



# FD 控制装置 操作说明书 机器人语言

第 5 版

- 在使用机器人之前，请详读本操作说明书，并请遵从所有关于安全事项与正文的指示。
- 关于本机器人的安装、操作、维修，请仅由接受过本公司机器人讲习的人员进行。
- 在使用本机器人的时候，必须遵守各个国家有关工业机器人的法律以及安全相关的法律条例。
- 务必将本操作说明书交付给实际操作的人员。
- 有关本操作说明书的不明之处以及有关本机器人的售后服务，请向记载在封底中的敝公司的各服务中心查询。

株式会社 不二越



# 目 录

## 第 1 章 概要

1.1	何为机器人语言 .....	1-1
1.1.1	概要 .....	1-1
1.1.2	特征及注意事项 .....	1-2
1.2	示教到再生的步骤 .....	1-4

## 第 2 章 机器人语言的语法

2.1	机器人语言程序示例 .....	2-1
2.2	行号码 .....	2-2
2.3	字符 .....	2-3
2.4	常数 .....	2-4
2.4.1	数值常数 .....	2-4
2.4.2	字符串常数 .....	2-4
2.4.3	姿势常数 .....	2-5
2.4.4	移位常数 .....	2-7
2.5	变量 .....	2-8
2.5.1	整数变量 .....	2-9
2.5.2	实数变量 .....	2-11
2.5.3	字符串变量 .....	2-12
2.5.4	定时器变量 .....	2-12
2.5.5	输入信号变量 .....	2-13
2.5.6	输出信号变量 .....	2-14
2.5.7	固定输入信号变量 .....	2-15
2.5.8	固定输出信号变量 .....	2-17
2.5.9	姿势变量 .....	2-18
2.5.10	移位变量 .....	2-19
2.6	任意变量 .....	2-20
2.6.1	任意变量的定义 .....	2-21
2.6.2	整数型(INTEGER 型) .....	2-25
2.6.3	实数型(SINGLE 型) .....	2-25
2.6.4	字符串型(String 型) .....	2-25
2.6.5	位置型(Position 型) .....	2-26
2.6.6	各轴角度型(角度)(Angle 型) .....	2-27
2.6.7	编码器型(Encoder 型) .....	2-28
2.6.8	转换命令 .....	2-29
2.6.9	位置获取命令(GET) .....	2-32
2.6.10	提取/代入命令(OPE) .....	2-34
2.6.11	任意变量监视器 .....	2-39
2.7	表达式与运算 .....	2-47
2.7.1	算术运算 .....	2-47
2.7.2	关系运算 .....	2-47

2.7.3	逻辑运算.....	2-48
2.7.4	字符串运算.....	2-48
2.7.5	姿势运算.....	2-48
2.7.6	一般函数.....	2-49
2.7.7	系统函数.....	2-51
2.7.8	优先顺序.....	2-54
2.8	语句.....	2-55
2.8.1	注释语句.....	2-56
2.8.2	标签语句.....	2-56
2.8.3	代入语句.....	2-56
2.8.4	流程控制语句.....	2-57
2.8.5	命令语句.....	2-60
2.9	用户程序.....	2-61
2.9.1	FN802 用户程序.....	2-63
2.9.2	FN803 退出用户程序.....	2-63
2.9.3	FN804 结束用户程序.....	2-64
2.9.4	FN805 返回用户程序.....	2-64
2.9.5	FN806 调用用户程序.....	2-65

## 第 3 章 程序的编辑

3.1	在个人电脑上编辑.....	3-1
3.1.1	编辑时的注意事项.....	3-1
3.1.2	将机器人语言程序拷贝到本控制装置.....	3-1
3.2	在悬式示教作业操纵按钮台上编辑.....	3-2
3.2.1	基本操作.....	3-2
3.2.2	便利的编辑功能.....	3-5
3.2.3	简单地输入基本命令.....	3-6
3.2.4	检索字符串.....	3-7
3.2.5	替换字符串.....	3-8
3.3	制作姿势文件.....	3-9
3.3.1	姿势文件的概要.....	3-9
3.3.2	制作姿势文件.....	3-10
3.3.3	修正姿势文件.....	3-12
3.3.4	显示姿势文件一览.....	3-12
3.3.5	(注意事项)姿势文件的更新.....	3-13
3.3.6	姿势文件保存命令.....	3-14

## 第 4 章 程序的编译

4.1	编译.....	4-1
4.2	反编译.....	4-6
4.3	语言转换的强制执行.....	4-7
4.4	语言转换强制执行的设定.....	4-8

## 第 5 章 命令

5.1	MOVEX (移动命令)	5-1
5.1.1	单机构的情况	5-1
5.1.2	多机构的情况	5-8
5.2	输出信号	5-9
5.3	输入信号	5-10
5.4	步骤转移(跳跃)/步骤调用	5-11
5.5	程序转移(跳跃)/程序调用	5-12
5.6	系统(单元)外启动/系统(单元)外调用	5-13
5.7	变量及运算	5-14
5.8	移位(移动)	5-15
5.9	姿势强制选择	5-16
5.10	坐标计算/姿势变量相关	5-17
5.11	用户任务	5-18
5.12	任意变量	5-19
5.13	用户程序	5-20
5.14	套接字通信	5-21
5.15	模拟输入输出	5-22
5.16	其他	5-23

## 第6章 机器人语言的使用方法

6.1	机器人语言使用上的注意事项	6-1
6.1.1	定位精度	6-1
6.1.2	使用机器人语言制作的程序的再生方法	6-1
6.1.3	检查运转	6-1
6.1.4	改写为编码器值的方法	6-1
6.1.5	姿势变量的记录	6-1
6.1.6	使用多机构时的注意事项	6-2
6.2	使用机器人语言可实现的用途	6-3
6.2.1	以机器人坐标值的数值指定目标位置	6-3
6.2.2	以数值指定动作距离	6-3
6.2.3	计算圆周轨迹	6-4
6.2.4	以数值移位作业程序	6-5
6.2.5	使用变量令1个作业程序对应多个工件	6-5
6.3	关于变量(内部变量、姿势变量)	6-6
6.3.1	关于内部变量	6-6
6.3.2	关于姿势变量	6-7
6.3.3	姿势变量的编辑方法	6-8
6.4	机器人语言的使用示例	6-9
6.4.1	读出机器人当前位置的方法1	6-9
6.4.2	读出机器人当前位置的方法2	6-9

6.4.3	使用用户坐标系的位置数据控制姿势变量.....	6-10
6.4.4	使用数值控制机器人的位置.....	6-11
6.4.5	使用数值指示外部轴的移动位置 .....	6-11
6.4.6	通过输入信号从外部接收数值 .....	6-12
6.4.7	使用从输入信号接收的数值对示教位置进行移位.....	6-13
6.4.8	(电弧焊)通过外部信号切换焊接条件、摆动条件的方法.....	6-14
6.4.9	(电弧焊)使用激光搜索检测长度不同的工件端 .....	6-16
6.4.10	(电弧焊)感应圆柱管内表面边缘的 3 个点求出管的中心位置.....	6-17
6.4.11	(电弧焊)生成沿圆柱管内表面的专用坐标系 .....	6-19
6.4.12	可以向姿势变量代入任意坐标系的指令值的应用命令.....	6-20
6.4.13	读出机器人当前位置并向外部输出 .....	6-21

# 第1章 概要

---

本章将对机器人语言的概要进行说明。

1.1 何为机器人语言 .....	1-1
1.1.1 概要 .....	1-1
1.1.2 特征及注意事项 .....	1-2
1.2 示教到再生的步骤.....	1-4





## 1.1 何为机器人语言

### 1.1.1 概要

所谓“机器人语言”是指本机器人控制装置用的编程语言。将机器人的移动命令（MOVEX）及输入信号等待命令（WAITI）等应用命令组合使用，可制作（示教）机器人的作业程序。由于也可使用各种函数及计算公式等，因此适用于制作通常的示教操作中需要较困难的、复杂的条件分支及坐标计算的程序。此外，也可使用机器人语言独特的命令组，制作用户专用的监视器画面。

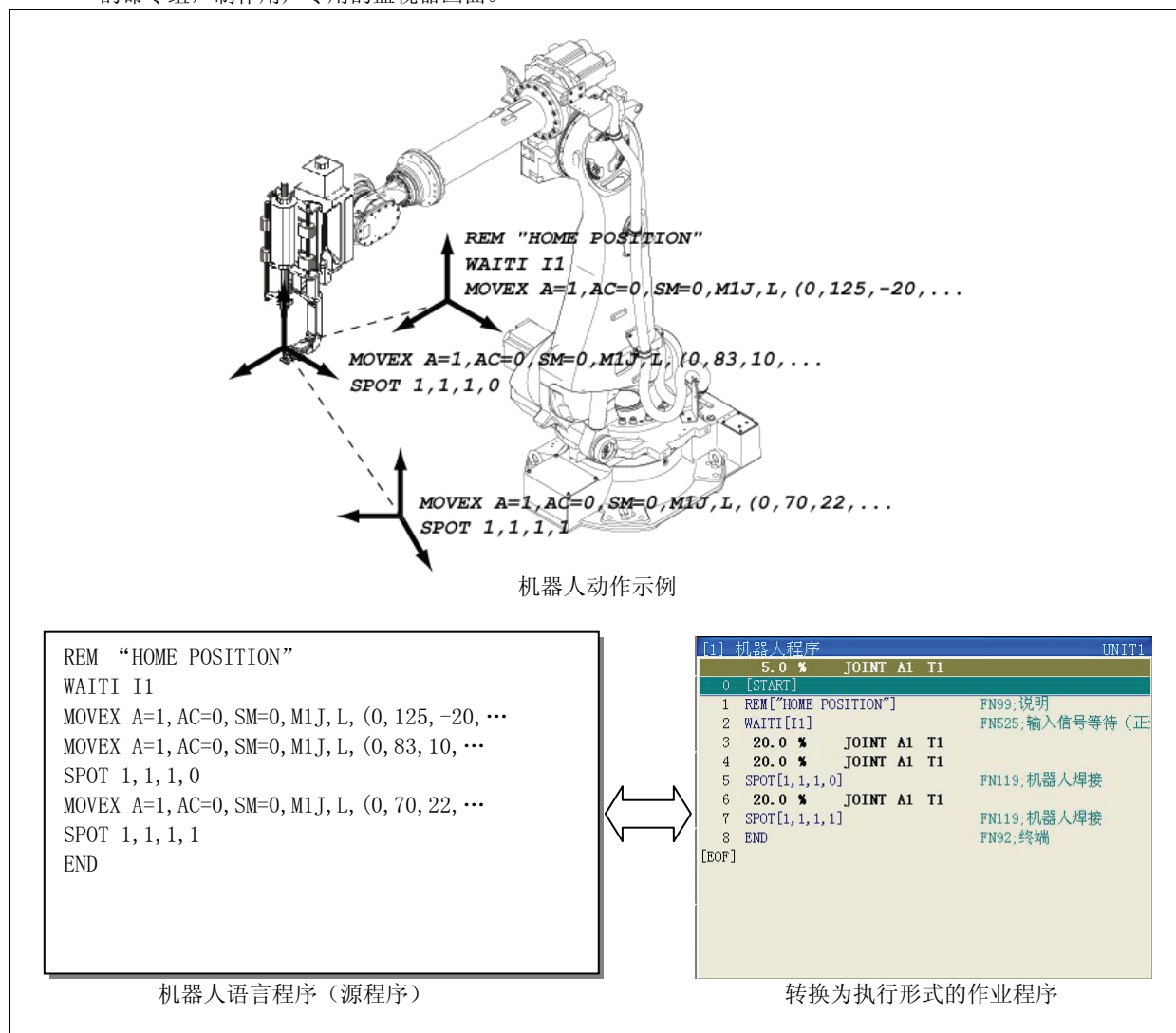


图 1.1.1 使用机器人语言制作的作业程序示例

关于机器人语言的语法问题，请参照第 2 章。此外，在尚未熟悉机器人语言形式前，请参考操作说明书“基本操作篇”，先使用通常的方法示教几个简单的作业程序，并尝试将其“反编译”为机器人语言形式。反编译的方法请参照第 4 章。



提示

本操作说明书中未详细说明各应用命令。各应用命令的详情请参照本控制装置的悬式示教作业操纵按钮台中的在线帮助或操作说明书中的“命令参考”。



提示

关于用户专用的监视器画面的制作方法，请参照操作说明书的“用户任务”。


1.1.2 特征及注意事项

- 机器人语言的源程序为纯文本形式。可使用您的个人电脑的文本编辑器及本控制装置的“ASCII 文件编辑功能”制作源程序。
- 例如，机器人型号为 SRA166 及 NB4 时，机器人的作业程序（执行形式）、机器人语言程序（源程序）的文件名如下所示。本控制装置无法识别不符合该命名规则的文件。

机器人语言程序（源程序）	SRA166-1-A. 001
	NB4-02-A. 001
作业程序（执行形式）	SRA166-1. 001
	NB4-02. 001

需确认可用于源程序的文件名时，请通过<维修>－[7 文件操作]－[2 显示一览]画面，打开“PROGRAM”文件夹。如作业程序已经保存到内部存储器中，将显示该文件名。

附带有“-A”的文件名，即为源程序的文件名。



SRA166-01 的源程序

名称	属性	尺寸	更新日期
SRA166-1-A. 001		1579	13/10/25 12:58
SRA166-1-A. 010		1579	13/10/25 12:58
SRA166-1-A. 100		1579	13/10/25 12:58
SRA166-1-A. 101		1579	13/10/25 12:58

NB4-02 的源程序

名称	属性	尺寸	更新日期
NB4-02-A. 001		1579	13/10/25 12:58
NB4-02-A. 010		1579	13/10/25 12:58
NB4-02-A. 100		1579	13/10/25 12:58
NB4-02-A. 101		1579	13/10/25 12:58
NB4-02-A. 102		1579	13/10/25 12:58

- 机器人语言以 JIS B9439 所规定的 SLIM 语言(Standard Language for Industrial Manipulator)为基准。
- 机器人语言的源程序，需要事先使用本控制装置转换（编译）为可执行形式的程序。机器人语言形式与执行形式可双向转换（编译 / 反编译）。并且，通常的示教操作所示教的作业程序从最初即以执行形式记录到内部存储器中，无需编译。
- 使用机器人语言制作的程序与通过通常的示教操作制作的程序可同时存在。也可仅使用机器人语言制作复杂的计算及条件分支处理，并使用通常的方法从这些机器人语言程序中调用示教的机器人程序。

- 在机器人语言中，支持 2 种移动命令的示教方法。分别为在移动命令内直接描述位置信息的方法及调用姿势变量的方法。姿势变量可通过事先的位置记录操作来制作，也可通过数值代入命令来制作。可用“姿势文件”的形式综合管理通过位置记录操作制作的多个姿势变量。详情请参照第 3 章。

#### 程序中描述姿势常数的方法 “2.4.3 姿势常数”

例 1: MOVEX-X 形式

MOVEX A=1, AC=0, SM=0, **M1X**, L, **(1800, 0, 2000, 0, -90, -180)**, R=5. 0, H=1, MS

例 2: MOVEX-J 形式

MOVEX A=1, AC=0, SM=0, **M1J**, L, **(0, 90, 0, 0, 0, 0)**, R=5. 0, H=1, MS

#### 使用姿势变量进行指定的方法 “2.5.9 姿势变量” “3.3 制作姿势文件”


MOVEX A=1, AC=0, SM=0, M1X, P, **P1**, R=5. 0, H=1, MS

- 转换为执行形式后，即使是使用机器人语言示教的移动命令，也可通过悬式示教作业操纵按钮台按照通常的操作进行位置修正。只是在这种情况下，无论原来的移动命令的形式如何，机器人的位置信息将被转为（与通过通常的位置记录操作制作时相同的）编码器值形式，还请注意。
- 即使是使用机器人语言制作的执行形式的程序，也与通常的作业程序相同，可进行步骤的追加及删除。但是，鉴于程序维护方面的理由，推荐修正原有的源程序，进行重新编译。此外，进行源程序的细微修正时，使用本控制装置的“ASCII 文件编辑功能”较为方便。详细内容请参照第 3 章。

## 1.2 示教到再生的步骤

### 作业的概要

1 请使用个人电脑的文本编辑器或本控制装置的“ASCII 文件编辑功能”制作源程序。


 “3.1 在个人电脑上编辑”


2 例如，机器人的型号为 SRA166-01, NB4-02 时，请用以下文件名保存到 USB 存储器中。


**SRA166-1-A.001（源程序的文件名）**

**NB4-02-A.001（源程序的文件名）**

- “SRA166-1” “NB4-02” 表示机器人型号。
- “-A” 是表示程序为机器人语言程序的识别符。
- 扩展名的“001”为程序号码。最大为 9999。

 “1.1.2 特征及注意事项”的文件名的项目

 “3.1.2 将机器人语言源程序拷贝到本控制装置”

 操作说明书“基本操作篇”第 6 章“文件操作”


3 请将制作好的源程序文件从 USB 存储器拷贝到本控制装置的内部存储器中。

通过<维修>—[7 文件操作]—[1 拷贝]画面实施。

 操作说明书“基本操作篇”第 6 章“文件操作”

4 请“编译”源程序，转换为“执行形式”。

通过<维修>—[9 程序转换]—[8 语言转换]画面实施。

 “4.1 编译”


编译正常结束后，将生成执行形式的程序文件。

在这一示例中，文件名如下所示：

**SRA166-1.001（执行形式的文件名）**

**NB4-02.001（执行形式的文件名）**

5 所生成的执行形式的程序，可按与通常的作业程序一样的顺序进行前进检查、再生。

 操作说明书“基本操作篇”第 4 章及第 5 章

6 机器人的动作出现问题时，请修正源程序进行重新编译。修正时，使用“ASCII 文件编辑”功能较为方便。

 “3.2 在悬式示教作业操纵按钮台上编辑”



在机器人语言中使用了整数变量等时，可使用“监视器功能”对这些值进行确认。“监视器功能”可通过<维修>菜单来执行。



使用机器人语言时，由于可使用计算公式等自由生成机器人的坐标，因此在进行动作确认时请充分注意。如机器人发生意想不到的动作，有可能造成重大事故。

## 第2章 机器人语言的语法

---

本章说明使用机器人语言制作程序时的语法。

2.1 机器人语言程序示例 .....	2-1
2.2 行号码 .....	2-2
2.3 字符 .....	2-3
2.4 常数 .....	2-4
2.4.1 数值常数 .....	2-4
2.4.2 字符串常数 .....	2-4
2.4.3 姿势常数 .....	2-5
2.4.4 移位常数 .....	2-7
2.5 变量 .....	2-8
2.5.1 整数变量 .....	2-9
2.5.2 实数变量 .....	2-11
2.5.3 字符串变量 .....	2-12
2.5.4 定时器变量 .....	2-12
2.5.5 输入信号变量 .....	2-13
2.5.6 输出信号变量 .....	2-14
2.5.7 固定输入信号变量 .....	2-15
2.5.8 固定输出信号变量 .....	2-17
2.5.9 姿势变量 .....	2-18
2.5.10 移位变量 .....	2-19
2.6 任意变量 .....	2-20
2.6.1 任意变量的定义 .....	2-21
2.6.2 整数型(INTEGER 型) .....	2-25
2.6.3 实数型(SINGLE 型) .....	2-25
2.6.4 字符串型(String 型) .....	2-25
2.6.5 位置型(Position 型) .....	2-26
2.6.6 各轴角度型(角度)(Angle 型) .....	2-27
2.6.7 编码器型(Encoder 型) .....	2-28
2.6.8 转换命令 .....	2-29
2.6.9 位置获取命令(GET) .....	2-32
2.6.10 提取/代入命令(OPE) .....	2-34
2.6.11 任意变量监视器 .....	2-39
2.7 表达式与运算 .....	2-47
2.7.1 算术运算 .....	2-47
2.7.2 关系运算 .....	2-47
2.7.3 逻辑运算 .....	2-48
2.7.4 字符串运算 .....	2-48
2.7.5 姿势运算 .....	2-48
2.7.6 一般函数 .....	2-49

2.7.7	系统函数 .....	2-51
2.7.8	优先顺序 .....	2-54
2.8	语句 .....	2-55
2.8.1	注释语句 .....	2-56
2.8.2	标签语句 .....	2-56
2.8.3	代入语句 .....	2-56
2.8.4	流程控制语句 .....	2-57
2.8.5	命令语句 .....	2-60
2.9	用户程序 .....	2-61
2.9.1	FN802 用户程序 .....	2-63
2.9.2	FN803 退出用户程序 .....	2-63
2.9.3	FN804 结束用户程序 .....	2-64
2.9.4	FN805 返回用户程序 .....	2-64
2.9.5	FN806 调用用户程序 .....	2-65

## 2.1 机器人语言程序示例

### 机器人语言程序


```

100 'SHIFT
110 USE 100
120 FOR V1%=1 TO 7
130 R1=(10,1,0,0,0,0)
140 P100=P[V1%]+R1
150 MOVEX A=1,M1X,P,P100,R=5.0,H=1,MS
160 NEXT
170 END

```


#### 第 100 行：注释语句

在行首附单引号（'），即成为说明语句。  
也可使用 REM 代替单引号。

 “2.8.1 注释语句”


#### 第 110 行：USE 命令

选择姿势文件 100 号。  
以后所使用的“姿势变量”将从这一姿势文件中获取。

 “2.5.9 姿势变量” “3.3 制作姿势文件”


#### 第 120 行：FOR 命令

在流程控制语句中，也有行间的跳跃（GOTO）及条件分支（IF）等。  
“V1%”为可代入整数的变量。在这里作为循环计数器使用。

 “2.8.4 流程控制语句”


#### 第 130 行：代入语句

向移位变量“R1”中直接代入移位置。

 “2.5.10 移位变量” “2.8.3 代入语句”


#### 第 140 行：代入语句

进行姿势变量“P[V1%]”与移位变量“R1”的运算，代入到姿势变量“P100”中。

 “2.5.9 姿势变量” “2.5.10 移位变量”


#### 第 150 行：MOVEX 命令

使用计算得出的姿势变量 P100 执行移动命令（MOVEX 命令）。  
至此，机器人开始动作。

 “2.5.9 姿势变量”

#### 第 160 行：NEXT 命令

与 FOR 语句成对的流程控制语句。在这里，跳跃到第 120 行。

 “2.8.4 流程控制语句”

#### 第 170 行：END 命令

结束程序。请务必在程序末尾附加 END 命令。

※行号码可省略。



 “2.2 行号码”

图 2.1.1 机器人语言的语法（例题）

## 2.2 行号码

“行”是组成机器人语言程序的最小单位，以每 1 行为单位执行。

1 行最多 **254 个字符**，1 个程序中最多可描述 **9999 行**，由注释语句/标签/代入语句/流程控制语句/命令语句中的任意 1 个语句组成（ 请参照“2.8 语句”）。

1 行中不允许有 2 个以上语句的表述（“FOR V1%=0 TO 100, NEXT”等）。



### 重点

- 1 行中不能描述 2 个以上的语句。
- 禁止在 1 个语句中途插入换行代码。
- 1 个语句最多为 254 个字符。  
如写入 255 个字符以上时，将出现编译错误。

行号码是用于显示程序的执行位置，可在 1~9999 的范围内指定，作为行跳跃等的跳往端使用。  
分配行号码时，请按照程序开始时小，后半部分渐渐增大的顺序。



### 重点

可省略行号码。  
此时行跳跃等的跳往端请使用标签。  
省略行号码时，请不要在整个程序中描述行号码。  
有时带行号码，有时不带行号码，则程序将无法正确执行。



### 重点

如执行反编译，行号码将改变。



## 2.3 字符

机器人语言中可使用的字符如下表所示。

表 2.1 机器人语言中可使用的字符

项目	可使用字符
英文数字	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9
符号	! " ' # \$ % & ( ) * + - . , / : ; = < > ? @ ` [ ] ¥ ^ { } ~   _
空白	(半角空格) (Tab)

重点

不区分英文字符大小写。  
除说明及字符串变量以外，均请使用半角字符。

## 2.4 常数

常数包含如下几个种类。  
下述所示的常数，即使主电源切断，也将作为停电保持数据被保存。

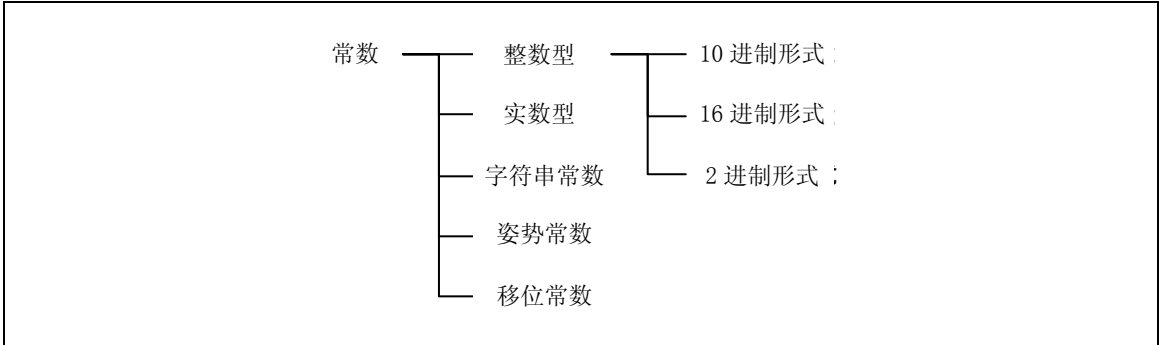


图 2.4.1 常数的种类

### 2.4.1 数值常数

数值常数是执行算数运算时所需的数值数据。  
整数形式与实数形式在内部自动判断。

- 【例】
- -10            整数
  - 1.23-E12      实数
  - &H1F3        16 进制形式
  - &B10101      2 进制形式
  - 90DEG        使用 deg（度）表示的角度值（实数）

整数的范围为-2147483648～+214748367；实数的范围为-1.0+E38～+1.0+E38。

### 2.4.2 字符串常数

将使用双引号（”）括起来的英文数字、符号称为字符串。

- 【例】
- PRINT #0, “ABCDE”

### 2.4.3 姿势常数

姿势常数是表示机器人的位置、姿势的常数。与移动命令 MOVEX 配合使用。姿势常数中包含如下 3 种形式。

- MOVEX-X 形式 以 (X, Y, Z, r, p, y) 表现工具前端的位置与姿势的形式
- MOVEX-J 形式 以各轴的角度或位置表现机器人姿势的形式
- MOVEX-E 形式 以各轴的编码器值表现机器人姿势的形式

#### MOVEX-X 形式

以 (X, Y, Z, Roll, Pitch, Yaw) 表现工具前端(TCP)的位置与姿势。“机构指定”为“M\*X” (“\*”为机构号码)。基准点为该机器人(机构)的机械坐标系原点。

(例) MOVEX A=1, M1X,P,(1200,0,1800, 0, 0,-180),R= 10.0,H=1,MS

表 2.2 工具前端的位置与姿势

参数	MOVEX-X
X	工具前端的 X 坐标 [mm]
Y	工具前端的 Y 坐标 [mm]
Z	工具前端的 Z 坐标 [mm]
r(Roll)	工具的横摇角 [deg]
p(Pitch)	工具的纵摇角 [deg]
y(Yaw)	工具的艏向角 [deg]

请以 (X, Y, Z) 指定从机械坐标系原点看时的工具前端的坐标。此外, 请以 (r, p, y) (横摇角、纵摇角、艏向角) 指定机械坐标系基准的工具坐标系的角度(方向)。并且, 如旋转顺序错误, 工具方向将改变, 还请注意。

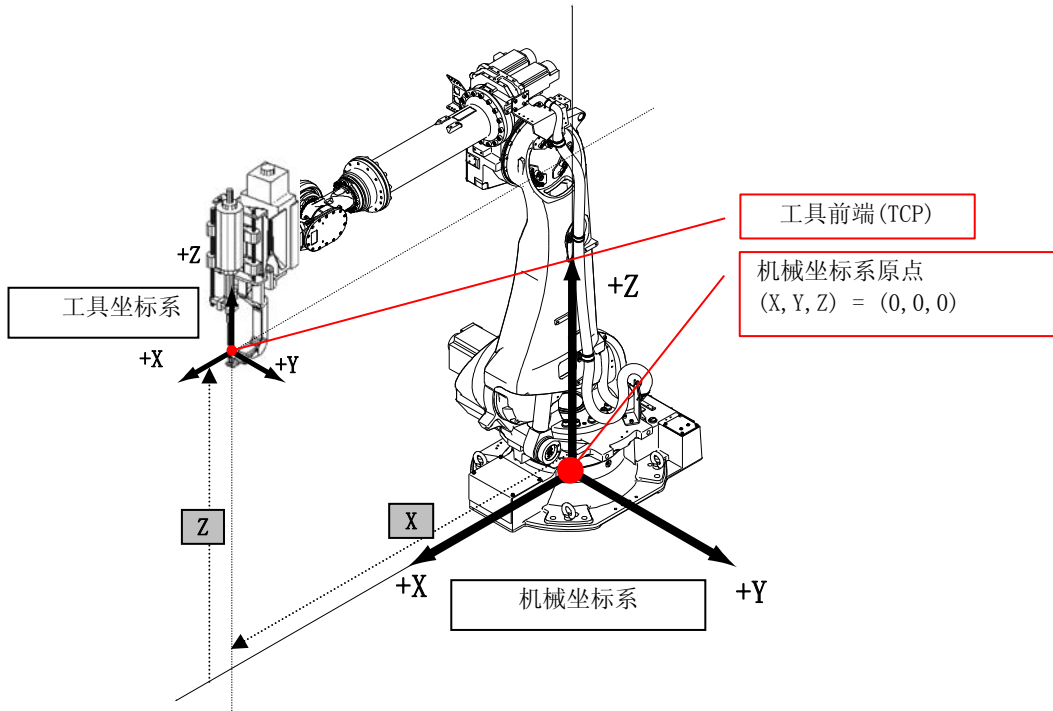


图 2.4.2 机械坐标系与工具坐标系



- 由于使用机械坐标系基准描述工具前端位置, 即使通过使用这一方式所制作的移动命令来变更工具常数的设定, 实际上工具前端所到达的位置并不改变。但是此时各轴角度将发生变化, 请充分注意以防发生干扰等问题。
- 不使用 CONF 指定时, 机器人有时会出现意想不到的姿势。CONF 指定的详情请参照第 5 章。此外, 推荐至少最开始时执行的移动命令要以后述的 MOVEX-J 形式来描述。(因为 MOVEX-J 形式可单独决定机器人的姿势)

MOVEX-J 形式

以角度[deg]或位置[mm]来表现机器人各轴的位置。“机构指定”为“M\*J”（“\*”为机构号码）。要素数根据机构不同而有差异。SRA 系列及 NB/NV 系列等通常的垂直多关节机器人为 6 轴。1 个机构的最大轴数为 6。

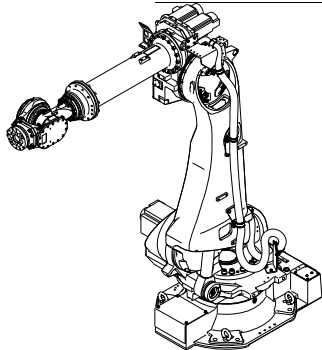
表 2.3 各轴角度

参数	MOVEX-J
J1	J1 轴角度 [deg] 或 位置[mm]
J2	J2 轴角度 [deg] 或 位置[mm]
J3	J3 轴角度 [deg] 或 位置[mm]
J4	J4 轴角度 [deg] 或 位置[mm]
J5	J5 轴角度 [deg] 或 位置[mm]
J6	J6 轴角度 [deg] 或 位置[mm]

通常的 6 轴机械手时，请描述 6 个值。

6 轴机器人（SRA）的示例

（例）MOVEX A=1,M1J,P,(0,90,0,0,0,0),R= 5.0,H=1,MS



有伺服焊枪及走行轴等追加轴时，请参考以下示例。

有走行轴的 6 轴机器人示例

（例）MOVEX A=1,M1J,P,(0,90,0,0,0,0),R= 5.0,H=1,MS,M2J,P,(100),R= 10.0,H=1

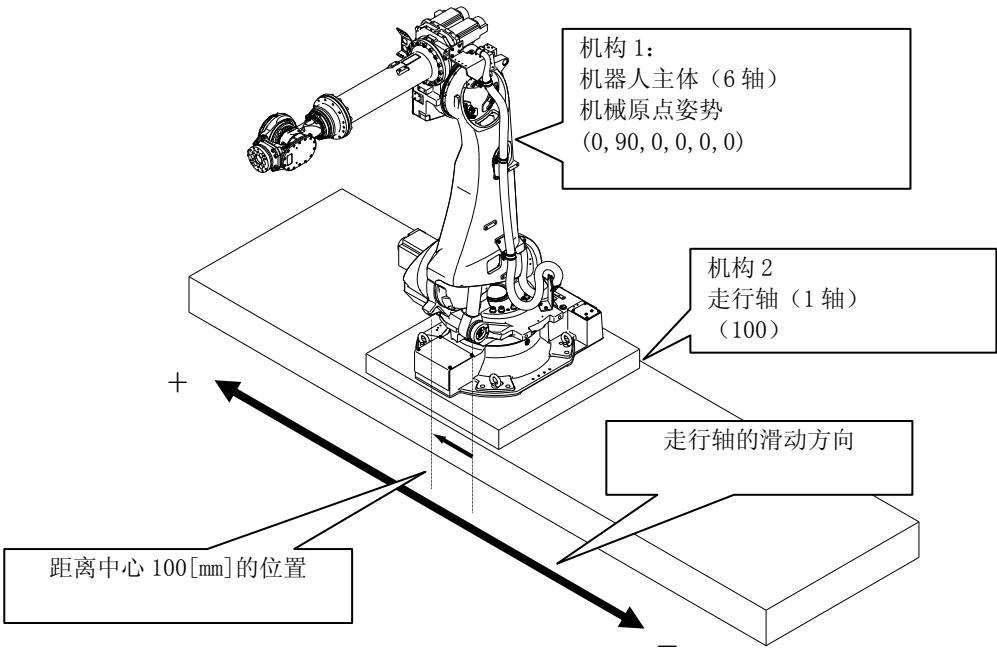


图 2.4.3 有走行轴的机器人

在这一示例中，机构 1 为机械原点姿势、机构 2 为 100[mm]的位置。

MOVEX-E 形式

以 16 进制的编码器值表现机器人各轴的位置。16 进制写作 “&H80000” 的格式。

(例): MOVEX A=1, **M1E,P,(&H80000,&H80000,&H80000,&H80000,&H80000,&H80000)**,R= 10.0,H=1,MS

存在 2 个以上的机构时，请参考 MOVEX-J 的示例。

表 2.4 各轴的编码器值

参数	MOVEX-E
E1	J1 轴编码器值 [16 进制]
E2	J2 轴编码器值 [16 进制]
E3	J3 轴编码器值 [16 进制]
E4	J4 轴编码器值 [16 进制]
E5	J5 轴编码器值 [16 进制]
E6	J6 轴编码器值 [16 进制]



以该形式制作的 MOVEX 命令编译后将转换为与使用悬式示教作业操纵按钮台上的“记录”键直接记录的移动命令相同的数据形式。

2.4.4 移位常数

移位常数是相对于机器人的姿势相加平移量时使用的常数。  
使用 (X, Y, Z, r, p, y) 的描述格式。

表 2.5 移位常数

参数	移位量
X	X 轴方向的移位 [mm]
Y	Y 轴方向的移位 [mm]
Z	Z 轴方向的移位 [mm]
r(Roll)	绕 Z 轴旋转的旋转移位 [deg]
p(Pitch)	绕 Y 轴旋转的旋转移位 [deg]
y(Yaw)	绕 X 轴旋转的旋转移位 [deg]

## 2.5 变量

将在程序中用来存储所使用的值的领域称为变量。变量用于运算参照等。变量名是事先预约好的，不能自由使用变量名。  
变量种类如下所示。

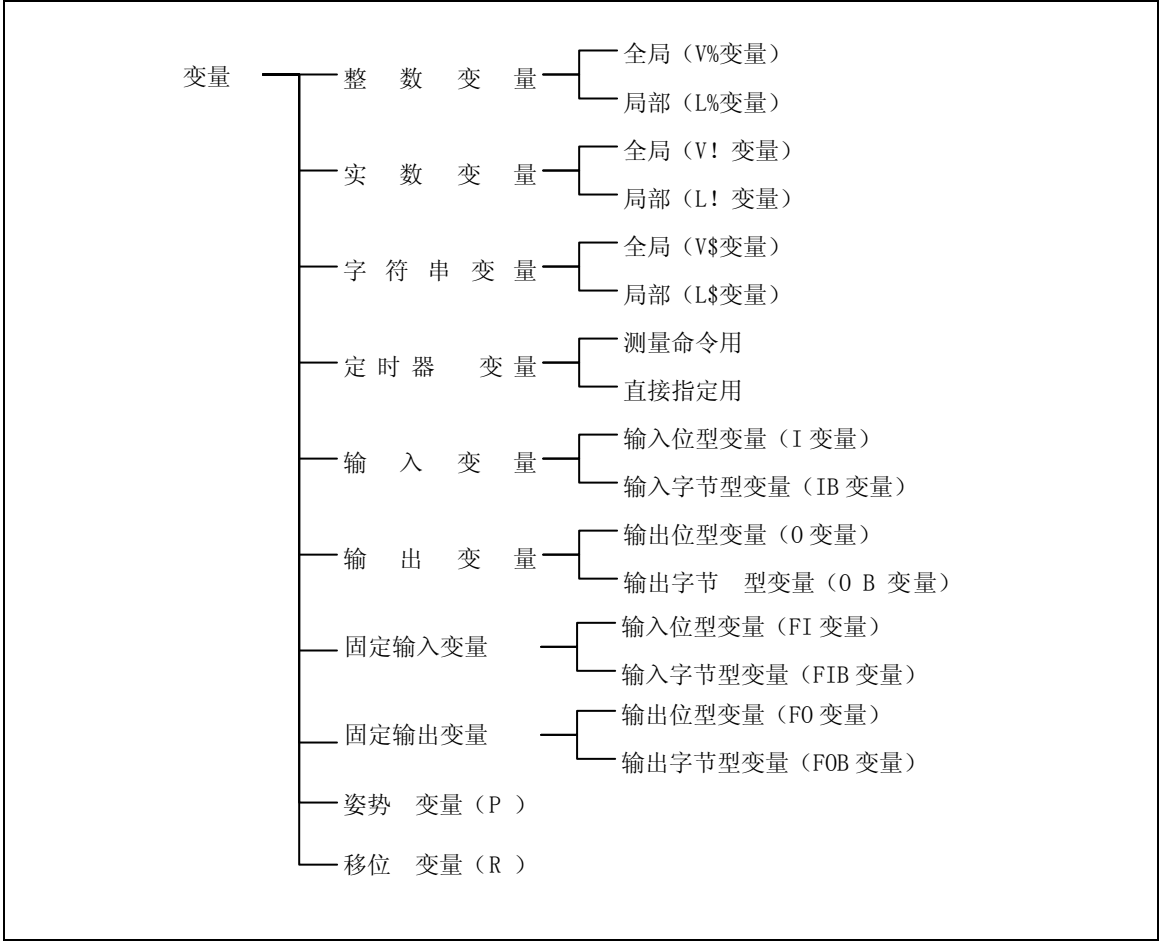


图 2.5.1 变量的种类

表 2.6 全局变量与局部变量

全局变量	可从所有单元中参照。 是以 V 开始的变量。
局部变量	分别存在于各个单元中的变量，不能从其他单元中参照。是以 L 开始的变量。

2.5.1 整数变量

“整数变量”处理不含小数点的数值。

表 2.7 整数变量

格式	Vn%, V%[n]                      n=1~250, 301~500 (也可使用变量。) Ln%, L%[n]                      n=1~200, 301~500 (也可使用变量。) ※ V1%与 V%[1]表示相同的变量领域。
范围	-2147483648~+2147483647
示例	V1%=V2%+1 L%[1]=10 CALLN 10, V1%, 10 JMPN 10, V1%, 20
保存	所有的全局整数变量在主电源切断时也可作为停电保持数据得以保存。而所有的局部整数变量均不保存。



