
华中科技大学

HRT-6 机器人示教器

用户使用说明书

华中科技大学国家数控系统工程技术研究中心

NCNER

2016-3-14

目录

1	HRT-6 机器人示教器简介	6
1.1	HRT 示教器按键与接口	6
1.2	示教器软件安装	7
1.2.1	安装步骤	7
1.3	HRT 示教器界面	9
1.3.1	初始化界面	9
1.3.2	状态栏	9
2	手动操作	11
2.1	概述	11
2.1.1	界面说明	11
2.1.2	手动操作说明	12
2.2	坐标模式	12
2.3	工具坐标系	13
2.4	工件坐标系	14
2.5	增量设置	15
2.6	倍率设置	15
2.7	回参考点	15
3	示教界面	16
3.1	概述	16
3.2	示教界面菜单	16
3.3	新建程序	17
3.4	打开程序	19
3.5	程序修改	20
3.5.1	END 行短按	20
3.5.2	行内编辑（短按）	21
3.5.3	行编辑（长按）	24
3.5.4	指令类型说明	31
3.5.5	程序检查	36
3.5.6	程序保存	36
4	自动运行	37
4.1	概述	37
4.2	加载程序	38
4.3	显示模式	38
4.4	循环模式	38
4.5	指定行运行	38
5	寄存器	41
5.1	概述	41

5.2	R 寄存器.....	41
5.3	位置寄存器.....	42
5.4	查找.....	43
6	IO 信号.....	44
6.1	概述.....	44
6.2	信号显示和设置.....	44
7	设置界面	46
7.1	概述.....	46
7.1.1	初始密码.....	46
7.1.2	界面简介.....	47
7.2	系统参数.....	48
7.3	组参数.....	48
7.4	轴参数.....	49
7.5	机械参数.....	50
7.6	工具坐标系设定.....	51
7.6.1	三点标定.....	53
7.6.2	六点标定.....	53
7.7	工件坐标系设定.....	55
8	生产管理	58
8.1	概述.....	58
8.2	报警历史.....	58
8.3	程序管理.....	58
8.4	操作记录.....	59
8.5	关于设备.....	59
8.6	修改密码.....	59
9	码垛.....	60
9.1	概述.....	60
9.1.1	基本概念.....	60
9.1.2	准备工作.....	60
9.2	码垛工艺设置.....	61
9.2.1	码垛工艺选择.....	61
9.2.2	基本参数设置.....	61
9.2.3	参考工件设置.....	67
9.2.4	排样设置.....	69
9.2.5	路径设置.....	76
9.2.6	干涉检测设置.....	78
11	编程篇	80
1	运动指令	81

1.1	动作类型.....	81
1.1.1	关节定位.....	81
1.1.2	直线运动 L.....	82
1.1.3	圆弧运动 C.....	83
1.2	位置数据.....	83
1.2.1	直角坐标系.....	83
1.2.2	关节坐标.....	85
1.2.3	位置变量和位置寄存器.....	85
1.3	进给速度.....	86
1.4	定位路径.....	86
1.5	附加动作指令.....	88
1.5.1	加速倍率 (ACC).....	88
1.5.2	增量指令 (INC).....	89
2	寄存器指令.....	90
2.1	寄存器 (R) 指令.....	90
2.2	位置寄存器 (PR) 指令.....	92
2.3	位置寄存器轴指令.....	93
3	I/O 指令.....	96
3.1	数字输入/输出 (DI/DO) 指令.....	96
3.2	模拟输入/输出 (AI/AO) 指令.....	98
4	条件指令.....	148
4.1	寄存器条件比较指令.....	148
4.2	输入输出条件比较指令.....	149
1.1	复合条件的使用.....	150
2	等待指令.....	152
2.1	指定时间的等待指令.....	152
2.2	条件等待指令.....	152
2.2.1	寄存器条件等待指令.....	153
2.2.2	输入/输出条件等待指令.....	153
3	流程控制指令.....	155
3.1	标签指令.....	155
3.2	程序结束指令.....	155
3.3	无条件跳转指令.....	156
3.4	子程序调用指令.....	156
4	码垛指令.....	0
4.1	码垛开始指令.....	0
4.2	码垛动作指令.....	0
4.3	码垛结束指令.....	1

4.4	码垛寄存器指令.....	1
4.5	码垛程序示例.....	2
5	其他指令	2
5.1	坐标系指令.....	2
5.1.1	坐标系设置指令.....	3
5.1.2	坐标系选择指令.....	4
5.2	用户报警指令.....	5
5.3	倍率指令.....	6
5.4	注释指令.....	6
5.5	信息指令.....	7
5.6	预读指令.....	7
5	附录：程序报警定义	9

1 HRT-6机器人示教器简介



图 1-1 HRT-6 机器人示教器

如图 1-1所示，HRT-6机器人示教器是为近旁在线操作机器人而设计的一款设备。它符合人机工程学设计，轻便灵活，配备 LCD 彩色触摸屏幕与简洁的按钮，为用户提供友好的操作体验以及快速准确的机器人运动控制功能。

1.1 HRT 示教器按键与接口

如图 1-2所示为机器人示教器的整体布局。



图 1-2 示教器整体布局图

HTP 机器人示教器按键可以分为轴控制键、倍率调节键、增量调节键、主页键、程序控制键；示教器背面的 MicroUSB 接口为调试接口，用于与 PC 机连接。另外的一个 USB 主口用于连接 U 盘。


编号	功能
1	暂未定义
2	急停按钮。用于紧急停机。
3	轴控制按键。用于手动移动机器人。
4	倍率调节按钮。用于调节机器人运动快慢。
5	增量调节按钮。用于调节增量模式下机器人运动快慢。
6	主页键。用于调出窗口切换菜单。
7	暂停键。用于暂停运行程序。
8	停止键。用于停止运行程序。
9	暂未定义。
10	启动键。用于启动运行程序。
11	暂未定义。

表 1-1 示教器按键功能说明

1.2 示教器软件安装

1.2.1 安装步骤

1. 将 MacHand.apk 文件放入 U 盘，将 U 盘插入示教器背部的 USB 接口。

2. 打开 ES 文件浏览器 ，在左侧本地菜单中点击“udisk”，即可浏览 U 盘中的文件。找到 MacHand.apk，点击安装。

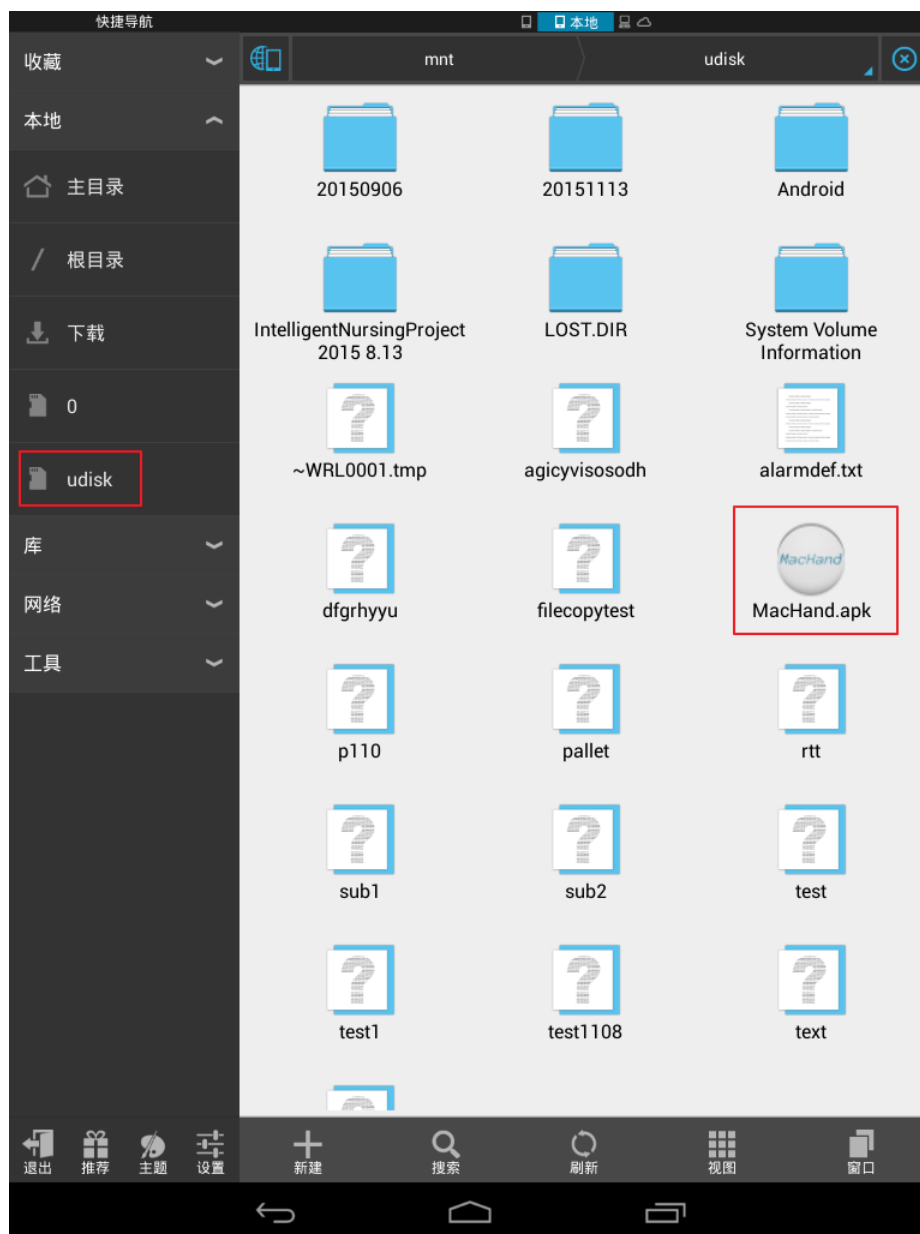


图 1-3 软件安装

3. 由于 Android 设备的安全性设置，点击安装包之后可能会弹出如下图所示的对话框。此时点击对话框中的设置，将安全设置中的“未知来源”勾选即可。

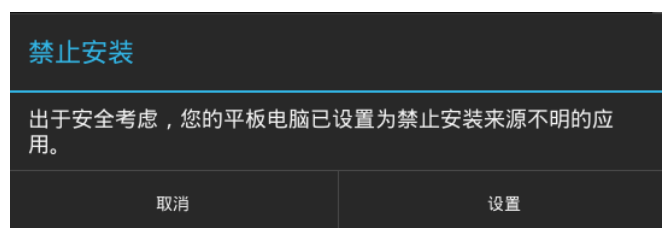


图 1-4 禁止安装提示对话框

4. 此时回到 U 盘文件浏览，点击 MacHand.apk 文件，点击安装即可。

1.3 HRT 示教器界面

1.3.1 初始化界面



图 1-5 初始化页面图

HRT-6工业机器人控制系统上电后，示教器载入应用程序后，会出现上述界面，如果连接正确连接，示教器软件会进入操作界面。

1.3.2 状态栏

状态栏用于提示网络状态和当前控制器状态。







图 1-6 状态栏

页面名称：显示当前处于哪个页面。

网络状态：●绿色表示网络正常，●红色表示网络不通。

报警状态：●绿色表示当前无报警，●红色表示机器人控制器当前发生报警。在发生报警状态下，点击“报警●”区域，会弹出对话框，提示当前报警信息。

运行模式：包含两种状态：手动  和自动 ，可以通过电柜上的模式选择开关进行切换。手动模式下，可通过手动操作键及滚轮配合操作机器人进行动作，自动模式下，可运行示教程序，运行过程可通过程序控制键来控制。

程序状态：分为程序运行、停止和暂停三种状态，在状态栏分别显示为  



2 手动操作

2.1 概述

2.1.1 界面说明

如图 2-1所示，手动操作界面是 HRT-6工业机器人示教器软件的初始界面，手动运行界面主要用于显示和设置机器人手动控制相关的操作。



图 2-1 手动操作界面

手动操作界面上方为坐标显示区，下方为参数设置区，右侧为轴控制显示区。

- (1) 坐标显示区：显示当前轴号及相应坐标值。
- (2) 参数设置区：用于显示当前的状态信息。点击中间按钮，会弹出对应的设置对话框。按左侧与右侧按钮，也可以直接切换相应选项。
- (3) 轴控制显示区：显示当前可移动的轴号及倍率和增量值。

2.1.2 手动操作说明

手动界面最主要功能是手动控制机器人动作，以下对如何进行手动操作机器人进行说明。

轻按（即停留在中间键位）背部黄色安全按钮，轴控制区的轴号会显示为绿色，表示这些轴现在处于可移动状态，用相应的轴控制按键即可控制相应轴的运动。

增量显示为无时，表示机器人的手动模式是连续运动，即按住轴控制按钮，机器人就运动，松开轴控制按钮，机器人停止运动。

增量模式为 x1、x10、x100时，表示机器人的手动模式是增量运动，即按一次轴控制按钮，机器人移动一定角度或距离之后即停下来。

2.2 坐标模式

坐标模式选择的操作方式与上述机械单元选择的操作方式一致。坐标模式有5种，分别为关节坐标、基坐标、工具坐标、工件坐标、外部轴。

修改坐标模式的操作步骤如下：

- (1) 手动操作界面下，点击图中的“关节坐标”控件，就会弹出坐标模式选择对话框。



图 2-2 坐标模式选择对话框

- (2) 在对话框内选中相应坐标模式，焦点停留在选中项时，点击对话框中的确定按钮，页面回到手动操作界面，坐标模式修改成功。如果取消，页面回到手动操作界面，坐标模式不修改。
- (3) 手动操作界面下，按下“关节坐标”的左侧或右侧按钮，即可直接切换坐标模式，切换顺序与坐标模式选择对话框中列表顺序一致。

关节坐标模式下，显示模式默认为关节坐标，其他坐标模式下，显示模式默认为直角坐标。关节坐标使用的坐标为 J1、J2、J3、J4、J5、J6，直角坐标使用的坐标是 X、Y、Z、A、B、C，所以位置显示及运动模式会有相应的变化。

位置							
J1	156.674	deg	J4	-3.798	deg	E1	0.000 mm
J2	78.038	deg	J5	-86.602	deg	E2	0.000 mm
J3	-0.238	deg	J6	-24.380	deg	E3	0.000 mm

图 2-3 关节坐标模式下坐标显示

位置							
X	-845.869	mm	A	0.107	deg	E1	0.000 mm
Y	364.744	mm	B	170.429	deg	E2	0.000 mm
Z	549.439	mm	C	-1.464	deg	E3	0.000 mm

图 2-4 其他坐标模式下的坐标显示

2.3 工具坐标系

本机器人系统的工具坐标系总共有16个，从工具0到工具15。设置当前工具坐标的操作步骤如下：

- (1) 手动操作界面下，点击工具坐标系中的“工具0”控件，就会如图 2-5所示弹出工具坐标系选择对话框。

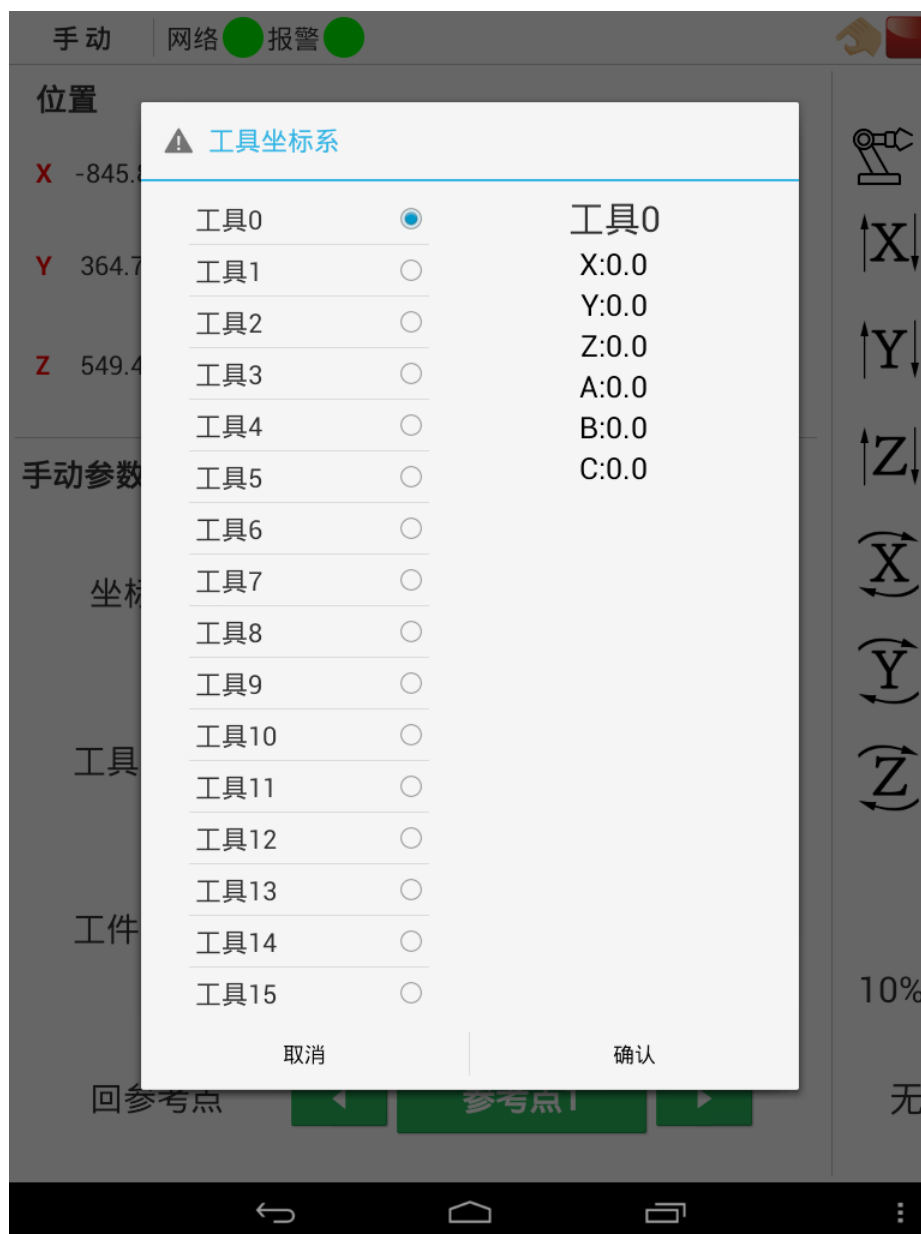


图 2-5 工具坐标系选择对话框

- (2) 在对话框内选中相应工具坐标系，右侧会显示该工具坐标系相应的值，点击对话框中的确定按钮，工具坐标系修改成功。如果取消，工具坐标系不修改。
- (3) 手动操作界面下，按下“工具0”的左侧或右侧按钮，即可直接切换工具坐标系，切换顺序与工具坐标系选择对话框中列表顺序一致。

2.4 工件坐标系

本机器人系统的工件坐标系总共有16个，从工件0到工件15。设置当前工件坐标的操作步骤与工具坐标系的选择一致，此处不再赘述。

2.5 增量设置

增量表示机器人一次增量运动的移动距离，本机器人系统可设置的增量为：无、×1、×10、×100。当系统设置为无增量模式时为连续运动模式。例如，×1：在工具坐标系（工件或基坐标系）平动模式下，机器人 TCP 移动0.01mm；若为关节模式，机器人关节轴旋转0.01deg。增量值通过增量设置按键来设置。

2.6 倍率设置

在增量选择为“无”时，为连续手动模式，倍率生效。倍率值通过倍率设置按键设置。

2.7 回参考点

回参考点的操作步骤如下：

- (1) 手动操作界面下，点击回参考点对应的“回参考点1”控件，点击其左右侧按钮可切换需要回的参考点，可以选择的有参考点1和参考点2。
- (2) 点击回参考点对应的“回参考点1”控件，即可进入如图 2-6所示的回参考点对话框。



图 2-6 回参考点对话框

- (3) 按住 J1按钮，J1关节开始转动，松开 J1按钮，J1关节停止转动。一直接住 J1按钮，J1关节会一直转动直至到达参考点停止转动。点击全部回零按钮，各个关节都以自己的速率回零。
- (4) 观察上侧的坐标区域可以知道是否回到参考点。点击【确定】或者【取消】关闭对话框。

3 示教界面

本章介绍一些示教界面中指令的基本操作方法，详细的编程过程请参照后面的指令编程章节。

3.1 概述

程序的基本信息包括：程序名、程序注释、子类型、组标志、写保护、程序指令和程序结束标志。

- (1) 程序名：用以识别存入控制器内存中的程序，在同一个目录下不能包含两个或更多拥有相同程序名的程序。程序名长度不超过8个字符，由字母、数字、下划线(_)组成。
- (2) 程序注释：程序注释连同程序名一起来描述选择界面上显示的附加信息。与程序名相关（非程序内容注释），最长16个字符，由字母、数字及符号（_、@、*）组成。新建程序之后可在程序选择中修改程序注释。
- (3) 子类型：用于设置程序文件的类型。目前本系统只支持机器人程序这一类型。
- (4) 组标志：设置程序操作的动作组，必须在程序执行前设置。目前本系统只有一个操作组，默认的操作组是组1（1，*，*，*，*）。
- (5) 写保护：指定该程序可否被修改。若设置为“是”，程序名、注释、子类型、组标志等不可修改。若此项设置为“否”，程序信息可修改。当程序创建且操作确定后，可将此项设置为“是”来保护程序，防止他人或自己误修改。
- (6) 程序指令：包括运动指令、寄存器指令等示教中涉及的所有指令。
- (7) 程序结束标志：程序结束标志（END）自动在程序的最后一条指令的下一行显示。只要有新的指令添加到程序中，程序结束标志会在屏幕上向下移动，所以，它总在最后一行。当执行完最后一条指令后，程序执行到程序结束标志时，就会自动返回到程序的第一行并终止。

