

RCX340

编程手册

前言

非常感谢您购买 YAMAHA 机器人控制器 RCX340。

本书记载了针对 YAMAHA 机器人控制器 RCX340 的机器人程序命令等的说明。

使用本 YAMAHA 控制器时，请在阅读本书及相关使用说明书并充分注意安全的前提下正确使用本产品。

关于 YAMAHA 机器人控制器的使用，请仔细阅读 YAMAHA 控制器附带的手册。

适用机型：RCX340

安全注意事项

请务必在使用前仔细阅读

使用本产品时，请在阅读本手册及相关手册并充分注意安全的前提下正确使用本产品。

本手册中所示的注意事项为本产品相关的事项。关于使用本产品的机器人控制器系统的安全注意事项，请客户自行考虑。

在本手册中，将安全注意事项的等级区分为“注意”和“要点”。



记载了不正确使用时，预计可能会造成的危险状况、受到的中度伤害或轻伤以及仅造成财物损失的情况。



主要记载由于版本不同而其功能的不同。



简洁而明确地记载了机器人的操作步骤。

此外，即使是“注意”中记载的事项，根据情况也有可能就会导致重大后果。

所有事项均记载了重要内容，请务必遵守。

请务必妥善保管本书以便随时可以查阅的同时，并务必将本书交至最终用户手中。

■ 设计上的注意事项



程序在执行中途停止时，将再次执行停止的命令。

例如，在使用通过 MOVE 命令执行的圆弧插补移动、MOVEI 命令或 DRIVEI 命令等相对移动命令或 SEND 命令的通信命令等命令时，再次执行程序时，请特别注意。

第 1 章 编程的记述方法

1 概要	1-1
2 字符	1-1
3 程序基础知识	1-1
4 程序名	1-2
5 标识符	1-4
6 LABEL 语句	1-4
7 注释	1-5
8 命令语句格式	1-5

第 2 章 常量

1 概要	2-1
2 数值型常量	2-1
2.1 整数型常量	2-1
2.2 实数型常量	2-1
3 字符型常量	2-2

第 3 章 变量

1 概要	3-1
2 用户变量与系统变量	3-2
2.1 用户变量	3-2
2.2 系统变量	3-2
3 变量名	3-3
3.1 动态变量名称	3-3
3.2 静态变量名称	3-3
4 变量类型	3-4
4.1 数值型	3-4
4.2 字符型	3-4

5	数组	3-5
6	赋值	3-5
7	类型转换	3-6
8	值传递与引用传递	3-6
9	系统变量	3-7
9.1	坐标点变量	3-7
9.2	位移变量	3-8
9.3	并行输入变量	3-8
9.4	并行输出变量	3-9
9.5	内部输出变量	3-10
9.6	机械臂锁定输出变量	3-11
9.7	定时器输出变量	3-12
9.8	串行输入变量	3-13
9.9	串行输出变量	3-14
9.10	串行字输入	3-15
9.11	串行双倍字输入	3-15
9.12	串行字输出	3-16
9.13	串行双倍字输出	3-16
10	指定比特	3-17
11	变量的有效范围	3-18
11.1	动态变量的有效范围	3-18
11.2	静态变量的有效范围	3-18
11.3	动态数组的有效范围	3-18
12	变量的清除	3-19
12.1	动态变量的清除	3-19
12.2	静态变量的清除	3-19
第 4 章 表达式与运算		
1	算术运算	4-1
1.1	算术运算符	4-1
1.2	关系运算符	4-1
1.3	逻辑运算符	4-2

1.4	运算的优先级	4-3
1.5	类型转换	4-3
<hr/>		
2	字符串运算	4-4
2.1	字符串的联结	4-4
2.2	字符串的比较	4-4
<hr/>		
3	坐标点数据的类型	4-5
<hr/>		
4	DI / DO 条件表达式	4-6
<hr/>		
第 5 章 多台机器人控制		
1	概要	5-1
<hr/>		
2	机器人指定命令一览表	5-2
<hr/>		
第 6 章 多任务		
1	概要	6-1
<hr/>		
2	任务的定义	6-1
<hr/>		
3	任务的状态与变化	6-2
3.1	任务的启动	6-2
3.2	任务的排程	6-3
3.3	任务的等待条件	6-4
3.4	任务的中断 (SUSPEND)	6-5
3.5	任务的恢复启动 (RESTART)	6-5
3.6	任务的删除	6-6
3.7	任务的停止	6-7
<hr/>		
4	多任务程序示例	6-8
<hr/>		
5	数据的共享	6-8
<hr/>		
6	注意事项	6-9
<hr/>		
第 7 章 机器人语言一览表		
机器人语言一览表的查看方法		7-1
<hr/>		
按字母顺序		7-2

按功能		7-6
函数：按字母顺序		7-11
函数：按功能		7-13
1 ABS	计算绝对值	7-15
2 ABSRPOS	获取机器参照	7-16
3 ACCEL	设置 / 获取加速度系数参数	7-17
4 ARCHP1 / ARCHP2	设置 / 获取参数的圆弧位置	7-18
5 ARMCND	获取机械臂状态	7-20
6 ARMSEL	设定 / 获取当前的手系统选择	7-21
7 ARMTYP	设定 / 获取程序复位时的手系统	7-22
8 ASPEED	设定 / 获取指定机器人的自动移动速度	7-23
9 ATN / ATN2	计算反正切值	7-24
10 AXWGHT	设置 / 获取轴前端重量	7-25
11 CALL	调用子过程	7-26
12 CHANGE	切换机械手	7-27
13 CHGPRI	更改任务的优先级	7-28
14 CHR\$	计算带有指定字符编码的字符	7-29
15 COS	计算余弦值	7-30
16 CURTQST	获取相对于指定轴额定扭矩的当前扭矩	7-31
17 CURTRQ	获取指定轴的当前扭矩值	7-32
18 CUT	强制结束启动中的其他任务	7-33
19 DATE\$	获取日期	7-34
20 DECEL	设置 / 获取减速比率参数	7-35
21 DEF FN	定义用户可使用的函数	7-36
22 DEGRAD	角度转换（度→弧度）	7-37
23 DELAY	在指定时间等待程序的执行	7-38
24 DI	获取并行端口的输入状态	7-39
25 DIM	声明数组变量	7-40
26 DIST	计算 2 个点之间的距离	7-41
27 DO	输出至并行端口	7-42
28 DRIVE	以轴为单位进行绝对移动	7-43
29 DRIVEI	以轴为单位进行相对移动	7-49
30 END SELECT	结束 SELECT CASE 语句	7-54
31 END SUB	结束子过程的定义	7-55

32	ERR / ERL	获取错误编码 / 错误发生行编号	7-56
33	EXIT FOR	强制结束 FOR ~ NEXT 语句的循环	7-57
34	EXIT SUB	强制结束使用 SUB ~ END 定义的子过程	7-58
35	EXIT TASK	结束当前正在执行的任务	7-59
36	FOR ~ NEXT	变量超过指定值为止将循环	7-60
37	GOSUB ~ RETURN	跳转至子例程	7-61
38	GOTO	无条件跳转至指定的行	7-62
39	HALT	停止程序并进行复位	7-63
40	HALTALL	停止并复位所有的程序	7-64
41	HAND	机械手定义	7-65
	41.1 水平多关节型机器人时		7-65
	41.2 正交机器人时		7-68
42	HOLD	暂停程序	7-70
43	HOLDALL	暂停所有的程序	7-71
44	IF	评估条件表达式的值，按照条件执行命令语句。	7-72
	44.1 简单 IF 语句		7-72
	44.2 区块 IF 语句		7-73
45	INPUT	从手持编程器中对变量赋值	7-74
46	INT	舍去小数点以下部分	7-75
47	JTOXY	轴单位制转换（脉冲→毫米）	7-76
48	LEFT\$	从左端抽出字符串	7-77
49	LEFTY	将水平多关节机器人的手系统设置为左手系统	7-78
50	LEN	获取字符串的长度	7-79
51	LET	对变量赋值	7-80
52	LO	机械臂锁定输出	7-83
53	LOCx	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据	7-84
54	LSHIFT	比特的左位移	7-85
55	MCHREF	获取机器参照	7-86
56	MID\$	从指定位置获取字符串	7-87
57	MO	将指定值输出至 MO 端口（内部输出）	7-88
58	MOTOR	控制马达电源状态	7-89
59	MOVE	将机器人的所有轴进行绝对位置移动	7-90
60	MOVEI	将机器人的所有轴进行相对位置移动	7-105
61	OFFLINE	将通信端口设置为脱机模式	7-110
62	ON ERROR GOTO	在发生错误时跳转至指定标签	7-111
63	ON ~ GOSUB	执行由表达式的值指定的子例程	7-112

64	ON ~ GOTO	跳转至表达式的值所指定的标签	7-113
65	ONLINE	将通信端口设置为联机模式	7-114
66	ORD	计算字符编码	7-115
67	ORGORD	设置 / 获取机器人的返回原点顺序	7-116
68	ORIGIN	原点归复	7-117
69	OUT	开启指定端口的输出	7-118
70	OUTPOS	设置 / 获取机器人的 OUT 有效位置参数	7-119
71	PDEF	定义托盘	7-121
72	PMOVE	执行机器人的托盘移动	7-122
73	Pn	定义坐标点	7-126
74	PPNT	创建托盘坐标点数据	7-128
75	PRINT	在手持编程器上显示指定表达式的值	7-129
76	PSHFRC	设定 / 获取推压力参数	7-130
77	PSHJGSP	设定 / 获取推压判定速度阈值参数	7-131
78	PSHMTD	设定 / 获取推压方式参数	7-132
79	PSHRSLT	获取 PUSH 语句结束时的状态	7-133
80	PSHSPD	设定 / 获取推压移动速度参数	7-134
81	PSHTIME	设定 / 获取推压时间参数	7-135
82	PUSH	以轴为单位进行推压动作	7-136
83	RADDEG	单位转换 (弧度→度)	7-141
84	REM	插入注释	7-142
85	RESET	关闭指定端口的比特或清除变量	7-143
86	RESTART	重新启动暂停中的其他任务	7-144
87	RESUME	错误恢复处理后恢复执行程序	7-145
88	RETURN	使 GOSUB 中的分支处理返回至 GOSUB 的下一行	7-146
89	RIGHT\$	从右端抽出字符串	7-147
90	RIGHTY	将水平多关节机器人的手系统设置为右手系统	7-148
91	RSHIFT	比特的右位移	7-149
92	SELECT CASE ~ END SELECT	执行由表达式的值指定的命令区	7-150
93	SEND	将读出文件的数据转发至写入文件	7-151
94	SERVO	控制伺服状态	7-153
95	SET	开启指定端口的比特	7-154
96	SHARED	使不提交变量便可在子过程中参照	7-155
97	SHIFT	设置位移坐标	7-156
98	SIN	计算正弦值	7-157

99 Sn	定义位移坐标	7-158
100 SO	输出串行端口上指定的值	7-159
100 SPEED	更改程序移动速度	7-160
102 SQR	计算平方根	7-161
103 START	启动新任务	7-162
104 STR\$	将数值转换为字符串	7-163
105 SUB ~ END SUB	定义子过程	7-164
106 SUSPEND	暂停执行中的其他任务	7-166
107 SWI	切换执行程序	7-167
108 TAN	计算正切值	7-168
109 TCOUNTER	定时器、计数器	7-169
110 TIME\$	获取当前时间	7-170
111 TIMER	获取当前时间	7-171
112 TO	将指定值输出至 TO 端口	7-172
113 TOLE	设置 / 获取公差参数	7-173
114 TORQUE	设置 / 获取最大扭矩指令值	7-174
115 VAL	将字符串转换为数值	7-176
116 WAIT	等待条件表达式成立	7-177
117 WAIT ARM	等待机器人轴动作结束	7-178
118 WEIGHT	设置 / 获取前端重量参数	7-179
119 WEND	结束 WHILE 语句的命令区	7-180
120 WHERE	获取机械臂当前位置（脉冲坐标）	7-181
121 WHILE ~ WEND	在条件成立时反复动作	7-182
122 WHRXY	通过正交坐标获取机械臂当前位置	7-183
123 XYTOJ	将正交坐标数据（毫米）转换为关节坐标数据（脉冲）	7-184

第 8 章 数据文件详细说明

1 概要	8-1
1.1 数据文件种类	8-1
1.2 注意点	8-1
2 程序文件	8-2
2.1 整个程序	8-2
2.2 程序单位	8-3
3 坐标点文件	8-4

3.1	所有坐标点	8-4
3.2	坐标点单位	8-6
<hr/>		
4	坐标点注释文件	8-7
4.1	所有坐标点注释	8-7
4.2	坐标点注释单位	8-8
<hr/>		
5	参数文件	8-9
5.1	所有参数	8-9
5.2	参数单位	8-11
<hr/>		
6	位移坐标定义文件	8-12
6.1	所有位移	8-12
6.2	位移单位	8-13
<hr/>		
7	机械手定义文件	8-14
7.1	所有机械手	8-14
7.2	机械手单位	8-15
<hr/>		
8	托盘定义文件	8-16
8.1	所有托盘定义	8-16
8.2	托盘定义单位	8-18
<hr/>		
9	ALL 文件	8-20
9.1	所有 ALL 文件	8-20
<hr/>		
10	程序目录文件	8-21
10.1	所有程序目录	8-21
10.2	程序单位	8-22
<hr/>		
11	参数目录文件	8-23
11.1	所有参数目录	8-23
<hr/>		
12	变量文件	8-24
12.1	所有变量	8-24
12.2	变量单位	8-26
<hr/>		
13	常量文件	8-27
13.1	单个字符串	8-27
<hr/>		
14	数组变量文件	8-28

14.1	所有数组变量	8-28
14.2	数组变量单位	8-29
<hr/>		
15	DI 文件	8-30
15.1	所有 DI	8-30
15.2	DI 端口单位	8-31
<hr/>		
16	DO 文件	8-32
16.1	所有 DO	8-32
16.2	DO 端口单位	8-33
<hr/>		
17	MO 文件	8-34
17.1	所有 MO	8-34
17.2	MO 端口单位	8-35
<hr/>		
18	LO 文件	8-36
18.1	所有 LO	8-36
18.2	LO 端口单位	8-37
<hr/>		
19	TO 文件	8-38
19.1	所有 TO	8-38
19.2	TO 端口单位	8-39
<hr/>		
20	SI 文件	8-40
20.1	所有 SI	8-40
20.2	SI 端口单位	8-41
<hr/>		
21	SO 文件	8-42
21.1	所有 SO	8-42
21.2	SO 端口单位	8-43
<hr/>		
22	EOF 文件	8-44
22.1	EOF 数据	8-44
<hr/>		
23	串行端口通信文件	8-45
23.1	串行端口通信文件	8-45
<hr/>		
24	SIW 文件	8-46
24.1	所有 SIW	8-46
24.2	SIW 单位	8-47

25 SOW 文件	8-48
25.1 所有 SOW	8-48
25.2 SOW 单位	8-49

26 Ethernet 端口通信文件	8-50
26.1 Ethernet 端口通信文件	8-50

第 9 章 用户程序示例

1 基本篇	9-1
1.1 将坐标点数据直接写入程序中	9-1
1.2 使用坐标点编号	9-2
1.3 使用位移坐标	9-3
1.4 码垛	9-4
1.4.1 算出坐标点坐标	9-4
1.4.2 使用托盘移动	9-6
1.5 DI / DO (数字输入输出) 动作	9-7

2 应用篇	9-8
2.1 2 点间的拾放	9-8
2.2 码垛	9-10
2.3 层堆部件的拾放	9-12
2.4 部件检查 1 (多任务的示例)	9-14
2.5 使用外部装置与 RS-232C 的连接 (其 1)	9-17
2.6 使用外部装置与 RS-232C 的连接 (其 2)	9-18

第 10 章 联机命令

1 联机命令一览表	10-1
1.1 联机命令一览表：按功能	10-1
1.2 联机命令一览表：按字母顺序	10-4

2 操作、设定命令	10-6
2.1 程序操作	10-6
2.2 手动模式操作	10-12
2.3 清除输出信息缓冲区	10-13
2.4 设定输入数据	10-13

3 参照命令	10-14
3.1 获取返回原点状态	10-14

3.2	获取伺服状态	10-15
3.3	获取马达电源状态	10-15
3.4	获取访问级状态	10-16
3.5	获取断点状态	10-16
3.6	获取模式状态	10-17
3.7	获取次序程序执行状态	10-17
3.8	获取版本	10-18
3.9	获取当前位置	10-18
3.10	获取启动 / 暂停状态任务的处理	10-19
3.11	获取任务运行状态	10-20
3.12	获取位移状态	10-20
3.13	获取机械手状态	10-21
3.14	获取存储器剩余容量	10-21
3.15	获取紧急停止状态	10-22
3.16	获取值	10-22
<hr/>		
4	动作命令	10-24
4.1	绝对复位	10-24
4.2	返回原点动作	10-25
4.3	手动移动：点动	10-26
4.4	手动移动：微动	10-27
<hr/>		
5	数据文件操作命令	10-28
5.1	复制操作	10-28
5.2	删除操作	10-29
5.3	程序名更改	10-34
5.4	程序属性更改	10-34
5.5	初始化处理	10-35
5.6	数据的读出处理	10-37
5.7	数据写入处理	10-38
<hr/>		
6	实用命令	10-39
6.1	设置次序程序执行标志	10-39
6.2	日期设定	10-39
6.3	时刻设定	10-40
<hr/>		
7	机器人语言的单独执行	10-41
<hr/>		
8	控制代码	10-41

第 11 章 附录

1	保留字一览表	11-1
2	机器人语言一览表：按字母顺序	11-3
3	机器人语言一览表：按功能	11-7
4	函数：按字母顺序	11-12
5	函数：按功能	11-14

索引

第 1 章

编程的记述方法

1	概要	1-1
2	字符	1-1
3	程序基础知识	1-1
4	程序名	1-2
5	标识符	1-4
6	LABEL 语句.....	1-4
7	注释	1-5
8	命令语句格式	1-5

YAMAHA 机器人的编程使用 YAMAHA 开发的独创机器人语言。该语言是类似 BASIC (Beginner's All-purpose Symbolic Instruction Code) 语言的简单编程语言，任何人都可容易地学会，并可简易地记述复杂的机器人动作。本书对该机器人语言的记述方法及应用示例进行说明。

YAMAHA 机器人语言中可使用的的字符及符号如下。

可使用的字符全部为半角。

- 英文字母
A ~ Z, a ~ z
- 数字
0 ~ 9
- 特殊符号
() [] + - * / ^ = < > & | ~ _ % ! # \$; , . '] ' @ ?
- 假名字符 (日语文字)



MEMO

- 在手持编程器上无法输入假名字符。只有通过从主机等的通信才可使用。
- 每个空格也算作 1 个字符。



要点

- 关于子过程，请参阅第 7 章“11 CALL”及“105 SUB ~ END SUB”。



要点

- 关于用户定义函数，请参阅第 7 章“21 DEF FN”。

程序每 1 行记述 1 个命令语句。请注意，所有的行必须记述命令语句，如果有空白行，则在执行程序时会出错 (请特别注意，最后一行不得留空白行)。

若要在程序中反复进行同一个处理，可记述为子例程或子过程，只需调用主要例程即可高效创建程序。

此外，在多个程序中进行同一个处理时，可在程序名为“COMMON”的程序中记述通用例程，从多个程序调用并进行该处理。

进行已确定的计算时，可定义用户函数。由于可以方便地调用之前定义过的用户函数，因此可简易地处理复杂的计算。

并且使用多任务程序还可同时执行多个命令语句进行并行处理。

使用上述功能后，可简易地创建复杂的处理程序。

程序名，指的是可在控制器内创建的程序固有名称。只需满足以下条件，可任意决定程序名。

- 采用长度 32 字符以内的字母数字与下划线 () 的组合表示。
- 不可重复起名。

以下 2 个程序名已事先预约，带有这些名称的程序有着特殊含义。

- A) SEQUENCE
- B) COMMON

各程序的功能如下所示。

A) SEQUENCE

功能 RCX340 控制器与通常的机器人程序不同，其可运行对机器人输入输出 (DI, DO, MO, LO, TO, SI, SO) 进行高速处理的程序 (次序程序)。使用此功能在控制器内创建简易类似 PLC 时，使用 "SEQUENCE" 的程序名。

在次序程序中输入了专用输入：DI10(次序控制输入)信号后，无论程序的执行状态如何，控制器在自动或手动模式时均将按照一定周期运行 (周期取决于程序容量。详细说明，请参阅“次序程序规格”。)

执行上述操作后，可监视传感器、按钮开关、电磁阀等的输入输出信号或使其运行。此外，次序程序采用机器人语言记述，因此无需使用新的语言即可简易地创建。

SAMPLE

```
DO (20) = ~DI (20)
DO (25) = DI (21) AND DI (22)
MO (26) = DO (26) OR DO (25)
:
```

参照 详细说明，请参阅“次序功能”。

B) COMMON

功能

当存在多个机器人程序且在各程序中进行同一个处理时，可通过用 "COMMON" 的名称另外准备一个程序，并在该程序中记述通用的处理例程，以便可从多个程序调用并执行该处理。使用上述方法可有效使用程序区域。

以下样本程序，在 COMMON 中记述了“计算 2 点间距离 (SUB *DISTANCE)”与“计算面积 (*AREA)”这 2 个通用例程，并从其他程序 (SAMPLE1 及 SAMPLE2) 中调出。执行了 SAMPLE1 或 SAMPLE2 后，将执行 SUB *DISTANCE (A!,B!,C!) 及 *AREA 例程。

SAMPLE

程序名：SAMPLE1

```
X!=2.5  
Y!=1.2  
CALL *DISTANCE (X!, Y!, REF C!)  
GOSUB *AREA  
PRINT C!, Z!  
HALT
```

程序名：SAMPLE2

```
X!=5.5  
Y!=0.2  
CALL *DISTANCE (X!, Y!, REF C!)  
GOSUB *AREA  
PRINT C!, Z!  
HALT
```

程序名：COMMON 通用例程

```
SUB *DISTANCE (A!, B!, C!)  
  C!=SQR (A!^2+B!^2)  
END SUB  
*AREA:  
  Z!=X!*Y!  
RETURN
```

参照

详细说明，请参阅本书中的各命令。

1

2

3

4

5

6

标识符、变量名、过程名等所使用的字符或数字组合称为“标识符”。可在以下范围内任意决定标识符。

- 使用英文字母、数字、下划线(_)中的任意一个。不包括特殊符号。此外,下划线(_)无法用作起始字符。
- 长度最多为 32 个字符(超过 32 个字符的部分将被舍去)。
- 可使用的标识符的最大数量,因标识符长度的不同而异。所有标识符的长度为 32 个字符时,标识符数为最大,可使用局部变量(本地变量):最多 128 个(每 1 个程序任务);全局变量(全球变量):最多 512 个。
- 无法使用与保留字相同的变量名以及以系统变量中定义名称起始的变量名。字符拼写的最前面必须以英文字母开始。但是,在标签名中,"*"之后的字符是数字也不要紧。

SAMPLE

LOOP, SUBROUTINE, GET_DATA

参照 关于保留字,请参阅第 11 章“1 保留字一览表”。

在程序行中定义<标签>。

格式

* <标签> :

<标签>必须一贯以“*”起始。此外,必须从行的起始开始记述。

进行标签定义时,必须在末尾加上“:”,但是在程序中指定跳转目标时则不需要。

1. 第一个字符必须以英文字母或数字起始。
2. 使用英文字母、数字、下划线(_)中的任意一个。不包括特殊符号。
3. 长度最多为 32 个字符(超过 32 个字符的部分将被舍去)。

SAMPLE

```
*ST:  定义标签 ..... *ST
      MOVE P, P0
      DO (20) = 1
      MOVE P, P1
      DO (20) = 0
      GOTO *ST ..... 跳转至 *ST
      HALT
```

REM 或 " " 以后的文字被识别为注释。不执行注释语句。
此外，也可从一行的中间部分开始书写注释。

SAMPLE

```
REM *** MAIN PROGRAM ***
    (主程序)
' *** SUBROUTINE ***
    (子例程)
HALT ' HALT COMMAND ..... 也可从一行的中间部分开始记述
```

格式

```
[ < 标签 > : ] < 语句 > [ < 操作数 > ]
```

机器人语言中的 1 个命令语句必须按照以下格式记述在 1 行以内。

- [] 表示可以省略。但是，指定机器人编号、点变量、位移变量的 [] 除外。这些语句必须直接使用 [] 记述。
- < > 中的项目采用规定格式记述。
- 不在 < > 中的项目直接记述其中所写内容。
- | | 中的项目记述了任意一个所记述的项目。
- 标签可以省略。声明标签时，必须以星号 (*) 开始，必须以冒号 (:) 结束（作为分支记述标签时，不需要冒号）。

关于标签请参阅本章“6 LABEL 语句”。

- 根据命令语句的不同，有时不需要操作数。
- 没有指定分支时，按照从上面的行至下面行的顺序执行程序。

1 行中可记述的最大字符数为 255 个字符。

第 2 章

常量

1	概要	2-1
2	数值型常量	2-1
3	字符型常量	2-2

常量大致可分为“数值型”与“字符型”，而这 2 种类型又可分别细分为以下类型。

分类	类型	详细内容 / 范围
数值型	整数型	10 进制常量 -1,073,741,824 ~ 1,073,741,823
		2 进制常量 &B0 ~ &B11111111
		16 进制常量 &H80000000 ~ &H7FFFFFFF
实数型	实数型	单精度实数 -999,999.9 至 +999,999.9
		指数形式单精度实数 -1.0*10 ³⁸ ~ +1.0*10 ³⁸
字符型	字符串	255 个字节以内的英文字母、数字、特殊字符、假名字符

2.1

整数型常量

- 10 进制常量
从 -1,073,741,824 至 1,073,741,823 的整数。
- 2 进制常量
不带 8 比特以下符号的 2 进制数字。在最前面加上表示 2 进制的 &B。
范围：&B0 (10 进制：0) ~ &B11111111 (10 进制：255)
- 16 进制常量
带有 32 比特以下符号的 16 进制数字。在最前面加上表示 16 进制的 &H。
范围：&H80000000 (10 进制：-2,147,483,648) ~ &H7FFFFFFF (10 进制：2,147,483,647)

2.2

实数型常量

- 单精度实数
范围：从 -999,999.9 至 +999,999.9 的实数
· 整数与小数加在一起 7 位数以下 (例如：.0000001 也可以)。
- 指数形式的单精度实数
范围：-1.0*10³⁸ ~ +1.0*10³⁸
· 尾数部分的整数与小数加在一起 7 位数以下

示例： -1. 23456E-12

3. 14E0

1. E5



- 如果用带符号的 16 进制表达 -1,073,741,824 ~ 1,073,741,823，则整数型常量的范围为 &HC0000000 ~ &H3FFFFFFF。

字符型常量指的是用双引号 (") 括起来的字符串数据。长度为 255 个字节以内，在字符串中可使用英文大写字母、数字、特殊符号及假名字符。

作为字符串使用双引号 (") 时，接在 (") 后面记述。

SAMPLE

```
"YAMAHA ROBOT"
```

```
"EXAMPLE OF" "A" " " ..... 变为 EXAMPLE OF "A"
```

```
PRINT "COMPLETED"
```

```
"YAMAHA ROBOT"
```

第 3 章

变量

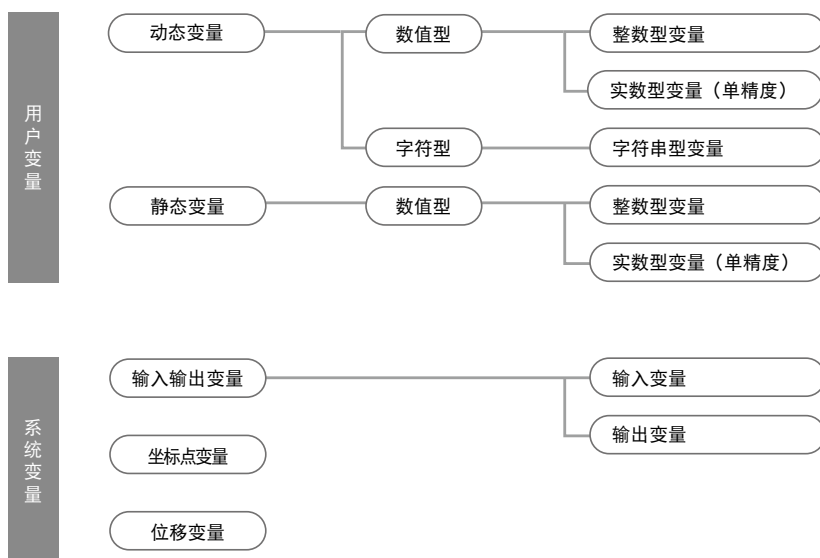
1	概要	3-1
2	用户变量与系统变量.....	3-2
3	变量名	3-3
4	变量类型.....	3-4
5	数组	3-5
6	赋值	3-5
7	类型转换.....	3-6
8	值传递与引用传递	3-6
9	系统变量.....	3-7
10	指定比特.....	3-17
11	变量的有效范围	3-18
12	变量的清除	3-19

变量分为可任意定义的“用户变量”与预先决定了名称及功能的“系统变量”。

用户变量分为通过程序编辑、程序复位、程序切换等清除内容的“动态变量”及只要不清除存储器内容就始终保留的“静态变量”。其中，动态变量可任意定义名称，还可使用数组变量。

若要使用变量，只需决定变量的名称及类型并记述至程序中即可，无需进行声明。但是，数组变量必须预先使用 DIM 语句进行定义。

用户变量与系统变量



35301-R9-00

参照 关于数组的详细内容，请参阅本章“5 数组”。

2 用户变量与系统变量

2.1 用户变量

数值型变量分为“整数型”与“实数型”，可使用的数值范围也有所不同。此外，本章节还将介绍字符型变量、数组变量等可使用的变量类型与数据范围。

分类	变量类型	详细内容、值的范围
动态变量	数值型	整数型变量 -1,073,741,824 ~ 1,073,741,823 (带符号的 16 进制: &HC0000000 ~ &H3FFFFFFF) 实数型变量 (单精度) -1.0*10 ³⁸ ~ +1.0*10 ³⁸
	字符型	字符串型变量 255 个字节以内的英文字母、数字、特殊字符、假名字符
静态变量	数值型	整数型变量 -1,073,741,824 ~ 1,073,741,823 实数型变量 (单精度) -1.0*10 ³⁸ ~ +1.0*10 ³⁸
	字符型	字符串型变量 255 个字节以内的英文字母、数字、特殊字符、假名字符
数组变量	数值型	整数型数组 -1,073,741,824 ~ 1,073,741,823 实数型数组 (单精度) -1.0*10 ³⁸ ~ +1.0*10 ³⁸
	字符型	字符串型数组 255 个字节以内的英文字母、数字、特殊字符、假名字符



要点

· 数组变量是动态变量。

2.2 系统变量

系统变量如下所示预先确定了名称，无法进行更改。

分类	类型	详细	具体示例
输入输出变量	输入变量	来自外部的信号、状态输入	DI, SI, SIW, SID
	输出变量	输出至外部的信号、状态	DO, SO, SOW, SOD
坐标点变量		使用坐标点数据	Pnnnn
位移变量		通过常量或表达式指定位移编号	Sn

参照 关于详细内容请参阅本章“9 系统变量”。

3.1 动态变量名称

遵照以下规格可任意确定动态变量的变量名。

- 使用英文字母、数字、下划线(_)中的任意一个。不包括特殊符号。
- 长度最多为 32 个字符(超过 32 个字符的部分将被舍去)。
- 第一个字符必须以英文字母起始。

SAMPLE

COUNT	○	可以使用
COUNT123	○	可以使用
2COUNT	×	不可使用

- 不可使用与保留字同名的变量名。
- 不可使用系统变量(已定义变量)中已使用的名称作为变量名的起始。
FN, DIn, DO_n, MO_n, LO_n, TO_n, SIn, SO_n, P_n, S_n, H_n(n 为数值)

SAMPLE

COUNT	○	可以使用
ABS	×	(保留字)
FNAME	×	(FN: 已定义变量)
S91	×	(S _n : 已定义变量)

参照 关于保留字, 请参阅第 11 章“1 保留字一览表”。

3.2 静态变量名称

静态变量的变量名如下规定, 不可使用其他名称。

变量类型	变量名
整数型	SGIn (n: 0 ~ 31)
实数型	SGRn (n: 0 ~ 31)

只要不通过系统模式或联机命令进行初始处理, 静态变量的内容将不会被清除。

参照 关于静态变量的清除, 请参阅本章“12 变量的清除”。

变量类型通过在变量名的最后加上类型声明字符来区分。

但是，静态变量的名称已经根据变量类型规定了，因此不需要类型声明字符。

类型声明字符	变量类型	具体示例
\$	字符型变量	STR1\$
%	整数型变量	CONT%, ACT% (1)
!	实数型变量	CNT1!, CNT1

MEMO

- 无类型声明字符时，将识别为实数型变量。
- 使用了相同标识符的变量，由于变量类型的不同而被识别为其他变量。
 - ASP_DEF%..... 整数型变量
 - ASP_DEF..... 实数型变量
) → 分别作为不同变量处理。
- ASP_DEF!..... 实数型变量
- ASP_DEF..... 实数型变量

4.1

数值型



要点

- 如果将实数赋值给整数型的变量，其将被四舍五入。详细说明，请参阅第4章“1.5 类型转换”。

整数型

整数型变量、整数型数组变量的元素可处理的整数值为 $-1,073,741,824 \sim 1,073,741,823$ (用带符号的16进制表示为 &HC0000000 ~ &H3FFFFFFF)。

示例：R1% = 10
R2% (2) = R1% + 10000



要点

- 实数型的"! "可以省略。

实数型

实数型变量、实数型数组变量的元素可处理的实数值为 $-1.0 \times 10^{38} \sim 1.0 \times 10^{38}$ 。

示例：R1! = 10.31
R2! (2) = R1% + 1.98E3

4.2

字符型

字符型变量、字符串数组的元素可处理的字符串最大为255个字符。

在字符串中可使用英文字母、数字、特殊字符及假名字符。

示例：R1\$ = "YAMAHA"
R2\$(2) = R1\$ + "MOTOR" "YAMAHA MOTOR"

动态变量可使用数值型、字符型中的任意一种数组。

使用数组后，可统一管理连续而相同类型的多个数据。

有关数组的各元素，请参照每个变量名后面的括号中加注的角标。角标可使用整数或 <表达式>，最多可使用 3 维。

若要使用数组，必须预先通过 DIM 语句声明，可使用的元素数最大为已声明的角标 + 1 为止 (0 ~ 角标)。

MEMO

- 数组变量全部为动态变量 (关于动态变量，请参阅本章“11 变量的有效范围”)。
- DIM 语句中可定义的数组变量的大小根据程序的大小变化。

格式

```
<变量名> [ | % | ] ( <表达式> , [ <表达式> , [ <表达式> ] ] )
           | ! |
           | $ |
```

SAMPLE

```
A% (1) ..... 整数型数组变量
DATA! (1, 10, 3) ..... 单精度实数型数组变量 (3 维数组)
STRING$ (10) ..... 字符型数组变量
```

对变量赋值时，也可使用赋值语句 (LET)。

MEMO

- "LET" 通常省略，可直接记述赋值语句。

格式

```
[LET] <变量> = <表达式>
```

将想要赋值的变量放在左边，想要赋给的值或 <表达式> 写入右边。可在 <表达式> 中记述常量、变量、计算公式等。

参照 详细说明，请参阅第 7 章“51 LET (赋值语句)”。

赋给与变量不同类型的值时，数据类型将如下转换。

- 将实数赋值给整数型
小数点以下四舍五入赋值。
- 将整数赋值给实数型
按照原样赋值，作为实数处理。
- 将数值赋值给字符串型
数值将被自动转换为字符串赋值。
- 将字符串赋值给数值型
无法赋值。执行程序时将出错。请通过 "VAL" 命令转换成数值后赋值。

通过 CALL 语句调用子过程时可传递变量。
变量的引用方式分为值传递与引用传递两种。

值传递

在子过程中传递变量值的方法。即使在子过程内更改了值，**调用源变量的内容也不会改变**。
在 CALL 语句的 * 实参中指定了常量、表达式、变量、数组的元素（数组名的后面带有 (<角标>)) 时，将进行值传递。

* 实参：实际参数的简称。

引用传递

在子过程中向变量传递参照（存储器中的地址）的方法。如果在子过程内更改了值，**源变量的内容也将相应更改**。

在 CALL 语句实参中指定全部数组（数组名后面带有 ()）或者在实参前面加上 REF 时，将进行引用传递。

值传递与引用传递

值传递

```
X%=5
CALL *TEST ( X% )
PRINT X%
HALT
` SUB ROUTINE
SUB *TEST ( A% )
  A%=A%*10
END SUB
```

执行结果 X% 的值仍然为“5”

引用传递

```
X%=5
CALL *TEST ( REF X% )
PRINT X%
HALT
` SUB ROUTINE
SUB *TEST ( A% )
  A%=A%*10
END SUB
```

执行结果 X% 的值变为“50”

35302-R7-00

以下变量已预先定义。无法使用以下变量名起始的名称。

变量种类	格式	含义
坐标点变量	Pnnn / P" [" <表达式> "]" "	指定坐标点编号
位移变量	Sn / S" [" <表达式> "]" "	通过常量或表达式指定位移编号
并行输入变量	DI (mb), DI _m (b)	并行输入信号的状态
并行输出变量	DO (mb), DO _m (b)	并行输出信号的指定及状态
内部输出变量	MO (mb), MO _m (b)	控制器内部输出信号的指定及状态
机械臂锁定输出变量	LO (mb), LO _m (b)	禁止各轴移动
定时器输出变量	TO (mb), TO _m (b)	用于次序程序的定时器功能
串行输入变量	SI (mb), SI _m (b)	串行输入信号的状态
串行输出变量	SO (mb), SO _m (b)	串行输出信号的指定及状态
串行字输入	SIW (m)	串行输入的字信息状态
串行双倍字输入	SID (m)	" (双倍字)
串行字输出	SOW (m)	串行输出的字信息输出及状态
串行双倍字输出	SOD (m)	" (双倍字)

9.1

坐标点变量

通过常量或表达式指定坐标点编号

格式

Pnnnnn 或 P" [" <表达式> "]" "

值 n: 坐标点编号 0 ~ 9
"["]" 并非省略符号。[] 表示直接记述。

功能 通过最前面的字符 P 及在其之后的 5 位数以下的数值或 "[" <表达式> "]" 表达坐标点编号。
可在 0 ~ 29999 的范围内指定坐标点编号。

示例： P0
P110
P[A]
P[START_POINT]
P[A (10)]

通过常量或表达式指定位移编号

格式

Snn 或 S" [<表达式>] "

值 n : 位移编号..... 0 ~ 9
"[]" 并非省略符号。[] 表示直接记述。

功能 通过最前面的字符 S 及在其之后的 2 位数数值或 "[<表达式>]" 表达位移编号。位移编号可指定 0 ~ 39。

示例 : S1
S[A]
S[BASE]
S[A (10)]



MEMO

· 若要更改各位移编号所设置的“位移坐标范围”，请通过手持编程器进行更改。

表示并行输入信号的状态

格式 1

DI_m ([b, . . . , b])

格式 2

DI (mb, . . . , mb)

值 m : 端口编号..... 0 ~ 7, 10 ~ 17, 20 ~ 27
b : 指定比特..... 0 ~ 7
在格式 1 中省略了 [b, . . . , b] 时, 8 比特均将成为对象。

示例 : A%=DI1 ()
→ 将从 DI (17) 至 DI (10) 的输入状态赋值给变量 A%
A% 将被赋予 0 ~ 255 的整数
A%=DI5 (7, 4, 0)
→ 将 DI (57), DI (54), DI (50) 的输入状态赋值给变量 A%
(上述信号均变为 1 (ON) 时, A%=7)
A%=DI (27, 15, 10)
→ 将 DI (27), DI (15), DI (10) 的输入状态赋值给变量 A%
(上述信号除 DI (10) 以外均变为 1 (ON) 时, A%=6)
WAIT DI (21) =1
→ 等待 DI (21) 变为 1 (ON)



MEMO

· 指定多个比特时, 请从左边开始降序(大→小)记述。
· 实际不存在输入端口时, 将输入 0。

表示并行输出信号的指定或输出状态

格式 1

DOm ([b, . . . , b])

格式 2

DO (mb, . . . , mb)

值 m: 端口编号.....0 ~ 7, 10 ~ 17, 20 ~ 27
 b: 指定比特.....0 ~ 7
 省略了 [b, . . . , b] 时, 8 比特均将成为对象。

示例: A%=DO2 ()

→ 将从 DO (27) 至 DO (20) 的输出状态赋值给变量 A%

A%=DO5 (7, 4, 0)

→ 将 DO (57), DO (54), DO (50) 的输出状态赋值给变量 A%
 (上述信号均变为 1 (ON) 时, A%=7)

A%=DO (37, 25, 20)

→ 将 DO (37), DO (25), DO (20) 的输出状态赋值给变量 A%
 (上述信号除 DO (20) 以外均变为 1 (ON) 时, A%=6)

DO3 () =B%

→ 使 DO (37) 至 DO (30) 的输出变化为用 B% 表示的状态

例如, B% 为 123 时, 用二进制表示 123 就变为 "01111011",
 DO (37) 及 DO (32) 变为 "0", 其他比特变为 "1"

DO4 (5, 4, 0) =&B101

→ 将 DO (45) 及 DO (40) 设置为 "1", 将 DO (44) 设置为 "0"

MEMO

- 指定多个比特时, 请从左边开始降序(大→小)记述。
- 实际不存在输出端口时, 将输入 0。

控制器内部输出信号的指定及状态

格式 1

MOm ([b, . . . , b])

格式 2

MO (mb, . . . , mb)

值 m : 端口编号.....0 ~ 7, 10 ~ 17, 20 ~ 27, 30 ~ 33
 b : 指定比特.....0 ~ 7
 · 在格式 1 中省略了 [b, . . . , b] 时, 8 比特均将成为对象。

功能 对于仅限控制器内部使用的内部输出变量, 可指定及参照内部输出变量的状态。
 用于与次序程序进行信号交换等用途。
 端口 30 ~ 33 是专用的内部输出变量, 只可参照 (无法更改)。

1. 端口 30 中保存了按顺序从 0 比特开始 1 ~ 8 轴原点传感器的状态, 端口 31 中保存了按顺序从 0 比特开始 9 ~ 16 轴的原点传感器的状态。
 原点传感器开启时为 1, 关闭时为 0。
2. 端口 34 中保存了按顺序从 0 比特开始 1 ~ 8 轴的保持状态, 端口 35 中保存了按顺序从 0 比特开始 9 ~ 16 轴的保持状态。
 轴, 保持时为 1, 未保持状态下为 0。

比特	7	6	5	4	3	2	1	0
端口 30	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
端口 31	16 轴	15 轴	14 轴	13 轴	12 轴	11 轴	10 轴	9 轴
原点传感器状态 0 : 关闭 / 1 : 开启 (未连接轴为 1)								
端口 34	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
端口 35	16 轴	15 轴	14 轴	13 轴	12 轴	11 轴	10 轴	9 轴
保持状态 0 : 未保持 / 1 : 保持 (未连接轴为 1)								

MEMO

- 未连接原点传感器的轴变为常开状态。
- 保持状态, 指的是轴的移动停止并定位至公差内的状态。
- 伺服禁止时, 将变为未保持状态。
- 未使用的轴变为 1。
- 状态保存了从机器人 1 所使用的小编号轴开始的各轴状态。
 例) 机器人 1 为 5 轴、机器人 2 为 4 轴构成的情况下, 端口 30 的比特 0 ~ 4 保存机器人 1 的 1 ~ 5 轴的状态; 端口 30 的比特 5 ~ 7 保存机器人 2 的 1 ~ 3 轴的状态; 端口 31 的比特 0 保存机器人 2 的第 4 轴的状态。

示例：A%=MO2 ()
 →将从 MO (27) 至 MO (20) 的内部输出状态赋值给变量 A%
 A%=MO5 (7, 4, 0)
 →将 MO (57),MO (54),MO (50) 的内部输出状态赋值给变量 A%
 (例如：上述信号均为 1 (ON) 时，A%=7)
 A%=MO (37, 25, 20)
 →将 MO (37),MO (25),MO (20) 的内部输出状态赋值给变量 A%
 (例如：上述信号除 MO (25) 以外均为 1 (ON) 时，A%=5)

MEMO

· 指定多个比特时，请从左边开始降序(大→小)记述。

9.6 机械臂锁定输出变量

对各轴进行禁止移动的设置

格式 1

LOm ([b, . . . , b])

格式 2

LO (mb, . . . , mb)

值 m：端口编号.....0、1
 b：指定比特.....0～7
 · 在格式 1 中省略了[b, . . . , b]时，8 比特均将成为对象。

功能 可输出至变量或参照变量。
 端口 0 从 0 比特开始依次对应 1～8 轴；端口 1 从 0 比特开始依次对应 9～16 轴。
 比特开启状态下，禁止对应轴移动。

示例：A%=LO0 ()
 →将 LO (07) 至 (00) 的机械臂锁定状态赋值给变量 A%
 A%=LO0 (7, 4, 0)
 →将 LO (07), LO (04), LO (00) 的机械臂锁定状态赋值给变量 A%
 () 上述信号均为 1 (ON) 时，A%=7)
 A%=LO0 (06, 04, 01)
 →将 LO (06), LO (04), LO (01) 的机械臂锁定状态赋值给变量 A%
 (上述信号除 LO (01) 以外均为 1 (ON) 时，A%=6)

MEMO

- 指定多个比特时，请从左边开始降序(大→小)记述。
- 即使 1 个轴的机械臂锁定有效，也无法从伺服 OFF 状态更改为伺服 ON 状态。
- 在手动模式中进行 JOG 移动时，即使存在机械臂锁定有效的轴，仍然可使机械臂锁定无效的轴动作。
- 通过程序等执行移动命令时，如果动作对象轴中包含了机械臂锁定有效的轴，将发生 [12.3 XX. Arm locked] 的错误。
- 机械臂锁定从机器人 1 所使用的小编号轴开始依次对应各轴。
 例) 机器人 1 为 5 轴、机器人 2 为 4 轴构成的情况下，端口 0 的比特 0～4 对应机器人 1 的 1～5 轴的状态，端口 0 的比特 5～7 对应机器人 2 的 1～3 轴的状态，端口 1 的比特 0 进行机器人 2 的第 4 轴的禁止移动设定。

用于次序程序的定时器功能

格式 1

TOm ([b, . . . , b])

格式 2

TO (mb, . . . , mb)

值

m : 端口编号.....0、1

b : 指定比特.....0 ~ 7

· 在格式 1 中省略了 [b, . . . , b] 时, 8 比特均将成为对象。

功能

可更改及参照变量的内容。

定时器功能仅在次序程序中有效。在普通程序中, 只可参照输出状态。

有关次序程序中的使用示例, 请参阅<输入输出变量>的定时器使用示例。

示例 : A%=TO0 ()

→ 将 TO (07) 至 TO (00) 的状态赋值给变量 A%

A%=TO0 (7, 4, 0)

→ 将 TO (07), TO (04), TO (00) 的状态赋值给变量 A%

(上述信号均为 1 (ON) 时, A%=7)

A%=TO (06, 04, 01)

→ 将 TO (06)、TO (04)、TO (01) 的状态赋值给变量 A%

(上述信号除 TO (01) 以外均为 1 (ON) 时, A%=6)



MEMO

· 指定多个比特时, 请从左边开始降序(大→小)记述。

表示串行输入信号的状态

格式 1

SI m ([b , . . . , b])

格式 2

SI (m b , . . . , m b)

值 m : 端口编号.....0 ~ 7, 10 ~ 17, 20 ~ 27

b : 指定比特.....0 ~ 7

· 在格式 1 中省略了 [b , . . . , b] 时, 8 比特均将成为对象。

示例: A%=SI1 ()

→ 将 SI (17) 至 SI (10) 的输入状态赋值给变量 A%

A%=SI5 (7, 4, 0)

→ 将 SI (57), SI (54), SI (50) 的输入状态赋值给变量 A%

(上述信号均变为 1 (ON) 时, A%=7)

A%=SI (27, 15, 10)

→ 将 SI (27), SI (15), SI (10) 的输入状态赋值给变量 A%

(上述信号除 SI (10) 以外均为 1 (ON) 时, A%=6)

WAIT SI (21) =1

→ 等待 SI (21) 变为 1 (ON)

MEMO

- 指定多个比特时, 请从左边开始降序(大→小)记述。
- 实际不存在串行端口时, 将输入 0。

表示串行输出信号的指定或输出状态

格式 1

SOm ([b, . . . , b])

格式 2

SO (mb, . . . , mb)

值

m : 端口编号 0 ~ 7, 10 ~ 17, 20 ~ 27

b : 指定比特 0 ~ 7

· 在格式 1 中省略了 [b, . . . , b] 时, 8 比特均将成为对象。

示例 : A%=SO2 ()

→ 将 SO (27) 至 SO (20) 的输出状态赋值给变量 A%

A%=SO5 (7, 4, 0)

→ 将 SO (57), SO (54), SO (50) 的输出状态赋值给变量 A%
(上述信号均变为 1 (ON) 时, A%=7)

A%=SO (37, 25, 20)

→ 将 SO (37), SO (25), SO (20) 的输出状态赋值给变量 A%
(上述信号除 SO (25) 以外均为 1 (ON) 时, A%=5)

SO3 () =B%

→ 使 SO (37) 至 SO (30) 的输出变化为用 B% 表示的状态

例如, B% 为 123 时, 用二进制表示 123 就变为 "01111011"。

SO (37) 及 SO (32) 变为 "0", 其他比特变为 "1"

SO4 (5, 4, 0) =&B101

→ 将 SO (45) 及 SO (40) 设置为 "1", 将 SO (44) 设置为 "0"



MEMO

- 指定多个比特时, 请从左边开始降序(大→小)记述。
- 实际不存在串行端口时, 将不进行外部输出。

表示串行输入的字信息状态

格式

SIW (m)

值 m : 端口编号 2 ~ 15
获取范围为 0 (&H0000) ~ 65, 535 (&HFFFF)。

示例 : A%=SIW (2)
→ 将 SIW (2) 的输入状态赋值给变量 A%
A%=SIW (15)
→ 将 SIW (15) 的输入状态赋值给变量 A%

MEMO

- 作为无符号的字数据处理。
- 实际不存在串行端口时, 将被输入 0。

通过双倍字表示串行输入的字信息状态

格式

SID (m)

值 m : 端口编号 2, 4, 6, 8, 10, 12, 14
获取范围为 -1,073,741,824 (&HC0000000) ~ 1,073,741,823 (&H3FFFFFFF)。

示例 : A%=SID (2)
→ 将 SIW (2), SIW (3) 的输入状态赋值给变量 A%
A%=SID (14)
→ 将 SIW (14), SIW (15) 的输入状态赋值给变量 A%

MEMO

- 作为带符号的双倍字数据处理。
- 实际不存在串行端口时, 将被输入 0。
- 使用获取范围以外 (&H80000000 ~ &HBFFFFFFF, &H40000000 ~ &H7FFFFFFF) 的值将出错。
- 配置数据时, 低位端口编号的数据配置至低位地址。
例如, SIW (2) = &H2345, SIW (3) = &H0001 时, SID (2) = &H00012345。

在串行输出的字信息中显示输出或输出状态

格式

SOW (m)

值

m : 端口编号..... 2 ~ 15

输出范围为 0 (&H0000) ~ 65, 535 (&HFFFF)。

但是, 输出了负值时, 将被输出转换为十六进制的低位字信息。

示例 : A%=SOW (2)

→ 将 SOW (2) 的输出状态赋值给变量 A%

SOW (15) =A%

→ 将变量 A% 的内容赋值给 SOW (15)

变量 A% 的值超过输出范围时, 将被赋值给低位字信息

SOW (15) =-255

→ 将 -255 (&HFFFFFF01) 的内容赋值给 SOW (15)

-255 因是负值, 将被赋值给低位的字信息 (&HFF01)

MEMO

- 作为无符号的字数据处理。
- 实际不存在串行端口时, 将不被输出至外部。
- 赋予了超出输出范围的值时, 将被输出低位的 2 字节信息。

在串行输出的字信息中用双倍字显示输出或输出状态

格式

SOD (m)

值

m : 端口编号 2, 4, 6, 8, 10, 12, 14

输出范围为 -1,073,741,824 (&HC0000000) ~ 1,073,741,823 (&H3FFFFFFF)。

示例 : A%=SOD (2)

→ 将 SOD (2) 的输入状态赋值给变量 A%

SOD (14) =A%

→ 将变量 A% 的内容赋值给 SOD (14)

- 作为带符号的双倍字数据处理。
- 实际不存在串行端口时, 将不被输出至外部。
- 使用输出范围以外 (&H80000000 ~ &HBFFFFFFF, &H40000000 ~ &H7FFFFFFF) 的值将出错。
- 配置数据时, 低位端口编号的数据被配置至低位地址。
例如, SOW (2) =&H2345, SOW (3) =&H0001 时, SOD (2) =&H00012345。

通过输入输出变量指定比特时，请使用以下任何一种方法。

1. 单独比特

只指定任意一个 1 比特时，请在括号内记述想要指定的端口编号及比特编号。
也可将端口编号记述在括号外面。

记述示例：DOm (b) DOm (b)

示例：DO (25) 指定端口 2 的 5 比特
 DO2 (5)

2. 同一端口的多个比特

想要统一指定同一端口的多个比特时，请在端口编号后面的括号内用逗号分隔比特编号进行记述。
也可在括号内记述端口编号。

记述示例：DOm (b,b, ... ,b) DO (mb, mb, ... ,mb)

示例：指定 DO2 (7, 5, 3) DO (27), DO (25), DO (23)
 DO (27, 25, 23)

3. 不同端口的多个比特

想要统一指定不同端口的多个比特时，请在括号内用逗号分隔端口编号与比特的两位数进行记述。

记述示例：DO (mb,mb, ... ,mb)

示例：指定 DO (37, 25, 20) DO (37), DO (25), DO (20)

4. 端口内的所有比特

指定一个端口内的所有比特时，请在端口编号后面的 () 中记述。
此外，也可通过上述 2. 或 3. 的方法进行记述。

记述示例：DOm ()

示例：DO2 () 指定 DO (27) ~ DO (20) 的所有端口
 → DO (27, 26, 25, 24, 23, 22, 21, 20) 或
 DO2 (7, 6, 5, 4, 3, 2, 1, 0) 也是一样

11 变量的有效范围

变量分为以下几类。

11.1 动态变量的有效范围

动态变量根据所声明的程序位置不同，可分为全局变量和局部变量，各自的有效范围也有所不同。

变量的类型	说明
全局变量	在子过程以外（SUB 语句～ENDSUB 语句中包含的区域以外）所声明的变量。程序整体有效。
局部变量	在子过程内所声明的变量。仅在此子过程内有效。

11.2 静态变量的有效范围

静态变量是通过执行程序复位无法清除数据的变量。

在任何程序中均可更改及参照变量的数据。

变量名如下规定，不可用于任意变量名。

变量类型	变量名
整数型	SGIn (n: 0 ~ 31)
实数型	SGRn (n: 0 ~ 31)

11.3 动态数组的有效范围

动态变量根据所声明的程序位置不同，可分为全局变量和局部变量。

变量的类型	说明
全局变量	在子过程以外（SUB 语句～ENDSUB 语句中包含的区域以外）所声明的变量。程序整体有效。
局部变量	在子过程内所声明的变量。仅在此子过程内有效。



MEMO

- 有关数组的详细说明，请参阅第 3 章“5 数组”。
- 请勿将在程序级别声明的变量作为临时参数提交，若要从子过程中参照，请使用 SHARED 语句（详细说明，请参阅第 7 章“96 SHARED”）。

12.1 动态变量的清除

以下任意情况下，数值型变量将被清零，字符型变量将被清除为空字符串。变量的数组也将同样被清除。

- 进行了程序编辑时。
- 切换了程序时(包括执行了 SWI 命令时)。
- 执行了程序复位时。
- 自动模式下，在程序停止时将专用输入信号 DI15(程序复位输入)从 OFF 设置为 ON 时。
- 在初始处理操作中对以下任意存储器进行了初始处理时。
 1. 程序存储器
 2. 所有存储器
- 执行了以下联机命令时。
 - @RESET、@INIT PGM、@INIT MEM、@INIT ALL、@SWI
- 在程序中执行了 HALT 语句时。

12.2 静态变量的清除

在以下任意情况下，整数型和实数型均将被清零。

- 在初始处理操作中对以下存储器进行了初始处理时。
 - 所有存储器
- 执行了以下联机命令时
 - @INIT MEM、@INIT ALL



MEMO

- 即使执行了程序编辑，静态变量的值也不会被清除。

第 4 章

表达式与运算

1	算术运算.....	4-1
2	字符串运算.....	4-4
3	坐标点数据的类型	4-5
4	DI / DO 条件表达式	4-6

1.1 算术运算符

运算符	记述示例	含义
+	A+B	A 加 B
-	A-B	A 减 B
*	A*B	A 乘以 B
/	A / B	A 除以 B
^	A^B	A 的 B 次方 (乘幂)
-	-A	使 A 的符号反转
MOD	A MOD B	A 除以 B 的余数 (剩余)

当剩余的运算中处理的数值为实数时，小数点以下将被舍去并转换为整数后再执行运算。运算结果为整数除法的余数。

示例： $A=15 \text{ MOD } 2 \rightarrow A=1 (15/2=7\dots 1)$
 $A=17.34 \text{ MOD } 5.98 \rightarrow A=2 (17/5=3\dots 2)$

1.2 关系运算符

关系运算符用于比较 2 个值。比较结果为真，则为“-1”，比较结果为“假”，则为“0”。

运算符	记述示例	含义
=	A = B	A 等于 B，则为“-1”，否则为“0”
<>, ><	A <> B	A 不等于 B，则为“-1”，否则为“0”
<	A < B	A 小于 B，则为“-1”，否则为“0”
>	A > B	A 大于 B，则为“-1”，否则为“0”
<=, =<	A <= B	A 小于等于 B，则为“-1”，否则为“0”
>=, =>	A >= B	A 大于等于 B，则为“-1”，否则为“0”

示例： $A=10 > 5 \rightarrow 10 > 5$ 为“真”，所以 $A=-1$

MEMO

· 对实数变量或实数数组使用等价关系运算符时，可能会由于数值修约误差等原因无法获得想要的结果。

示例：..... A=2
 B=SQR (A!)
 IF A!=B!*B! THEN ···
 → 此时，A! 不等于 B! * B!。

1.3 逻辑运算符

逻辑运算符用于对 1 个或 2 个值的每个比特分别进行操作。例如，可用于 I/O 端口输入输出状态的操作。

- 逻辑运算的结果在各个比特上用 0 或 1 表示
- 对实数数据的逻辑运算需要将其转换为整数后执行

运算符	功能	含义
NOT, ~	否定	使各个比特反转
AND、&	逻辑积	两个比特均为 1 时，逻辑积为 1
OR,	逻辑和	任意一个比特为 1 时，逻辑和为 1
XOR	排他逻辑和	两个比特不同时，排他逻辑和为 1
EQV	逻辑等价运算	双方的比特相等时结果为 1
IMP	逻辑包含运算	第 1 个比特为 1、第 2 个比特为 0 时结果为 0

示例：A%=NOT 13.05 → 对 A% 赋予“-14”（四舍五入为 13 后反转）

比特	7	6	5	4	3	2	1	0
13	0	0	0	0	1	1	0	1
NOT 13 = -14	1	1	1	1	0	0	1	0

示例：A%=3 AND 10 → 对 A% 赋予“2”

比特	7	6	5	4	3	2	1	0
3	0	0	0	0	0	0	1	1
10	0	0	0	0	1	0	1	0
3 AND 10 = 2	0	0	0	0	0	0	1	0

示例：A%=3 OR 10 → 对 A% 赋予“11”

比特	7	6	5	4	3	2	1	0
3	0	0	0	0	0	0	1	1
10	0	0	0	0	1	0	1	0
3 OR 10 = 11	0	0	0	0	1	0	1	1

示例：A%=3 XOR 10 → 对 A% 赋予“9”

比特	7	6	5	4	3	2	1	0
3	0	0	0	0	0	0	1	1
10	0	0	0	0	1	0	1	0
3 XOR 10 = 9	0	0	0	0	1	0	0	1

1.4 运算的优先级

按照以下优先级进行运算。当优先级相同时，将从写在该表达式左侧的运算开始执行。

优先级	运算内容
1	() 中的表达式
2	函数、变量
3	^ (指数)
4	单独的 +, - (单项运算符)
5	*, /
6	MOD
7	+, -
8	关系运算符
9	NOT, ~ (否定)
10	AND, & (逻辑积)
11	OR, , XOR (逻辑和、排他逻辑和)

1.5 类型转换

执行表达式中不同类型数据之间的运算时，数据类型将被转换。

1. 如果对整数赋予实数，其将被四舍五入。

示例： $A \% = 125.67 \rightarrow A \% = 126$

2. 如果用实数对整数进行四则运算，则结果为实数。

示例： $A (0) = 125 * 0.25 \rightarrow A (0) = 31.25$

3. 如果用整数除以整数，则余数将被舍去，结果为整数。

示例： $A (0) = 100 / 3 \rightarrow A (0) = 33$

2.1 字符串的联结

字符串可通过运算符 "+" 联结。

SAMPLE

```
A$="YAMAHA"  
B$="ROBOT"  
C$="LANGUAGE"  
D$="MOUNTER"  
E$=A$ + " " + B$ + " " + C$  
F$=A$ + " " + D$  
PRINT E$  
PRINT F$
```

