

目录

目录

1 KND 机器人控制器	8
1.1 KND 机器人控制器介绍	8
1.1.1 KND 机器人控制器特点	8
1.1.2 KND 机器人控制器功能简述	8
1.1.3 KRL 概述	9
1.2 示教编程器	10
1.2.1 KND 示教编程器外观介绍	10
1.2.2 示教器按键功能介绍	12
1.2.3 状态显示条	15
1.3 模式	18
1.3.1 示教模式【TEACH】	19
1.3.2 再现模式【PLAY】	19
1.3.3 示教模式优先	19
2 界面简介	20
2.1 监控管理	21
2.1.1 位置显示	22
2.1.2 版本信息管理	23
2.1.3 伺服扭矩监控	24
2.1.4 示波器采样	25
2.2 项目管理	29
2.2.1 项目管理	29

2.2.1 属性管理	31
2.2.3 程序页面管理	33
2.3 变量管理	35
2.3.1 项目变量	35
2.3.2 局部变量	38
2.3.3 系统变量	40
2.4 I/O 管理	41
2.4.1 输入输出	41
2.5 常规设置	44
2.6 报警报告处理	45
3 Robot 坐标系的观念与轴操作	48
3.1 坐标系的观念	48
3.2 控制组与坐标系	48
3.3 基本轴操作步骤	50
3.3.1 安全确认	50
3.3.2 示教模式选择	50
3.3.3 控制组的选择	50
3.3.4 坐标系的选择	50
3.3.5 速度选择	50
3.3.6 接通伺服	51
3.3.7 轴操作	51
3.4 坐标系与轴操作	52
3.4.1 关节坐标系	52
3.4.2 世界坐标系	53
3.4.3 工具坐标系	54

3.4.4 用户坐标系	55
3.4.5 机器人坐标系	56
3.4.6 机器人控制点保持不变的操作	57
4 示教编程操作	60
4.1 示教前准备	60
4.1.1 确保机械手活动范围安全	60
4.1.2 急停按钮使用确认	60
4.1.3 关于示教模式安全保障	60
4.1.4 如何创建新的程序	61
4.2 示教	62
4.2.1 示教编程界面	62
4.2.2 运动插补介绍	65
4.2.3 如何示教程序点	69
4.2.4 最初程序点与最后程序点的重合方法	73
4.3 确认程序点	74
4.3.1 前进/后退的操作	74
4.3.2 前进/后退中需要注意的事项:	74
4.3.3 速度选择	76
4.3.4 试运行	77
4.4 修改程序点	78
4.4.1 命令的修改	78
4.4.2 命令的插入	79
4.4.3 命令的删除	80
4.4.4 命令的复制和粘贴	82
4.4.5 命令的移动与粘贴	84

4.4.6 命令的注释	85
4.5 试运行	86
4.6 其它程序的编辑	87
4.6.1 程序跳转	87
4.6.2 程序调用	88
4.7 自定义变量	89
4.7.1 自定义变量简介	89
4.7.2 自定义变量使用方法	89
5 再现运行程序	95
5.1 再现前准备	95
5.1.1 再现模式下的操作限制	95
5.2 再现	96
5.2.1 再现过程	96
5.2.2 再现低速启动	97
5.3 停止与再启动	97
5.3.1 暂停操作	98
5.3.2 急停操作	98
5.3.3 报警引起的停止	99
5.3.4 其它停止	99
5.4 再现速度的修改	100
5.5 离线模式	100
6 应用举例	101
6.1 示教前准备	101
6.2 示教基本步骤	102
6.2.1 示教基本步骤	102

6.2.2 示教程序开始	103
6.3 轨迹的确认	109
6.4 修改程序	110
6.4.1 修改程序点的位置数据	110
6.4.2 插入程序点	111
6.4.3 删除程序点	111
6.5 程序再现	113
6.5.1 再现前准备	113
6.5.2 再现	113
7 用户坐标设定	114
7.1 用户坐标系的调用	114
7.1.1 手动调用	114
7.1.2 REFSYS 指令设定	115
7.2 用户坐标设置	115
7.2.1 三点（含原点）示教法	116
7.2.2 三点（不含原点）示教法	119
7.2.3 一点（保持姿态）示教法	123
8 工具尺寸设定	125
8.1 工具坐标的调用	125
8.1.1 手动调用	125
8.1.2 TOOL 指令设定	126
8.2 工具坐标设置	126
8.2.1 工具坐标值设置	126
8.2.2 工具姿势数据设置	128
8.3 工具自动校准	129

8.3.1 四点（保持姿态）示教法.....	130
8.3.2 一点（保持姿态）示教法.....	134
8.4 工具对齐	137
9 用户数据备份与升级	138
9.1 文件简介	138
9.1 示教器升级操作	139
9.2 控制器导入与导出	140
10 区域监控	148
10.1 区域	148
10.1.1 工作区域被激活状态	149
10.1.2 禁止区域被激活状态	150
10.1.3 普通区域被激活状态	151
11 特殊功能包介绍	152
11.1 传送带	152
11.2 视觉	152
11.3 冲压工艺	152
11.5 焊接工艺	152
11.6 料盘	152
11.4 力传感器	153
12 KRL 指令一览表	155
12.1 运动指令	155
12.2 设置指令	156
12.3 I/O 指令	157
12.4 控制指令	158
12.5 操作指令	160

12.6 系统指令	160
12.7 复合循环指令	161
IF	161
WHILE	162
LOOP	162

1 KND 机器人控制器

1.1 KND 机器人控制器介绍

1.1.1 KND 机器人控制器特点

KND 机器人控制器主要特点如下：

- ☐ KND 控制器+KND 伺服或其它伺服+KND 或其它电机。
- ☐ KND 伺服总线（KSSB）/ MechatrolinkII / EtherCAT。
- ☐ 双通道，每通道 6 轴机器人+2 辅助轴。
- ☐ 通道间协作。
- ☐ KRL 语言（KND Robot Language）。
- ☐ 示教器 3D 实时显示，机器人离线编程/仿真。
- ☐ 6 轴串联关节、桁架、SCARA、四轴码垛、并联以及为客户定制异形机构。

1.1.2 KND 机器人控制器功能简述

- ☐ 离线编程/仿真功能：
不需连接机器人，参照 3D 图形即可进行示教/仿真。
- ☐ 试运行/单段运行/运动回退：
示教完毕后，在自动运行之前可 JOG/单段/运动回退，用于校验程序。
- ☐ CAM 软件编程：
当用于打磨，实体加工等应用时，可从 CAM 软件生成运动轨迹，经转换后，输入到控制器。
- ☐ 支持桁架机器人：
除 XYZ 空间外，还可支持 ABC 轴，及 2 个附加轴。
- ☐ 运行时改变参考坐标系：
可在运行时改变参考坐标系，实现视觉跟踪。
- ☐ 通道间协作：
控制系统可配置为通道 1 连接机器人，通道 2 仿真运行。通道间通过 TS/RV 地址通信。

1.1.3 KRL 概述

KRL (KND Robot Language) 是 KND 机器人编程语言, 用于编程 job 文件。

- ☐ 包含关节/直线/圆弧插补。
- ☐ 非运动指令可与运动指令并行执行, 有时间/距离等多种同步方式。
- ☐ 子程序调用 CALL/返回 RET/流程跳转 JUMP。
- ☐ 支持项目变量命名/自定义局部变量。
- ☐ 支持数字变量自动类型 (32 位整数/64 位整数/双精度浮点)。
- ☐ 支持字符串变量/位置变量。
- ☐ 支持复杂表达式/数学函数。

详细用法请查阅《KND 机器人语言指令手册》。

1.2 示教编程器

1.2.1 KND 示教编程器外观介绍

KND 示教编程器上设有机器人示教和编程所需的操作键和按钮，其整体结构如图所示：



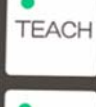
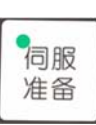
正面效果展示



背面与侧面效果展示

1.2.2 示教器按键功能介绍

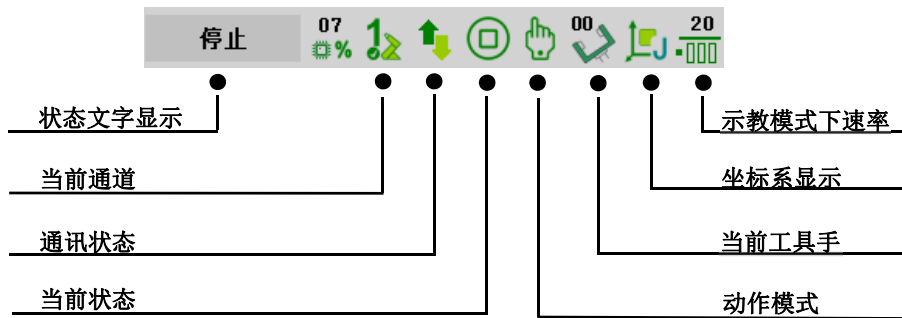
KND 示教编程器各按键功能如表所示：

名称	图标	功能
急停		按该键，伺服电源切断。 ● 切断伺服电源后，示教编程器的“伺服接通”灯灭。 ● 显示屏状态栏显示急停状态。
暂停		按该键，运动中的机器人暂停运动。 ● 该按键响应所有模式。 ● 按下该键，机器人保持暂停状态，直到得到下一个开始指示为止。
启动		按下该键，机器人开始再现运动。 ● 只在再现模式下生效。 ● 由于报警或暂停、模式切换等会停止再现运动。
模式	 	该按键若到【TEACH】，则为示教模式。用示教器可以进行轴操作或编程作业。
		该按键若到【PLAY】，则为再现模式。可再现示教过的程序。
		该按键若旋转到【EXT】，则为扩展模式，暂未使用。
安全开关		按该按键，伺服电源接通。 ● 模式键在【TEACH】时，轻轻按安全开关，可接通伺服电源。能够听见控制柜有接触器吸合的声音，在该状态下，若用力握安全开关，伺服电源断开。
伺服准备		按该键，伺服电源接通有效。 ● 当伺服电源由于急停被切断后，请使用该键使伺服电源接通有效。 ● 按该键 - 再现模式、在安全门关闭的情况下，伺服电源被接通。 - 伺服电源接通期间，伺服接通指示灯亮。

速度 调节 高 低		共用示教模式下的手动速度、试运行速度、再现模式下程序再现速度调节按键。 ● 手动速度修改：设定机器人轴操作速度。手动速度有 3 个等级（低、中、高）及微动可供选择。选定速度在显示屏状态显示区域显示。每按一次【高】键，按照微动—低—中—高顺序变化。每按一次【低】键，按照高—中—低—微动顺序变化。 ● 试运行倍率修改：先用【SHIFT】+【高】或【低】组合键进入试运行倍率修改状态，再使用【高】、【低】键修改试运行速度倍率，然后使用【SHIFT】+【高】或【低】组合键即可退出试运行倍率修改状态，回到手动速度调节状态。调节宽度在 10%~150%。 ● 再现程序速度修改：调节再现速度，不会改变手动、试运行速度状态，其调节变化在显示屏状态显示区域显示。调节宽度在 10%~150%。
光标、 选择键		光标、选择键： ● 上、下、左、右移动光标位置，如果是可修改框，在下拉菜单区域，进行菜单项目的选择。 ● 【选择】键，起到“回车”功能的作用。
坐标系 示教		轴操作时，用该键选择动作时的坐标系。可在关节、工具、参考（世界、用户和机器人）三种坐标系中循环切换选择。该键每按一次，坐标系按照以下顺序变化：关节—世界—参考。
SHIFT		与该键同时按，可使用其他功能。 可与 SHIFT 键同时按的键有： 【高】/【低】、数值键，同时使用的功能请参阅各键说明。
复位		复位键： ● 在特殊或紧急情况下，按复位可停止机械手动动作，光标同时返回到首行。 ● 可清除信息页面的报警内容。
试运行		可对示教过的程序点作为连续轨迹确认。 ● 需要在示教模式下。 ● 机器人按照示教速度运行。但是，示教速度若超过示教的最高速度，以示教最高速度运行。 ● 在连续运行中若松开【JOG】键，机器人则停止运行。

前进		只在按住该键期间，机器人按示教程序点的轨迹运行。 ● 需要在示教模式下。 ● 执行移动命令与非移动指令。 ● FWD 运行类型可在【设置】/【其他】页面进行设置。 ● 开始移动到下一步时，需要重新按该【FWD】键。 ● 机器人按照选定的试运行速度倍率运动，执行操作前，请确认速度倍率是否合理。
后退		只在按住该键期间，机器人沿示教程序点轨迹逆向运动。 ● 需要在示教模式下。 ● 只执行移动命令，遇到非移动指令即停止动作。 ● 开始移动到上一步时，需要重新按该【BWD】键。 ● 机器人按照选定的试运行速度倍率运动，执行操作前，请确认速度倍率是否合理。
翻页键		大容量内容向前、向后翻页浏览页面操作。
轴操作	 Grid of axis operation buttons: X1-, X1+, Y2-, Y2+, Z3-, Z3+, A4-, A4+, B5-, B5+, C6-, C6+, 7-, 7+, 8-, 8+.	操作机器人各个轴键。 ● 手动机器人只在该键按下时运动。 ● 轴操作键可同时进行 2 个轴以上运动，但是像同一个轴按下+X 与-X 键，轴不会动。 ● 机器人按照选定的坐标系与选定的速度运动。轴操作前，请确认坐标系与手动速度是否正确。

1.2.3 状态显示条



1: 可操作轴组，当系统带多台机器人时，显示当前可进行轴操作的控制轴组。

注意： 图标中含有两个状态信息：当前通道与运行模式。

运行模式**在线**时：

通道 1 通道 2

运行模式**离线**时：

通道 1 通道 2

2: Robot 状态，分为停止、暂停、急停、报警、运行。



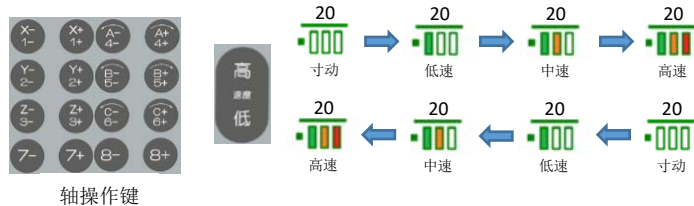
3: 动作切换：示教模式、再现模式。



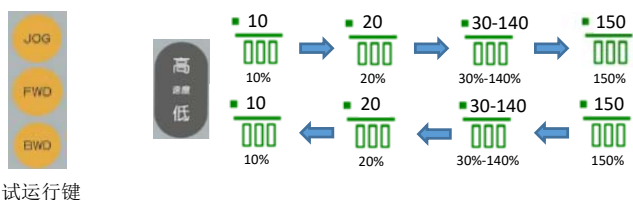
4: 示教模式下速率

注意： 状态图标中含有两个速率：**轴移动速率**与**试运行速率**；在图标左侧还有一个绿色实心点，表示当前可被调整速率。组合键 “SHIFT” + “高” 或 “低”，绿色实心标识点上下切换显示。

轴移动速率： 可按“高”或“低”键调节轴操作的速度,一个“微调寸动”+“低”、“中”、“高”三个挡位，共四种有效示教操作轴速度。



试运行速率： 可按“高”或“低”键调节始运行速率有效范围 10%~150%。



5: 再现速度倍率调节，切换到再现模式时，手动速度状态图标变换成再现速度倍率图标状态。

示教模式时：



再现模式时：



6: 动作坐标系: 显示可选择的坐标系。

注意: 状态图标中含有两个状态信息: **当前参考坐标系**与**轴移动坐标系**。

图标带笛卡尔坐标系指示为**参考坐标系**: 机器人坐标系、世界坐标系、用户坐标系
图标右下角指示轴**移动坐标系**: 关节坐标系、参考坐标系、工具坐标系

参考坐标系:

移动坐标系:

	关节坐标系	参考坐标系	工具坐标系
机器人坐标系			
世界坐标系			
用户坐标系			

7: 连接状态, 指示教器与控制系统通讯状态。



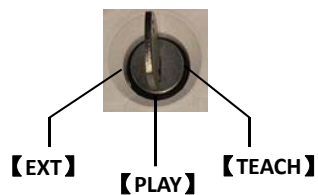
开机时自动连接, 状态条 通讯正常, 通讯连接失败, 点击通讯状态图标 , 弹出通讯窗口。



可手动输入 IP 地址，点击【连接】即连接到控制器，点击【断开】即断开连接。

1.3 模式

模式有示教模式【TEACH】、再现模式【PLAY】、扩展模式【EXT】三种。其中【EXT】模式暂未使用。



注：黑色外形示教器采用按键切换模式

1.3.1 示教模式【TEACH】

若需程序编辑或者对已加载的程序进行修改，要在示教模式下进行。另外，进行各种特性文件和各种参数设定也要在该模式下进行。

1.3.2 再现模式【PLAY】

机器人复现动作运行程序时使用的模式。

1.3.3 示教模式优先

示教模式下不能进行以下操作：

- ①【START】按钮不能进行再现操作。
- ②不能用外部输入信号进行操作。

2 界面简介

选择屏幕左下角图标  弹出主菜单项
其菜单结构框架如下：

第一级		子集	子集	功能
【监视】		【位置】	【关节】、【世界】、【用户】、【机器人】	显示坐标、3D 模型、版本信息，伺服的负载，使用示波器画面进行监测。
		【控制器】	【概况】、【程序】	
		【伺服】	【概览】、 【伺服 1】～【伺服 8】	
		【示波器】		
		【力传感器】		
【项目】		【信息】	【当前报警】、【历史报警】、【CNC 日志】、【HMI 日志】	程序编辑算数、计算结果、位置保存。 设定用户坐标系、工具坐标系，设定区域、传送带追踪、编辑和调试辅助动作、设定参考点、料盘。
		【项目管理】		
		【调试】	【辅助动作】、【返回参考点】	
		【项目变量】	【数字】、【字符串】、【位置】	
		【用户坐标系】		
		【工具】		
		【区域】		
		【传送带】		
		【参考点】		
		【辅助动作】		
【参数】		【料盘】		显示控制器、伺服参数。并可在此界面修改操作。（调试级别使用）
		【控制器】		
【I/O】		【伺服】	【伺服 1】～【伺服 8】	I/O 接口状态、通道间的状态
			【G 输入】 【F 输出】	
		【输入输出】	【G 输入】	I/O 接口状态、通道间的状态
			【F 输出】	

			【RV 输入】	位的传送、PLC 的查看、控制器和示教器数据导入导出。
			【TS 输出】	
		【PLC】	【概况】、【诊断】、【题图】 【PLC 轴】	
		【系统变量】	【数字】、【字符串】、【位置】	
		【数据备份】		
【设置】		【工具对齐】		机器人模型连杆参数，关节限位、运行模式及其他设置。
		【关节限位】		
		【机构配置】		
		【其他】		

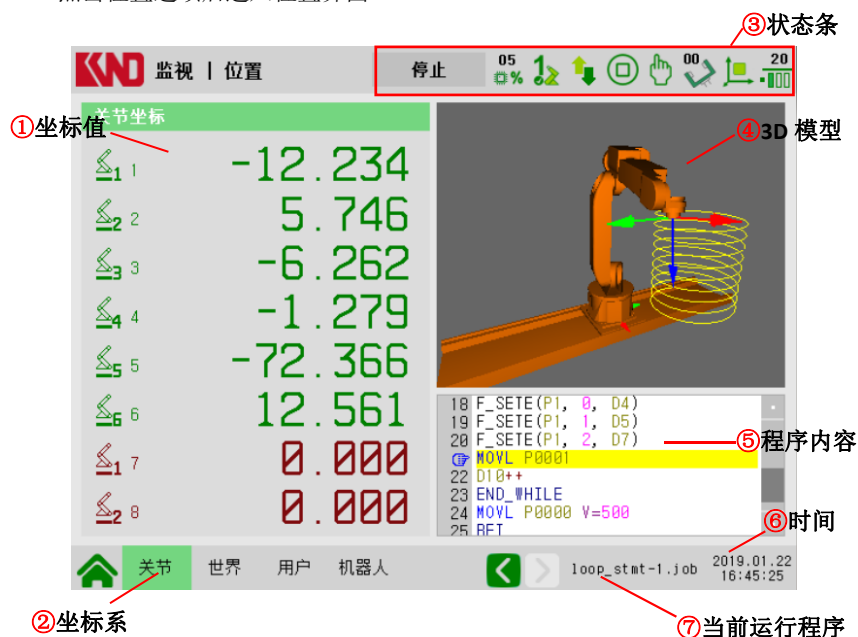
2.1 监控管理

选择左下角图标 ，弹出如下图主菜单项。



2.1.1 位置显示

点击位置选项后进入位置界面。



①坐标值：显示当前各轴坐标，在关节坐标系下如果是旋转轴则单位是“度”，如果是直线轴则是“mm”；在笛卡尔坐标系下，位置(XYZ)单位是“mm”，姿态(ABC)单位是“度”；

②坐标系：界面底部有4个按钮：关节、世界、用户、机器人。点击不同的坐标系会切换到相应的坐标值；

③状态条：机器人当前状态，详细请看 [1.2.3 状态显示条](#)

④3D 模型：画面右侧为固定显示区域，3D 模型实时仿真与运动轨迹；

⑤程序内容：显示当前被加载项目的主程序内容，如果在运行程序则表示当前正在运行的程序的内容；

⑥时间：年/月/日 时/分/秒；

⑦当前运行程序：显示当前被加载项目的主程序，如果在运行程序则表示当前正在运行的程序号。

2.1.2 版本信息管理

控制器界面包括“概况”和“程序”。

概况界面主要显示控制器的一些“控制器信息”、“控制器运行时间”。



控制器信息：主要包含了版本信息，当前机构编号；**控制器运行时间：**主要记录了上次开关机时间，以及总的运行时间。

程序界面主要显示当前运行程序状态、程序调用栈信息等。



2.1.3 伺服扭矩监控

点击伺服选项后进入伺服界面，在概况界面可显示可以检测各轴负载情况，实时显示当前负载转矩。如下为 7 轴在线时的界面：



图中“伺服就绪”状态高亮显示表明伺服通讯正常，“伺服开关”状态高亮显示表明伺服使能已接通。

点击伺服 01~08 可显示各个轴的伺服信息。如下为离线模式时的界面：



2.1.4 示波器采样

点击示波器选项后进入如下界面。



示波器可以对所要监控的对象进行数据采集。触发类型包括自动触发、正常触发与单次触发三种，监控的对象通过点击【设置】来添加与删除，弹出如下对话框。



通讯周期与采集频率是不可修改的，时间宽度设定代表一个屏横向的时间轴长，缓存大小在正常触发、与单次触发中有效，在这两种方式下，一旦触发开始，达到缓存容量就会结束运行，而自动触发不管有没有达到触发条件，都显示波形，不会结束运行，不受缓存影响。

如何添加监控对象？首先在左侧选择“合成速度”项，通过“增通道”按钮添加到右侧监控对象列表，还可以添加 I/O 信号，最多添加 4 个监控对象。移除对象则需要在右侧列表选中，“增通道”变成了“删通道”按钮，选中即可移除。

设置好后按“确定”按钮，回到原来页面。在该界面左下角图标是开始采集按钮，按下后波形开始显示，正下方是可调节按钮，可以调节纵轴幅值、横轴的宽度、触发电平，右下角可用来进行平移操作。

点击画面下方的“显示波形”或“隐藏波形”按钮可以用来显示或隐藏波形。

点击“保存 png”按钮可以将波形保存为图片。点击“保存 txt”按钮可以将波形保存为 txt 格式，例如合成速度 txt 文件中可查看各时间对应的合成速度数值。

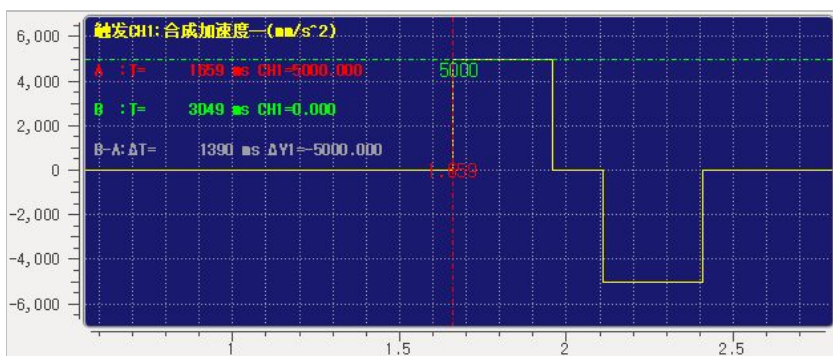
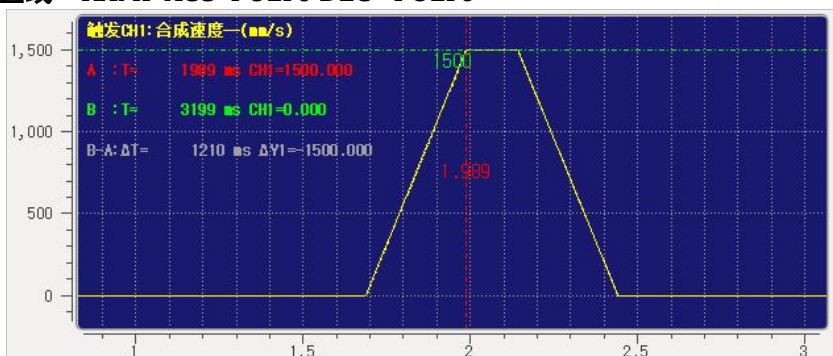
测量出结果后还可根据需要调整图形的显示，例如可进行调节幅值、宽度、平移等操作。

2.1.4.1 监测 RAMP 加速曲线类型

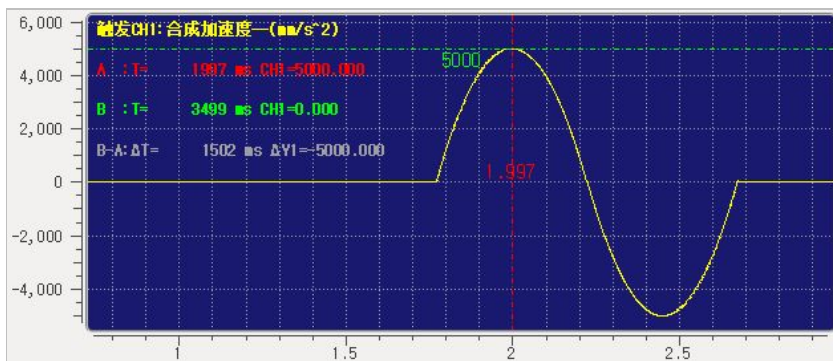
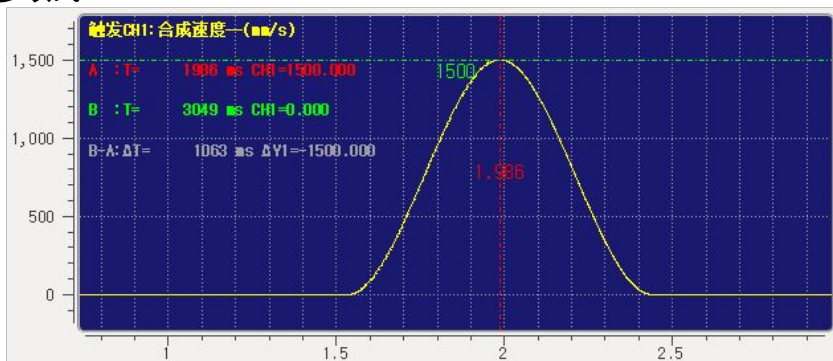
RAMP 设置加速度的加速类型。可设置类型有：直线 POLY0、多项式 POLY1、三角函数 0 SINO、三角函数 1 SIN1，通过采集速度、加速度数据反映四类加速度区别，下面以同一个程序举例，图形如下所示。

