

机器人培训教材



绪言

这章主要介绍了本手册的规划和在操作 FANUC 机器人时必须遵循的安全防护措施。

本章目录

1.1 手册规划

1.2 工作人员

1.3 一般安全防护

1.4 安全防护

1.1 手册规划

FANUC 系列机器人 (R--J3i MODEL B 控制器) 操作工具使用手册。

.这个手册描述了怎样操作多功能集成机器人——FANUC机器人。机器人是由FANUC R--J3i MODEL B控制器（下文都称作机器人控制器）控制的，并使用操作工具软件。本手册描述了以下项目，用来使用机器人操作工件。

- 设置系统为电弧焊接状态
- 设置系统为操作工件状态
- 操作机器人
- 创建和改变一个程序
- 执行一个程序
- 状态指示
- 警报代码和系统参数

手册的使用

手册的每一章介绍了机器人的一个操作，使用者可以根据需要的操作，选择和阅读不同章节的内容。如果使用者从头到尾阅读一次本手册，大约需要5到六个小时就可以理解手册上所有的信息。

第一章 绪言	主要介绍了本手册的规划和在操作FANUC机器人时必须遵循的安全防护措施。所有使用者必须阅读安全防护措施。
第二章 概述	介绍了机器人的基础知识，描述了机器人和操作工件电弧焊系统的基本设置。
第三章 设置操作工件电 弧焊系统	描述了设置操作工件电弧焊系统的步骤，包括输入输出，坐标系统和参考位置。
第四章 程序结构	介绍了程序结构和程序指令的语法。
第五章 创建程序	描述了怎样设计、创建、改变、删除和复制程序。这章还介绍了打开电源和通过手动进给运行机器人的程序。
第六章 执行程序	描述了怎样执行和停止一个程序。这里包含了测试操作、自动操作和从警报状态恢复的方法。
第七章 状态指示	介绍了怎样用状态指示LED灯检查机器人操作状态。
第八章 文件的输入输出	介绍了怎样存储、读取和打印一个程序文件或者系统文件。

认证

为了确认软件的版本和订购的文件，请阅读以下表格：

需要检验的项目	位置
软件版本	B.3 软件版本
您软件的订购号	A.1 菜单列表

标记

为防止人身伤害和财产损失，本手册介绍了安全保护措施。这些措施根据不同的重要级别，被贴上诸如“Warning”和“Caution”的标签，并在“Note”下给出辅助的解释。在开始使用机器人前，请仔细阅读“Warning” “Caution” 和“Note”。



警告：

不按照“Warning”下的指示行事，会导致使用者致命的或者严重的伤害。本信息在一个

说明：在“Note”下的信息是辅助解释，不是警告和注意。



注意：

不按照“Caution”下的指示行事，会导致使用者致命的或者严重的伤害。本信息在一个方框内表示，这样使其在手册的主体中显得醒目。

请仔细阅读并妥善保存本手册。

1.2 工作人员

机器人独自状态下，不能工作。只有当它装备了机械手臂或者其他设备，并且连接到外围设备，组成一个系统以后，机器人才能运转。

要不仅仅考虑到机器人的安全，而且要保证整个系统的安全。使用机器人时，需要提供安全护栏和采用其他的安全措施。**FANUC** 机器人需要如下的系统工作人员。请选择合适的工人进行专门机器人教程的培训。

普通操作者

操作者的工作是：

- 打开和关闭系统
- 开始和停止机器人程序
- 从警报状态恢复系统

禁止操作者进入由安全护栏封闭的区域进行相应操作。

程序员或者教导操作者

程序员或者教导操作者的任务包括了普通操作员和如下条款的内容：

- 教导机器人、调整外围设备和其他必须在由安全护栏封闭的区域里进行的工作。
- 程序员或者教导操作者必须接受专门的机器人课程的培训。

维护工程师

维护工程师的工作包括程序员的工作和如下的内容：

- 修理和维护机器人

维护工程师必须接受专门的机器人课程的培训。

1.3 一般安全防护

本节列举了一般安全防护措施。在开始使用机器人前，请阅读本条例。手册在随后会指出其他的防护措施。请注意每一个条款。

一般规则

警告：使用机器人时，需要采取以下防护措施。否则，机器人和外围设备会受到不利影响，或者工人会受严重伤害。

- 禁止在易燃环境中使用机器人
- 禁止在易爆环境中使用机器人
- 禁止在放射性环境中使用机器人
- 禁止在水中或者十分潮湿的环境中使用机器人
- 禁止使用机器人搭载人或者动物。
- 禁止利用机器人作为梯子。（不要攀爬或者从机器人上悬挂）

警告：机器人工作人员必须穿戴如下安全用品：

- 根据每一个工作穿着合适的衣服
- 安全工作鞋
- 安全头盔

说明：程序员和维护人员必须在FANUC接受适当的培训。

安装注意事项

警告：运输和安装机器人，应该严格的按照FANUC建议的程序进行。错误的运输和安装可能会导致机器人滑落，对工人造成严重的伤害。

注意：机器人安装后的第一次运行，应该限制在低速运转。然后，再逐渐增加速度，检查机器人的运行情况。

运行注意事项

警告：在机器人运行前，应该确认没有人在安全护栏区内。同时，要检查确信不存在危险位置的风险。如果检测到这样的位置，应该在运行前消除隐患。

注意：操作者在操作控制面板或者教导盒时不能戴手套。佩戴手套可能会引起操作失误。

说明：可以保存程序、系统参数和其他信息到存储卡或者磁盘。确定周期性的存储数据，以免在事故中造成数据丢失

编程注意事项

警告：应该在安全护栏区域外，尽可能远的地方编程。如果程序需要在安全护栏区域内完成，程序员应该遵守如下事项：

- 在进入安全护栏区域内前，确认在区域中没有危险位置的风险。
- 随着准备按紧急停机按钮。
- 机器人应该在低速运行。
- 程序运行前，检查整个系统状态，确认没有到外围设备的远程指令，确认没有动作对使用者有威胁。

注意：编程结束后，根据制定的步骤，给出相应的文字说明。（参见本手册中程序执行的章节）。在文字说明中，工人必须在安全护栏区域外。

说明：程序员应该在FANUC接受适当的培训。

维护注意事项

警告：在维护中，机器人和系统应该在断电状态。如果系统和机器人在电源开的状态下，一个维护操作可能会引起冲击性的危害。如果必要，应该提供相应的安全锁，以防止其他人打开机器人或者系统的电源。如果维护需要在电源开的状态下进行，紧急停机按钮需要保证可用，

警告：替换零件时，维护工人应该阅读维护手册，预先学习复位步骤。如果进行了错误的步骤，可能会引起事故，导致对机器人的损伤并伤害到工人。

警告：当要进入到由安全护栏封闭的区域时，维护工人应该检查整个系统，确认没有危险位置。在存在危险位置时，如果工人需要进入这个区域，工人必须时刻保持十分小心，检查当前系统状态。

警告：零件的更换应该按照FANUC的建议执行。如果应用其他零件，可能会发生故障或者损害。特别指出的是，请不要应用FANUC没有推荐的保险丝。这样的保险丝可能会导致火灾。

警告：当移除一个马达或者制动器时，应该用起重机或者其他设备预先支撑机器人的手臂，这样在装卸中，手臂不会坠落。

警告：如果在维护中需要让一个机器人进行一个动作，应该采取以下的保护措施：

- 维护工作中，使逃离通道顺畅，时刻检查整个系统的动作，以便通道不被机器人或者外围设备阻挡。
- 时刻注意危险位置的威胁，做好随时按紧急停机按钮的准备。

警告：当拧动马达、制动器或者其他重的负载时，应该用起重机或者其他装备保护维护工人不承受过量负载。否则，维护工人可能会受到严重伤害。

注意：只要有油脂洒落在地面，就应该尽快擦除以避免危险的滑倒。

注意：在维护过程中，不要攀爬机器人。这样的尝试会导致对机器人的不利的影响。此外，失足会导致对维护工人的伤害。

注意：以下零件会发热。如果维护工人需要接触发热的零件，需要佩戴放热手套或者用保护性的工具：

- 伺服马达
- 控制单元内部

注意：当更换了一个零件后，所有的螺钉和其他相关的组成部分应该放回各自原来的地方。必须做仔细的检查，确保没有组件丢失或者未被安装。

注意：在对气动系统的维护进行前，应该关闭供压系统并且排放管道内的气体使气压降至零。

注意：在零件被更换后，根据预先设定的方法，应该对机器人给出相应的文字说明。（参见本手册程序执行章节）。在制作文字说明时，维护人员应该工作在安全护栏外。

注意：维护结束后，机器人周围和安全护栏区域内的地板，应该清理干净洒落的油脂或者水、金属片等。

注意：替换零件时，小心防止灰尘进入机器人。

说明：每个维护工人或者检查工人应该在FANUC接受适当的训练。

说明：应该在适当的光线下进行维护。必须十分小心不让因光线导致任何危险。

说明：应该对机器人进行周期性的检查。（指人工维护）不正确执行周期性的维护，会对机器人造成不利的影响或者影响机器人的服役时间，同时会导致事故。

1.4 安全防护措施

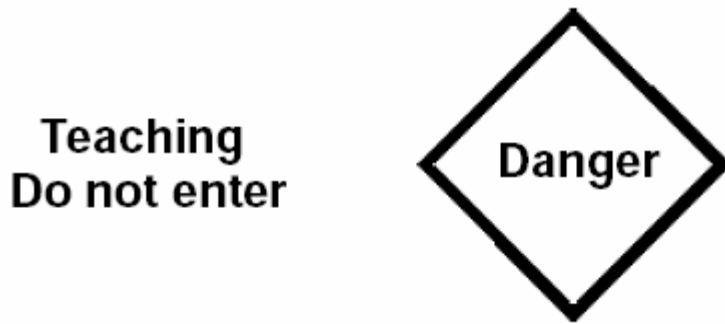
安全防护

和其他普通自动机械不同，机器人有机械手臂和腕关节，可以在所有的操作空间移动。机器人很灵活，同时也很危险。机器人通常都和包含了自动化系统的外围设备连接。使用者必须对整个系统采取安全防护措施。安全防护措施介绍如下：

安装和布线的安全措施

- 用警告灯和其他物品显示机器人正在工作。

图1--1. 警告指示



- 在系统周围设置有安全门的保护性的围栏，这样仅仅操作者才能通过这道门进入操作区。设计使系统使门打开时，系统停止。

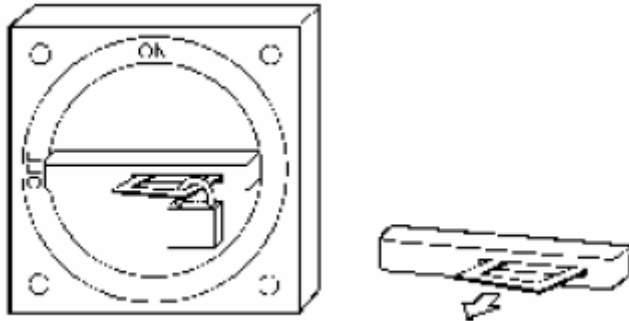
说明：连接*FENCE(围栏)输入信号到安全门。参考维护手册的连接说明。

说明：当 *SFSPD（安全速度）输入信号关闭时，控制单元暂停机器人。

- 设置保护性的围栏，将机器人的动作范围环绕。此外，把控制器放置在保护性围栏的外面。

- 根据需要，安装一把锁，仅仅授权人员可以打开电源。

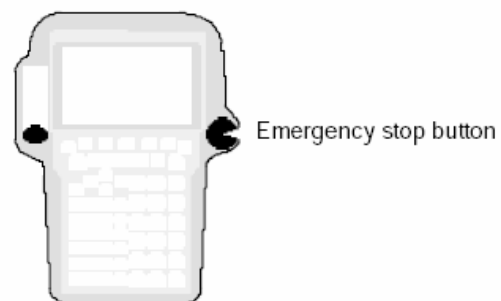
说明：控制单元门上的断路器设计，使得通过一个挂锁，限制电源开关的使用。
锁住断路器



有关检查和维护的安全措施

- 在开始检查和维护前，关掉控制器，锁上断路器或者设置看守，防止其他人合上开关。
- 在断开气动系统前，释放压力供应设备的气压。
- 检查中不需要操作机器人电气系统时，请在检查前按紧急停机按钮。
- 当需要在机器人运行时检查机器人，请仔细观察机器人动作。随时准备在需要时按紧急停机按钮。

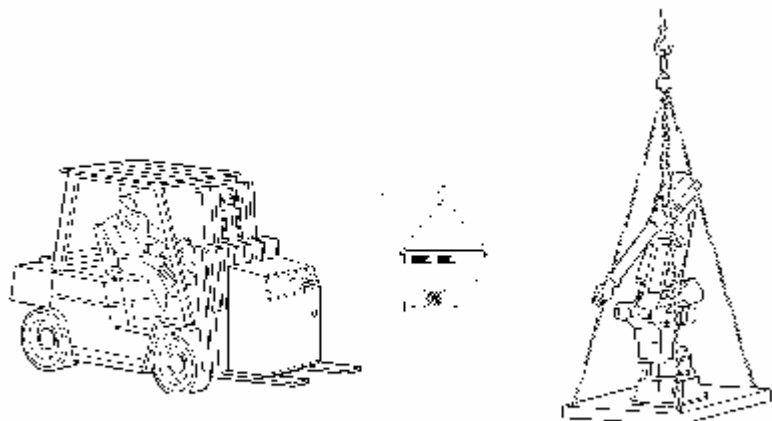
紧急停机按钮



运输相关的安全措施

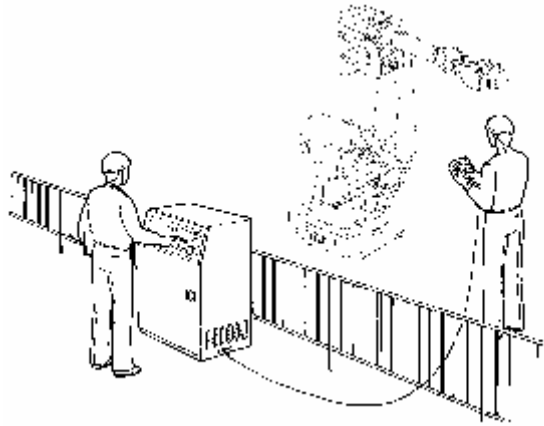
- 当用搬运器（例如起重机或者叉式搬运机）运输机器人或者其他设备，确认把机器人绑紧在搬运器上。

- 仔细观察起重机、叉式搬运机或者其他搬运设备，小心搬运本产品。
搬运机器人



操作相关安全措施

- 所有机器人系统的操作者，都应该参加FANUC 的训练课程，学习安全防护措施和使用机器人的功能。
 - 在开始运行机器人的之前，确认机器人和外围设备周围没有异常或者危险状况。
 - 在进入操作区域内工作前，即便机器人没有运行，也要关掉电源，或者按下紧急停机按钮。要设置看守，保证没有其他人进入机器人操作区域，防止其他人通过操作面板激活机器人。
 - 当在机器人工作区编程时，设置相应看守，保证机器人能在紧急情况，迅速停车。
- 两人共同监视，避免危险



安全防护项目

操作者	车间	运输和安装
禁止危险动作；穿戴工作服、安全鞋和安全头盔	保持车间的整洁、有序。安装一个保护性的护栏和相应的警告牌；保持空气畅通；禁止将易燃易爆物品带入车间。	保持运输通道的畅通。当用搬运器(例如起重机或者叉式搬运机)运输机器人或者其他设备,确认把机器人绑紧在搬运器上；给操作预留足够的空间；正确连线。
操作	维护和检查	焊接机和焊枪
（包括没有相应权限的职员）参加培训课程；掌握操作流程。	请用FANUC推荐的产品进行修理；在检查或者维护前，关掉电源，关闭控制门。	检查并小心观察电缆；检查气压；检查机械手；检查并维护电缆；检查气压；隔离焊枪和机器人；提供散射保护墙；检查油路水路的泄漏。

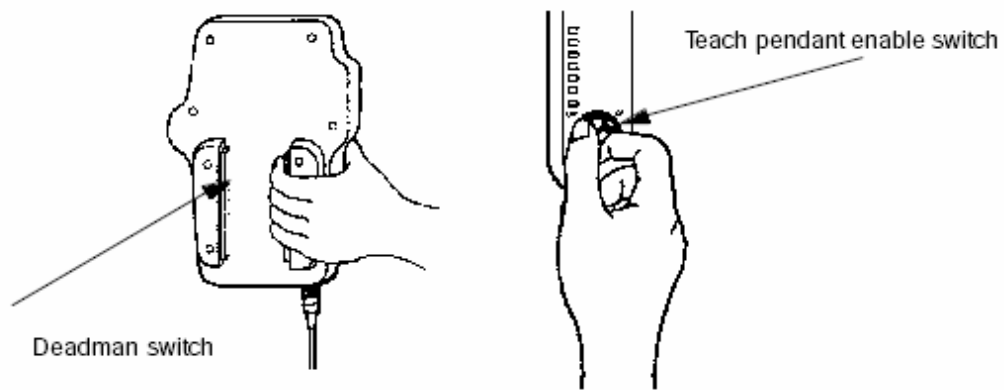
安全服和安全头盔



- 在接近机器人操作它之前，请手持教导盒，按下deadman开关（特殊手持型开关），设置教导盒开关使能为开。

说明：如果释放deadman开关（特殊手持型开关），同时教导盒使能开关打开，机器人会立即停止。

Deadman开关（特殊手持开关） 和教导盒使能开关



- 在通过手动进给运行机器人之前，请仔细观察进给键的操作和机器人的动作。
- 在通过手动进给运行机器人之前，要充分降低进给速度倍率。

概述

本章展示了 **FANUC** 机器人系统基本的配置，并简短描述了每个组件的功能。

本章目录

操作工具软件

机器人

控制器

一个FANUC机器人系统包括了用于操作工件的工具软件、机器人本身的机械单元（FANUC机器人系列）和机器人控制系统。

在操作汽车零件和其他产品时，机器人系统展现了杰出的性能。

用于操作工件的工具软件

工件操作软件是装在控制单元上的一个应用程序包。可以通过教导盒上的特定的指令和菜单，执行所有工作。用于操作工件的工具软件包括控制机器人手臂的指令、远程控制单元和其他外围设备。

可以用这个软件，控制附加轴，控制控制单元与其他外围设备的输入输出（I/O）。其他外围设备包括电池控制、软盘驱动器、打印机和图像传感器。

机器人

机器人拥有一个焊接喷枪或者其他用于能够控制而工作的受动连接端。这个FANUC机器人是用来操作工件弧焊的。

控制器

机器人控制单元提供驱动机械单元的动力。

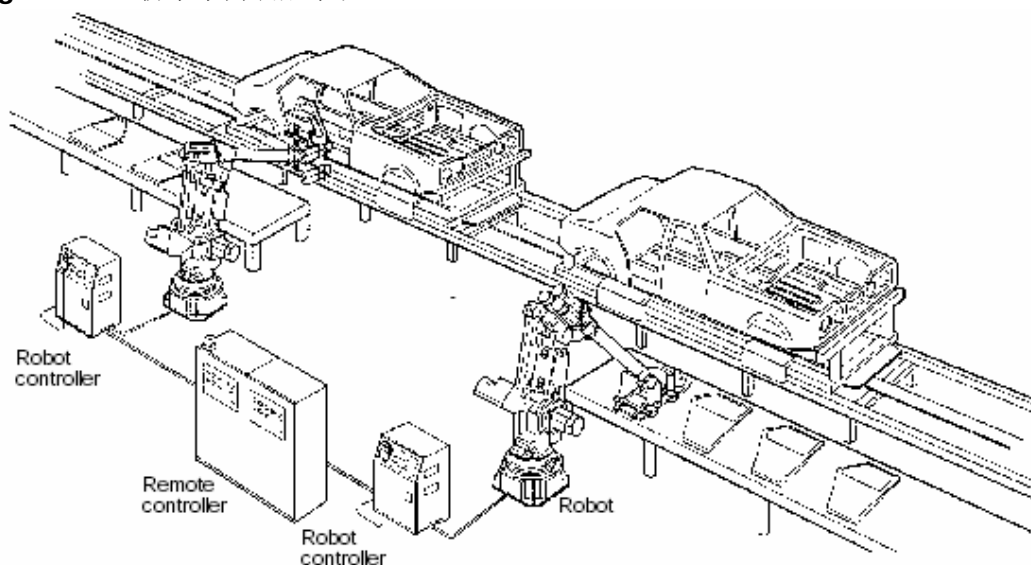
装在控制单元上的工件操作软件控制教导盒、操作面板和外围设备。

包括远程控制单元的外围设备，同样用于配置系统，以达到操作工件的目的。

- 远程控制单元用于控制机器人控制单元。
- 通过输入输出(I/O)接口和其他通讯单元系列，操作手柄、软驱、打印机和其他设备。

图. 2--1 示意了一个典型的操作工件的机器人系统。系统包括了一个机器人、控制单元和外围设备。

Figure 2--1. 轿车车门装配系统



这个控制器在一个机柜中有一个控制机构（以下文均指R--J2 Mate 控制器），它在单机状态有效（FANUC SYSTEM R--J2 Mate）。

在本手册中，外围单元是指操作弧焊系统中必须的单元，包括远程控制器。

- 远程控制器使R--J2 Mate控制器可以进行外部控制。
- 通过输入输出接口和一系列通讯方法，控制工件卡具，软驱和打印机。

操作工具软件

我们特别设计了用于材料处理操作的操作工具软件。工具软件包含在机器人中，它能够做到如下功能：

- 设置系统用于材料处理。
- 创建一个程序。
- 运行程序的测试操作。
- 运行自动操作。
- 状态显示或监控。

可选功能的组合，会扩展系统，并增强了管理功能。

1.1 系统设定

在弧焊操作工具软件的界面上，有参数设置用于操作弧焊系统。

通过弧焊操作工具软件，可以控制焊枪、焊接机、远程控制器和其他外围单元。在弧焊操作开始前，需要注意以下几点：焊枪、电焊机和其他外围设备从何输入、从何输出，坐标系、通讯和自动操作系统又是如何。

1.2 机器人的手动进给

机器人的手动进给指得是，根据需要，在教导盒上手动输入指令，操作机器人动作。当用机器人手动进给，执行程序运动指令时，机器人向目标运动，随之位置被记录下来（对于机器人的手动进给，参看第五章）。

1.3 编程

一个程序包括动作指令、输入输出指令、寄存指令和转移指令（对于程序结构，参看第四章）。每条指令都有一个语句号。通过连续执行这些指令，完成目标工作。

跟踪定位教导盒用于创建和改正程序（对于创建程序，参见第五章）。程序包括了以下指令。图 2—3 列举了操作弧焊的基本编程。

- **动作指令：**移动工具到操作范围内的目标位置。
- **附加运动指令：**在运动中，执行一个附加（特别）的操作。
- **弧焊指令：**控制焊接机和焊枪。
- **寄存指令：**向寄存器输入（读取）数据。
- **位置寄存指令：**向寄存器输入（读取）位置数据。
- **输入输出指令：**接收或向外围设备输入信号。
- **转移指令：**改变程序流程。
- **等待指令：**直到特定条件满足前，暂停程序的执行。
- **程序调入指令：**调入并执行一个子程序。
- **宏指令：**调用一个特定程序并执行。
- **装运指令：**装运工件。
- **程序结束指令：**终止程序。
- **注释指令：**添加程序的注释。
- **其他指令**

弧焊程序

	程序名称	SAMPLE1	JOINT 10%
程序指令	行号	1: J P[1] 100% FINE	1/8
	运动指令	2: HAND1CLOSE	
	宏指令	3: J P[2] 70% CNT50	
		4: L P[3] 500mm/sec CNT10	
		5: HAND1OPEN	
		6: L P[4] 500mm/sec CNT10	
	结束指令	7: HAND1CLOSE	
	程序结束标志	8: ABORT	
		[End]	
		POINT	TOUCHUP >

	程序名称	SAMPLE1	JOINT 10%
程序指令	行号	1: J P[1] 100% FINE	1/8
	运动指令	2: J P[2] 70% CNT50	
	弧焊指令	3: L P[3] 500mm/sec FINE	
		: Arc Start[1]	
		4: L P[4] 50mm/sec CNT80	
		5: L P[5] 50mm/sec CNT80	
		: Arc End[50.0V,75.0A,0.1s]	
		6: J P[1] 100% FINE	
	程序结束标志	[End]	
		POINT ARCSTRT WELD_PT ARCEnd TOUCHUP>	

1.4 测试操作 (测试执行模式)

在系统设置和程序创建完毕后，在测试执行模式下进行测试操作，检查正常操作下的程序（对于测试操作，参看6.2 节和6.3节）。

在创建程序的过程中，程序的测试执行是一个重要的步骤。在开始自动操作前，请执行测试程序。

1.5 自动操作(操作执行模式)

自动操作（操作执行模式）是执行程序的最后步骤。在自动运行中，将执行以下过程：

- 制定程序将连续运行。(对于自动运行请参看第3.11 节和 6.8节)
- 在自动运行阶段，可以调整位置数据。(参见6.9节的在线位置纠正)
- 在自动运行阶段，可以调整焊接状态数据。(对于焊接调整的解释，请看***节)
- 进程暂停，随后退出或者继续进行。(对于暂停程序，请看6.1节)

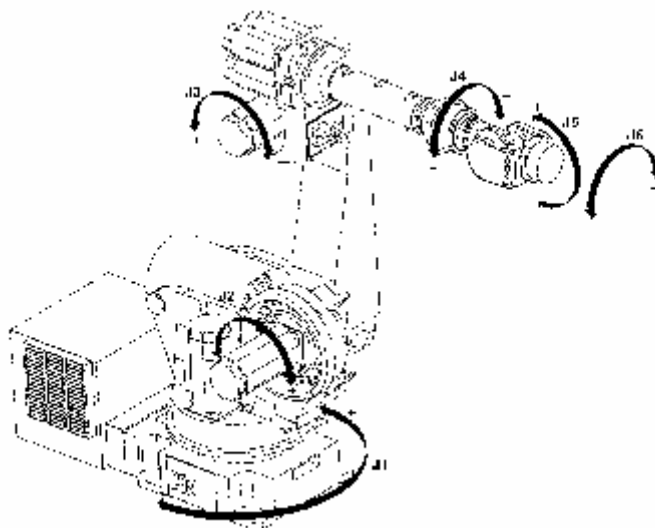
2 机器人

机器人是由伺服电机驱动的轴和机械手臂的机械单元。连接机械手臂的地方是一个机械关节或者一个轴。

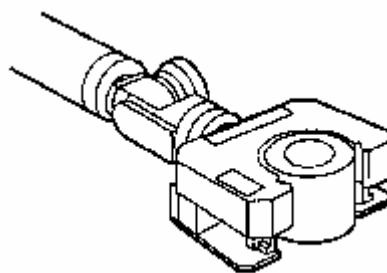
J1, J2, 和 J3 是主轴。机器人的基本设置依据的是, 每个主轴是作为线性轴还是旋转轴。

腕轴的作用是移动在腕边缘上的受动端(工具)。腕本身可以围绕另外一个腕轴旋转, 同时受动端可以绕另外一个腕轴旋转。

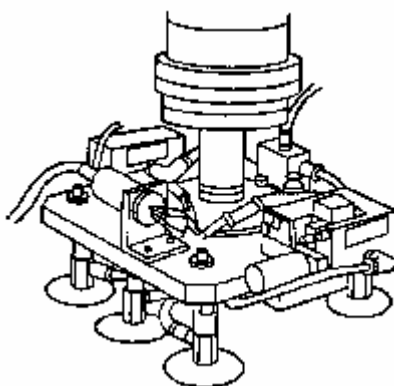
主轴和腕轴



有夹头的机械手

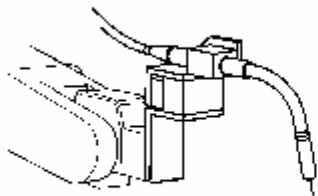


有吸盘无夹头的机械手



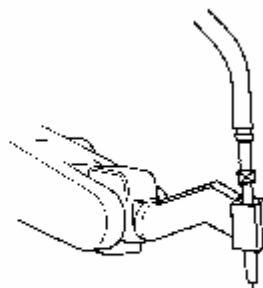
在一个弧焊系统中, 弧焊枪装在机器人的机械腕上。

在腕端上安装一个弧焊枪。LR弧焊软件会控制焊枪和焊接机，这样能确定最适宜的焊接方法。通过和系统设计者的商议，选择对您的应用最合适的焊枪。
焊枪



Curved torch for MAG welding

用于MAG焊接的弧形焊枪



Straight torch for TIG welding

用于TIG焊接的直线焊枪

3 控制器

机器人控制器包括动力单元、使用者接口电路、动作控制电路、存贮器电路和输入输出电路。

这个机器人控制器(这个指 **R--J2 Mate** 控制器)使用在单机型(**FANUC SYSTEM R--J2 Mate**)中, 对于详细资料, 请参考维护手册。

使用者应该使用教导盒和操作箱操作控制单元。

操作控制线路控制伺服放大, 通过主 **CPU** 电路, 从而运行包含所有附加轴的机器人轴。

存贮器电路可以存储由使用者设置在 **CPU** 电路板上的 **C—MOS** 存储器上的程序和数
据。

输入输出电路通过 **I/O** 接口电缆和外围连接电缆, 接受和传送信号, 使控制器与外围设备互联。远程输入输出信号用于和远程控制器的通讯。

机器人控制器

R-J3i MODEL B controller

Operator panel

Three mode switch

Teach pendant

控制器线路设计依赖于它控制的机器人和系统, 更详细的资料请参见维护手册。

3.1 教导盒

教导盒是操作工具软件和操作者之间的桥梁。它通过电缆与控制器的印刷板连接。可以用教导盒进行以下的操作：

机器人的连续手动进给J

- 程序生成
- 测试模式
- 实际工作的进行
- 状态的检查

教导盒包括以下部分：

- 16行，每行40字符的液晶显示器
- 11个LED（发光管）
- 11个LED（包括为LR弧焊软件设计的LED)
- 61个按键
- 61个按键（包括为LR弧焊软件设计的3个按键）



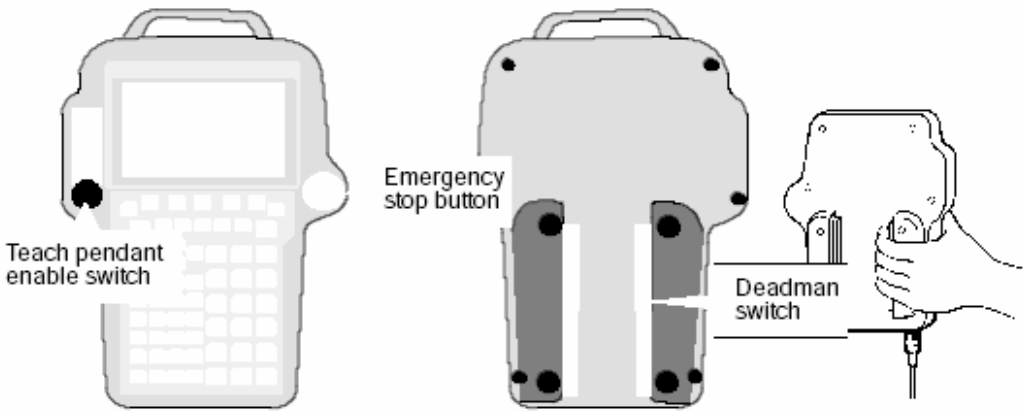
注意

为了不导致操作失误，教导盒的操作者不能佩戴手套

以下为提供的开关：

Teach pendant enable switch 教导盒使能开关	这个开关决定了教导盒是否可用。当开关关闭时，手动进给、程序的生成或者测试模式都不能进行。
deadman switch特殊手持式开关	这个开关是一个使能装置。在教导盒可操作时，只有deadman开关抱紧，机器人才可以动作，如果放开这个开关，机器人会立即停止。
Emergency stop button 紧急停机按钮	按下这个按钮会立即停止机器人的运行。

