

FACTORY AUTOMATION EQUIPMENT
YAMAHA 4轴/2轴机器人控制器

RCX 3 Series

编程手册

RCX340/RCX341/RCX320

前言

安全注意事项

第 1 章 编程的记述方法

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

第 2 章 常量

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

第 3 章 变量

1	概要	3-1
2	用户变量与系统变量	3-2
2.1	用户变量	3-2
2.2	系统变量	3-2
3	变量名	3-3
3.1	动态变量名称	3-3
3.2	静态变量名称	3-3

4	变量类型	3-4
4.1	数值型	3-4
4.2	字符型	3-4
5	数组	3-5
6	赋值	3-5
7	类型转换	3-6
8	值传递与引用传递	3-6
9	系统变量	3-7
9.1	坐标点变量	3-7
9.2	位移变量	3-8
9.3	并行输入变量	3-8
9.4	并行输出变量	3-9
9.5	内部输出变量	3-10
9.6	机械臂锁定输出变量	3-11
9.7	定时器输出变量	3-12
9.8	串行输入变量	3-13
9.9	串行输出变量	3-14
9.10	串行字输入	3-15
9.11	串行双倍字输入	3-15
9.12	串行字输出	3-16
9.13	串行双倍字输出	3-16
10	指定端口 / 比特	3-17
11	变量的有效范围	3-19
11.1	动态变量 · 数组的有效范围	3-19
11.2	静态变量的有效范围	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	机器人指定命令一览表	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	启动任务	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
6	注意事项	6-9

第 7 章 逻辑控制功能

1	关于逻辑控制功能	7-1
2	逻辑控制程序规格	7-1
3	创建逻辑控制程序	7-2
3.1	程序方法	7-2
3.2	编译	7-4
4	执行逻辑控制程序	7-5
4.1	逻辑控制程序的逐步执行	7-5
5	编写逻辑控制程序	7-6
5.1	赋值语句	7-6
5.2	输入输出变量	7-6
5.3	定时定义语句	7-8
5.4	运算符（逻辑运算符）	7-9
5.5	运算的优先级	7-9

第 8 章 机器人语言

机器人语言一览表	8-1
ABS	计算绝对值 8-6
ABSRPOS	获取机器参照 8-7
ACCEL	设置 / 获取加速度系数参数 8-8
ARCHP1 / ARCHP2	设置 / 获取参数的拱形距离 8-9
ARMCND	获取机械臂状态 8-10
ARMSEL	设定 / 获取当前的手系统选择 8-11
ARMTYP	设定 / 获取程序复位时的手系统 8-12
ASPEED	设定 / 获取机器人的自动移动速度 8-13
ATN / ATN2	计算反正切值 8-14
AXWGHT	设置 / 获取轴前端重量 8-15
BRKDIAG	获取指定轴的制动器故障诊断结果 8-16
CALL	调用子过程 8-17
CHANGE	切换机械手 8-18
CHGPRI	更改任务的优先级 8-19
CHGWRK	切换工件数据 8-20
CHR\$	计算带有指定字符编码的字符 8-21

CLOSE	关闭指定的通用以太网端口	8-22
CONTPLS	设定 / 获取指定机器人的 CONT 脉冲参数	8-23
CORRECTXY	获取修正工件把持位置偏移后的目标位置	8-24
COS	计算余弦值	8-25
CREWRK	创建以两点间差分为偏移的工件	8-26
CURTQST	获取相对于轴的定格扭矩值（定格电流值） 当前扭矩值（当前电流值）的比率	8-27
CURTRQ	获取相对于轴的最大扭矩值（最大电流值） 当前扭矩值（当前电流值）的比率	8-28
CUT	强制结束启动中的其他任务	8-29
DATE\$	获取日期	8-30
DECEL	设置 / 获取减速比率参数	8-31
DEF FN	定义用户可使用的函数	8-32
DEGRAD	角度转换（度→弧度）	8-33
DELAY	在指定时间等待程序的执行	8-34
DI	获取并行端口的输入状态	8-35
DIM	声明数组变量	8-36
DIST	计算 2 个点之间的距离	8-37
DO	向并行端口输出，或获取输出状态	8-38
DRIVE	以轴为单位进行绝对移动	8-40
DRIVEI	以轴为单位进行相对移动	8-43
END SELECT	结束 SELECT CASE 语句	8-48
END SUB	结束子过程的定义	8-49
ERR / ERL	获取错误编码 / 错误发生行编号	8-50
ETHSTS	获取以太网端口的状态	8-51
EXIT FOR	强制结束 FOR ~ NEXT 语句的循环	8-52
EXIT SUB	强制结束使用 SUB ~ END 定义的子过程	8-53
EXIT TASK	结束当前正在执行的任务	8-54
FOR ~ NEXT	变量超过指定值为止将循环	8-55
GEPSTS	获取通用以太网端口的状态	8-56
GOSUB ~ RETURN	跳转至子例程	8-57
GOTO	无条件跳转至指定的行	8-58
HALT	停止程序并进行复位	8-59
HALTALL	停止并复位所有的程序	8-60
HAND	机械手定义	8-61
48.1 水平多关节型机器人时		8-62
48.2 正交机器人时		8-64

HOLD	暂停程序	8-66
HOLDALL	暂停所有的程序	8-67
IF	评估条件表达式的值，按照条件执行命令语句。	8-68
51.1 简单 IF 语句		8-68
51.2 区块 IF 语句		8-69
INPUT	从手持编程器中对变量赋值	8-70
INROFST	设置 / 获取水平多关节 R 轴惯性力矩偏移量参数	8-71
INT	舍去小数点以下部分	8-72
JTOXY	轴单位制转换（脉冲→毫米）	8-73
LEFT\$	从左端抽出字符串	8-74
LEFTY	将水平多关节机器人的手系统设置为左手系统	8-75
LEN	获取字符串的长度	8-76
LET	对变量赋值	8-77
LO	输出机械臂锁定，或获取输出状态	8-80
LOCx	设定 / 获取坐标点数据为轴单位或者位移数据和工件数据为要素单位	8-82
LSHIFT	比特的左位移	8-84
MCHREF	获取机器参照（传感方式或撞块方式的轴）	8-85
MID\$	从指定位置获取字符串	8-86
MO	将指定数值输出到 MO 端口，或获取输出状态	8-87
MOTOR	控制马达电源状态	8-89
MOVE	进行绝对位置移动	8-90
MOVEI	进行相对位置移动	8-107
MOVET	对机器人的所有轴进行工具坐标相对位置移动	8-108
MTRDUTY	获取指定轴的马达负载率	8-109
OFFLINE	将通信端口设置为脱机模式	8-110
ON ERROR GOTO	在发生错误时跳转至指定标签	8-111
ON ~ GOSUB	执行由表达式的值指定的子例程	8-112
ON ~ GOTO	跳转至表达式的值所指定的标签	8-113
ONLINE	将通信端口设置为联机模式	8-114
OPEN	打开指定的通用以太网端口	8-115
ORD	计算字符编码	8-116
ORGORD	设置 / 获取机器人的返回原点顺序	8-117
ORIGIN	原点归复	8-118
OUT	开启指定端口的输出	8-119
OUTPOS	设置 / 获取机器人的 OUT 有效位置参数	8-120
PATH	设置 PATH 移动路径	8-121

PATH END	结束 PATH 的路径设置	8-125
PATH SET	开始 PATH 的路径设置	8-126
PATH START	开始 PATH 移动	8-128
PDEF	定义托盘	8-130
PGMTSK	获取已录入指定程序的任务编号	8-131
PGN	从指定的程序名中获取程序编号	8-132
PMOVE	执行机器人的托盘移动	8-133
Pn	定义坐标点	8-136
PPNT	创建托盘坐标点数据	8-138
PRINT	在手持编程器上显示指定表达式的值	8-139
PSHFRC	设定 / 获取推压力参数	8-140
PSHJGSP	设定 / 获取推压判定速度阈值参数	8-141
PSHMTD	设定 / 获取推压方式参数	8-142
PSHRSLT	获取 PUSH 语句结束时的状态	8-143
PSHSPD	设定 / 获取推压移动速度参数	8-144
PSHTIME	设定 / 获取推压时间参数	8-145
PUSH	以轴为单位进行推压动作	8-146
RADDEG	单位转换（弧度→度）	8-151
REFTSK	获取任务状态	8-152
REM	插入注释	8-153
RESET	关闭指定端口的比特或清除变量	8-154
RESTART	重新启动暂停中的其他任务	8-155
RESUME	错误恢复处理后恢复执行程序	8-156
RETURN	使 GOSUB 中的分支处理返回至 GOSUB 的下一行	8-157
RIGHT\$	从右端抽出字符串	8-158
RIGHTY	将水平多关节机器人的手系统设置为右手系统	8-159
RSHIFT	比特的右位移	8-160
SCRINR	设置 / 获取水平多关节 R 轴惯性力矩参数	8-161
SELECT CASE ~ END SELECT	执行由表达式的值指定的命令区	8-162
SEND	将读出文件的数据转发至写入文件	8-163
SERVO	控制伺服状态	8-165
SET	开启指定端口的比特	8-166
SETGEP	设置通用以太网端口	8-167
SGI	获取指定的整数型静态变量，或将值赋给该变量	8-168
SGR	获取指定的实数型静态变量，或将值赋给该变量	8-169

SHARED	使不提交变量便可在子过程中参照	8-170
SHIFT	设置位移坐标	8-171
SI	获取指定 SI 的状态	8-172
SID	获取指定串行输入的文字信息（双字）	8-173
SIN	计算正弦值	8-174
SIW	获取指定串行输入的文字信息	8-175
Sn	定义位移坐标	8-176
SO	输出串行端口上指定的值，或获取输出状态	8-177
SOD	输出指定串行输出的文字信息（双字），或获取输出状态	8-178
SOW	输出指定串行输出的文字信息，或获取输出状态	8-179
SPEED	更改程序移动速度	8-180
SQR	计算平方根	8-181
START	启动新任务	8-182
STR\$	将数值转换为字符串	8-183
SUB ~ END SUB	定义子过程	8-184
SUSPEND	暂停执行中的其他任务	8-186
SWI	切换执行程序	8-187
TAN	计算正切值	8-188
TCOUNTER	定时器、计数器	8-189
TIME\$	获取当前时间	8-190
TIMER	获取当前时间总秒数	8-191
TO	将指定值输出至 TO 端口，或获取输出状态	8-192
TOLE	设置 / 获取公差参数	8-193
TORQUE	设置 / 获取最大扭矩指令值	8-194
TSKPGM	获取录入指定任务编号的程序编号	8-196
VAL	将字符串转换为数值	8-197
WAIT	等待条件表达式成立	8-198
WAIT ARM	等待机器人轴动作结束	8-199
WEIGHT	设置 / 获取前端质量 (kg) 参数	8-200
WEIGHTG	设置 / 获取前端质量 (g) 参数	8-201
WEND	结束 WHILE 语句的命令区	8-202
WHERE	获取机械臂当前位置（脉冲坐标）	8-203
WHILE ~ WEND	在条件成立时反复动作	8-204
WHRXY	通过正交坐标获取机械臂当前位置	8-205
WRKDEF	工件定义（作成指定编号的工件）	8-206
XYTOJ	将正交坐标数据（毫米）转换为关节坐标数据（脉冲）	8-208

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	10-2
2 程序文件	10-3
2.1 整个程序	10-3
2.2 程序单位	10-4
3 坐标点文件	10-5
3.1 所有坐标点	10-5
3.2 坐标点单位	10-7
4 坐标点注释文件	10-8
4.1 所有坐标点注释	10-8
4.2 坐标点注释单位	10-9
5 坐标点名称文件	10-10
5.1 所有坐标点名称	10-10
5.2 坐标点名称单位	10-11
6 参数文件	10-12
6.1 所有参数	10-12
6.2 参数单位	10-14
7 位移坐标定义文件	10-16
7.1 所有位移	10-16
7.2 位移单位	10-17
8 机械手定义文件	10-18
8.1 所有机械手	10-18
8.2 机械手单位	10-19

9	工件定义文件	10-20
9.1	所有工件	10-20
9.2	工件单位	10-21
10	托盘定义文件	10-22
10.1	所有托盘定义	10-22
10.2	托盘定义单位	10-24
11	通用以太网端口定义文件	10-26
12	输入输出名称文件	10-28
12.1	所有输入输出名称	10-28
12.2	输入输出类型单位	10-29
12.3	输入输出端口单位	10-30
12.4	输入输出比特单位	10-31
13	区域判定输出定义文件	10-32
13.1	所有区域判定输出定义	10-32
13.2	区域判定输出定义单位	10-33
14	实时输出文件	10-34
14.1	所有实时输出设置	10-34
14.2	实时输出设置单位	10-35
14.3	实时输出设置一览	10-36
15	ALL 文件	10-38
15.1	所有 ALL 文件	10-38
16	程序目录文件	10-40
16.1	所有程序目录	10-40
16.2	程序单位	10-41
17	参数目录文件	10-42
17.1	所有参数目录	10-42
18	机器参照文件	10-43
18.1	机器参照（传感器轴、撞块轴）	10-43
18.2	机器参照（标记轴）	10-44
19	累计数据文件	10-45

20	系统构成信息文件	10-46
21	版本信息文件	10-47
22	选件板文件	10-48
23	自我诊断文件	10-49
24	报警履历文件	10-50
25	存储器剩余容量文件	10-52
26	变量文件	10-53
26.1	动态变量	10-53
26.1.1	所有动态变量	10-53
26.1.2	动态变量单位	10-54
26.2	静态变量	10-55
26.2.1	整数型静态变量 (SGI)	10-55
26.2.2	实数型静态变量 (SGR)	10-57
27	常量文件	10-59
27.1	单个字符串	10-59
28	数组变量文件	10-60
28.1	所有数组变量	10-60
28.2	数组变量单位	10-61
29	DI 文件	10-62
29.1	所有 DI	10-62
29.2	DI 端口单位	10-63
30	DO 文件	10-64
30.1	所有 DO	10-64
30.2	DO 端口单位	10-65
31	MO 文件	10-66
31.1	所有 MO	10-66
31.2	MO 端口单位	10-67
32	LO 文件	10-68
32.1	所有 LO	10-68
32.2	LO 端口单位	10-69

33 TO 文件	10-70
33.1 所有 TO	10-70
33.2 TO 端口单位	10-71
34 SI 文件	10-72
34.1 所有 SI	10-72
34.2 SI 端口单位	10-73
35 SO 文件	10-74
35.1 所有 SO	10-74
35.2 SO 端口单位	10-75
36 SIW 文件	10-76
36.1 所有 SIW	10-76
36.2 SIW 单位	10-77
37 SOW 文件	10-78
37.1 所有 SOW	10-78
37.2 SOW 单位	10-79
38 EOF 文件	10-80
39 串行端口通信文件	10-81
40 Ethernet 端口通信文件	10-82

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
	关于终端编码	12-1
1.1	键盘操作	12-2
1.2	实用	12-3
1.3	数据处理	12-4
1.4	单独执行机器人语言	12-5
1.5	控制代码	12-5
2	操作、设定命令	12-6
2.1	程序操作	12-6
2.1.1	任务登录	12-6
2.1.2	程序复位	12-7
2.1.3	程序执行	12-8
2.1.4	程序停止	12-9
2.1.5	执行 1 行程序	12-10
2.1.6	执行程序至下一行	12-11
2.1.7	执行程序至指定行之前	12-11
2.1.8	跳过 1 行程序	12-12
2.1.9	跳过程序至指定行之前	12-12
2.1.10	断点设定	12-13
2.1.11	程序语法检查 程序	12-14
2.1.12	主程序设置	12-15
2.1.13	编译逻辑控制程序	12-15
2.2	手动模式操作	12-16
2.2.1	更改手动速度	12-16
2.2.2	坐标点示教	12-16
2.2.3	更改微动距离	12-17
2.3	清除警报	12-17
2.4	清除输出信息缓冲区	12-18
2.5	设定输入数据	12-19
2.6	更改访问级	12-20
2.7	更改密码	12-21

3	参照命令	12-22
3.1	获取返回原点状态	12-22
3.2	获取伺服状态	12-23
3.3	获取马达电源状态	12-23
3.4	获取访问级状态	12-24
3.5	获取断点状态	12-24
3.6	获取模式状态	12-25
3.7	获取通信端口状态	12-25
3.8	获取主程序编号	12-26
3.9	获取次序程序执行状态	12-26
3.10	获取逻辑控制编译状态	12-27
3.11	获取版本	12-27
3.12	获取位移状态	12-28
3.13	获取启动 / 暂停状态任务的处理	12-28
3.14	获取任务运行状态	12-29
3.15	获取任务结束条件	12-29
3.16	获取机械手数据	12-30
3.17	获取机械手状态	12-30
3.18	获取工件数据	12-31
3.19	获取工件状态	12-31
3.20	获取存储器剩余容量	12-32
3.21	获取警报状态	12-32
3.22	获取紧急停止状态	12-33
3.23	获取手动移动速度	12-33
3.24	获取微动量	12-34
3.25	获取最终引用点位编号	12-34
3.26	获取输出信息	12-35
3.27	获取输入数据	12-35
3.28	获取制动器故障诊断结果	12-36
3.29	获取值	12-37
3.29.1	获取数值表达式的值	12-37
3.29.2	获取字符串表达式的值	12-37
3.29.3	获取坐标点表达式的值	12-38
3.29.4	获取位移表达式的值	12-38
4	动作命令	12-39
4.1	绝对复位	12-39

4.2	返回原点动作	12-39
4.3	手动移动：点动	12-40
4.4	手动移动：微动	12-41
5	数据文件操作命令	12-42
5.1	复制操作	12-42
5.1.1	程序的复制	12-42
5.1.2	坐标点的复制	12-42
5.1.3	坐标点注释的复制	12-43
5.2	删除操作	12-43
5.2.1	程序删除	12-43
5.2.2	删除坐标点	12-44
5.2.3	删除坐标点注释	12-44
5.2.4	删除坐标点名称	12-45
5.2.5	删除托盘	12-45
5.2.6	删除通用以太网端口	12-46
5.2.7	删除实时输出	12-46
5.3	程序名更改	12-47
5.4	程序属性更改	12-47
5.5	初始化处理	12-48
5.5.1	存储	12-48
5.5.2	通信端口的初始处理	12-49
5.5.3	警报记录的初始化处理	12-49
5.6	数据的读出处理	12-50
5.7	数据写入处理	12-51
6	实用命令	12-52
6.1	设置次序程序执行标志	12-52
6.2	日期设定	12-52
6.3	时刻设定	12-53
7	机器人语言的单独执行	12-54
7.1	命令	12-54
7.2	函数	12-55
8	控制代码	12-56

第 13 章 附录

1 保留字一览表	13-1
2 变化点（与以往机型相比）	13-3
1 程序名称	13-3
2 多台机器人控制	13-3
3 多任务	13-4
4 机器人语言	13-4
5 联机命令	13-4
6 数据文件	13-4

前言

非常感谢您购买 YAMAHA 机器人控制器 RCX340。

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

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

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

适用机型：RCX340, RCX341, RCX320

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”。



注意

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。



补充

主要记载由于版本不同而其功能的不同。



MEMO

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。

第 1 章

编程的记述方法

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

YAMAHA 机器人的编程使用 YAMAHA 开发的独创机器人语言。该语言是类似 BASIC (Beginner's All-purpose Symbolic Instruction Code) 语言的简单编程语言，任何人均可容易地学会，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。

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

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

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

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

并且使用多任务程序还可同时执行多个命令语句进行并行处理。

使用上述功能后，可简易地创建复杂的处理程序。

参照

- 关于子过程，请参阅第 8 章“11 CALL”及“127 SUB ~ END SUB”。
- 关于用户定义函数，请参阅第 8 章“24 DEF FN”。

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

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

以下 2 个程序名已事先预约，带有这些名称的程序有着特殊含义。

- A) SEQUENCE
- B) COMMON

各程序的功能如下所示。

A) SEQUENCE

功能 RCX340/RCX341/RCX320 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“2. 逻辑控制程序规格”。) 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“2. 逻辑控制程序规格”。

