

软 件

K R C...

配 置

KUKA系统软件(KSS)

版本**4.1**

出版日期**2004年1月09日**

版本:**06**

@版权 **KUKA Roboter GmbH**

复制或者向第三者传授本文，包括本文的段落章节，必须经过出版者的明确许可。

本文中未作描述的、控制部分中的其它功能有可能起作用。尽管如此，在重新供货或提供服务时，用户无权对上述功能提出要求。

我们对本印制品就其内容同它所描述的硬件和软件的一致性做过审查，但是它们之间的偏差在所难免。所以，我们对上述一致性不做承诺。本印制品中的数据和说明受到定期检查，必要的修改将在后续的版本中给出。

在不对系统功能产生影响的前提下，保留技术更改权。

PD Interleaf

目录

1 配置系统	7
1.1 综述	7
1.2 功能	11
1.2.1 屏幕保护	11
1.2.1.1 屏幕保护的设置	11
1.2.1.2 功能	14
1.3 “Configure”菜单	15
1.3.1 输入/输出	15
1.3.1.1 夹钳	16
1.3.1.2 外部自动方式	16
1.3.2输入/输出驱动	17
1.3.2.1 编辑Config	17
1.3.2.2驱动复位	18
1.3.2.3 I/O配置	21
1.3.3 Submit解释程序	22
1.3.4 状态键	23
1.3.5 手动(倍率)	24
1.3.5.1 Program--OV--Steps(POV)	24
1.3.5.2 Jog--OV--Steps(HOV)	24
1.3.5.3 鼠标位置	24
1.3.5.鼠标配置	24
1.3.6 U用户组	25
1.3.7 当前工具/基本	27
1.3.8 工具定义	28
1.3.9 On/Off选项	31
1.3.9.1 强制冷启动	31
1.3.9.2 不能断电延时	31
1.3.10 其他功能	32
1.3.10. 语言	32
1.3.10. 密码	34
1.3.10. 编辑器	35
1.3.10. Office GUI开/关	38
1.3.10.工作区域监测	40
1.3.10. 技术选择	41
1.3.10. 重新初始化	43
1.4 “File”菜单	44
1.4.1归档	44
1.4.2存储	44
2系统配置，专家	45
2.1 变量综述	45
2.1.1 显示	45
2.1.2 配置	45
2.1.3 编辑“ConfigMon.ini”	48
2.2 长文本	49
2.3启动时的速度降低	51
2.4自动外部方式下确认信息	53

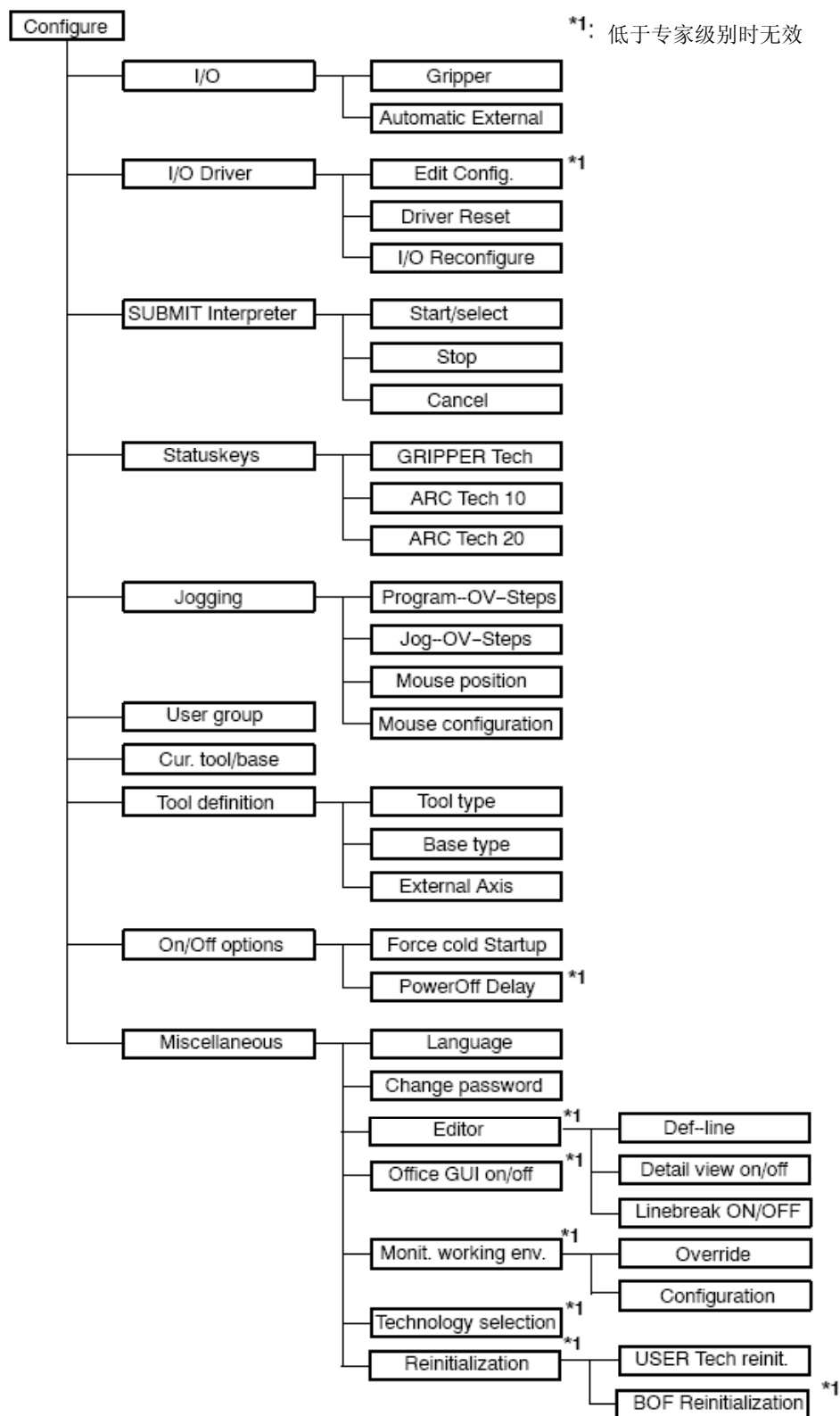
2.5 控制柜监测	54
2.5.1 外部风扇	54
2.5.2 PC 风扇	54
2.5.3 硬件报警的可配置输出	54
2.5.4 电机电缆监测	54
2.6 模拟输入/输出 (I/O 模拟)	55
2.6.1 功能	55
2.6.2 选项	57
2.6.3 使用的变量	58
2.7 5个原点位置	59
2.7.1 “R1\MaDa\$machine.dat” 文件	59
2.7.2 “Steu\MaDa\$machine.dat” 文件	60
2.8 工作区域监测	61
2.8.1笛卡儿坐标系工作区域监测	61
2.8.1.1 配置	61
2.8.1.2 倍率	66
2.8.1.3 例子	67
2.8.2 轴工作区域监测	72
2.8.2.1 功能的原则	72
2.8.2.2 配置	75
2.8.2.3 倍率	79
2.8.2.4 例子	80
2.9 力矩方式（软件伺服）	89
2.9.1 综述	89
2.9.1.1 限制和危险	89
2.9.1.2 力矩方式应用的例子	89
2.9.2 功能的原则	90
2.9.3 软轴有效的例子	91
2.9.3.1 轴1为软轴	91
2.9.3.2轴3为软轴	91
2.9.4 定义轴力矩的例子	92
2.9.5 力矩方式的变量	92
2.10 干涉的监测	94
2.10.1 功能	94
2.10.2 配置	94
3外部自动	99
3.1 综述	99
3.2 配置接口	99
3.2.1 输入	100
3.2.1.1 输出	101
3.3 自动系统开始	102
3.4 技术-特殊的组织程序CELL.SRC	103
3.5 P00 (AUTOMATIC EXTERNAL) 模块	105
3.5.1 EXT_PGNO 功能	105
3.5.1.1请求主计算机的程序号	105
3.5.1.2 收到程序号的通讯	105
3.5.1.3 错误处置	106
3.5.2 功能	107
3.6 信号说明	109

3.6.1 输入	109
3.6.1. PGNO_TYPE	109
3.6.1. PGNO_LENGTH	109
3.6.1. PGNO_FBIT	109
3.6.1. REFLECT_PROG_NR	110
3.6.1. PGNO_PARITY	110
3.6.1. PGNO_VALID	110
3.6.1. EXT_START	110
3.6.1. MOVE_ENABLE	111
3.6.1. CHCK_MOVENA	111
3.6.1. CONF_MESS	111
3.6.1.11 DRIVES_ON	112
3.6.1.12 DRIVES_OFF	112
3.6.2输出	113
3.6.2.1 STOPMESS	113
3.6.2.2 PGNO_REQ	113
3.6.2.3 PGNO_FBIT_REFL	113
3.6.2.4 APPL_RUN	113
3.6.2.5 PERI_RDY	113
3.6.2.6 ALARM_STOP	114
3.6.2.7 USER_SAF	114
3.6.2.8 T1,T2,AUT,EXTERN	114
3.6.2.9 ON_PATH	114
3.6.2.10 NEAR_POSRET	115
3.6.2.11 PRO_ACT	115
3.6.2.12 IN_HOME	116
3.6.2.13 ERR_TO_PLC	116
3.6.3 其它变量	117
3.6.3.1 PGNO	117
3.6.3.2 PGNO_ERROR	117
3.7 信号图	118
3.7.1 自动系统开始和借助于PGNO_VALID程序号确认的正常操作	118
3.7.2自动系统开始和借助于\$EXT_START程序号确认的正常操作	119
3.7.3 动态制动后重新开始(操作安全和重新开始)	120
3.7.4 轨迹保持急停后重新开始	120
3.7.5运动使能后的重新开始	122
3.7.6用户STOP后的重新开始	123
3.8 更多的说明	124
3.8.1被动停止后的重新开始	124
3.8.2单段执行程序	124
3.8.3返回到编程轨迹的速度	124
3.9 配置的实例	125
3.9.1 声明	125
3.10 信息	127

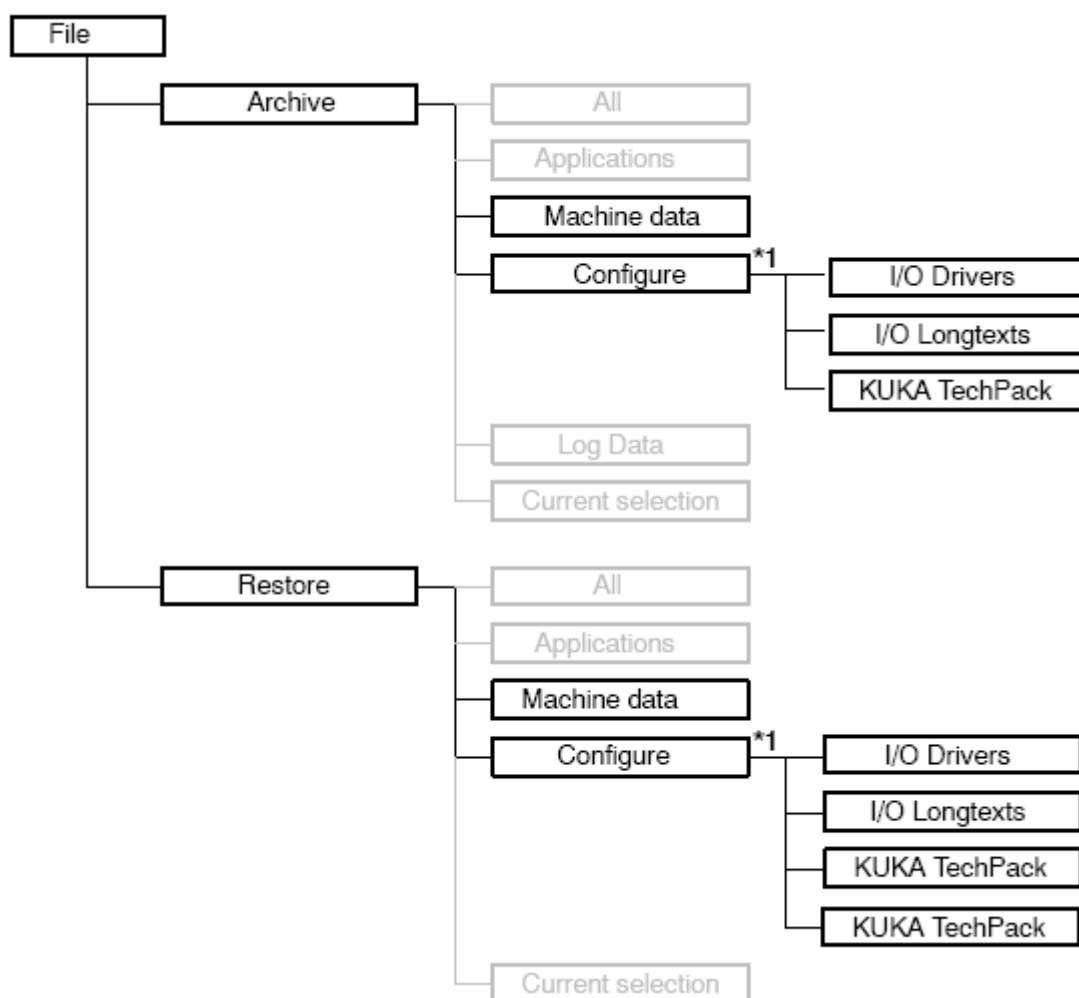
1 配置系统

1. 1 综述

这一章说明怎样配置系统.配置功能的大多数通过“Configure”菜单来实现.



进一步的功能在“File”菜单找到.



下表列出了本章叙述的功能

功能	意义
屏幕保护	规定的时间过后屏幕保护有效
配置菜单	
I/O	设置夹钳和外部自动接口
I/O Drive	配置和复位周边接口
SUBMIT interpreter	开始和停止Submit解释程序
Status keys	为Gripper Teach和ARC Teach分配状态键
Jogging	程序和手动倍率，鼠标设置
User group	使用密码的访问级别
Cur.tool/base	当前刀具、基本系统和外部运动系统
Tooldefinition	设定刀具类型、基本类型和外部运动系统
On/Offoptions	冷起动（下一次控制器运行时，进行一个冷起动）和电源断开等待时间（配置下一次启动的等待时间）
其他功能	
Language	用户图形接口上使用的语言
Changepassword	改变用户访问级别的密码
Editor-->DEF--line *1	在程序中显示DEF行
Editor-->Detailview on/off*1	在专家方式时的附加说明
Editor-->Linebreak ON/OFF*1	程序窗口的换行点
OfficeGUIon/off*1	为鼠标操作显示KCP操作控制组件
Monitor.work.envelope-->Override*1	断开工作区域保护的监测
Monitor.work.envelope-->Configuration*1	直角坐标系和轴工作区域的设定
Technologyselection*1	激活或取消附加的技术
Reinitialization-->USERTechreinitialize*1	USERTechfilesarereinitializedwithoutrebootingthesystem
Reinitialization-->BOFREinitialization*1	TheGUI(graphicaluserinterface)isreinitializedwithoutrebootingthesystem
*1低于专家级别时无效	

“File”菜单	
归档	
Machine date	保存机床资料到软区
Configure*1	保存各种配置设定到软区
恢复	
Machine date	从软区读回机床数据
Configure*1	从软区读回配置设定
*1低于专家级别时无效	

1. 2 功能

1. 2. 1 屏幕保护

增加KCP照明荧光灯的服务寿命, 应降低背景亮度. 正常荧光灯的服务寿命大约10000小时, 相当于连续工作一年. 由断开背景亮度, 理论上的服务寿命能增加一倍.



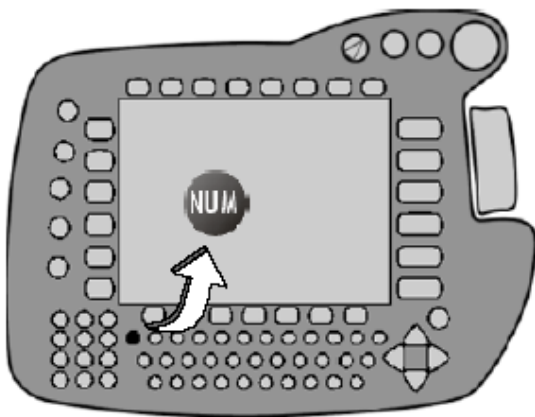
为了能够设置屏幕保护, 要求WINDOWS结构的键盘, 因而必须转换到专家级别. 这些应用在标准级别时不能实现.



有关WINDOWS结构的键盘的详细说明参阅Operating Handbook, 在资料[Operator Control]的[The KUKA Control Panel KCP]章的“Toggling to the Windows interface”节.

1.2.1.1 设置屏幕保护

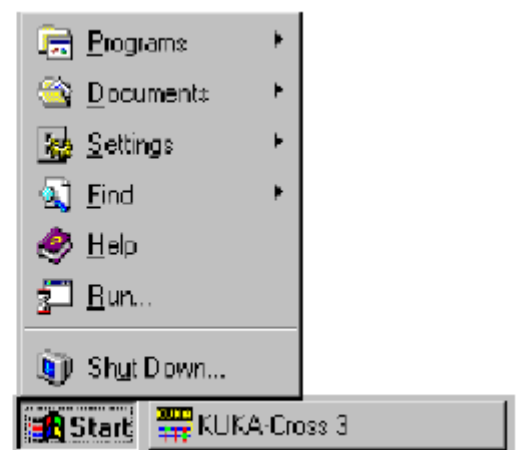
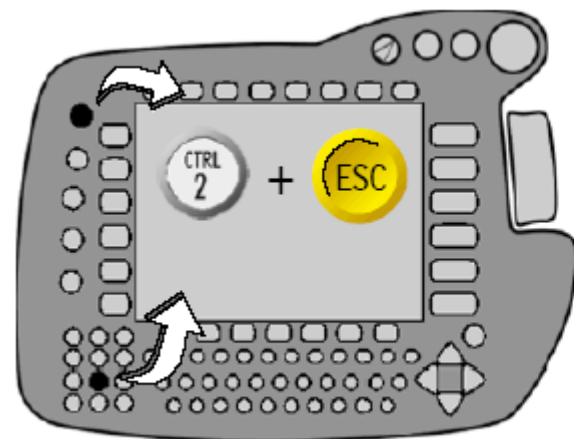
确保显示在状态窗口的“NUM”不在有效状态, 使用字母数字键盘的控制功能.



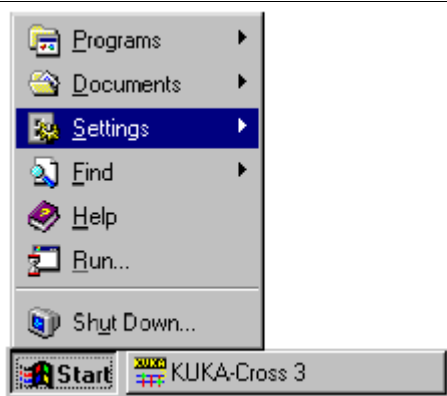
光标控制功能有效



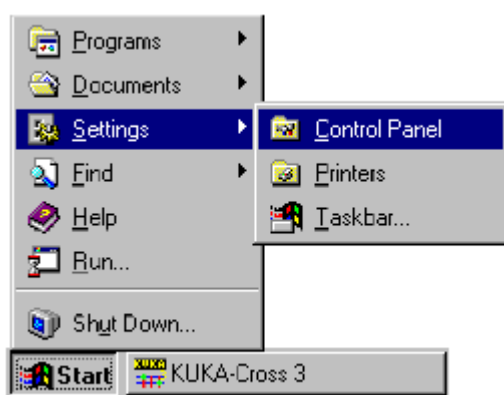
在Windows开始菜单里, KUKA屏幕保护功能必须是有
效的. 有同时按下“CTRL”和“ESC”键, 打开开始菜单.



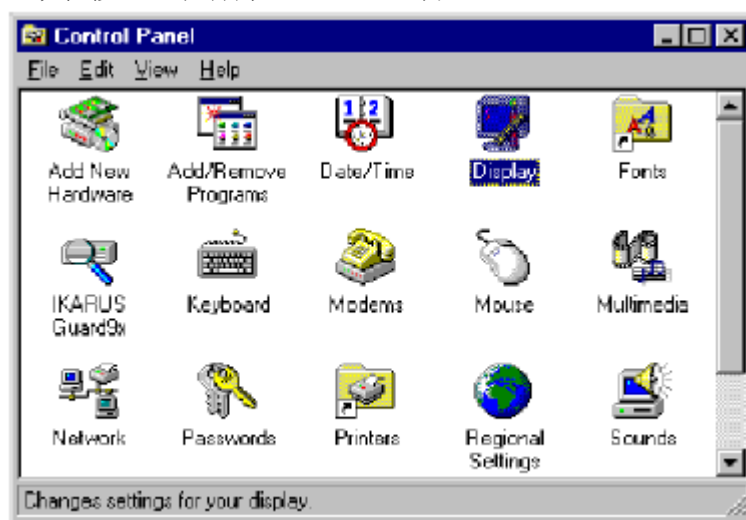
使用“↑”和“↓”键选择“Settings”菜单.



由按Enter键或“→”键打开子菜单.

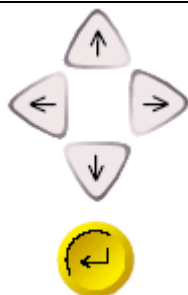


现在按Enter键打开Control Panel窗口



你需要让Control Panel到前边,为此,你保持按下“ALT”键,再重复按“Tab”键多次,直到你找见需要的窗口,然后释放这两个键.

使用箭头键使光标放在“Disply”图标上,按Enter键,打开适当的显示窗口.

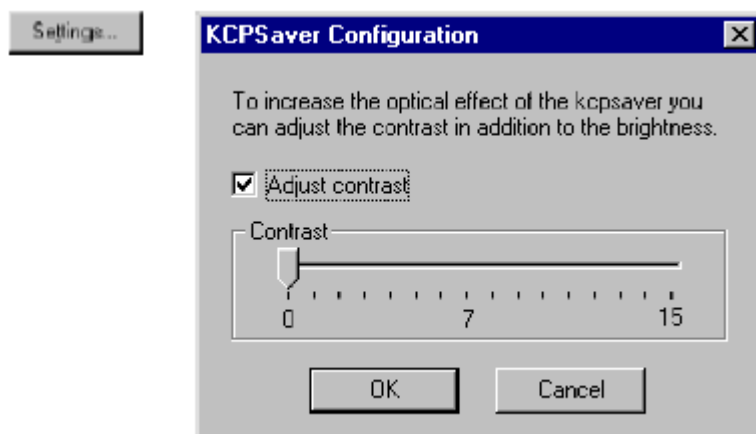


选择屏幕保护标记,为此,你需要保持按下“CTRL”键,再重复按“Tab”键多次,直到你找见需要的项目,然后释放这两个键。

按“↑”和“↓”键多次,直到项目“Kcpsaver”显示在屏幕上。



再次使用“Tab”键移动光标到“Settings”框,按Enter键,下面的窗口打开:



空格键打开或关闭项目“Adjust contrast”既调整对比度. 使用“Tab”键移动光标到“Contrast”设置,使用“←”和“→”键进行调整. 使用“Tab”键移动光标到“OK”按钮.如果这个键被再次按下,可使用“Cancel”按钮取消设定。

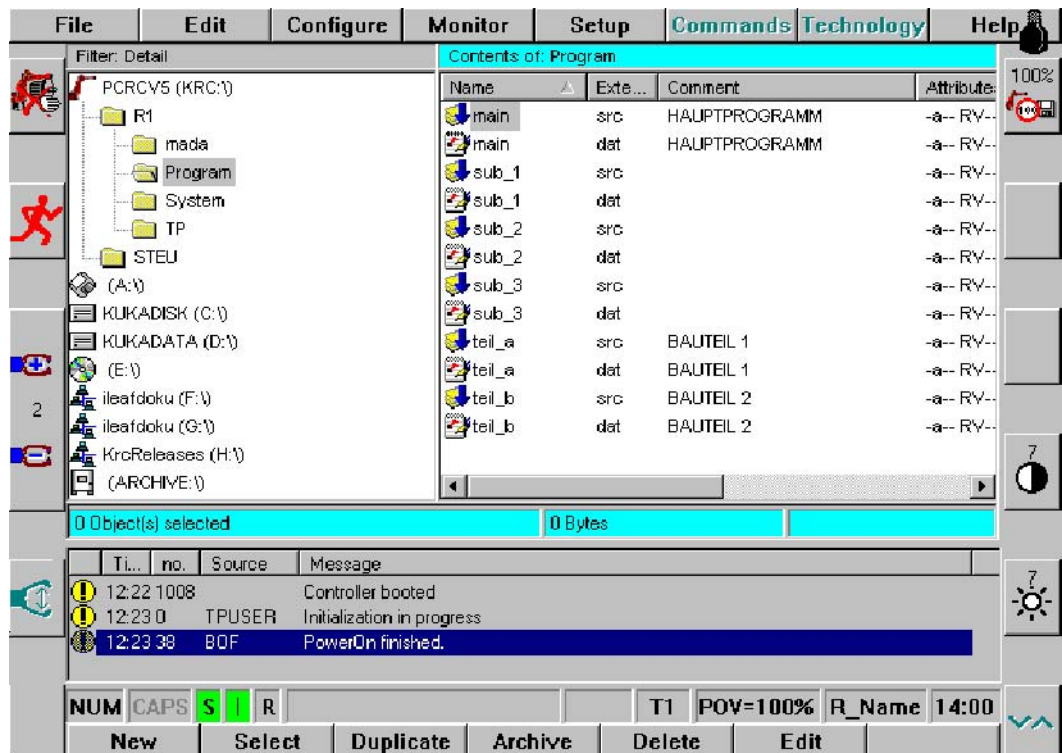


在显示特性“Display properties”窗口,你能够设置屏幕等待时间,由“Tab”键和“↑”和“↓”键,能够设定时间在1--60分钟。

接受屏幕保护设定使用“OK”按钮,取消使用“Cancel”按钮。

1.2.1.2功能

规定的时间内不做任何操作，屏幕保护功能有效，也就是说,亮度被设置到“0”.如果作过相应的设置，对比度降低到设定值。当屏幕保护有效时，鼠标变成一个亮点在显示器的右。



按KCP上的一个键后，屏幕保护被终结。取消屏幕保护时，所按的键也执行相对应的功能.菜单键、软键、字母数字键和箭头键例外，这些键仅仅是取消屏幕保护。



操作“紧急停止”“驱动ON/OFF”和“方式选择开关”，执行相应的功能，不取消屏幕保护。

1.3 “Configure”菜单

菜单“配置”根据功能分组，设置机器人控制系统。

在选择这个菜单键后, 下面的菜单打开:



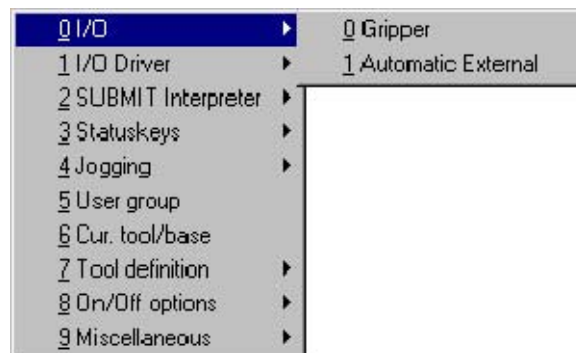
具体菜单项目的详细叙述在下面介绍。



关于菜单的处理、在线格式和窗口状态叙述在[The KUKA Control Panel KCP]章.

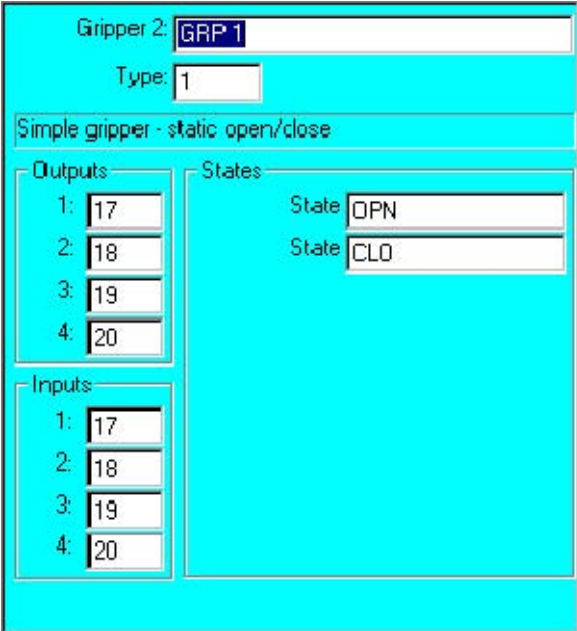
1.3.1 I/O

为夹钳和‘Automatic External’接口设置的项目叙述在下面的菜单“I/O”.



1.3.1.1 夹钳Gripper

一旦选择 “Gripper”，为Gripper配置的状态窗口被打开.有效的gripper的确省设置号是16.



Gripper<Name>

Gripper 的名字；24个符号。

Gripper type<Number>

Gripper的功能类型

Outputs<Number>

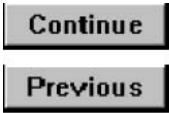
机器人控制器输出到Gripper执行机构的分配

Inputs <Number>

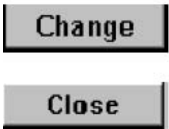
机器人控制器输入到Gripper传感器的分配

State<Name>

Gripper状态的选定，取决于Gripper的类型；11个字符。

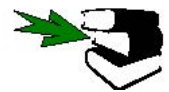


软键“Continue”和“Previous”用于在可能的16个夹钳Gripper之间转换。



按软键“Change”保存所做的设置。

使用软键“Close”关闭当前的状态接口。



关于 “Grippers”更详细的介绍在资料[Gripper Tech H50].

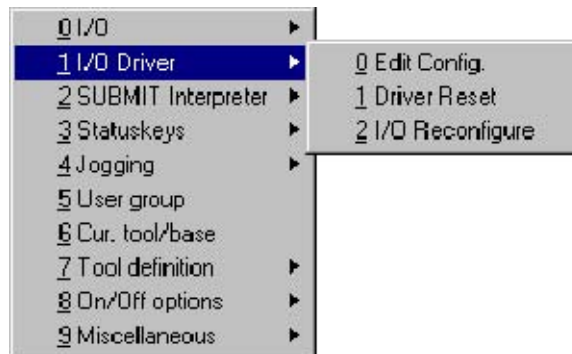
1.3.1.2外部自动



关于 “Automatic External” 接口，请参考 **Programming Handbook** 。在 [Configuration]章的[Automatic External]节.