教导盒上的开关



词汇对照：

Teach pendant enable switch

教导盒使能开关

Emergency stop button

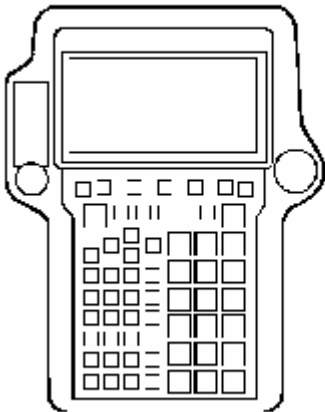
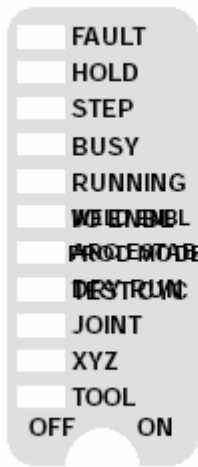
紧急停机按钮

Deadman switch

特殊手持式开关（Deanman开关）

教导盒上的LED灯

教导盒上的LED灯

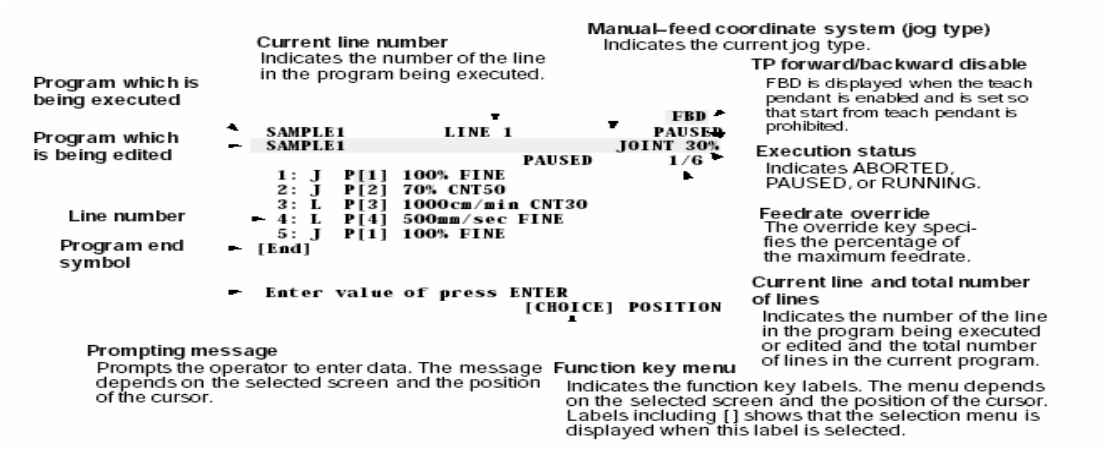


教导盒上的LED灯

LED灯	功能
FAULT	灯亮表明警报发生
HOLD	灯亮表明HOLD键（保持键）被按下或者正在输入保持信号（HOLD）
STEP	灯亮表明系统正运行在分步状态
BUSY	灯亮表明机器人正在工作。如果程序正在执行或者打印机、软驱单元正在运行，此灯也会点亮
RUNNING	灯亮表明程序正在执行
WELD ENBL	灯亮表明可以进行弧焊
ARC ESTAB	灯亮灯亮表明弧焊进行中
DRY RUN	灯亮表明在测试操作状态，预操作
JOINT	灯亮表明选择关节手动坐标（JOINT）作为手动进给坐标系（手动类型）
XYZ	灯亮表明选择笛卡儿坐标系（JGFRM World frame 或者 USER）作为手动进给坐标系（手动类型）
TOOL	灯亮表明选择工具坐标（TOOL）作为手动进给坐标系（手动类型）

3.1.1 显示教导盒界面

下图显示的就是液晶屏上显示的工具操作软件界面。根据操作机器人的需要，选择需要功能的界面。显示了通过屏幕菜单选择界面。



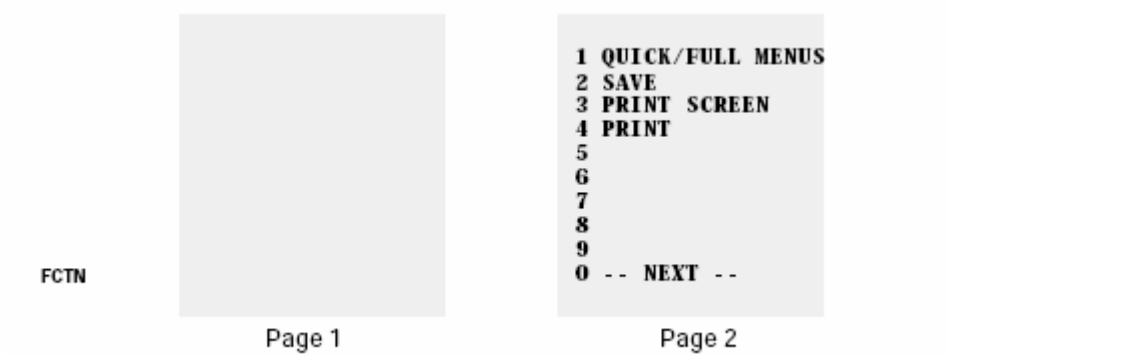
屏幕菜单

LED灯	功能
UTILITIES	实用界面用来显示一些提示
TEST CYCLE	用来设定测试操作数据
MANUAL FCTNS	这个手动操作界面用来执行宏指令
ALARM	警报历史记录界面显示了警报的历史记录和细节
I/O	输入输出界面用来显示和设定手动输出、仿真输入输出和信号的分配。
SETUP	设置界面用来设置系统
FILE	文件界面用来读取或者存储文件
USER	用户界面显示用户信息
SELECT	程序选择界面用来列举或者创建程序
EDIT	程序编辑界面用来校正和执行程序
DATA	程序数据界面用来显示位置寄存器和堆栈寄存器中的值。
STATUS	状态界面显示系统状态
POSITION	位置界面显示了机器人的当前位置
SYSTEM	系统界面用来设置系统参数和控制系统

功能菜单

功能菜单用来执行各种功能（对于菜单目录，请参看附录A.1）。按教导盒上的FCTN键，可以显示功能菜单。

功能菜单



功能菜单

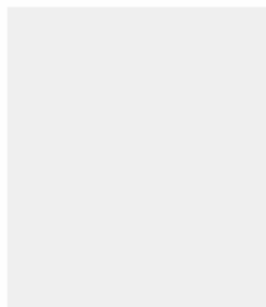
LED灯	功能
ABORT (ALL)	退出正在执行的程序或临时终止
Disable FWD/BWD	设置能否从教导盒启动程序
CHANGE GROUP	手动进给中，改变操作组。只有设置了多个组，才能显示此菜单。
TOGGLE SUB GROUP	套紧机器人主轴和扩展轴间的连接
TOGGLE WRIST JOG	套紧姿态控制进给和腕关节进给（不包括腕姿态线性进给）之间的连接
RELEASE WAIT	跳过当前执行程序的等待指令。当等待状态解除时，程序的执行会临时停止在随后的等待指令
QUICK/FULL MENUS	捆绑了一个由普通界面和快捷界面组成的菜单
SAVE	把与当前界面相关的数据存贮在一张软盘或者存储卡上
PRINT SCREEN	打印当前屏幕上的数据

快捷菜单

当选择了功能菜单中的快捷菜单（QUICK/FULL），屏幕会显示由以下缩减组成的屏幕菜单

- ALARM / 警报发生和警报历史界面
- UTILITIES / 提示界面
- Setup / 设置界面
- DATA / 寄存器界面
- MANUAL FCTNS
- STATUS
- I/O / 数字I/O、组I/O和机器人I/O 界面
- POSITION

快捷菜单



```
1 ALARM
2 UTILITIES
3 SET UP
4 DATA
5 MANUAL FCTNS
6
7 STATUS
8
9 POSITION
0
```

提示 可以用SELECT键调用程序选择界面。但是只有选择程序的功能可用。

提示 可以用EDIT键调用程序编辑界面。但是只能修改位置和速度值。

3.2 操作面板

操作面板由按钮、开关和连接器组成。
面板上按钮用来开关电源、开始程序、解除警报状态和进行其他操作。

 **注意**

操作面板有一个RS-232C通讯接口和一个存储卡槽。

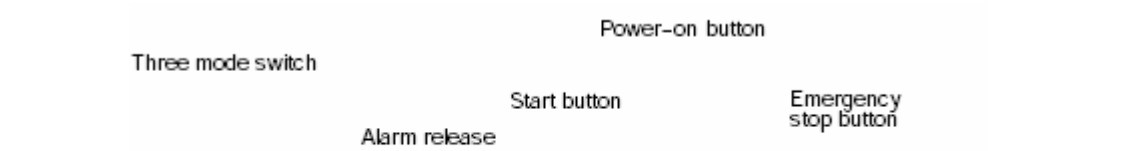
操作面板上的开关

开关	功能
Power-on/off button	开关机器人控制单元的电源。电源开时，按键亮。
Emergency stop button	按下此键立即停止机器人。顺时针方向旋转，解除此键。
Alarm release button	接触警报状态
User #1 and #2 buttons	执行用户键定义的功能
Three mode switch	通过此键选择适合机器人操作条件和使用状态的操作状态

操作面板上的LED灯

LED灯	功能
Alarm	表明警报状态。按下alarm release按钮解除警报状态

操作面板（标准）



3.3 远程控制器

系统配置中，远程控制器是属于外部设备，它连接到机器人控制单元。

使用者使用外围设备和机器人控制单元提供的输入输出设备，创建系统操作，这些控制单元控制这些操作。

显示器与键盘（**CRT/KB**）

显示器和键盘是可选操作单元。一个外部显示器和键盘通过 **RS--232—C** 标准的电缆连接到控制单元。

几乎所有的教导盒上的功能，都可以用显示器和键盘完成，但不包括与机器人运行相关的操作。机器人运行相关的操作只能用教导盒运行。

3.5 通讯

根据通讯需要，提供了以下接口

- 一个标准 **RS--232—C** 端口（外部）
- 两个可选 **RS--232—C** 端口（内部）

3.6 输入输出

通用信号和专用信号用来把外部单元的数据送达工具软件处理。通用信号（用户定义信号）由程序控制，用它从外部单元或焊枪收取信号，或者发送信号到外部单元或焊枪。专用信号（系统定义信号）具有特定作用。

3.7 外围设备输入输出

外围设备输入输出信号专门用于对外围设备或者远程控制器的信号读取与发送。（参见节 3.7 “外围设备输入输出”）

外围输入输出执行如下功能：

- 选择一个程序
- 开始和停止一个程序
- 从警报状态恢复系统
- 其他

3.8 机器人的动作

通过一个独立的动作指令，指定一个机器人的动作，或者一个工具中心点（TCP）从当前位置到目标位置的移动。机器人使用的是动作控制系统，它包括控制工具路径、加速（减速）、定位、进给和其他因数。

机器人控制单元可以控制多达 16 根轴，分成多达三个操作组（复合动作功能）。控制器可以一组控制多达 9 轴。组运行可以相互独立，但是可以通过同时运转机器人，使其同步。

机器人移动是通过教导盒上的手动进给或者程序中指定的动作指令进行的。

用教导盒上相应的按键，执行机器人的手动进给。手动进给中，所选手动进给坐标系（手动类型）和进给速度倍率，决定了机器人的动作。

使用一个动作指令时，机器人的动作依据于指令指定的位置数据、动作类型、定位路径、移动速度和进给超越量。

可以选择三种动作类型（线性、圆弧和关节运动）中的一种作为机器人动作。当选择关节动作（JOINT）时，工具会随机的从指定的两个点间移动；当选择了圆弧动作（Circular）时，工具回沿着一段圆弧连接三个指定的点；当选择线性运动（Linear）时，工具会沿着一一条直线，运行于指定的两点间。

可以从Fine和Cnt中选择定位路径。

3.9 紧急停机装置

机器人由如下紧急停机装置

- 两个紧急停机按钮（安装在操作面板和教导盒上）
- 外触发紧急停机（输入信号）

任何情况下按下紧急停机按钮时，机器人都会立即停机。外部紧急停机输出或输入紧急停机信号到外围设备（例如安全护栏或安全门）。信号终端在控制器上和操作箱内部。。

3.10 扩展轴

一个动作组中三轴中最大一个，可以被加到机器人的标准轴中（通常为 6 轴）。机器人可以控制多达 16 个的轴（用一个可选伺服控制卡）。扩展轴有如下两种类型。

- 外部轴
它只能在关节动作的时候移动，控制时，不用考虑机器人本身的动作。
- 联合轴
它在机器人线性或者圆弧操作中，和机器人本身一起被控制。通过这些轴的使用，实现机器人线性或者圆弧操作。

3.1 输入输出（I/O）

输入输出信号（I/O）是让控制器和机器人、末端执行器、外部设备以及其他系统外围设备通讯的电信号。信号分成两组：多用 I/O 和专用 I/O。

[i] 代表每个I/O信号和组信号的逻辑序号。

[i] 代表每个I/O信号和组信号的逻辑序号。

- 对于数字、组、模拟和外围设备I/O，逻辑端口被映射到物理端口。他们可以被重新定义。
- 机器人I/O的物理号和逻辑号总是一样的。他们不能被重新定义。

导轨

导轨指明了组成I/O模块的硬件类型

0 = 进程I/O印刷电路板

1 到16 = 单元I/O—模型A/B

插槽

插槽（SLOT）指明了组成导轨的I/O模块序号。

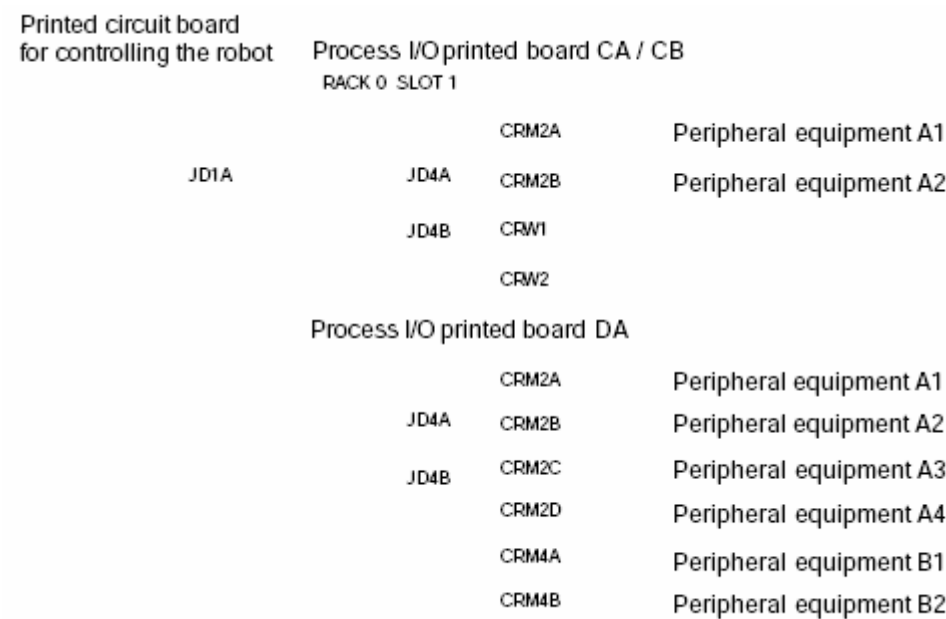
- 当使用印刷板进程I/O时，连接的第一块板即为SLOT1，第二块为SLOT2，以此类推。
- 当使用模型A/B的单元I/O时，SLOT就是确认连接模块的序号。

进程I/O印刷电路板

对于电路板上的输入输出信号，当分配外围I/O到电路板时，在外围设备I/O中，分配了18个输入信号和20个输出信号。除了外围设备信号以外，I/O信号被分配到数字I/O和组I/O中。**提示** 在进程I/O电路板上的头四个信号通常固定在24V。
进程I/O印刷电路板



进程I/O电路板配置



输入输出仿真

选择了仿真输入输出时，在没有和外部设备交换数据的情况下，可以用程序测试。

流程 配置数字I/O



注意

连接进程I/O印刷电路板时，工厂制定了标准分配。没有连接进程I/O印刷电路板时，连接单元I/O模型A/B，所有数字信号的输入输出由工厂分配到数字I/O。没有数字信号分配到外围I/O。区分外围设备I/O和数字I/O的数字输入输出信号，并重新分配它们。

- 步骤** 1 按MENUS键。显示屏幕菜单。
2 选择5 [I/O].
3 按F1 [TYPE]. 显示屏幕转换菜单。
4 选择 “Digital.”

数字I/O选择界面

4 ALARM		Digital I/O Selection Screen			
5 I/O		I/O Digital Out			
6 SETUP		JOINT 30%			
MENUS		#	SIM	STATUS	
		DO[1]	U	OFF	[]
		DO[2]	U	OFF	[]
		DO[3]	U	OFF	[]
		DO[4]	U	OFF	[]
		DO[5]	U	OFF	[]
		DO[6]	U	OFF	[]
		DO[7]	U	OFF	[]
		DO[8]	U	OFF	[]
		DO[9]	U	OFF	[]
Digital		[TYPE]	CONFIG	IN/OUT	ON OFF

5按F3键选择IN/OUT，切换输入界面到输出界面，反之亦然。

[TYPE] CONFIG IN/OUT

6 要分配I/O，按F2,选择CONFIG。要回到选择界面，按F2,选择MONITOR.

数字I/O设置界面

Digital I/O Configuration Screen

[TYPE] CONFIG		IN/OUT		I/O Digital Out				JOINT 10 %	
	#	RANGE	RACK	SLOT	START	STAT.			
1	DO	1 - 20	0	1	21	ACTIV			
2	DO	21-512	0	0	0	UNASG			
[TYPE] MONITOR IN/OUT DELETE HELP									
[TYPE] CONFIG		IN/OUT		I/O Digital Out				JOINT 30 %	
	#	RANGE	RACK	SLOT	START	PT			
1	DO	121-128	0	1	5				
2	DO	129-136	0	1	13				
3	DO	137-144	0	1	21				
4	DO	145-152	0	1	29				
5	DO	153-160	0	1	37				
6	DO	161-168	0	1	0				
7	DO	169-176	0	1	29				
8	DO	177-184	0	1	37				
9	DO	185-192	0	0	0				
[TYPE] MONITOR IN/OUT DETAIL HELP >									
[TYPE] VERIFY									

- 7 操作I/O分配
- 8 要回到目录界面，按F2,选择MONITOR.
- 9 要设置I/O属性，按NEXT键，并按F4, 选择DETAIL。
- 10 添加注释
- 11 移动光便到设置栏，选择功能菜单设置项目，
- 12 设置下一个数字I/O组，按F3, 选择NEXT.
- 13 完成后，按PREV 键返回选择菜单。
- 14 重新开关一次控制器，使新信息生效。



警告

重新开关一次电源，新设置才能生效。否则可能引起人身伤害或者财产损失。



注意

在重新分配后的第一次启动，即使电源恢复即使有效，也不能执行。



注意

所有I/O信号设置完毕，应在在外部存贮介质（例如软盘）中存贮设置信息，以便信息重载。否则当前设置随着改变会丢失。

- 15 移动光标到ON和OFF，并按下相应功能键，执行或者仿真输入输出信号。

模拟 I/O

模拟信号(AI/AO) 通过进程I/O印刷电路板上的输入输出信号行，发送到弧焊机和外围设备。当读取和写入时，模拟信号的电压值被转换为数字形式。所以，它们不是直接对应输入输出电压。

输入输出的配置

模拟I/O中的硬件序号，可以重新定义。

提示 工厂制定了标准配置。使用不同于标准配置的配置，请重新定义。



注意

在物理序号重新定义前，需要仔细检查信号的使用。否则会造成人身伤害或财产损失。

导轨

导轨指明了组成I/O模块的硬件类型

0 = 进程I/O印刷电路板

1 到16 = 单元I/O—模型A/B

单元I/O—模型A的基础单元和单元I/O—模型B的接口单元，依据连接的顺序，定义为RACK1、2……

插槽

插槽（SLOT）指明了组成导轨的I/O模块序号。模型A的单元I/O中，底座的插槽序号就是模块的SLOT序号。

- 当使用印刷板进程I/O时，连接的第一块板即为SLOT1，第二块为SLOT2，以此类推。
- 当使用模型A的单元I/O时，模块嵌入的插槽序号就是模型的SLOT值。

当使用模型B的单元I/O时，基础单元的SLOT序号由基础单元内的拨码开关（DIP）指定。

通道

分配物理序号到逻辑序号的信号映射。

提示 物理序号指明了I/O模块上的输入输出管脚逻辑号分配到物理序号。分配可以改变。

通过I/O配置界面和I/O详细界面，可以设置I/O。当I/O设置改变时，请重新开关一次控制器，以便新信息生效。



注意

在重新分配后的第一次启动，即使电源恢复即使有效，也不能执行。

输出的执行

执行程序或运行手动操作，可以指定数字输出值。

输入输出仿真

选择了仿真输入输出时，在没有和外部设备交换数据的情况下，可以用程序测试。仿真I/O实际上不输出和输入信号

机器人 I/O

机器人I/O 是数字信号

操作机器人按照如下进行:

- 通过机器人, 其他信号作为末端执行器I/O使用。末端执行器I/O链接到机器人手臂末端的连接器上, 保证其功能。

.末端执行器I/O包含了8个输入和8个输出多用信号。不能重新定义信号数。

提示 末端执行器I/O的多用I/O信号数依赖于机器人模型。请参考机械单元维护手册。

提示 要改变RI [1]的公用设置, 请把*RDICOM 信号接到接线盒的0 V或者+24 V。

机械手破损输入信号 *HBK

*HBK信号连接到机器人机械手, 探测工具中的损伤。在通常状况, *HBK信号为开, 当它关闭, 发生警报, 机器人立即停止。

提示 可以在系统设置界面中禁用机械手破损探测。

异常空气压力输入信号 *PPABN 输入信号

*PPABN信号探测空气压力的下落。在正常状态, *PPABN信号为开。当空气压力下降时, *PPABN关闭, 发生警报, 机器人立即停止。

*ROT 输入信号

倍率 (机器人倍率) 信号表明机器人机械单元沿着各自的轴运动的倍率。在正常状态, *ROT信号为开。当这个信号关闭, 发生警报, 机器人立即停止。

因为*ROT信号产生在机器人机械单元, 不会在末端执行器的电缆接口产生*ROT输入信号。

当*HBK或者*ROT信号关闭, 可以通过按下移动键, 同时按下警报解除键临时解除警报状态。按住移动键, 通过手动, 移动工具到合适的位置。

RDI [1 to 8] 输入信号

RDO [1 to 8] 输出信号

末端执行器信号, RDI [1 to 8]和RDO [1 to 8]是多用输入和输出信号)

外围设备 I/O

外围设备 I/O(UI/UO)是专用信号组，它的使用由系统定义。这些信号通过如下的界面和 I/O 接口，和一个远程控制器一些外围设备连接，用它们从外部控制机器人。

I/O的配置

在LR弧焊工具软件中，厂家校准的标准设定里，外围设备I/O自动分配到进程I/O印刷电路板。这样在CRM5界面中，可以使用20个多用输入信号和16个多用输出信号(SDI/SDO)。

外围设备信号会自动分配到第一块进程I/O印刷电路板的头18个输入信号和20个输出信号。



注意

连接进程I/O印刷电路板时，厂家制定了标准分配。没有连接进程I/O印刷电路板时，连接单元I/O模型A/B，所有数字信号的输入输出由厂家定义分配到数字I/O。没有数字信号分配到外围I/O。区分数字I/O和外围设备I/O的数字输出输入信号并重新分配它们。

远程条件

当机器人在远程状态时，可以用外围设备I/O启动程序。无论远程状态条件是否满足，和安全相关的信号(*HOLD,ENBL)总是有效。

当如下的远程状态条件满足时，机器人处于远程状态。

- 教导盒使能开关关闭。
- 远程信号(SI[2])开。外围设备I/O*SFSPD输入信号开。
- 外围设备ENBL输入信号开。
- 伺服系统动力开（不在警报状态）

CMDENBL信号表明以上条件是否满足。信号的输出需要满足如下条件：

- 远程条件满足
- 不在警报状态
- 选择了连续操作状态（禁用单步状态）

提示 在初始状态，外围设备I/O信号失效。在系统设置界面中，设置“Enable UI signals”为真（TRUE），使外围设备I/O信号有效。

设置自动操作

自动操作功能，使用外围设备I/O，控制控制器启动一个程序。

自动操作包括如下功能：

- 机器人服务请求（RSR）功能，选择和根据机器人服务请求信号（RSR1到RSR8输入信号），启动一个程序。当另外一个程序正在执行或者临时停止时，被选择的程序进入等待状态，并在当前程序执行完毕后，立即执行。
- 程序号择选(PNS)功能，用程序号选择信号（RSR1到RSR8 PNSTROBF）和START信号，选择或检查一个程序。当有程序正在执行或者临时停止时，这些信号被忽略。
- 自动操作开始信号（PROD_START input），从第一行运行当前选择的程序。当有另外一个程序临时停止或者正在运行时，这些信号被忽略。
- 循环停止信号(CSTOPI 输入信号)用于中止正在运行的程序。

-- 在系统设置菜单中，选中ABORT的CSTOPI 为FALSE时，本信号在一次执行结束后，中止正在执行的程序。可以用它来解除由RSR 定制的程序等待状态。（默认）

--在系统设置菜单中,选中ABORT的CSTOPI 为TURE时,本信号强制中止正在执行的程序。可以用它来解除由RSR 定制的程序等待状态。

- 外部启动信号（START输入信号）用于临时停止的程序。

-- 只有在系统设置菜单上，选择CONTINUE的START为FALSE时，这个信号才能从当前行，启动程序。可以用它来解除由RSR 定制的程序等待状态。（默认）

--在系统设置菜单上，选择CONTINUE的START为TRUE时，这个信号只能启动临时停止的程序。无临时停止的程序时，本信号被忽略。

只有机器人在远程状态时，可以通过输入外围设备I/O启动一个程序。当如下远程条件满足时，进入远程状态：

- 教导盒使能开关关闭。
- 远程信号(SI[2])开。外围设备I/O*SFSPD输入信号开。
- 外围设备ENBL输入信号开。
- 系统参数\$RMT_MASTER设置为零（外围设备）
- 伺服动力开（不在警报状态）

CMDENBL信号表明以上条件是否满足。信号的输出需要满足如下条件：

- 远程条件满足
- 准备条件满足
- 选择了连续操作状态（禁用单步状态）

提示 在系统设置界面中，设置”START for CONTINUE only”为真（TRUE），START仅在程序处于保持状态有效。

机器人服务请求 (RSR)

机器人服务请求从外部设备运行一个程序。8 个机器人服务请求信号 ((RSR1 到 RSR8) 用于这个功能。

1 控制单元通过RSR1到RSR8输入信号，决定是否RSR 信号有效。当信号无效时，忽略此信号。无论在系统参数\$RSR1 到\$RSR8中，RSR1到RSR8是否设置为有效，和在RSR 设置界面中或用程序RSR指令设置其可更改。

提示 在初始状态，外围设备输入信号 (UI) 无效。在系统设置界面上选择UI 信号使能为 TRUE,使其有效。

2 RSR用了8个RSR注册号注册。通过添加基数到RSR而获得的值，用做程序号（对于数字的）。例如，当RSR2作为输入，以下值被用做程序号：

(程序号) = (RSR2 注册号) + (基数)

所选程序如下命名

RSR + (程序号)

提示 用RSR + (程序号)的格式指定自动操作的程序名字。键入如RSR0121m 的4位数字，而不是RSR121。否则，机器人不会执行。

在\$SHELL_CFG.\$JOB_BASE 中设置基数，可以在RSR设置界面或者程序参数指令中，改变其值。

3 对应于RSR1到RSR8的输入信号，会有一个RSR确认脉冲 (ACK1到ACK8) 输出。当输出了ACK1到ACK8信号时，控制单元接收其他的RSR输入。

4 当一个程序在中止状态时，所选程序开始运行。当另外一个程序承载运行或者临时停止时，服务（工作）被列入对立，并在程序运行结束后，开始所选程序的运行。

工作（RSR程序）会按照进入队列的顺序执行。

5 通过循环停止信号 (CSTOP)输入信号) 取消（清除）等待程序或者强制程序中止。

设置PNS 功能

在PNS设置界面[6 (SETUP). RSR/PNS].设置PNS功能。

项目	描述
Program select mode（程序选择模式）	选择RSR或者PNS自动操作模式。两个功能不能同时应用。改变设置后，重新开启一次电源，使其生效。
Base number（基数）	为获得PNS程序号而加入PNS注册号的基数。
Acknowledge function（确认功能）	设置是否输出RSR确认信号（ACK1到ASK8）
Acknowledge pulse width（脉宽）（毫秒）	设置PNS确认脉冲信号(SNACK)的输出时间。（ACK1到ASK8）（单位：毫秒）

设置PNS功能

步骤1 按MENUS键. 显示屏幕菜单。

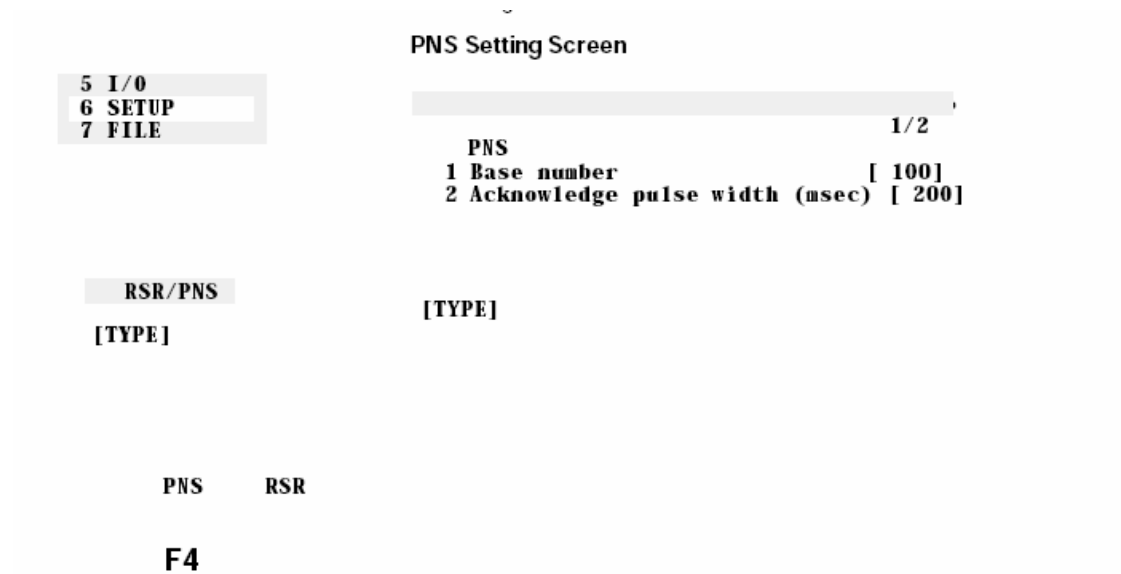
2 选择6 (SETUP).”

3 按 F1 键，选择TYPE. 显示界面切换界面。

4 选择 RSR/PNS.显示RSR/PNS设置界面

5 移动光标到 "Program select mode". 按 F4 [CHOICE]并选择RSR，接着按F3选择DETAIL.