- 全局整数变量的 201~250 是进行 PLC 内部的整数变量与整数值收发的变量。如向上述全局变量中写入值，PLC 内部的整数变量将被写入值。此外，如读取上述全局变量，将读取 PLC 内部的整数变量的值。全局整数变量与 PLC 内部的整数变量的对应关系如下表所示。
- 写入上述 201~250 的全局整数变量中时，在变更反映到 PLC 内部的整数变量中之前，将发生延迟。详情请参照“V201%~V250%的全局整数变量使用时的注意事项”。

表 2.8 全局整数变量与 PLC 内部的整数变量的对应

全局整数变量号码	PLC 内部的整数变量号码
V201% 或 V%[201]	D0001
V202% 或 V%[202]	D0002
V203% 或 V%[203]	D0003
⋮	⋮
V248% 或 V%[248]	D0048
V249% 或 V%[249]	D0049
V250% 或 V%[250]	D0050

V201%~V250%的全局整数变量使用时的注意事项

V201%~V250%的全局整数变量在进行读取/写入时，对 PLC 内部变量的整数变量进行整数值的读取/写入。

因此，不能像原有的全局整数变量那样先执行。

此外，向 V201%~V250%的全局整数变量中设定值后，在设定值反映到 PLC 内部的整数变量中之前，将发生延迟。

反之，变更 PLC 内部的整数变量时，在变更反映到 V201%~V250%的全局整数变量中之前，将发生延迟。

另外，延迟的大小根据机器人的构成、使用方法等不同而有差异。

为防止上述延迟的影响，须按下示方法在 PLC 与机器人语言间采取连锁措施。

下示是使用 V201%~V250%的全局整数变量时的 PLC 内部变量收发时的连锁方法。

此外，请注意内部变量读取时与写入时的连锁方法不同。

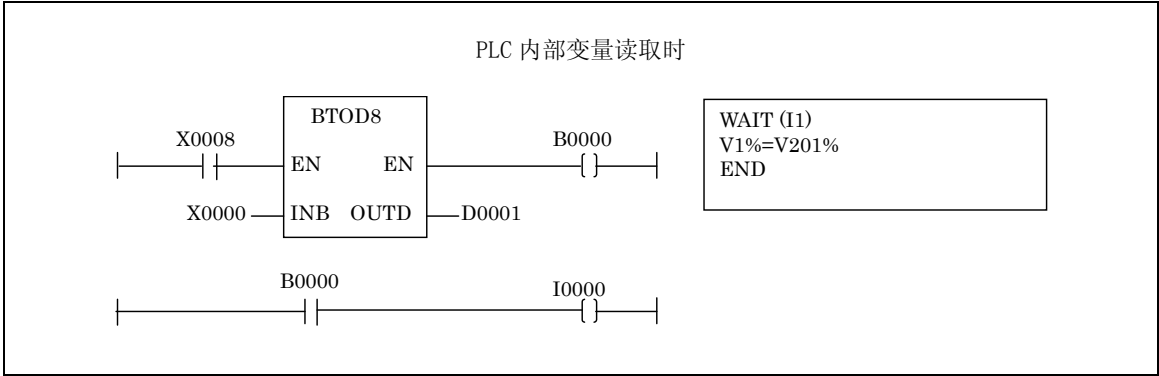


图 2.5.2 201~250 的全局整数变量使用示例

上图中，将本控制装置的标准输入 X0000~X0007 的 8Bit 作为整数值代入到全局整数变量 V1%中。

上图中，外部输入按照如下顺序被输入。

- ①将欲设定到本控制装置的整数值输入到 X0000~X0007 中
- ②显示输入已确定的 X0008 变为 ON

上图中，由于来自外部的信号按照 X0000~X0007→X0008 的顺序被输入，因此，I1 (PLC 时为 I0000) 为 ON 时，D0001 确定。

因此，PLC 与机器人语言间的联锁可通过 I1 转为 ON 后得以实现。

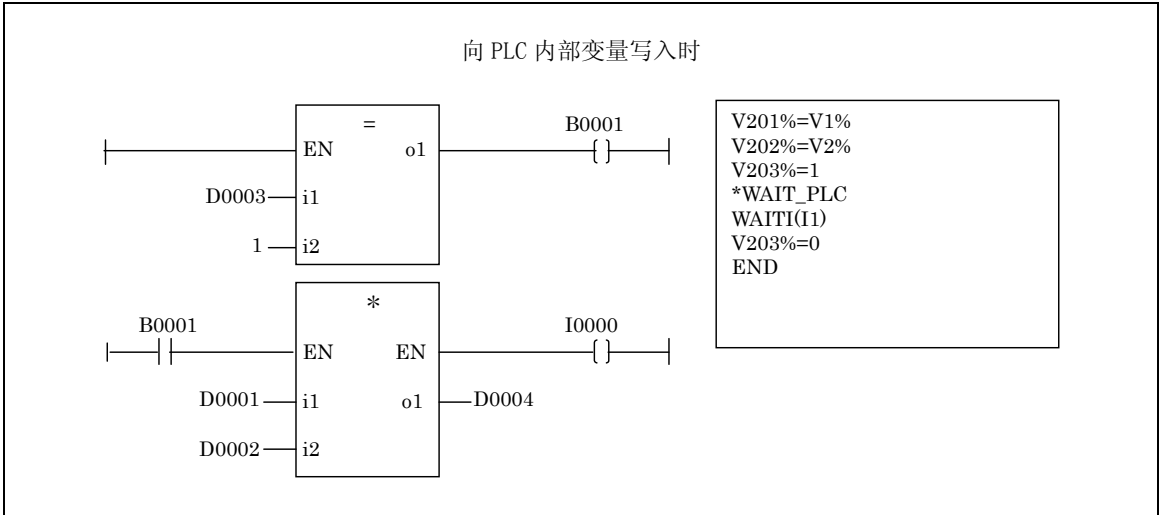


图 2.5.3 201~250 的全局实数变量使用示例

上图中，将全局整数变量 V1%, V2%分别存储到 PLC 内部的整数变量 D0001, D0002 中后，将其积存储到 PLC 内部的整数变量 D0004 中。

在上图中，向 V201%, V202%, V203%中设定值的顺序不会逆转，因此在 V203%(PLC 中为 D0003)中设定 1、I1 (PLC 中为 I0000) 为 ON 时，D0001, D0002 的内容确定。

因此，可通过在 V203%中设定 1、I1 为 ON，来实现 PLC 与机器人语言间的联锁。



## 2.5.2 实数变量

“实数变量”处理的是含有小数点的数值。

表 2.9 实数变量

格式	Vn!, V![n]            n=1~250, 301~500 (也可使用变量。) Ln!, L![n]            n=1~200 (也可使用变量。) ※ V1!与 V![1]表示相同的变量领域。
范围	-1.0E38~+1.0E38
示例	V1!=V2!*103.45 L![1]=1.567E-2 GETANGLE V1!
保存	所有的全局整数变量在主电源切断时也可作为停电保持数据得以保存。而所有的局部整数变量均不保存。

实数变量为整数时，计算结果有时为整数。

在机器人语言计算中，不是根据左边的变量形式，而是根据右边的变量形式计算。

请参照“表 2.10 实数变量与整数变量共存时的计算示例（1）”与“表 2.11 实数变量与整数变量共存时的计算示例（2）”的计算结果。

表 2.10 实数变量与整数变量共存时的计算示例（1）

格式	V42!的结果与显示	期待值	备注
V42!=152/10	15	15.2	-
V42!=152.0/10	15	15.2	152.0 作为整数处理
V42!=152/10.0	15	15.2	10.0 作为整数处理
V42!=152.0/10.0	15	15.2	分子、分母均作为整数处理
V42!=152/10+V41!	25	25.2	V41!=10, V41!=10.0
V42!=152/10+V41!	25.1	25.3	V41!=10.1
V42!=152/10+10	25	25.2	-
V42!=152/10+10.0	25	25.2	-
V42!=152/10+10.1	25.1	25.3	-

除法结果为整数的示例。

这些计算不能获得预期的计算结果。

表 2.11 实数变量与整数变量共存时的计算示例（2）

格式	V42!的结果与显示	期待值	备注
V42!=V41!/10	15.2	15.2	V41!=152
V42!=V41!/10	15.2	15.2	V41!=152.0
V42!=V41!/10	15.21	15.21	V41!=152.1
V42!=152*0.1	15.2	15.2	V42!=152/10 的代替方案示例

除法结果为实数的示例。

这些计算可以获得预期的计算结果。



- 全局实数变量的 201~250 是进行 PLC 内部的实数变量与实数值的收发的变量。如向上述全局变量中写入值，则值将被写入 PLC 内部的实数变量中。此外，如读取上述全局变量，将读取 PLC 内部的实数变量值。全局实数变量与 PLC 内部的实数变量之间的对应关系如下表所示。
- 写入上述 201~250 的全局实数变量中时，在变更反映到 PLC 内部的实数变量中之前，将发生延迟。



关于 V201!~V250!的全局实数变量使用时的注意事项，请参照“V201%~V250%的全局整数变量使用时的注意事项”（p2-9）。

表 2.12 全局实数变量与 PLC 内部的实数变量的对应

全局实数变量号码	PLC 内部的实数变量号码
V201! 或 V![201]	R0001
V202! 或 V![202]	R0002
V203! 或 V![203]	R0003
⋮	⋮
V248! 或 V![248]	R0048
V249! 或 V![249]	R0049
V250! 或 V![250]	R0050

2.5.3 字符串变量

“字符串变量”用来处理字符串。  
仅支持 ASCII 与 Shift\_JIS。也可使用 Shift\_JIS 的 2 字节代码。

表 2.13 字符串变量

格式	Vn\$, V\${n}            n=1~50（也可使用变量。） Ln\$, L\${n}            n=1~50（也可使用变量。） (V1\$与 V\${1}表示相同的变量领域)
范围	0~199 字符（199 字节）
示例	V1\$="RUNNING" L\${1}=V1\$+"STOPPING" LETVS V1\$, "1A"
保存	到第 10 号为止的全局字符串变量，即使主电源切断也可作为停电保持数据得以保存。另一方面，所有的局部字符串变量均不保存。

2.5.4 定时器变量

目前尚不支持定时器变量。

2.5.5 输入信号变量

以位(Bit)、或群组(Byte)单位（1 群组=10 位）处理输入端口的变量。

表 2.14 输入信号变量

	以位为单位处理时	以群组为单位处理时
范围	I <sub>n</sub> , I[n] n=1~2048, 5101~5196（可使用变量）	IB <sub>n</sub> , IB[n] n=1~205（可使用变量）
范围	0, 1	0~1023 （IB[205]时为 0~255）
示例	WAIT I[1], 0, 0 命令: WAIT FN 号码: FN552 名称: 带定时器输入信号等待 输入信号: 1 等待时间: 0 待避步骤: 0	WAITAD IB[1], 255, 0, 100 命令: WAITAD FN 号码: FN558 名称: 带定时器群组信号等待 BCD (AND) 群组号码: 1 条件: 255 等待时间: 0 待避步骤: 100

输入信号变量仅用于参照，不能写入。  
以群组为单位处理时请参照下表。下表的群组号码对应 IB[n]的 n。

表 2.15 输入信号的群组号码

群组	输入信号	群组	输入信号	群组	输入信号
1	1~10	11	101~110	21	201~210
2	11~20	12	111~120	...	...
3	21~30	13	121~130	30	291~300
4	31~40	14	131~140	...	...
5	41~50	15	141~150	50	491~500
6	51~60	16	151~160	...	...
7	61~70	17	161~170	100	991~1000
8	71~80	18	171~180	...	...
9	81~90	19	181~190	...	...
10	91~100	20	191~200	204	2031~2040
				205	2041~2048



（使用示例 1）

IF I1=1 THEN \*FINISH ELSE \*REPEAT  
在本例中，输入信号 I1 为 1 (=ON) 时跳转到标签 “\*FINISH”。



（使用示例 2）

```
SWITCH IB[1]
CASE 1
GOTO *LABEL1
BREAK
CASE 2
GOTO *LABEL2
BREAK
CASE
GOTO *LABEL3
BREAK
ENDS
```

在本例中，将根据输入信号选择跳往端。在 I1~I10 中，仅有 I1 为 ON 则跳至 “\*LABEL1”、仅有 I2 为 ON 则跳至 “\*LABEL2”，除此之外均跳至 “\*LABEL3”。

### 2.5.6 输出信号变量

以位(Bit)、或群组(Byte)单位（1 群组=10 位）处理输出端口的变量。  
作为 SET, OUT, SETM 等应用命令的参数使用。无法向输出信号变量中直接代入、操作输出信号。

表 2.16 输出信号变量

	以位为单位处理时	以群组为单位处理时
范围	On, O[n] n=1~2048（也可使用变量。）	OBn, OB[n] n=1~205（也可使用变量。）
范围	0, 1	0~1023 （OB[205]时为 0~255）
示例	SET O1/RESET O[2] 命令: SET FN 号码: FN32 名称: 输出信号 ON 输出信号号码: 01  →01 变为 ON。  命令: RESET FN 号码: FN34 名称: 输出信号 OFF 输出信号号码: 01  →01 变为 OFF。	OUT OB205, 0 命令: OUT FN 号码: FN44 名称: 输出信号二进制输出 群组号码: OB205 数据: 0  →02041~02048 全部变为 OFF。

以群组为单位处理时请参照下表。下表的群组号码对应 OB[n]的 n。

表 2.17 输出信号的群组号码

群组	输出信号	群组	输出信号	群组	输出信号
1	1~10	11	101~110	21	201~210
2	11~20	12	111~120	...	...
3	21~30	13	121~130	30	291~300
4	31~40	14	131~140	...	...
5	41~50	15	141~150	50	491~500
6	51~60	16	151~160	...	...
7	61~70	17	161~170	100	991~1000
8	71~80	18	171~180	...	...
9	81~90	19	181~190	...	...
10	91~100	20	191~200	204	2031~2040
				205	2041~2048



（使用示例）  
IF O1=1 THEN \*FINISH ELSE \*REPEAT

在本例中，输出信号 o1 为 1 (=ON) 时跳转到标签 “\*FINISH”。

## 2.5.7 固定输入信号变量

以位、或群组单位（1 群组=8 位）处理固定输入端口的变量。

表 2.18 输入信号变量

	以位为单位处理时	以群组为单位处理时
范围	FIn, FI[n] n=1~48（可使用变量）	FIBn, FIB[n] n=1~6（可使用变量）
范围	0, 1	0~255
示例	V1% = FI1	V1% = FIB1

固定输入信号变量仅用于参照，不能写入。

表 2.19 固定输入信号

No.	信号名称	状态
1	运转准备投入	操作面板（操作箱）的运转准备按钮 ON 并亮灯
2	G-STOP	G-STOP 输入 OFF 并灭灯 G-STOP 信号的详情请参照“设定篇”3.7 节。
3	起动 1	操作面板（操作箱）的起动 1 按钮 ON 并亮灯
4	起动 2	起动箱 2 的起动 2 按钮 ON 并亮灯
5	起动 3	起动箱 3 的起动 3 按钮 ON 并亮灯
6	起动 4	起动箱 4 的起动 4 按钮 ON 并亮灯
7	停止	通常为亮灯，悬式示教作业操纵按钮台、操作面板、操作箱、起动箱中任意一个的停止按钮为 ON 时灭灯
8	再生模式	操作面板（操作箱）的选择开关转向“再生”时亮灯。转向“示教”时灭灯
9	罩面开关	启动开关为 ON 且示教启动开关输入（TBEX1_13-14, 15-16）为 ON 时亮灯
10	-	-
11	高速示教	-
12	P1 正常	24V 供给时亮灯
13	外部紧急停止输入	“紧急停止”灭灯，或外部紧急停止输入（TBEX1_1-2, 3-4）为 OFF（紧急停止输入中）时灭灯 紧急停止输入解除时亮灯
14	紧急停止输入	“TP 紧急停止”灭灯，或操作面板（操作箱）、起动箱中任意一个紧急停止按钮被按下时（紧急停止输入中）灭灯 紧急停止输入全部解除时亮灯
15	安全插头	“再生模式”亮灯且安全插头输入（TBEX1_5-6, 7-8）ON 时亮灯
16	运转准备投入确认	满足下述所有条件时 ON 运转准备继电器（CRON1 和 CRON2）为 ON 时亮灯
17	TP 紧急停止	紧急停止输入解除时亮灯 按下悬式示教作业操纵按钮台的紧急停止按钮时（紧急停止输入中）灭灯
18	示教模式	操作面板（操作箱）的选择开关转向“示教”时亮灯。转向“再生”时灭灯。
19	-	-
20	自动停止	启动开关（自动停止开关）ON
21	-	-
22	CRON	伺服 ON 继电器（CRON1 和 CRON2）为 ON 时亮灯
23	伺服 ON	伺服 ON 时亮灯
24	伺服动作可能	放大器正常信号为 ON 时亮灯
25	-	-
26	-	-
27	-	-
28	电磁开关 ON	电磁开关（MS1 和 MS2）为 ON 时亮灯

29	-	-
30	熔断检测	在运转准备 OFF 状态下通常为亮灯。 在运转准备 OFF 状态下，电磁开关、或者继电器发生熔断时灭灯。
31	不一致检出	平时灭灯，检测到双重安全电路的不一致时亮灯
32	-	-
33	不一致（GSTOP）	平时亮灯，G-STOP 输入双重安全电路中发生不一致时灭灯
34	不一致（示教/再生）	平时亮灯，示教模式输入或再生模式输入双重安全电路发生不一致时灭灯
35	不一致（MAT-SW）	平时亮灯，罩面开关双重安全电路中发生不一致时灭灯
36	不一致（高速示教）	平时亮灯，高速示教输入双重安全电路中发生不一致时灭灯
37	不一致（外部 ES）	平时亮灯，外部紧急停止输入双重安全电路中发生不一致时灭灯
38	不一致（ES）	平时亮灯，紧急停止输入双重安全电路中发生不一致时灭灯
39	不一致（SP）	平时亮灯，安全插头输入双重安全电路中发生不一致时灭灯
40	不一致（TP-ES）	平时亮灯，TP 紧急停止双重安全电路中发生不一致时灭灯
41	不一致（TP 动作可能）	平时亮灯，启动开关的双重安全电路中发生不一致时灭灯
42	不一致（CRON）	平时亮灯，CRON 双重安全电路中发生不一致时灭灯
43	-	-
44	-	-
45	-	-
46	-	-
47	-	-
48	-	-

以字节为单位处理时请参照下表。下表的群组号码对应 FIB[n] 的 n。

表 2.20 固定输入信号的群组号码

群组号码	固定输入信号
1	1~8
2	9~16
3	17~24
4	25~32
5	33~40
6	41~48



固定输入信号的状态可通过以下菜单进行确认。  
<维修>-[监视器 1][固定输入信号]

## 2.5.8 固定输出信号变量

以位、或群组单位（1 群组=8 位）处理固定输出端口的变量。

表 2.21 固定输出信号变量

	以位为单位处理时	以群组为单位处理时
范围	FOn, FO[n] n=1~24（也可使用变量。）	FOBn, FOB[n] n=1~3（也可使用变量。）
范围	0, 1	0~255
示例	V1% = FO1	V1% = FOB1

固定输出信号变量仅用于参照，不能写入。

表 2.22 固定输出信号

No.	信号名称	状态
1	运转准备投入	灭灯：运转准备 OFF、亮灯：运转准备 ON 亮灯：可以通过使启动开关 ON 实现运转准备 ON 的状态
2	运转准备 ON 请求	在操作面板（操作箱）的运转准备按钮 ON 期间亮灯
3	起动指示灯 1	单元 1 起动中 ON
4	起动指示灯 2	单元 2 起动中 ON
5	起动指示灯 3	单元 3 起动中 ON
6	起动指示灯 4	单元 4 起动中 ON
7	暂停指示灯	暂停中亮灯 机器人起动时 OFF。 （轴操作中仍维持 ON）
8	TP 动作可能释放	放开启动开关时亮灯
9	运转准备 ON 许可	软件允许运转准备时亮灯 发生伺服异常、软件不允许运转准备时，以及 CPU 发生异常时灭灯
10	电磁开关 ON 许可	收到来自软件的伺服 ON 许可时亮灯
11	内部/外部选择	起动方式设定为“外部”时亮灯 起动方式为“内部”、或“多工位”时灭灯
12	WPS 紧急停止控制	向电焊机输出软件许可信号时亮灯 向电焊机输出紧急停止信号的条件之一
13	CPU 异常输出	平时灭灯（CPU 异常只有硬件的输出） 请确认输出状态为 PD3 或 TBEX2 的 CPU 异常输出
14	TP 模式选择	悬式示教作业操纵按钮台的 TP 选择开关为“手动”时亮灯 为“自动”时灭灯
15	外部准备 ON 请求	输入信号“外部运转准备投入”为 ON 时 ON。
16	—	—
17-24	—	—

以字节为单位处理时请参照下表。下表的群组号码对应 FOB[n] 的 n。

表 2.23 固定输出信号的群组号码

群组号码	固定输出信号
1	1~8
2	9~16
3	17~24

2.5.9 姿势变量

姿势变量是存储机器人姿势（POSE）的变量。通常是从事先制作的“姿势文件”中读取使用、但在程序中也能以如下的

```
(X, Y, Z, roll, pitch, yaw) 的形式代入。  
P1 = (1800, 0, 2000, 0, -90, -180)
```

```
有伺服焊枪等追加轴时，请在参数末尾后描述。  
P1 = (1800, 0, 2000, 0, -90, -180, -20)
```

表 2.24 姿势变量

格式	Pn, P[n]                    n=1~9999（也可使用变量。）
示例	P1 = (100, 0, 100, 0, 0, 90)

（使用示例 1）  
机构 1: 6 轴的机器人  
机构 2: 无

```
USE 1  
P1 = (1800,0,2000,0,-90,-180)  
MOVEX A=1,M1X,P,P1,R=10.0,H=1,MS  
END
```

在本例中姿势变量“P1”的要素数为(X, Y, Z, roll, pitch, yaw)这 6 个。

（使用示例 2）  
机构 1: 6 轴的机器人  
机构 2: 1 轴的伺服焊枪

```
USE 1  
P1 = (1800,0,2000,0,-90,-180,-20)  
MOVEX A=1,M1X,P,P1,R=10.0,H=1,MS,M2X,P,P1,R= 10,H=1  
END
```

在本例中姿势变量“P1”的要素数包括伺服焊枪的值总计为 7 个。请注意这一要素同时适用于机构 1 和 2。机器人读取第 1 个到第 6 个的数据，伺服焊枪读取第 7 个数据。



- 关于姿势文件的制作以及姿势变量的记录、修正请参照“3.3 制作姿势文件”。
- 在程序内使用姿势变量时请事先使用 USE 命令选择所使用的姿势文件的号码（1~9999）。未选择时将无法读取姿势变量，发生错误。
- 姿势文件中未记录的姿势变量不属于停电保持功能的对象，切断主电源时将被清除。
- 在程序内即使向姿势变量中设定值，姿势文件自身也不会被变更。需更改姿势文件时请执行“FN74 POSESAVE”。



向姿势变量中代入数值移动机器人时，必须保持充分的注意。所设定的坐标错误时机器人会做出意想不到的动作，非常危险。在移动实际的机器之前，推荐使用离线模拟器“FD on Desk”等进行确认。



2.5.10 移位变量

移位变量是存储移位量的变量。  
以 (X, Y, Z, roll, pitch, yaw) 的形式描述。  
不对应外部轴。

表 2.25 移位变量

格式	$Rn, R[n]$ $n=1\sim 9$ (也可使用变量。)
示例	$R1=(10, 1, 0, 0, 0, 0)$

## 2.6 任意变量

任意变量可以用英文数字为变量取名。还可以是数组变量。

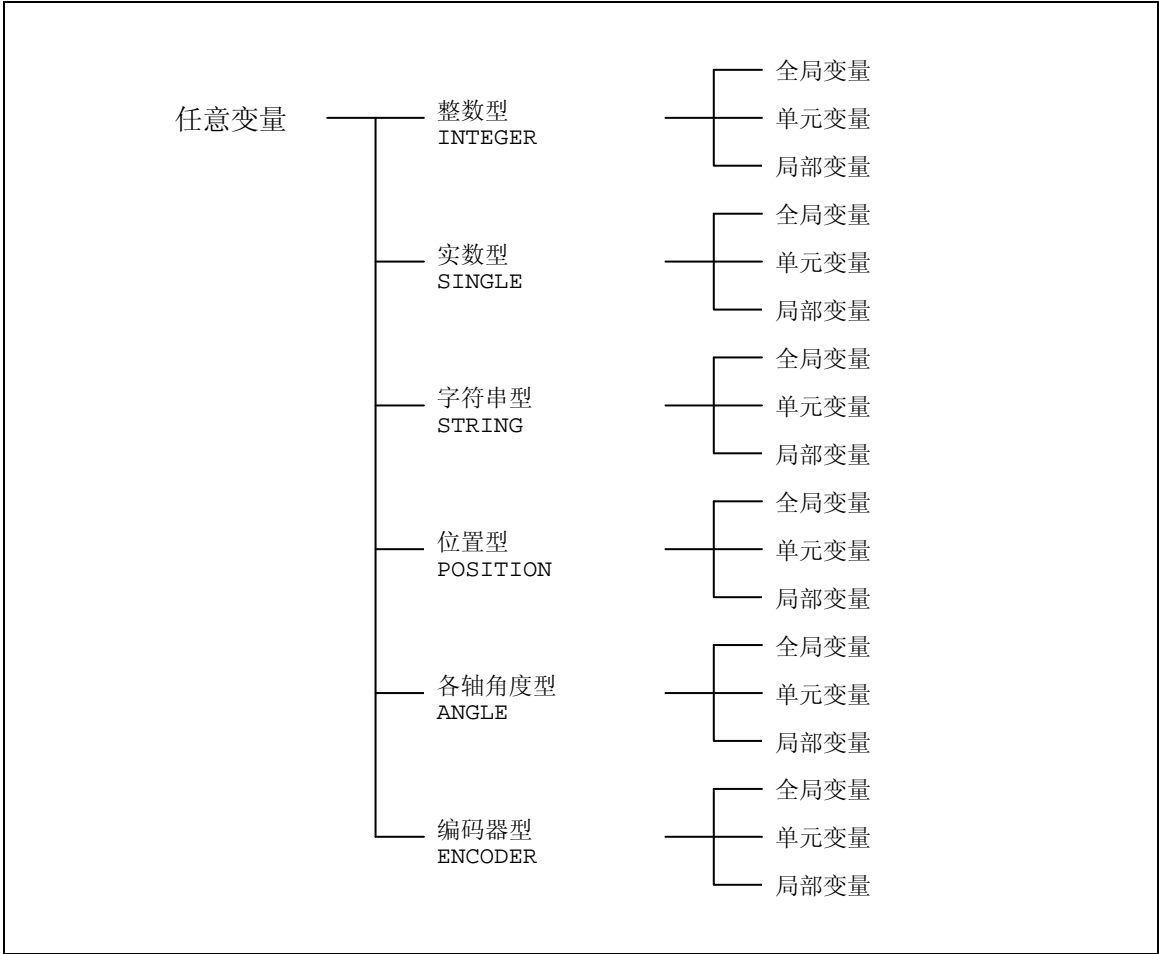


图 2.6.1 任意变量的种类

表 2.26 群组变量和单元变量以及局部变量

全局变量	可从所有单元中访问。 请在“Public.inc”内的[GLOBAL]段中进行定义。
单元变量	尽管可以从所有的单元中读取，但数据的保存只能通过各个单元完成。 请在“Public.inc”内的[UNIT*]段中进行定义。 (*为单元号码)
局部变量	在程序内（包括用户程序）所定义的变量，无法在其他程序及单元中访问。



各变量的内容保存在以下文件中。

全局变量和单元变量       : PUBLIC.DAT  
局部变量                   : LOCAL-\*\*.DAT (\*\*为单元号码)

(注意) “LOCAL-\*\*.DAT” 在执行程序结束命令时或程序选择时被删除。

2.6.1 任意变量的定义

任意变量可以按如下方式定义。

```
DIM intData[10] AS INTEGER
DIM angHOME AS ANGLE
```

详细的格式如下所示。

表 2.27 任意变量的定义









格式	DIM 变量名 ([数组]) AS 变量类型 ※不使用数组时，划线部分可省略。		
变量名	变量名最多可以指定 20 个字符。  请参照“表 2.28 变量名的可使用字符”。		
	首字符必须为英文字母。 英文字母区分大小写。 不能使用相同的变量名及保留字。		
数组	最多可以定义 3 维。 但位置型、各轴角度型、编码器型最多只能定义 2 维。 数组的要素从 1 开始。 数组的要素最多为 256 个。  请参照“表 2.29 数组的声明”		
变量类型	INTEGER	整数型	 参照“2.6.2 整数型”)
	SINGLE	实数型	 参照“2.6.3 实数型”)
	STRING	字符串型	 参照“2.6.4 字符串型”)
	POSITION	位置型	 参照“2.6.5 位置型”)
	ANGLE	各轴角度型	 参照“2.6.6 各轴角度型”)
	ENCODER	编码器型	 参照“2.6.7 编码器型”)

表 2.28 变量名的可使用字符

项目	可使用字符
英文数字	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 01234567890
符号	—

表 2.29 数组的声明

数组	例
1 维	DIM intData[6] AS INTEGER
2 维	DIM intData[3, 6] AS INTEGER
3 维	DIM intData[3, 3, 6] AS INTEGER

重点

推荐为变量使用易于理解其意义及类型的名称。例如，表示作业原点（HOME POSITION）的角度型（ANGLE）变量可以使用“angHOME”这样的名称。



- 目前下述的命令尚未支持任意变量。  
请像原先一样使用常数（整数变量 V%及实数变量 V!等）。

命令	名称
JUMP	步骤转移 (FN20)
CALL	步骤调用 (FN21)
IF	条件 (FN602)
ON	ON GOTO 转移 (FN603)
PRINT	字符串描绘 (FN606)
WHILE	循环时 (FN663)
IF	IF 条件 (FN676)
SWITCH	SWITCH (FN686)
MOVE	移动命令

任意变量（全局变量和单元变量）的定义

使用全局变量和单元变量时，请制作类似下例的变量定义文件 “Public.inc”，并放置在内部存储器的文件夹 “Work\Program\” 中。

```
[GLOBAL]
DIM g_intTotalCount As INTEGER
DIM g_sngShiftValue[3] As SINGLE

[UNIT1]
Dim u1_encHOME As ENCODER

[UNIT2]
Dim u2_encHOME As ENCODER
```

(1)  
(2)  
(3)  
(4)  
(5)  
(6)  
(7)

(1) 开始全局变量的段  
(2) 全局变量的声明  
(3) 全局变量的声明  
(4) 开始单元 1 变量的段  
(5) 单元 1 变量的声明  
(6) 开始单元 2 变量的段  
(7) 单元 2 变量的声明

图 2.6.2 变量定义文件 Public.inc

- 开始任意变量的声明之前，请描述[GLOBAL]或[UNIT\*]。[GLOBAL]表示全局变量的段开始，[UNIT\*]表示单元变量的段开始。
- 禁止在 1 个语句中途插入换行代码。
- 复制了变量定义文件时、或使用 ASCII 文件编辑功能编辑了变量定义文件时，为进行初始化，请重新投入一次电源。重新投入一次电源后，将执行 Public.inc 的语法检查和确保用于变量的存储领域以及 PUBLIC.DAT 文件的制作，当语法有错误时，将显示异常。

所定义变量的数据值，保存在数据文件（PUBLIC.DAT）中。  
变更变量名、或增加数组数时，数据值将变回初始值 0。

2-23

任意变量（局部变量）的定义

使用局部变量时，请在类似下例的程序内进行声明。

```
1 DIM intData[10] As INTEGER      (1)
2 FOR L1%=1 TO 10 STEP 1         (2)
3 intData[L1%] = I[L1%]          (3)
4 NEXT                           (4)
5 END                             (5)
```

(1) 局部变量的声明 制作任意变量（值不确定）

(2) 循环处理

(3) 用输入信号状态设定局部变量 向任意变量写入值

(4) 循环处理端

(5) 程序结束 删除任意变量

图 2.6.3 局部变量



- 在同一程序内无法定义相同的变量名。
- 无法定义与全局变量及单元变量相同的变量名。
- 无法定义与保留字相同的变量名。
- 在程序内使用局部变量时，定义和值这两者都必须设定。
- 数组的下标从 1 开始。
- 局部变量保存在数据文件 LOCAL-\*\*.DAT 中。但是执行程序结束命令（END）或程序选择时，该文件被删除。



所定义变量的数据值，保存在数据文件（LOCAL-\*\*.DAT）中。  
更改变量名时，或增加数组数时，请重新选择程序，或执行变量定义命令。

## 2.6.2 整数型 (INTEGER 型)

处理不含小数点的数值。

表 2.30 整数变量

格式	DIM 变量名 ( <u>[数组]</u> ) AS INTEGER ※不使用数组时, <u>划线部分</u> 可省略。
数组	最多可以定义 3 维。
范围	-2147483648~+2147483647
示例	获取 I1~I10 的信号状态。  <pre> DIM intData[10] AS INTEGER FOR L1%=1 TO 10 STEP 1   intData[L1%] = I[L1%] NEXT </pre>

## 2.6.3 实数型 (SINGLE 型)

处理含有小数点的数值。

表 2.31 实数变量

格式	DIM 变量名 ( <u>[数组]</u> ) AS SINGLE ※不使用数组时, <u>划线部分</u> 可省略。
数组	最多可以定义 3 维。
范围	-1.0E38~+1.0E38
示例	获取机构 1 的 TCP 速度 (指令、当前)。  <pre> DIM sngTCPSpeed[2] AS SINGLE sngTCPSpeed[1] = SYSTEM!(300) sngTCPSpeed[2] = SYSTEM!(800) </pre> <div> ‘机构 1 的 TCP 指令速度  ‘机构 1 的 TCP 当前速度 </div>

## 2.6.4 字符串型 (STRING 型)

处理字符串。

仅支持 ASCII 与 Shift\_JIS。也可使用 Shift\_JIS 的 2 字节代码。

表 2.32 字符串变量

格式	DIM 变量名 ( <u>[数组]</u> ) AS STRING ※不使用数组时, <u>划线部分</u> 可省略。
数组	最多可以定义 3 维。
范围	0~199 个字符 (199 字节)
示例	设定输出到 RS232C 的信息。  <pre> DIM strOutMessage AS STRING strOutMessage = "SHIFT l¥n¥r" </pre>

### 2.6.5 位置型 (POSITION 型)

位置型的任意变量可以以 (X, Y, Z, ROLL, PITCH, YAW, CONF) 的形式处理位置数据。

表 2.33 位置型变量

格式	DIM 变量名 ( [数组] ) AS POSITION ※不使用数组时, 划线部分可省略。
数组	最多可以定义 2 维。
范围	-1.0E38~+1.0E38
示例	<p>获取当前值, 根据输入信号, 在机械坐标系下进行动作。  I1 为沿+X 方向、I3 为沿+Y 方向、I5 为沿+Z 方向动作。  I2 为沿-X 方向、I4 为沿-Y 方向、I6 为沿-Z 方向动作。</p> <pre> <b><u>DIM posCurrent AS POSITION</u></b> DIM sgnPosition[7] AS SINGLE `(X,Y,Z,Roll,Pitch,Yaw,CONF) USE 1 GETPOS 0,1,posCurrent,0,1 OPEPOS 0,posCurrent,sgnPosition[1],0,0  WHILE(I10)   IF (I1)                                     `+X     sgnPosition[1] = sgnPosition[1] + 2.0   ELSEIF (I2)                                 `-X     sgnPosition[1] = sgnPosition[1] - 2.0   ELSEIF (I3)                                 `+Y     sgnPosition[2] = sgnPosition[2] + 2.0   ELSEIF (I4)                                 `-Y     sgnPosition[2] = sgnPosition[2] - 2.0   ELSEIF (I5)                                 `+Z     sgnPosition[3] = sgnPosition[3] + 2.0   ELSEIF (I6)                                 `-Z     sgnPosition[3] = sgnPosition[3] - 2.0 ENDIF OPEPOS 0,posCurrent,sgnPosition[1],0,1 POS2POSE 0,posCurrent,0,1,P1 MOVEX A=1,AC=0,SM=0,M1J,P,P1,R= 10,H=1,MS `MOVE to P1 ENDW  END </pre>

## 重点

- 对于伺服焊枪及走行轴、定位器，作为各轴值的数据。
- 旋转轴的单位为[DEG]，直动轴的单位为[mm]。
- 向位置型代入值时，必须使用专用的应用命令。
- 如下面多重机构规格时，1个 POSITION 变量将保持 9 个数据。

## M1: 6 轴机器人

M2: 1 轴伺服焊枪

M3: 1 轴滑块

```
(M1:X, M1:Y, M1:Z, M1:ROLL, M1:PITCH, M1:YAW, M1:CONF, M2:J1, M3:T1)
```



### 2.6.6 各轴角度型(角度)(ANGLE 型)

各轴角度型的任意变量可以以 (J1, J2, J3, J4, J5, J6) 的形式处理位置数据。

表 2.34 各轴角度型变量

格式	DIM 变量名 ( <u>[数组]</u> ) AS ANGLE ※不使用数组时， <u>划线部分</u> 可省略。
数组	最多可以定义 2 维。
范围	-1.0E38~+1.0E38
示例	<p>获取当前值，根据输入信号进行动作。（指定角度移动）</p> <pre> <b><u>DIM angCurrent AS ANGLE</u></b> DIM sgnPosition[6] AS SINGLE  GETANG 0,1, angCurrent OPEANG 0,posCurrent,sgnPosition[1],0,0  WHILE(I10)   IF (I1)                                'J1     sgnPosition[1] = sgnPosition[1] + 1.0   ELSEIF (I2)                            'J2     sgnPosition[2] = sgnPosition[2] + 1.0   ELSEIF (I3)                            'J3     sgnPosition[3] = sgnPosition[3] + 1.0   ELSEIF (I4)                            'J4     sgnPosition[4] = sgnPosition[4] + 1.0   ELSEIF (I5)                            'J5     sgnPosition[5] = sgnPosition[5] + 1.0   ELSEIF (I6)                            'J6     sgnPosition[6] = sgnPosition[6] + 1.0   ENDIF OPEPOSE 0,P1,sgnPosition[1],0,1 MOVEX A=1,AC=0,M1X,P,P1,R= 10.0,H=1,MS  'MOVE to P1 ENDW  END </pre>

**重点**

- 旋转轴为 Deg，直动轴为 mm。
- 必须使用专用的命令（函数）进行操作。

2.6.7 编码器型 (ENCODER 型)

处理编码器数据 (J1, J2, J3, J4, J5, J6)。

表 2.35 编码器型变量

格式	DIM 变量名 ([数组]) AS ENCODER ※不使用数组时可省略。
数组	最多可以定义 2 维。
范围	0x000100 ~ 0x0ffff00
示例	<div>获取当前值，根据输入信号进行动作。(指定编码器移动)</div> <div><div><div><div><b><u>DIM encCurrent AS ENCODER</u></b></div><div>DIM intPosition[6] AS INTEGER</div></div><div>GETENC 0,1, encCurrent</div><div>OPEENC 0,encCurrent,intPosition[1],0,0</div><div>WHILE(I10)</div><div><div>IF (I1)</div><div>intPosition[1] = intPosition[1] + 32</div><div>ELSEIF (I2)</div><div>intPosition[2] = intPosition[2] + 32</div><div>ELSEIF (I3)</div><div>intPosition[3] = intPosition[3] + 32</div><div>ELSEIF (I4)</div><div>intPosition[4] = intPosition[4] + 32</div><div>ELSEIF (I5)</div><div>intPosition[5] = intPosition[5] + 32</div><div>ELSEIF (I6)</div><div>intPosition[6] = intPosition[6] + 32</div><div>ENDIF</div></div><div>OPEENC 0,encCurrent,intPosition[1],0,1</div><div>ENC2POSE 0,encCurrent,P1</div><div>MOVEX A=1,AC=0,M1X,P,P1,R= 10.0,H=1,MS</div><div>ENDW</div><div>MOVE to P1</div><div>END</div></div></div>