B) COMMON

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

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

SAMPLE	说明
<pre>X!=2.5 Y!=1.2 CALL *DISTANCE(X!,Y!,REF C!) GOSUB *AREA PRINT C!,Z! HALT</pre>	程序名：SAMPLE1
<pre>X!=5.5 Y!=0.2 CALL *DISTANCE(X!,Y!,REF C!) GOSUB *AREA PRINT C!,Z! HALT</pre>	程序名：SAMPLE2
<pre>SUB *DISTANCE(A!,B!,C!) C!=SQR(A!^2+B!^2) END SUB *AREA: Z!=X!*Y! RETURN</pre>	程序名：COMMON 通用例程

参照 详细说明，请参阅本书中的各命令。

5 标识符

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

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

SAMPLE

```
LOOP, SUBROUTINE, GET_DATA
```

参照 关于保留字，请参阅第 13 章“1 保留字一览表”。
关于系统变量，请参阅第 3 章“9 系统变量”。

6 LABEL 语句

在程序行中定义标签。

格式

* 标签：

标签必须一贯以“*”起始。此外，必须从行的起始开始记述。
进行标签定义时，必须在末尾加上“:”，但是在程序中指定跳转目标时则不需要。

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

SAMPLE

说明

```
*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
```

7 注释

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	说明
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

8 命令语句格式

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。
- *斜体* 中的项目采用规定格式记述。
- | 中的项目记述了任意一个所记述的项目。
- 标签可以省略。声明标签时，必须以星号 (*) 开始，必须以冒号 (:) 结束 (作为分支记述标签时，不需要冒号)。

关于标签请参阅本章“6 LABEL 语句”。

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

1 行中可记述的最大字符数为 255 个字符。

第 2 章

常量

1	概要	2-1
2	数值型常量	2-1
3	字符型常量	2-2

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

分类	类型	详细内容 / 范围
数值型	整数型	十进制常量 -1,073,741,824 ~ 1,073,741,823
		二进制常量 &B0 ~ &B11111111
		八进制常量 &O20000000000 ~ &O17777777777
		十六进制常量 &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	255 个字节以内的英文字母、数字、特殊字符、假名字符

2.1 整数型常量

1. 十进制常量
从 -1,073,741,824 至 1,073,741,823 的整数。
2. 二进制常量
8 比特以下的不带符号二进制数字。在最前面加上表示二进制的 &B。
范围：&B0 (十进制：0) ~ &B11111111 (十进制：255)
3. 八进制常量
32 比特以下的带有符号八进制数字。在最前面加上表示八进制的 &O。
范围：&O20000000000 (十进制：-2,147,483,648) ~ &O17777777777 (十进制：2,147,483,647)
4. 十六进制常量
32 比特以下的带有符号十六进制数字。在最前面加上表示十六进制的 &H。
范围：&H80000000 (十进制：-2,147,483,648) ~ &H7FFFFFFF (十进制：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

SAMPLE

```
-1. 23456E-12
3. 14E0
1. E5
```



MEMO

如果用带符号的十六进制表达 -2,147,483,648 ~ 2,147,483,648，则整数型常量的范围为 &H80000000 ~ &H7FFFFFFF。

3 字符型常量

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

作为字符串使用双引号 (") 时，接在 (") 后面记述。

SAMPLE	说明
"YAMAHA ROBOT"	
"EXAMPLE OF" "A" " "	变为 EXAMPLE OF "A"
PRINT "COMPLETED"	

第 3 章

变量

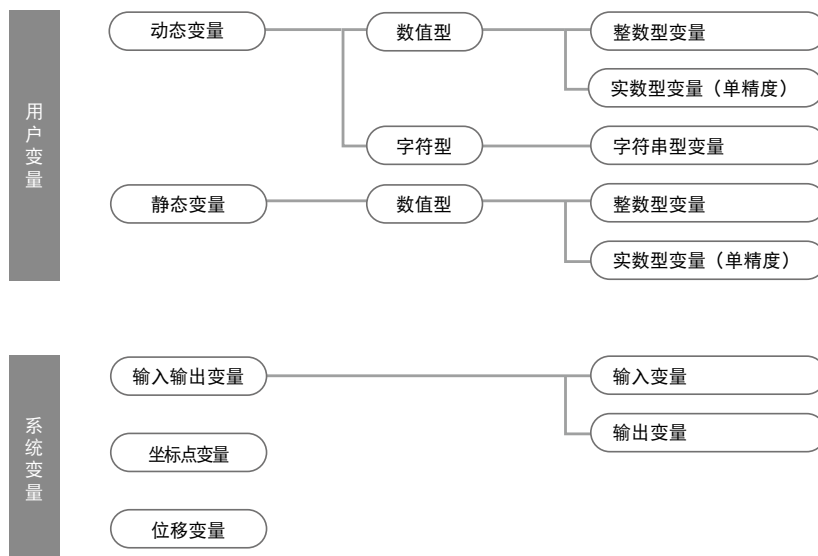
1	概要	3-1
2	用户变量与系统变量.....	3-2
3	变量名	3-3
4	变量类型.....	3-4
5	数组	3-5
6	赋值	3-5
7	类型转换.....	3-6
8	值传递与引用传递	3-6
9	系统变量.....	3-7
10	指定端口 / 比特	3-17
11	变量的有效范围	3-19
12	变量的清除.....	3-20

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

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

用户变量与系统变量



35301-R9-00

参照 关于数组的详细内容，请参阅本章“5 数组”。

2 用户变量与系统变量

2.1 用户变量

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

分类	变量类型	详细内容、值的范围
动态变量	数值型	整数型变量 -2,147,483,648 ~ 2,147,483,647 (带符号的十六进制：&H80000000 ~ &H7FFFFFFF) 实数型变量（单精度） $-1.0 \times 10^{38} \sim +1.0 \times 10^{38}$
	字符型	字符串型变量 255 个字节以内的英文字母、数字、特殊字符、假名字符
静态变量	数值型	整数型变量 -2,147,483,648 ~ 2,147,483,647 实数型变量（单精度） $-1.0 \times 10^{38} \sim +1.0 \times 10^{38}$
	字符型	字符串型变量 255 个字节以内的英文字母、数字、特殊字符、假名字符
数组变量	数值型	整数型数组 -2,147,483,648 ~ 2,147,483,647 实数型数组（单精度） $-1.0 \times 10^{38} \sim +1.0 \times 10^{38}$
	字符型	字符串型数组 255 个字节以内的英文字母、数字、特殊字符、假名字符



补充
数组变量是动态变量。

2.2 系统变量

系统变量如下所示预先确定了名称，无法进行更改。

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

参照 关于详细内容请参阅本章“9 系统变量”。

3.1 动态变量名称

遵照以下规格可任意确定动态变量的变量名。

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

SAMPLE	说明
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, DOn, MOn, LOn, TOn, SIn, SOn, Pn, Sn, Hn (n 为数值)

SAMPLE	说明
COUNT	○ 可以使用
ABS	× (保留字)
FNAME	× (FN : 用户定义变量)
S91	× (Sn : 已定义变量)

参照 关于保留字, 请参阅第 13 章“1 保留字一览表”。

3.2 静态变量名称

静态变量的变量名如下规定, 不可使用其他名称。

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

只要不通过系统模式或联机命令进行初始处理, 静态变量的内容将不会被清除。

参照 关于静态变量的清除, 请参阅本章“12 变量的清除”。

4 变量类型

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。
 - ASP_DEF!.....实数型变量
 - ASP_DEF.....实数型变量
- 均被作为相同变量处理。

4.1 数值型

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 -2, 147, 483, 648 ~ 2, 147, 483, 647 (用带符号的十六进制表示为 &H80000000 ~ &H7FFFFFFF)。

SAMPLE
R 1 % = 1 0
R 2 % (2) = R 1 % + 1 0 0 0 0



补充

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”。

实数型

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

SAMPLE
R 1 ! = 1 0 . 3 1
R 2 ! (2) = R 1 % + 1 . 9 8 E 3



补充

实数型的“!”可以省略。

4.2 字符型

字符型变量、字符串数组的元素可处理的字符串最大为 255 个字符。在字符串中可使用英文字母、数字、特殊字符。

SAMPLE	说明
R 1 \$ = " Y A M A H A "	为 " Y A M A H A M O T O R "
R 2 \$ (2) = R 1 \$ + " M O T O R "	

5 数组

动态变量可使用数值型、字符型中的任意一种数组。

使用数组后，可统一管理连续而相同类型的多个数据。

有关数组的各元素，请参照每个变量名后面的括号中加注的角标。角标可使用整数或<表达式>，最多可使用3维。若要使用数组，必须预先通过DIM语句声明，可使用的元素数最大为已声明的角标+1为止(0~角标)。

MEMO

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

格式		
变量名	% ! \$	(表达式, 表达式, 表达式)
SAMPLE		说明
A % (1)		整 数 型 数 组 变 量
DATA ! (1 , 1 0 , 3)		单 精 度 实 数 型 数 组 变 量 (3 维 数 组)
STRING \$ (1 0)		字 符 型 数 组 变 量

6 赋值

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“54 LET(赋值语句)”。

赋给与变量不同类型的值时，数据类型将如下转换。

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

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

值传递

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

引用传递

在子过程中向变量传递参照（存储器中的地址）的方法。如果在子过程内更改了值，**源变量的内容也将相应更改**。
在 CALL 语句实参中指定全部数组（数组名后面带有（））或者在实参前面加上 REF 时，将进行引用传递。

值传递与引用传递

值传递

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

执行结果 X% 的值仍然为“5”

引用传递

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

执行结果 X% 的值变为“50”

以下变量已预先定义。无法使用以下变量名起始的名称。

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

9.1 坐标点变量

通过常量或表达式指定坐标点编号

格式		
Pnnnnn 或 P[表达式]		
标注	值	范围
n	坐标点编号	0 ~ 9

功能

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

SAMPLE
P 0 P 1 1 0 P [A] P [START _ POINT] P [A (1 0)]

9.2 位移变量

通过常量或表达式指定位移编号

格式		
Snn 或 S[表达式]		
标注	值	范围
n	位移编号	0 ~ 9

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

SAMPLE
S 1
S [A]
S [B A S E]
S [A (1 0)]

 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。

9.3 并行输入变量

表示并行输入信号的状态

格式 1		
DI m(b, . . . ,b)		
格式 2		
DI(mb, . . . ,mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7

在格式 1 中省略了 b, . . . , b 时，8 比特均将成为对象。

SAMPLE	说明
A % = D I 1 ()	将从 D I (1 7) 至 D I (1 0) 的输入状态赋值给变量 A % A % 将被赋予 0 ~ 2 5 5 的整数
A % = D I 5 (7 , 4 , 0)	将 D I (5 7) , D I (5 4) , D I (5 0) 的输入状态赋值给变量 A % (上述信号均变为 1 (ON) 时 , A % = 7)
A % = D I (2 7 , 1 5 , 1 0)	将 D I (2 7) , D I (1 5) , D I (1 0) 的输入状态赋值给变量 A % (上述信号除 D I (1 0) 以外均变为 1 (ON) 时 , A % = 6)
W A I T D I (2 1) = 1	等待 D I (2 1) 变为 1 (ON)

 MEMO

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

9.4 并行输出变量

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)

标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7

在格式 1 中省略了 b, . . . , b 时，8 比特均将成为对象。

SAMPLE	说明
A % = DO 2 ()	将从 DO (27) 至 DO (20) 的输出状态赋值给变量 A %
A % = DO 5 (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)
DO 3 () = 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")
DO 4 (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。
- 实际不存在串行端口时，将不进行外部输出。

9.5 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)

标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27, 30 ~ 37
b	比特编号	0 ~ 7

在格式 1 中省略了 b, . . . , b 时，8 比特均将成为对象。

功能 对于仅限控制器内部使用的内部输出变量，可指定及参照内部输出变量的状态。
用于与次序程序进行信号交换等用途。
端口 30 ~ 37 是专用的内部输出变量，只可参照（无法更改）。

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

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

SAMPLE	说明
A % = MO 2 ()	将从 MO (2 7) 至 MO (2 0) 的内部输出状态赋值给变量 A %
A % = MO 5 (7 , 4 , 0)	将 MO (5 7) , MO (5 4) , MO (5 0) 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 (3 7 , 2 5 , 2 0)	将 MO (3 7) , MO (2 5) , MO (2 0) 的内部输出状态赋值给变量 A % (例如：上述信号除 MO (2 5) 以外均为 1 (ON) 时，A % = 5)

MEMO

指定多个比特时，请从左边开始降序（大→小）记述。

9.6 机械臂锁定输出变量

对各轴进行禁止移动的设置

格式 1
LOm(b, . . . ,b)
格式 2
LO(mb, . . . ,mb)

标注	值	范围
m	端口编号	0, 1
b	比特编号	0 ~ 7

在格式 1 中省略了 b, . . . , b 时，8 比特均将成为对象。

- 功能
- 可输出至变量或参照变量。

端口 0 从 0 比特开始依次对应 1 ~ 8 轴；端口 1 从 0 比特开始依次对应 9 ~ 16 轴。

比特开启状态下，禁止对应轴移动。

SAMPLE	说明
A % = LO 0 ()	将 LO (07) 至 (00) 的机械臂锁定状态赋值给变量 A %
A % = LO 0 (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 % = LO (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)
LO 1 () = & B 0 0 1 0	LO (11) 变为 1 (ON)、第 10 轴被禁止移动
LO 1 (2 , 0) = 3	LO (12) 与 LO (10) 变为 1 (ON)、 第 11 轴和第 9 轴被禁止移动

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[12.401 机械臂锁紧 ON]的错误。
 - 机械臂锁定从机器人 1 所使用的小编号轴开始依次对应各轴。
例) 机器人 1 为 5 轴、机器人 2 为 4 轴构成的情况下，端口 0 的比特 0 ~ 4 对应机器人 1 的 1 ~ 5 轴的状态，
端口 0 的比特 5 ~ 7 对应机器人 2 的 1 ~ 3 轴的状态，端口 1 的比特 0 进行机器人 2 的第 4 轴的禁止移动设定。

9.7 定时器输出变量

用于次序程序的定时器功能

格式 1
TOm(b, . . . ,b)
格式 2
TO(mb, . . . ,mb)

标注	值	范围
m	端口编号	0, 1
b	比特编号	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。

参照 有关次序程序中的使用示例，请参阅第 7 章“5.2 输入输出变量”的定时器使用示例。

SAMPLE	说明
A % = T O 0 ()	将 T O (0 7) 至 T O (0 0) 的状态赋值给变量 A %
A % = T O 0 (7 , 4 , 0)	将 T O (0 7) , T O (0 4) , T O (0 0) 的状态赋值给变量 A % (上述信号均为 1 (O N) 时 , A % = 7)
A % = T O (0 6 , 0 4 , 0 1)	将 T O (0 6) , T O (0 4) , T O (0 1) 的状态赋值给变量 A % (上述信号除 T O (0 1) 以外均为 1 (O N) 时 , A % = 6)

 **MEMO**
指定多个比特时，请从左边开始降序(大→小)记述。

9.8 串行输入变量

表示串行输入信号的状态

格式 1
SI(m,b, . . . ,b)
格式 2
SI(mb, . . . ,mb)

标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7

在格式 1 中省略了 b, . . . , b 时，8 比特均将成为对象。

SAMPLE	说明
A % = SI 1 ()	将 SI (1 7) 至 SI (1 0) 的输入状态赋值给变量 A %
A % = SI 5 (7 , 4 , 0)	将 SI (5 7) , SI (5 4) , SI (5 0) 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 (2 7 , 1 5 , 1 0)	将 SI (2 7) , SI (1 5) , SI (1 0) 的输入状态赋值给变量 A % (上述信号除 SI (1 0) 以外均为 1 (ON) 时 , A % = 6)
WAIT SI (2 1) = 1	等待 SI (2 1) 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。

9.9 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)

标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7

在格式 1 中省略了 b, . . . , b 时，8 比特均将成为对象。

SAMPLE	说明
A % = S O 2 ()	将 S O (2 7) 至 S O (2 0) 的输出状态赋值给变量 A %
A % = S O 5 (7 , 4 , 0)	将 S O (5 7) , S O (5 4) , S O (5 0) 的输出状态赋值给变量 A % (上述信号均变为 1 (O N) 时, A % = 7)
A % = S O (3 7 , 2 5 , 2 0)	将 S O (3 7) , S O (2 5) , S O (2 0) 的输出状态赋值给变量 A % (上述信号除 S O (2 5) 以外均为 1 (O N) 时, A % = 5)
S O 3 () = B %	使 S O (3 7) 至 S O (3 0) 的输出变化为用 B % 表示的状态 (例如, B % 为 1 2 3 时, 用二进制表示 1 2 3 就变为 " 0 1 1 1 1 0 1 1 "。 S O (3 7) 及 S O (3 2) 变为 " 0 " , 其他比特变为 " 1 ")
S O 4 (5 , 4 , 0) = & B 1 0 1	将 S O (4 5) 及 S O (4 0) 设置为 " 1 " , 将 S O (4 4) 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。

9.10 串行字输入

表示串行输入的字信息状态

格式		
SIW(m)		
标注	值	范围
m	端口编号	2 ~ 15 *
获取范围为 0 (&H0000) ~ 65, 535 (&HFFFF)。		
SAMPLE	说明	
A % = S I W (2)	将 S I W (2) 的输入状态赋值给变量 A %	
A % = S I W (1 5)	将 S I W (1 5) 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。
- * 在 SIOW 扩大有效的情况下，端口编号的范围为 2 ~ 15、24 ~ 127。
详细内容，请参阅别册《用户手册》“与选件板相关的参数”。

9.11 串行双倍字输入

通过双倍字表示串行输入的字信息状态

格式		
SID(m)		
标注	值	范围
m	端口编号	2, 4, 6, 8, 10, 12, 14 *
获取范围为 -2,147,483,648(&H80000000) ~ 2,147,483,647(&H7FFFFFFF)。		
SAMPLE	说明	
A % = S I D (2)	将 S I W (2) , S I W (3) 的输入状态赋值给变量 A %	
A % = S I D (1 4)	将 S I W (1 4) , S I W (1 5) 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。
 - 配置数据时，低位端口编号的数据配置至低位地址。
例如，SIW (2) = &H2345, SIW (3) = &H0001 时，SID (2) = &H00012345。
- * 在 SIOW 扩大有效的情况下，端口编号的范围为 2, 4, 6 ~ 14, 24, 26, 28 ~ 126。
详细内容，请参阅别册《用户手册》“与选件板相关的参数”。

9.12 串行字输出

在串行输出的字信息中显示输出或输出状态

格式		
SOW(m)		
标注	值	范围
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。		
SAMPLE	说明	
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 SIOW 扩大有效的情况下，端口编号的范围为 2 ~ 15、24 ~ 127。
详细内容，请参阅别册《用户手册》“与选件板相关的参数”。

9.13 串行双倍字输出

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)		
标注	值	范围
m	端口编号	2, 4, 6, 8, 10, 12, 14 *
输出范围为 -2,147,483,648(&H80000000) ~ 2,147,483,647(&H7FFFFFFF)。		
SAMPLE	说明	
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)	

MEMO

- 作为带符号的双倍字数据处理。
- 实际不存在串行端口时，将不被输出至外部。
- 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。

* 在 SIOW 扩大有效的情况下，端口编号的范围为 2, 4, 6 ~ 14, 24, 26, 28 ~ 126。
详细内容，请参阅别册《用户手册》“与选件板相关的参数”。

通过输入输出变量指定端口及比特时, 请使用以下任意一种方法。

MEMO

在 m(端口)和 b(比特)指定整数型常量时(请参考下面的记述示例), 不用记述 &O, 但是将作为 8 进制数来处理。

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)

SAMPLE	说明
DO (2 5)	指定端口 2 的 5 比特
DO 2 (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)

