



## ECR5 协作机器人

-----用户手册



使用前请仔细阅读本手册



埃夫特智能装备股份有限公司

EFORT INTELLIGENT EQUIPMENT CO.,LTD.

# ECR5 用户手册

(示教器操作)



版本号：V1.1

-----服务热线 (Tel) : 4000528877-----

# 声 明

感谢您购买埃夫特机器人产品，为确保已对产品进行正确的设置，请您在使用本产品之前，务必仔细阅读本操作手册。本声明及手册所提及的内容涉及您的人身及财产安全，若不遵循或不按照手册的说明与警告而擅自操作，可能会给您和周围的人带来人身伤害或给埃夫特机器人或周围的其他物品造成财产损失。本声明及手册为截至本批次产品出厂前的最新版本，后续请通过访问 [www.efort.com.cn](http://www.efort.com.cn) 官方网站以获取更新的信息。

本手册仅作为对产品进行正常操作的指导，在产品使用过程中，埃夫特公司并不对除产品缺陷外的其他原因引发的人身伤害、财产损失承担责任。埃夫特公司郑重建议：参与机器人操作、示教、维护、维修、点检等相关活动的人员，在学习完毕埃夫特公司准备的培训课程前，请勿赋予其对机器人的操作使用权限。

# 前言

## 关于本手册

本手册为 ECR 系列机器人示教器软件使用手册，详细介绍软件各功能界面的使用。

## 操作前提

操作机器人前，请仔细阅读产品的相关安全说明，必须在了解安全规范的基础上才可进行实机操作。

## 目标群体



- ◆ 生产操作人员
- ◆ 工程技术人员
- ◆ 技术服务人员

## 常见标识含义

本手册中使用警告标识，表示使用过程中的重要信息。警告标志及其含义，见下表 1：

表 1：本文中使用的标志

	<b>危险！</b> 如不按照说明进行操作，会发生事故，导致严重或致命的人员伤害或物品损坏。
	<b>警告！</b> 如不按照说明进行操作，会发生事故，导致严重或致命的人员伤害或物品损坏。
	<b>小心！</b> 如不采取相应的预防措施，可能发生危险的情况，导致财产损失或轻微的身体伤害。
	<b>危险！</b> 如不采取相应的预防措施，可能发生危险的用电情况，导致严重或致命的人员伤害。

	<p><b>警告!</b></p> <p>如不采取相应的预防措施，可能发生危险的用电情况，导致人员伤害或设备的严重损坏。</p>
	<p><b>警告!</b></p> <p>如果不采取相应的预防措施，可能发生危险的发热表面，导致人员伤害。</p>

## 风险评估

ECR5 协作机器人在使用时，原则上可以不设置外围栏，但必须评估操作及正常运行时的风险，风险评估必须在机器人投入使用前完成，集成商及用户务必全面考虑机器人在实际使用时其整个生命周期内的所有工作任务，来衡量风险等级及可接受程度，评估风险时，可从以下方面开始，但不限于：

- (1) 末端执行器/工件等尖锐边角对人员造成伤害；
- (2) 机器人工作空间内障碍物对人员与设备造成伤害；
- (3) 机器人运行时可能会与人员碰撞造成伤害；
- (4) 机器人在操作危险物品时可能存在风险；
- (5) 不同的安全参数设置在应用场景中可能存在风险；
- (6) 末端执行器的安装不可靠存在风险；
- (7) 机器人与刚性边界/障碍较近时存在风险；

机器人在非协作场合下集成/应用时，可关闭机器人内部安全模式，内部安全设置失效，集成商及用户须考虑额外的安全设备，并评估其安全性风险。

机器人在更改配置及更换末端执行器后，应重新进行风险评估，并在使用前完成评估。



# 目 录

第 1 章 示教器	1
1.1 概述	1
1.2 组成	1
1.2.1 模式切换旋钮	2
1.2.2 使能按键	3
1.2.3 控制键	4
1.2.4 功能键	4
1.2.5 急停按键	5
第 2 章 系统界面	7
2.1 主界面	7
2.1.1 导航栏	7
2.1.2 状态栏	8
2.2 登录页	9
2.2.1 用户权限	9
第 3 章 机器人基础操作	11
3.1 手动模式+上伺服	11
3.2 手动模式—关节运动	12
3.3 手动模式—笛卡尔运动	13
3.4 拖动机器人	15
第 4 章 程序模块	18
4.1 程序管理器	18
4.2 程序选项卡	19
4.2.1 程序界面结构	20



4.2.2 示例 1: 运行一个示例程序	20
4.2.3 示例 2: 编写第一个程序	25
4.2.4 拖动示教	40
4.2.5 软 PLC	46
<b>第 5 章 安装设置选项卡</b>	<b>51</b>
5.1 通用类	52
5.1.1 安装角度	52
5.1.2 工具坐标系	53
5.1.3 用户坐标系	58
5.1.4 负载辨识	59
5.1.5 轨迹跟踪	65
5.1.6 初始位姿	67
5.1.7 模块更新	71
5.1.8 IO 配置	72
5.1.9 固定视觉	75
5.1.10 附加轴	84
5.1.11 关节滑移补偿	95
5.1.12 传送带跟踪	96
5.2 安全类	111
5.2.1 区域监控	112
5.2.2 位置监控	119
5.2.3 法兰设置	122
5.2.4 碰撞检测	127
5.2.5 安全限制	136
5.2.6 安全平面	140
5.2.7 身体风险	137



5.2.8 安全传感器	138
5.3 出厂设置类	143
5.3.1 滑移基准点	144
5.3.2 驱动器清零	145
5.3.3 DH 参数	146
5.3.4 关节限制	147
5.4 总线类	147
5.4.1 TCP/IP	147
5.4.2 MES	156
5.4.3 EtherCAT 设置	159
5.4.4 EthernetIP 设置	160
第6章 移动选项卡	162
6.1 位姿信息	162
6.2 坐标系统	163
6.3 点动机器人 (JOG)	163
6.3.1 关节运动	164
6.3.2 笛卡尔运动	164
6.3.3 步进运动	165
6.4 激活坐标系	166
6.4.1 激活工具坐标系	166
6.4.2 激活用户坐标系	167
6.5 初始位姿&回零位	167
6.5.1 初始位姿	167
6.5.2 回零位	168
第7章 监控选项卡	169
7.1 关节状态监控	169



7.2 I/O 监控	170
7.2.1 强制 I/O 信号	170
7.3 现场总线数据监控	172
7.3.1 Modbus 功能	172
7.3.2 Profinet 功能	180
7.3.3 EtherCAT 功能	194
7.3.4 EthernetIP 功能	205
<b>第 8 章 日志选项卡</b>	<b>218</b>
8.1 报警处理	218
8.2 系统日志	219
8.3 程序日志	219
<b>第 9 章 HBB 菜单</b>	<b>220</b>
9.1 关于	220
9.2 系统设置	221
9.3 帮助	221
<b>附录 A 报警及警告代码表</b>	<b>223</b>
A.1 系统报警代码	223
A.2 用户报警代码 (1400~1499)	226
A.3 重大错误代码 (1800~1899)	232
A.4 一般报警代码 (3900~3999)	237
A.5 用户报警代码 (4900~4999)	240
A.6 事件日志 (4900~4999)	251
<b>附录 B 文档修订记录</b>	<b>252</b>



# 第 1 章 示教器

## 1.1 概述

示教器是操控机器人的手持设备，通常情况下采用左手握持示教器，右手点击触摸屏进行交互，下图 1-1 为正确的握持方式。

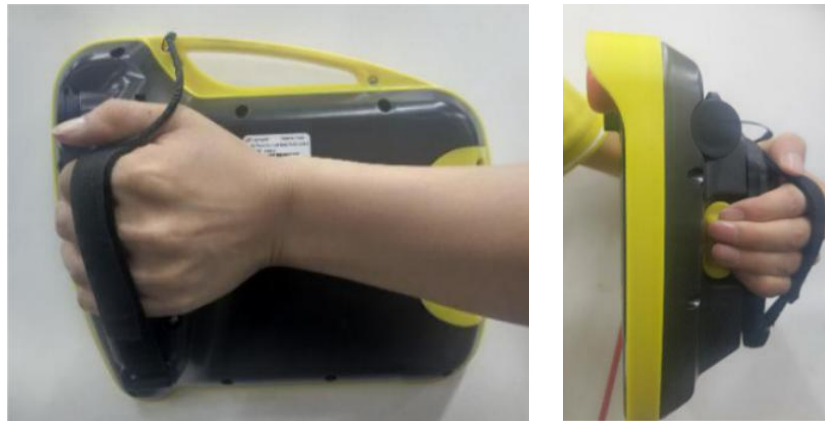


图 1-1 手持示教器正确姿势

## 1.2 组成

示教器的组成部件位置见图 1-2、图 1-3，示教器组成部件说明见表 1-1、表 1-2。



图 1-2 示教器正面

表 1-1 示教器正面部件说明

序号	名称	说明
1	显示屏（触摸屏）	HMI 操作区域。



2	功能键	提供部分功能的快捷键。
3	控制键	运行程序，手动点动机器人（JOG）。
4	指示灯	指示程序运行状态。
5	模式切换旋钮	自动、手动慢速、手动全速三种模式。
6	急停按键	按下急停按键，机器人停止运动； 顺时针旋转按钮，解除急停信号。

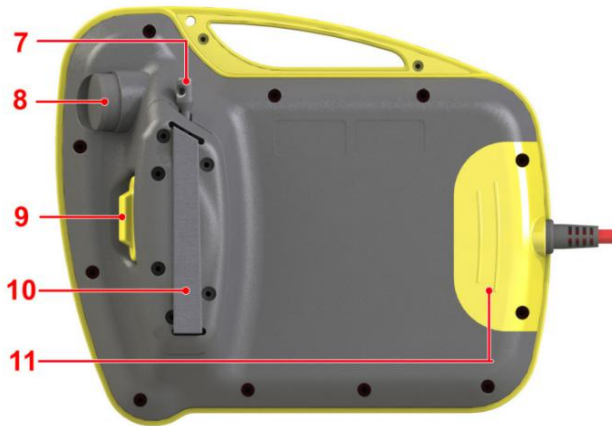


图 1-3 示教器背面

表 1-2 示教器背面部件说明

序号	名称	说明
7	触控笔	用于点击触摸屏。
8	USB 接口保护盖	保护 USB 接口。
9	使能键	具体使用方式参看 1.2.2 节。
10	手带	手持示教器时，防止其脱手。
11	出线盖板	更换线束时，需要打开此盖板。

### 1.2.1 模式切换旋钮

示教器面板右上方的模式切换旋钮（图 1-4），三种模式可参考表 1-3。



图 1-4 模式切换旋钮




图 1-5 状态栏对应位置

表 1-3 模式切换旋钮说明：

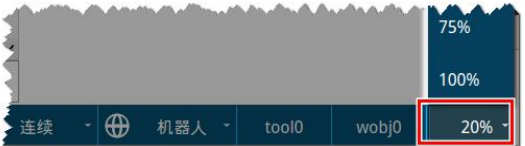


名称	图标	说明
Auto（自动模式）		运动速率可设置范围（0~100%）。
T1（手动慢速模式）		运动速率可设置范围（0~20%）。
T2（手动全速模式）		运动速率可设置范围（0~100%）。



**提示！**

“运动速率”：机器人实际运动速度与电机属性中所设置的最高运动速度的比率（该数值在系统界面状态栏最右一个图标，如下图）。



### 1.2.2 使能按键

使能按键位于示教器背部左侧，涉及到运动机器人的操作都需要持续按下该键，以保持机器人的运动（自动运行程序时除外）。按下使能键后，软件界面状态栏中对应图标变更为绿色底色（见图 1-7）。



图 1-6 使能键

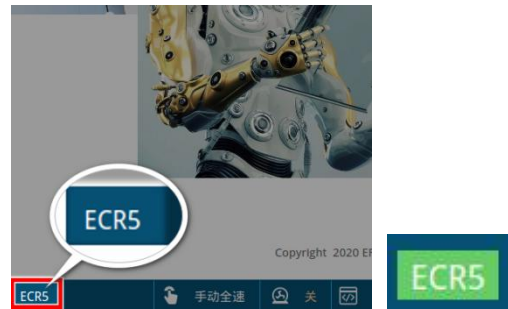


图 1-7 状态栏图标提示

表 1-4 机器人运动应用表

应用	使能键
手动模式下，JOG 机器人	使用
手动模式下，运行程序	使用
自动模式下，拖动机器人（参看 3.4 节）	使用
自动模式下，运行程序	不需要



### 1.2.3 控制键

手动运动机器人（又称 JOG 机器人，参看 3.1~3.3 节）有 2 种方式：

- ◆ 关节模式：控制机器人各关节独立正转/反转；
- ◆ 笛卡尔模式：控制机器人末端在笛卡尔坐标系下，沿 X、Y、Z 轴的正方向/负方向直线运动，或者绕 Z、Y、X 轴旋转。

手动运动机器人时，控制键在上述 2 种方式下，各键所对应的含义见下图 1-8。

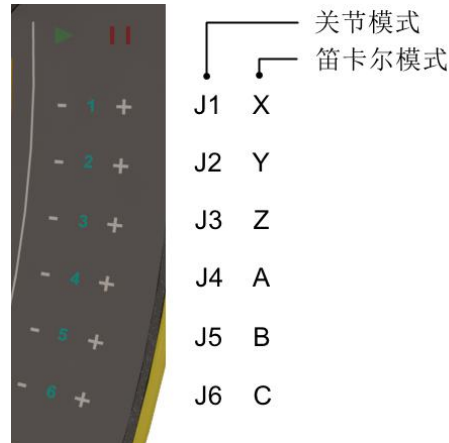


图 1-8 示教器控制键

表 1-5 控制键说明

标识	关节模式	笛卡尔模式
	开始执行程序	
	暂停程序	
	控制单个关节正向运动	控制机器人执行端沿笛卡尔坐标系相应坐标轴正向直线运动或绕轴旋转
	控制单个关节反向运动	控制机器人执行端沿笛卡尔坐标系相应坐标轴负向直线运动或绕轴旋转

### 1.2.4 功能键

功能键相应功能见表 1-6。

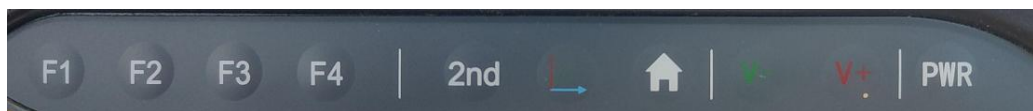


图 1-9 示教器功能键



表 1-6 功能键说明表

序号	图标	使用说明
1		未启用
2		未启用
3		切换程序运行方式，参看 4.2.2 节单步调试程序
4		未启用
5		使用附加轴时，可切换控制键对附加轴操控，详细参见附加轴章节
6		切换机器人运动参考系
7		未启用
8		减少机器人运动速率
9		增加机器人运动速率
10		机器人上/下伺服

### 1.2.5 急停按键

急停按键位于示教器面板右上角。当发生紧急情况时，快速按下此按键可停止机器人运动。

#### 使用步骤：

步骤 1：必要时按下急停按键，示教器界面弹出“1802 急停信号”报警横幅、“紧急停止”图标，见图 1-10；



图 1-10 示教器控制键

步骤 2: 顺时针旋转急停按键至按键弹起，可解除急停信号；

步骤 3: 点击示教器“报警”横幅中“处理核查”按钮（下图 1-11），手动清除该报警，即可将机器人恢复到正常状态。



图 1-11 清除报警



# 第 2 章 系统界面

## 2.1 主界面

ECR-HMI 是 ECR 系列协作机器人人机交互系统，用户可通过该系统操纵机器人的本体运动、设置参数、编写及运行程序。

该系统软件界面由三个区域组成，如下图 2-1：



图 2-1 HMI 界面

表 2-1 软件界面布局说明

#	区域	说明
1	导航栏	包括导航栏、程序管理器、HBB 菜单
2	内容区	内容显示区，包括编写程序、功能设置、参数设置等
3	状态栏	显示机器人的状态

### 2.1.1 导航栏





图 2-2 导航栏

表 2-2 导航栏说明表

#	说明
1	用户登录页
2	程序模块：编辑、调试 RPL 程序
3	安装设置模块：机器人基础功能设置
4	移动模块：切换工具、用户坐标系，显示机器人实时姿态
5	监控模块：监控 I/O、关节力矩/温度、总线数据
6	日志模块：查看运行记录
7	加载的 RPL 程序名
8	新建 RPL 程序
9	打开程序文件管理器
10	保存 RPL 程序
11	汉堡包菜单：包括关于、系统设置等

### 2.1.2 状态栏



图 2-3 状态栏

表 2-3 状态栏图标说明

#	说明
1	使能键状态（按下/未按下）
2	系统状态提示（未连接、运动受限、急停信号）
3	模式说明（自动模式、手动慢速、手动全速）
4	伺服状态（开/关）
5	程序状态（初始化、执行、暂停、停止、错误）
6	程序执行方式（连续运行、单步运行） 其中单步运行分为 4 种：单步进入、单步跳过、运动进入、运动跳过。（具体参看 4.2.2 节中“单步调试程序”）
7	运动机器人（JOG）时，机器人的运动参考系： ◆ 关节（关节模式）、 ◆ 机器人坐标系（笛卡尔模式）、 ◆ 工具坐标系（笛卡尔模式）、 ◆ 用户坐标系（笛卡尔模式）。 （关节模式、笛卡尔模式，参看 1.2.3 节。）



8	当前使用的工具坐标系
9	当前使用的用户坐标系
10	机器人运动速率

## 2.2 登录页

通电开机、系统加载完毕后，登录页为 ECR-HMI 系统初始显示页，如下图 2-4。

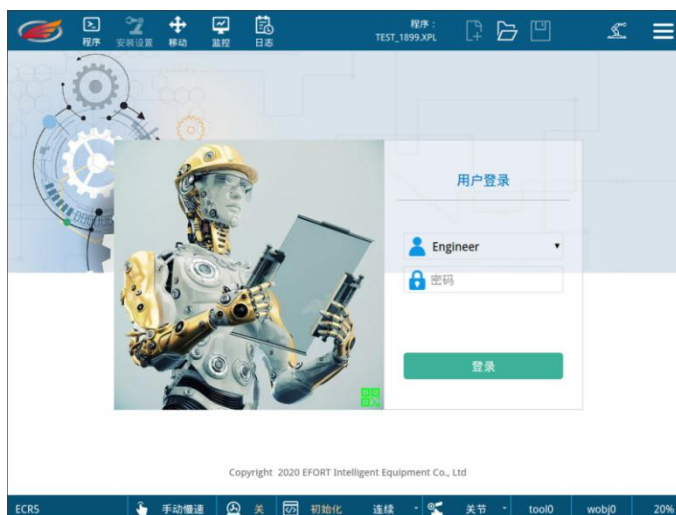


图 2-4 登录页

也可通过点击“LOGO”图标，切换出“登录页”界面，如下图 2-5；



图 2-5 导航栏“LOGO”选项卡

### 2.2.1 用户权限

ECR 机器人示教器操作系统有如下三种登录角色，分别是：

- ◆ Operator（操作员）：无法主动激活，默认登录用户，不需要密码；
- ◆ Engineer（工程师、技术员）：可主动激活，需要用户密码；
- ◆ Administrator（管理员）：可主动激活，需要用户密码。

操作员和工程师角色在使用示教器系统时，存在部分功能限制；


管理员没有任何限制。用户角色及其限制见下表 2-4；

表 2-4 用户权限表

	操作员	工程师	管理员
--	-----	-----	-----



程序模块	✘	✓	✓
安装设置模块	✘	○	✓
移动模块	✓	✓	✓
监控模块	✓	✓	✓
日志模块	○	✓	✓
程序管理器	○	✓	✓
HBB 菜单	○	○	✓
<p>注：</p> <ul style="list-style-type: none"> <li>✘表示模块中的所有功能都不能编辑；</li> <li>○表示模块中的部分功能可以编辑；</li> <li>✓表示模块中的所有功能都可以编辑。</li> </ul>			

	<p><b>提示！</b></p> <p>Engineer（工程师）、Administrator（管理员）登录密码，请咨询本公司售后人员。</p>
---	---



# 第 3 章 机器人基础操作

## 3.1 手动模式+上伺服

手动运动机器人（又称 JOG），运动方式可分为：

- ◆ 关节运动： 控制机器人各关节独立正转/反转；
- ◆ 笛卡尔运动： 控制机器人末端在笛卡尔坐标系下，沿 X、Y、Z 轴的正方向/负方向直线运动，或者绕 Z、Y、X 轴旋转。

**JOG 机器人的前提：**机器人处于“手动模式+上伺服”状态。

具体操作：

步骤 1： 将示教器面板上的“模式切换”旋钮，转动至“手动慢速（T1）”，见下图 3-1（左）。切换后，软件界面“状态栏”的相应图标状态为图 3-1（右）。

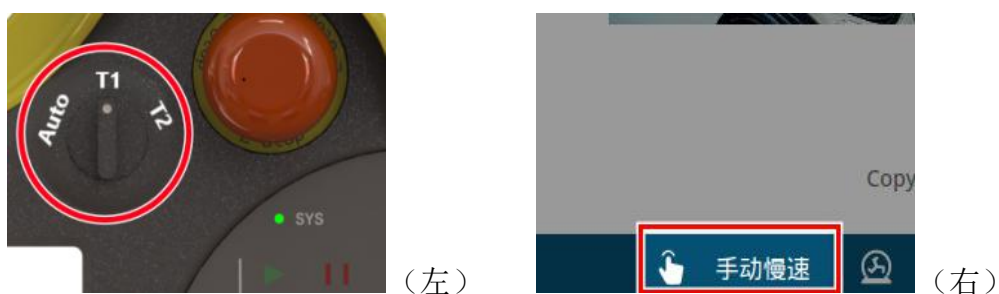


图 3-1 模式切换

步骤 2： 点击示教器面板上“PWR”按钮（图 3-2 左），使机器人伺服上电（上伺服）。上伺服过程需要 3~5 秒时间，期间机器人各关节发出 1 次“咔嚓”声音（共 6 次），上伺服成功后，软件界面“状态栏”相应图标状态见图 3-2（右）。



图 3-2 上伺服



## 3.2 手动模式—关节运动

首先完成 3.1 节“手动模式+上伺服”的操作，再进行如下操作：

步骤 1：点击示教器面板上的“坐标系”按键（见图 3-3 左），直到将机器人运动方式切换为“关节”，如图 3-3（右）。



图 3-3 “关节运动”模式

	<p><b>提示！</b></p> <p>在“自动模式”下，系统会禁用运动参考系。运动参考系的切换只可以在“手动慢速”、“手动全速”模式下生效。</p>
--	--

步骤 2：轻轻扣住图 3-4（左）中的“使能键”（位于示教器背面左侧），这时软件界面“状态栏”相应图标状态如图 3-4（右）。

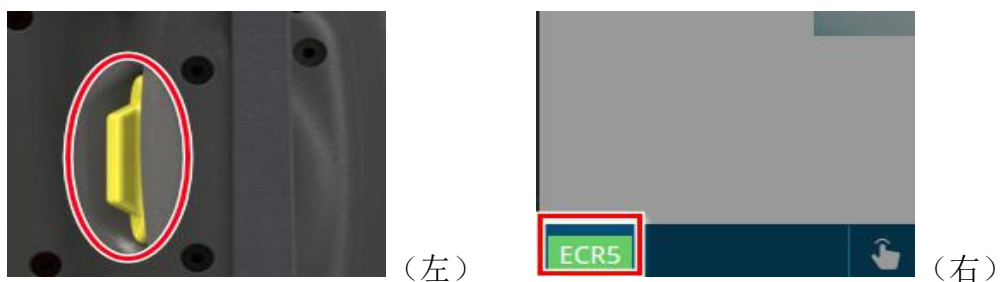


图 3-4 使能键

步骤 3：左手保持持续按下“使能键”，右手长按示教器面板上①控制键“-”或“+”（如下图 3-5），用以控制机器人 1 轴（1 号关节）反向或正向转动（松开“使能键”或“控制键”，机器人立即停止运动）。

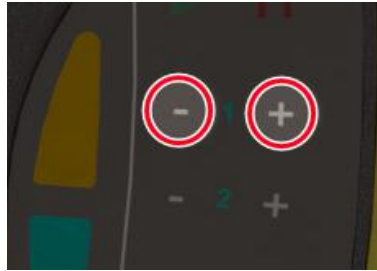


图 3-5 关节 正/反转动

	<p><b>提示!</b></p> <p>点击示教器面板上“V-、V+”按键（见下图），可调节机器人运动速率。</p> <div data-bbox="715 835 1157 929"><p>（面板）</p></div> <div data-bbox="715 967 1157 1064"><p>（界面）</p></div>
--	--

	<p><b>提示!</b></p> <p>“手动慢速”模式下，机器人运动速率范围为 0~20%， “手动全速”模式下，机器人运动速率范围为 0~100%。</p>
--	---

### 3.3 手动模式—笛卡尔运动

**预备操作:** 在使用笛卡尔运动方式之前，对机器人的起始姿态有一定要求，即同时具备以下 2 个条件：

- ◆ 机器人大臂、小臂之间需要产生夹角（10° 以上），见图 3-6（左）。
- ◆ 机器人#1 手腕、#3 手腕不能平行，需有夹角（20° 以上），见图 3-6（右）。

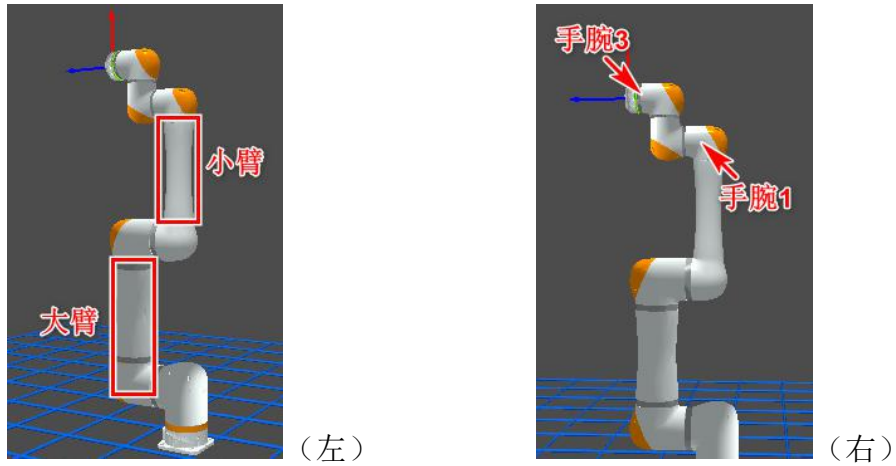


图 3-6 机器人本体

下图中，各关节角度为推荐的笛卡尔运动起始姿态，使用“手动+关节运动”将机器人运动至此位姿（参看 3.2 节内容）。

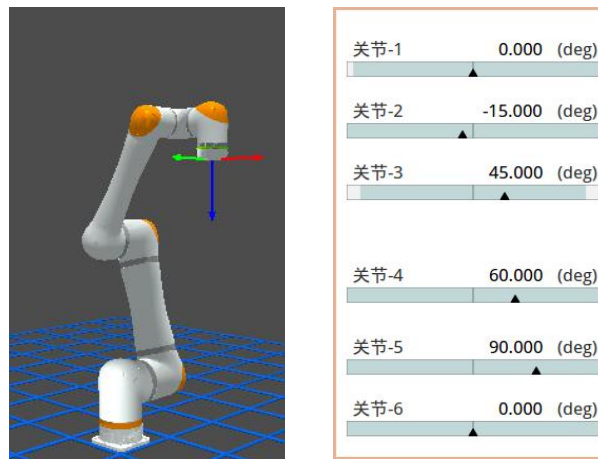




图 3-7 笛卡尔运动起始姿态



**提示！**

上图 3-7 内容可通过点击示教器软件界面中“移动”选项卡，进入移动页面查看。



**“笛卡尔运动”操作步骤：**

- 步骤 1： 确认当前模式为“手动模式（T1）+上伺服”状态（参看 3.1 节）；
- 步骤 2： 将机器人运动参考系切换至“机器人”（通过点击示教器面板上“坐标系”按键），见下图 3-8；



图 3-8 “机器人”运动参考系

	<p><b>提示!</b></p> <p>在“自动模式”下，系统会禁用运动参考系。运动参考系的切换只可以在“手动慢速”、“手动全速”模式下生效。</p>
--	--

步骤 3: 左手轻扣住“使能键”（下图 3-9 左），同时右手长按示教器面板上①的“-”或“+”按键（下图 3-9 右），机器人会以基座为坐标系原点的 X 轴负方向或正方向做直线运动。



图 3-9 机器人沿 X 轴直线运动

	<p><b>提示!</b></p> <p>当机器人以“机器人”为运动参考系时，示教器面板上的①~⑥号“-”“+”按键，分别对应机器人以基座为坐标系原点的运动，如下：</p> <table data-bbox="544 1702 1319 1861"><tr><td>①: X 轴 正/负方向</td><td>④: A (以 Z 轴顺/逆时针旋转)</td></tr><tr><td>②: Y 轴 正/负方向</td><td>⑤: B (以 Y 轴顺/逆时针旋转)</td></tr><tr><td>③: Z 轴 正/负方向</td><td>⑥: C (以 X 轴顺/逆时针旋转)</td></tr></table>	①: X 轴 正/负方向	④: A (以 Z 轴顺/逆时针旋转)	②: Y 轴 正/负方向	⑤: B (以 Y 轴顺/逆时针旋转)	③: Z 轴 正/负方向	⑥: C (以 X 轴顺/逆时针旋转)
①: X 轴 正/负方向	④: A (以 Z 轴顺/逆时针旋转)						
②: Y 轴 正/负方向	⑤: B (以 Y 轴顺/逆时针旋转)						
③: Z 轴 正/负方向	⑥: C (以 X 轴顺/逆时针旋转)						

### 3.4 拖动机器人

拖动示教是相较于 JOG 机器人的另一种运动方式。开启拖动示教模式后，操



作业人员可以通过拖拽机器人本体，以此改变机器人的姿态。

**预备操作：**同上节图 3-7 “笛卡尔运动起始姿态”相同，即大、小臂；手腕 1、3，需要形成一定的角度（关节角度仅供参考）；

**操作步骤：**

步骤 1：模式旋钮切换至“自动模式（Auto）”，点击“PWR”按钮是关节“上伺服”；

步骤 2：点击一次机器人末端法兰上的“□”按钮（见图 3-10）；

（“□”按钮的灯会以绿色呼吸状闪烁，表示机器人已进入“拖动模式”）

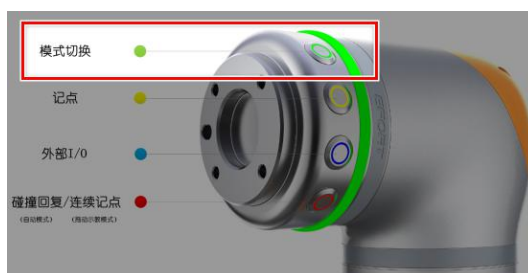


图 3-10 拖动模式按钮

步骤 3：左手轻扣住示教器背部“使能键”，此时末端法兰“●”按钮灯亮红色、末端法兰灯带亮绿色，表示现在可以以拖拽的方式移动机器人；

步骤 4：保持扣住“使能键”，同时拖动机器人，如下图 3-11；

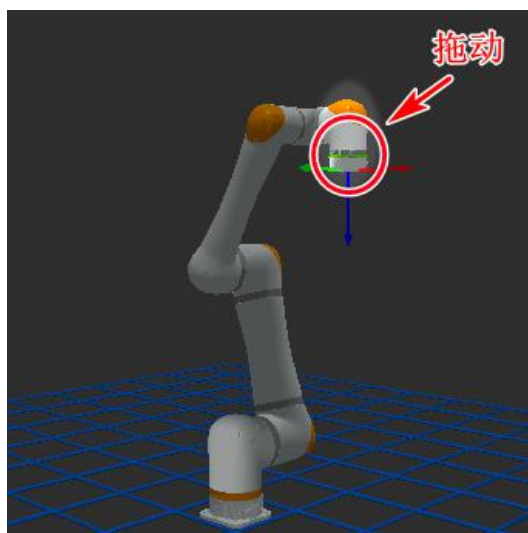


图 3-11 拖动示教



**提示！**

松开“使能键”，末端法兰灯带转为橙色，此时无法拖动机器人。

当末端装有负载时，拖动机器人前需进行负载



步骤 5: 退出“拖动模式”有 2 种方法:

方法 1: 将模式旋钮切换到“手动模式 (T1)”或“手动模式 (T2)”, 即退出拖动模式。

方法 2: 绿色按键灯闪烁的前提下, 点击 2 次“□”按键, 该按键灯熄灭, 表示退出拖动模式。



**提示!**

末端法兰上“□”按键灯有 3 种状态:

**【绿色呼吸闪烁】、【绿色常亮】、【灯灭】。**

每点击一次该按键 (必须在“自动模式+上伺服”状态下), 按键灯状态会依次切换, 即“呼吸闪烁” → “常亮” → “灯灭 (关闭)”。

呼吸闪烁、常亮所代表的意思, 参看 4.2.4 节内容。



# 第 4 章 程序模块

## 4.1 程序管理器



图 4-1 程序文件管理器

程序管理器中只可识别后缀为“XPL”的文件，在此页面中可对已有的程序文件进行打开、复制、删除等操作。点击导航栏中“打开文件”按钮（如下图 4-2）弹出程序管理器页面，点击“×”按钮（如下图 4-3）关闭管理器。



图 4-2 打开程序管理器



图 4-3 关闭/刷新程序管理器



### 提示！

Operator（操作员）用户只可使用“打开”、“USB 导入/导出”功能，如下图。



### 警告!

使用 U 盘向示教器导入程序时，程序文件的扩展名必须使用大写，比如“demo.XPL”；切忌使用小写比如“demo.xpl”，如果使用小写的扩展名会造成示教器软件无法识别出该程序文件。

程序文件视图		
文件名	大小	日期
CFDW.XPL	1.2 KB	2020-12-21 16:03
<u>DEMO.XPL</u>	8.6 KB	2020-10-16 11:23
ECR5_CALIB_TEST.XPL	11.2 KB	2020-12-21 15:40
FRICITIONUPATING.XPL	8.7 KB	2020-09-27 06:52
PACKAGE.XPL	765 B	2020-08-26 03:27
REHOME.XPL	586 B	2020-12-21 18:22
TEST.XPL	1.3 KB	2021-10-25 09:50

## 4.2 程序选项卡

程序选项卡实现：编辑程序、调试程序、运行程序功能。



图 4-4 程序选项卡



### 4.2.1 程序界面结构

界面由三个区域组成：指令栏、程序树、参数设置/调试区（见下图 4-5）。



图 4-5 界面布局

“指令栏”：包含机器人所有编程可用指令；

“程序树”：显示当前程序指令代码；

“参数设置”：显示程序树中当前选中指令的详细参数，以及编辑。




**提示！**

点击“扩展”按钮，可以扩展“程序树”区域。



### 4.2.2 示例 1：运行一个示例程序

在本示例中，按照操作步骤可实现：程序的加载、自动模式下执行程序、手动模式下单步调试程序。



**警告！**

操作以下步骤时，确保机器人臂展范围内空旷，没有人或物体。



## 自动运行程序

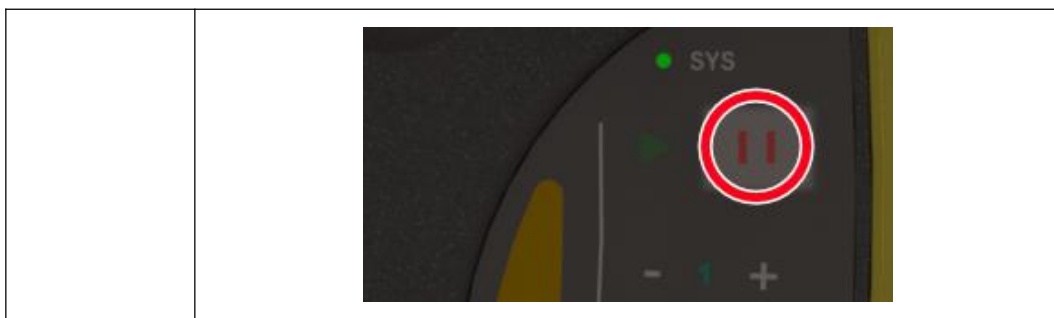
自动模式运行程序，操作步骤：

#	图示	说明
1		<p>① 点击“打开程序管理器”按钮；</p> <p>② 选择示例程序“DEMO.XPL”；</p> <p>③ 点击“打开”按钮，等待加载该程序完毕。</p>
2		<p>程序加载完毕后；</p> <p>① 旋转模式旋钮至“Auto”+“上伺服”（示教器面板上“PWR”按钮）。</p>
3		<p>① 点击“程序运行”键；</p> <p>机器人开始按照程序中的指令运动。</p>
4		<p>程序执行时，状态栏中状态由“初始化”变为“执行”。</p>



