目录

一、 产品概述 1
1 功能特性 1
2 规格描述 1
3 外形尺寸 2
4 接口使用说明 2
二、 运行界面 4
1 实时监控画面 4
2 实时检测步骤 5
3 实时统计 6
4 数据报表 6
5 审计日志 7
6 运行日志 8
三、 模型训练 8
1 样本采集 8
2 显卡驱动安装 8
3 数据标注与模型训练 10
4 模型导出 11
四、 账号登录 12
五、 创建作业 13
1 新建作业 13
2 新建工序 13
3 作业设置 14
4 作业脚本 15
六、 相机设置 16
1 MVS 相机 16
2 VFS 相机 17
3 IMG 18
七、 板载控制器设置 19
1 添加控制器 19
2 详细参数配置 20
2.1 输出引脚 20
2.2 PWM 通道 20
2.3 开启延时 21
2.4 关断延时 21
八、 IO 设备设置 22
1 串口 22
2 S7client 22
3 TCP 客户端 23
4 TCP 服务端 24
九、 心跳控制器 24
十、 工具说明 25
1 图像模块 25

1.1 图像采集 25
1.2 视频采集 26
1.3 图像转灰 27
2 测量模块 29
3 检测模块 29
3.1 斑点分析 29
3.2 多轮廓匹配 33
3.3 LAB 颜色检测 37
3.4 RGB 颜色检测 38
3.5 HSV 颜色检测 39
3.6 颜色识别 41
4 条码模块 42
4.1 扫码 42
5 逻辑模块 44
6 通信模块 44
6.1 IO 读 45
6.2 MB 读 46
6.3 MB 写 48
6.4 写二进制数据 49
6.5 写文本数据 50
6.6 警报控制 51
6.7 FT02 输入 53
6.8 FT02 输出 53
6.9 系统输出 54
7 校准模块 55
7.1 开始位置补正/结束位置补正 55
7.2 相机标定 57
7.3 世界—图像坐标映射 60
8 AI 模块 61
8.1 对象检测 61
9 系统模块 64
9.1 休眠等待 64
9.2 数据显示 64
9.3 标记点 66
9.4 完成工序 67
9.5 动作指引 67
9.6 图像保存 69
9.7 视频录制 73
9.8 执行脚本 73
十一、 执行脚本开发 74
1 脚本基础框架 74
2 参数说明（通用所有函数） 74
2.1 this（脚本实例对象） 74
2.2 ctx（全局执行上下文工具对象） 74

3 五大函数详细说明 75
3.1 作业初始化函数 setup 75
3.2 工序前置函数 on_process_begin 75
3.3 核心执行函数 exec 75
3.4 工序后置函数 on_process_end 75
3.5 作业清理函数 cleanup 75
4 执行上下文对象的接口 76
4.1 根据工具名字获取工具对象 76
4.2 根据工具 id 获取工具对象 76
4.3 访问指定的工具对象 76
4.4 根据设备名字获取设备 76
4.5 根据设备 id 获取设备 76
4.6 强制完成当前工序 77
5 工具对象可调用函数 77
5.1 获取工具 id 77
5.2 获取工具名称 77
5.3 执行工具 77
5.4 获取执行的错误码 77
5.5 获取执行的错误消息 77
6 系统空间函数 78
6.1 事件发送 78
6.2 设置引脚输出值 78
6.3 读取开关状态 78
6.4 设置开关状态 78
6.5 读取 pwm 状态 78
6.6 设置 pwm 状态 78
6.7 读取 pwm 设置值 79
6.8 设置 pwm 值 79
6.9 读取开启延时时间 79
6.10 设置开启延时时间 79
6.11 读取关闭延时时间 79
6.12 设置关闭延时时间 79
7 任务空间函数 80
7.1 事件发送 80
7.2 视频保存 80
8 日志函数 80
8.1 DEBUG 日志 80
8.2 INFO 日志 80
8.3 WARN 日志 81
8.4 ERROR 日志 81
8.5 审计日志空间函数 81
9 ROI 模块 81
9.1 获取 ROI 列表 81
9.2 清除工具中的 ROI 81

9.3 添加矩形 ROI 82
9.4 添加旋转矩形 ROI 82
9.5 添加圆形 ROI 82
9.6 添加环形 ROI 82
9.7 添加多边形 ROI 82
10 S7Client 设备 82
10.1 连接 PLC 83
10.2 断开连接 83
10.3 读取 M 区数据 83
10.4 读取 A 区数据 83
10.5 读取 E 区数据 83
10.6 读取 D 区数据 83
10.7 写 M 区数据 83
10.8 写 A 区数据 84
10.9 写 E 区数据 84
10.10 写 D 区数据 84
10.11 设置连接地址 84
10.12 获取连接地址 84
10.13 设置连接机架号 84
10.14 获取连接机架号 85
10.15 设置连接槽位号 85
10.16 获取连接槽位号 85
10.17 写 D 区一个 8 位整数 85
10.18 写 D 区一个 16 位整数 85
10.19 写 D 区一个 32 位整数 85
10.20 写 D 区一个 64 位整数 85
10.21 读 D 区一个 8 位整数 85
10.22 读 D 区一个 16 位整数 86
10.23 读 D 区一个 32 位整数 86
10.24 读 D 区一个 64 位整数 86
11 BarCode 工具 86
11.1 获取工具执行结果 86
11.2 获取识别条码的最大数量 86
11.3 设置识别条码最大数量 86
11.4 获取搜索条码阈值 87
11.5 设置搜索条码阈值 87
11.6 获取或设置最大条宽度 87
11.7 设置最大条宽度 87
12 DLObjDetect 工具 87
12.1 获取工具执行结果 87
13 DLPoseDetect 工具 88
13.1 获取工具执行结果 88
14 DLSegDetect 工具 88
14.1 获取工具执行结果 88

15 MarkPoint 工具 88
15.1 获取原始点数组 89
15.2 获取位置补正后点数组 89
16 MBWrite 工具 89
16.1 发送裸 MB 数据帧 89
16.2 写线圈 89
16.3 写单个寄存器 89
16.4 写字符串 89
16.5 写二进制数据 90
16.6 获取最后一次请求的响应数据 90
17 MBRead 工具 90
17.1 获取读到的数据 90
17.2 获取最后一次请求的响应数据 90
18 Blob 工具 90
18.1 获取工具执行结果 90
19 ImageSave 工具 91
19.1 获取图片保存后缀 91
19.2 设置图片保存路径 91
19.3 获取是否保存原图 91
19.4 设置是否保存原图 91
19.5 获取是否保存渲染图 92
19.6 设置是否保存渲染图 92
19.7 获取保存原图自定义文件名 92
19.8 设置保存原图自定义文件名 92
19.9 获取保存渲染图自定义文件名 92
19.10 设置保存渲染图自定义文件名 92
20 FT02Input 工具 92
20.1 获取读到的输入引脚状态 93
21 AlarmControl 工具 93
21.1 判断是否连接 93
22 ShowData 工具 93
22.1 设置单个 ui 参数 93
23 ActionGuide 工具 94
23.1 获取指示器状态 94
23.2 设置指示器状态 94
23.3 设置 ui 参数 94
23.4 清除指引列表 94
23.5 添加矩形指引 94
23.6 添加旋转矩形指引 95
23.7 添加圆形指引 95
23.8 添加环形指引 95
23.9 添加多边形指引 95
23.10 设置矩形指引 95
23.11 设置旋转矩形指引 95

23.12 设置圆形指引 96
23.13 设置环形指引 96
23.14 设置多边形指引 96
24 ColorIdentify 工具 96
24.1 判断是否有匹配的颜色 96
24.2 获取匹配颜色的名称 96
24.3 获取匹配颜色的分数 96
25 HSVMonitor 工具 97
25.1 判断颜色是否可接受 97
26 LABMonitor 工具 97
26.1 判断颜色是否可接受 97
27 RGBMonitor 工具 97
27.1 判断颜色是否可接受 97
28 WorldToImageTranslate 工具 98
28.1 清除图像坐标点 98
28.2 清除世界坐标点 98
28.3 清除源坐标点 98
28.4 添加图像与世界坐标点 98
28.5 添加源坐标点 98
28.6 获取转换结果 98
28.7 世界坐标转相机坐标 98
28.8 相机坐标转图像坐标 99
28.9 清除相机参数 99
28.10 设置相机参数 99
29 ScanCode 工具 99
29.1 获取扫码结果 99
30 EndTimeMark 工具 99
30.1 获取时间间隔 99
31 VideoRecordTool 工具 100
31.1 保存视频文件 100

一、 产品概述

本款产品为一块 SOP 嵌入式开发板，集成 4 进 4 出 IO 模块，实现多设备间的电平交互通信；内置 4 组独立触发模块与 PWM 光源控制模块，体积小巧紧凑，适应工业现场复杂环境下的快速安装部署。

1 功能特性

① 集成作业规范软件；

② 集成 4 路输入输出；

③ 支持连接千兆网口通信；

④ 支持连接一个 2.5G 网口相机；

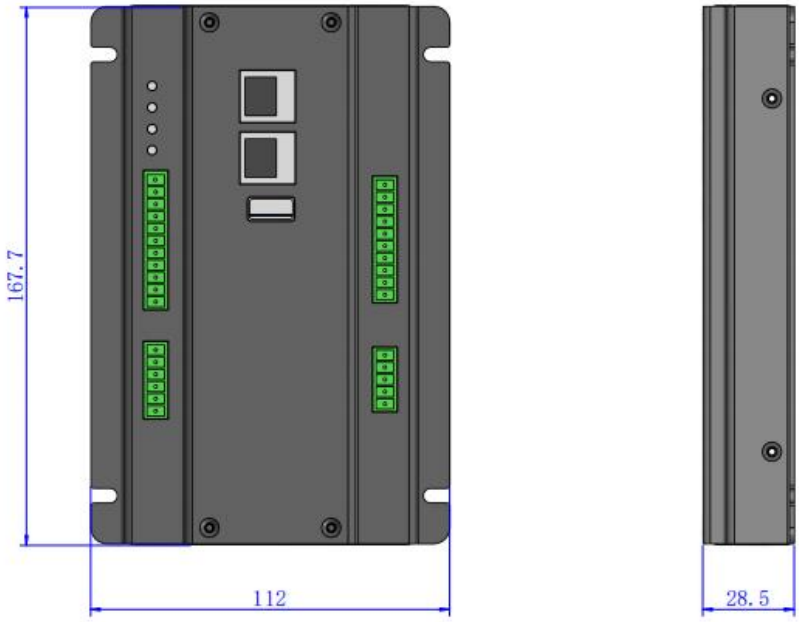
⑤ 支持外部触发控制 PWM 输出控制光源；

⑥ 具备掉电参数保存、过载 / 短路硬件保护功能。

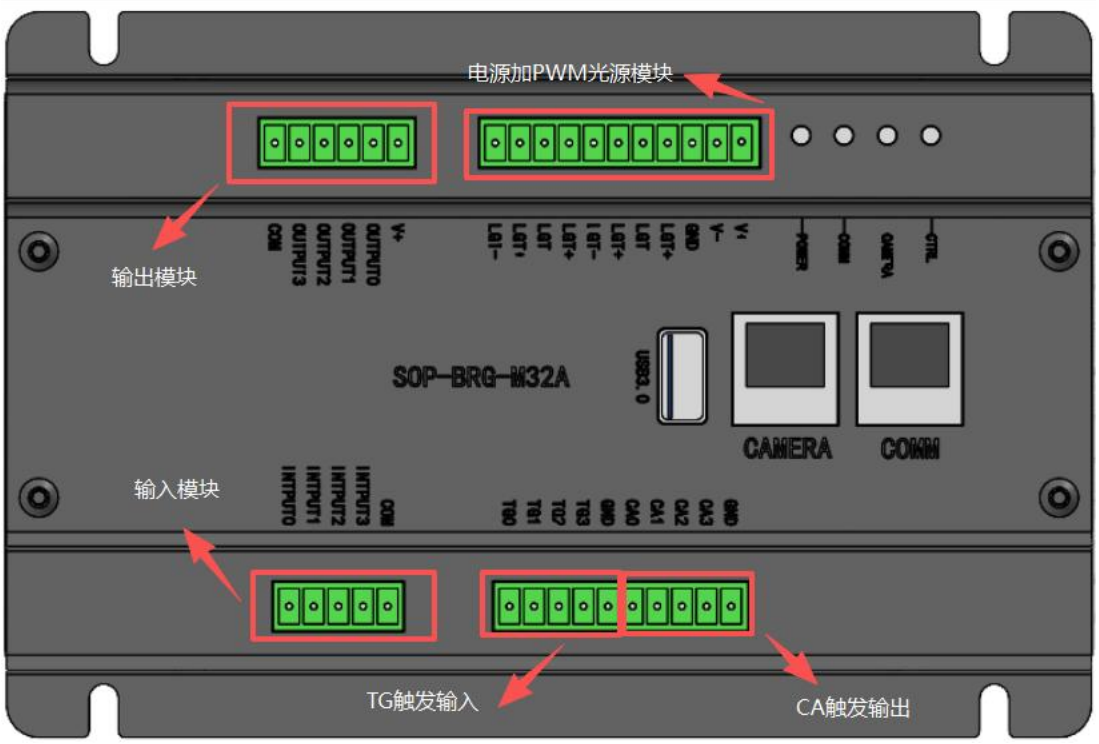
2 规格描述

产品名称	行为规范控制器
产品型号	SOP - BRG - M32A
硬件版本	Rev 1.0
输入电压	24V
光源输出电压	DC24V
单 PWM 通道最大功率	150W
4 路输入电压	24V
Usb3.0	1 个
网口	1 个千兆网口，1 个 2.5G 网口
散热方式	铜块导热

3 外形尺寸



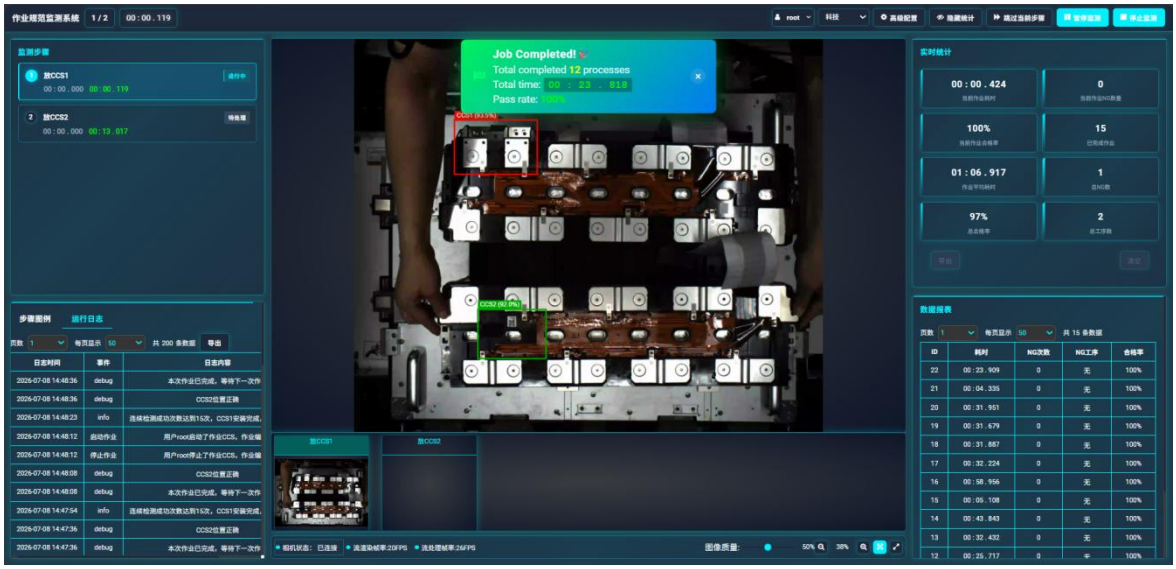
4 接口使用说明



开发板通电：将 24V 电源正负极接入电源和 PWM 光源模块 V+，V-即可通电
PWM 光源：光源正负按照 LGT+，LGT-依次接入即可
输出模块：需要外接 24V 电源到输出模块的 V+和 V-，各输出口输出低电平
输入模块：高电平接入输入口即可，负极接入 GND
触发模块：触发信号接入 TG 触发输入，按照正负极分别接入 TG 端和 GND 即可
CA 触发输出：接入 TG 触发后，对应 CA 按照 CA 端（高电平）和 GND 端接入即可有触发输出

二、 运行界面

运行界面中总共分为六大模块，分别是实时监控画面、实时检测步骤、实时统计、数据报表、审计日志、运行日志，总体布局如下图。



1 实时监控画面

监控画面来源为视频采集、图像采集工具，同时支持展示图像处理后的画面，画面按工具执行顺序依次展示，可实时监控现场作业全过程，界面叠加显示模型识别框、作业 OK 合格/NG 不合格弹窗；单道工序判定 OK 或 NG 后系统自动截图留存。界面底部显示五类运行信息：

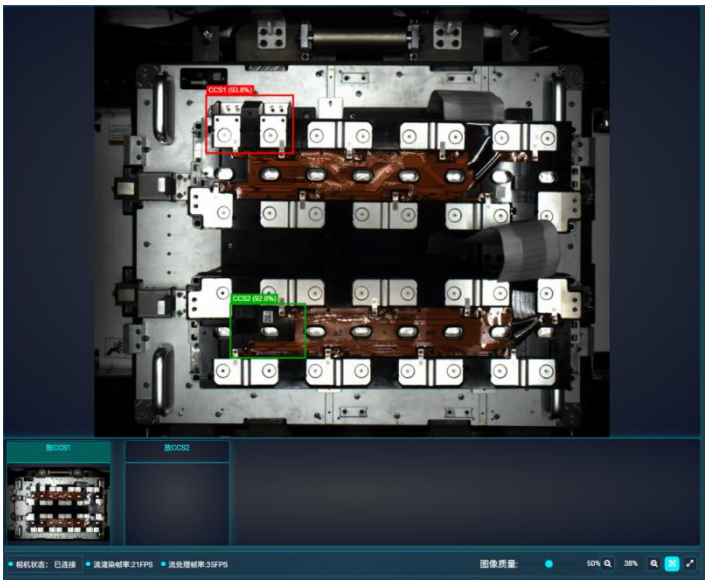
①相机连接状态：设备正常采集时显示在线，相机异常断开则提示断连；

②视频流渲染帧率：界面每秒渲染刷新的图像帧数；

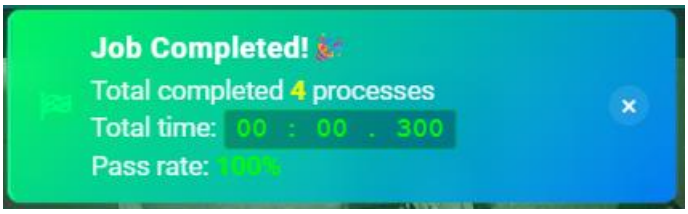
③视频流处理帧率：算法每秒实际运算处理的图像帧数。

④图像质量大小：可自由调节渲染图像的质量

⑤图像缩放：可对运行画面的图像大小进行调节



若全部工序正常执行完毕、全程无检测异常，系统将弹出对应完成提示弹窗，弹窗内展示本次作业完成工序总数、整体耗时以及检测通过率三项数据。



若任意工序检测判定为 NG/不合格，系统会弹出 NG 提示框并展示具体异常内容，同时暂停整套程序，需人工执行复位操作后才可继续运行；该弹窗由自定义 Task 事件触发，相关细节将在后续脚本开发章节详细说明。

2 实时检测步骤

实时检测区域会同步展示各工序运行状态：尚未启动检测的工序显示待处理，正在执行检测的工序显示正在处理，顺利完成检测的工序显示已完成。

处于正在处理状态的工序将实时统计运行时长，界面灰色进度代表预设超时阈值（超时参数配置方法详见创建作业章节）；若实时运行时长超出设定超时时间，系统自动弹出 NG 弹窗并暂停整套程序，需人工执行复位操作后才可继续运行。

3 实时统计

实时统计区域用于记录作业全过程各项数据，数据在未手动清空前持续保存，支持随时导出 CSV 格式表格，共记录八项统计数据：

①当前作业耗时：统计单次作业累计运行时长

②当前作业 NG 数量：统计单次作业内产生的不合格次数

③当前作业合格率：单次作业合格占比，计算公式：(工序总数-NG 工序数)÷工序总数

④已完成作业：程序启动至数据清空前，全部完成的作业总量

⑤作业平均耗时：全部完成作业的平均用时，计算公式：所有作业总耗时÷已完成作业数量

⑥总 NG 数：程序启动至清空前所有不合格总数，包含自定义事件 NG、超时触发 NG

⑦总合格率：全部作业整体合格占比，计算公式：(总作业数-NG 作业数)÷总作业数

⑧总工序数：当前流程一套作业所需执行的全部工序总量

4 数据报表

数据报表区域每完成一次作业自动生成一条记录，保存单次作业耗时、结果与本次作业合格率；支持自定义单页显示条数，可通过页码检索对应记录。

5 审计日志

审计日志包含日志时间、事件、日志内容三类信息，分为审计日志、用户自定义日志两种类型。审计日志用于记录平台启停、暂停退出、修改工具参数、切换模块等用户操作行为；用户自定义日志由用户在脚本内调用审计日志命名空间实现，可记录脚本检测状态、工序合格、NG 异常及其他自定义信息（具体自定义步骤将在后续脚本开发章节详细说明）。日志数据会持久保存，仅在手动执行清空操作时才会被清除，支持随时导出。

6 运行日志

运行日志页面仅展示用户自定义日志，该日志通过非审计日志命名空间下的相关接口实现自定义输出，支持 DEBUG、INFO、WARN、ERROR 四种日志等级。该页面仅允许 root 及工程师账号访问查看；运行日志单页最多展示 200 行，支持开启自动刷新功能，刷新间隔最低可设置为 1 秒，主要用于程序调试工作。

三、 模型训练

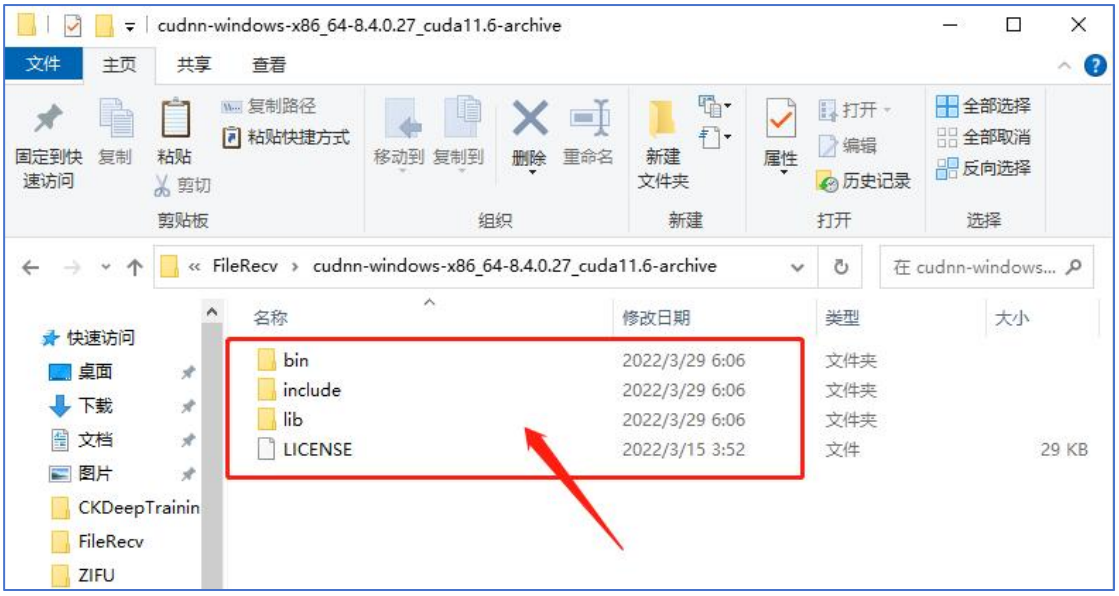
1 样本采集

在应用现场装配相机镜头并调好焦距与相机参数，借助作业规范检测处理器或电脑端相机客户端完成视频采集。样本采集需满足基础要求：光照均匀、拍摄视角稳定、作业流程规范、成像视野清晰，画面无遮挡、背景无多余干扰物。