SAMPLE	说明
DO 2 (7 , 5 , 3)	指定 DO (2 7) , DO (2 5) , DO (2 3)
DO (2 7 , 2 5 , 2 3)	

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)

SAMPLE	说明
DO (3 7 , 2 5 , 2 0)	指定 DO (3 7) , DO (2 5) , DO (2 0)

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. 同一端口的多个比特”或“3. 不同端口的多个比特”的方法进行记述。

记述示例：DOm()

SAMPLE	说明
DO 2 ()	指定 DO (2 7) ~ DO (2 0) 的所有端口
DO (2 7 , 2 6 , 2 5 , 2 4 , 2 3 , 2 2 , 2 1 , 2 0)	
DO 2 (7 , 6 , 5 , 4 , 3 , 2 , 1 , 0)	

5. 端口 / 比特编号的变量指定

使用变量指定 DIO、SIO、MO、LO、TO 端口编号和比特编号。

格式 1 (并行输出变量的例)		
DOm(b, . . . ,b)		
格式 2 (并行输出变量的例)		
DO(mb, . . . ,mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27、* 整数型变量、整数型静态变量 (SGI)
b	比特编号	0 ~ 7、* 整数型变量、整数型静态变量 (SGI)
* 可以在括号内使用整数型变量、整数型静态变量 (SGI)。		

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

SAMPLE	说明
A % = &O 2 1 B % = &O 2 2 DO (A % , B %) = 3	将 DO (2 1) , DO (2 2) 设置为 “ 1 ”
SGI 2 = 2 SGI 3 = 3 DO 2 (SGI 2 , SGI 3) = 3	将 DO (2 2) , DO (2 3) 设置为 “ 1 ”

参照 &O 表示使用 8 进制常量指定。详细说明，请参阅第 2 章“2.1 整数型常量”八进制常量。

变量分为以下几类。

11.1 动态变量・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，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。

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“111 SHARED”）。

11.2 静态变量的有效范围

静态变量是通过执行程序复位无法清除数据的变量。

在任何程序中均可更改及参照变量的数据。

变量名如下规定，不可用于任意变量名。

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

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。

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

12.2 静态变量的清除

在以下任意情况下，整数型和实数型均将被清零。

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

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。

SAMPLE	结果
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”

SAMPLE	结果
A=10 > 5	1 0 > 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。

SAMPLE	说明
A=2 B=SQR (A!) IF A!=B!*B! THEN...	此时, A! 不等于 B! * B!。

1.3 逻辑运算符

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

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

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

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

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

示例：A%=3 AND 10 → 对 A% 赋予“2”

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

示例：A%=3 OR 10 → 对 A% 赋予“11”

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

示例：A%=3 XOR 10 → 对 A% 赋予“9”

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

1.4 运算的优先级

按照以下优先级进行运算。当优先级相同时，将从写在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)
12	EQV (逻辑等价运算)
13	IMP (逻辑蕴含运算)

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(※)。

SAMPLE	结果
$A\% = 125.67$	$A\% = 126$

※ 右边是 $n + m$ (n : 整数、 m : 小数) 的时，将 n ($m < 0.5$)、 $n+1$ ($0.5 \leq m$) 赋值至左边。

2. 如果用实数对整数进行四则运算，则结果为实数。

SAMPLE	结果
$A(0) = 125 * 0.25$	$A(0) = 31.25$

3. 如果用整数除以整数，则余数将被舍去，结果为整数。

SAMPLE	结果
$A(0) = 100 / 3$	$A(0) = 33$

2 字符串运算

2.1 字符串的联结

字符串可通过运算符 " + " 联结。

SAMPLE	结果
A\$="YAMAHA" B\$="ROBOT" C\$="LANGUAGE" D\$="MOUNTER" E\$=A\$+" "+B\$+" "+C\$ F\$=A\$+" "+D\$ PRINT E\$ PRINT F\$	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

SAMPLE	结果
"AA" < "AB" "X&" > "X#" "DESK" < "DESKS"	以下示例均为“真”。

坐标点数据分为关节坐标类型与正交坐标类型 2 种。

坐标点编号范围为 0 ~ 29999。

坐标类型	数据格式	说明
关节坐标类型	±nnnnnnn	带符号的 8 位数以下的十进制常量。 可在 -99,999,999 ~ 99,999,999 的范围内指定。 单位：[脉冲]
正交坐标类型	±nnn.nn ~ ±nnnnnnn	小数部分为 3 位数以下，合计位数为 7 位数以下的带符号十进制小数。 单位：[mm] 或 [度]



补充

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

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

YK-TW 系列机器人的第 1 机械臂及第 2 机械臂的动作范围扩展到 360 度以上。

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

因此，将正交坐标数据(单位：毫米)转换为关节坐标数据(单位：脉冲)时，存在多解，无法确定位置。

为了在转换为关节坐标时能够正确地确定机器人的位置和机械臂的姿势，我们在以毫米为单位的坐标点数据的扩展设置的手系统标志之后，追加了第 1 机械臂转速信息和第 2 机械臂转速信息。

根据设定的第 1 机械臂转速信息和第 2 机械臂转速信息，将正交坐标数据(单位：毫米)转换为关节坐标数据(单位：脉冲)。

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

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

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

4 DI / DO 条件表达式

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、八进制常量、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

SAMPLE	说明
WAIT DI (31) =1 OR DI (34) =1	等待 DI 31 或 DI 34 中的任意一个执行 (ON) 为止

第 5 章

多台机器人控制

1	概要	5-1
2	机器人指定命令一览表.....	5-2

RCX340/RCX341/RCX320 控制器可对多台（最多 4 台）机器人进行控制。

此外，还可使用多任务功能使数台机器人非同步动作。

若要使用此功能，必须在出厂时在系统生成中进行多台机器人或附加轴的设置。

机器人的轴可设置如下。

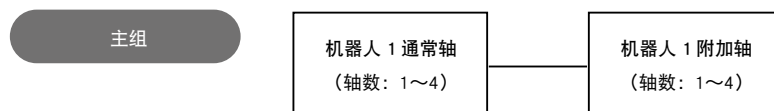
- 机器人 1 (4 轴)
- 机器人 1 (1 轴) + 机器人 2 (1 轴) + 机器人 3 (1 轴) + 机器人 4 (1 轴)
- 机器人 1 (6 轴) + 机器人 2 (2 轴) (使用了 YC-LINK/E 选件时)
- 机器人 1 (4 轴) + 机器人 2 (4 轴) (使用了 YC-LINK/E 选件时)
- 机器人 1 (2 轴) + 机器人 2 (2 轴)
- 机器人 1 (4 轴) + 机器人 2 (4 轴) + 机器人 3 (4 轴) + 机器人 4 (4 轴) (主控制器: RCX340。使用了 YC-LINK/E 选件时))
- 机器人 1 (2 轴) + 机器人 2 (2 轴) + 机器人 3 (4 轴) + 机器人 4 (4 轴) (主控制器: RCX320。使用了 YC-LINK/E 选件时))

此外，各机器人由通常轴和附加轴构成。

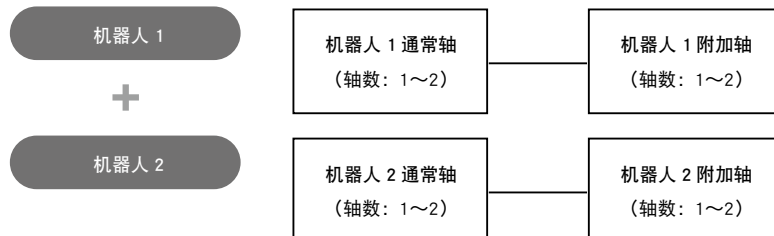
采用 1 台机器人、不使用附加轴时，将变为仅限通常轴的设置。

轴构成

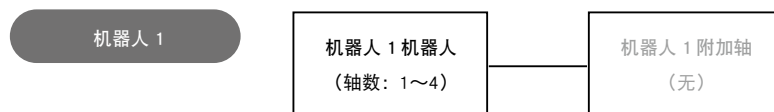
1. 机器人 1 时



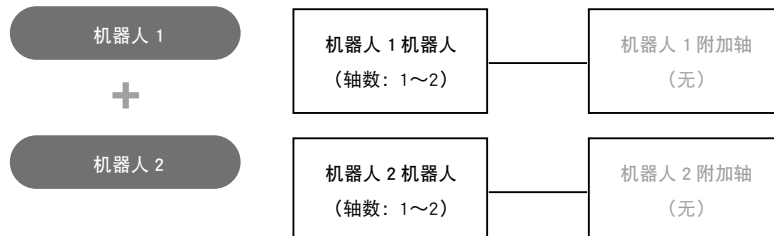
2. 机器人 1 + 机器人 2 时



3. 1 台机器人不使用附加轴时



4. 采用 2 台机器人、不设置附加轴时



35501-R9-00

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

运算符	命令名	
机器人动作	DRIVE	DRIVEI
	MOVE	MOVEI
	MOVET	PATH
	PMOVE	SERVO
	WAIT ARM	
坐标控制	CHANGE	HAND
	CHGWRK	WRKDEF
	LEFTY	PATH
	RIGHTY	SHIFT
状态更改	ACCEL	ARCHP1
	ARCHP2	ARMSEL
	ARMTYP	ASPEED
	AXWGHT	CONTPLS
	DECEL	INROFST
	MSPEED	ORGORD
	OUTPOS	SCRINR
	SPEED	TOLE
	WEIGHT	WEIGHTG
坐标点运算	CORRECTXY	JTOXY
	WHERE	WHRXY
	XYTOJ	
参数更改	ACCEL	ARCHP1
	ARCHP2	ARMTYP
	AXWGHT	CONTPLS
	DECEL	INROFST
	ORGORD	OUTPOS
	SCRINR	TOLE
	WEIGHT	WEIGHTG
状态参照	ABSRPOS	ARMCND
	ARMSEL	ARMTYP
	CORRECTXY	CURTQST
	CURTRQ	MCHREF
	WHERE	WHRXY
扭矩控制	TORQUE	TRQSTS
	TRQTIME	CURTRQ

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

第 6 章

多任务

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

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

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

- 最多可执行任务 1 ~ 任务 16 的 16 个任务
- 可对任务添加优先级，并优先执行优先级较高的任务
- 优先顺序可以在 1 ~ 64 之间自由设定。
- 数字越小，优先级越高，数字越大，优先级越低(高：1 ⇔ 64：低)

所谓“任务”，是指一系列工作的汇总，其定义如下所述。

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

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

3 任务的状态与变化

任务的状态有以下 6 种类型。

1. STOP 状态 (停止状态)

虽然存在任务，但是仍处于未执行任务的状态。

2. RUN 状态 (执行状态)

存在任务，且正在执行 CPU 分配的任务处理。

3. READY 状态 (可执行状态)

存在任务，且正在等待 CPU 分配任务处理的状态。

4. WAIT 状态 (等待状态)

存在任务，且任务处理因某种现象正在等待的状态。

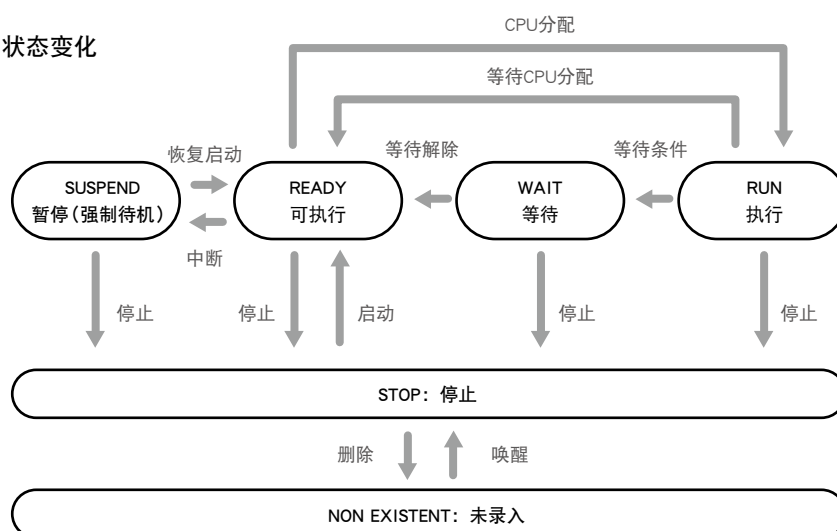
5. SUSPEND 状态 (强制待机状态)

存在任务，且任务处理正在强制待机的状态。

6. NON EXISTENT 状态 (未录入状态)

不存在任务的状态 (唤醒需要使用 START 命令)。

任务的状态变化



MEMO

当前任务的状态可以通过 REFTSK 函数获得。

参照 关于 REFTSK 函数的详细内容，请参阅第 8 章“101 REFTSK”。

3.1 启动任务

执行 START 命令后，指定程序将被注册为任务，变为 RUN 状态。

如果 START 命令中未指定任务号码 (1 ~ 16)，则自动指定未启动任务中号码最小的任务。

参照 关于 START 命令的详细内容，请参阅第 8 章“123 START”。

MEMO

- 执行 LOAD 命令后，指定程序将被注册为任务，变为 STOP 状态。
关于 LOAD 命令的详细内容，请参阅第 12 章“2.1 程序操作”的“1. 任务注册”。
- 在 START 命令、LOAD 命令所指定的任务号码上，已录入程序时，将出现“6.215:任务正在运行”的错误提示。
- 在所有任务号码已录入程序的状态下，不指定任务号码而执行 START 命令、LOAD 命令，将出现“6.263:任务数超限”的错误提示。
- 执行 HALTALL 命令时，结束所有任务，任务变为未注册状态。
已设置主程序时将主程序注册到任务 1，在首行停止。
未设置主程序时将最后执行的程序 (当前程序) 注册到任务 1，在首行停止。

参照 关于主程序的详细内容，请参阅别册《用户手册》的“设置主程序”。

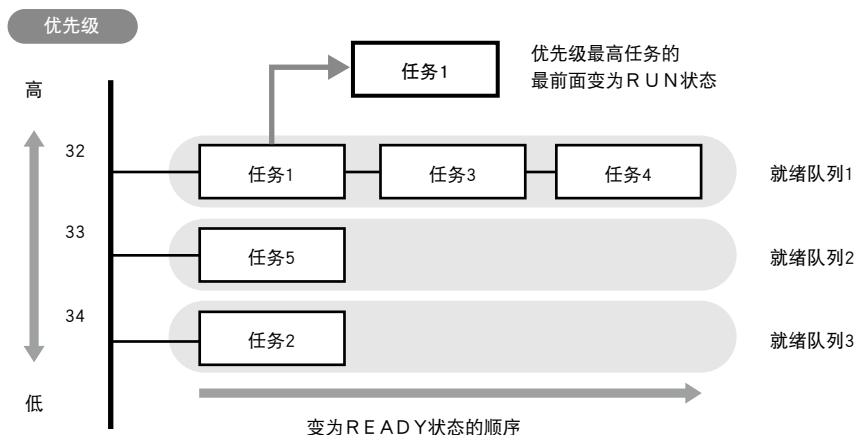
3.2 任务的排程

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，优先级越高。

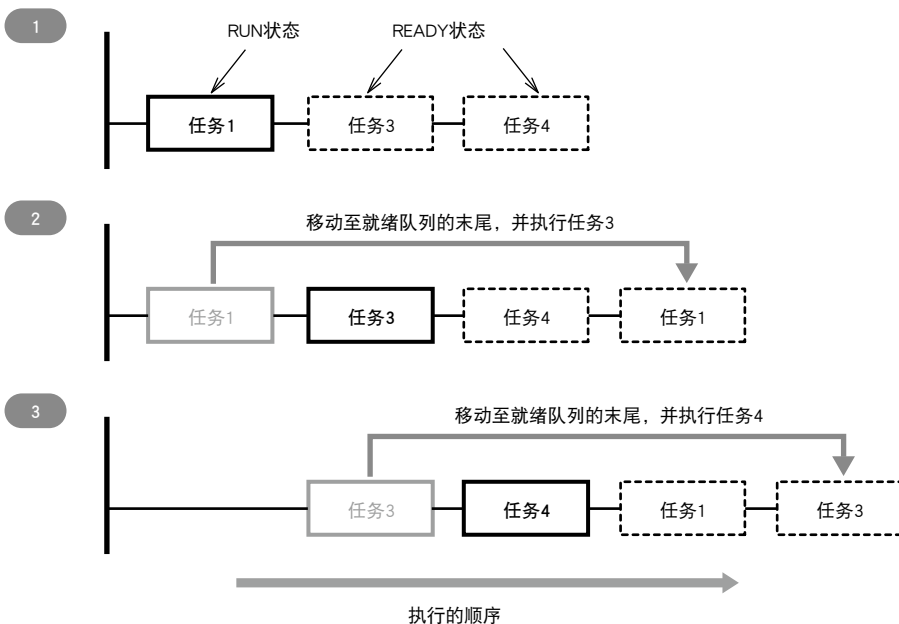
任务的排程



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



补充

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，在相同优先级（就绪队列）以上的任务中没有 READY 状态的其他任务时，将再次执行相同的任务。

3.3 任务的等待条件

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

1. 执行变为 WAIT 状态的命令后，将发生以下状态变化。

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

MEMO

例如，如果执行了 MOVE 语句（变为 WAIT 状态的命令之一），CPU 对驱动器发出“移动”的指示后，驱动器将等待“移动完成”的应答。这是等待的状态。

此时，执行了 MOVE 语句的任务将变为 WAIT 状态，并移动至就绪队列的末尾，下一个任务将变为 RUN 状态。

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

- 发生了 WAIT 状态等待条件的任务 → READY 状态

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

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

MEMO

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



补充

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。

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 DRIVEI SERVO	MOVEI PMOVE WAIT ARM	MOVET PATH	DRIVE MOTOR
	参数命令	ACCEL AXWGHT ORGORD WEIGHT	ARCHP1 CONTPLS OUTPOS WEIGHTG	ARCHP2 DECEL SCRINR	ARMTYP INROFST TOLE
	机器人状态 更改命令	CHANGE ASPEED	SHIFT SPEED	LEFTY MSPEED	RIGHTY
等待时间经过		DELAY, SET（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			

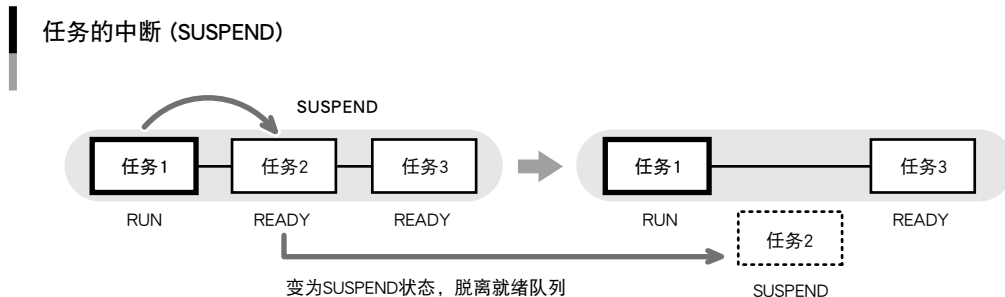
MEMO

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

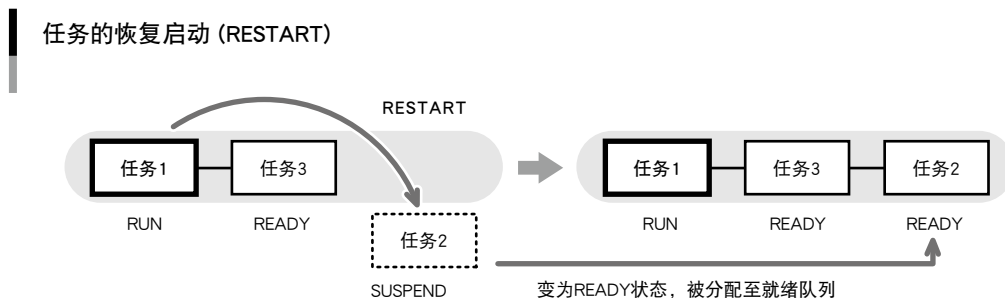


35604-R7-00

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



35605-R7-00

3.6 任务的删除

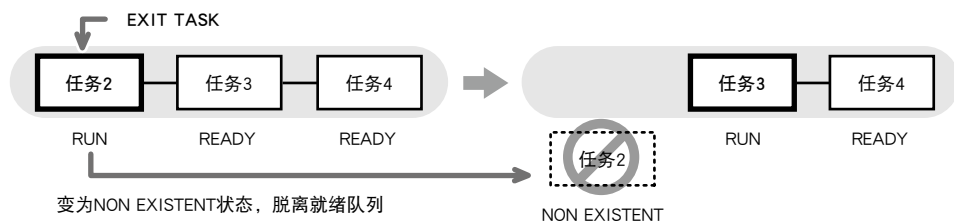
当前任务的删除 (EXIT TASK)

