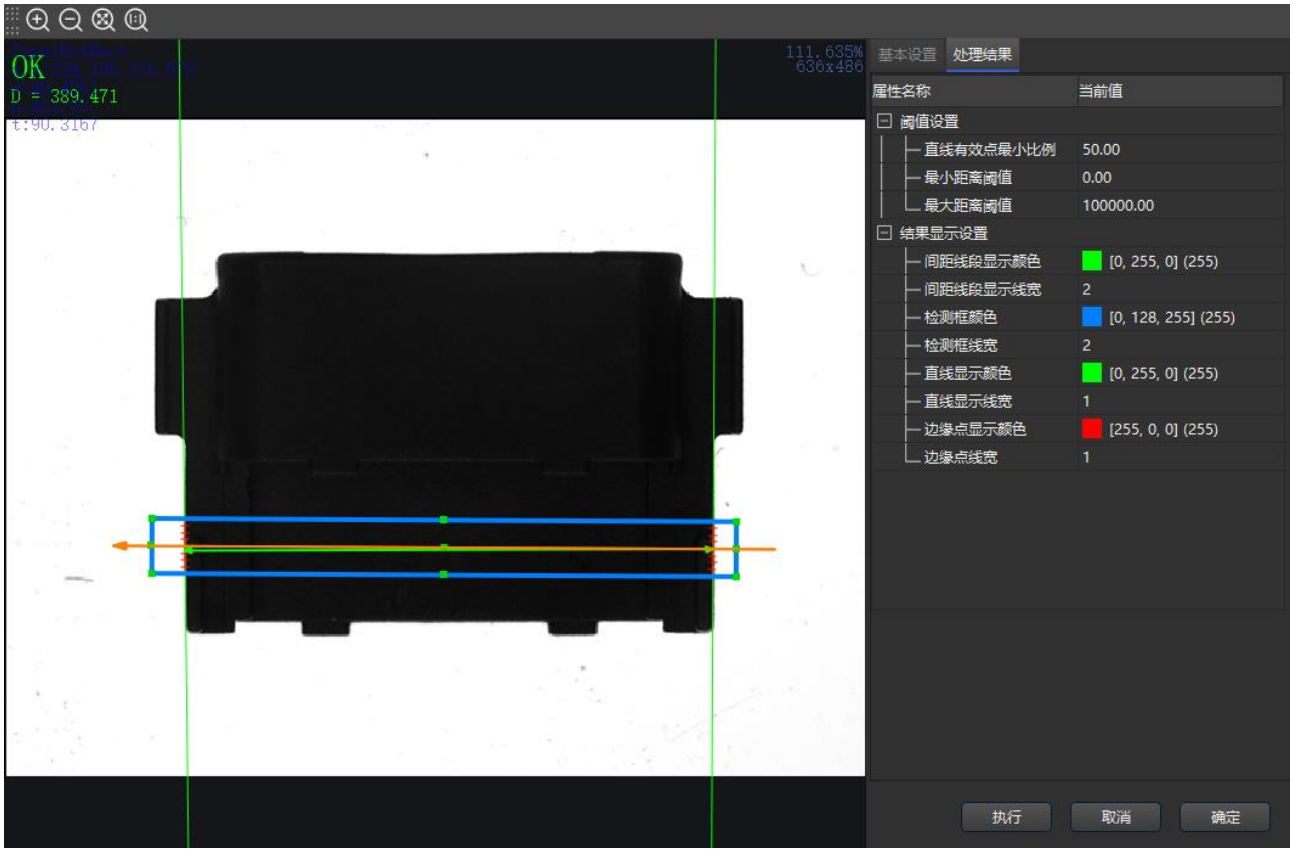
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置间距检测的测量参数。参数设置请参考直线查找章节。

- 处理结果页

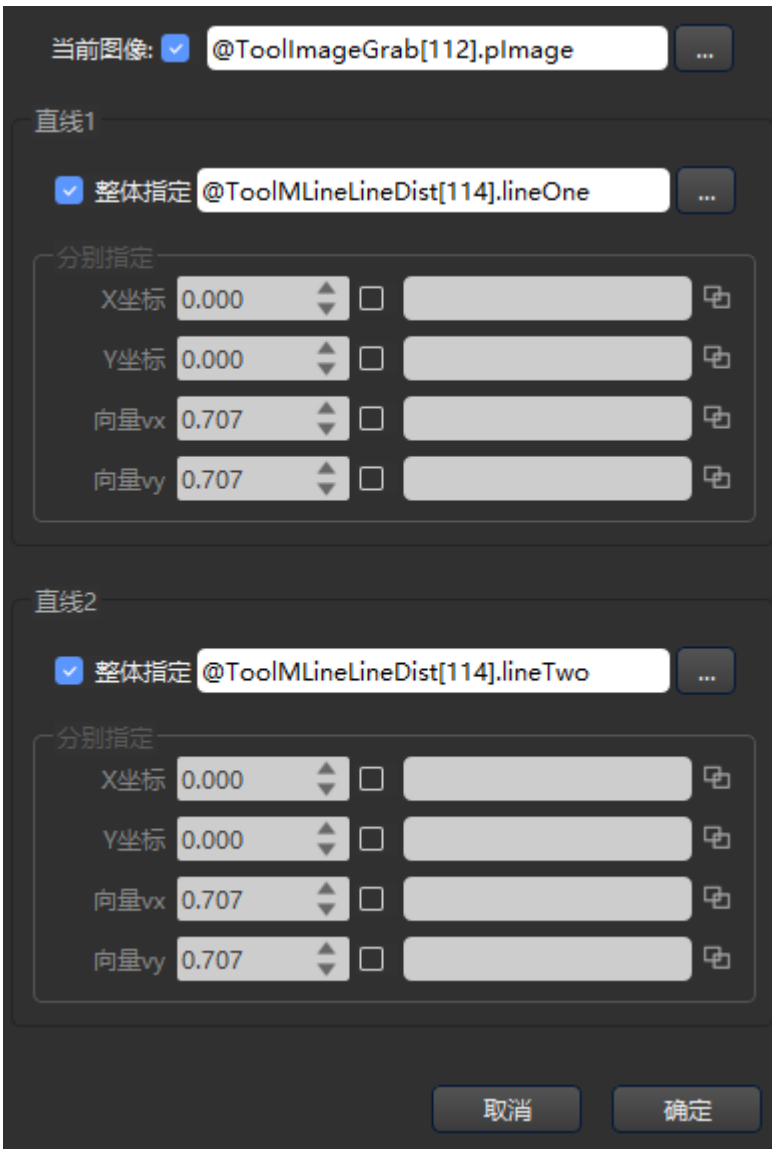
  查看间距值结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和间距值。



- 阈值设置：当有效边缘点最小比例、最大最小距离阈值均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：两直线的间距线段颜色、检测矩形框颜色、拟合的两条直线颜色、两直线边缘点颜色及线宽设置。

6.5.4 线线计算

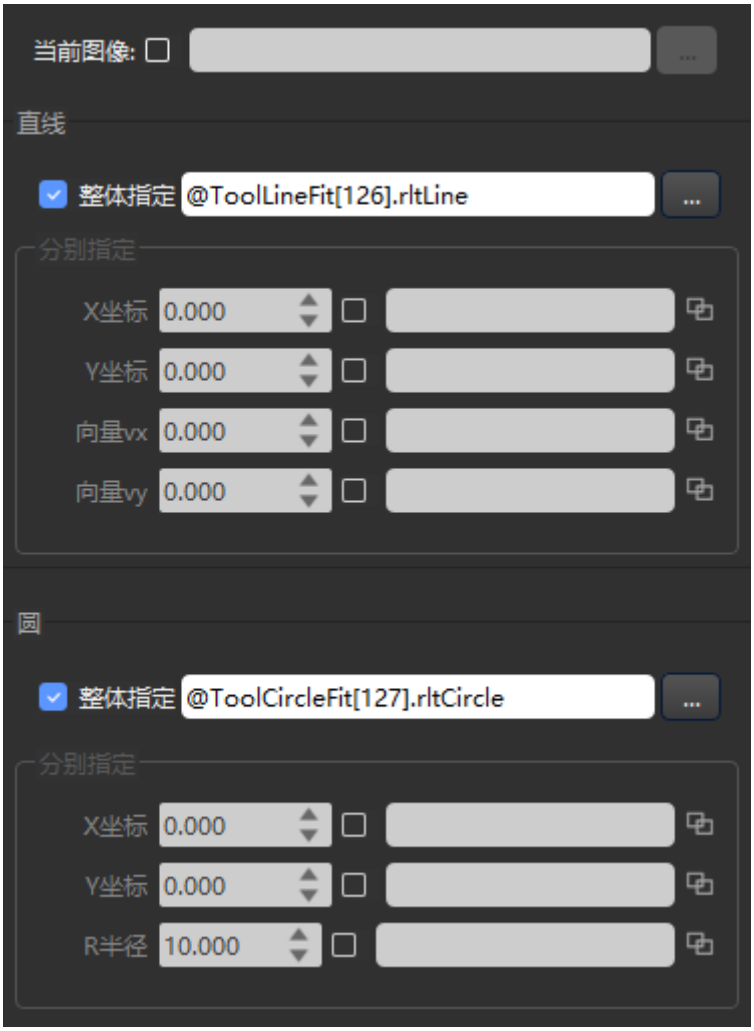
接收两条直线参数，根据接收参数求得直线与直线的交点和夹角并输出，可切换到工具结果界面查看输出信息。与“结果显示”工具搭配，可显示输出的交点及角度信息。



- 当前图像：算子运行时执行计算的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 整体指定：关联直线的来源。
- X 坐标：设置过直线的一个点的 X 坐标值，可关联。
- Y 坐标：设置过直线的一个点的 Y 坐标值，可关联。
- 向量 vx：设置直线单位向量 x 方向分量，可关联。
- 向量 vy：设置直线单位向量 y 方向分量，可关联。

6.5.5 线圆计算

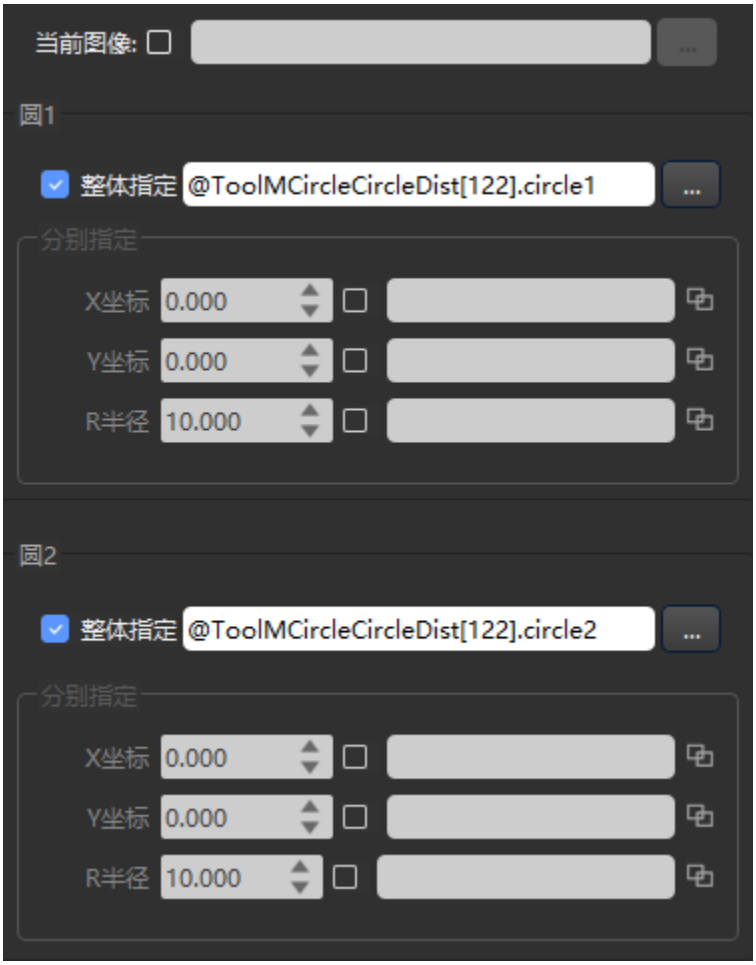
接收一条直线和一个圆参数，根据接收参数求得直线与圆的位置关系并输出，可切换到工具结果界面查看输出信息。与“结果显示”工具搭配，可显示输出的交点坐标、圆心到直线的距离、圆心到直线的垂线及垂足点坐标等信息。



- 当前图像：算子运行时执行计算的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 整体指定：关联直线或圆的来源。
- X 坐标：设置圆心或过直线的一个点的 X 坐标值，可关联。
- Y 坐标：设置圆心或过直线的一个点的 Y 坐标值，可关联。
- 向量 vx：设置直线单位向量 x 方向分量，可关联。
- 向量 vy：设置直线单位向量 y 方向分量，可关联。
- R 半径：设置圆的半径，可关联。

6.5.6 圆圆计算

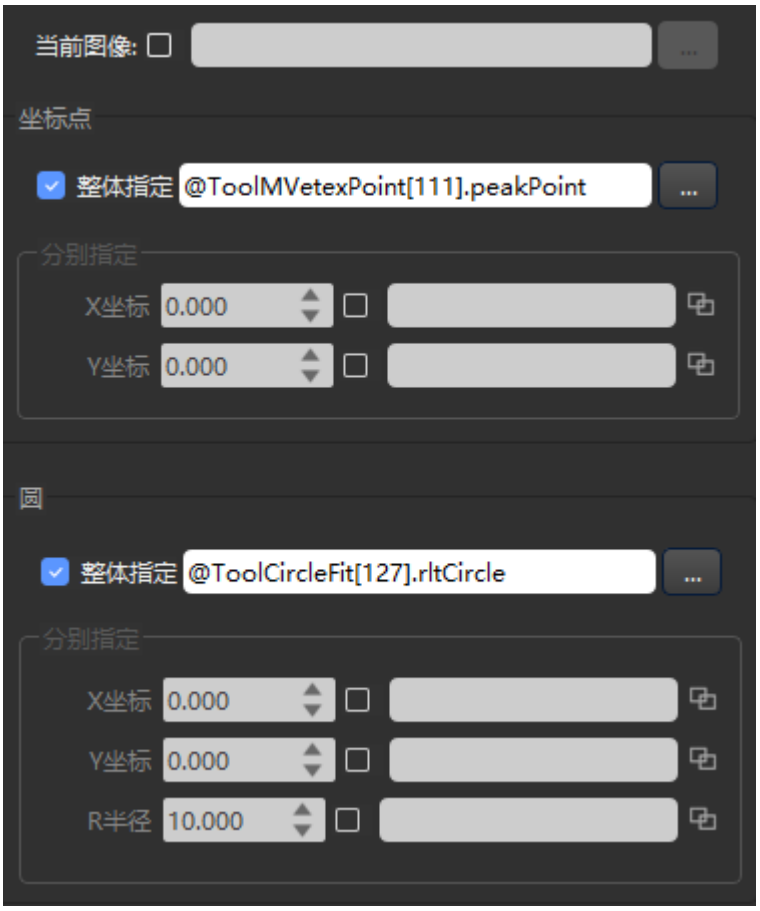
接收两个圆参数，根据接收参数求得圆与圆的位置关系并输出，可切换到工具结果界面查看输出信息。与“结果显示”工具搭配，可显示输出的两圆交点坐标、两圆圆心连线等信息。



- 当前图像：算子运行时执行计算的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 整体指定：关联圆的来源。
- X 坐标：设置圆心 X 坐标值，可关联。
- Y 坐标：设置圆心 Y 坐标值，可关联。
- R 半径：设置圆的半径，可关联。

6.5.7 点圆计算

接收一个点和一个圆参数，根据接收参数求得点到圆的距离并输出，可切换到工具结果界面查看输出信息（点到圆心的距离，点到圆的最近距离，点到圆的最远距离）。与“结果显示”工具搭配，可显示输出的距离线段。



- 当前图像：算子运行时执行计算的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 整体指定：关联点或圆的来源。
- X 坐标：设置点或圆心 X 坐标值，可关联。
- Y 坐标：设置点或圆心 Y 坐标值，可关联。
- R 半径：设置圆的半径，可关联。

6.5.8 点线计算

接收一个点和一条直线参数，根据接收参数求得点到直线的垂线并输出，可切换到工具结果界面查看输出信息（点到直线的垂足点、垂线、垂线角度、垂线距离）。与“结果显示”工具搭配，可显示输出的垂线信息。



- 当前图像：算子运行时执行计算的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 整体指定：关联点或直线的来源。
- X 坐标：设置点或过直线的一个点的 X 坐标值，可关联。
- Y 坐标：设置点或过直线的一个点的 Y 坐标值，可关联。
- 向量 vx：设置直线单位向量 x 方向分量，可关联。
- 向量 vy：设置直线单位向量 y 方向分量，可关联。

6.5.9 点点计算

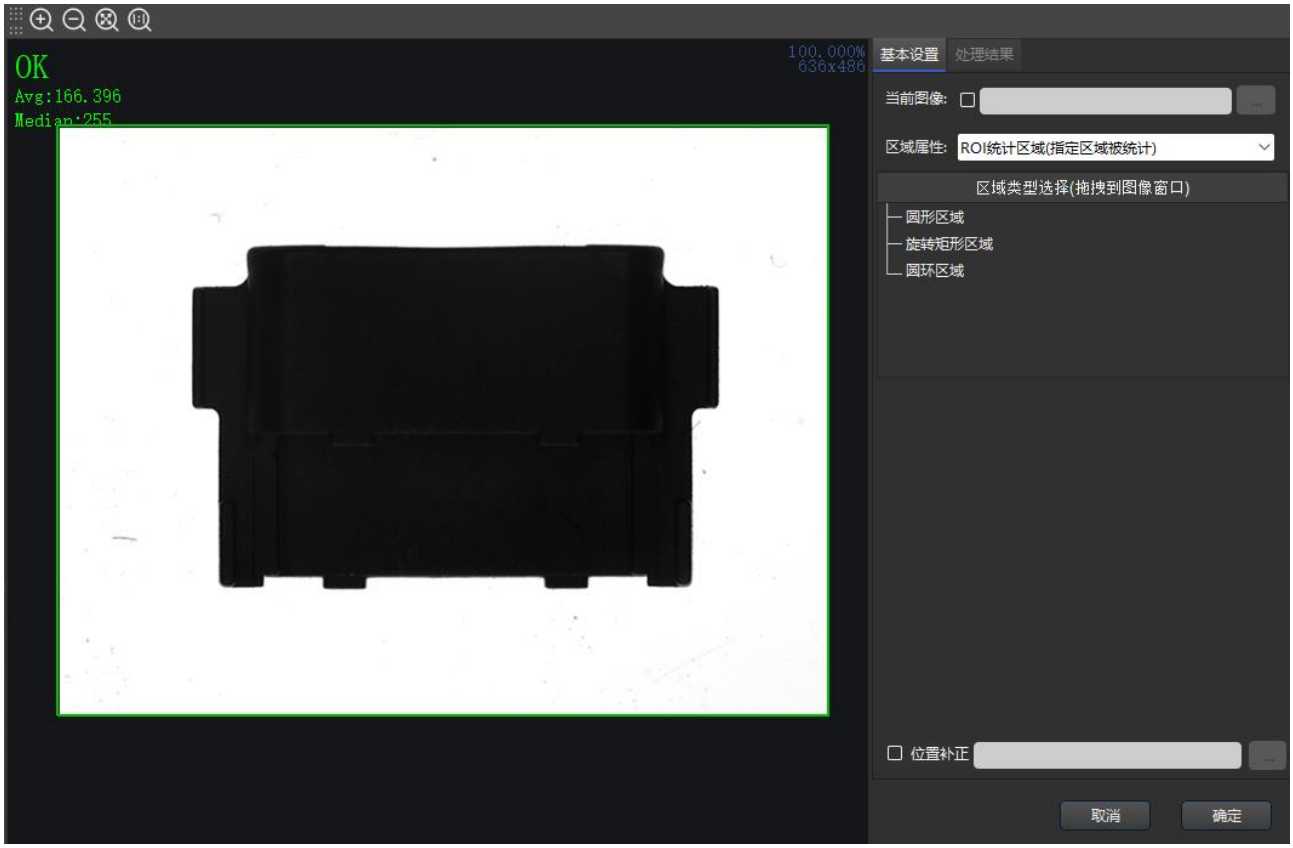
接收两个点参数，根据接收参数求得点到点的距离等信息并输出，可切换到工具结果界面查看输出信息（点到点的距离、两点生成的线段的角度、线段的中点坐标、两点生成的直线）。与“结果显示”工具搭配，可显示输出的线段信息。



- 当前图像：算子运行时执行计算的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 整体指定：关联点的来源。
- X 坐标：设置点的 X 坐标值，可关联。
- Y 坐标：设置点的 Y 坐标值，可关联。

6.5.10 直方图统计

检测图像上指定一个目标区域，统计目标区域中的像素个数、灰度平均值、中值、标准差。还会生成灰度直方图，可以清晰看到各个灰度值下的像素点分布状态。



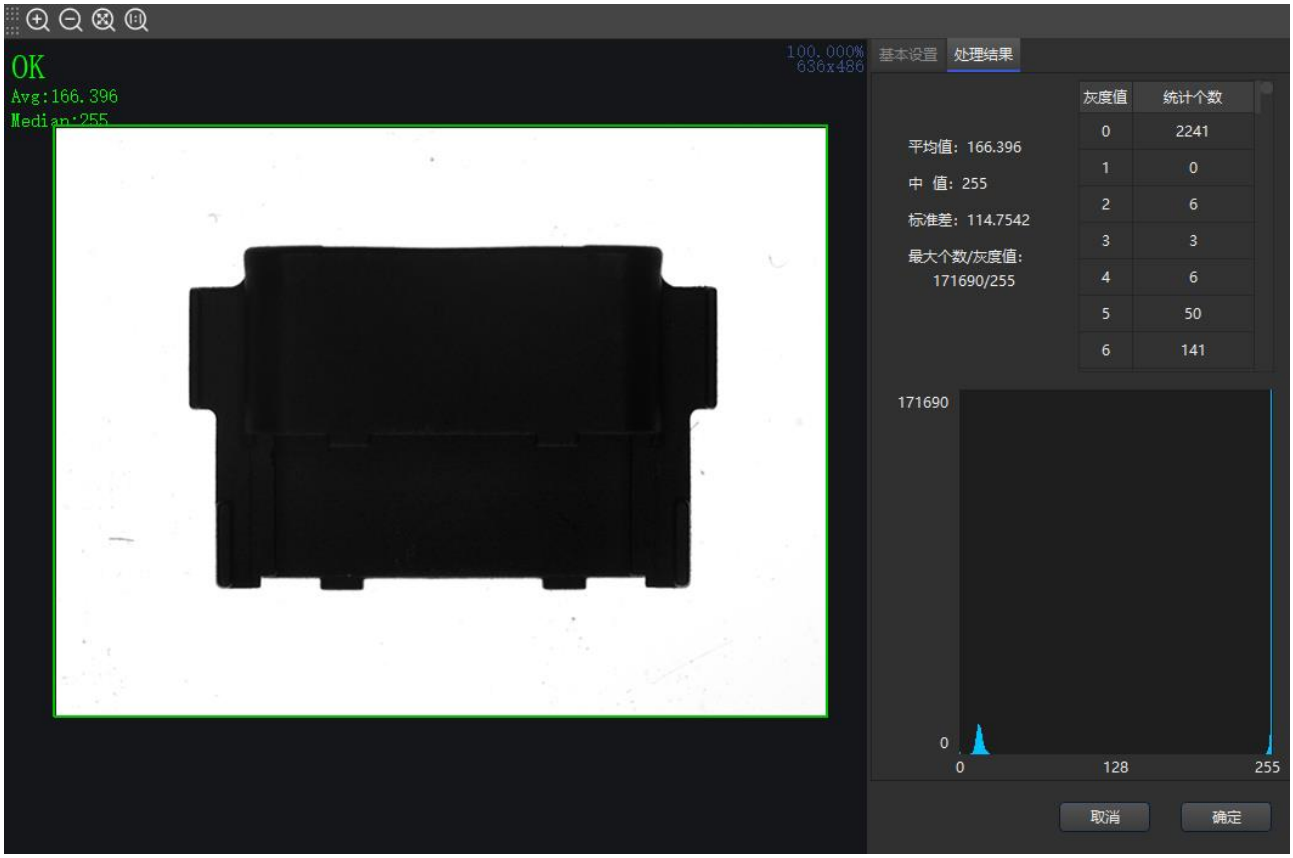
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

  主要用于设置统计直方图的 ROI 区域。参数设置请参考 BLOB 分析章节。

- 处理结果页

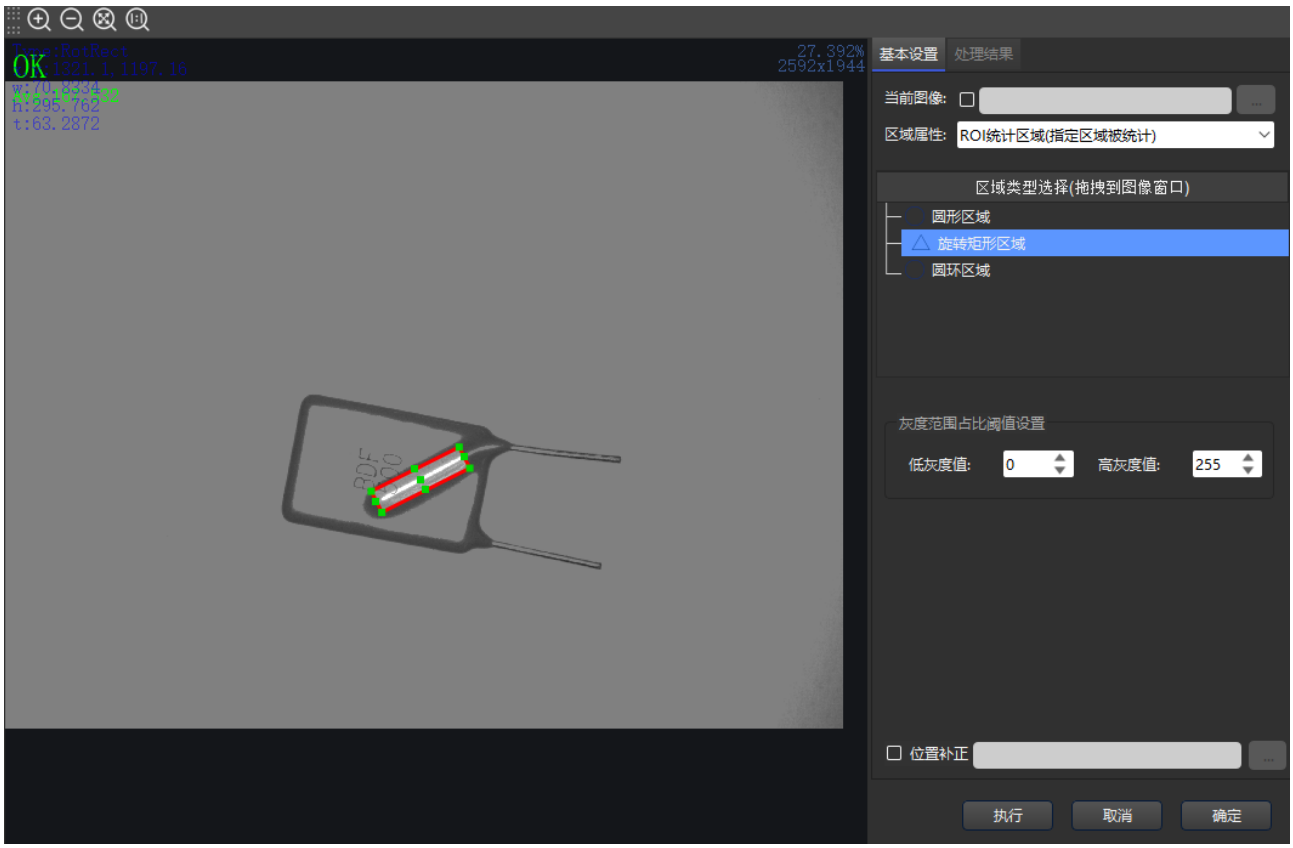
  查看统计结果。其中图形显示窗左上角输出该工具的状态结果和灰度中值、平均值。



- 数据：显示灰度平均值、中值、标准差数据、峰值灰度。
- 直方图：显示图像 ROI 区域内的灰度直方图。

6.5.11 像素统计

检测图像上指定一个目标区域，统计目标区域内满足高低阈值的像素个数和比率，一般用于有无判断。



该算子工具共 2 个设置页，分别为：基本设置和处理结果。

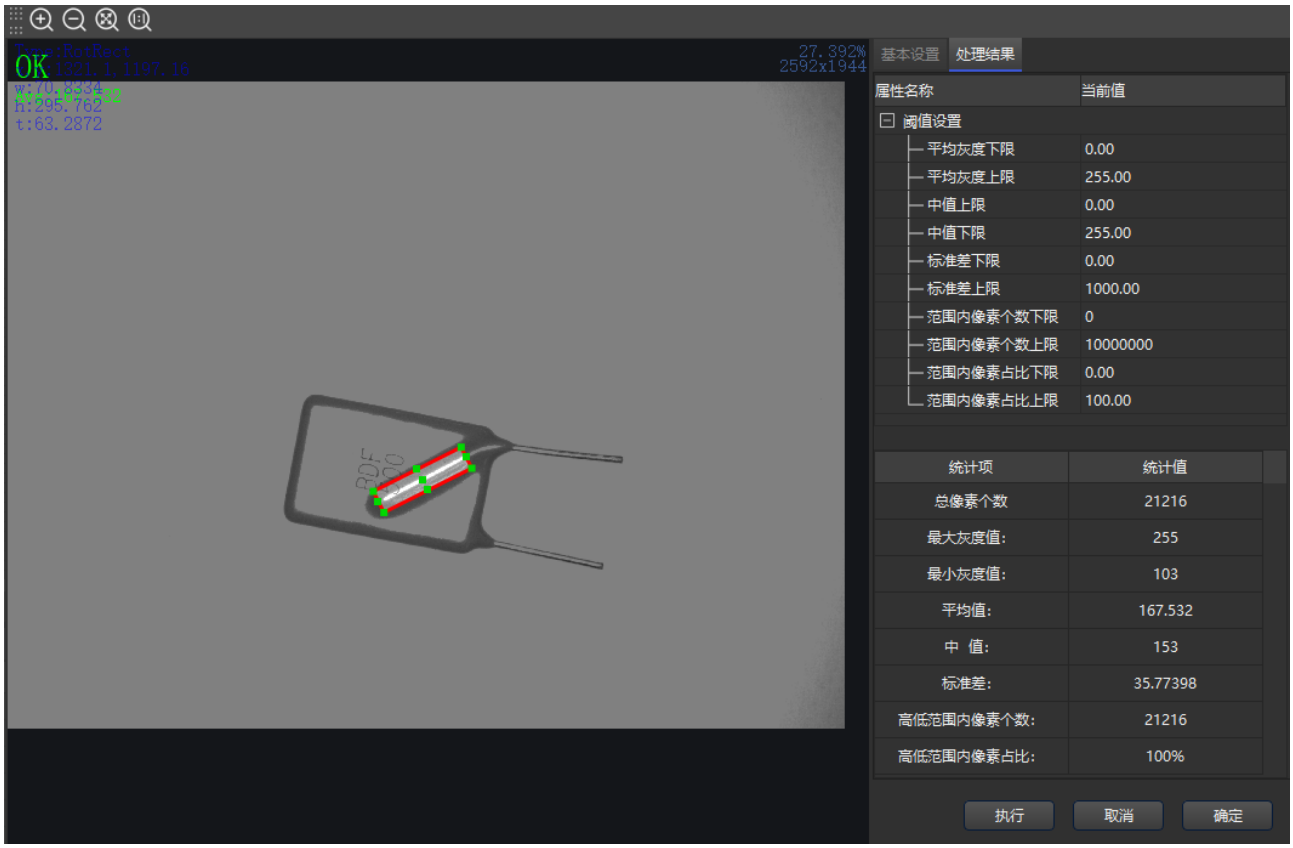
- 基本设置页

  主要用于设置像素统计的 ROI 区域。

  - 灰度范围占比阈值设置：设置目标区域内灰度阈值范围，用于统计该阈值范围内的像素个数和占比。
  - 其他参数设置请参考 BLOB 分析章节。

- 处理结果页

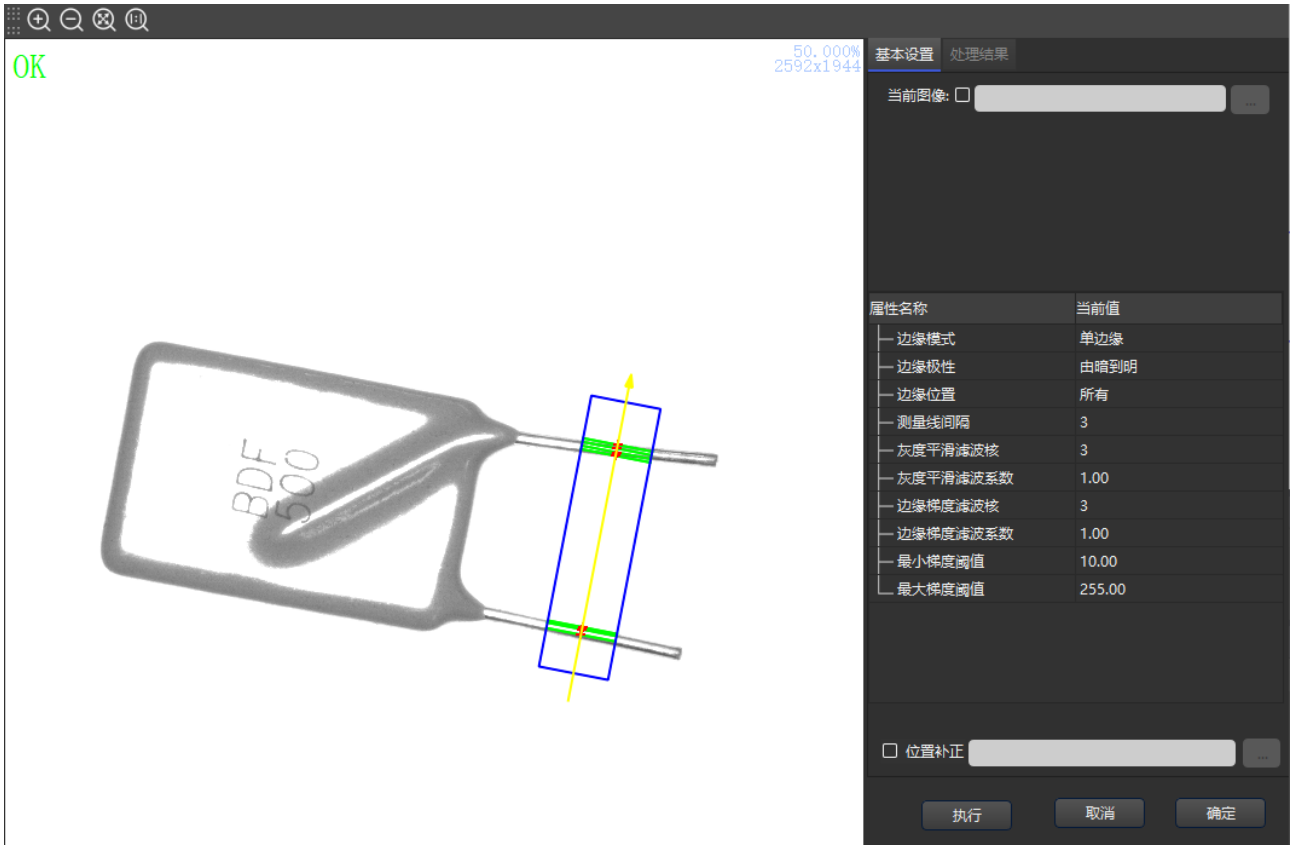
  查看像素统计结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和灰度平均值。