1.3.2 I/O 驱动器

使用这个功能，配置和复位机器人系统的周边接口。



1.3.2.1 编辑配置

将文件“IOSYS.INI”装入编辑器进行编辑。这个文件保存在目录“C:\KRC\Roboter\Init\”。



低于用户级别“Expert”时，此功能无效。

```

1  ;=====
2  ; IOSYS.INI - Configuration file for the IO-System
3  ;=====
4  ; For configuration help go to the end of this file.
5  ;-----
6
7  [CONFIG]
8  VERSION=2.00
9
10
11 [DRIVERS]
12 MFC=0,mFcEntry,mFcdrv.o
13 ;INTERBUS=1,ibusInit,ibusdrv.o
14 ;DEVNET=2,dnInit,dndrv.o
15 ;BOSCH=3,boschInit,boschdrv.o
16 ;PERCEPTRON=4,percInit,percdrv.o
17 ;SBIP=5,sbipInit,sbipdrv.o
18 ;FIP10=6,fip10Init,fipiodrv.o
19 ;PROFISL=7,pbslInit,pfbsldrv.o

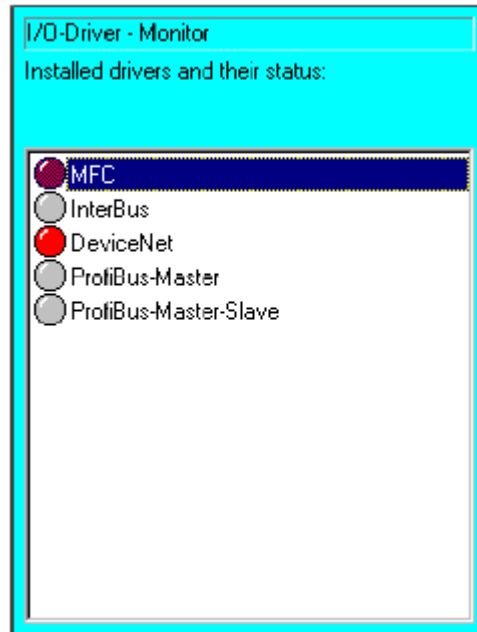
```



关于状态行的详细说明在Operating Handbook的[Operator Control]部分,[Executing, stopping and resetting a program]章,[Selecting and opening a program]节的“Program status line”里。

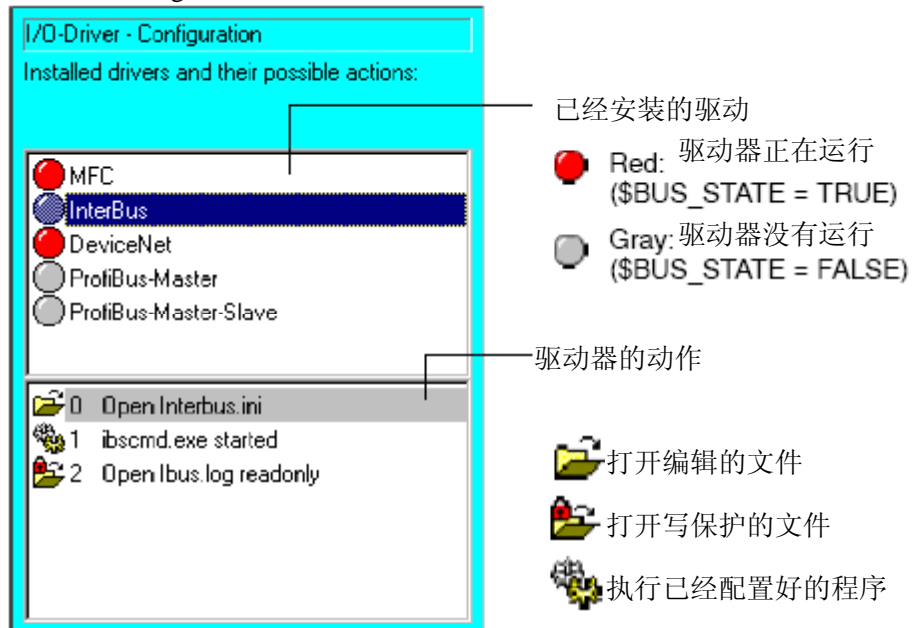
1.3.2. 2驱动复位

驱动能够被显示、安装、卸载和使用菜单指令进行变更.当选择菜单项目“I/O Drive”时，下面的状态窗口出现：

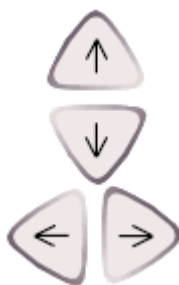


Configure

使用软键“Configure”改变I/O配置，下面的窗口出现：



状态窗口划分为两部分，上面的部分是驱动列表，下面的部分是动作列表。



使用“↑”和“↓”键选择要求的驱动或动作。

使用“→”和“←”在驱动表和动作表之间转换。

软键“New inst.”，“Iosys”和“Deinstall.”在低于专家级别时不能选择。

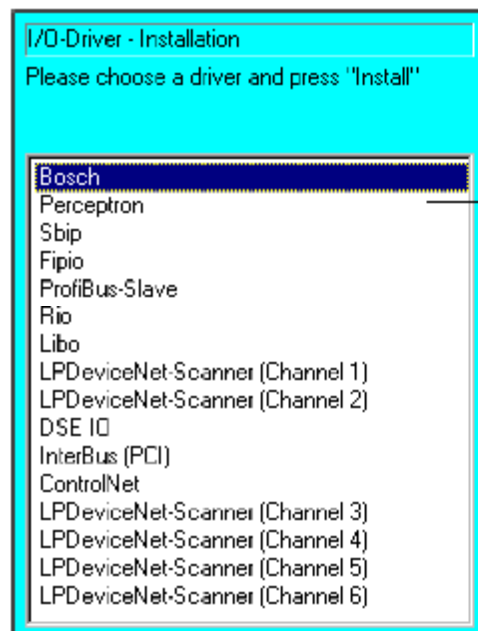


驱动表

New inst.

用这个功能,打开一个安装附加驱动的状态窗口

口.



能安装的驱动列表, 已安装的驱动不在这儿列表。

Install

“Install”安装选择的驱动

Back

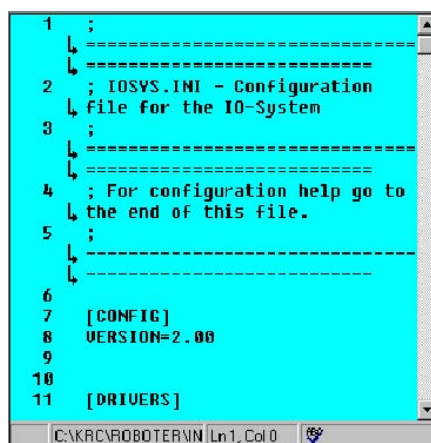
“Back” 返回到主I/O配置窗口

Close

“Close” 关闭I/O drivers的配置.

iosys

软键“iosys”装载文件IOSYS.INI 进入编辑器。

**Close**

软键“Close”保存改变, 关闭编辑器

NAVIGATOR

“Navigator”让你进入机器人控制器Navigator.

Reconfig.

所有的I/O已配置，没有程序可以选择。在这个功能执行期间，软功能键无效。

Info

为获得更详细的说明，按软键“info”，它装载文件“IOSYS.LOG”进入有写保护的编辑器。

```

1 KUKA IOSYS LOGFILE
2 -----
3 Log Date 10.5.1 : 08:55
4
5 open ini-file successful,
6 version 2.00
7
8 driver MFCDRU information
9 object file
10 driver ready 1
11 driver version 0200
12 function entry
13 input objects 1
14 output objects 2
15
16 driver DNDRU information
17 object file
18 driver ready 1
19 driver version 0215

```

Close

软键“Close”关闭编辑器

NAVIGATOR

“Navigator”带你进入机器人控制器Navigator.

Deinstall.

当软键“Deinstall”按下时，选择的驱动被卸载，记录从驱动表上删除。

Reset

选择的驱动被复位。

动作表

如果动作表有效，软键“Deinstall”和“Reset”无效,它们由”Jump”和”OK”替代.

Jump

光标移动到驱动表.

OK

选择的动作被执行.下列动作有效.

装入文件到编辑器

装入文件到有写保护的编辑器

执行定义的程序

Close

使用软键”Close”结束动作,关闭窗口.

1.3.2.3 I/O重新配置

这个菜单用于复位到控制器的安装状态,初始化文件.inf被装入,总线进行相应的配置

	Time	no.	Source	Message
	13:16	202		I/O Reconfiguration started, please wait
	13:16	203	/	General motion enable

出现在信息窗口的信息必须确认.

	Time	no.	Source	Message
	14:57	1210	/	Ackn. general motion enable

Ackn.

软键“Ackn.”删除选择的信息.

Ackn. All

软键 “Ackn.All”删除所有被选择的信息.

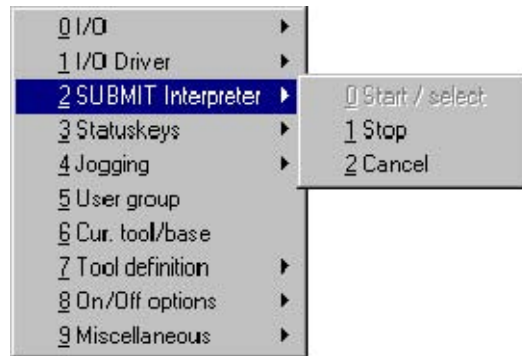


这个仅在方式“T1”,“T2”或“AUT”下有效.

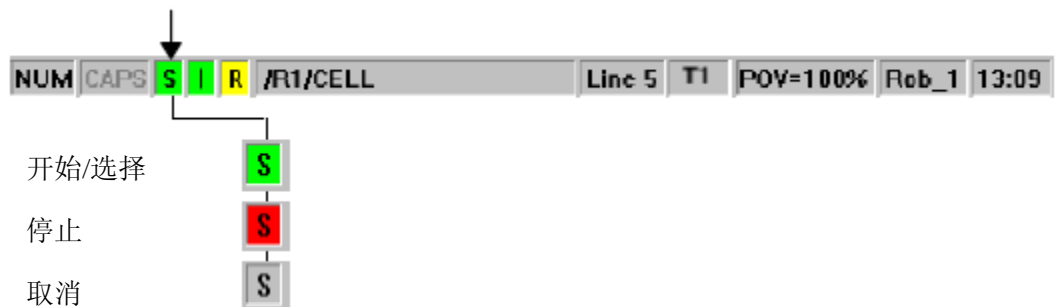
1.3.3 Submit 解释程序

Submit解释程序是一个后台运行程序,它与机器人程序同时运行.由于这个程序运行完全独立于选择的机器人程序,所以它能够用于不同控制任务的所有方式.这些可以包括,例如,冷却回路的控制与监视,安全设备的监视和附加的周边设备的监视.当这些任务能由KRC1调整时,这个能够呈现一个附加的PLC有较少的不必要的任务.

这儿你能够开始、停止或取消Submit解释器。



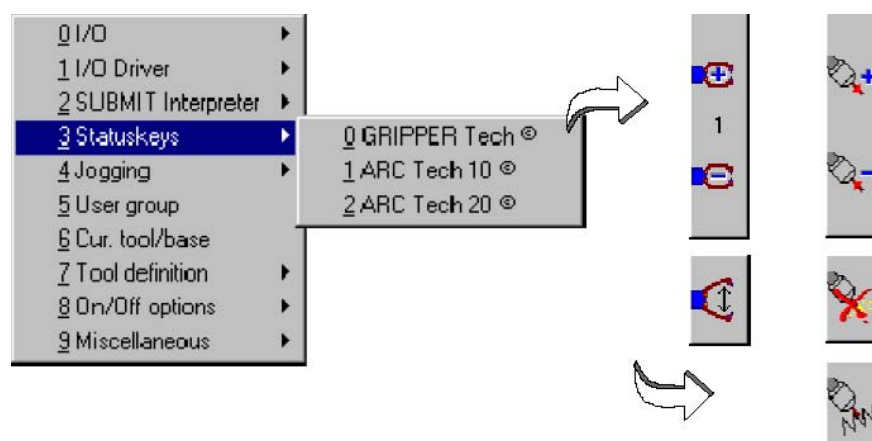
Submit解释器的状态在任何给定的时间都显示在状态行上，绿色意味着解释器正在运行，红色表示解释器在停止，如果色彩不明亮，表示没有选择Submit解释器。



仅可能在T1或T2方式

1.3.4 状态键

这儿选择的功能被分配到自由有效的状态键（在显示器的左下角）



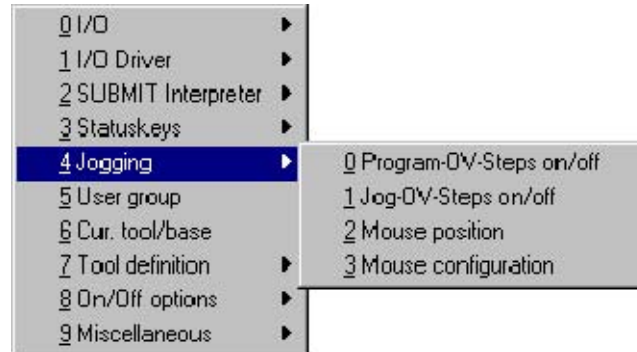
仅已经安装的技术软件包在这个菜单有效



如果其它技术被安装，这个子菜单可以变更。
在系统重新装入后选择的项目保留相同。

1.3.5手动(倍率)

这个菜单让你设置手动倍率（HOV）以及程序倍率（POV）的增量，鼠标位置和配置也能改变。



1.3.5.1Program--OV--Steps(POV)

POV值通常以1%的增量变化.项目“Program--OV—Steps on/off”使编程的速度改变为100、75、50、30、10、3、1%。



使用程序倍率状态键右面的“+/-”键设置倍率。



在 Operating Handbook 的 [Executing, stopping and resetting a program] 章的 “Setting workvelocity (program override)” 部分，也有关于程序倍率的说明。

1.3.5.2Jog--OV--Steps(HOV)

手动倍率值通常也以1%的增量改变.项目“Jog--OV--Stepson/off”使手动快速的改变为100、75、50、30、10、30、3、1%。



使用手动倍率状态键右面的“+/-”键设置倍率。



详细说明请参考Operating Handbook的[Manual traversing of the robot],的“Jog override (manual traversing velocity)”。

1.3.5.3 鼠标的位置

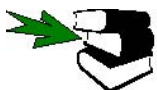
如果使用鼠标移动机器人在机器人坐标系里，操作者能获悉他所处位置的控制器。



关于鼠标定位的详细说明请参考Operating Hand book的[Manual traversing of the robot]章的“Moving the robotwith the Space Mouse”。

1.3.5.4鼠标配置

这儿你能够鼠标的灵活度和它控制的轴



关于鼠标配置的详细说明请参考Operating Hand book, 的[Manual traversing of the robot]章的“Moving the robotwith the Space Mouse” 部分。

1.3.6用户组

为了增加系统的安全性，对确定的用户限制机器人的功能和编程，由用户级别限制存取这些功能。使用密码进行保护。

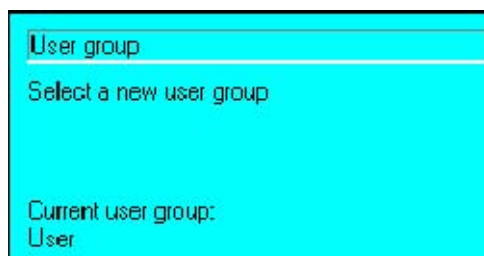
确省时，KRC控制器的软件对于一般用户和专家用户有着区别。用户借助于菜单编程时，不要求确认语法。当系统装入时用户级别自动设置到确省值。

如果用户级别的功能无效，可以转换到专家用户级别。专家用户能够使用ASCII键盘，采用KRL语言编程(KUKA 机器人语言)，编辑系统或初始化文件（总线系统）。KRL是一个高级别的PASCAL编程语言，它适合编程复杂的任务。

专家用户级别由密码保护。按菜单“Configure”，打开一个包含有 “User group” 菜单项目,改变到专家用户级别。



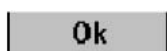
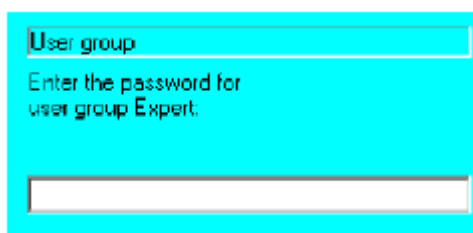
下面的状态窗口被打开



在正常情况下，系统开始时在“User”组，输入规定的密码,能够到达高级的“User group”。



一旦按下用户组有效的软键，状态窗口内容改变，请你为选择的用户组“Expert”输入密码。



为选择的用户组(上部和下部的情况忽略)输入密码，按软键“OK”。



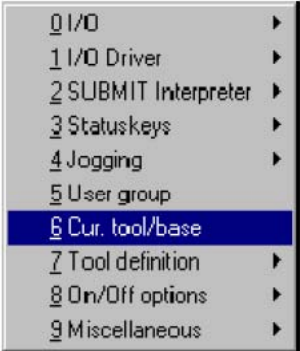
你能够退出这个功能由简单地按软键“Close”,在这种情况下,用户组不改变。
请阅读随后出现在信息窗口的说明。

1.3.7 Cur.tool/base

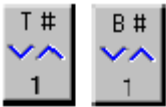
这儿能够选择使用的当前工具、基本系统、外部轴或外部的运动系统,使用工具定义

功能改变分配到个体的名称.

Configure 选择选项 “Cur.tool/base”



下面的窗口打开

A screenshot of a configuration window titled 'current tool/base/ext. Joint'. It contains several input fields and labels: 'Tool no.' with a value of '1' and a label '当前工具[0...16]'; 'Tool name:' with a text box containing 'Gun' and a label '当前工具的名称'; 'Base No.' with a value of '1' and a label '当前基本号:[0...16]'; 'Base system name:' with a text box containing 'Part1' and a label '当前基本系统的名称'; 'Ext. kinematic system' with a value of '0' and a label '当前的运动系统[0...6]'; and 'Name of ext. joint:' with an empty text box and a label '外部轴的名称'. At the bottom, there is a note: 'Select tool, base or ext. kinematics to be activated.'

你能够使用箭头键“↓”和“↑”改变到前一个或下一个输入对话框，写入数字使用数字键或者用右边对应的状态键改变。



只有一个基本系统或外部轴同时有效。

如果输入框的数字是“0”，工具、基本系统或外部的轴视为无效



软键 “OK”用于接受输入的值，如果输入框里数字超过允许的值或者一个没有定义的工具号、外部轴的基本号被选择，对应的错误说明出现在信息窗口。



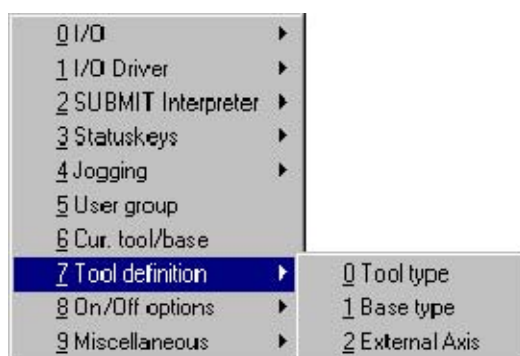
软键“Cancel”用于关闭状态窗口，不存储设置的值。

1.3.8工具定义

这个功能用于分配名称到工具类型、基本类型和外部轴。

打开“Configure”菜单选择子菜单“Tool definition”和相关选项。

Configure



下列的状态窗口“Tool type ”,“Base type”或“ext. Base type ”之一打开,三个不同的项目有效。

Edit

核准的类型的名称能够使用软键“Edit”改变,使用箭头键“↑”、“↓”选择希望的行和按软键“Edit”



工具名称

OK

按软键“OK”保存改变的内容,关闭状态窗口。

Cancel

软键“Cancel”用于关闭状态窗口,不保存改变的内容。

工具类型

Tooltype (TOOL_TYPE)		
#	Type	Name
1	Tool	Weld gun
2	Tool	Torch
3	Tool	Gripper
4	Workpiece	External gun
5	not defined	
6	not defined	
7	not defined	
8	not defined	
9	not defined	
10	not defined	
11	not defined	
12		
13	not defined	
14	not defined	
15	not defined	
16	not defined	

工具名称（24个符号）

工具类型（“Tool ”,“ Workpiece ”和
“not defined”）

工具号[1...16]

- Tool

机器人发兰上的一般工具

- Workpiece
机器人移动工件
- undefined
没有核准的工具类型

基本类型

Basetype (BASE_TYPE)		
#	Type	Name
1	Offset	Part A
2	Offset	Part B
3	Tool	Part C
4	not defined	
5	not defined	
6	not defined	
7	not defined	
8	not defined	
9	not defined	
10	not defined	
11		
12	not defined	
13	not defined	
14	not defined	
15	not defined	
16	not defined	

基本系统的名称（24个符号）

Base type(“Tool”,“Offset” 或 “not
defined”)

基本系统号[1...16]

- Tool

一个外部工具（例如，焊接枪）被安装

- Offset
一个被核准的基本类型
- not defiend
没有定义的基本类型号

ext. Basetype (EX_AX_TYPE)		
#	Type	Name
1	Offset	Turntable
2	Offset	Feeder
3	Tool	Gripper
4	not defined	
5	not defined	
6	not defined	

外部轴或运动系统的名称
(24个符号)

外部基本类型
(“Tool ”,“Offset”或“not
defined”)

外部轴或运动系统的号
[1...6]

- Tool
一个外部工具被安装
- Offset
一个外部轴或运动系统的已经被核准
- not defined
外部的基本号还没有被核准

Configure

设置冷启动和关机等待时间

**1.3.9.1强制冷启动**

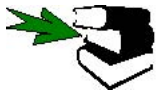
这个菜单对一般和专家级用户都有效。当强制进行一个冷启动、系统装入时，控制器显示Navigate,没有程序被选择，控制器被完全地初始化。



作为确省设置，菜单指令“Force cold start ” 不保留，当要求一个冷启动时，它必须是有效的。

另一方面，在热启动时，控制系统自身执行一个电源断开，以前选择的程序被恢复。系统内部的状态，例如：程序、块指针、变量内容和输出被完全保存。电源会发生故障，例如：供电故障、程序运行期间主开关故障。

如果控制器在启动后检测到一个系统故障或变更的数据，它自动地强制一个冷启动。



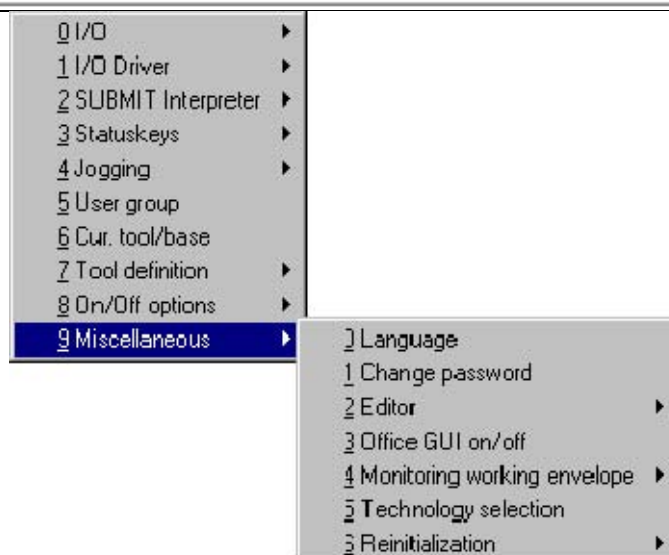
关于电源故障的附加说明在OperatingHand book 的[Operator Control]章,[Running up /shutting Down the controller]节。

1.3.9.2电源不延迟断开

这个指令对专家方式有效，提供给操作者减少预定的延迟时间，直到关断系统。

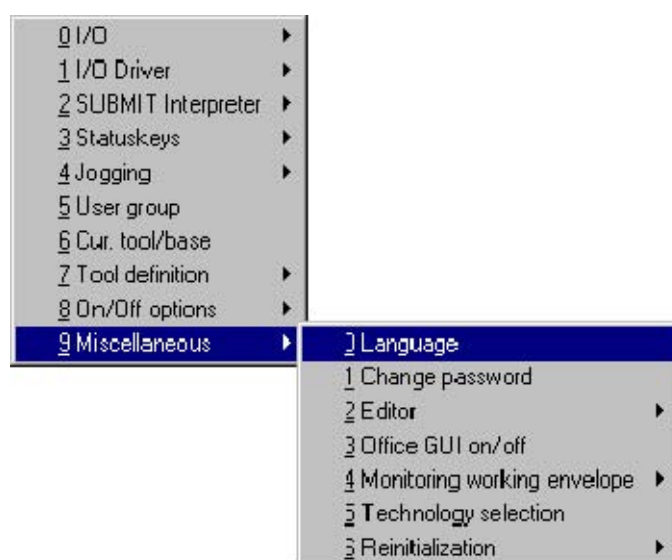
1.3.10 辅助功能**Configure**

在这个菜单下，更多的选项被分组。详细叙述如下

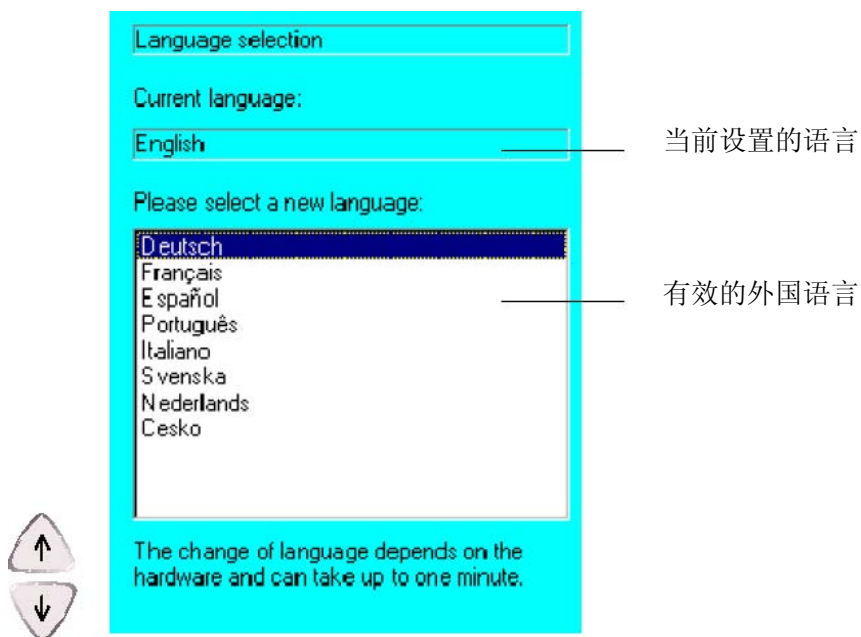


1.3.10.1 语言

这儿你能够设置你的语言到用户界面。

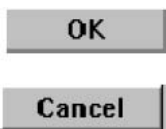


一个状态窗口被打开，从这儿选择要求的语言。



箭头键“↑”和“↓”能用于选择要求的语言
为用户界面选择的语言由安装的Windows语言来决定。

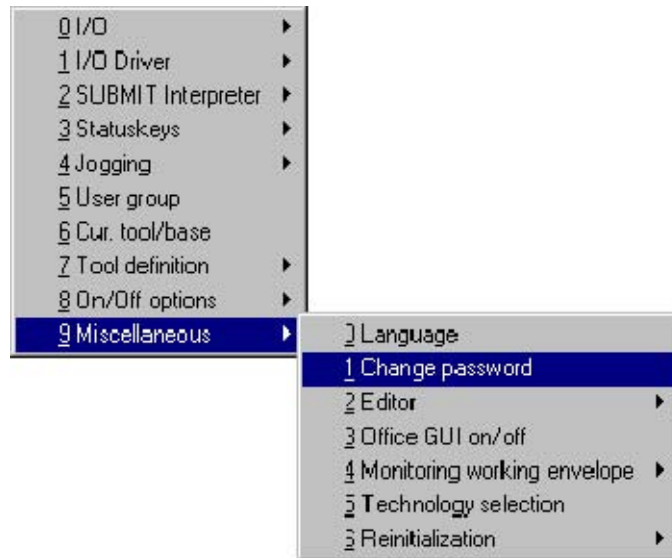
InstalledWindowslanguage	PossibleGUIlanguage
German	German
English	English
French	Dutch
Italian	French
Portuguese	Italian
Spanish	Portuguese
Czech	Swedish
	Spanish
	Czech
Chinese	Chinese
	English
Korean	Korean
	English
Russian	Russian
	English



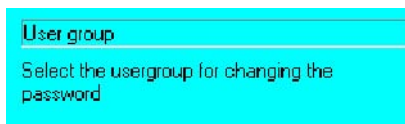
软键“OK”用于接受选择的语言和关闭状态窗口。短时间后，界面语言显示为要求的语言。
软键“Cancel”用于关闭状态窗口，不改变界面语言。