重点

▪ 必须使用专用命令（函数）进行操作。

2.6.8 转换命令

通过执行向不同的变量类型转换（代入）的命令，如由位置型向各轴角度型及编码器型、姿势变量转换，可以根据用途选择适合的变量进行操作。移动机器人时，最终还是必须转换为姿势变量赋予MOVEX。

表 2.36 转换命令

命令	Fn 号码	转换源	转换目标	备注
POS2POSE	FN809	位置型	姿势变量	
ANG2POSE	FN810	各轴角度型	姿势变量	
ENC2POSE	FN811	编码器型	姿势变量	
POSE2POS	FN812	姿势变量	位置型	
ANG2POS	FN813	各轴角度型	位置型	
ENC2POS	FN814	编码器型	位置型	
POSE2ANG	FN815	姿势变量	各轴角度型	
POS2ANG	FN816	位置型	各轴角度型	
ENC2ANG	FN817	编码器型	各轴角度型	
POSE2ENC	FN818	姿势变量	编码器型	
POS2ENC	FN819	位置型	编码器型	
ANG2ENC	FN820	各轴角度型	编码器型	
CVTCOORDPOS	FN821	位置型	位置型	坐标系转换

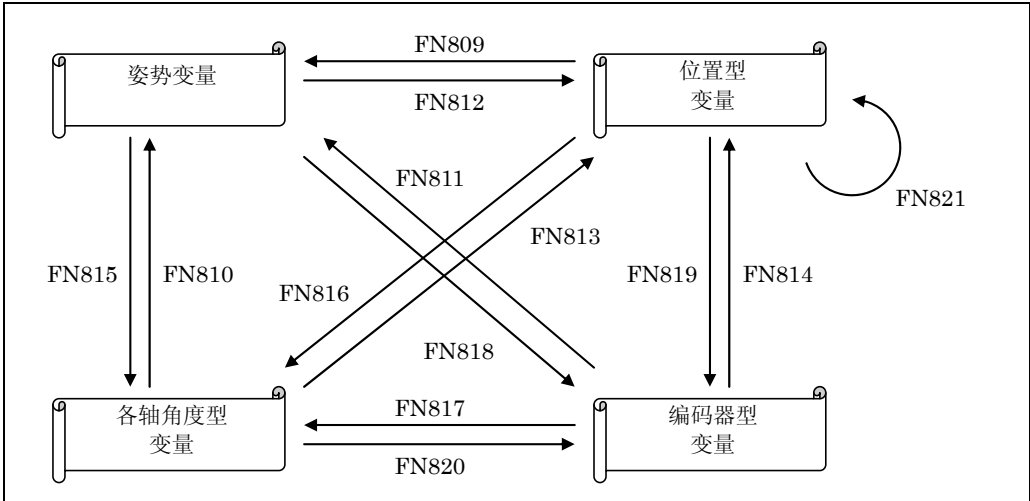


图 2.6.4 转换

重点

关于该说明书未记载的各命令参数的详情，请参照在线帮助。

（示例）自位置型的转换

位置型变量 (X, Y, Z, Roll, Pitch, Yaw, CONF) 将转换为姿势变量、各轴角度型变量 (角度) (ANGLE)、编码器型变量 (ENCODER)。

表 2.37 POS2POSE

功能	将位置型变量转换为姿势变量。	
格式	POS2POSE MechanismNo, PosVariableName, CoordinateSystem, CoordSystemNo, PoseVariableName	
参数	MechanismNo	机构编号 (0-9) (“0”表示单元内部的所有机构)
	PosVariableName	位置型变量的名称
	CoordinateSystem	位置型变量的坐标系 0：机械坐标系（机器人的基本坐标系） 1：工具坐标系 2：用户坐标系 3：直角坐标系（世界坐标系） 4：工件坐标系
	CoordSystemNo	工具编号或者用户坐标系编号 (NOTE) 其他坐标系时，通常为 1
	PoseVariableName	转换目标的姿势变量名称（例：“P1”）

表 2.38 POS2ANG

功能	将位置型变量转换为各轴角度型变量。	
格式	POS2POSE MechanismNo, PosVariableName, CoordinateSystem, CoordSystemNo, AngVariableName	
参数	MechanismNo	机构编号 (0-9) (“0”表示单元内部的所有机构)
	PosVariableName	位置型变量的名称
	CoordinateSystem	位置型变量的坐标系 0：机械坐标系（机器人的基本坐标系） 1：工具坐标系 2：用户坐标系 3：直角坐标系（世界坐标系） 4：工件坐标系
	CoordSystemNo	工具编号或者用户坐标系编号 (NOTE) 其他坐标系时，通常为 1
	AngVariableName	转换目标的各轴角度型变量名称

表 2.39 POS2ENC

功能	将位置型变量转换为编码器型变量。	
格式	POS2POSE MechanismNo, PosVariableName, CoordinateSystem, CoordSystemNo, EncVariableName	
参数	MechanismNo	机构编号 (0-9) (“0”表示单元内部的所有机构)
	PosVariableName	位置型变量的名称
	CoordinateSystem	0：机械坐标系（机器人的基本坐标系） 1：工具坐标系 2：用户坐标系 3：直角坐标系（世界坐标系） 4：工件坐标系
	CoordSystemNo	工具编号或者用户坐标系编号 (NOTE) 其他坐标系时，通常为 1
	EncVariableName	转换目标的编码器型变量名称

应用示例

系统结构	单元内部的机构数=1（6 轴机器人）
示例	<p>（源程序）</p> <pre>USE 1 DIM posCurrent AS POSITION DIM angCurrent AS ANGLE DIM encCurrent AS ENCODER DIM sngPosition[7] AS SINGLE  GETPOS 1, 1, posCurrent, 0, 1      '(1) OPEPOS 1, posCurrent, sngPosition[1], 0, 0  '(2) sngPosition[1] = sngPosition[1] + 10.0      '(3) OPEPOS 1, posCurrent, sngPosition[1], 0, 1  '(4) POS2POSE 1, posCurrent, 0, 1, P1          '(5) POS2ANG 1, posCurrent, 0, 1, angCurrent    '(6) POS2ENC 1, posCurrent, 0, 1, encCurrent    '(7)  MOVEX A=1,M1X,P,P1,R=10,H=1,MS END</pre>
	<p>（说明）</p> <p>(1) 从位置型变量获取机构 1 的当前位置</p> <p>(2) 从实数型数组提取位置型变量的机构 1 所有要素</p> <p>(3) 实数型数组的要素 1（X 坐标值）为+10[mm]</p> <p>(4) 将实数型数组代入位置型变量</p> <p>(5) 将位置型变量转换姿势变量</p> <p>(6) 将位置型变量转换各轴角度型变量</p> <p>(7) 将位置型变量转换编码器型变量</p>

多重机构规格的机器人时，从实数变量及实数型变量提取位置型变量时所需的数组要素为如下所示。不管机构形式，每 1 个机构需要 7 个要素，敬请注意。“-”要素通常为“0.0”。

（示例）  
M1：机器人（6 轴机器人）  
M2：伺服焊枪（1 轴）  
M3：滑块（1 轴）  
所需数组的要素： 7 x 3 = 21  
-> sngPosition[21]



No.	值	No.	值	No.	值
1	M1:X	8	M2:J1	15	M3:T1
2	M1:Y	9	-	16	-
3	M1:Z	10	-	17	-
4	M1:ROLL	11	-	18	-
5	M1:PITCH	12	-	19	-
6	M1:YAW	13	-	20	-
7	M1:CONF	14	-	21	-

2.6.9 位置获取命令(GET)

另外，通过命令还能执行如下所示的位置获取、任意变量和实数变量/整数变量之间的提取/代入。  
下面用位置获取命令来说明命令。

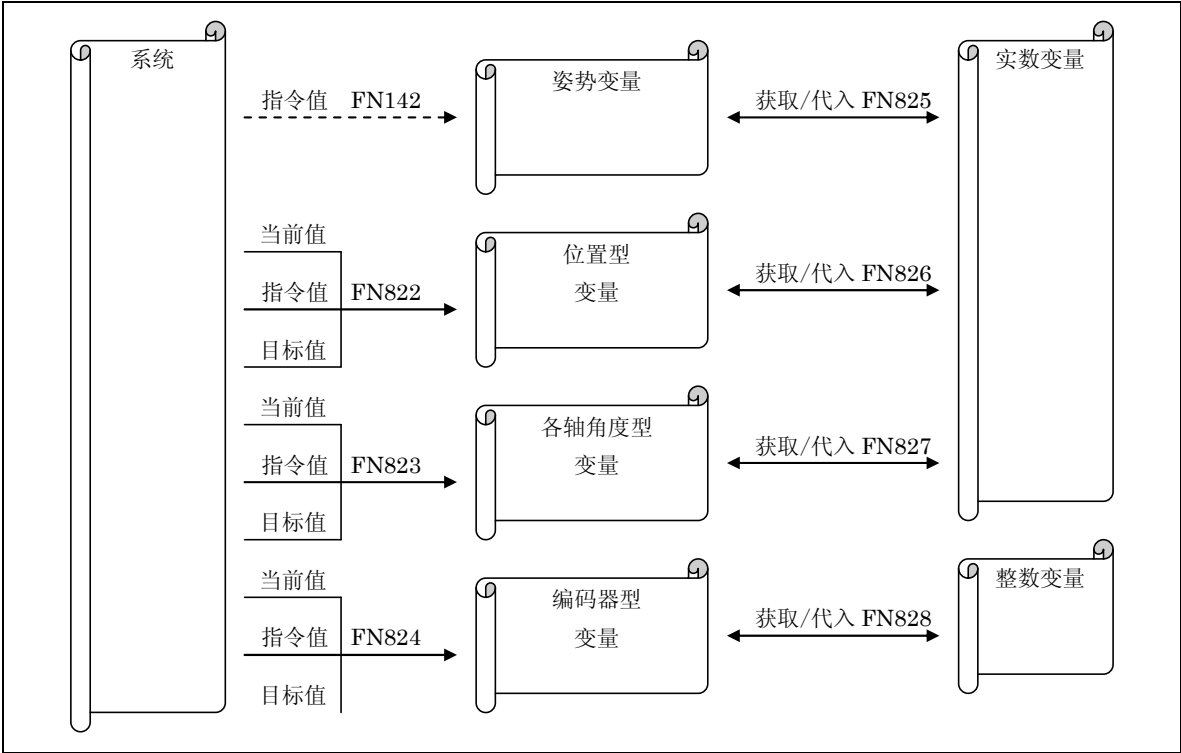


图 2.6.5 获取/提取/代入

通过使用下列命令，以位置型/各轴角度型/编码器型等变量形式获取位置。

表 2.40 FN822 GETPOS

功能	从位置型变量获取位置	
格式	GETPOS MechanismNo, PosType, <u>PosVariableName</u> , CoordinateSystem, CoordSystemNo	
参数	MechanismNo	机构编号 (0-9) ("0" 表示单元内部的所有机构)
	PosType	位置类型 (0-2) 0 : 指令位置 1 : 当前位置 2 : 目标位置
	PosVariableName	位置型变量的名称
	CoordinateSystem	位置型变量的坐标系 0 : 机械坐标系 (机器人的基本坐标系) 1 : 工具坐标系 2 : 用户坐标系 3 : 直角坐标系 (世界坐标系) 4 : 工件坐标系
	CoordSystemNo	工具编号或者用户坐标系编号 (NOTE) 其他坐标系时, 通常为 1

表2.41 FN823 GETANG

功能	从各轴角度型变量获取位置	
格式	GETANG MechanismNo, PosType, <u>AngVariableName</u>	
参数	MechanismNo	机构编号 (0-9) ("0" 表示单元内部的所有机构)
	PosType	位置类型 (0-2) 0 : 指令位置 1 : 当前位置 2 : 目标位置
	AngVariableName	各轴角度型变量的名称

表2.42 FN824 GETENC

功能	从编码器型变量获取位置	
格式	GETENC MechanismNo, PosType, <u>EncVariableName</u>	
参数	MechanismNo	机构编号 (0-9) ("0" 表示单元内部的所有机构)
	PosType	位置类型 (0-2) 0 : 指令位置 1 : 当前位置 2 : 目标位置
	EncVariableName	编码器型变量的名称

2.6.10 提取/代入命令(OPE)

表 2.43 FN825 OPEPOSE

功能	从实数变量(V! / L!)或者实数型变量（SINGLE 型变量）提取姿势变量 或者 将实数变量(V! / L!)或者实数型变量（SINGLE 型变量）代入姿势变量	
格式	OPEPOSE MechanismNo, PoseVariableNo, RealVariable, TargetData, OperationType	
参数	MechanismNo	机构编号（0-9） （“0”表示单元内部的所有机构）
	PoseVariableNo	姿势变量编号
	RealVariable	实数变量。可使用下列变量 ▪ 全局实数变量 V! ▪ 局部实数变量 L! ▪ 实数型变量（SINGLE 型变量） (NOTE) 根据操作对象，有时需要多个要数。 如果实数型变量时，指定数组。
	TargetData	指定操作对象。 0：所有 1：X 2：Y 3：Z 4：R 5：P 6：Y
	OperationType	指定提取/代入。 0: 提取 (Extract) 1: 代入 (Substitute)
示例	OPEPOSE 1, P1, V1!, 1, 0 从 V1!提取姿势变量 1 的 X 坐标值。  OPEPOSE 1, P1, V1!, 1, 1 将 V1!代入姿势变量 1 的 X 坐标值。	

（参照）关于所需要素

- M1: 机器人（6 轴机器人）
- M2: 伺服焊枪（1 轴）
- M3: 滑块（1 轴）

如果这种多重机构规格时，姿势变量为如下 8 要素。  
(M1X, M1Y, M1Z, M1ROLL, M1PITCH, M1YAW, M2J1, M3T1)



例如，当以 V1!开始的全局实数变量提取所有该姿势变量的要素时，将成为如下。也就是需要 6 × 3 = 18 个实数变量。如果实数型变量（SINGLE 型变量）时，敬请注意数组要素的数量。“-”要素通常为“0.0”。

OPEPOSE 0, P1, V1!, 0, 0

V!	值	V!	值	V!	值
1	M1X	7	M2J1	13	M3T1
2	M1Y	8	-	14	-
3	M1Z	9	-	15	-
4	M1ROLL	10	-	16	-
5	M1PITCH	11	-	17	-
6	M1YAW	12	-	18	-



表 2.44 FN826 OPEPOS

功能	从实数变量(V! / L!)或者实数型变量（SINGLE 型变量）提取位置型变量 或者 将实数变量(V! / L!)或者实数型变量（SINGLE 型变量）代入位置型变量	
格式	OPEPOS MechanismNo, PosVariableName, RealVariable, TargetData, OperationType	
参数	MechanismNo	机构编号（0-9） （“0”表示单元内部的所有机构）
	PoseVariableNo	位置型变量编号
	RealVariable	实数变量。可使用下列变量 ▪ 全局实数变量 V! ▪ 局部实数变量 L! ▪ 实数型变量（SINGLE 型变量） (NOTE) 根据操作对象，有时需要多个要数。 如果实数型变量时，指定数组。
	TargetData	指定操作对象。 0：所有 1：X 2：Y 3：Z 4：ROLL 5：PITCH 6：YAW 7：CONF (NOTE) 1 轴机构只能指定 0。
	OperationType	指定提取/代入。 0：提取（Extract） 1：代入（Substitute）
示例	OPEPOS 1, posPosition, V1!, 1, 0 从 V1!提取位置型变量内部机构 1 的 X 坐标值。  OPEPOS 1, posPosition, V1!, 1, 1 将 V1!代入位置型变量内部机构 1 的 X 坐标值。	

（参照）关于所需要素

M1：机器人（6 轴机器人）  
M2：伺服焊枪（1 轴）  
M3：滑块（1 轴）

如果这种多重机构规格时，位置型变量为如下 9 要素。  
(M1X, M1Y, M1Z, M1ROLL, M1PITCH, M1YAW, M1CONF, M2J1, M3T1)

例如，当以 V1!开始的全局实数变量提取所有该姿势变量的要素时，将成为如下。也就是需要 7×3＝21 个实数变量。如果实数型变量（SINGLE 型变量）时，敬请注意数组要素的数量。“-”要素通常为“0.0”。



OPEPOS 0, posPosition, V1!, 0, 0

V!	值	V!	值	V!	值
1	M1X	8	M2J1	15	M3T1
2	M1Y	9	-	16	-
3	M1Z	10	-	17	-
4	M1ROLL	11	-	18	-
5	M1PITCH	12	-	19	-
6	M1YAW	13	-	20	-
7	M1CONF	14	-	21	-

表 2.45 FN827 OPEANG

功能	从实数变量(V! / L!)或者实数型变量（SINGLE 型变量）提取各轴角度型变量 或者 将实数变量(V! / L!)或者实数型变量（SINGLE 型变量）代入各轴角度型变量	
格式	OPEANG MechanismNo, AngVariableName, RealVariable, TargetData, OperationType	
参数	MechanismNo	机构编号（0-9） （“0”表示单元内部的所有机构）
	PoseVariableNo	各轴角度型变量编号
	RealVariable	实数变量。可使用下列变量 ▪ 全局实数变量 V! ▪ 局部实数变量 L! ▪ 实数型变量（SINGLE 型变量） (NOTE) 根据操作对象有时需要多个要数。 如果实数型变量时，指定数组。
	TargetData	指定操作对象。 0：所有 1：J1 2：J2 3：J3 4：J4 5：J5 6：J6 (NOTE) 1 轴机构只能指定 0。
	OperationType	指定提取/代入。 0：提取（Extract） 1：代入（Substitute）
示例	OPEANG 1, angAngle, V1!, 1, 0 从 V1!提取各轴角度型变量内部机构 1 的 J1 值。  OPEANG 1, angAngle, V1!, 1, 1 将 V1!代入各轴角度型变量内部机构 1 的 J1 值。	

（参照）关于所需要素

- M1: 机器人（6 轴机器人）
- M2: 伺服焊枪（1 轴）
- M3: 滑块（1 轴）

如果这种多重机构规格时，位置型变量为如下 8 要素。  
(M1J1, M1J2, M1J3, M1J4, M1J5, M1J6, M2J1, M3T1)



例如，当以 V1!开始的全局实数变量提取所有该姿势变量的要素时，将成为如下。也就是需要 6×3=18 个实数变量。如果实数型变量（SINGLE 型变量）时，敬请注意数组要素的数量。“-”要素通常为“0.0”。

OPEANG 0, angAngle, V1!, 0, 0

V!	Value	V!	Value	V!	Value
1	M1J1	7	M2J1	13	M3T1
2	M1J2	8	-	14	-
3	M1J3	9	-	15	-
4	M1J4	10	-	16	-
5	M1J5	11	-	17	-
6	M1J6	12	-	18	-

表 2.46 FN828 OPEENC

功能	从实数变量(V% / L%)或者实数型变量（INTEGER 型变量）提取编码器型变量 或者 将实数变量(V% / L%)或者实数型变量（INTEGER 型变量）代入编码器型变量	
格式	OPEENC MechanismNo, EncVariableName, IntVariable, TargetData, OperationType	
参数	MechanismNo	机构编号（0-9） （“0”表示单元内部的所有机构）
	PoseVariableNo	编码器型变量编号
	RealVariable	整数变量。可使用下列变量 ▪ 全局整数变量 V% ▪ 局部整数变量 L% ▪ 整数型变量（INTEGER 型变量） (NOTE) 根据操作对象有时需要多个要数。 如果整数型变量时，指定数组。
	TargetData	指定操作对象。 0：所有 1：J1 2：J2 3：J3 4：J4 5：J5 6：J6 (NOTE) 1 轴机构只能指定 0。
	OperationType	指定提取/代入。 0：提取（Extract） 1：代入（Substitute）
示例	OPEENC 1, encEncoder, V1%, 1, 0 从 V1%提取 ENCODER 变量内部机构 1 的 J1 值。  OPEENC 1, encEncoder, V1%, 1, 1 将 V1%代入 ENCODER 变量内部机构 1 的 J1 值。	

（参照）关于所需要素

- M1：机器人（6 轴机器人）
- M2：伺服焊枪（1 轴）
- M3：滑块（1 轴）

如果这种多重机构规格时，编码器型变量为如下 8 要素。  
(M1J1, M1J2, M1J3, M1J4, M1J5, M1J6, M2J1, M3T1)

例如，当以 V1%开始的全局整数变量提取所有该姿势变量的要素时，将成为如下。也就是需要 6×3=18 个整数变量。如果整数型变量（INTEGER 型变量）时，敬请注意数组要素的数量。“-”要素通常为“0.0”。



OPEENC 0, encEncoder, V1%, 0, 0

V%	Value	V%	Value	V%	Value
1	M1J1	7	M2J1	13	M3T1
2	M1J2	8	-	14	-
3	M1J3	9	-	15	-
4	M1J4	10	-	16	-
5	M1J5	11	-	17	-
6	M1J6	12	-	18	-

（参照）16 进制和 10 进制

用 16 进制记载的 80000，如果转换为 10 进制的话，成为 524288。

示例程序

根据输入信号的状态，从当前位置以指定方法进行 JOG 动作时的示例程序。

USE 500	
DIM sgnPosition[7] As SINGLE	(1) 任意变量定义
DIM sgnShift As SINGLE	(2) 任意变量定义
DIM posCurrent As POSITION	(3) 任意变量定义
GETPOS 0,0,posCurrent,0,1	(4) 当前值获取
OPEPOS 0,posCurrent,sgnPosition[1],0,0	(5) 当前值代入
sgnShift = 20	(6) 移位量设置
IF (I1)	'X+
sgnPosition[1] = sgnPosition[1]+sgnShift	(7) 移位量相加
ELSEIF (i2)	'X-
sgnPosition[1] = sgnPosition[1]-sgnShift	(8) 移位量相减
ELSEIF (i3)	'Y+
sgnPosition[2] = sgnPosition[2]+sgnShift	(9) 移位量相加
ELSEIF (i4)	'Y-
sgnPosition[2] = sgnPosition[2]-sgnShift	(10) 移位量相减
ELSEIF (i5)	'Z+
sgnPosition[3] = sgnPosition[3]+sgnShift	(11) 移位量相加
ELSEIF (i6)	'Z-
sgnPosition[3] = sgnPosition[3]-sgnShift	(12) 移位量相减
ENDIF	
OPEPOSE 0,P1,sgnPosition[1],0,1	(13) 姿势变量代入
MOVEX A=1,AC=0,SM=0,M1X,P,P1,R= 5.0,H=1,MS, CONF=0000	(14) 动作指令
END	

(1) 用于加减 JOG 动作量的任意变量定义

(2) JOG 动作量的任意变量定义

(3) 当前位置的任意变量定义

(4) 获取当前位置的函数

(5) 提取用于加减当前位置的任意变量的函数

(6) 设定 JOG 动作量

(7) ~ (12) 根据输入信号，加减向各方向的 JOG 动作量

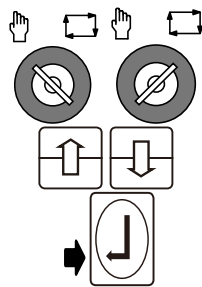
(13) 向姿势变量代入动作位置

(14) 移向动作位置的动作指令

图 2.6.6 示例程序

2.6.11 任意变量监视器

使用[任意变量监视器]，可以显示所定义的任意变量。[任意变量监视器]的更新频率为 0.5 秒，但根据程序的执行时机等有时会不显示。

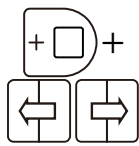


- 1 示教或再生模式均可。  
即使正处于再生运转中也无碍。

- 2 按下 f 键<维修>，选择[监视器 1（~4）][任意变量监视器]。

>> 显示如下监视器。

[2] 任意变量监视器		[国际]
intA	0	
sigA	0.000000	
strA		

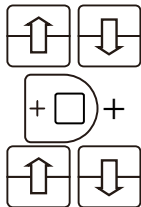


- 3 使用[动作可能]+[左右光标]键选择要参照的任意变量的种类。  
>> 切换任意变量监视器所显示的任意变量的种类。  
>> 当前显示的任意变量的种类标识在任意变量监视器的右上角。

[2] 任意变量监视器		[国际]
intA	0	
sigA	0.000000	
strA		

>> 任意变量的种类按照以下顺序切换。

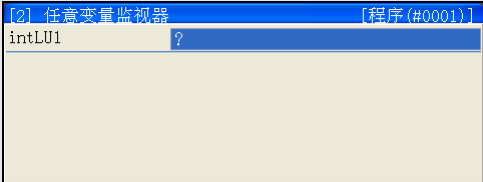
显示例	任意变量的种类	备注
[全局]	显示全局变量。	
[单元 (*U1) ]	显示当前单元的单元变量。 “*U1” 部分显示当前选中的当前单元的单元号码。	
[程序 (#0001) ]	显示当前单元正在执行中的程序内的局部变量。 “#0001” 部分显示执行中的程序的程序号码。	
[单元 (U1) ] ~ [单元 (U9) ]	显示单元变量。	仅显示控制装置中已登录的单元。
[用户任务 (1) ] ~ [用户任务 (4) ]	显示用户任务的单元变量。	仅在用户任务处于执行中时显示。
[用户任务 1 程序 (#0001) ] ~ [用户任务 4 程序 (#0001) ]	显示用户任务正在执行中的程序内的局部变量。 “#0001” 部分显示执行中的程序的程序号码。	仅在用户任务处于执行中时显示。



4 使用[上下光标]键逐行移动光标。

5 使用[动作可能]+[上下光标]键，以任意变量单位移动光标。

从任意变量定义开始，到程序结束或执行程序选择为止，在此期间局部变量可供参照。  
从执行任意变量定义开始，到第一次设定其值为止，在此期间下述任意变量监视器的值显示为“?”。



重点

位置型任意变量的显示

定义为位置型的任意变量，首行显示任意变量的名称，之后显示各数据的值。  
位置型的各数据以[M（机构号码）：（数据）]的形式显示。机械手类型的机构显示工具的前端位置，除此以外的机构显示轴值。

【显示例】

DIM posLU1 AS POSITION

[2] 任意变量监视器		[程序(#0001)]
posLU1		
[M1:X]	1690.00	
[M1:Y]	-0.00	
[M1:Z]	2030.00	
[M1:ROLL]	0.000000	
[M1:PITCH]	-90.000000	
[M1:YAW]	-180.000000	
[M1:C]	0	
[M2:J1 ]	0.00	
[M3:T1 ]	0.00	

机构类型	数据	显示的值	备注
机械手	X	显示工具前端的 X 坐标。	
	Y	显示工具前端的 Y 坐标。	
	Z	显示工具前端的 Z 坐标。	
	ROLL	显示工具前端的姿势（ROLL）。	
	PITCH	显示工具前端的姿势（PITCH）。	
	YAW	显示工具前端的姿势（YAW）。	
	C	显示工具前端的姿势的配置。	
除机械手以外	轴名称	显示轴值。 旋转轴显示角度（deg），直动轴显示长度。	

各轴角度(角度)型任意变量的显示

定义为各轴角度型的任意变量，首行显示任意变量的名称，之后显示各轴的轴值。  
各轴角度型的各数据以[M（机构号码）：（轴名称）]的形式显示。旋转轴显示角度（deg），直动轴显示长度。

【显示例】

DIM angLU1 AS ANGLE

[2] 任意变量监视器

[程序(#0001)]

angLU1	
[M1:J1 ]	0.000000
[M1:J2 ]	90.000000
[M1:J3 ]	0.000000
[M1:J4 ]	0.000000
[M1:J5 ]	0.000000
[M1:J6 ]	0.000000
[M2:J1 ]	0.00
[M3:T1 ]	0.00

编码器型任意变量的显示

定义为编码器型的任意变量，首行显示任意变量的名称，之后显示各轴的编码器值。  
编码器型的各数据以[M（机构号码）：（轴名称）]的形式显示。编码器值以 16 进制显示。

【显示例】

DIM encLU1 AS ENCODER

[2] 任意变量监视器

[程序(#0001)]

encLU1	
[M1:J1 ]	080000
[M1:J2 ]	080000
[M1:J3 ]	080000
[M1:J4 ]	080000
[M1:J5 ]	080000
[M1:J6 ]	080000
[M2:J1 ]	080000
[M3:T1 ]	080000

用数组定义的任意变量的显示

用数组定义的任意变量，首行显示任意变量的名称和数组数，之后显示各组号码的值。

【显示例】

DIM intLUArr[3,5] AS INTEGER

[2] 任意变量监视器

[程序(#0001)]

intLU1Arr[3,5]	
[1,1]	1
[1,2]	0
[1,3]	0
[1,4]	0
[1,5]	0



跳跃到指定的数组号码

使用数组的选择功能，可以跳跃到指定的数组号码。

重点

程序再生中无法执行数组的选择功能。

1

在任意变量监视器的激活状态下，按下[编辑]键。

>> 任意变量监视器切换到如下的编辑模式。

示教	程序	步骤	2013/10/28 13:59	
运转准备	2	10		
输入范围 -2147483648 - 2147483647				
[1] 机器人程序				
100 % JOINT A1 T1				
6 GETANG[0,1,angCurrent] FN823;设置角度变量(位				
7 GETENC[0,1,encCurrent] FN824;设置编码器变量(				
8 FOR V1% = 1 TO 2 STEP 1 FN604;循环开始				
9 FOR V2% = 1 TO 5 STEP 1 FN604;循环开始				
10 intMatrix[V1%,V2%] = 1 FN624;代入式				
11 NEXT FN605;循环结束				
12 NEXT FN605;循环结束				
[2] 任意变量监视器 [程序(#0002)]				
encCurrent				
intMatrix[2,5]				
上下移动	[1,1]	1		选择
	[1,2]	1		
	[1,3]	0		
画面分割	[1,4]	0		取消
				写入

2

将光标对准要参照的任意变量，按下 f 键[选择]。

>> 显示如下的数组号码输入画面。

排列
请输入排列号码。
intMatrix[2,5] [1 - 2]

3

输入要参照的数组号码，按下[Enter]键。

>> 2 维以上的数组需继续输入 2 维以后的数组号码。

排列
请输入排列号码。
intMatrix[2,5] [1 - 5]
2,

4

输入声明的相应维数的数组号码。

>> 光标跳跃到指定的数组号码。

编辑任意变量的值

使用编辑模式可以编辑任意变量的值。

- 重点

程序再生中无法编辑任意变量的值。
- 重点

执行了任意变量定义后未设定最初的值的局部变量无法编辑其值。



- 1

在任意变量监视器的活动状态下，按下[编辑]键。

>> 任意变量监视器切换到如下的编辑模式。

示教	程序	步骤	2013/10/28 14:05	
运转准备	2	10		M1: SRA166
输入范围 -50000.00 - 50000.0				
0				
[1] 机器人程序				
100 % JOINT A1 T1				
6 GETANG[0,1,angCurrent] FN823;设置角度变量(位				
7 GETENC[0,1,encCurrent] FN824;设置编码器变量(				
8 FOR V1% = 1 TO 2 STEP 1 FN604;循环开始				
9 FOR V2% = 1 TO 5 STEP 1 FN604;循环开始				
10 intMatrix[V1%,V2%] = 1 FN624;代入式				
11 NEXT FN605;循环结束				
12 NEXT FN605;循环结束				
[2] 任意变量监视器 [程序(#0002)]				
posCurrent				
上下移动	[M1:X]	1690.00		选择
	[M1:Y]	-0.00		取消
	[M1:Z]	2030.00		
	[M1:ROLL]	0.000000		
画面分割	[M1:PITCH]	-90.000000		写入
- 2

将光标对准要变更值的任意变量，使用数字键输入值，按下[Enter]键。
- 3

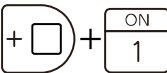
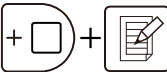
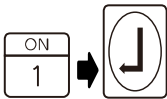
对于字符串型的任意变量，同时按下[动作可能]+[编辑]键，显示软键盘，使用软键盘输入值。
- 4

对于编码器型的任意变量，以 16 进制输入值。同时按下[动作可能]+[1] (~[6]) 可以输入 A (~F)。
- 5

值输入完成后按下 f 键[写入]。

>> 任意变量的值变更。

>> 要取消编辑时，按下 f 键[取消]或[复位]。



任意变量监视器的使用示例

示范执行以下三个应用命令时的示例。

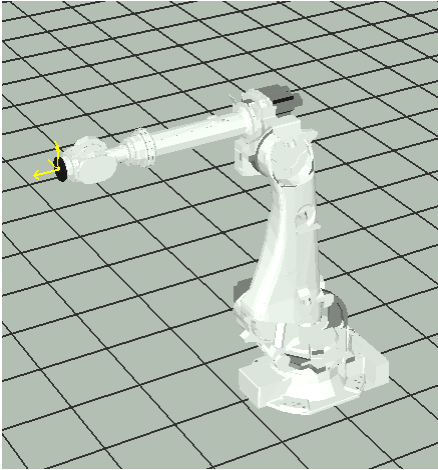
- FN822 GETPOS （从位置型变量中获取位置）
- FN823 GETANG （从角度型变量中获取位置）
- FN824 GETENC （从编码器型变量中获取位置）

程序

```
DIM posData AS POSITION
DIM angData AS ANGLE
DIM encData AS ENCODER
MOVEX A=1, AC=0, SM=0, M1J, P, (0, 90, 0, 0, 0, 0), R= 10, H=1, MS
GETPOS 0, 1, posData, 0, 1      ' 从位置型任意变量 posData 中获取当前位置(1)
GETANG 0, 1, angData            ' 从角度型任意变量 angData 中获取当前位置(2)
GETENC 0, 1, encData            ' 从编码器型任意变量 encData 中获取当前位置(3)
END
```

MOVEX 刚执行后的机器人状态

SRA166-01



No	指令值	现在值	角度	姿势
J1	080000	080000	0.0	X= 1690.0
J2	080000	080000	90.0	Y= -0.0
J3	080000	080000	0.0	Z= 2030.0
J4	080000	080000	0.0	r= 0.0 a= 0.0
J5	080000	080000	0.0	p= -90.0 b= 90.0
J6	080000	080000	0.0	y= -180.0 c= -180.0

执行 GETPOS 时 posData 的内容

- 由于机构 1（M1）为机械手，所以以 (X, Y, Z, ROLL, PITCH, YAW, CONF) 的顺序获取/显示数据。

posData	
[M1:X]	1690.00
[M1:Y]	-0.00
[M1:Z]	2030.00
[M1:ROLL]	0.000000
[M1:PITCH]	-90.000000
[M1:YAW]	-180.000000
[M1:C]	0

执行 GETANG 时 angData 的内容

- 获取/显示各轴的角度。单位为[DEG]。

angData	
[M1:J1]	0.000000
[M1:J2]	90.000000
[M1:J3]	0.000000
[M1:J4]	0.000000
[M1:J5]	0.000000
[M1:J6]	0.000000

执行 GETENC 时 encData 的内容

- 获取/显示各轴的编码器值。

encData	
[M1:J1]	080000
[M1:J2]	080000
[M1:J3]	080000
[M1:J4]	080000
[M1:J5]	080000
[M1:J6]	080000



- 存在多个机构时，未指定为第一参数对象机构的数据不会更新。（初始值为 0）

## 2.7 表达式与运算

所谓表达式是指使用运算表达式结合常数、变量的一般数学表达式及函数、或仅指字符、数值或变量。

【例】

- “ABC”
- 3\*5/4+10
- V1%+V2%\*V1!
- SIN(V3!)
- 1023
- V\$[4]

运算是指使用运算符及函数来操作表达式，分为以下 6 种。

- 算术运算
- 关系运算
- 逻辑运算
- 字符串运算
- 姿势运算
- 函数

### 2.7.1 算术运算

以下内容可用作算术运算符。

表 2.47 算术运算符

运算符	运算内容	例
^	指数（乘方）运算	10^2
-	负号	-100
*	相乘	3*5
/	相除	3/2
\	相除（商）	5\2
MOD	相除（余数）	5 MOD 2
+	相加	10+12
-	相减	103-99

### 2.7.2 关系运算

在比较 2 个数值时，使用关系运算。以真（1）、假（0）获取结果，用于在条件判断语句等程序执行顺序的变更等。

表 2.48 关系运算符

运算符	运算内容	例
=	等于	V1%=V2%
<>	不等于	V1%<>V2%
<	小于	V1%<V2%
>	大于	V1%>V2%
<=	小于等于	V1%<=V2%
>=	大于等于	V1%>=V2%

### 2.7.3 逻辑运算

在多个条件结合及位操作时使用逻辑运算。

表 2.49 逻辑运算符

运算符	运算内容	例
NOT	否定	NOT V1%
AND	逻辑与	V1% AND V2%
OR	逻辑或	V1% OR V2%
XOR	逻辑异或	V1% XOR V2%

### 2.7.4 字符串运算

字符串之间的运算，仅限于使用“+”运算符的字符串的结合、及使用关系运算符的字符串的比较。字符串的结合，通过在 2 个字符串之间插入“+”运算符来进行。

例如，

V1\$="ABC"  
V2\$=V1\$+"123"

如此，即向 V2\$中代入了“ABC123”这一字符串。

此外，也可使用关系运算符“=”、“<”来比较字符串。2 组字符串相等时，等号成立。不相等时，不等号成立。

### 2.7.5 姿势运算

姿势运算仅限于姿势常数（变量）与移位常数（变量）的相加。  
此外，运算结果不保存到所记录的姿势变量中。

- 【例】
- P1=P1+R1
  - P1=P1+(100,0,0,0,0,0)
  - P1=(0,1000,1200,90,0,90)+(80,0,0,0,0,0)
  - P1=(0,2000,100,0,0,0)+R1

## 2.7.6 一般函数

函数是针对某一参数进行确定的运算，并返回其运算结果。

### 函数的种类与内容

在 f/p/s 等中，也可使用变量。

各自的输入范围分别为：

- 整数值：-2147483648~+2147483647
- 实数值：-1.0E38~+1.0E38
- 字符串：0~199 字符（199 字节）

表 2.50 函数

函数名	功能	属性
DATE\$	将当前日期转换为字符串并返回。	字符串
TIME\$	将当前时刻转换为字符串并返回。	字符串
TIMER\$	返回自电源投入时起的时间（单位 ms）。	实数值
SQR(f)	求平方根。	实数值
SIN(f)	求 sin(f)（角度 f 为弧度）。	实数值
COS(f)	求 cos(f)（角度 f 为弧度）。	实数值
TAN(f)	求 tan(f)（角度 f 为弧度）。	实数值
ATN(f)	求 atan(f)。	实数值
ATN2(f1, f2)	求 atan(f1/f2)。	实数值
ABS(f)	求绝对值。	实数值
DEGRAD(f)	将角度值转换为弧度值。	实数值
RADDEG(f)	将弧度值转换为角度值。	实数值
MIN(f1, f2)	求 f1 与 f2 中较小的一个。	实数值
MAX(f1, f2)	求 f1 与 f2 中较大的一个。	实数值
ORD(s)	返回字符串开头的字符代码。	整数值
CHR\$(f)	返回以值 f 为字符代码的长度为 1 的字符串。	字符串
VAL(s)	将以字符串表现的数值转换为值。 数值以外的字符串（平假名、片假名、汉字、英文字符、符号等）转换为 0。	实数值
STR\$(f)	将数值转换为字符串。	字符串
BIN\$(i)	将数值转换为以 2 进制表现的字符串。	字符串
HEX\$(i)	将数值转换为以 16 进制表现的字符串。	字符串
MIRROR\$(s)	返回字符串 s 的逆序字符串。	字符串
LEFT\$(s, f)	从字符串 s 的左边开始截取长为 f 的字符串。	字符串
RIGHT\$(s, f)	从字符串 s 的右边开始截取长为 f 的字符串。	字符串
MID\$(s, f1, f2)	从字符串 s 的左边第 f1 个字符开始截取长为 f2 的字符串。	字符串
STRPOS(s1, s2)	返回在字符串 s1 中最初与字符串 s2 一致的位置（未发现时返回 199，s1 或 s2 没有数据时返回 0）。	整数值
LEN(s)	返回字符串的长度。	整数值