- 阈值设置：当灰度平均值、中值、标准差、灰度阈值范围内的像素个数及占比在设定的上下限范围内时，工具状态结果为 True。
- 总像素个数：目标区域内的像素总个数。
- 最大灰度值：目标区域内的灰度最大值。
- 最小灰度值：目标区域内的灰度最小值。
- 平均值：目标区域内的灰度平均值。
- 中值：目标区域内的灰度中值。
- 标准差：目标区域内的灰度标准差。标准差是方差的算术平方根，能反映一个数据集的离散程度。
- 高低范围内像素个数：设置的灰度阈值范围内的像素总个数。
- 高低范围内像素占比：设置的灰度阈值范围内的像素个数与总像素个数的比值。

6.5.12 卡尺工具

一种测量物体的宽度、边缘的位置、特征或边缘对的位置、距离的视觉工具。

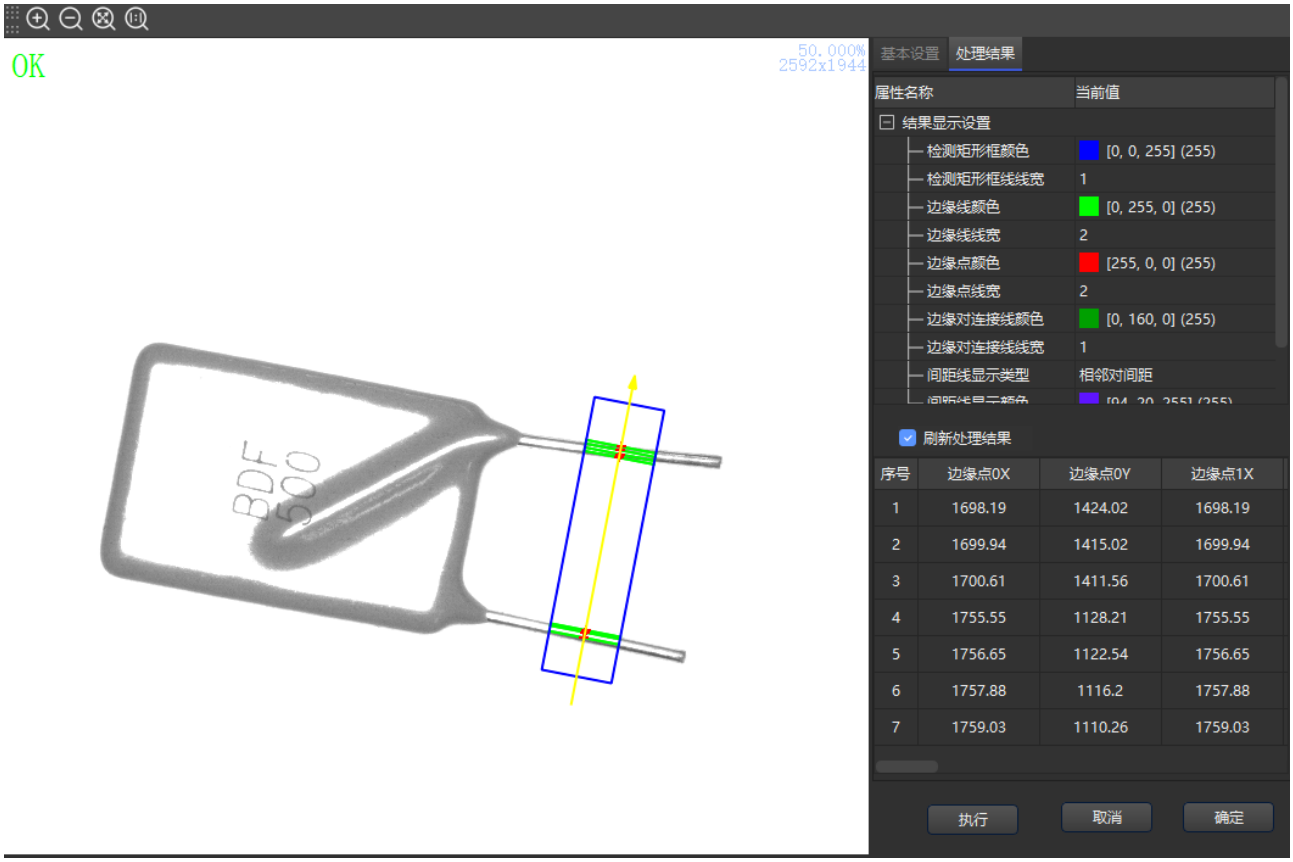


该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置卡尺的 ROI 区域及测量参数。以下仅说明该工具特有的参数，其他参数设置请参考直线查找章节。

- 边缘模式：①单边缘，检测指定区域内的边缘位置；②边缘对，检测指定区域内的边缘间距。
- 边缘位置：①所有，选择所有边缘点/对；②第一个，选择第一个边缘点/对；③最后一个，选择最后一个边缘点/对。

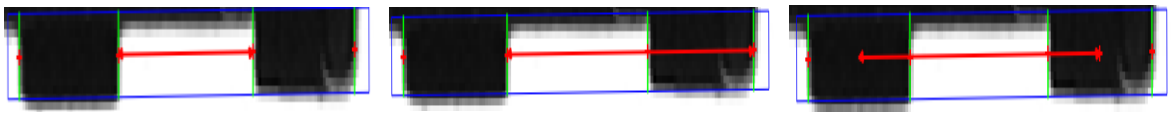


- 处理结果页

  查看边缘点集合及结果显示设置。其中图形显示窗左上角输出该工具的状态结果。

➢ 结果显示设置：检测矩形框、边缘点、边缘线、边缘对连接线及间距线的颜色、线宽的设置，当线宽设置为 0 时，不显示。

其中间距线类型可设置 3 种，相邻对间距、相邻对起始边间距和相邻对中心距，检测效果如下红线所示。



相邻对间距　　相邻对起始边间距　　相邻对中心距

➢ 序号：点集编号。

➢ 边缘点 X/Y：所有检测到的边缘点在当前图像的 XY 坐标位置。

➢ 边缘点极性：边缘的灰度变化，1 表示由明到暗，0 表示由暗到明。

➢ 测量宽度：检测到的两个边缘点的间距值。

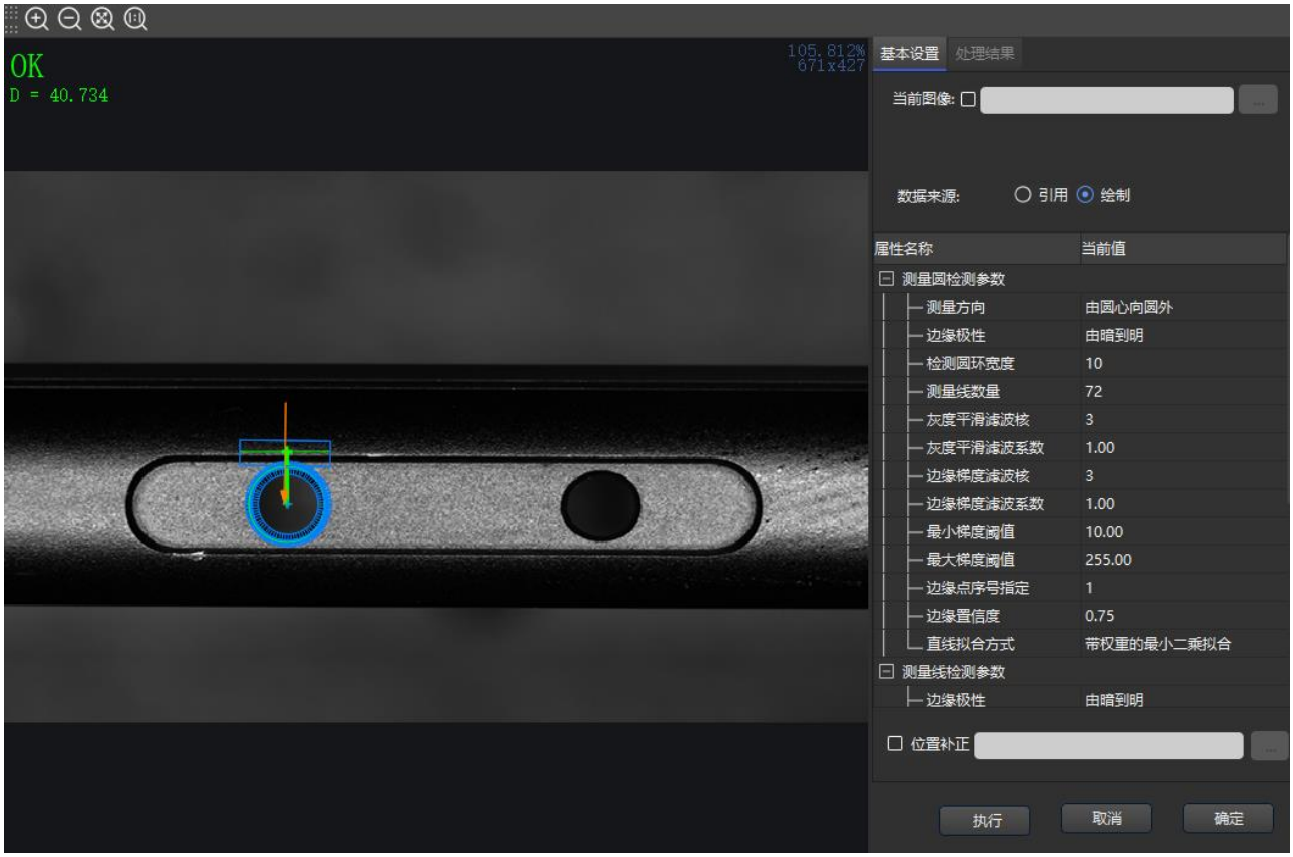
➢ 边缘点梯度：所有检测到的边缘点的边缘梯度阈值。

➢ 相邻对间距：相邻两边缘对的间距。

- 相邻对起始边间距：相邻两边缘对的起始边的间距。
- 相邻对中心距：相邻两边缘对的中点间距。

6.5.13 线圆测量

绘制 ROI 区域或接收一条直线和一个圆参数，通过过圆心点对直线作垂线的方式进行线圆测量，求得直线与圆的位置关系并输出被测物图像中直线和圆的垂直距离、垂足点坐标和相交点坐标。

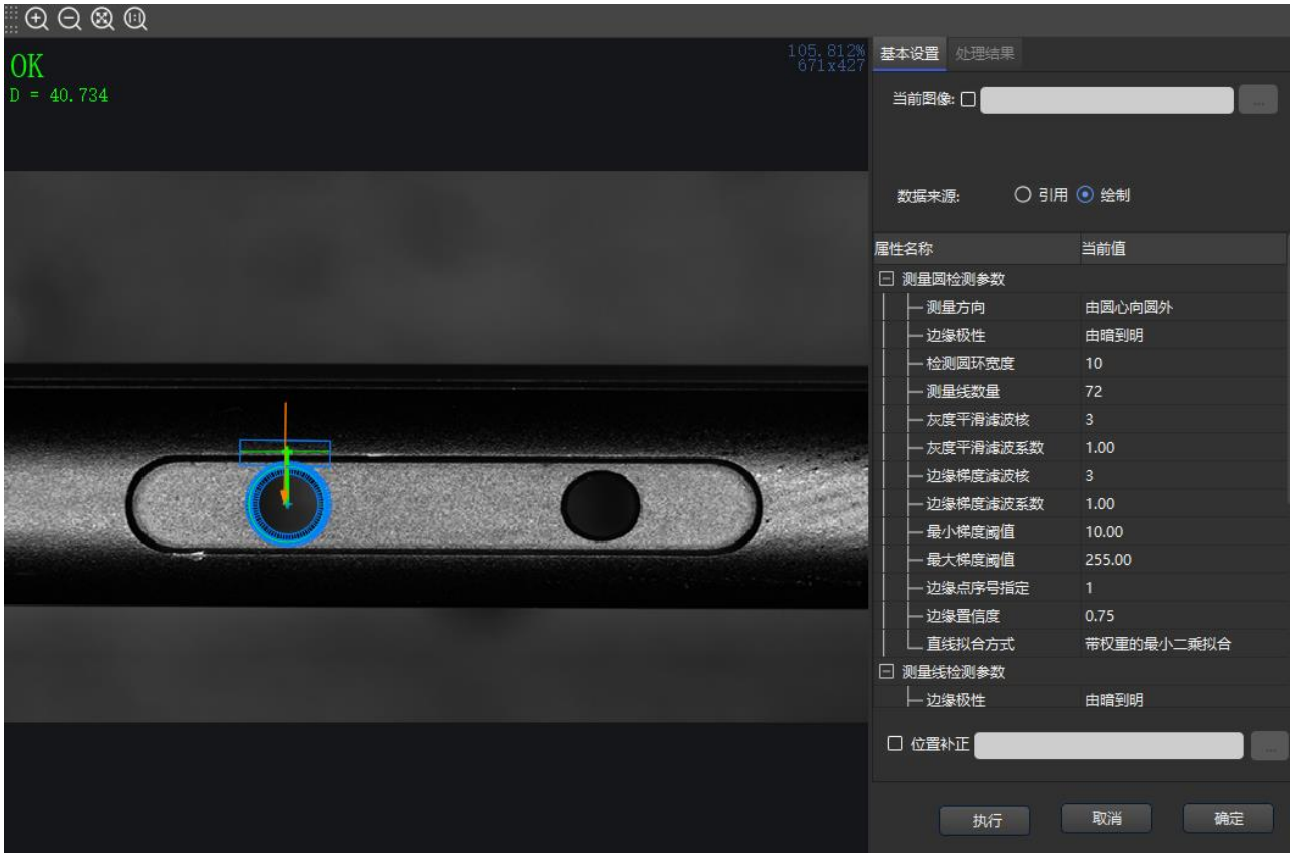


该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置直线、圆的 ROI 区域及测量参数。以下仅说明该工具特有的参数，其他参数设置请参考直线查找和圆查找章节。

- 数据来源：①引用，该方式同线圆计算工具，需要前述流程中有对应的直线数据（如直线查找）和圆数据（如圆查找），从而引用前述流程的直线输出和圆输出数据；②绘制，通过绘制的直线 ROI 和圆 ROI 区域进行直线查找和圆查找。



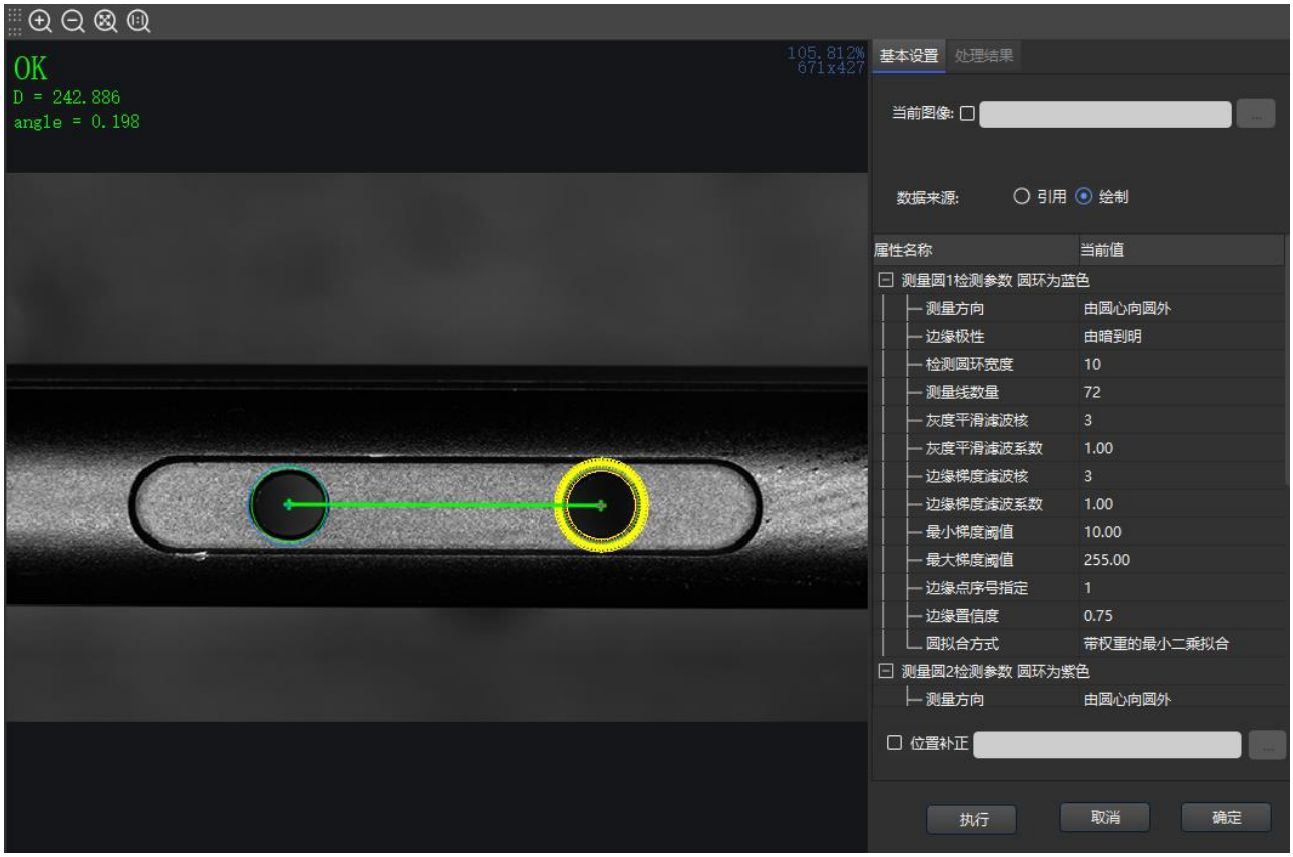
- 处理结果页

查看线圆距离、垂足点、交点及结果显示设置。其中图形显示窗左上角输出该工具的状态结果和线圆距离。

- 阈值设置：当圆心到直线的垂线距离均在最大最小距离阈值范围内时，工具状态结果为 True。
- 结果显示设置：圆心与直线的垂线、检测矩形框、直线、圆、边缘点的颜色、线宽的设置，当线宽设置为 0 时，不显示。
- 距离：圆心到直线的垂线距离。
- 垂足 X/Y：圆心到直线的垂线与直线的交点坐标。
- 交点 X/Y：如果直线和圆相交，输出直线和圆的交点坐标。

6.5.14 圆圆测量

绘制 ROI 区域或接收两个圆参数，求得圆与圆的位置关系并输出被测物图像中圆心与圆心连线长度、连线角度和两圆相交点坐标。

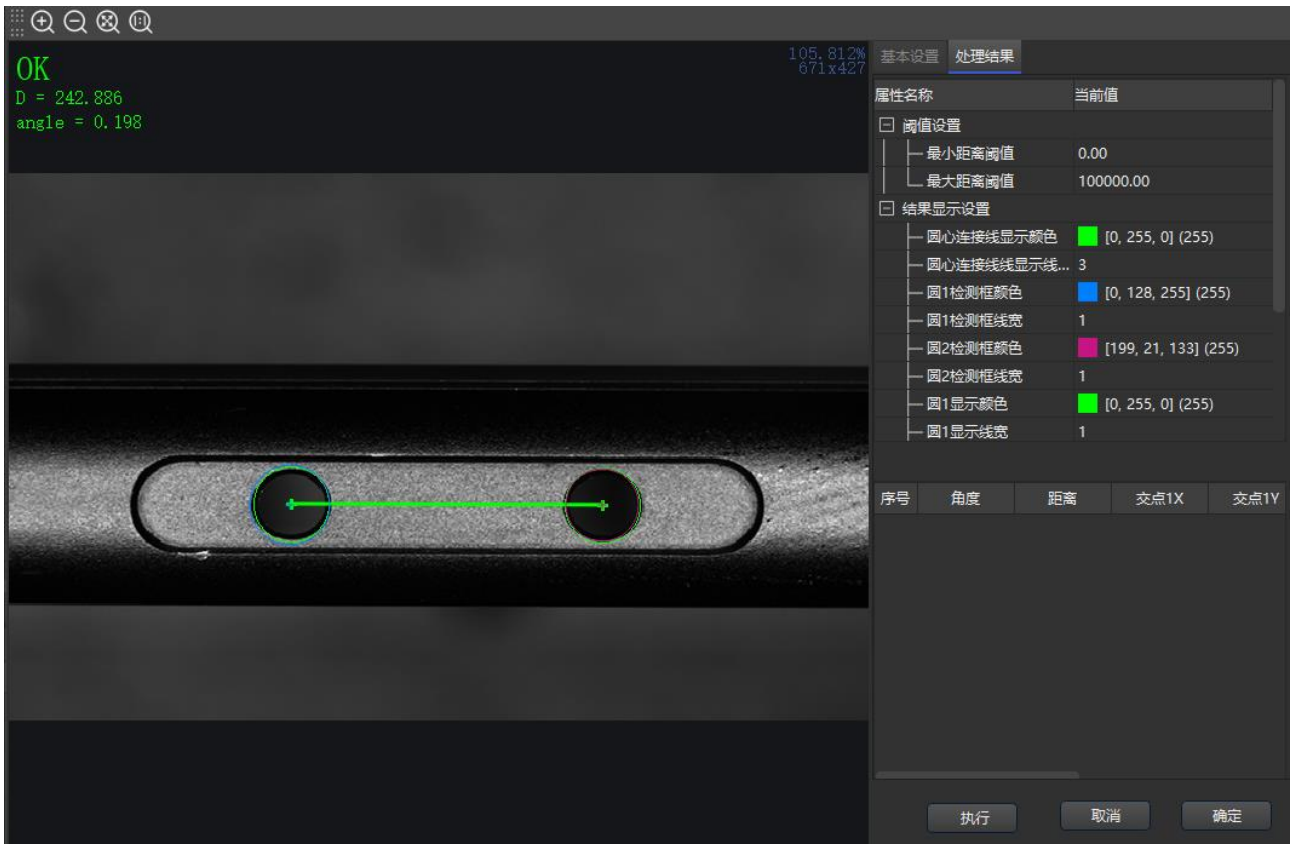


该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置两圆的 ROI 区域及测量参数。以下仅说明该工具特有的参数，其他参数设置请参考圆查找章节。

数据来源：①引用，该方式同圆圆计算工具，需要前述流程中有对应的圆数据（如圆查找），从而引用前述流程的圆输出数据；②绘制，通过绘制的两个圆形 ROI 区域进行圆查找。



- 处理结果页

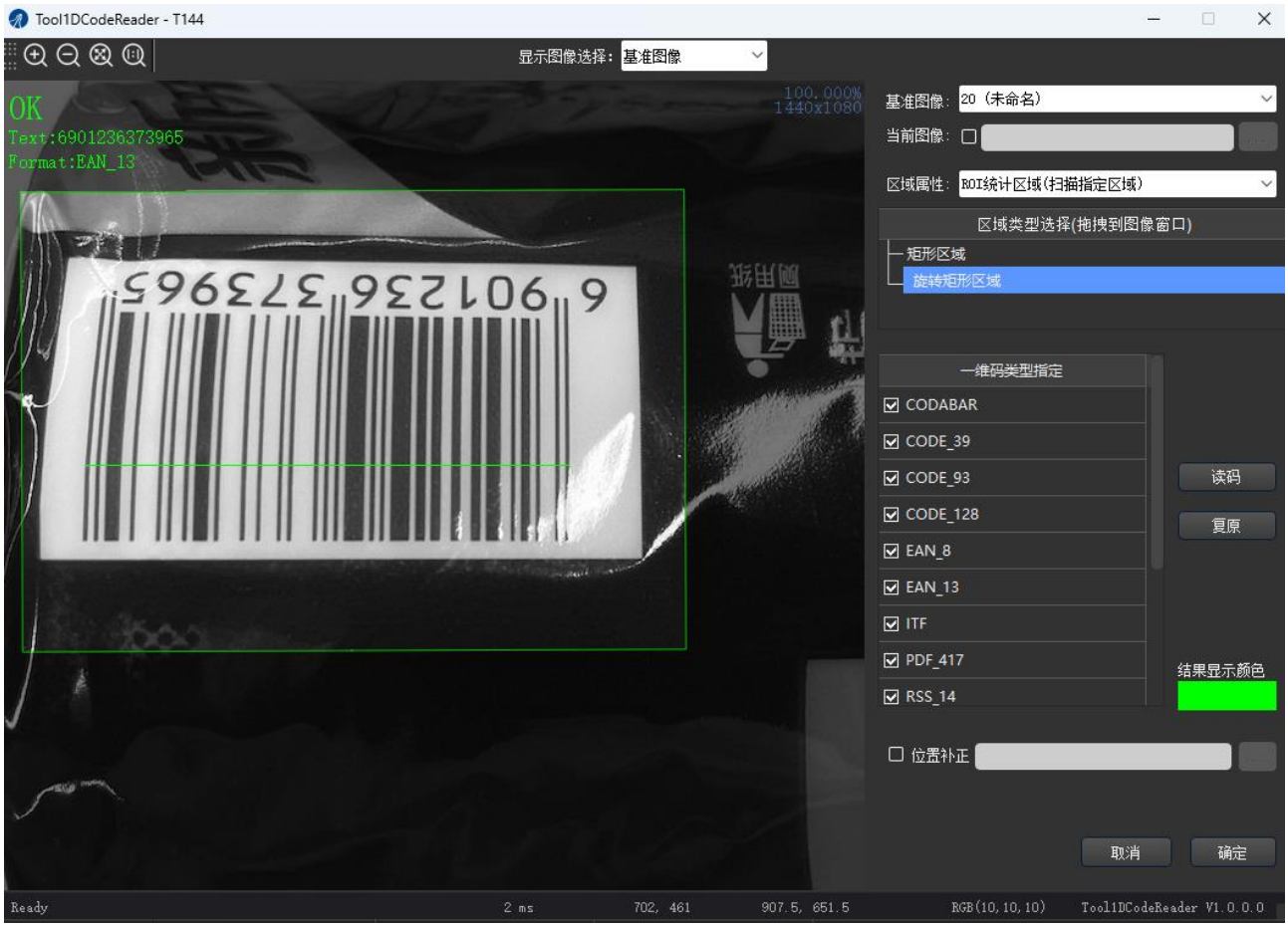
查看圆心与圆心连线距离、两圆交点及结果显示设置。其中图形显示窗左上角输出该工具的状态结果、两圆心连线距离及连线角度。

- 阈值设置：当圆心与圆心连线距离均在最大最小距离阈值范围内时，工具状态结果为 True。
- 结果显示设置：圆心与圆心连线、圆检测框、圆边缘点的颜色、线宽的设置，当线宽设置为 0 时，不显示。
- 角度：圆心到圆心连线角度。
- 距离：圆心到圆连线距离。
- 交点 X/Y：如果直线和圆相交，输出直线和圆的交点坐标。
- 位置关系：两圆的位置关系，包括相离（outside）、相交（intersect）和包含（inside）三种。

6.6 识别

6.6.1 一维码

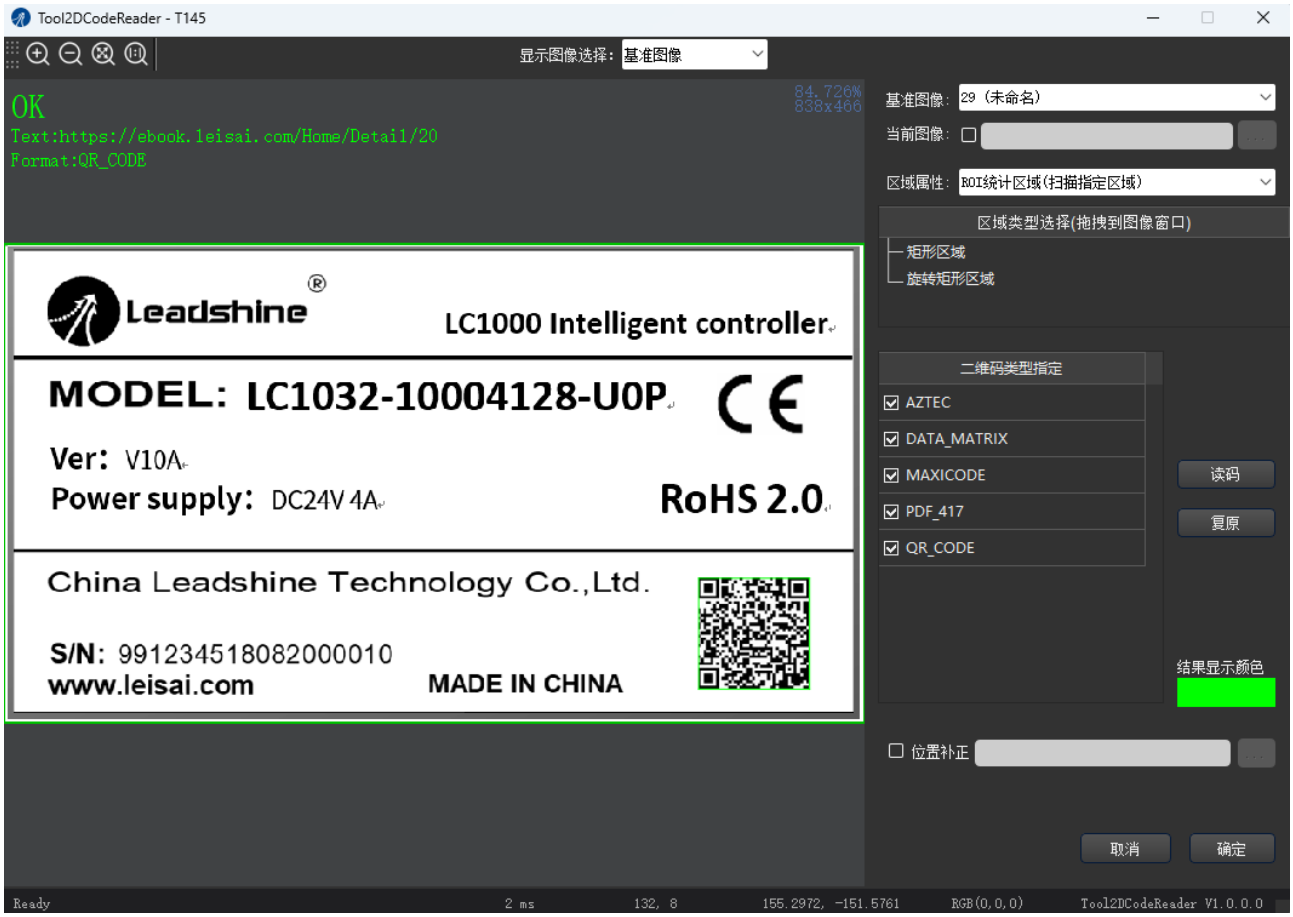
用于识别指定区域内的一维码并以字符的形式输出，容忍目标一维码具有一定量的角度旋转及倾斜，目前一次只能识别一个码，支持解码类型包括 CODABAR、CODE39、CODE93、CODE128、EAN-8、EAN-13、ITF、PDF_417、RSS_14、RSS_EXPANDED、UPC_A、UPC_E、UPC_EAN_EXTENSION。



- 一维码类型指定：勾选需要识别的码类型。
- 读码：执行一次读码操作。
- 复原：恢复到读码前的状态。
- 结果显示颜色：设置读码结果显示的文本颜色。

6.6.2 二维码

用于识别指定区域内的二维码并以字符的形式输出，容忍目标一维码具有一定量的角度旋转及倾斜，目前一次只能识别一个码，支持解码类型包括 QR_CODE、DATA_MATRIX、MAXICODE、AZTEC、PDF_417。