1.3.10.2改变密码

这个项目用于改变用户级别的密码



一旦选择这个项目，一个状态窗口被打开，你能立即按软件键选择你希望改变的密码。



在基本控制系统里，仅提供“User”和“Expert”用户组。

简单地按软键“Close”，你能够退出这个功能而不保存你写入的数据。

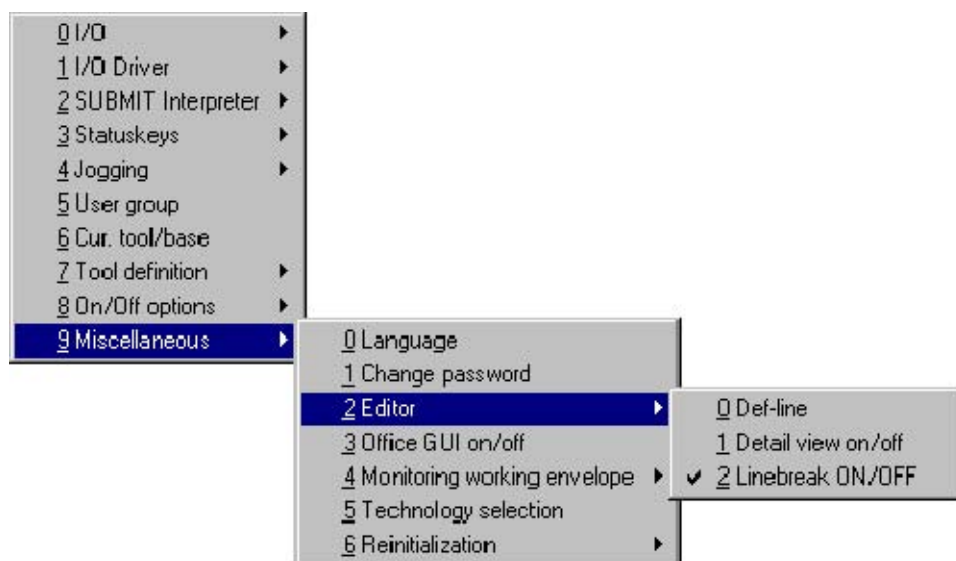


如果你选择了一个用户组，出现另一个状态窗口，写入旧密码和新密码。



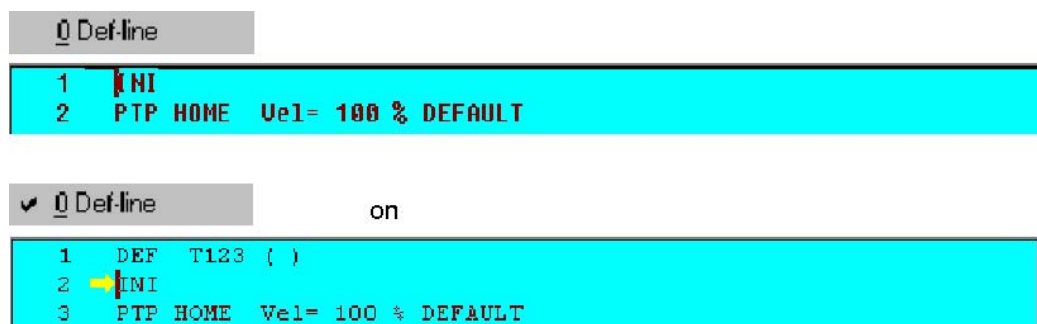
按软键“OK”.密码被改变。

1.3.10.3编辑器



DEF—行

如果这个功能有效，程序里显示DEF行。正常情况下，DEF行被隐藏。



声明仅能够在DEF行可见时实现。这个功能低于用户级别“Expert”，确省设置时无效。在操作者进行一个重新启动或返回到“User”方式时，这个功能自动无效。

详细叙述的打开/关闭(Limited Visibility)

这个功能仅在Expert方式有效，它的另一个作用是尽可能少保留在用户界面的信息量。在确省时“Detail view”无效。

如果功能“Detail view”无效，所有写在“;%”后面的文本记录在FOLD行上，例如，被禁止。这个信息需要时，显示在线形式。

1 Detail view on/off ☐ off

```

1  INI
2  PTP HOME  Vel= 100 % DEFAULT
3
4
5  PTP HOME  Vel= 100 % DEFAULT

```

当这个功能打开时，显示更多的信息，正常时该信息隐藏。

✓ 1 Detail view on/off ☒

```

1  DEF Prog_01( )
2  ;FOLD INI;%{PE}%U3.2.0,%MKUKATPBASIS,%CINIT,%UCOMMON,%P
3  ;FOLD PTP HOME  Vel= 100 % DEFAULT;%{PE}%U3.2.0,%MKUKATPBASIS,
4  ;%MOVE,%UPTP,%P 1:PTP, 2:HOME, 3:, 5:100, 7:DEFAULT
5
6  ;FOLD PTP HOME  Vel= 100 % DEFAULT;%{PE}%U3.2.0,%MKUKATPBASIS,
7  ;%MOVE,%UPTP,%P 1:PTP, 2:HOME, 3:, 5:100, 7:DEFAULT
8  END

```



低于用户级别“Expert”时，确省的设置是无效。



当所有的FOLD打开和“Detail view on/off”关闭时，编程者才可以存取所有的有效行。用户界面上的显示与一般文本编辑器上的显示一致。

```

1  DEF Prog_01( )
2  ;FOLD INI;%{PE}%U3.2.0,%MKUKATPBASIS,%CINIT,%UCOMMON,%P
3  ;FOLD BAS INI;%{E}%U3.2.0,%MKUKATPBASIS,%CINIT,%UINIT,%P
4  GLOBAL INTERRUPT DECL 3 WHEN $STOPNESS==TRUE DO IR_STOPM ( )
5  INTERRUPT ON 3
6  BAS (#INITMOV,0 )
7  ;ENDFOLD (BAS INI)
8  ;FOLD A20 INI;%{E}%U3.2.0,%MKUKATPA20,%CINIT,%UINIT,%P
9  IF ARC20==TRUE THEN
10 A20 (ARC_INI)
11 INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20(TECH_STOP2)
12 ENDIF
13 ;ENDFOLD (A20 INI)
14 ;FOLD A10 INI;%{E}%U3.2.0,%MKUKATPARC,%CINIT,%UINIT,%P
15 IF A10_OPTION==#ACTIVE THEN
16 INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
17 INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
18 INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPU_ARC)
19 INTERRUPT ON 5

```

KRC:\R1\PROGRAM\PROG_01.SRC Ln 1, Col 0

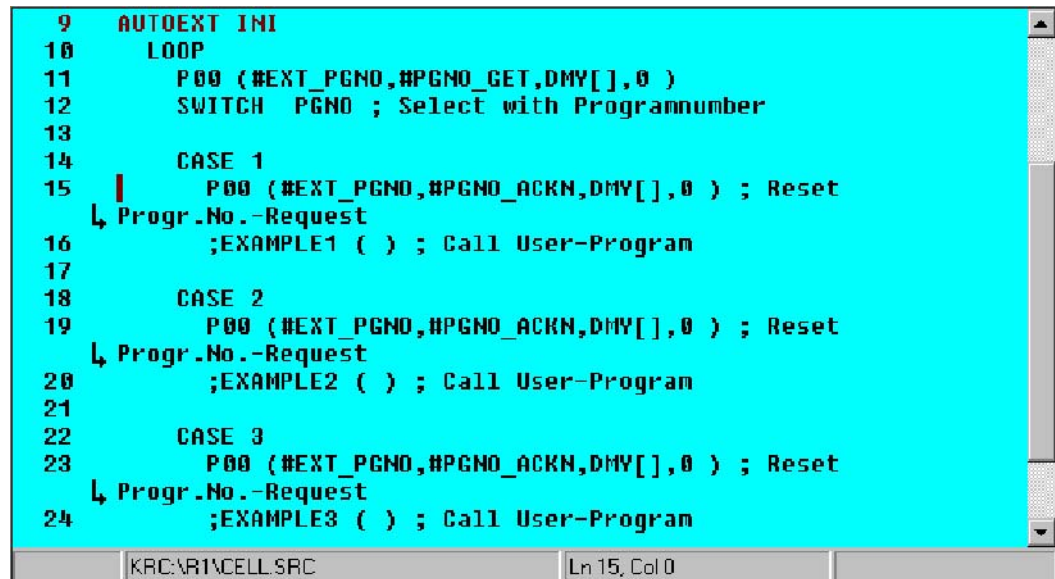
换行符号ON/OFF

在程序窗口打开或关闭换行符号。

...on

换行符号确省时有效，所有的信息显示在程序窗口。当行太长时，在适当的点换行。

✓ 2 Linebreak ON/OFF



```

9  AUTOEXT INI
10  LOOP
11    P00 (#EXT_PGNO,#PGNO_GET,DMV[],0 )
12    SWITCH PGNO ; Select with Programmnumber
13
14    CASE 1
15      P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 ) ; Reset
      ↳ Progr.No.-Request
16      ;EXAMPLE1 ( ) ; Call User-Program
17
18    CASE 2
19      P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 ) ; Reset
      ↳ Progr.No.-Request
20      ;EXAMPLE2 ( ) ; Call User-Program
21
22    CASE 3
23      P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 ) ; Reset
      ↳ Progr.No.-Request
24      ;EXAMPLE3 ( ) ; Call User-Program

```

KRC:\R1\CELL SRC Ln 15, Col 0

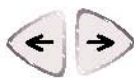
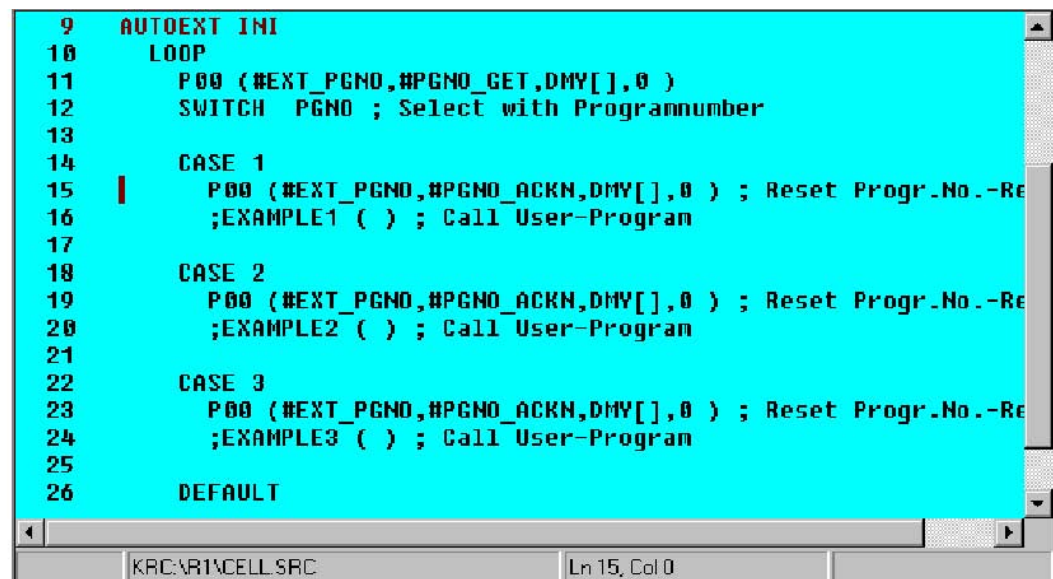


换行后的部分没有行号，用一个箭头代替。

...off

如果换行功能无效，行太长不适合在程序窗口时，程序窗口的内容使用箭头键“←”和“→”移走。

2 Linebreak ON/OFF

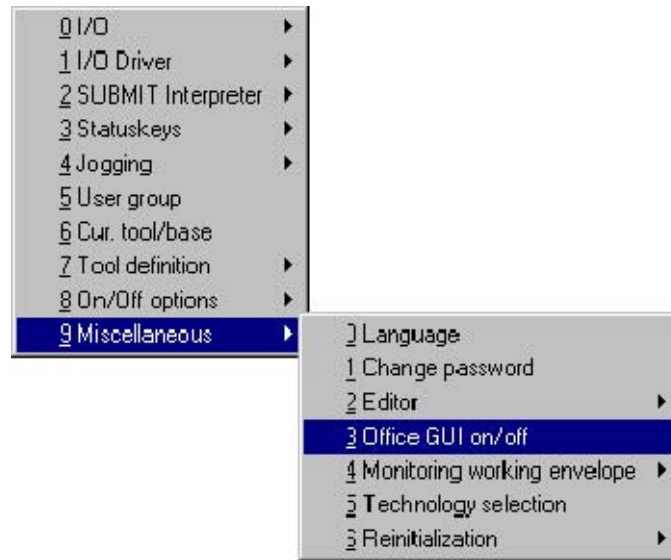
```

9  AUTOEXT INI
10  LOOP
11    P00 (#EXT_PGNO,#PGNO_GET,DMV[],0 )
12    SWITCH PGNO ; Select with Programmnumber
13
14    CASE 1
15      P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 ) ; Reset Progr.No.-Re
16      ;EXAMPLE1 ( ) ; Call User-Program
17
18    CASE 2
19      P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 ) ; Reset Progr.No.-Re
20      ;EXAMPLE2 ( ) ; Call User-Program
21
22    CASE 3
23      P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 ) ; Reset Progr.No.-Re
24      ;EXAMPLE3 ( ) ; Call User-Program
25
26  DEFAULT

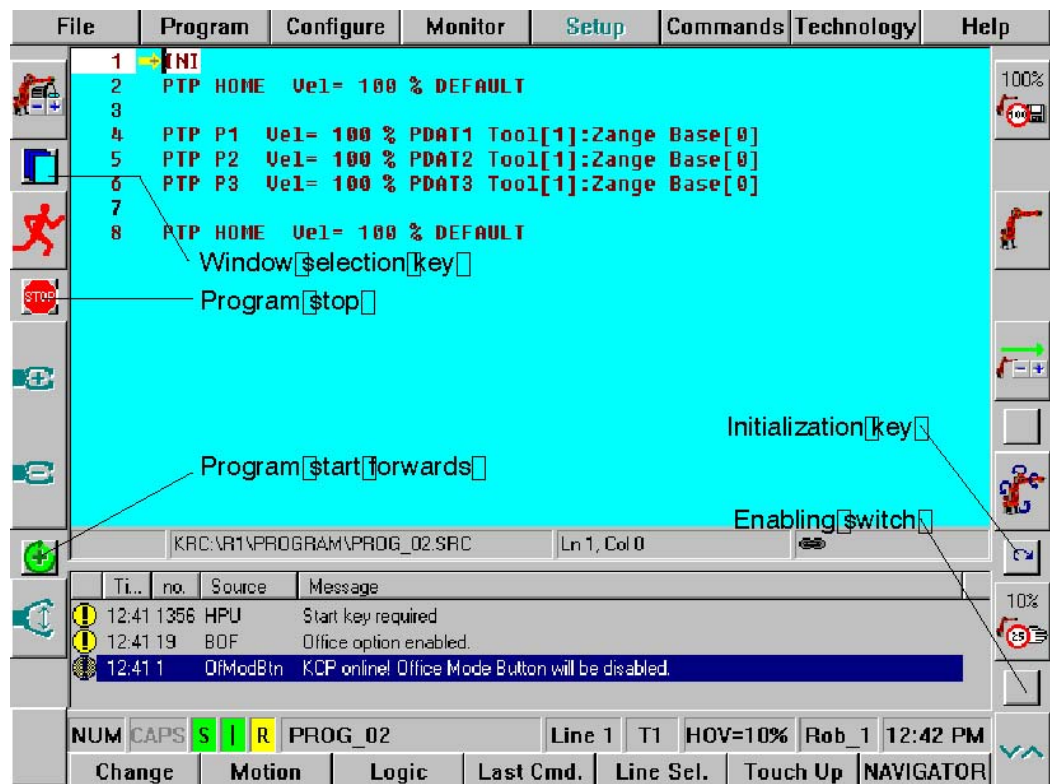
```

KRC:\R1\CELL SRC Ln 15, Col 0

这个子菜单显示kcp操作控制组件“Window selection”键,“Stop”,“Program start forwards”,初始花键和使能kcp显示开关。



当选择这个菜单指令时,“Office GUI”有效,相应的信息显示信息窗口。



符号有下列意义:



用左鼠标击打这个符号时，在程序、状态、信息窗口之间转换。



如果鼠标定位在上边这个符号，用鼠标击打这个符号时，停止程序的执行。



击打这个符号时，引起KUKA技术软件包、资料表和*.OCX文件初始化。



用户能够使用这个按钮模拟使能开关的处理。

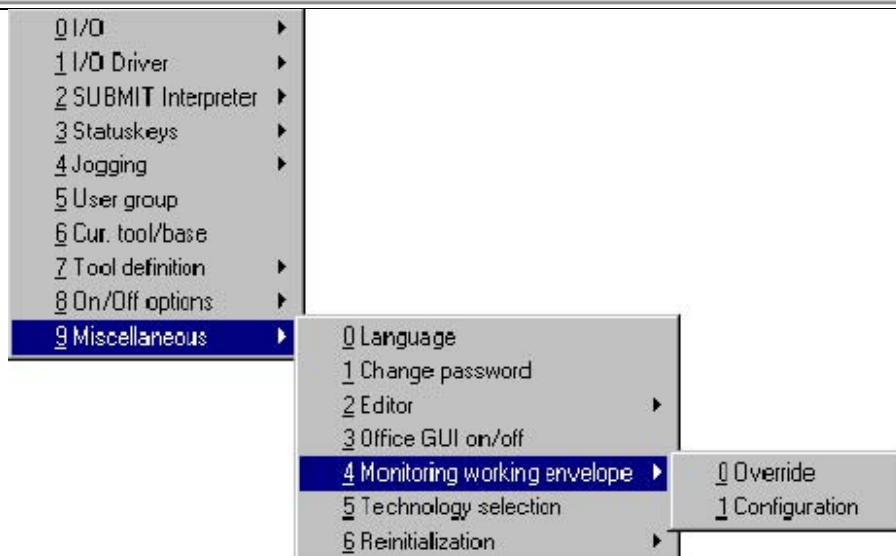


如果这个指令再次选择，选项被关闭，在信息窗口里再显示一个信息。

	Time	no.	Source	Message
	12:06:20	BOF		Office option disabled.

1.3.10.5 工作区域监测

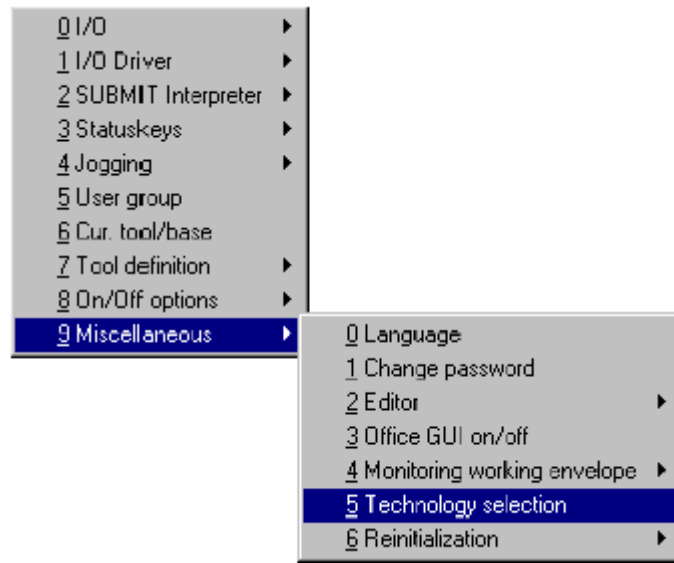
能够断开工作防护检测功能，例如，从超越防护处返回。



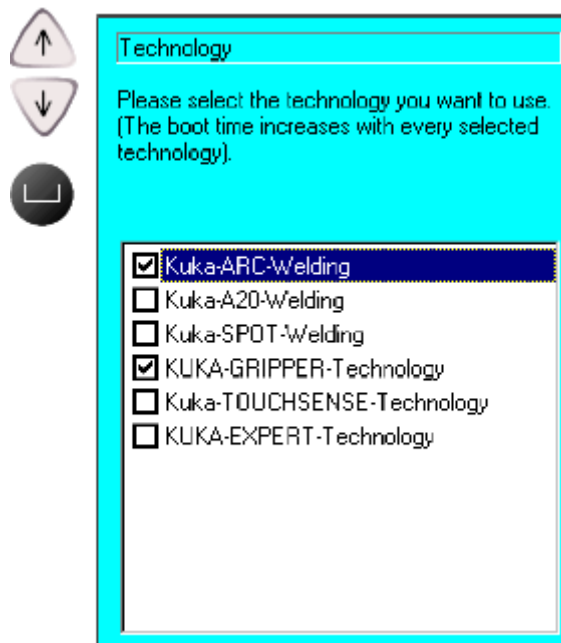
进一步的关于工作防护的信息请参考[Configuring the system ,Expert],section[Work envelope monitoring].

1.3.10.6 工艺选择

如果要求的选项有效，附加的工艺软件包被集成进系统的应用软件。菜单条目“Technology selection”用于该目的。



在状态窗口中可以激活或禁止一个或多个选项。使用“”“”选择需要的工艺软件包。通过按空格键激活或禁止具体的软件包。



Mark

软键“Mark”与空格键功能相同，用于激活或禁止具体的软件包。

OK

“OK”将选择的工艺软件包装载进系统，选择的关闭状态窗口



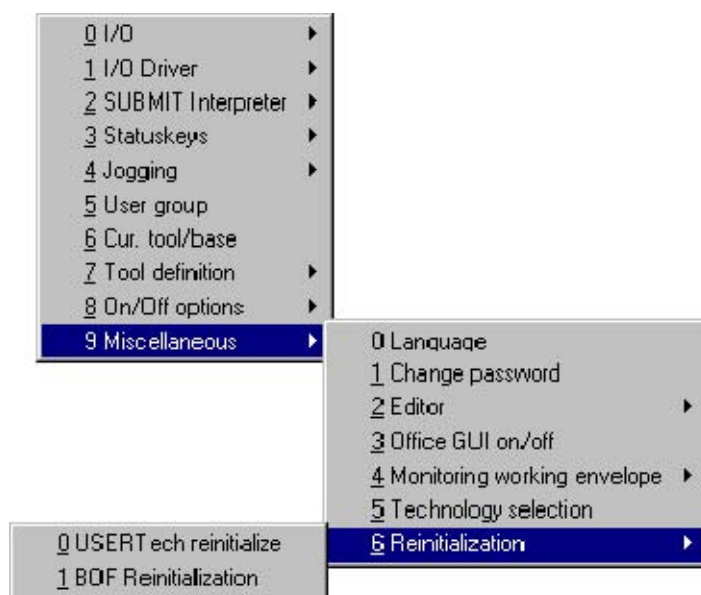
操作处理的过程显示在信息窗口。

	Time	no.	Source	Message
!	12:59:16	BOF	'KUKATPBASIS'	reinitialized
!	12:59:16	BOF	'KUKATPARC'	reinitialized
!	12:59:16	BOF	'KUKATPGRP'	reinitialized
!	12:59:16	BOF	'KUKATPUSER'	reinitialized

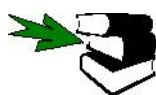
Cancel

软键“Cancel”用于关闭状态窗口而不保存改变的资料。

1.3.10.7 初始化

**USERTechreinitialize**

更新的用户USER Teach 资料能够被读入而不需要重装系统。



详细信息在资料[USERTech].

BOF 初始化

图形用户初始化而不重装系统。初始化的过程显示在信息窗口。

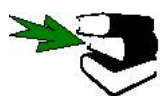
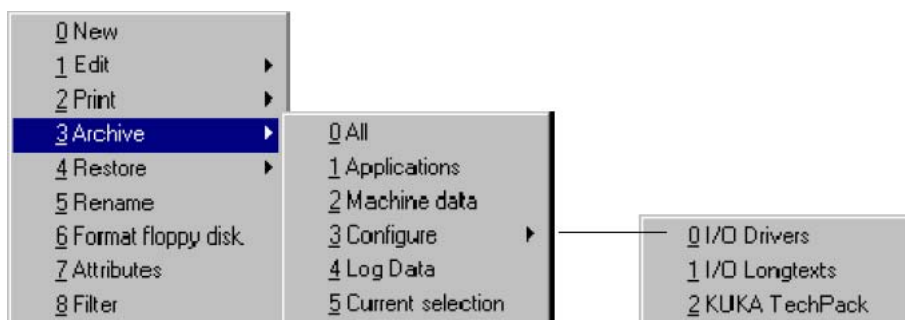


1.4 “File”菜单

这个功能用于备份一定的资料到软盘或从软盘上读取数据到系统上。

1.4.1 归档

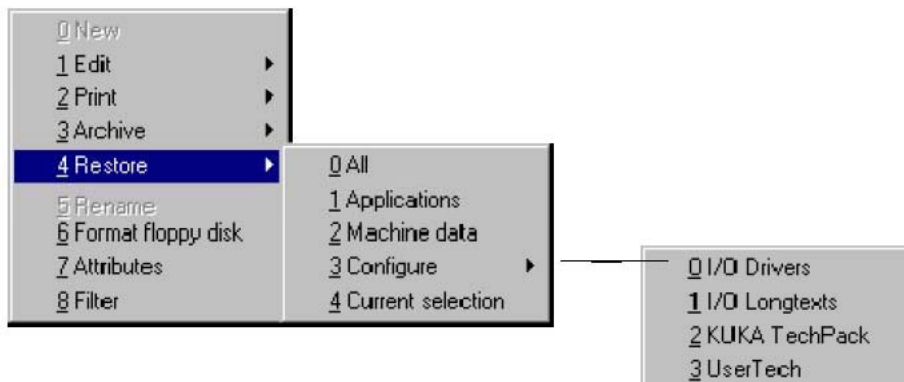
File 按菜单键“File”选择子菜单“Archive”，你能够选择下列项目。



详细信息见Operating Handbook的资料[Operator Control],章[Navigator],节[“File”menu]下的“Archive”。

1.4.2 恢复

File 按菜单键“File”，选择子菜单“Restore”

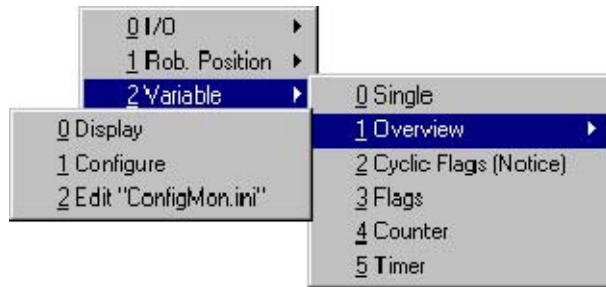


详细信息见Operating Handbook的资料[Operator Control],章[Navigator],节[“File”menu]下的“Restore”

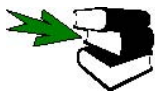
2 配置系统，专家级别

2.1 变量综述

菜单指令“Configure”和“Edit“ ConfigMon.ini””，能够同时显示多个变量。



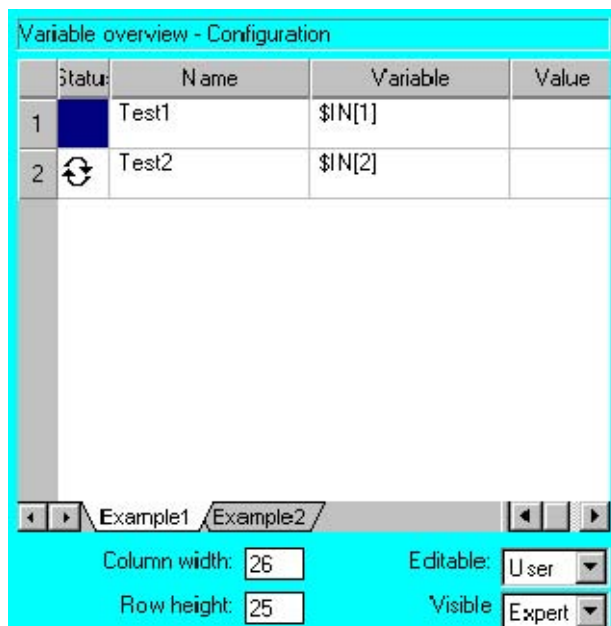
2.1.1 显示



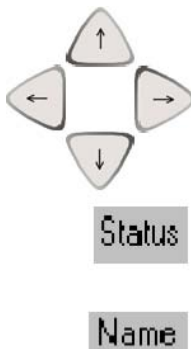
关于变量显示的功能综述在OperatingHandbook的[OperatorControl]章[Monitor].

2.1.2 配置

当选择这个命令时，显示对应的状态窗口显示。在这儿，变量被分组。



箭头键用于选择具体的元素和进行变更。



状态指示出是否一个变量连续变化（Setinfo），每组最大12个变量能够连续检测。

一个变量的描述，容易识别。

Variable 由系统使用的变量名称。

Value 选择一个菜单命令“Variable”-->“Overview”-->“Display”,变量的值在这儿。

Example1 指示出当前最大10个组显示在这儿。

Column width 使用数字键写入要求的行和列的宽度。



Test1 Row height: 25

Column width: 26



行和列的宽度也能由连接到系统的PC鼠标来改变，移动鼠标指向行和列的标题上，按住鼠标左键移动到要求的位置。

Editable

Editable: User
 Visible: Expert

在这个用户组或更高的级别里进行改变。

变量组能够被显示在这个用户组或更高的级别。

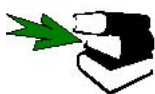
箭头键用于打开对应的选择菜单，选择不同的用户组。

Tab +

软键“Tab+”用于选择下一个用户组。

Monitor

命令“Monitor”调出一个变量综述。



关于变量显示综述的详细说明在资料[OPERATOR CONTROL]的[MONITOR]。

Jump

软键“Jump”移动状态窗口的游标到下一个有效输入框。

Insert

插入组

G. before

在当前组的前面插入一个新组。

G. after

在当前组的后面插入一个新组。

Variable overview - Configuration

Please type in a name for the new group.