程序中速度 $V=1500\text{mm/s}$ ，加速度设置 5000mm/s^2 。

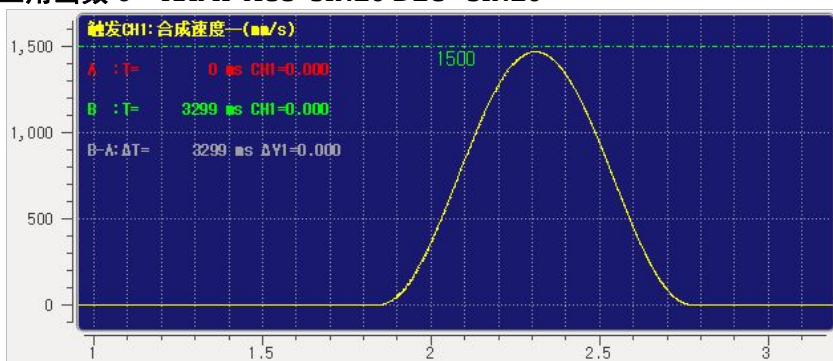
直线 RAMP ACC=POLY0 DEC= POLY0

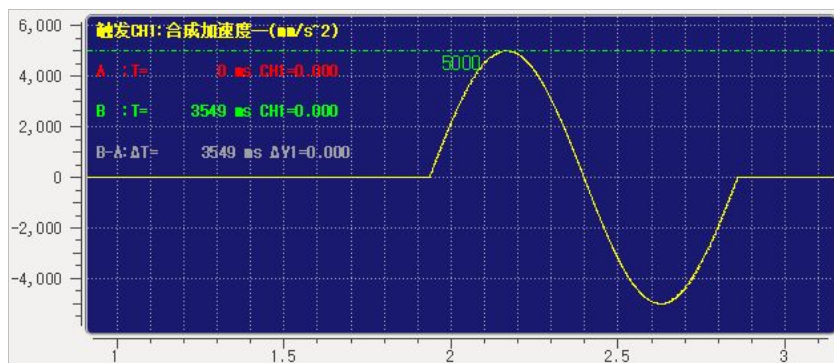


多项式 RAMP ACC= POLY1 DEC= POLY1

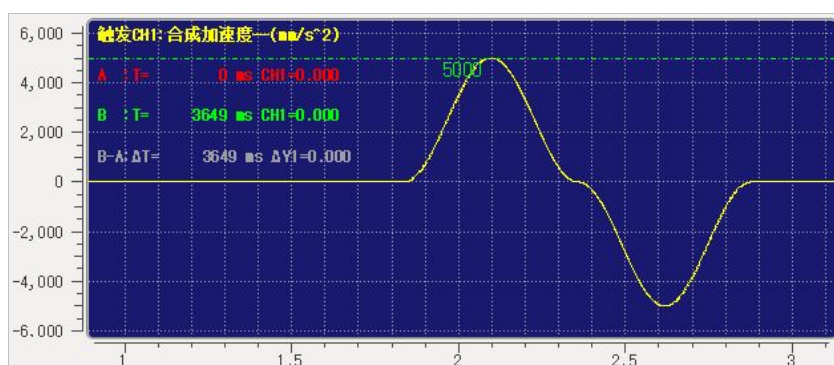
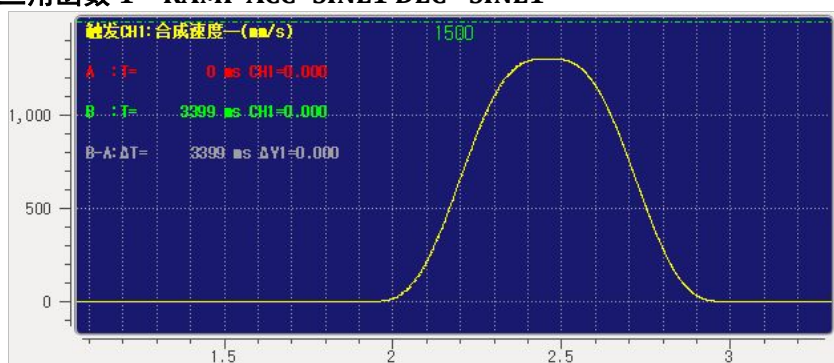


三角函数 0 RAMP ACC=SINE0 DEC= SINE0





三角函数 1 RAMP ACC=SINE1 DEC= SINE1



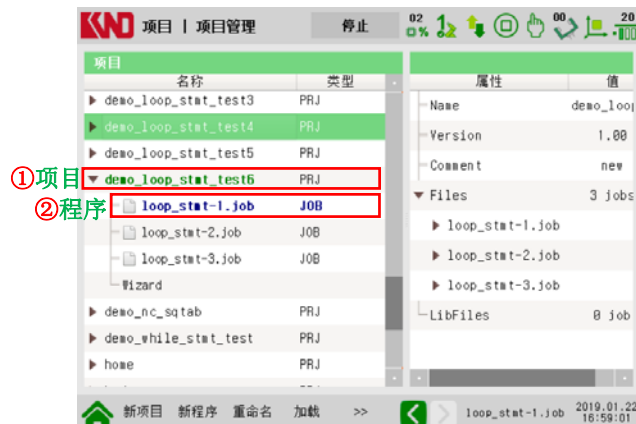
对比看出，相同的直线距离，直线 POLY0 加速类型加速最快，能够到达设定

速度并保持匀速，但是实际使用中需根据实际负载情况设置合理的速度与加速度，设置过快会引起不必要的故障。

2.2 项目管理

2.2.1 项目管理

选择左下角图标弹出如下图主菜单项。选择【项目】/【项目管理】。进入如下项目管理页面。



文件格式: 打开项目管理界面，该界面左侧列表显示当前存储的项目以及程序。右侧显示文件属性信息，如上图所示，类型“PRJ”是项目文件，类型“JOB”是程序文件，一个项目可包含多个程序文件，程序还支持格式“.nc”与“.cam”文件，可通过 CAM 软件转换导入，多用来抛光、打磨、焊接等点位密集使用专业机器人仿真软件代替人工示教处理，省时高效。

选中项目名称，界面右侧显示该项目下的所有程序，或者点击三角图标可展开项目下的程序名称列表。

项目加载: 只有加载的项目才可以运行程序，正常开机默认所有项目都是未加载状态，被加载的项目不允许“重命名”与“删除”操作，如果当前已经加载了一个项目，想要关闭当前项目加载状态，只需要选中需要继续被加载的项目，点击底部“加载”按钮即可对项目进行加载，同时关闭上一个被加载的项目。项目加载后项目名称会绿色高亮显示。

项目主程序: 上面说到只有加载的项目才可以运行，还要加一个限定条件就是

运行的程序必须是当前项目的入口主程序才可以，当一个项目中含有多个程序时，在运行前就需要设定哪一个为主程序，操作是选择想要运行的程序，并点击折叠符号选择“主程序”选项设置，可将程序设为主程序，程序设定主程序后，程序名称会以蓝色高亮显示。

项目中的主程序无法单独被删除，但是非加载项目可以整体被删除。



只有当前加载的项目的主程序才可以运行。所以在当前加载的项目下的程序谨慎修改，避免发生意外。还支持后台编辑功能，当再现模式运行程序时，除了严禁修改当前项目外，还是可以支持创建、编辑新项目以及新项目下的程序，不过这种使用情况不太常见



如上两张图中选择项目与选择程序，界面最下面一栏会有不同的操作选项。以下说明各选项如何使用。

新项目：引导创建新项目，名称不能使用中文，不能使用已有的名称，如果定义名称时字体为红色，表面此名称不合法，需从新定义。

新程序：在当前项目引导创建新程序，需注意，定义程序名称时需要在名称后面加“.job”后缀，否则字体始终为红色，导致不允许创建新程序。

编辑：进入程序编辑页面

重命名：放弃原来名字从新定义名称，当前加载的项目不允许重命名

牵引示教：基于力传感器应用的界面。可参考 [11.4 力传感器](#)

加载：加载新项目的同时关闭上一个被加载的项目

主程序：将选中的程序设置为项目的主程序，目的是为了运行当前选中的程序

删除：删除光标所选项目或者程序，当前加载的项目与主程序不允许删除

复制：复制项目则在本通道内复制相同内容的项目，并定义一个新的项目名称；复制程序则将选中的程序暂存在粘贴版中，直到下一次“复制”才会覆盖，需要继续选择“粘贴”，才能拷贝到项目中，也可将跨项目粘贴，但是需注意，跨项目复制程序不会复制变量、用户坐标系等里面的数值。

粘贴：将复制到剪切板的内容拷贝到另一个地方，对于重复编辑指令可大大节约编程时间，可以粘贴整个程序文件也可以粘贴程序的一段内容。

复制到另一个通道：复制到另一个通道名称与内容完全一样的项目

辅助 IO：定义该项目方案所用到的 io 地址，可自定义名称，更直观查看 io 状态方便调试

自定义变量：先定义变量别名，在程序中使用，增强程序的可读性可参考 [4.7 自定义变量](#)

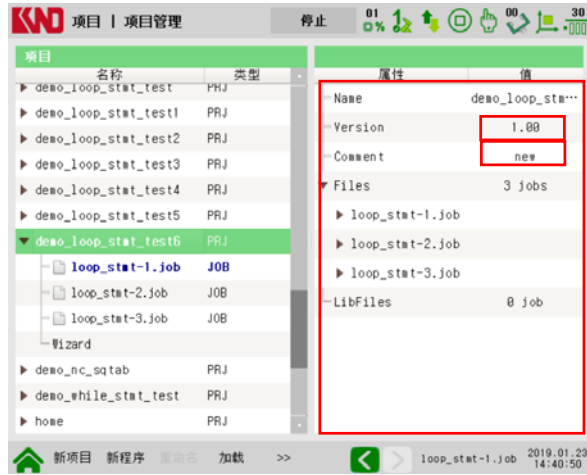
编辑 APD：目的是为了连续运动点位衔接的地方输出辅助动作同时运动平滑有效。

导入/导出项目：单独导入/导出一个项目文件到 U 盘中

2.2.1 属性管理

2.2.1.1 项目属性

选择项目，在界面右侧会显示项目相关信息，并可进行编辑。



选择“Version”一栏，在“值”一列中单击，变为可输入状态，可在输入框中输入项目的版本信息。如下图所示

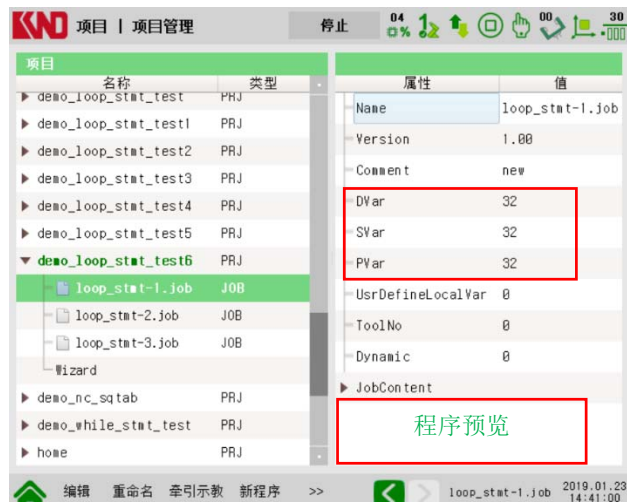
选择“Comment”一栏，在“值”一列中单击，变为可输入状态，可在修改框中输入项目的注释或描述，也可再次单击弹出全键盘，用于输入内容。



在“Files”一栏中会显示该项目包含的程序个数。各程序的“JobNo”一栏显示该程序的程序号，用户可进行编辑

2.2.1.2 程序属性

选择程序，在界面右侧会显示程序相关信息，也可以编辑程序的版本、备注，并可设置该程序用到的局部变量个数，如图所示



DVar (局部数字 LD 变量); SVar (局部字符串 LS 变量); PVar (局部位置 LP 变量), 默认各为 32 个, 数值可以修改, 范围在 (LX0~LX9999)。
在 jobContent 处可以显示程序预览, 点击三角图标即可查看。

2.2.3 程序页面管理

在项目管理页面点击“编辑”进入程序编辑页面, 界面如下所示:



 弹出折叠的指令选项。

返回: 返回到上一层项目管理页面

后/前插: 都是插入新指令，会弹出指令库，以光标所在行位置后/前插入

修改: 弹出指令库，修改光标选中的指令行

删除: 删除光标选中的指令行，可以一行或者多行同时选择删除

注释: 注释的程序不在译码执行，只是解释说明，前面会自动添加“//” 颜色统一是绿色，可以一行或者多行同时选择注释。

复制: 复制程序则将选中的程序暂存在粘贴版中，直到下一次“复制”或者“剪切”才会覆盖，需要继续选择“粘贴”，才能拷贝到项目中，也可将跨项目粘贴，可以一行或者多行同时选择复制

剪切: 将选中需要移动的程序暂存在剪切板中，原位置文件被删除，可以一行或者多行同时选择剪切

粘贴(后插/前插): 将剪切板的内容拷贝到另一个地方，以光标所在行位置后/前插入

查找: 输入想要查找的内容，并在程序中定位该位置。

跳转到 APD: 跳转到 APD 编辑页面，前提是在项目管理页面的确创建编辑了 APD，如果没有则不会跳转过去

辅助 IO: 与项目管理页面的辅助 IO 是同一窗口，定义该项目方案所用到的 io 地址，可自定义名称，更直观查看 io 状态方便调试

 返回到过去浏览的上一个或者下一页页面。

 **设置启动行** 设置程序启动行，程序指针  指向某行行号，运行时以程序指针  所在行号位置处开始执行。

指令库如下:

类别	名称
IO	MOV_C
控制	MOV_J
运动	MOV_L
操作	NOP
设置	END
系统	
循环	
条件	
其它	
历史	

☒ 更新位置坐标 位置类型: 关节

目标位置: PVAR_CONST C0000 位置常量

关节速度(%) 不使用

位置精度等级 不使用

非运动指令 不使用

运动结束条件 不使用

加速度(%) 不使用

减速度(%) 不使用

增量运动 不使用

APD 不使用

指令: MOV_J C0000

注释:

取消
确定

选择合适的功能指令，以及该指令附加项内容，指令详细介绍可参考《KND 机器人语言指令手册》。

2.3 变量管理

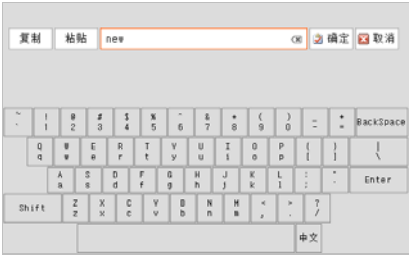
2.3.1 项目变量

2.2.2.1 数字型变量设定

项目变量页面包括数字、字串、位置，应用在程序中可对其读取修改，该变量只属于当前被加载的项目。



点击名称或者变量注释的空白处会弹出全键盘可对其修改。



手动输入数值：

- 1. 将光标移动变量号【值】的选项，该页面最多可显示 16 个数字变量，如果想要显示其他变量，只要在页面最上面 D 后面的白框里输入起始地址，即可显示该地址起之后的 8 个变量。

2. 选中后使其变成数值输入状态。
3. 用数值键输入数据。



项目变量可在程序中临时保存计数值、运算值和输入信号使用并且在程序中可自由定义。

由于同一项目多个程序使用同一个项目变量，故可在程序间信息的接收与传递中使用。

具体可用于以下几个用途：

- 工件个数管理。
- 作业次数管理。
- 程序间的信息接收与传递。

并且项目变量值即使在电源断开后，值被保持，再次开机仍可再次使用。

项目变量的种类见下表所示：

变量前缀	变量类型	数据范围	类型控制
D	数字变量（智能）	D0000~D9999	可为 U32/S32/U64/S64/DBL
S	字符变量	S0000~S9999	最大字符串长度 127
P	位置变量	P0000~P9999	XYZ 型/关节型

数据类型	最大值	表达式最小值	常量最小值	备注
S32	0x7FFFFFFF=2147483647	0x80000000=-2147483648	0x80000001=-2147483647	32 位有符号数
U32 (U)	0xFFFFFFFF=4294967295	0	0	32 位无符号数
S64 (L)	0x7FFFFFFFFFFFFFFF=9223372036854775807	0x8000000000000000=-9223372036854775808	0x8000000000000001=-9223372036854775807	64 位有符号数
U64 (UL)	0xFFFFFFFFFFFFFFFF=18446744073709551615	0	0	64 位无符号数
DBL	1.7976931348623158e+308	-1.7976931348623158e+308		IEEE754 浮点

2.2.2.2 字串型变量设定

1. 将光标移动变量号【值】的选项，该页面最多可显示 16 个字符串变量，如果想要显示其他变量，只要在页面最上面 S 后面的白框里输入起始地址，即可显示该地址起之后的 8 个变量。
2. 显示字符输入画面后，可输入阿拉伯数字、英文字母、汉字等作为字符。

2.2.2.3 位置型变量设定

1. 将光标移动变量号【值】的选项，该页面最多可显示 8 个位置 P 变量，如果想要显示其他变量，只要在页面最上面 P 后面的白框里输入起始地址，即可显示该地址起之后的 8 个变量。

2. 选择位置类型
包含 4 种类型 JOINT（关节坐标）、JOINT_DIST（关节相对坐标）、CARTE（笛卡尔坐标）、CARTE_DIST（笛卡尔相对坐标）
3. 设定变量根据选择类型不同，方法不同。

设定方法 位置类型	【示教】	手动输入数值
JOINT	可以	可以
JOINT_DIST	不可以	可以
CARTE	可以	可以
CARTE_DIST	不可以	可以

位置类型 JOINT、CARTE 可通过按钮  示教，把当前机器人位置记录到变量中，将轴移动到目标点，先选择变量类型，点击示教按钮即可，**需注意：直线插补 MOVL 与圆弧插补 MOVCL，位置变量类型必须是笛卡尔 CARTE 类型，关节插补 MOVJ 位置变量类型必须是关节。**类型 JOINT_DIST、CARTE_DIST 只支持手动输入，其作用是在上一个程序点基础上相对移动，保证一定的相对距离，往往可在平移运动中作为增量使用。

2.3.2 局部变量

2.2.2.4 局部变量的编辑

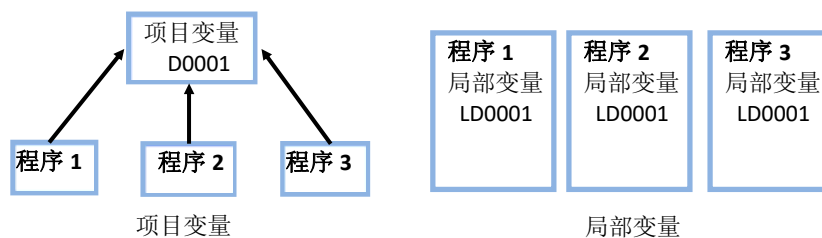
局部变量与项目变量同样，可在保存计数值、运算值、输入信号时使用。数据形式与项目变量一样。局部变量号码带“L”。

变量前缀	变量类型	数据范围	类型控制
LD	数字变量（智能）	由当前程序头指定 LD0000~LD9999	可为 U32/S32/U64/S64/DBL
LS	字符变量	由当前程序头指定 LS0000~LS9999	最大字符串长度 127

LP	位置变量	由当前程序头指定 LP0000~LP9999	XYZ 型/关节型
----	------	---------------------------	-----------

与项目变量的不同之处：

- 只在一个程序中使用。
项目变量时，多个程序可以定义或使用项目变量中的一个变量。而局部变量时，只能由定义的程序使用局部变量中的一个变量，其他程序不能读写。
所以局部变量不能影响到其他程序，如：LD0001 这一局部变量可由多个程序单独定义或使用。

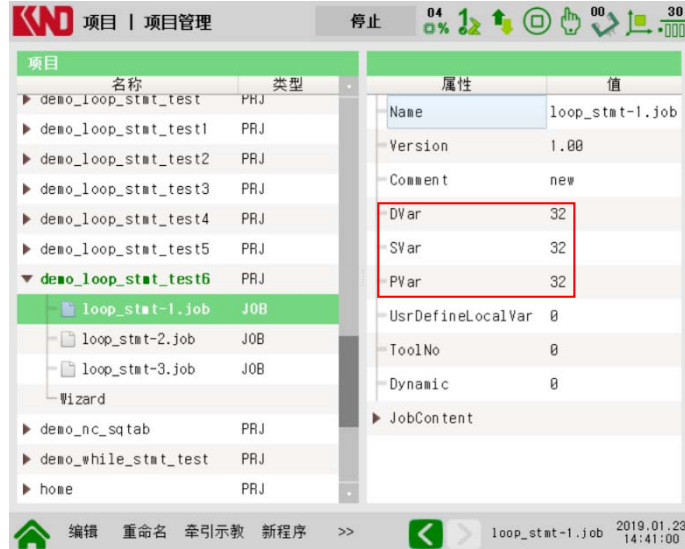


- 局部变量可自由设定使用几个。
在程序管理页面设定局部变量个数，个数一旦设定，该个数所占的内存量就会得到保证。
- 变量的内容显示需要使用项目变量。
若要查看局部变量 LD0000 时，先将该变量保存在项目变量中 D0001 中，执行该命令后，看变量 D0001 数字变量的页画面。
<D0001=LD0001>
- 局部变量的内容只在执行定义的程序中有效。
当定义局部变量的程序被调用（用 CALL 指令调用程序）时，局部变量所需内存量会得到保证。
执行程序后，由于 RET、END 的执行，一旦被退出，被设定的局部变量内容就会无效。但是使用局部变量的程序用 CALL 指令呼叫其他程序，用 CALL 返回时，可继续使用 CALL 执行前的内容。

2.2.2.5 局部变量个数设定

程序中使用的局部变量的个数在项目管理画面设定。个数一旦设定，该个数变量所需的内存量就会得到保证。

如下所示，选择程序文件，即可看到使用的局部变量个数，默认的数字局部变量、字符局部变量、位置局部变量均为为 32 个，也可以对个数进行修改。如果程序使用的局部变量个数超过设定的局部变量个数，运行程序时会有局部变量号超出范围报警。



2.3.3 系统变量

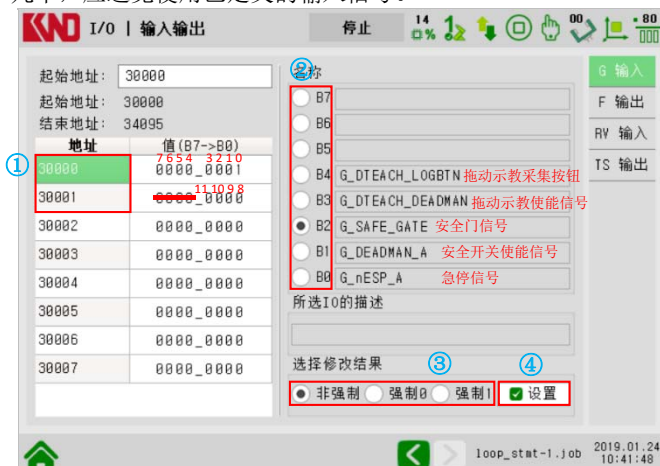
选择左下角图标  选择【I/O】/【系统变量】进入系统变量页面，包括数字、字符串、位置，可查看读取但是不可修改。系统变量在所有项目之间共享。用于控制器内核与 job 文件交互信息。已定义的变量均为固定用途。一些接口数据如并行控制器，视觉接口等已定义在\$D 变量中，可具体参考功能文档查看具体接口定义。此处不再附图片，可直接在示教器界面查看。

2.4 I/O 管理

2.4.1 输入输出

选择左下角主图标，选择【I/O】/【输入输出】，弹出如下页面，其中输入输出包括G输入、F输出、RV输入、TS输出。

选择**【G 输入】**，页面显示如下：图中可以看到，标准中 G 输入信号已经被占用定义了几个，应避免使用已定义的输入信号。



控制器的输入信号 G 接口状态，是通过 PLC 题图中将 X 地址映射过来，硬件标准配置中输入 12 个，硬件接口板编号是 DI(0~11)，界面显示输入 G 地址是 30000 与 30001 两个字节，如图中红色阿拉伯数字，这样就剩下了 4 位地址，处理方式是 30001 字节中的高四位地址舍弃。舍弃就是当扩展新的 IO 时，新的输入地址就从 30002 开始。

G 输入页面管理，可以强制输入状态，便于调试使用，被强制的地址不在随接口输入 X 的实际状态而变化，调试结束后要取消强制状态，正常工作的设备要谨慎此操作，避免发生意外故障导致财产或者人身损失。步骤如下：

①②要选对强制的地址，比如选择 30000 里的 B2 位，“安全门检测信号”，③选择强制状态，选择  强制，④点击  设置 按钮，弹出确认弹窗，选择“确定”，则 30000 里的 B2 位被强制位 1，字体是红色。



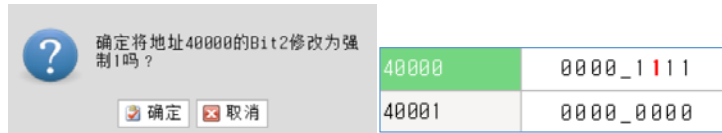
选择右侧【F 输出】，页面显示如下：图中可以看到，标准中 F 输出信号已经被占用定义了几个，应避免使用已定义的输入信号



控制器的输入信号 F 接口状态，PLC 题图又将 F 输出映射到 Y 接口，硬件标准配置中输出 12 个，硬件接口板编号是 DO (0~11)，界面显示输入 G 地址是 40000 与 40001 两个字节，如图中红色阿拉伯数字，这样就剩下了 4 位地址，处理方式是 40001 字节中的高四位地址舍弃。舍弃就是当扩展新的 IO 时，新的输出地址就从 40002 开始。

F 输出页面管理，可以强制输出状态，便于调试使用，正常工作的设备要谨慎此操作，避免发生意外故障导致财产或者人身损失。步骤如下：

①②要选对强制的地址，比如选择 40000 里的 B2 位，“未定义”，③选择强制状态，选择 **强制1**，④点击 **设置** 按钮，弹出确认弹窗，选择“确定”，则 40000 里的 B2 位被强制为 1，字体是红色，对应的输出接口被强制输出。