3.2 示教界面菜单

在示教界面按下【菜单】键，会弹出功能菜单，如下图所示：



图 3-1 示教界面功能菜单

示教界面的功能菜单包含“新建程序”、“打开程序”、“程序检查”、“保存”、“另存为”、“详情”。

3.3 新建程序

在功能菜单中选择“新建程序”，即弹出如下图所示的对话框

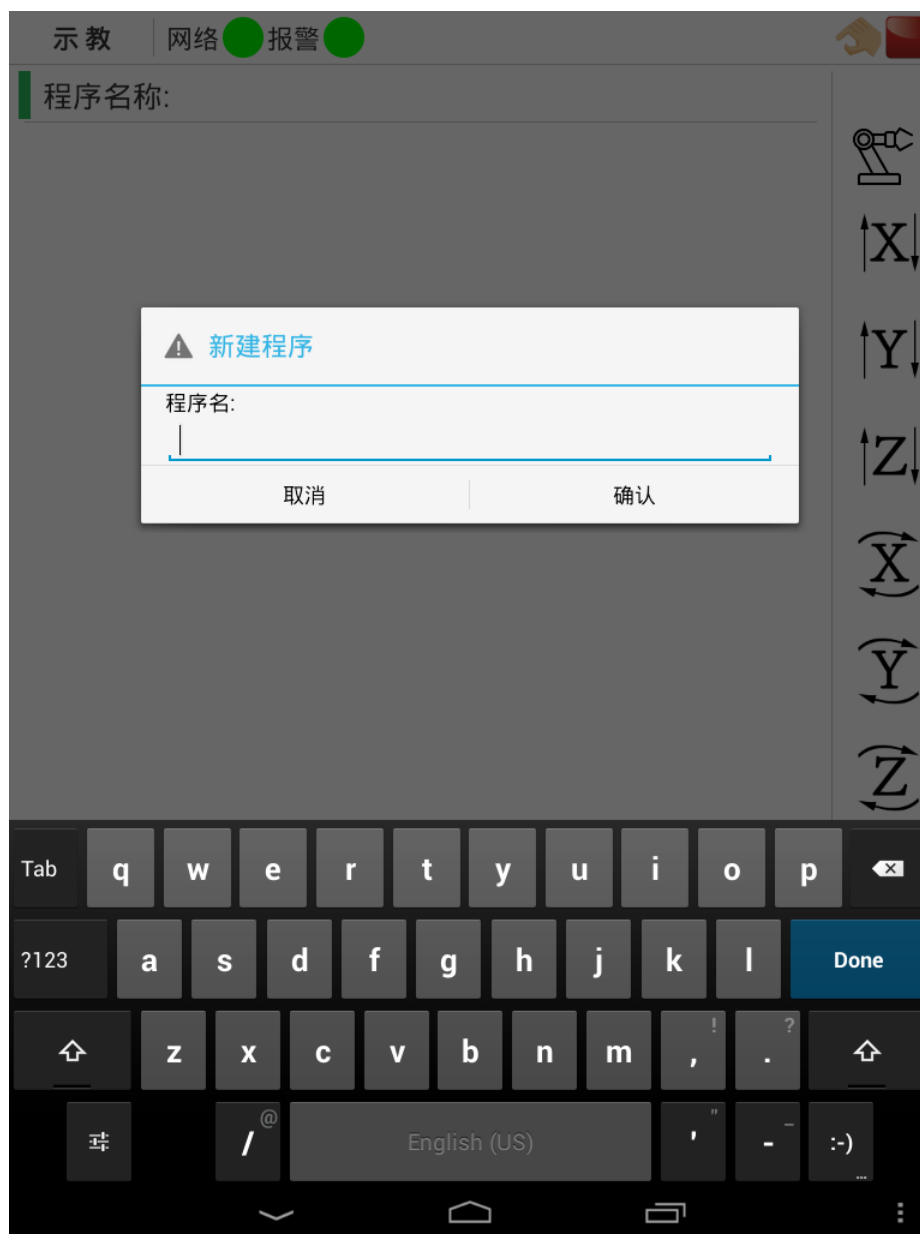


图 3-2 新建程序对话框

输入程序名，点击确定后，如果没有与已有程序名冲突，则会回到示教界面，新建的程序会自带结束行“END”。

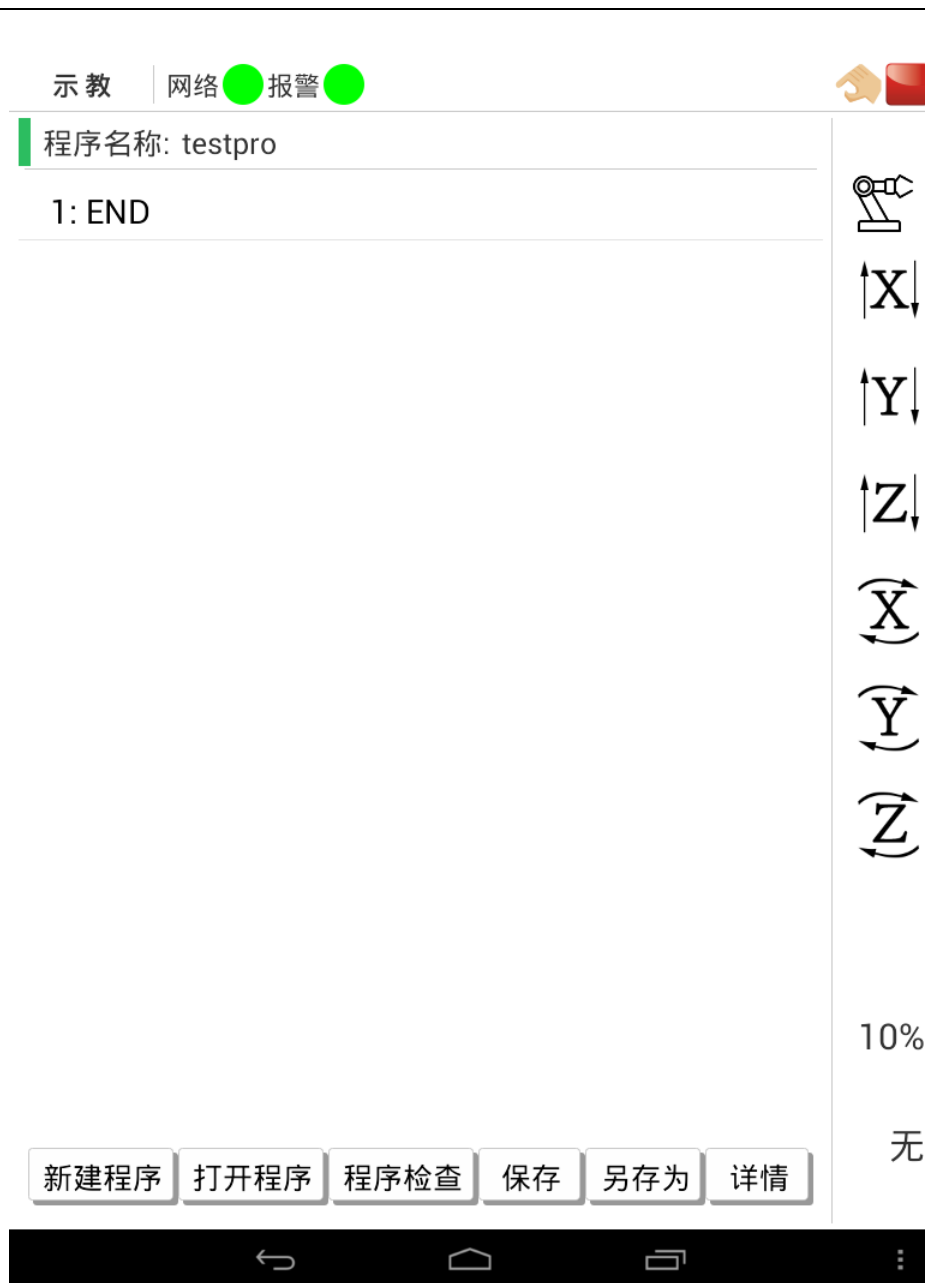


图 3-3 新建程序的示教界面

3.4 打开程序

在功能菜单中选择“打开程序”，可显示程序文件列表，选择一个现有的程序文件，点击“确认”按钮，即可打开选中的程序文件。如图 3-4所示：

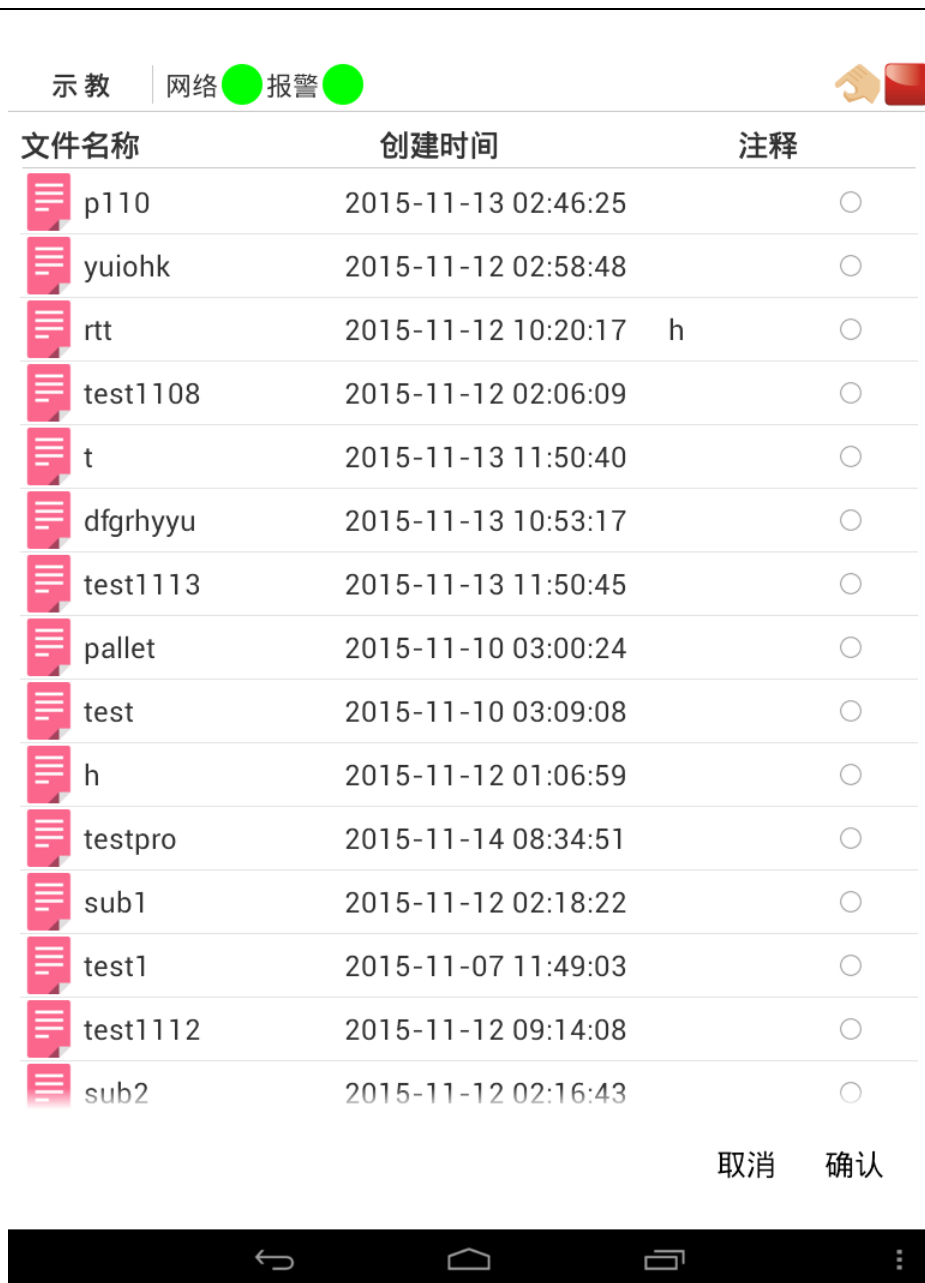


图 3-4 打开程序界面

3.5 程序修改

示教主要提供程序修改编辑功能，本示教器提供两种操作方式即：短按（即点击）和长按。

3.5.1 END 行短按

在新建程序完成后，会生成包含 END 行的程序。编辑已有程序时，如果光标停在 END 行，点击 END 行，会弹出如下图所示的指令选择菜单，编辑指令完成后，该条指令会插入到 END 行的上一行。

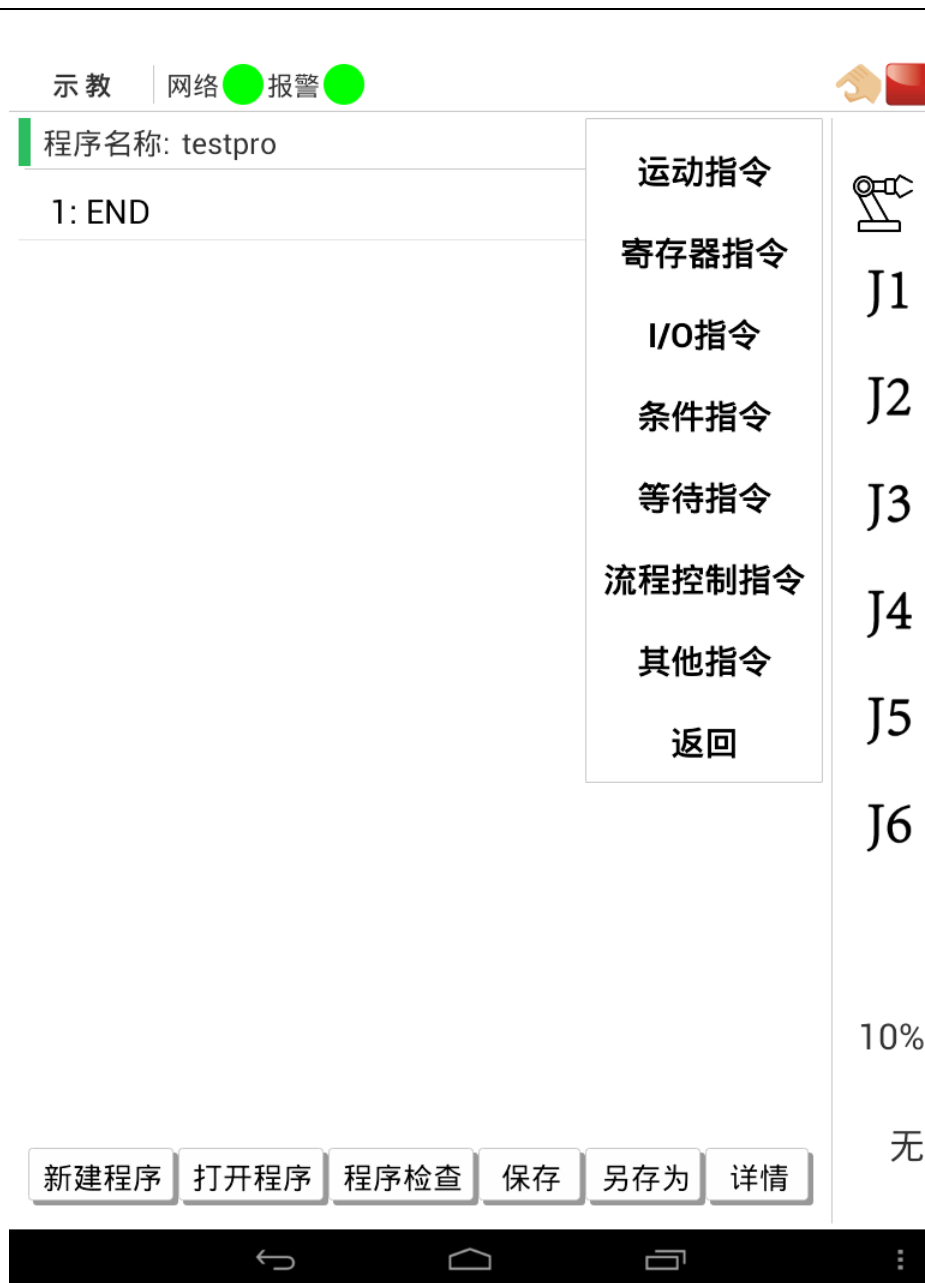


图 3-5 点击 END 行直接在上行插入指令

3.5.2 行内编辑（短按）

点击任一行的程序语句（最后一行“END”除外），可对该行程序语句的内容进行编辑。

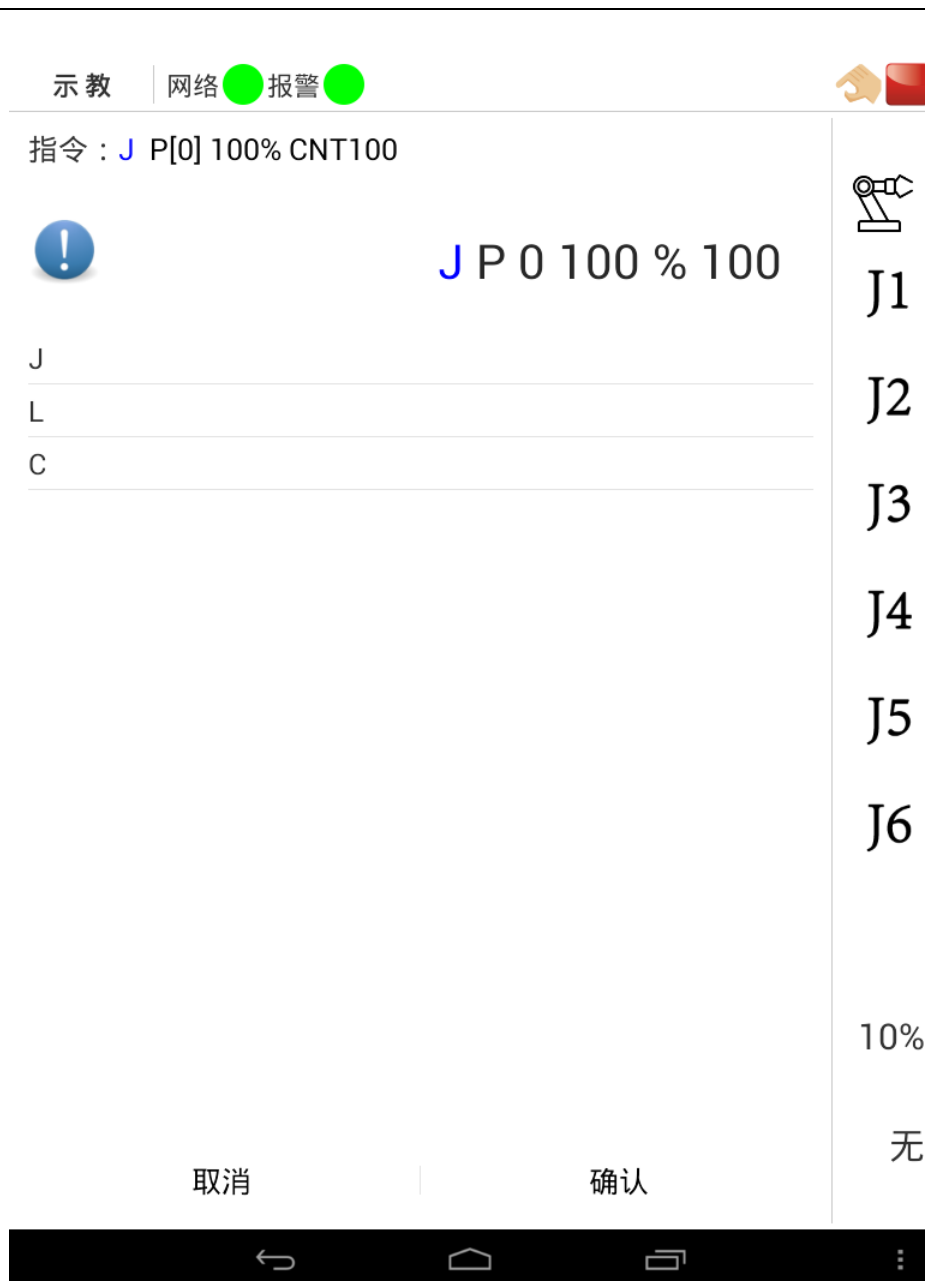


图 3-6 行内编辑对话框

在行内编辑对话框，左右键切换需要编辑的指令元素，上下键选择修改指令元素值，如上图中，光标指向 J 指令元素，下方的指令元素选择列表，移动光标至 L，按【确认】键则将 J 指令改为 L 指令。

运动指令默认指令值介绍：

- (1) 初始时，J 指令默认值为：J P [n] 100% CNT100，(n 为点 P 的标号，自增，下同)。L 指令的默认值为：L P [n] 100mm/sec FINE，C 指令的默认值为：C P [n] 100mm/sec FINE，运动指令默认采用 J 指令；
- (2) 如果编辑过程中出现过运动指令，则相应的运动指令默认值改变为上一条指令值。默认运动指令类型也变为上一条运动指令类型，就是说如果上一条运动指令是 L 指令，则下次插入运动指令时默认是 L 指令。例如：编辑程序过程中出

现了 J 指令，JP[n] 50% FINE，则下次选用 J 指令时，默认值就是 JP[n] 50% FINE。L 指令与 C 指令类似。

(3) 其他指令类型暂时只有初始默认值，默认值并不随编程过程动态变化。

下面以 J 指令为例，说明指令的行内编辑过程。

(1) 在左侧的程序显示区域点击“1:JP[0] 100% CNT100”行，则会弹出该行的编辑对话框，如下图所示：

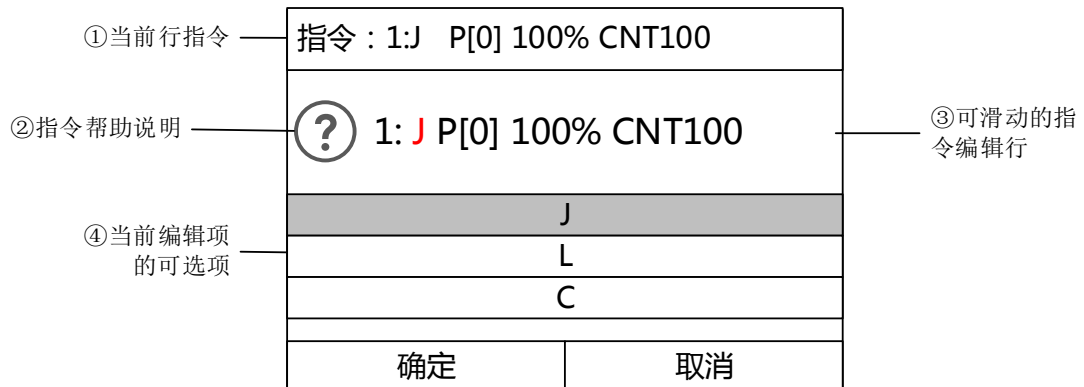


图 3-7 程序行编辑对话框

(2) 在该对话框中，图中红色的“J”表示当前编辑项是“J”，可使用触屏方式将光标移至需要编辑的指令元素，例如在屏幕上点击“100%”，则光标会定位至指令元素“100%”，下方的可选项也会做出相应改变。在该对话框，使用【左右导航】键可以切换光标，即切换需要编辑的指令元素，使用【上下导航】键可以直接改变该指令元素的选择项。