输入函数的种类与内容

表 2.51 输入函数



函数名	功能	属性
INP(i)	以 0(OFF) 或 1(ON) 返回通过参数 I 指定的输入信号的值。 (i 可指定为 1~2048, 5101~5196。)	整数值
INPB(i)	以字节为单位获取通过参数 i 指定的输入信号, 转换为 10 进制的整数值后返回。 (i 可指定为 1~256。)  请参照“表 2.52 输入函数 INPB 的参数 i”	整数值
GETSIG(i1,i2)	以位为单位获取通过 i1 指定的输入输出信号。 (i1 可指定为 1~2048, 5101~5196。但为输出信号时, 仅可指定为 1~2048。) i2=0 时, 获取输入信号。 i2=1 时, 获取输出信号。	整数值
GETSIGB(i1,i2)	以字节为单位获取通过 i1 指定的输入输出信号。 (i1 可指定为 0~259。) i2=0 时, 获取输入信号。 i2=1 时, 获取输出信号。  请参照“表 2.53 输入函数 GETSIGB 的参数 i1”	整数值
GETFIX(i1,i2)	以位为单位获取通过 i1 指定的固定输入输出信号。 i2=0 时, 获取固定输入信号。(i1 可指定为 1~32。) i2=1 时, 获取固定输出信号。(i1 可指定为 1~16。)	整数值
GETFIXB(i1,i2)	以字节为单位获取通过 i1 指定的固定输入输出信号。 i2=0 时, 获取固定输入信号。(i1 可指定为 1~4。) i2=1 时, 获取固定输出信号。(i1 可指定为 1~2。)	整数值

表 2.52 输入函数 INPB 的参数 i

i	输入号码	i	输入号码	i	输入号码	i	输入号码
1	0001~0008	9	0065~0072	~	***~***	249	1985~1992
2	0009~0016	10	0073~0080	242	1929~1936	250	1993~2000
3	0017~0024	11	0081~0088	243	1937~1944	251	2001~2008
4	0025~0032	12	0089~0096	244	1945~1952	252	2009~2016
5	0033~0040	13	0097~0104	245	1953~1960	253	2017~2024
6	0041~0048	14	0105~0112	246	1961~1968	254	2025~2032
7	0049~0056	15	0113~0120	247	1969~1976	255	2033~2040
8	0057~0064	~	***~***	248	1977~1984	256	2041~2048

将输入号码 1~2048 每 8 点分为一组。

表 2.53 输入函数 GETSIGB 的参数 i1

i1	输入输出号码	i1	输入输出号码	i1	输入输出号码	i1	输入输出号码
0	盘内 I/O	8	0033~0040	~	***~***	252	1985~1992
1	盘内 I/O	9	0041~0048	245	1929~1936	253	1993~2000
2	盘内 I/O	10	0049~0056	246	1937~1944	254	2001~2008
3	盘内 I/O	11	0057~0064	247	1945~1952	255	2009~2016
4	0001~0008	12	0065~0072	248	1953~1960	256	2017~2024
5	0009~0016	13	0073~0080	249	1961~1968	257	2025~2032
6	0017~0024	14	0081~0088	250	1969~1976	258	2033~2040
7	0025~0032	~	***~***	251	1977~1984	259	2041~2048

将盘内 I/O 及输入输出号码 1~2048 每 8 点分为一组。  
输入信号、输出信号两者号码相同。



## 2.7.7 系统函数

所谓系统函数是指为了读取机器人各轴位置等本控制装置的内部信息而事先准备的函数。系统函数有如下几种。

表 2.54 系统函数




函数名	功能	属性
ERRMESS(i)	返回错误号码 i 所对应的控制装置的错误信息。	字符串
SYSTEM%(i)	返回 i 所对应的控制装置所具有的整数信息。  请参照“表 2.55 系统函数 SYSTEM%的返回值”	整数值
SYSTEM!(i)	返回 i 所对应的控制装置所具有的实数信息。  请参照“表 2.56 系统函数 SYSTEM!的返回值”	实数值
SYSTEM\$(i)	返回 i 所对应的控制装置所具有的字符串信息。  请参照“表 2.57 系统函数 SYSTEM\$的返回值”	字符串

表 2.55 系统函数 SYSTEM%的返回值

参数	返回值
0	用户任务状态 第0位：用户任务程序启动中 第1位：自启动起第2个周期以后 第2位：用户画面打开 第3位：用户画面暂时关闭状态 第4位：自用户画面打开起第2个周期以后 第5位：用户画面激活（关键信息输入）状态 第6～31位：未使用
1	单元台数
2	模式（0=示教、1=再生、2=高速示教）
4	再生模式（0=1步骤、1=1循环、2=连续）
100	执行中的用户任务程序号码
101～109	单元1～9各自执行中的作业程序号码
110（120）	用户任务1的再生行号码
111～119	单元1～9各自的再生步骤号码
131～139	单元1～9各自的单元轴数
141～149	单元1～9各自的单元错误号码
151～159	单元1～9各自的移位状态（0:非移位中、1:移位动作中）
161～169	单元1～9各自的轴一致状态（0:非轴一致状态、1:轴一致状态）
200～205	机构1的J1轴～J6轴的当前编码器值
210～215	机构2 "
220～225	机构3 "
230～235	机构4 "
240～245	机构5 "
250～255	机构6 "
260～265	机构7 "
270～275	机构8 "
280～285	机构9 "
1000～1003	用户任务1～4的再生程序号码
1010～1013	用户任务1～4的再生行号码

2001~2006	返回在指定电焊机缝焊区间（开始~结束）中的执行中焊枪号码。
2011~2016	返回指定电焊机的缝常数中所设定的移动侧电极轮机构号码。
2021~2026	返回指定电焊机的缝常数中所设定的固定侧电极轮机构号码。

表 2.56 系统函数 SYSTEM! 的返回值

参数	返回值	单位
100~105	机构1的J1轴~J6轴的当前关节角度	rad
110~115	机构2 "	rad
120~125	机构3 "	rad
130~135	机构4 "	rad
140~145	机构5 "	rad
150~152	机构1当前的前端位置（X、Y、Z坐标）	mm
153~155	机构2 "	mm
156~158	机构3 "	mm
159~161	机构4 "	mm
162~167	机构5 "	mm
200~205	机构6的J1轴~J6轴的当前关节角度	rad
210~215	机构7 "	rad
220~225	机构8 "	rad
230~235	机构9 "	rad
250~252	机构6当前的前端位置（X、Y、Z坐标）	mm
253~255	机构7 "	mm
256~258	机构8 "	mm
259~261	机构9 "	mm
270	模拟电压1	V
271	模拟电压2	V
272	模拟电压3	V
273	模拟电压4	V
274	模拟电压5	V
275	模拟电压6	V
276	模拟电压7	V
277	模拟电压8	V
300	机构1的TCP指令速度	mm/sec
301	机构2的TCP指令速度	mm/sec
302	机构3的TCP指令速度	mm/sec
303	机构4的TCP指令速度	mm/sec
304	机构5的TCP指令速度	mm/sec
305	机构6的TCP指令速度	mm/sec
306	机构7的TCP指令速度	mm/sec
307	机构8的TCP指令速度	mm/sec
308	机构9的TCP指令速度	mm/sec
310~318	机构1的指令的前端位置（X、Y、Z、R、P、Y、A、B、C）	mm • RAD
319~327	机构2的指令的前端位置（X、Y、Z、R、P、Y、A、B、C）	mm • RAD
328~336	机构3的指令的前端位置（X、Y、Z、R、P、Y、A、B、C）	mm • RAD
337~345	机构4的指令的前端位置（X、Y、Z、R、P、Y、A、B、C）	mm • RAD
346~354	机构5的指令的前端位置（X、Y、Z、R、P、Y、A、B、C）	mm • RAD
355~363	机构6的指令的前端位置（X、Y、Z、R、P、Y、A、B、C）	mm • RAD
364~372	机构7的指令的前端位置（X、Y、Z、R、P、Y、A、B、C）	mm • RAD

373~381	机构8的指令的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
382~390	机构9的指令的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
800	机构1的TCP当前速度	mm/sec
801	机构2的TCP当前速度	mm/sec
802	机构3的TCP当前速度	mm/sec
803	机构4的TCP当前速度	mm/sec
804	机构5的TCP当前速度	mm/sec
805	机构6的TCP当前速度	mm/sec
806	机构7的TCP当前速度	mm/sec
807	机构8的TCP当前速度	mm/sec
808	机构9的TCP当前速度	mm/sec
810~818	机构1的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
819~827	机构2的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
828~836	机构3的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
837~845	机构4的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
846~854	机构5的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
855~863	机构6的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
864~872	机构7的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
873~881	机构8的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
882~890	机构9的当前的前端位置 (X、Y、Z、R、P、Y、A、B、C)	mm • RAD
1001~1006	指定电焊机的伺服焊枪指令加压力	Kgf
1011~1016	指定电焊机的伺服焊枪实际加压力	Kgf
2001~2006	指定缝焊机的移动侧电极轮旋转数	转
2011~2016	指定缝焊机的固定侧电极轮旋转数	转
2021~2026	指定缝焊机从焊接开始到焊接结束的焊接距离 缝焊开始时清零。	mm
2031~2036	指定缝焊机从焊接开始到焊接结束的焊接时间 缝焊开始时清零。	sec
2041~2046	指定缝焊机从焊接开始到焊接结束的通电距离 缝焊开始时清零。	mm
2051~2056	指定缝焊机从焊接开始到焊接结束的通电时间 缝焊开始时清零。	sec

伺服焊枪、滑块（走行装置）等的直动轴时，当前关节角度单位不是[rad]，而是[mm]。此外，英寸模式时的长度返回值也是[mm]。

表 2.57 系统函数 SYSTEM\$的返回值

参数	说明
0	软件版本
101	单元1的单元名称
102	单元2       "
103	单元3       "
104	单元4       "
105	单元5       "
106	单元6       "
107	单元7       "
108	单元8       "
109	单元9       "
300~2347	输入信号的名称
2348~4098	输出信号的名称
5001~5100	堆列名称

2.7.8 优先顺序

表达式中存在多个运算符时，这些运算符的适用顺序将根据每一个运算符被赋予的优先顺序来确定。运算符的优先顺序如下所示。

表 2.58 运算符的优先顺序

运算符	优先顺序
+, - (符号)	高
NOT	↑
~	
/, *, \, MOD	
+ (相加), - (相减)	
=, <>, >, <, >=, <=	
AND	
OR	↓
XOR	低

## 2.8 语句

所谓语句是指源程序的每 1 行中必须分配 1 个的成分。  
本书中如无特殊说明，程序的 1 行与 1 个语句将作相同处理。

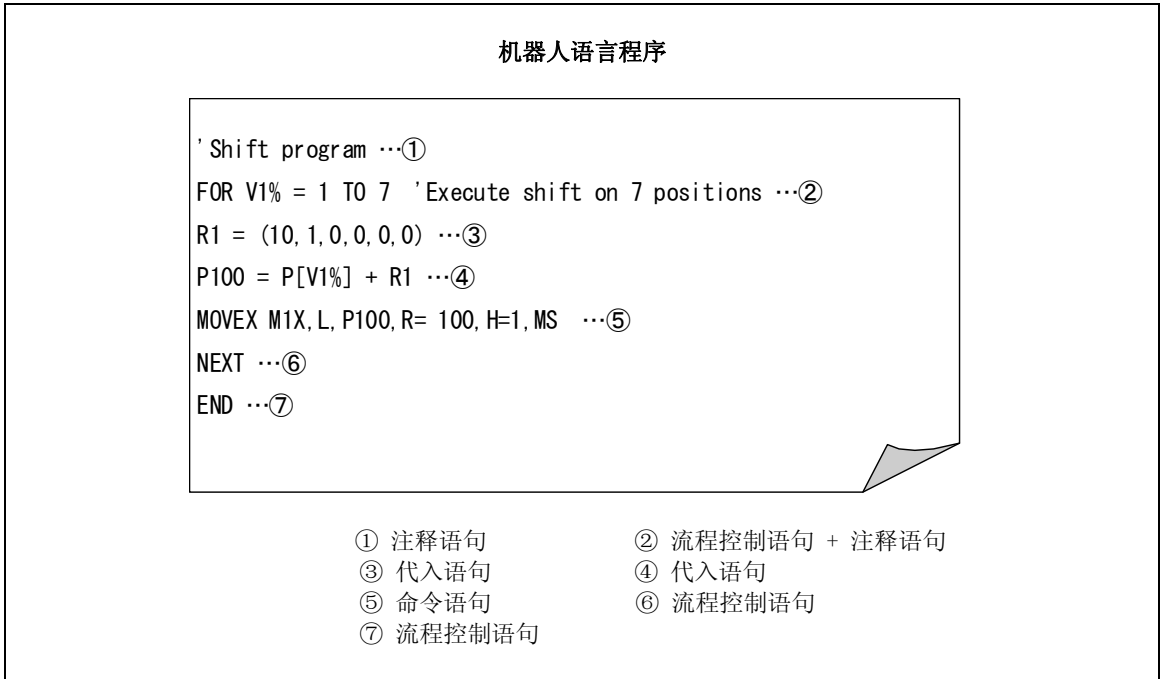


图 2.8.1 语句

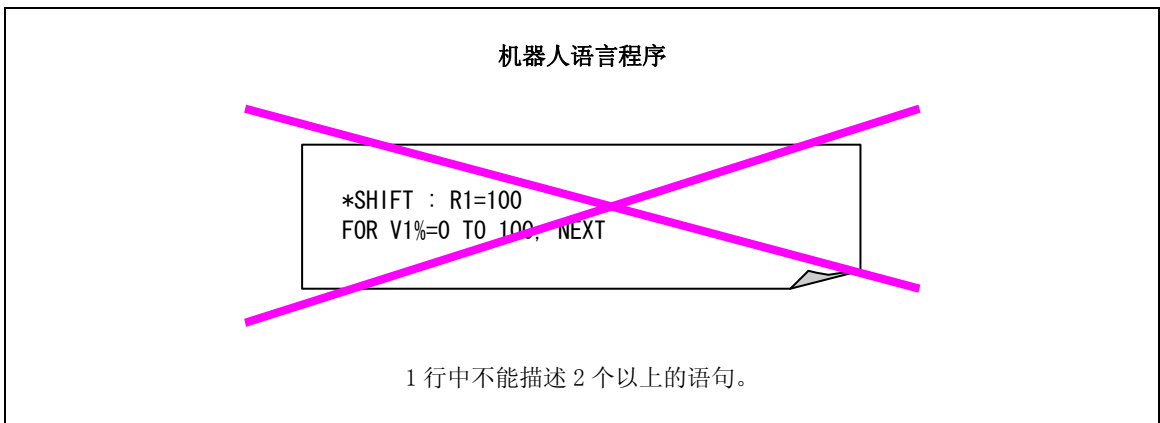


图 2.8.2 1 行中仅可描述 1 个语句



- 1 行中不能描述 2 个以上的语句。
- 禁止在 1 个语句中途插入换行代码。
- 1 个语句最多为 254 个字符。  
如写入 255 个字符以上时，将出现编译错误。

### 2.8.1 注释语句

如在语句的最初部分加入单引号（'）或 REM，则该行将成为注释语句（说明语句）。此外，也可在命令语句及流程控制语句等之后加入单引号（'）再描述说明。

描述注释语句是为了使程序更加易懂，其内容不会被执行。

说明的长度为 199 字符以内。

【例】

- 'SPOT1 Program
- REM "ARC ON"
- SET 01 'Task complete

**重点**

已经描述了说明的程序，由于需要用于描述说明的存储器空间，与未描述说明的程序相比，可用于描述的行数较少。可用于描述的行数，根据步骤说明的字符数等而变化。

### 2.8.2 标签语句

语句开头以星号（\*）开始，如后面接英文字母，便识别为标签。控制程序去向的命令有 GOTO、IF 语句等，指定标签名作为其分支端。

标签名以英文字符开始，16 字符以内。可使用的字符为英文数字及点（.）、下划线（\_）。程序内可描述的标签的总字符数应在 1024 字符以内。

【例】

- \*HANDLING1
- \*ARC\_WELDING

**提示**

标签可以像下面一样用作“FN602 IF”的跳跃端。

IF I1=1 THEN \*HANDLING1 ELSE \*ARC\_WELDING

### 2.8.3 代入语句

在程序中，可分别向整数变量、实数变量、字符串变量、移位变量、姿势变量中代入值。

在语句的开头描述变量名，然后如描述“=”，则识别为代入语句。

【例】

- V1%=10
- V3\$=V1\$
- R1=(100, 0, 0, 0, 0, 0)
- P1=(100, 1000, 1000, 0, 90, 90)

## 2.8.4 流程控制语句

重复执行及分支、子程序调用等变更程序流程的命令称为流程控制语句。以下为各流程控制语句的说明。

表 2.59 流程控制语句

命令	概要
GOTO 行号码/标签	无条件将控制转移到以行号码或标签表示的行。 【例】GOTO 100 GOTO *LOOP_END
GOSUB 行号码/标签	将控制转移到以行号码或标签名表示的子程序。 【例】GOSUB 120 GOSUB *FUNC1
RETURN	将控制从子程序返回到调用源（通过 GOSUB 调用的下一行）。
IF 条件表达式 THEN 行号码/标签 ELSE 行号码/标签	条件成立时，将控制转移到 THEN 之后所描述的行号码或标签。不成立时，将控制转移到 ELSE 之后所描述的行号码或标签。如未描述 ELSE，当条件不成立时，将控制转移到下一行。请务必在 THEN、ELSE 之后描述行号码或标签。不能描述执行语句。 【例】IF V1%=1 THEN 100 ELSE 200 IF V1%>100 AND V1%<200 THEN *START
FOR 变量名=初始值 TO 结束值 STEP 增量 NEXT	重复执行 FOR 与 NEXT 循环内所描述的命令。变量名必须是整数变量（V%，L%）或实数变量（V!，L!）。此外，必须直接指定变量（必须是 V1%，不能是 V%[1]）。初始值是在最初的循环中设定到变量中的值。当命令到达 NEXT 时，向变量中加上增量，与结束值比较。命令超过结束值时，将控制转移到 NEXT 的下一行。除上述以外的情况时，转移到 FOR 的下一个命令。省略 STEP 时的增量为+1。 可进一步在 FOR~NEXT 的循环中描述 FOR~NEXT。这称为“嵌套结构”。在嵌套结构中，变量名必须是各不相同的。此外，嵌套数最多为 4。 使用 FOR 语句时，需要注意以下事项。 ① 在同一 FOR~NEXT 循环内，不能变更 FOR 语句中所使用的变量。 ② 在 FOR~NEXT 循环内，不能描述跳出循环的命令（GOTO，RETURN 等）。 ③ 增量设定为 0 时，由于变量值不变，将无法跳出循环。 ④ 从初始值到结束值的符号与增量的符号不同时，不执行循环即结束。 ⑤ 进行逆转换时，一定会记录增量。 ⑥ 增量不为 1 时，在变量值超过结束值的时候结束循环。 ⑦ 变量为整数变量时，如将增量设定为小于 1 的小数，由于变量值不变（因为小数会被舍弃），所以将无法跳出循环。 ⑧ 如增量中含有小数点后的数字，变量值相加时有可能会产生误差，不能正确动作。 以上情况在编译时不会检测到错误，还请注意。 【例】FOR V1%=0 TO 100 FOR V2%=0 TO 10 V3%=V1%*10+V2% V1%=2 .....禁止 RETURN .....禁止 NEXT NEXT

命令	概要
WHILE 条件表达式 ENDW	<p>在条件成立期间，将重复执行 WHILE 与 ENDW 循环内所描述的命令。 可进一步在 WHILE~ENDW 的循环中描述 WHILE~ENDW。嵌套数最多为 4。</p> <p>例： V1% = 0 WHILE V1% &lt; 10 ... ... V1% = V1% + 1 ENDW</p>
IF 条件表达式 ELSEIF 条件表达式 ELSE ENDIF	<p>如 IF 语句的条件成立，将执行 IF~ELSEIF 间的命令。 如 IF 语句的条件不成立，而 ELSEIF 语句的条件成立，执行 ELSEIF~ELSE 间的命令。 如 IF 语句的条件不成立，且 ELSEIF 语句的条件也不成立时，执行 ELSE~ENDIF 间的命令。</p> <p>可进行如下的描述。</p> <p>例 1： IF 条件表达式  ENDIF</p> <p>例 2： IF 条件表达式  ELSEIF 条件表达式  ENDIF</p> <p>例 3： IF 条件表达式  ELSE  ENDIF</p> <p>例 4： IF 条件表达式  ELSEIF 条件表达式  ELSEIF 条件表达式  ELSE  ENDIF</p> <p>还可以进一步在 IF~ENDIF 中描述 IF~ENDIF。嵌套数最多为 4。</p>



命令	概要
SWITCH 表达式 CASE 整数常数 BREAK ENDS	<p>执行与 SWITCH 语句表达式的值一致的整数常数的 CASE 语句到 BREAK 间的命令。</p> <p>例： SWITCH V1% CASE 1 ... BREAK CASE 2 CASE 3 ... BREAK CASE ... BREAK ENDS</p> <p>在该示例中， V1%的值为 1 时，执行从 CASE 1 到最初的 BREAK 间的命令。 V1%的值为 2 时，执行从 CASE 2 到第 2 个 BREAK 间的命令。 V1%的值为 3 时，执行从 CASE 3 到第 2 个 BREAK 间的命令。 V1%的值非 1、2、3 时，执行从 CASE 语句到最后的 BREAK 间的命令。</p> <p>注意： 从最先描述的 CASE 语句开始按顺序判断。 对于例 1，不管 V1%的值如何，都将执行从 CASE 到最初的 BREAK 间的命令。</p> <p>例 1： SWITCH V1% CASE ... BREAK CASE 1 ... BREAK CASE 2 ... BREAK ENDS</p> <p>对于例 2，当 V1%的值为 1 时，将执行从最初的 CASE 1 到最初的 BREAK 间的命令。</p> <p>例 2： SWITCH V1% CASE 1 ... BREAK CASE 1 ... BREAK CASE 2 ... BREAK CASE ... BREAK ENDS</p>
ON 整数变量 GOTO 行号码 / 标签,行号码 / 标签...	<p>将控制转移到与以条件表达式表示的数值相同的顺序所写的行号码或标签。顺序左起依次为 1, 2, 3....</p> <p>【例】ON V1% GOTO 100, 200, 300 在该示例中，V1%为 1 时，控制转移到第 100 行，为 2 时，控制转移到第 200 行。行号码或标签最多可描述 10 个。</p>
STOP	停止程序的执行。

命令	概要
END	结束程序。在连续再生模式下执行再生用程序时，将从程序的最初开始重新执行。 每 1 个程序中必须描述 1 个以上的 END 命令。

2.8.5 命令语句

使机器人动作的 MOVE 命令及使输出信号 ON 的 SET 命令等语句实际上是程序的中心，机器人语言的程序基本上由这些命令语句组成。  
是非常重要的语句，有很多命令。

## 2.9 用户程序

可以在程序内描述用户程序，并作为子程序调用使用。由此，所有相同的处理可作为带有参数的函数来处理，从而提高了可读性，且易于修正。

```

DIM intOut[16] AS INTEGER
'Output the condition of the I1 through I8
FOR L1%=1 TO 8 STEP 1
  CallProc intOut[L1%] = FromInToOut(L1%)           (1)
NEXT

' Output the condition of the I31 through I38
FOR L2%=31 TO 38 STEP 1
  CallProc intOut[L1%] = FromInToOut(L2%)           (2)
  L1% = L1% + 1
NEXT
END

'Output the condition of the input signals
UsrProc FromInToOut(intSigNo AS INTEGER) AS INTEGER (3)
  DIM intRet AS INTEGER
  intRet = I[intSigNo]
  SETM O[intSigNo], intRet
  RetProc FromInToOut = intRet                       (4)
EndProc                                              (5)

```

- (1) (2) 调用用户程序  
调用用户程序。
- (3) (声明)用户程序  
进行用户程序的声明。可以指定参数及返回值的数据类型。
- (4) 返回用户程序  
指定当用户程序中未指定返回值时的返回值。
- (5) 结束用户程序  
结束用户程序，返回调用源。

图 2.9.1 用户程序的说明（例）

请按照顺序制作用户程序。

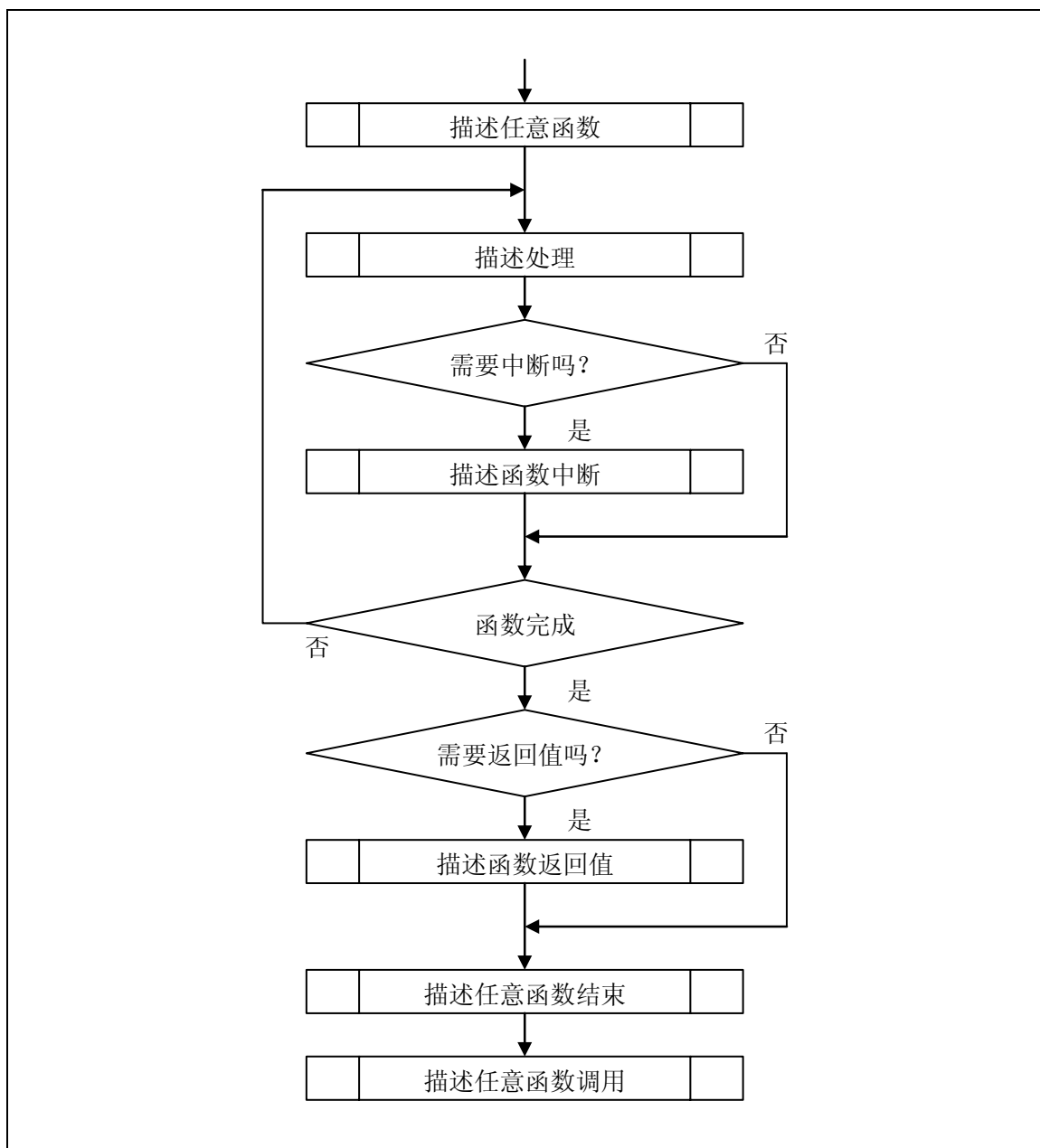


图 2.9.2 用户程序的制作步骤

2.9.1 FN802 用户程序

声明用户程序。

表 2.60 用户程序

格式	UsrProc 用户程序名（参数）返回值
函数名	函数名最多可以指定 32 个字符。 👉 请参照“表 2.28 变量名的可使用字符”。  首字符必须为英文字母。 英文字母区分大小写。 不能使用相同的变量名、函数名及保留字。
参数	最多可以指定 10 个，但请仅指定所需要的数量。 参数请用逗号(,)隔开。 不需要参数时请省略。 【示例】 intSigNo AS INTEGER…不需要 DIM。  用任意变量进行指定。 也可以指定数组。 作为参数递交的原始数据不会被更改。
返回值	可以指定返回值。不需要返回值时请省略。 不可以指定数组。 【示例】 AS INTEGER…不需要 DIM、变量名。  仅指定任意变量的数据类型。
示例	【无参数、无返回值】 UsrProc SignalCheck()  【仅有参数】 UsrProc SignalCheck(intInOut AS INTEGER)  【仅有返回值】 UsrProc SignalCheck() AS INTEGER  【参数、返回值皆有】 UsrProc SignalCheck(intInOut AS INTEGER, intSigNo AS INTEGER) AS INTEGER
功能	FN802 请参照用户程序。

2.9.2 FN803 退出用户程序

中途中断用户程序，返回调用源。

表 2.61 退出用户程序

格式	ExitProc
功能	FN803 请参照退出用户程序。



- 用户程序中需要有返回值时，必须在中断前设定。

### 2.9.3 FN804 结束用户程序

结束用户程序，返回调用源。

表 2.62 结束用户程序

格式	EndProc
返回值	可以指定返回值。不需要返回值时请省略。 【示例】 AS INTEGER...不需要 DIM、变量名。  仅指定任意变量的数据类型。
功能	FN804 请参照结束用户程序。

重点

- 用户程序中需要有返回值时，必须在结束前设定。

### 2.9.4 FN805 返回用户程序

设定用户程序的返回值。

表 2.63 返回用户程序

格式	RetProc 用户程序名 = 返回值
用户程序名	指定进行了用户程序声明的函数名。
返回值	以进行了用户程序声明的返回值的类型指定变量。
示例	UsrProc SignalCheck(intSigNo As Integer) As Integer Dim intStatus; intStatus = I[intSigNo] RetProc SignalCheck = intStatus EndProc
功能	FN805 请参照返回用户程序。

重点

- 用户程序中需要有返回值时，必须在中断前设定。

重点

- 目前无法将返回值指定为常数值。

2.9.5 FN806 调用用户程序

调用用户程序。

表 2.64 调用用户程序

格式	CallProc 代入 = 用户程序名（参数，...）
代入	进行调用的用户程序存在返回值时，指定代入该返回值的变量。 用户程序无返回值时省略。省略时，=也不再需要。
用户程序名	指定进行调用的用户程序名。
参数	进行调用的用户程序存在参数时，指定向参数设定值的变量。
示例	<div>【无参数、无返回值】 CallProc SignalCheck()</div> <div>【仅有参数】 CallProc SignalCheck(V1%)</div> <div>【仅有返回值】 CallProc V1% = SignalCheck()</div> <div>【参数、返回值皆有】 CallProc V10% = SignalCheck(V1%, V2%)</div>
功能	FN806 请参照调用用户程序。

重点

- 必须根据进行调用的用户程序，正确设定各种参数。

重点

- 目前无法将参数指定为常数值。

## 示例程序

将输入信号的状态从 BIN 转换为 BCD 的示例函数。

```

`Convert BIN to BCD
`IN : BIN data
`OUT : BCD data
UsrProc BinToBcd(intBIN AS INTEGER) AS INTEGER
  DIM intBCD AS INTEGER
  DIM intData AS INTEGER
  intBCD = (intBIN AND &H0F) MOD 10
  intData = intBIN / 16
  intBCD = intBCD + (((intData AND &H000F) MOD 10) * 10)
  intData = intBIN / 256
  intBCD = intBCD + (((intData AND &H000F) MOD 10) * 100)
  intData = intBIN / 4096
  intBCD = intBCD + (((intData AND &H000F) MOD 10) * 1000)

  RetProc BinToBcd = intBCD
EndProc

`Convert BCD to BIN
`IN : BCD data
`OUT : BIN data
UsrProc BcdToBin(intBCD AS INTEGER) AS INTEGER
  DIM intBIN AS INTEGER
  DIM intData AS INTEGER
  intBIN = (intBCD / 1000) * 4096
  intData = intBIN MOD 1000
  intBIN = intBIN + ((intData / 100) * 256)
  intData = intBIN MOD 100
  intBIN = intBIN + ((intData / 10) * 16)
  intData = intBIN MOD 10
  intBIN = intBIN + intData
  RetProc BcdToBin = intBIN
EndProc

```

图 2.9.3 示例程序



# 第3章 程序的编辑

本章将对机器人语言程序的编辑方法进行说明。

3.1 在个人电脑上编辑.....	3-1
3.1.1 编辑时的注意事项 .....	3-1
3.1.2 将机器人语言程序拷贝到本控制装置 .....	3-1
3.2 在悬式示教作业操纵按钮台上编辑.....	3-2
3.2.1 基本操作 .....	3-2
3.2.2 便利的编辑功能 .....	3-5
3.2.3 简单地输入基本命令 .....	3-6
3.2.4 检索字符串 .....	3-7
3.2.5 替换字符串 .....	3-8
3.3 制作姿势文件.....	3-9
3.3.1 姿势文件的概要 .....	3-9
3.3.2 制作姿势文件 .....	3-10
3.3.3 修正姿势文件 .....	3-12
3.3.4 显示姿势文件一览 .....	3-12
3.3.5 （注意事项）姿势文件的更新 .....	3-13
3.3.6 姿势文件保存命令 .....	3-14



### 3.1 在个人电脑上编辑

可使用个人电脑上的文本编辑器制作或编辑机器人语言程序（源程序）。

#### 3.1.1 编辑时的注意事项

文件命名规则	请参照“第1章 概要”。
编辑方法	请参照所使用的个人电脑及文本编辑器的说明书。
文件大小	请限制在 64KB(65534 字节)内。
语句、行、字符的注意事项	1 行最多 254 字符、1 个程序最多可描述 9999 行。 英文字符不区分大小写。除说明及字符串变量以外，均请使用半角字符。 详情请参照“第2章 语法”。
其他	制作时，全角请使用 Shift_JIS，换行代码请使用 CR+LF。

#### 3.1.2 将机器人语言程序拷贝到本控制装置

请在个人电脑上制作的机器人语言程序通过以下任一方法拷贝到本控制装置的内部存储器中。  
（使用 USB 存储器较为简单）

**(1) 使用 USB 存储器拷贝**

使用 USB 存储器拷贝文件时，请使用<维修>-[7 文件操作]-[1 拷贝]画面。详细的操作方法请参照下面的操作说明书。

 “基本操作篇”第6章

**(2) 将个人电脑及本控制装置连接到以太网，传送文件**

请使用以太网线缆连接您所使用的个人电脑及本控制装置，并使用个人电脑上的 FTP 客户端软件传送文件。详细的操作方法请参照下面的操作说明书。

 “设定篇”第8章



FTP 客户端软件可使用市售软件及免费软件等。

上述任一情况下，都请务必将机器人语言程序保存到如下所示文件夹中。不在该文件夹中的程序将被忽略。

\\内部存储器\\WORK\\PROGRAM\\

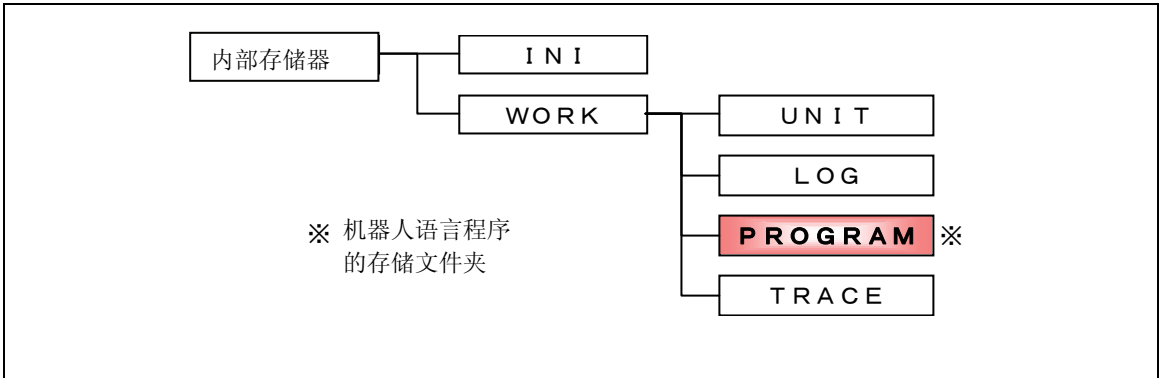
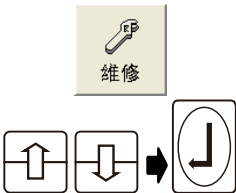


图 3.1 机器人语言程序的存储文件夹

## 3.2 在悬式示教作业操纵按钮台上编辑

使用“ASCII 文件编辑功能”，可在本控制装置的悬式示教作业操纵按钮台上直接编辑机器人语言文件。适合于简单修正程序的描述错误等。

### 3.2.1 基本操作

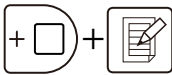


1 按下 f 键<维修>。  
>> 显示各种维修的菜单画面。

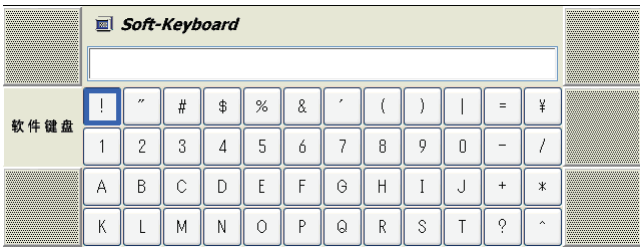
2 将光标对准[15 ASCII 文件编辑]，按下[Enter]。  
>> 显示如下的[ASCII 文件编辑]画面。



※ 列表中仅显示可使用本功能编辑的文件。可编辑的文件是指机器人语言文件，即如同“SRA166-1-A. 010”及“NB4-02-A. 010”一样在文件末尾附有表示文本形式的“-A”，且扩展名为“数字（程序号码）”或“TXT”的文件。

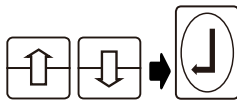


3 【制作新文件时】  
将光标对准“文件名”，输入新制作的文件名。  
同时按下[动作可能]与[编辑]，显示软键盘，毫无省略地输入所有字符串。（例：SRA166-1-A. 010）



4 字符输入结束后，按下 f12<确定>。  
>> 返回[ASCII 文件编辑]画面。



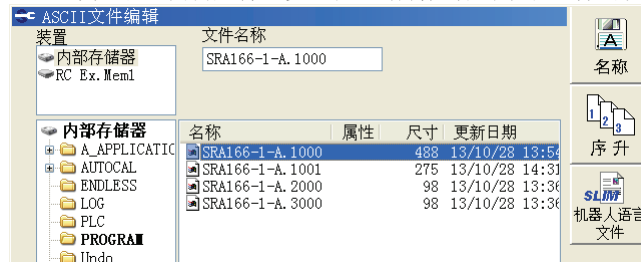


5

### 【选择已制作好的机器人语言文件时】

机器人语言文件已经存储到内部存储器时，指定“装置”为“内部存储器”、文件夹为“PROGRAM”。(\\WORK\\PROGRAM)