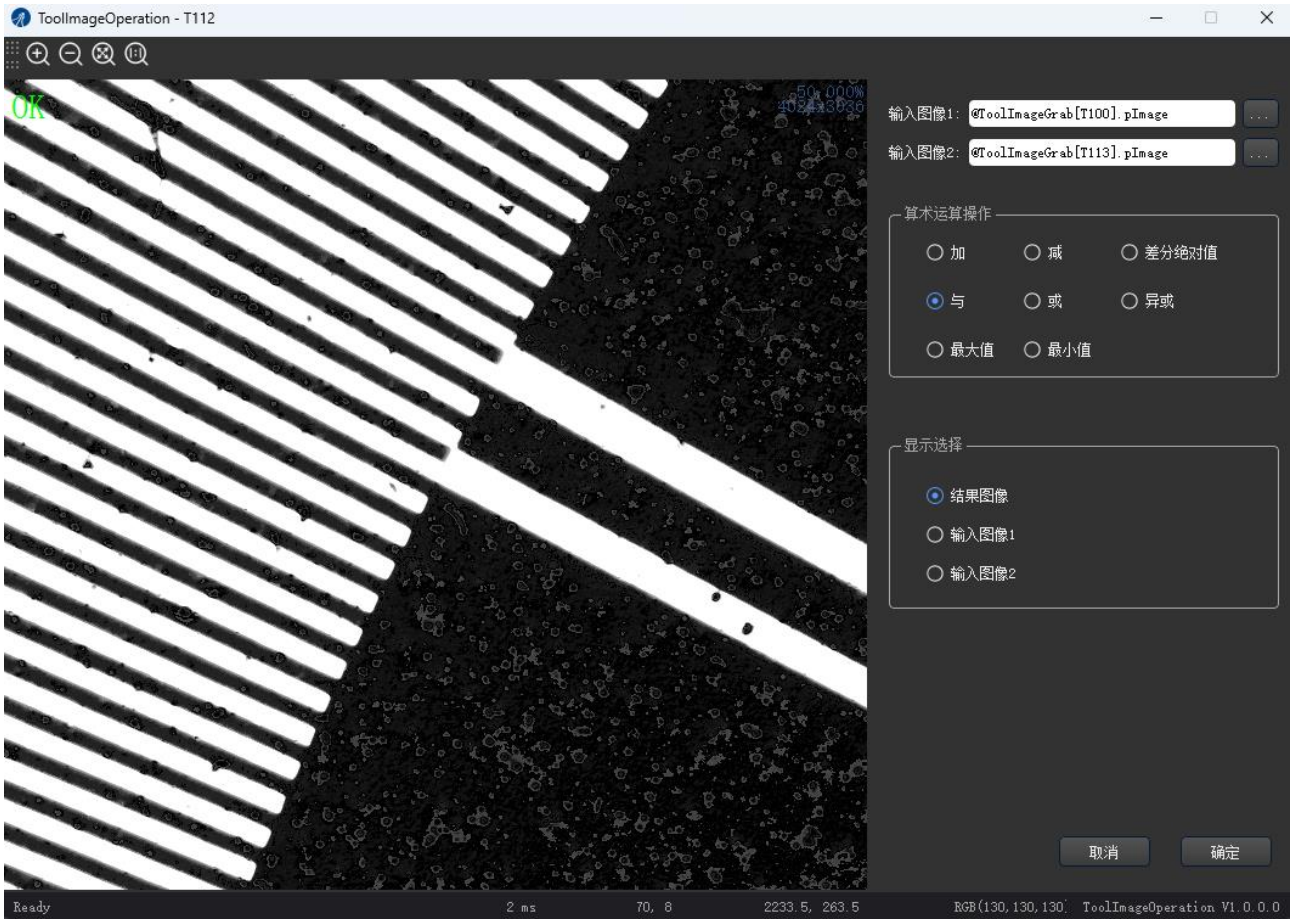


参数设置请参考一维码。

6.7 图像处理

6.7.1 图像运算

对两张图像做加、减、与、或等操作，并输出一张新的图像。



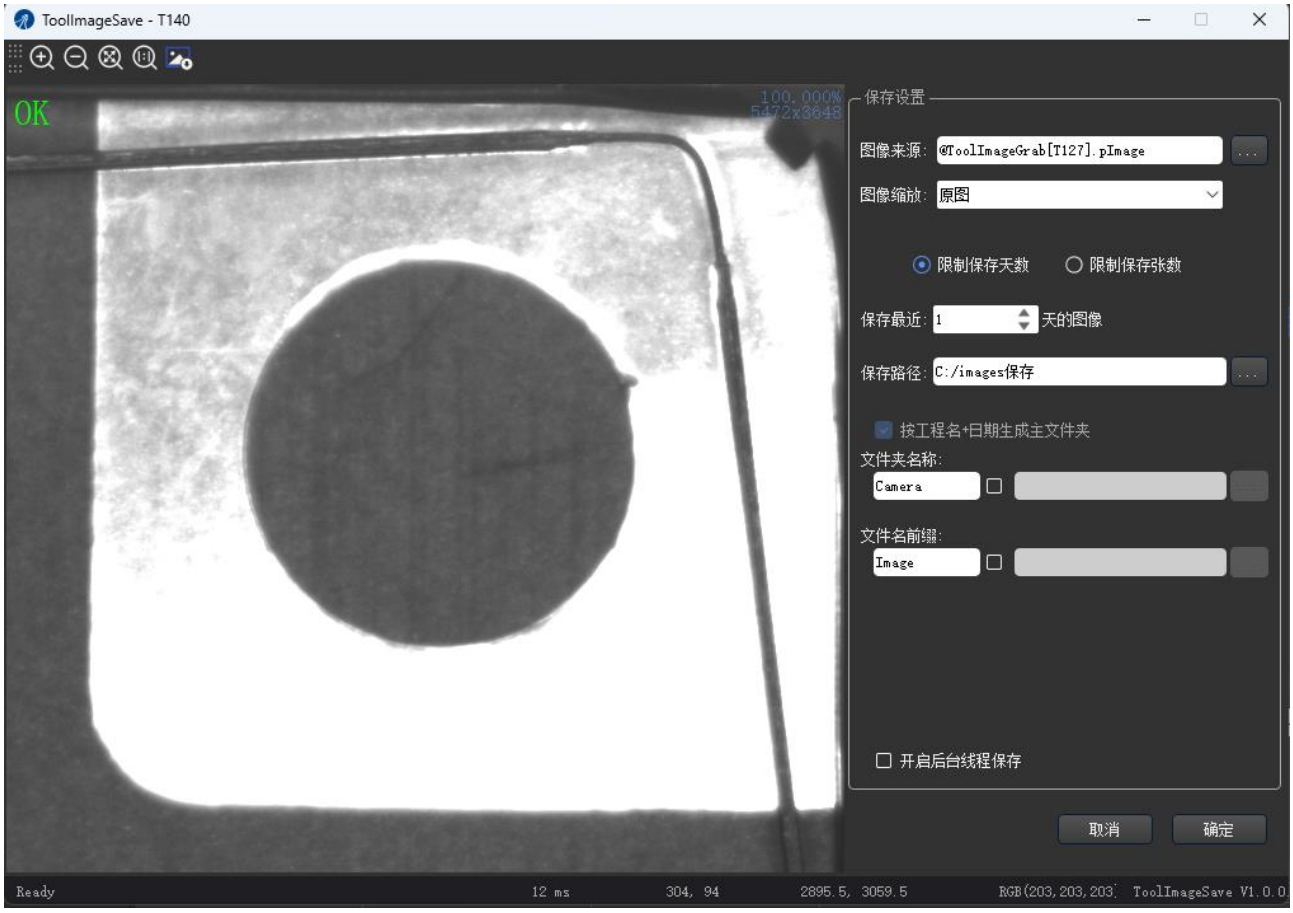
- 输入图像 1：关联第一张图像来源。
- 输入图像 2：关联第二张图像来源。
- 算术运算操作：可以选择加、减、差分绝对值、与、或、异或、最大值和最小值。

项目	概述
加	两张图像做加法运算，如果得出的值大于 255，则会被置为 255
减	两张图像做减法运算，如果得出的值小于 0，则会被置为 0
差分绝对值	两张图像做减法运算，如果得出的值小于 0，则会求绝对
与	两张图像做与运算，按二进制的每一位做与操作
或	两张图像做或运算，按二进制的每一位做或操作
异或	两张图像做异或运算，按二进制的每一位做异或操作
最大值	求两张图像的最大值，将每一个像素进行比较，输出值大的像素
最小值	求两张图像的最小值，将每一个像素进行比较，输出值小的像素

- 显示选择：设置当前需要显示的图像。

6.7.2 图像保存

流程运行一次保存一张图像到本地硬盘。

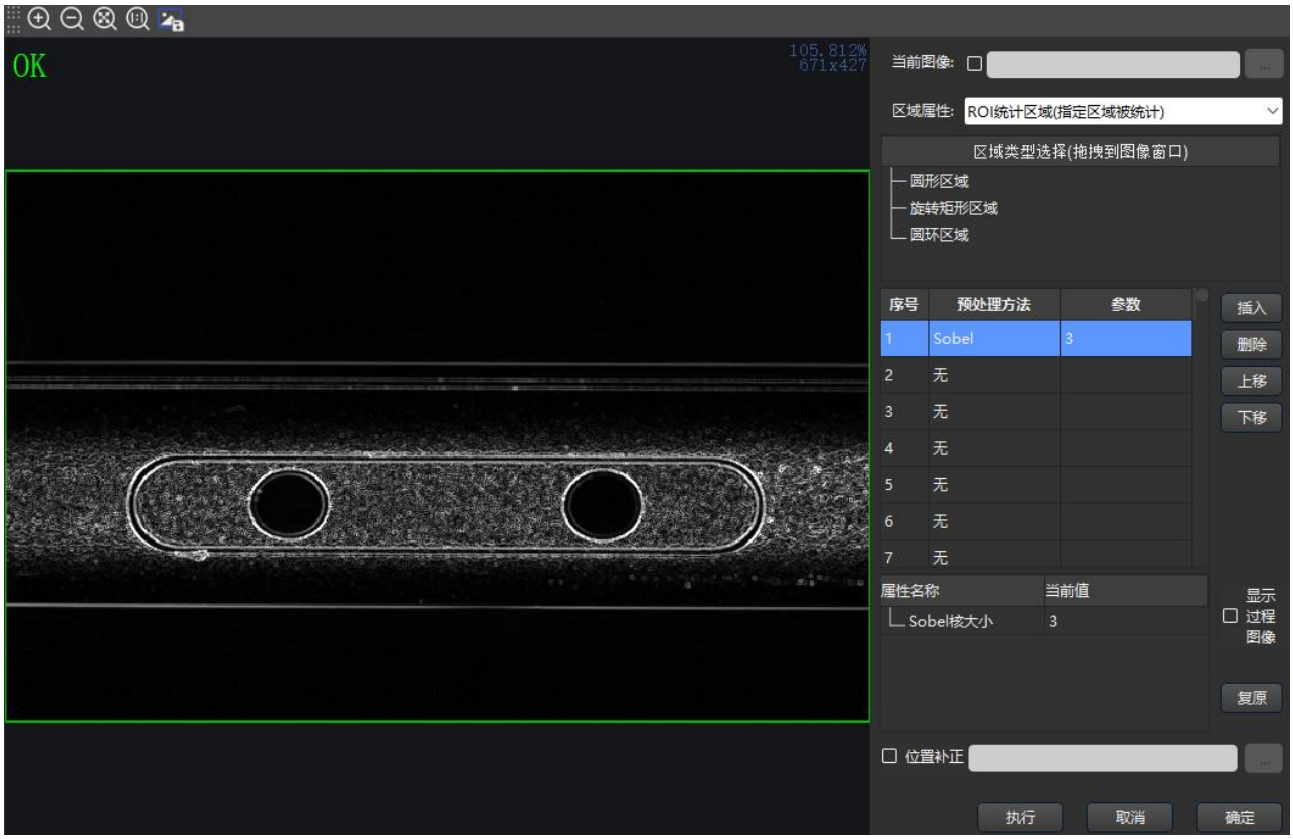


- 图像来源：关联当前需要保存的图像来源。
- 图像缩放：可将图像缩小后再保存。
- 限制保存天数：按运行天数进行保存，最大可保存最近 180 天的图像。
- 限制保存张数：按图像数量进行保存，最大可保存 5000 张图像。
- 保存路径：设置保存图像的目录。
- 按工程名+日期生成主文件夹：选择“限制保存天数”时勾选显示，根据当前工程名称和日期生成主文件夹名称（如 myProject_2024_12_02）。
- 按工程名生成主文件夹：选择“限制保存天数”时勾选显示，根据当前工程名称生成主文件夹名称（如 myProject）。
- 文件夹名称：设置主文件夹下的子文件夹名称，可关联。
- 文件名前缀：设置保存图像的名称，可关联，会自动在名称后增加日期和时间后缀（如 Image_12_02_09_10_32_203.bmp）。

- 开启后台线程保存：设置单独开启后台线程保存图像。

6.7.3 图像预处理

对输入图像做预处理并输出处理后的图像。



- 插入：在两个预处理方法中间插入一个预处理方法。
- 删除：删除选择的预处理方法。
- 上移/下移：将选中的方法上下移动。
- 显示过程图像：勾选后点击“执行”，可以查看当前选中的预处理方法执行后的结果。
- 执行：运行所有设置的预处理方法，输出最后的图像。
- 复原：恢复到源图像状态。
- 其他参数设置请参考 BLOB 分析章节。

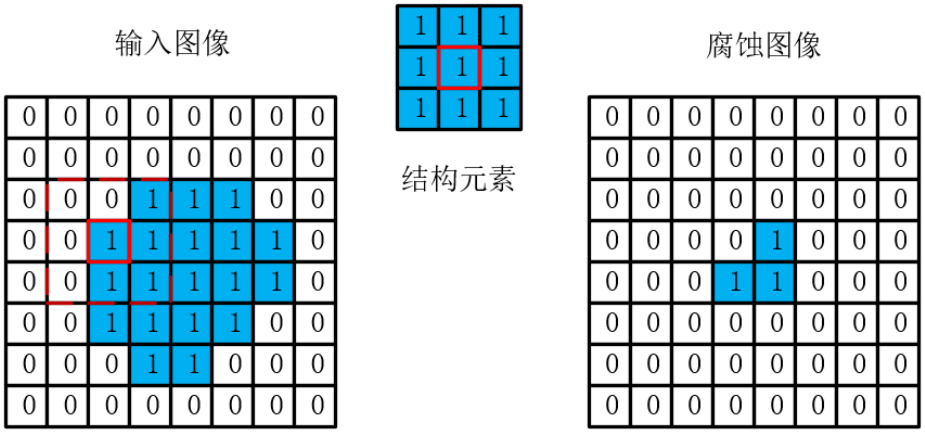
预处理方法包括以下几种：

- 二值化

二值化是将图像上的像素点的灰度值设置为 0 或 255，高于某个阈值全是 255，低于某个阈值全是 0，这样整个图像将呈现出明显的黑白效果。

- 形态学

形态学处理可以根据设定的结构元素和形态学方法对输入灰度图像进行形态学操作。形态学方法包括：膨胀、腐蚀、开运算、闭运算、梯度、顶帽、黑帽；支持的结构元素包括：矩形、十字、椭圆，它不是一个像素，而是一个几何形状的像素块。这个结构元素从左至右、从上到下遍历整个矩阵，每当在目标图像中找到一个与结构元素完全相同的子图像时，就把该子图像中与结构元素的中心位置对应的那个像素位置标记为 1，不一致则按 0 处理，这样图像中的高亮区域就会减少，这就是图像的腐蚀，如下图所示的示意图。



而膨胀与腐蚀正好相反，每当在目标图像中找到与结构元素有部分相同的子图像时，就把该子图像中与结构元素的中心位置对应的那个像素位置标记为 1，否则按 0 处理，这样图像中的高亮区域就会增加。

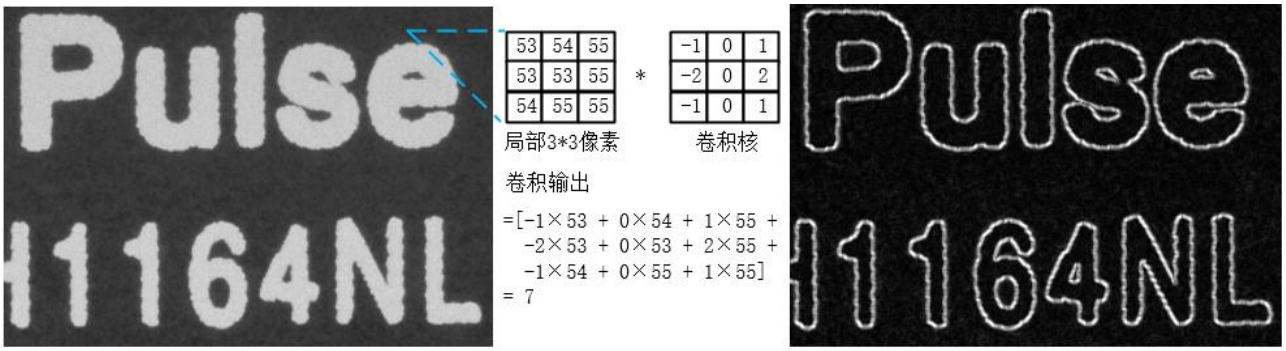
腐蚀和膨胀还可以组合起来对图像进行操作，比如开运算是对输入图像执行先腐蚀后膨胀操作，而闭运算是对图像执行先膨胀后腐蚀操作。腐蚀可对图像进行局部缩小，膨胀可对图像进行局部放大，因此开运算可有效清除二值图像中小的白色噪声像素块凸显高亮区域，闭运算则能填充二值图像中小的黑色像素凸显灰暗区域。

- 反色

对目标图像灰度取反。

- 滤波

图像滤波主要是为了图像平滑和去噪，是图像预处理很重要的一环。具体处理方法是使用一个小型滤波器（如 3x3），输出中心点的像素值由周围 8 个决定。



如果输出像素依赖的是输入像素的线性组合，则称为线性滤波器；如果输入输出是非线性的，则称为非线性滤波器。通过滤波器的方法对图像进行的操作称之为卷积，而滤波器则称为卷积核。

图像滤波可细分为均值滤波(卷积核所有像素对中心像素的贡献相等)，高斯滤波(邻域像素根据到中心像素距离不同有着不同的权重)，中值滤波(基于排序统计理论的一种能有效抑制噪声的非线性滤波)和边缘保持滤波(一种非线性、边缘保留、有效去噪声的滤波方法)。它们的区别如下：

- 均值滤波可对图像进行快速、简单的平滑处理，对于噪声的处理很差。
- 高斯滤波是一种比均值滤波稍微好一点的线性滤波，也不会改变原图像的边缘走向。
- 中值滤波处理速度快，对椒盐噪声处理效果很好，但对高斯噪声处理不好。
- 边缘保持滤波计算复杂，效率低，对高斯噪声处理效果很好，但对椒盐噪声处理不佳。

高斯滤波、中值滤波、均值滤波、边缘保持滤波的设置参数如下：

- x 方向滤波核：中值/高斯/均值滤波核的宽度，数值越大，图像越模糊。
- y 方向滤波核：中值/高斯/均值滤波核的高度，数值越大，图像越模糊。
- sigma：高斯滤波核的标准差，数值越大，图像越模糊。
- 滤波直径：边缘保持滤波核直径，数值越大，图像越模糊。
- 颜色滤波 sigma：边缘保持滤波颜色核的标准差，数值越大，越大范围内的颜色被聚合。
- 空间滤波 sigma：边缘保持滤波空间核的标准差，数值越大，越大距离范围内的像素参与聚合。

- Sobel

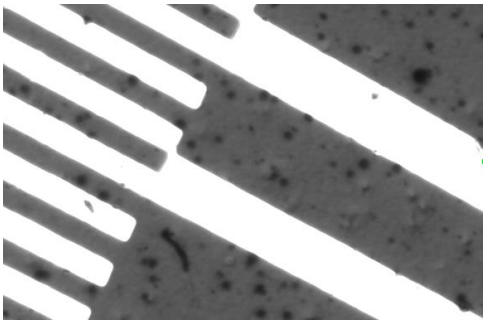
卷积核除了可以用作滤波，还能对图像进行梯度操作。比如图像从左向右看，像素值不断增大；或者从上到下看，像素值不断增大，这些都能反映出图像的梯度变化。Sobel 算子是一

种很经典的图像梯度提取算子，其本质是基于图像空间域卷积，背后的思想是图像一阶导数算子的理论支持。该算子主要用于获得数字图像的一阶梯度，这样可以把图像中的轮廓信息凸显出来。该算子包含两组 3x3 的矩阵，分别为水平方向及垂直方向，将之与图像作平面卷积，即可分别得出两个方向的亮度差分近似值。如下式卷积计算，其中 I 代表原始图像，Gx 及 Gy 分别代表经水平方向和垂直方向边缘检测的图像。

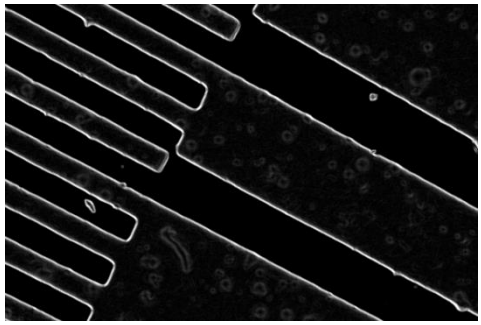
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

结合上述两个结果，在图像的每个像素点，按如下公式计算该点的梯度近似值，作为该像素点的输出值，其运算结果是一副边缘图像。

$$G = \sqrt{G_x^2 + G_y^2}$$



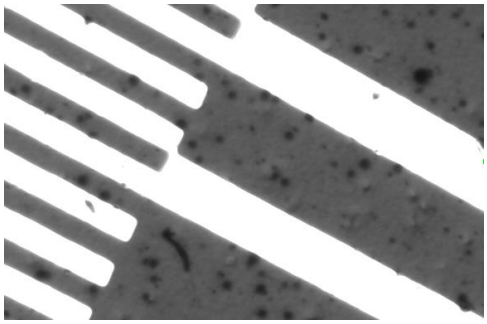
输入图像



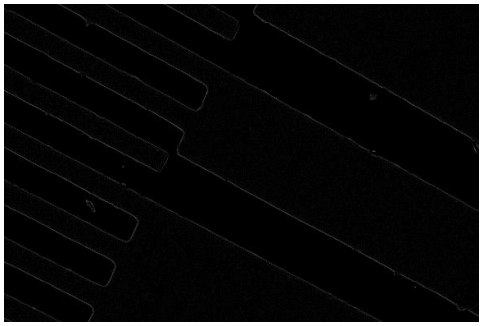
Sobel 图像

- 拉普拉斯

Sobel 算子属于一阶导数，获取到的边缘特征是极值，而拉普拉斯算子，是一种二阶导数滤波，获取到的边缘特征是过零点，能更好地提取边缘信息。



输入图像



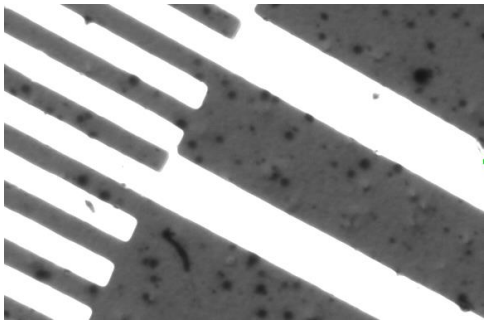
拉普拉斯图像

- 伽马变换

伽马变换是对输入图像灰度值进行非线性操作，使输出图像灰度值与输入图像灰度值呈指数关系，如下公式，这个指数就是 Gamma。

$$O = \left(\frac{I}{255}\right)^{\gamma} \times 255$$

伽马变换可以实现图像增强，当 gamma 值小于 1 时，图像整体亮度得到提升，同时低灰度处的对比度得到增加，更利于分辨低灰度值时的图像细节；当 gamma 值大于 1 时，则相反。



输入图像



gamma=5 图像

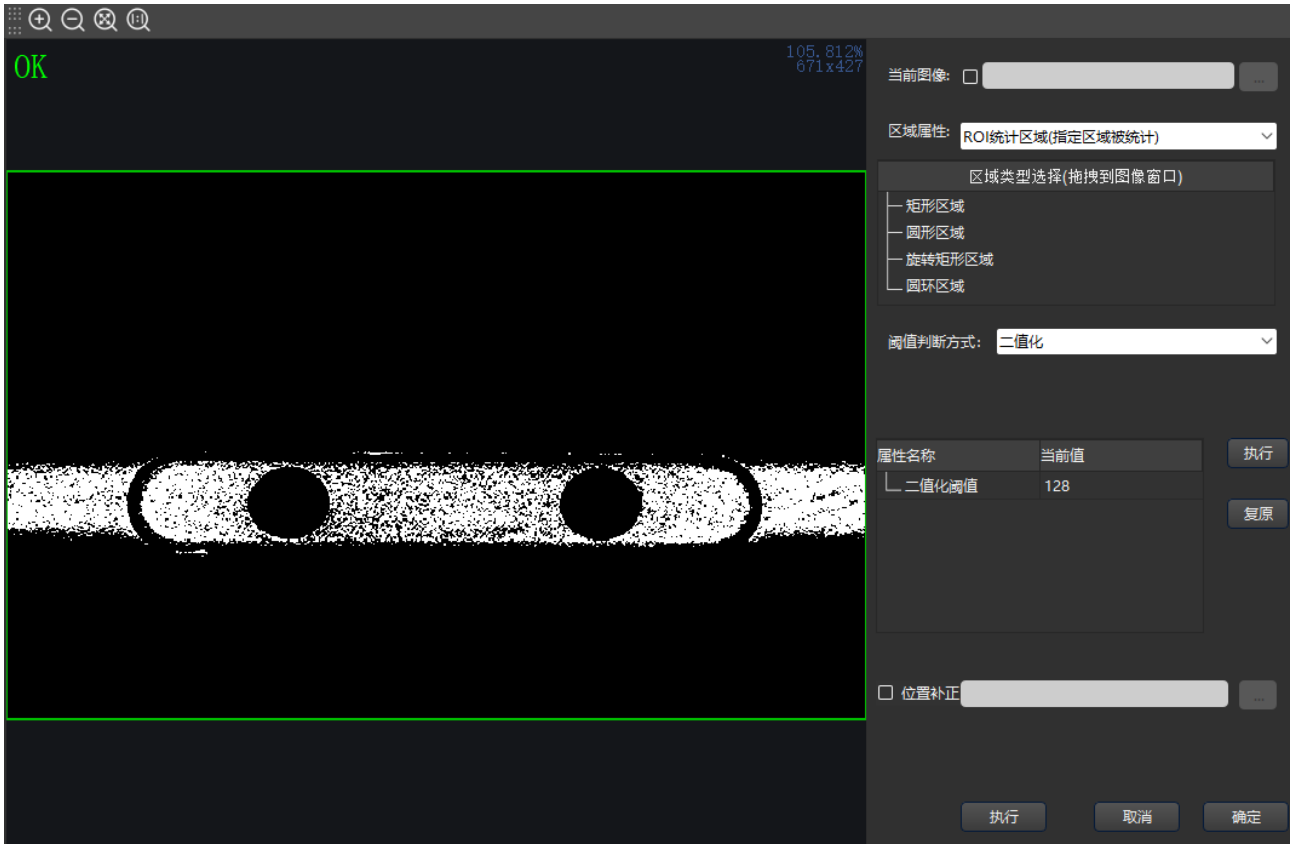
- 亮度调节

亮度调节可整体调节图像亮度和对比度，亮度和对比度调整公式如下，参数 α >0 和 β 通常称为比例系数和偏移量，I 为输入图像像素值，O 为输出图像像素值。

$$O=\alpha I+\beta$$

6.7.4 图像二值化

将原图像转换为仅包含 0 和 255 两个灰度值的图像，其目的是突出目标轮廓，方便后续图像处理。



图像二值化工具采用的二值化方式有如下九种，各方式的工作原理有所差异。

- 二值化

属于单阈值二值化，输入图像像素点的灰度值大于阈值时，则为目标（灰度值转换为 255），否则则为背景（灰度值转换为 0）。

- 反二值化

属于单阈值二值化，输入图像像素点的灰度值大于阈值时，则为背景（灰度值转换为 0），否则则为目标（灰度值转换为 255）。

- 截断

输入图像像素点的灰度值大于阈值时，则灰度值转换为阈值值，否则则灰度值保持不变。

- 阈值取零

输入图像像素点的灰度值大于阈值时，则灰度值保持不变，否则则灰度值转换为 0。

- 自动阈值

算法会遍历所有可能阈值，从而找到最佳的阈值 T，并根据该阈值进行图像二值化。输入图像中灰度值大于 T 的像素点为目标，小于的为背景。该方式无预设参数。

- 三角阈值

一种基于灰度直方图的图像分割方法，它可以自适应地选择阈值，因此可以适用于不同的图像。

- 自适应阈值

包括平均阈值和高斯阈值。平均阈值是在滤波核尺寸范围内对输入原图进行均值滤波，得到背景图像，然后计算原图和背景图上相同像素点的灰度值之差，根据差值、预设的阈值偏移量，判断像素点为目标还是背景。

高斯阈值与平均阈值类似，均为先滤波（一个对原图进行高斯滤波，一个对原图进行均值滤波）得到背景图，再将原图和背景图做差，最终得到满足预设关系的二值图。

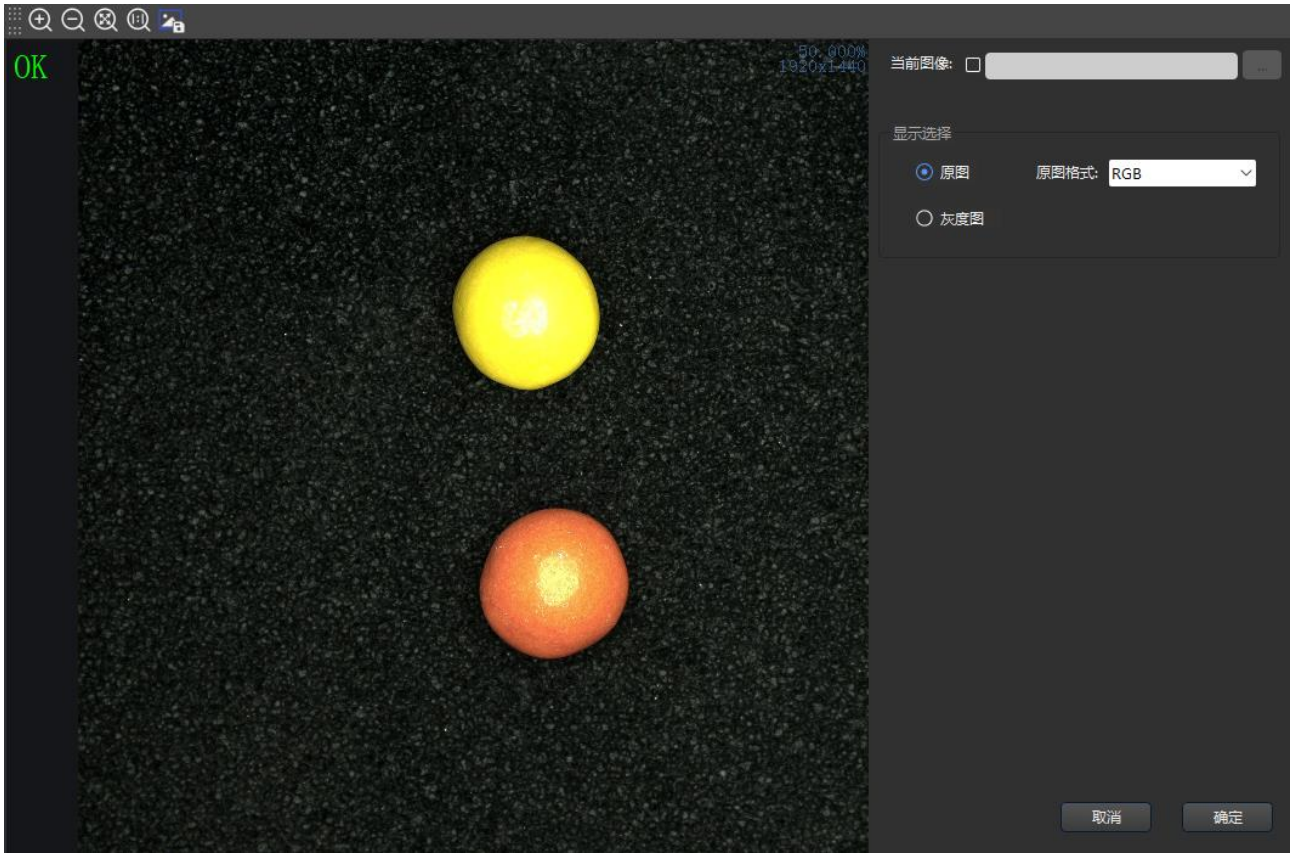
- 双阈值

根据预设的低阈值和高阈值进行图像二值化。输入图像像素点的灰度值大于低阈值或小于高阈值时，则为目标（灰度值转换为 255），否则则为背景（灰度值转换为 0）。

6.8 颜色处理

6.8.1 彩色转灰

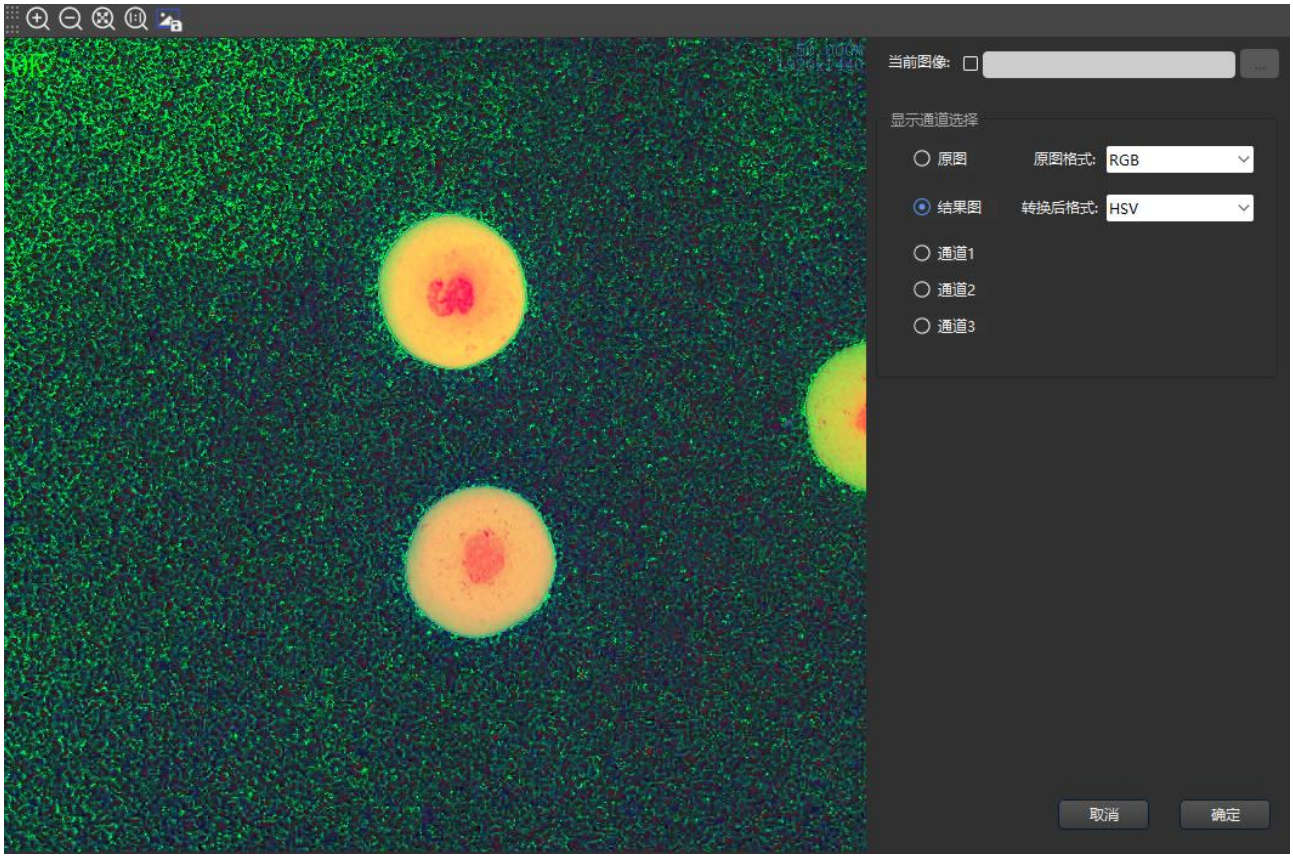
输入一幅 24 位彩色图像，转换为 8 位灰度图像输出。



- 当前图像：算子运行时执行处理的当前图像，一般绑定图像采集、图像处理等工具的输出图像。
- 原图格式：可选择 RGB、BGR 三通道彩色图像。
- 显示选择：设置当前需要显示的图像，可在当前画面中预览原图和转换后的灰度图像。