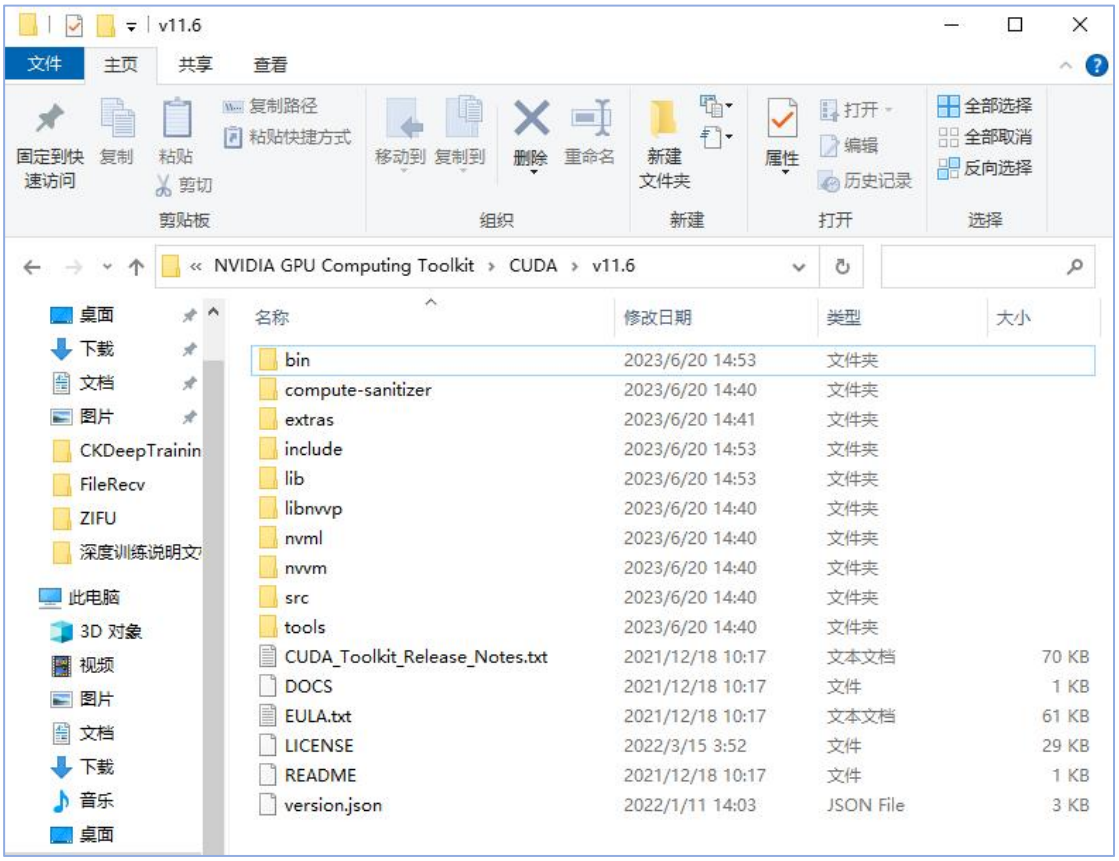
2 显卡驱动安装

安装 Cuda 显卡驱动包。下载 cuda_11.6.0_511.23_windows 安装程序，确保版本号为 11.6，安装过程使用默认设置即可。

安装完成后，再下载 cudnn-windows-x86_64-8.4.0.27_cuda11.6-archive 压缩包并解压得到如下图中的文件：

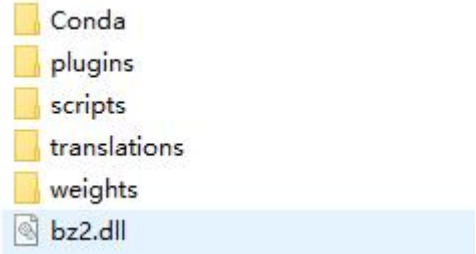


将这三个文件夹全选并复制粘贴到 CUDA 的安装目录下（通常目录为 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.6\），与 CUDA 原有的目录合并。默认情况下，如下图所示：

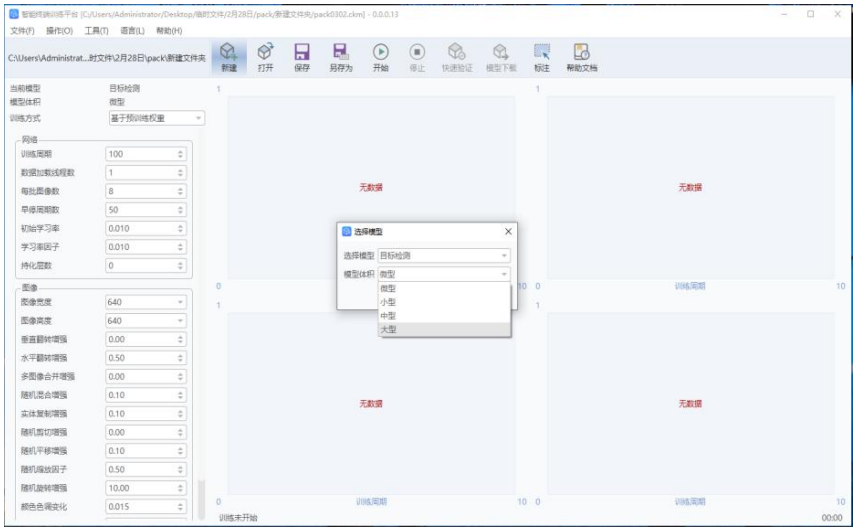


3 数据标注与模型训练

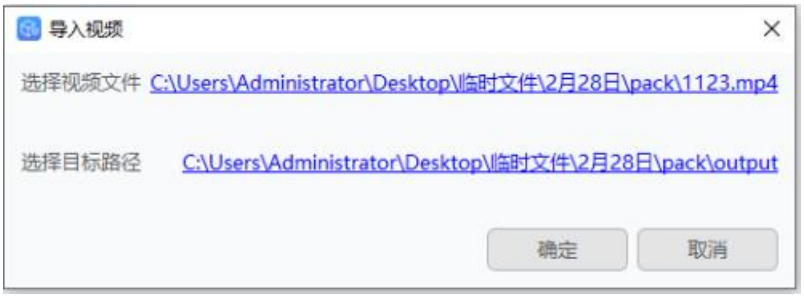
解压 WebDeepTraining 压缩包，再将 WebConda 压缩包解压，并命名为 Conda，WebDeepTraining 目录下，再点击 exe 文件进入平台，如下



模型训练需要借助可进行深度学习训练的工控机（以下简称 PC）完成，在 PC（目前仅支持 Windows 系统）中打开 WebDeepTraining 训练工具，点击新建，可以根据任务复杂度以及 PC 端显卡配置选择模型体积，通常情况下选择微型或小型。



点击工具栏-工具，选择导入视频，选择视频文件并设置图像的目标路径。



导入视频完成后，选择之前存放图片的目录，然后在主菜单中找到标注，开始标注工作，标注前需要根据任务对图片样本做相应的分类，标注完成后，即可开始训练，训练周期数建议设置为 800-1000，数据加载线程数依据设备 CPU 核心数量设定，取值通常小于 CPU 核数，常规推荐 8，早停周期数建议设置为 100-200，其他参数建议使用默认值。

4 模型导出

模型训练完成后，需将模型导出并下发至处理器平台。使用网线连接处理器与 PC 端，把两台设备配置为同一网段；在处理器平台主菜单打开模型下载功能，设备与 PC 建立正常通信后，选中对应设备 IP 地址，即可完成训练模型的下载。登录处理器平台后，在模型列表中可查看已导入文件，完整操作步骤详见工具说明章节。

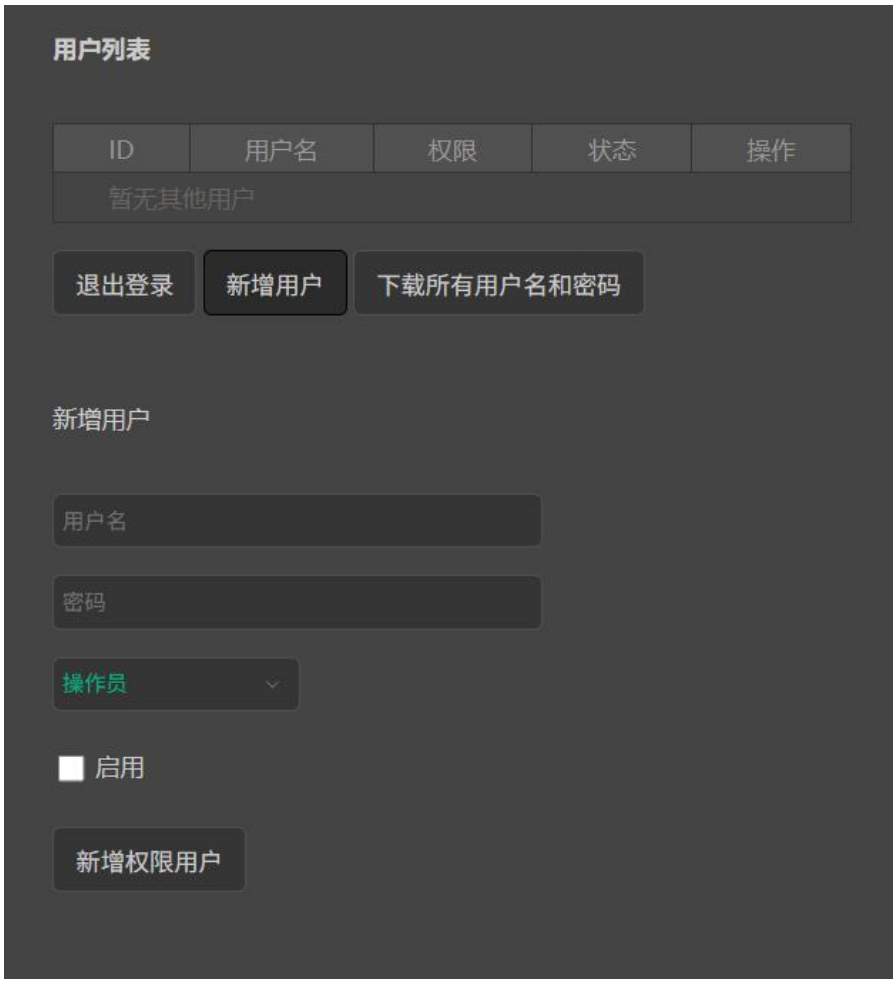
四、 账号登录

默认管理员账号：root，登录密码：WebCamera@CKVision。 进入编辑模式后可新增用户账号，系统权限划分为管理员、工程师、操作员三个等级：

①管理员：具备全部操作权限，可修改全部参数配置与脚本文件；

②工程师：拥有部分操作权限，仅可修改指定参数配置；

③操作员：仅能操作运行界面，无权限进入编辑模式修改程序



五、 创建作业

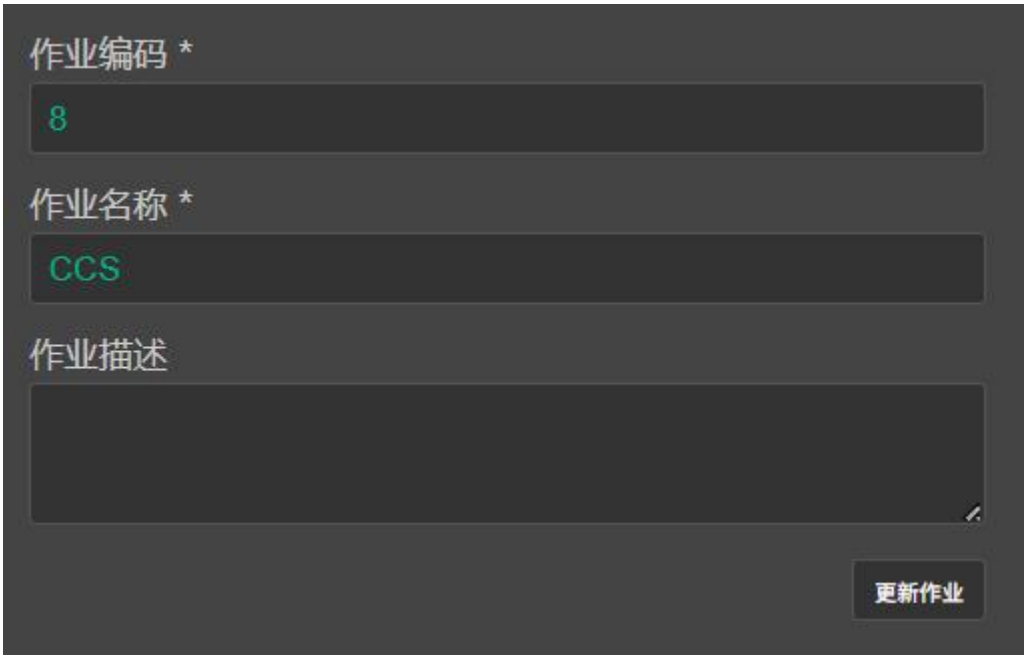
1 新建作业

可进入到

进去到编辑界面，点击图标 进入到作业管理界面。平台支持两种作业创建方式，分别为新建作业与导入已有作业，具体操作方式如下：

创建新作业：点击【新作业】按钮，系统将弹出作业创建弹窗（如下图）。操作人员可根据需求自主设置作业编码、作业名称，并填写对应的作业简介，完成基础信息配置后即可创建全新作业。

导入旧作业：平台支持导入格式为 JSON 的作业文件，该文件可完整留存作业的全部配置信息与参数内容。导入作业时，直接点击上传按钮，选取对应的 JSON 作业文件完成上传，即可快速复用已有作业内容，无需重复配置参数。

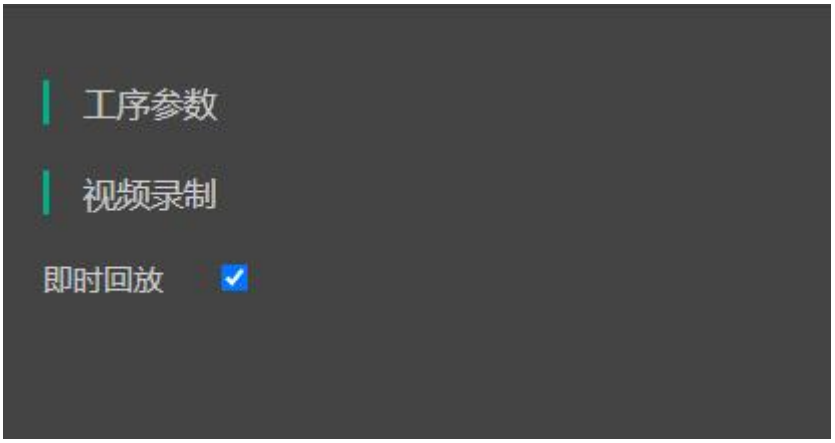


2 新建工序

完成新作业创建后，需为作业配置工艺流程。点击【添加工序】按钮，系统弹出工序配置界面（详见附图）。

依次填写以下配置项：工序名称、工序描述、工序插入位置（多工序场景下用于指定插入节点）、超时时间、工序图例、默认工具；勾选默认工具后，后续新增工序将自动沿用该工具。全部参数填写完毕，点击【添加】即可完成工序新增。

此外，每个工序可独立配置视频录制功能。开启视频录制后，可在脚本中调用 Task.save_video()接口保存当前工序的运行视频。



3 作业设置

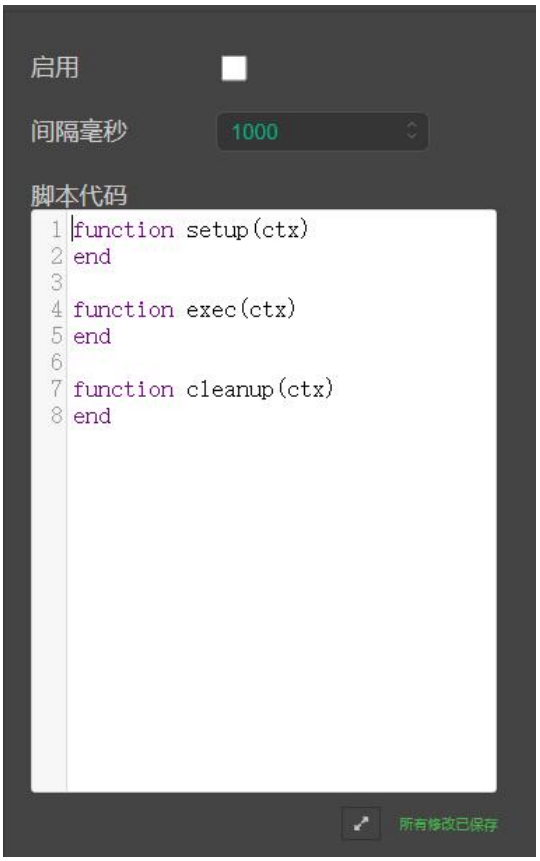
作业设置功能可解决工序数量较多、更换相机或模型时需逐工序手动修改的繁琐问题。点击【设置】按钮，目前可统一配置作业默认相机与默认模型：

默认相机：选定后，该作业下所有包含图像采集、视频采集工具的工序，相机来源将同步更新为此默认相机；

默认模型：选定后，作业中所有搭载对应模型工具的工序，模型参数统一替换为所选模型，当前支持对象检测、姿态估计、对象分割三类工具。

4 作业脚本

每个作业支持配置独立的作业脚本。脚本启用后，若作业在运行过程中被暂停，系统将按照预设间隔周期定时执行该脚本。该功能常用于作业暂停后进行信号收发，实现作业自动恢复启动。详细使用方式请查阅执行脚本开发章节。



六、 相机设置

在对工序进行编辑前，需要先配置相机，为检测流程提供图像来源，目前主要使用相机类型为 MVS 与 VFS，以下主要介绍相机的配置：

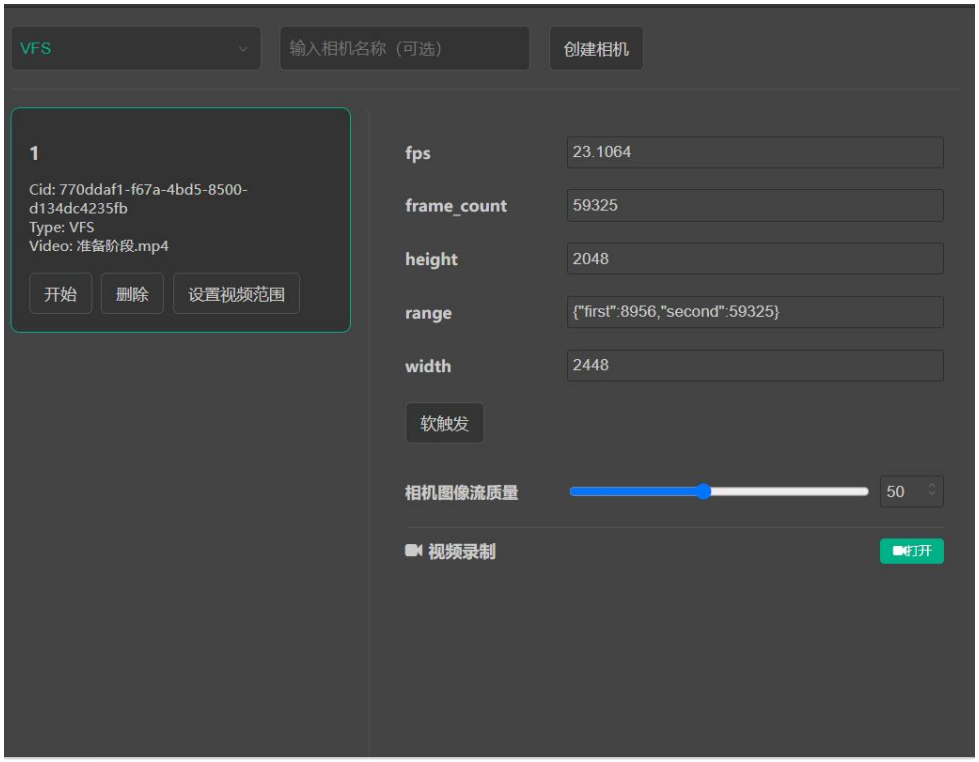
1 MVS 相机

本配置用于对接海康工业相机，添加设备时可自动检索同一网段内全部海康相机，选中目标相机即可完成设备接入。设备添加成功后，可配置相机各类参数，支持设置自动曝光、曝光时间、增益、触发延时、触发模式、触发源等；同时可调整相机图像流画质参数，画质设置会直接影响上位机界面渲染图像效果，操作界面如下：

2 VFS 相机

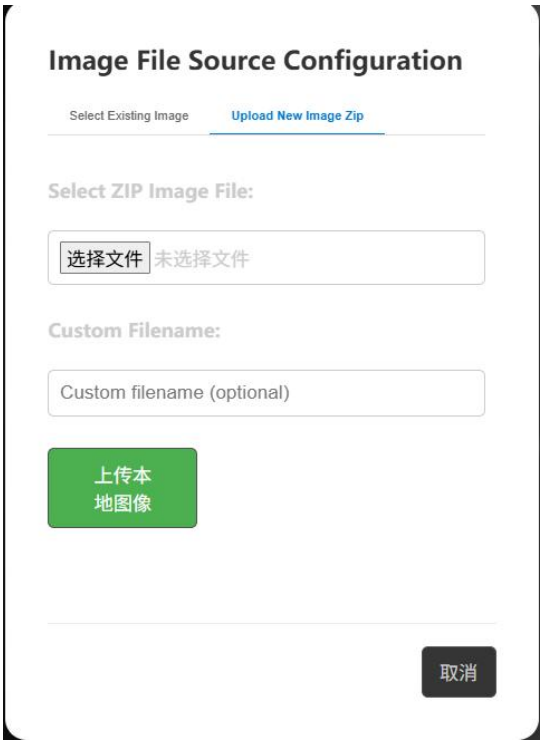
本配置用于导入本地视频文件，可在无实体工业相机的环境下，完成整套 SOP 工序流程验证。选中 VFS 相机并点击添加，将弹出配置窗口，支持选用历史加载视频或导入全新视频文件。

视频添加完成后，可自定义虚拟相机输出帧率，代表每秒推送图像帧数；同时能够设置视频流起始帧与结束帧，截取指定片段，便于单工序独立调试。可调整图像画质参数，画质配置将直接影响上位机界面画面渲染效果。工具同步展示视频基础参数：图像宽度、高度、视频总帧数。操作界面如下：



3 IMG

该配置用于上传图片目录。上传文件必须为 zip 压缩包，解压后根目录直接存放目标图片集，禁止嵌套子目录。多用于导入已采集完成的标定板图像，供工具执行首次标定校准。



七、 板载控制器设置

本功能仅适用于搭载嵌入式底板的 SOP 处理器版本。该硬件版本提供多路输入 / 输出接口，一方面可对外控制相机、光源及其他外设；另一方面也能够接收外部信号，实现信号交互、程序启停控制等功能。本章主要说明板载控制器各模块引脚的配置流程。硬件接口布局如下图所示。

1 添加控制器

进入编辑模式，点击图标 可进入板载控制器设置，新增 STM32 控制器将弹出以下参数配置窗口：

1. 端口：设备自带串口，直接选用对应端口即可；
2. 串口默认参数：波特率 9600、数据位 8Bits、校验无、停止位 1Bit；
3. 超时时间：代表通讯连接超时阈值，按需自定义设置。

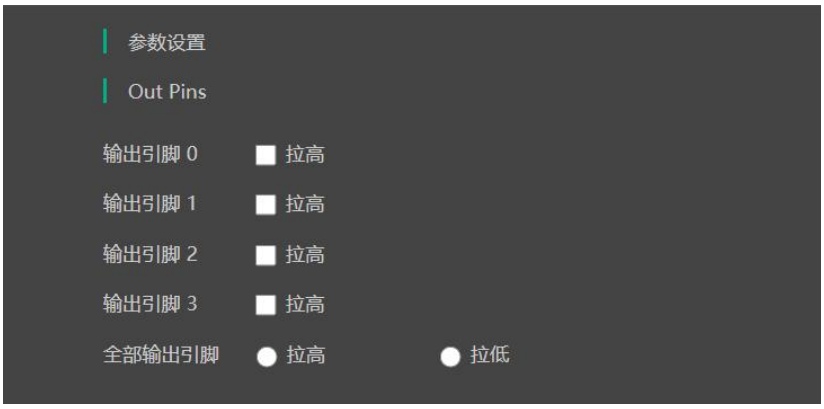
外部触发引脚对应硬件上的输入模块，用于接收外部开关信号，实现程序启动、暂停、停止、复位控制。一共提供 4 路引脚，分别对应四类作业触发事件，可根据现场实际接线需求自定义分配引脚与触发功能。

2 详细参数配置

参数配置完成后点击【开始】按钮建立与控制板的通讯连接，连接成功后自动跳转至详细参数配置界面。

2.1 输出引脚

设备共 4 路输出引脚，对应硬件的输出模块。支持单路独立控制与四路批量控制两种模式：勾选对应引脚的 “拉高” 选项，该引脚持续输出信号；取消勾选，引脚停止输出。也可使用【全部输出引脚】选项，选择拉高则四路引脚同步输出；选择拉低则四路引脚全部停止输出。操作界面如上所示。



2.2 PWM 通道

PWM 通道专用于光源控制，共四路，模式可选择常亮或触发：触发模式适用于外部信号触发场景，光源接对应引脚后依靠外部输入信号触发光源点亮；常亮模式由软件直接输出持续信号，无需

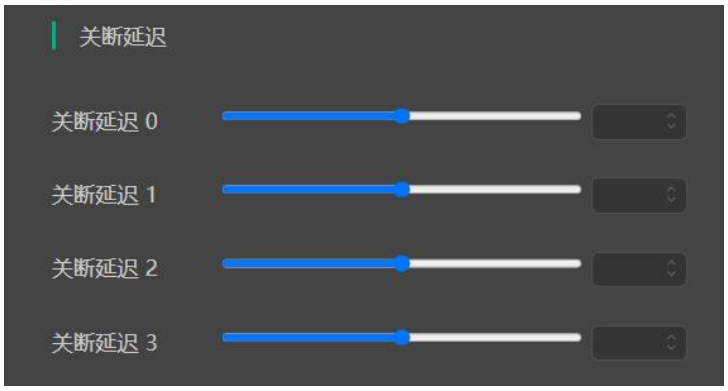
外部触发。所有 PWM 通道选项支持使能、失能切换，常亮与触发模式均需先使能才可点亮光源，失能时无任何输出；四路通道各对应一组 0~255 区间的 PWM 数值，用于调节光源亮度。操作界面如下：