Groupname:

Editable:

Visible:

在这儿显示组的名称(最大25个符号)

在这个用户组和高于这个级别的用户组的变量组能够被显示和编辑

Jump

光标定位在下一个元素上。

OK

保存记录按“OK”。

Cancel

不保存记录按“Cancel”。



最大10个组

插入行

R. above

在当前行的前面插入一个新行。

R. below

在当前行的后面插入一个新行。

在新行里能进入要求的数值。移动光标到到要求的对话框里，按Enter键，在选择的对话框里写入要求的值（名称、变量等）

	Name	Variable
1	VarTest	

Cancel

如果“Cancel”被按下，不创建新的行或组。



每组最大25行。

Delete

项目“Delete”允许你删除具体的行或整个组。一旦按下这个键，下列选择有效。:

Row 删除选择的行。

Group 删除当前的组。

Cancel “Cancel”用于取消改变。

Close 软键“Close”存储所做的改变，关闭状态窗口。

Cancel 使用软键“Cancel”结束当前的处理，退出当前的窗口。

2.1.3编辑“ConfigMon.ini”

这个命令用于装入文件“ConfigMon.ini”进入编辑器进行编辑。



```

1  |-----|
2  |-----|
3  ; KUKA Roboter GmbH
4  ; Feb 28, 2000
5  ; Configuration-Monitor settings for (U)KR C1
6  |-----|
7  [Version]
8  Version=2.0.0
9
10 [Group1]
11 GroupTitle=Example1
12 Editable=Expert
13 Visible=User
14 ColWidthState=26
15 ColWidthName=100
  
```

C:\KRC\ROBOTER\INIT\CONFIGMON.INI Ln 1, Col 0

Close 按软键“Close”存储当前的改变。

NAVIGATOR 这个软键使Navigator到背景编辑。



如果“ConfigMon.ini”包含无效的记录，尝试装入时系统自动地取消，在这种场合下不出现错误信息。

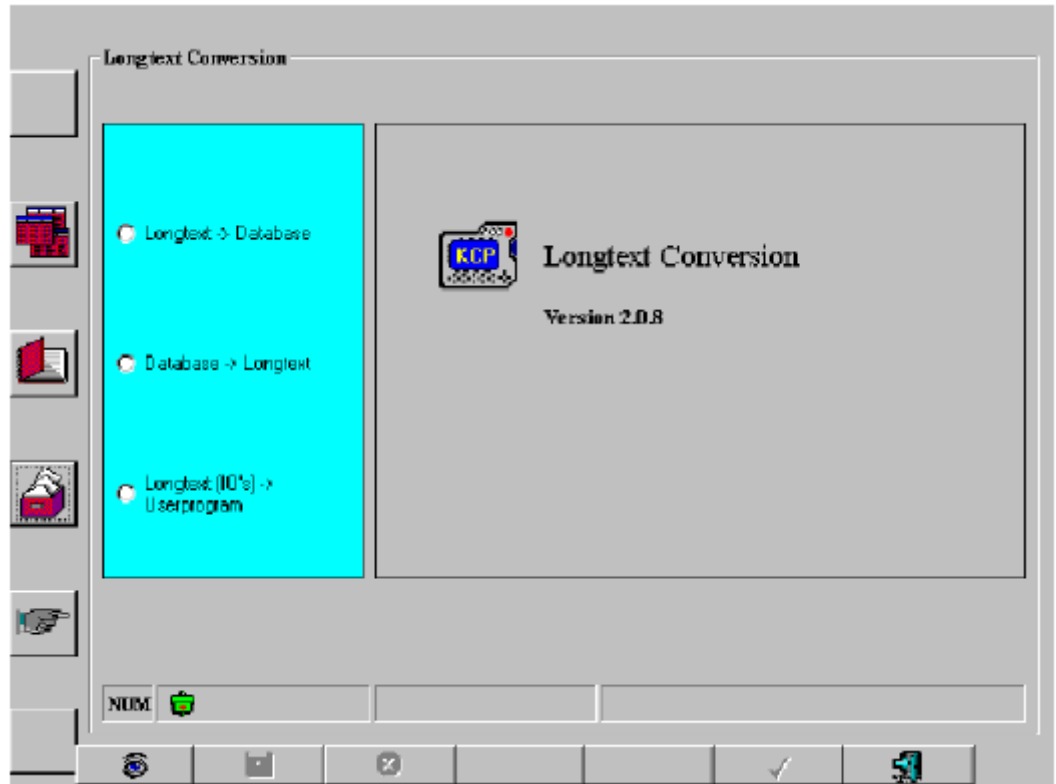
2.2长文本

这个在线程序传送一个分配表到KUKA长文本数据库.使用这种方法，在重新安装时，长的文本不需要为每个机器人手动安装。

长文本显示在状态窗口为输入/输出、循环标志、标志位、计数器、定时器。

Setup

通过菜单键“Setup”和菜单命令“Service”达到这个选项



下列信息显示在状态线上，与长文本转换相关联。

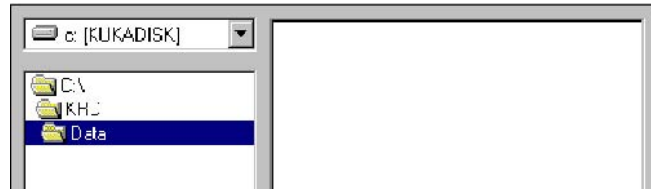
-  联结到数据库有效.
-  正在读数据库.
-  正在写数据库.
-  正在删除数据库
-  已经发生了一个错误



在KCP上的TAB键用于在驱动表、目录表、文本输入框之间转换。“NUM”显示必须无效。



状态键“Long text-->Database”用于装入一个文本文件，传送它到KUKA长文本数据库。



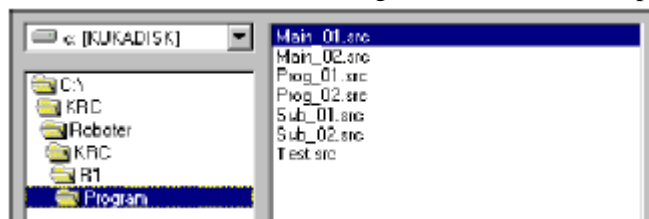
项目“Insert long text”完整放置记录在KUKA长文本数据库里.如果这个项目无效，以前的长文本数据库被删除。



状态键“Database-->Long text”从KUK长文本数据库创建一个文本文件.保存文件的路径和名称能够被规定。



应用程序中，用状态键 “Longtext (IO' s) →Userprogram” 更新长文本



项目“Select all files”适用长文本信息到所有应用程序。



这个软键用于观看和更新KUKA长文本数据库。下列项目是有效的。



这个软键记录KUKA长文本数据库的改变。



改变不保存，关闭数据库显示。



信息记录在KUKA长文本数据库或从它输出。



结束在线程序。

2.3 预热期间降低速度

在机器人起动时，低的环境温度增加齿轮摩擦引起“Regulator limit exceeded Ax”错误信息发生。在这种情况下，一个轴的电机电流到达最大的设定值。为避免这种情况发生，起动时降低速度，电机电流到达一个定义值。

在文件“\$MACHINE.DAT”(path“KRC:\R1\MADA”)里，相应的系统变量能够被变更。

...

BOOL \$WARMUP_RED_VEL = TRUE

REAL \$WARMUP_TIME = 30.0

REAL \$COOLDOWN_TIME = 360.0

INT \$WARMUP_CURR_LIMIT = 99

INT \$WARMUP_MIN_FAC = 60

REAL \$WARMUP_SLEW_RATE = 5.0

...

每个变量的意义如下：

Variable	Range of values	Unit	Meaning
\$WARMUP_RED_VEL	TRUE/FALSE		Warm—up on Warm—up off
\$WARMUP_TIME	Greater than 0	Minutes	Warm—up time
\$COOLDOWN_TIME	Greater than 0	Minutes	Cool--downtime
\$WARMUP_CURR_LIMIT	0...100	Percent	Monitoring value relative to the max. permissible motor current
\$WARMUP_MIN_FAC	0...100	Percent	Factor by which the override command value is reduced
\$WARMUP_SLEW_RATE	Greater than 0	Percent per second	Factor by which the override value is increased

如果数值超过规定的范围，速度不能被降低。

预热时间

当这个功能借助于“\$WARMUP_RED_VEL=TRUE”有效或跟随一个冷启动时，考虑到机器人还没有预热，启动期间，在“自动”或“外部自动”状态，在PTP—PTP和PTP—CP运动情况下，电机电流被检测。如果电机电流越过由变量*\$WARMUP_CURR_LIMIT定义的数值，指令倍率由变量 \$WARMUP_MIN_FAC降低，直到允许的电机电流不在超过。状态信息 “Warm—up active”显示在信息窗口。

当检测不在触发时，倍率被逐渐增加从变量\$WARMUP_SLEW_RATE值到指令设定值。

一旦预热时间超过，速度降低无效，电流不在被检测。

冷却时间

机器人不运动的时间作为冷却时间，冷却时间包括热启动的时间，如果冷却时间超过变量\$COOLDOWN_TIME的数值，机器人不预热，电机电流被重新检测。

记录书文件

特定的时间记录在文件“Warm up. log”(路径“KRC:\Roboter\Log”).

...

Date:	21.08.03	Time:	14:46:57	State:	Monitoring	active
Date:	21.08.03	Time:	14:54:06	State:	Controlling	active
Date:	21.08.03	Time:	14:54:07	State:	Controlling	inactive
Date:	21.08.03	Time:	18:23:43	State:	Monitoring	inactive

...

记录有下列意义:

Monitoring active : 电机电流检测有效

Monitoring inactive : 电机电流检测无效

Controlling active : 速度降低有效

Controlling inactive : 速度降低无效

例子“\$MACHINE.DAT”



```

BOOL$WARMUP_RED_VEL=TRUE
REAL$WARMUP_TIME=30.0
REAL$COOLDOWN_TIME=360.0
INT$WARMUP_CURR_LIMIT=99
INT$WARMUP_MIN_FAC=60
REAL$WARMUP_SLEW_RATE=5.0
  
```

速度降低有效，热启动时间设置为30分钟。如果机器人在静态时间超过360分钟，电机电流检测重新有效，必须重新进行热启动。如果电机电流到达允许最大电机电流的99%，倍率被降低到最大值的60%，如果电机电流下降到低于允许值，倍率由每秒5%逐渐增加。

2.4 自动外部状态下的信息确认

在确省的系统配置里，由PLC定义自动外部状态的信息确认。这就是说，在确定的场

合下，当信息出现时，机器人停止，必须手动重新启动机器人。

是否信息由PLC确认定义在文件“EXT_CONF.INI” (path “C:\Krc\Roboter\Init\”) 中

```

;-----
;--- ext_conf.ini
;-----
...
M_1413 = TRUE ;Axis Ax velocity not programmed
M_1444 = FALSE ;Array index inadmissible
M_1451 = TRUE ;Division by zero
...
PLC

```

具体的内容有下列意义

Entry	Meaning
;	解释
M-xxxx	规定对应的信息行
TRUE	在自动外部状态下能够确认信息
FALSE	在自动外部状态下不能够确认信息



由配置文件“EXT_CONF.INI”进行与系统确省配置相关的系统响应变更。

更新文件“EXT_CONF.INI”，要求一个冷启动。



仅允许确认信息，否则发出对应的错误信息。

下列情况下也出现错误信息：

- 规定了一个不存在的错误信息号；
- 除“TRUE”或“FALSE”以外的值分配到信息行；

在出错的情况下，所有有效的指令被拟制，一个状态信息发生；直到删除了配置错误和进行一个冷启动，机器人才能移动。

2.5 控制柜检测

2.5. 外部风扇

为了增加外部风扇的使用寿命、降低电源消耗和减少噪音，“KR C2”控制器的外部风

扇确省时断开。控制柜的温度被连续检测，如果温度超过一定值外部风扇接通一个预定的周期。



在一定的工作环境下，产生腐蚀性气体，为了预防有害气体进入控制柜，产生一个超过控制柜压力的正压。为此，外部风扇必须连续保持运转。

2.5.2 PC风扇

这个功能检查 “KR C2” 控制柜PC上风扇的速度，速度相对于两个值。

- 如果风扇的速度低于第一个值，显示在窗口的信息不能被确认。
- 如果风扇的速度低于第二个值，发出错误信息，机器人停止。

由在输入行写入变量“\$PC_FANSPEED”，变量更新功能允许你当前的风扇的速度。如果检测功能无效，值“-1”显示，如果正在使用一个服务PC或者没有MFC2出现，值“-2”显示。

2.5.3 硬件报警的可配置输出

在硬件报警或“KRC2”硬件故障的情况下，一个预定义的输出被设置。

这个应用于PC风扇速度检测、电池检测和主板温度检测，如果实际值越过或低于相关定义的值，预定义的输出被设置。

输出设置定义在文件“C:\KRC\Roboter\KRC\Steu\MaDa\\$machine.dat.

SIGNAL\$HW_WARNING\$OUT[48]



由确省“**SIGNAL \$HW_WARNING FALSE**”，这个功能无效。

2.5.4 电机电缆检测

为某些使用“KRC2”控制器的机器人类型，为轴1...3.附加的电机电缆被要求。检测功能有效 (TRUE)，无效 (FALSE) 经由文件“C:\KRC\Roboter\KRC\R1\MaDa\\$machine.dat”。

BOOL \$CABLE2_MON=FALSE

如果功能有效和要求的电缆没有连接，信息窗口的信息不能被确认。



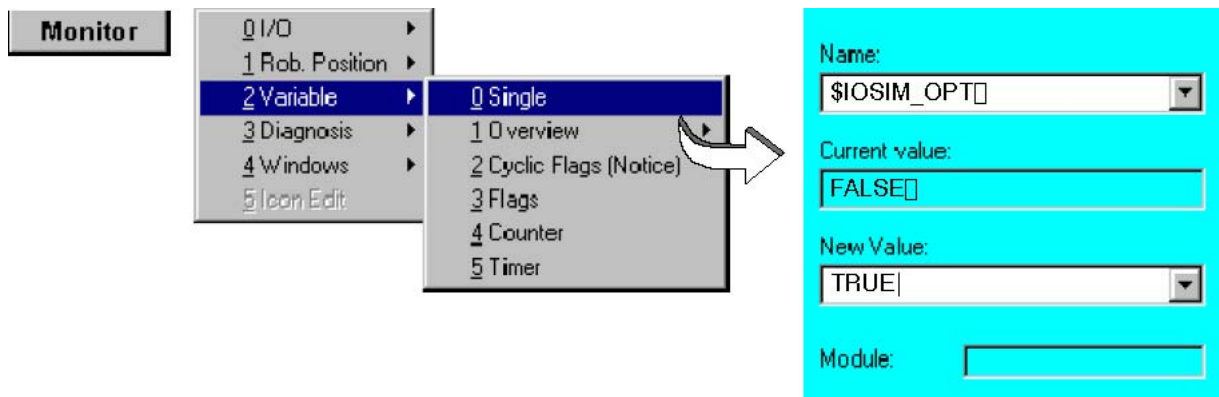
对由丢失附加电机电缆导致的机器人伤害不负责任。

2.6 模拟输入/输出 (I/O模拟)

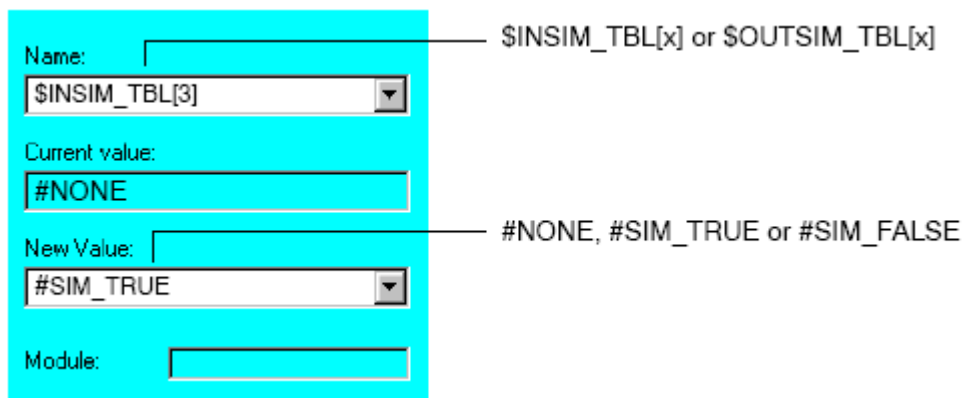
这个功能能够模拟一定的输入和输出.例如, 输入接口设备无效时, 借助于模拟, 输入能够被设置成“TRUE”或“FALSE”, 相同的原则也适用于输出。

2.6.1 功能

借助于变量“\$IOSIM_OPT”, 模拟功能被有效。打开变量改变功能, 改变值到“TRUE”。



模拟功能现在有效, 输入/输出必须被设置。使用变量 “\$INSIM_TBL[x]” 和 “\$OUTSIM_TBL[x]”完成这个功能。



输入和输出能够被分配到下列状态之一

- #NONE 输入和输出不能模拟, 保持不便;
- #SIM_TRUE 输入或输出被设置, 在数字输入/输出显示上能够看见。
- #SIM_FALSE 输入或输出无效, 在数字输入/输出显示上能够看见。



一些和系统相关的输入/输出不能使用, 它们被写保护, 所以不能被变更。

显示排列

变量“\$IOSIM_IN[]”和“\$IOSIM_OUT[]”显示所有的输入/输出在一条线上.输入和输出每个分配一个号，它的意义叙述如下.

Name:

Current value:

New Value:

Module:


\$IOSIM_IN[] or \$IOSIM_OUT[]

显示分配的所有输入输出

输入和输出有下列数值之一：

- 0 没有信号(FALSE)
- 1 有信号(TRUE)
- 2 模拟信号设置为“FALSE”(无效)
- 3 模拟信号设置为“TRUE”(有效)
- 4 系统信号设置为“FALSE”(无效)
- 5 系统信号设置为“TRUE”(有效)

在数字状态输入状态窗口，分配有下列显示：



Input

	SYS	SIM	Input
1			Input
2			Input
3		SIM	Input
4		SIM	Input
5	SYS		Input
6	SYS		Input
7			Input
8			Input
9			Input

在输入 1 无信号 (FALSE)

在输入 2 有信号 (TRUE)

在输入 3 无信号 (FALSE)

在输入 4 有信号 (TRUE)

在输入 5 无信号 (FALSE)

在输入 6 有信号 (TRUE)

如果没有规定序号，变量“\$IN[]”和“\$OUT[]”也显示输入和输出的分配在一条线上，在物理的和模拟的I/O之间没有差别，仅显示是否有信号出现。



Name:	<input type="text" value="\$IN[]"/>	—— \$IN[]或\$OUT[]
Current value:	<input type="text" value="B010100000000000000000000"/>	—— 输入/输出的分配
New Value:	<input type="text"/>	
Module:	<input type="text"/>	

输入和输出有下列状态之一:

- 0 没有信号(FALSE)
- 1 出现信号(TRUE)

2.6.2选项

为了输入/输出的模拟，下面叙述的选项是有用的，不是必须的.

使能开关(\$OUT_NODRIVE)

在输出之间转换，正常情况下必须保持一个使能开关。由设置变量“\$OUT_NODRIVE”到“TRUE”，这个能够被跳跃.

自动外部操作状态(\$IOBLK_EXT)

与\$OUT_NODRIVE=TRUE相关联,变量“\$IOBLK_EXT”允许你在自动外部状态下设置输出,在正常情况下,它是不可能的.由设置变量“\$IOBLK_EXT”到“FALSE”进行这个操作.

2.6.3 使用的变量

变量	值的范围	意义
\$IOSIM_OPT	TRUE FALSE	模拟有效 模拟无效
\$INSIM_TBL[x] x=1...1024	#NONE #SIM_TRUE #SIM_FALSE	输入不能模拟 输入有效 输入无效
\$OUTSIM_TBL[x] x=1...1024	#NONE#SIM_TRUE#SIM_FALSE	输入不能模拟 输入有效 输入无效
\$IOBLK_EXT*1	TRUE FALSE	输出不能设置在#EXT 输出能够设置
\$OUT_NODRIVE	TRUE FALSE	为在输出之间转换,不能按下使能开关 使能开关必须按下
\$IOSIM_IN[]		显示所有的输入
\$IOSIM_OUT[]		显示所有的输出
\$IN[x] x=1...1026 \$IN[]	TRUE FALSE	输入设置 输入不设置 所有的输入
\$OUT[x] x=1...1024 \$OUT[]	TRUE FALSE	输出设置 输出不设置 所有的输出
*1仅与\$OUT_NODRIVE=TRUE相关		



如果一个输出被模拟，模拟期间，在变量调整功能(“\$OUT[x]”)或在状态窗口不能转换，这个输出模拟首先必须无效。

2.7 5个原点位置

从软件版本2.3以后,除了参考点位置以外,用户能够定义另外5个参考点位置,因而共有6个不同的位置有效。

同样的方法,就以前的参考点位置“\$H_POS”而言,当位置到达,变量“\$IN_HOME”设置为“TRUE”时,为了位置“\$AXIS_HOME[1]”...“\$AXIS_HOME[5]”,变量“\$IN_HOME1”...“\$IN_HOME5”被设置为“TRUE”。

同“\$H_POS”一样,定义在变量“\$H_POS_TOL”里的误差范围对5个附加的参考点位置同样有效,如果所有的轴定位在这个误差范围内,对应的变量被设置。



仅在出口类型,对应的设置或变更才能够进行。

2.7.1 “\R1\MaDa\\$machine.dat”文件

轴1...6或外部轴E1...E6的坐标写在“\$AXIS_HOME[x]”显著字体的后面。

```
&PARAM VERSION=3.4.0
&REL 4
DEFDAT $MACHINE PUBLIC
```

```
·
·
·
```

```
E6AXIS $H_POS={ A10.0,A2-90.0,A390.0,A40.0,A50.0,A60.0,E1
0.0,E20.0,E30.0,E40.0,E50.0,E60.0}
```

```
E6AXIS $AXIS_HOME[5]
```

```
$AXIS_HOME[1]={ A10.0,A2-90.0,A390.0,A40.0,A50.0,A60.0,E1
0.0,E20.0,E30.0,E40.0,E50.0,E60.0}
```

```
$AXIS_HOME[2]={ A10.0,A2-90.0,A390.0,A40.0,A50.0,A60.0,E1
0.0,E20.0,E30.0,E40.0,E50.0,E60.0}
```

```
$AXIS_HOME[3]={ A10.0,A2-90.0,A390.0,A40.0,A50.0,A60.0,E1
0.0,E20.0,E30.0,E40.0,E50.0,E60.0}
```

```
$AXIS_HOME[4]={ A10.0,A2-90.0,A390.0,A40.0,A50.0,A60.0,E1
0.0,E20.0,E30.0,E40.0,E50.0,E60.0}
```

```
$AXIS_HOME[5]={ A10.0,A2-90.0,A390.0,A40.0,A50.0,A60.0,E1
0.0,E20.0,E30.0,E40.0,E50.0,E60.0}
```

```
·
·
·
```



确省时,这个文件位于目录“C:\KRC\Roboter\KRC\R1\MaDa”。

2.7.2 “\Steu\MaDa\\$machine.dat”file

在文件“\$machine.dat”里，确定的输出分配到“\$IN_HOME1”...“\$IN_HOME5”的每一个变量.

```
&PARAM VERSION=3.4.0
DEFDAT $MACHINE PUBLIC
.
.
.
SIGNAL $IN_HOME $OUT[1000];机器人在原点位置
.
.
.
SIGNAL $IN_HOME1 $OUT[977]
SIGNAL $IN_HOME2 $OUT[978]
SIGNAL $IN_HOME3 $OUT[979]
SIGNAL $IN_HOME4 $OUT[980]
SIGNAL $IN_HOME5 $OUT[981]
.
.
.
ENDDAT
```



确省时，这个文件位于目录“C:\KRC\Roboter\KRC\Steu\MaDa”.

2.8 工作区域检测

最大有8个立体的或轴规定的工作区域能够被检测.这些工作区域可以重叠形成更复杂的形状, 如果超越一个工作区域, 控制器发出一个预先定义的输出。输出的信号能由KRC程序进一步处理, 或由外部主计算机处理。此时, 机器人停止, 发出一个错误信息。



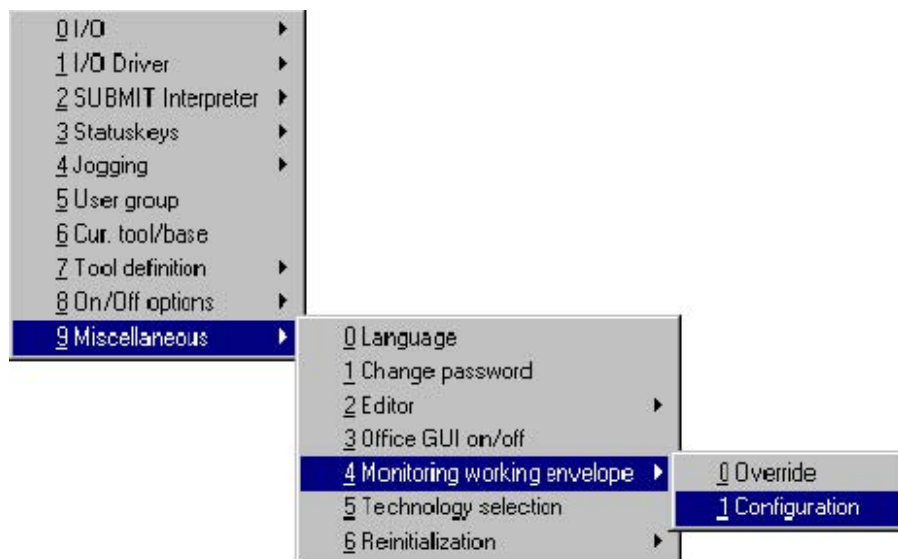
制动距离因机器人的速度而发生变化, 导致机器人停止前超出或进入工作区域, 所以在工作区域范围和终极限之间保留足够的距离是必须的。

2.8.1 笛卡儿坐标工作区域检测

能够定义长方体的工作区域, 取决与具体的定义, 机器人可以不进入或不超出工作区域。

2.8.1.1 配置

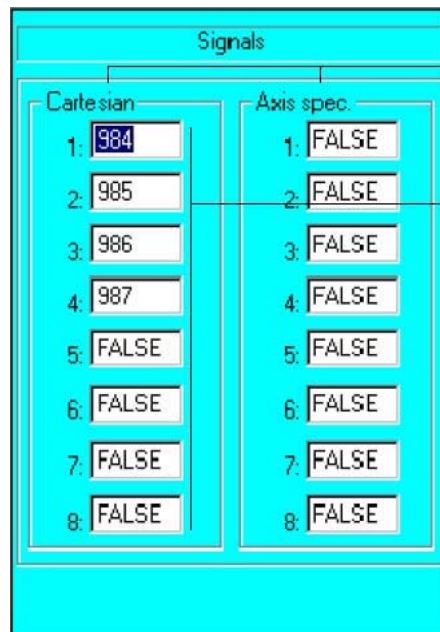
在出口类型, 使用状态窗口, 能够定义长方体的工作区域, 选择菜单指令如下: “Configure”-->“Miscellaneous”-->“Monitoring working envelope”-->“Configuration”。



在相关的状态窗口, 进入必要的参数。

Signal

在状态窗口“Signal”，能为每个工作区域定义一个输出.如果一个工作区域被超过，定义的输出被设置。



窗口的左边部分是笛卡儿坐标系工作区域检测，右边是具体轴的工作区域检测。

设置的输出写入相应的对话框。当超越工作区域时。

如果没有设置输出，对应的信号必须设置为“FALSE”。

声明的KRC信号的详细输出，依次地定义在机床数据里：

```

.
.
SIGNAL $WORKSTATE1 $OUT[984]
SIGNAL $WORKSTATE2 $OUT[985]
SIGNAL $WORKSTATE3 $OUT[986]
SIGNAL $WORKSTATE4 $OUT[987]
SIGNAL $WORKSTATE6 FALSE
SIGNAL $WORKSTATE7 FALSE
SIGNAL $WORKSTATE8 FALSE
.
.

```



如果对应于工作区域的信号已设置为“FALSE”，“\$WORKSPACE[n].STATE”能够用于读取是否工作区域已经被超越。

Cartesian

软键“Cartesian”使你进入立体的工作区域保护的状态窗口。

Cartesian workspaces

No. 1

Name: WORKSPACE 1

Origin

X: 500.00

A: 0.00

Y: 500.00

B: 0.00

Z: 1000.00

C: 0.00

Distance to origin

X1: 100.00

X2: -100.00

Y1: 100.00

Y2: -100.00

Z1: 100.00

Z2: -100.00

Mode:

OUTSIDE_STOP

工作区域的名称或代号

FRAME 定义相对于全局坐标系统的工作区域的起始点和方向

相对于由 {X, Y, Z, A, B, C} 定义的起始点工作区域的尺寸。

工作区域检测状态

对应于KRC变量“\$WORKSPACE[n]”由下面结构:

```
$WORKSPACE[n]={X 500, Y 500, Z 1000, A 0, B 0, C 0, X1 100, Y1 100, Z1 100, X2 -100, Y2 -100, Z2 -100, MODE #OUTSIDE_STOP, STATE FALSE}
```

结构“\$WORKSPACE[n]”的元素的意义:

X,Y,Z:相对于全局坐标系统的工作区域的起始点	
A,B,C: 相对于全局坐标系统的工作区域的方向	
X1,Y1,Z1:确定 Δx_1 , Δy_1 , Δz_1 相对于点X,Y,Z,A,B,C和展开的长方体的尺寸	
X2,Y2,Z2:确定 Δx_2 , Δy_2 , Δz_2 相对于起始点X,Y,Z,A,B,C和长方体的尺寸的增减	
“MODE”设置选项:	
#OFF	工作区域询问检测关闭.
#INSIDE	机床/工件的参考点(TCP)在工作区域的里边, 预定的输出被设置.
#OUTSIDE	机床/工件的参考点(TCP)在工作区域的外边, 预定的输出被设置.
#INSIDE_STOP	如果机床/工件的参考点(TCP)或The wrist root point 在工作区域里边, 预定的输出被设置, 而且机器人被停止, 产生错误信息 1 1 4 “ Work envelope Nr. n violated”.
#OUTSIDE_STOP	如果机床/工件的参考点(TCP)或The wrist root point 在工作区域的外边, 预定的输出被设置, 而且机器人被停止, 产生错误信息 1 1 4 “ Work envelope Nr. n violated”.

“STATE”可能的状态:

TRUE*1 工作区域已经被超越
FALSE*1 工作区域没有被超越
*1值仅能够被显示，不能被改变



在“INSIDE_STOP”或“OUTSIDE_STOP”状态，工作区域被超越，如果关闭工作区域检测，机器人可以恢复运动。

变量修正功能或K R C程序使用的变量

变量	意义	值的允许范围
\$WORKSPACE[n] <i>n=1...8</i>	对应的笛卡儿工作区域的定义	
\$WORKSPACE[n].MODE <i>n=1...8</i>	工作区域检测的类型的定义	#OFF #INSIDE #OUTSIDE #INSIDE_STOP #OUTSIDE_STOP
\$WORKSPACE[n].STATE <i>n=1...8</i>	仅读的变量，指示是否一个工作区域已经被超越， (TRUE)表示超越 (FALSE)表示没有超越	TRUE FALSE
\$WORKSPACE_NAME[n] <i>n=1...8</i>	定义的工作区域的名称	最大可使用 2 4 个符号
\$WBOXDISABLE	工作区域检测倍率功能有效 (TRUE)或无效(FALSE)。	TRUE FALSE



变量“\$WORKSPACE”的变更触发一个预先的运行停止。



在控制器操作期间如何改变系统变量的数值的详细资料请参阅数据 ProgrammingHandbook,在[Monitor]章的“Variables”节。

如果“\$TOOL”无效和至少一个工作区域有效，错误信息 1 1 2 出现在信息窗口：

“Invalid\$TOOL:工作区域监视不可能”

设置的输出被复位，显示的信息被删除。



使用不正确的“\$TOOL”数据导致不能预料的后果！

Axis spec.

按软键“ Axis spec.”，进入具体轴的工作区域定义的状态窗口。



详细请参阅章节2.8.2.

Change

按软键“Change”保存设置.