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。

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 EXISTENT 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

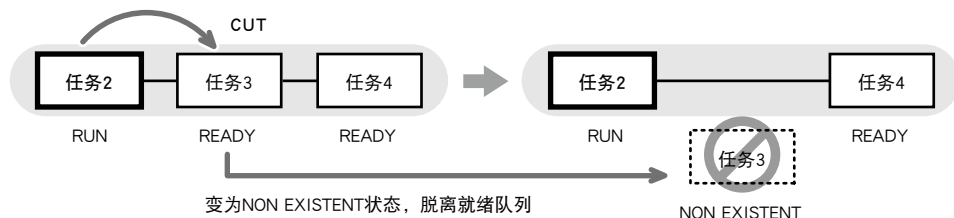
其他任务的删除 (CUT)

使用 CUT 命令可删除其他任务并使其变为 NON EXISTEN(未录入) 状态。

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 EXISTENT 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。

3.7 任务的停止

由于以下原因，所有任务均变为 STOP (停止) 状态。

1. 执行了 HALT 命令 (停止、复位)

程序将被复位，任务 1 以外的任务变为 NON EXISTENT 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。



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

各任务按照排程执行处理。以下所示为多任务程序的示例。

SAMPLE	说明
' TASK1 START <SUB_TSK2>, T2 START <SUB_TSK3>, 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 HALTALL	任务 1
' TASK2 *ST2: IF DI (20) = 1 MO (20) = 1 DELAY 100 ELSE MO (20) = 0 ENDIF GOTO *ST2 EXIT TASK	程序名称：SUB _ TSK2 从此处开始为任务 2 任务 2 至此为止
' TASK3 *ST3: IF DI (21) = 0 THEN *ST3 IF DI (30) = 0 THEN *ST3 IF DI (33) = 0 THEN *ST3 MO (21) = 1 EXIT TASK	程序名称：SUB _ TSK3 从此处开始为任务 3 任务 3 至此为止

所有全局变量、静态变量、输入输出变量、坐标点数据、位移坐标数据、机械手数据、工件数据、及托盘数据在所有任务中均是共享的。
因此，可在多个任务中使用相同的变量或数据、控制各任务的执行。

 MEMO

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

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

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

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

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

SAMPLE	说明
FLAG1 = 0	主任务
*MAIN_TASK:	
IF FLAG1=0 THEN	
FLAG1 = 1	
START <SUB_PGM>, T2, 31	将<SUB __ PGM> 作为任务 2 , 以优先级 31 启动
ENDIF	
GOTO *MAIN_TASK	
HALTALL	
 '=====	副任务(程序名:SUB __ PGM)
' TASK2	
'=====	
*TASK2:	
DRIVE (1,P1)	
WAIT ARM(1)	
DRIVE (1,P2)	
WAIT ARM(1)	
FLAG1 = 0	
EXIT TASK	

第 7 章

逻辑控制功能

1	关于逻辑控制功能	7-1
2	逻辑控制程序规格	7-1
3	创建逻辑控制程序	7-2
4	执行逻辑控制程序	7-5
5	编写逻辑控制程序	7-6

RCX340/RCX341/RCX320 与通常的机器人程序不同，可创建对机器人输入输出 (DI, DO, MO, LO, TO, SI, SO) 进行高速处理的程序 (逻辑控制程序)。

- 使用该功能，可监控传感器、按钮开关、电磁阀等的输入输出信号并使其运行。打开控制器电源的同时，开始运行。
- 使用机器人语言描述逻辑控制程序 (不使用梯形语言等)。
- 将创建程序时的名称设为“SEQUENCE”后，作为逻辑控制程序被控制器识别。
- 关于逻辑控制程序的执行条件，请参阅“3 执行逻辑控制程序”。
- 运行逻辑控制程序时，无法通过程序复位，对通用输出进行复位。



补充

- 专用输入“DI10*:逻辑控制输入”打开时，逻辑控制程序将按一定周期动作。*插入 SIO 基板时，则为“SI10”
- 执行逻辑控制程序时的输出如下：
 - > 插入专用 PIO 基板时…专用输出“DO12：逻辑控制程序运行中”信号
 - > 插入 SIO 基板时…专用输出“SO12：逻辑控制程序运行中”信号



MEMO

在逻辑控制程序运行过程中，也可按以下方法复位通用输出。

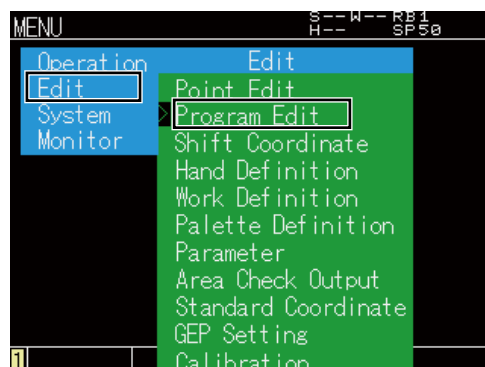
- 将控制器参数的逻辑控制标志值设为 3。
- 勾选支持软件逻辑控制执行标志对话框中的允许输出复位。

项 目	内 容
命 令	反相 (NOT)、逻辑与 (AND)、逻辑和 (OR)、异或 (XOR)、逻辑等价运算 (EQV)、逻辑蕴含运算 (IMP)
输入输出	与机器人语言相同
程序容量	8,192 字节 (最多可指定 2,048 个变量)
扫描时间	根据步数，自动在 2msec ~ 8msec 之间变化

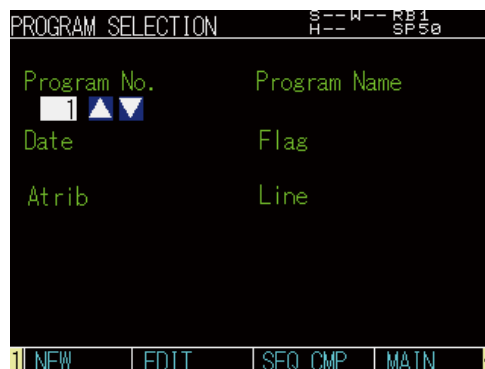
3 创建逻辑控制程序

3.1 程序方法

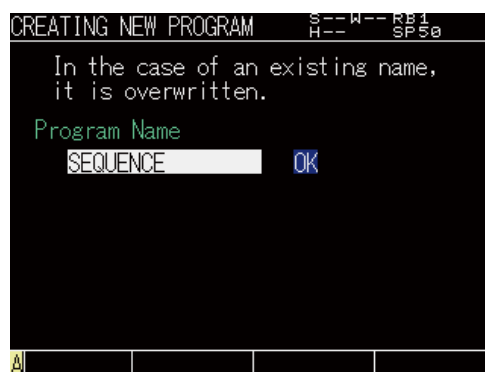
Step 1 在手持编程器菜单画面的“EDIT(编辑)”菜单中选择“Program Edit(编辑程序)”。



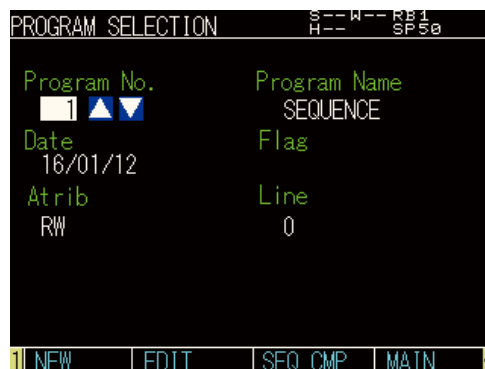
Step 2 在选择程序画面中按下“F1: NEW(新建)”。



Step 3 在新建程序画面的 Program Name 框中输入“SEQUENCE”后按下“OK”。



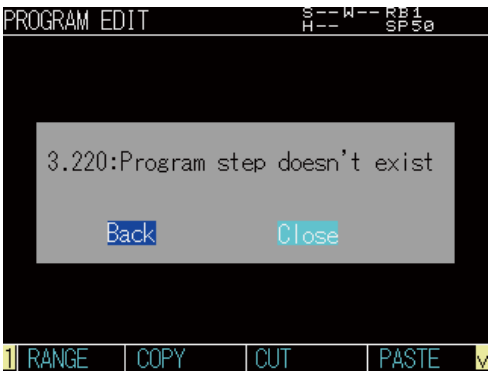
Step 4 通过选择程序画面的  选择“SEQUENCE”程序后，按下“F2: EDIT(编辑)”。



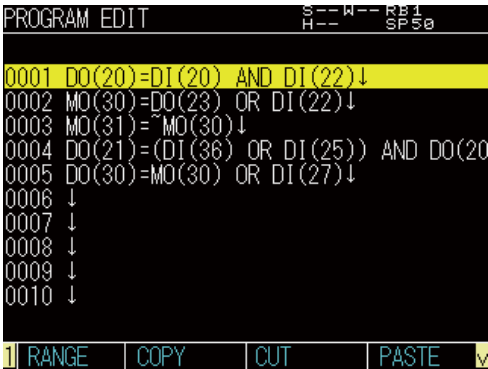
Step5 如果显示“3.220 : Program stop doesn't exist (程序步骤不存在)”的信息，请按下“Close(关闭)”。



补充
新建程序时，程序中未写入内容，因此发生警报。
如果程序中已存在机器人语言，则不发生该警报。



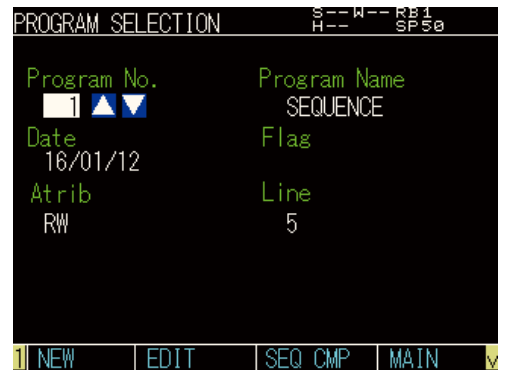
Step6 在“编辑程序”画面输入程序。可用命令受到一定限制，但编辑方法与通常的机器人程序相同。
可输入的命令可参考本章“4 编写逻辑控制程序”。



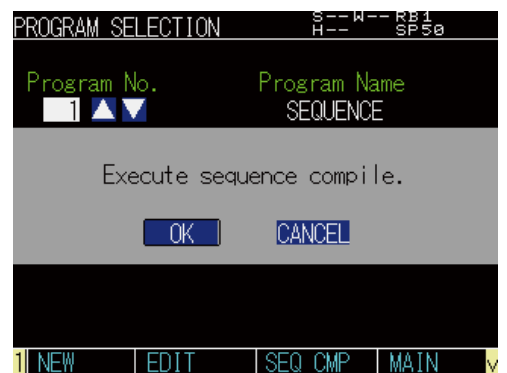
3.2 编译

进行编译，生成逻辑控制程序的执行程序。

Step 1 在选择程序画面中按下“F3: SEQ CMP（编译）”。

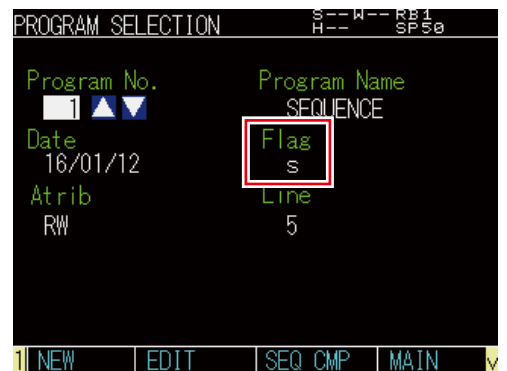


Step 2 显示是否编译逻辑控制程序的询问信息。
按下“OK”后开始编译。



MEMO

- 显示“输入的命令语句格式不正确”的错误提示。
- 编译正常结束后，返回至选择程序画面，Flag 显示为“s”。
这表示逻辑控制程序的编译正常结束，并已创建执行程序。



MEMO

在下列情况下，将删除逻辑控制程序用执行程序，且不显示“s”标志。此时，将禁止执行逻辑控制程序。

1. 删除逻辑控制程序时
2. 编辑逻辑控制程序时
3. 对程序数据进行初始化时
4. 发生“9.729：逻辑控制对象校验和错误”时

4 执行逻辑控制程序

只有以下条件全部成立，才会执行逻辑控制程序。

1. 已编译逻辑控制程序
2. 逻辑控制程序的执行标志已设为允许状态。（有关逻辑控制程序执行标志的设置，请参阅别册《用户手册》）
3. 已打开外部输入的逻辑控制输入“DI10”（插入 SIO 基板时，则为“SI10”）

逻辑控制程序执行状态



36710-R9-00

4.1 逻辑控制程序的逐步执行

如需逐行执行逻辑控制程序以确认动作，可按照与普通程序相同的方法执行。

详细内容，请参阅别册《用户手册》。

逐步执行时，无需满足上一项的执行条件。

仅通过输入输出变量和逻辑运算符的组合赋值语句，描述逻辑控制程序。

5.1 赋值语句

格式	
输出变量 = 表达式	
标注	说明
表达式	可使用以下任意项。 · 并行输入和输出变量 · 内部输出变量 · 机械臂锁定输出变量 · 定时输出变量 · 串行输入和输出变量 · 上述逻辑运算表达式

5.2 输入输出变量

各变量仅可使用 1 比特指定的格式。

SAMPLE	说明
DO (35)	正确示例
MO (24)	
DI (16)	
DO (37, 24)	错误示例
DI 3 (4)	
MO 3 ()	

5.2.1 输入变量

● 并行输入变量

格式		
DI(mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7

表示并行输入信号的状态。

● 串行输入变量

格式		
SI(mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7

表示串行输入信号的状态。仅限参照，无法进行控制。

5.2.2 输出变量

● 并行输入变量

格式		
DO(mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7
并行输出或参照输出状态。端口 0, 1 仅限参照，无法输出。		

● 内部输出变量

格式		
MO(mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27, 30 ~ 37
b	比特编号	0 ~ 7
在控制器内部使用。端口 30 ~ 37 仅限参照，无法控制开 / 关。		

● 机械臂锁定输出变量

格式		
LO(mb)		
标注	值	范围
m	端口编号	0、1
b	比特编号	0 ~ 7
用于禁止机械臂（轴）的移动。打开时禁止移动。 · LO(00) ~ LO(07)：分别对应第 1 轴到第 8 轴。 · LO(10) ~ LO(17)：分别对应第 9 轴到第 16 轴。		

● 定时输出变量

格式		
TO(mb)		
标注	值	范围
m	端口编号	0、1
b	比特编号	0 ~ 7
TO(00) ~ TO(17) 共计 16 点。 通过定时定义语句 TIM00 ~ 17 定义各变量的定时。 参照 本章“5.3 定时定义语句”		

● 串行输出变量

格式		
SO(mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7
控制或参照串行输出信号的状态。端口 0 仅限参照，无法进行控制。		

5.3 定时定义语句

格式		
TIMmb = 时间		
标注	值	范围
时间	时间	0 ~ 2, 147, 483, 647msec (0 秒~ 24 天)
m	端口编号	0、1
b	比特编号	0 ~ 7

含义 设置定时输出变量的定时值。该定义语句可出现在程序的任意位置。
如果省略定时输出变量，则定时输出变量的设定值将变为 0。

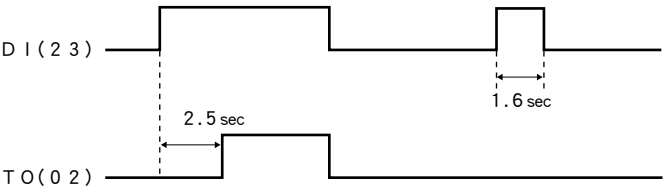
- TIM00 ~ TIM17 分别对应 TO(00) ~ TO(17) 的定时输出变量。
能以 1msec 为单位进行设置，但受到以下周期的影响。
- 逻辑控制程序的扫描周期 (2 ~ 8msec)
 - PIO 基板的更新周期 (2 ~ 4msec)
 - SIO 基板的更新周期 (5msec)

● 串行输入变量

SAMPLE	说明
TIM02 = 2500	将定时 02 设为 2.5 秒
TO (02) = DI (23)	打开 DI (23)，将启动定时

- DI(23) 开启 2.5 秒后 TO(02) 开启。
- DI(23) 关闭后，TO(02) 也将关闭。
- 如果 DI(23) 开启的时间小于 2.5 秒，则 TO(02) 不开启。

定时使用示例：时序图



5.4 运算符（逻辑运算符）

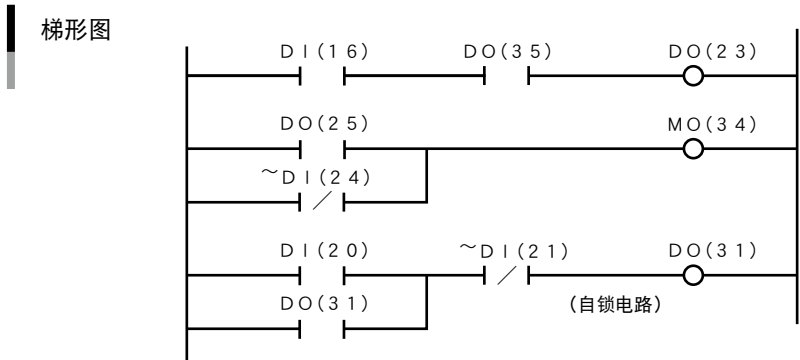
运算符	功能	含义
NOT, ~	否定	使各个比特反转
AND、&	逻辑积	两个比特均为 1 时，逻辑积为 1
OR,	逻辑和	任意一个比特为 1 时，逻辑和为 1
XOR	排他逻辑和	两个比特不同时，排他逻辑和为 1
EQV	逻辑等价运算	双方的比特相等时结果为 1
IMP	逻辑包含运算	第 1 个比特为 1、第 2 个比特为 0 时结果为 0

5.5 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)
5	XOR (排他逻辑和)
6	EQV (逻辑等价运算)
7	IMP (逻辑蕴含运算)

● 梯形语句替换示例

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



MEMO

- 反相运算符无法在“(”前及赋值语句左侧使用。例如，不可使用下列语句：
 - $DO(21) = \sim(DI(30) | DI(32))$
 - $\sim DO(30) = DI(22) \& DI(27)$
- 赋值语句右侧无法指定数值。
 - $MO(35) = 1$
 - $DO(26) = 0$
- 程序末尾无需定义“HALT”或“HOLD”。
- 逻辑控制程序中使用的各变量与机器人程序通用，使用相同变量时需引起注意。

第 8 章

机器人语言

机器人语言一览表	8-1
1 ABS.....	8-6
2 ABSRPOS.....	8-7
3 ACCEL.....	8-8
4 ARCHP1 / ARCHP2	8-9
5 ARMCND	8-10
6 ARMSEL	8-11
7 ARMTYP	8-12
8 ASPEED.....	8-13
9 ATN / ATN2	8-14
10 AXWGHT	8-15
11 BRKDIAG.....	8-16
12 CALL.....	8-17
13 CHANGE.....	8-18
14 CHGPRI	8-19
15 CHGWRK.....	8-20
16 CHR\$.....	8-21
17 CLOSE.....	8-22