(3) 指令帮助说明：点击帮助图标(?)后，显示指令类型说明，运动指令时，弹出帮助对话框。



图 3-8 示教帮助说明

(4) 本次编辑的确认和取消：点击“确认”后，即可将本次编辑好的指令行替换旧的编辑行。或点击“取消”按钮，取消当前编辑。

(5) 编辑过程：

a) 选择 L 指令，这时指令编辑行会变成 J 指令的默认值：LP[1] 100mm/sec FINE，光标自动右移至下一个指令元素，即 P 点的下标值；

- b) 光标移至 P 点的下标值时，当前编辑项的可选项区域也会做出相应变化，如下图所示：

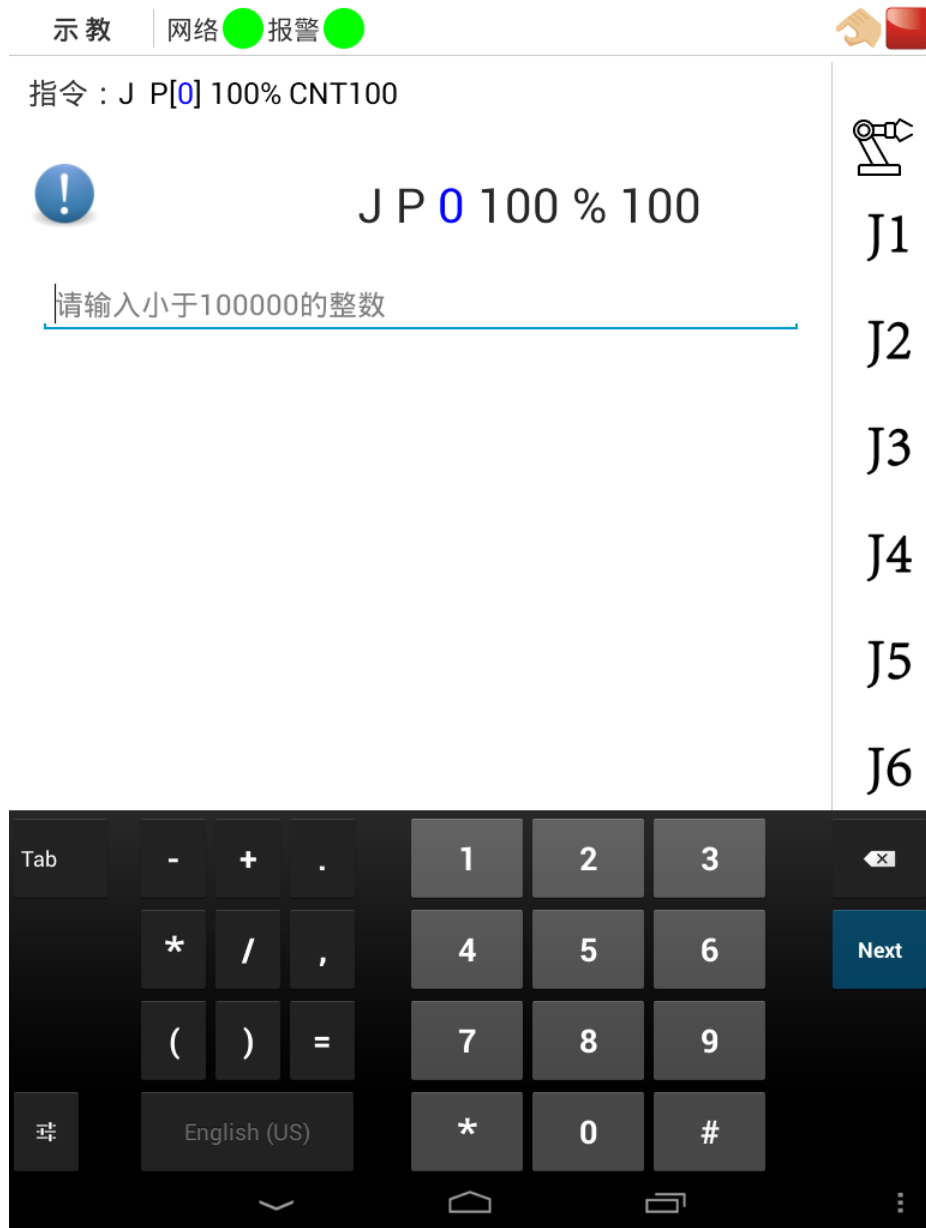


图 3-9 P 点下标值设置

这时光标是在下方编辑框内的，将光标重新移至区域③指令编辑行，可使用【上导航】键或者点击其他指令元素。光标移到其他控件上时，前面控件中的更改会显示出来。

- c) 点击确定，更改生效；点击取消，放弃更改。

3.5.3 行编辑（长按）

长按任一行的程序语句，可对该行程序语句做整体操作，包括删除、复制、剪切、粘贴、修改位置、上行插入、下行插入等，如图 3-10所示：

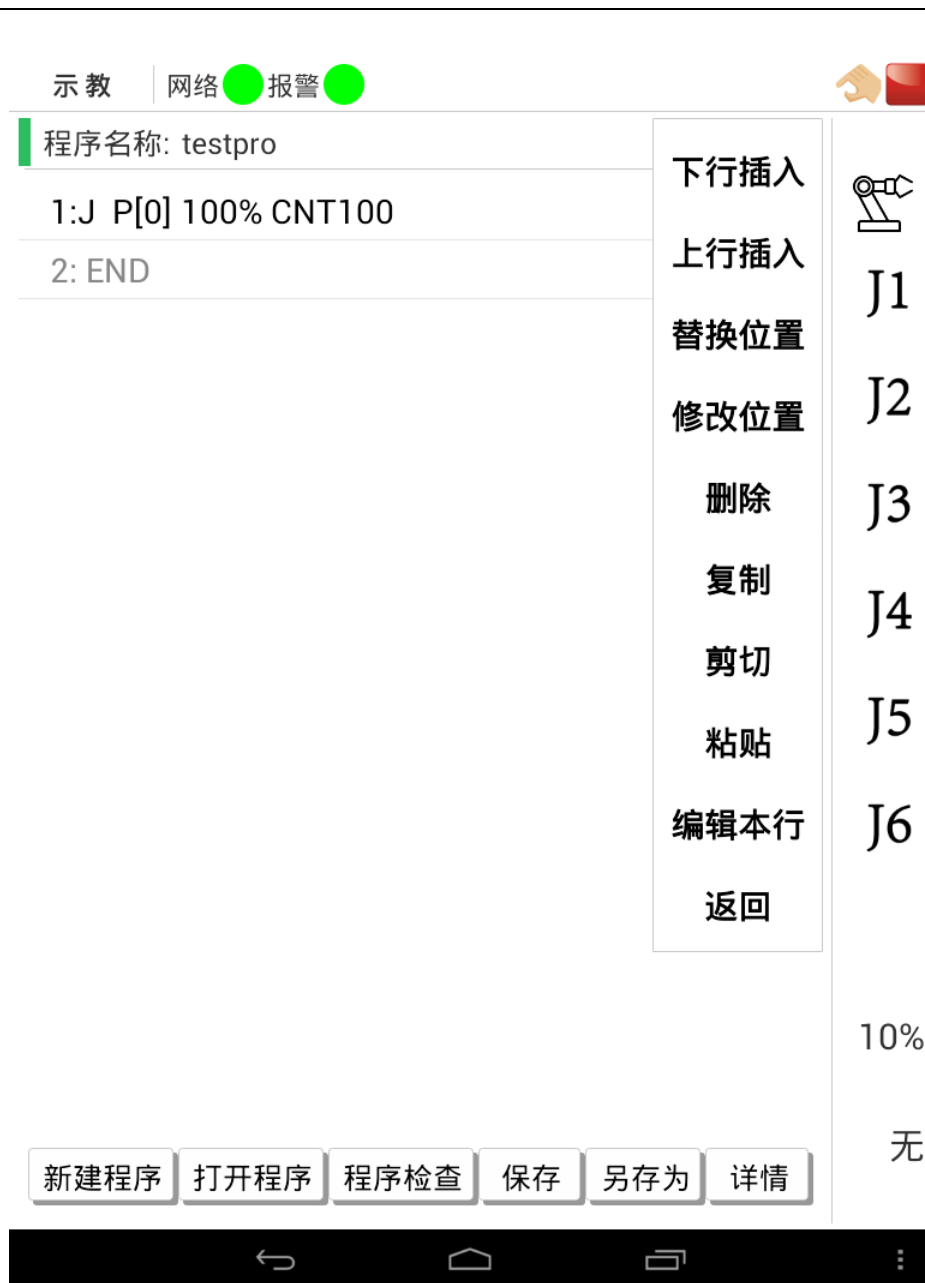


图 3-10 行编辑长按菜单

下面对长按菜单逐项进行说明：

(1) 插入（上行或下行）

根据指令行插入的位置，插入操作分为上行插入和下行插入。

- **上行插入**：是在当前所选择指令的前一行插入指令；
- **下行插入**：是在当前所选择指令的后一行插入指令。

下面以上行插入为例，介绍插入指令行的操作方法。

- a) 选择“上行插入”选项，弹出如下图所示指令选择框：

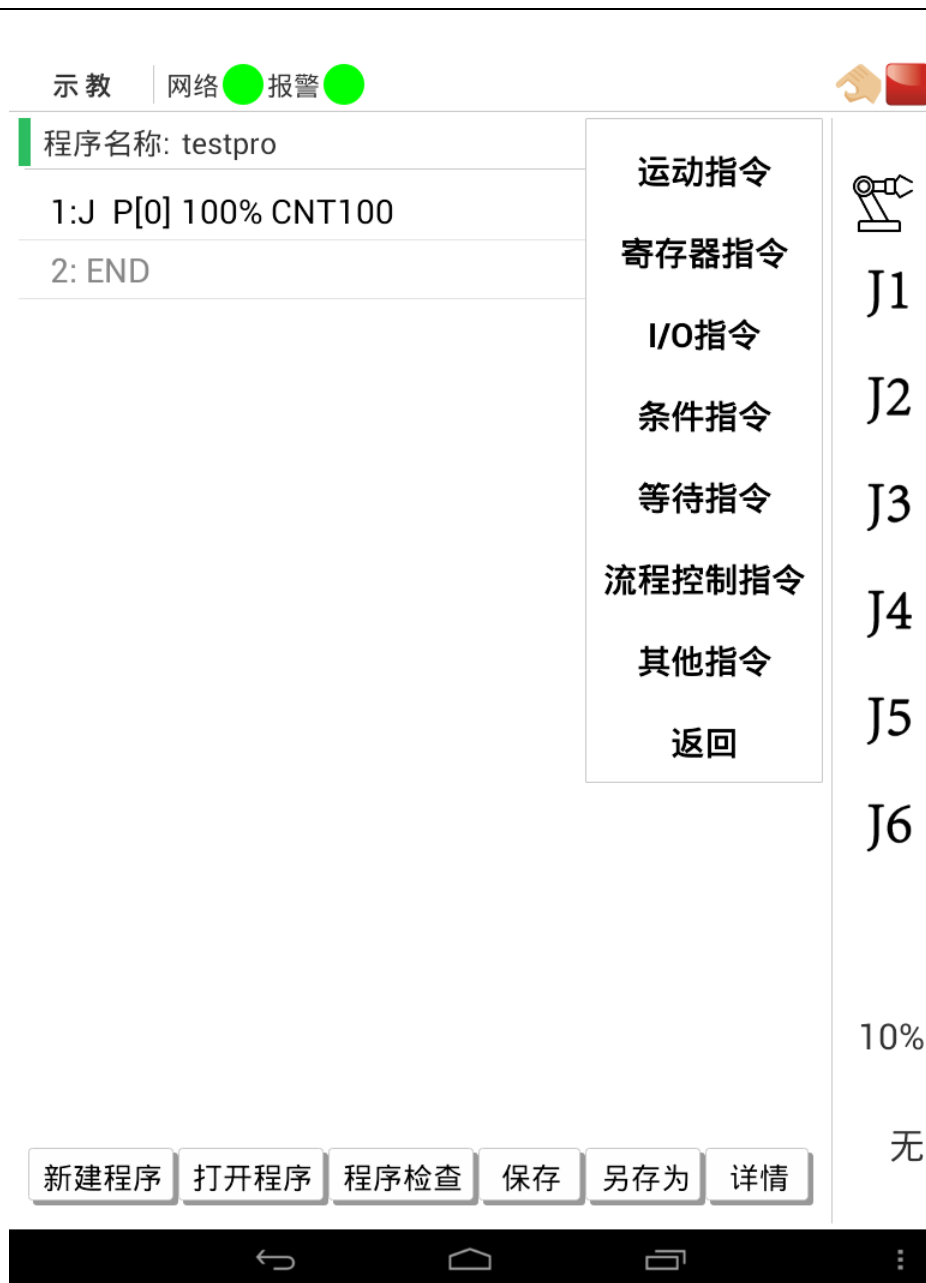


图 3-11 指令类型选择界面

b) 选择指令类型，如选择“I/O 指令”，界面切换至 IO 指令的编辑对话框

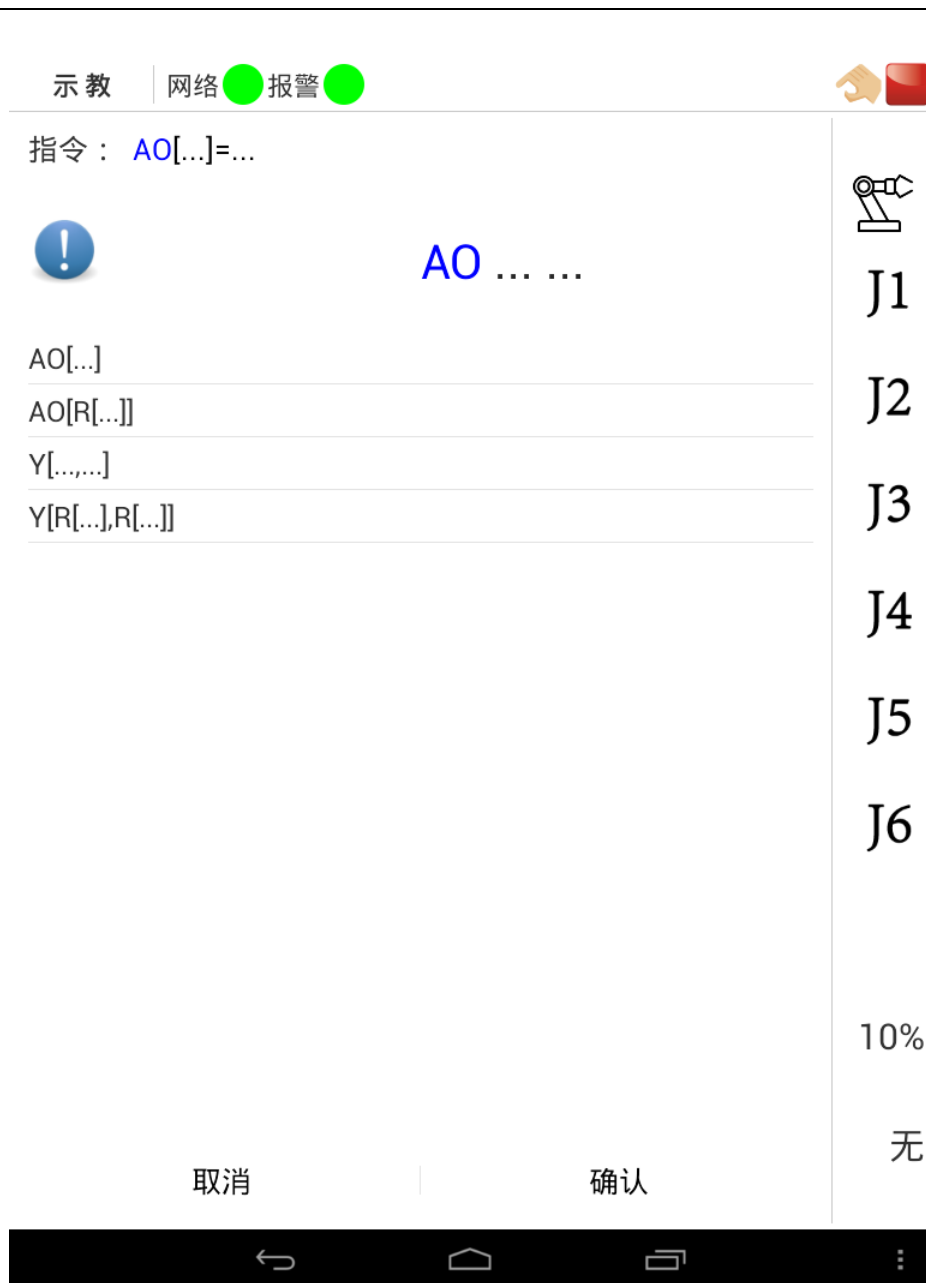


图 3-12 选择指令

c) 该编辑界面即为行内编辑的界面，编辑完成后点击确认即可完成上行插入操作。

(2) 替换位置

若当前行含有位置变量 **P** 或者位置寄存器 **PR**，且位置号都是直接寻址的（即为“**P[常量]**”或“**PR[常量]**”），长按当前行时，替换位置菜单颜色变为可操作，即可以对当前行的位置信息进行替换。

将当前指令中的位置变量替换为机器人当前位置值，选中后会弹出确认对话框。如图 3-13所示



图 3-13 替换位置确认对话框

(3) 修改位置

若当前行含有位置变量 **P** 或者位置寄存器 **PR**，且位置号都是直接寻址的（即为“**P[常量]**”或“**PR[常量]**”），长按当前行时，修改位置菜单颜色变为可操作，即可以对当前行的位置信息进行查看或修改。

如本例位置号为常量“0”，即可对位置信息进行查看和修改，在弹出的操作菜单中选择“修改位置”，弹出如所示对话框：

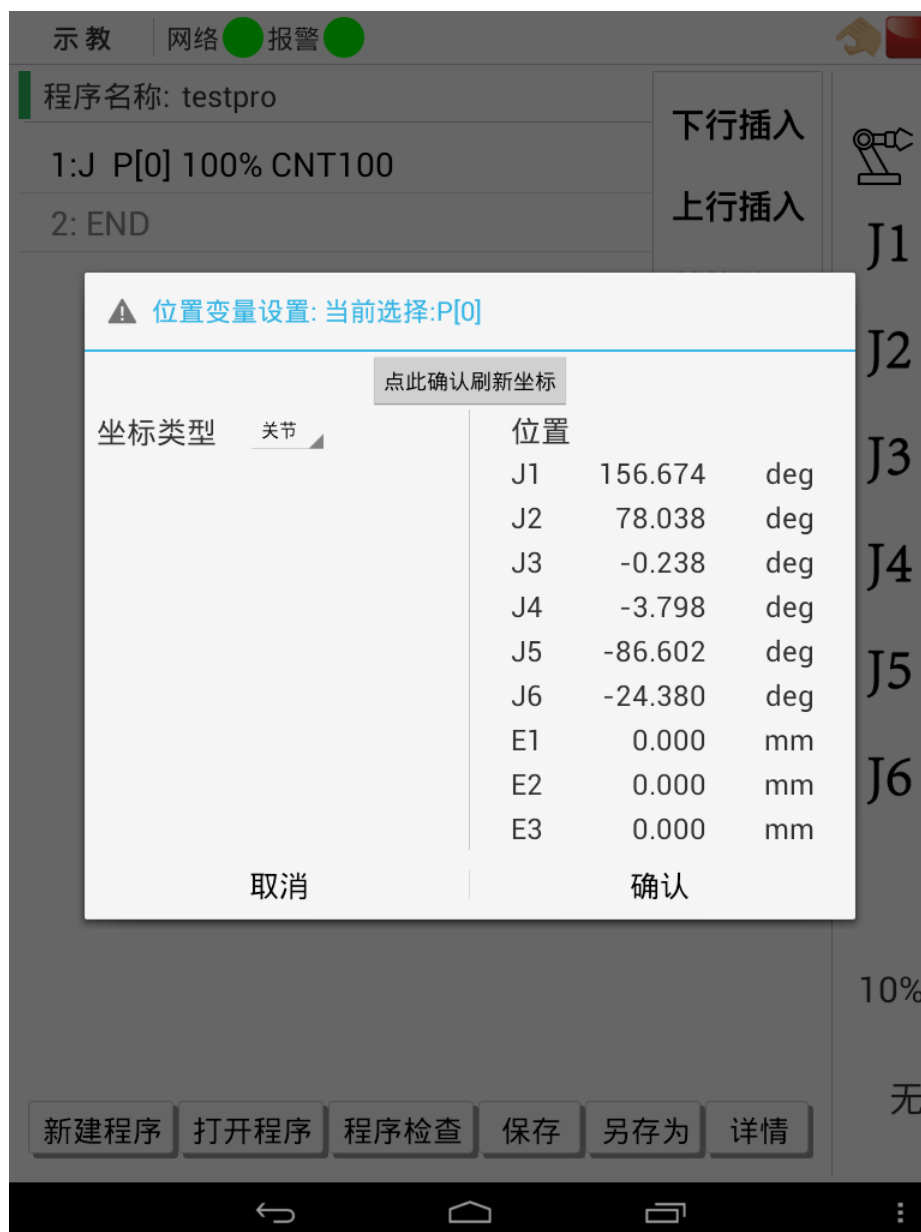


图 3-14 修改位置对话框

- (a) 直接修改坐标值：点击右侧位置显示区域的坐标值，在弹出的对话框中输入想要修改的值，确认即可。
- (b) 刷新为当前坐标值：点击“点击确认刷新坐标”，即可将坐标值刷新为当前机器人的位置值。刷新后，该按钮会消失，如图 3-15所示

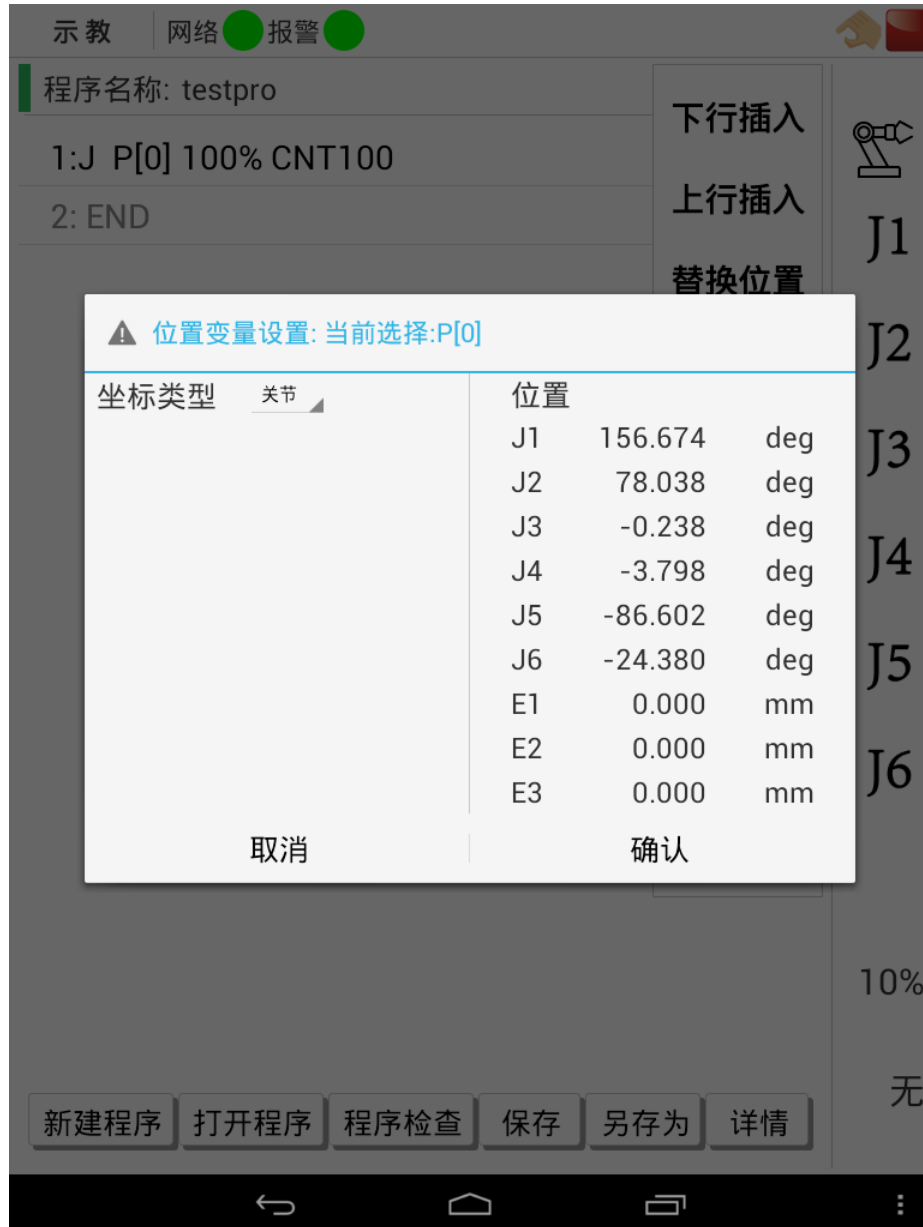


图 3-15 刷新坐标值

- (c) 更改坐标类型：刷新坐标值后，可以更改坐标类型，更改之后坐标值会刷新为当前坐标值。如图 3-16所示

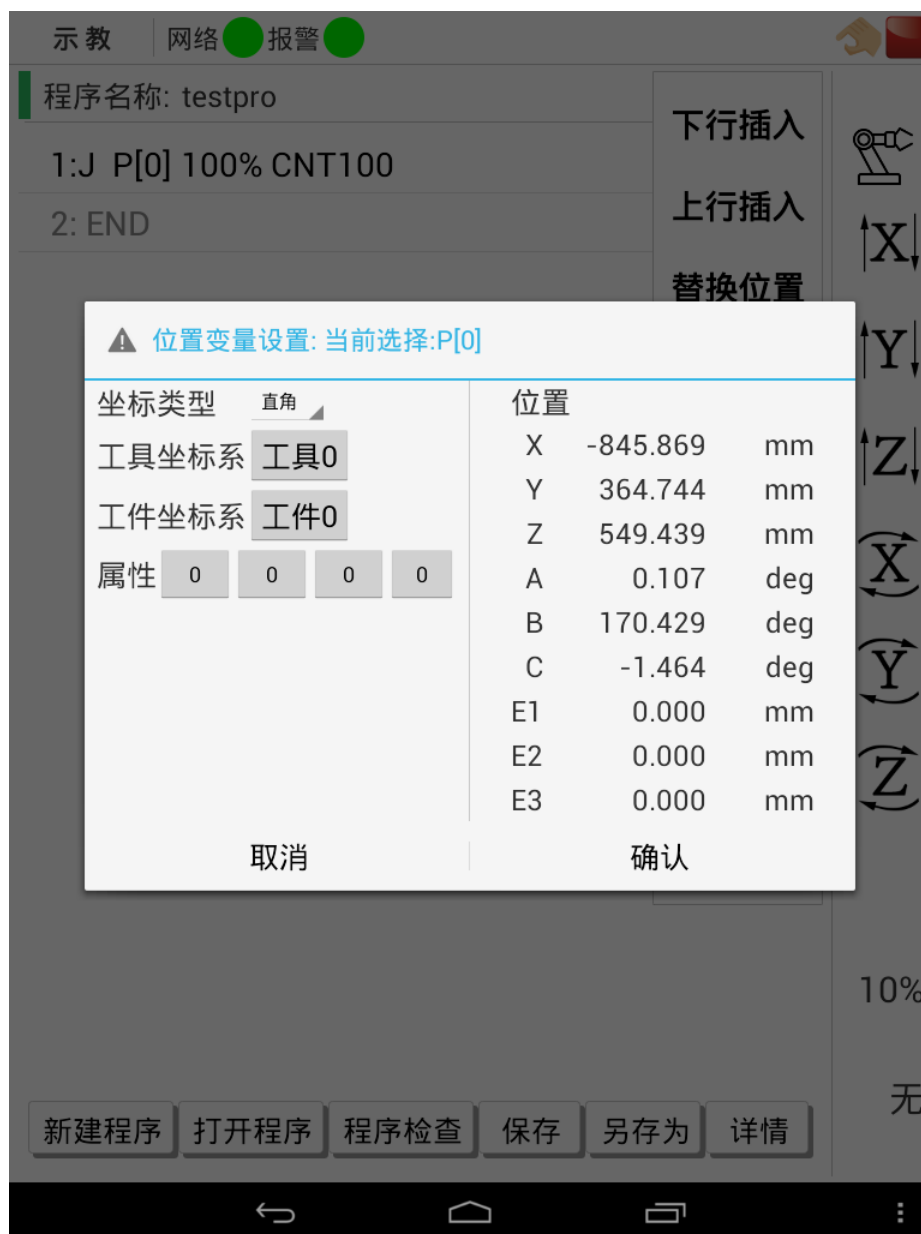


图 3-16 更改坐标类型

- (d) 其他参数修改：通过点击按钮弹出对话框进行修改。
- (e) 点击确认，则所有更改生效，点击取消，则放弃所有更改。
- (4) 编辑本行
编辑当前行，相当于短按。
- (5) 删除
删除当前选择的行。
- (6) 复制
复制当前行的内容到剪切板
- (7) 剪切
复制当前行的内容到剪切板，并删除当前行。

(8) 粘贴

当前行后移，将粘贴板上的行信息粘贴为当前行。

3.5.4 指令类型说明

本小节简单介绍不同类型的指令，详细编程方法请参看编程篇。

(1) 运动指令

运动指令类型包括三种：关节定位（J）、直线定位（L）、圆弧定位（C）。



图 3-17 运动指令

(2) 寄存器指令

寄存器指令主要是在寄存器上完成算术运算。根据运算表达式左值的类型，可以将寄存器指令分为：R 寄存器、位置寄存器 PR[i]指令及位置寄存器轴指令 PR[i,j]。

寄存器指令支持的运算操作有：+、-、*、/、MOD（取商的余数，即小数点后的值）、DIV（取商的整数）。如图3-15所示，但对于位置寄存器只支持+、-两种运算操作。寄存器指令的详细编程说明，请参看后面章节的编程说明。



图 3-18 寄存器指令

寄存器指令支持多项式运算，但使用时请注意：单独一行最多5个运算符。

本系统支持+、-、*、/、MOD、DIV 在一行中混合使用，MOD、DIV 属于同一级别，*、/属于同一级别，+、-属于同一级别，优先级别划分为：

- MOD、DIV优先级高于*、/；
- *、/优先级高于+、-；
- 同一级别下，优先级从左到右，即左边高于右边；

在实际的示教编程中，如果想要取消后面的多项式，如：

示例1： R[2]=R[3]-R[4]+R[5]-R[6]

示例1中，若要取消“+R[5]-R[6]”，选中“+”运算，在运算符（如+、-、*、/都是运算符）的输入选择窗口中选择“<cr>”，即可清除选中运算符后面的所有表达式即“+R[5]-R[6]”。

注：对于使用工件（或工具）坐标系作为**右值**的位置寄存器 PR 指令，只能进行直接赋值运算，不能进行+、-、*、/等多项式运算。

(3) I/O 指令

I/O 指令用于数字输入/输出(DI/DO)，或模拟量输入/输出(AI/AO)。主要是一些赋值语句，如**错误!未找到引用源**。16所示：



图 3-19 IO 指令

(4) 条件指令

条件指令由 IF 开头，用于比较判断是否满足条件，若满足则执行后面的 JMP 或 CALL 指令。支持的比较运算符有：>、>=、=、<=、<、<>，，如图3-17所示，还可以使用逻辑与（AND）和逻辑或（OR）指令对这些条件语句进行运算。



图 3-20 条件指令

在条件指令使用时，请注意以下几点：

- 对于寄存器（R）、模拟量输入/输出及组输入/输出比较指令，可使用全部比较符，但数字量的输入/输出比较时，只能使用=和<>；
- 可使用逻辑与（AND）和逻辑或（OR）指令，每行中允许最多使用5个，但不允许同时使用AND 和OR。

(5) 等待指令

机器人控制系统的等待指令包括两种：指定时间等待指令和条件等待指令。表达式如图3-18所示，“WAIT ...sec”为指定时间等待指令，其余的是条件等待指令。



图 3-21 等待指令

注：条件等待指令中的条件比较项也可进行 AND、OR 运算，添加和设置的操作方法与条件指令相同。

(6) 流程控制指令

机器人控制系统支持三种流程控制指令，分别为标签指令（LBL）、程序结束指令（END）和无条件分支指令。其中无条件分支指令包括跳转指令（JMP LBL）和程序调用指令（CALL）。

程序结束指令（END）是在程序创建初期就已默认添加的指令，表示程序已经结束，用户无需手动添加。在菜单中选择“流程控制”，示教界面显示流程控制指令的选择窗口，如图3-19所示：



图 3-22 流程控制指令

(7) 其他指令

机器人控制系统支持的其他指令包括：坐标系指令、用户报警指令、信息指令、注释指令、倍率指令和预读指令。在菜单树中选择“其他指令”，示教界面将显示其他指令的选择窗口，如图3-20所示：



图 3-23 其他指令

1) 坐标系指令

坐标系指令包括设置和选择两种：

- 坐标系设置：UTOOL[i]= ..., UFRAME[i]= ..., 其中“i”为常量或寄存器R，“...”为位置寄存器“PR”；
- 坐标系选择：UTOOL_NUM= ..., UFRAME_NUM=..., 其中“...”为常量或寄存器R。

2) 用户报警指令

UALM[i]：显示警报信息，并中止程序的运行。报警信息在界面设置，i指定报警信息的序号（从9000到9099），可为常量或寄存器R。

注：若为直接寻址，即输入数据为报警号时，其取值范围为9000到9099，不在此范围输入无效。若为间接寻址，即输入数据为寄存器号，其取值范围为0到199，不在此范围输入无效。

3) 信息指令

信息指令用于在界面上弹出提示信息，格式为：MESSAGE [...], 表示在界面上显示“...”指定的信息；

注：信息内容的长度限制为24个字符（包括字母、符号和数字），超出这个长度，输入无效。

4) 注释指令

由注释符“!”指定。表示“!”后面的字符为注释，不参入程序运行。在指令选择窗口中为“COMMENT”，在程序指令中表现为“!”。

注：信息内容的长度限制为32个字符（包括字母、符号和数字），超出这个长度，输入无效。

5) 倍率指令

倍率指令用于改变进给速度倍率，其格式为：**OVERRIDE=...**，其中“...”可为常量或寄存器 R，倍率的范围是1%-100%。

注：若为直接寻址，即输入数据为倍率值时，其取值范围为1到200，不在此范围输入无效。若为间接寻址，即输入数据为寄存器号，其取值范围为0到199，在此范围输入无效。

3.5.5 程序检查

系统支持对编写的程序进行语法检查，若程序有语法错误，提示报警号、出错程序及错误行号。

程序报警定义请参照本手册后面的附录，错误提示信息中括号中的数据即为报警号。若程序没有错误，提示程序检查完成。

建议用户在首次运行新编写的程序之前，先执行程序检查，以保证程序的正常运行。

3.5.6 程序保存

点击【菜单】键，选择“保存”，提示保存成功，操作完成。同时该程序会自动在自动界面进行加载。

点击【菜单】键，选择“另存为”，输入程序名，提示另存成功，择操作完成。

4 自动运行

4.1 概述

自动界面主要功能是运行机器人程序。用户在示教界面编辑并保存好示教程序之后，自动界面会自动加载该程序。

自动运行模式下可以运行机器人程序，任何程序都必须先加载到内存中才能运行。自动界面功能菜单主要由加载程序、显示模式切换、循环模式切换、指定运行起始行等组成。另外，对程序的操作运行、单步向前运行、单步向后运行、暂停、停止五项操作由示教器的自动操作按键控制。


自动界面包含程序显示区、坐标显示区、调修值设置区。如图 4-1所示：



图 4-1 自动运行界面

4.2 加载程序

加载程序过程与示教界面打开程序的过程类似，不再赘述。加载过程会出现进度对话框，对话框消失之后提示加载成功即表示完成程序加载。

- (1) 运行程序：程序加载完成后，按下【运行程序】键，程序即从前往后一步步执行。程序光标可定位至程序当前执行的程序行，程序名右侧也会显示当前运行行的行号及运行状态。
- (2) 程序暂停：在程序运行过程中，可按下【暂停】键暂停程序的执行。
- (3) 程序停止：在程序运行过程中，可按下【停止】键停止程序的执行。
- (4) 单步向前：单步模式开启后，在程序光标没有在最后一行的情况下，键每按下一次，程序向前执行一行。

4.3 显示模式

显示模式主要用于程序运行时操作者观察机器人的直角坐标与关节坐标。当前显示为关节坐标时，按【菜单】键，选择“切换至直角”，即可切换为直角坐标。同理，当前显示为直角坐标时，用同样的方法可以切换至关节坐标。

4.4 循环模式

循环模式包含单周、循环两种模式。单周模式指的是程序运行一次结束后自动停止。循环模式指的是程序运行至最后一行后再次回到程序起始行重新运行。单周模式下，按【菜单】键，选择“切换至循环”即可切换至循环模式。在循环模式下用同样的方式可以切换至单周模式。

