

RCX Series

编程手册

前言

首先非常感谢您购买 YAMAHA 机器人控制器 RCX 系列。

本书记载了针对 YAMAHA 机器人控制器系列的机器人程序命令等说明。

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：RCX240、RCX141、RCX142、RCX40、RCX221、RCX222

在本书中，分别归纳后进行如下标注。

RCX240 RCX240、RCX141、RCX142、RCX40 (4 轴控制器)

RCX14x RCX141、RCX142、RCX40 (RCX240 除外的 4 轴控制器) ※

RCX22x RCX221、RCX222 (2 轴控制器)

※ 带有 RCX14x 标注时，不包括 RCX240。

用于对应版本标注等 RCX240 与其他 4 轴控制器不同的情况下使用。

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	1-7
6 注释	1-7
7 命令语句格式	1-7

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
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	坐标点元素变量	3-8
9.4	位移元素变量	3-9
9.5	并行输入变量	3-9
9.6	并行输出变量	3-10
9.7	内部输出变量	3-11
9.8	机械臂锁定输出变量	3-12
9.9	定时器输出变量	3-13
9.10	串行输入变量	3-14
9.11	串行输出变量	3-15
9.12	串行字输入	3-16
9.13	串行双倍字输入	3-16
9.14	串行字输出	3-17
9.15	串行双倍字输出	3-17
10	指定比特	3-18
11	变量的有效范围	3-19
11.1	动态变量的有效范围	3-19
11.2	静态变量的有效范围	3-19
11.3	动态数组的有效范围	3-19
12	变量的清除	3-20
12.1	动态变量的清除	3-20
12.2	静态变量的清除	3-20

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
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
3	坐标点数据的类型	4-5
4	DI / DO 条件表达式	4-6
第 5 章 多台机器人控制		
1	概要	5-1
2	各组命令一览表	5-2
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
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
4	多任务程序示例	6-8
5	数据的共享	6-8
第 7 章 次序功能		
1	关于次序功能	7-1
2	次序程序的创建	7-1
2.1	编程方法	7-1
2.2	编译	7-2

3	次序程序的执行	7-4
3.1	次序程序的逐步执行	7-4
4	次序程序的编程	7-5
4.1	赋值语句	7-5
4.2	输入输出变量	7-5
4.3	定时器定义语句	7-7
4.4	运算符（逻辑运算符）	7-7
4.5	运算的优先级	7-8
4.6	次序程序规格	7-8

第 8 章 机器人语言一览表

机器人语言一览表的查看方法			8-1
按字母顺序			8-3
按功能			8-8
函数：按字母顺序			8-15
函数：按功能			8-18
1	ABS	计算绝对值	8-20
2	ABSRPOS / ABSRPOS2	获取机器参照	8-21
3	ABSRST	绝对式马达轴返回原点动作	8-22
4	ACCEL / ACCEL2	设置 / 获取加速度系数参数	8-23
5	ARCH / ARCH2	设置 / 获取参数的圆弧位置	8-24
6	ARMCND / ARMCND2	获取机械臂状态	8-26
7	ARMTYPE / ARMTYPE2	水平多关节型机器人手系统	8-27
8	ATN	计算反正切值	8-28
9	ASPEED / ASPEED2	设置自动移动速度	8-29
10	AXWGHT / AXWGHT2	设置 / 获取轴前端重量	8-30
11	CALL	调用子过程	8-31
12	CHANGE / CHANGE2	切换机械手	8-32
13	CHGPRI	更改任务的优先级	8-33
14	CHR\$	计算带有指定字符编码的字符	8-34
15	COS	计算余弦值	8-35
16	CUT	强制结束启动中的其他副任务	8-36

17	DATE\$	获取日期	8-37
18	DECEL / DECEL2	设置 / 获取减速比率参数	8-38
19	DECLARE	声明使用了 COMMON 程序内的子例程或子过程	8-39
20	DEF FN	定义用户可使用的函数	8-41
21	DEGRAD	角度转换（度→弧度）	8-42
22	DELAY	在指定时间等待程序的执行	8-43
23	DI	获取并行端口的输入状态	8-44
24	DIST	计算 2 个点之间的距离	8-45
25	DIM	声明数组变量	8-46
26	DO	输出至并行端口	8-47
27	DRIVE / DRIVE2	以轴为单位进行绝对移动	8-48
28	DRIVEI / DRIVEI2	以轴为单位进行相对移动	8-54
29	END SELECT	结束 SELECT CASE 语句	8-58
30	END SUB	结束子过程的定义	8-59
31	ERR / ERL	获取错误编码 / 错误发生行编号	8-60
32	EXIT FOR	强制结束 FOR ~ NEXT 语句的循环	8-61
33	EXIT SUB	强制结束使用 SUB ~ END 定义的子过程	8-62
34	EXIT TASK	结束当前正在执行的任务	8-63
35	FOR ~ NEXT	变量超过指定值为止将循环	8-64
36	GOSUB ~ RETURN	跳转至子例程	8-65
37	GOTO	无条件跳转至指定的行	8-66
38	HALT	停止程序并进行复位	8-67
39	HAND / HAND2	机械手定义	8-68
	39.1 水平多关节型机器人时		8-68
	39.2 正交机器人时		8-71
40	HOLD	暂停程序	8-73
41	IF	评估条件表达式的值，按照条件执行命令语句。	8-74
	41.1 简单 IF 语句		8-74
	41.2 区块 IF 语句		8-75
42	INPUT	从手持编程器中对变量赋值	8-76
43	INT	舍去小数点以下部分	8-77
44	JTOXY / JTOXY2	轴单位制转换（脉冲→毫米）	8-78
45	LABEL 语句	在程序行中定义标签	8-79
46	LEFT\$	从左端抽出字符串	8-80
47	LEFTY / LEFTY2	将水平多关节机器人的手系统设置为左手系统	8-81
48	LEN	获取字符串的长度	8-82

49	LET	对变量赋值	8-83
50	LO	机械臂锁定输出	8-86
51	LOCx	以轴为单位设置 / 获取坐标点数据或位移数据	8-87
52	LSHIFT	比特的左位移	8-88
53	MCHREF / MCHREF2	获取机器参照	8-89
54	MID\$	从指定位置获取字符串	8-90
55	MO	将指定值输出至 MO 端口（内部输出）	8-91
56	MOVE / MOVE2	将机器人的所有轴进行绝对位置移动	8-92
57	MOVEI / MOVEI2	将机器人的所有轴进行相对位置移动	8-107
58	OFFLINE	将通信端口设置为脱机模式	8-111
59	ORD	计算字符编码	8-112
60	ON ERROR GOTO	在发生错误时跳转至指定标签	8-113
61	ON ~ GOSUB	执行由表达式的值指定的子例程	8-114
62	ON ~ GOTO	跳转至表达式的值所指定的标签	8-115
63	ONLINE	将通信端口设置为联机模式	8-116
64	ORGORD / ORGORD2	设置 / 获取机器人的返回原点顺序	8-117
65	ORIGIN	对增量式马达轴执行返回原点动作	8-118
66	OUT	开启指定端口的输出	8-119
67	OUTPOS / OUTPOS2	设置 / 获取机器人的 OUT 有效位置参数	8-120
68	PATH	设置主机器人轴的 PATH 的移动路径	8-122
69	PATH END	结束 PATH 的路径设置	8-128
70	PATH SET	开始 PATH 的路径设置	8-129
71	PATH START	开始 PATH 移动	8-131
72	PDEF	定义托盘	8-132
73	PMOVE / PMOVE2	执行机器人的托盘移动	8-133
74	Pn	定义坐标点	8-137
75	PPNT	创建托盘坐标点数据	8-138
76	PRINT	在手持编程器上显示指定表达式的值	8-139
77	RADDEG	单位转换（弧度→度）	8-140
78	REM	插入注释	8-141
79	RESET	关闭指定端口的比特或清除变量	8-142
80	RESTART	重新启动暂停中的其他任务	8-143
81	RESUME	错误恢复处理后恢复执行程序	8-144
82	RETURN	使 GOSUB 中的分支处理返回至 GOSUB 的下一行	8-145
83	RIGHT\$	从右端抽出字符串	8-146

84	RIGHTY / RIGHTY2	将水平多关节机器人的手系统设置为右手系统	8-147
85	RSHIFT	比特的右位移	8-148
86	Sn	定义位移坐标	8-149
87	SELECT CASE	执行由表达式的值指定的命令区	8-150
88	SEND	将读出文件的数据转发至写入文件	8-151
89	SERVO / SERVO2	控制伺服状态	8-153
90	SET	开启指定端口的比特	8-154
91	SHARED	使不提交变量便可在子过程中参照	8-155
92	SHIFT / SHIFT2	设置位移坐标	8-156
93	SIN	计算正弦值	8-157
94	SO	输出串行端口上指定的值	8-158
95	SPEED / SPEED2	更改程序移动速度	8-159
96	START	启动新任务	8-160
97	STR\$	将数值转换为字符串	8-161
98	SQR	计算平方根	8-162
99	SUB ~ END SUB	定义子过程	8-163
100	SUSPEND	暂停执行中的其他任务	8-165
101	SWI	切换执行程序	8-166
102	TAN	计算正切值	8-167
103	TCOUNTER	定时器、计数器	8-168
104	TIME\$	获取当前时间	8-169
105	TIMER	获取当前时间	8-170
106	TO	将指定值输出至 TO 端口	8-171
107	TOLE / TOLE2	设置 / 获取公差参数	8-172
108	TORQUE / TORQUE2	设置 / 获取最大扭矩指令值	8-173
109	TRQSTS / TRQSTS2	获取 DRIVE 语句结束时的状态	8-175
110	TRQTIME / TRQTIME2	设置 / 获取扭矩限制指定选项的超时时间	8-176
111	VAL	将字符串转换为数值	8-178
112	WAIT	等待条件表达式成立	8-179
113	WAIT ARM / WAIT ARM2	等待机器人轴动作结束	8-180
114	WEIGHT / WEIGHT2	设置 / 获取前端重量参数	8-181
115	WEND	结束 WHILE 语句的命令区	8-182
116	WHERE / WHERE2	获取机械臂当前位置（脉冲坐标）	8-183
117	WHILE ~ WEND	在条件成立时反复动作	8-184
118	WHRXY / WHRXY2	通过正交坐标获取机械臂当前位置	8-185

119 XYTOJ / XYTOJ2	将主组轴的正交坐标数据（毫米）转换为关节坐标数据（脉冲）	8-186
120 _SYSFLG	用于监视轴状态的标志	8-186

第 9 章 PATH 语句

1 概要	9-1
2 特点	9-1
3 使用方法	9-1
4 注意事项	9-2

第 10 章 数据文件详细说明

1 概要	10-1
1.1 数据文件种类	10-1
1.2 注意点	10-1
2 程序文件	10-2
2.1 整个程序	10-2
2.2 程序单位	10-3
3 坐标点文件	10-4
3.1 所有坐标点	10-4
3.2 坐标点单位	10-5
4 坐标点注释文件	10-6
4.1 所有坐标点注释	10-6
4.2 坐标点注释单位	10-7
5 参数文件	10-8
5.1 所有参数	10-8
5.2 参数单位	10-10
6 位移坐标定义文件	10-11
6.1 所有位移	10-11
6.2 位移单位	10-12
7 机械手定义文件	10-13

7.1	所有机械手	10-13
7.2	机械手单位	10-14
8	托盘定义文件	10-15
8.1	所有托盘定义	10-15
8.2	托盘定义单位	10-17
9	ALL 文件	10-19
9.1	所有 ALL 文件	10-19
10	程序目录文件	10-20
10.1	所有程序目录	10-20
10.2	程序单位	10-21
11	参数目录文件	10-22
11.1	所有参数目录	10-22
12	变量文件	10-23
12.1	所有变量	10-23
12.2	变量单位	10-25
13	常量文件	10-26
13.1	单个字符串	10-26
14	数组变量文件	10-27
14.1	所有数组变量	10-27
14.2	数组变量单位	10-28
15	DI 文件	10-29
15.1	所有 DI	10-29
15.2	DI 端口单位	10-30
16	DO 文件	10-31
16.1	所有 DO	10-31
16.2	DO 端口单位	10-32
17	MO 文件	10-33
17.1	所有 MO	10-33
17.2	MO 端口单位	10-34
18	LO 文件	10-35

18.1 所有 LO	10-35
18.2 LO 端口单位	10-36
19 TO 文件	10-37
19.1 所有 TO	10-37
19.2 TO 端口单位	10-38
20 SI 文件	10-39
20.1 所有 SI	10-39
20.2 SI 端口单位	10-40
21 SO 文件	10-41
21.1 所有 SO	10-41
21.2 SO 端口单位	10-42
22 错误提示履历文件	10-43
22.1 所有错误提示履历	10-43
23 机器参照文件	10-44
23.1 所有机器参照文件	10-44
24 EOF 文件	10-45
24.1 EOF 数据	10-45
25 串行端口通信文件	10-46
25.1 串行端口通信文件	10-46
26 SIW 文件	10-47
26.1 所有 SIW	10-47
26.2 SIW 单位	10-48
27 SOW 文件	10-49
27.1 所有 SOW	10-49
27.2 SOW 单位	10-50
28 Ethernet 端口通信文件	10-51
28.1 Ethernet 端口通信文件	10-51

第 11 章 用户程序示例

1 基本篇	11-1
-------	------

1.1	将坐标点数据直接写入程序中	11-1
1.2	使用坐标点编号	11-2
1.3	使用位移坐标	11-3
1.4	码垛	11-4
1.4.1	算出坐标点坐标	11-4
1.4.2	使用托盘移动	11-6
1.5	DI / DO（数字输入输出）动作	11-7
2	应用篇	11-8
2.1	2 点间的拾放	11-8
2.2	码垛	11-10
2.3	层堆部件的拾放	11-12
2.4	部件检查 1（多任务的示例）	11-14
2.5	涂胶	11-17
2.6	使用外部装置与 RS-232C 的连接（其 1）	11-18
2.7	使用外部装置与 RS-232C 的连接（其 2）	11-19
第 12 章	联机命令	
1	联机命令一览表	12-1
1.1	联机命令一览表：按功能	12-1
1.2	联机命令一览表：按字母顺序	12-4
2	按键操作	12-6
2.1	模式更改	12-6
2.2	自动模式操作	12-7
2.3	手动模式操作	12-9
3	UTILITY 操作	12-12
3.1	获取程序执行状态	12-12
3.2	复制	12-12
3.3	删除操作	12-14
3.4	程序名更改	12-16
3.5	程序属性更改	12-16
3.6	初始处理	12-17
3.7	显示语言设置	12-18
3.8	手动模式时的坐标系单位设置	12-19
3.9	手持编程器显示错误提示的初始化	12-20
3.10	UTILITY 模式设置	12-20

3.11	日期的确认及设置	12-22
3.12	时间的确认及设置	12-23
4	数据处理	12-24
4.1	获取显示语言	12-24
4.2	获取访问级状态	12-24
4.3	获取机械臂状态	12-25
4.4	获取断点状态	12-25
4.5	获取控制器构成	12-26
4.6	获取执行级别状态	12-26
4.7	获取模式状态	12-27
4.8	获取提示	12-28
4.9	获取返回原点状态	12-29
4.10	获取绝对式原点复位状态	12-29
4.11	获取伺服状态	12-30
4.12	获取次序程序执行状态	12-30
4.13	获取速度状态	12-31
4.14	获取坐标点单位坐标系	12-31
4.15	获取版本	12-32
4.16	获取当前位置	12-32
4.17	获取启动 / 暂停状态任务的处理	12-33
4.18	获取任务运行状态	12-34
4.19	获取位移状态	12-34
4.20	获取机械手状态	12-35
4.21	获取存储器剩余容量	12-35
4.22	获取紧急停止状态	12-36
4.23	经自我诊断获取出错状态	12-36
4.24	获取选配插槽状态	12-37
4.25	获取值	12-38
4.26	数据的读出处理	12-40
4.27	数据写入处理	12-41
5	机器人语言的单独执行	12-42
5.1	程序切换命令	12-42
5.2	其他机器人语言命令	12-43
6	控制代码	12-44
6.1	执行中断处理	12-44

第 13 章 IO 命令

1	概要	13-1
2	IO 命令格式	13-1
3	IO 命令的收发	13-3
4	IO 命令一览表	13-5
5	IO 命令详细说明	13-6
5.1	MOVE 移动命令	13-6
5.2	MOVEI 移动命令	13-7
5.3	托盘移动命令	13-7
5.4	微动命令	13-8
5.5	点动命令	13-8
5.6	坐标点示教命令	13-9
5.7	用于绝对式原点复位的移动命令	13-9
5.8	绝对式原点复位命令	13-10
5.9	返回原点命令	13-10
5.10	伺服命令	13-11
5.11	手动移动速度更改命令	13-12
5.12	自动移动速度更改命令	13-12
5.13	程序速度更改命令	13-12
5.14	位移指定更改命令	13-13
5.15	机械手指定更改命令	13-13
5.16	机械臂指定更改命令	13-13
5.17	坐标点显示单位指定命令	13-13

第 14 章 附录

1	保留字一览表	14-1
2	机器人语言一览表：按字母顺序	14-3
3	机器人语言一览表：按功能	14-8
4	函数：按字母顺序	14-15
5	函数：按功能	14-18
6	执行级别	14-20

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	1-7
6	注释	1-7
7	命令语句格式	1-7

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，任何人均可容易地学会，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 < CALL > 及 < SUB ~ END SUB >。



要点

- 关于用户定义函数，请参阅 < DEF FN >。

一程序记述一条命令语句。请注意，所有的行必须记述命令语句，如果有空白行，则在编译（创建执行对象）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 8 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 4 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) FUNCTION
- B) SEQUENCE
- C) _SELECT
- D) COMMON

各程序的功能如下所示。

A) FUNCTION

- 功能** 在 PROGRAM 模式及 MANUAL 模式中，按 USER 键可使用用户功能。
- 使用用户功能后，可在 PROGRAM 模式中编辑程序时，用功能键输入常用命令 (MOVE、GOTO 等)。
- 此外，在 MANUAL 模式中，可不执行程序直接用功能键进行 DO 输出等操作。
- FUNCTION 程序，使用用户功能中的功能键进行定义 (可在各功能键上任意录入所需内容)。

SAMPLE

```
' FOR MANUAL MODE
*M_F1:' DO (20) ALTERNATE
    DO (20) =~DO (20)..... 按键时，D0 (20) 的开 / 关反转
*M_F2:' DO (21) ALTERNATE
    DO (21) =~DO (21)..... 反转 DO (21)
:
*M_F6:' DO (25) MOMENTARY
    DO (25) =1 ..... 按下按键时将 DO (25) 设置为 1
    DO (25) =0 ..... 放开按键时将 DO (25) 设置为 0
*M_F7:' MOTION
    MOVE P, P1 ..... 向坐标点 1 移动
    MOVE P, P2 ..... 向坐标点 2 移动
:
' FOR PROGRAM MODE
*P_F1:' MOVE P, ..... 按键时，在程序中记述 "MOVE P,"
*P_F6:' MOVE L, ..... 按键时，在程序中记述 "MOVE L,"
*P_F2:' GOTO * ..... 按键时，在程序中记述 "GOTO *"
:
```

● 在 PROGRAM 模式中用于编辑的功能键录入

格式：
*P_F < n > : ' < 字符串 >

值 < n > 录入的功能键编号 (n = 1 ~ 15)
 < 字符串 > 录入并显示在功能键上的字符串



· < 字符串 > 中虽然最多可输入 65 个字符，但是菜单中最多只可显示 7 个字符。

SAMPLE
*P_F2 : ' MOVE P, 在 F2 键上录入 "MOVE P, "
*P_F8 : ' DELAY 在 F8 键上录入 "DELAY"

● 在手动模式中录入用于输出命令的功能键

格式：
*M_F < n > : ' < 字符串 >
< 输出语句 1 >
< 输出语句 2 >

值 < n > 录入的功能键编号 (n = 1 ~ 15)
 < 字符串 > 显示在功能键上的字符串
 < 输入输出语句 1 > 按下按键时执行的命令语句
 < 输入输出语句 2 > 放开按键时执行的命令语句



· < 字符串 > 中虽然最多可输入 65 个字符，但是菜单中最多只可显示 7 个字符。

SAMPLE
*M_F2: ' MOMENT 用 F2 键显示 "MOMENT"
DO (20) = 1 按下 F2 键时开启 DO (20)
DO (20) = 0 放开 F2 键时关闭 DO (20)
*M_F14: ' ALTER 用 F14 键显示 "ALTER"
DO (20) = ~DO (20) 按 F14 键时，反转 DO (20) 的输出状态

参照 详细说明，请参阅各控制器的手册。

B) SEQUENCE

功能 RCX 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，请参阅第 7 章 < 4.6 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，请参阅第 7 章 < 次序功能 >。

C) _SELECT

功能

可创建复位机器人程序时必须执行、选择的程序。

若要使用此功能，请使用 "_SELECT" 的程序名。

例如，在存在多个程序，并想要通过 DI 输入切换至所需程序等时，请创建进行程序切换的程序，并加上 "_SELECT" 的名称。

即使在执行其他程序时，如果在程序停止后输入复位，一定会返回至该程序。各复位处理的不同之处如下所示。

(另请参阅以下程序示例。)

1. 从手持编程器执行复位时，将出现是否将程序更改至 "_SELECT" 的应答等待画面(在该画面上按 "NO" 将进入是否复位现在执行中程序的选择画面)。
2. 当通过程序中的 HALT 命令、专用 DI (复位信号) 及联机命令进行复位时，选择 "_SELECT"。
3. 接通电源时的动作因“执行级别”的设定而异。选择了“接通电源时有程序复位”执行级别时，请在接通电源时进行复位并选择 "_SELECT"。

使用 DI3 () 的输入选择程序。

DI3 () =0 时，反复监视 DI 输入

DI3 () =1 ~ 3 时，选择各程序

非上述情况下，退出程序



要点

- 关于“执行级别”，请参阅控制器的手册。



要点

- 使用了 ON ERROR 语句后，即使不存在 SWI 语句中指定的程序名，也可在不出错退出的情况下进行循环。
- 变量 ERR 中输入了程序中所发生的错误代码。ERR=&0303 表示 "Program doesn't exist"。

SAMPLE

```
ON ERROR GOTO *ER1
*ST:
  SELECT CASE DI3 ( ) ..... 根据 DI3 ( ) 的值分支
    CASE 0
      GOTO *ST ..... 为 0 时，返回 *ST 反复执行
    CASE 1
      SWI < PART1 > ..... 为 1 时
    CASE 2
      SWI < PART2 > ..... 为 2 时
    CASE 3
      SWI < PART3 > ..... 为 3 时
    CASE ELSE
      GOTO *FIN ..... 非上述情况下，跳转至 *FIN 并结束
  END SELECT
  GOTO *ST
*FIN:
  HALT
*ER1:
  IF ERR=&H0303 THEN *NEXT_L... 发生 [无程序] 错误时，返回
  ON ERROR GOTO 0 ..... 发生其他错误时，退出
*NEXT_L:
  RESUME NEXT
```

参照

详细说明，请参阅本书中的各命令。

D) 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 DECLARE 语句中声明的 SUB *DISTANCE (A!, B!, C!) 及 *AREA 例程。

SAMPLE

程序名：SAMPLE1

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

程序名：SAMPLE2

```
DECLARE SUB *DISTANCE (A!, B!, C!)
DECLARE *AREA
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。

标贴名、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 16 个字符 (超过 16 个字符的部分将被舍去)。
- 可使用的标识符最多为 500 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 14 章 < 1 保留字一览表 >。

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。
- < > 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，请参阅第 8 章 < 45 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 75 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	单精度实数 -999,999.9 至 +999,999.9 指数形式单精度实数 $-1.0 \times 10^{38} \sim +1.0 \times 10^{38}$
字符型	字符串	75 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

1. 10 进制常量
从 -1,073,741,824 至 1,073,741,823 的整数。
2. 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)
3. 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

1. 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)。
2. 指数形式的单精度实数
范围： $-1.0 \times 10^{38} \sim +1.0 \times 10^{38}$
· 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



MEMO

- 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 75 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

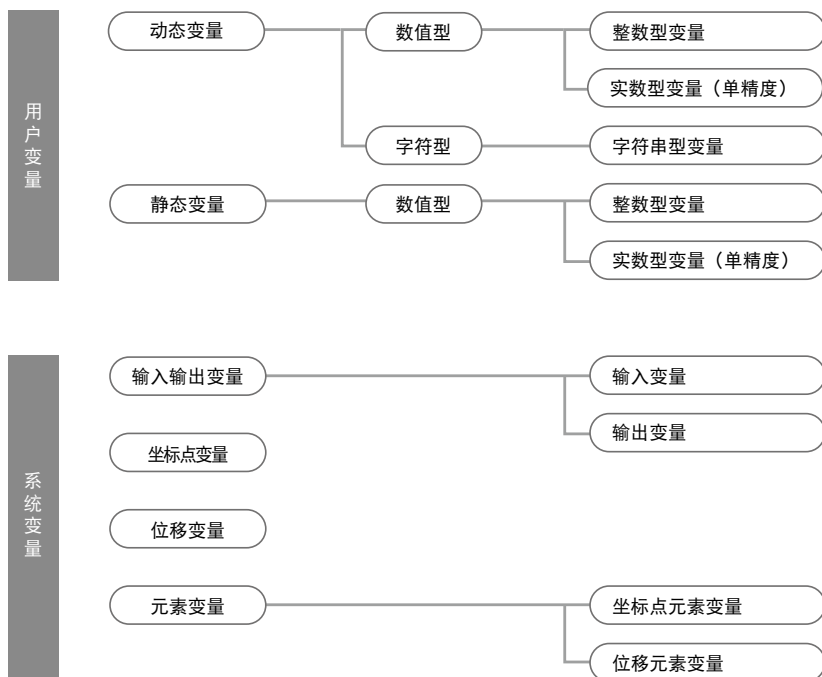
1	概要	3-1
2	用户变量与系统变量.....	3-2
3	变量名	3-3
4	变量类型.....	3-4
5	数组	3-5
6	赋值	3-5
7	类型转换.....	3-6
8	值传递与引用传递	3-6
9	系统变量.....	3-7
10	指定比特.....	3-18
11	变量的有效范围	3-19
12	变量的清除.....	3-20

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、编译、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-R7-00

参照 有关数组的详细说明，请参阅第 3 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
静态变量	字符型	字符串型变量 75 个字节以内的英文字母、数字、特殊字符、假名字符
	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
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 75 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
元素变量	坐标点元素变量	通过轴单位或手系统标志使用坐标点数据	LOCx（坐标点表达式）
	位移元素变量	通过元素单位使用位移数据	LOCx（位移表达式）

参照 详细说明，请参阅 < 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 16 个字符(超过 16 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 14 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 ~ 7)
实数型	SGRn (n: 0 ~ 7)

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< 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	CONT0%, 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
 - ASP_DEF!..... 实数型变量
 - ASP_DEF..... 实数型变量
- 分别作为不同变量处理。
- 均被作为相同变量处理。

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 ~ 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整数值为 -1.0*1038 ~ 1.0*1038。

示例：R1! = 10.31
R2! (2) = R1% + 1.98E3

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75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

- 数组变量均为动态变量(有关动态变量，请参阅第3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8章<49 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。程序编译时出错。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
坐标点元素变量	LOCx (<坐标点表达式>)	通过轴单位或手系统使用坐标点数据
位移元素变量	LOCx (<位移表达式>)	在元素范围内使用位移数据
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

Pnnnn 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 4 位数以下的数值或 "[" <表达式> "]" 表达坐标点编号。

可在 0 ~ 9999 的范围内指定坐标点编号。

示例：P0

P110

P[A]

P[START_POINT]

P[A (10)]



要点

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。

通过常量或表达式指定位移编号

格式

S_n 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 1 位数数值或 " [" < 表达式 > "] " 表达位移编号。

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。

通过轴单位及手系统标志指定坐标点数据

格式

LOC_x (< 坐标点表达式 >)

值 x X, Y, Z, R, A, B (指定轴), F (指定手系统标志)

功能 抽出坐标点数据指定的轴坐标或手系统标志，或者更改值。

示例 : $A(1) = LOCX(P10)$
→ 将 $P10$ 的 X 轴数据赋值给数组变量 $A(1)$
 $LOCZ(P[A]) = 100.0$
→ 将 $P[A]$ 的 Z 轴数据赋值给 100.0
 $LOCF(P100) = 1$
→ 将 $P100$ 的手系统标志更改为右手系统
(此时, $P100$ 的坐标点数据必须以毫米为单位)
 $B = LOCX(WHERE)$
→ 将当前的 X 轴马达赋值给数组变量 B
 $C(3) = LOCX(WHRXY)$
→ 将当前机械臂位置的 X 轴赋值给数组变量 $C(3)$
 $D = LOCX(JTOXY(WHERE))$
 $E = LOCX(XYTOJ(WHRXY))$

要点

· 手系统标志在软件版本 Ver.8.08 以上时有效。

要点

· 在水平多关节型机器人上，仅当坐标点数据以毫米为单位时，手系统标志才有效。
· 手系统标志的值为 0 (不指定)、1 (右手系统)、2 (左手系统)。

要点

· 在以下软件版本条件下， $WHRXY$ / $WHRXY2$ 有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

MEMO

· $JTOXY$ 是用于处理 < 坐标点表达式 > 的命令, " $JTOXY(LOC_x(WHERE))$ " 或 " $XYTOJ(LOC_x(WHRXY))$ " 会出错。

9.4 位移元素变量

以轴为单位处理位移数据

格式

LOCx (< 位移表达式 >)

值 x : 指定轴 X, Y, Z, R

功能 抽出位移数据指定的轴坐标，或者更改值。

示例：A (1) =LOCX (S1)
→将 S1 的 X 数据赋值给数组变量 A (1)
LOCX (S[A]) =45.0
→将 S[A] 的 R 数据更改为 45.0°

9.5 并行输入变量

表示并行输入信号的状态

格式 1

DIm ([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)



- 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。

9.7 内部输出变量

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
 - 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 0、1 是专用的内部输出变量, 只可参照 (无法更改)。

- 1. 端口 0 中保存了按顺序从 0 比特开始 1 ~ 8 轴原点传感器的状态。
原点传感器开启时为 1, 关闭时为 0。
- 2. 端口 1 中保存了按顺序从 0 比特开始 1 ~ 8 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 0	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
	原点传感器状态 0 : 关闭 / 1 : 开启							
端口 1	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
	保持状态 0 : 未保持 / 1 : 保持 (未使用轴为 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。

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。

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
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。
端口只有 1 个, 按照从 0 比特开始的顺序依次对应 1 ~ 8 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。(XX : 机械臂锁定有效轴 例如 : M1 S1)

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
- 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, 请参阅第 7 章 < 4.2 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)



· 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

SIm ([b, . . . , b])

格式 2

SI (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%=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。

9.11 串行输出变量

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。



要点

- 串行字输入在软件版本 Ver.8.08 以上时有效。



要点

- 串行双倍字输入在软件版本 Ver.8.08 以上时有效。

在串行输出的字信息中显示输出或输出状态

格式

SOW (m)



要点

- 串行字输出在软件版本 Ver.8.08 以上时有效。

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)



要点

- 串行双倍字输出在软件版本 Ver.8.08 以上时有效。

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，请使用以下任意一种方法。

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

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 ～ 7)
实数型	SGRn (n : 0 ～ 7)

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 8 章＜ 91 SHARED＞）。

12 变量的清除

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)。
- 编译了程序时。
- 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。
- 在系统模式的初始处理中对以下任意存储器进行了初始处理时。
 1. 程序存储器 (SYSTEM > INIT > MEMORY > PROGRAM)
 2. 所有存储器 (SYSTEM > INIT > MEMORY > ALL)
- 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。

- 在系统模式的初始处理中对以下存储器进行了初始处理时。
 - 所有存储器 (SYSTEM > INIT > MEMORY > ALL)
- 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 MOD 2 → A=1 (15/2=7...1)
 A=17.34 MOD 5.98 → A=2 (17/5=3...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 → 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!。

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

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 = 4	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)时等使用。

- 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

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。
- 关节坐标类型、正交坐标类型的数据格式均是 XYZRAB 通用的。
- + 可以省略。
- 手系统标志在软件版本 Ver.8.08 以上时有效。

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 ~ 9999。

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 2 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

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	各组命令一览表	5-2

RCX 控制器可对多台机器人进行控制。

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 2 台机器人或附加轴的设置。

机器人的轴将被以下任意一个分组设置。

- 主组 (6 轴)
- 主组 (6 轴) + 副组 (2 轴) (使用了 YC-LINK 选件时)
- 主组 (4 轴) + 副组 (4 轴) (使用了 YC-LINK 选件时)
- 主组 (2 轴) + 副组 (2 轴)
- 主组 (3 轴) + 副组 (3 轴) (仅可在使用了 YC-LINK 选件且软件版本为 RCX14x Ver.8.69 以上、RCX22x Ver.9.16 以上时才设置)

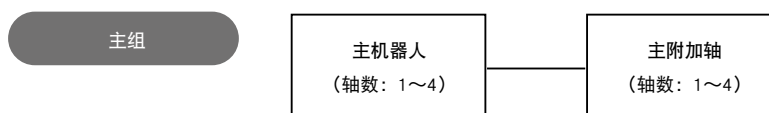
此外，主组由主机器人及主附加轴构成，副组由副机器人及副附加轴构成。

在 1 台机器人上不使用附加轴时，将变为仅限主组的主机器人的设置。

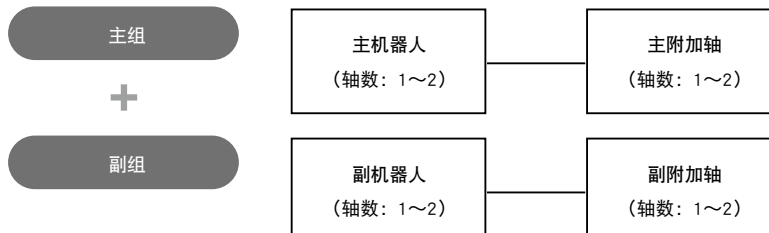
未设置主附加轴及副附加轴时，则为主组 = 主机器人，副组 = 副机器人。

轴构成

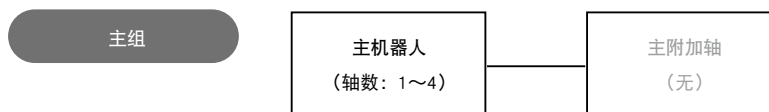
1. 仅主组时



2. 主组与副组时



3. 1 台机器人不使用附加轴时



4. 未设置主附加轴和副附加轴时



35501-R7-00

机器人动作、坐标控制等机器人专用的主要命令如下。

运算符	主组		副组	
机器人动作	DRIVE MOVE PMOVE WAIT ARM	DRIVEI MOVEI SERVO	DRIVE2 MOVE2 PMOVE2 WAIT ARM2	DRIVEI2 MOVEI2 SERVO2
坐标控制	CHANGE LEFTY SHIFT	HAND RIGHTY	CHANGE2 LEFTY2 SHIFT2	HAND2 RIGHTY2
状态更改	ACCEL ASPEED DECEL OUTPOS TOLE	ARCH AXWEIGHT ORGORD SPEED WEIGHT	ACCEL2 ASPEED2 DECEL2 OUTPOS2 TOLE2	ARCH2 AXWEIGHT2 ORGORD2 SPEED2 WEIGHT2
坐标点运算	JTOXY XYTOJ	WHERE	JTOXY2 WXTOJ2	WHERE2
参数更改	ACCEL AXWEIGHT ORGORD TOLE	ARCH DECEL OUTPOS WEIGHT	ACCEL2 AXWEIGHT2 ORGORD2 TOLE2	ARCH2 DECEL2 OUTPOS2 WEIGHT2
状态参照	ABSRPOS ARMTYPE	ARMCND MCHREF	ABSRPOS2 ARMTYPE2	ARMCND2 MCHREF2
PATH 控制	PATH PATH SET	PATH END PATH START		
扭矩控制	DRIVE（带有扭矩限制指定选项时） TORQUE TRQTIME	TRQSTS	DRIVE2（带有扭矩限制指定选项时） TORQUE2 TRQTIME2	TRQSTS2

- 主机机器人的动作使用 MOVE、MOVEI 命令，副机器人的动作使用 MOVE2、MOVEI2 命令
- 使用 MOVE / MOVE2、MOVEI / MOVEI2、PMOVE / PMOVE2 命令无法使设置为附加轴的轴动作。
需使用 DRIVE / DRIVE2、DRIVEI / DRIVEI2 命令
- 在 SERVO 或 SERVO2 命令中指定了所有轴时，主组及副组的所有轴均包括在内

MEMO

- 在 RCX14x : Ver.8.64 以下版本，RCX22x : Ver.9.11 以下版本的情况下，有以下限制。
- MOVE 语句的直线插补、圆弧插补只可执行任务 1（主任务）及直接命令。
- MOVE2 语句只可进行 PTP 控制。无法执行直线插补及圆弧插补。

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

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 8 的 8 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
- 任务 1（主任务）的优先级固定为 32，任务 2 至 8（副任务）的优先级可在 17～47 之间任意设置
- 数字越小，优先级越高，数字越大，优先级越低（高：17 ⇔ 47：低）

MEMO

- 在 RCX14x：Ver.8.64 以下版本、RCX22x：Ver.9.11 以下版本情况下，只可在任务 1（主任务）中执行 MOVE 语句的直线插补及圆弧插补。

任务指的是程序中记载的一系列工作的集合，其定义如下。

1. 在想要作为任务定义命令的区块首行加上标签。
2. 在任务 1（主任务）的 START 语句中指定上文中定义的标签，分配任务编号，然后启动。

只可对任务 2 至 8（副任务）进行任务定义。任务 1（主任务）被自动定义。

MEMO

- 虽然所有任务被集合在一个程序中进行记述，但是动作时各任务被分别进行并行处理。

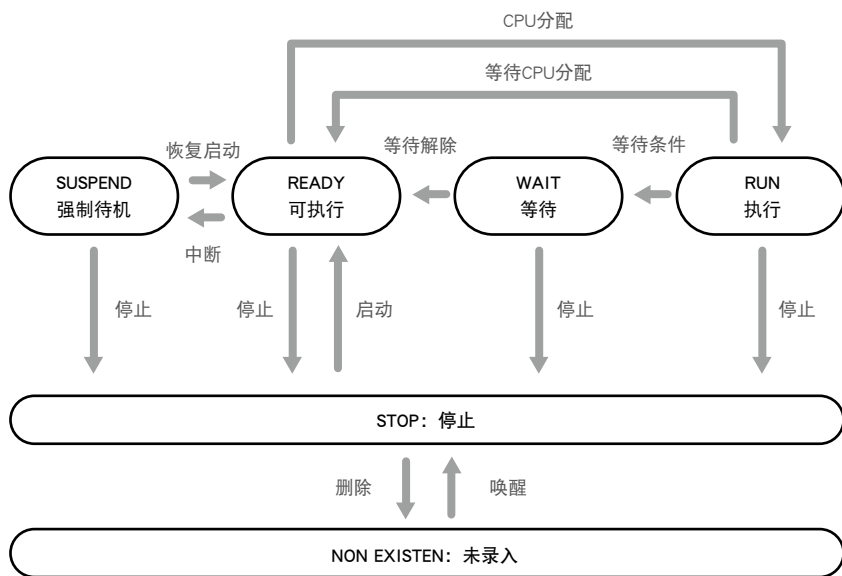
SAMPLE

```
' 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
' 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 *IOTASK ..... 任务 2 的处理至此处为止
EXIT TASK
```

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 EXISTEN 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-R7-00

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。

3.2 任务的排程

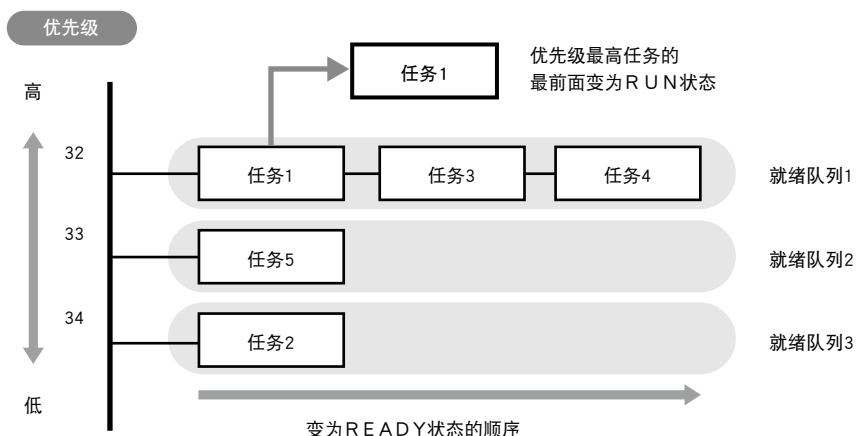
任务的排程，指的是将 READY (可执行) 状态的任务分配至 CPU 并决定执行的顺序。

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



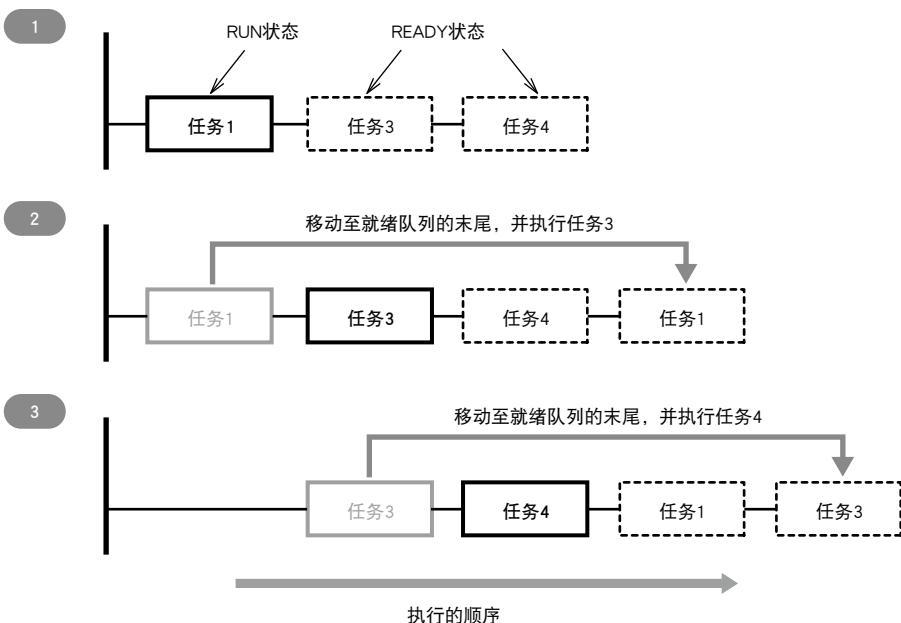
要点

- 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。

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

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的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	MOVE1	MOVE2	MOVE12
		DRIVE	DRIVE1	DRIVE2	DRIVE12
		PMOVE	PMOVE2	SERVO	SERVO2
		WAIT ARM	WAIT ARM2		
	参数命令	ACCEL	ACCEL2	ARCH	ARCH2
		AXWEIGHT	AXWEIGHT2	DECEL	DECEL2
		OUTPOS	OUTPOS2	TOLE	TOLE2
		ORGORD	ORGORD2	WEIGHT	WEIGHT2
	机器人状态更改命令	CHANGE	CHANGE2	SHIFT	SHIFT2
		LEFTY	LEFTY2	RIGHTY	RIGHTY2
		ASPEED	ASPEED2	SPEED	SPEED2
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

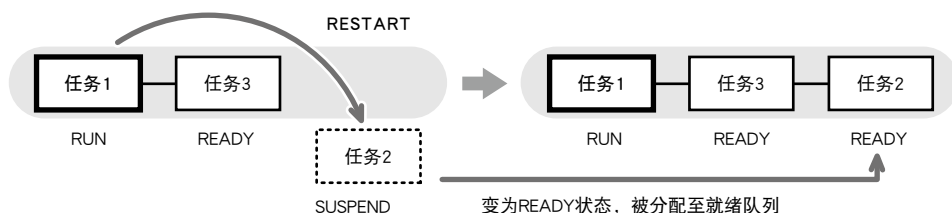
3.5 任务的恢复启动 (RESTART)

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)



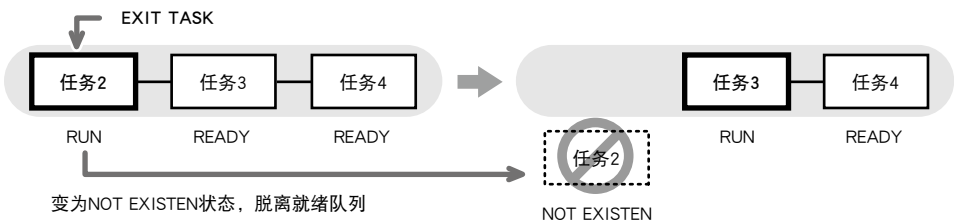
35605-R7-00

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)



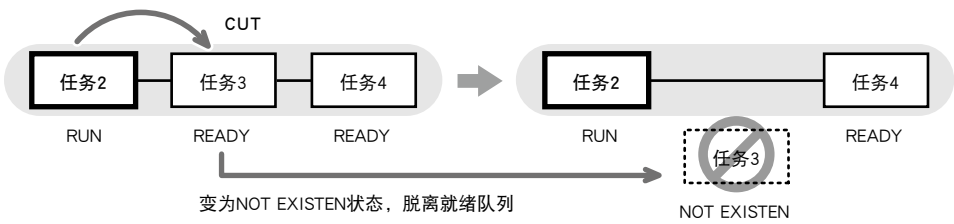
35606-R7-00

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。

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。

4 多任务程序示例

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
```