>> 显示如下可编辑文件一览画面，请将光标对准欲选择的程序，按下[Enter]键。



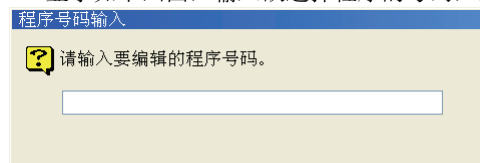
当机器人语言文件存储在内部存储器以外的设备上时，请选择目标设备、文件夹，以同样方式选择编辑文件。



### 【通过号码选择机器人语言文件时】

在“装置”选择“内部存储器”、文件夹选择“PROGRAM”的状态下，通过指定程序号码，可选择当前单元的机器人语言程序。

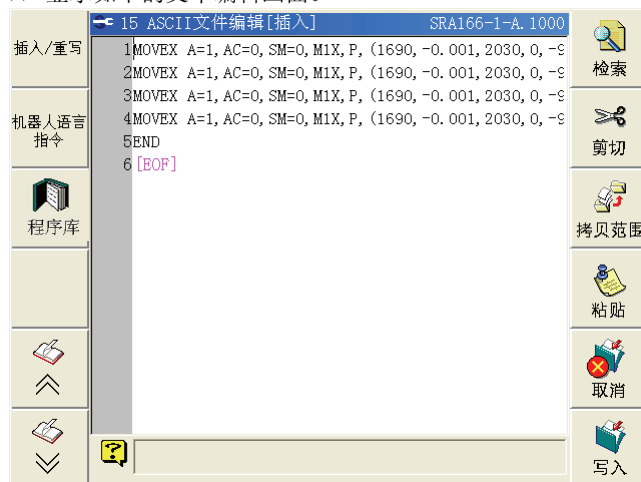
>> 显示如下画面，输入欲选择程序的号码，按下[Enter]键。



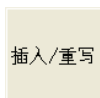
6

确认所显示的“文件名”，如正确，按下 f 键<执行>。

>> 显示如下的文本编辑画面。



显示示例为选择了已经制作完的程序时的情况。指定了新的程序号码时，中央的编辑画面上什么也不显示。



7

每次按下 f 键<插入/重写>，可切换“插入模式”与“重写模式”。

>> 处于“插入”还是“重写”，显示在标题栏上。

“插入”： 15 ASCII文件编辑[插入] SRA166-1-A.1000  
“重写”： 15 ASCII文件编辑[重写] SRA166-1-A.1000

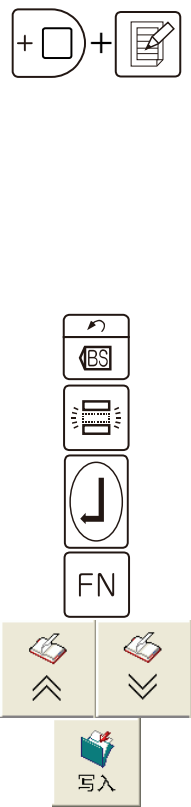


8

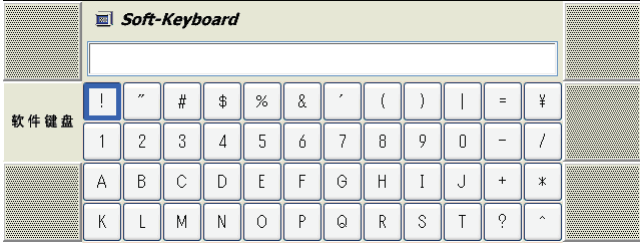
使用[↑][↓][←][→]键，将光标移动到欲编辑的位置（行）。此外，同时按下[动作可能]与[↑][↓]键，可以翻页。

9

输入数值时，请使用[数值输入]键（[0]～[9]）。



10 输入字母等非数值字符时,需同时按下[动作可能]键与[编辑]键,显示软键盘后输入。



字符输入结束,按下 f 键<确定>,返回[ASCII 文件编辑]画面。

11 按下[BS]键,可删除光标位置左侧的 1 个字符。

12 按下[删除]键,可删除光标位置右侧的 1 个字符。

13 按下[Enter]键,可输入换行 (CR+LF)。  
>> 实际上不显示 CR+LF。显示换行后,光标移动到新的行。

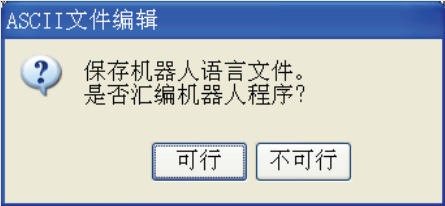
14 按下[FN]键,可输入空格。

15 按下 f 键<翻页>,可跳转到各个文件的开头或末尾。

16 编辑结束后,按下 f 键<写入>。  
>> 编辑的内容写入文件。不检查语法错误等,直接写入文件。此外,由于不保留备份文件,执行时请注意。文件保存完成后,结束[ASCII 文件编辑]画面。

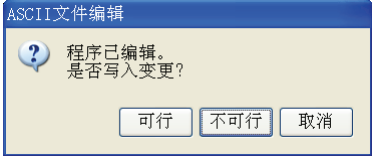
重点

在[内部存储器]的[WORK\PROGRAM]文件夹内,制作/修正了机器人语言文件时,在写入时将显示如下的编译执行确认信息。在编译执行确认信息上,如选择[可行],则执行编译作业。



17 中断编辑,不保存文件时,请按下 f 键<取消>或[复位/R]键。

>> 显示如下的确认弹出信息。



如选择“不可行”,则放弃所有已编辑内容,不保存文件,返回到各种维修的菜单画面,结束编辑。如选择“取消”,则弹出信息消失,返回到[ASCII 文件编辑]画面,可继续进行编辑。

重点

使用[ASCII 文件编辑]变更了机器人语言程序时,请在[维修]→[9 程序转换]→[8 语言转换]画面中进行编译作业,制作执行形式的程序。如不进行编译,变更部分将不会反映到执行形式的程序中。

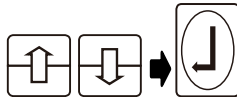
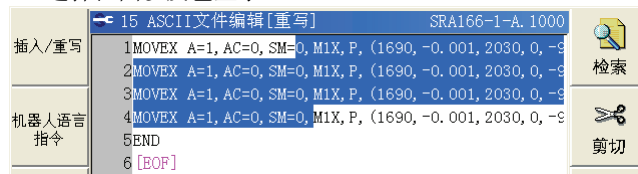
### 3.2.2 便利的编辑功能

#### 一次删除多个行



1 按下 f 键<剪切>，将光标对准欲删除的行。

>> 选择范围以反色显示。



2 使用[上下光标]选择删除范围，按下[Enter]键。

>> 可使用[上下光标]选择多个删除范围。所选择的范围以反色显示。如按下[Enter]键，反色显示的行将全部被删除，后面的行将上移，重新显示。中途欲放弃时，按下[复位/R]。

#### 一次移动多个行



1 按照与删除同样的要领确定剪切范围，将光标对准移动目的地的后面，按下 f 键<粘贴>。

>> 所删除的多个行插入到光标行的前面。

#### 一次拷贝多个行



1 按下 f 键<拷贝范围>，按照与删除同样的要领，确定欲拷贝的范围。

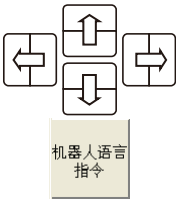


2 将光标移动到拷贝目的地的后面，按下 f 键<粘贴>。

>> 所选择的多个行将插入到光标行的前面。

3.2.3 简单地输入基本命令

本控制装置不配置键盘，输入命令时较花费时间。  
因此，对 IF/THEN/ELSE 等使用频率较高的命令，提供了可简单地记录的方法。  
可使用 f 键以一键输入方式记录总计 24 种关键字。



1 将光标移动到欲输入关键字处。

2 按下 f 键<机器人语言指令(命令)>。

>> 最初 12 个使用频率较高的关键字将显示在 f 键上。

15 ASCII文件编辑[插入] SRA166-1-A.1000		
IF	1MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	GOTO
	2MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	
THEN	3MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	WAIT
	4MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	
FOR	5END	MOVE
	6[BOF]	
TO		CALL
OR		INPUT
END	选择命令的文字列。应按下功能键。	STOP



3 按下[动作可能]键。

>> 其余 12 个将显示在 f 键上。

15 ASCII文件编辑[插入] SRA166-1-A.1000		
NOT	1MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	GOSUB
	2MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	
ELSE	3MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	DELAY
	4MOVEX A=1, AC=0, SM=0, MIX, P, (1690, -0.001, 2030, 0, -S	
NEXT	5END	JMP
	6[BOF]	
STEP		OUT
AND		PRINT
RETURN	选择命令的文字列。应按下功能键。	EXIT



4 按下目标 f 键（例如 f 键<IF>）。

>> f 键返回到原来编辑画面的状态，关键字显示在光标位置。  
>> 欲取消时，按下[复位/R]键。

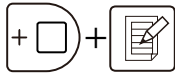
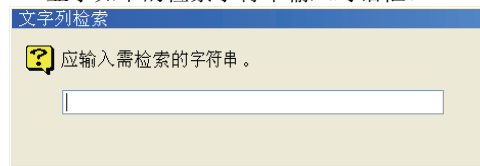


### 3.2.4 检索字符串

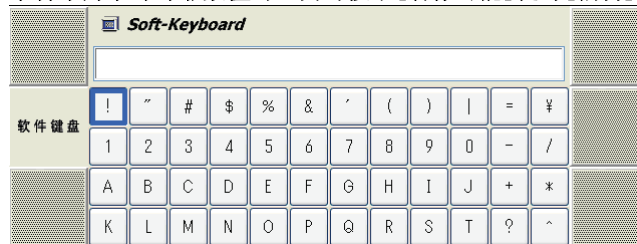
在编辑中的文件范围内，可检索任意字符串。



- 1 按下 **f** 键<检索>。  
>> 显示如下的检索字符串输入对话框。



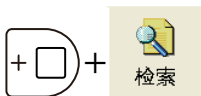
- 2 输入检索字符串。  
字符串为数值时，使用[数值输入]键直接输入。  
字符串为字母等非数值时，同时按下[动作可能]键与[编辑]键，显示软键盘进行输入。



字符串输入结束后按下 **f** 键<确定>，返回到检索字符串输入对话框。



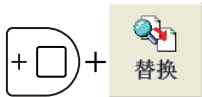
- 3 字符串确定后按下[ENTER]。  
>> 检索开始。自程序开头位置开始检索字符串，光标将出现在发现字符串的位置，显示内容将被更新。例如，指定检索“MOVEX”字符串时，光标移动到所发现字符串的开头的“M”位置。



- 4 如欲继续进行检索，同时按下[动作可能]键与 **f** 键<检索>。  
>> 光标移动到所发现的下一个“MOVEX”字符串上。检索时不从程序的末尾返回到开头位置。

3.2.5 替换字符串

在编辑中的文件范围内，可一次替换任意的字符串。



1

同时按下[动作可能]键与 f 键<替换>。

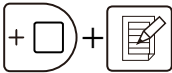
>> 显示如下的检索字符串输入对话框。

替换字符串

?

应输入需检索的字符串。

P



2

输入检索字符串。

字符串为数值时，使用[数值输入]键直接输入。

字符串为字母等非数值时，同时按下[动作可能]键与[编辑]键，显示软键盘进行输入。

Soft-Keyboard

软件键盘

! " # \$ % & ' ( ) | = ¥

1 2 3 4 5 6 7 8 9 0 - /

A B C D E F G H I J + \*

K L M N O P Q R S T ? ^

3

字符串确定后按下[ENTER]。

>> 显示如下替换字符串输入对话框。

替换字符串

?

请输入要替换的字符串。

L

4

输入替换字符串。

字符串为数值时，使用[数值输入]键直接输入。

字符串为字母等非数值时，同时按下[动作可能]键与[编辑]键，显示软键盘进行输入。

>> 请与检索字符串时一样进行输入。

5

字符串确定后，按下[ENTER]。

>> 替换开始，设定为检索字符串的字符串将被替换为替换字符串。

15 ASCII文件编辑[插入] SRA166-1-A.1000

插入/重写

1MOVEX A=1, AC=0, SM=0, M1X, L, (1690, -0.001, 2030, 0, -S

2MOVEX A=1, AC=0, SM=0, M1X, L, (1690, -0.001, 2030, 0, -S

3MOVEX A=1, AC=0, SM=0, M1X, L, (1690, -0.001, 2030, 0, -S

4MOVEX A=1, AC=0, SM=0, M1X, L, (1690, -0.001, 2030, 0, -S

5END

6 [BOF]

机器人语言指令

Library

↑

↓

检索

剪切

拷贝范围

粘贴

取消

写入

>> 欲取消时，按下[复位]键。

3-8

### 3.3 制作姿势文件

#### 3.3.1 姿势文件的概要

“姿势文件”是仅由机器人的位置/姿势信息（=姿势变量）构成的数据文件。

在机器人语言中，虽然可以在 MOVEX 命令中使用直接数值描述机器人的位置/姿势，但实际上由于装配误差等，完全按照设计图正确移动机器人是很困难的。此外，因布局变更等导致需要修正位置时，以及希望在同一示教点上反复使用时，一个一个地修正原始的源程序中的移动命令非常花费时间。

为解决这一问题，在机器人语言中可以将多个姿势变量汇集在另外的文件中进行管理。这就是姿势文件。包含在姿势文件中的各个姿势变量可以通过实际移动机器人，记录其位置/姿势来完成制作。之后只需要选择所要使用的姿势文件的号码，以 MOVEX 命令指定其中的姿势变量号码，即可移动机器人。

（例）

```
REM "Pose file No.10"  
USE 10  
REM "Pose variable P1,P2,P3"  
MOVEX A=1,M1J,P,P1,R=10.0,H=1,MS  
MOVEX A=1,M1J,P,P2,R=10.0,H=1,MS  
MOVEX A=1,M1J,P,P3,R=10.0,H=1,MS
```

关于文件名，如机器人型号为“SRA166-01”，程序号码为“10”时，姿势文件的名称即为“SRA166-1-P.010”。

（👉 还请参照“1.1.2 特征及注意事项”中的文件名的项目）

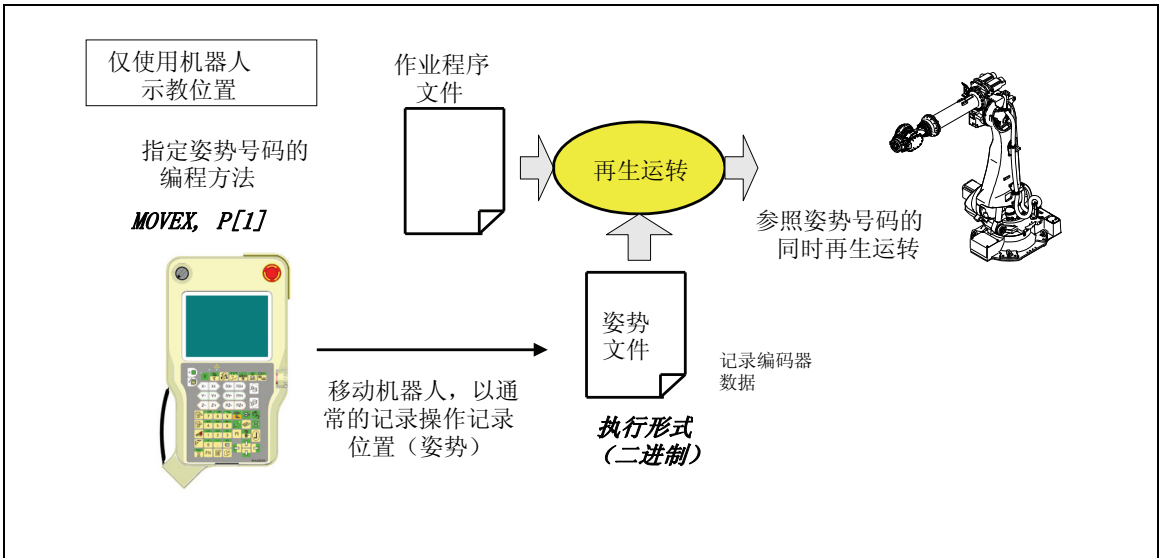


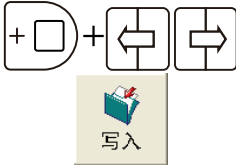

图 3.2 使用姿势文件的机器人语言程序

编译时不参照姿势文件。再生运转时参照姿势文件。此外，机器人语言程序与姿势文件不需要一一对应。可根据工件种类，准备多个姿势文件，使用“USE”命令选择需要的姿势文件进行再生。

3.3.2 制作姿势文件

制作姿势文件时，首先请按照以下操作，进入“姿势记录模式”。在姿势记录模式下，按下[记录]键 (=位置记录操作)，此时机器人的姿势将不记录在作业程序的步骤数据中，而记录在姿势文件内的姿势变量中。在姿势记录模式下，仅允许这一位置记录操作，不能记录应用命令。

进入姿势记录模式

- 
- 1 选择示教模式。

2 按下 f 键<示教、再生条件>。  
按下 f 键<维修>，选择[1 示教、再生条件]也可以选择相同的菜单。

3 将光标对准“13 姿势记录”。

4 同时按下[动作可能]键与[←][→]键，将姿势记录设定为“有效”。

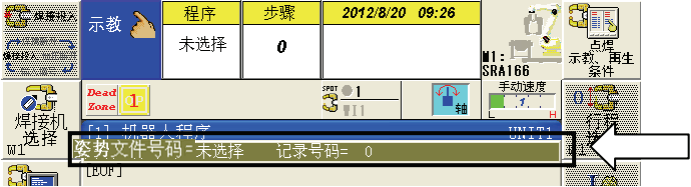
5 按下 f 键<写入>。  
>> 之后仅可进行姿势文件的编辑。可以编辑通常的作业程序。  
(即使切断电源也可以保持该状态。要离开姿势记录模式时，请切换为“无效”)

6 如返回模式画面，[程序监视器]画面的记录状态显示部分显示的是“姿势文件号码”/“记录号码”。这里可以区分姿势记录的“有效”/“无效”。

【姿势记录“无效”时】



【姿势记录“有效”时】

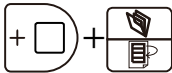


重点

画面上部所显示的“程序”及“步骤”框，与姿势文件号码及姿势变量号码无关，还请注意。选中（编辑中）的姿势文件号码及姿势变量号码（记录号码）可通过以下部分进行确认。

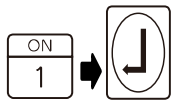
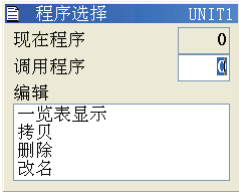
姿势文件号码= 未选择      记录号码= 0

在姿势文件内记录姿势变量



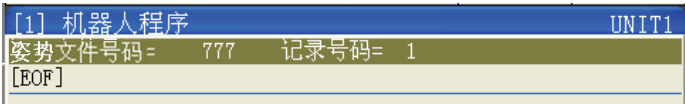
1 同时按下[动作可能]键与[程序/步骤]键。

>> 打开[姿势文件选择]画面。



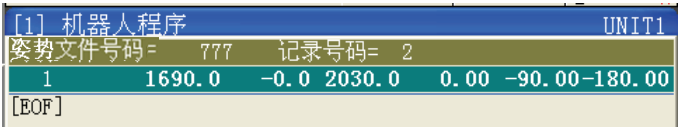
2 输入姿势文件号码，按下[Enter]键。

>> 在“姿势文件号码”/“记录号码”的显示栏中，显示所选择的姿势文件号码。  
图中的例子选择的是姿势文件 777 号。

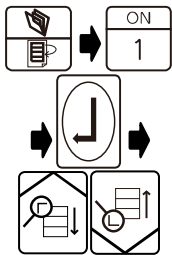
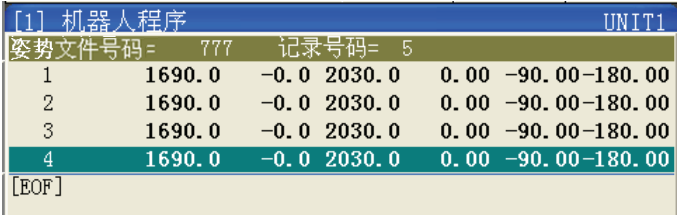


3 使运转准备 ON，使用[轴操作]键移动机器人，按下[记录]键。

>> 登录一个姿势变量，显示如下。



4 每按下一次[记录]键，都将记录新的姿势变量。



5 通过步骤选择操作选择姿势变量号码、进行前进检查/后退检查，可以确认该姿势变量下的机器人位置。

6 最后的姿势记录结束后，请前进到“结束姿势记录模式”。

结束姿势记录模式



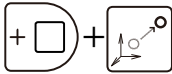
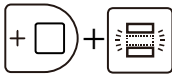
1 在[示教、再生条件]画面中，将光标对准“13 姿势记录”，同时按下[动作可能]键与[←][→]键，将姿势记录设定为“无效”。



2 按下 f 键<写入>。

>> 至此画面返回到通常的示教模式。

3.3.3 修正姿势文件



- 1
- 同时按下[动作可能]与[删除]，可以删除中途的姿势。  
但是，即使删除中途的姿势，其后的姿势号码也不会递补。被删除的姿势号码将成为空缺号码。
- 2
- 选择要进行位置修正的姿势变量号码，同时按下[动作可能]与[位置修正]，可以修正姿势的位置。

(补充) 在姿势记录模式下，通过以下操作也可获得相同的结果。

- 按下[记录]键
- 按下[动作可能]+[记录]键

- 3
- 可以在显示器编辑画面上修正位置、姿势。  
按下[编辑]键，显示[显示器编辑]画面，使用[数值输入]键直接输入值 (X, Y, Z, R, P, Y)。

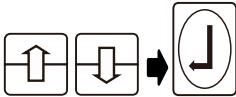
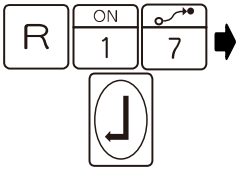
[1] 机器人程序							UNIT1
1:SRA166-01	J1/X	J2/Y	J3/Z	J4/A	J5/B	J6/C	
0 [START]							
1	1690.0	-0.0	2030.0	0.00	-90.00	-180.00	
2	1690.0	100.0	2030.0	-34.20	-90.00	-145.80	
3	1690.0	-0.0	2030.0	0.00	-90.00	-180.00	
4	1690.0	-0.0	2030.0	0.00	-90.00	-180.00	
[EOF]							

重点

- 显示器编辑画面中虽然是 (X, Y, Z, roll, pitch, yaw) 的形式，但内部数据是各轴的编码器值的形式。因此，通过数值输入进行较大的变更时，有可能会变成意想不到的姿势。进行修正时请使用检查前进操作确认机器人的姿势。
- 姿势变量（即使显示为 (X, Y, Z, roll, pitch, yaw)）在由控制装置进行处理时，在内部被转换为各轴的编码器值。因此，使用姿势变量执行 MOVEX 命令时 CONF 指定无效。

3.3.4 显示姿势文件一览

可以仅显示姿势文件的一览。  
这在查看几号文件空缺时很有用。此外，也可从一览表中选择已记录好的姿势文件。



- 1
- 将姿势记录设定为“有效”。
- 2
- 输入快捷代码 (R17)，按下[Enter]键。  
>> 显示如下姿势文件一览。
- 3
- 将光标对准任意的姿势文件，按下[Enter]键。  
>> 显示所选择的姿势文件。  
如按下[复位/R]键，将返回到原来的示教画面。

姿势文件一览显示				UNIT1
姿势文件一览表				UNIT1
姿势文件号码	号码	步骤	说明	
SRA166-1-P	777	4		序升

3.3.5 （注意事项）姿势文件的更新

这里将说明在**姿势记录模式**（☞参照“3.3.2 制作姿势文件”）下，进行姿势运算及代入到姿势变量时的机器人的动作及访问姿势文件。  
例如如下所示的执行包含姿势变量代入表达式的“作业程序 888”。其结果是“姿势文件 777”的“姿势变量 1~3”发生变化，机器人按照这些新的“姿势变量 1~3”进行动作。但是，此时新的“姿势变量 1~3”的数值仅保存到本控制装置的工作存储器中，“**姿势文件 777**”本身并未更新，还请注意。



（即使返回示教模式，显示姿势文件 777，姿势变量 1~3 也仍保持原来的值）



重点

执行“作业程序 888”的结果（通过代入语句被重写的姿势变量），仅保存在控制装置的工作存储器中。这里显示在监视器上的姿势变量为姿势文件的内容，不反映工作存储器中的内容。

但是，如果此时选择“姿势变量 4”，按下[记录]键，则如下所示，在向“姿势变量 4”记录当前的机器人位置的同时，**通过刚才的程序代入到“姿势变量 1~3”中的新值将会覆盖旧值，“姿势文件 777”也将被更新。**



重点


即使只希望记录“姿势变量 4”，但之前再生程序代入命令的结果也将反映出来，因此“姿势变量 1~4”全部都被重写，这一点还请注意。

重要

如上述示例所示，如直接在姿势记录模式的状态下执行作业程序，则姿势变量的状态将变化，监视器显示与机器人的实际动作有可能不一致。因此，在进行包含使用代入命令等对姿势变量的内容进行操作在内的处理时，建议在再生该作业程序之前，先离开姿势记录模式。

### 3.3.6 姿势文件保存命令

“姿势文件保存命令”是通过 USE 将姿势运算及姿势代入的结果保存到指定的姿势文件中的应用命令。

通常，姿势运算及姿势代入的内容不保存到姿势文件中（ 参照“3.3.5（注意事项）姿势文件的更新”），因此需要使用“姿势文件保存命令（FN74:POSESAVE）”，将运算及代入的结果保存到姿势文件中。

例如如下所示，执行包含“姿势变量代入表达式”与“姿势文件保存命令（FN74:POSESAVE）”的“作业程序 888”。

其结果是姿势代入后的“姿势变量 1~3”被保存到“姿势文件 777”中，切换到示教模式时，内容反映到监视器显示中。

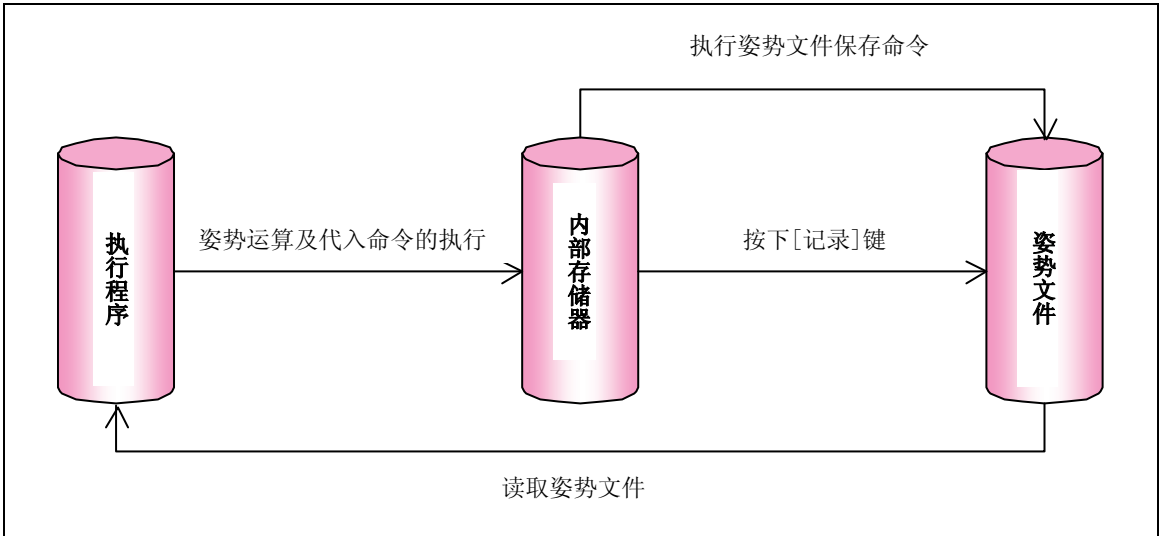


图 3.3 姿势文件更新的结构



## 第4章 程序的编译

---

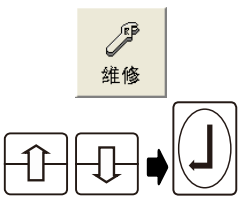
本章说明机器人语言程序的编译方法。

4.1 编译 .....	4-1
4.2 反编译 .....	4-6
4.3 语言转换的强制执行 .....	4-7
4.4 语言转换强制执行的设定 .....	4-8



# 4.1 编译

制作好的机器人语言程序（ASCII 文件），必须编译后方可执行。  
编译不仅是将程序转换为机器人可执行的文件，还可发现语法错误。按照以下步骤进行编译。



- 1 按下 f 键<维修>。  
>> 显示各种维修的菜单画面。
- 2 在[维修]菜单中，将光标对准[9 程序转换]→[8 语言转换]，按下[Enter]键。  
>> 显示如下[语言转换]画面。



- 3 在“转换种类”对话框中，选择“语言形式→执行形式”。
- 4 “输出形式”对话框是反编译（“语言形式←执行形式”）时的选项，编译时不指定。



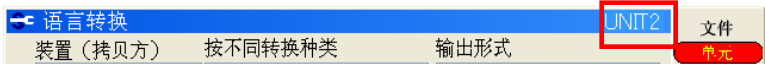
- 5 按下 f7<文件>，切换文件显示。  
请切换至显示目标文件。
- ※ 进行文件切换需要 EXPERT 以上的操作资格。
  - ※ 仅限编译时可用。

文件 单元 全部	显示选中单元的作业程序。
文件 单元 全部	显示所有作业程序。 可显示当前控制装置中尚未登录单元的作业程序。



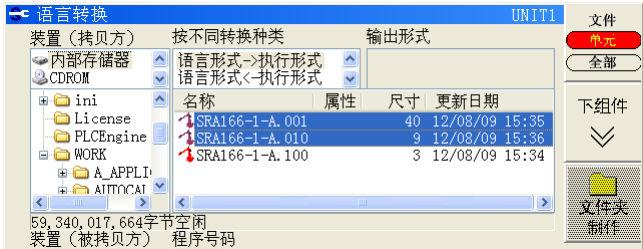
- 6 按下 f8<下组件>，切换单元。  
文件显示为“单元”时，文件一览将被更新。  
此处切换的单元，将用作转换时的单元信息。

>> 标题行的单元被更新。



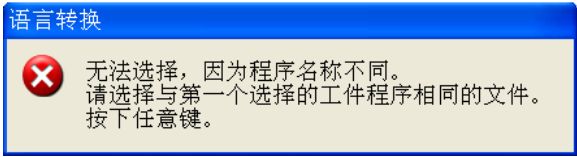
※ 此处切换的单元将成为手动操作的对象单元。

- 7 使用[Enter]键选择要编译的文件。解除选择时，选择要解除的文件，然后按下[BS]键。



文件显示为“全部”，如在“作业程序名称”部分选择不同的文件时，显示如下弹出消息，无法进行选择。

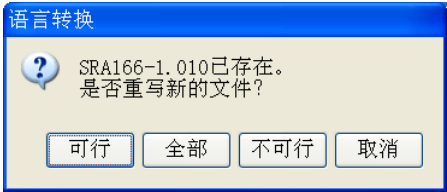
SRA166-1-A.1001  
UNIT2-A.1002



- 8 接下来指定编译端的“装置”、“文件夹”、“文件”等。编译时可以改变程序号码。此时只需将光标对准装置（被拷贝方）的程序号码栏，使用[数值输入]键输入程序号码。但是，选择了多个文件时，无法改变程序号码。此时，将按所选择文件的程序号码生成执行文件。

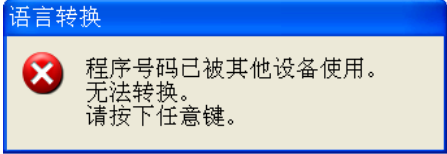


- 9 按下 f 键<执行>。  
>> 开始所选择文件的编译。选择已存在于存储器内的程序号码时，将显示如下弹出信息，请使用[←][→]键选择“可行”/“不可行”，按下[Enter]键。

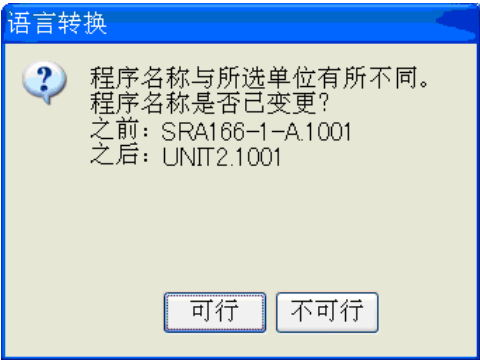


可行	替换。
全部	文件已经存在时全部进行替换。在选择了多个文件时显示，选择之后不再显示同样的信息。
不可行	不替换。进行下一个文件的转换。
取消	中止转换。

当转换后的程序号码已被其他单元的程序所使用时，将显示如下弹出信息，无法进行转换。请先删除文件，或变更程序号码后再进行转换。

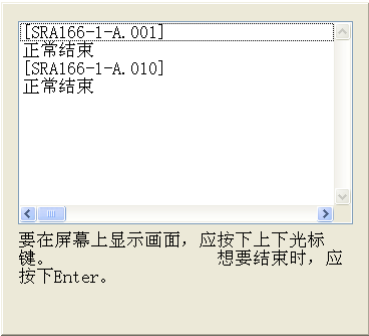


要执行与选中单元的作业程序名称不同的文件时，显示如下弹出信息，请使用 [←][→]键选择后，按下[Enter]键。



可行	变更生成文件的名称。
全部	变更全部生成文件的名称。在选择多个文件时显示，选择之后不再显示同样的信息。
取消	中止转换。

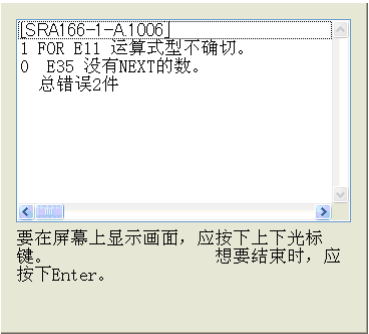
10 编译正确结束时，在画面中央部分将显示“正常结束”的信息。至此，可再生执行形式文件已生成。



【显示内容】

[SRA166-1-A.1000]                      →转换文件名  
正常结束                                      →结果

11 编译中检测到错误时，在画面中央部分显示检测到的所有编译错误。



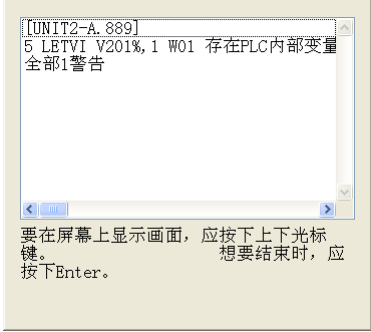
错误内容过多，超过 1 个画面时，可使用[↑][↓]键滚动显示异常信息。

【编译错误的显示例】

[SRA166-1-A.1006]                      →转换文件名  
2 END3                                      →检测到错误的行的行号码与该行的内容。  
无法识别 E18 命令。                      →显示错误的内容。(参照“表 4.1.1 机器人语言编译错误”)

12 检测出错误时，请通过 ASCII 文件编辑功能修正指出的部分，然后再次执行编译。在编译正确完成前，不会生成执行形式的文件。

13 编译中检测到警告时，在画面中央部分显示所检测到的所有编译警告。



【警告显示例】

[SRA166-1-A. 1007]

3 LETVI V201%, 1

正在访问 W01 PLC 内部变量

→转换文件名

→检测到警告的行的行号码与该行的内容。

→显示警告的内容。(☞参照“表 4.1.2 机器人语言编译警告”)

14 仅检测到警告时，将制作执行形式文件。请检查所指出的部分，以确认制作的文件是否可用。

表 4.1.1 机器人语言编译错误

错误内容	说明
E01 一行的字符数过多	请将一行控制在 254 个字符以内。
E02 行号码不正确	请确认行号码或标签名称是否正确。
E03 出现语法错误	请确认是否有语法错误
E04 数值超过范围	应用命令的参数超过了指定的范围。FLOAT、INT 的变量超过输入范围。
E05 计算表达式的描述出现错误	代入表达式的描述中有错误。 <ul style="list-style-type: none"><li>通用函数的参数中出现错误时</li><li>代入表达式缺少右边部分时 (V1%=)</li><li>以运算符结束时 (V1%=1+)</li></ul>
E06 识别符的字符数过多	标签的最大字符数为 16 个字符。
E07 标签不正确	请确认是否使用了标签所不能使用的符号。
E08 标签过多	请减少标签或分割程序。
E09 标签双重定义	请确认是否存在同一名称的标签。
E10 程序大小超过限度	请使用程序调用等方式分割程序。
E11 运算表达式形式不正确	请确认是否向字符串中代入数值。
E12 缺少(	请确认 ( ) 的对应情况。
E13 缺少)	请确认 ( ) 的对应情况。
E14 缺少[	请确认 [ ] 的对应情况。
E15 缺少]	请确认 [ ] 的对应情况。
E16 缺少\$	字符串形式必须有\$。
E17 寄存器不正确	请确认寄存器号码。
E18 无法识别命令	请确认命令名称、函数名称是否有误。
E19 步骤号码不正确	请确认所指定的步骤号码是否存在。步骤号码为 0~999。
E20 程序号码不正确	程序号码为 0~9999。
E21 参数的数量不一致	请确认函数或命令。
E22 缺少 THEN	请确认 THEN。
E23 必须有 ELSE	请确认 IF 语句中是否缺少命令。
E24 一行的程序过长	一行的命令因函数及括号等而过长。
E25 缺少 GOTO	缺少与 ON 命令对应的 GOTO 命令。
E26 插补的描述出现错误	请确认插补参数的设定。
E27 缺少逗号	请确认多个参数的数量是否不足。
E28 缺少=	FOR 语句中缺少代入初始值的部分

错误内容	说明
E29 出现精确度描述错误	请确认精确度号码。
E30 出现速度描述错误	请确认是否超过最大速度。
E31 出现工具描述错误	请确认工具号码是否有误。
E32 MOVE 命令过多(999)	MOVE 命令最多为 999 个。
E33 FOR 中缺少 LET 语句	FOR 语句中缺少代入初始值的部分
E34 FOR 的嵌套过深	FOR 的嵌套最多为 4 个。
E35 NEXT 的数量不一致	请确认 FOR 与 NEXT 的数量。
E36 输入输出装置号码不正确	
E37 缺少 T0	请确认 FOR 语句中是否存在 T0。
E38 必须为 STEP	请确认 FOR 语句是否描述了表示增量的 STEP 以外的命令。
E40 行号码不正确	附有行号码的行与未附有行号码的行同时存在。
E41 FOR 内存在不可使用的命令	FOR~NEXT 循环内存在跳出循环的命令。
E42 出现输送带描述错误	请确认输送带寄存器。
E43 出现 CONF 描述错误	请确认配置指定情况。
E44 无法打开临时文件	转换时，无法打开临时文件。
E45 该速度指定不能逆转换	想要将仅存在于本控制装置的速度指定逆转换为 AW 形式。
E46 该插补形式不能逆转换	想要将仅存在于本控制装置的插补形式逆转换为 AW 形式。
E47 无法打开 ASCII 文件	无法打开转换对象的 ASCII 文件。
E48 无法打开机器人程序	无法打开转换对象的机器人程序。
E49 转换作业被强制结束	按下了强制结束键。
E51 出现加速度描述错误。	请确认加速度指定。
E52 出现平滑度描述错误	请确认平滑度指定。
E53 找不到标签。	
E54 出现机构描述错误	
E55 无法根据姿势数据计算各轴角度	请确认姿势常数的值。
E56 执行文件保存时发生异常	
E58 其他编译处理正在执行中	请等待其他编译结束后再执行。
E59 无法转换该插补形式	
E60 转换表的尺寸超过上限	INCLUDE 命令的转换表尺寸太大。
E61 无法打开文件	无法打开转换表。
E62 MOVE, MOVEJ 不支持 7 轴机器人	
E63 ENDW 的数量不一致	请确认 WHILE 与 ENDW 的数量。
E64 WHILE~ENDW 语句的嵌套过深	WHILE~ENDW 的嵌套最多为 4 个。
E65 流程控制命令使用方法不正确	出现无法使用的命令。
E66 ENDIF 的数量不一致	请确认 IF 与 ENDIF 的数量。
E67 IF~ENDIF 语句的嵌套多深	IF~ENDIF 的嵌套最多为 4 个。
E69 SWITCH~ENDS 语句的嵌套过深	SWITCH~ENDS 的嵌套最多为 1 个。
E70 SWITCH~ENDS 的使用方法不正确	不能用作 SWITCH 语句的下一命令。