PNS选择界面



- 6 移动光标到需要的位置，输入值。
- 7 RSR到PNS设置完毕后，请重新开启一次电源，使设置生效。



警告

改变完自动操作功能的类型后，控制器单元的电源必须重新开启一次，以使改变生效。否则，设置不能被接受。

设置坐标系

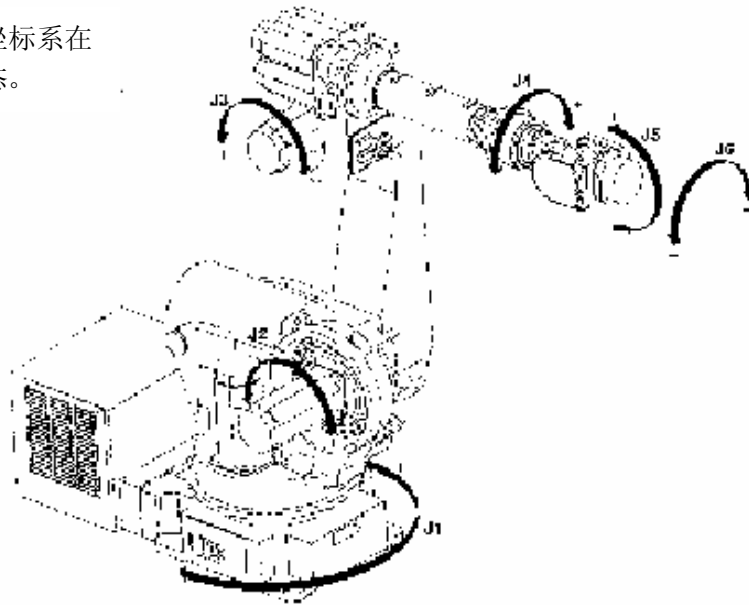
坐标系定义了机器人的位置和姿态。系统在一个工作区定义了机器人。系统使用了关节坐标系和笛卡儿坐标系。

关节坐标系

按照机器人的关节定义了关节坐标系。参照关节基座的关节坐标系的角度偏移，定义了机器人位置和姿态。

关节坐标系

图片右边每个轴的坐标系在
每个轴都是零的状态。

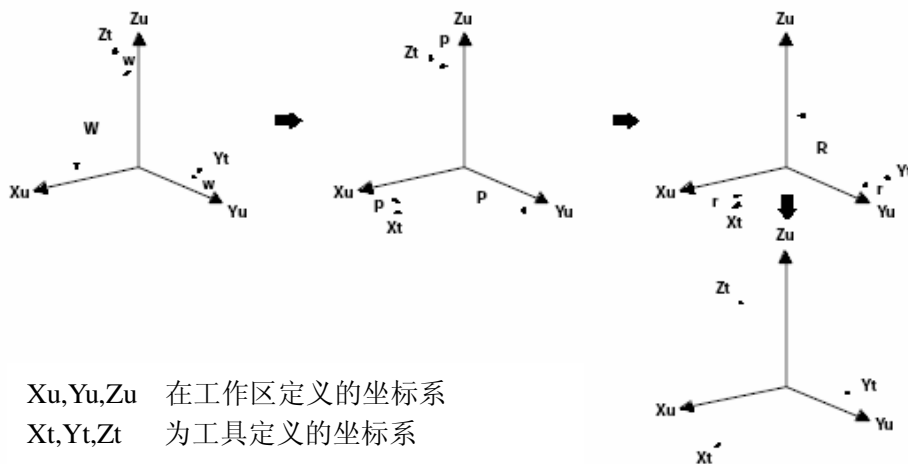


笛卡儿坐标系

按照空间笛卡儿坐标系的原点到工具坐标系原点（工具头点）的坐标 x , y , 和 z , 和工具笛卡儿坐标系相对空间笛卡儿坐标系的 X --, Y --,和 Z —轴的旋转角度偏差 w , p , 和 r , 定义了笛卡儿坐标系中机器人的位置和姿态。(w,p, r)的含义如下所示:

图 3--28. (w, p, r)的含义

Figure 3--28. Meaning of (w, p, r)



X_u, Y_u, Z_u 在工作区定义的坐标系
 X_t, Y_t, Z_t 为工具定义的坐标系

要在用户定义环境操作机器人，请使用相应的笛卡儿坐标系。以下5个坐标系有效：

机械接口坐标系（固定在工具上的坐标系）

对机器人机械接口（腕缘外部）定义的标准笛卡儿坐标系。坐标系固定在由机器人确定的位置。根据坐标系的基础，指定了工具坐标系。

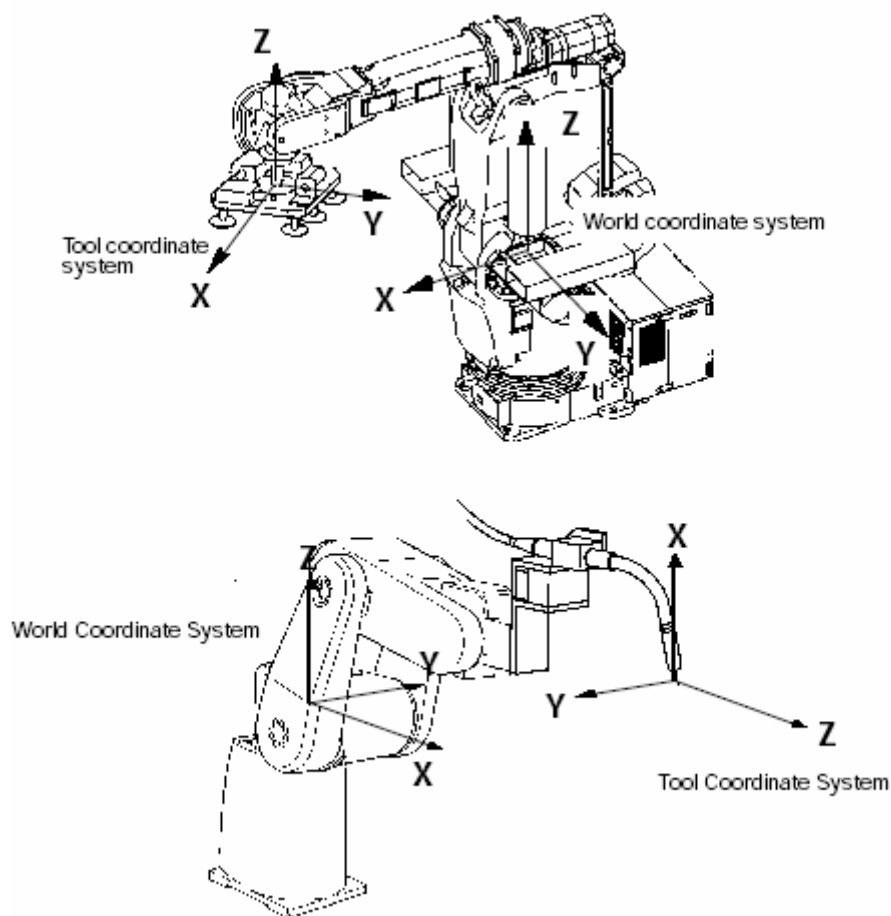
工具坐标系

定义了工具中心(TCP)和工具姿态的坐标系。必须指定工具坐标系。否则，机械接口坐标系会替换它。

全局坐标系（固定在工作区的坐标系）

固定在工作区的标准坐标系。它固定在由机器人决定的位置。根据坐标系，定义了用户坐标系和手动坐标系。全局坐标系用于指定位置数据并执行相应命令。关于全局坐标系详细内容。

全局坐标系和工具坐标系



用户坐标系

由在每个工作区的用户定义的笛卡儿坐标系。用它来指定位置记录，执行相应的位置记录指令和位置补偿指令，等等。如果没有指定这个坐标系，全局坐标系取代它的功能。



警告：

程序教导后，如果由工具或者用户坐标系改变，程序点和范围需要重启。否则设备

手动坐标系

由用户定义的坐标系。手动坐标系用于通过手动有效移动机器人。既然选择手动坐标系为手动进给坐标，您就不用考虑手动坐标的原点。如果没有定义这个坐标系，全局坐标系取代其位置。

设置工具坐标系

工具坐标系使一个笛卡儿坐标系，它定义了工具中心（TCP）位置和工具姿态。在工具坐标系中，零点常代表TCP，Z轴通常代表了工具轴。没有定义工具坐标系时，机械接口坐标系会代替它。

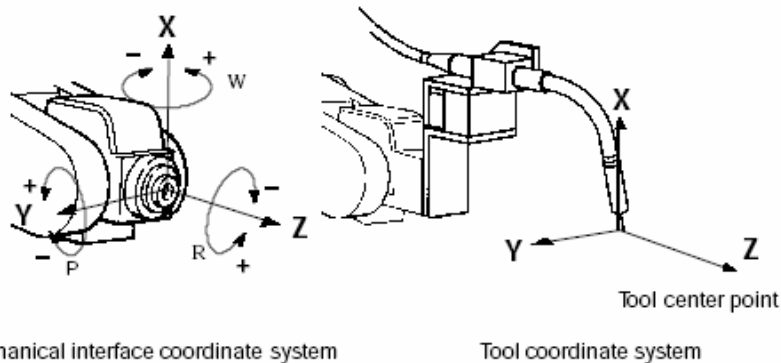
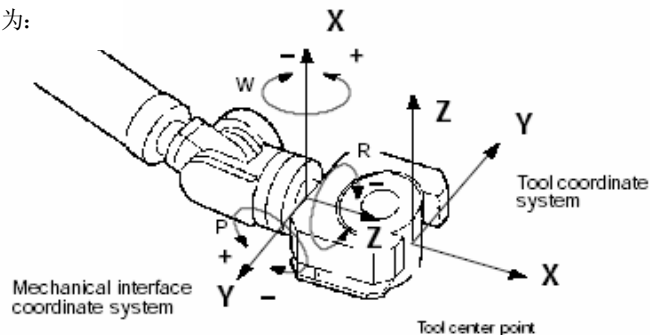
工具坐标系包括(x, y, z)，它表明了工具中心（TCP），还有(w, p, r)表明了工具的姿态。坐标X,Y和Z表明了机械接口坐标系中TCP的位置，W,P和R代表了工具的姿态和机械接口坐标系中X--、Y—和Z—轴的角度偏移。

工具中心用于指定位置数据。工具姿态需要执行工具姿态控制。

工具坐标系

图片右边，工具坐标系坐标为：

x	50.0 mm
y	0.0 mm
z	300.0 mm
w	180.0 deg
p	90.0 deg
r	0.0 deg



用参照系设置界面或按照如下的系统参出定义工具坐标系。可以定义10个工具坐标系。选择所需的即可。

设置F \$MNUTOOOL [1, i] (参照系号 i = 1到10)

F \$MNUTOOOLNUM [group]用于设置所用工具参照系号。

可以按照如下3个方法设置工具参照系。

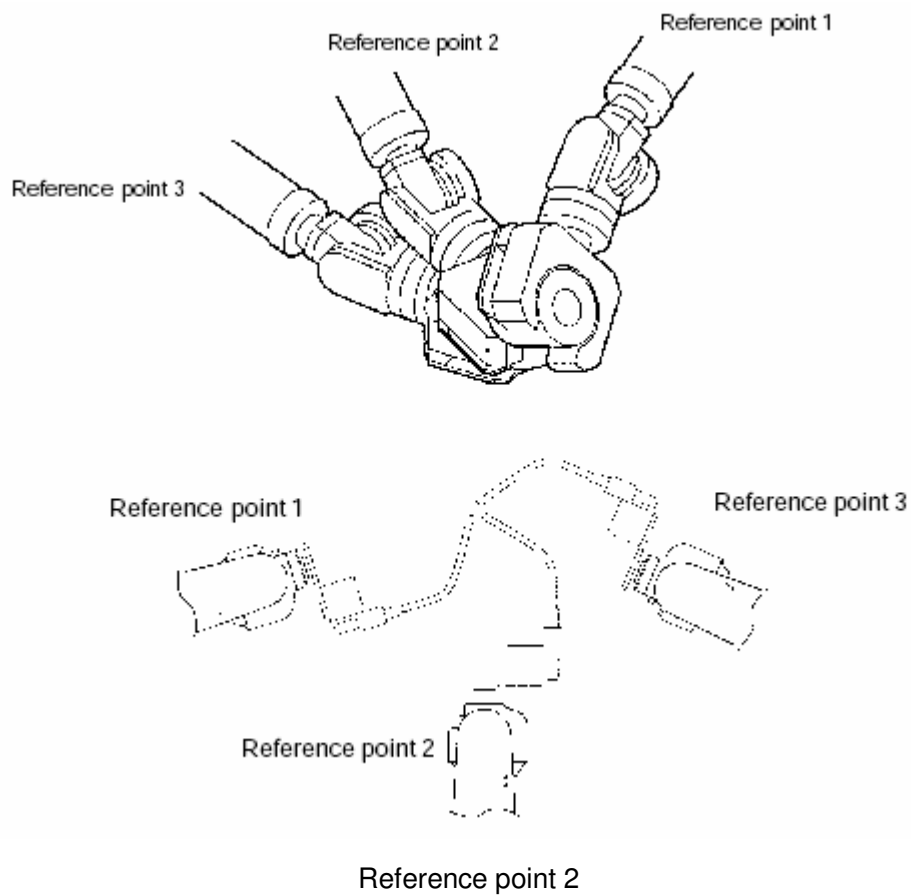
三点法(TCP 自动设置) [Optional function (选项功能)]

用三点法定义工具中心（TCP）。三点必须用工具，从三个不同的逼近状态，接近一个公共点，教导得来。作为结果，自动计算出TCP的位置。

为了TCP的精确性，三个逼近方向要尽量和其他不同。

在三点法中，只能设置工具中心（x, y, z）。工具方位（w, p, r）设置值为标准值（0, 0, 0）。工具方位应在位置设置后，由六点法或者直接录入法得到。

用三点法自动设置TCP



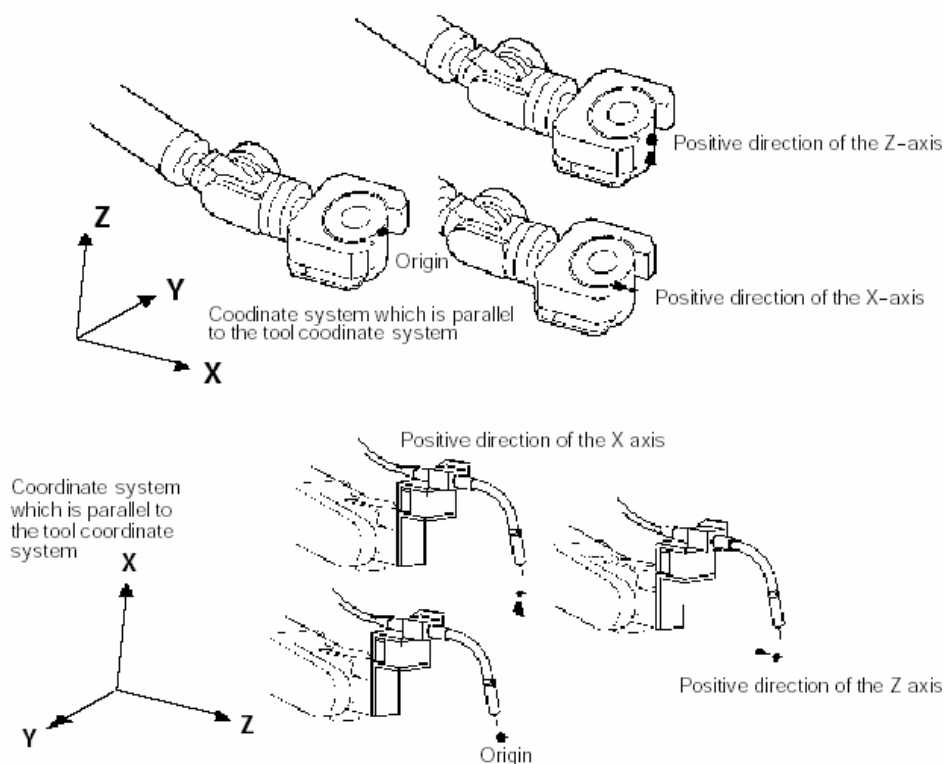
六点法

可以按照和三点法一样的方法设置工具中心。

然后设置工具姿态 (w , p , r)。

这样教导机器人，于是 w ， p 和 r 表示了空间给定一点，平行于工具坐标系 x -轴的正向的一点和在 XZ 平面上的一点。同样用笛卡儿坐标系或者工具手动坐标系教导机器人，这样工具的倾斜不会改变。

六点法



直接录入法

可以直接键入如下值。一个是TCP位置的(x,y,z)值。另外一个为绕机械连接坐标系的x--、y—和z—轴的旋转角(w,p,r) ,它指明了工具坐标方位。

TCP自动设置 (三点法)

步骤1 按MENUS键，显示屏幕菜单。

2 选择t “6 (SETUP).”

3 按F1 键，显示屏幕转换界面。

4 选择Frames.

5 按F3, OTHER 并选择Tool Frame. 显示工具参照系设置界面

工具参照系目录界面

Tool frame list screen

5 I/O

6 SETUP

7 FILE

Frames

[TYPE]

SETUP Frames

JOINT 30 %

Tool Frame Setup/ Direct Entry 1/9

	X	Y	Z	Comment
1:	0.0	0.0	0.0	*****
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****
6:	0.0	0.0	0.0	*****
7:	0.0	0.0	0.0	*****
8:	0.0	0.0	0.0	*****
9:	0.0	0.0	0.0	*****

Active TOOL \$MNUT00LNUM[1]=1

[TYPE] DETAIL [OTHER] CLEAR SETIND

1 Tool Frame

2 Jog Frame

3 User Frame

[TYPE] DETAIL OTHER

- 6 移动光标到您要设置的工具参照系号所在行。
- 7 按F2,DETAIL.显示所选参照系号的参照系设置界面。

[TYPE] DETAIL [OTHER]

- 8 按F2,METHOD 然后选择Three Point.
- 三点法设置界面

Tool frame setup screen (Three Point Method)

1 Three Point

2 Six Point

3 Direct Entry

[TYPE] METHOD FRAME

F2

SETUP Frames

JOINT 30 %

Tool Frame Setup/ Three Point 1/4

Frame Number: 1

X: 0.0 Y: 0.0 Z: 0.0

W: 0.0 P: 0.0 R: 0.0

Comment: TOOL 1

Approach point 1: UNINIT

Approach point 2: UNINIT

Approach point 3: UNINIT

Active TOOL \$MNUT00LNUM[1]=1

[TYPE] [METHOD] FRAME

9 添加注释。

a 移动光标到注释行，按下ENTER键。

```
1 [ ] ENTER
```

b 选择注释的命名方法。

c 选择适当的功能键加入注释。

d 完成后，按ENTER键

10 记录每个逼近点

a 移动光标到每个逼近点

b 手动进给机器人到你所要记录的点

c 按下并保持SHIFT键，并按F5,RECORD 记录下当前位置的数据，作为参考点。显示RECORDED。

提示 从三个不同的方向移动工具，使工具头到达同一点。然后记录参考点。

```
Approach point 3:
      SETUP Frames          JOINT 30 %
FRAME  MOVE_TO RECORD      Approach point 1:  RECORDED
      Approach point 2:  RECORDED
      Approach point 3:  UNINIT
SHIFT [ TYPE ][METHOD] FRAME  MOVE_TO RECORD
```

d 教导了所有参考点后，显示USED。工具参照系设置完成。

```
      SETUP Frames          JOINT 30 %
Tool Frame Setup/ Three Point 4/4
Frame Number: 1
X: 100.0 Y: 0.0 Z: 120.0
W: 0.0 P: 0.0 R: 0.0
Comment: TOOL 1
Approach point 1: USED
Approach point 2: USED
Approach point 3: USED
[ TYPE ][METHOD] FRAME  MOVE_TO RECORD
```

11 要移动机器人到一个记录点，庆按下并保持SHIFT 键，并按F4,选择MOVE_TO.

```
      SETUP Frames          JOINT 30 %
Tool Frame Setup/ Three Point 4/4
Frame Number: 1
X: 100.0 Y: 0.0 Z: 120.0
W: 0.0 P: 0.0 R: 0.0
Comment: TOOL 1
Approach point 1: USED
Approach point 2: USED
Approach point 3: USED
[ TYPE ][METHOD] FRAME  MOVE_TO RECORD
```

12 要参看每个记录点的数据，移动光标到每个参考点项目，按下ENTER键。会显示每个位置的详细数据界面。要返回到上一个界面，请按PREV键。

13 要显示工具参照系目录界面，请按PREV键。您可以看到所有工具参照系的设置。

SETUP Frames				JOINT	30 %
Tool Frame Setup/ Direct Entry				1/9	
	X	Y	Z	Comment	
1:	100.0	0.0	120.0	T00L1	
2:	0.0	0.0	0.0	*****	
3:	0.0	0.0	0.0	*****	
4:	0.0	0.0	0.0	*****	
5:	0.0	0.0	0.0	*****	
6:	0.0	0.0	0.0	*****	
7:	0.0	0.0	0.0	*****	
8:	0.0	0.0	0.0	*****	
9:	0.0	0.0	0.0	*****	
[TYPE] DETAIL [OTHER] CLEAR SETIND					

14 要立即设置激活工具参照系，按F5,SETIND.

15 要删除设置参照系的数据，移动光标到所需的参照系，按F4,CLEAR.

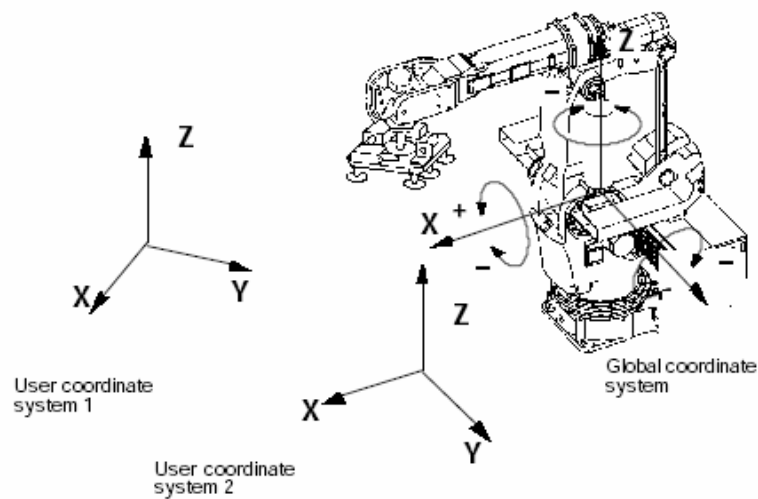
设置用户坐标系

用户坐标系是用户为每个工作区定义的笛卡儿坐标系。如果没有定义整个坐标系，全局坐标系会取代它。

定义用户坐标系，用(x, y, z)表明了零点位置，用(w, p, r)表明绕全局坐标系的 X、Y 和 Z 一轴旋转的角度偏移。

。

全局坐标系和用户坐标系

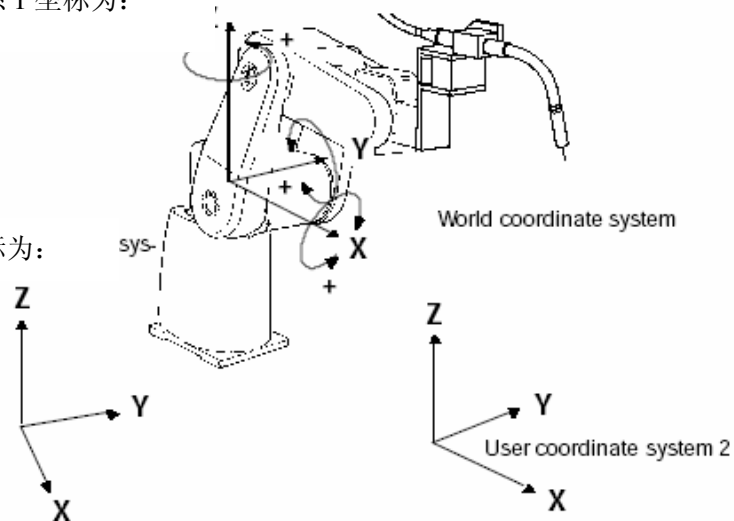


右图中，用户坐标系 1 坐标为：

x : 70.0 mm
y : -2500.0 mm
z : 200.0 mm
w : 0.0 deg
p : 0.0 deg
r : -30.0 deg

用户坐标系 2 的坐标为：

x : 500.0 mm
y : 0.0 mm
z : -200.0 mm
w : 0.0 deg
p : 0.0 deg
r : 0.0 deg



用参照系设置界面，定义用户参照系，改变如下系统参数的值。可以定义9个用户坐标系。可以选择其中的一个。

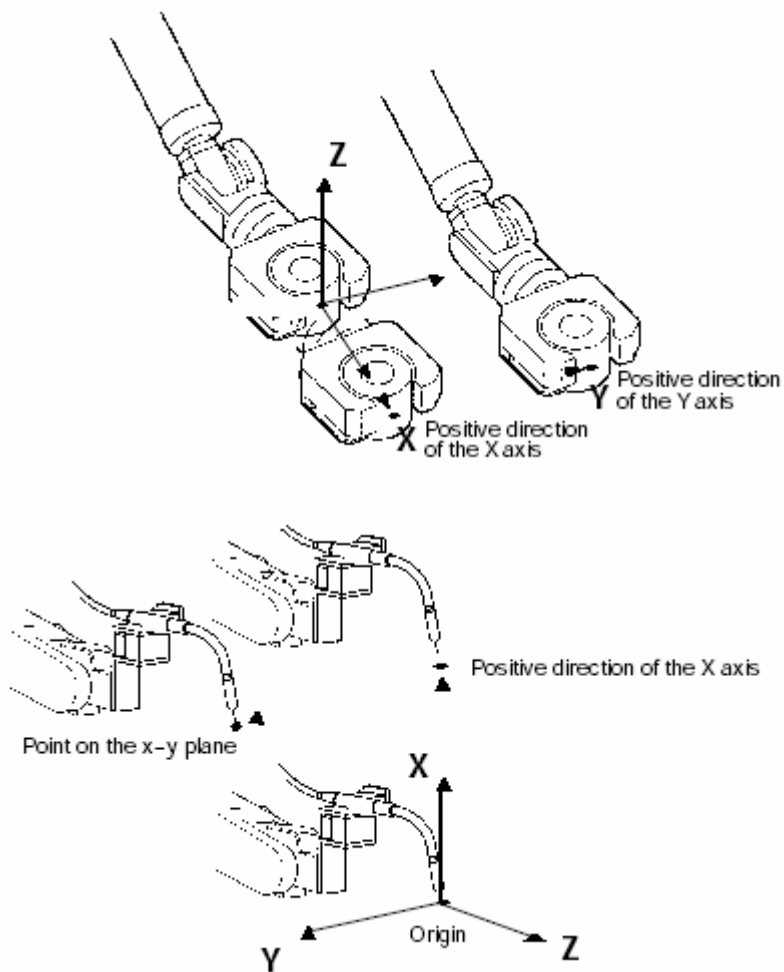
F \$MNUFRAME [1,i] (参照系号 i = 1 到9)。

F \$MNUFRAMENUM [1] 用于设置需要的参照系号。

可以用以下三种方法定义用户参照系。

三点法

按照以下三点，教导三点：x-轴原点，在x-轴正向的点和在x-y平面上的点。



三点法设置用户参照系

步骤1 按MENUS键. 显示屏幕菜单。

2 选择 “6 (SETUP).”

3 按 F1 键 TYPE. 显示屏幕转换界面。

4 选择Frames.

5 按 F3, OTHER 并选择User Frame. 显示用户参照系目录界面。

用户参照系目录界面

User frame list screen

5 I/O

6 SETUP

7 FILE

Frames

[TYPE]

SETUP Frames

JOINT 30 %

User Frame Setup/ Direct Entry 1/5

	X	Y	Z	Comment
1:	0.0	0.0	0.0	*****
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****

Active UFRAME \$MNUFRAMNUM[1]=0

[TYPE] DETAIL [OTHER] CLEAR SETIND

1 Tool Frame

2 Jog Frame

3 User Frame

[TYPE] DETAIL OTHER

- 6 移动光标到需要设定的用户参照系号所在行。
- 7 按 F2,DETAIL. 显示所选用户参照系的参照系设置界面。

[TYPE] DETAIL [OTHER]

- 7 按 F2,METHOD 并选择Three Point.

三点法设置界面

User frame setup screen (Three Point Method)

1 Three Point

2 Four Point

3 Direct Entry

[TYPE] METHOD FRAME

F2

SETUP Frames

JOINT 30 %

User Frame Setup/ Three Point 1/4

Frame Number: 1

X:	0.0	Y:	0.0	Z:	0.0
W:	0.0	P:	0.0	R:	0.0

Comment:*****

Orient Origin Point: UNINIT

X Direction Point: UNINIT

Y Direction Point: UNINIT

Active UFRAME \$MNUFRAMNUM[1]=0

[TYPE] [METHOD] FRAME

9 添加注释

a 移动光标到注释行，按下ENTER键。

```

                                JOINT 30 %
                                [      ] ENTER

```

- b 选择注释的命名方法。
- c 选择适当的功能键加入注释。
- d 完成后，按ENTER键

```

SETUP Frames                                JOINT 30 %
Comment:  REFERENCE FRAME
[ TYPE ][METHOD] FRAME

```

10 记录每个逼近点

- a 移动光标到每个逼近点
- b 手动进给机器人到你所要记录的点
- c 按下并保持SHIFT键，并按F5,RECORD 记录下当前位置的数据作为逼近点。对于已经教导过的参考点，会显示RECORDED。

```

:
FRAME  MOVE_TO RECORD
                                SETUP Frames                                JOINT 30 %
                                Orient Origin Point: RECORDED
                                X Direction Point:    RECORDED
                                Y Direction Point:    UNINIT
                                Active UFRAME $MNUFRAMNUM[1]=0
                                [ TYPE ][METHOD] FRAME  MOVE_TO RECORD
                                F5

```

d 教导了所有参考点后，显示USED。工具参照系设置完成。

```

SETUP Frames                                JOINT 30 %
User Frame Setup/ Three Point              4/4
Frame Number: 1
X: 1243.6  Y: 0.0  Z: 10.0
W: 0.1    P: 2.3  R: 3.2
Comment:  REFERENCE FRAME
Orient Origin Point: USED
X Direction Point:  USED
Y Direction Point:  USED
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME  MOVE_TO RECORD

```

11 要移动到一个记录的位置，按住并保持SHIFT键，然后按F4,MOVE_TO 。

```

FRAME  MOVE_TO RECORD

SHIFT

```

12 要看每个记录位置数据，移动光标到每个参考位置项目，按ENTER键。显示每个位置数据的位置详细内容界面。要返回上一个界面，请按PREV键。

13 要显示用户参照系目录，请按PREV键。可以参看所有用户参照系的设置。

PREV

SETUP Frames				JOINT	30 %
User	Frame	Setup/	Three	Point	1/9
	X	Y	Z	Comment	
1:	1243.6	0.0	43.8	REFERENCE FR>	
2:	0.0	0.0	0.0	*****	
3:	0.0	0.0	0.0	*****	
4:	0.0	0.0	0.0	*****	
5:	0.0	0.0	0.0	*****	
6:	0.0	0.0	0.0		
7:	0.0	0.0	0.0		
8:	0.0	0.0	0.0		
9:	0.0	0.0	0.0		
Active UFRAME \$MNUFRAMNUM[1]=0					
[TYPE] DETAIL [OTHER] CLEAR SETIND					

14 激活用户参照系设置，请按 (SETIND)，然后输入参照系号。

[OTHER] CLEAR SETIND



注意

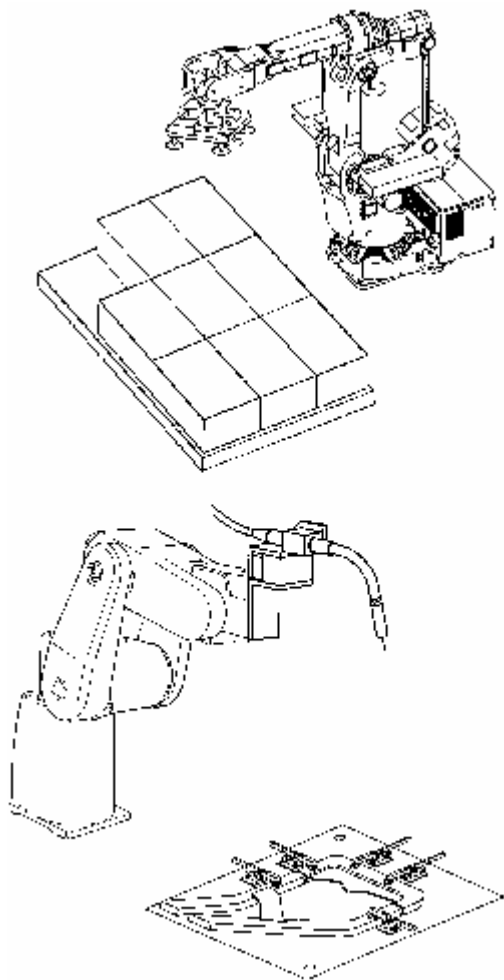
设置完所有坐标系后，应该在外部存储设备（例如磁盘）上记录设置信息，以便重新载入这些信息。一旦改变，当前设置信息将丢失。

15 要删除设置的参照系的数据，请移动光标到所需的参照系，按F4选择CLEAR.

