

涂胶工艺包

1.2

中科新松有限公司

2025 年 09 月 25 日

Contents

1	涂胶工艺包简介	1
1.1	适配说明	1
1.2	设备通讯方式	2
1.2.1	DeviceNet	2
1.2.2	Profinet	4
1.3	工艺包安装与卸载	4
2	涂胶工艺包设置	13
2.1	胶机设置	13
2.2	参数设置	15
2.3	状态 & 操作	16
2.4	涂胶信号	18
2.5	涂胶检测	19
2.6	涂胶跟踪	20
3	涂胶编程	22
3.1	涂胶色块	22
3.1.1	Dispinit	22
3.1.2	DispPurge	23
3.1.3	DispSetProg	24
3.1.4	DispSetParam	25
3.1.5	DispProcessStart	26
3.1.6	DispProcessEnd	27
3.1.7	DispErrorCheck	28
3.1.8	DispWaitPrepressure	29
3.1.9	ResetVisionDetect	30
3.1.10	RobotStart	31
3.1.11	RobotStop	32
3.1.12	CameraStart	33
3.1.13	CameraStop	34
3.1.14	WaitVisionReady	35
3.1.15	Setpart	36
3.2	编程脚本	37
3.3	程序示例	39
4	激光跟踪器	43

4.1	介绍	43
4.2	激光通讯连接	43
4.3	激光标定	44
4.4	编程功能块	49
4.5	编程脚本	52
4.6	激光寻位	53
4.7	激光跟踪	61
4.8	异常检测	62

CHAPTER 1

涂胶工艺包简介

涂胶应用中，通常需要机器人与涂胶直接通讯，以降低通讯延时，从而达到更好的涂胶质量。涂胶工艺包分为三大部分，适配说明、设备通讯方式与工艺包安装与卸载。

1.1 适配说明

机器人的版本信息，可以在机器人上电后，在“关于”界面中查询到。“关于”界面的打开方式，请见《DUCO CORE-用户手册》第 7 章节。

v1.0 版本及以上的涂胶工艺包，需要保证机器人的软件版本在 v3.3.0 及以上。

涂胶系统包括涂胶定量机、线扫激光、视觉检测三部分。具体内容请见如下表格：

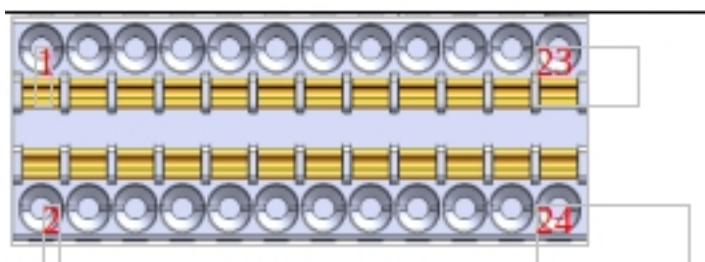
设备名称	厂商	通讯方式	备注
涂胶定量机	南京苏上	DeviceNet	使用 DeviceNet 通讯需要设置波特率与安装终端电阻，详情请见《涂胶定量机连接》章节
		Profinet	使用 Profinet 通讯需要额外配置 Profinet 主站模块。
	存融	Ethernet/IP	使用 Ethernet/IP 通讯需要额外配置 Ethernet/IP 主站模块
线扫激光	唐山英莱	TCP/IP	
	苏州全视	TCP/IP	
	北京创想	TCP/IP	
视觉检测	易思维	Profinet	使用 Profinet 通讯需要额外配置 Profinet 主站模块。
	瑞沃斯	Profinet	使用 Profinet 通讯需要额外配置 Profinet 主站模块。

1.2 设备通讯方式

1.2.1 DeviceNet

使用 DeviceNet 下的 CAN 的通讯时，需要参考《协作机器人用户手册（硬件部分）》，参考涂胶说明书，将两者的 CAN 通讯接口对接，为保证通讯质量，通常需要在接口处并联一个 120Ω 的终端电阻以消除信号反射。

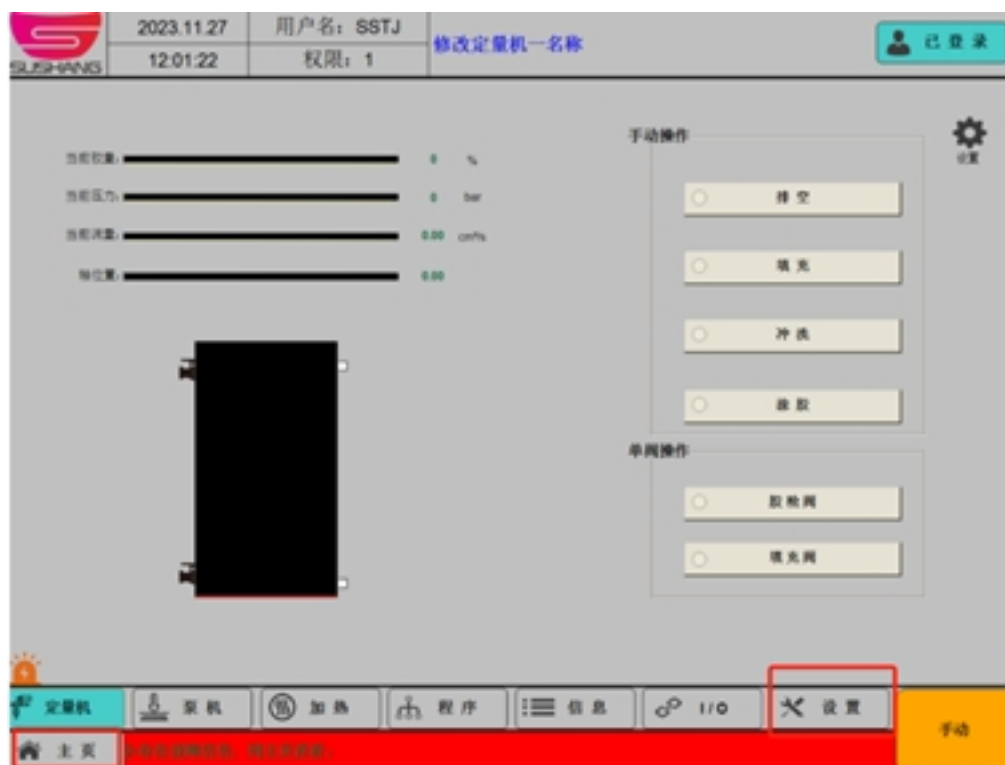
- 1、查询《协作机器人用户手册（硬件部分）》，找到 CAN 通讯接口，如下图所示：



编号	信号定义	编号	信号定义
1	AI1 (可配置模拟量输入 1)	2	AG (模拟地)
3	AI2 (可配置模拟量输入 2)	4	AG (模拟地)
5	AO1 (可配置模拟量输出 1)	6	AG (模拟地)
7	AO2 (可配置模拟量输出 2)	8	AG (模拟地)
9	24V	10	RC1 (远程开关 ON+)
11	0V	12	PowerON (远程开关 ON-)
13	A+ (INC 信号 A 相 +)	14	RC2 (远程开关 OFF+)
15	A- (INC 信号 A 相-)	16	PowerOFF (远程开关 OFF-)
17	A- (INC 信号 A 相-)	18	CAN_L
19	B- (INC 信号 B 相-)	20	CAN_H
21	Z+ (INC 信号 C 相 +)	22	485_B
23	Z- (INC 信号 C 相-)	24	485_A

从表格中可知，CAN 通讯使用引脚 18，20

2、涂胶侧，需要参考涂胶说明书，将涂胶的控制模式调至机器人控制模式，并选用相应的通讯协议。以适配南京苏上涂胶为例：





苏上胶机登录账号：SSTJ，密码：1；

点击“主页”，点击“设置”，点击“现场总线”。选择“DeviceNet”通讯。

(注：DeviceNet 通讯，需要苏上工程师提供支持)

3、DeviceNet，可配置波特率，配置信息与 CCI 页面显示同步；通讯波特率：支持 125kbps、250kbps、500kbps 三种波特率。需要注意的是，机器人与涂胶的波特率需要设置一致，才能实现通讯。

1.2.2 Profinet

使用 Profinet 通讯需要额外配置 Profinet 主站模块。

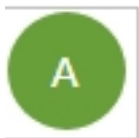
1.3 工艺包安装与卸载

涂胶工艺包文件为“dispense.plugin”的插件。安装步骤如下所示：

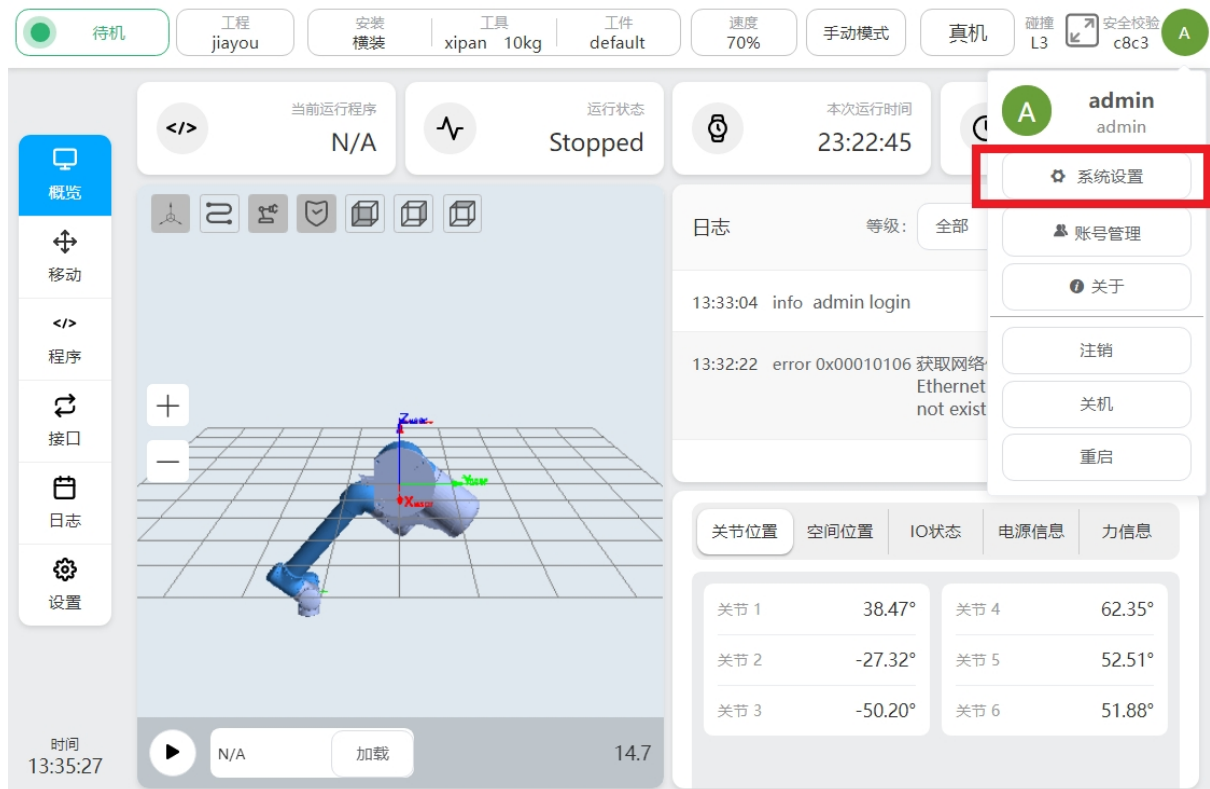
- 1、将涂胶工艺包放入 U 盘，U 盘要求格式为 FAT32
- 2、机器人系统启动后，在机器人控制柜上，USB 接口处插入 U 盘。



4、点击用户头像



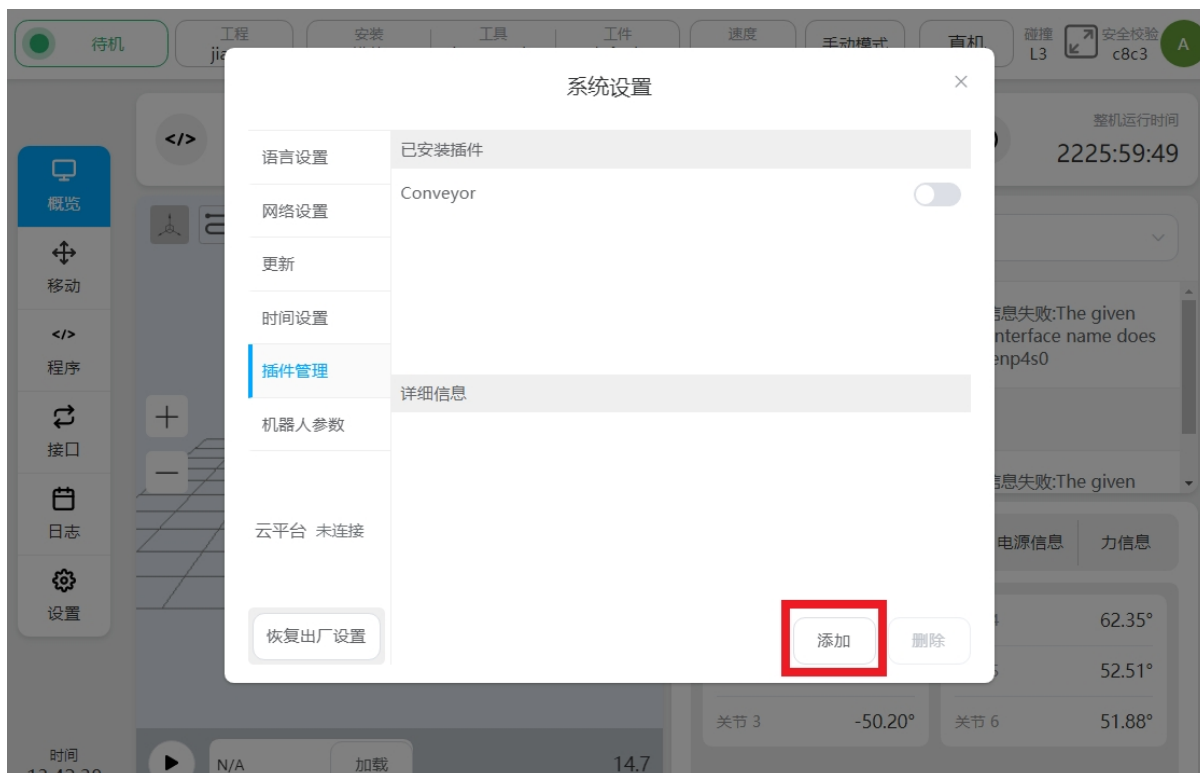
，选择系统设置。



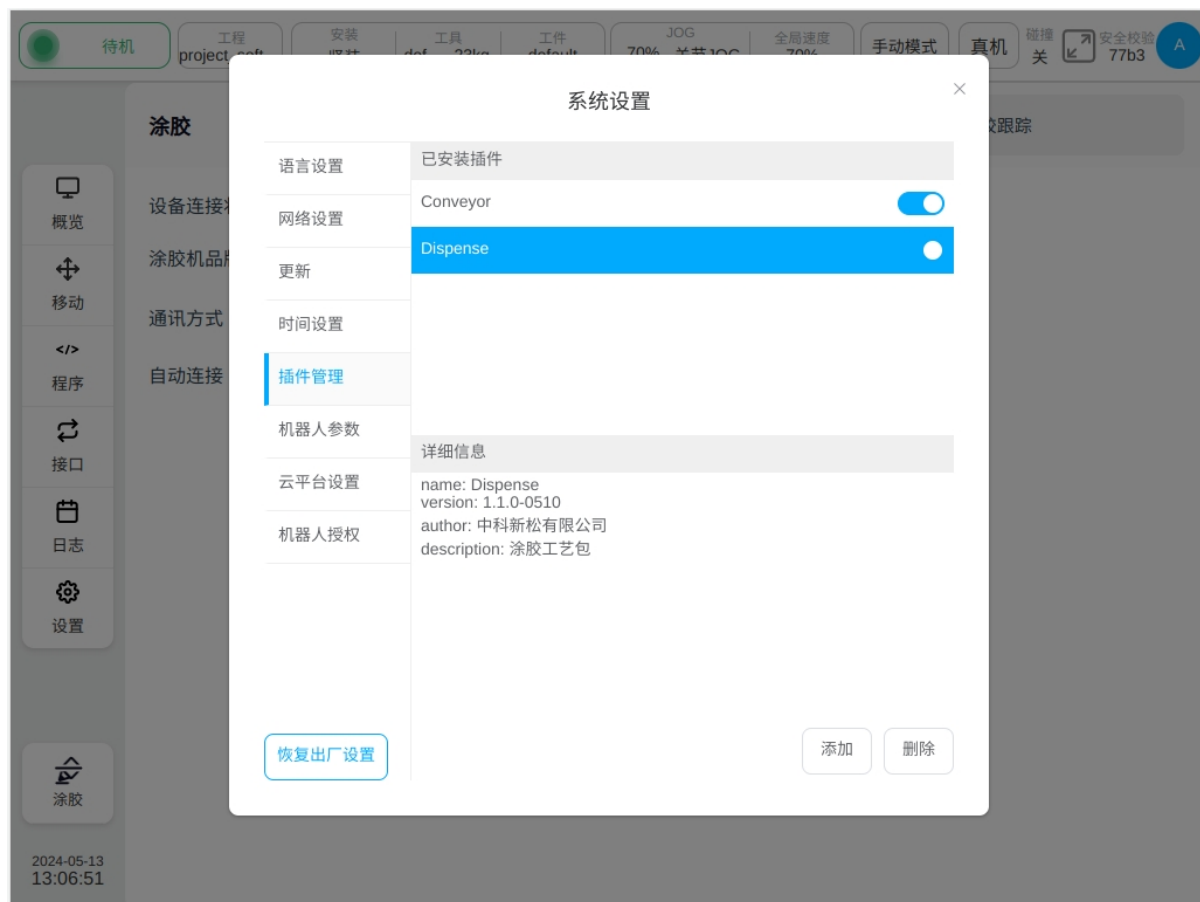
5、选择插件管理功能。



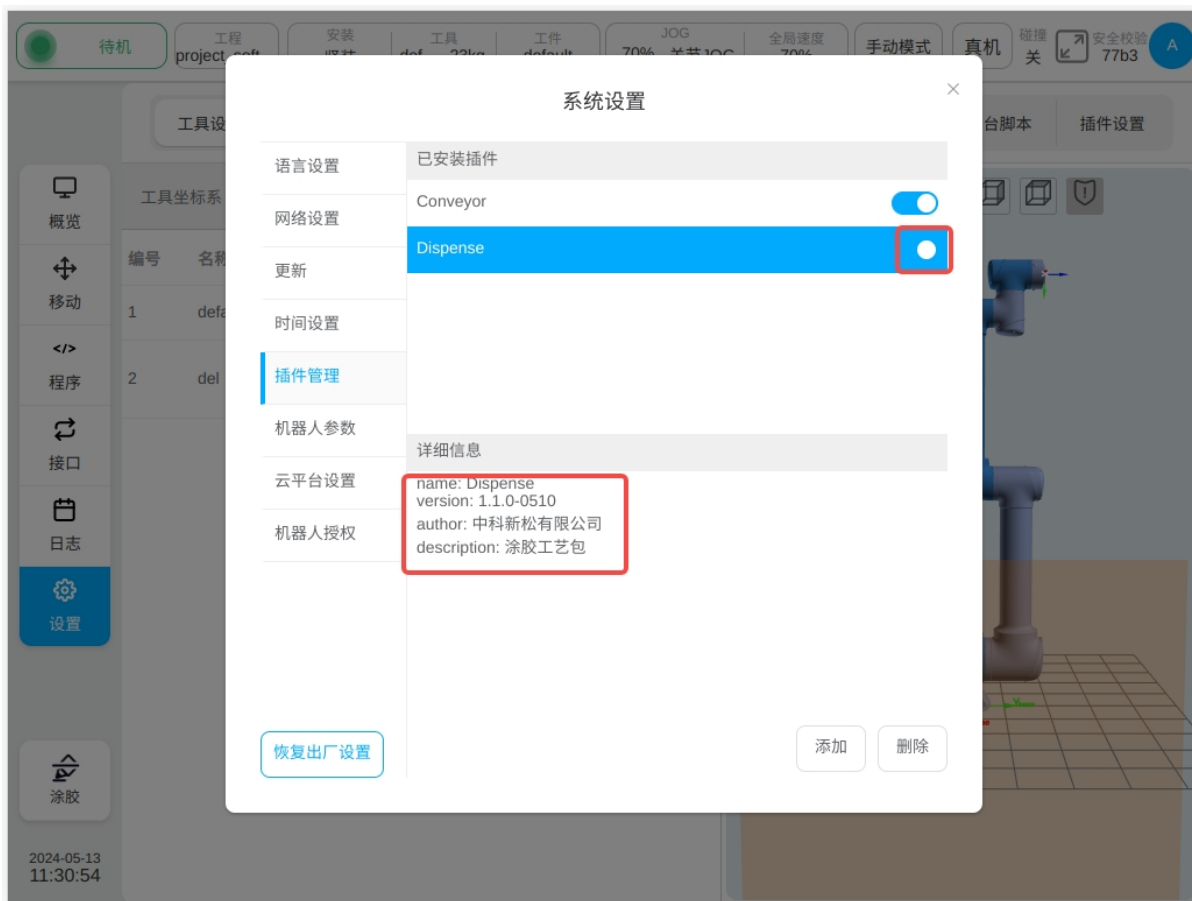
6、点击添加按钮。



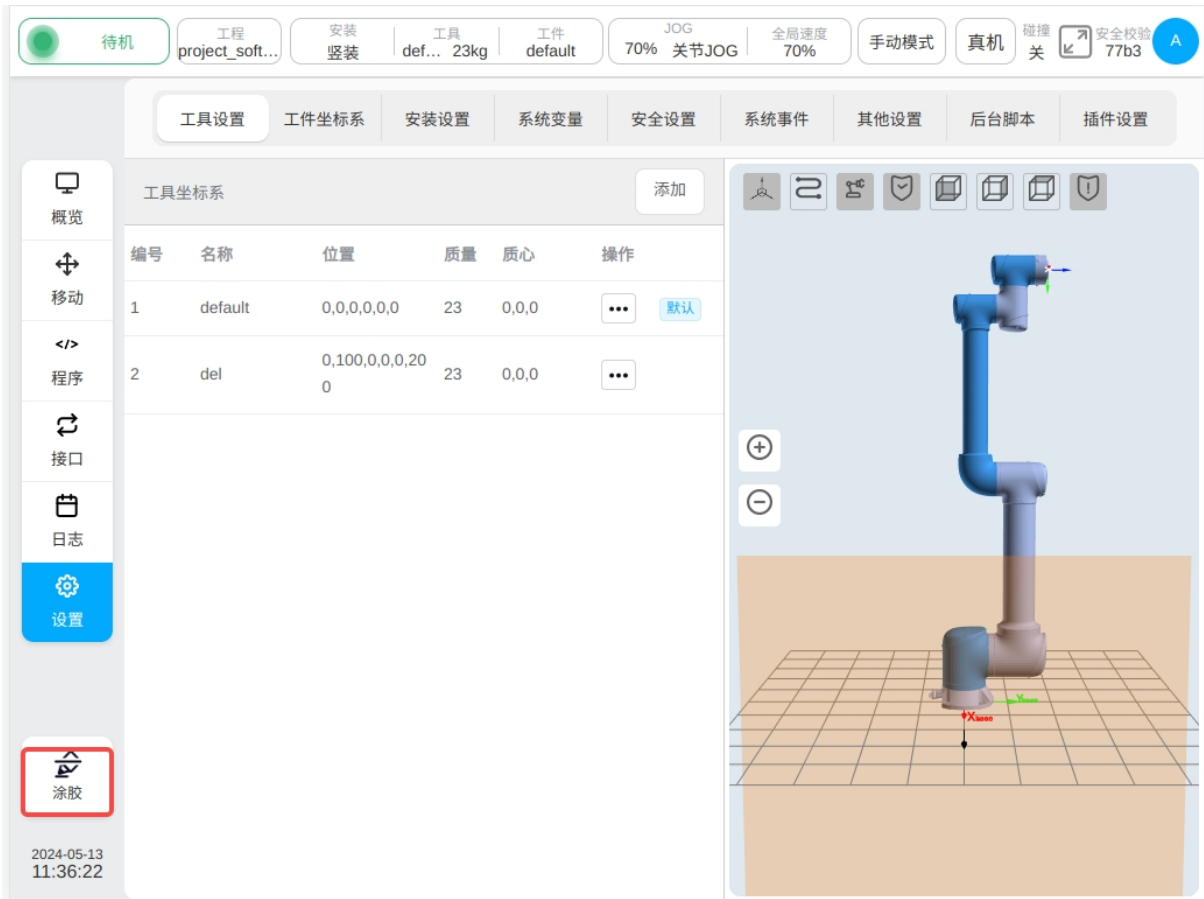
7、选择 U 盘并从文件列表中，找到并选择 “dispense.plugin” 插件包。