执行结果： YAMAHA ROBOT LANGUAGE
YAMAHA MOUNTER

2.2 字符串的比较

字符也可使用与数值比较呈完全相同的关系运算符予以比较。字符串的比较用于查看字符内容及按照字母顺序排列字符(排序)时使用。

- 从每一列起始开始对每个字符逐一进行比较
- 当字符串之间完全相同时，2个字符串相等
- 只要有1个地方不同，该字符的字符编码较大的字符串将变大
- 当字符串长度不同时，较长的字符串将变大

以下示例均为“真”。

```
示例： "AA" < "AB"  
"X&" > "X#"  
"DESK" < "DESKS"
```




要点

- 关节坐标类型、正交坐标类型的数据格式均是 1 ~ 6 轴通用的。
- + 可以省略。
- 除 YK500TW 之外的机器人，第 1 机械臂转速信息、第 2 机械臂转速信息无效。

坐标点数据分为关节坐标类型与正交坐标类型 2 种。

坐标点编号范围为 0 ~ 29999。

坐标类型	数据格式	说明
关节坐标类型	±nnnnnnn	带符号的 7 位数以下的十进制常量。 可在 -6,144,000 ~ 6,144,000 的范围内指定。 单位：[脉冲]
正交坐标类型	±nnn.nn ~ ±nnnnnnn	小数部分为 3 位数以下，合计位数为 7 位数以下的带符号十进制小数。 单位：[mm] 或 [度]

在水平多关节型机器人上设置扩展设置的手系统标志时，请在数据的末尾设置 1 和 2。1 及 2 以外的数值或未指定数值时，将视作无手系统标志的设定，被设置为 0。

手系统	数据值
右手系统	1
左手系统	2

YK500TW 机器人的第 1 机械臂及第 2 机械臂的动作范围扩展到 360 度以上。

(第 1 机械臂及第 2 机械臂的可动作范围均为 -225 度 ~ +225 度)

因此，将正交坐标数据(单位:毫米)转换为关节坐标数据(单位:脉冲)时，存在多解，无法确定位置。为了在转换为关节坐标时能够正确地确定机器人的位置和机械臂的姿势，我们在以毫米为单位的坐标点数据的扩展设置的手系统标志之后，追加了第 1 机械臂转速信息和第 2 机械臂转速信息。根据设定的第 1 机械臂转速信息和第 2 机械臂转速信息，将正交坐标数据(单位:毫米)转换为关节坐标数据(单位:脉冲)。

在 YK500TW 机器人中设定扩展设置的第 1 机械臂转速信息和第 2 机械臂转速信息时，在手系统标志的结尾设置 -1、0、1 的任何一个值。如果指定了除 -1、0、1 之外的其它数值或者没有指定数值，则设定为 0。

机械臂转速信息	数据值
毫米→脉冲转换后的角度数据 x (*1) 的范围是 $-180^\circ < x \leq 180^\circ$	0
毫米→脉冲转换后的角度数据 x (*1) 的范围是 $180^\circ < x \leq 540^\circ$	1
毫米→脉冲转换后的角度数据 x (*1) 的范围是 $-540^\circ < x \leq -180^\circ$	-1

*1：是将转换为关节坐标后的脉冲数据，转换为各机械臂离机械原点的角度后的数据。

在 MOVE 语句的 STOPON 选项与 WAIT 语句的条件设置中使用 DI / DO 条件表达式。
在 DI / DO 条件式中可使用的常量、变量及运算符如下。

- 常量
 - 十进制常量、二进制常量、十六进制常量
- 变量
 - 全局整数型变量、全局实数型变量、输入输出型变量
- 运算符
 - 关系运算符、逻辑运算符
- 运算的优先级
 1. 关系运算符
 2. NOT, ~
 3. AND, &
 4. OR, |, XOR

示例：WAIT DI (31) =1 OR DI (34) =1
→ 等待 DI31 或 DI34 中的任意一个执行 (ON) 为止

第 5 章

多台机器人控制

1	概要	5-1
2	机器人指定命令一览表.....	5-2

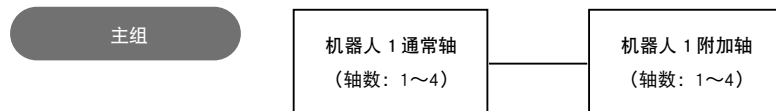
RCX340 控制器可对多台(最多 4 台)机器人进行控制。
此外, 还可使用多任务功能使数台机器人非同步动作。
若要使用此功能, 必须在出厂时在系统生成中进行多台机器人或附加轴的设置。
机器人的轴可设置如下。

- 机器人 1 (4 轴)
- 机器人 1 (1 轴) + 机器人 2 (1 轴) + 机器人 3 (1 轴) + 机器人 4 (1 轴)
- 机器人 1 (6 轴) + 机器人 2 (2 轴) (使用了 YC-LINK/E 选件时)
- 机器人 1 (4 轴) + 机器人 2 (4 轴) (使用了 YC-LINK/E 选件时)
- 机器人 1 (2 轴) + 机器人 2 (2 轴)
- 机器人 1 (4 轴) + 机器人 2 (4 轴) + 机器人 3 (4 轴) + 机器人 4 (4 轴) (使用了 YC-LINK/E 选件时)

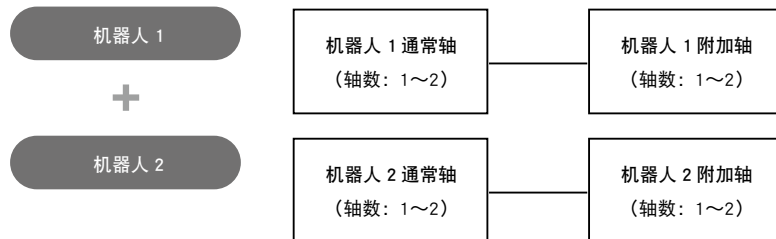
此外, 各机器人由通常轴和附加轴构成。
采用 1 台机器人、不使用附加轴时, 将变为仅限通常轴的设置。

轴构成

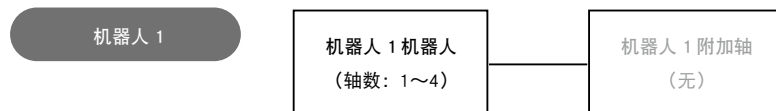
1. 机器人 1 时



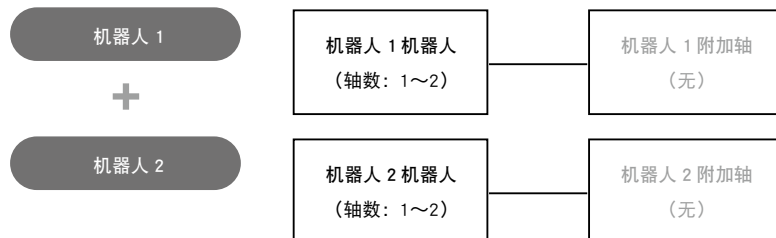
2. 机器人 1 + 机器人 2 时



3. 1 台机器人不使用附加轴时



4. 采用 2 台机器人、不设置附加轴时



35501-R9-00

机器人动作、坐标控制等机器人专用的命令对所有的机器人通用。机器人的指定可通过命令选项进行指定。主要命令如下。

运算符	命令名	
机器人动作	DRIVE MOVE PMOVE WAIT ARM	DRIVEI MOVEI SERVO
坐标控制	CHANGE LEFTY SHIFT	HAND RIGHTY
状态更改	ACCEL ASPEED DECEL OUTPOS TOLE	ARCHP1 AXWEIGHT ORGORD SPEED WEIGHT
坐标点运算	JTOXY XYTOJ	WHERE WHRXY
参数更改	ACCEL AXWEIGHT ORGORD TOLE	ARCHP1 DECEL OUTPOS WEIGHT
状态参照	ABSRPOS ARMSEL	ARMCND MCHREF
扭矩控制	DRIVE（带有扭矩限制指定选项时） TORQUE TRQTIME	
		TRQSTS CURTRQ

- 附加轴所设定的轴使用 MOVE、MOVEI、PMOVE 命令不会动作。应使用 DRIVE、DRIVEI 命令。

第 6 章

多任务

1	概要	6-1
2	任务的定义	6-1
3	任务的状态与变化	6-2
4	多任务程序示例	6-8
5	数据的共享	6-8
6	注意事项	6-9

多任务功能是一种对多个处理同时进行并行处理的功能，用于创建更复杂的高级程序。使用多任务功能时，请务必充分理解此项的内容后再使用。

多任务是一种并行执行多个任务(工作)的功能。但是，由于执行任务的 CPU 只有 1 个，并非同时执行任务。通过分享 CPU 的占用时间或对任务添加优先级提高了执行效率，对多个任务进行并行处理。

- 最多可执行任务 1 ~ 任务 16 的 16 个任务
- 可对任务添加优先级，并优先执行优先级较高的任务
- 优先顺序可以在 17 ~ 47 之间自由设定。
- 数字越小，优先级越高，数字越大，优先级越低(高：17 ⇔ 47：低)

所谓“任务”，是指一系列工作的汇总，其定义如下所述。

1. 编制 1 个程序对希望作为任务加以定义的命令块进行记述。
2. 在主任务程序的 START 语句中指定上述编制的程序，分配任务编号后进行启动。

SAMPLE

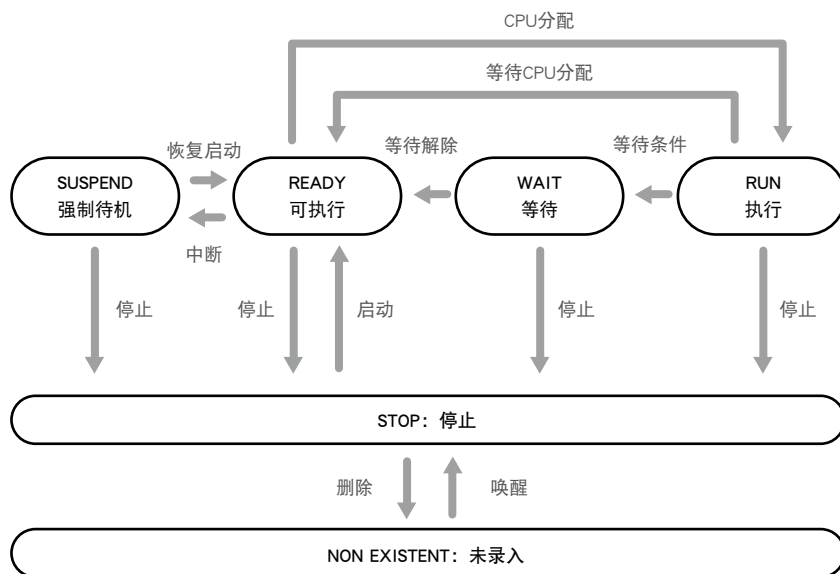
```
' MAIN TASK (TASK1)
START *IOTASK, T2..... 将 *IOTASK 作为任务 2 启动
*ST1:
MOVE P, P1, P0
  IF DI (20) = 1 THEN
    HALT
  ENDIF
GOTO *ST
HALT
程序名称 :SUB_PGM
' SUB TASK (TASK2)
*IOTASK: ..... 任务 2 从此处开始。
  IF DI (21) =1 THEN
    DO (30) =1
  ELSE
    DO (30) =0
  ENDIF
GOTO *SUBPGM ..... 任务 2 的处理至此处为止
EXIT TASK
```

3 任务的状态与变化

任务的状态有以下 6 种类型。

1. **STOP 状态 (停止状态)**
虽然存在任务，但是仍处于未执行任务的状态。
2. **RUN 状态 (执行状态)**
存在任务，且正在执行 CPU 分配的任务处理。
3. **READY 状态 (可执行状态)**
存在任务，且正在等待 CPU 分配任务处理的状态。
4. **WAIT 状态 (等待状态)**
存在任务，且任务处理因某种现象正在等待的状态。
5. **SUSPEND 状态 (强制待机状态)**
存在任务，且任务处理正在强制待机的状态。
6. **NON EXISTENT 状态 (未录入状态)**
不存在任务的状态 (唤醒需要使用 START 命令)。

任务的状态变化



35601-R9-00

3.1 任务的启动

在自动模式中程序被执行后，将自动分配至任务 1 (主任务)，并变为 RUN 状态。因此，无法对任务 1 执行删除、强制待机、强制结束等命令。

只执行程序无法唤醒其他任务 (2 ~ 8)。在任务 1 中使用 START 命令后，唤醒后将启动并变为 READY 状态。



- 无法对任务 1 执行 RESTART、SUSPEND、EXIT TASK、CUT 命令。

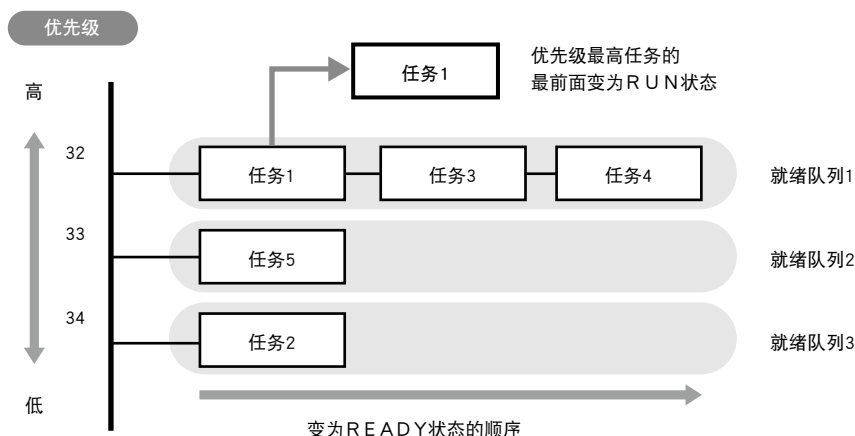
任务的排程, 指的是将 READY (可执行) 状态的任务分配至 CPU 并决定执行的顺序。

当 READY 状态的任务为 2 个以上时, 将被分配至称为就绪队列 (Ready Queue) 的 CPU 分配等待队列, 并确定任务的优先级。然后, 从多个处于 READY 状态的任务中选择 1 个任务, 使其变为 RUN (执行) 状态。

在 1 个就绪队列中只分配相同优先级的任务。因此, 存在多个优先级任务时, 将仅优先级的数字被创建就绪队列。

在同一个就绪队列中, 被采用 FCFS (First Come First Service: 先执行首先变为 READY 状态的任务) 方式管理任务。任务优先级的数字越小, 优先度越高。

任务的排程

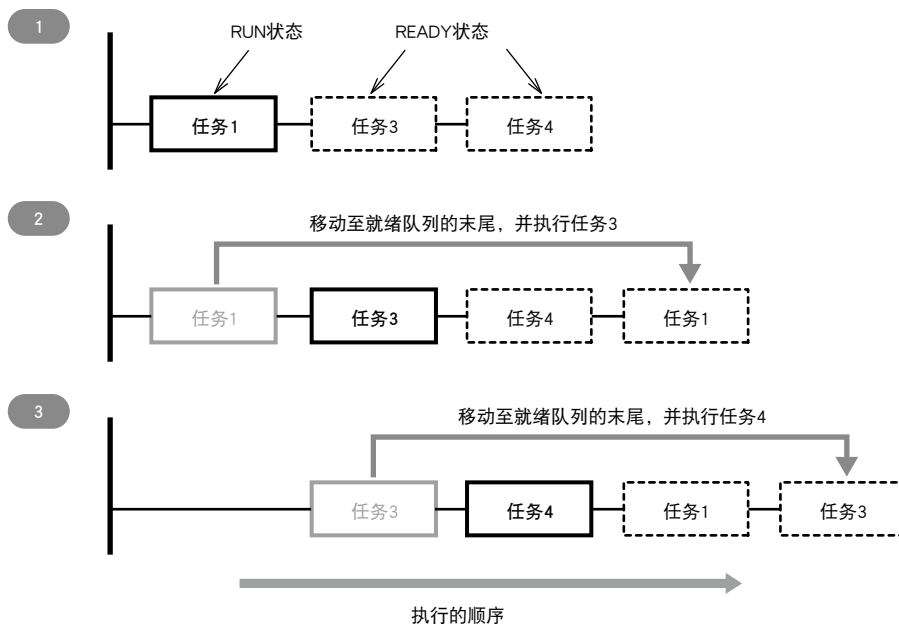


35602-R7-00

在以下任意情况下, RUN 状态的任务将变为 READY 状态, 并移动至就绪队列的末尾。

- 1) 执行了变为 WAIT 状态的命令时
- 2) CPU 占用时间经过了一定时间时
- 3) 比自己优先级更高的任务变为 READY 状态时

就绪队列



35603-R7-00



要点

- 如果 CPU 占用时间经过了一定时间, 将结束处理中的命令并移动至下一个任务。但是, 在相同优先度 (就绪队列) 以上的任务中没有 READY 状态的其他任务时, 将再次执行相同的任务。

3.3 任务的等待条件

任务执行了变为 WAIT 状态的命令后，将变为 WAIT 状态（因某种现象的等待状态）。此时，如果不解除等待条件，将无法变为 READY 状态。

1. 执行变为 WAIT 状态的命令后，将发生以下状态变化。

- 执行了变为 WAIT 状态命令后的任务 → WAIT 状态
- 位于优先级较高就绪队列最前面的任务 → RUN 状态

MEMO

- 例如，如果执行了 MOVE 语句（变为 WAIT 状态的命令之一），CPU 对驱动器发出“移动”的指示后，驱动器将等待“移动完成”的应答。这是等待的状态。此时，执行了 MOVE 语句的任务将变为 WAIT 状态，并移动至就绪队列的末尾，下一个任务将变为 RUN 状态。



要点

- 多个任务为同一个等待条件的等待现象，或者同时发生了不同条件的现象时，发生了等待条件现象的任务将全部变为 READY 状态。

2. 引起对 WAIT 状态的等待条件时，将只执行排程，并发生以下状态变化。

- 发生了 WAIT 状态等待条件的任务 → READY 状态

但是，当变为 READY 状态的任务位于最高级的优先级并在就绪队列的最前面时，将发生以下状态变化。

- 1) 当前 RUN 状态的任务 → READY 状态
- 2) 在优先级较高就绪队列最前面的任务 → RUN 状态

MEMO

- 如果在上述 MOVE 语句的示例中，任务移动至就绪队列的最末尾后收到“移动完成”的应答，则该任务将变为 READY 状态。

变为 WAIT 状态的命令如下。

现象		命令			
等待轴移动完成	轴移动命令	MOVE PMOVE	MOVEI SERVO	DRIVE WAIT ARM	DRIVEI
	参数命令	ACCEL DECEL WEIGHT	ARCHP1 OUTPOS	ARCHP2 TOLE	AXWEIGHT ORGORD
	机器人状态更改命令	CHANGE ASPEED	SHIFT SPEED	LEFTY	RIGHTY
等待时间经过		DELAY, SET (指定了时间)、WAIT ARM (指定了时间)			
等待条件成立		WAIT			
等待收发信号		SEND			
等待显示缓冲区空闲		PRINT			
等待键入		INPUT			

MEMO

- 开始执行上述命令的时候现象成立时，将不变为 WAIT 状态。

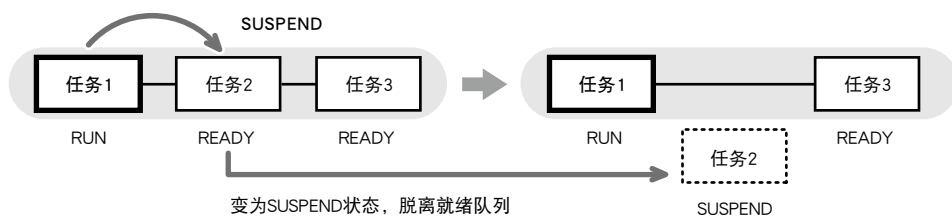
3.4 任务的中断 (SUSPEND)

可通过使用 SUSPEND 命令中断其他任务并使其变为 SUSPEND (强制待机) 状态。但是，无法对任务 1 执行 SUSPEND 命令。

执行 SUSPEND 命令后，将发生以下状态变化。

- 执行了 SUSPEND 命令的任务 → RUN 状态
- 被指定的任务 → SUSPEND 状态

任务的中断 (SUSPEND)



35604-R7-00

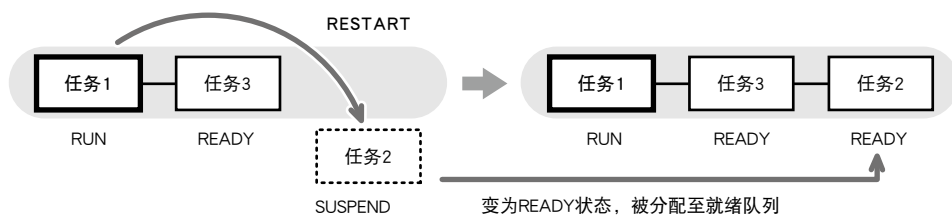
3.5 任务的恢复启动 (RESTART)

通过使用 RESTART 命令可恢复启动 SUSPEND (强制待机) 状态的任务。但是，无法对任务 1 执行 RESTART 命令。

执行 RESTART 命令后，将发生以下状态变化。

- 执行了 RESTART 命令的任务 → RUN 状态
- 被指定的任务 → READY 状态

任务的恢复启动 (RESTART)



35605-R7-00

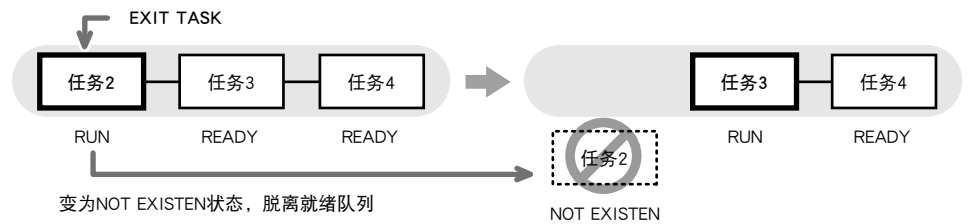
当前任务的删除 (EXIT TASK)

使用 EXIT TASK 命令可删除当前任务并使其变为 NON EXISTEN (未录入) 状态。但是, 无法对任务 1 执行 EXIT TASK 命令。

执行 EXIT TASK 命令后, 将发生以下状态变化。

- 执行了 EXIT TASK 命令的任务 → NON EXISTEN 状态
- 位于优先级较高就绪队列最前面的任务 → RUN 状态

当前任务的删除 (EXIT TASK)



35606-R7-00

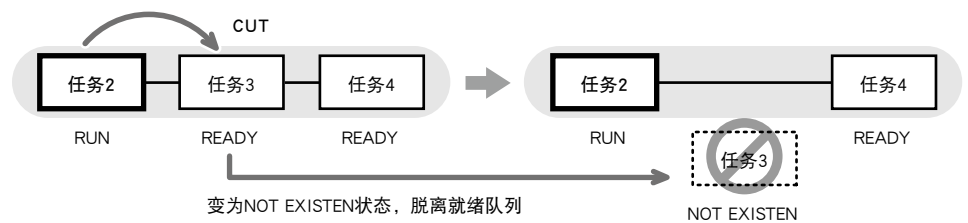
其他任务的删除 (CUT)

使用 CUT 命令可删除其他任务并使其变为 NON EXISTEN (未录入) 状态。但是, 无法对任务 1 执行 CUT 命令。

执行 CUT 命令后, 将发生以下状态变化。

- 执行了 CUT 命令的任务 → RUN 状态
- 指定的任务 → NON EXISTEN 状态

其他任务的删除 (CUT)



35607-R7-00

MEMO

- 对 WAIT 状态的任务执行 SUSPEND 命令时, 指定的任务将结束正在执行的命令。
- 无法对任务 1 执行上述任意 1 个命令。

3.7 任务的停止

由于以下原因，所有任务均变为 STOP (停止) 状态。

- 1. 执行了 HALT 命令 (停止、复位)**
程序将被复位，任务 1 以外的任务变为 NON EXISTEN 状态。任务 1 变为 STOP 状态。
- 2. 执行了 HOLD 命令 (暂时停止)**
所有任务将变为 STOP 状态。再次启动程序时，STOP 状态的任务将变为 READY 状态或 SUSPEND 状态。
- 3. 按了编程装置的 STOP 键或者联锁信号被切断**
与执行了 HOLD 命令时一样，所有任务均变为 STOP 状态。再次启动程序时，STOP 状态的任务将变为 READY 状态 (或者 READY 状态后变为 SUSPEND 状态)。
- 4. 按下了编程装置的紧急停止按钮或者紧急停止信号被切断**
所有任务变为 STOP 状态。此时，对机器人的供电将被切断，变为失去伺服保持力的状态。解除紧急停止状态并再次启动程序时，STOP 状态的任务将变为 READY 状态或 SUSPEND 状态。但是，若要恢复对机器人的供电，必须进行伺服使能处理。



MEMO

- 由于上述 1. 以外的原因造成任务停止后，在不进行程序复位启动程序时，各任务将从任务停止状态开始进行处理。上述处理与切断及接通控制器电源时一样。

各任务按照排程执行处理。以下所示为多任务程序的示例。

SAMPLE

```
' TASK1
START *ST2,T2
START *ST3,T3
*ST1:
  DO (20) = 1
  WAIT MO (20) = 1
  MOVE P, P1, P2, Z=0
  IF MO(21) =1 THEN *FIN
GOTO *ST1
*FIN:
CUT T2
HALT
' TASK2
*ST2: ..... 从此处开始为任务 2
  IF DI (20) = 1
    MO (20) = 1
    DELAY 100
  ELSE
    MO (20) = 0
  ENDIF
GOTO *ST2
EXIT TASK ..... 任务 2 至此为止
' TASK3 ..... 从此处开始为任务 3
*ST3:
  IF DI (21) = 0 THEN *ST3
  IF DI (30) = 0 THEN *ST3
  IF DI (33) = 0 THEN *ST3
  MO(21) = 1
EXIT TASK ..... 任务 3 至此为止
```

所有全局变量、其他变量、坐标点数据、位移坐标定义数据、机械手定义数据及托盘定义数据在所有任务中均是共享的。

因此，可在多个任务中使用相同的变量或数据、控制各任务的执行。



MEMO

· 多个任务共享相同的变量或数据且互换时，在各任务的处理中可能会发生故障。因此，在共享相同变量或数据时，请特别注意。

当交互、连续地通过主任务执行子任务启动处理 (START 命令) 以及子任务的任务结束处理 (EXIT TASK 命令) 时, 可能会发生无报警停止现象。

这是因为当主任务和子任务的优先顺序相同时, 在执行子任务结束处理的过程中会发生任务向主任务进行迁移, 主任务欲执行子任务启动处理的现象, 从而导致任务出现不正确状态。

因此, 当希望执行这种程序时, 请将子任务的优先顺序设定为高于主任务, 避免在执行 EXIT TASK 命令的过程中发生任务迁移现象。

在以下样本程序, 将任务 1 的优先顺序 (主任务) 设定为 32, 将任务 2 的设定为 31。(数值越小优先顺序越高)

SAMPLE

```
FLAG1 = 0
*MAIN_TASK:
  IF FLAG1=0 THEN
    FLAG1 = 1
    START *TASK2,T2,31..... 以优先顺序 31 启动任务 2( *TASK2) 。
  ENDIF
GOTO *MAIN_TASK
'=====
'      TASK2
'=====
*TASK2:
  DRIVE(1,P1)
  WAIT ARM(1)
  DRIVE(1,P2)
  WAIT ARM(1)
  FLAG1 = 0
EXIT TASK
HALT
```


第 7 章

机器人语言一览表

机器人语言一览表的查看方法	7-1
按字母顺序	7-2
按功能	7-6
函数：按字母顺序	7-11
函数：按功能	7-13
1 ABS	7-15
2 ABSRPOS	7-16
3 ACCEL	7-17
4 ARCHP1 / ARCHP2	7-18
5 ARMCND	7-20
6 ARMSEL	7-21
7 ARMTYP	7-22
8 ASPEED	7-23
9 ATN / ATN2	7-24
10 AXWGHT	7-25
11 CALL	7-26
12 CHANGE	7-27
13 CHGPRI	7-28
14 CHR\$	7-29
15 COS	7-30

16	CURTQST	7-31
17	CURTRQ.....	7-32
18	CUT.....	7-33
19	DATE\$.....	7-34
20	DECEL.....	7-35
21	DEF FN.....	7-36
22	DEGRAD	7-37
23	DELAY	7-38
24	DI.....	7-39
25	DIM.....	7-40
26	DIST.....	7-41
27	DO	7-42
28	DRIVE	7-43
29	DRIVEI.....	7-49
30	END SELECT	7-54
31	END SUB.....	7-55
32	ERR / ERL	7-56
33	EXIT FOR.....	7-57
34	EXIT SUB	7-58
35	EXIT TASK	7-59
36	FOR ~ NEXT.....	7-60
37	GOSUB ~ RETURN.....	7-61
38	GOTO	7-62
39	HALT.....	7-63
40	HALTALL.....	7-64
41	HAND.....	7-65
42	HOLD.....	7-70
43	HOLDALL	7-71
44	IF.....	7-72
45	INPUT	7-74
46	INT	7-75
47	JTOXY.....	7-76
48	LEFT\$.....	7-77
49	LEFTY	7-78
50	LEN	7-79
51	LET.....	7-80

52	LO.....	7-83
53	LOCx.....	7-84
54	LSHIFT.....	7-85
55	MCHREF.....	7-86
56	MID\$.....	7-87
57	MO.....	7-88
58	MOTOR.....	7-89
59	MOVE.....	7-90
60	MOVEI.....	7-105
61	OFFLINE.....	7-110
62	ON ERROR GOTO.....	7-111
63	ON ~ GOSUB.....	7-112
64	ON ~ GOTO.....	7-113
65	ONLINE.....	7-114
66	ORD.....	7-115
67	ORGORD.....	7-116
68	ORIGIN.....	7-117
69	OUT.....	7-118
70	OUTPOS.....	7-119
71	PDEF.....	7-121
72	PMOVE.....	7-122
73	Pn.....	7-126
74	PPNT.....	7-128
75	PRINT.....	7-129
76	PSHFRC.....	7-130
77	PSHJGSP.....	7-131
78	PSHMTD.....	7-132
79	PSHRSLT.....	7-133
80	PSHSPD.....	7-134
81	PSHTIME.....	7-135
82	PUSH.....	7-136
83	RADDEG.....	7-141
84	REM.....	7-142
85	RESET.....	7-143
86	RESTART.....	7-144
87	RESUME.....	7-145

88	RETURN.....	7-146
89	RIGHT\$.....	7-147
90	RIGHTY.....	7-148
91	RSHIFT.....	7-149
92	SELECT CASE ~ END SELECT.....	7-150
93	SEND.....	7-151
94	SERVO.....	7-153
95	SET.....	7-154
96	SHARED.....	7-155
97	SHIFT.....	7-156
98	SIN.....	7-157
99	Sn.....	7-158
100	SO.....	7-159
101	SPEED.....	7-160
102	SQR.....	7-161
103	START.....	7-162
104	STR\$.....	7-163
105	SUB ~ END SUB.....	7-164
106	SUSPEND.....	7-166
107	SWI.....	7-167
108	TAN.....	7-168
109	TCOUNTER.....	7-169
110	TIME\$.....	7-170
111	TIMER.....	7-171
112	TO.....	7-172
113	TOLE.....	7-173
114	TORQUE.....	7-174
115	VAL.....	7-176
116	WAIT.....	7-177
117	WAIT ARM.....	7-178
118	WEIGHT.....	7-179
119	WEND.....	7-180
120	WHERE.....	7-181
121	WHILE ~ WEND.....	7-182
122	WHRXY.....	7-183
123	XYTOJ.....	7-184

机器人语言一览表的查看方法

以下为您介绍机器人语言一览表的查看方法。

	(1) No.	(2) 功能	(3) 联机	(4) 种类
DIM	25	声明数组变量	×	命令

(1) No.

表示各机器人语言详细的项目编号。

"No." 栏的示例

No.

25	DIM 声明数组变量
----	---------------

格式

DIM <数组指定> [, <数组指定> , ...]

格式

<名称> [[%] (<常量> , [<常量> [, <常量>]])
 | !
 | \$

值 <常量> 数组的角标: 0 ~ 32,767 的正整数

含义 声明直接数组变量的名称及元素数。数组的角标最多可使用3维。(使用* (逗号) 隔开可在1行内声明多个数组。

MEMO

- 数组的角标为0~指定的数值为止, 合计为<常量> + 1个。
- 根据数组各维度的大小不同, 可能会发生 [9.31: Memory full] 的错误。

SAMPLE

DIM A% (10) 定义整数型的数组变量 A% (0) ~ A% (10) (元素数11)

DIM B (2, 3, 4) 定义实数型的数组变量 B (0, 0, 0) ~ B (2, 3, 4) (元素数 60)

DIM C% (2, 2), D! (10) 定义整数型的数组 C% (0, 0) ~ C% (2, 2) 与实数型的数组 D! (0) ~ D! (10)

(2) 功能

介绍机器人语言的功能。

(3) 联机

当该项目为“○”时, 表示可在联机命令中进行使用。

当该项目为“△”时, 表示有一部分无法在联机命令中执行的操作数。

(4) 种类

用“命令”或“函数”表示机器人语言的种类。

在命令与函数中均可使用时, 表示为“命令 / 函数”。

按字母顺序

No.	命令	功能	联机	种类
A				
1	ABS	计算指定值的绝对值	-	命令
2	ABSRPOS	计算指定机器人的指定轴的机台参考值(返回原点的方式仅为标记方式时有效)	-	命令/函数
3	ACCEL	设定/获取指定机器人的加速度系数参数	○	命令/函数
4	ARCHP1	设定/获取指定机器人的圆弧位置 1 参数	○	命令/函数
4	ARCHP2	设定/获取指定机器人的圆弧位置 2 参数	○	命令/函数
5	ARMCND	获取指定机器人的当前机械臂状态	-	函数
6	ARMSEL	获取指定机器人的当前手系统设定	-	函数
7	ARMTYP	获取指定机器人的手系统设定	-	函数
8	ASPEED	设定/获取指定机器人的自动移动速度	○	命令/函数
9	ATN	计算指定值的反正切值	-	函数
9	ATN2	计算指定 X-Y 坐标的反正切值	-	函数
10	AXWGHT	设定/获取指定机器人的轴前端质量参数	○	命令/函数
C				
11	CALL	调用子过程	×	命令
12	CHANGE	对指定机器人的机械手进行切换	○	命令
13	CHGPRI	更改指定任务的优先级	○	命令
14	CHR\$	获取带有指定字符编码的字符	-	函数
15	COS	计算指定值的余弦值	-	函数
16	CURTQST	获取相对于指定轴额定扭矩的当前扭矩	-	函数
17	CURTRQ	获取指定机器人中指定轴的当前扭矩值	-	函数
18	CUT	强制结束执行中或暂停中的其他任务	○	命令
D				
19	DATE\$	通过 "yy/mm/dd" 形式的字符串计算日期	-	函数
20	DECEL	设定/获取指定机器人的减速比率参数	○	命令/函数
21	DEF FN	定义用户可使用的函数	×	命令
22	DEGRAD	将值转换为弧度(↔ RADDEG)	-	函数
23	DELAY	仅等待指定的时间(单位 ms)	×	命令
24	DI	获取并行端口的输入状态	-	函数
25	DIM	对数组变量的名称及元素数进行声明	×	命令
26	DIST	计算指定 2 个点之间的距离	-	函数
27	DO	将指定值输出至 DO 端口	○	命令
28	DRIVE	以轴为单位使指定机器人进行绝对位置移动	○	命令
28	DRIVE	(带有 T 选项时)执行轴单位的绝对移动命令	○	命令
29	DRIVEI	以轴为单位使指定机器人进行相对位置移动	○	命令
E				
30	END SELECT	结束 SELECT CASE 语句	×	命令
31	END SUB	结束子过程的定义	×	命令
32	ERL	赋予发生错误的行编号	-	函数
32	ERR	赋予发生错误时的错误代码编号	-	函数
33	EXIT FOR	强制结束 FOR 语句~NEXT 语句的循环	×	命令
34	EXIT SUB	强制结束使用 SUB ~END 定义的子过程	×	命令
35	EXIT TASK	结束当前正在执行的任务	×	命令

No.	命令	功能	联机	种类
F				
36	FOR ~ NEXT	反复执行 FOR 语句的下一个语句至 NEXT 语句，直至变为控制反复的指定值为止	×	命令
G				
37	GOSUB ~ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	×	命令
38	GOTO	无条件跳转至标签所指定的行	×	命令
H				
39	HALT	停止程序并进行复位	×	命令
40	HALTALL	停止并复位所有的程序	×	命令
41	HAND	对指定机器人的机械手进行定义	○	命令
42	HOLD	暂停程序	×	命令
43	HOLDALL	暂停所有的程序	×	命令
I				
44	IF	根据条件分支控制流程	×	命令
45	INPUT	从手持编程器中对指定的变量赋值	○	命令
46	INT	获得舍去小数点以下部分的整数	-	函数
J				
47	JTOXY	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)	-	函数
L				
48	LEFT\$	从指定的字符串左侧抽出指定位数的字符串	-	函数
49	LEFTY	将指定机器人的手系统设定为左手系统	○	命令
50	LEN	获得指定字符串的长度(字节数)	-	函数
51	LET	执行指定的赋值语句	○	命令
52	LO	将被指定的值输出至 LO 端口，并进行轴移动的禁止或解除	○	命令
53	LOCx	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据	-	命令 / 函数
54	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	函数
M				
55	MCHREF	计算指定机器人轴的原点归复动作及绝对值搜索动作的机台参考	-	函数
56	MID\$	从指定的字符串中抽出任意长度的字符串	-	函数
57	MO	将指定值输出至 MO 端口	○	命令
58	MOTOR	控制马达电源状态	○	命令
59	MOVE	使指定机器人的所有轴进行绝对移动	○	命令
60	MOVEI	使指定机器人的所有轴进行相对移动	○	命令
O				
61	OFFLINE	将指定的通信端口设置为脱机模式	○	命令
62	ON ERROR GOTO	在不停止程序的情况下跳转至标签所示的错误处理例程，或者显示错误提示，并停止程序的执行	×	命令
63	ON ~ GOSUB	根据条件，通过 GOSUB 语句跳转至所指定的各标签的子例程，并执行子例程	×	命令
64	ON ~ GOTO	根据条件跳转至标签所指定的各行	×	命令
65	ONLINE	将指定的通信端口设置为联机模式	○	命令
66	ORD	获得指定字符串起始字符的字符编码	-	函数

No.	命令	功能	联机	种类
67	ORGORD	设定 / 获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数	○	命令 / 函数
68	ORIGIN	对增量式规格的轴执行返回原点动作	○	命令
69	OUT	开启指定输出端口的比特, 并结束命令语句	×	命令
70	OUTPOS	设定 / 获取指定机器人的 OUT 有效位置参数	○	命令 / 函数
P				
71	PDEF	定义用于执行托盘移动命令的托盘	○	命令
72	PMOVE	执行指定机器人的托盘移动命令	○	命令
73	Pn	在程序中定义坐标点	○	命令
74	PPNT	创建托盘定义编号与托盘位置编号中指定的坐标点数据	-	函数
75	PRINT	在手持编程器的画面上显示字符串	○	命令
76	PSHFRC	设定 / 获取推压力参数	○	命令 / 函数
77	PSHJGSP	设定 / 获取推压判定速度阈值参数	○	命令 / 函数
78	PSHMTD	设定 / 获取推压方式参数	○	命令 / 函数
79	PSHRSLT	获取 PUSH 语句结束时的状态	-	函数
80	PSHSPD	设定 / 获取推压移动速度参数	○	函数
81	PSHTIME	设定 / 获取推压时间参数	○	函数
82	PUSH	以轴为单位进行推压动作	○	命令
R				
83	RADDEG	将值转换为度 (↔ DEGRAD)	-	函数
84	REM	记述注释语句	×	命令
85	RESET	关闭指定输出端口的比特	○	命令
86	RESTART	重新启动暂停中的其他任务	○	命令
87	RESUME	进行错误的恢复处理后, 恢复执行程序	×	命令
88	RETURN	使 GOSUB 中的分支处理返回至 GOSUB 的下一行	×	命令
89	RIGHT\$	从指定的字符串右侧抽出指定位数的字符串	-	函数
90	RIGHTY	将指定机器人的手系统设定为右手系统	○	命令
91	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	函数
S				
92	SELECT CASE ~ END SELECT	根据条件分支控制流程	×	命令
93	SEND	转送文件	○	命令
94	SERVO	对指定机器人中指定轴或所有轴的伺服 ON / OFF 状态进行控制	○	命令
95	SET	开启指定输出端口的比特	△	命令
96	SHARED	使不提交变量便可在子过程中参照	×	命令
97	SHIFT	指定位移变量, 然后针对由该位移数据所指定的机器人位移坐标进行设定	○	命令
98	SIN	计算指定值的正弦值	-	函数
99	Sn	在程序中定义位移坐标	○	命令
100	SO	将指定值输出至 SO 端口	○	命令
101	SPEED	对指定机器人的程序移动速度进行变更	○	命令
102	SQR	计算指定值的平方根	-	函数
103	START	设定指定程序的任务编号及优先顺序, 并且启动该程序	○	命令
104	STR\$	将指定值转换为字符串 (↔ VAL)	-	函数
105	SUB ~ END SUB	定义子过程	×	命令
106	SUSPEND	暂停执行中的其他任务	×	命令
107	SWI	切换执行程序, 并在执行编译后从第 1 行开始执行	×	命令

No.	命令	功能	联机	种类
T				
108	TAN	计算指定值的正切值	-	函数
109	TCOUNTER	从 TCOUNTER 变量复位时开始，每隔 10ms 输出计数递增值	-	函数
110	TIME\$	通过 "hh:mm:ss" 形式的字符串计算当前时间	-	函数
111	TIMER	用上午 0 点开始的秒计算当前时间	-	函数
112	TO	将指定值输出至 TO 端口	○	命令
113	TOLE	设定 / 获取指定机器人的公差参数	○	命令 / 函数
114	TORQUE	设定 / 获取指定机器人中指定轴的最大扭矩指令值	○	命令 / 函数
V				
115	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)	-	函数
W				
116	WAIT	等待直至 DI / DO 条件表达式成立为止 (带超时)	×	命令
117	WAIT ARM	等待指定机器人的轴动作结束	×	命令
118	WEIGHT	设定 / 获取指定机器人的前端质量参数	○	命令 / 函数
119	WEND	结束 WHILE 语句的命令区	×	命令
120	WHERE	通过关节坐标 (脉冲) 读出指定机器人机械臂的当前位置	-	函数
121	WHILE ~ WEND	控制反复	×	命令
122	WHRXY	通过正交坐标 (mm、度) 读出指定机器人机械臂的当前位置	-	函数
X				
123	XYTOJ	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)	-	函数

按功能

程序命令

普通命令

No.	命令	功能	联机	种类
25	DIM	对数组变量的名称及元素数进行声明	×	命令
51	LET	执行指定的赋值语句	○	命令
84	REM	记述注释语句	×	命令

算术命令

No.	命令	功能	联机	种类
1	ABS	计算指定值的绝对值	-	命令
9	ATN	计算指定值的反正切值	-	函数
9	ATN2	计算指定 X-Y 坐标的反正切值	-	函数
15	COS	计算指定值的余弦值	-	函数
22	DEGRAD	将值转换为弧度 (↔ RADDEG)	-	函数
26	DIST	计算指定 2 个点之间的距离	-	函数
46	INT	获得舍去小数点以下部分的整数	-	函数
54	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	函数
83	RADDEG	将值转换为度 (↔ DEGRAD)	-	函数
91	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	函数
98	SIN	计算指定值的正弦值	-	函数
102	SQR	计算指定值的平方根	-	函数
108	TAN	计算指定值的正切值	-	函数

日期、时间

No.	命令	功能	联机	种类
19	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期	-	函数
109	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数值	-	函数
110	TIME \$	通过 "hh:mm:ss" 形式的字符串计算当前时间	-	函数
111	TIMER	用上午 0 点开始的秒计算当前时间	-	函数

字符串操作

No.	命令	功能	联机	种类
14	CHR \$	获取带有指定字符编码的字符	-	函数
48	LEFT \$	从指定的字符串左侧抽出指定位数的字符串	-	函数
50	LEN	获得指定字符串的长度(字节数)	-	函数
56	MID \$	从指定的字符串中抽出任意长度的字符串	-	函数
66	ORD	获得指定字符串起始字符的字符编码	-	函数
89	RIGHT \$	从指定的字符串右侧抽出指定位数的字符串	-	函数
104	STR \$	将指定数值转换为字符串 (↔ VAL)	-	函数
115	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)	-	函数

坐标点、坐标、位移坐标

No.	命令	功能	联机	种类
12	CHANGE	对指定机器人的机械手进行切换	○	命令
41	HAND	对指定机器人的机械手进行定义	○	命令
47	JTOXY	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)	-	函数
49	LEFTY	将指定机器人的手系统设定为左手系统	○	命令
53	LOCx	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据	-	命令 / 函数
73	Pn	在程序中定义坐标点	○	命令
74	PPNT	创建托盘定义编号与托盘位置编号中指定的坐标点数据	-	函数
90	RIGHTY	将指定机器人的手系统设定为右手系统	○	命令
99	Sn	在程序中定义位移坐标	○	命令
97	SHIFT	指定位移变量，然后针对由该位移数据所指定的机器人位移坐标进行设定	○	命令
123	XYTOJ	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)	-	函数

分支命令

No.	命令	功能	联机	种类
33	EXIT FOR	强制结束 FOR 语句~NEXT 语句的循环	×	命令
36	FOR ~ NEXT	反复执行 FOR 语句的下一个语句至 NEXT 语句，直至变为控制反复的指定值为止	×	命令
37	GOSUB ~ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	×	命令
38	GOTO	无条件跳转至标签所指定的行	×	命令
44	IF	根据条件分支控制流程	×	命令
63	ON ~ GOSUB	根据条件，通过 GOSUB 语句跳转至所指定的各标签的子例程，并执行子例程	×	命令
64	ON ~ GOTO	根据条件跳转至标签所指定的各行	×	命令
92	SELECT CASE ~ END SELECT	根据条件分支控制流程	×	命令
121	WHILE ~ WEND	控制反复	×	命令

错误控制

No.	命令	功能	联机	种类
62	ON ERROR GOTO	在不停止程序的情况下跳转至标签所示的错误处理例程，或者显示错误提示，并停止程序的执行	×	命令
87	RESUME	进行错误的恢复处理后，恢复执行程序	×	命令
32	ERL	赋予发生错误的行编号	-	函数
32	ERR	赋予发生错误时的错误代码编号	-	函数

程序、任务控制

程序控制

No.	命令	功能	联机	种类
11	CALL	执行其他程序	×	命令
39	HALT	停止程序并进行复位	×	命令
40	HALTALL	停止所有程序，并且对任务 1 进行复位，其它任务结束	×	命令
42	HOLD	暂停程序	×	命令
43	HOLDALL	暂停所有程序	×	命令
107	SWI	切换执行程序，并在执行编译后从第 1 行开始执行	×	命令

任务控制

No.	命令	功能	联机	种类
13	CHGPRI	更改指定任务的优先级	-	命令
18	CUT	强制结束执行中或暂停中的其他任务	○	命令
35	EXIT TASK	结束当前正在执行的任务	×	命令
82	PUSH	以轴为单位进行推压动作	○	命令
86	RESTART	重新启动暂停中的其他任务	○	命令
103	START	设置指定任务的任务编号及优先级，并启动该任务	○	命令
106	SUSPEND	暂停执行中的其他任务	×	命令

机器人控制

机器人动作

No.	命令	功能	联机	种类
12	CHANGE	对指定机器人的机械手进行切换	○	命令
28	DRIVE	以轴为单位使指定机器人进行绝对位置移动	○	命令
29	DRIVEI	以轴为单位使指定机器人进行相对位置移动	○	命令
41	HAND	对指定机器人的机械手进行定义	○	命令
49	LEFTY	将指定机器人的手系统设定为左手系统	○	命令
58	MOTOR	控制马达电源状态	○	命令
59	MOVE	使指定机器人的所有轴进行绝对移动	○	命令
60	MOVEI	使指定机器人的所有轴进行相对移动	○	命令
68	ORIGIN	执行指定机器人中轴的原点归复动作	○	命令
72	PMOVE	执行指定机器人的托盘移动命令	○	命令
90	RIGHTY	将指定机器人的手系统设定为右手系统	○	命令
94	SERVO	对指定机器人中指定轴或所有轴的伺服 ON / OFF 状态进行控制	○	命令

状态获取

No.	命令	功能	联机	种类
2	ABSRPOS	计算指定机器人的指定轴的机台参考值（返回原点的方式仅为标记方式时有效）	-	命令 / 函数
5	ARMCND	获取指定机器人的当前机械臂状态	-	函数
6	ARMSEL	获取指定机器人的当前手系统设定	-	函数
7	ARMTYP	获取指定机器人的手系统设定	-	函数
16	CURTQST	获取相对于指定轴额定扭矩的当前扭矩	-	函数
55	MCHREF	计算指定机器人的原点归复动作及绝对值搜索动作的机台参考	-	函数
79	PSHRSLT	获取 PUSH 语句结束时的状态	-	函数
80	PSHSPD	设定 / 获取推压移动速度参数	○	函数
81	PSHTIME	设定 / 获取推压时间参数	○	函数
117	WAIT ARM	等待指定机器人的轴动作结束	×	命令
120	WHERE	通过关节坐标（脉冲）读出指定机器人机械臂的当前位置	-	函数
122	WHRXY	通过正交坐标（mm、度）读出指定机器人机械臂的当前位置	-	函数

状态更改

No.	命令	功能	联机	种类
3	ACCEL	设定 / 获取指定机器人的加速度系数参数	○	命令 / 函数
4	ARCHP1	设定 / 获取指定机器人的圆弧位置 1 参数	○	命令 / 函数
4	ARCHP2	设定 / 获取指定机器人的圆弧位置 2 参数	○	命令 / 函数
8	ASPEED	设定 / 获取指定机器人的自动移动速度	○	命令 / 函数
10	AXWGHT	设定 / 获取指定机器人的轴前端质量参数	○	命令 / 函数
20	DECEL	设定 / 获取指定机器人的减速比率参数	○	命令 / 函数
67	ORGORD	设定 / 获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数	○	命令 / 函数
70	OUTPOS	设定 / 获取指定机器人的 OUT 有效位置参数	○	命令 / 函数
71	PDEF	定义用于执行托盘移动命令的托盘	○	命令
76	PSHFRC	设定 / 获取推压力参数	-	命令 / 函数
77	PSHJGSP	设定 / 获取推压判定速度阈值参数	○	命令 / 函数
78	PSHMTD	设定 / 获取推压方式参数	○	命令 / 函数
101	SPEED	对指定机器人的程序移动速度进行变更	○	命令
113	TOLE	设定 / 获取指定机器人的公差参数	○	命令 / 函数
118	WEIGHT	设定 / 获取指定机器人的前端质量参数	○	命令 / 函数

输入输出、通信控制

输入输出控制

No.	命令	功能	联机	种类
23	DELAY	仅等待指定的时间(单位 ms)	×	命令
27	DO	将指定值输出至 DO 端口	○	命令
52	LO	将被指定值输出至 LO 端口, 并进行轴移动的禁止或解除	○	命令
57	MO	将指定值输出至 MO 端口	○	命令
69	OUT	开启指定输出端口的比特, 并结束命令语句	×	命令
85	RESET	关闭指定输出端口的比特	○	命令
95	SET	开启指定输出端口的比特	△	命令
100	SO	将指定值输出至 SO 端口	○	命令
112	TO	将指定值输出至 TO 端口	○	命令
116	WAIT	等待直至 DI/DO 条件表达式成立为止(带超时)	×	命令

通信控制

No.	命令	功能	联机	种类
65	ONLINE	将指定的通信端口设置为联机模式	○	命令
61	OFFLINE	将指定的通信端口设置为脱机模式	○	命令
93	SEND	转送文件	○	命令

函数：按字母顺序

No.	函数	种类	功能
A			
1	ABS	算术函数	计算指定值的绝对值
2	ABSRPOS	算术函数	计算指定机器人的指定轴的机台参考值（返回原点的方式仅为标记方式时有效）
3	ACCEL	算术函数	获取指定机器人的加速度系数参数
4	ARCHP1	算术函数	获取指定机器人的圆弧位置 1 参数
4	ARCHP2	算术函数	获取指定机器人的圆弧位置 2 参数
5	ARMCND	算术函数	获取指定机器人的当前机械臂状态
6	ARMSEL	算术函数	获取指定机器人的当前手系统设定
7	ARMTYP	算术函数	获取指定机器人的手系统设定
8	ASPEED	算术函数	设定 / 获取指定机器人的自动移动速度
9	ATN	算术函数	计算指定值的反正切值
9	ATN2	算术函数	计算指定 X-Y 坐标的反正切值
10	AXWGHT	算术函数	获取指定机器人的轴前端质量参数
C			
14	CHR\$	字符串函数	获取带有指定字符编码的字符
15	COS	算术函数	计算指定值的余弦值
16	CURTQST	算术函数	获取相对于指定轴额定扭矩的当前扭矩
17	CURTRQ	算术函数	获取指定机器人中指定轴的当前扭矩值
D			
19	DATE\$	字符串函数	通过 "yy/mm/dd" 形式的字符串计算日期
20	DECEL	算术函数	获取指定机器人的减速比率参数
22	DEGRAD	算术函数	将值转换为弧度 (↔ RADDEG)
26	DIST	算术函数	计算指定 2 个点之间的距离
E			
32	ERL	算术函数	发生错误的行编号
32	ERR	算术函数	赋予发生错误时的错误代码编号
I			
46	INT	算术函数	获得舍去小数点以下部分的整数值
J			
47	JTOXY	坐标点函数	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)
L			
48	LEFT\$	字符串函数	从指定的字符串左侧抽出指定比特数的字符串
50	LEN	算术函数	获得指定字符串的长度 (字节数)
53	LOCx	坐标点函数	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据
54	LSHIFT	算术函数	仅将指定值的比特数向左偏移 (↔ RSHIFT)
M			
55	MCHREF	算术函数	计算指定机器人的原点归复动作及绝对值搜索动作的机台参考
56	MID\$	字符串函数	从指定的字符串中抽出任意长度的字符串
O			
66	ORD	算术函数	获得指定字符串起始字符的字符编码

No.	函数	种类	功能
67	ORGORD	算术函数	获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数
70	OUTPOS	算术函数	获取指定机器人的 OUT 有效位置参数
P			
74	PPNT	坐标点函数	创建托盘定义编号与托盘位置编号中指定的坐标点数据
76	PSHFRC	算术函数	设定 / 获取推压力参数
77	PSHJGSP	算术函数	设定 / 获取推压判定速度阈值参数
78	PSHMTD	算术函数	设定 / 获取推压方式参数
79	PSHRSLT	算术函数	获取 PUSH 语句结束时的状态
80	PSHSPD	算术函数	设定 / 获取推压移动速度参数
81	PSHTIME	算术函数	设定 / 获取推压时间参数
R			
83	RADDEG	算术函数	将值转换为度 (↔ DEGRAD)
89	RIGHT\$	字符串函数	从指定的字符串右侧抽出指定比特数的字符串
91	RSHIFT	算术函数	仅将指定值的比特数向右偏移 (↔ LSHIFT)
S			
98	SIN	算术函数	计算指定值的正弦值
102	SQR	算术函数	计算指定值的平方根
104	STR\$	字符串函数	将指定值转换为字符串 (↔ VAL)
T			
108	TAN	算术函数	计算指定值的正切值
109	TCOUNTER	算术函数	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数递增值
110	TIME\$	字符串函数	通过 "hh:mm:ss" 形式的字符串计算当前时间
111	TIMER	算术函数	用上午 0 点开始的秒计算当前时间
113	TOLE	算术函数	获取指定机器人的公差参数
114	TORQUE	算术函数	获取指定机器人中指定轴的最大扭矩指令值
V			
115	VAL	算术函数	将指定字符串标注的值转换为实际的数值 (↔ STR \$)
W			
118	WEIGHT	算术函数	获取指定机器人的前端质量参数
120	WHERE	坐标点函数	通过关节坐标 (脉冲) 读出指定机器人机械臂的当前位置
122	WHRXY	坐标点函数	通过正交坐标 (mm、度) 读出指定机器人机械臂的当前位置
X			
123	XYTOJ	坐标点函数	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)

相关坐标点

No.	函数名	功能
47	JTOXY	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)
53	LOCx	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据
74	PPNT	创建托盘定义编号与托盘位置编号中指定的坐标点数据
120	WHERE	通过关节坐标 (脉冲) 读出指定机器人机械臂的当前位置
122	WHRXY	通过正交坐标 (mm、度) 读出指定机器人机械臂的当前位置
123	XYTOJ	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)

相关参数

No.	函数名	功能
2	ABSRPOS	计算指定机器人的指定轴的机台参考值 (返回原点的方式仅为标记方式时有效)
3	ACCEL	设定 / 获取指定机器人的加速度系数参数
4	ARCHP1	获取指定机器人的圆弧位置 1 参数
4	ARCHP2	获取指定机器人的圆弧位置 2 参数
5	ARMCND	获取指定机器人的当前机械臂状态
6	ARMSEL	获取指定机器人的当前手系统设定
7	ARMTYP	获取指定机器人的手系统设定
10	AXWGHT	获取指定机器人的轴前端质量参数
16	CURTQST	获取相对于指定轴额定扭矩的当前扭矩
17	CURTRQ	获取指定机器人中指定轴的当前扭矩值
20	DECEL	获取指定机器人的减速比率参数
50	LEN	获得指定字符串的长度 (字节数)
55	MCHREF	计算指定机器人轴的原点归复动作的机台参考
66	ORD	获得指定字符串起始字符的字符编码
67	ORGORD	获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数
70	OUTPOS	获取指定机器人的 OUT 有效位置参数
79	PSHRSLT	获取 PUSH 语句结束时的状态
113	TOLE	获取指定机器人的公差参数
114	TORQUE	获取指定机器人中指定轴的最大扭矩指令值
118	WEIGHT	获取指定机器人的前端质量参数

相关数值运算

No.	函数名	功能
1	ABS	计算指定值的绝对值
9	ATN	计算指定值的反正切值
9	ATN2	计算指定 X-Y 坐标的反正切值
15	COS	计算指定值的余弦值
22	DEGRAD	将值转换为弧度 (↔ RADDEG)
26	DIST	计算指定 2 个点之间的距离
46	INT	获得舍去小数点以下部分的整数值
54	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)
83	RADDEG	将值转换为度 (↔ DEGRAD)
91	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)
98	SIN	计算指定值的正弦值
102	SQR	计算指定值的平方根
108	TAN	计算指定值的正切值
115	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)

相关字符串运算

No.	函数名	功能
14	CHR \$	获取带有指定字符编码的字符
19	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期
48	LEFT \$	从指定的字符串左侧抽出指定比特数的字符串
56	MID \$	从指定的字符串中抽出任意长度的字符串
89	RIGHT \$	从指定的字符串右侧抽出指定比特数的字符串
104	STR \$	将指定值转换为字符串 (↔ VAL)

相关参数

No.	函数名	功能
32	ERL	赋予发生错误的行编号
32	ERR	赋予发生错误时的错误代码编号
109	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 1ms 输出计数递增值
110	TIME \$	通过 "hh:mm:ss" 形式的字符串计算当前时间
111	TIMER	用上午 0 点开始的秒计算当前时间

格式

ABS (<表达式>)

含义 返回 <表达式> 中指定值的绝对值。

SAMPLE

A=ABS (-326.55)..... 将 -362.54 的绝对值 (362.54) 赋予变量 A

ABSRPOS

获取机器参照

格式

ABSRPOS [< 机器人编号 >] (< 轴编号 >)

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义 获取 < 机器人编号 > 指定机器人的 < 轴编号 > 指定轴的机台参考值 (单位 : %)。 < 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。
 此函数的对象为, 返回原点方式设置为标记方式的轴。
 与传感方式、撞块方式的轴无关。

 MEMO

· 当返回原点方式设置为标记方式的轴, 机器参照值在 44 ~ 56% 的范围内时, 可进行绝对式原点复位。

SAMPLE

A=ABSRPOS (4)..... 将机器人 1 中第 4 轴的机台参考值赋给变量 A。

格式

1. ACCEL [< 机器人编号 >] < 表达式 >
2. ACCEL [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 1 ~ 100 (单位 : %)

含义 将 < 机器人编号 > 指定机器人轴的直接加速度系数参数变更为 < 表达式 > 的值。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
 形式 1 对指定机器人中设定的所有轴进行变更。
 格式 2 只更改 < 轴编号 > 中指定的轴。

MEMO

- 在生成过程中，如果指定 no axis 中所设定的轴时，则会发生“5.37：规格不一致”错误，并且中止执行。
- 对通过悬吊式操作盒等操作终端设定的数值进行改写。通过程序声明的数值享有优先权。

函数

格式

ACCEL [< 机器人编号 >] (< 轴编号 >)

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义 获取 < 机器人编号 > 指定机器人轴之中 < 轴编号 > 指定轴的加速度参数值。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

SAMPLE

```

A=50
ACCEL A ..... 将机器人 1 中所有轴的加速度系数设为 50%。
ACCEL (3) =100 ..... 仅将机器人 1 中第 3 轴设为 100%。
' CYCLE WITH INCREASING ACCELERATION
FOR A=10 TO 100 STEP 10..... 使加速度系数参数从 10% 至 100%，以 10%
                               为步长逐步变化

  ACCEL A
  MOVE P, P0
  MOVE P, P1
NEXT A
A=ACCEL (3)..... 将机器人 1 中第 3 轴的加速度系数参数赋给变量 A
HALT "END TEST"

```

格式

1. ARCHP1 [< 机器人编号 >] < 表达式 >
2. ARCHP1 [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

格式

1. ARCHP2 [< 机器人编号 >] < 表达式 >
2. ARCHP2 [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 1 ~ 6144000 (单位: 脉冲)

含义 ARCHP1 对应圆弧位置 1 参数, ARCHP2 对应圆弧位置 2 参数。将参数的圆弧位置变更为 < 表达式 > 的值。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。
 形式 1 对指定机器人中设定的所有轴进行变更。
 格式 2 将 < 轴编号 > 中指定轴的圆弧位置参数更改为 < 表达式 > 中的值。

 MEMO

· 在生成过程中, 如果指定 no axis 中所设定的轴时, 则会发生“5.37: 规格不一致”错误, 并且中止执行。

函数

格式

ARCHP1 [< 机器人编号 >] (< 轴编号 >)

格式

ARCHP2 [< 机器人编号 >] (< 轴编号 >)

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义 ARCHP1 对应圆弧位置 1 参数, ARCHP2 对应圆弧位置 2 参数。
 获取 < 轴编号 > 中指定轴的圆弧位置参数值。
 < 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

SAMPLE

```

DIM SAV (3)
DIM SAV2 (3)
GOSUB *SAVE_ARCH
FOR A=1000 TO 10000 STEP 1000
  GOSUB *CHANGE_ARCH
  MOVE P, P0, A3=0
  DO3 (0) =1 ..... 夹持器关闭
  MOVE P, P1, A3=0
  DO3 (0) =0 ..... 夹持器开启
NEXT A
GOSUB *RESTORE_ARCH
HALT
*CHANGE_ARCH:
FOR B=1 TO 4
  ARCHP1 (B) =A
  ARCHP2 (B) =A
NEXT B
RETURN
*SAVE_ARCH:
FOR B=1 TO 4
  SAV (B-1) =ARCHP1 (B) ..... 将圆弧位置参数 ARCH (1) ~ (4) 赋值给数组
  变量 SAV (0) ~ (3)

  SAV2 (B-1) =ARCHP2 (B)
NEXT B
RETURN
*RESTORE_ARCH:
FOR B=1 TO 4
  ARCHP1 (B) =SAV (B-1)
  ARCHP2 (B) =SAV2 (B-1)
NEXT B
RETURN

```

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

ARMCND [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 对水平多关节型机器人赋予当前机械臂的状态。指定 < 机器人编号 > 中获取机械臂状态的机器人。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
当机械臂状态为右手系统时为 1，左手系统时为 2。
此函数仅在使用水平多关节型机器人时有效。

SAMPLE

```
A=ARMCND..... 将机器人 1 的当前机械臂状态赋值给变量 A
IF A=1 THEN ..... 右手系统状态
    MOVE P, P100, Z=0
ELSE ..... 左手系统状态
    MOVE P, P200, Z=0
ENDIF
```

格式

```
ARMSEL [ < 机器人编号 > ] < 表达式 >
```

值 < 机器人编号 > 1 ~ 4
 < 表达式 > 1 : 右手系统、2 : 左手系统

含义 设定水平多关节型机器人的当前手系统选择。指定利用 < 机器人编号 > 设定手系统选择的机器人。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。此函数仅在使用水平多关节型机器人时有效。

SAMPLE

```
ARMSEL[2] 2 ..... 将机器人 2 的手系统选择设定为左手系统
```

函数

格式

```
ARMSEL [ < 机器人编号 > ]
```

值 < 机器人编号 > 1 ~ 4

含义 赋予水平多关节型机器人当前所选手系统。指定利用 < 机器人编号 > 获取手系统的机器人。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。当所选手系统为右手系统时为 1，左手系统时为 2。此函数仅在使用水平多关节型机器人时有效。

SAMPLE

```
A=ARMSEL..... 赋予机器人 1 的机械臂类型值
IF A=1 THEN ..... 机械臂类型为右手系统
  MOVE P, P100, Z=0
ELSE ..... 机械臂类型为左手系统
  MOVE P, P200, Z=0
ENDIF
```

格式

ARMTYP [< 机器人编号 >] < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 表达式 > 1 : 右手系统、2 : 左手系统

含义 设定水平多关节型机器人在程序复位时的手系统。指定利用 < 机器人编号 > 设定手系统选择的机器人。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。此函数仅在使用水平多关节型机器人时有效。

SAMPLE

ARMTYP[2] 2 将机器人 2 的手系统设定为左手系统

函数

格式

ARMTYP [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 赋予水平多关节型机器人在程序复位时的手系统。指定利用 < 机器人编号 > 获取手系统的机器人。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。当所选手系统为右手系统时为 1，左手系统时为 2。此函数仅在使用水平多关节型机器人时有效。

SAMPLE

```
A=ARMTYP ..... 赋予机器人 1 的机械臂类型值
IF A=1 THEN ..... 机械臂类型为右手系统
  MOVE P, P100, Z=0
ELSE ..... 机械臂类型为左手系统
  MOVE P, P200, Z=0
ENDIF
HALTALL ..... 程序复位
```



要点

- 自动移动速度
通过手持编程器的操作或
ASPEED 命令设置
- 程序移动速度
由 SPEED 命令或 MOVE、
DRIVE 的速度指定设置

格式

ASPEED [< 机器人编号 >] < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 表达式 > 1 ~ 100 (单位 : %)

含义 将 < 机器人编号 > 指定机器人的直接自动移动速度变更为 < 表达式 > 的指定值。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

机器人构成轴与附加轴的所有速度均将被更改。动作速度由手持编程器的操作或 ASPEED 命令设置的自动移动速度与 SPEED 等命令指定的程序移动速度的乘积决定。

动作速度 = 自动移动速度 × 程序移动速度

示例：

自动移动速度 80%

程序移动速度 50%

移动速度 = 40% (80% × 50%)

函数

格式

ASPEED[< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 获取 < 机器人编号 > 指定机器人的自动移动速度值。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

SAMPLE

SPEED 70

ASPEED 100

MOVE P, P0 按照 70% (=100*70) 的速度从当前位置向 P0 移动

ASPEED 50

MOVE P, P1 按照 35% (=50*70) 的速度从当前位置向 P1 移动

MOVE P, P2, S=10 按照 5% (=50*10) 的速度从当前位置向 P2 移动
HALT

相关命令 SPEED

格式

ATN (<表达式>)

格式

ATN2 (<表达式 1> , <表达式 2>)

含义 ATN：赋予<表达式>中指定值的反正切值。获得的值为 $-\pi/2$ 至 $+\pi/2$ 范围内的弧度。
 ATN2：赋予<表达式 1>和<表达式 2>中指定 X-Y 坐标的反正切值。得到的数值是位于 $-\pi$ 至 $+\pi$ 范围内的弧度。

SAMPLE

A (0) =A*ATN (Y / X) 将表达式 (Y / X) 的反正切值与变量 A 的乘积赋值给数组 A (0)
 A (0) =ATN (0.5) 将 0.5 的反正切值赋值给 A (0)
 A(0)=ATN2(B,C)-D 将 X-Y 坐标上 (B,C) 的反正切值与变量 D 的差值赋值给数组 A(0)。
 A(1)=RADDEG(ATN2(B,C))..... 将 X-Y 坐标上 (B,C) 的反正切值转换为度数并且赋值给数组 A(1)。

相关命令 COS, DEGRAD, RADDEG, SIN, TAN

格式

AXWGHT [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 根据设置的机器人而异

含义 直接将 < 机器人编号 > 指定机器人轴之中 < 轴编号 > 指定轴的轴前端质量参数变更为 < 表达式 > 的值。 < 机器人编号 > 可以省略。 当进行省略时， 机器人 1 被指定。 仅对 "MULTI" 的轴或附加轴有效 (机器人的机型及附加轴在出厂时已设置)。

函数

格式

AXWGHT [< 机器人编号 >] (< 轴编号 >)

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义 赋予 < 机器人编号 > 指定机器人轴之中 < 轴编号 > 指定轴的轴前端重量参数值。 < 机器人编号 > 可以省略。 当进行省略时， 机器人 1 被指定。 仅对机器人为 "MULTI" 的轴或附加轴有效。

SAMPLE

```
A=5
B=0
C=AXWGHT (1) ..... 获取轴前端重量的值 (将当前值保存至变量 C)
AXWGHT (1) =A
DRIVE (1, P0)
AXWGHT (1) =B
DRIVE (1, P1)
AXWGHT (1) =C ..... 重新设置轴前端重量的值
HALT
```

相关命令 WEIGHT



要点

- 在进行值传递时，即使在子过程内更改了值，实参的原始值也不会发生变化。
- 在进行引用传递时，如果在子过程内更改了值，实参的原始值也会发生变化。
- 详细说明，请参阅第3章“8 值传递与引用传递”。

MEMO

格式

CALL <标签> [(<实参> [, <实参> ...])]