[OTHER] CLEAR SETIND

设置参考位置

参考位置是一个固定（预先设定的）的位置，它经常在一个程序中或者用手动进给运行机器人时使用。参考位置是一个安全位置，它通常远离机器人工具或者外围设备的操作区。可以定义3个参考位置。



当机器人在参考位置时，会输出一个预先设定的信号，SDO。如果参考位置无效，会输出一个DO信号。

当机器人在参考位置1，会输出外围设备I/O参考位置输出信号(ATPERCH)。

对于这个功能，可以用过参考位置设置，使其不输出。

要移动机器人到参考位置，构造指定返回路径的程序，并执行它。同时，程序中要指明各轴返回参考位置的顺序。此外，最好设置返回程序宏指令

在参考位置设置界面[6 (SETUP). Ref Position]指定参考位置。

特殊区域功能

当发出了一个移动指令，导致机器人要进入预设的干涉区域，特殊区域功能会自动停止机器人；如果另外一个机器人或者外围设备位于干涉区域，特殊区域功能会停止机器人；当确认其他机器人或者外围设备已离开干涉区域，会自动解除机器人停止状态，并重新启动其操作。

用一系列互锁信号（一个信号对应每个输入和输出信号），建立机器人和外围设备的通信。设置一个互锁信号到一个干涉区域。最多可以定义3个干涉区域。

互锁信号和机器人的关系如下所述。

输出信号

当工具端点位于干涉区域时，输出信号停止。当工具端点不在干涉区域时，输出信号开启。

状态	输出信号
安全（safe）：工具端点不在干涉区域	开
危险（dangerous）工具端点位于干涉区域	关

输入信号

当输入信号关闭，机器人正试图要进入干涉区域时，机器人进入保持状态。当输入信号打开，机器人从保持状态解除，自动重启其操作。




注意

在工具端点进入干涉区的点，机器人会减速到停止，这样机器人实际上在干涉区内一点停止。机器人运行速度越快，进入干涉区越深。考虑到以上和其他一些因素，例如工具尺寸，需要保证设置一个足够大的干涉区。

要设置特殊区域功能，用**SETUP Space**功能（设置空间）。

用Space/DETAILED SCREEN(空间详细内容)界面设置如下项目:

特殊区域功能项目 (Area Details Screen区域详细内容界面)

项目	功能
Enable/disable (使能)	使该功能有效或无效。要改变其他项目的设置, 必须先使该区域中, 要设置的项目功能无效。
Comment (注释)	允许用户输入一个最多10个字母的注释。
Output signal (输出信号)	设置输出信号
Input signal (输入信号)	设置输入信号
Priority	<p>当两个机器人用这个功能时, 此项会在两个机器人在同一时间, 都试图要进入干涉区域时, 指明哪个机器人会先进入干涉区域。设置为High的机器人会先进入干涉区。当机器人完成其操作并移出干涉区时, 设置为Low的机器人进入干涉区。机器人之间的设置必须不同。</p> <p>提示: 如果两个机器人都设置了High或者Low, 机器人试图要同时进入干涉区, 它们都会进入停止(死锁)状态。这种情况发生后, 运行如下所述的恢复操作, 并检查设置是否正确。</p> <ol style="list-style-type: none"> 1 对两个机器人都运行紧急停车。 <div style="border: 1px solid black; padding: 10px; margin: 10px 0;">  警告: 如果两个机器人没有紧急停车, 当一个移出干涉区时, 一个机器人会自动启动其操作。这是非常危险的。 </div> <ol style="list-style-type: none"> 2 检查是否有能被机器人碰到的物体或者旁观者。 3 禁用该功能 4 用手动操作, 移动其中的一个机器人移出干涉区。
inside/outside (内外界)	指明长方体的内界或外界是否为干涉区域。

要设置以下项目, 请用Rectangular Space/SPACE SETUP设置界面。

特殊区域功能项目 (Area Setting Screen区域设置界面)

项目	描述
BASIS VERTEX	成为参考的长方体的顶点位置。
SIDE LENGTH/SECOND VERTEX	<p>如果选择了SIDE LENGTH, 指明了在用户坐标系下按照X, Y, 和Z轴, 从参考定点到长方体边的长度。</p> <p>(长方体的边必须平行于用户坐标系各自的轴。) 如果选择了SECOND VERTEX, 在这定义的, 有参考定点和对角定点的长方体, 成为一个干涉区域。</p>

设置特殊区域功能

- Step 1 按MENUS键，显示屏幕菜单。
- 2 选择SETUP.
- 3 按 F1 "TYPE." 显示界面切换界面。
- 4选择Space fnct. 显示区域目录界面。

区域目录界面

Area List Screen			
LIST SCREEN			
No.	End/Dsbl	Comment	Usage
1	ENABLE	[Common Space
2	DISABLE	[Common Space
3	DISABLE	[Common Space

Space fnct.

TYPE

[TYPE]	DETAIL	ENABLE	DISABLE
--------	--------	--------	---------

5 用合适的功能键，区域目录界面允许用户禁用或激活每个干涉区域。用以下步骤，输入注释：

- a. 移动光标到要注释的行，按下Enter键。
- b. 指定用哪个字母还是片假名字符输入注释。
- c. 按字母功能键，输入注释。
- d. 输入完注释后，请按Enter键。

Rectangular Space			
LIST SCREEN			
No.	Enb/Dsbl		
1	ENABLE		
2	DISABLE	[
3	DISABLE	[

ENTER

6 除了要设置Enb/Dsbl 或者 Comment以外，请按F3 (DETAIL). 显示详细界面。

Rectangular Space			
DETAILED SCREEN			
SPACE:1		GROUP:1	
USAGE:		Common Space	
1 Enable/Disable:		ENABLE	
2 Comment:		[*****]	
3 Output Signal:		DO[0]	
4 Input Signal:		DI[0]	
5 Priority:		High	
6 Inside/Outside:		Inside	

DETAIL ENABLE DISABLE

[TYPE]	SPACE	ENABLE	DISABLE
--------	-------	--------	---------

7 移动光标到需要的项目。用功能键或者数字键改变项目的设置。

8 要设置一个区域，按 SPACE.键，显示区域设置界面。

[TYPE] SPACE	SPACE SETUP		%
	SPACE:1	GROUP:1	1/4
	UFRAME:0	UTOOL:1	
	1:BASIS VERTEX	[SIDE LENGTH]	
	2:X 0.0 mm	0.0 mm	
	3:Y 0.0 mm	0.0 mm	
	4:Z 0.0 mm	0.0 mm	
[TYPE] OTHER		RE10RD	

9 可以用以下两种方法之一，设置参考定点和边长或者对角定点。T

- a.移动光标到X,Y和Z坐标区，并直接用数字键，输入想要的坐标。
- b. 移动机器人到一个长方体的定点，然后用SHIFT 键+F5 RECORD读取当前的位置。

RE10RD

F5 #74

提示 如果改变了UF或者UT,先运行b。通过这个操作，选择了当前UF或者UT值。

提示 当用户坐标系值改变时，干涉区的空间位置不变。当用户坐标系值已经改变，干涉区域要在新的坐标系重新定义，按SHIFT 键+F5 RECORD，重新设置干涉区。

10 区域设置完后，按PREV，显示区域详细界面。要反活区域目录界面，请再按PREV一次。


PREV

设置通用项目

[6 SETUP General] 有如下项目

- Break on hold
- Current language
- Ignore Offset command
- Ignore Tool--offset

设置通用项目

项目	描述
Break on hold	<p>指明当HOLD键按下时，是否产生一个警报并关闭伺服系统警报。</p> <p>—如果该功能无效（DISABLED），用HOLD键停止操作时，没有警报产生(标准设置)。</p> <p>—如果该功能有效（ENABLE），用HOLD键停止操作时，产生警报并关掉伺服系统电源。</p> <p>要使该功能有效，需要重新开启电源。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> 警告：</p> <p>不是所有轴都装备了制动器。即便该功能有效，在一个没有制动器的轴上，保持功能的制动时没有效果的，在保持功能的制动有效设置前，请仔细检查哪个轴有制动器。否则，会引起人身伤害。</p> </div>
Current language	标准设置中，当前语言设置为“DEFAULT”。在特殊工作情况下，改变当前当前语言。通常，使用标准设定。
Ignore Offset command	<p>指明是否忽略补偿指令</p> <p>—该功能无效时（DISABLED），机器人移动到执行补偿后，指定的位置（标准设置）</p> <p>—该功能有效时（ENABLED），机器人移动到教导的位置（没有执行补偿指令）</p>
Ignore Tool--offset	<p>指明是否忽略工具补偿指令</p> <p>—该功能无效时（DISABLED），机器人移动到执行补偿后，指定的位置（标准设置）</p> <p>—该功能有效时（ENABLED），机器人移动到教导的位置（没有执行补偿指令）</p>

程序结构

本章介绍了程序结构和程序指令。

一个操作应用程序由用户代码指令和其他辅助信息组成，用户代码指令用于执行操作。程序包含程序信息和程序详细信息，其中程序信息说明操作怎样执行，而程序详细信息定义程序属性。

程序信息界面

Program detail		JOINT 30 %
		1/6
Creation Date:	10-MAR-1998	
Modification Date:	11-MAR-1998	
Copy Source:	[*****]	
Positions: FALSE	Size:	312 Byte
1 Program name:	[SAMPLE3]	
2 Sub Type:	[None]	
3 Comment:	[SAMPLE PROGRAM 3]	
4 Group Mask:	[1, *, *, *, *]	
5 Write protect:	[OFF]	
6 Ignore pause:	[OFF]	
END PREV NEXT		

程序详细信息包括下列信息项:

- 属性相关信息项，比如创建日期，修改日期，原文件名，起始和终止位置数据和程序数据大小。
- 执行环境相关信息项，比如程序名，子类型，注释，组标记，写保护和中断屏蔽。

程序选择界面

Memory available capacity		Select		JOINT 30 %
		58740 bytes free		1/7
		No.	Program name	Comment
		1	SAMPLE1	[SAMPLE PROGRAM 1]
		2	SAMPLE2	[SAMPLE PROGRAM 2]
		3	SAMPLE3	[SAMPLE PROGRAM 3]
		4	PROG001	[PROGRAM001]
		5	PROG002	[PROGRAM001]
		6	CLAMP1	[CLAMP OPEN]
		7	CLAMP2	[CLAMP CLOSE]
Program name		[TYPE]		Attribute
		[TYPE] CREATE DELETE MONITOR [ATTR] >		
		COPY DETAIL LOAD SAVE PRINT >		

程序编辑界面

Program name		- SAMPLE1	JOINT 10%
			1/8
Line number		-1: J P[1] 100% FINE	
Program statement		2: HANDICLOSE	
➤ Motion instruction		-3: J P[2] 70% CNT50	
➤ Macro instruction		4: L P[3] 500mm/sec CNT10	
		-5: HANDIOPEN	
		6: L P[4] 500mm/sec CNT10	
		7: HANDICLOSE	
➤ Abort instruction		-8: ABORT	
Program end symbol		-[End]	
		POINT	TOUCHUP >

Program name	SAMPLE1	JOINT 30 %
Line number	1:J P[1] 100% FINE	1/8
Motion instruction	2:J P[2] 70% CNT50	
Program statement	3:L P[3] 500mm/sec FINE	
Arc welding instruction	: Arc Start[1]	
	4:L P[4] 50mm/sec CNT80	
	5:L P[5] 50mm/sec CNT80	
	: Arc End[50.0V,75.0A,0.1s]	
	6:J P[1] 100% FINE	
Abort instruction	7: ABORT	
Program end symbol	{End}	
	[INST]	[EDCMD]>
	POINT ARCSTRT WELD_PT ARCEnd TOUCHUP>	

程序由下列信息组成:

- 指向每个程序命令的行号
- 指定机器人如何运动以及运动到哪的运动指令。
- 程序指令包括一下内容:
 - 执行堆积功能的堆积指令
 - 控制电弧焊接的电弧焊接指令
 - 将数据存入寄存器的指令（寄存器指令）
 - 将机器人位置数据存入位置寄存器的指令（位置寄存器指令）
 - 将信号从外围设备输入和将信号输出到外围设备的输入输出指令
 - 当已定义的条件被满足时，改变程序控制流的分支指令
 - 将执行程序挂起的等待指令
 - 直到接收到信号才能操作机器人的跳转条件指令。如果没有接收到信号，那么一个指向特定命令的分支将被执行。如果接收到了信号，那么下一条指令将被执行，并且取消操作。
 - 程序注释
 - 其他指令
- 程序结束标志说明这个程序不再有其他指令了。

序名

指定程序名是为了识别存入控制器内存中的程序。一个单独的控制器不能包含两个或是更多拥有相同程序名的程序。

长度

程序名必须包含 1 到 8 个字母。每个程序必须对应一个独一无二的程序名。

可用的字符

字符：字母字符，数字：0 到 9。程序名不能以数字开头。符号：只能用下划线（_）。不能使用 at 符号（@）和星号（*）。

信息性命名

程序应该这样命名：从程序的名字就可以知道程序的目的或功能。例如：当给一个用于查找工件 A 的程序命名时，“HAND_A”就是一个赋给这个程序的好名字，因为它指出了这个程序的功能。

提示 当为使用 RSR 或 PNS 的自动操作编写程序时注意观察以下内容。否则，程序无法运行。

- 使用 RSR 的程序必须命名为 RSRnnnn，其中 nnnn 是一个 4 个阿拉伯数字组成的数。例如：RSR0001.
- 使用 PNS 的程序必须命名为 PNSnnnn，其中 nnnn 是一个 4 个阿拉伯数字组成的数。例如：PNS0001.

例如：为工件 No. 1 的电弧焊接程序命名为 WELD_1，使得任何一个人都能猜测到这个程序的内容。

程序注释

当新程序创建时，可把程序注释加到程序名中。程序注释连同程序名一起来描述在选择界面上显示的附加信息。

长度

程序注释必须由 1 到 16 个字符组成。

可用字符

字符：字母字符，数字：0 到 9，符号：下划线 (_) , at 符号 (@) 和星号 (*) .

信息性注释

程序注释必须描述程序的目的或功能。

写保护

写保护指定了该程序可否被修改。

- 当此项为 ON 时，任何数据都不能添加到程序中，并且程序不能被修改；这就是说，程序被写了保护。当程序被创建后，并且它的操作确定后，用户可将此项设置为 ON 来保护程序防止其被用户本人或是其他人修改。

提示 当此项为 ON 是，在程序详细信息中的其他项不能被改变。

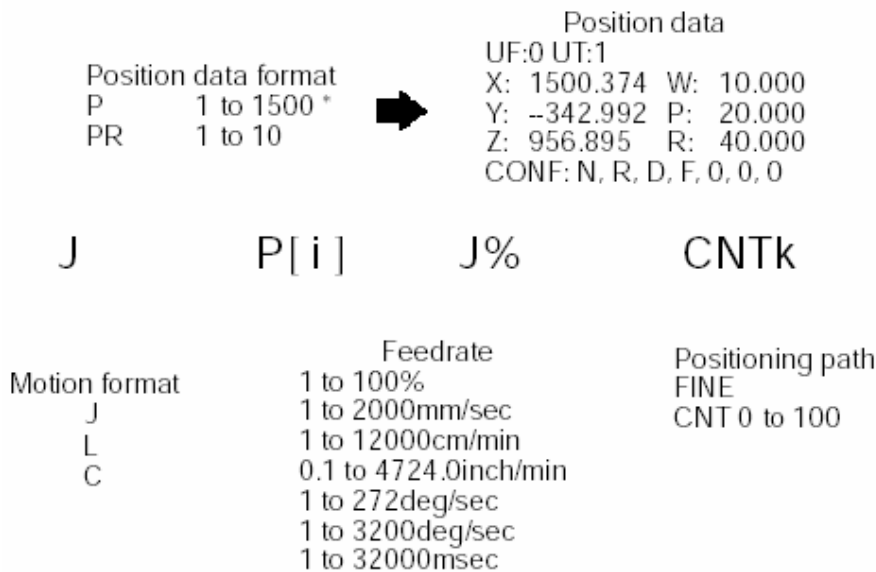
- 当此项被设置为 OFF 时，程序可以被修改；这就是说，程序指令可以添加到程序中，并且已经存在的指令可以被修改。按照规定，写保护常被设置为 OFF。

动作指令

动作指令推动 robot tool 在操作区域范围内，以指定的进给速度和指定的运动方式到指定地点。在动作指令中必须指定下面列出的各项。图 4-5 显示了动作指令的形式。

- 动作形式：确定如何控制到指定位置的动作路径
- 位置数据：教导机器人所要驱动到的位置。
- 进给速度：指定机器人的进给速度。
- 定位路径：指定是否将机器人放置到指定位置
- 附加动作指令：在机器人动作过程中，指定附加指令的执行。

动作指令



* A position number can be as large as the memory capacity allows.

动作类型

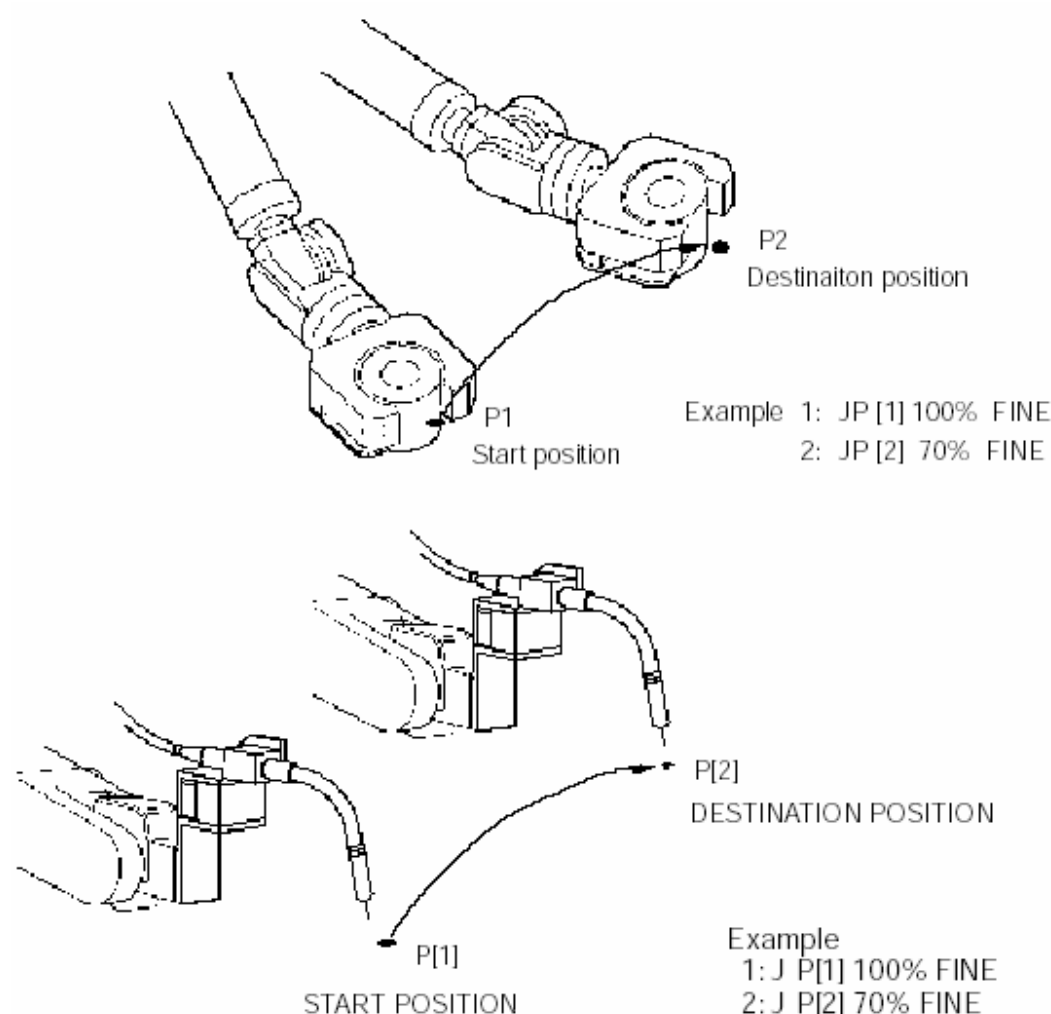
关于动作类型，它确定了到指定位置的动作路径。有三个可用选项：关节动作，它不使用路径控制和姿态控制，以及线性动作和圆周动作，它们需要使用路径控制和姿态控制。

- 关节动作 (J)
- 线性动作 (包括选择动作) L)
- 圆周动作 (C)

关节动作 J

关节动作模式是驱动机器人到指定位置的基本动作模式。机器人以指定进给速度，沿着所有轴的方向或者围绕所有轴的方向，同时加速，减速，或停止。动作路径通常是非线性的。为了教导终点需指定动作类型。最大进给速度的百分数被指定为关节动作的进给速度。没有控制被驱动的工具的姿态。

关节动作



start position:起始位置

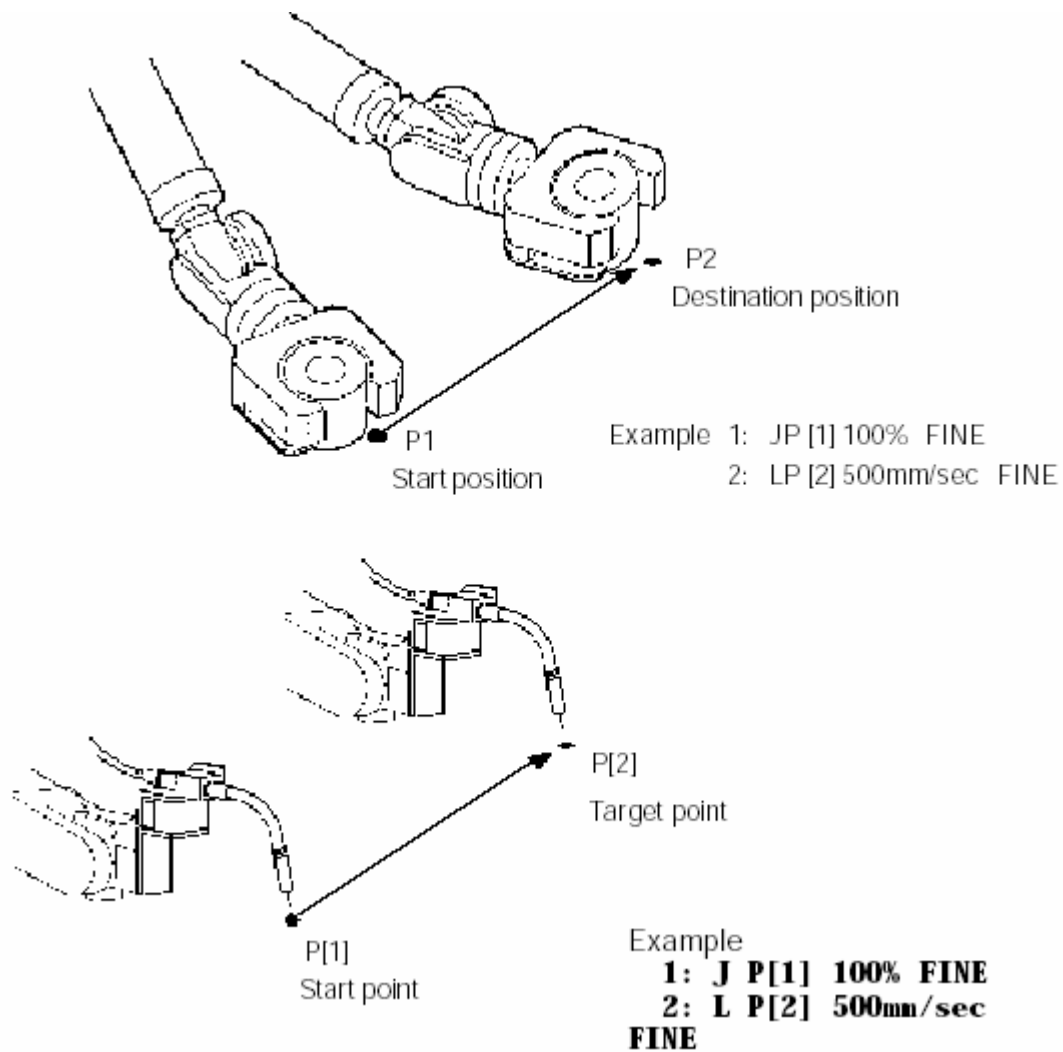
destination position:目标位置

example:例如

线性动作 L

线性动作模式控制工具中心点从起始到终点的动作路径(TCP)；工具中心点线性运动。关于线性进给速度的指定，一个期望的选项必须在 mm/sec (毫米/秒), cm/min (厘米/分) 和 inch/mm(英寸/毫米)中选出。被驱动工具的姿态可通过区别起点时的姿态和终点时的姿态加以控制。

线性动作



start point: 起始点

target point: 目标点

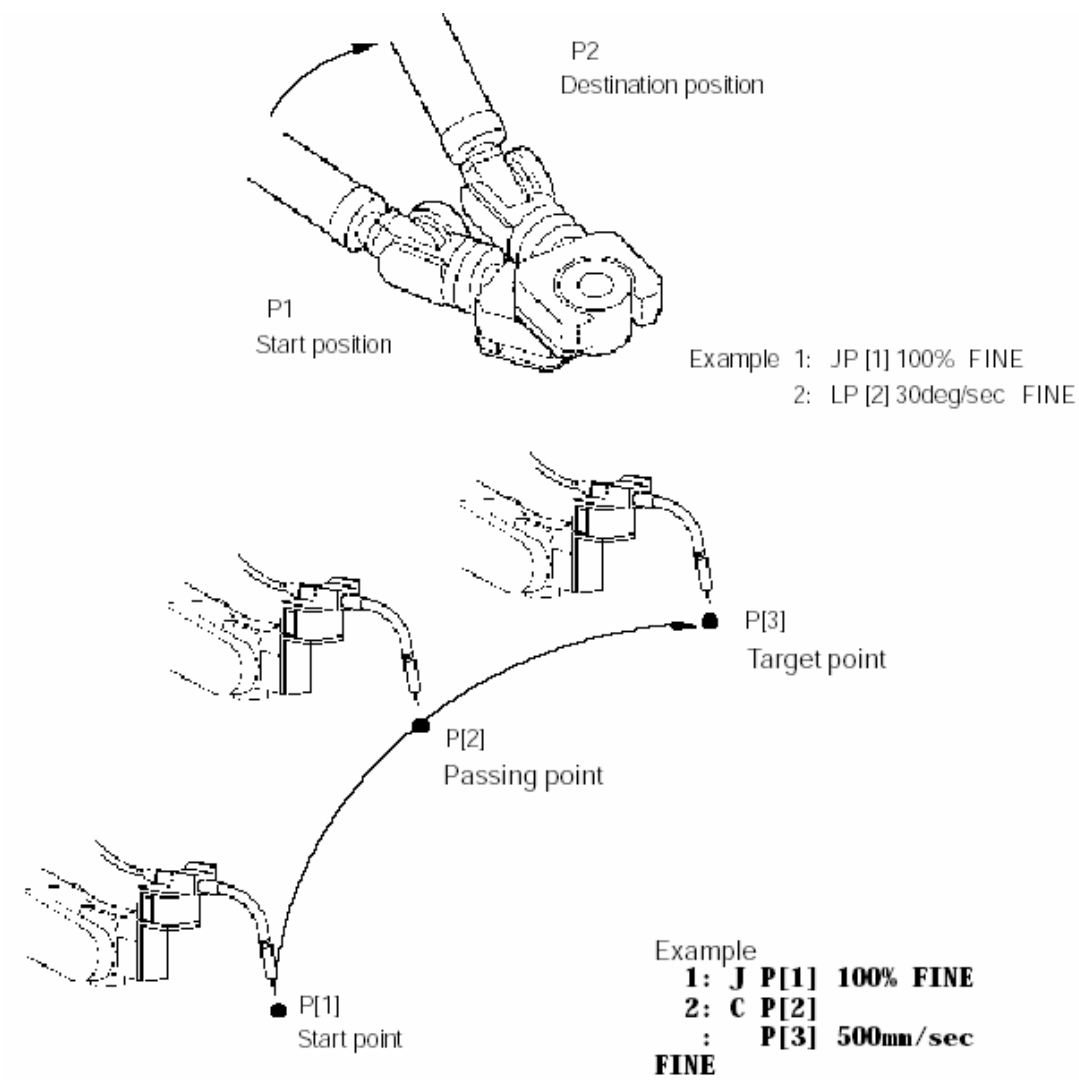
example : 例如

start position: 起始位置

destination position: 目标位置

旋转动作是一种动作方法，在旋转动作中，工具以线性动作方式从起始位置到终止位置绕工具的终点旋转。在动作过程中，工具的方向是通过区分起点位置的方向和终止位置的方向来控制的。进给速度指定为 deg/sec（度/秒）。焦点是控制线性（当工具终点移动时）。

旋转动作

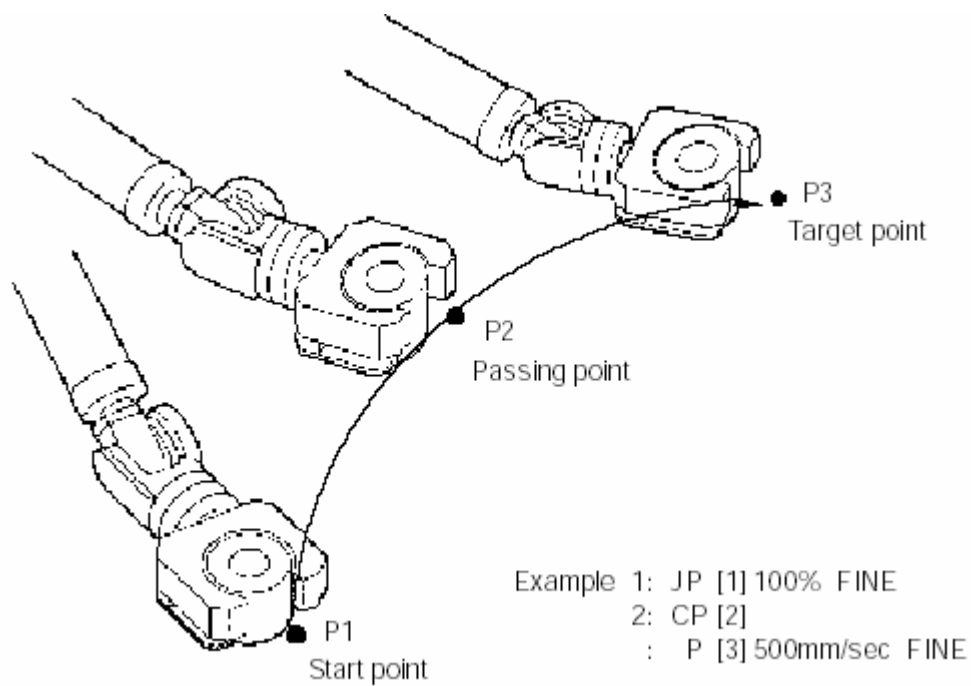


start point: 起始点 passing point: 相会点 target point: 目标点
start position: 起始位置 destination position: 目标位置 example : 例如

圆周动作

圆周动作模式控制工具中心点的从起始点经过相会点到终止点的动作路径。
相会点和目标点在同一个指令中给出。
关于圆周进给速度的指定，一个期望的选项必须在 mm/sec（毫米/秒）,cm/min（厘米/分）和 inch/mm(英寸/毫米)中选出。
被驱动工具的姿态可通过区别起点时的姿态和终点时的姿态加以控制。