2.3 开启延时

开启延时为四路 PWM 通道独立延时参数，取值范围 0~65535ms，设定数值后，收到外部触发信号会延迟对应毫秒再输出信号点亮光源。操作界面如下：

2.4 关断延时

关断延时为四路 PWM 通道独立参数，取值 0~65535ms，设定数值后，光源点亮对应时长后自动关断输出。



八、 IO 设备设置

进入到 IO 设备设置中，可添加 S7 客户端、串口、TCP 客户端、TCP 服务器这四种类型的通讯设备，下拉选择框即可选择对应的设备添加

1 串口

接入串口硬件设备后，在左侧配置区域选择对应串口端口，根据外接设备通讯规格设置波特率、数据位、奇偶校验、停止位，同时填写通讯超时时间。全部参数设置完成后点击连接按钮，建立串口通讯链路，连接成功后即可开展数据交互。

界面右侧分为接收区与发送区两大模块。接收区实时展示串口传回的所有数据，点击清理数据可清空窗口内历史报文；在发送区输入需要下发的指令内容，点击发送按钮即可向串口设备传输数据。

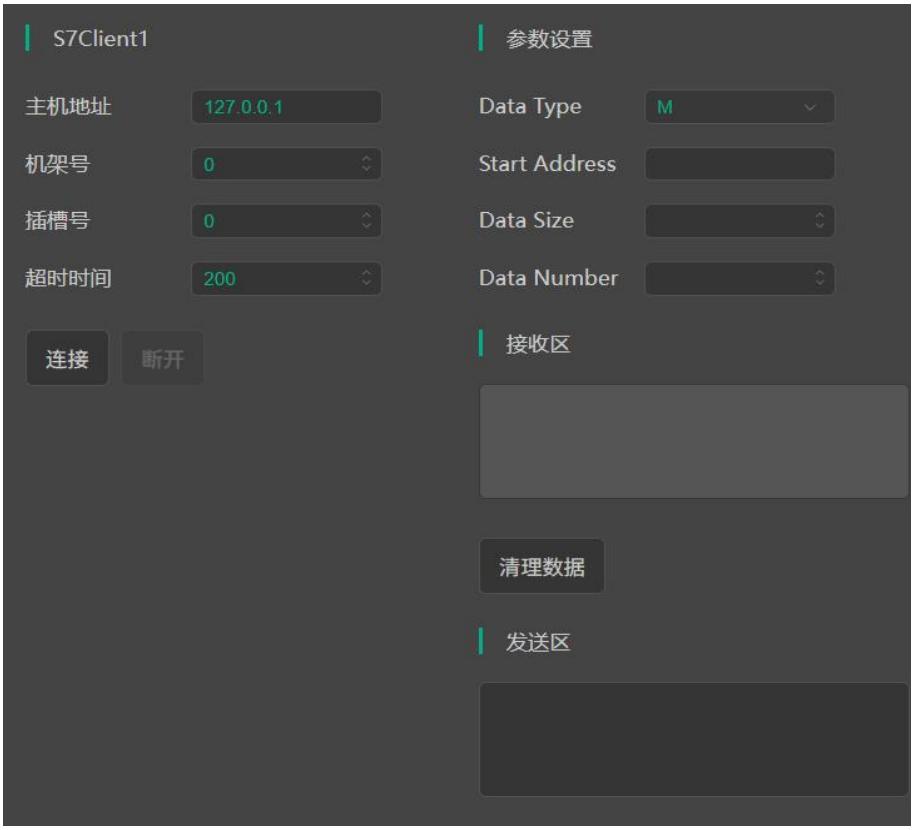
调试结束后可点击断开释放串口资源，平台配套工具可辅助完成串口通讯调试、报文查看等相关操作。使用时需保证串口两端通讯参数保持一致，避免出现乱码、无数据应答等通讯异常问题。

2 S7client

将设备通过网线接入后，在界面左侧配置栏填写西门子 S7 设备 IP 地址、机架号、插槽号，并设置通讯超时时长，全部参数填写完成后点击连接按钮，即可建立 S7 Client 通讯链路。通讯连接成功后，需在界面右侧配置数据交互参数，包含目标数据区、起始地址、数据长度、数据块数

量等内容，参数配置完成后即可启用接收、发送两大数据交互模块。

接收区会实时展示设备回传的全部报文数据，点击清理按钮可清空窗口内历史记录；在发送区编辑下发指令后，点击发送按钮即可向设备传输报文。整套通讯配置、收发数据逻辑均支持通过脚本函数自定义调整，相关规则可查阅本文档执行脚本开发章节。

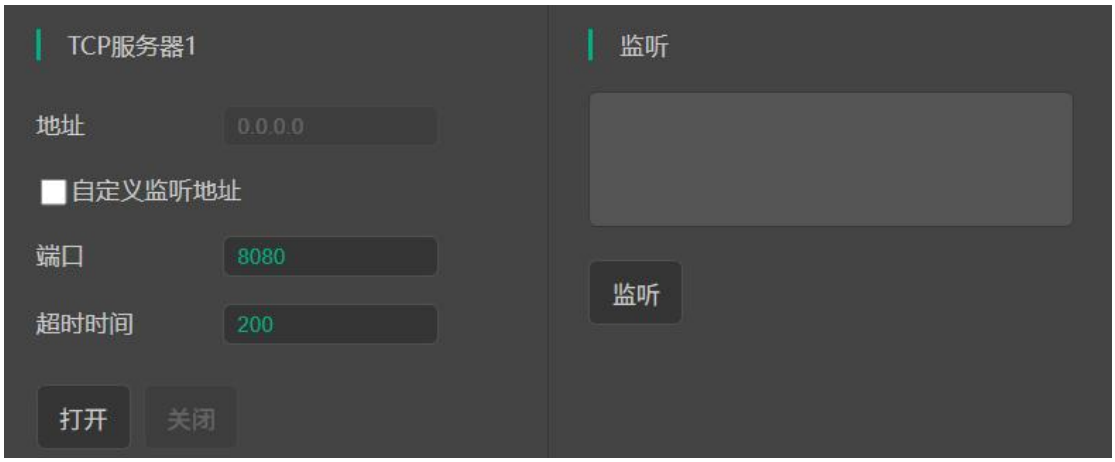


3 TCP 客户端

将设备通过网线完成接线接入后，在左侧 TCP 客户端配置栏填写目标服务端 IP 地址、通信端口与连接超时时间，建立连接成功后即可正常使用配套工具；在右侧发送区编辑待下发指令，点击发送按钮便可向设备传输报文，完成通信测试。

4 TCP 服务端

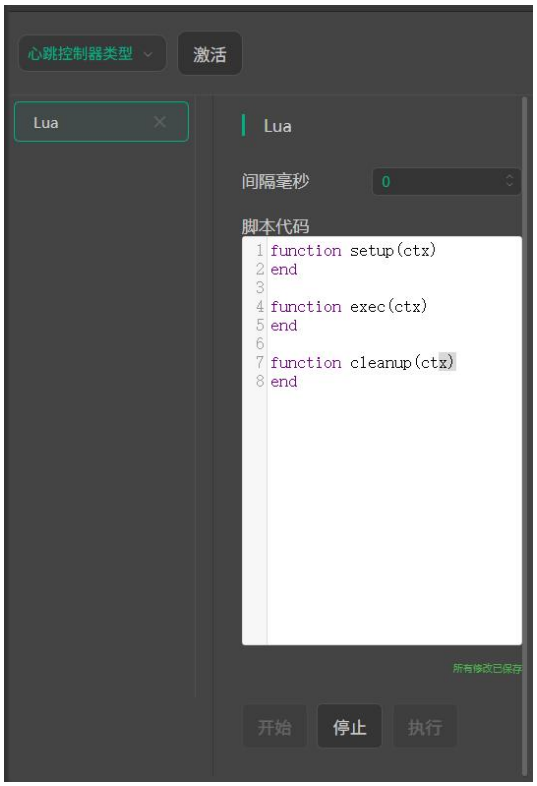
可将本平台切换为 TCP 服务端模式，按需配置监听地址、通信端口与连接超时时间；若需指定网卡 IP 可勾选自定义监听地址，参数设置完成后点击打开启动服务，即可正常使用配套调试工具，右侧监听区域将实时持续接收所有客户端上传报文。



九、 心跳控制器

心跳控制器用于视觉平台与 PLC 建立通信链路后，维持双方心跳信号交互；开启控制器后，系统将按预设间隔定时执行配套 Lua 脚本，该脚本独立运行于单独线程，不会干扰主程序工作。

操作时点击对应图标 即可进入心跳控制器设置界面，再切换至编辑模式；新增心跳控制器会弹出参数配置窗口，完成运行间隔设置后，可在脚本代码内编写心跳交互逻辑，脚本详细用法可查阅本书执行脚本开发相关章节。



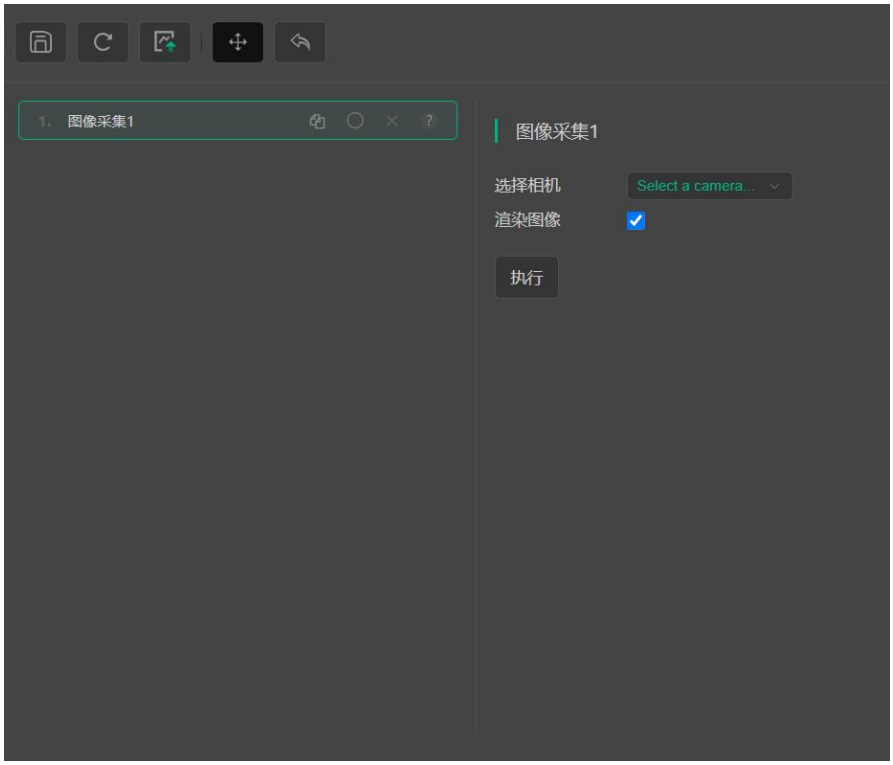
十、 工具说明

本章主要介绍平台各类工具的操作使用方法。

1 图像模块

1.1 图像采集

本工具用于采集工业相机单帧图像，每执行一次仅抓拍一帧画面，该工具为顺序执行，跟着本轮工序的工具由上至下顺序运行。界面示意如下。

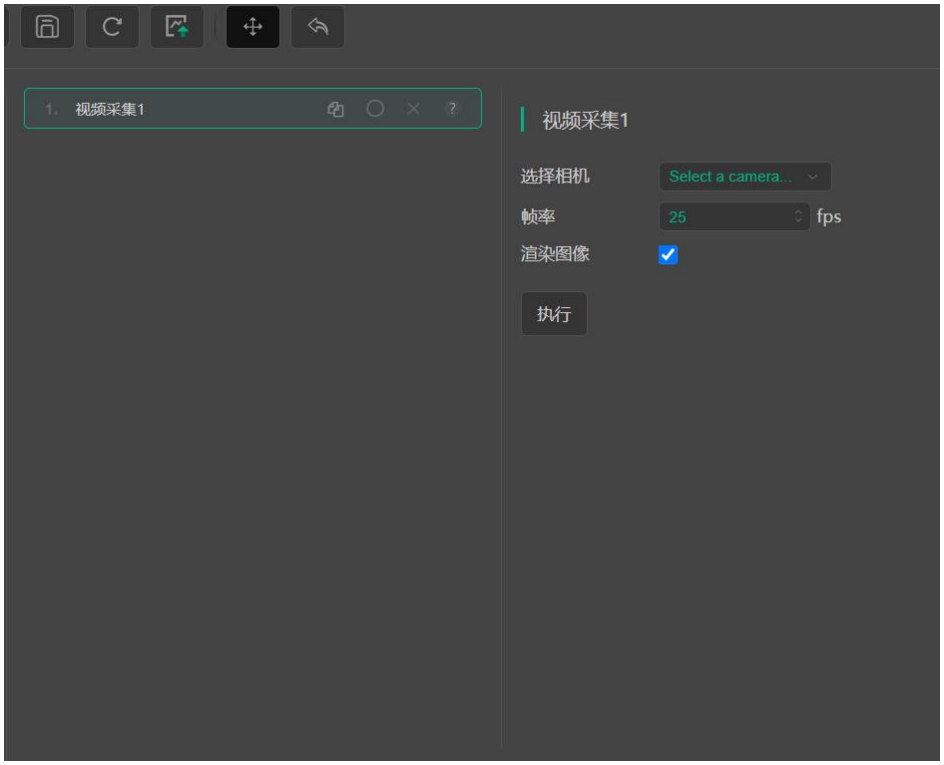


❖ 基本参数

编号	参数项	默认值	取值范围	注释
1	选择相机	Null		选用一个已完成参数配置的相机设备
2	渲染图像	勾选	勾选/不勾选	选择是否将采集图像渲染展示至运行界面。若取消勾选画面显示选项，图像采集完成后将不在运行界面呈现

1.2 视频采集

本工具用于采集工业相机图像，运行后将按照预设帧率向上位机传输连续帧数据流，不受工序顺序执行逻辑约束。算法处理时自动选取最新采集帧运算，采用视频采集模式可避免运行界面画面帧率变高，但可能会导致渲染框稍延迟于图片。界面示意如下。

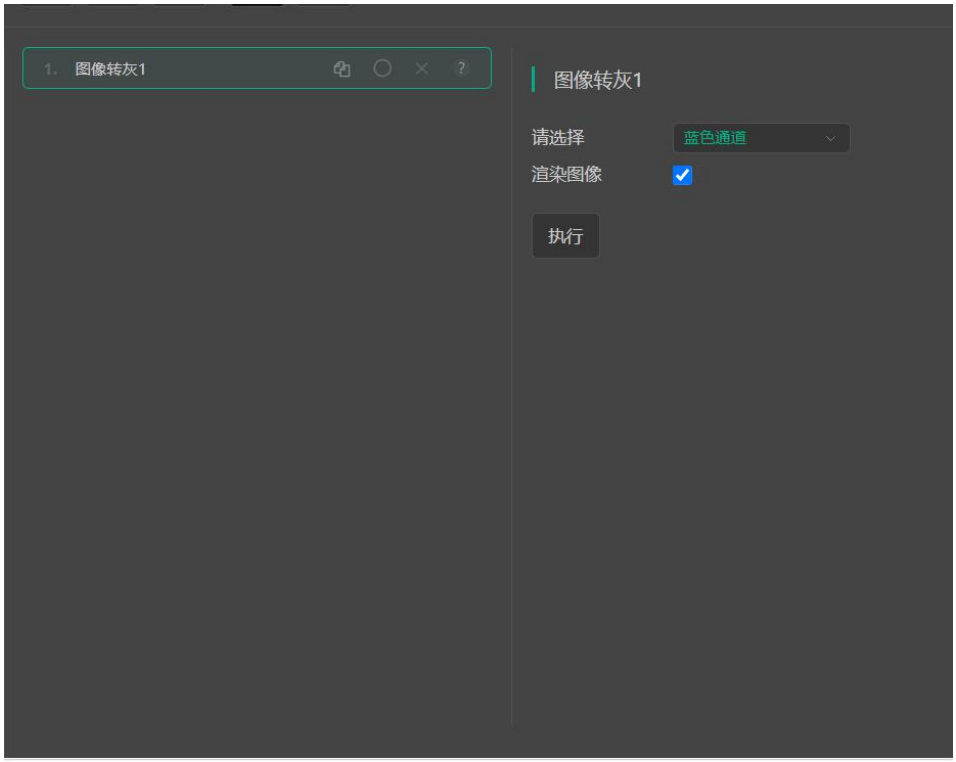


❖ **基本参数**

编号	参数项	默认值	取值范围	注释
1	选择相机	Null		选用一个已完成参数配置的相机设备
2	帧率	25	≥1	设置界面实时展示视频输出帧率（输出帧率通常不大于相机硬件采集帧率），若帧率大于相机实际采集帧率，将会采集重复图片
3	渲染图像	勾选	勾选/不勾选	选择是否将采集图像渲染展示至运行界面。若取消勾选画面显示选项，图像采集完成后将不在运行界面呈现

1.3 图像转灰

图像转灰工具能够依据颜色通道对当前画面图像进行灰度转换。界面示意如下。



❖ 基本参数

编号	参数项	默认值	取值范围	注释
1	转换方式	蓝色通道	蓝色通道	使用蓝色通道作为灰度图像。
			绿色通道	使用绿色通道作为灰度图像。
			红色通道	使用红色通道作为灰度图像。
			最小值	使用三个颜色通道中的最小值作为灰度图像。
			最大值	使用三个颜色通道中的最大值作为灰度图像。
			平均值	使用三个颜色通道中的平均值作为灰度图像。
2	渲染图像	勾选	勾选/不勾选	选择是否将采集图像渲染展示至运行界面。若取消勾选画面显示选项，图像采集完成后将不在运行界面呈现

2 测量模块

本模块为平台预留功能，当前版本暂未完成测试验证，暂不开放使用，相关配置与操作说明待后续版本更新补充。

3 检测模块

3.1 斑点分析

斑点分析工具用于检测图像中目标的数量和几何特征（面积、位置、方位、长轴和短轴），目标对象的定义为二值图像中黑色或白色像素连通区域，该功能要求图像背景均匀并亮度和目标的亮度区别比较明显。工具操作页面如下。

❖ 基本参数

编号	参数项	默认值	取值范围	注释

1	斑点类型	白色区域	白色区域/黑色区域	用于指定图像中待提取识别的目标色块类别，可选择白色区域或黑色区域，算法会依据选定的颜色阈值，筛选出对应明暗属性的连通色块作为候选斑点。
2	修剪斑块面积	10	≥0	作为斑点过滤的最小面积阈值，单位为像素，所有统计面积小于等于该数值的微小杂点、噪点斑块会被直接剔除，常用于过滤灰尘、画面噪点等无意义小斑点，减少后续计算干扰。
3	连通性	四联通	四联通	定义像素间判定为同一斑点的邻接规则，四连通仅上下左右 4 个相邻像素算作连通区域，轮廓规整、计算更快。
			八联通	八联通在四联通 4 个正交邻域的基础上，额外纳入四个斜向对角像素作为连通邻域。更容易粘连细碎色块，适合提取不规则、边缘碎片化的斑点。
4	孔洞面积	0	≥0	用于过滤斑点内部的闭合空洞，当斑点内部孔洞面积不大于该设定值时，会将孔洞填充合并至斑点主体，数值为 0 时代表不做孔洞填充处理，可用来消除斑点内部小黑点、空隙带来的面积与轮廓误差。
5	固定角度	0	-180° ~ 180°	仅在外接矩形类型选定固定角度模式时生效，设定一个基准倾斜角度，算法会以该角度为基准生成斑点外接矩形，统一所有斑点矩形的偏转角度，方便批量比对长宽、方位，多用于工件姿态规整定位场景。
6	外接矩形类型	固定角度	固定角度	仅在固定角度参数生效时使用，以人为设定的统一倾斜角度绘制所有斑点的外接矩形，强制所有检测目标矩形角度一致。

			主轴方向	按照斑点的主轴椭圆的角度来生成斑点的外接矩形。
			最小面积	按照斑点的最小面积外接矩形来生成斑点的外接矩形。
7	分割阈值	128	0~255	为灰度图像二值化分割阈值，灰度值高于该数值则划分为白色区域、低于该数值则划分为黑色区域，用于区分前景斑点与背景，阈值调高会筛选出更亮的目标，调低可识别偏暗斑点，是斑点提取最基础的分割控制参数

❖ **排序器**

排序器用于斑点识别完成后，选定斑点特征作为依据，对斑点集合进行有序重排。

编号	参数项	默认值	取值范围	注释
1	排序主体	Null	详见斑点特征参数	选定用来排序的斑点特征指标
2	排序顺序	Null	升序	按照特征数值从小到大依次排列
			降序	按照特征数值从大到小依次排列

❖ **过滤器**

过滤器用于筛选符合、或不符合特征阈值的斑点，可多次新增过滤器叠加约束条件，设定范围后执行即可完成斑点筛选。

编号	参数项	默认值	取值范围	注释
1	过滤主体	Null	详见斑点特征参数	选定用来排序的斑点特征指标
2	最小值	0	≥1	划定筛选区间下限。
3	最大值	0	≥1	划定筛选区间上限。
4	移除过滤主体	保留		移除后会过滤掉符合过滤器条件的斑点，保留则会保留符合过滤器条件的斑点

❖ **斑点特征参数**

此处用来说明可用的一些斑点特征。

编号	特征	注释
1	孔洞数量	表示检测到的目标对象上的孔洞数量
2	斑块面积	表示目标对象的面积
3	重心 X	表示目标对象的重心的 X 坐标

4	重心 Y	表示目标对象的重心的 Y 坐标
5	主轴角度	表示目标对象主轴的角度
6	主轴长度	表示目标对象主轴的长度
7	次轴长度	表示目标对象次轴的长度
8	主轴比	主轴比=次轴长度/主轴长度
9	凸包面积	表示目标对象的面积（像素数量）
10	凸包面积比	凸包面积比=斑块面积/凸包面积
11	矩形中心 X	表示目标对象外接矩形的中心 X 坐标
12	矩形中心 Y	表示目标对象外接矩形的中心 Y 坐标
13	矩形角度	表示目标对象外接矩形的角度
14	矩形宽度	表示目标对象外接矩形的宽度
15	矩形高度	表示目标对象外接矩形的高度
16	矩形纵横比	矩形纵横比=矩形高度/矩形宽度。
17	矩形面积比	矩形面积比=斑块面积/(矩形宽度*矩形高度)。

3.2 多轮廓匹配

多轮廓匹配工具提取模板边缘轮廓作为匹配基准，在前置工具输出图像内检索形状相似目标，支持多模板类别同时检索。工具操作页面如下。

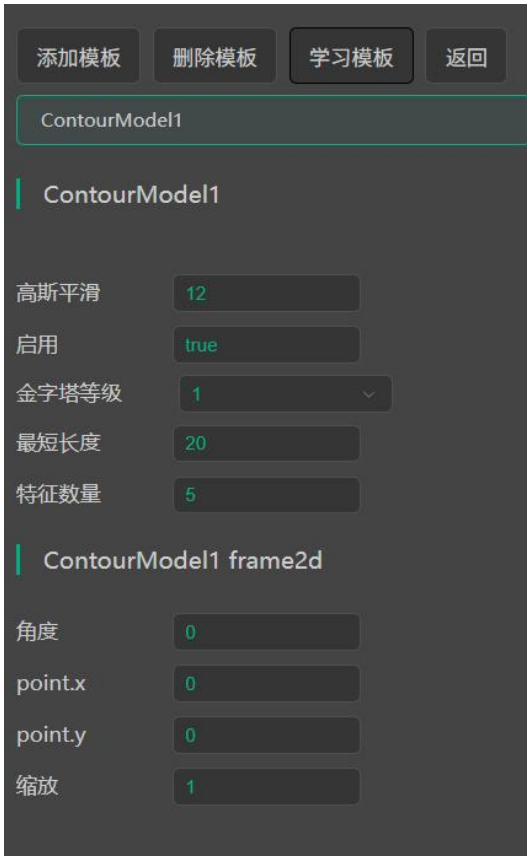
❖ 基本参数

编号	名称	默认值	取值范围	说明
1	搜索数量	1	1~1000	设置最大搜索目标数量，识别结果的数量将小于或等于该数量，搜索目标数量越多越耗时。
2	最小分数	80	0~100	设置待识别目标的最佳匹配度评分，在保证能搜索到的情况下尽量将分数值调大，分数值越大搜索速度越快。
3	最小角度	0	-360~360	设置待识别目标可能存在的最小角度，角度为目标相对模板的角度。
4	最大角度	0	-360~360	设置待识别目标可能存在的最大角度，角度为目标相对模板的角度。
5	最小比例	100	10~1000	设置待识别目标可能存在的最小比例（百分比），比例为目标相对于模板的比例。
6	最大比例	100	10~1000	设置待识别目标可能存在的最大比例（百分比），比例为目标相对于模板的比例。

7	重叠距离	20	0~1000	当两个目标的中心位置之间的距离小于该值（像素单位），则认为存在重叠，将会过滤掉分数低的目标。
8	特征强度	低	低	适合有细节特征且边缘平滑的图像。
			高	适合有噪点或边缘不平滑的图像，抗干扰能力更强。
9	目标极性	一致	一致	待识别目标的边缘方向跟模板一致。
			相反	待识别目标的边缘方向跟模板相反。
			任意	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反色的情况，该模式会花费更多的时间。
10	分割阈值	2	0~255	设置匹配目标边缘特征的最小梯度阈值，用于过滤掉部分模糊和低对比度的边缘特征。
11	金字塔等级	1	Layer0~9	搜索算法到达第几层停止，值越小精度越高，但速度越慢，在不要求精度的情况下可加大。
12	定位精度	6	Level0~9	设 0 为不使用亚像素插值，大于 0 为使用最小二乘法拟合进行亚像素插值的次数，每次插值都会在上一次的结果中进行再次拟合，从而提高定位精度，设的值越大将会消耗更多的搜索时间。
13	搜索速度	6	Level0~9	搜索速度值越大速度越快，但可能会导致识别率降低，需要根据实际情况来设置。