### 提示！

点击“程序暂停”按钮，可暂停程序，见下图：



## 单步调试程序

单步调试指：对一个程序段，以逐条指令的方式执行，而非连续执行程序段。

单步调试程序，操作步骤：

#	图示	说明
1		<p>① 点击“程序暂停”键，暂停程序；</p> <p>② 点击“终止”按钮，终止程序。</p>
	<p>(暂停程序)</p>	<p><b>提示！</b></p> <p>‘暂停程序’与‘终止程序’含义不同：</p> <p><b>暂停程序：</b>只是暂停程序的执行，而控制器中的程序仍处于工作状态。</p> <p><b>终止程序：</b>将程序由工作状态转为停止状态，程序进入可编辑状态。</p> <p>机器人控制系统中，很多操作（比如：Jog 机器人）都需要在‘终止程序’的状态下才可以操作成功。</p> <p>(终止程序)</p>



2		<p>① 将“模式旋钮”转至‘手动慢速(T1)’挡。</p>
3		<p>① 点击“重新开始”按钮，将程序转为就绪状态，即“初始化”状态。</p>
4		<p>① 选择程序运行方式为“单步进入”。</p>
5		<p>① 左手轻轻扣住示教器背面“使能键”，点击“程序运行键”。</p> <p>注：每点击一次“程序运行键”，程序段每执行完一行程序后，就会暂停程序，不再往下执行。</p>
		<p><b>提示！</b></p> <p>手动调试程序时“使能键”要始终扣住，如果松开，程序会立即“暂停”。</p>

表 4-1 程序执行模式说明

模式	说明
连续	程序开始执行后，一直运行到程序末尾结束执行。
单步进入	主程序（Main）每执行完一行后暂停程序。 主程序调用子程序时，会进入子程序，并且执行子程序中的一行后暂停程序。
单步跳过	主程序（Main）每执行完一行后暂停程序。



	主程序调用子程序时，一次性执行完子程序内的全部程序行。
运动进入	主程序每执行至下一条运动指令前停下，当执行子程序时会进入子程序的界面，并且一次性运行完子程序内全部指令。
运动跳过	程序开始执行后，一直运行到程序末尾结束执行。

## Setpc 操作

Setpc 是指：将程序光标设定到指定的某行，这样点击“程序运行”键之后，程序从该行开始执行程序。

Setpc 操作步骤：

#	图示	说明
1		<p>① 承接上一节“单步调试程序”的操作：</p> <p>② 以单步调试的方式将程序执行到第 11 行。</p>
2		<p>① 选中程序段中第 20 行即点击一下第 20 行；</p> <p>② 点击“Setpc”按钮。</p>
3		<p>程序光标跳到该行，表示 Setpc 完成；</p> <p>重复“单步调试程序”的操作。</p>



### 提示！

使用 Setpc 时，建议在 Setpc 之后的第一条运动机器人的指令使用 MJOINT 指令，否则会小概率出现意外的运动情况。



表 4-2 程序指针状态说明

图标	说明
	程序处于初始化状态，且当前行即为程序开始执行的第一条指令。
	表示当前行处于预备状态，可以执行。
	表示当前行处于激活状态，在运行中。
	表示当前程序行有错误。
	表示当前程序行存在机器人运动相关指令。
	表示当前行处于激活状态，且有运动在执行。
	当前行（含运动指令）有错误。

### 4.2.3 示例 2：编写第一个程序

本示例中会演示如何创建新程序、插入机器人运动指令、新建变量以及子程序。

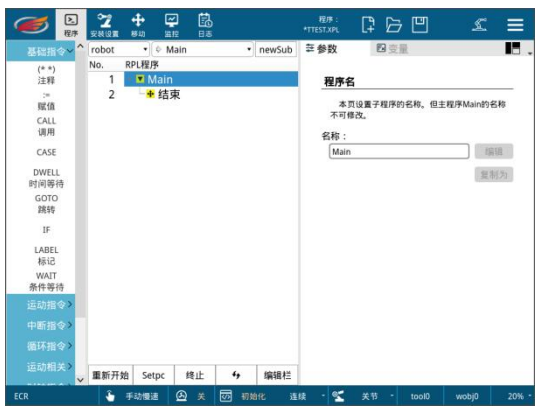
	<p><b>警告！</b></p> <p>操作以下步骤时，确保机器人臂展范围内空旷，没有人或物体。</p>
--	---

### 新建程序

新建程序，操作步骤：

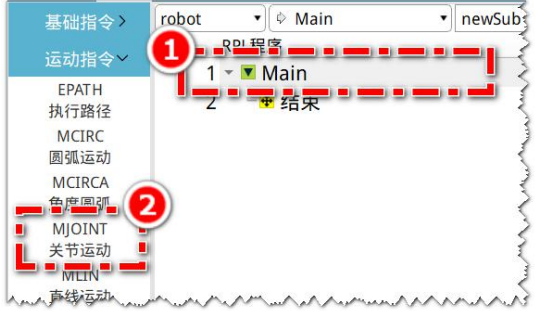
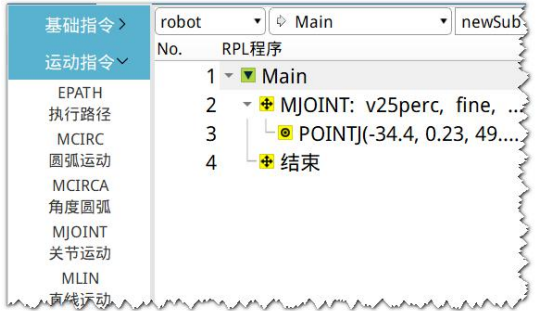
#	图示	说明
1		<p>① 点击导航栏上“新建”按钮；</p> <p>② 将键盘切换为英文输入；</p> <p>③ 输出新程序名称；</p>



		④ 点击“√”按钮，完成创建。
2		等待创建文件、并完成加载。

## 插入 MJOINT 指令

插入指令，操作步骤：

#	图示	说明
1		<p>① 点击选中“Main”行；</p> <p>② 点击指令栏中“运动指令/MJOINT”指令，完成指令插入到程序树中。</p>
2		完成指令插入。



### 提示！

这里有一个使用技巧：

当选中程序树中“Main”行，再点击指令栏中的指令，则该条指令会以顺序的方式添加为程序树的最后指令行。



如果选中的不是“Main”行（程序首行），那么再点击指令栏的指令时，该条指令会在选中行的“上方”倒序插入到程序树中。

以下演示倒序插入指令行：

倒序插入指令，操作步骤：

#	图示	说明
1		<p>① 承接上面，点击程序树中第 2 行，“MJJOINT”行；</p> <p>② 点击指令栏中“MLIN”指令。</p>
2		完成倒序插入指令。

## 示教路点

在程序树中插入了 2 条运动指令，系统会自动将机器人当前的姿态数据填入到这 2 条指令的路点数据中（即，target）。如果需要修改 target 值，有 2 种方式可以修改：

第 1 种：直接修改

#	图示	说明
---	----	----



1		<p>① 点击选中第 5 行；</p> <p>② 在右侧的“参数”页中，点击编辑框，并修改数值；</p> <p>③ 点击“应用”按钮，保存修改。</p>
---	--	--

第 2 种：示教方式

#	图示	说明
1		<p>① 以手动模式—关节运动方式将机器人各关节转动至图中的数值。（参看第 3 章 3.2 节。）</p>
2		<p>① 选中程序树中第 5 行；</p> <p>② 点击“参数”页中，“示教”按钮（图中红字为姿态数据变更）；</p> <p>③ 点击“应用”按钮，保存数据。</p>



3		示教完成。
---	--	-------

	<p><b>提示!</b></p> <p>程序树中涉及到机器人运动的第 1 条指令，建议使用 MJOINT。（所以删除示例中的 MLIN 指令，操作如下）</p>
--	---

4		<p>① 点击“编辑栏”按钮；</p> <p>② 选中第 2 行 MLIN 指令；</p> <p>③ 在弹出的编辑条中，点击“删除”按钮。</p>
---	--	---

5		① 点击“是”，完成删除操作
---	--	----------------



① 再次点击“编辑栏”按钮或“收起”按钮，隐藏编辑条。

以相同方式在程序树中再加入 4 个姿态（采用 MJOINT 指令），并保存程序，姿态数据，见下图：

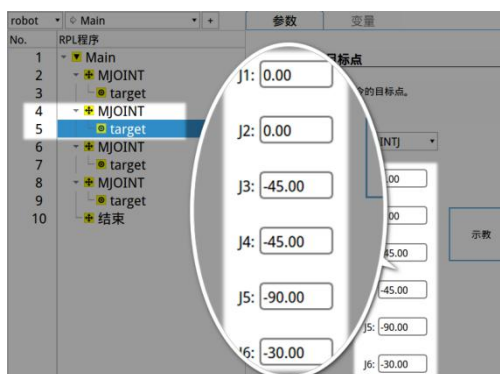


图 4-6 第 2 个点

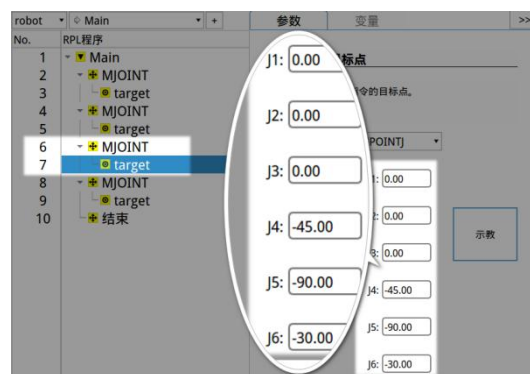


图 4-7 第 3 个点

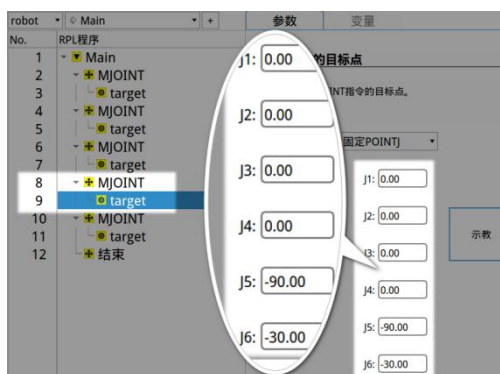


图 4-8 第 4 个点

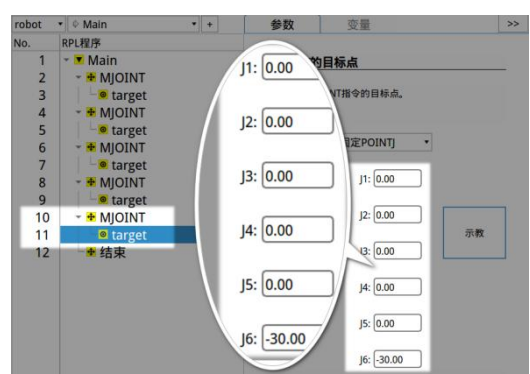


图 4-9 第 5 个点



**提示！**

点击导航栏“保存”按钮，可保存程序，如下图：





完成上述步骤后，以“自动运行程序”方式（参看 4.2.2 节）运行程序，可以发现机器人从一个起始姿态运动至零位（除了第 6 个轴），之后程序结束、机器人停止。

## 新建变量

现在将上述轨迹重复执行 6 次，可通过添加变量和“循环指令/FOR”指令实现。实现流程如下：

流程 1：创建变量；

流程 2：设置变量初始值；

流程 3：在程序树中插入 FOR 指令（FOR 指令需要该变量作为指令参数）。

流程 1 “创建变量”操作步骤：

#	图示	说明
1		<p>① 点击“变量”标签页；</p> <p>② 在标签页中，选中“Main”项；</p> <p>③ 点击“新建”按钮。</p>
2		<p>① 点击“数字/DINT”项，设置变量类型为 DINT 型；</p> <p>② 修改变量名称为“iter”（也可使用默认“i_0”）；</p> <p>③ 点击“确认”按钮，完成创建变量。</p>



3		<p>变量创建，完成。</p>
---	--	-----------------

	<p><b>提示！</b></p> <p>在上述流程步骤 1 中选择了“Main”，这是为了设置变量的【作用域】，作用域的概念，请查看另一份文档“RPL 编程手册”。</p>
--	---

流程 2 “设置变量初始值” 操作步骤:

#	图示	说明
1		<p>① 选中“iter”变量；</p> <p>② 点击“编辑”按钮。</p>
2		<p>① 点击“更多设置”按钮。</p>



<p>3</p>		<p>① 勾选“使用初始值”，并设定值为“0”；</p> <p>② 点击“确认”按钮。</p>
<p>4</p>		<p>变量设置初始值，完成。</p>

流程 3 “插入 FOR 指令” 操作步骤:

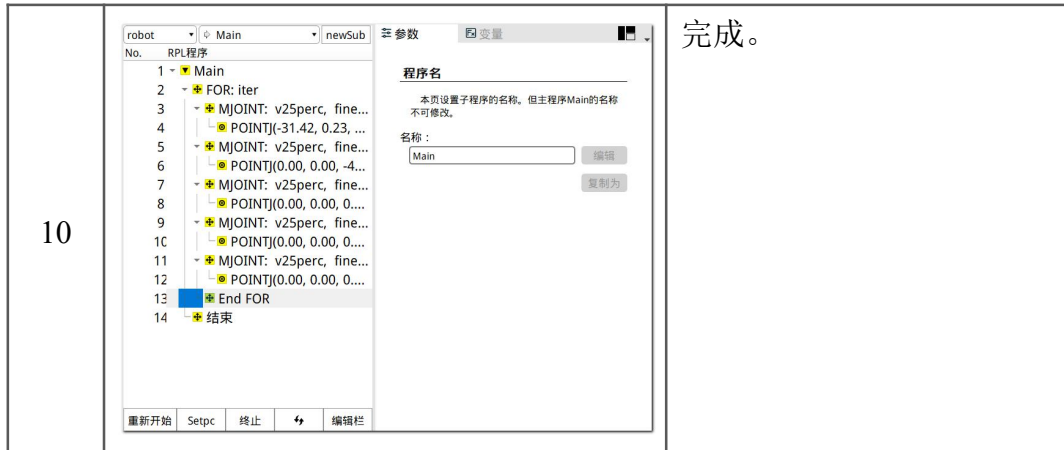
#	图示	说明
<p>1</p>		<p>① 选中第 2 行；</p> <p>② 点击“循环指令/FOR”指令。</p>



2		<p>① 选中第 2 行 FOR;</p> <p>② 点击“参数”页中，“选择变量”按钮，并在列表中选出“iter”变量。</p>
3		<p>① 点击“起始值”栏后面的“编辑”按钮，会弹出“表达式编辑器”;</p> <p>② 点击表达式编辑器中的“123 数字键盘”按钮;</p> <p>③ 在弹出的键盘键入“0”</p> <p>④ 点击“√”输入数值。</p>
4		<p>① 点击“确认”按钮，完成编辑。</p>
5		<p>依次完成“终止值”、“步进值”的设置。</p> <p>终止值：5 步进值：1</p>



<p>6</p>		<p>① 点击“应用”按钮，完成 FOR 指令设置。</p>
<p>7</p>		<p>① 选中程序树中多行程序。 ② （多选操作：点住第 4 行，向下拖动直至第 13 行）</p>
<p>8</p>		<p>① 点击“编辑栏”按钮，弹出编辑条； ② 点击编辑条中“剪切”按钮。</p>
<p>9</p>		<p>① 选中程序树中第 3 行“End FOR”； ② 再点击编辑条中“粘贴”按钮，将多条运动指令移入 FOR 循环体中。</p>



再次以自动模式运行程序，机器人会执行 6 次运动后停止，程序结束。

### 创建子程序

承接上一节操作内容，现在需要将主程序中第 3 行~第 13 行的程序行封装成一个子函数，并且被主程序调用，如下图 4-10：

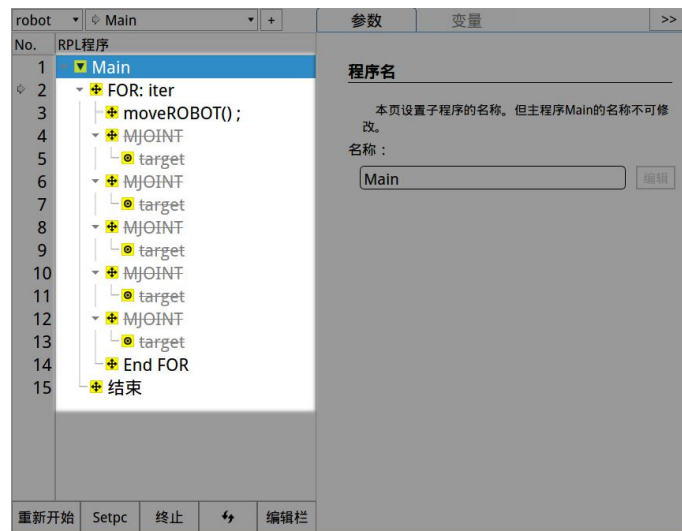


图 4-10 调用子程序

图 4-10 中第 3 行为子程序的名称；第 4 行~第 13 行表示被注释，即不再运行这段代码。

上图中程序实现步骤：

#	图示	说明
---	----	----



<p>1</p>		<p>① 点击“newSub”按钮，添加一个子程序；</p> <p>（点击“newSub”后，程序树自动创建 sub1 子程序，并跳转到子程序中）。</p>
<p>2</p>		<p>① 点击“编辑”按钮；</p> <p>② 更名为“moveROBOT”</p> <p>③ 点击“√”，保存。</p>
<p>3</p>		<p>① 在“moveROBOT”子程序中添加 5 条 MJOINT 指令，并示教 5 个路点。</p>
<p>4</p>		<p>① 点击“程序切换”下拉框，重新切回到主程序（“Main”）。</p>



<p>5</p>		<p>① 点击选中主程序中的第 3 行；</p> <p>② 点击“基础指令/CALL”指令，插入。</p>
<p>6</p>		<p>① 选中程序树中“???; ! error 1”（即 CALL 指令）；</p> <p>② 点击“编辑”按钮，跳转出调用列表。</p>
<p>7</p>		<p>① 在候选列表中，选出“moveROBOT()”；</p> <p>② 点击“确认”按钮。</p>
<p>8</p>		<p>① 点击“应用”按钮，保存 CALL 指令的设置。</p>



<p>9</p>		<p>① 多选第 3~13 行程序；</p> <p>② 点击“编辑栏/注释”按钮，注释掉该多行程序。</p>
		<p>完成。</p>

	<p><b>提示！</b></p> <p>CALL 指令可以调用：子程序，Module 函数等。</p> <p>编写程序时，涉及到“调用”的操作都是以 CALL 指令实现。</p>
--	--

	<p><b>提示！</b></p> <p>子程序删除操作：首先将程序树切换到子程序，再选中子程序第 1 行，最后点击“编辑栏/删除”按钮。</p>
--	---

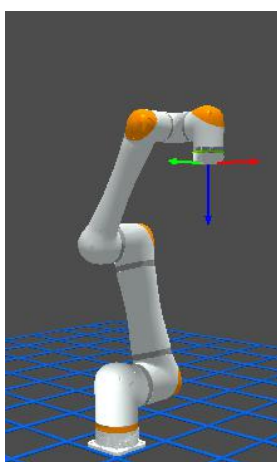


#### 4.2.4 拖动示教

第3章3.4节中介绍拖动机器人的操作，本节将说明结合拖动操作快速将机器人的位姿或运动轨迹插入到程序树中，即“拖动示教”。拖动示教有2种：示教点、示教轨迹。

##### 拖动示教点

**开启拖动的前提：**大臂与小臂需要形成一定角度，请使用JOG方式将机器人运动到下图中的推荐姿态。



关节-1	0.000 (deg)
关节-2	-15.000 (deg)
关节-3	45.000 (deg)
关节-4	60.000 (deg)
关节-5	90.000 (deg)
关节-6	0.000 (deg)

拖动示教点，操作步骤：

#	图示	说明
---	----	----



<p>1</p>		<p>① 新建程序，文件名为“DragTeaching”。</p>
<p>2</p>		<p>① 以当前机器人姿态，插入 MJoint 指令。</p>
<p>3</p>		<p>① 将机器人模式设置为：“自动模式”+“上伺服”。</p> <p>（注：因为拖动操作，需要在“自动模式”+“上伺服”模式下，才可以开启）</p>
<p>4</p>		<p>① 点击一次末端法兰上“□”方框按钮。</p> <p>（此时该按钮灯呈现绿色呼吸状闪烁）</p>
	<p><b>提示！</b></p> <p>“□”方框按钮，存在 3 种状态：呼吸闪烁、常亮、灯灭。每点击一次该按钮，就会依次切换为下一个状态，即“呼吸闪烁”-&gt;“常亮”-&gt;“灯灭（关闭）”。</p>	



5		<p>① 左手轻轻扣住“使能键”。</p> <p>（此时末端法兰的灯带会变为绿色,即表示可以开始拖动机器人）</p>
	<p><b>提示!</b></p> <p>扣住“使能键”的效果等同于,持续按下末端法兰上的“●”实心圆按钮,如下图:</p> 	
6		<p>① 将机器人拖动适当位置后,松开“使能键”; (本例中为竖直向下拖动)</p> <p>点击一次末端法兰上“△”三角按钮。</p>
	<p><b>提示!</b></p> <p>点击末端法兰上“△”三角按钮的作用为: 向程序树中插入当前机器人姿态。</p>	
7		<p>程序界面会弹出对话框 该对话框询问以哪种运动指令运动到待插入点。</p> <p>这里我们选择“MJOINT”选项。</p>



8

拖动示教点操作步骤，完成。

**提示！**

有 2 种方式可退出示教模式：

- ① 将当前的“自动 Auto 模式”转至“手动模式”；
- ② 再点击 2 次“□”方框按钮，该按钮灯（绿色）灭。

**提示！**

如果在上述步骤 2 中，程序树的选中行不是图中的位置（第 1 行），而是选中了第 2 行 MJOINT 指令行，那么上述的后续操作，弹出的询问框，如下图：

（覆盖：表示更新当前指令的机器人姿态 target  
插入：会弹出上述步骤 7 中的“MJOINT”、“MLIN”询问框）

## 拖动示教轨迹

承接“拖动示教点”操作，以下介绍如何以拖动的方式示教一个机器人的运动轨迹。



拖动示教轨迹，操作步骤：

#	图示	说明
1		<p>① 选中程序树中“结束”行；</p> <p>② 机器人模式切换至“自动 Auto”+“上伺服”模式。</p>
2		<p>① 点击 2 次末端法兰上“□”方框按钮。</p> <p>（此时该按钮灯为绿色常亮）</p>
3		<p>① 点击 1 次末端法兰上“△”三角按钮。</p> <p>（此时该按钮灯为蓝色常亮）</p>
4		<p>① 轻轻扣住“使能键”，并且开始拖动机器人一小段轨迹；</p> <p>拖动完毕后，松开“使能键”。</p>



<p>5</p>		<p>① 再点击 1 次末端法兰上“△”三角按钮。</p> <p>（此时程序树中会自动插入一行 MPATH 指令,该指令即为拖动轨迹的再现指令）</p>
<p>6</p>		<p>拖动示教轨迹操作完成,关闭拖动示教模式。</p>

### 提示!


拖动示教产生的轨迹文件,可在“安装设置”选项卡中“通用/轨迹跟踪”页面中查看。

文件名	注释
20201121144951321...	
8888888888888888.CSV	



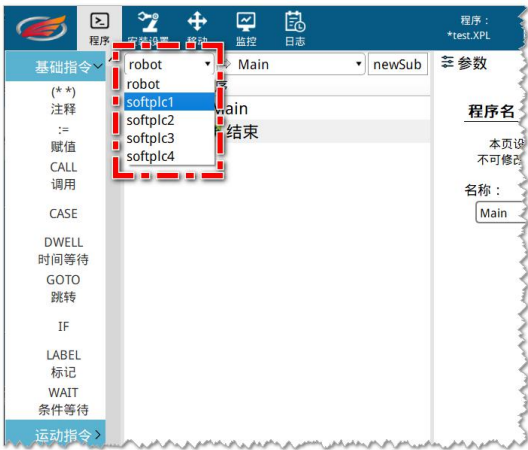

### 4.2.5 软 PLC

软 PLC 功能是内置于控制器的一个虚拟 PLC。可以使用 RPL 指令中的逻辑类指令进行编写运行于软 PLC 中的程序，但是不能使用机器人运动类指令。

	<p><b>提示！</b></p> <p>控制器系统中内置 4 个软 PLC 环境，可以分别加载 4 个软 PLC 程序文件到这 4 个环境中。</p>
---	--

### 新建软 PLC 程序

新建软 PLC 程序，操作步骤：

#	图示	说明
1		<p>① 点击下拉框，选出“softplc1”选项。</p> <p>（此时程序树由主程序 Main 环境，切换到软 PLC 环境）</p>
2		<p>① 按照正常创建程序的流程，即可创建软 PLC 程序，程序名为：“plc_1”。</p>



3		新建软 PLC 程序，完成。
---	--	----------------

	<p><b>提示！</b></p> <p>在软 PLC 编辑环境下，指令栏中“运动指令”、“运动相关”类，处于禁用状态，如下图：</p>
--	--

## 运行软 PLC 程序

以下介绍 2 种方式，运行软 PLC 程序：

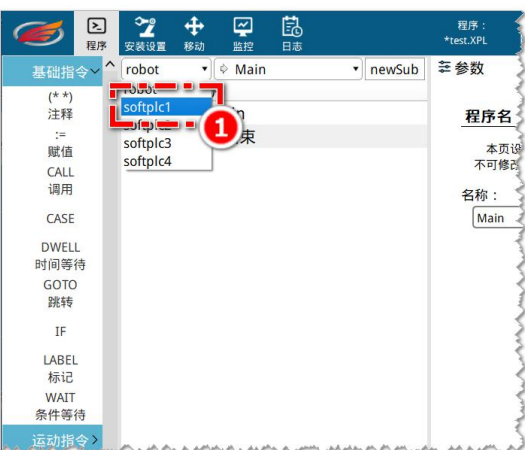
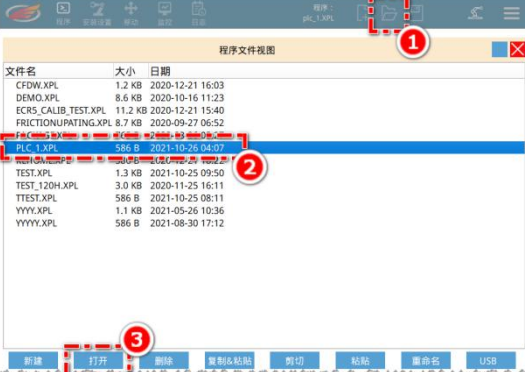
方式 1：手动启动软 PLC 程序；

方式 2：主程序中调用并执行软 PLC 程序；

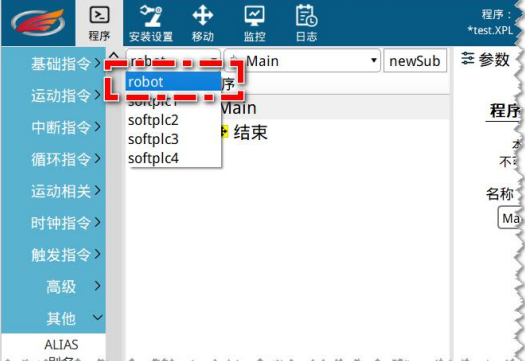
方式 1：手动启动软 PLC 程序，操作步骤：

#	图示	说明
---	----	----



1		<p>将软 PLC 程序加载到“softplc1”环境中。</p> <p>① 点击下拉框,切换到“softplc1”环境。</p>
2		<p>打开“plc_1”文件。</p>
3	<p>在当前软 PLC 环境下,以运行程序的相同操作(自动运行、单步调试,参看 4.2.3 节),即可运行当前加载的软 PLC 程序。</p>	

方式 2: 主程序调用启动, 操作步骤:

#	图示	说明
1		<p>① 将程序树环境,切换回“robot”。</p>



<p>2</p>		<p>① 插入“CALL”指令；</p> <p>② 选中插入到程序树中的CALL指令(“?? ? ; ! error 1”行)；</p> <p>③ 在右侧的“参数”标签页中，点击“编辑”按钮。</p>
<p>3</p>		<p>① 在候选列表中，选中“system.startPLC1()”项；</p> <p>② 点击“确认”按钮。</p>
<p>4</p>		<p>① 在“参数 0”项中，填入“RPL/plc_1.XPL”； (本例中新建的软 PLC 文件名) (“RPL/”是 PLC 程序所在文件夹)【操作查看步骤 5】</p> <p>② 在“参数 1”中，填入“1”；</p> <p>③ 点击“应用”按钮，完成设置。</p>
<p>5</p>		<p>步骤 4 中操作补充</p>



表 4-3 system.startPLC 指令说明

system.startPLC 指令格式	
<p>Call system.startPLC1(“string”，type)                      Call system.startPLC2(“string”，type)                      Call system.startPLC3(“string”，type)                      Call system.startPLC4(“string”，type)                      上述四条指令可分别对应指定启动 PLC1~PLC4 任务；</p>	
说明项	说明
参数个数	2
参数说明	<p>#1: string:                      PLC 程序的完整路径,可点击程序界面右上角按钮，在弹出界面中进行查看 PLC 程序存储路径，例：/RPL/PLC1.XPL。</p>
	<p>#2: type:                      PLC 运行方式，                      “0”表示主程序运行时 PLC 程序运行，主程序暂停时，PLC 暂停；                      “1”表示无论主程序是否运行或暂停，PLC 程序一直在后台运行。</p>



# 第 5 章 安装设置选项卡

安装设置选项卡只能以 Engineer（工程师）、Administrator（管理员）用户登录才可使用。Operator（操作员）用户无法进入安装设置界面。

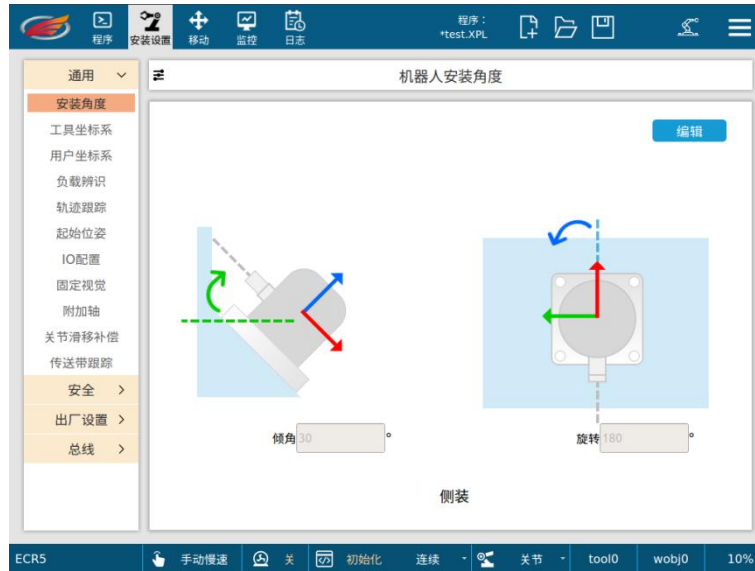


图 5-1 “安装设置”选项卡

该选项卡中包含涉及机器人正常运行的多项参数配置，所含功能页见表 5-1。

表 5-1 安装设置选项卡内容

通用	安装角度	安全	区域监控
	工具坐标系		位置监控
	用户坐标系		法兰设置
	负载辨识		碰撞检测
	轨迹跟踪		安全限制
	初始位姿		安全平面
	模块更新		身体风险
	IO 配置		安全传感器
	固定视觉		
	附加轴		
	关节滑移补偿		
	传送带跟踪		
	出厂设置		滑移基准点



	驱动器清零		MES
	DH 参数		EtherCAT
	关节限制		EthernetIP

## 5.1 通用类

通用类中包含安装角度、工具坐标系、用户坐标系、负载辨识、轨迹跟踪、初始位姿、IO 配置、固定视觉、附加轴、关节滑移补偿等功能设置。

### 5.1.1 安装角度

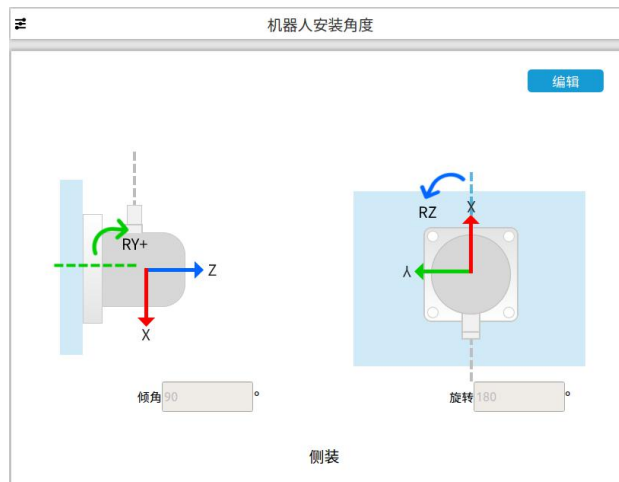



图 5-2 “安装角度”功能页


安装角度功能页可对机器人安装进行正装，倒装，侧装三种设置，进入安装角度界面，显示的安装角度是由控制器中保存的上次安装设置读出并显示。

安装角度，使用步骤：

#	图示	说明
1		<p>① 点击“编辑”按钮，进入编辑模式。</p>



2		<p>① 点击所需要安装姿态的对应按钮;</p> <p>② 点击“应用”按钮, 即保存设置;</p> <p>③ 点击“是”, 重启示教器和控制器设置生效。</p>
---	---	---

	<p><b>提示!</b></p> <p>若选择侧装按钮, 可调整安装的倾角和旋转角度, 其中倾角可调整范围为 <math>1^{\circ} \sim 179^{\circ}</math> (倾角度数改变实际只分锐角, 钝角和直角三种安装姿态, 并不会精确到具体的度数), 旋转角度可调整范围为 <math>0^{\circ} \sim 359^{\circ}</math>。</p>
--	---

### 5.1.2 工具坐标系

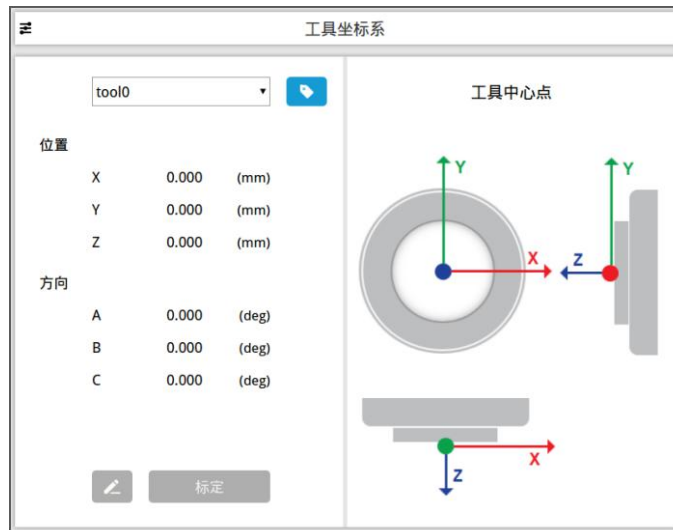



图 5-3 “工具坐标系” 功能页

工具坐标系功能页, 可对工具中心点 (Tool Center Point) 进行标定、修改以及激活操作。系统提供 10 个工具对象 (tool1~tool10) 可以由用户自定义。

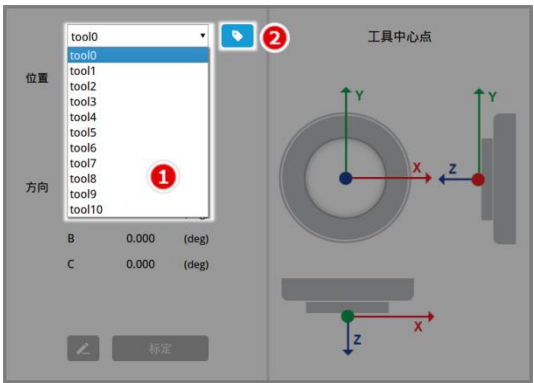


	<p><b>提示!</b></p> <p>tool0 的数据全部为‘0’并且不可修改、标定，表示末端法兰中心点，即不加载工具。</p>
---	---

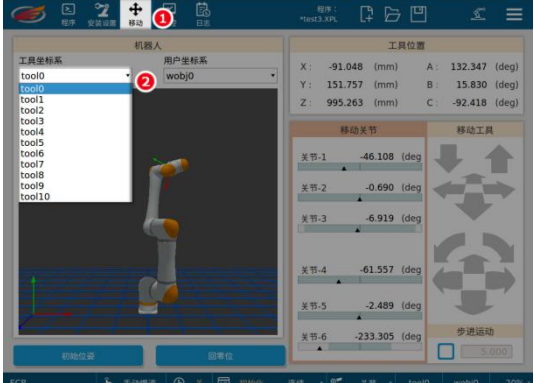
## 激活工具

2 种操作方法可激活 tool:

操作方法 1:

#	图示	说明
1		<p>① 点击“下拉框”选中需要激活的 tool 对象;</p> <p>② 点击“激活”按钮，即可激活选中项。</p>

操作方法 2:

#	图示	说明
1		<p>① 点击“移动”选项卡，进入移动界面;</p> <p>② 点击“工具坐标系”下拉框，选择 tool 对象，即可立即激活该工具。</p>

	<p><b>提示!</b></p> <p>可在状态栏中查看当前机器人激活的工具项，如下图</p>
---	--





## 标定工具

标定操作是：将工具执行端（工具末端）以多个不同的姿态校到“校准点”（如下图 5-4）、记录当时姿态，最后通过这些记录数据计算得出工具坐系数值。

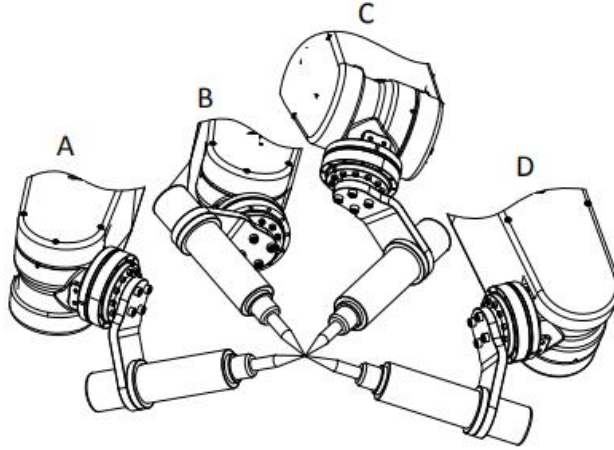


图 5-4 工具标定示意图

使用步骤：

#	图示	说明
1		<p>① 选择除 tool0 之外的任一 tool 项（比如 tool1）；</p> <p>② 点击“标定”按钮，进入标定操作界面。</p>
2		<p>① 点击“标定方式”下拉框，选择“TCP&amp;default”；</p> <p>② 点击“下一步”按钮。（这里以“TCP&amp;default”方法为例，其他标定方法说明，参见下表 5-2）</p>



3		<p>① 点击“示教第1点”，UI界面将跳转到“移动”选项卡。</p>
4		<p>① JOG 机器人,将工具的末端执行点,对准“校准点”(如图 5-4);</p> <p>② 点击“√”按钮,完成第1点示教(界面会自动跳转回“工具坐标系”界面)。</p>
5		<p>重复步骤 3、4,依次示教完 4 个点;</p> <p>① 点击“计算”按钮,得出 TCP 数据。</p>
6		<p>① 点击“应用”，保存计算结果。</p>



表 5-2 标定方法说明

标定方法	点数	说明
TCP & default	4	将工具末端从 4 个方向或姿态对准到空间中的“校准点”。
TCP & Z	5	前 4 个示教点同 TCP&default 方法，第 5 个点以‘机器人坐标系’方式，从 Z 轴方向相对“校准点”移动一段距离并示教该点。
TCP & Z,X	6	前 5 个点操作同 TCP&Z 方法，第 6 个点以‘机器人坐标系’方式，从 X 轴方向移动一段距离并示教该点。

## 编辑工具

可通过“直接编辑”修改 TCP 数值，使用步骤如下：

#	图示	说明
1		<p>① 选择除 tool0 之外的任一 tool;</p> <p>② 点击“编辑”按钮，进入编辑操作界面。</p>
2		<p>① 直接修改相关项数值;</p> <p>② 点击“应用”按钮，保存修改。</p>



### 5.1.3 用户坐标系

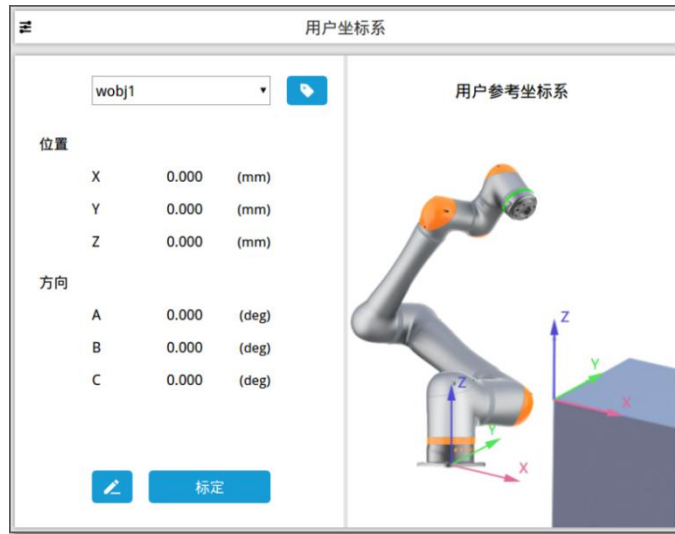



图 5-5 “用户坐标系”功能页

用户坐标系功能页，可对用户坐标系进行标定、修改以及激活操作。系统提供 5 个对象（wobj1~wobj5）可供用户自定义。

	<p><b>提示！</b></p> <p>wobj0 的数据全部为 ‘0’ 并且不可修改、标定，使用 wobj0 表示以机器人基座为原点的坐标系，即“基坐标系”。</p>
---	---

用户坐标系示意图如下图：

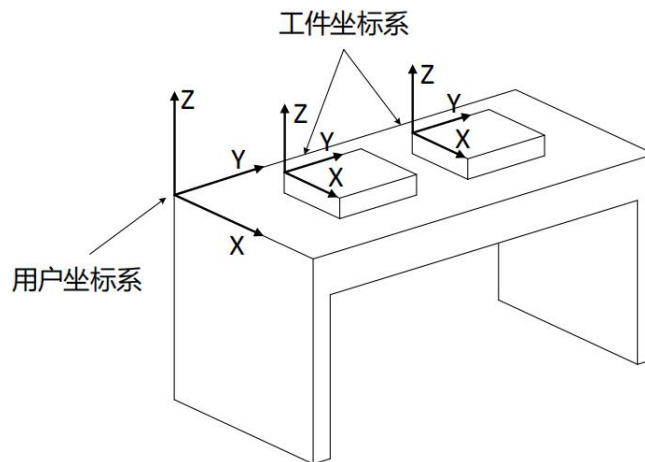


图 5-6 用户坐标系示意

用户坐标系页面中激活、标定、编辑等操作同 5.1.2 节，仅在示教点环节与工



具坐标系存在区别。

用户坐标系标定方法说明，见下表 5-3:

表 5-3 用户坐标系标定方法

标定方法	说明
有原点法	第一点：坐标系原点； 第二点：X 轴正方向轴上一点； 第三点：在 Y 轴正方向的 XY 平面内任意一点。
无原点法	第一点：X 轴上任意一点； 第二点：相对第一点，指向 X 轴正方向上一点； 第三点：Y 轴正方向上一点。

### 5.1.4 负载辨识

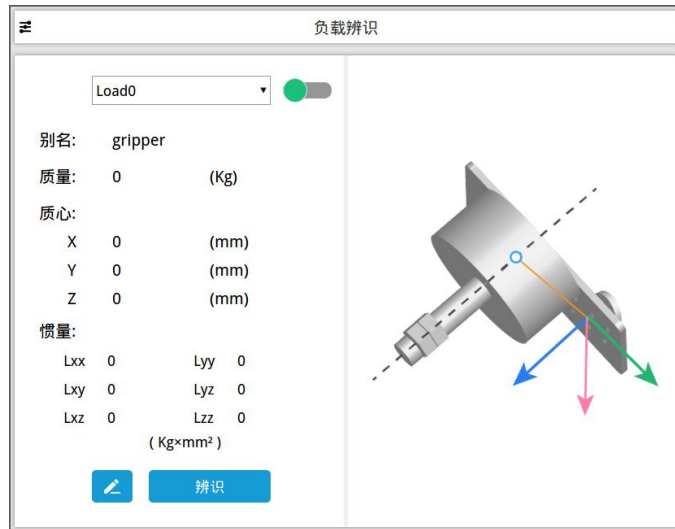


图 5-7 负载辨识

当机器人末端加装负载时（包括工具、工件），该负载会影响机器人的拖动示教功能、碰撞检测功能效果。负载辨识功能用于识别加载对象的质心、惯量等参数数值，该功能为拖动示教功能、碰撞检测功能准确性提供前置基础。

	<p><b>提示!</b></p> <p>负载：表示加载到机器人末端的任何物体。</p>
--	--

### 辨识负载

辨识负载，使用步骤：



#	图示	说明
1		<p>① 选择待辨识的负载项（以 Load0 为例）；</p> <p>② 点击“辨识”按钮，进入“辨识向导”界面。</p>
2		<p>“向导步骤一”页面说明：</p> <p>“操作”：分别点击“示教”（也可直接编辑改变机器人轴 3~6 的位置），去记录当前机器人的关节位置（其中上下限位置的各个轴的位置范围必须符合提示 1 的条件，否则会出现错误报警）；</p> <p>“轨迹路点数”：设置生成路点的数量；（点数越多，辨识过程越漫长，精度相对越高，推荐使用默认数值）。</p>



<p>3</p>		<p>“向导步骤二”：</p> <p>按照辨识向导步骤二中的提示，逐步操作：</p> <p>满足向导要求后，“开始辨识”按钮会激活，点击“开始辨识”（满足操作1~3，操作4未执行前，可将速度调到低速试运行机器人进行负载辨识，确认辨识轨迹是否安全，若安全，返回到操作4未执行前，将速度调为50%，满足所有操作，进行负载辨识）。</p>
<p>4</p>		<p>辨识进行中：</p> <p>注意：等待辨识完成的过程中，不要调节机器人运动速度，或按下示教器面板的“程序暂停”键，以免造成辨识误差或辨识终止。</p>
<p>5</p>		<p>辨识完成：</p> <p>点击“下一步”，跳转到辨识向导最后步骤。</p>



① 点击“完成”，保存辨识结果。

## 编辑负载

点击“编辑”按钮（如图 5-8 左图），之后直接修改当前负载项（如图 5-8 右图）。如果需要直接编辑负载参数，建议客户使用设备提供商所提供的 3D 数模图中的数据。

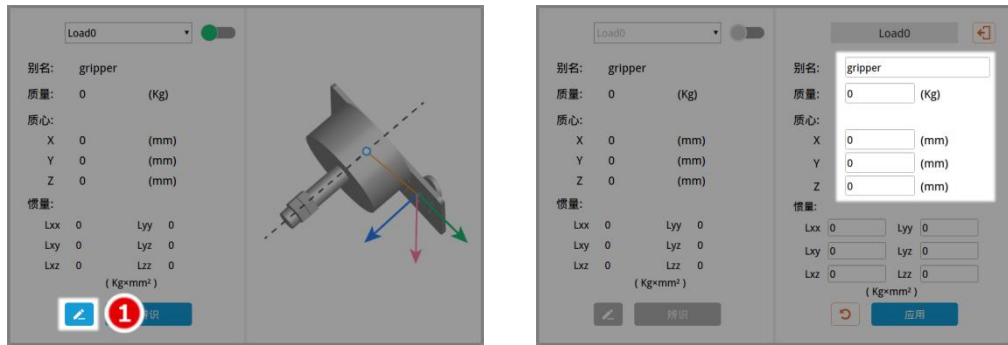


图 5-8 编辑负载

## 激活负载

选择待加载的负载项，点击“激活”按钮（如下图 5-9 左）。

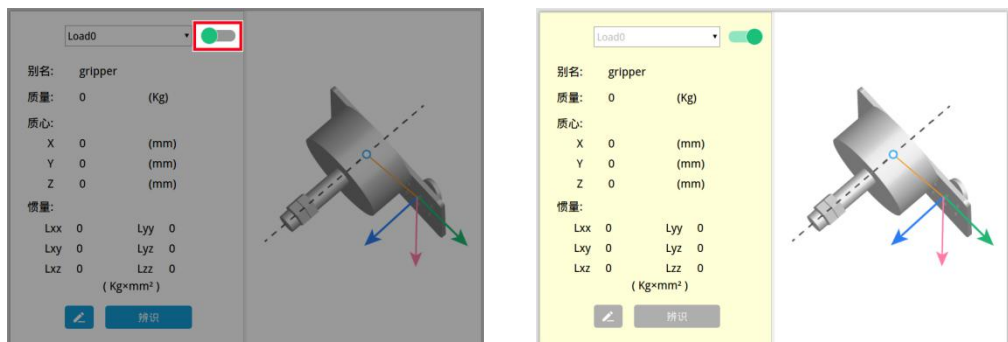


图 5-9 激活负载



## Module 函数及 RPL 程序用例

表 5-4 负载相关 module 指令

说明项	说明
函数名	collision.setPayLoadPar( )
函数功能	在程序中，选择以及激活负载项。
参数个数	2
参数说明	#1: isEnabledPayLoad (bool 类型，表示：激活/失活负载)
	#2: payLoadIndex (DINT 类型，表示：负载项索引，范围 0~9)

程序用例，操作步骤：

#	图示	说明
1		新建 RPL 程序，插入“CALL”指令。
2		<p>① 点击程序树中“???; ! error 1”处（即 CALL 指令）；</p> <p>② 点击“编辑”，设置该条指令具体参数。</p>



3		<p>① 在“候选子程序列表”中选择“collision.setPayloadPar()”；</p> <p>② 点击“确认”按钮，返回到 CALL 指令参数设置页。</p>
4		<p>① 将参数 0，参数 1 分别填入“true”、“0”； (注：true、0 表示：激活 0 号负载项)</p> <p>② 点击“应用”按钮，保存设置。</p>
5		<p>完成。</p>



### 5.1.5 轨迹跟踪

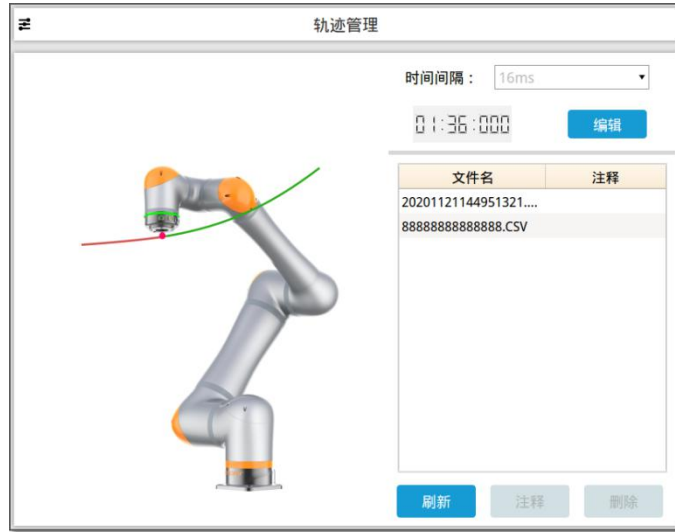



图 5-10 轨迹跟踪

轨迹跟踪功能页面：

- ① 录制轨迹的采样时间间隔；
- ② 管理已有轨迹；

	<p><b>提示！</b></p> <p>生成轨迹的方法参看 4.2.4 节中拖动示教轨迹相关内容。</p>
---	--

采样间隔使用步骤：

#	图示	说明
1		<p>① 点击“编辑”按钮，可激活下拉框。</p>



2		<p>① 选择需要的采样时间； (注：采样间隔越长，录制轨迹时间越久)</p>
3		<p>① 点击“应用”按钮，完成修改。</p>

轨迹文件管理：

#	图示	说明
1		<p>① 点击列表栏中的任一项，可对该项进行“注释”、“删除”操作。</p> <p>② 可点击“刷新”，“注释”，“删除”进行相应操作。</p>

选中多个文件操作：

#	图示	说明
---	----	----



1		<p>① 双击列表栏中的任一项，界面会显示复选框。</p>
2		<p>① 可勾选列表项，进行批量删除操作。 注：点击“取消”按钮，退出多选模式。</p>

### 5.1.6 初始位姿



图 5-11 初始位姿

初始位姿功能页用以设置并保存一个机器人姿态，结合“移动”界面中的“初始位姿”按钮（见下图 5-12），可实现让机器人‘自动’运动到该预设姿态，属于一种快捷操作。



图 5-12 移动界面

## 编辑初始位姿

使用步骤如下：

#	图示	说明
1		<p>① 点击“编辑位姿”按钮，界面跳转到“移动”选项卡页面。</p>
2		<p>① 运动机器人至目标姿态；</p> <p>② 点击“√确定”按钮，以保存该姿态，并且界面会自动跳转回“初始位姿”功能页。</p>



3	<p>关节-1 0.000 (deg) 关节-2 6.336 (deg) 关节-3 -34.099 (deg) 关节-4 -47.856 (deg) 关节-5 -89.263 (deg) 关节-6 0.000 (deg)</p> <p style="text-align: center;">编辑位姿</p>	完成。
---	--	-----

### 移至初始位姿

使用“初始位姿”，有2种方式。

方式1:

#	图示	说明
1		① 点击导航栏中“移动”选项卡。
2		① 机器人状态设置为：“手动模式”+“上伺服”； ② 保持按住“使能键”； ③ 持续按住屏幕中“初始位姿”按钮，即可使机器人向默认姿态运动。



3		<p>机器人运动就位后，会弹出提示，表示机器人已运动到初始位姿</p> <p>① 点击“OK”按钮，完成。</p>
---	--	---

方式 2:

#	图示	说明
1		<p>① 机器人状态设置为：“自动模式”+“上伺服”；</p> <p>② 持续按住屏幕中“初始位姿”按钮，使机器人运动。</p>
2		<p>机器人运动就位后，会弹出提示，表示机器人已运动到初始位姿</p> <p>① 点击“OK”按钮，完成。</p>

	<p><b>注意!</b></p> <p>使用该功能前需考虑机器人当前实际可移动区域，避免自动回位过程中造成碰撞。</p>
--	---



### 5.1.7 模块更新

控制器背板插入新的硬件扩展模块，系统通电开机后，示教器界面会出现“1497”和“988”报警，如下图：



图 5-13 1497、988 报警

正确处理该报警步骤如下：

#	图示	说明
1		<p>① 点击进入“安装设置”导航页；</p> <p>② 点击“模块更新”菜单，进入模块更新界面。</p>



2		<p>① 点击“更新”按钮；</p> <p>② 点击“是”，确认更新模块；</p> <p>③ 点击“OK”，等待模块更新完成。</p>
---	--	---

### 5.1.8 IO 配置

IO 配置功能页有以下 2 个功能：

对某个数字量 IO 端口，将其与某个系统功能进行绑定。

DI：当外界对该端口置入电平，会触发所绑定的系统功能；

DO：当该系统功能生效时，会将所绑定的数字量 IO 端口置高、低电平。

设置模拟量 IO 端口的模拟量类型：电流类型、电压类型。



图 5-14 IO 配置主界面



## 功能性数字 IO 配置



图 5-15 功能性数字 IO 配置

目前系统开放可供用户使用的功能性 IO 有 4 个大类：通用，区域、位置监控，附加轴，安全。

操作步骤：

#	图示	说明
1		① 点击“编辑”按钮，界面进入可编辑状态；
2		① 点击列表中的某一项功能，会弹出设置对话框；



<p>3</p>		<p>在弹出的设置框中进行设置，其中“有效电平”下拉框中，“0”表示“低电平”；“1”表示“高电平”。</p>
<p>4</p>		<p>“滤波时间”仅作用于数字量输入端口，用以过滤噪声电平信号。 通常设置为 0.1 秒。</p>
<p>5</p>		<p>① 点击“应用”按钮，完成设置。</p>



## 模拟 IO 设置（电控柜）



图 5-16 模拟 IO 设置（电控柜）

通过该界面对电控柜内部的 4 组输入端口、4 组输入端口进行电压/电流类型切换。

## 模拟 IO 设置（法兰端）



图 5-17 模拟 IO 设置（法兰端）

通过该界面对机器人末端法兰上的 2 组输入端口进行电压/电流类型切换。



### 提示！

末端法兰只有模拟量输入端口，没有模拟量输出端口。

### 5.1.9 固定视觉

视觉功能是指机器人与视觉系统通过 TCP/IP 协议进行通讯，视觉系统作为服




务器，机器人作为客户端，视觉系统将获取的基于视觉系统坐标下物体的位置信息转化成机器人坐标下的位置，从而实现机器人按指定轨迹运动。固定视觉是指相机安装在固定台架上，拍摄的物体在固定的工作台面上。



图 5-18 固定视觉

固定视觉是同时具备以下 3 个条件的一种视觉功能：

- (1)使用单目相机（2D 相机）；
- (2)相机安装在固定台架上；
- (3)拍摄对象（工件）在被相机拍照后直到被机器人抓取前，始终处于静止状态，即空间位置上保持不变。

	<p><b>提示！</b></p> <p>相机识别工件、像素标定等配置，需要联系相机厂商提供技术支持。</p>
---	---

## 相机通讯格式


在使用过程中，视觉系统（相机）需要将图像处理后的工件信息通过机器人提供的固定通讯格式传输给机器人，机器人根据接收到的数据进行取放动作。因此，机器人通讯格式主要包括三个部分：

表 5-5 通讯格式组成部分

名称	组成	说明
----	----	----



物体坐标参数	X、Y、A	物体在相机视野范围内的位置及旋转角度，该位置为相机/像素坐标系（单位 mm 或 px）下的位置。
物体属性参数	ATTR	根据物体不同属性（例如：形状、颜色等）给出物体的对应属性值，以数字：0、1、2、3……来表示。
物体 ID 编码	ID	为了方便管理给每一个物体制定的唯一编码。

	<p><b>提示！</b></p> <p>属性参数与 ID 编码用户可根据实际情况选择是否使用以及具体的使用方式，如不需要应用，在相机通讯格式设置时将其默认为 0 即可。</p>
---	---

具体通讯数据格式如下：


Image\r\n

[……]\r\n

Done\r\n

上述格式中：

“Image”	表示数据头，即一组图像数据下发开始的标志。	
“Done”	表示数据尾，即一组图像数据下发完成（注意“Image”和“Done”区分大小写）。	
“\r\n”	回车换行符。	
“[……]”	表示相机下发的一个物体的数据，其中包含了物体坐标参数 XYA，物体属性参数 ATTR 和物体 ID，每个物体的数据均以 “[” 开始，以 “]” 结束。	
	1	X:--;Y:--;A:--;ATTR:--;ID:--
	2	--;--;--;--;--; （分号分隔）
	3	--,--,--,--, （逗号分隔）

	<p><b>提示！</b></p> <p>固定视觉通讯一次，只能传输一组数据。</p> <p>当相机没有拍到物体或者识别物体失败时发送字符串“Error”。</p>
---	--



## 相机设置

#	图示	说明
1		<p>① 点击“设置”按钮，打开相机设置界面。</p>
2		<p>① 设置相机参数（如相机品牌，拍照间隔，触发方式等）；</p> <p>② 点击“应用”按钮保存此次相机设置。</p>

## 相机连接

#	图示	说明
1		<p>① 固定视觉功能未开启时，显示“已关闭”；</p> <p>② 点击“启用”按钮打开固定视觉功能。</p>



2		<p>① 固定视觉启用状态下，相机未连接，显示红色未连接信息；</p> <p>② 点击“连接”按钮，连接相机。</p>
---	--	---

## 手眼标定

#	图示	说明
1		<p>① 在“相机设置”页中，确认勾选“使用手眼标定”。</p>
2		<p>① 确保固定视觉功能开启，并且成功连接相机；</p> <p>② 点击“手眼标定”按钮。</p>



<p>3</p>		<p>弹出“是否使用像素分辨率标定”询问框。</p> <p>注意：选择“否”，请直接跳转到步骤6</p>
<p>4</p>		<p>① 点击“拍照”按钮获取工件的像素值；</p> <p>② 点击“示教”按钮获取工件在机器人坐标系下的值；</p> <p>③ 点击“下一步”按钮进行下一个点的标定工作（共2个点）。</p>
<p>5</p>		<p>① 标定全部2个点后，点击“计算”按钮。</p>
<p>6</p>		<p>上述步骤4、5为“标定像素分辨率”的操作。以下为“手眼标定”内容</p> <p>① 通过下拉框选择标定点点个数；</p> <p>② 点击“下一步”按钮进入具体示教界面。</p>



<p>7</p>		<p>① 点击“拍照”按钮获取相机坐标系下工件位置；</p> <p>② 点击“示教”按钮获取机器人坐标系下工件位置；</p> <p>③ 完成所有点的标定工作后，点击“计算”按钮。</p>
<p>8</p>		<p>① 点击“确定”，保存标定结果。</p>

### 工件详细信息

#	图示	说明
---	----	----



1		点击“详细信息”按钮；
2		进入工件详细信息界面，查看固定视觉坐标系，像素分辨率，工件信息。

## Module 函数及 RPL 程序用例

表 5-6 固定视觉相关 module 函数

说明项	说明
指令函数名	vision._Init_()
函数功能	该命令视觉相关功能的初始化，在正常使用中不需要调用，该函数会在程序开始自动运行。
形参个数	0
形参说明	无
返回参数	无
说明项	说明
指令函数名	vision.getData()
函数功能	调用该命令，触发相机拍照动作并返回相应数据。
形参个数	0
形参说明	无
返回参数	Bool 类型：true 表示获取成功；false 表示获取失败。
说明项	说明



指令函数名	vision.setTrigCmd()
函数功能	相机在指令触发的模式下，该命令可设置相机触发的指令。
形参个数	1
形参说明	#1: p (int 类型) 数值与相机端配置一致，从而才可触发相机的拍照。(同“相机设置”页面中的“指令触发”设置项)。
返回参数	无
说明项	说明
指令函数名	vision.trigCam()
函数功能	相机在指令触发的模式下，该命令可触发拍照。
形参个数	0
形参说明	无
返回参数	无

表 5-7 固定视觉相关 module 变量

变量名	类型	说明
vision.x	lreal	工件位置: X 方向坐标
vision.y	lreal	工件位置: Y 方向坐标
vision.z	lreal	工件位置: Z 方向坐标
vision.a	lreal	工件姿态: 绕 Z 轴角度
vision.b	lreal	工件姿态: 绕 Y 轴角度
vision.c	lreal	工件姿态: 绕 X 轴角度
vision.attr	int	工件属性
vision.id	int	工件 ID

```

1 LABEL a :
2 MJOINT (*, v500, fine, tool1) ;
3 vision.setTrigCmd(1) ;
4 hasObj := vision.getData() ;
5 IF hasObj THEN
6     point1pick := POINTC(vision.x, vision.y, vision.z + 5, -180 - vision.a, 180, 0) ;
7     point1 := POINTC(vision.x, vision.y, vision.z + 50, -180 - vision.a, 180, 0) ;
8     MJOINT (point1, v500, fine, tool1) ;
9     MJOINT (point1pick, v500, fine, tool1) ;
10    DWELL (5) ;
11    MJOINT (point1, v500, fine, tool1) ;
12 END_IF ;
13 GOTO a ;

```



图 5-19 测试程序代码

- Line1:循环开始;
- Line2:机器人运动到姿态 1;
- Line3:设置相机触发指令为 1; (当相机设置为 IO 触发时, 不需要该步骤)
- Line4:触发相机拍照, 当成功获取数据时变量 hasObj=true;
- Line5:如果成功获取相机数据, 则执行 Line6-11;
- Line6:定义机器人姿态 2 (抓取点);
- Line7:定义机器人姿态 3 (抓取点上方位置);
- Line8:机器人运动到姿态 3;
- Line9:机器人向下运动到姿态 2;
- Line10:等待抓取工件;
- Line11:机器人向上运动到姿态 3;
- Line13:循环结束。

	<p><b>注意!</b></p> <p>因为在机器人端做手眼标定, 所以 vision.x, vision.y, vision.z 三个工件位置数据结果可以直接应用, vision.a, vision.b, vision.c, 需要根据现场实际情况进行数值上的补偿。</p>
--	--


### 5.1.10 附加轴



图 5-20 附加轴配置



附加轴功能主要是在标准的机器人基础上，增加 1 到 4 个附加轴，来配合机器人完成客户现场复杂工作的功能。

	<p><b>提示！</b></p> <p>协作机器人控制系统目前最多支持 4 个附加轴。</p>
---	--

## 配置附加轴系统

附加轴轴数配置，使用步骤：

#	图示	说明
1		<p>① 点击“安装设置”模块下的“通用”，再点击“附加轴”进入附加轴配置界面。</p>
2		<p>① 点击附加轴界面“系统配置”进入轴数配置界面。</p>



<p>3</p>		<p>① 配置附加轴轴数。选择附加轴数后，左边页面会对应出现需要配置的附加轴个数。</p>
<p>4</p>		<p>① 配置是否存在龙门轴。若存在龙门轴需配置主从轴最大误差，范围：0.01~100mm。</p>
<p>5</p>		<p>① 配置插补轴数量。插补轴数量不同，对应的系统模型存在差别，系统模型对应出现当前轴数可配置的类型。插补轴数量不超过附加轴总数，且不超过数量3。</p>



6		<p>① 配置插补轴的系统模型。不同插补轴数，对应不同系统模型。（附加轴轴数配置完成后，再进行轴参数配置）</p>
---	--	---

	<p><b>提示！</b></p> <p>龙门轴在软件中是指两个电机同步运动，控制龙门架移动运行。使用龙门轴需要两个电机型号参数应保持相同。在本系统中龙门轴的设置及运动使用等同一个附加轴，且龙门轴必须为所有附加轴中的最后一个轴。</p> <p>如果存在龙门轴，注意两个电机的运动方向：</p> <ol style="list-style-type: none"> <li>1. 主从轴电机相向运动即为两个电机转动方向反一般电机如果安装在导轨相对两侧，或者相对放置选择此选项。</li> <li>2. 主从轴电机同向运动即为电机转动方向相同，一般电机安装在导轨同一边，或者同向放置选择此选项。</li> </ol> <p>附加轴作为插补轴，各个轴的运动方向必须和机器人在机器人坐标系下的运动方向相同。</p>
--	---

## 轴参数设置

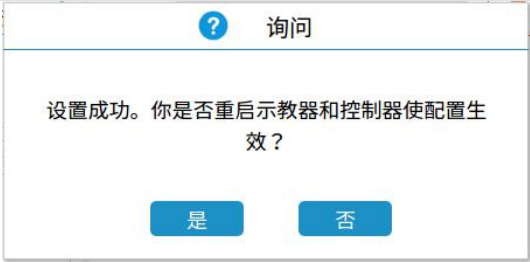
附加轴参数设置，使用步骤：

#	图示	说明
1		<p>① 点击“设置”按钮进入附加轴设置限制参数界面。</p>



<p>2</p>		<p>① 在此界面进行转换参数设置；</p> <p>② 除了基础参数外，非插补轴且异步轴功能开关打开后，在底部需要配置异步轴参数；</p> <p>③ 设置过程中不想保存设置的参数，可点击“返回”按钮放弃修改。</p>
<p>3</p>	  	<p>① 设置完成后，点击“退出”按钮，返回到系统配置界面。</p> <p>② 点击“应用”，保存配置信息。</p>



4		<p>根据询问框“重启控制器与示教器”，点击“是”。</p> <p>（等待重启完毕，即可使用附加轴）</p>
---	---	--

附加轴参数说明：

参数项	说明
负限位/正限位	可根据实际情况填写附加轴的机械限位值。
最大速度	机械机构最大速度值，可根据选型的电机最大速度来确认该值，一般情况下， 最大速度 ≤ (电机最大转速 * 位置转换) / (60 * 减速比)。
最大加/减速度	一般可按照最大速度参数的 2-5 倍设置。
最大加加速度	一般可按照最大 20-100 倍最大加速度参数设置。
异步轴速度	附加轴作为异步轴时的运行速度，值是最大速度的百分比，可输入 0-100（必须大于 0）。
异步轴功能	选择异步轴开关，开表示附加轴作为同步轴或异步轴使用，两者可以在程序中切换使用。关表示附加轴只能作为同步轴使用。
异步轴运动方式	异步轴运动方式分为单步和连续。
	<p>单步模式：接收到对应的单步信号，机器人运行设置的角度，正在运动的轴，不再接收此类信号。</p> <p>连续模式：单个轴分为正负两个方向运动，接收到信号，附加轴往信号对应的方向运动，取消信号，附加轴立即停止运动。</p> <p>（注：详细信号配置见 IO 配置）。</p>
路径规划方式	分为：正常路径、最快路径。
	<p>正常路径：附加轴在设置的正负限位中间运动，设置多少度，运行到多少度。</p> <p>最快路径：设置最快路径，需要将附加轴负限位和正限位设置为 0° 和 360°。当为异步轴的时候，机器人运行到 360° 后，轴位置又从 0° 开始。当为同步轴时候，机器人会以最快的路径运行到目标位置，比如</p>



	机器人当前位置在 90°，运行到 300°，机器人会以 90° ~0° ~300° 方式运行，如果运行到 250°，会以 90° ~180° ~250° 方式运行。
编码器分辨率	电机的编码器分辨率。（比如 17 位的编码器分辨率就是 131072）
减速比	减速机输出端转一圈，电机需要转多少圈。如减速机转 1 圈，电机转 50 圈，减速比为 50。此数值可以为小数，但是不能为负数。
方向	改变减速机输出转动方向，1 表示正方向，-1 表示负方向。
位置转换	表示减速机输出端转一圈，机械结构运行的行程，比如减速机转 1 圈，机械结构转 360°，则位置转换的值为 360。
步进角度 1 步进角度 2	表示附加轴作为异步轴时，接收到一次信号后，运行的行程。运动分正负方向运动，数值为负数表示反方向运动。 (注：附加轴步进信号设置可以在 IO 配置中完成。详细设置方法见 IO 配置)

## JOG 附加轴

在附加轴配置成功后，示教器面板上的控制键可以控制附加轴的转动。

同步非插补轴指令，操作步骤：

#	图示	说明
1		① 进入“移动”选项卡界面。
2		① 点击一次示教器面板“2nd”键； 示教器界面会有 2 处相应变化。



3		<p>此时，在“自动模式+上伺服”状态下，按下控制键①的“-/+”按键，即可控制附加轴转动。</p>

### 附加轴指令

同步非插补轴指令，操作步骤：

#	图示	说明
1		<p>① 在附加轴参数配置完成后，附加轴功能已经开启，通过特定的 RPL 语句完成对附加轴运动的控制。</p> <p>（注：j1-j6 代表机器人轴，ea1-ea6 代表附加轴）</p>
2		<p>① 附加轴的类型属于同步非插补轴，不参与机器人运动轨迹的插补，在编程时，建议将圆弧过渡参数（zone）设置为 fine（即运动到目标位置后再执行下一指令）。</p> <p>② 同步轴关节运行指令为 MJOINT (EPOINTJ)，</p>



		<p>直线运行指令为 MLIN(E POINTC)。</p> <p>(注：同步轴关节运行指令不可使用 MJOINT (EPOINTC)，否则将发生报错。)</p>
--	--	---

异步轴指令

异步轴的运行主要分为两个模式，自动模式下编程运动和手动模式下 IO 信号控制运动。

异步轴指令，操作步骤：

#	图示	说明
1		<p>① 在相关程序文件中添加“CALL”指令；</p> <p>② 选中“???; ! error 1”并编辑；</p> <p>③ 在“候选子程序列表”里面选择相关附加轴函数。</p>
2		<p>① <code>auxj.AuxMoveStep (AuxNum, Step)</code> 指令中，AuxNum 参数是附加轴序号，参数值为 1-4，表示选择运动的附加轴，Step 参数是附加轴 APP 中设置的步进角度值，参数为 1 或者 2，表示一次完成步进角度为步进角度 1 或者步进角度 2。</p> <p>例：<code>auxj.AuxMoveStep(1,2)</code> 表示附加轴 1，一次运行完成步进角度 2 表示的角度。</p>



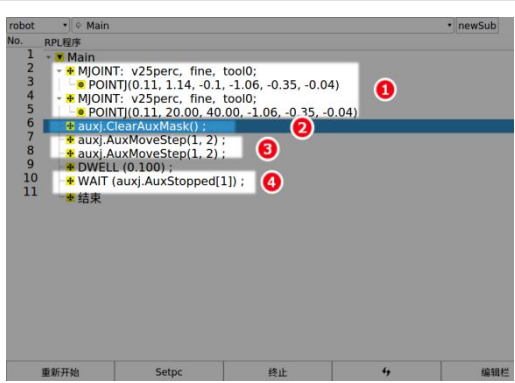
		<p>② 对同一个轴的连续两次脉冲信号，第二次将被屏蔽（①），为保证异步轴运动已经完成，增加等待信号（②）。</p> <p>③ 为确保异步轴运动完成，在发送异步轴运动脉冲信号后紧接着等待信号，如果异步轴未运动完成则 WAIT 语句一直运行等待（③）。增加 DWELL 语句，0.1 秒的缓冲时间确保异步轴开始运动后再执行 WAIT 语句，否则异步轴还未开始运动，则 WAIT 语句执行结束。</p>
3		<p>① 在手压信号存在的情况下，通过给特定的输入口脉冲信号，来控制选定的异步轴按照设定的步进角度进行运行。</p> <p>② 在异步轴运行期间，屏蔽所有的控制异步轴运动的脉冲信号。</p> <p>（注：IO 配置方法见 IO 设置功能。）</p>