4.5 指定行运行

“指定行运行”的意思是在指定特定行号之后，运行程序时从该行开始运行。注意：“指定行”只在程序运行状态为停止状态时才能使用。设置指定行的方式有以下两种：

- (1) 将焦点移至想要指定的行，然后长按【确定】键，弹出确认对话框，点击确定后，即完成设置。如图 4-2所示：

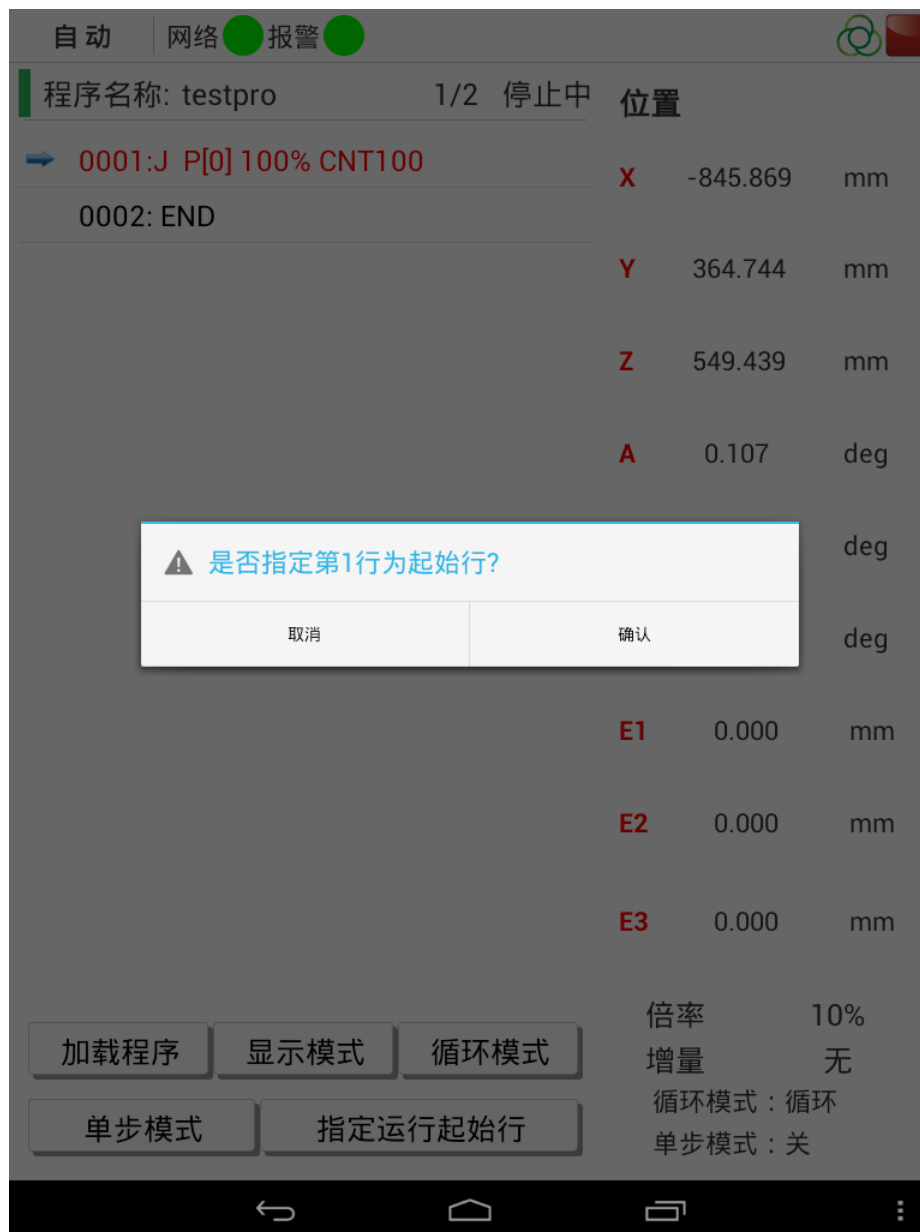


图 4-2 长按指定起始行

- (2) 按【菜单】键，选择“指定运行起始行”在对话框中填入符合条件的行号，点击确定完成设置。如图 4-3所示：

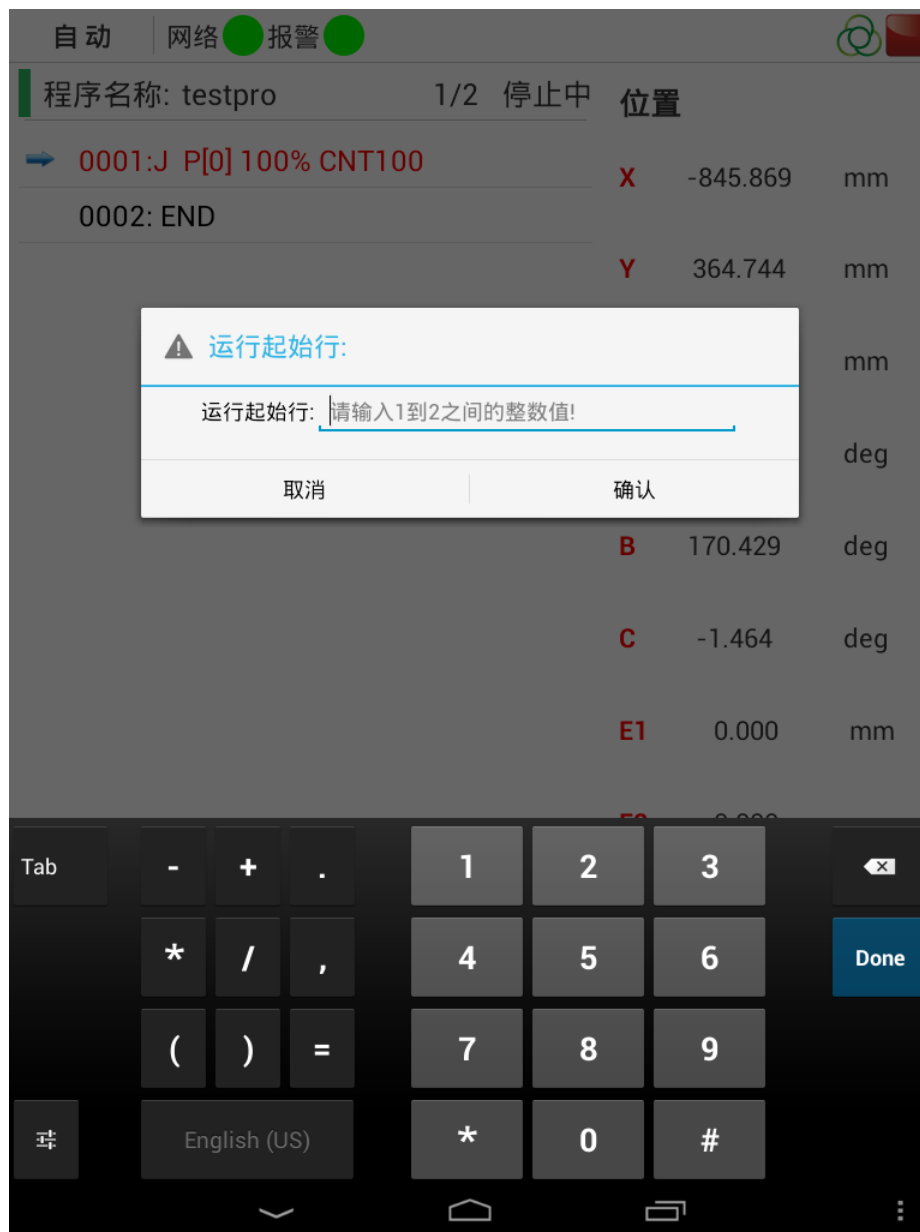


图 4-3 指定起始行对话框

5 寄存器

5.1 概述

寄存器页面主要用于查看和更改寄存器的值。如所示：



图 5-1 寄存器界面

5.2 R 寄存器

控制系统支持200个 R 寄存器变量，寄存器从0开始编号，向上或向下滑动屏幕可

看见0至199 R 寄存器的值。

点击 R 寄存器列表中任一行，会弹出 R 寄存器设置对话框，修改所选 R 寄存器的值，如图21所示，在确认修改后会自动刷新列表。



图 5-2 R 寄存器设置对话框

5.3 位置寄存器

位置寄存器作为全局变量，用于存放位置信息。机器人控制系统支持100个位置寄存器，寄存器编号为0~99。支持对指定位置寄存器的坐标类型、组和坐标值进行设置修改。

点击位置寄存器列表中任一行，弹出位置寄存器设置对话框，修改所选位置寄存器

的各个参数，点击直角坐标类型如图24所示，点击关节坐标类型如图25所示，在确认修改后会自动刷新该位置寄存器列表。

位置寄存器的修改与示教页面的修改位置对话框操作一致，此处不再赘述。

5.4 查找

点击【菜单】键，选择“查找”，弹出查找对话框，选择 **R** 寄存器或者 **PR** 寄存器查找，输入符合条件的序号后，点击确定，寄存器界面会自动滚动至查找的目标位置。

6 IO 信号

6.1 概述

机器人控制系统提供了完备的 I/O 通信接口，可以方便地与周边设备进行通信。本系统的 I/O 板提供的常用信号处理有输入信号 X 和输出信号 Y。输入/输出主要是对这些输入/输出状态进行管理和设置。

输入/输出点数及状态从下位机获取。

6.2 信号显示和设置

进入该界面时，软件会自动刷新信号状态并显示。IO 信号的输入信号界面显示如图26所示，输出信号界面如图27所示：

对一个信号进行设置时，要先解开相应的“锁”，即点击相应的开关按钮，开锁后即可修改相应信号的状态。

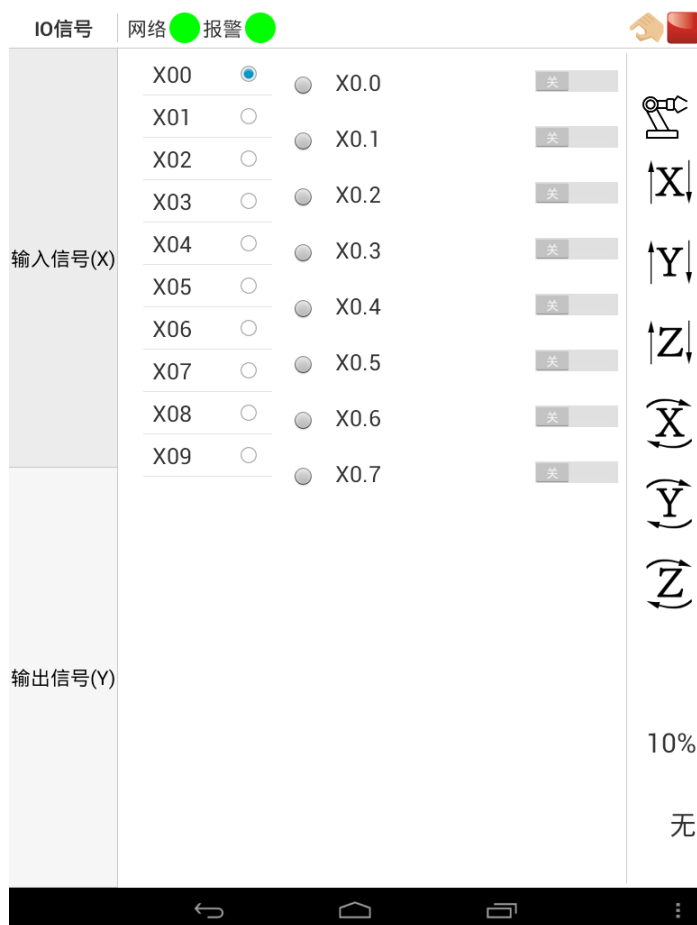


图1 输入信号界面

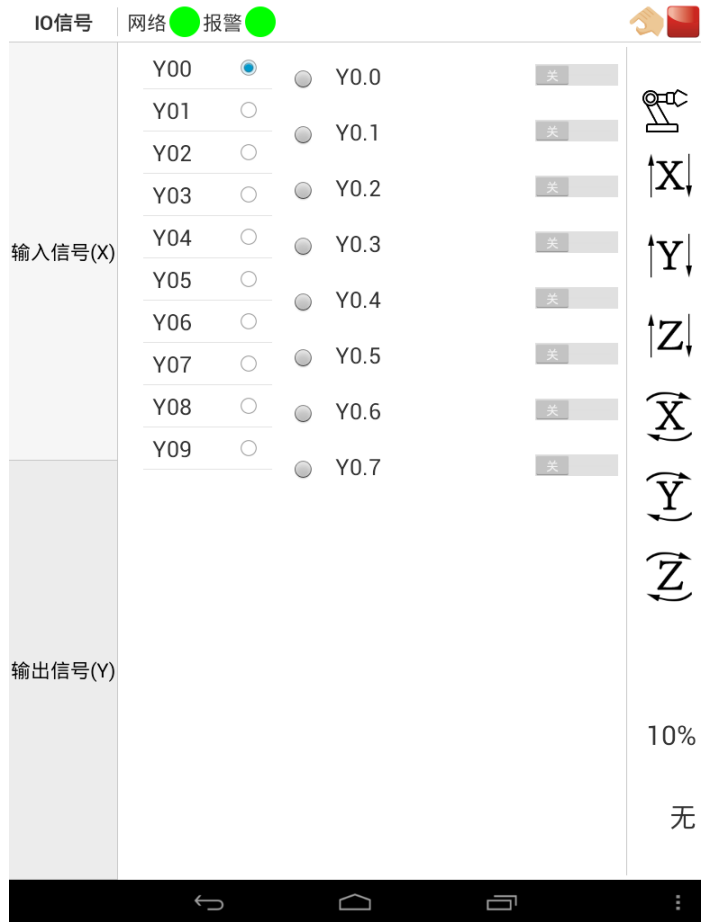


图2 输出信号界面

7 设置界面

7.1 概述

7.1.1 初始密码

首次打开设置界面会出现提示“设置参数需要密码”，初始密码为“443524”，正确输入密码后，会增加显示系统参数、组参数、轴参数、机械参数四个子界面。如所示



图 7-1 密码输入对话框

7.1.2 界面简介

设置界面包含的子界面有：系统参数、组参数、轴参数、机械参数、工具坐标系设定、工件坐标系设定、校准。点击左侧选项卡，右侧会切换至相应界面。参数值是从下位机同步上来的，如果上下位机网络并未连接时，参数值为0或空。

本系统支持5个控制组，最多32个物理轴。组参数与轴参数相互关联，每个组最多可以配置9个逻辑轴。用户可根据需要设置物理轴与逻辑轴之间的映射关系，详见组参数设置详细信息。

注意，每个物理轴只能对应一个组的一个逻辑轴，不能进行多重映射。配置好的物理轴可以在轴参数列表中查看该轴所属控制组情况。

设置界面如下图所示：

	参数号	参数含义	参数值	
系统参数	20000	插补周期(μ s)	2000	 
	20005	硬件通讯方式	0	
组参数	20100	报警履历最大记录数	200	 
	20200	WAIT指令TIMEOUT时间(s)	120.0	
轴参数				  
机械参数				
工具坐标系设定				
工件坐标系设定				10%
校准				无

图 7-2 设置界面

7.2 系统参数

目前，本机器人控制系统支持的系统参数共4个：插补周期、硬件通讯方式（中断或扫描）、报警履历最大记录数及 WAIT 指令 TIMEOUT 时间。系统参数界面如下图所示：



系统参数	参数号	参数含义	参数值
系统参数	20000	插补周期(μ s)	2000
组参数	20005	硬件通讯方式	0
组参数	20100	报警履历最大记录数	200
组参数	20200	WAIT指令TIMEOUT时间(s)	120.0
轴参数			
机械参数			
工具坐标系设定			
工件坐标系设定			
校准			

图 7-3 系统参数

点击相应的行即可对选中的参数进行设置。

7.3 组参数

点击“组参数”进入到组参数界面。如下图所示：

设置		网络	报警	
系统参数	参数号	参数含义	参数值	
	30000	组启用标志	1	
组参数	30005	组控制单元类型	HRT-5	
	30010	J1轴号	1	
	30011	J2轴号	2	
轴参数	30012	J3轴号	3	
	30013	J4轴号	4	
	30014	J5轴号	5	
机械参数	30015	J6轴号	6	
	30016	J7轴号	7	
	30017	J8轴号	8	
工具坐标系设定	30018	J9轴号	9	
	30040	手动下关节轴运动最大速度(mm/s)	30.0	
工件坐标系设定	30041	手动下关节轴运动加速度(mm/s ²)	250.0	
	30042	手动下关节轴运动加加速度(mm/s ³)	0.0	10%
校准	30050	手动下平动最大速度(mm/s)	200.0	无

图 7-4 组参数设置页面

点击参数列表中的行，设置相应参数。

注：对轴号进行设置时，由于每个物理轴只能对应单个组的一个逻辑轴，不能进行多重映射，所以如果有某个组已使用了某个轴号，其他组就不能再使用了。

7.4 轴参数

目前，本机器人控制系统支持的轴数共有9个。每个轴可以属于不同的组号。点击轴参数，切换到轴参数设置界面，如下图所示。选择要设置的轴号，进入参数列表，点击参数列表中的行，设置相应参数。

设置		网络	报警		
系统参数	参数号	参数含义	参数值		
	60000	轴名	Jn		
组参数	60001	轴类型(0关节轴/1旋转轴/2直线轴)	0		
	60010	是否带反馈(0是/1否)	0		
轴参数	60020	螺距(mm或°)	360.0		
	60030	指令类型(0增量/1绝对)	1		
	60031	电子齿轮比分子	120		
机械参数	60032	电子齿轮比分母	1		
	60040	电机方向取反(0是/1否)	0		
工具坐标系设定	60041	编码器脉冲数	131072		
	60042	编码器类型	0		
	60043	反馈齿轮比分子	1		
工件坐标系设定	60044	反馈齿轮比分母	1		
	60045	反馈位置偏移	0		
	60050	跟踪误差允许值(mm)	30.0	10%	
校准	60060	正向软限位(°)	170.0		
	60061	负向软限位(°)	-170.0		
			返回		无

图 7-5 轴参数设置界面

7.5 机械参数

机械参数的设置与上述参数设定过程基本一致，机械参数设置界面如下图所示：

设置		网络	报警	
系统参数	参数号	参数含义	参数值	
	40000	机械结构类型	6008	
组参数	40001	连杆1长度a0	0.0	
	40002	连杆1扭角a0	0.0	
	40003	旋转关节1偏置d1或线性 关节1关节角θ1	0.0	
轴参数	40004	连杆2长度a1	150.0	
	40005	连杆2扭角a1	90.0	
机械参数	40006	旋转关节2偏置d2或线性 关节2关节角θ2	0.0	
	40007	连杆3长度a2	570.0	
	40008	连杆3扭角a2	0.0	
工具坐标系设定	40009	旋转关节3偏置d3或线性 关节3关节角θ3	0.0	
	40010	连杆4长度a3	130.0	
工件坐标系设定	40011	连杆4扭角a3	3.0	
	40012	旋转关节4偏置d4或线性 关节4关节角θ4	640.0	10%
校准	40013	连杆5长度a4	0.0	
	40014	连杆5扭角a4	-90.0	无

图 7-6 机械参数设置界面

7.6 工具坐标系设定

机器人控制系统支持16个工具坐标系设定。点击“工具坐标设定”，可设置相应工具坐标系或工件坐标系的各个坐标值，如下图所示：

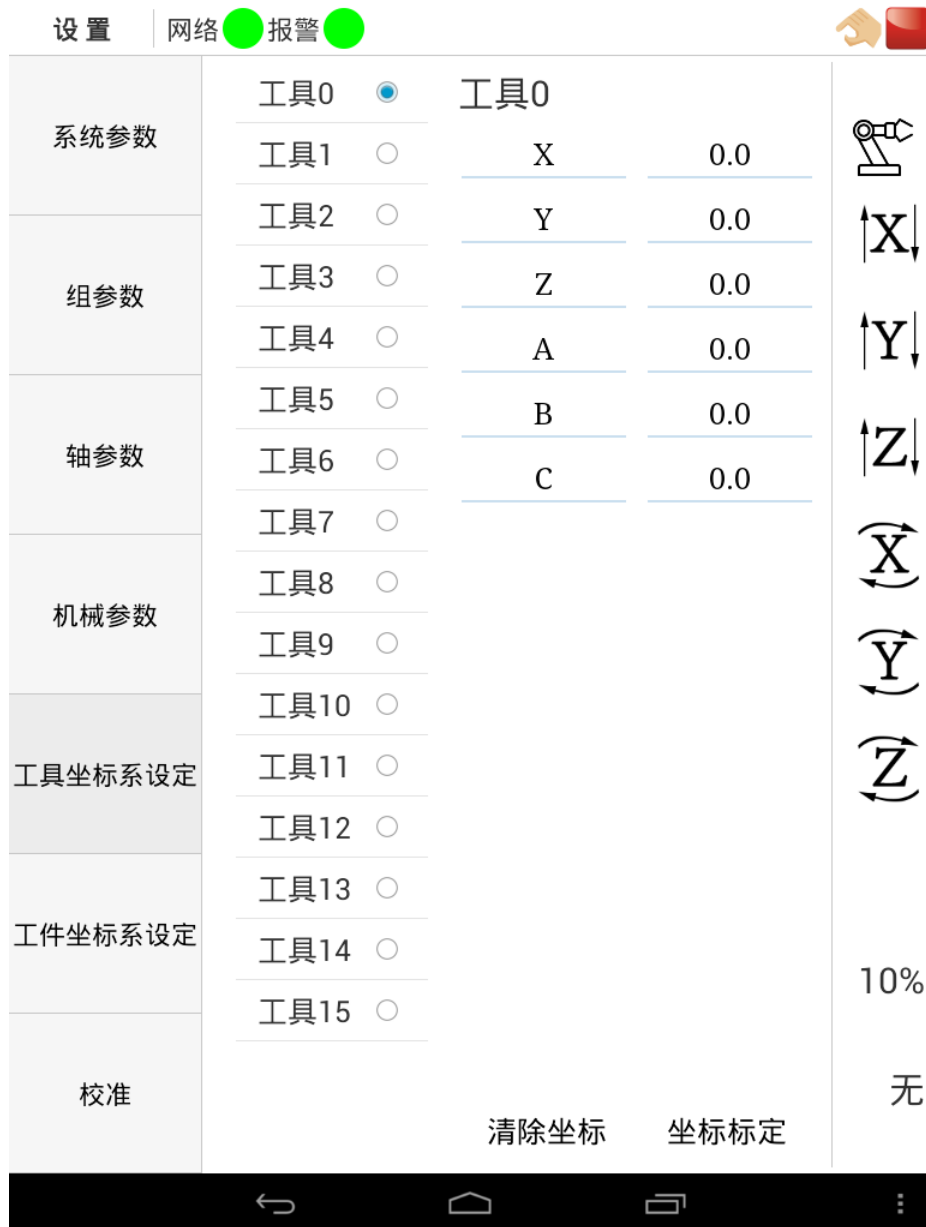


图 7-7 工具坐标系设定

选择一个工具坐标系，右侧将显示选中坐标系的坐标值。点击一个轴的坐标值，即可进行坐标值的修改。点击“清除坐标”，可将当前选中的坐标系的各个坐标值清零。点击“坐标标定”，可弹出坐标标定对话框，如下图所示：



图 7-8 工具坐标标定对话框

7.6.1 三点标定

通过标定空间中机器人末端在坐标系中的三个不同位置来计算工具坐标系。

工具坐标系三点标定操作步骤如下：

- (1) 在标定对话框中，点击“三点标定”弹出下拉框可选择标定方式，此处选择三点标定，选中一个被标定点，如选择“接近点1”，然后使用实体按键移动机器人，点击“修改位置”即可将机器人当前位置记录到“接近点1”。



图 7-9 接近点 1 指定完成

- (2) 按照上述方法指定接近点2、3的位置，当三个位置都显示已修改时，按下“确认”，即完成三点标定，此时相应工具标定完成。

7.6.2 六点标定

通过标定空间中机器人末端六个不同姿态的位置来计算工具坐标系。

工具坐标系六点标定的详细操作步骤如下：

- (1) 点击工具坐标系进入工具坐标系界面，选中需要标定的工具号（工具0不能被

标定), 点击“坐标标定”, 弹出坐标标定对话框, 如下图所示:



图 7-10 六点标定

- (2) 在图2-11所示对话框中, 点击“三点标定”弹出下拉框可选择标定方式, 此处选择六点标定, 选中一个被标定点, 如选择“接近点1”, 然后使用实体按键移动机器人, 点击“修改位置”, 即可将机器人当前位置记录到“接近点1”。



图 7-11 接近点 1 指定完成

- (3) (4) 按照上述方法指定其余点的位置, 当六个位置都显示“已修改”时, 按

下“确认”，即完成六点标定，此时相应工具标定完成。

7.7 工件坐标系设定

工件坐标系是由用户在工件空间定义的一个笛卡尔坐标系。工件坐标包括:(X,Y,Z)用来表示距原点的位置,(A,B,C)用来表示绕 X-, Y-, Z-轴旋转的角度。与工具坐标系相同,机器人控制系统支持16个工件坐标系设定,每个工件坐标系可以属于不同的组号,也可为每个工件坐标系添加相应的注释说明。

在设置界面左侧选项卡中点击“工件坐标系设定”,即可进入工件坐标系设定界面。如下图所示:

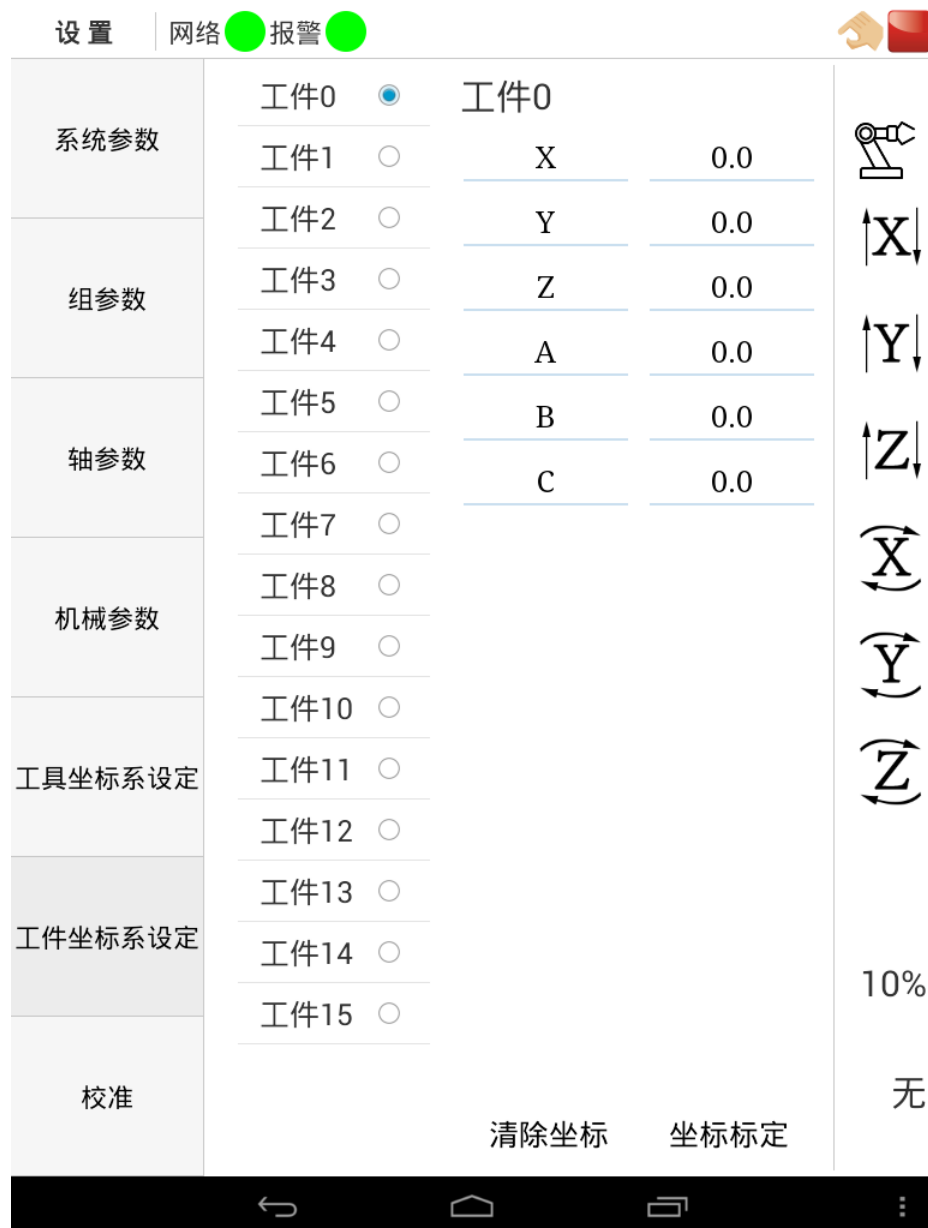


图 7-12 工件坐标系设定

选择一个工件坐标系，右侧将显示选中坐标系的坐标值。点击一个轴的坐标值，即可弹出窗口修改选中的坐标值。点击“清除坐标”，可将当前选中的坐标系的各个坐标值清零。

工件坐标系可以用以下两种方式进行标定：

1. 三点法

将第一个标定点击为工件坐标系 X 轴的起点（如图2-15所示），将工具 TCP（即工具坐标中心点）沿工件坐标系+X 方向移动一定距离作为 X 方向延伸点，再从工件坐标系 XOY 平面第一或第二象限内选取任意点作为 Y 方向延伸点。第一个标定点击为工件坐标系绝对原点。由此三个点计算出工件坐标系。