6.8.2 颜色空间转换

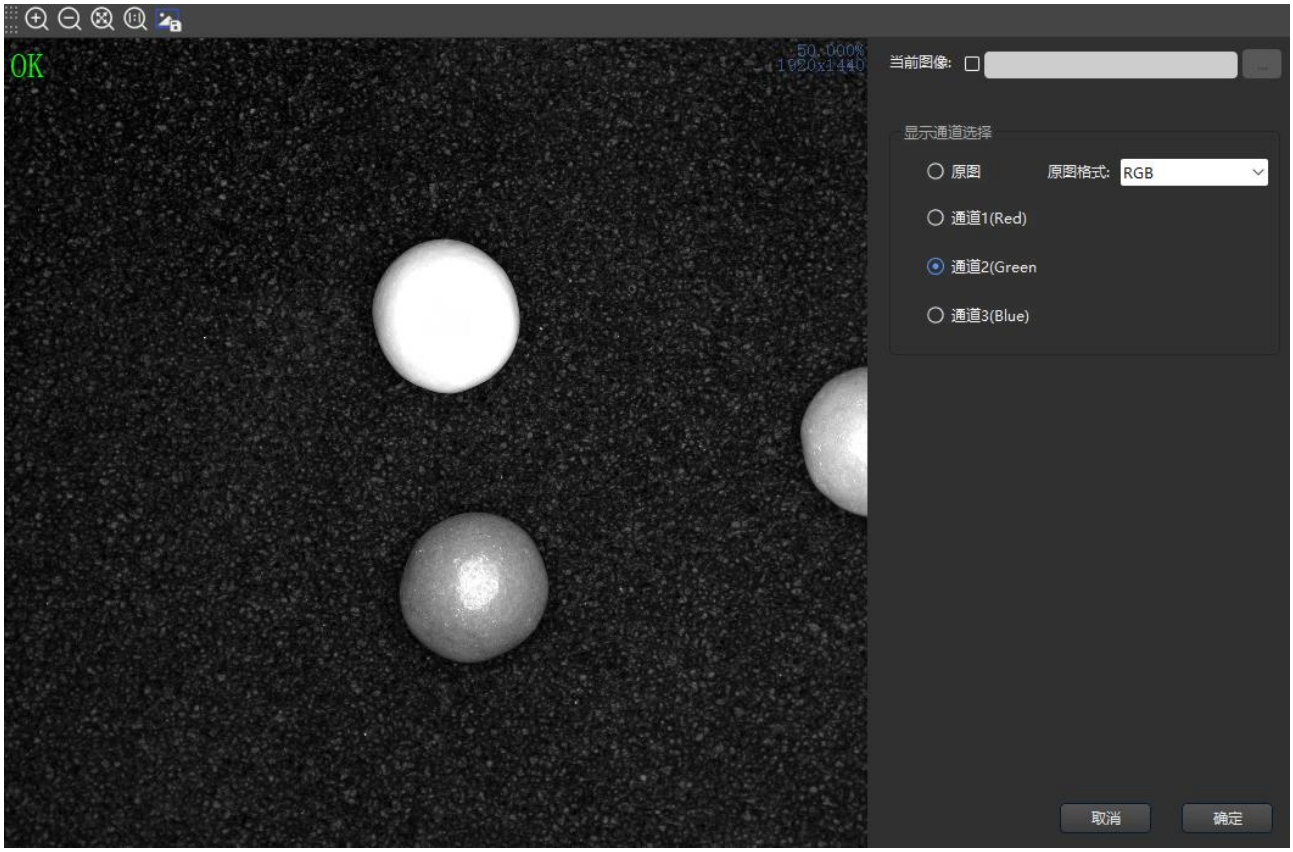
输入一幅 24 位彩色图像，对其进行颜色空间转换，用户可选择转 RGB、BGR、HSV、YUV、LAB 空间，并输出转换后图像指定颜色通道的灰度图像。



- 当前图像：算子运行时执行处理的当前图像，一般绑定图像采集、图像处理等工具的输出图像。
- 原图格式：可选择 RGB、BGR 三通道彩色图像。
- 转换后格式：可选择转 RGB、BGR、HSV、YUV、LAB 空间。
- 显示通道选择：设置当前需要显示的图像，可在当前画面中预览原图和转换后的彩色图像或单通道灰度图像。

6.8.3 图像通道分离

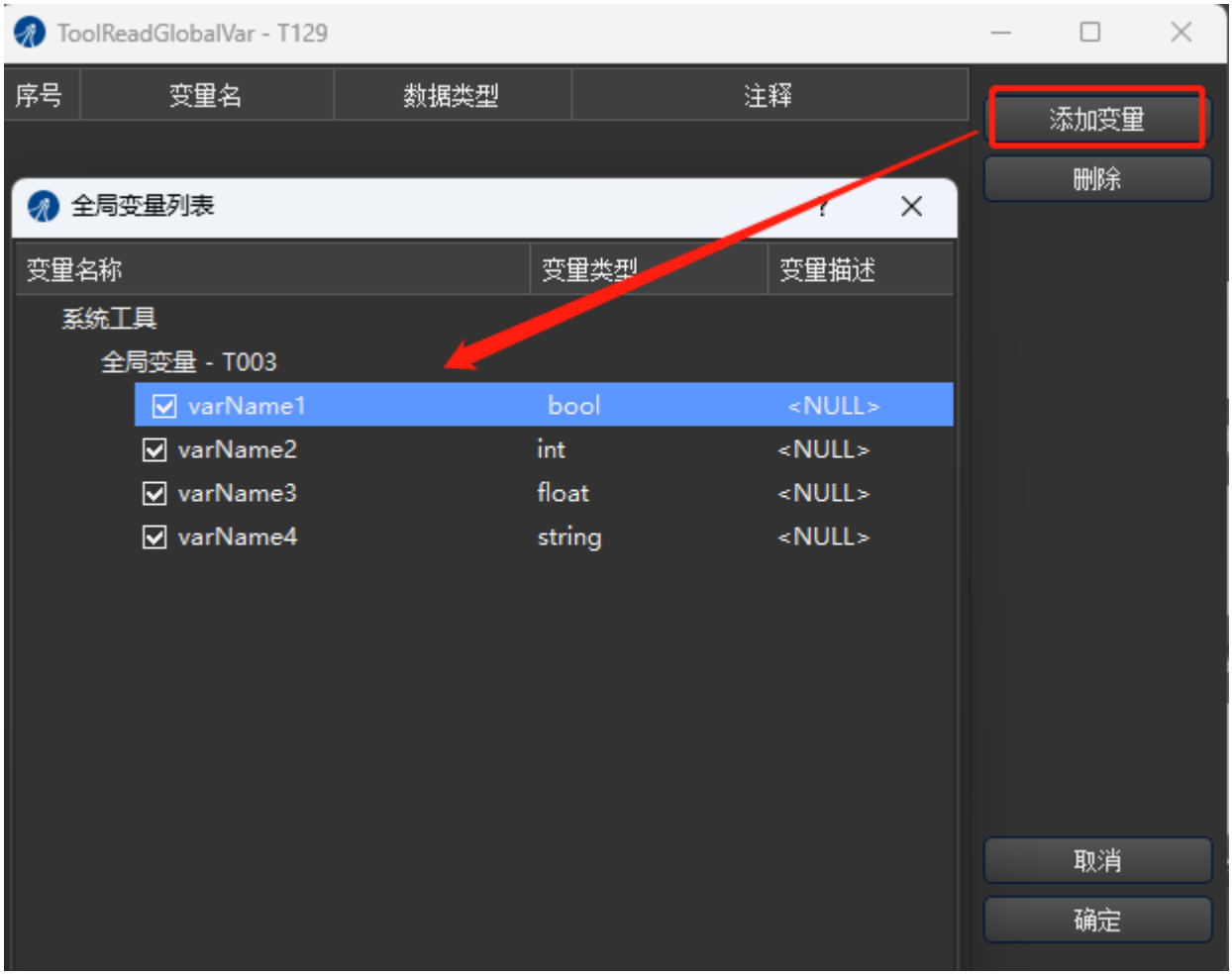
对一幅 24 位彩色图像，按 RGB 或 BGR 通道分离，输出每个通道的 8 位灰度图像。参数设置请参考颜色空间转换章节。



6.9 系统工具

6.9.1 读取全局变量

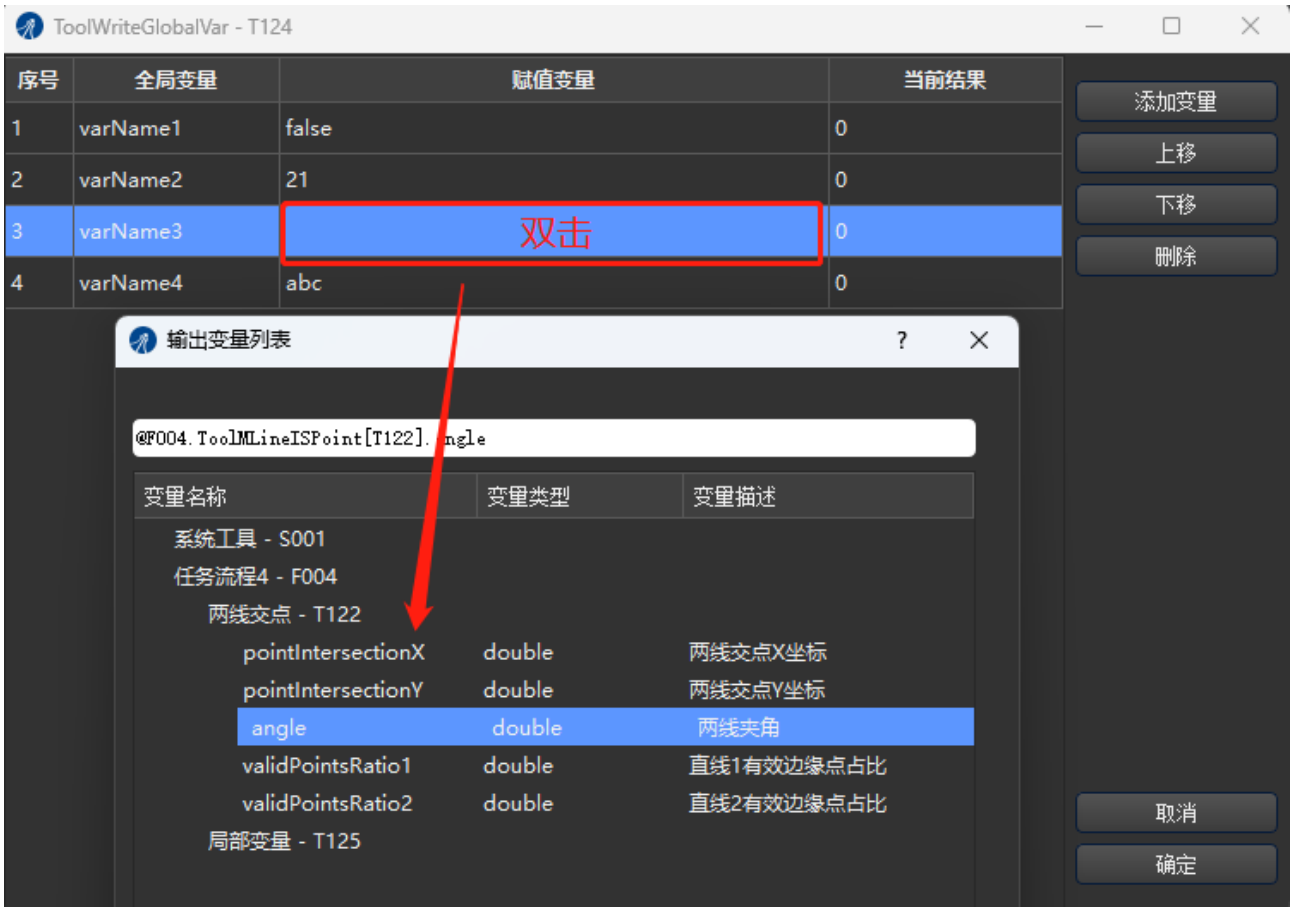
获取全局变量的值，线程安全的读取方式。需先添加全局变量，再使用全局变量。点击“添加变量”在全局变量列表中勾选要读取的变量。



6.9.2 写入全局变量

设置全局变量的值，线程安全的写入方式。需先添加全局变量，再使用全局变量。

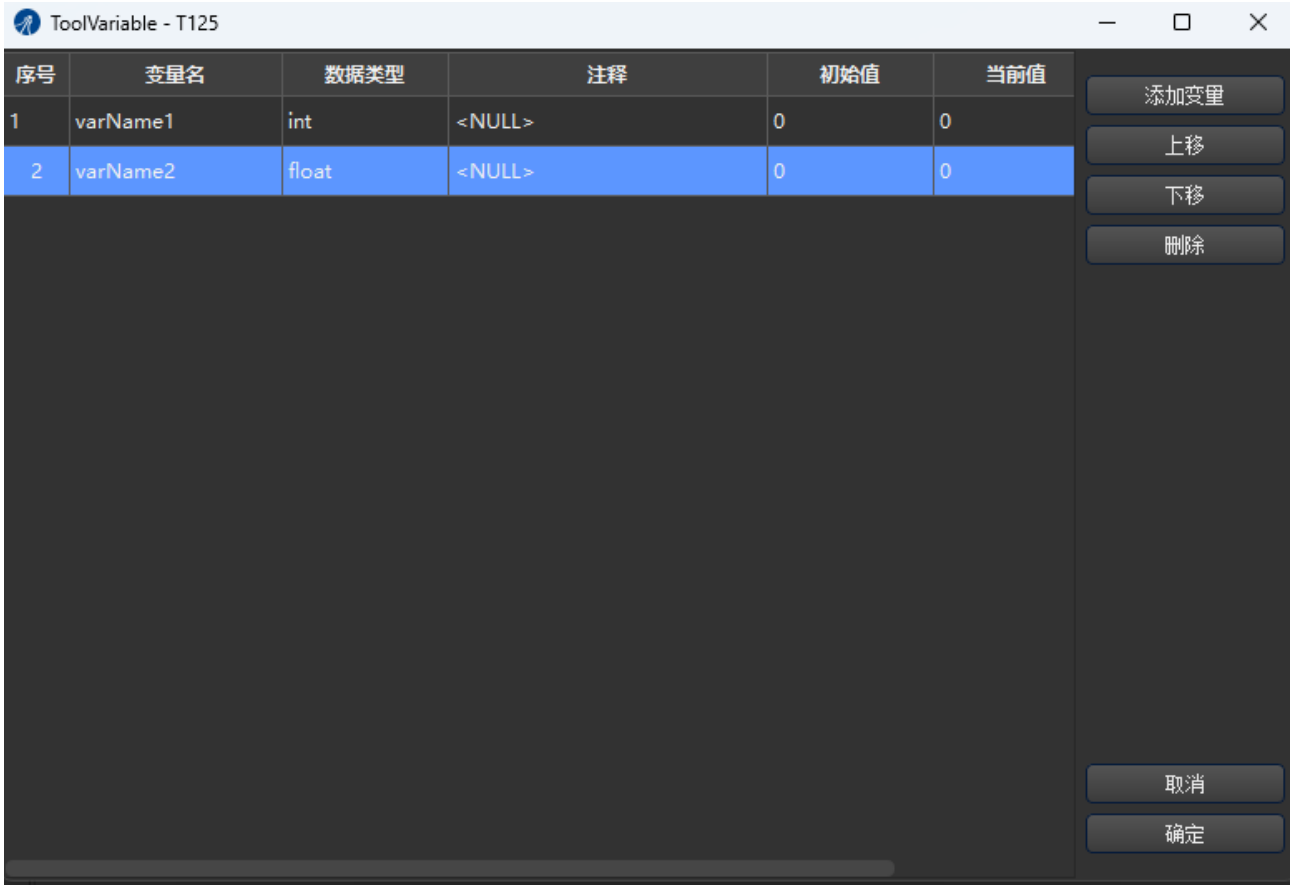
点击“添加变量”在全局变量列表中选择要写入的变量。添加变量后，双击“赋值变量”列，给添加的变量赋值。



- 添加变量：单击后弹出全局变量列表，选择写入的变量。
- 上移/下移：可对变量的位置进行上下调整。
- 删除：删除列表中当前选择的条目。
- 取消：取消添加的变量。
- 确定：保存添加的变量。

6.9.3 局部变量

局部变量是任务流程中自定义的变量，只能由创建它的任务流程调用或修改，可自定义变量名称、类型和初始值。



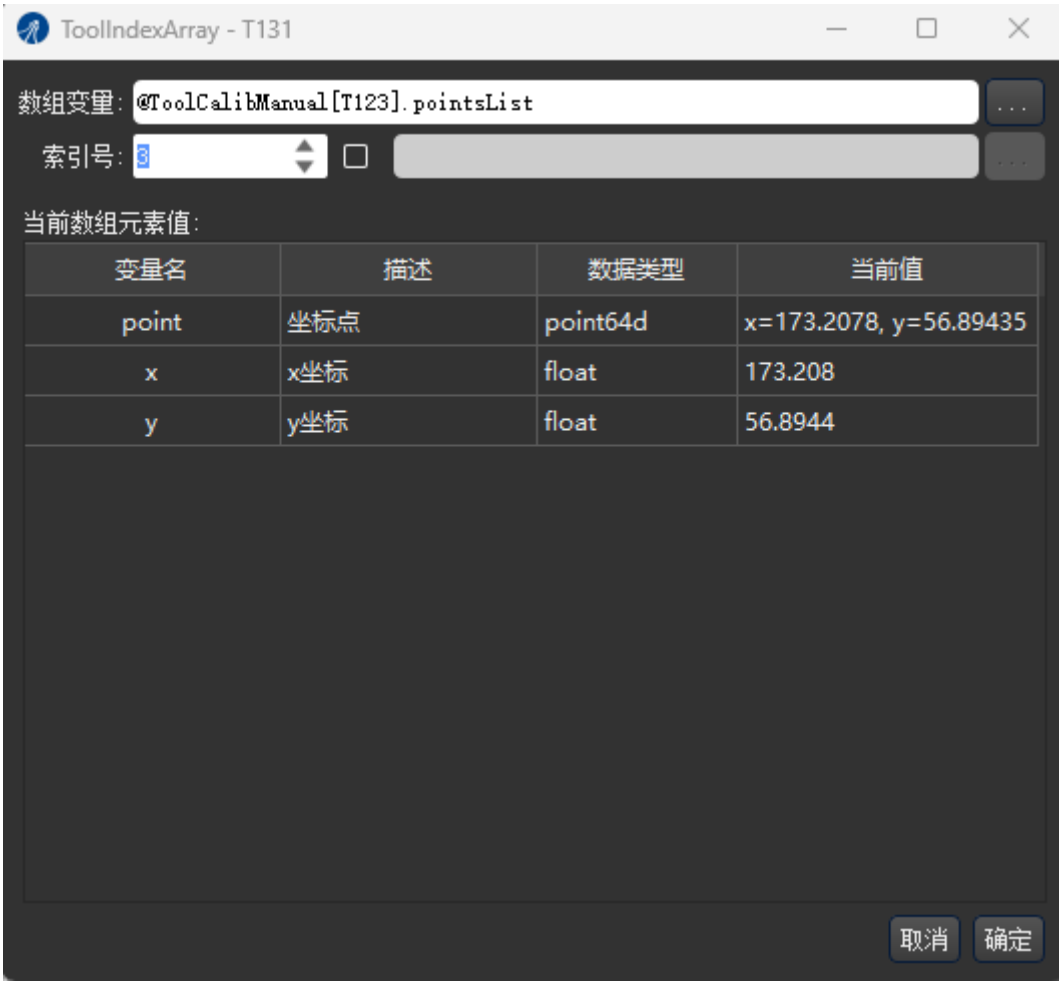
- 添加变量：单击后可新增全局变量。
- 上移/下移：可对变量的位置进行上下调整。
- 删除：删除列表中当前选择的条目。
- 取消：取消添加的变量。
- 确定：保存添加的变量。

6.9.4 写入局部变量

设置局部变量的值，需先使用“局部变量”工具创建变量。然后在局部变量列表中选择要写入的变量。操作方法参考写入全局变量。

6.9.5 数组索引

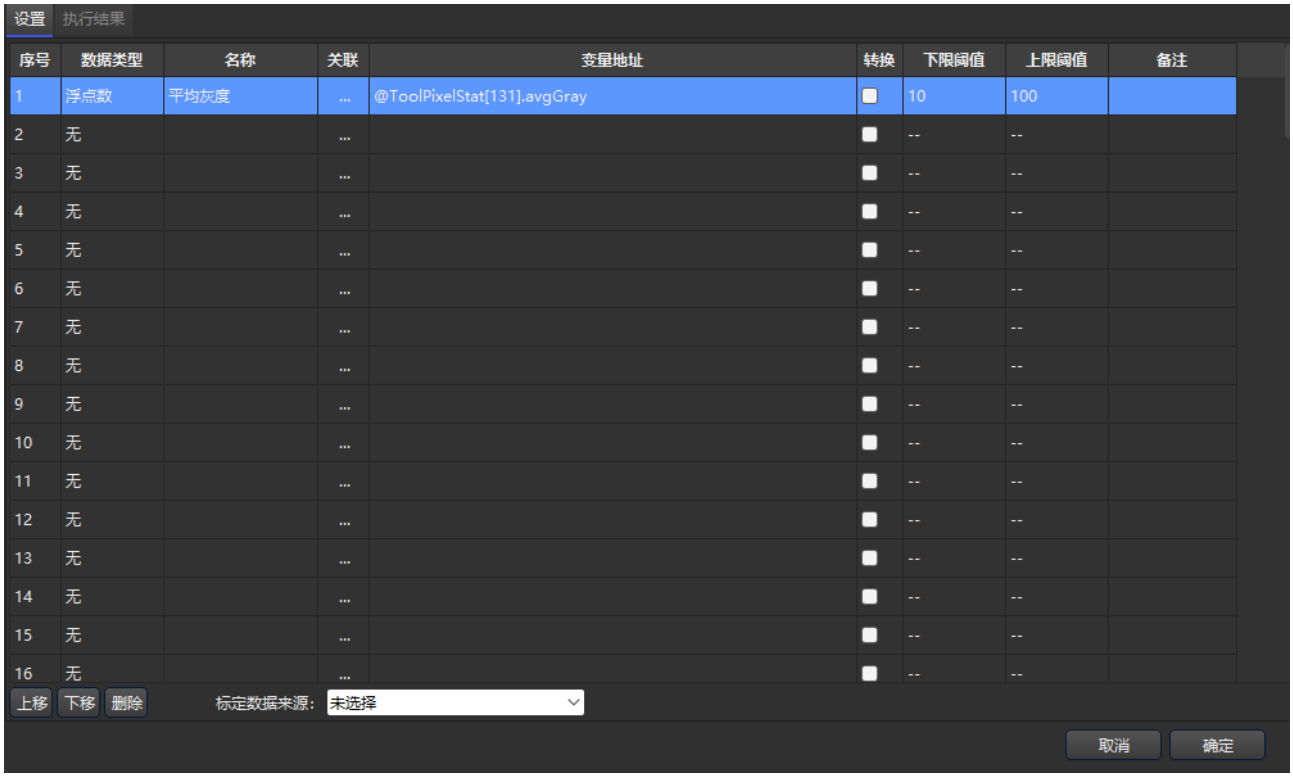
获取数组中指定索引号的元素。



- 数组变量：关联数组输出变量。
- 索引号：指定要获取的数组元素编号，编号范围[0,数组长度-1]。

6.9.6 阈值判断

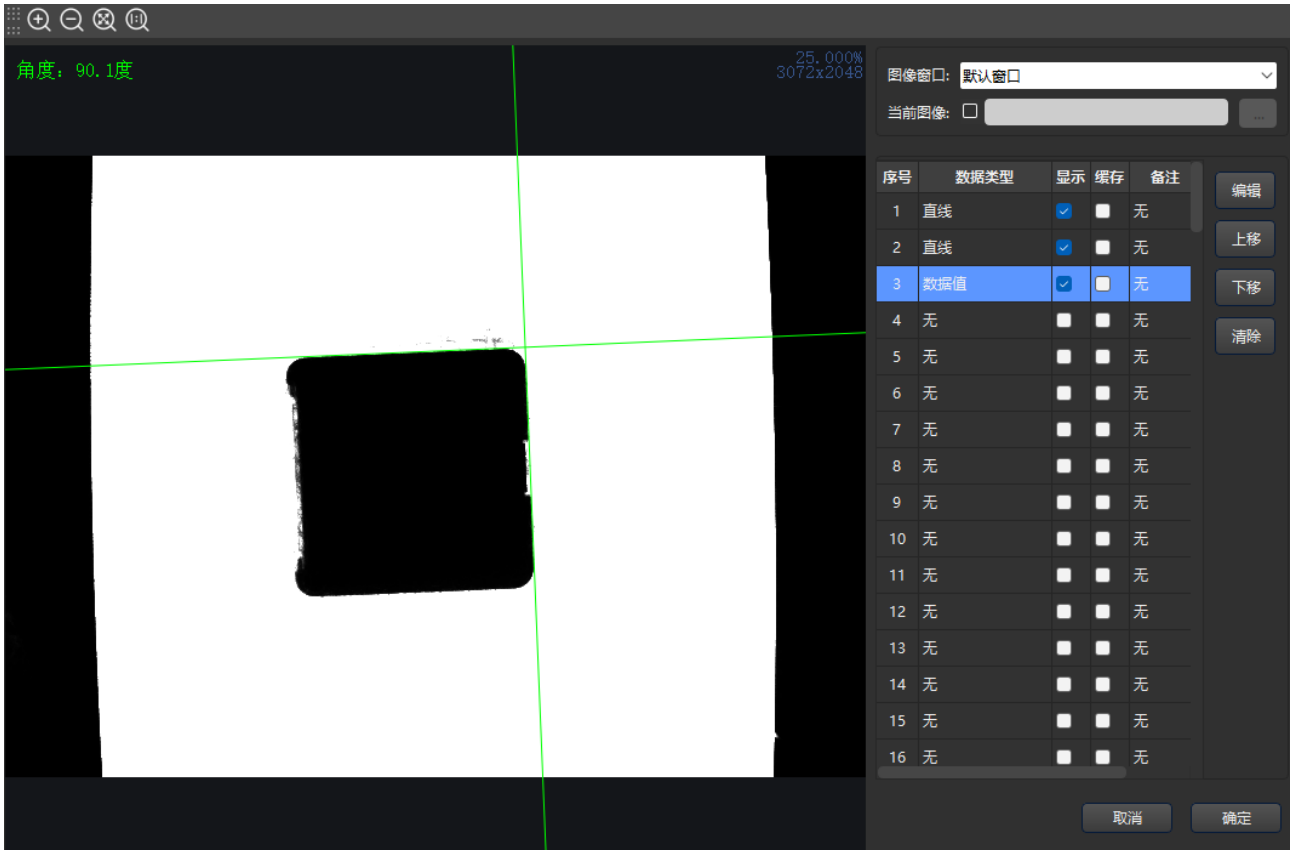
判断一组数据是否在设定的上下限阈值范围内，如果其中的一个数据超出范围则返回NG，一般用于输出结果的综合判定，也可使用像素坐标转换成物理坐标的值进行判断。



- 添加：添加需要判断上下限的数据。
- 删除：删除列表中当前选择的条目。
- 上移/下移：可对列表中当前选择的条目进行上下移动。

6.9.7 结果显示

用于在图像上显示检测结果数据，支持显示布尔值、数据值、字符串、点、线、圆、矩形。



- 图像设置

➢ 图像窗口：选择结果数据显示的窗口。

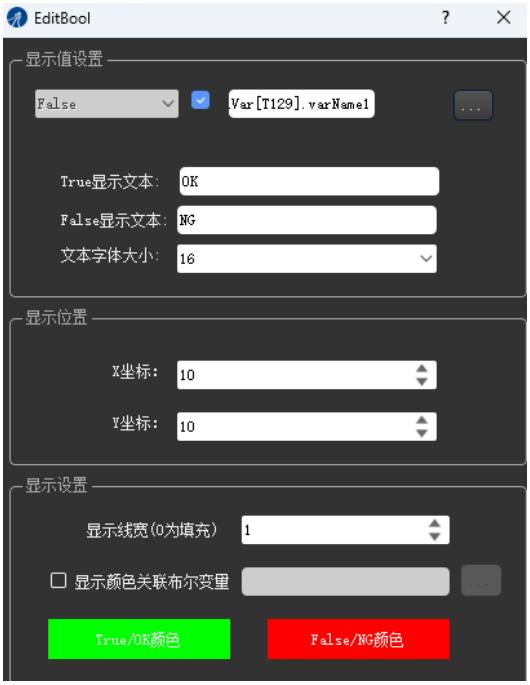
➢ 当前图像：工具运行时显示结果数据的目标图像，一般绑定图像采集、图像处理等工具的输出图像。

- 数据设置

在数据类型下拉列表中，选择显示的条目，单击“编辑”会弹出该条目的设置页面。

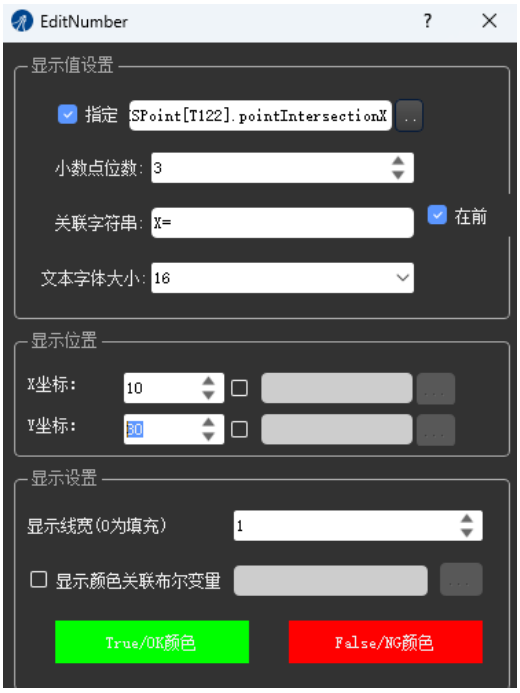


- 布尔值设置



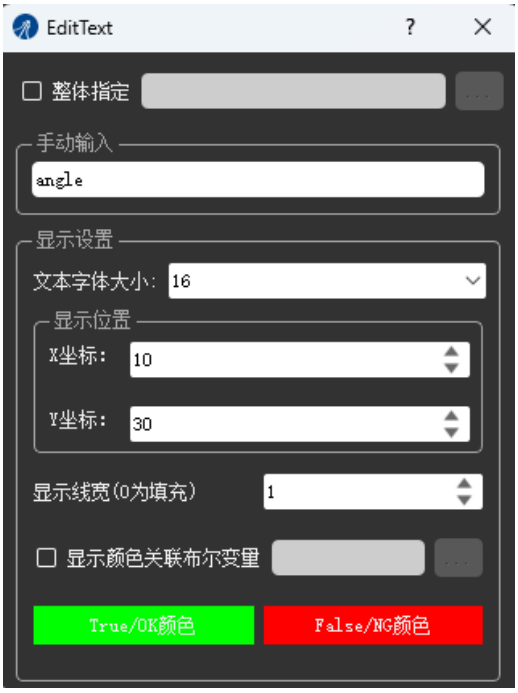
- 显示值需指定布尔变量来源。
- True 显示文本：设置布尔值为真时显示的文本。
- False 显示文本：设置布尔值为假时显示的文本。
- 文本字体大小：设置文本大小。
- X 坐标：设置文本显示的位置 X 轴坐标（窗口坐标系）。
- Y 坐标：设置文本显示的位置 Y 轴坐标（窗口坐标系）。
- 显示线宽：参数无效。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

- 数据值设置



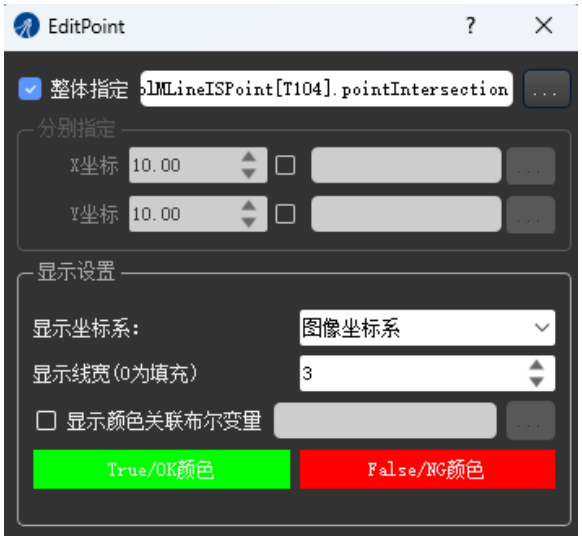
- 指定：设置需要显示的数据值来源，支持整型和浮点型。
- 小数点位数：指定需要显示的数据值小数位数。
- 关联字符串：在数据值前或后增加字符串一起显示。
- 文本字体大小：设置文本大小。
- X 坐标：设置文本显示的位置 X 轴坐标（窗口坐标系），可以关联来源。
- Y 坐标：设置文本显示的位置 Y 轴坐标（窗口坐标系），可以关联来源。
- 显示线宽：参数无效。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