### 同步异步在线切换

同步异步指令在线切换，操作步骤：

#	图示	说明
---	----	----



1	 <pre>robot Main newSub RPL程序 1 Main 2 MJOINT: v25perc, fine, tool0; 3 POINT[0.11, 1.14, -0.1, -1.06, -0.35, -0.04] 4 MJOINT: v25perc, fine, tool0; 5 POINT[0.11, 20.00, 40.00, -1.06, -0.35, -0.04] 6 auxj.ClearAuxMask(); 7 auxj.AuxMoveStep(1, 2); 8 auxj.AuxMoveStep(1, 2); 9 DWELL(0.100); 10 WAIT (auxj.AuxStopped[1]); 11 结束</pre>	<ul style="list-style-type: none"><li>① 同步轴运动指令;</li><li>② 清除同步轴标志位;</li><li>③ 异步轴运行信号;</li><li>④ 确保异步轴运行完成, 增加等待信号。</li></ul>
---	--	--

同步和异步在线切换说明:

- (1)同步轴运行期间, 一切异步轴请求将不被响应;
- (2)异步轴运行期间, 如果有同步非插补轴请求, 将中断异步轴运行指令。

在同步插补轴切换成异步轴时:

- (1)如果圆滑过渡指令为 fine (即没有圆滑过渡) 或者后面紧跟着的指令为 MJOINT(POINTJ/POINTC), 则不需要执行 auxj.ClearAuxMask()指令;
- (2)如果圆弧过渡指令为 zone 类型 (z50 等等) 且其后跟随异步轴运动信号: auxj.AuxMoveStep()指令, 则需要在同步轴运动结束后执行标志位清除指令: auxj.ClearAuxMask()。

在异步轴切换成同步插补轴时,为保证异步轴运行已经结束, 需执行指令 WAIT (auxj.AuxStoped[auxNum])指令让程序一直在此行等待, 直至指定异步轴运动完成继续执行下一行指令。否则, 异步轴的运行有可能被打断, 其中 auxNum 参数表示附加轴序号, 可设置为 1~4, 表示附加轴 1 到附加轴 4。

例如: 当附加轴 1 在运动过程中, auxj.AuxStoped[1]的值为 false, WAIT (auxj.AuxStoped[1])使得程序一直这一行等待, 当附加轴 1 运动结束后, auxj.AuxStoped[1]的值为 true, WAIT (auxj.AuxStoped[1])指令执行完成, 继续执行下一行指令。




### 5.1.11 关节滑移补偿




图 5-21 关节滑移补偿

关节滑移补偿：当机器人关节因碰撞致使编码器值与关节实际角度值产生误差，可通过该功能页进行补偿。关节滑移补偿的前提是：必须已设置了滑移基准点（滑移基准点的设置，参看 5.3.1 节）。

	<p><b>提示！</b></p> <p>偏移度数必须在 5 度以内才能补偿。</p>
---	---

关节滑移补偿操作步骤：

#	图示	说明
1		<p>① 手动模式+上伺服，扣住“使能键”；</p> <p>② 长按“移至此位姿”按钮，直至弹出提示框。</p>



<p>2</p>		<p>① 点击“OK”按钮；</p> <p>② 点击“计算”按钮计算补偿值。</p>
<p>3</p>		<p>① 计算成功后，下伺服；</p> <p>② 点击“补偿”按钮，完成关节滑移补偿。</p>

### 5.1.12 传送带跟踪

跟踪（Tracking）是指机器人工具末端 TCP（Tool Center Point）跟随一个运动的物体。目前该功能仅支持结合 2D 相机实现对传送带上面的工件进行跟踪。一个典型的例子是跟随传送带上的一个物体。该功能通过相机拍照，获取传送带上物体的位姿（X、Y、A 等值），机器人末端跟随该物体运动，并抓取该物体。



图 5-22 传送带跟踪



## 设置 2D 相机参数

操作步骤:

#	图示	说明
1		<p>点击“安装设置”→“传送带跟踪”菜单→“2D视觉跟踪”；</p>
2		<p>点击“设置”按钮，打开相机参数设置面板；</p>

相机参数说明:



图 5-23 相机设置界面



■ 连接设置：

(1) 相机品牌：

目前内部匹配的有 2 款：康耐视相机，麦克玛视相机。  
如果选用的非上述 2 款，则选择“通用相机”选项。

(2) 相机登录账号：

此项设置，只适用于“康耐视相机”，其他相机无需进行设置。

(3) 登录密码：

此项设置，只适用于“康耐视相机”，其他相机无需进行设置。

(4) 相机 IP 地址：

	<p><b>提示！</b> IP 地址应设置为相机算法服务器的 IP 地址，而非相机的 IP 地址。</p>
--	--

(5) 相机端口：

	<p><b>提示！</b> 端口号应设置为：相机算法服务器监听外部连接的端口号。</p>
--	--

■ 自动连接相机

(1) “自动连接”复选框：

勾选该项，机器人断电重启后自动与 2D 相机尝试连接，无需人工介入。

(2) 连接超时：

该参数，通常设置为 500 毫秒，即可。

	<p><b>提示！</b> 调试设备时不要勾选“自动连接”，待设备转为生产阶段再勾选该复选框。</p>
--	---



## ■ 数据参数

### (1) 拍照时间间隔:

该参数为触发相机拍照的间隔时间。

传送带速度越快, 该项数值应设置越低, 但不可低于相机一帧图片的处理周期。



#### 提示!

相机图片处理周期 = 拍照时间 + 处理图片时间 + 识别目标对象时间 + 向机器人传输运算数据时间 (即, 目标坐标值)。

### (2) 数据格式:

目前仅可选取“1”。同时需要 2D 相机提供商, 将其对外传输的数据信息, 设置为选项“1”所对应的数据格式。

数据格式为: [X:--;Y:--;A:--;Attr:--;ID:--;]

其中, “--”表示填入实际数值, “;”分号隔断后续数据项。

### (3) 相机数据获取指令:

该参数项, 只适用于“康耐视相机”。其他相机类型, 无需此项设置。

## ■ 拍照触发设置

机器人来触发相机的拍照, 目前支持 2 种方式, 分别是: IO 触发、指令触发。

**IO 触发:** 需要将相机的 IO 端口 (IO 输入端口) 与机器人控电柜中的 IO 端口 (IO 输出端口) 连接, 同时相机设置软件中触发方式也要设置为 IO 触发方式。

**指令触发:** 无需实施额外布线, 可直接通过机器人与相机服务器之间的网线, 发送指令触发相机拍照。但是需要在相机设置软件中进行设置, 来识别和响应机器人端发出的触发指令。



#### 提示!

“指令”: 即, 一个字符串, 机器人与相机双方需要协调一致。比如, 双方都协定以字符串“123”作为触发指令。

## ■ 以下组图为例设置:

该套设置使用“麦克玛视相机”结合“IO 触发”方式。



图 5-24 相机设置参考

## 测试编码器模块

操作步骤:

#	图示	说明
1		① 点击“开启”按钮
2		② 点击“连接”按钮



3		<p>相机成功连接后，操作界面如下图所示，同时，编码器计数开始自动刷新。</p> <p>图中左数值为：-2022 图中右数值为：-5718</p>
---	--	---

	<p><b>提示！</b></p> <p>上图中编码器计数有实时数据，前提是编码器模块已正确安装至机器人控制器，并且传送带处于开启状态。</p>
--	--

### 计算相机像素分辨率

注意：后续操作需要先满足以下 2 点：

- (1) 必须在相机连接成功后并且编码器计数可正确刷新数值后，才可进行。
- (2) 再次确认当前的工具坐标系，是否已激活为实际使用的工具坐标系。

操作步骤：

#	图示	说明
1		停止传送带运行；
2		<p>① 点击“传送带标定”</p> <p>② “是否使用像素分辨率标定？”选择“是”；</p> <p>以打开“传送带标定 / 像素分辨率标定”面板</p>



	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">&lt;像素&gt;</th> <th colspan="2">&lt;机器人&gt;</th> </tr> </thead> <tbody> <tr><td>X</td><td>0.000 (像素)</td><td>X</td><td>0.000 (毫米)</td></tr> <tr><td>Y</td><td>0.000 (像素)</td><td>Y</td><td>0.000 (毫米)</td></tr> <tr><td>Z</td><td>0.000 (像素)</td><td>Z</td><td>0.000 (毫米)</td></tr> <tr><td>A</td><td>0.000 (度)</td><td>A</td><td>0.000 (度)</td></tr> <tr><td>B</td><td>0.000 (度)</td><td>B</td><td>0.000 (度)</td></tr> <tr><td>C</td><td>0.000 (度)</td><td>C</td><td>0.000 (度)</td></tr> </tbody> </table>	<像素>		<机器人>		X	0.000 (像素)	X	0.000 (毫米)	Y	0.000 (像素)	Y	0.000 (毫米)	Z	0.000 (像素)	Z	0.000 (毫米)	A	0.000 (度)	A	0.000 (度)	B	0.000 (度)	B	0.000 (度)	C	0.000 (度)	C	0.000 (度)	
<像素>		<机器人>																												
X	0.000 (像素)	X	0.000 (毫米)																											
Y	0.000 (像素)	Y	0.000 (毫米)																											
Z	0.000 (像素)	Z	0.000 (毫米)																											
A	0.000 (度)	A	0.000 (度)																											
B	0.000 (度)	B	0.000 (度)																											
C	0.000 (度)	C	0.000 (度)																											
<p>3</p>		<p>将目标工件放置相机视野中的一角</p>																												
<p>4</p>		<p>点击“拍照”按钮，触发相机拍照从而得到目标工件的数据；</p>																												



<p>5</p>		<p>JOG 机器人直至末端工具的末端垂直触碰目标工件的抓取点；</p>
<p>6</p>		<p>① 点击“示教”按钮 ② “下一步”按钮；</p>
<p>7</p>		<p>JOG 机器人移出相机视野，之后将目标工件摆放至相机视野的对角端，重复上述步骤 3~步骤 6，标定第二个点；</p>
<p>8</p>		<p>① 完成第二个点的示教，点击“计算” ② 点击“OK”，得出“像素分辨率”数值；</p>



9		<p>点击“下一步”，进入下一节的“标定传送带坐标系”环节。</p>
---	--	------------------------------------

	<p><b>提示!</b></p> <p>现场情况不同，计算得出的结果与图中的数值有所不同。</p>
--	--

## 标定传送带坐标系

#	图示	说明
1		<p><b>确认传动带方向:</b></p> <p>开启传送带，调整传送带的运行方向为现场所需的方向。根据“编码器计数”数值变化选择传送带的“正向 / 反向”。</p> <p>比如：编码器计数的数值为递减，则在软件界面中选择“反向”，反之亦然。</p>
2		<p><b>示教第 1 点:</b></p> <p>将目标工件放置在相机视野内靠近传送带传输方向的上游端①，点击“示教”按钮②，再点击“下一步”按钮③;</p>



3		<p>示教第 2 点：</p> <p>开启传送带将目标工件输送到第 2 个点位，但不要超出相机的视野，暂停传送带，重复上一步的操作；</p>
4		<p>示教第 3 点：</p> <p>开启传送带，将目标工件输送到该点位置，暂停传送带。</p> <p>JOG 机器人，使末端工具垂直触碰目标工件的抓取点① → 点击“示教”按钮② → 点击“下一步”按钮③；</p>



5		<p>示教第 4、5 点：</p> <p>操作同上一步骤，启/停传送带，之后 JOG 机器人使末端工具尖端触碰到抓取点。</p> <p>第 4 点为跟踪中间点取样点；</p> <p>第 5 点为跟踪放弃点，即：目标工件被传送带输送超出第 5 点位之前，机器人尚未完成对该工件的跟踪与抓取，则放弃此次目标工件</p>
6		<p>示教第 6 点：</p> <p>第 6 点不需要传送带来输送，而是直接将工件手动搬运到指定位置。</p> <p>示教第 6 点的目的在于标定出传送带坐标系的‘Y+’方向，因此，将工件从第 1 点~第 5 点的移动轨迹视作‘X+’方向，采用右手坐标系，则‘Y+’方向为该条轨迹线的左侧；</p>



7		<p>将工件手动移动到 ‘X+’ 前进方向的左侧</p>
8		<p>JOG 机器人，将末端工具尖端垂直触碰目标工件抓取点，点击“示教”按钮① → 点击“计算”按钮②</p>
9		<p>点击“完成”，完成传送带坐标系的标定。</p>

## Module 指令、RPL 程序用例

### ■ Module 内置指令、内置变量

表 5-8 Module 内置指令、内置变量说明

指令	作用
track.init( )	初始化传送带跟踪（2D 视觉）状态
track.getWobj( )	获取跟踪队列中的工件数据，同时将 wobj_cvy（可视为一种用户坐标系）建立在该工件




	上
track.setTrigCmd(int p)	
track.dropObj( )	删除跟踪队列中首工件的数据，即：丢弃首工件
track.dropAllObj( )	删除跟踪队列中所有工件的数据
变量	作用
wobj_cvy	建立在跟踪工件上的坐标系，即：跟踪工件坐标系
wobj_cvy_fixed	传送带坐标系
ivt_objid	工件 id

■ RPL 程序示例

传送带跟踪（2D 视觉）模块，track.getWobj()指令将用户坐标系 wobj\_cvy 直接建立在被传送带输送的工件上，即：wobj\_cvy 坐标系的原点位于该工件上。从而使用 wobj\_cvy 可实现跟踪和抓取工件。

本示例中除了主程序 Main，还存在一个子程序 skipmove:

- (1)Main 程序：以实现对传送带上工件实现跟踪和抓取的工序流；
- (2)skipmove 子程序：当机器人抓取失败后的一些善后处理工序；

	<p><b>提示！</b></p> <p>机器人在对工件跟踪与抓取时会由于工件摆放密度、传送带传输速度过快等因素，造成机器人对工件的跟踪和抓取的失败。</p>
---	---

程序中设计了 5 个位置点，用户可根据实际的现场情况示教调整这 5 点数值。

表 5-9 位置点信息

位置 1	POINTJ	机器人起始位姿
位置 2	POINTC	机器人等待工件时的位姿（可结合 wobj_cvy_fixed 使用）
位置 3	POINTC	机器人跟踪工件起始点（该点必须与 wobj_cvy 配合使用）
位置 4	POINTC	机器人抓取工件位置点（该点必须与 wobj_cvy 配合使用）
位置 5	POINTJ	工件放置点



注意：针对位置 3、位置 4，这两个位置点的数值，不是通过示教的方式得到，而是必须对其直接编辑，并且数值根据抓取姿态有严格要求：

位置 3: POINTC (0, 0, Z, 0, 0, 180)

位置 4: POINTC (0, 0, 0, 0, 0, 180)

其中“Z”表示工具末端与工件抓取点之间的高度值。具体参看下方的程序截图。

以下为 RPL 程序代码、用户自定义变量截图：

### 一、Main 主程序

```
robot  Main newSub
No.  RPL程序
1  Main
2  MJOINT: v500, fine, tool9;
3  POINTJ(-40.90, 4.84, 95.48, -10.08, 90.20, 19.13)
4  IF: firstrun==false
5  + firstrun := true;
6  + track.init();
7  + cvyaxisgroup := ROBOT("linearbelt");
8  + INTRSET (intr, skipmove());
9  + INTRERRNO (intr);
10 + INTRDIS (intr);
11 + track.dropAllObj();
12 + End IF
13 + wobj_cvy_fixed := GETWOBJ(false, cvyaxisgroup);
14
15
16 + LABEL loop :
17 + cnt := cnt + 1;
18 + MESSAGE ("piece number = %1", cnt);
19 + getobjready := false;
```

```
20 + IF: !getobjready
21 + MLIN: v500, fine, tool9, wobj_cvy_fixed;
22 + POINTC(510.71, -231.87, 337.57, 29.97, -0.11, -179.71)
23 + getobjready := track.getWobj(20);
24 + End IF
25 + INTRENA (intr);
26 + MLIN: v500, fine, tool9, wobj_cvy;
27 + POINTC(0.00, 0.00, 50.00, 0.00, 0.00, 180.00)
28 + MLIN: v500, fine, tool9, wobj_cvy;
29 + POINTC(0.00, 0.00, 0.00, 0.00, 0.00, 180.00)
30 + DWELL (0.500);
31 + INTRDIS ();
32 + MJOINT: v500, fine, tool9;
33 + POINTJ(-40.90, 4.84, 95.48, -10.08, 90.20, 19.13)
34 + track.dropObj();
35 + GOTO loop;
36 + 结束
```



- Line 2 : 机器人以 P2P 的方式运动到“位置 1”处，即：起始姿态；
- Line 4 : 一些初始准备设置，包括：
  - (1) 跟踪视觉功能的初始化 (line 6)、
  - (2) 获取系统中名为“linearbelt”的轴组数据 (line 7)、
  - (3) 将中断变量“intr”与中断响应子程序“skipmove”进行关联 (line 8)、
  - (4) 设置“intr”的触发条件，即：当\_errno\_ != 0 时，触发中断 (line 9)、
  - (5) 关闭中断的触发 (line 10)、
  - (6) 清空跟踪队列中所有数据 (line 11)
- Line 13: 内置变量 wobj\_cvy\_fixed (传送带固定坐标系) 进行赋值；
- Line 23: 程序会一直阻断在此行，直到能成功获取到跟踪目标的数据；
- Line 25: 开启中断触发；
- Line 26: 位置 3，跟踪工件点。机器人会运动到距离工件抓取点高度 50mm 处；同时姿态调整为工具垂直向下，POINTC 后三个数值 (0, 0, 180)
- Line 28: 位置 4，抓取工件点。机器人运动到工件的抓取位置，同样采用工具垂直向下姿态。
- Line 30: 保持跟踪 0.5 秒；  
可在此行代码上方添加一行 IO 开/闭指令，控制工具 吸住/夹紧 工件。
- Line 31: 关闭中断触发；
- Line 32: 放置工件；
- Line 33: 删除跟踪队列中首个工件 (即：刚刚跟踪抓取的工件)，用以控制程序跟踪和抓取下一个工件；

## 二、skipmove 子程序

```
robot skipmove newSub
No. RPL程序
1 skipmove
2 track.dropObj();
3 STOPMOVE ();
4 CLEARMOVE;
5 STARTMOVE ();
6 MJOINT: v500, fine, tool9; 位置1: 回到起始点
7 POINTJ(-40.42, 1.33, 99.62, -10.72, 90.20, 19.61)
8 RESTART;
9 结束
```

- Line 2: 删除跟踪队列中首个工件数据；
- Line 3: 机器人停止当前动作；



- Line 4: 清除未完成的动作序列;
- Line 5: 开始执行当前程序行之后的指令;
- Line 6: 机器人运动到位置 1;
- Line 8: 程序重新开始;

### 三、程序中自定义变量



表 5-10 用户自定义变量列表

变量名	类型	变量归属	说明
firstrun	BOOL	Main 程序	判断是否是程序第一次运行标志
intr	INTR	Main 程序	中断用变量
cvyaxisgroup	ROBOT	Main 程序	获取“传送带”轴组参数
cnt	DINT	Main 程序	成功跟踪并抓取目标个数
getobjready	BOOL	Main 程序	用于接收是否成功获取跟踪目标

## 5.2 安全类

安全类中的各个功能页都与机器人协作安全相关, 在进行设置这些功能页中的参数前, 都需要输入安全密码, 才可以对其进行编辑, 如下图:



图 5-25 安全类功能页

**提示!**

安全类中各个功能页默认的解锁密码请咨询本司售后人员。

### 5.2.1 区域监控



图 5-26 区域监控

区域监控：以机器人 TCP 为参考点，监控该参考点与用户设定区域的相对位置。当 TCP 违反了设定，即越出或进入设定区域，系统会发出报警，同时停止机器人运动。



## 界面介绍




图 5-27 区域监控界面元素

表 5-11 界面介绍

#	名称	说明
1	区域监控使能	勾选，使能“区域监控”功能。
2	IO 控制使能	勾选时，禁用“监视”“控制”按钮。
3	区域名称	目前支持八个区域，可分别对其进行设置、启用。
4	区域类型	显示当前区域设置的类型，包括： 工作区：TCP 只能在区域内运动； 禁止区：TCP 不能进入此区域； 共享区：TCP 在区域内发出信号，反之取消。
5	状态图标	显示 TCP 端与设定区域的相对位置关系。
		 没有启用监视
		 在工作区内，且不违反
		 在工作区外，且违反
		 在禁止区外，且不违反
		 在禁止区内，且违反
6	“设置”按钮	进入“监控区域、TCP 区域设置”编辑页面。
7	“监视”按钮	开启/关闭该区域的监视。 监视：仅显示 TCP 端与区域的相对关系，当发生违反时，不做控制。




8	“控制”按钮	<p>在“监视”功能开启的基础上，开启“控制”功能，即当发生违反时，系统会立即报警、停止机器人运动。</p> <p>（注：当区域为“共享区”时，控制开关决定是否进入被占用的共享区域。）</p>
9	违反指示灯	<p>绿色，表示不违反。</p> <p>红色，表示违反。</p>

	<p><b>提示！</b></p> <p>“违反/不违反”只适用于“工作区”、“禁止区”； “共享区”不存在违反情况。</p>
---	---

## 设置“工作区、禁止区”

区域监控，设置步骤：

#	图示	说明
1		<p>① 点击“区域1设置”按钮，进入设置页。（此处以区域1为例）</p>
2		<p>① 区域类型选择工作区、禁止区、共享区。</p> <p>② TCP形状可以设置为立方体和球体。</p> <p>③ 区域形状仅可设置为立方体。</p> <p>④ 设置区域的方式。</p>



**提示!**

当选择“共享区”时，需要设置 DO 端口，将功能与该 DO 口绑定，可在“通用/IO 配置/区域、位置监控”功能页中操作。



3



① 设置监控区域的中心点位置及其姿态。

② 设置长方体边长。

③ 点击“>”按钮，进入下个设置界面。

4



① 设置 TCP 数值。

② TCP 形状：为“长方体”则设置边长，为“球体”则设置球半径。

③ 点击“保存”按钮，完成。



**提示!**

TCP 区域是指用长方体或者球体将机器人末端工具包围起来。



## 启用“监视、控制”

区域监控激活，操作步骤：

#	图示	说明
1		<p>① 勾选“激活”按钮，开启区域监控功能；</p> <p>② 点击“监视开关”按钮，开启对区域 1 的监视；</p> <p>③ 点击“控制开关”按钮，开启对区域 1 的控制。</p>



### 提示！

仅仅开启监视开关，控制系统仅会监视机器人的位置，不会进行任何管控操作。

## 违反恢复

区域违反报警后恢复步骤：

#	图示	说明
1		<p>① 区域违反指示灯变为红色，表示发生了区域违反。</p>



2		<p>① 点击“控制开关”按钮，关闭控制功能；</p> <p>② 点击“处理核查”，清除报警弹窗，此时上伺服将不会报警；</p> <p>务必在手动模式下运动机器人，JOG 机器人至工作区内或者禁止区外。</p>
---	--	---

## 启用 IO 控制

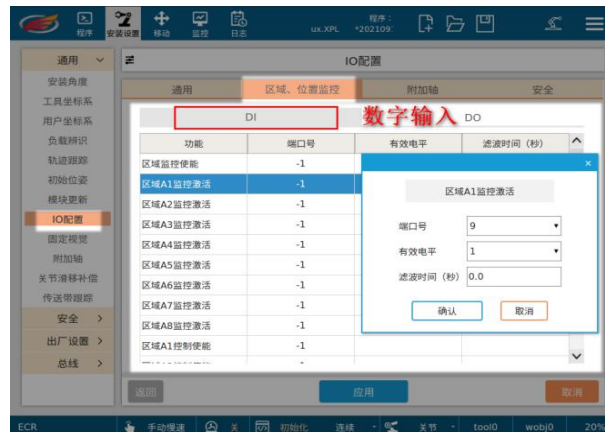
IO 控制激活，使用步骤：

#	图示	说明
1		<p>① 勾选“IO 控制”复选框后；</p> <p>UI 界面中“监视”、“控制”按钮，都会处于禁用状态，不可以通过点击，来开启“监视、监控”功能。</p>



### 提示！

需要在“通用/IO 配置”功能页中，配置相应的 DI 端口，通过外部设备对该 DI 端口输入高低电平，开启/关闭监视监控功能。





## Module 函数及 RPL 程序用例

编程实现区域监控功能步骤：

#	图示	说明
1	<p>The screenshot shows the RPL editor interface. On the left, a list of functions is visible, including 'areasmonitor'. A dropdown menu is open, displaying a list of functions: 'areasmonitor_init()', 'areasmonitor_bactive()', 'areasmonitor_baxsfact()', 'areasmonitor_baxsfena()', 'areasmonitor_bctrl()', and 'areasmonitor_benable()'. The 'areasmonitor_bactive()' function is highlighted.</p>	<p>① 三条不同指令代码完成不同功能的打开和关闭。</p>
2	<p>The second row contains three screenshots of RPL code. The first screenshot shows lines 2 and 3: <code>areasmonitor.bactive(true);</code> and <code>areasmonitor.bactive(false);</code>. The second screenshot shows lines 2 through 10: <code>areasmonitor.benable(1, true);</code> through <code>areasmonitor.benable(8, true);</code>. The third screenshot shows lines 2 through 10: <code>areasmonitor.bctrl(1, true);</code> through <code>areasmonitor.bctrl(8, true);</code>.</p>	<p>① 指令“areasmonitor.bactive()”功能是勾选“使能区域监控”，参数为“true”勾选，“false”时取消。</p> <p>② “areasmonitor.benable()”和 areasmonitor.bctrl() 可以控制监视列和控制列某个按钮开关，其中第一个参数为（1，2，3，4，5，6，7，8），选择八个区域中的某个区域，第二个参数为“true”时打开按钮，为“false”时关闭按钮。</p>



## 5.2.2 位置监控

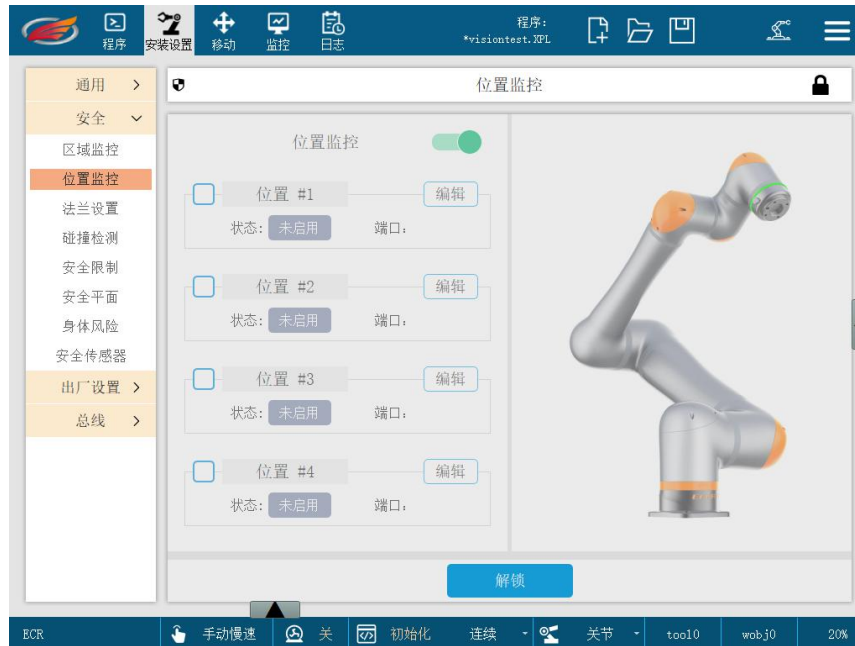


图 5-28 位置监控

点击“解锁”输入密码进入位置监控界面，位置监控功能被激活后，系统开始监视机器人的运动姿态，判断机器人是否运动至用户设定的目标位置（姿态）。同时，结合 IO 配置，将机器人已就位的信号发送给外部设置。该功能页中为用户提供了 4 个待监视的位置栏位。

### 启用位置监控

配置位置，操作步骤：


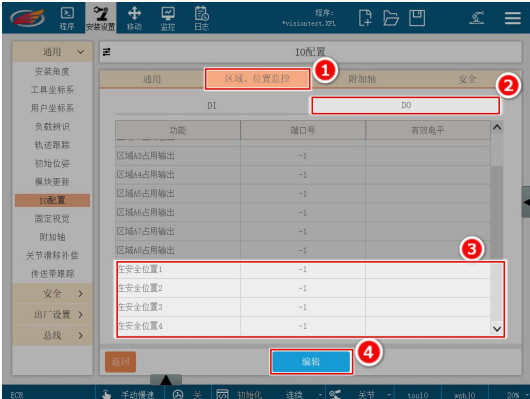
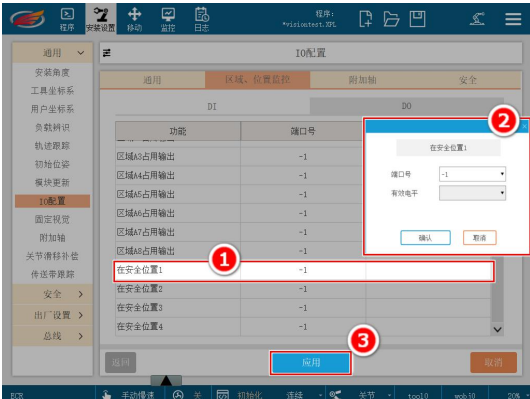
#	图示	说明
1		<p>① 点击开关按钮，激活位置监控功能。</p>



<p>2</p>		<p>① 点击“编辑”按钮，打开“监视位姿设定”页。</p>														
<p>3</p>	<table border="1" data-bbox="742 772 933 952"> <thead> <tr> <th>关节坐标</th> <th>允许误差</th> </tr> </thead> <tbody> <tr> <td>J1   0.00 deg +/- 0.01 deg</td> <td></td> </tr> <tr> <td>J2   0.00 deg +/- 0.01 deg</td> <td></td> </tr> <tr> <td>J3   0.00 deg +/- 0.01 deg</td> <td></td> </tr> <tr> <td>J4   0.00 deg +/- 0.01 deg</td> <td></td> </tr> <tr> <td>J5   0.00 deg +/- 0.01 deg</td> <td></td> </tr> <tr> <td>J6   0.00 deg +/- 0.01 deg</td> <td></td> </tr> </tbody> </table>	关节坐标	允许误差	J1   0.00 deg +/- 0.01 deg		J2   0.00 deg +/- 0.01 deg		J3   0.00 deg +/- 0.01 deg		J4   0.00 deg +/- 0.01 deg		J5   0.00 deg +/- 0.01 deg		J6   0.00 deg +/- 0.01 deg		<p>① 设置待监视的“关节角度”，以及“允许误差”； ② 点击“应用”按钮，保存设置。</p>
关节坐标	允许误差															
J1   0.00 deg +/- 0.01 deg																
J2   0.00 deg +/- 0.01 deg																
J3   0.00 deg +/- 0.01 deg																
J4   0.00 deg +/- 0.01 deg																
J5   0.00 deg +/- 0.01 deg																
J6   0.00 deg +/- 0.01 deg																
<p>4</p>		<p>① 勾选复选框，即可启用对该姿态的监视。</p>														



## 配置 IO

<p>1</p>		<p>① 进入“安装设置/通用/IO配置”功能页，在输出列表中；</p> <p>② 点击“功能性数字 IO 配置”，进入 IO 配置界面；</p> <p>本例中，上述步骤启用的是 #1 监控栏位，因此 DO 配置时，选择“安全位置 1”。</p>
<p>2</p>		<p>① 进入 IO 配置界面，点击“区域、位置监控”按钮；</p> <p>② 点击“DO”进入端口配置界面；</p> <p>③ 下拉菜单栏，找到安全位置部分；</p> <p>④ 点击“编辑”按钮，对应安全位置进行配置。</p>
<p>3</p>		<p>① 点击对应安全位置，会弹出编辑框；</p> <p>② 在编辑框设置安全位置 1 的端口号和有效电平；</p> <p>③ 点击“应用”完成 IO 配置。</p>



4		<p>如左图，表示 IO 配置应用成功。</p>
5		<p>切换回“安全/位置监控”页面；</p> <p>如左图，表示 IO 配置已完成。</p>

### Module 函数及 RPL 程序用例

表 5-12 位置监控相关 module 函数

说明项	说明
指令函数名	areasmonitor.baxsfact()
函数功能	激活“位置监控”功能。
形参个数	1
形参说明	BOOL 类型：true 表示激活、false 表示关闭。
返回参数	无
说明项	说明
指令函数名	areasmonitor.baxsfena()
函数功能	启用#n 号栏位的位置监视。
形参个数	2
形参说明	#1: DINT 类型，1~4 表示待启用的栏位。
	#2: BOOL 类型，true 表示启用监视、false 表示停用监视。
返回参数	无



表 5-13 位置监控相关 module 变量

变量名	类型	说明
safeJPos1	POINTJ	#1 监视位置
safeJPos2	POINTJ	#2 监视位置
safeJPos3	POINTJ	#3 监视位置
safeJPos4	POINTJ	#4 监视位置

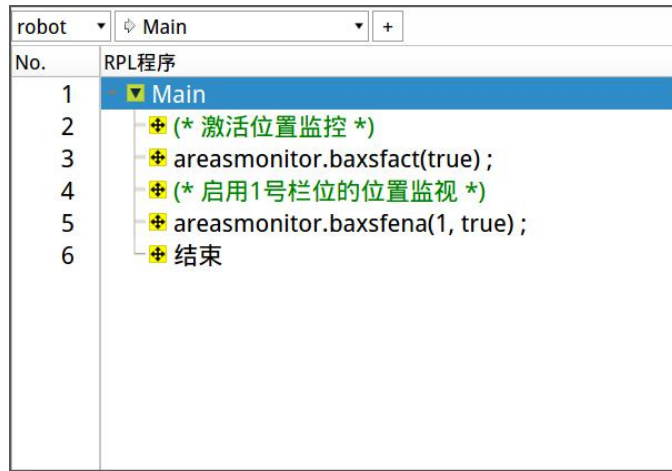


图 5-29 RPL 编程示例

### 5.2.3 法兰设置



图 5-30 末端配置

法兰设置功能页可设置：

- (1) 启用/停用“虚拟末端按钮”；
- (2) 设置末端法兰上的输出电压（0V/12V/24V），该电压为末端工具供电；
- (3) 配置末端“I/O 强制”按键的响应（末端法兰上的“○”空心圆实体按键）。



当该按钮被按下时，末端 2 个 DO 端口输出强制信号，电平信号是高或低由配置决定。

## 虚拟末端按钮

显示虚拟按钮，使用步骤：

#	图示	说明
1		<p>① 勾选“复选框”按钮；</p> <p>② 点击“应用”按钮。</p>
2		<p>软件界面上会出现一个悬浮按钮；</p> <p>① 点击该悬浮按钮。</p>
3		<p>窗口中的按钮与末端法兰上的实体按钮功能一致。</p>



### 提示！

“虚拟按钮”与“实体按键”，不可同时使用，即当虚拟按钮被启用后，实体的末端法兰按钮处于失效状态；必须将虚拟按钮停用后，实体按钮才可恢复使用。



## “IO 强制” 按钮配置

配置 IO 按钮，使用步骤：

#	图示	说明
1		<p>① 勾选“输出端口-1-”复选框；</p> <p>② 在“下拉框”中选择“高电平”；</p> <p>③ 点击“应用”按钮。</p>
2		<p>① 软件界面切换到“监控/内部/IO”界面。</p>
3		<p>① 将滚动条下拉到最底部（其中 DO-[24]、DO-[25] 分别对应法兰末端界面的输出端口-1，输出端口-2）；</p> <p>“DO-[24]”指示灯，会随着后续步骤，发生变化。</p>



<p>4</p>		<p>① 点击“虚拟按钮”悬浮按钮；</p> <p>② 点击“开”按钮；（此操作等同于 按下末端法兰“○”按键）</p> <p>这时“DO-[24]”指示灯现绿色，表示强制状态成功。</p>
<p>5</p>		<p>① 点击“关”按钮，取消强制状态。</p>



### 提示!

DO-[24]、DO-[25]两个数字输出端口，只可在末端输出电压不为“0”时，才可以发出高电平。



## 5.2.4 碰撞检测

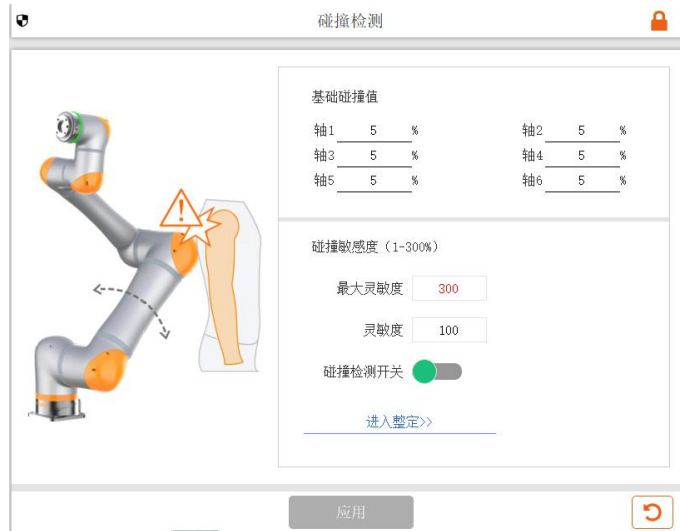


图 5-31 碰撞检测

碰撞检测功能用于监控机器人是否发生了碰撞事故，防止机器人出现进一步的损伤或损坏被撞物体。

### 检测原理

ECR 系列机器人采用基于动力学的碰撞检测算法。机器人在出厂前，已经建立了动力学模型、记录 6 个关节电机各自的力矩峰值，这些数据都是基于空载的状态（即机器人末端没有加装任何工具）。