含义

调用 SUB 语句~ END SUB 语句中定义的子过程。

在 <标签> 中指定与 SUB 语句所定义名称相同的名称。

1. 在实参中指定了常量或表达式时，将变为值传递。
2. 在实参中指定了变量或数组元素时，将变为值传递。但是，如果在实参前面附加 REF，将变为引用传递。
3. 在实参中指定了所有数组（数组名后面带有（））时，将变为引用传递。

- CALL 语句可以连续使用最多 120 次。但是，针对 FOR 语句、GOSUB 语句等使用堆栈的命令，根据其使用状态及识别符的使用状况等，次数可能有所减少。
- 在利用 CALL 语句调用子过程时，请务必使用 END SUB 语句或 EXIT SUB 语句来结束子过程。通过 GOTO 等语句从子例程跳转时，可能会发生 [5.12: Stack overflow] 等错误。

SAMPLE 1

```
X%=4
Y%=5
CALL *COMPARE ( REF X%,REF Y% )
HALT
' SUB ROUTINE: COMPARE
SUB *COMPARE ( A%, B% )
  IF A% < B% THEN
    TEMP%=A%
    A%=B%
    B%=TEMP%
  ENDIF
END SUB
```

SAMPLE 2

```
I = 1
CALL *TEST ( I )
HALT
' SUB ROUTINE: TEST
SUB *TEST
  X = X + 1
  IF X < 15 THEN
    CALL *TEST ( X )
  ENDIF
END SUB
```

相关命令

SUB, END SUB, CALL, EXIT SUB, SHARED

格式

CHANGE [< 机器人编号 >]	Hn
	OFF

值 < 机器人编号 > 1 ~ 4
 n : 机械手编号 0 ~ 31

含义 通过 CHANGE 进行 < 机器人编号 > 指定机器人的机械手的切换。
 指定为 OFF 时，表示无机械手设定。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
 在切换机械手之前，请利用 HAND 语句对机械手进行定义。
 详细内容请参照“41 HAND”。当指定其它机器人及已被定义的机械手数据时，将会发生“机械手数据不一致错误”。

SAMPLE

```
HAND H1=      0      150.0      0.0
HAND H2=  -5000      20.00      0.0
P1=150.00 300.00 0.00 0.00 0.00 0.00
CHANGE H2 ..... 更改为机械手 2
MOVE P, P1 ..... 机械手 2 的前端向 P1 移动 (1)
CHANGE H1 ..... 更改为机械手 1
MOVE P, P1 ..... 机械手 1 的前端向 P1 移动 (2)
HALT
```

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

CHGPRI	Tn “< “<程序名称>” >” PGm	,p
--------	-----------------------------	----

值

m : 程序编号..... 0 ~ 99
n : 任务编号..... 1 ~ 16
p : 任务优先级..... 1 ~ 64

含义 将直接指定任务 n 的优先级更改为 p。
数字越小，优先级越高，数字越大，优先级越低(高 1 ⇔ 低 64)。
当任务优先级较高的任务处于 READY 状态时，优先级较低的任务也将依旧保持 READY 状态。

SAMPLE

```
START <SUB_PGM>,T2,33
*ST:
  MOVE P, P0, P1
  IF DI (20) = 1 THEN
    CHGPRI T2, 32
  ELSE
    CHGPRI T2, 33
  ENDIF
GOTO *ST
HALT
程序名称 :SUB_PGM
*SUBPGM:
' SUBTASK ROUTINE
*SUBTASK:
  IF LOC3 (WHERE) > 10000 THEN
    DO (20) = 1
    GOTO *SUBTASK
  ENDIF
  DO (20) = 0
GOTO *SUBPGM
EXIT TASK
```

相关命令 CUT, EXIT TASK, RESTART, SUSPEND, START

格式

CHR\$ (<表达式>)

值 <表达式>0 ~ 255**含义** 返回带有指定字符编码的字符。<表达式>的值不在0 ~ 255范围内时，将出错。

SAMPLE

A\$=CHR\$(65)..... 将"A"赋值给A\$

相关命令 ORD

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

COS (<表达式>)

值 <表达式> 角度(单位: 弧度)

含义 赋予<表达式>中值的余弦值。

SAMPLE

A (0) =B*COS (C) 将变量 C42 的余弦值与变量 B 的乘积赋值给数组 A (0)

A (1) =COS (DEGRAD (20)) 将 20.0° 的余弦值赋值给数组 A (1)

相关命令 ATN, DEGRAD, RADDEG, SIN, TAN

格式

CURTQST [< 机器人编号 >] (< 轴编号 >)

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义 获取相对于 < 机器人编号 > 指定机器人的 < 轴编号 > 指定轴额定扭矩的当前扭矩值 (-1000 ~ 1000)。< 机器人编号 > 可以省略。
 当进行省略时，机器人 1 被指定。

SAMPLE

A = CURTQST(3) 将相对于机器人 1 中第 3 轴额定扭矩的当前扭矩值赋给变量 A。



要点

- 在生成过程中，如果指定 no axis 中所设定的轴时，则会发生“5.37：规格不一致”错误，并且中止执行。

格式

CURTRQ [< 机器人编号 >] (< 表达式 >)

值 < 机器人编号 > 1 ~ 4
 < 表达式 > 1 ~ 6

含义 获取 < 机器人编号 > 指定机器人轴之中 < 表达式 > 指定轴的当前扭矩值 (-100 ~ 100)。
 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
 该值表示相对于最大扭矩指令值的比率。另外，符号表示方向。

SAMPLE

A = CURTRQ(3) 将机器人 1 中第 3 轴的当前扭矩值赋给变量 A。

格式

CUT	Tn “< “<程序名称>” >” PGm
-----	-----------------------------

值 m : 程序编号 0 ~ 99
n : 任务编号 1 ~ 16

含义 强制结束直接执行中或暂停中的其他任务。任务可根据正在执行的程序名称或者编号进行指定。
无法强制结束当前的任务。

MEMO

· 当指定并执行了未启动的任务(程序)时, 将会发生错误。

SAMPLE

```
' TASK1 ROUTINE
*ST:
  MO (20) = 0
  START <SUB_PGM>,T2
  MOVE P, P0
  MOVE P, P1
  WAIT MO (20) = 1
  CUT T2
GOTO *ST
HALT
程序名称 :SUB_PGM
*SUBPGM:
' TASK2 ROUTINE
*SUBTASK2:
  P100=JTOXY (WHERE)
  IF LOC3 (P100) > = 100.0 THEN
    MO (20) = 1
  ELSE
    DELAY 100
  ENDIF
GOTO *SUBPGM
EXIT TASK
```

相关命令 EXIT TASK, CUT, RESTART, START, SUSPEND

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

DATES\$

含义 通过 yyyy/mm/dd 形式的字符串赋予日期。
yyyy 表示公历、mm 表示月、dd 表示日。
日期通过悬吊式操作盒等操作终端进行设定。

SAMPLE

```
A$=DATES$  
PRINT DATES$  
HALT
```

相关命令 TIMES\$

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

1. DECEL [<机器人编号>] <表达式>
2. DECEL [<机器人编号>] (<轴编号>) = <表达式>

值 <机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6
 <表达式> 1 ~ 100 (单位: %)

含义 将<机器人编号>指定机器人轴的减速比率参数变更为<表达式>的值。<机器人编号>可以省略。当进行省略时, 机器人 1 被指定。
 形式 1 为变更指定机器人的所有轴。
 格式 2 仅更改<轴编号>中指定的轴。

MEMO

- 在生成过程中, 如果指定 no axis 中所设定的轴时, 则会发生“5.37 : 规格不一致”错误, 并且中止执行。
- 命令语句: 当使用 ACCEL 时, 可以变更加速度参数。

函数

格式

DECEL [<机器人编号>] (<轴编号>)

值 <机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6

含义 赋予<机器人编号>指定机器人轴之中<轴编号>指定轴的减速比率参数值。<机器人编号>可以省略。当进行省略时, 机器人 1 被指定。

SAMPLE

```
A =50
DECEL A
DECEL (3) =100
' CYCLE WITH INCREASING DECELERATION
FOR A =10 TO 100 STEP 10
  DECEL A
  MOVE P, P0
  MOVE P, P1
NEXT A
A=DECEL (3) ..... 将机器人 1 中第 3 轴的减速比率参数赋给变量 A。
HALT "END TEST"
```

格式

```
DEF FN <名称> [|%|] [( <形参> , [ <形参> ... ] ) ] = <函数的定义表达式>
                |!|
                |$|
```

值 <名称> 函数名称。包括 FN 在 16 个字符以内
<形参> 数值型或字符型的变量

含义 DEF FN 是定义用户可使用函数的命令。已定义的函数可通过 FN <名称> (<变量>) 的形式调出并使用。

MEMO

- <形参> 中使用与 <函数的定义表达式> 中同名的变量。此变量名仅在评估 <函数的定义表达式> 时有效，程序中也可以存在相同变量名。
- 调出使用 <形参> 的函数时，指定与 <形参> 相同类型的常量、变量或表达式。<伪参数> 可以省略。
- <形参> 列表中没有 <函数的定义表达式> 中所使用的变量时，变量将使用此时所持值。
- 请在 DEF 与 FN 之间插入空格。如果不插入空格，DEFFN 则将被作为变量处理。
- DEF FN 语句无法在子过程内使用。
- DEF FN 语句的定义必须在使用函数之前声明。

SAMPLE

```
DEF FNPAI=3.141592
DEF FNASIN (X) =ATN (X / SQR (-X^2+1))
..... <形参> <函数的定义表达式> 任何一个均使用 X
.
.
A=FNASIN (B) *10 ..... 调出时，也可不为 X
```

格式

DEGRAD (<表达式>)

值 <表达式> 角度(单位:度)

含义 将<表达式>的值转换为弧度。
若要将弧度转换为度, 请使用 RADDEG。

SAMPLE

A=COS (DEGRAD (30)) 将 30° 的余弦值赋值给变量 A

相关命令 ATN, COS, RADDEG, SIN, TAN

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

DELAY <表达式>

值 <表达式> 1 ~ 3600000 (单位 : ms)**含义** 仅等待<表达式>中指定的时间(单位 ms)。但是, 最小执行等待时间为 1ms。

SAMPLE

DELAY 3500 等待 3,500ms (3.5 秒)

DELAY A*10

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

1. [LET] <表达式> = DI_m ([b, . . . , b])
2. [LET] <表达式> = DI (mb, . . . , mb)

值 m..... 端口编号：0 ~ 7, 10 ~ 17, 20 ~ 27
 b..... 指定比特：0 ~ 7

含义 表示并行输入信号的状态。
 指定多个比特时，请从左边开始降序(大→小)记述。
 实际不存在输入端口时，将输入 0。
 省略了 [b, . . . , b] 时，8 比特均将成为对象。

SAMPLE

A%=DI2 () 将 DI (27) 至 DI (20) 的输入状态赋值给变量 A%
 A%=DI5 (7, 4, 0) 将 DI (57), DI (54), DI (50) 的输入状态赋值
 给变量 A% (上述信号均为 1 (ON) 时，将变为
 A%=7)
 A%=DI (37, 25, 20)..... 将 DI (37), DI (25), DI (20) 的输入状态赋值
 给变量 A% (上述信号除 DI (20) 以外均为 1
 (ON) 时，将变为 A%=6)

参照 详细说明，请参阅第 3 章“9.3 并行输入变量”。

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

```
DIM <数组指定> [, <数组指定> ,...]
```

数组指定

```
<名称> [ | % | ] ( <常量> [, <常量> [, <常量> ] ] )
          | ! |
          | $ |
```

值 <常量> 数组的角标：0 ~ 32,767 的正整数

含义 对直接数组变量的名称及元素数进行声明。数组的角标最多可使用 3 维。使用 ","(逗号) 隔开可在 1 行内声明多个数组。

MEMO

- 数组的角标从 0 至指定的值为止，合计为 <常量> + 1 个。
- 根据数组各维度的大小不同，可能会发生 [9.31: Memory full] 的错误。

SAMPLE

```
DIM A% (10) ..... 定义整数型数组变量 A% (0) ~ A% (10) (元素数 11)
DIM B (2, 3, 4) ..... 定义实数型数组变量 B (0, 0, 0) ~ B (2, 3, 4) (元素数 60)
DIM C% (2, 2), D! (10) ..... 定义整数型数组 C% (0, 0) ~ C% (2, 2) 与实数型数组 D! (0) ~ D! (10)
```

格式

DIST (<坐标点表达式 1 > , <坐标点表达式 2 >)

值 <坐标点表达式 1 > 正交坐标系的坐标点
<坐标点表达式 2 > 正交坐标系的坐标点

含义 赋予 <坐标点表达式 1 > 与 <坐标点表达式 2 > 中所表示的 2 点间 (X, Y, Z) 的距离。
如果 2 个 <坐标点表达式 > 不采用正交坐标形式，将出错。

SAMPLE

A=DIST (P0, P1)..... 将 P0 与 P1 间的距离赋值给变量 A

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

1. [LET] DOm ([b, . . . , b]) = <表达式>
2. [LET] DO ([mb, . . . , mb]) = <表达式>

值

m : 端口编号..... 2 ~ 7, 10 ~ 17, 20 ~ 27

b : 指定比特..... 0 ~ 7

对于左边指定的比特, 输出的值将变为<值>整数化后的后位比特。

含义

将直接指定值输出至 DO 端口。

指定多个比特时, 请从左边开始降序(大→小)记述。

指定了实际不存在的 DO 端口时, 将无任何输出。

省略了 [b, . . . , b] 时, 8 比特均将成为对象。

无法输出至 DO0 () 与 DO1 ()。仅限参照。

SAMPLE

DO2 () = &B10111000 DO (27, 25, 24, 23) 开启,
DO (26, 22, 21, 20) 关闭

DO2 (6, 5, 1) = &B010 DO (25) 开启, DO (26, 21) 关闭

DO3 () = 15 DO (33, 32, 31, 30) 开启,
DO (37, 36, 35, 34) 关闭

DO (37, 35, 27, 20) = A 将变量 A 整数化后的后 4 比特内容输出至 DO
(37, 35, 27, 20)

相关命令

RESET, SET

格式

DRIVE [<机器人编号>](<轴编号>, <表达式>)[, (<轴编号>, <表达式>)...][, 选项]

值

<机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6
 <表达式> 马达位置(毫米、度、脉冲)或坐标点表达式

含义

执行<机器人编号>指定机器人的以轴为单位的绝对移动命令。
 <机器人编号>可以省略。当进行省略时, 机器人 1 被指定。
 当附加轴进行绝对移动时, 将使用该命令。

- 移动种类: 轴指定 PTP
- 坐标点指定: 直接数值指定、坐标点定义指定
- 选项: 速度指定、STOPON 条件指定、XY 指定、指定移动方向 可指定多个选项。

移动种类

● 轴指定 PTP (Point to Point)

<轴编号>中指定的所有轴动作完成后(轴至达公差范围内以后), 开始移动, 并在进入 OUT 有效位置范围内后结束命令。指定多个轴时, 将同时到达。

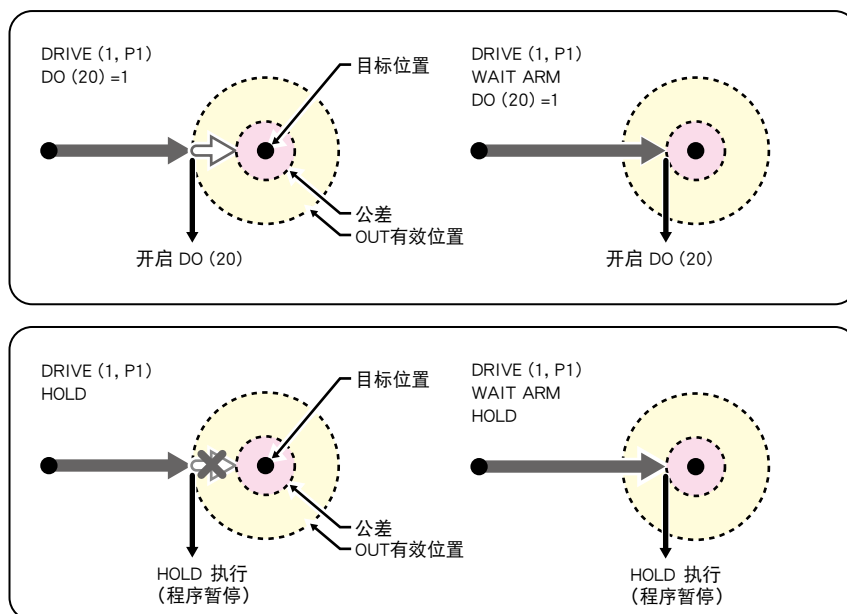
DRIVE 命令的下一个命令为信号输出命令等可执行命令时, 移动对象轴进入 OUT 有效位置范围内后, 将执行下一个命令。因此, 请在轴到达目标位置的公差范围之内之前, 执行下一个命令。

示例:

信号输出 (DO 等)	进入 OUT 有效位置范围内后, 将输出信号。
DELAY	进入 OUT 有效位置范围内后, 将执行 DELAY 命令, 并开始等待时间。
HALT	进入 OUT 有效范围内后, 程序将停止并被复位。因此, 移动将停止。
HALTALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部停止, 任务 1 被复位, 其它任务结束。因此, 移动也会停止。
HOLD	进入 OUT 有效范围内后, 程序将暂停。因此, 移动将停止。
HOLDALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部暂停。因此, 移动也会停止。
WAIT	进入 OUT 有效位置范围内后, 将执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时，请使用 "WAIT ARM" 语句。

DRIVE命令



35819-R7-00

SAMPLE

`DRIVE (1, P0)` 第 1 轴从当前位置向 P0 指定的位置移动

坐标点指定的种类

● 直接数值指定

在 <表达式> 中直接指定马达位置

当马达位置的数值为整数时，判断为脉冲单位。而当数值为实数值时，判断为毫米 / 度单位，每个轴从 0 脉冲位置向脉冲换算位置移动。

但是，使用选项指定 XY 时，则从坐标原点开始移动。

SAMPLE

`DRIVE (1, 10000)` 机器人 1 中第 1 轴从当前位置向 10000 脉冲处移动。

`DRIVE [2] (2, 90.00)` 机器人 2 中第 2 轴从当前位置开始，向由 0 脉冲位置沿 + 方向旋转 90 度的位置进行移动 (当为旋转轴时)。

● 坐标点定义指定

在<表达式>中指定坐标点表达式。使用<轴编号>中指定轴的数据。

当坐标点表达式为毫米/度单位时，每个轴从0脉冲位置向脉冲换算位置移动。

但是，使用选项指定XY时，从坐标原点开始移动。



要点

- 同时存在两种指定单位时(整数与实数同时存在时)，所有值被视作毫米/度单位。

SAMPLE

DRIVE (1, P1)	机器人 1 中第 1 轴从当前位置向 P1 指定的位置进行移动。
DRIVE (4, P90)	第 4 轴从当前位置 0 脉冲位置向 P90 所指定的度数位置移动(旋转轴情况下)

选项种类

● 速度指定

格式

- 1.SPEED = <表达式>
- 2.S = <表达式>



要点

- 虽然规定了最高速度，但并不保证按照指定速度移动。

值 <表达式>1 ~ 100(单位: %)

含义 使用<表达式>指定程序移动速度。

实际速度如下所示。

- 机器人的最高速度(mm/秒或度/秒) × 自动移动速度(%) × 程序移动速度(%) 仅对指定的 DRIVE 语句有效。

SAMPLE

DRIVE [2] (1,10000),S=10	机器人 2 中第 1 轴以相当于自动移动速度 10% 的速度，从当前位置向 10000 脉冲处进行移动。
--------------------------------	--

格式

- 1.DSPEED = <表达式>
- 2.DS = <表达式>



要点

- SPEED 选项与 DSPEED 选项无法并用。

值 <表达式>0.01 ~ 100.00(单位: %)

含义 使用<表达式>指定轴的移动速度。

实际速度如下所示。

- 机器人的最高速度(mm/秒或度/秒) × 轴的移动速度(%)

仅对指定的 DRIVE 语句有效。

- 不受自动移动速度值(%)的影响，一贯按照 DSPEED 的<表达式>值(%)移动。

SAMPLE

DRIVE [2] (1,10000),DS=0.1	机器人 2 中第 1 轴以相当于机器人最高速度 0.1% 的速度，从当前位置向 10000 脉冲处进行移动。
----------------------------------	--

● STOPON 条件指定

格式

STOPON <条件表达式>

含义 移动中条件表达式不成立时，移动将停止。此时的停止动作为**减速停止**，在**条件成立后将产生减速移动距离**。

移动开始时如果条件表达式成立，将不移动，直接结束命令。
仅程序执行时有效。

SAMPLE

DRIVE (1, 10000), STOPON DI (20) =1

..... 第 1 轴从当前位置向 10000 脉冲位置移动，如果中途 DI (20) =1 的条件成立，则停止移动并执行下一个步骤。

MEMO

- 在指定 STOPON 条件中使用的条件表达式表示数值的情况下，当表达式的值非 0 时为“真”，当成为 0 时则为“假”。

● XY 指定

格式

XY

含义 将多个指定轴移动至正交坐标中指定的位置。
指定的轴同时到达。

在 MOVE 语句中指定了所有可移动的轴时，将变为与 MOVE 语句相同的动作。
以下所示为此命令的限制事项。

1. <轴编号>中指定的轴必须包含第 1 轴和第 2 轴 (2 个轴)。
2. 可指定的机器人为水平多关节型及 XY 型。
3. 指定坐标点时必须采用毫米 / 度 (实数型) 单位。

SAMPLE

DRIVE (1, P100), (2, P100), (4, P100), XY

..... 第 1 轴、第 2 轴、第 4 轴从当前位置向 P100 指定的正交坐标位置移动

- 指定移动方向

格式

PLS
MNS

含义

< 有指定移动方向时 >

- 向指定方向进行移动。当指定 PLS 时，必须向 + 方向移动；当指定 MNS 时，则必须向 - 方向移动。
- 当目标位置和当前位置相等时，沿指定方向移动 1 个周期后停止。

< 无指定移动方向时 >

- 向缩短移动距离的方向进行移动。
- 当目标位置和当前位置相等时，不进行移动。

· 注意事项

1. 当使用该选项时，每次的最大移动距离相当于 1 个周期的距离（ 360° ）。当希望移动超过 1 个周期的距离时，请分为 2 次以上进行移动。

2. 当使用该选项时，DRIVE 语句的软限位按照下列数值进行。

+ 软限位： 67,000,000[pulse]

- 软限位： -67,000,000[pulse]

· 限制事项

1. 可指定的机器人仅限于单轴的旋转型机器人。
2. 当指定了移动方向时，不能同时对多根轴进行移动。当指定了多根轴时，将发生“5.37: 规格不一致”错误。（参照下述内容）
例：DRIVE (3,P1),(4,P1),PLS
3. 不能同时指定 PLS 选项和 MNS 选项。
4. 对于无限位动作作为无效的轴，当使用了本选项时，则会发生“2.29 无限位动作不可”错误。
5. 在使用本选项进行动作的过程中，在利用 STOP 键等使其停止（包括减速中）的情况下，当重新启动时，将会移动相当于 1 个周期（ 360° ）的距离。

SAMPLE

DRIVE(4,270.00),PLS

..... 当机器人的当前位置为 260° 时：从当前位置沿 + 方向移动 10°

当机器人的当前位置为 280° 时：从当前位置沿 + 方向移动 350°

DRIVE(4,270.00),MNS

..... 当机器人的当前位置为 260° 时：从当前位置沿 - 方向移动 350°

当机器人的当前位置为 280° 时：从当前位置沿 - 方向移动 10°

DRIVE(4,270.00)

..... 当机器人的当前位置为 260° 时：从当前位置沿 + 方向移动 10°

当机器人的当前位置为 280° 时：从当前位置沿 - 方向移动 10°

DRIVE[2](3,270.00),PLS

当机器人的当前位置为 260° 时：从当前位置沿 + 方向移动 10°

当机器人的当前位置为 280° 时：从当前位置沿 + 方向移动 350°

DRIVE[2](3,270.00),MNS

..... 当机器人的当前位置为 260° 时：从当前位置沿 - 方向移动 350°

当机器人的当前位置为 280° 时：从当前位置沿 - 方向移动 10°

DRIVE[2](3,270.00)

..... 当机器人的当前位置为 260° 时：从当前位置沿 + 方向移动 10°

当机器人的当前位置为 280° 时：从当前位置沿 - 方向移动 10°

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

DRIVEI (<轴编号> , <表达式>) [, (<轴编号> , <表达式>) ...] [, 选项]

值

<机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6
 <表达式> 马达位置(毫米、度、脉冲)或坐标点表达式

含义

直接执行组内以轴为单位的相对移动命令。
 当对附加轴进行相对移动时, 将使用该移动命令。

- 移动种类: 轴指定 PTP
- 坐标点指定: 直接数值指定、坐标点定义指定
- 选项: 速度指定、STOPON 条件指定 可指定多个

MEMO

- 中途停止 DRIVEI 语句并重新启动时, 可在控制器 "other parameters" 的 "MOVEI / DRIVEI start position" 中选择移动的目标位置。详细说明, 请参阅控制器的手册。
 - 1) 保持(初始值) 继续执行之前的移动。目标位置与再次执行前相同。
 - 2) 复位 从当前位置开始重新进行相对移动。目标位置与再次执行前不同。

移动种类

● 轴指定 PTP (Point to Point)

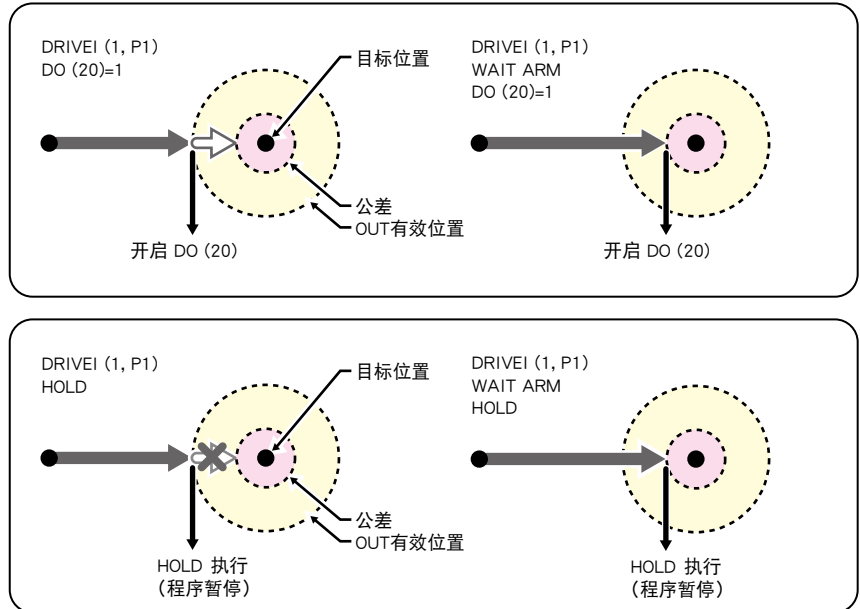
<轴编号> 中指定的所有轴动作完成后(轴到达公差范围内以后), 开始移动, 并在进入 OUT 有效位置范围内后结束命令。指定多个轴时, 将同时到达。

DRIVEI 命令的下一个命令为信号输出命令等可执行命令时, 移动对象轴进入 OUT 有效位置范围内后, 将执行下一个命令。因此, 请在轴到达目标位置的公差范围内之前, 执行下一个命令。
 示例:

信号输出 (DO 等)	进入 OUT 有效位置范围内后, 将输出信号。
DELAY	进入 OUT 有效位置范围内后, 将执行 DELAY 命令, 并开始等待时间。
HALT	进入 OUT 有效范围内后, 程序停止并被复位。因此, 移动将停止。
HALTALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部停止, 任务 1 被复位, 其它任务结束。因此, 移动也会停止。
HOLD	进入 OUT 有效范围内后, 程序将暂停。因此, 移动将停止。
HOLDALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部暂停。因此, 移动也会停止。
WAIT	进入 OUT 有效位置范围内后, 将执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时，请使用 "WAIT ARM" 语句。

DRIVEI 命令 WAIT ARM 语句



35820-R7-00

无限位动作时的注意点

- 当将无限位动作参数设定为有效时，DRIVEI 语句的软限位检查值将成为如下所示数值。
 - + 软限位： 67,000,000[pulse]
 - 软限位： -67,000,000[pulse]
- 当使用 DRIVEI 语句时，每次的最大移动距离为上述数值。

SAMPLE

DRIVEI (1, P0)..... 第 1 轴从当前位置向 P0 指定的位置移动

坐标点指定的种类

● 直接数值指定

在<表达式>中直接指定马达位置

当马达位置的数值为实数值时,判断为毫米/度单位,每个轴从0脉冲位置向脉冲换算位置移动。

SAMPLE

DRIVEI (1, 10000)..... 第1轴从当前位置向+10000脉冲位置移动
 DRIVEI (4, 90.00)..... 第4轴从当前位置同时+90度位置移动(旋转轴
 情况下)



要点

- 同时存在两种指定单位时(整数与实数同时存在时),所有值被视作毫米/度单位。

● 坐标点定义指定

在<表达式>中指定坐标点表达式。使用<轴编号>中指定轴的数据。

将根据坐标点表达式中定义的坐标点数据指定马达位置。当坐标点表达式为毫米/度单位时,各轴从0脉冲位置向脉冲换算位置移动。

SAMPLE

DRIVEI (1, P1)..... 第1轴从当前位置移动P1所指定的移动量
 DRIVEI (4, P90)..... 第4轴从当前位置移动P90所指定的度数移动
 量(旋转轴情况下)

A

B

C

D

E

F

G

H

I

J

K

L

M

选项种类

● 速度指定

格式

1. SPEED= <表达式>
2. S= <表达式>

值 <表达式>1 ~ 100 (单位: %)

含义 使用<表达式>指定程序移动速度。
运行速度如下所示。
· 机器人的最高速度 (mm / 秒或度 / 秒) × 自动移动速度 (%) × 程序移动速度 (%)
仅对指定的 DRIVEI 语句有效。

SAMPLE

DRIVEI (1, 10000), S=10 第 1 轴按照自动移动速度的 10% 从当前位置向 +10000 脉冲位置移动

格式

1. DSPEED= <表达式>
2. DS= <表达式>

值 <表达式>0.01 ~ 100.00 (单位: %)

含义 使用<表达式>指定轴的移动速度。
实际速度如下所示。
· 机器人的最高速度 (mm / 秒或度 / 秒) × 轴的移动速度 (%)
仅对指定的 DRIVE 语句有效。
· 不受自动移动速度值 (%) 的影响, 一贯按照 DSPEED 的<表达式>值 (%) 移动。

SAMPLE

DRIVEI (1, 10000), DS=0.1 第 1 轴按照自动移动速度的 0.1% 从当前位置向 +10000 脉冲位置移动



要点

- 虽然规定了最高速度, 但并不保证按照指定速度移动。



要点

- SPEED 选项与 DSPEED 选项无法并用。

- STOPON 条件指定

格式

STOPON < 条件表达式 >

含义 移动中条件表达式成立时，移动将停止。此时的停止动作作为**减速停止**，在条件成立后将产生**减速移动距离**。

移动开始时如果条件表达式成立，将不移动，直接结束命令。

仅程序执行时有效。

 MEMO

- 在指定 STOPON 条件中使用的条件表达式表示数值的情况下，当表达式的值非 0 时为“真”，当成为 0 时则为“假”。

SAMPLE

DRIVEI (1, 10000), STOPON DI (20) =1

..... 第 1 轴从当前位置向 +10000 脉冲位置移动，
如果中途 DI (20) =1 的条件成立，则停止移动
并执行下一个步骤。

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

```

SELECT [CASE] <表达式>
  CASE <表达式的列表 1 >
    [命令区 1]
  [CASE <表达式的列表 2 >
    [命令区 2]]
  :
  [CASE ELSE
    [命令区 n]]
END SELECT

```

含义 直接结束 SELECT CASE 的命令区。
 详细内容，请参阅“92 SELECT CASE ~ END SELECT”。

SAMPLE

```

WHILE -1
SELECT CASE DI3 ( )
  CASE 1, 2, 3
    CALL *EXEC (1, 10)
  CASE 4, 5, 6, 7, 8, 9, 10
    CALL *EXEC (11, 20)
  CASE ELSE
    CALL *EXEC (21, 30)
END SELECT
WEND
HALT

```

相关命令 SELECT CASE

格式

```
SUB <标签> [( <形参> [, <形参> ... ] )]  
    <命令区>  
END SUB
```

含义

结束以 SUB 语句开始的子过程定义。
详细内容，请参阅“105 SUB ~ END SUB”。

SAMPLE 1

```
I=1  
CALL *TEST  
PRINT I  
HALT  
' SUB ROUTINE: TEST  
SUB *TEST  
    I=50  
END SUB
```

相关命令

CALL, EXIT SUB, SUB ~ END SUB

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

```
ERR (<任务编号>)  
ERL (<任务编号>)
```

值 <任务编号> 1 ~ 4

含义 变量 ERR 及 ERL 用于在 ON ERROR GOTO 语句指定的错误处理例程中进行错误处理。利用<任务编号>指定的任务 ERR 将获取发生错误时的错误代码编号，ERL 则获取发生错误的行编号。

SAMPLE 1

```
IF ERR (1) <> &H604 THEN HALT  
IF ERL (1) =20 THEN RESUME NEXT
```

相关命令 ON ERROR GOTO, RESUME

格式

EXIT FOR

含义 直接强制结束 FOR 语句~ NEXT 语句的循环。并跳转至 NEXT 语句的下一个命令。
此命令语句仅在 FOR 语句~ NEXT 语句之间有效。

 MEMO

· 请使用 FOR 语句的条件或 EXIT FOR 结束 FOR ~ NEXT 的循环。使用 GOTO 等语句跳出循环时，可能会发生 [5.12: Stack overflow] 等错误。

SAMPLE

```
*ST:
WAIT DI (20) =1
FOR A%=101 TO 109
  MOVE P, P100, Z=0
  DO (20) =1
  MOVE P, P [A%] , Z=0
  DO (20) =0
  IF DI (20) =0 THEN EXIT FOR
NEXT A%
GOTO *ST
HALT
```

相关命令 FOR, NEXT

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

EXIT SUB

含义 强制结束 SUB 语句~ END SUB 语句中定义的子过程。跳转至调出子过程的 CALL 语句的下一个命令。

此命令语句仅在 SUB 语句~ END SUB 语句定义的子过程内有效。

MEMO

· 请使用 END SUB 语句或 EXIT SUB 语句结束 SUB ~ END SUB 所定义的子过程。使用 GOTO 等语句跳出循环时，可能会发生 [5.12: Stack overflow] 等错误。

SAMPLE

```
' MAIN ROUTINE
CALL *SORT2 (REF X%,REF Y%)
HALT
' SUB ROUTINE: SORT
SUB *SORT2 (X%, Y%)
  IF X% > =Y% THEN EXIT SUB
  TMP%=Y%
  Y%=X%
  X%=TMP%
END SUB
```

相关命令 CALL, SUB ~ END SUB, END SUB

格式

EXIT TASK

含义 结束当前正在执行的任务。

SAMPLE

```
' TASK1 ROUTINE
*ST:
  MO (20) =0
  START <SUB_PGM>,T2
  MOVE P, P0, P1
  WAIT MO (20) =1
  GOTO *ST
HALT
程序名称 :SUB_PGM
*SUBPGM:
' TASK2 ROUTINE
*SUBTASK2:
  P100=JTOXY (WHERE)
  IF LOCZ (P100) > =100.0 THEN
    MO (20) =1
    EXIT TASK
  ENDIF
  DELAY 100
GOTO *SUBPGM
EXIT TASK
```

相关命令 CUT, RESTART, START, SUSPEND, CHGPRI

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

```
FOR <控制变量> = <开始值> TO <结束值> [STEP <步骤>]  
  <命令区>  
NEXT [<控制变量>]
```

含义 直接将<控制变量>的值按照<步骤>的值逐步变化，反复执行FOR语句的下一命令至NEXT语句，直至<开始值>超过<结束值>为止。

省略了“STEP <步骤>”时的步骤值为1。

<步骤>的值也可为负数。

<控制变量>为数值型的<简单变量>或<数组变量>。

FOR语句与NEXT语句必须成对使用。

SAMPLE

```
'CYCLE WITH CYCLE NUMBER OUTPUT TO DISPLAY  
FOR A=1 TO 10  
  MOVE P, P0  
  MOVE P, P1  
  MOVE P, P2  
  PRINT "CYCLE NUMBER=";A  
NEXT A  
HALT
```

相关命令 EXIT FOR

格式

```
GOSUB <标签>                                ※GOSUB 也可写成 GO SUB
:
<标签> :
:
RETURN
```

含义 跳转至由 GOSUB 语句中指定 <标签> 的子例程。
通过子例程执行中的 RETURN 语句跳转至 GOSUB 语句的下一行。

 MEMO

- GOSUB 语句最多可连续使用 120 次。但是，根据 FOR 语句或 CALL 语句等使用堆栈命令的使用状态的不同，次数将减少。
- 使用 GOSUB 语句跳转至子例程时，请务必使用 RETURN 语句结束子例程。通过 GOTO 等语句从子例程跳转时，可能会发生 [5.12: Stack overflow] 等错误。

SAMPLE

```
*ST:
MOVE P, P0
GOSUB *CLOSEHAND
MOVE P, P1
GOSUB *OPENHAND
GOTO *ST
HALT
' SUB ROUTINE
*CLOSEHAND:
    DO (20) = 1
RETURN
*OPENHAND:
    DO (20) = 0
RETURN
```

相关命令 RETURN

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

GOTO < 标签 >

※GOTO 也可写成 GO TO

含义

无条件跳转至 < 标签 > 所指定的行。

要根据条件选择跳转目标位置时，请使用 ON ~ GOTO 或 IF 语句。

SAMPLE

```
' MAIN ROUTINE
*ST:
  MOVE P, P0, P1
  IF DI (20) = 1 THEN
    GOTO *FIN
  ENDIF
GOTO *ST
*FIN:
HALT
```

格式

```
HALT [ |<表达式>|
      |<字符串>| ]
```

含义 直接停止程序并进行复位。HALT 执行后重新启动程序时，将从程序起始开始执行。记述<表达式>或<字符串>后，手持编程器的画面上将显示<表达式>或<字符串>的内容。

MEMO

· HALT 仅对执行过的任务起作用。在其它任务中正在执行的程序将继续执行。

SAMPLE

```
' MAIN ROUTINE
*ST:
  MOVE P, P0, P1
  IF DI (20) = 1 THEN
    GOTO *FIN
  ENDIF
GOTO *ST
*FIN:
HALT "PROGRAM FIN"
```

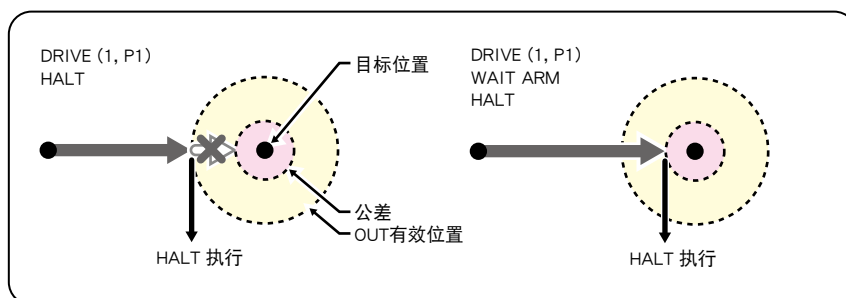
在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。

因此，如果在 PTP 移动命令的后面紧接着 HALT 命令时，在轴到达目标位置的公差范围内之前，将执行 HALT 命令。

同样地，MOVE 命令的插补移动由于在移动开始后立即执行下一个命令，如果在 MOVE 的插补移动命令后面紧接着 HALT 命令，则将在移动开始后立即执行 HALT 命令。

在任意一种情况下，如果在轴到达目标位置的公差范围内后要执行 HALT 命令时，请使用 WAIT ARM。

HALT命令



35821-R7-00

格式

```
HALTALL [ <表达式> | <字符串> ]
```

含义 直接停止并复位所有的程序。当执行 HALTALL 后重新启动程序时，将从主程序或者任务 1 中最后执行程序的最前端开始执行。当对 <表达式> 或 <字符串> 进行记述时，编程盒的画面上将会显示 <表达式> 或 <字符串> 的内容。

SAMPLE

```
' MAIN ROUTINE
*ST:
  MOVE P,P0,P1
  IF DI (20) = 1 THEN
    GOTO *FIN
  ENDIF
GOTO *ST
*FIN:
  HALTALL "PROGRAM FIN"
```

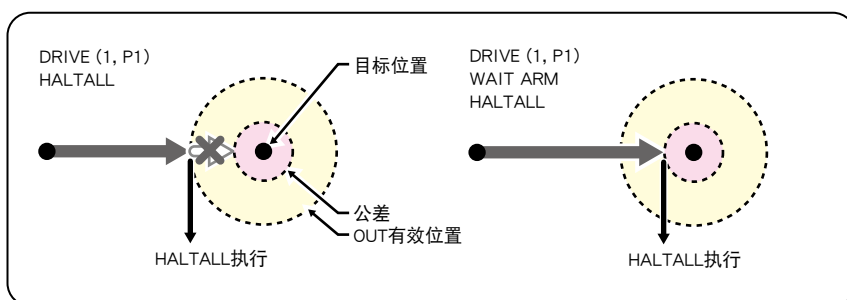
在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。

因此，如果在 PTP 移动命令的后面紧接着 HALTALL 命令时，在轴到达目标位置的公差范围内之前，将执行 HALTALL 命令。

同样，在 MOVE 命令的插值移动中指定了 CONT 选项的情况下，刚开始移动后将执行下一个命令，因此，当指定了 MOVE 中 CONT 选项的插值移动命令之后紧接有 HALTALL 命令时，刚开始移动后将执行 HALTALL 命令。

无论在哪种情况下，当希望在轴到达目标位置的公差范围以内后，开始执行 HALTALL 命令时，请使用 WAIT ARM。

HALTALL命令



35701-R9-00

格式

定义语句：

HAND [<机器人编号>] Hn= <第1参数> <第2参数> <第3参数> [R]

选择语句：

CHANGE [<机器人编号>] Hn

值 <机器人编号> 1 ~ 4
n: 机械手编号 0 ~ 31

含义 HAND 语句只可用于机械手的定义。实际进行机械手的切换时，使用 CHANGE 语句。关于 CHANGE 语句，请参阅“12 CHANGE”。

MEMO

· 如果在执行机械手定义语句过程中关闭了电源，将发生[9.7: Hand data destroyed]错误。

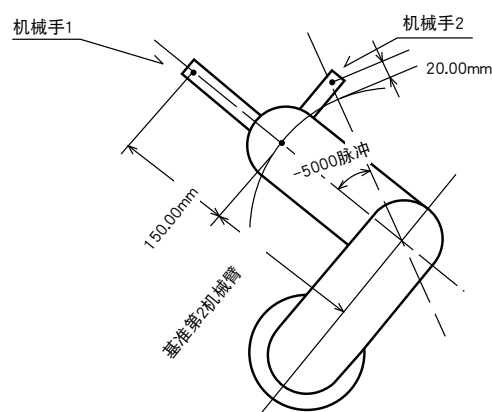
41.1 水平多关节型机器人时

1. 未指定<第4参数>的R时

成为固定在第2机械臂前端的机械手(参照下图)。

<第1参数> 基准第2机械臂与机械手n的假设第2机械臂之间的偏移脉冲数、逆时针方向为+[脉冲]
<第2参数> 机械手n的假设第2机械臂长度与基准第2机械臂长度之间的差[mm]
<第3参数> 机械手n的Z轴偏移量[mm]

未指定<第4参数>的R时



35803-R9-00

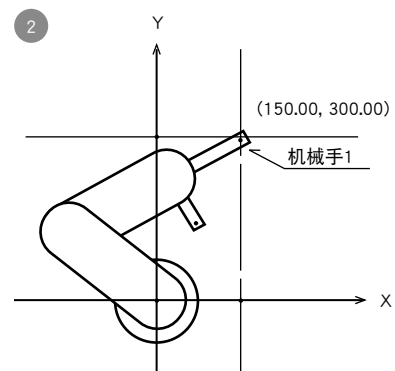
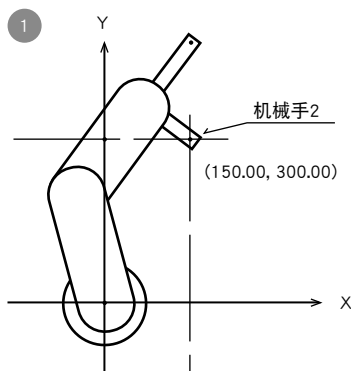
SAMPLE

```

HAND H1=      0   150.0   0.0
HAND H2=    -5000   20.00   0.0
P1=      150.00  300.00   0.00   0.00   0.00   0.00
CHANGE H2
MOVE P, P1 ..... 机械手 2 的前端向 P1 移动 ①
CHANGE H1
MOVE P, P1 ..... 机械手 1 的前端向 P1 移动 ②
HALT

```

SAMPLE: HAND



35802-R7-00

2. 指定了<第4参数>的R时

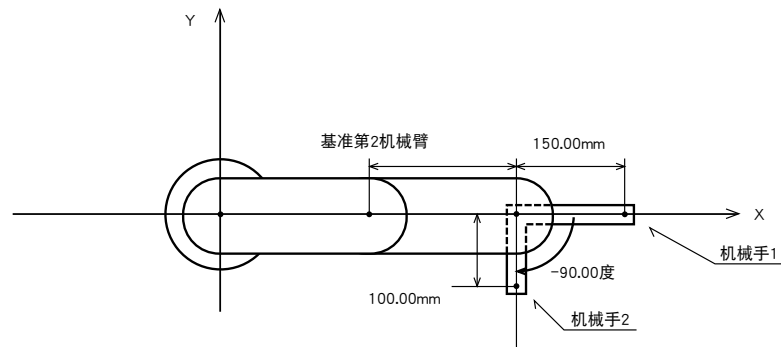
R轴为伺服时成为从R轴旋转中心偏移的机械手(参照下图)。

<第1参数> R轴的当前位置为0.00时, 正交坐标的+X方向与机械手n之间的角度, 逆时针方向为+[度]

<第2参数> 机械手n的长度[mm](> 0)

<第3参数> 机械手n的Z轴偏移量[mm]

指定了<第4参数>的R时



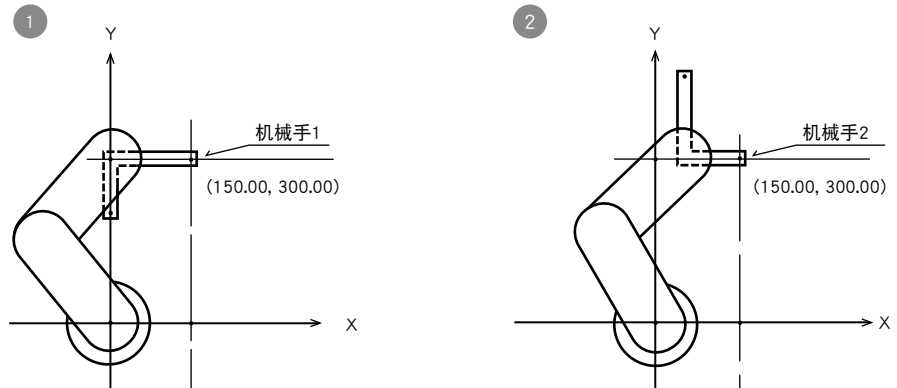
35804-R9-00

SAMPLE

```

HAND H1=    0.00    150.0    0.0    R
HAND H2=   -90.00    100.0    0.0    R
P1=    150.00    300.00    0.00    0.00    0.00    0.00
CHANGE H1
MOVE P, P1 ..... 机械手1的前端向P1移动 ①
CHANGE H2
MOVE P, P1 ..... 机械手2的前端向P1移动 ②
HALT
  
```

SAMPLE: HAND



35804-R7-00

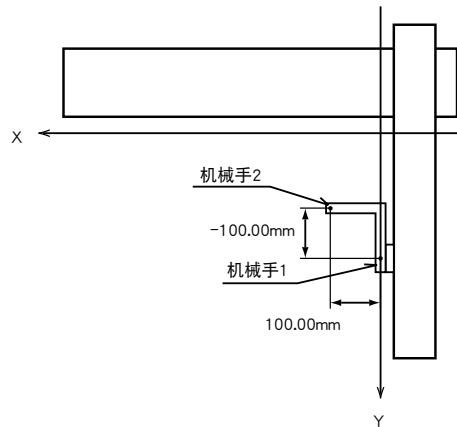
41.2 正交机器人时

1. 未指定<第4参数>的R时

成为固定在第2机械臂前端的机械手(参照下图)。

<第1参数> 机械手 n 的 X 轴偏移量 [mm]
 <第2参数> 机械手 n 的 Y 轴偏移量 [mm]
 <第3参数> 机械手 n 的 Z 轴偏移量 [mm]

未指定<第4参数>的R时



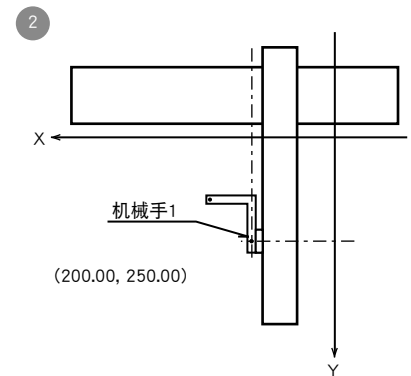
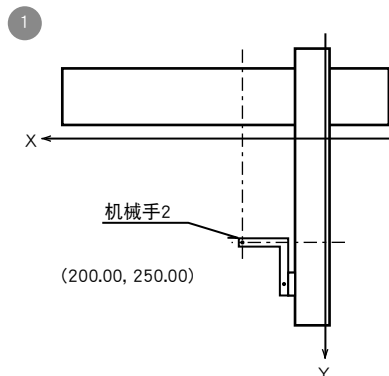
35805-R9-00

SAMPLE

```

HAND H1=    0.00    0.00    0.00
HAND H2=   100.0  -100.0  -100.0
P1=    200.00  250.00    0.00    0.00    0.00    0.00
CHANGE H2
MOVE P, P1 ..... 机械手 2 的前端向 P1 移动 ①
CHANGE H1
MOVE P, P1 ..... 机械手 1 的前端向 P1 移动 ②
HALT
  
```

SAMPLE: HAND



35806-R7-00

2. 指定了<第4参数>的R时

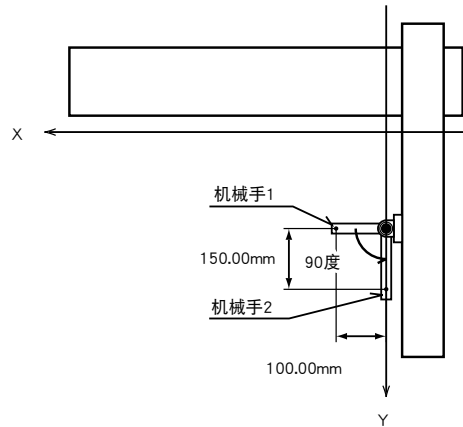
R轴为伺服时成为从R轴旋转中心偏移的机械手(参照下图)。

<第1参数> R轴的当前位置为0.00时, 正交坐标的+X方向与机械手n之间的角度, 逆时针方向为+(度)

<第2参数> 机械手n的长度[mm](> 0)

<第3参数> 机械手n的Z轴偏移量[mm]

指定了<第4参数>的R时



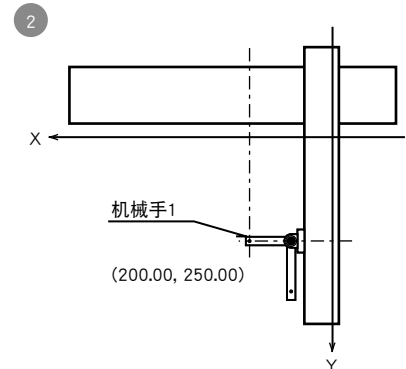
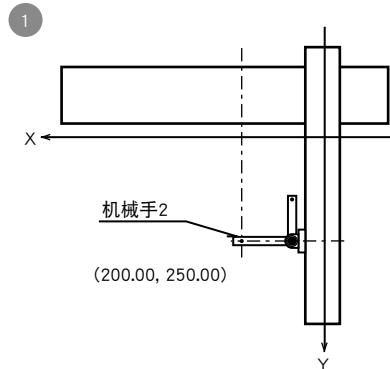
35806-R9-00

SAMPLE

```

HAND H1=    0.00   100.00   0.00    R
HAND H2=    90.00   150.00   0.00    R
P1=         200.00  250.00   0.00    0.00   0.00   0.00
CHANGE H2
MOVE P, P1----- 机械手2的前端向P1移动 ①
CHANGE H1
MOVE P, P1----- 机械手1的前端向P1移动 ②
HALT
  