2.4.3 通道通信接口 RV

选择【RV 输入】，页面显示如下：

KND I/O | 输入输出 停止 14 00 80

起始地址: 50000
起始地址: 50000
结束地址: 59999

地址	值 (B7->B0)
50000	0000_0000
50001	0000_0000
50002	0000_0000
50003	0000_0000
50004	0000_0000
50005	0000_0000
50006	0000_0000
50007	0000_0000

名称

- ☐ B7
- ☐ B6
- ☐ B5
- ☐ B4
- ☐ B3
- ☐ B2
- ☐ B1
- ☒ B0

所选I/O的描述

选择修改结果

☐ 非强制 ☐ 强制0 ☐ 强制1 ☒ 设置

G 输入
F 输出
RV 输入
TS 输出

loop_stat-1.job 2019.01.24 10:42:05

2.4.4 通道通信接口 TS

选择【TS 输出】，页面显示如下：

KND I/O | 输入输出 停止 14 00 80

起始地址: 60000
起始地址: 60000
结束地址: 69999

地址	值 (B7->B0)
60000	0000_0000
60001	0000_0000
60002	0000_0000
60003	0000_0000
60004	0000_0000
60005	0000_0000
60006	0000_0000
60007	0000_0000

名称

- ☐ B7
- ☐ B6
- ☐ B5
- ☐ B4
- ☐ B3
- ☐ B2
- ☐ B1
- ☒ B0

所选I/O的描述

选择修改结果

☐ 非强制 ☐ 强制0 ☐ 强制1 ☒ 设置

G 输入
F 输出
RV 输入
TS 输出

loop_stat-1.job 2019.01.24 10:42:13

TS/RV 实现两个通道间的信号传输，TS 接口是发出端，RV 接口是接收端。

2.5 常规设置

选择左下角图标选择【设置】/【其它】进入页面。在其它页面可进行相关设置如下。



①**始运行倍率**：选择【倍率-】或【倍率+】，可调节试运行的倍率。倍率范围从10%~150%。与状态栏的试运行倍率显示一致。

②**伺服参数同步**：可进行控制器和伺服参数同步，在初调试或者修改伺服参数时会出现“伺服参数不一致”，需要此操作，在同步时注意同步方向，【**控制器→伺服**】是将控制器的伺服参数同步到伺服驱动器；【**伺服→控制器**】是将伺服驱动器参数同步到控制器；

③**重启**：主要作用是重启控制器与示教器，无需合闸断电，属于软重启。

④**HMI 设置**：【**屏幕**】设定屏幕进入省电模式时间，在设定时间内无任何操作，屏幕进入屏保模式，或者可以点击“关闭屏幕”，直接进入屏保模式，需要通过点击屏幕任何地方或者按键操作唤醒屏幕；【**屏幕校准**】可以校准点击屏幕的准确度，根据提示的规定时间，点击屏幕依次出现的目标点，完成校验；【**连接**】用户可修改控制器与示教器的以太网接口 IP 地址方便与其他设备的接入。

⑤**伺服挂载**：伺服轴连接挂载情况，正常情况下，实际伺服连接数量与此处挂载勾选应保持一致，当某些轴出现故障，或者未真实连接伺服时，可以将该轴挂载取消以方便调试。

⑥**运行模式**：正常机器人连接使用时一定是勾选“在线”，如果勾选“离线”模式，此时与伺服处于失联状态，示教器的轴移动操作不会影响到本体移动，再次勾选“在线”即可恢复与伺服正常通讯，之前的轴移动不再影响。

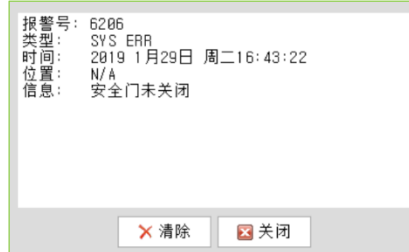
⑦**输出状态**：当前模式允不允许“DO+AO 输出”，勾选表明允许相反则禁止。

2.6 报警报告处理

若发生报警，机器人停止动作，但一些低级别警告报警，会有提示但不会切断伺服能停止作业中机器人的运动，并且在状态条显示区有闪烁的报警提示符。



点击提示符会弹出报警内容。



解除：

- 轻故障报警
只需要按复位即可清除报警，报警符消失。
- 重故障报警
需要断开主电源，排除故障原因。
若发生重大的硬件故障时，会自行切断伺服电源，停止机器人动作。需要彻底排除问题，才可复位。

打开报警页面，可查看机器人“当前报警”与“历史报警”内容。

选择左下角主图标，选择【监视】/【信息】，进入“报警”“日志”监视页面如下：



在信息页面可显示**当前报警**、**历史报警**、**CNC 日志**、**HMI 日志**。

当前报警：显示未断电前的最近一次报警，报警信息包含：**报警号**、**类型**、**时间**、**报警信息**。

报警号：故障报警是 6000+；报警类型显示“SYS ERR”

警告报警是 3000+；报警类型显示“SYS WARN”

时间：显示在什么时间触发的报警

信息：显示报警原因，什么导致的报警。

对于一些轻微的警告，可通过“清警告”按钮清除，而一些硬件故障的报警，则需要彻底排除后才可清除。

在**历史报警**页面，可查看到历史报警，也可以导出最近一段时间的报警历史记录，首先将 U 盘插到示教器 usb 接口上，点击画面下方“导出告警”按钮，弹出保存历史报警对话框，选择 U 盘保存路径，如下图选择文件夹“KNDRobot”并点击“确定”，可将历史报警导出。



导出成功提示完成窗口，在文件夹“KNDRobot”有 CNC1Alm.log 记录，以供查看。



在 CNC 日志页面，可查看到控制器日志信息，点击画面下方“导出日志”按钮，弹出保存控制器日志对话框，选择保存路径并点击“确定”，可将控制器日志导出。

在 HMI 日志页面，里可供查看最近的操作履历，当遇到出现异常问题时查看使用。

3 Robot 坐标系统的概念与轴操作

机器人坐标系：

工业机器人的作业点在空间通过坐标进行标识，通过选取不同的坐标系可以快速有效地进行机器人工作系统示教，提高作业效率。

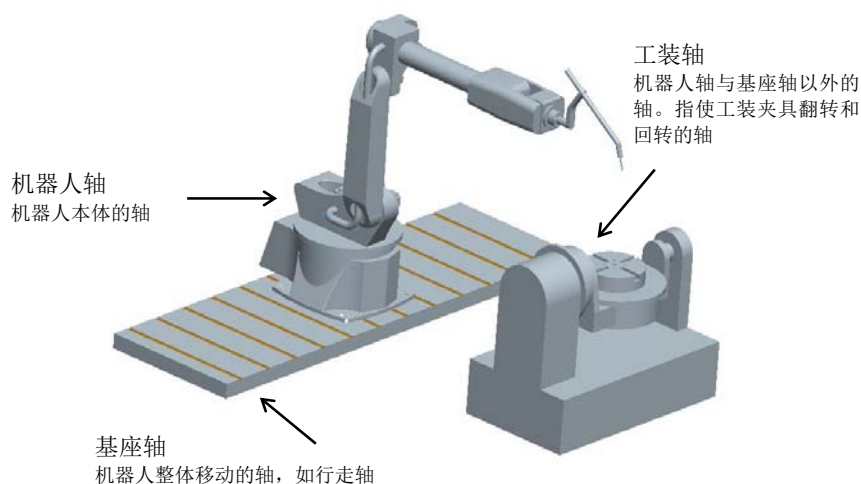
3.1 坐标系统的概念

规定把固定在地面上的坐标系称为**世界坐标系**。对于固定安装的机器人，当安装完成后，坐标系之间的对应关系即唯一确定。

3.2 控制组与坐标系

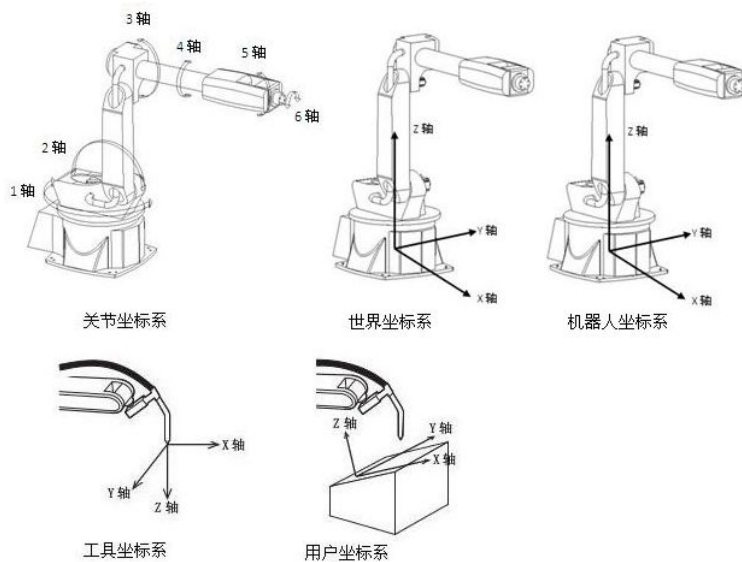
KND 将单轴或多轴的操作称为“控制组”，机器人本身的轴称为“机器人轴”，使机器人整体平行移动的轴叫“基座轴”，除此之外还有“工装轴”，配合夹具和刀具的使用。

另外，基座轴、工装轴还叫外部轴。



对机器人本体“机器人轴”操作时，其坐标系还有以下几种形式。

1. 关节坐标系：机器人本体各轴单独运动。
2. 世界坐标系：机器人前端沿设定的 X 轴、Y 轴、Z 轴平行移动。
3. 机器人坐标系：机器人前端沿设定的 X 轴、Y 轴、Z 轴平行移动。
4. 工具坐标系：工具坐标系把机器人腕部工具的有效方向作为 Z 轴，把 XYZ 世界坐标系定义在工具的尖端点。本体尖端点根据坐标平行运动。
5. 用户坐标系：XYZ 直角坐标在任意位置定义。本体尖端点根据坐标平行运动。



3.3 基本轴操作步骤

3.3.1 安全确认

操作前，排除操作对象机器人及周边设备对周围环境带来的潜在危险，确保安全。

3.3.2 示教模式选择

将手持示教编程器模式旋钮转到【TEACH】示教模式。

3.3.3 控制组的选择

当控制组为多个系统或协调系统（选项）时，首先选择要操作的对象控制组。

示教器上有个  键，用来切换想要操作的 1 或 2 通道的机器人。
请在状态显示区对即将操作的控制组进行确认。

3.3.4 坐标系的选择

按  坐标系键，选择要操作对象的坐标系：
关节—世界—工具—用户—机器人。

每按一次  键，就切换一次，请在状态区进行确认。



3.3.5 速度选择

可按示教器调节速度的【高】或【低】键，选择轴操作时的手动速度。



每按一次【高】键，按照微动—低—中—高顺序变化。

每按一次【低】键，按照高—中—低—微动顺序变化。

3.3.6 接通伺服

解除急停状态，检查急停按钮（示教器、配电柜）处于旋开状态，状态区无急停报警显示。



握住安全开关。
（伺服接通的 LED 灯常亮）

3.3.7 轴操作

请再次确认机器人周边的安全。

在此状态下，按轴操作键，轴动作按照选择的控制组、坐标系、手动速度、轴操作键进行运动。

3.4 坐标系与轴操作

3.4.1 关节坐标系

在关节坐标系，机器人本体各轴单独运动。

当按下机器人没有的轴操作键时，不作任何动作。

各轴动作关系见如下所示：

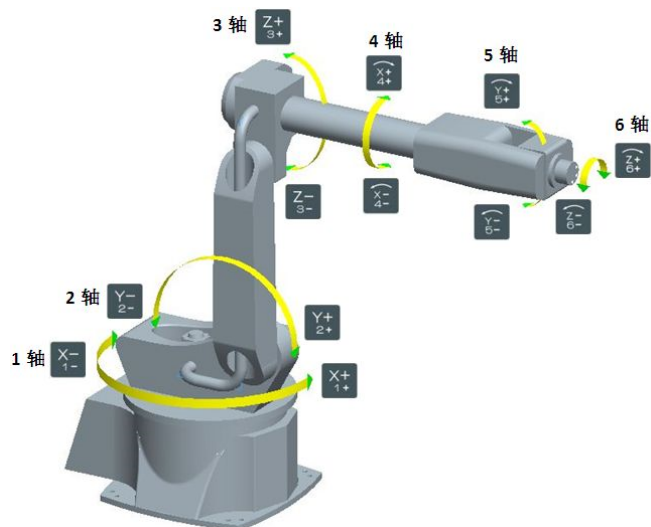
轴名称	操作轴		动作
1 轴			本体左右旋转
2 轴			下臂前后运动
3 轴			上臂上下运动
4 轴			上臂旋转
5 轴			手腕上下运动
6 轴			手腕旋转

参考：

关节坐标系的轴操作

当同时按 2 个以上的多个轴操作时，机器人呈合成式运动。

但是，像【X-】+【X+】这样同轴反向方向的 2 个键同时按下时，所有轴不动。



3.4.2 世界坐标系

在世界坐标系，机器人前端沿设定的 X 轴、Y 轴、Z 轴平行移动
各轴动作关系见如下所示：

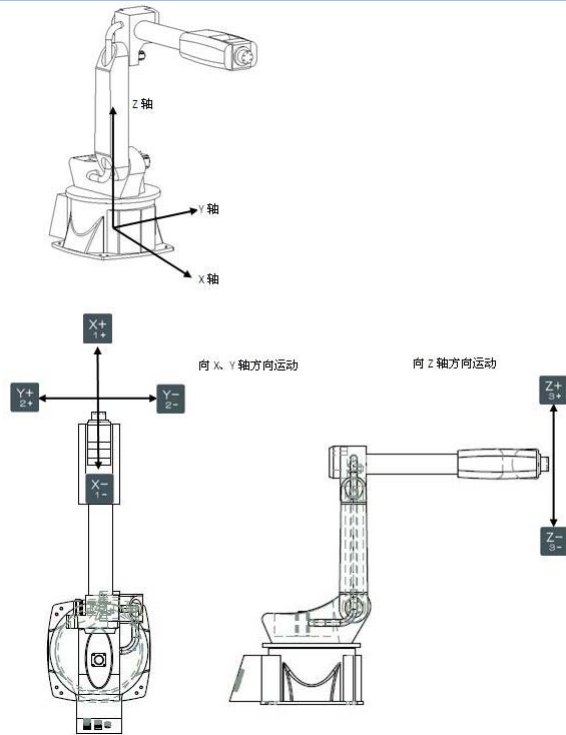
轴名称		操作轴	动作
基本轴	X 轴	X ₁ ⁻ X ₁ ⁺	沿 X 轴平行运动
	Y 轴	Y ₂ ⁻ Y ₂ ⁺	沿 Y 轴平行运动
	Z 轴	Z ₃ ⁻ Z ₃ ⁺	沿 Z 轴平行运动
手腕轴		手腕动作时控制点保持不变，沿 X、Y、Z 轴中心回转	

参考:

世界坐标系的轴操作

当同时按 2 个以上的多个轴操作时，机器人呈合成式运动。

但是，像【X-】+【X+】这样同轴反向方向的 2 个键同时按下时，所有轴不动。



3.4.3 工具坐标系

在工具坐标系，机器人沿定义在工具尖端点 X、Y、Z 轴平行运动。
各轴动作关系见如下所示：

轴名称	操作轴	动作
-----	-----	----

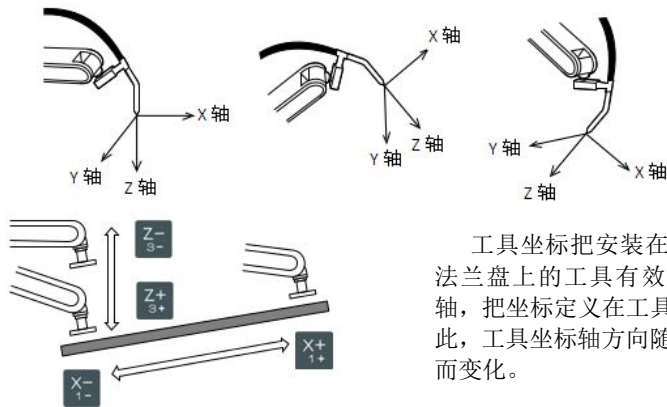
基本轴	X 轴	 	沿工具坐标系 X 轴平行运动
	Y 轴	 	沿工具坐标系 Y 轴平行运动
	Z 轴	 	沿工具坐标系 Z 轴平行运动
手腕轴		手腕动作时控制点保持不变，沿 X、Y、Z 轴中心回转	

参考：

工具坐标系的轴操作

当同时按 2 个以上的多个轴操作时，机器人呈合成式运动。

但是，像【X-】+【X+】这样同轴反向方向的 2 个键同时按下时，所有轴不动。



工具坐标把安装在机器人腕部法兰盘上的工具有效方向定为 Z 轴，把坐标定义在工具尖端点。为此，工具坐标轴方向随腕部的动作而变化。

3.4.4 用户坐标系

在用户坐标系，在机器人动作范围的任意位置，设定任意角度的 X、Y、Z 直角坐标，本体尖端点根据该坐标轴平行运动。

各轴动作关系见如下所示：

轴名称		操作轴	动作
基本轴	X 轴	 	沿用户坐标系 X 轴平行运动

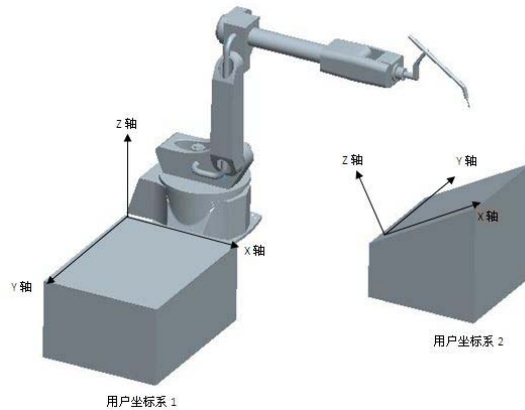
	Y 轴	 	沿用户坐标系 Y 轴平行运动
	Z 轴	 	沿用户坐标系 Z 轴平行运动
手腕轴		手腕动作时控制点保持不变，沿 X、Y、Z 轴中心回转	

参考：

用户坐标系的轴操作

当同时按 2 个以上的多个轴操作时，机器人呈合成式运动。

但是，像【X-】+【X+】这样同轴反向方向的 2 个键同时按下时，所有轴不动。

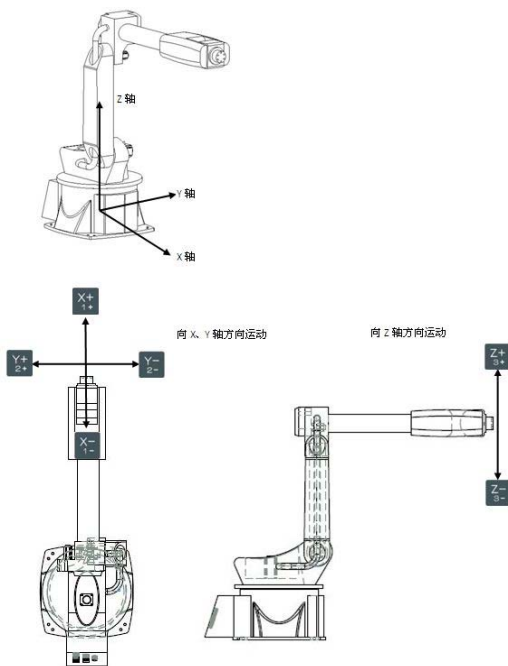


3.4.5 机器人坐标系

在机器人坐标系，机器人前端沿设定的 X 轴、Y 轴、Z 轴平行移动。
各轴动作关系见如下显示：

轴名称		操作轴		动作
基本轴	X 轴	 		沿 X 轴平行运动
	Y 轴	 		沿 Y 轴平行运动
	Z 轴	 		沿 Z 轴平行运动

手腕轴	手腕动作时控制点保持不变，沿 X、Y、Z 轴中心回转
-----	----------------------------



参考：

机器人坐标系的轴操作

当同时按 2 个以上的多个轴操作时，机器人呈合成式运动。

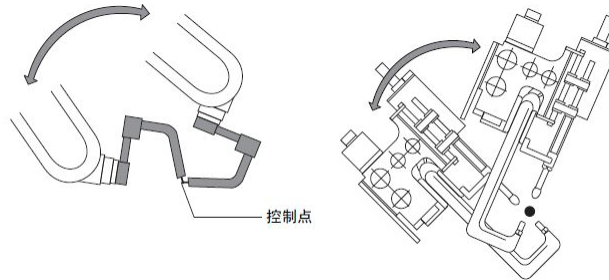
但是，像【X-】+【X+】这样同轴反向方向的 2 个键同时按下时，所有轴不动。

3.4.6 机器人控制点保持不变的操作

机器人在作业过程中，经常需要围绕同一个点变换位置，调整姿态，以适应作业对象工艺或动作要求。控制点不变的操作是指不改变工具尖端点的位置(TCP 控制点)，只改变工作姿态的轴操作。

控制点保持不变的操作，在世界坐标、机器人坐标、工具坐标和用户坐标系下均可使用。当选择相应的坐标系进行轴操作时，工具围绕控制点以相应的坐标系轴

进行旋转。下图是控制点不变，围绕控制点选择的示意图。关节坐标下不能进行控制点保持不变的操作。

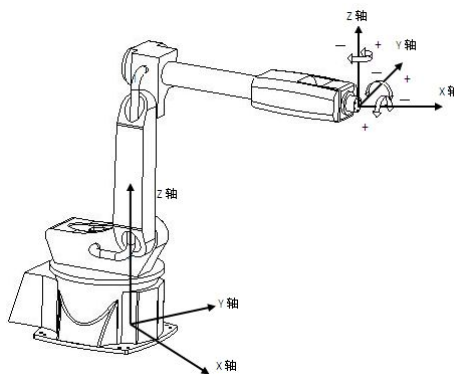


控制点保持不变的轴操作：

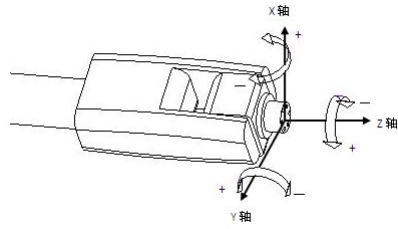
轴名称		操作轴	动作
手腕轴	4 轴	$\begin{matrix} \curvearrowleft \\ X- \\ 4- \end{matrix}$ $\begin{matrix} \curvearrowright \\ X+ \\ 4+ \end{matrix}$	使控制点位置保持不变，只有工具姿势改变。围绕指定坐标系的坐标轴运动中，工具姿势变化。
	5 轴	$\begin{matrix} \curvearrowleft \\ Y- \\ 5- \end{matrix}$ $\begin{matrix} \curvearrowright \\ Y+ \\ 5+ \end{matrix}$	
	6 轴	$\begin{matrix} \curvearrowleft \\ Z- \\ 6- \end{matrix}$ $\begin{matrix} \curvearrowright \\ Z+ \\ 6+ \end{matrix}$	

在控制点不变的操作中，由于选择的坐标系不同。因此手腕的回转也各异。

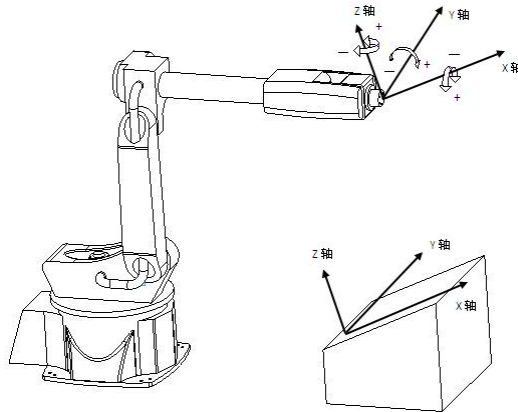
- 在世界/机器人坐标系中
以本体轴的 X、Y、Z 为基准，做回转运动。



- 在工具坐标系中
以工具坐标的 X、Y、Z 轴为基准，做回转运动。



- 在用户坐标系中
以用户坐标的 X、Y、Z 轴为基准，做回转运动。



4 示教编程操作

参考：

本章节涉及到的所有操作都是在示教模式下进行，请勿混淆。

4.1 示教前准备

出于安全上的考虑，示教前请实施以下操作。

第一步：确认机械手工作区没有障碍物干涉。

第二步：确认急停按钮是否正常（两个急停：一个在电气柜里，一个在示教编程器上）。

第三步：将模式设定为“示教”模式.