❖ 模板配置

开展轮廓匹配检测前，需完成模板学习与配置工作，点击配置模板并添加模板进入编辑界面，按需设置参数，在图像画面中圈选目标区域绘制 ROI 模板范围，核对设置无误后点击学习模板，完成轮廓模板特征采集，即可调用该模板执行后续轮廓匹配检测，界面内还可按需选用模板。

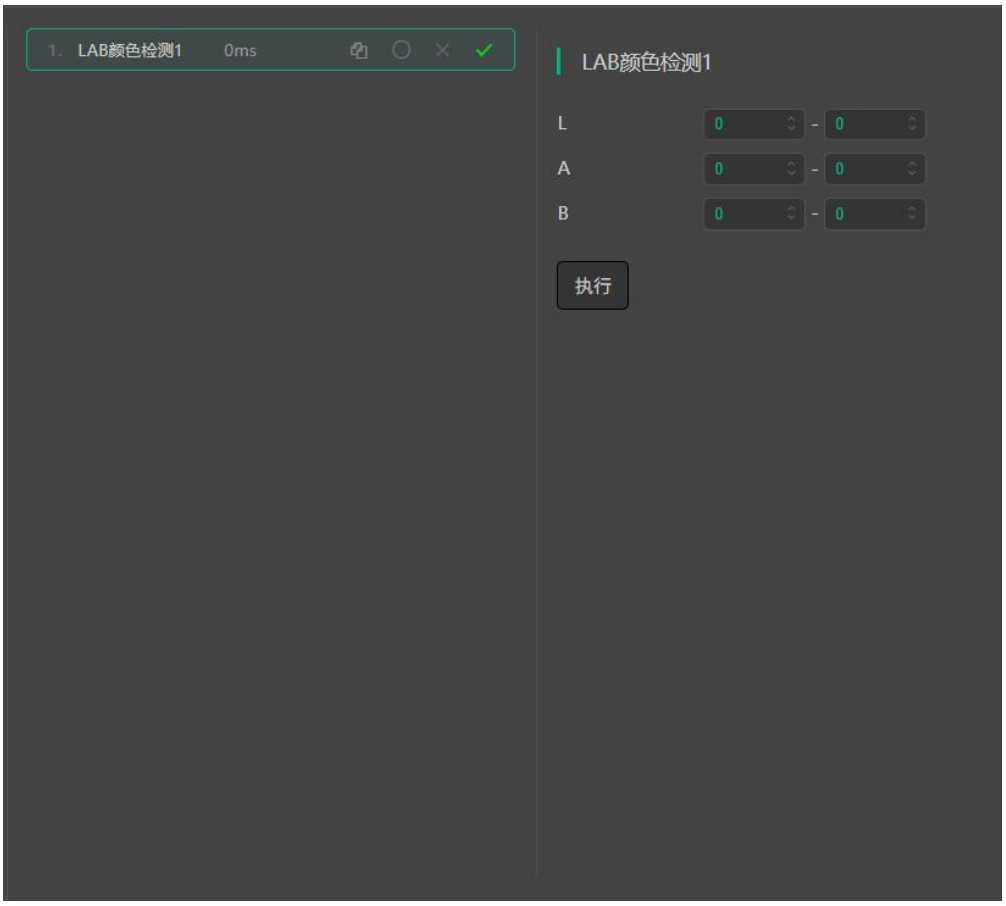


编号	名称	默认值	取值范围	说明
ContourModel				
1	高斯平滑	12	1~100	设置提取轮廓特征的平滑程度，值越小能够提取的特征越精细，值越大越适合模糊和噪点多的图像，
2	金字塔层数	1	1~9	可手动设置金字塔最大层数，最终生成的层数小于或等于该值。
3	轮廓长度	20	0~999999	轮廓长度（轮廓点数量）参数用于过滤长度小于该值的轮廓特征。
4	特征数量	5	1~100	待提取特征点数量所占学习区域总面积的百分比，值越大能够提取的边缘特征越多。
ContourModel frame2d				
1	基准角度	0	-360~360	代表轮廓的基准角度

2	基准坐标 x	0	≥0	代表轮廓的基准坐标 x
3	基准坐标 y	0	≥0	代表轮廓的基准坐标 y

3.3 LAB 颜色检测

LAB 颜色检测工具用于检测 ROI 图像区域内的 LAB 颜色参数，其中 L 代表亮度通道，A 代表红- 绿色度通道，B 代表黄- 蓝色度通道，工具可输出该图像区域 L、A、B 三个通道的检测结果；同时支持限定 L、A、B 各通道的取值范围，当区域内识别到的颜色落在设定范围内时，输出 accepted 为 true，反之则为 false。

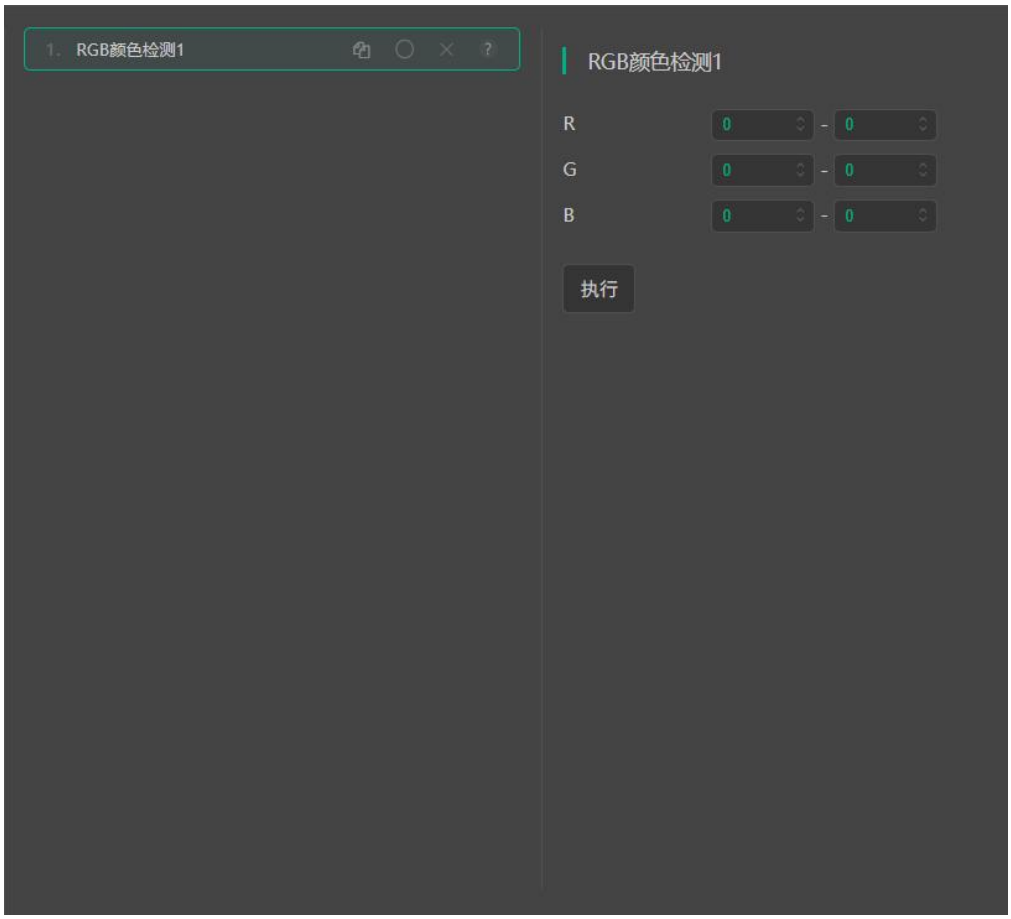


❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	L	0	0~100	亮度通道，0 代表纯黑，100 代表纯白，控制图像明暗信息
2	A	0	-128~127	红- 绿色度通道，负值偏向绿色，正值偏向红色
3	B	0	-128~127	黄- 蓝色度通道，负值偏向蓝色，正值偏向黄色

3.4 RGB 颜色检测

RGB 颜色检测工具用于检测 ROI 图像区域内的 RGB 颜色参数，其中 R 代表红色通道，G 代表绿色通道，B 代表蓝色通道，工具可输出该图像区域 R、G、B 三个通道的检测结果；同时支持限定 R、G、B 各通道的取值范围，当区域内识别到的颜色落在设定范围内时，输出 accepted 为 true，反之则为 false。

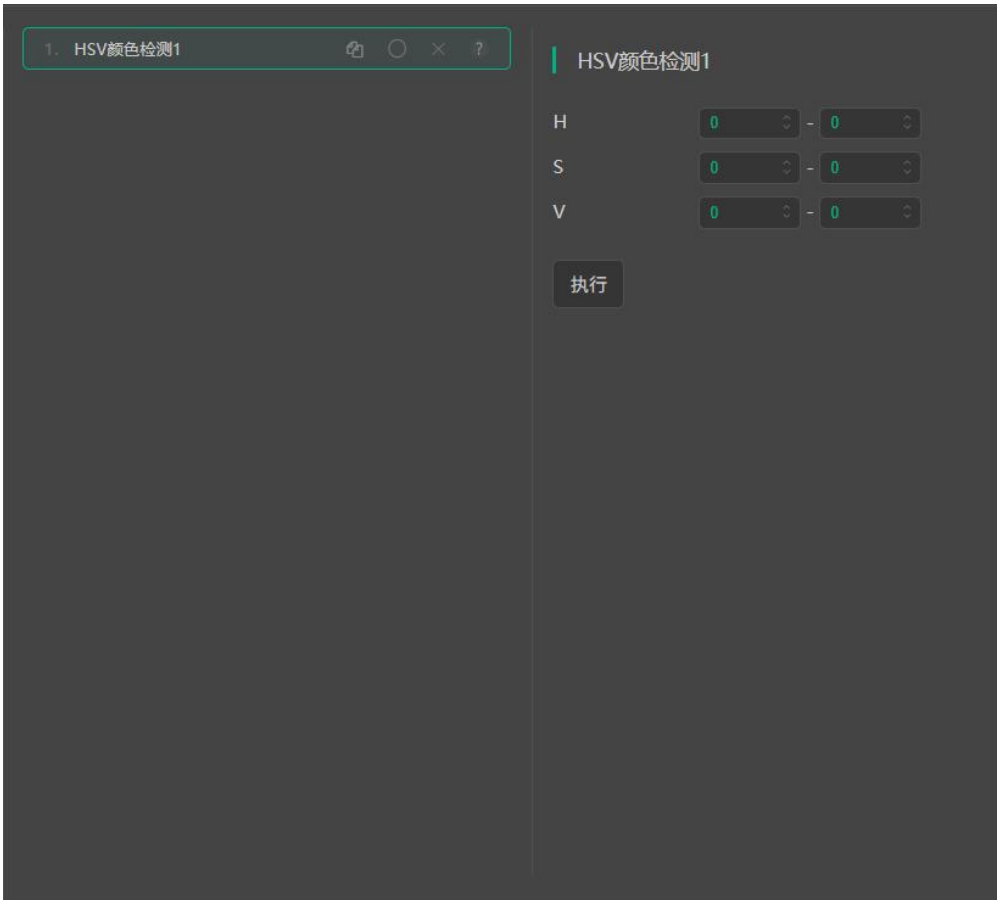


❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	R	0	0~255	红色通道，数值越大红色分量越强，0 为无红色，255 为红色最大强度
2	G	0	0~255	绿色通道，数值越大绿色分量越强，0 为无绿色，255 为绿色最大强度
3	B	0	0~255	蓝色通道，数值越大蓝色分量越强，0 为无蓝色，255 为蓝色最大强度

3.5 HSV 颜色检测

HSV 颜色检测工具用于检测 ROI 图像区域内的 HSV 颜色参数，其中 H 代表色相通道，S 代表饱和度通道，V 代表明度通道，工具可输出该图像区域 H、S、V 三个通道的检测结果；同时支持限定 H、S、V 各通道的取值范围，当区域内识别到的颜色落在设定范围内时，输出 accepted 为 true，反之则为 false。



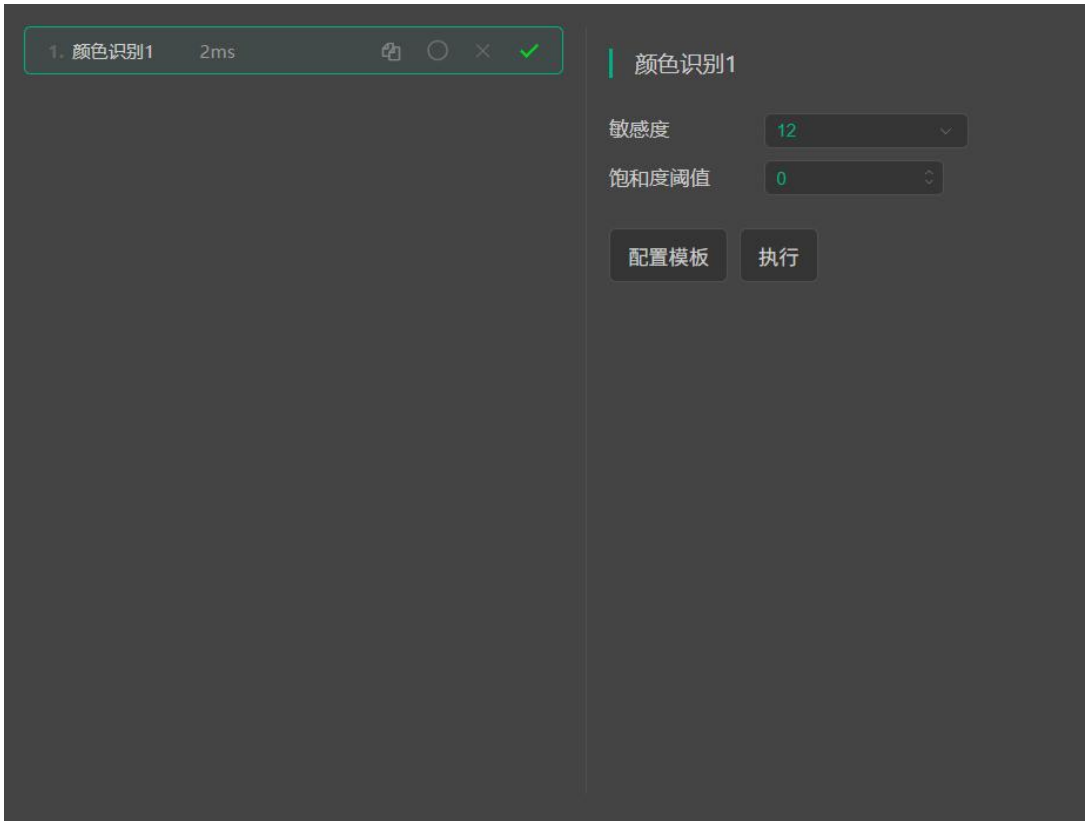
❖ 基本参数

编号	名称	默认值	取值范围	说明
1	H	0	0~360	色相，代表颜色种类，0° 代表红色、60° 代表黄色、120° 代表绿色、180° 代表青色、240° 代表蓝色、300° 代表品红，循环 360° 回到红色
2	S	0	0~100	饱和度，色彩纯净度；0 为灰度无彩色，数值越大颜色越鲜艳，100 为最饱和

3	V	0	0~100	明度，颜色明亮程度；0 代表纯黑，100 代表最大亮度，不改变色相与饱和度

3.6 颜色识别

颜色识别基于 HSV 色彩空间，该工具会将图像中需识别的部分与已添加模板的色彩做比对，并最终输出匹配的模板以及匹配分数。支持使用脚本函数获取相应结果，详情可查阅脚本开发章节。



❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	敏感度	12	12/24/48/64	控制色相 H 的允许波动区间。数值越大，识别宽容度越高；数值越小，颜色筛选条件越严格。
2	饱和度阈值	0	0~255	用于过滤灰度、黑白以及低饱和度杂点；仅当像素饱和度≥该阈值时，方可参与颜色匹配。

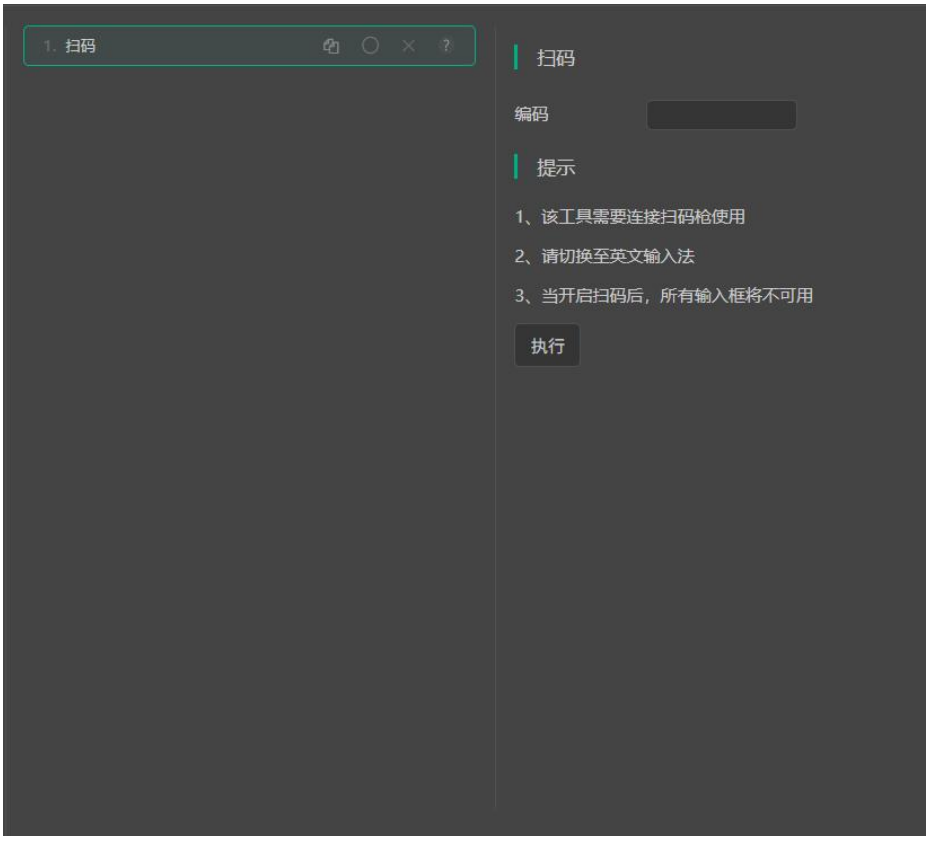
❖ **模板配置**

进行颜色识别前需完成模板配置与学习。点击配置模板、添加模板进入编辑界面，设置参数并在图像中圈选 ROI 模板区域，确认后学习模板。系统支持多个模板，工具运行时逐一完成比对，输出相似度最高的模板名称及匹配得分。

4 条码模块

4.1 扫码

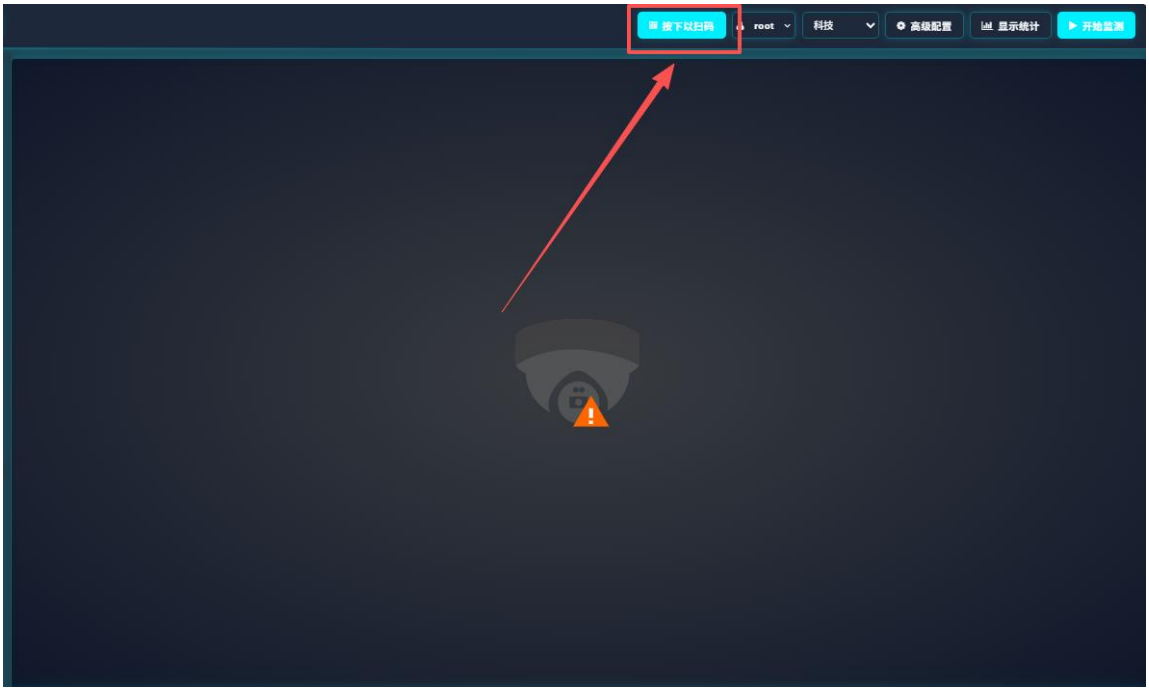
扫码工具用于接收扫码枪读取的条码结果，参数配置界面如下。

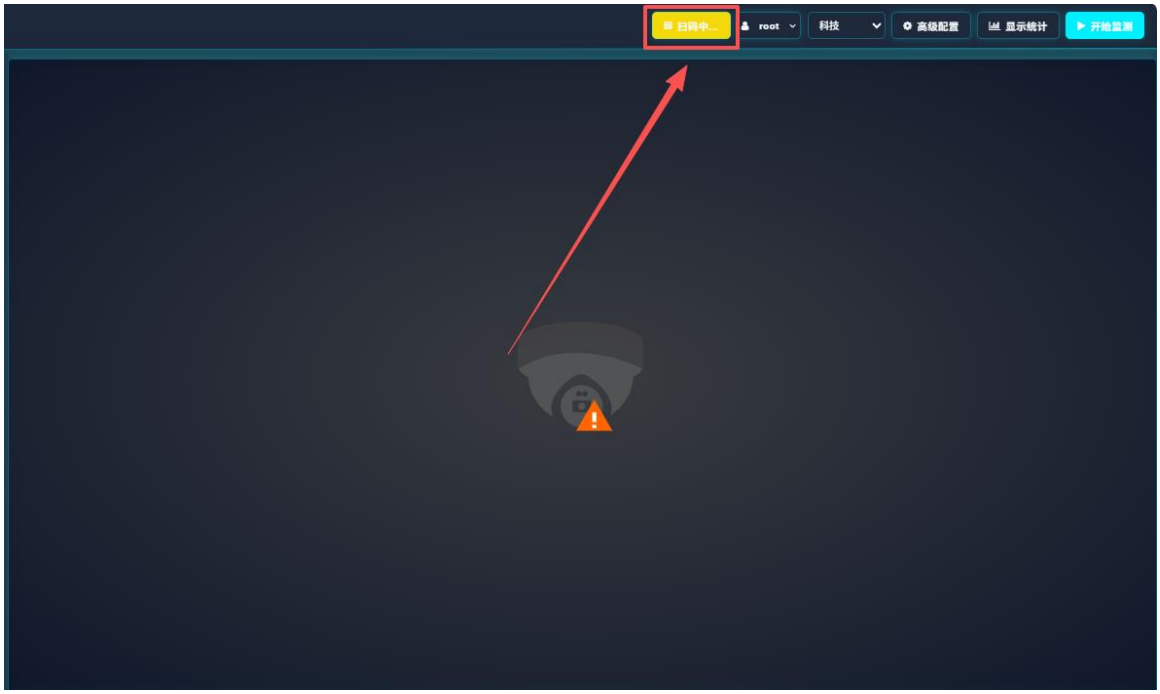


扫码完成后，解析得到的编码结果将显示在编码输入框内。当作业流程中任一工序存在扫码工具时，运行界面会展示【按下以扫码】按钮。使用扫码枪前，需要点击该按钮；按钮变为黄色，代表进入扫码中状态，只有处于该状态，软件才可以正常获取扫码枪返回的数据，否则会出现扫码结果获取失败。