18	CONTPLS	8-23
19	CORRECTXY	8-24
20	COS	8-25
21	CREWRK	8-26
22	CURTQST	8-27
23	CURTRQ	8-28
24	CUT	8-29
25	DATE\$	8-30
26	DECEL	8-31
27	DEF FN	8-32
28	DEGRAD	8-33
29	DELAY	8-34
30	DI	8-35
31	DIM	8-36
32	DIST	8-37
33	DO	8-38
34	DRIVE	8-40
35	DRIVEI	8-43
36	END SELECT	8-48
37	END SUB	8-49
38	ERR / ERL	8-50
39	ETHSTS	8-51
40	EXIT FOR	8-52
41	EXIT SUB	8-53
42	EXIT TASK	8-54
43	FOR ~ NEXT	8-55
44	GEPSTS	8-56
45	GOSUB ~ RETURN	8-57
46	GOTO	8-58
47	HALT	8-59
48	HALTALL	8-60
49	HAND	8-61
50	HOLD	8-66
51	HOLDALL	8-67
52	IF	8-68

53	INPUT	8-70
54	INROFST.....	8-71
55	INT	8-72
56	JTOXY	8-73
57	LEFT\$	8-74
58	LEFTY	8-75
59	LEN.....	8-76
60	LET	8-77
61	LO	8-80
62	LOCx	8-82
63	LSHIFT.....	8-84
64	MCHREF.....	8-85
65	MID\$.....	8-86
66	MO	8-87
67	MOTOR.....	8-89
68	MOVE	8-90
69	MOVEI	8-107
70	MOVET	8-108
71	MTRDUTY.....	8-109
72	OFFLINE	8-110
73	ON ERROR GOTO.....	8-111
74	ON ~ GOSUB	8-112
75	ON ~ GOTO.....	8-113
76	ONLINE.....	8-114
77	OPEN.....	8-115
78	ORD.....	8-116
79	ORGORD	8-117
80	ORIGIN	8-118
81	OUT	8-119
82	OUTPOS.....	8-120
83	PATH.....	8-121
84	PATH END.....	8-125
85	PATH SET	8-126
86	PATH START	8-128
87	PDEF.....	8-130

88	PGMTSK.....	8-131
89	PGN	8-132
90	PMOVE	8-133
91	Pn	8-136
92	PPNT	8-138
93	PRINT	8-139
94	PSHFRC	8-140
95	PSHJGSP	8-141
96	PSHMTD.....	8-142
97	PSHRSLT	8-143
98	PSHSPD	8-144
99	PSHTIME	8-145
100	PUSH.....	8-146
101	RADDEG.....	8-151
102	REFTSK	8-152
103	REM	8-153
104	RESET	8-154
105	RESTART	8-155
106	RESUME.....	8-156
107	RETURN	8-157
108	RIGHT\$.....	8-158
109	RIGHTY.....	8-159
110	RSHIFT	8-160
111	SCRINR	8-161
112	SELECT CASE ~ END SELECT.....	8-162
113	SEND	8-163
114	SERVO.....	8-165
115	SET	8-166
116	SETGEP.....	8-167
117	SGL.....	8-168
118	SGR	8-169
119	SHARED	8-170
120	SHIFT.....	8-171
121	SI	8-172
122	SID.....	8-173

123	SIN	8-174
124	SIW	8-175
125	Sn	8-176
126	SO	8-177
127	SOD	8-178
128	SOW	8-179
129	SPEED	8-180
130	SQR	8-181
131	START	8-182
132	STR\$	8-183
133	SUB ~ END SUB	8-184
134	SUSPEND	8-186
135	SWI	8-187
136	TAN	8-188
137	TCOUNTER	8-189
138	TIME\$	8-190
139	TIMER	8-191
140	TO	8-192
141	TOLE	8-193
142	TORQUE	8-194
143	TSKPGM	8-196
144	VAL	8-197
145	WAIT	8-198
146	WAIT ARM	8-199
147	WEIGHT	8-200
148	WEIGHTG	8-201
149	WEND	8-202
150	WHERE	8-203
151	WHILE ~ WEND	8-204
152	WHRXY	8-205
153	WRKDEF	8-206
154	XYTOJ	8-208

机器人语言一览表

以下为您介绍机器人语言一览表的查看方法。

(1) 名称	(2) 功能	(3) 联机	(4) 种类
DIM	声明数组变量	×	命令

(1) 名称

表示各机器人语言的名称。

(2) 功能

介绍机器人语言的功能。

(3) 联机

当该项目为“✓”时，表示可在联机命令中进行使用。

当该项目为“×”时，表示有一部分无法在联机命令中执行的操作数。

对于使用在联机命令执行机器人语言时的记述格式，[参照](#) 第 12 章“7 机器人语言的单独执行”

(4) 种类

用“命令”或“函数”表示机器人语言的种类。

在命令与函数中均可使用时，表示为“命令 / 函数”。

名称	功能	联机	种类
A			
ABS	计算指定值的绝对值	✓	函数
ABSRPOS	计算指定机器人的指定轴的机台参考值 (返回原点的方式仅为标注方式时有效)	✓	函数
ACCEL	设定 / 获取指定机器人的加速度系数参数	✓	命令 / 函数
ARCHP1	设定 / 获取指定机器人的拱形距离 1 参数	✓	命令 / 函数
ARCHP2	设定 / 获取指定机器人的拱形距离 2 参数	✓	命令 / 函数
ARMCND	获取指定机器人的当前机械臂状态	✓	函数
ARMSEL	获取指定机器人的当前手系统设定	✓	命令 / 函数
ARMTYP	获取指定机器人的手系统设定	✓	命令 / 函数
ASPEED	设定 / 获取指定机器人的自动移动速度	✓	命令 / 函数
ATN	计算指定值的反正切值	✓	函数
ATN2	计算指定 X-Y 坐标的反正切值	✓	函数
AXWGHT	设定 / 获取指定机器人的轴前端质量参数	✓	命令 / 函数
B			
BRKDIAG	获取指定轴的制动器故障诊断结果	✓	函数
C			
CALL	调用子过程	×	命令
CHANGE	对指定机器人的机械手数据进行切换	✓	命令
CHGPRI	更改指定任务的优先级	✓	命令
CHGW RK	对指定机器人的工件数据进行切换	✓	命令
CHR\$	获取带有指定字符编码的字符	✓	函数
CLOSE	关闭指定的通用以太网端口	✓	命令
CONTPLS	设定 / 获取指定机器人的 CONT 脉冲参数	✓	命令 / 函数
CORRECTXY	获取修正工件把持位置偏移后的目标位置	✓	函数

名称	功能	联机	种类
COS	计算指定值的余弦值	✓	函数
CREWRK	创建以两点间差分为偏移的工件	✓	函数
CURTQST	获取相对于轴的定格扭矩值当前扭矩值的比率	✓	函数
CURTRQ	获取相对于轴的最大扭矩值当前扭矩值的比率	✓	函数
CUT	强制结束执行中或暂停中的其他任务	✓	命令
D			
DATE\$	通过 "yy/mm/dd" 形式的字符串计算日期	✓	函数
DECEL	设定 / 获取指定机器人的减速比率参数	✓	命令 / 函数
DEF FN	定义用户可使用的函数	×	命令
DEGRAD	将值转换为弧度 (↔ RADDEG)	✓	函数
DELAY	仅等待指定的时间 (单位 ms)	×	命令
DI	获取并行端口的输入状态	✓	函数
DIM	对数组变量的名称及元素数进行声明	×	命令
DIST	计算指定 2 个点之间的距离	✓	函数
DO	将指定值输出至 DO 端口	✓	命令 / 函数
DRIVE	以轴为单位使指定机器人进行绝对位置移动	✓	命令
DRIVEI	以轴为单位使指定机器人进行相对位置移动	✓	命令
E			
END SELECT	结束 SELECT CASE 语句	×	命令
END SUB	结束子过程的定义	×	命令
ERL / ERR	赋予发生错误的行编号 / 赋予发生错误时的错误代码编号	✓	函数
ETHSTS	获取以太网端口的状态	✓	函数
EXIT FOR	强制结束 FOR 语句 ~ NEXT 语句的循环	×	命令
EXIT SUB	强制结束使用 SUB ~ END 定义的子过程	×	命令
EXIT TASK	结束当前正在执行的任务	×	命令
F			
FOR ~ NEXT	反复执行 FOR 语句的下一个语句至 NEXT 语句，直至变为控制反复的指定值为止	×	命令
G			
GEPSTS	获取通用以太网端口的状态	✓	函数
GOSUB ~ RETURN	通过 GOSUB 语句跳转至指定标签的子例程，并执行子例程	×	命令
GOTO	无条件跳转至标签所指定的行	×	命令
H			
HALT	停止程序并进行复位	×	命令
HALTALL	停止并复位所有的程序	×	命令
HAND	定义指定机器人的机械手	✓	命令
HOLD	暂停程序	×	命令
HOLDALL	暂停所有的程序	×	命令
I			
IF	根据条件分支控制流程	×	命令
INPUT	从手持编程器中对指定的变量赋值	✓	命令
INROFST	设定 / 获取水平多关节 R 轴惯性力矩偏移量参数	✓	命令 / 函数
INT	获得舍去小数点以下部分的整数值	✓	函数
J			
JTOXY	将关节坐标数据转换为指定机器人的正交坐标数据 (↔ XYTOJ)	✓	函数

名称	功能	联机	种类
L			
LEFT\$	从指定的字符串左侧抽出指定位数的字符串	✓	函数
LEFTY	将指定机器人的手系统设定为左手系统	✓	命令
LEN	获得指定字符串的长度(字节数)	✓	函数
LET	执行指定的赋值语句	✓	命令
LO	将被指定的值输出至 LO 端口, 并进行轴移动的禁止或解除	✓	命令 / 函数
LOCx	设定 / 获取坐标点数据为轴单位或者位移数据和工件数据为要素单位	✓	命令 / 函数
LSHIFT	仅将指定值的比特数向左偏移 (↔ RSHIFT)	✓	函数
M			
MCHREF	计算指定机器人轴的原点归复动作及绝对值搜索动作的机台参考	✓	函数
MID\$	从指定的字符串中抽出任意长度的字符串	✓	函数
MO	将指定值输出至 MO 端口	✓	命令 / 函数
MOTOR	控制马达电源状态	✓	命令
MOVE	使指定机器人的所有轴进行绝对移动	✓	命令
MOVEI	使指定机器人的所有轴进行相对移动	✓	命令
MOVET	对指定机器人的所有轴进行工具坐标相对移动	✓	命令
MTRDUTY	获取指定轴的马达负载率	✓	函数
O			
OFFLINE	将指定的通信端口设置为脱机模式	✓	命令
ON ERROR GOTO	在不停止程序的情况下跳转至标签所示的错误处理例程, 或者显示错误提示, 并停止程序的执行	×	命令
ON ~ GOSUB	根据条件, 通过 GOSUB 语句跳转至所指定的各标签的子例程, 并执行子例程	×	命令
ON ~ GOTO	根据条件跳转至标签所指定的各行	×	命令
ONLINE	将指定的通信端口设置为联机模式	✓	命令
OPEN	打开指定的通用以太网端口	✓	命令
ORD	获得指定字符串起始字符的字符编码	✓	函数
ORGORD	设定 / 获取针对指定机器人进行原点归复动作及绝对值搜索动作的轴顺序参数	✓	命令 / 函数
ORIGIN	对增量式规格的轴执行返回原点动作	✓	命令
OUT	开启指定输出端口的比特, 并结束命令语句	×	命令
OUTPOS	设定 / 获取指定机器人的 OUT 有效位置参数	✓	命令 / 函数
P			
PATH	PATH 设置 PATH 移动路径	×	命令
PATH END	PATH END 结束 PATH 移动的路径设置	×	命令
PATH SET	PATH SET 开始 PATH 移动的路径设置	×	命令
PATH START	PATH START 开始 PATH 移动	×	命令
PDEF	定义用于执行托盘移动命令的托盘	✓	命令
PGMTSK	获取已录入指定程序的任务编号	✓	函数
PGN	从指定的程序名中获取程序编号	✓	函数
PMOVE	执行指定机器人的托盘移动命令	✓	命令
Pn	在程序中定义坐标点	✓	命令
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
PRINT	在手持编程器的画面上显示字符串	×	命令
PSHFRC	设定 / 获取推压力参数	✓	命令 / 函数
PSHJGSP	设定 / 获取推压判定速度阈值参数	✓	命令 / 函数
PSHMTD	设定 / 获取推压方式参数	✓	命令 / 函数

名称	功能	联机	种类
PSHRSLT	获取 PUSH 语句结束时的状态	✓	函数
PSHSPD	设定 / 获取推压移动速度参数	✓	命令 / 函数
PSHTIME	设定 / 获取推压时间参数	✓	命令 / 函数
PUSH	以轴为单位进行推压动作	✓	命令
R			
RADDEG	将值转换为度 ($\leftrightarrow$ DEGRAD)	✓	函数
REFTSK	获取任务的状态	✓	函数
REM	记述注释语句	×	命令
RESET	关闭指定输出端口的比特	✓	命令
RESTART	重新启动暂停中的其他任务	✓	命令
RESUME	进行错误的恢复处理后, 恢复执行程序	×	命令
RETURN	使 GOSUB 中的分支处理返回至 GOSUB 的下一行	×	命令
RIGHT\$	从指定的字符串右侧抽出指定位数的字符串	✓	函数
RIGHTY	将指定机器人的手系统设定为右手系统	✓	命令
RSHIFT	仅将指定值的比特数向右偏移 ($\leftrightarrow$ LSHIFT)	✓	函数
S			
SCRINR	设定 / 获取水平多关节 R 轴惯性力矩参数	✓	命令 / 函数
SELECT CASE ~ END SELECT	根据条件分支控制流程	×	命令
SEND	转送文件	✓	命令
SERVO	对指定机器人中指定轴或所有轴的伺服 ON / OFF 状态进行控制	✓	命令
SET	开启指定输出端口的比特	×	命令
SETGEP	设置通用以太网端口	✓	命令
SGL	获取指定的整数型静态变量, 或将值赋给该变量	✓	命令 / 函数
SGR	获取指定的实数型静态变量, 或将值赋给该变量	✓	命令 / 函数
SHARED	使不提交变量便可在子过程中参照	×	命令
SHIFT	指定位移变量, 然后针对由该位移数据所指定的机器人位移坐标进行设定	✓	命令
SI	获取指定 SI 的状态	✓	函数
SID	以双字方式获取指定串行输入的文字信息状态	✓	函数
SIN	计算指定值的正弦值	✓	函数
SIW	获取指定串行输入的文字信息	✓	函数
Sn	在程序中定义位移坐标	✓	命令
SO	将指定值输出至 SO 端口	✓	命令 / 函数
SOD	向指定串行输出的文字信息输出双字, 或获取输出状态	✓	命令 / 函数
SOW	向指定串行输出的文字信息输出, 或获取输出状态	✓	命令 / 函数
SPEED	对指定机器人的程序移动速度进行变更	✓	命令
SQR	计算指定值的平方根	✓	函数
START	设定指定程序的任务编号及优先顺序, 并且启动该程序	✓	命令
STR\$	将指定值转换为字符串 ($\leftrightarrow$ VAL)	✓	函数
SUB ~ END SUB	定义子过程	×	命令
SUSPEND	暂停执行中的其他任务	×	命令
SWI	切换执行程序, 并在执行编译后从第 1 行开始执行	×	命令
T			
TAN	计算指定值的正切值	✓	函数
TCOUNTER	从 TCOUNTER 变量复位时开始, 每隔 10ms 输出计数递增值	✓	函数

名称	功能	联机	种类
TIME\$	通过 "hh:mm:ss" 形式的字符串计算当前时间	✓	函数
TIMER	用上午 0 点开始的秒计算当前时间	✓	函数
TO	将指定值输出至 TO 端口	✓	命令 / 函数
TOLE	设定 / 获取指定机器人的公差参数	✓	命令 / 函数
TORQUE	设定 / 获取指定机器人中指定轴的最大扭矩指令值	✓	命令 / 函数
TSKPGM	获取录入指定任务编号的程序编号	✓	函数
V			
VAL	将指定字符串标注的值转换为实际的数值 (↔ STR \$)	✓	函数
W			
WAIT	等待直至 DI / DO 条件表达式成立为止 (带超时)	×	命令
WAIT ARM	等待指定机器人的轴动作结束	×	命令
WEIGHT	设定 / 获取指定机器人的前端质量 (kg) 参数	✓	命令 / 函数
WEIGHTG	设定 / 获取指定机器人的前端质量 (g) 参数	✓	命令 / 函数
WEND	结束 WHILE 语句的命令区	×	命令
WHERE	通过关节坐标 (脉冲) 读出指定机器人机械臂的当前位置	✓	函数
WHILE ~ WEND	控制反复	×	命令
WHRXY	通过正交坐标 (mm、度) 读出指定机器人机械臂的当前位置	✓	函数
WRKDEF	工件定义 (作成指定编号的工件)	✓	命令
X			
XYTOJ	将坐标点变量的正交坐标数据转换为指定机器人的关节坐标数据 (↔ JTOXY)	✓	函数

ABS 计算绝对值	联机
	✓

函数

格式
ABS (表达式)

含义 返回<表达式>中指定值的绝对值。

SAMPLE	说明
A=ABS (-362.54) …… -362.54 的绝对值 (362.54) 赋予变量 A	

ABSRPOS

获取机器参照

联机



8

函数

格式	
ABSRPOS [机器人编号] (轴编号)	
值	范围
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6

含义 获取指定轴的机台参考值 (单位 : %)。

此函数的对象为, 返回原点方式设置为标注方式的轴。
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注方式的轴, 机器参照值在 44 ~ 56% 的范围内时, 可进行绝对式原点复位。

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

- A
- B
- C
- D
- E
- F
- G
- H
- I
- J
- K
- L
- M

ACCEL

设置 / 获取加速度系数参数

联机



命令

格式	
1. ACCEL [机器人编号] 表达式	
2. ACCEL [机器人编号] (轴编号) = 表达式	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6
表达式	1 ～ 100（单位：％）

含义 将直接加速度系数参数变更为＜表达式＞的值。

[形式 1] 对指定机器人中设定的所有轴进行变更。

[格式 2] 只更改＜轴编号＞中指定的轴。

函数

格式	
ACCEL [机器人编号] (轴编号)	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6

含义 获取＜轴编号＞指定轴的加速度参数值。

SAMPLE	说明
A=50	
ACCEL A	……………将机器人 1 中所有轴的加速度系数设为 50% 。
ACCEL (3) =100	……………仅将机器人 1 中第 3 轴设为 100% 。
' CYCLE WITH INCREASING ACCELERATION	
FOR A=10 TO 100 STEP 10	……………使加速度系数参数从 10% 至 100%，以 10% 为步长逐步变化
ACCEL A	
MOVE P,P0	
MOVE P,P1	
NEXT A	
A=ACCEL (3)	……………将机器人 1 中第 3 轴的加速度系数参数赋给变量 A
HALT “END TEST”	

ARCHP1 / ARCHP2

设置 / 获取参数的拱形距离

联机



8

命令

格式

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

格式

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

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6
表达式	0 ~ 99999999 (单位: 脉冲)

含义 ARCHP1 对应拱形距离 1 参数, ARCHP2 对应拱形距离 2 参数。
将参数的拱形距离变更为 <表达式> 的值。

[形式 1] 对指定机器人中设定的所有轴进行变更。
[格式 2] 将 <轴编号> 中指定轴的拱形距离参数更改为 <表达式> 中的值。

函数

格式

ARCHP1 [机器人编号] (轴编号)

格式

ARCHP2 [机器人编号] (轴编号)