表 4.1.2 机器人语言编译警告

警告的内容	说明
正在访问 W01 PLC 内部变量	PLC 内部变量访问用的整数变量、实数变量正在使用中。请确认使用中是否有问题。

## 4.2 反编译

所谓反编译，是指将执行形式的程序转换为机器人语言程序（ASCII 文件）。  
在机器人语言程序中，允许移动命令有多个格式，因此在反编译时，必须进行指定。之后的操作与编译时完全相同。  
不能进行用户任务程序的反编译。  
此外，反编译时，不进行自动缩进处理。

下面仅对与编译不同的操作进行说明。

- 1
- 在“转换种类”对话框中，选择“执行形式→语言形式”。
- 2
- 如将光标移动到“输出形式”对话框上，将显示如下候选内容。使用[↑][↓]键，将光标对准要选择的移动命令形态，按下[Enter]键。

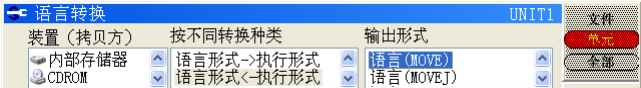


表 4.2.1 移动命令的形态

输出形式	位置表现
MOVEX-X	前端坐标 (X, Y, Z, r, p, y)
MOVEX-J	各轴角度（位置） (J1, J2, ..., J6)
MOVEX-E	各轴编码器值 (E1, E2, ..., E6)

详细内容请参照“2.4.3 姿势常数”或操作说明书的“命令参考”。



**移动命令形态的相关注意事项：**

对由机器人语言转换成执行形式的程序进行反编译时，与在该画面所选择的输出形式（MOVE/MOVEJ/MOVEX 等）无关，该移动命令将转为原有的机器人语言程序中所使用的形式。这是因为执行形式程序内部保留了原有的机器人语言程序的移动命令。

例：  
对于使用 MOVEX-X 制作的机器人语言程序进行编译所制作的执行形式程序，即使指定 MOVEX-J 形式作为输出形式进行反编译操作，仍将以 MOVEX-X 形式进行反编译。



# 4.3 语言转换的强制执行

编译时，即使发生特定的错误（☞参照“表 4.3.1 强制转换错误一览”），也可以强制继续编译。错误的步骤将被置换成说明（REM 命令）。这是仅在编译时（“语言形式”→“执行形式”）有效的功能。

下面仅对与编译不同的操作进行说明。

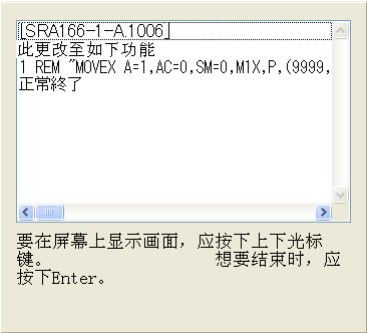


- 1

确认“强制执行语言转换”设定为“有效”。  
※关于设定、请参照☞“4.4 语言转换强制执行的设定”
- 2

选择要编译的文件，按下 f12<执行>。
- 3

编译中执行强制转换时，画面中央部分将显示相关提示信息。如发生强制转换错误以外的错误，则不会生成执行形式的文件。



【强制转换时的显示例】

[SRA166-1-A. 1006]

已置换为以下命令。

1 REM “MOVEX A=1, AC=0, SM=0, ...

正常结束

→转换文件名

→强制转换时的提示信息


→强制转换的行号码与转换后的命令

→结果

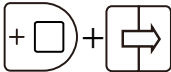
表 4.3.1 强制转换错误一览

错误内容	说明
E55 无法根据姿势数据计算各轴角度	姿势常数的值不正确。

# 4.4 语言转换强制执行的设定

语言转换强制执行的设定，是在发生了特定错误（ 参照“表 4.3.1 强制转换错误一览”）时，是否强制执行转换的设定。按以下步骤进行设定。

变更这一设定需要 *EXPERT* 以上的操作资格。



- 1

按下 f 键<常数设定>。  
>>显示各种常数的菜单画面。
- 2

在<常数>菜单中，选择[5 操作和示教条件]-[1 操作条件]。  

31 强制执行语言转换 ☐ 无效 ☒ 有效
- 3

将光标对准“强制执行语言转换”，按住[动作可能]键的同时，按下[右]，切换为“有效”。  
  
【设定值】  
无效                   → 不进行强制转换。  
有效                   → 进行强制转换。
- 4

按下 f12<写入>。  
保存设定。

## 第5章 命令

---

所谓命令是指使机器人动作的 MOVEX 命令及使输出信号 ON/OFF 的 SETM 命令等，实际上作为程序的中心内容的语句。

通过悬式示教作业操纵按钮台实施的通常的示教作业，分为使用[记录]键示教的移动命令与使用[FN]键记录的应用命令两种，但在机器人语言程序中，这些全部作为“命令”来处理。

关于各命令的详情以及本章中未记述的命令，请参照本控制装置的在线帮助或操作说明书的“命令参考”，以及各种选购手册等。

5.1	MOVEX (移动命令)	5-1
5.1.1	单机构的情况	5-1
5.1.2	多机构的情况	5-8
5.2	输出信号	5-9
5.3	输入信号	5-10
5.4	步骤转移(跳跃)/步骤调用	5-11
5.5	程序转移(跳跃)/程序调用	5-12
5.6	系统(单元)外启动/系统(单元)外调用	5-13
5.7	变量及运算	5-14
5.8	移位(移动)	5-15
5.9	姿势强制选择	5-16
5.10	坐标计算/姿势变量相关	5-17
5.11	用户任务	5-18
5.12	任意变量	5-19
5.13	用户程序	5-20
5.14	套接字通信	5-21
5.15	模拟输入输出	5-22
5.16	其他	5-23



5.1 MOVEX（移动命令）

MOVEX 是用于使机器人动作至示教点的最为基础的命令。  
MOVEX 每一个都与示教点对应，机器人朝向该示教点动作。



请综合参照以下的说明。

- 操作说明书“基本操作篇”的“第4章 示教”
- “2.4.3 姿势常数”
- “2.4.4 移位常数”
- “2.5.9 姿势变量”
- “2.5.10 移位变量”
- “2.7.5 姿势运算”

5.1.1 单机构的情况

例如仅有一个6轴机构时，MOVEX 的描述如下所示。

MOVEX A=1P, AC=1, SM=1, F, M1X, P, (1200, 0, 1800, 0, 0, -180), R=10.0, H=1, MS, CONF=0000  
(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11)

No.	符号	名称	值	说明
1	A	精确度	1~8 或 1P~8P	指定内侧旋转动作的精度（精确度水平）。 如省略，则使用之前的数值。 带有 P 时，机器人不进行内侧旋转动作，而在示教点暂时停止（定位动作）。
2	AC	加速度	0~3	是“加速度（D）”的水平。 未指定时视为“0”。
3	SM	平滑	0~3	是“平滑（S）”的水平。 未指定时视为“0”。
4	F	精细动作	-	描述有 F 时精细动作功能有效。 未描述 F 时精细动作功能无效。 请参照操作说明书“精细动作”。
	HM	协调	-	描述有 HM 时协调动作有效。 未描述 HM 时协调动作无效（=同时动作）。 请参照操作说明书“同步动作（协调）”。
5	M1X M1J M1E  M2X M2J M2E  M3X M3J M3E  :	机构指定	-	请指定机构号码和与之对应的姿势常数的形式。  （例） M1X：机构 1 中使用 MOVEX-X 形式。 M1J：机构 1 中使用 MOVEX-J 形式。 M1E：机构 1 中使用 MOVEX-E 形式。 M2X：机构 2 中使用 MOVEX-X 形式。  • 关于姿势常数，请参照“2.4.3 姿势常数”。 • MOVEX-X 形式仅在该机构为机械手（多关节机器人）时使用。 滑块及定位器等辅助轴时无法使用。 • 省略 X、J、E 时视为 X。



使用 MOVEX-X 使机器人动作时，有可能由于异常点等的原因使机器人做出意想不到的姿势。想要完全确定机器人的姿势时推荐使用 MOVEX-J 或 MOVEX-E。



精细动作等一部分功能为选购功能。

No.	符号	名称	值	说明
6	P L C1 C2 LE C1E C2E	插补类型	—	<p>是机器人的 TCP（Tool Center Point）向示教点移动时所使用的“插补类型”。 可使用以下 7 种。</p> <p>P      JOINT      各轴插补（切线插补） L      LIN        直线插补 C1     CIR1      圆弧插补（经过点） C2     CIR2      圆弧插补（终点） LE     S-LIN     固定工具直线插补 C1E    S-CIR1    固定工具圆弧插补（经过点） C2E    S-CIR2    固定工具圆弧插补（终点）</p> <p>JOINT 由于各轴进行动作时相互间不协调，TCP 的轨迹不为直线。</p> <p>LIN 各轴相互协调进行动作，TCP 的轨迹为直线。</p> <p>CIR1 参照自身、面前的点、身后的点这 3 个点生成虚拟圆弧，绘出该圆弧的前半段。</p> <p>CIR2 参照自身与面前的点这 2 个点生成虚拟圆弧，绘出该圆弧的后半段。</p> <p>固定工具插补 (LE, C1E, C2E) 时，以事先所定义的用户坐标系的位置与方向为基准执行插补动作。关于固定工具请参照“FN67 STOOL”以及&lt;维修&gt;[10 用户坐标系登记]的在线帮助。</p>

No.	符号	名称	值	说明
7	-	姿势常数 或 姿势变量	-	<p>使用姿势常数设定示教点。 请参照“2.4.3 姿势常数”。</p> <p><b>MOVEX-X 形式姿势常数</b>            请以(X, Y, Z, roll, pitch, yaw)的形式描述示教点。            MOVEX A=1, <b>M1X</b>, P, <b>(1465, 0, 1500, 0, 0, -180)</b>, R=5. 0, H=1, MS</p> <ul style="list-style-type: none"> <li>转换为执行形式后数据仍将保持该形式。也可以使用“屏幕编辑功能”进行编辑。</li> <li>反编译时，与输出指定无关，返回原来的形式（MOVEX-X）。</li> </ul> <p><b>MOVEX-J 形式姿势常数</b>            请以(J1, J2, J3, J4, J5, J6)的形式描述示教点。            MOVEX A=1, <b>M1J</b>, P, <b>(0, 90, 0, 0, 0, 0)</b>, R=5. 0, H=1, MS</p> <ul style="list-style-type: none"> <li>转换为执行形式后数据仍将保持该形式。也可以使用“屏幕编辑功能”进行编辑。</li> <li>反编译时，与输出指定无关，返回原来的形式（MOVEX-J）。</li> </ul> <p><b>MOVEX-E 形式姿势常数</b>            请以编码器值描述示教点。            MOVEX A=1, <b>M1E</b>, P, <b>(&amp;H80000, &amp;H80000, &amp;H80000, &amp;H80000, &amp;H80000, &amp;H80000)</b>, R=5. 0, H=1, MS</p> <ul style="list-style-type: none"> <li>进行编译后，将转换为<b>与实际设备上按下“记录”键</b>  所记录的移动命令相同的数据形式。对于该形式适用反编译时，将根据输出形式指定生成语言形式文件。</li> </ul>
				<p>也可以使用姿势变量设定示教点。 请参照“2.5.9 姿势变量”</p> <p><b>姿势变量</b>            USE 1            P1 = (1200, 0, 1800, 0, 0, -180)            MOVEX A=1, <b>M1X</b>, P, <b>P1</b>, R=5. 0, H=1, MS</p> <ul style="list-style-type: none"> <li>使用姿势变量时，必须事先选择姿势文件。详情请参照“FN98 USE”的在线帮助。</li> <li>关于姿势变量的编辑（记录及位置修正）请参照“3.3 编辑姿势文件”。</li> <li>也可以使用通过“姿势运算”计算出的姿势变量（参照“2.7.5 姿势运算”）</li> <li>转换为执行形式后数据仍将保持姿势变量形式。也可以通过“屏幕编辑功能”变更姿势变量号码等。</li> <li>反编译时，与输出指定无关，返回原来的形式（姿势变量形式）。</li> <li>也可以使用以下的变量间接描述姿势变量号码。 P[V1%]</li> </ul>

No.	符号	名称	值	说明
				<p><b>移位动作</b></p> <p>仅在使用姿势变量时，才能像下面一样以移位变量或移位常数组合使用。</p> <p>USE 1</p> <p>P1 = (1200, 0, 1800, 0, 0, -180) ' (X, Y, Z, roll, pitch, yaw)</p> <p>R1 = (10, 0, 0, 0, 0, 0) ' X+10[mm] shift</p> <p>MOVEX A=1, MIX, P, <b>P1+R1</b>, R=5. 0, H=1, MS</p> <p>MOVEX A=1, MIX, P, <b>P1+(10, 0, 0, 0, 0, 0)</b>, R=5. 0, H=1, MS</p> <p>MOVEX A=1, MIX, P, <b>P**R1</b>, R=5. 0, H=1, MS</p> <p>MOVEX A=1, MIX, P, <b>P**+(10, 0, 0, 0, 0, 0)</b>, R=5. 0, H=1, MS</p> <ul style="list-style-type: none"> <li>• 姿势变量“P*”表示当前位置。</li> <li>• 移位动作以及包含“P*”的移动命令在编译后无法通过“屏幕编辑功能”进行编辑。要编辑时请编辑机器人语言的源代码，然后重新编译为执行形式。</li> <li>• 移位常数以及移位变量只能以(X, Y, Z, rol, pitch, yaw)的形式使用。</li> </ul>
8	R T R D	移动速度	-	<p>是机器人的移动速度。“R=10.0”指“10.0%”。可以使用以下4种。</p> <p>S 线速度 (1.0[mm/s]~5000[mm/s])</p> <p>T 移动时间 (0.01[sec]~100[s])</p> <p>R 能力指定 (1.0[%]~100[%])</p> <p>D 手腕姿势变更速度 (1~500[deg/s])</p>
9	H	工具号码	1~32	是移动命令中所使用的工具号码。
10	MS	速度基准指定	-	<p>请在机构描述的末尾描述速度基准。</p> <p>可以对多个机构进行描述。</p>
11	CONF	配置	0000~1112	<p>对于指定的(X, Y, Z, roll, pitch, yaw)，有时可能存在多个可实现的机器人姿势。CONF 是用于从这些姿势中选择一个辅助参数。请按以下形式进行描述。</p> <p>CONF = ijkl</p> <p>i 0:FLIP / 1:NONFLIP 翻转/不翻转</p> <p>j 0:ABOVE / 1:BELOW 肘部上方/肘部下方</p> <p>k 0:LEFTY / 1:RIGHTY 左腕/右腕</p> <p>l 0:小于±180 指定凸缘轴旋转方向</p> <p>1:0~360 度</p> <p>2:0~-360 度</p> <ul style="list-style-type: none"> <li>• 本参数仅在 MOVEX-X 形式的姿势常数时有效。姿势变量以编码器值形式处理内部数据，因此无视 CONF。</li> <li>• 未描述 CONF 时，机器人在再生动作时将自动选择与之前的姿势相近的姿势。提前执行了 FLIP 等的姿势强制选择命令时，将根据该命令进行动作。</li> <li>• 请参照以下应用命令的在线帮助。</li> </ul> <p>FN160 POSAUTO</p> <p>FN161 LEFTY</p> <p>FN162 RIGHTY</p> <p>FN163 ABOVE</p> <p>FN164 BELOW</p> <p>FN165 FLIP</p> <p>FN166 NONFLIP</p> <p>FN202 FRANGE</p>





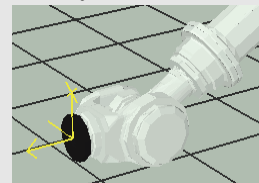
如果 CONF 的参数有误，机器人将做出意想不到的动作，有可能造成机械手装置及配线破损，引发事故。在实际设备上确认动作时，请充分引起注意。一般情况下推荐省略 CONF 的参数，由机器人自动选择姿势。此时也需要事先确认动作。(如果持有简易模拟软件[FD on Desk]，就无需直接运转机器人，而模拟检查机器人动作。)



#### < FLIP / NONFLIP >

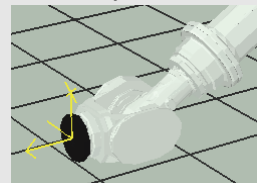
对于同样的(X, Y, Z, roll, pitch, yaw)，J5 轴的角度有 2 个样式时，选择其中的哪一个。

0:FLIP(J5<0)



J1	0.0	X=	2229.6
J2	60.0	Y=	0.0
J3	0.0	Z=	1263.8
J4	-180.0	r=	0.0
J5	-30.0	p=	-90.0
J6	180.0	y=	180.0

1:NONFLIP(J5>0)

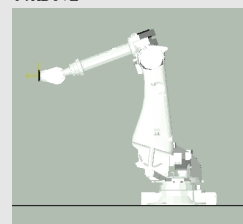


J1	0.0	X=	2229.6
J2	60.0	Y=	0.0
J3	0.0	Z=	1263.8
J4	0.0	r=	0.0
J5	30.0	p=	-90.0
J6	-0.0	y=	180.0

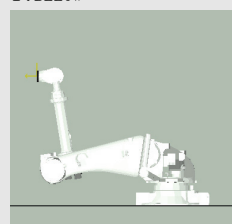
#### < ABOVE / BELOW >

对于同样的(X, Y, Z, roll, pitch, yaw)，“手肘”的位置有 2 个样式时，选择其中的哪一个。ABOVE 为上侧，BELOW 为下侧。但是对于无法做出“后翻动作”的机器人，无法做出 BELOW 的姿势。即使指定为 BELOW 中途也会发生动作范围错误。(本图为便于说明，制作时未考虑动作范围)

0:ABOVE



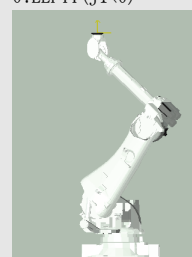
1:BELOW



#### < LEFTY / RIGHTY >

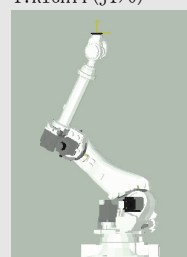
对于同样的(X, Y, Z, roll, pitch, yaw)，J1 轴的角度有 2 个样式时，选择其中的哪一个。

0:LEFTY(J1<0)



J1	-30.0	X=	259.9
J2	130.0	Y=	-150.0
J3	-0.0	Z=	2675.4
J4	0.0	r=	150.0
J5	50.0	p=	-0.0
J6	-0.0	y=	0.0

1:RIGHTY(J1>0)



J1	150.0	X=	259.9
J2	144.5	Y=	-150.0
J3	7.7	Z=	2675.4
J4	0.0	r=	150.0
J5	27.8	p=	0.0
J6	180.0	y=	-0.0



想要解除通过 FLIP 等强制选择姿势的状态时，请执行 POSAUTO。之后将像平常一样自动选择与前一个步骤相近的姿势。

**<异常点（死区）>**

对于一般的 6 轴垂直多关节构造机器人，实现某一目标点(X, Y, Z, roll, pitch, yaw)所可能的各轴角度(J1, J2, J3, J4, J5, J6)的组合的数量有时会是无限大。这样的点就被称作“异常点”。通过异常点会导致 TCP 速度极度变慢、无法进行正确的插补计算、手腕姿势发生巨大变化，因此请尽可能避免通过异常点。

例 1: J4 轴与 J6 轴平时 (J5 轴为 0 度)

例 2: J1 轴的旋转轴上方与 J5 轴重叠时

（补充）

本控制装置中，将异常点附近的一定范围称作“死区(死角)”。死区的范围在出厂时设定为 10 度。在该范围内无法保证插补计算的精度。

<常数设定>-[机械常数][姿势控制]“第 5 轴死角角度”

“移动速度”中可以以如下方式使用变量。

R = V1!

提示

可以使用以下变量。

Vn%      全局变量（整数）  
Vn!      全局变量（实数）  
Ln%      局部变量（整数）  
Ln!      局部变量（实数）

编译以 MOVEX-X 形式及 MOVEX-J 形式描述的机器人语言时，之后可以在如下画面中编辑各设定值。

### （机器人语言）

MOVEX A=1,M1X,P,(1690,0,2030,0,-90,-180),R= 5.0,H=1,MS, CONF=0000

### （编辑画面的显示）

提示

### （补充）

- 可以通过悬式示教作业操纵按钮台的按钮输入作为“FN645 MOVEX”进行记录。



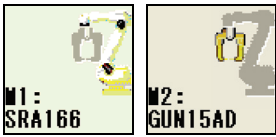
- 通过[动作可能] + [位置修正] 进行位置修正操作时，将显示如下警告信息。如选择“OK”，当前的编码器值将被选中的移动命令所覆盖，失去原先的数据形式。敬请注意。

“直角坐标” = 以 MOVEX-X 形式示教的步骤

“各轴角度” = 以 MOVEX-J 形式示教的步骤

“姿势变量” = 以姿势变量形式示教的步骤

5. 1. 2 多机构的情况



像附有伺服焊枪的机器人那样有多个机构时，MOVEX 可如下描述。

■姿势常数

MOVEX A=1, M1X, P, (1690, 0, 2030, 0, -90, -180), R= 10, H=1, MS, CONF=0000, M2J, P, (-20. 0), R=10, H=1  
MOVEX A=1, M1J, P, (0, 90, 0, 0, 0, 0), R= 10, H=1, MS, CONF=0000, M2J, P, (-20. 0), R=10, H=1

- 机构 1（机器人）以 MOVEX-X 形式或 MOVEX-J 形式设置坐标。
- 机构 2（伺服焊枪）以 MOVEX-J 形式设置焊枪轴的位置。（此处为 20mm 开放）

■姿势变量

USE 1  
P1 = (1690, 0, 2030, 0, -90, -180, -20)  
MOVEX A=1, M1X, P, P1, R= 10, H=1, MS, CONF=0000, M2J, P, P1, R=10, H=1

- 请在姿势变量中设定包含焊枪轴的值在内的 7 个参数。
- 机器人读取姿势变量的第 1~6 个的数据，伺服焊枪读取第 7 个数据。



对于走行轴也使用同样的描述方法。



MOVEX A=1, M1X, P, (1690, 0, 2030, 0, -90, -180), R= 10, H=1, MS, CONF=0000, M2J, P, (-100. 0), R=100, H=1




对于 2 轴定位器，该机构的轴值必须有 2 个。

MOVEX A=1, M1X, P, (1690, 0, 2030, 0, -90, -180), R= 10, H=1, MS, CONF=0000, M2J, P, (0, 0), R=100, H=1

## 5.2 输出信号

使用通用输出，可以经由外部 PLC 与其他装置实现联锁，进行夹具的开闭。请使用以下命令进行通用输出信号的 ON/OFF。

FN0	ALLCLR	输出信号全部清除
FN32	SET	输出信号 ON
FN34	RESET	输出信号 OFF
FN35	SETMD	附带脉冲和延迟输出信号
FN43	OUTDIS	输出信号分离输出
FN44	OUT	输出信号二进制输出
FN100	SETO	连续的输出信号 ON/OFF
FN105	SETM	输出信号
FN264	MULTIM	复数输出信号
FN280	DPRESETM	距离指定输出信号

详情请参照以下内容。  
 在线帮助或操作说明书“命令参考”



**<脉冲模式和延迟模式>**  
输出信号可以分配为脉冲模式和延迟模式。详情请参照操作说明书“外部输入输出篇”。如使用 SETMD，可以通过命令的参数任意设定延迟与脉冲的长度。



**<多个输出信号 05101~05196>**  
用于最多将 16 个输出信号整合在 1 个命令中进行操作。下例中 1 个命令可以对 8 个信号 ON/OFF。  
(本例中将 01~08 分配给 05101)

MULTIM 05101, 255

执行结果

0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
0011	0012	0013	0014	0015	0016	0017	0018	0019	0020

在以下画面中进行设定。  
<常数设定>[6 输入输出信号][3 输出信号分配][7 多个输出信号]



SETM 与可通过“输出信号”键输入的应用命令相同。



被分配了“状态信号”的信号由本控制装置的 CPU 自动控制，因此无法使用上述命令进行操作。


# 5.3 输入信号

使用来自外部的输入信号进行联锁控制时请使用以下命令。

FN525	WAITI	输入信号等待（正逻辑）
FN526	WAITJ	输入信号等待（负逻辑）
FN528	FETCH	读取输入条件
FN552	WAIT	附带定时输入信号等待
FN553	WAITA	附带定时组信号等待 (AND)
FN554	WAITO	附带定时组信号等待 (OR)
FN555	WAITE	附带定时组信号等待
FN557	WAITL	带定时器的等待输入信号条件 2
FN558	WAITAD	附带定时组信号等待 BCD (AND)
FN559	WAITOD	附带定时组信号等待 BCD (OR)
FN560	WAITED	附带定时输入信号 BCD 等待

如下例时，WAITI 执行时，如 I1 信号 OFF，机器人将停止（联锁状态），I1 信号 ON 时再次开始动作（联锁解除）。

```
MOVEX
MOVEX
WAITI I1
MOVEX
MOVEX
END
```

详情请参照以下内容。  
 在线帮助或操作说明书“命令参考”



### <多个输入信号>

可以通过 AND 条件或 OR 条件捆绑最多 16 个输入信号，作为 1 个输入信号进行使用。例如如果想要仅在 I1, I2, I3, I4 全部 ON 时联锁解除，请进行如下设定/描述。

5101

AND

OR

1			
1	2	3	4
0	0	0	0

WAITI I5101

在以下画面中进行设定。

<常数设定>[6 输出输入信号][2 输入信号分配][7 复数输入信号]



WAITI 与可通过“输入信号”键输入的应用命令相同。



另外还有根据输入信号的状态决定是否执行的命令。这种命令的名称中包含“附条件”、命令末尾带有“1”。

例：“FN81 CALLPI 附条件程序调用”




FN 557 WAITL 可以以标签指定 FN552 WAIT 的退避目标步骤。

# 5.4 步骤转移(跳跃)/步骤调用

进行步骤跳跃及步骤调用时使用以下命令。

FN20	JMP	步骤转移
FN21	CALL	步骤调用
FN22	RETURN	步骤返回
FN23	JMPI	附带条件步骤转移
FN24	CALLI	附带条件步骤调用
FN25	RETI	附带条件步骤返回
FN26	JMPN	附带次数条件步骤转移
FN27	CALLN	附带次数条件步骤调用
FN28	RETN	附带次数条件步骤返回
FN86	FCASEN	附带次数条件机身转移
FN87	FCASEI	附带条件机身转移
FN88	FCASEEND	机身转移终端
FN90	GOTO	行跳跃
FN91	GOSUB	行调用
FN601	*	标签
FN603	ON	ON GOTO 转移

详情请参照以下内容。

 在线帮助或操作说明书“命令参考”

可以通过标签名执行跳跃及调用。  
(例 1) GOSUB 中可以使用标签。返回时使用 RETURN。

```
USE 1
MOVEX A=1, MIX, P, P1, R=10, H=1, MS
GOSUB *INIT
MOVEX A=1, MIX, P, P1, R=10, H=1, MS
END
*INIT
ALLCLR
LETVI V1%, 0
LETVI V2%, 0
RETURN
END
```

```
1 USE[1]
2 10.0 % JOINT A1 T1
3 GOSUB[*INIT]
4 10.0 % JOINT A1 T1
5 END
6 *[*INIT]
7 ALLCLR
8 LETVI[V1%,0]
9 LETVI[V2%,0]
10 RETURN
11 END
```



(例 2) 可使用标签作为 IF 语句的跳往端。

```
*HOME
MOVEX A=1, MIX, P, P1, R=10, H=1, MS
IF I1=1 THEN *HOME ELSE *FIN
*FIN
MOVEX A=1, MIX, P, P2, R=10, H=1, MS
END
```

```
1 *[*HOME]
2 10.0 % JOINT A1 T1
3 IF I1=1 THEN *HOME ELSE *FIN
4 *[*FIN]
5 10.0 % JOINT A1 T1
6 END
```

(例 3) 根据 V1% 的值 (1~10) 选择跳往端。  
ON V1% GOTO \*LB1, \*LB2, \*LB3, \*LB4, \*LB5, \*LB6, \*LB7, \*LB8, \*LB9, \*LB10




跳跃及调用时使用标签代替行号码可以提高程序的可维护性。但是在 1 个程序中过多地使用跳跃及调用有可能会使作业的流程变得难以理解。

## 5.5 程序转移(跳跃)/程序调用

进行程序跳跃及程序调用时请使用以下命令。

FN80	CALLP	程序调用
FN81	CALLPI	附带条件程序调用
FN82	CALLPN	附带次数条件程序调用
FN83	JMPP	程序转移
FN84	JMPPI	附带条件程序转移
FN85	JMPPN	附带次数条件程序转移
FN102	CALLPR	相对程序调用
FN103	CALLPRI	相对程序调用 (I-条件)
FN104	CALLPRN	带次数条件相对程序调用
FN400	JMPPBCD	程序跳跃 (至外部 BCD 程序)
FN401	JMPPBIN	程序跳跃 (至外部 BIN 程序)
FN402	CALLPBCD	程序召回 (外部 BCD 程序)
FN403	CALLPBIN	程序召回 (外部 BIN 程序)
FN590	LCALLP	调用带参数的程序
N591	LCALLPI	调用带参数 (I) 的程序
FN592	LCALLPN	调用带参数 (频率) 的程序
FN680	JMPPV	程序跳跃 (可变量)
FN681	JMPPIV	程序跳跃 (I-状态) (可变量)
FN682	JMPPNV	程序跳跃 (频率) (可变量)
FN690	CALLPV	程序调用 (可变量)
FN691	CALLPIV	程序调用 (I-状态) (可变量)
FN692	CALLPNV	程序调用 (频率) (可变量)

详情请参照以下内容。

 在线帮助或操作说明书“命令参考”



执行程序调用时，如调用目标的程序中执行 END 命令，将返回调用源的程序的下一步骤。  
执行程序跳跃时，不返回原先的程序。



JMPPV, JMPPIV, JMPPNV, CALLPV, CALLPIV, CALLPNV 可以选择以下任意项作为参数。  
使用整数变量时，其内容将作为程序号码使用。除此以外的规范与 JMPP, JMPPI, JMPPN, CALLP, CALLPI, CALLPN 相同。  
1: 程序号码 (1~9999)  
2: 全局整数变量 Vn% (n=1~200, 301~500)  
3: 局部整数变量 Ln% (n=1~200, 301~500)



调用/跳往端的程序的开头附有“FN99 REM”的名字时，转换为执行形式时将显示如下名称。

调用源的程序

CALLP 1

1

5.0 %

JOINT A1 T1

2

CALLP[1](HOME POSITION)

3

200 mm/s LIN A1 T1

调用端的程序

REM “HOME POSITION”

示教

程序

1

[有]

步骤

0

2013/10/28 16:07

HOME POSITION

5-12





## 5.6 系统(单元)外启动/系统(单元)外调用

在多单元组成的系统中，要启动/调用其他单元的程序时请使用以下命令。

FN450	FORK	系统外启动
FN451	FORKI	系统外启动（输入）
FN452	FORKN	系统外启动（次数）
FN453	FORKWAIT	Fork 结束等待
FN454	CALLFAR	系统外调用
FN455	CALLFARI	系统外调用（输入）
FN456	CALLFARN	系统外调用（次数）

详情请参照以下内容。

 在线帮助或操作说明书“命令参考”

 操作说明书“多单元”

# 5.7 变量及运算

要使用变量进行运算时，可以使用以下命令。

FN75	LETVI	代入整数变数
FN76	LETVF	代入实数变数量
FN77	LETVS	代入字符串变数
FN142	GETP	代入实数变数（座标值）
FN143	GETPOSE	设定实数变量（姿势）
FN144	LETPOSE	设定姿势变量
FN145	GETSFT	代入实数变数（移动）
FN157	GETANGLE	代入实数变数（各轴角度）
FN158	GETFIGURE	设置实变量（位置获取）
FN628	LETLI	设定局部整数变量
FN629	LETLF	设定局部实数变量
FN634	LET	代入变量
FN635	ADDP	加姿势变量
FN637	ADDVI	加整数变量
FN638	ADDVF	加实数变量
FN639	SUBVI	减整数变量
FN640	SUBVF	减实数变量
FN641	MULVI	乘整数变量
FN642	MULVF	乘实数变量
FN643	DIVVI	除整数变量
FN644	DIVVF	除实数变量
FN648	ASIN	允许 ASIN 功能
FN649	ACOS	允许 ACOS 功能
FN650	TIMER	代入 TIMER 函数
FN651	SQR	代入 SQR 函数
FN652	SIN	代入 SIN 函数
FN653	COS	代入 COS 函数
FN654	TAN	代入 TAN 函数
FN655	ATN	代入 ATN 函数
FN656	ATN2	代入 ATN2 函数
FN657	ABS	代入 ABS 函数
FN658	MIN	代入 MIN 函数
FN659	MAX	代入 MAX 函数

详情请参照以下内容。  
👉 在线帮助或操作说明书“命令参考”  
👉 “2.7 表达式与运算”




可以制作自创的函数。  
👉 “2.9 用户程序”


## 5.8 移位(移动)

执行移位动作时，请使用以下命令。

FN29	RINT	机器人挤进(中断) (I 信号)
FN30	RINTA	机器人中断 (模拟信号)
FN51	SREQ	移位量要求
FN52	SHIFTR	移位 (移动)
FN53	LOCCVT	座标转换 (偏离量)
FN54	LOCCVT1	座标转换 (座标值)
FN58	SHIFTA	XYZ 移动
FN59	SEA	检索
FN68	LETR	代入移动量
FN69	ADDR	合计移动量
FN101	PRINT	字符串输出
FN111	RSCLR	RS232C 缓冲清除
FN113	CHGCOORD	移动座标系选择
FN127	WAITR	移动量收信号等待
FN145	GETSFT	代入实数变数 (移动)
FN224	REGC	移动寄存器拷贝
FN271	INPUT	字符串输入
FN275	LOCCVT3	基角移动
FN315	SREQ2	二进制移位数据要求
FN606	PRINT	字符串描绘
FN633	LETRE	代入移动元素
FN634	LET	代入变量
FN669	PRINTF	打印带格式的字符串
FN699	CLRREGWR	清除移动寄存器的写入状态
FN723	SIGREQ	移位量获取 (信号)

详情请参照以下内容。

 在线帮助或操作说明书“命令参考”

 操作说明书“通过外部输入移位”



移位动作执行中姿势强制选择的命令有效。

(参考)

“5.1 MOVEX (移动命令)”

“5.9 姿势强制选择”

# 5.9 姿势强制选择

是在以下情况中强制选择机器人所选择的姿势的命令。





(1) 执行以 MOVEX-X 形式的姿势常数描述的 MOVEX 命令时



(2) 执行移位功能时  
(此时与 MOVEX 示教点的数据形式无关，姿势强制选择可用)

FN160	POSAUTO	姿势控制无效
FN161	LEFTY	左臂系统
FN162	RIGHTY	右臂系统
FN163	ABOVE	肘上侧系统
FN164	BELOW	肘下侧系统
FN165	FLIP	腕触发系统
FN166	NONFLIP	腕非触发系统
FN202	FRANGE	凸缘轴基准角度

详情请参照以下内容。

-  在线帮助
-  操作说明书“命令参考”
-  操作说明书“通过外部输入移位”
-  “5.1 MOVEX（移动命令）”




- MOVEX 命令中存在 CONF 参数时 CONF 参数优先。
- 执行一次后之后的步骤全部适用。想要解除姿势强制选择时，请执行 POSAUTO。


## 5.10 坐标计算/姿势变量相关

是使用实数变量等计算机器人的姿势及坐标的命令。  
另外也有与姿势变量及姿势文件相关的命令。

FN71	LETX	姿势 X
FN72	LETY	姿势 Y
FN73	LETZ	姿势 Z
FN74	POSESAVE	存储姿势文件
FN94	GETPELR	代入实数变数（欧拉角座标值）
FN98	USE	姿势文件选择
FN142	GETP	代入实数变数（座标值）
FN143	GETPOSE	设定实数变量（姿势）
FN144	LETPOSE	设定姿势变量
FN157	GETANGLE	代入实数变数（各轴角度）
FN158	GETFIGURE	设置实变量(位置获取)
FN171	NRLCRD	机器人语言座标系选择
FN626	MODUSRCOORD	用户座标系修正
FN630	LETCOORDP	代入姿势变量
FN632	LETPE	代入姿势元素
FN634	LET	代入变量
FN635	ADDP	加姿势变量

详情请参照以下内容。

 在线帮助或操作说明书“命令参考”

 “2.6 任意变量”





在机器人语言程序内使用姿势变量时，请事先通过 FN98 USE 选择使用的姿势文件。未选择姿势文件时将发生错误。

# 5.11 用户任务

用户任务是可以在机器人作业程序的后台执行的用户宏程序。  
结合绘制命令可以制作自创的窗口（用户画面）。既可以在控制装置的电源投入时自动启动，也可以使用“FN671 CALLMCR”等从机器人作业程序中调用。用户任务相关命令如下所示。

PRINT	字符串描绘	
WINDOW	用户画面开启/关闭	
TITLE	用户画面标题设定	
CLS	用户画面清除	
LOCATE	显示位置指定	
GLINE	绘制直线	
GBOX	绘制长方形	
BARC	绘制圆形	
GPAINT	填充	
GSETP	绘制像素	
COLOR	指定颜色	
BGCOLOR	指定背景色	
EXIT	结束用户任务	
PAUSE	中断用户任务	
GARC	绘制椭圆	
GFONT	选择字体	
GSOFTKEY	绘制软键	
GMSGBOX	信息框的绘制	
PRINTF	打印带格式的字符串	
FN593	LDCALLMCR	调用带参数的 UT 程序
FN670	FORKMCR	用户任务程序起动
FN671	CALLMCR	用户任务程序调用
FN672	FORKMCRTM	时间指定用户任务程序起动
FN673	FORKMCRDST	距离指定用户任务程序起动

详情请参照以下内容。  
 在线帮助或操作说明书“命令参考”  
 操作说明书“用户任务”




- 用户任务专用命令只能在用户任务程序内执行。
- 只有在 PRINT，PRINTF 的输出目标在用户画面之外时，可以在机器人的作业程序内执行。


## 5.12 任意变量

“任意变量”是可以随意定义名称的变量。6 轴多关节机器人的坐标计算中具有非常便利的位置型等，通过组合各种命令及姿势变量可以进行复杂的示教点生成。

FN801	DIM	任何(任意)变量
FN809	POS2POSE	设置姿势变量(位置)
FN810	ANG2POSE	设置姿势变量(角度)
FN811	ENC2POSE	设置姿势变量(编码器)
FN812	POSE2POS	设置位置变量(姿势)
FN813	ANG2POS	设置位置变量(角度)
FN814	ENC2POS	设置位置变量(编码器)
FN815	POSE2ANG	设置角度变量(姿势)
FN816	POS2ANG	设置角度变量(位置)
FN817	ENC2ANG	设置角度变量(编码器)
FN818	POSE2ENC	设置编码器变量(姿势)
FN819	POS2ENC	设置编码器变量(位置)
FN820	ANG2ENC	设置编码器变量(角度)
FN821	CVTCOORDPOS	坐标变换(位置)
FN822	GETPOS	设置位置变量(位置数据)
FN823	GETANG	设置角度变量(位置数据)
FN824	GETENC	设置编码器变量(位置数据)
FN825	OPEPOSE	提取姿势变量
FN826	OPEPOS	位置型提取
FN827	OPEANG	提取角度变量
FN828	OPEENC	提取编码器变量