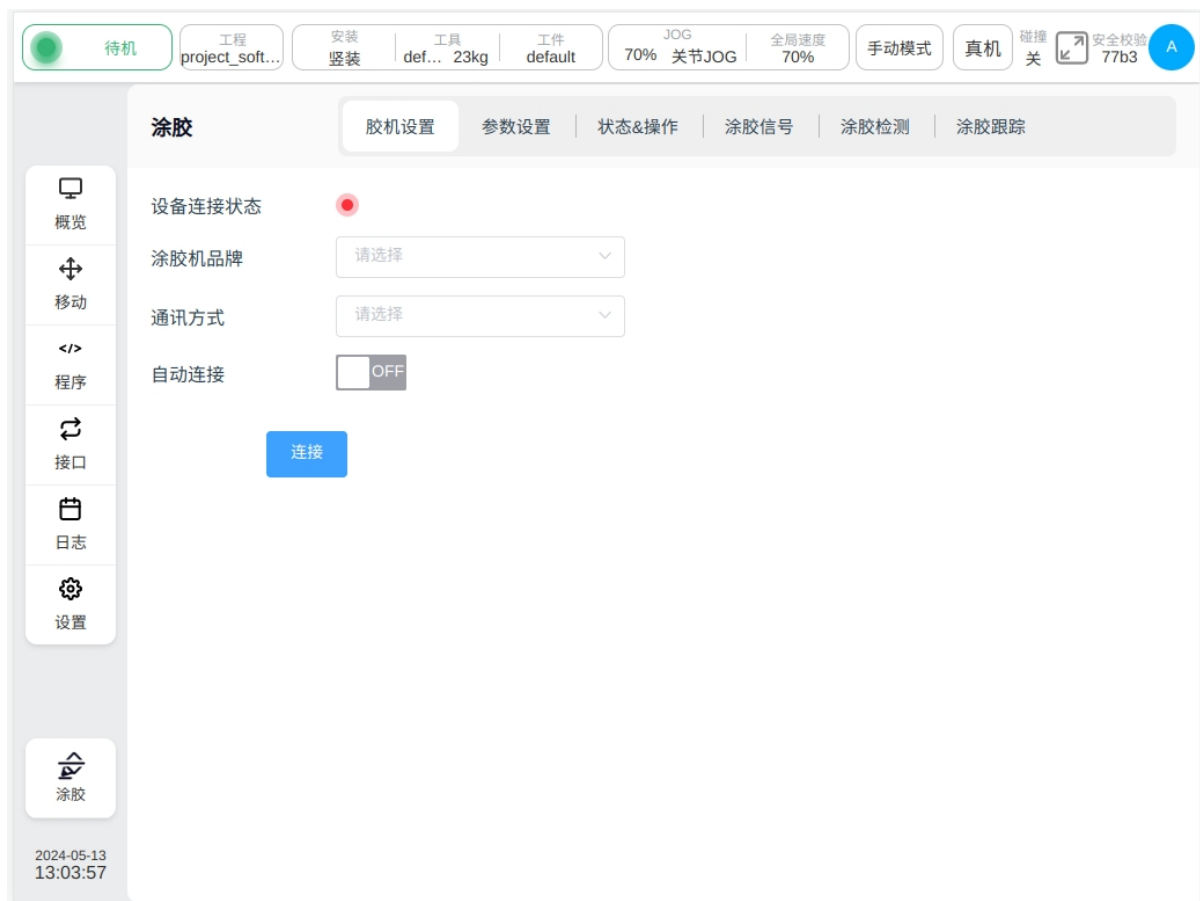
8、插件包安装完成后，如下图所示，将在标注 ① 处显示插件包安装完成。在标注 ② 处，新安装的插件包将默认为使能。在标注 ③ 处，将显示当前插件包的名称，版本信息等内容。



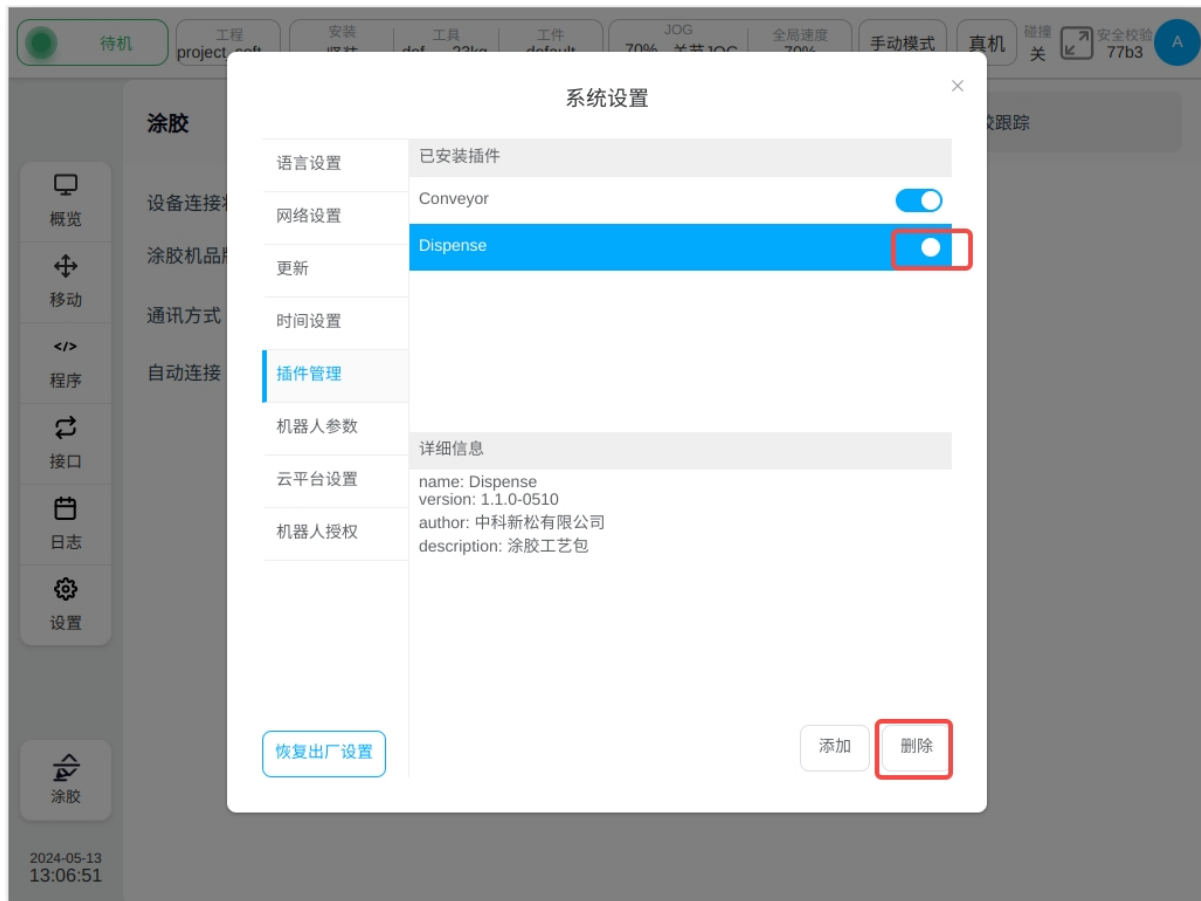
9、涂胶工艺包安装完成后，将在主页面的左下方，出现工艺包设置入口。



10、点击此工艺包设置入口按钮，即可进入涂胶工艺包设置页面。



11、已安装好的涂胶工艺包，可以通过关闭使能信号的方式，暂时失效。删除插件会弹出“保留数据”和“清空数据”。



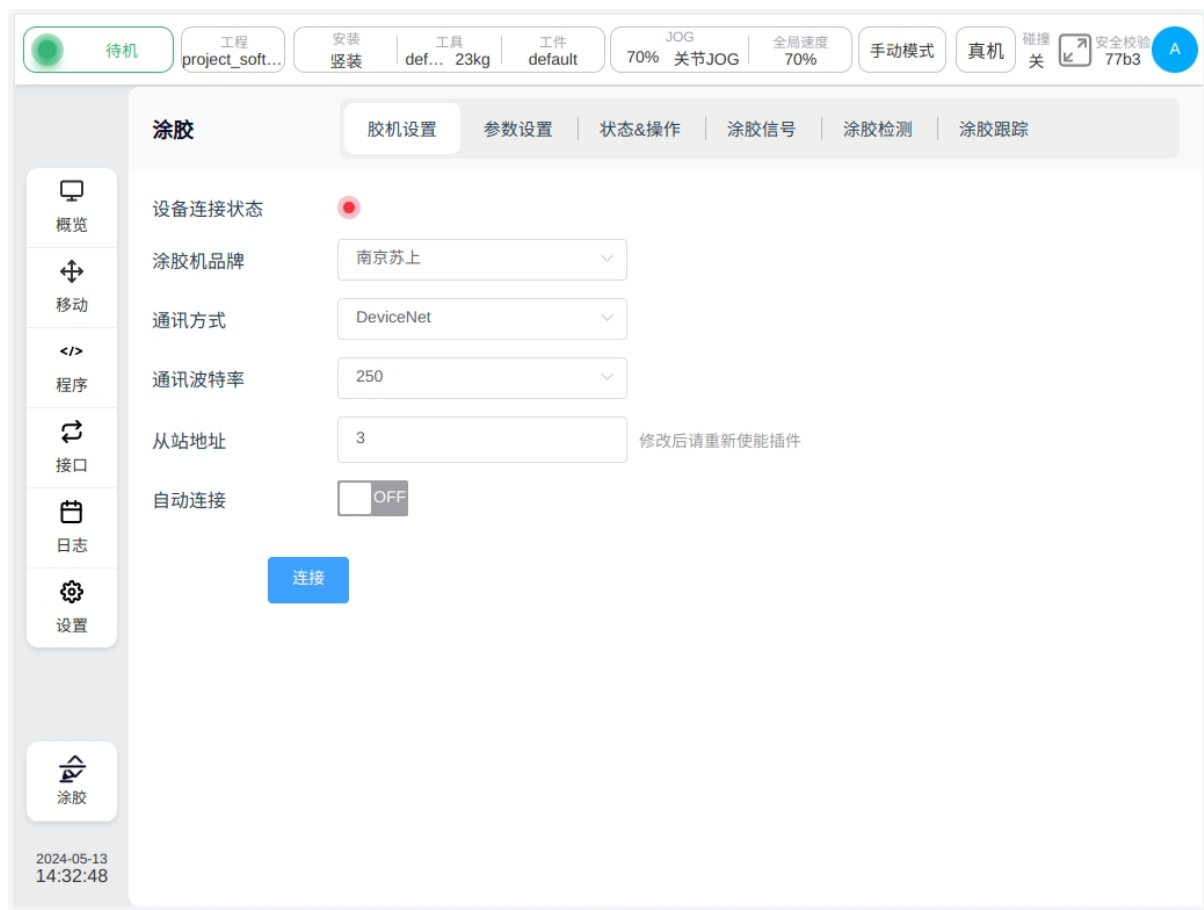
CHAPTER 2

涂胶工艺包设置

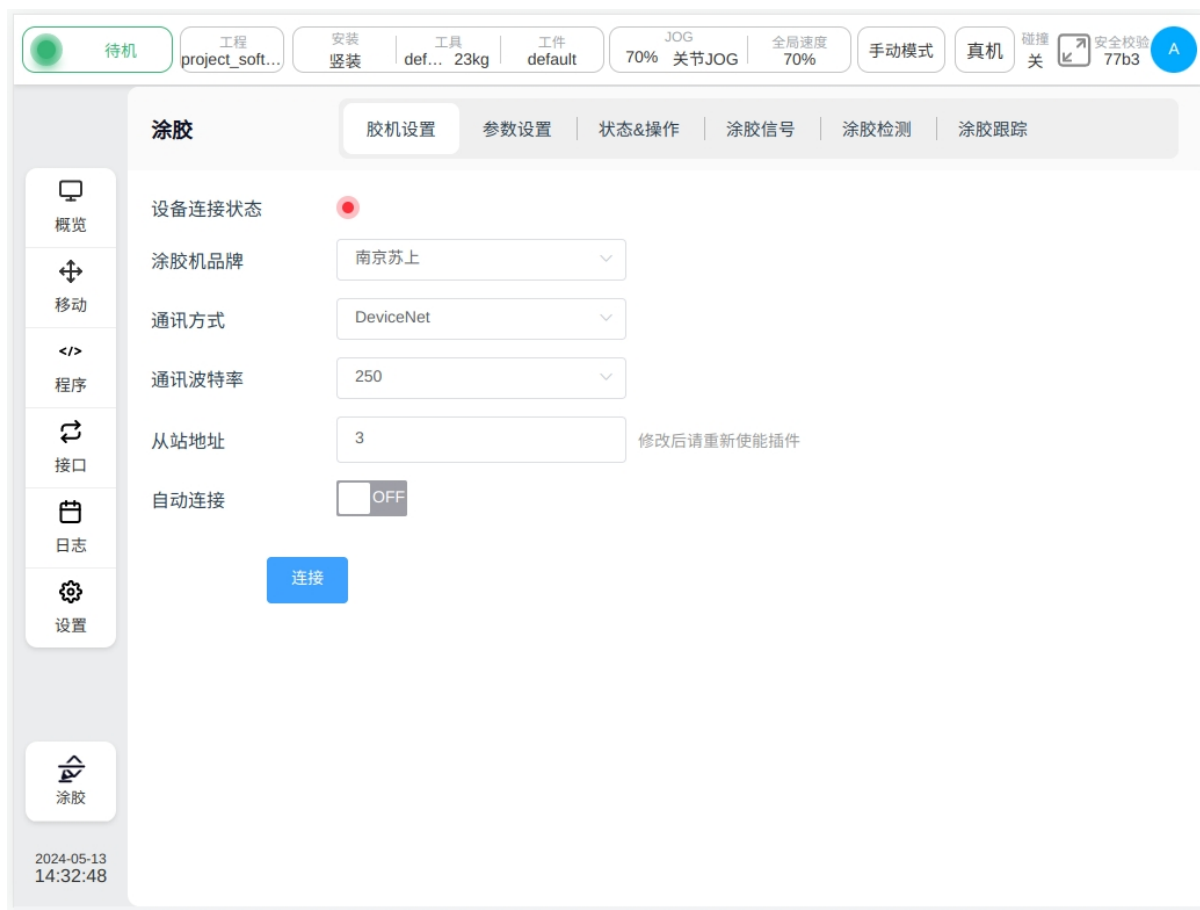
涂胶工艺包设置，包括胶机设置、参数设置、状态 & 操作、涂胶信号、涂胶检测、涂胶跟踪。

2.1 胶机设置

胶机设置用于显示通讯状态、涂胶机品牌及对应的通讯设置。



设备连接状态：当机器人与涂胶通讯建立后，状态指示灯将由红色转为绿色。如下图所示：



涂胶机品牌：用于配置当前所使用的涂胶。正确选择涂胶后，机器人将调用相应的通讯协议。

通讯方式：1、DeviceNet，可配置波特率，配置信息与 CCI 页面显示同步；2、使用 Profinet 通讯需要额外配置 Profinet 主站模块。

通讯波特率：支持 125kbps、250kbps、500kbps 三种波特率。

从站地址：选择 DeviceNet 后，修改后需要重新使能插件。

自动连接：用于机器人下次开机时，自动连接涂胶。当设置为 ON 时，机器人将保存当前的设置信息，下一次机器人开机后，将在机器人完成机械臂上电操作后，自动连接涂胶。

连接：“连接”按钮通常在首次机器人与涂胶通讯调试时使用。点击“连接”按钮，将按照涂胶的报文格式，向涂胶发起连接请求。当收到涂胶正确的反馈信息后，涂胶连接状态灯变绿。

2.2 参数设置

机器人涂胶参数设置包括以下几点：

涂胶使能：一般情况下，涂胶默认使能；在真机模式下，当关闭使能时，机器人接收到涂胶指令时，不去执行涂胶；在仿真模式下，涂胶使能关闭。

恢复策略：涂胶过程中暂停运行或者故障时暂停，需要再次运行时，需要选择恢复策略，恢复策略包括两种：1、当前涂胶段打开胶枪；2、当前段不打开。

恢复时提前开枪时间（ms）：涂胶恢复运行状态时，提前打开胶枪的时间。

暂停时关闭胶枪延时 (ms)：涂胶运行过程中暂停，关闭胶枪需要延长的时间。

详细日志打印：ON，将打印通讯数据，在日志记录里可查看；OFF，不打印通讯数据，日志没有相关信息。

设置具体界面可参考下图。



2.3 状态 & 操作

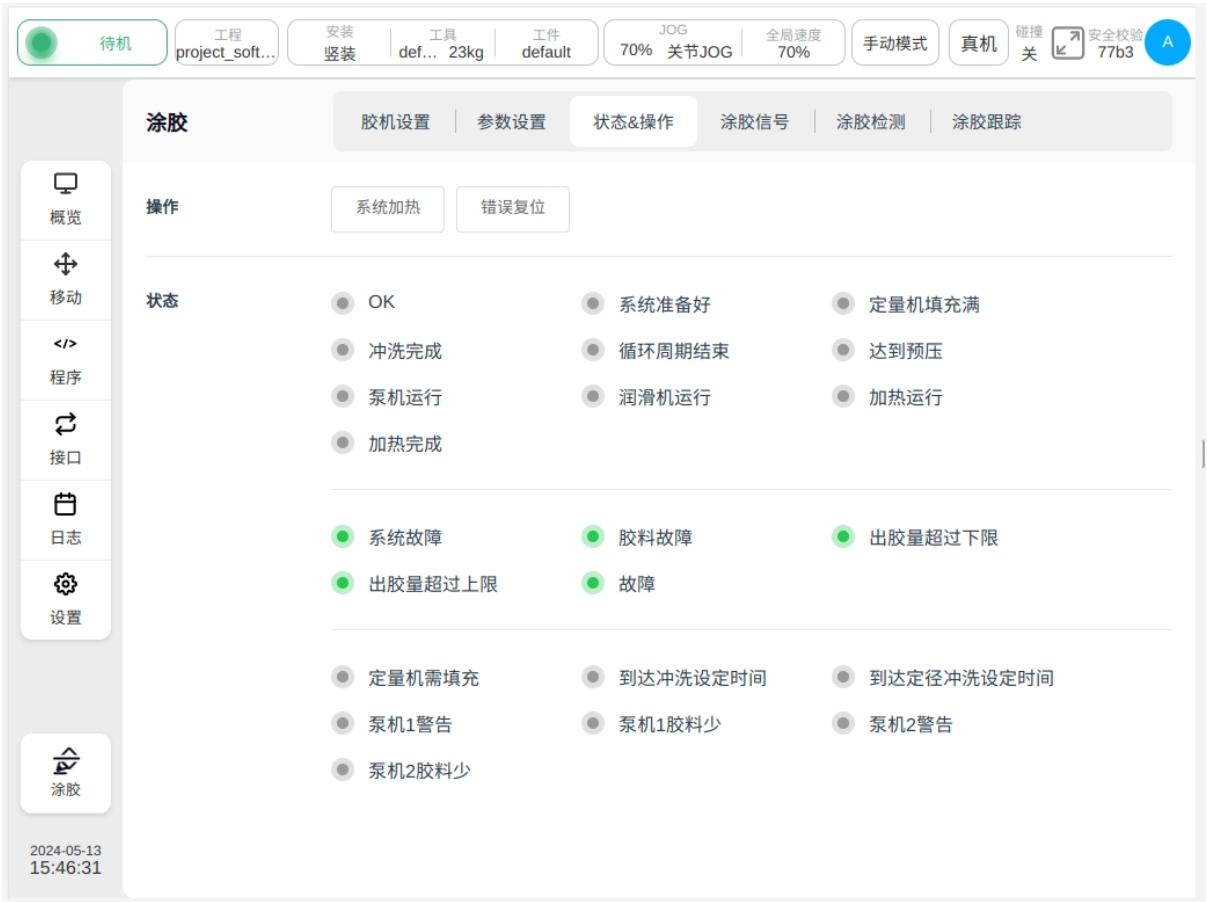
操作主要包括系统加热、错误复位。在手动模式下，两个按钮可点击；在自动模式下，两个按钮不可点击。