在*.SRC文件里，工作区域保护也能够定义、接通或断开.规定在这儿的数值自动地进入在文件“\$CUSTOM.DAT“，在下一次控制器接通时有效。
工作区域设置也能够通过更新变量进行改变。

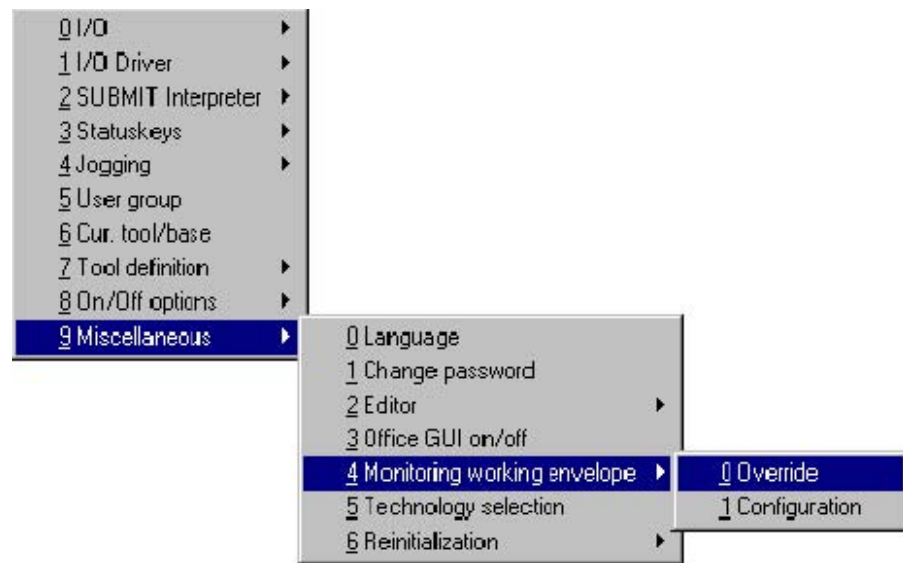
Close

使用软键“Close”关闭配置菜单，除非你使用软键 “Change”，所有的变动不会保存.

经由菜单更新工作区域检测

工作区域检测能够被“overridden”，经由菜单

“Configure”-->“Miscellaneous”-->“Monitoring working envelope”-->“Override”.



这个功能可使机器人移动到工作区域的外边。



这个功能仅在TEST(T1)操作状态.

当工作区域被超越时，报警 1 1 4 出现：

“WorkenvelopeNr.*n*violated”

如果工作区域检测被倍率，这个信息由 1 1

5 代替：“FreeworkenvelopeNr.*n*”

一旦机器人退出超越区域，信息被删除。

经由变量修正功能或使用 K R L 程序取消工作区域检测。

改变与工作区域相关的元素“MODE”的值，例如改变

“\$WORKSPACE[1].MODE”到“#OFF”.

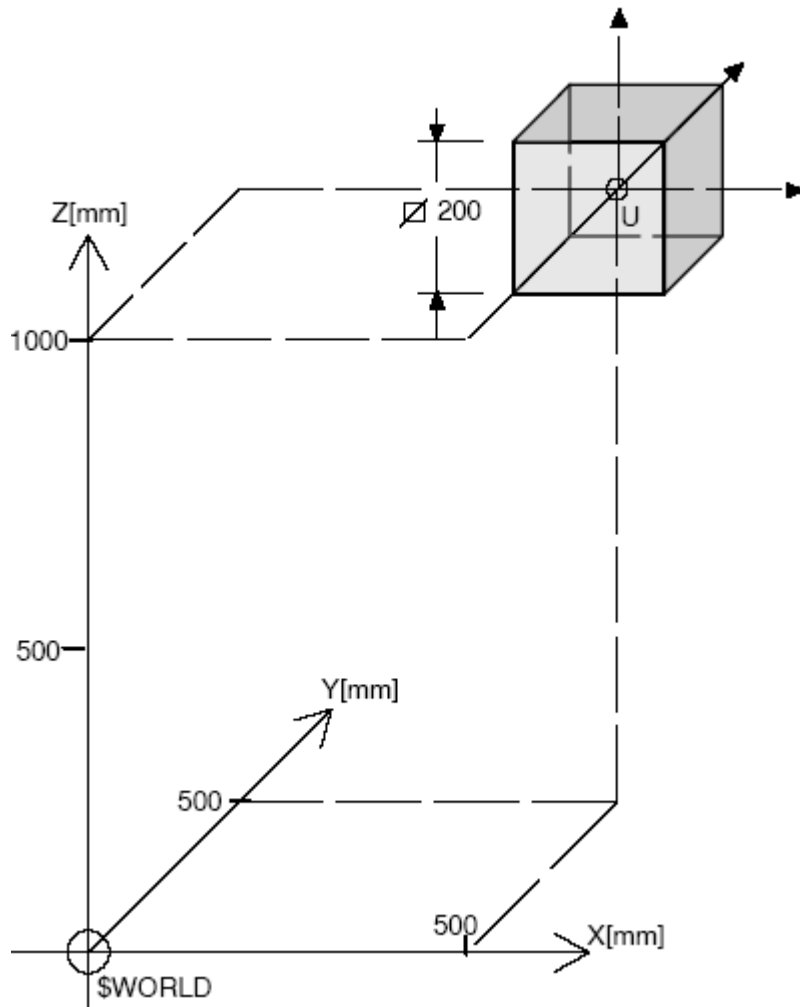


直到元素被重新设为除“#OFF”以外的值，工作区域检测保持保持有效。

2.8.1.3例子



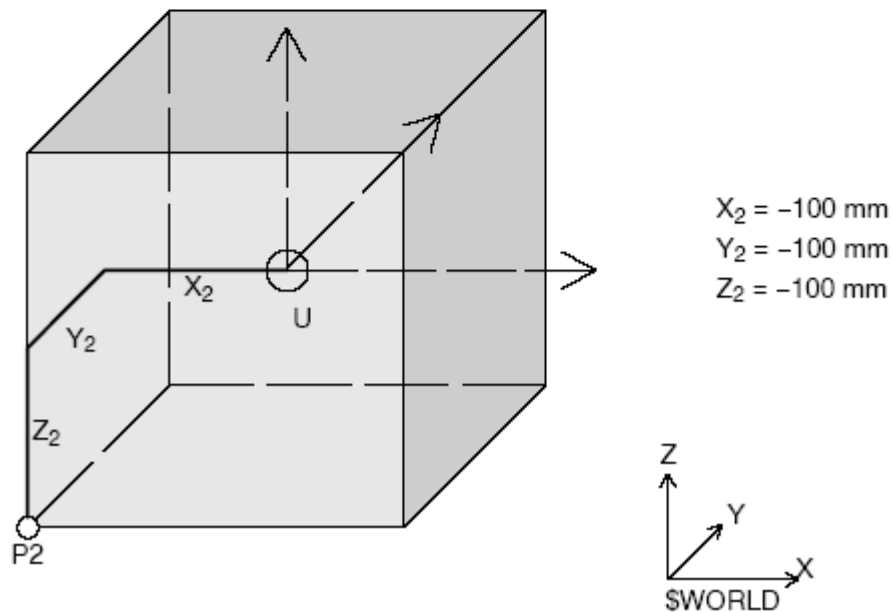
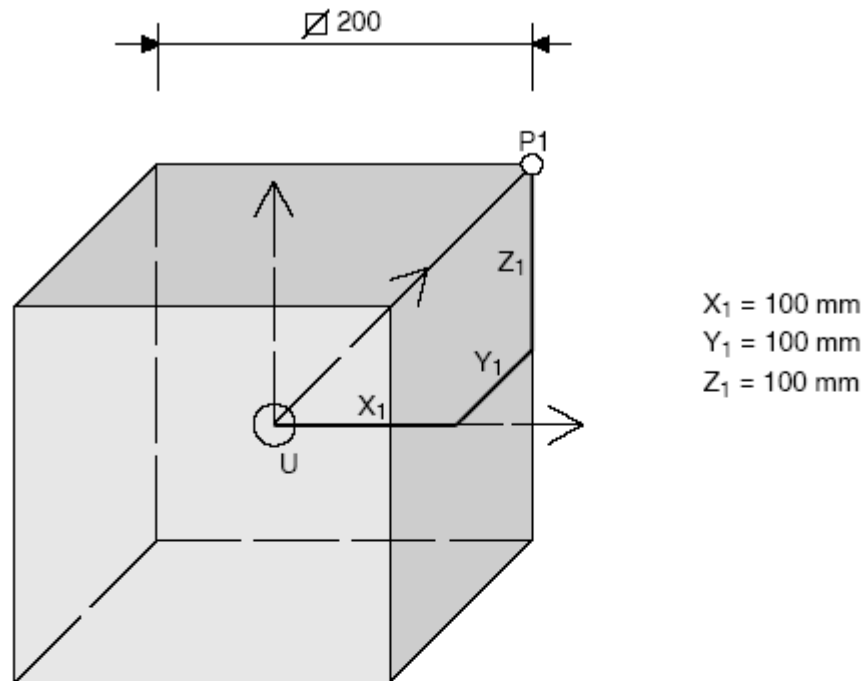
监测一个边长为200mm的立体工作区域，中心点坐标是X=500mm，Y=500mm，Z=1000mm，角度A,B,C的值是0。



工作区域的原点(frame“U”)与全局坐标系相关:

```
$WORKSPACE[n]={X 500, Y 500, Z 1000, A 0, B 0, C 0, X1 100,
Y1 100, Z1 100, X2 -100, Y2 -100, Z2 -100, MODE #INSIDE,
STATE FALSE}
```

工作区域的位置和方向有frame“U”来定义，它的尺寸由原点“U”两边相对的两点来定义。



点“P1”和“P2”在后面的参数行里：

```
$WORKSPACE[1]={X 500, Y 500, Z 1000, A 0, B 0, C 0, X1 100,
Y1 100, Z1 100, X2 -100, Y2 -100, Z2 -100, MODE #INSIDE,
STATE FALSE}
```

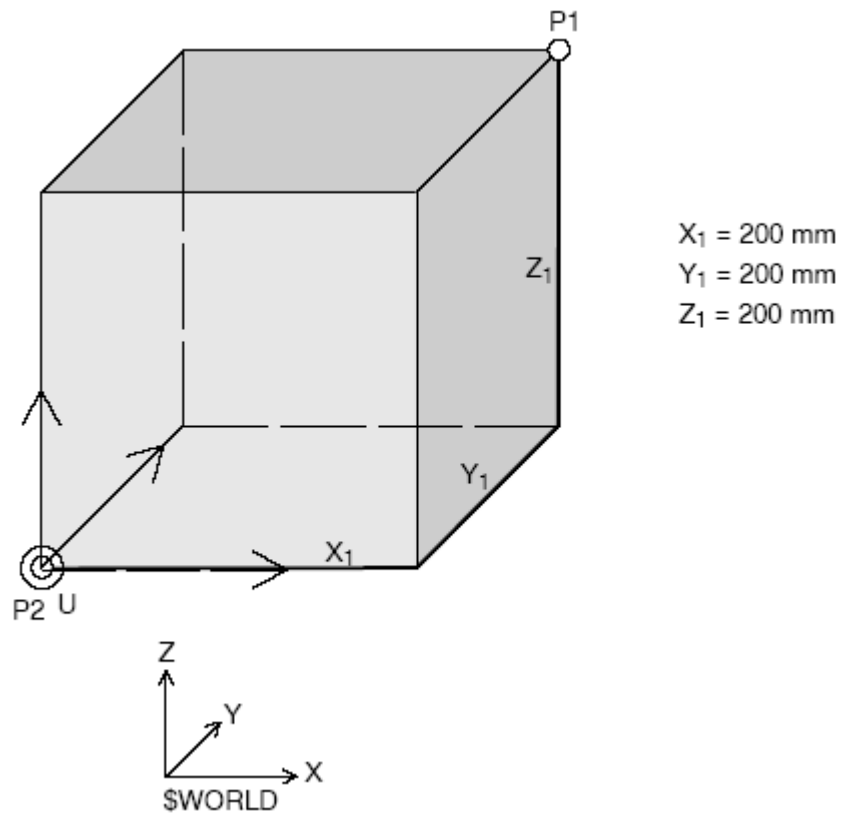
立方体的边长计算公式是 $|X1-X2|$ ， $|Y1-Y2|$ 和 $|Z1-Z2|$ 。一旦刀具或工件的参考点记入工作区域，分配的输出生被设置。在状态窗口的倒数第二个参数或KRC指令用于这个目的。

```
$WORKSPACE[n]={X 500, Y 500, Z 1000, A 0, B 0, C 0, X1 100,
Y1 100, Z1 100, X2 -100, Y2 -100, Z2 -100, MODE #INSIDE,
STATE FALSE}
```



如果点

“P2”位于工作区域的原点，仅 “P1”的坐标需要确定。



如果刀具或工件的参考点是在工作区域的外边,选定的输出被设置.

对应的指令如下:

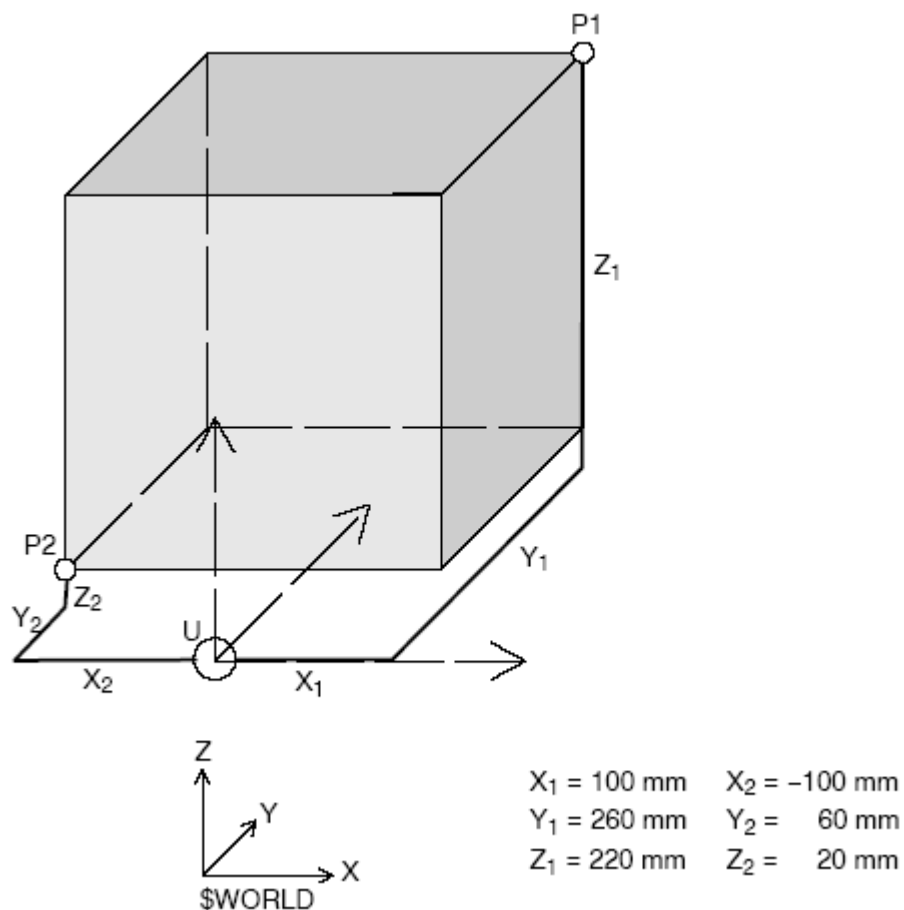
```
$WORKSPACE[2]={X 400, Y 400, Z 900, A 0, B 0, C 0, X1 200, Y1 200, Z1 200, X2 0, Y2 0, Z2 0, MODE #OUTSIDE_STOP, STATE FALSE}
```



这个和先前的例子有相同的工作区域,位置和方向一样,但原点是在立方体的一个角上。



如果坐标点“X1”和“X2”，“Y1”和“Y2”，“Z1”和“Z2”任何一个符号相同，择原点“U”在工作区域的外边。



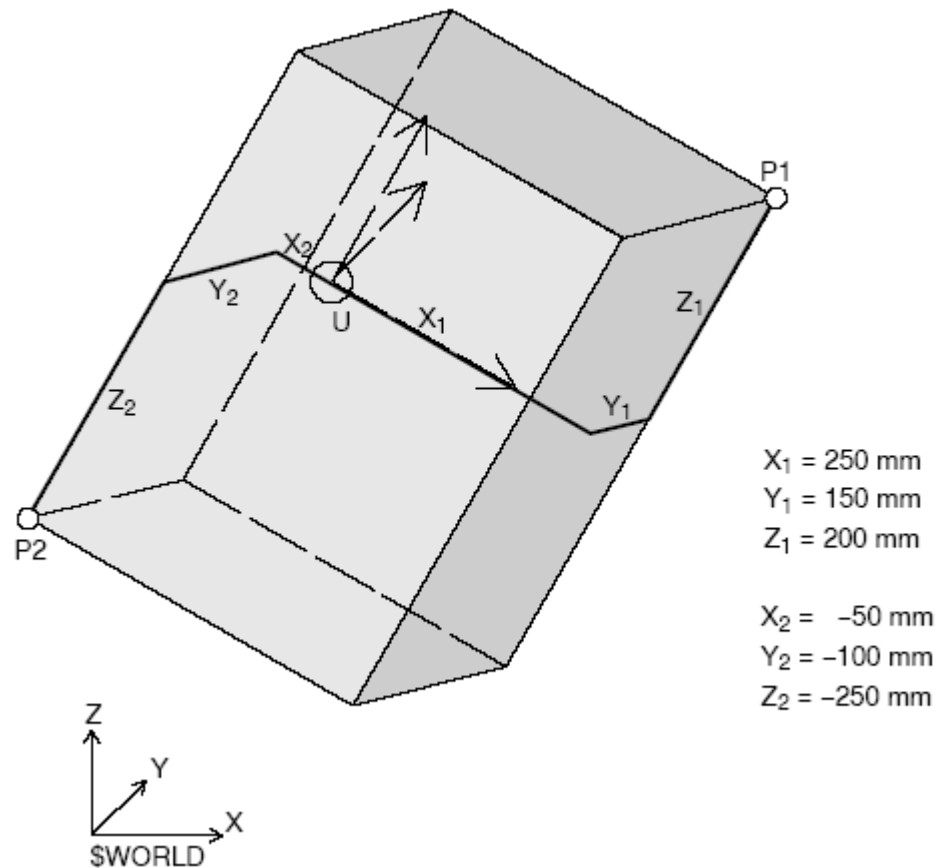
在这儿,刀具或工件的参考点在工作区域的外边,选定的输出被设置.

对应的指令如下:

```
$WORKSPACE[3]={X 500, Y 500, Z 2000, A 0, B 0, C 0, X1 100,
Y1 260, Z1 220, X2 -100, Y2 60, Z2 20, MODE #OUTSIDE, STATE
FALSE}
```



该例中，工作区域的尺寸是 $x=300\text{mm}$ ， $y=250\text{mm}$ ， $z=450\text{mm}$ ，相关于全局坐标系，绕Y轴旋转30度，原点“U”不在长方体的中心。



如果刀具或工件的参考点在工作区域的外边,选定的输出被设置,同时,机器人停止,发出一个错误信息.

对应的指令如下:

```
WORKSPACE[4] = {X 500, Y 500, Z 2000, A 0, B 30, C 0, X1 250,
Y1 150, Z1 200, X2 -50, Y2 -100, Z2 -250, MODE #INSIDE_STOP,
STATE FALSE}
```

2.8.2 轴的工作区域检测

由软限位开关定义的区域可进一步保护机器人、刀具或工件。

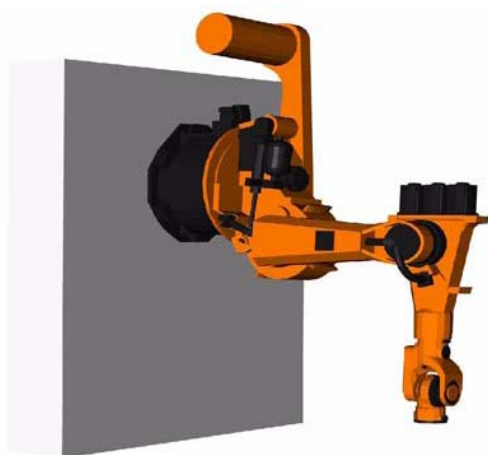
取决于定义，超出或离开工作区域，选定的输出被设置，相对应的，机器人能够停止，显示信息。



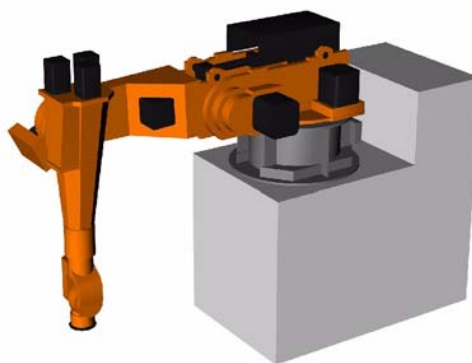
在 J O G 状态，如果越过工作区域，发生斜坡制动；在其他状态，触发动态制动。

2.8.2.1 功能的原则

有一些系统有判断力，它限制机器人的工作区域，进一步保护机器人或周边设备，这个应用于下面的机器人类型：



墙安装类型

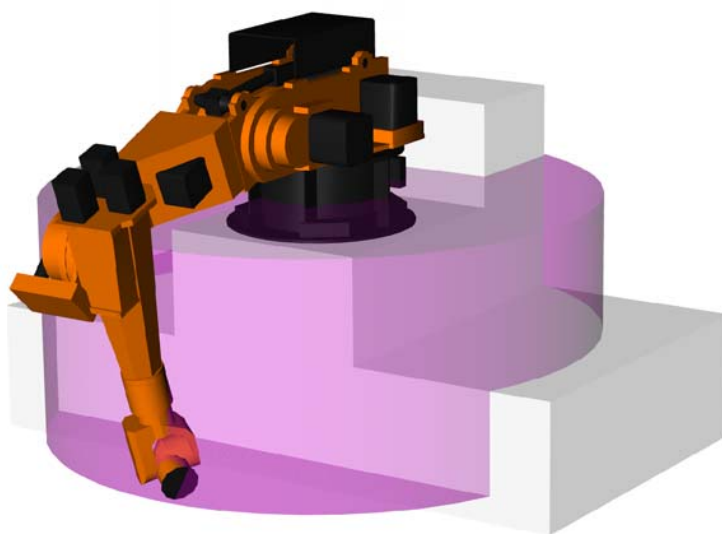


架子安装类型

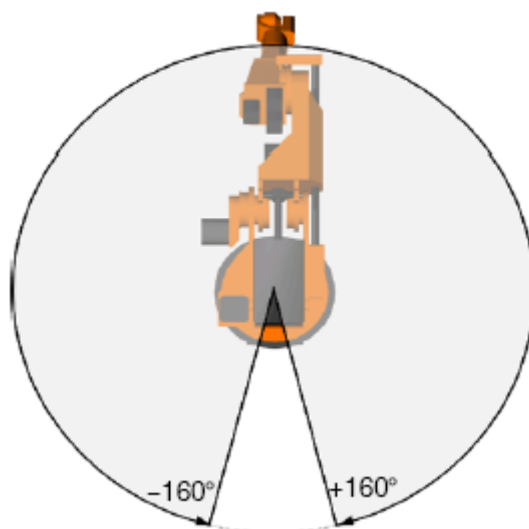


放在拖板上的

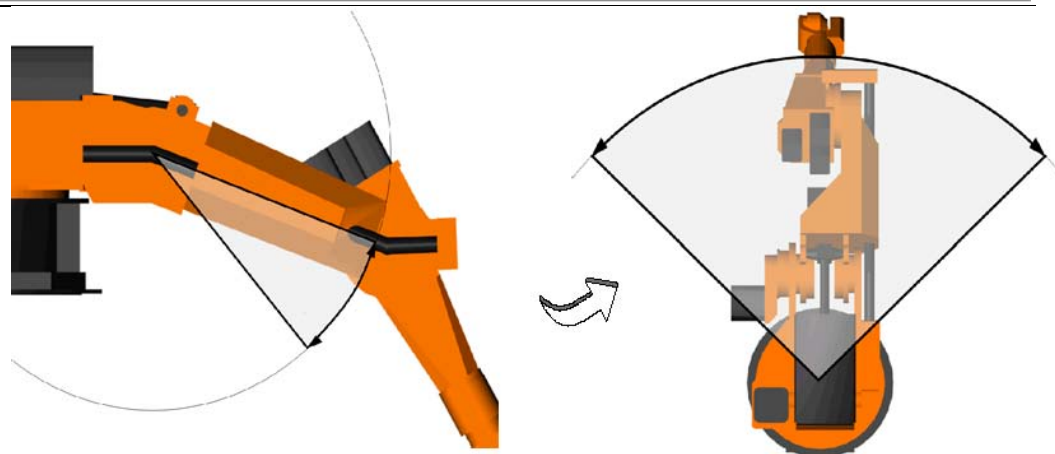
例如，如果有供应电缆，组合式附件，或其他位于工作区域范围的设备，没有对应的工作区域监测，可以损伤这些设备。



机器人最大的活动范围由软限位开关限制，它的范围取决于机器人的类型，在这个例子中，轴 1 最大的移动范围是 $\pm 160^\circ$ 度。



轴工作区域监测能够被限制，例如，限制轴 1 的活动范围大小取决于轴 2 的位置。



轴 2

轴1

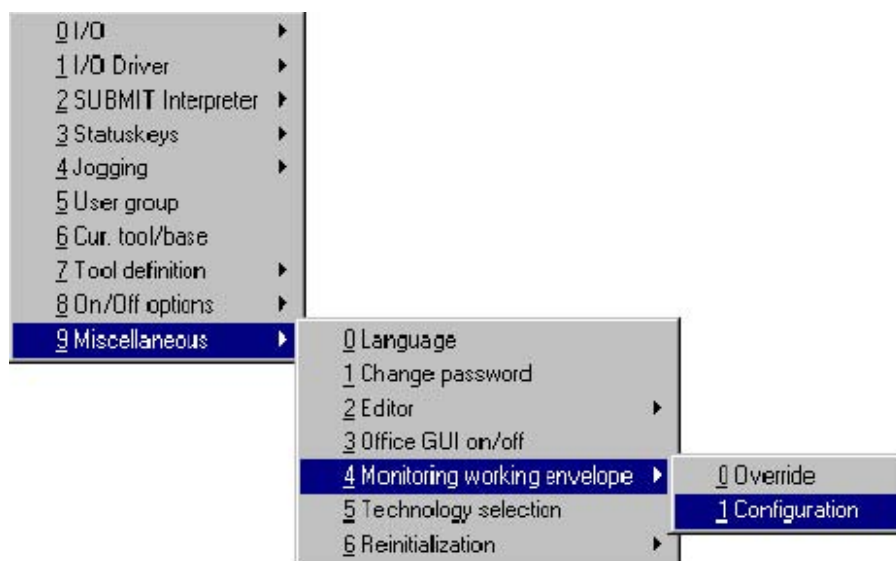
如果轴 2 位于阴影区域,轴 1 不能离开它的阴影区域.
为每个监测的轴,最大和最小的值必须被规定.



对于具体的机床, KUKA 已经将轴规定监测区域定义好。这些值必须不能改变, 否则, 机器人会受到损害。
为避免机器人受到损害, 当进入数值时, 考虑机床和工件。

2.8.2.2配置

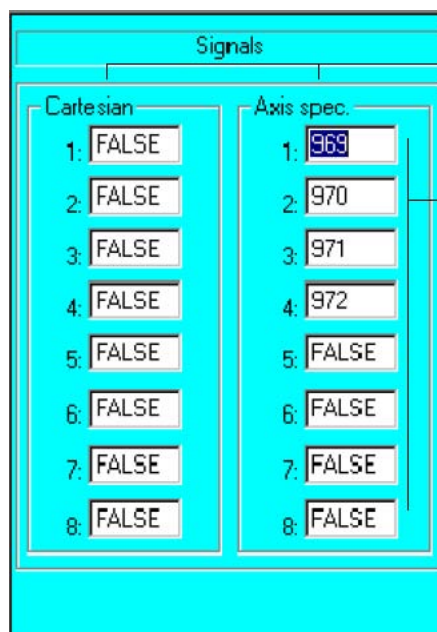
在出口类型，使用一些状态窗口定义工作区域.选择菜单指令“Configure”-->“Miscellaneous”-->“Monitoringworkingenvelope”-->“Configuration”.



在相关的状态窗口进入必要的参数.

在状态窗口“Signal”,能够为每个工作区域定义一个输出.如果工作区域被超越,定义的输出被设置.

Signal



窗口的左边部分是笛卡儿坐标系工作区域监测,右边部分是轴规定工作区域监测.

设置的输出进入在文本框。

当越过工作空间时如果不要设置输出，则对应的信号必须设置为“FALSE”。

定义各个输出的KRL信号声明依次在机床数据中定义:

```

SIGNAL$AXWORKSTATE1$OUT[969]
SIGNAL$AXWORKSTATE2$OUT[970]
SIGNAL$AXWORKSTATE3$OUT[971]
SIGNAL$AXWORKSTATE4$OUT[972]
SIGNAL$AXWORKSTATE5FALSE
SIGNAL$AXWORKSTATE6FALSE
SIGNAL$AXWORKSTATE7FALSE
SIGNAL$AXWORKSTATE8FALSE

```



如果对应于工作区域监测的设置到“FALSE”,元素“\$WORKSPACE[n].STATE”能够用于读取是否工作区域被超越.

Cartesian

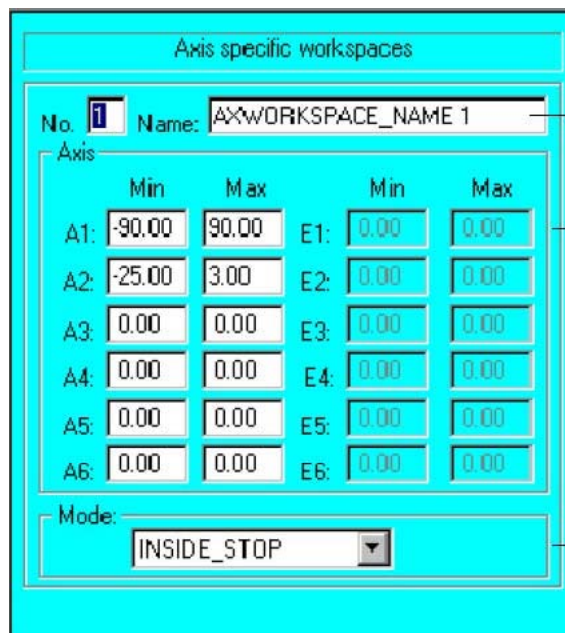
按软键“Cartesian”,使你进入立方体工作区域的定义状态窗口.



详细请参阅章节2.8.1.

按软键“Axis spec.”使你进入轴工作区域监测的定义状态窗口.

Axis spec.



No.	Name	Axis	Min	Max
1	AXWORKSPACE_NAME 1			
		A1	-90.00	90.00
		A2	-25.00	3.00
		A3	0.00	0.00
		A4	0.00	0.00
		A5	0.00	0.00
		A6	0.00	0.00
		E1	0.00	0.00
		E2	0.00	0.00
		E3	0.00	0.00
		E4	0.00	0.00
		E5	0.00	0.00
		E6	0.00	0.00

Mode: INSIDE_STOP

轴的名称或代号

具体轴的最大和最小值,对标准轴和外部轴有效

工作区域监测的方式



如果最大/最小值包含确省值“0.00”,无论是否这个轴在监测状态,这个轴都不被监测,这个适用于标准轴(A1...A6)和外部轴(E1...E6).