- 字符串设置



- 整体指定：设置需要显示的字符串来源。
- 文本字体大小：设置文本大小。
- X 坐标：设置文本显示的位置 X 轴坐标（窗口坐标系）。
- Y 坐标：设置文本显示的位置 Y 轴坐标（窗口坐标系）。
- 显示线宽：参数无效。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

- 点设置



- 整体指定：设置需要显示的点来源。
- 分别指定：点按 X 坐标 + Y 坐标的方式指定，可以关联来源。
- 显示坐标系：可以显示在图像坐标系，也可以显示在窗口坐标系。
- 显示线宽：指定点的十字标识线的宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

- 线设定



- 整体指定：设置需要显示的线段或直线来源。
- 分别指定：直线按中心点 + 方向向量的方式指定，线段按起点 + 终点的方式指定，可以关联来源。
- 显示线宽：指定线段或直线的轮廓宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

- 圆设置



- 整体指定：设置需要显示的圆来源。
- 分别指定：按圆心坐标 + 半径的方式指定圆，可以关联来源。
- 显示线宽：指定圆轮廓线的宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

- 矩形设置

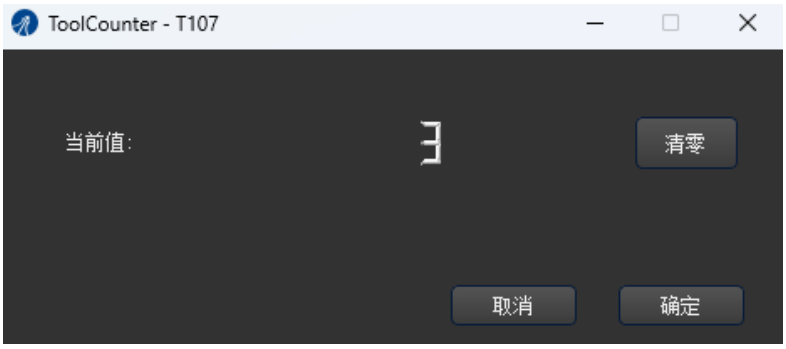


- 整体指定：设置需要显示的矩形来源。

- 分别指定：按矩形中心坐标 + 矩形宽高 + 角度的方式指定矩形，可以关联来源。
- 显示线宽：指定矩形轮廓线的宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

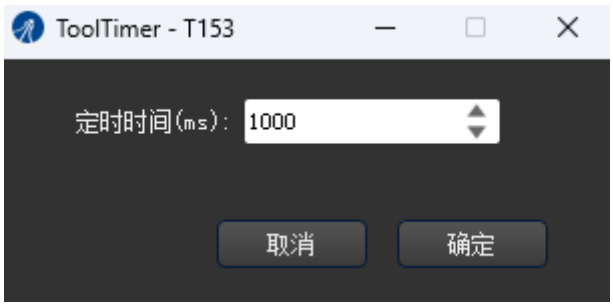
6.9.8 计数器

统计流程执行次数。



6.9.9 定时器

延时设定的时间，再执行之后的程序。

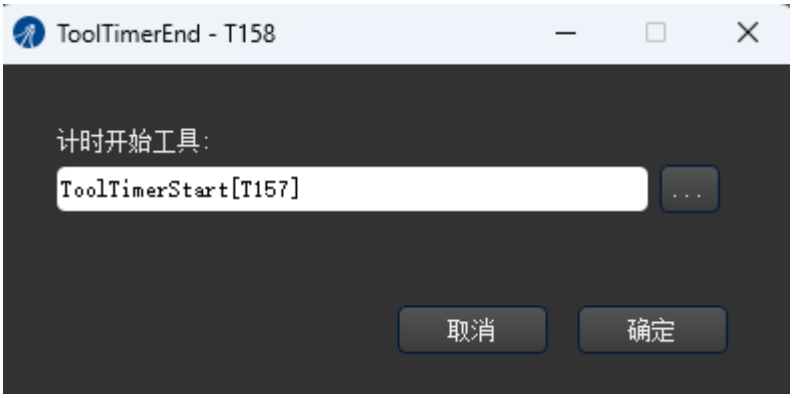


6.9.10 计时开始

记录开始计算时间的位置，该工具无参数配置。与“计时结束”工具搭配使用，用于计算流程中某一段落执行完成所需要的时间。

6.9.11 计时结束

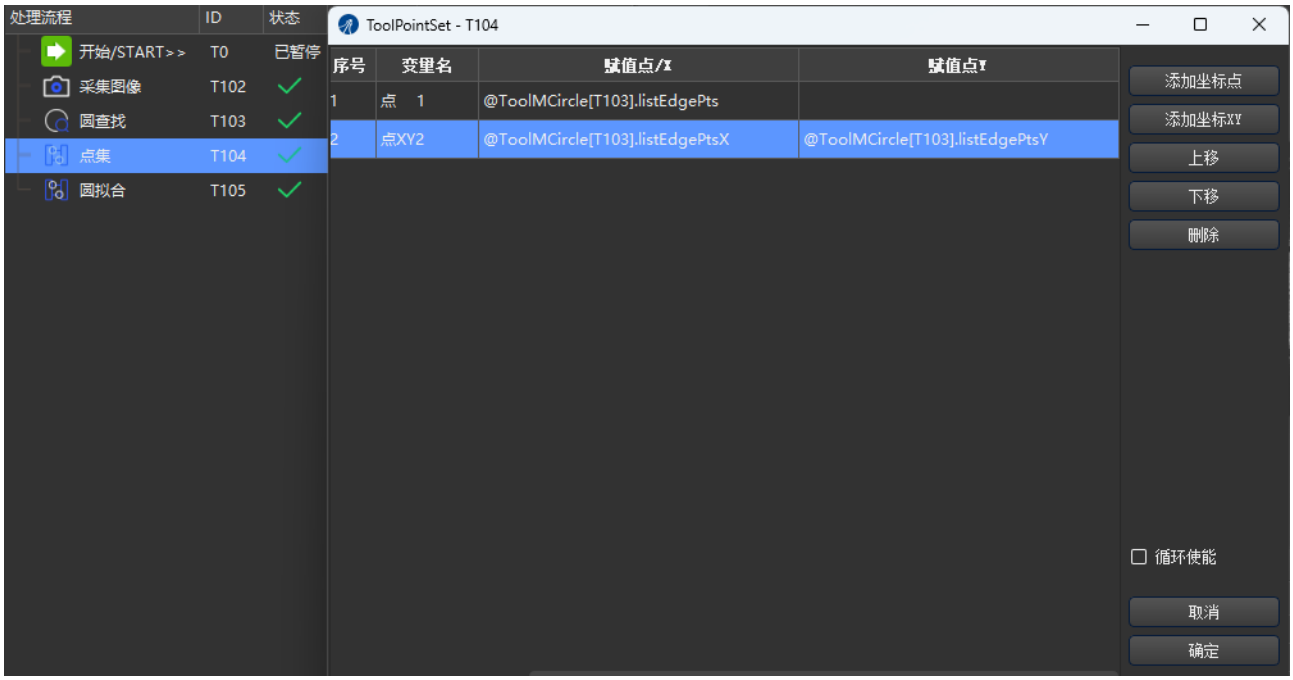
计算“计时开始”到“计时结束”所产生的时间差，单位为毫秒。



➢ 计时开始工具：关联统计耗时的“计时开始”工具。

6.9.12 点集

将其他工具输出的数据组合为点的集合，用于后续工具获取或传输，最多添加 128 组点集，最大 10000 个点坐标，点集支持点和坐标两种输入。



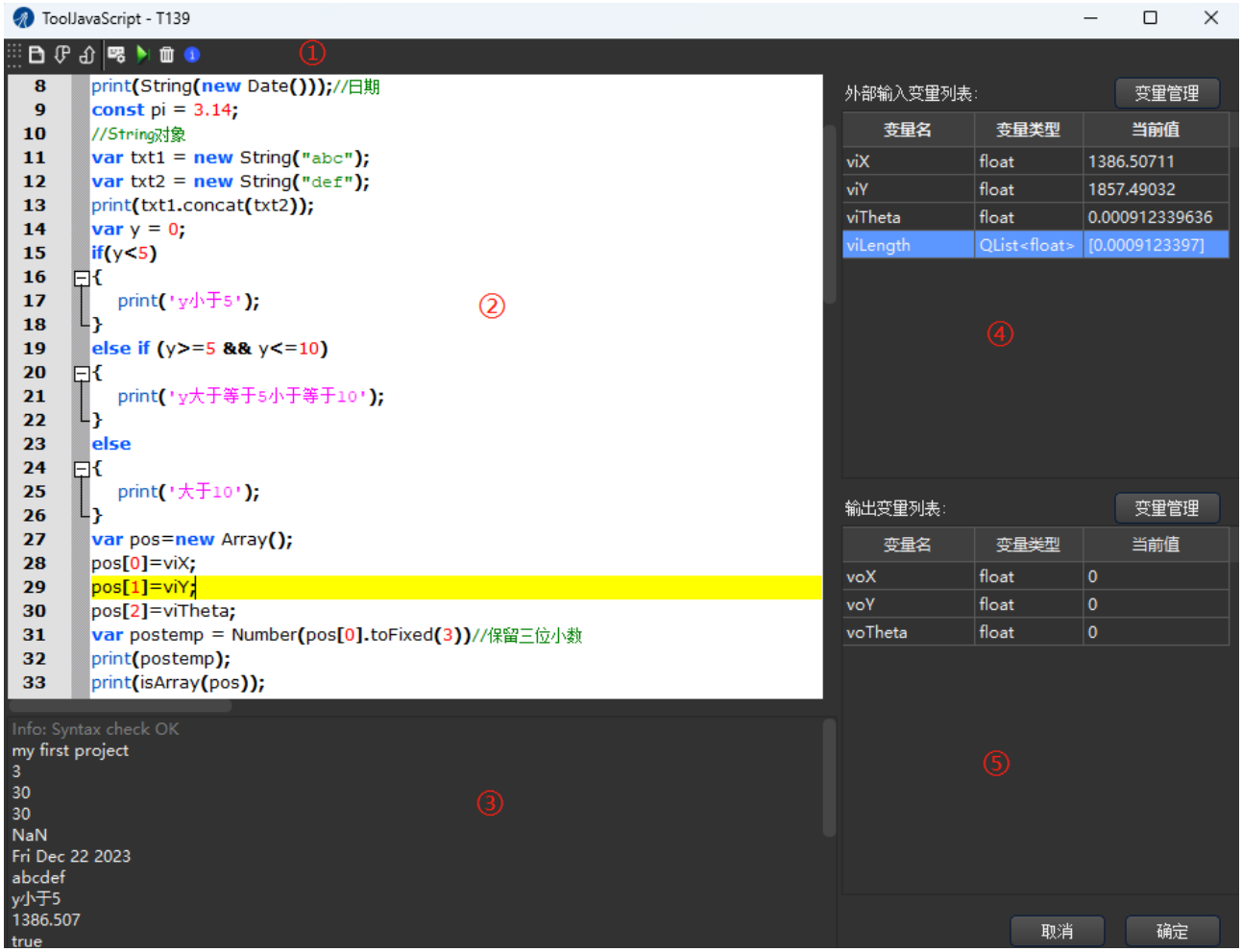
添加坐标点：按点的方式添加单个点或多个点。

添加坐标 XY：按 XY 坐标的方式添加单个或多个坐标。

循环使能：开启循环使能后，需放入循环模块使用，将每次循环的点汇集，大于 10000 个点时，后续点舍弃。

6.9.13 JavaScript 脚本

JavaScript 是一个轻量级，但功能强大的编程语言，使用 JavaScript 脚本可以进行复杂的数据处理。脚本工具可保存和加载已编写的脚本内容，脚本代码长度无限制，支持导入导出。



该脚本工具分为如下五个区域：

① 工具栏

主要对脚本进行保存、导入、导出、运行测试及删除调试信息。

② 代码编辑区

JavaScript 脚本编写区域，用户直接敲入代码即可。

③ 调试窗口

程序编译信息输出，编译失败后将打印异常信息，用户也可通过 print 函数输出中间变量信息。

④ 变量输入

外部传入脚本工具的变量，单击“变量管理”可添加变量并关联输入来源。

⑤ 变量输出

脚本工具对外输出的变量，单击“变量管理”可添加输出变量。

6.9.13.1 输出

在编写脚本程序时，往往需要监视过程值，来确认代码执行是否正确，此时可以使用 print 语句打印输出到调试窗口，如下所示。

```
print("my first project");
var a=30;
print(a);
```

6.9.13.2 注释

注释可提高代码的可读性，脚本不会执行注释代码。单行注释以 // 开头，多行注释以 /* 开始，以 */ 结尾。

6.9.13.3 变量

变量用于存放值（如 a=5）和表达式（如 z=x+y）。变量命名应按以下原则：

- 变量必须以字母开头
- 变量也能以 $ 和 _ 符号开头（不推荐）
- 变量名称对大小写敏感（a 和 A 是不同的变量）

创建变量通常称为"声明"变量，使用 var 关键词来声明变量，如下声明变量 theta。

```
var theta;
```

变量声明之后，该变量没有值，未使用值来声明的变量，其值实际上是 undefined，Undefined 这个值表示变量不含有值，如需向变量赋值，请使用等号。

```
theta= 90;
```

可以通过将变量的值设置为 null 来清空变量。

theta= null;

也可以在声明变量时对其赋值 var theta= 90;

支持一条语句中声明多个变量。该语句以 var 开头，并使用逗号分隔变量。

var x=3.14, i=10, name="Zoe";

当然，为了明确数据类型，可以在声明新变量时，使用关键词 "new" 来声明其类型。

var name=new String;

var x= new Number;

var y= new Boolean;

var cars= new Array;

var person= new Object;

6.9.13.4 数据类型

JavaScript 脚本有 6 种不同的数据类型：字符串（string）、数字(number)、布尔(boolean)、对象(object)、函数(function)、符号（symbol）。3 种对象类型：对象(Object)、数组(Array)和日期（Date）。2 个不包含任何值的数据类型：空（null）和未定义（undefined）。

变量的数据类型可以使用 typeof 操作符来查看，实例如下：

```
typeof "Zoe"                    // 返回 string
typeof 6.35                     // 返回 number
typeof false                    // 返回 boolean
typeof [1,2,3]                  // 返回 object
typeof new Date()               // 返回 object
typeof {name:'Lucy', age:20} // 返回 object
typeof NaN                      // 返回 number
typeof function () {}           // 返回 function
typeof myHome                   // 返回 undefined (如果 myHome 没有声明)
typeof null                     // 返回 object
```

通过上例可以看出，如果对象是 Array 或 Date ，则无法通过 **typeof** 来判断他们的类型，因为都是返回 object，此时可通过 constructor 属性，在脚本语言中 constructor 属性返回所有变量的构造函数。实例如下：

```
"John".constructor                 // 返回函数 String()  { [native code] }
(3.14).constructor                 // 返回函数 Number()  { [native code] }
false.constructor                  // 返回函数 Boolean() { [native code] }
[1,2,3].constructor                // 返回函数 Array()   { [native code] }
{name:'Lucy', age:20}.constructor  // 返回函数 Object()  { [native code] }
new Date().constructor             // 返回函数 Date()    { [native code] }
function () {}.constructor         // 返回函数 Function(){ [native code] }
```

使用 constructor 属性来查看对象是否为数组 (包含字符串 "Array"):

```
function isArray(myArray)
{
   return myArray.constructor.toString().indexOf("Array") > -1;
}
```

使用 constructor 属性来查看对象是否为日期 (包含字符串 "Date"):

```
function isDate(myDate)
{
   return myDate.constructor.toString().indexOf("Date") > -1;
}
```

6.9.13.4.1 字符串

字符串是存储字符（如 "position"）的变量，引号中的任意文本都是字符串，可以使用单引号或双引号。

```
var answer = "right";
var answer = 'right';
```

反斜杠是一个转义字符，转义字符将特殊字符转换为字符串字符，转义字符 (\) 可以用于转义撇号、换行、引号等其他特殊字符。例如，下面的 str 将输出 This is big "A\"。

var str = "This is big \"A\"";

下表中列举了在字符串中可以使用转义字符转义的特殊字符。

属性	描述
\'	单引号
\"	双引号
\\	反斜杠
\n	换行
\r	回车
\t	tab(制表符)
\b	退格符
\f	换页符

字符串属性如下表：

属性	描述
constructor	返回创建字符串属性的函数
length	返回字符串的长度
prototype	允许您向对象添加属性和方法

字符串方法如下表：

方法	描述
charAt()	返回指定索引位置的字符
charCodeAt()	返回指定索引位置字符的 Unicode 值
concat()	连接两个或多个字符串，返回连接后的字符串
fromCharCode()	将 Unicode 转换为字符串
indexOf()	返回字符串中检索指定字符第一次出现的位置
lastIndexOf()	返回字符串中检索指定字符最后一次出现的位置
match()	找到一个或多个正则表达式的匹配
replace()	替换与正则表达式匹配的子串
search()	检索与正则表达式相匹配的值
slice()	提取字符串的片断，并在新的字符串中返回被提取的部分

split()	把字符串分割为子字符串数组
substr()	从起始索引号提取字符串中指定数目的字符
substring()	提取字符串中两个指定的索引号之间的字符
toLowerCase()	把字符串转换为小写
toUpperCase()	把字符串转换为大写
toString()	返回字符串对象值
trim()	移除字符串首尾空白
valueOf()	返回某个字符串对象的原始值

实例：

```
var str="Hello World!";
print(str.length);// 输出 12
print(str.indexOf("or"));//输出 7
txt="a,b,c" ;
txt.split(",");  // 使用逗号分隔提取 a b c，可用数组 new Array()存储提取后的值
```

6.9.13.4.2 数字

JavaScript 脚本不是类型语言。与许多其他编程语言不同，它不定义不同类型的数字，比如整数、短、长、浮点等，在脚本语言中数字统一为 Number。所有的数字都是浮点型类型。JavaScript 采用 IEEE754 标准定义的 64 位浮点格式表示数字，它能表示最大值（Number.MAX_VALUE）为 ±1.7976931348623157e+308，最小值（Number.MIN_VALUE）为 ±5e-324。默认情况下，数字为十进制；如果前缀为 0，则为八进制数；如果前缀为 0x，则为十六进制数。

```
var x=34.56;      //使用小数点
var y=12;         //不使用小数点
var z=123e5;      // 12300000
var u = 0377;     //前缀 0 为 8 进制
var v = 0xFF;     //前缀 0x 为 16 进制
```

数字可以是数字也可以是对象。如下实例：

var x = 123;

var y = new Number(123);

typeof(x) // 返回 Number

typeof(y) // 返回 Object

Number 属性

方法	描述
Number.MAX_VALUE	最大值，1.7976931348623157e+308
Number.MIN_VALUE	最小值，5e-324
Number.NaN	非数字，NaN
Number.NEGATIVE_INFINITY	负无穷，在溢出时返回
Number.POSITIVE_INFINITY	正无穷，在溢出时返回

数字原型的方法：

方法	描述
toExponential()	把对象的值转换为指数计数法。如：1.23e+2

toFixed()	把数字转换为字符串，结果的小数点后有指定位数的数字。
	var a=123;
	b=a.toFixed(2); // b="123.00"

toPrecision()	把数字格式化为指定的长度。如下例中，a=123 中，3 会由于精度限制消失。
	var a=123;
	b=a.toPrecision(2); // b="1.2e+2"

数字对象的方法：

方法	描述
Number.parseFloat()	将字符串转换成浮点数。
Number.parseInt()	将字符串转换成整型数字。
Number.isFinite()	判断传递的参数是否为有限数字。
Number.isInteger()	判断传递的参数是否为整数。
Number.isNaN()	判断传递的参数是否为 isNaN()。

Number.isSafeInteger()	判断传递的参数是否为安全整数。

6.9.13.4.3 布尔

布尔（逻辑）只有两个值：true 或 false，常用在条件语句中。

var a=true;

var b=false;

Boolean（布尔）对象用于将非布尔值转换为布尔值（true 或者 false）。布尔对象定义如下：

var myBoolean=new Boolean();

Boolean 对象属性

方法	描述
constructor	返回对创建此对象的 Boolean 函数的引用

Boolean 对象方法

方法	描述
toString()	把布尔值转换为字符串，并返回结果。
valueOf()	返回 Boolean 对象的原始值。

6.9.13.4.4 数组

数组（Array）对象是一组数据的集合，数组下标是基于零的，所以第一个元素是 [0]，第二个是 [1]，以此类推，下面的代码创建名为 cars 的数组：

var cars=new Array();

cars[0]="A";

cars[1]="B";

cars[2]="M";

Array 对象属性

属性	描述
constructor	返回创建数组对象的原型函数。
length	设置或返回数组元素的个数。

prototype	允许你向数组对象添加属性或方法。如下实例给数组增加一个新方法 myFun，并在 fruits 数组实例中调用，将其数组值全部转为大写。
	Array.prototype.myFun = function()
	{
	for (i=0;i<this.length;i++)
	{
	this[i]=this[i].toUpperCase();
	}
	}
	var fruits=["Banana","Orange","Apple","Mango"];
	fruits.myUcase();

Array 对象方法

方法	描述

concat()	连接两个或更多的数组，并返回结果。实例如下：
	var a = ["year", "month"];
	var b = ["date", "time"];
	var c = a.concat(b);

copyWithin()	从数组的指定位置拷贝元素到数组的另一个指定位置中。语法如下： array.copyWithin(target, start, end)

every()	检测数值元素的每个元素是否都符合条件。如下实例为检测数组 *ages* 的所有元素是否都大于等于 18。
	var ages = [51, 22, 16, 35];
	function checkAdult(age) {
	return age >= 18;
	}

	ages.every(checkAdult);
filter()	检测数值元素，并返回符合条件所有元素的数组。在上例中返回数组 *ages* 中所有元素都大于 18 的元素。 ages.filter(checkAdult);
indexOf()	搜索数组中的元素，并返回它所在的位置。语法如下： array.indexOf(*item*,*start*);
isArray()	判断对象是否为数组。语法如下： Array.isArray(obj);
join()	把数组的所有元素放入一个字符串。语法如下： array.join(*separator*); 省略 *separator* 参数，则使用逗号作为分隔符。
lastIndexOf()	搜索数组中的元素，并返回它最后出现的位置。语法如下： array.lastIndexOf(*item*,*start*);

map()	返回一个新数组，数组中的元素为原始数组元素调用函数处理后的值。语法如下：
	array.map(function(currentValue,index,arr), thisValue);
	如下实例返回一个数组，数组中元素为原始数组的平方根。
	var numbers = [4, 9, 16, 25];
	numbers.map(Math.sqrt);

pop()	删除数组的最后一个元素并返回删除的元素。语法如下： array.pop()
push()	向数组的末尾添加一个或更多元素，并返回新的长度。语法如下： array.push(*item1*, *item2*, ..., *itemX*);

reduce()	将数组元素计算为一个值（从左到右）。语法如下：
	array.reduce(function(total, currentValue, currentIndex, arr), initialValue);
	如下实例求 numbers 数组的和为 54。
	var numbers = [4, 9, 16, 25]];
	function getSum(total, num)

	{
	return total + num;
	}
	numbers.reduce(getSum);

reduceRight()	将数组元素计算为一个值（从右到左）。
reverse()	反转数组的元素顺序。语法如下： array.reverse();
shift()	删除并返回数组的第一个元素。语法如下： array.shift();
slice()	选取数组的一部分，并返回一个新数组。语法如下： array.slice(*start*, *end*);

some()	检测数组元素中是否有元素符合指定条件。语法如下：
	array.some(function(currentValue,index,arr),thisValue);
	如下实例将输出 true。
	var ages = [51, 22, 16, 35];
	function checkAdult(age) {
	return age >= 18;
	}
	ages. some (checkAdult);

sort()	对数组的元素进行排序。语法如下： array.sort();
splice()	从数组中添加或删除元素，会改变原始数组。语法如下： array.splice(*indexadd*,*howmanydel*,*item1*,.....,*itemX*);
toString()	把数组转换为字符串，并返回结果。语法如下： array.toString();
unshift()	向数组的开头添加一个或更多元素，并返回新的长度。语法如下： array.unshift(*item1*,*item2*, ..., *itemX*);
valueOf()	返回数组对象的原始值。语法如下：

	array.valueOf();

6.9.13.4.5 对象

JavaScript 对象是拥有属性和方法的数据。对象也是一个变量，但对象可以包含多个值（多个变量），每个值以 name:value 对呈现。在以下实例中，3 个值 ("Axis", 500, 100) 赋予变量 point。

```
var point = {name:"Axis", x:500, y:100};
```

- 对象定义

可以使用字符来定义和创建 JavaScript 对象，定义对象可以跨越多行。

```
var robot = {
    name:"Nio",
    x:10,
    y:10,
    z:30,
    methodSum:function(a,b){
            return a+b;
        }
    };
```

- 对象属性

对象是变量的容器"。但是，通常认为 "JavaScript 对象是键值对的容器"。键值对通常写法为 name : value (键与值以冒号分割)，键值对即为对象属性。

- 访问对象属性

可以通过两种方式访问对象属性:

```
person.lastName;
person["lastName"];
```

- 对象方法

对象的方法定义了一个函数，并作为对象的属性存储。对象方法通过添加 () 调用，可以使用以下语法创建对象方法：

```
methodName : function() {
    // 代码
}
```

使用以下语法访问对象方法：

```
objectName.methodName()
```

如访问上文定义的 robot 对象的 methodSum 方法，该方法返回值为 7。

```
robot. methodSum (3,4);
```

6.9.13.4.6 Math 对象

Math 对象的作用是执行普通的算数任务。Math 对象提供多种算数值类型和函数。使用之前无需对它进行定义。

使用 Math 的属性/方法的语法如下：

```
var x = Math.PI;
var y = Math.sqrt(16);
```

Math 对象属性

属性	描述
E	返回算术常量 e，即自然对数的底数（约等于 2.718）。
LN2	返回 2 的自然对数（约等于 0.693）。
LN10	返回 10 的自然对数（约等于 2.302）。
LOG2E	返回以 2 为底的 e 的对数（约等于 1.4426950408889634）。
LOG10E	返回以 10 为底的 e 的对数（约等于 0.434）。
PI	返回圆周率（约等于 3.14159）。
SQRT1_2	返回 2 的平方根的倒数（约等于 0.707）。
SQRT2	返回 2 的平方根（约等于 1.414）。

Math 对象方法

方法	描述
abs(x)	返回 x 的绝对值。
acos(x)	返回 x 的反余弦值。
asin(x)	返回 x 的反正弦值。
atan(x)	以介于 -PI/2 与 PI/2 弧度之间的数值来返回 x 的反正切值。
atan2(y,x)	返回从 x 轴到点 (x,y) 的角度（介于 -PI/2 与 PI/2 弧度之间）。
ceil(x)	对数进行上舍入。
cos(x)	返回数的余弦。
exp(x)	返回 Ex 的指数。
floor(x)	对 x 进行下舍入。
log(x)	返回数的自然对数（底为 e）。
max(x,y,z,...,n)	返回 x,y,z,...,n 中的最高值。
min(x,y,z,...,n)	返回 x,y,z,...,n 中的最低值。
pow(x,y)	返回 x 的 y 次幂。
random()	返回 0 ~ 1 之间的随机数。
round(x)	四舍五入。
sin(x)	返回数的正弦。
sqrt(x)	返回数的平方根。
tan(x)	返回角的正切。
tanh(x)	返回一个数的双曲正切函数值。
trunc(x)	将数字的小数部分去掉，只保留整数部分。

6.9.13.4.7 Date 对象

Date 对象用于处理日期与时间。

创建 Date 对象如下：

```
var d = new Date();
```

Date 对象属性

属性	描述
constructor	返回对创建此对象的 Date 函数的引用。
prototype	允许向对象添加属性和方法

Date 对象方法

方法	描述
getDate()	从 Date 对象返回一个月中的某一天 (1 ~ 31)。
getDay()	从 Date 对象返回一周中的某一天 (0 ~ 6)。
getFullYear()	从 Date 对象以四位数字返回年份。
getHours()	返回 Date 对象的小时 (0 ~ 23)。
getMilliseconds()	返回 Date 对象的毫秒(0 ~ 999)。
getMinutes()	返回 Date 对象的分钟 (0 ~ 59)。
getMonth()	从 Date 对象返回月份 (0 ~ 11)。
getSeconds()	返回 Date 对象的秒数 (0 ~ 59)。
getTime()	返回 1970 年 1 月 1 日至今的毫秒数。
getTimezoneOffset()	返回本地时间与格林威治标准时间 (GMT) 的分钟差。
getUTCDate()	根据世界时从 Date 对象返回月中的一天 (1 ~ 31)。
getUTCDay()	根据世界时从 Date 对象返回周中的一天 (0 ~ 6)。
getUTCFullYear()	根据世界时从 Date 对象返回四位数的年份。
getUTCHours()	根据世界时返回 Date 对象的小时 (0 ~ 23)。
getUTCMilliseconds()	根据世界时返回 Date 对象的毫秒(0 ~ 999)。
getUTCMinutes()	根据世界时返回 Date 对象的分钟 (0 ~ 59)。
getUTCMonth()	根据世界时从 Date 对象返回月份 (0 ~ 11)。
getUTCSeconds()	根据世界时返回 Date 对象的秒钟 (0 ~ 59)。
setDate()	设置 Date 对象中月的某一天 (1 ~ 31)。
setFullYear()	设置 Date 对象中的年份（四位数字）。
setHours()	设置 Date 对象中的小时 (0 ~ 23)。
setMilliseconds()	设置 Date 对象中的毫秒 (0 ~ 999)。

setMinutes()	设置 Date 对象中的分钟 (0 ~ 59)。
setMonth()	设置 Date 对象中月份 (0 ~ 11)。
setSeconds()	设置 Date 对象中的秒钟 (0 ~ 59)。
setTime()	setTime() 方法以毫秒设置 Date 对象。
setUTCDate()	根据世界时设置 Date 对象中月份的一天 (1 ~ 31)。
setUTCFullYear()	根据世界时设置 Date 对象中的年份（四位数字）。
setUTCHours()	根据世界时设置 Date 对象中的小时 (0 ~ 23)。
setUTCMilliseconds()	根据世界时设置 Date 对象中的毫秒 (0 ~ 999)。
setUTCMinutes()	根据世界时设置 Date 对象中的分钟 (0 ~ 59)。
setUTCMonth()	根据世界时设置 Date 对象中的月份 (0 ~ 11)。
setUTCSeconds()	setUTCSeconds() 方法用于根据世界时 (UTC) 设置指定时间的秒字段。
toDateString()	把 Date 对象的日期部分转换为字符串。
toJSON()	以 JSON 数据格式返回日期字符串。
toLocaleDateString()	根据本地时间格式，把 Date 对象的日期部分转换为字符串。
toLocaleTimeString()	根据本地时间格式，把 Date 对象的时间部分转换为字符串。
toLocaleString()	根据本地时间格式，把 Date 对象转换为字符串。
toString()	把 Date 对象转换为字符串。
toTimeString()	把 Date 对象的时间部分转换为字符串。
toUTCString()	根据世界时，把 Date 对象转换为字符串。
valueOf()	返回 Date 对象的原始值。

6.9.13.4.8 类型转换

- String()方法

String() 方法可以将数字、布尔、日期等类型转换为字符串。实例如下：

```
String(123);       // 将数字 123 转换为字符串并返回
```

```
String(false);        // 返回 "false"
String(new Date()) ;     // 返回 Mon Dec 25 2023
```

Number 方法 toString() 也能实现同样的效果。

```
(123).toString();
false.toString();    // 返回 "false"
obj = new Date();
obj.toString() ;  // 返回 Mon Dec 25 2023
```

- Number()方法

Number() 方法可以将字符串、布尔、日期转换为数字。其中空字符串转换为 0，非数字字符串会转换为 NaN (不是数字)。实例如下：

```
Number("3.14");   // 返回 3.14
Number(" ");      // 返回 0
Number("99 ab");  // 返回 NaN
Number(true);     // 返回 1
d = new Date();
Number(d);        // 返回 1703483796513
```

6.9.13.5 函数

函数是包裹在大括号中的代码块，前面使用了关键词 function，当调用该函数时，会执行函数内的代码。函数内部声明的变量是局部变量，只能在函数内部访问它。

```
function myFunction ()
{
    // 执行代码
}
```

函数也可以带参数，参数由逗号 (,) 分隔，在调用带参数的函数时，可以向其传递值，这些值可以在函数中使用。

```
function myFunction(var1,var2)
{
```

// 执行代码

}

函数也可以带返回值，使用 return 语句。

```
function myFunction(a,b)
{
    return a*b;
}
```

6.9.13.6 运算符

6.9.13.6.1 算术运算

算术运算符用于给变量进行四则运算。

运算符	描述	举例	运算结果
+	加法	x = y +3,式中 y 值为 5	8
-	减法	x = y-3,式中 y 值为 5	2
*	乘法	x = y*2,式中 y 值为 5	10
/	除法	x = y/2,式中 y 值为 5	2.5
%	取模（余数）	x = y%2,式中 y 值为 5	1
++	自增	x = y++,式中 y 值为 5 x = ++y,式中 y 值为 5	5 6
--	自减	x = y--,式中 y 值为 5 x = --y,式中 y 值为 5	5 4

6.9.13.6.2 字符串加法

+ 运算符可用于把文本值或字符串变量加起来（连接起来）。

如需把两个或多个字符串变量连接起来，请使用 + 运算符，如下实例：

```
txt1 = "How are you ";
```

txt2 = "Today";

txt3 = txt1 + txt2;

txt3 运算结果为"How are you Today"。

如果数字和字符串相加，将返回字符串，如下实例：

x = "5"+5;

y = "Hio"+5;

x 输出 55，y 输出 Hio5。

6.9.13.6.3 赋值运算

赋值运算符用于给变量赋值。

运算符	描述	举例	运算结果
=	等于	x = y,式中 y 值为 5	5
+=	加等	x += y,式中 y 值为 5,x 初始值为 5	10
-=	减等	x -= y,式中 y 值为 5,x 初始值为 5	0
*=	乘等	x *= y,式中 y 值为 5,x 初始值为 5	25
/=	除等	x /= y,式中 y 值为 5,x 初始值为 5	1
%=	模等	x %= y,式中 y 值为 5,x 初始值为 5	0

6.9.13.6.4 比较运算

比较运算符在逻辑语句中使用，以测定变量或值是否相等。

运算符	描述	举例	运算结果
==	等于	x == 1,式中 x 初始值为 5	false
!=	不等于	x != 1,式中 x 初始值为 5	true
>	大于	x > 1,式中 x 初始值为 5	true
<	小于	x < 1,式中 x 初始值为 5	false
>=	大于或等于	x >= 5,式中 x 初始值为 5	true
<=	小于或等于	x <= 5,式中 x 初始值为 5	true