状态主要包括三种类型：

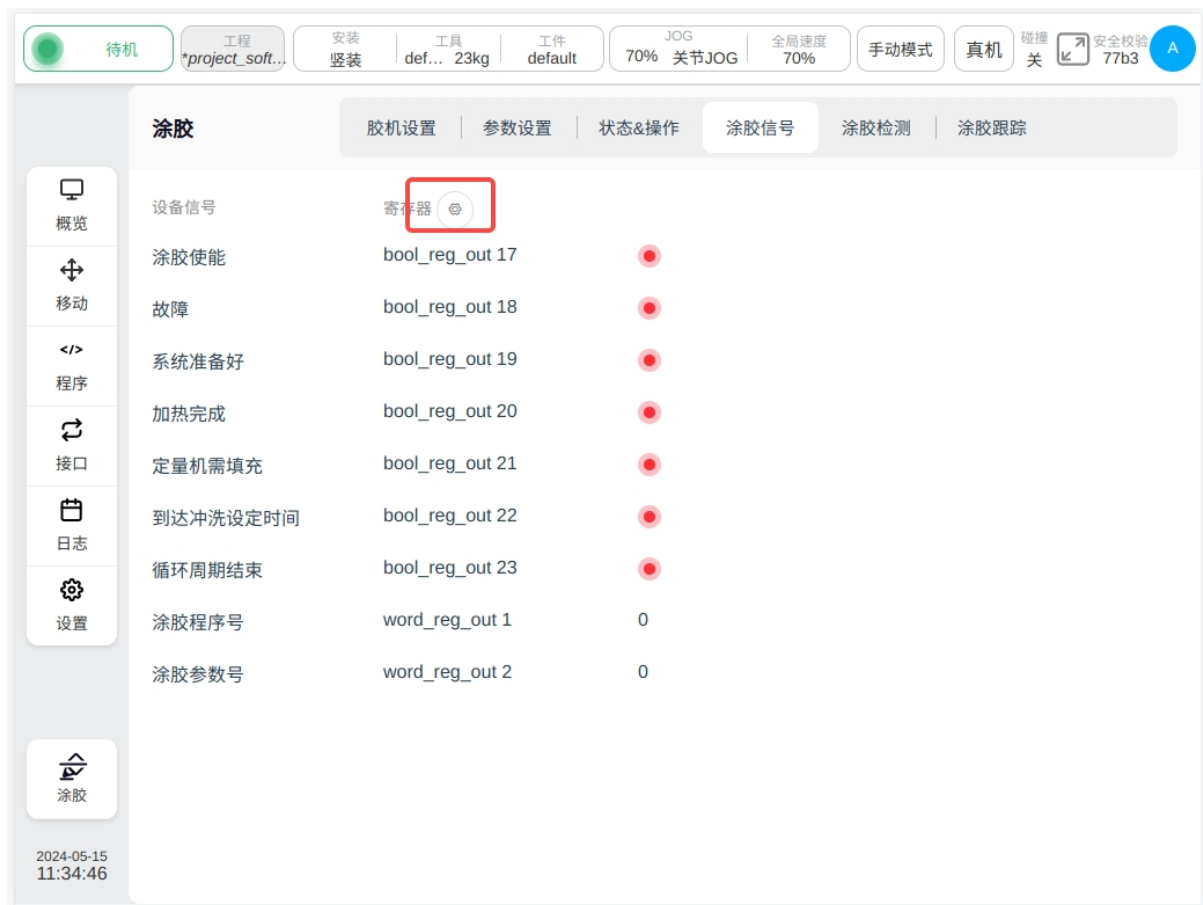
- 1、组件的运行状态，系统执行时序；
- 2、故障；
- 3、提示类：到达冲洗设定时间等；

当出现哪类状态时，状态前面的灯变成绿色，否则灰色。

具体界面可参考下图。



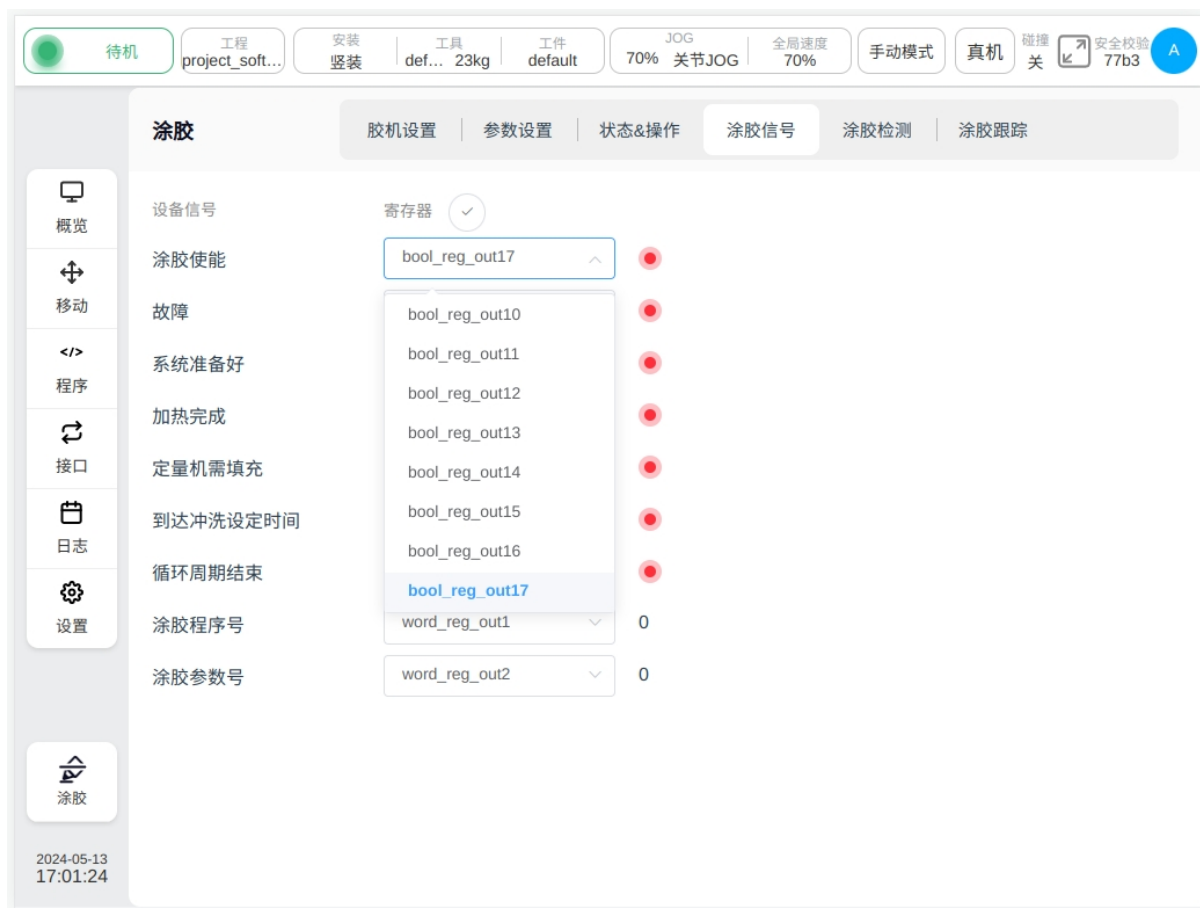
2.4 涂胶信号



涂胶信号页面用于显示涂胶信号，同时将信号配置给寄存器，以便于机器人上位控制器可以通过 profinet、modbus、TCP/IP 等方式，获取到涂胶信息。点击



即可进行寄存器配置，信号右侧将显示可配置的寄存器列表，如下图所示：



完成寄存器配置后，点击



即可保存配置。

2.5 涂胶检测

涂胶检测主要包括设备连接状态、视觉检测、通讯方式、系统状态、检测结果；当视觉检测不使用时，不显示系统状态和检测结果。

具体界面可参考下图。



2.6 涂胶跟踪

涂胶跟踪页面用于设置激光传感器厂商、连接传感器、控制传感器、标定传感器。如图，左侧区域进行传感器的连接和操作，右侧为可以查看标定参数以及进行激光传感器的标定。具体使用详见下面章节激光跟踪器



CHAPTER 3

涂胶编程

涂胶编程主要包括涂胶色块、编程脚本、程序示例。

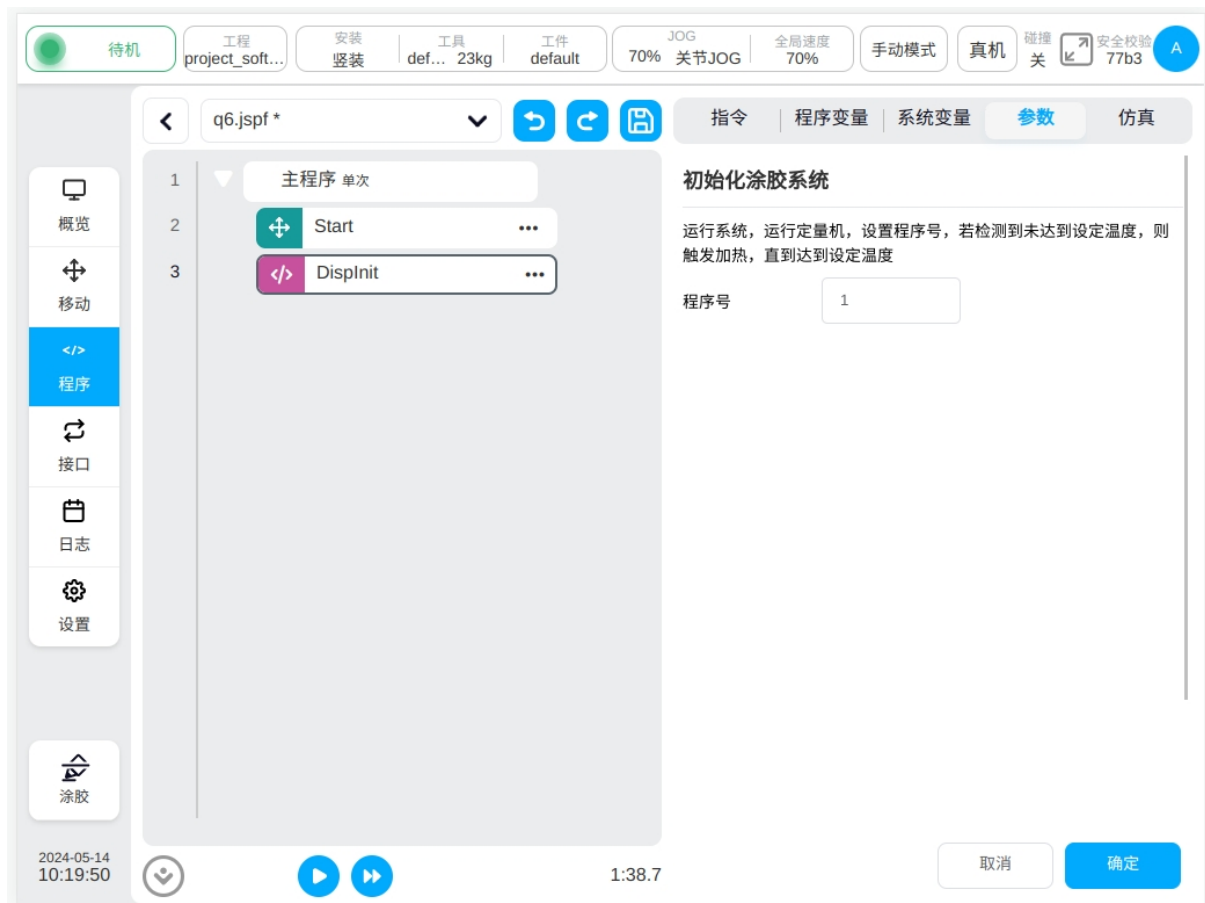
3.1 涂胶色块

涂胶色块主要包括: Dispinit、DispPurge、DispSetProg、DispSetParam、DispProcessStart、DispProcessEnd、DispErrorCheck、DispWaitPrepressure、ResetVisionDetect、RobotStart、RobotStop、CameraStart、CameraStop、WaitVisionReady、Setpart; 其中 ResetVisionDetect、RobotStart、RobotStop、CameraStart、CameraStop、WaitVisionReady、Setpart 为涂胶检测色块。

3.1.1 Dispinit

Dispinit 主要用来设置涂胶系统的初始化。

配置页面可参考下图。



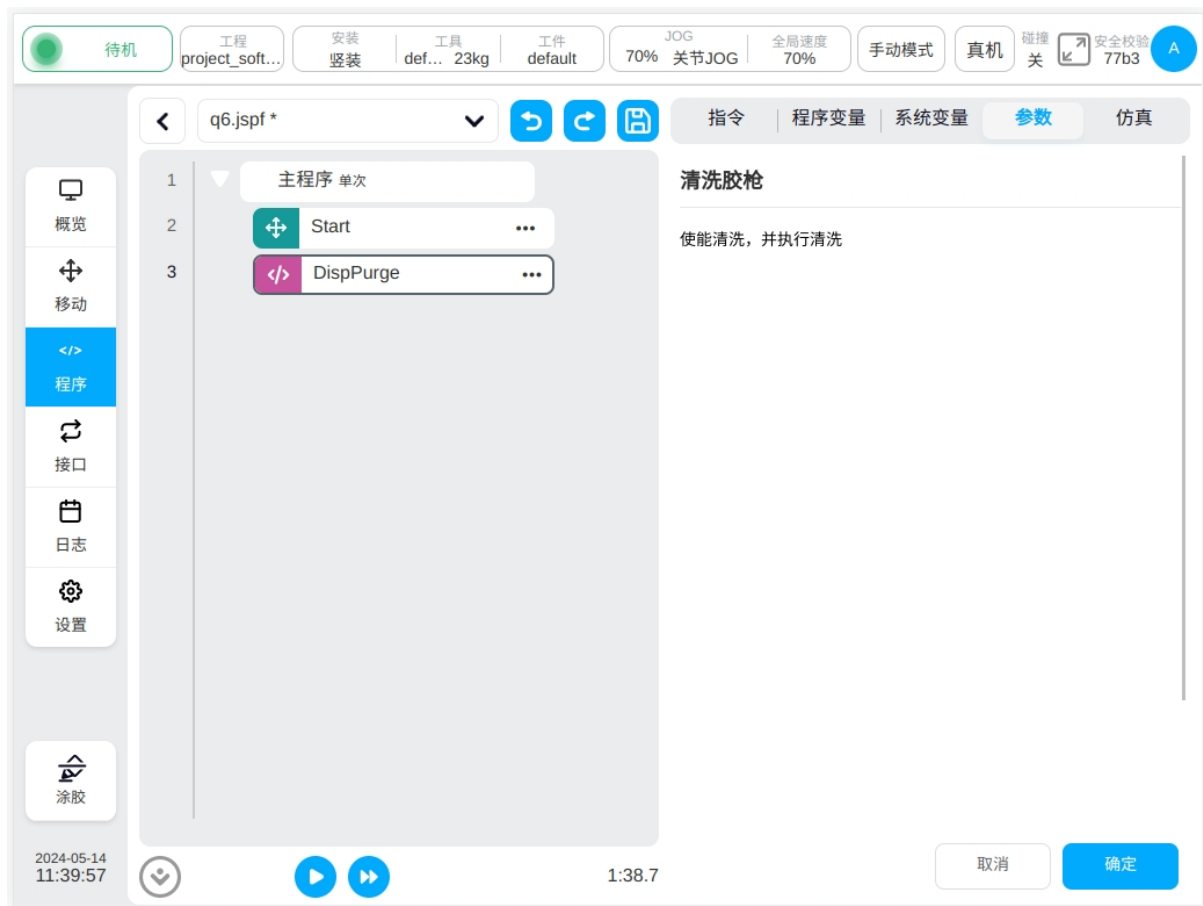
有如下设置项：程序号

机器人通过程序号去选择胶机定义的参数。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.2 DispPurge

DispPurge 主要用来设置清洗胶枪。

配置页面可参考下图。

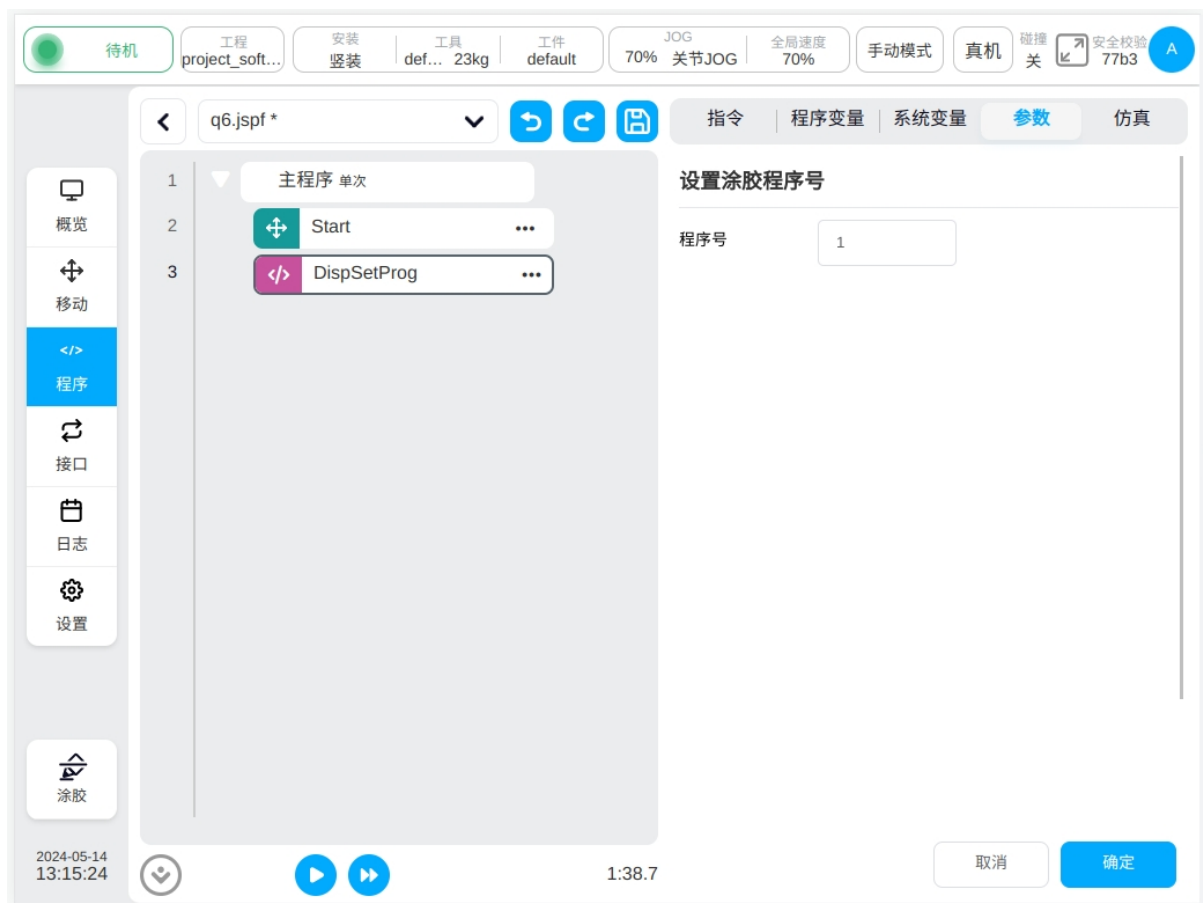


此模块功能：在运行程序前，检查胶机的请求清洗信号，然后把胶枪里面的胶排空。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.3 DispSetProg