5 数据的共享

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。

第 7 章

次序功能

1	关于次序功能	7-1
2	次序程序的创建	7-1
3	次序程序的执行	7-4
4	次序程序的编程	7-5

RCX 控制器与通常的机器人程序不同，可对机器人的输入输出 (DI, DO, MO, LO, TO, SI, SO) 运行高速处理程序 (次序程序)。即，运行次序程序时，与机器人程序加在一起，同时执行了 2 个程序。



要点

- 执行次序程序过程中，输出专用输出“DO12：次序程序运行过程中”的信号。

次序程序与机器人程序的执行或停止无关，当专用输入“DI10：次序控制”为 ON 时，次序程序在自动模式及手动模式中将按照一定周期运行。

在控制器电源 ON (通常为手动模式) 的同时开始执行该程序，可对传感器、按钮开关、电磁阀等的输入输出信号进行监视和动作。

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。

在次序功能运行过程中，即使执行程序复位，通用输出也不会被复位。

但是，在编译次序程序时，也可执行复位设置。

有关执行次序程序所需的设置，请参阅 < 3 次序程序的执行 >。

2.1

编程方法

若要使用次序功能，必须创建次序程序。

首先，在程序模式中，以 "SEQUENCE" 的程序名创建文件。加上该名称后，控制器将自动判断为次序程序。

次序程序录入

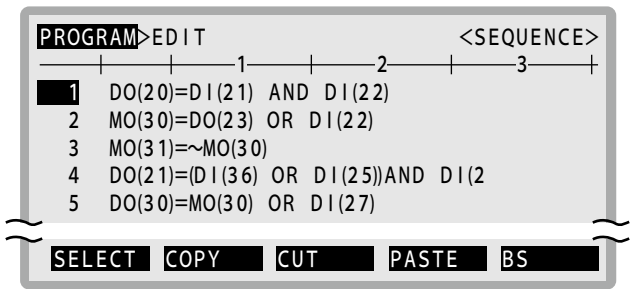
PROGRAM>DIR					<TEST10 >	
No.	Name	Line	Byte	RW/RO		
1	TEST10	12	145	RW		
2	LOCATE20	25	320	RW		

Enter program name >SEQUENCE

36701-R7-00

接着输入程序，方法与普通的机器人程序创建一样。可输入的命令将在后面介绍。

创建次序程序

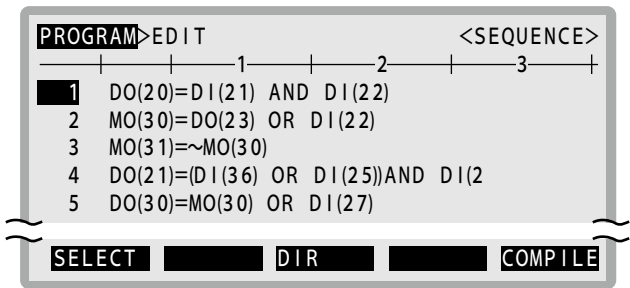


36702-R7-00

2.2 编译

完成编辑的程序进行次序程序的编译。与机器人程序中的编译一样，在程序模式最高级的画面上按 F5 键。

次序程序



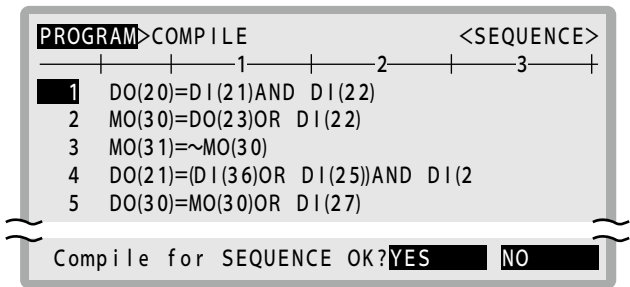
36703-R7-00

将显示是否编译次序程序的确认提示。此时按 F4 将开始编译。如果按 F5 取消，则将显示普通的机器人程序编译处理画面。

按 F4 键进行次序程序的编译。

请在编译主程序之前，先对次序程序进行编译。

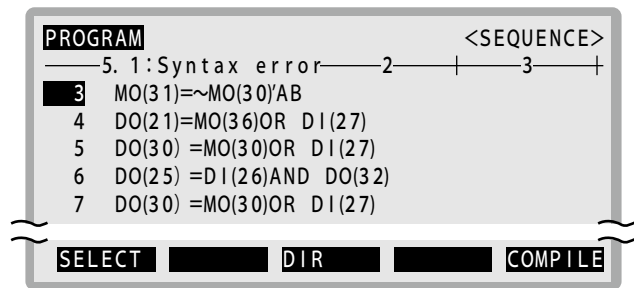
次序程序的编译



36704-R7-00

如果已输入的命令语句格式错误，错误提示与错误行将显示程序列表。如果正常结束，将从程序列表的最前面开始显示。

编译错误



36705-R7-00

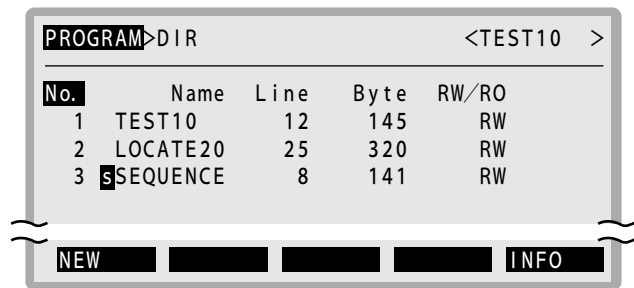
MEMO

· 在以下情况下，次序程序专用的执行程序将被删除，且其前面将不显示 "s"。此时，UTILITY 模式的次序程序的执行变为禁止状态。

1. 已删除次序程序时
2. 已编辑次序程序时
3. 对次序程序进行了普通的机器人程序编译时（在选择了 SEQUENCE 程序状态下切换至自动模式时也执行相同的处理。）
4. 执行了程序数据的初始化时
5. 发生了 [9.39 : Sequence object destroyed] 错误时

此外，在正常结束时如果显示目录，则在 "SEQUENCE" 的程序名前面将显示 "s"。这表示次序程序的编译正常结束并正在创建执行程序。

次序执行程序



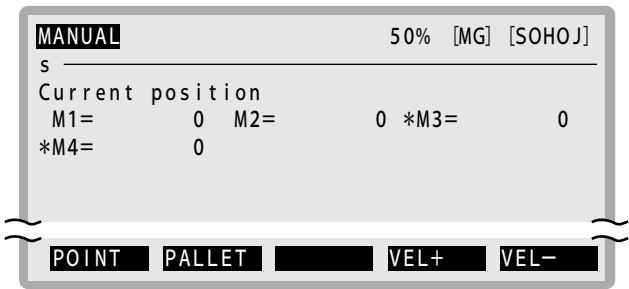
36706-R7-00

执行次序程序所需的条件如下所示。请注意，如果以下项目未全部成立，将无法执行。

- 1. 必须创建次序程序的执行程序。
- 2. 必须在 UTILITY 模式中将次序程序的执行设置为许可状态。(关于 UTILITY 模式，请参阅控制器的手册。)
- 3. 外部输入的次序控制输入 (DI10) 的触点必须关闭。
- 4. 必须处于手动模式或自动运行模式中。

如果上述项目全部成立，将执行次序程序。执行过程中，画面第 2 行的左端将显示 "s"。

次序程序执行状态



36707-R7-00

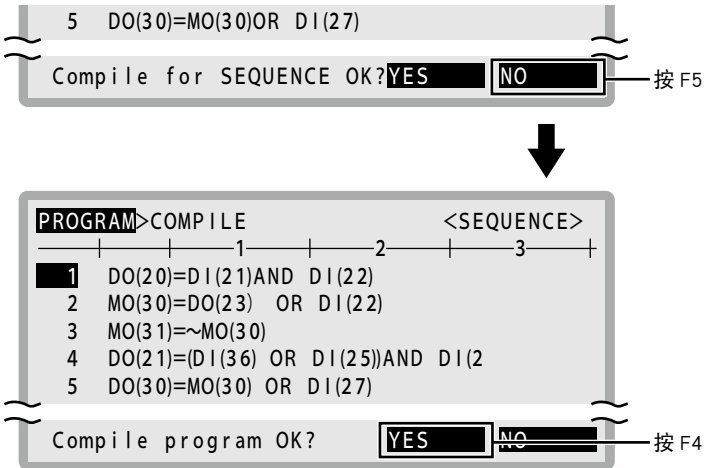
关于 UTILITY 模式中的设置方法，请参阅控制器的手册。

3.1

次序程序的逐步执行

如果要对次序程序每行进行逐步执行并确认动作，请按照以下步骤进行操作。
如果在编译次序程序时按 F5 键取消，则将显示普通的机器人程序编译画面。
按 F4 键创建普通程序的执行程序，然后在自动运行模式中逐步执行确认动作。

次序程序的逐步执行



36708-R7-00

次序程序的记述只可使用输入输出变量与逻辑运算符的组合赋值语句。使用了赋值语句以外的命令语句时将出错，无法完成编译。

4.1 赋值语句

格式

< 输出变量 > < 内部输出变量 > < 机械臂锁定输出变量 > < 定时器输出变量 >	= < 表达式 >
--	-----------

值 < 表达式 > 可使用以下任何一种。

- 并行输入输出变量
- 内部输出变量
- 机械臂锁定输出变量
- 定时器输出变量
- 串行输入输出变量
- 上述逻辑运算公式

4.2 输入输出变量

各变量只可使用 1 指定比特的形式。

· 正确示例	DO (35) MO (24) DI (16)
· 错误示例	DO (37, 24) DI3 (4) MO3 ()

● 并行输入变量

格式

DI (mb)	m : 端口编号 0 ~ 7, 10 ~ 17, 20 ~ 27 b : 指定比特 0 ~ 7
---------	--

表示并行输入信号的状态。

● 并行输出变量

格式

DO (mb)	m : 端口编号 0 ~ 7, 10 ~ 17, 20 ~ 27 b : 指定比特 0 ~ 7
---------	--

指定并行输出或者参照输出状态。端口 0, 1 只可参照，无法输出。

- 内部输出变量

格式		
MO (mb)	m : 端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
	b : 指定比特	0 ~ 7

无法输出至外部的信号，在控制器内部使用。端口 0, 1 只可参照，无法输出。

- 机械臂锁定输出变量

格式		
LO (mb)	m : 端口编号	仅为 0
	b : 指定比特	0 ~ 7

用于禁止机械臂的移动。ON 时，禁止移动。LO (00) ~ LO (07) 分别对应第 1 机械臂至第 8 机械臂。

- 定时器输出变量

格式		
TO (mb)	m : 端口编号	仅为 0
	b : 指定比特	0 ~ 7

TO (00) ~ TO (07) 合计 8 个端口，各变量的定时器通过定时器定义语句 TIM00 ~ 07 进行定义。

- 串行输入变量

格式		
SI (mb)	m : 端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
	b : 指定比特	0 ~ 7

表示串行输入信号的状态。只可参照，无法指定。

- 串行输出变量

格式		
SO (mb)	m : 端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
	b : 指定比特	0 ~ 7

指定或参照串行输出信号的状态。端口 0, 1 只可参照，无法输出。

4.5 运算的优先级

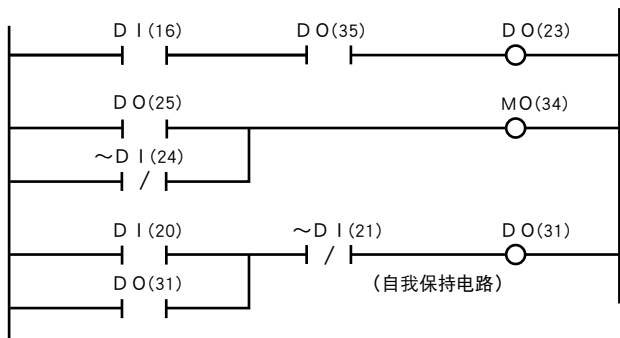
优先级	运算内容
1	() 中的表达式
2	NOT, ~ (否定)
3	AND, & (逻辑积)
4	OR, (逻辑和)

● 替换梯形语句的示例

SAMPLE

```
DO (23) = DI (16) & DO (35)
MO (34) = DO (25) | ~DI (24)
DO (31) = (DI (20) | DO (31)) & ~DI (21)
```

梯形图



35702-R7-00

MEMO

- 运算符的否定无法用在 "(" 的近前及赋值语句的左边。例如，无法用于以下语句。
 - DO (21) = ~(DI (30) | DI (32))
 - ~DO (30) = DI (22) & DI (27)
- 无法在赋值语句的右边指定数值。
 - MO (35) = 1
 - DO (26) = 0
- 无需在程序的最后定义 "HALT" 或 "HOLD"。
- 次序程序中使用的输入输出、内部输出变量的各变量与机器人程序通用，因此请在使用相同变量时特别注意。

4.6 次序程序规格

项目	内容
命令	否定 (NOT)、逻辑积 (AND)、逻辑和 (OR)
输入输出	与机器人语言相同
程序容量	4,096 字节 (最多可指定 512 个变量)
扫描时间	根据步级数，最小从 10msec 至最大 30msec 的范围内自动进行变化

第 8 章

机器人语言一览表

机器人语言一览表的查看方法	8-1
按字母顺序	8-3
按功能	8-8
函数：按字母顺序	8-15
函数：按功能	8-18
1 ABS	8-20
2 ABSRPOS / ABSRPOS2	8-21
3 ABSRST	8-22
4 ACCEL / ACCEL2	8-23
5 ARCH / ARCH2	8-24
6 ARMCND / ARMCND2	8-26
7 ARMTYPE / ARMTYPE2	8-27
8 ATN	8-28
9 ASPEED / ASPEED2	8-29
10 AXWGHT / AXWGHT2	8-30
11 CALL	8-31
12 CHANGE / CHANGE2	8-32
13 CHGPRI	8-33
14 CHR\$	8-34

15	COS.....	8-35
16	CUT.....	8-36
17	DATE\$.....	8-37
18	DECEL / DECEL2.....	8-38
19	DECLARE.....	8-39
20	DEF FN.....	8-41
21	DEGRAD.....	8-42
22	DELAY.....	8-43
23	DI.....	8-44
24	DIST.....	8-45
25	DIM.....	8-46
26	DO.....	8-47
27	DRIVE / DRIVE2.....	8-48
28	DRIVEI / DRIVEI2.....	8-54
29	END SELECT.....	8-58
30	END SUB.....	8-59
31	ERR / ERL.....	8-60
32	EXIT FOR.....	8-61
33	EXIT SUB.....	8-62
34	EXIT TASK.....	8-63
35	FOR ~ NEXT.....	8-64
36	GOSUB ~ RETURN.....	8-65
37	GOTO.....	8-66
38	HALT.....	8-67
39	HAND / HAND2.....	8-68
40	HOLD.....	8-73
41	IF.....	8-74
42	INPUT.....	8-76
43	INT.....	8-77
44	JTOXY / JTOXY2.....	8-78
45	LABEL 语句.....	8-79
46	LEFT\$.....	8-80
47	LEFTY / LEFTY2.....	8-81
48	LEN.....	8-82
49	LET.....	8-83
50	LO.....	8-86

51	LOCx.....	8-87
52	LSHIFT.....	8-88
53	MCHREF / MCHREF2	8-89
54	MID\$	8-90
55	MO.....	8-91
56	MOVE / MOVE2.....	8-92
57	MOVEI / MOVEI2	8-107
58	OFFLINE	8-111
59	ORD	8-112
60	ON ERROR GOTO.....	8-113
61	ON ~ GOSUB.....	8-114
62	ON ~ GOTO	8-115
63	ONLINE	8-116
64	ORGORD / ORGORD2.....	8-117
65	ORIGIN.....	8-118
66	OUT.....	8-119
67	OUTPOS / OUTPOS2.....	8-120
68	PATH.....	8-122
69	PATH END	8-128
70	PATH SET	8-129
71	PATH START.....	8-131
72	PDEF.....	8-132
73	PMOVE / PMOVE2	8-133
74	Pn.....	8-137
75	PPNT	8-138
76	PRINT	8-139
77	RADDEG	8-140
78	REM	8-141
79	RESET	8-142
80	RESTART.....	8-143
81	RESUME.....	8-144
82	RETURN.....	8-145
83	RIGHT\$.....	8-146
84	RIGHTY / RIGHTY2.....	8-147
85	RSHIFT	8-148

86	Sn.....	8-149
87	SELECT CASE.....	8-150
88	SEND	8-151
89	SERVO / SERVO2	8-153
90	SET.....	8-154
91	SHARED	8-155
92	SHIFT / SHIFT2	8-156
93	SIN	8-157
94	SO.....	8-158
95	SPEED / SPEED2	8-159
96	START	8-160
97	STR\$.....	8-161
98	SQR.....	8-162
99	SUB ~ END SUB	8-163
100	SUSPEND.....	8-165
101	SWI	8-166
102	TAN.....	8-167
103	TCOUNTER.....	8-168
104	TIME\$.....	8-169
105	TIMER.....	8-170
106	TO.....	8-171
107	TOLE / TOLE2	8-172
108	TORQUE / TORQUE2	8-173
109	TRQSTS / TRQSTS2	8-175
110	TRQTIME / TRQTIME2	8-176
111	VAL.....	8-178
112	WAIT	8-179
113	WAIT ARM / WAIT ARM2.....	8-180
114	WEIGHT / WEIGHT2	8-181
115	WEND.....	8-182
116	WHERE / WHERE2	8-183
117	WHILE ~ WEND	8-184
118	WHRXY / WHRXY2	8-185
119	XYTOJ / XYTOJ2.....	8-186
120	_SYSFLG.....	8-186

以下为您介绍机器人语言一览表的查看方法。

	(1) 	(2) 	(3) 	(4) 	(5)
	No.	功能	条件	直接	种类
DIM	25	对数组变量的名称及元素数进行声明	6	×	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) 定义整型数组变量 A% (0) ~ A% (10) (元素数11)
DIM B (2, 3, 4) 定义实型数组变量 B (0, 0, 0) ~ B (2, 3, 4) (元素数 60)
DIM C% (2, 2), D! (10) 定义整型数组 C% (0, 0) ~ C% (2, 2) 与实型数组 D! (0) ~ D! (10)

(2) 功能

介绍机器人语言的功能。

(3) 条件

表示可执行的条件。

- 条件 1：直接命令与联机命令中也可执行的命令。
- 条件 2：条件 1 之外再加上，仅任务 1（主任务）可执行的命令。
- 条件 3：条件 1 之外再加上，在直接命令与联机命令中部分无法执行操作数的某种命令。
- 条件 4：条件 1 之外再加上，定位完成后执行的命令。
- 条件 5：MOVE L、MOVE C 可在直接命令与联机命令中执行。
但是，要在定位完成后执行。
此外，STOPON 选项无法在直接命令与联机命令中执行。
- 条件 6：无法在直接命令与联机命令中执行的命令。

机器人语言中，有关命令与函数均可使用的语言，可能在命令与函数中可执行的条件有所不同。

在此情况下，请用 / 间隔表示命令与函数各自的条件。

例如，作为命令使用时条件为 4，并且作为函数使用时无条件时，表示为“4/-”。

(4) 直接

此项为“○”时，可在直接命令及联机命令中使用。



- 直接命令，在自动模式中通过手持编程器直接输入命令使机器人临时动作。详细说明，请参阅控制器的手册。

(5) 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”。

No.	命令	功能	条件	直接	种类
A					
1	ABS	计算指定值的绝对值	-	-	函数
2	ABSRPOS	计算主组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
2	ABSRPOS2	计算副组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
3	ABSRST	执行机器人的绝对马达轴返回原点动作	4	○	命令
4	ACCEL	设置 / 获取主组的加速度系数参数	4/-	○	命令 / 函数
4	ACCEL2	设置 / 获取副组的加速度系数参数	4/-	○	命令 / 函数
5	ARCH	设置 / 获取主组的圆弧位置参数	4/-	○	命令 / 函数
5	ARCH2	设置 / 获取副组的圆弧位置参数	4/-	○	命令 / 函数
6	ARMCND	获取主机器人当前机械臂的状态	-	-	函数
6	ARMCND2	获取副机器人当前机械臂的状态	-	-	函数
7	ARMTYPE	获取主机器人当前手系统的设置	-	-	函数
7	ARMTYPE2	获取副机器人当前手系统的设置	-	-	函数
9	ASPEED	更改主组的自动移动速度	4	○	命令
9	ASPEED2	更改副组的自动移动速度	4	○	命令
8	ATN	计算指定值的反正切值	-	-	函数
10	AXWGHT	设置 / 获取主组的轴前端重量参数	4/-	○	命令 / 函数
10	AXWGHT2	设置 / 获取副组的轴前端重量参数	4/-	○	命令 / 函数
C					
11	CALL	执行其他程序	6	×	命令
12	CHANGE	进行主机器人的机械手切换	4	○	命令
12	CHANGE2	进行副机器人的机械手切换	4	○	命令
13	CHGPRI	更改指定任务的优先级	6	×	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	CUT	强制结束执行中或暂停中的其他任务	6	×	命令
D					
17	DATES\$	通过 "yy/mm/dd" 形式的字符串计算日期	-	-	函数
18	DECEL	设置 / 获取主组的减速比率参数	4/-	○	命令 / 函数
18	DECEL2	设置 / 获取副组的减速比率参数	4/-	○	命令 / 函数
21	DEGRAD	将值转换为弧度（↔ RADDEG）	-	-	函数
22	DELAY	仅等待指定的时间（单位 ms）	6	×	命令
25	DIM	对数组变量的名称及元素数进行声明	6	×	命令
24	DIST	计算指定 2 个点之间的距离	-	-	函数
26	DO	将指定值输出至 DO 端口	1	○	命令
27	DRIVE	以轴为单位对主组进行绝对位置移动	4	○	命令
27	DRIVE	（带有 T 选项时）执行轴单位的绝对移动命令	4	○	命令
27	DRIVE2	以轴为单位对副组进行绝对位置移动	4	○	命令
28	DRIVEI	以轴为单位对主组进行相对位置移动	4	○	命令
28	DRIVEI2	以轴为单位对副组进行相对位置移动	4	○	命令

No.	命令	功能	条件	直接	种类
E					
31	ERL	赋予发生错误的行编号	-	-	函数
31	ERR	赋予发生错误时的错误代码编号	-	-	函数
32	EXIT FOR	强制结束 FOR 语句～NEXT 语句的循环	6	×	命令
34	EXIT TASK	结束当前正在执行的任务	6	×	命令
F					
35	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	6	×	命令
G					
36	GOSUB ～ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	6	×	命令
37	GOTO	无条件跳转至标签所指定的行	6	×	命令
H					
38	HALT	停止程序并进行复位	6	×	命令
39	HAND	定义主机器人的机械手	4	○	命令
39	HAND2	定义副机器人的机械手	4	○	命令
40	HOLD	暂停程序	6	×	命令
I					
41	IF	根据条件分支控制流程	6	×	命令
42	INPUT	从手持编程器中对指定的变量赋值	1	○	命令
43	INT	获得舍去小数点以下部分的整数值	-	-	函数
J					
44	JTOXY	将关节坐标数据转换为主组的正交坐标数据 (↔ XYTOJ)	-	-	函数
44	JTOXY2	将关节坐标数据转换为副组的正交坐标数据 (↔ XYTOJ2)	-	-	函数
L					
46	LEFT\$	从指定的字符串左侧抽出指定位数的字符串	-	-	函数
47	LEFTY	将主机器人的手系统设置为左手系统	4	○	命令
47	LEFTY2	将副机器人的手系统设置为左手系统	4	○	命令
48	LEN	获得指定字符串的长度(字节数)	-	-	函数
49	LET	执行指定的赋值语句	1	○	命令
50	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	1	○	命令
51	LOCx	以轴为单位设置 / 获取坐标点数据或位移数据	-	-	命令 / 函数
52	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	-	函数
M					
53	MCHREF	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
53	MCHREF2	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
54	MID\$	从指定的字符串中抽出任意长度的字符串	-	-	函数
55	MO	将指定值输出至 MO 端口	1	○	命令
56	MOVE	对主组机器人的所有轴进行绝对移动	5	○	命令
56	MOVE2	对副组机器人的所有轴进行绝对移动	5	○	命令
57	MOVEI	对主组机器人的所有轴进行相对移动	4	○	命令
57	MOVEI2	对副组机器人的所有轴进行相对移动	4	○	命令

No.	命令	功能	条件	直接	种类
O					
58	OFFLINE	将指定的通信端口设置为脱机模式	1	○	命令
60	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	6	×	命令
61	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	6	×	命令
62	ON ~ GOTO	根据条件跳转至标签所指定的各行	6	×	命令
63	ONLINE	将指定的通信端口设置为联机模式	1	○	命令
59	ORD	获得指定字符串起始字符的字符编码	-	-	函数
64	ORGORD	设置 / 获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
64	ORGORD2	设置 / 获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
65	ORIGIN	对增量式规格的轴执行返回原点动作	4	○	命令
66	OUT	开启指定输出端口的比特，并结束命令语句	6	×	命令
67	OUTPOS	设置 / 获取主组的 OUT 有效位置参数	4/-	○	命令 / 函数
67	OUTPOS2	设置 / 获取副组的 OUT 有效位置参数	4/-	○	命令 / 函数
P					
68	PATH	设置 PATH 移动路径	6	×	命令
69	PATH END	结束 PATH 移动路径的设置	6	×	命令
70	PATH SET	开始 PATH 移动路径的设置	6	×	命令
71	PATH START	开始 PATH 移动	6	×	命令
72	PDEF	定义用于执行托盘移动命令的托盘	1	○	命令
73	PMOVE	执行主机器人的托盘移动命令	4	○	命令
73	PMOVE2	执行副机器人的托盘移动命令	4	○	命令
74	Pn	在程序中定义坐标点	1	○	命令
75	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
76	PRINT	在手持编程器的画面上显示字符串。	1	○	命令
R					
77	RADDEG	将值转换为度 (↔ DEGRAD)	-	-	函数
78	REM	记述注释语句	6	×	命令
79	RESET	关闭指定输出端口的比特	1	○	命令
80	RESTART	重新启动暂停中的其他任务	6	×	命令
81	RESUME	进行错误的恢复处理后，恢复执行程序	6	×	命令
83	RIGHT\$	从指定的字符串右侧抽出指定位数的字符串	-	-	函数
84	RIGHTY	将主机器人的手系统设置为右手系统	4	○	命令
84	RIGHTY2	将副机器人的手系统设置为右手系统	4	○	命令
85	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	-	函数
S					
86	Sn	在程序中定义位移坐标	4	○	命令
87	SELECT CASE ~ END SELECT	根据条件分支控制流程	6	×	命令
88	SEND	转送文件	1	○	命令
89	SERVO	控制主组指定轴或所有轴的伺服 ON / OFF	4	○	命令
89	SERVO2	控制副组指定轴或所有轴的伺服 ON / OFF	4	○	命令
90	SET	开启指定输出端口的比特	3	部分 ×	命令

No.	命令	功能	条件	直接	种类
92	SHIFT	指定位移变量，并在该处指定的位移数据上设置主机机器人的位移坐标	4	○	命令
92	SHIFT2	指定位移变量，并在该处指定的位移数据上设置副机器人的位移坐标	4	○	命令
93	SIN	计算指定值的正弦值	-	-	函数
94	SO	将指定值输出至 SO 端口	1	○	命令
95	SPEED	更改主组的程序移动速度	4	○	命令
95	SPEED2	更改副组的程序移动速度	4	○	命令
98	SQR	计算指定值的平方根	-	-	函数
96	START	设置指定任务的任务编号及优先级，并启动该任务	6	×	命令
97	STR\$	将指定值转换为字符串 (↔ VAL)	-	-	函数
100	SUSPEND	暂停执行中的其他任务	6	×	命令
101	SWI	切换执行程序，并在执行编译后从第 1 行开始执行	2	○	命令
T					
102	TAN	计算指定值的正切值	-	-	函数
103	TCOUNTER	从 TCOUNTER 变量复位时开始，每隔 10ms 输出计数递增值	-	-	函数
104	TIME\$	通过 "hh : mm : ss" 形式的字符串计算当前时间	-	-	函数
105	TIMER	用上午 0 点开始的秒计算当前时间	-	-	函数
106	TO	将指定值输出至 TO 端口	1	○	命令
107	TOLE	设置 / 获取主组的公差参数	4/-	○	命令 / 函数
107	TOLE2	设置 / 获取副组的公差参数	4/-	○	命令 / 函数
108	TORQUE	设置 / 获取主组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
108	TORQUE2	设置 / 获取副组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
109	TRQSTS	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
109	TRQSTS2	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
110	TRQTIME	设置 / 获取主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数
110	TRQTIME2	设置 / 获取副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数
V					
111	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)	-	-	函数
W					
112	WAIT	等待直至 DI / DO 条件表达式成立为止 (带超时)	6	×	命令
113	WAIT ARM	等待主组的机器人轴动作结束	6	×	命令
113	WAIT ARM2	等待副组的机器人轴动作结束	6	×	命令
114	WEIGHT	设置 / 获取主机器人的前端重量参数	4/-	○	命令 / 函数
114	WEIGHT2	设置 / 获取副机器人的前端重量参数	4/-	○	命令 / 函数
116	WHERE	用关节坐标 (脉冲) 读出主组机械臂的当前位置	-	-	函数
116	WHERE2	用关节坐标 (脉冲) 读出副组机械臂的当前位置	-	-	函数
117	WHILE ~ WEND	控制反复	6	×	命令
118	WHRXY	用正交坐标 (mm、度) 读出主组机械臂的当前位置	-	-	函数
118	WHRXY2	用正交坐标 (mm、度) 读出副组机械臂的当前位置	-	-	函数

No.	命令	功能	条件	直接	种类
X					
119	XYTOJ	将坐标点变量的正交坐标数据转换为主组的关节坐标数据 (↔ JTOXY)	-	-	函数
119	XYTOJ2	将坐标点变量的正交坐标数据转换为副组的关节坐标数据 (↔ JTOXY2)	-	-	函数
120	_SYSFLG	监视轴状态的标志	-	-	函数

按功能

程序命令

普通命令

No.	命令	功能	条件	直接	种类
25	DIM	对数组变量的名称及元素数进行声明	6	×	命令
49	LET	执行指定的赋值语句	1	○	命令
78	REM	记述注释语句	6	×	命令

算术命令

No.	命令	功能	条件	直接	种类
1	ABS	计算指定值的绝对值	-	-	函数
8	ATN	计算指定值的反正切值	-	-	函数
15	COS	计算指定值的余弦值	-	-	函数
21	DEGRAD	将值转换为弧度 (↔ RADDEG)	-	-	函数
24	DIST	计算指定 2 个点之间的距离	-	-	函数
43	INT	获得舍去小数点以下部分的整数值	-	-	函数
52	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	-	函数
77	RADDEG	将值转换为度 (↔ DEGRAD)	-	-	函数
85	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	-	函数
93	SIN	计算指定值的正弦值	-	-	函数
98	SQR	计算指定值的平方根	-	-	函数
102	TAN	计算指定值的正切值	-	-	函数

日期、时间

No.	命令	功能	条件	直接	种类
17	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期	-	-	函数
103	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数递增值	-	-	函数
104	TIME \$	通过 "hh:mm:ss" 形式的字符串计算当前时间	-	-	函数
105	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	条件	直接	种类
14	CHR \$	获取带有指定字符编码的字符	-	-	函数
46	LEFT \$	从指定的字符串左侧抽出指定位数的字符串	-	-	函数
48	LEN	获得指定字符串的长度(字节数)	-	-	函数
54	MID \$	从指定的字符串中抽出任意长度的字符串	-	-	函数
59	ORD	获得指定字符串起始字符的字符编码	-	-	函数
83	RIGHT \$	从指定的字符串右侧抽出指定位数的字符串	-	-	函数
97	STR \$	将指定值转换为字符串(↔ VAL)	-	-	函数
111	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	条件	直接	种类
12	CHANGE	进行主机器人的机械手切换	4	○	命令
12	CHANGE2	进行副机器人的机械手切换	4	○	命令
39	HAND	定义主机器人的机械手	4	○	命令
39	HAND2	定义副机器人的机械手	4	○	命令
44	JTOXY	将关节坐标数据转换为主组的正交坐标数据(↔ XYTOJ)	-	-	函数
44	JTOXY2	将关节坐标数据转换为副组的正交坐标数据(↔ XYTOJ2)	-	-	函数
47	LEFTY	将主机器人的手系统设置为左手系统	4	○	命令
47	LEFTY2	将副机器人的手系统设置为左手系统	4	○	命令
74	Pn	在程序中定义坐标点	1	○	命令
75	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
84	RIGHTY	将主机器人的手系统设置为右手系统	4	○	命令
84	RIGHTY2	将副机器人的手系统设置为右手系统	4	○	命令
86	Sn	在程序中定义位移坐标	4	○	命令
92	SHIFT	指定位移变量,并在该处指定的位移数据上设置主机器人的位移坐标	4	○	命令
92	SHIFT2	指定位移变量,并在该处指定的位移数据上设置副机器人的位移坐标	4	○	命令
119	XYTOJ	将坐标点变量的正交坐标数据转换为主组的关节坐标数据(↔ JTOXY)	-	-	函数
119	XYTOJ2	将坐标点变量的正交坐标数据转换为副组的关节坐标数据(↔ JTOXY2)	-	-	函数
51	LOCx	以轴为单位设置/获取坐标点数据或位移数据	-	-	命令/函数

分支命令

No.	命令	功能	条件	直接	种类
32	EXIT FOR	强制结束 FOR 语句～NEXT 语句的循环	6	×	命令
35	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	6	×	命令
36	GOSUB ～ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	6	×	命令
37	GOTO	无条件跳转至标签所指定的行	6	×	命令
41	IF	根据条件分支控制流程	6	×	命令
61	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	6	×	命令
62	ON ～ GOTO	根据条件跳转至标签所指定的各行	6	×	命令
87	SELECT CASE ～ END SELECT	根据条件分支控制流程	6	×	命令
117	WHILE ～ WEND	控制反复	6	×	命令

错误控制

No.	命令	功能	条件	直接	种类
60	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	6	×	命令
81	RESUME	进行错误的恢复处理后，恢复执行程序	6	×	命令
31	ERL	赋予发生错误的行编号	-	-	函数
31	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	条件	直接	种类
11	CALL	执行其他程序	6	×	命令
38	HALT	停止程序并进行复位	6	×	命令
40	HOLD	暂停程序	6	×	命令
101	SWI	切换执行程序，并在执行编译后从第 1 行开始执行	2	○	命令

任务控制

No.	命令	功能	条件	直接	种类
13	CHGPRI	更改指定任务的优先级	6	×	命令
16	CUT	强制结束执行中或暂停中的其他任务	6	×	命令
34	EXIT TASK	结束当前正在执行的任务	6	×	命令
80	RESTART	重新启动暂停中的其他任务	6	×	命令
96	START	设置指定任务的任务编号及优先级，并启动该任务	6	×	命令
100	SUSPEND	暂停执行中的其他任务	6	×	命令

No.	命令	功能	条件	直接	种类
3	ABSRST	执行机器人的绝对式马达轴返回原点动作	4	○	命令
12	CHANGE	进行主机器人的机械手切换	4	○	命令
12	CHANGE2	进行副机器人的机械手切换	4	○	命令
27	DRIVE	以轴为单位对主组进行绝对位置移动	4	○	命令
27	DRIVE2	以轴为单位对副组进行绝对位置移动	4	○	命令
28	DRIVEI	以轴为单位对主组进行相对位置移动	4	○	命令
28	DRIVEI2	以轴为单位对副组进行相对位置移动	4	○	命令
39	HAND	定义主机器人的机械手	4	○	命令
39	HAND2	定义副机器人的机械手	4	○	命令
47	LEFTY	将主机器人的手系统设置为左手系统	4	○	命令
47	LEFTY2	将副机器人的手系统设置为左手系统	4	○	命令
56	MOVE	对主组机器人的所有轴进行绝对移动	5	○	命令
56	MOVE2	对副组机器人的所有轴进行绝对移动	5	○	命令
57	MOVEI	对主组机器人的所有轴进行相对移动	4	○	命令
57	MOVEI2	对副组机器人的所有轴进行相对移动	4	○	命令
65	ORIGIN	对增量式规格的轴执行返回原点动作	4	○	命令
73	PMOVE	执行主机器人的托盘移动命令	4	○	命令
73	PMOVE2	执行副机器人的托盘移动命令	4	○	命令
84	RIGHTY	将主机器人的手系统设置为右手系统	4	○	命令
84	RIGHTY2	将副机器人的手系统设置为右手系统	4	○	命令
89	SERVO	控制主组指定轴或所有轴的伺服 ON / OFF	4	○	命令
89	SERVO2	控制副组指定轴或所有轴的伺服 ON / OFF	4	○	命令

状态获取

No.	命令	功能	条件	直接	种类
2	ABSRPOS	计算主组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
2	ABSRPOS2	计算副组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
6	ARMCND	获取主机器人当前机械臂的状态	-	-	函数
6	ARMCND2	获取副机器人当前机械臂的状态	-	-	函数
7	ARMTYPE	获取主机器人当前手系统的设置	-	-	函数
7	ARMTYPE2	获取副机器人当前手系统的设置	-	-	函数
53	MCHREF	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
53	MCHREF2	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
109	TRQSTS	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
109	TRQSTS2	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
116	WHERE	用关节坐标（脉冲）读出主组机械臂的当前位置	-	-	函数
116	WHERE2	用关节坐标（脉冲）读出副组机械臂的当前位置	-	-	函数

No.	命令	功能	条件	直接	种类
118	WHRXY	用正交坐标 (mm、度) 读出主组机械臂的当前位置	-	-	函数
118	WHRXY2	用正交坐标 (mm、度) 读出副组机械臂的当前位置	-	-	函数
112	WAIT	等待机器人轴动作结束	6	×	命令

状态更改

No.	命令	功能	条件	直接	种类
4	ACCEL	设置 / 获取主组的加速度系数参数	4/-	○	命令 / 函数
4	ACCEL2	设置 / 获取副组的加速度系数参数	4/-	○	命令 / 函数
5	ARCH	设置 / 获取主组的圆弧位置参数	4/-	○	命令 / 函数
5	ARCH2	设置 / 获取副组的圆弧位置参数	4/-	○	命令 / 函数
9	ASPEED	更改主组的自动移动速度	4	○	命令
9	ASPEED2	更改副组的自动移动速度	4	○	命令
10	AXWGHT	设置 / 获取主组的轴前端重量参数	4/-	○	命令 / 函数
10	AXWGHT2	设置 / 获取副组的轴前端重量参数	4/-	○	命令 / 函数
18	DECEL	设置 / 获取主组的减速比率参数	4/-	○	命令 / 函数
18	DECEL2	设置 / 获取副组的减速比率参数	4/-	○	命令 / 函数
64	ORGORD	设置 / 获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
64	ORGORD2	设置 / 获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
67	OUTPOS	设置 / 获取主组的 OUT 有效位置参数	4/-	○	命令 / 函数
67	OUTPOS2	设置 / 获取副组的 OUT 有效位置参数	4/-	○	命令 / 函数
72	PDEF	定义用于执行托盘移动命令的托盘	1	○	命令
95	SPEED	更改主组的程序移动速度	4	○	命令
95	SPEED2	更改副组的程序移动速度	4	○	命令
107	TOLE	设置 / 获取主组的公差参数	4/-	○	命令 / 函数
107	TOLE2	设置 / 获取副组的公差参数	4/-	○	命令 / 函数
114	WEIGHT	设置 / 获取主机器人的前端重量参数	4/-	○	命令 / 函数
114	WEIGHT2	设置 / 获取副机器人的前端重量参数	4/-	○	命令 / 函数

PATH 控制

No.	命令	功能	条件	直接	种类
68	PATH	设置 PATH 移动路径	6	×	命令
69	PATH END	结束 PATH 移动路径的设置	6	×	命令
70	PATH SET	开始 PATH 移动路径的设置	6	×	命令
71	PATH START	开始 PATH 移动	6	×	命令

扭矩控制

No.	命令	功能	条件	直接	种类
27	DRIVE	(带有 T 选项时) 执行轴单位的绝对移动命令	4	○	命令
108	TORQUE	设置 / 获取主组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
108	TORQUE2	设置 / 获取副组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
110	TRQTIME	设置 / 获取主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数
110	TRQTIME2	设置 / 获取副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数

输入输出、通信控制

输入输出控制

No.	命令	功能	条件	直接	种类
22	DELAY	仅等待指定的时间(单位 ms)	6	×	命令
26	DO	将指定值输出至 DO 端口	1	○	命令
50	LO	将指定值输出至 LO 端口, 并进行轴移动的禁止或解除	1	○	命令
55	MO	将指定值输出至 MO 端口	1	○	命令
66	OUT	开启指定输出端口的比特, 并结束命令语句	6	×	命令
79	RESET	关闭指定输出端口的比特	1	○	命令
90	SET	开启指定输出端口的比特	3	部分 ×	命令
94	SO	将指定值输出至 SO 端口	1	○	命令
106	TO	将指定值输出至 TO 端口	1	○	命令
112	WAIT	等待直至 DI / DO 条件表达式成立为止(带超时)	6	×	命令

手持编程器

No.	命令	功能	条件	直接	种类
42	INPUT	从手持编程器中对指定的变量赋值	1	○	命令
76	PRINT	在手持编程器的画面上显示字符串	1	○	命令

通信控制

No.	命令	功能	条件	直接	种类
63	ONLINE	将指定的通信端口设置为联机模式	1	○	命令
58	OFFLINE	将指定的通信端口设置为脱机模式	1	○	命令
88	SEND	转送文件	1	○	命令