第四步：创建新的程序。

4.1.1 确保机械手活动范围安全

在通用示教过程中，在定下示教点位置时，一般都需要机械手通过轴操作移到该位置，虽然可以设置干涉区域来保护安全，与周围设备互不干涉，但是示教前我们还是要确保机械手工作区没有障碍物的干涉，以免发生意外碰撞事故。

4.1.2 急停按钮使用确认

在使用机器人时，请分别对电气柜、示教编程器上的急停按钮进行确认。

1. 只有在急停按钮没有被按下的时候才能给伺服通电时，要检查急停（电气柜与示教编程器上的急停）都要处于放开状态才可以。
2. 按急停按钮（电气柜或示教编程器上的急停）都会切断伺服电源，“伺服准备”灯灭。所以在出现紧急情况时可以使用急停切断伺服电源，使机械手停止动作。

4.1.3 关于示教模式安全保障

出于安全考虑，示教时，请务必将示教器上的模式旋转到“TEACH”。若在设定“TEACH”时，错误的将“START”键按下，即使从外部输入开始信号，也不会变成再现状态。

4.1.4 如何创建新的程序

参考:

只有在“TEACH”模式下才可以创建、编辑程序。

创建新的程序

1. 选择【项目】菜单下的子菜单【项目管理】。

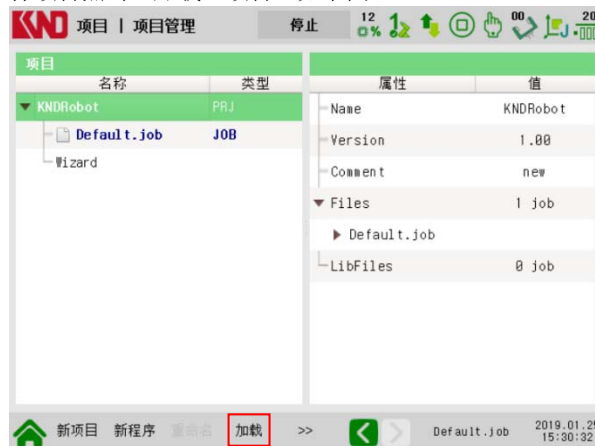


2. 选择【新项目】，弹出全键盘，输入项目名称“KndRobot”。



3. 按“确定”。

4. 新项目创建完成，并自动创建一个名字为“Default.job”的程序。记得选择项目点击“加载”项目。如下图：



在该页面下还可以对项目以及项目所属程序进行删除、重命名、复制等操作。需要运行程序时其项目本身需要先加载，程序设置为主程序才可以运行。

当前加载的项目在自动运行时禁止修改。如果不是当前加载的项目，任何时候都可以修改，在自动运行时也可修改。

参考：

项目和程序命名：

项目和程序名称支持英文字母、数字、下划线_、中横线-及其组合，不支持其他字符。以 Sys 开头的程序为系统生成的程序，用户新建程序无法使用 Sys 开头。

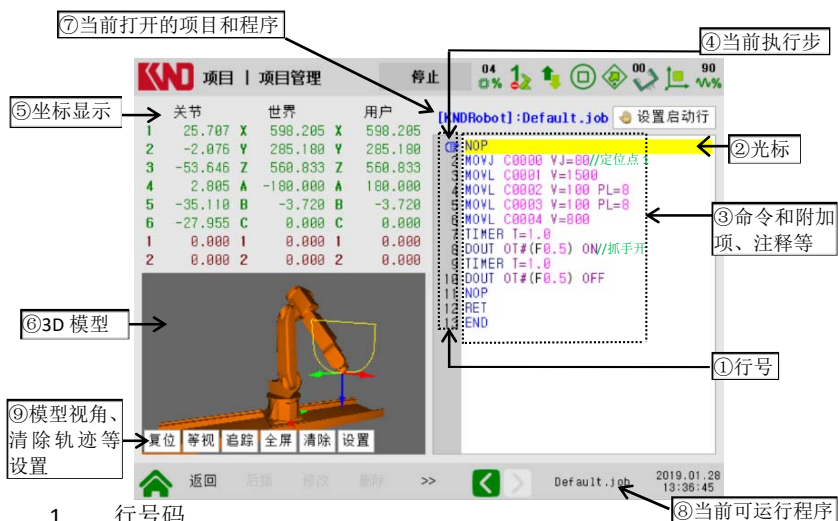
4.2 示教

4.2.1 示教编程界面

选中新建程序 Default.job，点击“编辑”按钮，进入如下编辑界面。程序的标准开头、结尾如下，在新建程序时自动编入。



示教后程序内容显示如下：



1. 行号码
显示程序行的号码，为自动显示，添加或删除行后，行号码自动被改写。
2. 光标
用于命令编辑的光标。
使用【前插】、【后插】、【修改】、【删除】进行程序内容命令的添加，更改和删除操作。
3. 命令和附加项、注释。

MOVJ C0004 VJ=100 //关节插补方式

命令 添加项 注释

命令：功能指令，用来进行指示的处理和作业的执行。

附加项：根据命令种类，进行速度、时间等设定。

注释：根据需求每一行指令后都可以添加相关注释。

4. 当前执行步
无论是示教，还是再现程序，小手标志代表当前正在被执行的指令行，用来辅助查看当前程序运行情况。
5. 坐标显示
实时显示机器人坐标数值，包括各轴关节、世界、用户坐标。
6. 3D 模型
机器人动画仿真模型，能够实时显示机器人动态、作业姿态、程序轨迹图形。
7. 当前打开的项目和程序。如果项目名与程序名相同时，简化显示为一个名称。
8. 当前可运行程序名。如果是主程序在执行过程中调用子程序，此时会显示正在执行的子程序名。
9. 在模型区域点击，可选择观察模型的视角，并可以选择清除生成的轨迹。

4.2.1.1 命令组的说明

命令按照不同的功能类型，被分为不同的指令组：

指令组	功能概况	命令
IO 指令	执行输入输出、模拟电压控制指令。	DOUT、DIN、AOUT、AIN、PULSE
控制指令	控制程序执行流程。	ABORT、CALL、GETA、GETR、JUMP LABEL、PAUSE、RET、TIMER、WAITIO、WAITMTN
运动指令	与移动或速度有关的指令。	MOVL、MOVCL、MOVJ、NOP、END
运算指令	读、写位置变量 P 指定元素坐标值。进行变量或常量的运算。	GETE、SETE、内置标准函数、三角函数、反三角函数、常用数学函数
设置指令	对加减速类型、速度、参考坐标系、工具以及区域的设定。	RAMP、REFSYS、SPEED、TOOL、AREA

系统指令	显示输出、读写系统参数。	ALM、CLS、GETPRM、PRINT、SETPRM
循环	复合循环	WHILE...END_WHILE;LOOP...END_LOOP
条件	条件跳转	IF...ELSE...ELSIF...END_IF
其他	写入表达式。	EXPRESS

4.2.2 运动插补介绍

移动命令的插补方法与再现速度种类：

再现运行机器人时，决定两个程序点之间以何种轨迹移动的方法叫**插补方法**。

程序点与程序点之间的移动速度就是**再现速度**。

通常**位置数据、插补方法、再现速度**这 3 个数据同时被记录到机器人轴的程序点中。

示教时，若省略插补方法或再现速度，会自动使用与上一次完全相同的设定。

4.2.2.1 关节插补（MOVJ）

机器人向目标点移动过程中，在不受轨迹约束的区间使用关节插补。

若用关节插补示教机器人轴，移动命令是 **MOVJ**。

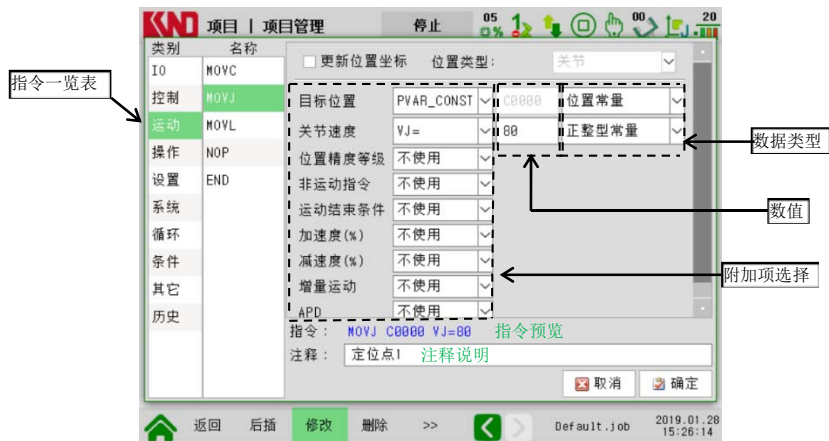
出于安全考虑，通常情况下，请用关节插补示教第一步。

关节插补的再现速度 **VJ** 的设定：（%）

显示相对于最高速度的比例

若速度被省略，被默认为事先设定的速度。

- 如果是第一次插入指令，点击【后插】，弹出指令编辑页面，在类别中选择【运动】，再选择“**MOVJ**”，右侧会显示与之匹配的附加项，【不使用】表明该添加项是可选择使用，指令详细介绍请见《KND 机器人语言指令手册》，【关节速度】代表再现速度，点开  选择 **VJ**，选择好数据类型后，再选择数值框，赋予想要的比率数值。
- 想要修改之前指令过的速度，需要在程序页面通过光标选中该指令行，再选择【修改】，想要修改的内容被显示，然后选择修改想要的速度。

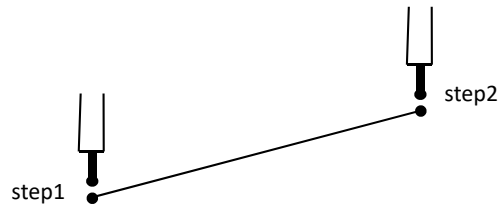


4.2.2.2 直线插补 (MOVL)

用直线轨迹向目标程序点移动的插补方式。

若用直线插补示教机器人轴，如下图：移动命令是 MOVJ。

直线插补再现速度需要设定（与圆弧插补通用）。



参考：

位置等级的使用

在使用移动命令 MOVJ（关节插补）、MOVL（直线插补）、MOVJ（圆弧插补）时可添加。

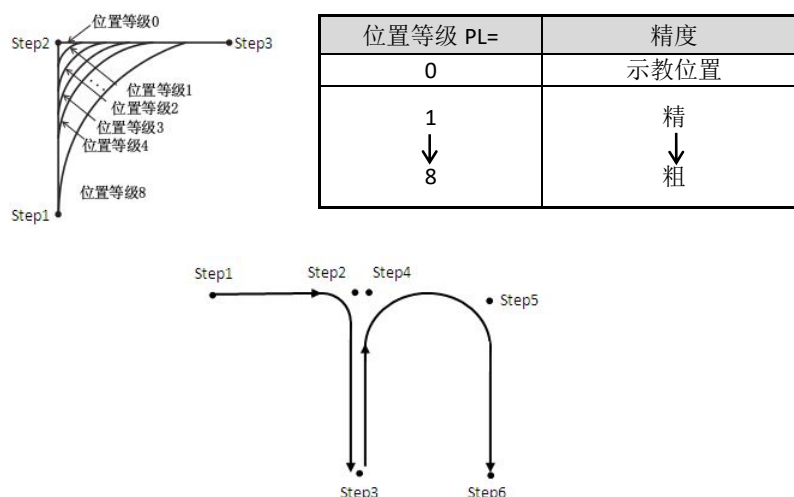
在指令编辑时，附加项 PL 即为等级设置，范围 1-8。

参数 P0501~P0508 可以修改定位等级 1~8 区间距离（mm）。

与位置等级的接近程度的设定，是指机器人通过示教位置时，实际位置与示教位置的接近程度。

以下显示的是位置等级的轨迹与精度之间的关系。

示例：考虑如下图程序点位置，该怎样动作。



运动步 step2、step4、step5 目标点仅仅是经过点，不需要准确的定位。并且，若在这些程序点上添加 PL=1~8，就会变成内循环运动，可缩短步与步之间启停节拍。请在运动步 step3、step6 的目标终点这种需要准确定位的地方，默认不使用 PL 或者添加 PL=0。

运动步 step2、step4、step5 目标经过点：

MOVL C000n V=100 PL=4。

运动步 step3、step6 的目标定位点：

MOVL C000n V=100 或者 MOVL C000n V=100 PL=0。

注意：

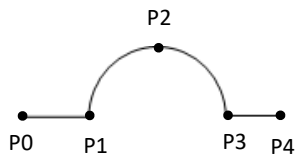
如果 step1 后有非运动指令执行，那 step2 后的 PL 等级设定无效，会按着 PL=0 执行动作。

4.2.2.3 圆弧插补 (MOVC)

机器人通过圆弧插补示教的 3 个点画圆移动。

若用圆弧插补示教机器人轴，移动命令是 MOVC。

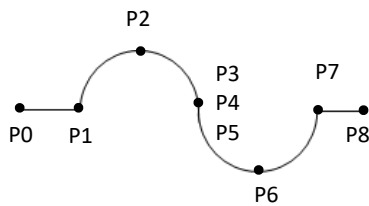
◆ 单一圆弧



点	插补方法	命令
P0	关节 或直线	MOVJ MOVL
P1-P3	圆弧	MOV C
P4	关节 或直线	MOVJ MOVL

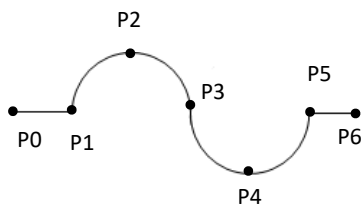
◆ 连续圆弧

如下图所示，当曲率发生变化的圆弧连续有 2 个或以上时，圆弧最终将逐个分离完成，因此，如下图，请在前一个圆弧与后一个圆弧的连接点加入关节或直线插补点。



点	插补方法	命令
P0	关节 或直线	MOVJ MOVL
P1-P3	圆弧	MOV C
P4	关节 或直线	MOVJ MOVL
P5-P7	圆弧	MOV C
P8	关节 或直线	MOVJ MOVL

或者在想要改变曲率的点上加上“CST”附加项，即使是在同一个点上不插入其他关节或直线插补点，也可以使其动作完成。



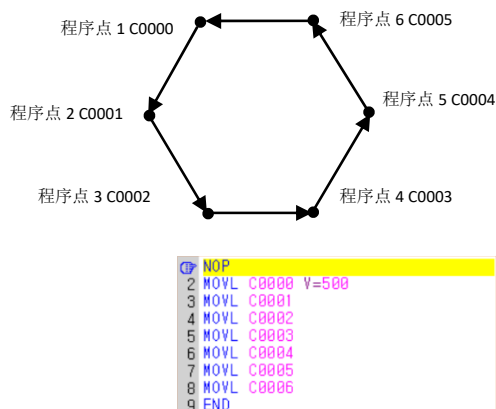
点	插补方法	命令
P0	关节 或直线	MOVJ MOVL
P1-P2	圆弧	MOV C
P3	圆弧	MOV C CST
P4-P5	圆弧	MOV C
P6	关节 或直线	MOVJ MOVL

4.2.3 如何示教程序点

位置数据的设定。

4.2.3.1 位置常量设定

完成如图所示一个边长 200mm 的六边形，需要示教程序点 6 个，最初程序点与最后程序点重合可视为一个。



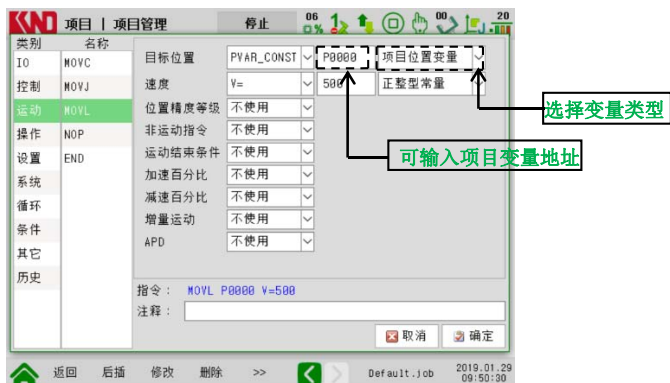
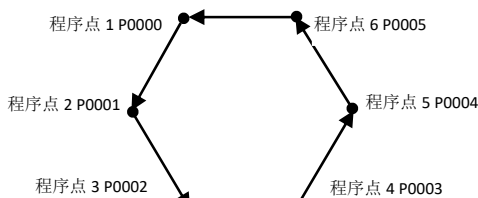
- ①进入到【示教】页面。
- ②握住安全开关：伺服电源接通。
- ③按  切换想要使用的坐标系，按【高】或【低】键选择手动示教速度。
- ④按轴操作键，移动机器人本体。
- ⑤移动光标位置通过【前插】、【后插】按钮选择合适的插入位置。
- ⑥弹出如下图指令编辑，选择位置常量类型，当前机器人的数据位置被写入，选择完成。
- ⑦依次示教完成六个点位置。



但是通过轴操作移动到目标位置示教记忆位置，得到六边形精度跟示教位置有关，只能近似得到一个图形，如果想要更精准的轮廓，可通过专业的机器人 cam 软件自动生成或者宏程序计算得出。

4.2.3.2 位置变量设定

同样是六边形示教，通过位置变量可以设定比较精确的六边形。



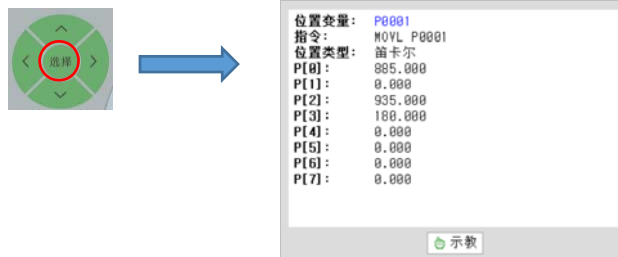
当程序编写如下时，位置变量里的点位数据都是 0.000，需要进入位置变量页面，将通过一键  示教的方法将当前点记忆到变量中或者手动输入都是可以的，步骤如下：

1. 光标选中目标变量

```

1 NOP
2 MOVL P0000 Y=500
3 MOVL P0001
4 MOVL P0002
5 MOVL P0003
6 MOVL P0004
7 MOVL P0005
8 MOVL P0006
9 END
    
```

2. 选择按键中的“选择”键，弹出快捷串口如下：



3. 点击“选择”按钮，便进入到位置变量页面，且第一个显示的变量便是我们目标变量，已经帮我们快速定位。



4. 首先在“位置类型”中选择数据类型。类型的选择跟插补方式相关，插补方式：**MOVL/MOVC**，那类型必须选择“笛卡尔”或者“笛卡尔相对”；插补方式：**MOVJ**，那类型必须选择“关节”或者“关节相对”。
5. 选对类型后，将机器人移动到目标位置，选择按钮 ，当前位置写入成功。位置变量点位数据还可以手动写入。

填写如下表格位置变量数据同样显示六边形轨迹。

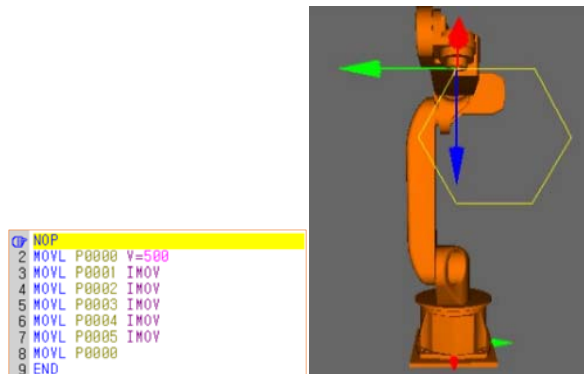
变量		P0000	P0001	P0002	P0003	P0004	P0005
类型		笛卡尔	笛卡尔	笛卡尔	笛卡尔	笛卡尔	笛卡尔
数据	X	885.000mm	885.000mm	885.000mm	885.000mm	885.000mm	885.000mm
	Y	0.000mm	-100.000mm	0.000mm	200.000mm	300.000mm	200.000mm
	Z	1090.000mm	916.800mm	743.600mm	743.600mm	916.800mm	1090.000mm
	A	180.000°	180.000°	180.000°	180.000°	180.000°	180.000°
	B	0.000°	0.000°	0.000°	0.000°	0.000°	0.000°
	C	0.000°	0.000°	0.000°	0.000°	0.000°	0.000°
	J1						
	J2						

显然上述方法虽然适用但略显麻烦，每一个位置点的数据都需要填写，如果知道点与点之间相对坐标，可简便计算如下表的数据，相对距离运动只相对于紧邻的上一位置点，正确选对类型“笛卡尔相对”。

变量		P0000	P0001	P0002	P0003	P0004	P0005
类型		笛卡尔	笛卡尔相对	笛卡尔相对	笛卡尔相对	笛卡尔相对	笛卡尔相对
数据	X	885.000mm	0.000mm	0.000mm	0.000mm	0.000mm	0.000mm
	Y	0.000mm	-100.000mm	100.000mm	200.000mm	100.000mm	-100.000mm
	Z	1090.000mm	-173.200mm	-173.200mm	0.00mm	173.200mm	173.200mm
	A	180.000°	0.000°	0.000°	0.000°	0.000°	0.000°
	B	0.000°	0.000°	0.000°	0.000°	0.000°	0.000°
	C	0.000°	0.000°	0.000°	0.000°	0.000°	0.000°
	J1						
	J2						

相对运动，有自己的编程约束，位置变量 P 的必须是 CARTE_DIST 类型，MOVL 插补指令必须配有 IMOV 项。

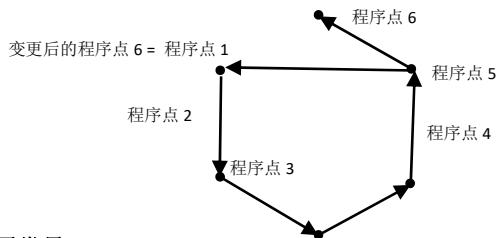
程序编辑如下。



4.2.4 最初程序点与最后程序点的重合方法

为什么要使最初的程序点与最后的程序点重合？

比如：当连续运行如下图所示的程序动作时，发生从最后的程序点 6 向程序点 1 移动。若程序点 6 和程序点 1 在相同位置重合，可提高作业效率。

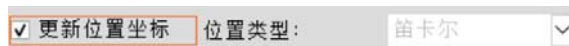


位置常量 C:

最初程序点与最后程序点的重合的操作方法如下：

- ①将程序点 1 作为运动开始行，光标选中该行，选择右上角【设置启动行】按钮，即可。
- ②按【FWD】机器人向最初的程序点位置移动。
- ③到达后，将程序光标向最后程序点行移动。

④选择【修改】，弹出修改对话框，勾选更新位置坐标选项。



⑤选择【完成】

最后程序点记录的是最初程序点的位置数据。

此时，最后程序点修改的只是位置数据而已，插补方法与再现速度没有被修改。

此方法更适用位置常量类型，对于位置变量只需使用相同位置变量即可。

项目位置变量 P: 重复上面的①②③步骤，将机器人移动到目标位置后。进入位置变量页面，选择按钮 ，当前位置写入成功。

更便捷方法: 通过“复制”“粘贴”操作，将起点点位程序复制并粘贴到终点，再删除原目标终点即可。即程序开头结尾使用同一个位置常量或者项目位置变量。但是要搞清楚一点，项目内的项目位置变量 P 是共用的，但是位置常量 C 在同一项目不同程序存在形式是可以不一样的，所以在同项目不同程序间复制粘贴位置常量 C 时要谨慎操作。

4.3 确认程序点

4.3.1 前进/后退的操作

示教后的程序目标位置是否合适，可用示教器的【FWD】与【BWD】进行确认。连续按下【FWD】与【BWD】键时，机器人可一步一步动作。

【FWD】: 只在按住该键期间，机器人按照程序点轨迹运行。

【BWD】: 只在按住该键期间，机器人按照程序点轨迹逆向运动。

重要:

为了确保安全，请先确认试运行倍率并建议调到较慢速度进行。

4.3.2 前进/后退中需要注意的事项:

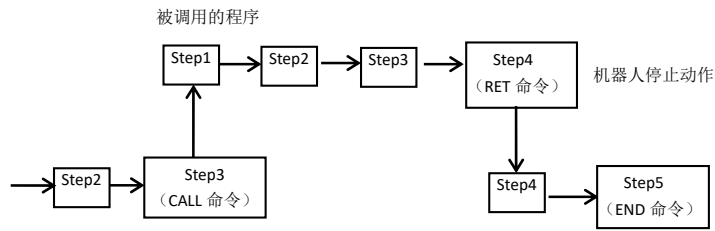
◆ 前进动作

只在按住该键期间，机器人按照程序点轨迹运行，包括非运动命令。

运动类型分为单步、单行两种。

动作一个循环后结束。

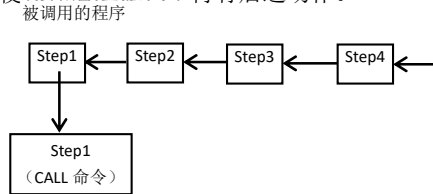
到达结束命令时，即使再按【FWD】键，机器人也不会动作。但是如果是正在调用的程序，机器人向调用程序的下一个命令移动。



◆ 后退动作

只在按住该键期间，机器人按照程序点轨迹逆向运动。只执行移动命令。

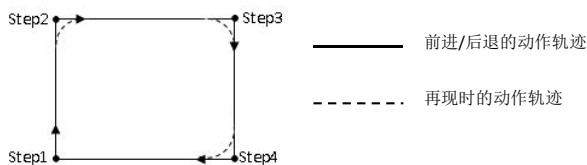
到达第一个程序点 Step1 后，即使再按【BWD】键，机器人也不会动作。但是如果是调用中的程序，机器人向调用命令前的移动命令返回，如果 Step 之前有非运动指令，即使按住后退键也不再有后退动作。



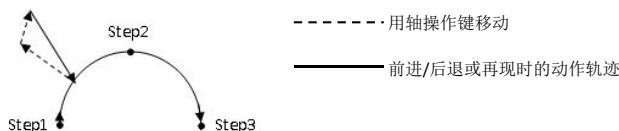
◆ 前进/后退的位置等级动作

1. 如果程序没有编辑 PL 等级，前进/后退的操作会按照示教的位置依次走到 step1、step2、step3、step4 的位置。
2. 示教位置不变，在 step2 这一步添加“位置等级 PL=”，前进/后退的操作的轨迹，由于单步执行没有预读的关系，位置等级不执行，按照如下图实线轨迹运行。
3. 在位置等级前进/后退操作中途中，想要切换到“试运行”或“再现”，下一个程序点位置等级无效。（比如：在 step2 的位置停止前进/后退操作，切换“试运行”或“再现”，step3 这一步轨迹按实线运动，等级恢复从 step4 开始。）反过来，在位置等级进行“试运行”或“再现”中途，切换到前进/后退操作，下一个程序点位置等级还未失效（比如：在 step2 的位置停止“试运行”或“再现”，切换前

进/后退操作 step3 这一步轨迹按虚线运动，等级失效从 step4 开始。)



- ◆ 前进/后退，（圆弧插补）偏离轨迹后的动作
1. 向圆弧插补最初的程序点的移动为直线运动。
 2. 圆弧插补的程序点如果不是连续的 3 个点，则不能进行圆弧动作。
 3. 如图所示，中途停止前进/后退，进行轴操作后偏离程序轨迹，重新进行前进/后退操作的话，机器人先要回到偏离轨迹的起始点，然后继续圆弧动作。（与再现时偏离轨迹后动作相同），返回偏离起始位置的插补轨迹与速度，可由参数控制。



参考：

偏移轨迹后再返回偏离起始点的动作行为称为“低速启动”。

4.3.3 速度选择

试运行倍率修改，【SHIFT】+【高】或【低】组合键切换进入、退出试运行倍率修改状态，【高】、【低】键设定速度倍率，设定范围在 10%-150%。

也可在【设置】的【其他】界面中修改速度倍率。



4.3.4 试运行

所谓试运行，是在示教模式不变的情况下模拟再现动作的功能。该功能在连续轨迹、各种命令的动作确认时使用，非常方便。只有按住该键期间，可对程序示教过的点作为连续轨迹进行确认。

1. 将该程序设置为主程序，选择【编辑】。
2. 按下安全开关，“伺服接通”灯常亮。



3. 按试运行【JOG】。
 - 机器人开始按照设定的程序运行。
 - 机器人只有在试运行按键按下期间动作。
 - 如离开试运行【JOG】键，机器人立刻停止动作。

4.4 修改程序点

4.4.1 命令的修改

4.4.1.1 移动命令位置修改

1. 在【示教】页面显示程序内容，将程序点光标向需要修改位置数据的移动命令行移动。

如图：修改行号 5 的位置，黄色光标已选中该行。

```
5 MOVL C0002 V=100 PL=8
6 MOVL C0003 V=100
7 MOVL C0004 V=80
```

2. 接通伺服电源，按轴操作键，将机器人向修改后的位置移动。
3. 选择【修改】操作。

返回 后插 修改 删除 >>

4. 弹出指令编辑窗口，勾选更新位置坐标，不要做其他修改。

☒ 更新位置坐标 位置类型: 笛卡尔

5. 选择【完成】

参考：

此操作只会修改位置数据，不会修改附加项内容。
方法适用于修改位置常量类型。
若移动命令设定的是位置变量，位置变量值无法修改。

4.4.1.2 移动命令插补方式修改

1. 在【手动示教】页面显示程序内容，将程序点光标向需要修改的移动命令行移动。

如图：修改行号 6 的位置，黄色光标已选中该行。

```
5 MOVL C0002 V=100 PL=8
6 MOVL C0003 V=100
7 MOVL C0004 V=80
```