DispSetProg 主要用来设置涂胶程序号

配置页面可参考下图。



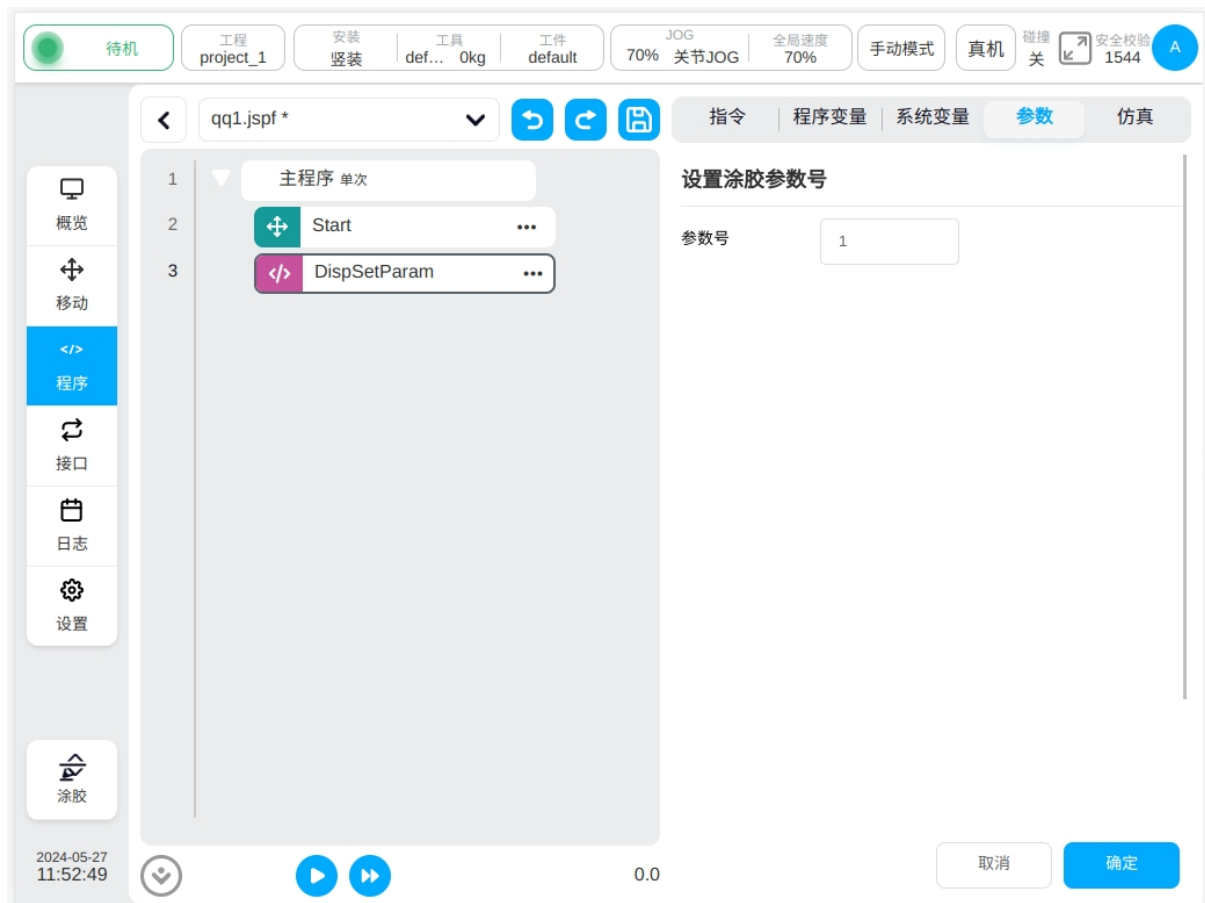
有如下设置项：程序号

机器人通过程序号去选择胶机定义的参数。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.4 DispSetParam

DispSetParam 主要用来设置涂胶参数号。

配置页面可参考下图。



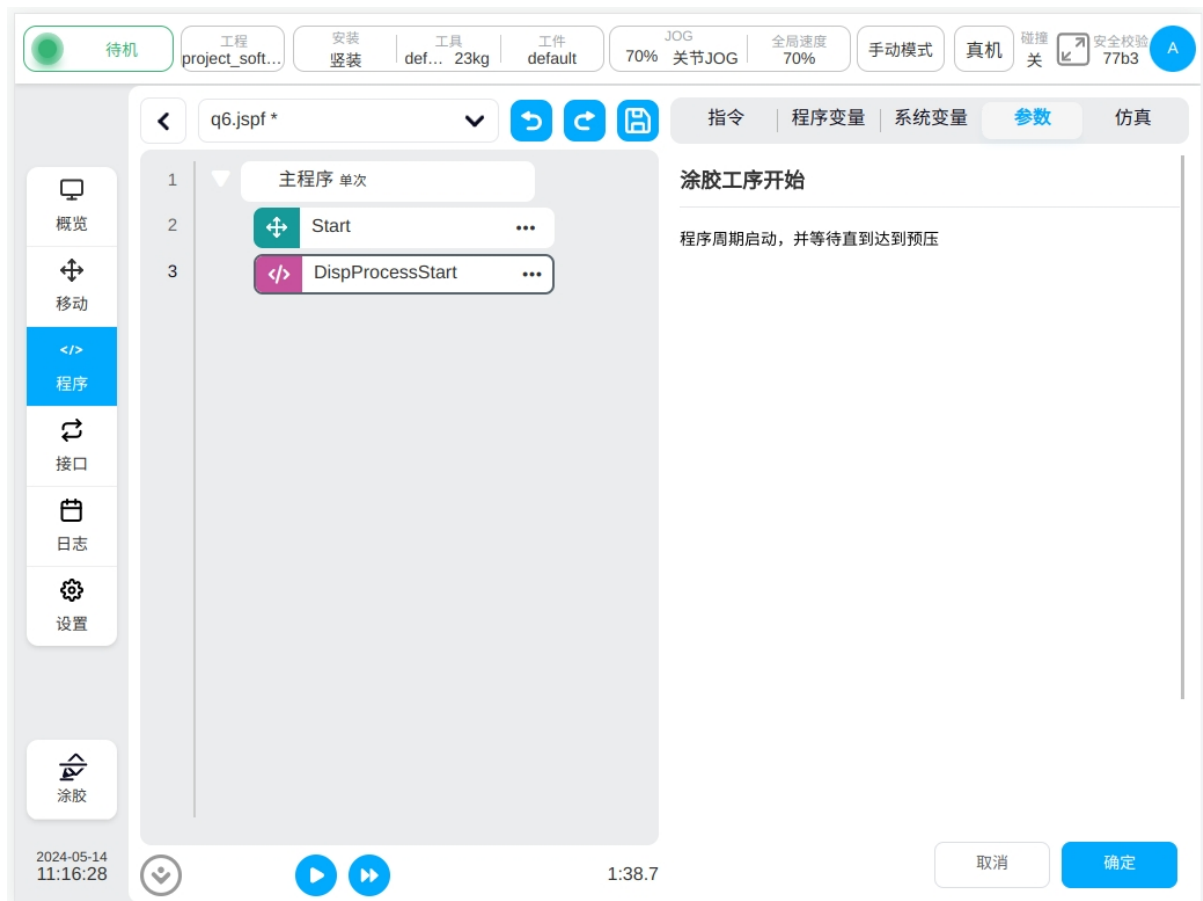
有如下设置项：参数号

机器人通过参数号去选择胶机定义的参数。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.5 DispProcessStart

DispProcessStart 主要用来设置涂胶工序开始。

配置页面可参考下图。



程序周期启动，并等待直到达到预压之后。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.6 DispProcessEnd

DispProcessEnd 主要用来设置涂胶工序结束。

配置页面可参考下图。



程序周期结束，并确认胶机结束循环。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.7 DispErrorCheck

DispErrorCheck 主要用来设置涂胶故障处理。

配置页面可参考下图。

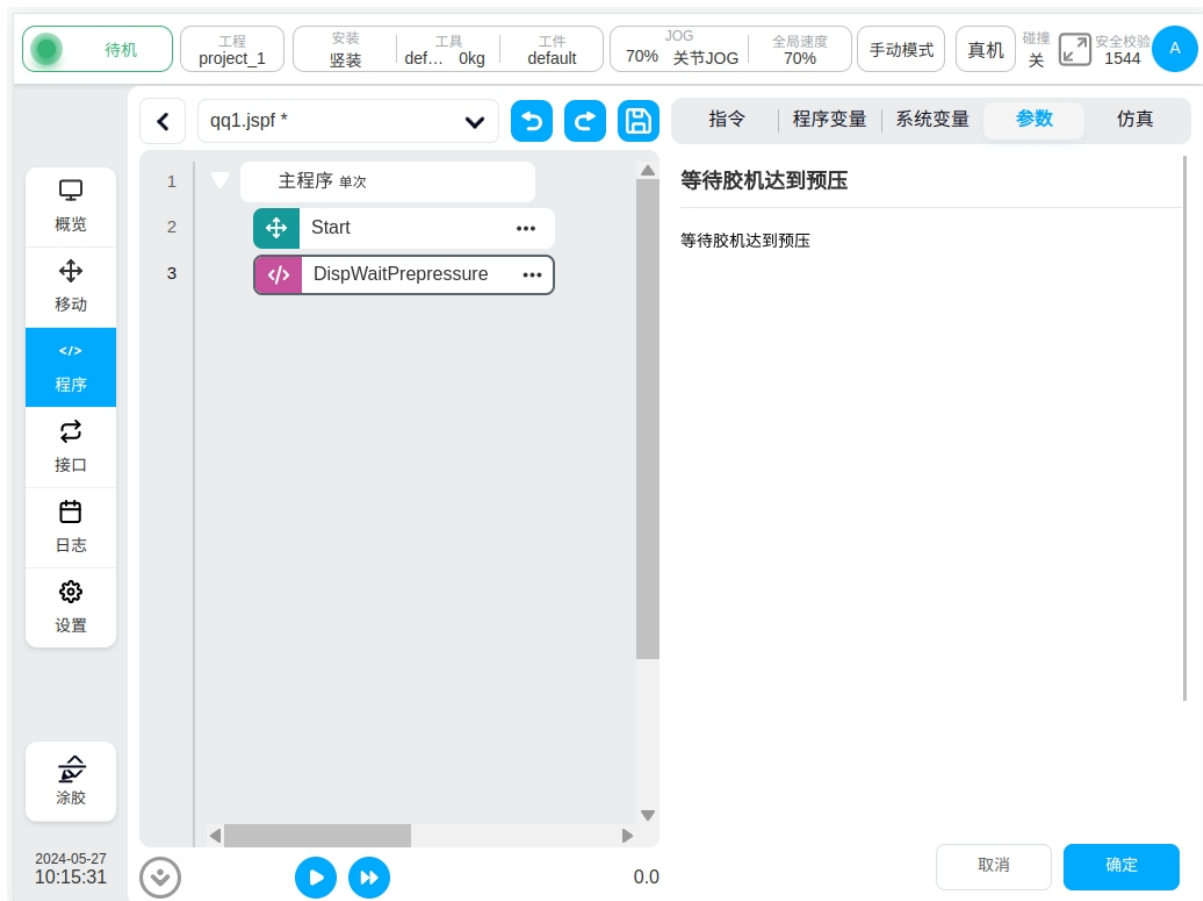


检查是否存在故障，若存在故障，则进行故障复位，1s 后检测仍有故障，则提示。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.8 DispWaitPrepressure

DispWaitPrepressure 主要用来设置等待胶机达到预压之后。

配置页面可参考下图。

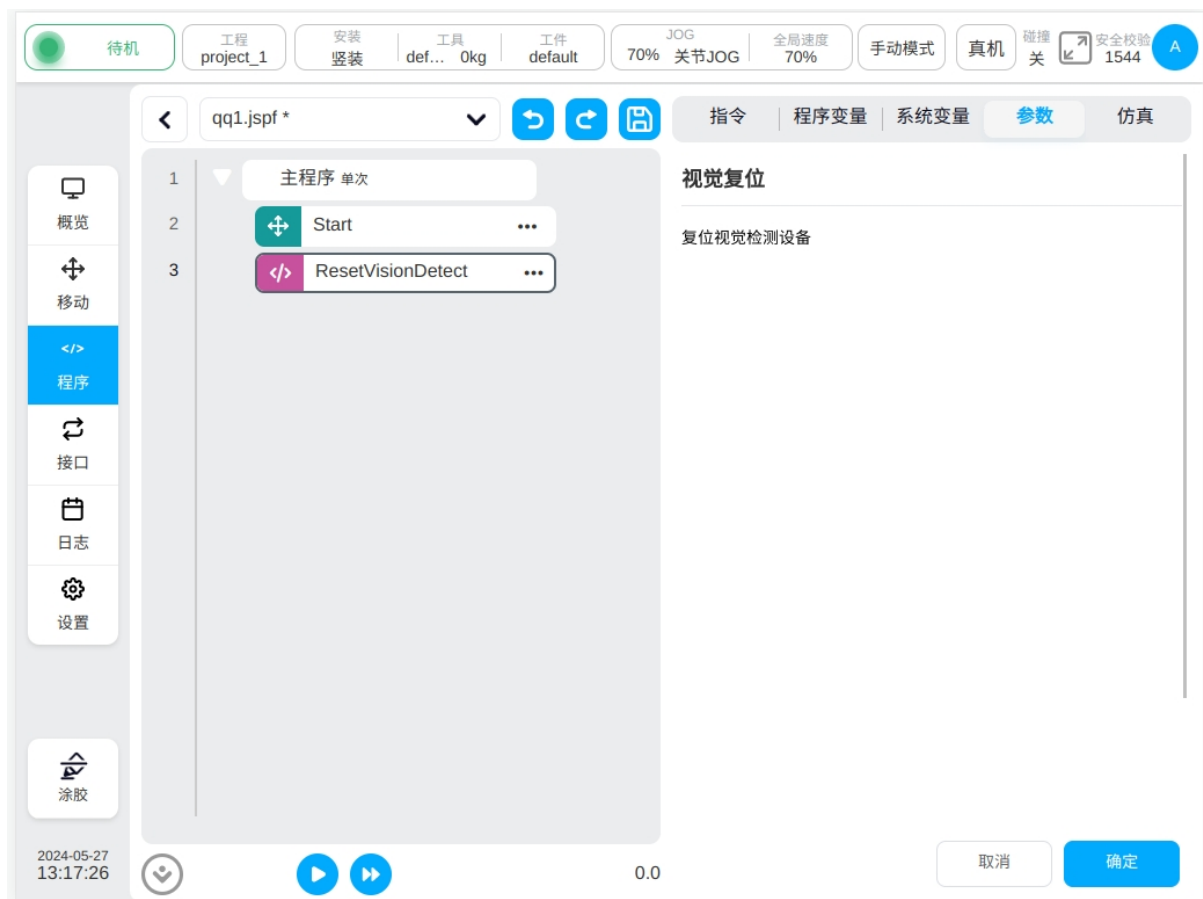


等待胶机达到预压。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.9 ResetVisionDetect