圆周动作



start point: 起始点
passing point: 相会点
target point: 目标点
example : 例如

位置数据

位置数据包括位置和机器人的姿态。当教导动作指令时，同时位置数据写入了程序。

位置数据被划分为两种类型。一种是包含在关节坐标系下的关节坐标。另一种类型包含表征工具在工作区域内的位置和工具的姿态笛卡尔坐标。标准的位置数据使用笛卡尔坐标。

笛卡尔坐标

包含笛卡尔坐标的位置数据由四个元素来定义：在笛卡尔坐标系下工具中心点的位置（工具坐标系的原点），驱动工具时，所沿轴的倾角（工具坐标系），配置和使用的笛卡尔坐标。笛卡尔坐标系可能是全局坐标系。如何选择笛卡尔坐标系在这个分段的后面做了解释。位置数据（笛卡尔坐标）

$$\begin{array}{ccccccc} \text{UF} & , & \text{UT} & , & (X, Y, Z) & , & (W, P, R) & , & \text{Configuration} \\ \hline \text{User coordinate} & & & & \text{Position} & & \text{Attitude} & & \text{Configuration} \\ \text{system number} & & & & & & & & \\ & & \text{Tool coordinate} & & & & & & \\ & & \text{system number} & & & & & & \end{array}$$

user coordinate system number: 用户坐标系序号

tool coordinate system number: 工具坐标系序号

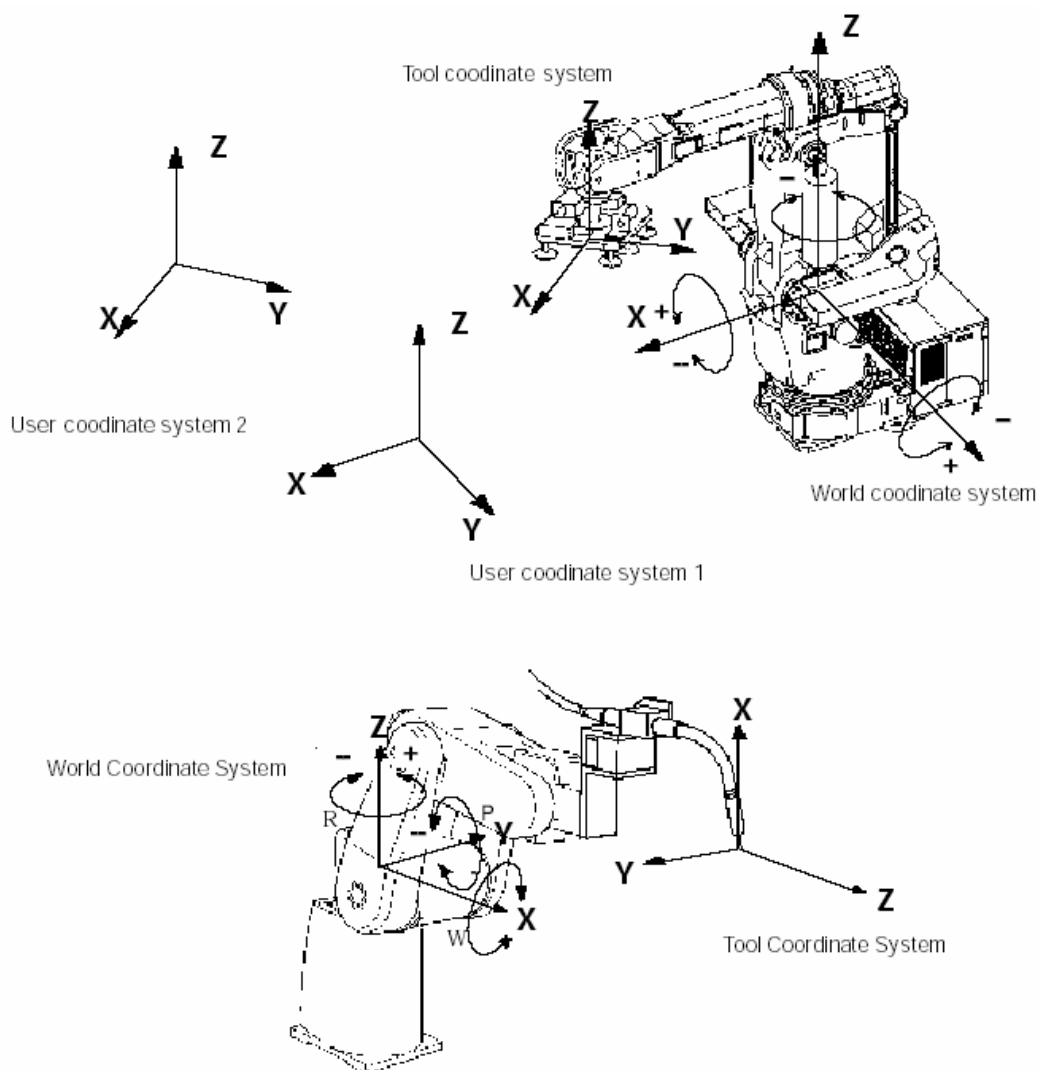
position: 位置

attitude: 姿态

configuration: 配置

位置和姿态

- 位置 (x,y,z) 代表在笛卡尔坐标系下工具中心点（工具坐标系的原点）的三维位置。
- 姿态 (w,p,r) 代表在笛卡尔坐标系下，绕 X—轴，Y—轴，Z—轴的角位移。



配置

配置指机器人的姿态。有几种配置是可用的，它们满足笛卡尔坐标（x,y,z,w,p,r）的情况。必须指定每个轴的转数和关节定位。

进给速度

进给速度指机器人运动的速度。在程序执行过程中，进给速度通过进给速度倍率控制。进给速度倍率值可以使用 1%到 100%。用来指定进给速度的单位取决于所教导的动作指令的动作类型。

提示 编写的运动速度不能超过机器人的许可范围。如果编写了一个超出变化范围的速度，将会发出警告。

J P[1] 50% FINE

当动作类型为关节动作时，进给速度的定义如下：

- 指定最大进给速度的百分数，这个百分数从 1%到 100%之间的一个百分数。
- 当单位为秒时，指定值 0.1 到 3200 秒作为执行动作所需要花费的时间。当执行动作所需要花费时间很重要时，这种指定是必要的。操作有时不能在指定的时间内

发生。

- 当单位为毫秒时，指定其值 1 到 32000 毫秒作为执行动作所需要花费的时间。

L P[1] 100mm/sec FINE

如果指定的动作类型为线性动作或者圆周动作时，进给速度的定义如下：

- 当单位为毫米/秒时，指定进给速度从 1 到 2000 毫米/秒。
- 当单位为厘米/分钟时，指定进给速度从 1 到 12000 厘米/分钟。
- 当单位为英寸/分钟时，指定进给速度从 0.1 到 4724.4 英寸/分钟。
- 当单位为秒时，指定其值 0.1 到 3200 秒作为执行动作所需要花费的时间。
- 当单位为毫秒时，指定其值 1 到 32000 毫秒作为执行动作所需要花费的时间。

L P[1] 50 deg/sec FINE

如果指定的动作类型为绕工具中心点的旋转动作时，指定角位移如下：

- 当单位是度/秒时，指定角位移 从 1 到 272 度/秒。
- 当单位为秒时，指定其值 0.1 到 3200 秒作为执行动作所需要花费的时间。
- 当单位为毫秒时，指定其值 1 到 32000 毫秒作为执行动作所需要花费的时间。

用寄存器指定进给速度

进给速度可以用寄存器指定。这就允许用户在用寄存器计算好进给速度后为动作指令指定进给速度。进给速度还可以通过使用如：组输入（GI）或者数据传输从外表指定。



注意

这个功能允许用户通过使用寄存器改变机器人的进给速度。这就意味着机器人可能以一种出乎意料的速度动作，这个速度取决于指定的寄存器值。因此，当使用这个功能时，在教导或是操作过程中都必须小心的指定寄存器值。

当用寄存器指定进给速度时，操作指令的显示类型

- 关节 J P[1] R[i]% FINE
- 线性 L P[1] R[i]毫米/秒 FINE
- 电弧 C P[1]
P[2] R[i]毫米/秒 FINE
- 堆垛操作指令
J PAL_1[A_1] R[i]% FINE
J PAL_1[BTM] R[i]% FINE
J PAL_1[R_1] R[i]% FINE

提示 堆垛操作指令是堆垛的一个软件选
操作组指令

异步操作组

GP1 JP[1] R[i]% FINE

GP2 JP[1] R[i]% FINE

提示 操作组指令是一个多动作的软件选项。

标准操作指令的进给速度也是支持的。

查找/替换功能。

— 查找功能

不支持查找功能。

不能完成使用寄存器选项的查找功能。

— 替换功能

使用注释编辑项可以完成替换功能。

不能完成使用寄存器项的替换功能。

不支持操作附加指令的辅助轴进给速度。

在程序编辑中，进给速度（寄存器值）没有进行范围核对。

当进给速度单元改变时，进给速度（寄存器值）不能自动转换。

如果操作注释的进给速度是用寄存器指定的，那么执行的头部的读入就会停止。（可以确定是否需要使用系统的变量停止头部的读入。关于这一点后面进行了描述。）

如果输入寄存器的值不在上限和下限之间，或者如果值的类型不是适合进给速度的类型（整数/实数），在执行过程中就会发出警报。

单位	许可范围
%	1 到 100 整数
秒	0.1 到 3200.0 (*1) 实数/有效为数多至第一个小数点的位置
毫秒	1 到 32000 (*1) 整数
毫米/秒	1 到 2000 (*1) 整数
厘米/分钟	1 到 12000 (*1) 整数
英尺/分钟	0.1 到 4724.2 (*2) 实数/有效为数多至第一个小数点的位置
度/秒	1 到 272 (*3) 整数

机器人的类型决定了许可范围（最大值）。

可以激活头部的读入。

如果操作注释的进给速度是用寄存器指定的，那么执行的头部的读入就会停止。可以确定是否需要使用下列系统的寄存器停止头部的读入。默认的设置是 FALSE（头部的读入就会停止）。

提示 如果寄存器进给速度的头部的读入用上面的系统的变量激活了，那么，新值可能不能从操作速度中反映出来，导致机器人仍以原来的速度值驱动，这取决于寄存器值改变的时间。如果寄存器进给速度的头部的读入被激活了，就有必要采取适当的方法，比如：在程序的执行过程中锁定或者不改变用于进给速度寄存器的值。

```
10: R [1] = 100  
11: J P[5] R[1]% FINE  
12: R[1]=10  
14: J P[6] R[1]% FINE
```

如果寄存器进给速度的头部的读入被激活，在第 10 行为 100，第 12 行不是 10，用于操作第 14 行上的速度。

定位路径

定位路径指定了在动作指令中结束机器人操作的方法。有以下三种可用的定位路径模式：

- FINE 定位路径
- CNT 定位路径
- 指定转角距离：只有当需要使用高精度对焦功能时，它才是可用的。关于转角距离的解释，见 4.3.6.

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，停止在这个目标点上。

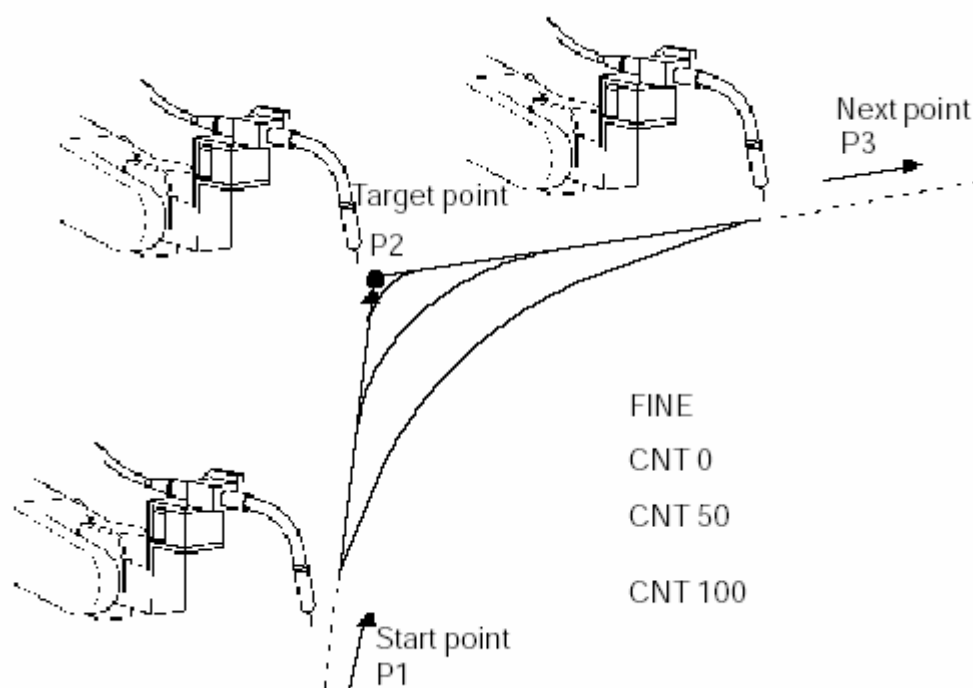
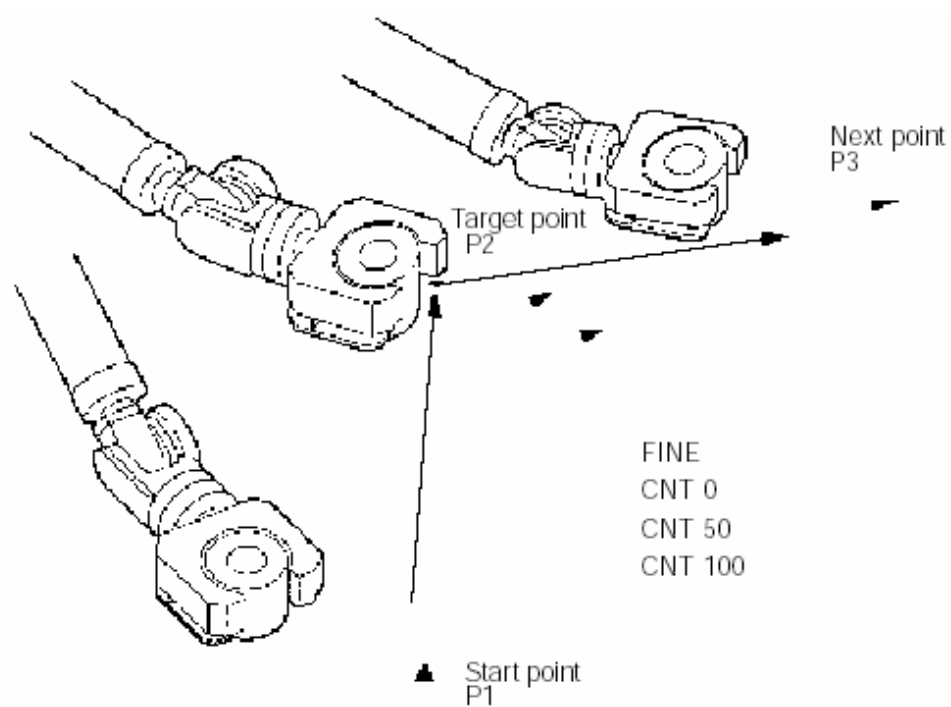
CNT 定位路径 J P[i] 50% CNT 50

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驱动。当指定值为 100 时，机器人沿着最远的路径向目标点驱动，这是因为机器人不再目标点附近减速并且不久它又向下一个目标点驱动了。

提示 当教导如等待指令这样的指令时，机器人停止在目标点上来执行该指令。

提示 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 值为 100 时，也可以减速。

使用 continuous termination 类型的机器人动作路径



附加动作指令

附加动作指令使得机器人完成特殊的任务。下列附加动作指令是可用的：

- 腕关节动作指令（Wjnt）
- 跳转指令（Skip, LBL[i]）
- 补偿条件指令（Offset, PR[i]）
- 工具补偿指令（Tool_Offset, PR[i]）
- 增量指令（INC）
- 同时 EV 指令（EVi%）
- 相互独立 EV 指令（ind.EVi%）
- 路径指令（PTH）
- 预执行指令（预执行/后执行）

当教导附加动作指令时，在动作指令后移动指针，然后按 F4[CHOICE]来显示附加动作指令的列表。然后选择期望的附加动作指令。

JOINT 30%		Motion Modify		JOINT 30%
4/5		1 No option		5 Incremental
500mm/sec CNT10		2 Wrist Joint		6 Skip, LBL[]
CHOICE		3 Offset/Frames		7
		4 Offset, PR[]		8
		PROGRAM1		

腕关节动作指令 LP[i] 50% FINE Wjnt

Motion Modify	
1 No option	5 In
2 Wrist Joint	6 Sk
3 Offset	7
4 Offset, PR[]	8

腕关节动作指令指定路径控制操作，它不控制腕关节的姿态。（在标准模式下，腕关节的姿态只在动作结束时才被控制。）当指定为线性动作或者圆周动作时，才指定腕关节动作指令。

当使用腕关节动作指令时，在动作过程中，腕关节的姿态是改变的。然而，工具的中心点可以沿着已经编写的路径移动，而不会因为腕关节轴奇点导致腕关节轴的颠倒。

Acceleration override

J P[1] 50% FINE ACC80

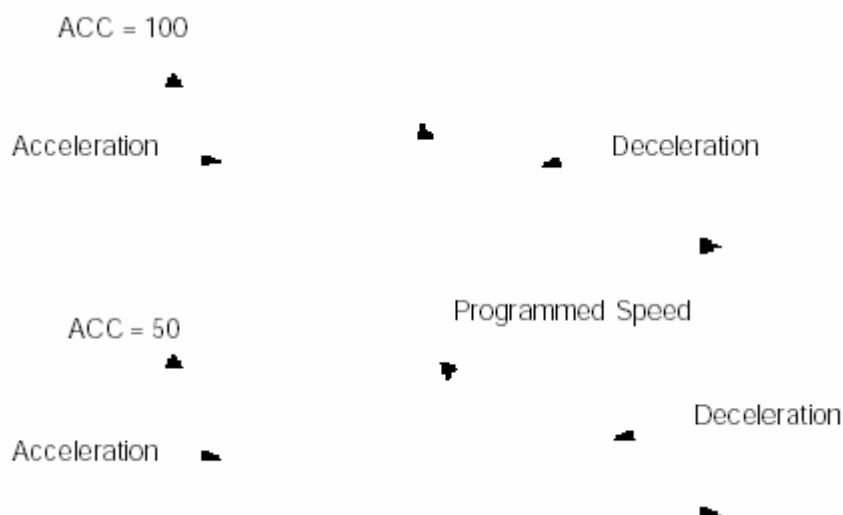
```
Motion modify
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
PROGRAM1
```

acceleration override: 加速倍率

这个指令在指定动作工程中的加速率/减速率的百分数。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）。要执行一个具有潜在危险的操作比如在热水中捞球，使用的 ACC 值小于 100%。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）。对于那些操作感觉非常慢的部分，可以使用 ACC 的值大于 100%。

从起点到目标点，用于执行动作的时间取决于加速倍率。加速倍率值在 0%到 150%之间变化。在目标位置处编写加速倍率。

加速倍率



注意

如果加速倍率很大，可能会发生剧烈的运动和颤动。这将引起伺服警报。如果是因为增加了加速/减速倍率指令的操作指令而导致上述情况的发生，那么，要么减小加速/减速倍率值，要么删除加速/减速倍率指令。

Skip instruction

SKIP CONDITION [I/O] = [value] J P[1] 50 FINE Skip,LBL[3]

JOINT 30%

5 Incremental

6 Skip,LBL[]

7

8

skip instruction: 跳转指令

如果跳转条件不满足时，跳转指令使得分支目标标记（branch destination label）跳转。在机器人向目标点驱动过程中，如果跳转条件满足时，机器人取消该动作而且程序执行下一行的程序注释。如果跳转条件不满足时，程序在完成机器人的动作后跳转到分支目标标记所在行。

跳转条件指令提前指定要使用的跳转条件（执行跳转指令的条件）。在跳转指令执行前，必须执行跳转条件指令。跳转条件一旦指定，在程序执行完成前或者执行下一个条件指令前都是有效的。

增量指令

Incremental instruction

J P[1] 50% FINE INC

JOINT 30 %

5 Offset

6 Offset, PR[]

7 Incremental

8 ---next page---

增量指令将动作指令中的位置数据用作当前位置的增量,并且使得机器人向目标位置驱动,其中目标位置指在当前位置加上增量的数值。这就意味着从当前位置的增量动作的数值存储在动作指令的位置数据中。

增量指令通过以下元素指定:

- 当位置数据为关节坐标值时,提供了每个轴的增量数值。
- 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。无论如何,坐标系是被核对过的。关于笛卡尔坐标系基准,见 4.3.2)
- 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。
- 当 INC 指令和 OFFSET 指令同时使用时,动作指令中位置数据的类型应该和补偿所使用的位置寄存器的类型一致。在这种情况下,补偿数量被用作指定的增量数值的补偿数量。

高精度轨迹功能（选项）

高精度轨迹功能改善机器人的动作执行，其中，动作指具有 CNT 或转角距离指定的线性动作和圆周动作。不管这个功能是否提供线性位置操作和关节操作都是相同的。

- 定值轨迹

通过使用高精度轨迹功能，机器人不使用倍率，沿着定值轨迹驱动。关于具有较低的倍率值的程序的教导和调试，机器人沿着相同的轨迹驱动，即使当倍率值为 100% 时也是如此。

- 在临时停止或紧急停机后重新启动

当临时停止或紧急停机后的重新启动执行时，这个功能使得机器人驱动所沿着的轨迹和不停止时的驱动所沿着的轨迹相同，然而方位可能会略有差别。

- 改善轨迹精度

这个功能改善线性动作和圆周动作的轨迹精度。

- 常速度值

这个功能使得机器人沿圆周驱动时保持指定速度。如果由于机器人自身的机械装置的操作性能的原因使得机器人驱动时不能保持指定速度，那么，速度将自动的减小。如果沿圆周驱动的速度减小，那么调整可以通过使用圆周定位类型和圆周速率添加指令来完成。

圆周位置形式和圆周速率添加指令，可以和正常程序一样，从教导盒中编辑。

- 转角距离指定

这个功能允许用户指定转角附加指定的转角位置与实际经过点之间的距离。使用转角距离指定来指定这个距离，而不是用 CNT 来指定这个距离。如果指定了 CNT100，那么就采用允许机器人等速驱动的最小转角距离。

提示 指定为 CNT 类型的操作的轨迹和速度的控制是不同的，这个不同取决于是否添加了高精度轨迹功能。位置教导操作的不同不取决于是否添加了高精度轨迹功能。

转角距离指定类型 LP[1] 100 毫米/秒 CDy。

在操作指令的执行过程中，要调整指定的转角位置与路径间的距离，可以通过使用转角距离指定类型来完成。当使用转角距离指定类型时，必须指定转角距离。

转角距离指操作过程中指定的转角位置与路径的距离。

当指定转角距离时，注意以下几点：

- 以毫米为单位指定转角距离。
- 转角距离在 0 毫米到 1000 毫米的范围内变化
- 转角距离越小，机器人越接近于指定的转角位置。
- 转角距离越大时，机器人越远离指定的转角位置。
- 转角决不会大于指定的先前的转角位置和随后的转角位置之间的距离的一半。在这种控制得到许可的操作中，即使增加程序中的转角距离，转角距离还是会受到限制。CNT 定位类型和转角距离定位类型都将受到这种限制的影响。

电弧焊接指令

当机器人需要执行电弧焊接时，电弧焊接指令可以直接使用。

- 电弧焊接开始指令：指示机器人开始电弧焊接。
- 电弧焊接结束指令：指示机器人结束电弧焊接。

要教导电弧焊接动作指令，按 F2 键或者 F4 键，然后选择标准电弧焊接指令

POINT ARCSTRT WELD_PT ARCEND TOUCHUP>

要指定单一的电弧焊接指令，按 F1，INST 键，显示子菜单。然后，从子菜单中选择电弧焊接。

Instruction		JOINT	30 %
1 Registers	5 JMP/LBL		
2 I/O	6 CALL		
3 IF/SELECT	7 Arc		
4 WAIT	8 ---next page---		

SAMPLE1

电弧焊接开始指令

电弧焊接开始指令用来指示机器人开始电弧焊接。支持下面两种电弧焊接开始指令:

- 电弧焊接开始[i]: 指定焊接条件序号。
- 电弧焊接开始[V.A]: 指定焊接条件。

电弧焊接开始[i]

电弧焊接开始[1]指令根据预先确定的焊接条件开始电弧焊接。

电弧焊接结束指令

电弧焊接结束指令用于指示机器人停止电弧焊接。支持下面两种电弧焊接结束指令：

- 电弧焊接结束[i]: 指定焊接条件序号。
- 电弧焊接结束[V,A,s]: 指定焊接条件。

电弧焊接结束[i]

电弧焊接结束[1]指令实行陷口保护,根据预先设定的焊接条件,结束电弧焊接。

焊口保护功能减小焊接完成后的电压和电流,防止由于电压突然下降产生陷孔。要取消焊口保护功能,把 TIME 0 设置为焊接条件。

焊接输入输出指令

焊接输入 (WDI) 和焊接输出 (WO) 信号是可以被用户控制的输入输出信号。

$$\mathbf{R}[\mathbf{i}] = \mathbf{W}\mathbf{D}\mathbf{I}[\mathbf{i}]$$

R[i]=WDI[i]指令在寄存器中存储焊接输入状态（On=1，Off=0）。

堆垛指令

根据储存在堆垛寄存器中的值，堆垛指令从堆垛类型计算当前堆垛点的位置，从路径类型计算当前路径的位置。然后，它将建立的值写入堆垛动作指令的位置数据中。

堆垛指令

PALLETIZING--[pattern]_i

B, BX, E, EX

Palletizing number(1 to 16)

palletizing number (1 to 16) : 堆垛序号 (1 到 16)

根据堆放类型和路径类型，将堆垛分成了4种堆垛类型

关于教导盒，堆垛被分为以下几种堆垛类型。

堆垛类型

类型	允许的堆垛
B	简单的堆放类型和单个的路径类型
BX	简单的堆放类型和多个路径类型
E	复杂的堆放类型和单个的路径类型
EX	复杂的堆放类型和多个路径类型

寄存器指令

寄存器指令在寄存器上完成算术计算。下列寄存器指令是可用的：

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Arc	
4 WAIT	8 ---next page---	
PROGRAM		

- 寄存器指令
- 位置寄存器指令
- 位置寄存器轴指令
- 堆垛寄存器指令

在寄存器操作中，下面列出的多项式运算是可以进行的：

Example 1: R[2]=R[3]-R[4]+R[5]-R[6]
 2: R[10]=R[2]*[100/R[6]]

需要施加以下限制：

- 单独一行可以写下多至 5 个运算符（算子）。

Example 1: R[2]=R[3]+R[4]+R[5]+R[6]+R[7]+R[8]

多至 5 个运算符（算子）

- “+”和“-”运算符可以在单独一行上混合出现。同样“*”和“/”运算符也可以。然而“+”和“-”运算符不可以和“*”和“/”运算符混合。

4.6.1 寄存器指令

寄存器指令在寄存器上完成算术计算。寄存器是一个存储整数和十进制小数的变量。

R[i]=(value)

R[i]=(value)指令把数值载入指定的寄存器中。

输入输出指令

输入输出（输入/输出信号）指令用于改变输出到外围设备的信号输出状态并且读取输入信号的状态。

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Arc	
4 WAIT	8 ---next page---	
PROGRAM		

- （信号）数字输入/输出指令
- 机器人（数字）输入/输出指令
- 模拟输入/输出指令
- 组输入/输出指令

提示 关于输入/输出信号，在使用前需要给物理序号分配合理的序号。

数字输入/输出指令

数字输入指令（SDI）和数字输出指令（SDO）是可以被使用者控制的输入输出信号。

R[i] = SDI[i]

这个指令R[i] = SDI[i]，把数字输入信号(on = 1/off = 0)载入到指定的寄存器中。

指令R[i] = SDI[i]

$$R[i] = SDI[i]$$

Register number (1 to 32)

Digital input signal number

Example 1: R[1] = SDI[1]
 2: R[R[3]] = SDI[R[4]]

register number: 寄存器序号

position register number (1 to 10): 位置寄存器序号 (1 到 10)

digital input signal number: 数字输入信号序号

SDO[i]=ON/OFF

指令 SDO[i]=ON/OFF 将指定的数字输出信号在 on 和 off 之间转换。

指令 SDO[i]=ON/OFF

SDO [i] = (value)

Digital output signal number

ON : Turns on the digital output signal.

OFF: Turns off the digital output signal.

Example **3: SDO[1] = ON**
 4: SDO[R[3]] = OFF

SDO[i] = PULSE, [WIDTH]

SDO[i] = PULSE, [TIME]指令在一段指定的时间内，转换指定的数字输出的当前状态。当没有指定时间段时，用\$DEFPULSE(0.1—second units (秒 单位))来指定执行脉冲输出的时间。

指令 SDO[i] = PULSE, [WIDTH]

SDO [i] = PULSE, (value)

Digital output signal number

Pulse width (sec) (0.1 to 25.5 sec)

Example **5: SDO[1] = PULSE**
 6: SDO[2] = PULSE, 0.2sec
 7: SDO[R[3]] = PULSE, 1.2sec

digital output signal number: 数字输出信号序号

Pulse width (sec) (0.1 to 25.5 sec): 脉宽 (秒) ((0.1 to 25.5秒)

SDO[i] = R[i]

指令SDO[i] = R[i]，根据指定的寄存器的值，打开或者关闭指定的数字输出信号。当指定的寄存器的值为0时，关闭指定的数字输出信号。当指定的寄存器的值为非0时，打开指定的数字输出信号。

. 指令SDO[i] = R[i]

SDO [i] = R [i]

Digital output signal number

Register number (1 to 32)

Example **7: SDO[1] = R[2]**
 8: SDO[R[5]] =R [R[1]]

digital output signal number: 数字输出信号序号

register number (1to32): 寄存器序号 (1 到 32)

JMP LBL [i]

Label (1 to 32767)

Example **3: JMP LBL[2:hand open]**
 4: JMP LBL[R[4]]

程序调用指令 CALL (程序)

程序调用指令 CALL (程序) 将程序控制转移到另一个程序 (子程序) 的第一行，从而执行子程序。当执行调用程序的程序结束指令 (END) 时，控制会迅速地返回到调用程序 (主程序) 中的程序调用指令后的那条指令上。要输入调用程序名，选择自动显示的子菜单或者按 F5,STRINGS 直接输入字符。

条件分支指令

条件分支指令，当某个条件满足时，在程序中的一个位置到另外一个位置处产生分支。
两种条件分支指令是可用的。

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

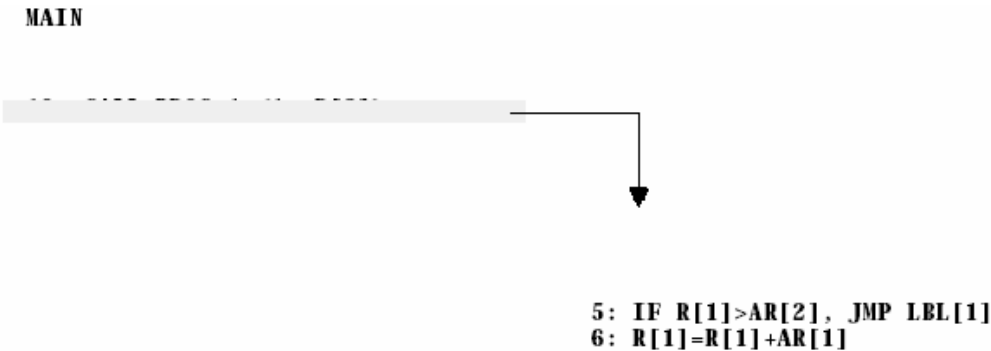
Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

- 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。寄存器条件比较指令和输入输出条件比较指令是可用的。
- 条件选择指令：根据寄存器的值，在指定的跳转指令或者子程序调用指令处产生分支。

变量

通过使用“变量”和“变量寄存器”，也可以在两个程序传递数据。

例如 在这个例子中，主程序MAIN用两个变量调用子程序PROC_1。PROC_1可以利用变量寄存器使用变量的值。第一个变量与AR[1]对应，同时第二个变量与AR[2]对应。



在宏指令中可以以同样的方式使用变量。

变量类型

支持下列变量：

变量类型

变量类型	例子
常数	1,3.5
字符串	'Perch'
变量寄存器	A R[3]
寄存器	R[6]

可以设置变量的指令

指令	例子
程序调用指令	CALL SUBPRG (1, R[3], AR[1])
宏指令	Vacuum hand open (2.5)

提示 使用指令（比如条件分支指令）来实现分支的程序调用不能使用变量。这个问题可以通过如下编程来解决：

(不能设置变量)

IF R[1]=3, CALL PROC_5

(可以设置变量)

! IF R[1]<>3, JMP LBL[1]

CALL PROC_5 (1, R[2])

LBL[1]

可以使用变量寄存器的指令

可以使用变量寄存器的指令

指令	例子
	R[1]=AR+R[2]+AR[4] IF R[1]=AR[1], JMP LBL[1]
	AO[1]=AR[2] GO[1]=AR[2]
	IF AO[1]=AR[1], JMP LBL[1] WAIT GI[1]<>AR[2], TIMEOUT, LBL[1]
	UTOOL_NUM=AR[4]
间接索引指令	R[AR[1]]=R[AR[2]] SDO[AR[1]]=ON
程序调用指令变量	CALL SUBPRG1 (AR[5])
宏指令变量	hand 3 open (AR[1])

变量限制

必须给变量添加以下的限制：

- 可以设置多至10个变量。
- 字符串类型的变量的长度为1到16个字符。（0个字符的变量被认为是没有被初始化。）
- 对于一个已经被间接指定了的索引的元素，不能使用间接指定。

if R[AR[1]]

R[R[AR[1]]]

- 存储在变量寄存器中的值不能够在子程序中改变。

指定变量

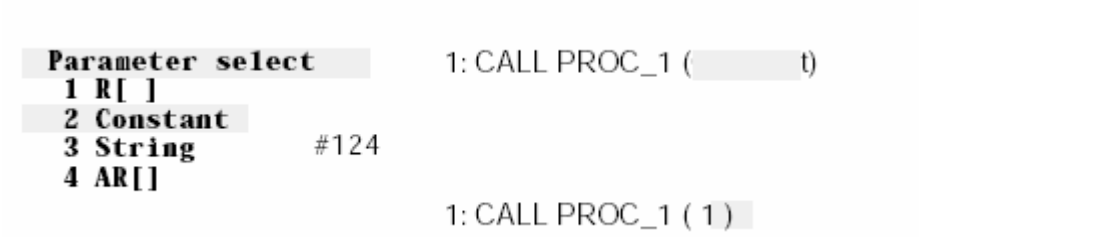
当指定程序调用指令或者宏指令时，指针停止在最后一行上。如果没有变量需要指定，按ENTER回车键或者"! "或者"# "键，来移动指针到下一行上。

要显示变量选择子菜单，按功能键[CHOICE].