```

SAMPLE: HAND



35808-R7-00

格式

```
HOLD [ <表达式> |
        <字符串> ]
```

含义 暂停程序。重新启动时，从 HOLD 语句的下一行开始执行。

记述 <表达式> 或 <字符串> 后，手持编程器的画面上将显示 <表达式> 或 <字符串> 的内容。

MEMO

· HOLD 仅对执行过的任务起作用。在其它任务中正在执行的程序将会继续执行。

SAMPLE

```
' MAIN ROUTINE
*ST:
  MOVE P, P0, P1
  IF DI (20) =1 THEN
    HOLD "PROGRAM STOP"
  ENDIF
GOTO *ST
HALT
```

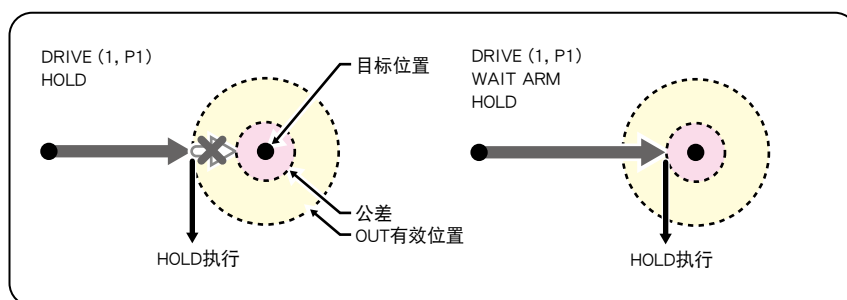
在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。

因此，如果在 PTP 移动命令的后面紧接着 HOLD 命令时，在轴到达目标位置的公差范围内之前，将执行 HOLD 命令。

同样地，MOVE 命令的插补移动由于在移动开始后立即执行下一个命令，如果在 MOVE 的插补移动命令后面紧接着 HOLD 命令，则将在移动开始后立即执行 HOLD 命令。

在任意一种情况下，如果在轴到达目标位置的公差范围内后要执行 HOLD 命令时，请使用 WAIT ARM。

HOLD命令



35822-R7-00

格式

```
HOLD [ | <表达式> | ]
        | <字符串> | ]
```

含义

暂停所有的程序。当重新启动时，执行了 HOLDALL 的程序将从语句的下一行开始执行，其它程序则将从中断的执行行开始重新运行。当对 <表达式> 或 <字符串> 进行记述时，编程盒的画面上将显示 <表达式> 或 <字符串> 的内容。

SAMPLE

```
SAMPLE
' MAIN ROUTINE
*ST:
MOVE P,P0,P1
IF DI (20) =1 THEN
  HOLD "PROGRAM STOP"
ENDIF
GOTO *ST
HALT
```

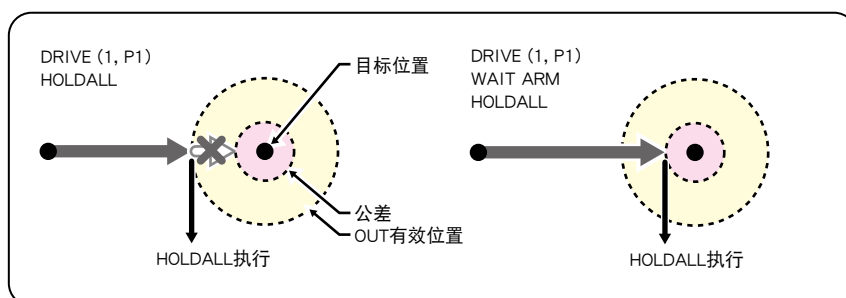
在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。

因此，如果在 PTP 移动命令的后面紧接着 HOLDALL 命令时，在轴到达目标位置的公差范围内之前，将执行 HOLDALL 命令。

同样，在 MOVE 命令的插值移动中指定了 CONT 选项的情况下，刚开始移动后将执行下一个命令，因此，当指定了 MOVE 中 CONT 选项的插值移动命令之后紧接有 HOLDALL 命令时，刚开始移动后将执行 HOLDALL 命令。

无论在哪种情况下，当希望在轴到达目标位置的公差范围以内，开始执行 HOLDALL 命令时，请使用 WAIT ARM。

HOLDALL命令



35702-R9-00



- 在 IF 语句所使用的条件表达式表示数值的情况下，当表达式的值非 0 时为“真”，表达式的值成为 0 时则为“假”。

44.1 简单 IF 语句

格式

```
IF <条件表达式> THEN | <标签 1> | [ELSE | <标签 2> | ]
                   | <命令语句 1> | | <命令语句 2> | ]
```

含义

<条件表达式>中指定的条件成立时，跳转至 THEN 以后指定的<标签 1>或者执行<命令语句 1>后，跳转至下一行。

<条件表达式>中指定的条件不成立时，进行以下动作。

1. 跳转至 ELSE 以后指定的<标签 2>或者执行<命令语句 2>后，跳转至下一行。
2. ELSE 语句以后无指定时，什么都不执行，跳转至下一行。

SAMPLE

```
' MAIN ROUTINE
*ST:
  MOVE P, P0, P1
  IF DI (20) =1 THEN *L1 ..... DI (20) 为 1 时，则跳转至 *L1
  DO (20) =1
  DELAY 100
*L1:
  IF DI (21) =1 THEN *ST ELSE *FIN
  ..... DI (21) 为 1 时，则跳转至 *ST。如果不是，
  则跳转至 *FIN
*FIN:
HALT
```

44.2 区块 IF 语句

格式

```
IF <条件表达式 1> THEN
  <命令区 1>
[ELSEIF <条件表达式 2> THEN
  <命令区 2>]
[ELSE
  <命令区 n>]
ENDIF
```

含义

<条件表达式 1> 中指定的条件成立时，执行<命令区 1> 中指定的命令群并跳转至 ENDIF 的下一行。

存在 ELSEIF 语句时，如果<条件表达式 2> 中指定的条件成立，将执行<命令区 2> 中指定的命令群。

所有条件表达式均不成立时，将执行<命令区 n>。

SAMPLE

```
' MAIN ROUTINE
*ST:
  MOVE P, P0, P1
  IF DI (21, 20) = 1 THEN
    DO (20) = 1
    DELAY 100
    WAIT DI (20) =0
  ELSEIF DI (21, 20) =2 THEN
    DELAY 100
  ELSE
    GOTO *FIN
  ENDIF
GOTO *ST
*FIN:
HALT
```

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

INPUT [<提示语句>	;]	<变量>	[,	<变量>	,...]
	,		<坐标点变量>		<坐标点变量>	
			<位移变量>		<位移变量>	

含义 从手持编程器中对指定的变量赋值。

输入定义如下所示。

1. 当用逗号 (,) 分隔指定多个变量时, 在输入数据中也必须用逗号 (,) 分隔输入指定个数的变量。
2. <提示语句> 中为了显示数据输入提示, 通过 "" 中的字符串指定。在 <提示语句> 的之后指定分号 (;) 时, 在提示的后面将显示问号 (?) 及空格。指定了逗号 (,) 时, 提示的后面将没有任何显示。
3. 省略了 <提示语句> 时, 将显示问号 (?) 及空格。
4. 输入的数据类型必须与对应变量的类型一致。在输入至坐标点变量或位移变量时, 不足的元素变为 0。
5. 不进行任何输入, 仅输入了 ENTER 键时, 将被视作输入了 0 或空字符串。但是, 进行多个指定时, 必须输入指定个数的逗号 (,)。
6. 指定变量为字符型变量, 且要在逗号 (,)、双引号 (") 或字符串的前后输入具有含义的空格 (1 字符空白) 时, 需要在字符串的外面加上双引号 (")。此外, 输入双引号 (") 时, 必须连续输入双引号 (")。
7. 输入了 ESC 键时, 此命令将被跳过。

MEMO

- 当变量和赋值类型不同时, 将显示指定的信息, 等待再次输入。
- 指定变量为字符型变量时, 即使不在字符串外面加上双引号 (") , 也可输入字母数字。

SAMPLE

```

INPUT A
INPUT "INPUT POINT NUMBER"; A1
INPUT "INPUT STRING", B$ (0), B$ (1)
INPUT P100
HALT

```


格式

INT (<表达式>)

含义

获得舍去小数点以下部分后的整数值。获取不超过<表达式>值的最大整数值。

SAMPLE

A=INT (A (0))

B=INT (-1. 233) B 被赋予 -2

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

JTOXY [< 机器人编号 >] (< 坐标点表达式 >)

值 < 机器人编号 > 1 ~ 4

含义 将 < 坐标点表达式 > 所表示的关节坐标数据 (单位: 脉冲) 转换为 < 机器人编号 > 指定机器人的正交坐标数据 (单位: mm、度)。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

在 YK500TW 机器人的情况下, 还应设定第 1 机械臂转数信息和第 2 机械臂转数信息。

SAMPLE

P10=JTOXY (WHERE)..... 将当前位置数据转换为正交坐标数据

相关命令 XYTOJ

格式

LEFT\$ (<字符串表达式> , <表达式>)

值 <表达式>0 ~ 25**含义** 从<字符串表达式>中指定的字符串左侧(最开始位置)提取<表达式>中指定位数的字符串。

<表达式>的值不在0 ~ 25范围内时,将出错。

<表达式>的值为0时,LEFT\$变为空字符串。

<表达式>的字符数大于<字符串表达式>的字符数时,LEFT\$将变为与<字符串表达式>相同的内容。

SAMPLE

B\$=LEFT\$(A\$, 4) 从A\$左侧将4个字符赋给B\$。

相关命令 MID\$, RIGHT\$

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

LEFTY [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 选择通过左手系统将 < 机器人编号 > 指定机器人移动到正交坐标系所指定的坐标点。
 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
 即使执行了命令**仅选择也无法使机器人移动**。轴在进行动作时，请等待动作完成（被定位至公差内）再执行。
 此命令仅对水平多关节机器人有效。

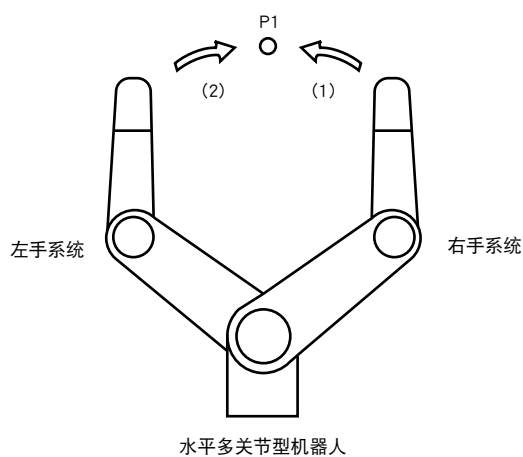
SAMPLE

```

RIGHTY
MOVE P, P1
LEFTY
MOVE P, P1
RIGHTY
HALT

```

SAMPLE : LEFTY / RIGHTY



35809-R7-00

相关命令 RIGHTY

格式

```
LEN (<字符串表达式>)
```

含义 返回<字符串表达式>中表示的字符串长度(字节数)

SAMPLE

```
B=LEN (A$)
```

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

[LET]	<算术赋值语句> <字符串赋值语句> <坐标点赋值语句> <位移赋值语句>
-------	--

含义 直接执行指定的赋值语句。赋值时，将右边的值赋值至左边。
"LET" 通常可省略，可直接记述赋值语句。

MEMO

· 当执行<坐标点赋值语句>、<位移赋值语句>过程中控制器电源被切断，则可能会发生 [9.2: Point data destroyed] 与 [9.6: Shift data destroyed] 等存储器相关错误。

1. 算术赋值语句

格式

[LET]	<算术变量> <并行输出变量> <内部输出变量> <机械臂锁定输出变量> <定时输出变量> <串行输出变量>	= <表达式>
-------	---	---------

值 <表达式> 变量、函数、数值

含义 将<表达式>的值赋值给左边的变量。

SAMPLE

```
A!=B!+1
B% (1, 2, 3) =INT (10.88)
DO2 ( ) =&B00101101
MO (21, 20) =2
LO (00) =1
TO (01) =0
SO12 ( ) =255
```

2. 字符串赋值语句

格式

```
[LET] <字符串变量> = <字符串表达式>
```

含义 将<字符串表达式>的值赋值给字符串变量。
<字符串表达式>中可使用的运算符仅限+。此外，不可使用括号。

SAMPLE

```
A$ = "YAMAHA"
B$ = "ROBOT"
D$ = A$ + "-" + B$
```

执行结果：YAMAHA-ROBOT

MEMO

· 运算符+用于联结字符串。

3. 坐标点赋值语句

格式

```
[LET] <坐标点变量> = <坐标点表达式>
```

含义 将<坐标点表达式>的值赋值给坐标点变量。
<坐标点表达式>中可使用的运算符仅限四则运算。此外，乘除法只可在进行常量或变量运算时使用。

- 加减法 对每个轴元素进行加减法
- 乘法 将每个轴元素乘以常量 / 变量的值
- 除法 将每个轴元素除以常量 / 变量的值

乘除法的结果因坐标点数据的类型而异。

- 使用脉冲单位时 转换为整数后再赋值。
- 使用毫米单位时 转换为带有小数点后2位的实数后赋值。

SAMPLE

```
P1 = P10 ..... 将坐标点 10 赋值给坐标点 1
P20 = P20 + P5 ..... 将坐标点 20 与坐标点 5 分别加上各个元素，并赋值给 P20
P30 = P30 - P3 ..... 将坐标点 30 至坐标点 3 分别减去各个元素，并赋值给 P30
P80 = P70 * 4 ..... 将坐标点 70 的各元素乘以 4，并赋值给 P80
P60 = P5 / 3 ..... 将坐标点 5 的各元素乘以 1/3，并赋值给 P60
```

MEMO

- 乘除法的使用示例如下。
 - 可以的示例 P15*5, P[E] / A 等
 - 不可以的示例 P10^P11, 3 / P10 等

4. 位移赋值语句

格式

[LET] <位移变量> = <位移表达式>

含义

将<位移表达式>的值赋值给偏移变量。

<位移表达式>中可使用的元素仅限位移元素，运算符仅限加减法。此外，不可使用括号。

- 加减法 对每个轴元素进行加减法

SAMPLE

S1=S0 将位移 0 赋值给位移 1

S2=S1+S0 将位移 1 与位移 0 分别加上每个元素，并赋值给位移 2

MEMO

- <位移表达式>中的加减法示例。

· 可以的示例 S1+S2

· 不可以的示例 S1+3

格式

1. LOm ([b, . . . , b]) = <表达式>
2. LO (mb, . . . ,mb) = <表达式>

值

m : 端口编号.....0、1

b ; 指定比特.....0 ~ 7

<表达式> 整数化后, 对左边指定的指定比特对应的后位比特有效

含义

将指定值输出至 LO 端口, 并进行轴移动的禁止或解除。

LO (00) ~ LO (17) 对应 1 轴 ~ 16 轴。

分配了比特的轴变为机械臂锁定 ON 状态, 轴的移动被禁止。

此外, 指定多个比特时, 请从左边开始降序 (大 → 小) 记述。

MEMO

- 对开始移动的轴有效。

SAMPLE

LO0 () = &B00001010 禁止第 2 轴、第 4 轴的移动

LO0 (2, 1) = &B10 禁止第 3 轴的移动

相关命令

RESET, SET

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

1. LOCx (<坐标点表达式>) = <表达式>
2. LOCx (<位移表达式>) = <表达式>

值

格式 1 : x..... 1 ~ 6 (指定轴), F (手系统标志指定),
F1 (第 1 机械臂转数信息), F2 (第 2 机械臂转数信息)

格式 2 : x..... 1 ~ 4 (指定元素)

<表达式> 指定轴或指定元素时: 坐标值
指定手系统标志时: 1 (右手系统) 或 2 (左手系统)
0 (无指定)
第 1 机械臂转数信息、第 2 机械臂转数信息 (*1)
时: 0、1、-1

*1: 关于第 1 机械臂转数信息、第 2 机械臂转数信息的详细说明, 请参照第 4 章“3 坐标点数据形式”的内容。

含义

直接格式 1: 变更坐标点数据指定轴的数值、手系统标志、第 1 机械臂转数信息或者第 2 机械臂转数信息。

格式 2: 根据位移数据变更指定元素的值。

MEMO

· 要更改值的坐标点, 必须事先录入。要更改未录入的坐标点 (坐标值为空白) 值时, 将出错。

函数

格式

1. LOCx (<坐标点表达式>)
2. LOCx (<位移表达式>)

值

格式 1 : x..... 1 ~ 6 (指定轴), F (手系统标志指定),
F1 (第 1 机械臂转数信息), F2 (第 2 机械臂转数信息)

格式 2 : x..... 1 ~ 4 (指定元素)

含义

格式 1: 根据坐标点数据获取指定轴的数值、手系统标志、第 1 机械臂转数信息或者第 2 机械臂转数信息。

格式 2: 根据位移数据获取指定元素的值。

SAMPLE

LOCX (P10) =A (1) 将 P10 的第 1 轴数据变更为数组 A(1) 的值。

LOCY (S1) =B 将 S1 的第 2 轴数据变更为 B 的值。

A (1) =LOCX (P10) 将 P10 的第 1 轴数据赋给数组 A(1)。

B (2) =LOCX (S1) 将 S1 的第 1 元素 (X 方向) 数据赋给数组 B(2)。

相关命令

坐标点变量、位移变量

格式

LSHIFT (<表达式 1>, <表达式 2>)

含义 将<表达式 1>的比特值仅向左移动<表达式 2>的量。位移后, 空着的位置插入 0。

SAMPLE

A=LSHIFT (&B10111011, 2) 将 &B10111011 向左移 2 比特的值 (&B11101100) 赋值给 A

相关命令 RSHIFT

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

MCHREF (<轴编号>)

值 <轴编号> 1 ~ 6**含义** 赋予<轴编号>指定轴的原点复归动作及绝对查找动作的机器参考(单位:%)。
此函数的对象轴为, 返回原点方式设置为传感方式或撞块方式的轴。

SAMPLE

A=MCHREF (1) 将机器人 1 中第 1 轴的原点复归动作机台参考
赋给变量 A。

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

MID\$ (<字符串表达式> , <表达式 1> [, <表达式 2>])

值 <表达式 1> 1 ~ 255
 <表达式 2> 0 ~ 255

含义 从<字符串表达式>指定的字符串中抽出任意长度的字符串。从字符串中抽出<表达式 1>的第几位字符开始、<表达式 2>的字符数。

当<表达式 1>、<表达式 2>的值超出范围以外时，将出错。

省略了<表达式 2>或从字符串的<表达式 1>的第几位字符开始右侧所有的字符数均小于<表达式 2>时，将从字符串的<表达式 1>的第几位字符开始抽出右侧所有字符。

如果<表达式 1>大于字符串的字符数，MID\$ 的值将变为空字符串。

SAMPLE

B\$=MID\$ (A\$, 2, 4) 将 A\$ 的第 2 字符开始的 4 个字符（最多 5 个字符）赋值给 B\$

相关命令 LEFT\$, RIGHT\$

A

B

C

D

E

F

G

H

I

J

K

L

M



·关于指定比特，请参阅第 3 章“10 指定比特”。

格式

1. MOm ([b, . . . ,b]) = <表达式>
2. MO (mb, . . . ,mb) = <表达式>

值

m : 端口编号..... 2 ~ 7, 10 ~ 17, 20 ~ 27, 30 ~ 37

b : 指定比特..... 0 ~ 7

<表达式> 整数化后，对左边指定的指定比特对应的后位比特有效

含义

此命令语句将指定值输出至 MO 端口。

端口 30 ~ 37 保持各轴的原点传感器状态以及轴的保持状态，因此仅可查看无法输出。（端口 32、33、36、37 为系统保留）

指定多个比特时，请从左边开始降序（大→小）记述。

省略了 [b, . . . , b] 时，8 比特均将成为对象。

· 端口 30、31、34、35 的输出

比特	7	6	5	4	3	2	1	0
端口 30	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
端口 31	16 轴	15 轴	14 轴	13 轴	12 轴	11 轴	10 轴	9 轴
原点传感器状态 0 : 关闭 / 1 : 开启（未连接轴为 1）								
端口 34	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
端口 35	16 轴	15 轴	14 轴	13 轴	12 轴	11 轴	10 轴	9 轴
保持状态 0 : 未保持 / 1 : 保持（未连接轴为 1）								

MEMO

· 关于 MO 端口 30 ~ 37 的详细说明，请参照第 3 章“9.5 内部输出变量”的内容。

SAMPLE

MO2 () = &B10111000..... MO (27, 25, 24, 23) 开启，
MO (26, 22, 21, 20) 关闭

MO2 (6, 5, 1) = &B010..... MO (25) 开启，MO (26, 21) 关闭

MO3 () = 15..... MO (33, 32, 31, 30) 开启，
MO (37, 36, 35, 34) 关闭

MO (37, 35, 27, 20) = A..... 将变量 A 整数化后的后 4 位比特内容输出至
MO (37, 35, 27, 20)

相关命令 RESET, SET

格式

MOTOR	ON
	OFF
	PWR

含义

控制马达电源的开启 / 关闭 F。此时，也可同时控制所有机器人的伺服开启 / 关闭。

- ON.....开启马达电源。也可同时开启所有机器人的伺服。
- OFF.....关闭马达电源。同时关闭所有机器人的伺服，并且进行动态制动。带制动器的轴进行制动并且锁定。
- PWR.....仅开启马达电源。

SAMPLE

MOTOR ON..... 开启马达电源，并且开启所有机器人的伺服。

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

MOVE [< 机器人编号 >] (< 指定轴 > 、 ...)	PTP P L C J	, < 坐标点指定 > , 选项 , 选项 ...
--------------------------------------	-------------------------	---------------------------

含义 执行指定机器人中所设定轴的绝对位置移动命令。

针对其它机器人轴及附加轴所设定的轴，将无法通过该移动命令进行移动。

机器人编号：利用 1 ~ 4 来指定机器人。当省略机器人编号时，机器人 1 被指定。当省略机器人编号时，请不要记述 []。

例：MOVE P, P0...指定机器人 1

指定轴：利用 1 ~ 6 来指定轴(可指定多个)。当省略指定轴时，所有轴将会被指定。

- 移动种类：PTP、直线插补、圆弧插补、连接 PTP
- 坐标点指定：直接数值指定、坐标点定义指定
- 选项：速度指定、圆弧插补移动指定、STOPON 条件指定、CONT 指定、加速度指定、减速度指定、坐标平面指定、端口输出指定(可指定多个)
可以省略选项。

选配件	PTP	连接 PTP	直线插补	圆弧插补	备注
速度指定 (SPEED)	○	○	○	○	仅指定 MOVE 语句有效
速度指定 (VEL)	×	×	○	○	仅指定 MOVE 语句有效
圆弧插补移动	○	○	×	×	仅指定 MOVE 语句有效
STOPON 条件指定	○	○	○	×	仅程序执行时有效
CONT 指定	○	○	○	○	仅指定 MOVE 语句有效
加速度指定	○	○	○	×	仅指定 MOVE 语句有效
减速度指定	○	○	○	×	仅指定 MOVE 语句有效
平面坐标指定	×	×	×	○	仅指定 MOVE 语句有效
端口输出指定	×	×	○	○	仅指定 MOVE 语句有效

移动种类

● PTP (Point to Point) 移动

执行开始条件：所有的移动对象轴均结束移动(到达公差范围内)。

执行结束条件：所有的移动对象轴均到达 OUT 有效位置范围内。

移动对象轴同时到达。无法保证移动的轨迹。

● 继 MOVE P 命令之后的命令注意点

MOVE P 命令的下一个命令为信号输出命令等可执行命令时，移动对象轴进入 OUT 有效位置范围内后，将开始执行下一个命令。即，在轴到达目标位置的公差范围内之前，已开始下一个命令。

示例：

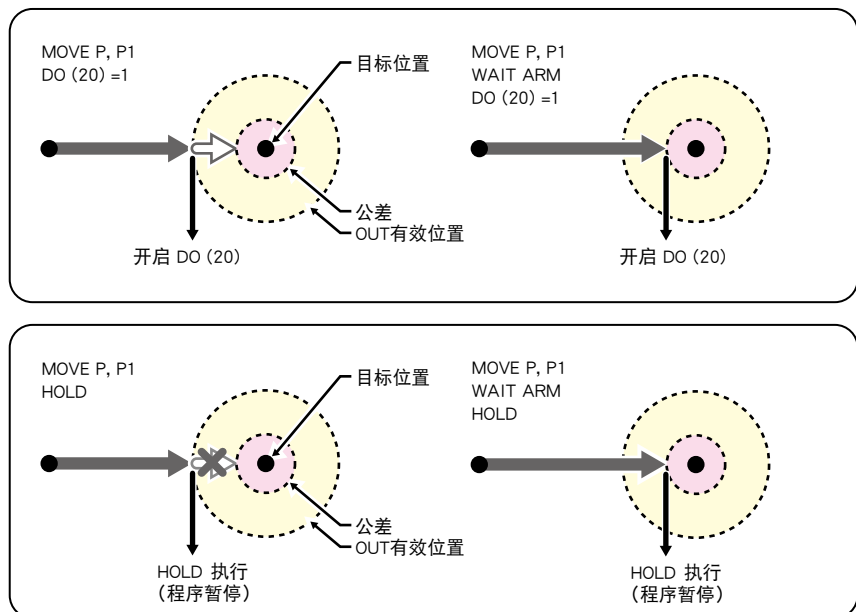
信号输出 (DO 等)	进入 OUT 有效位置范围内后，将输出信号。
DELAY	进入 OUT 有效位置范围内后，将执行 DELAY 命令，并开始等待时间。
HALT	进入 OUT 有效范围内后，程序停止并被复位。因此，移动将停止。
HALTALL	当进入 OUT 有效位置范围以内时，正在执行的程序将全部停止，任务 1 被复位，其它任务结束。因此，移动也会停止。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
HOLDALL	当进入 OUT 有效位置范围以内时，正在执行的程序将全部暂停。因此，移动也会停止。
WAIT	进入 OUT 有效位置范围内后，将执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时，请使用 "WAIT ARM" 语句。

MEMO

- OUT 有效位置的数值由参数设置。
也可在程序中通过 OUTPOS 命令更改值。

MOVE命令



35823-R7-00

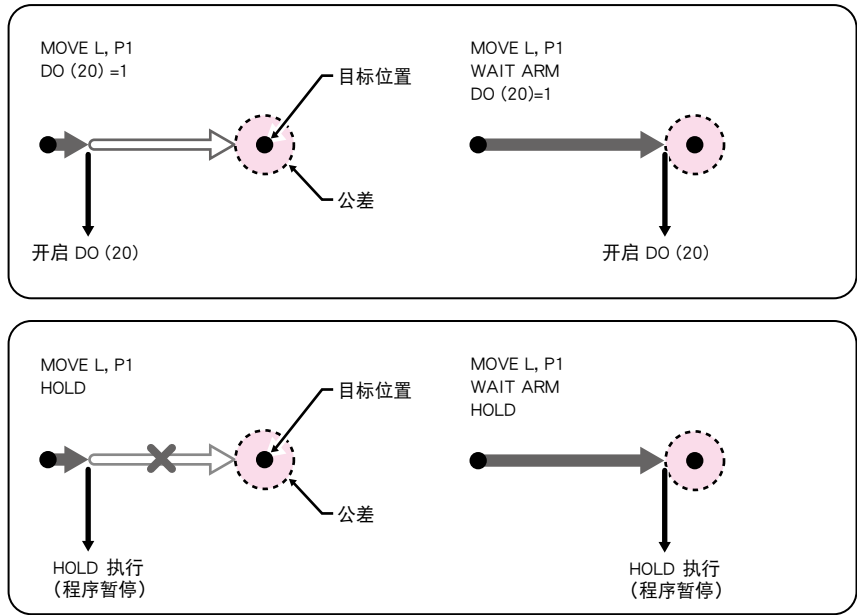
SAMPLE

MOVE P, P0 主机器人轴机器人 1 从当前位置向 P0 所指定位置进行移动 (MOVE PTP, P0 也一样)

MEMO

· PTP 移动比插补移动动作速度快, 但是当连续移动多个坐标点时, 将每次进行定位并停止。

MOVE命令

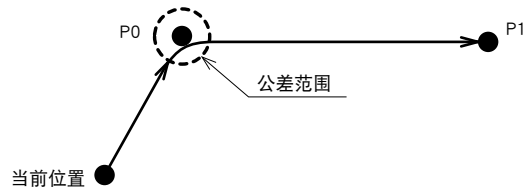


35824-R7-00

SAMPLE

MOVE L, P0, P1 机器人 1 从当前位置向 P0、P1 所指定的位置进行移动。

SAMPLE: MOVE L



35810-R7-00

- 圆弧插补移动

执行开始条件：所有的移动对象轴均结束动作（到达公差范围内）。

执行结束条件：所有的移动对象轴已开始动作。

紧接着记述的命令在轴开始移动后将立即执行。

连续执行直线插补移动或圆弧插补移动时，2个移动路径通过连接减速与加速部分来联结，将继续移动，中途不停止。

所有移动对象轴均同时到达。

进行圆弧插补时，将当前位置、中间位置与目标位置这3个位置形成1个圆弧。**因此，必须指定偶数个数进行坐标点指定。**

- MOVE C 命令之后的命令注意点

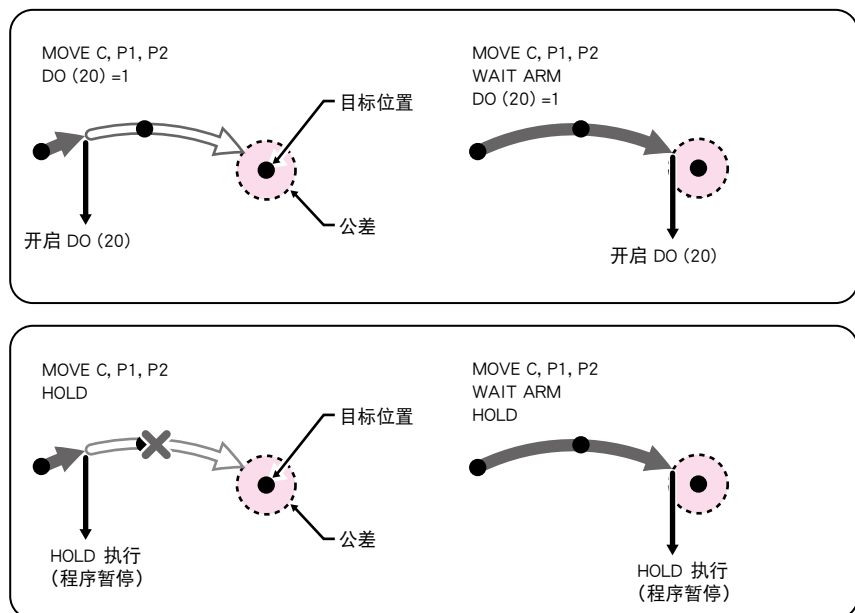
MOVE C 命令的下一个命令为信号输出命令等可执行命令时，将从移动开始后立即开始执行。

示例：

信号输出（DO 等）	移动开始后，将立即输出信号。
DELAY	移动开始后，将立即执行 DELAY 命令，并开始等待时间。
HALT	移动开始后，程序立即停止，并被复位。因此，移动将停止。
HALTALL	当进入 OUT 有效位置范围以内时，正在执行的程序将全部停止，任务 1 被复位，其它任务结束。因此，移动也会停止。
HOLD	移动开始后，程序立即暂停。因此，移动将停止。
HOLDALL	当进入 OUT 有效位置范围以内时，正在执行的程序将全部暂停。因此，移动也会停止。
WAIT	移动开始后，将立即执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时，请使用“WAIT ARM”语句。

MOVE命令

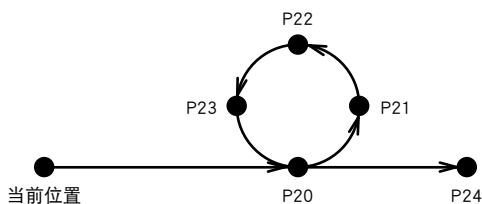


35825-R7-00

SAMPLE

MOVE L, P20 从当前位置向 P20 直线移动
 MOVE C, P21, P22, P23, P20 进行由 P21, P22, P23, P20 构成的圆弧移动
 MOVE L, P24 向 P24 直线移动

SAMPLE: MOVE C



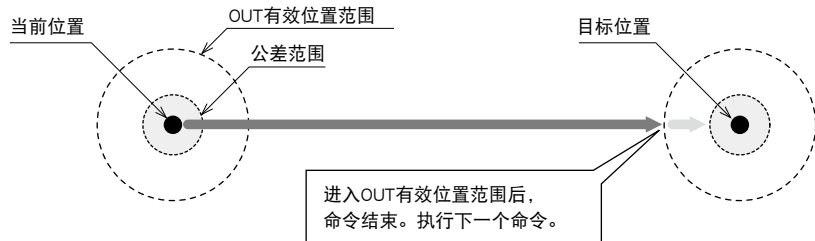
35811-R7-00

MEMO

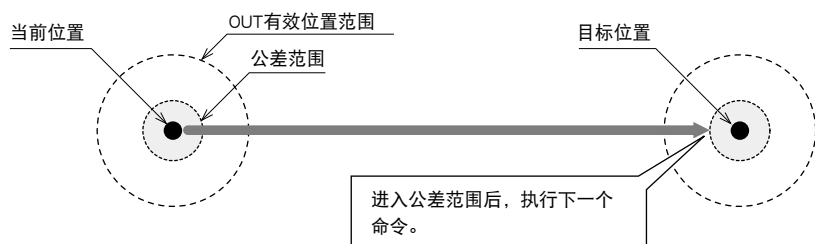
- 即使插补移动采用连续动作时，也不会在中途的坐标点停止。但是，最高速度低于 PTP。
- 可进行圆弧插补的范围为半径 1.00mm ~ 5,000.00mm。
- 根据速度、加速度、坐标点间距离不同，圆弧可能会发生歪斜。
- 对于有 R 轴指定的机器人，根据 R 轴的移动距离不同，可能会由于 R 轴速度变得过快而出错。
- 在 MOVE L 之后执行 DELAY 语句时，执行 MOVE L 后，DELAY 的定时器立即动作。因此，**要在到达目标坐标点后执行 DELAY 时，请在 MOVE 语句的后面使用 "WAIT ARM" 语句。** HALT 命令等其他命令也同样处理。
- 对于 MULTI 型机器人，将发生 [5.37: Specification mismatch] 的错误，无法执行圆弧插补。
- 插补移动中，如果进行锐角方向转换，连接部分的加减速度可能会变快而导致出错。此时，请降低连接部分的加减速度或重新使用 WAIT ARM 命令等修改动作形式。

移动命令动作的区别

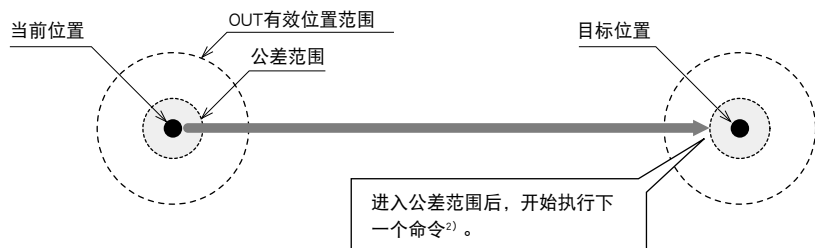
1. PTP动作



2. WAIT ARM



3. STOPON条件表达式



1) 执行插补动作之后的 DELAY 语句时, 移动开始后立即启动 DELAY 的定时器。

在到达目标位置后要执行 DELAY 时, 请在 MOVE 的后面插入 WAIT ARM。

2) 中途 STOPON 条件表达式成立时, 将减速停止。(详细说明, 请参阅 <STOPON 条件指定>。)

35703-R9-00

坐标点指定的种类

● 直接数值指定

PTP

连接 PTP

直线插补

圆弧插补

格式

p1 p2 p3 p4 p5 p6 [f][f1][f2]

值

p1 ~ p6.....用空格隔开的各轴坐标值
 F.....手系统标志 (仅限水平多关节型机器人)
 f1.....第 1 机械臂转数信息 (仅限于 YK500TW)
 f2.....第 2 机械臂转数信息 (仅限于 YK500TW)

含义

指定坐标值。此时的数值为整数时，判断为脉冲单位；数值为实数值（带有小数点）时，判断为毫米 / 度单位并移动。整数与实数同时存在时，所有的坐标值将被作为毫米 / 度单位处理。

可进行该指定的移动种类包括 PTP 移动、连接 PTP 移动以及直线插值移动。

对水平多关节型机器人直接指定数值并以毫米单位指定坐标值时，可指定手系统标志。

在水平多关节型机器人上设置扩展设置的手系统标志时，请对 f 指定 1 或 2。1 及 2 以外的数值或未指定数值时，将视作无手系统标志设定 (0)。

- 1: 表示设置为在右手系统中向指定位置移动。
- 2: 表示设置为在左手系统中向指定位置移动。

在直接指定数值过程中，当为 YK500TW 且以毫米为单位来指定坐标值时，可以指定第 1 机械臂转数信息及第 2 机械臂转数信息(*1)。

在 YK500TW 机器人中，当指定扩展指定的第 1 机械臂转数信息及第 2 机械臂转数信息时，在 f1 及 f2 可指定 -1、0、1 中的任何一个。

如果指定了 -1、0、1 以外的数值或者未设定第 1 机械臂转数信息以及第 2 机械臂转数信息时，则作为 0 进行处理。

- 0: 表示将机械臂转数信息设定为向 0 的位置进行移动。
- 1: 表示将机械臂转数信息设定为向 1 的位置进行移动。
- 1: 表示将机械臂转数信息设定为向 -1 的位置进行移动。

*1: 关于第 1 机械臂转数信息、第 2 机械臂转数信息的详细说明，请参照第 4 章“3 坐标点数据形式”的内容。

要点

- 同时存在两种指定单位时（整数与实数同时存在时），所有值被视作毫米 / 度单位。

注意

- 指定手系统标志进行直线插补时，请使当前位置的手系统与指定移动目标位置的手系统一致。手系统不同时，将出错并无法动作。
- 当进行直线插值时，请使当前位置下的第 1 机械臂转数信息、第 2 机械臂转数信息和指定移动目标下的第 1 机械臂转数信息、第 2 机械臂转数信息保持一致。如果这些信息不同，则会发生错误，并导致无法动作。

MEMO

- 水平多关节型机器人且移动目标的坐标数据中指定了手系统标志时，指定的手系统优先于当前机械臂类型或 LEFTY / RIGHTY 的指定。

SAMPLE

```
MOVE P, 10000 10000 1000 1000 0 0
```

..... 从当前位置向指定位置进行 PTP 移动



注意

- 向设置了手系统标志的坐标点进行直线插补或圆弧插补移动时，请使当前位置的手系统与指定坐标点的手系统一致。当前位置的手系统与指定坐标点的手系统不同时，将出错并无法动作。



MEMO



注意

- 当通过直线插值或者圆弧插值进行移动时，请使当前位置下的第 1 机械臂转数信息、第 2 机械臂转数信息和指定坐标点位置下的第 1 机械臂转数信息、第 2 机械臂转数信息保持一致。如果这些信息不同，则会发生错误，并导致无法动作。

● 坐标点定义指定

PTP

连接 PTP

直线插补

圆弧插补

格式

< 坐标点表达式 > [, < 坐标点表达式 > ...]

含义

指定 < 坐标点表达式 >。通过用逗号 (,) 隔开可指定多个数据。
对于圆弧插补，需指定偶数个数的坐标点。

- 水平多关节型机器人且移动目标的坐标数据中指定了手系统标志时，指定的手系统优先于当前机械臂类型或 LEFTY / RIGHTY 的指定。

SAMPLE

MOVE P, P1 从当前位置向 P1 指定的位置移动
MOVE P, P20, P0, P100 从当前位置向 P20, P0, P100 指定的位置依次移动

A

B

C

D

E

F

G

H

I

J

K

L

M

选项种类

● 速度指定 1

PTP

连接 PTP

直线插补

圆弧插补

格式

1. SPEED = <表达式>
2. S = <表达式>

值 <表达式>1 ~ 100(单位: %)

含义 使用<表达式>指定程序移动速度。

运行速度如下所示。

· 机器人的最高速度(mm / 秒) × 自动移动速度(%) × 程序移动速度(%)

仅对指定的 MOVE / MOVE2 语句有效。

SAMPLE

MOVE P, P10, S=10 从当前位置按照 10% 的程序移动速度向 P10 指定的位置移动

● 速度指定 2

PTP

连接 PTP

直线插补

圆弧插补

格式

VEL = <表达式>

值 <表达式>水平多关节型机器人: 1 ~ 750
XY 型机器人: 1 ~ 1000(单位: mm / 秒)

含义 在 mm / 秒单位的<表达式>中执行机器人的 XYZ 轴合成最高速度。针对水平多关节型机器人或者 XY 型机器人, 当移动种类为直线插值移动以及圆弧插值移动时, 可以指定该选项。

仅对指定的 MOVE 语句有效。

SAMPLE

MOVE L, P10, VEL=100 以 XYZ 属性轴的合成最高速度 100mm / 秒, 从当前位置向 P10 所指定的位置进行移动。



要点

- 虽然规定了最高速度, 但并不保证按照指定速度移动。



要点

- 虽然规定了合成速度的最高速度, 但并不保证按照指定速度移动。

● STOPON 条件指定

PTP

连接 PTP

直线插补

圆弧插补

格式

STOPON < 条件表达式 >

含义

移动中条件表达式不成立时，移动将停止。此时的停止动作为减速停止，因此在条件成立后将产生减速移动距离。

移动开始时如果条件表达式成立，将不移动，直接结束命令。

可指定此移动的种类为 PTP 移动及直线插补移动。

仅程序执行时有效。

SAMPLE

```
MOVE P, P100, STOPON DI (20) =1
```

..... 从当前位置移动至 P100 指定的位置。如果中途 DI (20) =1 的条件成立，将减速停止移动并执行下一个步骤。

MEMO

- 在指定 STOPON 条件中使用的条件表达式表示数值的情况下，当表达式的值非 0 时为“真”，当成为 0 时则为“假”。

● CONT 指定

PTP

连接 PTP

直线插补

圆弧插补

格式

CONT



要点

- 通过指定 CONT，可缩短 PTP 移动开始时的定位时间。
- 无法保证路径。

含义

执行了 CONT 所指定的移动后，当所有的移动对象轴均进入 OUT 有效位置范围内时，在移动对象轴的动作结束前（到达公差范围内）便开始移动。仅对指定 MOVE 语句有效。

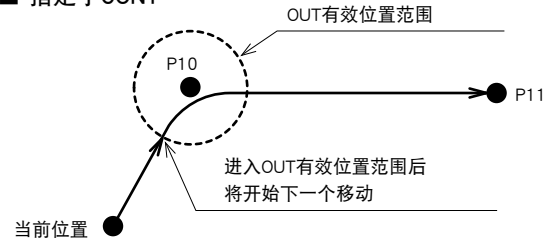
SAMPLE

MOVE P, P10, P11,CONT

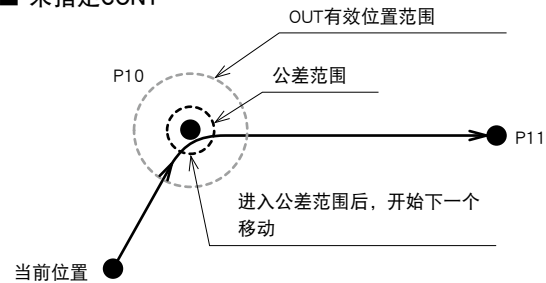
.....从当前位置移动至 P10 指定的位置，不等移动轴到达公差范围内，就开始向 P11 移动

SAMPLE: MOVE CONT

■ 指定了CONT



■ 未指定CONT



35814-R7-00

● 加速度指定

PTP

连接 PTP

直线插补

圆弧插补

格式

ACC = <表达式>

值 <表达式>1 ~ 100(单位: %)

含义 在<表达式>中指定由加速度系数参数决定的相对于机器人加速度的比率, 并决定动作时的加速度。

可指定此移动的种类为直线插补。

仅对指定的 MOVE 语句有效。

SAMPLE

MOVE L, P100, ACC=10从当前位置按照 10% 的加速度比率向 P100 指定的位置移动

● 减速度指定

PTP

连接 PTP

直线插补

圆弧插补

格式

DEC = <表达式>

值 <表达式>1 ~ 100(单位: %)

含义 在<表达式>中指定由加速度系数参数决定的相对于机器人加速度的比率, 并决定动作时的减速度(将设置的加速度作为 100%, 为相对于该速度的比率)。

可指定此移动的种类为直线插补。

仅对指定的 MOVE 语句有效。

SAMPLE

MOVE L, P100, DEC=20从当前位置按照 20% 的减速度比率向 P100 指定的位置移动

● 坐标平面指定

PTP

连接 PTP

直线插补

圆弧插补

格式

XY
YZ
ZX

值	XY.....XY 坐标平面
	YZ.....YZ 坐标平面
	ZX.....ZX 坐标平面

含义 执行坐标指定的圆弧插补时，向指定坐标平面的投射执行为像圆形的圆弧插补。可指定此移动的种类为圆弧插补移动。仅对指定的 MOVE 语句有效。

SAMPLE

```
P10 = 100.00 100.00 20.00 0.00 0.00 0.00
P11 = 150.00 100.00 0.00 0.00 0.00 0.00
P12 = 150.00 150.00 20.00 0.00 0.00 0.00
P13 = 100.00 150.00 40.00 0.00 0.00 0.00
MOVE P, P10
MOVE C, P11, P12
MOVE C, P13, P10.....连续移动在 P10、P11、P12 与 P12, P13, P10
                        上形成的 3 维圆弧
MOVE C, P11, P12, XY
MOVE C, P13, P10, XY.....连续移动在 P10、P11、P12 与 P12、P13、
                        P10 上形成的 XY 平面圆弧。此时 Z 轴向成为
                        圆弧目标位置并在 P12 与 P10 的指定位置移动。
```



要点

- 不指定坐标平时，路径将变为 3 维圆弧。
- 对于 2 轴规格的机器人，将变为 XY 平面上的圆弧动作。

A

B

C

D

E

F

G

H

I

J

K

L

M

● 端口输出指定

PTP

连接 PTP

直线插补

圆弧插补

格式 1

DO		m ([b, . . . , b])= <表达式 1 > @ <表达式 2 >
MO		
SO		

格式 2

DO		(mb, . . . , mb) = <表达式 1 > @ <表达式 2 >
MO		
SO		

值 m : 端口编号2 ~ 7, 10 ~ 17, 20 ~ 27

b : 指定比特0 ~ 7

<表达式 1 >输出至指定端口的值, 仅整数部分有效。

<表达式 2 >进行端口输出的位置。最多可在 mm 单位中指定至小数点以下 2 位。

含义

执行直线插补或圆弧插补时, 在仅离开移动开始位置 <表达式 2 > 距离 (单位: mm) 的时候将 <表达式 1 > 的值输出至指定的输出端口。

表达式 2 中指定的数值表示以移动开始点为中心的圆的半径。

可指定此移动的种类为直线插补移动及圆弧插补移动。

此选项在 1 个 MOVE 语句中可指定 2 次。

指定多个比特时, 请从左边开始降序 (大→小) 记述。

形式 1 中省略了 [b, . . . , b] 时, 8 个比特均将成为对象。

指定了硬件上不存在的端口时, 将无任何输出。

SAMPLE 1

```
MOVE P, P0
```

```
MOVE L, P1, DO2 ( ) =105 @ 25.85
```

.....向 P1 进行直线插补移动过程中, 在离开 P0
25.85 mm 的时候向 DO2 () 输出 105
(&B01101001)

SAMPLE 2

```
A!=10
```

```
B!=20
```

```
MOVE L, P2, MO (22) =1 @ A!, MO (22) =0 @ B!
```

.....开始向 P2 移动后, 在离开 10 mm 的时候开启
MO (22), 并在离开 20 mm 的时候关闭 MO
(22)

相关命令 MOVEI, DRIVE, DRIVEI, WAIT ARM

**注意**

· DO, MO, SO 均无法输出至端口 0 和 1。



参考

· 关于指定比特, 请参阅第 3 章“10 指定比特”。

格式

```
MOVEI | PTP | , <坐标点指定> , [, 选项 [, 选项] ...]
      | P      |
```

含义 执行机器人的相对位置移动命令。

MOVEI 以设定的所有轴为对象。

设置为其他组或附加轴的轴使用此移动命令时将不移动。

- 移动种类： PTP
- 坐标点指定： 直接数值指定、坐标点定义指定
- 选项： 速度指定

MEMO

· 中途停止 MOVEI 语句并重新启动时，可在控制器的 "MOVEI / DRIVEI start position" 中选择移动的目标位置。详细说明，请参阅控制器的手册。

- 1) 保持 (初始值) 继续执行之前的移动。目标位置与再次执行前相同。
- 2) 复位 从当前位置开始重新进行相对移动。目标位置与再次执行前不同 (与之前位置互换)。

移动种类

● PTP (Point to Point) 移动

执行开始条件：所有的移动对象轴均终止移动 (到达公差范围内)。

执行结束条件：所有的移动对象轴均到达有效位置范围内。

移动对象轴将同时到达。无法保证移动的轨迹。

● MOVEI 命令之后的命令注意点

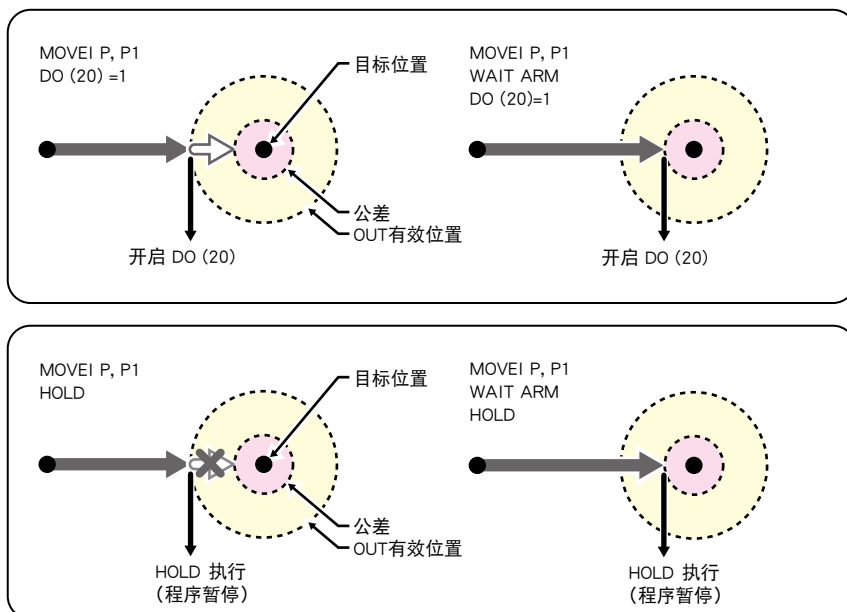
MOVEI 命令的下一个命令为信号输出命令等可执行命令时，移动对象轴进入 OUT 有效位置范围内后，将开始执行下一个命令。即，在轴到达目标位置的公差范围之内之前，将开始下一个命令。

示例：

信号输出 (DO 等)	进入 OUT 有效位置范围内后，将输出信号。
DELAY	进入 OUT 有效位置范围内后，将执行 DELAY 命令，并开始等待时间。
HALT	进入 OUT 有效范围内后，程序停止并被复位。因此，移动将停止。
HALTALL	当进入 OUT 有效位置范围以内时，正在执行的程序将全部停止，任务 1 被复位，其它任务结束。因此，移动也会停止。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
HOLDALL	当进入 OUT 有效位置范围以内时，正在执行的程序将全部暂停。因此，移动也会停止。
WAIT	进入 OUT 有效位置范围内后，将执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时, 请使用 "WAIT ARM" 语句。

MOVEI命令



35826-R7-00

SAMPLE

MOVEI P, P0 主机器人轴向相对当前位置加上 P0 中指定的移动量位置进行 PTP 移动

坐标点指定的种类

● 直接数值指定

PTP

格式

p1 p2 p3 p4 p5 p6 [f] [f1] [f2]

值

p1 ~ p6 用空格隔开的各轴坐标值
 f 手系统标志 (仅限水平多关节型机器人)
 f1 第 1 机械臂转数信息 (仅限于 YK500TW)
 f2 第 2 机械臂转数信息 (仅限于 YK500TW)

含义

直接指定坐标值。此时的数值为整数时，判断为脉冲单位；数值为实数值时，判断为毫米 / 度单位并移动。

对水平多关节型机器人直接指定数值并以毫米单位指定坐标值时，可指定手系统标志。

在水平多关节型机器人上设置扩展设置的手系统标志时，请对 f 指定 1 或 2。1 及 2 以外的数值或未指定数值时，将视作无手系统标志设定 (0)。

- 1: 表示设置为在右手系统中向指定位置移动。
- 2: 表示设置为在左手系统中向指定位置移动。

在直接指定数值过程中，当为 YK500TW 且以毫米为单位来指定 YK500TW 坐标值时，可以指定第 1 机械臂转数信息及第 2 机械臂转数信息 (*1)。

在 YK500TW 机器人中，当指定扩展指定的第 1 机械臂转数信息以及第 2 机械臂转数信息时，在 f1 及 f2 可指定 -1、0、1 中的任何一个。

如果指定了 -1、0、1 以外的数值或者未设定第 1 机械臂转数信息以及第 2 机械臂转数信息时，则作为 0 进行处理。

- 0: 表示将机械臂转数信息设定为向 0 的位置进行移动。
 1: 表示将机械臂转数信息设定为向 1 的位置进行移动。
 -1: 表示将机械臂转数信息设定为向 -1 的位置进行移动。

*1: 关于第 1 机械臂转数信息、第 2 机械臂转数信息的详细说明，请参照第 4 章“3 坐标点数据形式”的内容。



要点

- 同时存在两种指定单位时 (整数与实数同时存在时)，所有值被视作毫米 / 度单位。

MEMO

- 水平多关节型机器人且移动目标的坐标数据中指定了手系统标志时，指定的手系统优先于当前机械臂类型或 LEFTY / RIGHTY 的指定。

SAMPLE

```
MOVEI P, 10000 10000 1000 1000 0 0
```

..... 向仅加上相对当前位置的指定量 (脉冲单位) 位置进行 PTP 移动

- 坐标点定义指定

PTP

格式

<坐标点表达式> [, <坐标点表达式> ...]

含义 指定<坐标点表达式>。通过用逗号(,)隔开可指定多个数据。

MEMO

- 水平多关节型机器人且移动目标的坐标数据中指定了手系统标志时，指定的手系统优先于当前机械臂类型或 LEFTY / RIGHTY 的指定。

SAMPLE

MOVEI P, P1 从当前位置仅进行 P1 指定移动量的 PTP 移动

选项种类

● 速度指定

PTP

格式

1. SPEED = <表达式>
2. S = <表达式>

值 <表达式>1 ~ 100(单位: %)

含义 使用<表达式>指定程序移动速度。

实际速度如下所示。

- 机器人的最高速度(mm / 秒) × 自动移动速度(%) × 程序移动速度(%)
- 仅对指定的 MOVEI 语句有效。

SAMPLE

MOVEI P, P10, S=10 从当前位置按照 10% 的程序移动速度仅移动 P1 指定的移动量

相关命令 MOVE, DRIVE, DRIVEI, WAIT ARM



要点

- 虽然规定了最高速度，但并不保证按照指定速度移动。

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

```
OFFLINE [ | ETH | ]
         [ | CMU | ]
```

值 <表达式> ETH、CMU 或无指定

含义 更改指定端口的通信模式参数，并使通信模式脱机。

ETH 将 Ethernet 的通信模式参数更改为脱机，并清除收发缓冲区。

CMU 将 RS-232C 的通信模式参数更改为脱机，并清除通信错误的解除与接收缓冲区。

无指定 将 Ethernet 以及 RS-232C 的通信模式参数变更为脱机，并且清除收发信缓存。

SAMPLE

```
OFFLINE
SEND CMU TO A$
SEND CMU TO P10
ONLINE
HALT
```


格式

ON <表达式> GOSUB <标签 1> [, <标签 2> ...] ※GOSUB 也可写成 GO SUB

值 <表达式> 0 或正整数

含义 <表达式> 的值决定程序的跳转目标位置。
 <表达式> 的值为 1 时, 将跳转至第 1 位的 <标签 1>; 值为 2 时, 跳转至第 2 位的 <标签 2>。

以此类推, <表达式> 的值为 n 时, 将跳转至第 n 位的 <标签 n>。

<表达式> 的值为 0 或者超过了标签数时, 将不跳转, 直接执行下一个命令。

执行跳转目标位置的子例程后, 将执行 ON ~ GOSUB 语句的下一个命令。

SAMPLE

```
' MAIN ROUTINE
*ST:
ON DI3 ( ) GOSUB *SUB1, *SUB2, *SUB3 ..... 执行 *SUB1 ~ *SUB3
GOTO *ST ..... 返回 *ST
HALT
' SUB ROUTINE
*SUB1:
  MOVE P, P10, Z=0
  RETURN
*SUB2:
  DO (30) = 1
  RETURN
*SUB3:
  DO (30) = 0
  RETURN
```

相关命令 GOSUB, RETURN

格式

ON <表达式> GOTO <标签 1> [, <标签 2> ...] ※GOTO 也可写成 GO TO

值 <表达式>0 或正整数

含义 <表达式> 的值决定程序的跳转目标位置。
 <表达式> 的值为 1 时，将跳转至第 1 位的 <标签 1>，值为 2 时，跳转至第 2 位的 <标签 2>。

以此类推，<表达式> 的值为 n 时，将跳转至第 n 位的 <标签 n>。

<表达式> 的值为 0 或者超过了标签数时，将不跳转，直接执行下一个命令。

SAMPLE

```
' MAIN ROUTINE
*ST:
ON DI3 ( ) GOTO *L1,*L2,*L3 ..... 根据 DI3 ( ) 的值跳转至 *L1 ~ *L3
GOTO *ST ..... 返回 *ST
HALT
' SUB ROUTINE
*L1:
  MOVE P, P10, Z=0
  GOTO *ST
*L2:
  DO (30) = 1
  GOTO *ST
*L3:
  DO (30) = 0
  GOTO *ST
```

相关命令 GOTO

格式

```
ONLINE [ | ETH | ]
        [ | CMU | ]
```

值 <表达式>ETH、CMU 或无指定

含义 更改指定端口的通信模式参数，并使通信模式联机。

ETH 将 Ethernet 的通信模式参数更改为联机，并清除收发缓冲区。

CMU 将 RS-232C 的通信模式参数更改为联机，并清除通信错误的解除与接收缓冲区。

无指定 将 Ethernet 以及 RS-232C 的通信模式参数变更为联机，并且解除通信错误和清除接收缓存。

SAMPLE

```
OFFLINE
SEND CMU TO A$
SEND CMU TO P10
ONLINE
HALT
```


格式

ORD (< 字符串表达式 >)

含义 计算 < 字符串表达式 > 起始字符的字符编码。

SAMPLE

A=ORD ("B") 将 66 (= &H42) 赋值给 A

相关命令 CHR\$

格式

ORGORD [< 机器人编号 >] < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 表达式 > n ~ nnnnnn (n : 0 ~ 6)

含义 设定针对 < 机器人编号 > 指定机器人进行原点归复动作的轴顺序参数。
 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
 指定时，分别用 1 ~ 6 表示第 1 轴 ~ 第 6 轴，并用 1 位 ~ 6 位的整数指定。
 不可对同一个轴重复指定。
 按照从左至右的顺序进行指定轴的返回原点后，剩余的轴全部同时返回原点。
 值为 0 时，所有的轴同时返回原点。

函数

格式

ORGORD [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 获取针对 < 机器人编号 > 指定机器人进行原点归复动作的轴顺序参数。
 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

SAMPLE

```
A=3
ORGORD A ..... 指定为第 3 轴先返回原点
ORIGIN ..... 机器人 1 中第 3 轴的原点归复动作完成之后，
              执行其余轴的同步原点归复动作。

MOVE P, P0
A=ORGORD ..... 将主组的返回原点顺序参数赋值给变量 A
HALT
```

相关命令 ORIGIN

格式

ORIGIN

含义

执行机器人增量式马达轴的返回原点动作。对于半绝对式规格的轴执行绝对位置搜索动作。

如果中途停止，将变为未返回原点状态。

当设定了多台机器人时，机器人 1 的原点归复动作结束之后，依次进行机器人 2 ~ 4 的原点归复动作。

SAMPLE

ORIGIN 进行原点归复动作

相关命令

ORGORD, MCHREF

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

OUT	DOm ([b, . . . , b]) DO (mb, . . . , mb) MOm ([b, . . . , b]) MO (mb, . . . , mb) SOm ([b, . . . , b]) SO (mb, . . . , mb) LO0 ([b, . . . , b]) LO (0b, . . . , 0b) TO0 ([b, . . . , b]) TO (0b, . . . , 0b)	[, <时间>]
-----	---	-----------

值 m : 端口编号..... 2 ~ 7, 10 ~ 17, 20 ~ 27
b : 指定比特..... 0 ~ 7
<表达式> 1 ~ 3600000 (单位 : ms)。

含义 开启指定端口的输出，并结束命令语句（进入程序下一行）。之后，经过<表达式>指定的时间后，关闭此端口的输出。中途暂停时，重新启动后将在经过<表达式>中指定时间的剩余时间后，关闭此端口的输出。

省略了<表达式>时，指定端口的输出将仍然开启着。

指定了<表达式>的 OUT 语句，最多可同时执行 16 个。如果要执行 17 个以上，将发生 [6.26: No sufficient memory for OUT] 的错误。

指定多个比特时，请从左边开始降序（大→小）记述。

指定了硬件上不存在的端口时，将无任何输出。

SAMPLE

OUT DO2 (), 200 DO (27 ~ 20) 开启，200 ms 后关闭
OUT DO (37, 35, 27, 20)..... DO (37, 35, 27, 20) 开启

相关命令 DO, MO, SO, TO, LO



注意

· DO, SO 均无法输出至端口 0 和 1。



参考

· 关于指定比特，请参阅第 3 章“10 指定比特”。

格式

1. OUTPOS [< 机器人编号 >] < 表达式 >
2. OUTPOS [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 1 ~ 6144000 (单位 : 脉冲)

含义 将 < 机器人编号 > 指定机器人轴的参数的 OUT 有效位置变更为 < 表达式 > 的值。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
 形式 1 对指定机器人设定的所有轴进行变更。
 格式 2 仅更改 < 轴编号 > 中指定的轴。

 MEMO

- 在生成过程中，如果指定 no axis 中所设定的轴时，则会发生“5.37 : 规格不一致”错误，并且中止执行。

函数

格式

OUTPOS [< 机器人编号 >] (< 轴编号 >)

值

< 机器人编号 > 1 ~ 4
 < 轴编号 1 > 1 ~ 6

含义 获取 < 机器人编号 > 指定机器人轴之中 < 表达式 > 指定轴的 OUT 有效位置参数值。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

 MEMO

- 在生成过程中，如果指定 no axis 中所设定的轴时，则会发生“5.37 : 规格不一致”错误，并且中止执行。

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

SAMPLE

```
' CYCLE WITH DECREASING OUTPOS
DIM SAV (3)
GOSUB *SAVE_OUTPOS
FOR A=1000 TO 10000 STEP 1000
    GOSUB *CHANGE_OUTPOS
    MOVE P, P0
    DO3 (0) =1
    MOVE P, P1
    DO3 (0) =0
NEXT A
GOSUB *RESTORE_OUTPOS
HALT
*CHANGE_OUTPOS:
    FOR B=1 TO 4
        OUTPOS (B) =A
    NEXT B
    RETURN
*SAVE_OUTPOS:
    FOR B=1 TO 4
        SAV (B-1) =OUTPOS (B)
    NEXT B
    RETURN
*RESTORE_OUTPOS:
    FOR B=1 TO 4
        OUTPOS (B) =SAV (B-1)
    NEXT B
    RETURN
```

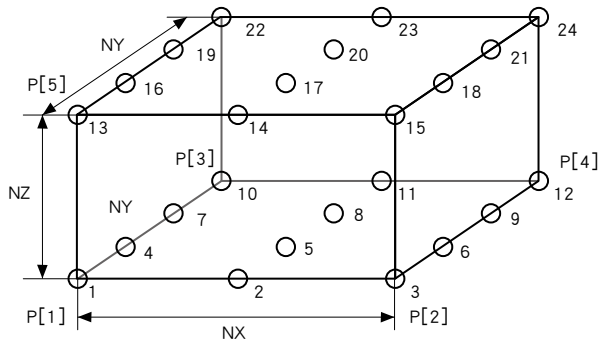
格式

PDEF (< 托盘定义编号 >) = < 表达式 1 > , < 表达式 2 > [, < 表达式 3 >]

- 值**
- < 托盘定义编号 > 0 ~ 39
 - < 坐标点指定 > 托盘定义中所使用的坐标点。使用以指定坐标点作为开头的连续 5 个坐标点。
 - < 表达式 1 > P[1] - P[2] 之间的坐标点数 (NX)
 - < 表达式 2 > P[1] - P[3] 之间的坐标点数 (NY)
 - < 表达式 3 > P[1] - P[5] 之间的坐标点数 (NZ)
- 总坐标点数 : < 表达式 1 > × < 表达式 2 > × < 表达式 3 > 必须为 32767 以下。
P[1] ~ P[5] 的定义请参照下图。

- 含义**
- 定义用于执行托盘移动命令的托盘。
 - 可对已定义的托盘数据更改分割条件。
 - 当指定了各轴的坐标点数时, 将按照下图顺序对各轴等间隔指定数对应坐标点进行自动计算并定义。
 - 省略了 < 表达式 3 > (Z 轴方向) 时, 高度方向变为 1。
 - 各托盘的总坐标点数必须控制在 32,767 以下。

坐标点的自动计算



35815-R7-00

SAMPLE

PDEF (1) = P3991,3,4,2..... 使用 P3991 ~ P3995 将托盘定义 1 定义为
3 × 4 × 2。

格式

PMOVE [< 机器人编号 >] (< 托盘定义编号 > , < 托盘位置编号 >)
[, 选项 [, 选项] ...]