ResetVisionDetect 主要用来设置视觉检测信号复位。

配置页面可参考下图。

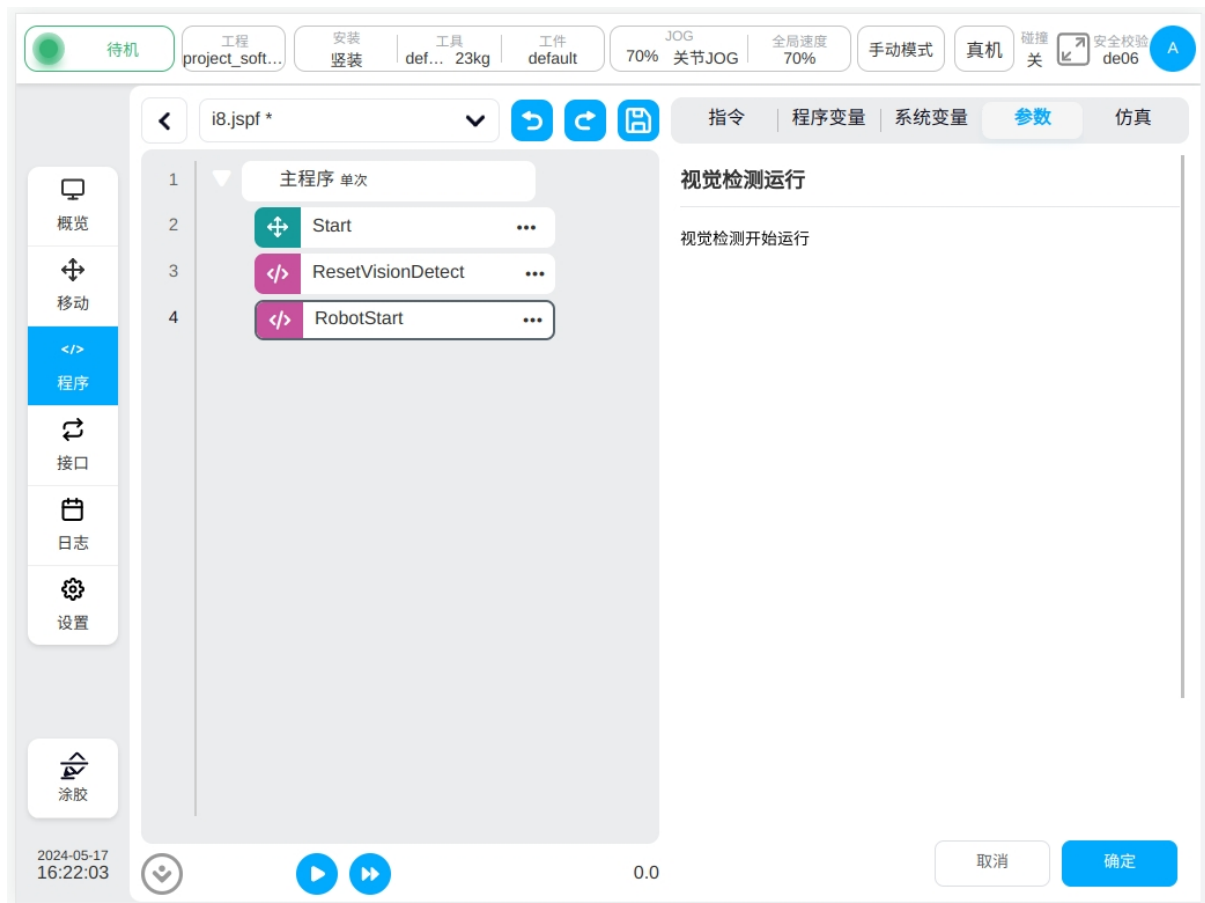


复位视觉检测设备。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.10 RobotStart

RobotStart 主要用来设置视觉检测运行。

配置页面可参考下图。

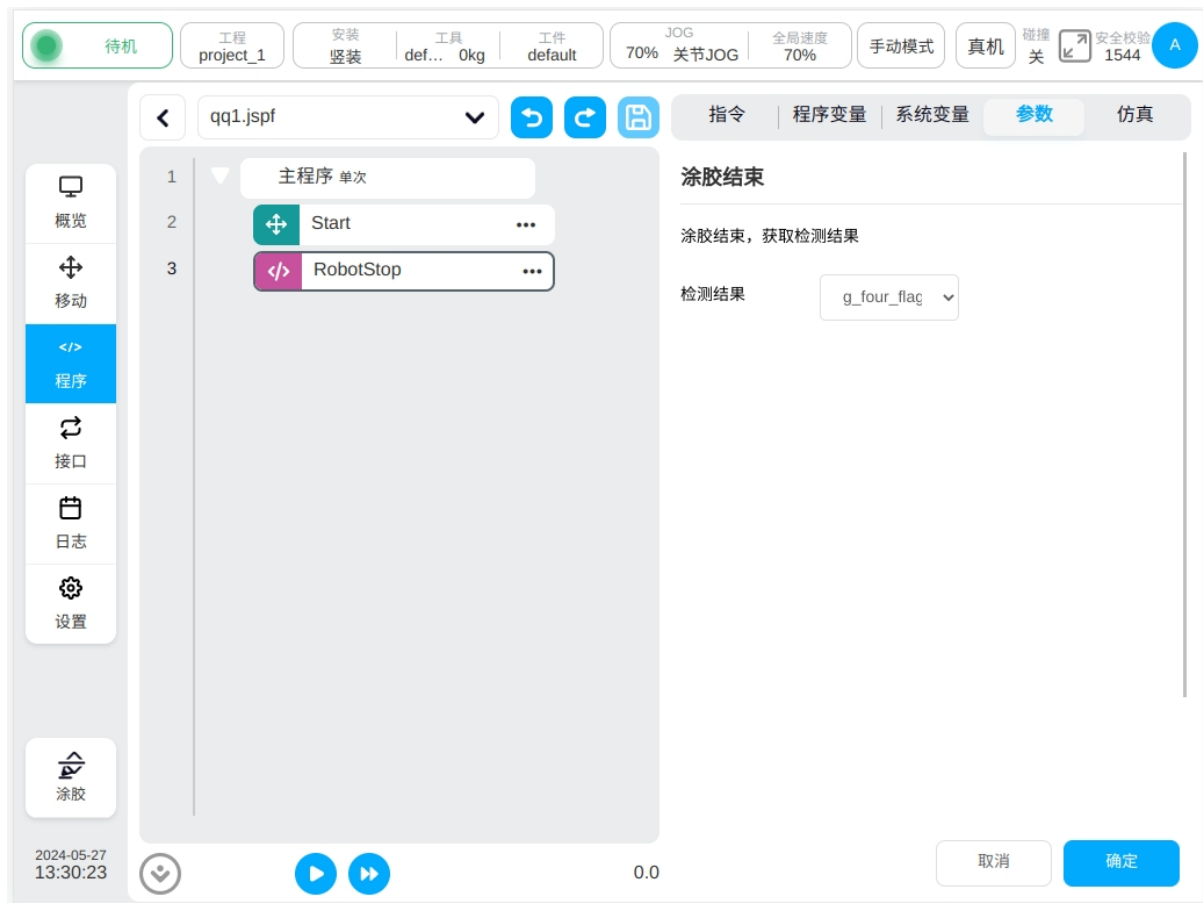


视觉检测开始运行。点击确定，参数页面保存；点击取消，回到指令页面。

3.1.11 RobotStop

RobotStop 主要用来设置涂胶结束。

具体参数页面可参考下图。



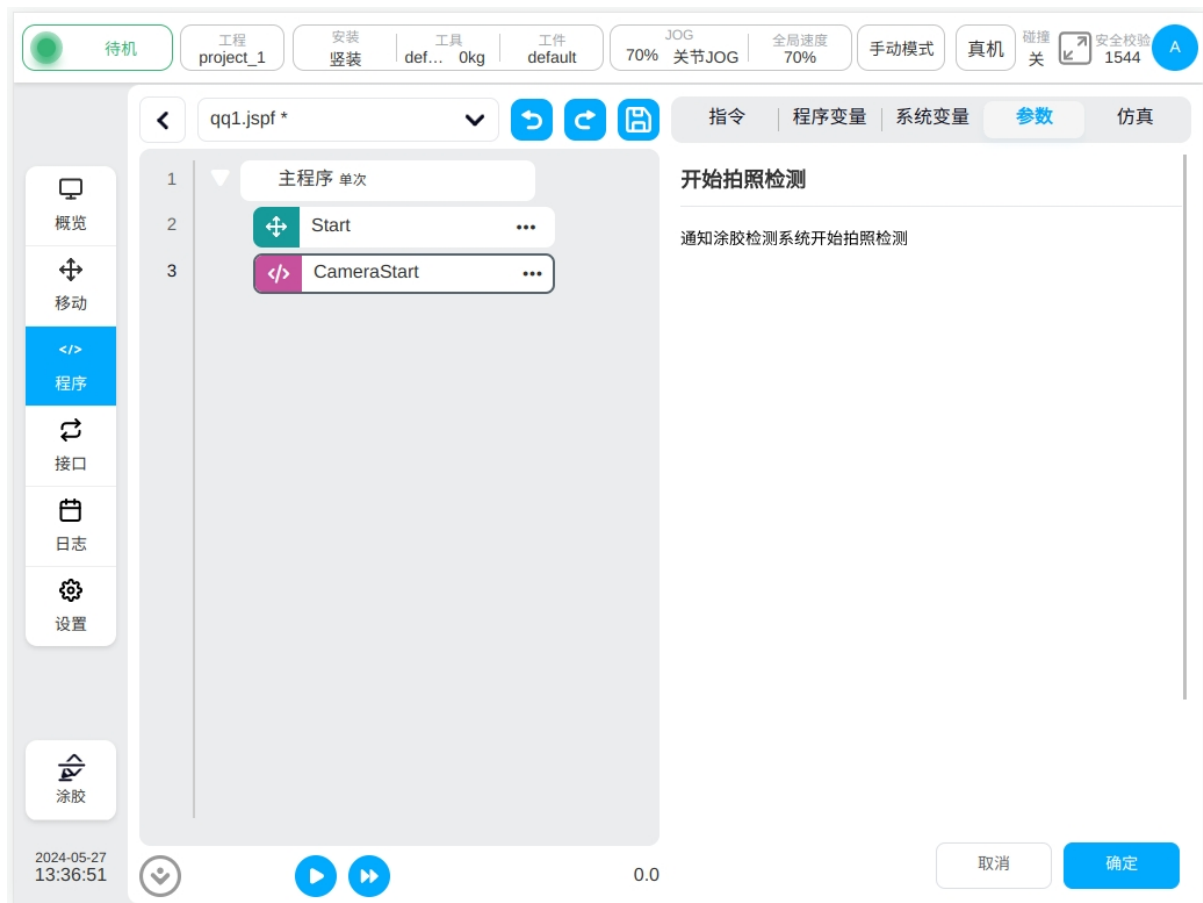
有如下设置项：检测结果，下拉框可选择程序变量和全局变量。

检测结果编辑完成，点击确定，保存参数页面；点击取消，回到指令页面。

3.1.12 CameraStart

CameraStart 主要用来设置开始拍照检测

配置页面可参考下图。



通知涂胶检测系统开始拍照检测。点击确定，保存参数页面；点击取消，回到指令页面。

3.1.13 CameraStop

CameraStop 主要用来设置停止拍照检测。

配置页面可参考下图。

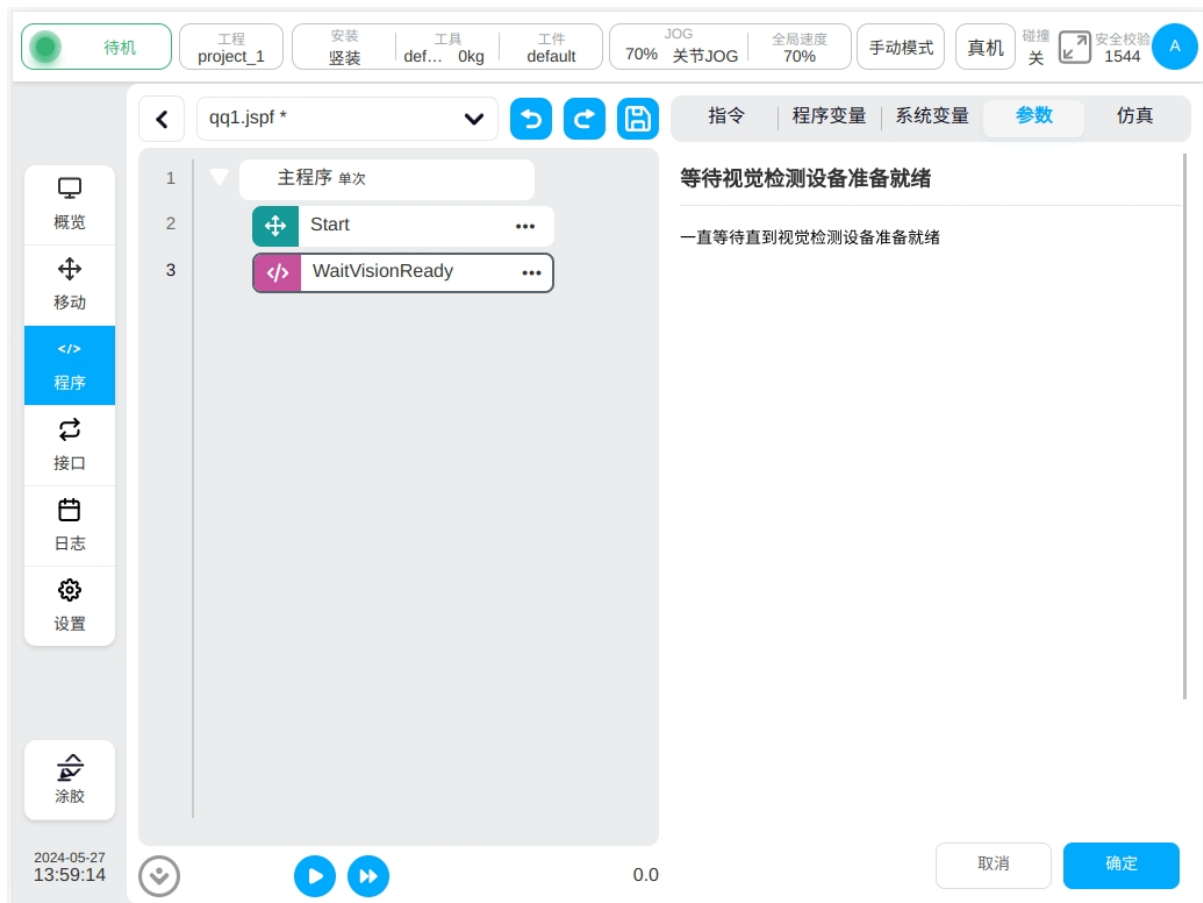


通知涂胶检测系统停止拍照检测。点击确定，保存参数页面；点击取消，回到指令页面。

3.1.14 WaitVisionReady

WaitVisionReady 主要用来设置等待视觉检测设备准备就绪。

配置页面可参考下图。

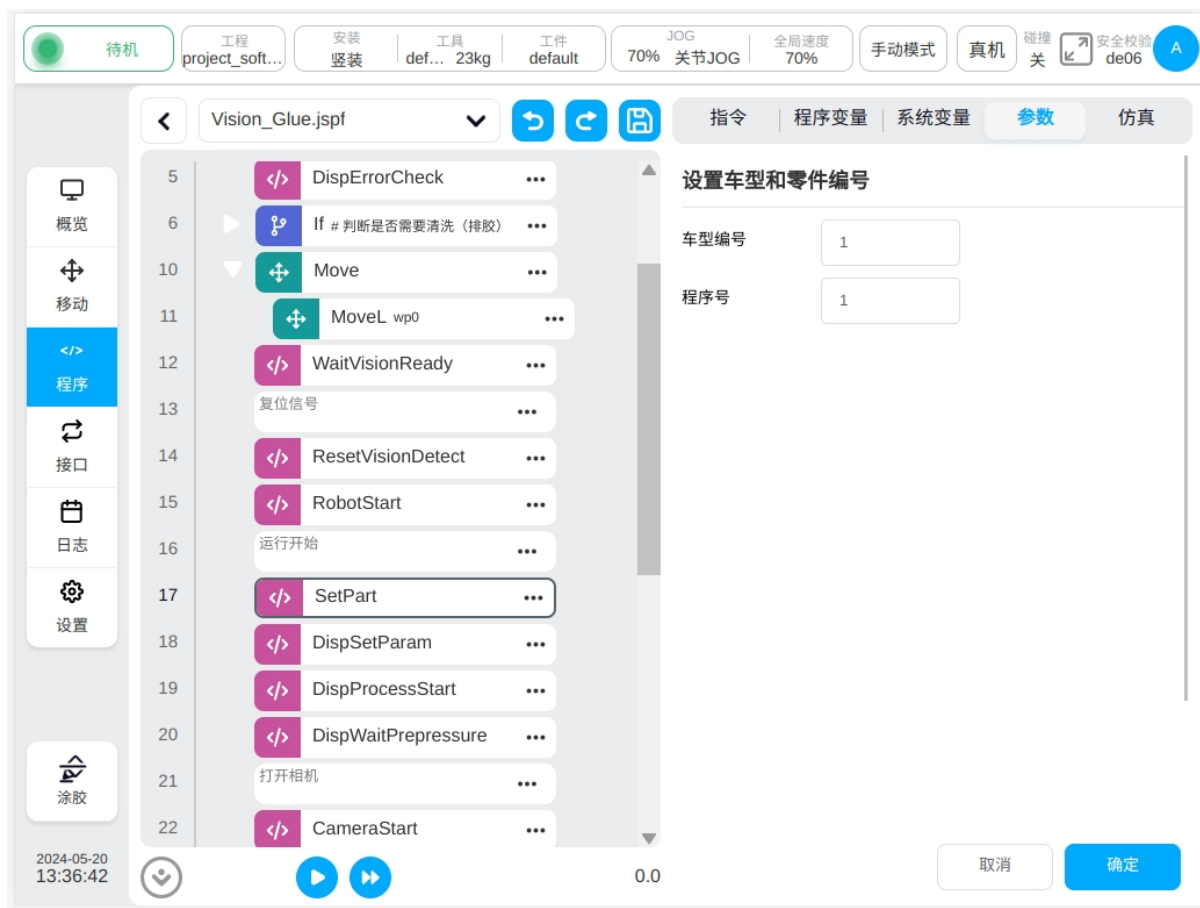


一直等待直到视觉检测设备准备就绪。点击确定，保存参数页面；点击取消，回到指令页面。

3.1.15 Setpart

Setpart 主要用来设置车型和零件编号。

配置页面可参考下图。



有如下设置项：车型编号和程序号。

机器人通过车型编号和程序号去选择胶机定义的参数。点击确定，保存参数页面；点击取消，回到指令页面。

3.2 编程脚本

涂胶工艺包会增加如下脚本，可以在使用脚本编程时调用

```
_plugin_dispense.init(progNo)
```

涂胶初始化

```
_plugin_dispense.check_purge_request()
```

检查是否需要清洗

```
_plugin_dispense.purge()
```

清洗

```
_plugin_dispense.set_prog_no(progNo)
```

设定程序号

```
_plugin_dispense.set_param_no(paramNo)
```

设定参数号

```
_plugin_dispense.check_prepressure()
```

检查是否达到预压

```
_plugin_dispense.wait_prepressure()
```

等待预压

```
_plugin_dispense.start_process()
```

工序开始

```
_plugin_dispense.end_process()
```

工序结束

```
_plugin_dispense.fill()
```

填充

```
_plugin_dispense.check_error()
```

检查故障

```
_plugin_dispense.reset_error()
```

故障复位

```
_plugin_dispense.check_reset_error()
```

检查并复位错误

```
_plugin_dispense.robot_start()
```

开始涂胶

```
_plugin_dispense.robot_stop()
```

结束涂胶

```
_plugin_dispense.camera_start()
```

触发拍照

```
_plugin_dispense.reset_vision()
```

复位

```
_plugin_dispense.set_part_code(code)
```

设置零件编号

```
_plugin_dispense.set_vision_sensor_code(code)
```

设置传感器编号

```
_plugin_dispense.set_car_type(code)
```

设置车型编号

```
_plugin_dispense.get_vision_check_ok()
```

视觉检测系统结果

```
_plugin_dispense.get_vison_ready()
```

视觉检测系统准备好

```
_plugin_dispense.calc_pose_from_raw(pose, laser)
```

根据激光及机器人当前位置计算寻位点坐标系

```
_plugin_dispense.get_laser_search_offset (type, pose_list)
```

计算三点四点寻位偏差

```
_plugin_dispense.three_offset (pose_list)
```

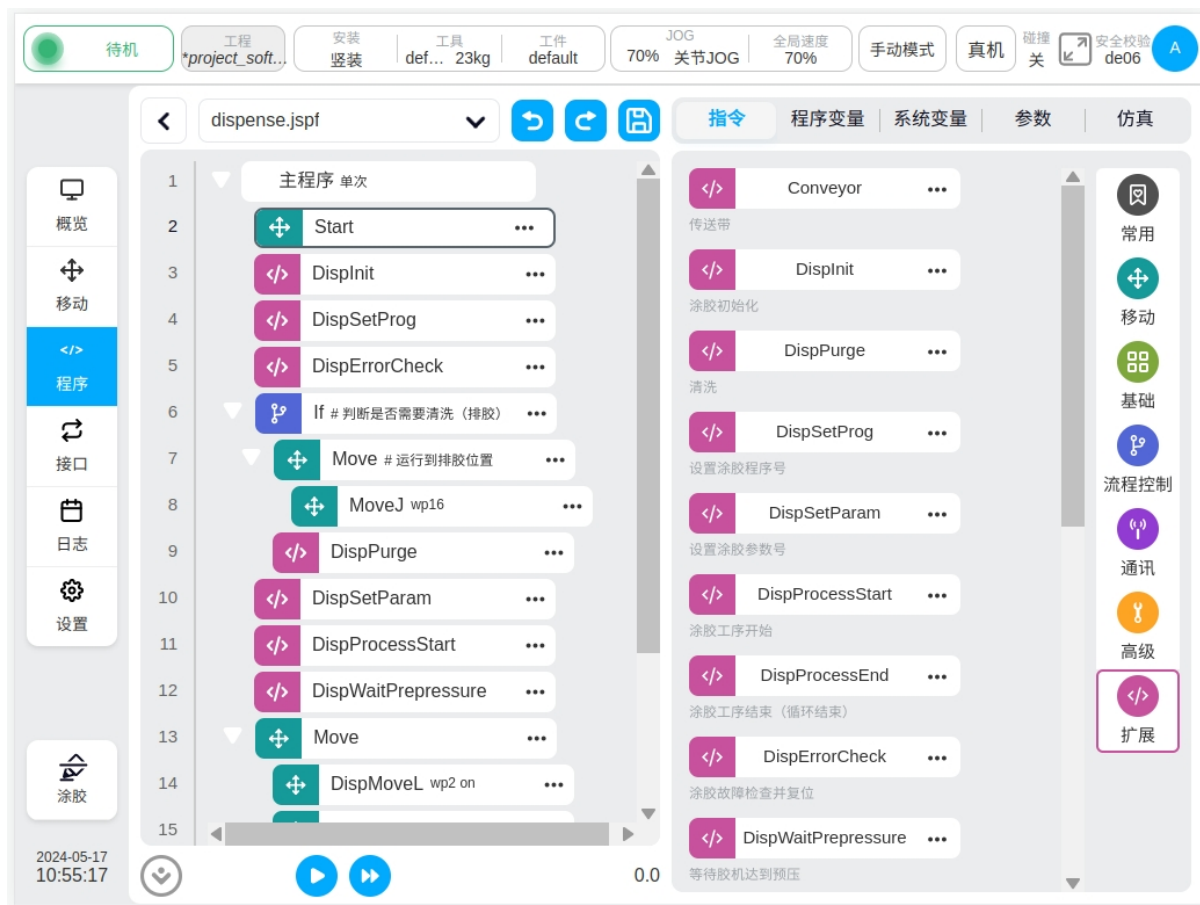
三点求旋转偏移坐标系

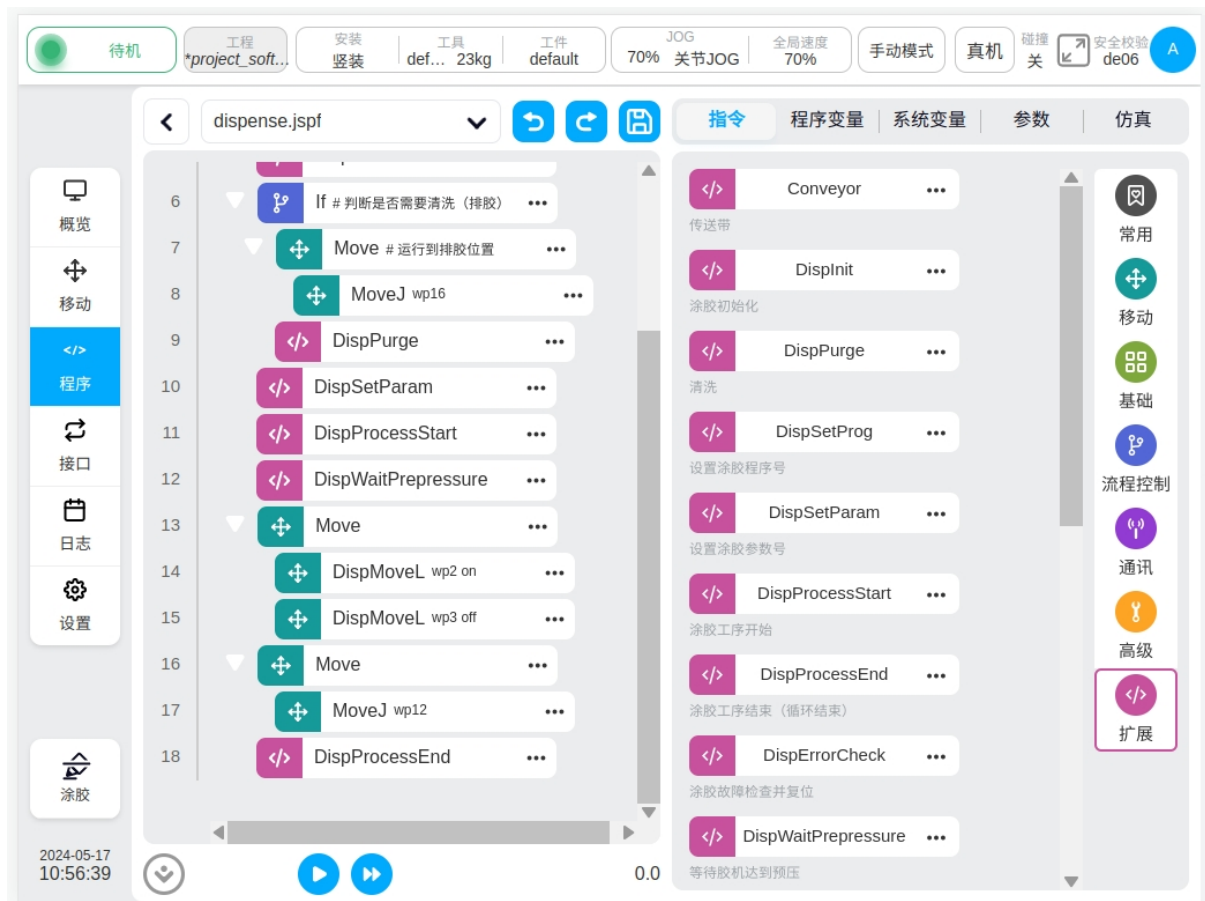
```
_plugin_dispense.four_offset (pose_list)
```

四点求旋转平移坐标系

3.3 程序示例

如下程序示例包含 Dispinit、DispPurge、DispSetProg、DispSetPram、DispProcessStart、DispProcessEnd、DispErrorCheck、DispWaitPrepressure 这八个模块，通过程序是否跑通验证模块的正确性。





如下程序示例包含涂胶检测 ResetVisionDetect、RobotStart、RobotStop、CameraStart、CameraStop、WaitVisionReady、Setpart 这七个模块，通过程序是否跑通验证模块的正确性。

2024-05-20 13:43:56

待机 工程 project_soft... 安装 竖装 工具 def... 23kg 工件 default JOG 70% 关节JOG 全局速度 70% 手动模式 真机 碰撞 关 安全校验 de06 A

Vision_Glue.jspf 指令 程序变量 系统变量 参数 仿真

1 主程序 单次

2 Start ...

3 DispInit ...

4 DispSetProg ...

5 DispErrorCheck ...

6 If # 判断是否需要清洗 (排胶) ...

10 Move ...

11 MoveL wp0 ...

12 WaitVisionReady ...

13 复位信号 ...

14 ResetVisionDetect ...

15 RobotStart ...

16 运行开始 ...

17 SetPart ...

18 DispSetParam ...

LaserTrack ... 激光跟踪

ResetLaserTrack ... 清空激光跟踪数据

ResetVisionDetect ... 视觉检测信号复位

WaitVisionReady ... 等待视觉检测设备准备就绪

RobotStart ... Robot Start

RobotStop ... Robot Stop

CameraStart ... Camera Start

CameraStop ... Camera Stop

SetPart ... 为涂胶检测设备设置车型, 设置零件编号

常用 移动 基础 流程控制 通讯 高级 扩展

2024-05-20 13:46:30

待机 工程 project_soft... 安装 竖装 工具 def... 23kg 工件 default JOG 70% 关节JOG 全局速度 70% 手动模式 真机 碰撞 关 安全校验 de06 A

Vision_Glue.jspf 指令 程序变量 系统变量 参数 仿真

17 SetPart ...

18 DispSetParam ...

19 DispProcessStart ...

20 DispWaitPrepressure ...

21 打开相机 ...

22 CameraStart ...

23 Move ...

24 DispMoveL wp2 on ...

25 DispMoveL wp3 off ...

26 关闭相机 ...