2. 选择【修改】操作



3. 选择想要的插补方式，可从“历史指令”或“运动指令”选择 MOVJ 关节指令，配上适合的 VJ 速度，选择“完成”，放弃修改请选择“取消”。

类别	名称	更新位置坐标	位置类型
IO	MOV		
控制	MOVJ	☒	位置常量
运动	MOVL		
操作	NOP		
设置	END		
系统			
循环			
条件			
其它			
历史			

目标位置	PVAR_CONST	C0003	位置常量
关节速度	VJ=	100	正整型常量
位置精度等级	不使用		
非运动指令	不使用		
运动结束条件	不使用		
加速度(%)	不使用		
减速度(%)	不使用		
增量运动	不使用		
APD	不使用		

指令: MOVJ C0003 VJ=100

注释:

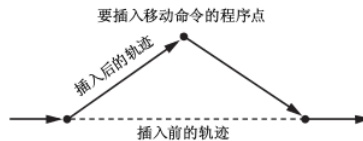
注意:

修改插补方式，一定要选中 ☒ 更新位置坐标。MOVL 与 MOVJ 运动的位置坐标类型是不一样的，也不可混用的。

4. 选择【完成】，插补方式被修改成功。

```
5 MOVL C0002 V=100 PL=8
6 MOVJ C0003 VJ=100
7 MOVL C0004 V=80
```

4.4.2 命令的插入



1. 将程序点光标移动到插入移动命令的前一行。

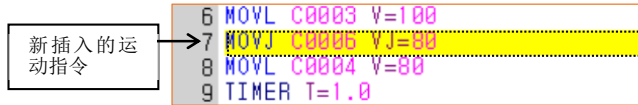
要插入的移动命令的前一行

```
6 MOVL C0003 V=100
7 MOVL C0004 V=80
8 TIMER T=1.0
```

2. 接通伺服电源，按轴操作键，让机器人向插入位置移动。
3. 按【后插】操作，如果想要插入光标指向的上一行，选择【前插】按钮。

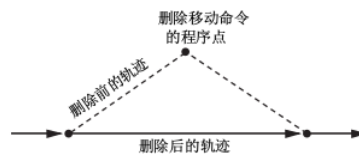
返回 后插 修改 删除 >>

4. 选择合适的插补方式与再现速度，选择【完成】。

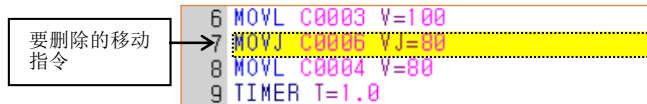


4.4.3 命令的删除

删除一行程序的操作如下：



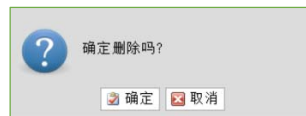
1. 光标选中要删除的程序行。



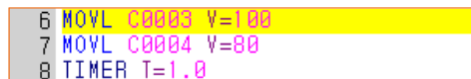
2. 选择【删除】操作。

返回 后插 修改 删除 >>

3. 弹出删除提示对话框，【确认】按钮确认删除，【取消】则取消删除。



4. 原光标行号为 7 的移动指令被删除，行号自动更新排列。



删除多行程序的操作如下：

1. 光标选中删除程序段的起始行：

```

1 NOP
2 MOVJ C0000 VJ=80 //定位点1
3 MOVL C0005 V=1500
4 MOVL C0001 V=1500
5 MOVL C0002 V=100 PL=8
6 MOVL C0003 V=100
7 MOVL C0004 V=80
    
```

2. 按下示教器上的 **SHIFT** 按键，同时按下方向键中的 ▽ 向下选择程序行。



3. 光标一直移动至删除程序段的最后行时，松开 **SHIFT** 按键和方向键。黄色区域为选中的删除的程序段。

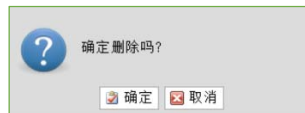
```

1 NOP
2 MOVJ C0000 VJ=80 //定位点1
3 MOVL C0005 V=1500
4 MOVL C0001 V=1500
5 MOVL C0002 V=100 PL=8
6 MOVL C0003 V=100
7 MOVL C0004 V=80
    
```

4. 选择【删除】操作。

返回 后插 修改 删除 >>

5. 弹出删除提示对话框，Yes 确认删除，No 取消删除。



6. 选中的黄色区域的程序段被删除，行号自动更新排列。

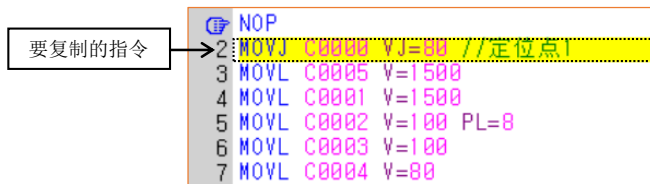
```

1 NOP
2 MOVJ C0000 VJ=80 //定位点1
3 MOVL C0003 V=100
4 MOVL C0004 V=80
    
```

4.4.4 命令的复制和粘贴

复制一行程序的操作如下：

1. 光标选中要复制的程序行。



2. 选择【复制】操作。复制后会看到“粘贴（后插）”、“粘贴（前插）”变为可用状态。



3. 选择要粘贴插入的程序行的位置。选择【粘贴（后插）】时在当前程序之后插入，选择【粘贴（前插）】时在当前程序之前插入，

```
11 WAITIO IN#(X0.5) == ON
12 MOVL C0003 V=100 PL=8
13 MOVL C0002 V=100 PL=8
14 MOVL C0001 V=1500
```

4. 本例中选择在第 14 程序行后粘贴，选择【粘贴（后插）】。

- 前插
- 注释
- 复制
- 剪切
- 粘贴(后插)
- 粘贴(前插)
- 查找
- 跳转到APD
- 辅助IO

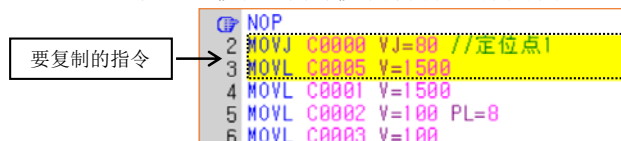
5. 粘贴后的程序如下所示。

```

1 NOP
2 MOVJ C0000 VJ=80 //定位点1
3 MOVL C0005 V=1500
4 MOVL C0001 V=1500
5 MOVL C0002 V=100 PL=8
6 MOVL C0003 V=100
7 MOVL C0004 V=80
8 TIMER T=1.0
9 DOUT OT#(F0.5) ON
10 TIMER T=1.0
11 WAITIO IN#(X0.5) == ON
12 MOVL C0003 V=100 PL=8
13 MOVL C0002 V=100 PL=8
14 MOVL C0001 V=1500
15 MOVJ C0000 VJ=80 //定位点1
16 NOP
17 RET
18 END
    
```

复制多行程序的操作如下：

- 与删除多行时的操作类似，光标选中复制程序段的起始行，按下示教器上的 SHIFT 按键，同时按下方向键选择程序行：



- 按下示教器上下方向键，选择要粘贴插入的程序行的位置。本例选择在第 14 程序行后粘贴，点击“粘贴（后插）”按钮。

```

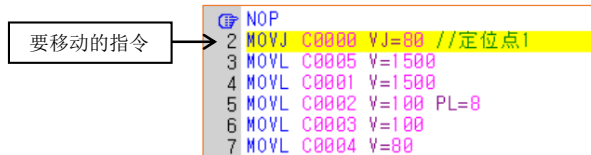
11 WAITIO IN#(X0.5) == ON
12 MOVL C0003 V=100 PL=8
13 MOVL C0002 V=100 PL=8
14 MOVL C0001 V=1500
    
```

3. 粘贴后的程序如下所示。

```
10 TIMER T=1.0
11 WAITIO IN#(X0.5) == ON
12 MOVL C0003 V=100 PL=8
13 MOVL C0002 V=100 PL=8
14 MOVL C0001 V=1500
15 MOVJ C0000 VJ=80 //定位点1
16 MOVL C0005 V=1500
```

4.4.5 命令的移动与粘贴

1. 光标选中要移动的程序行。



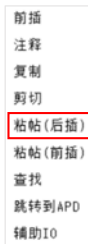
2. 选择【剪切】操作。源位置文件被移除，剪切后会看到“粘贴（后插）”、“粘贴（前插）”变为可用状态。



3. 选择要粘贴插入的程序行的位置。选择【粘贴（后插）】时在当前程序之后插入，选择【粘贴（前插）】时在当前程序之前插入，

```
11 WAITIO IN#(X0.5) == ON
12 MOVL C0003 V=100 PL=8
13 MOVL C0002 V=100 PL=8
14 MOVL C0001 V=1500
```

4. 本例中选择在第 14 程序行后粘贴，选择【粘贴（后插）】。



5. 粘贴后的程序如下所示。

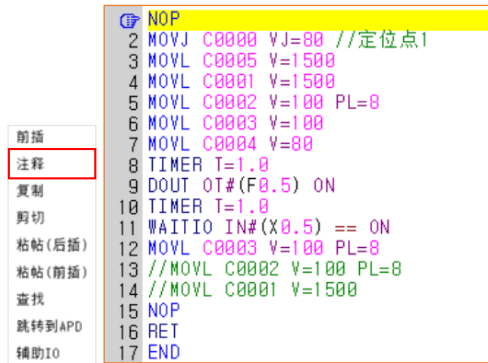
```

1 NOP
2 MOVL C0005 V=1500
3 MOVL C0001 V=1500
4 MOVL C0002 V=100 PL=8
5 MOVL C0003 V=100
6 MOVL C0004 V=80
7 TIMER T=1.0
8 DOUT OT#(F0.5) ON
9 TIMER T=1.0
10 WAITIO IN#(X0.5) == ON
11 MOVL C0003 V=100 PL=8
12 MOVL C0002 V=100 PL=8
13 MOVL C0001 V=1500
14 MOVL C0000 VJ=80 //定位点1
15 NOP
16 RET
17 END
    
```

也可已移动多行指令，过程不在详细介绍，可参看复制粘贴的介绍过程。

4.4.6 命令的注释

如果某一行或者某一段指令不想执行，但又不想删除以备使用，只需要将指令注释即可，程序就不再译码执行，注释后为被视为文字说明，特征是指令前面有双斜杠且颜色都是绿色。如果想让程序生效，则取消注释即可恢复。



4.5 试运行

试运行是指：不改变示教模式，模拟再现动作的功能。
该功能可在连续轨迹的确认，各种命令的动作确认时使用。

试运行操作：

1. 打开程序编辑页面，如图所示：



2.  示教模式，握住安全开关，伺服接通灯亮。
3. 调整合适试运行倍率。

4. 通过【设置启动行】，让启动光标指向程序开头。
5. 按住试运行【JOG】机器人开始按程序连续动作。
出于安全考虑，机器人只有在按键按住期间动作。
离开试运行【JOG】键，机器人立即停止动作。

重要：

在运行机器人前，请务必确认机器人周围安全。

4.6 其它程序的编辑

4.6.1 程序跳转

程序跳转指在程序内部的任意跳转，需要用到控制指令的 LABEL 标号与 JUMP 指定跳转两个指令。

可用在多次重复的循环，也可根据条件的判断来跳转 LABEL 指定的位置，用法雷同数控系统的中宏程序（GOTO）、（IF，GOTO）。

4.6.1.1 无条件跳转

```
0004 PULSE Y0.0 T=1.5;
0005 TIMER=1.5;
0006 L00;
0007 MOVL C0001 V=100;
0008 MOVL C0002 V=100;
0009 MOVL C0003 V=100;
0010 JUMP L00;
```

```
0004 PULSE Y0.0 T=1.5;
0005 TIMER=1.5;
0006 JUMP L01;
0007 MOVL C0001 V=100;
0008 MOVL C0002 V=100;
0009 L01;
0010 MOVL C0003 V=100;
```

左边程序行 0006 即为 LABEL 所指标号位置，执行到 0010 行的 JUMP L00，会自动跳到 LABEL 所指位置，会再次顺序执行程序行 0007 到 0009 的移动命令，重复循环。

右边程序行 0009 即为 LABEL 所指标号位置，执行到 0006 行的 JUMP L00，会自动跳到 LABEL 所指位置，直接跳过程序行 0007 到 0008 的移动指令，不执行该两点位置机器人的移动，直接去执行 0010 行的程序。

如果跳转没有指定相同的 LABEL 标签，跳转被终止，无法找到需要跳转的位置，会提示错误报警。

一个 LABEL 标签号只能指定一个位置，不允许在两处或者两处以上的位置重复使用同一 LABEL，但是可不限次数跳转到该 LABEL 所指位置。

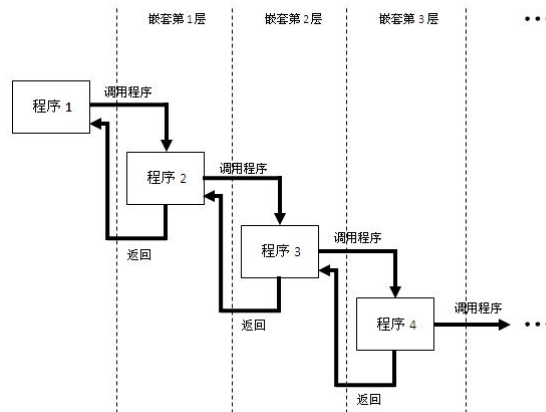
```
0003 D0001=0;
0004 L02;
0005 PULSE Y0.0 T=1.5;
0006 TIMER=1.5;
0007 MOVL C0001 V=100;
0008 MOVL C0002 V=100;
0009 D011++;
0010 JUMP L02 IF (D0001<7);
0011 MOVL C0003 V=500;
```

4.6.1.2 有条件跳转

L02 为 LABEL 标号，JUMP 指令跳转条件是数字变量 D0001 小于 7，变量 D0001 初始值为 0，每循环一次程序行 0005 到 0008，后自动加 1 自保存，遇到 JUMP 判断是否满足 D 小于 7 的条件，可见，要在程序行 0005 到 0008 循环 6 次后，D0001 的值为 7 才会不再满足跳转条件，跳出循环，执行 JUMP 后的指令。

4.6.2 程序调用

程序调用结构框架：



程序调用示例：

KND	Robot
0000 NOP;	0000 NOP;
0002 MOVL C0001 V=100;	0002 PULSE Y0.0 T=1.5;
0002 MOVL C0002 V=100;	0002 TIMER=1.5;
0003 MOVL C0003 V=100;	0003 MOVL C0003 V=100;
0004 CALL "robot";	0004 RET;
0005 MOVL C0004 V=500;	
0006 END;	

程序“KND”是主程序，“robot”是被调用的程序，当主程序运行到“CALL”robot”（）；”自动调用并执行“robot”的程序内容，直到遇到 RET 才会返回主程序，继续执行 CALL 后的指令。

被调用的程序必须有 RET 指令，是返回的必要条件，如果缺失会出现错误报警。

参考：

调用程序最多嵌套 127 层。

4.7 自定义变量

4.7.1 自定义变量简介

自定义变量可以使用户对数值、字符、位置进行自定义，在编写程序时可直接使用自定义名称来代替对应的变量，使程序显示直观，使用方便。

自定义变量的使用范围为 job 内使用，在一个 job 中设定的自定义变量无法被其他 job 使用。

4.7.2 自定义变量使用方法

选中需要设定自定义变量的程序，点击画面下方的“自定义变量”按钮，如下所示。



进入自定义变量界面，如下所示，可自定义数值、字符、位置变量。



首先建立自定义数值变量，类型为 32 位有符号数。在“数值”栏的“U32”一行点击，如下所示。

点击画面下方的“新建”，即可新建数值变量。



点击自定义变量名称栏，可修改变量名称。名称使用英文字母和数字的格式，字母区分大小写。

点击数组长度栏，可设定数组长度。

点击注释栏，可对该变量添加注释。

点击初始化值栏，可设定该数值变量的值。

新建的自定义数值变量如下所示。



建立位置变量，如下所示。点击“位置”一行，然后点击画面下方“新建”，

即可新建自定义位置变量。

位置变量选择位置类型：关节、关节相对、笛卡尔、笛卡尔相对。可直接在坐标栏中输入坐标值，也可点击画面下方的“示教”按钮进行示教。新建的自定义位置变量如下。



自定义变量建立完毕后，点击“保存”按钮进行保存，然后点击“返回”按钮返回到项目界面。点击“示教编辑”进入到程序编辑界面，如下。



在编写指令时即可直接使用 loc1、loc2、loc3 作为位置，将 Speed1、Speed2、Speed3 的作为速度。点击“后插”，插入指令对话框，选择 MOVJ 指令，目标位置

选择“位置变量表达式”，输入 pos1，关节速度选择“数字变量表达式”，输入 num1。如下所示。



输入完毕后点击“完成”按钮，指令输入结果如下所示。



按照同样方法输入其他指令，如下所示。编写完毕后，试运行该程序，该程序能够正常执行。



5 再现运行程序

5.1 再现前准备

所谓再现就是执行示教后的程序。
作为示教前的准备，首先登录再现程序。

5.1.1 再现模式下的操作限制

不允许编辑程序、不允许切换程序、不允许切换坐标系、不允许修改参数、示教坐标系等。可以说再现模式下只允许程序的再现动作，在示教模式下的操作受到限制，页面状态为灰色，未激活不允许相关操作。

进入：

1. 选择主菜单 .
2. 选择【项目】里的项目管理。显示如下项目管理页面。



3. 加载所需的项目，并点击“设置主程序”按钮，将需要再现的程序设定为主程序。
4. 点击“编辑”进入程序编辑页面。



5.2 再现

重要:

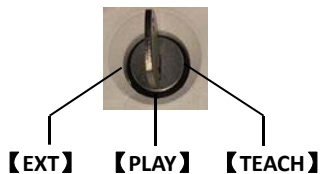
在运行机器人前，请务必确认机器人周围安全。

5.2.1 再现过程

所谓再现，就是再现示教程序的动作。
程序的再现、启动可通过以下装置进行。

1. 启动装置选择

将示教编程器的模式键扭到【PLAY】。此时示教编程器设定为再现模式。
查看状态区的模式的变化。



2. 接通伺服电源

按示教编程器上的【伺服准备】键，伺服接通。示教编程器上的【伺服接通】灯亮。

3. 通过按下“高”、“低”键，观察状态条，调整合适的再现倍率。
4. 操作【开始】

按示教编程器的【START】按钮，指示灯亮，机器人开始动作。

5.2.2 再现低速启动

低速启动：机器人在偏离程序轨迹后，想要从暂停的光标位置继续启动，这时机器人只在启动的最初一个点低速启动。

低速启动适用在【再现】模式，以及【示教】模式时确认示教轨迹的“FWD”或“BWD”操作。

在运行时因按下急停键由于急停惯性或暂停时进行过轴操作都算是偏离了轨迹。

任何情况，只要是【复位】操作，程序光标返回到程序的起点同时，低速启动同样被解除。

机器人正常完成程序点动作时，低速启动都是出于失效状态。

当完成一个程序点的低速运行，以下的程序点均按再现速度/示教速度运动。

重要：

低速启动动作，是把从已经偏离的点回到程序被偏离起始位置的动作作为低速启动的条件。

低速启动有两种运动插补方式，可有参数 P00955 来选择。

低速启动速度选择参数：P00956、P00957。

5.3 停止与再启动

使运行中的程序或机器人停止运行或自动停止运行有以下几种情况：

- 暂停操作
- 急停操作
- 报警引起的停止
- 其它停止

5.3.1 暂停操作

使用示教编程器

- 暂停
 1. 按示教编程器的【暂停】键。
 2. 机器人暂停、【暂停】指示灯在按下期间亮。

解除：要使机器人重新动作，使示教编程器的【暂停】处于“OFF”状态，再次按下【开始】按键，检测到信号有“ON”的状态，机器人从暂停状态重新开始运动。

5.3.2 急停操作

实施急停后，伺服电源关闭，机器人立即停止动作。急停可通过以下方式操作：

- 控制柜面板。
- 示教编程器。
- 外部输入信号（专用）。
- 解除：
 1. 将急停按钮向箭头方向旋转。
 2. 再次接通伺服电源时，按示教编程器的【伺服准备】键，并握住示教器上的安全开关。



5.3.2.1 急停后再启动



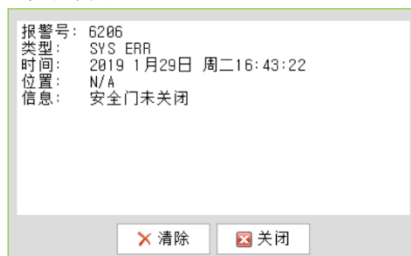
- 急停后重新启动前，请用“前进”的操作确认机器人位置，确认与工件、夹具等没有干涉。
- 连续程序点在高速再现过程中被急停后，机器人有时会在显示程序点之间的附近停止，此时位置偏离轨迹，启动时满足低速启动的条件，按照设定的插补方式、速度返回程序被偏离的起始位置，如此时重新启动，可能与工件、夹具干涉。

5.3.3 报警引起的停止

若发生报警，机器人停止动作，但一些低级别报警，会有提示但不会切断伺服能停止作业中机器人的运动，并且在状态条显示区有闪烁的报警提示符。



点击提示符会弹出报警内容。



解除：

- 轻故障报警
只需要按复位即可清除报警，报警符消失。
- 重故障报警
需要断开主电源，排除故障原因。
若发生重大的硬件故障时，会自行切断伺服电源，停止机器人动作。需要彻底排除问题，才可复位。

打开报警页面，可查看机器人“当前报警”与“历史报警”两项内容。

5.3.4 其它停止

模式切换引起的暂停：

若再现模式中，从再现模式切换到示教模式，程序立刻停止。

重新开始运动时，回到再现模式，【开始】键继续开始动作。

执行 PAUSE 命令引起的暂停：

若执行 PAUSE 命令，程序暂停。

重新开始运动时，请执行开始操作。机器人将从下一个命令继续运动。

5.4 再现速度的修改

速度调节

【PLAY】模式下，按【高】与【低】修改再现速度的比率，比率设定范围为 0%~150%。

5.5 离线模式

所谓离线模式是指与机械手本体脱开操作的模式。运动操作只体现在界面的仿真模型，而不改变机器手本体动作。



设置方法，进入设置/其他界面，选中离线。
还可以选择离线模式下对 DO+AO 输出的影响。

6 应用举例

通用用途示教举例如下。

6.1 示教前准备

开始示教前，应做好以下准备：

1. 确认示教编程器上的模式旋钮对准“TEACH”，设定为示教模式，
2. 同时查看状态模式图标为示教图标 。
3. 保证急停开关  处于旋开状态否则即使按住安全开关，伺服电源也不会接通。
4. 按下主菜单  图标，选择【项目】 里的【项目管理】，进入项目管理界面如图：



5. 在项目列表中，选择其中一个项目单击，然后选择“新程序”，弹出全键盘，输入程序名，字母排布与使用方法与电脑键盘基本一致。现以“KND”为程序名举例说明如下：首先要选择 Cap “字母大写锁定状态”，再选择“K”、“N”、“D”，加“.job” 文件属性输入程序名。



6. 按【确定】键，名为“KND”的程序创建完成。
7. 将 KND 设置为主程序，选择“编辑”进入程序编辑页面，页面显示该程序，“NOP”和“END”命令会自动生成。



6.2 示教基本步骤

6.2.1 示教基本步骤

为了使机器人能够进行再现，就必须把机器人的运动命令编成程序。控制机器人运动的命令就是移动命令。在移动命令中，记录有移动目标位置、插补方式、再现速度等。

命令使用示例如图所示：

MOVJ C0000 VJ=50 —— 机器人采用关节插补运动方式以 50%基准速度运动到指定点。

MOVL C0006 V=800——机器人采用直线运动方式以 800mm/s 运动到指定点。

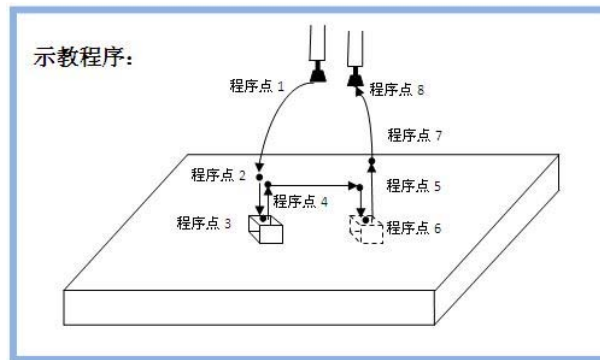
当再现如右图所示程序内容时，机器人按照程序点 1 的移动命令中输入的插补方式和再现速度移动到程序点 1 的位置。然后，在程序点 1 和 2 之间，按照程序点 2 的移动命令中输入的插补方式和再现速度移动。同样，在程序点 2 和 3 中，按照程序点 3 的移动命令中输入的插补方式和再现速度移动。当机器人到达程序点 3 的位置后，依次执行 TIMER 命令和 DOUT 命令，然后移动程序点 4 的位置，如果没有编辑速度，默认按照之前的速度执行。



6.2.2 示教程序开始

所谓程序就是把机器人的作业内容用机器人的语言进行描述的作业程序。

现在我们尝试实际示教，如下图所示，假设是一次搬运过程，将工件从 A 搬运到 B，（暂不考虑机械手抓取动作）。按照箭头的程序点轨迹，对机器人进行移动程序的示教。

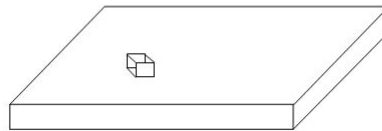


● **程序点 1——待机开始位置示教**

1. 握住安全开关，接通伺服电源，使机器人进入可运行状态。



2. 用轴操作键将机器人移到待机开始位置。
3. 请将待机位置选择在适合作业准备的地方。
4. 选择“后插”操作。



返回 后插 修改 删除 >>

5. 弹出指令编辑对话框，选择运动指令，插补方式。选择关节插补，命令 MOVJ，设定关节速度 VJ=50.0%，以关节运动方式 50%基准速度运动到目标点)，预览处显示 MOVJ C0000 VJ=50.0。
6. 选择“完成”，输入程序点 1。

NOP
2 MOVJ C0000 VJ=50.0 //过渡点!