			%
CALL PROC_1	1 R[]	5 <None>	#123
	2 Constant	6 <Insert>	
	3 String	7	
	4 AR[]	8	
[CHOICE]			

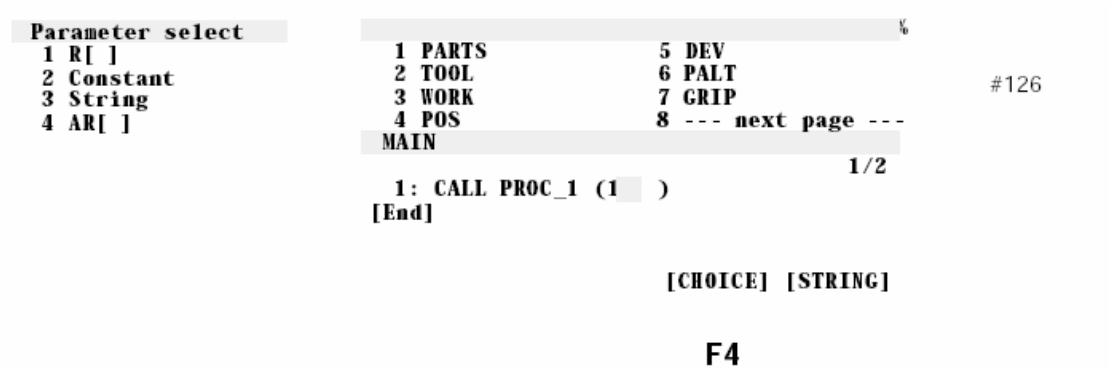
指定常数类型的变量

要指定常数类型的变量，按功能键[CHOICE]，然后在子菜单中选择“1常数”（见“指定变量”）。

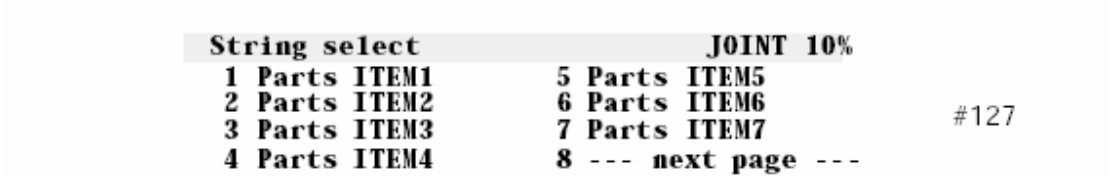


指定字符型的变量

要指定字符型的变量，按功能键[CHOICE]，然后在子菜单中选择“字符串”（见“指定变量”）。字符串类型选择菜单出现。



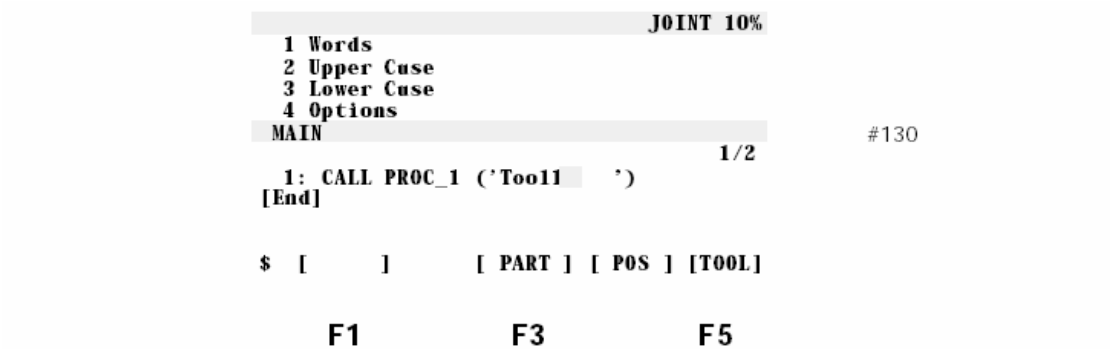
当选择了字符串类型后，出现字符串选择菜单。



在菜单中选择字符串。字符串就确定了。

1: CALL PROC_1 ('Parts ITEM2')，从菜单中选择Parts ITEM2。

要直接输入字符串，按字符串类型选择菜单或者字符串选择菜单上的功能键STRINGS。



按ENTER回车键确定字符串

1: CALL PROC_1 ('Tool 12')

JOINT 10%

CALL PROC_1 (1, 'T00L12') #132

[CHOICE] CHANGE

要改变字符串，移动指针到字符串处，然后按[CHANGE]功能键。字符串类型选择菜单出现。

指定变量寄存器型的变量

要指定变量寄存器型的变量，按功能键[CHOICE]，然后在从子菜单中选择AR[]（见“指定变量”）。

Parameter select

1 R[]

2 Constant

3 String

#133

4 AR[]

1: CALL PROC_1 (AR[...])

Enter an index: 输入索引

1: CALL PROC_1 (AR[1])

要锁定直接或者是间接索引的指定，按功能键[INDIRECT]。显示改变如下：

AR[R[...]]! AR[AR[...]]! AR[R[...]]!

指定寄存器型的变量

要指定寄存器型的变量，按功能键[CHOICE]，然后在从子菜单中选择4R[]。

Parameter select

1 R[]

2 Constant

3 String

#134

4 AR[]

1: CALL PROC_1 (R[...])

Enter an index: 输入索引

1: CALL PROC_1 (R[1])

要锁定直接或者是间接索引的指定，按功能键[INDIRECT]。显示改变如下：

R[R[...]]! R[AR[...]]! R[R[...]]!

添加变量

移动指针到一行结束的“)”处。

1: CALL PROC_1 (1

按功能键[CHOICE]，然后在从子菜单中选择变量类型。一个新的变量可以添加到指针的位置。

1: CALL PROC_1 (1, Constant)

选择一个变量类型然后设置一个值。

1: CALL PROC_1 (1, Constant) 选择一个变量类型

1: CALL PROC_1 (1, 2) 设置一个值2

插入变量

把指针移动到需要插入变量的变量处。



1: CALL PROC_1 (1, 1)

按功能键[CHOICE], 然后在从子菜单中选择<插入>。一个新的变量可以添加到指针的位置。

1: CALL PROC_1 (1, ..., 2)

选择一个变量类型然后设置一个值,索引和其他。

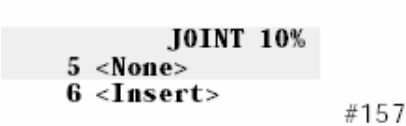
1: CALL PROC_1 (1, R[...], 2) 选择常数类型

1: CALL PROC_1 (1, R[3],) 设置一个值"3"

提示 当没有指定变量, 并且")"在行的结束处时, 不能插入变量。相同的子菜单出现; 选择变量类型。

删除变量

把指针移动到需要删除的变量处。



1: CALL PROC_1 (1, , 3)

按功能键[CHOICE], 然后在从子菜单中选择<None>。变量指针的位置处删除。

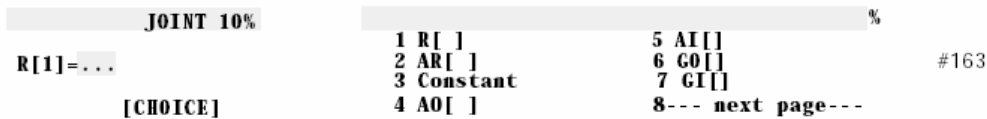
1: CALL PROC_1 (1, 3)

提示 当没有指定变量, 并且")"在行的结束处时, 选择<None>只是简单的关闭子菜单; 不删除任何变量。

指定变量寄存器

下述解说用寄存器指令作为例子。

寄存器指令的右边的选择如下:



要用这个指令选择变量，在菜单上选择AR[]

1: R[1]=AR[...]

指定索引

1: R[1]=AR[1]

#167

DIRECT INDIRECT [CHOICE]

如果功能键F3“间接指定”在索引元素处被按了两次，那么，间接索引指定就可以使用变量寄存器。

1:WAIT R[R ...]]: 当功能键F3按一次时

1:WAIT R[AR ...]]: 当再次按功能键F3时

在变量使用上的提示

当指定变量时，注意以下几点：

- 当指定变量时，并不检验变量的内容。如果变量的类型与子程序中相对应的变量类型不匹配，那么，在执行过程中，会发生错误。

例如 在这个例子中，虽然将AR[1]的值，指配给了子程序中的寄存器PROC_1，但是，在主程序中指定为字符串类型的变量。当子程序中的第5行执行时，就会发生错误。

MAIN

10: CALL PROC_1 ('ABCD')

5: R[1]=AR[1] X

当指定变量时，并不检验变量的序号。即使变量的序号不正确，只要在主程序中指定的变量不在子程序中使用，那么，就不会发生错误。

例如 在这个例子中，主程序中只指定了一个变量，但是，在子程序PROC_1中却使用了两个变量。当PROC_1中的第6行执行时，就会发生错误。

MAIN

10: CALL PROC_1 (1, 2)

30: CALL PROC_1 (R[1])

5: R[1]=AR[1]

6: R[1]=R[1]+AR[2] X

在指定程序调用指令的变量上的提示

- 当程序名改变时，已经设置的变量保持不变。
- 当程序调用指令自身重新指定时，程序名和所有的变量都被删除。

在指定宏指令的变量上的提示

- 当宏名改变时，已经设置的变量保持不变。

在执行上的提示

和在使用变量时的提示一样，当指定变量时，并不检验在调用程序和被调用程序中使用的变量的内容和序号。当变量设置不当或者不正确的使用时，在执行程序过程中，会在检测到冲突的行上发生错误。

- 确保在主程序中指定的变量序号与在子程序中使用的变量的序号一致。
- 如果在主程序中指定的变量不在子程序中使用，那么，就不会发生错误。
- 确保在主程序中指定的变量的内容与在子程序中使用的变量的指令的类型匹配。
- 确保index（索引）和指定的变量的值设置的正确。

1: CALL PROC_1 (Constant)：由于值没有初始化而出现错误

2: CALL PROC_1 (R[...])：索引没有初始化

当执行包含这些的行时，出现”INTP--201 Unspecified Statement(未指定语句)”错误

系统变量的相关变量

附带程序调用指令或宏指令功能的变量显示,作为可选项,当字符串型的变量被选择时,字符串就被设置为系统变量。这些系统变量如下示:

系统变量的相关变量

项	系统变量	注释
单字符串类型	\$STRING_PRM=TRUE/FALSE Standard value=FALSE	(提示)
两个字符的字符串	\$ARG_STRING[i].\$TITLE (i = 1 to 10)	大于一个到多至16个字符 (提示)
三个字符的字符串	\$ARG_STRING[i].\$ITEMJ (i = 1 to 10, j=1 to20)	多至16个字符 (提示)
四字节字符输入	\$ARG_WORD [i] (i = 1 to 5)	多至7个字符 (提示)

提示 字符串型的变量只能在KAREL程序中使用。

其他指令

下列混合指令是可用的:

Instruction		JOINT 30%
1 Miscellaneous	5 MACRO	
2 Program control	6 SENSOR	
3 SKIP	7 Multiple control	
4 Offset	8 ---next page---	
PROGRAM		

- RSR 指令
- 用户警报指令
- 计数器指令
- 倍率指令
- 注释指令
- 信息指令
- 参数指令
- 最大速度指令

RSR 指令

RSR[i]=(value (值))

RSR 指令使得 RSR 指令在可用和不可用之间交替。

RSR 指令

RSR [i] (value)

(1 to 4) ENABLE: Enable RSR function
 DISABLE: Disable RSR function

Example RSR[2:Workproc.2.]=ENABLE

RSR [i] (value): RSR [i] (值)

ENABLE: Enable RSR function: 使RSR功能可用

DISABLE: Disable RSR function: 使 RSR 功能不可用

用户警报指令

UALM[i]

用户警报指令显示警报信息，该警报信息与在警报显示行上设置的用户警报序号相对应。用户警报指令中止正在进行中的程序。用户警报是在用户警报设置界面设置的(见 3.15)而且此设置被存储在系统的变量\$UALM_MSG 中。在受控启动时，可以改变用户警报的总数。

用户警报指令

UALM [i]

Alarm number

Example **1: UALM[1] (\$UALRM_MSG[1] = WORK NOT FOUND**

Alarm number: 警报序号

计数器指令

timer [i] = (state (状态))

计数器指令开始或者结束程序的计数器。程序的计数器的操作状态可以在程序计数器界面 STATUS PRGTIMER (项)上观看。

信息指令

MESSAGE[message statement(信息语句)]

信息指令在用户界面上显示指定的信息。一个信息可以包含多至24个字符，包括字母数字字符，信号(*)，下划线(_)和@符号(@)。要添加注释，按ENTER回车键。

参数指令

多轴控制指令

多轴控制指令控制多任务程序的执行。

只有支持多任务选项时，这些指令才可以被指定和执行。

1 Miscellaneous	5 MACRO
2 Program control	6 SENSOR
3 Skip	7 Multiple control
4 Offset/Frames	8 --- next page ---

- 旗号指令
- 旗号等待指令
- 程序执行指令

旗号指令

旗号指令用指定的序号在 on 和 off 之间开关旗号。旗号是一种用来同步任务的执行的开关。

- 例如，当某一步和程序中其后的步骤的执行等待，直到另外一个同时执行的程序中的某个条件满足时，首先使用旗号等待指令，关闭指定序号的旗号并且使这个程序等待直到打开旗号为止。
- 每个旗号都可以记忆它被打开的次数。当一个等待旗号（等待旗号的“on”值不是 0）的等待语句执行时，执行迅速结束并且“on”的值减小 1。
- 当程序使用等待语句等待旗号（等待旗号的“on”值为 0）时，程序进入等待状态，直到旗号被另外一个程序打开为止。
- 当关闭旗号时，“on”值被清零。
- 旗号可用序号为从 1 到 32。

旗号等待指令

旗号等待指令使得程序等待，直到指定序号的旗号被打开为止。

- 程序处于等待状态直到指定的旗号被另一个程序打开为止
- 可以选择限时（等待时间没有限制）等待和指定时间等待。程序执行指令在程序的执行过程中，程序执行指令开始另外一个程序的执行。
- 和程序调用指令的区别在于：调用程序执行完之后，调用指令后面的行继续执行，然而，使用程序执行指令，起动另外一个程序执行的程序是同时继续执行的。
- 要将同时执行的程序同步，可使用旗号指令和旗号等待指令。
- 如果试图执行一个指定了相同的动作组的程序，就会发出警报。如果发生这样的情况，可以指定一个不同的动作组。

异步操作组指令

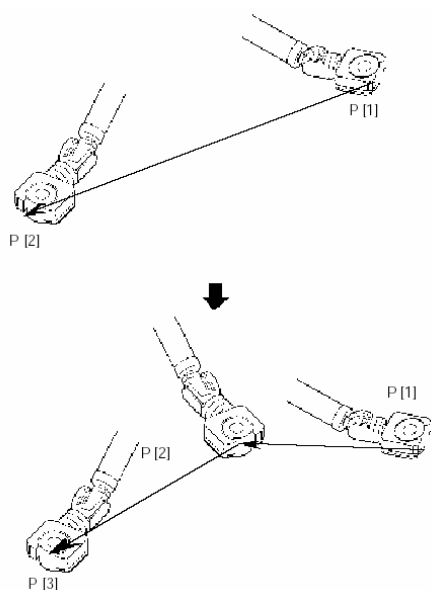
异步操作组指令，通过为每个操作组单独指定操作类型，进给速度和定位类型来实现异步地控制操作组。

固定工具的姿态

当机器人的动作相当得改变工具的姿态的时候，就会浪费一个周期时间。当工具的姿态被平稳而逐渐地被改变的时候，机器人运动速度会相对快很多。通过教导位置的方法，能够让工具姿态的变化尽可能连贯的和机器人的动作相匹配。

当工具的姿态必须做出相当得改变的时候，可以把一个动作划分成几个比较小的动作来教导。也就是说，通过教导使工具的姿态逐渐的改变。

根据工具的姿态来教导位置



尽可能平稳的改变手的姿态：

- 1 教导工作的起始位置，这样机器人就处于正常的姿态。
- 2 手动移动机器人到工作的终止位置。然后检查机器人是否还处于正常姿态。
- 3 向机器人教导终止位置
- 4 为了和工作一致，在起始位置和终止位置之间再教导一个点。
- 5 选择一个笛卡尔坐标系（世界坐标系，用户坐标系或者手动坐标系）并且手动移动机器人到起始位置
- 6 选择笛卡尔坐标系，并且收到移动机器人到终止位置，然后把机器人移动到下一个要被教导的位置。
- 7 纠正教导位置，使机器人处于正常姿态。



警告

当机器人的移动方式被改变成线性移动的时候，如果 J5 轴通过奇异点（接近 0 度），那么这些点必须使用附加的不改变姿态的移动指令，或者移动类型必须从线性移动改变为轴向移动。

- 8 重复步骤 6 和 7，把在起始点和终止点之间剩下的需要被教导的点都教导给机器人。

为了使工具的姿态改变平稳:

- 1 手动移动工具到焊接起始点，然后，教导位置和需要的姿态。
- 2 手动移动工具到焊接终止点，然后，再一次教导位置和需要的姿态。
- 3 给在起始点和终止点之间，给工具要通过的每一个点都指定动作指令。
- 4 使用手动进给，把工具移动到程序最先设定的位置点上。
- 5 把光标定位到“运动到程序最后指定点”的指令处，然后，低速激活机器人。
- 6 工具会移动到程序最后指定的位置点，在过程中改变它的姿态。在步骤 3 中指定的要经过的点附近，把工具停止在一个需要的位置。
- 7 调整经过的位置点的动作指令，调整到步骤 6 中工具停止的点的位置。
- 8 对焊接起始点和终止点之间的所有经过的点，都重复步骤 6 和 7。

预定义位置

预定义位置就是在程序里多次被用到的位置。使用的预定义位置通常是 **Pounce Position**（抓取位置）或者起始（标杆）位置。应该在程序里有效的定义这些位置，或者定期的删除这些位置。

Pounce position 抓取位置

抓取位置是所有工作都参考的位置。它是在机器工具和外围设备的动作区域外的安全位置。

起始（标杆）位置

起始位置或者标杆位置是在机器工具和工件传递区域外的安全位置。当机器人处在这个位置的时候，参考位置的数字输出信号会被打开。

一个焊接周期应该这样来编程：工具首先应该运动到参考位置，然后再运动到焊接起始点。参考点应该始终远离正在操作的外围设备。与此同时，参考点应该尽可能的接近焊接起始点，从而减少焊接周期时间。应该把和起始焊接点同样的姿态教导给参考点。

提示：HOME 是外围输入输出设备的输入信号，并且不表示起始位置。参考位置是起始位置之一，HOME 并没有把机器人移动到参考位置的功能。

其他预定义位置

抓取位置，参考位置，或者任何其他位置都可以被定义为预定义位置。应该指定在程序中经常被用到的位置作为预定义位置。

当使用固定位置的时候，使用位置寄存器。

电弧焊接

当编写电弧焊接指令的时候，应该特别注意以下问题：

- 为了将工具移动到焊接起始点，可以使用关节动作和 **FINE** 定位指令。
- 为了将工具移动到经过的焊接点，可以使用线性动作和 **CNT** 定位指令。
- 指定吹焰器相对于要焊接的工件的正确方位
- 使用合适的焊接条件。详细信息参见电弧焊接的详细描述。



注意

如果位置变量和位置寄存器根据关节类型被教导，那么当改变用户坐标系的时候，他们不受影响。如果位置变量是根据笛卡尔坐标系被教导的，并且用户坐标系输入选项没有被使用，那么位置变量不受用户坐标系的影响。在其他情况下，当改变用户坐标系的时候，位置变量和位置寄存器都受影响。

提示：当位置寄存器被两个程序共用的时候，为了让机器人移动到同一个空间位置，这两个程序必须拥有相同的工具和用户坐标系统。

当你在启动机器人系统的时候，如果已经设置好了热启动，那么你就可以选择热启动。热启动的功能就是能够存储电源关闭前的系统设置，并且在下次打开电源的时候唤醒系统。在冷启动模式下，控制器的系统软件在启动过程中初始化。当你改变了系统的设置，比如输入输出设置，你必须通过冷启动来开机。

自动启动的程序

用户可以指定一个自动启动的程序。当电源打开的时候，程序就会自动启动。如果在要被启动的程序里指定了倍率和参数指令，那么系统在打开电源的时候，用户就可以定制自己的系统。

系统状态

下面的表格列出了不同的启动模式
在不同启动模式下的系统状态

	热启动	
	有效	屏蔽（默认设置）
寄存器容量	f	f
倍率	f	⌘[10%]
选择程序	f	N(提示 3)
执行行	f	⌘[第一行]
输入输出状态	f(提示 1)	⌘[全部关闭]
TP 界面	n（提示 2）	⌘[提示界面]

F: 电源关闭时的所有的值都被保存下来，并且在电源打开的时候被恢复。

N: 只有电源关闭时的某些值被保存下来。

⌘：电源关闭时的所有的值都没有被保存，当系统启动的时候，都设置成默认值。


提示 1: 通常情况下，电源关闭时的状态是会被恢复的。但是在系统关闭时被脉冲指令执行的数字输出(SDO)是被关闭的。如果要恢复输入输出口的状态，可以在[6 SYSTEM Config 系统配置]中设定需要的恢复状态。

即使是在电源中断处理启动的时候，任何输出信号都不会被恢复。但是在下列情况下，所有的输出信号都会被关闭：

- 在电源关闭之前，改变了输入输出分配。
- 输入输出设备的保险丝熔断了，或者输入输出设备的电源被关闭。
- 输入输出设备的设备被改变了。

提示 2: 在电源关闭时的选中界面类型会被存储下来，但是页面，光标和界面的其他状态没有被存储下来。界面使用和冷启动结束后相同的页面，光标和其他界面状态。

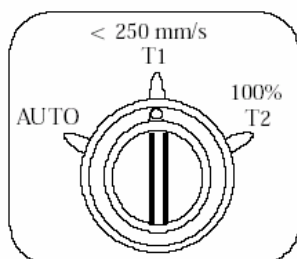
提示 3: 调用子程序的主程序的名字被存储下来。

**注意**
在打开系统电源之前，应该检查上述各个启动模式对应的系统状态。

三模式开关

三模式开关是安装在操作面板或者操作盒上的一个钥匙操作开关。这个开关的作用是，根据机器人的操作条件和使用状况选择一个最合适的机器人操作模式。总共有三种操作模式可以选择：AUTO（自动），T1 和 T2。

· 三模式开关



AUTO 自动 mm/s 毫米/秒

当使用三模式开关进行操作模式之间的切换的时候，在教导盒的屏幕上会出现一条信息，同时机器人动作暂停。当把钥匙从开关里拿走之后，开关的设定位置就会被固定。根据 CE 和 RIA 规程，当开关处于 T2 模式时，钥匙不能被拿走，设定位置也不能被固定。）



注意

根据 RIA 规程，如果在锁住 deadman 开关的情况下，开关在 T1 或者 T2 模式和 AUTO 模式之间切换，会产生一个系统错误。在这种情况下，直到 deadman 开关被松开之后，才能选定模式。松开 deadman 开关，然后再把 deadman 开关锁住。

连接：连接 FENCE 信号到保护导流栅上。制作这样一个连接：当保护导流栅打开的时候，信号不能输入给机器人；当它关闭时，信号能够输入给机器人。

*SFSPD 信号能够满足用户系统的要求。

下文介绍通过选择三模式开关选定的操作模式：

T1(<250mm/s)：测试模式 1

这个模式的作用是把操作位置教导给机器人。这个模式还可以用来低速检查机器人的移动路径和程序顺序。

程序执行：程序只能从教导盒中被执行。

机器人手动进给速度

- 工具顶端和边缘的速度都被限制到不能超过 250mm/s。

执行程序时的机器人速度

- 倍率值最高能提高到 100%，但是工具顶端和边缘的速度被限制到不能超过 250mm/s。举例来说，如果教导速度是 300mm/s，那么工具顶端和边缘的速度会被限制到 250mm/s。如果教导速度是 200mm/s，那么速度就不会受到限制。即使在教导速度是 250mm/s 或者更低的时候，边缘面上某些部分的速度（比如角上的速度）还是会超过 250mm/s，那么工具上这些地方的姿态会发生改变。在这种情况下，实际操作速度会受到限制。

只有当操作速度受到限制并且教导速度小于等于 250mm/s 时，警告信息 MOTN-231 T1 速度限制 (G:i) 才会出现。

- 在教导速度倍率值超过 100%的基础上，速度限制才会被执行。所以如果教导速度是 2000mm/s，那么操作速度就会被限制到 250mm/s。但是，还能进一步的减小操作速度，比如把倍率值降低到 50%的时候，操作速度就会减小到 125mm/s。

- 只有当教导盒打开而且 **deadman** 开关被按下（夹住）的时候，才能操作机器人。
- 关闭教导盒能够使机器人进入紧急停机警告状况，这样机器人就不会运作了。
- 当教导盒打开，但是 **deadman** 开关没有被锁住，那么机器人处于紧急停机警告状况，这样机器人也不会运作。



注意

在检查你创建的程序时，一定要按照安全手册来执行。

固定操作模式：当开关设定到 **T1** 模式位置的时候，可以通过拿掉钥匙来固定操作模式到 **T1** 模式。

发现并修理故障

- 当开关设定在 **T1** 模式位置，关闭教导盒能使开关把机器人停止并且显示错误信息。打开教导盒开关，然后按下 **RESET** 按钮就可以排除错误。

T2 (100%) : 测试模式 2

T2 模式的作用是对用户创建的程序进行最后的检查。在 **T1** 模式下，因为操作速度是受到限制的，所以不可能检查机器人实际的工具路径和周期时间。在 **T2** 模式下，因为基本上没有速度限制，所以有可能以生产速度来运行机器人，从而达到检查的目的。

即使在 **T2** 模式下，通过降低倍率值，在 ***SFSPD** 信号的基础上使用安全速度能够限制机器人的操作速度。

程序执行：程序只能从教导盒中执行。

机器人手动进给速度

- 工具顶端和边缘的速度被限制到不能超过 250mm/s

执行程序时的机器人速度

- 倍率值可以最高到 100%，没有特别的速度限制。

保护导流栅：如果用户希望保护导流栅始终保持打开，那么需要在开始操作机器人之前设置三模式开关到 **T1** 或者 **T2**。

- 只有当教导盒打开而且 **deadman** 开关被夹住的时候，才能操作机器人。
- 关闭教导盒能够使机器人进入紧急停机警告状况，这样机器人就不会运作了。
- 当教导盒打开，但是 **deadman** 开关没有被按下，那么机器人处于紧急停机警告状况，这样机器人也不会运作。



注意

在检查你创建的程序时，一定要按照安全手册来执行。

固定操作模式：当开关设定到 **T2** 模式位置的时候，可以通过拿掉钥匙来固定操作模式到 **T2** 模式。（但是根据 **CE** 和 **RIA** 规程，不能拿走钥匙）

发现并修理故障

- 当开关设定在 **T2** 模式位置，关闭教导盒能使开关把机器人停止并且显示错误信息。打开教导盒开关，然后按下 **RESET** 按钮就可以排除错误。

AUTO: 自动模式

自动模式在生产的时候使用。

程序执行：程序执行的时候没有任何约束。程序可以从外围设备，操作面板和教导盒执行。但是，只有当使用 RIA 规程的时候，如果开关被设定在 AUTO 模式的位置，程序不能从教导盒执行。

奇异点检测功能

当机器人位于奇异点附近时，如果教导动作语句或者在直角坐标位置数据的基础上编辑位置，那么，当动作语句执行时，机器人的驱动姿态和教导的姿态不同。

要防止这种故障，当教导位置时，奇异点检测功能可以检测教导位置是否是奇异点。然后，这个功能可以根据基于用户选择的轴的类型来教导位置。

功能

要使这个功能可用，把系统变量\$MNSING_CHK 设置为 TRUE

当机器人位于奇点附近时，如果用 SHIFT 和 POINT 键教导动作语句或者用 SHIFT 和 TOUCH UP 键编辑位置，这个功能可以检测教导位置是否是奇异点。当满足如下的条件时，进行此检测：

- 附加指令不包括增量指令，位置补偿指令和工具补偿指令。
- 位置数据的 UF（用户坐标系序号）为 0
- 寄存器的位置类型为直角式类型。

如果检测发现教导的位置是奇异点，那么，教导盒顶部的两行上显示如下警告信息：

TPIF--060 Can't record on cartesian (G:1): TPIF—060不能在笛卡尔(G:1)上记录。

MOTN--023 In singularity: MOTN—023奇异

i: Move group number at a singular point: 动作组序号在奇点

同时，教导盒底部显示如下提示信息：

在关节坐标下记录当前位置

同时，显示功能键 YES 和 NO。选择两个功能键的一个。

- YES: 根据轴的类型删除位置数据。
- NO: 不执行位置的教导/修改。

具有多动作组的程序的位置数据，按组序号上升的顺序进行奇异点检测。

如果多动作组在奇异点，那么，每个组都会出现警告信息和提示信息。

提示

这个功能不能应用于典型的堆垛装载点和经过点的教导。

程序暂停和恢复

程序暂停的作用是让一个正在运行的程序停下来。

程序暂停由以下几种情况产生：

- 当程序正在运行的时候，突然产生一个警报。
- 由操作者特意发出一个指令来停止正在运行的程序。

正在工作的机器人可能用以下的方式之一来停止工作：

- 快停：机器人快速减速直到停止。
- 慢停：机器人慢速减速直到停止。

程序暂停可以分为两种方式：

- 强迫终止(end)：现实程序执行的终止状态。在教导盒的屏幕上显示“ABORTED”。
如果当子程序正在被执行的时候主程序被终止了，那么对主程序的控制返回值信息就会丢失。

SAMPLE1	LINE 7	ABORTED
----------------	---------------	----------------

- 暂停(暂时的停止)：程序的被暂时地停止。教导盒的屏幕上显示“PAUSED”。
被暂时停止的程序可以恢复执行。程序调用指令调用出的子程序返回去控制主程序。

SAMPLE1	LINE 7	PAUSED
SAMPLE1		JOINT 30%

为了从同一个程序的另外一行或者另外一个程序开始，可以通过退出程序来释放 paused 状态。

有两种方法来有意地暂停程序：

- 按下教导盒或者机器操作面板上的紧急停机按钮，或者关闭伺服电源的开关。
外围设备输入输出口的输入信号 IMSTP
- 按下教导盒上的 HOLD 按钮，或者使用外围设备输入输出口的输入信号 HOLD，这些输入能够暂停程序的执行。
- 从混合菜单中选择 1 ABORT(ALL)或者使用外围设备的输入输出口的输入信号 CSTOPI