KRL变量“\$WORKSPACE[n]”有下面结构:

```

$AXWORKSPACE[1]={A1_N -90.0,A1_P 90.0, A2_N -25.0,A2_P 3.0,A3_N
0.0,A3_P 0.0,A4_N 0.0,A4_P 0.0,A5_N 0.0,A5_P 0.0,A6_N 0.0,A6_P
0.0,E1_N 0.0,E1_P 0.0,E2_N 0.0,E2_P 0.0,E3_N 0.0,E3_P 0.0,E4_N
0.0,E4_P 0.0,E5_N 0.0,E5_P 0.0,E6_N 0.0,E6_P 0.0,MODE #INSIDE_STOP,
STATE FALSE}

```

“\$AXWORKSPACE[n]”的元素的意义:	
A1_N...A6_N	标准轴的最小值
A1_P...A6_P	标准轴的最大值
E1_N...E6_N	配置的外部轴的最小值
E1_P...E6_P	配置的外部轴的最大轴
“MODE”设置:	
#OFF	关闭工作区域监测.
#INSIDE	所有轴位于定义的工作区域的里边,规定的输出被设置.
#OUTSIDE	所有轴位于定义的工作区域的外边,规定的输出被设置.
#INSIDE_STOP	所有轴位于定义的工作区域的里边,规定的输出被设置,机器人停止,发出错误信息.
#OUTSIDE_STOP	所有轴位于定义的工作区域的外边,规定的输出被设置,机器人停止,发出错误信息.
可能的状态“STATE”:	
TRUE*1	工作区域已经被超越
FALSE*1	工作区域没有被超越
*1这个值仅可以显示,不能改变	



在“INSIDE_STOP”或“OUTSIDE_STOP”方式,如果工作区域被超越,机器人可以被恢复仅在关闭工作区域监测或由倍率增大工作区域后.

使用变量修正功能或在KRL程序使用的变量

变量	意义	Range of values
\$AXWORKSPACE[n] <i>n=1...8</i>	对应轴的工作区域的定义	
\$AXWORKSPACE[n].MODE <i>n=1...8</i>	定义的工作区域监测的类型	#OFF #INSIDE #OUTSIDE #INSIDE_STOP #OUTSIDE_STOP
\$AXWORKSPACE[n].STATE <i>n=1...8</i>	仅读变量,指示是否工作区域被超越 (TRUE)或 (FALSE)	TRUE FALSE
\$AXWORKSPACE_NAME[n] <i>n=1...8</i>	定义的工作区域的名称	最多24个符号

\$WBOXDISABLE	激活 (TRUE) 或禁止 (FALSE) 工作区域倍率功能	TRUE FALSE
---------------	--------------------------------	---------------



“\$AXWORKSPACE”变量的更新触发一个预先的运行停止.



在控制器操作期间,如何改变系统变量的数值,详细请参阅资料**ProgrammingHandbook**,在[Monitor]章的,“Variables”部分.

Change

存储所做的改变,按软键“Change”.



工作区域也能够在文件*.SRC里进行定义、关闭和打开,规定在这儿的数值自动进入在文件“\$MACHINE.DAT”里,控制器下一次接通时有效。

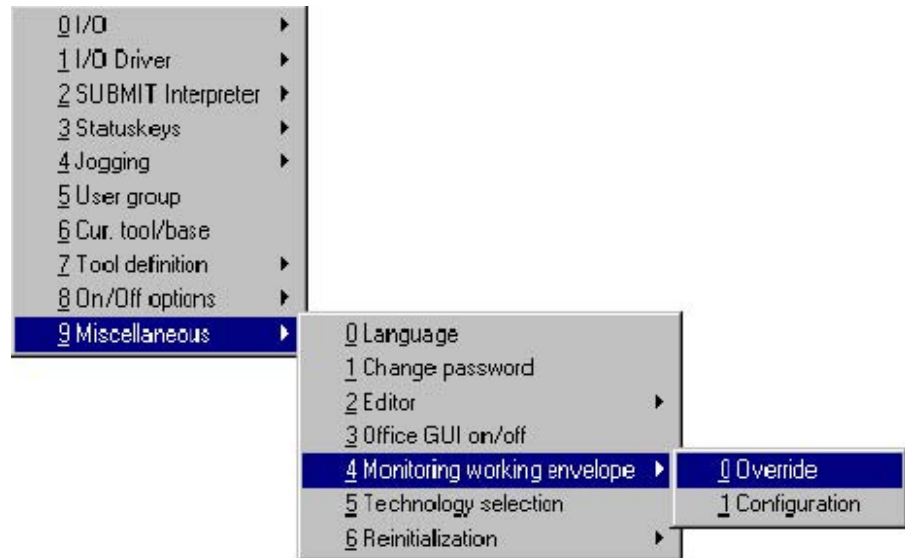
工作区域也能够由更新变量进行改变。

Close

使用软键“Close”关闭配置菜单,除由软键“Change”进行的保存外,其他改变将不保存.

2.8.2.30倍率

工作区域监测能够被倍率，经由菜单“overridden”viathemenu“Configure”-->“Miscellaneous”-->“Monitoringworkingenvelope”-->“Override”.



这个功能可使机器人移动到工作区域的外边.

这个功能仅使用在操作方式TEST(T1).

当个工作区域超越时,下列状态信息出现:

“Axis--specificworkenvelopeno.*n*violated”

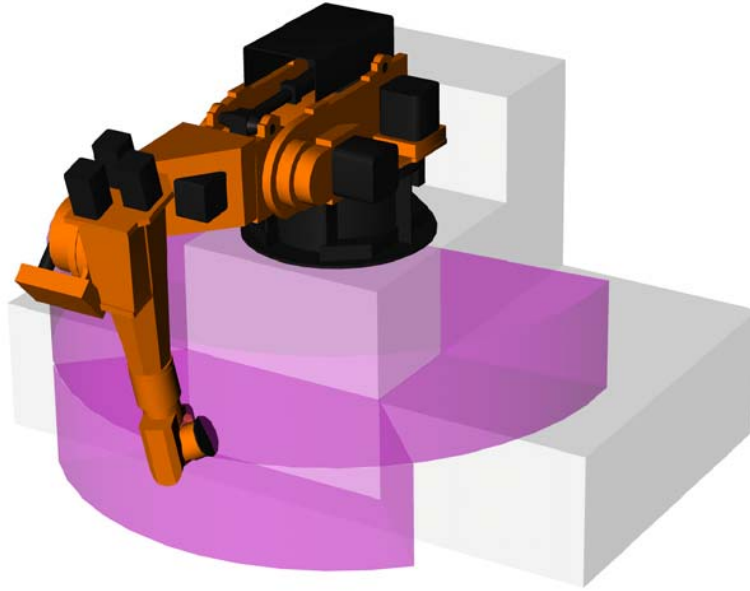
如果此时工作区域检测被倍率,信息由下列内容代替:

“Drivefreeaxis--specificworkenvelopeno.*n*”

一旦机器人不再超越工作区域,信息被自动删除.

2.8.2.4例子

在这个例子中,工作区域定义在机器人可移动的范围,这个工作区域有下列外形:



必须定义几个轴的工作区域,在这个例子中,要求总共4个工作区域,机器人不能超越.



上边/下边的阴影区域,软限位开关的设置有效.

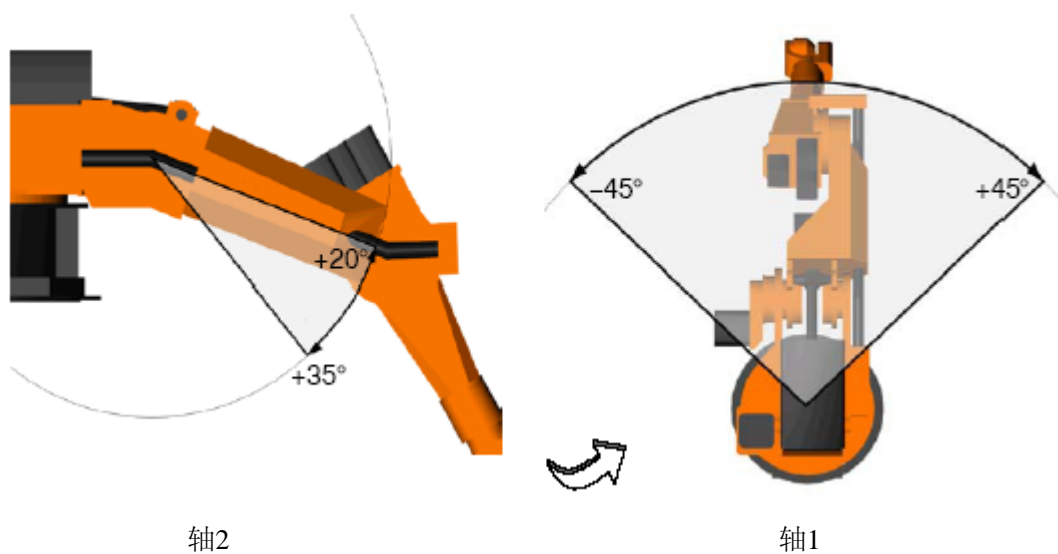


如果要指定的区域由几个不同分工作区域组成,使用模式“#INSIDESTOP”定义非一允许和禁止区域,一旦离开定义的区域之一,则触发监测。

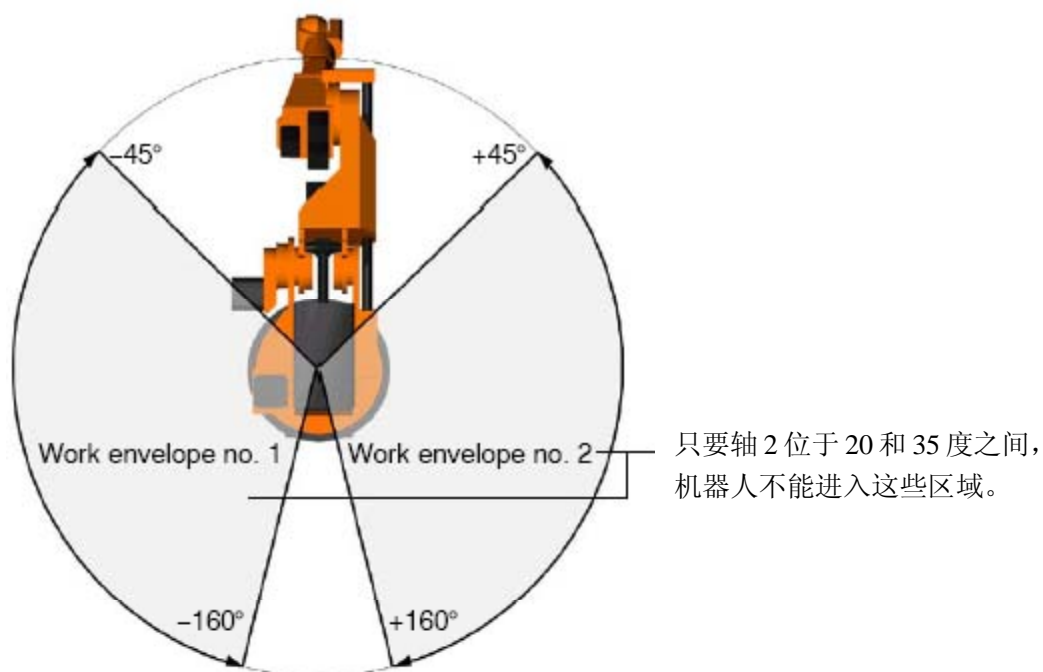
工作区域1+2



一旦轴2位于+20和+35度之间，架安装的机器人的轴1受到限制。



为此,为第一个轴定义两个区域,机器人不能在此区域内移动.



在状态窗口要求的记录有下列外形:

信号

Signals

Cartesian

1: FALSE

2: FALSE

3: FALSE

4: FALSE

5: FALSE

6: FALSE

7: FALSE

8: FALSE

Axis spec.

1: 969

2: 970

3: FALSE

4: FALSE

5: FALSE

6: FALSE

7: FALSE

8: FALSE

根据方式设置具体的输出

工作区域监测的指定信号存储在文件“KRC\Steu\Mada\SMachi-ne.dat”里，一般经由用户接口定义.

```
SIGNAL  $AXWORKSTATE1  $OUT[969]
SIGNAL  $AXWORKSTATE2  $OUT[970]
SIGNAL  $AXWORKSTATE3  FALSE
SIGNAL  $AXWORKSTATE4  FALSE
SIGNAL  $AXWORKSTATE5  FALSE
SIGNAL  $AXWORKSTATE6  FALSE
SIGNAL  $AXWORKSTATE7  FALSE
SIGNAL  $AXWORKSTATE8  FALSE
```

轴

第一个工作区域定义如下:

轴工作区域 1 号, 名称为 “AXWORKSPACE_DOWN1”

轴 1 和轴 2 的的角度条件, 其他轴不监测

一旦轴 1 和轴 2 同时位于限定的工作区域, 机器人停止, 预先规定的输出被设置

Axis	Min	Max	Min	Max
A1:	-160.00	-45.00	E1:	0.00
A2:	20.00	35.00	E2:	0.00
A3:	0.00	0.00	E3:	0.00
A4:	0.00	0.00	E4:	0.00
A5:	0.00	0.00	E5:	0.00
A6:	0.00	0.00	E6:	0.00

Mode: **INSIDE_STOP**

第二个工作区域:

轴工作区域 2 号, 名称为 “AXWORKSPACE_DOWN2”

轴 1 和轴 2 的的角度条件, 其他轴不监测

一旦轴 1 和轴 2 同时位于限定的工作区域, 机器人停止, 预先规定的输出被设置

Axis	Min	Max	Min	Max
A1:	45.00	160.00	E1:	0.00
A2:	20.00	35.00	E2:	0.00
A3:	0.00	0.00	E3:	0.00
A4:	0.00	0.00	E4:	0.00
A5:	0.00	0.00	E5:	0.00
A6:	0.00	0.00	E6:	0.00

Mode: **INSIDE_STOP**

相应的KRL指令写法如下:

工作区域的名称:

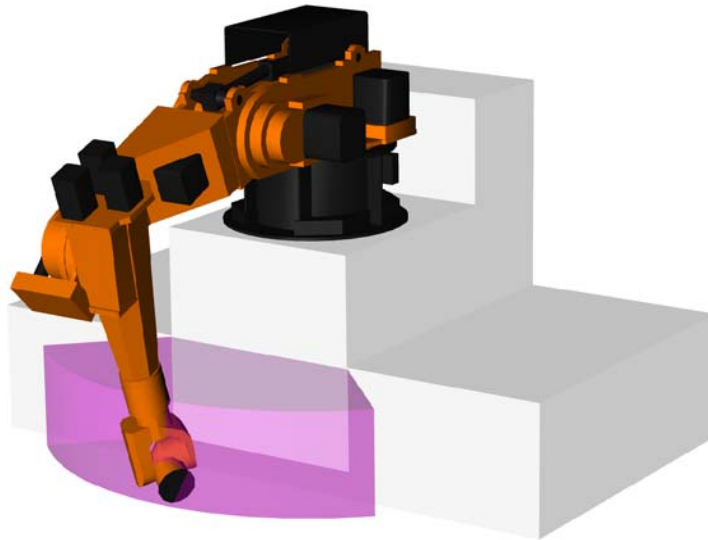
```
$AXWORKSPACE_NAME1[]="AXWORKSPACE_DOWN1"
$AXWORKSPACE_NAME2[]="AXWORKSPACE_DOWN2"
```

轴监测:

```
$AXWORKSPACE[1]={A1_N-160.0,A1_P-45.0,A2_N20.0,A2_P35.0,A3_N0.0,A3_P0.0,A4_N0.0,A4_P0.0,A5_N0.0,A5_P0.0,A6_N0.0,A6_P
```

```
0.0,E1_N 0.0,E1_P 0.0,E2_N 0.0,E2_P 0.0,E3_N 0.0,E3_P 0.0,E4_N  
0.0,E4_P 0.0,E5_N 0.0,E5_P 0.0,E6_N 0.0,E6_P 0.0,MODE #INSIDE_STOP,  
STATE FALSE}  
  
$AXWORKSPACE[2]={A1_N 45.0,A1_P 160.0, A2_N 20.0,A2_P 35.0,A3_N  
0.0,A3_P 0.0,A4_N 0.0,A4_P 0.0,A5_N 0.0,A5_P 0.0,A6_N 0.0,A6_P  
0.0,E1_N 0.0,E1_P 0.0,E2_N 0.0,E2_P 0.0,E3_N 0.0,E3_P 0.0,E4_N  
0.0,E4_P 0.0,E5_N 0.0,E5_P 0.0,E6_N 0.0,E6_P 0.0,MODE #INSIDE_STOP,  
STATE FALSE}
```

不考虑其他轴,导致机器人可以自由运行的外形大致如下:

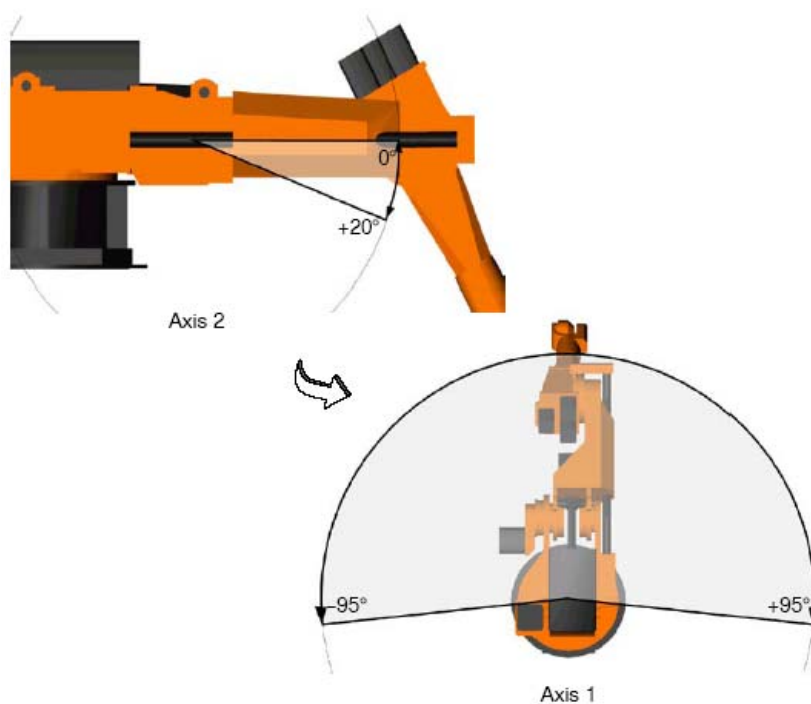


下一步是规定机器人可以移动的范围.

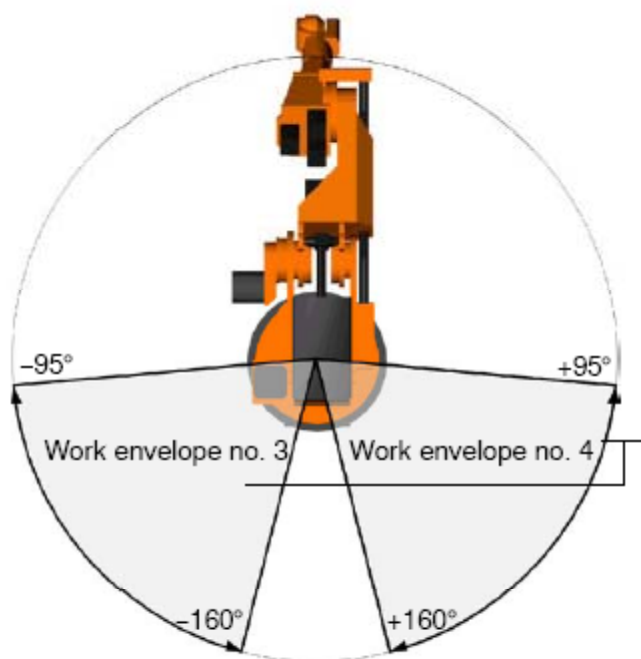
工作区域3+4



一旦轴2位于+0和+20度之间，机器人的轴1受到限制。



在这儿，软限位开关进入坐标系，划分为两个机器人不能进入的分开区域。



只要轴 2 在 0 到 20 度之间，机器人不能进入这些区域

在状态窗口要求的记录有下列外形:

信号

Signals

Cartesian

1: FALSE

2: FALSE

3: FALSE

4: FALSE

5: FALSE

6: FALSE

7: FALSE

8: FALSE

Axis spec.

1: 969

2: 970

3: 971

4: 972

5: FALSE

6: FALSE

7: FALSE

8: FALSE

根据条件设置具体的输出

工作区域监测的指定信号存储在文件“KRC\Steu\Mada\SMachi-ne.dat”里，一般经由用户接口定义.

```
SIGNAL $AXWORKSTATE1 $OUT[969]
SIGNAL $AXWORKSTATE2 $OUT[970]
SIGNAL $AXWORKSTATE3 FALSE
SIGNAL $AXWORKSTATE4 FALSE
SIGNAL $AXWORKSTATE5 FALSE
SIGNAL $AXWORKSTATE6 FALSE
SIGNAL $AXWORKSTATE7 FALSE
SIGNAL $AXWORKSTATE8 FALSE
```

轴

第三个工作区域定义如下:

Axis specific workspaces

No. **3** Name: **AXWORKSPACE_DOWN 3**

Axis

	Min	Max		Min	Max
A1:	-160.00	-95.00	E1:	0.00	0.00
A2:	0.00	20.00	E2:	0.00	0.00
A3:	0.00	0.00	E3:	0.00	0.00
A4:	0.00	0.00	E4:	0.00	0.00
A5:	0.00	0.00	E5:	0.00	0.00
A6:	0.00	0.00	E6:	0.00	0.00

Mode: **INSIDE_STOP**

轴规定工作区域3号, 名称为“AXWORKSPACE_DOWN3”

轴1和轴2的角度说明, 不监测其他的轴

一旦机器人轴1和轴2同时位于指定的工作区域, 机器人停止并设置指定的输出

第四个工作区域:

Axis specific workspaces

No. **4** Name: **AXWORKSPACE_DOWN 4**

Axis

	Min	Max		Min	Max
A1:	95.00	160.00	E1:	0.00	0.00
A2:	0.00	20.00	E2:	0.00	0.00
A3:	0.00	0.00	E3:	0.00	0.00
A4:	0.00	0.00	E4:	0.00	0.00
A5:	0.00	0.00	E5:	0.00	0.00
A6:	0.00	0.00	E6:	0.00	0.00

Mode: **INSIDE_STOP**

工作区域4的名称 “AXWORKSPACE_DOWN 4”

轴 1 和轴 2 的角度说明, 其它轴不监测

一旦轴 1 和轴 2 同时位于指定的工作区域, 机器人停止并输出指定的输出。

相应的KRL指令写法如下:

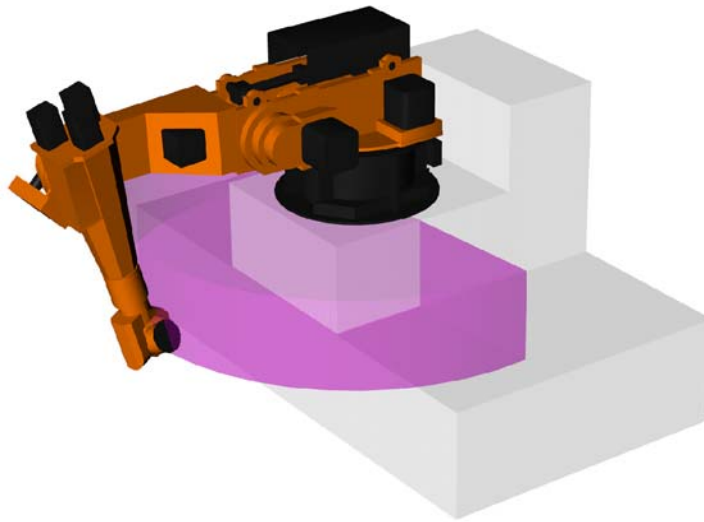
工作区域的名称:

```
$AXWORKSPACE_NAME3[ ]="AXWORKSPACE_DOWN 3"
$AXWORKSPACE_NAME4[ ]="AXWORKSPACE_DOWN 4"
```

轴监测:

```
$AXWORKSPACE[3]={A1_N -160.0,A1_P -95.0, A2_N 0.0,A2_P 20.0,A3_N  
0.0,A3_P 0.0,A4_N 0.0,A4_P 0.0,A5_N 0.0,A5_P 0.0,A6_N 0.0,A6_P  
0.0,E1_N 0.0,E1_P 0.0,E2_N 0.0,E2_P 0.0,E3_N 0.0,E3_P 0.0,E4_N  
0.0,E4_P 0.0,E5_N 0.0,E5_P 0.0,E6_N 0.0,E6_P 0.0,MODE #INSIDE_STOP,  
STATE FALSE}  
  
$AXWORKSPACE[4]={A1_N 95.0,A1_P 160.0, A2_N 0.0,A2_P 20.0,A3_N  
0.0,A3_P 0.0,A4_N 0.0,A4_P 0.0,A5_N 0.0,A5_P 0.0,A6_N 0.0,A6_P  
0.0,E1_N 0.0,E1_P 0.0,E2_N 0.0,E2_P 0.0,E3_N 0.0,E3_P 0.0,E4_N  
0.0,E4_P 0.0,E5_N 0.0,E5_P 0.0,E6_N 0.0,E6_P 0.0,MODE #INSIDE_STOP,  
STATE FALSE}
```

导致工作区域有下列外形:



2.9 力矩方式(软件伺服)

2.9.1 综述

从软件版本2.2发行以后,转换单个轴或几个轴到力矩状态成为可能.



在正常操作中,机器人克服外力来保持编程轨迹.如果外力太大,能够引起工具或机器人的损害.

这个同样也使用于在编程的轨迹上,机器人同其他障碍物发生干涉的场合.

为了某些应用,例如,泄载铸造模具或用电机驱动枪焊接,机器人必须屈服于外力.

在力矩方式,力矩由速度控制器输出极限(正极限和负极限)来定义时,轴用定义的力矩克服外力,或者轴可以由一个外力来移动(推机器人或用其它工具将其移开)



在力矩方式下的轴不能与其他轴协力移动移动.

当力矩方式无效时,轴移动从当前位置到编程的下一个位置.

2.9.1.1 限制和危险

在理论上,力矩方式应用于机器人的每个轴.但是,由于技术上的局限和危险,应用于所有的轴是不明智的.下面的表给出了可能性和局限性的说明.

轴	力矩方式,局限性,危险性.
1	力矩方式对地板和天花板安装的机器人没有限制,没有特殊的危险. 对于墙上安装的机器人,由于轴下垂的原因,力矩方式不可能.
2	力矩方式不可能.由机器人手臂的位置和机器人平衡系统产生的不同力矩使轴有下垂的危险.
3	力矩方式可能.但应避免由连接手臂位置的不同引起的轴下垂.
4 5 6	力矩方式可能,但不明智.由于腕关节齿轮装置的低作用力,很大的力量被要求.

2.9.1.2力矩方式应用的实例

卸载一个铸造模具



机器人紧握零件



零件从铸模中移开,拆卸
器的撞击力量由机器人
围绕轴 1 的旋转来吸收



机器人拿着零件铸造机床移
开

- 机器人紧握零件.
- 零件由一个液压弹射装置从模型中弹出.
- 在弹射发生的瞬间,弹射的撞击不能由机器人控制器测量.
- 如果机器人位置正确,他将使轴1在力矩方式,弹射的撞击由轴1的旋转来吸收,破坏力不会施加到机器人上.



不能进行斜线的弹射运动,因为这个要求一个编程轨迹,涉及几个轴在力矩方式。无论如何,如果机器人以这样的方法安装,既底部的安装发兰平行于弹射运动(例如:倾斜安装),轴1的控制不可成为力矩方式.



机器人的倾斜安装必须在许可的范围内.
看2.9.1.章.

由设置相应的速度控制器输出的正负极限,可补偿由重力导致的力矩,甚至在拖拉零件时,使零件由弹射装置推出时,没有施加一个反作用力.

2.9.2功能的原则

速度控制器的输出极限能够由系统变量“\$CURR_RED[x,x]”来变更. An index is available for both the positive and negative limits. The axis in question is activated in the bitfield “\$TORQUE_AXIS”.

由巧妙地处理速度控制器的输出极限,定义轴是否施加一个“柔和的”或全部的动力. 一旦轴转换到力矩方式,这个轴的监测功能“Regulator limit exceeded”, “Stopped”, “Positioning time” and “Motor blocked”不起作用,只有这个轴的速度被连续监测. 在方式“Test1”,速度必须不能超过

$$\text{\$RED_T1} * \text{\$VEL_ACT_MA}$$

的最大轴速度(机床数据10%*110%=11%),否则,监测功能被出发,驱动断电.

在方式Test2和自动方式,能够借助于变量“\$TORQ_VEL[]”在程序中改变监测极限,以这种方法,速度可达到最大150%.

当力矩方式不有效时,轴从当前位置移动到下一个编程位置,变量“\$TORQUE_AXIS”的改变触发一个预先的运行停止.



如果程序取消或复位,力矩方式自动无效.

在手动快速的场合下,在手动快速运行期间,力矩方式无效.机器人重新定位后,力矩方式重新有效.

在重新定位期间,如果它被设置到“soft”,轴不移动.系统内部在移动,移动到编程的路径

(BCO),也就是说,开始键必须保持按下,直到到达BCO.

2.9.3 软轴的例子

2.9.3.1 轴1

轴1设置成软轴,速度控制器输出限制到0%.这样做的结果是,轴不再由速度和位置控制器监测,可由外部的力量移动

```
$TORQUE_AXIS='B000001'      ;轴1到力矩状态
$CURR_RED[1,1]=0            ;正的力矩极限设置到0
$CURR_RED[1,2]=0            ;负的力矩极限设置到0
PTP{A190}                   ;释放制动;
                             ;轴变软,能被移动
WAITFOR$IN[17]               ;等待取消力矩方式的信号;(例如,
                             弹射装置)
$TORQUE_AXIS=0               ;再次设置轴到硬
                             "hard"
PTP{A1-20}                   ;移动到下一个位置
```

2.9.3.2 轴3

由限制速度控制器输出到当前的保持力矩,轴3也能设置到软.轴在保持力矩处由速度和位置控制器监测,轴可由外部力量移动.“

当轴3在水平位置时,到达最大的保持力矩.