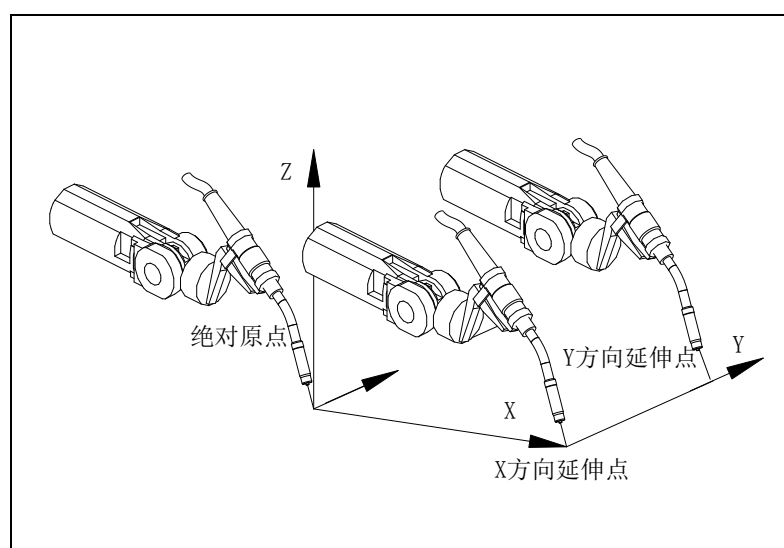


图 7-13 工件三点法标定

2. 四点法

将第一个点作为相对原点，将工具 TCP 沿工件坐标系+X 方向移动一定距离作为 X 方向延伸点，再从工件坐标系 XOY 平面第一或第二象限内选取任意点作为 Y 方向延伸点，最后操作机器人到第四个点，作为绝对原点。由此四个点计算出工件坐标系。

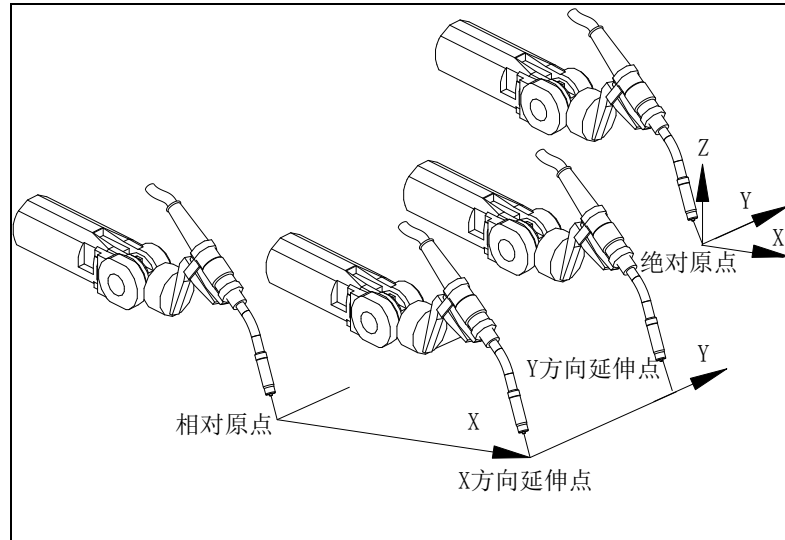


图 7-14 工件四点法标定

工件坐标系的标定操作步骤与工具坐标系标定类似，在此不再介绍。

8 生产管理

8.1 概述

该界面主要显示与生产相关的一些信息，如软件版本、使用期限、操作人员、以及当前连接的网络状态，报警历史、程序管理和操作记录。

8.2 报警历史

该界面是生产管理的默认界面，可以查看当前的报警历史列表。

8.3 程序管理

点击左侧“程序管理”按钮，可以切换至程序管理界面，该界面用于管理示教器中的程序，并支持与 U 盘之间互相复制程序文件。程序管理界面如下图所示：



图 8-1 程序管理界面

在程序管理界面，复选所需删除的程序文件即可删除所选定的程序文件，复选所需复制的程序文件，选择相应 U 盘，即可复制所选定的程序文件至相应 U 盘。

8.4 操作记录

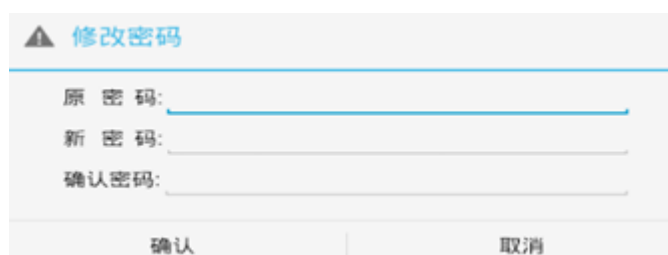
点击“操作记录”按钮即可切换至操作记录界面，该界面用于查看当前的操作记录列表（包含自动运行、手动运行、修改参数等）。

8.5 关于设备

点击“关于设备”按钮，切换至关于设备界面，该界面主要包含控制器软件版本、示教器软件版本、网络状态、IP 地址。

8.6 修改密码

点击功能菜单中的“修改密码”会弹出如下图所示对话框，输入原密码和新密码即可修改密码。



The image shows a dialog box with a title bar containing a triangle icon and the text '修改密码'. Below the title bar are three input fields labeled '原密码:', '新密码:', and '确认密码:'. At the bottom of the dialog box are two buttons: '确认' (Confirm) on the left and '取消' (Cancel) on the right.

图 8-2 修改密码对话框

9 码垛

9.1 概述

码垛工艺是指通过对垛的外形尺寸、垛数、层数基本参数设置，对垛的摆放位置进行简单确认就能实现所有垛的整齐摆放。

9.1.1 基本概念

1. 垛：一个垛即是工件。
2. 托盘：用于放置垛的物品（区域）。
3. 码垛工艺指令数量：共 10 个。
4. 排样数：1-2，即最多可实现种不同的排放方式，通常 1 层 1 种排样。
5. 参考点：即第一个垛的摆放位置，以后每个垛的坐标以其为基准进行偏移。
6. 排样模式：本系统支持三种排样模式：矩阵模式、标准模式、全点模式。
 - a) 矩阵模式
即以 M 行 N 列矩阵的方式进行排放。
 - b) 标准模式
本系统内置了两种标准垛型。标准模式的垛型均采用奇数层与偶数层排样不同的设定。
 - c) 全点模式
即需要制定每一个工件的位置，然后根据指定的位置进行排放。

9.1.2 准备工作

1. 夹具准备
2. 建立工件坐标系（参考7.7节）

9.2 码垛工艺设置

9.2.1 码垛工艺选择

下图是码垛工艺选择界面，本系统支持10个码垛工艺。

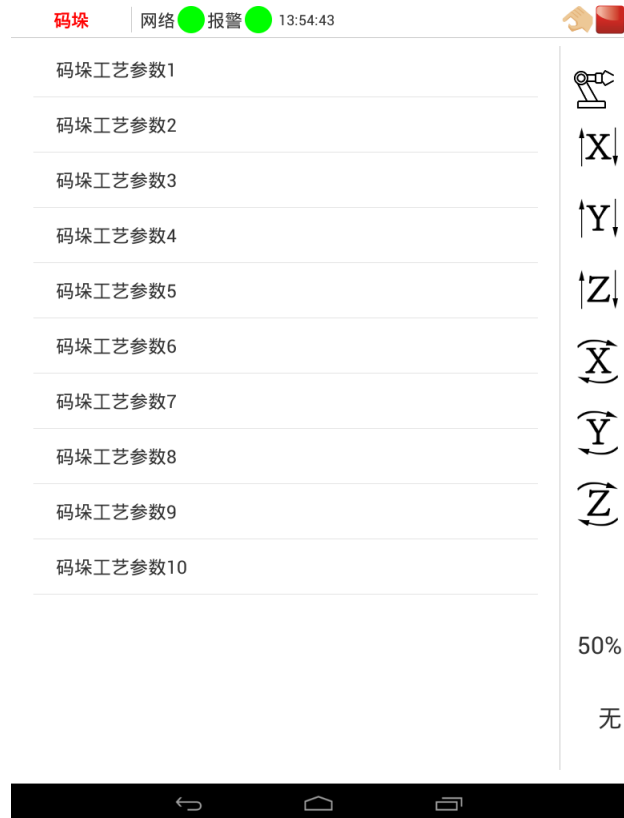
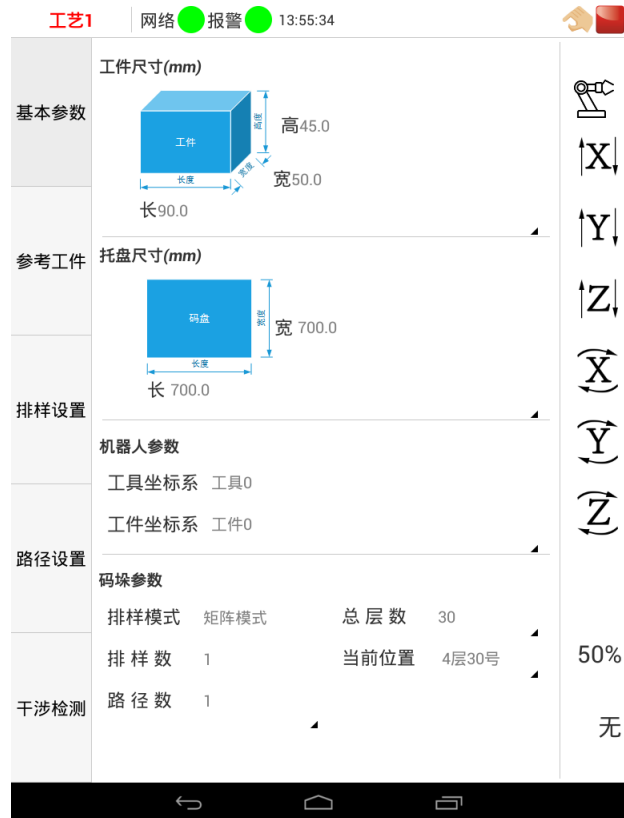


图 9-1 码垛工艺选择界面

9.2.2 基本参数设置

基本参数包含工件尺寸、码盘尺寸、坐标系选择、基本排样参数等。如下图所示：



1. 工件尺寸设置

包含工件的长度、宽度、高度的设置。取值范围均在0~1000mm内。设置对话框如下图所示：



图 9-2 工件尺寸设置对话框

2. 码盘尺寸设置

包含码盘的长度、高度的设置。取值范围均在0~20000mm 范围内。设置对话框如下图所示：



图 9-3 码盘尺寸设置对话框

3. 坐标系参数设置

用于指定该码垛工艺所使用的工具坐标系及工件坐标系。设置对话框如下图所示：

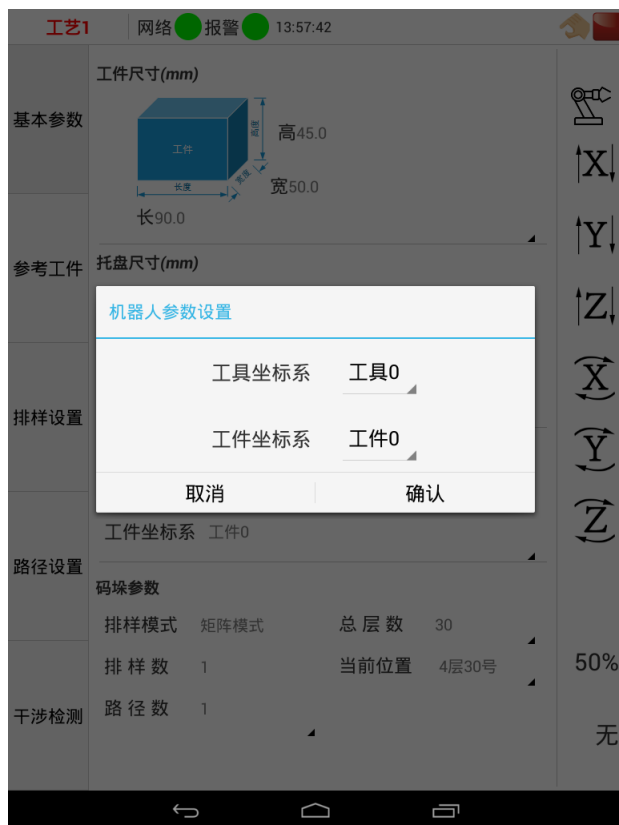


图 9-4 码垛工艺坐标系选择对话框

4. 排样模式、排样数、路径数的设置

排样模式有三种：矩阵模式、标准模式、全点模式。下表是不同排样模式排样数和路径数的限制情况

排样模式	矩阵	标准	全点
排样数	1	2	1
路径数	1	1	1~6

表 9-1 排样模式

5. 层数设置

即该码垛工艺的总层数。本系统支持最多100层的码垛工艺。设置对话框如下图所示：



图 9-5 码垛层数

6. 起始工件设置

即设置从整个垛的第几号工件开始进行码垛。每次码垛之后，该参数均会指向即将堆垛的那个工件。如需从新开始码垛则设置为1层1号即可。设置对话框如下图所示：



图 9-6 开始工件号设置

9.2.3 参考工件设置

在矩阵模式及标准模式下，参考工件即第一个堆垛点的位置。而在全点模式下参考工件是设置其他工件的坐标参考点。

参考工件的 X、Y 值与选定工件坐标系的 X、Y 值一致。相对位姿 Z、A、B、C 均是该点相对于选定工件坐标系的原点的值。



图 9-7 参考工件设置

将机器人移动至第一个堆垛点的位置，点击“将当前位置设置为参考点”按钮即自动计算出该点的相对位姿。也可手动输入相对位姿的值。设置对话框如下图所示：



图 9-8 参考工件位姿设置

9.2.4 排样设置

- 矩阵模式下的排样设置

矩阵模式下的排样参数主要有行数、列数、行优先、行间隔、列间隔。设置完参数后，界面上会显示本排样的示意图。

以下对各参数进行说明：

行优先：即排样时是一行一行地排放，如设置为否则是一列一列地排放。

行间隔：两行工件中心的距离。

列间隔：两列工件中心的距离。

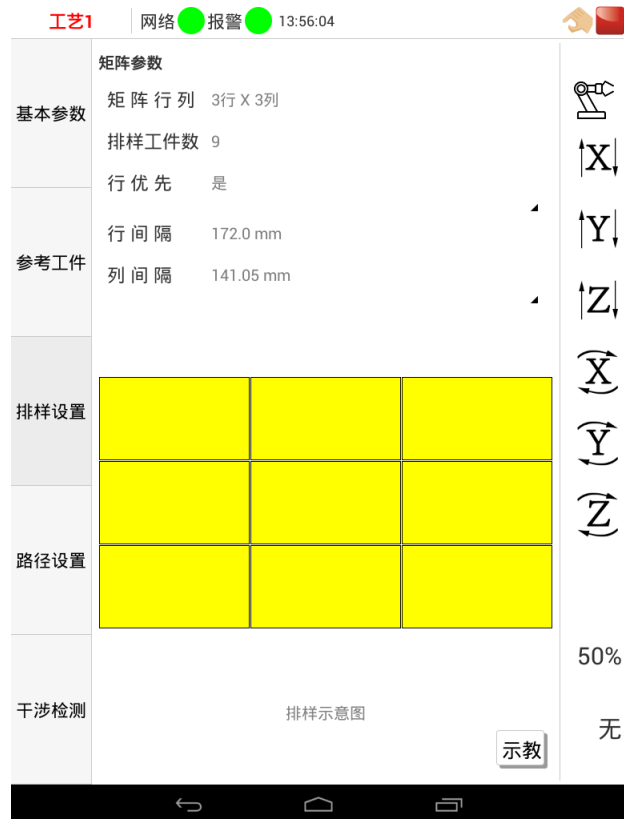


图 9-9 矩阵排样设置

- 矩阵排样的示教

矩阵排样模式的示教主要是用于示教行间隔以及列间隔。示教时，指定工件1、行方向最后一个工件以及列方向最后一个工件的位置，即可计算得出行间隔和列间隔。点击“完成示教”，示教结束，行列间隔的数据也会刷新到界面上。



图 9-10 矩阵模式示教

- 标准模式下的排样设置

标准模式提供两种基本垛型：回形排列、纵横排列。

1. 回形排列

根据工件尺寸的长宽摆放成相应的回形，工件尺寸变化后，回形排列的排样工件数也会变化。

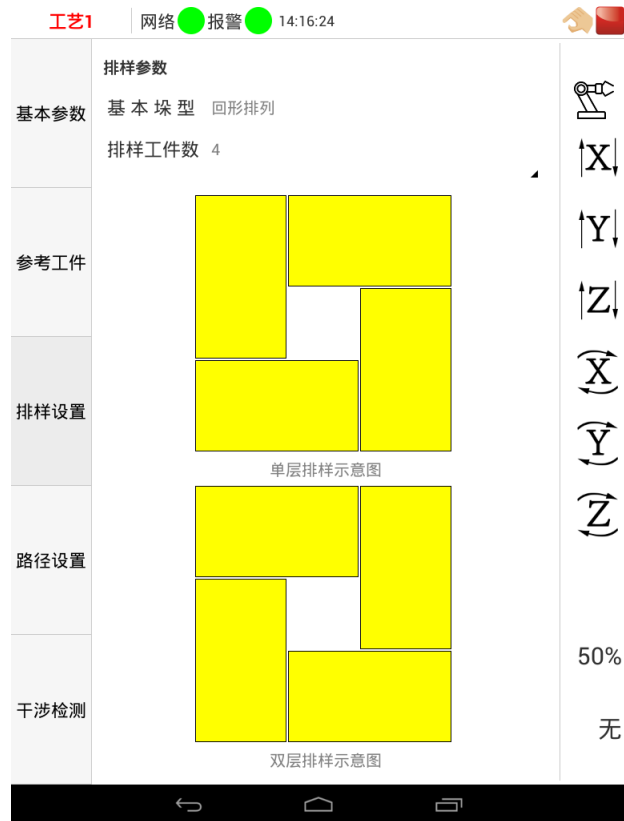


图 9-11 回形排样示意图 1

当工件尺寸修改后，点击查看回形排样示意图，会发现排样相应发生变化。如下图所示：

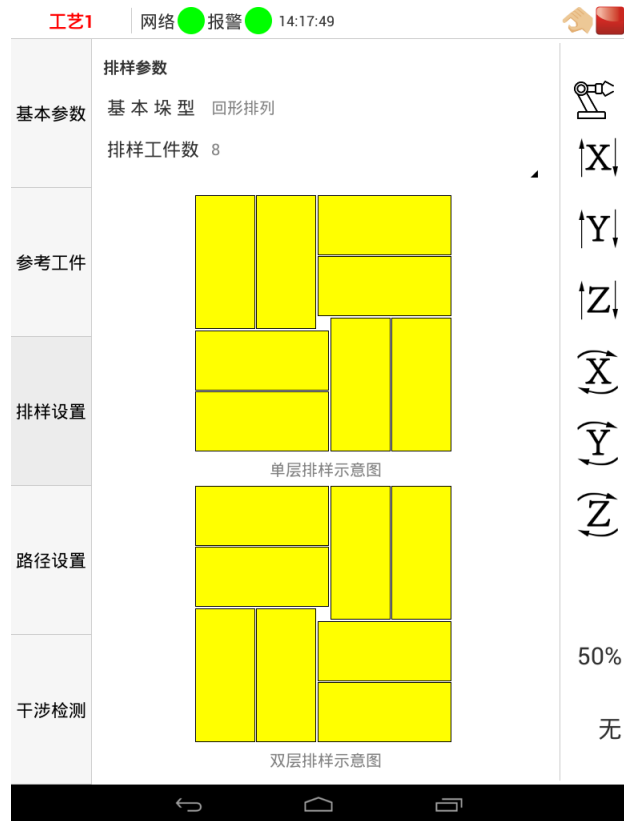


图 9-12 回形排样示意图 2

2. 纵横排列

纵横排列是首行将工件以宽方向为底进行排放，而其他行以长方向为底进行排放。如下图所示：

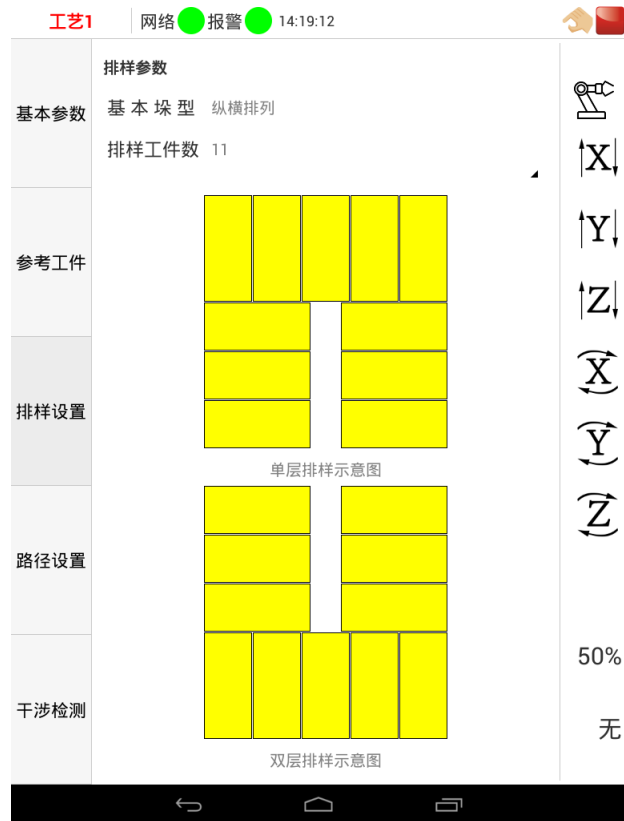


图 9-13 纵横排列示意图

纵横排列的设置参数有：首行列数，其他行数。

首行列数：即首行有几列。

其他行数：即除去首行之外的行数。

纵横排列的设置对话框如下图所示：

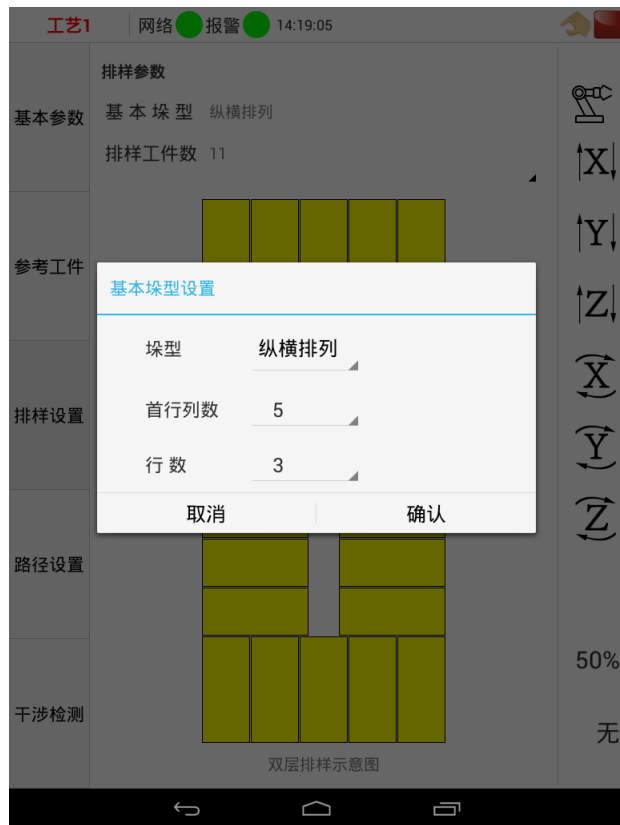


图 9-14 纵横排列参数设置

- 全点模式下的排样设置

全点模式下需要设置各工件相对于参考工件的相对位置，故设置参数主要是排样工件数以及各工件的相对位置及路径号。如下图所示：

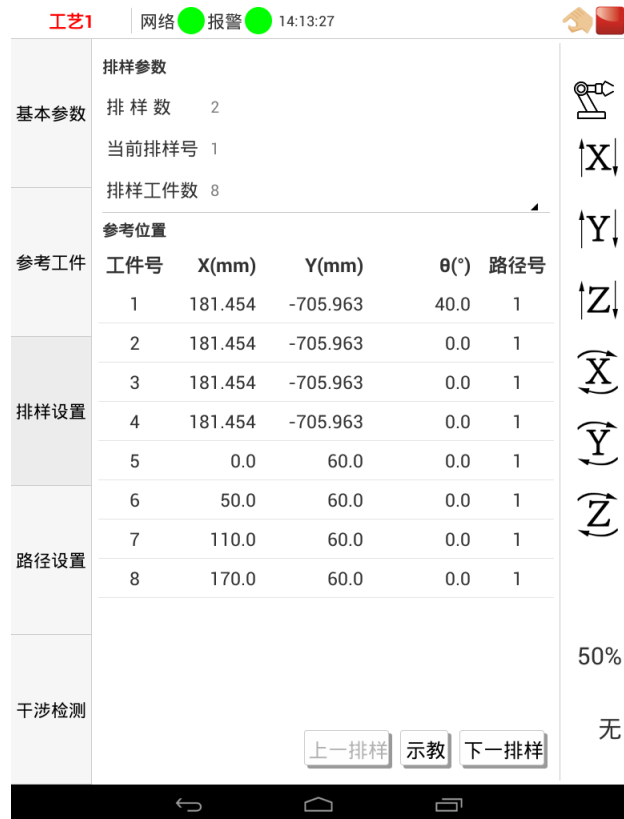


图 9-15 全点模式排样设置

9.2.5 路径设置

路径即堆垛时趋近和回退的路径，通过设置趋近点、回退点相对于堆垛点的相对位置即可确定路径。点击“上一路径”、“下一路径”在各路径参数间切换。路径设置界面如下图所示：