6.9.13.6.5 逻辑运算符

逻辑运算符用于测定变量或值之间的逻辑。

给定 x = 5、y = 2，下表解释了逻辑运算符：

运算符	描述	举例	运算结果
&&	and	(x > 1 && y<10)	true
||	or	(x == 5 || y == 7)	true
!	not	!(x == y)	true

6.9.13.6.6 条件运算符

脚本包含基于某些条件对变量进行赋值的条件运算符。语法如下：

Variablename = (condition)?value1:value2 ;

如果 condition 条件为真，输出 value1，否则，输出 value2。

ang=(theta<30)?"角度正常":"角度过大";

6.9.13.6.7 条件语句

条件语句用于基于不同的条件来执行不同的动作。

- if 语句

只有当指定条件为 true 时，该语句才会执行代码。语法如下：

```
if (condition)
{
  //当条件为 true 时执行的代码;
  }
```

- if...else 语句

当条件为 true 时执行代码，当条件为 false 时执行其他代码。语法如下：

```
if (condition)
{
```

```
   //当条件为 true 时执行的代码;
}
else
{
  //当条件不为 true 时执行的代码;
}
```

- if...else if....else 语句

使用该语句来选择多个代码块之一来执行。语法如下：

```
if (condition1)
{
   //当条件 1 为 true 时执行的代码;
}
else if (condition2)
{
    // 当条件 2 为 true 时执行的代码;
}
else
{
   //当条件 1 和 条件 2 都不为 true 时执行的代码;
}
```

6.9.13.6.8 switch 语句

switch 语句用于基于不同的条件来执行不同的动作。语法如下：

```
switch(n)
{
    case 1:
        //执行代码块 1;
        break;
```

```
    case 2:
      //执行代码块 2;
      break;
    default:
      //与 case 1 和 case 2 不同时执行的代码;
}
```

6.9.13.6.9 循环

- for 循环

  for 循环可以将代码块执行指定的次数。语法如下：

```
for (语句 1; 语句 2; 语句 3)
{
  //被执行的代码块;
}
```

语句 1:（代码块）开始前执行。

语句 2:定义运行循环（代码块）的条件。

语句 3: 在循环（代码块）已被执行之后执行。

- while 循环

  while 只要指定条件为 true，循环就可以一直执行代码块。语法如下：

```
while (条件)
{
  //需要执行的代码;
}
```

- do/while 循环

  do/while 循环会在检查条件是否为真之前执行一次代码块，然后如果条件为真的话，就会重复这个循环。语法如下：

```
do
{
```

```
  //需要执行的代码;
}
while (条件);
```

6.9.13.6.10 中断语句

- break 语句

  break 语句用于跳出循环。break 语句跳出循环后，会继续执行该循环之后的代码。如下实例，当 i 等于 6 时，将退出 for 循环。

```
for (i=0;i<10;i++)
{
    if (i==6)
    {
        break;
    }
    x= "The number is " + i;
}
```

- continue 语句

  continue 语句中断当前的循环中的迭代，然后继续循环下一个迭代

6.9.13.7 正则表达式

正则表达式（regex）使用单个字符串来描述、匹配一系列符合某个句法规则的字符串搜索模式。搜索模式可用于文本搜索和文本替换。语法如下：

/正则表达式主体/修饰符(可选)

实例：

```
var a= /Lead/i
```

实例解析：

Lead 是一个正则表达式主体 (用于检索)。

i 是一个修饰符 (搜索不区分大小写)。

- 正则表达式修饰符

修饰符	描述
i	执行对大小写不敏感的匹配。
g	执行全局匹配（查找所有匹配而非在找到第一个匹配后停止）。
m	执行多行匹配。

- 正则表达式模式

方括号用于查找某个范围内的字符。

表达式	描述
[abc]	查找方括号之间的任何字符。
[0-9]	查找任何从 0 至 9 的数字。
(x|y)	查找任何以 | 分隔的选项。

- 元字符

元字符是拥有特殊含义的字符。

元字符	描述
\d	查找数字。
\s	查找空白字符。
\b	匹配单词边界。
\uxxxx	查找以十六进制数 xxxx 规定的 Unicode 字符。

- 量词

元字符	描述
n+	匹配任何包含至少一个 *n* 的字符串。
n*	匹配任何包含零个或多个 *n* 的字符串。
n?	匹配任何包含零个或一个 *n* 的字符串。

在脚本中，正则表达式通常用于两个字符串方法 : search() 和 replace()。

search() 方法用于检索字符串中指定的子字符串，或检索与正则表达式相匹配的子字符串，并返回子串的起始位置。

replace() 方法用于在字符串中用一些字符串替换另一些字符串，或替换一个与正则表达式匹配的子串。

实例：

```
var str = "Welcom to LeadShine!";
var n = str.search(/lead/i);       // 返回 10
var txt = str.replace(/leadshine /i,"LS");//返回 Welcom to LS!
```

- RegExp 对象

RegExp 对象是一个预定义了属性和方法的正则表达式对象。语法如下：

```
var pat=new RegExp(pattern,modifiers);
```

等价于 var pat = /pattern/modifiers;

其中 pattern（模式）描述表达式的字符串。

modifiers(修饰符) 用于指定全局匹配、区分大小写的匹配和多行匹配的字符。

RegExp 对象的方法

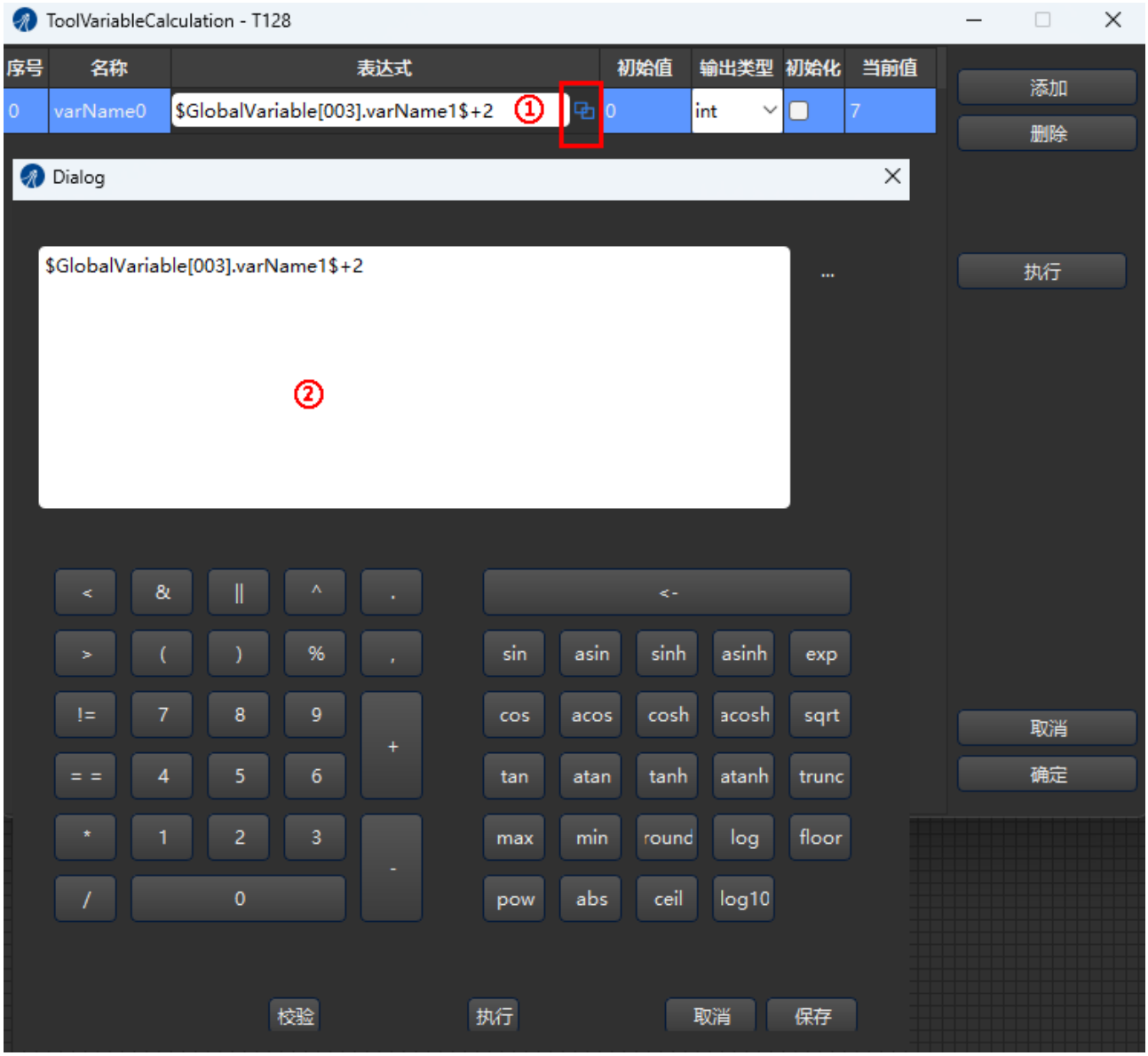
方法	描述
exec	检索字符串中指定的值。返回找到的值，并确定其位置。
test	检索字符串中指定的值。返回 true 或 false。
toString	返回正则表达式的字符串。

实例：

```
var str = 'cctv';
var patt1 = new RegExp('v', 'i'); /
patt1.test(str) //输出 true
var patt2=new RegExp("e");
print(patt2.exec("The best things in life are free")); //输出 e
```

6.9.14 变量计算

通过设计表达式，完成变量之间的算术或逻辑运算。该工具本质是一个计算器，相对于脚本工具块，使用更简单。



6.9.15 信息提示

输出信息到用户界面的“信息打印”和“报警统计”控件。



参数说明如下：

提示/警告/错误：以不同颜色的文本输出信息到信息打印控件。

是否弹窗：勾选后，将同步弹窗提示。

是否记录到报警页面：勾选后，将信息同步输出到报警统计控件。

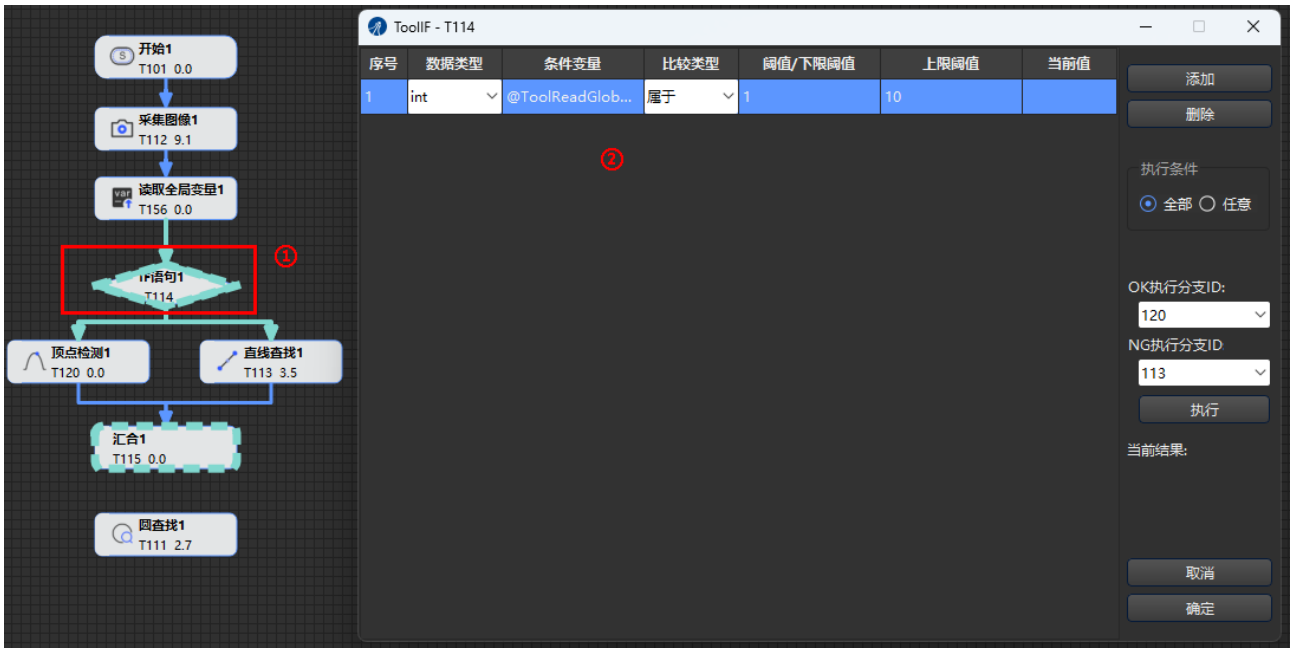
是否增加 NO：勾选后，弹窗对话框存在 YES 与 NO 的选择，当点击 YES 时，该工具块输出 True 状态；点击 NO 时该工具块输出 False 状态，供后续工具判断使用。

6.10 逻辑控制

6.10.1 IF 语句

IF 语句即 If_Else 分支，需与汇合节点一起使用，If_Else 分支块激活后，判断条件成立，激活 OK 分支的低位的工具块，否则执行 NG 分支的低位的工具块。

双击①IF 语句可进入②属性配置窗口。如下图所示。

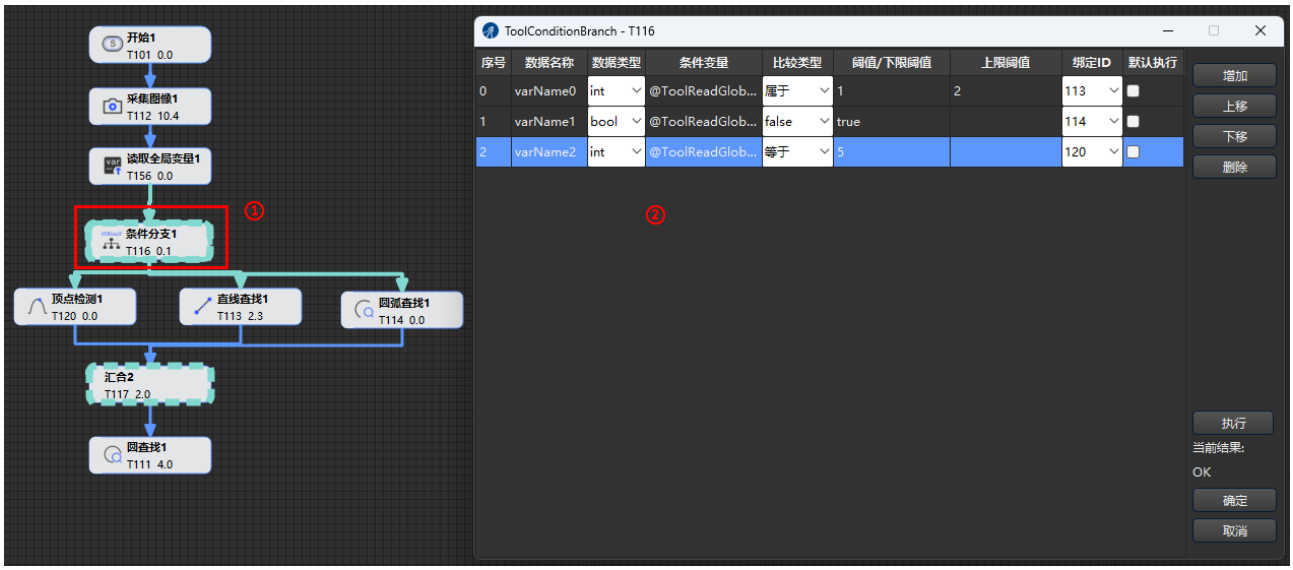


- 数据类型：比较数据类型，支持布尔、整型、浮点型和字符串。
- 条件变量：订阅比较数据源。
- 比较类型：条件变量与阈值的比较方式，支持的比较方式为属于、不属于、大于、小于、等于、大于等于、小于等于、不等于。
- 阈值/下限阈值：条件变量的阈值或下限值。
- 上限阈值：条件变量的上限值。
- 执行条件：全部，所有添加的条件进行与运算；任意，所有添加的条件进行或运算。
- OK/NG 执行分支 ID：绑定后续执行的分支，连线后可绑定。

6.10.2 条件分支

条件分支与汇合节点一起使用，条件分支激活后，当订阅的条件符合要求时执行设定的工具块。

条件分支与汇合节点一起使用，所有的选择分支都必须连接到汇合节点，条件分支块激活后，根据设定的判断条件（int/bool/float/sting 类型），激活条件成立的低位的工具块，可以一次执行多个分支，可配置 default 分支，当条件都不满足时，执行 default 分支，如果无 default 分支，汇合后的工具不执行，直接返回开始块。



- 数据名称：用户设定变量名称。
- 数据类型：比较数据类型，支持布尔、整型、浮点型和字符串。
- 条件变量：订阅比较数据源。
- 比较类型：条件变量与阈值的比较方式，支持的比较方式为属于、不属于、大于、小于、等于、大于等于、小于等于、不等于。
- 阈值/下限阈值：条件变量的阈值或下限值。
- 上限阈值：条件变量的上限值。
- 绑定 ID：绑定后续执行的工具 ID 号。
- 默认执行：default 分支，不设置 default 分支，条件都不成立时，汇合后的后续工具不执行。

6.10.3 选择分支

选择分支与汇合节点一起使用，所有的选择分支都必须连接到汇合节点，选择分支块激活后，根据设定的判断条件（int 类型），激活条件成立的低位的工具块，可以一次执行多个分支，可配置 default 分支，当条件都不满足时，执行 default 分支，如果无 default 分支，汇合后的工具不执行，直接返回开始块。

双击①选择分支可进入②属性配置窗口。如下图所示。

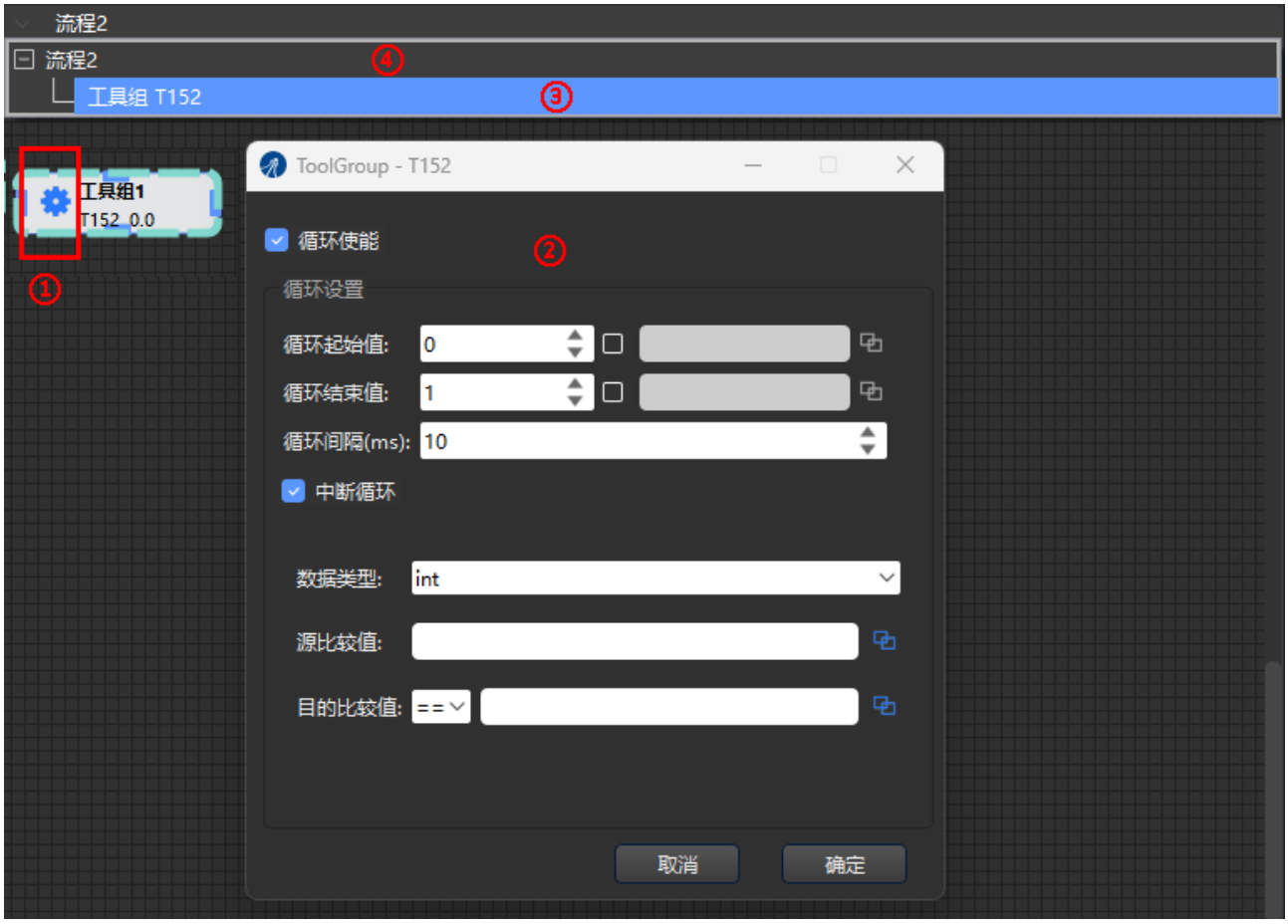


- 分支值：订阅一个用于控制分支的整型数据。通过判断前序工具块的输入值，确认该值是否有对应的分支条件。
- 工具名称：执行的工具号，连线后自动生成。
- 条件变量：执行分支的条件值。
- 默认执行：default 分支。不设置 default 分支，条件都不成立时，汇合后的后续工具不执行。

6.10.4 工具组

在复杂项目中工具过多会导致查看不够直观，此时可利用工具组进行折叠整合，同时工具组也支持循环功能。

单击工具组①设置图标可进入工具组②设置窗口，双击工具组或点击③下拉工具组即可进入工具组内部，在该界面中，可继续拖动工具块搭建流程，搭建完成后可单击④返回工具组外部。如下图所示。



工具组的参数说明如下：

- 循环使能：开启循环使能后，在工具组内订阅参数时，可嵌套订阅循环索引，用于表示工具组循环执行的次数。
- 循环起始值：自定义循环计数起始值，一般设置为 0。
- 循环结束值：需订阅相关参数值，循环结束值与循环起始值之差即为循环次数。
- 循环间隔(ms)：单次循环的间隔时间。
- 中断循环：开启后，需完成相关参数设置。当满足条件时，工具组内循环终止。

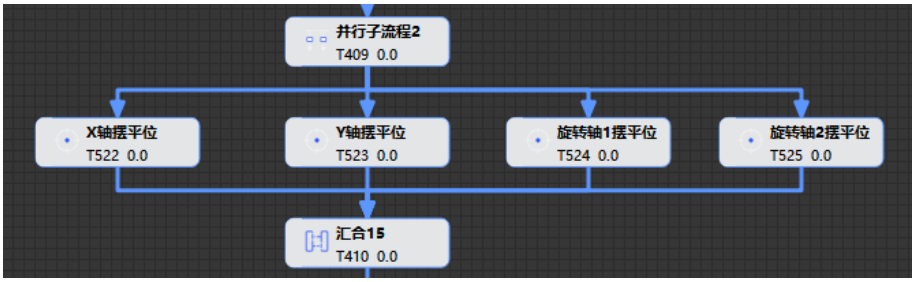
  数据类型：选择中断循环的数据类型，包括 int、float 和 string 类型。

  源比较值：可自定义或订阅源比较值。

  目的比较值：目的比较值和源比较值之间进行比对，满足条件时即达到循环中断要求。

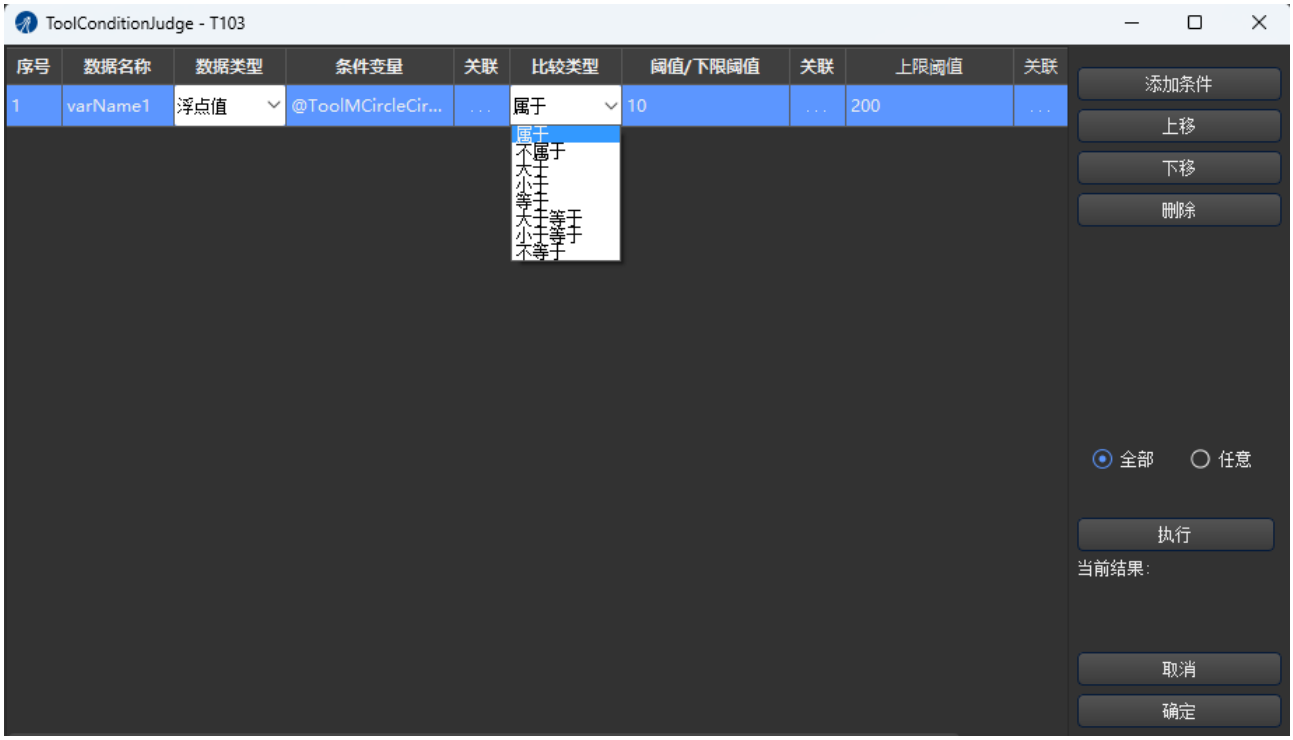
6.10.5 并行子流程

并行子流程与汇合节点一起使用，并行子流程块激活后，同时激活所连接的低位的子流程块。并行子流程允许嵌套多层，同一层的分支允许汇合，不允许跨层直接汇合。所有子流程分支块结束后，激活低位的程序块。该工具块无参数配置。



6.10.6 条件判断

根据设置的多个条件，进行与/或运算后，输出综合结果 TRUE/FALSE。

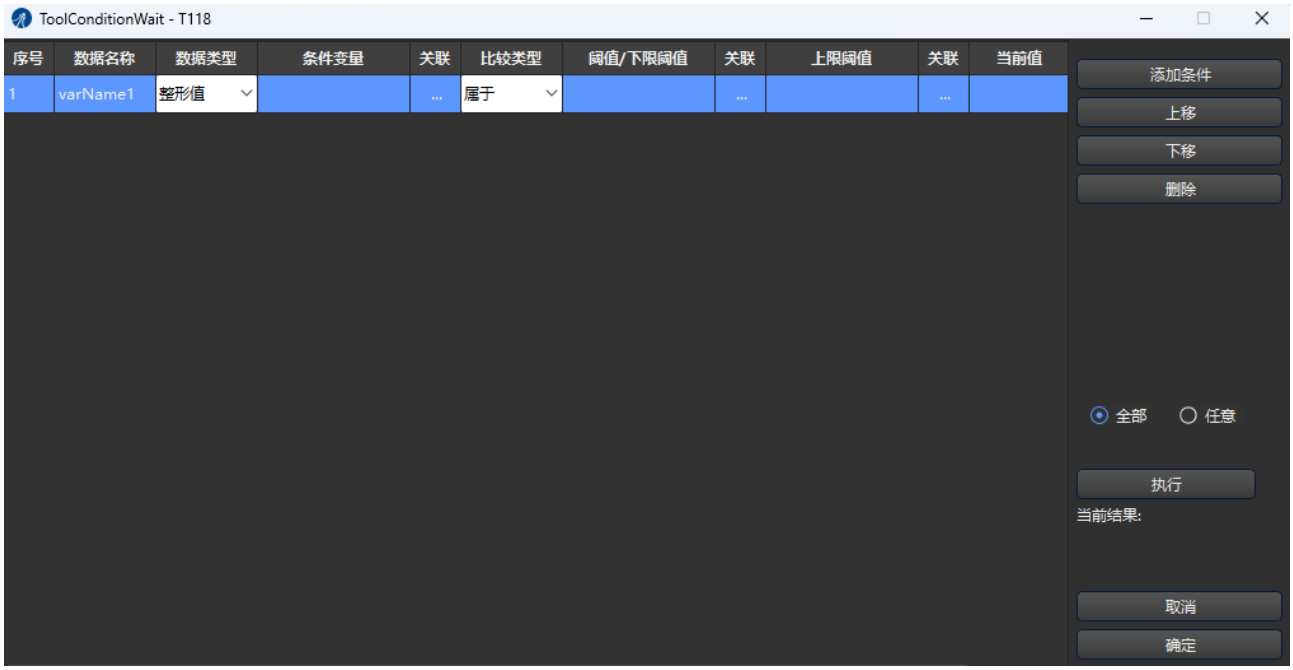


- 数据类型：包括浮点数、整型、布尔、字符串四种类型。
- 条件变量：关联需作为判断条件的数据来源，但变量类型需和**数据类型**对应。
- 比较类型：包括属于、不属于、大于、小于、等于、大于等于、小于等于、不等于。

- 阈值/下限阈值：关联**条件变量**的下限阈值。符合**比较类型**范围内的判定为 OK，否则判定为 NG。
- 上限阈值：关联**条件变量**的上限阈值。符合**比较类型**范围内的判定为 OK，否则判定为 NG。
- 判断方式：①全部，对判断条件中所有判断结果进行逻辑与运算；②任意，对判断条件中所有判断结果进行逻辑或运算。

6.10.7 条件等待

根据设置的多个条件，进行与/或运算后，工具循环计算结果，直到结果为 true 时，否则流程卡在当前工具。参数设置请参见条件判断。

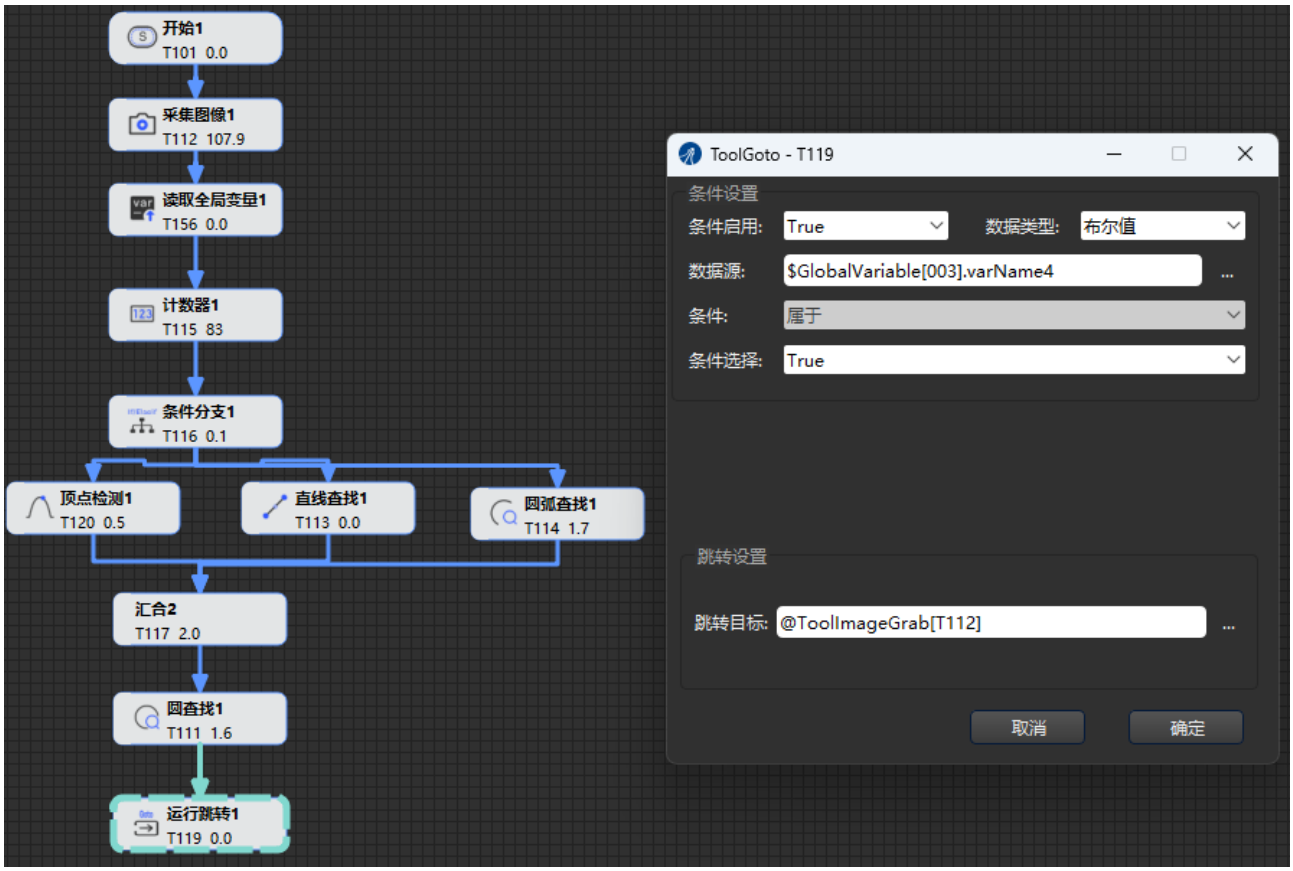


6.10.8 运行跳转

无条件或满足某一个条件时，执行跳转目标。跳转规则要求如下：

（1） 不允许跨线程跳转，只能跳转到和当前跳转工具块同一线程的工具块；

（2）可以跳出分支，不允许跳入分支。即不能跳进像条件判断、IF 语句等（有分叉有汇合的工具块）的分支里面。



➢ 条件设置

根据设置的条件，决定跳转功能是否执行。默认不启用，直接执行跳转块。当条件启用设置为 True 时，需配置执行条件，根据所选数据类型，订阅数据源，数据源满足设定的条件后，如大于、小于、等于、不等于、属于、不属于等，再执行跳转目标。