如果轴承受一个极大的负荷,或当前极限设置不正确,机器人手臂将会弯曲.

```
PTP{A3 90}                   ;轴3在水平位置
$TORQUE_AXIS='B000100'       ;轴3到力矩方式
$CURR_RED[3,1]=ABS($CURR_ACT[3]) ;正力矩极限到保持力矩
$CURR_RED[3,2]=ABS($CURR_ACT[3]) ;负力矩极限到保持力矩

PTP {A30}                    ; 移动A3到垂直方向
                             ;轴变软并且能够被移动
```

```

WAIT FOR $IN[17]           ;等待取消力矩方式的信号
                             ;(例如:弹射装置)
$TORQUE_AXIS = 0           ;再次设置轴到“硬”状态
PTP {A1-20,A380}           ;移动轴到下一个轴位置

```

在任何情况下轴3必须移动到压迫范围(在例子中是垂直方向)以外的位置,否则,由于轴位置控制器用保持力矩维持轴的定位,轴不能向上推.如果你试图推轴离开,你必须克服轴的重量.

腕轴用0%的力矩设置到软“soft”,由于腕轴齿轮装置的作用而不适合这种做法;轴不能很容易地由外力移动.



轴2从不设置到软,因为当它在垂直位置的时候,轴有最低的力矩和压力补偿.轴下垂的危险很大!

2.9.4轴的定义力矩的例子



下面用插图说明了一个电机驱动焊接枪应用的例子:

电机驱动焊接枪(外部轴E1)施加一个定义的力矩在零件上.移动焊枪到零件上,设置产生定义力矩的电流,产生的力矩施加到零件上.

由于枪的电极先触摸到金属,它接触住零件用一个定义的力矩,所以轴不能到达编程的位置.如果内部运动的电流极限被降低,并且第二电流极限设置到 100%,它以最大速度从零件返回.

```

PTP{E10}                   ;在接触金属之前移动枪
$TORQUE_AXIS='B1000000'    ;力矩方式有效
$CURR_RED[7,1]=20          ;设置定义的力矩;正的极限力矩
PTP{E1-10}                 ;“移动”10mm进入零件;外部力矩
WAITFORWELDEND             ;等待焊接完成“weldend”信号;再次设置正极限到高“on the fly”
TRIGGERWHENDISTANCE=0DELAY=50DO$CURR_RED[7,1]=100
PTP{E120};                 ; 打开枪
$TORQUE_AXIS=0             ;力矩方式无效

```

2.9.5力矩方式的变量

下列变量在力矩方式时有效:

REAL\$CURR_ACT[12]

轴1--12的电流的百分比

\$CURR_MAX*\$CURR_LIM(--100%to+100%).

REAL\$CURR_RED[12,2]

轴1--12的电流的电流极限百分比(0%to+100%).

Index1的正极限index2的负极限.这些极限是绝对值的0%到100%之间.



电流限制可能引起没有足够的保持，制动或移动轴力矩。可能引起机器或人员的伤害！

电流限制仅与“\$SORQUE_AXIS”一起使用。

INT \$TORQUE_AXIS

这是一个为力矩驱动轴A1--A6和外部轴E1--E6的位排列.

每个轴对应一个位.为这个轴的监测功能无效.



改变变量的值触发一个预先的运行停止.

REAL\$TORQ_VEL[12]

监测力矩轴的极限速度，使用最大速度的百分比.



一个轴在力矩方式时，所有这个轴的监测功能无效.

为了确保轴的硬件故障监测或轴的下垂监测，速度被监测.

在操作方式T2和自动方式，在程序中使用变量“\$TORQ_VEL”设置最大速度.

在操作方式T1,速度设置在机床数据里，如果速度被超越，驱动被断开，出现对应的错误信息。

2.10干涉监测

2.10.1功能

如果机器人碰撞到一个零件或工具抓住一个零件，位置和速度控制器作出反应，相应轴的力矩增加。取决于应用，机器人可以克服阻力，连续运动，这个可以损伤工具或零件。



外部轴不能被监测。

用户能够配置力矩限制和响应时间，一个KRL变量可定义一个力矩区域（也称作“monitoring tunnel”）。如果力矩离开这个区域，在一个规定的响应时间后，位置控制器停止响应，触发一个对应的信息。

确省时，监测是有效的。确省的检测区域是200%。为个别的运动或程序，监测能更灵敏。



在一个复位的场合下，选择段或不选择程序，被设置的极限复位到Custom.dat里的确省值。



干涉监测功能没有任何损害保证,但是能够降低损害的程度。移动速度和力矩值决定损害的程度。

在KRC:\STEU\MACHINE.DAT，输出信号\$COLL_ENABLE和\$COLL_ALARM保持有效。如果一个轴的电流监测极限既\$torqmon小于200%，COLL_ENABLE被设置。如果信息“117 Torque exceeded axis Ax”出现，COLL_ALARM 被设置。只要\$STOPMESS有效，输出保持设置。

2.10.2配置

为了能够使用干涉监测功能，加速度适配器必须被接通,这是一个变量“\$ADAP_ACC”为值“#STEP1”的场合。变量能够被发现在“C:\KRC\Roboter\KRC\R1\MaDa\ROBCOR.DAT”:

:DECLADAP_ACC\$ADAP_ACC=#STEP1;加速度适配:



负荷数据必须根据干涉精确计算，力矩限制必须适配到具体的应用。

监测响应的时间单位用[ms]，能够使用变量“\$TORQMON_TIME”进行设置.变量能够被发现在文件“C:\KRC\Roboter\KRC\Steu\MaDa\SCUSTOM.DAT”:

:REAL\$TORQMON_TIME=0.0;力矩监测的响应时间

监测区域的大小能够借助下面的变量进行设置，在文件“C:\KRC\Roboter\KRC\MaDa\SCUSTOM.DAT”:

- 程序模式监测通道的缺省值用百分比:

```
:
$TORQMON_DEF[1]=200
$TORQMON_DEF[2]=200
$TORQMON_DEF[3]=200
$TORQMON_DEF[4]=200
$TORQMON_DEF[5]=200
$TORQMON_DEF[6]=200
:
```

- 命令模式监测通道的缺省值用百分比:

```
:
$TORQMON_COM_DEF[1]=200
$TORQMON_COM_DEF[2]=200
$TORQMON_COM_DEF[3]=200
$TORQMON_COM_DEF[4]=200
$TORQMON_COM_DEF[5]=200
$TORQMON_COM_DEF[6]=200
:
```



误差带的宽度=最大力矩[Nm] * “\$TORQMON...”的百分比。

运动力矩被监测在

- 程序方式

--提前运行:

在KRL程序里，一个规定力矩误差带的段能够用变量“\$TORQMON[]”进行规定.

--主运行:

当变量写入中断程序时，监测立即有效或无效.

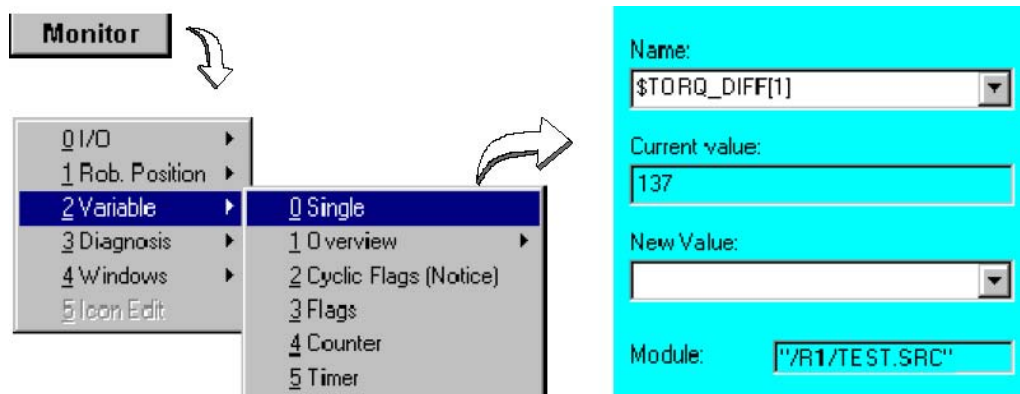
- 指令方式

--在文件“SCUSTOM.DAT”里，变量“\$TORQMON_COM_DEF[1]...[6]”的数值为力矩监测确省时有有效。借助于变量更新功能用户能够改变变量“\$TORQMON_COM[1]...[6]”的数值改变监测极限.

发生的最大力矩偏差能够使用变量“\$TORQ_DIFF[1]...[6]”用百分比读出.能够用于最佳化力矩监测。



在移动或执行一个移动指令前，使用变量变更功能设置变量到“0”。



现在执行移动指令和读变量,给定值对应于最大的力矩。

现在设置监测到变量“\$TORQ_DIFF[]”的值，加上5--10%的富裕量。



仅数值“0”能够分配到变量“\$TORQ_DIFF[]”。



对不正确的设置造成的损害不负任何责任！

3 外部自动

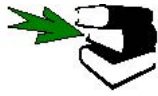
3.1 概述

当使用互联生产线时，可以从中心位置开始机器人加工。

一个主计算机可以通过“外部自动”接口进行通讯并激活各种机器人加工。 机器人控制器可以发送关于操作状态和缺省信号的信息到主计算机。

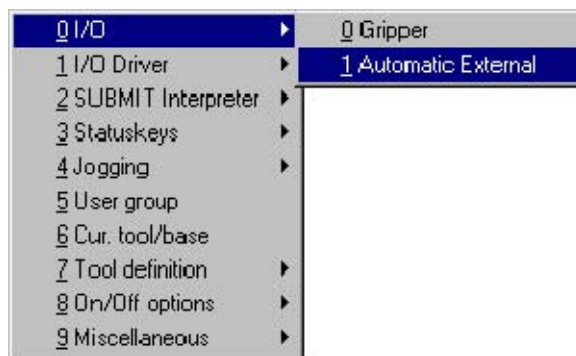
在KRC1中，这将全部通过自动系统启动有关工艺的组织程序CELL.SRC和P00模块的功能。

3.2 配置接口



有关“外部自动”接口显示的信息可以参考章节[监视器]。

Configure



“外部自动”接口信号必须分配给机器人控制器的物理输入和输出。要进行此项工作，从“配置”菜单选择选项“I/O”。



打开对于外部自动接口输入的状态窗口。

使用“↑”和“↓”箭头键，移动焦点(黑蓝高亮)到要求的框。然后使用数字键盘输入分配信号的接口号。



外部自动的输入和输出解释可以参考3.6节。

3.2.1 输入

此状态窗口显示外部自动接口的输入。

Automatic External - Configuration: Inputs				
	Term	Type	Name	Value
1	Type programno.	Var	PGNO_TYPE	1
2	REFLECT_PROG_NR	Var	REFLECT_PROG_M 0	
3	Bitwidth programno.	Var	PGNO_LENGTH	8
4	First bit programno.	IO	PGNO_FBIT	33
5	Parity bit	IO	PGNO_PARITY	41
6	Programno. valid	IO	PGNO_VALID	42
7	Programstart	IO	\$EXT_START	1026
8	Move enable	IO	\$MOVE_ENABLE	1025
9	Error confirmation	IO	\$CONF_MESS	1026
10	Drives off (invers)	IO	\$DRIVES_OFF	1025
11	Drives on	IO	\$DRIVES_ON	140
12	Activate interface	IO	\$I_O_ACT	1025

Term

有关变量或输入的功能说明



类型可以是变量（黄色）或输入（绿色）

Name

相应输入的变量名

Value

输入或通道号的 值

Edit

软键“EDIT”用于修改输入或通道号的 值

140

OK

Cancel

当按下“OK”时，将检查修改值的有效性；如果输入是许可的，将接受新的值。

“CANCEL”用于取消修改。

Outputs

软键“Outputs”触发相关的状态窗口。

Close

通过软键“Close”关闭状态窗口。

3.2.1.1 输出

状态窗口显示外部自动接口的输出。

Automatic External - Configuration: Outputs				
	Term	Type	Name	Value
1	Control ready	NO	\$RC_RDY1	137
2	Alarm stop active	NO	\$ALARM_STOP	1013
3	User safety switch closed	NO	\$USER_SAF	1011
4	Drives ready	NO	\$PERI_RDY	1012
5	Robot calibrated	NO	\$ROB_CAL	1001
6	Interface active	NO	\$I_O_ACTCONF	140
7	Error collection	NO	\$STOPMESS	1010
8	PGNO_FBIT_REFL	NO	PGNO_FBIT_REFL	999

Term 特定变量或输出的功能说明

Var NO 类型可以是变量(黄色)或输出(绿色)

Name 相应输出的变量名

Value 输出或通道号的值

Tab+ 软键“Tab+” 和“Tab-” 允许上下移动到相邻的变量组

Tab- 软键“Edit” 用于修改输出值或变量号

Edit

137

OK 当按下“OK” 时，将检查修改值的有效性；如果输入是许可的，将接受新的值。

Cancel “CANCEL” 用于取消修改。

Inputs 软键“Inputs” 触发相关的状态窗口。

Close 通过软键“Close” 关闭状态窗口

i 如果修改“\$MACHINE.DAT” 文件，当数据传送时提交编译器会短暂取消，一旦保存它们将自动重新选择。

3.3 自动系统启动

如通过设置系统变量\$I_O_ACT为真，作为反馈信号输出\$I_O_ACTCONF也转换为真。如果所有其它的启动条件满足，通过\$EXT_START行上的信号启动程序CELL.SRC。

CELL.SRC程序当然也可以在任何时候从用户接口中打开。

对于自动系统启动，下列值必须指定给在文件“C:\KRC\Roboter\KRC\Steu\MaDa\%CUSTOM.DAT”中的系统变量\$PRO_I_O:

```
CHAR $PRO_I_O[] = “/R1/SPS()”
```



控制器接通后, 总是试图执行在\$PRO_I_O中指定的程序。



I/O接口完全激活前，必须首先设置在变量\$I_O_ACT信号声明指定的输入。

3.4 特殊的工艺组织程序CELL.SRC

对于连接用户定义的外部子程序...

```
;EXT EXAMPLE1 ( )
;EXT EXAMPLE2 ( )
;EXT EXAMPLE3 ( )
```

初始化序列...

```
INIT
BAS INI
CHECK HOME
PTP HOME Vel=100% DEFAULT
AUTOEXT INI
```

循环启动...

```
LOOP
```

从主计算机调用连同程序号的P00模块...

```
P00(#EXT_PGNO, #PGNO_GET, DMY[], 0)
```

根据接收到的程序号检查结构...

```
SWITCHPGNO
```

如果程序号PGNO=1...

```
CASE1
```

... 与主计算机通讯程序号的接收...

```
P00(#EXT_PGNO, #PGNO_ACKN, DMY[], 0)
```

... 并调用用户定义的程序EXAMPLE1

```
;EXAMPLE1 ( )
```

如果程序号PGNO=2...

```
CASE2
```

... 与主计算机通讯程序号的接收...

```
P00(#EXT_PGNO, #PGNO_ACKN, DMY[], 0)
```

... 并调用用户定义的程序EXAMPLE2

;EXAMPLE2()

如果程序号PGNO=3...

CASE3

... 与主计算机通讯程序号的接收...

P00 (#EXT_PGNO, #PGNO_ACKN, DMY[], 0)

... 并调用用户定义的程序EXAMPLE3

;EXAMPLE3()

如果通过主计算机通讯的程序号未发现CASE分支

DEFAULT

执行错误处理...

P00 (#EXT_PGNO, #PGNO_FAULT, DMY[], 0)

检查结构的结束...

ENDSWITCH

循环的结束...

ENDLOOP

程序的结束...

END

3.5 P00(外部自动)模块

P00模块包含通过主计算机传送程序号的功能。功能INIT_EXT, EXT_PGNO, CHK_HOME和EXT_ERR一起编组在通用子程序中。

3.5.1 EXT_PGNO功能

此功能接收完全的信号处理通过主计算机传送程序号的进程。

可以通过下列三种参数之一来调用：

#PGNO_GET程序号请求

#PGNO_ACKN程序号接收的通讯

#PGNO_FAULT错误处理

3.5.1.1 来自主计算机程序号的请求

EXT_PGNO(#PGNO_GET)

如果主计算机检测到PGNO_REQ行上请求的程序号，它在提供的机器人控制器上设置程序号为二进制值。

为了增加传送的可靠性，可以在程序号外传送校验位PGNO_PARITY到主计算机。如果信号电平保持稳定，主计算机设置PGNO_VALID或EXT_START行以便请求机器人控制器来读取程序号。EXT_PGNO功能计算接收的程序号的校验并与检验位比较。如果结果为正，功能返回接收的程序号为一个整数值。如果接收的和计算的校验不一致，程序号设置为“0”值。一个错误信息显示在KCP的状态窗口。



由于在校验错误的情况下总是设置程序号为0，这个值当然不能在CELL.SRC中作为有效的程序号！

3.5.1.2 程序号接收的通讯

EXT_PGNO(#PGNO_ACKN)

一旦程序号正确地传送，在CELL.SRC中的检查结构试图指定此程序号到应用程序。如果成功功能自动地取消程序号的请求。通过设置行APPL_RUN发送信号到主计算机。

如果另一方面失败，调用下面说明的错误处理功能。

3.5.1.3 错误处理

EXT_PGNO (#PGNO_FAULT)

如果程序号不能正确传送，如：

- (1) 校验检查失败, 或
- (2) BCD编码不正确或更准确地：译码不能产生一个有效的结果或
- (3) 没有应用程序指定这个程序号,

在KCP的信息窗口EXT_PGNO功能显示传送错误。PGNO_REQ行保持设置状态。通知主计算机传送包含错误。



故障传送可以通过主计算机基于超时原则来识别。当PGNO_VALID行已设置时启动超时。如果PGNO_REQ行请求的程序号不能在规定的时间内（大约200ms）后取消，传送将产生一个错误。主计算机现在对错误可能没有反应。

3.5.2 EXT_ERR功能

该功能可以用于发送8个规定的输出和1...255之间的错误号到主计算机。另外，64个最近的错误存储在ERR_FILE环形存储器中用于分析。

为了使用EXT_ERR功能，需要编辑如下所述的hep00.dat文件：

&ACCESSR	
&COMMENT EXTERNAL package	
DEFDATP00	
BOOLPLC_ENABLE=TRUE	设置此值为真
INTI	
INTF_NO=1	
	INTMAXERR_C=1
	此处输入定义的参数传送的控制器错误号
INTMAXERR_A=1	此处输入定义的参数传送的应用错误号
SIGNAL ERR \$OUT[25] TO \$OUT[32]	此处规定机器人控制器的输出主计算机应用于读取错误号在例子中输出从25到32
BOOL FOUND	
STRUC PRESET INT OUT, CHARPKG[3], INTERR	
DECL PRESET P[255]	在下面部分必须输入错误的参数：
	OUT-
	传送到主计算机的错误号
	PKG[]--
	工艺包
	ERR-
	选择工艺包中的错误号
P[1]={OUT2, PKG[] " P00" , ERR10}	... 仅可以输入
	范围P[1]...P[127]
P[127]={OUT27, PKG[] " S00" , ERR11}	应用错误
P[128]={OUT12, PKG[] " CTL" , ERR1}	... 仅可以输入
	范围P[128]...P[255]
P[255]={OUT25, PKG[] " CTL" , ERR10}	控制器错误

配置

```
STRUC ERR_MESS CHARP[3], INTE  
DECL ERR_MESSERR_FILE[64]  
ERR_FILE[1]={P[] ” XXX” ,E0}  
...  
ERR_FILE[64]={P[] ” XXX” ,E0}  
ENDDAT
```

3.6 信号说明

信号被写保护，但是任何时候可以读取或在程序中使用。

3.6.1 输入

3.6.1.1 PGNO_TYPE

这不是输入也不是信号只是一个变量。它的值决定格式，通过主计算机发送的程序号按此格式读取。

PGNO_TYPE	读为	含义	例子
1	二进制数	通过更高级的控制器传送程序号作为一个二进制编码的整数	<pre> 0010 011 1 => PGNO = 39 </pre>
2	BCD值	通过更高级的控制器传送程序号作为一个二进制编码的十进制数	<pre> 0010 0111 2 7 => PGNO = 27 </pre>
3	“n中之一” *1	通过更高级的控制器或外围传送程序号作为“n中之一”编码值	<pre> 00000001 => PGNO = 1 00001000 </pre>
*1当使用此传送格式时，PGNO_REQ, PGNO_PARITY和PGNO_VALID的值无法赋值，因此没有符号。			

3.6.1.2 PGNO_LENGTH

这不是一个输入也不是一个信号，只是变量。它的值决定定义通过主计算机发送程序号的位数。

PGNO_LENGTH=1...16

例子：

PGNO_LENGTH=6=>外部程序号是6位



当PGNO_TYPE是值2(程序号读取为BCD值)时，仅4, 8, 12和16是允许的位数值。

3.6.1.3 PGNO_FBIT

输入表示程序号的首位。

PGNO_FBIT=1...1024 (PGNO_LENGTH)

例子：

PGNO_FBIT=5=>外部程序号以\$IN[5]开头。

3.6.1.4 REFLECT_PROG_NR

该选项允许你决定程序号是否镜像在一个定义的输出区。变量值可以通过外部自动接口配置来改变。

REFLECT_PROG_NR	功能
0	失效
1	激活



信号输出使用“PGNO_FBIT_REFL”定义的输出开始。

3.6.1.5 PGNO_PARITY

输入来自主计算机传送的校验位。

输入	功能
负值	奇校验
0	无法赋值
正值	偶校验



当 PGNO_TYPE 是值 3（程序号读取为“n 中之一”值）时，PGNO_PARITY 不能赋值。

3.6.1.6 PGNO_VALID

输入到读取来自主计算机传送的程序号的命令。

输入	功能
负值	信号下降沿处传送号
0	有关EXT_START行的信号上升沿处传送号
正值	信号上升沿处传送号



当 PGNO_TYPE 是值 3（程序号读取为“n 中之一”值）时，PGNO_PARITY 不能赋值。

3.6.1.7 EXT_START

如果I/O接口激活，此输入可以设置为开始或继续程序。



仅上升沿被赋值



在外部自动方式下，没有BC0运行，因此也没有程序停止在第一个编程位置处。这将应用到路径偏离时的动态制动后或机器人手动移动后的路径之外。

在这个情况下编址的第一个位置是中断前保存在\$POS_RET 中的位置。这意味着当设置EXT_START时，操作者必须确认机器人已处于此位置或可以安全的接触到它。

第一个运动程序段必须是带有目标点绝对坐标的 PTP 程序段。机器人总是以最大速度准确地移动到目标点，具有任何近似定位编程的指令被忽略！

3.6.1.8 MOVE_ENABLE

通过主计算机使用的输入来检查机器人驱动。

信号	功能
真	手动移动和程序执行是可以的
假	所有驱动被停止并且所有有效的指令被抑制



如果通过主计算机切断驱动器，信息“GENERALMOTIONENABLE”出现在KCP的信息窗口。如果该信息被复位并且给出启动信号，机器人可以再次启动。



启动期间，运动允许变量“\$MOVE_ENABLE”经常配置“\$IN[1025]”值。如果随后没有配置一个不同的输入，外部开始不可能。

3.6.1.9 CHCK_MOVENA

如果变量\$CHCK_MOVENA是值“FALSE”，MOVE_ENABLE可以通过。在“C:\KRC\Roboter\KRC\Steu\MaDa\OPTION.DAT”中可以改变变量值。

信号	功能
真	MOVE_ENABLE监控是激活的
假	MOVE_ENABLE监控是无效的



为了使用MOVE_ENABLE监控，\$MOVE_ENABLE必须配置输入“\$IN[1025]”。否则，“\$CHCK_MOVENA无效。

3.6.1.10 CONF_MESS

设置该输入使能计算机自动地删除(确认)错误信息。



仅在信号的上升沿被赋值。

一旦排除错误原因就可以确认错误信息。

3.6.1.11 DRIVES_ON

该输入的高电平脉冲至少有20ms宽度，主计算机才接通机器人驱动。



对于软件版本1.1.7以上，使用电源模块PM6—600版本A, B或C，或PM0—600Pro时，在一个驱动和下一个驱动之间有13—18.5秒的延时以保护驱动继电器K2。
18.5秒结束时检测激活的DRIVES_ON的正边沿并且驱动被延时然后重新激活。

3.6.1.12 DRIVES_OFF

该输入的低电平脉冲至少有20ms宽度，主计算机才切断机器人驱动。

3.6.2 输出

3.6.2.1 STOPMESS

该输出通过控制器设定以便与主计算机通讯要求机器人停止产生的任何信息。
(例如急停, 驱动状态, 操作者安全, 指令的速度等)

3.6.2.2 PGNO_REQ

该输出上信号的改变请求主计算机发送一个程序号。



信号的两个边沿被赋值。

当 PGNO_TYPE 是值 3 (程序号读取为 “n 中之一” 值) 时, PGNO_REQ 不被赋值。

3.6.2.3 PGNO_FBIT_REFL

镜像的输出表示程序号的第一位。为使用该选项, 变量 “REFLECT_PROG_NR” 必须指定为 “1”。

如果通过 PLC 选择的程序被用户取消, 输出区的开始使用 PGNO_FBIT_REFL 设定为 “FALSE”。这样, PLC 可以防止手动重新启动程序。



如果编译器处于 CELL 中 PGNO_FBIT_REFL 也被设定为 “FALSE”。
PGNO_FBIT_REFL=1...1024 (PGNO_LENGTH)

输出的大小取决于程序号 (PGNO_LENGTH) 确定的位数。

例子:

PGNO_FBIT_REFL=5=>程序号以 \$OUT[5] 开始。

3.6.2.4 APPL_RUN

通过设定这个输出, 机器人控制器与主计算机通讯程序正被执行的细节。



APPL_RUN 的值不能小于 “0”。

3.6.2.5 PERI_RDY

通过设定这个输出, 机器人控制器与主计算机通讯驱动器接通的细节。

3.6.2.6 ALARM_STOP

在急停情况下复位该输出。

3.6.2.7 USER_SAF

如果安全栅栏监控开关被打开(在AUTO方式下)或使能开关被释放(在TEST方式下)复位该输出。

3.6.2.8 T1, T2, AUT, EXTERN

当选择相应的操作方式时设定这些输出。

3.6.2.9 ON_PATH

只要机器人停留在编程路径上, 该输出保持设定。

BCO运行后设定输出ON_PATH。该输出保持设定直到机器人离开路径; 复位程序或执行程序段选择。ON_PATH信号没有误差窗口, 只要机器人离开路径信号才复位。

3.6.2.10 NEAR_POSRET

第二个信号NEAR_POSRET允许主计算机决定是否机器人处于保存在\$POS_RET的位置的球范围内。用户可以使用变量\$NEARPATHTOL设置在 \$CUSTOM.DAT文件中的球半径。



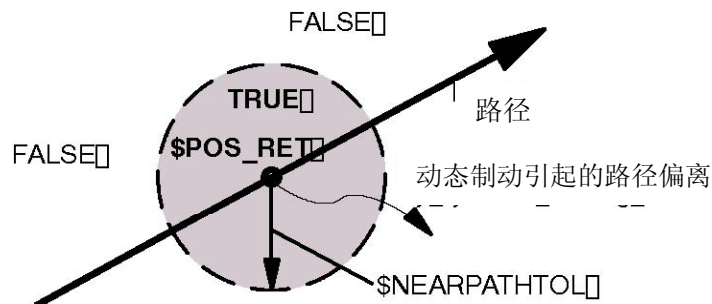
主计算机可

以使用该信息来决定程序是否可以重新启动。

返回位置\$POS_RET是机器人离开路径的位置。

当转换到“外部自动”方式时，进行一个检查来看是否变量“\$NEAR_POSRET”设置为“TRUE”。如果不是这种情况，相应的错误信息将显示在信息窗口中。

\$NEAR_POSRET



NEAR_POSRET可能的状态:

TRUE:

ON_PATH已设定，或如果ON_PATH未设定：\$POS_RET是有效的并且位置在\$POS_RET的球范围内

FALSE:

ON_PATH已复位，\$POS_RET是无效的或位置在\$POS_RET的球范围外。

设置:

文件:

\$MACHINE.DAT

SIGNAL \$NEAR_POSRET \$OUT[XXX]

3.6.2.11 PRO_ACT

如果机器人级上有处理或程序执行，该输出总是设定的。

它的信号状态来自系统变量:

\$PRO_STATE1=#P_ACTIVE→\$PRO_ACT=TRUE

其它的处理状态 →\$PRO_ACT=FALSE

只要程序或中断正被处理，处理就是激活的。仅在所有的脉冲输出或所有触发器已被处理后程序结束时，程序处理设置为无效。错误停止的情况下有3种可能的情形。

- 错误停止时如果中断已激活但不执行，处理被认为是非激活的。（PRO_ACT=FALSE）
- 错误停止时如果中断已激活并且已执行，处理被认为是激活的（PRO_ACT=TRUE）直到完成中断程序或在它中间发生一个STOP（PRO_ACT=FALSE）为止
- 如果在应用程序中激活中断或产生一个停止，处理就认为是非激活的（\$PRO_ACT=FALSE），如果中断条件满足，处理就认为是激活的（PRO_ACT=TRUE）直到完成中断程序或在它中间发生一个STOP（PRO_ACT=FALSE）为止。

3.6.2.12 IN_HOME

该输出与主计算机通讯机器人是否在HOME位置。

3.6.2.13 ERR_TO_PLC

通过设定该输出，机器人控制器与主计算机通讯控制器或发生的应用错误的细节。



该功能仅在PLC_ENABLE值为FALSE时被激活。.