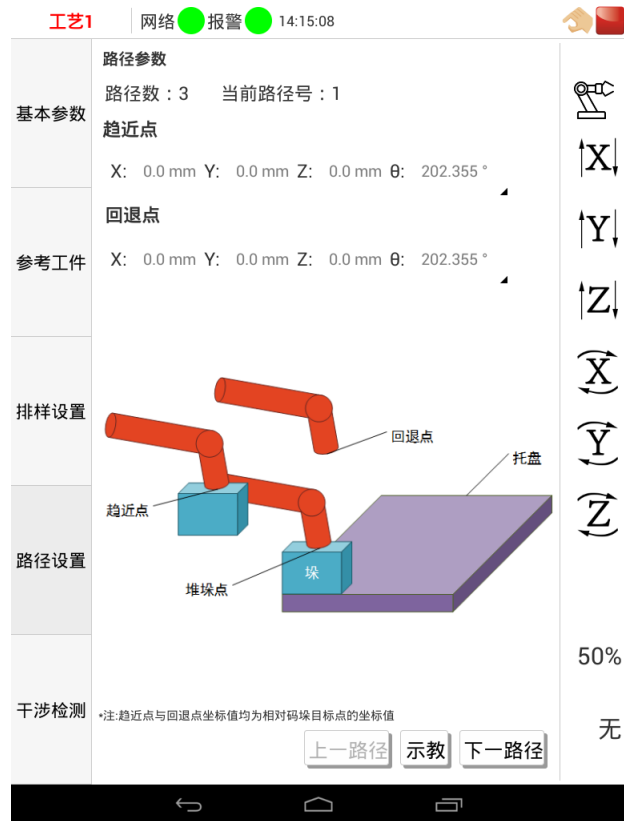


图 9-16 路径设置

趋近点与回退点的参数值可以手动输入也可通过示教得出。

路径示教界面如下图所示。路径示教的步骤如下：

- 1) 将机器人移动至堆垛点，然后点击记录位置；
- 2) 再将机器人移动至趋近点，然后点击相应的记录位置
- 3) 同样将机器人移动至回退点，然后点击相应的记录位置
- 4) 点击完成示教即可得出路径参数。

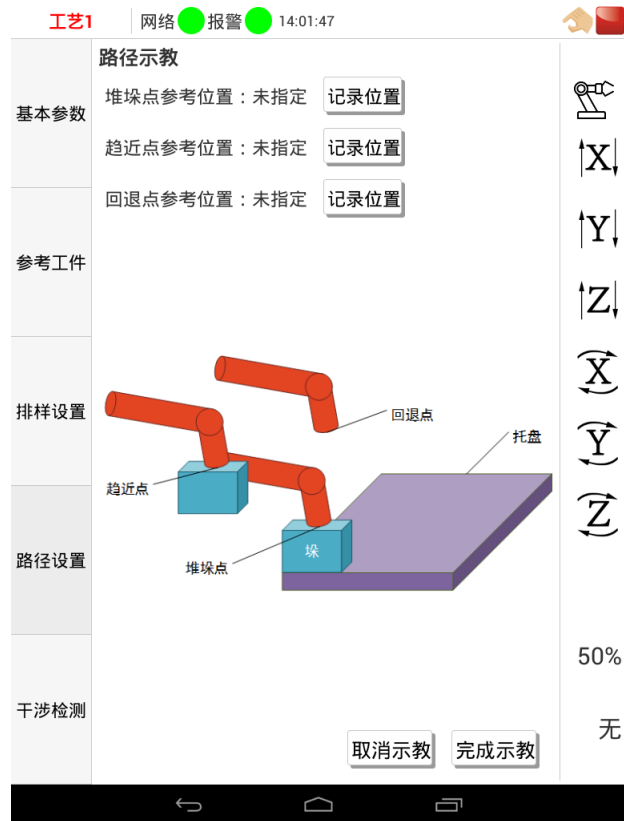


图 9-17 路径示教

9.2.6 干涉检测设置

干涉检测的检测内容包括：

- 1) 码垛之间是否干涉
- 2) 辅助点到码垛点的平移运动是否产生碰撞

干涉检测的设置界面如下图所示：

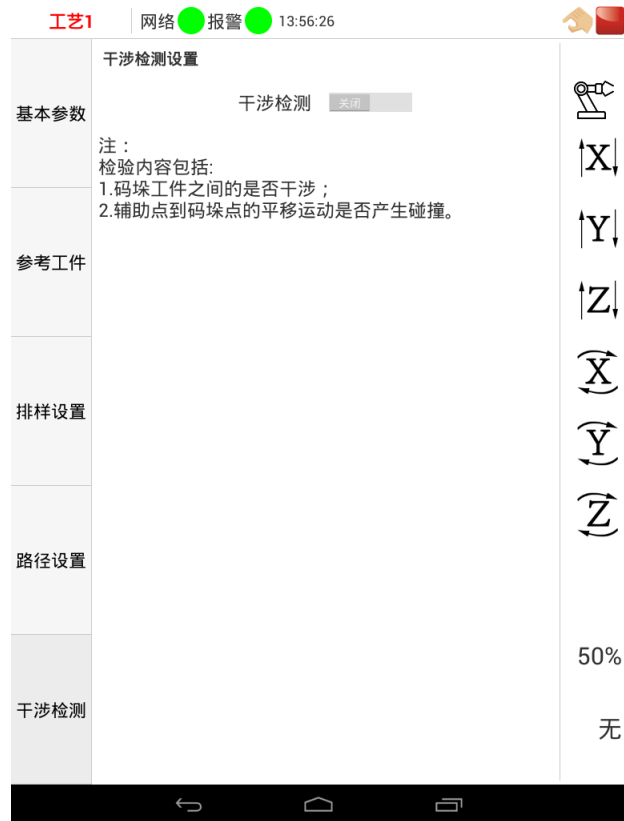


图 9-18 干涉检测设置界面

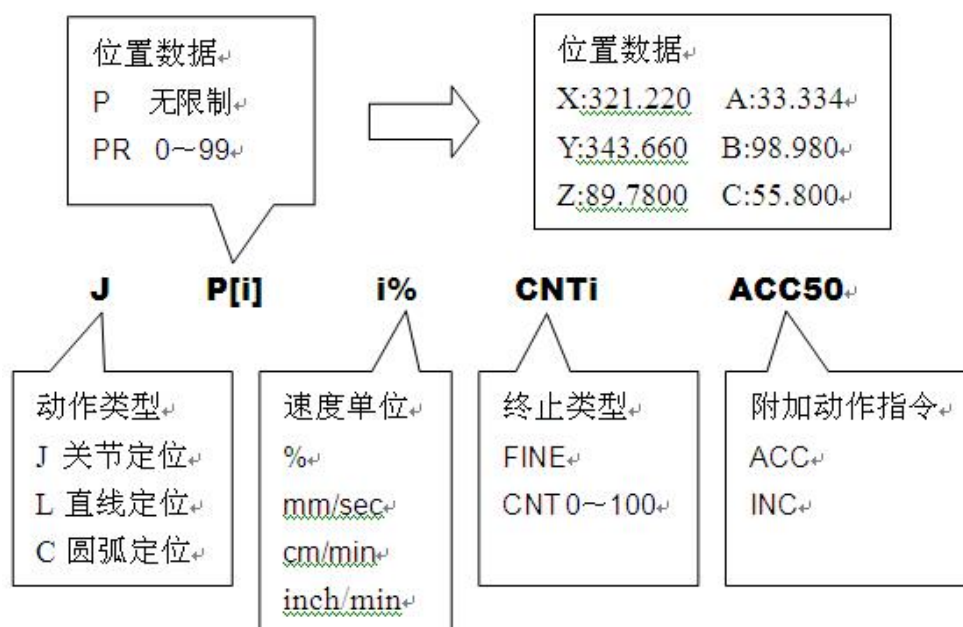
II 编程篇

1 运动指令

运动指令实现以指定速度、特定路线模式等将工具从一个位置移动到另一个指定位置。在使用运动指令时需指定以下几项内容：

- (1) 动作类型：指定采用什么运动方式来控制到达指定位置的运动路径；
- (2) 位置数据：指定运动的目标位置；
- (3) 进给速度：指定机器人运动的进给速度；
- (4) 定位路径：指定相邻轨迹的过渡形式；
- (5) 附加运动指令：指定机器人在运动过程中的附加执行指令。

运动指令格式如下所示：



在程序示教的过程中，使用菜单树中的“运动指令”即可添加标准的运动指令，详细操作步骤请参阅前面章节的操作说明。

1.1 动作类型

动作类型，即指定到达指定位置的运动路径，机器人运动的类型有三种：关节定位（J）、直线运动（L）、圆弧运动（C）。

1.1.1 关节定位

关节定位是移动机器人各关节到达指定位置的基本动作模式。独立控制各个关节同时运动到目标位置，即机器人以指定进给速度，沿着（或围绕）所属轴的方向，同时

加速、减速或停止。工具的运动路径通常是非线性的，在两个指定的点之间任意运动。以最大进给速度的百分数作为关节定位的进给速度，其最大速度由参数设定，程序指令中只给出实际运动的倍率。关节定位过程中没有控制被驱动的工具的姿态。

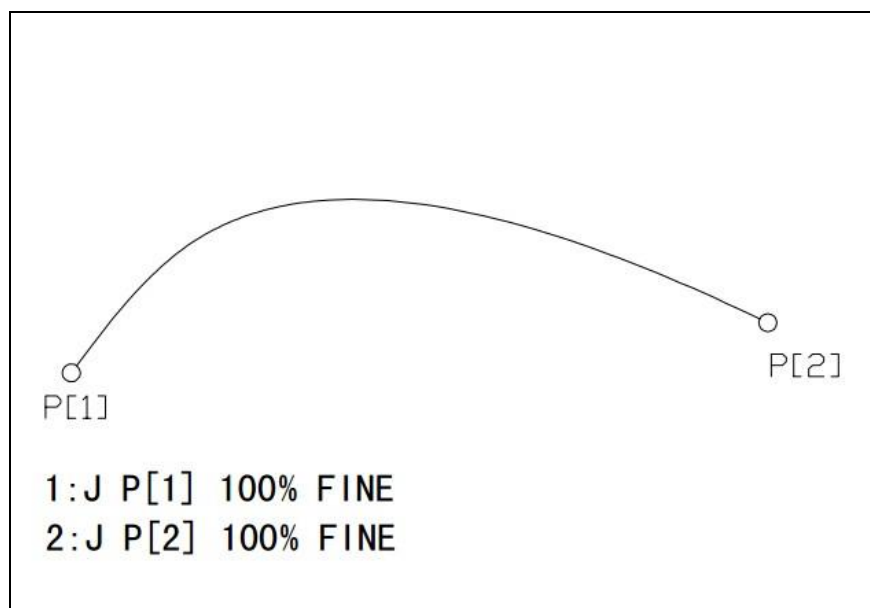


图 1-1 J 关节定位

图中由 P[1] 点以 100% 速度采用关节定位方式移动至 P[2] 点。

1.1.2 直线运动 L

直线运动指令控制 TCP（工具中心点）沿直线轨迹运动到目标位置，其速度由程序指令直接指定，单位可为 mm/sec、cm/min、inch/min。通过区别起点和终点时的姿态，来控制被驱动的工具的姿态。

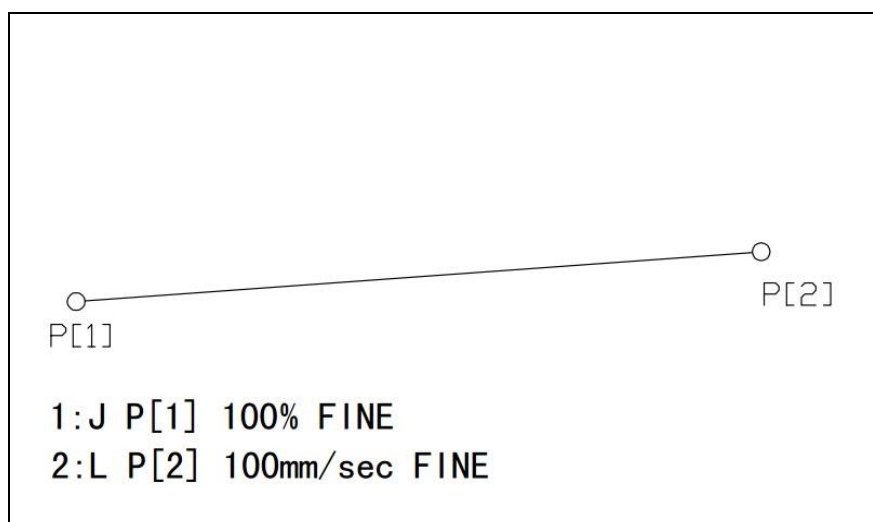


图 1-2 L 直线运动

图中由 P[1]点以100mm/sec 的速度采用直线运动方式移动至 P[2]点。

1.1.3 圆弧运动 C

圆弧运动指令控制 TCP（工具中心点）沿圆弧轨迹从起始点经过中间点移动到目标位置，中间点和目标点在指令中一并给出。其速度由程序指令直接指定，单位可为 mm/sec、cm/min、inch/min。通过区别起点和终点时的姿态，来控制被驱动的工具的姿态。

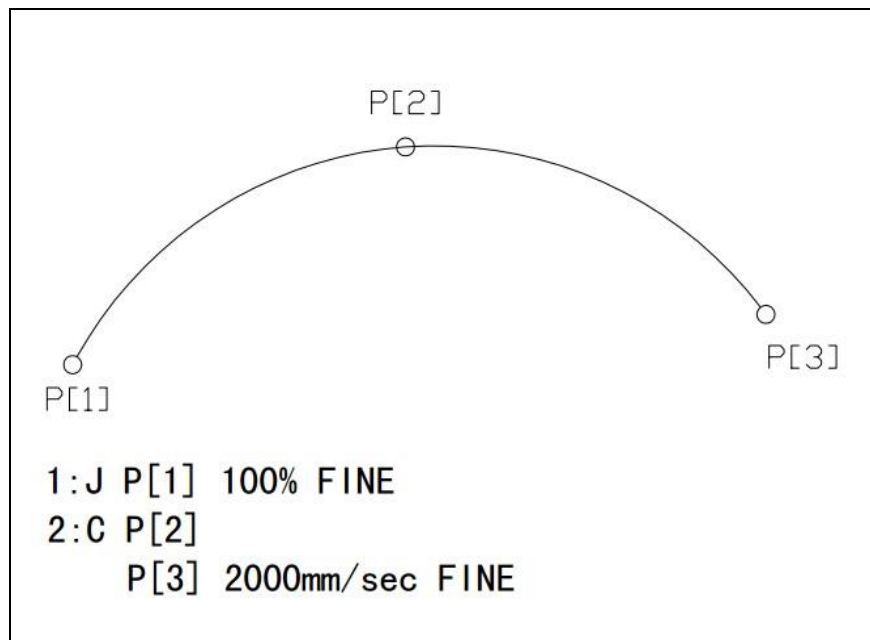


图 1-3 C 圆弧定位

图中由 P[1]点开始沿着过 P[2]点的圆弧以2000mm/sec 的速度运动至 P[3]点

1.2 位置数据

位置数据包括位置和机器人的姿态。在进行运动指令示教时，位置数据也同时被写入程序文件。

位置数据被划分为两种类型。一种是在关节坐标系下的关节坐标（J1、J2、J3、J4、J5、J6），没有姿态信息；另一种类型在直角坐标系下的位置和姿态两种信息（X、Y、Z、A、B、C）。

1.2.1 直角坐标系

直角坐标系下的位置数据包含四个元素：用户坐标系序号、工具坐标系序号、位置

/姿态和配置。

<u>UF, UT, (X, Y, Z, A, B, C), Configuration</u>				
用户坐标 系序号	工具坐标 系序号	位置	姿态	配置

图 1-4 位置数据（直角坐标）

位置 (X, Y, Z): 代表在直角坐标系下工具中心 (TCP, 即工具坐标系的原点) 的三维坐标;

姿态 (A, B, C): 代表在直角坐标系下, 绕 X 轴、Y 轴、Z 轴的角位移。

配置: 指机器人的姿态。包括关节定位和转数两种类型。

关节定位的配置: 机械腕俯仰、机械臂的上面或下面、机械臂的前面或后面。

对于关节型六轴机器人, 一般有三个奇异形位。以这三个奇异行为作为临界点, 可以定义机器人的三个关节属性 $cfg1$ 、 $cfg2$ 、 $cfg3$ 。

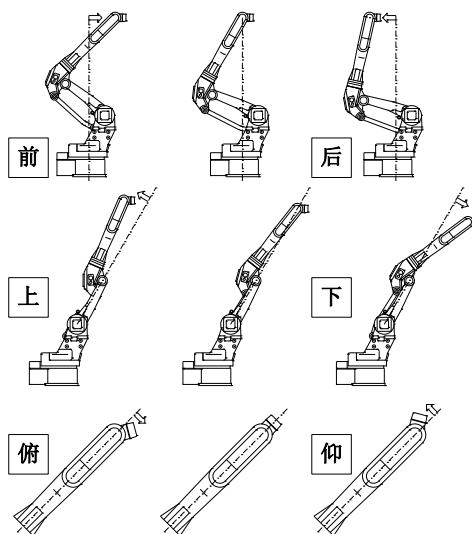


图 1-5

如图9-5图所示, 以机器人第一关节的转动轴线与第二关节的轴线平行线两者所构成的平面为参考平面, 当机器人手腕中心点位于平面两侧时, 分别定义机器人的第一个关节属性为“前” ($cfg1=0$)和“后” ($cfg1=1$); 以机器人大臂和第三关节的轴线所构成的平面作为参考平面, 当手腕中心点处于该参考平面两边时分别定义机器人的第二个关节属性为“上” ($cfg2=0$)和“下” ($cfg2=1$); 以机器人小臂和第五关节的轴向所构成的平面作为参考平面, 当手腕中心点处于该参考平面两边时分别定义机器人的第二个关节

属性为“俯”(cfg3=0)和“仰”(cfg3=1)。这三个关节属性恰好可以组合为“前下俯”、“前下仰”、“前上俯”、“前上仰”、“后上俯”、“后上仰”、“后下俯”、“后下仰” 8种关节属性的组合。本控制系统通过关节属性来确定机器人在直角坐标系下的姿态。

1.2.2 关节坐标

关节坐标系下的位置数据用每个关节的角度位置定义，关节坐标系位于每个关节的基准面。



图 1-6 位置数据（关节坐标）

1.2.3 位置变量和位置寄存器

在运动指令中，位置数据通过位置变量 (P[i]) 和位置寄存器 (PR[i]) 表示。一般情况下，使用位置变量。

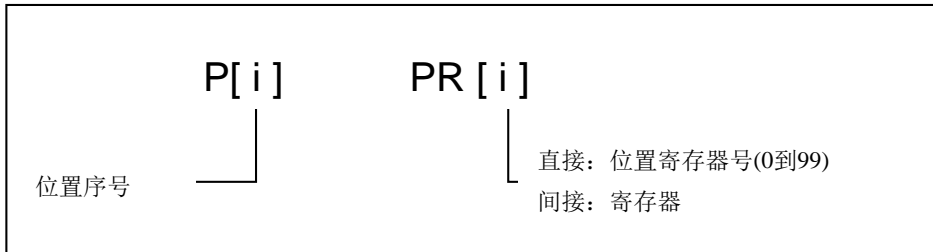


图 1-7 位置变量和位置寄存器

(1) 位置变量：是指用于保存位置数据的变量。在示教过程中，位置数据被自动保存到程序文件中，此时的坐标系均为当前所选择的，当复制指令时，位置及相关信息也一同被复制。位置变量的取值范围无限制。

(2) 位置寄存器：用于存放位置数据，类似通用寄存器 (R 寄存器) 一样使用。在菜单树窗口的“寄存器”下的“位置寄存器”中，可以对位置寄存器组号、属性等进行查看、设置和修改。

示例：

1:J P[12] 30% FINE

2:L PR[1] 300mm/sec CNT50

3:L PR[R[3]] 300mm/sec CNT50

1.3 进给速度

进给速度指机器人的运动速度。在程序执行过程中，进给速度可以通过倍率进行调节。倍率值范围为0%到150%。进给速度单位取决于动作指令类型。

J P[1] 50% FINE //机器人以最大进给速度的50%采用关节定位方式移动至 P[1]
当动作类型为关节定位时，指定最大进给速度的百分数，范围从1%到100%；

L P[1] 100mm/sec FINE//机器人以100毫米每秒的速度采用直线运动方式移动至 //P[1]

如果指定的动作类型为直线运动或者圆弧运动时，直接指定运动的速度值(mm/sec、cm/min、inch/min)，最大值由参数限制；

用寄存器指定进给速度

进给速度可以用寄存器指定。允许用户在寄存器中计算好进给速度后，再为动作指令指定进给速度。由于此时的进给速度取决于指定的寄存器，这就意味着机器人可能以一种出乎意料的速度运动，所以在使用这个功能时，在示教和操作过程中都必须小心谨慎的指定寄存器。

示例：

```
1:J P[1] R[1]% FINE
2:L P [2] R[2]mm/sec FINE
3:C P [1]
P[2] R[3]mm/sec CNT50
```

1.4 定位路径

定位路径指定了相邻轨迹间的过渡形式，有以下两种形式：

- (1) FINE：相当于准确停止。
- (2) CNT：相当于圆弧过渡，CNT后的数值为过渡误差，该数值的取值范围为0到100。CNT0等价于FINE。

FINE 定位路径: J P[i] 50% FINE

当指定 FINE 定位路径时,机器人在向下一个目标点驱动前,停止在当前目标点上。

CNT 定位路径: J P[i] 50% CNT50

当指定 CNT 定位路径时,机器人逼近一个目标点但是不停留在这个目标点上,而是向下一个目标点移动。其取值为逼近误差。例如 CNT50,表示目标 P[i]点到机器人实际运行路径的最短距离为50mm。

提示:

- (1) 示教如等待指令这样的指令时,机器人应停止在目标点上来执行该指令,即使用 FINE 定位路径。
- (2) 一些短距离、高速度的动作,使用 CNT 定位路径连续不断的执行可以减速,即使指定的 CNT 值很大时,也可以减速。

使用 CNT 和 FINE 时,机器人运动路径如图9-8所示:

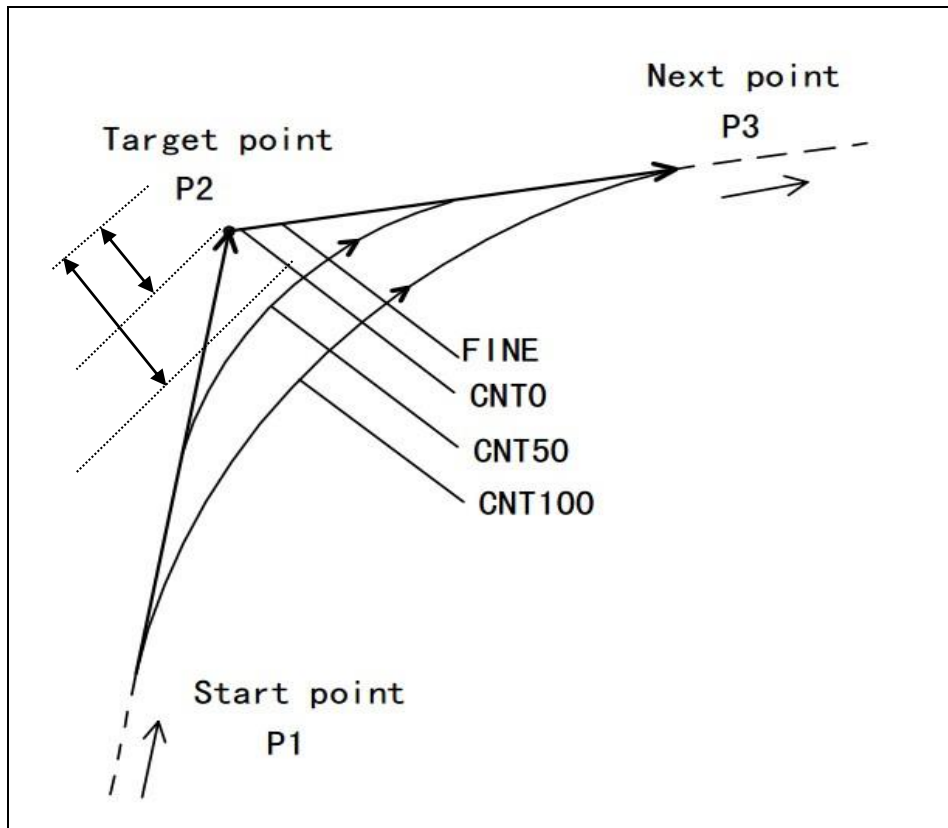


图 1-8

1.5 附加动作指令

附加动作指令让机器人完成特殊的任务，本系统目前支持的附加指令有：

- 加速倍率（ACC）
- 增量指令（INC）

1.5.1 加速倍率（ACC）

该指令指定运动过程中的加速度的倍率。ACC 后紧跟数字，表示加速度的倍率。如 ACC80，即80%的加速度。当减小加速倍率时，加速的时间会变长（加速和减速慢慢的完成）。当加速倍率提高时，加速时间就会变短（加速和减速快速的完成）。

从起点到目标点，用于执行动作的时间取决于加速倍率。加速倍率值的范围在1%到2000%之间变化。在目标位置处添加加速倍率。

加速倍率值对比如图9-9所示：

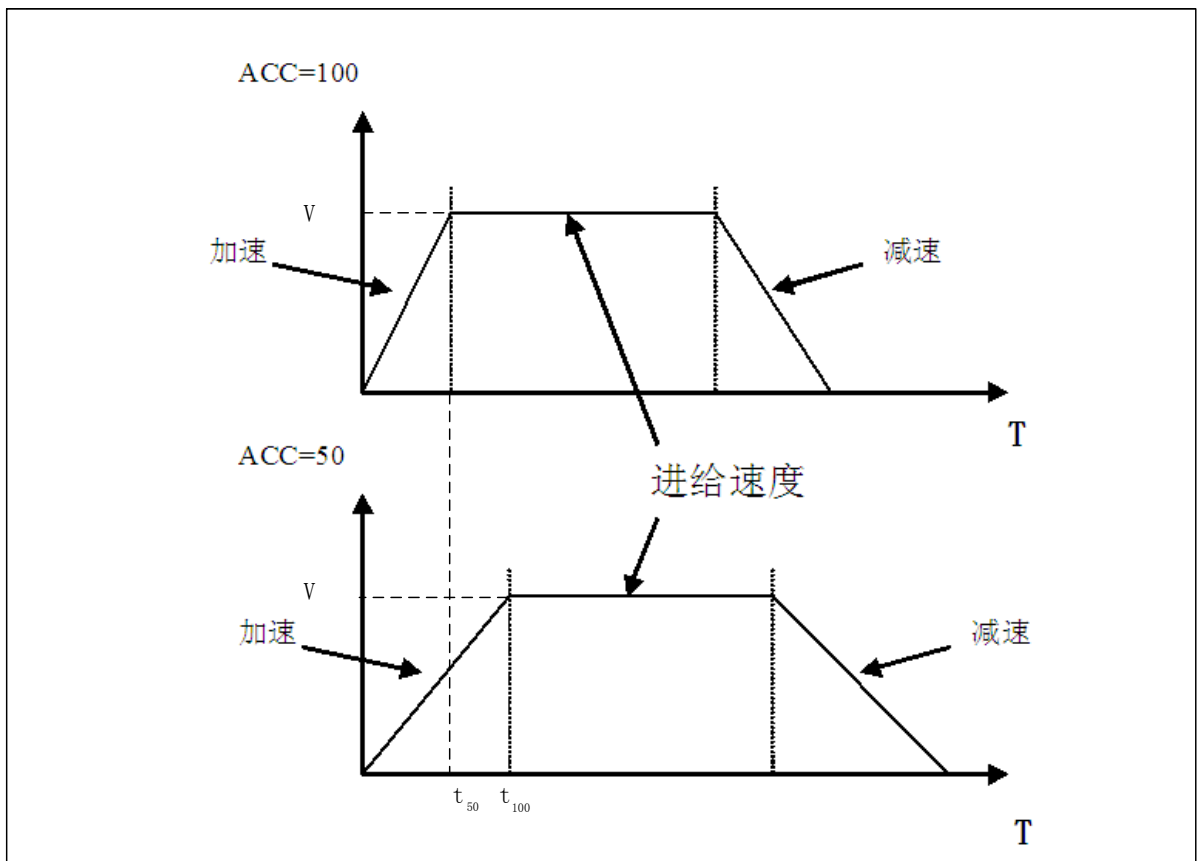


图 1-9 加速倍率值对比

图中 ACC 越大，加速达到进给速度所用时间越短

注：如果加速倍率很大，可能会发生剧烈的颤动，从而引起伺服报警。如果是因为增加了加速倍率指令而导致上述情况的发生，要么减小加速倍率值，要么删除加速倍率指令。

1.5.2 增量指令 (INC)