27 CameraStop ...

28 Move ...

29 MoveL wp4 ...

30 DispProcessEnd ...

31 RobotStop ...

LaserTrack ... 激光跟踪

ResetLaserTrack ... 清空激光跟踪数据

ResetVisionDetect ... 视觉检测信号复位

WaitVisionReady ... 等待视觉检测设备准备就绪

RobotStart ... Robot Start

RobotStop ... Robot Stop

CameraStart ... Camera Start

CameraStop ... Camera Stop

SetPart ... 为涂胶检测设备设置车型, 设置零件编号

常用 移动 基础 流程控制 通讯 高级 扩展

The screenshot displays the Vision_Glue.jspf program within the DUCO software interface. The top status bar indicates the system is in '待机' (Standby) mode. The main workspace is divided into two panels: a left panel showing a sequence of steps (20-33) and a right panel showing a list of available functions categorized by '常用' (Common), '移动' (Move), '基础' (Basic), '流程控制' (Flow Control), '通讯' (Communication), '高级' (Advanced), and '扩展' (Extension). The '扩展' category is currently selected, highlighting the '扩展' (Extension) function. The left panel shows a sequence of steps including 'DispWaitPrepressure', 'CameraStart', 'Move', 'DispMoveL wp2 on', 'DispMoveL wp3 off', 'CameraStop', 'Move', 'MoveL wp4', 'DispProcessEnd', 'RobotStop', 'Move', and 'MoveJ wp12'. The right panel lists functions such as 'LaserTrack', 'ResetLaserTrack', 'ResetVisionDetect', 'WaitVisionReady', 'RobotStart', 'RobotStop', 'CameraStart', 'CameraStop', and 'SetPart'. The bottom status bar shows the date and time as 2024-05-20 13:48:23 and a progress indicator at 0.0.

CHAPTER 4

激光跟踪器

4.1 介绍

激光跟踪器通过采集激光图像，获取缝的特征，识别缝。机器人获取其识别到的特征信息，实现缝的自动监测和校正。实际应用中，可以解决工件摆放误差、工件一致性差、涂胶热量导致的缝变形等问题。

涂胶工艺包支持激光寻位和激光跟踪功能。

4.2 激光通讯连接

使用网线连接机器人控制柜上的网口和激光传感器，将机器人的 IP 地址与激光传感器的 IP 配置到同一个网段。如图输入激光传感器的 IP 和端口号，点击连接。连接成功后会显示连接状态。各厂家连接设置如下

北京创想：

需要先使用创想的上位机将激光的协议设置为 Modbus。连接端口号 502

唐山英莱：

端口号设置为 5020。

苏州全视：

端口号设置为 502

激光传感器控制

可以手动操作激光传感器，打开/关闭激光，获取激光检测到的数据。



4.3 激光标定

激光传感器的标定用来计算机器人 TCP 和传感器的位置关系。即根据标定计算的结果，可以将激光识别到的特征点转换到其在机器人坐标系下的位姿。

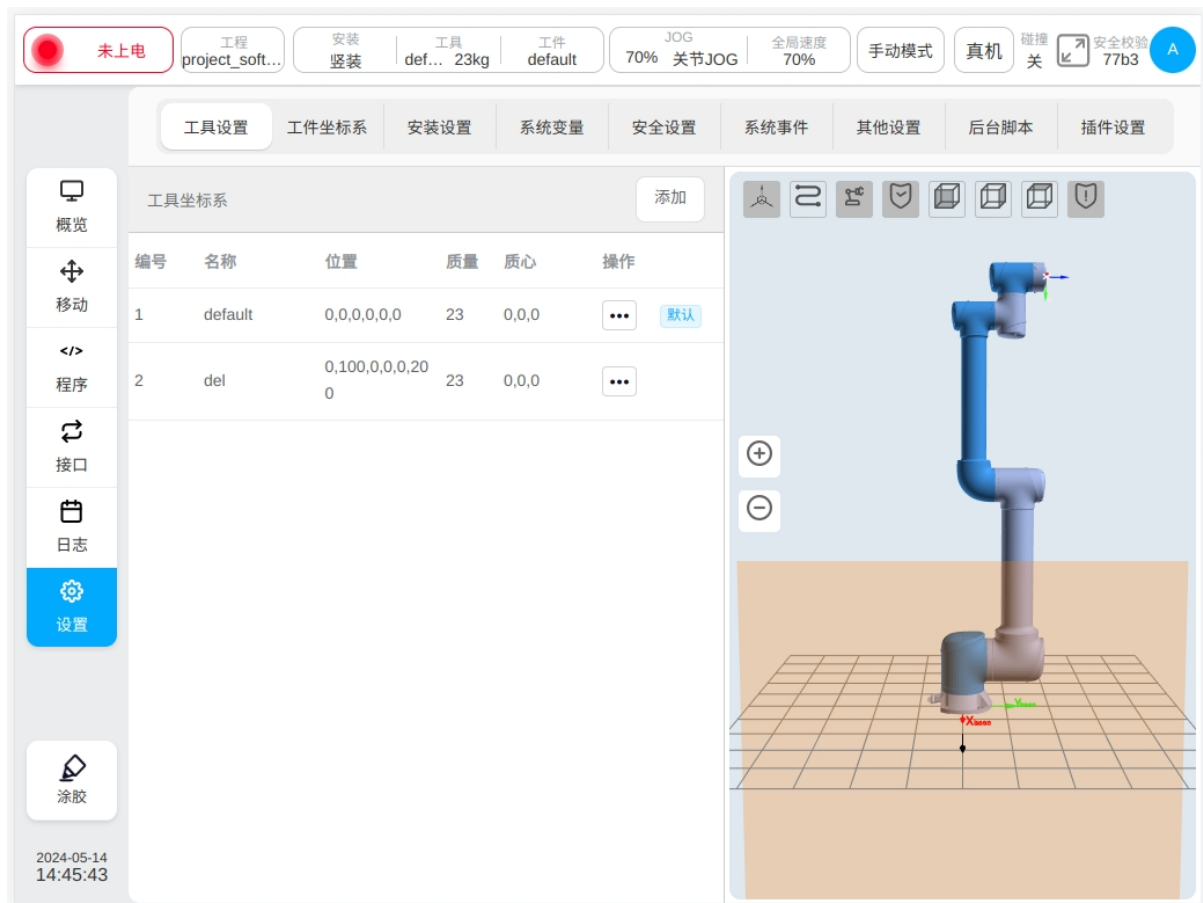
标定准备：

- 1、安装好胶枪，调整好胶枪长度为涂胶作业过程中使用的长度
- 2、根据激光传感器的安装要求和现场情况等安装好激光传感器
- 3、准备标定机器人 TCP 所使用的尖点
- 4、准备交换机等网络设备连接激光传感器、激光传感器所用上位机、机器人，设置其同一网段，确保通讯正常
- 5、确定在激光传感器的上位机上能看到激光图像
- 6、机器人涂胶工艺包传感器设置页面中验证激光开关、获取数据等能正常工作

标定过程：

- 1、工具（TCP）标定，即标定涂胶末端位置，此时涂胶的长度即为实际涂胶时的长度。标定方法如下：

在机器人的设置页面——工具设置中新建一个 TCP，按照 4 点标定法或者 6 点标定法



工具坐标系

×

名称

weld

TCP标定

X(mm)

0

RX(°)

0

Y(mm)

0

RY(°)

0

Z(mm)

0

RZ(°)

0

负载

负载标识

质量 (kg)

0

负载惯性张量设置

X(mm)

0

X

0

Y

0

Z

0

Y(mm)

0

Y

0

0

0

Z(mm)

0

Z

0

0

0

取消

确定



移动机器人使涂胶末端对准尖点，分别记录四个点位



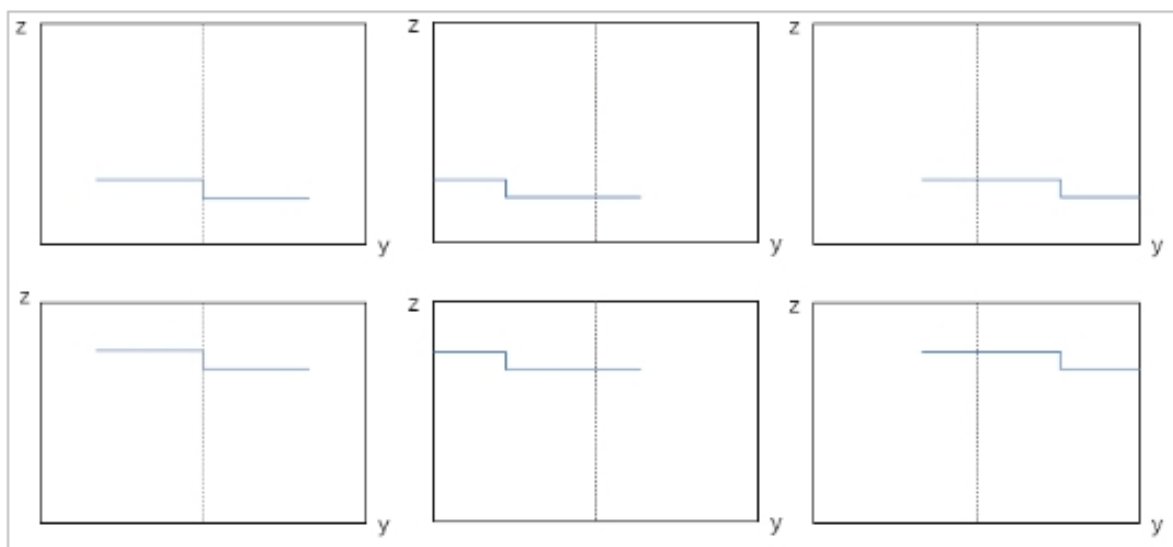
点击确定，将计算标定出的 TCP 坐标以及误差。控制平均误差和最大误差在 0.5mm 以内。

将标定好的 TCP 设置为当前 TCP

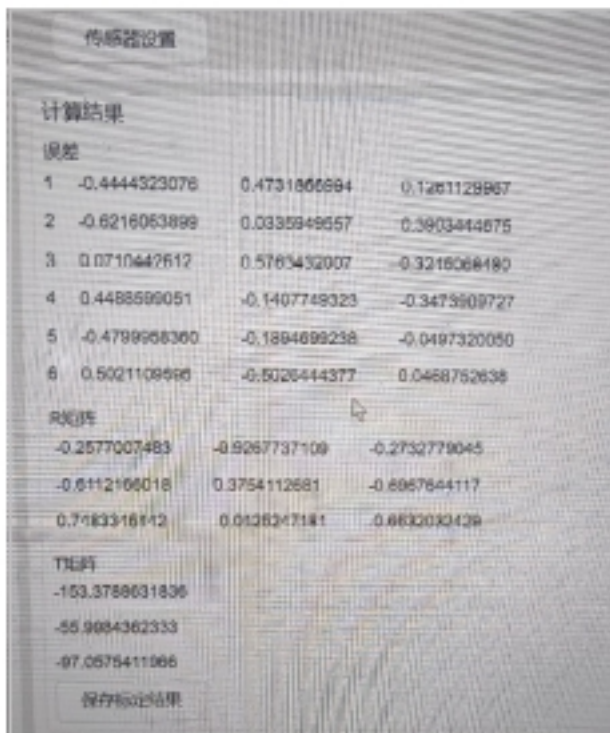
2、在涂胶上取一个标记点，移动机器人，使其 TCP 对准标记点，点击记录



3、保持机器人姿态不变，打开激光，开启使能，移动机器人，让激光落在标记点上，在标记过程中观察激光软件上的激光图像，依次使激光图像分别为下图的六种状态，记录点位 1-6



4、点击标定计算，下方会显示标定误差和计算结果。若每个值小于 1mm，标定有效，否则重新标定。



5、点击保存标定结果，确定后会将计算结果保存。注意：该计算结果存放在机器人当前工程中，建议此时保存下工程，以防丢失。

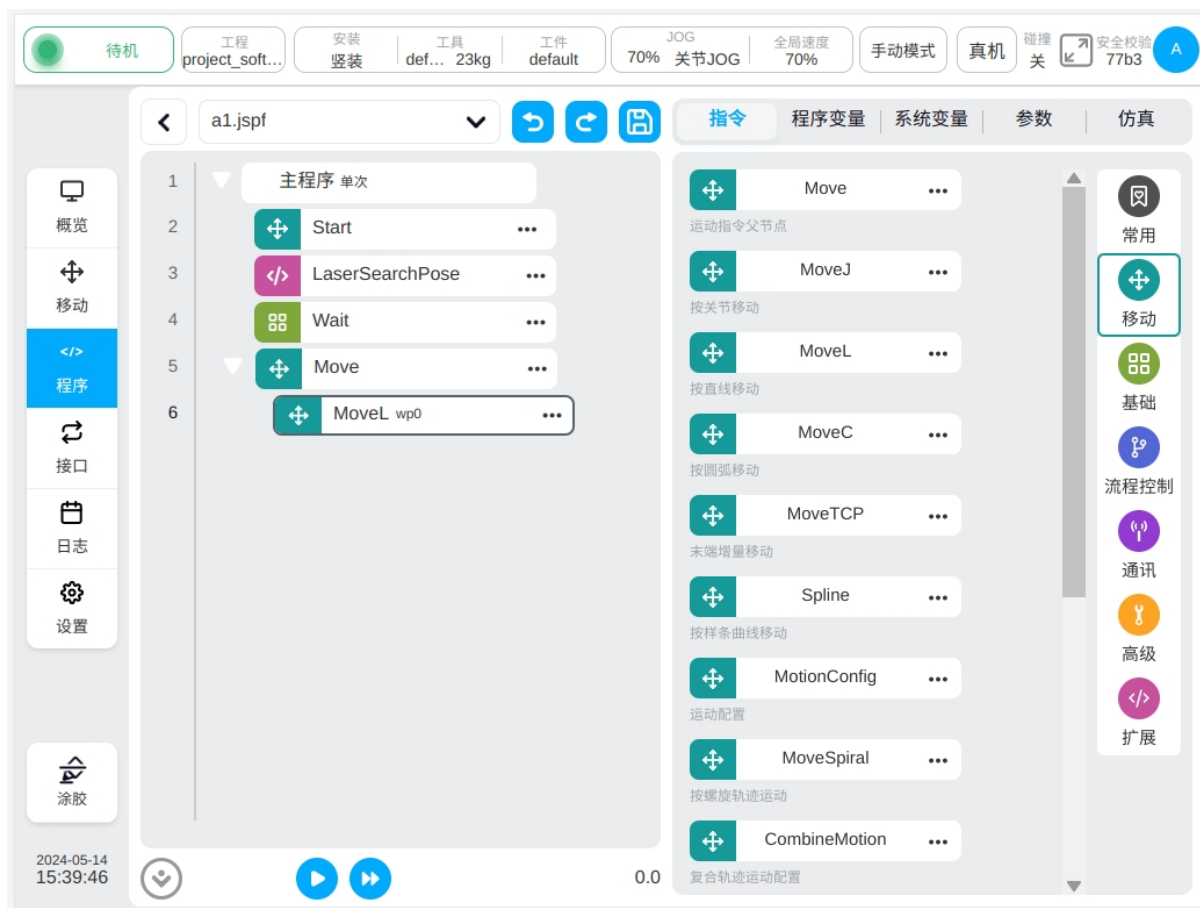
注意：

- 1、标定前需要标定机器人 TCP，保证标定精度
- 2、标定前，将标定好的工具坐标系设为当前
- 3、记录 7 个点位时，涂胶的长度与标定 TCP 时一致
- 4、记录 7 个点位时，机器人的姿态不能改变
- 5、记录这 7 个点位时，保证激光图像检测稳定

标定精度验证

标定好传感器和机器人后，可以使用单点寻位验证标定的精度。编写如下程序，使用功能块 LaserSearchPose 和 MoveL。LaserSearchPose 将打开并使能激光，获取激光识别到的特征点位信息，转换为其在机器人基坐标系下的值，将值赋值给变量。MoveL 移动到该变量处。运行程序，检查机器人 TCP 是否移动到识别到的特征点处。

注意：请保证此时的 TCP 为上述标定使用的 TCP

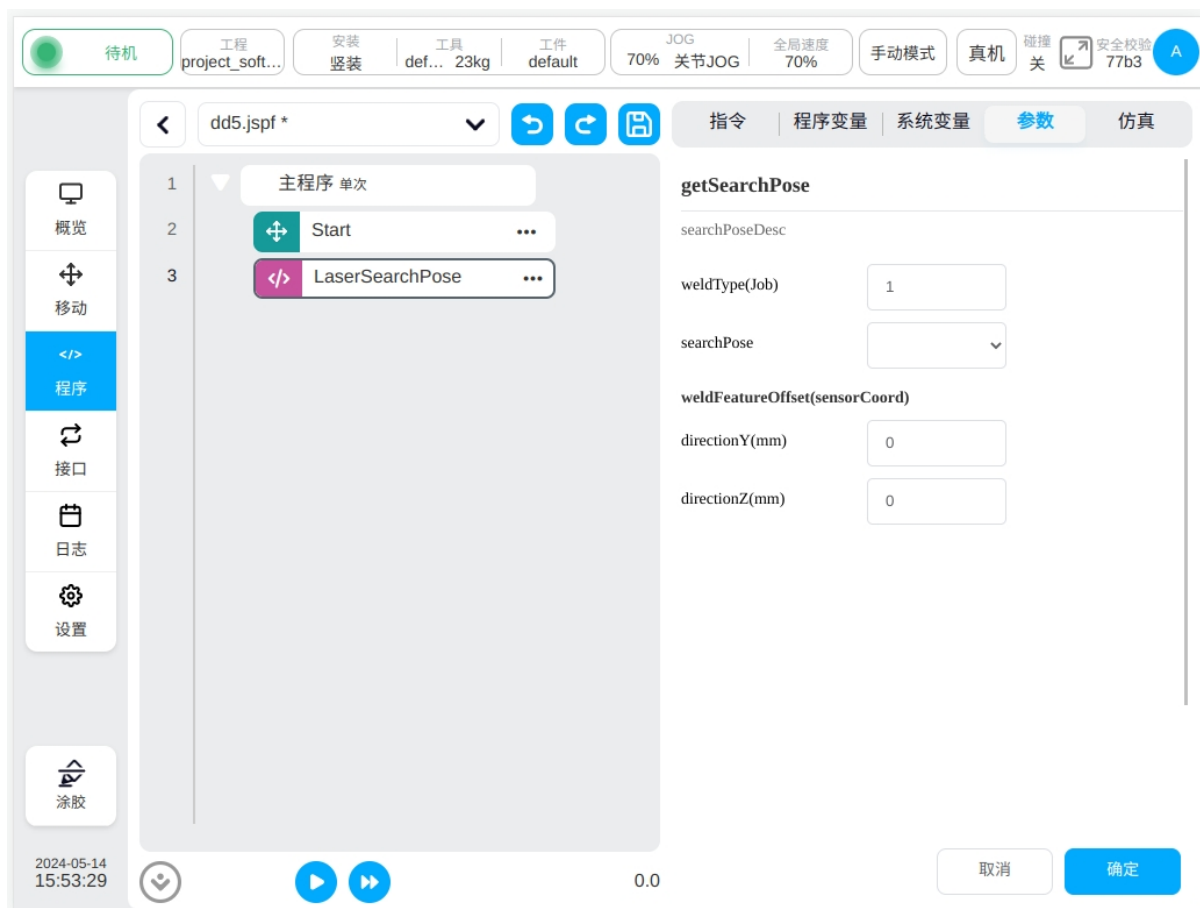


4.4 编程功能块

与激光传感器相关的功能块有三个：LaserSearchPose, LaserSearchOffset, LaserTrack。

LaserSearchPose

主要用来获取激光识别到的特征点在机器人基坐标系下的位姿。该功能块封装了以下操作：选择使用的 Job 号，打开并使能激光，获取激光识别到的特征点位信息，获取机器人当前 TCP 在基坐标系下的值，根据标定关系，计算特征点在机器人基坐标系下的位姿信息，并赋值给 pose 类型变量。若未检测到缝，返回 nil 配置页面如下