当机器人发生碰撞，则碰撞时相应关节电机会产生实际力矩，该力矩值超出了出厂时设定的力矩峰值，系统认定发生了碰撞。

但是，由于机器人运动是存在轻微的抖动、批次之间存在结构上的细微差异，如果严格依据出厂时的力矩峰值，容易出现误报警。为了避免误报警的发生，需要用户根据自身使用情况设置一个“阈值”，即，在原力矩峰值的基础上再提高数值，从而减少误报。



“碰撞检测”页面中的各个设置项，供用户自定义设定一组“误差阈值”，其中：

$$\text{误差阈值} = \text{碰撞灵敏度} \times \text{额定力矩};$$

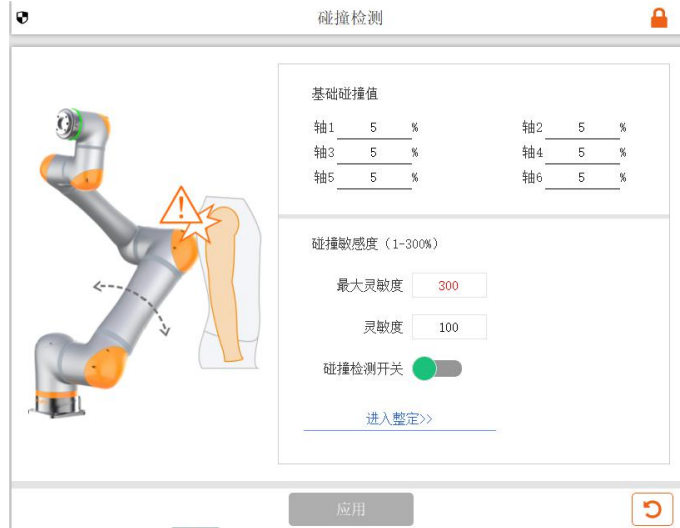


图 5-32 碰撞检测界面元素

表 5-14 界面元素说明

#	名称	说明
1	基础碰撞值	用于设置每个轴的基础灵敏度；
2	碰撞敏感度	包括： (1) 最大灵敏度：设置最大碰撞参数； (2) 灵敏度：设置当前手动、自动执行程序碰撞检测的灵敏度
3	碰撞检测开关	在运行程序以及手动示教，这两个使用场景下，同时开启/关闭碰撞检测功能。

在运行 RPL 程序时，各个关节的碰撞检测灵敏度，即判断发生碰撞的力矩值与出厂设定的力矩峰值比率为：

$$S(RPL) =$$

$$\text{“程序运行时灵敏度”} \times \text{“Module 函数中设定参数”} \times \text{“基础碰撞值”}$$



**提示！**

“Module 函数中的设定参数”，在下面“程序使用碰撞检测”小节中详述。




在 JOG 机器人时，各个关节的碰撞检测灵敏度为：


$$S(\text{Jog}) = \text{“示教运行时”} \times \text{“基础碰撞值”}$$

一旦：

$$\text{实时力矩} > \text{力矩峰值} + (S(\text{RPL}) \times \text{额定力矩}) ;$$

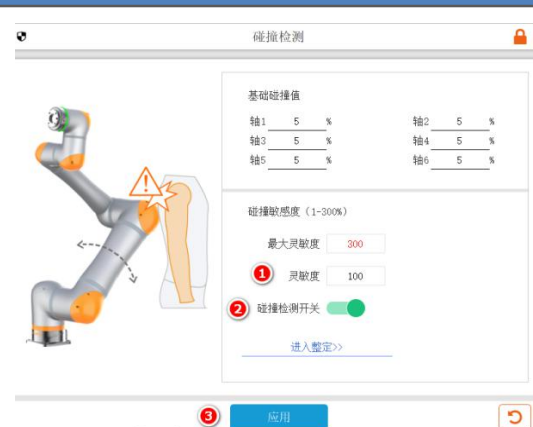
则判定机器人发生了碰撞。

	<p><b>提示！</b></p> <p><math>S(\text{RPL})</math>、<math>S(\text{Jog})</math>数值越小，则碰撞检测越灵敏，越容易产生误报警。</p>
---	---

	<p><b>提示！</b></p> <p>如果阈值设置过大可减少误报，但同时会增加漏报率，因此“基础碰撞值”推荐为 5%（5kg 负载及以内），用户通常情况下只需要适当调整“灵敏度”数值。</p>
---	--

## 程序中使用碰撞检测

程序中使用碰撞检测，步骤

#	图示	说明
1		<p>① 设置“程序运行时”碰撞灵敏度值；</p> <p>② 点击碰撞检测开关“开/关”按钮；</p> <p>③ 点击“应用”按钮，保存上述应用。</p>

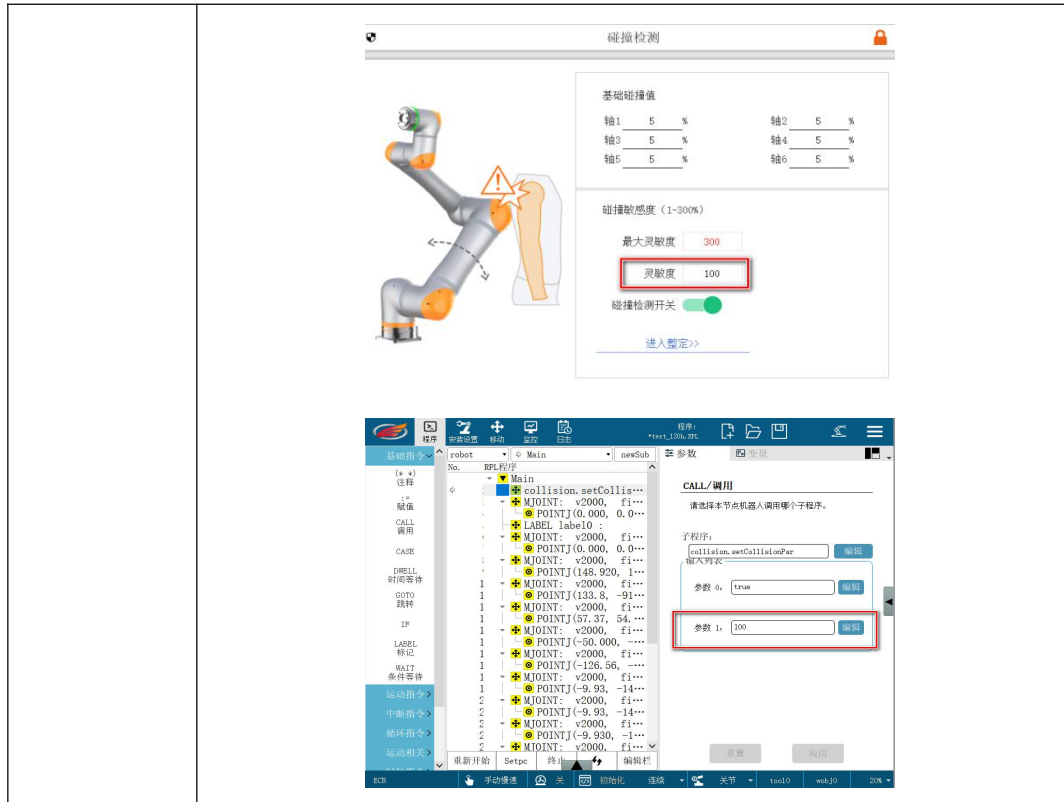


<p>2</p>		<p>① 在程序选项卡中打开 RPL 程序文件； (这里以“test-120h.XPL”为例)</p>
<p>3</p>		<p>① 点击“基础指令”，找到“CALL”指令；</p> <p>② 点击“CALL”指令，将其插入程序树中，程序树中显示“??? ;!error 1”。</p>
<p>4</p>		<p>① 选中 CALL 指令所在行，即第 1 行“??? ;!error 1”；</p> <p>② 点击“编辑”按钮。</p>
<p>5</p>		<p>① 选中“collision.setCollisionPar()”；</p> <p>② 点击“确认”按钮。</p>



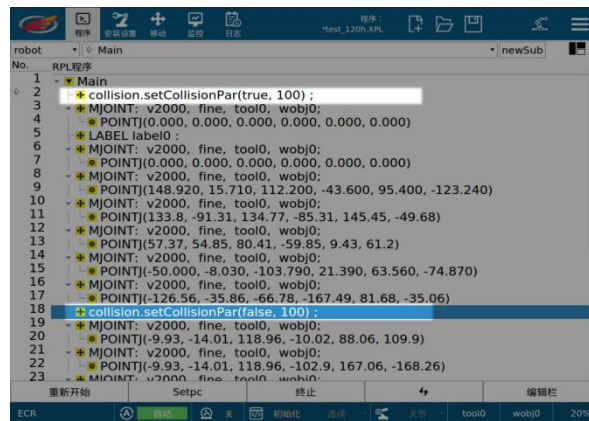
6		<p>① 点击“编辑”按钮，设置“参数 0”、“参数 1”中的数值分别为：“true”、“100”；</p> <p>② 点击“应用”按钮，保存上述设置。</p>
	<p><b>提示！</b></p> <p>参数 0: BOOL 类型，表示从此程序段开始，开启/关闭碰撞检测；</p> <p>参数 1: DINT 类型，表示上述 <math>S_{(RPL)}</math>公式中的“Module 函数中设定参数”，一个灵敏度增益。</p>	
7		<p>① 机器人状态设置为“自动模式+上伺服”；</p> <p>② 点击“程序运行”按键，使机器人执行程序。</p>
8	<p>机器人运动期间，使用物体去碰撞机器人本体，机器人能够停止，则碰撞检测功能正常工作。</p>	

	<p><b>提示！</b></p> <p>如果机器人在执行程序时，没有任何碰撞的情况下发出碰撞报警；</p> <p>或实际发生碰撞但没有停止及报警；</p> <p>则需要调节“灵敏度”或程序代码行中 collision.setCollisionPar 的参数 1。（数值越低越容易触发碰撞报警）</p>	
--	---	--



### 提示!

程序段中，如果在程序段的开头处使用 `collision.setCollisionPar(true)` 指令，则结尾处使用 `collision.setCollisionPar(false)`，该程序段可实现碰撞检测。在程序段之外，机器人运动不会受到检测。



## 加载负载

上述的操作都是在机器人处于空载的情况，当机器人使用于真实的作业中，即机器人末端加装了工具和工件，再启用碰撞检测功能，必须事先进行负载辨识



操作（参看 5.1.4 节），并在“负载辨识”页面加载该负载项，如下图：

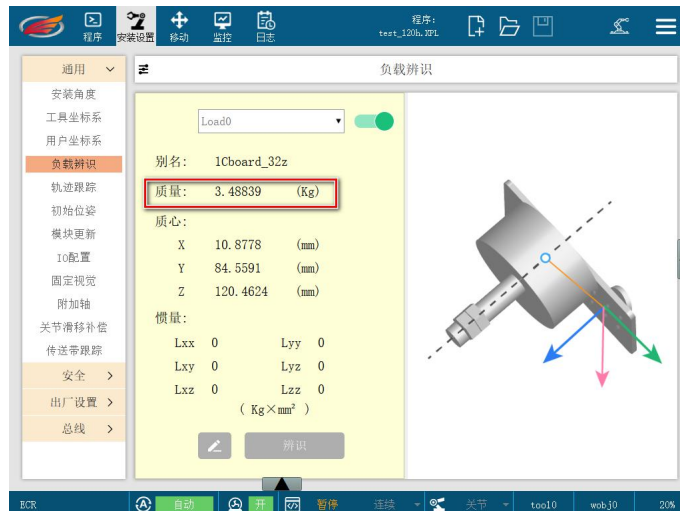


图 5-33 激活负载项

上述操作为动力学模型预估新的力矩峰值，之后，开启碰撞检测功能。如果负载数据设置不正确，或者有过大的过程力作用在机器人上，有可能会产生误报。

以下为部分可能出现的误报警情况，以及推荐的解决措施：

序号	异常情况	解决措施
1	工具负载设置不正确	使用负载辨识功能来定义负载，或者直接手动输入负载信息。
2	负载的质量或者惯量非常大	降低检测灵敏度。
3	工艺过程引入了很多外力	按照 30%的间隔降低碰撞检测的灵敏度，直到不再产生误报。
4	有短时间或者临时的外力作用	使用 <code>collision.setCollisionPar</code> 指令来降低灵敏度，或者关闭碰撞检测功能。
5	自动模式下，速度过快且灵敏度较高时	降低速度或降低灵敏度。
6	其他造成误报的情况	关闭碰撞检测功能。

## 整定

整定功能必须结合程序的自动运行才可使用，其作用是：采集机器人运动过程中各轴的力矩峰值。



整定使用步骤:

#	图示	说明
1		① 点击“进入整定”按钮。
		<p><b>注意!</b></p> <p>进入整定界面的同时，系统会自动关闭碰撞检测功能。</p>
2		① 点击“整定”按钮，系统会进入到力矩采集状态。
3		进行整定。



<p>4</p>		<p>① 以“自动模式”运行程序。</p> <p>（以“test-120h.XPL”程序为例，该程序中第1行 collision.setCollisionPar 参数设置为 true，即使碰撞检测界面中“碰撞检测开关”为关闭状态，程序中 collision.setCollisionPar 仍会起作用。）</p>
<p>5</p>		<p>待程序完整执行 2 个周期后，终止程序；</p> <p>① 切换回“碰撞检测”界面，点击“应用”按钮，以此关闭数据采集。</p>
<p>6</p>		<p>① 点击“退出”按钮。整定全部操作完成。</p>

执行完整定环节，之后再使用“程序运行时”碰撞检测，可以优化碰撞误报或漏报的现象。



## 5.2.5 安全限制



图 5-34 安全限制

表 5-15 安全限制界面说明

#	名称	说明
1	TCP 速度	限制机器人末端工具的笛卡尔空间速度，单位 m m/s。
2	TCP 力	限制机器人末端工具受到的力，单位 N。
3	停止时间	限制机器人停止所需时间，单位 s。
4	TCP 动量	限制机器人末端工具的动量，单位 kg.m/s。
5	TCP 功率	限制机器人末端工具的功率，单位 W。
6	停止距离	限制机器人停止所需距离，单位 s。
7-12	力矩误差	限制机器人 1-6 关节的力矩误差，单位 Nm。
13	激活	点击，即可激活当前设置的限制
14	应用	点击，即将当前设置保存，并重启控制器

安全限制功能页，可开启/关闭机器人运动的性能限制。如上图 5-34 所示。通过限制机器人的功率、TCP 端速度、电机力矩等参数保证人机协作安全。当系统检测超过了限定值，机器人将报警停止。



## 5.2.6 身体风险

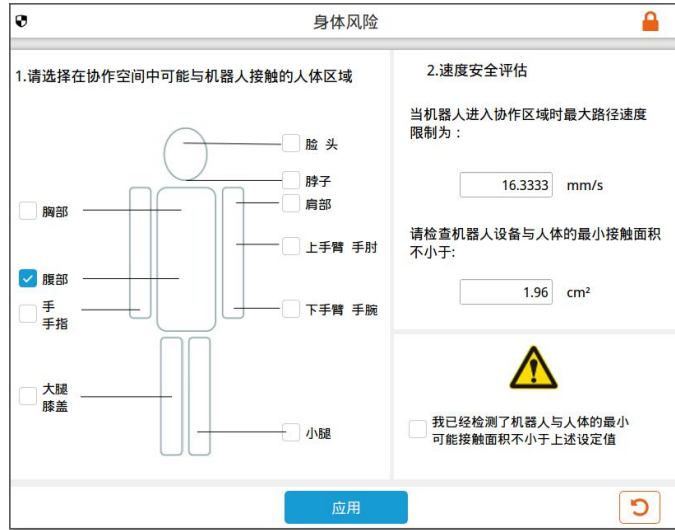
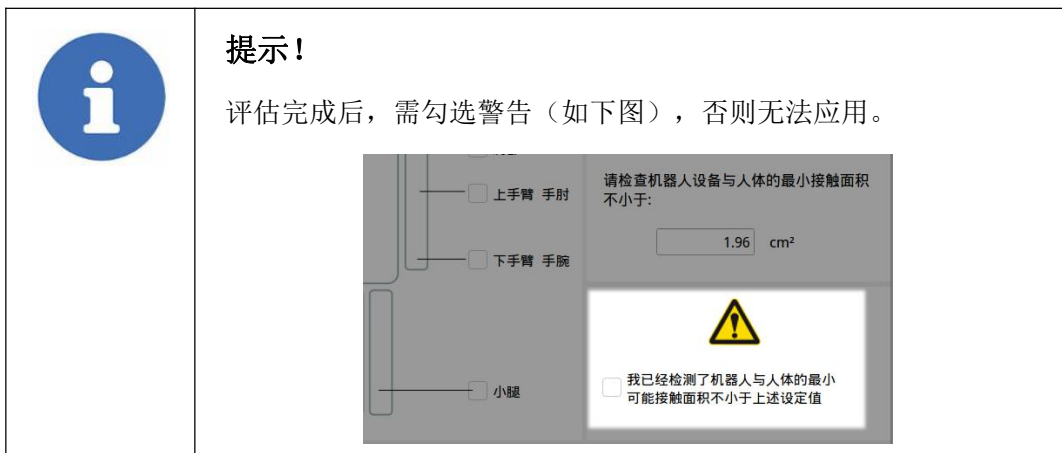


图 5-36 身体风险

身体风险功能页：根据实际工作中可能与机器人接触的人体区域（人机协作时人体的接触区域），在该功能页中勾选相应的身体部位，以此评估出人机协作时机器人应该使用的缩减模式的程度。点击界面上“应用”按钮，则使用当前评估得出的一组缩减程度。





### 5.2.7 安全传感器



图 5-37 安全传感器

安全传感器需要结合外围安全类传感设备，比如：安全激光扫描仪。ECR 机器人可以使用 2 种途径，进行传感设备对机器人单向的信号传输，分别是：“网络”（如下图 5-38）、“数字 IO”（如下图 5-39）：



图 5-38 TCP/IP 方式通信



图 5-39 数字输入端口方式通信

**区域参数：**每个区域都有一个参数，保护区域对应停止时间，警戒区域对应速度倍率。停止时间表示机器人停止需要的时间，最快为 0.2s。速度倍率表示当该区域触发时，机器人减速至用户设置的倍率。若警戒 1 和警戒 2 都启用，则警戒 1 的倍率应小于警戒 2，否则会提示错误。

**区域类型：**最多可以设置三个区域，勾选需要的区域即可，未勾选的区域无法设置对应的区域参数。区域包括保护、警戒 1、警戒 2，如下图 5-40。

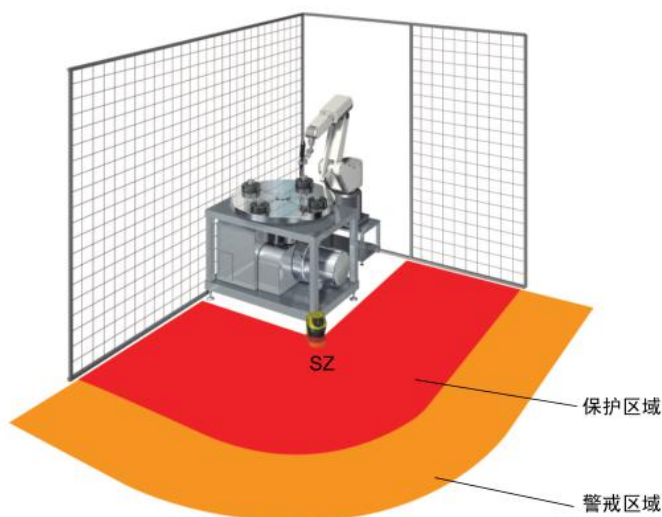


图 5-40 区域类型定义



**提示！**

“保护区”、“警戒区 1”、“警戒区 2”需要使用传感器厂家提供的配置软件进行设置。



**恢复模式：**当有外物进入敏感区域、触发机器人停止，之后等外物退出敏感区机器人以主动（自动）或被动（手动）的方式恢复运行。

- ◆ 主动方式：即“自动”，当外物离开后，机器人可自行恢复运动，无需人工参与；
- ◆ 被动方式：即“手动”，当外部离开后，机器人依旧停止，等待人工点击“恢复”按钮，才可继续运动，如下图：



图 5-41 恢复按钮

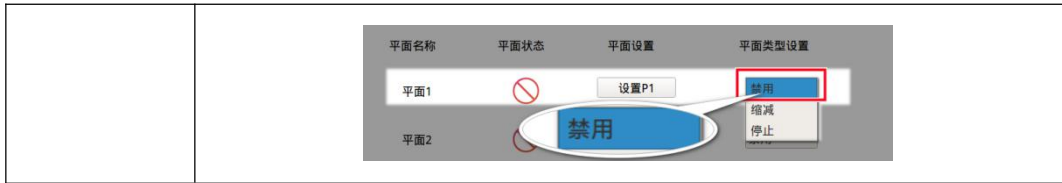
### 5.2.8 安全平面



图 5-35 安全平面

安全平面用于设置一个平面，当机器人运动进入到该平面后，则表示接下来要进行的是人机协作的操作，也可以理解为机器人进入了需要激活安全机制的平面。安全机制有 2 种，分别是：缩减、停止。

	<p><b>提示！</b></p> <p>选择“禁用”，表示所对应的平面，暂时不启用安全机制，如下图：</p>
--	---



安全平面，使用步骤：

#	图示	说明
1		<p>① 点击“设置 P1”按钮，进入“平面设置向导”页。</p>
2		<p>① 根据向导提示，完成 3 个位置点的设置；</p> <p>注：需要将机器人通过 Jog 方式运动到位置点后，点击“设置位置”按钮。</p>
3		<p>① 点击“设置位置”按钮，设置完最后 1 个位置点；</p> <p>② 点击“完成”按钮。</p>
4		<p>弹出“成功”提示框，表示设置平面完毕。</p>



平面名称	平面状态	平面设置	平面类型设置
平面1	<input type="checkbox"/>	设置P1	禁用 缩减 停止
平面2	<input type="checkbox"/>	设置P2	禁用
平面3	<input type="checkbox"/>	设置P3	禁用
平面4	<input type="checkbox"/>	设置P4	禁用
平面5	<input type="checkbox"/>	设置P5	禁用
平面6	<input type="checkbox"/>	设置P6	禁用

5

① 选择“缩减”或“停止”其中一个选项；

② 点击“开”按钮，即可应用当前设置。

**提示!**

一直点击“上一步”，可退出平面设置。如下图：

### 使用 TCP/IP 方式通信

如下图 5-42，将 IP、端口设置为传感器自身的 IP 地址（①）以及监听端口号（②），再勾选所要使用的区域类型（③），填入“区域参数”数值（④），再开启“速度与分离监控”按钮（⑤），即可正常使用。

图 5-42 网络方式通信设置

### 使用 DI 方式通信

DI 端口方式，其页面中的配置同上述的 TCP/IP 中的③~⑤，除此以外还需要



在“通用/IO 配置”页面中对传感器接入控制柜中的 DI 端口进行功能配置，如下图。

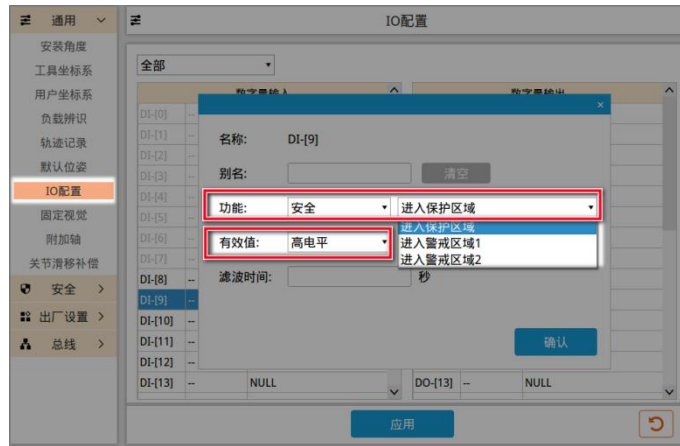


图 5-43 安全传感器 IO 配置



**提示!**

要注意传感器设备中 IO 信号线，要与安全区域一一对应，以及电平有效值。

### 5.3 出厂设置类

出厂设置类中的功能页，必须使用 Administrator 用户登录系统后，才可以使用。在使用具体功能页前，还需要输入“授权码”（如下图 5-44），之后才进行编辑操作。



图 5-44 “授权”按钮



**提示!**

授权码，请咨询本公司售后人员。



### 5.3.1 滑移基准点



图 5-45 滑移基准点

滑移基准点是在机器人进行零点标定后，记录一组关节角度与编码器值。当用户在使用机器人发生严重碰撞后，造成机器人走位出现偏差，这时可以结合“通用/关节滑移补偿”功能进行偏差补偿，即“滑移基准点”是“关节滑移补偿”功能的前提。

设置滑移基准点，操作步骤：

#	图示	说明
1		<p>① 直接编辑文本框，设置关节角度； （注：需要根据现场环境，设置可以运动到的姿态）</p> <p>② 点击“应用”按钮，保存关节角度设置。</p>
2		<p>① 将机器人设为“手动模式+上伺服”；</p> <p>② 扣住“使能键”；</p> <p>③ 按住“移至此位姿”按钮，将机器人移动至预设位姿。</p>



3		<p>① 点击“获取值”按钮，可获取当前位姿机器人的编码器值。</p>
---	--	-------------------------------------

### 5.3.2 驱动器清零

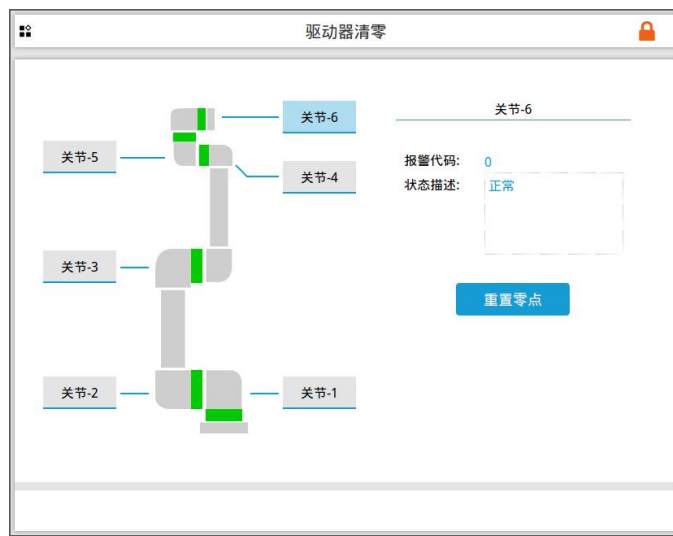


图 5-46 驱动器清零

在驱动器清零功能页中，可以查看机器人各个关节的报警代码、错误描述，还可以进行重置零位操作。

重置零点，使用步骤：

#	图示	说明
1		<p>① 以“步进运动”方式，JOG 机器人各轴，使机器人运动至实际现实中零点位置。</p> <p>（注意：此时各轴的数值不一定为‘0’）</p>



2		<p>进入“驱动器清零”页面；</p> <p>① 选中“关节-6”；</p> <p>② 点击“重置零点”按钮；即,完成对关节-6 的清零。</p>
3		<p>依次重复步骤 2，完成所有关节的重置操作。</p>

**提示！**

如果配置了附加轴，对附加轴的重置零位操作也在该界面进行。

**注意！**

重置零位的操作，必须由专业技术人员在“零点标定”操作之后才可执行。

由零点标定后得出的零点位置，才是最精确的数值；任何目测的方式 JOG 机器人至零位都存在偏差。

### 5.3.3 DH 参数

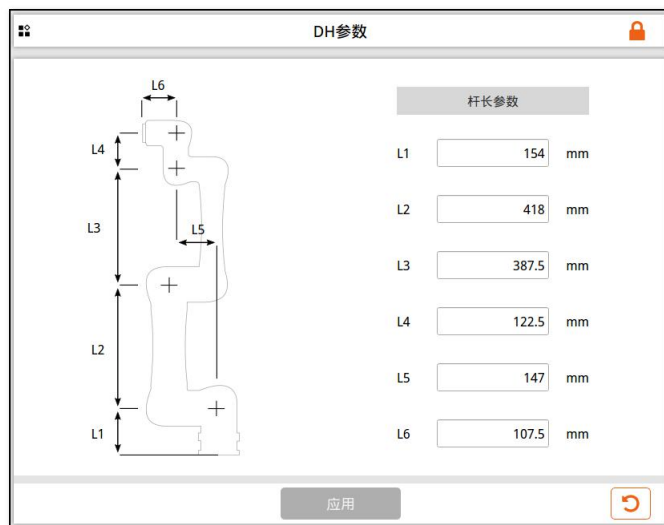


图 5-47 DH 参数



**注意！**

自行修改机器人 DH 参数存在风险，如果需要修改，建议联系售后技术人员确认后再进行修改。

### 5.3.4 关节限制



图 5-48 关节限值



**注意！**

自行修改机器人关节限值存在风险，如果需要修改，建议联系售后技术人员确认后再进行修改。

## 5.4 总线类

### 5.4.1 TCP/IP



图 5-49 TCP/IP 功能页-服务器



图 5-50 TCP/IP 功能页-客户端

在 TCP/IP 功能页中,可将控制器配置为 TCP/IP 通信协议中的客户端或服务器,以及配置其通讯参数。其中,服务器可以设置 3 个,客户端可以设置 5 个。

### 客户端通讯设置



图 5-51 客户端界面元素

表 5-16 客户端界面说明

#	名称	说明
1	连接状态	显示是否连接成功。
2	服务器 IP 地址	填写第三方设备的 IP 地址,要和机器人控制器网口在同一网段。
3	服务器端口号	填写服务器端口号。
4	连接超时	接收信息时如果时间超出这个值就会认为接收失败。



5	心跳时间	客户端与服务端在没有数据交互时，定时检测与服务器的通信链路是否存在。若检测不存在，客户端将主动断开。
6	首字符	默认为空，用户可在接受和发送的有效数据前自定义首字符和结束符，若首字符或结束符不为空，则在接收数据时，控制器会自动去除首尾字符，只存储有效数据；在发送数据时，则会自动加上首尾字符。
7	结束符参数	同编号 6，同时可通过勾选编号 8 和编号 9 在结束符后面添加回车换行符。
8	回车符	勾选，即可在结束符后面添加回车符。
9	换行符	勾选，即可在结束符后面添加换行符。

## 服务器通讯设置



图 5-52 服务器界面元素

表 5-17 服务器界面说明

#	名称	说明
1	连接状态	显示是否连接成功。
2	本机 IP	显示本机设置的 IP 地址。
3	监听端口	设置本机服务器的端口号。
4	连接超时	接收信息时如果时间超出这个值就会认为接收失败。
5	监听选项	勾选，下次开机默认自动打开监听功能。
6	启动监听按钮	手动设置停止和开始监听。
7	首字符	默认为空，用户可在接受和发送的有效数据前



		自定义首字符和结束符，若首字符或结束符不为空，则在接收数据时，控制器会自动去除首尾字符，只存储有效数据；在发送数据时，则会自动加上首尾字符。
8	结束符参数	同编号 7，同时可通过勾选编号 9 和编号 10 在结束符后面添加回车换行符。
9	回车符	勾选，即可在结束符后面添加回车符。
10	换行符	勾选，即可在结束符后面添加换行符。

## TCPIP 函数及 RPL 程序用例

表 5-18 位置监控相关 module 函数

1	说明
指令函数名	tcpip.sockopen( id )
函数功能	客户端通道开启。
形参个数	1
形参说明	DINT 类型，表示需要开启的套接字号。
返回参数	1
返参说明	DINT 类型： 返回值=1：开启成功 返回值=-1：开启失败 返回值=-2：重复开启
2	说明
指令函数名	tcpip.sockclose( id )
函数功能	客户端通道关闭。
形参个数	1
形参说明	DINT 类型，表示需要关闭的套接字号。
返回参数	1
返参说明	DINIT 类型： 返回值=1：关闭成功
3	说明
指令函数名	tcpip.socksend( id, str, type )
函数功能	发送信息。
形参个数	3
形参说明	#1: DINT 类型，需发送信息的套接字号。
	#2: STRING 类型，需发送的信息内容。
	#3: BOOL 类型：



	<p>true 表示 ‘客户端’ false 表示 ‘服务器’</p>
返回参数	1
返参说明	<p>DINIT 类型： 返回值=1：发送成功 返回值≠1：发送失败</p>
<b>4</b>	<b>说明</b>
指令函数名	tcpip.sockrecv( id, type )
函数功能	接收信息。
形参个数	2
形参说明	#1: DINT 类型，需接收信息的套接字号。
	#2: BOOL 类型： true 表示 ‘客户端’ false 表示 ‘服务器’
返回参数	2
返参说明	#1: STRING 类型，接收的字符串。
	#2: DINT 类型： 返回值=1：接收成功 返回值=-1：接收失败 返回值=-2：接收时间超时 返回值=-3：信息格式错误
<b>5</b>	<b>说明</b>
指令函数名	tcpip.getrobotdata( )
函数功能	获取机器人状态。
形参个数	0
形参说明	无
返回参数	4
返参说明	#1: STRING 类型：机器人状态
	#2: STRING 类型：报警信息
	#3: STRING 类型：笛卡尔坐标
	#4: STRING 类型：关节坐标

表 5-19 机器人状态格式说明

首字符	@
Byte1	伺服状态
Byte2	急停状态



Byte3	报警状态
Byte4	RPL 程序状态
Byte5	自动 / 手动 (T1) / 手动 (T2) 模式
Byte6	速率百分比
尾字符	&

表 5-20 机器人报警信息格式说明

首字符	@
Byte1	报警号
Byte2	报警信息
尾字符	&

表 5-21 机器人笛卡尔坐标格式说明

首字符	@
Byte1	X
Byte2	Y
Byte3	Z
Byte4	A
Byte5	B
Byte6	C
尾字符	&

表 5-22 关节坐标格式说明

首字符	@
Byte1	J1
Byte2	J2
Byte3	J3
Byte4	J4
Byte5	J5
Byte6	J6
Byte7-10	Aux1-4
尾字符	&

表 5-23 TCP/IP 相关 module 变量



变量名	类型	说明
tcpip.client_con[i]	BOOL 类型数组	客户端连接状态 其中 i 表示客户端的第几个套接字。
tcpip.server_con[i]	BOOL 类型数组	服务器连接状态 其中 i 表示服务器的第几个套接字。

## 字符串函数说明

表 5-24 字符串 module 函数

1	说明
指令函数名	str_fun.str2int( str )
函数功能	将字符串转换成整型。
形参个数	1
形参说明	STRING 类型：待转换的数据变量。
返回参数	1
返参说明	DINT 类型：转换完成的数据变量。
2	说明
指令函数名	str_fun.str2real( str )
函数功能	将字符串转换成实数。
形参个数	1
形参说明	STRING 类型：待转换的数据变量。
返回参数	1
返参说明	LREAL 类型：转换完成的数据变量。
3	说明
指令函数名	str_fun.int2str( dint )
函数功能	将整型转换成字符串。
形参个数	1
形参说明	DINT 类型：待转换的整型变量。
返回参数	1
返参说明	STRING 类型：转换完成的数据变量。
4	说明
指令函数名	str_fun.real2str( lreal )
函数功能	将实数转换成字符串。
形参个数	1
形参说明	LREAL 类型：待转换的数据变量。



返回参数	1
返参说明	STRING 类型：转换完成的数据变量。
<b>5</b>	<b>说明</b>
指令函数名	str_fun.strlen( str )
函数功能	获取字符串长度。
形参个数	1
形参说明	STRING 类型：待处理的字符串变量。
返回参数	1
返参说明	DINT 类型：字符串长度值。
<b>6</b>	<b>说明</b>
指令函数名	str_fun.strcmp( str1, str2 )
函数功能	字符串比较。
形参个数	2
形参说明	#1: STRING 类型：待比较字符串 1 #2: STRING 类型：待比较字符串 2
返回参数	1
返参说明	DINT 类型： 返回值=0：表示 str1=str2 返回值>0：表示 str1>str2 返回值<0：表示 str1<str2
<b>7</b>	<b>说明</b>
指令函数名	str_fun.strsplit( str1, delim, N )
函数功能	字符串分隔并提取。
形参个数	3
形参说明	#1: STRING 类型：待分隔的字符串 #2: STRING 类型：分隔符 #3: DINT 类型：拆分后的第 N 个字符串
返回参数	1
返参说明	STRING 类型：第 N 个分隔得出的字符串
<b>8</b>	<b>说明</b>
指令函数名	str_fun.strleft( str1, N )
函数功能	提取子字符串，从左开始的 N 个字符所组成的字符串。
形参个数	2
形参说明	#1: STRING 类型：待提取的字符串 #2: DINT 类型：待取字符串长度，N 从 1 开始