● **程序点 2——快速靠近工件**

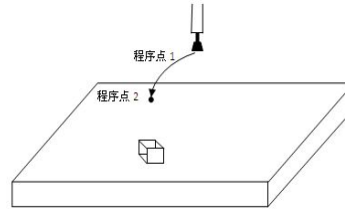
1. 用轴操作键，使机器人姿态为作业姿态，移到示教点 2 的位置。

2. 选择“后插”操作。

返回 后插 修改 删除 >>

3. 弹出指令编辑对话框，选择运动指令，插补方式。
选择关节插补，命令为 MOVJ，设定关节速度 VJ=50.0%，(以关节运动方式 50%基准速度运动到目标点)，预览处显示 MOVJ C0001 VJ=50.0。
4. 选择“完成”，输入程序点 2。

```
NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
```



● 程序点 3——抓取工件的位置

1. 按手动速度【高】或【低】键，直到在状态区域显示为中速。

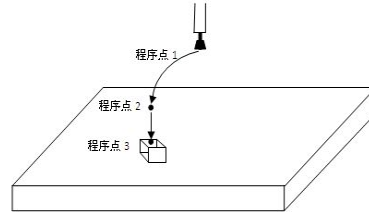


2. 保持程序点 2 的姿态不变，按【坐标】键，设定机器人坐标为世界坐标系。
3. 用轴操作键将机器人移到抓取位置。
4. 选择“后插”操作。

返回 后插 修改 删除 >>

5. 弹出指令编辑对话框，选择运动指令，插补方式。
选择直线插补，命令 MOVJ，设定速度 V=30.0 (30mm/s)，预览处显示 MOVJ C0002 V=30.0。
6. 按【完成】，输入程序点 3

```
NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVJ C0002 V=30.0 //抓取点
```



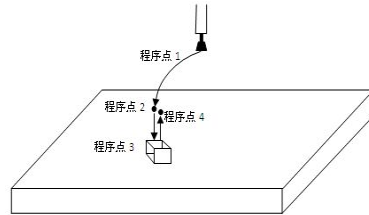
● 程序点 4——抓取工件抬起安全高度

1. 保持程序点 2 的姿态不变，用轴操作键将机器人移到一定高度。

2. 选择“后插”操作。

返回 后插 修改 删除 >>

3. 弹出指令编辑对话框，选择运动指令，插补方式。
选择直线插补，命令 MOVJ，设定速度 V=100.0
(100mm/s)，预览处显示 MOVJ C0003 V=100.0。
4. 按【完成】，输入程序点 4。



```

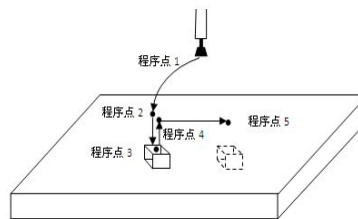
1 NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVJ C0002 V=30.0 //抓取点
5 MOVJ C0003 V=100 //离开目标工件

```

● 程序点 5——搬运目标位置正上方

1. 保持程序点 2 的姿态不变，用轴操作键将机器人移到目标位置上方。
2. 选择“插入”操作。

返回 后插 修改 删除 >>



3. 弹出指令编辑对话框，选择运动指令，插补方式。
选择直线插补，命令 MOVJ，设定速度 V=160.0 (160mm/s)，预览 MOVJ C0004 V=160.0。
4. 按【完成】，输入程序点 5

```

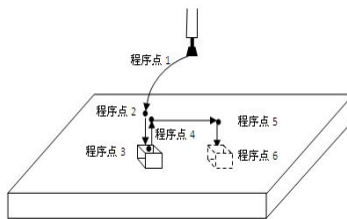
1 NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVJ C0002 V=30.0 //抓取点
5 MOVJ C0003 V=100 //离开目标工件
6 MOVJ C0004 V=160 //目标位置过渡点

```

● 程序点 6——放置目标位置

1. 保持姿态不变，用轴操作键将机器人移到目标位置。
2. 选择“后插”操作。

返回 后插 修改 删除 >>



3. 弹出指令编辑对话框，选择运动指令，插补方式选择直线插补，命令 **MOVL**，设定速度 **V=30.0** (30mm/s)，预览 **MOVL C0005 V=30.0**。
4. 按【完成】，输入程序点 6。

```

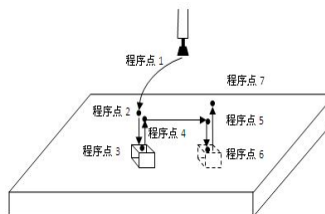
1 NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVL C0002 V=30.0 //抓取点
5 MOVL C0003 V=100 //离开目标工件
6 MOVL C0004 V=160 //目标位置过渡点
7 MOVL C0005 V=30.0 //慢慢靠近目标位置

```

● 程序点 7——作业完成快速抬起

1. 按手动速度【高】，设定为高速。
2. 选择“后插”操作。

返回 后插 修改 删除 >>



3. 弹出指令编辑对话框，选择运动指令。插补方式选择 **MOVL**，设定速度 **V=800.0** (800mm/s)，预览 **MOVL C0006 V=800.0**。
4. 按【完成】，输入程序点 7。

```

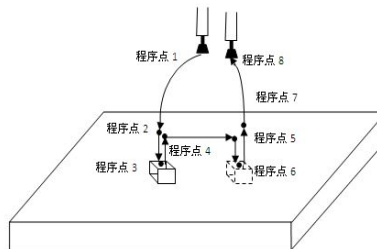
1 NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVL C0002 V=30.0 //抓取点
5 MOVL C0003 V=100 //离开目标工件
6 MOVL C0004 V=160 //目标位置过渡点
7 MOVL C0005 V=30.0 //慢慢靠近目标位置
8 MOVL C0006 V=800 //完成取放作业快速抬起

```

● 程序点 8——回到待机位置附近

1. 用轴操作键把机器人移动到开始位置附近。
2. 选择“后插”操作。

返回 后插 修改 删除 >>



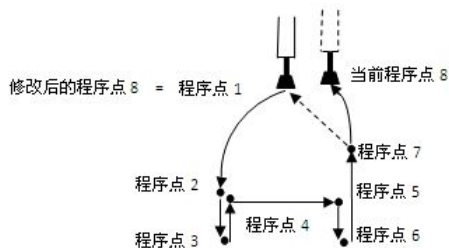
3. 弹出指令编辑对话框，选择运动指令，插补方式。
选择关节插补，命令 **MOVJ**，设定关节速度 **VJ=50.0%**，
以关节运动方式 **50%基准速度运动到目标点**）预览处显示 **MOVJ C0007 VJ=50.0**。
4. 按【完成】，输入程序点 8。

```

1 NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVL C0002 V=30.0 //抓取点
5 MOVL C0003 V=1.00 //离开目标工件
6 MOVL C0004 V=1.60 //目标位置过渡点
7 MOVL C0005 V=30.0 //慢慢靠近目标位置
8 MOVL C0006 V=800 //完成取放作业快速抬起
9 MOVJ C0007 VJ=80 //快速回到待机位置等待命令
    
```

● 最初程序点和最后程序点重合

现在，机器人停在程序点 1 附近的程序点 8 的位置，如果作业结束位置的程序点 7 直接到程序点 1，即可开始后面的作业，故可提高作业效率。下面，我们试着把最终位置的程序点 8 与最初的程序点 1 设在同一个位置上。



1. 将光标移到程序点 1。

```

1 NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVL C0002 V=30.0 //抓取点
5 MOVL C0003 V=100 //离开目标工件
6 MOVL C0004 V=160 //目标位置过渡点
7 MOVL C0005 V=30.0 //慢慢靠近目标位置
8 MOVL C0006 V=800 //完成取放作业快速抬起
9 MOVJ C0007 VJ=80 //快速回到待机位置等待命令
    
```

2. 由于程序点 1 是程序的第一行，按【FWD】键，机器人移动到程序点 1，或者，可以通过“设置启动行”操作设置该行为启动开始行，然后按【FWD】键，一样可以到达程序点 1。
3. 将光标移动到程序点 8。

```

1 NOP
2 MOVJ C0000 VJ=50.0 //过渡点1
3 MOVJ C0001 VJ=50 //过渡点2
4 MOVL C0002 V=30.0 //抓取点
5 MOVL C0003 V=100 //离开目标工件
6 MOVL C0004 V=160 //目标位置过渡点
7 MOVL C0005 V=30.0 //慢慢靠近目标位置
8 MOVL C0006 V=800 //完成取放作业快速抬起
9 MOVJ C0007 VJ=80 //快速回到待机位置等待命令
    
```

4. 按【修改】。

返回 后插 修改 删除 >>

5. 选中“☐ 更新位置坐标”。

☒ 更新位置坐标 位置类型: 关节

6. 按【确定】。

6.3 轨迹的确认

在完成了机器人的动作程序输入后，运行一下该程序，以便检查一下各程序点位置是否合理。

1. 把光标移到程序点 1（行 0001）。
2. 在【设置】/【运行模式】界面，调整试运行倍率。
3. 握住安全开关，伺服接通灯亮。
4. 按【FWD】键，通过机器人动作确认各程序点。每按一次【FWD】，机器人移动一个程序点。

5. 程序点确认完成后，将光标移到程序起始处。
6. 最后我们来校验一下所有程序点的连续动作。按试运行【JOG】键，机器人连续再现所有的点，一个循环后停止运行。

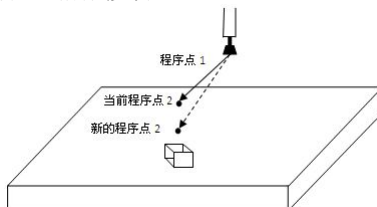


机器人在试运行【JOG】键下连贯动作，就是一个程序的完整再现，是否与你预想的一样。如果不太满意，那么请尝试修改程序中程序点的**位置与速度**。

6.4 修改程序

6.4.1 修改程序点的位置数据

试着对程序点 2 的位置稍作修改。



1. 把光标移到待修改程序点 2 的位置。
2. 用轴操作键，把机器人移到修改后的位置。
3. 选择【修改】。

返回 后插 修改 删除 >>

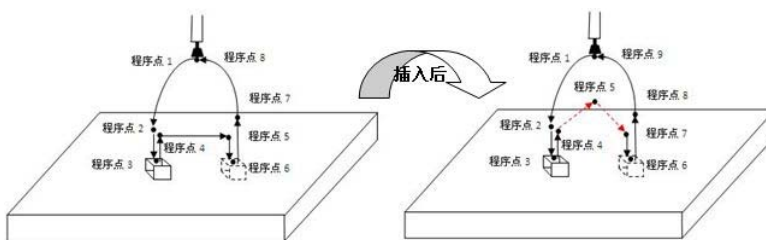
4. 弹出光标所在行的程序内容选中“☐ 更新位置坐标”。

☒ 更新位置坐标 位置类型:

5. 按“确定”，程序点 2 的位置数据被修改。

6.4.2 插入程序点

预防周围的干涉影响需要在程序点 4、5 之间，插入一个新程序点。



1. 把光标移到待修改程序点 4 的位置。
2. 用轴操作键，把机器人移到插入点的位置。
3. 选择【后插】操作。

返回 后插 修改 删除 >>

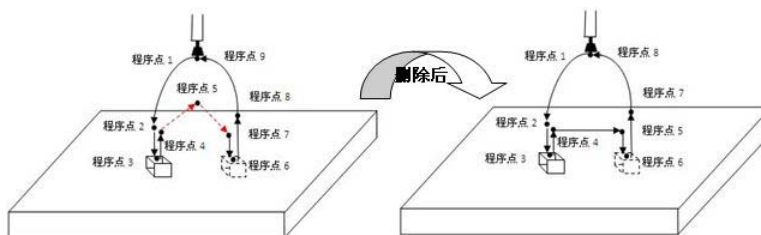
4. 弹出指令编辑对话框，选择合适插补方式、速度。
5. 按“确定”，新的程序点位置被插入程序点 4、5 之间。

重要：

前插、后插操作都是插入新指令，以光标行所在位置前后插入。

6.4.3 删除程序点

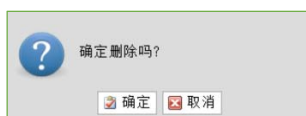
这次我们试着删除刚刚插入过的程序点。
从下面左图的状态返回到原来右图的状态。



1. 首先确认想要删除程序的位置，【FWD】键操作，把机器人移到待删除的程序点 5，看该位置是否是想要删除的位置。
2. 确认光标位于待删除的程序点上，按【删除】。

返回 后插 修改 删除 >>

3. 弹出删除确认对话框，确认删除选择“确定”。



4. 该程序点成功删除，后续的程序点序号自动减 1。

? 实际删除程序点，光标选中想要删除的程序行，按“删除”就可以直接删除，但是如果方便建议通过【FWD】键操作，将机器人移到该程序点位置确认后再删除，避免出现误删错操作。

6.5 程序再现

6.5.1 再现前准备

重要：

做好程序再现前准备，为了从程序头开始运行，请务必先进行以下操作。

- ① 把启动标志符移到程序开头。
- ② 用轴操作键将机器人移到安全位置。

6.5.2 再现

准备就绪后，开始我们程序的再现。操作前，请确认机器人周围没有人。

1. 把示教编程器上的旋钮设置在【PLAY】模式上，成为再现模式 .
2. 按【伺服准备】键，接通伺服电源。
3. 请在状态栏确认当前再现速度倍率。



4. 按示教编程器的【启动】按钮，机器人把示教后的程序运行一个循环后停止。

7 用户坐标设定

目的：设定用户坐标系可以很方便地对机器人进行示教，使机器人执行机构很快地到达指定点，进而提高工作效率。

用户坐标系界面存储该项目下的用户坐标系，可通过示教操作建立用户坐标系，该用户坐标系只属于当前被加载的项目。

选择屏幕左下角图标 ，选择【用户】/【用户坐标系】，进入用户坐标系界面操作

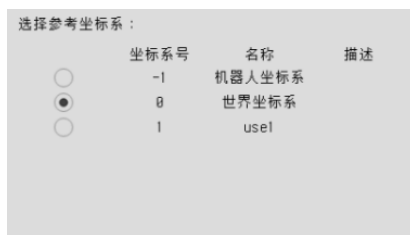
7.1 用户坐标系的调用

7.1.1 手动调用

1. 点击状态栏中的参考坐标系图标，即图中红色线框。



2. 弹出参考坐标系选择窗口，勾选中用户坐标系前的圆圈。



3. 用户坐标系“use1”被调用，用户坐标系数值发生变化。



7.1.2 REFSYS 指令设定

REFSYS 是设置参考系指令。通过该指令可以为后续运行的位置指令设定一个新的参考坐标系。如果程序中没有设定参考坐标系，系统默认参考坐标系为世界坐标系。

7.2 用户坐标设置

用户坐标最多可输入 64 个。

选择主菜单图标 ，在选择“项目”里的“用户坐标系” ，进入如下界面。

新建名为“use1”的用户坐标系。

首先要设置基准坐标系，以该坐标系为参考建立新的参考坐标系即用户坐标系。默认的参考坐标系为世界坐标系。如下图所示



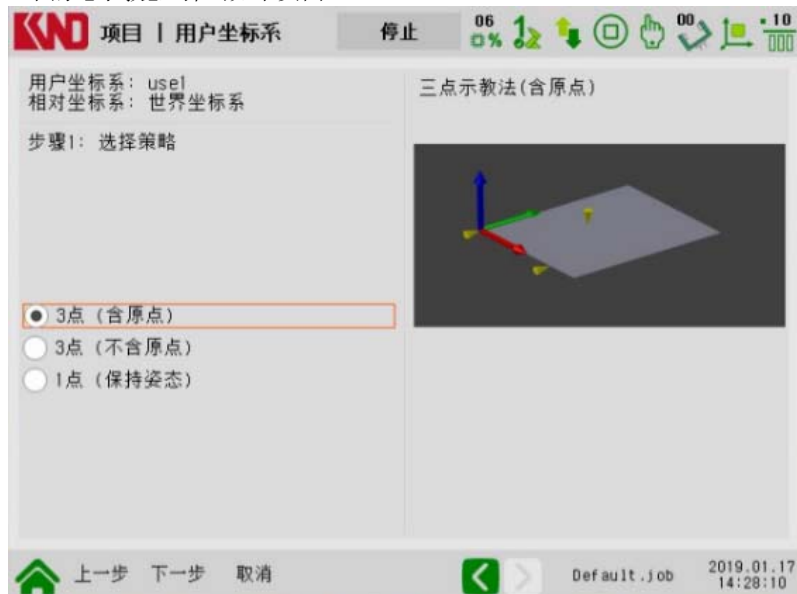
相对参考坐标系的位置和姿态		TCP在世界坐标系的位置和姿态	
数值	坐标属性	数值	
X 0.000	常值	X 935.500	
Y 0.000	常值	Y 0.000	
Z 0.000	常值	Z 1210.123	
A 0.000	常值	A 100.000	
B 0.000	常值	B -90.000	
C 0.000	常值	C 0.000	

选择【示教】弹出示教方法选择界面。

- 三点（含原点）示教法
- 三点（不含原点）示教法
- 一点（保持姿态）示教法

7.2.1 三点（含原点）示教法

三点（含原点）示教法是以操作机器人三个点来定义的，按下右上角“操作菜单”下的【示教】，弹出如下页面：



第一个示教点：首先示教原点。

第二个示教点：X、Y、Z 任意轴上点。

第三个示教点：与第二个点（轴上点）所在平面的点。

示教三个点成功后，三个点的位置数据被输入到用户坐标文件里，计算出了用户原点位置。

那么接下来开始我们的示教过程：

登录用户坐标系后，选择屏右方的【示教】，显示示教页面按照步骤进行示教。

步骤 1: 选择策略, “●”即选中了示教方法, 请选择“3 点(含原点)”示教, 继续【下一步】。

步骤 2: 通过轴操作键将机器人移到 ORG 位置, 选择【示教】, 继续【下一步】。



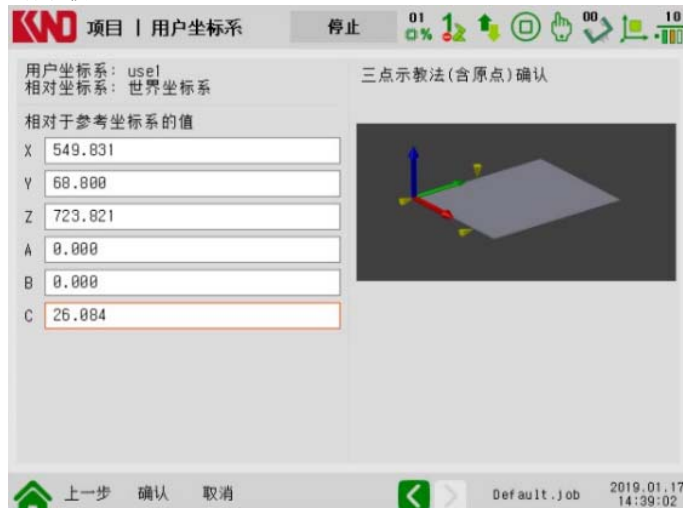
步骤 3: 示教轴上点, “●”即选中了要示教的轴点, □选择方向, 通过轴操作键将机器人移到想要建立的用户坐标系与之相对应的轴上点, 选择【示教】, 继续【下一步】。



步骤 4: 示教平面点, “●”即选中了要示教的平面点, □选择方向, 通过轴操作键将机器人移到建立的用户坐标系与之相对应的平面点, 选择【示教】, 继续【下一步】。



步骤 5: 确认示教数据是否正确, 如果正确选择【确认】完成, 否则继续【上一步】重新示教。



步骤 6: 选择【确认】完成, “use1” 用户坐标系示教结束, 结果保存完成。

7.2.2 三点（不含原点）示教法

第一个示教点: 首先示教一个点是用户坐标系的任意轴上点。

第二个示教点: 在第一示教点的轴上继续选择轴上点判断方向。

第三个示教点: 选择坐标系另外一个轴上点, 确定了方向。

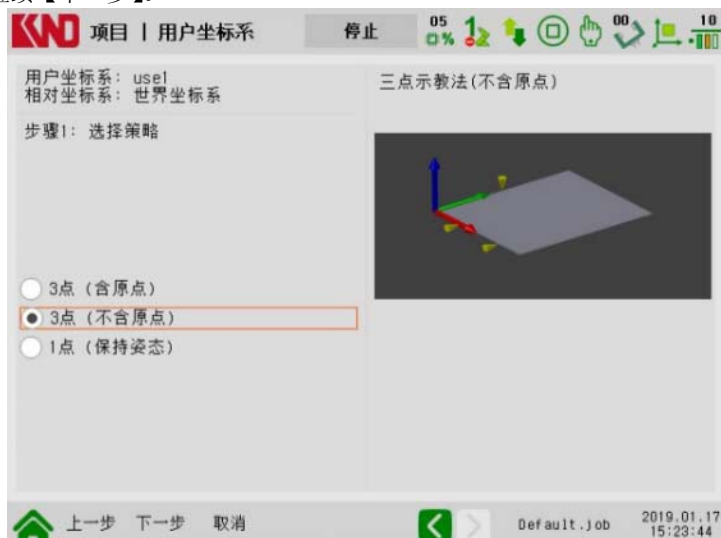
示教三个点成功后, 三个点的位置数据被输入到用户坐标文件里, 建立用户坐标位置。

那么接下来开始我们的示教过程:

新建名为“use1”的用户坐标系。

设置用户坐标系后, 选择屏幕左下方的【示教】, 显示示教页面按照步骤进行示教。

步骤 1: 选择策略, “●”即选中了示教方法, 请选择“3 点 (不含原点)”示教, 继续【下一步】。



步骤 2: 示教第一个轴上点, “●”即选中了用户坐标系轴, □选择方向, 通过轴操作键将机器人移到示教位置, 选择【示教】, 继续【下一步】。



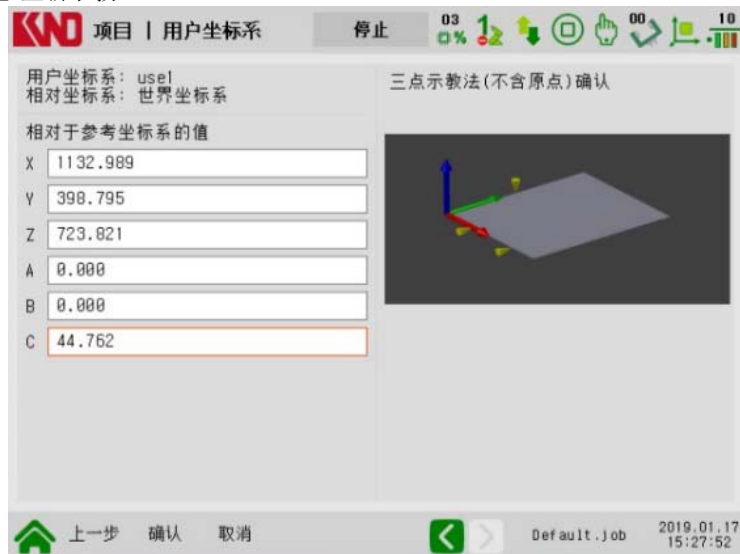
步骤 3: 示教轴上的第二个点, 按照第一个示教点的轴与方向去延伸选择第二点, 通过轴操作键将机器人移到示教位置选择【示教】, 继续【下一步】。



步骤 4: 示教第三个点, “●”即选中用户坐标系另外两个未示教过的轴上点, □选择方向, 通过轴操作键将机器人移到示教位置, 选择【示教】, 继续【下一步】。



步骤 5：确认示教数据是否正确，如果正确选择【确认】完成，否则继续【上一步】重新示教。



步骤 6：选择【确认】完成，“use1”用户坐标系示教结束，结果保存完成。



7.2.3 一点（保持姿态）示教法

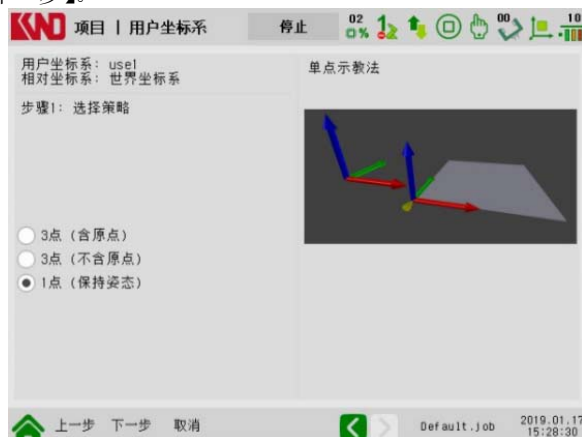
一点示教法，以世界坐标为参考，方向已经确定，与世界坐标方向保持一致，该点的选择是在用户坐标系上选择一个原点位置即可设置成功。

那么接下来开始我们的示教过程：

新建名为“use1”的用户坐标系。

设置用户坐标系后，选择屏右方的【示教】，显示示教页面按照步骤进行示教。

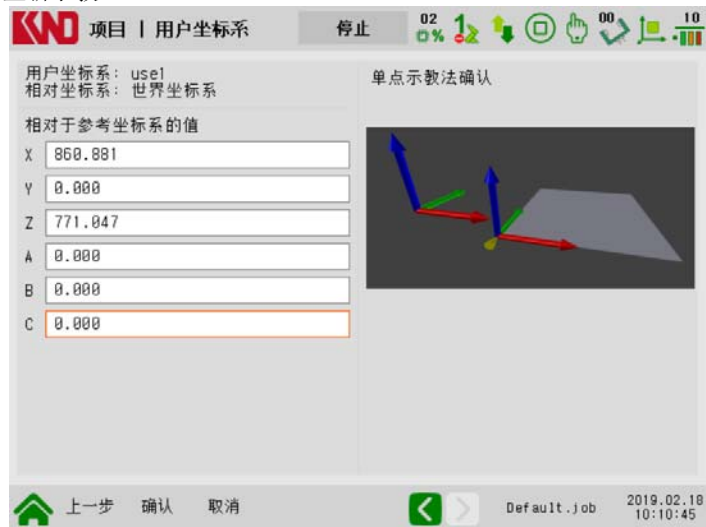
步骤 1：选择策略，“●”即选中了示教方法，请选择“一点（保持姿态）”示教，继续【下一步】。



步骤 2：示教坐标原点，通过轴操作键将机器人移到示教位置，选择【示教】，继续【下一步】。



步骤 3：确认示教数据是否正确，如果正确选择【确认】完成，否则继续【上一步】重新示教。



步骤 4：选择【确认】完成，“use1”用户坐标系示教结束，结果保存完成。



8 工具尺寸设定

目的：对于一个具体的搬运对象，需要合理的前端抓取机构。在工业机器人内部需要对前端抓起机构的尺寸、重量、重心、惯性矩等参数进行设置，才能使机器人在合适的状态工作，减少机器人的故障率。