缝模式参数：即 Job 号，是激光跟踪器端设置并保存的一组参数：如缝类型、曝光、检测能力等。

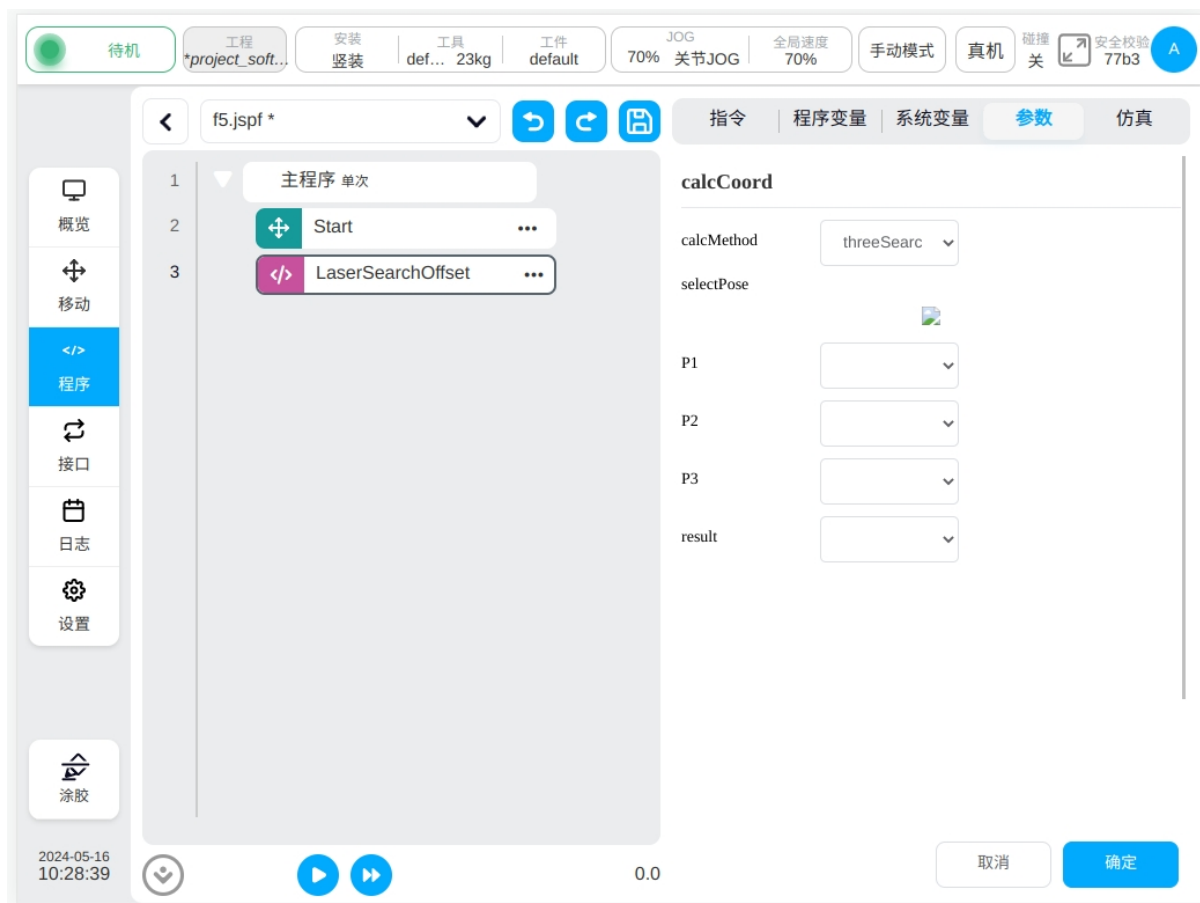
寻位点：计算的结果，可以关联一个 pose 类型的变量

缝特征点偏移：将识别的特征点上加上偏移量，计算偏移后点位转换后的位姿。可以用来处理机器人误差、检测点与涂胶工艺不匹配等问题（若激光跟踪器提供了功能，也可以在激光跟踪器端设置）

LaserSearchOffset

主要用来计算三点寻位或四点寻位得到的坐标系的位姿

配置页面如下



有如下设置项

计算方法：选择三点寻位或者四点寻位

选择点位：关联点位信息

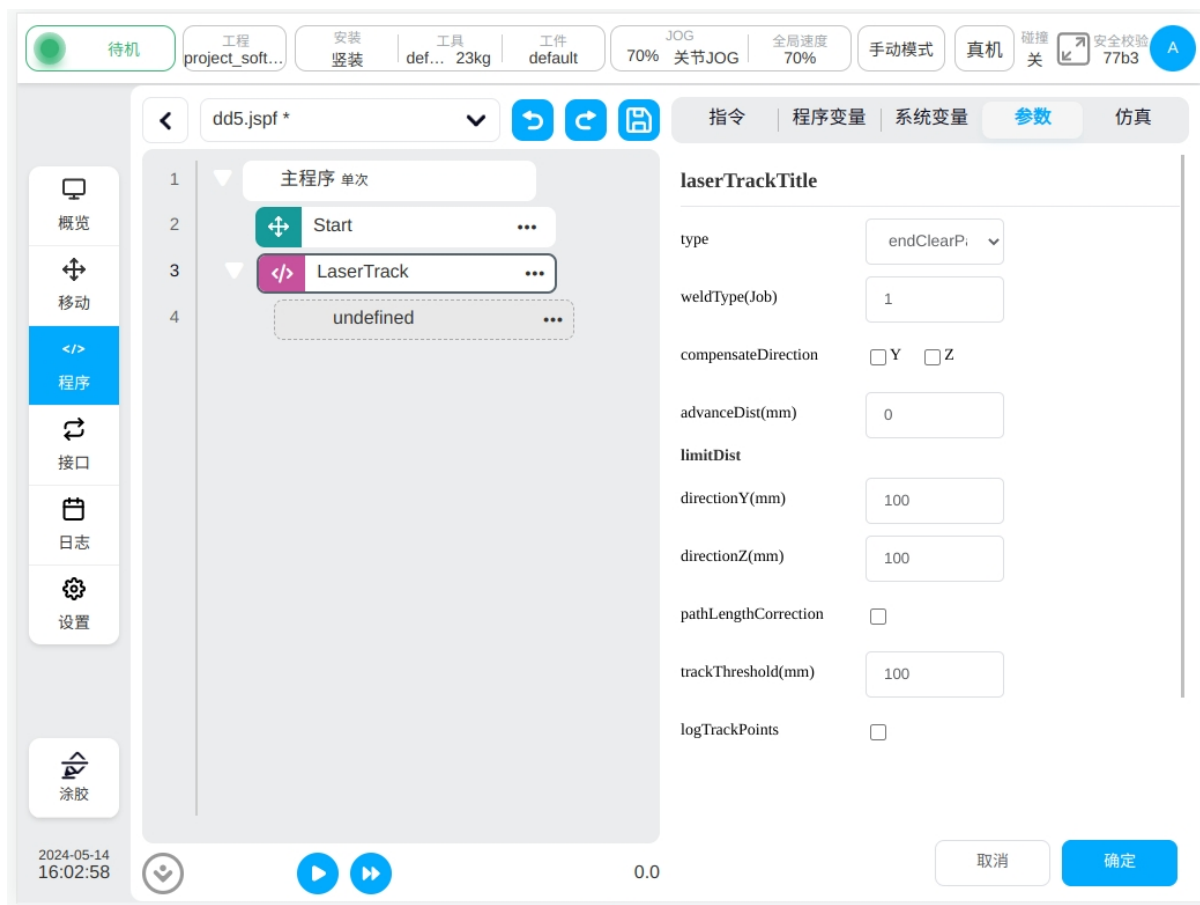
结果：计算得到的结果，关联到 pose 类型变量

LaserTrack

实现激光跟踪功能。

主要实现功能：打开并使能激光，设置激光使用的 Job 号，设置补偿方向和提前距离，实时获取缝特征点，计算偏差并补偿到实际轨迹上。

配置页面如下



有如下设置项

缝模式参数：Job 号

补偿方向：选择 Y 向或者 Z 向是否开启补偿

提前距离：设置激光线与涂胶的距离，单位 mm。

最大补偿距离限制：示教轨迹和激光测得的轨迹的允许最大偏差

缝特征点偏移：将识别的特征点上加上偏移量（若激光跟踪器提供了功能，也可以在激光跟踪器端设置）

4.5 编程脚本

涂胶工艺包会增加如下脚本，可以在使用脚本编程时调用

```
_plugin_dispense.open_laser()
```

打开激光

```
_plugin_dispense.close_laser()
```

关闭激光

```
_plugin_dispense.enable_laser()
```

使能激光（创想激光传感器需要调用）

```
_plugin_dispense.disable_laser()
```

下使能激光（创想激光传感器需要调用）

```
_plugin_dispense.set_laser_dispense_type(type)
```

设置缝类型参数（Job 号）

```
_plugin_dispense.search_pose(pose, offset)
```

寻位，功能同 LaserSearchPose 功能块，打开激光，获取缝特征点，计算其在机器人坐标系下的位姿，关闭激光。

pose: pose 类型，机器人 TCP 位姿，单位 m, rad

offset: number_list 类型，缝特征点偏移量 {y, z}，默认 {0, 0}，单位 mm

返回值

pose 类型，特征点在机器人坐标系下的位姿，单位 m, rad，为 nil 时表示未检测到缝

```
_plugin_dispense.get_laser_search_pose(pose, offset)
```

与 search_pose 类似，不开关激光，获取缝特征点，计算其在机器人坐标系下的位姿

pose: pose 类型，机器人 TCP 位姿，单位 m, rad

offset: number_list 类型，缝特征点偏移量 {y, z}，默认 {0, 0}，单位 mm

返回值 pose 类型，特征点在机器人坐标系下的位姿，单位 m, rad，为 nil 时表示未检测到缝

```
_plugin_dispense.get_laser_raw_data()
```

获取激光原始数据

返回如下数组 {status, y, z, width, depth}，分别表示

status: boolean 表示是否检测到缝

y: number, 缝特征点 Y 向坐标，单位 mm

z: number, 缝特征点 Z 向坐标，单位 mm

width: number, 检测到的缝的宽度，单位 mm

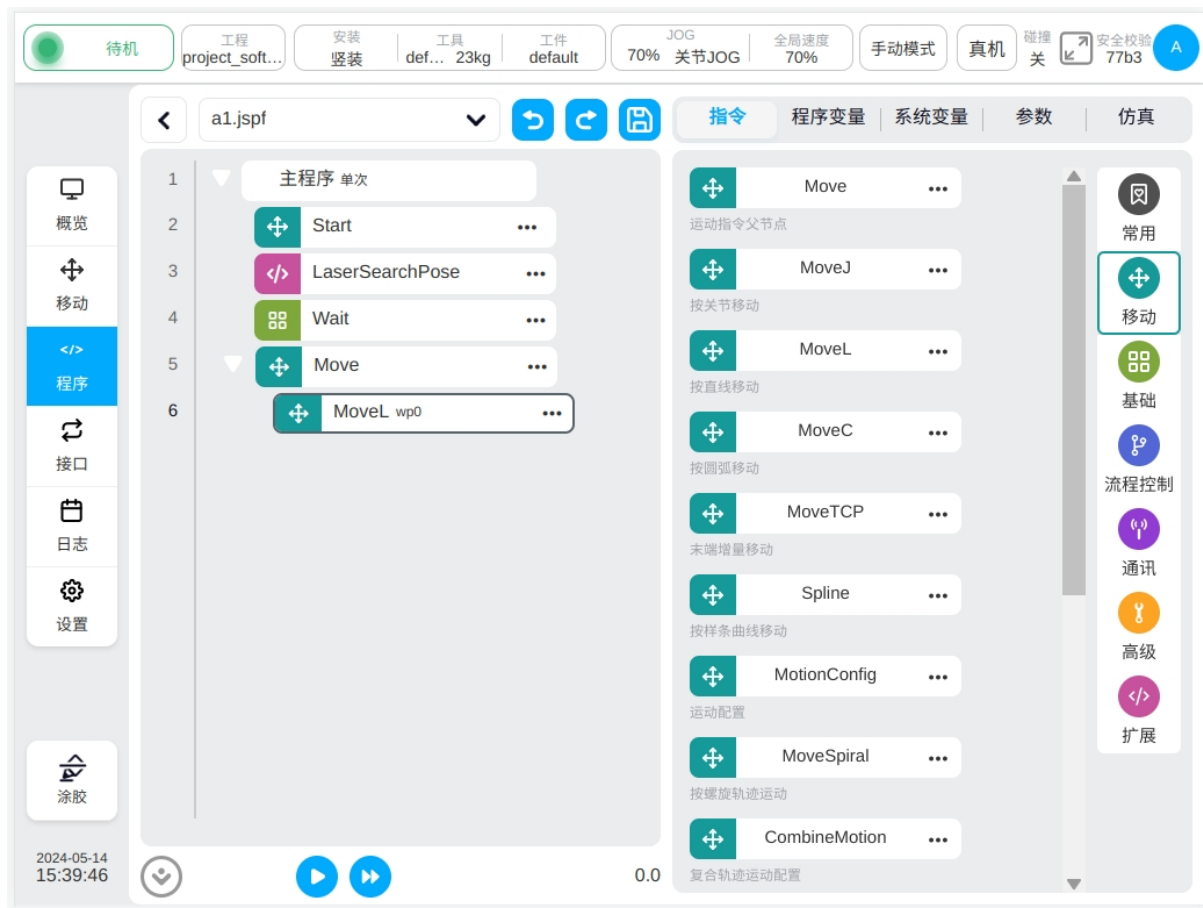
depth: number, 检测到的缝的深度，单位 mm

4.6 激光寻位

单点寻位

适合应用在涂胶等场景

编程示例如下：

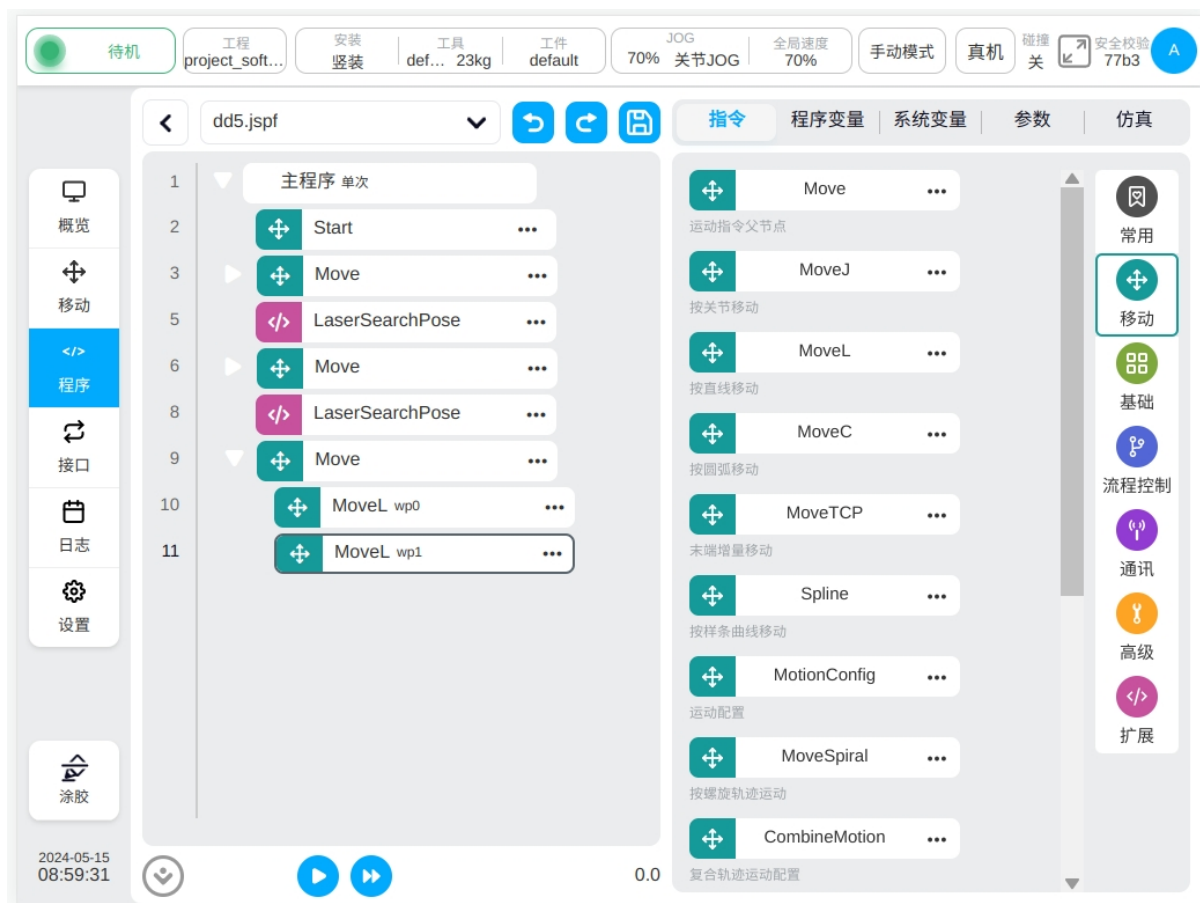


两点寻位

适合应用在间断涂胶等场景

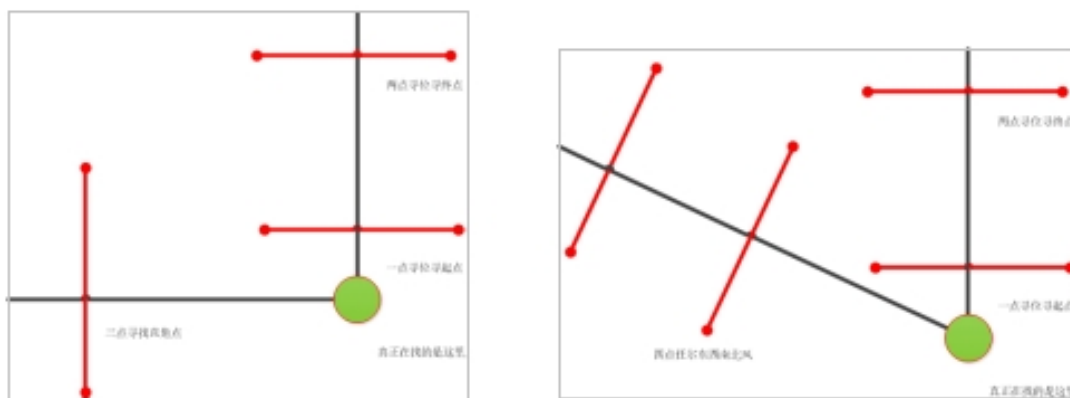


编程示例如下

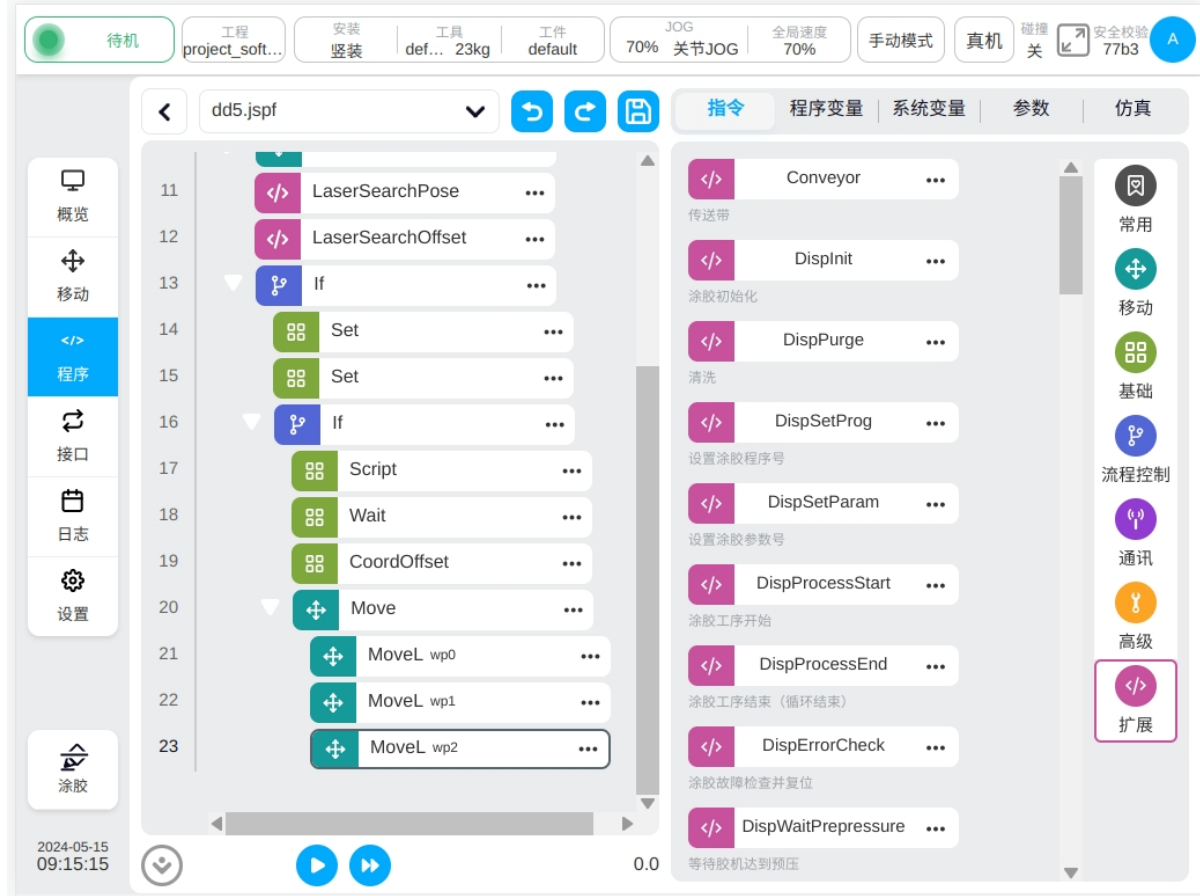
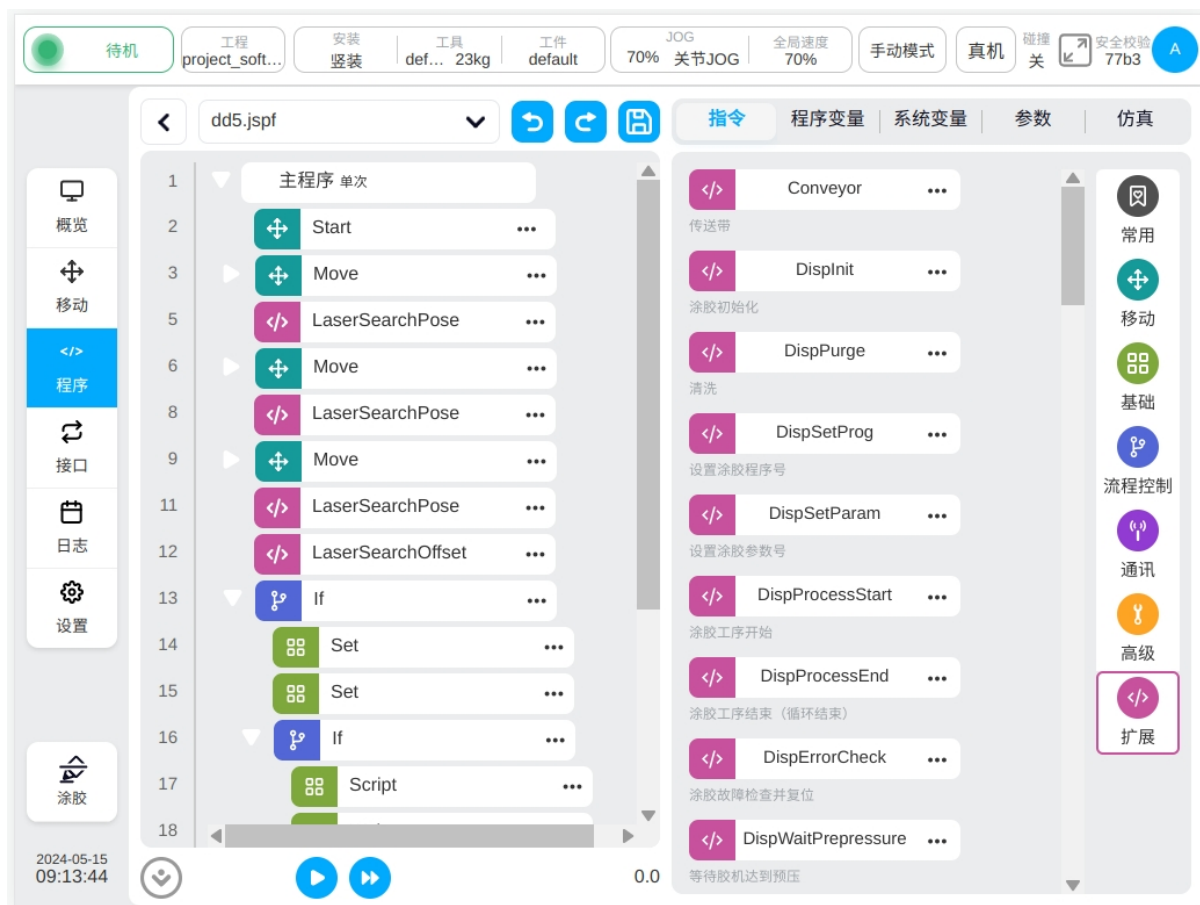