其他

其他

No.	命令	功能	条件	直接	种类
120	_SYSFLG	监视轴状态的标志	-	-	函数

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	计算主组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）
2	ABSRPOS2	算术函数	计算副组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）
4	ACCEL	算术函数	获取主组的加速度系数参数
4	ACCEL2	算术函数	获取副组的加速度系数参数
5	ARCH	算术函数	获取主组的圆弧位置参数
5	ARCH2	算术函数	获取副组的圆弧位置参数
6	ARMCND	算术函数	获取主机器人当前机械臂的状态
6	ARMCND2	算术函数	获取副机器人当前机械臂的状态
7	ARMTYPE	算术函数	获取主机器人当前手系统的设置
7	ARMTYPE2	算术函数	获取副机器人当前手系统的设置
8	ATN	算术函数	计算指定值的反正切值
10	AXWGHT	算术函数	获取主组的轴前端重量参数
10	AXWGHT2	算术函数	获取副组的轴前端重量参数
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
D			
17	DATE\$	字符串函数	通过 "yy/mm/dd" 形式的字符串计算日期
18	DECEL	算术函数	获取主组的减速比率参数
18	DECEL2	算术函数	获取副组的减速比率参数
21	DEGRAD	算术函数	将值转换为弧度（↔ RADDEG）
24	DIST	算术函数	计算指定 2 个点之间的距离
E			
31	ERL	算术函数	发生错误的行编号
31	ERR	算术函数	赋予发生错误时的错误代码编号
I			
43	INT	算术函数	获得舍去小数点以下部分的整数
J			
44	JTOXY	坐标点函数	将关节坐标数据转换为主组的正交坐标数据（↔ XYTOJ）
44	JTOXY2	坐标点函数	将关节坐标数据转换为副组的正交坐标数据（↔ XYTOJ2）
L			
46	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
48	LEN	算术函数	获得指定字符串的长度（字节数）
51	LOCx	坐标点函数	以轴为单位获取坐标点数据或位移数据
52	LSHIFT	算术函数	仅将指定值的比特数向左偏移（↔ RSHIFT）

No.	函数	种类	功能
M			
53	MCHREF	算术函数	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照
53	MCHREF2	算术函数	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照
54	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			
59	ORD	算术函数	获得指定字符串起始字符的字符编码
64	ORGORD	算术函数	获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数
64	ORGORD2	算术函数	获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数
67	OUTPOS	算术函数	获取主组的 OUT 有效位置参数
67	OUTPOS2	算术函数	获取副组的 OUT 有效位置参数
P			
75	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
R			
77	RADDEG	算术函数	将值转换为度 (↔ DEGRAD)
83	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
85	RSHIFT	算术函数	仅将指定值的比特数向右偏移 (↔ LSHIFT)
S			
93	SIN	算术函数	计算指定值的正弦值
98	SQR	算术函数	计算指定值的平方根
97	STR\$	字符串函数	将指定值转换为字符串 (↔ VAL)
T			
102	TAN	算术函数	计算指定值的正切值
103	TCOUNTER	算术函数	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数递增值
104	TIME\$	字符串函数	通过 "hh:mm:ss" 形式的字符串计算当前时间
105	TIMER	算术函数	用上午 0 点开始的秒计算当前时间
107	TOLE	算术函数	获取主组的公差参数
107	TOLE2	算术函数	获取副组的公差参数
108	TORQUE	算术函数	获取主组指定轴的最大扭矩指令值
108	TORQUE2	算术函数	获取副组指定轴的最大扭矩指令值
109	TRQSTS	算术函数	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态
109	TRQSTS2	算术函数	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态
110	TRQTIME	算术函数	获取主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间
110	TRQTIME2	算术函数	获取副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间
V			
111	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			
114	WEIGHT	算术函数	获取主机器人的前端重量参数
114	WEIGHT2	算术函数	获取副机器人的前端重量参数
116	WHERE	坐标点函数	用关节坐标 (脉冲) 读出主组机械臂的当前位置
116	WHERE2	坐标点函数	用关节坐标 (脉冲) 读出副组机械臂的当前位置
118	WHRXY	坐标点函数	用正交坐标 (mm、度) 读出主组机械臂的当前位置
118	WHRXY2	坐标点函数	用正交坐标 (mm、度) 读出副组机械臂的当前位置

No.	函数	种类	功能
X			
119	XYTOJ	坐标点函数	将坐标点变量的正交坐标数据转换为主组的关节坐标数据 (↔ JTOXY)
119	XYTOJ2	坐标点函数	将坐标点变量的正交坐标数据转换为副组的关节坐标数据 (↔ JTOXY2)
120	_SYSFLG	算术函数	监视轴状态的标志

函数：按功能

相关坐标点

No.	函数名	功能
44	JTOXY	将关节坐标数据转换为主组的正交坐标数据 (↔ XYTOJ)
44	JTOXY2	将关节坐标数据转换为副组的正交坐标数据 (↔ XYTOJ2)
51	LOCx	以轴为单位获取坐标点数据或位移数据
75	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
116	WHERE	用关节坐标 (脉冲) 读出主组机械臂的当前位置
116	WHERE2	用关节坐标 (脉冲) 读出副组机械臂的当前位置
118	WHRXY	用正交坐标 (mm、度) 读出主组机械臂的当前位置
118	WHRXY2	用正交坐标 (mm、度) 读出副组机械臂的当前位置
119	XYTOJ	将坐标点变量的正交坐标数据转换为主组的关节坐标数据 (↔ JTOXY)
119	XYTOJ2	将坐标点变量的正交坐标数据转换为副组的关节坐标数据 (↔ JTOXY2)

相关参数

No.	函数名	功能
2	ABSRPOS	计算主组指定轴的机器参照值 (返回原点的方式仅为标记方式时有效)
2	ABSRPOS2	计算副组指定轴的机器参照值 (返回原点的方式仅为标记方式时有效)
4	ACCEL	获取主组的加速度系数参数
4	ACCEL2	获取副组的加速度系数参数
5	ARCH	获取主组的圆弧位置参数
5	ARCH2	获取副组的圆弧位置参数
6	ARMCND	获取主机器人当前机械臂的状态
6	ARMCND2	获取副机器人当前机械臂的状态
7	ARMTYPE	获取主机器人当前手系统的设置
7	ARMTYPE2	获取副机器人当前手系统的设置
10	AXWGHT	获取主组的轴前端重量参数
10	AXWGHT2	获取副组的轴前端重量参数
18	DECEL	获取主组的减速比率参数
18	DECEL2	获取副组的减速比率参数
48	LEN	获得指定字符串的长度 (字节数)
53	MCHREF	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照
53	MCHREF2	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照
59	ORD	获得指定字符串起始字符的字符编码
64	ORGORD	获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数
64	ORGORD2	获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数
67	OUTPOS	获取主组的 OUT 有效位置参数
67	OUTPOS2	获取副组的 OUT 有效位置参数
107	TOLE	获取主组的公差参数
107	TOLE2	获取副组的公差参数
108	TORQUE	获取主组指定轴的最大扭矩指令值
108	TORQUE2	获取副组指定轴的最大扭矩指令值
109	TRQSTS	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态
109	TRQSTS2	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态

No.	函数名	功能
110	TRQTIME	获取主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间
110	TRQTIME2	获取副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间
114	WEIGHT	获取主机器人的前端重量参数
114	WEIGHT2	获取副机器人的前端重量参数

相关数值运算

No.	函数名	功能
1	ABS	计算指定值的绝对值
8	ATN	计算指定值的反正切值
15	COS	计算指定值的余弦值
21	DEGRAD	将值转换为弧度 (↔ RADDEG)
24	DIST	计算指定 2 个点之间的距离
43	INT	获得舍去小数点以下部分的整数值
52	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)
77	RADDEG	将值转换为度 (↔ DEGRAD)
85	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)
93	SIN	计算指定值的正弦值
98	SQR	计算指定值的平方根
102	TAN	计算指定值的正切值
111	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
17	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期
46	LEFT \$	从指定的字符串左侧抽出指定比特数的字符串
54	MID \$	从指定的字符串中抽出任意长度的字符串
83	RIGHT \$	从指定的字符串右侧抽出指定比特数的字符串
97	STR \$	将指定值转换为字符串 (↔ VAL)

相关参数

No.	函数名	功能
120	_SYSFLG	监视轴状态的标志
31	ERL	赋予发生错误的行编号
31	ERR	赋予发生错误时的错误代码编号
103	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数递增值
104	TIME \$	通过 "hh:mm:ss" 形式的字符串计算当前时间
105	TIMER	用上午 0 点开始的秒计算当前时间

1

ABS

计算绝对值

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

格式

主组

ABSRPOS (< 轴编号 1 >)

格式

副组

ABSRPOS2 (< 轴编号 2 >)

值

< 轴编号 1 > 主组 : 1 ~ 6

< 轴编号 2 > 副组 : 1 ~ 4

含义

获取 < 轴编号 > 中指定轴的机器参照值 (单位 : %)。

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)..... 将主组第 4 轴的机器参照值赋值给变量 A

A

B

C

D

E

F

G

H

I

J

K

L

M

ABSRST
绝对式马达轴返回原点动作

格式

ABSRST

含义 执行机器人的绝对式马达轴直接返回原点动作（绝对式原点复位）。
如果中途停止，将变为未返回原点状态。
设置 2 台机器人时，主组返回原点动作完成后，将进行副机器人组的返回原点动作。

 MEMO

- 此命令的对象轴为返回原点方式未设置为“标记”的轴。
- 返回原点方式指定为“标记”的轴处于未返回原点状态时，将无法执行该命令。
- 绝对式马达轴与增量式马达轴混在一起时，如果执行 ABSRST，将只进行绝对式马达轴的返回原点动作。
- 若要进行增量马达轴的返回原点，请执行 ORIGIN 命令。
此外，返回原点按照返回原点顺序参数中指定的顺序进行，增量马达轴不动作。

SAMPLE

```
*ABS_RST:
IF DI (20) =1 THEN ..... DI (20) 为 1 时执行 ABSRST
ABSRST
ENDIF
*START:ABSRST ..... 进行绝对式马达返回原点动作
```

相关命令 ORIGIN, ORGORD, ORGORD2, MCHREF, MCHREF2



要点

- ACCEL2 仅限系统生成过程中设置了副组时使用。

格式

主组

1. ACCEL <表达式>
2. ACCEL (<轴编号>) = <表达式>

格式

副组

1. ACCEL2 <表达式>
2. ACCEL2 (<轴编号>) = <表达式>

值

<轴编号> 主组：1 ~ 6
副组：1 ~ 4
<表达式> 1 ~ 100 (单位：%)

含义

将直接加速度系数参数更改为<表达式>中的值。

格式 1 更改组内的所有轴。

格式 2 只更改<轴编号>中指定的轴。



MEMO

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并中止执行。
- 重写 SYSTEM > PARAMETER > AXIS > ACCEL 中设置的值。程序中已声明的值优先。

获取

格式

主组

ACCEL (<轴编号 1>)

格式

副组

ACCEL2 (<轴编号 2>)

值

<轴编号 1> 主组：1 ~ 6
<轴编号 2> 副组：1 ~ 4

含义

获取<轴编号>中指定轴的加速度参数值。

SAMPLE

```

A=50
ACCEL A ..... 将所有轴的加速度系数设置为 50%
ACCEL (3)=100 ..... 仅将第 3 轴设置为 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)..... 将主组第 3 轴的加速度系数参数赋值给变量 A
HALT "END TEST"
  
```



要点

- ARCH2 仅限系统生成过程中设置了副组时使用。

格式

主组

1. ARCH <表达式>
2. ARCH (<轴编号>) = <表达式>

格式

副组

1. ARCH2 <表达式>
2. ARCH2 (<轴编号>) = <表达式>

值

<轴编号> 主组：1 ~ 6
副组：1 ~ 4
<表达式> 1 ~ 6144000 (单位：脉冲)

含义

将参数的圆弧位置更改为<表达式>中的值。
格式 1 更改组内的所有轴。
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

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并中止执行。

获取

格式

主组

ARCH (<轴编号 1>)

格式

副组

ARCH2 (<轴编号 2>)

值

<轴编号 1> 主组：1 ~ 6
<轴编号 2> 副组：1 ~ 4

含义

获取<轴编号>中指定轴的圆弧位置参数值。

SAMPLE

```

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

```

A

B

C

D

E

F

G

H

I

J

K

L

M

6

ARMCND / ARMCND2
获取机械臂状态

格式	主组
ARMCND	

格式	副组
ARMCND2	

含义 对水平多关节型机器人赋予当前机械臂的状态。
当机械臂状态为右手系统时为 0，左手系统时为 1。
此函数仅在使用水平多关节型机器人时有效。

SAMPLE	
A=ARMCND.....	将主机器人的当前机械臂状态赋值给变量 A
IF A=0 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

主组

ARMTYPE

格式

副组

ARMTYPE2

含义

赋予水平多关节型机器人当前所选手系统。

当所选手系统为右手系统时为 0，左手系统时为 1。

此函数仅在使用水平多关节型机器人时有效。

SAMPLE

```
A=ARMTYPE ..... 赋予主机器人的机械臂类型值
IF A=0 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
```

A

B

C

D

E

F

G

H

I

J

K

L

M

8

ATN
计算反正切值

格式	
ATN (<表达式>)	
含义	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。
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)
相关命令	COS, DEGRAD, RADDEG, SIN, TAN



要点

- ASPEED2 仅限系统生成过程中设置了副组时使用。



要点

- 自动移动速度
通过手持编程器的操作或
ASPEED / ASPEED2 命令设置
- 程序移动速度
由 SPEED / SPEED2 命令或
MOVE / MOVE2、DRIVE /
DRIVE2 的速度指定设置

格式

主组

ASPEED <表达式>

格式

副组

ASPEED2 <表达式>

值

<表达式> 1 ~ 100 (单位 : %)

含义

将直接自动移动速度更改为<表达式>中指定的值。

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 / ASPEED2 命令设置的自动移动速度与 SPEED / SPEED2 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%)

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, SPEED2



要点

- AXWGHT2 仅限系统生成过程中设置了副组时使用。

格式

主组

AXWGHT (<轴编号>) = <表达式>

格式

副组

AXWGHT2 (<轴编号>) = <表达式>

值

<轴编号> 主组：1 ~ 6

副组：1 ~ 4

<表达式> 根据设置的机器人而异

含义

将直接<轴编号>中指定组内轴的轴前端重量参数更改为<表达式>中的值。
 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)。

获取

格式

主组

AXWGHT (<轴编号 1>)

格式

副组

AXWGHT2 (<轴编号 2>)

值

<轴编号 1> 主组：1 ~ 6

<轴编号 2> 副组：1 ~ 4

含义

赋予<表达式>中指定轴的轴前端重量参数值。
 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, WEIGHT2



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。

- 带有 1 个实参的 CALL 语句，最多可连续使用 15 次。但是，根据 IF 语句或 GOSUB 语句等使用堆栈命令的使用状态或 CALL 语句的自变量（参数的一种）数量不同，次数将减少。
- 通过 CALL 语句调用子过程时，请务必用 END SUB 语句结束子过程。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, DECLARE, EXIT SUB, SHARED

12

CHANGE / CHANGE2
切换机械手

格式	主组
CHANGE Hn	

格式	副组
CHANGE2 Hn	

值	n : 主组用及副组用机械手编号的指定范围有所不同。 主组 0 ~ 3 副组 4 ~ 7
含义	通过 CHANGE / CHANGE2 切换机械手。 在切换机械手之前，请使用 HAND / HAND2 语句进行机械手的定义。 详情内容，请参阅 < 39 HAND / HAND2 >。
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	

格式

CHGPRI Tn, P

值 n : 任务编号 2 ~ 8
 p : 任务优先级 17 ~ 47

含义 将直接指定任务 n 的优先级更改为 p。
 任务 1 (主任务) 的优先级固定为 32。此外，不指定优先级时，优先级将变为 32。
 数字越小，优先级越高，数字越大，优先级越低 (高 17 ⇔ 低 47)。
 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 *SUBTASK, T2, 33
*ST:
  MOVE P, P0, P1
  IF DI (20) = 1 THEN
    CHGPRI T2, 32
  ELSE
    CHGPRI T2, 33
  ENDIF
GOTO *ST
HALT
' SUBTASK ROUTINE
*SUBTASK:
  IF LOCZ (WHERE) > 10000 THEN
    DO (20) = 1
    GOTO *SUBTASK
  ENDIF
  DO (20) = 0
GOTO *SUBTASK
EXIT TASK
```

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

A

B

C

D

E

F

G

H

I

J

K

L

M

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

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

A

B

C

D

E

F

G

H

I

J

K

L

M

格式	
CUT T _n	
值	n：任务编号.....2～8
含义	强制结束直接执行中或暂停中的其他任务。 无法强制结束当前的任务。此外，无法强制结束任务 1。
SAMPLE	
<pre>' TASK1 ROUTINE *ST: MO (20) = 0 START *SUBTASK2,T2 MOVE P, P0 MOVE P, P1 WAIT MO (20) = 1 CUT T2 GOTO *ST HALT ' TASK2 ROUTINE *SUBTASK2: P100=JTOXY (WHERE) IF LOCZ (P100) > = 100.0 THEN MO (20) = 1 ELSE DELAY 100 ENDIF GOTO *SUBTASK2 EXIT TASK</pre>	
相关命令	EXIT TASK, CUT, RESTART, START, SUSPEND

格式

DATE\$

含义

通过 yy/mm/dd 形式的字符串赋予日期。
yy 表示公历年份的后 2 位，mm 表示月，dd 表示日。
日期的设置在系统模式的初始处理中进行。

SAMPLE

A\$=DATE\$
PRINT DATE\$
HALT

相关命令

TIME\$

A

B

C

D

E

F

G

H

I

J

K

L

M



要点

- DECEL / DECEL2 在软件版本 Ver.8.15 以上有效。
- DECEL2 仅限系统生成过程中设置了副组时使用。

格式

主组

1. DECEL <表达式>
2. DECEL (<轴编号>) = <表达式>

格式

副组

1. DECEL2 <表达式>
2. DECEL2 (<轴编号>) = <表达式>

值

<轴编号> 主组：1 ~ 6
副组：1 ~ 4
<表达式> 1 ~ 100 (单位：%)

含义

将减速比率参数更改为<表达式>中的值。
格式 1 更改组内的所有轴。
格式 2 仅更改<轴编号>中指定的轴。



MEMO

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并中止执行。
- 命令语句：使用 DECEL / DECEL2 可更改加速度参数。

获取

格式

主组

DECEL (<轴编号 1 >)

格式

副组

DECEL2 (<轴编号 2 >)

值

<轴编号 1 > 主组：1 ~ 6
<轴编号 2 > 副组：1 ~ 4

含义

赋予<表达式>中指定轴的减速比率参数值。

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) ..... 将主组第 3 轴的减速比率参数赋值给变量 A
HALT "END TEST"
```



注意

· 可使用外部标签的仅限 GOSUB、CALL、ON ~ GOSUB 语句。

格式

- 1. DECLARE <标签> [, <标签> ...]
- 2. DECLARE SUB <名称> [(<形参> [, <形参>] ...)]

值

<标签>COMMON 程序中定义的子例程标签
<名称>COMMON 程序中定义的子过程名称
<*形参>子过程的自变量(参数的一种)。仅自变量的个数与数据类型具有含义。
*形参：形式参数的简称。

含义

声明直接标签或子过程在 COMMON 程序中。声明子过程时，也检查自变量的数据类型。此语句在子过程内无法定义。
DECLARE 语句是声明 COMMON 程序中的标签或子过程的命令，无法在 COMMON 程序中记述。
程序整体有效。

SAMPLE

```
COMMON 程序的标签共享
程序名：DIST1
' =====
'   MAIN PROGRAM
' =====

DECLARE *DISTANCE,*AREA
X!=2.5
Y!=1.2
GOSUB *DISTANCE
GOSUB *AREA
HALT

程序名：COMMON
' =====
'   'COMMON' PROGRAM
' =====

*DISTANCE:
    PRINT SQR (X!^2+Y!^2)
RETURN
*AREA:
    PRINT X!*Y!
RETURN
```

A

B

C

D

E

F

G

H

I

J

K

L

M

SAMPLE

外部程序的子过程共享

程序名：DIST2

```
' =====  
'      MAIN PROGRAM  
' =====  
      DECLARE SUB *DISTANCE (X!, Y!, D!)  
      DECLARE SUB *AREA (X!, Y!, A!)  
      CALL *DISTANCE (2. 5, 1. 2, REF D!)  
      PRINT D!  
      CALL *AREA (2. 5, 1. 2, REF A!)  
      PRINT A!  
      HALT  
      程序名：COMMON  
' =====  
'      ' COMMON' PROGRAM  
' =====  
      SUB *DISTANCE (X!, Y!, D!)  
          D!=SQR (X!^2+Y!^2)  
      END SUB  
      SUB *AREA (X!, Y!, A!)  
          A!=X!*Y!  
      END SUB
```

相关命令 CALL, EXIT SUB, GOSUB, ON ~ GOSUB, SUB, END SUB

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，程序中也也可以存在相同变量名。
- 调用使用<形参>的函数时，指定与<形参>相同类型的常量、变量或表达式。
- <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
```

DEGRAD
角度转换（度→弧度）

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

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 10ms。

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> = DIm ([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 $[b, \dots, b]$ 时, 8 比特均将成为对象。

SAMPLE

A%=DI12 ()	将 DI (27) 至 DI (20) 的输入状态赋值给变量 A%
A%=DI15 (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 < 9.5 并行输入变量 >。

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	
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	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

1. [LET] $\text{DOm}([b, \dots, b]) = \langle \text{表达式} \rangle$
2. [LET] $\text{DO}([b, \dots, b]) = \langle \text{表达式} \rangle$

值

m: 端口编号.....2 ~ 7, 10 ~ 17, 20 ~ 27

b: 指定ビット..... 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 $[b, \dots, b]$ 时, 8 比特均将成为对象。

无法输出至 D00 () 与 D01 ()。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	主组
DRIVE (< 轴编号 > , < 表达式 >) [, (< 轴编号 > , < 表达式 >) ...] [, 选项]	

格式	副组
DRIVE2 (< 轴编号 > , < 表达式 >) [, (< 轴编号 > , < 表达式 >) ...] [, 选项]	

值	< 轴编号 >主组：1 ～ 6 副组：1 ～ 4
	< 表达式 >马达位置(毫米、度、脉冲)或坐标点表达式

含义	执行组内以轴为单位的绝对移动命令。 要对组内附加轴进行绝对移动时，请使用此命令。 · 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、扭矩限制指定、XY 指定
----	---

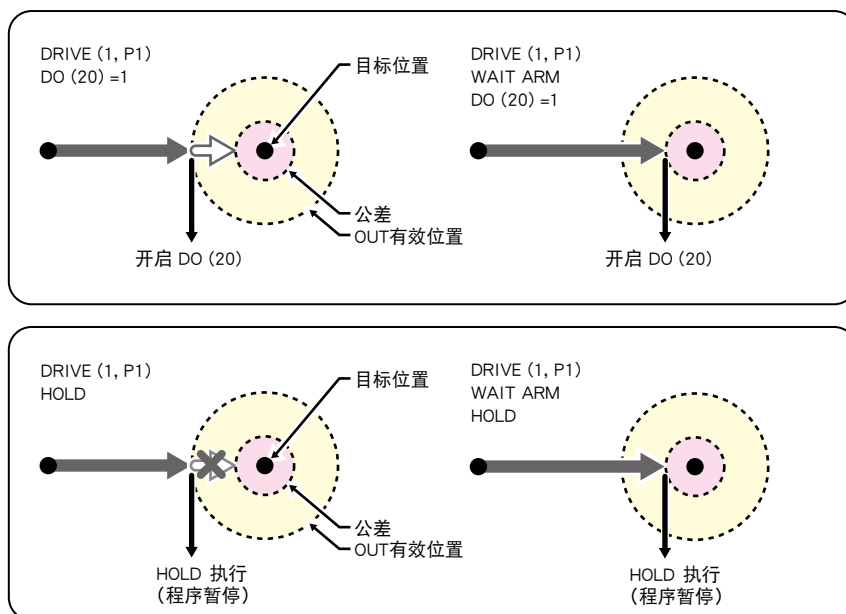
移动种类

- 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。
DRIVE / DRIVE2 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。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
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 / WAIT ARM2" 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) 主轴的第 1 轴从当前位置向 10000 脉冲位置移动

DRIVE2 (2, 90.00) 副组的第 2 轴从当前位置向 0 脉冲位置的 +90 度位置方向移动 (旋转轴情况下)

A

B

C

D

E

F

G

H

I

J

K

L

M



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。

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。

SAMPLE

DRIVE (1, P1) 主轴的第 1 轴从当前位置向 P1 所指定位置移动
 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>

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 / DRIVE2 语句有效。

SAMPLE

DRIVE2 (1, 10000), S=10 副组的第 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 / DRIVE2 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

DRIVE2 (1, 10000), DS=0.1 副组的第 1 轴按照机器人最高速度的 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

- 速度指定 (DSPEED) 在以下软件版本条件下有效。

RCX14x Ver.8.71 以上
 RCX22x Ver.9.18 以上

- 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

移动中条件表达式不成立时，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。

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。



要点

- 此参数在以下软件版本条件下有效。

RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上

- 在比上述版本更旧版本的控制器上，-1 时为“真”，非 -1 时为“假”（不可更改）。



注意

- 扭矩限制指定中的扭矩偏移值指定在软件版本 Ver.8.18 以上有效。
- 扭矩控制指定的设置范围根据所使用的机器人而异。设置超过了机器人的使用范围时，可能会成为机器人故障或误动作的原因。
- 指定的扭矩限制值较小时，轴有时可能无法移动。此外，垂直轴可能会跌落。

MEMO

- 当 STOPON 条件指定中使用的条件表达式为数值表达式时，要变为“真”、“假”的条件可在控制器的 "Other parameters" 中的 "TRUE conditions" 中更改。此条件在 IF、WHILE、WAIT、STOPON 等所有条件表达式中通用。详细说明，请参阅控制器的手册。

- 1) -1 (初始值) 表达式的值为 -1 时，变为“真”；值为 0 时，变为“假”。
值变为 -1、0 以外时，将发生 [6.35: Expression value error] 错误。
- 2) not 0 表达式的值非 0 时，变为“真”；值为 0 时，变为“假”。

● 扭矩限制指定

格式

1. T = < 扭矩限定值 >
2. T = (< 扭矩限制值 > [, < 扭矩偏移值 >])

值

< 扭矩限制值 > 1 ~ 100 (单位 : %)
< 扭矩偏移值 > -100 ~ 100

含义

在进行扭矩控制的同时移动轴。

此时扭矩的最大值限制为额定扭矩 × < 扭矩限制值 > / 100 的值。

使用垂直规格轴等的 < 扭矩限制值 >，由一贯恒定负荷作用的轴进行扭矩控制时的指定。省略此值时，以及在格式 1 中，扭矩偏移值将被视作指定为“0”。

关于扭矩偏移值，请参考轴在伺服保持呈停止状态时，所显示的电流指令监控值，并进行指定。

但是，电流指令监视器上显示的值为最大扭矩比。由于扭矩偏移值变为额定扭矩比，所以约为监视器上所显示值的 1/3 为准。

MEMO

- 按手持编程器的 DISPLAY 键可显示电流指令监视器。详细说明，请参阅各控制器的手册。

如果指定了本选项并执行了 DRIVE / DRIVE2，请将轴的最大扭矩值更改为＜扭矩限制值＞中指定的值，并在进行扭矩控制的同时，向着目标位置移动。

此时移动速度的上限为通常速度的 10%。移动中即使接触到障碍物，也不会出错，轴的扭矩（推力）不会超过限制值。



要点

· 右侧所示的命令结束条件在软件版本 Ver.8.45 以上时有效。未达到上述版本时的命令结束条件如下。

1. 向目标位置移动，经过所需时间时，轴到达目标位置的情况下。
2. 向目标位置移动，经过所需时间时，轴的扭矩已经达到限制值状态超过 1 秒的情况下。
3. 向目标位置移动，经过所需时间时，扭矩达到限制值，且到达限制值状态持续 1 秒以上的情况下。



注意

· 驱动器版本为 Ver. 2.17 以下时，最大扭矩指令值在 TORQUE / TORQUE2 语句中指定时生效。

· 命令结束条件

1. 轴到达目标位置。
2. 轴的扭矩（推力）达到限制值状态超过了 TRQTIME / TRQTIME2 语句中指定的时间（超时时间）。

· TRQSTS 命令的值

1. 轴的扭矩达到限制值状态超过了超时时间，命令结束时将被设置为“1”。
2. 由于其他原因命令结束时，将被设置为“0”。

· 注意事项

1. 使用 TORQUE / TORQUE2 语句更改的最大扭矩指令值不会立即生效。将在执行下一个 MOVE 语句或 DRIVE 语句等移动命令时生效。
2. 即使本命令结束，最大扭矩限制及扭矩控制状态也将继续。在执行本命令过程中由于联锁等原因停止时也是一样。
3. 如果执行了伺服 ON / OFF 操作或 MOVE 命令等对轴执行了任何操作，扭矩控制将被解除。
4. 若要解除最大扭矩限制，请使用 TORQUE 语句重新设置最大扭矩指令值。
5. 在以下情况下，无论是否执行 TORQUE 语句，最大扭矩限制均将被解除。
 - 开启控制器的电源时
 - 伺服禁止时
 - 执行了返回原点或绝对式原点复位（标记方式除外）时
 - 更改了参数或进行了初始化时

· 限制事项

1. 本选项无法用于指定多个轴。
2. 移动时的最高速度为通常的 10%。
3. 执行本命令并变为扭矩控制状态的轴无法进行手动移动。

SAMPLE

```

TRQTIME (3) = 2500 ..... 将第 3 轴的扭矩控制超时时间设置为 2.5 秒
DRIVE (3, P1), T = (20, 15) ..... 将最大扭矩值设置为额定扭矩的 20%，将扭矩
                                   偏移值设置为 15，第 3 轴从当前位置向 P1 指
                                   定位置移动（推进动作）
IF TRQSTS (3) = 1 THEN ..... 确认是否超时
  DO (21) = 1 ..... 已超时（推进结束。此处将结果输出至 DO (21)）
ELSE
  DO (21) = 0 ..... 未超时（到达目标位置，推进不良。此处将结果
                                   输出至 DO (21)）
ENDIF
TORQUE (3) = 100 ..... 恢复（100%）最大扭矩指令值
DRIVE (3, P0) ..... 结束扭矩限制及扭矩控制并向 P0 移动
  
```

● 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 / MOVE2 语句中指定了所有可移动的轴时，将变为与 MOVE / MOVE2 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
  
```

相关命令

TORQUE, TORQUE2, TRQTIME, TRQTIME2, TRQSYS, TRQSYS2

格式

主组

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

格式

副组

DRIVEI2 (<轴编号> , <表达式>) [, (<轴编号> , <表达式>) ...] [, 选项]

值

<轴编号> 1 ~ 4
<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。
要对组内附加轴进行相对移动时, 请使用此移动命令。
· 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

MEMO



要点

- 此参数在以下软件版本条件下有效。
RCX14x Ver.8.66 以上
RCX22x Ver.9.13 以上
- 在比上述版本更旧版本的控制器上, 变为“复位”动作。

- 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(与之前位置互换)。

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。

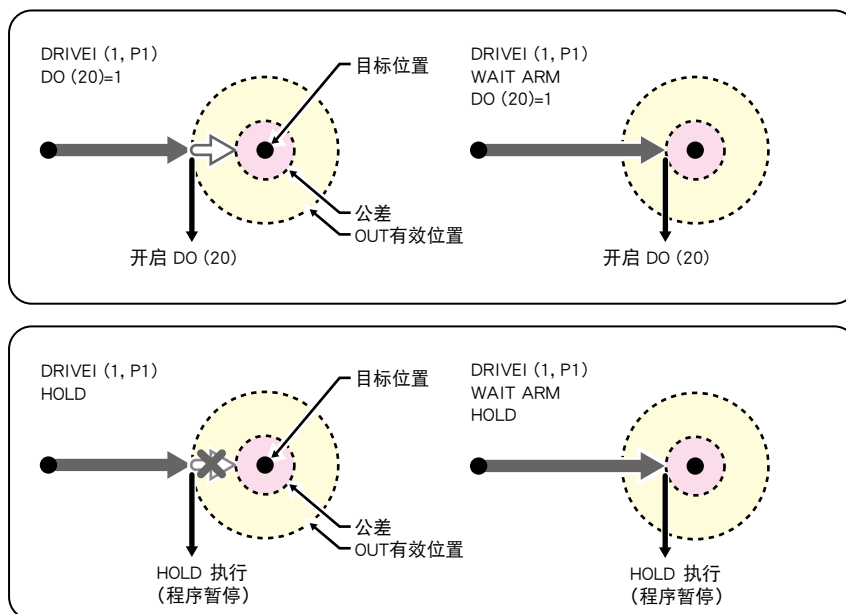
DRIVEI / DRIVEI2 命令的下一个命令为信号输出命令等可执行命令时, 移动对象轴进入 OUT 有效位置范围内后, 将执行下一个命令。因此, 请在轴到达目标位置的公差范围内之前, 执行下一个命令。

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。
HOLD	进入 OUT 有效范围内后, 程序将暂停。因此, 移动将停止。
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 / WAIT ARM2" 语句。

DRIVEI 命令 WAIT ARM 语句



35820-R7-00

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
`DRIVE (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)

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 / DRIVEI2 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 / DRIVE2 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

- 速度指定 (DSPEED) 在以下软件版本条件下有效。
RCX14x Ver.8.71 以上
RCX22x Ver.9.18 以上
- 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



要点

· 此参数在以下软件版本条件下有效。

RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上

· 在比上述版本更旧版本的控制器上，-1 时为“真”，非 -1 时为“假”（不可更改）。

· 当 STOPON 条件指定中使用的条件表达式为数值表达式时，要变为“真”、“假”的条件可在控制器的 "Other parameters" 中的 "TRUE conditions" 中更改。此条件在 IF、WHILE、WAIT、STOPON 等所有条件表达式中通用。详细说明，请参阅控制器的手册。

- 1) -1 (初始值) 表达式的值为 -1 时，变为“真”；值为 0 时，变为“假”。
 值变为 -1、0 以外时，将发生 [6.35 : Expression value error] 错误。
- 2) not 0 表达式的值非 0 时，变为“真”；值为 0 时，变为“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	
<pre>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</pre>	
含义	直接结束 SELECT CASE 的命令区。 详细内容，请参阅< 87 SELECT CASE >。
SAMPLE	
<pre>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</pre>	
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 < 99 SUB ~ ENDSUB >。

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
ERL

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。
ERR 获取发生错误时的错误编码编号，ERL 获取错误发生的行编号。

SAMPLE 1

IF ERR < > &H604 THEN HALT
IF ERL=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。
此命令语句仅在任务 1 以外的任务时有效（无法结束任务 1）。

 MEMO

- 作为副任务启动的任务即使通过 GOTO 等语句跳转至其他任务处理例程，也将该处理例程作为此副任务的处理进行执行。

SAMPLE

```
' TASK1 ROUTINE
*ST:
    MO (20) =0
    START *SUBTASK2,T2
    MOVE P, P0, P1
    WAIT MO (20) =1
    GOTO *ST
HALT
' TASK2 ROUTINE
*SUBTASK2:
    P100=JTOXY (WHERE)
    IF LOCZ (P100) > =100.0 THEN
        MO (20) =1
        EXIT TASK
    ENDIF
    DELAY 100
GOTO *SUBTASK2
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 29 次。但是，根据 IF 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

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。

MEMO

- 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。
- 存在程序名为 "_SELECT" 的程序时，在执行 HALT 后，将选择 "_SELECT"。

SAMPLE

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

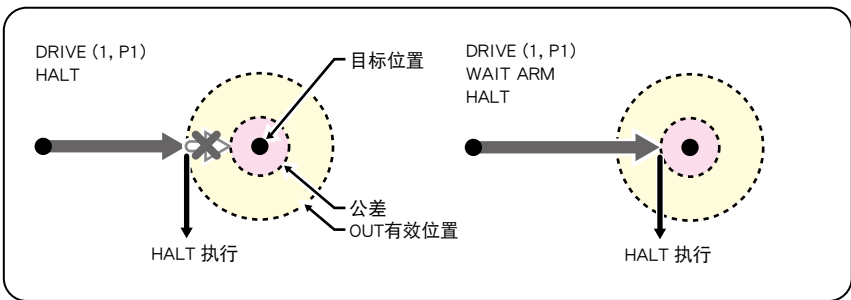
在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。

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 / MOVE2 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 / MOVE2 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 / WAIT ARM2。

HALT命令



35821-R7-00

参照 关于 "_SELECT", 请参阅第 1 章 < 4 程序名 >。

格式

主组

定义语句：

HAND Hn= <第 1 参数> <第 2 参数> <第 3 参数> [R]

选择语句：

CHANGE Hn

格式

副组

定义语句：

HAND2 Hn= <第 1 参数> <第 2 参数> <第 3 参数> [R]

选择语句：

CHANGE2 Hn

值

n：机械手编号主组：0 ~ 3 → HAND 使用
副组：4 ~ 7 → HAND2 使用

含义

HAND / HAND2 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 / CHANGE2 语句。

关于 CHANGE / CHANGE2 语句, 请参阅< 12 CHANGE / CHANGE2 >。

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。

39.1

水平多关节型机器人时

1. 未指定<第 4 参数>的 R 时

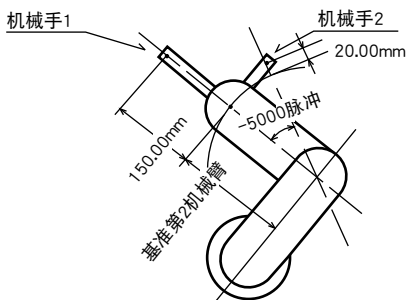
成为固定在第 2 机械臂前端的机械手(参照下图)。

<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



35801-R7-00

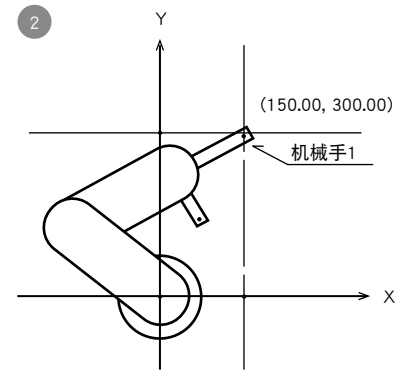
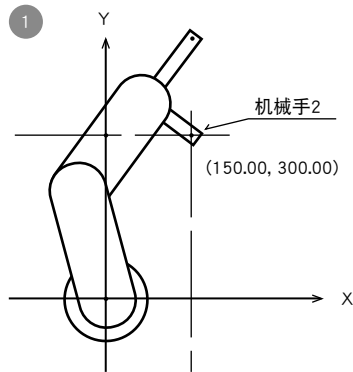
SAMPLE

```

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

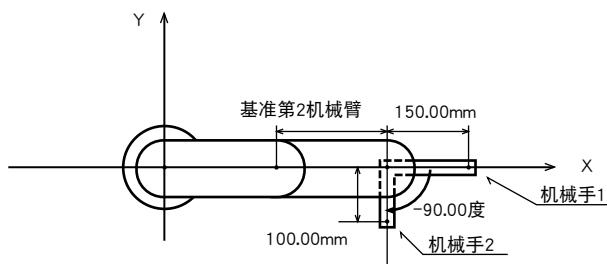
R轴为伺服时成为从R轴旋转中心偏移的机械手(参照下图)。

<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

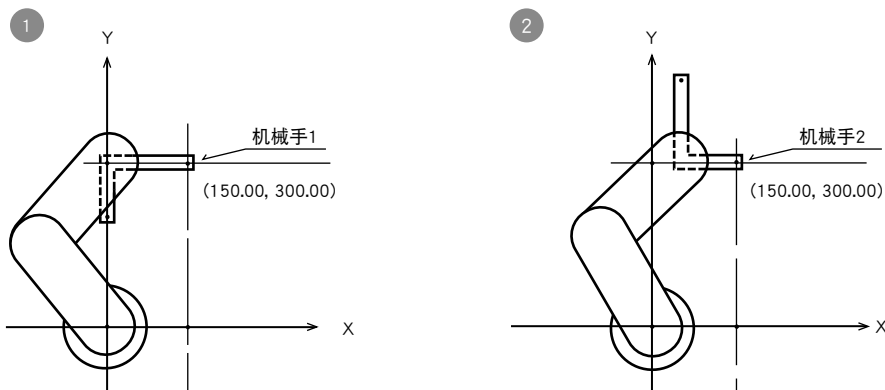


35803-R7-00

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

39.2 正交机器人时

1. 未指定<第4参数>的R时

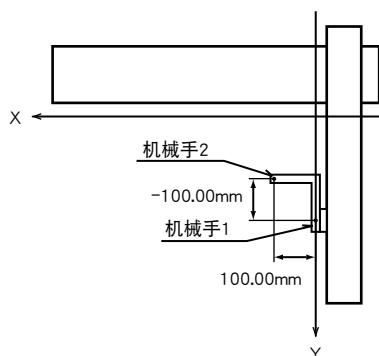
成为固定在第2机械臂前端的机械手(参照下图)。

<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



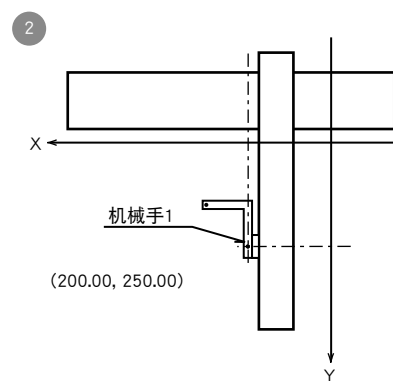
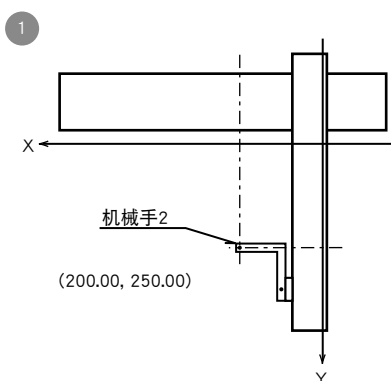
35805-R7-00

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

A

B

C

D

E

F

G

H

I

J

K

L

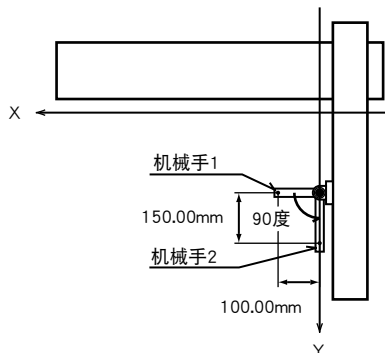
M

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

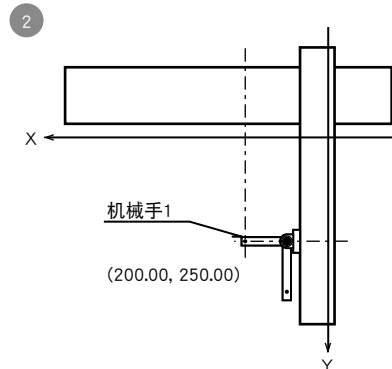
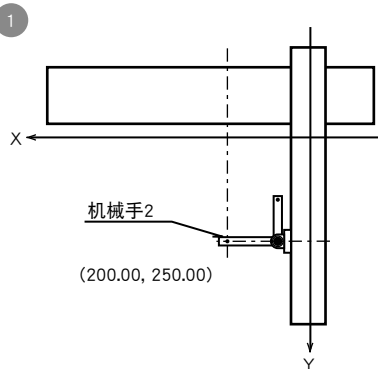


35807-R7-00

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。

SAMPLE

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

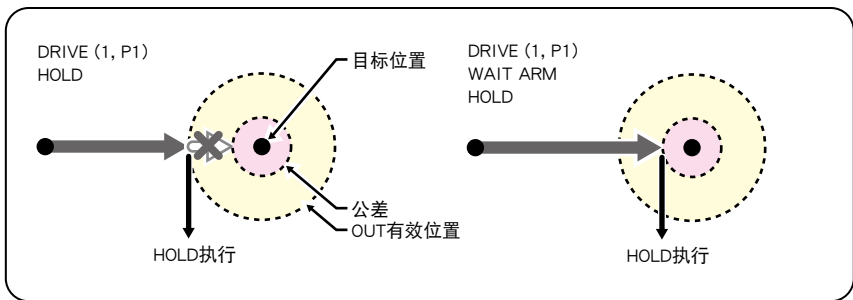
在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。

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 / MOVE2 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 / MOVE2 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 / WAIT ARM2。

HOLD命令



35822-R7-00



要点

- 此参数在以下软件版本条件下有效。

RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上

- 在比上述版本更旧版本的控制器上，-1 时为“真”，非 -1 时为“假”（不可更改）。



MEMO

- 当 IF 语句中使用的条件表达式为数值表达式时，变为“真”、“假”的条件可在控制器的 "Other parameters" 中的 "TRUE conditions" 中更改。此条件在 IF、WHILE、WAIT、STOPON 等所有条件表达式中通用。详细说明，请参阅控制器的手册。