示例： L P[1] 500mm/sec FINE INC

增量指令将运动指令中的位置数据用作当前位置的增量，即增量指令中的位置数据为机器人移动的增量。

- 当位置数据为关节坐标值时，提供了每个轴的增量数据。
- 当位置变量 (P[i]) 作为位置数据时，用户坐标系的基准通过用户坐标系的序号指定，而用户坐标系的序号是在位置数据中指定的。
- 当位置寄存器作为位置数据时，基准坐标系即为当前用户坐标系。

2 寄存器指令

寄存器指令在寄存器上完成算术运算。本机器人控制系统支持两种寄存器指令，分别为：R 寄存器指令和位置寄存器指令。

在寄存器操作中，支持以下多项式运算：

示例：1: $R[2]=R[3]-R[4]+R[5]-R[6]$

2: $R[10]=R[2]*100/R[6]$

寄存器指令支持多项式运算，但使用时请注意：单独一行最多5个运算符。

本系统支持+、-、*、/、MOD、DIV 在一行中混合使用。在多项式中，MOD、DIV 属于同一级别，*、/属于同一级别，+、-属于同一级别，优先级别划分为：

- MOD、DIV优先级高于*、/；
- *、/优先级高于+、-；
- 同一级别下，优先级从左到右，即左边高于右边；

2.1 寄存器（R）指令

说 明

寄存器指令在寄存器上完成算术运算。寄存器是一个存储数据的变量，本机器人控制系统提供200个R 寄存器。

指令格式

1) $R[i]=(value)$

$R[i]=(value)$ 指令把数值(value)赋值给指定的R 寄存器。

其中，i 的范围是0到199，（value）可以取常数（constant）、寄存器（R）、位置寄存器中的某个轴（PR[i,j]）、数字量输入/输出（DI[i]/DO[i]）、模拟量输入/输出（AI[i]/AO[i]）。

示例：

1: $R[1] = DI[3]$

2: $R[R[4]] = AI[R[1]]$

2) R[i]=(value)+(value)

R[i]=(value)+(value)指令把两个数值的和赋值给指定的 R 寄存器。

3) R[i]=(value)-(value)

R[i]=(value)-(value)指令把两个数值的差赋值给指定的 R 寄存器。

4) R[i]=(value)*(value)

R[i]=(value)*(value)指令把两个数值的乘积赋值给指定的 R 寄存器。

5) R[i]=(value)/(value)

R[i]=(value)/(value)指令把两个数值的商赋值给指定的 R 寄存器。

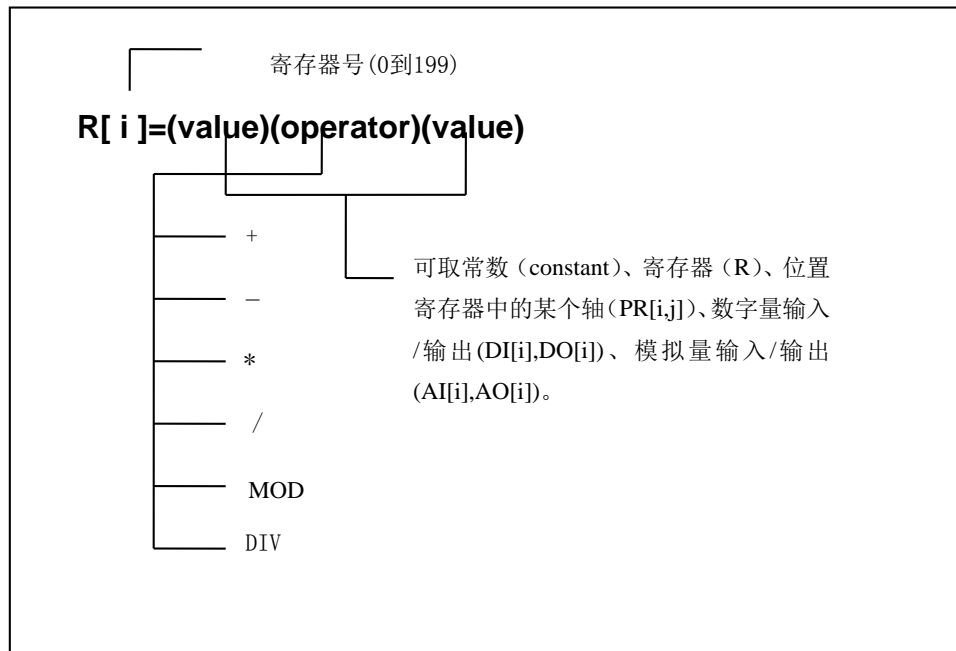
6) R[i]=(value)MOD(value)

R[i]=(value)MOD(value)指令把两个数值的商的余数（小数部分）赋值给指定的 R 寄存器。

7) R[i]=(value)DIV(value)

R[i]=(value)DIV(value)指令把两个数值的商（整数部分）赋值给指定的 R 寄存器。

对于运算寄存器指令，可归纳如下：



示例：

1: R[3] = DI[4]+PR[1,2]

2: R[R[4]] = R[1]+1

2.2 位置寄存器（PR）指令

说 明

位置寄存器指令在位置寄存器上完成算术操作。位置寄存器指令可以把位置数据、两个数值的和、差赋值给指定的位置寄存器。

位置寄存器是一个存储位置数据(x、y、z、w、p、r)的变量，本系统提供100个位置寄存器。

指令格式

1) $PR[i]=(value)$

$PR[i]=(value)$ 指令把数值(value)赋值给指定的位置寄存器。

其中，i 的范围是0到99，（value）可以取位置寄存器（PR）、位置变量（P）、直角坐标系中的当前位置（Lpos）、关节坐标系中的当前位置（Jpos）、用户坐标系（UFRAME[i]）、工具坐标系（UTOOL[i]）。

示例：

- 1: $PR[1] = Lpos$
- 2: $PR[R[4]] = UFRAME[R[1]]$
- 3: $PR[9] = UTOOL[1]$

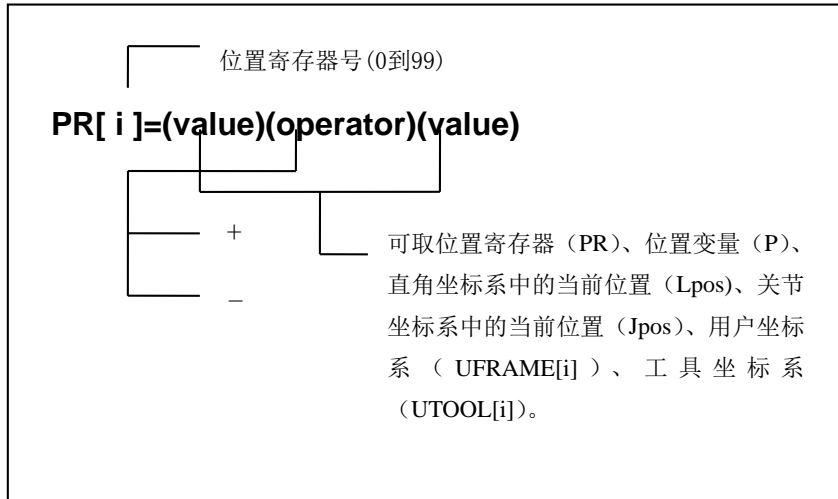
2) $PR[i]=(value)+(value)$

$PR[i]=(value)+(value)$ 指令把两个数值的和赋值给指定的位置寄存器。

3) $PR[i]=(value)-(value)$

$PR[i]=(value)-(value)$ 指令把两个数值的差赋值给指定的位置寄存器。

对于运算位置寄存器指令，可归纳如下：



示例：

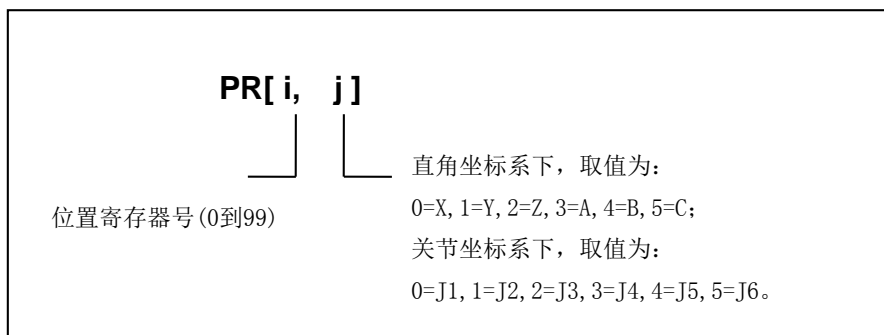
- 1: PR[3] = PR[4]+Lpos
- 2: PR[4] = PR[R[1]]

2.3 位置寄存器轴指令

说 明

位置寄存器轴指令在位置寄存器上完成计算操作。PR[i,j]中的元素 i 代表位置寄存器的序号，j 代表位置寄存器元素序号。位置寄存器轴指令可以将位置数据元素的值，或两个数据的和、差、商、余数等赋值给指定的位置寄存器元素。

PR[i, j]类型如下：



指令格式

1) PR[i, j]=(value)

PR[i, j]=(value)指令把数值(value)赋值给指定的位置寄存器元素。

其中，i 的范围是0到99，(value)可以取常数 (constant)、寄存器 (R)、位置寄

寄存器中的某个轴(PR[i,j])、位置变量中的某个轴(P[i,j])、数字量输入/输出(DI[i]/DO[i])、模拟量输入/输出(AI[i]/AO[i])。

示例:

1: PR[1, 2] = R[3]

2: PR[4, 3] = 324.5

2) PR[i,j]=(value)+(value)

PR[i,j]=(value)+(value)指令把两个数值的和赋值给指定的位置寄存器元素。

3) PR[i,j]=(value)-(value)

PR[i,j]=(value)-(value)指令把两个数值的差赋值给指定的位置寄存器元素。

4) PR[i,j]=(value)*(value)

PR[i,j]=(value)*(value)指令把两个数值的乘积赋值给指定的位置寄存器元素。

5) PR[i,j]=(value)/(value)

PR[i,j]=(value)/(value)指令把两个数值的商赋值给指定的 R 寄存器位置寄存器元素。

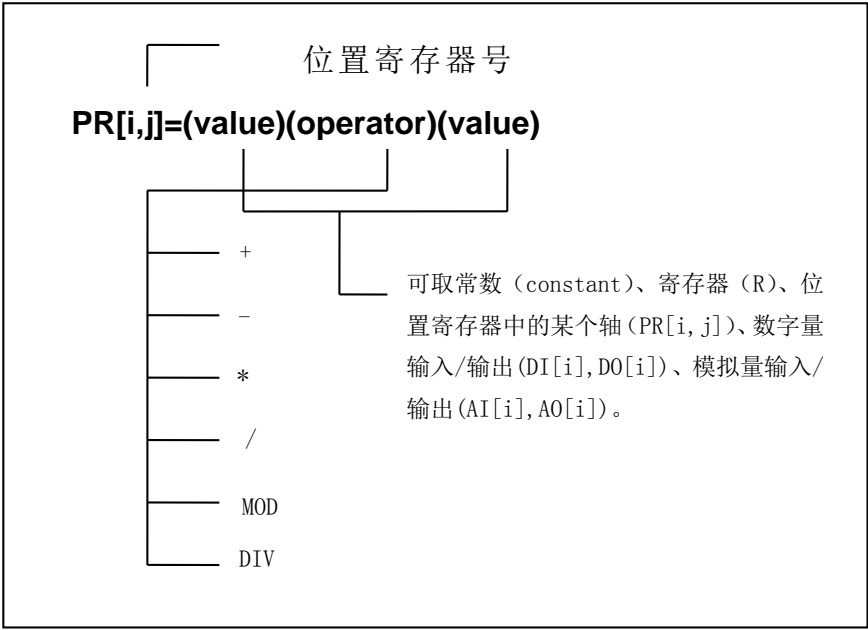
6) PR[i,j]=(value)MOD(value)

PR[i,j]=(value)MOD(value)指令把两个数值的商的余数赋值给指定的位置寄存器元素。

7) PR[i,j]=(value)DIV(value)

PR[i,j]=(value)DIV(value)指令把两个数值的商的整数赋值给指定的位置寄存器元素。

对于运算位置寄存器轴指令，可归纳如下:



示例：

1: $PR[3, 5] = R[3] + DI[4]$

2: $PR[4, 3] = PR[1, 3] - 3.528$

3 I/O 指令

输入输出指令用于操作 IO 的状态（读取输入或设置输出）。

- 数字信号输入/输出指令
- 模拟信号输入/输出指令

3.1 数字输入/输出（DI/DO）指令

说 明

数字输入指令（DI）和数字输出指令（DO）是可以被用户控制的输入输出信号。

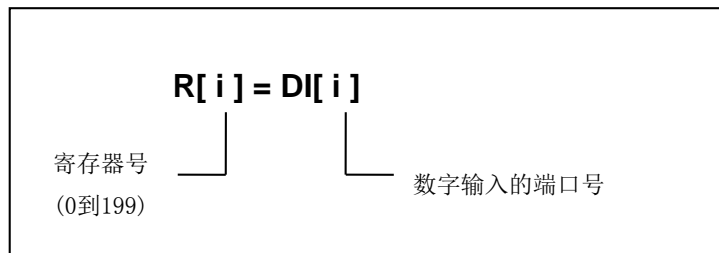
指令格式

1) 读操作

(1) $R[i]=DI[i]$

$R[i]=DI[i]$ 指令把数字输入信号（ON=1 / OFF=0）赋值给指定的 R 寄存器。

指令结构如下：



示例：

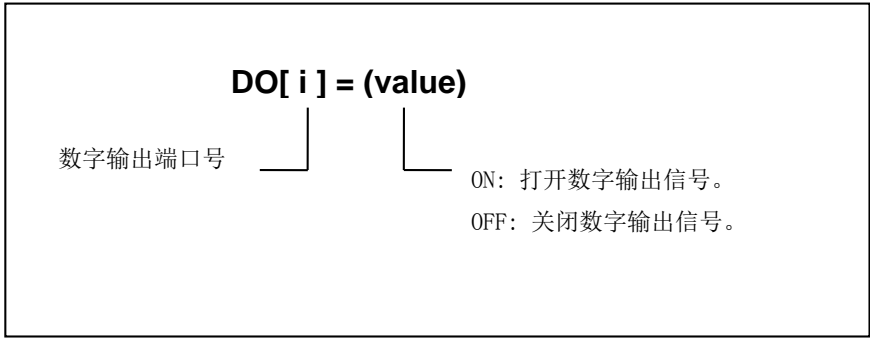
- 1: $R[1] = DI[1]$
- 2: $R[R[3]] = DI[R[4]]$

2) 写操作

(1) $DO[i]=ON/OFF$

$DO[i]=ON/OFF$ 指令把 ON=1 / OFF=0赋值给指定的数字输出信号。

指令结构如下：

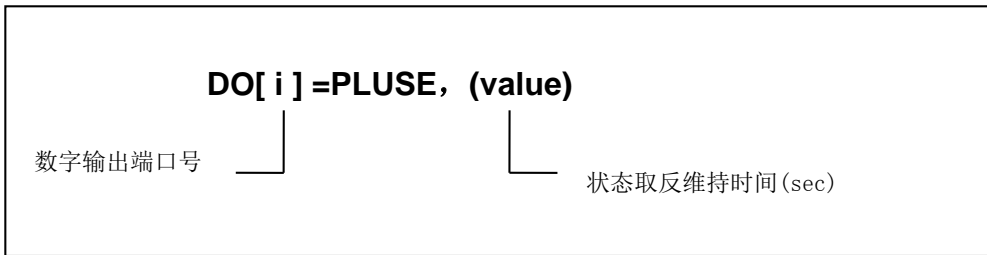


示例:

- 1: DO[1] = ON
- 2: DO[R[3]] = OFF

(2) DO[i]=PLUSE, (value)

DO[i]=PLUSE, (value)指令使 DO[i]的状态取反, 并维持一段指定时间 value。
指令结构如下:



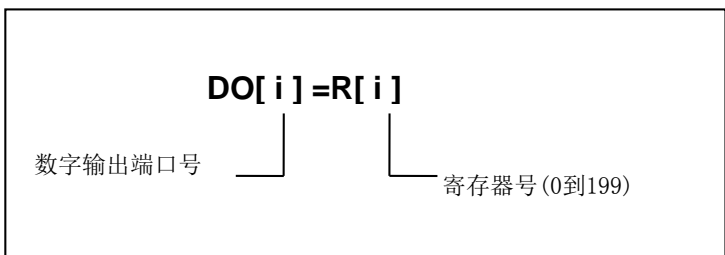
示例:

- 1: DO[2] = PULSE, 0.2sec
- 2: DO[R[3]] = PULSE, 1.2sec

(3) DO[i]=R[i]

DO[i]=R[i]指令根据指定寄存器 R 的值, 设置指定的数字输状态。当指定的 R 寄存器的值为0时, 数字输出 OFF; 当为非0, 数字输出 ON。

指令结构如下:



示例:

- 1: DO[1] = R[2]

2: DO[R[5]] = R[R[1]]

3.2 模拟输入/输出 (AI/AO) 指令

说 明

模拟量输入输出值为连续值，信号的幅值可代表温度、电压或其他数据。

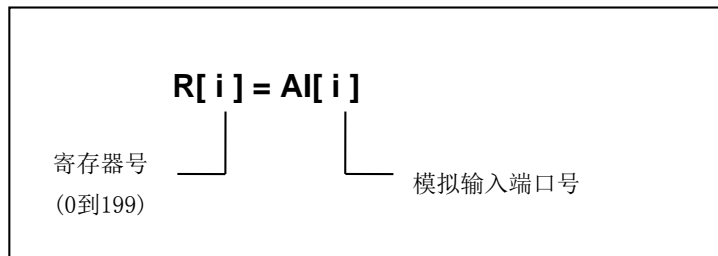
指令格式

1、读操作

(1) $R[i]=AI[i]$

$R[i]=AI[i]$ 指令将模拟输入信号赋值给指定的 R 寄存器。

指令结构如下：



示例：

1: $R[1] = AI[1]$

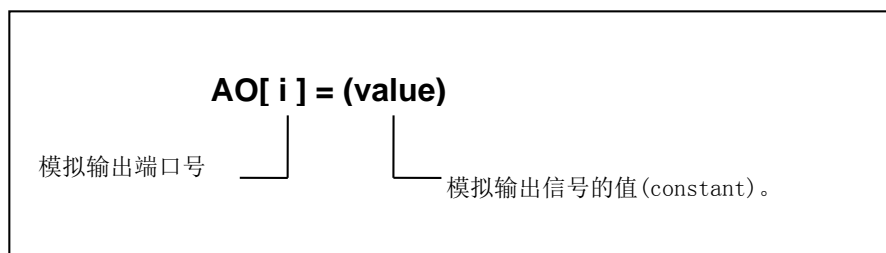
2: $R[R[3]] = AI[R[4]]$

2、写操作

(1) $AO[i]=(value)$

$AO[i]=(value)$ 指令将数值(value)作为指定的模拟输出信号的值。

指令结构如下：



示例：

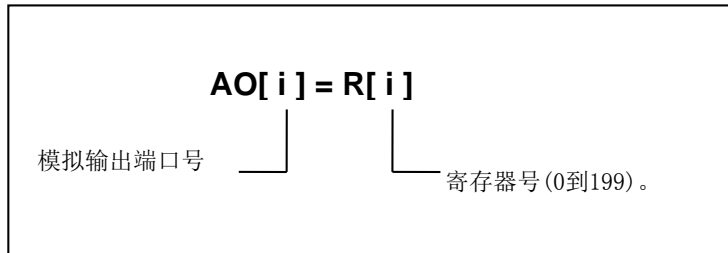
1: AO[1] = 0

2: AO[R[3]] = 3276

(2) **AO[i]=R[i]**

AO[i]=(value)指令将数值(value)作为指定的模拟输出信号的值。

指令结构如下：



示例：

1: AO[1] = R[2]

2: AO[R[5]] = R[R[1]]

4 条件指令

条件指令即条件比较指令，当某些条件满足时，在指定的标签或者程序里产生分支。条件比较指令包括寄存器条件比较指令和输入输出条件比较指令。

比较运算符包括：

- >: 大于
- >=: 大于或等于
- =: 等于
- <=: 小于或等于
- <: 小于
- <>: 不等于

对于寄存器（R）、模拟量输入/输出比较指令，可使用全部的比较符：>、>=、=、<=、<、<>；但对于数字量输入/输出比较时，只能使用=（等于）和<>（不等于）两种比较符。

4.1 寄存器条件比较指令

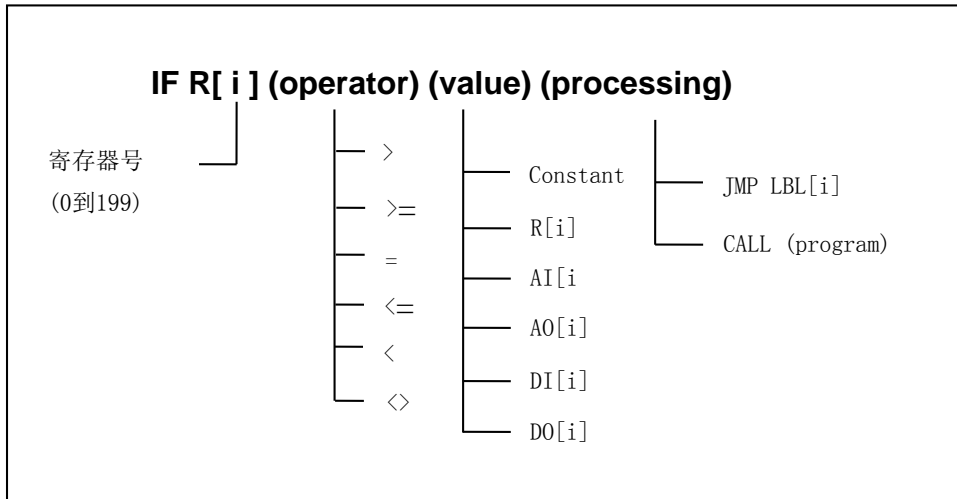
说 明

寄存器条件比较指令将存储在寄存器中的值与另一个值比较。当比较条件满足时，执行指定的操作。

指令格式

IF R[i] (运算符) (value) (操作)

指令结构如下：



示例:

- 1: IF R[1] = R[2], JMP LBL[1]
- 2: IF R[R[3]] >= 123, CALL subprog1

4.2 输入输出条件比较指令

说 明

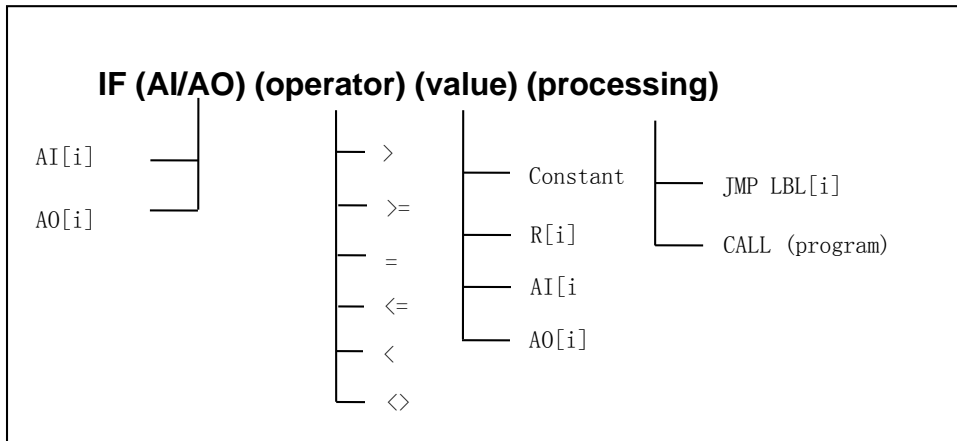
输入输出条件比较指令将输入/输出信号的值与另一个值比较。当满足比较条件时，执行指定的操作。

指令格式

(1) 模拟输入/输出条件比较指令

IF (AI/AO) (运算符) (value) (操作)

指令结构如下:



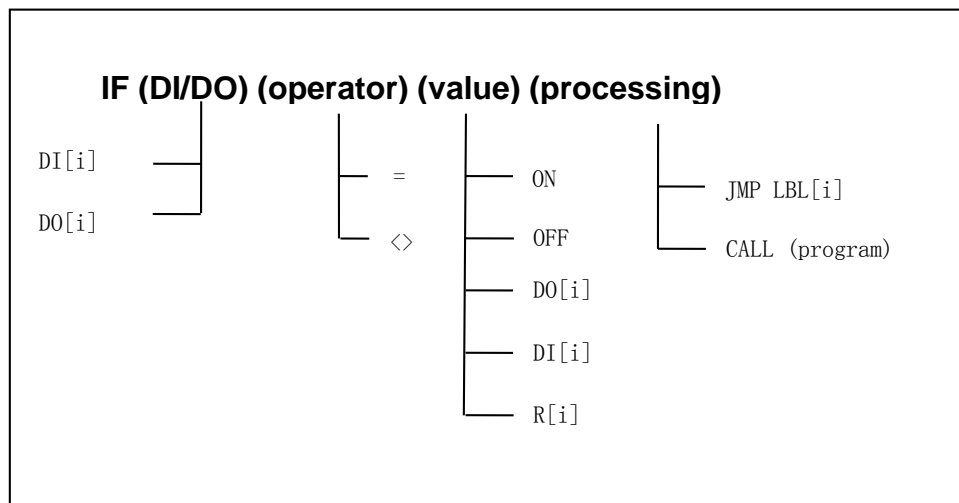
示例:

- 1: IF AO[2] >= 3000, JMP LBL[2]
- 2: IF AI[R[3]] <> R[R[2]], CALL subprog2

(2) 数字输入输出条件比较指令

IF (DI/DO) (运算符) (value) (操作)

指令结构如下:



示例:

- 1: IF DO[2] <> OFF, JMP LBL[1]
- 2: IF DI[R[3]] = R[R[4]], CALL subprog2

1.1 复合条件的使用

在比较条件中，可以使用逻辑与（AND）和逻辑或（OR）来指定复合条件。这样就简化了程序的结构，使得条件的比较更加高效。

- 逻辑与(AND)