三点寻位/四点寻位

基本原理为：根据激光寻位的点位建立了一个工件坐标系，计算其与原有的工件坐标系的偏差，从而得到工件上缝的偏移量。可以用于复杂结构件的涂胶。四点寻位相比三点寻位可以更好的处理工件的来料误差、组装误差等。



下面为三点寻位的 Demo，四点寻位与三点寻位类似。



变量定义

变量名	类型	
P1	pose	获取到的第一个寻位点
P2	pose	获取到的第二个寻位点
P3	pose	获取到的第三个寻位点
teach_pose	pose	三点寻位计算的结果
g_enable_o	boole	控制程序的运行。为 false 时，用来做第一次寻位计算初始的工件坐标系；为 true 时，执行正常作业流程
g_base_pos	pose	计算得到的初始工件坐标系
g_pose	pose	三次寻位计算的结果
g_offset	pose	与初始坐标系的偏差

程序说明

3 行第一个寻位点

5 行寻位，获取寻位结果，保存在变量 p1

6 行第二个寻位点

8 行寻位，获取寻位结果，保存在变量 p2

9 行第三个寻位点

11 行寻位，获取寻位结果，保存在变量 p3

12 行三点寻位，计算 p1,p2,p3 确定的坐标系，保存在变量 teach_pose

13-16 行若变量 g_enable_offset 为 false 时，将 teach_pose 的值赋值给 g_base_pose；否则，赋值给变量 g_pose

17 行当 g_enable_offset 为 true 时执行后续过程

18 行脚本使用 pose_offset() 计算 g_base_pose 和 g_pose 的坐标系偏差，并将结果赋值给变量 g_offset

20 行将 g_offset 作为坐标系偏移量添加到当前坐标系上

21-24 行实际作业点，使用工件坐标系。

实际使用过程如下：

1、示教机器人，确定激光的寻位点

2、摆好工件，将 g_enable_offset 值设为 false，运行程序

3、在变量区查看 g_base_pose 的值，新建一个工件坐标系 wobj1，工件坐标系使用 g_base_pose 的值

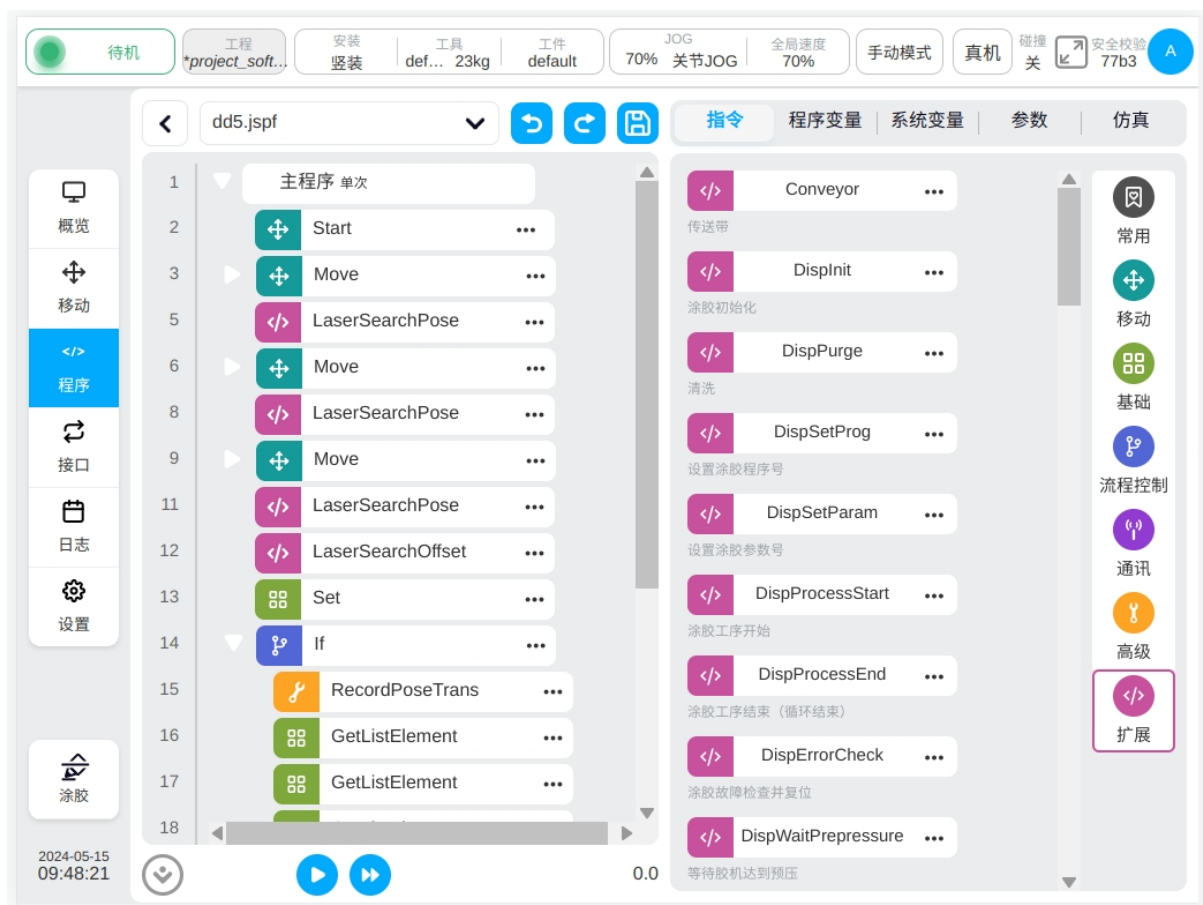
4、设置 wobj1 为当前工件坐标系，以该坐标系作为基准示教工件的作业轨迹，即 22-24 行的点位

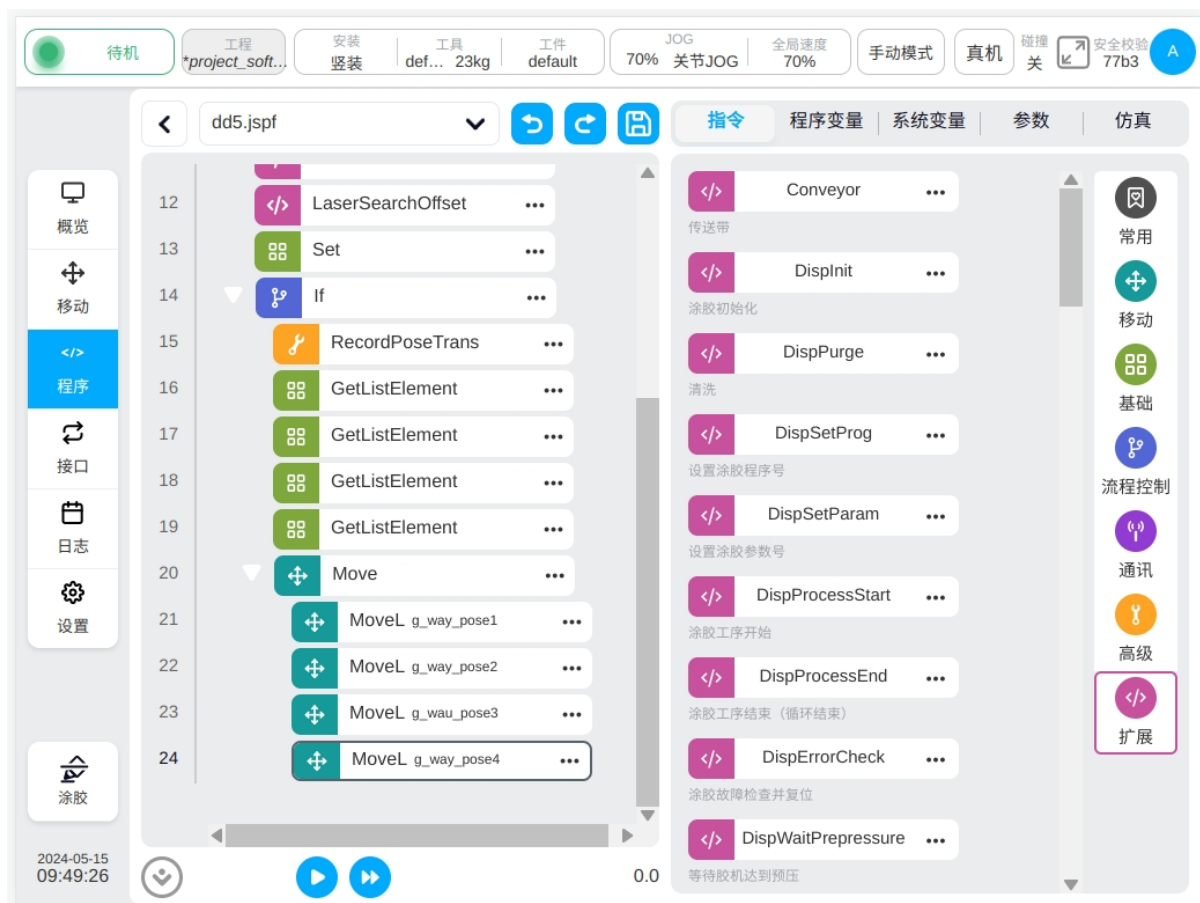
5、将 g_enable_offset 值设为 true，运行程序，执行正常的作业流程。

上述 Demo 使用了工件坐标系，通过第一次执行三点寻位计算得到了一个初始的工件坐标系，基于该坐标系示教作业点位。保证若工件由于摆放等发生了一定的偏移，可以计算三点

寻位得到的坐标系与初始坐标系的偏差，将该偏差添加到工件坐标系上，使作业轨迹能适应变化。

由于该 Demo 需要设置工件坐标系，某些情况下示教需要频繁切换坐标系，可能比较繁琐。可以参考如下 Demo，使用了 RecordPoseTrans 功能块，可以都基于基坐标系编程。





变量定义

变量名	类型	
P1	pose	获取到的第一个寻位点
P2	pose	获取到的第二个寻位点
P3	pose	获取到的第三个寻位点
teach_pose	pose	三点寻位计算的结果
g_enable_o	boolean	控制程序的运行。为 false 时，用来做第一次寻位计算初始的工件坐标系；为 true 时，执行正常作业流程
g_base_pos	pose	计算得到的初始工件坐标系
g_result_1	pose_1	实际作业点在基坐标系下的位姿
g_way_pose	pose	第一个作业点
g_way_pose	pose	第二个作业点
g_way_pose	pose	第三个作业点
g_way_pose	pose	第四个作业点

程序说明

- 3 行第一个寻位点
- 5 行寻位，获取寻位结果，保存在变量 p1
- 6 行第二个寻位点
- 8 行寻位，获取寻位结果，保存在变量 p2

- 9 行第三个寻位点
- 11 行寻位，获取寻位结果，保存在变量 p3
- 12 行三点寻位，计算 p1,p2,p3 确定的坐标系，保存在变量 teach_pose
- 13 行将 teach_pose 值赋值给 g_pose
- 14 行当 g_enable_offset 为 true 时执行后续过程
- 15 行操作点示教转换，并将结果保存在变量 g_result_list
- 16-19 行从 g_result_list 中获取各个点位并将其赋值给各个作业点变量
- 20-24 行执行实际作业，点位使用各个作业点变量

实际使用过程如下：

- 1、示教机器人，确定激光的寻位点
- 2、摆好工件，将 g_enable_offset 值设为 false，运行程序

3、在变量区查看 g_base_pose 的值，打开 RecordPoseTrans 功能块，如图，在输入参考位姿中选择变量 g_base_pose，点击“设置为基准”按钮，基准位姿处将显示此时的基准位姿值。

输入-参考位姿

请选择用户操作所定义的参考位姿 (机器人基坐标系下)

g_base_pose ▼ 设置为基准

基准位姿 [-856.6, 580.33, 94.39, 2.74, 0.43, 48.35]

操作点偏置

X	Y	Z	RX	RY	RZ	
-862.97	538.46	97.24	-144.09	8.1	66.14	🗑️
-856.5	593.66	98.23	-144.09	8.1	66.14	🗑️
-853.94	593.65	96.97	-144.09	8.1	44.14	🗑️
-819.42	632.59	97.76	-144.09	8.1	44.14	🗑️

示教添加

输出-转换位姿

g_result_list ▼

4、在操作点偏置中，示教添加作业点位，在输出中选择一个 pose_list 类型变量存放计算得到的作业点位结果

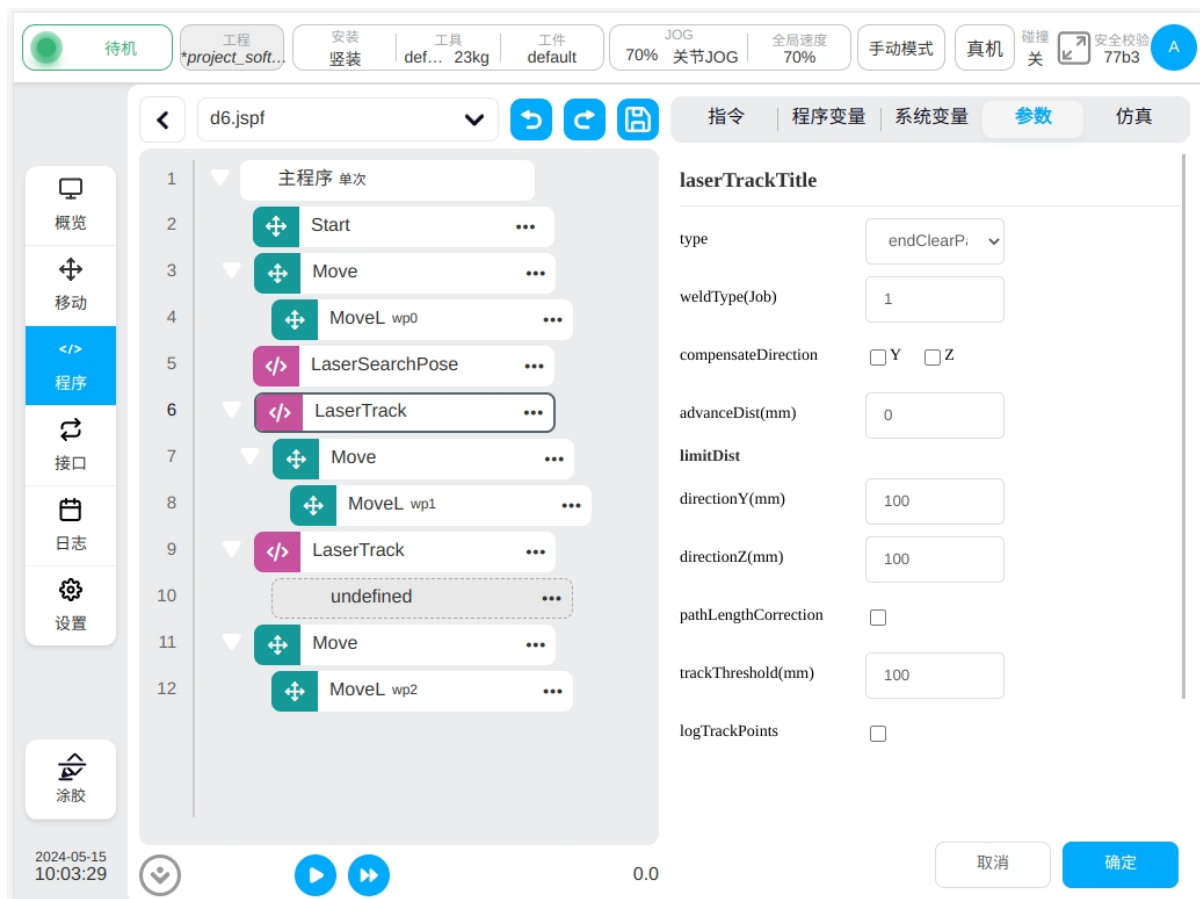
- 5、将 g_enable_offset 值设为 true，运行程序，执行正常的作业流程。

相比上一个 Demo，使用变量作为实际作业点位，通过 RecordPoseTrans 计算点位。以第一次使用得到的坐标系作为基准示教各个作业点位，添加到 RecordPoseTrans 块中。后续若工件发生了偏移，根据寻位计算的坐标系，通过 RecordPoseTrans 功能块计算出各个作业点位在基坐标系下的值。

4.7 激光跟踪

当需要在涂胶过程中实时检测缝的变化，此时就需要使用激光跟踪功能。

如下示例



程序说明

- 4 行机器人开启跟踪的点位 wp1
- 5 行起始点寻位，将结果放在变量 start 中
- 6 行开启跟踪，设置结束时不关闭跟踪
- 7 行移动到起点
- 9 行开启跟踪，设置结束时关闭跟踪
- 11 行示教一条原始的轨迹 wp0

机器人将执行如下操作，打开并使能激光，获取起点，打开跟踪并移动到起点，根据实时获取激光传感器得到的特征点，计算其与示教轨迹的偏差，将偏差量补偿到示教的轨迹上，使机器人 TCP 按照检测到的缝运行。

4.8 异常检测

缝有无检测

LaserSearchPose 可以判断其返回值是否为 nil 来判断是否检测到缝。示例如下：

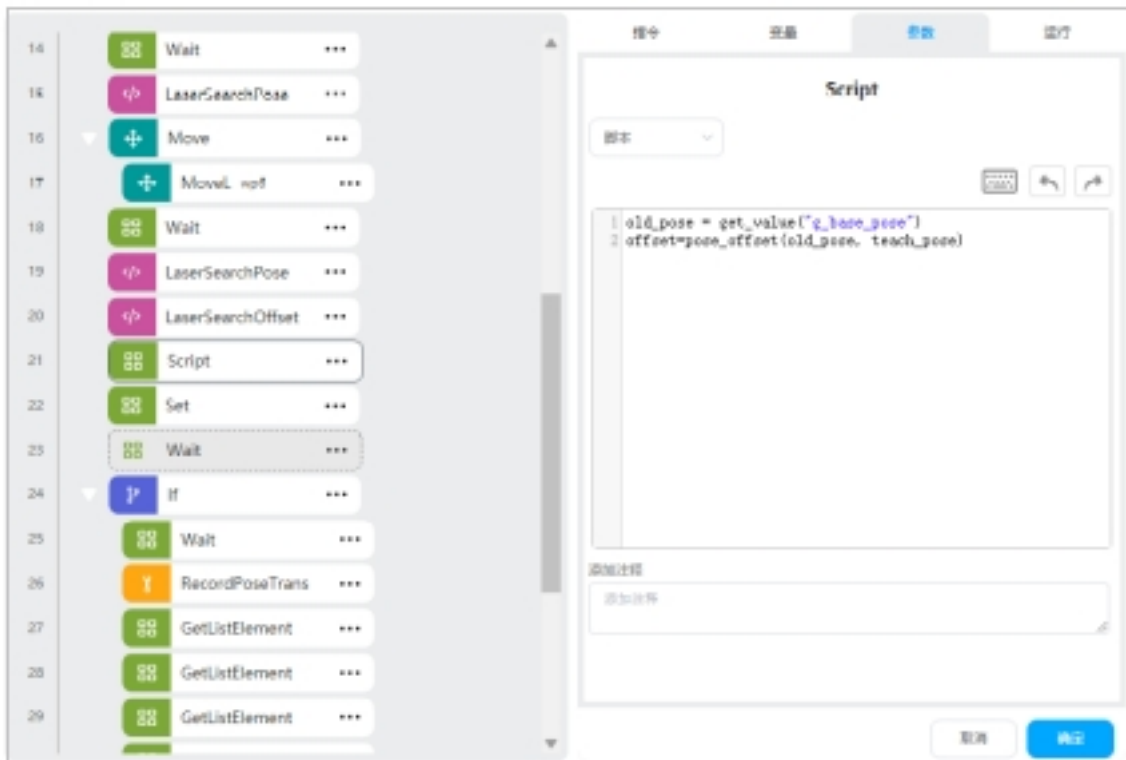


LaserSearchPose 将寻位结果放在变量 target 中，If 条件判断 target 是否为 nil 来执行检测到缝和未检测到缝时的逻辑

三/四点寻位偏差过大保护

某些情况下缝可能检测错误，导致计算出来的工件坐标系偏差过大，可以增加偏差限制，防止出现撞枪等安全问题。

示例如下：



Script 功能块使用 pose_offset 脚本计算原有的和新计算坐标系的偏差，增加判断偏差大于某个范围的执行逻辑。