返回参数	1
返参说明	STRING 类型：提取出的子字符串
<b>9</b>	<b>说明</b>
指令函数名	str_fun.strright( str1, N )
函数功能	提取子字符串，从右开始的 N 个字符所组成的字符串。
形参个数	2
形参说明	#1: STRING 类型：待提取的字符串
	#2: DINT 类型：待取字符串长度，N 从 1 开始
返回参数	1
返参说明	STRING 类型：提取出的子字符串
<b>10</b>	<b>说明</b>
指令函数名	str_fun.strmid( str1, pos, N )
函数功能	提取子字符串，从第 pos 个字符开始连取 N 个字符。
形参个数	3
形参说明	#1: STRING 类型：待提取的字符串
	#2: DINT 类型：取字符的起始位
	#3: DINT 类型：待取字符串长度
返回参数	1
返参说明	STRING 类型：提取出的子字符串
<b>11</b>	<b>说明</b>
指令函数名	str_fun.getbit( int, N )
函数功能	整型取位。
形参个数	2
形参说明	#1: DINT 类型：待处理的整型数值
	#2: DINT 类型：待取位的第 N 位
返回参数	1
返参说明	BOOL 类型：取位的值
<b>12</b>	<b>说明</b>
指令函数名	str_fun.strconcat( str1, str2, delim )
函数功能	字符串拼接。
形参个数	3
形参说明	#1: STRING 类型：待拼接的字符串 1
	#2: STRING 类型：待拼接的字符串 2
	#3: STRING 类型：分隔符
返回参数	1



返参说明	STRING 类型：输出的字符串为“str1+delim+str2” 注意：字符串中没有“+”，只是为了说明。
------	--

### 5.4.2 MES



图 5-53 MES 通讯配置页面

MES 功能页：设置监听的相关配置、启用监听。



图 5-54 MES 界面元素

表 5-25 MES 界面说明

#	名称	说明
1	开启监听按钮	开启/停止监听，即允许/禁止建立连接请求。
2	控制器 IP	显示当前控制器的 IP,为禁用不可编辑状态。
3	监听模式	分为“广播”和“问询”2 种模式。
		广播 指服务器主动发送数据，支持机器人协议、附加轴协议。



		问询	指客户端发送问询指令，服务器应答机器人状态，只支持机器人协议。
4	间隔时间		只在广播模式下有效，发送数据的间隔时间。
5	端口		设置监听端口号。
6	监听内容		分为“机器人”和“附加轴”2种内容： 机器人：机器人状态信息。 附加轴：只有附加轴位置信息。

## MES 系统协议

协议内数据以内存的方式进行存储。

## 机器人协议内容

表 5-26 MES 系统主发机器人

#	意义	变量	类型	字节	备注
1	报文开始标记	Packet_Start	string	16	MessageHead
2	数据长度	Packet_Length	short	2	38
3	数据命令	Packet_Orders	short	2	1001
4	数据心跳	Packet_Heartbeat	short	2	
5	报文结束标记	Packet_End	string	16	MessageTail



### 提示！

“数据心跳”需要本次和上次发送不一样的数值。

表 5-27 机器人回发 MES 系统

#	意义	变量	类型	字节	备注
1	报文开始标记	Packet_Start	string	16	MessageHead
2	数据长度	Packet_Length	short	2	788
3	数据命令	Packet_Orders	short	2	1002
4	数据心跳	Packet_Heartbeat	short	2	
5	报警状态	bErrorStatus	bool	1	1: 有报警; 0: 无报警
6	急停状态	bHstopStatus	bool	1	1: 无急停; 0: 有急停
7	权限状态	bAuthorityStatus	bool	1	没有外部 PLC 模



					式
8	伺服状态	bServoStatus	bool	1	1: 有使能; 0: 未使能
9	轴运动状态	bAxisMoveStatus	bool	1	1: 有运动; 0: 未运动
10	程序运行状态	bProgMoveStatus	bool	1	1: 有运行; 0: 未运行
11	程序加载状态	bProgLoadStatus	bool	1	1: 有加载; 0: 无加载
12	程序暂停状态	bProgHoldStatus	bool	1	1: 有暂停; 0: 无暂停
13	模式状态	nModeStatus	short	2	0:手动慢速; 1:手动全速; 2:自动
14	速度状态	nSpeedStatus	short	2	百分比(单位)
15	IoDOut 状态	bIoDOut[0..31]	bool	1*3 2	数字量输出: 0-7 本地数字量 输出 1-8; 8-31 远程数字 量输 0-23;
16	IoDIn 状态	bIoDIn[0..31]	bool	1*3 2	数字量输入: 0-7 本地数字量 输入 1-8; 8-31 远程数字量 输入 0-23
17	IoIOut 状态	nIoIOut[0..31]	int	4*3 2	预留
18	IoIIn 状态	nIoIIn[0..31]	int	4*3 2	预留
19	加载工程名	strProjectName	string	32	预留
20	加载程序名	strProgramName	string	32	
21	错误信息	strErrorText	string	128	
22	各轴角度	dbAxisPos[0..6]	float	4*7	度(单位)
23	姿态位姿	dbCartPos[0..5]	float	4*6	毫米 度(单位)
24	各轴速度	dbAxisVel[0..6]	float	4*7	度/秒(单位)
25	各轴加速度	dbAxisAcc[0..6]	float	4*7	度/秒^2(单位)
26	各轴加加速度	dbAxisJer[0..6]	float	4*7	度/秒^3(单位)



27	各轴力矩	dbAxisTor[0..6]	float	4*7	额定力矩百分比 (单位)
28	各轴反向计数	nAxisDirCnt[0..6]	unsigned int	4*7	次(单位)统计
29	各轴工作总时长	nAxisTime[0..6]	unsigned int	4*7	秒(单位)统计
30	设备开机总时长	nDeviceTime	unsigned int	4	秒(单位)统计
31	报文结束标记	Packet_End	string	16	MessageTail

**提示!**

“数据心跳”需要发送接收到的数据心跳数值。

### 附加轴协议内容

表 5-28 MES 系统主发机器人

#	附加轴数量	数据类型	备注
1	0	string	Current robot has no auxiliary axes\r\n
2	1	string	EA1:xx.xx;\r\n
3	2	string	EA1:xx.xx;EA2:xx.xx;\r\n

### 5.4.3 EtherCAT 设置



图 5-55 EtherCAT 设置页面



EtherCat 操作步骤:

#	图示	说明
1		① 点击使能 EtherCAT 从站按钮选中或取消，以此开启或关闭 EtherCAT 功能。
2		① 点击“应用”，应用此设置。

#### 5.4.4 EthernetIP 设置



图 5-56 EthernetIP 设置页面

EthernetIP 操作步骤:

#	图示	说明
---	----	----



1		<p>① 选择对应本地网口，修改 EtherNet 网口（目前本地网口只存在 ETH3，选择后自动读取对应网口设置的 IP 和子网掩码）。</p>
2		<p>① 点击右移箭头“→”，应用此设置，出现提示框，选择“是”，重启示教器和控制器应用此设置。</p>



# 第 6 章 移动选项卡



图 6-1 移动界面

移动选项卡界面可使用的功能如下：

- (1) 查看机器人关节角度值、工具末端实时位姿；
- (2) 切换工具坐标系、用户坐标系；
- (3) 运动机器人，包括：初始位姿、回零位、步进运动。

## 6.1 位姿信息

下图 6-2 中高亮处为数据显示区，包括：

- a、工具末端坐标位置（基于机器人坐标系）；
- b、机器人各个关节实时角度。

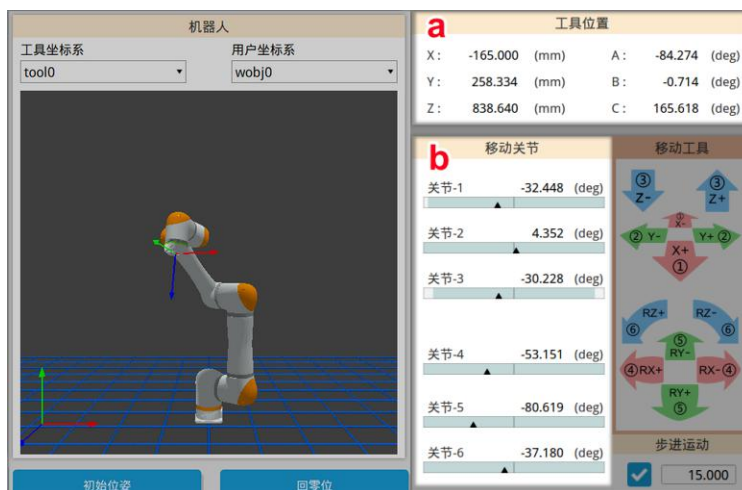




图 6-2 位置信息

## 6.2 坐标系统

ECR 系列机器人使用直角坐标系（即笛卡尔坐标系）来描述三维空间中的对象的位置和姿态。下图 6-3 为 ECR 机器人涉及到的坐标系：

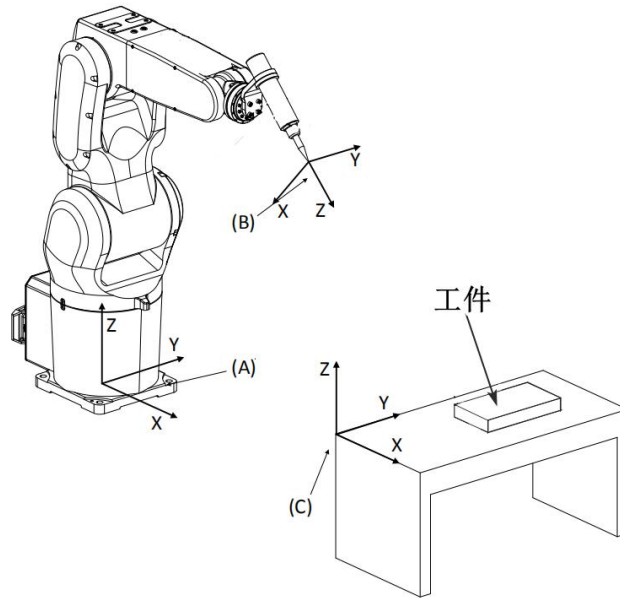


图 6-3 坐标系

- (A) 机器人坐标系（又称“基坐标系”），坐标系原点在机器人底座的中心点。
- (B) 工具坐标系，坐标系原点在工具末端尖点位置。
- (C) 用户坐标系，配合工件充当参考系，通常不单独使用。

## 6.3 点动机器人（JOG）

“点动机器人”是指：在“手动模式”下以“关节方式”或“笛卡尔方式”运动机器人。第 3 章已描述基本操作步骤。本节内容结合“移动”界面介绍点动机器人（即 Jog）。

在“手动模式”下，点击示教器面板上“坐标系”按钮（见下图 6-4），可依次切换运动参考坐标系为：关节、机器人、工具、用户。示教器 HMI 状态栏中相应图标，如下表 6-1：



图 6-4 “坐标系”按钮

表 6-1 运动参考系



图标	说明
关节	控制键“-”“+”按下时，机器人以单关节运动。
机器人	控制键按下时，机器人末端以机器人坐标系做笛卡尔运动。
工具	控制键按下时，机器人末端以工具坐标系做笛卡尔运动。
用户	控制键按下时，机器人末端以用户坐标系做笛卡尔运动。

### 6.3.1 关节运动

当运动参考系切换为“关节”时，“移动”界面中“移动关节”标签被激活（见下图 6-5），表示控制键①~⑥与关节 1 轴~6 轴相对应，



图 6-5 关节运动

### 6.3.2 笛卡尔运动

当运动参考系切换为“机器人”、“工具”、“用户”时，UI 界面中“移动工具”标签被激活（见下图 6-6），表示控制键①~⑥在相应的轴线上做直线运动或绕轴旋转运动。





图 6-6 笛卡尔运动



### 6.3.3 步进运动

勾选“步进运动”复选框后 JOG 机器人，每按一次控制键，机器人首先运动到相应方向距离最近的步进长度值整数倍位置，然后再按键机器人每次运动一个步进长度距离。

比如一轴关节角度目前为 11.96 度，选择步进运动设置步进长度为 15，按一轴“+”按键一次，此时机器人运动至距离 11.96 最近的 15 度，其中 15 度是此时步进长度最近的正整数倍位置，而后每次按一轴“+”按键一次，一轴关节角度增加 15 度。下表为步进运动前/后效果。

步进运动，示例（本例中只按一次控制键 1 的“+”键）

方式	运动前	运动后
关节		
方式	运动前	运动后



笛卡尔  
(机器人)

## 6.4 激活坐标系

“工具坐标系”、“用户坐标系”下拉框表示当前机器人运动学所使用的坐标系，选中下拉框中的项即可对两个坐标系进行相应的激活，如下图 6-7。



图 6-7 切换坐标系

### 6.4.1 激活工具坐标系

保持机器人姿态不变，激活不同的工具坐标系后，“工具位置”栏中的数值相应变化，以 tool1 的数据为【X: 0, Y: 0, Z: 120, A: 0, B: 0, C: 0】为例，下图为加载 tool1 前后，工具末端位置数值变化。



图 6-8 切换工具坐标系

同时，如果机器人使用“工具坐标系”进行笛卡尔方式的运动，则控制键中的1、2、3会以该加载的工具坐标系的X轴、Y轴、Z轴直线运动。



图 6-9 笛卡尔运动（工具坐标系）

### 6.4.2 激活用户坐标系

激活不同“用户坐标系”后，再以“用户”方式运动机器人，则机器人会以当前激活的用户坐标系轴线方向运动。



图 6-10 笛卡尔运动（用户坐标系）

## 6.5 初始位姿&回零位

### 6.5.1 初始位姿

初始位姿，操作步骤：

#	图示	说明
---	----	----



<p>1</p>		<p>① 机器人状态设置为：“手动模式”+“上伺服”；</p> <p>② 保持按住“使能键”；</p> <p>③ 持续按住屏幕中“初始位姿”按钮，使机器人运动。</p>
<p>2</p>		<p>或者：</p> <p>① 机器人状态设置为：“自动模式”+“上伺服”；</p> <p>② 持续按住屏幕中“初始位姿”按钮，使机器人运动。</p>
<p>3</p>		<p>机器人运动就位后，会弹出提示，表示机器人已运动到初始位姿。</p> <p>① 点击“OK”按钮，完成。</p>



**注意！**

使用该功能前需考虑机器人当前实际可移动区域，避免自动回位过程中造成碰撞。

### 6.5.2 回零位

“回零位”操作同“初始位姿”。



# 第 7 章 监控选项卡

监控选项卡中可实时监控控制柜收发的 I/O 信号、机器人各个关节运行状态，以及来查看总线通讯中的数据，同时可以设置数据的输出值。



图 7-1 监控选项卡

## 7.1 关节状态监控

可在此界面查看关节运转时力矩值、实时温度值，见下图 7-2；

温度/力矩监控		
关节	力矩	温度
关节1	-- N-m	-- °C
关节2	-- N-m	-- °C
关节3	-- N-m	-- °C
关节4	-- N-m	-- °C
关节5	-- N-m	-- °C
关节6	-- N-m	-- °C

图 7-2 关节状态监控页面



## 7.2 IO



图 7-3 I/O 监控页面

### 7.2.1 强制 I/O 信号

点击“强制模式”开关，打开强制信号模式，该模式下可对模拟输出信号、数字输出信号进行真实的强制，对模拟输入信号、数字输入信号进行虚拟的强制（注：输入信号的强制多为调试程序时使用）。

表 7-1 强制操作步骤：

#	图示	说明
1		数字量 I/O 强制： ① 点击相应端口“强制”开关，即可强制该端口。



<p>2</p>		<p>模拟量 I/O 强制：</p> <p>① 点击“输出”，进入设置界面；</p> <p>② 点击“强制”可强制该端口；</p> <p>③ 点击对应模拟量端口，输入数值；</p> <p>④ 点击“√”，数值设置完成。</p>
<p>3</p>		<p>① 点击“应用”，设置成功。</p>

	<p><b>提示！</b></p> <p>“强制”是指：</p> <ul style="list-style-type: none"> <li>■ 数字量 IO，将其当前状态取反；</li> <li>■ 模拟量 IO，可修改其输出的数值。</li> </ul>
--	--

	<p><b>提示！</b></p> <p>在自动模式（Auto）下：</p> <p>DO 信号强制无效；</p> <p>DI 信号可以强制（调试程序时模拟外部设备向机器人发送电平信号）。</p>
--	---



**提示!**

DO-[24]、DO-[25]两个数字输出端口，只可在末端输出电压不为“0”时，才可以发出高电平。



### 7.3 现场总线数据监控

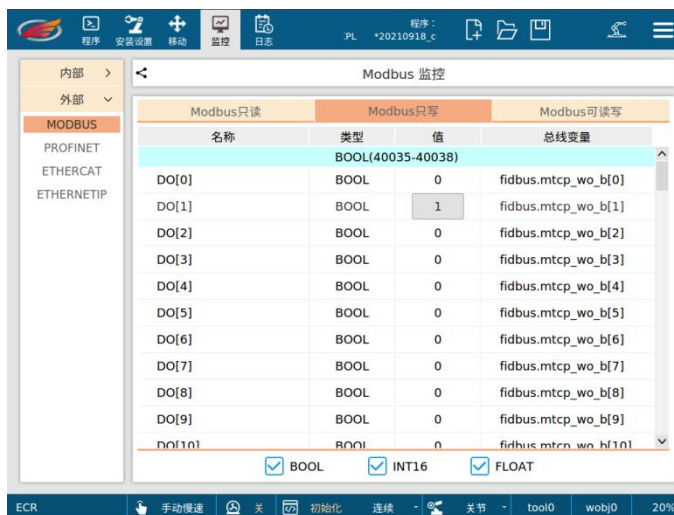


图 7-4 现场总线监控界面

目前主要包括 ModbusTCP, Profibus (仅支持 Robox RP2 Pro 控制器), EtherCAT, EthernetIP。ModbusTCP 该功能页分为只读, 只写和可读可写三个部分, 其中 ModbusTCP 只读, 只写的输入输出各自包括 64 个 bool、6 个 int、24 个 float 类型的数据; 可读可写部分包括 64 个 bool、64 个 int、16 个 float 类型的数据; Profibus, EtherCat, EthernetIP 功能页输入输出各自包含 64 个 bool、6 个 int、24 个 float 类型数据。

该功能页可进行总线数据监控与设置总线输出等操作。

#### 7.3.1 Modbus 功能

Modbus 数据查看使用步骤:

#	图示	说明
---	----	----



<p>1</p>		<p>在界面中选择需要查看的通讯协议的数据。</p> <p>① 点击“监控/外部/MODBUS”进入 Modbus 监控界面；</p> <p>② 点击“Modbus 只读”，“Modbus 只写”或“Modbus 可读写”，此处以“Modbus 只读”为例。</p>
<p>2</p>		<p>① 点击“Modbus 只读”后可以看到数据列表。</p> <p>在数据中可以查看以下数据：</p> <p>第 1 列是数据名称； 第 2 列是数据类型； 第 3 列是数据的值； 第 4 列是对应的示教器程序中的变量名称。</p> <p>② 可以通过勾选对应的数据类型 BOOL, INT16, FLOAT 框显示相应类型的数据。</p>

## Modbus 数据输出

设置 Modbus 数据输出使用步骤：

#	图示	说明
---	----	----



<p>1</p>		<p>① 点击“Modbus 只写”可看到数据输出列表;</p> <p>② 点击需要输出的变量的“值”这一列, 则弹出数字键盘。以一个 Bool 变量为例。</p>
<p>2</p>		<p>① 在弹出的键盘中输入值;</p> <p>② 输入完成后点击“√”, 完成输入。</p>
		<p>① 输入完成后该变量值被成功写入, 该监控界面将显示为写入后的值。</p>

## ModbusTCP 协议

ModbusTcp 的 IP 地址即为控制器设置地址, 端口号固定为 502。功能分为两个部分: 可读可写部分和只读只写部分。其中只读只写部分又分为机器人的接收端和发送端。

### 只读、只写变量

只读只写部分按协议内容分为系统协议和用户协议。

系统协议内容, 这部分由开发人员进行配置及维护, 主要是读取及设置机器人状态及运动参数等。用户只能通过固定的地址对固定的变量进行读写。



用户层协议内容，该部分有终端用户自行配置及维护，该部分工程只有在使用机器人语言编程中可以使用，不支持在工艺包中使用。

接收端和发送端各留有 64 个 BOOL, 6 个 int,24 个 float 数据接口，终端用户可以通过示教器编写程序读写主站 PLC 的数据。具体的地址与变量映射关系如下表 7-2、表 7-3 所示。

表 7-2 ModbusTCP 接收端协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	40101	I16	2	BOOL	Bit0: 上/下伺服（脉冲）
					Bit1: 运行（脉冲）
					Bit2: 停止（脉冲）
					Bit3: 清除报警（脉冲）
					Bit4: 加载程序（脉冲）
					Bit5: 重新开始（程序回到第一行）（脉冲）
					Bit6: Plc 报警（高电平）
					Bit7: 伺服准备确认（脉冲）
	40102	I16		BOOL	系统预留 BOOL 变量, 用户不可用
	40103	I16	2	I16	设置机器人速度
40104	I16	2	I16	加载目标程序号, 例: 首先设置目标程序号为 2, 然后给 40101 的 Bit4 高电平触发信号, 完成程序加载 (在程序运行过程中不可加载)	
40105	I16	2	I16	附加轴轴号选择 (1: 七轴 2: 八轴 3: 九轴 4: 十轴)	
40106	I16	2	I16	附加轴速度设定	
40107~ 40110	I16		I16	系统预留 Int 变量, 用户不可用	
40111~ 40133	FLOAT		FLOAT	系统预留 Float 变量, 用户不可用	
变量类别	物理地址	单位	字节	子单位	备注



			数		
用户变量	40135	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_ro_b[0] ~ fidbus.mtcp_ro_b[15]
	40136	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_ro_b[16] ~ fidbus.mtcp_ro_b[31]
	40137	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_ro_b[32] ~ fidbus.mtcp_ro_b[47]
	40138	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_ro_b[48] ~ fidbus.mtcp_ro_b[63]
	40139	I16	2	I16	对应 fidbus.mtcp_ro_i[0]
	40140	I16	2	I16	对应 fidbus.mtcp_ro_i[1]
	40141	I16	2	I16	对应 fidbus.mtcp_ro_i[2]
	40142	I16	2	I16	对应 fidbus.mtcp_ro_i[3]
	40143	I16	2	I16	对应 fidbus.mtcp_ro_i[4]
	40144	I16	2	I16	对应 fidbus.mtcp_ro_i[5]
	40145	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[0]
	40147	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[1]
	40149	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[2]
	40151	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[3]
	40153	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[4]
	40155	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[5]
	40157	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[6]
	40159	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[7]
	40161	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[8]
	40163	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[9]
	40165	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[10]
	40167	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[11]
	40169	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[12]
	40171	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[13]
	40173	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[14]
	40175	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[15]
40177	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[16]	



	40179	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[17]
	40181	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[18]
	40183	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[19]
	40185	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[20]
	40187	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[21]
	40189	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[22]
	40191	FLOAT	4	FLOAT	对应 fidbus.mtcp_ro_r[23]

表 7-3 ModbusTCP 发送端协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	40001	I16	2	BOOL	Bit0: 手动状态
					Bit1: 自动状态
					Bit2: 远程状态
					Bit3: 伺服状态
					Bit4: 报警状态
					Bit5: 急停状态
					Bit6: 程序运行状态
					Bit7: 安全位置 1 状态
					Bit8: 安全位置 2 状态
					Bit9: 安全位置 3 状态
					Bit10: 安全位置 4 状态
					Bit11: 程序加载状态
					Bit12: 伺服准备状态
40002	I16		BOOL	系统预留 BOOL 变量, 用户不可用	
40003	I16	2	I16	机器人运行速度(全局)	
40004	I16	2	I16	报警代码 1	
40005	I16	2	I16	报警代码 2	
40006	I16	2	I16	程序号 (用于反馈加载目标程序是否完成, 例: 在 Plc 端加载程序号为 2 的程序, 如果加载完成, 则程序号反馈为 2, 否则为其他值)	
40007~	I16		I16	系统预留 Int 变量, 用户不	



	40010				可用
	40011~ 40022	FLOAT	2 *	FLOAT	J1~J6 关节角度值
	40023~ 40030	FLOAT	2 *	FLOAT	J7~J10(附加轴 1-4)关节角度值
	40031~ 40033	FLOAT		FLOAT	系统预留 Float 变量，用户不可用
<b>变量类别</b>	<b>物理地址</b>	<b>单位</b>	<b>字节数</b>	<b>子单位</b>	<b>备注</b>
用户变量	40035	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_wo_b[0] ~ fidbus.mtcp_wo_b[15]
	40036	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_wo_b[16] ~ fidbus.mtcp_wo_b[31]
	40037	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_wo_b[32] ~ fidbus.mtcp_wo_b[47]
	40038	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_wo_b[48] ~ fidbus.mtcp_wo_b[63]
	40039	I16	2	I16	对应 fidbus.mtcp_wo_i[0]
	40040	I16	2	I16	对应 fidbus.mtcp_wo_i[1]
	40041	I16	2	I16	对应 fidbus.mtcp_wo_i[2]
	40042	I16	2	I16	对应 fidbus.mtcp_wo_i[3]
	40043	I16	2	I16	对应 fidbus.mtcp_wo_i[4]
	40044	I16	2	I16	对应 fidbus.mtcp_wo_i[5]
	40045	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[0]
	40047	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[1]
	40049	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[2]
	40051	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[3]
	40053	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[4]
	40055	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[5]
40057	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[6]	



	40059	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[7]
	40061	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[8]
	40063	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[9]
	40065	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[10]
	40067	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[11]
	40069	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[12]
	40071	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[13]
	40073	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[14]
	40075	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[15]
	40077	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[16]
	40079	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[17]
	40081	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[18]
	40083	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[19]
	40085	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[20]
	40087	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[21]
	40089	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[22]
	40091	FLOAT	4	FLOAT	对应 fidbus.mtcp_wo_r[23]

### 可读、可写变量

具体的地址与变量映射关系如下表 7-4 所示：

表 7-4 ModbusTCP 可读、可写部分协议：

变量类别	物理地址	单位	字节数	子单位	备注
用户变量	40301	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_rw_b[0] ~ fidbus.mtcp_rw_b[15]
	40302	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_rw_b[16] ~ fidbus.mtcp_rw_b[31]
	40303	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_rw_b[32] ~ fidbus.mtcp_rw_b[47]
	40304	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_rw_b[48] ~ fidbus.mtcp_rw_b[63]



	40305	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_rw_b[64] ~ fidbus.mtcp_rw_b[79]
	40306	I16	2	BOOL	Bit0~Bit15 对应 fidbus.mtcp_rw_b[80] ~ fidbus.mtcp_rw_b[95]
	40307	I16	2	I16	对应 fidbus.mtcp_rw_i[0]
	40308	I16	2	I16	对应 fidbus.mtcp_rw_i[1]
	...	...	.	...	...
	40405	I16	2	I16	对应 fidbus.mtcp_rw_i[98]
	40406	I16	2	I16	对应 fidbus.mtcp_rw_i[99]
	40407	FLOAT	4	FLOAT	对应 fidbus.mtcp_rw_r[0]
	40409	FLOAT	4	FLOAT	对应 fidbus.mtcp_rw_r[1]
	...	...	.	...	...
	40603	FLOAT	4	FLOAT	对应 fidbus.mtcp_rw_r[98]
	40605	FLOAT	4	FLOAT	对应 fidbus.mtcp_rw_r[99]

### 7.3.2 Profinet 功能

Profinet 数据查看使用步骤:

#	图示	说明
1		<p>在界面中选择需要查看的通讯协议的数据。</p> <p>① 点击“监控/外部/MODBUS”进入 Profinet 监控界面；</p> <p>② 点击“Profinte 只读”或“Profinet 只写”，此处以“Profinet 只读”为例。</p>



2

① 点击“Profinet 只读”后可以看到数据列表。

在数据中可以查看以下数据：

第 1 列是数据名称；  
第 2 列是数据类型；  
第 3 列是数据的值；  
第 4 列是对应的示教器程序中的变量名称。

2

② 可以通过勾选对应的数据类型 BOOL, INT16, FLOAT 框显示相应类型的数据。

## Profinet 数据输出

设置 profinet 数据输出使用步骤：

#	图示	说明
1		<p>① 点击“Profinet 只写”可看到数据输出列表；</p> <p>② 点击需要输出的变量的“值”这一列，则弹出数字键盘。以一个 Bool 变量为例。</p>



2		<p>① 在弹出的键盘中输入值;</p> <p>② 输入完成后点击“√”，完成输入。</p>
3		<p>① 输入完成后该变量值被成功写入, 该监控界面将显示为写入后的值。</p>

## Profinet 协议

Profinet 功能支持机器人做从站（slave），其地址为 2，包括 6 个 input block 和 6 个 output block，每个 block 包含 32byte，数据类型为 unsigned int（2byte）或者是 float（4byte）。

数据的存储模式为大端模式，所以当将 unsigned int 拆分成 byte 类型时，注意数据的高低位转换。

注：Profinet 功能仅支持 Robox RP2 Pro 控制器！Robox RP2 Eco 的 TPU 界面中无该显示内容。

## PLC 到机器人

PLC 到机器人，PLC 作为发送端，机器人作接收端。接收数据主要包括系统变量和 TPU 变量。其中 Bus\_Get[0]-Bus\_Get[191]为数据接收的 192BYTE。

表 7-5 Profinet 接收数据协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	Bus_Get[0]-[1]	I16	2	BOOL	Bit0: 上/下伺服（脉冲）



					Bit1: 运行程序 (脉冲)
					Bit2: 暂停程序 (脉冲)
					Bit3: 清除报警 (脉冲)
					Bit4: 加载程序 (脉冲)
					Bit5: 重新开始 (程序回到第一行) (脉冲)
					Bit6: Plc 报警 (高电平)
					Bit7: 伺服准备 确认 (脉冲)
					Bit8: 机器人位 置类型
					Bit9: 程序预约 添加确认
					Bit10: 程序预 约删除确认
					Bit11: 预约程 序启动 (脉冲)
					Bit12: 伺服使 能 (脉冲)
					Bit13: 取消伺 服使能 (脉冲)
	Bus_Get[2]-[3]	I16	2	BOOL	系统预留 BOO L 变量, 用户不 可使用
	Bus_Get[4]-[5]	I16	2	I16	运行速度 (全 局)
	Bus_Get[6]-[7]	I16	2	I16	加载目标程序 号, 例: 首先设 置目标程序号 为 2, 然后给 Bus_Get[0]-[1]



					的 Bit4 高电平触发信号, 完成程序加载(在程序运行过程中不可加载)。
	Bus_Get[8]-[9]	I16	2	I16	附加轴轴号选择(1: 七轴 2: 八轴 3: 九轴 4: 十轴)
	Bus_Get[10]-[11]	I16	2	I16	附加轴速度设定
	Bus_Get[12]-[19]	I16	2 * 4	I16	系统预留 I16 类型变量, 用户不可以使用
变量类别	物理地址	单位	字节数	子单位	备注
用户变量	Bus_Get[20]-[21]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_ro_b[0]-[15]/fidbus.pfn_ro_b[0]-[15]
	Bus_Get[22]-[23]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_ro_b[16]-[31]/fidbus.pfn_ro_b[16]-[31]
	Bus_Get[24]-[25]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_ro_b[32]-[47]/fidbus.pfn_ro_b[32]-[47]
	Bus_Get[26]-[27]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_ro_b[48]-[63]/fidbus.pfn_ro_b[48]-[63]



	Bus_Get[28]-[29]	I16	2	I16	对应 fidbus.pfb_ro_i[0]/fidbus.pfn_ro_i[0]
	Bus_Get[30]-[31]	I16	2	I16	对应 fidbus.pfb_ro_i[1]/fidbus.pfn_ro_i[1]
	Bus_Get[32]-[33]	I16	2	I16	对应 fidbus.pfb_ro_i[2]/fidbus.pfn_ro_i[2]
	Bus_Get[34]-[35]	I16	2	I16	对应 fidbus.pfb_ro_i[3]/fidbus.pfn_ro_i[3]
	Bus_Get[36]-[37]	I16	2	I16	对应 fidbus.pfb_ro_i[4]/fidbus.pfn_ro_i[4]
	Bus_Get[38]-[39]	I16	2	I16	对应 fidbus.pfb_ro_i[5]/fidbus.pfn_ro_i[5]
系统变量	Bus_Get[40]-[87]	FLOAT	4* 12	FLOAT	系统预留 FLOA 类型变量, 用户不可以使用
用户变量	Bus_Get[88]-[91]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[0]/fidbus.pfn_ro_r[0]
	Bus_Get[92]-[95]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[1]/fidbus.pfn_ro_r[1]
	Bus_Get[96]-[99]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[



					2]/fidbus.pfn_ro_r[2]
	Bus_Get[100]-[103]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[3]/fidbus.pfn_ro_r[3]
	Bus_Get[104]-[107]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[4]/fidbus.pfn_ro_r[4]
	Bus_Get[108]-[111]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[5]/fidbus.pfn_ro_r[5]
	Bus_Get[112]-[115]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[6]/fidbus.pfn_ro_r[6]
	Bus_Get[116]-[119]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[7]/fidbus.pfn_ro_r[7]
	Bus_Get[120]-[123]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[8]/fidbus.pfn_ro_r[8]
	Bus_Get[124]-[127]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[9]/fidbus.pfn_ro_r[9]
	Bus_Get[128]-[131]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[10]/fidbus.pfn_ro_r[10]
	Bus_Get[132]-[135]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[11]/fidbus.pfn_ro_r[11]



	Bus_Get[136]-[139]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[12]/fidbus.pfn_ro_r[12]
	Bus_Get[140]-[143]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[13]/fidbus.pfn_ro_r[13]
	Bus_Get[144]-[147]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[14]/fidbus.pfn_ro_r[14]
	Bus_Get[148]-[151]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[15]/fidbus.pfn_ro_r[15]
	Bus_Get[152]-[155]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[16]/fidbus.pfn_ro_r[16]
	Bus_Get[156]-[159]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[17]/fidbus.pfn_ro_r[17]
	Bus_Get[160]-[163]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[18]/fidbus.pfn_ro_r[18]
	Bus_Get[164]-[167]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[19]/fidbus.pfn_ro_r[19]
	Bus_Get[168]-[171]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[20]/fidbus.pfn_ro_r[20]
	Bus_Get[172]-[175]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[



					21]/fidbus.pfn_ro_r[21]
	Bus_Get[176]-[179]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[22]/fidbus.pfn_ro_r[22]
	Bus_Get[180]-[183]	FLOAT	4	FLOAT	对应 fidbus.pfb_ro_r[23]/fidbus.pfn_ro_r[23]
	Bus_Get[184]-[191]	\	\	\	未定义

### 机器人到 PLC

表 7-6 Profinet 发送数据协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	Bus_Set[0]-[1]	I16	2	BOOL	Bit0: 手动状态
					Bit1: 自动状态
					Bit2: 远程状态
					Bit3: 伺服状态
					Bit4: 报警状态
					Bit5: 急停状态
					Bit6: 程序运行状态
					Bit7: 安全位置 1 状态
					Bit8: 安全位置 2 状态
					Bit9: 安全位置 3 状态
					Bit10: 安全位置 4 状态
Bit11: 加载程序状态					



					Bit12: 伺服确认状态
					Bit13: 程序预约激活状态
					Bit14: 程序复位状态（程序重新开始）
	Bus_Set[2]-[3]	I16	2	BOOL	Bit0: 安全位置 5 状态
					Bit1: 安全位置 6 状态
					Bit2: 安全位置 7 状态
					Bit3: 安全位置 8 状态
	Bus_Set[4]-[5]	I16	2	I16	运行速度（全局）
	Bus_Set[6]-[7]	I16	2	I16	报警代码 1
	Bus_Set[8]-[9]	I16	2	I16	报警代码 2
	Bus_Set[10]-[11]	I16	2	I16	程序号（用于反馈加载目标程序是否完成，例：在 Plc 端加载程序号为 2 的程序，如果加载完成，则程序号反馈为 2，否则为其他值）
	Bus_Set[12]-[13]	I16	2	I16	预约程序预约状态
	Bus_Set[14]-[15]	I16	2	I16	预约程序运行状态
	Bus_Set[16]-[19]	I16	2	I16	系统预留 I16 类型变量，用户不可用
变量类别	物理地址	单位	字节数	子单位	备注