该工具坐标系只属于当前被加载的项目。

选择屏幕左下角图标 ，选择【用户】/【工具】，进入工具界面操作

8.1 工具坐标的调用

8.1.1 手动调用

1. 点击状态条当前工具图标。



2. 弹出工具选择窗口，选中工具前的圆圈。



3. 当前工具被更改。



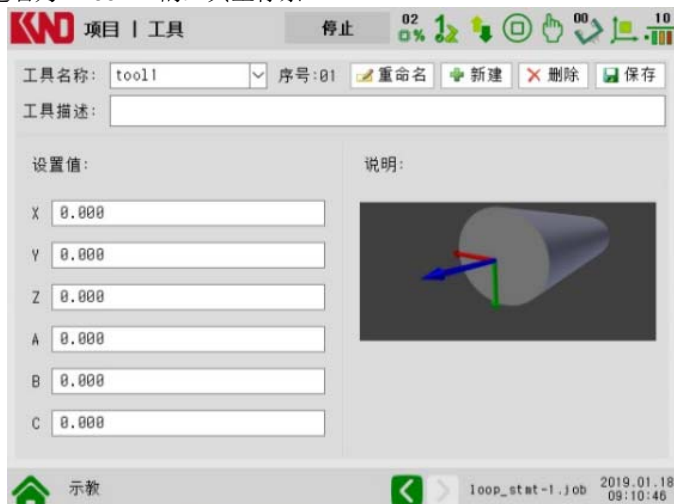
8.1.2 TOOL 指令设定

TOOL 为设置工具坐标指令，为机器人设置一个新工具坐标，通过该指令可以修改机器人末端工作点。

8.2 工具坐标设置

工具文件最多可以设置 64 个

选择主菜单图标 ，在选择项目  里的“工具”，进入如下界面。
新建名为“Tool1”的工具坐标系。



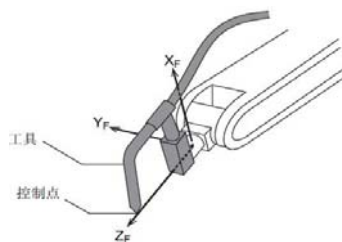
工具手动设定

- 工具坐标值设置。
- 工具姿势数据设置。

8.2.1 工具坐标值设置

工业机器人在工作时需要在前端安装相应的作业工具，如下图所示显示是一种典型工具安装示意图。为了便于对机器人进行示教，需要对工业机器人参数进行设

定。用数值输入设置工具文件时，把工具的控制点位置作为法兰盘坐标各轴上的坐标值来输入。



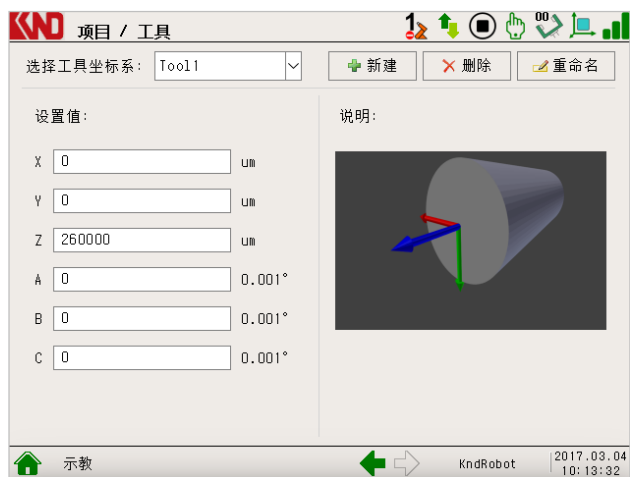
工具坐标值的设置步骤如下：

步骤 1：选择主菜单【项目】，请选择【工具】，打开设置工具坐标系页面。

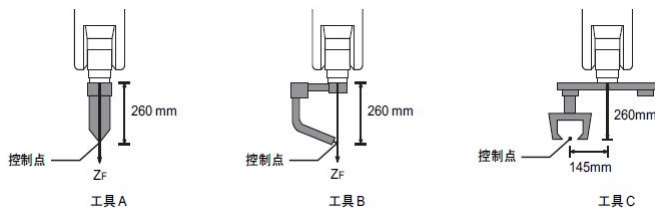
步骤 2：选择工具坐标系，在设置坐标系页面，把光标移动到“选择工具坐标系”，选择想要示教的坐标系，或者“新建”一个新的用户坐标系。

步骤 3：选择想要设定的坐标值，把光标移动到画面左侧下列表的坐标。

步骤 4：使用数值键输入相应的数值。如下图所示。



对于不同的工具，其工作点坐标的设定是不同的，如图，在图中标出了其工作点的尺寸位置。



● 工具 A、B 的设定

X	0	um
Y	0	um
Z	260000	um
A	0	0.001°
B	0	0.001°
C	0	0.001°

● 工具 C 的设定

X	0	um
Y	145000	um
Z	260000	um
A	0	0.001°
B	0	0.001°
C	0	0.001°

8.2.2 工具姿势数据设置

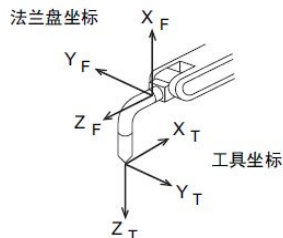
工具姿势数据是指表示机器人法兰盘坐标和工具坐标的角度数据。**输入值是把法兰盘坐标和工具坐标调整一致时的角度数据。沿着法兰盘各轴坐标正向（即沿着箭头方向）看，右旋转是正方向，左旋转为负方向（即顺正、逆负）。按照 Rz—Ry—Rx 的顺序登录。如右图，C=180、B=90、A=0。**

工具姿势数据设定步骤如下所示：

步骤 1：选择主菜单【设置】。

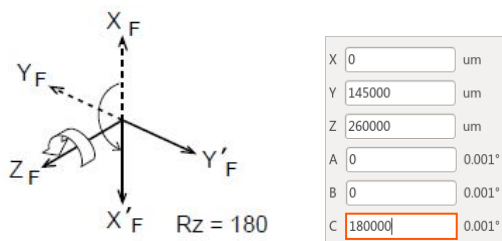
步骤 2：选择【工具】。

步骤 3：选择想要设定的工具号。

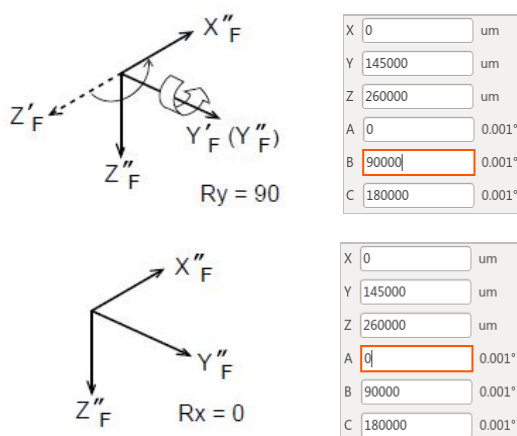


步骤 4: 想要选择设定坐标系的轴, 首先选择 C。

步骤 5: 输入数值回转角度, 用数字键输入法兰盘坐标 Z_F 周围回转角度。



步骤 6: 按【回车】, C 的回转角度被设定。同样的操作, 输入 B、A 的回转角度, 如图。



8.3 工具自动校准

为了给机器人正确地进行直线插补、圆弧插补等插补动作, 有必要正确的设置焊枪、抓手、焊钳等工具的尺寸信息, 定义控制点位置。

工具自动校准是指为了能够容易并正确的进行尺寸信息输入的功能。利用此功能, 工具控制点的位置自动计算出来, 设置到工作坐标系里 (在工具校准里设置的

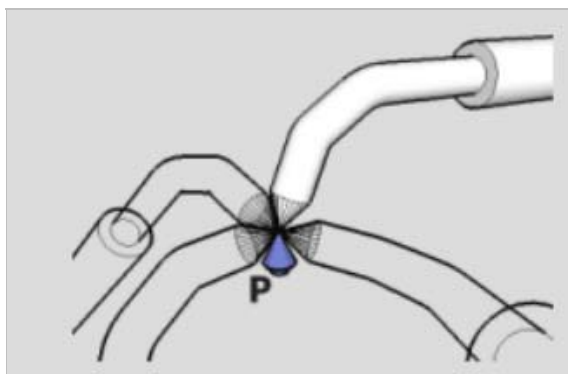
是法兰盘上的控制点的坐标值和工具姿势)。

关于工具校准有以下两种，常用的是四点（保持姿态）示教法。

- **一点（保持姿态）示教法**
需要示教两个点，一个已知参考点与该工具点对同一个点先后进行示教，通过已知参考点计算出该工具点的尺寸，已知参考点可以是已知尺寸的工具或者机械手未装任何工具的法兰控制点，示教结果被设定到工具坐标系里，这种情况不改变姿势数据。
- **四点（保持姿态）示教法**
对同一点采取示教 4 个不同角度的位置，计算出来工具尺寸，被设定到工具坐标系里，这种情况不改变姿势数据。

8.3.1 四点（保持姿态）示教法

控制点 TCP 末端示教到示教物体处以不同的 4 个姿态。



示教前要先选好想要设定的工具坐标系，选择主菜单【项目】，请选择【工具】。

登录工具坐标系 Tool1 后，选择左下方的【示教】，显示示教页面，按照步骤进行示教。

步骤 1: 选择策略, “●” 即选中了示教方法, 请选择 “4 点 (保持姿势)” 示教, 继续【下一步】。



步骤 2: 示教第 1 个姿态点, 通过轴操作键将机器人移动到尖端示教点位置, 选择【示教】, 继续【下一步】。



步骤 3：示教第 2 个姿态点，通过轴操作键将机器人移动到尖端示教点位置，选择【示教】，继续【下一步】。



步骤 4：示教第 3 个姿态点，通过轴操作键将机器人移动到尖端示教点位置，选择【示教】，继续【下一步】。



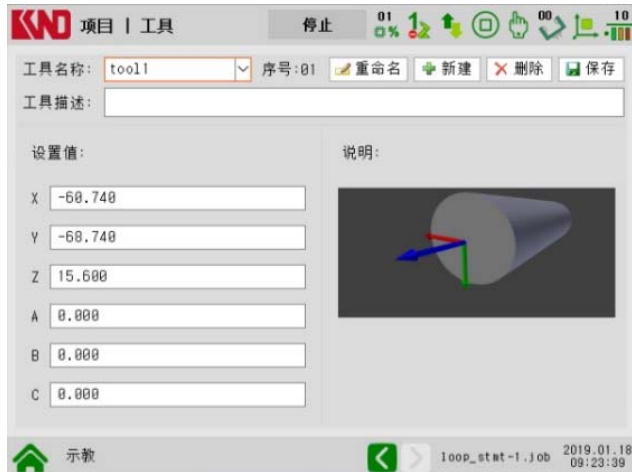
步骤 5: 示教第 4 个姿态点, 通过轴操作键将机器人移动到尖端示教点位置, 选择【示教】, 继续【下一步】。



步骤 6: 示教结束, 已计算出工具尺寸, 需要确认结果, 否则继续【上一步】重新示教。



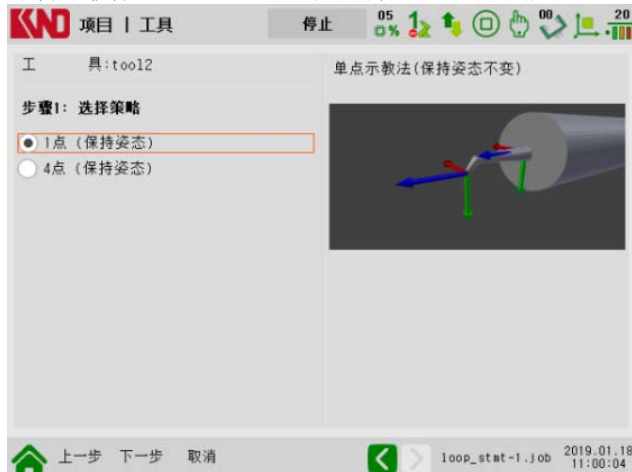
步骤 7: 选择“确认”，示教结果保存到坐标系里。



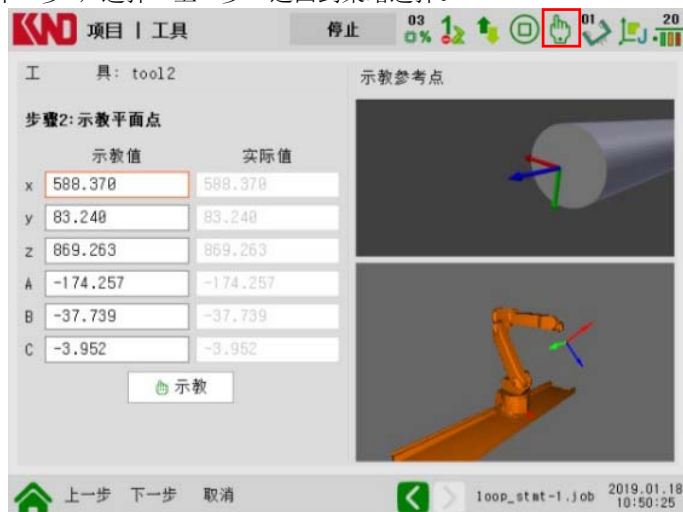
8.3.2 一点（保持姿态）示教法

新建工具坐标系 Tool2 后，选择左下方的【示教】，显示示教页面按照步骤进行示教。本例中将 Tool1 作为尺寸的工具，将 Tool1 和 Tool2 分别对同一个点进行示教。

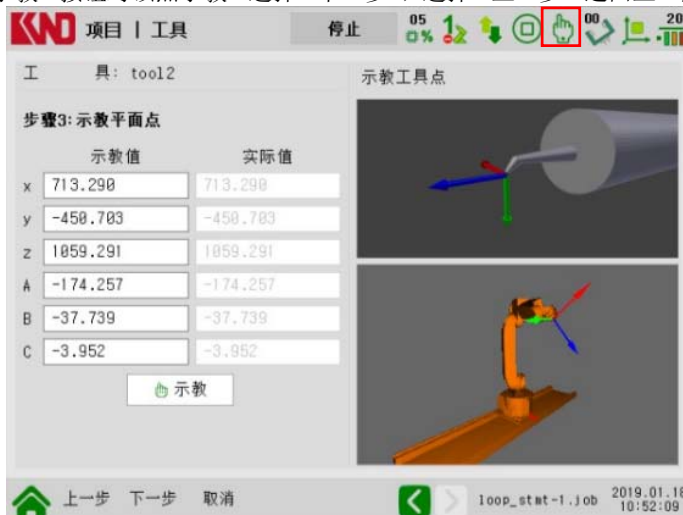
步骤 1: 选择示教方法，勾选中“1 点（保持姿态）”前的圆圈，选择“下一步”。如图所示。放弃示教操作选择“取消”，页面会自动回到工具页面。



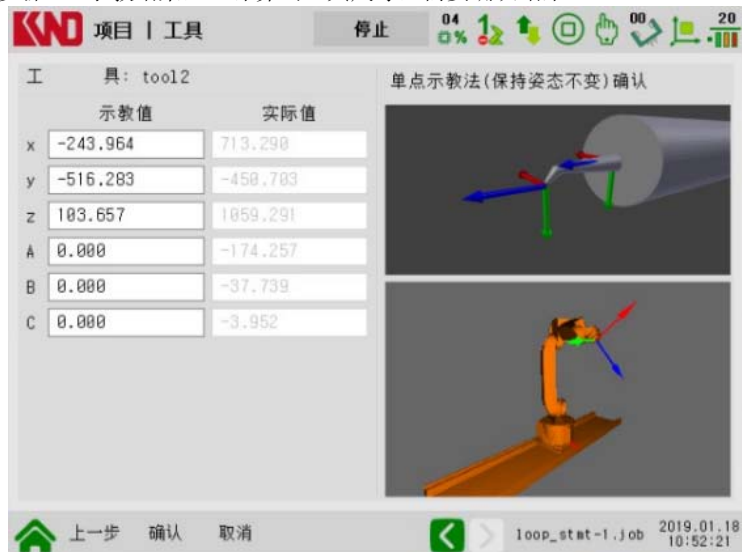
步骤 2: “示教参考点”, 安装 01 号工具手并在状态栏中选择 01 号工具 Tool1, 通过轴操作键, 将机器人移动到尖端示教点位置, 点击 “示教” 按钮对该点示教。选择 “下一步”, 选择 “上一步” 返回到策略选择。



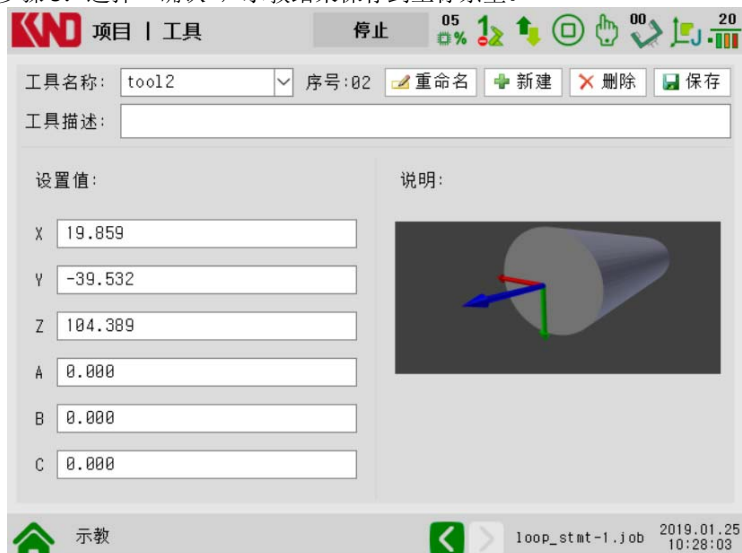
步骤 3: “示教工具点”, 卸下工具手 01, 安装尺寸未知的工具手 02, 在状态栏中选择 0 号工具 FLANGE, 通过轴操作键, 将机器人移动到同一尖端示教点位置, 点击 “示教” 按钮对该点示教。选择 “下一步”, 选择 “上一步” 返回上一点示教。



步骤 4: 示教结束，已计算出工具尺寸，需要确认结果。



步骤 5: 选择“确认”，示教结果保存到坐标系里。



8.4 工具对齐

选择屏幕左下角图标 ，选择【设置】/【工具对齐】，进入工具对齐界面操作，目的是把当前使用的工具手坐标系，实现与目标参考坐标系的 x, y, z 轴方向平行的设定。



操作说明：目标参考坐标系有世界坐标系、机器人坐标系、用户坐标系三种，选择合适的参考坐标系，计算当前**偏差角度**，**就绪**被点亮激活，计算最终**目标轴**的方向。在**对齐操作**选项，选择对齐轴，按照设定好的速度执行直到与目标参考轴方向平行即该轴对齐停止，当对齐操作完成时，该轴的**对齐完成**选项被点亮激活。

只有在按住轴对齐按钮时才会有动作执行，直到对齐完成，对齐过程中松开按钮动作立刻停止。

参考：

对齐速度设定在 P0950 号参数，默认值 30deg/s。

9 用户数据备份与升级

在 HMI 界面中使用导入导出功能可以将示教器和控制器的数据进行导入导出、升级等操作。使用导入导出时将 u 盘插入示教器背面的 u 盘接口即可操作，在升级过程中保证不要断电合闸，否则会升级不成功或者异常，可能导致原系统文件也无法正常运行。

导入导出时支持文件放在 u 盘多级文件夹路径下导入，只需选择文件存放的路径选择文件导入即可。

9.1 文件简介

导入导出文件时各个文件说明如下：

类别	名称	说明	导入导出
示教器	hmi_v****_panel.bin	HMI 升级包	只可导入升级
	log.txt	示教器日志文件	只可导出
控制器	cfg.tar.gz	控制器配置文件打包	可导入导出
	para.tar.gz	控制器参数文件打包	可导入导出
	prg.tar.gz	控制器项目和程序文件打包	可导入导出
	robot.tar.gz	控制器所有文件打包	可导入导出，导入可用于升级。

注意：

1. 示教器升级相当于导入示教器文件，升级时需要导入的文件包名字后缀为.bin，例如 hmi_v****_panel.bin，hmi 表明为示教器界面所用升级包，v****为发布版本。
2. 默认文件名请勿修改，否则导入时可能无法正常导入。
3. robot.tar.gz 是控制器所有文件打包，不分通道，控制器配置、参数、程序文件包是属于 robot.tar.gz 中的一部分，并且分通道。
4. 备份控制器时建议分别备份 robot.tar.gz、配置文件、参数、项目，如果导入 robot.tar.gz 控制器文件包，会改变原系统版本、PLC 等所以根据实际情况导入所需文件。

9.1 示教器升级操作

当示教器更新了软件版本（例如示教器新增了某些界面、界面新增了某些控件等），就需要升级示教器软件版本，升级示教器相当于把示教器的界面文件升级包全部导入到示教器中，并替换原有的示教器界面文件。可通过查看示教器界面软件版本号知道当前的示教器界面软件版本。

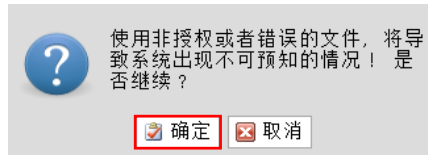
选择屏幕左下角图标 ，选择【IO】/【数据备份】，进入到数据备份页面如下：



升级示教器

升级示教器的操作步骤如下：

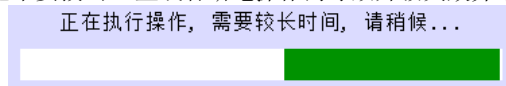
1. 升级之前进入【监视】/【控制器】/【概况】中，查看并记下当前的示教器 HMI 版本号。
2. 将示教器升级包 hmi_v****_panel.bin 后缀名为.bin 考入 u 盘中，然后将 u 盘插入示教器后面的 usb 接口。
3. 进入数据备份页面，选择“升级示教器”，弹出信息确定对话框。



4. 选择“确定”后弹出文件对话框，选择升级文件包。



5. 如上图：选中“确定”按钮，HMI 开始升级，弹出如下进度条。升级过程中切记不要拔出 U 盘或者断电操作而导致升级失败异常。



6. 导入期间界面示教器相关各按键会变为灰色，请耐心等待，大约几秒钟后弹出是否重启对话框。
7. 在“是否重启对话框”中，选择“是”，示教器屏幕变黑，然后重启。
8. 重启完毕后，升级之前进入【监视】/【控制器】/【概况】中，查看并确认示教器 HMI 版本号已经更新。

9.2 控制器导入与导出

在界面里，备份控制器可以分别导入导出**配置、程序、参数**，也可以将其全部导入导出包含软件版本以及 PLC 等。

使用控制器全部导入相当于将控制器文件全部替换为导入的文件。若用高版本的控制器文件全部导入，相当于将控制器软件升级为高版本。

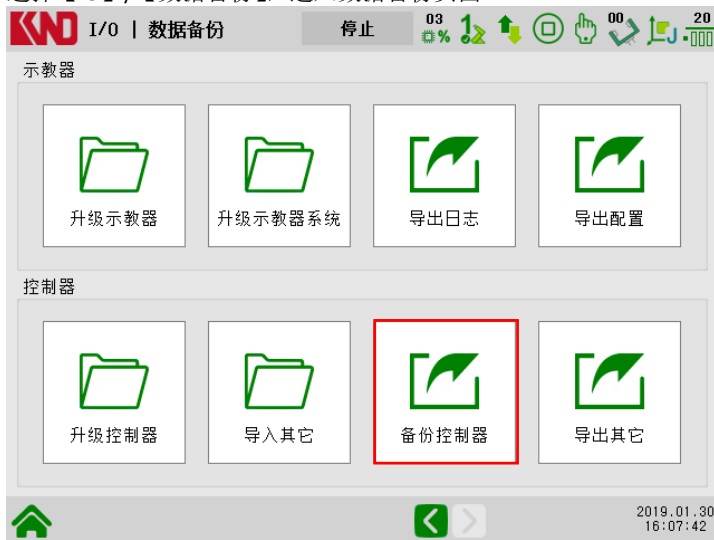
备份控制器时，不分通道，是将控制器中所有文件打包导出。导入时也无需选择通道，全部导入即可。

控制器单独导入配置、程序、参数时，是分通道的，两个通道均有各自的配置、程序、参数，因此导入导出时一定注意选择是哪个通道的。导出时文件均存放于 u 盘主目录下，而且两个通道导出的参数、程序、配置文件名字相同，因此导出第一个通道的参数、程序、配置后，建议将其放在文件夹中，建议文件夹命名为“ch0”，表明为通道 1 的数据。然后再导出第二个通道的参数、程序、配置，将其放在另一个文件夹“ch1”中，表明为通道 2 的数据。导入数据时将 u 盘目录定位到“ch0”或者“ch1”文件夹内，然后再选择导入的数据。

9.2.1 备份控制器

备份控制器操作如下：

1. 将 u 盘插入示教器后面的 usb 接口。
2. 选择【IO】/【数据备份】，进入数据备份页面。



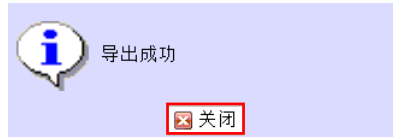
3. 在页面下方的【控制器】部分中，选择“备份控制器”。



4. 如上图：  按钮进入到选中文件的下一级菜单，选中“确定”按钮，开始备份，弹出如下进度条。备份过程中切记不要拔出 U 盘或者断电，避免备份失败，相同名字的文件会被覆盖。

正在执行操作，需要较长时间，请稍候...