使用提示：

1. 本工具需要外接扫码枪设备才可正常使用；
2. 使用时请切换至英文输入法；
3. 开启扫码状态后，其余输入框将不可操作。





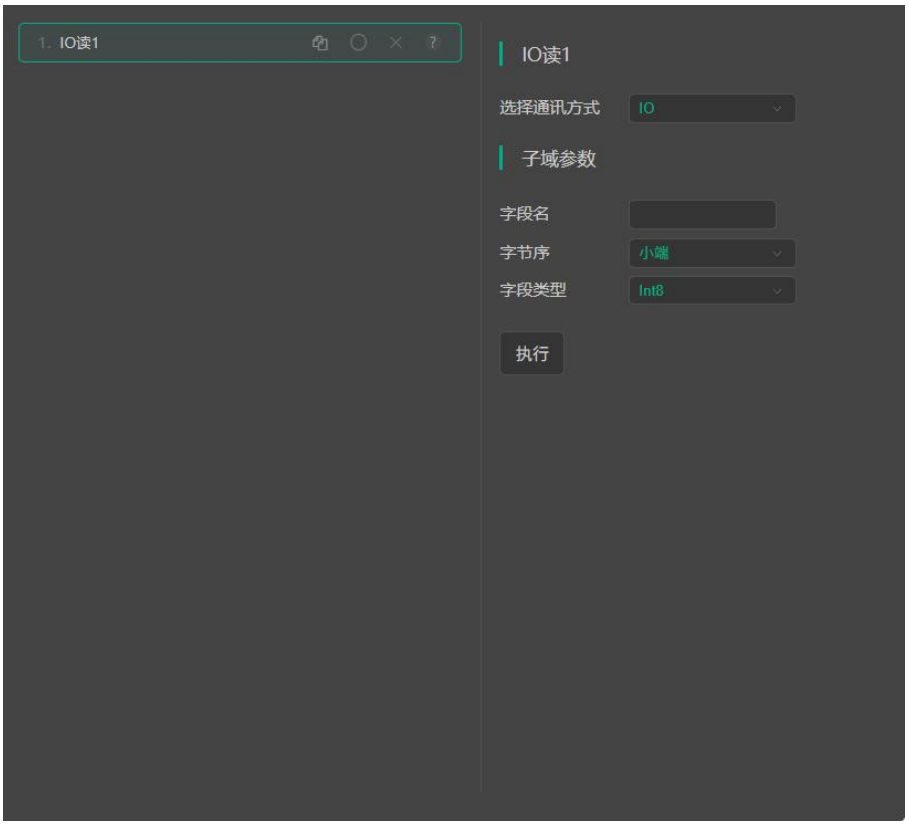
5 逻辑模块

本模块为平台预留功能，当前版本暂未完成测试验证，暂不开放使用，相关配置与操作说明待后续版本更新补充。

6 通信模块

6.1 IO 读

IO 读工具用于接收数据工具可用于接收串行口或以太网的数据，使用前需在 IO 设备设置中添加 IO 设备，参数配置界面如下



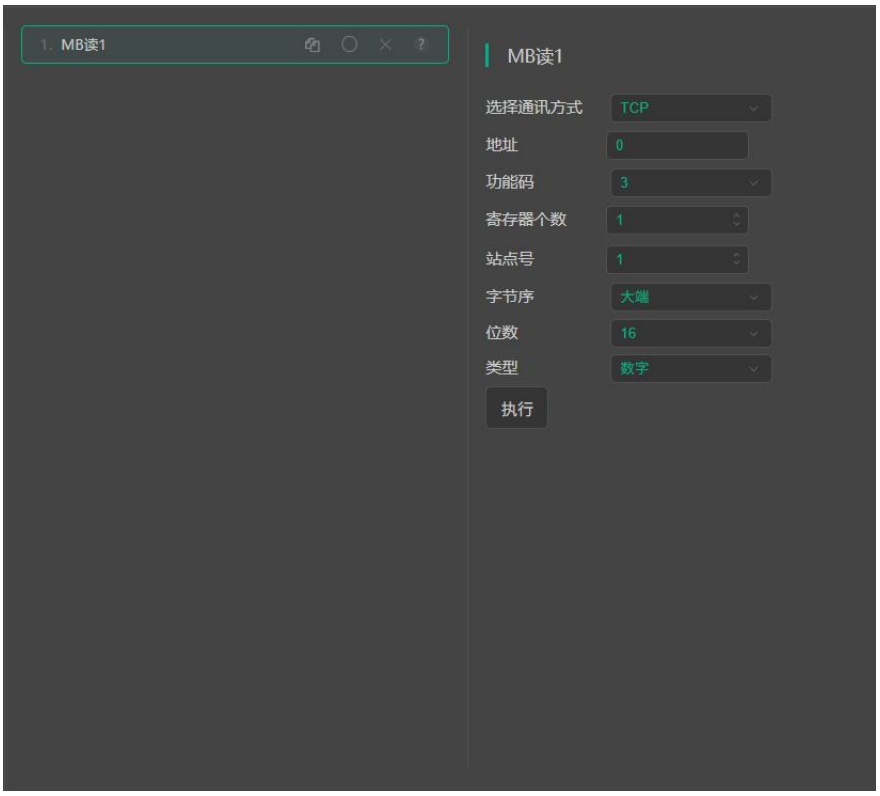
❖ 基本参数

编号	名称	默认值	取值范围	说明
1	选择通讯方式	Null		下拉选择已建立的通讯通道，标识数据来源，例如 IO，对应底层串口 / 以太网链路实例。
子域参数				
1	字段名	Null		自定义输出字段名称，解析完成后，该名称用于脚本获取读取结果。
2	字段序	小端	小端	高字节在前，低字节在后。
			大端	低字节在前，高字节在后
3	字段类型	Int8	Int8	有符号 8 位整数，占 1 字节，范围- 128 ~ 127

			UInt8	无符号 8 位整数，占 1 字节，范围 0 ~ 255
			Int16	有符号 16 位整数，占 2 字节
			UInt16	无符号 16 位整数，占 2 字节
			Int32	有符号 32 位整数，占 4 字节
			UInt32	无符号 32 位整数，占 4 字节
			Int64	有符号 64 位整数，占 8 字节
			UInt64	无符号 64 位整数，占 8 字节
			Float	32 位单精度浮点数，占 4 字节
			Double	64 位双精度浮点数，占 8 字节
			UInt8+String	以 UInt8 数值作为字符串长度，后续字节读取对应长度字符串
			UInt16+String	以 UInt16 数值作为字符串长度，读取对应长度字符串
			UInt32+String	以 UInt32 数值作为字符串长度，读取对应长度字符串
			UInt64+String	以 UInt64 数值作为字符串长度，读取对应长度字符串
			String+Line Flag	读取字符串，换行标记\n 作为结束符
			String+Return+Line Flag	读取字符串，回车换行\r\n 作为结束符
			String	读取固定长度原始字符串

6.2 MB 读

MB 读工具使用 Modbus 协议，本程序作为主站，从下位机（从站）中读取寄存器的值。该工具支持使用脚本进行数据读取，详情可查阅执行脚本开发章节。



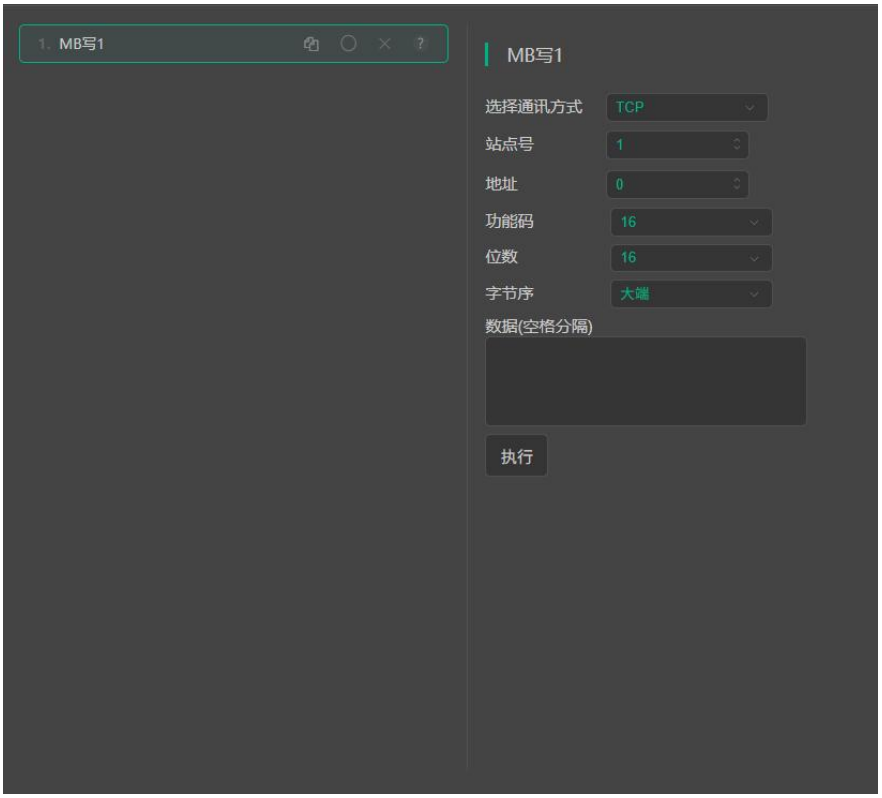
❖ 基本参数

编号	名称	默认值	取值范围	说明
1	选择通讯方式	Null		选择已创建的 Modbus 通讯通道，支持 TCP（Modbus- TCP 以太网）、RTU（串口 485）两种链路。
2	地址	0		目标起始寄存器 / 线圈地址，填写十进制地址，注意区分协议 0 起始还是 1 起始；部分设备手册地址需要- 1 偏移。
3	功能码	3	1~4	Modbus 标准功能码：01：读线圈 (DO)；02：读离散输入 (DI)；03：读保持寄存器；04：读输入寄存器。
4	寄存器个数	1		要读取的连续寄存器 / 线圈数量。寄存器模式：1 个 = 2 字节；线圈模式代表读取 bit 点位数量。
5	站点号	1		从站设备 ID，RTU 模式必须和从站地址匹配；TCP 模式大部分设备也需要填写站点号。
6	字节序	大端	小端/大端	多字节数值解析顺序，工业 Modbus 标准默认为大端(低字节在前，高字节在后)；仅对 16 位及以上数据生效(即

				读寄存器时生效）。
7	位数	16	16/32	解析输出位宽，仅在读寄存器时生效
8	类型	数字	数字/文本	选择输出数据类型

6.3 MB 写

MB 写工具使用 Modbus 协议，本程序作为主站，向下位机（从站）的寄存器写入数据。该工具支持使用脚本进行数据写入，详情可查阅执行脚本开发章节。



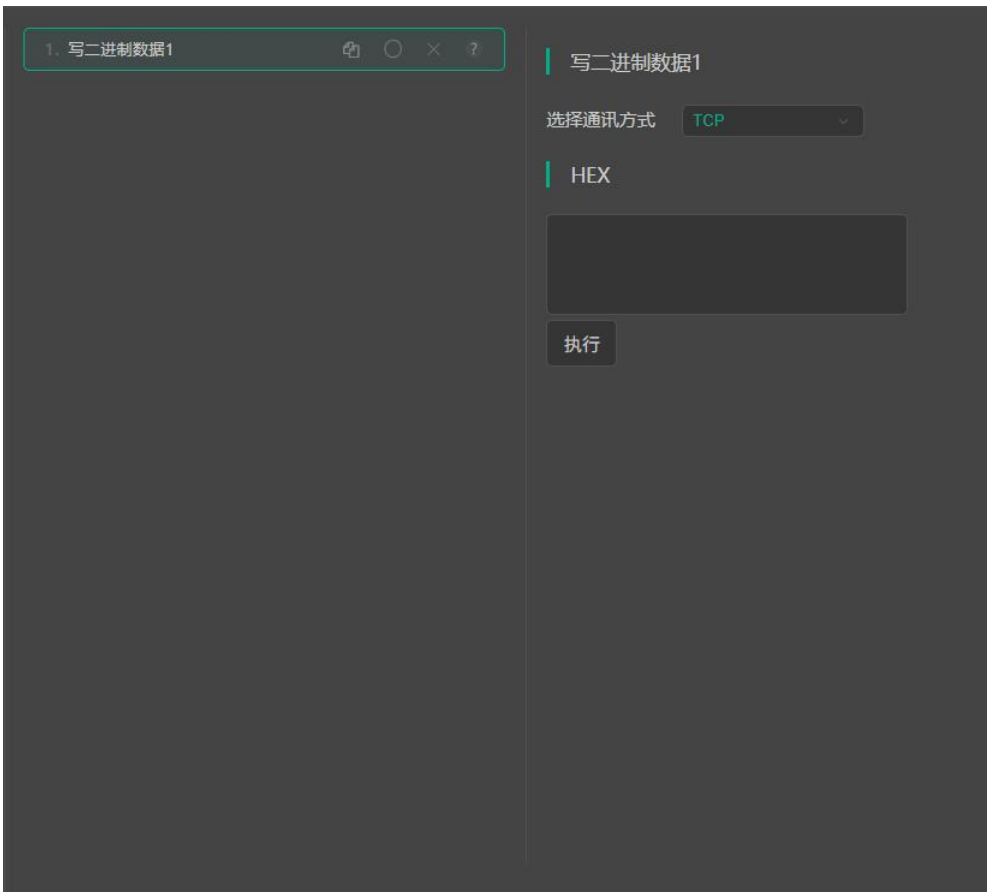
❖ 基本参数

编号	名称	默认值	取值范围	说明

1	选择通讯方式	Null	TCP/RTU	选择已创建的 Modbus 通讯通道，支持 TCP（Modbus- TCP 以太网）、RTU（串口 485）两种链路。
2	站点号	1	1- 247	Modbus 从站设备地址号，对应 PLC / 下位机从站 ID。
3	地址	0	0- 65535	保持寄存器起始地址，为 0 起始偏移地址。
4	功能码	06	05/06/16	05：写单线圈；06：写单个保持寄存器；16：批量写保持寄存器。
5	位数	16	16/32	仅功能码为 16 时生效；16：单寄存器存储；32：双寄存器存储 32 位数据。
6	字节序	大端	大端/小端	仅功能码为 16 时生效；设置数据高低字节存储顺序。
7	数据(空格分隔)	Null		仅功能码为 16 时生效；待写入寄存器数据，多组数据使用空格隔开

6.4 写二进制数据

写二进制数据工具，用于通过串口或者 TCP 网络通道，向外发送一串二进制原始数据。

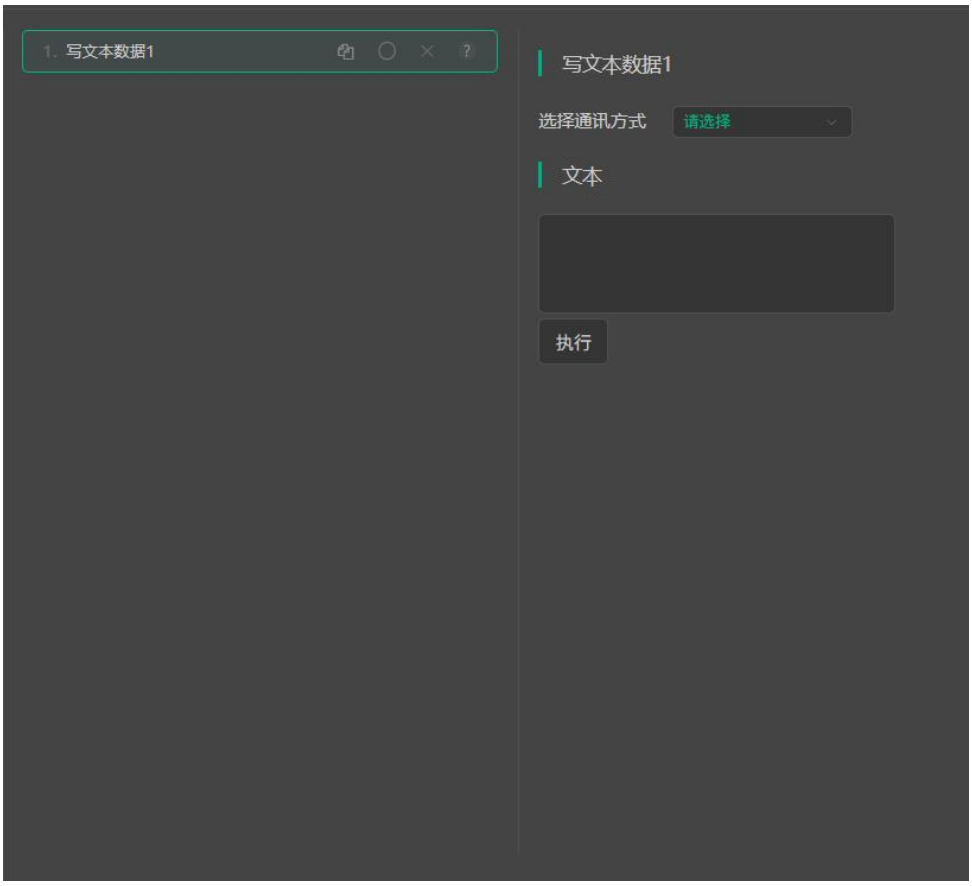


❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	选择通讯方式	Null		选择数据输出的通信通道，支持 TCP 网络通信或者串口通信
2	HEX	Null		待发送的十六进制指令字符串，工具内部自动解析转换为原始二进制字节流进行发送，禁止输入普通 ASCII 文本

6.5 写文本数据

写文本工具，用于通过串口或者 TCP 网络通道，向外发送一串文本数据。

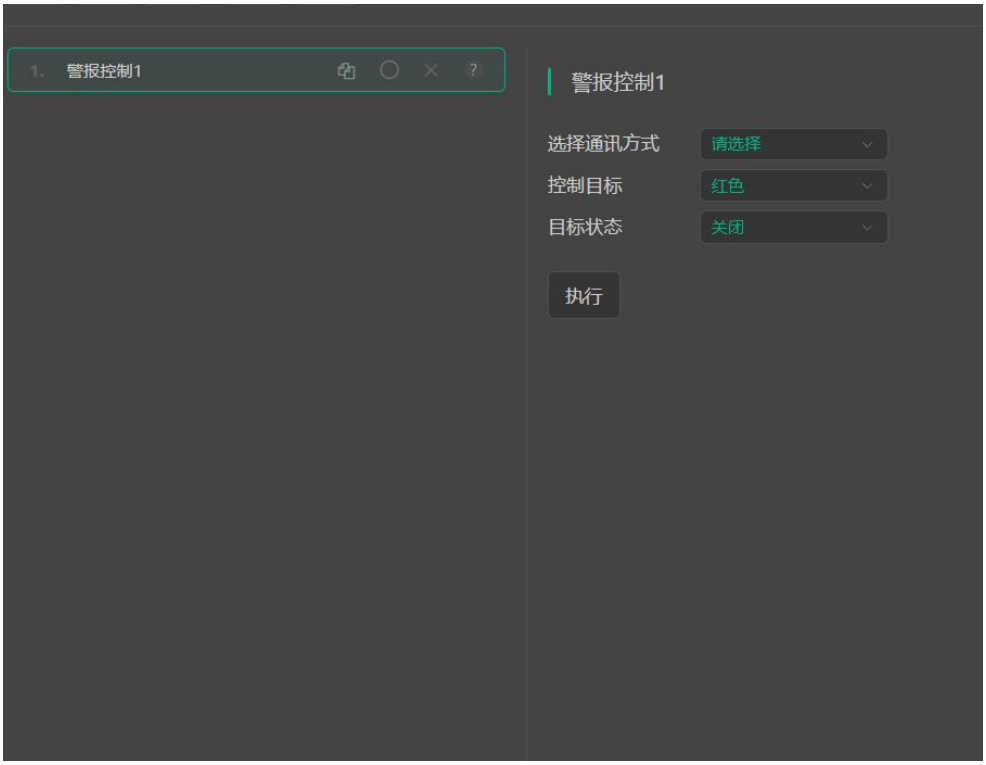


❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	选择通讯方式	Null		选择数据输出的通信通道，支持 TCP 网络通信或者串口通信
2	文本	Null		需要向外发送的原始文本内容

6.6 警报控制

警报控制工具用于三色报警灯控制，使用前必须先在 IO 模块完成串口配置，配置完成后即可选择对应设备实现报警灯控制。

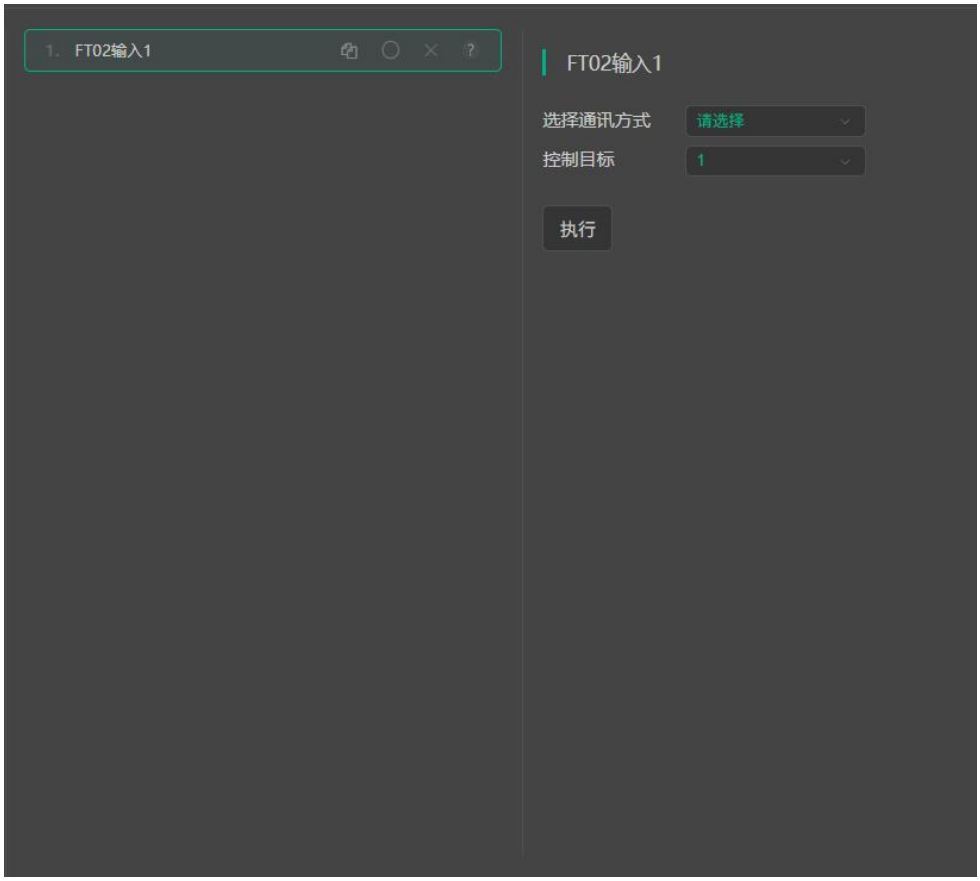


❖ 基本参数

编号	名称	默认值	取值范围	说明
1	选择通讯方式	Null		选择 IO 模块预先配置好的串口通讯设备，未选择设备时工具无法执行控制输出
2	控制目标	红色	红色	控制红色指示灯
			绿色	控制绿色指示灯
			蓝色	控制蓝色指示灯
			黄色	控制黄色指示灯
			蜂鸣器	控制设备蜂鸣器
3	目标状态	关闭	关闭	关闭对应灯或蜂鸣器，停止输出
			打开	对应灯常亮，如选择蜂鸣器则会持续鸣叫
			慢闪	对应灯低频闪烁，如选择蜂鸣器则会低频间歇鸣叫
			快闪	对应灯高频闪烁，如选择蜂鸣器则会高频间歇鸣叫
			爆闪	对应灯极速闪烁，如选择蜂鸣器则会急促间歇鸣叫

6.7 FT02 输入

FT02 输入工具用于读取 FT- 02RL 模块的输入通道状态。工具界面仅完成串口设备选择，实际输入数据需要调用脚本接口读取，详细用法可参考执行脚本章节。

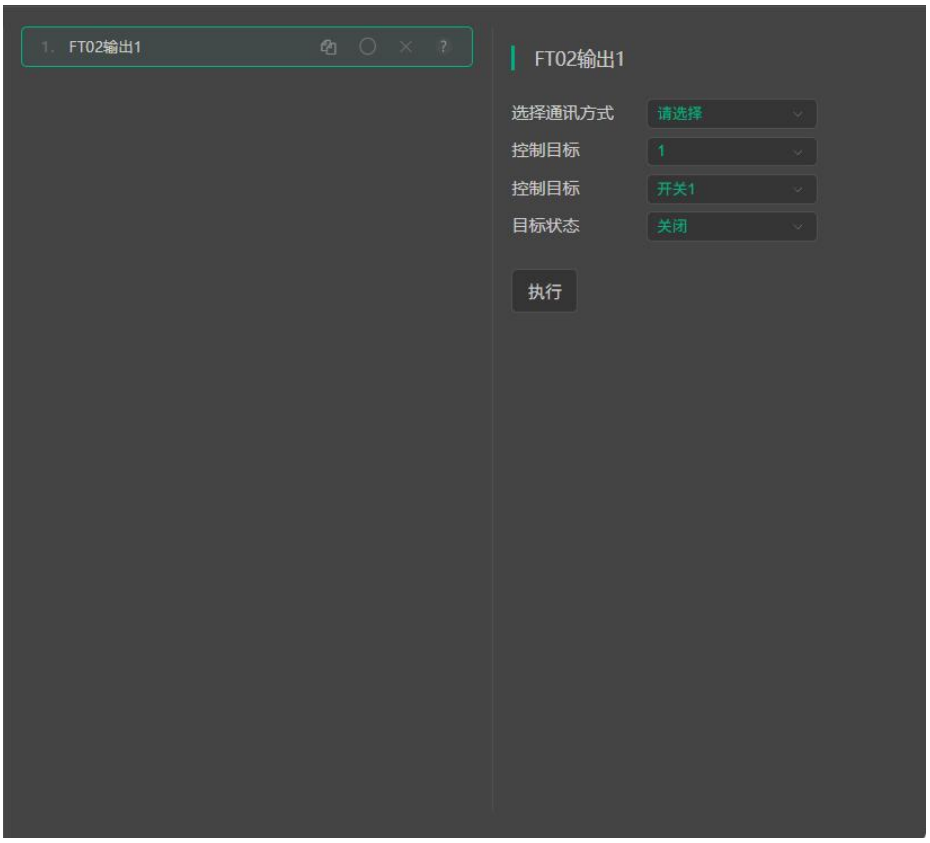


❖ 基本参数

编号	名称	默认值	取值范围	说明
1	选择通讯方式	Null		选择 IO 模块预先配置好的串口通讯设备，未选择设备时工具无法读取输入信息
2	站点号	1		Modbus 从站设备地址，0 为广播地址，填写模块硬件设置的实际站点编号