值

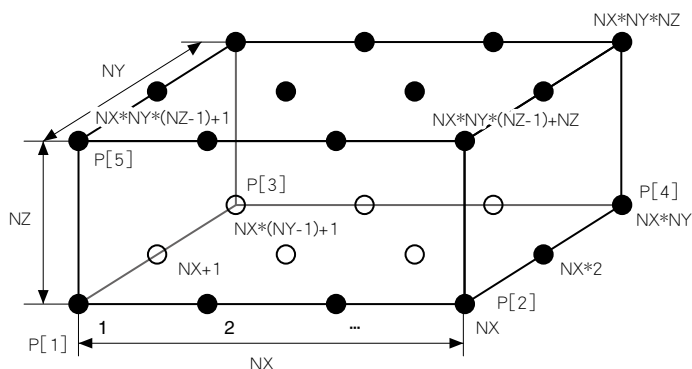
- < 机器人编号 > 1 ~ 4
- < 托盘定义编号 > 0 ~ 39
- < 托盘位置编号 > 1 ~ 32767

含义 执行机器人轴的托盘移动命令 (必须事先设置要指定的托盘编号)。
在 PMOVE 中设置为主机器人轴, 在 PMOVE2 中设置为副机器人轴的所有轴均为对象。
设置为其他组或附加轴的轴使用此移动命令时将不移动。

- 移动种类: PTP
- 托盘定义编号: 数值表达式
- 托盘位置编号: 数值表达式
- 选项: 速度指定、圆弧插补移动指定、STOPON 条件指定

各托盘定义中的位置编号如下。

各托盘定义中的位置编号



35816-R7-00

MEMO

- XYZ 属性轴每次都会移动到计算所得位置, R 属性轴则会移动到通过各托盘坐标点数据的 P[1] 所指定的位置。

选配件	PTP	备注
速度指定 (SPEED)	○	仅指定 PMOVE 语句有效
圆弧插补移动	○	仅指定 PMOVE 语句有效
STOPON 条件指定	○	仅程序执行有效

SAMPLE

PMOVE (1, 16) 主机器人轴从当前位置向托盘定义编号 1 的托盘位置编号 16 中指定的位置移动

移动种类

● PTP (Point to Point) 移动

所有移动对象轴动作结束后(轴到达公差范围内以后),开始PTP移动,并在**所有移动对象轴进入OUT有效位置范围内后结束命令**。移动对象轴将同时到达。并无法保证此时的路径。

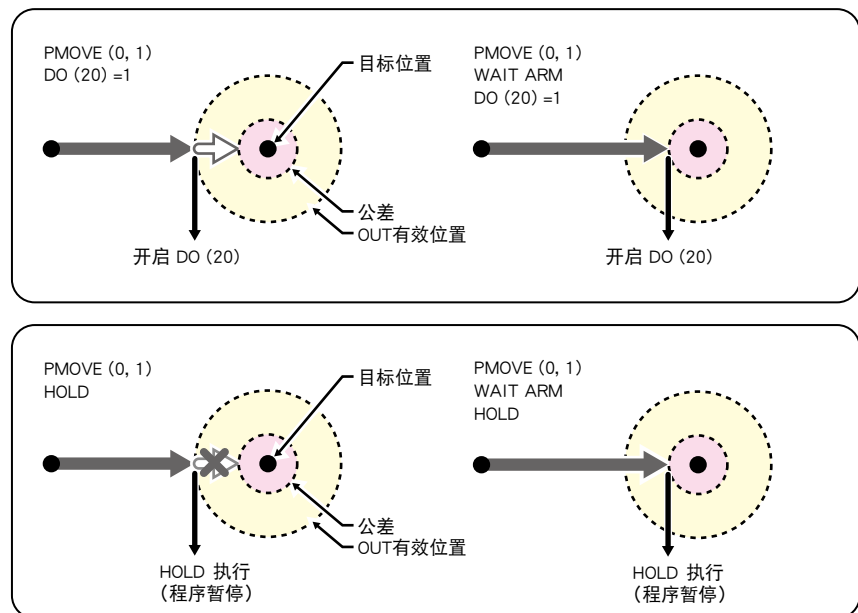
● PMOVE 命令之后的命令注意点

PMOVE 命令的下一个命令为信号输出命令等可执行命令时,移动对象轴进入OUT有效位置范围内后,将开始执行下一个命令。即,在轴到达目标位置的公差范围内之前,将开始下一个命令。
示例:

信号输出 (DO 等)	进入 OUT 有效位置范围内后, 将输出信号。
DELAY	进入 OUT 有效位置范围内后, 将执行 DELAY 命令, 并开始等待时间。
HALT	进入 OUT 有效范围内后, 程序停止并被复位。因此, 移动将停止。
HALTALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部停止, 任务 1 被复位, 其它任务结束。因此, 移动也会停止。
HOLD	进入 OUT 有效范围内后, 程序将暂停。因此, 移动将停止。
HOLDALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部暂停。因此, 移动也会停止。
WAIT	进入 OUT 有效位置范围内后, 将执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时, 请使用 "WAIT ARM" 语句。

PMOVE命令



35827-R7-00

选项种类

● 速度指定

PTP

格式

1. SPEED = <表达式>
2. S = <表达式>

值 <表达式>1 ~ 100(单位: %)

含义 使用<表达式>指定程序移动速度。移动速度由自动移动速度与程序移动速度的乘积决定。
仅对指定的 PMOVE 语句有效。

SAMPLE

PMOVE (1, 3), S=10 从当前位置按照 10% 的程序速度向托盘定义编号 1 的托盘位置编号 3 中指定位置移动

● 圆弧插补移动指定

PTP

格式

x = <表达式> [, x = <表达式> ...]

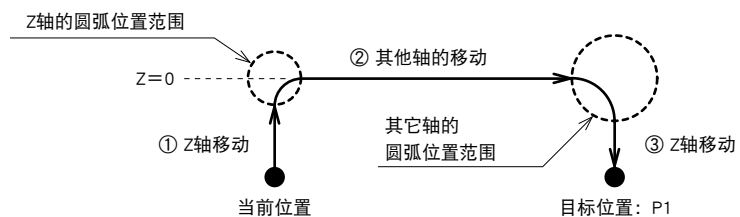
值 xZ, R, A, B 的轴指定
<表达式> 整数值的情况下, 为脉冲单位
实数值(带有小数点)的情况下为毫米 / 度单位

- 含义
1. 使 x 中指定的轴移动至<表达式>中指定的位置。
 2. 当指定轴进入圆弧位置范围内时, 指定轴以外的轴将移动到目标位置。
 3. 当指定轴以外的轴进入圆弧位置范围内, 且指定轴进入<表达式>中指定位置的公差范围内, 指定的轴将移动到目标位置。
 4. 所有轴均进入 OUT 有效位置范围内后, 结束命令。

SAMPLE

PMOVE (1, A), Z=0 从当前位置移动至第 1、Z 轴移动到 0 脉冲, 其他轴向托盘定义编号 1 的托盘位置编号 A 中指定的位置移动, 最后 Z 轴向托盘位置编号 A 中指定的位置移动

SAMPLE: PMOVE Z



35817-R7-00

- STOPON 条件指定

PTP

格式

STOPON < 条件表达式 >

含义

移动中条件表达式成立时，移动将停止。此时的停止动作为减速停止，所以在条件成立后将产生减速移动距离。

移动开始时如果条件表达式成立，将不移动，直接结束命令。

仅程序执行时有效。

SAMPLE

```
PMOVE (A, 16), STOPON DI (20) =1
```

..... 向托盘定义编号 A 的托盘位置编号 16 中指定的位置移动。如果中途 DI (20) =1 的条件成立，将减速停止移动

MEMO

- 在指定 STOPON 条件中使用的条件表达式表示数值的情况下，当表达式的值非 0 时为“真”，当成为 0 时则为“假”。

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

```
[LET] Pn = p1 p2 p3 p4 p5 p6[ f ][ f 1][ f 2]
```

值

n.....坐标点编号：0 ~ 29999
 p1 ~ p6.....坐标点数据：根据形式的不同，范围有所差异
 f1.....第 1 机械臂转数信息：-1、0、1（仅限于 YK500TW）
 f2.....第 2 机械臂转数信息：-1、0、1（仅限于 YK500TW）

含义

定义坐标点数据。

1. n 表示坐标点编号。
2. p1 ~ p6 的输入数据用空格(空白行)分隔。
3. p1 ~ p6 的输入数据全部为整数形式(无小数点)时, 将被识别为脉冲单位。此时, p1 ~ p6 对应第 1 轴~第 6 轴。
4. 在 x ~ b 的输入数据中, 只要有 1 个**实数形式(带小数点)的元素, 即被识别为毫米单位**。此时, x ~ z 对应正交坐标系的 x, y, z 坐标, r ~ b 对应第 4 轴~第 6 轴。
5. p1 ~ p6 的输入数据请在以下范围内输入。
 脉冲单位：-6,144,000 ~ 6,144,000 的范围
 毫米单位：-99,999.99 ~ 99,999.99 的范围



要点

- 同时存在两种指定单位时(整数与实数同时存在时), 所有值被视作毫米 / 度单位。

水平多关节型机器人且以毫米单位指定坐标点定义数据时, 可指定手系统标志。
 在水平多关节型机器人上设置扩展设置的手系统标志时, 对 f 设置 1 及 2。1 及 2 以外的数值或未指定数值时, 将视作无手系统标志设定 (0)。

- 1: 表示在右手系统中设置了坐标点。
- 2: 表示在左手系统中设置了坐标点。

当为 YK500TW 且以毫米为单位指定坐标点定义时, 可以指定第 1 机械臂转数信息以及第 2 机械臂转数信息(*1)。

在 YK500TW 机器人中, 当指定扩展指定的第 1 机械臂转数信息以及第 2 机械臂转数信息时, 在 f1 及 f2 将指定 -1、0、1 中的任何一个。如果指定了 -1、0、1 以外的数值或者未设定第 1 机械臂转数信息及第 2 机械臂转数信息时, 则作为 0 进行处理。

- 0: 表示将机械臂转数信息设定为 0。
 1: 表示将机械臂转数信息设定为 1。
 -1: 表示将机械臂转数信息设定为 -1。

*1: 关于第 1 机械臂转数信息、第 2 机械臂转数信息的详细说明, 请参照第 4 章“3 坐标点数据形式”的内容。



要点

- 输入全部变为常量。
- 如果在执行坐标点定义语句过程中控制器电源被切断，则可能会发生[9.2: Point data destroyed]等存储器相关错误。

SAMPLE

```

P1 =      0      0      0      0      0      0
P2 =  100.00  200.00  50.00  0.00  0.00  0.00
P3 =   10.00   0.00   0.00  0.00  0.00  0.00
P10=  P2
FOR A=10 TO 15
  P [A+1] =P [A] +P3
NEXT A
FOR A=10 TO 16
  MOVE P, P1, P [A]
NEXT A
HALT

```

相关命令 坐标点赋值语句 (LET)

格式

PPNT (托盘定义编号、托盘位置编号)

含义 创建托盘定义编号与托盘位置编号中指定的坐标点数据。

SAMPLE

P10=PPNT (1, 24) 在 P10 创建托盘定义编号 1 的托盘位置编号 24 中指定的坐标点数据

相关命令 PDEF, PMOVE

格式

```
PRINT [ <表达式> ] [ | , | <表达式> ... ] [ | , | ]
                | ; |                               | ; |
```

值 <表达式> 字符串、数值、变量

含义 此命令语句在手持编程器的画面上显示指定变量的值。输出定义如下所示。

1. 在<表达式>中指定了数值或字符串时，将按原样显示。指定了变量或数组时，将显示赋值给指定变量或数组的值。
2. 省略了<表达式>时，仅进行换行。
3. 数据项目的显示超出画面行宽时，将换行并在下一行中显示。
4. 显示分隔符采用逗号(,)时，显示项目之间将显示空格(空白行)。
5. 显示分隔符采用分号(;)时，显示项目将连续显示。
6. 显示分隔符结束时，将不换行，并在显示行执行下一个 PRINT 语句。显示分隔符未结束时，将换行。

MEMO

- PRINT 语句用于在手持编程器画面上显示而与画面进行数据通信。因此，当连续执行 PRINT 语句时，程序的执行时间可能会延长。

SAMPLE

```
PRINT A ..... 显示变量 A 的值
PRINT "A1 ="; A1 ..... 在 "A1 =" 之后显示变量 A1 的值
PRINT "B (0), B (1) = "; B (0); "," ; B (1)
PRINT P100 ..... 显示 P100 的值
```

相关命令 INPUT

格式

1. PSHFRC [< 机器人编号 >] < 表达式 >
2. PSHFRC [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > -1000 ~ 1000 (单位 : %)

含义

将 < 机器人编号 > 指定机器人轴的推压力参数变更为 < 推压力值 > 的值。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

在 PUSH 语句中省略了 F 选项指定时，通过推压力参数的设定值进行推压控制。

实际的推压力如下所示。

· 额定推力 × < 推压力值 > / 100

形式 1 对所有轴的参数进行变更。

形式 2 对按 < 轴编号 > 指定轴的参数进行变更。

SAMPLE

PSHFRC (1) = 10 变更为机器人 1 中第 1 轴推压力参数的 10%

函数

格式

PSHFRC [< 机器人编号 >] (< 轴编号 >)

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义

获取 < 机器人编号 > 指定机器人轴之中 < 轴编号 > 指定轴的推压力参数值。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

SAMPLE

A=PSHFRC (1) 将机器人 1 中第 1 轴的推压力参数赋给变量 A。

格式

1. PSHJGSP [< 机器人编号 >] < 表达式 >
2. PSHJGSP [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 0 : 无效, 1 ~ 100 (单位 : %)

含义

将 < 机器人编号 > 指定机器人轴的推压判定速度阈值参数变更为 < 表达式 > 的值。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。
 在推压判定速度阈值参数有效, 且 PUSH 语句中的推压力达到了指定值的状态下, 仅当移动速度低于 < 表达式 > 时, 才可判定为处于推压中。
 < 表达式 > 的设定值指定方式如下所示。
 0 : 将阈值设定作为无效, 当推压力达到指定值时, 判定为处于推压中。
 1 ~ 100 : 将 PUSH 语句中的移动速度作为 100%, 按比例指定阈值。

SAMPLE

PSHJGSP (1) = 50 将机器人 1 中第 1 轴的推压判定速度阈值参数变更为 50%。

函数

格式

PSHJGSP [< 机器人编号 >] (< 轴编号 >)

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义

获取 < 机器人编号 > 指定机器人轴之中 < 轴编号 > 指定轴的推压判定速度阈值参数值。
 < 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

SAMPLE

A=PSHJGSP (1) 将机器人 1 中第 1 轴的推压判定速度阈值参数赋给变量 A。

格式

1. PSHMTD [< 机器人编号 >] < 表达式 >
2. PSHMTD [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 0 : 累计方式, 1 : 复位方式

含义

将 < 机器人编号 > 指定机器人轴的推压方式参数变更为 < 值 > 的值。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

可以根据 < 值 > 并按下列方式对 PUSH 语句的推压方式进行指定。

0 : 对推压力达到指定值的时间进行累计, 进行推压控制结束判定。

1 : 仅在推压力连续达到指定值时, 才可进行推压控制结束判定。当推压力低于指定值时, 将推压经过时间复位为 0。

形式 1 对所有轴的参数进行变更。

形式 2 对按 < 轴编号 > 指定轴的参数进行变更。

SAMPLE

PSHMTD (1) = 1 将机器人 1 中第 1 轴的推压方式参数变更为复位方式。

函数

格式

PSHMTD [< 机器人编号 >] (< 轴编号 >)

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义

获取 < 机器人编号 > 指定机器人轴之中 < 轴编号 > 指定轴的推压方式参数值。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

SAMPLE

A=PSHMTD (1) 将机器人 1 中第 1 轴的推压方式参数赋给变量 A。

格式

PSHRSLT [< 机器人编号 >] (< 轴编号 >)

值 < 机器人编号 > 1 ~ 4
< 轴编号 > 1 ~ 6

含义 获取针对 < 机器人编号 > 指定机器人的 < 轴编号 > 指定轴所执行的 PUSH 语句结束状态。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

0 由于到达推压时间以外的原因结束

1 因到达推压时间而结束时

SAMPLE

```
PUSH(3,P1) ..... 在对机器人 1 中第 3 轴进行推压控制的同时，
                    向 P1 所指定的位置进行移动
IF PSHRSLT(3) = 1 THEN ..... 因到达推压时间而结束时
GOTO *OK
ELSE ..... 因到达推压时间以外的原因而结束时
GOTO *NG
ENDIF
```

格式

1. PSHSPD [< 机器人编号 >] < 表达式 >
2. PSHSPD [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 推压速度 > 0 : 无效, 1 ~ 100 (单位 : %)

含义

将 < 机器人编号 > 指定机器人轴的推压移动速度参数变更为 < 表达式 > 的值。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

PUSH 语句中的移动速度如下所示。

· PUSH 语句中没有 S、DS 选项指定时 :

机器人的最高速度 (mm/ 秒或者度 / 秒) × 推压移动速度 (%) × 自动移动速度 (%)

· PUSH 语句中有 S 选项指定时 :

机器人的最高速度 (mm/ 秒或者度 / 秒) × 推压移动速度 (%) × 自动移动速度 (%)
 × 程序移动速度 (%)

· PUSH 语句中有 DS 选项指定时 :

机器人的最高速度 (mm/ 秒或者度 / 秒) × 推压移动速度 (%) × 轴的移动速度 (%)

SAMPLE

PSHSPD (1) = 50 将机器人 1 中第 1 轴的推压移动速度参数变更为 50%。

函数

格式

PSHSPD [< 机器人编号 >] (< 轴编号 >)

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义

获取 < 机器人编号 > 指定机器人轴之中 < 轴编号 > 指定轴的推压移动速度参数值。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

SAMPLE

A=PSHSPD (1) 将机器人 1 中第 1 轴的推压移动速度参数赋给变量 A。

格式

1. PSHTIME [<机器人编号>] <表达式>
2. PSHTIME [<机器人编号>] (<轴编号>) = <表达式>

值

<机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6
 <表达式> 1 ~ 32767 (单位: ms)

含义

将<机器人编号>指定机器人轴的推压时间参数变更为<表达式>的值。<机器人编号>可以省略。当进行省略时, 机器人 1 被指定。

当 PUSH 语句中省略了 TIM 选项指定时, 通过推压时间参数的设定值来进行推压控制。

形式 1 对所有轴的参数进行变更。

形式 2 对<轴编号>指定轴的参数进行变更。

SAMPLE

PSHTIME (1) = 1000 将机器人 1 中第 1 轴的推压时间参数变更为 1000ms。

函数

格式

PSHTIME [<机器人编号>] (<轴编号>)

值

<机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6

含义

获取<机器人编号>指定机器人轴之中<轴编号>指定轴的推压时间参数值。<机器人编号>可以省略。当进行省略时, 机器人 1 被指定。

SAMPLE

A=PSHTIME (1) 将机器人 1 中第 1 轴的推压时间参数赋给变量 A。

格式

PUSH[< 机器人编号 >] (< 轴编号 > , < 表达式 >) [, 选项]

值

< 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 电机位置 (毫米、度、脉冲) 或者坐标点表达式

含义

以轴为单位对前进方向的推压力进行控制, 同时针对 < 机器人编号 > 指定机器人轴执行绝对位置移动。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

- 移动的种类: 轴指定推压 PTP
- 坐标点指定: 直接数值指定、坐标点定义指定
- 选项: 推压力指定、推压时间值、推压速度指定、STOPON 指定

移动种类

● 轴指定 PTP (Point to Point)

当 < 轴编号 > 指定轴的动作结束后 (到达公差范围以内后), 对轴的前进方向的推压力进行控制, 同时开始移动。

推压控制开始条件如下所示。

- 通过 PUSH 语句开始轴移动之后
- 连接动作结束后 (在有 CONT 指定的移动命令的下一行指定了 PUSH 语句时)

命令结束的条件如下所示。

- 轴到达目标位置的交差范围内。
 - 轴的推压力达到 < 推压力值 >, 并保持该状态经过 < 推压时间值 > 以上的时间。
- PUSH 语句结束状态可通过 PSHRSLT 语句进行确认。

推压力解除条件如下所示。

- 包含 STOP 停止的 PUSH 命令结束后, 当执行移动命令时;
- 伺服关闭时;
- 断开控制器的电源然后重新接通时

当 PUSH 语句的下一个命令是信号输出命令等可执行的命令时、移动对象轴的推压条件成立时、或者轴到达了目标位置的公差范围内时，将执行下一个命令。

示例：

信号输出 (DO 等)	当推压条件成立或者进入公差范围内时，将输出信号。
DELAY	当推压条件成立或者进入公差范围内时，将执行 DELAY 命令，并开始等待时间。
HALT	进入 OUT 有效范围内后，程序停止并被复位。因此，移动将停止。
HALTALL	当推压条件成立或者进入公差范围内时，正在执行的程序将会全部停止，对任务 1 进行复位，其它任务结束。因此，移动也会停止。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
HOLDALL	当推压条件成立或者进入公差范围内时，正在执行的程序将会全部暂停。因此，移动也会停止。
WAIT	当推压条件成立或者进入公差范围内时，将执行 WAIT 命令。

SAMPLE

PUSH(1,P0)..... 第 1 轴从当前位置向 P0 指定的位置移动

坐标点指定的种类

● 直接数值指定

在 <表达式> 中直接指定马达位置

当马达位置的数值为整数时，判断为脉冲单位。而当数值为实数值时，判断为毫米 / 度单位，每个轴从 0 脉冲位置向脉冲换算位置移动。

SAMPLE

PUSH(1,10000)..... 机器人 1 中第 1 轴将从当前位置向 100000 脉冲处移动。

PUSH[2](2,90.00) 机器人 2 中第 2 轴将从当前位置向 90 度位置进行移动 (旋转轴时)。

● 坐标点定义指定

在 <表达式> 中指定坐标点表达式。使用 <轴编号> 中指定轴的数据。

当坐标点表达式为毫米 / 度单位时，每个轴从 0 脉冲位置向脉冲换算位置移动。

SAMPLE

PUSH(1,P1)..... 机器人 1 中第 1 轴将从当前位置向 P1 所指定的位置进行移动。

PUSH[2](2,P90)..... 机器人 2 中第 2 轴将从当前位置向 P90 所指定的度数位置进行移动 (旋转轴时)。

选项种类

● 推压力指定

格式

F = < 表达式 >

值 < 表达式 >-1000 ~ 1000 (单位 : %)

含义 通过 < 表达式 > 指定相对于轴前进方向的推压力。
实际的推压力如下所示。
· 额定推力 × < 表达式 > / 100
当省略 < 推压力值 > 时，将使用通过参数设定的 < 推压力值 >。

SAMPLE

PUSH(1,10000),F=10..... 机器人 1 中第 1 轴以相当于额定推力 10% 的推压力，从当前位置向 100000 脉冲处移动。

● 推压时间指定

格式

TIM = < 表达式 >

值 < 表达式 >1 ~ 32767 (单位 : ms)

含义 通过 < 表达式 > 来指定以指定推压力持续进行推压的时间。
推压力达到设定值的状态持续 < 表达式 > 以上的时间时，PUSH 语句会结束。
当省略了本选项时，将使用参数的设定值。

SAMPLE

PUSH(1,10000),S=10..... 第 1 轴以相当于推压移动速度与自动移动速度之积 10% 的速度，从当前位置向 100000 脉冲处移动。

- 速度指定

格式

1. SPEED = < 表达式 >
2. S = < 表达式 >

值 < 表达式 >1 ~ 100 (单位 : %)

含义 使用 < 表达式 > 指定程序移动速度。

实际速度如下所示。

· 机器人的最高速度 (mm/ 秒或度 / 秒) × 推压移动速度 (%) × 自动移动速度 (%) × 程序移动速度 (%)

仅对指令的 PUSH 语句有效。

SAMPLE

PUSH(1,10000),S=10..... 第 1 轴以相当于推压移动速度与自动移动速度之积 10% 的速度, 从当前位置向 100000 脉冲处移动。

格式

1. DSPEED = < 表达式 >
2. DS = < 表达式 >

值 < 表达式 >0.01 ~ 100.00 (单位 : %)

含义 使用 < 表达式 > 指定轴的移动速度。

实际速度如下所示。

· 机器人的最高速度 (mm/ 秒或度 / 秒) × 推压移动速度 (%) × 轴的移动速度 (%)

仅对指令的 PUSH 语句有效。

· 不受自动移动速度值 (%) 的影响, 一贯按照 DSPEED 的 < 表达式 > 值 (%) 移动。

SAMPLE

PUSH(1,10000),DS=0.1 第 1 轴以相当于推压移动速度 0.1% 的速度, 从当前位置向 100000 脉冲处移动。

- STOPON 条件指定

格式

STOPON < 条件表达式 >

含义

移动中条件表达式不成立时，移动将停止。此时的停止动作为**减速停止**，在条件成立后将产生**减速移动距离**。

移动开始时如果条件表达式成立，将不移动，直接结束命令。

仅程序执行时有效。

SAMPLE

DRIVE (1, 10000), STOPON DI (20) =1

..... 第 1 轴从当前位置向 10000 脉冲位置移动，如果中途 DI (20) =1 的条件成立，则停止移动并执行下一个步骤。

 **MEMO**

- 在指定 STOPON 条件中使用的条件表达式表示数值的情况下，当表达式的值非 0 时为“真”，当成为 0 时则为“假”。

相关命令

PSHFRC, PSHTIME, PSHMTD, PSHDSP, PSHSPD, PSHRSLT, CURTRQ, CURTQST

格式

RADDEG (<表达式>)

值 <表达式> 角度 (单位: 弧度)**含义** 将<表达式>的值转换为度。

SAMPLE

LOC R (P0) =RADDEG (ATN (B)) ... 将变量 B 的反正切值转换为度, 并赋值给 P0 的 R 数据

相关命令 ATN, COS, DEGRAD, SIN, TAN

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

1. REM <字符串>
2. ' <字符串>

含义 REM 或 " " 以后的字符被视作注释。不执行注释语句。" ' " 也可写入行的中间。

SAMPLE

```
REM *** MAIN PROGRAM ***  
    ' *** SUBROUTINE ***  
HALT ' HALT COMMAND
```

格式 1

RESET	DOm ([b, . . . ,b]) DO (mb, . . . ,mb) MOm ([b, . . . ,b]) MO (mb, . . . ,mb) TOm ([b, . . . ,b]) TO (n-b, . . . ,nb) LOm ([b, . . . ,b]) LO (nb, . . . ,nb) SOm ([b, . . . ,b]) SO (mb, . . . ,mb)
-------	--

格式 2

RESET TCOUNTER

值

m: 端口编号..... 2 ~ 7, 10 ~ 17, 20 ~ 27
n: 端口编号..... 0、1
b: 指定比特..... 0 ~ 7

含义

格式 1 关闭指定端口的比特。
格式 2 清除 10ms 计数器变量 (10ms 计数器变量用于以 10ms 为单位进行时间测量)。

指定多个比特时, 请从左边开始降序 (大→小) 记述。
省略了 [b, . . . ,b] 时, 8 比特均将成为对象。

SAMPLE

RESET DO2 () DO (27 ~ 20) 关闭
RESET DO2 (6, 5, 1) DO (26, 25, 21) 关闭
RESET (37, 35, 27, 20) DO (37, 35, 27, 20) 关闭
RESET TCOUNTER 清除 10ms 计数器变量

相关命令 SET, DO, MO, SO, TO, LO



注意

· 无法对 DO, SO 指定端口 0 和 1。



参考

· 关于指定比特, 请参阅第 3 章“10 指定比特”。

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

RESTART	Tn “< “< 程序名称 >” >” PGm
---------	-------------------------------

值 m : 程序编号..... 0 ~ 99
n : 任务编号..... 1 ~ 16

含义 重新启动暂停中 (SUSPEND 状态) 的其他任务。任务也可通过正在执行的程序名称或者编号进行指定。

MEMO

· 指定并执行了非暂停中的任务 (程序) 时将发生错误。

SAMPLE

```

START <SUB_PGM>,T2
  FLAG=1
*L0:
  IF FLAG=1 AND DI2 (0) =1 THEN
    SUSPEND T2
    FLAG=2
  WAIT DI2 (0) =0
  ENDIF
  IF FLAG=2 AND DI2 (0) =1 THEN
    RESTART T2
    FLAG=1
    WAIT DI2 (1) =0
  ENDIF
  MOVE P, P0
  MOVE P, P1
  GOTO *L0
  HALTALL
程序名称 :SUB_PGM
*SUBPGM:
' SUBTASK ROUTINE
*SUBTASK:
  DO2 (0) =1
  DELAY 1000
  DO2 (0) =0
  DELAY 1000
  GOTO *SUBPGM
  EXIT TASK

```

相关命令 CUT, EXIT TASK, START, SUSPEND

参照 详细内容, 请参阅多任务。

格式

1. RESUME NEXT
2. RESUME < 标签 >



参考

· 详细说明，请参阅“62 ON ERROR GOTO”。

含义

进行错误的恢复处理后，恢复执行程序。
按照程序恢复启动的位置，有以下 3 种方法。

- | | |
|------------------|----------------------------|
| 1. RESUME | 从错误原因命令开始恢复启动程序。 |
| 2. RESUME NEXT | 从错误原因命令的下一个命令开始恢复启动程序。 |
| 3. RESUME < 标签 > | 从显示有 < 标签 > 行上的命令开始恢复启动程序。 |

MEMO

- 无法在错误处理例程以外执行 RESUME 语句。
- 发生了 [17.4: Over load] 等重大错误时，无法执行错误恢复处理。

相关命令 ON ERROR GOTO

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

使 GOSUB 中的分支处理返回至 GOSUB 的下一行

格式

```
GOSUB <标签>                                ※GOSUB 也可写成 GO SUB
:
<标签> :
:
RETURN
```

含义 结束子例程的处理，并返回到跳转源 GOSUB 语句的下一个命令行。
使用 GOSUB 语句跳转到子例程时，请务必使用 RETURN 语句结束子例程。使用 GOTO 等语句从例程跳转时，可能会发生 [5.12: Stack overflow] 等错误。

SAMPLE

```
*ST:
  MOVE P, P0
  GOSUB *CLOSEHAND
  MOVE P, P1
  GOSUB *OPENHAND
GOTO *ST
HALT
' SUB ROUTINE
*CLOSEHAND:
  DO (20) = 1
RETURN
*OPENHAND:
  DO (20) = 0
RETURN
```

相关命令 GOSUB

格式

RIGHT\$ (<字符串表达式>, <表达式>)

值 <表达式>0 ~ 75

含义 从<字符串表达式>中指定的字符串右侧(最后面)提取<表达式>中指定位数的字符串。

<表达式>的值不在0 ~ 75范围内时,将出错。

<表达式>的值为0时,RIGHT\$变为空字符串。

<表达式>的字符数大于<字符串表达式>的字符数时,RIGHT\$变为与<字符串表达式>相同的内容。

SAMPLE

B\$=RIGHT\$(A\$, 4) 将A\$右侧开始的4个字符赋值给B\$

相关命令 LEFT\$, MID\$

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

RIGHTY [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 选择通过右手系统使 < 机器人编号 > 指定机器人向正交坐标系指定坐标点移动。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

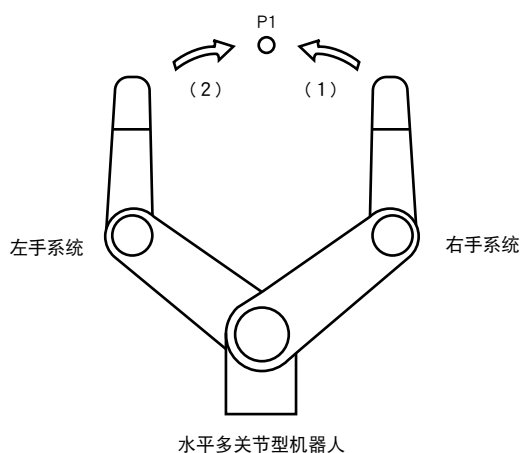
即使执行了命令仅选择也无法使机器人移动。轴在动作时，请等待动作结束（定位至公差内）再执行。

此命令只对水平多关节机器人有效。

SAMPLE

```
RIGHTY ..... 将机器人 1 设置为右手系统（下图中的 1）
MOVE P, P1
RIGHTY ..... 将机器人 1 设置为左手系统（下图中的 2）
MOVE P, P1
RIGHTY
HALT
```

SAMPLE: LEFTY / RIGHTY



35818-R7-00

相关命令 LEFTY

格式

RSHIFT (<表达式 1 > , <表达式 2 >)

含义 将<表达式 1 >的比特值仅向右位移<表达式 2 >的量。位移后,空着的位置插入 0。

SAMPLE

A=RSHIFT (&B10111011, 2)..... 将 &B10111011 向右位移 2 比特的值 (&B00101110) 赋值给 A

相关命令 LSHIFT

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

```

SELECT [CASE] <表达式>
  CASE <表达式的列表 1 >
    [命令区 1]
  [CASE <表达式的列表 2 >
    [命令区 2]
    :
  [CASE ELSE
    [命令区 n]]
END SELECT

```

含义 执行由<表达式>的值指定的命令区。设置方法如下所示。

1. CASE 语句的后面的记述<表达式列表>采用逗号(,)分隔并列的多个数值表达式或字符串表达式。
2. <表达式>的值与<表达式列表>中的任意一个表达式值一致时,将执行指定的命令区。执行命令区后,将跳转至END SELECT 语句的下一个命令。
3. <表达式>的值与所有CASE 语句的<表达式列表>的值不一致时,将执行CASE ELSE 语句后面记述的<命令区>。执行命令区后,将跳转至END SELECT 语句的下一个命令。
4. <表达式>的值与所有CASE 语句的<表达式列表>的值不一致且无CASE ELSE 语句的记述时,将跳转至END SELECT 语句的下一个命令。

SAMPLE

```

WHILE -1
SELECT CASE DI3 ( )
  CASE 1, 2, 3
    CALL *EXEC (1, 10)
  CASE 4, 5, 6, 7, 8, 9, 10
    CALL *EXEC (11, 20)
  CASE ELSE
    CALL *EXEC (21, 30)
END SELECT
WEND
HALT

```

格式

SEND < 读出文件 > TO < 写入文件 >



要点

- 写入读出专用文件时的错误示例

SEND CMU TO DIR

SEND PNT TO SI()

- 数据格式不匹配的错误示例

SEND PGM TO PNT

SEND SI() TO SFT

含义

将<读出文件>的数据转发至<写入文件>。

在 DO 端口、MO 端口、TO 端口、LO 端口、SO 端口、SOW 端口中，不可作为写入文件指定各端口全部 (DO ()、MO () 等)。

对于部分个别文件 (DOn ()、MOn () 等)，有时也可能无法作为写入文件指定。详细内容，请参阅后述第 8 章“数据文件详情”。

无法对读出专用文件(下表“写入”栏中带有“×”的项目)写入。

即使正确指定了读出 / 写入文件，只要双方的数据格式不匹配，也无法执行。

分类	文件名	记述格式		读出	写入
		全部	个别文件		
用户	所有文件	ALL	_____	○	○
	程序	PGM	<bbbbbbbb>	○	○
	坐标点	PNT	Pn	○	○
	坐标点注释	PCM	PCn	○	○
	参数	PRM	/cccccc/	○	○
	位移定义	SFT	Sn	○	○
	机械手定义	HND	Hn	○	○
	托盘定义	PLT	PLn	○	○
变量、常量	变量	VAR	ab..by	○	○
	数组变量	ARY	ab..by(x)	○	○
	常量		"cc...c"	○	×
状态	程序目录	DIR	<<bbbbbbbb>>	○	×
	参数目录	DPM	_____	○	×
	机器参照	MRF	_____	○	×
	错误履历	LOG	_____	○	×
	存储器剩余空间	MEM	_____	○	×
设备	DI 端口	DI()	DIn()	○	×
	DO 端口	DO()	DOn()	○	○
	MO 端口	MO()	MOn()	○	○
	TO 端口	TO()	TOn()	○	○
	LO 端口	LO()	LOn()	○	○
	SI 端口	SI()	SIn()	○	×
	SO 端口	SO()	SOn()	○	○
	SIW 端口	SIW()	SIWn()	○	×
	SOW 端口	SOW()	SOWn()	○	○
	RS-232C	CMU	_____	○	○
	Ethernet	ETH	_____	○	○
其他	文件结束代码	EOF	_____	○	×

n: 数字 a: 英文字母 b: 英文字母数字或下划线 (_)

○: 可以

c: 英文字母数字或特殊符号

x: 表达式 (数组自变量)

y: 变量类型

×: 不可以

MEMO

· 中途停止 SEND 语句并再次启动时，请注意以下事项。

1. 从 RS-232C / Ethernet 读入时 (SEND CMU TO XXX / SEND ETH TO XXX)
在读出接收缓冲区半途状态下中途停止时，已获得的数据将被损坏。
2. 写入 RS-232C / Ethernet 时 (SEND XXX TO CMU / SEND XXX TO ETH)
在写入发送缓冲区半途状态下中途停止时，将再次从起始开始写入数据。

SAMPLE

SEND PGM TO CMU 从 RS-232C 端口输出所有用户程序
 SEND < PRG1 > TO CMU 从 RS-232C 端口输出程序名 PRG1
 SEND CMU TO PNT 从 RS-232C 端口输入坐标点数据文件
 SEND "T1" TO CMU 从 RS-232C 端口输出 "T1" 字符串
 SEND CMU TO A\$ 从 RS-232C 端口将字符串数据输入至变量 A\$

参照 详细内容，请参阅第 8 章“数据文件详情”。

格式

SERVO [<机器人编号>]	ON OFF FREE PWR	[(<轴编号>)]
-----------------	--------------------------	-----------

值 <机器人编号> 1 ~ 4
<轴编号> 1 ~ 6

含义 控制指定编号的轴或所有轴的伺服 ON / OFF。

- 当指定了<轴编号>时，仅以指定轴作为对象；当未指定时，则以所有全轴作为对象。
- ON 伺服使能。未指定轴时，马达电源也将开启。
 - OFF 伺服禁止并启用动力制动。带有制动器的轴将进行制动并锁定。未指定轴时，马达电源也将关闭。
 - FREE 伺服禁止并解除动力制动。带有制动器的轴也将解除制动。未指定轴时，马达电源也将关闭。

· 此命令语句将在指定机器人的所有轴动作结束后（在公差范围内定位后）执行。

SAMPLE

SERVO ON 开启所有轴的伺服。
SERVO OFF 针对关闭所有轴伺服且附带制动器的轴，其制动器将被锁定。
SERVO FREE (3) 第3轴（Z轴）伺服禁止，并解除制动



注意

- 请勿在使用 SERVO OFF 语句并关闭马达电源状态下进行机器人动作范围内的作业。请务必在紧急停止电路开启状态下进行机器人动作范围内的作业。



MEMO

格式

SET	DOm ([b, . . . , b]) DO (mb, . . . , mb) MOm ([b, . . . , b]) MO (mb, . . . , mb) TOm ([b, . . . , b]) TO (nb, . . . , nb) LOm ([b, . . . , b]) LO (nb, . . . , nb) SOm ([b, . . . , b]) SO (mb, . . . , mb)	[, <时间>]
-----	---	-----------

值	m : 端口编号..... 2 ~ 7, 10 ~ 17, 20 ~ 27
	n : 端口编号..... 0、1
	b : 指定比特..... 0 ~ 7
	<时间> 10 ~ 3600000 (单位 : ms)

含义 开启指定端口的比特。
在<时间>中指定脉冲输出时间(单位:ms)。经过指定时间后, 关闭输出并结束执行。

指定多个比特时, 请从左边开始降序(大→小)记述。

省略了[b, . . . , b]时, 8 比特均将成为对象。

指定了硬件上不存在的端口时, 将无任何输出。

SAMPLE

SET DO2 () DO (27 ~ 20) 开启
SET DO2 (6, 5, 1), 200 DO (26, 25, 21) 开启 200ms
SET DO (37, 35, 27, 20) DO (37, 35, 27, 20) 开启

相关命令 RESET, DO, MO, SO, TO, LO



注意

· DO, SO 均无法输出到端口 0 和 1。



参考

· 关于指定比特, 请参阅第 3 章“10 指定比特”。



要点

- 程序级别代码指的是子过程以外编写的程序。



MEMO

格式

```
SHARED <变量> [( )] [, <变量> [( )] ... ]
```

含义

不将程序级别代码中声明的变量作为形参提交，使在子过程中可参照。

在<变量>中指定子过程使用的程序级别变量。

指定简单变量名或后面带有括号()的数组变量名。指定了数组时，此数组全体均为对象。

- 若要对子过程提交变量，通常指定<形参>或使用 SHARED 变为可参照。
- SHARED 语句中可共享的变量仅限同一程序级别内的程序级别代码或子过程之间。

SAMPLE

```
DIM Y! (10)
X!=2. 5
Y! (10) =1. 2
CALL *DISTANCE
CALL *AREA
HALT
SUB *DISTANCE
    SHARED X!, Y! ( ) ..... 通过 SHARED 声明变量的参照
    PRINT X!^2+Y! (10)^2 ..... 共享变量
END SUB
SUB *AREA
    DIM Y! (10)
    PRINT X!*Y! (10) ..... 不共享变量
END SUB
```

相关命令 SUB, END SUB

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

SHIFT [< 机器人编号 >]	< 位移变量 > OFF
---------------------	-----------------

值 < 机器人编号 > 1 ~ 4

含义 根据 < 位移变量 > 指定的位移数据对 < 机器人编号 > 指定机器人设定位移坐标。

 **MEMO**

· 此命令语句在轴的动作结束后(定位在公差范围内后)执行。

SAMPLE

```
SHIFT S1
MOVE P, P10
SHIFT S [A]
MOVE P, P20
HALT
```

相关命令 位移定义语句、位移赋值语句

格式

SIN (<表达式>)

值 <表达式> 角度(单位: 弧度)

含义 赋予<表达式>中, 值相对的正弦值。

SAMPLE

A (0) =SIN (B*2+C) 将表达式 B*2+C 的正弦值赋值给数组 A (0)
A (1) =SIN (DEGRAD (30)) 将 30.0° 的正弦值赋值给 A (1)

相关命令 ATN, COS, DEGRAD, RADDEG, TAN

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

$$S_n = x \ y \ z \ r$$

值

n..... 0 ~ 39

x, y, z, r..... -99,999.99 ~ 99,999.99

含义

定义用于使机器人动作的坐标位移时的位移坐标。坐标单位仅限毫米(不可使用脉冲)。



要点

- 输入全部变为常量。
- 如果在执行位移坐标定义语句过程中控制器电源被切断, 则可能会发生 [9.6: Shift data destroyed] 等存储器相关错误。

1. n 表示位移编号。
2. x ~ r 的输入数据用空格(空白)隔开。
3. x ~ r 的输入数据被识别为毫米单位。
4. x ~ z 对应正交坐标系的 x, y, z 坐标位移量, r 对应 xy 坐标的旋转位移量。

SAMPLE

```

S0 =    0.00    0.00    0.00    0.00
S1 =   100.00   200.00   50.00   90.00
P3 =   100.00    0.00    0.00    0.00    0.00    0.00
SHIFT S0
MOVE P, P3
SHIFT S1
MOVE P, P3
HALT

```

相关命令 位移赋值语句、SHIFT

格式

- 1. [LET] SOm ([b, . . . , b]) = <表达式>
- 2. [LET] SO (mb, . . . , mb) = <表达式>

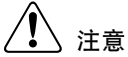
值 m : 端口编号.....2 ~ 7, 10 ~ 17, 20 ~ 27
 b : 指定比特.....0 ~ 7

含义 将指定值输出到 SO 端口。
 对于左边指定的比特，输出的值将变为 <值> 整数化后的后位比特。
 指定多个比特时，请从左边开始降序(大→小)记述。
 省略了 [b, . . . , b] 时，8 比特均将成为对象。
 指定了硬件上不存在的端口时，将无任何输出。

SAMPLE

```
SO2 ( ) =&B10111000 ..... SO (27, 25, 24, 23) 开启,
                               SO (26, 22, 21, 20) 关闭
SO2 (6, 5, 1) =&B010 ..... SO (25) 开启, SO (26, 21) 关闭
SO3 ( ) =15 ..... SO (33, 32, 31, 30) 开启,
                               SO (37, 36, 35, 34) 关闭
SO (37, 35, 27, 20) =A..... 将变量 A 整数化后的后 4 比特内容输出到 SO
                               (37, 35, 27, 20)
```

相关命令 RESET, SET



注意

· 无法输出至 SO0() 与 SO1()。



· 关于指定比特，请参阅第 3 章“10 指定比特”。

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

SPEED [< 机器人编号 >] < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 表达式 > 1 ~ 100 (单位 : %)

含义 将 < 机器人编号 > 指定机器人的程序移动速度变更为 < 表达式 > 指定的值。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

对指定机器人的所有轴 (包括机器人构成轴和附加轴) 的全部速度进行变更。

动作速度由手持编程器的操作或 ASPEED 命令设置的自动移动速度与

SPEED 命令指定的程序移动速度的乘积决定。

动作速度 = 自动移动速度 × 程序移动速度

示例 :

自动移动速度 80%

程序移动速度 50%

移动速度 = 40% (80% × 50%)



要点

- 自动移动速度
通过手持编程器的操作或 ASPEED 命令设置
- 程序移动速度
由 SPEED 命令或 MOVE、DRIVE 的 SPEED 选项设置

函数

格式

SPEED [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 获取利用 < 机器人编号 > 所指定机器人的程序移动速度值。
 < 机器人编号 > 可以省略。当省略时，机器人 1 被指定。

SAMPLE

ASPEED 100

SPEED 70

MOVE P, P0 按照 70% (=100*70) 的速度从当前位置向 P0 移动

SPEED 50

MOVE P, P1 按照 50% (=100*50) 的速度从当前位置向 P1 移动

MOVE P, P2, S=10 按照 10% (=100*10) 的速度从当前位置向 P2 移动

HALT

相关命令 ASPEED

格式

SQR (<表达式>)

值 <表达式> 0 或正数**含义** 赋予<表达式>中指定值的平方根。<表达式>的值为负数时，将出错。

SAMPLE

A=SQR (X^2+Y^2)..... 将 X^2+Y^2 的平方根赋值给变量 A

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

START	“< “< 程序名称 >” >”	Tn [, P]
	PGm	

值 m : 程序编号.....0 ~ 99

n : 任务编号.....1 ~ 16

p : 任务优先级.....1 ~ 64

含义 根据任务 n、优先顺序 p 启动指定程序。

当省略任务编号 n 时，未启动任务中编号最小的任务将会被自动指定。当省略任务优先顺序 p 时，将默认为 32。

省略了任务优先级 p 时，为 32。

数字越小，优先级越高；数字越大，优先级越低（高 1 ~ 低：64）。

当任务优先级较高的任务处于 RUNNING 状态时，优先级较低的任务也将保持着 READY 状态。

SAMPLE

```
START <SUB_PGM>,T2,33
*ST:
  MOVE P, P0, P1
  GOTO *ST
  HALT
  程序名称 :SUB_PGM
  *SUBPGM:
  'SUBTASK ROUTINE
  *SUBTASK:
    P100 = WHERE
    IF LOCZ (P100) > 10000 THEN
      DO (20) = 1
    ELSE
      DO (20) = 0
    ENDIF
  GOTO *SUBPGM
  EXIT TASK
```

相关命令 CUT, EXIT TASK, RESTART, SUSPEND, CHGPRI

格式

STR\$ (<表达式>)

含义 将<表达式>中指定的值转换为字符串。<表达式>中可指定整数型及实数型的数值。

SAMPLE

BS=STR\$ (10.01)

相关命令 VAL

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

```
SUB <标签> [( <形参> [, <形参> ... ] ) ]
  <命令区>
END SUB
```

含义 定义子过程。

通过 CALL 语句可执行子过程。执行了 END SUB 语句后，将跳转至调出的 CALL 语句的下一个命令。定义如下所示。

1. 子过程中声明的变量均为局部变量，仅在子过程内有效。每次调出子过程时，局部变量均将被初始化。
2. 使用全局变量（程序级别）时，使用 SHARED 语句。
3. 进行变量的提交时，使用 <形参>。<形参>通过逗号 (,) 分隔。
4. <形参>中变量名与所有数组（数组名的后面带有 ()）有效。如果指定了数组的元素（数组名后面带有 (<角标>)），将出错。

MEMO

- 在子过程内无法定义子过程。
- 虽然在子过程内可定义标签，但是通过 GOTO 或 GOSUB 等语句无法跳转至子过程以外的标签。
- 局部变量无法在 PRINT 语句、SEND 语句中使用。

SAMPLE 1

```
A=1
CALL *TEST
PRINT A
HALT
' SUB ROUTINE: TEST
SUB *TEST
  A=50 ..... 被视作与上文中使用的 A 为不同的变量
END SUB
```

MEMO

- 上述示例中，程序级别的变量 A 与子过程内的变量 A 无关。因此，第 3 行的 PRINT 语句中显示的值变为“1”。

SAMPLE 2

```
X% = 4
Y% = 5
CALL *COMPARE ( REF X%,REF Y% )
PRINT X%, Y%
Z% = 7
W% = 2
CALL *COMPARE ( REF Z%,REF W% )
PRINT Z%,W%
HALT
' SUB ROUTINE: COMPARE
SUB *COMPARE ( A%, B% )
  IF A% < B% THEN
    TEMP% = A%
    A% = B%
    B% = TEMP%
  ENDIF
END SUB
```

 MEMO

· 在上述示例中，各个不同变量作为自变量提交，并将子过程调出 2 次。

相关命令 CALL, EXIT SUB, SHARED

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

SUSPEND	Tn “< “<程序名称>” >” PGm
---------	-----------------------------

值 m : 程序编号..... 0 ~ 99
n : 任务编号..... 1 ~ 16

含义 暂停正在执行的其它任务。任务也可通过正在执行的程序名称或者编号进行指定。也可用于比当前任务优先级更高的任务。

MEMO

· 当指定并执行了未启动的任务(程序)时将会发生错误。

SAMPLE

```
START <SUB_PGM>,T2
SUSFLG=0
*L0:
  MOVE P, P0
  MOVE P, P1
  WAIT SUSFLG=1
  SUSPEND T2
  SUSFLG=0
GOTO *L0
HALT
程序名称 :SUB_PGM
*SUBPGM
' SUBTASK ROUTINE
*SUBTASK:
  WAIT SUSFLG=0
  DO2 (0) =1
  DELAY 1000
  DO2 (0) =0
  DELAY 1000
  SUSFLG=1
  GOTO *SUBPGM
EXIT TASK
```

相关命令

CUT, EXIT TASK, RESTART, SUSPEND

格式

```
SWI" < " <程序名> " > "
```

含义

从当前正在执行的程序切换至指定程序，并在执行编译后，从第 1 行开始执行。虽然切换程序时，输出变量的状态不会变化，但动态变量及数组变量将被清除。在程序名的前后加上 " < " 和 " > " (括号)。

MEMO

· 当指定程序不存在时，将会发生“3.3：无程序(代码：&H0303)”错误，并且停止运行。

SAMPLE

```
SWI < ABC > ..... 将执行程序切换至 ABC
```

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

TAN (<表达式>)

值 <表达式> 角度(单位: 弧度)

含义 赋予<表达式>中, 值相对的正切值。<表达式>的值为负数时, 将出错。

SAMPLE

A (0) =B-TAN (C) 将变量 B 与变量 C 的正切值之差赋值给数组 A (0)

A (1) =TAN (DEGRAD (20)) 将 20.0° 的正切值赋值给数组 A (1)

相关命令 ATN, COS, DEGRAD, RADDEG, SIN

格式

TCOUNTER

含义 从复位 TCOUNTER 变量时开始，每隔 1ms 输出计数值。
值计数至 65,535 后，返回 0。

SAMPLE

```
MOVE P, P0
WAIT ARM
RESET TCOUNTER
MOVE P, P1
WAIT ARM
A = TCOUNTER
PRINT TCOUNTER ..... 在手持编程器上显示进入 P0 至 P1 公差为止的
                        移动时间
```

相关命令 RESET

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

TIMES\$

含义 通过 hh: mm: ss 形式的字符串赋予当前时间。
hh 表示小时，mm 表示分钟，ss 表示秒。
时间的设置在系统模式的初始处理中进行。

SAMPLE

```
A$=TIMES$  
PRINT TIMES$
```

相关命令 DATE\$, TIMER



注意

- 内部使用的时钟与实际时间之间可能会有误差。

格式

TIMER

功能

通过上午 0 点开始的秒获取当前时间。可用于程序运行时间的测量等用途。
时间的设置在系统模式的初始处理中进行。

SAMPLE

```
A%=TIMER
FOR B=1 TO 10
MOVE P, P0
MOVE P, P1
NEXT
A%=TIMER-A%
PRINT A%/60; " : " ; A% MOD 60
HALT
```

相关命令 TIMES

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

1. [LET] TOm ([b, . . . ,b]) = <表达式>
2. [LET] TO (mb, . . . ,mb) = <表达式>

值

m : 端口编号.....0、1

b : 指定比特.....0~7

含义

将指定值输出至 TO 端口。对于左边指定的比特，输出的值将变为<表达式>的值整数化后的后位比特。

指定多个比特时，请从左边开始降序(大→小)记述。

省略了[b, . . . ,b]时，8 比特均将成为对象。

对于 SEQUENCE 程序中使用的比特，如果 SEQUENCE 程序正在 RUN 中，则该设置(开启、关闭状态)优先。

SAMPLE

```
TO0 ( ) = &B00000110
```

相关命令

RESET, SET

格式

1. TOLE [<机器人编号>] <表达式>
2. TOLE [<机器人编号>] (<轴编号>) = <表达式>

值 <机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6
 <表达式> 因设置的马达而异(单位:脉冲)

含义 将<机器人编号>指定机器人轴的公差参数变更为<表达式>的值(单位:脉冲)。<机器人编号>可以省略。当进行省略时,机器人1被指定。

形式1:对指定机器人的所有轴进行变更。

形式2:仅对指定机器人的<轴编号>指定轴进行变更。

MEMO

- 在生成过程中,当指定 no axis 中设定的轴时,将会发生“5.37:规格不一致”错误,并且停止程序的执行。
- 此命令语句在指定轴的动作结束后(定位在公差范围内后)执行。

函数

格式

TOLE [<机器人编号>] (<轴编号>)

值 <机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6

含义 赋予<机器人编号>指定机器人轴之中<轴编号>指定轴的公差参数值。<机器人编号>可以省略。当进行省略时,机器人1被指定。

SAMPLE

```
' CYCLE WITH DECREASING TOLERANCE
DIM TOLE (5)
FOR A=200 TO 80 STEP -20
  GOSUB *CHANGE_TOLE
  MOVE P, P0
  MOVE P, P1
NEXT A
C=TOLE (2) ..... 将机器人1中第2轴的公差参数赋给变量C。
HALT
*CHANGE_TOLE:
FOR B=1 TO 4
  TOLE (B) =A
NEXT B
RETURN
```

格式

TORQUE [< 机器人编号 >] (< 轴编号 >) = < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6
 < 表达式 > 1 ~ 100 (单位 : %)

含义 将 < 机器人编号 > 指定机器人的 < 轴编号 > 指定轴的最大扭矩指令值变更为 < 表达式 > 的值。更改后的值将在执行下一个 MOVE 语句或 DRIVE 语句等移动命令时有效。参数值不更改。

扭矩限制的解除条件如下所示。

- 对同一轴执行 TORQUE 命令。
- 切断控制器电源并再次接通电源。
- 进行轴极性参数的变更或者参数的初始化。
- 变为伺服禁止状态。

对扭矩限制值进行暂时变更的条件如下所示。

- 正在执行伴有原点归复的动作。
- 正在执行 PUSH 语句 (仅移动方向可变更)。

在执行下一个 MOVE 语句等移动命令时, 扭矩限制值将返回指定值。

MEMO

- 在生成过程中, 当指定 no axis 中设定的轴时, 将会显示“5.37 : 规格不一致”信息, 并且中止执行。

函数

格式

TORQUE [< 机器人编号 >] (< 轴编号 >)

值 < 机器人编号 > 1 ~ 4
 < 轴编号 > 1 ~ 6

含义 赋予 < 机器人编号 > 指定机器人的 < 轴编号 > 指定轴的扭矩设定值。 < 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。



注意

- 如果指定的值太小, 轴可能无法移动。此时, 请务必设置为紧急停止状态后再进行作业。
- 指定值变为额定扭矩以下时, 碰撞到障碍物时也有可能不出错。

SAMPLE

TORQUE (1) = 50	将机器人 1 中第 1 轴的最大扭矩变更为 50%。
DRIVE (1,P1)	使机器人 1 中第 1 轴从当前位置向 P1 所指定的位置进行移动。 (在开始移动的同时,最大扭矩被变更。)
WAIT ARM	等待机器人 1 中第 1 轴的动作结束。
TORQUE (1) = 100.....	使机器人 1 中第 1 轴的最大扭矩恢复原值 (100%)。
MOVE P,P0	使机器人 1 从当前位置向 P0 所指定的位置进行移动。 (在开始移动的同时,第 1 轴的最大扭矩恢复原值 (100%)。)

相关命令 CURTRQ, PUSH

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

VAL (<字符串表达式>)