3.6.3 其它变量

3.6.3.1 PGNO

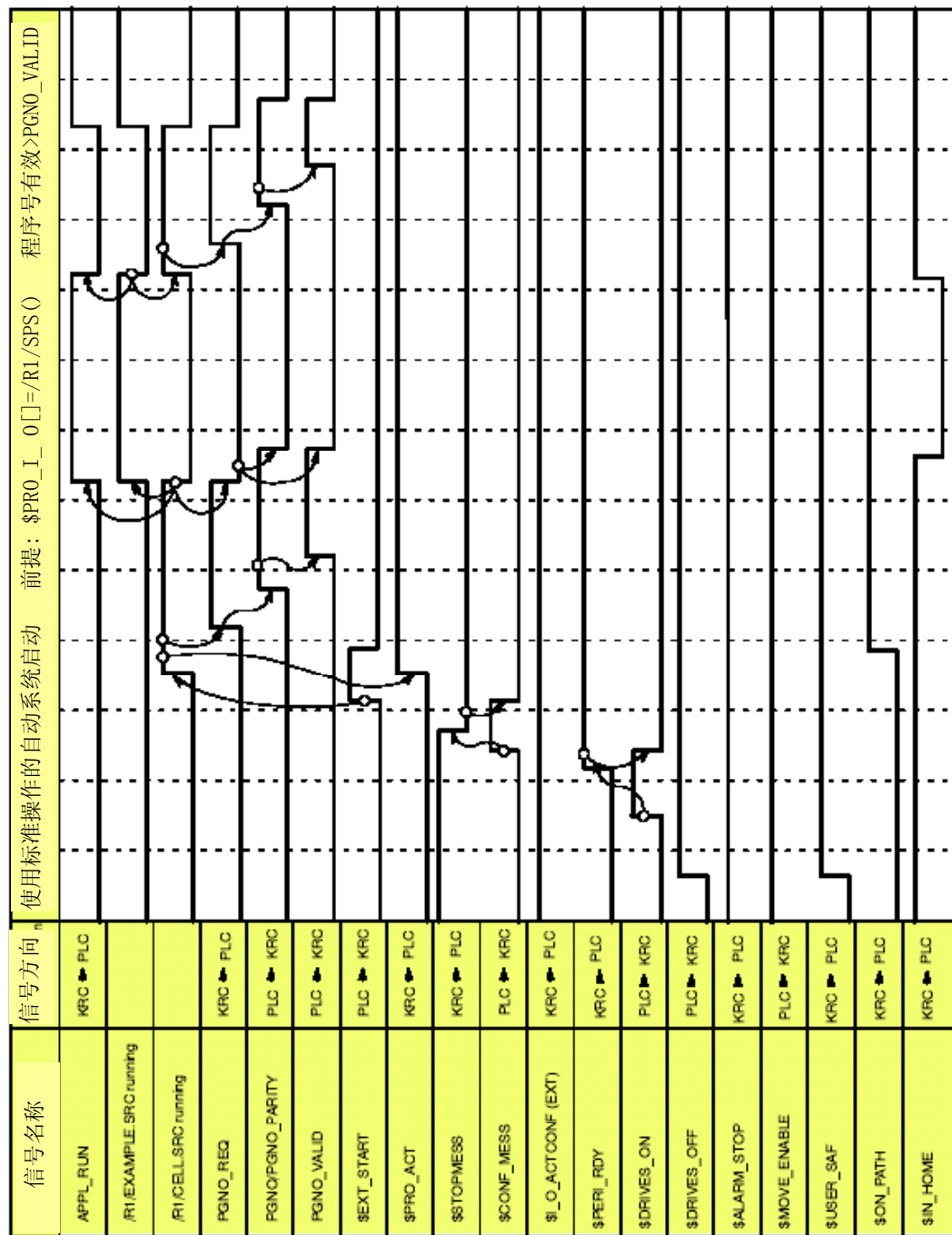
程序EXT_PGNO. SRC在这个变量（不考虑参数化的数据类型）中把从主计算机接收程序号作为一个整数值来存储。

特定的工艺组织程序CELL. SRC使用该变量指定相应程序的程序号。

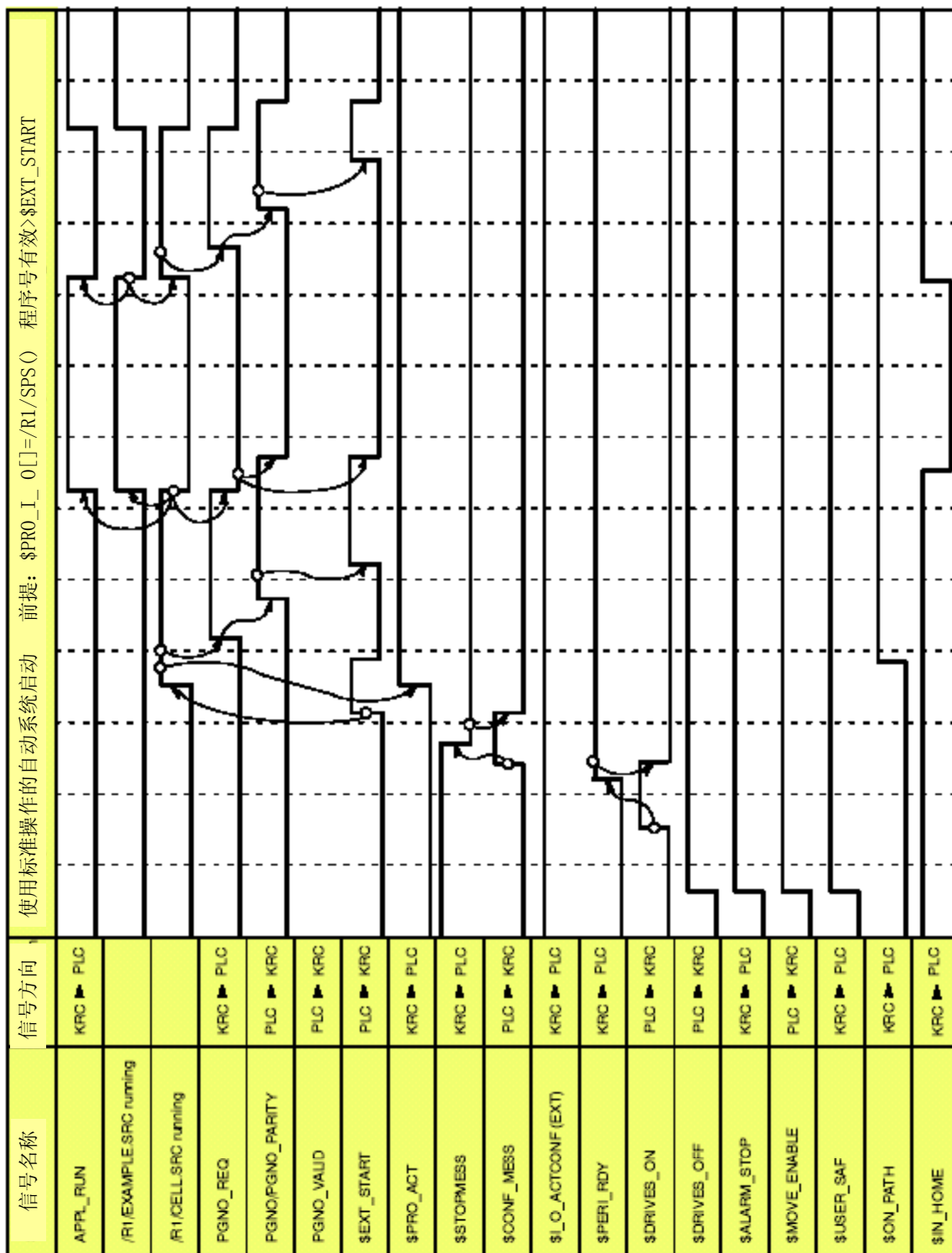
3.6.3.2 PGNO_ERROR

该变量用于程序EXT_PGNO. SRC内部错误管理，千万不要使用或修改！

动与通过 PGNO_VALID 确认的程序号的标准操作

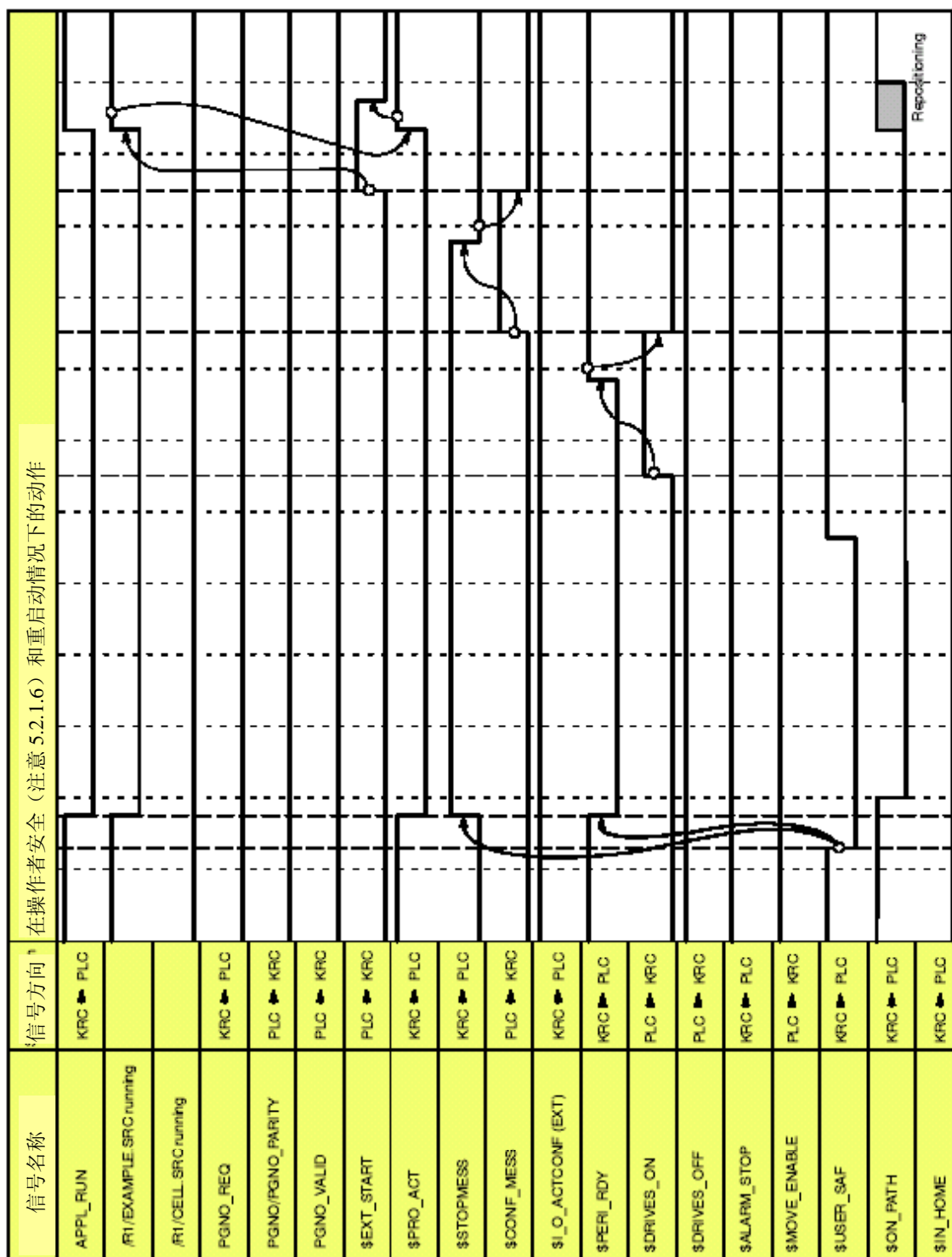


3.7.2 自动系统启动与通过\$EXT_START 确认的程序号的标准操作

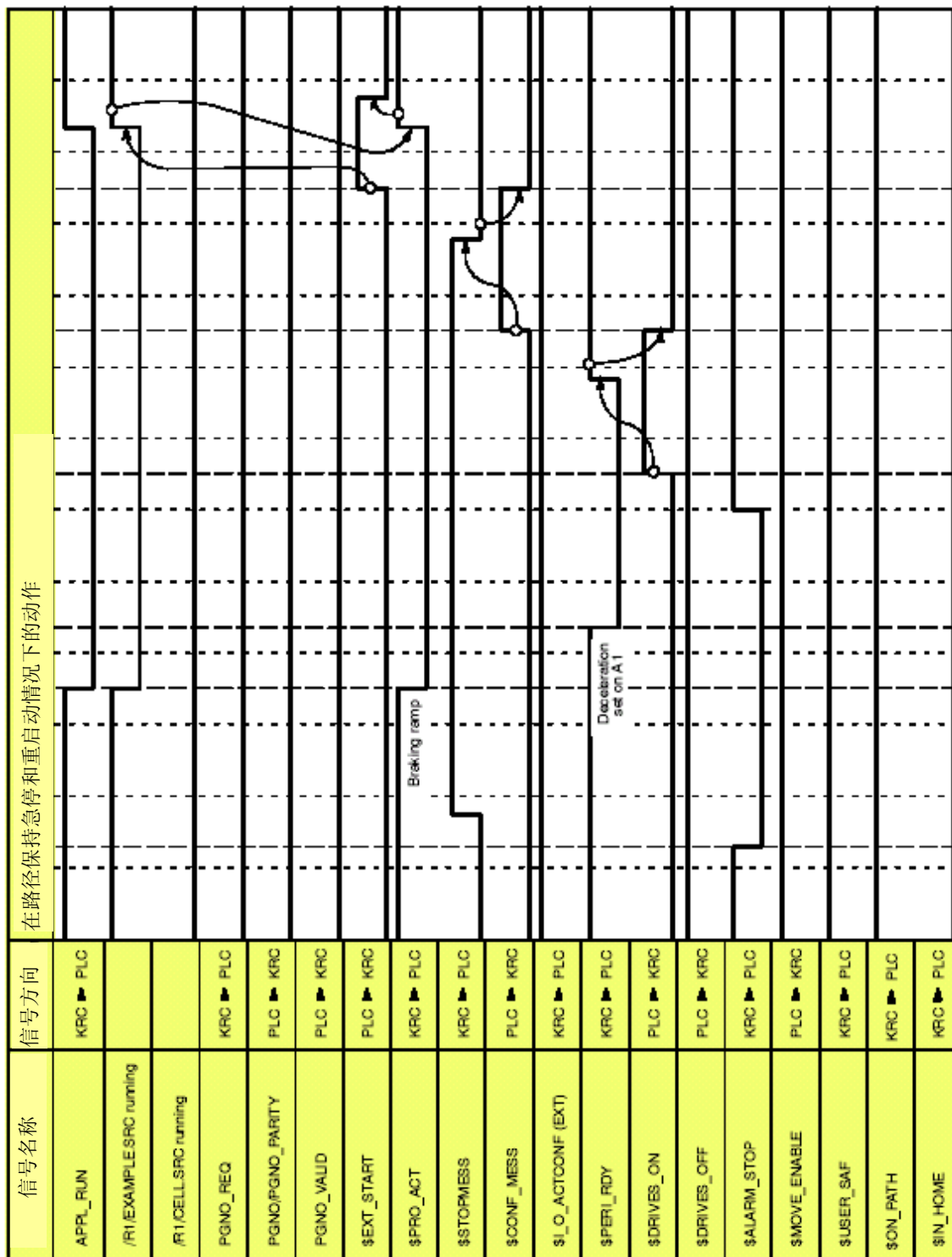


配置

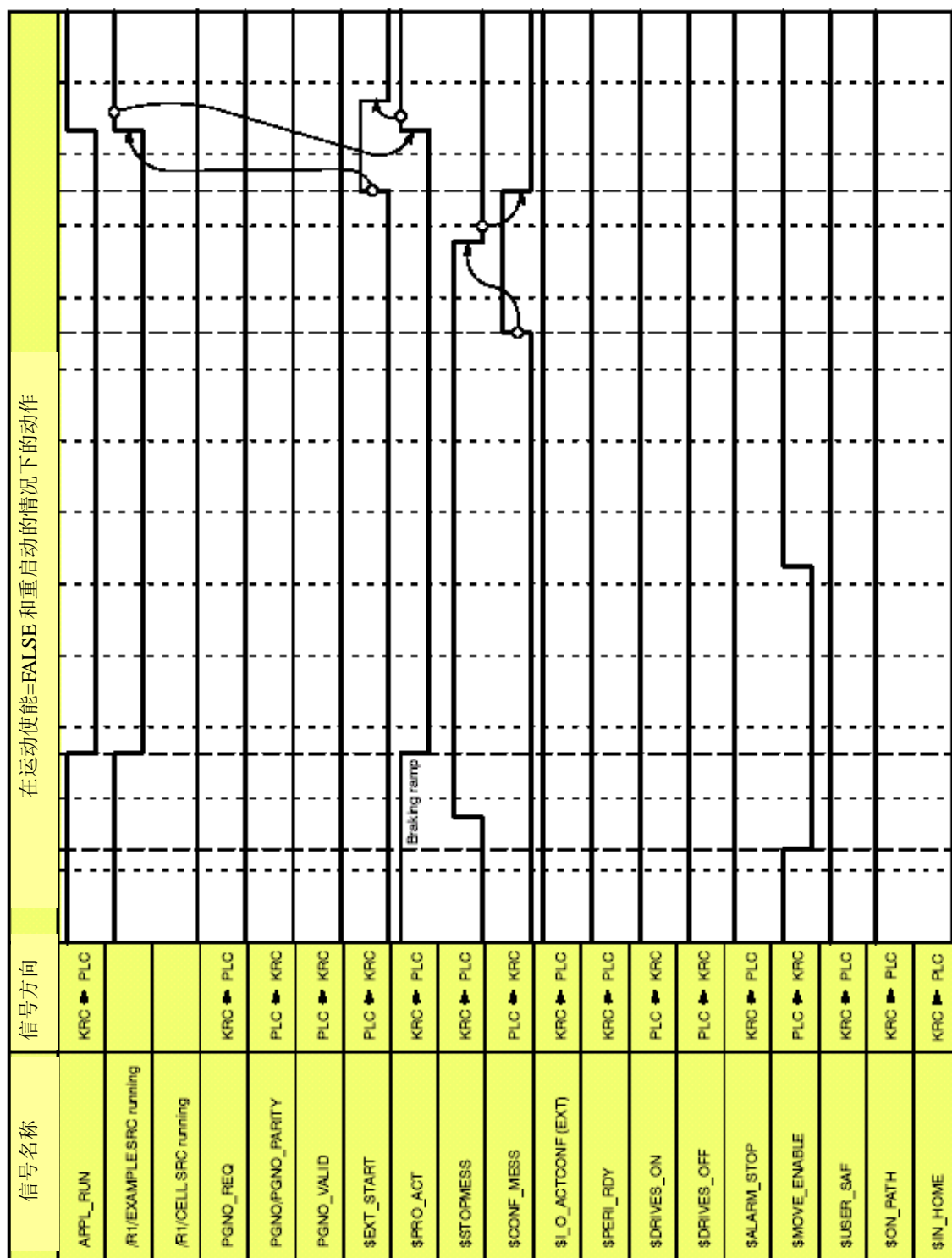
3.7.3 动态制动后的重新启动（操作者安全与重新启动）



3.7.4 路径保持急停后的重新启动



3.7.5 运动使能后的重新启动



3.7.6 用户STOP后的重新启动

信号名称	信号方向	在用户 STOP 和重新启动情况下的动作
APPL_RUN	KRC → PLC	
/R1/EXAMPLESRC running		
/R1/CELLSRC running		
PGNO_REQ	KRC → PLC	
PGNO/PGNO_PARITY	PLC → KRC	
PGNO_VALID	PLC → KRC	
\$EXT_START	PLC → KRC	
\$PRO_ACT	KRC → PLC	Programmed user STOP
\$STOPMESS	KRC → PLC	
\$CONF_MESS	PLC → KRC	
\$I_O_ACTCONF(EXT)	KRC → PLC	
\$PERI_RDY	KRC → PLC	
\$DRIVES_ON	PLC → KRC	
\$DRIVES_OFF	PLC → KRC	
\$ALARM_STOP	KRC → PLC	
\$MOVE_ENABLE	PLC → KRC	
\$USER_SAF	KRC → PLC	
\$ON_PATH	KRC → PLC	
\$IN_HOME	KRC → PLC	

3.8 更多的信息

3.8.1 被动停止后的重新启动

如果操作方式未改变从KCP进行了被动停止，必须在KCP上进行确认错误信息“A1370KCP:PASSIVESTOP”。然后使用外部启动可以继续程序。



如果操作方式改变，从KCP进行了被动停止，机器人必须重新手动定位。

3.8.2 逐步执行程序

通常，仅程序运行方式“#G0”允许在“外部自动”方式下。对于逐步操作方式，操作方式“#MSTEP”是有效的。

对此，必须在文件“C:\KRC\Roboter\Init\Progress.ini”中修改下列的行：

[FEATURES]

.

.

MSTEP_IN_EXT=TRUE

.

.

在下次系统重新引导后该选项有效。可以通过按状态键在程序运行方式之间切换。

3.8.3 返回编程路径的速度

如果机器人已离开编程路径，将降低速度移回到这个点。如果机器人执行了一个BC0运行同样适合。执行定位的速度对应手动移动的速度。

对此的前提是在文件“C:\KRC\Roboter\Init\Progress.ini”中修改下列的行：

[FEATURES]

.

.

SLOW_BCO_EXT=TRUE

.

.

要求仅在“C:\KRC\Roboter\KRC\R1\MaDa\\$Machine.dat”文件中可以修改的变量\$RED_VEL_AXC[1]...[12]来降低轴的移动速度。



可能的话，避免修改变量“\$RED_VEL_AXC[1]...[12]”。将不承担责任对于修改变量导致的损坏或停机。

3.9 配置举例

3.9.1 声明

- 程序号作为二进制数被发送。--程序号是7位并且在输入1接收。
- 校验位在输入8并且检查奇校验。
- 通过输出1信号的上升沿指示新程序号的请求。
- 主计算机使用输入9上升沿通讯程序号的存在。
- 通过输出2与主计算机通讯程序运行的细节。
- 如果I/O接口激活，主计算机通过输出3指示。
- 主计算机使用输入10执行外部启动。
- 通过输出4与主计算机通讯组错误。
- 主计算机通过输入11确认错误。



文件“C:\KRC\Roboter\KRC\R1\System\\$CONFIG.DAT”中必要的输入

(配置举例)

PGNO=0	程序号的预指定
PGNO_TYPE=1	程序号的数据格式：二进制数
PGNO_FBIT=1	程序号的首位：输入1
PGNO_LENGTH=7	程序号的长度：7位
PGNO_PARITY=-8	奇校验，校验位在输入8
PGNO_REQ=1	通过输出1请求新的程序号
PGNO_VALID=9	确认程序号已发送，发送到输入9
APPL_RUN=2	通知程序已被执行：输出2设定
PGNO_ERROR=0	错误标记的预指定

在文件\$MACHINE.DAT中的条目(配置举例)

\$EXT_START\$IN[10] ; 外部启动
 \$I_O_ACTCONF \$OUT[3] ; I/O接口激活
 \$STOPMESS \$OUT[4] ; 停止错误
 \$CONF_MESS \$IN[11] ; 组确认

接口分配(配置举例)

控制	信号名称	主计算机
----	------	------

\$IN[1]	PGNO Bit 1	A 20.0
\$IN[2]	PGNO Bit 2	A 20.1
\$IN[3]	PGNO Bit 3	A 20.2
\$IN[4]	PGNO Bit 4	A 20.3
\$IN[5]	PGNO Bit 5	A 20.4
\$IN[6]	PGNO Bit 6	A 20.5
\$IN[7]	PGNO Bit 7	A 20.6
\$IN[8]	PGNO_PARITY	A 20.7
\$IN[9]	PGNO_VALID	A 21.0
\$IN[10]	\$EXT_START	A 21.1
\$IN[11]	\$CONF_MESS	A 21.2
\$IN[12]	\$DRIVES_OFF	A 21.3
\$IN[13]	\$DRIVES_ON	A 21.4
\$IN[14]	\$MOVE_ENABLE	A 21.5
\$OUT[1]	PGNO_REQ	E 20.0
\$OUT[2]	APPL_RUN	E 20.1
\$OUT[3]	\$I_O_ACTCONF	E 20.2
\$OUT[4]	\$STOPMESS	E 20.3
\$OUT[5]	\$PERI_RDY	E 20.4
\$OUT[6]	\$PRO_ACT	E 20.5

3.10 信息

该节包含连接“外部自动”接口产生的错误信息说明。

信息号	信息文本	原因
P00:1	PGNO_TYPE 不正确的值 允许值 (1,2,3)	程序号的数据类型不正确地输入
P00:2	PGNO_LENGTH不正确的值的范围 $1 < \text{PGNO_LENGTH} < 16$	选择的程序号长度位数太高
P00:3	PGNO_LENGTH 不正确的值 允许值(4,8,12,16)	对于读取程序号选择的BCD格式，必须设置相应的位数。
P00:4	PGNO_FBIT 不正确的值 不在\$IN范围内	对于程序号的首位指定了值“0”或 不存在的输入
P00:7	PGNO_REQ 不正确的值 不在\$OUT范围内	对于请求程序号的输出指定了值“0”或 不存在的输出。
P00:10	传送错误 不正确的校验	当检查校验时检测到差异，产生一个传送错误。
P00:11	传送错误 不正确的程序号	对于在CELL.SRC检查结构中还未创建的执行分支，通过主计算机发送程序号
P00:12	传送错误 不正确的BCD编码	试图按BCD格式读取程序号导致一个无效的结果
P00:13	不正确的操作方式	I/O接口输出还未激活，如当前的系统变量\$I_O_ACTCONF是FALSE值。 由下列原因： 钥匙开关不在“Ext.”位置。 当前的信号 \$I_O_ACT是FALSE。
P00:14	在操作方式T1下移动到原点位置	机器人还未到达原点位置。
P00:15	不正确的程序号	对于“1 of n”超过一个输入设置。



Symbols

#INSIDE, 63, 77
 #INSIDE_STOP, 63, 77
 #OUTSIDE, 63, 77
 #OUTSIDE_STOP, 63, 77
 #PGNO_ACKN, 105
 #PGNO_FAULT, 106
 #PGNO_GET, 105
 #STEP1, 94
 \$_I_O_ACTCONF, 102
 \$ADAP_ACC, 94
 \$CONFIG.DAT, 125
 \$CURR_LIM, 92
 \$CURR_MAX, 92
 \$CURR_RED[x,x], 90, 91, 92
 \$CUSTOM.DAT, 94
 \$EXT_START, 102, 119
 \$EXT_START\$IN[], 126
 \$I_O_ACTCONF, 102
 \$I_O_ACTCONF \$OUT[], 126
 \$IN[x], 58
 \$INSIM_TBL[x], 58
 \$IOBLK_EXT, 58
 \$IOSIM_IN[], 58
 \$IOSIM_OPT, 58
 \$IOSIM_OUT[], 58
 \$MACHINE.DAT, 126
 \$NEARPATHTOL, 115
 \$OUT[x], 58
 \$OUT_NODRIVE, 58
 \$OUTSIM_TBL[x], 58
 \$POS_RET, 111, 115
 \$PRO_I_O[], 102
 \$RED_T1, 90
 \$ROBCOR.DAT, 94
 \$STOPMESS \$OUT[], 126
 \$TOOL, 64
 \$TORQ_DIFF, 95
 \$TORQ_VEL[], 91, 93
 \$TORQMON_COM, 95
 \$TORQMON_COM_DEF, 95
 \$TORQMON_TIME, 94
 \$TORQUE_AXIS, 91, 92

Numbers

5个原点位置, 59

A

存取口令, 34
 预先运行停止, 93
 ALARM_STOP, 114
 APPL_RUN, 113
 APPL_RUN , 125
 AUT, 114
 外部自动, 16, 99
 自动系统启动, 102
 Axis 1 soft, 91
 Axis 3 soft, 91
 定义转矩的轴, 92

B

BCD 植, 109
 二进制数, 109
 BOF 重新初始化, 43

C

天花板安装的机器人, 89
 CELL.SRC, 102, 103, 117
 修改口令, 34
 CHCK_MOVENA, 111
 碰撞监控, 94
 CONF_MESS, 111
 对于硬件警告的可配置输出, 54
 "配置" 菜单, 15
 配置接口, 99

D

DEF--line, 35
 详细观点, 36
 DRIVES_OFF, 112
 DRIVES_ON, 112
 动态制动, 120

E

编辑 "ConfigMon.ini", 48
 编辑器, 35
 急停, 121
 使能开关, 57
 ERR_FILE, 107
 ERR_TO_PLC, 116

转矩方式应用的例子, 90

专家, 25

EXT_ERR, 107

EXT_PGNO, 105, 106

EXT_PGNO.SRC, 117

EXT_START, 110

EXTERN, 114

外部风扇, 54

F

”文件” 菜单, 44

地面安装的机器人, 89

焦点, 99

牵制冷启动, 31

G

概述, 89

机械手, 16

H

硬件警告, 54

主计算机, 99

HOV, 24

I

I/O, 15, 99

I/O 驱动器, 17

I/O 接口, 102

I/O 仿真, 55

IN_HOME, 116

机器人的倾斜, 90

INSIDE_STOP, 64, 77

INT \$TORQUE_AXIS, 93

接口分配, 126

J

Jogging, 24

K

Kcpsaver, 13

KRL, 25

KUKA 长文本数据库, 49

KUKA 屏保, 11

L

语言, 32

限制, 89

有限的可见度, 36

Linebreak ON/OFF, 37

长文本, 49

M

宏分配列表, 49

手动 (jog)倍率, 24

混合的, 32

监控通道, 94

监控加工范围, 40

运动使能, 122

电机电缆监控, 54

MOVE_ENABLE, 111

N

NEAR_POSRET, 115

O

Office GUI 开/关, 38

ON_PATH, 114

外部自动操作方式, 57

操作者安全装置, 120

OUTSIDE_STOP, 64, 77

超程, 24, 66, 79

超程工作范围监控, 66

P

P00 模块, 105

被动停止, 124

口令, 25

PC 风扇, 54

PERI_RDY, 113

外围接口, 17

PGNO, 117, 125

PGNO_ERROR, 117, 125

PGNO_FBIT, 109, 125

PGNO_FBIT_REFL, 113

PGNO_LENGTH, 109, 125

PGNO_PARITY, 110, 125

PGNO_REQ, 105, 113, 125

PGNO_TYPE, 109, 125

PGNO_VALID, 110, 118

POV, 24

PRO_ACT, 115

程序号确认, 118

程序倍率, 24

R

反应, 89, 92

REAL \$CURR_ACT[12], 92

REAL \$CURR_RED[12,2], 93

REFLECT_PROG_NR, 110

重新初始化, 43

响应时间, 94

重启动, 120, 124

危险, 89

S

下垂轴, 89

屏保, 11

信号说明, 109

模拟的输入/输出, 55

速度控制器输出, 90

状态键, 23

逐步执行程序, 124

STOPMESS, 113

提交编译器, 22

T

T1, 114

T2, 114

特定工艺的组织程序, 103

转矩限制, 94

转矩方式, 89

转矩方式不可能, 89

转矩方式可能, 89

U

User, 25

User 组, 25

User 级, 25

User 级, 25

User STOP, 123

USER_SAF, 114

USERTech 重新初始化, 43

V

转矩方式的变量, 92

返回编程路径的速度,
124

W

WAIT FOR \$IN[], 91

墙上安装的机器人, 89

工作范围监控, 61

关节轴, 92

关节轴齿轮装置, 89