IF <条件1> and <条件2> and <条件3>, JMP LBL [3]

- 逻辑或(OR)

IF <条件1> or <条件2>, JMP LBL [3]

如果逻辑与（AND）和逻辑或（OR）同时出现在一个指令中，逻辑就会变复杂，从而削弱了程序的可读性和可编辑性。因此，禁止在一个条件指令中同时使用 AND 和 OR。

如果在单行上为一个指令指定多个“AND”或“OR”逻辑运算，并且这些逻辑运算中有一个是从“AND”变成“OR”（或是从“OR”变成“AND”），那么其他所有的“AND”（或者“OR”也会相应的改变）。

每行中最多允许使用5个“AND”（或者“OR”）逻辑运算符。

示例： 1: IF R[1]>234 AND DO[R[2]]<>OFF AND AI[3]>=R[R[4]], JMP LBL[1]

2: IF DO[2]<>OFF OR AO[R[3]]>=R[4], CALL subprog1

2 等待指令

等待指令用于在一个指定的时间段内，或者直到某个条件的满足时的时间段内，结束程序的指令，等待指令包括如下两种：

- 指定时间的等待指令：等待一个指定的时间后，再执行后续程序；
- 条件等待指令：等待指定的条件满足后，再执行后续程序。

2.1 指定时间的等待指令

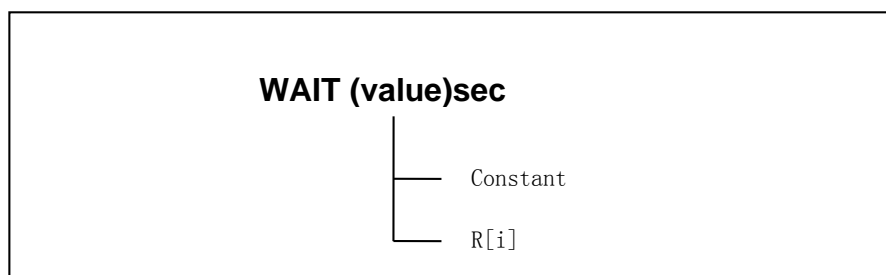
说 明

指定时间的等待指令，等待一个指定的时间（以秒为单位）后，再执行后续程序。

指令格式

WAIT (value)sec

指令结构如下：



示例：

- 1: WAIT 10.5sec
- 2: WAIT R[1]sec

2.2 条件等待指令

条件等待指令，等待程序的执行，直到指定的条件满足或者经过了指定的时间段。有以下两种指定暂停程序的方法：

- 如果没有指定操作（processing），程序将无限期等待，直到满足指定条件为止；
- 如果指定了操作（如TIMEOUT LBL[i]），当指定的条件不满足，且等待超时后，程序将跳转到指定的目标处运行。超时等待时间由系统参数设置。

2.2.1 寄存器条件等待指令

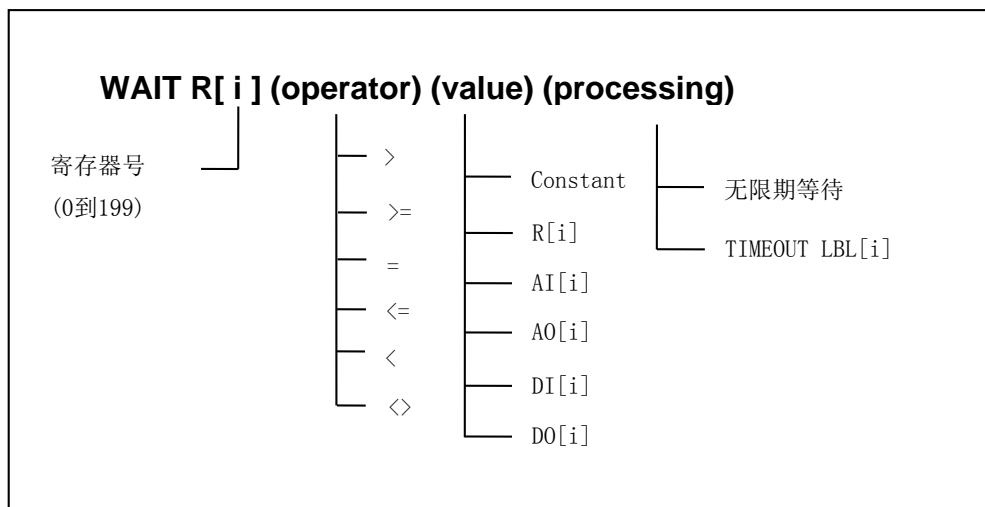
说 明

寄存器条件等待指令将 R 寄存器的值与另外一个值进行比较，并等待直到满足比较条件为止。

指令格式

WAIT R[i] (比较符) (value) (操作)

指令结构如下：



示例：

- 1: WAIT R[2] <> 1, TIMEOUT LBL[1]
- 2: WAIT R[R[1]] >= R[3]

2.2.2 输入/输出条件等待指令

说 明

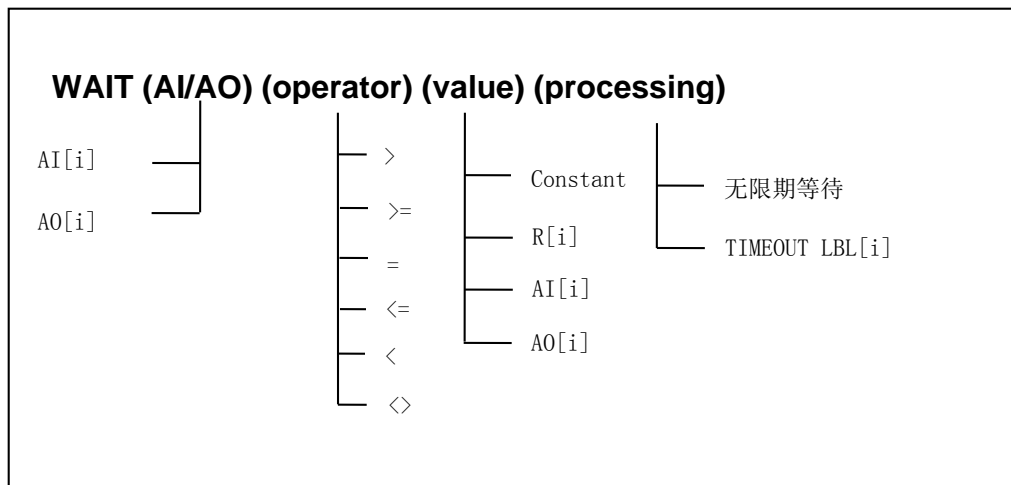
输入/输出条件等待指令将输入/输出信号的值与另一个值进行比较，并等待直到满足比较条件为止。

指令格式

(1) 模拟输入输出条件等待指令

WAIT (AI/AO) (比较符) (value) (操作)

指令结构如下：



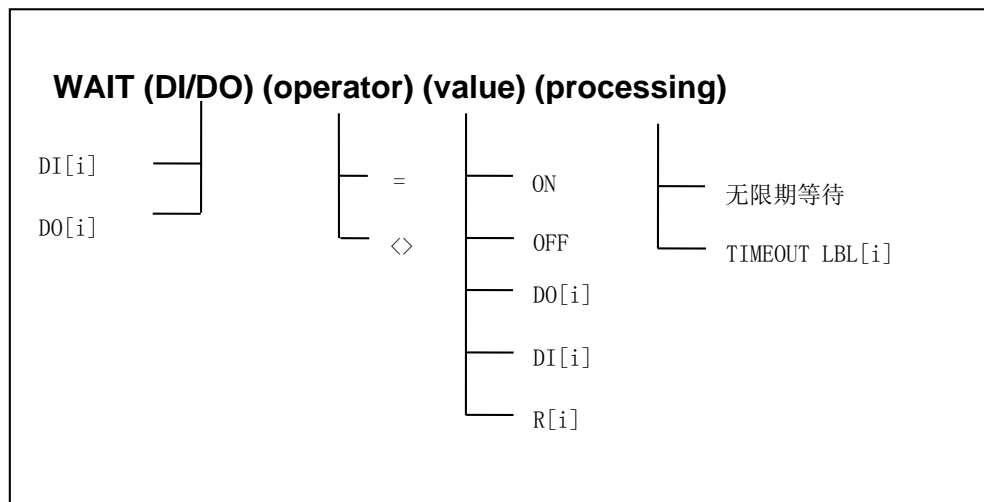
示例：

- 1: WAIT AI[2] <> 1, TIMEOUT LBL[1]
- 2: WAIT AO[R[1]] >= R[3]

(2) 数字输入输出条件比较指令

WAIT (DI/DO) (比较符) (value) (操作)

指令结构如下：



示例：

- 1: WAIT DI[2] <> ON, TIMEOUT LBL[1]
- 2: WAIT DO[R[1]] = R[3]

3 流程控制指令

流程控制指令用来控制程序的执行顺序，控制程序从当前行跳转到指定行去执行，流程控制指令包括以下几种指令：

- 标签指令；
- 程序结束指令；
- 无条件跳转指令；
- 子程序调用指令。

3.1 标签指令

说 明

标签指令用于指定程序执行的分支跳转的目标。

标签一经执行，对于条件指令、等待指令和无条件跳转指令都是适用的。不能把标签序号指定为间接寻址（如 `LBL[R[1]]`）。

指令格式

LBL[i]

指令结构如下：



示例：

1: LBL[2]

3.2 程序结束指令

说 明

程序结束指令标志着一个程序的结束。通过这个指令终止程序的执行。如果该程序是被其他的主程序调用，则控制该子程序返回到主程序中。

程序结束指令在新建程序时，系统已自动添加到程序文件的末尾，无需用户自己添加。

指令格式

END

示例：

1:END

3.3 无条件跳转指令

说 明

无条件跳转指令是指在同一个程序中，无条件的从程序的一行跳转到另一行去执行，即将程序控制转移到指定的标签。

指令格式

JMP LBL[i]

指令结构如下：



示例：

1: JMP LBL[2]

3.4 子程序调用指令

说 明

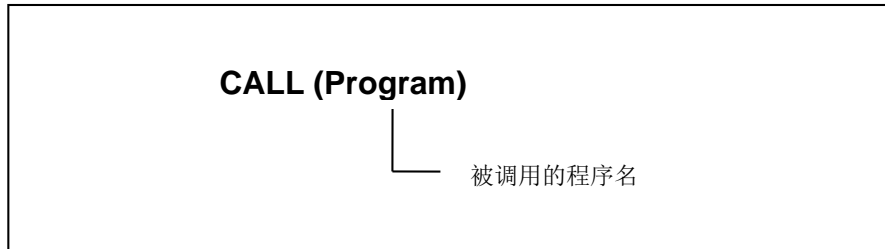
子程序调用指令将程序控制转移到另一个程序(子程序)的第一行，并执行子程序。当子程序执行到程序结束指令(END)时，控制会迅速的返回到调用程序(主程序)中的子程序调用指令的下一条指令，继续向后执行。

需要选择子程序名，或直接新建一个子程序，详细操作请参照前面章节的操作说明。

指令格式

CALL (子程序名)

指令结构如下：



示例：

1: CALL SUB1

4 码垛指令

码垛指令有以下几种：

指令	指令功能说明
码垛（开始）指令	码垛开始指令，用于得到码垛工艺号。
码垛动作指令	用于得到码垛趋近点、堆垛点和回退点数据的码垛专用动作指令
码垛结束指令	使码垛寄存器的值加减
码垛寄存器指令	用于码垛的控制

表 4-1 码垛指令类型

4.1 码垛开始指令

码垛开始指令是用于提供当前码垛所使用的码垛工艺号。根据码垛工艺号得到相应的码垛模式、路径数据，用于计算码垛运动指令中的趋近、堆垛和回退点位置。码垛开始指令格式如下图所示：



图 4-1 码垛开始指令格式

例1：

PALLET[1] 选择码垛工艺1

4.2 码垛动作指令

码垛动作指令是以码垛的趋近点、堆垛点和回退点作为运动目标点的动作指令，是码垛的专用动作指令。每一个堆垛点最多有一个趋近点和回退点，而且根据堆垛点的变化，相应的趋近和回退点的位置数据也会发生变化。

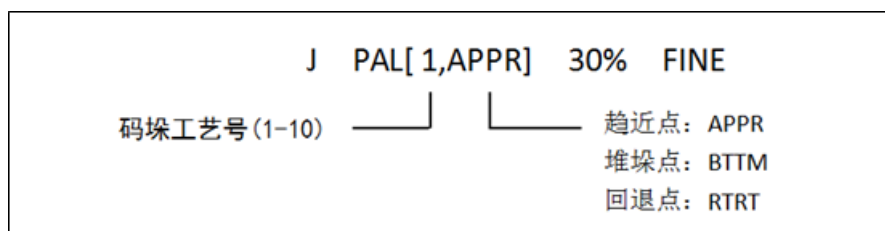


图 4-2 码垛动作指令格式

例2:

```
L PAL[ 1,APPR] 300mm/sec FINE
J PAL[ 1,BTTM] 30% CNT50
L PAL[ 1,RTRT] 300mm/sec FINE
```

4.3 码垛结束指令

码垛结束指令,计算相应的码垛工艺内下一个堆垛点所处的位置(几行几列),并改写 PL 寄存器的值。

PALLET_END [i]
└── 码垛工艺号(1-10)

图 4-3 码垛结束指令格式

例3:

```
PALLET_END [ 2 ]
```

4.4 码垛寄存器指令

码垛寄存器指令,用于码垛的控制,进行堆垛点的指定、比较等。

PL [i] = [m, n]

码垛工艺号(1-10) └── └── m, n 为 PL 寄存器要素

图 4-4 码垛寄存器指令格式

例4:

PL[1]=[2,3] 赋值指令将 PL 寄存器的第1和第二位分别赋值为2和3
IF PL[1]=[1,1],JMP LAB[2] 比较指令,如果 PL 寄存器的第1和2位为1,1,跳转到 LAB[2]

4.5 码垛程序示例

1: LBL[1]	
2: J P[1] 50% FINE	到达取件点
3: PALLET[1]	选择码垛工艺1
4: J PAL[1,APPR] 50% FINE	到达趋近点
5: L PAL[1,BTTM] 600mm/sec FINE	达到堆垛点
6: J PAL[1,RTRT] 50% FINE	到达回退点
7: PALLET_END[1]	寄存器自加
8: IF PL[1] = [1,1],JMP LBL[2]	判断下一个码垛点是否为第1行 第1个工件，若不是则继续运行， 跳转到 LBL[1]继续循环；若是则 跳转到 LBL[2]，结束循环
9: JMP LBL[1]	
10: LBL[2]	
11: J P[3] 30% FINE	
12: END	

5 其他指令

将上面未介绍的指令，归结为其他指令，这些指令包括：

- 坐标系指令；
- 用户报警指令；
- 倍率指令；
- 注释指令；
- 信息指令；
- 预读指令。

5.1 坐标系指令

坐标系指令用于改变机器人当前工作所使用的坐标系的设置。有以下两种坐标系指令：

- 坐标系设置指令—改变指定的坐标系的定义；
- 坐标系选择指令—改变当前选择的坐标系的序号。

5.1.1 坐标系设置指令

说 明

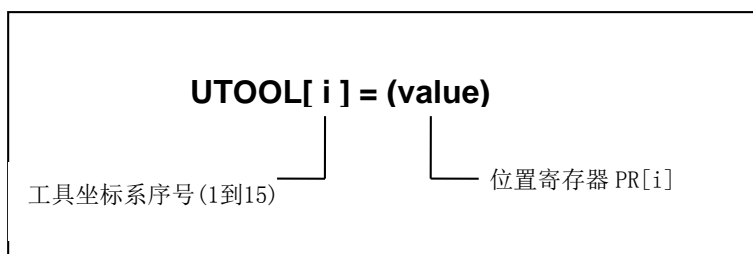
- 工具坐标系设置指令，改变由工具坐标系序号指定的工具坐标系的设置；
- 工件坐标系设置指令，改变由工件坐标系序号指定的工件坐标系的设置。

指令格式

(1) 工具坐标系设置指令

UTOOL[i]=(value)

指令结构如下：



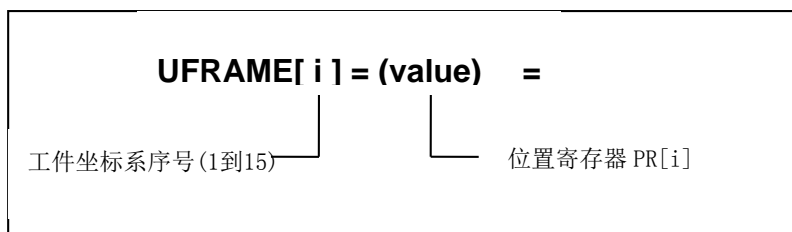
示例：

- 1: UTOOL[1] = PR[1]
- 2: UTOOL[R[2]] = PR[R[3]]

(2) 工件坐标系设置指令

UFRAME[i]=(value)

指令结构如下：



示例：

- 1: UFRAME[1] = PR[1]
- 2: UFRAME[R[2]] = PR[R[3]]

5.1.2 坐标系选择指令

说 明

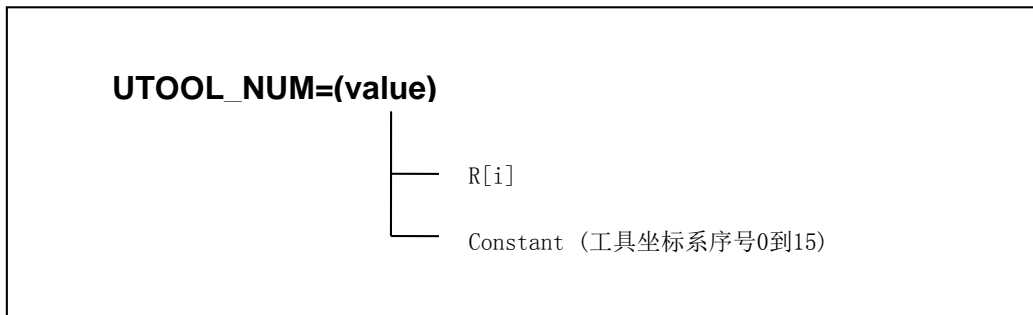
- 工具坐标系选择指令，改变当前工具坐标系的序号；
- 工件坐标系选择指令，改变当前工件坐标系的序号。

指令格式

(1) 工具坐标系选择指令

UTOOL_NUM=(value)

指令结构如下：



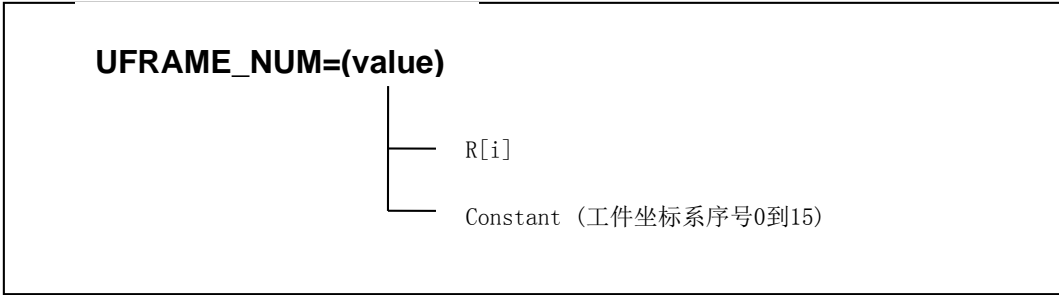
示例：

- 1: UTOOL_NUM = 1
- 2: J P[1] 100% FINE
- 3: L P[2] 500mm/sec FINE
- 4: UTOOL_NUM = R[2]
- 5: L P[3] 500mm/sec FINE
- 6: L P[4] 500mm/sec FINE

(2) 工件坐标系选择指令

UFRAME_NUM=(value)

指令结构如下：



示例：

- 1: UFRAME_NUM = 1
- 2: J P[1] 100% FINE
- 3: L P[2] 500mm/sec FINE
- 4: UFRAME_NUM = R[2]
- 5: L P[3] 500mm/sec FINE
- 6: L P[4] 500mm/sec FINE

5.2 用户报警指令

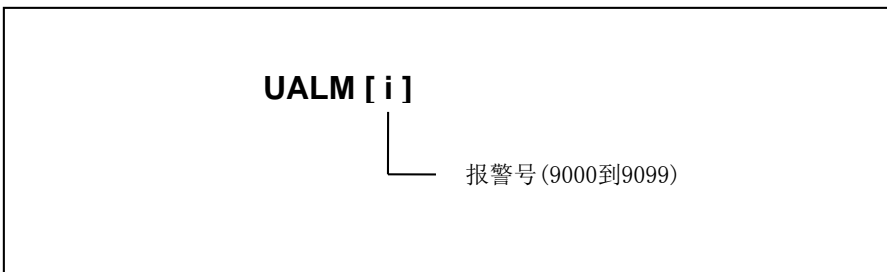
说 明

用户报警指令显示报警信息，并中止程序的运行。报警信息在系统信息的用户报警界面中设置。

指令格式

UALM[i]

指令结构如下：



示例：

- 1: UALM[9004]
- 2: UALM[R[2]]

5.3 倍率指令

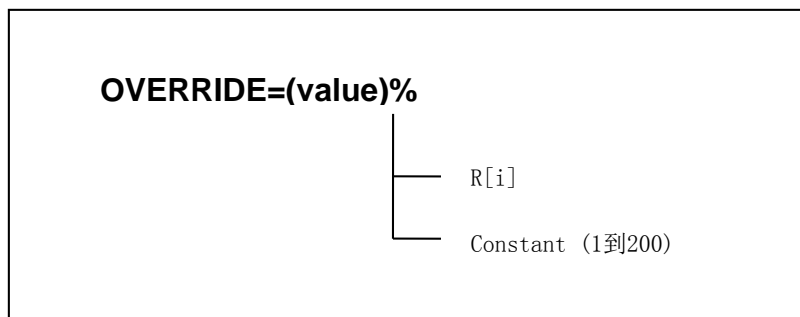
说 明

倍率指令改变进给速度倍率。

指令格式

OVERRIDE = (value)%

指令结构如下：



示例：

1: OVERRIDE = 100%

5.4 注释指令

说 明

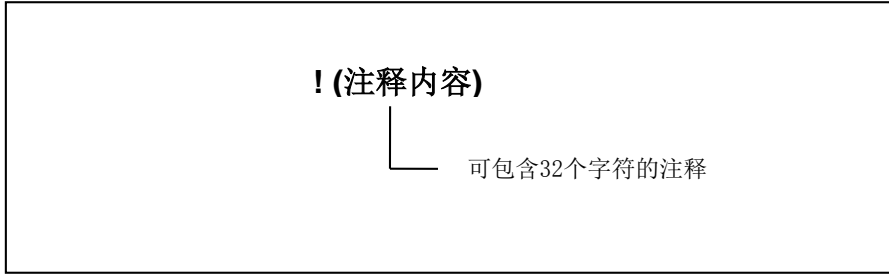
注释指令在程序中添加注释。注释不影响程序的执行。在注释指令中指定注释，一条注释最多可以包含32个字符。

注释指令在程序选择输入界面中的表现形式是 **COMMENT**(注释)，选中并输入到示教界面后，示教界面显示为“！”，详见前面章节的操作说明。

指令格式

!(注释内容)

指令结构如下：



示例：

1: ! APPROACH POSITION

5.5 信息指令

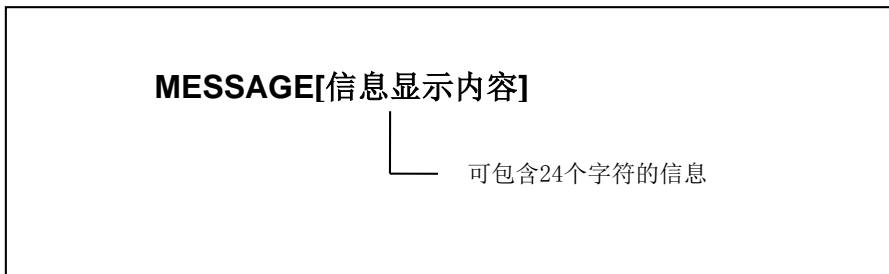
说 明

信息指令即在用户界面上显示指定的信息。一条信息最多可以包含24个字符。

指令格式

MESSAGE [信息显示内容]

指令结构如下：



示例：

1: MESSAGE[DI[1] NOT INPUT]

5.6 预读指令

说 明

预读指令即设置开启或关闭超前预读功能。

指令格式

LOOKAHEAD=ON/OFF

指令结构如下：



示例：

1: LOOKAHEAD=OFF

超前预读功能通常配合运动指令中的定位路径方式 **CNT** 一起使用，当预读功能开启时，系统提前读取将要执行的程序行中的 **CNT** 指令，从而实现多点连续运动式的圆弧过渡。预读功能关闭后，**CNT** 方式不再生效，实际效果等同于 **FINE** 方式。

5 附录：程序报警定义

报警号	级别	报警定义	备注
2000	3	急停	
2100	3	正解出错	
2200	3	反解出错	
3100~3108	2	轴正向硬限位	按轴号递加，即： 3100: J1 轴正向限位 3101: J2 轴正向限位
3200~3208	2	轴负向硬限位	按轴号递加，同上。
3300~3308	2	轴正向软限位	按轴号递加，同上。
3400~3408	2	轴负向软限位	按轴号递加，同上。
3500~3508	3	轴跟踪误差过大	按轴号递加，同上。
3600~3608	3	轴超速	按轴号递加，同上。
3700~3708	2	轴绝对位置初始化失败	按轴号递加，即： 3700: J1 轴绝对位置初始化失败 3701: J2 轴绝对位置初始化失败
3800	2	引弧失败	
4000~4999	3	HSV-18 伺服驱动报警(总线型专用)	小循环按轴号递加(9 轴,0~8)，十位百位为驱动器报警号，如： 4203: J4 轴伺服驱动 20 号报警
5100	2	分配内存失败	
5200	2	加载程序失败	
5300	2	语法匹配失败	
5400	2	指令格式错	
5500	2	位置变量(P)地址无效	位置变量(P)地址最大 999
5600	2	位置寄存器(PR)地址无效	位置寄存器(PR)地址最大 99
5700	2	寄存器(R)地址无效	寄存器(R)地址最大 199
5800	2	寄存器(AR)地址无效	参数寄存器(AR)地址最大 9
5900	2	缺少位置参数	移动指令(J、L、C)未指定位置参数
6000	2	缺少速度参数	移动指令(J、L、C)未指定速度参数
6100	2	缺少数据单位	缺少单位，如 mm/sec
6200	2	缺少轨迹过渡类型	移动指令(J、L、C)缺少过渡类型(FINE/CNT)
6300	2	跳转目标无效	
6400	2	子程序不存在	
6500	2	子程序嵌套层次过多	最多允许嵌套调用 10 层

6600	2	缺少执行语句	IF 指令缺少执行语句 JMP 或 CALL
6700	2	参数无效	CALL 指令传递的参数(AR)无效, 仅支持字符串型和数型参数
6800	2	参数过多	CALL 指令最多可传递 10 个参数 (AR)
6900	2	除零	
7000	2	数据类型不一致	位置寄存器指令操作数类型不一致, 应统一为关节位置或直角坐标位置
7100	2	用户报警号无效	用户报警号范围 9000-9099, 超出此范围无效
7200	2	数字 IO 地址无效	地址超出允许范围
7300	2	模拟量 IO 地址无效	地址超出允许范围
7700	2	无效运动 (平板示教器专用)	直角坐标下运动的前后两个示教点采用了不同的关节属性, 机器人无法生成轨迹