这个方法异常终止程序。

使用紧急停机按钮暂停和恢复

按下在机器操作面板(盒)或者教导盒上的紧急停机按钮，可以让机器人立即停止下来。此时会发出一个紧急停止警报。

按下紧急停机按钮会产生以下结果：

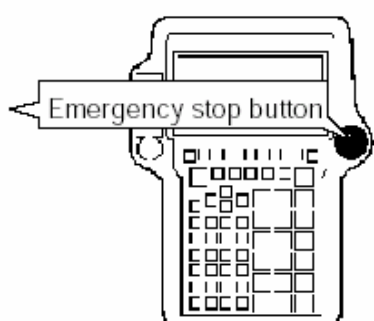
- 机器人立即停止操作，并且程序暂停。
- 发出一个警报并且关断伺服系统的电源。

紧急停机与恢复

紧急停机流程

步骤 1: 按下在教导盒或者机器操作面板上的紧急停机按钮。这个操作将暂停正在运行的程序，教导盒屏幕上显示 **PAUSED**。

紧急停机按钮将被锁住以保持被按下的状态。教导盒屏幕上显示紧急停机警报消息，并且 **FAULT** 灯亮起。



SRV0-002 Teach Pendant E-Stop
SAMPLE1 **LINE 2** **ABORTED**
SAMPLE1 **JOINT 30%**

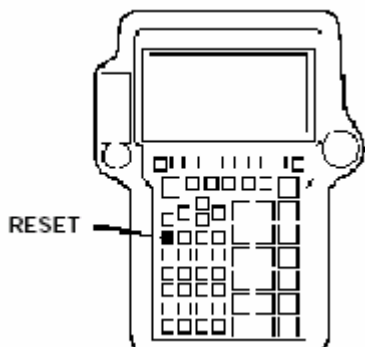
emergency stop button 紧急停机按钮

恢复流程

- 2 消除紧急停止的原因，比如改正程序。
- 3 顺时针旋转紧急停机按钮，解开被锁住的按钮。



- 4 按下在教导盒或者操作面板上的 **RESET** 按钮。这样，教导盒屏幕上的警报信息就会消失，而且 **FAULT** 灯会灭掉。



reset 恢复

使用 HOLD 按钮进行暂停和恢复

按下教导盒或者操作面板上的 HOLD 按钮能够让机器人慢慢的减速直到停止下来。

按下 HOLD 按钮，会引起以下效果：

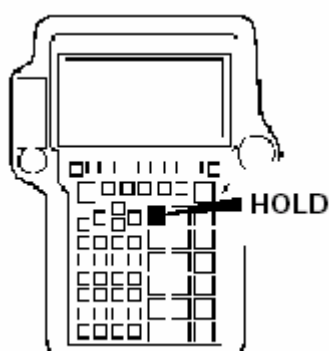
- 机器人会慢慢的减速，直到完全停止下来。
- 能够设定一个装置来引发一个警告关闭伺服电源。在普通项目设置界面中选择 **SETUP General** 来设定这个装置。

保持与恢复

保持规程

步骤 1 按下在教导盒上的 HOLD 按钮。这样，正在运行的程序将被停止，并且在教导盒上显示 **PAUSED**（暂停）。

只有在暂停警告开启的时候，警告信息才会被显示出来。



HOLD 保持

恢复规程

2 重新启动程序就能离开暂停状态。

由警告引起的暂停

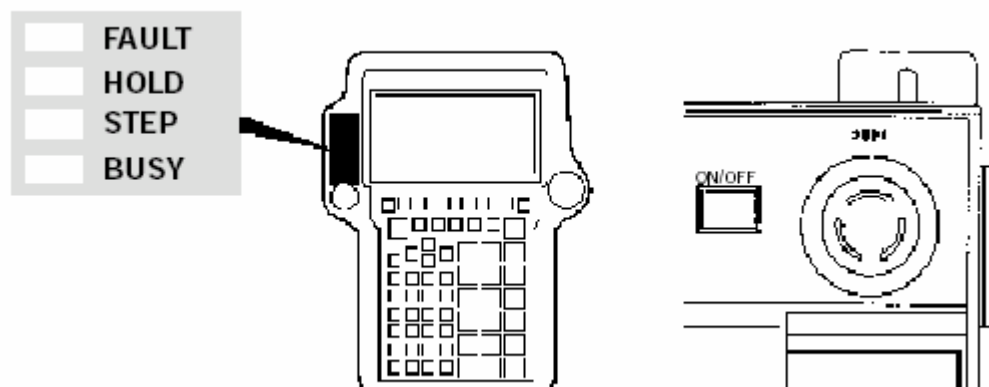
教导或者重放一个程序的时候，如果系统发现了一个错误，或者系统接收到一个紧急停机信号或者其他从外围设备输入的警告信号，系统就会发出一个警告信息。当警告系统发出警告信息之后，该信息就会被显示在教导盒上。与此同时，在确定操作安全之前，一些例如机器人操作和程序执行之类的处理都将被停止。

警告显示

操作者可以通过观察教导盒上的 **FAULT**（错误）灯是否亮起，或者操作面板屏幕上的第一第二两行来检查警告是否已经发生。

通过警告代码，可以区分不同的警告类型。警告的起因和矫正方法都可以通过查询警告代码来得知。

警告的显示



Alarm code
INTP--224
ID No.
Alarm detail code
MEMO-027

INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Specified line does not exist
Alarm JOINT 30 %
1/7
1 INTP-224 (SAMPLE1, 7) Jump label is
2 SRV0-002 Teach pendant E-stop

警告历史

选择警告历史界面（4 ALARM）可以显示警告历史。

3 MANUAL FCTNS
4 ALARM
5 I/O

MENUS

INTP-224 (SAMPLE1, 7) Jump label is fail
INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Specified line does not exist
30-MAY-98 07:15
STOP.L 00000110
Alarm history 1/7
1 INTP-224 (SAMPLE1, 7) Jump label is
2 SRV0-002 Teach pendant E-stop
3 R E S E T
4 SRV0-027 Robot not mastered(Group:1)
5 SYST-026 System normal power up
[TYPE] CLEAR HELP

警告有详细的相关信息。在警告历史界面（4 ALARM）,按下 F5, HELP 就能显示警告的详细信息。

CLEAR HELP

INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Specified line does not exist
30-MAY-98 07:15
STOP.L 00000110
Alarm History

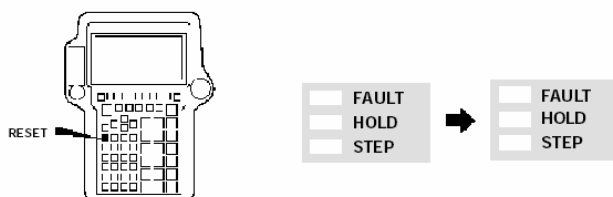
1 Alarm code
2 Alarm detail code
3 Generation date/time
4 Alarm severity

- 警告代码：确定一个警告。
- 警告详细代码：确定一个警告的详细内容。
- 产生日期/时间：显示警告的产生日期和时间。（目前还不支持本功能。）
- 警告程度：显示警告的严重程度。

解除警告

在排除了警告的起因之后，按下 RESET 按钮来重置警告。这样教导盒上第一第二行显示的警告信息就会消失。如果伺服电源已经被关闭的，那么它会被打开。通常重置警告能够是机器人运作起来。

RESET 按钮



暂停警告

当操作者按下 **HOLD** 按钮来暂停机器人的时候，暂停警告的功能就是发出警告并且关闭伺服系统的电源。

在普通项目设置界面的（6 SETUP General）上指定错误警告功能。

警告程度

警告程度显示警告的严重程度和警告的起因。是否停止程序的执行和机器人的操作，是否关闭伺服系统的电源取决于警告的严重程度。

警告程度

	程序	机器人操作	伺服电源	范围
NONE(无)	无	无		-----
WARN（警告）				-----
PAUSED.L PAUSED.G （暂停）	暂停	慢慢的使机器人减速，直到停止	无	局部 全局
STOP.L STOP.G （停止）				局部 全局
SERVO （伺服）		立即停止机器人	关闭	全局
ABORT.L ABORT.G （异常终止）	异常终止	慢慢的使机器人减速，直到停止	无	局部 全局
SERVO2 （伺服）		立即停止机器人	关闭	全局
SYSTEM （系统）				全局

执行程序

执行一个程序就是去播放一个教导过的程序。播放一个教导过的程序就好像播放一盘录像带一样。

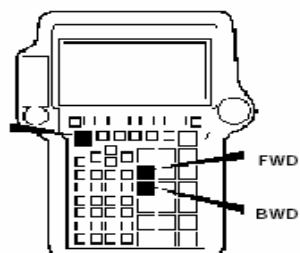
启动一个程序

一个程序可以以下面方式启动：

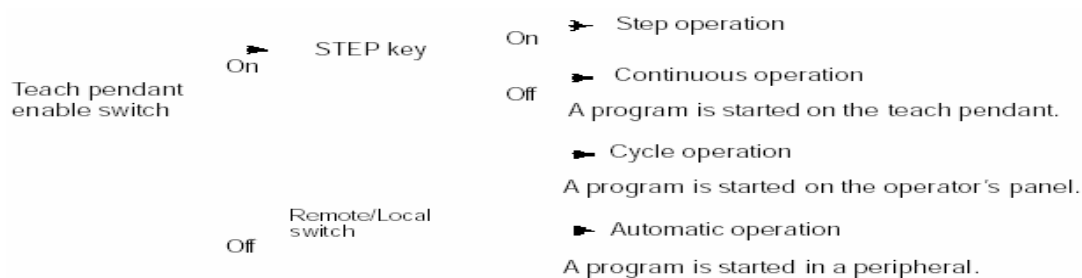
- 使用教导盒（SHIFT 键和 FWD 或者 BWD 键）
- 设置在操作面板上的 **START**(启动)按钮：仅当面板是可选项。

- 使用外部设备（RSR 从 1 到 4 的输入、PROD_START 输入和 START 输入）启动程序

教导盒



出于安全考虑，仅当设备有动作控制时，程序才可以被启动。操作面板上教导盒使能开关和远程本地模式开关，可以用来切换动作控制。如何正确设置来启动程序。



Teach pendant enable switch	教导盒使能开关
STEP key	单步键
Remote/Local switch	远程/当地 开关
Step operation	单步操作
Continuous operation A program is started on the teach pendant	连续操作 在教导盒上启动一个程序。
Cycle operation A program is started on the operator's panel.	循环操作 在操作面板上启动一个程序
Automatic operation A program is started in a peripheral	自动操作 从外部设备启动一个程序



注意

当使用操作面板的教导盒使能开关或者远程/本地模式开关来打开 **start right**(正确启动)时，当前运行的所有程序都暂时停止。

机器人动作

只有当程序中的动作指令发命令时，机器人才移动。参见 4.3 节“动作指令”。
下列因素决定机器人的动作：

- 进给倍率：机器人动作速度（操作速度）
- 笛卡儿坐标系：机器人移动的工作区域

进给倍率

进给倍率决定操作速度。进给倍率是程序中指定进给速度（编程的速度）的百分比。
当前进给倍率显示在教导盒界面的右上角。
按进给倍率键时，界面的右上角会显示一个反白显示的弹出窗口，以引起操作者的注意。
在按下别的键后，反白显示的弹出窗口会自动消失几秒钟。

100%的进给倍率会使机器人以当前设置中指定的最大速度进行操作。

要改变进给倍率，按进给倍率键。无论何时当按住 **SHIFT** 键后，按下负倍率键，进给速度依次减少：**VFINE,FINE,5%,50%,100%**。注意，仅当在手动进给期间，**FINE** 和 **VFINE** 是有效的。当指定了 **FINE** 和 **VFINE** 后，机器人以 1% 的进给倍率移动。

进给倍率必须根据加工室的条件、机器人动作类型和操作者的技能来决定。因此，一个不熟练的机器人操作者应该使用低的进给倍率。

操作速度

操作速度是机器人在程序播放时的移动速度。操作速度从下列的表达式获得：

操作速度

$$\begin{aligned} \text{Operating speed (joint control motion) (deg/sec)} &= \frac{\text{Programmed feedrate}}{100} \times \frac{\text{Coefficient of a joint feedrate}}{2000} \times \text{Maximum joint feedrate} \\ \text{Operating speed (motion under path control) (mm/sec)} &= \frac{\text{Programmed feedrate}}{100} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100} \\ \text{Operating speed (motion under attitude control) (deg/sec)} &= \frac{\text{Programmed feedrate}}{100} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100} \\ \text{Programmed override} &= \$\text{MCR_GRP}:\$ \text{PRGOVERRIDE} (\%) \\ \text{Coefficient of a joint feedrate} &= \$\text{PARAM_GROUP}:\$ \text{SPEEDLIMJNT} \end{aligned}$$

Operating speed (joint control motion) (deg/sec)	操作速度（关节控制动作） 度/秒
Programmed feedrate	编程的进给速率
Coefficient of a joint feedrate	一个关节进给速率的系数
Maximum joint feedrate	最大关节进给速率
Programmed override	编程的倍率
Feedrate override	进给倍率
Operating speed (motion under path control) (mm/sec)	操作速度（在路径控制下的动作） 毫米/秒
Operating speed (motion under attitude control) (deg/sec)	操作速度（在姿态控制下的动作） 度/秒

检查笛卡儿坐标系

当按照笛卡儿坐标系播放位置数据时，系统会检查所采用的笛卡儿坐标系的坐标系号。
当指定 0 到 9 的坐标系号中的一个时，如果指定的坐标系号与当前选择的坐标系号不一致，那么程序就不能够执行。

当教导位置时，坐标系号被指定给位置数据。使用工具改变功能/坐标系改变功能[可选]可以改变一个已经写入的坐标系号。

工具坐标系号（UT）

指定一个mechanical interface（机械接口系）坐标系或者工具坐标系的序号为一个工具坐标系号(UT)。这个号决定了工具坐标系。

- 0 : 应用 mechanical interface （机械接口系）坐标系时。
- 1 到 9: 应用具有指定工具坐标系的工具坐标系时。
- F : 应用具有当前被选择的坐标系号的坐标系时。

用户坐标系号（UF）

指定全局坐标系或者用户坐标系的序号作为一个用户坐标系号（UF）。这个序号决定工作区域的坐标系。

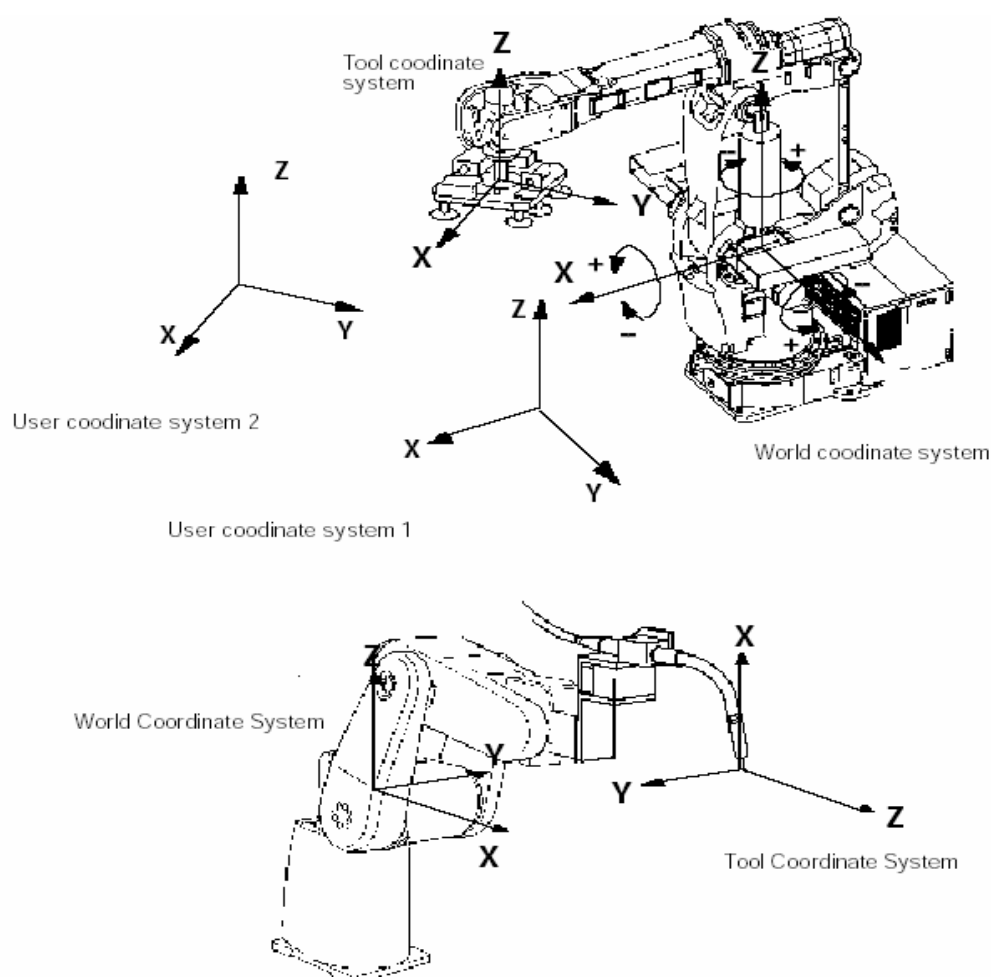
- 0 : 应用全局坐标系时。
- 1 到 9: 应用具有指定用户坐标系的工具坐标系时。
- F : 具有当前被选择的工具坐标系号的坐标系被应用。

位置数据信息

按 F5,[DETAIL](详细)键显示位置数据信息。

Position Detail				JOINT 30%	
P[1]	UF:0	UT:1	CONF:FT,0		
X:	1500.374	mm	W:	40.000	deg
Y:	-242.992	mm	P:	10.000	deg
Z:	956.895	mm	R:	20.000	deg
EDCMD					

选择工具坐标系和用户坐标系



Tool coordinate system	工具坐标系
World coordinate system	全局坐标系
User coordinate system	用户坐标系

恢复执行一个程序

恢复执行程序意味着重新启动一个被暂停的程序。一个程序被暂停前，系统记录了该程序。作为结果，下列情况可能发生：

- 通过程序调用指令的调用，控制转回到主程序。
- 圆周动作的路径可能被再现。

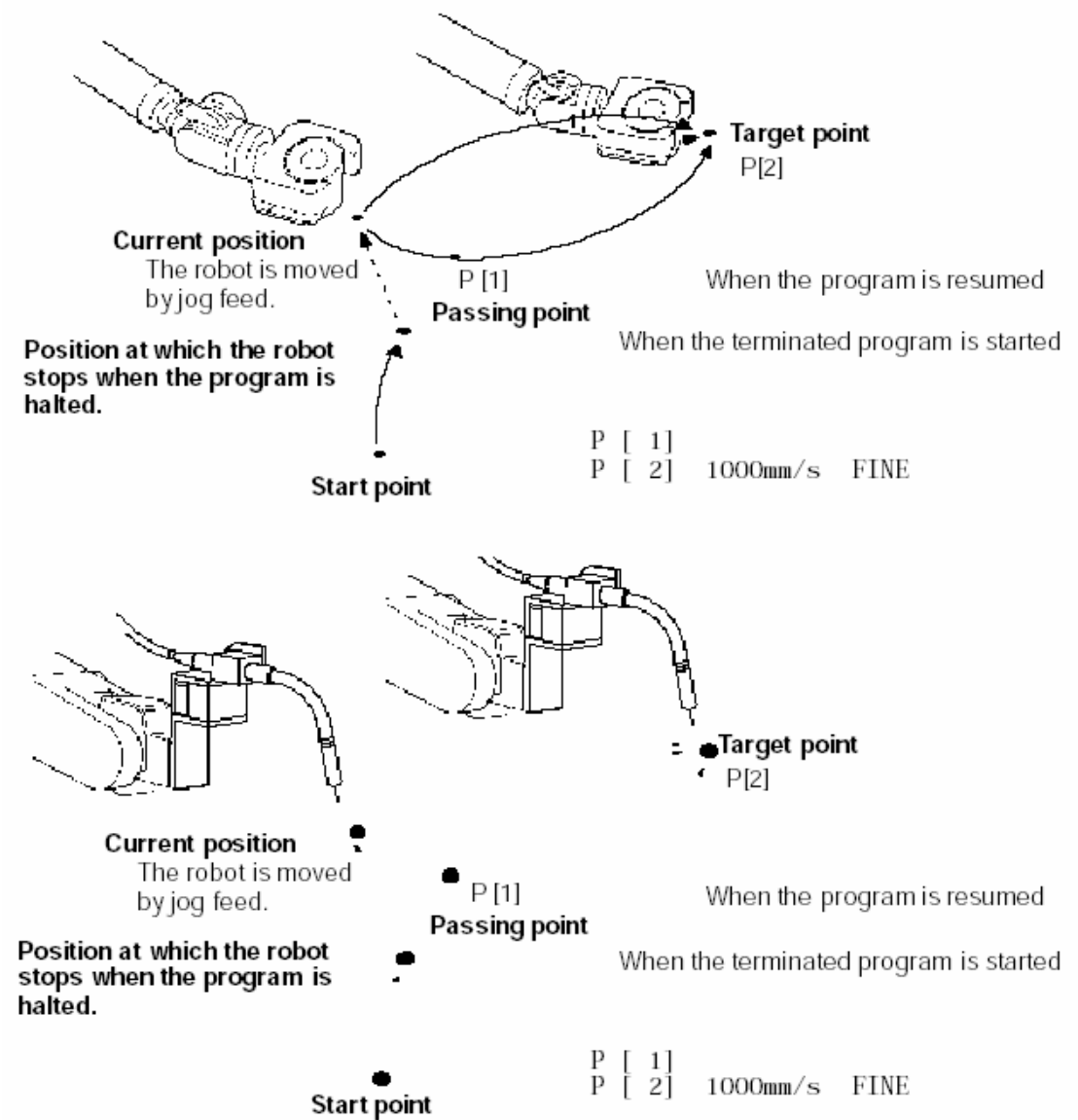
圆周动作的路径

圆周动作中，机器人沿着经过交叉点的路径从当前位置移动到目标位置。程序暂停中断机器人的动作后，机器人通过手动进给来移动，然后程序恢复执行。在这种情况下，机器人会沿着与程序中断前指定的相似路径来移动。

（一个弧度的轨迹会按照一种假设重新计算，该假设假定交叉点是手动进给后的当前位置，开始位置是中断前使用的位置。）

一个圆周动作结束时暂停的一个单步测试，在手动进给后恢复执行时，工具通过一个线性动作返回圆周动作的结束点。动作以圆周动作指令中指定的移动速度来执行。

圆周动作的路径



Current position The robot is moved by jog feed.	当前位置 机器人通过手动进给移动
Position at which the robot stops when the program is halted.	当程序暂停时，机器人的停止位置
Start point Passing point Target point	起始点 经过点 目标点
When the program is resumed When the terminated program is started	当程序恢复执行时。 当启动终止的程序时。

解除暂停状态

下列情况时，程序的暂停状态被解除：

- 从多项菜单中选择了 1 PROGRAM ABORT(程序中止)一项。
- 当教导盒有效时，产生了新的程序。
- 当教导盒有效时，选择了其他的程序。

在暂停状态移动光标

当在暂停的程序中将光标移动到相应的行时，程序恢复执行，系统会询问操作者是否从光标移动到的行来程序启动。当响应这个信息而选择的 YES(是的)时，程序暂停在光标移动到的行。当选择 NO(否)时，光标返回到它移动前的行（初始行），然后程序被停止在这一行。无论选择 YES(是)和 NO(否)，当程序恢复执行时，程序都会从光标移动到的那一行开始执行。

解除停止状态

条件 H 程序必须是暂停的。 屏幕上显示 PAUSED(暂停)。)

LINE 2 PAUSED

- 步骤** 1 按 FCTN 键来显示多项菜单。
- 2 选择 1 PROGRAM ABORT(程序中止)。程序被终止。 ABORTED(中止)显示在屏幕上。)

1 ABORT (ALL)	LINE 2	ABORTED
2		JOINT 30%

FCTN

在暂停状态移动光标

条件 **H** 程序必须是暂停的。 屏幕上显示 PAUSED(暂停)。）

LINE 2

PAUSED

- 步骤 1 移动光标到将被恢复执行的程序行。
- 2 重新启动程序。
- 系统会询问操作者，程序是否会在光标移动到的行恢复执行。

3: L P[3] 1(
4: L P[4] 5(
5: J P[1] 1(
[End]

The cursor is on a different
line from where the program
PAUSED [2].
Are you sure you want to run
from this line ?
YES NO

- 3 选择 YES(是)从光标移动到的地方恢复执行程序。于是这一行就被指定为当前行。

NO

SAMPLE1

LINE 4

PAUSED

SAMPLE1

JOINT 30%
4/6

1: J P[1] 100% FINE
2: J P[2] 70% CNT50
3: L P[3] 1000cm/min CNT30
4: L P[4] 500mm/sec FINE
5: J P[1] 100% FINE
[End]

当选择 NO(否)时，程序从光标被移动前的行（初始行）恢复执行。然后光标返回初始行。

YES

重新启动位置检查功能

当一个程序在 AUTO(自动)模式重新启动时，这项功能会将机器人的当前位置与程序被暂停时机器人所在的位置进行比较。如果比较结果显示位置差异超出了系统允许误差范围，此功能发出一个警报并停止启动程序。

如果有一个警报发生，那么可以从下列可选项来选择重新启动的方法。通过教导盒来做一个选择。

- (1) 用非特殊的动作来启动程序。
- (2) 改变模式并返回到机器人停止位置，然后重新启动机器人。

当重新启动一个程序时，在重新启动位置检查设置菜单的界面。设置当前机器人位置和机器人被暂停位置间允许的误差距离。

SETUP RESUME TOL

6/6

1 Group : **1**
2 Enable Tolerance checking : **YES**
3 Distance Tolerance (mm) [250.0]
4 Orientation Tolerance (deg) [20.0]

Axes Tolerance

5 Rotary Axes (deg) [20.0]
6 Translational Axes (mm) [250.0]

[TYPE]

1. 组

对于每一组，你可以使重新启动位置检查功能有效或者无效，可以设置允许误差。为设置一个目标组序号。当有效的重新启动位置检查功能超过一组时，如果一个组中，有一个允许误差被超过，一个警报就会产生。

2. 使能/关闭允许误差检查

要使重新启动位置检查功能有效，选择 YES(是)。缺省的设置是 YES(是)。

3. 允许距离误差 (mm(毫米))

在程序重启时，如果当前机器人位置和机器人暂停位置间的距离差，超过该处的设置值时，就产生一个警报，并且程序不会被重新启动。

4. 允许空间方位角误差 (deg(度))

在程序重启时，如果当前机器人位置和机器人停止位置间的关节角的差值，超过该处的设置值时，就产生一个警报，并且程序不会被重新启动。

5. 轴的允许误差：旋转轴 (deg(度))

当机器人中的一个旋转轴的当前位置和机器人停止位置间的角度差距超过该处在程序重新启动的设置值时，就产生一个警报，并且程序不会被重新启动。

6. 轴的允许误差：线性轴 (mm(毫米))

当机器人中的一个线性轴的当前位置和机器人暂停位置间的差距超过该处在程序重新启动的设置值时，就产生一个警报，并且程序不会被重新启动。

当一个程序被重新启动时，这项功能会将机器人的当前位置与程序被停止时机器人所在的位置进行比较。如果比较结果显示距离、空间方位角和轴的位置中任何一项的差异超出了系统允许误差范围，此功能发出一个警报并停止启动程序。在这种情况下，下列信息会显示在教导盒上：

**The robot position is out of
stop tolerance.
Please select action.
Choosing CONTINUE will
require cycle start.**

STOP CONTINUE

(1) 选择了 STOP(停止)时

选择了 STOP(停止)时，弹出菜单就消失了，并且程序仍然保持暂停。选择了 STOP(停止)后，如果有启动信号输入，允许误差检查被执行，并且弹出菜单又显示出来了。

要恢复程序运行，请通过手动进给，使机器人在允许误差范围内移动，然后输入启动信号。

(2) 选择了 CONTINUE(继续)时

弹出菜单消失了，并且程序保持暂停状态。当在前面的情况下有 start right（立即启动）信号输入时，程序就会被启动。

如果选择 CONTINUE(继续)后，执行手动进给，当下一步程序重新启动时，会再次进行检查。



注意

这项功能不能够被同时应用在线性轨迹功能和常数关节路径功能上

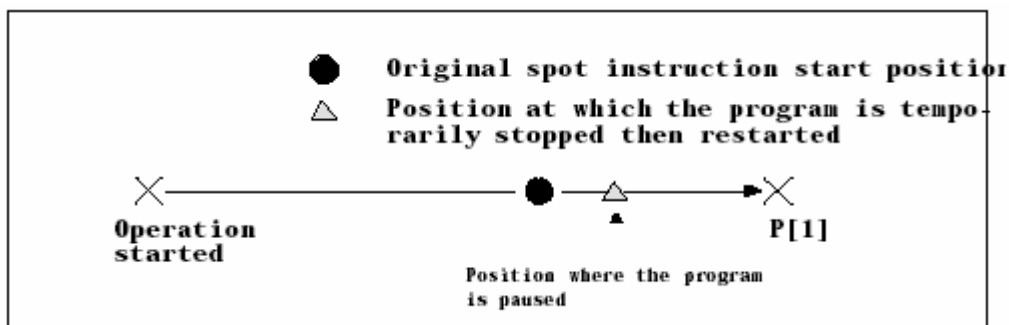
实例 下列给出了使这个功能有效的例子

在下面的例子中，假设一条动作语句被执行，这条动作语句包含一条作为一个附加动作指令的现场指令，如下面所描述：

1: J P[1] 100% FINE SPOT[1]

同时假设设置了一个正的期望值，并且执行了那个期望动作。

例 1 在现场指令执行期间，程序被暂停。



如果程序在现场指令的起始和结尾之间被暂停，当应用一个单个的现场指令的同时，这项功能执行现场指令，这个单个的现场指令描述如下所示。

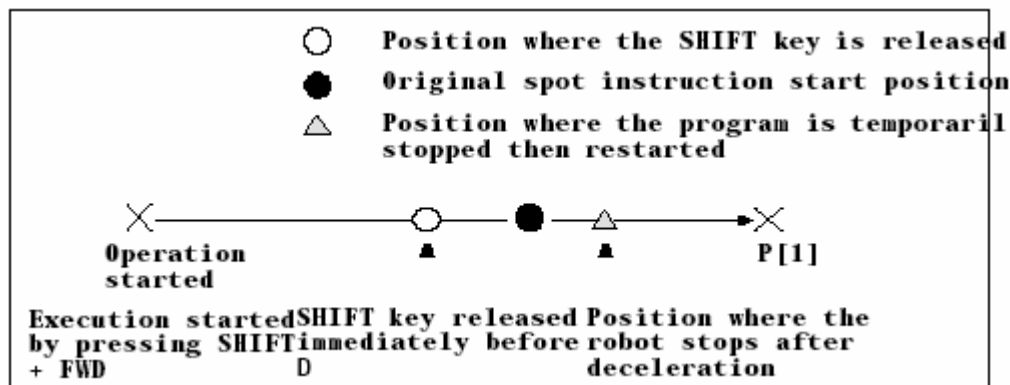
1: J P[1] 100% FINE

2: SPOT[1]

这意味着在到达暂停的动作语句的目标位置后，SPOT[1]指令才被执行。

- * 这仅仅意味着在动作完成时，启动现场指令执行。注意一个指令行不能被分成两行。
- * 上面所述也可以应用到单步执行中。

例 2: 在现场指令执行前, 但机器人已经经过了现场指令的起始位置时, 尝试将机器人立即减速到停止, 。



假设通过按教导盒上的 SHIFT+FWD 键来执行相同的指令。在现场指令将启动的位置之前, 将 SHIFT 键立即释放, 机器人开始减速。然而, 机器人不能够立刻停下来, 只有当它经过如上面显示的现场指令开始点后才能够停下来。