➢ 跳转设置

跳转目标：跳转的目标执行块。

6.10.9 流程中断

中断当前的任务流程，中断实现的是暂停和暂停重置当前的流程，通过勾选重置流程决定且支持弹窗提示功能。



6.11 通信

6.11.1 TCP 通信发送

可将数据发送到 TCP 通信设备。通信设备在项目目录树中的“通信设备”中添加，详见[添加通信设备]章节。



在“选择通信设备”中选取发送数据的通信设备。在“数据发送组装表”中添加要发送的数据类型并绑定要发送的变量，通过分割符组装要发送的数据，并可设置组装后的数据是否添加结束符。工具执行时，“组装后发送的数据”栏将显示当前发送的数据，数据仅在当前窗口打开时刷新显示。

6.11.2 TCP 通信接收

TCP 通信接收可从通信设备中获取数据。通信设备在项目目录树中的“通信设备”中添加，详见添加通信设备章节。



在“选择通信设备”中选取接收数据的通信设备。需在“数据解析表”中添加接收数据的变量及类型，通过分隔符解析接收到的数据。在“接收模式选择”中配置是否等待数据或匹配等待数据。当选择不等待时，工具立即执行；选择无数据等待时，工具将阻塞在此处，

直到收到数据才继续执行流程中的后续工具；选择不匹配等待时，可在数据解析表中勾选需要匹配的数据值，收到的数据与匹配的数据值相等时才会继续执行流程中的后续工具。

6.11.3 Modbus 写入

可将数据发送到 Modbus 通信设备。通信设备在项目目录树中的“通信设备”中添加，详见添加通信设备章节。本开发平台支持作为主站，使用 Modbus 协议，包括 ModbusTCP 和 ModbusRTU，将数据写入从站的寄存器。



- 起始地址：功能码对应的寄存器区的首地址。
- 功能类型：modbus 功能码，支持读线圈、读输入、读保持寄存器、读输入寄存器。
- 数据类型：指定写入的单个数据类型长度
- 大小端格式：数据高低字节对齐方式。

在“选择通信设备”中选取发送数据的 Modbus 通信设备。在“数据发送组装表”中添加要发送的数据类型并绑定要发送的变量（也可在数据值列输入固定值发送）。工具执行时，当前值列将显示本次发送的数据。

6.11.4 Modbus 读取

可从 Modbus 通信设备中获取数据。通信设备在项目目录树中的“通信设备”中添加，详见添加通信设备章节。本开发平台支持作为主站，使用 Modbus 协议，包括 ModbusTCP 和 ModbusRTU，接收从站寄存器的数据。



- 起始地址：功能码对应的寄存器区的首地址。
- 功能类型：modbus 功能码，支持读线圈、读输入、读保持寄存器、读输入寄存器。
- 数据类型：指定读取的单个数据类型长度
- 大小端格式：数据高低字节对齐方式。

在“选择通信设备”中选取接收数据的 Modbus 通信设备。需在“数据解析表中”添加接收数据的变量名称及类型。在“读取模式选择”中配置是否等待数据或匹配等待数据。当选择不等待时，工具立即执行；选择无数据等待时，工具将阻塞在此处，直到收到数据才继续

执行流程中的后续工具；选择不匹配等待时，可在数据解析表中的匹配值列勾选需要匹配的数据值，收到的数据与匹配的数据值相等时才会继续执行流程中的后续工具。

6.11.5 串口发送

可将数据发送到串口通信设备。通信设备在项目目录树中的“通信设备”中添加，详见添加通信设备章节。使用方式请参见 TCP 通信发送。



6.11.6 串口接收

串口接收可从串口通信设备中获取数据。通信设备在项目目录树中的“通信设备”中添加，详见添加通信设备章节。使用方式请参见 TCP 通信接收。



7. 案例搭建

7.1 上下料定位抓取

7.1.1 工艺流程

（1）检测皮带线上产品到位，控制皮带线停止，触发相机采集。

（2）采集完成后使用多特征定位工具块获取四边形中心及角度，并转换成世界坐标。

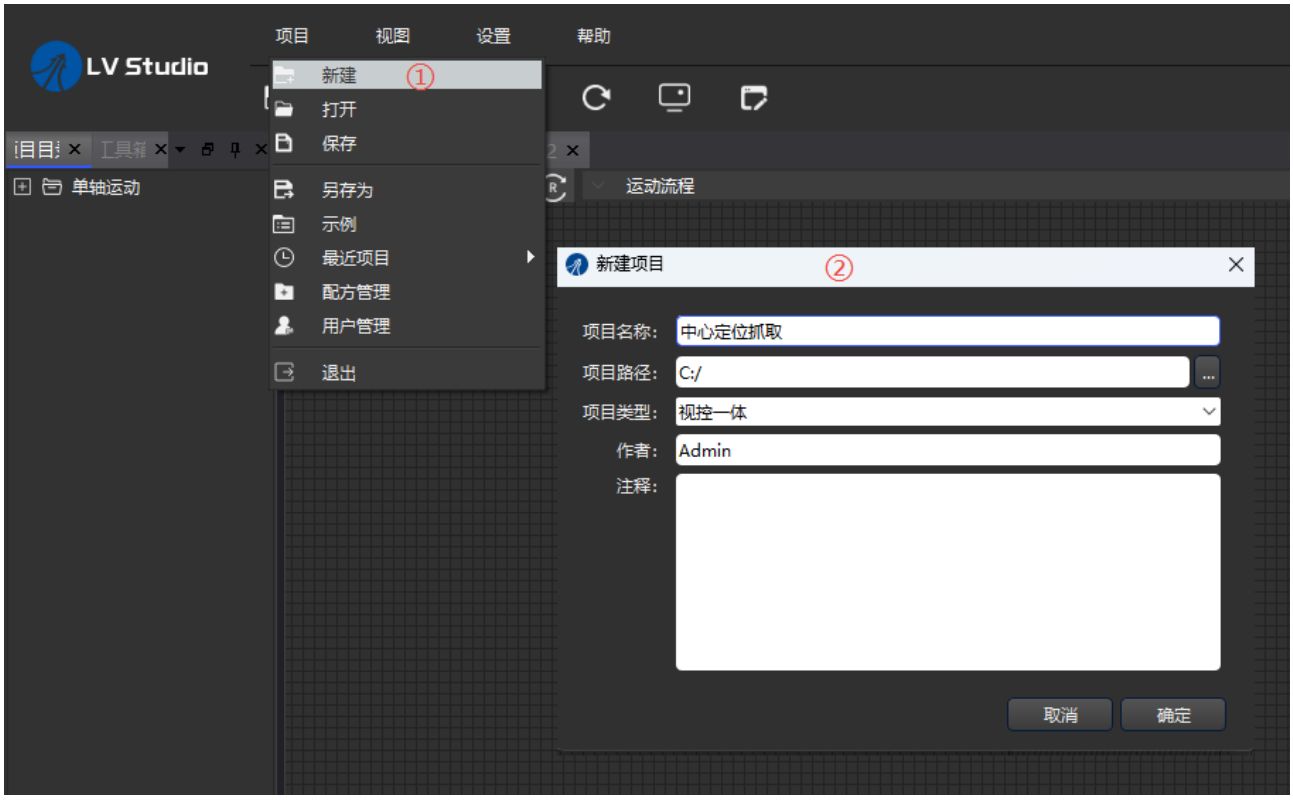
（3）XYZR 平台将产品抓取后，摆正放入加工台。

7.1.2 实现过程

本项目使用的视觉算法是多特征定位工具块的模板匹配+四条直线定位模式，模板匹配进行初定位，四条直线进行精定位，检测的四条直线构成的四边形对角线的中心点作为定位抓取点，其中一条直线的角度作为摆正角度值。

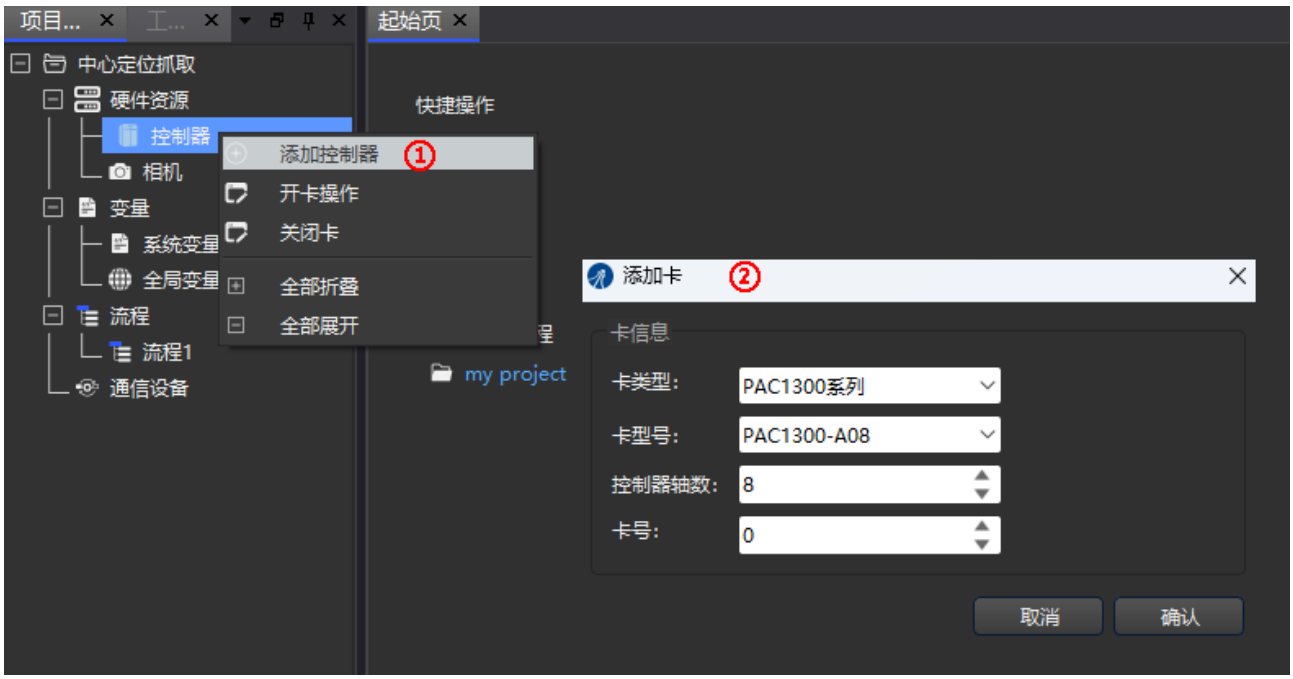
7.1.2.1 新建项目

选择菜单栏【项目】->【新建】，弹窗新建项目窗口，输入项目名称为中心定位抓取，项目类型为视控一体，完成项目创建。



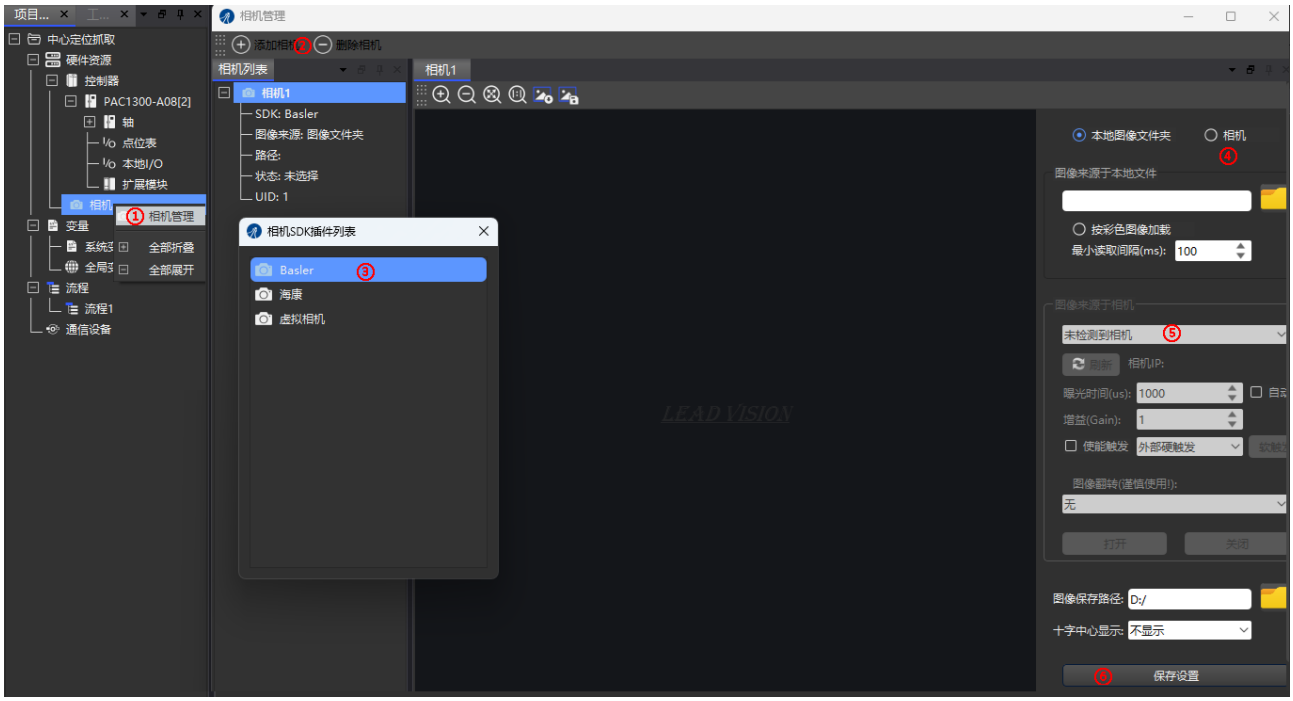
7.1.2.2 添加控制器及相机

按实际硬件及卡号拨码，添加一台 8 轴版本的 PAC 控制器 PAC1300-A08。



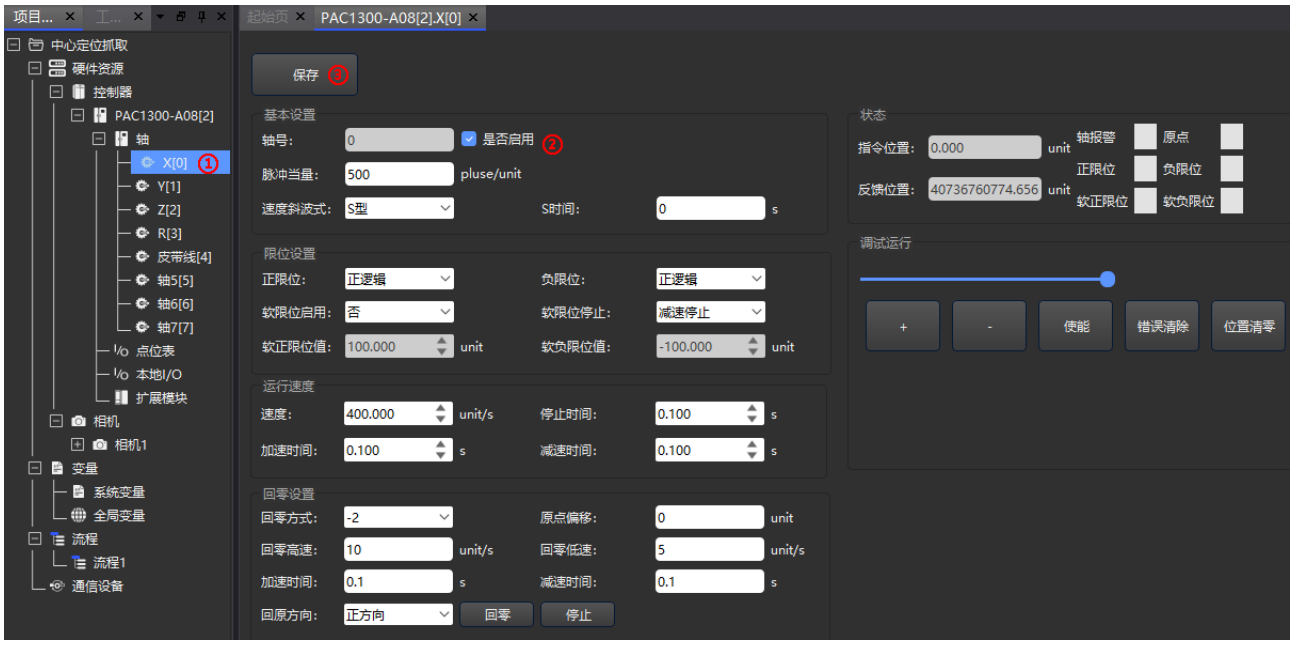
按实际硬件，添加一台 Basler 相机。本例图像来源为本地图像文件夹（如果实际连接了相机，此处勾选相机及相机触发模式）；图像来源路径中输入相对路径（./samples/images），当然也可以通过打开文件对话框，选择图片存放的绝对路径；其他参数按默认配置即可，最后单击“保存设置”，如下图所示。

注意：图像来源于路径为相对路径时，输入完路径后，需按回车生效。

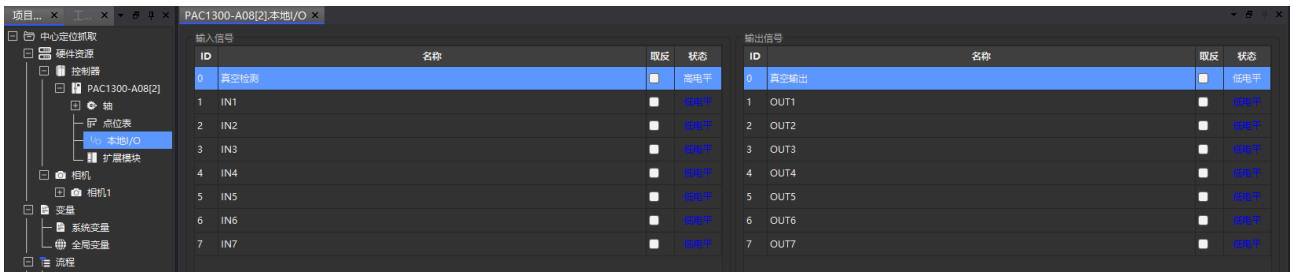


7.1.2.3 配置轴参数及 IO

点击【轴】展开轴列表，双击【轴 0】，弹出轴 0 参数配置窗口，根据实际设备配置，配置完成后点击保存，并修改名称，依次设置其他各轴参数，未使用的轴需取消【是否启用】勾选。



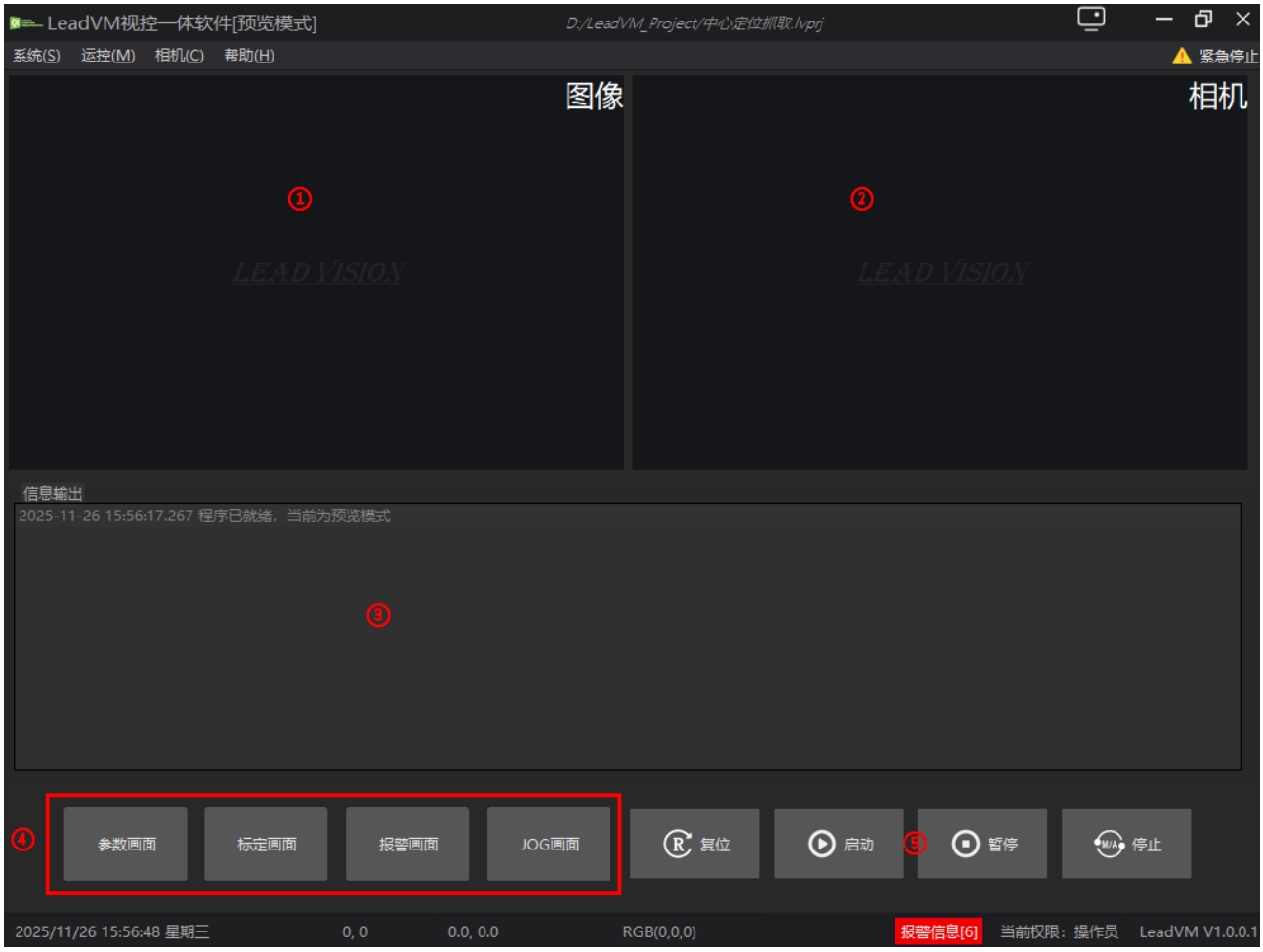
IO 配置如下。



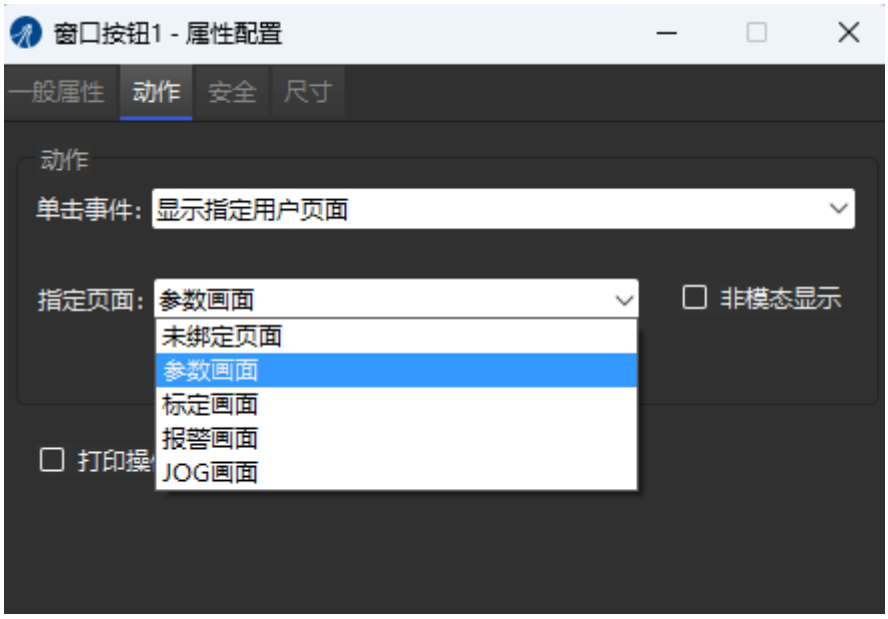
7.1.2.4 组态画面

单击 LV Studio 开发平台主界面工具条中的可进入用户界面组态窗口，在用户界面组态 5 个画面：主画面、参数画面、标定画面、报警画面、JOG 画面。

主画面如下图所示，所使用的控件①图像窗口，②相机窗口，③信息打印，④窗口按钮，⑤启停复位按钮。



其中④窗口按钮的事件动作为【显示指定用户页面】，分别配置 4 个按钮的页面为参数画面、标定画面、报警画面、JOG 画面。如下图所示。



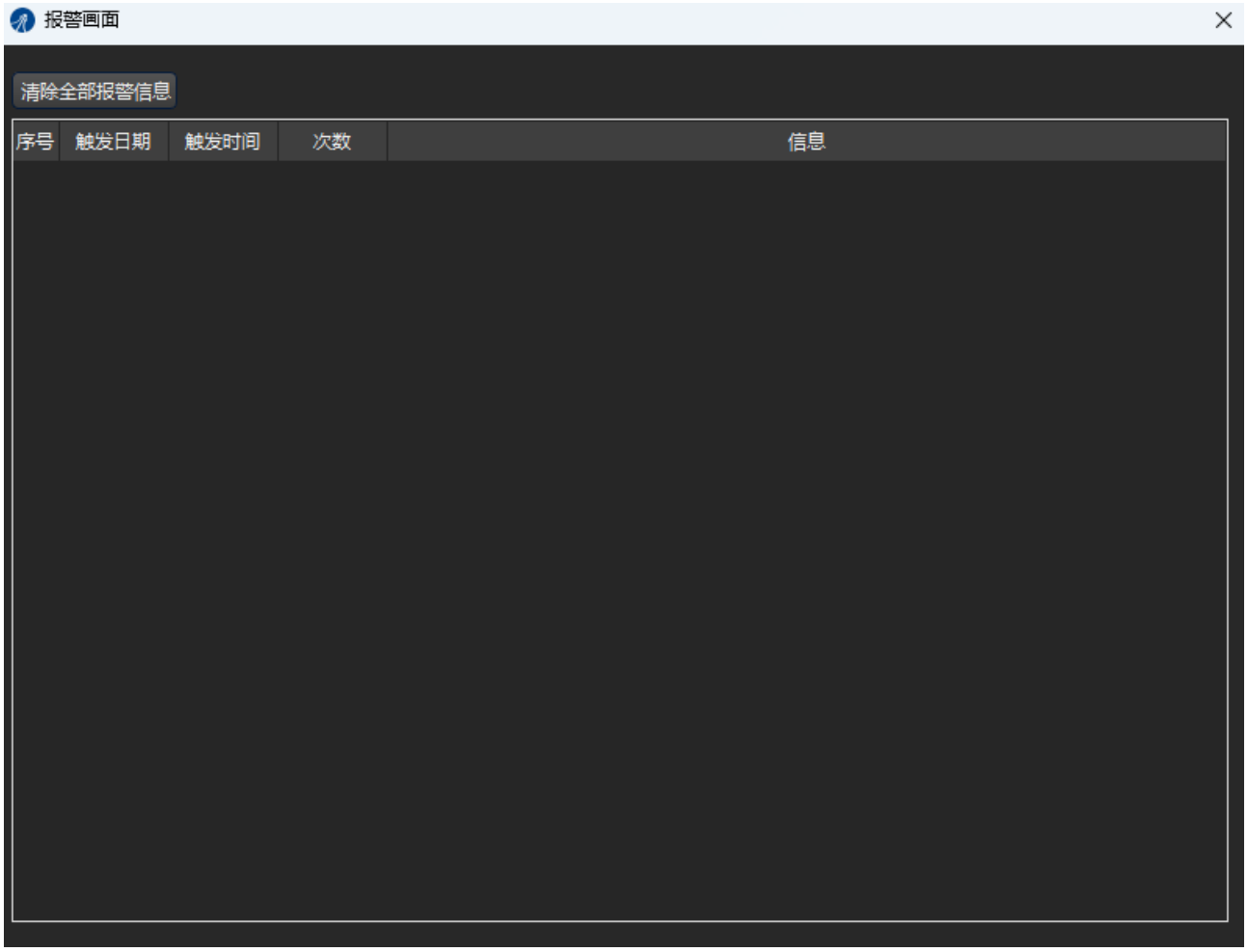
参数画面如下图所示，所使用的控件为文本框、数值框和组合框， 所有数值框的模式设置为“输入”，即用于输入数值。



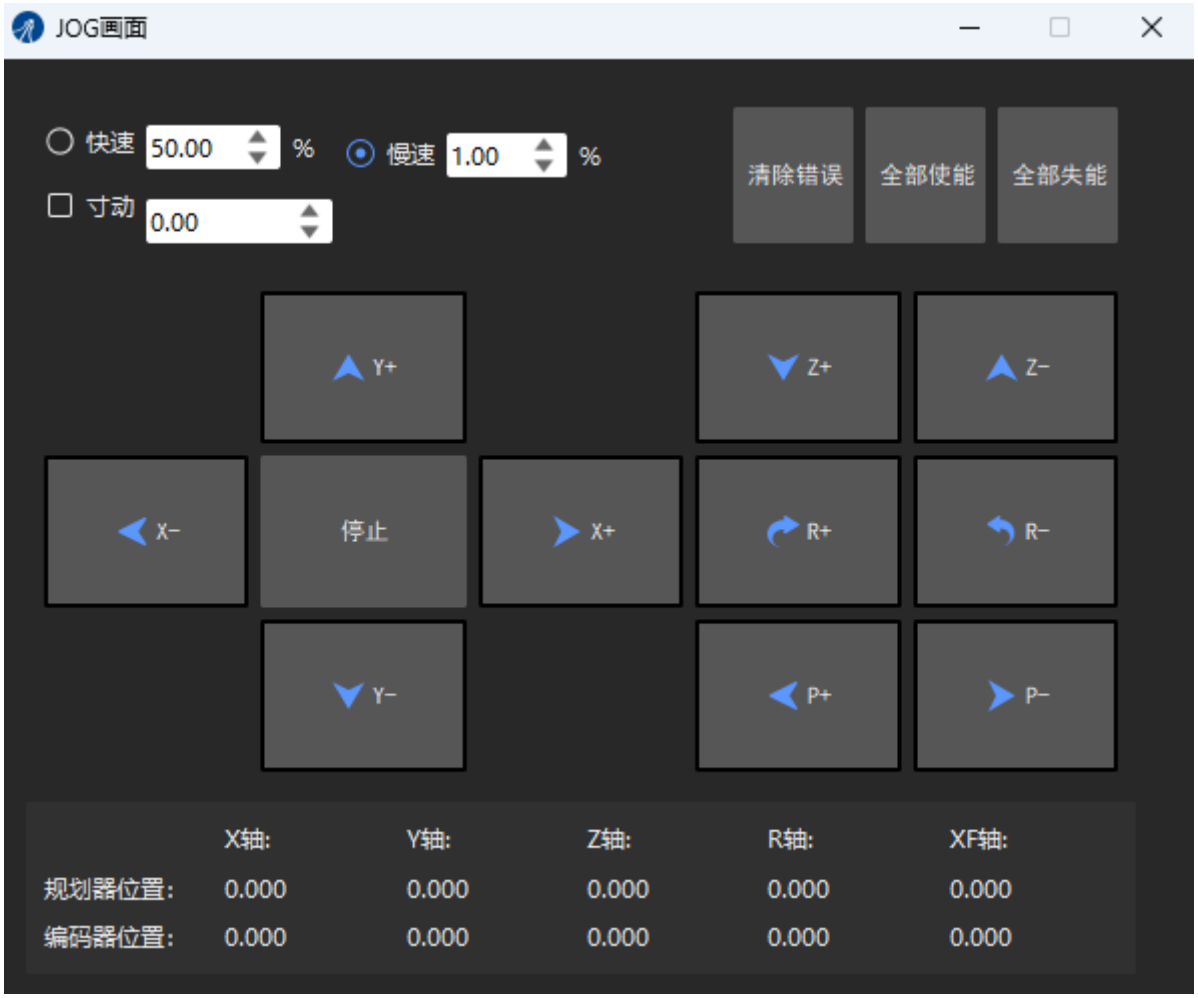
标定画面如下图所示，所使用的控件①/②文本和数值框（输入模式），③位置记录，④位按钮，⑤窗口按钮，⑥事件按钮，⑦窗口按钮。


标定画面
标定参数
X偏移
10. 000
Y偏移
10. 000
R角度偏移
5. 000
①
标定中心点位置
X
50.000
Y
50.000
Z
0.000
R
30.000
③
移动
记录
标定速度
X轴速度
100. 000
Y轴速度
100. 000
②
Z轴速度
50. 000
R轴速度
15. 000
标定步骤
开真空
④
模板设定
⑤
开始标定
⑥
结果查看
⑦