详情请参照以下内容。

 在线帮助或操作说明书“命令参考”


 “2.6 任意变量”


## 5.13 用户程序

“用户程序”是用户可以自由定义的函数。

FN802	UserProc	用户程序
FN803	ExitProc	退出用户程序
FN804	EndProc	结束用户程序
FN805	RetProc	返回用户程序
FN806	CallProc	调用用户程序

详情请参照以下内容。

 在线帮助或操作说明书“命令参考”

 “2.9 用户程序”




## 5.14 套接字通信

使用用户任务程序，可以与连接到以太网的外部装置进行 Socket 通信。

SOCKCREATE	创建套接字
SOCKCLOSE	套接字关闭
SOCKBIND	套接字绑定
SOCKWAIT	监听
SOCKCONNECT	服务器连接
SOCKSEND	发送数据
SOCKSENDSTR	发送字符串
SOCKRECV	数据接收
SETSTR	缓冲存储（字符串）
SETINT	缓冲存储（整数）
SETREAL	缓冲存储（实数）
SETBYTE	缓冲存储
GETSTR	缓冲读取（字符串）
GETINT	缓冲读取（整数）
GETREAL	缓冲读取（实数）
GETBYTE	缓冲读取

详情请参照以下内容。




 操作说明书“套接字通信”

## 5.15 模拟输入输出

是使用选购基板输入输出模拟信号的命令。

FN169	SPDDOWNA	模拟输入速度超驰
FN46	AOUT	模拟输出
FN319	AUTOZERO	模拟输入自动归零

详情请参照以下内容。

-  在线帮助或操作说明书“命令参考”
-  操作说明书“TCP 速度数据输出功能（包含模拟输出）”
-  操作说明书“通过外部输入的速度倍率（包含模拟输入）”



（参考）另外还有使用通用输入输出信号（数字信号）的类似命令。














FN277	SPDDOWND	数字输入速度倍率
FN278	DOUT	数字输出

## 5.16 其他

除此之外，还有如下的各种命令。各种应用程序及可选功能的相关命令，请参照各自的手册。

FN41	STOP	机器人停止
FN42	STOPI	附带条件机器人停止
FN50	DELAY	计时器
FN67	STOOL	固定工具号码选择
FN92	END	终端
FN99	REM	说明
FN230	COLSEL	设定干扰检测水平
FN252	PAUSEINPUT	输入暂停指令
FN438	SPN	伺服开
FN439	SPF	伺服关
FN467	USRERR	用户错误
FN600	NOP	NOP（不执行任何命令）
FN697	INCLUDE	转换表读取
FN698	INCLUDEIO	转换表读取（I/O 名称）

详情请参照以下内容。

-  在线帮助或操作说明书“命令参考”
-  操作说明书“应用篇 点焊” (FN119 SPOT 等)
-  操作说明书“应用篇 弧焊” (FN414 AS 等)
-  操作说明书“堆列功能” (FN249 PALLET3 等)
-  操作说明书““PALLET2” Palletize function(PALLET2 堆列功能)” (FN47 PALLET2 等)
-  操作说明书“Seam Welding(缝焊)” (FN245 SEAMST 等)
-  操作说明书“FLEXhand Operation(柔性手功能)” (FN362 FHCLAMP 等)
-  操作说明书“自适应动作功能” (FN364 ADAPTON 等)
-  操作说明书“MECHANISM-by-MECHANISM SERVO ON/OFF FUNCTION(各机构伺服 ON/OFF 功能)” (FN438 SPN 等)
-  操作说明书“CONVEYOR TRACKING(传送带同步功能)” (FN550 CNVI 等)
-  操作说明书“MECHANISM CHANGE(机构改变功能)” (FN95 CHGGUN/FN301 CHGMEC 等)
-  操作说明书“FORCE CONTROL(力控制)” (FN326 FORCECTRL 等)
-  其他 各种可选手册



提示

执行 STOP，STOPI 将停止程序的再生。要再次启动停止的程序，必须要有启动按钮的操作及外部启动信号。



提示

DELAY 与可通过“定时”键  输入的应用命令相同。



提示

执行固定工具插补时，以事先所定义的用户坐标系的位置与方向为基准执行插补动作。关于用户坐标系请参照<维修>[10 用户坐标系登记]的在线帮助。

NOTE

# 第6章 机器人语言的使用方法

本章将对机器人语言程序的用途及使用方法进行说明。

6.1 机器人语言使用上的注意事项 .....	6-1
6.1.1 定位精度 .....	6-1
6.1.2 使用机器人语言制作的程序的再生方法 .....	6-1
6.1.3 检查运转 .....	6-1
6.1.4 改写为编码器值的方法 .....	6-1
6.1.5 姿势变量的记录 .....	6-1
6.1.6 使用多机构时的注意事项 .....	6-2
6.2 使用机器人语言可实现的用途 .....	6-3
6.2.1 以机器人坐标值的数值指定目标位置 .....	6-3
6.2.2 以数值指定动作距离 .....	6-3
6.2.3 计算圆周轨迹 .....	6-4
6.2.4 以数值移位作业程序 .....	6-5
6.2.5 使用变量令 1 个作业程序对应多个工件 .....	6-5
6.3 关于变量（内部变量、姿势变量） .....	6-6
6.3.1 关于内部变量 .....	6-6
6.3.2 关于姿势变量 .....	6-7
6.3.3 姿势变量的编辑方法 .....	6-8
6.4 机器人语言的使用示例 .....	6-9
6.4.1 读出机器人当前位置的方法 1 .....	6-9
6.4.2 读出机器人当前位置的方法 2 .....	6-9
6.4.3 使用用户坐标系的位置数据控制姿势变量 .....	6-10
6.4.4 使用数值控制机器人的位置 .....	6-11
6.4.5 使用数值指示外部轴的移动位置 .....	6-11
6.4.6 通过输入信号从外部接收数值 .....	6-12
6.4.7 使用从输入信号接收的数值对示教位置进行移位 .....	6-13
6.4.8 (电弧焊)通过外部信号切换焊接条件、摆动条件的方法 .....	6-14
6.4.9 (电弧焊)使用激光搜索检测长度不同的工件端 .....	6-16
6.4.10 (电弧焊)感应圆柱管内表面边缘的 3 个点求出管的中心位置 .....	6-17
6.4.11 (电弧焊)生成沿圆柱管内表面的专用坐标系 .....	6-19
6.4.12 可以向姿势变量代入任意坐标系的指令值的应用命令 .....	6-20
6.4.13 读出机器人当前位置并向外部输出 .....	6-21



## 6.1 机器人语言使用上的注意事项

### 6.1.1 定位精度

使用机器人语言示教移动命令的方法中，以下的精度非常重要。  
这些精度将会影响通过机器人语言所指定的位置与实际的 TCP 之间的误差。

- (1) 工具（喷灯安装）精度
- (2) 母带精度
- (3) 机器人定位精度
- (4) 工件加工精度
- (5) 工件设置精度

### 6.1.2 使用机器人语言制作的程序的再生方法

检查运转或自动运转使用机器人语言制作的程序时，原则上请从步骤 1 开始动作。  
但在再生中暂时停止后再次启动时，可以从停止的步骤开始动作。



使用机器人语言描述的移动命令的示教点，基本上都不是通过示教所记忆的位置，而是从前一个步骤的姿势变量中读取的位置，或通过计算式求出的位置。  
从中途的步骤开始执行无法获取正确的位置信息，机器人将无法按照预期动作。

### 6.1.3 检查运转

使用机器人语言制作的程序在再生前必须通过检查运转确认机器人的动作。



即使从机器人语言形式向执行形式的转换正常结束，若计算方法及获取位置的顺序有误时，机器人有可能无法按照预期动作。

### 6.1.4 改写为编码器值的方法

使用机器人语言示教的程序的移动命令，通过坐标值、关节角度或变量记录位置。  
使用悬式示教作业操纵按钮台的<位置修正>对使用机器人语言示教的程序的移动命令进行位置修正时，进行位置修正步骤的记录数据将改写为当前的各轴编码器值。

### 6.1.5 姿势变量的记录

姿势变量储存在机器人控制装置的内部存储器上。  
姿势变量在重新投入机器人控制装置的电源时数据会丢失，因此必须重新计算。

想要保持当前的姿势变量时，请使用 POSESAVE 命令记录到姿势文件中。  
姿势文件中的姿势变量可以使用 USE 命令从内部存储器中读出。

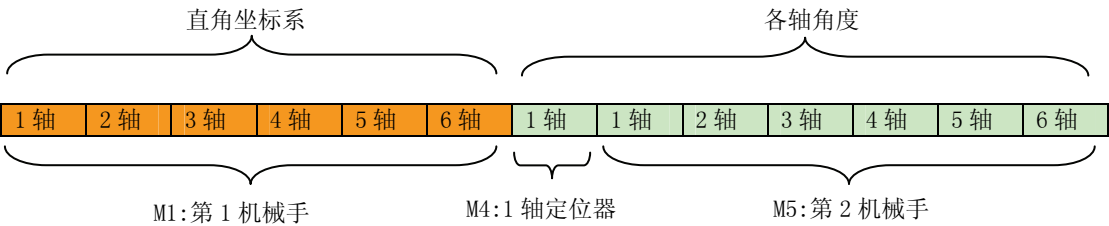
6.1.6 使用多机构时的注意事项

对于单元内存在多个机构的系统，关于处理姿势变量的应用命令有以下的注意事项和限制事项。

- (1) 请为多机构设定相应姿势变量的记录数据。

将<常数设定>-[5 操作和示教条件]-[1 操作条件]-“12 机器人语言（GETP，GETPOSE）”  
设定为“有效”。

- (2) 姿势变量的记录数据按照单元内的机构号码顺序以左对齐的方式记录。  
例如，M1：第 1 机械手、M4：1 轴定位器、M5：第 2 机械手这 3 个机构所属单元所处理的记录数据如下图所示。



- (3) 使用取得或代入姿势变量的记录数据的应用命令时，请仅对第 1 机械手的记录数据以直角坐标系处理，其余的机构全部以各轴角度处理。请参照（2）项的图。

取得或代入姿势变量的记录数据的应用命令有以下命令。

- GETP，GETPOSE，LETPOSE → 第 1 机械手以外请以各轴角度处理。
- ADDP，LETPE → 不支持第 2 机械手以后的机构。
- LETCOORDP → 支持所有机构。



## 6.2 使用机器人语言可实现的用途

机器人语言对可以通过机器人所具有的坐标系上的数值指定示教点的工件有效。

如同 NC（数值控制）一样，焊接线的位置、轨迹可以通过数值及算式计算指定。

即使是尺寸不同的工件、焊接位置及焊接长度有所变化，使用机器人语言只需改变指定的数值即可使用同一作业程序，因此无需对各工件分别示教。

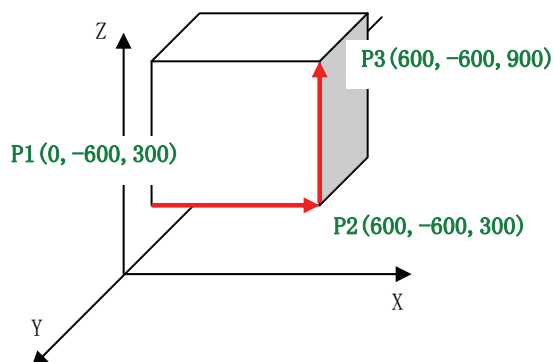
### 6.2.1 以机器人坐标值的数值指定目标位置

如下图般在 P1→P2→P3 这 3 点连接而成的直线上动作时，

- 1) 在示教方式下，分别手动运转机器人至这 3 点并记忆位置。

#### 【示教方式】

```
Step1 MOVE Joint P1 示教位置
Step2 MOVE Line P2 示教位置
Step3 MOVE Line P3 示教位置
```



- 2) 在机器人语言方式下，在作业程序中描述 3 个点的坐标值（包括姿势）。  
无需手动运转机器人。

#### 【机器人语言方式】离线示教方式

```
Step1 MOVEX P (0, -600, 300, 0, -90, 0)
Step2 MOVEX L (600, -600, 300, 0, -90, 0)
Step3 MOVEX L (600, -600, 900, 0, -90, 0)
```

※描述内容有所简化。

### 6.2.2 以数值指定动作距离

P1 开始位置 (700.0, -300.0, 600.0)

→ 从 P1 沿 X 轴 (+) 方向直线动作 100mm 到达 P2

→ 从 P2 沿 Z 轴 (-) 方向直线动作 100mm 到达 P3

该操作可以以机器人语言进行如下描述。

#### 【机器人语言方式】通过计算目标位置（使用姿势变量 Pn）描述动作命令。

- 1) 目标位置的计算使用姿势变量 Pn 来决定。

```
Step1 P1 = (700.0, -300.0, 600.0, -150, 0, 150)
```

```
Step2 P2 = P1 + (100, 0, 0, 0, 0, 0) ← 加上动作距离（沿 X 方向+100mm）
```

```
Step3 P3 = P2 + (0, 0, -100, 0, 0, 0) ← 加上动作距离（沿 Z 方向-100mm）
```

- 2) 姿势变量的位置按 P1→P2→P3 的顺序作直线动作。

```
Step4 MOVEX A=1, AC=0, SM=0, M1X, P, P1, R= 100, H=1, MS, CONF=0020
```

```
Step5 MOVEX A=1, AC=0, SM=0, M1X, L, P2, S= 16.6, H=1, MS, CONF=0020
```

```
Step6 MOVEX A=1, AC=0, SM=0, M1X, L, P3, S= 16.6, H=1, MS, CONF=0020
```

```
Step7 END
```

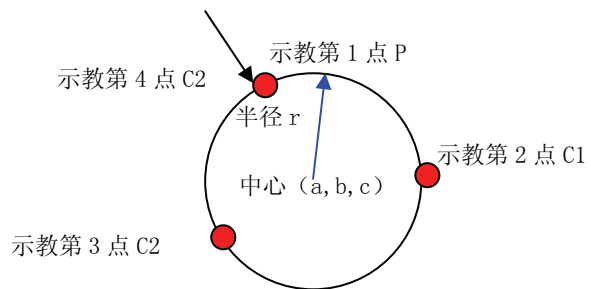
### 6.2.3 计算圆周轨迹

示教已知中心位置和半径值的圆周轨迹时，

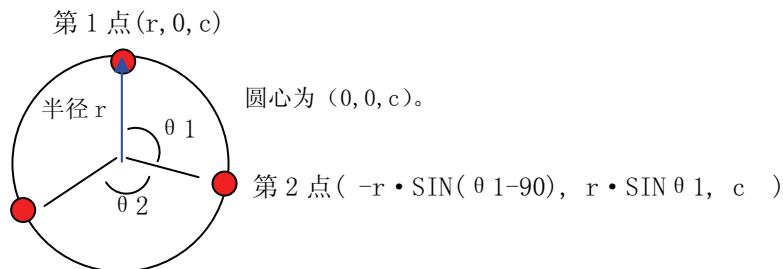
- 1) 在示教方式下，分别手动运转机器人至圆周上具有代表性的3点并记忆位置。

**【示教方式】**

Step1 MOVE Joint P1 示教位置  
Step2 MOVE CIR1 P2 示教位置  
Step3 MOVE CIR2 P3 示教位置  
Step4 MOVE CIR2 P1 示教位置



- 2) 在机器人语言方式下，可以根据圆的中心位置和半径值计算圆周上的位置。  
这里以 XY 平面上的圆周进行说明 (Z 位置固定)。



**【机器人语言方式】**

- ① 用实数变量 (Vn! 或 Ln!) 设定计算式中所需要的各数值 ( $r, c, \theta_1, \theta_2$ )。
 

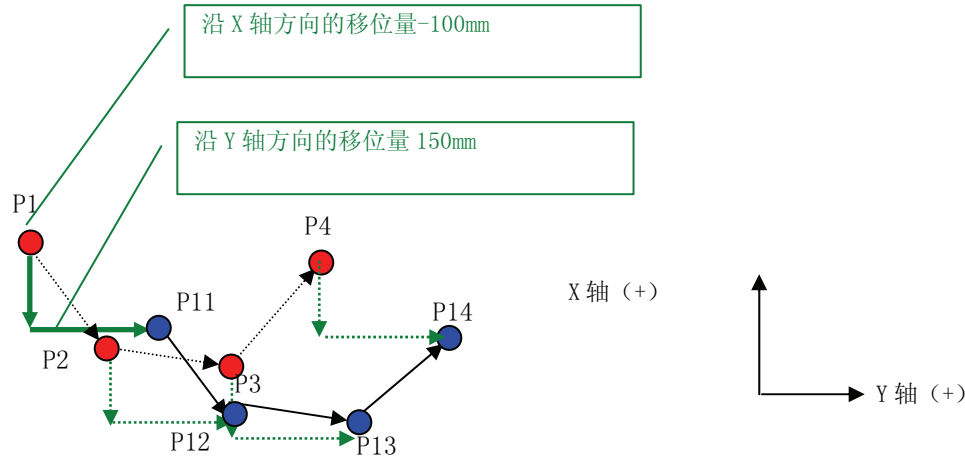
```
Step1 V1! = r
Step2 V2! = c
Step3 V3! =  $\theta_1$ 
Step4 V4! =  $\theta_2$ 
```
- ② 先将圆周上 3 点的计算位置代入姿势变量 P1 (第 1 点)、P2 (第 2 点)、P3 (第 3 点)。
 

```
Step5 P1=( V1!, 0, V2!, 0, -90, 0)
Step6 L1! = V3! - 90
Step7 L2! = V3! + V4!
Step8 P2=(-1 * V1! * SIN(L1!), V1! * SIN(V3!), V2!, 0, -90, 0)
Step9 P3=(V1! * COS(L2!), V1! * SIN(L2!), V2!, 0, -90, 0)
```
- ③ 按姿势变量的位置 P1→P2→P3→P1 的顺序进行圆弧操作。
 

```
Step10 MOVEX P P1
Step11 MOVEX C1 P2
Step12 MOVEX C2 P3
Step13 MOVEX C2 P1
```

6.2.4 以数值移位作业程序

将作业程序(P1, P2, P3, P4)沿 X 轴方向移位-100mm、沿 Y 轴方向移位+150mm。



这里示教位置使用姿势变量 (P1~P4、P11~P14)。  
移位变量 R 以坐标表示(X, Y, Z, r, p, y) 设置移位量，在原始示教位置上加上移位变量 R。

```
【机器人语言描述】
Step1 R1 = ( -100.0, 150.0, 0, 0, 0, 0 ) → 在移位变量中设置各坐标轴的移位量
Step2 P11 = P1 + R1 → 原始位置 P1 加上移位量
Step3 P12 = P2 + R1 → 原始位置 P2 加上移位量
Step4 P13 = P3 + R1 → 原始位置 P3 加上移位量
Step5 P14 = P4 + R1 → 原始位置 P4 加上移位量
Step6 MOVEX P P11
Step7 MOVEX P P12
Step8 MOVEX P P13
Step9 MOVEX P P14
```

6.2.5 使用变量令 1 个作业程序对应多个工件

通常的示教方式时，作业程序中所记录的机器人位置数据（以后简称为记录数据）是使用实际的工件所示教的位置（编码器值）。

工件的形状及尺寸有多种时，每次更换工件都要进行示教，非常麻烦。  
对于这种情况，如果可以根据形状及尺寸的变化通过数值计算（移位计算、算术计算）求出示教位置，便可以仅用 1 个作业程序进行对应。  
使用机器人语言对示教位置进行数值计算。形状及尺寸信息作为参数设置为变量。  
在计算结果中代入变量，使用变量数据设定移动命令的位置数据。

变量数据为机器人所具有的坐标系中的 TCP(X, Y, Z, r, p, y)。

所谓机器人所具有的坐标系，是指以下 5 种坐标系。  
机械坐标系、工件坐标系、用户坐标系、绝对坐标系、工具坐标系

可以使用上述任一坐标系指定 TCP 位置 X, Y, Z (mm) 和工具角度 r, p, y (度)。

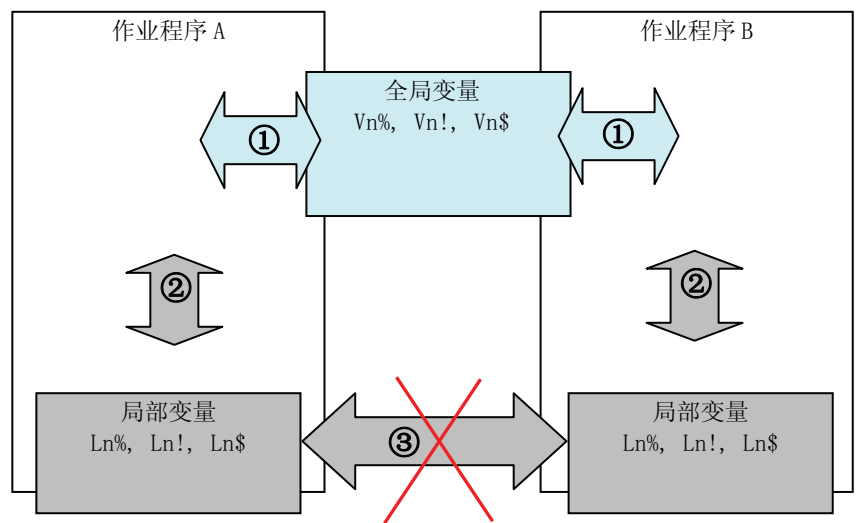
- 通过使用变量，
- 可以根据工件的设置位置及形状尺寸，即使没有实际的工件也能制作作业程序。
  - 1 个作业程序能够对应多个工件，提高了适应性。
  - 结合机器人语言的算术计算，可以进行高级的轨迹计算。

## 6.3 关于变量（内部变量、姿势变量）

对机器人语言中使用的变量进行说明。

### 6.3.1 关于内部变量

- (1) 全局变量与局部变量  
内部变量是机器人语言中储存所使用数值的区域。  
整数变量%、实数变量!、字符串变量\$具有各自的全局变量（V）和局部变量（L）。

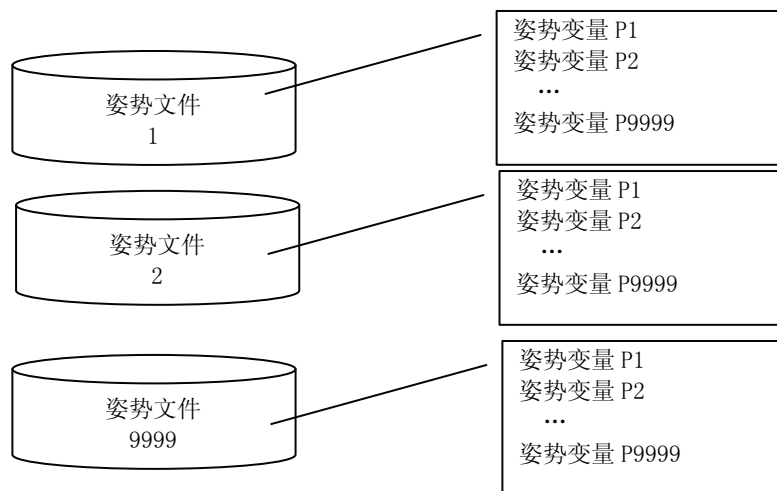


全局变量是系统内的通用区域。是无论单元内外，在所有作业程序间的公共变量。  
作业程序 A 所描述的整数变量 V1%和作业程序 B 所描述的 V1%为同一个变量（区域）。  
局部变量是作业程序内的通用区域。  
作业程序 A 所描述的整数变量 L1%和作业程序 B 所描述的 L1%为不同的变量（区域）。  
作业程序 A 所描述的局部变量，无法从作业程序 B 中访问。反之亦然，作业程序 B 的局部变量，无法从作业程序 A 中访问。

- (2) 移位变量  
移位变量是局部变量。  
作业程序 A 所描述的变量 R1 和作业程序 B 所描述的 R1 为不同的变量（区域）。

### 6.3.2 关于姿势变量

- (1) 姿势变量最多可以使用 P1~P9999。
- (2) 姿势变量在存储器上采用与作业程序的步骤数据相同的构造进行管理。
- (3) 可编辑的姿势变量值，通过 POSESAVE 命令保存在姿势文件中（由于不具有停电保持功能，如未保存在文件中，在切断主电源后将被清除）。
- (4) 姿势文件最多可以制作与作业程序相同的文件数（9999），但每次只能调用 1 个姿势文件。
- (5) 姿势文件的文件名  
“单元名称-P.姿势文件号码”  
例如 UNIT1 的姿势文件号码=1000 时，文件名为“UNIT01-P.1000”。
- (6) 姿势文件的构造  
姿势变量与姿势文件的关联如下图所示。

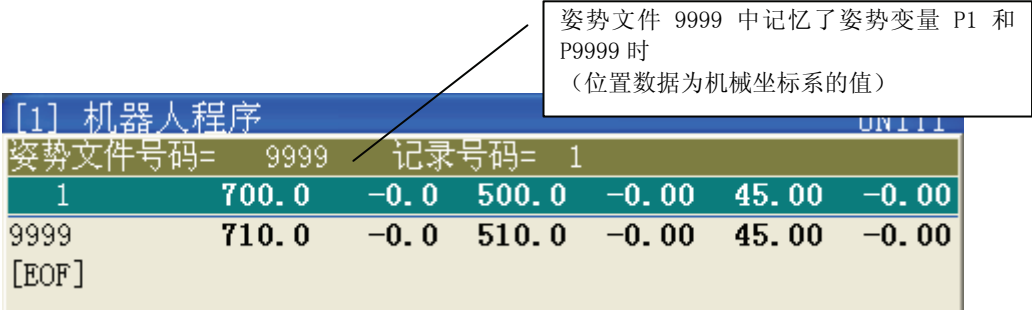


6.3.3 姿势变量的编辑方法

- 姿势变量可以通过以下 2 种方法进行设定。
- 在悬式示教作业操纵按钮台上编辑
  - 使用机器人语言读写

在悬式示教作业操纵按钮台上编辑

- (1) 将<维修>-[示教、再生条件]的“13 姿势记录”设定为“有效”，程序监视器画面切换为姿势文件画面。
- (2) 程序号码=姿势文件号码、步骤号码=姿势变量号码。
- (3) 与作业程序的读取操作相同，输入姿势文件号码读取姿势文件。
- (4) 显示对象文件所记忆的姿势变量（位置数据）。
- (5) 与作业程序相同，可以通过[编辑]变更姿势变量值。



使用机器人语言读写

- (1) 使用“USE”命令指定姿势文件号码（USE 命令还具有切换姿势文件的功能）。  
  
“USE”命令之后的步骤中描述的姿势变量（P1～P9999），作为“USE”所指定的姿势文件的步骤数据（Step1～Step9999）进行管理。  
  
【例】在姿势文件 2（存储器上）的步骤 3 中记忆数值。  
USE 2  
P3=(100, 10, -200, 0, 90, 0)
- (2) 使用“POSESAVE”，将当前所记忆的姿势变量，保存到“USE”所指定的姿势文件中。  
  
【例】将姿势文件 2 的步骤 3 的数值保存到姿势文件 2 中。  
USE 2  
P3=(100, 10, -200, 0, 90, 0)  
POSESAVE
- (3) 使用“GETPOSE”命令，可以从“USE”所指定的姿势文件中读出姿势变量。  
  
【例】从内部实数变量 V11!～V16!中读出姿势文件 2 的步骤 3 中记忆的姿势变量。  
USE 2  
GETPOSE V11!, 3
- (4) 使用“LETPOSE”命令，可以向“USE”所指定的姿势文件写入姿势变量。  
  
【例】向姿势文件 2 的步骤 4 写入内部实数变量 V11!～V16!的姿势变量。  
USE 2  
LETPOSE 4, V11!

## 6.4 机器人语言的使用示例

### 6.4.1 读出机器人当前位置的方法 1

使用 GETP 命令<FN142>。  
也可以通过作业程序的功能一览进行示教。

**【例】**

从实数变量 V11!~V16!中读出当前位置（正交位置 X, Y, Z, a, b, c）。  
读出的位置数据为 GETP 命令执行时的当前位置。

```
GETP V11!
```

- ※1) 通常读出机械坐标系上的正交位置。
- ※2) 如果在 GETP 命令的前一个步骤中示教 NRLCRD 命令<FN171>，则将读出用户坐标系上的正交位置。  
(参照 7.4.3 项)。

### 6.4.2 读出机器人当前位置的方法 2

使用 SYSTEM!() 函数。  
该函数为宏命令。仅可在用户任务中使用。  
使用 SYSTEM!() 函数读出的位置数据，储存在实数变量 (Vn!) 中。

例如要随时监控机械坐标系的机器人当前位置时，

- (1) 制作用户任务程序。
- (2) 用户任务在制作成文本文件后，通过程序转换动作可能为执行形式。
- (3) 对于用户任务 P100，应制作“USERTASK-A.100”的文本文件。

以下为“USERTASK-A.100”的描述示例。  
以 50msec 为周期，从实数变量 V11!~V13!中读出机构 1 的机械坐标系 XYZ 位置。

```
REM “读出机器人当前位置（机械坐标系的 TCP）”
REM “V11! ← 机构 1 的当前 TCP（X 坐标）”
V11! = SYSTEM!(150)
REM “V12! ← 机构 1 的当前 TCP（Y 坐标）”
V12! = SYSTEM!(151)
REM “V13! ← 机构 1 的当前 TCP（Z 坐标）”
V13! = SYSTEM!(152)
PAUSE 50
GOTO 1
END
```

※ 关于 SYSTEM 函数的参数（括号（）内的数值），请参照“2.7.7 系统函数”。

- (4) 通过程序转换—语言转换功能，将语言形式转换为执行形式。  
+
- (5) 生成“USERTASK.100”。
- (6) 用户任务 P100 启动。

**【启动方法 1】** 通过<维修>-[12 用户任务]启动 P100

**【启动方法 2】** 在作业程序中使用 FORKMCR 命令启动 P100

6.4.3 使用用户坐标系的位置数据控制姿势变量

使用姿势变量时，有以下常用命令。

- LETX (FN71): 姿势 X
- LETY (FN72): 姿势 Y
- LETZ (FN73): 姿势 Z
- GETP (FN142): 代入实数变数 (座标值)
- GETPOSE (FN143): 设定实数变量 (姿势)
- LETPOSE (FN144): 设定姿势变量

上述命令中使用的姿势变量为机械坐标系的正交值。  
要使用用户坐标系控制姿势变量时，在 NRLCRD 命令中包括对象步骤。  
NRLCRD 是指定 (切换) 用户坐标系的命令。

这些应用命令也可以通过悬式示教作业操纵按钮台的应用命令一览进行示教。

【例】 要使用用户坐标系 2 上的正交值进行动作时

```
...
NRLCRD 2          ← 指定用户坐标系 2
LETX  [P1, 100]   ← 将用户坐标系 2 上的 X 坐标值 100 代入变量 P1 的 X 坐标值
LETY  [P2, 100]   ← 将用户坐标系 2 上的 Y 坐标值 100 代入变量 P2 的 Y 坐标值
LETZ  [P3, 100]   ← 将用户坐标系 2 上的 Z 坐标值 100 代入变量 P3 的 Z 坐标值
GETP  [V11!]      ← 从实数变量 V11!~V16! 中获取用户坐标系 2 上的当前位置 X, Y, Z, a, b, c
GETPOSE [V21!, 1] ← 以用户坐标系 2 上的正交值从 V21!~V26! 中读出变量 P1 的数据
LETPOSE [2, V31!] ← 将 V31!~V36! 的正交值作为用户坐标系 2 上的正交值写入变量 P2
NRLCRD 0          ← 解除用户坐标系的指定 (之后的姿势变量以机械坐标系处理)
...
```

重点

要使用应用命令修正用户坐标系时  
通过执行 MODUSRCOORD <Fn626>可以修正所指定的用户坐标系。  
  
关于应用命令的详情请参照 HELP。



### 6.4.4 使用数值控制机器人的位置

- (1) 使用数值指定机械坐标系的位置数据

```
MOVEX A=8, AC=0, SM=0, M1X, P, ( 710, 50, 300, 150, -45, -135), R= 3.0, H=1, MS
|           | — 使用数值指定 X, Y, Z, a, b, c
| ————— MnX: 指定机械坐标系
```

- (2) 使用实数变量指定机械坐标系的位置数据

```
MOVEX A=8, AC=0, SM=0, M1X, P, ( V11!, V12!, V13!, V14!, V15!, V16!), R= 3.0, H=1, MS
|           | — 使用实数变量指定 X, Y, Z, a, b, c
| ————— MnX: 指定机械坐标系
```

※ 通过改变实数变量 V11!~V16! 的值, 可以移动机器人的 TCP。

- (3) 使用姿势变量指定机械坐标系的位置数据

```
P11=(800, -20, 190, 150, -45, -135)
MOVEX A=8, AC=0, SM=0, M1X, P, P11, R= 3.0, H=1, MS
```

- (4) 将用户坐标系的位置数据代入姿势变量

直接值 ( ) 之后附加坐标系的识别符号 ‘U’。

作为在通过用户坐标系转换命令 CHGCOORD 指定的用户坐标系号码的坐标系上的直接值进行处理。

未执行 CHGCOORD 的状态下, 附加 ‘U’ 时无效。

下例中, 判断为用户坐标系 2 上的正交值。

※ 代入姿势变量 P11 时转换为机械坐标系的正交值。

```
CHGCOORD 2
```

```
P11=(800, -20, 190, 150, -45, -135)U
```

```
MOVEX A=8, AC=0, SM=0, M1X, P, P11, R= 3.0, H=1, MS
```

```
CHGCOORD 0 ← 用户坐标系运算结束时请务必通过 “CHGCOORD 0” 关闭状态。
```

- (5) 使用数值指定各轴角度的数据

```
MOVEX A=8, AC=0, SM=0, M1J, P, (0, 90, 0, 0, -90, 0), R= 3.0, H=1, MS
|           | — 使用数值指定 6 根轴的角度
| ————— MnJ: 指定各轴角度
```

- (6) 使用实数变量指定各轴角度的数据

```
MOVEX A=8, AC=0, SM=0, M1J, P, (V11!, V12!, V13!, V14!, V15!, V16!), R= 3.0, H=1, MS
|           | — 使用实数变量 Vn! 指定 6 根轴的角度
| ————— MnJ: 指定各轴角度
```

※ 通过改变实数变量 V11!~V16! 的值, 可以移动机器人的动作角度。

### 6.4.5 使用数值指示外部轴的移动位置

- (1) 使用数值对正交的 2 轴进行定位控制 (机构 No=3 分配为正交的 2 轴滑块)

```
MOVEX A=1, AC=0, SM=0, M3J, P, ( 100, -300), R= 100, H=1, MS
|           | — 使用数值指定 2 轴的移动位置 (mm)
| ————— MnJ: 指定各轴角度
```

- (2) 通过指定 V 变量对正交的 2 轴进行定位控制 (机构 No=3 分配为正交的 2 轴滑块)

```
MOVEX A=1, AC=0, SM=0, M3J, P, ( V11!, V12!), R= 100, H=1, MS
|           | — 使用实数变量指定 2 轴的移动位置 (mm)
| ————— MnJ: 指定各轴角度
```

※ 通过改变实数变量 V11!, V12! 的值, 可以改动滑块的移动位置。

6.4.6 通过输入信号从外部接收数值

GETSIGB(i1, i2)	以字节为单位从通过 i1 指定的输入输出端口获取数据。 (i1 为输入输出端口的群组号码) i2=0 时, 获取输入信号。 i2=1 时, 获取输出信号。  V1% = GETSIGB( 10, 0 ) 在上述描述中, 将从输入信号 IN49 到 IN56 的 8 位输入信号转换为二进制整数, 设定在整数变量 V1%中。	返回整数
-----------------	---	------

- 【例】 要从外部夹具向机器人控制装置传输小数点后 1 位的实数时,
- (1) 将由输入信号 I0009~I0016 的低 8 位、输入信号 I0017~I0024 的高 8 位所构成的 16 位作为二进制整数处理。
  - (2) (1) 的二进制整数是系统中处理的实数值的 10 倍的整数值。
  - (3) 机器人控制装置侧将接收到的 16 位的二进制整数乘以 1/10, 转化为实数。
  - (4) 符号位由另一输入信号 I0100 设定。I0100=1 时为负号。

上述规范中, 以机器人语言描述将外部输入信号转换为实数的处理(下记)。

V101! = GETSIGB(5, 0)+2^8\*GETSIGB(6, 0)

V111! = V101!/10

V121! = -1^I0100\*V111!

← I0009~I0016(低)、I0017~I0024(高)

← 从 V101!中获取二进制整数并除以 10

← 符号检查

最终结果, 实数变量 V121!中储存了从外部夹具接收的小数点后 1 位的实数。

输入函数 GETSIGB 的参数 i1  
将盘内 I/O 及输入输出号码 1~2048 每 8 点分为一组。  
输入信号、输出信号号码相同。

i1	输入输出信号	i1	输入输出信号	i1	输入输出信号	i1	输入输出信号
0	盘内 I/O	8	0033~0040	~	***~***	252	1985~1992
1	盘内 I/O	9	0041~0048	245	1929~1936	253	1993~2000
2	盘内 I/O	10	0049~0056	246	1937~1944	254	2001~2008
3	盘内 I/O	11	0057~0064	247	1945~1952	255	2009~2016
4	0001~0008	12	0065~0072	248	1953~1960	256	2017~2024
5	0009~0016	13	0073~0080	249	1961~1968	257	2025~2032
6	0017~0024	14	0081~0088	250	1969~1976	258	2033~2040
7	0025~0032	~	***~***	251	1977~1984	259	2041~2048

详细的使用方法请参照“第 2 章 机器人语言的语法”。

### 6.4.7 使用从输入信号接收的数值对示教位置进行移位

#### (1) 通过 SIGREQ 命令的移位控制

通过输入信号接收移位量。

详情请参照选购操作说明书《通过外部输入的移位功能》的“2.25 SIGREQ:通过移位量获取(信号)(FN723)”  
(移位量请求信号、移位设定完成信号等的同步信号,及用于提高设定数据可靠性的校验位等也需要参照)。

#### (2) 通过 SHIFTA 命令的移位控制

SHIFTA 可以在 XYZ 的各移位量中使用实数变量 V!。实数变量可以使用 1-194, 301-494 的变量,且没有限制。  
通过将使用 GETSIGB 取得的各移位量代入实数变量 V!, 可以实现移位控制。

#### (3) 通过 SHIFTR 命令的移位控制

SHIFTR 使用移位变量 R。移位变量 R 的数量为 R1~R9。

将使用 GETSIGB 取得的各移位量代入实数变量 V!后, 向移位变量 R 设定实数变量 V!的值。

$R1=(V101!, 0, 0, 0, 0, 0) \leftarrow$  将实数变量 V101!的值代入 X 坐标的移位量  
SHIFTR R1, ...