- 1 (初始值) 表达式的值为 -1 时，变为“真”；值为 0 时，变为“假”。
值变为 -1、0 以外时，将发生 [6.35: Expression value error] 错误。
- not 0 表达式的值非 0 时，变为“真”；值为 0 时，变为“假”。

41.1

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。

- 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。
- 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
```

41.2 区块 IF 语句



要点

- 此参数在以下软件版本条件下有效。
RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上
- 在比上述版本更旧版本的控制器上，-1 时为“真”，非 -1 时为“假”（不可更改）。

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>。



MEMO

- IF 语句最多可连续使用 48 次。但是，根据 GOSUB 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。

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
```

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

- 当变量与赋值的类型不同时，将显示 [? Input again] 的提示时，并等待再次输入。
- 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

44

JTOXY / JTOXY2

轴单位制转换（脉冲→毫米）

格式	主组
JTOXY（<坐标点表达式>）	
格式	副组
JTOXY2（<坐标点表达式>）	
含义	将<坐标点表达式>中表示的关节坐标数据（单位：脉冲）转换为正交坐标数据（单位：mm、度）。
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, XYTOJ2

格式

* < 标签 > :

含义

在程序行中定义<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 16 个字符(超过 16 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
```

A

B

C

D

E

F

G

H

I

J

K

L

M

格式

LEFT\$ (<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 ~ 75 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\$

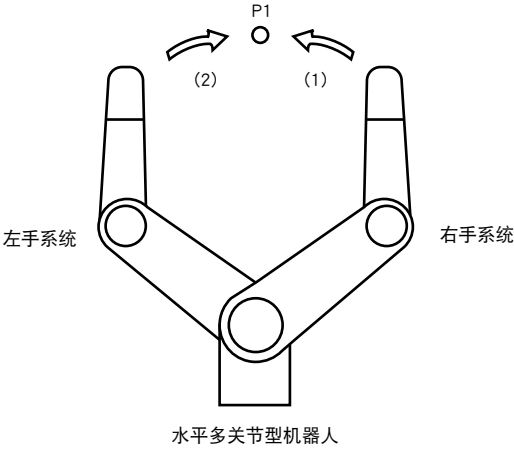
格式	主组
LEFTY	

格式	副组
LEFTY2	

含义 选择使用左手系统向正交坐标系中指定的坐标点移动。
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, RIGHTY2
------	-----------------

A

B

C

D

E

F

G

H

I

J

K

L

M

LEN

获取字符串的长度

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>	= <表达式>
	<坐标点元素变量>	
	<位移元素变量>	
	<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>	

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)
LOCZ (P0) =A!
LOCX (S1) =100.00
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

A

B

C

D

E

F

G

H

I

J

K

L

M

格式	
1. LO0 ([b, . . . , b])	= <表达式>
2. LO (0b, . . . , 0b)	= <表达式>
值	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 (07) 对应 1 轴 ~ 8 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

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.....X, Y, Z, R, A, B (指定轴), F (指定手系统标志)
 格式 2 : x.....X, Y, Z, R (指定轴)
 <表达式> 指定轴时 : 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)

含义 直接格式 1 : 更改坐标点数据指定轴的值或手系统标志。
 格式 2 : 更改从位移数据指定轴的值。

 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。

获取

格式

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

值 格式 1 : x.....X, Y, Z, R, A, B (指定轴), F (指定手系统标志)
 格式 2 : x.....X, Y, Z, R (指定轴)

含义 格式 1 : 从坐标点数据获取指定轴的值或手系统标志。
 格式 2 : 从位移数据获取指定轴的值。

SAMPLE

LOCX (P10) =A (1) 将 P10 的 X 轴数据更改为数组 A (1) 的值
LOCY (S1) =B 将 S1 的 Y 轴数据更改为 B 的值

A (1) =LOCX (P10) 将 P10 的 X 轴数据赋值至数组变量 A (1)
B (2) =LOCX (S1) 将 S1 的 X 轴数据赋值给数组 B (2)

相关命令 坐标点元素变量, 位移元素变量

A

B

C

D

E

F

G

H

I

J

K

L

M

52

LSHIFT
比特的左位移

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 向左位移 2 比特的值 (&B11101100) 赋值给 A	
相关命令	RSHIFT

格式		主组
MCHREF (< 轴编号 1 >)		
格式		副组
MCHREF2 (< 轴编号 2 >)		
值	< 轴编号 1 > 主组 : 1 ~ 6 < 轴编号 2 > 副组 : 1 ~ 4	
含义	提供<轴编号>中指定轴的返回原点动作及绝对位置搜索动作的机器参照(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) 将主组第 1 轴的返回原点动作机器参照赋值给 变量 A		

格式

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

值

<表达式 1> 1 ~ 75

<表达式 2> 0 ~ 75

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\$



· 关于指定比特，请参阅第 3 章 < 10 指定比特 >。

格式

- 1. MO0 ([b, . . . , b]) = < 表达式 >
- 2. MO (0b, . . . , 0b) = < 表达式 >

值

m : 端口编号 2 ~ 7, 10 ~ 17, 20 ~ 27
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 0、1 因保持各轴的原点传感器状态及轴的保持状态，所以仅可参照，无法输出。
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 0、1 的输出

比特	7	6	5	4	3	2	1	0
端口 0	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
	原点传感器状态 0 : 关闭 / 1 : 开启							
端口 1	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴
	保持状态 0 : 未保持 / 1 : 保持（未使用轴为 1）							

MEMO

· 关于 MO 端口 0、1 的详细信息，请参阅第 3 章 < 9.7 内部输出变量 >。

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

主组

MOVE	PTP	, <坐标点指定> , [, 选项 [, 选项] ...]
	P	
	L	
	C	

格式

副组

MOVE2	PTP	, <坐标点执行> , [, 选项 [, 选项] ...]
	P	
	L	
	C	



要点

- MOVE2 的直线插补和圆弧插补移动在以下软件版本条件下有效。

RCX14x	Ver.8.64 以上
RCX22x	Ver.9.11 以上

含义

直接执行组内轴的绝对位置移动命令。

在 MOVE 中设置为主机器人的轴, 在 MOVE2 中设置为副机器人的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、直线插补、圆弧插补
- 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

选配件	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 / MOVE2 P 命令之后的命令注意点

MOVE P / MOVE2 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。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
WAIT	进入 OUT 有效位置范围内后，将执行 WAIT 命令。

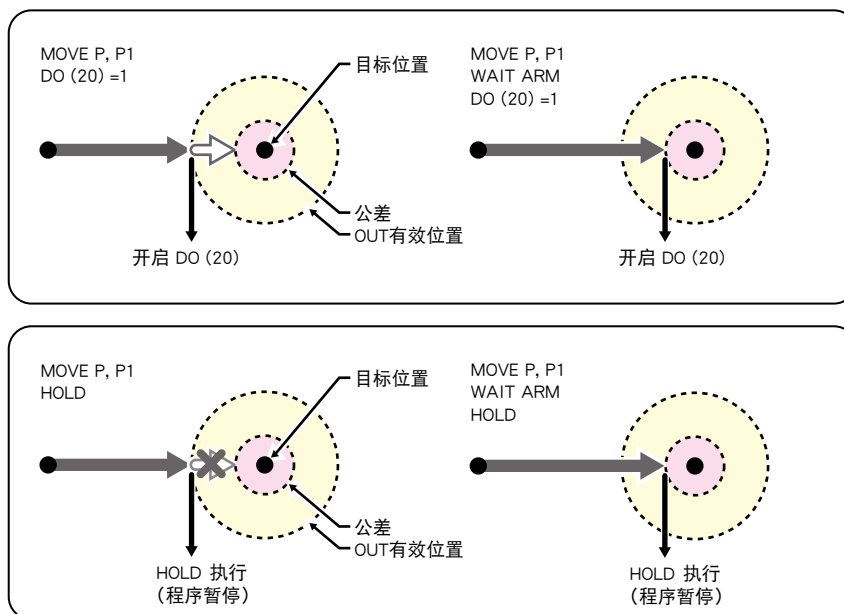
轴到达公差范围内后要执行下一个命令时，请使用 "WAIT ARM / WAIT ARM2" 语句。



MEMO

- OUT 有效位置的数值由参数设置。
也可在程序中通过 OUTPOS / OUTPOS2 命令更改值。

MOVE命令



35823-R7-00

SAMPLE

MOVE P, P0 主机器人轴从当前位置向 P0 指定的位置移动
(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。

- 直线插补移动

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。

但是，指定了 STOPON 条件时，执行结束条件如下所示。

执行结束条件：所有的移动对象轴均结束动作（到达公差范围内）。

随此，因连续 2 次移动时路径联结将无法继续移动。

- MOVE L / MOVE2 L 命令之后的命令注意点

MOVE L / MOVE2 L 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：

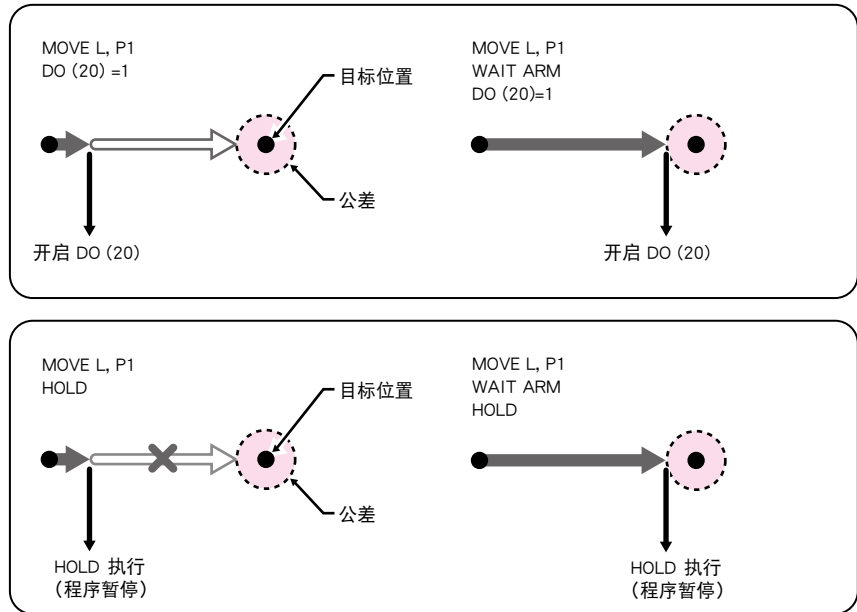
信号输出（DO 等）	移动开始后，将立即输出信号。
DELAY	移动开始后，将立即执行 DELAY 命令，并开始等待时间。
HALT	移动开始后，程序立即停止，并被复位。因此，移动将停止。
HOLD	移动开始后，程序立即暂停。因此，移动将停止。
WAIT	移动开始后，将立即执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时，请使用“WAIT ARM / WAIT ARM2”语句。

MEMO

- RCX14x: Ver.8.64 以下，RCX22x: Ver.9.11 以下情况下，直线插补仅在任务 1 中有效。
- 根据速度、加速度、坐标点间的距离不同，可能无法进行顺畅的路径移动，或者路径间的速度可能不恒定。
- 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 / MOVE2 L 之后执行 DELAY 语句时，执行 MOVE L / MOVE2 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 / WAIT ARM2”语句。
- HALT 命令等其他命令也同样处理。
- 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，导致出错。此时，请考虑降低连接部分的加减速度或使用 WAIT ARM 命令等修改动作形式。

MOVE命令

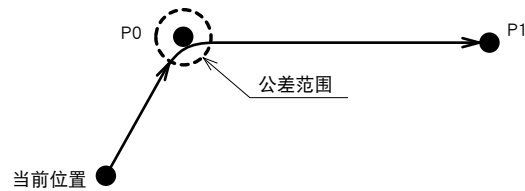


35824-R7-00

SAMPLE

MOVE L, P0, P1 主机器人轴从当前位置向 P0、P1 指定的位置移动

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 / MOVE2 C 命令之后的命令注意点

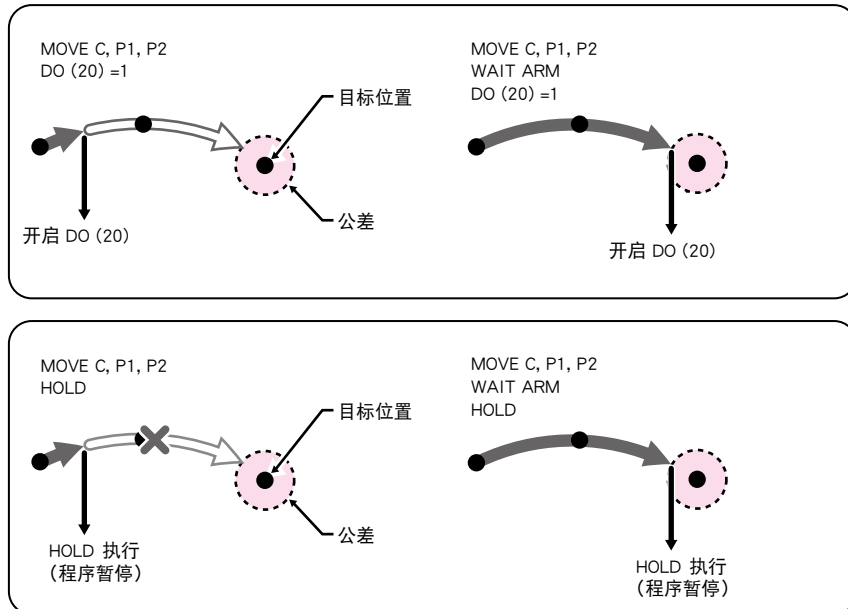
MOVE C / MOVE2 C 命令的下一个命令为信号输出命令等可执行命令时，将从移动开始后立即开始执行。

示例：

信号输出（DO 等）	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。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
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 / WAIT ARM2" 语句。

MOVE 命令

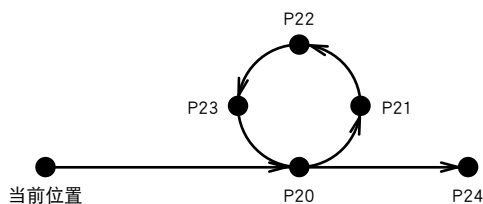


35825-R7-00

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



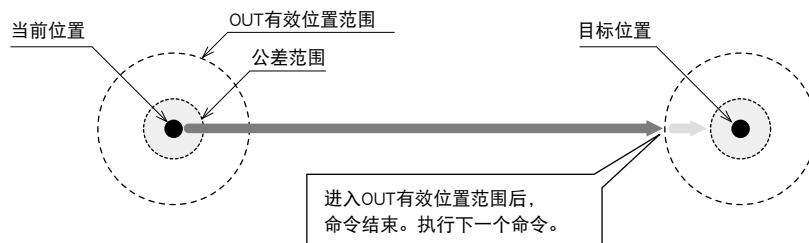
35811-R7-00

MEMO

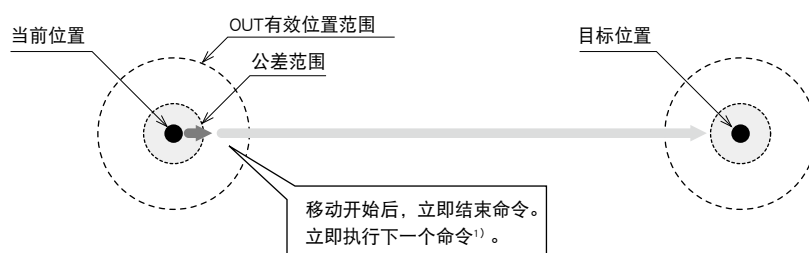
- 即使插补移动采用连续动作时，也不会在中途的坐标点停止。但是，最高速度低于 PTP。
- RCX14x: Ver.8.64 以下，RCX22x: Ver.9.11 以下情况下，圆弧插补仅在任务 1 中有效。
- 可进行圆弧插补的范围为半径 1.00mm ~ 5,000.00mm。
- 根据速度、加速度、坐标点间距离不同，圆弧可能会发生歪斜。
- 对于有 R 轴指定的机器人，根据 R 轴的移动距离不同，可能会由于 R 轴速度变得过快而出错。
- 在 MOVE L / MOVE2 L 之后执行 DELAY 语句时，执行 MOVE L / MOVE2 L 后，DELAY 的定时器立即动作。因此，**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 / WAIT ARM2" 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

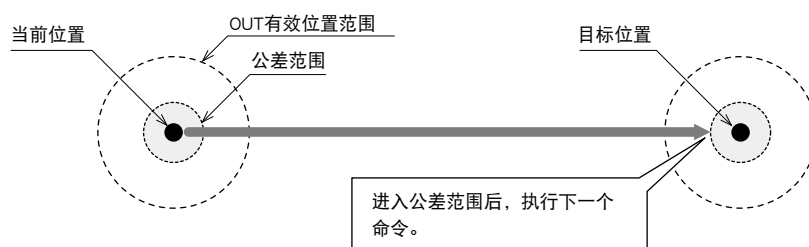
1. PTP动作



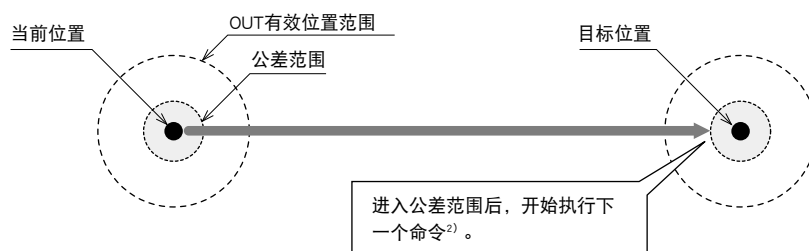
2. 插补移动



3. WAIT ARM



4. 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 / MOVE2 的后面插入 WAIT ARM / WAIT ARM2。

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>。)

坐标点指定的种类

● 直接数值指定

PTP

直线插补

圆弧插补

格式

X Y Z R A B [F]

值

X, Y, Z, R, A, B 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)

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。

可指定此移动的种类为 PTP 移动及直线插补移动。

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。

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

● 坐标点定义指定

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。

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, 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



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。
- 手系统标志在软件版本 Ver.8.08 以上时有效。



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。



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。

选项种类

● 速度指定 1

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

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。此选项为水平多关节型机器人或 XY 型机器人, 且可指定此移动的种类为直线插补移动及圆弧插补移动。
仅对指定的 MOVE / MOVE2 语句有效。

SAMPLE

MOVE L, P10, VEL=100 从当前位置按照 XYZ 的合成最高速度
100 mm / 秒向 P10 指定的位置移动



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

- 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。

● 圆弧插补移动指定

PTP 直线插补 圆弧插补

格式

x = <表达式>

值

xX, Y, Z, R, A, B 的轴指定
<表达式>整数值的的情况下, 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

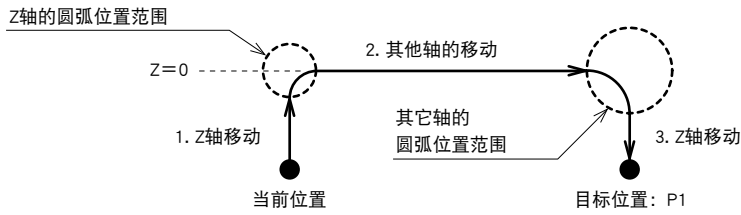
- 1. 使 x 中指定的轴开始移动至<表达式>中指定的位置(下图的 1)。
- 2. 如果 x 中指定的轴进入圆弧位置内, 其他轴将移动至目标位置(下图的 2)。
- 3. 如果 x 中指定轴以外的轴进入圆弧位置范围内, x 中指定的轴将移动至目标位置(下图的 3)。
- 4. 所有轴均进入 OUT 有效位置范围内后, 结束命令。

可指定此移动的种类为 PTP 移动。
x 中指定的轴为 X 轴或 Y 轴时, 根据机器人种类的不同, <表达式> 及目标位置的值仅限整数(脉冲单位)。
 <表达式> 的值: 水平多关节型机器人及 XY 型机器人
 目标值的值: 水平多关节型机器人
各值均显示为马达位置, 而不是坐标值。

SAMPLE

MOVE P, P1, Z=0 Z 轴从当前位置移动至 0 脉冲位置。之后, 其他轴移动至 P1, 最后 Z 轴移动至 P1

SAMPLE: MOVE Z



35813-R7-00



要点

· 轴的圆弧位置参数可通过 ARCH / ARCH2 更改。此值越大, 越可缩短移动的执行时间。

● STOPON 条件指定

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。



要点

- 此参数在以下软件版本条件下有效。

RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上

- 在比上述版本更旧版本的控制器上，-1 时为“真”，非 -1 时为“假”（不可更改）。



MEMO

- 当 STOPON 条件指定中使用的条件表达式为数值表达式时，变为“真”、“假”的条件可在控制器的 "Other parameters" 中的 "TRUE conditions" 中更改。此条件在 IF、WHILE、WAIT、STOPON 等所有条件表达式中通用。详细说明，请参阅控制器的手册。

- 1 (初始值) 表达式的值为 -1 时，变为“真”；值为 0 时，变为“假”。
值变为 -1、0 以外时，将发生 [6.35: Expression value error] 错误。
- not 0 表达式的值非 0 时，变为“真”；值为 0 时，变为“假”。

CONT 指定

PTP 圆弧插补 圆弧插补

格式
CONT



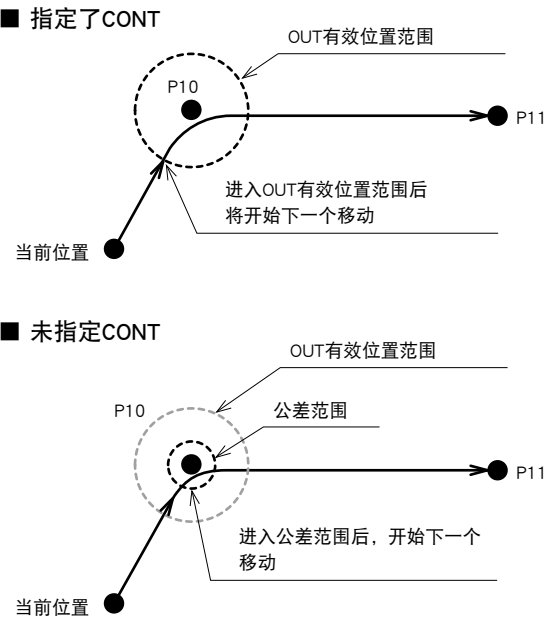
要点

- 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 指定的 PTP 移动时，所有移动对象轴进入 OUT 有效位置范围内结束命令后立即执行的 PTP 移动，将不等待移动对象轴动作结束（到达公差范围内），开始 PTP 移动。
可指定此移动的种类为 PTP 移动。
仅对指定的 MOVE / MOVE2 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



35814-R7-00

● 加速度指定

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 / MOVE2 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

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 / MOVE2 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 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 / MOVE2 语句有效。



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。

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

- 端口输出指定在软件版本 Ver.8.22 以上时有效。

● 端口输出指定

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 / MOVE2 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

MOVE1, MOVE12, DRIVE, DRIVE2, DRIVE1, DRIVE12, WAIT ARM, WAIT ARM2



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

主组

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

格式

副组

MOVEI2 | 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。
在 MOVEI 中设置为主组轴，在 MOVEI2 中设置为副组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
· 坐标点指定： 直接数值指定、坐标点定义指定
· 选项： 速度指定

选配件	PTP	直线插补	圆弧插补	备 注
速度指定 (SPEED)	○	○	○	仅指定 MOVE 语句有效



要点

- 此参数在以下软件版本条件下有效。
RCX14x Ver.8.66 以上
RCX22x Ver.9.13 以上
- 在比上述版本更旧版本的控制器上，将变为“复位”动作。



MEMO

- 中途停止 MOVEI 语句并重新启动时，可在控制器的 "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 (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 / MOVEI2 命令之后的命令注意点

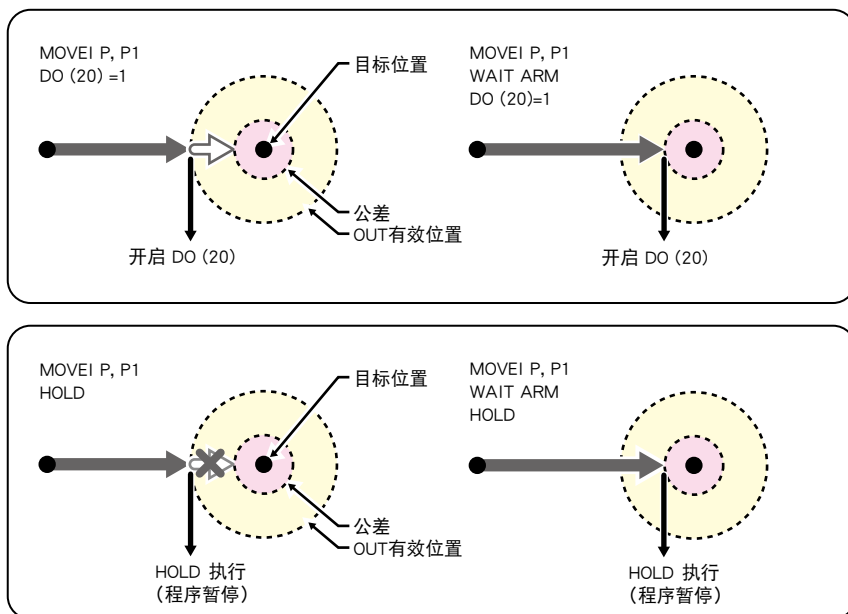
MOVEI / MOVEI2 命令的下一个命令为信号输出命令等可执行命令时，移动对象轴进入 OUT 有效位置范围内后，将开始执行下一个命令。即，在轴到达目标位置的公差范围之内之前，将开始下一个命令。

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。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
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 / WAIT ARM2" 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

X Y Z R A B [F]

值

X, Y, Z, R, A, B 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)

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。

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



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。
- 手系统标志在软件版本 Ver.8.08 以上时有效。

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 / MOVEI2 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, MOVE2, DRIVE, DRIVE2, DRIVEI, DRIVEI2, WAIT ARM, WAIT ARM2



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

- 对象端口的指定在以下软件版本条件下有效。
RCX240 Ver.10.20 以上
RCX22x Ver.9.27 以上
- 在比上述版本更旧版本的控制器上，无法进行对象端口的指定，仅限 RS-232C 可指定。

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 或无指定 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。

SAMPLE

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

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

59

ORD

计算字符编码

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\$

1. ON ERROR GOTO <标签>
2. ON ERROR GOTO 0

含义 在执行机器人语言过程中发生了错误时，不停止程序，跳转至<标签>中指定的错误处理例程，然后程序将继续（但是，在发生重大错误时，并不适用。）
指定了“0”代替<标签>时，将显示错误提示并停止程序。
在错误处理例程以外执行了 ON ERROR GOTO 0 时，ON ERROR GOTO 命令将被解除（中断解除）。
在错误处理例程中，使用错误输出信息 (ERR、ERL) 与 RESUME 语句进行错误处理

- 发生了 [17.4: Over load] 等重大错误时，执行将停止。
- ON ERROR GOTO <标签> 对最后执行的操作有效。
- 在错误处理例程中发生了错误时，执行将停止。
- 无法在错误处理例程内使用 ON ERROR GOTO <标签>。

```
ON ERROR GOTO *ER1
FOR A = 0 TO 9
    P [A+10] = P [A]
NEXT A
*L99: HALT
' ERROR ROUTINE
*ER1:
IF ERR = &H0604 THEN *NEXT1 ..... 确认是否发生了 [Point doesn't
                                         exist] 的错误
IF ERR = &H0606 THEN *NEXT2 ..... 确认是否发生了 [Subscript out of
                                         range] 的错误
ON ERROR GOTO 0 ..... 显示错误并停止执行
*NEXT1:
    RESUME NEXT ..... 跳转至错误发生行的下一行继续执行
*NEXT2:
    RESUME *L99 ..... 跳转至标签 *L99 并继续执行
```

ON ERROR GOTO ● 8-113

O

W

61

ON ~ GOSUB

执行由表达式的值指定的子例程

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, DECLARE

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, DECLARE

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

63

ONLINE

将通信端口设置为联机模式



要点

- 对象端口的指定在以下软件版本条件下有效。

RCX240

Ver.10.20 以上

RCX22x

Ver.9.27 以上
- 在比上述版本更旧版本的控制器上，无法进行对象端口的指定,仅限 RS-232C可指定。

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 或无指定	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。

SAMPLE

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

格式	主组
ORGORD <表达式>	

格式	副组
ORGORD2 <表达式>	

值	<表达式> 主组：n ~ nnnnnn (n：0 ~ 6) 副组：n ~ nnnn (n：0 ~ 4)
含义	设置进行机器人的返回原点动作及绝对位置搜索动作的轴顺序参数。 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(副组为 1 ~ 4)。 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。

获取

格式	主组
ORGORD	

格式	副组
ORGORD2	

含义	获取进行返回原点动作及绝对位置搜索动作的轴顺序参数。
----	----------------------------

SAMPLE	
A=3	
ORGORD A	指定为第 3 轴先返回原点
ABSRST	主组第 3 轴的返回原点动作结束后, 进行剩余轴的同时返回原点动作, 并执行绝对式原点复位
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	ABSRST, ORIGIN
------	----------------

ORIGIN
对增量式马达轴执行返回原点动作

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。 设置 2 台机器人时，主组返回原点动作完成后，将进行副机器人组的返回原点动作。
SAMPLE	
ORIGIN	 执行增量式马达的返回原点动作
相关命令	ABSRST, ORGORD, ORGORD2, MCHREF, MCHREF2

执行 ORIGIN 命令时，必须将执行级别设置为 0 级以外。
关于执行级别的确认及更改方法,请参阅第 12 章＜3.10 实用模式设置＞及第 14 章＜6 执行级别＞。



要点

- OUT 语句在软件版本 Ver.8.22 以上有效。



注意

- DO, MO, SO 均无法输出至端口 0 和 1。



- 关于指定比特，请参阅第 3 章< 10 指定比特>。

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（进至程序下一行）。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。

< 表达式 > 的值中 10 以下的部分将被舍去（示例：113 → 110）。此外，值为 10 以下时，将被视作指定了 10。

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



要点

- OUTPOS2 仅限系统生成过程中设置了副组时使用。

格式

主组

- OUTPOS <表达式>
- OUTPOS (<轴编号>) = <表达式>

格式

副组

- OUTPOS2 <表达式>
- OUTPOS2 (<轴编号>) = <表达式>

值

<轴编号> 主组：1 ~ 6
副组：1 ~ 4
<表达式> 1 ~ 6144000 (单位：脉冲)

含义

将参数的 OUT 有效位置更改为 <表达式> 中的值。

格式 1 更改组内的所有轴。

格式 2 仅更改 <轴编号> 中指定的轴。



MEMO

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并中止执行。

获取

格式

主组

OUTPOS (<轴编号 1>)

格式

副组

OUTPOS2 (<轴编号 2>)

值

<轴编号 1> 主组：1 ~ 6
<轴编号 2> 副组：1 ~ 4

含义

获取 <表达式> 中指定轴的 OUT 有效位置参数值。



MEMO

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并中止执行。



要点

- OUTPOS2 仅限系统生成过程中设置了副组时使用。

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
```

8

N

O

P

Q

R

S

T

U

V

W

X

Y

Z



要点

- PATH 语句在软件版本 Ver.8.38 以上有效。

格式

PATH	L	, <坐标点指定> , [, 选项 [, 选项] ...]
	C	

含义

设置主机器人轴的 PATH 的移动路径。仅在 PATH SET 命令与 PATH END 命令之间可执行。在上述区间以外执行时，将出错。

- 移动种类：直线插补、圆弧插补
- 坐标点指定：直接数值指定、坐标点定义指定
- 选项：速度指定、坐标平面指定（仅限圆弧插补）、端口输出指定

PATH 移动种类

● 直线插补移动

在 "PATH L..." 中设置直线插补移动。

● 圆弧插补移动

在 "PATH C..." 中设置圆弧插补移动。

在 PATH 移动中，仅指定坐标点的 X、Y、Z 坐标值有效，其他的坐标使用 PATH 移动路径起点的坐标值。

通过反复执行 PATH 命令 ("PATH L"、"PATH C") 可连接移动路径，不停止而直接移动。

坐标点指定的种类

● 直接数值指定

直线插补

圆弧插补

格式

X Y Z R A B [F]

值

X, Y, Z, R, A, B 各轴的坐标值 (用空格隔开)

F 手系统标志 (仅限水平多关节型机器人)

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（带小数点）时，判断为毫米单位。整数与实数同时存在时，所有的坐标值将被作为毫米单位处理。

此形式可指定的移动目标位置坐标点为 1 个。此外，可进行此坐标点指定的移动种类仅限直线插补移动。

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。

此外，在移动路径的起点至终点之间，手系统必须一贯相同。中途无法更改手系统。



注意

- 请使得 PATH 移动中的手系统与 PATH 移动路径的起点中的手系统相同。此外，指定手系统标志移动时，也应使 PATH 移动路径的起点与指定移动目标的手系统一致。手系统不同时，将出错并无法动作。



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

PATH L, 10000 10000 1000 1000 0 0

..... 以脉冲单位设置目标位置进行直线插补移动

PATH L, 150.00 250.00 10.00 30.00 0.00 0.00 1

..... 通过右手系统的指定，设置坐标值的目标位置进行直线插补移动



注意

· 请使得 PATH 移动中的手系统与 PATH 移动路径的起点中的手系统相同。此外，指定手系统标志移动时，也应使 PATH 移动路径的起点与指定移动目标的手系统一致。手系统不同时，将出错并无法动作。



MEMO

● 坐标点定义指定

直线插补

圆弧插补

格式

< 坐标点表达式 > [, < 坐标点表达式 > ...]

含义

< 在坐标点表达式 > 中指定移动目标的坐标点。通过用逗号 (,) 隔开可指定多个坐标点。
移动种类为圆弧插补时，必须对 1 个圆弧指定 2 个坐标点。

· 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

PATH L, P1, P2, P3 设置为向 P1、P2、P3 指定的位置依次进行直线插补移动
PATH C P5, P6, P7, P8 设置为进行由当前位置、P5、P6 及 P6、P7、P8 构成的圆弧插补移动

选项种类

● 速度指定

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> (单位：%) 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 PATH 语句有效。

SAMPLE

PATH L, P5, S=40 按照程序移动速度的 40% 向 P5 指定的位置移动

格式

VEL = <表达式>

值 <表达式>可指定的值范围因机器人而异 (单位：mm / 秒)

含义 使用<表达式> (单位：mm / 秒) 指定移动速度。指定的速度过高时，将出错。

仅对指定的 PATH 语句有效。

SAMPLE

PATH L, P10, VEL=150 按照 150mm / 秒的速度向 P10 指定的位置移动



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

- 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。

● 坐标平面指定

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

在圆弧插补移动中，描画圆弧并指定坐标平面。不指定坐标平面时，进行 3 维圆弧插补移动。

可指定此移动的种类仅限圆弧插补。

仅对指定的 PATH 语句有效。

SAMPLE

PATH C, P1, P2, XY 在 XY 平面内进行圆弧插补移动，Z 轴向 P2 的 Z 轴坐标位置移动

● 端口输出指定

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		



注意

- DO, MO, 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 >。

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, 仅对整数部分有效。

<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

PATH 移动中, 从开始位置仅移动 <表达式 2> 中指定距离的时候, 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。

指定多个比特时, 请从左边开始降序(大→小)记述。

格式 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

PATH SET

PATH L, P1, DO (20) = 1 @ 10 向 P1 进行直线插补移动时, 在开始位置至半径 10 mm 的位置中将 1 输出至 DO (20)

PATH L, P2, DO (21) = 1 @ 12.5 ... 向 P2 进行直线插补移动时, 在 P1 至半径 12.5 mm 的位置中将 1 输出至 DO (21)

PATH END

PATH START

相关命令

PATH SET, PATH END, PATH START

含义

关于 PATH 功能, 请参阅第 9 章 < PATH 语句 >。



要点

- PATH END 语句在软件版本 Ver.8.38 以上有效。

格式

PATH END

含义

结束 PATH 移动的路径设置。
PATH END 命令必须与 PATH SET 命令成对使用。
在 PATH SET 命令至 PATH END 命令之间,最后执行的 PATH 命令 ("PATH L"、"PATH C") 的最终坐标点为 PATH 移动路径的终点。
如果不执行 PATH SET 命令而执行 PATH END 命令, 将出错。

相关命令

PATH, PATH SET, PATH START

参照

关于 PATH 功能, 请参阅第 9 章< PATH 语句>。



要点

- PATH SET 语句在软件版本 Ver.8.38 以上有效。

格式

PATH SET [<坐标点指定>]

含义

开始 PATH 移动的路径设置。

将移动路径的起点设置为<坐标点指定>中指定的位置（仅设置移动路径的起点而并非使机器人实际动作。）省略了<坐标点指定>时，机器人的当前位置将被设置为起点。但是，机器人移动过程中，该移动的目标位置将变为起点。

（示例：PATH SET 命令前面的 MOVE 命令中的 OUT 有效位置范围较广，执行 PATH SET 命令时机器人还在移动。）

PATH SET 命令必须与 PATH END 命令成对使用。

如果执行 PATH SET 命令，以前设置的 PATH 移动路径信息将丢失。

- 坐标点指定：直接数值指定、坐标点定义指定

坐标点指定的种类

● 直接数值指定

格式

X Y Z R A B [F]

值

X, Y, Z, R, A, B 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）

含义

直接指定 PATH 移动路径的起点坐标值。此时的数值为整数时，判断为脉冲单位；数值为实数值（小数点以下 2 位有效）时，判断为毫米单位。

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 PATH 移动路径起点的手系统设置为右手系统。

2: 表示 PATH 移动路径起点的手系统设置为左手系统。



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

- 请使得 PATH 移动中的手系统与 PATH 移动路径的起点中的手系统相同。手系统不同时，将出错并无法动作。



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

PATH SET 120 250.00 55.2 20.33 0 0

..... PATH 移动路径的起点变为以毫米为单位
(120.00 250.00 55.20 20.33 0.00 0.00)

PATH SET -51200 80521 7045 204410 0 0

..... PATH 移动路径的起点以脉冲为单位



注意

· 请使得 PATH 移动中的手系统与 PATH 移动路径的起点中的手系统相同。手系统不同时，将出错并无法动作。



MEMO

● 坐标点定义指定

格式

< 坐标点表达式 >

含义

在 < 坐标点表达式 > 中指定 PATH 移动路径的起点。

- 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

PATH SET P10.....PATH 移动路径的起点为 P10
PATH SET WHERE.....PATH 移动路径的起点为机器人的当前位置

相关命令

PATH, PATH END, PATH START

参照

关于 PATH 功能，请参阅第 9 章 < PATH 语句 >。



要点

- PATH START 语句在软件版本 Ver.8.38 以上有效。

格式

PATH START

含义

开始 PATH 移动。

执行此命令之前，必须事先通过 PATH SET 命令、PATH 命令 ("PATH L"、"PATH C") 及 PATH END 命令设置 PATH 移动路径。此外，开始 PATH 移动时，必须将机器人事先定位到 PATH SET 命令指定的移动路径起点。

机器人的 PATH 移动速度为执行 PATH START 命令时的自动移动速度 (%) 与 SPEED 命令或 PATH 命令的速度由选项 "SPEED" 或 "S" 指定程序移动速度 (%) 的乘积。在 PATH 命令的速度选项 "VEL" 中指定了速度时，则不依靠自动移动速度。

PATH 移动开始后，如果由于机器人到达 PATH 移动路径的终点或联锁等原因停止，将结束本命令。

本命令仅在任务 1 (主任务) 中可执行。

相关命令

PATH, PATH SET, PATH END

参照

关于 PATH 功能，请参阅第 9 章 < PATH 语句 >。

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

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 ~ 19

<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。

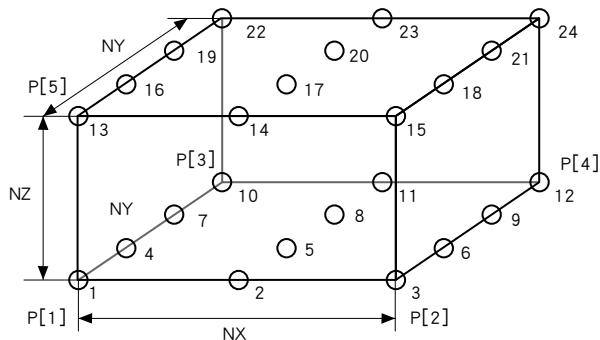
可对已定义的托盘数据更改分割条件。

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

托盘定义中的坐标点数据请使用以下数据区域。

托盘定义	P[1]	P[2]	P[3]	P[4]	P[5]
托盘 0	P3996	P3997	P3998	P3999	P4000
托盘 1	P3991	P3992	P3993	P3994	P3995
:	:	:	:	:	:
托盘 19	P3901	P3902	P3903	P3904	P3905

SAMPLE

PDEF (1) = 3, 4, 2 将托盘定义 1 定义为 3 × 4 × 2



要点

· PMOVE2 仅限系统生成过程中设置了副组时使用。

格式

主组

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

格式

副组

PMOVE2 (<托盘定义编号> , <托盘位置编号>) [, 选项 [, 选项] ...]

值

<托盘定义编号> 0 ~ 19
<托盘位置编号> 1 ~ 32767

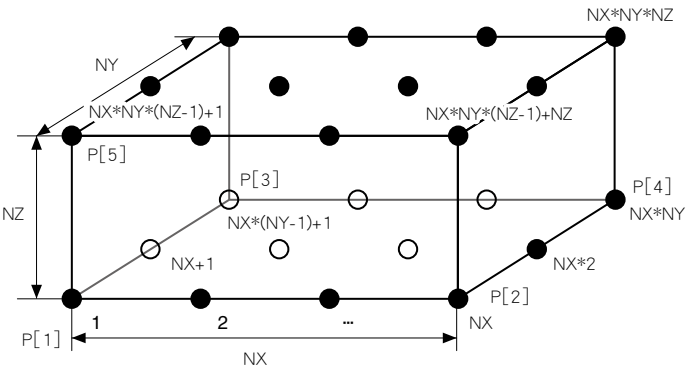
含义

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 轴向每次计算出的位置移动, R 轴向各托盘坐标点数据的 P[1] 中指定的位置移动。

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 / PMOVE2 命令之后的命令注意点

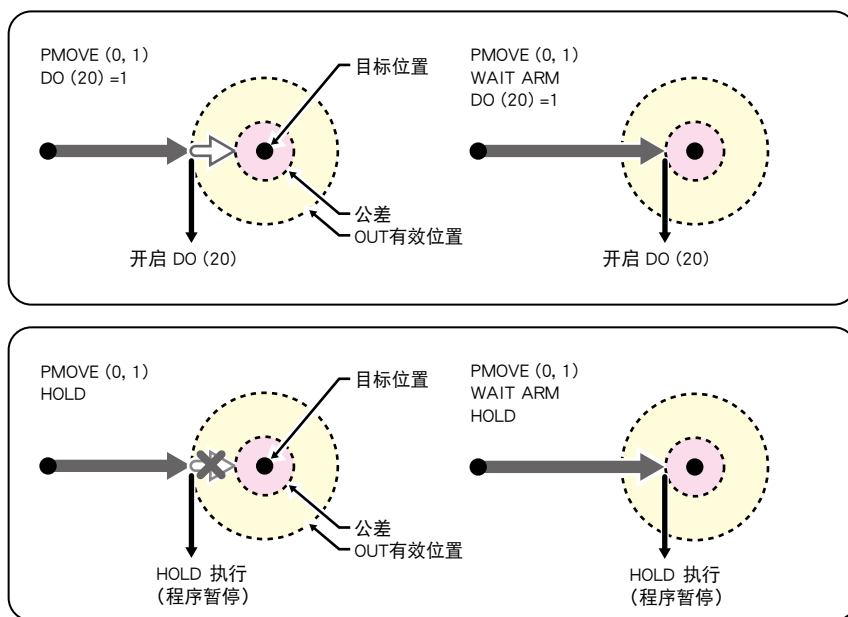
PMOVE / PMOVE2 命令的下一个命令为信号输出命令等可执行命令时，移动对象轴进入 OUT 有效位置范围内后，将开始执行下一个命令。即，在轴到达目标位置的公差范围内之前，将开始下一个命令。

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。
HOLD	进入 OUT 有效范围内后，程序将暂停。因此，移动将停止。
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 / WAIT ARM2" 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 / PMOVE2 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> ...]