用户变量	Bus_Set[20]-[21] ]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_wo_b[0]-[15]/fidbus.pfn_wo_b[0]-[15]
	Bus_Set[22]-[23] ]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_wo_b[16]-[31]/fidbus.pfn_wo_b[16]-[31]
	Bus_Set[24]-[25] ]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_wo_b[32]-[47]/fidbus.pfn_wo_b[32]-[47]
	Bus_Set[26]-[27] ]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.pfb_wo_b[48]-[63]/fidbus.pfn_wo_b[48]-[63]
	Bus_Set[28]-[29] ]	I16	2	I16	对应 fidbus.pfb_wo_i[0]/fidbus.pfn_wo_i[0]
	Bus_Set[30]-[31] ]	I16	2	I16	对应 fidbus.pfb_wo_i[1]/fidbus.pfn_wo_i[1]
	Bus_Set[32]-[33] ]	I16	2	I16	对应 fidbus.pfb_wo_i[2]/fidbus.pfn_wo_i[2]
	Bus_Set[34]-[35] ]	I16	2	I16	对应 fidbus.pfb_wo_i[3]/fidbus.pfn_wo_i[3]
	Bus_Set[36]-[37] ]	I16	2	I16	对应 fidbus.pfb_wo_i[4]/fidbus.pfn_wo_i[4]
	Bus_Set[38]-[39] ]	I16	2	I16	对应 fidbus.pfb_wo_i[



					5]/fidbus.pfn_wo_i[5]
系统变量	Bus_Set[40]-[63]	FLOAT	4 *	FLOAT	J1~J6 关节角度值/笛卡尔空间位姿
	Bus_Set[64]-[79]	FLOAT	4 *	FLOAT	J7~J10(附加轴1-4)关节角度值
	Bus_Set[80]-[87]	FLOAT	4 *	FLOAT	系统预留 FLOAT 类型变量,用户不可用
用户变量	Bus_Set[88]-[91]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[0]/fidbus.pfn_wo_r[0]
	Bus_Set[92]-[95]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[1]/fidbus.pfn_wo_r[1]
	Bus_Set[96]-[99]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[2]/fidbus.pfn_wo_r[2]
	Bus_Set[100]-[103]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[3]/fidbus.pfn_wo_r[3]
	Bus_Set[104]-[107]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[4]/fidbus.pfn_wo_r[4]
	Bus_Set[108]-[111]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[5]/fidbus.pfn_wo_r[5]
	Bus_Set[112]-[115]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[



					6)/fidbus.pfn_wo_r[6]
	Bus_Set[116]-[119]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[7]/fidbus.pfn_wo_r[7]
	Bus_Set[120]-[123]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[8]/fidbus.pfn_wo_r[8]
	Bus_Set[124]-[127]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[9]/fidbus.pfn_wo_r[9]
	Bus_Set[128]-[131]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[10]/fidbus.pfn_w_o_r[10]
	Bus_Set[132]-[135]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[11]/fidbus.pfn_w_o_r[11]
	Bus_Set[136]-[139]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[12]/fidbus.pfn_w_o_r[12]
	Bus_Set[140]-[143]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[13]/fidbus.pfn_w_o_r[13]
	Bus_Set[144]-[147]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[14]/fidbus.pfn_w_o_r[14]
	Bus_Set[148]-[151]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[15]/fidbus.pfn_w_o_r[15]



	Bus_Set[152]-[155]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[16]/fidbus.pfn_w o_r[16]
	Bus_Set[156]-[159]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[17]/fidbus.pfn_w o_r[17]
	Bus_Set[160]-[163]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[18]/fidbus.pfn_w o_r[18]
	Bus_Set[164]-[167]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[19]/fidbus.pfn_w o_r[19]
	Bus_Set[168]-[171]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[20]/fidbus.pfn_w o_r[20]
	Bus_Set[172]-[175]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[21]/fidbus.pfn_w o_r[21]
	Bus_Set[176]-[179]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[22]/fidbus.pfn_w o_r[22]
	Bus_Set[180]-[183]	FLOAT	4	FLOAT	对应 fidbus.pfb_wo_r[23]/fidbus.pfn_w o_r[23]
	Bus_Set[184]-[191]	\	\	\	未定义



### 7.3.3 EtherCAT 功能

Modbus 数据查看使用步骤:

#	图示	说明
1		<p>在界面中选择需要查看的通讯协议的数据。</p> <p>① 点击“监控/外部/ETHERCAT”进入 EtherCAT 监控界面；</p> <p>② 点击“EtherCAT 只读”或“EtherCAT 只写”，此处以“EtherCAT 只写”为例。</p>
2		<p>① 点击“EtherCAT 只写”后可以看到数据列表。</p> <p>在数据中可以查看以下数据：</p> <ul style="list-style-type: none"> <li>第 1 列是数据名称；</li> <li>第 2 列是数据类型；</li> <li>第 3 列是数据的值；</li> <li>第 4 列是对应的示教器程序中的变量名称。</li> </ul> <p>② 可以通过勾选对应的数据类型 BOOL, INT16, FLOAT 框显示相应类型的数据。</p>



## 设置 EtherCAT 数据输出

设置 EtherCAT 数据输出使用步骤:

#	图示	说明
1		<p>① 点击“EtherCAT 只写”可看到数据输出列表;</p> <p>② 点击需要输出的变量的“值”这一列, 则弹出数字键盘。以一个 Float 变量为例。</p>
2		<p>① 在弹出的键盘中输入值;</p> <p>② 输入完成后点击“√”, 完成输入。</p>
3		<p>① 输入完成后该变量值被成功写入, 该监控界面将显示为写入后的值。</p>

## EtherCAT 协议

目前机器人仅支持 EtherCAT 协议, 用户可通过 EtherCAT 协议转为 CCLink 等其他协议, EtherCAT 功能分为数据接收端与发送端, 两端各留有 64 个 BOOL, 6 个 int, 24 个 float 数据接口, 终端用户可以通过示教器编写程序读写主站 PLC 的数据。



注：ANYBUS\_ECAT 是通用的网关协议转换功能，硬件需安装 ANYBUS 网关

## PLC 到机器人

PLC 到机器人，PLC 作为发送端，机器人作接收端。接收数据主要包括系统变量和 TPU 变量。其中 Bus\_Get[0]-Bus\_Get[191]为数据接收的 192BYTE。

表 7-7 EtherCAT 接收数据协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	Bus_Get[0]-[1]	I16	2	BOOL	Bit0: 上/下伺服 (脉冲)
					Bit1: 运行程序 (脉冲)
					Bit2: 暂停程序 (脉冲)
					Bit3: 清除报警 (脉冲)
					Bit4: 加载程序 (脉冲)
					Bit5: 重新开始 (程序回到第一行) (脉冲)
					Bit6: Plc 报警(高电平)
					Bit7: 伺服准备确认 (脉冲)
					Bit8: 机器人位置类型
					Bit9: 程序预约添加确认
					Bit10: 程序预约删除确认
					Bit11: 预约程序启动 (脉冲)
Bit12: 伺服使能 (脉冲)					



					Bit13: 取消伺服使能 (脉冲)
	Bus_Get[2]-[3]	I16	2	BOOL	Bit0: 安全位置 5 状态
					Bit1: 安全位置 6 状态
					Bit2: 安全位置 7 状态
					Bit3: 安全位置 8 状态
	Bus_Get[4]-[5]	I16	2	I16	运行速度 (全局)
	Bus_Get[6]-[7]	I16	2	I16	报警代码 1
	Bus_Get[8]-[9]	I16	2	I16	报警代码 2
	Bus_Get[10]-[11]	I16	2	I16	程序号 (用于反馈加载目标程序是否完成, 例: 在 Plc 端加载程序号为 2 的程序, 如果加载完成, 则程序号反馈为 2, 否则为其他值)
	Bus_Get[12]-[19]	I16	8	I16	预约程序预约状态
<b>变量类别</b>	<b>物理地址</b>	<b>单位</b>	<b>字节数</b>	<b>子单位</b>	<b>备注</b>
用户变量	Bus_Get[20]-[21]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_ro_b[0]-[15]
	Bus_Get[22]-[23]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_ro_b[16]-[31]
	Bus_Get[24]-[25]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_ro_b[32]-[47]



	Bus_Get[26]-[27] ]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_ro _b[48]-[63]
	Bus_Get[28]-[29] ]	I16	2	I16	对应 fidbus.ethcat_ro _i[0]
	Bus_Get[30]-[31] ]	I16	2	I16	对应 fidbus.ethcat_ro _i[1]
	Bus_Get[32]-[33] ]	I16	2	I16	对应 fidbus.ethcat_ro _i[2]
	Bus_Get[34]-[35] ]	I16	2	I16	对应 fidbus.ethcat_ro _i[3]
	Bus_Get[36]-[37] ]	I16	2	I16	对应 fidbus.ethcat_ro _i[4]
	Bus_Get[38]-[39] ]	I16	2	I16	对应 fidbus.ethcat_ro _i[5]
系统变量	Bus_Get[40]-[87] ]	FLOAT	4* 12	FLOAT	系统预留 FLOAT 类型变 量, 用户不可以 使用
用户变量	Bus_Get[88]-[91] ]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[0]
	Bus_Get[92]-[95] ]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[1]
	Bus_Get[96]-[99] ]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[2]
	Bus_Get[100]-[1 03]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[3]



Bus_Get[104]-[107]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[4]
Bus_Get[108]-[111]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[5]
Bus_Get[112]-[115]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[6]
Bus_Get[116]-[119]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[7]
Bus_Get[120]-[123]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[8]
Bus_Get[124]-[127]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[9]
Bus_Get[128]-[131]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[10]
Bus_Get[132]-[135]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[11]
Bus_Get[136]-[139]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[12]
Bus_Get[140]-[143]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[13]
Bus_Get[144]-[147]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[14]
Bus_Get[148]-[151]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro _r[15]
Bus_Get[152]-[155]	FLOAT	4	FLOAT	对应



	55]				fidbus.ethcat_ro_r[16]
	Bus_Get[156]-[159]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro_r[17]
	Bus_Get[160]-[163]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro_r[18]
	Bus_Get[164]-[167]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro_r[19]
	Bus_Get[168]-[171]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro_r[20]
	Bus_Get[172]-[175]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro_r[21]
	Bus_Get[176]-[179]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro_r[22]
	Bus_Get[180]-[183]	FLOAT	4	FLOAT	对应 fidbus.ethcat_ro_r[23]
	Bus_Get[184]-[191]	\	\	\	未定义

## 机器人到 PLC

机器人到 PLC，机器人作为发送端，PLC 作为接收端。发送数据主要包括系统变量和 TPU 变量。其中 Bus\_Set[0]-Bus\_Set[191]为数据发送的 192BYTE。

表 7-8 EtherCAT 发送数据协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	Bus_Set[0]-[1]	I16	2	BOOL	Bit0: 手动状态
					Bit1: 自动状态
					Bit2: 远程状态
					Bit3: 伺服状态



					Bit4: 报警状态
					Bit5: 急停状态
					Bit6: 程序运行状态
					Bit7: 安全位置 1 状态
					Bit8: 安全位置 2 状态
					Bit9: 安全位置 3 状态
					Bit10: 安全位置 4 状态
					Bit11: 加载程序状态
					Bit12: 伺服确认状态
					Bit13: 程序预约激活状态
					Bit14: 程序复位状态 (程序重新开始)
	Bus_Set[2]-[3]	I16	2	BOOL	Bit0: 安全位置 5 状态
					Bit1: 安全位置 6 状态
					Bit2: 安全位置 7 状态
					Bit3: 安全位置 8 状态
	Bus_Set[4]-[5]	I16	2	I16	运行速度 (全局)
	Bus_Set[6]-[7]	I16	2	I16	报警代码 1
	Bus_Set[8]-[9]	I16	2	I16	报警代码 2
	Bus_Set[10]-[11]	I16	2	I16	程序号 (用于反馈加载目标程序是否完成, 例: 在 Plc 端加载程序号为 2 的程序, 如果加载完成, 则程序号反



					馈为 2，否则为其他值)
	Bus_Set[12]-[13]	I16	2	I16	预约程序预约状态
	Bus_Set[14]-[15]	I16	2	I16	预约程序运行状态
	Bus_Set[16]-[19]	I16	2	I16	系统预留 I16 类型变量，用户不可用
变量类别	物理地址	单位	字节数	子单位	备注
用户变量	Bus_Set[20]-[21]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_wo_b[0]-[15]
	Bus_Set[22]-[23]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_wo_b[16]-[31]
	Bus_Set[24]-[25]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_wo_b[32]-[47]
	Bus_Set[26]-[27]	I16	2	BOOL	Bit0~Bit15 对应 fidbus.ethcat_wo_b[48]-[63]
	Bus_Set[28]-[29]	I16	2	I16	对应 fidbus.ethcat_wo_i[0]
	Bus_Set[30]-[31]	I16	2	I16	对应 fidbus.ethcat_wo_i[1]
	Bus_Set[32]-[33]	I16	2	I16	对应 fidbus.ethcat_wo_i[2]
	Bus_Set[34]-[35]	I16	2	I16	对应 fidbus.ethcat_wo_i[3]
	Bus_Set[36]-[37]	I16	2	I16	对应 fidbus.ethcat_wo_i[4]



					4]
	Bus_Set[38]-[39]	I16	2	I16	对应 fidbus.ethcat_wo_i[5]
系统变量	Bus_Set[40]-[63]	FLOAT	4*6	FLOAT	J1~J6 关节角度值/ 笛卡尔空间位姿
	Bus_Set[64]-[79]	FLOAT	4*4	FLOAT	J7~J10(附加轴 1-4) 关节角度值
	Bus_Set[80]-[87]	FLOAT	4*2	FLOAT	系统预留 FLOAT 类型变量,用户不可用
用户变量	Bus_Set[88]-[91]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[0]
	Bus_Set[92]-[95]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[1]
	Bus_Set[96]-[99]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[2]
	Bus_Set[100]-[103]	FLOAT	4	FLOAT	对应 idbus.ethcat_wo_r[3]
	Bus_Set[104]-[107]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[4]
	Bus_Set[108]-[111]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[5]
	Bus_Set[112]-[115]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[6]
	Bus_Set[116]-[119]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[7]



Bus_Set[120]-[1 23]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 8]
Bus_Set[124]-[1 27]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 9]
Bus_Set[128]-[1 31]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 10]
Bus_Set[132]-[1 35]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 11]
Bus_Set[136]-[1 39]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 12]
Bus_Set[140]-[1 43]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 13]
Bus_Set[144]-[1 47]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 14]
Bus_Set[148]-[1 51]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 15]
Bus_Set[152]-[1 55]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 16]
Bus_Set[156]-[1 59]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 17]
Bus_Set[160]-[1 63]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 18]
Bus_Set[164]-[1 67]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[ 19]
Bus_Set[168]-[1	FLOAT	4	FLOAT	对应



	71]				fidbus.ethcat_wo_r[20]
	Bus_Set[172]-[175]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[21]
	Bus_Set[176]-[179]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[22]
	Bus_Set[180]-[183]	FLOAT	4	FLOAT	对应 fidbus.ethcat_wo_r[23]
	Bus_Set[184]-[191]	\	\	\	未定义

### 7.3.4 EthernetIP 功能

EthernetIP 数据查看使用步骤:

#	图示	说明
1		<p>在界面中选择需要查看的通讯协议的数据。</p> <p>① 点击“监控/外部/ETHERNETIP”进入 EthernetIP 监控界面；</p> <p>② 点击“EthernetIP 只读”或“EthernetIP 只写”，此处以“EthernetIP 只读”为例。</p>



2

① 点击“EthernetIP 只读”后可以看到数据列表。

在数据中可以查看以下数据：

第 1 列是数据名称；  
第 2 列是数据类型；  
第 3 列是数据的值；  
第 4 列是对应的示教器程序中的变量名称。

② 可以通过勾选对应的数据类型 BOOL, INT16, FLOAT 框显示相应类型的数据。

## 设置 EthernetIP 数据输出

设置 EthernetIP 数据输出使用步骤：

#	图示	说明
1		<p>① 点击“EthernetIP 只写”可看到数据输出列表；</p> <p>② 点击需要输出的变量的“值”这一列，则弹出数字键盘。以一个 Bool 变量为例。</p>



2		<p>① 在弹出的键盘中输入值;</p> <p>② 输入完成后点击“√”，完成输入。</p>
3		<p>① 输入完成后该变量值被成功写入, 该监控界面将显示为写入后的值。</p>

## EthernetIP 协议

Ethernet/IP 功能支持机器人做从站 (slave)，其 IP 地址与控制器一致，控制器默认站号为 0，端口号默认为 2222，数据的存储模式为小端模式。Ethernet/IP 功能分为数据接收端与发送端，两端各留有 64 个 BOOL, 6 个 int, 24 个 float 数据接口，终端用户可以通过示教器编写程序读写主站 PLC 的数据。

## PLC 到机器人

PLC 到机器人，PLC 作为发送端，机器人作接收端。接收数据主要包括系统变量和 TPU 变量。其中 Bus\_Get[0]-Bus\_Get[191]为数据接收为 192BYTE。

表 7-9 EthernetIPt 接收数据协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	Bus_Get[0]	I16	1	BOOL	Bit0: 上/下伺服 (脉冲)
					Bit1: 运行程序 (脉冲)
					Bit2: 暂停程序 (脉冲)



					Bit3: 清除报警 (脉冲)
					Bit4: 加载程序 (脉冲)
					Bit5: 重新开始 (程序回到第一行) (脉冲)
					Bit6: Plc 报警 (高电平)
					Bit7: 伺服准备 确认 (脉冲)
					Bit8: 机器人位置 类型
					Bit9: 程序预约 添加确认
					Bit10: 程序预 约删除确认
					Bit11: 预约程 序启动 (脉冲)
					Bit12: 伺服使 能 (脉冲)
					Bit13: 取消伺 服使能 (脉冲)
	Bus_Get[1]	I16	1	BOOL	系统预留 BOOL 变量, 用户不可 使用
	Bus_Get[2]	I16	1	I16	运行速度 (全 局)
	Bus_Get[3]	I16	1	I16	加载目标程序 号, 例: 首先设 置目标程序号 为 2, 然后给 Bus_Get[0]的 Bit4 高电平触 发信号, 完成程 序加载 (在程序 运行过程中不可



					加载)。
	Bus_Get[4]	I16	1	I16	附加轴轴号选择(1: 七轴 2: 八轴 3: 九轴 4: 十轴)
	Bus_Get[5]	I16	1	I16	附加轴速度设定
	Bus_Get[6]-[9]	I16	1*4	I16	系统预留 I16 类型变量, 用户不可以使用
	Bus_Get[10]-[33]	FLOAT	2*12	FLOAT	系统预 FLOAT 类型变量, 用户不可以使用
<b>变量类别</b>	<b>物理地址</b>	<b>单位</b>	<b>字节数</b>	<b>子单位</b>	<b>备注</b>
用户变量	Bus_Get[34]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[0]-[15]
	Bus_Get[35]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[16]-[31]
	Bus_Get[36]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[32]-[47]
	Bus_Get[37]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[48]-[63]
	Bus_Get[38]	I16	1	I16	对应 fidbus.eip_wo_i[0]
	Bus_Get[39]	I16	1	I16	对应 fidbus.eip_wo_i[



					1]
	Bus_Get[40]	I16	1	I16	对应 fidbus.eip_wo_i[ 2]
	Bus_Get[41]	I16	1	I16	对应 fidbus.eip_wo_i[ 3]
	Bus_Get[42]	I16	1	I16	对应 fidbus.eip_wo_i[ 4]
	Bus_Get[43]	I16	1	I16	对应 fidbus.eip_wo_i[ 5]
	Bus_Get[44]-[4 5]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 0]
	Bus_Get[46]-[4 7]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 1]
	Bus_Get[48]-[4 9]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 2]
	Bus_Get[50]-[5 1]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 3]
	Bus_Get[52]-[5 3]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 4]
	Bus_Get[54]-[5 5]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 5]
	Bus_Get[56]-[5 7]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 6]
	Bus_Get[58]-[5 9]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[ 7]



Bus_Get[60]-[61]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[8]
Bus_Get[62]-[63]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[9]
Bus_Get[64]-[65]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[10]
Bus_Get[66]-[67]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[11]
Bus_Get[68]-[69]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[12]
Bus_Get[70]-[71]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[13]
Bus_Get[72]-[73]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[14]
Bus_Get[74]-[75]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[15]
Bus_Get[76]-[77]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[16]
Bus_Get[78]-[79]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[17]
Bus_Get[80]-[81]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[18]
Bus_Get[82]-[83]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[19]
Bus_Get[84]-[85]	FLOAT	2	FLOAT	对应



	5]				fidbus.eip_wo_r[20]
	Bus_Get[86]-[87]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[21]
	Bus_Get[88]-[89]	FLOAT	2	FLOAT	对应 fidbus.eip_wo_r[22]
	Bus_Get[90]-[91]	FLOAT	4	FLOAT	对应 fidbus.eip_wo_r[23]

### 机器人到 PLC

机器人到 PLC，机器人作为发送端，PLC 作为接收端。发送数据主要包括系统变量和 TPU 变量。其中 Bus\_Set[0]-Bus\_Set[191]为数据发送的 192BYTE。

表 7-10 EthernetIP 发送数据协议

变量类别	物理地址	单位	字节数	子单位	备注
系统变量	Bus_Set[0]	I16	1	BOOL	Bit0: 手动状态
					Bit1: 自动状态
					Bit2: 远程状态
					Bit3: 伺服状态
					Bit4: 报警状态
					Bit5: 急停状态
					Bit6: 程序运行状态
					Bit7: 安全位置 1 状态
					Bit8: 安全位置 2 状态
					Bit9: 安全位置 3 状态
					Bit10: 安全位置 4 状态
					Bit11: 加载程序状态



					Bit12: 伺服确认状态
					Bit13: 程序预约激活状态
					Bit14: 程序复位状态(程序重新开始)
	Bus_Set[1]	I16	1	BOOL	Bit0: 安全位置5状态
					Bit1: 安全位置6状态
					Bit2: 安全位置7状态
					Bit3: 安全位置8状态
	Bus_Set[2]	I16	1	I16	运行速度(全局)
	Bus_Set[3]	I16	1	I16	报警代码
	Bus_Set[4]	I16	1	I16	报警代码
	Bus_Set[5]	I16	1	I16	程序号(用于反馈加载目标程序是否完成, 例: 在Plc端加载程序号为2的程序, 如果加载完成, 则程序号反馈为2, 否则为其他值)
	Bus_Set[6]	I16	1	I16	预约程序预约状态
	Bus_Set[7]	I16	1	I16	预约程序运行状态
	Bus_Set[8]-[9]	I16	1*2	I16	系统预留I16类型变量, 用户不可用
	Bus_Set[10]-[21]	FLOAT	2*6	FLOAT	J1~J6 关节角度值/笛卡尔空间



					位姿
	Bus_Set[22]-[31]	FLOAT	2*4	FLOAT	J7~J10(附加轴1-4)关节角度值
	Bus_Set[32]-[33]	FLOAT	2	FLOAT	系统预留 FLOAT 类型变量,用户不可用
<b>变量类别</b>	<b>物理地址</b>	<b>单位</b>	<b>字节数</b>	<b>子单位</b>	<b>备注</b>
用户变量	Bus_Set[34]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[0]-[15]
	Bus_Set[35]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[16]-[31]
	Bus_Set[36]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[32]-[47]
	Bus_Set[37]	I16	1	BOOL	Bit0~Bit15 对应 fidbus.eip_wo_b[48]-[63]
	Bus_Set[38]	I16	1	I16	对应 fidbus.eip_wo_i[0]
	Bus_Set[39]	I16	1	I16	对应 fidbus.eip_wo_i[1]
	Bus_Set[40]	I16	1	I16	对应 fidbus.eip_wo_i[2]
	Bus_Set[41]	I16	1	I16	对应 fidbus.eip_wo_i[3]



Bus_Set[42]	I16	1	I16	对应 fidbus.eip_wo_i [4]
Bus_Set[43]	I16	1	I16	对应 fidbus.eip_wo_i [5]
Bus_Set[44]-[45] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[0]
Bus_Set[46]-[47] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[1]
Bus_Set[48]-[49] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[2]
Bus_Set[50]-[51] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[3]
Bus_Set[52]-[53] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[4]
Bus_Set[54]-[55] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[5]
Bus_Set[56]-[57] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[6]
Bus_Set[58]-[59] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[7]
Bus_Set[60]-[61] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[8]
Bus_Set[62]-[63] ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[9]
Bus_Set[64]-[65]	FLOAT	2	FLOAT	对应



]					fidbus.ethcat_w o_r[10]
Bus_Set[66]-[67] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[11]
Bus_Set[68]-[69] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[12]
Bus_Set[70]-[71] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[13]
Bus_Set[72]-[73] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[14]
Bus_Set[74]-[75] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[15]
Bus_Set[76]-[77] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[16]
Bus_Set[78]-[79] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[17]
Bus_Set[80]-[81] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[18]
Bus_Set[82]-[83] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[19]
Bus_Set[84]-[85] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[20]
Bus_Set[86]-[87] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w o_r[21]
Bus_Set[88]-[89] ]	FLOAT	2	FLOAT	对应	fidbus.ethcat_w



					o_r[22]
	Bus_Set[90]-[91 ]	FLOAT	2	FLOAT	对应 fidbus.ethcat_w o_r[23]



# 第 8 章 日志选项卡

日志选项卡中可查看警告、报警详细信息，系统日志，程序执行日志。

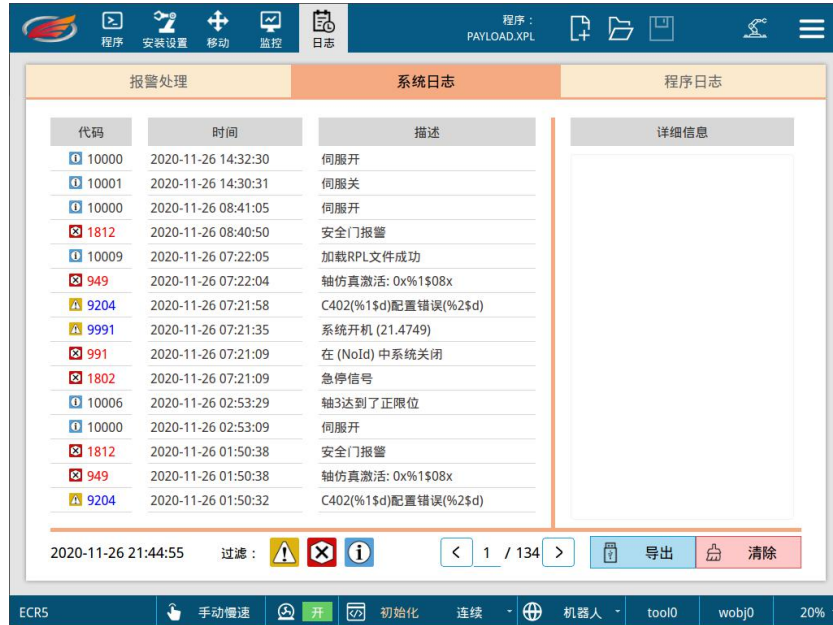


图 8-1 日志选项卡

## 8.1 报警处理

用户在使用机器人时发生报警，引起机器人保护性停止，需要在处理完导致报警的原因后，点击“处理核查”按钮，这时控制器会自行检查所有报警项并且核对状态是否恢复正常。在该界面中没有任何一条报警条后，机器人可恢复使用。



图 8-2 报警处理页面



## 8.2 系统日志

在系统日志界面，用户可查看包括报警、警告、信息的历史记录。界面中的过滤按钮可对信息类型进行筛选。还提供信息的导出、清除操作。



图 8-3 系统日志页面

## 8.3 程序日志

程序日志中记录了所有关于程序加载、程序执行等操作信息。



图 8-4 程序日志页面



### 提示！

程序日志中“清屏”按钮，不是清除日志的操作，而是将屏幕中的信息清空，而这些信息仍保存在内存中。



# 第 9 章 HBB 菜单

HBB 菜单中包括：关于页面、系统设置页面、帮助页面，如下图 9-1。



图 9-1 HBB 菜单

## 9.1 关于

关于页面中主要涵盖机器人类型、软件版本号。



图 9-2 关于页面



## 9.2 系统设置

系统设置页面中，用户可以做如下设置：

- (1) 切换系统语言；
- (2) 设置控制器/示教器 IP 地址；
- (3) 更改“安装设置”选项卡中的“安全”功能页的解锁密码，即“安全密码”；
- (4) 重置“安全密码”（忘记安全密码时使用）。

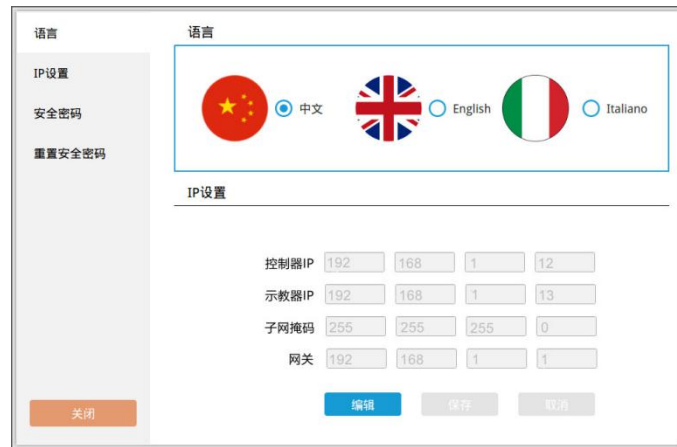


图 9-3 系统设置页面

	<p><b>提示！</b></p> <p>修改 IP、安全密码功能，只在 Engineer、Administrator 用户登录时可用。</p> <p>重置安全密码功能，仅 Administrator 用户登录时可用。</p>
--	---

## 9.3 帮助

帮助页面中介绍包括：示教器面板信息、功能引导等内容，如下图 9-4：



图 9-4 帮助页面





# 附录 A 报警及警告代码表

## A.1 系统报警代码

报警代码	报警内容(英文)	报警内容(中文)	原因	解决方法
1	User Registers loss	可保持用户寄存器值异常	可保持用户寄存器读取值与上次关机前不一致	如果在控制器通电时发生警报, 请持续按压控制器上的 ADV 红色按钮, 并重启。
2	Parameter Registers loss	可保持参数寄存器值异常	可保持参数寄存器读取值与上次关机前不一致	如果在控制器通电时发生警报, 请持续按压控制器上的 ADV 红色按钮, 并重启。
3	Alarm History loss	系统报警信息异常	系统报警信息与上次关机前不一致	如果在控制器通电时发生警报, 请持续按压控制器上的 ADV 红色按钮, 并重启。
10	#nn Transducer Fault (code)	编码器连接异常	编码器故障	检查编码器和 Robox 控制器之间的接线。
11	#nn Following Error (Servo system)	跟随误差异常	目标位置 and 实际位置之间的差超过了最大允许值	<ol style="list-style-type: none"> <li>1. 调整驱动器 PID 参数或者最大允许误差;</li> <li>2. 检查机器人目标位置与当前位置是否合理(距离太远);</li> <li>3. 降低机器人运动参数(速度, 加速度, 加加速度);</li> <li>4. 检查控制器和驱动器的接线、驱</li> </ol>



				动器和伺服电机的接线。
14	#nn Lower E. E.O.S reached	到达负限位	目标位置超出设置的最小限位	/
15	#nn Upper E. E.O.S reached	到达正限位	目标位置超出设置的最大限位	/
21	#nn Transducer is NOT yet ready	编码器连接异常	/	1. 检查编码器和控制器之间的接线; 2. 检查总线通讯。
82	ECAT(int ) No Communic.(code )	总线通讯中断	EtherCAT 通信诊断	检查总线通讯
83	ECAT(int) Lost communication with lost nodes out of total	丢失节点, 总线通讯中断	EtherCAT 通信诊断	检查总线通讯
90	Lost SYNCH. (Rule length similar to SI) period	系统同步周期异常	/	1. 重启控制器; 2. 检查外部 PLC 等设备是否有干扰; 3. 联系厂家。
804	R3 task load failure <tt>. Use REPORT for details	task 加载失败	/	1. 重启控制器; 2. 联系厂家。
807	OB load failure <%d>. Use REPORT for d	OB 加载失败	/	1. 重启控制器; 2. 联系厂家。



	etails			
984	Illegal transducer definition in RTE.CFG	RTE.CFG 文件中编码器定义非法	编码器配置参数错误	检查工程配置中或 RTE.CFG 文件中的编码器定义
985	Error in RTE.CFG file	RTE.CFG 文件错误	/	/
988	Bad RHW.CFG code=code	RHW.CFG IO 模块等硬件配置错误	/	1. 检查 IO 模块等硬件设备是否连接正常； 2. 前往设置->IO 配置页面，自动更新匹配硬件信息。
990	! System LOCKED ! task <t> step <ss>	系统锁定	/	1. 重启控制器； 2. 检查外部 PLC 等设备是否有干扰； 3. 联系厂家。
999	Error in LICENSE.lic. Use REPORT for details	LICENSE 文件异常	/	请检查 CF 卡 KEY 文件夹内的 LICENSE 是否正常



## A.2 用户报警代码（1400~1499）

报警代码	FLAGS	报警信息	报警原因	解决办法
1400	0x22000000	轴%d 0x2250 驱动器短路	1.驱动器输出接线短路； 2.驱动器损坏。	1.检查驱动器接线； 2.更换驱动器。
1401	0x22000000	轴%d 0x2341 驱动器 UV 短路	驱动器损坏。	维修或更换驱动器。
1402	0x22000000	轴%d 0x2342 驱动器 VW 短路	驱动器损坏。	维修或更换驱动器。
1403	0x22000000	轴%d 0x5210 AD采样电路 异常	驱动器损坏。	维修或更换驱动器。
1404	0x22000000	轴%d 0x5530 EEPROM 异 常	参数CRC校验错 误。	检查驱动器硬件。
1405	0x22000000	轴%d 0xFF28 栈空间溢出	驱动器固件运行 错误。	维修或更换驱动器。
1406	0x22000000	轴%d 0x6310 参数未初始 化	驱动器损坏。	维修或更换驱动器。
1407	0x22000000	轴%d 0xFF04 编码器数据 异常	1.编码器接线错 误； 2.编码器线缆损 坏； 3.编码器损坏。	1.检查编码器接线； 2.检查编码器线缆； 3.更换编码器。
1408	0x22000000	轴%d 0xFF07 转子定位错 误	1.电机转子位置 补偿角设置错 误； 2.驱动器损坏。	1.重新检测电机转子位 置补偿角； 2.维修或更换驱动器。
1409	0x22000000	轴%d 0xFF1 A 电机接线 相序错误	电机相序接线错 误。	检测电机接线相序。



1410	0x22000000	轴%d 0xFF00 系统初始化失败	1.伺服参数设置错误; 2.电机编码器错误; 3.驱动器损坏。	1.检查伺服参数; 2.检查电机编码器; 3.维修或更换驱动器。
1411	0x22000000	轴%d 0xFF2B 编码器内部通信异常	1.编码器接线错误; 2.编码器连线损坏; 3.编码器损坏。	1.检查编码器接线; 2.检查编码器连线; 3.更换编码器。
1412	0x22000000	轴%d 0xFF03 目标位置偏差过大	目标位置偏差过大	1、检查通讯周期 2、检查位置指令
1413	0x22000000	轴%d 0xFF05 编码器内部故障	1.编码器接线错误; 2.编码器连线损坏; 3.编码器损坏。	1.检查编码器接线; 2.检查编码器连线; 3.更换编码器。
1414	0x22000000	轴%d 0xFF3D 增量式编码器故障	1.码盘受污染; 2.编码器损坏。	1.清理码盘; 2.更换编码器。
1415	0x22000000	轴%d 0xFF3E 绝对式编码器故障	1.编码器连线错误; 2.编码器连线损坏; 3.编码器损坏; 4.负载太大。	1.检查编码器接线; 2.检查编码器连线; 3.更换编码器; 4.检查负载。
1416	0x22000000	轴%d 0x2310 驱动器过流 U	1.驱动器 U 相输出短路; 2.电机负载过大; 3.驱动器损坏。	1.检测 U 相接线; 2.降低电机负载; 3.维修或更换驱动器。
1417	0x22000000	轴%d 0x2311 驱动器过流 V	1.驱动器 V 相输出短路; 2.电机负载过大; 3.驱动器损坏。	1.检测 V 相接线; 2.降低电机负载; 3.维修或更换驱动器。
1418	0x22000000	轴%d 0x3210 直流母线过压	1.电源电压过高; 2.制动电阻阻值过大。	1.检查电源电压; 2.选择合适电阻。