在这种情况下, 机器人在现场指令开始点后停止, 但现场指令没有被执行。这是因为已经释放 SHIFT 键了。这个点不与功能使用与否相关联。)

如果在这个功能使能的状态下, 程序重新启动, 在到达例 1 情况中所描述的动作指令的目标位置 (P[1]) 后, 机器人才开始执行现场指令。

在下面的例子中, 假设执行了一个动作语句, 该动作语句包含一条作为附加动作指令的现场指令, 附加动作指令如下面所描述:

1: J P[1] 100% FINE SPOT[1]

同时假设设置了一个正的期望值, 并且执行了那个期望动作。

在下面的例子中, 假设执行了一个动作语句, 该动作语句包含一条作为附加动作指令的现场指令, 附加动作指令如下面所描述:

1: J P[1] 100% FINE SPOT[1]

同时假设设置了一个正的期望值, 并且执行了那个期望动作。

如果该项功能有效, 如上所述的动作语句暂停, 然后在现场指令 look-ahead (预作准备) 动作开始后, 重新启动程序执行。只有在那一行的机器人动作完成后, 现场指令的执行才被重新启动。

测试

测试是指在现场生产线上自动操作操作机器人前，单独检查机器人的操作。测试这个程序是非常重要的。它必须进行以确保工人和周围设备的安全。

下列两种方法用于测试：

- 单步测试：使用教导盒或者操作面板一行一行的执行程序。
- 连续测试：使用教导盒或者操作面板，从当前程序行执行程序到程序末尾（到达程序结尾标记或者程序结束指令）。

使用教导盒执行测试前，教导盒必须是有效状态。当下列情况教导盒是有效的：

- ✧ 教导盒使能开关被打开。

从操作面板/盒启动测试操作前，操作面板必须处于有效状态。当下列条件满足时，操作面板可以处于这种有效状态：

- ✧ 教导盒使能开关被设置为 OFF（关闭）。
- ✧ 在操作面板/盒上的远程开关被设为本地位置。
- ✧ 外围设备 I/O(输入/输出)“SFSPD”输入是打开的。

开始一个包含动作指令的程序前，必须满足下列操作条件：

- ✧ 为外围设备 I/O(输入/输出)的输入信号 ENBL 必须被打开。
- ✧ 警报必须没有被占据。

典型的测试流程如下所述：

- 1 打开机器锁定开关，执行单步操作，并且检查程序指令盒 I/O(输入/输出)。
- 2 关闭机器锁定，然后从教导盒来执行单步操作以检查机器人操作、程序指令、I/O(输入/输出),等等。
- 3 在低速下执行连续操作。
- 4 在高速下执行连续操作，然后检查机器人的位置盒操作时间。

指定测试执行

指定测试执行是去指定对一个程序执行测试的要求。

```
TEST CYCLE Setup          JOINT 30 %
                           1/7
GROUP: 1
 1 Robot lock:             OFF
 2 Dry run:                OFF
 3 Cart. dry run speed:    300.000 mm/s
 4 Joint dry run speed:    25.000 %
 5 Digital/Analog I/O:    ENABLE
 6 Step statement type:    STATEMENT
 7 Step path node:        OFF

[ TYPE ] GROUP            ON      OFF
```

执行测试的设置

项目	说明
机器人锁定	这个功能指定机器人是否是处于有效状态。 —ON(开)：机器人处于无效状态；它对所有的动作指令均无响应。 —OFF(关)：机器人处于有效状态，一般它接受动作指令。 当机器人锁定功能是 ON(开)，假定伺服系统的电源被打开，按 RESET(重启)键，会重置所有的伺服警报。 注意 即使机器人锁定是 ON(开)，当紧急停止按钮被按下时，不能操作机器人。
试操作	当这个功能设为有效时，机器人以指定的速度“Cart dry run speed”移动。

执行测试的设置项目

项目	说明
笛卡儿试操作速度	这个参数指定了在试操作期间机器人的进给速度。当机器人的动作是在路径的控制之下（线性或者圆周运动控制），机器人以指定的速度做恒定运动。 单位：毫米/秒）
关节试操作速度	这个参数指定了在试操作期间机器人的进给速度。当机器人的动作是在关节的控制之下，机器人以指定的速度做恒定运动。
关节试操作速度	这个试操作速度（手动）表明当执行操作时，机器人的移动速度使用的是试操作设置。当机器人动作是线性或者圆周时，从机器人动作的开始到结束都运用此项显示的速度。
数字/模拟 I/O(输入/输出)	<p>数字/模拟 I/O(输入/输出)指明与外围设备的通讯是否是通过数字 I/O(输入/输出)和群 I/O(输入/输出)信号线。当此项设为无效状态，机器人不能同外围设备发送或接收数字 I/O(输入/输出)信号。在内部，所有的 I/O 信号都被赋予了仿真标志（S），并且这种仿真标志只有当设置被设为有效时，才能清除。</p> <p>当你设置了无效标志，就不能够改变对外围设备的输出。你可以在不改变外围设备状态的情况下，仿真输出。当你设置标志为有效，输出返回到无效标志被设置之前的状态。外围设备的控制返回给处理器。当你设置无效标志时，控制器保留从外围设备到控制器的输入。当你设置标志为有效时，输入返回到无效标志被设置之前的状态。</p>
单步语句类型	<p>单步语句类型指定了如何在单步模式来执行一个程序。</p> <p>—语句：程序每执行一行都被暂停。</p> <p>—动作：程序每执行一个动作指令就暂停。</p> <p>—常规：几乎与 STATEMENT(语句)，然而，被一个 CALL（调用）指令的一个程序不暂停。</p> <p>—TP&动作：除了动作指令，所有的 KAREL 指令处，程序不暂停。</p> <p>注意 “TP&动作” 当前没有被使用。</p>
单步路径节点	当“单步路径节点”被设为 ON(开) 在 KAREL 指令“MOVE ALONG”执行期间，在每个节点处机器人都暂停。

指定测试执行

步骤 1 按 MENUS(菜单)键来显示界面菜单。

2 选择 2 TEST CYCLE，测试循环界面就显示出来了。

- 1 UTILITIES
- 2 TEST CYCLE
- 3 MANUAL FCTNS

TEST CYCLE Setup		JOINT 30 %
		1/7
GROUP:1		
1 Robot lock:	OFF	
2 Dry run:	OFF	
3 Cart. dry run speed:	300.000 mm/s	
4 Joint dry run speed:	25.000 %	
5 Digital/Analog I/O:	ENABLE	
6 Step statement type:	STATEMENT	
7 Step path node:	OFF	
[TYPE] GROUP	ON	OFF

- 3 指定测试执行的要求。
- 4 要改变组号，按 F2 GROUP（组）。

- 1 焊接无效，设置机器锁定开关为 ON(开)，然后从教导盒执行单步操作。检查程序指令和 I/O(输入/输出)。
- 2 焊接无效，然后从教导盒执行单步操作。检查程序指令和 I/O(输入/输出)。
- 3 焊接无效，然后从教导盒执行低速连续操作。

-
- 4 焊接为无效状态，然后从教导盒执行高速连续操作。检查机器人位置和操作时间。
 - 5 焊接为有效状态，然后从教导盒执行连续操作。检查焊接状态。

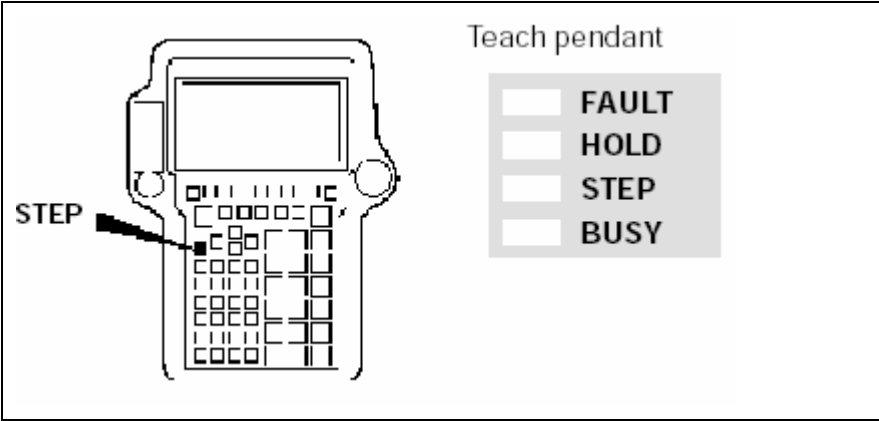
6.3.2 单步测试

执行一个单步测试（单步操作）就是一行一行地来执行程序。当执行一行程序后，程序就进入暂停。在执行一条逻辑指令后，下一行变为当前行，并且光标移动到下一行，但对于动作指令，光标呆在执行完成的一行。

指定单步模式（单步）

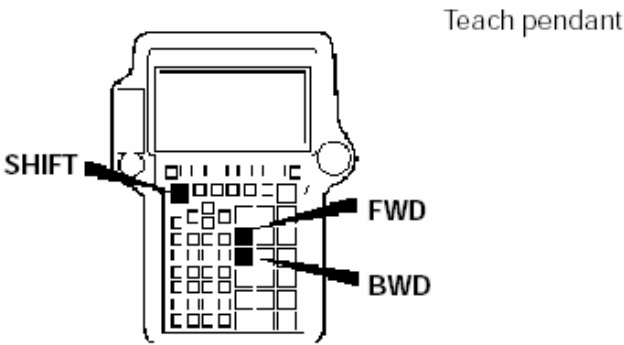
要指定单步模式，在教导盒上按 STEP 键。当指定单步模式时教导盒上的 LED(发光二极管)就亮了。当指定连续操作模式时，STEP LED(单步模式发光二极管)就灭了。

STEP(单步)键



FAULT	HOLD	STEP	BUSY
故障	保持	单步	忙

启动单步操作



单步模式可以以两种方式执行：向前执行和向后执行。

向前执行

在向前执行中，程序以正常的顺序来执行。要向前执行程序，按住 SHIFT 键同时，然后按下和释放教导盒上的 FWD 键。

SHIFT FWD

当启动一个程序时，程序执行了光标指向的一行，然后程序就中止。

当执行一条动作指令时，光标停留在被执行的行上。当执行一条逻辑指令，光标移到下一行。

每次启动程序的向前执行，就执行了程序的下一行。

当在单步模式下执行圆周动作指令，机器人暂停在一个弧的过点附近。而且，如果机器人刚好暂停在过点前，在恢复执行一个程序后，机器人不会停在过点。

向后执行

在向后执行中，程序以反向的顺序来执行。要向后执行程序，按住 **SHIFT** 键同时，然后按下和释放教导盒上的 **BWD** 键。

BWD **SHIFT**

- 在向后执行过程中，仅可以执行动作指令。然而，当程序执行时，一个跳转指令、向前执行指令、向后执行指令、软浮点指令和其他的可选择的移动指令会被忽略。在程序的一行被执行后，光标移动到前一行
- 向后执行不能够执行在下列程序指令被教导的行之前的指令。当你用向后执行来执行这些指令时，光标会移动到包含教导这些指令的行之后的行。
 - 停止指令（PAUSE(暂停)）
 - 终止指令（ABORT(终止)）
 - 程序结束指令（END(结束)）
 - 跳转指令（JMP LBL[]）
 - 用户警报指令（UALM[]）
 - 执行指令（RUN(运行)）
- 下列程序指令不能够执行：
 - 增量指令（INC）

- 一个空白行不影响程序的执行（无论向前或向后执行）。

当被终止的程序重新启动时，执行光标指向行的动作指令，然后程序就暂停了。

每次启动向后执行程序，程序以当前行的动作形式和指定的进给速率以及前一行动作指令的定位数据和定位路径来执行。

- 当前行的动作指令指定一个圆周运动时，机器人沿一定路径移动到目标点，该路径通过当前行指定的交汇点（正常程序执行中的弧动作的起始点）。
- 前一行的动作指令指定一个圆周运动时，机器人应用当前行的动作形式和当前行指定的进给速率移动到一个前一行指定的目的位置。

在程序正被执行时，要使程序的向后执行变为无效状态，在想要的位置插入停止指令（PAUSE）。在暂停指令执行之后，光标返回到程序执行前的位置。

当暂停指令被指定在光标指向行的前一行时，程序的向后执行被禁止。要重新启动程序的向后执行功能，将光标移动到包含有暂停指令一行的前一行（光标指向一行的前面两行）。

在程序里反向执行程序

使用在程序里反向执行程序功能，通过执行反向操作时（**SHIFT+BWD**）来调用子程序，可以让控制从子程序返回到主程序。

提示 即使在一个主程序反向操作过程中一个子程序存在，子程序也不能够被调用。

提示 当在子程序中产生程序终止时，程序不能够返回到主程序。

当从一个子程序到一个主程序执行反向执行时，光标停止在调用主程序教导的子程序的指令所在行。

文件输入/输出单元

通过机器人控制单元，可以使用下列文件 I/O(输入/输出)设备：

- 存储卡
- 软盘

标准设置指定使用存储卡。当使用软盘的时候，按照下面的步骤去改变文件 I/O(输入/输出)设备。用存储卡保存和读取文件的速度很快，这能够大大的提升工作效率。

改变文件 I/O(输入/输出)设备

- 步骤 1 按 MENUS(菜单)键来显示界面菜单。
- 2 选择 “7 FILE”（7 文件）。文件界面就出现了。

6 SETUP
7 FILE
8

MENUS

FILE

MC: *. * 1/17

1 * * (all files)
2 * KL (all KAREL source)
3 * CF (all command files)
4 * TX (all text files)
5 * LS (all KAREL listings)
6 * DT (all KAREL data files)
7 * PC (all KAREL p-code)
8 * TP (all TP programs)
9 * MN (all MN programs)
10 * VR (all variable files)
Press DIR to generate directory
[TYPE] [DIR] LOAD [BACKUP] [UTIL] >

JOINT 10%

- 3 按 F5 UTIL 键,选择 Set Device（设置设备）.于是显示出下面的界面。

Set Device

LOAD [BACKUP] UTIL

F5

1 Floppy disk
2 Back up (FRA:)
3
4

FILE

1 * * (all files)
2 * KL (all KAREL source)
3 * CF (all command files)
4 * TX (all text files)
5 * LS (all KAREL listings)
6 * DT (all KAREL data files)
Press DIR to generate directory
[TYPE] [DIR] LOAD [BACKUP] [UTIL] >

JOINT 10%

- 4 选择使用的文件 I/O(输入/输出)设备。当前选择的文件 I/O(输入/输出)设备的缩写会出现在屏幕的左上角。

FILE FLPY:	Abbreviation	File I/O device
	MC :	Memory card
	FLPY :	Floppy disk
	FRA :	Area used for automatic backup of the F-ROM in the controller

缩写	文件 I/O(输入/输出)设备
MC:	存储卡
FLPY:	软盘
FRA:	在控制器里用来自动备份 F-ROM 的区域

提示 当选中 FLPY 后，在前面的端口设置界面设置软盘驱动器。
插入内存卡

外部存储单元

两种型号的软盘驱动器可以使用:

- Floppy Cassette adapter (A16B--0150--B001) 盒式软盘适配器
- Handy File (A16B--0159--B002) (便携文件驱动器)

3.5 英寸的软盘可以使用。在一个新软盘使用前, 必须按下列的方法进行格式:

软盘格式规范

软盘型号	3.5 英寸, 双面高密盘(2HD)或双面低密 (2DD)
Floppy Cassette adapter (盒式软盘适配器)	2HD, FANUC 格式, 最大 71 个文件
Handy File (便携文件驱动器)	2HD, FANUC 格式, 最大 71 个文件 2HD, MS-DOS 格式 2DD, MS-DOS 格式

磁盘驱动器通过 RS-232-C 串口连接。磁盘驱动器的端口 1 (Port 1) 用于连接。列举了标准的磁盘驱动器设置。

软盘驱动器的标准设置

设备	波特率	停止位	奇偶检验位	数据编码	超时值
Floppy Cassette adapter	9600 baud (波特)	2 bit (比特)	无	ISO	0 sec (秒)
Handy File	9600 baud (波特)	2bit (比特)	无	ISO	0 sec (秒)
Handy FMS-DOS	9600 baud (波特)	1bit (比特)	无	ISO	0 sec (秒)

当使用一个软盘时, 按照改变文件 I/O(输入/输出)设备的说明
另外, 设置软盘驱动器被用来设置通讯端口。



注意

当软盘在进行存取时, 不要将软盘从外存储设备取出, 否则, 会毁坏软盘中的内容。



注意

如果打印机, 软盘驱动器, 显示系统, 或别的设备连接到控制单元, 这些设备应该在机器人开机后才能打开。否则, 这些设备会受到损坏。

在程序选择界面中保存

程序选择界面可以让你把指定的程序作为程序文件保存。

保存程序文件的要求

- 条件 H 正确设置了文件输入/输出设备。
H 当程序要被保存到一张软盘上时，软盘已经准备就绪，并且已经正确设置端口。

- 步骤 1 按 MENUS(菜单)键来显示界面菜单。
2 选择“下一步”(NEXT),然后在下一页选择“1 选择”(1 SELECT)。程序选择界面就出现了。

1 SELECT

2 EDIT

MENUS

Select

56080 bytes free

JOINT 30%

5/5

1

PRG1

PR

[PROGRAM001

]

2

PRG2

PR

[PROGRAM002

]

3

SAMPLE1

JB

[SAMPLE PROGRAM1

]

4

SAMPLE2

JB

[SAMPLE PROGRAM2

]

5

SAMPLE3

JB

[SAMPLE PROGRAM3

]

[TYPE]

CREATE

DELETE

MONITOR

[ATTR]

>

COPY

DETAIL

LOAD

SAVE

PRINT

>

3 按 下一步 (NEXT),然后在下一页按 F4, 保存 (F4 SAVE)。就出现了程序保存界面。

```
LOAD  SAVE  PRINT >                                JOINT 30%
                                                    1 Words
                                                    2 Upper Case
                                                    3 Lower Case          ---Insert---
                                                    4 Options
Select
                                                    ---Save Teach Pendant Program---
Program Name [SAMPLE3 ]
Enter program name
PRG      MAIN      SUB      TEST
```

4 输入被保存程序的程序名, 然后按 ENTER(回车)键。指定的程序就被保存到软盘当中了。

```
Select
---Save Teach Pend
Program Name [SAMPLE3]ENTER
```

提示 在程序文件名中不要含有文件的扩展名。

5 当软盘当中有一个与要保存程序的文件名相同的文件时, 文件无法被保存。

File already exists



注意

如果当前的设备中已经存在一个与指定文件名相同的文件时, 保存功能将不能够覆盖该文件。如果要将新文件保存, 必

6 当一个软盘已经存满, 更换一张软盘, 然后按 F4,继续 (CONTINUE)。

```
No room to save file          CONTINUE CANCEL
```

使用文件界面来保存所有的程序文件

文件界面可以将 RAM 内存中的程序文件或者系统文件，保存到软盘当中。

按 F4，备份（BACKUP）键可以保存下列文件：

- 程序文件(*.TP)：用于储存所有包含程序内容的程序文件。
- 缺省的逻辑文件(*.DF)：用于储存缺省逻辑指令的设置。
- 系统文件(*.SV)：用于储存下列文件。

- 系统变量文件（SYSVARS.SV）
- 伺服参数文件（SYSSERVO.SV）
- 控制数据文件（SYSMAST.SV）
- 宏数据文件（SYSMACRO.SV）
- 参考系设置文件（FRAMEVAR.SV）
- 输入/输出(I/O)配置数据文件(DIOCFGV.SV)
- 寄存器数据文件(NUMREG.VR)
- 机器人设置数据文件

在保存过程中，要中断保存操作，按 前一步（PREV）键。

提示 在控制启动时间，F4 被设置为 RESTOR（恢复），而不是 BACKUP（备份）。当从辅助菜单选择 RESTOR/BACKUP（恢复/备份）时，BACKUP（备份）就显示出来了。

FCTN

2 RESTORE/BACKUP

使用文件界面保存文件

条件 H 正确设置了文件输入/输出设备。

H 当程序要被保存到一张软盘上时，软盘已经准备就绪，并且已经正确设置端口。

- 步骤**
- 1 按 MENUS(菜单)键来显示界面菜单。
 - 2 选择“7 文件”（7 FILE）。文件界面就出现了。

6 SETUP
7 FILE
8

MENUS

FILE	JOINT	30 %
P3: *.*		
1 *	*	(all files)
2 *	KL	(all KAREL source)
3 *	CF	(all command files)
4 *	TX	(all text files)
5 *	LS	(all KAREL listings)
6 *	DT	(all KAREL data files)

Press DIR to generate directory

[TYPE] [DIR] LOAD [BACKUP][UTIL]>

DELETE COPY DISPLAY >

保存程序文件

3 按 F4 “BACKUP”(备份), 然后选择 “TPE programs” (TPE 程序)。

```
1 System files
2 TPE programs
3 Application
LOAD  BACKUP  [UTIL] >
F4
FILE                               JOINT 30%
7 *                                1/13
8 *                                PC (all KAREL p-code)
9 *                                TP (all TP programs)
10 *                               MN (all MN programs)
                               VR (all variable files)
Save FLPY:\SAMPLE1.TP ?
EXIT  ALL  YES  NO
```

-- F2, EXIT (退出) 结束保存程序文件。

-- F3, ALL (全部) 保存全部的程序文件和缺省逻辑指令文件

-- F4 YES (是) 保存指定文件 (程序、缺省逻辑指令)

-- F5, NO (否) 不保存指定文件 (程序、缺省逻辑指令)。在文件保存后, 系统会提

示是否要保存下一个程序文件。

4 选择想要的功能键。在这种情况下, 程序文件 (*.MN) 被保存到软盘。

```
EXIT  ALL  Saving FLPY:\SAMPLE1.TP, please wait...
```

F3

5 当你指定的文件名在软盘中已经存在时, 会显示下列的信息。

```
FLPY:\SAMPLE1.TP already exists
OVERWRITE SKIP CANCEL
```

-- F3, OVERWRITE (覆盖) 指定文件覆盖磁盘中的同名文件, 然后保存。

-- F4, SKIP (跳过) 不保存指定的文件。

-- F5, CANCEL (取消) 结束文件保存操作。

保存系统文件

6 按 F4, SAVE(保存), 然后选系统文件。就显示下列文件。

```
1 System files
2 TPE programs
3 Application
LOAD  BACKUP  [UTIL] >
F4
FILE Backup                               JOINT 30 %
FLPY:\*. *
Saving the following files to FLPY:\
DIOCFGSV.IO
FRAMEVAR.SV
NUMREG.VR
SYSVARS.SV
SYSSERV0.SV
SYSMASST.SV
SYSMACRO.SV
Backup to disk?
YES  NO
```

7 要保存全部的系统文件，按 F4,YES(是)。系统文件 (DIOCFG.SV、FRAMEVAR.SV、NUMREG.VR SYSVARS.SV,SYSSERVO.SV,SYSMAST.SV,SYSMACRO.SV)被保存到软盘上。

YES NO Backing up to disk: FLPY:\SYSVARS.SV

F4

8 当你指定的文件名在软盘中已经存在时，会显示下面的信息。

FLPY:\SYSVARS.SV already exists
OVERWRITE SKIP CANCEL

- F3,OVERWRITE （覆盖）指定文件覆盖磁盘中的同名文件，然后保存。
- F4,SKIP（跳过） 不保存指定的文件。
- F5,CANCEL （取消）结束文件保存操作。

9 当一个软盘已经存满，更换一张软盘，然后按 F4,继续（CONTINUE）。

Disk is full, change to empty disk
CONTINUE CANCEL

批量保存

10 按 F4 BACKUP(备份)，然后选择 ALL of above（上面所有的文件）。

ALL of above

LOAD BACKUP [UTIL] >

F4

FILE

FLPY: * , *

1 * * (all files)

2 * KL (all KAREL source)

3 * CF (all command files)

4 * TX (all text files)

5 * LS (all KAREL listings)

6 * DT (all KAREL data files)

7 * PC (all KAREL p-code)

8 * TP (all TP programs)

9 * MN (all MN programs)


10 * VR (all variable files)

Del Handy File, backup all files?

YES NO

提示 因为 F4 BACKUP(备份)没有出现在控制启动（不是控制启动 2）,批量保存操作不能够被使用。

11 当 F4,YES（是）被选择时，在外部存储设备中的所有文件会被擦除，然后保存所有

 **注意**

在开始一个批量保存前，所有在外部存储设备中的文件会被擦除。在执行一个批量

的数据。使用向后退键可以中断正在进行的处理。一旦当前文件被处理了，就发生一个中断事件

YES	NO
F4	

使用功能菜单来保存

通过从一个功能菜单选择 **SAVE**（保存），可以将一个当前显示在界面的数据保存到软盘中。下列界面上的数据可以被保存：

- 程序编辑界面 程序文件 (*.TP)
- 系统变量界面 系统变量文件 (SYSVARS.SV)
- 位置界面 控制数据文件 (SYSMAS.TSV)
- 宏指令设置界面 宏数据文件 (SYSMACRO.SV)
- 参考系设置界面 参考系设置文件 (FRAMEVAR.SV)
- 寄存器界面 寄存器数据文件 (NUMREG.VR)
- 位置寄存器界面 位置寄存器数据文件 (POSREG.VR)
- 堆垛寄存器界面 堆垛寄存器数据文件 (PALREG.VR)
- 输入/输出(I/O)界面 输入/输出(I/O)配置时间界面 (DIOCFGSV.IP)
- 每个缺省逻辑指令的编辑界面 每个缺省逻辑指令 (*.DF)

使用一个功能菜单保存文件

条件 H 正确设置了文件输入/输出设备。

H 当程序要被保存到一张软盘上时，软盘已经准备就绪，并且已经正确设置端口。

保存程序文件

步骤 1 显示程序编辑界面或者程序选择界面。

Select		JOINT 30 %	
		49828 bytes free 1/5	
No.	Program name	Comment	
1	PR0G001	PR [PROGRAM001]	
2	PR0G002	PR [PROGRAM002]	
3	SAMPLE1	JB [SAMPLE PROGRAM 1]	
4	SAMPLE2	JB [SAMPLE PROGRAM 2]	
5	SAMPLE3	JB [SAMPLE PROGRAM 3]	

[TYPE] CREATE DELETE MONITOR [ATTR]>

2 要显示一个功能菜单，按 **FCTN**（功能）键。

3 选择 “2 SAVE”(2 保存)。

被选择的程序文件就被保存了。

```
1 QUICK/FULL MENUS
2 SAVE
3 PRINT SCREEN
```

FCTN

4 当软盘当中有一个与要保存程序的文件名相同的文件时，文件无法被保存。

File already exists

5 当一个软盘已经存满，更换一张软盘，然后按 **F4**, **CONTINUE**（继续）。所有要存储到被更换软盘的数据都保存到更换后的软盘中。

Disk is full, change to empty disk
CONTINUE CANCEL

保存别的文件

步骤 1 显示你想保存的界面。

DATA Registers		JOINT 30 %
		1/32
R[1:	COUNTER1]=12	
R[2:]=0	
R[3:]=0	
R[4:]=0	
R[5:]=0	
R[6:]=0	
[TYPE]		

2 要显示一个功能菜单，按 FCTN（功能）键。

3 选择 “2 SAVE”(2 保存)。

正被显示界面的内容就被保存了。

1 QUICK/FULL MENUS
2 SAVE
3 PRINT SCREEN

FCTN

4 当软盘当中有一个与要保存程序的文件名相同的文件时，该文件被覆盖。

5 当一个软盘已经存满，更换一张软盘，然后按 F4, CONTINUE（继续）。所有要存储到被更换软盘的数据都保存到更换后的软盘中。

FLPY-005 Disk is full	
DATA Registers	JOINT 30 %

自动备份概述

- 在下列定时的时间自动备份功能执行文件菜单中的“全部备份”处理。
 - 指定时间（最多设置 5 个）。
 - 指定的 DI 被打开。
 - 控制器启动。（可以指定间隔。）
- 存储卡(MC:)和在控制单元的F-ROM的自动备份区（FRA）可以被指定为备份拷贝的目的存储区。缺省指定的是FRA.
- 自动备份功能可以在同一个设备中管理备份的许多版本。即使你备份了一个错误的程序或者设置，你可以加载备份的前一个版本来更正。可以保存的版本的数目可以从1设置到99。 缺省值是2。）
- 用于自动备份的存储设备在使用前需要为自动备份进行初始化。许多外部存储设备不能够应用于自动备份，因为它们不能够进行自动备份的初始化。因此，如果试图为一个没有为自动备份进行初始化的存储卡自动产生一个备份拷贝，那么它的内容可能会丢失。
FRA:不需要初始化，因为它已经提前被初始化过了。
- 如果在自动备份过程中关闭了控制单元，或者自动备份未完成而停止，最后的备份拷贝自动恢复到系统中。不完整的备份文件会留在存储设备中，最后的备份文件会在任何时间内读取。

提示 这项功能自动保存所有文件，如果用于自动备份的存储设备发生故障，存储于其中的数据可能无法读取。在这种情况下，可能发生不可预知的事故，因此非常必要将备份保存到另外的存储设备，比如同样的存储卡等。

版本管理

自动备份功能会在一个设备里保存许多备份。要保留的版本的数目在初始化设备的时候被设置。你可以在任何时候，通过“Maximum number of versions”(版本的最大数目)选项来改变要保留版本的数目。版本的数目超出指定的数目时，最早的版本会被自动删除。

如果设备是 **FRA**：当控制单元的 **F-ROM** 中存储区，自由空间的大小小于 1 兆字节，最早的备份版本会被自动删除。在这种情况下，实际保有的备份版本的数目小于“Maximum number of versions”(版本的最大数目)。如果控制单元的 **F-ROM** 中的存储区自由空间的大小，小到不足以保留另一个备份版本，那么，在自动备份执行的过程中，将会产生一个错误。

如果由于存储卡的存储空间容量不足而导致不能够保存指定数目的备份版本，在自动备份执行的过程中，将会产生一个错误。要指定备份版本的一个大致的数目，可以假定一个备份版本需要的存储空间是“程序大小 + 200 千字节”。

在自动备份的过程中，如果由于存储空间不足而产生一个错误，请减小“Maximum number of versions”(版本的最大数目)的指定值。这会使一个老的版本被删除，因此而增加了存储设备的自由空间的大小。

一旦一个备份的版本由于减少“Maximum number of versions”(版本的最大数目)而被删除，将不能够通过增加这个值而来恢复。

备份被存储在单个的子目录当中。

执行自动备份时，备份文件被保存到根目录下，然后这些文件被拷贝到相应的目录当中。

通过文件菜单仅仅可以访问根目录下的文件，因此备份的最新版本可以通过文件菜单来加载。你可以加载更老的版本。 (→8.7.6 备份的恢复)

当执行文件菜单中的“all backup”(所有备份)一项，将备份保存到为自动备份初始化的设备当中时，同自动备份一样，文件的会被拷贝到相应的子目录当中。

如果在自动备份过程中，控制单元被关闭或者备份未完成就被停止，在当前备份任务中产生的所有备份文件会被删除，选择的最后的备份版本被恢复到根目录。

备份的恢复

通过自动备份保存的备份文件，可以通过文件菜单来加载。在被控制的开始菜单的文件菜单中，选中上面所述所有的文件，它们就能够被同时读取。

通常最新版本的备份位于根目录下，该版本可以通过文件菜单来加载。

能够通过下列的操作来加载以前的版本。

- (1) 在“Loadable version”(可加载的版本)一项，按 F4(CHOICE) (选择)，就显示出包含设备当中所有版本备份时间的菜单。

```
1 99/06/16 12:00      5 99/06/14 12:00
2 99/06/15 23:30      6 99/06/13 23:30
3 99/06/15 12:00      7 99/06/13 12:00
4 99/06/14 23:30      8 -- Next Page --
AUTO BACKUP          JOINT 100 %
Version Management-----
13 Maximum number of versions:      1
14 Loadable version:      99/06/16 12:00
```

[TYPE] INIT_DEV [CHOICE]

- (2) 请选择要加载的版本，然后“Loadable version”(可加载的版本)一项显示了选择版本的时间。此时，被选择备份版本的文件被拷贝到根目录下。

- (3) 可以在文件菜单加载选择版本的文件。

当被控制的开始被执行时，在被控制的开始菜单的文件菜单中，选取上面所述所有的所有，他们就能够被同时读取。