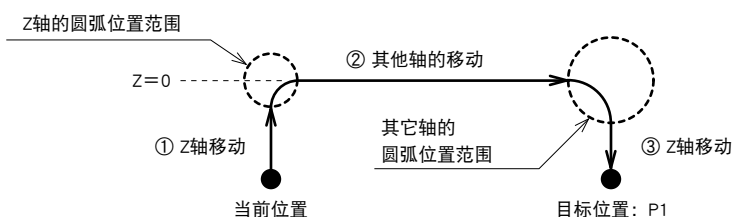
值 x Z, R, A, B 的轴指定
<表达式> 整数值的情况下, 为脉冲单位
实数值 (带有小数点) 的情况下为毫米 / 度单位

- 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



要点

- 此参数在以下软件版本条件下有效。

RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上

- 在比上述版本更旧版本的控制器上，-1 时为“真”，非 -1 时为“假”（不可更改）。



MEMO

- 当 STOPON 条件指定中使用的条件表达式为数值表达式时，变为“真”、“假”的条件可在控制器的 "Other parameters" 中的 "TRUE conditions" 中更改。此条件在 IF、WHILE、WAIT、STOPON 等所有条件表达式中通用。详细说明，请参阅控制器的手册。

- 1) -1 (初始值) 表达式的值为 -1 时，变为“真”；值为 0 时，变为“假”。
值变为 -1、0 以外时，将发生 [6.35: Expression value error] 错误。
- 2) not 0 表达式的值非 0 时，变为“真”；值为 0 时，变为“假”。



要点

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。



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。
- 手系统标志在软件版本 Ver.8.08 以上时有效。



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。

格式

Pn = x y z r a b [f]

值

- n.....坐标点编号：0 ~ 9999
x, y, z, r, a, b.....坐标点数据：根据形式的不同，范围有所差异
f手系统标志：1 或 2 (仅限水平多关节型机器人)

含义

定义坐标点数据。

- n 表示坐标点编号。
- x ~ b 的输入数据用空格(空白行)分隔。
- x ~ b 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，x ~ b 对应第 1 轴~第 6 轴。
- 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。
- x ~ b 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

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。

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, PMOVE2

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。

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

RADDEG
单位转换（弧度→度）

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

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
```

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

79

RESET

关闭指定端口的比特或清除变量

格式 1		
RESET	DOm ([b, . . . , b]) DO (mb, . . . , mb) MOm ([b, . . . , b]) MO (mb, . . . , mb) TO0 ([b, . . . , b]) TO (0b, . . . , 0b) LO0 ([b, . . . , b]) LO (0b, . . . , 0b) SOm ([b, . . . , b]) SO (mb, . . . , mb)	

格式 2	
RESET TCOUNT	

值	m : 端口编号..... 2 ~ 7, 10 ~ 17, 20 ~ 27 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 TCOUNT	清除 10ms 计数器变量

相关命令	SET, DO, MO, SO, TO, LO
------	-------------------------

-  注意
- 无法对 DO, MO, 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 >。

格式

RESTART T_n

值 n : 任务编号.....2 ~ 8

含义 重新启动暂停中 (SUSPEND 状态) 的其他任务。
但是，无法对任务 1 执行 RESTART 命令。

SAMPLE

```
START *SUBTASK,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
  HALT
' SUBTASK ROUTINE
*SUBTASK:
  DO2 (0) =1
  DELAY 1000
  DO2 (0) =0
  DELAY 1000
  GOTO *SUBTASK
EXIT TASK
```

相关命令 CUT, EXIT TASK, START, SUSPEND

参照 详细内容，请参阅多任务。

N

O

P

Q

R

S

T

U

V

W

X

Y

Z



· 详细说明，请参阅 < 60 ON
ERROR GOTO >。

格式

- 1. RESUME NEXT
- 2. RESUME < 标签 >

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。 |



- 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

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\$



要点

- RIGHTY2 仅限系统生成过程中设置了副组时使用。

格式

主组

RIGHTY

格式

副组

RIGHTY2

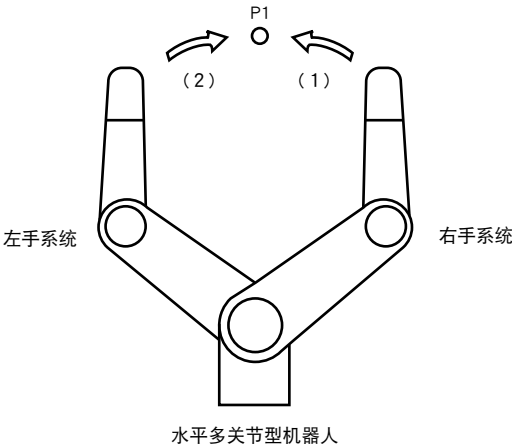
含义

选择使用右手系统使机器人向正交坐标系中指定的坐标点移动。
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 ..... 将机器人设置为右手系统（下图中的 1）  
MOVE P, P1  
RIGHTY ..... 将机器人设置为左手系统（下图中的 2）  
MOVE P, P1  
RIGHTY  
HALT
```

SAMPLE: LEFTY / RIGHTY



35818-R7-00

相关命令

LEFTY, LEFTY2

85

RSHIFT
比特的右位移

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

格式

Sn = x y z r

值 n.....0 ~ 9
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)。

- 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、SHIFT2



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。

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 10 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。



要点

- 坐标点注释的个别文件 (PCn) 在以下软件版本条件下有效。

RCX14x Ver.8.64 以上

RCX22x Ver.9.11 以上

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	ART	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 (_)

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

○: 可以

×: 不可以

MEMO

- 中途停止 SEND 语句并再次启动时，请注意以下事项。
 - 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。
 - 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 10 章<数据文件详情>。



要点

- SERVO2 仅限系统生成过程中设置了副组时使用。



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。请必须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

主组

SERVO	ON OFF FREE PWR	[(< 轴编号 >)]
-------	--------------------------	---------------

格式

副组

SERVO2	ON OFF FREE PWR	[(< 轴编号 >)]
--------	--------------------------	---------------

值 < 轴编号 > 主组：1 ~ 6
副组：1 ~ 4

含义

- 控制指定编号的轴或所有轴的伺服 ON / OFF。
指定了 < 轴编号 > 时，仅各组的指定轴为对象；不指定时，主组和副组中的所有轴皆为对象。此时，马达电源也将同时 ON / OFF。
- 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。
 - PWR 仅只开启马达电源。

- 此命令语句将在所有轴（主组、副组的所有轴）动作结束后（定位到公差范围内后）执行。
- 马达电源关闭时，无法使个别轴伺服使能。

SAMPLE

SERVO ON	在开启马达电源后，所有轴伺服使能
SERVO OFF	关闭马达电源，所有轴伺服禁止 带有制动器的轴将制动并被锁定
SERVO FREE (3)	第 3 轴（Z 轴）伺服禁止，并解除制动

90

SET
开启指定端口的比特

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

值	m : 端口编号..... 2 ~ 7, 10 ~ 17, 20 ~ 27 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, MO, 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

92

SHIFT / SHIFT2

设置位移坐标



要点

- SHIFT2 仅限系统生成过程中设置了副组时使用。



MEMO

格式

主组

SHIFT < 位移变量 >

格式

副组

SHIFT2 < 位移变量 >

含义

使用<位移变量>中指定的位移数据设置位移坐标。

- 此命令语句在轴的动作结束后(定位在公差范围内后)执行。
- 初始值为位移的 S0 (无位移量)。

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

 注意

· 无法输出至 SO0() 与 SO1()。

 参考

· 关于指定比特，请参阅第 3 章 < 10 指定比特 >。

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



要点

- SPEED2 仅限系统生成过程中设置了副组时使用。



要点

- 自动移动速度
通过手持编程器的操作或 ASPEED / ASPEED2 命令设置
- 程序移动速度
由 SPEED / SPEED2 命令或 MOVE / MOVE2、DRIVE / DRIVE2 的 SPEED 选项设置

格式

SPEED <表达式>

格式

SPEED2 <表达式>

值

<表达式> 1 ~ 100(单位：%)

含义

将程序移动速度更改为<表达式>中指定的值。

各组所有轴(机器人构成轴与附加轴中的任意一个)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 / ASPEED2 命令设置的自动移动速度与 SPEED / SPEED2 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%)

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, ASPEED2

格式	
START < 标签 > ,Tn [, P]	
值	n : 任务编号..... 2 ~ 8 p : 任务优先级..... 17 ~ 47
含义	按照<标签>中指定的任务 n、优先级 p 启动。 省略了任务优先级 p 时，为 32。 任务 1 (主任务) 的优先级固定为 32。 数字越小，优先级越高；数字越大，优先级越低 (高 17 ~ 低：47)。 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保持着 READY 状态。
SAMPLE	
<pre>START *SUBTASK,T2, 33 *ST: MOVE P, P0, P1 GOTO *ST HALT 'SUBTASK ROUTINE *SUBTASK: P100 = WHERE IF LOCZ (P100) > 10000 THEN DO (20) = 1 ELSE DO (20) = 0 ENDIF GOTO *SUBTASK EXIT TASK</pre>	
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

B\$=STR\$ (10.01)

相关命令 VAL

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

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	

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。
- 在子过程内无法使用 DECLARE 语句。
- 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, DECLARE, EXIT SUB, SHARED

格式

SUSPEND T_n**值** n : 任务编号.....2 ~ 8**含义** 暂停执行中的其他任务。
也可用于比当前任务优先级更高的任务。
无法指定主任务(任务编号 : 1)。

SAMPLE

```
START *SUBTASK,T2
SUSFLG=0
*L0:
    MOVE P, P0
    MOVE P, P1
    WAIT SUSFLG=1
    SUSPEND T2
    SUSFLG=0
GOTO *L0
HALT
' SUBTASK ROUTINE
*SUBTASK:
    WAIT SUSFLG=0
    DO2 (0) =1
    DELAY 1000
    DO2 (0) =0
    DELAY 1000
    SUSFLG=1
    GOTO *SUBTASK
EXIT TASK
```

相关命令 CUT, EXIT TASK, RESTART, SUSPEND

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

101

SWI
切换执行程序

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。编译异常结束时，停止运行。
在程序名的前后加上 " < " 和 " > "(括号)。
本命令只可在任务 1 (主任务) 中执行。

 MEMO

- 指定程序不存在时，将显示 [3.3: Program doesn't exist (コード: &H0303)] 并停止运行。
- 切换程序时，必须进行编译。编译所需的时间因程序容量而异。
- 如果在编译过程中发生了错误，将显示错误发生行并停止程序。
- 在任务 2 ~ 8 中执行时，将显示 [6.1: Illegal command] 并停止运行。
- 编译过程中，STOP 键无效。

SAMPLE

SWI < ABC > 将执行程序切换至 ABC

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

103

TCOUNTER

定时器、计数器

格式	
TCOUNTER	
含义	从复位 TCOUNTER 变量时开始，每隔 10ms 输出计数值（计数器变量值 1 = 10ms）。值计数至 65,535 后，返回 0。
SAMPLE	
<pre>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</pre>	
相关命令	RESET

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

AS=TIMES\$

PRINT TIMES\$

相关命令

DATES\$, TIMER

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

105

TIMER

获取当前时间



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

TIME\$

格式

1. [LET] TO0 ([b, . . . , b]) = <表达式>
2. [LET] TO (0b, . . . , 0b) = <表达式>

值 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

N

O

P

Q

R

S

T

U

V

W

X

Y

Z



要点

- TOLE2 仅限系统生成过程中设置了副组时使用。



MEMO

格式

主组

1. TOLE <表达式>
2. TOLE (<轴编号>) = <表达式>

格式

副组

1. TOLE2 <表达式>
2. TOLE2 (<轴编号>) = <表达式>

值

<轴编号> 主组：1 ~ 6
副组：1 ~ 4
<表达式> 因设置的马达而异(单位：脉冲)

含义

将公差参数更改为<表达式>中的值(单位：脉冲)。
格式 1：更改各组内的所有轴。
格式 2：仅更改各组<轴编号>中指定的轴。

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并停止程序的执行。
- 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。

获取

格式

主组

TOLE (<轴编号 1 >)

格式

副组

TOLE2 (<轴编号 2 >)

值

<轴编号 1 > 主组：1 ~ 6
<轴编号 2 > 副组：1 ~ 4

含义

赋予<轴编号>中指定轴的公差参数值。



要点

- TOLE2 仅限系统生成过程中设置了副组时使用。

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) ..... 将主组第 2 轴的公差参数赋值给变量 C
HALT
*CHANGE_TOLE:
FOR B=1 TO 4
  TOLE (B) =A
NEXT B
RETURN
```



要点

- TORQUE2 仅限系统生成过程中设置了副组时使用。



注意

- 驱动器版本为 Ver. 2.17 以下时，最大扭矩指令值在 TORQUE / TORQUE2 语句中指定时有效。
- 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。



MEMO

格式

主组

TORQUE (<轴编号>) = <表达式>

格式

副组

TORQUE2 (<轴编号>) = <表达式>

值

<轴编号> 主组：1 ~ 6
副组：1 ~ 4
<表达式> 1 ~ 100 (单位：%)

含义

将各组<轴编号>中指定轴的最大扭矩指令值更改为<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。此命令的最大扭矩指令值在变为以下状态为止有效。

- 对同一个轴执行 TORQUE / TORQUE2 命令。
- 通过 DRIVE / DRIVE2 语句对同一个轴执行扭矩限制选项。
- 切断控制器电源并再次接通电源。
- 更改参数或进行初始化。
- 随着返回原点或绝对式原点复位将执行返回原点动作。
- 变为伺服禁止状态。

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并中止执行。

获取

格式

主组

TORQUE (<轴编号 1>)

格式

副组

TORQUE2 (<轴编号 2>)

值

<轴编号 1> 主组：1 ~ 6
<轴编号 2> 副组：1 ~ 4

含义

赋予<轴编号>中指定轴的扭矩设置值。

SAMPLE	
TRQTIME (3) = 2500	将第 3 轴的扭矩控制超时时间设置为 2.5 秒
DRIVE (3, P1),T = (20, 15)	将最大扭矩值设置为额定扭矩的 20%，将扭矩偏移值设置为 15，使第 3 轴从当前位置移动至 P1 指定的位置（推进动作）
IF TRQSTS (3) = 1 THEN.....	确认是否超时
DO (21) = 1.....	已超时（推进结束。此处将结果输出至 DO (21)）
ELSE	
DO (21) = 0	未超时（到达目标位置，推进不良。此处将结果输出至 DO (21)）
ENDIF	
TORQUE (3) = 100	返回至原来的最大扭矩指令值（100%）
DRIVE (3, P0)	结束扭矩限制及扭矩控制并向 P0 移动
A=TORQUE (2)	将主组第 2 轴的扭矩设置值赋值给变量 A
相关命令 DRIVE, DRIVE2, TRQTIME, TRQTIME2, TRQSTS, TRQSTS2	



要点

- TRQSTS / TRQSTS2 语句在软件版本 Ver.8.45 以上有效。

格式

1. TRQSTS (<轴编号 1>)
2. TRQSTS (<轴编号 2>)

值

<轴编号 1> 主组：1 ~ 6
 <轴编号 2> 副组：1 ~ 4

含义

对<轴编号>中指定的轴执行后，获取带有扭矩限制指定选项的 DRIVE 语句结束时的状态。

- 0 由于扭矩限制超时以外的原因结束
 1 由于扭矩限制超时而结束



MEMO

- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息并中止执行。

SAMPLE

```
DRIVE (3, P1), T=20 ..... 扭矩限制主组的第 3 轴后移动
IF TRQSTS (3) =1 THEN ..... 到达扭矩限制值后由于超时而结束时
GOTO *OK
ELSE ..... 未超时而结束移动时
GOTO *NG
ENDIF
```

相关命令

DRIVE, DRIVE2, TRQTIME, TRQTIME2



要点

- TRQTIME / TRQTIME2 语句
在软件版本 Ver.8.45 以上有效。
- TRQTIME2 仅限系统生成过程中
设置了副组时使用。

格式

主组

TRQTIME (< 轴编号 >) = < 超时时间 >

格式

副组

TRQTIME2 (< 轴编号 >) = < 超时时间 >

值

< 轴编号 > 主组 : 1 ~ 6
副组 : 1 ~ 4
< 超时时间 > 1 ~ 10000 (单位 : ms)

含义

设置使用 DRIVE 语句的扭矩限制指定选项时的电流限制超时时间。在此命令语句中，
将 < 轴编号 > 中指定轴的超时时间设置为 < 超时时间 > 的值。

执行指定了扭矩限制指定选项的 DRIVE / DRIVE2 语句，如果轴到达了目标位置，或者
达到指定扭矩限制值状态超过了 TRQTIME / TRQTIME2 语句中指定的时间(超时时间)，
则该 DRIVE / DRIVE2 命令将结束。

此时，达到指定的扭矩值状态超过了超时时间，在 TRQSTS / TRQSTS2 中设置是否结束
命令的值。

接通控制器电源时，超时时间设置为 1 秒 (1,000ms)。



MEMO

- 超时时间的指定以 ms 为单位，但是在实际动作时以 10ms 为单位。因此，设置值以 10ms 为单
位舍去。但是，设置值为 1 ~ 9 时，视作 10ms。指定了 0 时，视作 1 秒 (1000ms)。
- 如果在生成时指定了设置为 no axis 的轴，则将显示 [5.37: Specification mismatch] 的提示信息
并停止程序的执行。

获取

格式

主组

TRQTIME (< 轴编号 1 >)

格式

副组

TRQTIME2 (< 轴编号 2 >)

值

< 轴编号 1 > 主组 : 1 ~ 6
< 轴编号 2 > 副组 : 1 ~ 4

含义

获取 < 轴编号 > 中指定轴的扭矩限制超时时间。
单位为 ms。



要点

- TRQTIME / TRQTIME2 语句
在软件版本 Ver.8.45 以上有效。
- TRQTIME2 仅限系统生成过程中
设置了副组时使用。

SAMPLE

```

TRQTIME (3) =2500 ..... 将第 3 轴的扭矩控制超时时间设置为 2.5 秒
DRIVE (3, P1), T= (20, 15) ..... 将最大扭矩值设置为额定扭矩的 20%，将扭矩
                                  偏移值设置为 15，使第 3 轴从当前位置移动到
                                  P1 指定的位置（推进动作）
IF TRQSTS (3) =1 THEN ..... 确认是否超时
    DO (21) =1 ..... 已超时（推进结束。此处将结果输出至 DO (21)）
ELSE
    DO (21) =0 ..... 未超时（到达目标位置，推进不良。此处将结果
                    输出至 DO (21)）
ENDIF
TORQUE (3) =100 ..... 返回至原来的最大扭矩值（100%）
DRIVE (3, P0) ..... 结束扭矩限制及扭矩控制并向 P0 移动
A%=TRQTIME (3) ..... 将主组第 3 轴的扭矩限制超时时间赋值给变量 A

```

相关命令 DRIVE, DRIVE2, TRQSTS, TRQSTS2

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。但是,
最小执行等待时间为 10ms。

MEMO



要点

- 此参数在以下软件版本条件下有效。

RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上

- 在比上述版本更旧版本的控制器上, -1 时为“真”, 非 -1 时为“假”(不可更改)。

当条件表达式为数值表达式时, 变为“真”; “假”的条件可在控制器的 "Other parameters" 中的 "TRUE conditions" 中更改。此条件在 IF、WHILE、WAIT、STOPON 等所有条件表达式中通用。
详细说明, 请参阅控制器的手册。

- 1) -1 (初始值) 表达式的值为 -1 时, 变为“真”; 值为 0 时, 变为“假”。
值变为 -1、0 以外时, 将发生 [6.35: Expression value error] 错误。
- 2) not 0 表达式的值非 0 时, 变为“真”; 值为 0 时, 变为“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, DRIVE2, DRIVEI, DRIVEI2, MOVE, MOVE2, MOVEI, MOVEI2



要点

- WAIT ARM2 仅限系统生成过程中设置了副组时使用。

格式		主组
WAIT ARM [(< 轴 编号 >)]		
格式		副组
WAIT ARM2 [(< 轴 编号 >)]		
值	< 轴编号 > 主组：1 ～ 6 副组：1 ～ 4	
含义	等待机器人轴动作结束(定位于在公差范围内)。 < 轴编号 > 中有指定时，组内指定轴将变为对象；未指定时，组内的所有轴变为对象。	
SAMPLE		
WAIT ARM 等待主机器人动作结束		
WAIT ARM2 (2)..... 等待副机器人第 2 轴动作结束		
相关命令	DRIVE, DRIVE2, DRIVEI, DRIVEI2, MOVE, MOVE2, MOVEI, MOVEI2	



要点

- WEIGHT2 仅限系统生成过程中设置了副组时使用。

格式

主组

WEIGHT <表达式>

格式

副组

WEIGHT2 <表达式>

值

<表达式> 范围因所设置的机器人而异。

含义

将主机器人或副机器人的前端重量参数更改为<表达式>的值。对附加轴无任何影响。

获取

格式

主组

WEIGHT

格式

副组

WEIGHT2

含义

获取机器人的前端重量参数值。

SAMPLE

```

A=5
B=2
C=WEIGHT
WEIGHT A
MOVE P, P0
WEIGHT B
MOVE P, P1
WEIGHT C
D=WEIGHT ..... 将主机器人的前端重量参数赋值给变量 D
HALT

```



要点

- WEIGHT2 仅限系统生成过程中设置了副组时使用。

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

主组

WHERE

格式

副组

WHERE2

含义

通过关节坐标获取机械臂的当前位置。

SAMPLE

P10=WHERE 将当前位置的脉冲坐标值赋值给 P10

相关命令

WHRXY, WHRXY2

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

<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



要点

- 此参数在以下软件版本条件下有效。

RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上

- 在比上述版本更旧版本的控制器上，-1 时为“真”，非 -1 时为“假”（不可更改）。

- 当条件表达式为数值表达式时，变为“真”、“假”的条件可在控制器的 "Other parameters" 中的 "TRUE conditions" 中更改。此条件在 IF、WHILE、WAIT、STOPON 等所有条件表达式中通用。详细说明，请参阅控制器的手册。

- 1) -1 (初始值) 表达式的值为 -1 时，变为“真”；值为 0 时，变为“假”。
值变为 -1、0 以外时，将发生 [6.35: Expression value error] 错误。
- 2) not 0 表达式的值非 0 时，变为“真”；值为 0 时，变为“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 ..... 条件表达式经常为真 (-1)，变为永久循环。
  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
```



要点

- WHRXY / WHRXY2 在以下软件版本条件下有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

格式

主组

WHRXY

格式

副组

WHRXY2

含义

通过正交坐标获取机械臂的当前位置。

SAMPLE

P10=WHRXY..... 将当前位置的正交坐标值赋值给 P10

相关命令

WHERE, WHERE2

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

119

XYTOJ / XYTOJ2

将主组轴的正交坐标数据（毫米）转换为关节坐标数据（脉冲）

格式

主组

XYTOJ（<坐标点表达式>）

格式

副组

XYTOJ2（<坐标点表达式>）

含义

将<坐标点表达式>中显示的正交坐标数据（单位:mm、度）转换为关节坐标数据（单位：脉冲）。

- 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。
- 将关节坐标数据转换为正交坐标数据时，使用 JTOXY / JTOXY2。

SAMPLE

P10=XYTOJ (P10)..... 将 P10 转换为关节坐标数据

相关命令

JTOXY2

120

_SYSFLG

用于监视轴状态的标志

格式

_SYSFLG

含义

根据 SYSFLG 变量设置的值，用作监视轴状态的标志。

SAMPLE

_SYSFLG = 1

相关命令

RESET

第 9 章

PATH 语句

1	概要	9-1
2	特点	9-1
3	使用方法.....	9-1
4	注意事项.....	9-2

1 概要

PATH 功能是按照指定速度在由直线和圆弧构成的路径上移动的功能。由于移动中的速度变化较少，最适合涂胶等用途。

2 特点

- 在整个移动路径上保持恒定速度动作。（从停止状态开始的加速区间及动作结束前的减速区间除外）
- 由于速度不会因为坐标点示教位置的疏密而发生变化，所以坐标点示教很简单。
- 除了可更改整个移动路径的移动速度之外，还可通过速度选项仅更改部分路径的速度。
- 使用 DO 选项可在移动中的任意区间向指定端口发送信号。

3 使用方法

若要使用 PATH 功能，必须设置使用以下机器人语言。

- PATH SET开始路径设置
- PATH (PATH L, PATH C).....设置路径
- PATH END结束路径设置
- PATH START.....实际开始移动

如下所示，在 PATH SET 语句及 PATH END 语句之间的内容，使用 PATH 语句设置动作路径。此时，机器人不动作。

当路径设置结束后执行 PATH START 语句时，将执行已设置的动作路径移动。

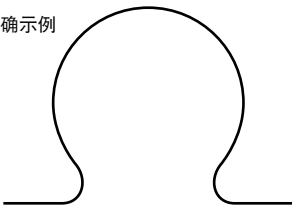
SAMPLE

```
MOVE P, P0, Z=0
PATH SET ..... 开始路径设置
PATH L, P1, DO (20) = 1 @ 10.0
PATH L, P2
.
.
.
PATH C, P12, P13
PATH L, P14, DO (20) = 0 @ 20.0
PATH END ..... 结束路径设置
MOVE P, P1, Z=0
.
.
.
MOVE P, P0, Z=0
PATH START ..... 执行 PATH 移动
HALT
```

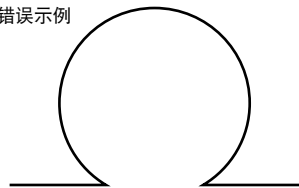
- 直线与圆弧路径的总数合计最多 300 条。
- 执行 PATH 移动 (执行 PATH START 语句) 时, 必须将机器人定位至已设置路径的起点。
- 圆弧与直线连接的点, 请尽量使其移动方向一致。方向发生较大变化时, 可能会发生振动或机器人出错。

连接圆弧与直线的点: 方向发生大变化时

正确示例



错误示例

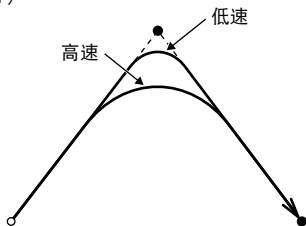


35901-R7-00

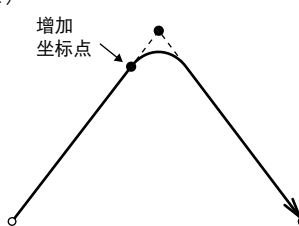
- 直线与直线连接的点, 请使移动时的路径从连接点的内侧通过。此外, 速度较快时也从内侧通过 (1)。不想由于速度而使路径发生较大变化时, 请如 (2) 所示增加坐标点。但是, 请注意, 如果不降低速度, 可能会出错。

直线与直线连接的点: 抑制路径的变化

(1)



(2)



35902-R7-00

- 当由于不按照指定速度移动而出错时
PATH 移动开始时、停止时及移动路径中途改变速度指定时, 机器人将进行加速或减速。此时, 由于坐标点之间的距离较短而无法达到指定速度或无法减速时, 将在动作开始前出错。在上述情况下, 必须降低指定速度。即使降低速度也出错时, 请调整坐标点, 使包含加减速区间的直线或圆弧延长。
- 请使 PATH 移动中的手系统与 PATH 移动路径起点时的手系统相同。向设置了手系统标志的坐标点进行 PATH 移动时, 也同样。手系统不同时, 将出错并无法动作。
- PATH 移动中当机器人由于联锁等原因停止时, 将被视作执行结束, 即使重新启动也无法进行剩余的路径移动。

请仔细阅读各命令的注意事项。

第 10 章

数据文件详细说明

1	概要	10-1
2	程序文件.....	10-2
3	坐标点文件.....	10-4
4	坐标点注释文件	10-6
5	参数文件.....	10-8
6	位移坐标定义文件	10-11
7	机械手定义文件	10-13
8	托盘定义文件.....	10-15
9	ALL 文件.....	10-19
10	程序目录文件.....	10-20
11	参数目录文件.....	10-22
12	变量文件.....	10-23
13	常量文件.....	10-26
14	数组变量文件.....	10-27
15	DI 文件	10-29
16	DO 文件.....	10-31
17	MO 文件	10-33
18	LO 文件	10-35

19	TO 文件	10-37
20	SI 文件	10-39
21	SO 文件	10-41
22	错误提示履历文件	10-43
23	机器参照文件	10-44
24	EOF 文件	10-45
25	串行端口通信文件	10-46
26	SIW 文件	10-47
27	SOW 文件	10-49
28	Ethernet 端口通信文件	10-51

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 27 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. 错误提示履历文件
22. 机器参照文件
23. EOF 文件
24. 串行端口通信文件
25. SIW 文件
26. SOW 文件
27. Ethernet 端口通信文件

**要点**

- “27. Ethernet 端口通信文件”
在软件版本 Ver.8.22 以上时
有效。

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 75 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

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。

DATA FORMAT

NAME=< 程序名 > [cr/lf]
aaaaa ...aaaaaaaaaaaaa [cr/lf]
:
aaaaa ...aaaaaaaaaaaaa [cr/lf]
:
NAME=< 程序名 > [cr/lf]
aaaaa ...aaaaaaaaaaaaa [cr/lf]
:
aaaaa ...aaaaaaaaaaaaa [cr/lf]
[cr/lf]

值

a..... 字符编码

- < 程序名 > 采用 8 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。
- TAB 代码 (09H) 转换为空格。

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]
A=1 [cr/lf]
RESET DO2 () [cr/lf]
:
HALT [cr/lf]
[cr/lf]

格式

< 程序名 >

- 含义
- 表示指定程序。
 - 程序名采用 8 字符以内的英文字母数字或“_”表示，并在外面加上“<”和“>”。
 - 不指定程序名时，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]
aaaaa ...aaaaaaaaaaaaaa [cr/lf]
      :
aaaaa ...aaaaaaaaaaaaaa [cr/lf]
[cr/lf]
```

值 a..... 字符编码

- 程序名采用 8 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。
- TAB 代码转换为空格。

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。
- 当前模式设置为自动模式或程序模式时，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

```
Pmmmm=xxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
Pmmmm=xxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
:
Pmmmm=xxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
Pmmmm=xxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
[cr/lf]
```



要点

- 在软件版本为 Ver.8.28 以前的控制器上，mmmm 为 0 ~ 4000。
- 手系统标志在软件版本 Ver.8.08 以上时有效。
- 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 ~ 9999
f 坐标点符号：+/- 空格
xxxxxx/yyyyy/./bbbbbb 8 位以内的数值。数值中带有点时，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。

- 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)。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

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
```

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]
:
P9999= -1.00        0.00        0.00        0.00        0.00        0.00        [cr/lf]
[cr/lf]
```



要点

- 在软件版本为 Ver.8.28 以前的控制器上，mmmm 为 0 ~ 4000。



要点

- 手系统标志在软件版本 Ver.8.08 以上时有效。
- 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。

格式									
Pmmmm									
含义	· 表示指定坐标点。 · mmmm 通过 0 ~ 9999 表示。								
DATA FORMAT									
Pmmmm=fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]									
值	<table><tr><td>mmmm</td><td>坐标点编号 : 0 ~ 9999</td></tr><tr><td>f</td><td>坐标点符号 : +/- 空格</td></tr><tr><td>xxxxxx/yyyyy/./bbbbbb</td><td>8 位数以内的数值。数值中带有点时，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。</td></tr><tr><td>t</td><td>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。</td></tr></table> ■ 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)。	mmmm	坐标点编号 : 0 ~ 9999	f	坐标点符号 : +/- 空格	xxxxxx/yyyyy/./bbbbbb	8 位数以内的数值。数值中带有点时，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。
mmmm	坐标点编号 : 0 ~ 9999								
f	坐标点符号 : +/- 空格								
xxxxxx/yyyyy/./bbbbbb	8 位数以内的数值。数值中带有点时，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。								
SAMPLE									
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]									

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

PCmmmm= sssssssssssss [cr/lf]
PCmmmm= sssssssssssss [cr/lf]
:
PCmmmm= sssssssssssss [cr/lf]
PCmmmm= sssssssssssss [cr/lf]
[cr/lf]



要点

- 在软件版本为 Ver.8.28 以前的控制器上，mmmm 为 0 ~ 4000。

值

mmmm 坐标点编号：0 ~ 9999
ss...ss 注释数据：仅可使用半角、最多 15 个字符。写入数据时超过 15 个字符时，第 16 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]



要点

· 坐标点注释单位在以下软件版本条件下有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

格式	
PCmmmm	
含义	· 表示指定坐标点注释。 · mmmm 通过 0 ~ 9999 表示。
DATA FORMAT	
PCmmmm= ssssssssssssss [cr/lf]	
值	mmmm 坐标点编号 : 0 ~ 9999 ss...ss 注释数据 : 仅可使用半角、最多 15 个字符。写入数据时超过 15 个字符时, 第 16 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]	

9

5.1 所有参数

10

格式

11

PRM

- 12
- 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。

13

DATA FORMAT

14

```
/ 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]
A1= xxxxxx A2= yyyyyy A3= zzzzzz A4= rrrrrr [cr/lf]
/ 参数标签 / ' <注释> [cr/lf]
:
[cr/lf]
```

- 15
- 值
- RC 表示所有控制器。
 - R? 机器人指定: ? / 1: 主机器人、2: 副机器人
 - A? 轴指定: ? : 轴编号
 - <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。

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至旧版本的控制器, 可能会发生 [10.14: Undefined parameter found] 的错误。在上述情况下, 如果在 "Other parameters" 的设置中将 "Skip undefined parameters" 设置为有效, 有时可正常载入未定义参数以外的参数 (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：

/RBTNUM/ 'Robot number (V8.01 / R1001) [cr/lf]

R1= 3000 R2= 3010 [cr/lf]

/AXES / 'Number of axes [cr/lf]

R1= 2 R2= 2 [cr/lf]

/AXSNUM/ 'Axis number (V1.01/V1.01/V1.01/V1.01/-----/-----/-----/-----/) [cr/lf]

A1= 5000 A2= 5001 A3= 5010 A4= 5011 [cr/lf]

A5= 0 A6= 0 A7= 0 A8= 0 [cr/lf]

/ATTRIB/ 'Axis attribute [cr/lf]

A1= 33792 A2= 33792 A3= 33792 A4= 33792 [cr/lf]

A5= 256 A6= 256 A7= 256 A8= 256 [cr/lf]

/WEIGHT/ 'Tip weight [kg] [cr/lf]

R1= 2 R2= 12 [cr/lf]

:

/CURPNO/ 'Port number of output [cr/lf]

RC= 20 [cr/lf]

/CURPT1/ 'Compare point number 1 [cr/lf]

RC= 0 [cr/lf]

/CURPT2/ 'Compare point number 2 [cr/lf]

RC= 0 [cr/lf]

[cr/lf]

格式

/ 参数级别 /

- 含义
- 参数标签采用 6 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]
A?= xxxxxx [cr/lf]
[cr/lf]

- 值
- RC 表示所有控制器。
 - R?..... 机器人指定：? / 1：主机器人、2：副机器人
 - A? 轴指定：?：轴编号
 - <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。

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 / 'Accel coefficient [%]
A1= 100 A2= 100 A3= 100 A4= 100 [cr/lf]
[cr/lf]

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/yyyyyy/./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]
      :
SM9 =      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：0～9 f.....坐标点符号：+ / - / 空格 xxxxxx/yyyyyy/./rrrrrr.....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 = fxxxxxx fyzzzzz {R} [cr/lf]
:
Hm = fxxxxxx fyzzzzz {R} [cr/lf]
[cr/lf]
```

值

m..... 机械手编号 / 0 ~ 3 : 主机器人使用
4 ~ 7 : 副机器人使用

f..... 坐标点符号 : + / - / 空格

xxxxxx/yyzzzz 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 =      0.00      0.00      0.00 [cr/lf]
H1 =      1.00      1.00      1.00 [cr/lf]
H2 =      2.00      2.00      2.00 [cr/lf]
H3 =      3.00      3.00      3.00 [cr/lf]
H4 =      4.00      4.00      4.00 [cr/lf]
H5 =      5.00      5.00      5.00 [cr/lf]
H6 =      6.00      6.00      6.00 [cr/lf]
H7 =      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 = fxxxxxx fyyyyyy fzzzzzz {R}[cr/lf]

值

m	机械手编号 / 0 ~ 3 / 主机器人使用 4 ~ 7 : 副机器人使用
f	坐标点符号 : + / - / 空格
xxxxxx/yyyyyy/zzzzzz	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 = 3.00 3.00 3.00 [cr/lf]

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

```
PLm [cr/lf]
PLN = XY [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [cr/lf]
P [1] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
      :
P [5] = fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
PLm [cr/lf]
      :
[cr/lf]
```

值

m..... 托盘编号：0～19
 nnn..... 各轴的坐标点数：正整数
 f..... 坐标点符号：+ / - / 空格
 xxxxxx/yyyyy/./bbbbbb..... 8 位数以内的数值。数值中带有点时，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。



要点

- 手系统标志在软件版本 Ver.8.08 以上时有效。

- 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。
- 使用托盘定义的动作中忽略手系统标志。
- 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)。
- 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]

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]

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 ~ 19 表示。

DATA FORMAT
PLm [cr/lf]
PLN = XY [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [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]



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 ~ 19
nnn	各轴的坐标点数：正整数
f	坐标点符号：+ / - / 空格
xxxxxx/yyyyy/./bbbbbb	8 位数以内的数值。数值中带有点时，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。



要点

- 手系统标志在软件版本 Ver.8.08 以上时有效。

- 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。
- 使用托盘定义的动作中忽略手系统标志。
- 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)。
- 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]

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]

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= sssssssssssss [cr/lf]

:

PCmmmm= sssssssssssss [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 程序目录文件

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	如果对程序进行了编译，且创建了程序对象时为 "o" 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：8 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	fgssssssss	llll	bbbbbb	xx	yy/mm/dd	hh:mm [cr/lf]

值	nnn	程序目录的编号：3 位数
	f	如果对程序进行了编译，且创建了程序对象时为 "o" 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：8 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]
3o*	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]

12.1 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16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: 70 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 16 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: 70 个字符以下的字符串

MEMO

· 动态全局变量在程序编译时录入。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 常量文件

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：61 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 16 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: 70 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 16 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 : 70 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 () =&Bnnnnnnnn [cr/lf]					
值	<table><tr><td>m.....</td><td>0 ~ 7、10 ~ 17、20 ~ 27</td></tr><tr><td>n.....</td><td>0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)。</td></tr></table>	m.....	0 ~ 7、10 ~ 17、20 ~ 27	n.....	0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)。
m.....	0 ~ 7、10 ~ 17、20 ~ 27				
n.....	0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (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]

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]
:
MO27 ( ) = &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]
:
MO26 ( ) = &B00000000 [cr/lf]
MO27 ( ) = &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 MO0 与 MO1()。

- 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
 - 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 MO0 () 与 MO1 ()。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]

18.1 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]
[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]
[cr/lf]

格式	
LO0 ()	
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	
LO0 () =&Bnnnnnnnn [cr/lf]	
· 写入文件	
DATA FORMAT	
&Bnnnnnnnn [cr/lf] 或 k [cr/lf]	
值	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]	

19.1 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]
[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]
[cr/lf]

格式	
TO ()	
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	
TO0 () =&Bnnnnnnnn [cr/lf]	
· 写入文件	
DATA FORMAT	
&Bnnnnnnnn [cr/lf] 或 k [cr/lf]	
值	n.....0 或 1 (合计 8 位数)。从左开始对应 07, 06, ..., 00。 k0 ~ 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 所有错误提示履历

格式
LOG

- 含义
- 表示所有错误提示履历。
 - 对于读出文件，将读出所有错误提示履历。
 - 无法在写入文件中指定。

DATA FORMAT
nnn:yy/mm/dd, hh:mm:ss gg.bb:msg [cr/lf]
nnn:yy/mm/dd, hh:mm:ss gg.bb:msg [cr/lf]
:
nnn:yy/mm/dd, hh:mm:ss gg.bb:msg [cr/lf]
[cr/lf]

- 值
- | | |
|-----------------|--------------------|
| nnn | 错误履历的序列号(最多 500 号) |
| yy, mm ,dd..... | 年、月、日 |
| hh, mm, ss..... | 时、分、秒 |
| gg | 错误提示组 |
| bb..... | 错误提示分类 |
| msg..... | 错误提示 |

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE
SEND LOG TO CMU 从通信端口输出所有错误提示履历
应答 :
1 :01/06/14,13:19:20 14.22:No start code (@) [cr/lf]
2 :01/06/14,13:18:34 22.3: DC24V power low [cr/lf]
:
498:01/06/12,21:49:54 5.39:Illegal identifier [cr/lf]
499:01/06/12,21:49:14 14.22:No start code(@)[cr/lf]
500:01/06/12,21:49:00 22.3: DC24V power low [cr/lf]
[cr/lf]

23.1 所有机器参照文件

格式

MRF

含义

- 表示机器人返回原点动作及绝对位置搜索动作的机器参照数据。
- 对于读出文件，将读出所有机器参照数据。
- 无法在写入文件中指定。

DATA FORMAT

```
M1=nnn% M2=nnn% ... nnn% [cr/lf]
S1=nnn% S2=nnn% ... nnn% [cr/lf]
[cr/lf]
```

值

nnn 0 ~ 100 的整数
M? 主机器人组的轴
S? 副机器人组的轴

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

```
SEND MRF TO CMU ..... 从通信端口输出所有机器参照
```

应答：

```
M1= 55% M2= 23% M3= 33% M4= 26% [cr/lf]
[cr/lf]
```

24.1 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。

25

串行端口通信文件

25.1

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

26.1 所有 SIW



要点

- SIW 在软件版本 Ver.8.08 以上有效。

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)



要点

· SIW 在软件版本 Ver.8.08 以上有效。

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]

**要点**

- SOW 在软件版本 Ver.8.08 以上有效。

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]
```

8

9

10

11

12

13

14

15



要点

· SOW 在软件版本 Ver.8.08 以上有效。

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]

28.1 Ethernet 端口通信文件

格式

ETH



要点

- ETH 在软件版本 Ver.8.22 以上有效。

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 11 章

用户程序示例

1	基本篇	11-1
2	应用篇	11-8

■ 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						

8

9

10

11

12

13

14

15

1.2使用坐标点编号

■ 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

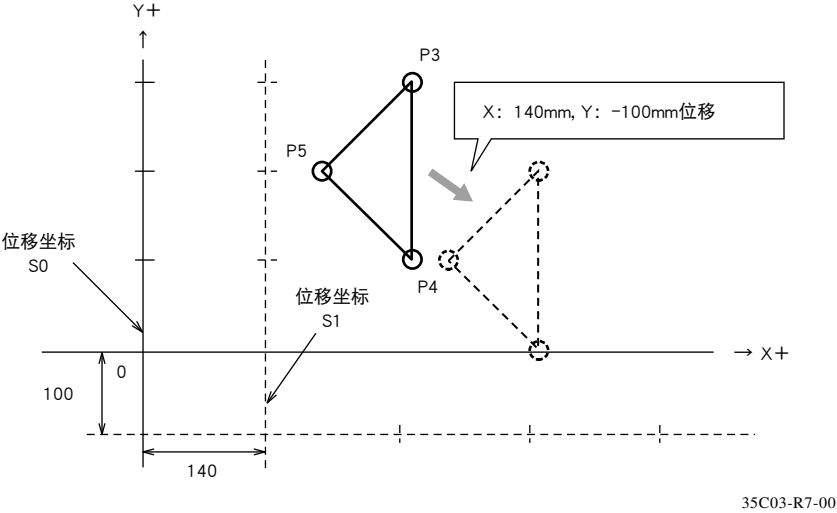
SAMPLE 2
FOR J=0 TO 5
MOVE P, P [J]
NEXT J
HALT

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