6.8 FT02 输出

FT02 输出工具用于对 FT- 02RL 模块下发输出指令，控制继电器吸合与断开。

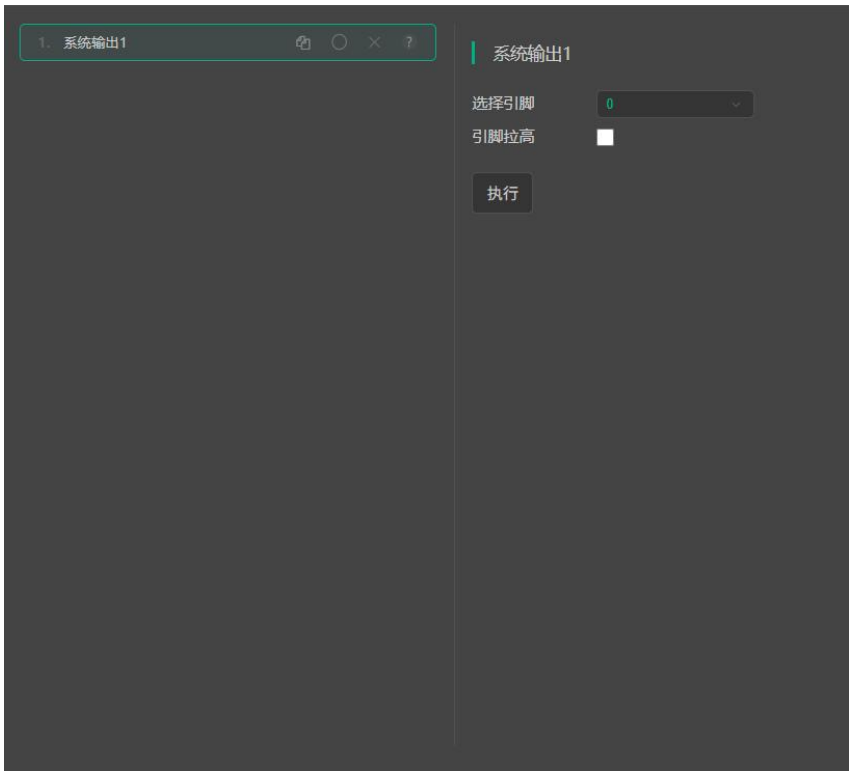


❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	选择通讯方式	Null		选择 IO 模块预先配置好的串口通讯设备，未选择设备时工具无法读取输入信息
2	站点号	1		Modbus 从站设备地址，0 为广播地址，填写模块硬件设置的实际站点编号
3	控制目标	开关 1	开关 1	选择 1 号继电器输出通道
			开关 2	选择 2 号继电器输出通道
4	目标状态	关闭	关闭	控制对应继电器断开
			打开	控制对应继电器吸合导通

6.9 系统输出

系统输出工具用于实现第二代板载控制器输出引脚的控制。除本工具外，也可以调用执行脚本系统空间接口完成控制器输出控制，具体用法请参考执行脚本开发章节。



❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	选择引脚	0	0~3	选择输出引脚号
2	引脚拉高	不勾选	勾选/不勾选	勾选后将选定引脚置为高电平，否则将选定引脚置为低电平

7 校准模块

7.1 开始位置补正/结束位置补正

位置补正工具用于对 ROI、标记点、轮廓做位置纠偏。被测目标发生位置偏移时，先通过轮廓匹配、斑点分析、边缘测量等算法定位目标固定特征点，使后续关联的 ROI、标记点跟随特征点完成坐标同步变换。该工具示教阶段需配置基准坐标，并绑定定位工具输出的实时坐标；需要补正的工具放置在【开始位置补正】与【结束位置补正】之间，补正工具依据实时坐标与基准坐标的位姿差异，解算出平移、旋转偏移量以及缩放系数，完成目标纠偏。

❖ 基本参数

编号	名称	默认值	取值范围	说明
实时坐标				
1	选择工具	Null		绑定上游定位工具（轮廓匹配 / 灰度匹配等），作为位姿数据源
2	选择数据	Null		多目标匹配场景下，指定读取匹配结果列表中某一条目标的位姿数据。
3	角度	0		自动读取绑定定位工具输出的目标旋转角度，随每次工具运行实时更新。

4	位置 X	0		自动读取绑定定位工具输出的目标中心点 X 像素坐标，随每次工具运行实时更新。
5	位置 Y	0		自动读取绑定定位工具输出的目标中心点 Y 像素坐标，随每次工具运行实时更新。
6	缩放	0		自动读取绑定定位工具输出尺度缩放系数，代表实际目标相对模板的尺寸比例，随每次工具运行实时更新。
基准坐标				
1	选择工具	Null		示教阶段绑定定位工具，用于一键快速写入基准位姿；运行阶段不生效。
2	选择数据	Null		多目标匹配示教场景，指定读取匹配结果列表中某一条目标位姿作为基准。
3	角度	0		工件标准姿态下的基准旋转角度；可通过快速示教自动填入，也支持手动输入修改。
4	位置 X	0		工件标准位置下的基准中心点 X 像素坐标；可通过快速示教自动填入，也支持手动输入修改。
5	位置 Y	0		工件标准位置下的基准中心点 Y 像素坐标；可通过快速示教自动填入，也支持手动输入修改。
6	缩放	0		工件标准状态下的基准缩放系数；可通过快速示教自动填入，也支持手动输入修改。

7.2 相机标定

相机标定工具用于求解相机内参，包含焦距 fx、fy，主点 cx、cy 以及径向畸变系数 k1、k2。工具需输入不少于 3 组标定板模板参数，模板参数来源于对标定板圆点特征的识别结果。

实际间距：标定板上特征点之间的真实物理间距，单位为 mm。

导出相机参数：将配置完成的全部模板参数导出为 JSON 文件。

导入相机参数：导入已导出的 JSON 格式相机参数文件，支持跨作业复用同一套标定参数。

配置模板：点击进入模板配置界面，完成多组标定板特征数据录入。

执行：完成模板参数配置后，点击执行，求解相机内参。

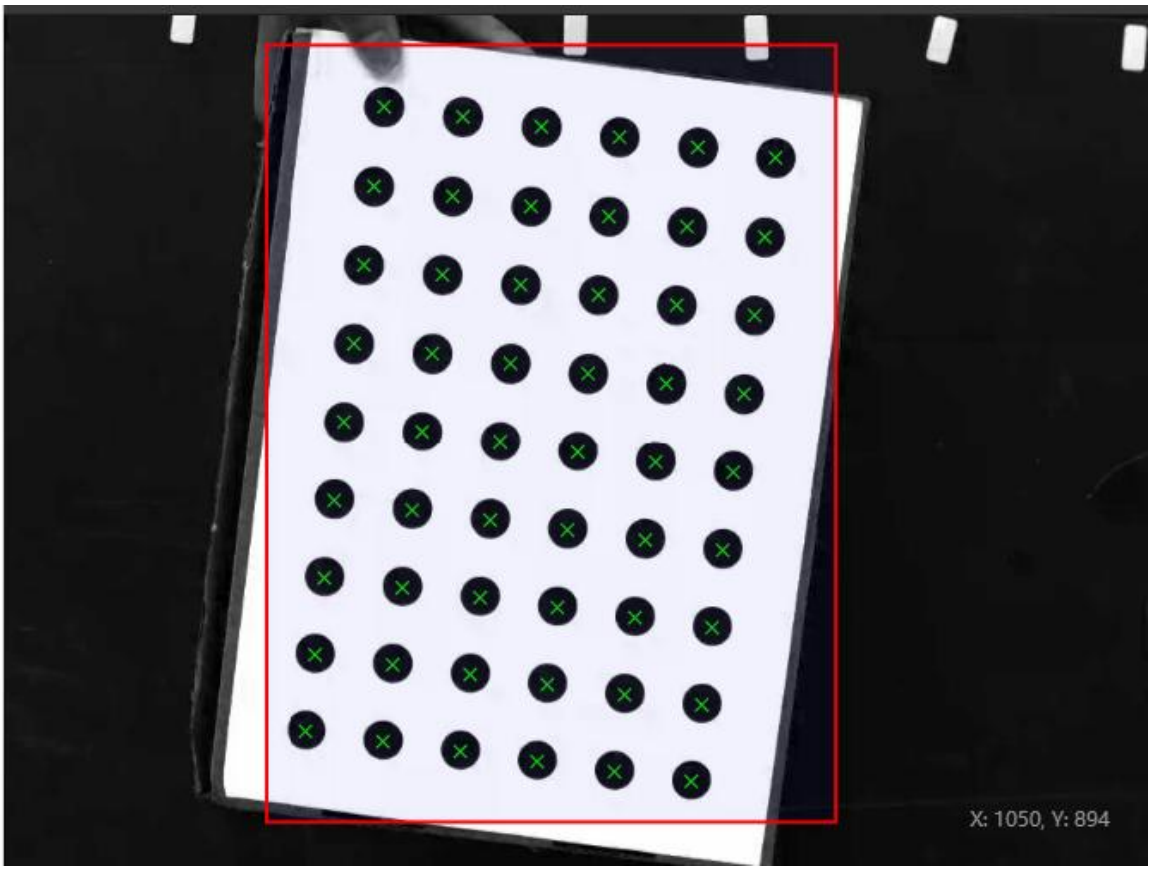
❖ 配置模板

点击配置模板按钮进入模板配置界面，点击添加模板，即会出现一个 DotMatrix 空模板，点击该模板即可进入该子模板的配置界面，如下：

❖ **模板参数**

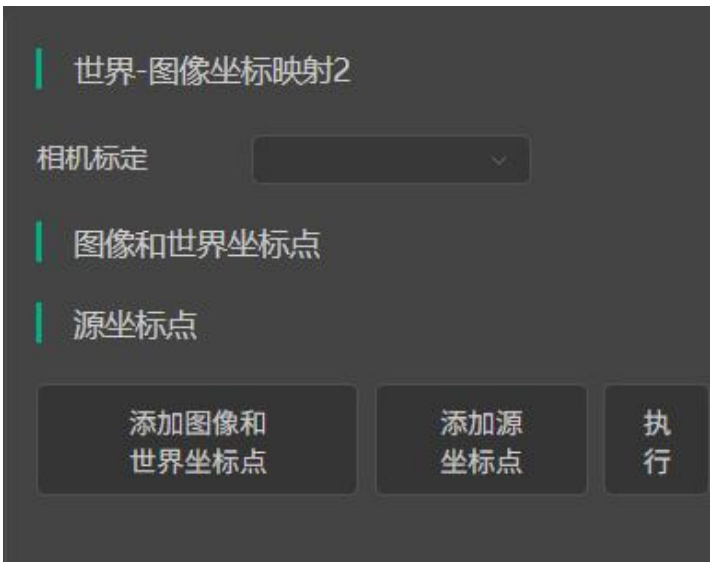
编号	名称	默认值	取值范围	说明
1	圆点类型	黑色	黑色/白色	选择标定板圆点的颜色，区分前景圆点与背景。
2	最大面积	2000	≥1	圆点轮廓允许的最大像素面积，过滤面积过大的干扰连通域
3	最小面积	20	≥1	圆点轮廓允许的最小像素面积，过滤噪点、小光斑等干扰。
4	分割阈值	127	0~255	灰度图像二值化阈值，用于分离圆点目标与背景。
5	列数	0	＞0	标定板圆点阵列的列方向数量。
6	行数	0	＞0	标定板圆点阵列的行方向数量。

调整完成模板参数后，点击学习模板，工具将一次性提取全部圆点特征参数，完成当前模板的配置。



7.3 世界—图像坐标映射

世界- 图像坐标映射工具，在获取相机内参的基础上，输入不少于 4 组世界坐标系与图像坐标系的匹配点对，求解相机外参。结合已知相机内参与解算得到的外参，即可实现点在世界坐标系、相机坐标系、图像坐标系三者之间的相互转换。该工具的全部参数配置与结果获取均可通过脚本函数完成，具体请参考执行脚本开发章节。



相机标定：关联已配置完成的相机标定工具，读取该工具输出的相机内参。

添加图像和相机坐标点：添加一组图像- 世界匹配点对，用于解算相机外参，至少需要添加 4 组有效点对。

添加源坐标点：输入待转换的世界坐标系源点；执行运算后，将该点转换输出为图像坐标系像素点。

执行：执行完成源坐标点的跨坐标系转换计算。

8 AI 模块

8.1 对象检测

对象检测工具加载模型训练章节导出的训练模型，工具执行后对输入图像完成目标检测，输出目标类别、置信度、检测框坐标等信息。可通过脚本接口读取各个目标的检测结果，详细用法请参考执行脚本开发章节。

模型选择：选择系统内已导入完成的目标检测模型

刷新：点击后清空模型选择下拉框的选中状态，需要重新选择目标模型。

上传新模型：点击弹出模型上传界面，详细操作参见下文模型上传小节。

下载模型：将当前选中模型导出为 zip 压缩包保存，后续可重新上传复用。

删除模型：删除当前选中的模型，删除后模型不可选用，如需再次使用需要重新上传。

检测阈值：置信度过滤阈值，取值范围 0~1，低于该阈值的检测结果将直接过滤，不会输出。

执行检测：手动触发一次图像检测，用于调试参数、预览检测效果。

❖ 模型上传

模型上传支持导入 WebDeepTraining 平台训练导出的 zip 模型包，也可使用本平台下载导出的 zip 模型文件。点击上传新模型按钮后即可弹出以下页面。

模型名称：为待上传模型设置自定义名称，用于工具内识别选用

模型文件(ZIP)：选取本地 zip 格式模型文件，仅支持合规模型压缩包

取消：关闭模型上传页面，放弃本次上传操作。

上传：填写模型名称并选中 zip 文件后，点击完成模型导入。

❖ 对象检测脚本

每一个对象检测工具都内置独立 Lua 脚本，用于对检测输出的目标做二次筛选。脚本遵循 Lua 原生语法，筛选逻辑在 accept()函数内实现；入参 item 代表单条检测目标，其数据结构与 DLObjDetectTool:result()返回列表元素完全一致，数据字段说明可查阅执行脚本开发章节中 DLObjDetectTool 工具介绍。

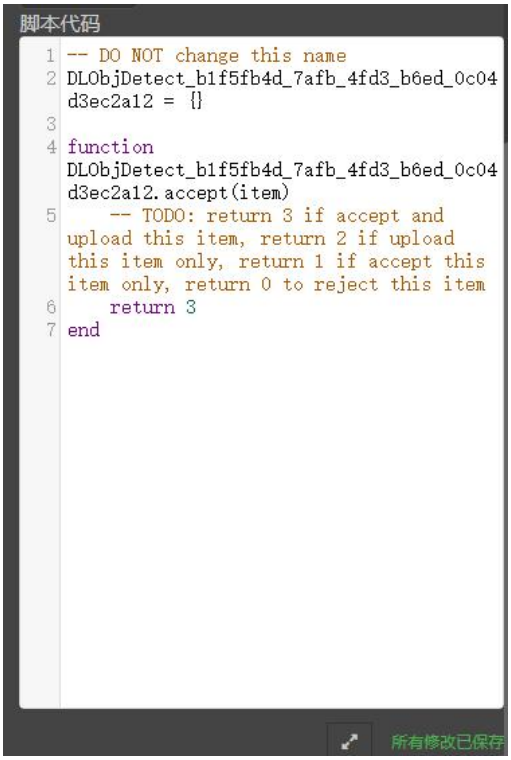
accept()函数支持返回 0、1、2、3 四种状态码，各返回值含义：

返回 0：表示该 item 被拒绝，该目标对象在脚本中不可使用，且不会推送到界面上显示

返回 1：表示该 item 被接受，该目标对象可以在脚本中使用，但不会推送到界面上显示

返回 2：表示该 item 被接受，该目标对象可以推送到界面，但在脚本中不可用

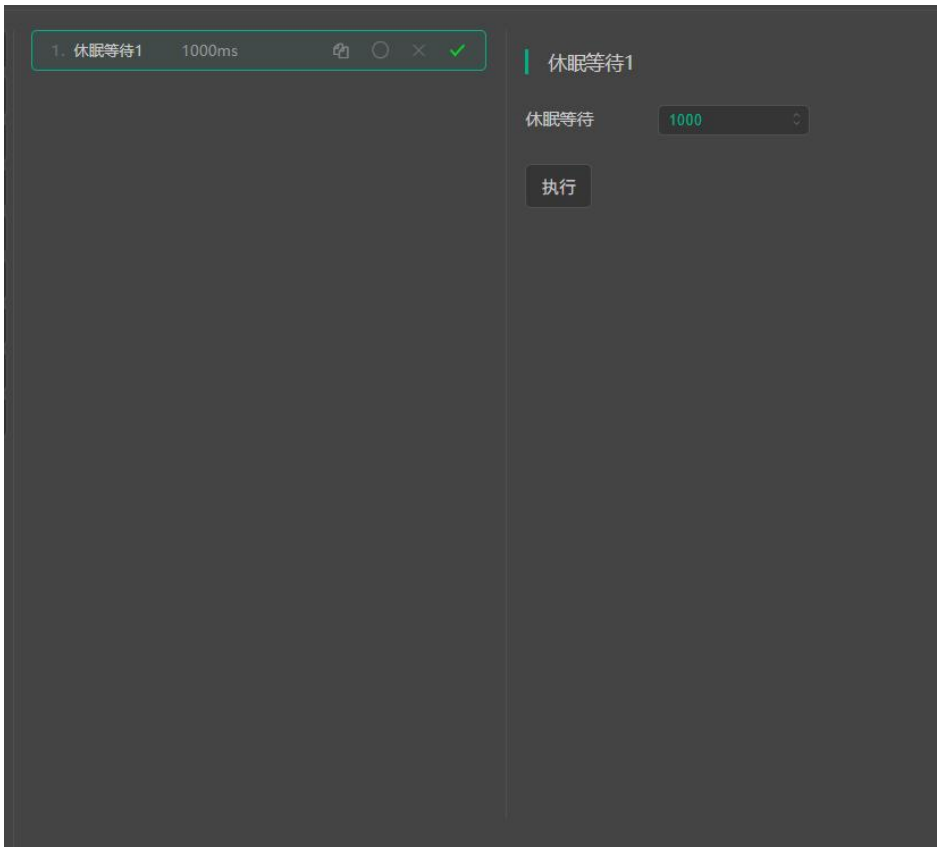
返回 3：表示该 item 被接受，该目标对象可以在脚本中使用，也会被推送到界面



9 系统模块

9.1 休眠等待

休眠等待工具用于等待一段时间后再往下执行，用于控制流程的执行时间或是等待一些事件。

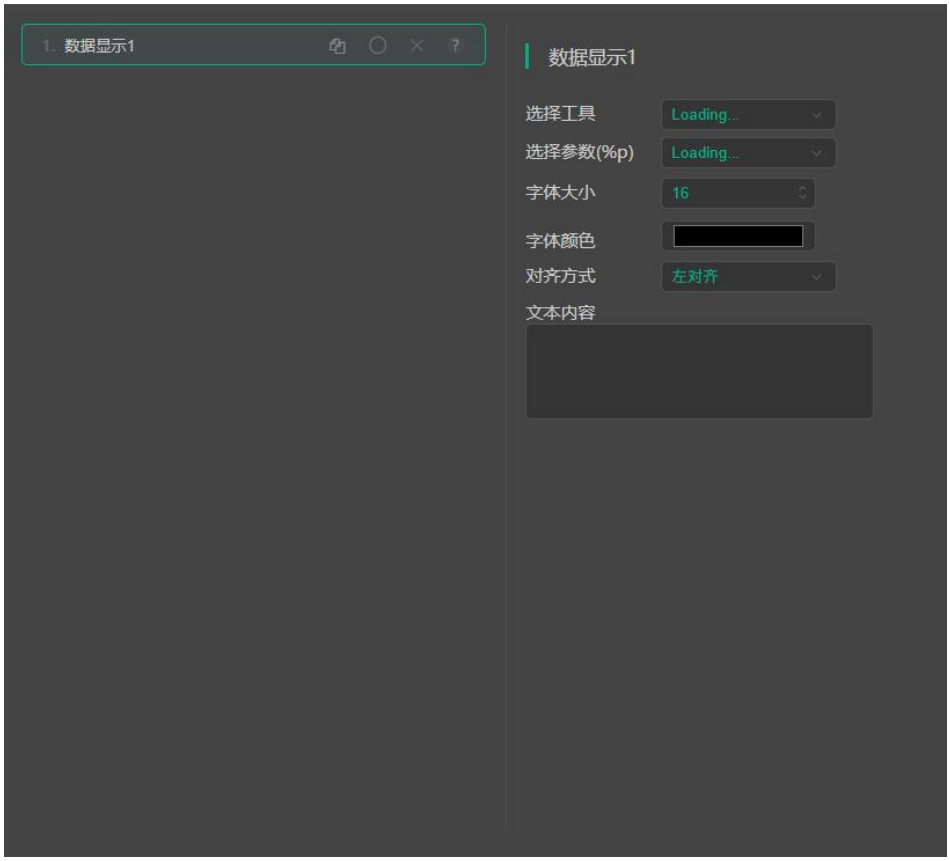


休眠等待：配置等待时长，单位 ms。

执行：触发工具单次运行

9.2 数据显示

本工具用于在图像上叠加渲染数值、文字信息，单次执行即可将指定文本绘制至图片上，工序运行结束后渲染内容自动清除。参数配置项支持绑定上游工具，提取对应工具的运算参数渲染至图像；可自由调节文字字号、字体颜色，文字摆放位置通过在图像中打标机点设定。本工具可配合执行脚本共同使用，具体说明参考执行脚本开发章节。界面示意如下。



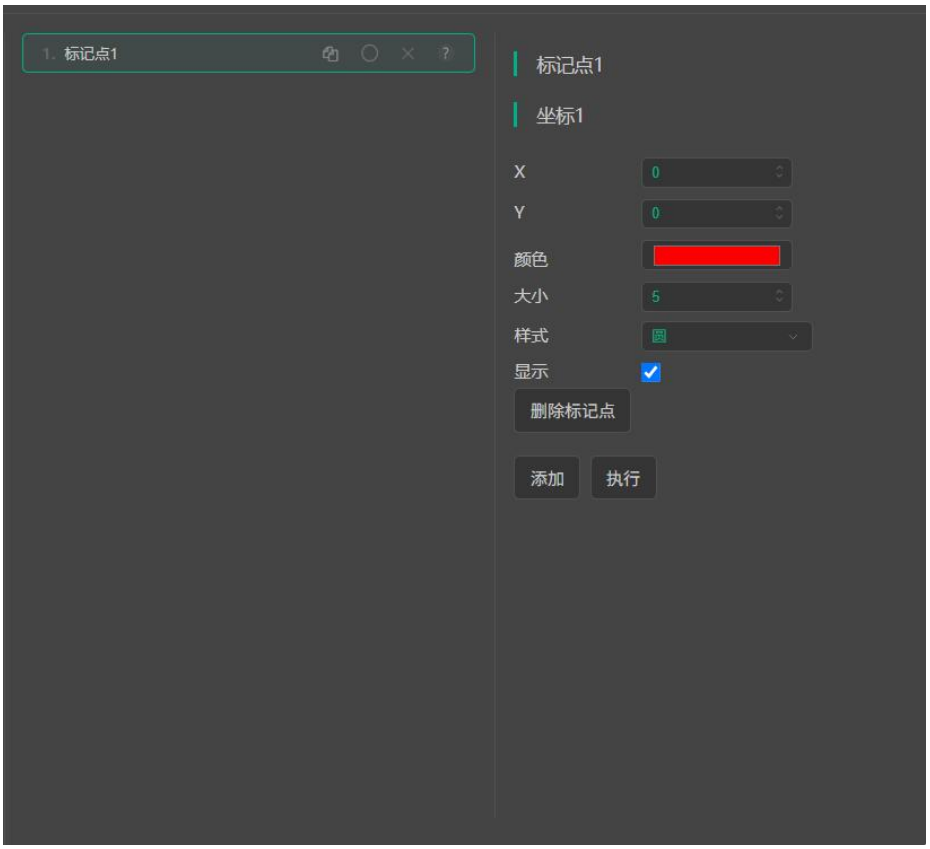
❖ 基本参数

编号	名称	默认值	取值范围	说明
1	选择工具	Null		选择流程内已存在的工具，作为数据源。
2	选择参数(%p)	Null		选中目标工具输出的参数，作为待显示的数据来源；在文本内容中填入%p，即可引用该参数
3	字体大小	16	≥0	设置显示文本的字号，单位为像素。
4	字体颜色	#000000	#000000~#ffffff	设置文本颜色，使用十六进制颜色码。
5	对齐方式	左对齐	左对齐/居中/右对齐	设置文本在显示区域内的排版对齐方式。