含义

将<字符串表达式>中指定的字符串数值转换成实际数值。

字符串中可使用整数标注(二进制形式、十进制形式、十六进制形式)及实数标注(通常的小数点标注、指数形式)。

字符串起始的字符为+、-、&或数字以外时, VAL 的值变为 0。

当字符串中含有表示数字字符以外的字符或空格时, 则该字符后面将被忽略。

但是, 十六进制标注的 A~F 被视作数字。

SAMPLE

```
A=VAL("&B100001")
```

格式

WAIT <条件表达式> [, <表达式>]

值 <表达式> 10 ~ 3600000 (单位 : ms)

含义 等待<条件表达式>成立。通过<表达式>设置超时时间(单位 ms)。设置了超时时间时,即使超过了超时时间且 WAIT 条件不成立时,也结束命令。最小等待执行时间为 1ms,但是根据其它任务的执行情况,该时间会有所变化。

MEMO

· 当条件表达式为表示数值的表达式时,表达式的值非 0 时为“真”,成为 0 时则为“假”。

SAMPLE

WAIT A=10 等待至变量 A 变为 10
 WAIT DI2 () =&B01010110 等待至 DI (21), (22), (24), (26) 变为开启、DI (20), (23), (25), (27) 变为关闭
 WAIT DI2 (4, 3, 2) =&B101 等待至 DI (22) 与 DI (24) 变为开启、DI (23) 变为关闭
 WAIT DI (31) =1 OR DO (21) =1 等待至 DI (31) 与 DO (21) 中的任意一个变为开启
 WAIT DI (20) =1, 1000 等待至 DI (20) 变为开启 经过 1 秒仍未开启时, 结束命令

相关命令 DRIVE, DRIVEI, MOVE, MOVEI

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

WAIT ARM [<机器人编号>][(<轴编号>)]

值 <机器人编号> 1 ~ 4
 <轴编号> 1 ~ 6

含义 等待<机器人编号>指定机器人的轴动作完成（定位于公差内）。<机器人编号>可以省略。当进行省略时，机器人 1 被指定。
 当存在<轴编号>的指定时，指定机器人的指定轴成为对象；当无指定时，指定机器人的所有轴成为对象。

SAMPLE

WAIT ARM 等待机器人 1 的动作结束。
 WAIT ARM[2] (2) 等待机器人 2 中第 2 轴的动作结束。

相关命令 DRIVE, DRIVEI, MOVE, MOVEI

格式

WEIGHT [< 机器人编号 >] < 表达式 >

值 < 机器人编号 > 1 ~ 4
 < 表达式 > 范围因所设置的机器人而异。

含义 将 < 机器人编号 > 指定机器人的前端质量参数变更为 < 表达式 > 的值。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。对附加轴无任何影响。

函数

格式

WEIGHT [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 获取 < 机器人编号 > 指定机器人的前端质量参数值。
 < 机器人编号 > 可以省略。当进行省略时，视为指定了机器人 1。

SAMPLE

```
A=5
B=2
C=WEIGHT
WEIGHT A
MOVE P, P0
WEIGHT B
MOVE P, P1
WEIGHT C
D=WEIGHT ..... 将机器人 1 前端质量参数赋给变量 D。
HALT
```

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

格式

```
WHILE <条件表达式>  
    <命令区>  
WEND
```

含义 结束以 WHILE 语句开始的命令区。WHILE 语句必须成对使用。
也可通过 GOTO 等语句从 WHILE ~ WEND 循环跳转至其他位置。

SAMPLE

```
A=0  
WHILE DI3 (0) =0  
    A=A+1  
    MOVE P, P0  
    MOVE P, P1  
    PRINT "COUNTER=" ; A  
WEND  
HALT
```

相关命令 WHILE

格式

WHERE [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 根据关节坐标获取 < 机器人编号 > 指定机器人的机械臂的当前位置。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

SAMPLE

P10=WHERE 将当前位置的脉冲坐标值赋值给 P10

相关命令 WHRXY

格式

```
WHILE <条件表达式>
  <命令区>
WEND
```

含义

<条件表达式>中指定的条件成立中，反复执 WHILE 语句与 WEND 语句之间的命令区，并返回到 WHILE 语句。

当<条件表达式>中指定的条件不成立（变为“假”）时，将跳转至 WEND 语句的下一个命令语句。

当<条件表达式>中指定的条件最初就不成立（假）时，则一次也不执行 WHILE 语句与 WEND 语句之间的语句，并跳转至 WEND 语句的下一个语句。

通过 GOTO 等语句可从 WHILE ~ WEND 循环内跳转至 WHILE ~ WEND 循环之外。

MEMO

· 当条件表达式为表示数值的表达式时，表达式的值非 0 时为“真”，成为 0 时则为“假”。

SAMPLE 1

```
A=0
WHILE DI3 (0) =0
  A=A+1
  MOVE P, P0
  MOVE P, P1
  PRINT "COUNTER=";A
WEND
HALT
```

SAMPLE 2

```
A=0
WHILE -1 ..... 由于条件表达式始终为真（0 以外），因此将进入永久循环。

  A=A+1
  MOVE P, P0
  IF DI3 (0) =1 THEN *END
  MOVE P, P1
  PRINT "COUNTER=" ; A
  IF DI3 (0) =1 THEN *END
WEND
*END
HALT
```

格式

WHRXY [< 机器人编号 >]

值 < 机器人编号 > 1 ~ 4

含义 根据正交坐标获取 < 机器人编号 > 指定机器人的机械臂的当前位置。 < 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。
当为 YK500TW 机器人时，还要获取 X 机械臂转数信息和 Y 机械臂转数信息。

SAMPLE

P10=WHRXY..... 将机器人 1 的当前位置直行坐标值赋给 P10。

相关命令 WHERE

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

XYTOJ

将正交坐标数据（毫米）转换为关节坐标数据（脉冲）

格式

XYTOJ [< 机器人编号 >] (< 坐标点表达式 >)

值 < 机器人编号 > 1 ~ 4

含义 将 < 坐标点表达式 > 中表示的正交坐标数据（单位：mm、度）转换为 < 机器人编号 > 指定机器人的关节坐标数据（单位：脉冲）。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

- 根据执行时设置的基准坐标、位移坐标及机械手定义转换。
- 请务必注意，根据水平多关节机器人右手系统 / 左手系统的指定，结果会有所不同。
- 当 YK500TW 机器人时，由于 X 机械臂转数信息、Y 机械臂转数信息的指定不同，结果也会出现差异，因此必须加以注意。
- 将关节坐标数据转换为正交坐标数据时，使用 JTOXY。

SAMPLE

P10=XYTOJ (P10)..... 将 P10 转换为关节坐标数据

第 8 章

数据文件详细说明

1	概要	8-1
2	程序文件.....	8-2
3	坐标点文件.....	8-4
4	坐标点注释文件	8-7
5	参数文件.....	8-9
6	位移坐标定义文件	8-12
7	机械手定义文件	8-14
8	托盘定义文件.....	8-16
9	ALL 文件.....	8-20
10	程序目录文件.....	8-21
11	参数目录文件.....	8-23
12	变量文件.....	8-24
13	常量文件.....	8-27
14	数组变量文件.....	8-28
15	DI 文件	8-30
16	DO 文件.....	8-32
17	MO 文件	8-34
18	LO 文件	8-36

19	TO 文件	8-38
20	SI 文件	8-40
21	SO 文件	8-42
22	EOF 文件	8-44
23	串行端口通信文件	8-45
24	SIW 文件	8-46
25	SOW 文件	8-48
26	Ethernet 端口通信文件	8-50

1.1 数据文件种类

在本章节中，将向您介绍在 SEND 语句或联机命令的 READ / WRITE 中可使用的数据文件。数据文件共有以下 25 种。

1. 程序文件
2. 坐标点文件
3. 坐标点注释文件
4. 参数文件
5. 位移坐标定义文件
6. 机械手定义文件
7. 托盘定义文件
8. ALL 文件
9. 程序目录文件
10. 参数目录文件
11. 变量文件
12. 常量文件
13. 数组变量文件
14. DI 文件
15. DO 文件
16. MO 文件
17. LO 文件
18. TO 文件
19. SI 文件
20. SO 文件
21. EOF 文件
22. 串行端口通信文件
23. SIW 文件
24. SOW 文件
25. Ethernet 端口通信文件

1.2 注意点

处理数据文件时，请注意以下要点。

- 可使用的字符全部为半角。
- 数据全部基于 ASCII (美国信息交换标准码) 进行字符编码的字符串处理。
- 命令语句全部仅限大写英语字母 (不可使用小写英语字母)。
- 所有行必须在 255 个字符以内。
- 数据格式的标注中，[cr/lf] 表示 CR 代码 (0Dh)+LF 代码 (0Ah)。
- 本书中读出及写入的语言使用以下数据流程的含义。
读出：控制器 → 外部通信设备
写入：外部通信设备 → 控制器

2.1 整个程序

格式

PGM

含义

- 表示整个程序。
- 对于读出文件，将读出所有已录入的程序。
- 对于写入文件，在控制器中录入 NAME=< 程序名 > 行中指定的程序名。
- 当为写入文件时，如果对程序编号进行了指定，则以新的程序替换。
- 当在写入文件中省略了程序编号时，可分配空置的最小编号。当存在同名但程序编号不同的程序时，较早的程序将被删除。

DATA FORMAT

```

NAME=< 程序名 > [cr/lf]
PGN=mmm
aaaaa ...aaaaaaaaaaaaaaaa [cr/lf]
      :
aaaaa ...aaaaaaaaaaaaaaaa [cr/lf]
      :
NAME=< 程序名 > [cr/lf]
PGN=mmm
aaaaa ...aaaaaaaaaaaaaaaa [cr/lf]
      :
aaaaa ...aaaaaaaaaaaaaaaa [cr/lf]
[cr/lf]

```

值

a..... 字符编码
mmm..... 程序编号：1 ~ 100

- < 程序名 > 采用 32 字符以内的英文字母数字或下划线 () 表示。
- 在文件的末尾仅附加表示结束的 [cr/lf] 行。

SAMPLE

```

SEND PGM TO CMU ..... 从通信端口输出整个程序
SENDCMU TO PGM ..... 从通信端口输入整个程序

```

```

应答：
NAME=TEST [cr/lf]
PGN=1
A=1 [cr/lf]
RESET DO2 ( ) [cr/lf]
      :
HALT [cr/lf]
[cr/lf]

```

格式

1. <程序名>
2. PGmmm

含义

- 表示指定程序。
- mmm 以 1 ~ 100 表示。
- 程序名采用 32 字符以内的英文字母数字或“_”表示，并在外面加上“<”和“>”。
- 当在 1 的形式中未指定程序名称时，将选择当前指定的程序。
- 格式上指定的程序名与数据上的程序名不同时，将出错。

DATA FORMAT

```
NAME= 程序名 [cr/lf]
PGN=mmm
aaaaa ...aaaaaaaaaaaaaaaa [cr/lf]
:
aaaaa ...aaaaaaaaaaaaaaaa [cr/lf]
[cr/lf]
```

值

a..... 字符编码
 mmm..... 程序编号：1 ~ 100

- 程序名采用 32 字符以内的英文字母数字或下划线(_)记述。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

MEMO

- 进行程序的写入时，请必须通过 NAME 语句指定程序名。不指定时，无法对程序进行写入。
- 无法写入当前所选程序。
- 执行次序程序过程中，无法写入程序名 "SEQUENCE"。

SAMPLE

```
SEND < TEST1 > TO CMU ..... 从通信端口输出程序 TEST1
SEND CMU TO < TEST1 > ..... 从通信端口输入程序 TEST1
```

应答：

```
NAME=TEST1 [cr/lf]
A=1 [cr/lf]
RESET DO2 ( ) [cr/lf]
:
HALT [cr/lf]
[cr/lf]
```

3.1 所有坐标点

格式

PNT

含义

- 表示所有坐标点。
- 对于读出文件，将读出所有已录入的坐标点。
- 对于写入文件，通过坐标点编号写入。

DATA FORMAT (YK500TW 以外的机器人)

```
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
:
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
[cr/lf]
```

DATA FORMAT (当机器人为 YK500TW 时)

```
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
:
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
[cr/lf]
```



要点

- 整数值的坐标点数据被识别为脉冲单位，实数值被识别为毫米单位。

值

mmmm 坐标点编号：0 ~ 29999

f 坐标点符号：+/-/ 空格

xxxxxx/./bbbbb 8 位以内的数值。数值中带有. 时，将被识别为毫米单位的坐标系。数据采用 1 个以上的空格隔开。

t 水平多关节型机器人上扩展设置的手系统标志。
1：右手系统、2：左手系统。

xr 扩展设定的第 1 机械臂转数信息。
0：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-180.00^\circ < x \leq 180.00^\circ$
1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $180.00^\circ < x \leq 540.00^\circ$
-1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-540.00^\circ < x \leq -180.00^\circ$

yr 扩展设定的第 2 机械臂转数信息。
0：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-180.00^\circ < y \leq 180.00^\circ$
1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $180.00^\circ < y \leq 540.00^\circ$
-1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-540.00^\circ < y \leq -180.00^\circ$

*1：这是将转换为关节坐标后的脉冲数据，再转换为与各机械臂机械原点间的角度后的数据。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK500TW 且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE (YK500TW 以外的机器人)

SEND PNT TO CMU 从通信端口输出所有坐标点
SEND CMU TO PNT 从通信端口输入所有坐标点

应答：

P0 =	1	2	3	4	5	6	[cr/lf]
P1 =	1.00	2.00	3.00	4.00	5.00	6.00	[cr/lf]
P2 =	1.00	0.00	0.00	0.00	0.00	0.00	[cr/lf]
:							
P29999=	-1.00	0.00	0.00	0.00	0.00	0.00	[cr/lf]
[cr/lf]							

SAMPLE (当机器人为 YK500TW 时)

SEND PNT TO CMU 从通信端口输出所有坐标点
SEND CMU TO PNT 从通信端口输入所有坐标点

应答：

P0 =	1	2	3	4	5	6	[cr/lf]
P1 =	426.20	-160.77	0.01	337.21	0.00	0.00	0 1 0 [cr/lf]
P2 =	-27.57	-377.84	0.36	193.22	0.00	0.00	0 -1 0 [cr/lf]
:							
P29999=	-251.66	-419.51	0.00	-127.79	0.00	0.00	2 -1 -1 [cr/lf]
[cr/lf]							

7

8

9

10

11

格式

Pmmmm

- 含义**
- 表示指定坐标点。
 - mmmmm 通过 0 ~ 29999 表示。

DATA FORMAT (YK500TW 以外的机器人)

Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]

DATA FORMAT (当机器人为 YK500TW 时)

Pmmmm= fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]

值	mmmm	坐标点编号：0 ~ 29999
	f	坐标点符号：+/-/ 空格
	xxxxxx/./bbbbb	8 位数以内的数值。数值中带有 . 时，将被识别为毫米单位的坐标系。数据采用 1 个以上的空格分隔。
	t	水平多关节型机器人上扩展设置的手系统标志。 1：右手系统、2：左手系统。
	xr	YK500TW 机器人的第 1 机械臂转数信息。 0：mm→脉冲转换后的脉冲数据 x(*1) 的范围为 $-180.00^\circ < x \leq 180.00^\circ$ 1：mm→脉冲转换后的脉冲数据 x(*1) 的范围为 $180.00^\circ < x \leq 540.00^\circ$ -1：mm→脉冲转换后的脉冲数据 x(*1) 的范围为 $-540.00^\circ < x \leq -180.00^\circ$
	yr	YK500TW 机器人的第 2 机械臂转数信息。 0：mm→脉冲转换后的脉冲数据 x(*1) 的范围为 $-180.00^\circ < y \leq 180.00^\circ$ 1：mm→脉冲转换后的脉冲数据 x(*1) 的范围为 $180.00^\circ < y \leq 540.00^\circ$ -1：mm→脉冲转换后的脉冲数据 x(*1) 的范围为 $-540.00^\circ < y \leq -180.00^\circ$

*1：这是将转换为关节坐标后的脉冲数据，再转换为与各机械臂机械原点间的角度后的数据。



要点

- 脉冲单位时，坐标点数据为整数，毫米单位时，坐标点数据为实数值。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK500TW 且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。

SAMPLE (YK500TW 以外的机器人)

```
SEND P100 TO CMU ..... 从通信端口输出指定坐标点
SEND CMU TO P100 ..... 从通信端口输入指定坐标点
```

应答：

P100= 1 2 3 4 5 6 [cr/lf]

SAMPLE (当机器人为 YK500TW 时)

```
SEND P100 TO CMU ..... 从通信端口输出指定坐标点
SEND CMU TO P100 ..... 从通信端口输入指定坐标点
```

应答：

P100= 1 2 3 4 5 6 0 1 0 [cr/lf]

4.1 所有坐标点注释

格式

PCM

含义

- 表示所有坐标点注释。
- 对于读出文件，将读出所有已录入的坐标点注释。
- 对于写入文件，通过坐标点注释编号写入。

DATA FORMAT

```
PCmmm= sssssssssssss [cr/lf]
PCmmm= sssssssssssss [cr/lf]
:
PCmmm= sssssssssssss [cr/lf]
PCmmm= sssssssssssss [cr/lf]
[cr/lf]
```

值

mmm.....坐标点编号：0 ~ 29999
 ss...ss注释数据：仅可使用半角、最多 16 个字符。写入数据时
 超过 16 个字符时，第 17 个字符之后将被删除。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

```
SEND PCM TO CMU ..... 从通信端口输出所有坐标点注释
SEND CMU TO PCM ..... 从通信端口输入所有坐标点注释
```

应答：

```
PC1 = ORIGIN POS [cr/lf]
PC3 = WAIT POS [cr/lf]
:
PC3999 = WORK100 [cr/lf]
[cr/lf]
```

格式

PCmmm

- 含义**
- 表示指定坐标点注释。
 - mmmm 通过 0 ~ 29999 表示。

DATA FORMAT

PCmmm= sssssssssss [cr/lf]

- 值**
- mmm 坐标点编号：0 ~ 29999
- ss...ss 注释数据：仅可使用半角、最多 16 个字符。写入数据时超过 16 个字符时，第 17 个字符之后将被删除。

SAMPLE

SEND PC1 TO CMU 从通信端口输出坐标点注释

SEND CMU TO PC1 从通信端口输入坐标点注释

应答：

PC1 = ORIGIN POS [cr/lf]

格式

PRM

含义

- 表示所有参数(包含 UTILITY 中的设置值)。
- 对于读出文件, 将读出所有参数。
- 对于写入文件, 将仅写入标签指定的参数。

DATA FORMAT

```

/ 参数标签 / ' <注释> [cr/lf]
RC= xxxxxx[cr/lf]
/ 参数标签 / ' <注释> [cr/lf]
R1= xxxxxx R2= yyyyyy [cr/lf]
/ 参数标签 / ' <注释> [cr/lf]
R1= xxxxxx, yyyyyy, zzzzzz, rrrrrr[cr/lf]
/ 参数标签 / ' <注释> [cr/lf]
:
[cr/lf]

```

值

RC 表示所有控制器。
R? 机器人指定 : ? : 机器人编号
A 表示轴参数。各轴的数据以逗号进行间隔。
<注释> 参数名称

- 参数标签采用 8 个英文字母表示。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

MEMO

- 写入参数时, 请设置为伺服禁止状态。
- 参数虽然具备与高版本之间的兼容性, 但是有时不具备与低版本之间的兼容性(高版本兼容)。
- 如果将新版本的参数文件载入旧版本的控制器时, 则会发生“10.14: 未定义的参数”错误。这种情况下, 如果在参数设定中将“不要载入未定义的参数”设为有效, 则可以正常地载入除未定义参数之外的其它参数(详细说明, 请参阅控制器的手册)。

SAMPLE

```
SEND PRM TO CMU ..... 从通信端口输出所有参数
SEND CMU TO PRM ..... 从通信端口输入所有参数
```

应答：

```
' V1.01/R0001
```

```
' V1.01/V1.01/V1.01/V1.01/V1.01/V1.01/V1.01/V1.01/
```

```
' -----/-----/-----/-----/-----/-----/-----/-----/
```

```
/RBTNUM/[cr/lf]
```

```
R1=      3000      R2=      3010      [cr/lf]
```

```
/AXSNUM/[cr/lf]
```

```
R1A=    5000,    5001,    5010,    5011    [cr/lf]
```

```
R2A=      0,      0,      0,      0    [cr/lf]
```

```
/ATTRIB/
```

```
R1A=   33792,   33792,   33792,   33792    [cr/lf]
```

```
R1A=     256,     256,     256,     256    [cr/lf]
```

```
/WEIGHT/ '[kg][cr/lf]
```

```
R1=        2      R2=        12      [cr/lf]
```

```
:
```

```
[cr/lf]
```

格式

/ 参数标签 /

含义

- 参数标签采用 8 个英文字母表示。
- 对于读出文件，将仅读出标签指定参数。
- 对于写入文件，将仅写入标签指定参数。

DATA FORMAT 1

```
/ 参数标签 / ' <注释> [cr/lf]
RC= xxxxxx [cr/lf]
[cr/lf]
```

DATA FORMAT 2

```
/ 参数标签 / ' <注释> [cr/lf]
R?= xxxxxx [cr/lf]
[cr/lf]
```

DATA FORMAT 3

```
/ 参数标签 / ' <注释> [cr/lf]
R?= xxxxxx,xxxxxx,xxxxxx,xxxxxx[cr/lf]
[cr/lf]
```

值

RC 表示所有控制器。
 R? 机器人指定：? : 机器人编号
 A 表示轴参数。各轴的数据以逗号进行间隔。
 <注释> 参数名称

- 参数标签采用 8 个英文字母表示。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

MEMO

- 写入参数时，请设置为伺服禁止状态。
- 参数虽然具备与高版本之间的兼容性，但是有时不具备与低版本之间的兼容性(高版本兼容)。
- 如果将新版本的参数数据载入到旧版本的控制器，可能会发生 [10.14 : Undefined parameter found] 的错误。

SAMPLE

```
SEND / ACCEL / TO CMU ..... 从通信端口输出加速度参数
SEND CMU TO / ACCEL / ..... 从通信端口输入加速度参数
```

应答：

```
/ACCEL / '[%]
R1A= 100, 100, 100, 100 [cr/lf]
[cr/lf]
```

6.1 所有位移

格式

SFT

含义

- 表示所有位移。
- 对于读出文件，将读出所有位移。
- 对于写入文件，通过位移编号写入。

DATA FORMAT

```

Sm      = fxxxxxx fyyyyyy fzzzzzz frrrrrr [cr/lf]
SPm     = fxxxxxx fyyyyyy fzzzzzz frrrrrr [cr/lf]
SMm     = fxxxxxx fyyyyyy fzzzzzz frrrrrr [cr/lf]
      :
Sm      = fxxxxxx fyyyyyy fzzzzzz frrrrrr [cr/lf]
SPm     = fxxxxxx fyyyyyy fzzzzzz frrrrrr [cr/lf]
SMm     = fxxxxxx fyyyyyy fzzzzzz frrrrrr [cr/lf]
[cr/lf]

```

值

m..... 位移编号：0～9
 f..... 坐标点符号：+ / - / 空格
 xxxxxx/yyyyy/./rrrrr..... 8 位数以内小数点 2 位以下的数值

- 对于写入文件，可任意输入 SPm 与 SMm。
 SPm：位移坐标范围 + 侧；SMm：位移坐标范围 - 侧
- 在文件的末尾只加上表示文件结束的 [cr/lf] 行。

SAMPLE

```

SEND SFT TO CMU ..... 从通信端口输出所有位移
SEND CMU TO SFT ..... 从通信端口输入所有位移

```

应答：

```

S0 =      0.00      0.00      0.00      0.00 [cr/lf]
SP0 =     0.00      0.00      0.00      0.00 [cr/lf]
SM0 =     0.00      0.00      0.00      0.00 [cr/lf]
S1 =      1.00      1.00      1.00      1.00 [cr/lf]
      :
SM39 =    9.00      9.00      9.00      9.00 [cr/lf]
[cr/lf]

```

格式

Sm

含义 · 表示指定位移。

DATA FORMAT

Sm = fxxxxxx fyyyyyy fzzzzzz frrrrrr [cr/lf]

值

m.....位移编号：0 ~ 39
 f.....坐标点符号：+ / - / 空格
 xxxxxx/yyyyy/./rrrrr.....8 位数以内小数点 2 位以下的数值

SAMPLE

SEND S0 TO CMU..... 从通信端口输出指定位移坐标
 SEND CMU TO S0..... 从通信端口输入指定位移坐标

应答：

S0 =0.00 0.00 0.00 0.00 [cr/lf]
 SP0 =0.00 0.00 0.00 0.00 [cr/lf]
 SM0 =0.00 0.00 0.00 0.00 [cr/lf]
 [cr/lf]

7.1 所有机械手

格式

HND

- 含义**
- 表示所有机械手。
 - 对于读出文件，将读出所有机械手。
 - 对于写入文件，通过机械手编号写入。

DATA FORMAT

```
Hm = n,fxxxxxx, fyyyyyy, fzzzzzz [{R}][cr/lf]
:
Hm = n,fxxxxxx, fyyyyyy, fzzzzzz [{R}][cr/lf]
[cr/lf]
```

- 值**
- m.....机械手编号：0 ~ 31
 - n.....机器人编号：1 ~ 4
 - f.....坐标点符号：+ / - / 空格
 - xxxxxx/yyyyy/zzzzz 8 位数以内小数点 2 位以下的数值或 7 位数以下的整数
值（数值形式取决于机器人类型的设置或机械手定义的种类）
 - {R}.....是否在 R 轴上安装了机械手

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

```
SEND HND TO CMU ..... 从通信端口输出所有机械手
SEND CMU TO HND ..... 从通信端口输入所有机械手
```

应答：

```
H0 = 1, 0.00, 0.00, 0.00 [cr/lf]
H1 = 1, 1.00, 1.00, 1.00 [cr/lf]
H2 = 2, 2.00, 2.00, 2.00 [cr/lf]
H3 = 2, 3.00, 3.00, 3.00 [cr/lf]
H4 = 3, 4.00, 4.00, 4.00 [cr/lf]
H5 = 3, 5.00, 5.00, 5.00 [cr/lf]
H6 = 4, 6.00, 6.00, 6.00 [cr/lf]
H7 = 4, 7.00, 7.00, 7.00 [cr/lf]
[cr/lf]
```

格式

Hm

含义 · 表示指定机械手定义。

DATA FORMAT

Hm = n,fxxxxxx, fyyyyyy, fzzzzz[, {R}][cr/lf]

值	m.....	机械手编号 / 0 ~ 31
	n	机器人编号 : 1 ~ 4
	f	坐标点符号 : + / - / 空格
	xxxxxx/yyyyy/zzzzz	8 位数以内小数点 2 位以下的数值或 7 位数以下的整数值 (数值形式取决于机器人类型的设置或机械手定义的种类)
	{R}.....	是否在 R 轴上安装了机械手

SAMPLE

SEND H3 TO CMU 从通信端口输出指定机械手定义

SEND CMU TO H3 从通信端口输入指定机械手定义

应答 :

H3 = 2, 3.00, 3.00, 3.00 [cr/lf]

8.1 所有托盘定义

格式

PLT

含义

- 表示所有托盘定义。
- 对于读出文件，将读出所有已设置的托盘定义。
- 对于写入文件，通过托盘编号写入。

DATA FORMAT (YK500TW 以外的机器人)

```

PLm [cr/lf]
PLN = XY [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [cr/lf]
PLP = ppppp [cr/lf]
P [1] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbsbbb t [cr/lf]
      :
P [5] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbsbbb t [cr/lf]
PLm [cr/lf]
      :
[cr/lf]

```

DATA FORMAT (当机器人为 YK500TW 时)

```

PLm [cr/lf]
PLN = XY [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [cr/lf]
PLP = ppppp [cr/lf]
P[1] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbsbbb t xr yr[cr/lf]
      :
P[5] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbsbbb t xr yr[cr/lf]
PLm [cr/lf]
      :
[cr/lf]

```

值

- | | |
|-----------------------|--|
| m | 托盘编号：0 ~ 39 |
| nnn | 各轴的坐标点数：正整数 |
| ppppp | 托盘定义中使用的坐标点编号。使用以指定坐标点开头的连续 5 个坐标点。 |
| f | 坐标点符号：+ / - / 空格 |
| xxxxxx/yyyyy/./bbsbbb | 8 位数以内的数值。数值中带有 . 时，将被识别为毫米单位的坐标系。数据采用 1 个以上的空格分隔。 |
| t | 水平多关节型机器人上扩展设置的手系统标志。1：右手系统、2：左手系统。 |


```

xr.....扩展设定的 第 1 机械臂转数信息。
      0 : mm→脉冲转换后的角度数据 x(*1)的范围为 -180.00° < x <= 180.00°
      1 : mm→脉冲转换后的角度数据 x(*1)的范围为 180.00° < x <= 540.00°
      -1 : mm→脉冲转换后的角度数据 x(*1)的范围为 -540.00° < x <= -180.00°

yr.....扩展设定的 第 2 机械臂转数信息。
      0 : mm→脉冲转换后的角度数据 x(*1)的范围为 -180.00° < y <= 180.00°
      1 : mm→脉冲转换后的角度数据 x(*1)的范围为 180.00° < y <= 540.00°
      -1 : mm→脉冲转换后的角度数据 x(*1)的范围为 -540.00° < y <= -180.00°

```

*1 : 这是将转换为关节坐标后的脉冲数据，再转换为与各机械臂机械原点间的角度后的数据。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 在使用了托盘定义的动作中，可忽视手系统标志、第 1 机械臂转数信息、第 2 机械臂转数信息。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK500TW 且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

```

SEND PLT TO CMU ..... 从通信端口输出已定义的所有托盘
SEND CMU TO PLT..... 从通信端口输入已定义的所有托盘

应答：
PL0 [cr/lf]
PLN=XY [cr/lf]
NX = 3 [cr/lf]
NY = 4 [cr/lf]
NZ = 2 [cr/lf]
PLP= 3996[cr/lf]
P [1] = 0.00 0.00 0.00 0.00 0.00 0.00 [cr/lf]
P [2] = 100.00 0.00 0.00 0.00 0.00 0.00 [cr/lf]
P [3] = 0.00 100.00 0.00 0.00 0.00 0.00 [cr/lf]
P [4] = 100.00 100.00 0.00 0.00 0.00 0.00 [cr/lf]
P [5] = 0.00 0.00 50.00 0.00 0.00 0.00 [cr/lf]
PL1 [cr/lf]
PLN= XY [cr/lf]
NX = 3 [cr/lf]
NY = 4 [cr/lf]
NZ = 2 [cr/lf]
PLP= 3991[cr/lf]
P [1] = 0.00 0.00 0.00 0.00 0.00 0.00 [cr/lf]
P [2] = 100.00 100.00 0.00 0.00 0.00 0.00 [cr/lf]
P [3] = 0.00 200.00 0.00 0.00 0.00 0.00 [cr/lf]
P [4] = 100.00 200.00 0.00 0.00 0.00 0.00 [cr/lf]
P [5] = 0.00 0.00 100.00 0.00 0.00 0.00 [cr/lf]
[cr/lf]

```

格式

PLm

- 含义**
- 表示指定托盘定义。
 - m 通过 0 ~ 39 表示。

DATA FORMAT (YK500TW 以外的机器人)

```

PLm [cr/lf]
PLN = XY [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [cr/lf]
PLP = ppppp [cr/lf]
P [1] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
:
P [5] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
[cr/lf]

```

DATA FORMAT (当机器人为 YK500TW 时)

```

PLm [cr/lf]
PLN = XY [cr/lf]
PLP = ppppp [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [cr/lf]
P[1] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr[cr/lf]
:
P[5] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr[cr/lf]
[cr/lf]

```



要点

- 脉冲单位时，坐标点数据为整数，毫米单位时，坐标点数据为实数值。

值

m	托盘编号：0 ~ 39
nnn	各轴的坐标点数：正整数
ppppp	托盘定义中使用的坐标点编号。使用以指定坐标点开头的连续 5 个坐标点。
f	坐标点符号：+ / - / 空格
xxxxxx/yyyyy/./bbbbb	8 位数以内的数值。数值中带有 . 时，将被识别为毫米单位的坐标系。数据采用 1 个以上的空格分隔。
t	水平多关节型机器人上扩展设置的手系统标志。1：右手系统、2：左手系统。
xr	扩展设定的第 1 机械臂转数信息。 0：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-180.00^\circ < x \leq 180.00^\circ$ 1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $180.00^\circ < x \leq 540.00^\circ$ -1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-540.00^\circ < x \leq -180.00^\circ$
yr	扩展设定的第 2 机械臂转数信息。 0：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-180.00^\circ < y \leq 180.00^\circ$ 1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $180.00^\circ < y \leq 540.00^\circ$ -1：mm→脉冲转换后的角度数据 x(*1) 的范围为 $-540.00^\circ < y \leq -180.00^\circ$

*1：这是将转换为关节坐标后的脉冲数据，再转换为与各机械臂机械原点间的角度后的数据。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 在使用了托盘定义的动作中，可忽视手系统标志、第 1 机械臂转数信息、第 2 机械臂转数信息。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK500TW 且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND PL2 TO CMU..... 如下所示从通信端口输出指定托盘定义
 SEND CMU TO PL2..... 如下所示从通信端口输入指定托盘定义

应答：

```

PL2 [cr/lf]
PLN=XY [cr/lf]
NX= 3 [cr/lf]
NY= 3 [cr/lf]
NZ= 2 [cr/lf]
PLP= 3986[cr/lf]
P [1]= 100.00 100.00 50.00 90.00 0.00 0.00 [cr/lf]
P [2]= 200.00 100.00 50.00 90.00 0.00 0.00 [cr/lf]
P [3]= 100.00 200.00 50.00 90.00 0.00 0.00 [cr/lf]
P [4]= 200.00 200.00 50.00 90.00 0.00 0.00 [cr/lf]
P [5]= 100.00 10.00 100.00 90.00 0.00 0.00 [cr/lf]
[cr/lf]

```

9.1 所有 ALL 文件

格式

ALL

含义 表示数据文件中机器人系统动作所需最小限度的文件。

DATA FORMAT

```
[PGM]..... 所有程序格式
NAME=< 程序名 >
aaaaa ...aaaaaaaaaaaaaa [cr/lf]
:
aaaaa ...aaaaaaaaaaaaaa [cr/lf]
[cr/lf]
[PNT] ..... 所有坐标点格式
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
:
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
[cr/lf]
[PCM] ..... 所有坐标点注释格式
PCmmmm= ssssssssssssss [cr/lf]
:
PCmmmm= ssssssssssssss [cr/lf]
[cr/lf]
[PRM] ..... 所有参数格式
/ 参数标签 / ' <注释> [cr/lf]
RC= xxxxxx [cr/lf]
/ 参数标签 / ' <注释> [cr/lf]
R1= xxxxxx R2= yyyyyy [cr/lf]
:
[cr/lf]
[END] ..... ALL 文件的结束
```

**要点**

- 有关各文件的详细说明，请参阅各文件的说明。

MEMO

- 对于写入文件，将根据 [xxx] 决定数据文件的格式，并保存到控制器中。
示例：[HND]···将直至下次出现 [xxx] 为止的文本数据作为所有机械手形式的数据文件保存至控制器中。

SAMPLE

```
SEND ALL TO CMU..... 从通信端口输出所有系统
SEND CMU TO ALL..... 从通信端口输入所有系统
```

10.1 所有程序目录

格式

DIR

含义

- 表示整个程序目录。
- 对于读出文件，将读出所有程序目录信息。
- 无法指定写入文件。

DATA FORMAT

No.	Name	Line	Byte	RW/RO	Date	Time [cr/lf]
nnnfgssssssss		llll	bbbbbb	xx	yy/mm/dd	hh:mm [cr/lf]
:						
nnnfgssssssss		llll	bbbbbb	xx	yy/mm/dd	hh:mm [cr/lf]
END[cr/lf]						

值

nnn	程序目录的编号：3 位数
f	如果对次序程序进行了编译且创建了次序对象，则为 "s"
g	如果是当前所选程序，则为 "*"
ssssssss	程序名：32 位数
llll	程序的行数：4 位数
bbbbbb	程序的使用字节数：6 位数
xx	文件属性：2 位数
		RW：可读写
		RO：不可写入
yy/mm/dd	更新日期：8 位数（包含 "/"）
hh:mm	更新时间：5 位数

SAMPLE

SEND DIR TO CMU 从通信端口输出已创建的所有程序信息

应答：

No.	Name	Line	Byte	RW/RO	Date	Time [cr/lf]
1o*	12345678	5	21	RW	01/06/20	10:35 [cr/lf]
2	PGM1	5	66	RW	01/06/20	10:35 [cr/lf]
3	PGM2	5	66	RW	01/06/20	10:35 [cr/lf]
4	PGM3	5	66	RW	01/06/20	10:35 [cr/lf]
5	PGM4	5	66	RW	01/06/20	10:35 [cr/lf]
6	PGM5	5	66	RW	01/06/20	10:35 [cr/lf]
7	PGM6	5	66	RW	01/06/20	10:35 [cr/lf]
8s	SEQUENCE	1	15	RW	01/06/20	10:35 [cr/lf]
END [cr/lf]						

格式

```
" << " <程序名> " >> "
```

- 含义**
- 表示 1 个程序信息。
 - 程序名外面加上 << >> 记述。

DATA FORMAT

No.	Name	Line	Byte	RW/RO	Date	Time [cr/lf]
nnn	fgsssssss	llll	bbbbbb	xx	yy/mm/dd	hh:mm [cr/lf]

值

nnn 程序目录的编号：3 位数

f 如果对次序程序进行了编译且创建了次序对象，则为 "s"

g 如果是当前所选程序，则为 "*"

sssssss 程序名：32 位数

llll 程序的行数：4 位数

bbbbbb 程序的使用字节数：6 位数

xx 文件属性：2 位数
RW：可读写
RO：不可写入

yy/mm/dd 更新日期：8 位数（包含 "/"）

hh:mm 更新时间：5 位数

MEMO

- 程序对象表示专为机器人程序编译的执行程序，次序对象表示专为次序程序编译的执行程序。

SAMPLE

```
SEND << TEST >> TO CMU ... 从通信端口输出指定程序信息
```

应答：

No.	Name	Line	Byte	RW/RO	Date	Time [cr/lf]
30*	PGM2	5	66	RW	01/06/20	10:35 [cr/lf]

11.1 所有参数目录

格式

DPM

含义

- 表示所有参数目录。
- 对于读出文件，将读出所有参数目录信息。
- 无法在写入文件中指定。

DATA FORMAT

```

/ 参数标签 / ' <注释> [cr/lf]
/ 参数标签 / ' <注释> [cr/lf]
:
/ 参数标签 / ' <注释> [cr/lf]
[cr/lf]

```

值

<注释> 参数名称

- 参数标签采用 6 个英文字母表示。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND DPM TO CMU 从通信端口输出已创建的所有参数信息

应答：

```

/RBTNUM/ 'Robot number (V8.01/R1001) [cr/lf]
/AXES/ 'Number of axes [cr/lf]
/AXSNUM/ 'Axis number (V1.01/V1.01/V1.01/V1.01/-----/-----/-----/-----) [cr/lf]
/ATTRIB/ 'Axis attribute [cr/lf]
/WEIGHT/ 'Tip weight [kg] [cr/lf]
/ORIGIN/ 'Origin sequence [cr/lf]
/RORIEN/ 'R axis orientation [cr/lf]
:
/CURPNO/ 'Port number of output [cr/lf]
/CURPT1/ 'Compare point number 1 [cr/lf]
/CURPT2/ 'Compare point number 2 [cr/lf]
[cr/lf]

```

12 变量文件

12.1 所有变量

格式

VAR

意味

- 表示所有全局变量。
- 对于读出文件，将读出所有全局变量。
- 对于写入文件，通过指定全局变量写入。

DATA FORMAT

<变量名>t = xxxxxx [cr/lf]

<变量名>t = xxxxxx [cr/lf]

:

<变量名>t = xxxxxx [cr/lf]

[cr/lf]

值

<变量名> 程序内部定义的全局变量。变量名通过采用英文字母数字或"_"的32字符以内的字符串表示。

t 变量类型 / !: 实数、%: 整数、\$: 字符串

xxxxxx 变量的值

整数型: 8位数以下的整数值

实数型: 整数部分 + 小数部分为7位数以下的数值

字符型: 255个字符以下的字符串

- 在文件的末尾仅附加表示文件结束的[cr/lf]行。

SAMPLE 1

SEND VAR TO CMU..... 从通信端口输出所有全局变量

应答：

```
SGI0=0 [cr/lf]
SGI1=1111 [cr/lf]
SGI2=2222 [cr/lf]
SGI3=3333 [cr/lf]
SGI4=4444 [cr/lf]
SGI5=5555 [cr/lf]
SGI6=6666 [cr/lf]
SGI7=7777 [cr/lf]
SGR0=0 [cr/lf]
SGR1=1.1111E3 [cr/lf]
SGR2=2.2222E3 [cr/lf]
SGR3=3.3333E3 [cr/lf]
SGR4=4.4444E3 [cr/lf]
SGR5=5.5555E3 [cr/lf]
SGR6=6.6666E3 [cr/lf]
SGR7=7.7777E3 [cr/lf]
B1%=111 [cr/lf]
B2%=222 [cr/lf]
C1$="CNS_1" [cr/lf]
C2$="CNS_2" [cr/lf]
[cr/lf]
```

SAMPLE 2

SEND CMU TO VAR..... 从通信端口输入所有全局变量

格式

< 变量名 > t

含义 · 表示 1 个变量。

DATA FORMAT

xxxxxx [cr/lf]

值 < 变量名 > 程序内部定义的全局变量。变量名通过采用英文字母数字或 "_" 的 32 字符以内的字符串表示。

t 变量类型 / !: 实数、%: 整数、\$: 字符串

xxxxxx 变量的值

 整数型: 8 位数以下的整数值

 实数型: 整数部分 + 小数部分为 7 位数以下的数值

 字符型: 255 个字符以下的字符串

MEMO

· 在执行程序时, 可对动态全局变量进行登录。无法参照未录入的变量。

SAMPLE 1

SEND SGI6 TO CMU [cr/lf] 从通信端口输出指定变量 SGI6

应答:

6666 [cr/lf]

SAMPLE 2

SEND CMU TO SGI6 [cr/lf] 从通信端口对指定变量 SGI6 输入值

应答:

6666 [cr/lf] 将数据输入至控制器

OK [cr/lf] 从控制器输出结果



要点

- SGIx 表示整数型静态变量。
- SGRx 表示实数型静态变量。

13.1 单个字符串

格式

" <字符串> "

含义

- 表示指定的字符串。
- 对于读出文件，将读出指定字符串。
- 无法在写入文件中指定。

DATA FORMAT

sssss...sssss [cr/lf]

值

sssss...sssss.....字符串：255 个字符以内

- 双引号 (") 输出应采用双引号 (") 连续记述。

SAMPLE

```
SEND ""YAMAHA ROBOT"" TO CMU
..... 从通信端口输出指定字符串
应答：
"YAMAHA ROBOT" [cr/lf]
```

14.1

所有数组变量

格式

ARY

意味

- 表示所有数组变量。
- 对于读出文件，将读出所有数组变量。
- 对于写入文件，通过指定数组变量写入。

DATA FORMAT

```
<变量名>t (l{, m{, n}}) = xxxxxx [cr/lf]
<变量名>t (l{, m{, n}}) = xxxxxx [cr/lf]
:
<变量名>t (l{, m{, n}}) = xxxxxx [cr/lf]
[cr/lf]
```

值

<变量名> 在程序内部通过 DIM 语句定义的全局变量。变量名通过采用英文字母数字或 "_" 的 32 字符以内的字符串表示。

t 变量类型 / !: 实数、%: 整数、\$: 字符串

l, m, n 数组自变量

xxxxxx 变量的值。根据数组变量的类型不同而异。

 整数型: 8 位数以下的整数

 实数型: 整数部分 + 小数部分为 7 位数以下的数值

 字符型: 255 个字符以下的字符串

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE 1

```
SEND ARY TO CMU ..... 从通信端口输出所有全局数组变量
应答:
A! (0) =0 [cr/lf]
A! (1) =1.E2 [cr/lf]
A! (2) =2.E2 [cr/lf]
B% (0, 0) =0 [cr/lf]
B% (0, 1) =1111 [cr/lf]
B% (1, 0) =2222 [cr/lf]
B% (1, 1) =3333 [cr/lf]
C$ (0, 0, 0) ="ARY1" [cr/lf]
C$ (0, 0, 1) ="ARY2" [cr/lf]
C$ (0, 1, 0) ="ARY3" [cr/lf]
C$ (0, 1, 1) ="ARY4" [cr/lf]
C$ (1, 0, 0) ="ARY5" [cr/lf]
C$ (1, 0, 1) ="ARY6" [cr/lf]
C$ (1, 1, 0) ="ARY7" [cr/lf]
C$ (1, 1, 1) ="ARY8" [cr/lf]
[cr/lf]
```

SAMPLE 2

```
SEND CMU TO ARY..... 从通信端口输入所有全局数组变量
```

格式

<变量名> t (l {, m {, n } })

含义 · 表示 1 个数组变量。

DATA FORMAT

xxxxxx [cr/lf]

值 <变量名> 在程序内部通过 DIM 语句定义的全局变量。变量名通过采用英文字母数字或 "_" 的 32 字符以内的字符串表示。

t 变量类型 / !: 实数、%: 整数、\$: 字符串

l, m, n 数组自变量

xxxxxx 变量的值。根据数组变量的类型不同而异。

 整数型: 8 位数以下的整数值

 实数型: 整数部分 + 小数部分为 7 位数以下的数值

 字符型: 255 个字符以下的字符串

 MEMO

· 在 DIM 语句中定义的数组变量在程序编译时录入。无法参照未录入的数组变量。

SAMPLE 1

SEND C1\$ (2) TO CMU [cr/lf] 从通信端口输出指定数组变量 C1\$ (2)

应答:

YAMAHA ROBOT [cr/lf]

SAMPLE 2

SEND CMU TO C1\$ (2) [cr/lf] 从通信端口输入指定数组变量 C1\$ (2)

应答:

OK [cr/lf]

15.1 所有 DI

格式

DI ()

- 含义**
- 表示所有 DI (并行输入变量)。
 - 对于读出文件, 将读出所有 DI 信息。
 - 无法在写入文件中指定。

DATA FORMAT

```
DI0 ( ) = &Bnnnnnnnn [cr/lf]
DI1 ( ) = &Bnnnnnnnn [cr/lf]
      :
DI27 ( ) = &Bnnnnnnnn [cr/lf]
[cr/lf]
```

- 值** n..... 0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

```
SEND DI ( ) TO CM ..... 从通信端口输出所有 DI
```

```
应答 :
DI0 ( ) = &B10001001 [cr/lf]
DI1 ( ) = &B00000010 [cr/lf]
DI2 ( ) = &B00000000 [cr/lf]
      :
DI7 ( ) = &B00000000 [cr/lf]
DI10 ( ) = &B00000000 [cr/lf]
DI11 ( ) = &B00000000 [cr/lf]
DI12 ( ) = &B00000000 [cr/lf]
      :
DI17 ( ) = &B00000000 [cr/lf]
DI20 ( ) = &B00000000 [cr/lf]
      :
DI26 ( ) = &B00000000 [cr/lf]
DI27 ( ) = &B00000000 [cr/lf]
[cr/lf]
```

格式

```
DI $m$  ( )
```

含义

- 表示 1 个 DI 端口的状态。
- 对于读出文件，将读出指定的 DI 端口。
- 无法在写入文件中指定。

DATA FORMAT

```
DI $m$  ( ) = &B $n$  [cr/lf]
```

值

m 0 ~ 7、10 ~ 17、20 ~ 27
 n 0 或 1 (合计 8 位数)。从左开始对应 m_7, m_6, \dots, m_0
 (m 为端口编号)。

SAMPLE

```
SEND DI5 ( ) TO CMU ..... 从通信端口输出 DI5 端口
```

应答：

```
DI5 ( ) = &B00000000 [cr/lf]
```

16.1 所有 DO

格式

DO ()

- 含义**
- 表示所有 DO (并行输出变量)。
 - 对于读出文件, 将读出所有 DO 信息。
 - 无法在写入文件中指定。

DATA FORMAT

```
DO0 ( ) = &Bnnnnnnnn [cr/lf]
DO1 ( ) = &Bnnnnnnnn [cr/lf]
      :
DO27 ( ) = &Bnnnnnnnn [cr/lf]
[cr/lf]
```

值 n.....0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND DO () TO CMU 从通信端口输出所有 DO

```
应答 :
DO0 ( ) = &B10001001 [cr/lf]
DO1 ( ) = &B00000010 [cr/lf]
DO2 ( ) = &B00000000 [cr/lf]
      :
DO7 ( ) = &B00000000 [cr/lf]
DO10 ( ) = &B00000000 [cr/lf]
DO11 ( ) = &B00000000 [cr/lf]
DO12 ( ) = &B00000000 [cr/lf]
      :
DO17 ( ) = &B00000000 [cr/lf]
DO20 ( ) = &B00000000 [cr/lf]
      :
DO26 ( ) = &B00000000 [cr/lf]
DO27 ( ) = &B00000000 [cr/lf]
[cr/lf]
```


格式

DOm ()

- 含义**
- 表示 1 个 DO 端口的状态。
 - 对于读出文件，将读出指定的 DO 端口。
 - 对于写入文件，值将被写入至指定的 DO 端口。但是，无法写入至 DO0() 与 DO1()。
- 读出文件

DATA FORMAT

DOm () = &Bnnnnnnnn [cr/lf]

- 写入文件

DATA FORMAT

&Bnnnnnnnn [cr/lf] 或 k [cr/lf]

- 值**
- m..... 端口编号：0 ~ 7、10 ~ 17、20 ~ 27
- n..... 0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0。
- k..... 0 ~ 255 的整数

MEMO

- 无法写入至 DO0 () 与 DO1 ()。仅限参照。

SAMPLE 1

SEND DO5 () TO CMU 从通信端口输出 DO5 端口的状态

应答：

DO5 () = &B00000000 [cr/lf]

SAMPLE 2

SEND CMU TO DO5 () 从通信端口输入 DO5 端口
&B00000111

应答：

OK [cr/lf]

17.1 所有 MO

格式

MO ()

- 含义**
- 表示所有 MO (内部输出变量)。
 - 对于读出文件, 将读出所有 MO 信息。
 - 无法在写入文件中指定。

DATA FORMAT

```
MO0 ( ) = &Bnnnnnnnn [cr/lf]
MO1 ( ) = &Bnnnnnnnn [cr/lf]
      :
MO33 ( ) = &Bnnnnnnnn [cr/lf]
[cr/lf]
```

值 n.....0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

```
SEND MO ( ) TO CMU..... 从通信端口输出所有 MO
```

应答:

```
MO0 ( ) = &B10001001 [cr/lf]
MO1 ( ) = &B00000010 [cr/lf]
MO2 ( ) = &B00000000 [cr/lf]
      :
MO7 ( ) = &B00000000 [cr/lf]
MO10 ( ) = &B00000000 [cr/lf]
MO11 ( ) = &B00000000 [cr/lf]
MO12 ( ) = &B00000000 [cr/lf]
      :
MO17 ( ) = &B00000000 [cr/lf]
MO20 ( ) = &B00000000 [cr/lf]
      :
MO32 ( ) = &B00000000 [cr/lf]
MO33 ( ) = &B00000000 [cr/lf]
[cr/lf]
```

格式

MOm ()

含义

- 表示 1 个 MO 端口的状态。
- 对于读出文件，将读出指定的 MO 端口。
- 对于写入文件，值将被写入至指定的 MO 端口。但是，无法写入至 MO30() ~ MO37()。

- 读出文件

DATA FORMAT

MOm () =&Bnnnnnnnn [cr/lf]

- 写出文件

DATA FORMAT

&Bnnnnnnnn [cr/lf] 或 k [cr/lf]

值

m..... 端口编号：0 ~ 7、10 ~ 17、20 ~ 27、30 ~ 37
 n..... 0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0。
 k..... 0 ~ 255 的整数

MEMO

- 无法写入至 MO30() 与 MO37()。仅限参照。

SAMPLE 1

SEND MO5 () TO CMU..... 从通信端口输出 MO5 端口的状态

应答：

MO5 () =&B00000000 [cr/lf]

SAMPLE 2

SEND CMU TO MO5 ()..... 从通信端口输入 MO5 端口
&B00000111

应答：

OK [cr/lf]

18 LO 文件

18.1 所有 LO

格式

LO ()

- 含义**
- 表示所有 LO。
 - 对于读出文件，将读出所有 LO 信息。
 - 无法在写入文件中指定。

DATA FOMAT

```
LO0 ( ) =&Bnnnnnnnn [cr/lf]
LO1()=&Bnnnnnnnn [cr/lf]
[cr/lf]
```

值 n.....0 或 1 (合计 8 位数)。从左开始对应 07, 06, ..., 00。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND LO () TO CMU 从通信端口输出所有 LO 的状态

应答：

```
LO0 ( ) =&B10001001 [cr/lf]
LO1()=&B00100100[cr/lf]
[cr/lf]
```

格式

LOm ()

- 含义**
- 表示 1 个 LO 端口的状态。
 - 对于读出文件，将读出指定的 LO 端口。
 - 对于写入文件，值将被写入至指定的 LO 端口。
- 读出文件

DATA FORMAT

LOm ()=&Bnnnnnnnn[cr/lf]

- 写入文件

DATA FORMAT

&Bnnnnnnnn [cr/lf] 或 k [cr/lf]

- 值**
- m..... 端口编号：0、1
 - n..... 0 或 1 (合计 8 位数)。从左开始对应 07, 06, ..., 00。
 - k..... 0 ~ 255 的整数

SAMPLE 1

SEND LO0 () TO CMU..... 从通信端口输出 LO0 端口的状态

应答：

LO0 () =&B00000000 [cr/lf]

SAMPLE 2

SEND CMU TO LO0 ()..... 从通信端口输入 LO0 端口
&B00000111

应答：

OK [cr/lf]

19 TO 文件

19.1 所有 TO

格式

TO ()

- 含义**
- 表示所有 TO(定时器输出变量)。
 - 对于读出文件, 将读出所有 TO 信息。
 - 无法在写入文件中指定。

DATA FORMAT

```
TO0 ( ) =&Bnnnnnnnn [cr/lf]
TO1 ( ) =&Bnnnnnnnn [cr/lf]
[cr/lf]
```

值 n.....0 或 1(合计 8 位数)。从左开始对应 07, 06, ..., 00。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND TO () TO CMU 从通信端口输出所有 TO 的状态

应答:

```
TO0 ( ) =&B10001001 [cr/lf]
TO1 ( ) =&B10001001 [cr/lf]
[cr/lf]
```

格式

```
TOm ( )
```

含义

- 表示 1 个 TO 端口的状态。
 - 对于读出文件，将读出指定的 TO 端口。
 - 对于写入文件，值将被写入至指定的 TO 端口。
- 读出文件

DATA FORMAT

```
TOm ( ) = &Bnnnnnnnn [cr/lf]
```

- 写入文件

DATA FORMAT

```
&Bnnnnnnnn [cr/lf] 或 k [cr/lf]
```

值

m..... 端口编号：0、1
 n..... 0 或 1 (合计 8 位数)。从左开始对应 07, 06, ..., 00。
 k..... 0 ~ 255 的整数

SAMPLE 1

```
SEND TO0 ( ) TO CMU..... 从通信端口输出 TO0 端口的状态
```

应答：

```
TO0 ( ) = &B00000000 [cr/lf]
```

SAMPLE 2

```
SEND CMU TO TO0 ( )..... 从通信端口输入 TO0 端口  
&B00000111
```

应答：

```
OK [cr/lf]
```

20.1 所有 SI

格式

SI ()

- 含义**
- 表示所有 SI (串行输入变量)。
 - 对于读出文件, 将读出所有 SI 信息。
 - 无法在写入文件中指定。

DATA FORMAT

```
SI0 ( ) = &Bnnnnnnnn [cr/lf]
SI1 ( ) = &Bnnnnnnnn [cr/lf]
      :
SI27 ( ) = &Bnnnnnnnn [cr/lf]
[cr/lf]
```

- 值** n.....0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

```
SEND SI ( ) TO CMU..... 从通信端口输出所有 SI 的状态
```

应答 :

```
SI0 ( ) = &B10001001 [cr/lf]
SI1 ( ) = &B00000010 [cr/lf]
SI2 ( ) = &B00000000 [cr/lf]
      :
SI7 ( ) = &B00000000 [cr/lf]
SI10 ( ) = &B00000000 [cr/lf]
SI11 ( ) = &B00000000 [cr/lf]
SI12 ( ) = &B00000000 [cr/lf]
      :
SI17 ( ) = &B00000000 [cr/lf]
SI20 ( ) = &B00000000 [cr/lf]
      :
SI26 ( ) = &B00000000 [cr/lf]
SI27 ( ) = &B00000000 [cr/lf]
[cr/lf]
```


格式

SI m ()

含义

- 表示 1 个 SI 端口的状态。
- 对于读出文件，将读出指定的 SI 端口。
- 无法在写入文件中指定。

DATA FORMAT

SI m () = &Bnnnnnnnn [cr/lf]

值

m..... 端口编号 : 0 ~ 7、10 ~ 17、20 ~ 27
 n..... 0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0。

SAMPLE

SEND SI5 () TO CMU..... 从通信端口输出 SI5 端口的状态

应答 :

SI5 () = &B00000000 [cr/lf]

21.1 所有 SO

格式

SO ()

- 含义**
- 表示所有 SO(串行输出变量)。
 - 对于读出文件, 将读出所有 SO 信息。
 - 无法在写入文件中指定。

DATA FORMAT

```
SO0 ( ) =&Bnnnnnnnn [cr/lf]
SO1 ( ) =&Bnnnnnnnn [cr/lf]
      :
SO27 ( ) =&Bnnnnnnnn [cr/lf]
[cr/lf]
```

- 值** n.....0 或 1(合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND SO () TO CMU 从通信端口输出所有 SO 的状态

应答 :

```
SO0 ( ) =&B10001001 [cr/lf]
SO1 ( ) =&B00000010 [cr/lf]
SO2 ( ) =&B00000000 [cr/lf]
      :
SO7 ( ) =&B00000000 [cr/lf]
SO10 ( ) =&B00000000 [cr/lf]
SO11 ( ) =&B00000000 [cr/lf]
SO12 ( ) =&B00000000 [cr/lf]
      :
SO17 ( ) =&B00000000 [cr/lf]
SO20 ( ) =&B00000000 [cr/lf]
      :
SO26 ( ) =&B00000000 [cr/lf]
SO27 ( ) =&B00000000 [cr/lf]
[cr/lf]
```

格式

SOm ()

- 含义**
- 表示 SO 端口单位的输出状态。
 - 对于读出文件，将读出指定的 SO 端口。
 - 对于写入文件，值将被写入至指定的 SO 端口。但是，无法写入至 SO0 () 与 SO1 ()。
- 读出文件

DATA FORMAT

SOm () =&Bnnnnnnnn [cr/lf]

- 写入文件

DATA FORMAT

&Bnnnnnnnn [cr/lf] 或 k [cr/lf]

- 值**
- m..... 端口编号：0 ~ 7、10 ~ 17、20 ~ 27
 - n..... 0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0。
 - k..... 0 ~ 255 的整数

MEMO

- 无法写入至 SO0 () 与 SO1 ()。仅限参照。

SAMPLE 1

SEND SO5 () TO CMU..... 从通信端口输出 SO5 端口的状态

应答：

SO5 () =&B00000000 [cr/lf]

SAMPLE 2

SEND CMU TO SO5 ()..... 从通信端口输入 SO5 端口
&B00000111

应答：

OK [cr/lf]

22.1 EOF 数据

格式

EOF

含义

- 此文件是仅限 ^Z(= 1Ah) 的特殊文件。使用通信向外部装置发送数据时，在文件结束时使用 ^Z。
- 对于读出文件，将读出 ^Z(= 1Ah)。
- 无法在写入文件中指定。

DATA FORMAT

^Z (=1Ah)

SAMPLE

```
SEND PGM TO CMU
SEND EOF TO CMU ..... 从通信端口输出 EOF 数据

NAME=TEST1 [cr/lf]
A=1 [cr/lf]
:
HALT [cr/lf]
[cr/lf]
^Z
```

MEMO

- 根据数据接收侧设备、应用程序的规格不同，在发送文件的最后有必要用 ^Z 时使用。

23.1 串行端口通信文件

格式

CMU

含义

- 表示串行通信端口。
- 取决于各种数据格式。

SAMPLE

```
SEND PNT TO CMU ..... 从通信端口输出所有坐标点  
SEND CMU TO PNT ..... 从通信端口输入所有坐标点
```

24.1 所有 SIW

格式

SIW ()

含义

- 表示所有 SIW(串行字输入)。
- 对于读出文件, 将以十六进制形式读出所有 SIW 信息。
- 无法在写入文件中指定。

DATA FORMAT

SIW (0) =&Hnnnn [cr/lf]

SIW (1) =&Hnnnn [cr/lf]

:

SIW (15) =&Hnnnn [cr/lf]

[cr/lf]

值

n.....0~9, A~F: 4 位数(十六进制)

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND SIW () TO CMU 从通信端口输出所有 SIW 的状态

应答:

SIW (0) =&H1001 [cr/lf]

SIW (1) =&H0010 [cr/lf]

SIW (2) =&H0000 [cr/lf]

:

SIW (15) =&H0000 [cr/lf]

[cr/lf]

格式

SIW (m)