5. 备份期间界面示教器相关各按键会变为灰色，请耐心等待，大约几秒钟后弹出完成对话框。



6. 关闭对话框即可，在 u 盘根目录下的 KNDRobot 文件夹里存放着名称为“robot.tar.gz”控制器打包文件。
7. 将文件放入电脑中保存时，建议文件夹名字用控制器的版本命名，例如 robot_v2019。控制器版本可进入示教器【监视】/【控制器】/【概况】界面中查看。

注意：

使用导入导出控制器全部时，所需时间跟文件的大小有关，例如如果控制器中有很多较大的项目和程序，导入导出的时间也相应变长，请耐心等待。

9.2.2 备份配置、程序、参数

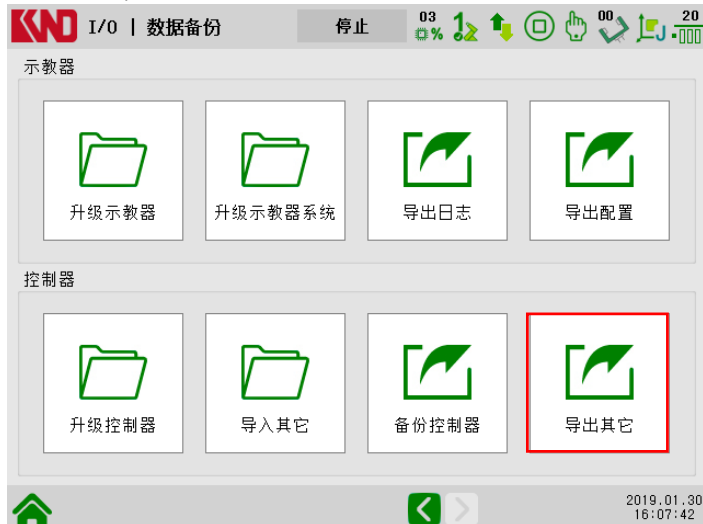
控制器导出配置、程序、参数可用于分别将两个通道内的数据单独备份。

由于两个通道导出的参数、程序、配置文件名字相同，因此导出第一个通道的参数、程序、配置后，建议将其放在文件夹中，文件夹命名为“ch0”。然后再导出第二个通道的参数、程序、配置，将其放在另一个文件夹“ch1”中。

导出配置、程序、参数的操作步骤类似，只需在导出时选择不同的数据即可。另外一定要注意选择正确的通道。

备份配置、程序、参数的步骤如下：

1. 将u盘插入示教器后面的usb接口。
2. 选择【IO】/【数据备份】，进入数据备份页面。



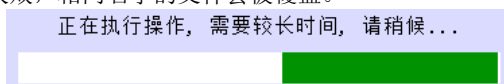
3. 在页面下方的【控制器】部分中，选择“导出其它”。弹出如下对话框：



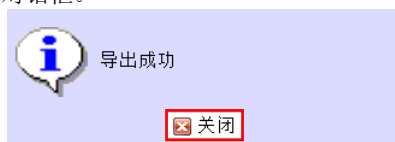
4. 勾选的方式选择想要导出的数据，还要选择哪个通道，上图中选择备份通道 1 的配置文件，点击“浏览”按钮弹出目标文件夹窗口，如下：



6. 如上图：  按钮进入到选中文件的下一级菜单，选择号目标文件夹“KNDRobot”，选中“确定”按钮，继续选择““确定”按钮”，开始备份，弹出如下进度条。备份过程中切记不要拔出 U 盘或者断电操，避免备份失败，相同名字的文件会被覆盖。



7. 备份期间界面示教器相关各按键会变为灰色，请耐心等待，大约几秒钟后弹出完成对话框。



关闭对话框即可，在 u 盘根目录下的 KNDRobot 文件夹里存放着名称为“cfg.tar.gz”控制器配置打包文件。

按照上面的步骤操可分别导出的控制器的参数、程序备份文件。

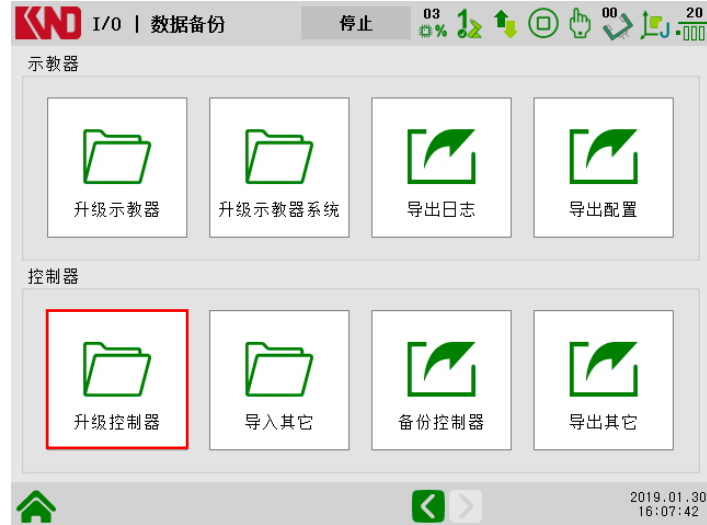
9.2.3 升级控制器

升级控制器相当于把控制器打包的文件考入控制器内，可用于控制器升级。控制器导入的全部的操作如下：

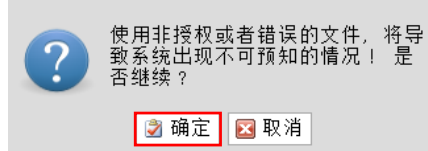
1. 升级控制器之前，建议先备份控制器原文件，方便数据还原，备份的步骤上面都有介绍，先进入【监视】/【控制器】/【概况】中，查看并记下当

前的控制器版本号。

2. 将控制器文件考入 u 盘，控制器升级文件名字 **robot.tar.gz**，然后将 u 盘插入示教器后面的 **usb** 接口。
3. 选择【IO】/【数据备份】，进入数据备份页面。



4. 下方的【控制器】部分中，选择“升级控制器”。弹出信息确定对话框。



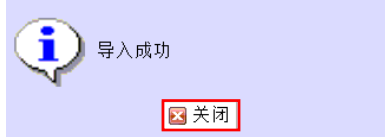
5. 选择“确定”后弹出文件对话框，选择升级文件包。



6. 如上图：选中“确定”按钮，控制器开始升级，弹出如下进度条。升级过程中切记不要拔出 U 盘或者断电操作而导致升级失败异常。

正在执行操作，需要较长时间，请稍候...

7. 升级期间界面示教器相关各按键会变为灰色，请耐心等待，大约几秒钟后弹出是导入成功对话框。



8. 重启控制器，点击主菜单 ，选择【设置】/【其它】/【重启控制器】。
9. 重启完毕后，查看并确认控制器版本号已经更新。

9.2.4 导入配置、程序、参数

1. 将 u 盘插入示教器后面的 usb 接口。
2. 选择【IO】/【数据备份】，进入数据备份页面。



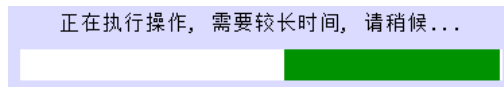
3. 在页面下方的【控制器】部分中，选择“导入其它”。弹出如下对话框：



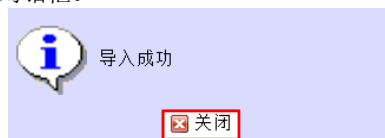
4. 勾选的方式选择想要导入的数据，还要选择哪个通道，上图中选择备份通道 1 的配置文件，点击“浏览”按钮弹出目标文件夹窗口，如下：



5. 如上图： “进入”按钮进入到选中文件的下一级菜单，选择号目标文件夹“cfg.tar.gz”，选中“确定”按钮，继续选择““确定”按钮”，开始备份，弹出如下进度条。导入过程中切记不要拔出 U 盘或者断电操，避免导入失败。



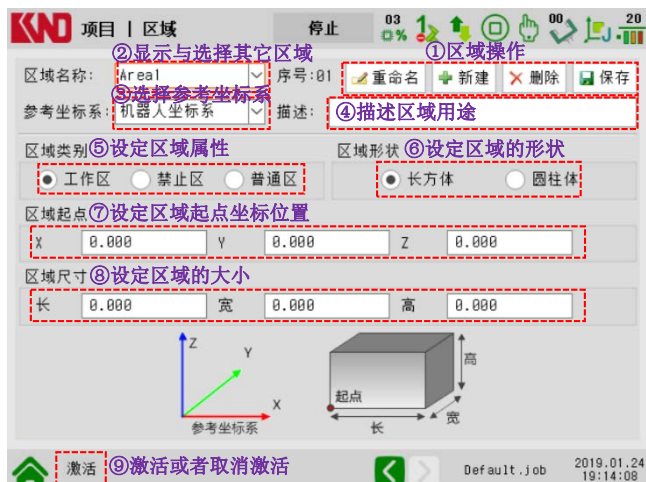
6. 导入期间界面示教器相关各按键会变为灰色，请耐心等待，大约几秒钟后弹出完成对话框。



关闭对话框即可，按照上面的步骤操可分别导入的控制器的参数、程序备份文件。

10 区域监控

10.1 区域



参考坐标系：区域的参考坐标系分为机器人坐标系、世界坐标系、用户坐标系三种，参考坐标系的设定，说明区域与该参考坐标轴是平行的。默认配置是机器人坐标系。

新建区域：首先需要点击“新建”，输入区域名，建立新的区域。

选择参考坐标系：配置区域的参考坐标系

选择区域类别：有工作区域、禁止区域、普通区域。

选择区域形状：有长方体和圆柱体两种。

配置区域的起点坐标 xyz：起点坐标是相对于参考坐标系设定的。

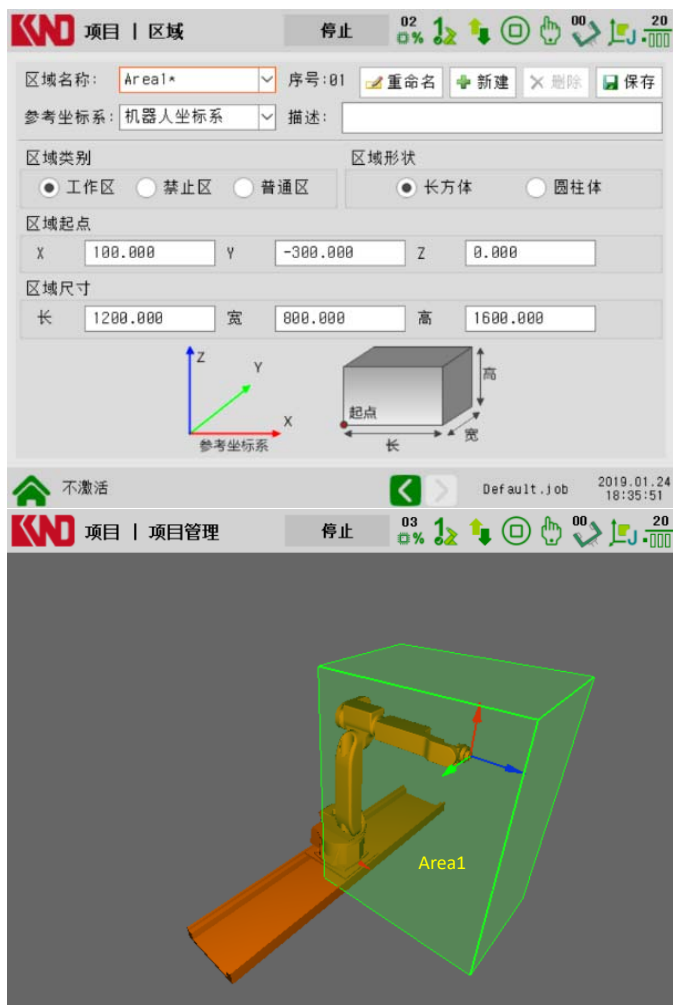
配置区域尺寸：长方体区域需要输入长、宽、高三个尺寸，对于圆柱体，需要配置底圆半径和圆柱体的高度。

激活区域后自动判断 TCP 点在区域内部或区域外部，只有区域被激活，区域才会在模型附近被显示出来。

激活区域：选择左下方“激活”按钮，当前设定区域生效。

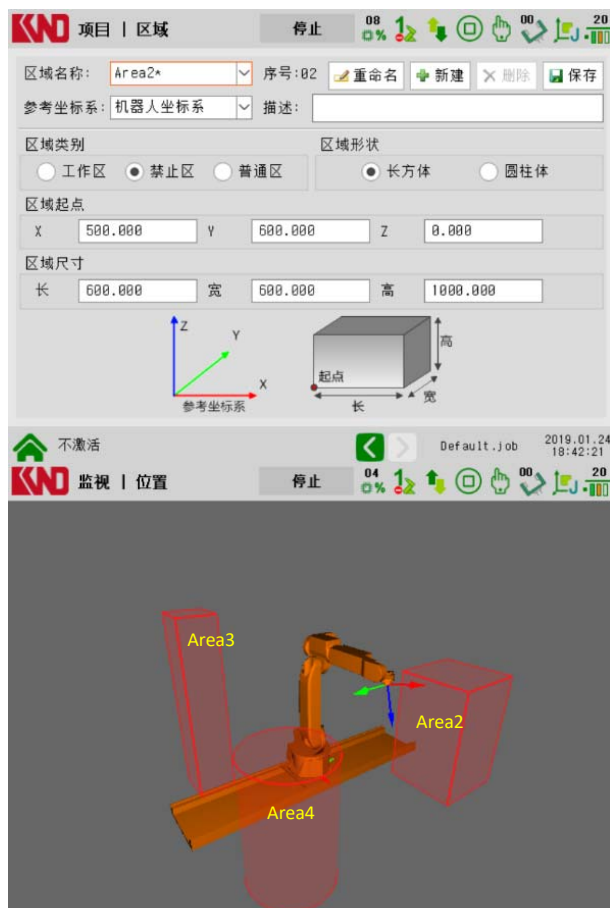
10.1.1 工作区域被激活状态

下图所示是工作区域的设置，点击“激活”按钮即可激活该区域，工作区域用绿色显示，如下所示。



10.1.2 禁止区域被激活状态

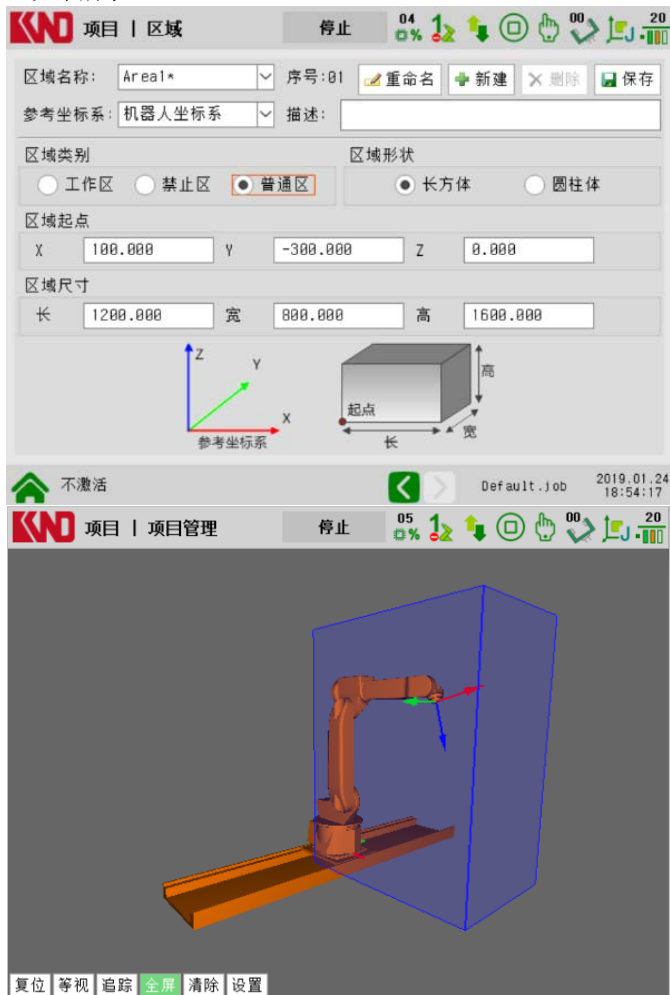
下图所示是禁止区域的设置，点击“激活”按钮即可激活该区域，禁止区域用红色显示，可以同时激活多个禁止区域，如下所示。



10.1.3 普通区域被激活状态

当出现控制点 TCP 从工作区移出，或者 TCP 进入禁止区导致报警无法操作，可先设定为普通区，轴操作机器人移到合理位置，再恢复区域属性。

下图所示是普通区域的设置，点击“激活”按钮即可激活该区域，普通区域用蓝色显示，如下所示。



11 特殊功能包介绍

11.1 传送带

传送带跟踪功能，应用在工业、制造业、食品、医药、物流等领域中，在自动化生产线中多采用流水线形式，为了提高效率，通过传送带连接下一个工位，待加工或者待挑拣的工件跟随传送带，从一个工位到达另一个工位，由工位上的机器人从传送带上抓取或放置。

11.2 视觉

资料正在更新

11.3 冲压工艺

详见冲压工艺说明书

11.5 焊接工艺

详见焊接工艺说明书

11.6 料盘

参考《编程向导》篇

11.4 力传感器

力传感器用于牵引示教，当使用牵引示教时，操作者通过拖动机器人末端安装的直接示教手柄进行示教，通过力传感器感知操作者的示教力以及力矩，分别为 F_x , F_y , F_z , T_x , T_y , T_z ，将六个方向力/力矩大小方向变化转为相应位移，同时记录示教过程中的特征轨迹点。

在力传感器状态界面，可以显示传感器的状态以及传感器的数值。

传感器状态界面的指示灯可显示传感器是否正常，

未连接时 **Ready** 灯为灰色，如下所示。连接正常状态时 **Ready** 灯为绿色，信号错误时 **Error** 灯为红色，如果使用时的力或力矩超过传感器最大负载，“ $F_{x0V} \sim T_{z0V}$ ”指示灯会变为红色。



在力传感器信息界面，可显示力传感器的相关信息和连接状态。当传感器未连接或连接异常时，会提示力传感器未连接。当连接正常时，会显示传感器相应的信息。



界面下方的传感器数值显示条可实时动态显示力或力矩的大小变化，在显示框中显示力或力矩的数值。

画面右侧可设定传感器显示值的范围，可使用 、 调整数值范围，或直接在输入框中输入数值。

当传感器不受力时，各力、力矩显示数值在 0 附近，若显示数值有较大偏差时，按下“清零偏”按钮，使显示数值为 0。

12 KRL 指令一览表

12.1 运动指令

MOVJ	功能	以关节插补方式向示教位置移动	
	添加项目	位置数据	C 常量/P 项目变量/LP 局部变量/\$P 系统变量/P[变量别名]
		VJ= (再现速度)	VJ: 0%~100%
		PL= (定位等级)	PL: 0~8
		NMTCON (非运动指令执行条件)	
		UNTIL (运动结束条件)	UNTIL: IN# (输入 I 地址) ==状态
		ACC= (加速度调整比例)	ACC:
		DEC= (减速度调整比例)	DEC:
		IMOV (增量运动)	
	使用示例	MOVJ P0000 VJ=50 PL=4	
MOVL	功能	以直线插补方式向示教位置移动	
	添加项目	位置数据	C 常量/P 项目变量/LP 局部变量/\$P 系统变量/P[变量别名]
		V= (再现速度)	V: mm/s
		VR= (姿态的再现速度)	VR: deg/s
		VE= (外部轴的再现速度)	VE: deg/s; mm/s
		PL= (定位等级)	PL: 0~8
		NMTCON (非运动指令执行条件)	
		UNTIL (运动结束条件)	UNTIL: IN# (输入 I 地址) ==状态
		ACC= (加速度调整比例)	ACC:
		DEC= (减速度调整比例)	DEC:
	使用示例	MOVL P0001 V=1200 PL=3	
MOVC	功能	以圆弧插补方式向示教位置移动	
	添加项目	位置数据	

		V= (再现速度) VR= (姿态的再现速度) VE= (外部轴的再现速度)	VJ: 0%~100% VR: deg/s VE: deg/s; mm/s
		PL= (定位等级)	PL: 0~8
		NMTCON	
		ACC= (加速度调整比例)	ACC:
		DEC= (减速度调整比例)	DEC:
		CST (下一个起点)	
	使用示例	MOVC P0002 V=200	
NOP	功能	空操作;	
	说明	空操作指令通常用在程序开始; 编程设计时, 某些未实现详细设计的模块也可以先用 NOP 占用一个语句行。	
END	功能	程序结束;	
	说明	END 指令结束本任务, END 之后的指令将无效。	

12.2 设置指令

RAMP	功能	设置加、减速度类型	
	添加项目	ACC=POLY0 直线 ACC=POLY1 多项式 ACC=SINE0 三角函数 0 ACC=SINE1 三角函数 1	
		DEC=POLY0 直线 DEC=POLY1 多项式 DEC= SINE0 三角函数 0 DEC= SINE1 三角函数 1	
	使用示例	RAMP ACC=POLY1 DEC= SINE0	
REFSYS	功能	设置参考系统指令	
	添加项目	坐标系号	
	使用示例	REFSYS “CSYS1”	
SPEED	功能	设置再现速度	
	添加项目	VJ=(关节速度)	VJ: 与 MOVJ 相同

		V=(控制点速度)	V、VR、VE： 与 MOVL 相同
		VR=(姿态角速度)	
		VE=(外部轴速度)	
		使用示例	SPEED VJ=50
TOOL	功能	设置使用的工具	
	添加项目	工具号	
	使用示例	Tool “T001”	
AREA	功能	选择当前要激活的区域	
	添加项目	区域号	
	使用示例	AREA “Work”	

12.3 I/O 指令

AIN	功能	把接口电压值读到指定的变量中	
	添加项目	D<变量号> 、LD<变量号>	
		AO#：模拟量输出端口 AI#：模拟量输入端口	
		IF 语句	
	使用示例	AIN D5 AI#（1）	
AOUT	功能	向通用模拟输出口输出设定的电压值	
	添加项目	AO#（<输出口号>）	
		<输出电压值>	
		IF 语句	
	使用示例	AOUT AO#（1）10.8	
DIN	功能	把信号状态读到指定的变量中	
	添加项目	D<变量号> 、LD<变量号>	
		IN#（<输入号>）、 IG#（<输入组号>）、 OT#（<输出号>）、 OG#（<输出组号>）	1 个点 8 个点（1 各组） 1 个点 8 个点（1 各组）
		IF 语句	

	使用示例	DIN D0001 IN# (10000.5); DIN D0007 OG# (20000)	
DOUT	功能	ON/OFF 通用输出信号	
	添加项目	OT# (<输出号>)、 OG# (<输出组号>	1 个点 8 个点 (1 各组)
		ON/OFF/INVER/D/LD	状态: 开/关/取反/变量/变量
		IF 语句	
	使用示例	DOUT OT# (20000.0) ON; DOUT OG# (20001) D0003	
PULSE	功能	通用输出信号输出脉冲	
	添加项目	OT# (<输出号>)、 OG# (<输出组号>	1 个点 8 个点 (1 各组)
		INVERT (脉冲反向)	
		T=	时间(s)
	使用示例	PULSE OT# (20000.4) T=3 PULSE OG# (200001) D0005	

12.4 控制指令

ABORT	功能	停止执行任务，并返回任务首行	
	添加项目	IF 语句	
	使用示例	ABORT IF (D0004==1)	
CALL	功能	调用指定程序	
	添加项目	“程序名” D/LD IG#	
		ARG<所有类型变量或者常数>	最多支持 8 个参数
		IF 语句	
	使用示例	CALL “knd”	
GETA	功能	用于接受 CALL 指令中的参数。当执行 GETARG 指令时，CALL 指令附加的参数数据将被检索，并存储在 GETA 指令下指定的局部变量中	

	添加项目	D 变量 S 变量 P 变量	
		#()	将要被保存参数的检索号
	使用示例	GETA LD0001 # (1)	
GETR	功能	获取 RET 指令返回值	
	添加项目	D 变量 S 变量 P 变量	
	使用示例	GETR LD0004	
JUMP	功能	向指定标号跳转	
	添加项目	Lxx LABEL= <通过端口或 D 变量/常量指定标号>	
		IF 语句	
	使用示例	JUMP L0000	
LABEL	功能	(标签) 显示跳转的目的地	
	添加项目	<跳转目的地>	
	使用示例	L0000	
RET	功能	从被调用的程序返回调用程序	
	添加项目	D/P/S <返回值>	
		IF 语句	
	使用示例	RET LD0001	
PAUSE	功能	临时暂停任务	
	添加项目	IF 语句	
	使用示例	PAUSE IF IN#(10000.3)==ON	
TIMER	功能	暂停运行，直到指定的时间结束	
	添加项目	T=<指定的暂停时间>	0~12*60*60 秒(12 小时)
		IF 语句	
	使用示例	TIMER T=10;	

WAITIO	功能	当通用 IO 信号未达到指定的状态前，始终处于待机状态	
	添加项目	IN# (<输入号>)、	1 个点
		IG# (<输入组号>)、	8 个点 (1 各组)
		OT# (<输出号>)、	1 个点
		OG# (<输出组号>)	8 个点 (1 各组)
		(状态)、D (变量)	
WATMTN	使用示例	WAITIO IN#(10000.0)==ON WAITIO IN#(10000.1)==D0008	
	功能	等待当前步 (step) 的运动指令运行结束后，继续执行。	
	添加项目	IF 语句	
	使用示例	WATMTN	

12.5 操作指令

GETE	功能	（提取位置变量元素）提取数据 2 中的指定元素的位置变量，并存储在数据 1 中 格式： GETE<数据 1><数据 2>（元素）		
	添加项目	数据 1	D<变量号>、 LD<变量号>	
		数据 2	P 变量<变量号>(<元素号>)、 LP 变量<变量号>(<元素号>)	
	使用示例	GETE D0000 P0001 (2);		
SETE	功能	（设定位置变量的元素数据）把数据 2 中的数据设定到数据 1 指定元素的位置变量中 格式： SETE <数据 1>（元素）<数据 2>		
	添加项目	数据 1	P 变量<变量号>(<元素号>)、 LP 变量<变量号>(<元素号>)	
		数据 2	D<变量号>、 LD<变量号>	
	使用示例	SETE P0002 (1) D0001;		

12.6 系统指令

ALM	功能	产生一个报警，包括报警号和报警信息字符串	
	添加项目	D/LD<数字变量> Constant<常数>	

		S/LS <字符串变量>	
	使用示例	ALM 2000 "error"	
CLS	功能	删除在显示器显示的字符	
	添加项目	无	
	使用示例	CLS	
PRINT	功能	在终端显示指定的字符串和变量	
	添加项目	" " <格式字符串>	
		S/LS <字符串变量>	
		P/LP <位置变量>	
		D/LD<数字变量>	
		Constant<常数>	
	使用示例	PRINT "Knd robot"	
GETPRM	功能	获取系统参数值到变量	
	添加项目	D 变量<存放参数值的目标>	
		PRMNO<参数号标签>	
		MAXV: 最大值 MINV: 最小值 DFTV: 默认值	
	使用示例	GETPRM D0003 PRMNO=45	
SETPRM	功能	设置系统参数值	
	添加项目	PRMNO<参数号标签>	
		DFTV: 默认值 D/LD/Constant<自定义值>	
	使用示例	SETPRM PRMNO=D0045 1500	

12.7 复合循环指令

IF

条件跳转指令，**expr** 是条件判断的表达式，每一个 **IF** 都要与之对应一个关键字 **END_IF** 做为条件结束，否则会语法错误，支持一下四种格式：

```
(1) IF expr THEN
    ...
END_IF

(2) IF expr THEN
    ...
ELSE
    ...
END_IF

(3) IF expr THEN
    ...
ELSIF expr THEN
    ...
END_IF

(4) IF expr THEN
    ...
ELSIF expr THEN
    ...
ELSE
    ...
END_IF
```

支持多级嵌套。

WHILE

WHILE 指令在满足条件的时候循环执行子语句。**expr** 循环控制表达式。WHILE 指令必须以关键字 **END_WHILE** 做为循环控制结束，否则会语法错误

WHILE 格式：

```
WHILE expr DO
    ...
END_WHILE
```

LOOP

循环次数控制指令，LOOP 指令必须以关键字 **END_LOOP** 做为循环控制结束，否则会语法错误

LOOP 格式：

(1) 指定常值循环次数

LOOP n DO

...

END_LOOP

n 为正整数，指定循环次数

(2) 通过\$D/D/LD 变量指定循环次数

D0000=10

LOOP D0000 DO

...

END_LOOP

(3) 无限循环

LOOP DO

...

END_LOOP

LOOP 指令内部由\$D7900~\$D7999 作为循环变量，主控制器和 8 个并行控制器各占 10 个循环变量，因此每个控制器内部最大可嵌套层数为 10 层。