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6

含义 ARCHP1 对应拱形距离 1 参数, ARCHP2 对应拱形距离 2 参数。
获取 <轴编号> 中指定轴的拱形距离参数值。

SAMPLE

说明

```
ARCHP1 (3) =10          .....将机器人 1 的第 3 轴的拱形距离 1 的参数值更改为 10 脉冲
ARCHP2 (3) =20          .....将机器人 1 的第 3 轴的拱形距离 2 的参数值更改为 20 脉冲
.
.
.
FOR B=1 TO 4
  SAV (B-1) =ARCHP1 (B) .....将 ARCHP1 (1) ~ (4) 赋值给数组变量 SAV (0) ~ (3)
NEXT
```

ARMCND

获取机械臂状态

联机



函数

格式	
ARMCND [<i>机器人编号</i>]	
值	范围
<i>机器人编号</i>	1 ～ 4（省略时，机器人 1 被指定。）

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

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

ARMSEL

设定 / 获取当前的手系统选择

联机



8

命令

格式

ARMSEL [机器人编号] 表达式

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
表达式	1 : 右手系统、2 : 左手系统

含义 设定水平多关节型机器人的当前手系统选择。
指定利用<机器人编号>设定手系统选择的机器人。
此命令仅在使用水平多关节型机器人时有效。

SAMPLE

说明

ARMSEL [2] 2将机器人 2 的手系统选择设定为左手系统

函数

格式

ARMSEL [机器人编号]

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
-------	-------------------------

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

SAMPLE

说明

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

ARMTYP

设定 / 获取程序复位时的手系统

联机



命令

格式	
ARMTYP [<i>机器人编号</i>] <i>表达式</i>	
值	范围
<i>机器人编号</i>	1 ～ 4（省略时，机器人 1 被指定。）
<i>表达式</i>	1：右手系统、2：左手系统

含义

设定水平多关节型机器人在程序复位时的手系统。
指定利用<机器人编号>设定手系统选择的机器人。
此函数仅在使用水平多关节型机器人时有效。

SAMPLE	说明
ARMTYP [2] 2	……将机器人 2 的手系统设定为左手系统

函数

格式	
ARMTYP [<i>机器人编号</i>]	
值	范围
<i>机器人编号</i>	1 ～ 4（省略时，机器人 1 被指定。）

含义

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

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

ASPEED

设定 / 获取机器人的自动移动速度

联机

✓

8

命令

格式

ASPEED [机器人编号] 表达式

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

表达式 1 ~ 100 (单位 : %)

含义 将直接自动移动速度变更为 <表达式> 的指定值。
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 命令设置的自动移动速度与 SPEED 等命令指定的程序移动速度的乘积决定。

动作速度 = 自动移动速度 × 程序移动速度

示例 :

自动移动速度	80%
× 程序移动速度	50%
移动速度 =	40% (80% × 50%)



补充

虽然规定了最高速度, 但并不保证按照指定速度移动。

[自动移动速度] 通过手持编程器的操作或 ASPEED 命令设置

[程序移动速度] 由 SPEED 命令或 MOVE、DRIVE 的速度指定设置

函数

格式

ASPEED [机器人编号]

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

含义 获取自动移动速度值。

SAMPLE

说明

```
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

ATN / ATN2

计算反正切值

联机



函数

格式
ATN (表达式)
格式
ATN2 (表达式 1, 表达式 2)

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

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

相关命令	COS, DEGRAD, RADDEG, SIN, TAN
------	-------------------------------

AXWGHT

设置 / 获取轴前端重量

联机



8

命令

格式

AXWGHT [机器人编号] (轴编号) = 表达式

值	范围
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6
表达式	根据设置的机器人而异

含义 将指定轴的轴前端重量参数变更为 <表达式> 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)。

函数

格式

AXWGHT [机器人编号] (轴编号)

值	范围
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6

含义 赋予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	说明
A=5 B=0 C=AXWGHT (1)获取机器人 1 的第 1 轴的前端重量的值 (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, WEIGHTG

BRKDIAG

获取指定轴的制动器故障诊断结果

联机



命令

格式	
BRKDIAG [机器人编号] (轴编号)	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6

含义 获取指定轴的制动器故障诊断结果。
根据诊断结果，获得以下值。

-1	未实施诊断
0	正常
1	马达制动器释放电压上升 1（无法释放 16 V 制动器）
2	马达制动器释放电压上升 2（无法释放 18 V 制动器）
3	马达制动器释放电压上升 3（无法释放 20 V 制动器）
4	故障（无法释放 22 V 制动器）

SAMPLE	说明
A=BRKDIAG [2] (3) ……将机器人 2 的第 3 轴的制动器故障诊断结果赋给变量 A	

参照 RCX3 用户手册“轴参数 制动器诊断次数 < BFDNUM >”

CALL

调用子过程

联机

×

8

命令

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。



补充

[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。

参照 第 3 章“8 值传递与引用传递”



MEMO

- CALL 语句可以连续使用最多 120 次。但是，针对 FOR 语句、GOSUB 语句等使用堆栈的命令，根据其使用状态及识别符的使用状况等，次数可能有所减少。
- 在利用 CALL 语句调用子过程时，请务必使用 END SUB 语句或 EXIT SUB 语句来结束子过程。通过 GOTO 等语句从子例程跳转时，可能会发生 [5.212：堆栈溢出] 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
```

参照 样本程序的说明请参阅本章“SUB ～END SUB”

相关命令

SUB, END SUB, EXIT SUB, SHARED

CHANGE

切换机械手

联机



命令

格式		
CHANGE [机器人编号]	Hn OFF	
标注	值	范围
	机器人编号	1 ~ 4（省略时，机器人 1 被指定。）
n	机械手编号	0 ~ 31

含义 通过 CHANGE 进行机械手的切换。

指定为 OFF 时，表示无机械手设定。
在切换机械手之前，请通过 HAND 语句、手持编程器和支持软件设置机械手。
如果指定了定义其他机器人的机械手数据，则出现 [6.258：机器人编号错误] 的提示。
参照 “HAND”

SAMPLE	说明					
HAND H1=	0	150.000	0.000			
HAND H2=	-5000	20.000	0.000			
P1=	150.000	300.000	0.000	0.000	0.000	0.000
CHANGE H2	……更改为机器人 1 的机械手 2					
MOVE P, P1	……机械手 2 的前端向 P1 移动 (1)					
CHANGE H1	……更改为机器人 1 的机械手 1					
MOVE P, P1	……机械手 1 的前端向 P1 移动 (2)					
HALT						

CHGPRI

更改任务的优先级

联机



8

命令

格式		
CHGPRI	Tn <程序名称> PGm	,p
标注	值	范围
m	程序编号	1 ~ 100
n	任务编号	1 ~ 16
p	任务优先级	1 ~ 64

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

SAMPLE	说明
START <SUB_PGM>, T2, 33	……主程序
*ST:	
MOVE P, P0, P1	
IF DI (20) = 1 THEN	……DI (20) 为 ON 时,
CHGPRI T2, 32	任务 2 的优先顺序为 No. 32
ELSE	……DI (20) 为 OFF 时,
CHGPRI T2, 33	任务 2 的优先顺序为 No. 33
ENDIF	
GOTO *ST	
HALTALL	
' SUBTASK ROUTINE	……程序名称 : SUB_PGM (副任务)
*SUBTASK:	……任务 2 从这里开始
IF LOC3 (WHERE) > 10000 THEN	
DO (20) = 1	
GOTO *SUBTASK	
ENDIF	
DO (20) = 0	
GOTO *SUBTASK	……任务 2 的处理至此结束
EXIT TASK	……退出任务 2 (当前任务 : SUB_PGM)

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

CHGWRK

切换工件数据

联机



命令

格式		
CHGWRK	[机器人编号]	Wn OFF
标注	值	范围
	机器人编号	1 ~ 4（省略时，机器人 1 被指定。）
n	工件编号	0 ~ 39

含义	通过 CHGWRK 进行工件数据的切换。
	指定为 OFF 时，表示无工件数据设定。
	在切换工件数据之前，请通过 WRKDEF 语句、手持编程器和支持软件设置工件数据。
参照	“WRKDEF” RCX3 用户手册“工件定义”

SAMPLE	说明
CHANGE H1	更改为机器人 1 的机械手 1
MOVE P, P1	机械手 1 的前端向 P1 移动
WRKDEF W1=	115.000 -50.000 0.000 30.000
CHGWRK W1	更改为机器人 1 的工件 1
MOVE P, P2	工件 1 的前端向 P2 移动
HALT	

相关命令	CREWRK, WRKDEF
------	----------------

CHR\$

计算带有指定字符编码的字符

联机



8

函数

格式

CHR\$ (表达式)

值

范围

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

说明

A\$=CHR\$ (65)将 "A" 赋值给 A\$

相关命令

ORD

A

B

C

D

E

F

G

H

I

J

K

L

M

CLOSE

关闭指定的通用以太网端口

联机



命令

格式

CLOSE GPm

标注	值	范围
m	通用以太网端口编号	0 ~ 7

含义 关闭指定的通用以太网端口编号的通信端口。

SAMPLE	说明
OPEN GP1	……打开通用以太网端口编号 1 的通信端口
SEND “123” TO GP1	……从通用以太网端口编号 1 发送字符串 “123”
SEND GP1 TO A\$	……从通用以太网端口编号 1 接收数据，将接收的数据保存至变量 A\$
CLOSE GP1	……关闭通用以太网端口编号 1 的通信端口

相关命令 OPEN, SEND, SETGEP, GEPSTS

CONTPLS

设定 / 获取指定机器人的 CONT 脉冲参数

联机



8

命令

格式

1. CONTPLS [机器人编号] 表达式
2. CONTPLS [机器人编号] (轴编号) = 表达式

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6 (省略时, 所有轴被指定。)
表达式	0 ~ 9999999 (单位: 脉冲)

含义 将轴的 CONT 脉冲参数变更为 <表达式> 的值。
[格式 1] 对指定机器人中设定的所有轴进行变更。
[格式 2] 只更改 <轴编号> 中指定的轴。

参照 本章“MOVE”CONT 指定

函数

格式

CONTPLS [机器人编号] (轴编号)

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6 (省略时, 所有轴被指定。)

含义 获取 <轴编号> 指定轴的 CONT 脉冲参数值。

SAMPLE 1

说明

CONTPLS 20000将机器人 1 的所有轴的 CONTPLS 参数变更为 20000

SAMPLE 2

说明

CONTPLS (3) = 20000将机器人 1 的第 3 轴的 CONTPLS 参数变更为 20000

SAMPLE 3

说明

A% = CONTPLS (3)获取机器人 1 的第 3 轴的 CONTPLS 参数的值,
赋值到整数型变量 A%

CORRECTXY

获取修正工件把持位置偏移后的目标位置

联机

✓

函数

形式

CORRECTXY [机器人编号] (机器人前端基准位置, 工件基准位置, 工件位置, 校正前目标位置)

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
机器人前端基准位置	坐标点 (单位: mm)
工件基准位置	坐标点 (单位: mm)
工件位置	机器人前端位于机器人前端基准位置时的工件位置坐标 (单位: mm)
校正前目标位置	坐标点 (单位: mm)

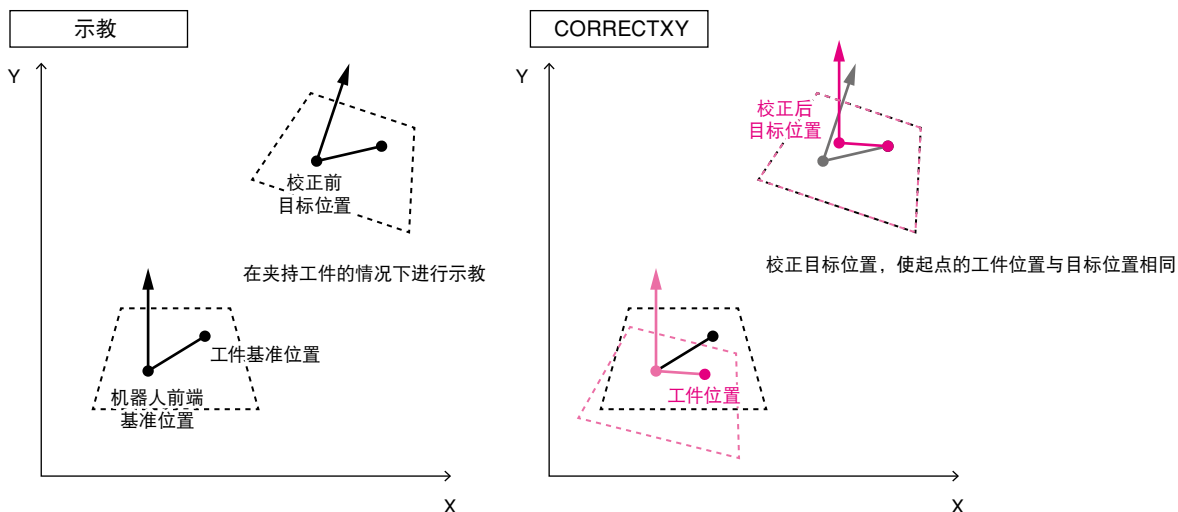
含义

就机器人前端所把持的工件, 获取对 XY 平面上的把持位置偏差进行校正后的目标位置。

=> 经<校正前目标位置>的 XYR 轴校正后, 获取目标位置点数据。

轴 (XYR 轴除外) 的位置为校正前目标位置点数据值。

- 事先, 须在夹持工件的情况下进行示教对<机器人前端基准点>、<工件基准点>、<校正前目标位置>。
- 可在联机命令中进行使用。



注意

- 所有坐标点数据均以毫米为单位。
- 该函数仅供使用机器人包括每个 XYR 轴。当用于其他机器人时, 会发生 "2.354 位置校正 坐标误差" 的警报

SAMPLE

明

```
P1 = CORRECTXY [1] (P2, P3, P4, P5)
```

.....将获取校正后目标位置赋值给 P1

机器人 1、机器人前端基准位置 P2、工件基准位置 P3、工件位置 P4、
校正前目标位置 P5

COS

计算余弦值

联机



8

函数

格式

COS (表达式)

值

内容

表达式

角度 (单位: 弧度)

含义 赋予 <表达式> 中值的余弦值。

SAMPLE

说明

A (0) =B *COS (C)将变量 C 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

CREWRK

创建以两点间差分为偏移的工件

联机



函数

形式	
CREWRK [机器人编号] (差分点, 基准点)	
值	范围
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
差分点	直角坐标系点位
基准点	直角坐标系点位 (省略时, 当前位置被使用)

- 含义

以基准点与差分点的各坐标的差分值为工件定义的偏移，作成工件数据。
基准点省略情况下，使用当前位置作为基准点。
差分点、基准点都不是直角坐标系形式的话，会发生错误。
- 参照

“WRKDEF”
RCX3 用户手册“工件定义”

MEMO

差分点、基准点中指定点的单位是脉冲时，会发生“6.205：坐标类型错误”。

SAMPLE	明
CHANGE H1	更改为机器人 1 的机械手 1
MOVE P, P1	机械手 1 的前端向 P1 移动
WRKDEF W1= CREWRK (P3)	以当前位置与 P3 的差分为偏移，定义工件 1
CHGWRK W1	更改为机器人 1 的工件 1
MOVE P, P2	工件 1 的前端向 P1 移动
HALT	

相关命令	CHGWRK, WRKDEF
------	----------------

CURTQST

获取相对于轴的定格扭矩值（定格电流值）当前扭矩值（当前电流值）的比率

联机



8

函数

格式	
CURTQST [机器人编号] (轴编号)	
值	范围
机器人编号	1 ~ 4（省略时，机器人 1 被指定。）
轴编号	1 ~ 6

含义 获取相对于轴的定格扭矩值（定格电流值）当前扭矩值（当前电流值）的比率（-1000 ~ 1000 [%]）。符号表示方向。

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

A

B

C

D

E

F

G

H

I

J

K

L

M

CURTRQ

获取相对于轴的最大扭矩值（最大电流值）当前扭矩值（当前电流值）的比率

联机



函数

格式	
CURTRQ [机器人编号] (轴编号)	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6

含义 获取轴的当前扭矩值 (-100 ～ 100)。该值表示相对于最大扭矩指令值的比率。
符号表示方向。

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

CUT

强制结束启动中的其他任务

联机



8

命令

格式		
CUT	Tn <程序名称> PGm	
标注	值	范围
m	程序编号	1 ~ 100
n	任务编号	1 ~ 16

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

MEMO

当指定并执行了未启动的任务(程序)时,将会发生错误。

SAMPLE	说明
' TASK1 ROUTINE	任务 1
*ST:	
MO (20) = 0	
START <SUB_PGM>, T2	将 SUB_PGM 程序登入到任务 No. 2 并启动
MOVE P, P0	
MOVE P, P1	
WAIT MO (20) = 1	
CUT T2	强行退出启动中的任务 2
GOTO *ST	
HALTALL	
' TASK2 ROUTINE	任务 2 (程序名称 : SUB_PGM)
*SUBTASK2:	
P100=JTOXY (WHERE)	
IF LOC3 (P100) >= 100.0 THEN	
MO (20) = 1	
ELSE	
DELAY 100	
ENDIF	
GOTO *SUBTASK2	
EXIT TASK	退出任务 2 (当前任务 : SUB_PGM)

相关命令 EXIT TASK, RESTART, START, SUSPEND

DATE\$

获取日期

联机



函数

格式

DATE\$

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

SAMPLE	说明
A\$=DATE\$	……以 yyyy/mm/dd 形式的字符串将日期赋值到 A
PRINT DATE\$	……以 yyyy/mm/dd 形式显示日期
HALT	

相关命令 TIMES

DECEL

设置 / 获取减速比率参数

联机



8

命令

格式

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

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6
表达式	1 ~ 100 (单位 : %)

含义 将轴的减速比率参数变更为 <表达式> 的值。

形式 1 为变更指定机器人的所有轴。

格式 2 仅更改 <轴编号> 中指定的轴。

MEMO

命令语句 : 当使用 ACCEL 时, 可以变更加速度参数。

函数

格式

DECEL [机器人编号] (轴编号)

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6

含义 赋予轴的减速比率参数值。

SAMPLE

说明

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

DEF FN

定义用户可使用的函数

联机

X

命令

格式	
DEF FN 名称	(形参, 形参 ...) = 函数的定义表达式
值	含义
名称	函数名称。包括 FN 在 16 个字符以内
形参	数值型或字符型的变量

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

 MEMO

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

SAMPLE	说明
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）

联机



8

函数

格式

DEGRAD (表达式)

值

内容

表达式

角度（单位：度）

含义 将<表达式>的值转换为弧度。



MEMO

若要将弧度转换为度，请使用 RADDEG。

SAMPLE

说明

A=COS (DEGRAD (30))将 30° 的余弦值赋值给变量 A

相关命令

ATN, COS, RADDEG, SIN, TAN

A

B

C

D

E

F

G

H

I

J

K

L

M

DELAY

在指定时间等待程序的执行

联机

×

命令

格式	
DELAY 表达式	
值	范围
表达式	0 ~ 3600000 (单位 : ms)

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

SAMPLE	说明
DELAY 3500 A =50	……等待 3,500ms (3.5 秒)
DELAY A*10	……等待 500ms (0.5 秒)

DI

获取并行端口的输入状态

联机



8

函数

形式		
1.LET 表达式 = DI m (b , . . . , b) 2.LET 表达式 = DI (mb , . . . , mb) 3.LET 表达式 = 输入输出名称		
标注	值	范围、含义
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7 (如果省略, 则将所有 8 个比特作为对象) 指定多个比特时, 从左边开始降序 (大→小) 记述
	输入输出名称	用户可为 DI 或 DO 的特定比特设置的任意名称 (注)仅可在支持软件上添加和编辑。无法通过手持编程器进行添加和编辑。

含义 表示并行输入信号的状态。
实际不存在输入端口时, 将输入 0。

SAMPLE	说明
A%=DI 2 ()	将 DI (27) 至 DI (20) 的输入状态赋值给变量 A%
A%=DI 5 (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)
A%=END_SENSOR	将输入输出名称 “END_SENSOR” 的输入状态赋值给变量 A% (“END_SENSOR” 为 ON 时, A%=1)