含义

- 表示 1 个 SIW 的状态。
- 对于读出文件，将以十六进制形式读出指定 SIW。
- 无法在写入文件中指定。

DATA FORMAT

SIW (m) =&Hnnnn [cr/lf]

值

m.....0 ~ 15
 n.....0 ~ 9, A ~ F : 4 位数 (十六进制)

SAMPLE

SEND SIW (5) TO CMU..... 从通信端口输出 SIW (5) 的状态

应答：

SIW (5) =&H1001 [cr/lf]

25.1 所有 SOW

格式

SOW ()

- 含义**
- 表示所有 SOW(串行字输出)。
 - 对于读出文件, 将以十六进制形式读出所有 SOW 信息。
 - 无法在写入文件中指定。

DATA FORMAT

SOW (0) =&Hnnnn [cr/lf]

SOW (1) =&Hnnnn [cr/lf]

:

SOW (15) =&Hnnnn [cr/lf]

[cr/lf]

值 n..... 0 ~ 9, A ~ F 的 4 位数(十六进制)

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

SEND SOW () TO CMU 从通信端口输出所有 SOW 的状态

应答:

SOW (0) =&H1001 [cr/lf]

SOW (1) =&H0010 [cr/lf]

SOW (2) =&H0000 [cr/lf]

:

SOW (15) =&H0000 [cr/lf]

[cr/lf]

格式

SOW (m)

含义

- 表示 1 个 SOW 的状态。
 - 对于读出文件，将以十六进制形式读出指定 SOW。
 - 对于写入文件，值将被写入至指定 SOW。但是，无法写入至 SOW (0) 与 SOW (1)。
- 读出文件

DATA FORMAT

SOW (m) =&Hnnnn [cr/lf]

- 写入文件

DATA FORMAT

&Hnnnn

值

m.....2 ~ 15
 n.....0 ~ 9, A ~ F 的 4 位数 (十六进制)

SAMPLE 1

SEND SOW (5) TO CMU..... 从通信端口输出 SOW (5) 的状态

应答：

SOW (5) =&H1001 [cr/lf]

SAMPLE 2

SEND CMU TO SOW (5)..... 从通信端口输入至 SOW (5)
&H1001

应答：

OK [cr/lf]

26.1 Ethernet 端口通信文件

格式

ETH

含义

- 表示 Ethernet 端口。
- 取决于各种数据格式。

SAMPLE

SEND PNT TO ETH..... 从 Ethernet 端口输出所有坐标点
SEND ETH TO PNT..... 从 Ethernet 端口输入所有坐标点

第 9 章

用户程序示例

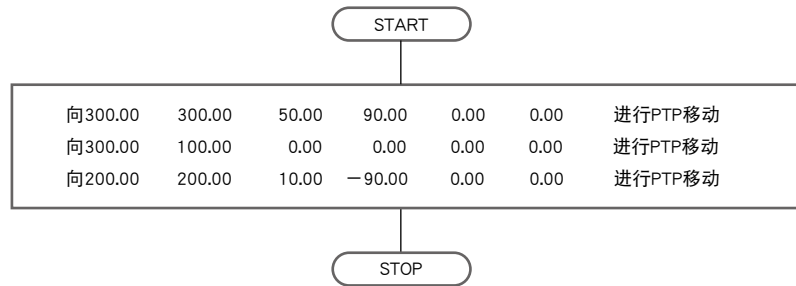
1	基本篇	9-1
2	应用篇	9-8

1.1 将坐标点数据直接写入程序中

■ 概要

通过在程序中直接指定坐标点数据，使机器人机械臂进行 PTP 动作。

处理流程



35C01-R7-00

SAMPLE

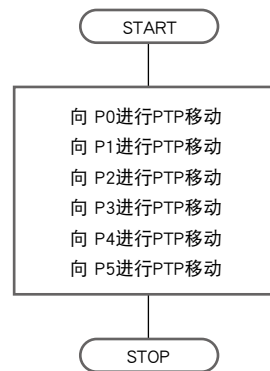
MOVE P,	300.00	300.00	50.00	90.00	0.00	0.00
MOVE P,	300.00	100.00	0.00	0.00	0.00	0.00
MOVE P,	200.00	200.00	10.00	-90.00	0.00	0.00
HALT						

■ 概要

在程序中使用坐标点编号指定坐标值。在坐标点 (MANUAL > POINT) 模式中预先输入坐标值。请输入以下数据。

POINT DATA						
P0=	0.00	0.00	0.00	0.00	0.00	0.00
P1=	100.00	0.00	150.00	30.00	0.00	0.00
P2=	0.00	100.00	50.00	0.00	0.00	0.00
P3=	300.00	300.00	0.00	0.00	0.00	0.00
P4=	300.00	100.00	100.00	90.00	0.00	0.00
P5=	200.00	200.00	0.00	0.00	0.00	0.00

处理流程



35C02-R7-00

SAMPLE 1

```

MOVE P, P0
MOVE P, P1
MOVE P, P2
MOVE P, P3
MOVE P, P4
MOVE P, P5
HALT
  
```

SAMPLE 2

```

FOR J=0 TO 5
  MOVE P, P [J]
NEXT J
HALT
  
```

虽然 SAMPLE1 与 SAMPLE2 的动作相同，但是使用坐标点编号与 FOR 语句可使程序缩短。

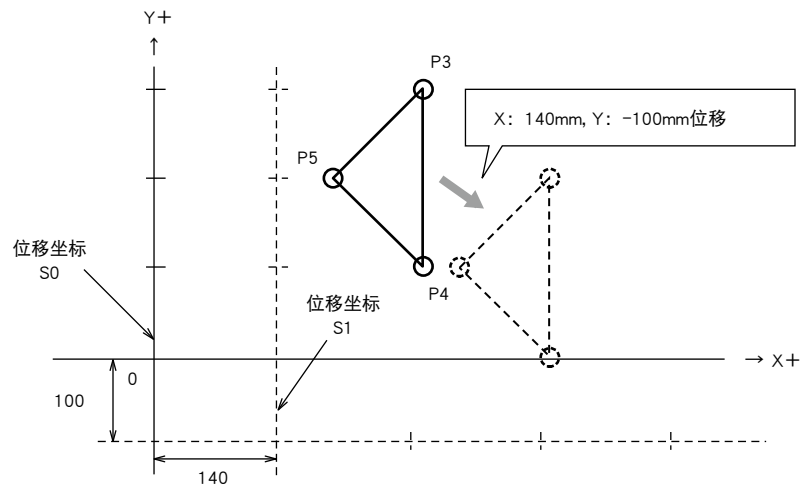
■ 概要

如下图所示，从 P3 至 P5 进行 PTP 移动后，向 X 方向位移 +140，向 Y 轴方向位移 -100，并再次从 P3 至 P5 进行 PTP 移动。位移坐标预先设置为 S1，P3, P4, P5 使用前项“1.2：使用坐标点编号”的设置。

SHIFT DATA

S0=	0.00	0.00	0.00	0.00
S1=	140.00	-100.00	0.00	0.00

位移坐标



35C03-R7-00

SAMPLE

```

SHIFT S0 ..... 位移 0
FOR J=3 TO 5 ..... 在 P3 ~ P5 之间反复移动
    MOVE P, P [J]
NEXT J
SHIFT S1 ..... 更改为位移 1
    FOR K=3 TO 5 ..... 相同地在 P3 ~ P5 之间反复移动
        MOVE P, P [K]
    NEXT K

```

1.4.1 算出坐标点坐标

■ 概要

创建使等间距排列的托盘上的各坐标点与工件供给位置 P0 交互移动的程序。

在下图上, N1 ~ N20 在正交坐标上排列着向 X 方向按照 50mm 间距的 5 个点, 向 Y 方向按照 25mm 的间距的 4 个点。机械臂从一个坐标点移动到另一个坐标点, 按照 P0 N1 P0 N2...N5 P0 - N6 - P0... 的顺序依次在 P0 与各点之间往返移动。

POINT DATA

工件供给位置:

P0= 0.0 0.0 0.0 0.0 0.0 0.0

X 方向的间距:

P10= 50.0 0.0 0.0 0.0 0.0 0.0

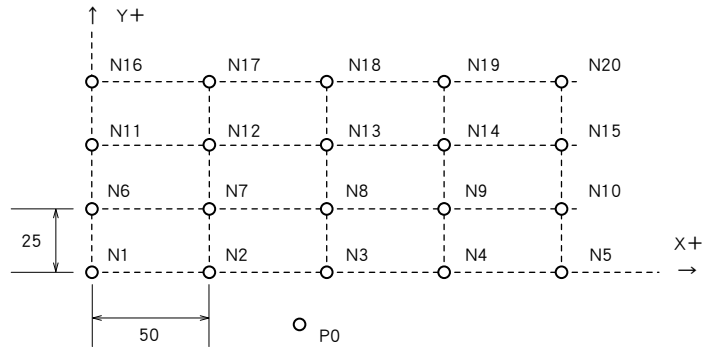
Y 方向的间距:

P20= 0.0 25.0 0.0 0.0 0.0 0.0

N1 的位置:

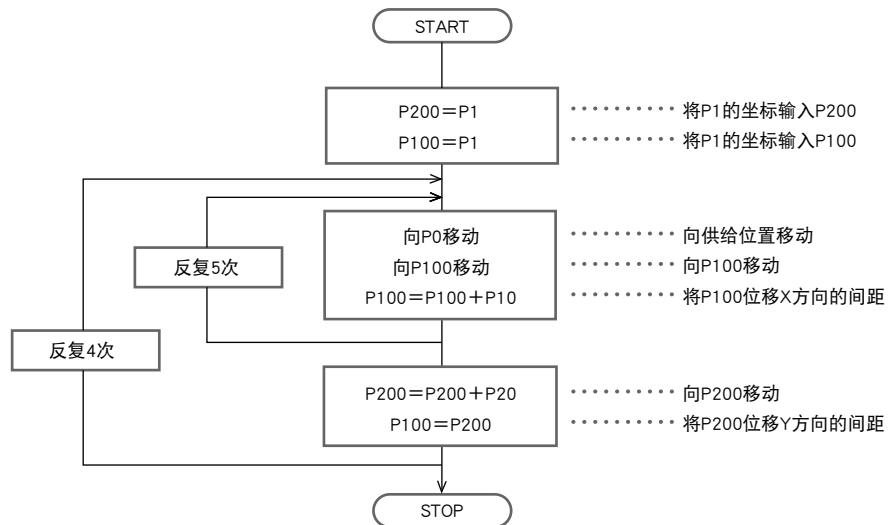
P1 = 100.0 50.0 0.0 0.0 0.0 0.0

算出坐标点坐标



35C04-R7-00

处理流程



35C05-R7-00

SAMPLE

```
P100=P1
P200=P1
FOR J=1 TO 4
  FOR K=1 TO 5
    MOVE P, P0
    MOVE P, P100
    P100=P100+P10
  NEXT K
  P200=P200+P20
  P100=P200
NEXT J
```

7

8

9

10

11

1.4.2 使用托盘移动

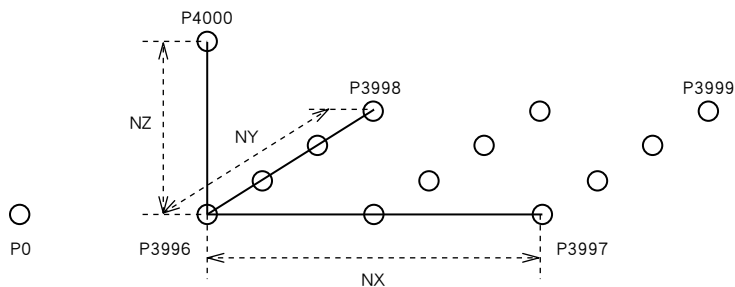
■ 概要

创建使等间距排列的托盘上的各坐标点与工件供给位置 P0 交互移动的程序。

在下图上, N1 ~ N24 在正交坐标上排列着向 X 方向按照 50mm 间距的 3 个点, 向 Y 方向按照 50mm 的间距的 4 个点, 向 Z 方向按照 100mm 间距的 2 个点。机械臂从一个坐标点移动至另一个坐标点, 按照 P0 → N1 → P0 → N2 → P0 → N3... 的顺序依次在 P0 与各点之间往返移动。

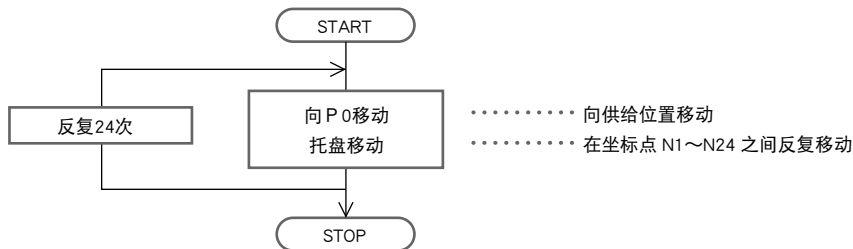
POINT DATA						
工件供给位置 :						
P0=	0.00	0.00	100.00	0.00	0.00	0.00
托盘定义 :						
PL0						
NX=	3					
NY=	4					
NZ=	2					
PLP=	3996 (使用的坐标点为 P3996 ~ P4000)					
P3996=	100.00	50.00	100.00	0.00	0.00	0.00
P3997=	200.00	50.00	100.00	0.00	0.00	0.00
P3998=	100.00	200.00	100.00	0.00	0.00	0.00
P3999=	200.00	200.00	100.00	0.00	0.00	0.00
P4000=	100.00	50.00	200.00	0.00	0.00	0.00

使用托盘移动



35C06-R7-00

处理流程



35C07-R7-00

SAMPLE

```

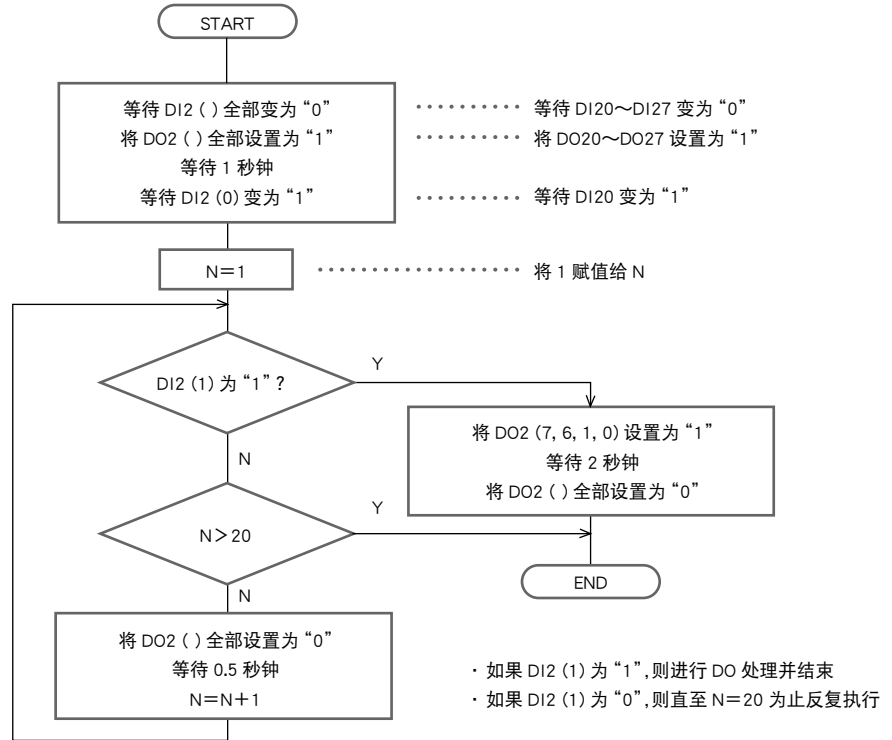
FOR I=1 TO 24 ..... 在 I = 1 ~ 24 之间反复移动
  MOVE P, P0, Z=0.00 ..... 向供给位置移动
  PMOVE (0, I), Z=0.00 ..... 向托盘上的坐标点移动
NEXT I
MOVE P, P0, Z=0.00

```

■ 概要

使用通用输入和通用输出装置进行信号的输入输出。

处理流程



35C08-R7-00

SAMPLE

```

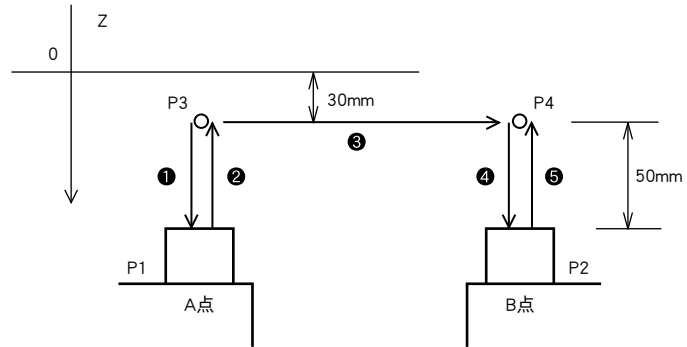
WAIT DI2 ( ) =0 ..... 等待 DI20 ~ 27 全部变为 “0”
DO2 ( ) =&B11111111..... 将 DO20 ~ DO27 全部设置为 “1”
DELAY 1000
WAIT DI2 (0) =1 ..... 等待 DI21 变为 “1”
N=1
*LOOP1 :
IF DI2 (1) =1 THEN *PROGEND .... 如果 DI21 = 1, 将跳转至 *PROGEND
IF N>20 THEN *ALLEND..... 如果 N > 20, 则结束 (跳转至 *ALLEND)
DO2 ( ) =0 ..... 将 DO20 ~ DO27 全部设置为 “0”
DELAY 500
N=N+1
GOTO *LOOP1 ..... 反复执行循环
' END ROUTINE
*PROGEND : ..... 结束时的处理
DO2 (7, 6, 1, 0) =&B1111 ..... 将 DO27, 26, 21, 20 设置为 “1”
DELAY 2000 ..... 等待 2 秒
DO2 ( ) =0 ..... 将 DO20 设置为 “0”
*ALLEND :
HALT
  
```

2.1 2点间的拾放

■ 概要

抓住 A 点的部件并将其放置到 B 点。

2点间的拾放



35C09-R7-00

■ 前提

1. 确定了机器人的移动路径。

- 移动路径：P3 → P1 → P3 → P4 → P2 → P4
- P3, P4 分别为 P1, P2 上方 50mm 的位置，并通过示教设置 P1, P2。

2. 输入输出信号

DO (20)	夹持器开关	0 : 开 / 1 : 关
---------	-------	---------------

- 夹持器开关时，等待时间为 0.1 秒。

SAMPLE : 通过计算算出 P3, P4 时

```

P3=P1 ..... 将 P1 的坐标赋值给 P3
P4=P2 ..... 将 P2 的坐标赋值给 P4
LOC3 (P3) =LOC3 (P3) -50.0 ..... 将 P3 朝着 Z 的上方向位移 50mm
LOC3 (P4) =LOC3 (P4) -50.0 ..... 将 P4 朝着 Z 的上方向位移 50mm
MOVE P, P3
GOSUB *OPEN
MOVE P, P1
GOSUB *CLOSE
MOVE P, P3
MOVE P, P4
MOVE P, P2
GOSUB *OPEN
MOVE P, P4
HALT
*OPEN : ..... 夹持器开启例程
DO2 (0) =0
DELAY 100
RETURN
*CLOSE : ..... 夹持器关闭例程
DO2 (0) =1
DELAY 100
RETURN

```

SAMPLE : 使用圆弧插补移动时

```
P4=P2 ..... 将 P2 的坐标赋值给 P4
LOC3 (P4) = LOC3 (P4)-50.0 ... 将 P4 的第 3 轴数据设为位移 50mm
GOSUB *OPEN
MOVE P, P1, A3=30.0 ..... 以 A3 = 30mm 进行圆弧插补移动
GOSUB *CLOSE
MOVE P, P2, A3=30.0 ..... 以 A3 = 30mm 进行圆弧插补移动
GOSUB *OPEN
MOVE P, P4
HALT
*OPEN : ..... 夹持器开启例程
DO2 (0) =0
DELAY 100
RETURN
*CLOSE : ..... 夹持器关闭例程
DO2 (0) =1
DELAY 100
RETURN
```

7

8

9

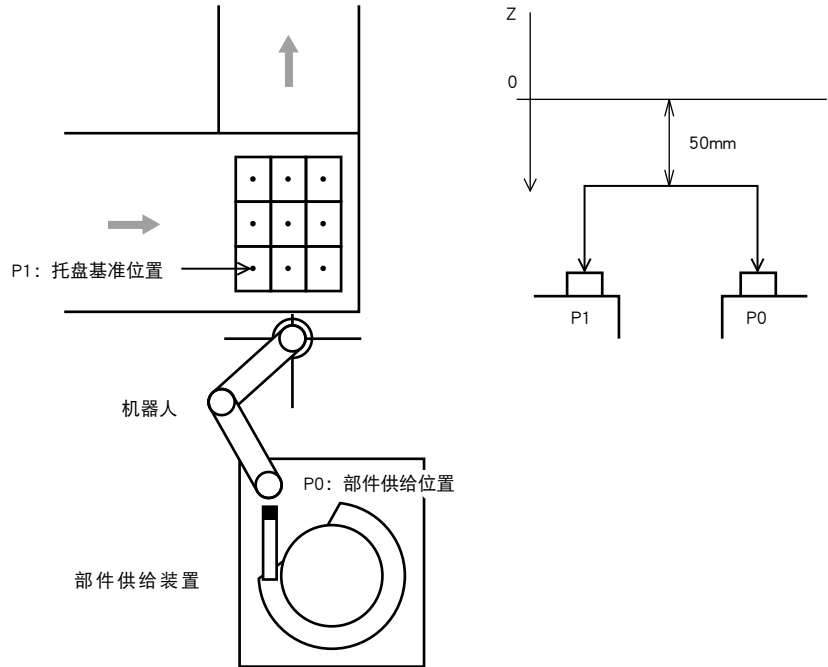
10

11

■ 概要

将部件供给装置供给的部件依次装载至传送带上的托盘中。如果托盘已满，则将放出托盘。

码垛



35C10-R7-00

■ 前提

1. 输入输出信号

DI (30)	部件检出传感器	1 : 有部件
DI (31)	托盘传感器	1 : 有托盘

DO (30)	机器人机械手开关	0 : 开 / 1 : 关
DO (31)	放出托盘	1 : 放出

机器人机械手开关时，等待时间为 0.1 秒；放出托盘时，等待时间为 0.5 秒。

2. 以下坐标点预先输入了坐标点数据。

P0	部件供给位置
P1	托盘基准位置
P10	X 方向的间距
P11	Y 方向的间距

3. 动作至托盘上方、部件供给装置上方 Z=50mm 的位置为止。

SAMPLE 1 : 通过计算算出坐标点时

```

WHILE -1 ..... 反复执行整体 (-1 时常为真)
FOR A=0 TO 2
FOR B=0 TO 2
    WAIT DI (31) =1 ..... 等待直至托盘变为“有”
    WAIT DI (30) =1 ..... 等待直至供给部件变为“有”
    DO (30) =0 ..... 将机器人机械手“开启”
    DELAY 100
    MOVE P, P0, A3=50.0 ..... 移动至供给位置
    DO (30) =1 ..... 将机器人机械手“关闭”
    DELAY 100
    P100=P1+P10*B+P11*A ..... 计算下一个坐标点
    MOVE P, P100, A3=50.0 ..... 移动至已计算的坐标点
    DO (30) =0 ..... 将机器人机械手“开启”
    DELAY 100
NEXT
NEXT
DRIVE (3, 0) ..... 仅 Z 轴移动至 0
DO (31) =1 ..... 放出托盘
DELAY 500
DO (31) =0
WEND ..... 反复执行循环
HALT

```

SAMPLE 2 : 使用码垛功能时

* 前提：必须对托盘 0 进行过定义

```

WHILE -1 ..... 反复执行整体
FOR A=1 TO 9
    WAIT DI (31) =1 ..... 等待直至托盘变为“有”
    WAIT DI (30) =1 ..... 等待直至供给部件变为“有”
    DO (30) =0 ..... 将机器人机械手“开启”
    DELAY 100
    MOVE P, P0, A3=50.0 ..... 移动至供给位置
    DO (30) =1 ..... 将机器人机械手“关闭”
    DELAY 100
    PMOVE (0, A), A3=50.0 ..... 移动至托盘上的坐标点
    DO (30) =0 ..... 将机器人机械手“开启”
    DELAY 100
NEXT
DRIVE (3, 0) ..... 仅 Z 轴移动至 0
DO (31) =1 放出托盘
DELAY 500
DO (31) =0
WEND ..... 反复执行循环
HALT

```

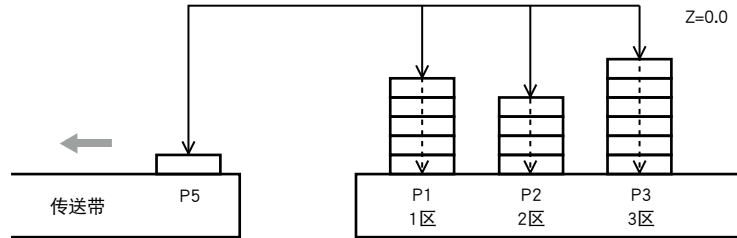
■ 概要

依次抓住最多堆积至 6 层 3 个区的部件，并将其放置到传送带上。

1 个区的部件个数可任意设置。

部件的检出通过安装在机器人机械手上的传感器执行。

层堆部件的拾放



35C11-R7-00

■ 前提

1. 输入输出信号

DI (30)	部件检出传感器	1 : 有部件
DI (31)	机器人机械手开关	0 : 开 / 1 : 关

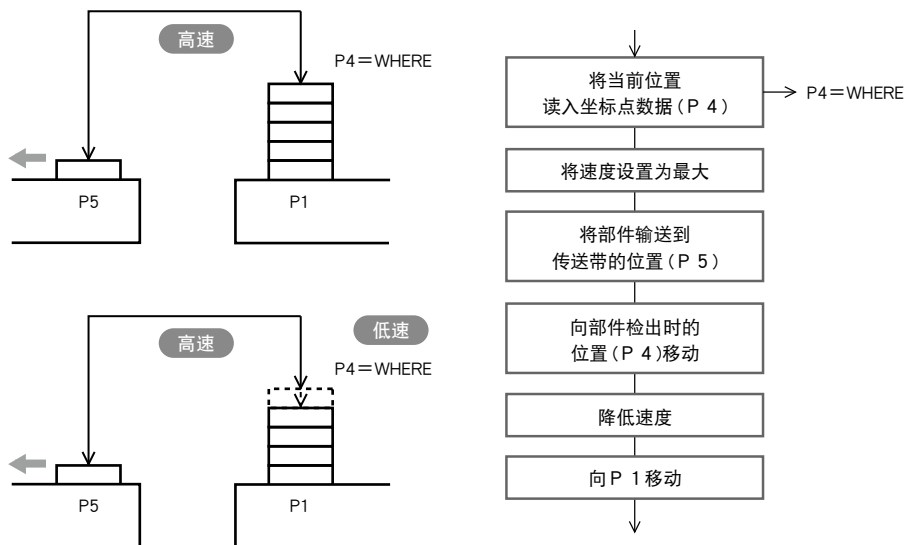
· 机器人机械手开关时，等待时间为 0.1 秒。

2. 以下坐标点预先输入了坐标点数据。

P1	1 区的下降端
P2	2 区的下降端
P3	3 区的下降端
P5	传送带上的位置

3. 移动时采用最高速度，移动至部件近前时采用较慢速度动作。

处理流程



35C12-R7-00

4. 移动中的传感器检出使用 MOVE 语句的 STOPON 条件指定。

SAMPLE

```
FOR A=1 TO 3
SPEED 100
GOSUB *OPEN
P6=P[A]
LOC3 (P6) =0.00
MOVE P, P6, A3=0.0
WHILE -1
    SPEED 20
    MOVE P, P[A], STOPON DI3 (0) =1
    IF DI3 (0) =0 THEN *L1
    ' SENSOR ON
    P4=JTOXY (WHERE)
    GOSUB *CLOSE
    SPEED 100
    MOVE P, P5, A3=0.0
    GOSUB *OPEN
    MOVE P, P4, A3=0.0
WEND
*L1 : ' SENSOR OFF
NEXT A
SPEED 100
DRIVE (3, 0)
HALT
*OPEN :
DO3 (0) =0
DELAY 100
RETURN
*CLOSE :
DO3 (0) =1
DELAY 100
RETURN
```

7

8

9

10

11

■ 概要

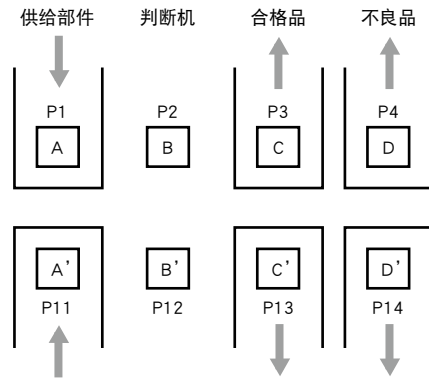
使用 1 台机器人用判断机判断 2 个不同的部件，并区分出合格品与不良品。

抓住 A 点的部件并移动至 B 点的判断机。判断机进行合格品 / 不良品的判断，如果是合格品，向 C 点移动，如果是不良品，则向 D 点移动。

同样地，抓住 A' 点的部件并移动至 B' 点的判断机。判断机进行合格品 / 不良品的判断，如果是合格品，向 C' 点移动，如果是不良品，则向 D' 点移动。

判断机的判断需要 10 秒钟的时间。

部件检查 1 (多任务的示例)

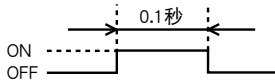


35C13-R7-00



要点

- ※1：启动信号向判断机提供 0.1 秒的脉冲信号。

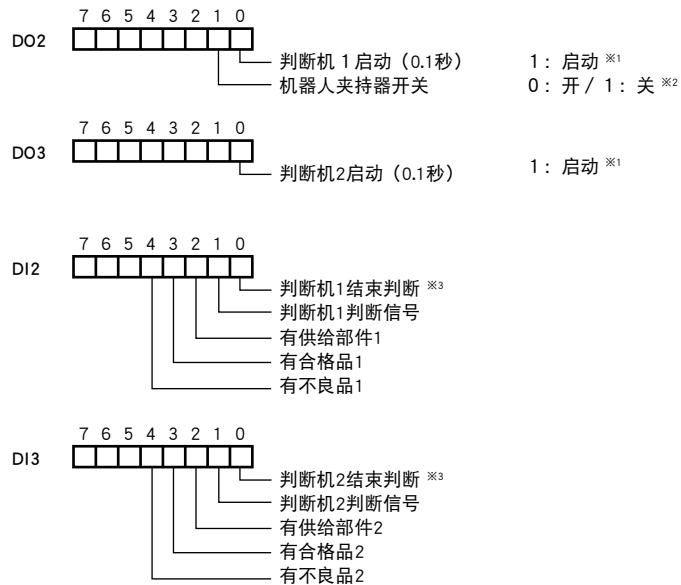


- ※2：夹持器开关时，等待时间为 0.1 秒。
- ※3：从判断机获得的信号中，有结束及合格品 / 不良品的判断信号。判断结束后，结束信号为 ON (= 1)，当判断信号为合格品时将变为 ON (= 1)，当判断信号为不良品将变为 OFF (= 0)。

■ 前提

1. 输入输出信号

输入输出信号

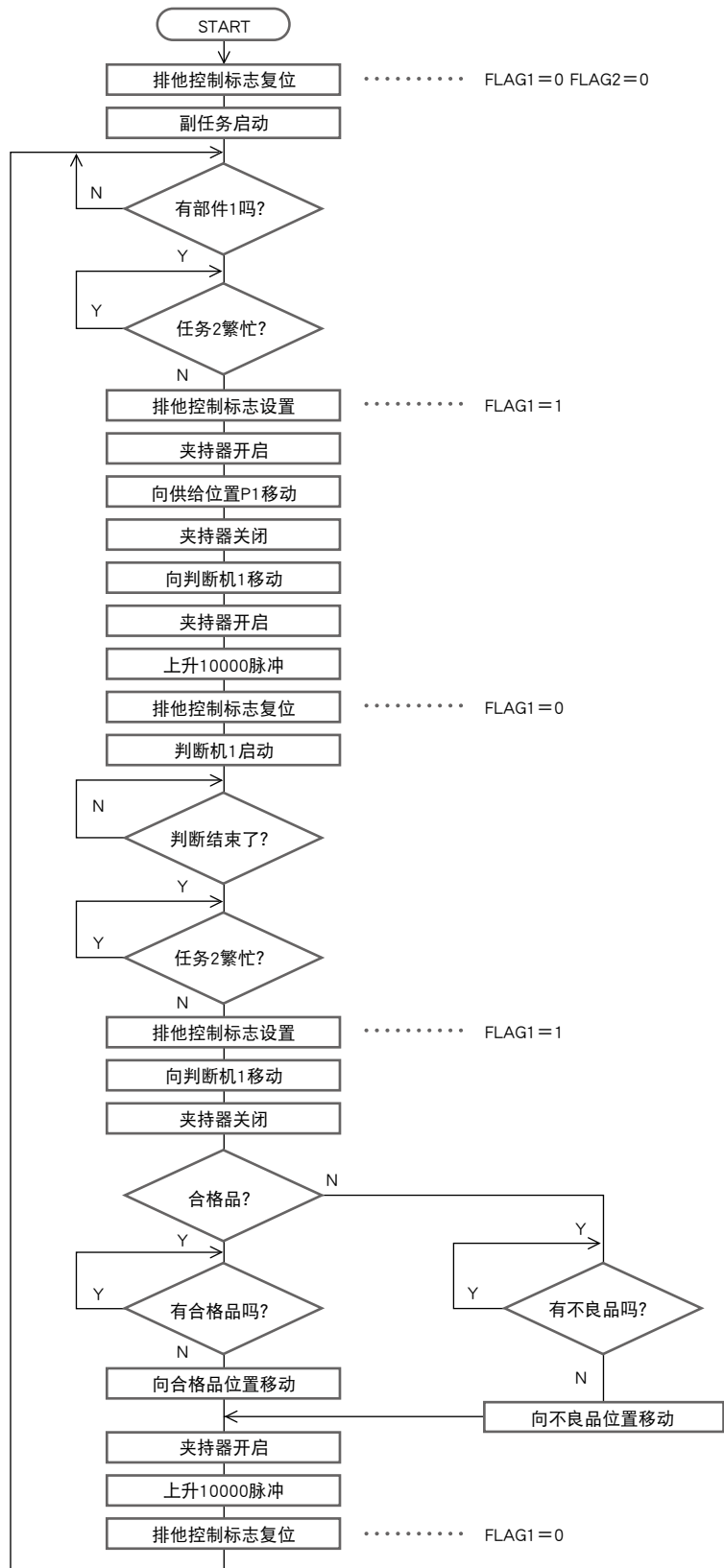


35C14-R7-00

- 启动副任务，由主任务 (任务 1) 控制部件 1，由副任务 (任务 2) 控制部件 2。
- 若要仅在等待判断机的判断结束信号时使其他任务运行，请使用排他控制标志。

FLAG1	0: 任务 1 执行中	(可执行任务 2)
	1: 任务 1 待机中	(不可执行任务 2)
FLAG2	0: 任务 2 执行中	(可执行任务 1)
	1: 任务 2 待机中	(不可执行任务 1)

4. 流程表
处理流程



35C15-R7-00

任务 2 (副任务) 也采用相同的流程表。

■ 程序示例

SAMPLE

<主任务>

```

FLAG1=0
FLAG2=0
UPPOS=0.0
START <SUB_PGM>, T2 副任务启动
*L1 :
WAIT DI2 (2) =1          等待部件供给
WAIT FLAG2=0            等待其他任务的待机状态
FLAG1=1                 排他控制标志设置
GOSUB *OPEN             夹持器开启
MOVE P, P1, Z=UPPOS     向供给位置移动
GOSUB *CLOSE           夹持器关闭
MOVE P, P2, Z=UPPOS     向判断机移动
GOSUB *OPEN             夹持器开启
DRIVEI (3, -10000)     Z 上升 10000 脉冲
FLAG1=0                 排他控制标志复位
DO2 (0) =1              判断机启动
DELAY 100
DO2 (0) =0
WAIT DI2 (0) =1          等待判断结束
WAIT FLAG2=0            等待任务动作结束
FLAG1=1                 排他控制标志设置
MOVE P, P2, Z=UPPOS     向判断机移动
GOSUB *CLOSE           夹持器关闭
IF DI2 (1) =1 THEN      判断
'GOOD
WAIT DI4 (2) =0          等待部件移动
MOVE P, P3, Z=UPPOS     向合格品位置移动
ELSE
'NG
WAIT DI2 (4) =0          等待部件移动
MOVE P, P4, Z=UPPOS     向不良品位置移动
ENDIF
GOSUB *OPEN             夹持器开启
DRIVEI (3, -10000)     Z 上升 10000 脉冲
FLAG1=0                 排他控制标志复位
GOTO *L1
*OPEN :
DO2 (1) =0
DELAY 100
RETURN
*CLOSE :
DO2 (1) =1
DELAY 100
RETURN
    
```

<副任务>

```

*S1 :
WAIT DI3 (2) =1
WAIT FLAG1=0
FLAG2=1
GOSUB *OPEN
MOVE P, P11, Z=UPPOS
GOSUB *CLOSE
MOVE P, P12, Z=UPPOS
GOSUB *OPEN
DRIVEI (3, -10000)
FLAG2=0
DO3 (0) =1
DELAY 100
DO3 (0) =0
WAIT DI3 (0) =1
WAIT FLAG1=0
FLAG2=1
MOVE P, P12, Z=UPPOS
GOSUB *CLOSE
IF DI3 (1) =1 THEN
'GOOD
WAIT DI3 (3) =0
MOVE P, P13, Z=UPPOS
ELSE
'NG
WAIT DI3 (4) =0
MOVE P, P14, Z=UPPOS
ENDIF
GOSUB *OPEN
DRIVEI (3, -10000)
FLAG2=0
GOTO *S1
    
```

■ 概要

使用外部装置与 RS-232C 连接 RCX340 控制器，并写入坐标点数据。

■ 前提

1. 从控制器输入至外部装置

SDATA/X/Y [cr/lf]

2. 从外部装置输出至控制器

POINT DATA

P10=	156.42	243.91	0.00	0.00	0.00	0.00	[cr/lf]
------	--------	--------	------	------	------	------	---------

SAMPLE

```
' INIT
  VCMD$="SDATA/X/Y"
  P0= 0.00 0.00 0.00 0.00 0.00 0.00
' MAIN ROUTINE
  MOVE P, P0
*ST :
  SEND VCMD$ TO CMU
  SEND CMU TO P10
  MOVE P, P10
GOTO *ST
```



要点

- [cr/lf] 表示 CR 代码 (=0Dh)+LF 代码 (=0Ah)。

MEMO

- "SEND xxx TO CMU" 使用 RS-232C 输出 xxx 中指定的内容。
- "SEND CMU TO xxx" 使用 RS-232C 输入 xxx 中指定的文件。

■ 概要

使用外部装置与 RS-232C 连接 RCX340 控制器，并从任意字符串数据创建坐标点数据并写入。

■ 前提

1. 从控制器输入至外部装置

SDATA/X/Y [cr/lf]

2. 从外部装置输出至控制器

X=156.42, Y=243.91 [cr/lf]



要点

- [cr/lf] 表示 CR 代码 (=0Dh)+LF 代码 (=0Ah)。



MEMO

- "SEND xxx TO CMU" 使用 RS-232C 输出 xxx 中指定的内容。
- "SEND CMU TO xxx" 使用 RS-232C 输入 xxx 中指定的文件。
- LEN () 计算字符串的长度。
- MID\$ () 计算字符串中指定的字符串。
- VAL () 从字符串计算数值。

SAMPLE

```

' INT
  VCMD$="SDATA/X/Y"
  VINS$=""
  VX$=""
  VY$=""
  P0=      0.00      0.00      0.00      0.00      0.00      0.00
  P11=    100.00    100.00      0.00      0.00      0.00      0.00
' MAIN ROUTINE
  MOVE P, P0
*ST :
  SEND VCMD$ TO CMU
  SEND CMU TO VINS
  I=1
  VMAX=LEN (VINS)
*LOOP :
  IF I>VMAX THEN GOTO *E_LOOP
  C$=MID$ (VINS, I, 1)
  IF C$="X" THEN
    I=I+2
    J=I
*X_LOOP :
  C$=MID$ (VINS, J, 1)
  IF C$="," THEN
*X1_LP :
  L=J-I
  VX$=MID$ (VINS, I, L)
  I=J+1
  GOTO *LOOP
  ENDIF
  J=J+1
  IF J>VMAX THEN GOTO *X1_LP
  GOTO *X_LOOP
  ENDIF
  IF C$="Y" THEN
    I=I+2
    J=I
*Y_LOOP :
  C$=MID$ (VINS, J, 1)
  IF C$="," THEN
*Y1_LP :
  L=J-I
  VY$=MID$ (VINS, I, L)
  I=J+1
  GOTO *LOOP
  ENDIF
  J=J+1
  IF J>VMAX THEN GOTO *Y1_LP
  GOTO *Y_LOOP
  END IF
  I=I+1
  GOTO *LOOP
*E_LOOP :
  WX=VAL (VX$)
  WY=VAL (VY$)
  LOC1 (P11) =WX
  LOC2 (P11) =WY
  MOVE P, P11
  GOTO *ST

```

7

8

9

10

11

第 10 章

联机命令

1	联机命令一览表	10-1
2	操作、设定命令	10-6
3	参照命令	10-14
4	动作命令	10-24
5	数据文件操作命令	10-28
6	实用命令	10-39
7	机器人语言的单独执行	10-41
8	控制代码	10-41

使用联机命令后，可通过 RS-232C 或 Ethernet 对控制器执行命令。

在本章节中，将介绍在联机命令中可使用的命令。关于 RS-232C 以及 Ethernet 的连接方法，请参照“RCX340 控制器手册”。

关于终端编码

控制器发送数据时，在发送数据的行尾附加下一个代码。

- RS-232C
 - 通信参数的终端编码参数设置为 "CRLF" 时，在行的最后附加 CR (0Dh) 与 LF (0Ah)。
 - 通信参数的终端编码参数设置为 "CR" 时，在行的最后附加 CR (0Dh)。
- Ethernet
 - 在行的最后附加 CR (0Dh) 与 LF (0Ah)。

接收数据时，无论终端编码参数的设置如何，至 CR (0Dh) 为止将作为 1 行，LF (0Ah) 将被忽略。在本章“2 按钮操作”以后的各联机命令的详细说明中，终端编码表示为 [cr/lf]。

1.1 联机命令一览表：按功能

键盘操作

	含义	命令	选配件
模式更改	自动模式 手动模式	AUTO MANUAL	
程序	复位 执行 执行 1 行 跳过 1 行 执行直至下一行 停止	RESET RUN STEP SKIP NEXT STOP	
设置断点		BREAK	m, n (m: 断点编号, n: 行)
更改手动速度		MSPEED	k (k: 1-100)
可进行绝对式原点复位的位置移动		ABSADJ	k, 0 或 k, 1 (k: 1-6)
返回原点		ORGRTN	k (k: 1-6)
手动移动 (点动)		INCH	k+ 或 k- (k: X, Y, Z, R, A, B)
手动移动 (微动)		JOG	k+ 或 k- (k: X, Y, Z, R, A, B)
坐标点数据示教		TEACH	m (m: 坐标点编号)

UTILITY

含义		命令	选配件
将程序从 1 复制至 2 将坐标点 m 至 n 复制至 k 将坐标点注释 m 至 n 复制至 k		COPY	<程序 1 > TO <程序 2 > Pm-Pn TO Pk PCm-PCn TO Pck
删除程序 删除坐标点 m 至 n 删除坐标点注释 m 至 n 删除托盘 m		ERA	<程序 > Pm-Pn PCm-PCn PLm
将程序名从 1 更改为 2		REN	<程序 1 > TO <程序 2 >
更改程序属性		ATTR	<程序 > TO s (s : RW/RO)
数据初始化	程序 坐标点 位移 机械手 托盘 坐标点注释 参数以外的全部 参数 所有数据 (MEM+PRM)	INIT	PGM PNT SFT HND PLT PCM MEM PRM ALL
数据初始化	通信参数	INIT	CMU
数据初始化	错误履历	INIT	LOG
设置	访问级 次序执行标志	ACCESS SEQUENCE	k k
日期的确认、设置		DATE	
时间的确认、设置		TIME	

关于条件

1. 时常可执行
2. 手持编程器输入操作中，无法执行
3. 除了条件 2. 以外，程序执行过程中也不可执行
4. 除了条件 3. 以外，还有固有的限制

数据处理

含义	命令	选配件	条件
获取 访问级 机械臂状态 断点状态 模式显示 返回原点状态 伺服状态 次序执行标志状态 版本 当前位置 (脉冲) 当前位置 (XY 坐标) 任务编号 任务运行状态 选择位移状态 选择机械手状态 存储器剩余容量 紧急停止状态 数值数据 字符串数据 坐标点数据 位移数据	?	ACCESS ARM BREAK MOD ORIGIN SERVO SEQUENCE VER WHERE WHRXY TASKS TSKMON SHIFT HAND MEM EMG 数值表达式 字符串表达式 坐标点表达式 位移表达式	1
数据读出	READ		2
数据写出	WRITE		2

单独执行机器人语言

含义	命令	选配件	条件
可单独执行的机器人语言			4

控制代码

含义	命令	选配件	条件
执行语言的中断	^C (=03H)		1

关于条件

1. 时常可执行
2. 手持编程器输入操作中, 无法执行
3. 除了条件 2. 以外, 程序执行过程中也不可执行
4. 除了条件 3. 以外, 还有固有限制

命令	选配件	含义	条件
?	ACCESS	获取访问级	1
	ARM	获取机械臂状态	1
	BREAK	获取断点状态	1
	CONFIG	获取控制器构成	1
	EMG	获取紧急停止状态	1
	EXELVL	获取执行级别	1
	HAND	获取选择机械手状态	1
	MEM	获取存储器剩余容量	1
	MOD	获取模式显示	1
	MSG [m, n]	获取错误提示	1
	OPSLOT	获取选配插槽状态	1
	ORIGIN	获取返回原点状态	1
	SELFCHK	经自我诊断获取出错状态	1
	SEQUENCE	获取次序执行标志状态	1
	SERVO	获取伺服状态	1
	SHIFT	获取选择位移状态	1
	MSPEED	获取手动速度状态	1
	TASKS	获取任务编号	1
	TSKMON	获取任务运行状态	1
	VER	获取版本	1
	WHERE	获取当前位置 (脉冲)	1
	WHRXY	获取当前位置 (XY 坐标)	1
	WHRXYEX	获取当前位置 (XY 坐标)	1
	位移表达式	获取位移数据	1
坐标点表达式	获取坐标点数据	1	
数值表达式	获取数值数据	1	
字符串表达式	获取字符串数据	1	
^C (=03H)		执行语言的中断	1
ABSADJ	k, 0 或 k, 1 (k: 1-6)	可进行绝对式原点复位的位置移动	3
ACCESS	k	设置访问级	3
ATTR	<程序> TO s (s: RW/RO)	更改程序属性	3
AUTO		模式更改：自动模式	3
BREAK	m, n (m: 断点编号, n: 行)	设置断点	4
COPY	<程序 1> TO <程序 2>	将程序从 1 复制至 2	3
	PCm-PCn TO PCK	将坐标点注释 m 至 n 复制至 k	3
	Pm-Pn TO Pk	将坐标点 m 至 n 复制至 k	3
DATE		日期的确认、设置	2
EMGRST		解除内部紧急停止标志	1
ERA	<程序>	删除程序	3
	PCm-PCn	删除坐标点注释 m 至 n	3
	PLm	删除托盘 m	3
	Pm-Pn	删除坐标点 m 至 n	3
EXELVL	k	执行级别	3
INCH	k+ 或 k- (k: 1 ~ 6)	手动移动 (点动)	3
INIT	ALL	所有数据初始化 (MEM+PRM)	3
	CMU	通信参数初始化	3
	HND	机械手数据初始化	3
	LOG	错误履历初始化	3
	MEM	参数以外的所有数据初始化	3
	PCM	坐标点注释数据初始化	3
	PGM	程序数据初始化	3
	PLT	托盘数据初始化	3
	PNT	坐标点数据初始化	3

命令	选配件	含义	条件
	PRM	参数初始化	3
	SFT	位移数据初始化	3
JOG	k+ 或 k- (k : 1 ~ 6)	手动移动 (微动)	3
MANUAL		模式更改 : 手动模式	3
MRKSET	k (k : 1-6)	绝对式原点复位轴单位	3
MSGCLR		设置 行信息的取消	1
MSPEED	k (k : 1-100)	更改手动速度	3
NEXT		将程序执行至下一行为止	4
ORGRTN	k (k : 1-6)	返回原点	3
READ		数据读出	2
REN	<程序 1 > TO <程序 2 >	将程序名从 1 更改为 2	3
RESET		程序复位	4
RUN		执行程序	4
SEQUENCE	k	设置次序执行标志	3
SKIP		程序 : 跳过 1 行	4
STEP		程序 : 执行 1 行	4
STOP		程序停止	2
TEACH	m (m : 坐标点编号)	坐标点数据示教	3
TIME		时间的确认、设置	2
WRITE		数据输出	2
-		可单独执行的机器人语言	4

关于条件

1. 时常可执行
2. 手持编程器输入操作中, 无法执行
3. 除了条件 2. 以外, 程序执行过程中也不可执行
4. 除了条件 3. 以外, 还有固有的限制

2 操作、设定命令

2.1 程序操作

1. 任务登录

命令格式

```
@ LOAD “<” <程序名> “>” [,Tn [, p]] [cr/lf]
PGm
```

应答格式

```
OK[cr/lf]
```

值

m..... 程序编号：0～99
n..... 任务编号：1～16
p..... 任务优先级：1～64

含义 按照优先顺序 p 将指定的程序登录到任务 n 中。已登录程序变为 STOP 状态。当省略了任务编号 n 时,未启动任务中最小编号的任務將自动被指定。省略了任务优先顺序 p 时,默认为 32。

值越小优先顺序越高, 值越大优先顺序越低(高 1～低: 64)。

任务优先顺序高的任务变为 RUNNING 状态时, 优先顺序低的任务保持 READY 状态。

SAMPLE

```
命令： @ LOAD <PG_MAIN>, T1 [cr/lf] .... 将程序登录到任务 1
应答： OK [cr/lf]
```

2. 程序复位

命令格式

```
1. @ RESET [cr/lf]
2. @ RESET Tn [cr/lf]
   “<” <程序名> “>”
   PGm
```

应答格式

```
OK[cr/lf]
```

值

n..... 任务编号：1～16
m..... 程序编号：0～99

含义 执行程序复位。

格式 1 对所有的程序进行复位。当重新启动了程序时，从主程序或者任务 1 中最后执行的程序开头开始执行。

格式 2 仅对指定的程序进行复位。当重新启动复位后的程序时，从该程序的开头开始执行。

SAMPLE

命令： @ RESET [cr/lf]..... 对所有程序进行复位
 应答： OK [cr/lf]
 命令： @ RESET T3 [cr/lf]..... 仅对 T3 内正在执行的程序进行复位
 应答： OK [cr/lf]

3. 程序执行

命令格式

1. @ RUN [cr/lf]
 2. @ RUN | Tn | [cr/lf]
 | “<” <程序名> “>” |
 | PGm |

应答格式

OK[cr/lf]

值 n..... 任务编号：1 ~ 16
 m..... 程序编号：0 ~ 99

含义 执行程序复位。

格式 1 执行所有的 STOP 状态程序。

格式 2 仅执行指定的 STOP 状态程序。

SAMPLE

命令： @ RUN [cr/lf]..... 执行所有的 STOP 状态程序
 应答： OK [cr/lf]
 命令： @ RUN T3 [cr/lf]..... 仅执行登录到 T3 中的 STOP 状态程序
 应答： OK [cr/lf]



要点

· 仅能在自动模式中执行。

4. 程序停止

命令格式

```

1. @ STOP [cr/lf]
2. @ STOP | Tn | [cr/lf]
   | " < " < 程序名 > " > " |
   | PGm |

```

应答格式

OK[cr/lf]

值 n.....任务编号：1 ~ 16
m.....程序编号：0 ~ 99

含义 停止程序。
格式 1 停止所有的程序。
格式 2 仅停止指定的程序。

SAMPLE

命令： @ STOP [cr/lf]..... 停止所有的程序
应答： OK [cr/lf]
命令： @ STOP T3 [cr/lf] 仅停止 T3 内正在执行的程序
应答： OK [cr/lf]

5. 执行 1 行程序

命令格式

```

@ STEP | Tn | [cr/lf]
   | " < " < 程序名 > " > " |
   | PGm |

```

应答格式

OK[cr/lf]

值 n.....任务编号：1 ~ 16
m.....程序编号：0 ~ 99

含义 执行 1 行指定的程序。当执行了 1 行 GOSUB 语句、CALL 语句时，进入子例程、子过程的内部。

SAMPLE

命令： @ STEP T3 [cr/lf] 执行 1 行 T3 内正在执行的程序
应答： OK [cr/lf]

6. 跳过 1 行程序

命令格式

@ SKIP	Tn “<” <程序名> “>” PGm	[cr/lf]
--------	----------------------------	---------

应答格式

OK[cr/lf]

值 n.....任务编号：1～16
m.....程序编号：0～99

含义 跳过 1 行指定的程序。当跳过了 1 行 GOSUB 语句、CALL 语句时，将跳过全部的子例程、子过程。

SAMPLE

命令： @ SKIP T3 [cr/lf]..... 跳过 1 行 T3 内正在执行的程序

应答： OK [cr/lf]

7. 执行程序至下一行

命令格式

@ NEXT	Tn “<” <程序名> “>” PGm	[cr/lf]
--------	----------------------------	---------

应答格式

OK[cr/lf]

值 n.....任务编号：1～16
m.....程序编号：0～99

含义 执行指定程序至下一行。当执行了 GOSUB 语句、CALL 语句时，将子例程、子过程作为 1 行来执行。

SAMPLE

命令： @ NEXT T3 [cr/lf] 跳过 1 行 T3 内正在执行的程序

应答： OK [cr/lf]

8. 执行程序至指定行之前

命令格式

```
@ RUNTO | Tn | , k [cr/lf]
          | "<" <程序名> ">" |
          | PGm |
```

应答格式

```
OK[cr/lf]
```

值

n.....任务编号：1～16
 m.....程序编号：0～99
 K.....指定行编号：1～9999

含义 执行指定程序至指定行之前的行。

SAMPLE

```
命令： @ RUNTO T3, 15 [cr/lf] .... 执行 T3 内正在执行的程序至第 15 行
应答： OK [cr/lf]
```

9. 跳过程序至指定行之前

命令格式

```
@ SKIPTO | Tn | , k [cr/lf]
          | "<" <程序名> ">" |
          | PGm |
```

应答格式

```
OK[cr/lf]
```

值

n.....任务编号：1～16
 m.....程序编号：0～99
 K.....设置断点的行号：1～9999

含义 跳过指定程序至指定行之前的行。

SAMPLE

```
命令： @ SKIPTO T3, 15 [cr/lf] .... 跳过 T3 内正在执行的程序至第 14 行
应答： OK [cr/lf]
```

10. 断点设定

命令形式

- | | | |
|------------|---------------|--------------------------|
| 1. @ BREAK | “<” <程序名> “>” | (n[,n,n,...]), k [cr/lf] |
| | PGm | |
| 2. @ BREAK | “<” <程序名> “>” | 0 [cr/lf] |
| | PGm | |
| 3. @ BREAK | 0 | [cr/lf] |

应答格式

OK[cr/lf]

值

m..... 程序编号：0～99
n..... 指定行编号：1～9999
K..... 设定/解除：0：设定、1：解除

含义

对程序执行过程中因暂停出现的断点进行设定。
格式 1 在指定程序的指定行对断点进行设定/解除。也可以指定多行。
格式 2 对在指定程序中设定的所有断点进行解除。
格式 3 解除所有的断点。

SAMPLE

命令： @ BREAK PG3 (1, 3), 1 [cr/lf] 在 PG3 的第 1 行和第 3 行设定断点
应答： OK [cr/lf]

2.3 清除输出信息缓冲区

命令格式

```
@ MSGCLR [cr/lf]
```

应答格式

```
OK[cr/lf]
```

含义 清除控制器的输出信息缓冲区。删除 PRINT 语句等输出的、残留在缓冲区中的信息。

SAMPLE

```
命令： @ MSGCLR [cr/lf]
```

```
应答： OK[cr/lf]
```

2.4 设定输入数据

命令格式

```
@ INPUT | SET d | ( [cr/lf]  
          | CAN  
          | CLR
```

应答格式

```
OK[cr/lf]
```

值 d: 输入数据.....符合 INPUT 语句中指定变量类型的值(字符串加“ ”)

含义 针对 INPUT 语句的数据请求设定输入数据。

SET 设定数据。
CAN 取消数据请求。
CLR 清除控制器未接收的数据。

SAMPLE

```
命令： @ INPUT "DATA" [cr/lf]
```

```
应答： OK [cr/lf]
```


命令格式

```
@ ?SERVO [< 机器人编号 >] [cr/lf]
```

应答格式

```
x y {,y {, {...} }} [cr/lf]
OK [cr/lf]
```

- 值**
- < 机器人编号 > 1 ~ 4
 - x : 机器人伺服状态 0 : 伺服关闭状态、1 : 伺服开启状态
 - y : 轴伺服状态 从左向右显示第 1 轴、第 2 轴、...、第 6 轴的状态。
 - 0 : 机械制动器开启 + 动态制动器开启状态
 - 1 : 伺服开启状态
 - 2 : 机械制动器关闭 + 动态制动器关闭状态
(轴未连接时省略)

含义 获取伺服状态。< 机器人编号 > 可以省略。当进行省略时，机器人 1 被指定。

SAMPLE

```
命令 : @ ?SERVO[3] [cr/lf]
应答 : 0 0,1,0,0 ..... 仅机器人 3 的第 2 轴处于伺服开启状态
      OK [cr/lf]
```

命令格式

```
@?MOTOR [cr/lf]
```

应答格式

```
x [cr/lf]
OK [cr/lf]
```

- 值**
- x : 马达电源状态 0 : 马达电源关闭状态、
 - 1 : 马达电源开启状态
 - 2 : 马达电源开启 + 所有机器人伺服开启状态

含义 获取马达电源状态。

SAMPLE

```
命令 : @ ?MOTOR [cr/lf]
应答 : 2
      OK [cr/lf]
```

3.4

获取访问级状态

命令格式

```
@ ?ACCESS [cr/lf]
```

应答格式

```
k [cr/lf]
```

值 k.....访问级：0～1

含义 进行访问级状态的获取处理。

SAMPLE

```
命令： @ ?ACCESS [cr/lf]
```

```
应答： 1 [cr/lf]
```

```
OK [cr/lf]
```



参考

· 关于访问级的详细说明，请参阅控制器的手册。

3.5

获取断点状态

命令格式

```
@ ?BREAK [cr/lf]
```

应答格式

```
k1, k2, k3, k4 [cr/lf]
```

值 kn.....断点 n 的设置行号：1～9999

含义 进行断点状态的获取处理。

kn 的值为 0 时，表示无设置。

对 COMMON 程序的设置显示为行号 +10000。

SAMPLE

```
命令： @ ?BREAK [cr/lf]
```

```
应答： 12, 35, 0, 0 [cr/lf]
```

命令格式

```
@ ?MODE [cr/lf]
```

应答格式

```
k [cr/lf]
OK [cr/lf]
```

值 k..... 模式状态

含义 获取控制器的模式状态。

SAMPLE

```
命令： @ ?MODE [cr/lf]
应答： 1 [cr/lf]
      OK [cr/lf]
```

命令格式

```
@ ?SEQUENCE [cr/lf]
```

应答格式

```
1. 1, s [cr/lf]
2. 3, s [cr/lf]
3. 0 [cr/lf]
```

值 s..... 次序程序的执行状态通过 1 或 0 表示。
 1..... 执行中
 0..... 停止中

含义 获取次序程序的执行状态。

应答输出表示含义如下。

1..... 许可状态
 3..... 在许可状态且紧急停止时清除输出
 0..... 禁止状态

SAMPLE

```
命令： @ ? SEQUENCE [cr/lf]
应答： 0 [cr/lf]
```

命令格式

```
@ ?VER [cr/lf]
```

应答格式

```
cv,cr-mv-dv1,dr1/dv2,dr2 [cr/lf]
```

值

cv 主机的版本
 cr 主机的修订版本 (Rxxxx)
 mv PLO 的版本 (Vx.xx)
 dv?(? : 1、2) 驱动器的版本 (Vx.xx)
 dr?(? : 1、2) 驱动器的修订版 (Rxxxx)

含义 获取版本。

SAMPLE

```
命令： @ ?VER [cr/lf]
```

```
应答： V8.02,R1021-V5.10-V1.01,R0001/V1.01,R0001 [cr/lf]
```

1. 获取脉冲坐标系的当前位置

命令格式

```
@ ?WHERE [cr/lf]
```

应答格式

```
xxxxxx yyyyyy zzzzzz rrrrrr aaaaaa bbbbbb[cr/lf]
```

值

xxxxxx 第 1 轴在脉冲单位制中的当前位置
 yyyyyy 第 2 轴在脉冲单位制中的当前位置
 :
 bbbbbb 第 6 轴在脉冲单位制中的当前位置

含义 获取当前位置。
 WHERE : 获取指定机器人的当前位置。

SAMPLE

```
命令： @ ?WHERE [cr/lf]
```

```
应答： 1000 2000 3000 -40000 0 0 [cr/lf]
```