1419	0x22000000	轴%d 0xFF02 驱动器持续过载	1.机械卡阻; 2.驱动器负载过大; 3.电机故障; 4.驱动器故障;	1.检查机械传动部分,改善机械传动性能; 2.检查电机负载,或加大电机驱动模块容量; 3.维修或更换电机; 4.维修或更换驱动器。
1420	0x22000000	轴%d 0x6010 看门狗溢出	内部堆栈溢出。	1.重新上电; 2.维修或更换驱动器。
1421	0x22000000	轴%d 0xFF29 CPU 过载	1.控制指令超过CPU 负载能力; 2.驱动器损坏。	1.降低控制指令操作频率; 2.更换或维修驱动器。
1422	0x22000000	轴%d 0xFF15 电机动力线断开	1.电机动力线断线; 2.电机损坏; 3.驱动器损坏。	1.检查电机动力线接线; 2.更换或维修电机; 3.更换或维修驱动器。
1423	0x22000000	轴%d 0xFF25 驱动器瞬时过载	1.输出侧短路; 2.因干扰误动作; 3.控制参数不合理; 4.驱动器损坏。	1.检查输出侧电缆接线; 2.接线可靠接地; 3.重新调整控制参数; 4.维修或更换驱动器。
1424	0x22000000	轴%d 0xFF2C 编码器外部通信发送异常	1.编码器连线错误; 2.编码器连线损坏; 3.编码器损坏。	1.检查编码器接线; 2.检查编码器连线; 3.更换编码器。
1425	0x22000000	轴%d 0xFF2F 编码器外部通信接收异常	1.编码器连线错误; 2.编码器连线损坏; 3.编码器损坏。	1.检查编码器接线; 2.检查编码器连线; 3.更换编码器。
1426	0x22000000	轴%d 0x2320 驱动器硬件过流	1.机械卡组; 2.电机负载过大; 3.电机参数或控制参数设置不正确; 4.电机故障; 5.驱动器故障。	1.检查机械传动部分,包括电机抱闸,改善机械传动性能; 2.检查电机负载; 3.检查电机参数和控制参数设置; 4.维修或更换电机; 5.维修或更换驱动器。



1427	0x22000000	轴%d 0x3220 直流母线欠压	电源电压过低。	检查电源电压。
1428	0x22000000	轴%d 0x4110 环境温度过高	环境温度过高。	保持环境温度正常。
1429	0x22000000	轴%d 0x4210 逆变功率模块过热	1.环境温度过热; 2.驱动器损坏。	1.保持环境温度正常; 2.维修或更换驱动器。
1430	0x22000000	轴%d 0x8311 电机持续过载	1.机械卡阻; 2.超过电机额定转矩运行时间过长。	1.检查机械传动部分,查看是否有堵转现象; 2.检查负载,降低加减速度,或更换更大容量的驱动器和电机。
1431	0x22000000	轴%d 0x6320 参数数据异常	1.参数范围超限; 2.位置单位设置错误。	1.检查参数设置是否超出设定的参数范围; 2.检查位置单位设置。
1432	0x22000000	轴%d 0x8611 位置跟随误差过大	1.编码器接线错误或连接器接触不良; 2.控制器参数不合适; 3.外部负载波动或干扰过大。	1.检查编码器连线; 2.重新调整控制参数; 3.增加抗干扰措施。
1433	0x22000000	轴%d 0x8800 位置控制溢出	反馈位置或指令位置超过 32 位有符号数。	编码器清零后软复位或重启驱动器。
1434	0x22000000	轴%d 0xFF18 速度跟随误差过大	1.编码器接线错误或连接器接触不良; 2.控制参数不合适; 3.外部负载波动或干扰过大。	1.检查编码器连线; 2.重新调整控制参数; 3.增加抗干扰措施。
1435	0x22000000	轴%d 0xFF1E 写 EEPROM 失败	驱动器损坏。	维修或更换驱动器。



1436	0x22000000	轴%d 0xFF16 正向软限位	位置反馈值超过(正向软限位值+定位完成阈)。	1.如果不需要正向软限位功能, 可通过参数 0x2000 禁止; 2.检查位置指令规划范围。
1437	0x22000000	轴%d 0xFF17 负向软限位	位置反馈值超过(负向软限位值-定位完成阈)。	1.如果不需要反向软限位功能, 可通过参数 0x2000 禁止; 2.检查位置指令规划范围。
1438	0x22000000	轴%d 0xFF23 上电位置偏差过大	驱动器上电时, 与上一次掉电保存位置不一, 超过设定阈值。	1.检查机械位置是否改变, 确认机械零; 2.点无异常后可清除。
1439	0x22000000	轴%d 0xFF26 非法更改伺服参数	修改伺服参数超过了限制值。	在伺服参数可修改范围内修改参数值。
1440	0x22000000	轴%d 0xFF0B 电机超速	1.反馈速度超过预设速度, 误差超过设定阈值; 2.编码器异常。	1.优化电机参数和控制参数; 2.检查编码器设置和编码器接线。
1441	0x22000000	轴%d 0xFF08 断开释放失败	1.机械卡阻; 2.驱动器负载过大; 3.电机故障; 4.驱动器故障。	检查 2002h 碰撞力矩阈值。
1442	0x22000000	轴%d 0x8130 CAN 通讯心跳异常	can 通讯异常	检查 Robox 控制器与驱动器之间连接线
1443	0x22000000	轴%d 0x2312 驱动器过流 U	1.驱动器 W 相输出短路; 2.电机负载过大; 3.驱动器损坏。	1.检测 W 相接线; 2.降低电机负载; 3.维修或更换驱动器。
1444	0x22000000	轴%d 0X7500 Can 通讯失败	Can 节点关闭	1、检查 120 匹配电阻; 2、检查 can 线; 3、检查干扰源



1497	0x22000000	CANOpen 虚拟节点不存在，请检查接线并重启电柜	CANOpen 虚拟节点不存在	检查 Robox 控制器与 I/O 板之间的 CANOpen 接线
1498	0x22000000	CANOpen 通信故障	1.机器人本体断电； 2.CANOpen 节点丢失。	1.检查本体是否断电； 2.检查 CANOpen 通讯节点是否丢失。
1499	0x22000000	轴%d 未记录的警报	驱动器未收录报警。	驱动器未收录报警。



### A.3 重大错误代码（1800~1899）

报警代码	FLAGS	报警信息	报警原因	解决办法
1800	0x220000 00	关节运动错误 (代码=%d)	关节运动错误。	1.代码=1：无效轴组。检查控制器轴组设置； 2.代码=2：无效目标点。检查目标点是否合理； 3.代码=3：无效的速度和加速度参数。请重新设置相关参数； 4.代码=4：目标点超出关节运动范围。请重新设置目标点位置； 5.代码=5：运动队列已满。重试或检查控制器； 6.代码=6：运动轴组未执行(3)。重试或检查控制器。
1801	0x220000 00	直线运动错误 (代码=%d)	直线运动错误。	1.代码=1：无效轴组。检查控制器轴组设置； 2.代码=2：无效目标点。检查目标点是否合理； 3.代码=3：无效的速度和加速度参数。请重新设置相关参数； 4.代码=4：目标点超出关节运动范围。请重新设置目标点位置； 5.代码=5：运动队列已满。重试或检查控制器； 6.代码=6：运动轴组未执行(3)。重试或检查控制器。
1802	0x220000 00	急停信号	急停按钮被按下。	请检查示教器和电柜上的急停按钮是否被松开。
1803	0x220000 00	配置错误	配置内容出错。	检查配置的内容重新配置。



1804	0x220000 00	IO 模块配置错误, 请前往设置界面修改	IO 模块配置错误。	请前往 IO 设置界面修改。
1805	0x220000 00	控制柜过热报警	控制柜过热报警。	1.检查控制柜内风扇、温控开关是否正常工作; 2.温控开关值设置是否正确。
1807	0x220000 00	区域违反报警	区域违反报警。	1.将机器人置于手动模式; 2.关闭区域监控的控制命令; 3.报警复位; 4.手动 jog 移动机器人至非违反状态。
1809	0x220000 00	附加轴%d 驱动器报警%x	附加轴驱动器报警。	请查阅驱动器报警手册。
1810	0x220000 00	附加轴%d 目标位置超过最大限位	附加轴目标位置超过最大限位。	重新设置运行的目标位置或者对附加轴进行清零。
1811	0x220000 00	外部 Plc 报警	外部 Plc 报警。	外部 Plc 报警, 检查 plc 状态。
1812	0x220000 00	安全门报警	安全门报警。	1.检查安全门是否被打开并确认工作区域内是否有工作人员; 2.检查安全门信号是否正常。
1813	0x220000 00	附加轴%d:请检查运动模式是否设置正确	请检查运动模式是否设置正确。	检查附加轴 APP 中, 运动模式是否设置正确。
1814	0x220000 00	非法状态重启控制器	非法状态重启控制器。	检查机器人当前伺服状态。
1815	0x220000 00	mot 程序: 非法状态单步运行	mot 程序: 非法状态单步运行。	检查机器人当前伺服状态。
1816	0x220000 00	mot 程序: 单步运行错误	mot 程序: 单步运行错误。	检查当前位置点、工具坐标系、用户坐标系设置是否正确。



1817	0x220000 00	mot 程序：打 开程序文件错 误！	mot 程序：打开 程序文件错误！	检查文件路径是否正 确。
1818	0x220000 00	mot 程序：读 取程序文件错 误！	mot 程序：读取 程序文件错误！	检查文件格式是否正 确。
1819	0x220000 00	动力学：初始 化失败	动力学辨识文件 或负载辨识文件 错误。	检查动力学辨识文件及 负载辨识文件是否正 确。
1820	0x220000 00	安全板：外部 急停信号	外部急停按钮被 按下。	检查外部急停按钮是否 被松开。
1821	0x220000 00	安全板：安全 门报警	安全门报警。	1.检查安全门是否被打 开并确认工作区域内是 否有工作人员； 2.检查安全门信号是否 正常。
1822	0x220000 00	安全板：安全 光栅报警	安全光栅报警。	1.检查安全光栅是否被 触发并确认工作区域内 是否有工作人员； 2.检查安全光栅信号是 否正常。
1823	0x220000 00	安全板急停 信号错误	安全板急停信号 错误。	1.请检查急停按钮按下 后是否复位； 2.按下电柜前面板复位 按钮，观察安全板 EMG 与 EMGO 灯是否亮起； 3.检查安全板 X4 连接器 与电柜前面板急停按钮 接线是否断线 或接触不良； 4.检查安全板 X5 连接器 与示教器急停接线是否 断线或接触不良 5.更换安全板； 6.重启电柜。



1824	0x220000 00	安全板外部急停错误	安全板外部急停错误	<ol style="list-style-type: none"> <li>1.检查外部设备急停被按下后是否复位；</li> <li>2.按下电柜前面板复位按钮，观察安全板 EEM G 灯是否亮起；</li> <li>3.检查安全板 X3 连接器与电柜前面板外部急停端子之间的接线是否断开或者接触不良；</li> <li>4.更换安全板；</li> <li>5.重启电柜。</li> </ol>
1825	0x220000 00	安全板手压信号错误	安全板手压信号错误	<ol style="list-style-type: none"> <li>1.按下电柜前面板复位按钮后再次按压手压按钮，观察安全板 ENA 灯是否亮起；</li> <li>2.检查示教器与安全板 X9 连接器之间手压信号接线是否断线或者接触不良；</li> <li>3.更换安全板；</li> <li>4.重启电柜。</li> </ol>
1826	0x220000 00	安全板 STOP1 错误	安全板 STOP1 错误	<ol style="list-style-type: none"> <li>1.按下电柜前面板复位按钮，观察安全板 STO P1 灯是否亮起；</li> <li>2.检查安全板 X7 连接器与电柜前面板外部急停端子间接线是否断线或接触不良；</li> <li>3.更换安全板；</li> <li>4.重启电柜。</li> </ol>
1827	0x220000 00	安全板 STOP2 错误	安全板 STOP2 错误	<ol style="list-style-type: none"> <li>1.按下电柜前面板复位按钮，观察安全板 STO P2 灯是否亮起；</li> <li>2.检查安全板 X8 连接器与电柜前面板外部急停端子间接线是否断线或接触不良；</li> <li>3.更换安全板；</li> </ol>



				4.重启电柜。
1850	0x220000 00	弧焊：焊机异常,代码：%d。	焊机异常。	查看焊机报警，解决焊机异常后再开始焊接。 代码 0：总线通讯时，焊机与机器人未连接；模拟量通讯时，焊机异常。 代码 1~100：总线通讯时，焊机面板有错误显示。
1851	0x220000 00	弧焊：冷却系统异常。	焊机冷却系统异常。	解决冷却系统异常后，再开始焊接。
1852	0x220000 00	弧焊：保护气异常。	保护气异常。	解决保护气异常后，再开始焊接。
1853	0x220000 00	弧焊：焊枪发生碰撞。	焊枪发生碰撞。	1.依次打开“弧焊 APP-设备设置-碰撞检测开关”，关闭焊枪碰撞检测信号，并清除报警； 2.将焊枪复位； 3.重新开启焊枪碰撞检测信号。
1854	0x220000 00	弧焊：工具末端速度超过设定值。	当前机器人末端的运动速度，超过设定值。	1.检查机器人是否发生飞车； 2.设置合理的工具末端速度最大值。



## A.4 一般报警代码（3900~3999）

报警代码	FLAGS	报警信息	报警原因	解决办法
3900	0x220000 00	冲压机急停	冲压机急停。	1.冲压机急停输入； 2.检查信号是否正确。
3901	0x220000 00	冲压机上死点丢失	冲压机上死点丢失。	1.冲压机上死点丢失信号输入； 2.检查信号是否正确。
3902	0x220000 00	非单张料片	非单张料片。	1.检测到非单张料片信号； 2.检查信号是否正确。
3903	0x220000 00	冲压机脉宽小于最短时间	冲压机脉宽小于最短时间。	冲压机属性设置：脉宽时间小于最短时间，请重新设置。
3904	0x220000 00	冲压机脉宽大于最长时间	冲压机脉宽大于最短时间。	冲压机属性设置：脉宽时间大于最长时间，请重新设置。
3905	0x220000 00	冲压机非单次模式	冲压机非单次模式。	1.检测到冲压机非单次模式信号输入； 2.检查冲压机属性设置是否合理； 3.检查信号是否正确。
3906	0x220000 00	等待超时	等待超时。	1.输入信号等待超时； 2.检查自定义动作时间设置是否合理。
3907	0x220000 00	手动 JOG 模式下区域被占用	手动模式下区域被占用。	区域内机器人移出后，自动复位。
3908	0x220000 00	运行程序时轴%d 接近限位	运行程序时当前轴接近限位。	检查轴是否到达限位。
3909	0x220000 00	示教器未连接	示教器未连接。	检查示教器是否连接好。
3910	0x220000 00	轴%d 发生碰撞	轴发生碰撞。	检查轴是否发生碰撞。
3911	0x220000 00	码件掉落	码件掉落	将掉落的码件放回夹爪上。



3920	0x22000000	弧焊：焊丝粘丝。	粘丝检测开，机器人检测到焊丝粘丝。	手动处理粘丝。若需要自动解除粘丝，需要开启粘丝解除功能。
3921	0x22000000	弧焊：焊丝粘丝解除失败。	粘丝自动解除开，但粘丝后，机器人解除粘丝失败。	手动处理粘丝。设置合理的粘丝解除电流电压。
3922	0x22000000	弧焊：焊接过程断弧。	断弧检测开，焊接过程中，发生电弧中断。	断弧重启或重新焊接。
3923	0x22000000	弧焊：焊接起弧失败。	开始起弧时，起弧失败。	将焊件表面处理干净后再执行起弧命令。
3924	0x22000000	弧焊：焊接收弧失败。	开始收弧时，收弧失败。	设定合理的收弧参数。
3925	0x22000000	弧焊：刮擦起弧失败。	刮擦起弧功能开，机器人在起弧点起弧失败，机器人边运动边起弧仍然失败。	1.将焊件表面处理干净后再起弧； 2.设置合理的起弧参数。
3926	0x22000000	弧焊：寻位失败，未寻到工件。	1.寻位距离设置不合理； 2.焊件表面有油或漆等覆盖物。	1.设置合理的寻位距离； 2.清理焊件表面覆盖物。
3927	0x22000000	弧焊：启用电弧跟踪之前请先开启摆弧功能。	启用电弧跟踪之前摆弧功能未开启。	启用电弧跟踪之前请先开启摆弧功能。
3928	0x22000000	弧焊：参数设置非法，请检查，代码：%d	参数设置超出了规定范围。 代码 0：预留； 代码 1：起弧、收弧指令文件号非法； 代码 2：预留； 代码 3：预留； 代码 4：预留； 代码 5：摆弧指令文件号非法； 代码 6：电弧跟踪	请在规定范围内设置参数。



			指令文件号非法。	
3929	0x22000000	弧焊：指令使用错误，请重新执行程序。	1.起弧失败后未重新执行该指令； 2.非断弧情况下，机器人产生故障； 3.非断弧情况下，手动暂停了机器人。	将弧焊信号复位，重新执行程序。
3930	0x22000000	弧焊：当前机器人运动非笛卡尔运动	弧焊焊接过程中，机器人非笛卡尔运动。	弧焊过程中选择笛卡尔空间的运动方式，如直线运动，圆弧运动。
3931	0x22000000	pallet2 码垛模式中产品抓取信息有错误	高级码垛模式中产品抓取信息有错误。	1.检查模式文件的产品抓取信息是否正确。 2.如果初次运行出现此报警，请到码垛生产界面，初始化该工艺流。



## A.5 用户报警代码（4900~4999）

报警代码	FLAGS	报警信息	报警原因	解决办法
4900	0x22000000	轴组%s: 非法轴组模式 <%d>	非法轴组模式。	
4901	0x22000000	轴组%s: 零点位置丢失	零点位置丢失。	请检查机器人编码器零点与机械零点是否对应。
4902	0x22000000	XPL API 错误,请在程序日志中查找原因	XPL API 错误。	请查看程序日志,根据日志中的内容排查问题。
4903	0x22000000	XPL 程序错误,请在程序日志中查找原因	XPL 程序错误。	请查看程序日志,根据日志中的内容排查问题。
4904	0x22000000	XPL 程序错误,程序运行时远程加载程序	加载当前程序错误:其他程序正在运行。	请停止当前运行的程序,然后再进行远程加载程序。
4905	0x22000000	XPL 程序错误,远程加载程序不存在	远程加载的程序不存在。	请确保加载的程序存在。
4906	0x22000000	碰撞检测,读取参数文件失败	读取碰撞检测参数文件失败。	检查/application/collisiondetect/collisionpardata.xml 是否存在或者文件内容是否正常。
4907	0x22000000	碰撞检测,读取配置文件失败	读取碰撞检测配置文件失败。	检查 application/collisiondetect/目录下是否有此机器人配置文件,或文件内容是否正确。
4908	0x22000000	附加轴%d 正在运行,运动指令无效	附加轴运行期间接受到运动指令。	请在附加轴运行结束后,再发运动指令。
4909	0x22000000	读取附加轴参数文件失	读取附加轴参数文件失败提示报	检查卡中目录/Application/auxaxis/auxpardata.x



		败	警。	ml 文件是否存在，不存在则重新配置参数生成文件
4910	0x22000000	更新附加轴参数文件失败	更新附加轴参数文件失败提示报警。	检查卡中目录/Application/auxaxis/auxpardata.xml 文件是否存在以及是否保存成功，不存在则重新配置参数生成文件
4911	0x22000000	在上电情况下修改配置参数	在上电情况下修改配置参数提示报警。	请在手动掉伺服的情况下，点击保存按钮。
4912	0x22000000	目标位置超过最大限位	目标位置超过最大限位。	检查单步运行目标位置是否超过限位，以及附加轴是否进行清零。
4913	0x22000000	读取文件时，打开参数文件失败	读取文件时，打开参数文件失败。	检查卡中目录/Application/auxaxis/auxpardata.xml 文件是否存在，不存在则重新配置参数生成文件。
4914	0x22000000	读取附加轴文件错误	读取附加轴文件错误。	检查卡中目录/Application/auxaxis/auxpardata.xml 文件是否存在或者文件内容是否正确，不存在则重新配置参数生成文件。
4915	0x22000000	写文件时，打开参数文件失败	写文件时，打开参数文件失败。	检查卡中目录/Application/auxaxis/auxpardata.xml 文件是否存在，若不存在则重新生成参数文件。
4916	0x22000000	程序运行模式下区域被占用	程序运行模式下区域被占用。	将区域内机器人移出，自动复位。
4917	0x22000000	等待取料超时	等待取料超时提示报警。	1.请检查取料点是否有料； 2.请检查信号是否正确。
4918	0x22000000	等待放料超	等待放料超时。	1.请检查取料点是否有



	00	时		料; 2.请检查信号是否正确。
4919	0x220000 00	工具 1 无反 馈	工具 1 无反 馈。	1.请检查工具 1 是否有 料; 2.请检查信号是否正确。
4920	0x220000 00	工具 2 无反 馈	工具 2 无反 馈。	1.请检查工具 2 是否有 料; 2.请检查信号是否正确。
4921	0x220000 00	机器人运行 时切换模式 是非法的	机器人在运行过 程中被切换了运 行模式。	直接清除报警, 不要在 机器人运行过程中切换 运行模式。
4922	0x220000 00	读取外部 IO 文件失败	读取外部 IO 文件 失败。	检查卡中目录/Applicati on/io/remotepardata.xml 1 文件是否存在, 若不存 在则重新配置生成文 件。
4923	0x220000 00	外部 IO 模块 组态错误	外部 IO 模块组态 错误。	检查模块数量和顺序是 否和 IO 配置中的组态一 致。
4924	0x220000 00	外部 IO 设备 故障	外部 IO 设备故 障。	检查外部 IO 设备是否正 常。
4925	0x220000 00	外部 IO 模块 故障	外部 IO 模块出现 故障。	检查外部 IO 模块是否正 常。
4926	0x220000 00	外部 IO EC T 计数器错 误	外部 IO ECT 计 数器错误。	检查外部 IO 模块通讯是 否正常。
4927	0x220000 00	热插示教器 后有错误的 钥匙模式	示教器热插拔前 后模式不同。	确保热插拔示教器前 后, 示教器模式相同。
4928	0x220000 00	pallet2 码垛 配置信息错 误或者文件 不存在	高级码垛配置信 息错误或者文件 不存在。	检查配置文件是否存在 或者信息是否正确。
4929	0x220000 00	pallet2 码垛 模式信息错 误或者文件 不存在	高级码垛模式信 息错误或者文件 不存在。	检查模式文件是否存在 或者信息是否正确。



4930	0x220000 00	码件掉落	码件掉落	将掉落的码件放回夹爪上。
4931	0x220000 00	pallet2 码垛模式中产品码放信息有错误	高级码垛模式中产品码放信息有错误。	检查模式文件的产品放置信息是否正确。
4932	0x220000 00	读取模拟量 I O 配置文件失败	读取模拟量 IO 配置文件失败。	检查配置文件是否存在。
4933	0x220000 00	轴%d 多圈值丢失,请在零点位置重置多圈值	当前轴多圈值丢失,请在零点位置重置多圈值。	请勿重启机器人 1.将机器人回零; 2.重置多圈值; 3.清除报警; 4.重置零点。
4934	0x220000 00	附加轴%d: 请确认远程示教信号是否打开	没有附加轴远程确认信号。	检查 IO 自由配置界面中附加轴远程示教信号是否存在。
4935	0x220000 00	附加轴%d: 请检查运动模式是否正确	附加轴运动模式设置错误。	检查附加轴 APP 中, 运动模式是否正确。
4936	0x220000 00	简单码垛: 更新码垛位置错误	1.码垛序号输入错误; 2.码垛模式输入错误; 3.工件编号选择错误; 4.输入垛盘未标定。	1.请确认当前输入码垛序号正确 (0-8); 2.请确认码垛模式输入值符合要求 (0/1); 3.请确定正确的工件编号; 4.请选择已标定的垛盘, 或标定当前选择垛盘。
4937	0x220000 00	动力学: 读取动力学辨识文件路径错误	动力学辨识文件路径错误。	查看动力学辨识文件路径是否正确。
4938	0x220000 00	动力学: 读取动力学辨识参数错误	动力学辨识参数错误。	查看动力学辨识文件参数是否正确。



4939	0x22000000	动力学: 读取负载辨识辨识文件路径错误	负载辨识辨识文件路径错误。	查看负载辨识文件路径是否正确。
4940	0x22000000	动力学: 读取负载辨识辨识参数错误	负载辨识辨识参数错误。	查看负载辨识文件参数是否正确。
4941	0x22000000	动力学: 辨识过程中发生异常	辨识过程中发生异常。	1.辨识过程中是否出现异常报警; 2.检查 RPL 程序是否终止或者正在运行。
4942	0x22000000	动力学: 请在自动上伺服模式下开启运行轨迹	机器人模式或者伺服状态错误。	检查机器人模式及伺服状态。
4943	0x22000000	动力学: 生成负载辨识轨迹错误	负载辨识参数错误。	检查负载辨识参数是否正确。
4944	0x22000000	跟踪视觉: 相机自动连接超时	跟踪视觉: 相机自动连接超时。	1.检查相机是否正确开启; 2.检查相机硬件连接是否正常; 3.检查超时时间设置是否合理; 4.尝试手动连接。
4945	0x22000000	附加轴: 在 PLC 控制模式下, 设置附加轴运行轴数错误	设置附加轴运行轴数错误。	在 PLC 控制运动轴数时, 请检查设置运行的附加轴是否正确, 其中七轴为 1, 依次类推。
4946	0x22000000	附加轴: 在 PLC 控制模式下, 在改变附加轴运行轴数之前请将附加轴运行信号按钮关闭	在改变附加轴运行轴数之前请将附加轴运行信号按钮关闭。	在 PLC 控制运动轴数时, 请检查附加轴运行按钮是否被按下。



4947	0x22000000	附加轴: 在 PLC 控制模式下, 伺服关闭后, 请将附加轴运行信号按钮关闭	伺服关闭后, 请将附加轴运行信号按钮关闭。	在 PLC 控制运动轴数时, 伺服关闭后, 请检查附加轴运行按钮是否被按下。
4948	0x22000000	远程 IO: 不能同时打开汇川 IO 和 EFORT IO	远程 IO: 不能同时打开汇川 IO 和 EFORT IO。	1.关闭汇川 IO 或者 EFORT IO。 2.点击保存按键生成新的配置文件。
4949	0x22000000	EFORT IO 板: ARM 初始化异常, 设备需要重新启动	EFORT IO 板: ARM 初始化异常。	ARM 初始化异常, 设备需要重新启动。
4950	0x22000000	弧焊: 尝试重新起弧中。	设置起弧次数大于 2 后, 当首次起弧失败, 机器人将再次起弧。	等待机器人自动处理。
4951	0x22000000	弧焊: 断弧后, 尝试回退并重新起弧。	断弧重启开, 断弧后, 机器人回退一定距离再重新起弧。	等待机器人自动处理。
4952	0x22000000	弧焊: 加载配置文件失败, 代码: %d。	控制器中配置文件缺失, 导致配置文件读取失败。	在机器人示教器界面重新配置参数文件并保存。 代码 0: 加载焊机设置文件失败; 代码 1: 加载焊接参数文件失败; 代码 2: 加载设备设置文件失败; 代码 3: 加载电流特性文件失败; 代码 4: 加载电压特性文件失败; 代码 5: 加载摆弧文件失败;



				代码 6: 加载电弧跟踪文件失败。
4953	0x22000000	弧焊: 将机器人移动到故障点附近后再开始故障启动。	故障启动前, 机器人末端与故障点位置距离超过设定距离。	1. 将机器人移动到故障点附近; 2. 设置合理的距离范围。
4954	0x22000000	弧焊: 尝试通过刮擦方式起弧。	刮擦启动开, 正在进行刮擦启动。	等待机器人自动处理。
4955	0x22000000	弧焊: 请将机器人的模式开关切换至自动模式并使能伺服。	PC 端控制机器人运动, 模式非自动或伺服未使能。	将机器人运动模式切换到自动并伺服使能。
4956	0x22000000	弧焊: 协议文件与硬件不匹配, 请在装置设置中重新配置	开机时检测到协议文件与硬件不匹配。	在装置设置中重新配置协议并保存重启机器人。
4957	0x22000000	弧焊: 手动起弧运行失败	手动起弧运行失败	手动起弧运行失败, 尝试检查以下原因。 原因 1: 机器人未处于暂停状态; 原因 2: 机器人伺服未使能; 原因 3: 未执行过起弧指令, 当前运动非焊缝路径。
4961	0x22000000	EFORT IO 板: 系统时钟初始化异常, 设备需要重	EFORT IO 板: 系统时钟初始化异常。	设备需要重新启动。



		新启动		
4962	0x22000000	EFORT IO板:系统内部总线初始化异常,设备需要重新启动	EFORT IO板:系统内部总线初始化异常。	设备需要重新启动。
4963	0x22000000	EFORT IO板: DI 运行异常,设备需要重新初始化或者重新启动;	EFORT IO板: DI 运行异常。	设备需要重新初始化或者重新启动。
4964	0x22000000	EFORT IO板: DO 运行异常,设备需要重新初始化或者重新启动;	EFORT IO板: DO 运行异常。	设备需要重新初始化或者重新启动。
4965	0x22000000	EFORT IO板:模拟输出运行异常,设备需要重新初始化或者重新启动;	EFORT IO板:模拟输出运行异常。	设备需要重新初始化或者重新启动。
4966	0x22000000	EFORT IO板:模拟输入运行异常,设备需要重新初始化或者重新启动;	EFORT IO板:模拟输入运行异常。	设备需要重新初始化或者重新启动。
4967	0x22000000	拖动示教:加载拖动配置文件错误	拖动示教:加载拖动配置文件错误。	检查拖动配置文件是否存在或内容是否有误。
4968	0x22000000	拖动示教:加载驱动参数文件错误	拖动示教:加载驱动参数文件错误。	检查驱动参数文件是否存在或内容是否有误。
4969	0x22000000	拖动示教:加	拖动示教:加载末	检查末端设置文件是否



	00	载末端工具 配置文件错 误	端工具配置文件 错误。	存在或内容是否有误。
4970	0x220000 00	拖动示教:无 法开启拖动 功能	拖动示教:无法开 启拖动功能。	1.确保机器人已上伺服 并处于自动模式的执行 模式; 2.检查驱动器的力矩控 制状态是否正确; 3.请先停止机器人再开 启拖动示教; 4.检查动力学模型是否 有误,负载设置是否正 确; 5.检查各关节当前位置 是否在拖动示教的允许 范围内或2,3轴位置距 离零点是否过近(小于2 0度); 6.检查拖动示教配置文 件或驱动器力矩控制文 件是否有误。
4971	0x220000 00	拖动示教:拖 动过程太危 险,关节超出 限位或限速	拖动示教:拖动过 程太危险,关节超 出限位或限速。	检查拖动过程中的力是 否过大或检查拖动配置 文件参数设置是否有 误。
4972	0x220000 00	协作安全:超 过安全 TCP 速度限制	协作安全:超过安 全 TCP 速度限 制。	降低机器人运行速度或 调高 TCP 速度限制值。
4973	0x220000 00	协作安全:超 过安全 TCP 力限制	协作安全:超过安 全 TCP 力限制。	调高 TCP 力限制值,或 者减小对机器人的力。
4974	0x220000 00	协作安全:超 过安全 TCP 动量限制	协作安全:超过安 全 TCP 动量限 制。	降低机器人运行速度, 或者减小负载大小,又 或者调高 TCP 力限制 值。
4975	0x220000 00	协作安全:超 过安全关节 力矩限制	协作安全:超过安 全关节力矩限制。	减小对机器人关节的 力,或者调高力矩限制 值。



4976	0x22000000	协作安全:超过安全功率限制	协作安全:超过安全功率限制。	降低机器人速度,或者减小负载,又或者调高力矩限制值。
4977	0x22000000	协作安全:一个或多个安全平面被入侵	协作安全:一个或多个安全平面被入侵。	示教器切换到手动模式并将机器人 TCP 远离安全平面。
4978	0x22000000	未识别 IO 板	控制器未识别扩展 IO 模块	检查扩展 IO 模块是否为机器人适配类型,若不是请更换。
4979	0x22000000	奇异点处机器人停止运行	1. 机器人程序经过奇异点。 2. 点动机器人时,接近奇异点。	1. 请重新示教程序,避开机器人奇异点。 2. 避开奇异点,或者切换到关节空间下运动机器人。
4980	0x22000000	折弯:工件厚度不合格。	折弯:工件厚度不合格。	请检查工件数量,并重新放置。
4981	0x22000000	折弯:折弯刀速度不合法。	折弯:折弯刀速度不合法。	请重新配置折弯刀速度。
4982	0x22000000	动力学:更新摩擦参数失败	1、摩擦更新时程序并未一直在运行 2、摩擦更新时不能安装负载 3、摩擦更新时倍率不得低于 70% 4、关节实际最大速度超过允许范围 5、速度未达到最大速度的 65%,无法求解 6、更新动力学配置文件失败	1、摩擦参数更新时保持机器人程序运行 2、拆下负载,并取消负载设定; 3、增大倍率至 70%以上; 4、降低倍率至 95%; 5、增大倍率且增大各轴的运动范围 6、检查动力学配置文件是否存在及格式是否正确
4983	0x22000000	动力学:更新摩擦参数成功,请重启机器人	重启机器人控制器	重启机器人控制器



4984	0x22000000	工具平齐: 方向夹角计算超限, 请调整工具姿态	工具坐标系与参考坐标之间的夹角过大	调整工具姿态, 使工具坐标系与参考坐标之间的夹角尽可能减少
4985	0x22000000	工具平齐: J C 转换失败	逆解失败	调整工具姿态, 使工具坐标系与参考坐标之间的夹角尽可能减少
4986	0x22000000	工具平齐: 关节角计算超限, 请调整工具姿态	当前位置与目标位置之间误差过大	调整工具姿态, 使工具坐标系与参考坐标之间的夹角尽可能减少
4987	0x22000000	工具平齐: 目标位置不可达	目标位置不可达	调整工具姿态, 使工具坐标系与参考坐标之间的夹角尽可能减少
4988	0x22000000	上伺服前没有进行伺服确认操作	/	1. 按下电柜面板的伺服确认按钮; 2. 清除示教器关于模式切换的弹窗报警
4989	0x22000000	驱动器: 关节滑移超限	关节滑移超过限值	1. 检查原始绝对值编码器数据是否存在; 2. 记录当前绝对值编码器位置时, 是否走到原始位置; 3. 关节滑移超过 5° ;



## A.6 事件日志（4900~4999）

事件代码	FLAGS	事件信息
10000	0x23000000	伺服开
10001	0x23000000	伺服关
10002	0x23000000	XPL Task %d: 开始按钮按下
10003	0x23000000	XPL Task %d: 停止按钮按下
10004	0x23000000	XPL Task %d: 停止
10005	0x23000000	轴%d 达到了负限位
10006	0x23000000	轴%d 达到了正限位
10007	0x23000000	示教器已拔出
10008	0x23000000	示教器已插入
10009	0x23000000	加载 RPL 文件成功
10010	0x23000000	压机逃离
10011	0x23000000	压机异常紧急停机
10012	0x23000000	EC2x 驱动参数:%d 已保存
10013	0x23000000	EC2x 驱动参数:%d 已加载



# 附录 B 文档修订记录

序号	版本号	修订日期	修订概述
1	V1.0	2020-11-18	新建文档
2	V1.1	2021-10-28	<p>第 4 章 程序选项卡</p> <p>1. 更新 程序模块界面图片</p> <p>第 5 章 安装设置选项卡</p> <p>1. 更新 5.1.1 安装角度内容；</p> <p>2. 更新 5.1.4 负载辨识内容；</p> <p>3. 更新 5.1.6 默认位姿 为 初始位姿</p> <p>4. 新增 5.1.7 模块更新功能；</p> <p>5. 更新 5.1.8 IO 配置内容；</p> <p>6. 更新 5.1.10 附加轴内容；</p> <p>7. 新增 5.1.12 传送带功能；</p> <p>8. 更新 5.2.1 区域监控内容；</p> <p>9. 更新 5.2.2 位置监控内容；</p> <p>10. 更新 5.2.4 碰撞检测内容；</p> <p>11. 更新 5.2.5 安全限制内容；</p> <p>12. 更新 5.4.1 TCPIP 内容；</p> <p>13. 新增 5.4.3 EtherCAT 设置功能</p> <p>14. 新增 5.4.4 EthernetIP 设置功能；</p> <p>第 6 章 移动选项卡</p> <p>1. 更新 默认位姿按钮为初始位姿按钮</p> <p>第 7 章 监控选项卡</p> <p>1. 更新 7.2 IO 监控内容；</p> <p>2. 新增 7.3 现场总线数据监控功能</p>

**服务热线：4000528877**

本产品的额定功率、规格、外部尺寸等如需改良而进行变更，恕不另行通告。技术数据和插图仅作为供货参考，保留更改权利。



埃夫特智能装备股份有限公司

**Efort Intelligent Equipment Co.,Ltd**

地址：安徽省芜湖市鸠江经济开发区万春东路 96 号

Address: No.96 Wanchun East Road, Jiujiang Economic Development Zone, Wuhu, Anhui.

网址：<http://www.efort.com.cn>