通过该控制, SHIFTR 区间的 X 坐标位置数据以 V101!中所储存的值进行移位。

#### (4) 通过 SF8+SF3 命令的移位控制

SF3 读出记录了修正量的移位长度文件 (DEVxxx), 按储存的移位长度 (XYZrpy) 对目标位置进行修正=移位。

通过从移位长度文件获取来自外部的移位量, 可以使用 SF3 事先移位控制。

来自外部的指定移位量使用 GETSIGB 从实数变量 V!中获取, 但由于实数变量的值储存在移位长度文件中, 需使用 SF8 命令。

使用 SF3、SF8 命令时必须要有传感器。对于不使用传感器的系统, 请将“触摸传感器”登录为虚拟。只有在登录后, 才不再需要传感器。

6.4.8 （电弧焊）通过外部信号切换焊接条件、摆动条件的方法

ASV、AEV、WFPV、WAXV 命令使可以通过变量 Vn%指定焊接条件或摆动开始条件的命令。

（步骤 1）使用整数变量 Vn%指定的数值是条件文件号码。

因此，应事先制作要使用的焊接开始条件文件、焊接结束条件文件、或摆动开始条件文件。

（步骤 2）制作焊接程序。

下例中，焊接开始文件号码、结束条件文件号码均由整数变量 V11%指定，摆动开始条件文件号码由整数变量 V21%指定。

```
0 [START]
1 100 % JOINT A1 T1
2 ASV[W1, V11%, 00, 00, 00, 00, 00, 00→]
3 WFPV[V21%, →] FN667;设置摆动模式（可
4 100 cm/m LIN A1 T1
5 WE FN443;横摆运条结束
6 AEV[W1, V11% →] FN666;焊接结束（可变量）
7 100 % JOINT A1 T1
8 END FN92;终端
```

（步骤 3）通过外部输入信号，向整数变量设定文件号码。

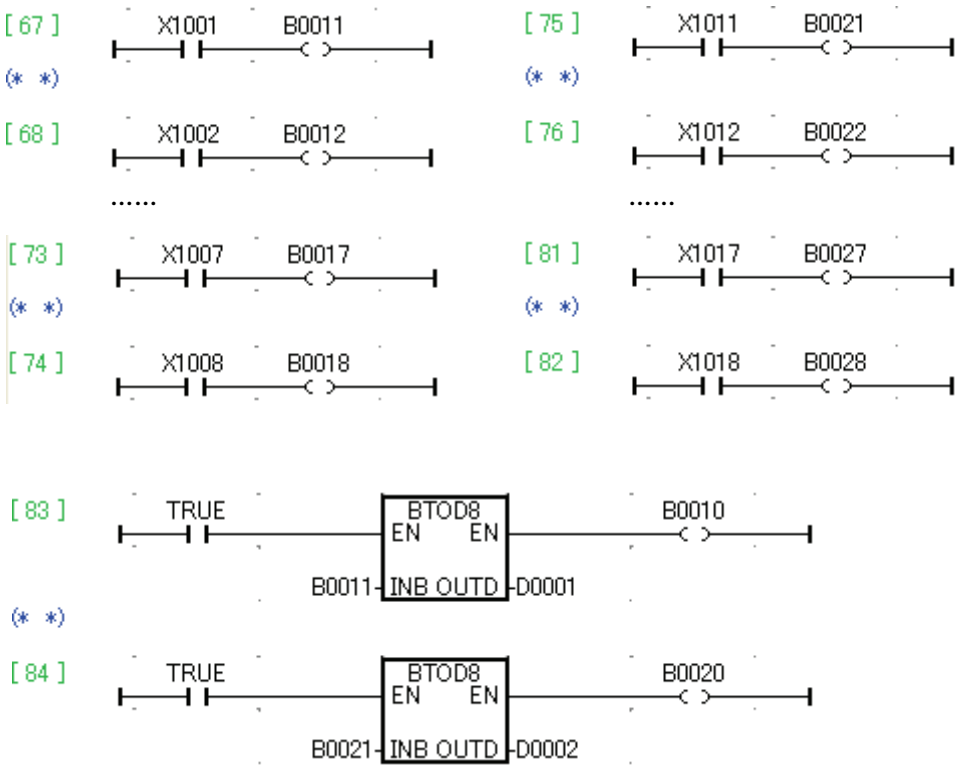
(1) 使用 GETSIGB 获取的示例

IN41—IN48 → V11%、IN49—IN56 → V21% 按此方式获取。

```
0 [START]
1 V11% = GETSIB(9, 0) FN625;代入式
2 V21% = GETSIB(10, 0) FN625;代入式
3 100 % JOINT A1 T1
4 ASV[W1, V11%, 00, 00, 00, 00, 00, 00→]
5 WFPV[V21%, →] FN667;设置摆动模式（可
6 100 cm/m LIN A1 T1
7 WE FN443;横摆运条结束
8 AEV[W1, V21% →] FN666;焊接结束（可变量）
9 100 % JOINT A1 T1
10 END FN92;终端
```

(2) 使用软件 PLC 获取的示例

- ① 使用整数变量 Vn%和软件 PLC 内部的整数变量 Dxxxx 的共享区域 (D0001 → V201%、D0002 → V202%)。
- ② 由于条件文件号码为 1~99，因此以 8 位进行交换。
- ③ 在软件 PLC 内部从整数变量中获取从外部读出的 8 位信号。  
(IN1002-IN1009 → 整数变量 D0001、IN1012-IN1019 → 整数变量 D0002 按此方式获取)



- ④ 在焊接程序的开头，从 PLC 中获取的值 V201%、V202%储存在 V11%、V21%中。

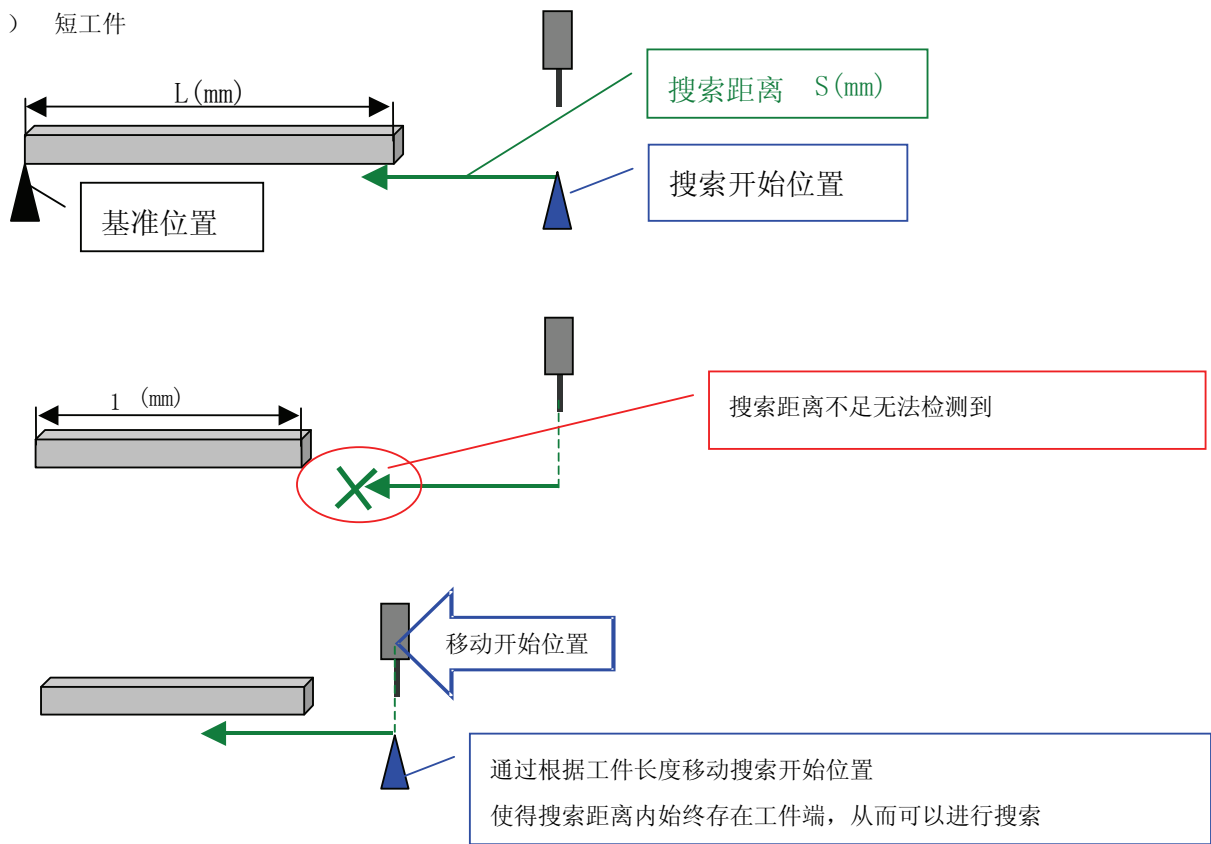
※ 指定条件文件号码无法使用整数变量 V201%~V299%。

0	[START]	
1	V11% = V201%	FN624;代入式
2	V21% = V202%	FN624;代入式
3	100 % JOINT A1 T1	
4	ASV[W1,V11%, 00, 00, 00, 00, 00→]	
5	WFPV[V21%,	→] FN667;设置摆动模式 (可
6	100 cm/m LIN A1 T1	
7	WE	FN443;横摆运条结束
8	AEV[W1,V21%	→] FN666;焊接结束(可变量)
9	100 % JOINT A1 T1	
10	END	FN92;终端

6.4.9 (电弧焊)使用激光搜索检测长度不同的工件端

(1) 长工件

(2) 短工件



使用机器人语言，根据工件长度自动计算开始位置。

<前提条件>

搜索距离  $S$  为搜索命令 SF1 的参数中所示教的数值。

工件长度为已知。

搜索距离  $S$ 、工件长度  $L$  (I) 可以通过外部输入信号从机器人侧的实数变量  $Vn!$  中获取。

从实数变量中获取外部输入信号的方法，请参照“6.4.6 通过输入信号从外部接收数值”。

【方法 1】

以工件的左端为基准位置，工件长度+搜索距离- $\alpha$  (mm) 的位置为搜索开始位置。

基准位置可以在系统中锁定。

搜索开始位置使用机器人语言计算得出。

【方法 2】

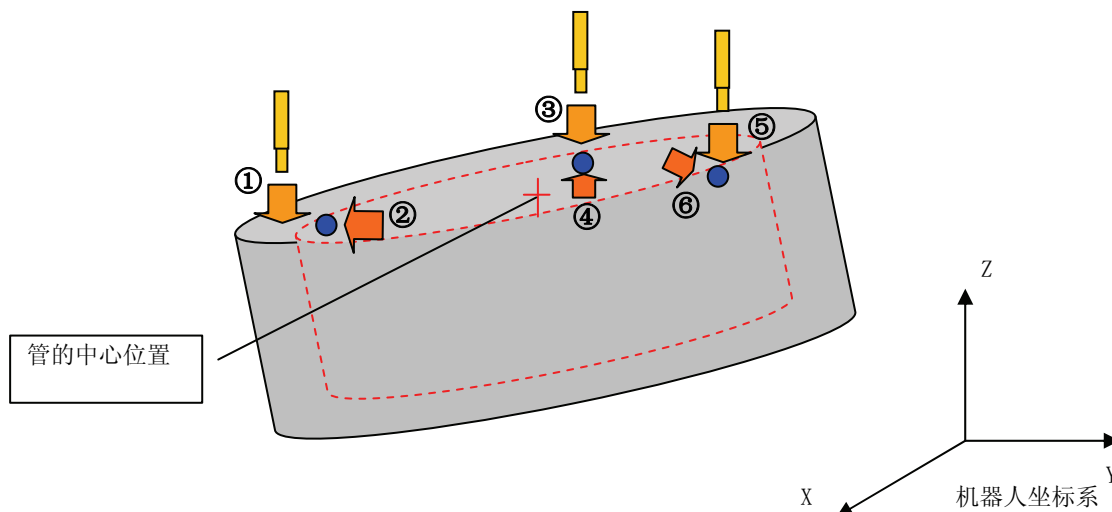
将长工件作为主工件，事先实际示教搜索开始位置，根据工件长度的变化（与长工件间的差距），对搜索开始位置进行移位。

工件长的差距通过机器人语言计算。

使用差距的值对搜索开始位置进行移位。

移位方法请参照“6.4.7 使用从输入信号接收的数值对示教位置进行移位”。

## 6.4.10 (电弧焊)感应圆柱管内表面边缘的3个点求出管的中心位置



使用触摸传感器，检测出圆柱管工件内表面边缘上的3点位置。

根据圆周上的3点，可以计算出圆的中心位置。

(1) 以主工件为基准，执行上图①~⑥的搜索动作&位置检测，制作主控程序。

- ①③⑤检测垂直方向的移位 (=边缘的高度位置)。
- 然后根据①③⑤所检测到的位置，使在②④⑥的水平方向上不摆动，自动调整检测时的高度。

该主控程序只需取得管内表面上的3点位置，  
即使不使用机器人语言制作，也没有问题。

(2) 之后对每个工件执行主控程序，计算出各个对象工件的圆心。

下页中记述了计算圆心位置的主控程序的范例。

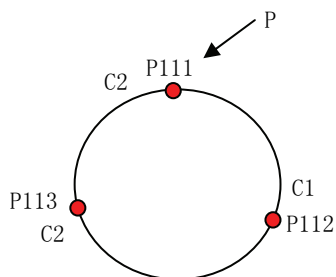
此外，将触摸检测到的内表面上的3点分别记录在姿势变量中后，可以使用这些姿势变量生成通过这3点的圆弧。

例如，将触摸检测到的内表面上的3点记录在姿势变量 P111、P112、P113 中，

便可以以变量描述 Joint→P111→圆弧 C1→P112→圆弧 C2→P113→圆弧 C2→P111 的圆周轨迹。

1 个程序便可以对应圆管的尺寸变化。

```
MOVEX A=1, AC=0, SM=0, MIX, P, P111, R= 100, H=1, MS, CONF=0020
MOVEX A=1, AC=0, SM=0, MIX, C1, P112, S= 16.6, H=1, MS, CONF=0020
MOVEX A=1, AC=0, SM=0, MIX, C2, P113, S= 16.6, H=1, MS, CONF=0020
MOVEX A=1, AC=0, SM=0, MIX, C2, P111, S= 16.6, H=1, MS, CONF=0020
```



以下为计算圆管中心位置的主控程序概要。中途有所省略。

```

REM “管内表面 3 点感应&计算中心”
...
REM “以姿势记录 P7501 示教”
USE 7501
...
REM “搜索第 1 点”
SF1 垂直方向触摸移位检测
SF3 开始 ...垂直方向的移位修正
SF1 水平方向触摸检测
SF3 结束
REM “水平方向触摸位置→V111”
GETP V111! ... 获取当前位置
LETPOSE 111,V111! ... 在姿势文件中记忆第 1 点的位置
...
REM “搜索第 2 点”
SF1 垂直方向触摸移位检测
SF3 开始 ...垂直方向的移位修正
SF1 水平方向触摸检测
SF3 结束
REM “触摸位置 V121”
GETP V121!
LETPOSE 112,V121! ... 在姿势文件中记忆第 2 点的位置
...
REM “搜索第 3 点”
SF1 垂直方向触摸移位检测
SF3 开始 ...垂直方向的移位修正
SF1 水平方向触摸检测
SF3 结束
REM “触摸位置 V131”
GETP V131!
LETPOSE 113,V131! ... 在姿势文件中记忆第 3 点的位置
NOP
REM “获取 3 点的圆心位置”
GETPOSE V401!, 111
GETPOSE V404!, 112
GETPOSE V407!, 113
REM “根据 3 点计算圆心”
GETPOSE V441!, 113 ← 保存第 3 点的位置姿势（计算出中心后更新 XYZ）
REM “算出圆心”
CALLP 5100 （※1）
REM “在 P1 中记录圆心位置”
LETPOSE 1,V441! ← 在姿势文件 P7501 的姿势变量 1 中储存计算出的圆心位置
POSESAVE
END

```

（※1） P5100 是根据 3 点计算圆心位置的公共函数。

如有必要，请与本公司进行商谈。

① 传给函数的输入数据（传递 3 点各自的正交值 XYZ）

第 1 点 (V401!, V402!, V403!) 第 2 点 (V404!, V405!, V406!) 第 3 点 (V407!, V408!, V409!)

② 来自函数的输出数据（输出计算出的管中心位置 XYZ）

V441!=中心 X V442!=中心 Y V443!=中心 Z



6.4.11 (电弧焊)生成沿圆柱管内表面的专用坐标系

如通过 7.4.10 项的方法能够计算出圆柱管的中心 A 以及中心 B 这 2 点，则可以检测出圆柱管的斜率。还可以以该斜线 (=中心 AB) 为轴生成专用的用户坐标系。

用户坐标系的生成使用应用命令 MODUSRCOORD，指定用于形成坐标轴的 3 个基点。  
3 个点的指定有 OXY / OZX / OYX 3 种方式。

以下为以 **OZX 方式** 指定的示例。

将中心 B 点的位置指定为 O (用户坐标系的原点)、从中心 B 点往 A 点的方向指定为 Z (用户坐标系的 Z 轴+方向)、边缘上感应到的 3 点中的 1 点指定为 X (用户坐标系的 X 轴+方向)。

沿下图的工件生成新的用户坐标系的步骤如下所示。

- (1) 事先以 OZX 方式登录用户坐标系。登录时的 3 个基点任意。
- (2) 执行修正用户坐标系的应用命令 MODUSRCOORD，制作作业程序。

MODUSRCOORD (用户坐标系号码、第 1 姿势变量号码、第 2 姿势变量号码、第 3 姿势变量号码)

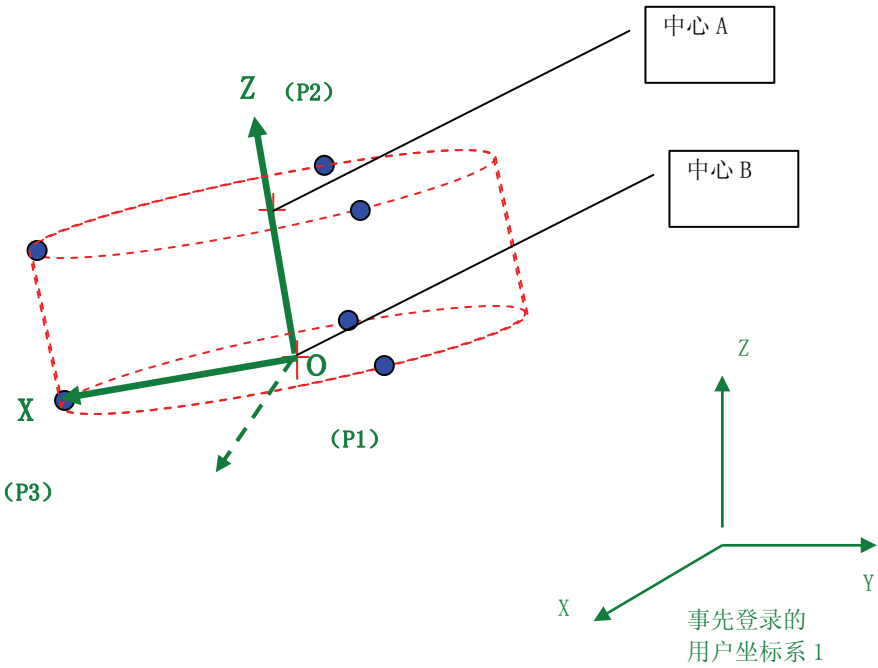
以下图为例，

- 用户坐标系号码 : 1 (事先登录的坐标系号码)
- 第 1 姿势变量号码: 1 (相当于 OZX 方式的 O 的姿势变量=P1)
- 第 2 姿势变量号码: 2 (相当于 OZX 方式的 Z 的姿势变量=P2)
- 第 3 姿势变量号码: 3 (相当于 OZX 方式的 X 的姿势变量=P3)

- (3) 新坐标系的 3 个基点分别储存在姿势变量 P1, P2, P3 中。
- (4) 执行 (2) 项中制作的作业程序。

至此，便可将登录的用户坐标系 1 变更为沿工件的坐标系。

若使用该方法，以用户坐标系为基准制作焊接程序，则即使工件的位置及斜率发生改变也能以同一焊接程序对应。



6.4.12 可以向姿势变量代入任意坐标系的指令值的应用命令

应用命令 Fn630 LETCOORDP 可以从悬式示教作业操纵按钮台上示教任意坐标系下的 TCP 和工具角度。  
使用 LETCOORDP 命令指定的 TCP 和工具角度，记录在设定画面中指定的姿势变量号码的“姿势变量”中。  
之后使用应用命令 Fn645 MOVEX 记录，读出 LETCOORDP 命令所记录的“姿势变量”，可以使机器人动作到指定的位置和工具角度。

LETCOORDP 代入姿势变量(指定坐标系)

姿势号码1

机构1:NV6-02

坐标系

X-101.2[-101.2]毫米

Y-180.0[-180.0]毫米

Z-288.9[-288.9]毫米

r90.0[90.0]度

p45.0[45.0]度

y180.0[180.0]度

坐标系

☐ 机器人 ☒ 用户 ☐ 世界 ☐ 工件

用户坐标号码1

轴角度参照☒ 前一步进 ☐ 配置

手动速度

记录现在的位置

现在的配置

写入

?

输入姿势变量号码。 [1 - 9999]

关于使用悬式示教作业操纵按钮台的数值示教，在“通过 TP 示教的机器人语言”中有着详细的说明。  
该手册中还记述有使用姿势变量的以下使用示例，请一并阅读。

- 机器人位置姿势的获取
- 在以数值指定的 4 点间作直线动作
- 指定中心位置的圆弧动作（90 度分割）
- 管中心位置的检测
- 指定中心位置的圆弧动作（45 度分割）

### 6.4.13 读出机器人当前位置并向外部输出

#### (1) 监控机器人的当前位置

使用宏函数 SYSTEM!() 的示例。

宏命令(函数)仅可使用在用户任务中。

读出的位置数据, 储存在实数变量(Vn!)中。

例如要随时监控机械坐标系的机器人当前位置时,

- ① 制作在后台定期运行的用户任务程序。  
用户任务在制作成文本文件后, 通过程序转换动作可能为执行形式。  
对于用户任务 P100, 应制作“USERTASK-A.100”的文本文件。
- ② 制作“USERTASK-A.100”的文本文件  
以 50msec 为周期, 从实数变量 V11!~V13!中读出机构 1 的机械坐标系 XYZ 位置。

```
REM “读出机器人当前位置(机械坐标系的 TCP)”
REM “V11! ← 机构 1 的当前 TCP (X 坐标)”
V11! = SYSTEM!(150)
REM “V12! ← 机构 1 的当前 TCP (Y 坐标)”
V12! = SYSTEM!(151)
REM “V13! ← 机构 1 的当前 TCP (Z 坐标)”
V13! = SYSTEM!(152)
V201% = V11! * 10
V202% = V12! * 10
V203% = V13! * 10
PAUSE 50
GOTO 1
END
```

在“(2)项 向外部输出当前位置”中进行说明。

- ③ 通过程序转换-语言转换功能, 将语言形式转换为执行形式(编码器形式或机械/各轴形式均可)。生成“USERTASK.100”。
- ④ 用户任务 P100 启动。  
【启动方法 1】 通过<维修>-[12 用户任务]启动 P100  
【启动方法 2】 在作业程序中使用“FORKMCR”命令启动 P100

#### (2) 以信号向外部输出当前位置

通过外部信号输出读出的当前位置 XYZ (V11!, V12!, V13!) 的值。

由于外部信号中无法以实数表示, 因此将值乘以 10 转化为整数输出。

在接收侧“夹具侧”, 将信号乘以 1/10, 恢复成实数值。

输出值储存在软件 PLC 和内部变量(整数变量)的共享区域中。

(共享区域: V201%—V250% → 与软件 PLC 的整数变量 D0001—D0050 对应)

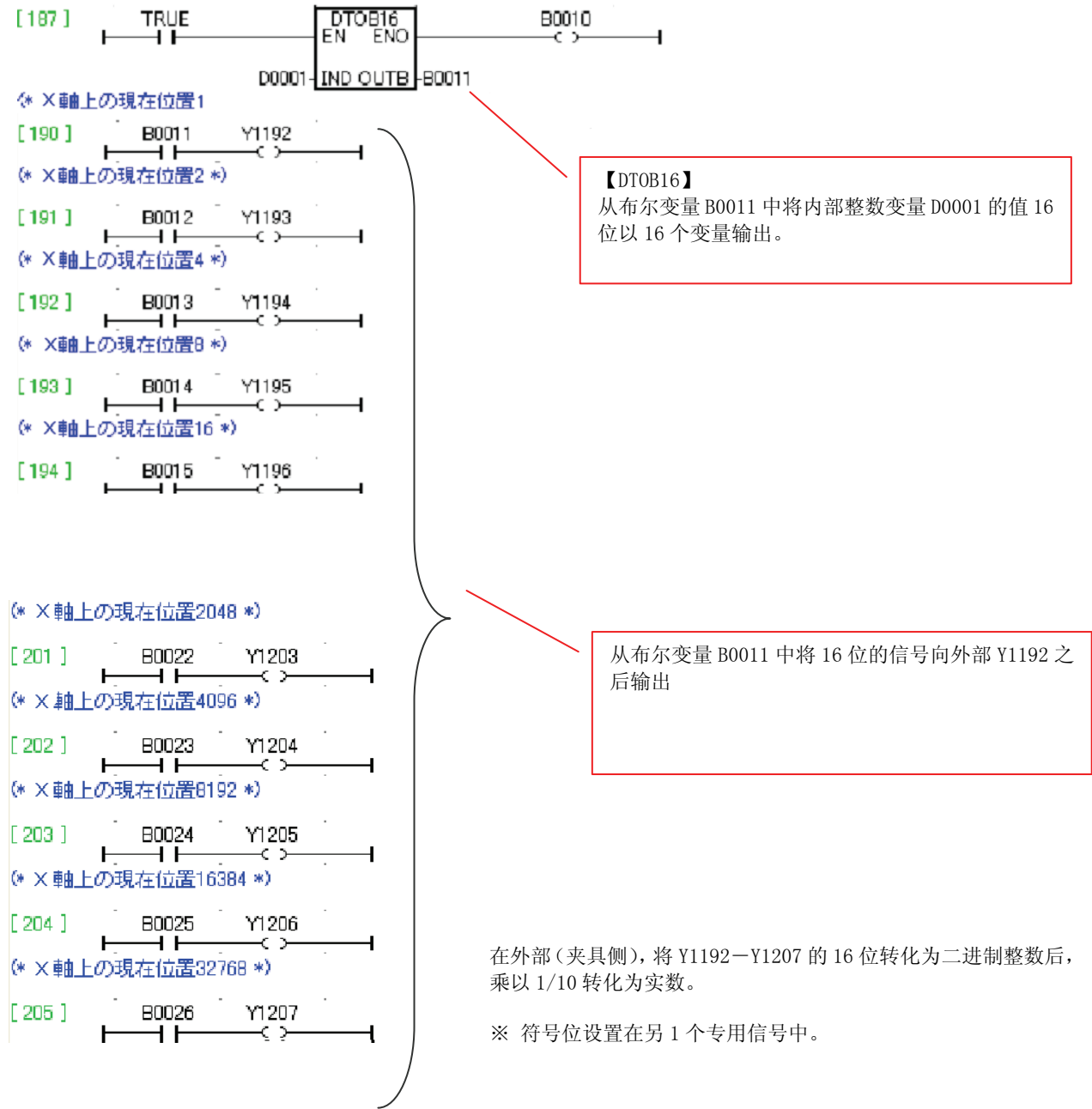
将读出的当前位置 XYZ (V11!, V12!, V13!) 的实数值乘以 10 后设定在与软件 PLC 的共享区域 (V201%, V202%, V203%) 中。

```
V201% = V11! * 10
V202% = V12! * 10
V203% = V13! * 10
```

(3) 通过软件 PLC 读取机器人语言的内部变量数据

V201%—V250% → 与软件 PLC 的整数变量 D0001—D0050 对应，  
V201% → D0001、V202% → D0002、V203% → D0003  
分别对应。

以下为软件 PLC 的示例逻辑。



## 附录：关于 2 进制数/10 进制数/16 进制数

### 附录 1) 关于 2 进制数/10 进制数/16 进制数

#### 10 进制数

10 进制数 (Decimal Number) 使用 0 到 9 这 10 个数字表示。

数字以 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 的顺序增加, 进位后变成 10。

像这样, 10 进制数按 1、10、100、1000、10000... 不断进位。

也可以换一种说法, 1 为 10 的 0 次方 ( $10^0$ )、10 为 10 的 1 次方 ( $10^1$ )、100 为 10 的 2 次方 ( $10^2$ )、1000 为 10 的 3 次方 ( $10^3$ ) ...。

所以也可以说 10 进制数按  $10^0$ 、 $10^1$ 、 $10^2$ 、 $10^3$ ... 不断进位。

例如 10 进制数的 2976 可以表示为如下形式。

$10^3$ 的位	$10^2$ 的位	$10^1$ 的位	$10^0$ 的位
2	9	7	6

如此, 以数学算式可以表示如下。

$$\begin{aligned} & 2 \times 10^3 + 9 \times 10^2 + 7 \times 10^1 + 6 \times 10^0 \\ &= 2 \times 1000 + 9 \times 100 + 7 \times 10 + 6 \times 1 \\ &= 2976 \end{aligned}$$

2 进制数

2 进制数（Binary Number）使用 0、1 这 2 个数字表示。  
数字以 0，1 的顺序增加，进位后变成 10。  
像这样，2 进制数按  $2^0(1)$ 、 $2^1(2)$ 、 $2^2(4)$ 、 $2^3(8)\cdots$  不断进位。（（）内为 10 进制数）  
例如 2 进制数的 1101 可以表示为如下形式。

$2^3$ 的位	$2^2$ 的位	$2^1$ 的位	$2^0$ 的位
1	1	0	1

如此，以数学算式可以表示如下。  
 $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$   
 $= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$   
 $= 13$ （10 进制）

实际将 2 进制数转换为 10 进制数时，采用如下的纵向计算较为简单。

$1 \times 2^3$	=	8
$1 \times 2^2$	=	4
$0 \times 2^1$	=	0
$1 \times 2^0$	=	<u>+) 1</u>
		13

反过来将 10 进制数转换为 2 进制数时，用 2 除 10 进制数，所余的商继续用 2 除，所余的商再继续用 2 除 $\cdots$ 直到所余的商为 0 为止不断重复。然后从最后的余数开始从下向上依次排列。  
例如 10 进制数的 19 可以进行如下计算。

$19 \div 2$	=	9	余	1
$9 \div 2$	=	4	余	1
$4 \div 2$	=	2	余	0
$2 \div 2$	=	1	余	0
$1 \div 2$	=	0	余	1
	=	10011		

10 进制数、2 进制数对照表																	
10 进制数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2 进制数	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000

## 16 进制数

16 进制数 (Hexadecimal Numbre) 使用 0 到 9 的数字以及 A 到 F 的字母表示。

数字以 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F 的顺序增加, 进位后变成 10。

A 为 10 进制数的 10、B 为 10 进制数的 11、C 为 10 进制数的 12、D 为 10 进制数的 13、E 为 10 进制数的 14、F 为 10 进制数的 15。

像这样, 16 进制数按  $16^0(1)$ 、 $16^1(16)$ 、 $16^2(256)$ 、 $16^3(4096)$ ... 不断进位。( ) 内为 10 进制数)

例如 16 进制数的 4E5F 可以表示如下。

$16^3$ 的位	$16^2$ 的位	$16^1$ 的位	$16^0$ 的位
4	E	5	F

如此, 以数学算式可以表示如下。

$$\begin{aligned}
 & 4 \times 16^3 + E \times 16^2 + 5 \times 16^1 + F \times 16^0 \\
 &= 4 \times 4096 + E(14) \times 256 + 5 \times 16 + F(15) \times 1 \\
 &= 20063 \text{ (10 进制)}
 \end{aligned}$$

实际将 16 进制数转换为 10 进制数时, 采用如下的纵向计算较为简单。

$4 \times 16^3$	=	16384
$E(14) \times 16^2$	=	3584
$5 \times 16^1$	=	80
$F(15) \times 16^0$	=	<u>+) 15</u>
		20063

反过来将 10 进制数转换为 16 进制数时, 用 16 除 10 进制数, 所余的商继续用 16 除, 所余的商再继续用 16 除... 直到所余的商为 0 为止不断重复。然后从最后的余数开始从下向上依次排列。例如 10 进制数的 1000 可以进行如下计算。

$1000 \div 16$	=	62	余	<b>8</b>
$62 \div 16$	=	3	余	<b>14 (E)</b>
$3 \div 16$	=	0	余	<b>3</b>
	=	3E8	(16 进制数)	

在计算机的世界中，使用 0 和 1、即 2 进制数表示一切，因此与人类世界中平常使用的 10 进制数相比，更擅长于使用 2 进制数及是 2 的倍数的 16 进制数进行计算。因此，编程中常常使用 16 进制数。另外，标注时需要区分 10 进制数、2 进制数和 16 进制数时，使用如下形式标注。

10 进制数	15d	15 <sub>(10)</sub>
2 进制数	1001b	1001 <sub>(2)</sub>
16 进制数	10h	10 <sub>(16)</sub>

“d” = Decimal 的首字母、“b” = Binary 的首字母、“h” = Hexadecimal 的首字母

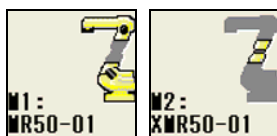
10 进制数、2 进制数、16 进制数对照表

10 进制数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2 进制数	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
16 进制数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10



## 附录：各机器人的补充

### MR 系列



对于 MR 系列，机器人语言中机构 1 包含 J2~J7 轴，机构 2 包含 J1 轴。因此，请按如下形式描述。请注意 MOVEX-J 形式时，机构 1 的轴的顺序为 J2, J7, J3, J4, J5, J6。

USE 1

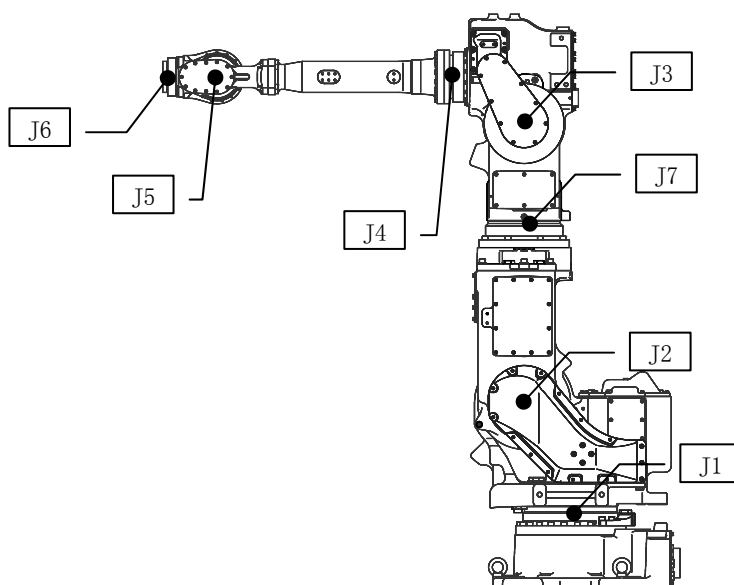
P1 = (X, Y, Z, roll, pitch, yaw, J1)

MOVEX A=1, M1X, P, (X, Y, Z, roll, pitch, yaw), R=10, H=1, MS, M2J, P, (J1), R=5, H=1

MOVEX A=1, M1J, P, (J2, J7, J3, J4, J5, J6), R=10, H=1, MS, M2J, P, (J1), R=5, H=1

MOVEX A=1, M1X, P, P1, R=5. 0, H=1, MS, M2J, P, P1, R=5, H=1

• J1~J7 表示各自轴的角度。请参考下图。



重点

通过悬式示教作业操纵按钮台进行手动操作时，为便于操作，J7 轴作为机构 2 处理（画面右上角的图标显示也如此处理）。但是通过机器人语言动作时 J1 轴作为机构 2 处理。敬请注意。

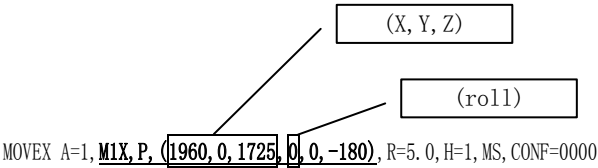
LP 系列



MOVEX-X 形式

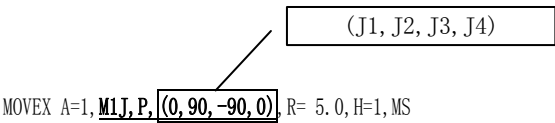
MOVEX-X 形式与其他机器人相同，可以以 (X, Y, Z, roll, pitch, yaw) 的形式描述。

(roll, pitch, yaw) 的初始值依赖于工具常数的“角度”参数，但随着 J4 轴（手腕凸缘轴）的旋转而变化的仅有 roll，余下的两个总是同一个值。推荐事先确认实际设备的工具常数和轴监视器。错误的 (roll, pitch, yaw) 设定会导致错误的动作，还请注意。

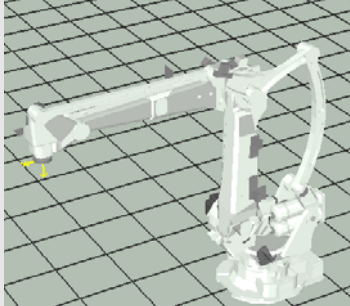


MOVEX-J 形式

请注意 MOVEX-J 形式的要素数为 4 个，机械原点姿势为 (0, 90, -90, 0)。



(参考) LP 系列的机械原点姿势



No	指令值	现在值	角度	姿势
J1	080000	080000	0. 0	X= 1960. 0
J2	080000	080000	90. 0	Y= -0. 0
J3	080000	080000	-90. 0	Z= 1725. 0
J4	080000	080000	0. 0	r= 0. 0 a= 0. 0
				p= -0. 0 b= 180. 0
				y= -180. 0 c= 180. 0

※本例中工具常数全部为初始设定。



总公司 东京都港区东新桥1-9-2 汐留住友大厦17层 邮编 105-0021  
Tel +81-3-5568-5245 Fax +81-3-5568-5236

**中国**

**那智不二越（上海）贸易有限公司**

上海市普陀区丹巴路98弄7号 龙裕财富中心11层 邮编 200062  
Tel 021-6915-2200 Fax 021-6915-5427

**重庆分公司**

重庆市江北区红鼎国际名苑C座17-18, 17-19 邮编 400020  
Tel 023-8816-1967 Fax 023-8816-1968

**沈阳分公司**

辽宁省沈阳市沈河区悦宾街1号方圆大厦304室 邮编 110000  
Tel 024-3120-2252 Fax 024-2250-5316

**北京分公司**

北京市朝阳区朝外大街乙12号 昆泰国际大厦 0-1110室 邮编 100020  
Tel 010-5879-0181 Fax 010-5879-0182

**长春事务所**

长春市绿园区普阳街1688号长融大厦B座707室 邮编 130061  
Tel 0431-8507-8700 Fax 0431-8507-8701

**广州事务所**

广州市番禺区东环路431号港信城B座505室 邮编 510120  
Tel 020-2293-9503 Fax 020-2293-9503

**那智不二越（江苏）精密机械有限公司**

江苏省张家港市经济技术开发区(南区)南园路39号 邮编 215618  
Tel 0512-3500-7616 Fax 0512-3500-7615

**上海不二越精密轴承有限公司**

上海市嘉定区马陆镇丰茂路258号易通工业园 邮编 201801  
Tel 021-6915-6200 Fax 021-6915-6202

**耐锯（上海）精密刀具有限公司**

上海市嘉定区马陆镇丰茂路258号易通工业园 邮编 201801  
Tel 021-6915-5899 Fax 021-6915-5898

**东莞建越精密轴承有限公司**

东莞市洪梅镇幽涌村  
Tel 0769-8843-1300 Fax 0769-8843-1330

**著作权 株式会社 不二越  
机器人事业部**

富山市不二越本町1-1-1, JAPAN 邮编930-8511  
Tel +81-76-423-5137  
Fax +81-76-493-5252

关于本著作的诸权利归株式会社 那智不二越公司所有。任何人在不以正式书面文件形式通知株式会社不二越公司的情况下, 禁止复制翻印其中的一部或者全部。因情况需要改版时我司将不予以特别通知。  
如存在缺页或者错页的情况下给予更换。

本产品的最终使用客户如从事军事相关, 或者武器制造的情况下, 因「外国外汇及外贸管理法」的限制, 将成为出口受限对象。在出口时, 请务必做好全面的审查并取得相关出口手续资格。

本说明书的原文是日文版。