```
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
```

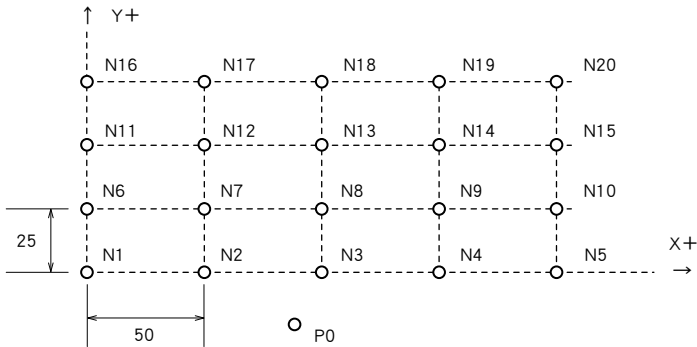
1.4.1 算出坐标点坐标

■ 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。

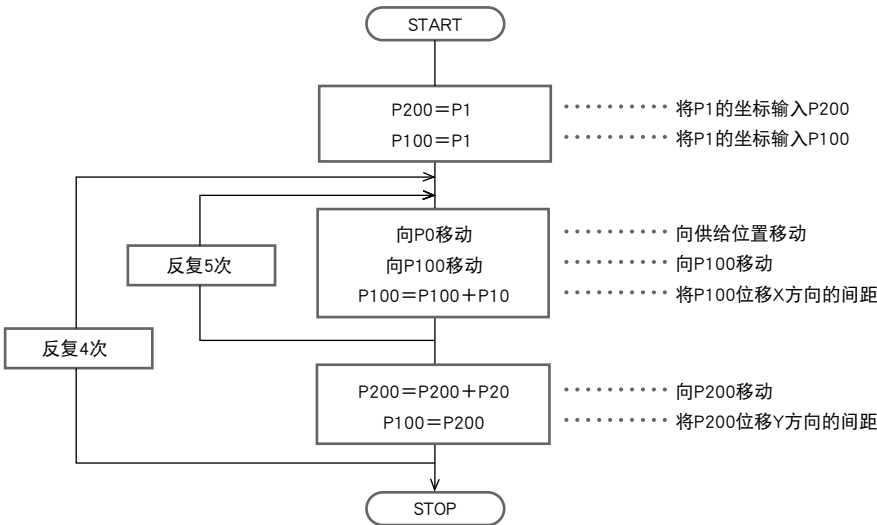
POINT DATA						
工件供给位置：						
P0=	0.0	0.0	0.0	0.0	0.0	0.0
X 方向的间距：						
P10=	50.0	0.0	0.0	0.0	0.0	0.0
Y 方向的间距：						
P20=	0.0	25.0	0.0	0.0	0.0	0.0
N1 的位置：						
P1 =	100.0	50.0	0.0	0.0	0.0	0.0

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
```

8

9

10

11

12

13

14

15

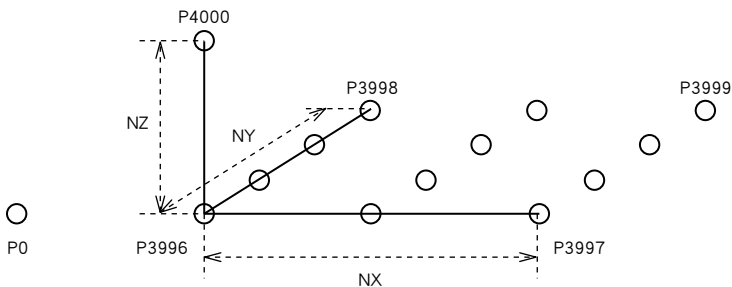
1.4.2 使用托盘移动

■ 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。

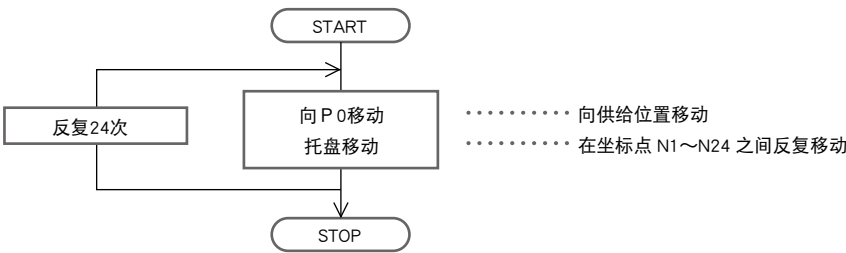
POINT DATA						
工件供给位置：						
P0=	0.00	0.00	100.00	0.00	0.00	0.00
托盘定义：						
PL0 (使用的坐标点为 P3996 ~ P4000)						
NX=	3					
NY=	4					
NZ=	2					
P3996=	100.00	50.00	100.00	0.00	0.00	0.00
P3997=	200.00	50.00	100.00	0.00	0.00	0.00
P3998=	100.00	200.00	100.00	0.00	0.00	0.00
P3999=	200.00	200.00	100.00	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

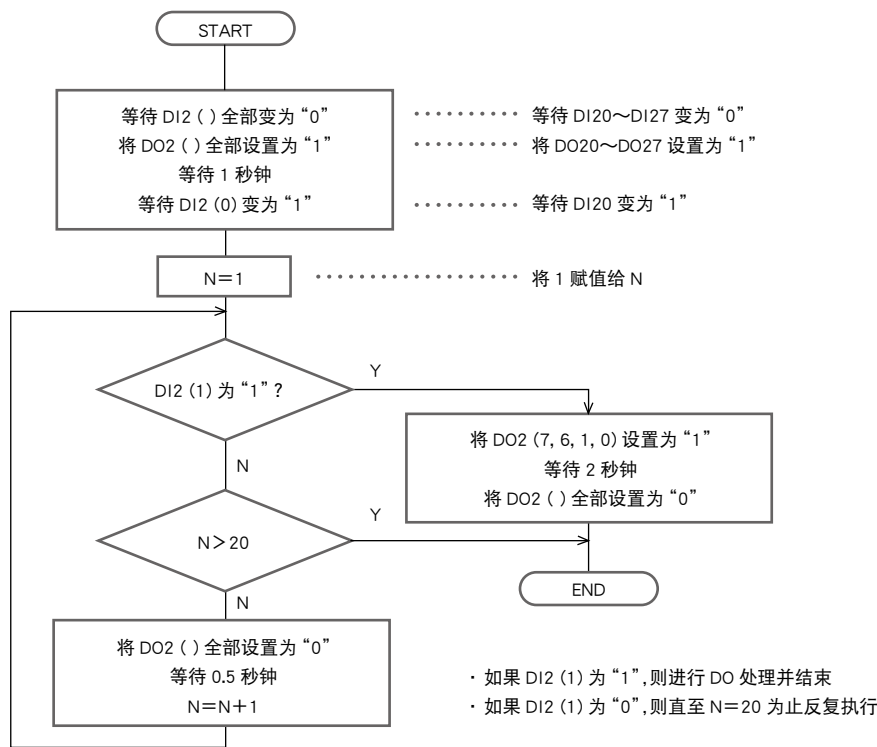


35C07-R7-00

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
```

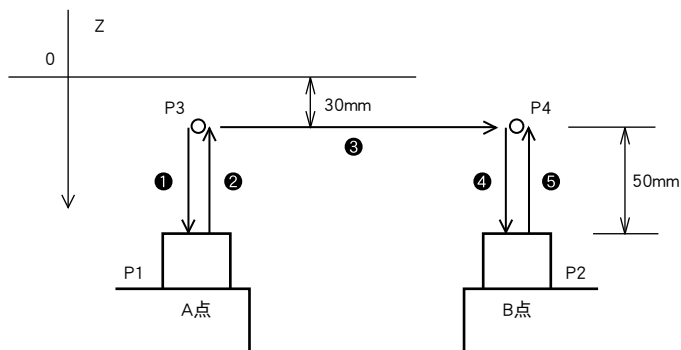
2.1

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
LOCZ (P3) =LOCZ (P3) -50.0 ..... 将 P3 朝着 Z 的上方向位移 50mm
LOCZ (P4) =LOCZ (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
LOCZ (P4) = LOCZ (P4)-50.0..... 将 P4 朝着 Z 的上方向位移 50mm
GOSUB *OPEN
MOVE P, P1, Z=30.0..... 以 Z = 30mm 进行圆弧插补移动
GOSUB *CLOSE
MOVE P, P2, Z=30.0..... 以 Z = 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
```

8

9

10

11

12

13

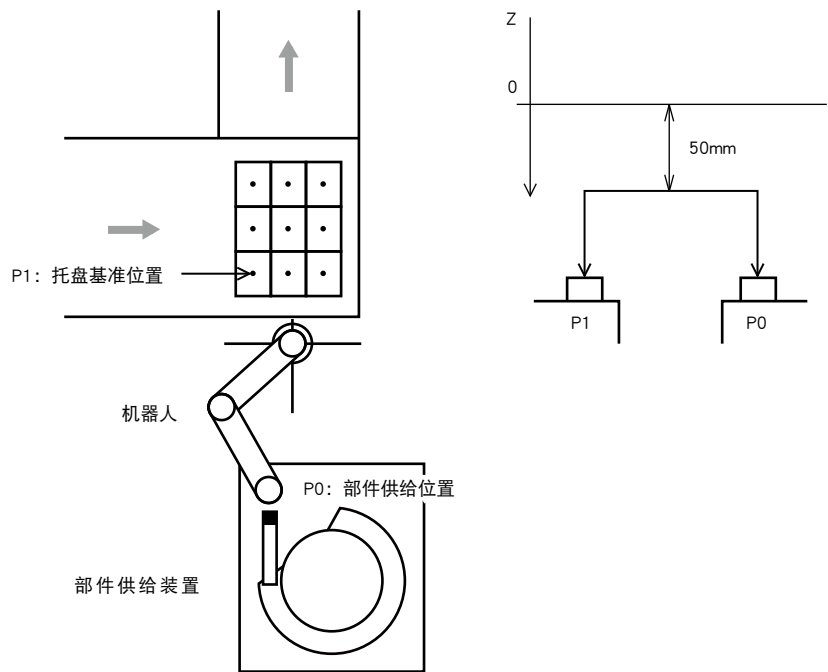
14

15

■ 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, Z=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, Z=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

```

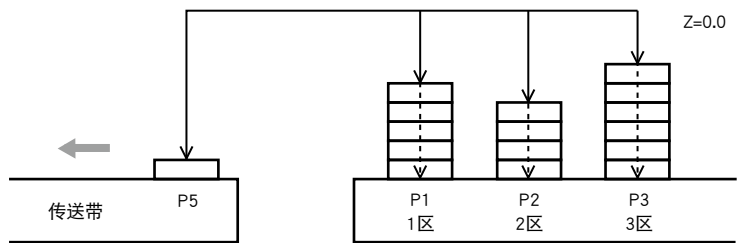
WHILE -1 ..... 反复执行整体
FOR A=1 TO 9
    WAIT DI (31) =1 ..... 等待直至托盘变为“有”
    WAIT DI (30) =1 ..... 等待直至供给部件变为“有”
    DO (30) =0 ..... 将机器人机械手“开启”
    DELAY 100
    MOVE P, P0, Z=50.0 ..... 移动至供给位置
    DO (30) =1 ..... 将机器人机械手“关闭”
    DELAY 100
    PMOVE (0, A), Z=50.0 ..... 移动至托盘上的坐标点
    DO (30) =0 ..... 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

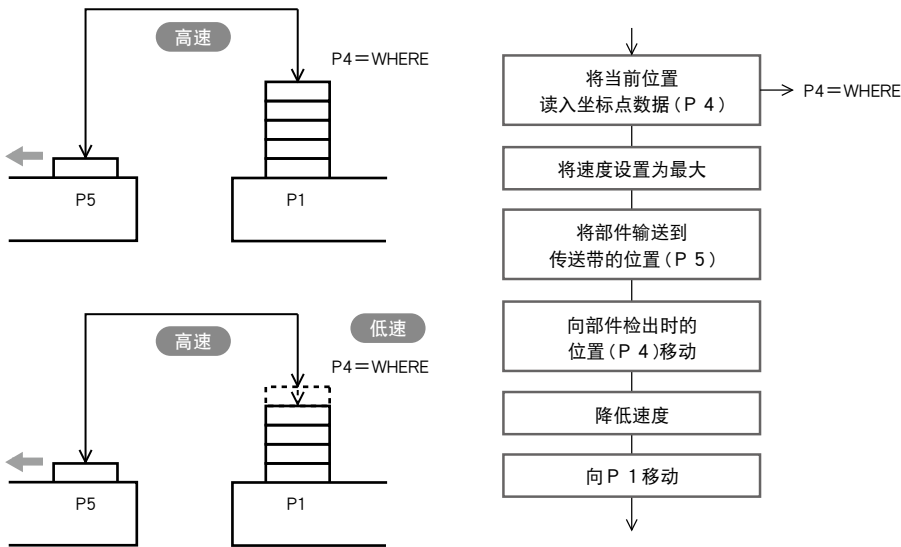
· 机器人机械手开关时，等待时间为 0.1 秒。

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]
LOCZ (P6) =0.00
MOVE P, P6, Z=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, Z=0.0
    GOSUB *OPEN
    MOVE P, P4, Z=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
```

8

9

10

11

12

13

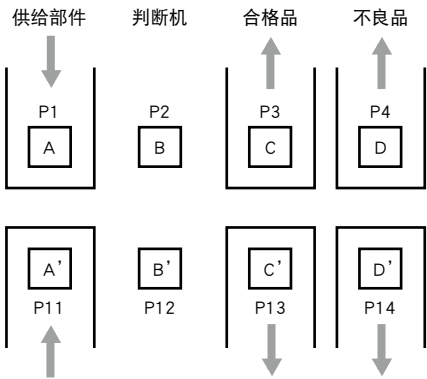
14

15

■ 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）

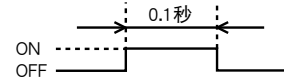


35C13-R7-00



要点

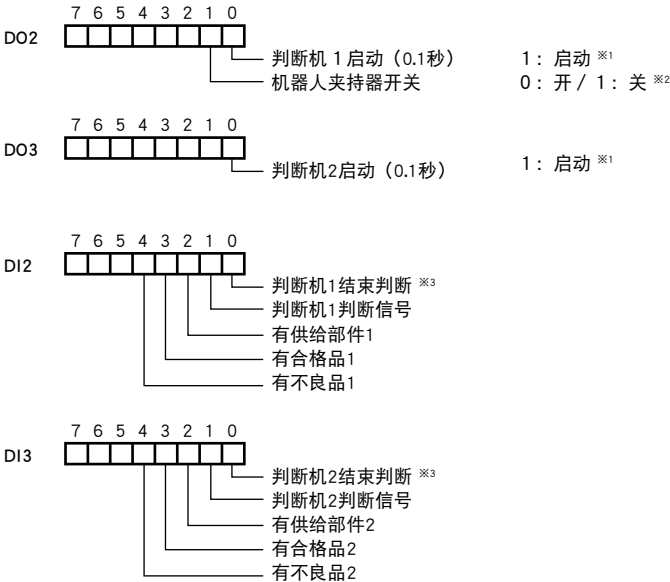
- ※1：启动信号向判断机提供 0.1 秒的脉冲信号。
- ※2：夹持器开关时，等待时间为 0.1 秒。
- ※3：从判断机获得的信号中，有结束及合格品 / 不良品的判断信号。判断结束后，结束信号为 ON (= 1)，当判断信号为合格品时将变为 ON (= 1)，当判断信号为不良品将变为 OFF (= 0)。



■ 前提

1. 输入输出信号

输入输出信号



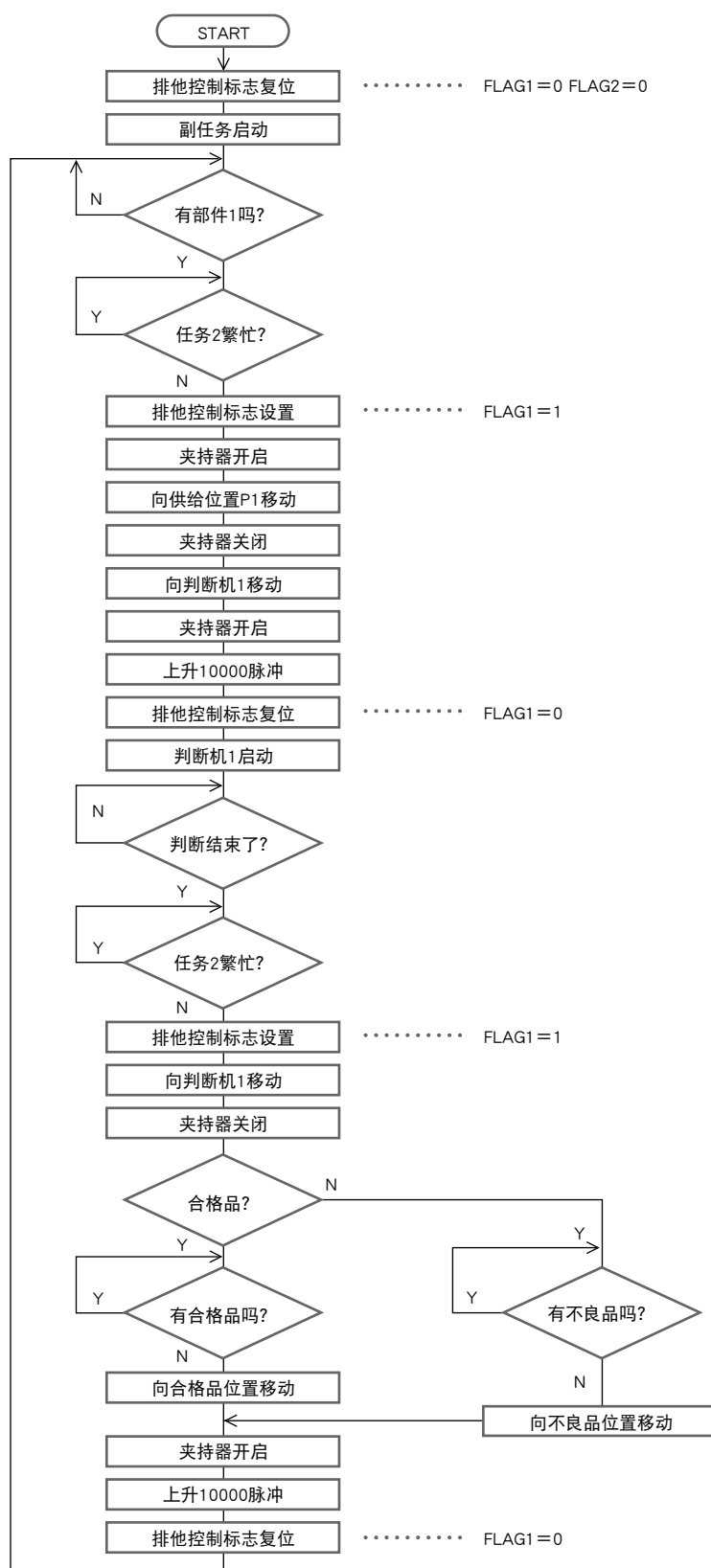
35C14-R7-00

- 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

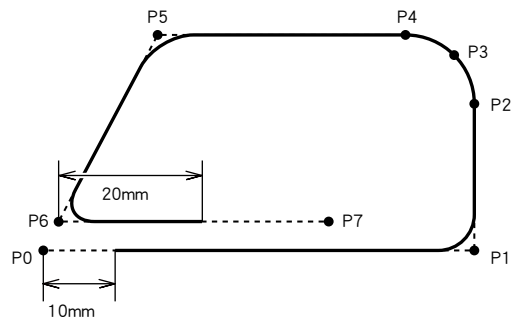
任务 2 (副任务) 也采用相同的流程表。

■ 程序示例

SAMPLE		
<主任务>		<副任务>
FLAG1=0		
FLAG2=0		
UPPOS=0.0		
START *S1, T2	副任务启动	
*L1 :		*S1 :
WAIT DI2 (2) =1	等待部件供给	WAIT DI3 (2) =1
WAIT FLAG2=0	等待其他任务的待机状态	WAIT FLAG1=0
FLAG1=1	排他控制标志设置	FLAG2=1
GOSUB *OPEN	夹持器开启	GOSUB *OPEN
MOVE P, P1, Z=UPPOS	向供给位置移动	MOVE P, P11, Z=UPPOS
GOSUB *CLOSE	夹持器关闭	GOSUB *CLOSE
MOVE P, P2, Z=UPPOS	向判断机移动	MOVE P, P12, Z=UPPOS
GOSUB *OPEN	夹持器开启	GOSUB *OPEN
DRIVEI (3, -10000)	Z 上升 10000 脉冲	DRIVEI (3, -10000)
FLAG1=0	排他控制标志复位	FLAG2=0
DO2 (0) =1	判断机启动	DO3 (0) =1
DELAY 100		DELAY 100
DO2 (0) =0		DO3 (0) =0
WAIT DI2 (0) =1	等待判断结束	WAIT DI3 (0) =1
WAIT FLAG2=0	等待任务动作结束	WAIT FLAG1=0
FLAG1=1	排他控制标志设置	FLAG2=1
MOVE P, P2, Z=UPPOS	向判断机移动	MOVE P, P12, Z=UPPOS
GOSUB *CLOSE	夹持器关闭	GOSUB *CLOSE
IF DI2 (1) =1 THEN	判断	IF DI3 (1) =1 THEN
'GOOD		'GOOD
WAIT DI4 (2) =0	等待部件移动	WAIT DI3 (3) =0
MOVE P, P3, Z=UPPOS	向合格品位置移动	MOVE P, P13, Z=UPPOS
ELSE		ELSE
'NG		'NG
WAIT DI2 (4) =0	等待部件移动	WAIT DI3 (4) =0
MOVE P, P4, Z=UPPOS	向不良品位置移动	MOVE P, P14, Z=UPPOS
ENDIF		ENDIF
GOSUB *OPEN	夹持器开启	GOSUB *OPEN
DRIVEI (3, -10000)	Z 上升 10000 脉冲	DRIVEI (3, -10000)
FLAG1=0	排他控制标志复位	FLAG2=0
GOTO *L1		GOTO *S1
*OPEN :		
DO2 (1) =0		
DELAY 100		
RETURN		
*CLOSE :		
DO2 (1) =1		
DELAY 100		
RETURN		

■ 概要
进行下图所示的部件涂胶。

涂胶



35C16-R7-00

■ 前提

1. 输入输出信号

DO (20)	阀开关	1 : 开 / 0 : 关
---------	-----	---------------

2. P0 ~ P7 通过示教设置。

SAMPLE

```
MOVE P, P0, Z=0
SPEED 40
PATH SET ..... 开始路径设置
PATH L, P1, DO (20) =1 @ 10.0 ..... 从 10mm 的位置开始涂胶
PATH L, P2
PATH C, P3, P4
PATH L, P5
PATH L, P6, S=30
PATH L, P7, DO (20) =0 @ 20.0 ..... 在 20mm 的位置结束涂胶
PATH END ..... 结束路径设置
PATH START ..... PATH 移动的执行（机器人从 P0 开始动作，在 P7 处停止）
HALT
```

动作路径的设置
(机器人不动作)

2.6 使用外部装置与 RS-232C 的连接（其 1）

■ 概要
使用外部装置与 RS-232C 连接 RCX 系列控制器，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



要点

· [cr/lf]表示 CR 代码 (=0Dh)+LF 代码 (=0Ah)。

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							



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 RCX 系列控制器，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$="SDATA/X/Y"
  VIN$=""
  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 VIN$
  I=1
  VMAX=LEN (VIN$)
*LOOP :
  IF I>VMAX THEN GOTO *E_LOOP
  C$=MID$ (VIN$, I, 1)
  IF C$="X" THEN
    I=I+2
    J=I
*X_LOOP :
  C$=MID$ (VIN$, J, 1)
  IF C$="," THEN
*X1_LP :
    L=J-I
    VX$=MID$ (VIN$, I, L)
    I=J+1
    GOTO *LOOP
  ENDIF
  J=J+1
  IF J>VMAX THEN GOTO *X1_LP
  GOTO *X_LOOP
ENDIF
IF C$="Y" THEN
  I=I+2
  J=I
*Y_LOOP :
  C$=MID$ (VIN$, J, 1)
  IF C$="," THEN
*Y1_LP :
    L=J-I
    VY$=MID$ (VIN$, I, L)
    I=J+1
    GOTO *LOOP
  ENDIF
  J=J+1
  IF J>VMAX THEN GOTO *Y1_LP
  GOTO *Y_LOOP
ENDIF
I=I+1
GOTO *LOOP
*E_LOOP :
  WX=VAL (VX$)
  WY=VAL (VY$)
  LOCX (P11) =WX
  LOCY (P11) =WY
  MOVE P, P11
GOTO *ST

```

第 12 章

联机命令

1	联机命令一览表	12-1
2	按键操作.....	12-6
3	UTILITY 操作.....	12-12
4	数据处理.....	12-24
5	机器人语言的单独执行.....	12-42
6	控制代码.....	12-44

使用联机命令后，可通过 RS-232C 或 Ethernet (选项) 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。有关 RS-232C 的连接方法，请参阅《RCX 控制器手册》，有关 Ethernet，请参阅《RCX 系列专用 Ethernet 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 (选项)
 - 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 按键操作 > 以后的各联机命令的详细说明中，终端编码标注为 [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	条件
模式更改	自动模式 程序模式 手动模式 系统模式	AUTO PROGRAM MANUAL SYSTEM		3
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		4 2
设置断点		BREAK	m, n (m : 断点编号, n : 行)	4
切换执行任务		CHGTSK		3
更改手动速度	主机器人 副机器人	MSPEED MSPEED2	k (k : 1-100) k (k : 1-100)	3
可进行绝对式原点复位的位置移动	主机器人 副机器人	ABSADJ ABSADJ2	k, 0 或 k, 1 (k : 1-6) k, 0 或 k, 1 (k : 1-6)	3
绝对式原点复位轴单位	主机器人 副机器人	ABSRESET ABSRESET2	k (k : 1-6) k (k : 1-6)	
返回原点	主机器人 副机器人	ORGRTN ORGRTN2	k (k : 1-6) k (k : 1-6)	3
手动移动 (点动)	主机器人 副机器人	INCH INCH2	k+ 或 k- (k : X, Y, Z, R, A, B) k+ 或 k- (k : X, Y, Z, R, A, B)	3
手动移动 (微动)	主机器人 副机器人	JOG JOG2	k+ 或 k- (k : X, Y, Z, R, A, B) k+ 或 k- (k : X, Y, Z, R, A, B)	3
坐标点数据示教	主机器人 副机器人	TEACH TEACH2	m (m : 坐标点编号) m (m : 坐标点编号)	3

UTILITY

含义	命令	选配件	条件
获取程序执行状态	PADDR		4
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	3
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	3
将程序名从 1 更改为 2	REN	<程序 1> TO <程序 2>	3
更改程序属性	ATTR	<程序> TO s (s: RW/RO)	3
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	3
数据初始化 通信参数	INIT	CMU	3
数据初始化 错误履历	INIT	LOG	3
设置 显示语言	LANGUAGE	k	3
设置 坐标点单位制 行信息的取消 访问级 执行级别 次序执行标志	UNIT MSGCLR ACCESS EXELVL SEQUENCE	k k k k k	3 1 3 3 3
设置 主手系统 副手系统	ARMTYP ARMTYP2	m, k m, k	3
解除内部紧急停止标志	EMGRST		1
日期的确认、设置	DATE		2
时间的确认、设置	TIME		2

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	?	LANGUAGE ACCESS ARM BREAK CONFIG EXELVL MOD MSG [m, n] ORIGIN ABSRST SERVO SEQUENCE SPEED UNIT VER WHERE WHERE2 WHRXY WHRXY2 TASKS TSKMON SHIFT HAND MEM EMG SELFCHK OPSLOT 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
程序的切换	SWI	<程序>	4
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，还有固有的限制

命令	选配件	含义	条件
?	ABSRST	获取绝对式原点复位状态	1
	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
	LANGUAGE	获取显示语言	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
	OPSLLOT	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
	SPEED	获取自动手动速度状态	1
	TASKS	获取任务编号	1
	TSKMON	获取任务运行状态	1
	UNIT	获取坐标点位置状态	1
	VER	获取版本	1
	WHERE	获取主当前位置（脉冲）	1
	WHERE2	获取副当前位置（脉冲）	1
	WHRXY	获取主当前位置（XY 坐标）	1
	WHRXY2	获取副当前位置（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 主机器人	3
ABSADJ2	k, 0 或 k, 1 (k : 1-6)	可进行绝对式原点复位的位置移动 副机器人	3
ABSRESET	k (k : 1-6)	绝对式原点复位轴单位 主机器人	3
ABSRESET2	k (k : 1-6)	绝对式原点复位轴单位 副机器人	3
ACCESS	k	设置访问级	3
ARMTYP	m, k	设置主手系统	3
ARMTYP2	m, k	设置副手系统	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
CHGTSK		切换执行任务	3
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

命令	选配件	含义	条件
INCH	k+ 或 k- (k : X, Y, Z, R, A, B)	手动移动 (点动) 主机器人	3
INCH2	k+ 或 k- (k : X, Y, Z, R, A, B)	手动移动 (点动) 副机器人	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
	PRM	参数初始化	3
	SFT	位移数据初始化	3
JOG	k+ 或 k- (k : X, Y, Z, R, A, B)	手动移动 (微动) 主机器人	3
JOG2	k+ 或 k- (k : X, Y, Z, R, A, B)	手动移动 (微动) 副机器人	3
LANGUAGE	k	显示语言设置	3
MANUAL		模式更改 : 手动模式	3
MSGCLR		设置 行信息的取消	1
MSPEED	k (k : 1-100)	更改手动速度 主机器人	3
MSPEED2	k (k : 1-100)	更改手动速度 副机器人	3
NEXT		将程序执行至下一行为止	4
ORGRTN	k (k : 1-6)	返回原点 主机器人	3
ORGRTN2	k (k : 1-6)	返回原点 副机器人	3
PADDR		获取程序执行状态	4
PROGRAM		模式更改 : 程序模式	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
SWI	<程序>	程序的切换	4
SYSTEM		模式更改 : 系统模式	3
TEACH	m (m : 坐标点编号)	坐标点数据示教 主机器人	3
TEACH2	m (m : 坐标点编号)	坐标点数据示教 副机器人	3
TIME		时间的确认、设置	2
UNIT	k	坐标点单位制设置	3
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 按键操作

2.1 模式更改

命令格式

```
@ AUTO [cr/lf]
@ PROGRAM [cr/lf]
@ MANUAL [cr/lf]
@ SYSTEM [cr/lf]
```

应答格式

```
OK [cr/lf]
```



要点

- 键盘操作的联机命令如果接收到命令，基本上会应答"OK"。
- 在发生了错误且无法执行联机命令时，应答错误提示。

含义

进行模式更改。

AUTO	将模式更改为自动模式
PROGRAM	将模式更改为程序模式
MANUAL	将模式更改为手动模式
SYSTEM	将模式更改为系统模式

SAMPLE

```
命令： @ AUTO [cr/lf]
应答： OK [cr/lf]
```

1. 程序执行

命令格式

@ RESET [cr/lf]
@ RUN [cr/lf]
@ STEP [cr/lf]
@ SKIP [cr/lf]
@ NEXT [cr/lf]
@ STOP [cr/lf]

应答格式

OK [cr/lf]



要点

· 仅能在自动模式中执行。

含义

进行程序执行操作。

RESET..... 复位程序

RUN..... 执行程序

STEP..... 执行 1 行程序 (进入子例程内部)

SKIP..... 跳过 1 行程序 (子例程跳过 1 行)

NEXT..... 将程序执行至下一行 (子例程作为 1 行执行)

STOP..... 停止程序

SAMPLE

命令： @ RUN [cr/lf]

应答： OK [cr/lf]



要点

· 仅能在自动模式中执行。

2. 设置断点

命令格式

@ BREAK m, n [cr/lf]

应答格式

OK [cr/lf]

值

m.....断点编号：1 ~ 4

n.....设置断点的行号：0 ~ 9999

含义

在程序执行过程中，设置用于暂停的断点。

对 COMMON 程序的设置为行号 +10000。

指定了 0 行号时，断点将被取消。

SAMPLE

命令： @ BREAK 1, 28 [cr/lf]

应答： OK [cr/lf]

3. 切换执行任务

命令格式

@ CHGTASK [cr/lf]

应答格式

OK [cr/lf]

含义

在程序执行停止过程中，进行选择任务的切换操作。

进行任务切换时，按照 1 → 2 → ... → 8 → 1 的顺序切换执行停止中的任务编号。仅能切换启动中的任务。

SAMPLE

命令： @ CHGTASK [cr/lf]

应答： OK [cr/lf]



要点

· 仅能在自动模式中执行。

1. 更改手动速度

命令格式

@ MSPEED k [cr/lf]
@ MSPEED2 k [cr/lf]

应答格式

OK [cr/lf]

值 k 手动移动速度：1 ~ 100

含义 进行手动移动模式的速度更改。
MSPEED : 进行主机器人的手动移动速度更改。
MSPEED2 : 进行副机器人的手动移动速度更改。

SAMPLE

命令 : @ MSPEED 50 [cr/lf]
应答 : OK [cr/lf]

2. 绝对式原点复位

命令格式

@ ABSADJ k, f [cr/lf]
@ ABSADJ2 k, f [cr/lf]
@ ABSRESET k [cr/lf]
@ ABSRESET2 k [cr/lf]

应答格式

OK [cr/lf]

值 k 指定轴：1 ~ 6
f 移动方向 / 0 : + 方向、1 : - 方向

含义 进行绝对式原点复位操作。
ABSADJ 主机器人轴移动至可进行绝对式原点复位的位置
ABSADJ2 副机器人轴移动至可进行绝对式原点复位的位置
ABSRESET 主机器人轴的绝对式原点复位
ABSRESET2 副机器人轴的绝对式原点复位

SAMPLE

命令 : @ ABSADJ 1, 0 [cr/lf]
应答 : OK [cr/lf]

MEMO · ABSADJ、ABSADJ2 可对标记方式的轴使用。



要点

· 仅能在手动模式中执行。



要点

· 仅能在手动模式中执行。



要点

- 轴指定在软件版本 Ver.8.45 以上有效。



要点

- 仅能在手动模式中执行。



要点

- 在移动结束后，发送应答。



要点

- 仅能在手动模式中执行。

3. 返回原点动作

命令格式

```
@ ORGRTN k [cr/lf]
@ ORGRTN2 k [cr/lf]
```

应答格式

```
OK [cr/lf]
```

值 k.....轴指定：1 ～ 6

含义 进行指定轴的返回原点。
对于增量式规格的轴，执行返回原点将进行返回原点的动作。
对于半绝对式规格的轴，执行返回原点后就将进行绝对位置搜索。
ORGRTN.....主机器人轴的返回原点
ORGRTN2.....副机器人轴的返回原点

SAMPLE

```
命令： @ ORGRTN 1 [cr/lf]
应答： OK [cr/lf]
```

4. 手动移动：点动

命令格式

```
@ INCH km [cr/lf]
@ INCH2 km [cr/lf]
```

应答格式

```
OK [cr/lf]
```

值 k.....轴指定：X, Y, Z, R, A, B
m.....移动方向 / +、-

含义 进行指定轴的手动移动（点动）。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。
INCH.....主机器人轴的点动
INCH2.....副机器人轴的点动

SAMPLE

```
命令： @ INCH X+ [cr/lf]
应答： OK [cr/lf]
```

5. 手动移动：微动

命令格式

@ JOG km [cr/lf]
@ JOG2 km [cr/lf]

应答格式

OK [cr/lf]

值 k.....轴指定：X, Y, Z, R, A, B
m.....移动方向 / +、-

含义 进行指定轴的手动移动（微动）。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。

移动开始后，如果变为以下任意一种状态，机器人将停止，并发送应答。

- 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（^C (=03H)）

JOG.....主机器人轴的微动

JOG2.....副机器人轴的微动

SAMPLE

命令： @ JOG X+[cr/lf]
应答： OK [cr/lf]



要点

- 在移动结束后，发送应答。

6. 坐标点示教

命令格式

@ TEACH mmmm [cr/lf]
@ TEACH2 mmmm [cr/lf]

应答格式

OK [cr/lf]

值 mmmm.....录入坐标点数据的坐标点编号：0 ~ 9999

含义 将机器人的当前位置录入至指定坐标点编号的坐标点数据中。
当指定的坐标点编号已经存在于坐标点数据中时，坐标点数据将被覆盖。坐标点按照当前所选的单位制格式录入。

TEACH.....将主组的当前位置录入到指定的坐标点编号

TEACH2.....将副组的当前位置录入到指定的坐标点编号

SAMPLE

命令： @ TEACH 100 [cr/lf]
应答： OK [cr/lf]



要点

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。
- 仅能在手动模式中执行。

3.1

@ PADDR [cr/lf]

<程序名> T_n, m, k [cr/lf]

值	<程序名>	当前所选的程序名
	Tn.....	当前的任务编号：1 ~ 8
	m.....	当前的程序行号：1 ~ 9999
	k.....	当前的任务优先级：17 ~ 47

含义 获取程序执行状态。

- 使用 COMMON 程序时，可能会变为以下所示的应答格式。
`< COMMON > / < 程序名 > , Tn,m,k[cr/lf]`

命令： @ PADDR [cr/lf]
 应答： < TEST > , T3, 134, 32 [cr/lf]

3.2

1. 程序的复制

COPY <程序名 1> TO <程序名 2> [cr/lf]

@ COPY <程序名 1> TO <程序名 2> [cr/lf]

值

<程序名 1>	复制源的程序名 (采用英文字母数字及“_”，在 8 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 8 个字符以内)

含义 进行从程序名 1 复制至程序名 2。

命令: @ COPY < TEST1 > TO < TEST2 > [cr/lf]
 应答: OK [cr/lf]



- 仅能在自动模式中且程序停止状态下才可执行。



要点

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。

2. 坐标点的复制

命令格式

@ COPY Pmmmm-Pnnnn TO Pkkkk [cr/lf]

应答格式

OK [cr/lf]

值	mmmm	复制源的起始坐标点编号：0 ~ 9999
	nnnn	复制源的结尾坐标点编号：0 ~ 9999
	kkkk	复制目的地的起始坐标点编号：0 ~ 9999

含义 将 Pmmmm 至 Pnnnn 的坐标点复制至 Pkkkk。

SAMPLE

命令： @ COPY P101-P200 TO P1101 [cr/lf]

应答： OK [cr/lf]

3. 坐标点注释的复制

命令格式

@ COPY PCmmmm-PCnnnn TO PCkkkk [cr/lf]

应答格式

OK [cr/lf]

值	mmmm	复制源的起始坐标点注释编号：0 ~ 9999
	nnnn	复制源的结尾坐标点注释编号：0 ~ 9999
	kkkk	复制目的地的起始坐标点注释编号：0 ~ 9999

含义 将 PCmmmm 至 PCnnnn 的坐标点注释复制至 PCkkkk。

SAMPLE

命令： @ COPY PC101-PC200 TO PC1101 [cr/lf]

应答： OK [cr/lf]



要点

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。

1. 程序删除

命令格式

@ ERA <程序名> [cr/lf]

应答格式

OK [cr/lf]

值 <程序名>删除的程序名(采用英文字母数字及 "_", 8 个字符以内)

含义 进行程序的删除操作。

SAMPLE

命令： @ ERA < TEST1 > [cr/lf]

应答： OK [cr/lf]

2. 删除坐标点

命令格式

@ ERA Pmmmm-Pnnnn [cr/lf]

应答格式

OK [cr/lf]

值 mmmmm删除的起始坐标点编号：0 ~ 9999

nnnn删除的结尾坐标点编号：0 ~ 9999

含义 进行 Pmmmm 至 Pnnnn 的坐标点删除操作。

SAMPLE

命令： @ ERA P101-P200 [cr/lf]

应答： OK [cr/lf]



要点

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。



要点

- 对于软件版本为 Ver.8.28 以前的控制器，坐标点变量中可指定的坐标点编号为 0 ~ 4000。

3 删除坐标点注释

命令格式

@ ERA PCmmmm-PCnnnn [cr/lf]

应答格式

OK [cr/lf]

值 mmmmm.....删除的起始坐标点注释编号：0 ~ 9999
nnnn.....删除的结尾坐标点注释编号：0 ~ 9999

含义 进行 PCmmmm 至 PCnnnn 的坐标点注释删除操作。

SAMPLE

命令： @ ERA PC101-PC200 [cr/lf]

应答： OK [cr/lf]

4. 删除托盘

命令格式

@ ERA PLm [cr/lf]

应答格式

OK [cr/lf]

值 m.....删除的托盘编号：0 ~ 19

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]

命令格式

@ REN <程序名 1> TO <程序名 2> [cr/lf]

应答格式

OK [cr/lf]

值 <程序名 1>更改前的程序名(采用英语字母数字及"_" ,8 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"_" ,8 个字符以内)

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： OK [cr/lf]

命令格式

@ ATTR <程序名> 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"_" , 8 个字符以内)
s属性 / RW : 可读写、RO : 仅可读出

含义 进行程序名的属性更改操作。

SAMPLE

命令： @ ATTR < TEST1 > TO RO [cr/lf]
应答： OK [cr/lf]

1. 存储器区域的初始处理

命令格式
@ INIT < 存储器区域 > [cr/lf]

应答格式
OK[cr/lf]

- 值 < 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
 - 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： OK [cr/lf]

2. 通信端口的初始处理

命令格式
@ INIT CMU [cr/lf]

应答格式
OK [cr/lf]

含义 进行通信端口的初始处理操作。
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： OK [cr/lf]

3. 错误履历的初始处理操作

命令格式

```
@ INIT LOG [cr/lf]
```

应答格式

```
OK [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]

3.7

显示语言设置

命令格式

```
@ LANGUAGE k [cr/lf]
```

应答格式

```
OK [cr/lf]
```

值 k.....显示语言 / 0：日语、1：英语

含义 进行控制器显示语言的设置操作。

SAMPLE

命令： @ LANGUAGE 1 [cr/lf]

应答： OK [cr/lf]

3.8 手动模式时的坐标系单位设置

命令格式

@ UNIT k [cr/lf]

应答格式

OK [cr/lf]

值 k 单位指定
0 : 脉冲单位
1 : mm/deg 单位
2 : mm/deg 单位且工具坐标模式

含义 选择当前位置的显示单位。

仅当在正交型机器人或水平多关节型机器人上设置了机械手定义且选择了安装 R 轴的机械手定义时，才可设置 k=2 (工具坐标模式)。

SAMPLE

命令 : @ UNIT 1 [cr/lf]

应答 : OK [cr/lf]



要点

- K=2 (mm/deg 单位且工具坐标模式) 的指定在软件版本 Ver.8.19 以上有效。

3.9 手持编程器显示错误提示的初始化

命令格式

@ MSGCLR [cr/lf]

应答格式

OK [cr/lf]

含义 进行手持编程器上的错误提示显示的初始化处理。

SAMPLE

命令： @ MSGCLR [cr/lf]

应答： OK [cr/lf]

3.10 UTILITY 模式设置

1. 设置访问级

命令格式

@ ACCESS k [cr/lf]

应答格式

OK [cr/lf]

值 k.....访问级：0～3

含义 进行访问级的设置处理。

SAMPLE

命令： @ ACCESS 1 [cr/lf]

应答： OK [cr/lf]

2. 设置执行级别

命令格式

@ EXELVL k [cr/lf]

应答格式

OK [cr/lf]

值 k.....执行级别：0～8

含义 进行执行级别的设置处理。

SAMPLE

命令： @ EXELVL 1 [cr/lf]

应答： OK [cr/lf]



· 关于访问级的详细说明，请参阅控制器的手册。



· 关于执行级别的详细说明，请参阅控制器的手册。

3. 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]

4. 水平多关节型机器人的手系统设置

命令格式

@ ARMTYP m, k [cr/lf]

@ ARMTYP2 m, k [cr/lf]

应答格式

OK [cr/lf]

值 m 当前的手系统 / 0 : 右手系统、1 : 左手系统
k 程序复位时的手系统 / 0 : 右手系统、1 : 左手系统

含义 进行水平多关节型机器人的手系统设置处理。
ARMTYP 主机器人的手系统设置更改
ARMTYP2 副机器人的手系统设置更改

SAMPLE

命令 : @ ARMTYP 0, 0 [cr/lf]