6	文本内容	Null		自定义静态文本内容

9.3 标记点

标记点工具用于在图像上绘制一组标记点，支持点击添加按钮录入坐标参数，也可直接在图像画面上点击完成标记点添加。



❖ 基本参数

每个标记点都有以下参数可供配置

编号	名称	默认值	取值范围	说明
1	X	0	≥0	设置标记点的图像 X 像素坐标。

2	Y	0	≥0	设置标记点的图像 Y 像素坐标。
3	颜色	#ff0000	#000000~#ffffff	配置标记点的绘制颜色。
4	大小	5	≥1	设置标记点的显示尺寸。
5	样式	圆	圆/点/十字	选择标记点绘制样式
6	显示	勾选	勾选/不勾选	勾选控制该标记点是否在图像上渲染输出。

9.4 完成工序

完成工序工具执行后，将直接结束当前工序，跳转至下一工序，无需等待本工序内剩余工具全部执行完成。该工具无任何参数配置。

9.5 动作指引

动作指引工具用于引导作业人员至目标点位开展作业，标记点位作业状态（未做 / 已做）；工具会在画面上渲染闪烁 ROI 提示区域，多用于打螺丝等工位引导场景。选定状态与类型后，点击添加指引，在图像上框选对应区域即可完成指引配置，工具支持同时配置多路指引。

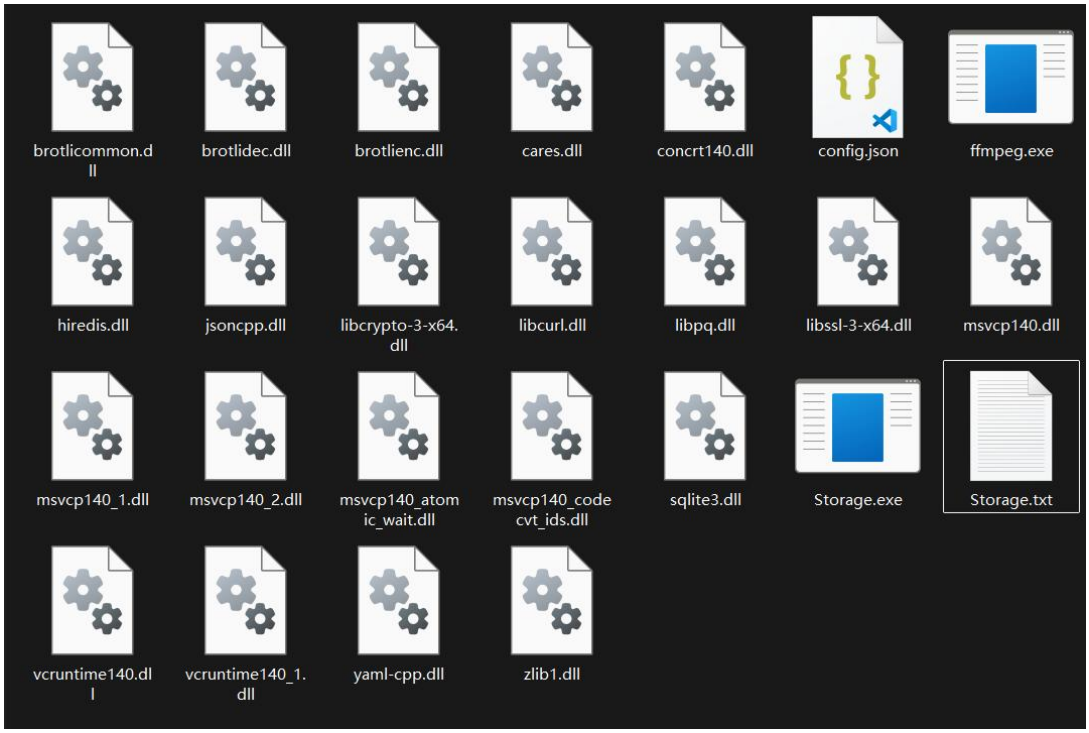
❖ **基本参数**

编号	名称	默认值	取值范围	说明
1	状态	闪烁	闪烁	ROI 会在界面上闪烁
			隐藏	ROI 在界面隐藏不显示
			展示	ROI 固定显示在界面
			呼吸灯	ROI 呈现呼吸渐变效果
2	类型	矩形	矩形	指引框为矩形
			旋转矩形	指引框为可旋转矩形
			圆	指引框为圆形
			环	指引框为圆环
			多边形	指引框为多边形
3	指引	Null		显示当前选中第几个指引
ROI 参数				
1	ROI 类型	Null		显示用户添加该指引时的类型，不可改

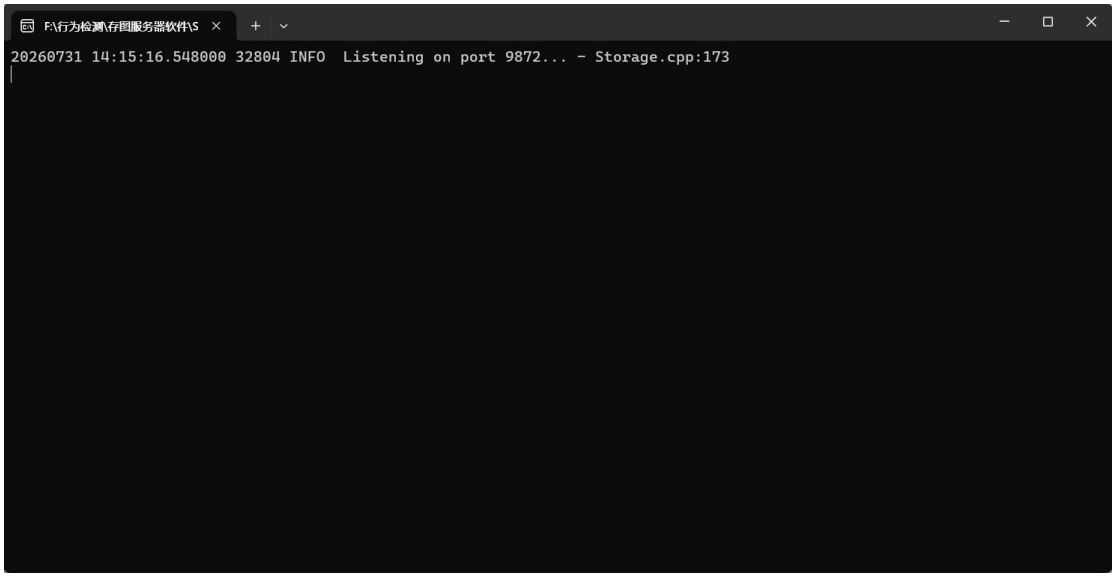
2	ROI 颜色	#ff0000	#000000~ #ffffff	设置指引 ROI 框的绘制颜色

9.6 图像保存

该工具均依赖 Storage 软件配套使用，软件需安装至作为显示端的 Windows 工控上位机，安装完成后打开目录内的 Storage.exe 程序。



程序启动后弹出黑色控制台窗口，如下图：



Storage.exe 运行占用本机 9872 端口；上位机打开浏览器访问 http://localhost:9872，页面显示 Hello, I’m working!即代表服务启动正常。

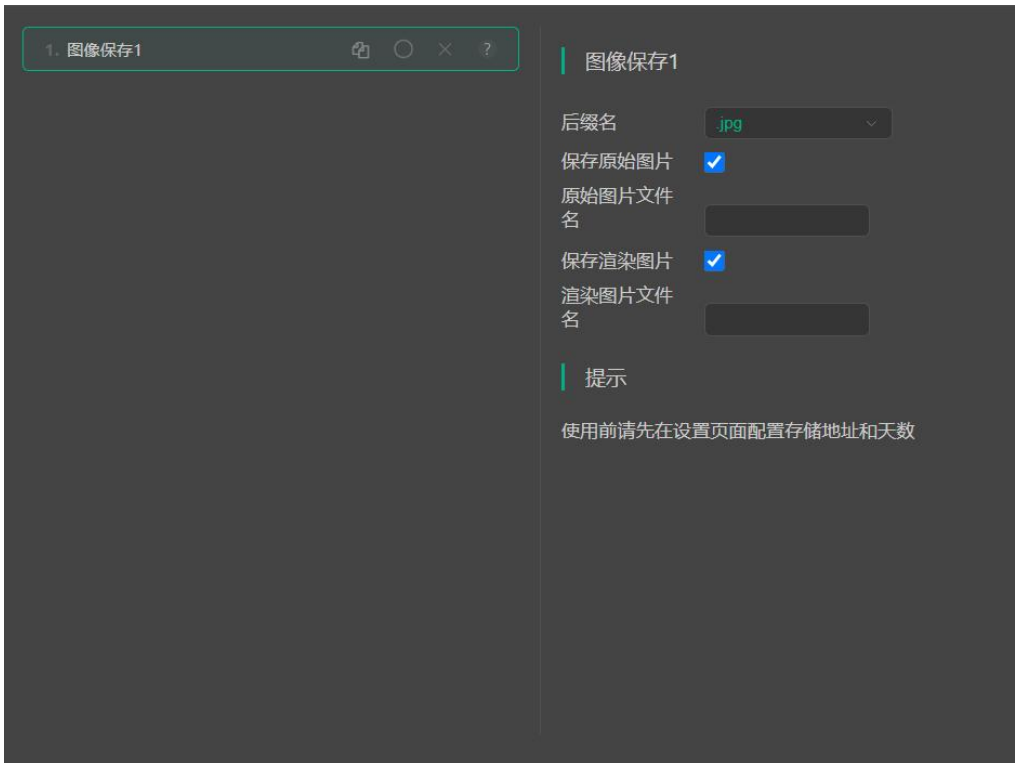
随后进入行为检测平台，将配置内远程存储地址填写为 http://<ip>:9872，其中<ip>替换为工控机实际 IP 地址，并可设置存储天数，工具运行时便会将图片推送至该服务器，并将存储时间超过存储天数的的图片统一清除。

若未自定义存储路径，Storage.exe 自动在程序同级目录生成 images 文件夹，图片按照文件名时间归类至对应日期子目录（例如 images/2026/06/04 存放当日图像）；配置指定存储目录后，图片则保存至自定义路径。

上述删除功能默认仅对 Storage 根目录下的图片、视频生效。若文件存储在其他路径，需要修改软件目录下 config.json 文件，在 watch_dirs 数组中填写需要监控的目录列表；路径字符串需使用双引号包裹，多个目录路径之间使用逗号分隔。

```
"custom_config" : {
    "watch_dirs" : [
    ]
}
```

工具操作页面如下，全部参数均支持通过脚本函数配置，具体用法参考执行脚本开发章节。



❖ 基本参数

编号	参数项	默认值	取值范围	注释
1	后缀名	.jpg	.jpg/.png/.webp	选择图片保存后图片格式
2	保存原始图片	勾选	勾选/不勾选	选择是否保存原始图片(不含数值检测框以及数值显示等信息)
3	原始图片文件名	Null		仅勾选保存原图后生效，填写不带后缀的完整路径与文件名，自定义原图存储位置。
4	保存渲染图片	勾选	勾选/不勾选	选择是否保存渲染图片(含数值检测框以及数值显示等信息)

5	渲染图片文件名	Null		仅勾选保存渲染图后生效，填写不带后缀的完整路径与文件名，自定义渲染图存储位置。

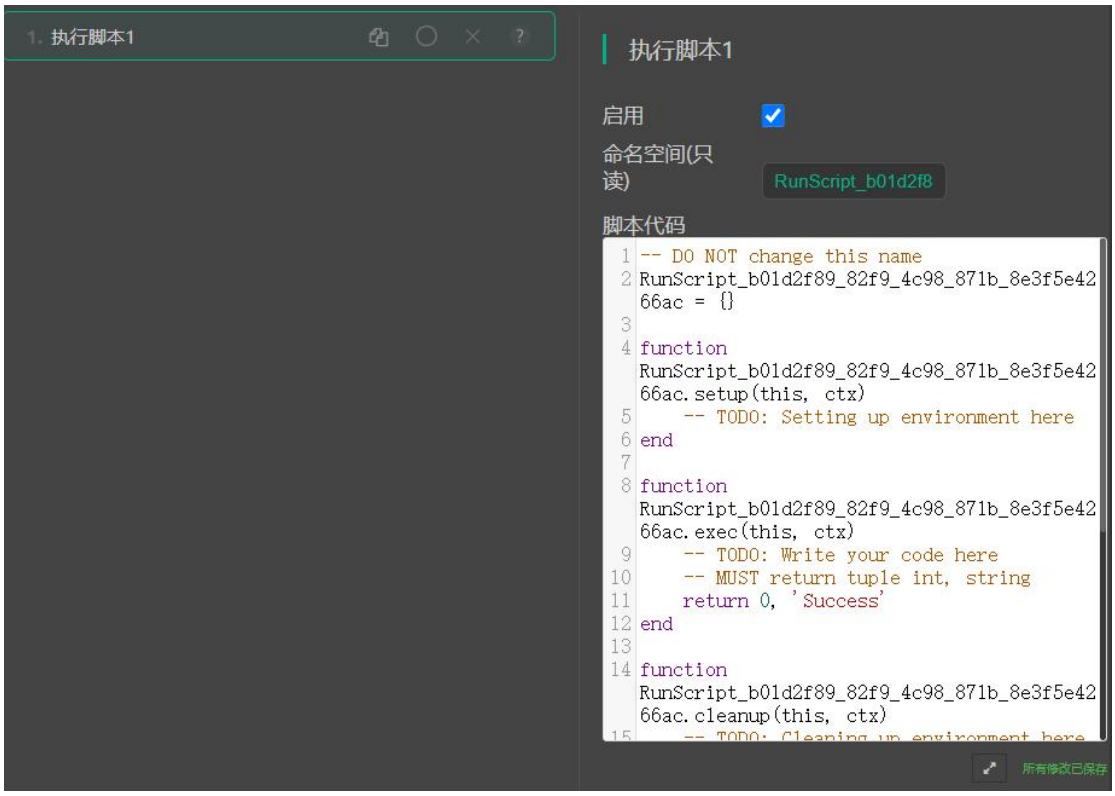
9.7 视频录制

视频录制功能依托于 Storage 软件，需要在显示上位机安装并运行 Storage.exe，程序需保持持续运行。Storage 相关说明可参考图像保存工具，操作步骤流程完全一致。使用前需在工序设置- 视频录制中开启即时回放功能，在脚本内调用 Task.save_video(file_name)，即可将本工序视频保存至 file_name 指定路径。本工具界面上无任何参数配置。

注：视频录制功能需要浏览器处于最新版本，否则录制。

9.8 执行脚本

执行脚本是本平台用于实现自定义检测逻辑的核心工具。由于该模块内容较多，详细说明请查阅【执行脚本开发】章节。



十一、 执行脚本开发

本设备 SOP 工序核心业务逻辑均依托执行脚本工具完成，涵盖检测算法逻辑、IO 信号收发、各类工具调用及参数配置、PLC 数据交互、上位机不良品弹窗展示等功能，下文将重点说明脚本整体架构及内置可用函数：

1 脚本基础框架

本脚本为视觉流程自定义运行脚本，主要使用语言为Lua，每个脚本均分配一个独立的命名空间。固定框架如下，包含五个函数，所有函数入参统一为 this、ctx，生命周期独立：

```
RunScript_xx = {}
-- 作业全局初始化
function RunScript_xxx.setup(this, ctx)
end
-- 检测逻辑执行
function RunScript_xxx.exec(this, ctx)
    return 0, 'Success'
end
-- 作业全局销毁清理
function RunScript_xxx.cleanup(this, ctx)
end
-- 单轮工序开始前置处理
function RunScript_xxx.on_process_begin(this, ctx)
end
-- 单轮工序结束后置处理
function RunScript_xxx.on_process_end(this, ctx)
End
```

2 参数说明（通用所有函数）

2.1 this（脚本实例对象）

代表当前正在运行的脚本工具自身实例，为脚本私有对象，仅作用于当前脚本。

2.2 ctx（全局执行上下文工具对象）

代表整个作业的全局运行上下文工具表格，为全作业共享对象。主要用于获取工具对象，并对工具对象进行执行、参数修改等操作。

3 五大函数详细说明

3.1 作业初始化函数 setup

触发时机：整套检测作业启动、流程首次加载时执行，全局仅执行一次
用途：用于作业运行前的全局初始化操作。主要完成全局变量定义、固定参数配置、设备初始状态初始化、全局缓存创建等一次性初始化工作。

3.2 工序前置函数 on_process_begin

触发时机：工序开始前执行一次
用途：用于工序检测的前置准备。主要进行清空本工序临时缓存、重置上轮检测状态、初始化本轮运算变量、复位单次流程标记等初始化操作。

3.3 核心执行函数 exec

触发时机：每轮工序前置准备完成后，会开始执行该函数，该函数会重复执行，直到返回 0 为止
用途：编写当前工序的核心业务逻辑，为脚本核心功能函数。包含检测通过条件、报警灯、运行日志发送、NG 事件、IO 控制、存图等逻辑。
返回值规则：必须固定返回两个参数，顺序不可颠倒

返回格式：return 整型状态码, 字符串提示信息
状态码 = 0：判定当前工具执行成功，自动进入下一道工序
状态码 ≠ 0：判定当前工具执行异常/NG/未达到检测次数要求，终止本轮检测，系统会重新触发执行当前 exec 函数重试

3.4 工序后置函数 on_process_end

触发时机：检测工序执行完成（exec 返回 0 执行成功）后触发，仅工序正常结束时执行，异常终止时不执行
用途：用于单轮检测结束后的收尾清理工作。主要进行清空本工序临时数据、复位单次报警状态、关闭临时光源、清空滑动窗口等该工序收尾操作。

3.5 作业清理函数 cleanup

触发时机：整套检测作业停止、软件退出时执行，全局仅执行一次
用途：用于整套作业结束后的全局资源释放与变量清理。主要销毁全局缓存、复位设备全局状态、释放图像资源、清空全局统计数据等收尾工作。

4 执行上下文对象的接口

所有已存在的工具对象可通过执行上下文的接口被获取，针对执行上下文对象的工具，有以下几种用法：

4.1 根据工具名字获取工具对象

方法：ctx:get_tool_by_name("tool_name')
说明：参数是一个字符串，是流程栏中可以看到的工具名字。若成功，则返回指定的工具对象，失败则返回 nil。

4.2 根据工具 id 获取工具对象

方法：ctx:get_tool_by_id("tool_id")
说明：参数是一个字符串，是工具栏中工具 id。若成功，则返回指定的工具对象，失败则返回 nil。

4.3 访问指定的工具对象

方法：ctx:visit(tool)
说明：遍历当前所有的工具访问者，访问参数中指定的工具。参数是一个工具对象，可以通过 ctx::get_tool_by_name 或者 ctx::get_tool_by_id 获得，或者通过工具构造函数来创建。目前访问工具主要是用于把工具执行结果发送到上位机（如果有上位机的话）。

4.4 根据设备名字获取设备

方法：ctx:get_device_by_name("device_name')
说明：参数是一个字符串，是创建设备时指定的名字。若成功，则返回指定的设备对象，失败则返回 nil。

4.5 根据设备 id 获取设备

方法：ctx:get_device_by_id("device_id")
说明：参数是一个字符串，是创建设备时系统分配的唯一标识。若成功，则返回指定的设备对象，失败则返回 nil。

4.6 强制完成当前工序

方法：ctx:accept_task()
说明：无参数，无返回值

5 工具对象可调用函数

所有工具对象均可调用以下函数：

5.1 获取工具 id

方法：tool:id()
说明：返回工具的字符串 id。

5.2 获取工具名称

方法：tool:name()
说明：返回工具的字符串名称。

5.3 执行工具

方法：tool:exec(ctx)
说明：可执行该工具对象，参数是执行上下文对象。

5.4 获取执行的错误码

方法：tool:error_code()
说明：获取该工具执行后返回的错误码，返回 0 为工具正常执行，否则工具执行错误，可用于工具执行后的状态判断，防止直接使用工具结果进行逻辑判断而程序崩溃。

5.5 获取执行的错误消息

方法：tool:error_msg()
说明：获取该工具执行后返回的错误信息，返回 nil 为无错误。可用于工具执行错误后获取相应错误信息。

6 系统空间函数

系统空间函数现大致为两类，一类为事件发送，主要用于控制作业的启停；另一类为控制器模块的函数

6.1 事件发送

方法：System.send_event(event,[, arg])
说明：其中的参数 event 是一个字符串，表示事件的名称，目前可发送的事件有”start”，”pause”，”resume”，“stop”，分别表示开始作业，暂停作业，复位作业，停止作业；第二个参数 arg 是可选的参数，也是个字符串，将会以 arg 的值传给上位机。

6.2 设置引脚输出值

方法：System.output(pin, value)
说明：参数 pin 是引脚编号，取值 0-3，value 是输出值，取值 0 或者 1。成功返回 true，失败返回 false。

6.3 读取开关状态

方法：System.switch_state([channel])
说明：可选参数 channel 是通道号，取值 0-3 如果不指定通道，则读取所有通道的状态，以一个无符号整数返回，最高 8 位是通道 0 的状态。如果系统控制器未打开，则返回 nil。

6.4 设置开关状态

方法：System.set_switch_state(channel, state)
说明：参数 channel 是通道号，取值 0-3，state 是想要设置的状态，取值 true 或者 false。成功返回 true，失败返回 false。

6.5 读取 pwm 状态

方法：System.pwm_state([channel])
说明：可选参数 channel 是通道号，取值 0-3。如果不指定通道，则读取所有通道的状态，以一个无符号整数返回，最高 8 位是通道 0 的状态。如果系统控制器未打开，则返回 nil。

6.6 设置 pwm 状态

方法：System.set_pwm_state(channel, state)
说明：参数 channel 是通道号，取值 0-3，state 是想要设置的状态，取值 true 或者 false。成功

返回 true，失败返回 false。

6.7 读取 pwm 设置值

方法：System.pwm_value([channel])
说明：可选参数 channel 是通道号，取值 0-3。如果不指定通道，则读取所有通道的 pwm 值，以一个无符号整数返回，最高 8 位是通道 0 的 pwm 值，如果系统控制器未打开，则返回 nil。

6.8 设置 pwm 值

方法：System.set_pwm_value(channel, value)
说明：参数 channel 是通道号，取值 0-3，value 是想要设置的 pwm 值，取值 0-255。成功返回 true，失败返回 false。

6.9 读取开启延时时间

方法：System.open_delay([channel])
说明：可选参数 channel 是通道号，取值 0-3。如果不指定通道，则读取所有通道的开启延时，以一个无符号 64 位整数返回，最高 16 位是通道 0 的开启延时。如果系统控制器未打开，则返回 nil。

6.10 设置开启延时时间

方法：System.set_open_delay(channel, value)
说明：参数 channel 是通道号，取值 0-3，value 是想要设置的开启延时，取值 0-65535，单位为 ms。成功返回 true，失败返回 false。

6.11 读取关闭延时时间

方法：System.close_delay([channel])
说明：可选参数 channel 是通道号，取值 0-3。如果不指定通道，则读取所有通道的关闭延时，以一个无符号 64 位整数返回，最高 16 位是通道 0 的关闭延时。如果系统控制器未打开，则返回 nil。

6.12 设置关闭延时时间

方法：System.set_close_delay(channel, value)
说明：参数 channel 是通道号，取值 0-3，value 是想要设置的关闭延时，取值 0-65535，单位为 ms。成功返回 true，失败返回 false。

7 任务空间函数

7.1 事件发送

方法：Task.send_event(event[, arg])
说明：其中的参数 event 是一个字符串，表示事件的名称，目前可取值有 NG、action_start、action_stop。第二个参数 arg 为可选的参数，也是个字符串，将会以 arg 的值传给上位机。event 的取值及对应的 arg 如下所示：

①NG：发送该事件时，作业将会 NG 报警并展示报警信息。arg 参数可填可不填，可填入自定义的报警信息。arg 内可以使用两个占位符：

%n：工序名称

%t：当前工序耗时

当 arg 内包含有占位符时会自动修改为对应的数据

②action_start：发送该事件时，表示行为开始检测。arg 参数必填，需填入行为分析中自定义的行为名称

③action_stop：发送该事件时，表示行为结束检测。arg 参数必填，需填入行为分析中自定义的行为名称

7.2 视频保存

方法：Task.save_video(file_name)
说明：其中的参数 file_name 是一个字符串，表示保存视频的文件名（不带后缀），可以是绝对路径，也可以是相对路径。
提示：仅当工序中【即时回放】处于勾选状态时，且 Storage 软件正常运行时，Task.save_video()接口才可以正常执行视频保存。

8 日志函数

该类型函数专门用于运行日志的记录与显示：

8.1 DEBUG 日志

方法：LOG.debug(msg)
说明：参数 msg 为一个字符串，用于描述 debug 信息。只显示在运行日志界面上。

8.2 INFO 日志

方法：LOG.info(msg)
说明：参数 msg 为一个字符串，用于描述 info 信息。只显示在运行日志界面上。

8.3 WARN 日志

方法：LOG.warn(msg)
说明：参数 msg 为一个字符串，用于描述 warn 信息。只显示在运行日志界面上。

8.4 ERROR 日志

方法：LOG.error(msg)
说明：参数 msg 为一个字符串，用于描述 error 信息。只显示在运行日志界面上。

8.5 审计日志空间函数

方法：AuditLog.add(event, detail)
说明：参数 event 是一个字符串，表示事件的名称；第二个参数 detail 是关于这个事件的详细说明。该方法的日志可以显示在审计日志界面。