参照 第 3 章 “9.3 并行输入变量”

DIM

声明数组变量

联机

×

命令

格式		
DIM 数组指定 , 数组指定 , ...		
数组指定		
名称	% ! \$	(常量 , 常量 , 常量)
标注	值	范围、含义
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 [9.300：内存已满] 的错误。

SAMPLE	说明
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)

DIST

计算 2 个点之间的距离

联机



8

函数

格式

DIST (坐标点表达式 1, 坐标点表达式 2)

值

含义

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

说明

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

DO

向并行端口输出，或获取输出状态

联机



命令

格式		
1.LET	DOm (b, . . . ,b) = 表达式	
2.LET	DO (mb, . . . ,mb) = 表达式	
3.LET	输入输出名称 = 表达式	
标注	值	范围、含义
m	端口编号	0 ~ 7, 10 ~ 17,20 ~ 27
b	比特编号	0 ~ 7（如果省略，则将所有 8 个比特作为对象） 指定多个比特时，从左边开始降序（大→小）记述
	表达式	超出整数值（指定实数时，将小数点以下四舍五入后转换为整数） 赋值目标所需比特数的比特将被舍去。 （如果指定端口，则使用低位 8 比特。 如果在比特编号中指定的比特数为 1 ~ 8 比特，则使用低位 1 ~ 8 比特）
	输入输出名称	用户可为 DI 或 DO 的特定比特设置的任意名称 （注）仅可在支持软件上添加和编辑。无法通过手持编程器进行添加和编辑。

- 含义

将直接指定值输出至 DO 端口。
指定了实际不存在的 DO 端口时，将无任何输出。
- 参照

关于比特编号，请参阅第 3 章“10 指定比特”。



注意
无法输出至 DO0 () 与 DO1 ()。仅限参照。

SAMPLE	说明
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)
CHUCK=1	打开输入输出名称“CHUCK”（输出 1）

函数

格式		
1.LET 表达式 = DOm (b, . . . ,b)		
2.LET 表达式 = DO (mb, . . . ,mb)		
3.LET 表达式 = 输入输出名称		
标注	值	范围、含义
m	端口编号	0 ~ 7, 10 ~ 17,20 ~ 27
b	比特编号	0 ~ 7 (如果省略, 则将所有 8 个比特作为对象) 指定多个比特时, 从左边开始降序 (大→小) 记述
	表达式	超出整数值 (指定实数时, 将小数点以下四舍五入后转换为整数) 赋值目标所需比特数的比特将被舍去。 (如果指定端口, 则使用低位 8 比特。 如果在比特编号中指定的比特数为 1 ~ 8 比特, 则使用低位 1 ~ 8 比特)
	输入输出名称	用户可为 DI 或 DO 的特定比特设置的任意名称 (注) 仅可在支持软件上添加和编辑。无法通过手持编程器进行添加和编辑。

含义 参照并行端口信号的状态。

SAMPLE	说明
A%= DO2 ()	将从 DO (27) 到 DO (20) 的输出状态赋值给变量 A%
A%= DO0 (6, 5, 1)	将 DO (06), DO (05), DO (01) 的输出状态赋值给变量 A% (上述信号都变为 1 (ON) 时, A%=7)
A%=DO (37, 35, 27, 10)	将 DO (37), DO (35), DO (27), DO (10) 的输出状态赋值给变量 A% (上述信号的 DO (27) 以外都变为 1 (ON) 时, A%=13)
A%=CHUCK	将输入输出名称 “CHUCK” 的输出状态赋值给变量 A% (“CHUCK” 为 ON 时, A%=1)

相关命令	RESET, SET
------	------------

命令

格式	
DRIVE [机器人编号] (轴编号, 表达式), (轴编号, 表达式) ..., 选项, 选项 ...	
值	范围、含义
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6
表达式	马达位置 (毫米、度、脉冲) 或坐标点表达式

- 含义
- 执行轴为单位的绝对移动命令。

当附加轴进行绝对移动时, 将使用该命令。

 - 移动种类: 轴指定 PTP
 - 坐标点指定: 直接数值指定、坐标点定义指定
 - 选项: 相对速度指定、绝对速度指定、STOPON 条件指定、CONTON 条件指定、CONT 条件指定、XY 指定

移动种类

- 轴指定 PTP (Point to Point)

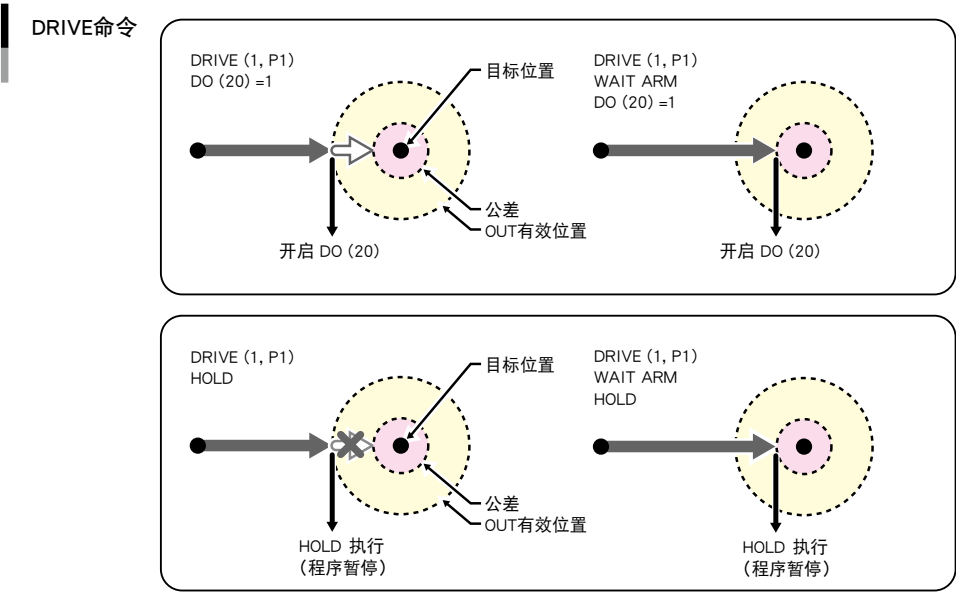
<轴编号> 中指定的所有轴动作完成后 (轴到达公差范围内以后), 开始移动, 并在进入 OUT 有效位置范围内后结束命令。指定多个轴时, 将同时到达。

DRIVE 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, 程序将停止并被复位。因此, 移动将停止。
HALTALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部停止, 任务 1 被复位, 其它任务结束。因此, 移动也会停止。
HOLD	进入 OUT 有效范围内后, 程序将暂停。因此, 移动将停止。
HOLDALL	当进入 OUT 有效位置范围以内时, 正在执行的程序将全部暂停。因此, 移动也会停止。
WAIT	进入 OUT 有效位置范围内后, 将执行 WAIT 命令。

轴到达公差范围内后要执行下一个命令时，请使用 "WAIT ARM" 语句。



SAMPLE	说明
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	说明
DRIVE (1, 10000)	机器人 1 中第 1 轴从当前位置向 10000 脉冲处移动。
DRIVE [2] (2, 90.00)	机器人 2 中第 2 轴从当前位置开始，向由 0 脉冲位置沿 + 方向旋转 90 度的位置进行移动（当为旋转轴时）。

●坐标点定义指定

在<表达式>中指定坐标点表达式。使用<轴编号>中指定轴的数据。
当坐标点表达式为毫米 / 度单位时，每个轴从 0 脉冲位置向脉冲换算位置移动。
但是，使用选项指定 XY 时，从坐标原点开始移动。

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

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

选项种类

- 相对速度指定
- 绝对速度指定
- STOPON 条件指定
- CONTON 条件指定
- CONT 指定

参照 "MOVE"

● XY 指定

格式

XY

含义 将多个指定轴移动至正交坐标中指定的位置。
指定的轴同时到达。
在 MOVE 语句中指定了所有可移动的轴时，将变为与 MOVE 语句相同的动作。



- 注意** 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	说明
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

DRIVEI

以轴为单位进行相对移动

联机



8

命令

格式

DRIVEI [机器人编号] (轴编号, 表达式), (轴编号, 表达式) ..., 选项, 选项 ...

值	范围、含义
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6
表达式	马达位置 (毫米、度、脉冲) 或坐标点表达式

- 含义** 直接执行组内以轴为单位的相对移动命令。
- 当对附加轴进行相对移动时, 将使用该移动命令。
- 移动种类(*) : 轴指定 PTP
 - 坐标点指定(*) : 直接数值指定、坐标点定义指定
 - 选项(*) : 相对速度指定、绝对速度指定、STOPON 条件指定、CONTON 条件指定、CONT 条件指定

参照 "DRIVE" "MOVE" 命令。
(样本命令, 请置换使用 [DRIVEI]。)

MEMO

中途停止 DRIVEI 语句并重新启动时, 可在控制器 "other parameters" 的 "MOVEI / DRIVEI start position" 中选择移动的目标位置。

参照 详细说明, 请参阅控制器的手册。

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

移动种类

● 轴指定 PTP (Point to Point)

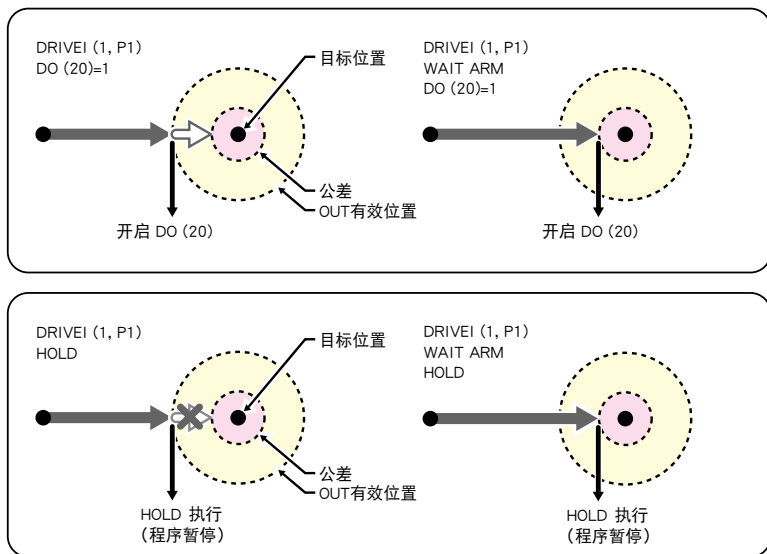
<轴编号>中指定的所有轴动作完成后(轴到达公差范围内以后), 开始移动, 并在进入 OUT 有效位置范围内后结束命令。指定多个轴时, 将同时到达。

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

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

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

DRIVEI 命令
WAIT ARM 语句



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	说明
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）

●坐标点指定

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 脉冲位位置向脉冲换算位置移动。

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

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

选项种类

● 速度指定

格式

- 1.SPEED = 表达式
- 2.S = 表达式

值

范围

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 DRIVEI 语句有效。



补充

虽然规定了最高速度，但并不保证按照指定速度移动。

SAMPLE

说明

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

范围

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 DRIVEI 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。



补充

SPEED 选项与 DSPEED 选项无法并用。

SAMPLE

说明

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

● STOPON 条件指定

格式
STOPON 条件表达式

含义 移动中<条件表达式>不成立时，移动将停止。
此时的停止动作为**减速停止**，在条件成立后将产生减速移动距离。
移动开始时如果条件表达式成立，将不移动，直接结束命令。
仅程序执行时有效。

 MEMO

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

SAMPLE	说明
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

END SELECT

结束 SELECT CASE 语句

联机

×

命令

格式

SELECT CASE 表达式

CASE 表达式的列表 1
命令区 1
CASE 表达式的列表 2
命令区 2
:
CASE ELSE
命令区 n
END SELECT

含义

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

SAMPLE	说明
WHILE 1 SELECT CASE DI3(7,6,5,4,3,2,1,0) CASE &B00000001 MOVE P,P1 CASE &B00000010 MOVE P,P2 CASE &B00000011 MOVE P,P3 CASE ELSE MOVE P,P0 END SELECT WEND	……在 DI3（8 位）的输入状态下执行各命令区 ……DI3（0）为 ON 时 ……DI3（1）为 ON 时 ……DI3（0）和 DI3（1）为 ON 时 ……上述以外时 ……退出 SELECT CASE 语句

相关命令	SELECT CASE
------	-------------

END SUB

结束子过程的定义

联机

×

8

命令

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“SUB ~ END SUB”。

SAMPLE	说明
I=1	
CALL *TEST	调用子过程 *TEST
PRINT I	
HALT	
' SUB ROUTINE: TEST	 (注释：将子过程命名为“TEST”)
SUB *TEST	定义子过程 *TEST
I=50	
END SUB	退出子过程的定义

相关命令 CALL, EXIT SUB, SUB ~ END SUB

A

B

C

D

E

F

G

H

I

J

K

L

M

ERR / ERL

获取错误编码 / 错误发生行编号

联机



函数

格式	
ERR (任务编号)	
ERL (任务编号)	
值	范围
任务编号	1 ~ 16（省略时，指定执行中的任务。）

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

SAMPLE	说明
IF ERR (1) <> &H000600CC THEN HALT	……倘出现“无点位”错误，则会停止程序并重置
IF ERL =20 THEN RESUME NEXT	……倘发生错误行为 20,则跳转至发生错误行之下一行，并继续执行

相关命令	ON ERROR GOTO, RESUME
------	-----------------------

ETHSTS

获取以太网端口的状态

联机



8

函数

格式

ETHSTS

含义

获取以太网端口的状态。
根据状态，获得以下值。

-2	以太网端口未打开
-1	未连接 LAN 电缆
0	未建立连接
1	已建立连接
2	已建立连接 · 将数据存储至接收缓冲区

SAMPLE

说明

A=ETHSTS 将以太网端口的状态赋给变量 A

A

B

C

D

E

F

G

H

I

J

K

L

M

EXIT FOR

强制结束 FOR ～ NEXT 语句的循环

联机

×

命令

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 [5.212：堆栈溢出] 等错误。

SAMPLE	说明
<pre>*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.....DI (20)=0（假）时，跳出整个 FOR ～ NEXT 循环 NEXT A% GOTO *ST HALT</pre>	

相关命令 FOR, NEXT

EXIT SUB

强制结束使用 SUB ~ END 定义的子过程

联机

×

8

命令

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[5.212：堆栈溢出]等错误。

SAMPLE	说明
' MAIN ROUTINE	
CALL *SORT2 (REF X%,REF Y%)	调用子过程 *SORT2
HALT	
' SUB ROUTINE: SORT	将 *SORT2 定义为子过程
SUB *SORT2 (X%, Y%)	
IF X% >=Y% THEN EXIT SUB	如果 X 为 Y 以上，则退出 SUB ~ END SUB 循环
TMP%=Y%	
Y%=X%	
X%=TMP%	
END SUB	

相关命令 CALL, SUB ~ END SUB, END SUB

A

B

C

D

E

F

G

H

I

J

K

L

M

EXIT TASK

结束当前正在执行的任务

联机

X

命令

格式

EXIT TASK

含义 结束当前正在执行的任务。

SAMPLE	说明
' TASK1 ROUTINE *ST: MO (20) =0 START <SUB_PGM>, T2 MOVE P, P0, P1 WAIT MO (20) =1 GOTO *ST HALTALL	…将“SUB_PGM”程序登入到任务 No. 2 并启动
程序名称 : SUB_PGM ' TASK2 ROUTINE *SUBTASK2: P100=JTOXY (WHERE) IF LOC3 (P100) >=100.000 THEN MO (20) =1 EXIT TASK ENDIF DELAY 100 GOTO *SUBTASK2 EXIT TASK	…退出任务 2 (当前任务 : SUB_PGM) …跳至 *SUBTASK 2 …退出任务 2 (当前任务 : SUB_PGM)

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

FOR ~ NEXT

变量超过指定值为止将循环

联机

×

8

命令

格式

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

- 含义** 直接将<控制变量>的值按照<步骤>的值逐步变化，反复执行 FOR 语句的下一命令至 NEXT 语句，直至<开始值>超过<结束值>为止。
- 省略了“STEP <步骤>”时的步骤值为 1。
- <步骤>的值也可为负数。
- <控制变量>为数值型的<简单变量>或<数组变量>。
- FOR 语句与 NEXT 语句必须成对使用。

SAMPLE

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

使用 FOR~NEXT 语句，可以简化程序。

SAMPLE（使用 FOR~NEXT 语句的记述）

```
FOR A=1 TO 5 .....FOR~NEXT 之间重复 5 次  
    MOVE P,P[A],A3=0.00 MOVE 语句的移动目标点位号根据变量值可变化。  
NEXT A  
MOVE P,P10,A3=0.00  
DO(20)=1
```

不使用 FOR~NEXT 语句时，记述即使简单，移动目标一旦变多，程序的行数就会增加。

SAMPLE（不使用 FOR~NEXT 语句的记述）

```
MOVE P,P1,A3=0.00  
MOVE P,P2,A3=0.00  
MOVE P,P3,A3=0.00  
MOVE P,P4,A3=0.00  
MOVE P,P5,A3=0.00  
MOVE P,P10,A3=0.00  
DO(20)=1
```

MEMO

在进行 FOR~NEXT 间的处理时，用 GOTO 等命令无法跳出 FOR~NEXT 循环。

若要跳出 FOR~NEXT 循环时，请使用“EXIT FOR”。

相关命令 EXIT FOR

GEPSTS

获取通用以太网端口的状态

联机



函数

格式	
GEPSTS(通用以太网端口编号)	
值	范围
通用以太网端口编号	0 ~ 7

含义 获取通用以太网端口的状态。
根据状态，获得以下值。

-2	通用以太网端口未打开
-1	未连接 LAN 电缆
0	未建立连接
1	已建立连接
2	已建立连接 · 将数据存储至接收缓冲区

SAMPLE	说明
OPEN GP1	………打开通用以太网端口编号 1 所指定的通信端口
IF GEPSTS(1) > 0 THEN	………确认连接是否建立
SEND “ABC” TO GP1	………发送字符串 “ABC”
IF GEPSTS(1)=2 THEN	………确认接收缓冲区是否存储有数据
SEND GP1 TO RET\$	………从通用以太网端口编号 1 接收数据，将接收的数据保存至变量 RET\$
ENDIF	………关闭通用以太网端口编号 1 所指定的通信端口
ENDIF	
CLOSE GP1	
HALT	

相关命令	OPEN, CLOSE, SEND, SETGEP
------	---------------------------

命令

格式

```
GOSUB  标签          *GOSUB 也可写成 GO  SUB
:
标签:
:
RETURN
```

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

MEMO

- GOSUB 语句最多可连续使用 120 次。但是，根据 FOR 语句或 CALL 语句等使用堆栈命令的使用状态的不同，次数将减少。
- 使用 GOSUB 语句跳转至子例程时，请务必使用 RETURN 语句结束子例程。通过 GOTO 等语句从子例程跳转时，可能会发生[5.212：堆栈溢出]等错误。

SAMPLE

说明

```
*MAIN:
MOVE P,P1,A3=0.00
GOSUB *SUB_DO          .....跳至标签 “*SUB_DO”
MOVE P,P2,A3=0.00      .....子例程 *SUB_DO 到达 RETURN 后，从该行开始执行
HALT

*SUB_DO:                .....将 *SUB_DO 定义为子例程的标签
DO (20)=1
DELAY 500
DO (20)=0
RETURN                  .....从 GOSUB 语句的下一行开始执行
```

相关命令

RETURN

GOTO

无条件跳转至指定的行

联机

X

命令

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。

MEMO

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。
关于详细内容请参阅 1 章“LABEL 语句”。

SAMPLE	说明
*TOP: MOVE P,P1 DELAY 1000 MOVE P,P2 DELAY 1000 GOTO *TOP	将 *TOP 定义为标签 跳至标签 “*TOP”