应答 : OK [cr/lf]

5. 解除内部紧急停止标志

命令格式

@ EMGRST [cr/lf]

应答格式

OK [cr/lf]

含义 进行内部紧急停止标志的解除处理。

SAMPLE

命令 : @ EMGRST [cr/lf]

应答 : OK [cr/lf]



要点

- 日期的确认及设置在以下软件版本条件下有效。
- | | |
|--------|-------------|
| RCX14x | Ver.8.64 以上 |
| RCX22x | Ver.9.11 以上 |

命令格式

@ DATE [cr/lf]

应答格式

current date : yy/mm/dd [cr/lf]
enter new date : (YY/MM/DD) [cr/lf]

值

yy/mm/dd..... 当前日期(年月日)
yy 公历的后 2 位数 (00 ~ 99)
mm 月 (01 ~ 12)
dd..... 日 (01 ~ 31)

含义

获取控制器的当前日期，并设置新的日期。

命令格式

aa/bb/cc [cr/lf]

应答格式

OK [cr/lf] 或 “错误提示” [cr/lf]

值

aa/bb/cc 重新设置的日期(年月日)
aa 公历的后 2 位数 (00 ~ 99 : 可省略)
bb..... 月 (01 ~ 12 : 可省略)
cc 日 (01 ~ 31 : 可省略)



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

- 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 [cr/lf]
应答 : current date : 07/05/10 [cr/lf]
 enter new date : (YY/MM/DD) [cr/lf]
发送 : 07/05/11 [cr/lf]
应答 : OK [cr/lf]



要点

- 时间的确认与设置在以下软件版中有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

命令格式

@ TIME [cr/lf]

应答格式

current time : hh : mm : ss [cr/lf]
enter new time : (HH : MM : SS) [cr/lf]

值 hh : mm : ss 当前时间
 hh 时 (00 ~ 23)
 mm 分 (00 ~ 59)
 ss 秒 (00 ~ 59)

含义 获取控制器的当前时间，并设置新的时间。

命令格式

aa:bb:cc [cr/lf]

应答格式

OK [cr/lf] 或 “错误提示” [cr/lf]

值 aa : bb : cc 重新设置时间
 aa 时 (00 ~ 23 : 可省略)
 bb 分 (00 ~ 59 : 可省略)
 cc 秒 (00 ~ 59 : 可省略)

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 [cr/lf]
应答 : current time : 10:21:35 [cr/lf]
 enter new time : (HH/MM/SS) [cr/lf]
发送 : 10:25:00 [cr/lf]
应答 : OK [cr/lf]

4.1 获取显示语言

命令格式

@ ?LANGUAGE [cr/lf]

应答格式

m [cr/lf]

值 m.....显示语言 / JAPANESE : 日语显示、ENGLISH : 英语显示

含义 进行显示语言的获取处理。

SAMPLE

命令 @ ?LANGUAGE [cr/lf]

应答 JAPANESE [cr/lf]

4.2 获取访问级状态

命令格式

@ ?ACCESS [cr/lf]

应答格式

LEVELk [cr/lf]

值 k.....访问级 : 0 ~ 3

含义 进行访问级状态的获取处理。

SAMPLE

命令 @ ?ACCESS [cr/lf]

应答 LEVEL2 [cr/lf]



参考

· 关于访问级的详细说明，请参阅控制器的手册。

4.3 获取机械臂状态

命令格式	
@ ?ARM [cr/lf]	
应答格式	
m1/s1, m2/s2 [cr/lf]	
值	主机器人
m1	当前的机械臂设置状态 / RIGHTY : 右手系统、LEFTY : 左手系统
m2	程序复位时的机械臂设置状态 / RIGHTY : 右手系统、LEFTY : 左手系统
	副机器人
s1	当前的机械臂设置状态 / RIGHTY : 右手系统、LEFTY : 左手系统
s2	程序复位时的机械臂设置状态 / RIGHTY : 右手系统、LEFTY : 左手系统
含义	进行机械臂状态的获取处理。

MEMO

- 仅在设置水平多关节型机器人时有效。
- 未设置副机器人时，不显示 s1 及 s2。

SAMPLE	
命令	@ ?ARM [cr/lf]
应答	RIGHTY, RIGHTY [cr/lf]

4.4 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

@ ?CONFIG [cr/lf]

应答格式

mr/sr-ma/sa-r-o1-o2 [cr/lf]

值	mr.....	主机器人的设置名
	sr.....	副机器人的设置名
	ma.....	主机器人的轴设置(附加轴用 + 分隔)
	sa.....	副机器人的轴设置(附加轴用 + 分隔)
	r.....	标准接口设备
	o1.....	选配设备
	o2.....	其他设置

含义 获取控制器的构成。



· 未设置副机器人时，不显示 s1 及 s2。

SAMPLE

命令 @ ?CONFIG [cr/lf]
应答 YK250X-XYZR-SRAM/196kB, DIO_N-DIO_N (1/2) [cr/lf]

命令格式

@ ?EXELVL [cr/lf]

应答格式

LEVELk [cr/lf]

值	k.....	执行级别：0 ~ 8
---	--------	------------

含义 获取执行级别的状态。

SAMPLE

命令 @ ?EXELVL [cr/lf]
应答 LEVEL2 [cr/lf]



· 关于执行级别的详细说明，请参阅控制器的手册。

命令格式

@ ?MOD [cr/lf]

应答格式

s [cr/lf]

值 s..... 模式状态

s		含义
英语显示	日语显示	
AUTO	ジドウ	自动模式
PROGRAM	プログラム	程序模式
MANUAL	シュドウ	手动模式
SYSTEM	システム	系统模式

含义 获取控制器的模式状态。

SAMPLE

命令 @ ?MOD [cr/lf]

应答 AUTO [cr/lf]

命令格式 1

@ ?MSG [c/r]

应答格式 1

gg, bb: msg [c/r] 或 OK [c/r]

命令格式 2

@ ?MSG m, n [cr/lf]

应答格式 2

yy/mm/dd, hh : mm : ss gg.bb : msg [cr/lf]
yy/mm/dd, hh : mm : ss gg.bb : msg [cr/lf]
:
OK [cr/lf]

值	gg.....	错误组
	bb.....	错误分类
	msg.....	错误提示
	m.....	获取起始编号 : 1 ~ 500
	n.....	获取结尾编号 : 1 ~ 500
	yy/mm/dd.....	错误发生年月日
	hh:mm:ss	错误发生时分秒

含义 格式 1 : 获取手持编程器上的提示行信息。
格式 2 : 获取错误履历的提示。

SAMPLE 1

命令 @ ?MSG [cr/lf]

应答 5.30 : Undefined identifier [cr/lf] 或 OK [cr/lf]

SAMPLE 2

命令 @ ?MSG 1, 5 [cr/lf]

应答 01/10/28, 14 : 20 : 20 5.30 : Undefined identifier [cr/lf]
01/10/28,14 : 18 : 34 5.1 : Syntax error [cr/lf]
01/10/28,14 : 10 : 54 5.30 : Undefined identifier [cr/lf]
01/10/28,14 : 05 : 40 14.22 : No start code (@) [cr/lf]
01/10/28,14 : 05 : 00 5.52 : Command doesn't exist [cr/lf]
OK [cr/lf]

4.9 获取返回原点状态

命令格式	
@ ?ORIGIN [cr/lf]	
应答格式	
COMPLETE [cr/lf] 或 INCOMPLETE [cr/lf]	
含义	获取返回原点状态。 应答格式 COMPLETE : 结束状态 INCOMPLETE : 未结束状态
SAMPLE	
命令	@ ?ORIGIN [cr/lf]
应答	COMPLETE [cr/lf]

4.10 获取绝对式原点复位状态

命令格式	
@ ?ABSRST [cr/lf]	
应答格式	
COMPLETE [cr/lf] 或 INCOMPLETE, xxxxxxxx [cr/lf]	
值	xxxxxxx 从左开始依次显示 8 轴、7 轴、...、1 轴的状态。 0 : 未结束状态 1 : 结束状态 9 : 非执行对象
含义	获取绝对式原点复位状态。 应答格式 COMPLETE : 结束状态 INCOMPLETE : 未结束状态
SAMPLE	
命令	@ ?ABSRST [cr/lf]
应答	INCOMPLETE, 99991011 [cr/lf]

4.11 获取伺服状态

命令格式

@ ?SERVO [cr/lf]

应答格式

OFF, xxxxxxxx [cr/lf] 或 ON, xxxxxxxx [cr/lf]

值 xxxxxxxx从左开始依次显示 8 轴、7 轴、…、1 轴的状态。
0：机械制动开启 + 动态制动开启状态
1：伺服使能状态
2：机械制动关闭 + 动态制动关闭状态
9：非执行对象

含义 获取伺服状态。
应答输出表示含义如下。
ON 马达电源开启状态
OFF 马达电源关闭状态

SAMPLE

命令 @ ?SERVO [cr/lf]
应答 ON, 99991011 [cr/lf]

4.12 获取次序程序执行状态

命令格式

@ ?SEQUENCE [cr/lf]

应答格式

1. ENABLE, s [cr/lf]
2. ENABLE (RST.DO), s [cr/lf]
3. DISABLE [cr/lf]

值 s次序程序的执行状态通过 "RUNNING" 或 "STOP" 表示。
RUNNING 执行中
STOP 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。
ENABLE 许可状态
ENABLE (RST.DO) 在许可状态且紧急停止时清除输出
DISABLE 禁止状态

SAMPLE

命令 @ ? SEQUENCE [cr/lf]
应答 DISABLE [cr/lf]

4.13 获取速度状态

命令格式	
@ ?SPEED [cr/lf]	
应答格式	
ma/sa, mm/sm [cr/lf]	
值	主组
	ma 自动移动速度设置状态：1 ~ 100
	mm 手动移动速度设置状态：1 ~ 100
	副组
	sa 自动移动速度设置状态：1 ~ 100
	sm 手动移动速度设置状态：1 ~ 100
含义	获取移动速度的设置值。

MEMO

· 未设置副机器人时，不显示 sa 及 sm。

SAMPLE	
命令	@ ?SPEED [cr/lf]
应答	100, 50 [cr/lf]

4.14 获取坐标点单位坐标系

命令格式	
@ ?UNIT [cr/lf]	
应答格式	
s [cr/lf]	
值	s：单位坐标系 PULSE：脉冲单位的关节坐标 MM：毫米 / DEG 单位的正交坐标
含义	获取坐标点单位坐标系。
SAMPLE	
命令	@ ?UNIT [cr/lf]
应答	PULSE [cr/lf]

命令格式

@ ?VER [cr/lf]

应答格式

cv, cr-mv-d1/d2/d3/d4/d5/d6/d7/d8{-ov} [cr/lf]

值 cv主机的版本 (RCX14x : V8.xx RCX22x : V9.xx)
cr主机的修订版本 (Rxxxx)
mv手持编程器的版本 (Vx.xx)
d? (? : 1-8)驱动器的版本 (Vx.xx)
ov选配设备的版本 (Vx.xx)

含义 获取版本。

SAMPLE

命令 @ ?VER [cr/lf]

应答 V8.02, R1021-V5.10-V1.01/V1.01/V1.01/V1.01/-----/-----/-----/-----/ [cr/lf]

1. 获取脉冲坐标系的当前位置

命令格式

@ ?WHERE [cr/lf]
@ ?WHERE2 [cr/lf]

应答格式

[POS] 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 : 获取主组轴的当前位置。
WHERE2 : 获取副组轴的当前位置。



· 未设置副机器人时，无法使用 WHERE2。

SAMPLE

命令 @ ?WHERE [cr/lf]

应答 [POS] 1000 2000 3000 -40000 0 0[cr/lf]

2. 获取 XY 坐标系的当前位置

命令格式

```
@ ?WHRXY [cr/lf]
@ ?WHRXY2 [cr/lf]
```

应答格式

```
[POS] xxxxxx yyyyyy zzzzzz rrrrrr aaaaaa bbbbbb [cr/lf]
```

值 xxxxxx..... 第 1 轴在毫米 / DEG 单位制中的当前位置
 yyyyyy 第 2 轴在毫米 / DEG 单位制中的当前位置
 :
 bbbbbb..... 第 6 轴在毫米 / DEG 单位制中的当前位置

含义 获取当前位置。
 WHRXY : 获取主组轴的当前位置。
 WHRXY2 : 获取副组轴的当前位置。



· 未设置副机器人时, 无法使用 WHRXY2。

SAMPLE

```
命令 @ ?WHRXY [cr/lf]
应答 [POS] 100.00 200.00 300.00 -40.00 0.00 0.00 [cr/lf]
```

4.17

获取启动 / 暂停状态任务的处理

命令格式

```
@ ?TASKS [cr/lf]
```

应答格式

```
n{, n{, {...}} } [cr/lf]
```

值 n : 任务编号 1 ~ 8 (当前启动状态或暂时休止状态的任务)

含义 获取启动 / 暂时休止状态的任务。

SAMPLE

```
命令 @ ?TASKS [cr/lf]
应答 1, 3, 4, 6 [cr/lf]
```

获取任务运行状态

4.20 获取机械手状态

命令格式

@ ?HAND [cr/lf]

应答格式

m/s [cr/lf]

值 m.....主机器人的选择机械手编号状态：0～3
 s.....副机器人的选择机械手编号状态：4～7

含义 获取机械手状态。



· 未设置副机器人时，不显示 s。

SAMPLE

命令 @ ?HAND [cr/lf]

应答 1 [cr/lf]

4.21 获取存储器剩余容量

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.....对象区域剩余容量(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]

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]

命令格式

@ ?SELFCHK [cr/lf]

· 正常状态

应答格式

OK [cr/lf]

· 异常状态

应答格式

m.n : "message" [cr/lf]

:

END [cr/lf]

值 m.....错误组
n.....错误分类
"message".....错误提示。

含义 通过自我诊断获取出错状态。确认控制器内部的错误状态。

SAMPLE

命令 @ ?SELFCHK [cr/lf]

应答 12.1 : Emg.stop on [cr/lf]

END [cr/lf]

命令格式	
@ ?OPSLOT [cr/lf]	
应答格式	
OP. 1 : < 选件板名 > [cr/lf] OP. 2 : < 选件板名 > [cr/lf] OP. 3 : < 选件板名 > [cr/lf] OP. 4 : < 选件板名 > [cr/lf]	
值	< 选件板名 > 安装的选件板名 DIO_Nm NPN 规格的 DIO 板 (m : 板 ID) DIO_Pm PNP 规格的 DIO 板 (m : 板 ID) CCLnk CC_LINK 扩展板 D_Net DeviceNet 扩展板 Profi Profibus 扩展板 E_Net Ethernet 扩展板 no board 没有板 illegal board 未扩展的板
含义	获取选配插槽状态。确认选件板的状态。
SAMPLE	
命令 :	@ ?OPSLOT [cr/lf]
应答 :	OP. 1 : DIO_N2 [cr/lf] OP. 2 : DIO_N1 [cr/lf] OP. 3 : no board [cr/lf] OP. 4 : CCLnk [cr/lf]



要点

· 值获取处理在以下软件版本中有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

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]

SAMPLE 2

命令： @ ?LOCX (WHERE) [cr/lf]
应答： 102054 [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]



要点

· 值获取处理在以下软件版本中有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上



要点

- 值获取处理在以下软件版本中有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

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]

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]



要点

- 值获取处理在以下软件版本中有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

命令格式

@ READ <读出文件> [cr/lf]

应答格式

应答输出取决于读出文件。



要点

- 有关各文件的详细说明，请参阅第 10 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



要点

- 坐标点注释的个别格式 (PCn) 在以下软件版本条件下有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

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	< 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	_____
	存储器剩余容量	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 ()
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 (_)

x：表达式 (数组参数)

c：英文字母数字或特殊符号

y：变量类型

SAMPLE

命令： @ READ PGM [cr/lf]
@ READ P100 [cr/lf]

命令格式

@ WRITE <写入文件> [cr/lf]

应答格式

输入要求显示 ▶ * * * Input the data! [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10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10章<数据文件详细说明>。



要点

- 坐标点注释的个别格式 (PC_n) 在以下软件版本条件下有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

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	< bbbbbbbb >
	坐标点	PNT	P _n
	坐标点注释	PCM	PC _n
	参数	PRM	/cccccc/
	位移定义	SFT	S _n
	机械手定义	HND	H _n
变量・常量	托盘定义	PLT	PL _n
	变量	VAR	ab...by
设备	数组变量	ARY	ab...by(x)
	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 (_)

x : 表达式 (数组参数)

c : 英文字母数字或特殊符号

y : 变量类型

SAMPLE 1

命令 : @ WRITE PRM [cr/lf]

@ WRITE P100 [cr/lf]

5.1 程序切换命令

命令格式

@ SWI <程序名> [cr/lf]

应答格式

OK [cr/lf] 或 LINEx, m, n : "message"

值	OK	命令正常结束
	LINEx, m, n : "message"	编译时发生错误
	x	错误发生行行号
	m	错误组
	n	错误分类
	"message"	错误提示

- 含义 进行程序切换处理。
- 自动模式中：执行程序的切换及编译。
 - 其他模式中：仅进行程序的切换。
- 但是，当指定程序为 "SEQUENCE" 时，将创建次序对象。

SAMPLE 1

自动模式中：
命令 @ SWI < TEST1 > [cr/lf]
应答 Line2,5.39 : Illegal identifier [cr/lf]

SAMPLE 2

其他模式中：
命令 @ SWI < TEST1 > [cr/lf]
应答 OK [cr/lf]

命令格式

@ “机器人语言” [cr/lf]

应答格式

OK [cr/lf] 或 * * * Aborted

值 OK.....命令正常结束
 * * * Aborted.....发生错误

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 OK [cr/lf]

6.1 执行中断处理

命令格式

^C (=03H)

应答格式

* * * Aborted

含义 中断执行中的命令。

SAMPLE

命令： @ MOVE P, P100, S=20 [cr/lf]
^C

应答： * * * Aborted [cr/lf]

第 13 章

IO 命令

1	概要	13-1
2	IO 命令格式.....	13-1
3	IO 命令的收发.....	13-3
4	IO 命令一览表.....	13-5
5	IO 命令详细说明	13-6

通过使用通用输入输出的比特信息 (DI / DO 端口), 可从 PLC 直接发行命令。同 MOVE 命令一样, 可执行以前必须使用机器人程序或 RS-232C 才能执行的命令。

MEMO

- 要使用 IO 命令, 必须先将系统模式中的 "Other parameters" 中的 "STD.DIO IO command (DI05)" 设置为有效。
 - RCX14x "Other parameters" 中的 "STD.DIO IO command (DI05)"
 - RCX22x "OP board parameters" 中的 "IO command (DI05)"
 详细说明, 请参阅控制器的手册。
- 如果要执行 IO 命令, 将强制使用通用输出 DO26 与 DO27 (RCX22x 使用通用输出 DO16 与 DO17)。用于其他用途时, 请特别注意。



要点

- IO 命令在软件版本 Ver.8.18 以上有效。

要使用 IO 命令时, 请将以下功能分配到各 IO。

输出: 控制器 → PLC

	输出端口	内容
RCX 14x/40	DO26	执行判定输出
	DO27	执行中输出
RCX 22x	DO16	执行判定输出
	DO17	执行中输出

输入: PLC → 控制器

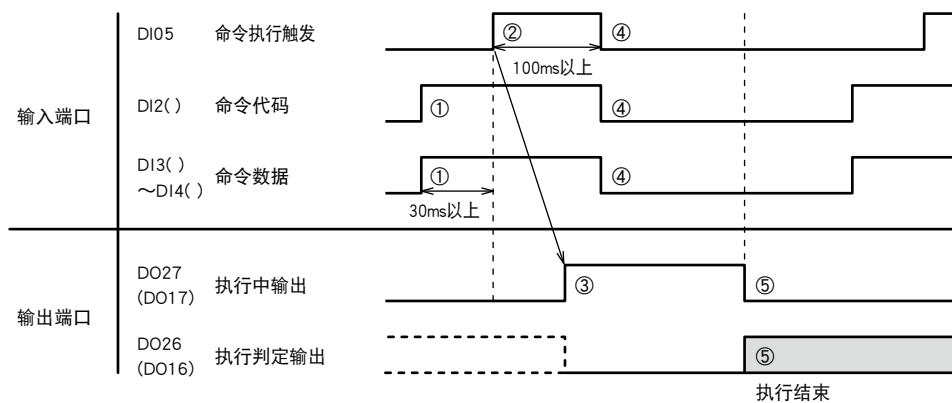
输入端口	内容
DI05	IO 命令执行触发输入
DI2 ()	命令代码
DI3 ()	命令数据
DI4 ()	

- 使用部分通用输入与通用输出可执行 IO 命令。此外, 未连接选配 DIO 时, DI4 () 将被识别为全部关闭状态。
- IO 命令在程序执行过程中 (DO13 开启状态) 无法执行。
- IO 命令无法与联机命令同时执行。
- **IO 命令将命令代码赋值于 DI2 () 中要执行的命令代码及 DI3 () 与 DI4 (), 使 DI05 从关闭变化为开启状态后执行。** 控制器接收到 IO 命令后, 将执行该处理并通过 DO26 与 DO27 (RCX22x 通过 DO16 与 DO17) 向 PLC 发送执行判定结果及执行中信息。
- IO 命令中附加的命令数据因 IO 命令而异。详细说明, 请参阅第 13 章 < 5 IO 命令详细说明 >。
- 数据设置通过二进制代码进行。当数据尺寸超过 8 比特时, 将高比特数据设置至较高的地址 (小端序)。
 - 例如, 要在 DI3 () 与 DI4 () 的端口上设置 0x0F9F (十六进制) (=3999) 时, 将 0x0F (十六进制) 设置至 DI4 (), 将 0x9F (十六进制) 设置至 DI3 ()。
- 执行中输出为开启状态时, 不接受 IO 命令执行触发输入。

- 执行中输出在以下情况下变为开启状态。
 - 接受 IO 命令执行触发输入，并执行 IO 命令。
 - 虽然接受 IO 命令执行触发输入并执行完成了 IO 命令，但由于 IO 命令执行触发输入处于开启状态，所以将保持最大 100ms 状态。
- **IO 命令执行触发输入的脉冲输入必须保持在 100ms 以上。**如果不保持，有时可能不接受命令。
- 从 IO 命令执行触发输入的上升沿脉冲至执行中输出变为开启状态，有时需要 20ms 以上的时间。在此时间段内，有时可能不接受 IO 命令执行触发输入。
- IO 命令执行触发输入的脉冲输入必须在执行中输出关闭后等待 100ms 之后再执行下一个命令。如果等待时间较短，有时可能不接受 IO 命令执行触发输入。
- 当 IO 命令被接受时，执行判定输出将关闭。
- 当 IO 命令正常结束时，执行判定输出会变为开启状态。但是，当 IO 命令异常结束时，将保持关闭的状态。

如下所示，进行 IO 寄存器的收发。

IO 命令收发



*RCX22x的输出端口变为 "DO27→DO17、DO26→DO16"。

35E01-R7-00

- ① 命令代码与命令数据的设置 (① - ②之间 : 30ms 以上)
- ② IO 命令执行触发输入的设置 (脉冲宽度 : 100ms 以上)
- ③ 在执行中的变化
- ④ IO 命令执行触发输入与命令代码及命令数据的清除
- ⑤ IO 命令的执行结束与执行判定输出的设置

MEMO

- 执行判定输出的输出端口
 - RCX14x : DO26
 - RCX22x : DO16
- 执行中输出的输出端口
 - RCX14x : DO27
 - RCX22x : DO17

例如，向坐标点 19 执行 PTP 移动命令时的收发示例

- ① 若要执行坐标点指定的 PTP 移动命令，需对以下寄存器设置数值。
 - DI2 () : 命令代码 (0x01)
 - DI3 () : 低位坐标点指定 (0x13= 坐标点 19)
 - DI4 () : 高位坐标点指定 (0x00= 坐标点 19)
- ② 将 DI05 从关闭设置为开启。
- ③ 控制器接收到 IO 命令，如果从命令与命令数据中可执行，则开始执行。此时，执行中输出变为开启，执行判断输出变为关闭。机器人移动至坐标点 19 已指定的位置。
- ④ 确认执行中输出已开启后，清除 DI2 () ~ DI4 ()。
- ⑤ 随着命令的执行结束，将执行中输出从开启设置为关闭，如果执行正常结束，则将执行判断输出从关闭设置为开启。执行异常结束时，执行判断输出将保持关闭状态。

MEMO

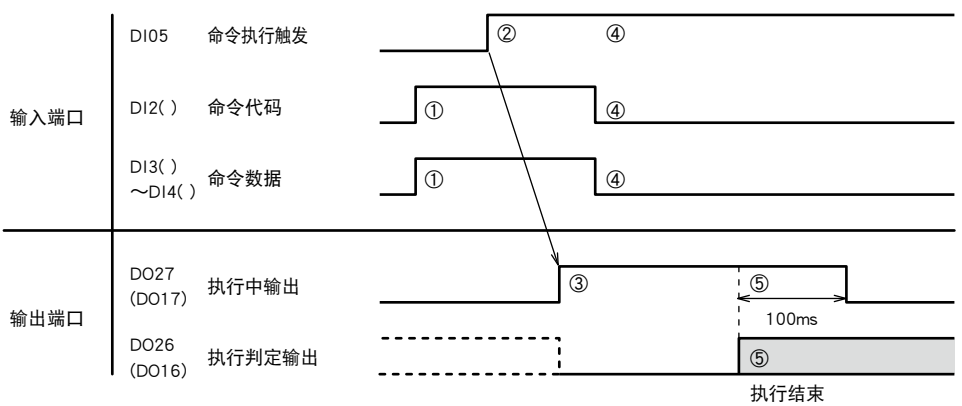
- 不在④中将 DI05 设置为关闭时，执行中输出将从⑤开始保持最大 100ms 的开启状态。



要点

- 水平多关节型机器人且在指定坐标点数据中指定了手系统标志时，此手系统比当前机械臂类型优先级更高。

IO命令收发(2)



*RCX22x的输出端口变为 "DO27→DO17、DO26→DO16"。

IO 命令代码采用十六进制。



要点

- 以副机器人为执行对象的 MOVE 移动的直线插补命令 (0x83) 在以下软件版本中有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

- 返回原点命令在软件版本 Ver.8.45 以下的控制器中，以所有轴（主机器人+副机器人）为对象，命令代码仅有 0x32。

No.	命令内容		命令代码 (DI2 ())	
			主机器人	副机器人
1	MOVE 移动命令	PTP 坐标点指定	0x01	0x81
		直线插补	0x03	0x83
2	MOVEI 移动命令	PTP 坐标点指定	0x09	0x89
3	托盘移动命令	在托盘 0 上 PTP 指定	0x18	0x98
4	微动命令		0x20	0xA0
5	点动命令		0x24	0xA4
6	坐标点示教命令		0x28	0xA8
7	用于绝对式原点复位的移动命令		0x30	0xB0
8	绝对式原点复位命令		0x31	0xB1
9	返回原点命令		0x32	0xB2
10	伺服命令	使能	0x34	0xB4
		禁止	0x35	0xB5
		自由	0x36	0xB6
		接通电源	0x37	
11	手动移动速度更改命令		0x38	0xB8
12	自动移动速度更改命令		0x39	0xB9
13	程序速度更改命令		0x3A	0xBA
14	位移指定更改命令		0x3B	0xBB
15	机械手指定更改命令		0x3C	0xBC
16	机械臂指定更改命令		0x3D	0xBD
17	坐标点显示单位指定命令		0x3E	

备注

- *3 托盘移动命令仅对托盘 0 有效。
- *4 微动、点动、坐标点示教的各命令根据坐标点单位指定状态的不同，其移动方法也有所不同。
- *6 坐标点示教命令根据坐标点单位指定状态的不同，坐标点单位也有所不同。
- *8 绝对式原点复位命令不执行轴指定时，即使命令代码为 0x31、0xB1 中的任意一个，执行对象也为所有轴（主机器人+副机器人）
- *9 返回原点命令在设置 2 台机器人时，对象为主机器人及副机器人。
- *17 坐标点显示单位指定命令的对象为控制器。

5.1

MOVE 移动命令

对机器人进行绝对位置移动

1. PTP 坐标点指定

通过指定坐标点编号，使机器人采用 PTP 动作移动至目标位置。

命令

DI 端口	内容		值
DI2 ()	命令代码	对象为主机器人	0x01
		对象为副机器人	0x81
DI3 ()	坐标点编号		0xpppp
DI4 ()			

值

pppp : 坐标点编号 16 指定比特

指定范围 0 (=0x0000) ~ 9999 (=0x270F)



MEMO

- 在软件版本为 Ver.8.28 以前的控制器上，坐标点编号的指定范围为 0 ~ 4000。
- 无 DI4 () 时，坐标点编号的指定范围为 0 ~ 255。
- 水平多关节型机器人且在指定坐标点数据中指定了手系统标志时，此手系统比当前机械臂类型优先度更高。



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。

2. 直线插补

通过指定坐标点编号，使机器人采用直线插补动作移动至目标位置。

命令

DI 端口	内容		值
DI2 ()	命令代码	对象为主机器人	0x03
		对象为副机器人	0x83
DI3 ()	坐标点编号		0xpppp
DI4 ()			

值

pppp : 坐标点编号 16 指定比特

指定范围 0 (=0x0000) ~ 9999 (=0x270F)



MEMO

- 在软件版本为 Ver.8.28 以前的控制器上，坐标点编号的指定范围为 0 ~ 4000。
- 无 DI4 () 时，坐标点编号的指定范围为 0 ~ 255。
- 水平多关节型机器人且在指定坐标点数据中指定了手系统标志时，此手系统比当前机械臂类型优先度更高。



要点

- 以副机器人为对象的 MOVE 移动的直线插补命令 (0x83) 在以下软件版本中有效。

RCX14x Ver.8.64 以上
RCX22x Ver.9.11 以上

MOVEI 移动命令

对机器人进行相对位置移动

1. PTP 坐标点指定

通过指定坐标点编号，使机器人采用 PTP 动作移动指定距离。

命令

DI 端口	内容		值
DI2 ()	命令代码	对象为主机器人	0x09
		对象为副机器人	0x89
DI3 ()	坐标点编号		0xpppp
DI4 ()			

值

ppppp 用 16 指定比特坐标点编号。
指定范围：0 (=0x0000) ~ 9999 (=0x270F)

MEMO



要点

- 此参数在以下软件版本条件下有效。
RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上
- 在比上述版本更旧版本的控制器上，将变为“复位”动作。

- 中途停止 MOVEI 语句并重新启动时，可在控制器的 "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相对移动。目标位置与再次执行前不同(与以前兼容)。
- 在软件版本为 Ver.8.28 以前的控制器上，坐标点编号的指定范围为 0 ~ 4000。
- 无 DI4() 时，坐标点编号的指定范围为 0 ~ 255。
- 水平多关节型机器人且在指定坐标点数据中指定了手系统标志时，此手系统比当前机械臂类型优先度更高。

托盘移动命令

使机器人对托盘 0 进行托盘移动

1. PTP 指定

通过指定工件位置编号，使机器人采用 PTP 动作移动至目标位置。

命令

DI 端口	内容		值
DI2 ()	命令代码	对象为主机器人	0x18
		对象为副机器人	0x98
DI3 ()	工件位置编号		0xwwwww
DI4 ()			

值

wwwww 通过 16 指定比特工件位置编号。
指定范围：1 (=0x0001) ~ 32767 (=0x7FFF)

MEMO

- 无 DI4() 时，工件位置编号的指定范围为 0 ~ 255。

微动命令

在手动模式中对机器人进行微动

此命令仅能在手动模式中执行。

此外,由于与控制器的坐标点显示单位制联动,当显示单位为脉冲单位时,以轴为单位进行PTP移动,当显示单位为毫米单位时,在正交坐标中进行直线插补移动。

微动的速度取决于手动移动速度。

若要停止微动命令,请将专用输入的联锁信号切换至关闭。在确认停止后,请将专用输入的联锁信号切换至开启。

命令

DI 端口	内容			值
DI2 ()	命令代码		对象为主机器人	0x20
			对象为副机器人	0xA0
DI3 ()	移动轴与方向	bit 0	第 1 轴	tt
		bit 1	第 2 轴	
		bit 2	第 3 轴	
		bit 3	第 4 轴	
		bit 4	第 5 轴	
		bit 5	第 6 轴	
		bit 6	(0 : 固定)	0
		bit 7	方向	d
DI4 ()	未使用			0x00

值 tt : 轴指定.....用 0 ~ 3 指定比特 1 轴
 d : 移动方向1 比特 / 0 : + 方向、1 : - 方向

点动命令

在手动模式中对机器人进行点动

点动距离与手动移动速度联动。仅能在手动模式中执行。

此命令与控制器的坐标点显示单位制联动。因此,当显示单位为脉冲单位时,移动轴将移动手动移动速度值的脉冲量;当显示单位为毫米单位时,在正交坐标中将直线插补移动手动移动速度值 / 100 的数据量。

命令

DI 端口	内容			值
DI2 ()	命令代码		对象为主机器人	0x24
			对象为副机器人	0xA4
DI3 ()	移动轴与方向	bit 0	第 1 轴	tt
		bit 1	第 2 轴	
		bit 2	第 3 轴	
		bit 3	第 4 轴	
		bit 4	第 5 轴	
		bit 5	第 6 轴	
		bit 6	(0 : 固定)	0
		bit 7	方向	d
DI4 ()	未使用			0x00

值 tt : 轴指定.....用 0 ~ 3 指定比特 1 轴
 d : 移动方向1 比特 / 0 : + 方向、1 : - 方向

对指定坐标点编号进行当前机器人位置的示教

此命令中的坐标点数据单位与控制器的坐标点显示单位制联动。

命令

DI 端口	内容		值
DI2 ()	命令代码	对象为主机器人	0x28
		对象为副机器人	0xA8
DI3 ()	坐标点编号		0xpppp
DI4 ()			

值 pppp : 坐标点编号 16 指定比特
指定范围 0 (=0x0000) ~ 9999 (=0x270F)

MEMO

- 在软件版本为 Ver.8.28 以前的控制器上，坐标点编号的指定范围为 0 ~ 4000。
- 无 DI4 () 时，坐标点编号的指定范围为 0 ~ 255。

移动至可进行绝对式原点复位的最近位置

指定轴采用标记指定的绝对式原点复位方式时，将从可执行绝对式原点复位的位置移动至最近的位置。

马达每旋转 1/4 圈，就有可进行绝对式原点复位的位置。

命令

DI 端口	内容			值
DI2 ()	命令代码	对象为主机器人		0x30
		对象为副机器人		0xB0
DI3 ()	移动轴与方向	bit 0	第 1 轴	tt
		bit 1	第 2 轴	
		bit 2	第 3 轴	
		bit 3	第 4 轴	
		bit 4	第 5 轴	
		bit 5	第 6 轴	
		bit 6	(0 : 固定)	0
		bit 7	方向	d
DI4 ()	未使用			0x00

值 tt : 轴指定 用 0 ~ 3 指定比特 1 轴
d : 移动方向 1 比特 / 0 : + 方向、1 : 一方向

绝对式原点复位命令

执行指定轴的绝对式原点复位。

不进行轴指定 (DI3 () 为 0) 时, 即使命令代码为 0x31、0xB1 中的任意一个, 执行对象也为所有轴 (主机器人+副机器人)。

此时, 如果标记指定轴处于未返回原点状态, 将无法执行命令。请对每个轴执行绝对式原点复位。

命令

DI 端口	内容		值
DI2 ()	命令代码	对象为主机器人	0x31
		对象为副机器人	0xB1
DI3 ()	轴指定	bit 0	第 1 轴
		bit 1	第 2 轴
		bit 2	第 3 轴
		bit 3	第 4 轴
		bit 4	第 5 轴
		bit 5	第 6 轴
		bit 7 - bit 6	(0 : 固定)
DI4 ()	未使用		0
			0x00

值

tt : 轴指定.....在 0 ~ 3 比特上指定 1 轴 (如果不指定, 则主机器人+副机器人的所有轴将变为对象)

返回原点命令

对增量轴执行返回原点。

如果对增量式规格的轴执行此命令, 则将执行返回原点动作。同样地, 如果对半绝对式规格的轴执行此命令, 其将进行绝对位置搜索动作。

不进行轴指定 (DI3 () 为 0) 时, 即使命令代码为 0x32、0xB2 中的任意一个, 执行对象也为所有轴 (主机器人+副机器人)。

命令

DI 端口	内容		值
DI2 ()	命令代码	对象为主机器人	0x32
		对象为副机器人	0xB2
DI3 ()	轴指定	bit 0	第 1 轴
		bit 1	第 2 轴
		bit 2	第 3 轴
		bit 3	第 4 轴
		bit 4	第 5 轴
		bit 5	第 6 轴
		bit 7 - bit 6	(0 : 固定)
DI4 ()	未使用		0
			0x00

值

tt : 轴指定.....在 0 ~ 3 比特上指定 1 轴 (如果不指定, 则主机器人+副机器人的所有轴将变为执行对象)



要点

- 此参数在以下软件版本条件下有效。
RCX14x Ver.8.63 以上
RCX22x Ver.9.08 以上
- 在比上述版本更旧版本的控制器上, 将变为“复位”动作。
- 返回原点命令的轴指定在软件版本 Ver.8.45 以上有效。如果在该版本以下, 则轴指定对象仅限所有轴, 命令代码仅限 0x32。

以下命令可操作机器人的伺服，并分别对各轴进行伺服使能、禁止和自由等切换。

操作	内容
使能指定	用于开启指定轴的伺服。进行轴指定时，马达电源必须开启。 未进行轴指定时，将进行所有控制器的伺服使能。
禁止指定	用于禁止指定轴的伺服。 未进行轴指定时，将进行所有控制器的伺服禁止。
自由指定	用于在禁止了指定轴的伺服后关闭机械制动及动态制动。因此，如果连续执行，将反复执行禁止指定+自由指定。 未进行轴指定时，将进行所有控制器的伺服自由。
电源接通指定	用于接通马达电源。 无法进行轴指定。

命令

DI 端口	内容		值
DI2 ()	命令代码	使能指定	对象为主机器人
			对象为副机器人
		禁止指定	对象为主机器人
			对象为副机器人
		自由指定	对象为主机器人
			对象为副机器人
		电源接通指定	所有控制器
DI3 ()	指定	bit 0	第 1 轴
		bit 1	第 2 轴
		bit 2	第 3 轴
		bit 3	第 4 轴
		bit 4	第 5 轴
		bit 5	第 6 轴
		bit 7 - bit 6	(0 : 固定)
DI4 ()	未使用		0
			0x00

值 tt : 轴指定.....在 0 ~ 3 比特上指定 1 轴（如果不指定，则对所有控制器进行处理）
进行了接通电源指定时，无法进行轴指定。

手动移动速度更改命令

在手动模式中更改手动移动速度

此命令仅能在手动模式中执行。

命令

DI 端口	内容	值
DI2 ()	命令代码	
	对象为主机器人	0x38
	对象为副机器人	0xB8
DI3 ()	速度指定	0xss
DI4 ()	未使用	0x00

值

ss : 手动移动速度 用 8 指定比特
指定范围 1 (=0x01) ~ 100 (=0x64)

自动移动速度更改命令

在自动模式中更改自动移动速度

此命令仅能在自动模式中执行。

命令

DI 端口	内容	值
DI2 ()	命令代码	
	对象为主机器人	0x39
	对象为副机器人	0xB9
DI3 ()	速度指定	0xss
DI4 ()	未使用	0x00

值

ss : 自动移动速度 用 8 指定比特
指定范围 1 (=0x01) ~ 100 (=0x64)

程序速度更改命令

在自动模式中更改程序移动速度

如果对此命令中设置的程序速度进行程序复位或程序更改，该速度将被初始化至 100%。

命令

DI 端口	内容	值
DI2 ()	命令代码	
	对象为主机器人	0x3A
	对象为副机器人	0xBA
DI3 ()	速度指定	0xss
DI4 ()	未使用	0x00

值

ss : 程序移动速度 用 8 指定比特
指定范围 1 (=0x01) ~ 100 (=0x64)

位移指定更改命令

将选择位移更改为指定位移编号

命令

DI 端口	内容	值
DI2 ()	命令代码	
	对象为主机器人	0x3B
	对象为副机器人	0xBB
DI3 ()	指定位移编号	0xss
DI4 ()	未使用	0x00

值 ss : 位移编号 用 8 指定比特
指定范围 0 (=0x00) ~ 9 (=0x09)

机械手指定更改命令

将选择机械手更改为指定机械手编号

命令

DI 端口	内容	值
DI2 ()	命令代码	
	对象为主机器人	0x3C
	对象为副机器人	0xBC
DI3 ()	指定机械手编号	0xss
DI4 ()	未使用	0x00

值 ss : 机械手编号 用 8 指定比特
指定范围 对象为主机器人 : 0 (=0x00) ~ 3 (=0x03)
对象为副机器人 : 4 (=0x04) ~ 7 (=0x07)

机械臂指定更改命令

更改机械臂指定状态

命令

DI 端口	内容	值
DI2 ()	命令代码	
	对象为主机器人	0x3D
	对象为副机器人	0xBD
DI3 ()	指定机械臂状态	0xss
DI4 ()	未使用	0x00

值 ss : 机械臂状态 用 8 指定比特 / 0x00 : 右手系统、0x01 : 左手系统

坐标点显示单位指定命令

更改坐标点显示单位制

此命令的执行对象为控制器。

命令

DI 端口	内容	值
DI2 ()	命令代码	0x3E
DI3 ()	指定坐标点显示单位制	0xss
DI4 ()	未使用	0x00

值 ss : 坐标点显示单位制 用 8 指定比特 / 0x00 : 脉冲单位制、0x01 : 毫米单位

第 14 章

附录

1	保留字一览表	14-1
2	机器人语言一览表：按字母顺序.....	14-3
3	机器人语言一览表：按功能	14-8
4	函数：按字母顺序	14-15
5	函数：按功能	14-18
6	执行级别.....	14-20

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	CONFIG	FLIP	LOCY
ABSADJ	CONT	FN	LOCZ
ABSADJ2	COO	FOR	LOG
ABSRESET	COPY	FREE	LOOP
ABSRESET2	COS	FUNCTION	LSHIFT
ABSRPOS	CUT	G	M
ABSRPOS2	D	GASP	MANUAL
ABSRST	DATE	GEN	MCHREF
ABOVE	DEC	GO	MCHREF2
ABS	DECEL	GOHOME	MEM
ACC	DECEL2	GOSUB	MID
ACCEL	DECLARE	GOTO	MIRROR
ACCEL2	DEF	H	MOD
ACCESS	DEFIO	HALT	MOVE
ALL	DEFPOS	HAND	MOVE2
AND	DEGRAD	HAND2	MOVEI
ARCH	DELAY	HND	MOVEI2
ARCH2	DI	HEX	MRF
ARM	DIM	HOME	MSG
ARM2	DIR	HOLD	MSGCLR
ARMTYPE	DIST	I	MSPEED
ARMTYPE2	DO	IF	MSPEED2
ARY	DPM	IN	N
ASPEED	DRIVE	INCH	NAME
ASPEED2	DRIVE2	INCH2	NEXT
ATN	DRIVEI	INIT	NONFLIP
ATN2	DRIVEI2	INPUT	NOT
ATTR	DS	INT	O
AUTO	DSPEED	INTEGER	OFF
AXWGHT	E	IRET	OFFLINE
AXWGHT2	ELSE	J	ON
B	ELSEIF	JTOXY	ONLINE
BELOW	EMGRST	JTOXY2	OR
BIN	EMG	JOG	ORD
BIT	END	JOG2	ORGORD
BREAK	ENDIF	L	ORGORD2
BYTE	EOF	LANGUAGE	ORGRTN
C	ERA	LEFT	ORGRTN2
CALL	ERL	LEFTY	ORIGIN
CASE	ERR	LEFTY2	OUT
CHANGE	ERROR	LEN	OUTPOS
CHANGE2	ETH	LET	OUTPOS2
CHGPRI	EXELVL	LOCA	P
CHGTSK	EXIT	LOCB	P
CHR	EXITTASK	LOCF	PASS
CMU	F	LOCR	PADDR
CMU1	FDD	LOCX	PATH