9 ROI 模块

以下函数通用于所有可设置 ROI 的工具：

9.1 获取 ROI 列表

方法：tool:roi()
说明：返回对应工具设置的 roi 列表，列表中的每个元素是一个对象，包含以下属性：
①type，roi 的类型。可取值有 rect（矩形），rotrect（旋转矩形），circle（圆形），ring（环形），polygon（多边形）。
②value，roi 的具体数据。根据上面的类型不同，该对象的域不同，如下：
rect 的 value 属性：left, top, right, bottom;
rotrect 的 value 属性：center, width, height,angle;
circle 的 value 属性：center, radius;
ring 的 value 属性：center, radius, angle, sweep, range;
polygon 的 value 属性：count, vertex，count 是 vertex 中点的数量，vertex 是一个数组，数组中每个元素是一个对象，对象有两个属性：x, y。

9.2 清除工具中的 ROI

方法：tool:clear_roi()

说明：可清除该工具已存在的 ROI，无返回值。

9.3 添加矩形 ROI

方法：tool:add_rect_roi(left, top, right, bottom)
说明：可往工具中添加一个矩形 ROI。参数中 left, top, right, bottom 分别是矩形的左边，上边，右边和下边，都需要是整数。无返回值。

9.4 添加旋转矩形 ROI

方法：tool:add_rotrect_roi(center_x, center_y, width, height, angle)
说明：可往工具中添加一个旋转矩形 ROI。参数中 center_x, center_y, width, height, angle 分别是旋转矩形的中心 X，中心 Y，宽度，高度和旋转角度，都可以是浮点数。无返回值。

9.5 添加圆形 ROI

方法：tool:add_circle_roi(center_x, center_y, radius)
说明：可往工具中添加一个圆形 ROI。参数中 center_x, center_y, radius 分别是圆形的圆心 X，圆心 Y 和半径，都可以是浮点数。无返回值。

9.6 添加环形 ROI

方法：tool:add_ring_roi(center_x, center_y, radius, range, sweep, angle)
说明：可往工具中添加一个环形 ROI。参数中 center_x, center_y, radius, range, sweep, angle 分别是环形的圆心 X，圆心 Y，半径，半径范围,扫描角度范围，起始角度，都可以是浮点数。无返回值。

9.7 添加多边形 ROI

方法：tool:add_polygon_roi(points)
说明：可往工具中添加一个多边形 ROI。参数中 points 是一个浮点数的数组，数量要求是偶数，每两个数表示一个点的 x 和 y 坐标。无返回值。

10 S7Client 设备

以下方法适用于西门子 S71500 型号 PLC 设备，在使用以下方法前需先使用执行上下文的接口中的方法先获取设备对象：

10.1 连接 PLC

方法：device:connect_to(address, rack, slot)
说明：该方法可与 PLC 建立起连接。参数 address，rack，slot 分别是 PLC 的 IP 地址，机架号，槽号。成功返回 true,失败返回 false。

10.2 断开连接

方法：device:disconnect()
说明：该方法可与 PLC 断开连接。

10.3 读取 M 区数据

方法：device:mb_read(start, size)
说明：该方法用于读取 M 区数据。参数 start，size 分别是起始地址，读取的长度。成功返回读到的数据，失败返回 nil。

10.4 读取 A 区数据

方法：device:ab_read(start, size)
说明：该方法用于读取 A 区数据。参数 start，size 分别是起始地址，读取的长度。成功返回读到的数据，失败返回 nil。

10.5 读取 E 区数据

方法：device:eb_read(start, size)
说明：该方法用于读取 E 区数据。参数 start，size 分别是起始地址，读取的长度。成功返回读到的数据，失败返回 nil。

10.6 读取 D 区数据

方法：device:db_read(num, start, size)
说明：该方法用于读取 D 区数据。参数 num，start，size 分别是数据块数量，起始地址，是读取的长度。成功返回读到的数据，失败返回 nil。
注：该方法读取的数据会自动转成 ASCII 码表对应的字符，建议使用该方法读取字符串即可。

10.7 写 M 区数据

方法：device:mb_write(start, data)
说明：该方法可将数据写入 M 区。参数 start，data 分别是起始地址，想要发送的二进制数据。成功返回 true，失败返回 false。

10.8 写 A 区数据

方法：device:ab_write(start, data)
说明：该方法可将数据写入 A 区。参数 start，data 分别是起始地址，想要发送的二进制数据。成功返回 true，失败返回 false。

10.9 写 E 区数据

方法：device:eb_write(start, data)
说明：该方法可将数据写入 E 区。参数 start，data 分别是起始地址，想要发送的二进制数据。成功返回 true，失败返回 false。

10.10 写 D 区数据

方法：device:db_write(num, start, data)
说明：该方法可将数据写入 D 区。参数 num，start，data 分别是数据块数量，起始地址，想要发送的二进制数据。成功返回 true，失败返回 false。
注：写入单个地址支持两种入参格式：传入’\x<>’（<>为十六进制数值），代表传入十六进制字节数据；直接传入’<>’（<>为任意字符内容），代表传入普通字符串,一个字符会占用 plc 一个地址。建议使用该方法写入字符串即可。

10.11 设置连接地址

方法：device:set_host_address(address)
说明：该方法可设置连接地址。参数 address 是 PLC 的 IP 地址，以字符串形式传入，比如"192.168.1.1"。

10.12 获取连接地址

方法：device:host_address()
说明：该方法可获取此时的连接地址。返回字符串形式的连接地址。

10.13 设置连接机架号

方法：device:set_rack(rack)

说明：该方法可设置连接机架号。参数 rack 是机架号，以数字形式传入，比如 1。

10.14 获取连接机架号

方法：device:rack()
说明：该方法可获取此时连接机架号。返回数字形式的机架号。

10.15 设置连接槽位号

方法：device:set_slot(slot)
说明：该方法可设置连接槽位号。参数 slot 是槽位号，以数字形式传入，比如 1。

10.16 获取连接槽位号

方法：device:slot()
说明：该方法可获取此时连接槽位号。返回数字形式的槽位号。

10.17 写 D 区一个 8 位整数

方法：device:db_write_uint8(num, start, data)
说明：参数 num 是数据块数量，参数 start 是起始地址， 参数 data 是想要写的 8 位整数。

10.18 写 D 区一个 16 位整数

方法：device:db_write_uint16(num, start, data)
说明：参数 num 是数据块数量，参数 start 是起始地址，参数 data 是想要写的 16 位整数。

10.19 写 D 区一个 32 位整数

方法：device:db_write_uint32(num, start, data)
说明：参数 num 是数据块数量，参数 start 是起始地址，参数 data 是想要写的 32 位整数。

10.20 写 D 区一个 64 位整数

方法：device:db_write_uint64(num, start, data)
说明：参数 num 是数据块数量，参数 start 是起始地址，参数 data 是想要写的 64 位整数。

10.21 读 D 区一个 8 位整数

方法：device:db_read_uint8(num, start)
说明：参数 num 是数据块数量，参数 start 是起始地址。返回读到的 8 位整数，或者 nil。

10.22 读 D 区一个 16 位整数

方法：device:db_read_uint16(num, start)
说明：参数 num 是数据块数量，参数 start 是起始地址。返回读到的 16 位整数，或者 nil。

10.23 读 D 区一个 32 位整数

方法：device:db_read_uint32(num, start)
说明：参数 num 是数据块数量，参数 start 是起始地址。返回读到的 32 位整数，或者 nil。

10.24 读 D 区一个 64 位整数

方法：device:db_read_uint64(num, start)
说明：参数 num 是数据块数量，参数 start 是起始地址。返回读到的 64 位整数，或者 nil。

11 BarCode 工具

以下介绍条码工具可使用的一些方法：

11.1 获取工具执行结果

方法：tool:result()
说明：返回工具检测到的结果列表，列表中的每个对象包含以下属性：
①text,检测到的条码文本
②type,检测到的条码类型
③border,条码的边界，是一个以左上角、右上角、右下角以及左下角顺序坐标点组成的数组

11.2 获取识别条码的最大数量

方法：tool:max_count()
说明：返回一个整数

11.3 设置识别条码最大数量

方法：tool:set_max_count(num)

说明：参数 num 需要是一个整数，无返回值

11.4 获取搜索条码阈值

方法：tool:find_threshold()
说明：返回一个 0 到 255 间的值

11.5 设置搜索条码阈值

方法：tool:set_find_threshold(value)
说明：参数 value 需要是一个 0 到 255 间的值，无返回值

11.6 获取或设置最大条宽度

方法：tool:max_space()
说明：返回一个整数

11.7 设置最大条宽度

方法：tool:set_max_space(value)
说明：参数 value 需要是一个整数，无返回值

12 DLObjDetect 工具

以下介绍对象检测工具可使用的方法：

12.1 获取工具执行结果

方法：tool:result()
说明：返回工具检测到的结果列表，列表中的每个对象包含以下属性：
①class_name，检测到的对象标签名
②class_id，检测到的对象标签 id
③color，检测到的对象边界框的颜色
④prob，检测到的对象的置信度
⑤bounding，检测到的对象的边办框坐标，其中包含有 left, right, bottom,top 四个属性

13 DLPoseDetect 工具

以下介绍姿态检测工具可使用的方法：

13.1 获取工具执行结果

方法：tool:result()
说明：返回工具检测到的结果列表，列表中的每个对象包含以下属性：
①class_name，检测到的对象标签名
②class_id，检测到的对象标签 id
③color，检测到的对象边界框的颜色
④prob，检测到的对象的置信度
⑤bounding，检测到的对象的边办框坐标，其中包含有 left, right, bottom,top 四个属性
⑥keypoints，检测到的对象姿态关键点数组，数组的每个元素是包含 x,y 和 attr 三个元素的对象，其中 x 和 y 是该关键点的坐标，attr 是该关键点属性，0 表示该关键点没有被检测到，1 表示该关键点被检测到，2 表示该关键点被检测到且被标记为异常。

14 DLSegDetect 工具

以下介绍分割检测工具可使用的方法：

14.1 获取工具执行结果

方法：tool:result()
说明：返回工具检测到的结果列表，列表中的每个对象包含以下属性：
①class_name，检测到的对象标签名。
②class_id，检测到的对象标签 id。
③color，检测到的对象边界框的颜色。
④prob，检测到的对象的置信度。
⑤bounding，检测到的对象的边框坐标，其中包含有 left, right, bottom,top 四个属性。
⑥mask，检测到的对象掩码，该对象包含 width 和 height 两个属性，表示掩码的宽和高。

15 MarkPoint 工具

以下介绍标记点工具可使用的方法：

15.1 获取原始点数组

方法：tool:origin_point_list()
说明：返回原始点的数组，数组中每个元素是一个对象，其中有 x 和 y 两个属性。

15.2 获取位置补正后点数组

方法：tool:mapped_point_list()
说明：返回位置补正后点的数组，数组中每个元素是一个对象，其中有 x 和 y 两个属性。

16 MBWrite 工具

在使用 MB 写工具的方法写入数据时，需配置好工具的通讯方式、站点号以及地址。以下介绍 MB 写工具可使用的方法：

16.1 发送裸 MB 数据帧

方法：tool:write_raw(data)
说明：参数 data 要求是一个完整的二进制 MB 数据帧(如果是 rtu，则要求后面有 2 字节的 crc 校验码；如果是走 tcp，则要求前面有 6 个字节的前缀)。成功返回 true，失败返回 false。

16.2 写线圈

方法：tool:write_bool(value)
tool:write_coil(value)
说明：参数 value 是布尔值，true 或者 false。成功返回 true，失败返回 false

16.3 写单个寄存器

方法：tool:write_word(value)
说明：参数 value 是一个 16 位无符号整数。 成功返回 true，失败返回 false。

16.4 写字符串

方法：tool:write_text(str)
说明：参数 str 是一个字符串，如果长度不是 2 的倍数，则最后会填充 0。成功返回 true，失败返回 false。

16.5 写二进制数据

方法：tool:write_bin(bin)
说明：参数 bin 是一个二进制字符串，如果不是 2 的倍数，则最后会填充 0.成功返回 true，失败返回 false。

16.6 获取最后一次请求的响应数据

方法：tool:last_response()
说明：返回最后一次请求的完整响应数据

17 MBRead 工具

以下介绍 MB 读工具可使用的方法：

17.1 获取读到的数据

方法：tool:data()
说明：返回读到的数据或者 nil

17.2 获取最后一次请求的响应数据

方法：tool:last_response()
说明：返回最后一次请求的完整响应数据

18 Blob 工具

以下介绍斑点分析工具可使用的方法：

18.1 获取工具执行结果

方法：tool:result()
说明：返回工具检测到的结果列表，列表中的每个对象包含以下属性：
①convex 凸包点数据
②removed 是否被排除
③hole_nums 孔洞数量
④area 斑块面积
⑤center_mass_x 重心 X(椭圆中心 X)
⑥center_mass_y 重心 Y(椭圆中心 Y)
⑦rotation 主轴角度(椭圆角度)

⑧major_axis 主轴长度(椭圆长轴)
⑨minor_axis 次轴长度(椭圆短轴)
⑩axis_ratio 次轴/主轴比
⑪convex_area 凸包面积
⑫convex_area_ratio 斑点面积/凸包面积比
⑬box_center_x 外接矩形中心 X
⑭box_center_y 外接矩形中心 Y
⑮box_angle 外接矩形角度
⑯box_width 外接矩形宽度
⑰box_height 外接矩形高度
⑱box_ratio 外接矩形宽高比
⑲box_area_ratio 斑点面积/外接矩形面积比
⑳mask 掩码

19 ImageSave 工具

图像保存工具可以使用 ImageSaveTool()来创建一个工具实例，以下介绍该实例可使用的一些方法：

19.1 获取图片保存后缀

方法：tool:suffix()
说明：返回一个字符串，工具中设置的图像保存文件名后缀。

19.2 设置图片保存路径

方法：tool:set_suffix(suffix)
说明：参数中的 suffix 是一个字符串，可以是.jpg,.png,.bmp,.tiff 等，需要注意的是一定要有点。

19.3 获取是否保存原图

方法：tool:save_original_image()
说明：返回一个布尔值

19.4 设置是否保存原图

方法：tool:set_save_original_image(value)
说明：参数是一个布尔值，true 表示保存原图，false 表示不保存原图

19.5 获取是否保存渲染图

方法：tool:save_rendered_image()
说明：返回一个布尔值

19.6 设置是否保存渲染图

方法：tool:set_save_rendered_image(value)
说明：参数是一个布尔值，true 表示保存渲染图，false 表示不保存渲染图

19.7 获取保存原图自定义文件名

方法：tool:original_image_custom_file_name()
说明：返回一个字符串

19.8 设置保存原图自定义文件名

方法：tool:set_original_image_custom_file_name(file_name)
说明：参数 file_name 是一个字符串，表示不包含后缀的文件名，文件名可以包含路径，中间以/分隔，但是不要以/开头。

19.9 获取保存渲染图自定义文件名

方法：tool:rendered_image_custom_file_name()
说明：返回一个字符串

19.10 设置保存渲染图自定义文件名

方法：tool:set_rendered_image_custom_file_name(file_name)
说明：参数 file_name 是一个字符串，表示不包含后缀的文件名，文件名可以包含路径，中间以/分隔，但是不要以/开头。

20 FT02Input 工具

以下介绍 FT02 输入工具可使用的一些方法：

20.1 获取读到的输入引脚状态

方法：tool:state()
说明：返回的是一个整数，最低位是引脚 0 的状态，接着是引脚 1 的状态，以此类推。

21 AlarmControl 工具

以下介绍报警灯工具可使用的一些方法：

21.1 判断是否连接

方法：tool:is_connected()
说明：判断报警灯是否连接，即串口是否已经成功打开

22 ShowData 工具

以下介绍数据显示工具可使用的一些方法：

22.1 设置单个 ui 参数

方法：tool:set_ui_option(key, value)
说明：参数 key 是一个字符串，是参数中的关键标识，参数 value 是参数的值，可是字符串，数字，或者布尔值。可使用的 key: value 如下所示:
①font_color 字体颜色，取值为十六进制颜色码(#000000~#ffffff)
②font_size 字体大小
③param 选择参数的名称
④text_content 文本内容，可以自定义文本，同时也可以使用占位符%p，%p 对应选择的参数 param
⑤pos 位置，该参数是一个表，表中需要有 x 和 y 两个参数表示位置
⑥alignment 对齐方式 ，有左对齐、居中、右对齐三种方式，分别对应的字符串为'left'、'center'、'right'

22.2 设置所有 ui 参数

方法：tool:set_ui_options(table)
说明：参数 table 是一个表，可同时设置多个 ui 参数，若未设置其中某个参数，则该参数置空，例：

```
{
  font_color = '#ff0000',
  font_size = 12,
  param = 'data',
```

```
    text_content = 'OK',
    pos = {
      x = 100,
      y = 200
    },
    alignment = 'left'
  }
```

23 ActionGuide 工具

以下介绍动作指引工具可使用的一些方法：

23.1 获取指示器状态

方法：tool:status()
说明：返回一个字符串，表示当前指示器的状态

23.2 设置指示器状态

方法：tool:set_status(status)
介绍：参数 status 一个字符串，表示要设置的指示器的状态

23.3 设置 ui 参数

方法：tool:set_option(key, value)
tool:set_option(index, key, value)
说明：参数 index 是一个整数，取值是 0 开始，表示设置第几个 roi 的 ui 参数。如果不给该参数，则设置所有 roi 的 ui 参数。参数 key 是一个字符串，是参数中的关键标识 参数 value 是参数的值，可是字符串，数字，或者布尔值。
目前可使用的 key: value 如下所示:
①color roi 颜色---取值为十六进制颜色码(#000000~#ffffff)

23.4 清除指引列表

方法：tool:clear_guide()
说明：调用后会删除当前工具所有的指引数据，无返回值。

23.5 添加矩形指引

方法：tool:add_rect_guide(left, top, right, bottom, options)

说明：参数中 left, top, right, bottom 分别是矩形的左边，上边，右边和下边，都需要是整数。参数中 options 应该是一个表，以 key 和 value 的方式指定界面所需要的参数。无返回值。

23.6 添加旋转矩形指引

方法：tool:add_rotrect_guide(center_x, center_y, width, height, angle, options)
说明：参数中 center_x, center_y, width, height, angle 分别是旋转矩形的中心 X，中心 Y，宽度，高度和旋转角度，都可以是浮点数。参数中 options 应该是一个表，以 key 和 value 的方式指定界面所需要的参数。无返回值。

23.7 添加圆形指引

方法：tool:add_circle_guide(center_x, center_y, radius, options)
说明：参数中 center_x, center_y, radius 分别是圆形的圆心 X，圆心 Y 和半径，都可以是浮点数。参数中 options 应该是一个表，以 key 和 value 的方式指定界面所需要的参数。无返回值。

23.8 添加环形指引

方法：tool:add_ring_guide(center_x, center_y, radius, range, sweep, angle, options)
说明：参数中 center_x, center_y, radius, range, sweep, angle 分别是环形的圆心 X，圆心 Y，半径，半径范围,扫描角度范围，起始角度，都可以是浮点数。参数中 options 应该是一个表，以 key 和 value 的方式指定界面所需要的参数。无返回值。

23.9 添加多边形指引

方法：tool:add_polygon_guide(points, options)
说明：参数中 points 是一个浮点数的数组，数量要求是偶数，每两个数表示一个点的 x 和 y 坐标。参数中 options 应该是一个表，以 key 和 value 的方式指定界面所需要的参数。无返回值。

23.10 设置矩形指引

方法：tool:set_rect_guide(index, left, top, right, bottom)
说明：参数中的 index 表示指定的指引的索引，从 0 开始。参数中 left, top, right, bottom 分别是矩形的左边，上边，右边和下边，都需要是整数。无返回值。

23.11 设置旋转矩形指引

方法：tool:set_rotrect_guide(index, center_x, center_y, width, height, angle)
说明：参数中的 index 表示指定的指引的索引，从 0 开始。参数中 center_x, center_y, width,

height, angle 分别是旋转矩形的中心 X，中心 Y，宽度，高度和旋转角度，都可以是浮点数。无返回值。

23.12 设置圆形指引

方法：tool:set_circle_guide(index, center_x, center_y, radius)
说明：参数中的 index 表示指定的指引的索引，从 0 开始。参数中 center_x, center_y, radius 分别是圆形的圆心 X，圆心 Y 和半径，都可以是浮点数。无返回值。

23.13 设置环形指引

方法：tool:set_ring_guide(index, center_x, center_y, radius, range, sweep, angle)
说明：参数中的 index 表示指定的指引的索引，从 0 开始。参数中 center_x, center_y, radius, range, sweep, angle 分别是环形的圆心 X，圆心 Y，半径，半径范围,扫描角度范围，起始角度，都可以是浮点数。返回值。

23.14 设置多边形指引

方法：tool:set_polygon_guide(index, points)
说明：参数中的 index 表示指定的指引的索引，从 0 开始。参数中 points 是一个浮点数的数组，数量要求是偶数，每两个数表示一个点的 x 和 y 坐标。无返回值。

24 ColorIdentify 工具

以下介绍颜色识别工具可使用的一些方法：

24.1 判断是否有匹配的颜色

方法：tool:has_match_sample()
说明：如果有匹配的颜色返回 true，否则返回 false

24.2 获取匹配颜色的名称

方法：tool:match_sample_name()
说明：返回匹配颜色的名称字符串

24.3 获取匹配颜色的分数

方法：tool:score()
说明：返回一个 0 到 1 之间浮点数

25 HSVMonitor 工具

以下介绍 HSV 颜色检测 工具可使用的一些方法：

25.1 判断颜色是否可接受

方法：tool:accepted()
说明：可接受返回 true，否则返回 false

26 LABMonitor 工具

以下介绍 LAB 颜色检测 工具可使用的一些方法：

26.1 判断颜色是否可接受

方法：tool:accepted()
说明：可接受返回 true，否则返回 false

27 RGBMonitor 工具

以下介绍 RGB 颜色检测 工具可使用的一些方法：

27.1 判断颜色是否可接受

方法：tool:accepted()
说明：可接受返回 true，否则返回 false

28 WorldToImageTranslate 工具

以下介绍 WorldToImageTranslate 工具可使用的一些方法，该方法计算矩阵时默认世界坐标 Z 轴为零点，平面上为负方向，平面下为正方向：

28.1 清除图像坐标点

方法：tool:clear_image_points()
说明：清除添加的图像坐标点。

28.2 清除世界坐标点

方法：tool:clear_world_points()
说明：清除添加的世界坐标点。

28.3 清除源坐标点

方法：tool:clear_source_points()
说明：清除添加的源坐标点。

28.4 添加图像与世界坐标点

方法：tool:add_image_world_point(mx, my, wx, wy)
说明：mx, my 为图像坐标点，wx, wy 为世界坐标点。添加完成之后需要执行该工具计算转换矩阵。

28.5 添加源坐标点

方法：tool:add_source_point(x, y, z)
说明：x, y, z 为想要转换的世界坐标点。

28.6 获取转换结果

方法：tool:result_points()
说明：与添加源坐标点一同使用，返回世界坐标点转换后的图像坐标点数组，与源坐标点一一对应，返回格式类似{{x=,y=},{x=,y=},...}。

28.7 世界坐标转相机坐标

方法：tool:world_to_camera(x, y, z)

说明：返回一个对象，分别有 x,y,z 三个属性。

28.8 相机坐标转图像坐标

方法：tool:camera_to_image(x, y, z)
说明：返回一个对象，分别有 x,y 两个属性。

28.9 清除相机参数

方法：tool:clear_camera_params()
说明：如果设置了相机的参数，则需要调用此函数来清除，才会从关联工具中获取相机参数。如果从来没有设置过相机参数，则不需要调用此函数。

28.10 设置相机参数

方法：tool:set_camera_params(fx, fy, cx, cy, k1, k2)
说明：设置了相机参数，则不会从关联的工具中获取相机参数。如需要从关联工具中获取相机参数，则需要调用 clear_camera_params()。

29 ScanCode 工具

该工具在获取扫码结果前至少需要被执行两次，第一次执行的作用是通知上位机需要进行扫码，执行后进行扫码，第二次执行可以获取扫码的结果。以下介绍扫码工具可使用的一些方法。

29.1 获取扫码结果

方法：tool:scanned_code()
说明：成功返回扫码结果，以字符串形式表示。失败则返回 nil。

30 EndTimeMark 工具

以下介绍开始时间记录/结束时间记录工具可使用的一些方法。

30.1 获取时间间隔

方法：tool:delta()
说明：返回的是一个整数，表示与前一个最接近的开始时间记录到现在的时间间隔，单位是毫秒。

31 VideoRecordTool 工具

在所有脚本中，可以使用 VideoRecordTool()来创建一个工具实例，比如：local tool = VideoRecordTool()

随后，可以调用下面的函数来设置工具的参数，需要的时候可以调用 tool:exec(ctx)来执行工具。

31.1 保存视频文件

方法：tool:save_video(file_name)

说明：参数中 file_name 可以是绝对路径，也可以是相对路径，文件名可以有.mp4 后缀，或者不带后缀，如果不带.mp4 后缀，工具会自动添加.mp4 后缀。

注：正常会在工序开始的时候开始录制，工序结束的时候停止，如果在录制的过程中调用了 save_video，会停止录制，然后把文件上传。