命令格式

```
@ ?HAND[< 机器人编号 >][cr/lf]
```

应答格式

```
m [cr/lf]
```

值 < 机器人编号 > 1 ~ 4
 m..... 指定机器人的选择机械手编号状态 : 0 ~ 31

含义 获取< 机器人编号 > 指定机器人的机械手状态。< 机器人编号 > 可以省略。当进行省略时, 机器人 1 被指定。

SAMPLE

```
命令 : @ ?HAND [cr/lf]
应答 : 1 [cr/lf]
      OK[cr/lf]
```

命令格式

```
@ ?MEM [cr/lf]
```

应答格式

```
k/m [cr/lf]
```

值 k..... 源区域剩余容量(单位: 字节)
 m..... 全局识别符区域剩余容量(单位: 字节)

含义 获取当前存储器的剩余容量。

SAMPLE

```
命令 : @ ?MEM [cr/lf]
应答 : 102543/1342 [cr/lf]
      OK[cr/lf]
```

命令格式

```
@ ?EMG [cr/lf]
```

应答格式

```
k [cr/lf]
```

值 k.....紧急停止状态 / 0 : 正常状态、1 : 紧急停止状态

含义 获取紧急停止状态。确认内部紧急停止标志。

SAMPLE

```
命令 : @ ?EMG [cr/lf]
应答 : 1 [cr/lf]
      OK[cr/lf]
```

1. 获取数值表达式的值

命令格式

```
@ ? “数值表达式” [cr/lf]
```

应答格式

```
“数值” [cr/lf]
```

含义 获取指定数值表达式的值。
数值表达式的值，采用十进制或实数形式。

SAMPLE 1

```
命令 : @ ?SQR (100*5) [cr/lf]
应答 : 2.23606E01 [c/lf]
      OK[cr/lf]
```

SAMPLE 2

```
命令 : @ ?LOC1 (WHERE) [cr/lf]
应答 : 102054 [cr/lf]
      OK[cr/lf]
```


2. 获取字符串表达式的值

命令格式

@ ? “字符串表达式” [cr/lf]

应答格式

“字符串” [cr/lf]

含义 获取指定字符串表达式的值(字符串)。

SAMPLE

A\$="ABC"、B\$="DEF" 时
命令： @ ?A\$+B\$+"123" [cr/lf]
应答： ABCDEF123 [cr/lf]
OK[cr/lf]

3. 获取坐标点表达式的值

命令格式

@ ? “坐标点表达式” [cr/lf]

应答格式

“坐标点数据” [cr/lf]

含义 获取指定坐标点表达式的值(坐标点数据)。

SAMPLE

命令： @ ?P1+WHRXY [cr/lf]
应答： 10.41 -1.60 52.15 3.00 0.00 0.00 0 [cr/lf]
OK[cr/lf]

4. 获取位移表达式的值

命令格式

@ ? “位移表达式” [cr/lf]

应答格式

“位移数据” [cr/lf]

含义 获取指定位移表达式的值(位移数据)。

SAMPLE

命令： @ ?s1 [cr/lf]
应答： 25.00 12.60 10.00 0.00 [cr/lf]
OK[cr/lf]

4.1 绝对复位

命令格式

```
@ ABSADJ [< 机器人编号 >] k, f [cr/lf]
@ MRKSET [< 机器人编号 >] k [cr/lf]
```

应答格式

```
开始移动时：RUN[cr/lf]
结束移动时：END[cr/lf]
```

值 < 机器人编号 > 1 ~ 4
 k 指定轴：1 ~ 6
 f 移动方向 / 0：+方向、1：-方向

含义 对利用< 机器人编号 >指定的机器人进行绝对复位操作。
 < 机器人编号 >可以省略。当进行省略时，机器人 1 被指定。
 ABSADJ 指定机器人轴向可进行绝对值复位的位置进行移动
 ABSRESET 指定机器人轴的绝对值复位

SAMPLE

```
命令： @ ABSADJ 1, 0 [cr/lf]
应答： RUN[cr/lf] ..... 开始移动
       END[cr/lf] ..... 结束移动
```

MEMO

· ABSADJ、MRKSET 可对标记方式的轴使用。

命令格式

```
@ ORGRTN[< 机器人编号 >] k [cr/lf]
```

应答格式

```
开始移动时 : RUN[cr/lf]
```

```
结束移动时 : END[cr/lf]
```

值 < 机器人编号 > 1 ~ 4
k 轴指定 : 1 ~ 6

含义 对< 机器人编号 >指定机器人的指定轴进行原点归复。< 机器人编号 >可以省略。当进行省略时，机器人 1 被指定。
在半绝对规格轴情况下，当执行原点复归后，实施绝对查找动作。

SAMPLE

```
命令 : @ ORGRTN 1 [cr/lf]
```

```
应答 : RUN[cr/lf] ..... 开始移动
```

```
END[cr/lf] ..... 结束移动
```

命令格式

```
@ INCH [< 机器人编号 >] km [cr/lf]
@ INCHXY [< 机器人编号 >] km [cr/lf]
@ INCHT [< 机器人编号 >] km [cr/lf]
```

应答格式

```
开始移动时：RUN[cr/lf]
结束移动时：END[cr/lf]
```

值

< 机器人编号 > 1 ~ 4
 k 轴指定：1 ~ 6
 m 移动方向 / +、-

含义 对< 机器人编号 >指定机器人的指定轴进行手动移动（寸动）。< 机器人编号 >可以省略。当进行省略时，机器人 1 被指定。机器人进行与在手动模式中操作手持编程器的微动键进行点动操作（每按一下按键，移动一定的量）时一样的动作。

各命令的移动量单位以及动作种类如下所示。

INCH 脉冲单位。仅指定轴移动。

INCHXY mm 单位。根据机器人的构成，机器人机械臂的前端沿正交坐标方向移动。

INCHT mm 单位。根据机器人的构成，安装在机器人机械臂前端的机械手移动。

SAMPLE

```
命令： @ INCH 1+ [cr/lf]
应答： RUN [cr/lf] ..... 开始移动
      END [cr/lf] ..... 结束移动
```

命令格式

```
@ JOG [< 机器人编号 >] km [cr/lf]
@ JOGXY [< 机器人编号 >] km [cr/lf]
@ JOGT [< 机器人编号 >] km [cr/lf]
```

应答格式

```
开始移动时：RUN[cr/lf]
结束移动时：END[cr/lf]
```

值

< 机器人编号 > 1 ~ 4
 k 轴指定：1 ~ 6
 m 移动方向 / +、-

含义 对< 机器人编号 >指定机器人的指定轴进行手动移动(点动)。< 机器人编号 >可以省略。当进行省略时，机器人 1 被指定。机器人进行与在手动模式中持续按住手持编程器的微动键时一样的动作。

微动命令为了持续进行动作，联机命令的持续执行处理(ˆV (=16H))输入需要设为200ms。未输入时会出现错误停止。

另外，移动开始后，当变为下列任何一种状态时，机器人停止。

- 到达软极限位置
- 联锁信号关闭
- 按下手持编程器的 STOP 键
- 通过联机命令输入了执行中断处理(ˆC (=03H))

各命令的移动量单位以及动作种类如下所示。

JOG 脉冲单位。仅指定轴移动。

JOGXY mm 单位。根据机器人的构成，机器人机械臂的前端沿正交坐标方向移动。

JOGT mm 单位。根据机器人的构成，安装在机器人机械臂前端的机械手移动。

SAMPLE

```
命令： @ JOG 1+[cr/lf]
应答： RUN[cr/lf] ..... 开始移动
      END[cr/lf] ..... 结束移动
```

5 数据文件操作命令

5.1 复制操作

1. 程序的复制

命令格式

```
@ COPY <程序名 1> TO <程序名 2> [cr/lf]
PGn
```

应答格式

```
处理开始时：RUN [cr/lf]
处理完成时：END [cr/lf]
```

值 <程序名 1> 复制源的程序名
(采用英文字母数字及“_”，在 32 个字符以内)
<程序名 2> 复制目的地的程序名
(采用英文字母数字及“_”，在 32 个字符以内)
n：程序编号 0 ~ 99

含义 从程序名称 1 或者利用程序编号指定的程序复制到程序名称 2。

SAMPLE

```
命令： @ COPY <TEST1> TO <TEST2> [cr/lf]
应答： RUN [cr/lf] ..... 处理开始
      END [cr/lf] ..... 处理完成
```

2. 坐标点的复制

命令格式

```
@ COPY Pmmmmm-Pnnnnn TO Pkkkkk [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
处理完成时：END [cr/lf]
```

值 mmmmm 复制源的起始坐标点编号：0 ~ 29999
nnnnn 复制源的结尾坐标点编号：0 ~ 29999
kkkkk 复制目的地的起始坐标点编号：0 ~ 29999

含义 将 Pmmmmm 至 Pnnnnn 的坐标点复制至 Pkkkkk。

SAMPLE

```
命令： @ COPY P101-P200 TO P1101 [cr/lf]
应答： RUN [cr/lf] ..... 处理开始
      END [cr/lf] ..... 处理完成
```

3. 坐标点注释的复制

命令格式

```
@ COPY PCmmmmm-PCnnnnn TO PCkkkkk [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值

mmmmm 复制源的起始坐标点注释编号：0 ~ 29999
 nnnnn 复制源的结尾坐标点注释编号：0 ~ 29999
 kkkkk 复制目的地的起始坐标点注释编号：0 ~ 29999

含义 将 PCmmmmm 至 PCnnnnn 的坐标点注释复制至 PCkkkkk。

SAMPLE

```
命令： @ COPY PC101-PC200 TO PC1101 [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
      END [cr/lf] ..... 处理完成
```

5.2

删除操作

1. 程序删除

命令格式

```
@ ERA <程序名> [cr/lf]
      PGn
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值

<程序名> 删除的程序名(采用英文字母数字及"_", 32个字符以内)
 n: 程序编号 0 ~ 99

含义 进行程序的删除操作。

SAMPLE

```
命令： @ ERA < TEST1 > [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
      END [cr/lf] ..... 处理完成
```

2. 删除坐标点

命令格式

```
@ ERA Pmmmmm-Pnnnnn [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值 mmmmm 删除的起始坐标点编号：0 ~ 29999
 nnnnn..... 删除的结尾坐标点编号：0 ~ 29999

含义 进行 Pmmmmm 至 Pnnnnn 的坐标点删除操作。

SAMPLE

```
命令： @ ERA P101-P200 [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
      END [cr/lf] ..... 处理完成
```

3 删除坐标点注释

命令格式

```
@ ERA PCmmmmm-PCnnnnn [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值 mmmmm 删除的起始坐标点注释编号：0 ~ 29999
 nnnnn..... 删除的结尾坐标点注释编号：0 ~ 29999

含义 进行 PCmmmmm 至 PCnnnnn 的坐标点注释删除操作。

SAMPLE

```
命令： @ ERA PC101-PC200 [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
      END [cr/lf] ..... 处理完成
```


4. 删除坐标点名称

命令格式

```
@ ERA PNmmmmm-PNnnnnn [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值 mmmmm 删除的起始坐标点注释编号：0 ~ 29999
nnnnn 删除的结尾坐标点注释编号：0 ~ 29999

含义 进行 PNmmmmm 至 PNnnnnn 的坐标点注释删除操作。

SAMPLE

```
命令： @ ERA PN101-PN200 [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

5. 删除托盘

命令格式

```
@ ERA PLm [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值 m..... 删除的托盘编号：0 ~ 39

含义 进行 PLm 的托盘数据删除操作。

SAMPLE

```
命令： @ ERA PL1 [cr/lf]
```

```
应答： OK [cr/lf]
```

6. 删除机械手

命令格式

```
@ ERA Hm [cr/lf]
```

应答格式

```
处理开始时 : RUN [cr/lf]
```

```
处理完成时 : END [cr/lf]
```

值 m..... 将要删除的机械手编号 : 0 ~ 31

含义 进行 Hm 的机械手定义数据的删除操作。

SAMPLE

```
命令 : @ ERA H2 [cr/lf]
```

```
应答 : RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

7. 删除位移

命令格式

```
@ ERA Sm [cr/lf]
```

应答格式

```
处理开始时 : RUN [cr/lf]
```

```
处理完成时 : END [cr/lf]
```

值 m..... 将要删除的位移编号 : 0 ~ 39

含义 进行 Sm 的位移数据的删除操作。

SAMPLE

```
命令 : @ ERA S1 [cr/lf]
```

```
应答 : RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

8. 删除区域判定输出设定

命令格式

```
@ ERA ACm [cr/lf]
```

应答格式

```
处理开始时 : RUN [cr/lf]
```

```
处理完成时 : END [cr/lf]
```

值 m..... 将要删除的区域判定输出设定编号 : 0 ~ 31

含义 进行 ACm 的区域判定输出设定的删除操作。

SAMPLE

```
命令 : @ ERA AC3 [cr/lf]
```

```
应答 : RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

9. 删除通用以太网端口

命令格式

```
@ ERA GPm [cr/lf]
```

应答格式

```
处理开始时 : RUN [cr/lf]
```

```
处理完成时 : END [cr/lf]
```

值 m..... 将要删除的通用以太网端口编号 : 0 ~ 15

含义 进行 GPm 的通用以太网端口的删除操作。

SAMPLE

```
命令 : @ ERA GP5 [cr/lf]
```

```
应答 : RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

命令格式

```
@ REN <程序名 1> TO <程序名 2> [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值 <程序名 1> 更改前的程序名(采用英语字母数字及"_", 32个字符以内)
 <程序名 2> 更改后的程序名(采用英语字母数字及"_", 32个字符以内)
 n: 程序编号 0 ~ 99

含义 进行指定程序的程序名更改。

SAMPLE

```
命令： @ REN < TEST1 > TO < TEST2 > [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

命令格式

@ ATTR	<程序名> PGn	TO s [cr/lf]
--------	--------------	--------------

应答格式

```
OK [cr/lf]
```

值 <程序名> 更改属性的程序名(采用英文字母数字及"_", 32个字符以内)
 s: 属性 RW: 可读写、RO: 只读、H: 隐藏
 n: 程序编号 0 ~ 99

含义 进行程序名称或者利用程序编号指定程序的属性变更操作。

SAMPLE

```
命令： @ ATTR < TEST1 > TO RO [cr/lf]
```

```
应答： OK [cr/lf]
```

1. 存储器区域的 初始化处理

命令格式

```
@ INIT <存储器区域> [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值	<存储器区域>	将要进行初始化处理的存储器区域 指定以下任意一个存储器区域。
	PGM	初始化程序区域
	PNT	初始化坐标点区域
	SFT	初始化位移区域
	HND.....	初始化机械手区域
	PLT.....	初始化托盘区域
	PCM	初始化坐标点注释区域
	PNM	对坐标点名称区域进行初始化
	ACO.....	对区域判定输出设定区域进行初始化
	GEP	对通用以太网端口设定区域进行初始化
	MEM.....	对上述区域 (PGM···PCM 的所有区域) 进行初始化
	PRM	初始化参数区域
	ALL	初始化所有区域 (MEM+PRM)

含义 进行存储器区域的初始化处理操作。

SAMPLE

```
命令： @ INIT PGM [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

2. 通信端口的初始处理

命令格式

```
@ INIT < 通信端口 > [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

值 <通信端口> 将要进行初始化处理的通信端口
指定下列任一通信端口。

CMU 对 RS-232C 端口进行初始化

ETH 对以太网端口进行初始化

含义 进行通信端口的初始化处理操作。
有关通信端口的初始值，请参阅控制器的手册。

SAMPLE

```
命令： @ INIT CMU [cr/lf]
```

```
应答： RUN [cr/lf] ..... 处理开始
```

```
END [cr/lf] ..... 处理完成
```

3. 警报记录的初始化处理

命令格式

```
@ INIT LOG [cr/lf]
```

应答格式

```
处理开始时：RUN [cr/lf]
```

```
处理完成时：END [cr/lf]
```

含义 对错误履历进行初始化。

SAMPLE

```
命令： @ INIT LOG [cr/lf]
```

```
应答： OK [cr/lf]
```

命令格式

```
@ READ <读出文件> [cr/lf]
```

应答格式

```
处理开始时：BEGIN [cr/lf]
```

(数据输出：内容依据读取文件。)

```
处理完成时：END [cr/lf]
```



要点

· 有关各文件的详细说明，请参阅第8章“数据文件详细说明”。

值 <读出文件> 表示读出文件。

含义 进行数据读出处理。

通过 RS-232C 的联机命令时，与以下命令的含义相同。

· SEND <读出文件> TO CMU

使用 Ethernet 时，与以下命令的含义相同。

· SEND <读出文件> TO ETH

分类	读出文件	记述格式	
		所有	个别
用户存储器	所有文件	ALL	_____
	程序	PGM	<bb···b>>
	坐标点	PNT	Pn
	坐标点注释	PCM	PCn
	坐标点名称	PNM	PNn
	参数	PRM	/cccccccc/
	位移定义	SFT	Sn
	机械手定义	HND	Hn
	托盘定义	PLT	PLn
	通用以太网端口	GEP	GPn
	输入输出名称	ION	iNMn (n)
	区域判定输出	ACO	ACn
	变量·常量	变量	VAR
数组变量		ARY	ab..by(x)
常量		_____	"cc..c"
状态	程序目录	DIR	<<bb···b>>
	参数目录	DPM	_____
	机器参照	MRF	_____
	机器参考(标记轴)	ARP	_____
	控制器构成	CFG	_____
	选配板	OPT	_____
	自我诊断	SCK	_____
设备	警报记录	LOG	_____
	DI 端口	DI ()	DIn ()
	DO 端口	DO ()	DOn ()
	MO 端口	MO ()	MOn ()
	TO 端口	TO ()	TOn ()
	LO 端口	LO ()	LOn ()
	SI 端口	SI ()	SIn ()
	SO 端口	SO ()	SOn ()
SIW 端口	SIW ()	SIWn ()	
SOW 端口	SOW ()	SOWn ()	
其他	文件结束代码	EOF	_____

a: 英文字母 n: 数字 b: 英文字母数字或下划线() c: 英文字母数字或特殊符号

x: 表达式(数组参数) y: 变量类型 i: 输入输出类型

SAMPLE

```
命令： @ READ PGM [cr/lf]
        @ READ P100 [cr/lf]
        @ READ DINM2(0) [cr/lf]
```

命令格式

```
@ WRITE <写入文件> [cr/lf]
```

应答格式

```
输入要求显示 ▶ READY [cr/lf]
```

```
输入结束后 ▶ OK [cr/lf]
```



要点

- 有关各文件的详细说明，请参阅第8章“数据文件详细说明”。

值 <写入文件>..... 表示写入文件。

含义 进行数据写入处理。

通过 RS-232C 的联机命令时，与以下命令的含义相同。

- SEND CMU TO <写入文件>

使用 Ethernet 时，与以下命令含义相同。

- SEND ETH TO <写入文件>

MEMO

- 在 DO、MO、TO、LO、SO、SOW 的各端口中，不可指定所有端口 (DO ()、MO () 等) 作为写入文件。
- 对于部分个别文件 (DO_n ()、MO_n () 等)，也有不可指定为写入文件的。详细说明，请参阅第8章“数据文件详细说明”。

分类	读出文件	记述格式	
		所有	个别
用户存储器	所有文件	ALL	——
	程序	PGM	<bb...b>
	坐标点	PNT	P _n
	坐标点注释	PCM	PC _n
	坐标点名称	PNM	PN _n
	参数	PRM	/cccccccc/
	位移定义	SFT	S _n
	机械手定义	HND	H _n
	托盘定义	PLT	PL _n
	通用以太网端口	GEP	GP _n
变量·常量	IO 名称	ION	iNM _n (n)
	区域判定输出	ACO	AC _n
变量·常量	变量	VAR	ab...by
	数组变量	ARY	ab...by(x)
设备	DO 端口	——	DO _n ()
	MO 端口	——	MO _n ()
	TO 端口	——	TO _n ()
	LO 端口	——	LO _n ()
	SO 端口	——	SO _n ()
	SOW 端口	——	SOW _n ()

a: 英文字母 n: 数字 b: 英文字母数字或下划线 (_) c: 英文字母数字或特殊符号
x: 表达式 (数组参数) y: 变量类型 i: 输入输出类型

SAMPLE

```
命令: @ WRITE PRM [cr/lf]
       @ WRITE P100 [cr/lf]
       @ WRITE DINM2(0) [cr/lf] f]
```


6.1 设置次序程序执行标志

命令格式

```
@ SEQUENCE k [cr/lf]
```

应答格式

```
OK [cr/lf]
```

值 k..... 执行标志 / 0 : 禁止、1 : 允许、3 : 允许 (DO 复位)

含义 进行次序程序执行标志的设置处理。

SAMPLE

```
命令 : @ SEQUENCE 1 [cr/lf]
```

```
应答 : OK [cr/lf]
```

6.2 日期设定

命令格式

```
@ DATE yy/mm/dd[cr/lf]
```

应答格式

```
OK[cr/lf]
```

值 yy/mm/dd..... 设定日期 (年月日)
 yy 公历的后 2 位数 (00 ~ 99)
 mm 月 (01 ~ 12)
 dd 日 (01 ~ 31)

含义 设定控制器的日期。

- 省略的项目使用当前值。
- 仅发送 [cr/lf] 时, 将不更新日期。
- 输入了不可能的日期时, 将发生 [5.2 : Data error] 的错误。

SAMPLE 1

仅更改日期时

```
//15 [cr/lf] ..... 将日期设置为 15 日
```

SAMPLE 2

```
命令 : @ DATE 14/01/14[cr/lf]
```

```
应答 : OK [cr/lf]
```



要点

- 仅更改公历或月时, 可省略后面的 "/" (斜线)。
- 示例)

```
设置为 2007 年.....07[cr/lf]
设置为 6 月 ...../06[cr/lf]
```



MEMO

命令格式

```
@ TIME hh:mm:ss[cr/lf]
```

应答格式

```
OK[cr/lf]
```

值

hh : mm : ss 当前时间
 hh 时 (00 ~ 23)
 mm 分 (00 ~ 59)
 ss 秒 (00 ~ 59)

含义 设定控制器的时间。

MEMO

- 省略的项目使用当前值。
- 仅发送 [cr/lf] 时，将不更新时间。
- 输入不可能的时间时，将发生 [5.2 : Data error] 的错误。

SAMPLE 1

仅更改分时

```
:20:[cr/lf] ..... 将分设置为 20 分
```

SAMPLE 2

命令： @ TIME 10:21:35[cr/lf]

应答： OK [cr/lf]

命令格式

@ “机器人语言” [cr/lf]

应答格式 1

OK [cr/lf] 或 NG=gg.eee [cr/lf]

应答格式 2

处理开始时：RUN [cr/lf] 或者 NG=gg.eee [cr/lf]

处理完成时：END [cr/lf] 或者 NG=gg.eee [cr/lf]

值	OK、END.....	命令正常结束
	NG.....	发生错误
	RUN.....	命令正常开始
	gg：警报组编号.....	0 ~ 99
	ccc：警报分类编号.....	0 ~ 999

- 含义** 进行机器人语言命令处理。
- 仅执行可单独执行的命令。
 - 命令格式取决于各命令。

SAMPLE 1

命令： @ SET DO (20) [cr/lf]

应答： OK [cr/lf]

SAMPLE 2

命令： @ MOVE P, P100, S=20 [cr/lf]

应答： RUN [cr/lf] 处理开始

END [cr/lf] 处理完成

命令格式

^C (=03H)

应答格式

NG=1.8

- 含义** 中断执行中的命令。

SAMPLE

命令： @ MOVE P, P100, S=20 [cr/lf]

^C

应答： NG=1.8[cr/lf]

第 11 章

附录

1	保留字一览表.....	11-1
2	机器人语言一览表：按字母顺序.....	11-3
3	机器人语言一览表：按功能.....	11-7
4	函数：按字母顺序.....	11-12
5	函数：按功能.....	11-14

以下语言因被作为机器人语言保留，所以无法用作标识符(变量等)。

A	DEC	HND	MOVET
ABS	DECEL	HOLD	MRF
ABSADJ	DEF	HOLDALL	MRKSET
ABSRPOS	DEGRAD	I	MSG
ACC	DELAY	IDIST	MSGCLR
ACCEL	DI	IF	MSPEED
ACCESS	DIM	IMP	N
ACO	DIR	INCH	NAME
ALL	DIST	INCHT	NEXT
ALM	DO	INCHXY	NOT
ALMRST	DPM	INIT	O
AND	DRIVE	INPUT	OFF
ARCHP1	DRIVEI	INT	OFFLINE
ARCHP2	DRV	ION	ON
ARM	DS	J	ONLINE
ARMCND	DSPEED	JL	OPEN
ARMSEL	E	JOG	OPT
ARMTYP	ELSE	JOGT	OR
ARP	ELSEIF	JOGXY	ORD
ARY	EMG	JTOXY	ORGORD
ASPEED	END	L	ORGRTN
ATN	ENDIF	LEFT	ORIGIN
ATN2	EOF	LEFTY	OUT
ATTR	EQV	LEN	OUTPOS
AXWGHT	ERA	LET	P
B	ERL	LINEMODE	P
BIN	ERR	LOAD	PATH
BREAK	ERROR	LOC1	PC
C	ETH	LOC2	PCM
CALL	ETHSTS	LOC3	PDEF
CASE	EXIT	LOC4	PGM
CFG	EXITTASK	LOC5	PGMTSK
CHANGE	F	LOC6	PGN
CHGPRI	FN	LOCF	PLS
CHR	FOR	LOG	PLT
CLOSE	FREE	LSHIFT	PMOVE
CMU	G	M	PNM
CNT	GEP	MAINPG	PNT
CONT	GEPSTS	MCHREF	PPNT
COPY	GO	MEM	PRINT
COS	GOSUB	MID	PRM
CURTQST	GOTO	MNS	PSHFRC
CURTRQ	H	MOD	PSHJGSP
CUT	HALT	MODE	PSHMTD
D	HALTALL	MOTOR	PSHRSLT
DATE	HAND	MOVE	PSHSPD
DBP	HEX	MOVEI	PSHTIME

PUSH	SET	SWI	VEL
PWR	SETGEP	SYNCHK	VER
R	SETPW	T	W
RADDEG	SFT	XYTOJ	WAIT
RBT	SGI	Y	WEIGHT
READ	SGR	YZ	WEND
REF	SHARED	Z	WHERE
REM	SHIFT	TAG	WHILE
REN	SI	TAN	WHRXY
RESET	SID	TASKS	WHRXYEX
RESTART	SIN	TCHXY	WRITE
RESUME	SIW	TCOUNTER	X
RETURN	SKIP	TEACH	XOR
RIGHT	SKIPTO	THEN	XY
RIGHTY	SO	TIM	XYTOJ
RSHIFT	SOD	TIME	Y
RUN	SOW	TIMER	YZ
RUNTO	SPEED	TO	Z
s	SQR	TOLE	ZX
S	START	TORQUE	
SCK	STEP	TSKECD	
SELECT	STOP	TSKMON	
SEND	STOPON	TSKPGM	
SEQCML	STR	V	
SEQUENCE	SUB	VAL	
SERVO	SUSPEND	VAR	

以下语言因被作为系统变量名，所以无法使用以这些名称起始的变量名（n：数值）。

Acn	FN	PCn	Sn
Dln	GPn	Pn	Son
DINMn	Hn	PNn	SONMn
DO _n	LO _n	Sln	TO _n
DONMn	MO _n	SINMn	

变量名的使用示例

- 作为被保留的机器人语言，其关键词本身无法使用，但是如果添加了**英语字母数字**，则可用作变量名。

示例：虽然无法使用 ABS，但是可使用 (ABS1) 或 (ABSX)。

- 作为被保留的系统变量，其**关键词起始的变量名全都无法使用**。添加了英文字母数字时也一样。

示例：除了 FN 以外，(FNA) 或 (FN123) 等也全都无法使用。

No.	命令	功能	联机	种类
A				
1	ABS	计算指定值的绝对值	-	命令
2	ABSRPOS	计算指定机器人的指定轴的机台参考值（返回原点的方式仅为标记方式时有效）	-	命令 / 函数
3	ACCEL	设定 / 获取指定机器人的加速度系数参数	○	命令 / 函数
4	ARCHP1	设定 / 获取指定机器人的圆弧位置 1 参数	○	命令 / 函数
4	ARCHP2	设定 / 获取指定机器人的圆弧位置 2 参数	○	命令 / 函数
5	ARMCND	获取指定机器人的当前机械臂状态	-	函数
6	ARMSEL	获取指定机器人的当前手系统设定	-	函数
7	ARMTYP	获取指定机器人的手系统设定	-	函数
8	ASPEED	设定 / 获取指定机器人的自动移动速度	○	命令 / 函数
9	ATN	计算指定值的反正切值	-	函数
9	ATN2	计算指定 X-Y 坐标的反正切值	-	函数
10	AXWGHT	设定 / 获取指定机器人的轴前端质量参数	○	命令 / 函数
C				
11	CALL	调用子过程	×	命令
12	CHANGE	对指定机器人的机械手进行切换	○	命令
13	CHGPRI	更改指定任务的优先级	○	命令
14	CHR\$	获取带有指定字符编码的字符	-	函数
15	COS	计算指定值的余弦值	-	函数
16	CURTQST	获取相对于指定轴额定扭矩的当前扭矩	-	函数
17	CURTRQ	获取指定机器人中指定轴的当前扭矩值	-	函数
18	CUT	强制结束执行中或暂停中的其他任务	○	命令
D				
19	DATES\$	通过 "yy/mm/dd" 形式的字符串计算日期	-	函数
20	DECEL	设定 / 获取指定机器人的减速比率参数	○	命令 / 函数
21	DEF FN	定义用户可使用的函数	×	命令
22	DEGRAD	将值转换为弧度 (↔ RADDEG)	-	函数
23	DELAY	仅等待指定的时间 (单位 ms)	×	命令
24	DI	获取并行端口的输入状态	-	函数
25	DIM	对数组变量的名称及元素数进行声明	×	命令
26	DIST	计算指定 2 个点之间的距离	-	函数
27	DO	将指定值输出至 DO 端口	○	命令
28	DRIVE	以轴为单位使指定机器人进行绝对位置移动	○	命令
28	DRIVE	(带有 T 选项时) 执行轴单位的绝对移动命令	○	命令
29	DRIVEI	以轴为单位使指定机器人进行相对位置移动	○	命令
E				
30	END SELECT	结束 SELECT CASE 语句	×	命令
31	END SUB	结束子过程的定义	×	命令
32	ERL	赋予发生错误的行编号	-	函数
32	ERR	赋予发生错误时的错误代码编号	-	函数
33	EXIT FOR	强制结束 FOR 语句 ~ NEXT 语句的循环	×	命令
34	EXIT SUB	强制结束使用 SUB ~ END 定义的子过程	×	命令
35	EXIT TASK	结束当前正在执行的任务	×	命令

No.	命令	功能	联机	种类
F				
36	FOR ~ NEXT	反复执行 FOR 语句的下一个语句至 NEXT 语句，直至变为控制反复的指定值为止	×	命令
G				
37	GOSUB ~ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	×	命令
38	GOTO	无条件跳转至标签所指定的行	×	命令
H				
39	HALT	停止程序并进行复位	×	命令
40	HALTALL	停止并复位所有的程序	×	命令
41	HAND	对指定机器人的机械手进行定义	○	命令
42	HOLD	暂停程序	×	命令
43	HOLDALL	暂停所有的程序	×	命令
I				
44	IF	根据条件分支控制流程	×	命令
45	INPUT	从手持编程器中对指定的变量赋值	○	命令
46	INT	获得舍去小数点以下部分的整数值	-	函数
J				
47	JTOXY	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)	-	函数
L				
48	LEFT\$	从指定的字符串左侧抽出指定位数的字符串	-	函数
49	LEFTY	将指定机器人的手系统设定为左手系统	○	命令
50	LEN	获得指定字符串的长度(字节数)	-	函数
51	LET	执行指定的赋值语句	○	命令
52	LO	将被指定的值输出至 LO 端口，并进行轴移动的禁止或解除	○	命令
53	LOCx	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据	-	命令 / 函数
54	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	函数
M				
55	MCHREF	计算指定机器人轴的原点归复动作及绝对值搜索动作的机台参考	-	函数
56	MID\$	从指定的字符串中抽出任意长度的字符串	-	函数
57	MO	将指定值输出至 MO 端口	○	命令
58	MOTOR	控制马达电源状态	○	命令
59	MOVE	使指定机器人的所有轴进行绝对移动	○	命令
60	MOVEI	使指定机器人的所有轴进行相对移动	○	命令
O				
61	OFFLINE	将指定的通信端口设置为脱机模式	○	命令
62	ON ERROR GOTO	在不停止程序的情况下跳转至标签所示的错误处理例程，或者显示错误提示，并停止程序的执行	×	命令
63	ON ~ GOSUB	根据条件，通过 GOSUB 语句跳转至所指定的各标签的子例程，并执行子例程	×	命令
64	ON ~ GOTO	根据条件跳转至标签所指定的各行	×	命令
65	ONLINE	将指定的通信端口设置为联机模式	○	命令
66	ORD	获得指定字符串起始字符的字符编码	-	函数

No.	命令	功能	联机	种类
67	ORGORD	设定 / 获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数	○	命令 / 函数
68	ORIGIN	对增量式规格的轴执行返回原点动作	○	命令
69	OUT	开启指定输出端口的比特, 并结束命令语句	×	命令
70	OUTPOS	设定 / 获取指定机器人的 OUT 有效位置参数	○	命令 / 函数
P				
71	PDEF	定义用于执行托盘移动命令的托盘	○	命令
72	PMOVE	执行指定机器人的托盘移动命令	○	命令
73	Pn	在程序中定义坐标点	○	命令
74	PPNT	创建托盘定义编号与托盘位置编号中指定的坐标点数据	-	函数
75	PRINT	在手持编程器的画面上显示字符串	○	命令
76	PSHFRC	设定 / 获取推压力参数	○	命令 / 函数
77	PSHJGSP	设定 / 获取推压判定速度阈值参数	○	命令 / 函数
78	PSHMTD	设定 / 获取推压方式参数	○	命令 / 函数
79	PSHRSLT	获取 PUSH 语句结束时的状态	-	函数
80	PSHSPD	设定 / 获取推压移动速度参数	○	函数
81	PSHTIME	设定 / 获取推压时间参数	○	函数
82	PUSH	以轴为单位进行推压动作	○	命令
R				
83	RADDEG	将值转换为度 (↔ DEGRAD)	-	函数
84	REM	记述注释语句	×	命令
85	RESET	关闭指定输出端口的比特	○	命令
86	RESTART	重新启动暂停中的其他任务	○	命令
87	RESUME	进行错误的恢复处理后, 恢复执行程序	×	命令
88	RETURN	使 GOSUB 中的分支处理返回至 GOSUB 的下一行	×	命令
89	RIGHT\$	从指定的字符串右侧抽出指定位数的字符串	-	函数
90	RIGHTY	将指定机器人的手系统设定为右手系统	○	命令
91	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	函数
S				
92	SELECT CASE ~ END SELECT	根据条件分支控制流程	×	命令
93	SEND	转送文件	○	命令
94	SERVO	对指定机器人中指定轴或所有轴的伺服 ON / OFF 状态进行控制	○	命令
95	SET	开启指定输出端口的比特	△	命令
96	SHARED	使不提交变量便可在子过程中参照	×	命令
97	SHIFT	指定位移变量, 然后针对由该位移数据所指定的机器人位移坐标进行设定	○	命令
98	SIN	计算指定值的正弦值	-	函数
99	Sn	在程序中定义位移坐标	○	命令
100	SO	将指定值输出至 SO 端口	○	命令
101	SPEED	对指定机器人的程序移动速度进行变更	○	命令
102	SQR	计算指定值的平方根	-	函数
103	START	设定指定程序的任务编号及优先顺序, 并且启动该程序	○	命令
104	STR\$	将指定值转换为字符串 (↔ VAL)	-	函数
105	SUB ~ END SUB	定义子过程	×	命令
106	SUSPEND	暂停执行中的其他任务	×	命令
107	SWI	切换执行程序, 并在执行编译后从第 1 行开始执行	×	命令

No.	命令	功能	联机	种类
T				
108	TAN	计算指定值的正切值	-	函数
109	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数递增值	-	函数
110	TIME\$	通过 "hh : mm : ss" 形式的字符串计算当前时间	-	函数
111	TIMER	用上午 0 点开始的秒计算当前时间	-	函数
112	TO	将指定值输出至 TO 端口	○	命令
113	TOLE	设定 / 获取指定机器人的公差参数	○	命令 / 函数
114	TORQUE	设定 / 获取指定机器人中指定轴的最大扭矩指令值	○	命令 / 函数
V				
115	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)	-	函数
W				
116	WAIT	等待直至 DI / DO 条件表达式成立为止 (带超时)	×	命令
117	WAIT ARM	等待指定机器人的轴动作结束	×	命令
118	WEIGHT	设定 / 获取指定机器人的前端质量参数	○	命令 / 函数
119	WEND	结束 WHILE 语句的命令区	×	命令
120	WHERE	通过关节坐标 (脉冲) 读出指定机器人机械臂的当前位置	-	函数
121	WHILE ~ WEND	控制反复	×	命令
122	WHRXY	通过正交坐标 (mm、度) 读出指定机器人机械臂的当前位置	-	函数
X				
123	XYTOJ	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)	-	函数

程序命令

普通命令

No.	命令	功能	联机	种类
25	DIM	对数组变量的名称及元素数进行声明	×	命令
51	LET	执行指定的赋值语句	○	命令
84	REM	记述注释语句	×	命令

算术命令

No.	命令	功能	联机	种类
1	ABS	计算指定值的绝对值	-	命令
9	ATN	计算指定值的反正切值	-	函数
9	ATN2	计算指定 X-Y 坐标的反正切值	-	函数
15	COS	计算指定值的余弦值	-	函数
22	DEGRAD	将值转换为弧度 (↔ RADDEG)	-	函数
26	DIST	计算指定 2 个点之间的距离	-	函数
46	INT	获得舍去小数点以下部分的整数	-	函数
54	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	函数
83	RADDEG	将值转换为度 (↔ DEGRAD)	-	函数
91	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	函数
98	SIN	计算指定值的正弦值	-	函数
102	SQR	计算指定值的平方根	-	函数
108	TAN	计算指定值的正切值	-	函数

日期、时间

No.	命令	功能	联机	种类
19	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期	-	函数
109	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数值	-	函数
110	TIME \$	通过 "hh : mm : ss" 形式的字符串计算当前时间	-	函数
111	TIMER	用上午 0 点开始的秒计算当前时间	-	函数

字符串操作

No.	命令	功能	联机	种类
14	CHR \$	获取带有指定字符编码的字符	-	函数
48	LEFT \$	从指定的字符串左侧抽出指定位数的字符串	-	函数
50	LEN	获得指定字符串的长度 (字节数)	-	函数
56	MID \$	从指定的字符串中抽出任意长度的字符串	-	函数
66	ORD	获得指定字符串起始字符的字符编码	-	函数
89	RIGHT \$	从指定的字符串右侧抽出指定位数的字符串	-	函数
104	STR \$	将指定数值转换为字符串 (↔ VAL)	-	函数
115	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)	-	函数

坐标点、坐标、位移坐标

No.	命令	功能	联机	种类
12	CHANGE	对指定机器人的机械手进行切换	○	命令
41	HAND	对指定机器人的机械手进行定义	○	命令
47	JTOXY	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)	-	函数
49	LEFTY	将指定机器人的手系统设定为左手系统	○	命令
53	LOCx	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据	-	命令 / 函数
73	Pn	在程序中定义坐标点	○	命令
74	PPNT	创建托盘定义编号与托盘位置编号中指定的坐标点数据	-	函数
90	RIGHTY	将指定机器人的手系统设定为右手系统	○	命令
99	Sn	在程序中定义位移坐标	○	命令
97	SHIFT	指定位移变量, 然后针对由该位移数据所指定的机器人位移坐标进行设定	○	命令
123	XYTOJ	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)	-	函数

分支命令

No.	命令	功能	联机	种类
33	EXIT FOR	强制结束 FOR 语句~ NEXT 语句的循环	×	命令
36	FOR ~ NEXT	反复执行 FOR 语句的下一个语句至 NEXT 语句, 直至变为控制反复的指定值为止	×	命令
37	GOSUB ~ RETURN	通过 GOSUB 语句跳转至指定标签的子例程, 并执行子例程	×	命令
38	GOTO	无条件跳转至标签所指定的行	×	命令
44	IF	根据条件分支控制流程	×	命令
63	ON ~ GOSUB	根据条件, 通过 GOSUB 语句跳转至所指定的各标签的子例程, 并执行子例程	×	命令
64	ON ~ GOTO	根据条件跳转至标签所指定的各行	×	命令
92	SELECT CASE ~ END SELECT	根据条件分支控制流程	×	命令
121	WHILE ~ WEND	控制反复	×	命令

错误控制

No.	命令	功能	联机	种类
62	ON ERROR GOTO	在不停止程序的情况下跳转至标签所示的错误处理例程, 或者显示错误提示, 并停止程序的执行	×	命令
87	RESUME	进行错误的恢复处理后, 恢复执行程序	×	命令
32	ERL	赋予发生错误的行编号	-	函数
32	ERR	赋予发生错误时的错误代码编号	-	函数

程序、任务控制

程序控制

No.	命令	功能	联机	种类
11	CALL	执行其他程序	×	命令
39	HALT	停止程序并进行复位	×	命令
40	HALTALL	停止所有程序，并且对任务 1 进行复位，其它任务结束	×	命令
42	HOLD	暂停程序	×	命令
43	HOLDALL	暂停所有程序	×	命令
107	SWI	切换执行程序，并在执行编译后从第 1 行开始执行	×	命令

任务控制

No.	命令	功能	联机	种类
13	CHGPRI	更改指定任务的优先级	-	命令
18	CUT	强制结束执行中或暂停中的其他任务	○	命令
35	EXIT TASK	结束当前正在执行的任务	×	命令
82	PUSH	以轴为单位进行推压动作	○	命令
86	RESTART	重新启动暂停中的其他任务	○	命令
103	START	设置指定任务的编号及优先级，并启动该任务	○	命令
106	SUSPEND	暂停执行中的其他任务	×	命令

机器人控制

机器人动作

No.	命令	功能	联机	种类
12	CHANGE	对指定机器人的机械手进行切换	○	命令
28	DRIVE	以轴为单位使指定机器人进行绝对位置移动	○	命令
29	DRIVEI	以轴为单位使指定机器人进行相对位置移动	○	命令
41	HAND	对指定机器人的机械手进行定义	○	命令
49	LEFTY	将指定机器人的手系统设定为左手系统	○	命令
58	MOTOR	控制马达电源状态	○	命令
59	MOVE	使指定机器人的所有轴进行绝对移动	○	命令
60	MOVEI	使指定机器人的所有轴进行相对移动	○	命令
68	ORIGIN	执行指定机器人中轴的原点归复动作	○	命令
72	PMOVE	执行指定机器人的托盘移动命令	○	命令
90	RIGHTY	将指定机器人的手系统设定为右手系统	○	命令
94	SERVO	对指定机器人中指定轴或所有轴的伺服 ON / OFF 状态进行控制	○	命令

状态获取

No.	命令	功能	联机	种类
2	ABSRPOS	计算指定机器人的指定轴的机台参考值（返回原点的方式仅为标记方式时有效）	-	命令 / 函数
5	ARMCND	获取指定机器人的当前机械臂状态	-	函数
6	ARMSEL	获取指定机器人的当前手系统设定	-	函数
7	ARMTYP	获取指定机器人的手系统设定	-	函数
16	CURTQST	获取相对于指定轴额定扭矩的当前扭矩	-	函数
55	MCHREF	计算指定机器人的原点归复动作及绝对值搜索动作的机台参考	-	函数
79	PSHRSLT	获取 PUSH 语句结束时的状态	-	函数
80	PSHSPD	设定 / 获取推压移动速度参数	○	函数
81	PSHTIME	设定 / 获取推压时间参数	○	函数
117	WAIT ARM	等待指定机器人的轴动作结束	×	命令
120	WHERE	通过关节坐标（脉冲）读出指定机器人机械臂的当前位置	-	函数
122	WHRXY	通过正交坐标（mm、度）读出指定机器人机械臂的当前位置	-	函数

状态更改

No.	命令	功能	联机	种类
3	ACCEL	设定 / 获取指定机器人的加速度系数参数	○	命令 / 函数
4	ARCHP1	设定 / 获取指定机器人的圆弧位置 1 参数	○	命令 / 函数
4	ARCHP2	设定 / 获取指定机器人的圆弧位置 2 参数	○	命令 / 函数
8	ASPEED	设定 / 获取指定机器人的自动移动速度	○	命令 / 函数
10	AXWGHT	设定 / 获取指定机器人的轴前端质量参数	○	命令 / 函数
20	DECEL	设定 / 获取指定机器人的减速比率参数	○	命令 / 函数
67	ORGORD	设定 / 获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数	○	命令 / 函数
70	OUTPOS	设定 / 获取指定机器人的 OUT 有效位置参数	○	命令 / 函数
71	PDEF	定义用于执行托盘移动命令的托盘	○	命令
76	PSHFRC	设定 / 获取推压力参数	-	命令 / 函数
77	PSHJGSP	设定 / 获取推压判定速度阈值参数	○	命令 / 函数
78	PSHMTD	设定 / 获取推压方式参数	○	命令 / 函数
101	SPEED	对指定机器人的程序移动速度进行变更	○	命令
113	TOLE	设定 / 获取指定机器人的公差参数	○	命令 / 函数
118	WEIGHT	设定 / 获取指定机器人的前端质量参数	○	命令 / 函数

输入输出、通信控制

输入输出控制

No.	命令	功能	联机	种类
23	DELAY	仅等待指定的时间(单位 ms)	×	命令
27	DO	将指定值输出至 DO 端口	○	命令
52	LO	将被指定值输出至 LO 端口, 并进行轴移动的禁止或解除	○	命令
57	MO	将指定值输出至 MO 端口	○	命令
69	OUT	开启指定输出端口的比特, 并结束命令语句	×	命令
85	RESET	关闭指定输出端口的比特	○	命令
95	SET	开启指定输出端口的比特	△	命令
100	SO	将指定值输出至 SO 端口	○	命令
112	TO	将指定值输出至 TO 端口	○	命令
116	WAIT	等待直至 DI/DO 条件表达式成立为止(带超时)	×	命令

通信控制

No.	命令	功能	联机	种类
65	ONLINE	将指定的通信端口设置为联机模式	○	命令
61	OFFLINE	将指定的通信端口设置为脱机模式	○	命令
93	SEND	转送文件	○	命令

No.	函数	种类	功能
A			
1	ABS	算术函数	计算指定值的绝对值
2	ABSRPOS	算术函数	计算指定机器人的指定轴的机台参考值（返回原点的方式仅为标记方式时有效）
3	ACCEL	算术函数	获取指定机器人的加速度系数参数
4	ARCHP1	算术函数	获取指定机器人的圆弧位置 1 参数
4	ARCHP2	算术函数	获取指定机器人的圆弧位置 2 参数
5	ARMCND	算术函数	获取指定机器人的当前机械臂状态
6	ARMSEL	算术函数	获取指定机器人的当前手系统设定
7	ARMTYP	算术函数	获取指定机器人的手系统设定
8	ASPEED	算术函数	设定 / 获取指定机器人的自动移动速度
9	ATN	算术函数	计算指定值的反正切值
9	ATN2	算术函数	计算指定 X-Y 坐标的反正切值
10	AXWGHT	算术函数	获取指定机器人的轴前端质量参数
C			
14	CHR\$	字符串函数	获取带有指定字符编码的字符
15	COS	算术函数	计算指定值的余弦值
16	CURTQST	算术函数	获取相对于指定轴额定扭矩的当前扭矩
17	CURTRQ	算术函数	获取指定机器人中指定轴的当前扭矩值
D			
19	DATE\$	字符串函数	通过 "yy/mm/dd" 形式的字符串计算日期
20	DECEL	算术函数	获取指定机器人的减速比率参数
22	DEGRAD	算术函数	将值转换为弧度 (↔ RADDEG)
26	DIST	算术函数	计算指定 2 个点之间的距离
E			
32	ERL	算术函数	发生错误的行编号
32	ERR	算术函数	赋予发生错误时的错误代码编号
I			
46	INT	算术函数	获得舍去小数点以下部分的整数
J			
47	JTOXY	坐标点函数	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)
L			
48	LEFT\$	字符串函数	从指定的字符串左侧抽出指定比特数的字符串
50	LEN	算术函数	获得指定字符串的长度 (字节数)
53	LOCx	坐标点函数	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据
54	LSHIFT	算术函数	仅将指定值的比特数向左偏移 (↔ RSHIFT)
M			
55	MCHREF	算术函数	计算指定机器人的原点归复动作及绝对值搜索动作的机台参考
56	MID\$	字符串函数	从指定的字符串中抽出任意长度的字符串
O			
66	ORD	算术函数	获得指定字符串起始字符的字符编码

No.	函数	种类	功能
67	ORGORD	算术函数	获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数
70	OUTPOS	算术函数	获取指定机器人的 OUT 有效位置参数
P			
74	PPNT	坐标点函数	创建托盘定义编号与托盘位置编号中指定的坐标点数据
76	PSHFRC	算术函数	设定 / 获取推压力参数
77	PSHJGSP	算术函数	设定 / 获取推压判定速度阈值参数
78	PSHMTD	算术函数	设定 / 获取推压方式参数
79	PSHRSLT	算术函数	获取 PUSH 语句结束时的状态
80	PSHSPD	算术函数	设定 / 获取推压移动速度参数
81	PSHTIME	算术函数	设定 / 获取推压时间参数
R			
83	RADDEG	算术函数	将值转换为度 (↔ DEGRAD)
89	RIGHT\$	字符串函数	从指定的字符串右侧抽出指定比特数的字符串
91	RSHIFT	算术函数	仅将指定值的比特数向右偏移 (↔ LSHIFT)
S			
98	SIN	算术函数	计算指定值的正弦值
102	SQR	算术函数	计算指定值的平方根
104	STR\$	字符串函数	将指定值转换为字符串 (↔ VAL)
T			
108	TAN	算术函数	计算指定值的正切值
109	TCOUNTER	算术函数	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数递增值
110	TIME\$	字符串函数	通过 "hh : mm : ss" 形式的字符串计算当前时间
111	TIMER	算术函数	用上午 0 点开始的秒计算当前时间
113	TOLE	算术函数	获取指定机器人的公差参数
114	TORQUE	算术函数	获取指定机器人中指定轴的最大扭矩指令值
V			
115	VAL	算术函数	将指定字符串标注的值转换为实际的数值 (↔ STR \$)
W			
118	WEIGHT	算术函数	获取指定机器人的前端质量参数
120	WHERE	坐标点函数	通过关节坐标 (脉冲) 读出指定机器人机械臂的当前位置
122	WHRXY	坐标点函数	通过正交坐标 (mm、度) 读出指定机器人机械臂的当前位置
X			
123	XYTOJ	坐标点函数	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)

相关坐标点

No.	函数名	功能
47	JTOXY	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)
53	LOCx	以轴为单位或者将位移数据作为要素单位来设定 / 获取坐标点数据
74	PPNT	创建托盘定义编号与托盘位置编号中指定的坐标点数据
120	WHERE	通过关节坐标 (脉冲) 读出指定机器人机械臂的当前位置
122	WHRXY	通过正交坐标 (mm、度) 读出指定机器人机械臂的当前位置
123	XYTOJ	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)

相关参数

No.	函数名	功能
2	ABSRPOS	计算指定机器人的指定轴的机台参考值 (返回原点的方式仅为标记方式时有效)
3	ACCEL	设定 / 获取指定机器人的加速度系数参数
4	ARCHP1	获取指定机器人的圆弧位置 1 参数
4	ARCHP2	获取指定机器人的圆弧位置 2 参数
5	ARMCND	获取指定机器人的当前机械臂状态
6	ARMSSEL	获取指定机器人的当前手系统设定
7	ARMTYP	获取指定机器人的手系统设定
10	AXWGHT	获取指定机器人的轴前端质量参数
16	CURTQST	获取相对于指定轴额定扭矩的当前扭矩
17	CURTRQ	获取指定机器人中指定轴的当前扭矩值
20	DECEL	获取指定机器人的减速比率参数
50	LEN	获得指定字符串的长度 (字节数)
55	MCHREF	计算指定机器人轴的原点归复动作的机台参考
66	ORD	获得指定字符串起始字符的字符编码
67	ORGORD	获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数
70	OUTPOS	获取指定机器人的 OUT 有效位置参数
79	PSHRSLT	获取 PUSH 语句结束时的状态
113	TOLE	获取指定机器人的公差参数
114	TORQUE	获取指定机器人中指定轴的最大扭矩指令值
118	WEIGHT	获取指定机器人的前端质量参数

相关数值运算

No.	函数名	功能
1	ABS	计算指定值的绝对值
9	ATN	计算指定值的反正切值
9	ATN2	计算指定 X-Y 坐标的反正切值
15	COS	计算指定值的余弦值
22	DEGRAD	将值转换为弧度 (↔ RADDEG)
26	DIST	计算指定 2 个点之间的距离
46	INT	获得舍去小数点以下部分的整数值
54	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)
83	RADDEG	将值转换为度 (↔ DEGRAD)
91	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)
98	SIN	计算指定值的正弦值
102	SQR	计算指定值的平方根
108	TAN	计算指定值的正切值
115	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)

相关字符串运算

No.	函数名	功能
14	CHR \$	获取带有指定字符编码的字符
19	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期
48	LEFT \$	从指定的字符串左侧抽出指定比特数的字符串
56	MID \$	从指定的字符串中抽出任意长度的字符串
89	RIGHT \$	从指定的字符串右侧抽出指定比特数的字符串
104	STR \$	将指定值转换为字符串 (↔ VAL)

相关参数

No.	函数名	功能
32	ERL	赋予发生错误的行编号
32	ERR	赋予发生错误时的错误代码编号
109	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 1ms 输出计数递增值
110	TIME \$	通过 "hh:mm:ss" 形式的字符串计算当前时间
111	TIMER	用上午 0 点开始的秒计算当前时间

索引

索引

A

ALL 文件 8-20

B

保留字一览表 11-1

编程示例

 基本篇 9-1

 应用篇 9-8

变量的清除 3-19

变量类型 3-4

变量名 3-3

变量文件 8-24

变为 WAIT 状态的命令 6-4

编译 7-167

标签 1-4

并行端口 7-39, 7-42

并行输出变量 3-9

并行输入变量 3-8

C

参数目录文件 8-23

参数文件 8-9

CASE 7-150

常量文件 8-27

程序

 复制 10-28

 切换 7-167

 删除 10-29

 停止 7-63

 暂停 7-70

程序级别 7-155

程序名 1-2

程序名更改 10-34

程序目录文件 8-21

程序属性更改 10-34

程序文件 8-2

串行端口 7-159

串行端口通信文件 8-45

串行双倍字输出 3-16

串行双倍字输入 3-15

串行输出变量 3-14

串行输入变量 3-13

串行字输出 3-16

串行字输入 3-15

初始处理

 通信端口 10-36

次序程序 1-2

 获取执行状态 10-17

 设置执行标志 10-39

COMMON 1-3

CONT 指定 7-90, 7-101

错误处理 7-111, 7-144, 7-145

错误代码编号 7-111

错误发生行编号 7-111

错误恢复处理 7-145

D

等待程序的执行 7-38

DI / DO 条件式 4-6

定时器输出变量 3-12

定义用户可使用的函数 7-36

定义坐标点 7-126

DI 文件 8-30

动态变量 3-18

动态数组的有效范围 3-18

动作速度 7-23

DO 文件 8-32

端口输出指定 7-104

读出文件 7-151

多任务 5-1, 6-1

E

EOF 文件 8-44

Ethernet 端口通信文件 8-50

F

返回原点顺序 7-116

赋值语句 7-80

G

更改手动速度 10-12

公差参数 7-173

关节坐标类型 4-5

关系运算符 4-1

H

函数：按功能 7-13, 11-14

函数：按字母顺序 7-11, 11-12

获取版本 10-18

获取次序程序执行状态 10-17

获取存储器剩余容量 10-21

获取当前位置 10-18

获取断点状态 10-16

获取访问级状态	10-16
获取返回原点状态	10-14
获取紧急停止状态	10-22
获取脉冲坐标系的当前位置	10-18
获取模式状态	10-17
获取启动 / 暂停状态任务的处理	10-19
获取任务运行状态	10-20
获取伺服状态	10-15
获取位移状态	10-20
获取 XY 坐标系的当前位置	10-19

I

IF	7-72
区块 IF 语句	7-73
简单 IF 语句	7-72

J

减速比率	7-35
减速度指定	7-102
加速度系数	7-17
加速度指定	7-102
静态变量	3-18
机器人语言一览表：按功能	11-7
机器人语言一览表：按字母顺序	11-3
就绪队列	6-3
机械臂锁定输出	7-83
机械臂锁定输出变量	3-11
机械手	
定义	7-65
定义文件	8-14
获取机械手状态	10-21
切换	7-27
局部变量	3-18, 7-164
绝对位置移动	7-90

K

控制代码	10-41
控制伺服状态	7-153

L

LABEL 语句	1-4
类型转换	3-6, 4-3
联机命令	
获取数值表达式的值	10-22
获取位移表达式的值	10-23
获取字符串表达式的值	10-23
获取坐标点表达式的值	10-23
联机命令一览表	10-1
逻辑运算符	4-2

LO 文件	8-36
-------	------

M

码垛	9-4, 9-10
命令语句格式	1-5
MO 文件	8-34

N

内部输出	7-88
内部输出变量	3-10
扭矩指令值	7-174

O

OUT 有效位置	7-119
----------	-------

Q

前端重量	7-179
全局变量	3-18, 8-24

R

任务	
编号	7-162
程序示例	6-8
重新启动	7-144
等待条件	6-4
定义	6-1
恢复启动	6-5
结束	7-59
排程	6-3
强制结束	7-33
启动	6-2, 7-162
删除	6-6
数据的共享	6-8
停止	6-7
优先级	6-1, 7-28, 7-162
暂停	7-166
中断	6-5
状态与变化	6-2
任务的状态	
NON EXISTEN	6-2
READY	6-2
RUN	6-2
STOP	6-2
SUSPEND	6-2
WAIT	6-2
RS-232C	9-17, 9-18

S

- SEQUENCE 1-2
- 删除坐标点 10-30
- 删除坐标点注释 10-30
- 拾放 9-12
- 实数型常量 2-1
- 使用位移坐标 9-3
- 使用坐标点编号 9-2
- 手动模式操作 10-12
- 手系统标志 4-5, 7-96, 7-107, 7-126, 8-4, 8-17
- 数据的读出处理 10-37
- 数据文件 8-1
 - 种类 8-1
- 数据写入处理 10-38
- 数值型常量 2-1
- 数组 3-5
- 数组变量的声明 7-40
- 数组变量文件 8-28
- 数组的角标 7-40
- 伺服
 - FREE 7-153
 - OFF 7-153
 - ON 7-153
- SI 文件 8-40
- SO 文件 8-42
- SOW 文件 8-48
- STOPON 条件指定 7-46, 7-53, 7-125, 7-140
- 算术赋值语句 7-80
- 算术运算 4-1
- 算术运算符 4-1

T

- TO 端口 7-172, 7-176
- 通信端口 7-110, 7-114
- TO 文件 8-38
- 托盘
 - 定义 7-121
 - 定义编号 7-121, 7-122
 - 定义文件 8-16
 - 删除 10-31
 - 位置编号 7-122
 - 移动 7-122

W

- 位移变量 3-8
- 位移赋值语句 7-82
- 位移坐标 7-156, 7-158
 - 定义文件 8-12

X

- 相对位置移动 7-105
- 写入文件 7-151
- 形参 7-164
- 系统变量 3-2, 3-7
- 系统生成 5-1
- XY 指定 7-46

Y

- 移动速度 7-160
- 以轴为单位进行绝对移动 7-43
- 以轴为单位进行相对移动 7-49
- 用户变量 3-2
- 右手系统 7-148
- 圆弧插补 7-93
- 圆弧插补移动指定 7-99, 7-124
- 圆弧位置 7-18
- 运算的优先级 4-3

Z

- 正交坐标类型 4-5
- 整数型常量 2-1
- 值传递与引用传递 3-6
- 指定比特 3-17
- 轴前端重量 7-25
- 轴指定 PTP 7-43
- 注释 1-5, 7-142
- 自动移动速度 7-23
- 字符串
 - 比较 4-4
 - 联结 4-4, 7-81
 - 运算 4-4
- 字符串赋值语句 7-81
- 字符型常量 2-2
- 子过程 7-26, 7-155, 7-164
- 子例程 7-61, 7-112
- 坐标点变量 3-7
- 坐标点的复制 10-28
- 坐标点赋值语句 7-81
- 坐标点数据
 - 类型 4-5
- 坐标点文件 8-4
- 坐标点注释的复制 10-29
- 坐标点注释文件 8-7
- 坐标平面指定 7-103
- 左手系统 7-78

修订记录

修订日期	修订内容
2014年 2月	1.00 版 初版
2014年 8月	1.10 版 误记修正等

程序设计说明书

RCX340

2014年8月

Version 1.10版

雅马哈发动机株式会社 IM事业部

禁止复制或转印本书的全部或部分内容。

联系我们

雅马哈发动机株式会社 IM事业部

YAMAHA MOTOR CO., LTD.
IM OPERATIONS Robot Business Division

邮编 435-0054 静冈县滨松市中区早出町882
822 Soude, NaKa-ku, Hamamatu, Shizuoka 435-0054, Japan

E-mail robotn@yamaha-motor.co.jp

最新版的使用说明书可从下记网站下载
<http://www.yamaha-motor.com.cn/robot/>