PC	RO	STEP	UNTIL
PCM	ROTATE	STOP	V
PDEF	RSHIFT	STOPON	VAL
PLN	RUN	STR	VAR
PLT	RW	SUB	VER
PMOVE	S	SUSPEND	VEL
PMOVE2	S	SWI	W
PRINT	SELECT	SYS	WAIT
PROGRAM	SEND	SYSTEM	WEIGHT
PGM	SEQUENCE	T	WEIGHT2
PNT	SERVO	TAN	WEND
POS	SERVO2	TASK	WHERE
PPNT	SET	TASKS	WHERE2
PRM	SFT	TCOUNTER	WHILE
PTP	SGI	TEACH	WHRXY
PWR	SGR	TEACH2	WHRXY2
R	SHARED	THEN	WORD
RADDEG	SHIFT	TIME	WRITE
READ	SHIFT2	TIMER	X
REF	SI	TO	XOR
REN	SID	TOLE	XY
RELESE	SIN	TOLE2	XYTOJ
REM	SIW	TORQUE	XYTOJ2
REMOTE	SKIP	TORQUE2	Y
RESET	SO	TRQSTS	YZ
RESTART	SOD	TRQSTS2	Z
RESUME	SOW	TRQTIME	ZX
RETURN	SPEED	TRQTIME2	SYMBOL
RIGHT	SPEED2	TSKMON	_SYSFLG
RIGHTY	SQR	U	
RIGHTY2	START	UNIT	

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）。

DIn	Hn	Pn	SO _n
DO _n	LO _n	SIn	TO _n
FN	MO _n	Sn	

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	条件	直接	种类
A					
1	ABS	计算指定值的绝对值	-	-	函数
2	ABSRPOS	计算主组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
2	ABSRPOS2	计算副组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
3	ABSRST	执行机器人的绝对马达轴返回原点动作	4	○	命令
4	ACCEL	设置 / 获取主组的加速度系数参数	4/-	○	命令 / 函数
4	ACCEL2	设置 / 获取副组的加速度系数参数	4/-	○	命令 / 函数
5	ARCH	设置 / 获取主组的圆弧位置参数	4/-	○	命令 / 函数
5	ARCH2	设置 / 获取副组的圆弧位置参数	4/-	○	命令 / 函数
6	ARMCND	获取主机器人当前机械臂的状态	-	-	函数
6	ARMCND2	获取副机器人当前机械臂的状态	-	-	函数
7	ARMTYPE	获取主机器人当前手系统的设置	-	-	函数
7	ARMTYPE2	获取副机器人当前手系统的设置	-	-	函数
9	ASPEED	更改主组的自动移动速度	4	○	命令
9	ASPEED2	更改副组的自动移动速度	4	○	命令
8	ATN	计算指定值的反正切值	-	-	函数
10	AXWGHT	设置 / 获取主组的轴前端重量参数	4/-	○	命令 / 函数
10	AXWGHT2	设置 / 获取副组的轴前端重量参数	4/-	○	命令 / 函数
C					
11	CALL	执行其他程序	6	×	命令
12	CHANGE	进行主机器人的机械手切换	4	○	命令
12	CHANGE2	进行副机器人的机械手切换	4	○	命令
13	CHGPRI	更改指定任务的优先级	6	×	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	CUT	强制结束执行中或暂停中的其他任务	6	×	命令
D					
17	DATE\$	通过 "yy/mm/dd" 形式的字符串计算日期	-	-	函数
18	DECEL	设置 / 获取主组的减速比率参数	4/-	○	命令 / 函数
18	DECEL2	设置 / 获取副组的减速比率参数	4/-	○	命令 / 函数
21	DEGRAD	将值转换为弧度 (↔ RADDEG)	-	-	函数
22	DELAY	仅等待指定的时间 (单位 ms)	6	×	命令
25	DIM	对数组变量的名称及元素数进行声明	6	×	命令
24	DIST	计算指定 2 个点之间的距离	-	-	函数
26	DO	将指定数值输出至 DO 端口	1	○	命令
27	DRIVE	以轴为单位对主组进行绝对位置移动	4	○	命令
27	DRIVE	(带有 T 选项时) 执行轴单位的绝对移动命令	4	○	命令
27	DRIVE2	以轴为单位对副组进行绝对位置移动	4	○	命令
28	DRIVEI	以轴为单位对主组进行相对位置移动	4	○	命令
28	DRIVEI2	以轴为单位对副组进行相对位置移动	4	○	命令

No.	命令	功能	条件	直接	种类
E					
31	ERL	赋予发生错误的行编号	-	-	函数
31	ERR	赋予发生错误时的错误代码编号	-	-	函数
32	EXIT FOR	强制结束 FOR 语句～NEXT 语句的循环	6	×	命令
34	EXIT TASK	结束当前正在执行的任务	6	×	命令
F					
35	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	6	×	命令
G					
36	GOSUB ～ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	6	×	命令
37	GOTO	无条件跳转至标签所指定的行	6	×	命令
H					
38	HALT	停止程序并进行复位	6	×	命令
39	HAND	定义主机器人的机械手	4	○	命令
39	HAND2	定义副机器人的机械手	4	○	命令
40	HOLD	暂停程序	6	×	命令
I					
41	IF	根据条件分支控制流程	6	×	命令
42	INPUT	从手持编程器中对指定的变量赋值	1	○	命令
43	INT	获得舍去小数点以下部分的整数值	-	-	函数
J					
44	JTOXY	将关节坐标数据转换为主组的正交坐标数据 (↔ XYTOJ)	-	-	函数
44	JTOXY2	将关节坐标数据转换为副组的正交坐标数据 (↔ XYTOJ2)	-	-	函数
L					
46	LEFT\$	从指定的字符串左侧抽出指定位数的字符串	-	-	函数
47	LEFTY	将主机器人的手系统设置为左手系统	4	○	命令
47	LEFTY2	将副机器人的手系统设置为左手系统	4	○	命令
48	LEN	获得指定字符串的长度(字节数)	-	-	函数
49	LET	执行指定的赋值语句	1	○	命令
50	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	1	○	命令
51	LOCx	以轴为单位设置 / 获取坐标点数据或位移数据	-	-	命令 / 函数
52	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	-	函数
M					
53	MCHREF	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
53	MCHREF2	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
54	MID\$	从指定的字符串中抽出任意长度的字符串	-	-	函数
55	MO	将指定值输出至 MO 端口	1	○	命令
56	MOVE	对主组机器人的所有轴进行绝对移动	5	○	命令
56	MOVE2	对副组机器人的所有轴进行绝对移动	5	○	命令
57	MOVEI	对主组机器人的所有轴进行相对移动	4	○	命令

No.	命令	功能	条件	直接	种类
57	MOVEI2	对副组机器人的所有轴进行相对移动	4	○	命令
O					
58	OFFLINE	将指定的通信端口设置为脱机模式	1	○	命令
60	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	6	×	命令
61	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	6	×	命令
62	ON ~ GOTO	根据条件跳转至标签所指定的各行	6	×	命令
63	ONLINE	将指定的通信端口设置为联机模式	1	○	命令
59	ORD	获得指定字符串起始字符的字符编码	-	-	函数
64	ORGORD	设置 / 获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
64	ORGORD2	设置 / 获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
65	ORIGIN	对增量式规格的轴执行返回原点动作	4	○	命令
66	OUT	开启指定输出端口的比特，并结束命令语句	6	×	命令
67	OUTPOS	设置 / 获取主组的 OUT 有效位置参数	4/-	○	命令 / 函数
67	OUTPOS2	设置 / 获取副组的 OUT 有效位置参数	4/-	○	命令 / 函数
P					
68	PATH	设置 PATH 移动路径	6	×	命令
69	PATH END	结束 PATH 移动路径的设置	6	×	命令
70	PATH SET	开始 PATH 移动路径的设置	6	×	命令
71	PATH START	开始 PATH 移动	6	×	命令
72	PDEF	定义用于执行托盘移动命令的托盘	1	○	命令
73	PMOVE	执行主机器人的托盘移动命令	4	○	命令
73	PMOVE2	执行副机器人的托盘移动命令	4	○	命令
74	Pn	在程序中定义坐标点	1	○	命令
75	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
76	PRINT	在手持编程器的画面上显示字符串。	1	○	命令
R					
77	RADDEG	将值转换为度 (↔ DEGRAD)	-	-	函数
78	REM	记述注释语句	6	×	命令
79	RESET	关闭指定输出端口的比特	1	○	命令
80	RESTART	重新启动暂停中的其他任务	6	×	命令
81	RESUME	进行错误的恢复处理后，恢复执行程序	6	×	命令
83	RIGHT\$	从指定的字符串右侧抽出指定位数的字符串	-	-	函数
84	RIGHTY	将主机器人的手系统设置为右手系统	4	○	命令
84	RIGHTY2	将副机器人的手系统设置为右手系统	4	○	命令
85	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	-	函数
S					
87	SELECT CASE ~ END SELECT	根据条件分支控制流程	6	×	命令
88	SEND	转送文件	1	○	命令
89	SERVO	控制主组指定轴或所有轴的伺服 ON / OFF	4	○	命令
89	SERVO2	控制副组指定轴或所有轴的伺服 ON / OFF	4	○	命令

No.	命令	功能	条件	直接	种类
90	SET	开启指定输出端口的比特	3	部分 ×	命令
92	SHIFT	指定位移变量，并在该处指定的位移数据上设置主机机器人的位移坐标	4	○	命令
92	SHIFT2	指定位移变量，并在该处指定的位移数据上设置副机器人的位移坐标	4	○	命令
93	SIN	计算指定值的正弦值	-	-	函数
86	Sn	在程序中定义位移坐标	4	○	命令
94	SO	将指定数值输出至 SO 端口	1	○	命令
95	SPEED	更改主组的程序移动速度	4	○	命令
95	SPEED2	更改副组的程序移动速度	4	○	命令
98	SQR	计算指定值的平方根	-	-	函数
96	START	设置指定任务的任务编号及优先级，并启动该任务	6	×	命令
97	STR\$	将指定数值转换为字符串 (↔ VAL)	-	-	函数
100	SUSPEND	暂停执行中的其他任务	6	×	命令
101	SWI	切换执行程序，并在执行编译后从第 1 行开始执行	2	○	命令
T					
102	TAN	计算指定值的正切值	-	-	函数
103	TCOUNTER	从 TCOUNTER 变量复位时开始，每隔 10ms 输出计数值	-	-	函数
104	TIME\$	通过 "hh : mm : ss" 形式的字符串计算当前时间	-	-	函数
105	TIMER	用上午 0 点开始的秒计算当前时间	-	-	函数
106	TO	将指定值输出至 TO 端口	1	○	命令
107	TOLE	设置 / 获取主组的公差参数	4/-	○	命令 / 函数
107	TOLE2	设置 / 获取副组的公差参数	4/-	○	命令 / 函数
108	TORQUE	设置 / 获取主组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
108	TORQUE2	设置 / 获取副组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
109	TRQSTS	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
109	TRQSTS2	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
110	TRQTIME	设置 / 获取主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数
110	TRQTIME2	设置 / 获取副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数
V					
111	VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)	-	-	函数
W					
112	WAIT	等待机器人轴动作结束	6	×	命令
112	WAIT	等待直至 DI / DO 条件表达式成立为止 (带超时)	6	×	命令
114	WEIGHT	设置 / 获取主机机器人的前端重量参数	4/-	○	命令 / 函数
114	WEIGHT2	设置 / 获取副机器人的前端重量参数	4/-	○	命令 / 函数
116	WHERE	用关节坐标 (脉冲) 读出主组机械臂的当前位置	-	-	函数
116	WHERE2	用关节坐标 (脉冲) 读出副组机械臂的当前位置	-	-	函数
117	WHILE ~ WEND	控制反复	6	×	命令
118	WHRXY	用正交坐标 (mm、度) 读出主组机械臂的当前位置	-	-	函数
118	WHRXY2	用正交坐标 (mm、度) 读出副组机械臂的当前位置	-	-	函数

No.	命令	功能	条件	直接	种类
X					
119	XYTOJ	将坐标点变量的正交坐标数据转换为主组的关节坐标数据 (↔ JTOXY)	-	-	函数
119	XYTOJ2	将坐标点变量的正交坐标数据转换为副组的关节坐标数据 (↔ JTOXY2)	-	-	函数
120	_SYSFLG	监视轴状态的标志	-	-	函数

8

9

10

11

12

13

14

15

程序命令

普通命令

No.	命令	功能	条件	直接	种类
25	DIM	对数组变量的名称及元素数进行声明	6	×	命令
49	LET	执行指定的赋值语句	1	○	命令
78	REM	记述注释语句	6	×	命令

算术命令

No.	命令	功能	条件	直接	种类
1	ABS	计算指定值的绝对值	-	-	函数
8	ATN	计算指定值的反正切值	-	-	函数
15	COS	计算指定值的余弦值	-	-	函数
21	DEGRAD	将值转换为弧度 (↔ RADDEG)	-	-	函数
24	DIST	计算指定 2 个点之间的距离	-	-	函数
43	INT	获得舍去小数点以下部分的整数值	-	-	函数
52	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	-	-	函数
77	RADDEG	将值转换为度 (↔ DEGRAD)	-	-	函数
85	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)	-	-	函数
93	SIN	计算指定值的正弦值	-	-	函数
98	SQR	计算指定值的平方根	-	-	函数
102	TAN	计算指定值的正切值	-	-	函数

日期、时间

No.	命令	功能	条件	直接	种类
17	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期	-	-	函数
103	TCOUNTER	从 TCOUNTER 变量复位时开始，每隔 10ms 输出计数值	-	-	函数
104	TIME \$	通过 "hh:mm:ss" 形式的字符串计算当前时间	-	-	函数
105	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	条件	直接	种类
14	CHR \$	获取带有指定字符编码的字符	-	-	函数
46	LEFT \$	从指定的字符串左侧抽出指定位数的字符串	-	-	函数
48	LEN	获得指定字符串的长度(字节数)	-	-	函数
54	MID \$	从指定的字符串中抽出任意长度的字符串	-	-	函数
59	ORD	获得指定字符串起始字符的字符编码	-	-	函数
83	RIGHT \$	从指定的字符串右侧抽出指定位数的字符串	-	-	函数
97	STR \$	将指定数值转换为字符串(↔ VAL)	-	-	函数
111	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	条件	直接	种类
12	CHANGE	进行主机器人的机械手切换	4	○	命令
12	CHANGE2	进行副机器人的机械手切换	4	○	命令
39	HAND	定义主机器人的机械手	4	○	命令
39	HAND2	定义副机器人的机械手	4	○	命令
44	JTOXY	将关节坐标数据转换为主组的正交坐标数据(↔ XYTOJ)	-	-	函数
44	JTOXY2	将关节坐标数据转换为副组的正交坐标数据(↔ XYTOJ2)	-	-	函数
47	LEFTY	将主机器人的手系统设置为左手系统	4	○	命令
47	LEFTY2	将副机器人的手系统设置为左手系统	4	○	命令
74	Pn	在程序中定义坐标点	1	○	命令
75	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
84	RIGHTY	将主机器人的手系统设置为右手系统	4	○	命令
84	RIGHTY2	将副机器人的手系统设置为右手系统	4	○	命令
86	Sn	在程序中定义位移坐标	4	○	命令
92	SHIFT	指定位移变量,并在该处指定的位移数据上设置主机器人的位移坐标	4	○	命令
92	SHIFT2	指定位移变量,并在该处指定的位移数据上设置副机器人的位移坐标	4	○	命令
119	XYTOJ	将坐标点变量的正交坐标数据转换为主组的关节坐标数据(↔ JTOXY)	-	-	函数
119	XYTOJ2	将坐标点变量的正交坐标数据转换为副组的关节坐标数据(↔ JTOXY2)	-	-	函数
51	LOCx	以轴为单位设置/获取坐标点数据或位移数据	-	-	命令 / 函数

分支命令

No.	命令	功能	条件	直接	种类
32	EXIT FOR	强制结束 FOR 语句～NEXT 语句的循环	6	×	命令
35	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	6	×	命令
36	GOSUB ～ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	6	×	命令
37	GOTO	无条件跳转至标签所指定的行	6	×	命令
41	IF	根据条件分支控制流程	6	×	命令
61	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	6	×	命令
62	ON ～ GOTO	根据条件跳转至标签所指定的各行	6	×	命令
87	SELECT CASE ～ END SELECT	根据条件分支控制流程	6	×	命令
117	WHILE ～ WEND	控制反复	6	×	命令

错误控制

No.	命令	功能	条件	直接	种类
60	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	6	×	命令
81	RESUME	进行错误的恢复处理后，恢复执行程序	6	×	命令
31	ERL	赋予发生错误的行编号	-	-	函数
31	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	条件	直接	种类
11	CALL	执行其他程序	6	×	命令
38	HALT	停止程序并进行复位	6	×	命令
40	HOLD	暂停程序	6	×	命令
101	SWI	切换执行程序，并在执行编译后从第 1 行开始执行	2	○	命令

任务控制

No.	命令	功能	条件	直接	种类
13	CHGPRI	更改指定任务的优先级	6	×	命令
16	CUT	强制结束执行中或暂停中的其他任务	6	×	命令
34	EXIT TASK	结束当前正在执行的任务	6	×	命令
80	RESTART	重新启动暂停中的其他任务	6	×	命令
96	START	设置指定任务的任务编号及优先级，并启动该任务	6	×	命令
100	SUSPEND	暂停执行中的其他任务	6	×	命令

机器人控制

机器人动作

No.	命令	功能	条件	直接	种类
3	ABSRST	执行机器人的绝对式马达轴返回原点动作	4	○	命令
12	CHANGE	进行主机器人的机械手切换	4	○	命令
12	CHANGE2	进行副机器人的机械手切换	4	○	命令
27	DRIVE	以轴为单位对主组进行绝对位置移动	4	○	命令
27	DRIVE2	以轴为单位对副组进行绝对位置移动	4	○	命令
28	DRIVEI	以轴为单位对主组进行相对位置移动	4	○	命令
28	DRIVEI2	以轴为单位对副组进行相对位置移动	4	○	命令
39	HAND	定义主机器人的机械手	4	○	命令
39	HAND2	定义副机器人的机械手	4	○	命令
47	LEFTY	将主机器人的手系统设置为左手系统	4	○	命令
47	LEFTY2	将副机器人的手系统设置为左手系统	4	○	命令
56	MOVE	对主组机器人的所有轴进行绝对移动	5	○	命令
56	MOVE2	对副组机器人的所有轴进行绝对移动	5	○	命令
57	MOVEI	对主组机器人的所有轴进行相对移动	4	○	命令
57	MOVEI2	对副组机器人的所有轴进行相对移动	4	○	命令
65	ORIGIN	对增量式规格的轴执行返回原点动作	4	○	命令
73	PMOVE	执行主机器人的托盘移动命令	4	○	命令
73	PMOVE2	执行副机器人的托盘移动命令	4	○	命令
84	RIGHTY	将主机器人的手系统设置为右手系统	4	○	命令
84	RIGHTY2	将副机器人的手系统设置为右手系统	4	○	命令
89	SERVO	控制主组指定轴或所有轴的伺服 ON / OFF	4	○	命令
89	SERVO2	控制副组指定轴或所有轴的伺服 ON / OFF	4	○	命令

状态获取

No.	命令	功能	条件	直接	种类
2	ABSRPOS	计算主组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
2	ABSRPOS2	计算副组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）	-	-	函数
6	ARMCND	获取主机器人当前机械臂的状态	-	-	函数
6	ARMCND2	获取副机器人当前机械臂的状态	-	-	函数
7	ARMTYPE	获取主机器人当前手系统的设置	-	-	函数
7	ARMTYPE2	获取副机器人当前手系统的设置	-	-	函数
53	MCHREF	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
53	MCHREF2	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照	-	-	函数
109	TRQSTS	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
109	TRQSTS2	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态	-	-	函数
116	WHERE	用关节坐标（脉冲）读出主组机械臂的当前位置	-	-	函数
116	WHERE2	用关节坐标（脉冲）读出副组机械臂的当前位置	-	-	函数

No.	命令	功能	条件	直接	种类
118	WHRXY	用正交坐标 (mm、度) 读出主组机械臂的当前位置	-	-	函数
118	WHRXY2	用正交坐标 (mm、度) 读出副组机械臂的当前位置	-	-	函数
112	WAIT	等待机器人轴动作结束	6	×	命令

状态更改

No.	命令	功能	条件	直接	种类
4	ACCEL	设置 / 获取主组的加速度系数参数	4/-	○	命令 / 函数
4	ACCEL2	设置 / 获取副组的加速度系数参数	4/-	○	命令 / 函数
5	ARCH	设置 / 获取主组的圆弧位置参数	4/-	○	命令 / 函数
5	ARCH2	设置 / 获取副组的圆弧位置参数	4/-	○	命令 / 函数
9	ASPEED	更改主组的自动移动速度	4	○	命令
9	ASPEED2	更改副组的自动移动速度	4	○	命令
10	AXWGHT	设置 / 获取主组的轴前端重量参数	4/-	○	命令 / 函数
10	AXWGHT2	设置 / 获取副组的轴前端重量参数	4/-	○	命令 / 函数
18	DECEL	设置 / 获取主组的减速比率参数	4/-	○	命令 / 函数
18	DECEL2	设置 / 获取副组的减速比率参数	4/-	○	命令 / 函数
64	ORGORD	设置 / 获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
64	ORGORD2	设置 / 获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数	4/-	○	命令 / 函数
67	OUTPOS	设置 / 获取主组的 OUT 有效位置参数	4/-	○	命令 / 函数
67	OUTPOS2	设置 / 获取副组的 OUT 有效位置参数	4/-	○	命令 / 函数
72	PDEF	定义用于执行托盘移动命令的托盘	1	○	命令
95	SPEED	更改主组的程序移动速度	4	○	命令
95	SPEED2	更改副组的程序移动速度	4	○	命令
107	TOLE	设置 / 获取主组的公差参数	4/-	○	命令 / 函数
107	TOLE2	设置 / 获取副组的公差参数	4/-	○	命令 / 函数
114	WEIGHT	设置 / 获取主机器人的前端重量参数	4/-	○	命令 / 函数
114	WEIGHT2	设置 / 获取副机器人的前端重量参数	4/-	○	命令 / 函数

PATH 控制

No.	命令	功能	条件	直接	种类
68	PATH	设置 PATH 移动路径	6	×	命令
69	PATH END	结束 PATH 移动路径的设置	6	×	命令
70	PATH SET	开始 PATH 移动路径的设置	6	×	命令
71	PATH START	开始 PATH 移动	6	×	命令

扭矩控制

No.	命令	功能	条件	直接	种类
27	DRIVE	(带有 T 选项时) 执行轴单位的绝对移动命令	4	○	命令
108	TORQUE	设置 / 获取主组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
108	TORQUE2	设置 / 获取副组指定轴的最大扭矩指令值	4/-	○	命令 / 函数
110	TRQTIME	设置 / 获得主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数
110	TRQTIME2	设置 / 获得副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间	1/-	○	命令 / 函数

输入输出、通信控制

输入输出控制

No.	命令	功能	条件	直接	种类
22	DELAY	仅等待指定的时间(单位 ms)	6	×	命令
26	DO	将指定值输出至 DO 端口	1	○	命令
50	LO	将被指定值输出至 LO 端口, 并进行轴移动的禁止或解除	1	○	命令
55	MO	将指定值输出至 MO 端口	1	○	命令
66	OUT	开启指定输出端口的比特, 并结束命令语句	6	×	命令
79	RESET	关闭指定输出端口的比特	1	○	命令
90	SET	开启指定输出端口的比特	3	部分 ×	命令
94	SO	将指定值输出至 SO 端口	1	○	命令
106	TO	将指定值输出至 TO 端口	1	○	命令
112	WAIT	等待直至 DI/DO 条件表达式成立为止(带超时)	6	×	命令

手持编程器

No.	命令	功能	条件	直接	种类
42	INPUT	从手持编程器中对指定的变量赋值	1	○	命令
76	PRINT	在手持编程器的画面上显示字符串	1	○	命令

通信控制

No.	命令	功能	条件	直接	种类
63	ONLINE	将指定的通信端口设置为联机模式	1	○	命令
58	OFFLINE	将指定的通信端口设置为脱机模式	1	○	命令
88	SEND	转送文件	1	○	命令

其他

其他

No.	命令	功能	条件	直接	种类
120	_SYSFLG	监视轴状态的标志	-	-	函数

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	计算主组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）
2	ABSRPOS2	算术函数	计算副组指定轴的机器参照值（返回原点的方式仅为标记方式时有效）
4	ACCEL	算术函数	获取主组的加速度系数参数
4	ACCEL2	算术函数	获取副组的加速度系数参数
5	ARCH	算术函数	获取主组的圆弧位置参数
5	ARCH2	算术函数	获取副组的圆弧位置参数
6	ARMCND	算术函数	获取主机器人当前机械臂的状态
6	ARMCND2	算术函数	获取副机器人当前机械臂的状态
7	ARMTYPE	算术函数	获取主机器人当前手系统的设置
7	ARMTYPE2	算术函数	获取副机器人当前手系统的设置
8	ATN	算术函数	计算指定值的反正切值
10	AXWGHT	算术函数	获取主组的轴前端重量参数
10	AXWGHT2	算术函数	获取副组的轴前端重量参数
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
D			
17	DATE\$	字符串函数	通过 "yy/mm/dd" 形式的字符串计算日期
18	DECEL	算术函数	获取主组的减速比率参数
18	DECEL2	算术函数	获取副组的减速比率参数
21	DEGRAD	算术函数	将值转换为弧度（↔ RADDEG）
24	DIST	算术函数	计算指定 2 个点之间的距离
E			
31	ERL	算术函数	赋予发生错误的行编号
31	ERR	算术函数	赋予发生错误时的错误代码编号
I			
43	INT	算术函数	获得舍去小数点以下部分的整数值
J			
44	JTOXY	坐标点函数	将关节坐标数据转换为主组的正交坐标数据（↔ XYTOJ）
44	JTOXY2	坐标点函数	将关节坐标数据转换为副组的正交坐标数据（↔ XYTOJ2）
L			
46	LEFT\$	字符串函数	从指定的字符串左侧抽出指定位数的字符串
48	LEN	算术函数	获得指定字符串的长度（字节数）
51	LOCx	坐标点函数	以轴为单位获取坐标点数据或位移数据
52	LSHIFT	算术函数	仅将指定值的比特数向左偏移（↔ RSHIFT）

No.	函数	种类	功能
M			
53	MCHREF	算术函数	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照
53	MCHREF2	算术函数	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照
54	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			
59	ORD	算术函数	获得指定字符串起始字符的字符编码
64	ORGORD	算术函数	获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数
64	ORGORD2	算术函数	获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数
67	OUTPOS	算术函数	获取主组的 OUT 有效位置参数
67	OUTPOS2	算术函数	获取副组的 OUT 有效位置参数
P			
75	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
R			
77	RADDEG	算术函数	将值转换为度 (↔ DEGRAD)
83	RIGHT\$	字符串函数	从指定的字符串右侧抽出指定位数的字符串
85	RSHIFT	算术函数	仅将指定值的比特数向右偏移 (↔ LSHIFT)
S			
93	SIN	算术函数	计算指定值的正弦值
98	SQR	算术函数	计算指定值的平方根
97	STR\$	字符串函数	将指定值转换为字符串 (↔ VAL)
T			
102	TAN	算术函数	计算指定值的正切值
103	TCOUNTER	算术函数	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数值
104	TIME\$	字符串函数	通过 "hh:mm:ss" 形式的字符串计算当前时间
105	TIMER	算术函数	用上午 0 点开始的秒计算当前时间
107	TOLE	算术函数	获取主组的公差参数
107	TOLE2	算术函数	获取副组的公差参数
108	TORQUE	算术函数	获取主组指定轴的最大扭矩指令值
108	TORQUE2	算术函数	获取副组指定轴的最大扭矩指令值
109	TRQSTS	算术函数	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态
109	TRQSTS2	算术函数	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态
110	TRQTIME	算术函数	获得主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间
110	TRQTIME2	算术函数	获得副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间
V			
111	VAL	算术函数	将指定字符串标注的数值转换为实际的数值 (↔ STR \$)
W			
114	WEIGHT	算术函数	获取主机器人的前端重量参数
114	WEIGHT2	算术函数	获取副机器人的前端重量参数
116	WHERE	坐标点函数	用关节坐标 (脉冲) 读出主组机械臂的当前位置
116	WHERE2	坐标点函数	用关节坐标 (脉冲) 读出副组机械臂的当前位置
118	WHRXY	坐标点函数	用正交坐标 (mm、度) 读出主组机械臂的当前位置
118	WHRXY2	坐标点函数	用正交坐标 (mm、度) 读出副组机械臂的当前位置

No.	函数	种类	功能
X			
119	XYTOJ	坐标点函数	将坐标点变量的正交坐标数据转换为主组的关节坐标数据 ($\leftrightarrow$ JTOXY)
119	XYTOJ2	坐标点函数	将坐标点变量的正交坐标数据转换为副组的关节坐标数据 ($\leftrightarrow$ JTOXY2)
120	_SYSFLG	算术函数	监视轴状态的标志

相关坐标点

No.	函数名	功能
44	JTOXY	将关节坐标数据转换为主组的正交坐标数据 (↔ XYTOJ)
44	JTOXY2	将关节坐标数据转换为副组的正交坐标数据 (↔ XYTOJ2)
51	LOCx	以轴为单位获取坐标点数据或位移数据
75	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
116	WHERE	用关节坐标 (脉冲) 读出主组机械臂的当前位置
116	WHERE2	用关节坐标 (脉冲) 读出副组机械臂的当前位置
118	WHRXY	用正交坐标 (mm、度) 读出主组机械臂的当前位置
118	WHRXY2	用正交坐标 (mm、度) 读出副组机械臂的当前位置
119	XYTOJ	将坐标点变量的正交坐标数据转换为主组的关节坐标数据 (↔ JTOXY)
119	XYTOJ2	将坐标点变量的正交坐标数据转换为副组的关节坐标数据 (↔ JTOXY2)

相关参数

No.	函数名	功能
2	ABSRPOS	计算主组指定轴的机器参照值 (返回原点的方式仅为标记方式时有效)
2	ABSRPOS2	计算副组指定轴的机器参照值 (返回原点的方式仅为标记方式时有效)
4	ACCEL	获取主组的加速度系数参数
4	ACCEL2	获取副组的加速度系数参数
5	ARCH	获取主组的圆弧位置参数
5	ARCH2	获取副组的圆弧位置参数
6	ARMCND	获取主机器人当前机械臂的状态
6	ARMCND2	获取副机器人当前机械臂的状态
7	ARMTYPE	获取主机器人当前手系统的设置
7	ARMTYPE2	获取副机器人当前手系统的设置
10	AXWGHT	获取主组的轴前端重量参数
10	AXWGHT2	获取副组的轴前端重量参数
18	DECEL	获取主组的减速比率参数
18	DECEL2	获取副组的减速比率参数
48	LEN	获得指定字符串的长度 (字节数)
53	MCHREF	计算主组轴的返回原点动作及绝对位置搜索动作的机器参照
53	MCHREF2	计算副组轴的返回原点动作及绝对位置搜索动作的机器参照
59	ORD	获得指定字符串起始字符的字符编码
64	ORGORD	获取执行主组的返回原点动作及绝对位置搜索动作的轴顺序参数
64	ORGORD2	获取执行副组的返回原点动作及绝对位置搜索动作的轴顺序参数
67	OUTPOS	获取主组的 OUT 有效位置参数
67	OUTPOS2	获取副组的 OUT 有效位置参数
107	TOLE	获取主组的公差参数
107	TOLE2	获取副组的公差参数
108	TORQUE	获取主组指定轴的最大扭矩指令值
108	TORQUE2	获取副组指定轴的最大扭矩指令值
109	TRQSTS	获取主组带有扭矩限制指定选项的 DRIVE 命令的结束状态
109	TRQSTS2	获取副组带有扭矩限制指定选项的 DRIVE 命令的结束状态
110	TRQTIME	获得主组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间
110	TRQTIME2	获得副组中使用 DRIVE 语句的扭矩限制指定选项时对指定轴电流限制的超时时间

No.	函数名	功能
114	WEIGHT	获取主机器人的前端重量参数
114	WEIGHT2	获取副机器人的前端重量参数

相关数值运算

No.	函数名	功能
1	ABS	计算指定值的绝对值
8	ATN	计算指定值的反正切值
15	COS	计算指定值的余弦值
21	DEGRAD	将值转换为弧度 (↔ RADDEG)
24	DIST	计算指定 2 个点之间的距离
43	INT	获得舍去数值小数点以下部分的整数值
52	LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)
77	RADDEG	将数值转换为度 (↔ DEGRAD)
85	RSHIFT	仅将指定值的比特数向右偏移 (↔ LSHIFT)
93	SIN	计算指定值的正弦值
98	SQR	计算指定值的平方根
102	TAN	计算指定值的正切值
111	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
17	DATE \$	通过 "yy/mm/dd" 形式的字符串计算日期
46	LEFT \$	从指定的字符串左侧抽出指定位数的字符串
54	MID \$	从指定的字符串中抽出任意长度的字符串
83	RIGHT \$	从指定的字符串右侧抽出指定位数的字符串
97	STR \$	将指定值转换为字符串 (↔ VAL)

相关参数

No.	函数名	功能
120	_SYSFLG	监视轴状态的标志
31	ERL	赋予发生错误的行编号
31	ERR	赋予发生错误时的错误代码编号
103	TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数值
104	TIME \$	通过 "hh:mm:ss" 形式的字符串计算当前时间
105	TIMER	用上午 0 点开始的秒计算当前时间

如下所示可设置用于执行程序的级别。
但是，以下命令仅在已返回原点时才可执行。

移动命令 MOVE, MOVE2, MOVEI, MOVEI2, DRIVE, DRIVE2
 DRIVEI, DRIVEI2, PMOVE, PMOVE2, PATH START
位置获取命令 WHERE, WHERE2, WHRXY, WHRXY2

级别	内 容									
	未返回原点时的 程序执行		开启电源时				程序启动时的 程序复位		自动模式中的 返回原点信号输入	
			模式		程序复位					
	可以	不可以	手动	自动	是	否	是	否	有效	无效
0		○	○			○		○		○
1	○		○			○		○		○
2	○		○		○			○		○
3	○			○		○		○		○
4	○			○	○			○		○
5	○		○		○		○			○
6	○			○	○		○			○
7	○			○		○		○	○ *1	
8	○			○	○		○		○ *1	

*1：在自动模式中的绝对式原点复位信号输入（DI17）有效时，在自动模式中通过绝对式原点复位信号输入的处理动作中，机器人程序运行中（DO13）的信号将开启。

参照 关于执行级别的详细说明，请参阅各控制器的手册。

索引

索引

符号

_SELECT 1-5

A

ALL 文件 10-19

B

保留字一览表 14-1

编程示例

 基本篇 11-1

 应用篇 11-8

变量的清除 3-20

变量类型 3-4

变量名 3-3

变量文件 10-23

变为 WAIT 状态的命令 6-4

编译 8-166

标签 8-79

并行端口 8-44, 8-47

并行输出变量 3-10

并行输入变量 3-9

C

参数目录文件 10-22

参数文件 10-8

CASE 8-150

常量文件 10-26

程序

 复制 12-12

 切换 1-5, 8-166

 删除 12-14

 停止 8-67

 选择 1-5

 暂停 8-73

程序速度更改命令 13-12

程序级别 8-155

程序名 1-2

程序名更改 12-16

程序目录文件 10-20

程序切换命令 12-42

程序属性更改 12-16

程序文件 10-2

串行端口 8-158

串行端口通信文件 10-46

串行双倍字输出 3-17

串行双倍字输入 3-16

串行输出变量 3-15

串行输入变量 3-14

串行字输出 3-17

串行字输入 3-16

初始处理 12-17

 存储器区域 12-17

 通信端口 12-17

次序程序 1-4, 7-1

 编程 7-5

 编程方法 7-1

 编译 7-2

 程序容量 7-8

 规格 7-8

 获取执行状态 12-30

 禁止执行状态 7-3

 扫描时间 7-8

 设置执行标志 12-21

 运算的优先级 7-8

 执行 7-4

 逐步执行 7-4

次序功能 7-1

COMMON 1-6

CONT 指定 8-92, 8-103

错误处理 8-113, 8-144

错误代码编号 8-113

错误发生行编号 8-113

错误恢复处理 8-144

错误提示履历文件 10-43

D

等待程序的执行 8-43

DI / DO 条件式 4-6

点动命令 13-8

定时器输出变量 3-13

定义用户可使用的函数 8-41

定义坐标点 8-137

DI 文件 10-29

动态变量 3-19

动态数组的有效范围 3-19

动作速度 8-29

DO 文件 10-31

端口输出指定 8-106, 8-127

读出文件 8-151

多任务 5-1, 6-1

多台机器人的控制 5-1

E

EOF 文件 10-45

Ethernet 端口通信文件 10-51

F

返回原点	8-22
返回原点命令	13-10
返回原点顺序	8-117
FUNCTION	1-2
赋值语句	8-83

G

更改手动速度	12-9
更改手系统标志	8-87
各组命令一览表	5-2
公差参数	8-172
功能键录入	1-3
关节坐标类型	4-5
关系运算符	4-1

H

函数：按功能	8-18, 14-18
函数：按字母顺序	8-15, 14-15
获取版本	12-32
获取程序执行状态	12-12
获取次序程序执行状态	12-30
获取存储器剩余容量	12-35
获取当前位置	12-32
获取断点状态	12-25
获取访问级状态	12-24
获取返回原点状态	12-29
获取紧急停止状态	12-36
获取机械臂状态	12-25
获取绝对式原点复位状态	12-29
获取控制器构成	12-26
获取脉冲坐标系的当前位置	12-32
获取模式状态	12-27
获取启动 / 暂停状态任务的处理	12-33
获取任务运行状态	12-34
获取伺服状态	12-30
获取速度状态	12-31
获取提示	12-28
获取位移状态	12-34
获取显示语言	12-24
获取选配插槽状态	12-37
获取 XY 坐标系的当前位置	12-33
获取执行级别状态	12-26
获取坐标点单位坐标系	12-31

I

IF	8-74
区块 IF 语句	8-75
简单 IF 语句	8-74

IO 寄存器	13-3
IO 命令	
格式	13-1
收发	13-3
详细说明	13-6
一览表	13-5
执行触发输入	13-1, 13-3

J

减速比率	8-38
减速度指定	8-104
加速度系数	8-23
加速度指定	8-104
解除内部紧急停止标志	12-21
静态变量	3-19
经自我诊断获取出错状态	12-36
机器参照文件	10-44
机器人语言一览表：按功能	14-8
机器人语言一览表：按字母顺序	14-3
就绪队列	6-3
机械臂锁定输出	8-86
机械臂锁定输出变量	3-12, 7-6
机械臂指定更改命令	13-13
机械手	
定义	8-68
定义文件	10-13
获取机械手状态	12-35
切换	8-32
指定更改命令	13-13
局部变量	3-19, 8-163
绝对式原点复位	12-9
绝对式原点复位命令	13-10
绝对位置移动	8-92

K

控制代码	12-44
控制伺服状态	8-153

L

LABEL 语句	8-79
类型转换	3-6, 4-3
联机命令	
获取数值表达式的值	12-38
获取位移表达式的值	12-39
获取字符串表达式的值	12-38
获取坐标点表达式的值	12-39
联机命令一览表	12-1
逻辑运算符	4-2
LO 文件	10-35

M	
马达电源	8-153
码垛	11-4, 11-10
命令语句格式	1-7
模式更改	12-6
MOVEI 移动命令	13-7
MOVE 移动命令	13-6
MO 文件	10-33

N	
内部输出	8-91
内部输出变量	3-11
扭矩偏移值	8-51
扭矩限制值	8-51
扭矩限制指定	8-51, 8-176
扭矩指令值	8-173

O	
OUT 有效位置	8-120

P	
PATH	8-122, 9-1
结束路径设置	8-128
开始路径设置	8-129
开始移动	8-131
设置移动路径	8-122
使用方法	9-1
特点	9-1
注意事项	9-2

Q	
前端重量	8-181
切换执行任务	12-8
全局变量	3-19, 10-23

R	
任务	
编号	8-160
程序示例	6-8
重新启动	8-143
等待条件	6-4
定义	6-1
恢复启动	6-5
结束	8-63
排程	6-3
强制结束	8-36
启动	6-2, 8-160

删除	6-6
数据的共享	6-8
停止	6-7
优先级	6-1, 8-33, 8-160
暂停	8-165
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	11-18, 11-19

S	
SEQUENCE	1-4, 7-1
删除坐标点	12-14
删除坐标点注释	12-15
设置断点	12-8
设置访问级	12-20
设置执行级别	12-20
拾放	11-12
实数型常量	2-1
使用位移坐标	11-3
使用坐标点编号	11-2
手持编程器	12-20
手动模式操作	12-9
手动模式时的坐标系单位设置	12-19
手动移动速度更改命令	13-12
手系统设置	12-21
手系统标志	3-8, 4-5, 8-99, 8-109, 8-137, 10-4, 10-15
水平多关节型机器人的手系统设置	12-21
数据处理	12-24
数据的读出处理	12-40
数据文件	10-1
种类	10-1
数据写入处理	12-41
数值型常量	2-1
数组	3-5
数组变量的声明	8-46
数组变量文件	10-27
数组的角标	8-46
伺服	
FREE	8-153
OFF	8-153
ON	8-153
伺服命令	13-11
SI 文件	10-39
SO 文件	10-41
SOW 文件	10-49

STOPON 条件指定	8-51, 8-57, 8-136
算术赋值语句	8-83
算术运算	4-1
算术运算符	4-1

T

TO 端口	8-171, 8-178
通信端口	8-111, 8-116
TO 文件	10-37
涂胶	11-17
托盘	
定义	8-132
定义编号	8-132, 8-133
定义文件	10-15
删除	12-15
位置编号	8-133
移动	8-133
移动命令	13-7

U

UTILITY 操作	12-12
UTILITY 模式设置	12-20

W

外部程序	8-40
微动命令	13-8
位移变量	3-8
位移赋值语句	8-85
位移元素变量	3-9
位移指定更改命令	13-13
位移坐标	8-149, 8-156
定义文件	10-11

X

相对位置移动	8-107
显示错误提示的初始化	12-20
显示语言设置	12-18
写入文件	8-151
形参	8-163
系统变量	3-2, 3-7
系统生成	5-1
XY 指定	8-53

Y

移动速度	8-159
以轴为单位进行绝对移动	8-48
以轴为单位进行相对移动	8-54
用户变量	3-2

用于绝对式原点复位的移动命令	13-9
右手系统	8-147
圆弧插补	8-96, 8-122
圆弧插补移动指定	8-101, 8-135
圆弧位置	8-24
运算的优先级	4-3

Z

增量式马达	8-118
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	3-18
直线插补	8-94, 8-122
执行级别	1-5, 14-20
执行中断处理	12-44
轴前端重量	8-30
轴指定 PTP	8-48
主任务	8-33
注释	1-7, 8-141
主组	5-1
自动模式操作	12-7
自动移动速度	8-29
自动移动速度更改命令	13-12
字符串	
比较	4-4
联结	4-4, 8-84
运算	4-4
字符串赋值语句	8-84
字符型常量	2-2
子过程	8-31, 8-155, 8-163
子例程	8-65, 8-114
坐标点变量	3-7
坐标点的复制	12-13
坐标点赋值语句	8-84
坐标点示教命令	13-9
坐标点数据	
类型	4-5
以轴为单位	8-87
坐标点文件	10-4
坐标点显示单位指定命令	13-13
坐标点元素变量	3-8
坐标点注释的复制	12-13
坐标点注释文件	10-6
坐标平面指定	8-105, 8-126
左手系统	8-81

修订记录

修订日期	修订内容
2013年 3月	4.01 版 初版

程序设计说明书

RCX Series

2013年3月

Version 4.01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 机器人商务部

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/home/350.html>