报警画面如下图所示，所使用的控件为报警统计，需配置最大报警数量，如 1000 条。

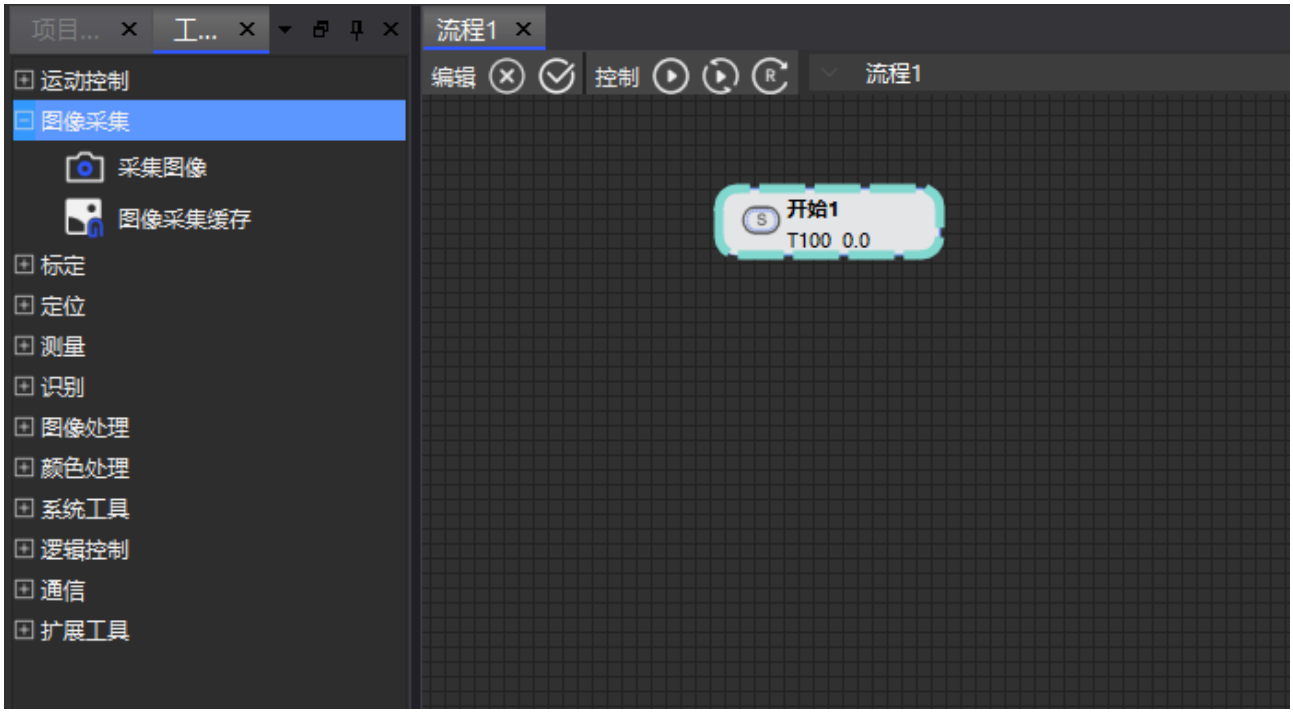


JOG 画面如下图所示，所使用的控件为五轴 JOG 运动，只需订阅各轴轴号及输入各轴名称即可。

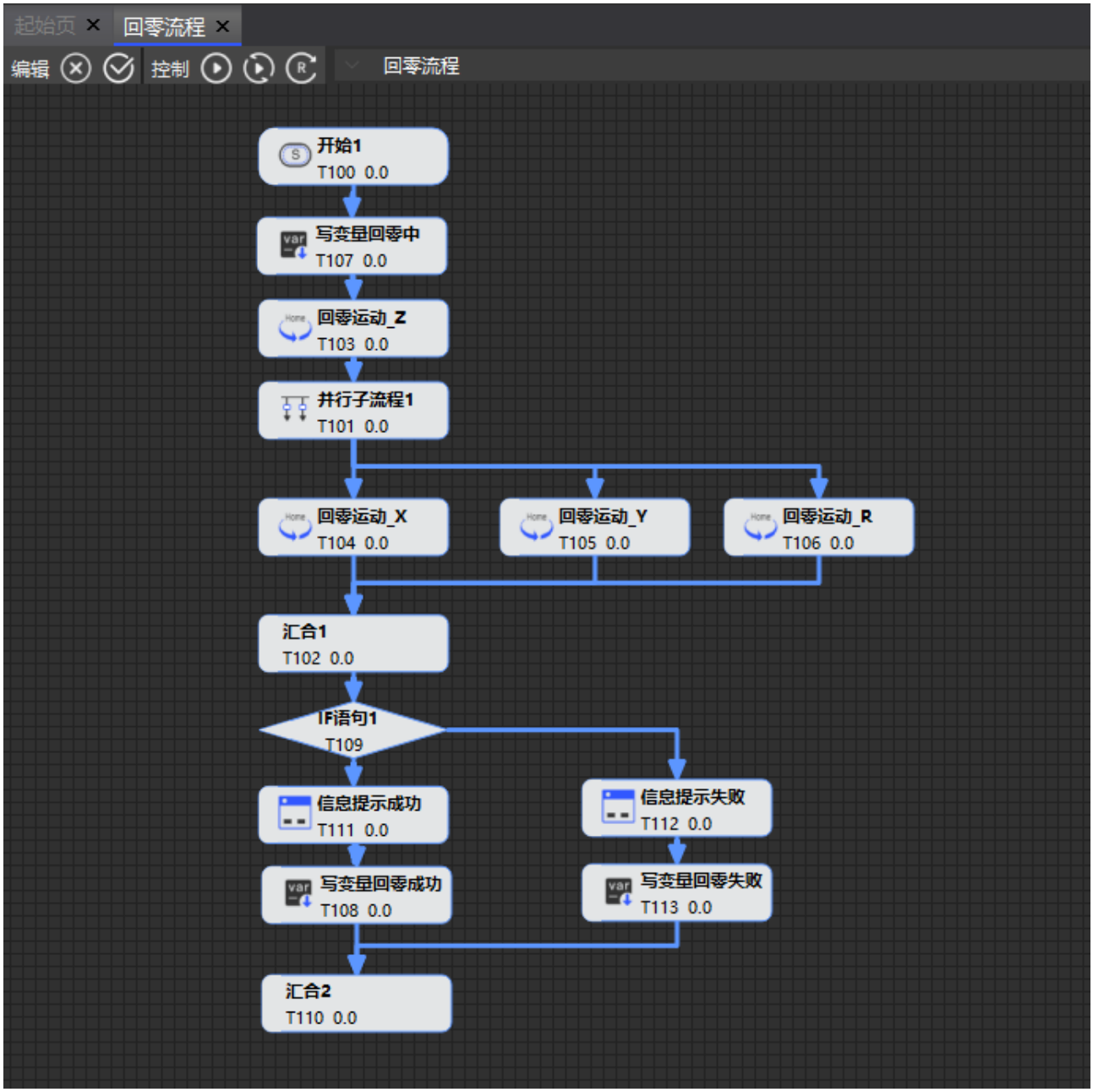


7.1.2.5 编写流程

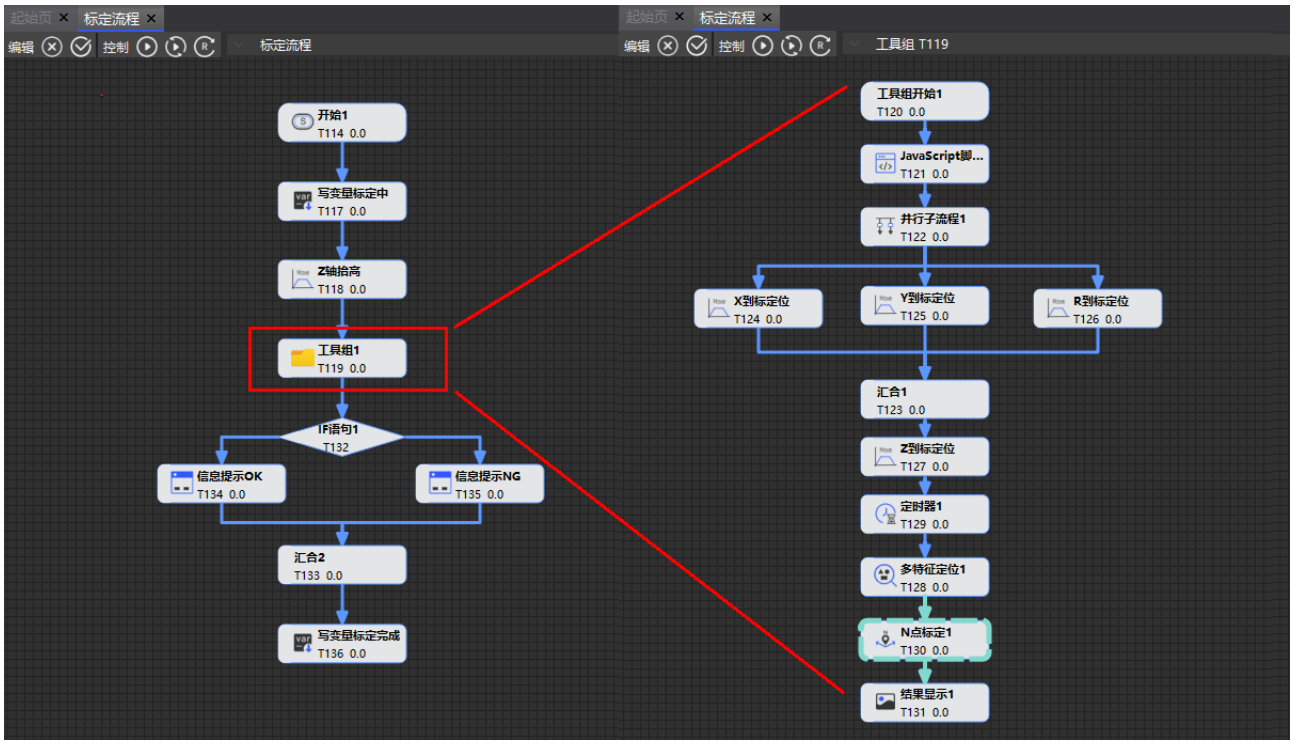
（1）双击项目目录树中的【流程 1】，打开【流程 1】窗口。点击【工具箱】将【项目树】切换到【工具箱】树，从【工具箱】中拖拽工具块到【流程 1】窗口，开始【流程 1】编写。编写完成后，修改【流程 1】的名称为回零流程。本例共建 3 个流程，分别为回零流程、定位流程、标定流程，并将回零流程和标定流程设置为触发执行，将标定流程设置为循环执行。流程模式配置请参见配置流程章节。



（2）回零流程编写如下所示。回零工艺说明：启动回零写全局变量回零中标志->Z 轴回零，-> X/Y/R 同时回零->将回零成功或失败信息给到主画面的信息打印->写全局变量回零成功/失败标志。

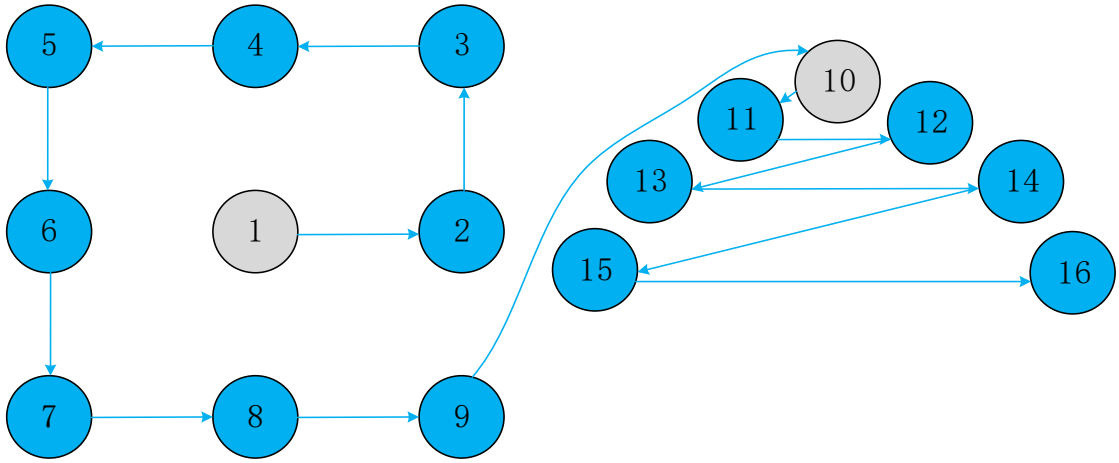


（3）标定流程编写如下所示。标定工艺说明：Z 轴抬高到安全高度（单轴运动工具块）->工具组使能循环 16 次->JavaScript 计算 16 个标定位置->X/Y/R 到第一个标定位（单轴运动工具块）->Z 到标定高度（单轴运动工具块）->定时器延时->多特征定位（拍照并提取特征中心）->将第 N（1~16）个标定位置的图像坐标及物理坐标给到 N 点标定工具块->结果显示第一个特征点->循环 16 次完成->判断标定结果 OK/NG 并输出。

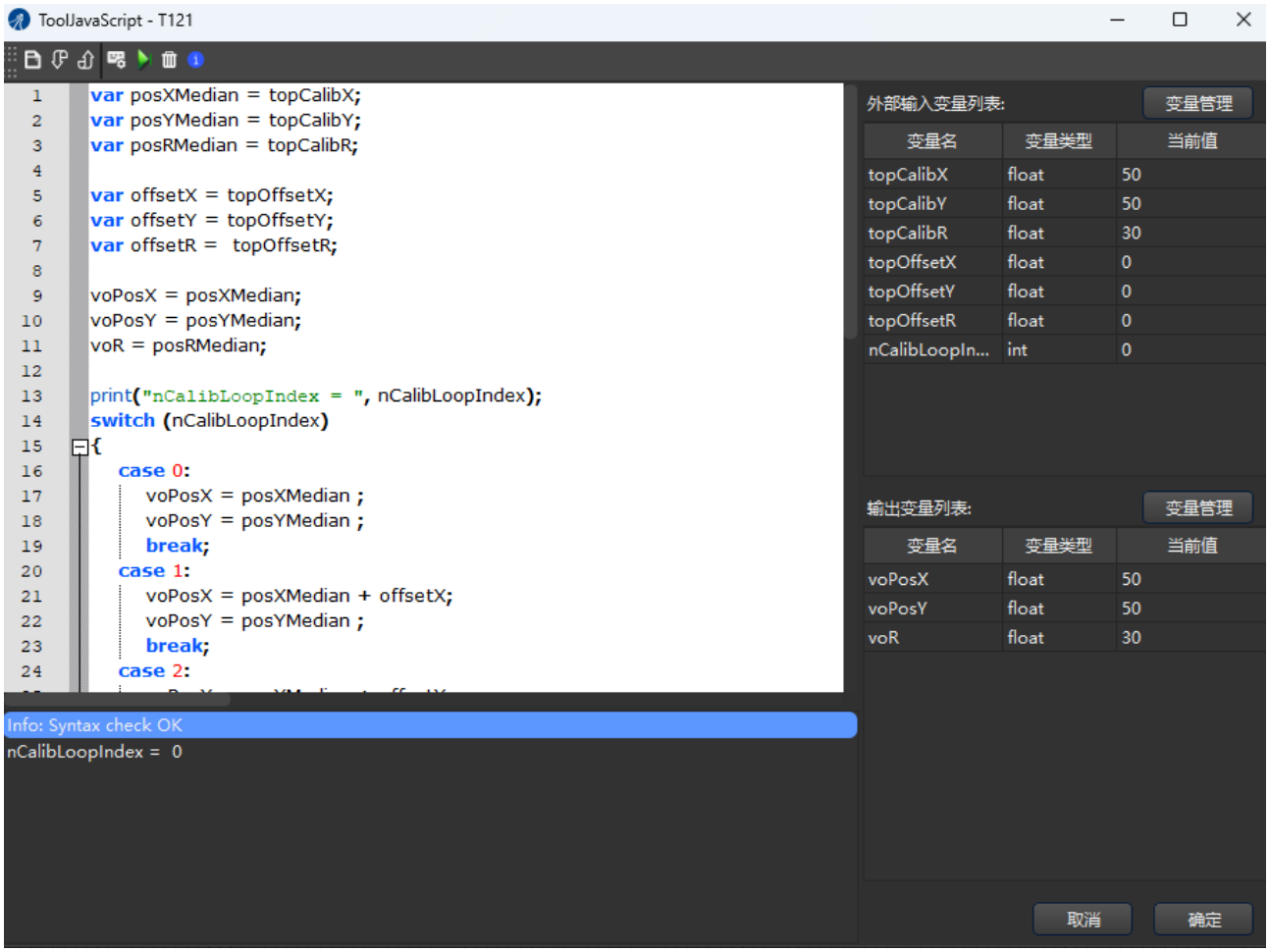


标定流程使用的 N 点标定工具块说明如下：

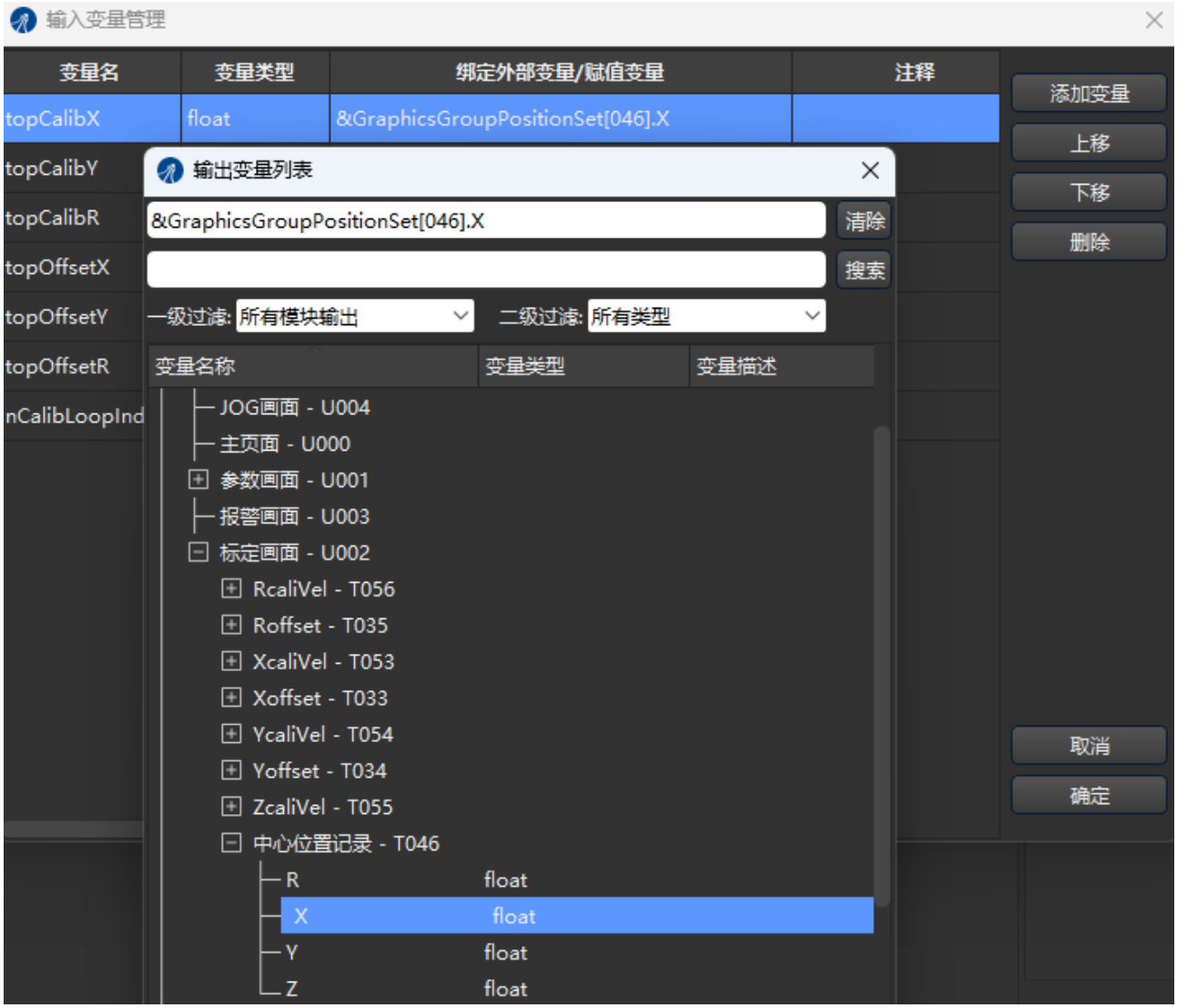
N 点标定包括 9 点标定（相对坐标）和 16 点标定，16 点标定是 9 点平移标定和 7 点旋转标定的组合，可以标定图像坐标系和物理坐标系的绝对关系，本例采用 16 点标定绝对关系，16 点位置示意图如下，其中 10 号位置与 1 号位置相同。



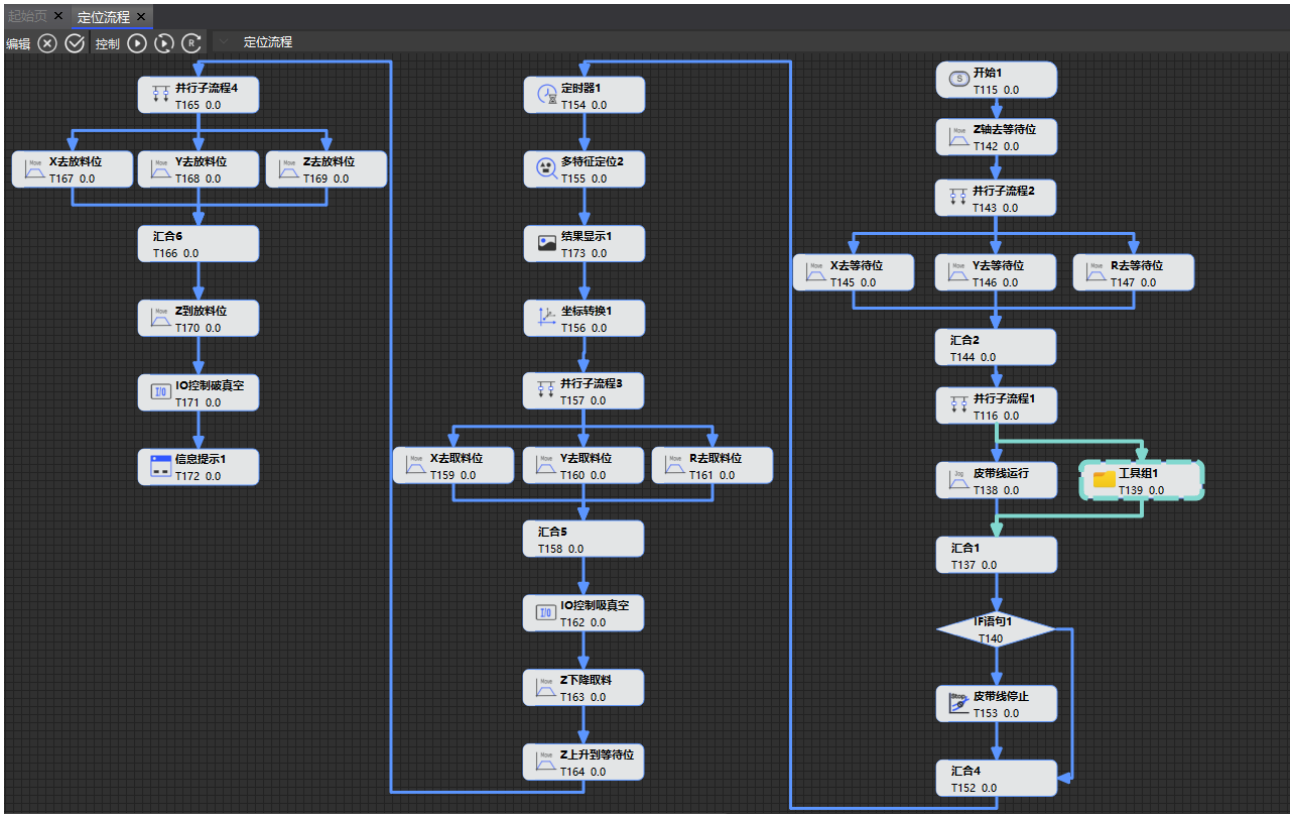
可以在用户界面示教标定中心点 1 的位置，其他位置根据 X 偏移、Y 偏移及角度偏移计算得出。计算过程需使用 JavaScript 脚本工具，如下图所示。



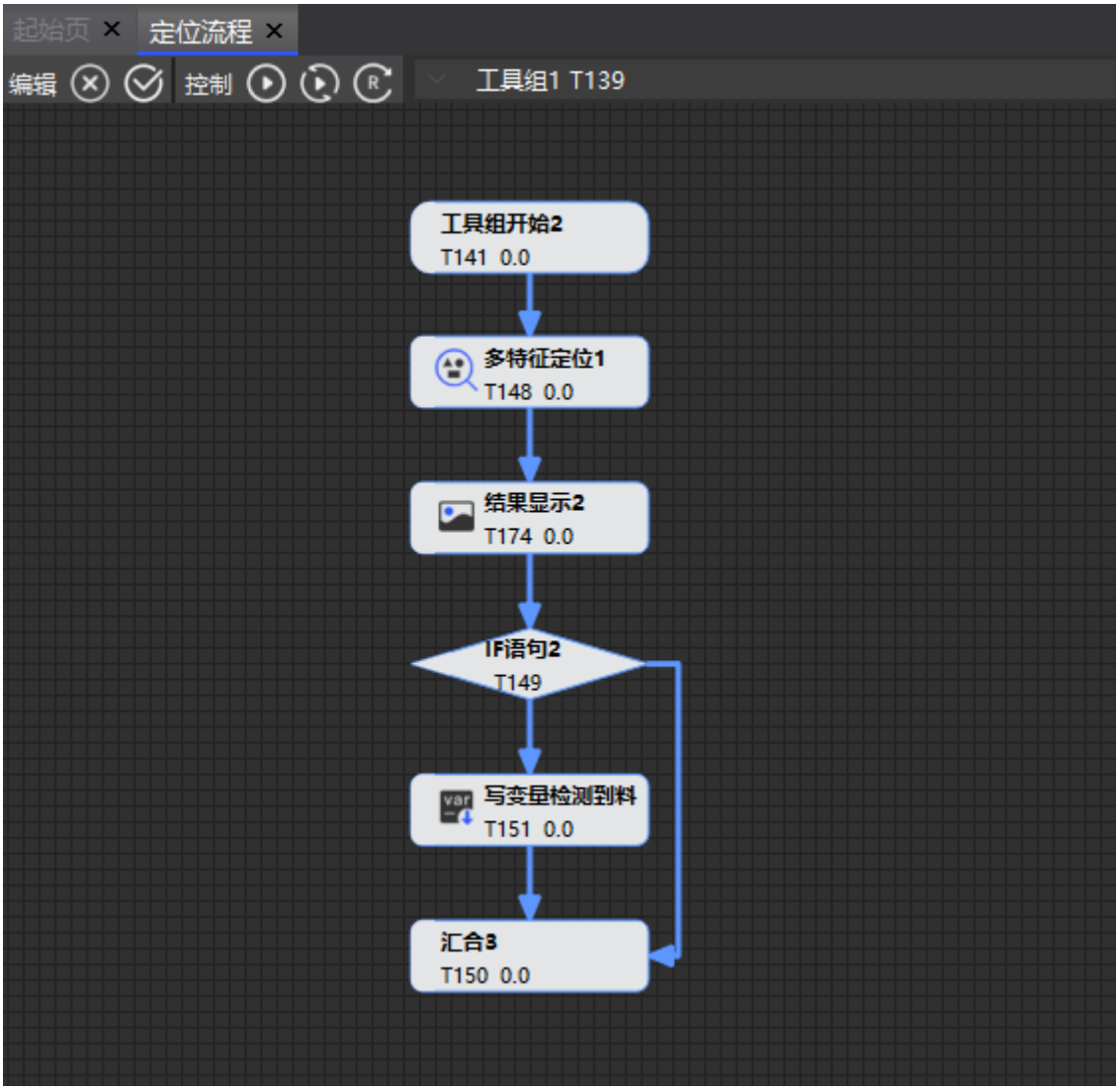
JavaScript 脚本变量订阅标定画面的位置值及偏移值，如下图所示。



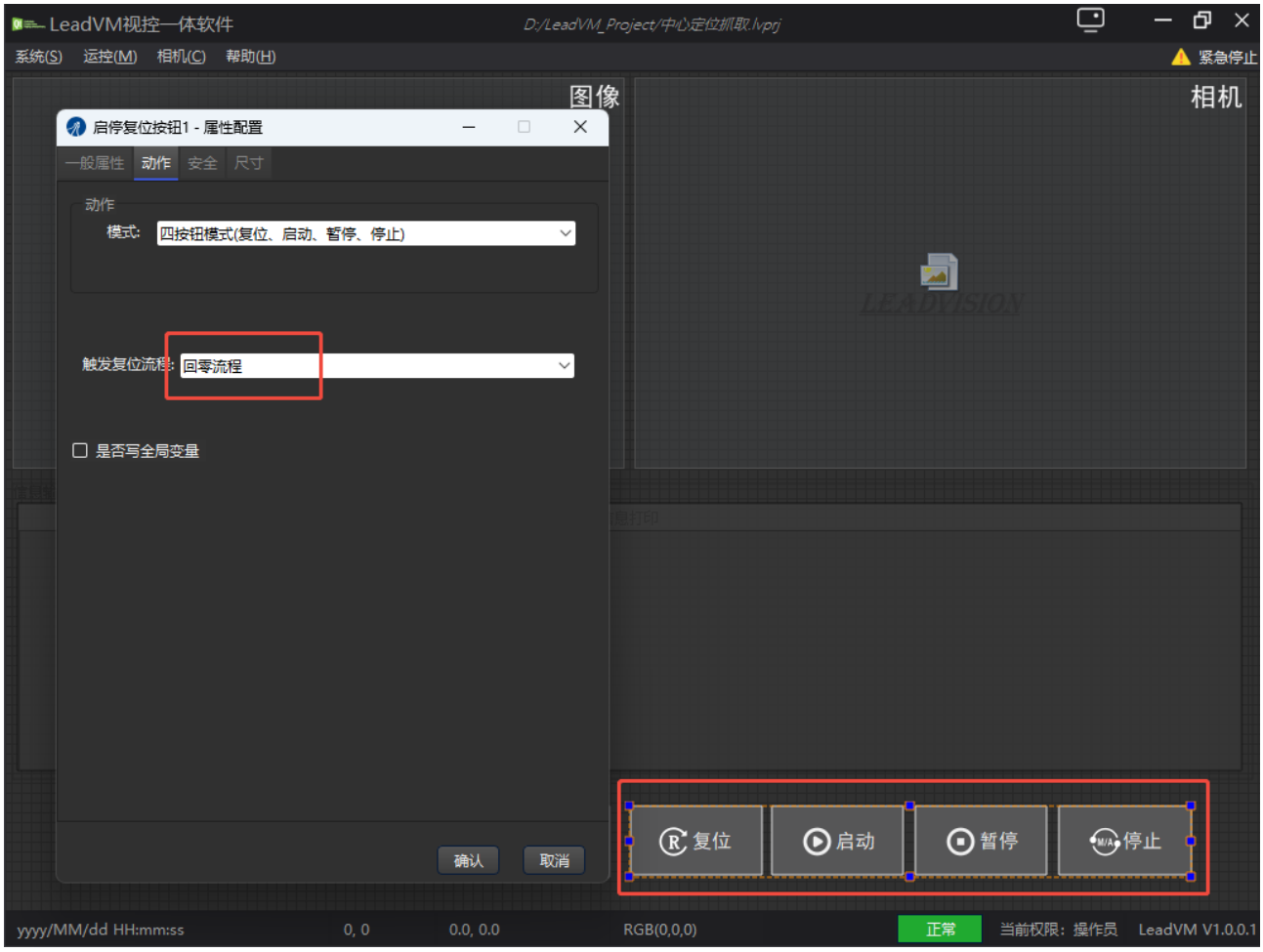
（4）定位流程编写如下所示。视觉检测到皮带线有料，停止皮带线，通过特征定位进行精定位，输出取料中心点图像坐标及角度，转换为物理坐标给到实际机台轴。


起始页
定位流程
编辑
控制
定位流程
并行子流程4
T165 0.0
X去放料位
T167 0.0
Y去放料位
T168 0.0
Z去放料位
T169 0.0
汇合6
T166 0.0
Z到放料位
T170 0.0
IO控制破真空
T171 0.0
信息提示1
T172 0.0
定时器1
T154 0.0
多特征定位2
T155 0.0
结果显示1
T173 0.0
坐标转换1
T156 0.0
并行子流程3
T157 0.0
X去取料位
T159 0.0
Y去取料位
T160 0.0
R去取料位
T161 0.0
汇合5
T158 0.0
IO控制吸真空
T162 0.0
Z下降取料
T163 0.0
Z上升到等待位
T164 0.0
开始1
T115 0.0
Z轴去等待位
T142 0.0
并行子流程2
T143 0.0
X去等待位
T145 0.0
Y去等待位
T146 0.0
R去等待位
T147 0.0
汇合2
T144 0.0
并行子流程1
T116 0.0
皮带线运行
T138 0.0
工具组1
T139 0.0
汇合1
T137 0.0
IFi语句1
T140
皮带线停止
T153 0.0
汇合4
T152 0.0

工具组 1 内部流程



（5）在用户组态页面配置启停复位按钮的复位流程，自此整个项目创建完成。


LeadVM视控一体软件
D:/LeadVM_Project/中心定位抓取.lvproj
系统(S)
运控(M)
相机(C)
帮助(H)
紧急停止
图像
相机
启停复位按钮1 - 属性配置
一般属性
动作
安全
尺寸
动作
模式:
四按钮模式(复位、启动、暂停、停止)
触发复位流程:
回零流程
是否写全局变量
确认
取消
复位
启动
暂停
停止
yyyy/MM/dd HH:mm:ss
0, 0
0.0, 0.0
RGB(0,0,0)
正常
当前权限：操作员
LeadVM V1.0.0.1

客户咨询中心
目录索取·技术咨询·产品解惑

400-885-5521 销售热线
400-885-5501 技术热线

更多最新的雷赛资讯，请扫码关注！

公众号　　视频号

成就客户 共创共赢

深圳市雷赛智能控制股份有限公司
China Leadshine Technology Co., Ltd.

深圳市南山区沙河西路3157号南山智谷产业园B栋15-20层
邮 编：518052
电 话：400-885-5521　　传 真：0755-26402718
网 址：www.leisai.com　E-Mail：marketing@leisai.com

上海分公司
上海市嘉定区金园五路601号

北京办事处
北京市大兴区绿地启航国际3号楼1109室

山东办事处
济南市天桥区滨河商务中心D座2003室

苏州办事处
江苏省苏州市苏州工业园区金尚路1号仙峰大厦南楼7层

合肥办事处
合肥市蜀山区潜山路与高河东路交口绿地蓝海大厦A座1209室

武汉办事处
湖北省武汉市东湖新技术开发区长城园路2号海贝孵化器209

温州办事处
浙江省温州市瓯海区潘桥街道宁波路阳光城愉景嘉园8幢2604

青岛办事处
山东省青岛市城阳区金日紫都小区12号楼1单元301室

杭州办事处
浙江省杭州市余杭区瓶窑镇桂花溪园(南区)2幢1单元402

广州办事处
广州市番禺区汉溪大道280号时代E-park A3栋1707单元

※本产品目录中所刊载的产品性能和规格，如因产品改进等原因发生变更时，恕不另行通知，敬请谅解。

（版权所有，翻版必究）

2023年07月版