HALT

停止程序并进行复位

联机

×

8

命令

格式		
HALT	表达式 字符串	

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

MEMO

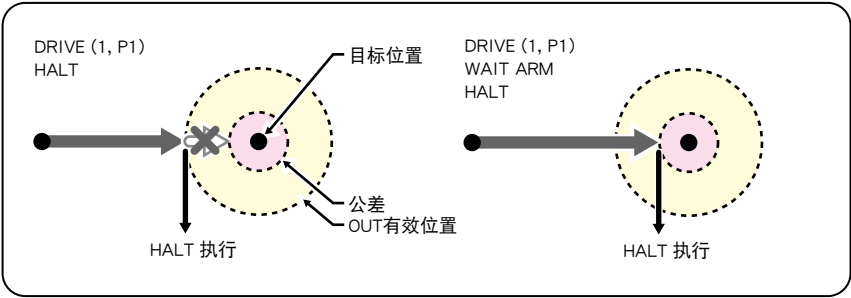
- HALT 仅对执行过的任务起作用。在其它任务中正在执行的程序将继续执行。
- HALT 仅对已执行的任务有效。在其他任务中执行的程序将继续执行。

SAMPLE	说明
' MAIN ROUTINE	
*ST:	
MOVE P, P0, P1	
IF DI (20) = 1 THEN	如果 DI (20) 为 ON
GOTO *FIN	跳至标签 “*FIN”
ENDIF	
GOTO *ST	
*FIN:	将 *FIN 定义为标签
HALT “PROGRAM FIN”	停止程序并复位（显示 “PROGRAM FIN”）

在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。因此，如果在 PTP 移动命令的后面紧接着 HALT 命令时，在轴到达目标位置的公差范围内之前，将执行 HALT 命令。同样地，MOVE 命令的插补移动由于在移动开始后立即执行下一个命令，如果在 MOVE 的插补移动命令后面紧接着 HALT 命令，则将在移动开始后立即执行 HALT 命令。

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

HALT命令



HALTALL

停止并复位所有的程序

联机

×

命令

格式

HALTALL

表达式
字符串

含义

直接停止并复位所有的程序。

此时，动态变量、数组变量和输出变量均被复位。

输出变量 (DO/SO/MO/LO/TO/SOW) 仅在以下条件成立时复位。

- 1) 参数“程序复位时的 DO 输出”为“复位”时
- 2) 执行逻辑控制程序时，“逻辑控制执行标志”为“允许复位”时

当执行 HALTALL 后重新启动程序时，将从主程序或者任务 1 中最后执行程序的最前端开始执行。

当对 <表达式> 或 <字符串> 进行记述时，编程盒的画面上将会显示 <表达式> 或 <字符串> 的内容。



MEMO

- HALTALL 是停止多任务中正在运行的所有程序。
复位处理后，主程序或当前程序会自动登录到任务 1, 其他任务全部清除。
- 执行 HALTALL 后所有的通用输出和变量被复位。
但是，将参数“I/O”内的“程序复位时的 DO 输出”的设定变为“保持”时，通用输出复位将不执行。

SAMPLE

说明

```
MOVE P, P1
DELAY 1000
HALTALL
```

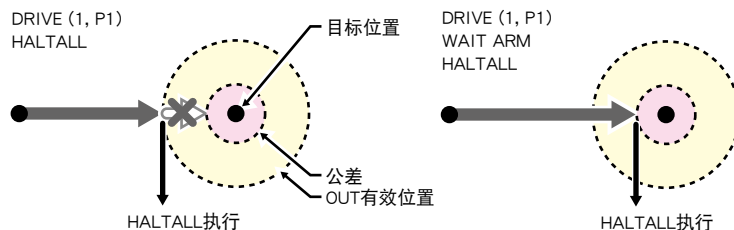
……………停止并复位所有的程序

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

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

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

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



HAND

机械手定义

联机



8

命令

形式

定义语句：

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

选择语句：

CHANGE [机器人编号] Hn

标注	值	范围
	机器人编号	1 ~ 4
n	机械手编号	0 ~ 31
R	在 R 轴上安装机械手	

含义 HAND 语句只可用于机械手的定义。实际进行机械手数据的切换时，使用 CHANGE 语句。
关于 CHANGE 语句，请参阅“CHANGE”。
指定 R 后，变为从 R 轴旋转中心偏移的机械手。

MEMO

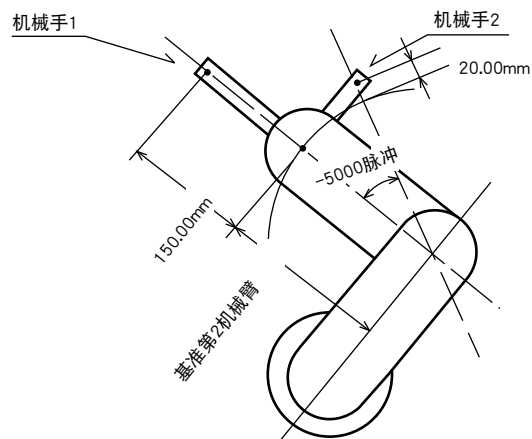
- 如果在执行机械手定义语句过程中关闭了电源，将发生[9.707：机械手校验和错误]错误。
- 如果在 CHANGE 语句中指定了定义其他机器人的机械手数据，则出现[6.258：机器人编号错误]的提示。

48.1 水平多关节型机器人时

48.1.1 未指定<第 4 参数>的 R 时

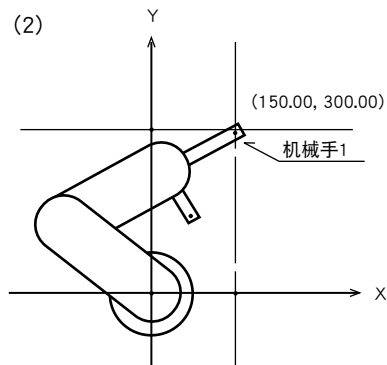
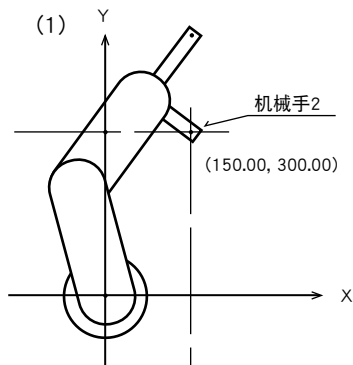
成为固定在第 2 机械臂前端的机械手(参照下图)。

- <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]



SAMPLE	说明						
HAND H1=	0	150.000	0.000				
HAND H2=	-5000	20.000	0.000				
P1=	150.000	300.000	0.000	0.000	0.000	0.000	
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							

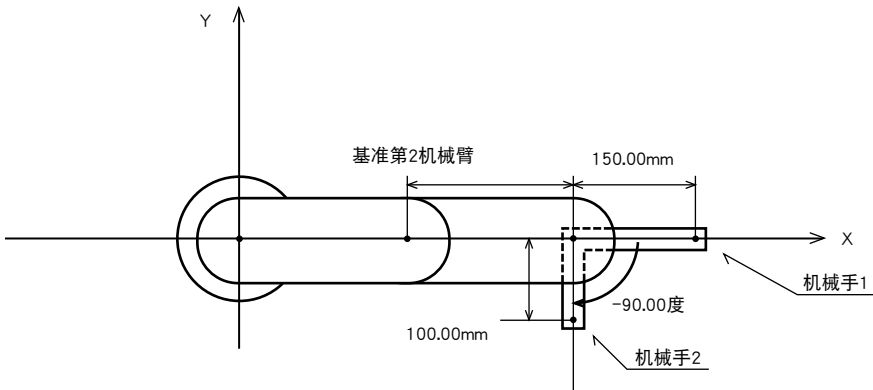
SAMPLE: HAND



48.1.2 指定了<第 4 参数>的 R 时

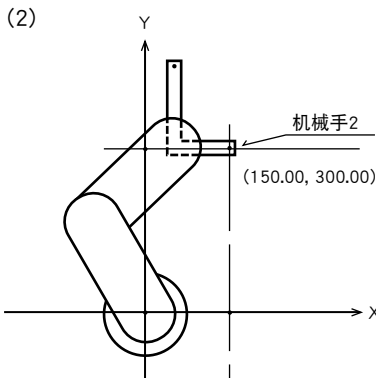
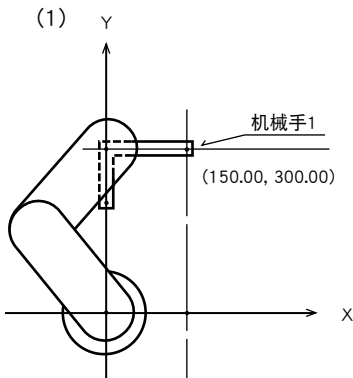
R 轴为伺服时成为从 R 轴旋转中心偏移的机械手 (参照下图)。

- <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]



SAMPLE	说明					
HAND H1=	0.000	150.000	0.000	R		
HAND H2=	-90.000	100.000	0.000	R		
P1=	150.000	300.000	0.000	0.000	0.000	0.000
CHANGE H1	更改为机械手 1					
MOVE P,P1	机械手 1 的前端向 P1 移动 (1)					
CHANGE H2	更改为机械手 2					
MOVE P,P1	机械手 2 的前端向 P1 移动 (2)					
HALT						

SAMPLE: HAND

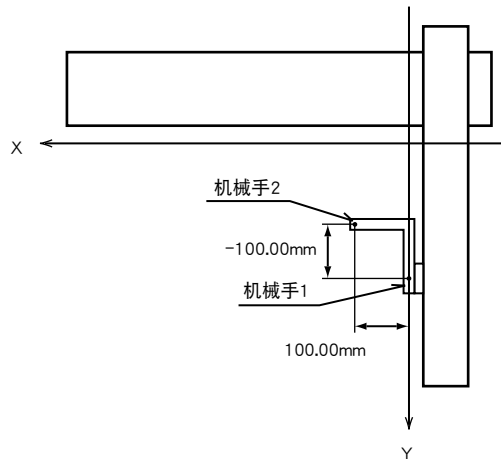


48.2 正交机器人时

48.2.1 未指定<第4参数>的R时

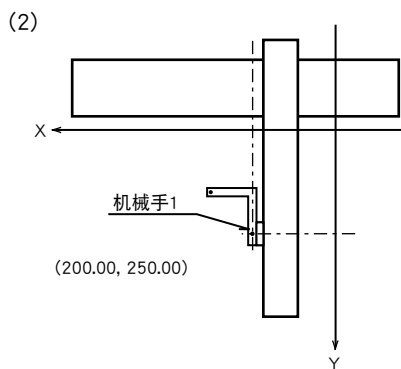
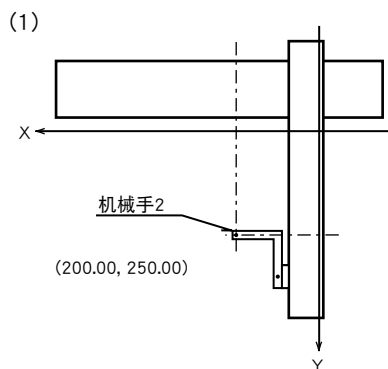
成为固定在第2机械臂前端的机械手(参照下图)。

<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]



SAMPLE	说明					
HAND H1=	0.000	0.000	0.000			
HAND H2=	100.000	-100.000	-100.000			
P1=	200.000	250.000	0.000	0.000	0.000	0.000
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						

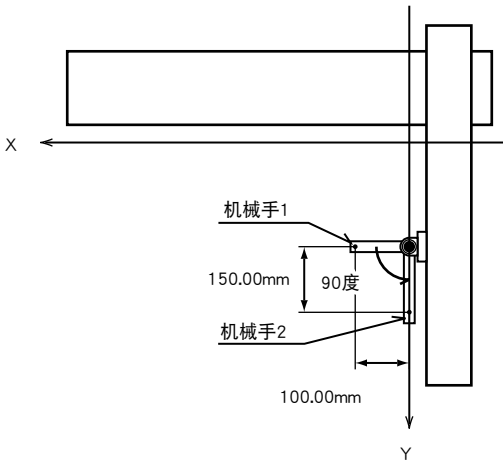
SAMPLE: HAND



48.2.2 指定了<第 4 参数>的 R 时

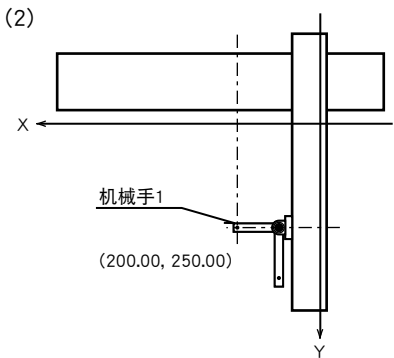
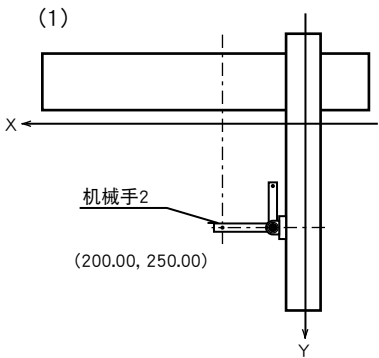
R 轴为伺服时成为从 R 轴旋转中心偏移的机械手 (参照下图)。

- <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]



SAMPLE	说明					
HAND H1=	0.000	100.000	0.000	R		
HAND H2=	90.000	150.000	0.000	R		
P1=	200.000	250.000	0.000	0.000	0.000	0.000
CHANGE H2	更改为机械手 1					
MOVE P, P1	机械手 1 的前端向 P1 移动 (1)					
CHANGE H1	更改为机械手 2					
MOVE P, P1	机械手 2 的前端向 P1 移动 (2)					
HALT						

SAMPLE: HAND



HOLD

暂停程序

联机

X

命令

格式

HOLD

表达式
字符串

含义

暂停程序。重新启动时，从 HOLD 语句的下一行开始执行。

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



MEMO

HOLD 仅对执行过的任务起作用。在其它任务中正在执行的程序将会继续执行。

SAMPLE

说明

' MAIN ROUTINE

*ST:

MOVE P, P0, P1

IF DI (20) =1 THEN

HOLD "PROGRAM STOP"

ENDIF

GOTO *ST

HALT

..... 如果 DI (20) 为 ON

..... 暂停执行对象的任务程序（显示“PROGRAM STOP”）

..... 重新启动时，从该行开始执行

..... 停止程序并复位

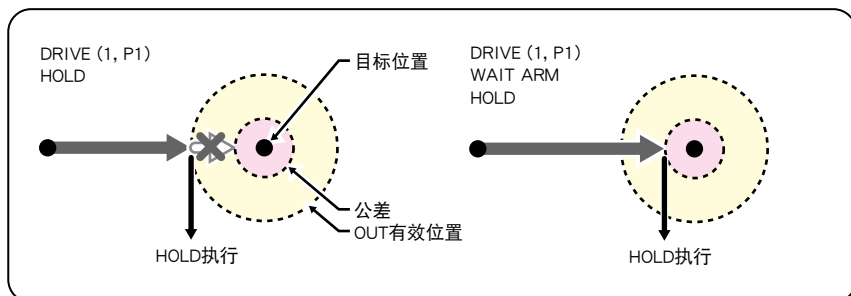
在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。

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

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

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

HOLD命令



HOLDALL

暂停所有的程序

联机

×

8

命令

格式		
HOLDALL	表达式 字符串	

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

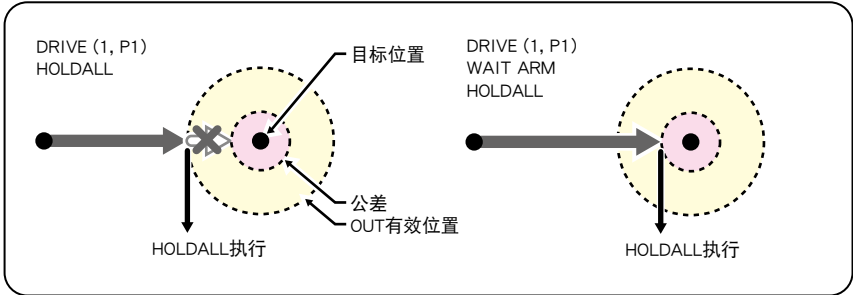
SAMPLE	说明
' MAIN ROUTINE *ST: MOVE P,P0,P1 IF DI (20)=1 THEN HOLDALL "PROGRAM STOP" ENDIF GOTO *ST HALT	 如果 DI (20) 为 ON 暂停所有的程序（显示“PROGRAM STOP”） 停止程序并复位

在进行 MOVE、DRIVE 移动命令的 PTP 移动过程中，如果进入了 OUT 有效位置范围内，将执行下一行的命令。因此，如果在 PTP 移动命令的后面紧接着 HOLDALL 命令时，在轴到达目标位置的公差范围内之前，将执行 HOLDALL 命令。

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

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

HOLDALL命令



IF

评估条件表达式的值，按照条件执行命令语句。

联机

✕

51.1 简单 IF 语句

命令

格式				
IF 条件表达式	THEN	标签 1 命令语句 1	ELSE	标签 2 命令语句 2
值		范围		
条件表达式		输入输出信号或变量		

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

<条件表达式>中指定的条件不成立时，进行以下动作。

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

 MEMO

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

SAMPLE	说明
IF DI(20)=1 THEN *MAIN1 ELSE *MAIN2 : *MAIN1: DO(20)=1 : *MAIN2: DO(21)=1 :	如果 DI(20) 为 ON “真”， 跳至标签 “*MAIN1” 如果 DI(20) 为 OFF “假”， 跳至标签 “*MAIN2”

51.2 区块 IF 语句

命令

形式	
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	
值	范围
条件表达式	输入输出信号、变量 等

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

SAMPLE 1 (区块 IF 语句 1)	说明
MOVE P, P1, A3=0.00 IF DI (20)=1 THEN DO (20)=1 MOVE P, P10, A3=0.00 DO (20)=0 ELSE DO (21)=1 MOVE P, P11, A3=0.00 DO (21)=0 ENDIF MOVE P, P20, A3=0.00 :	DI (20) 为 ON 时，执行“真”的命令区“真”的命令区从这里开始“真”的命令区至此结束DO (20) 为 OFF 时，执行“假”的命令区“假”的命令区从这里开始“假”的命令区至此结束IF 语句结束定义执行 IF 语句后，跳至该行
SAMPLE 2 (区块 IF 语句 2)	说明
MOVE P, P1, A3=0.00 IF DI (20)=1 THEN DO (20)=1 MOVE P, P10, A3=0.00 ENDIF DO (20)=0 :	DI (20) 为 ON 时，执行“真”的命令区“真”的命令区从这里开始“真”的命令区至此结束IF 语句结束定义执行 IF 语句后，跳至该行

命令

形式					
INPUT	提示语句	;	变数 坐标点变量 位移变量	,	变量 坐标点变量 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 (")。
- 例: ABC : A\$ 的内容变为 ABC
 ABC : A\$ 的内容变为 ABC (即使在 ABC 的前后输入空格, 也不会反映在内容上)
 " ABC " : A\$ 的内容变为 ABC (ABC 的前后有空格)
 ABC,XYZ : 如果 A\$ 的内容变为 ABC 后, 执行 INPUT 语句, 则输入 XYZ
 "ABC,XYZ" : A\$ 的内容变为 ABC,XYZ
 ""ABC"" : A\$ 的内容变为 "ABC"
7. 输入了 ESC 键时, 此命令将被跳过。



MEMO

- 当变量和赋值类型不同时, 将显示指定的信息, 等待再次输入。
- 指定变量为字符型变量时, 即使不在字符串外面加上双引号 (") , 也可输入字母数字。
- 使用 INPUT 语句时, 从控制器参数的 "INPUT / PRINT 正在使用的频道" 所指定的频道, 向变量赋值。

SAMPLE	说明
INPUT A	输入的字符串转换为实数后, 被赋给变量 A!
INPUT "INPUT POINT NUMBER";A1	手持编程器等显示 INPUT POINT NUMBER 的提示信息, 输入的字符串转换为实数后, 被赋给变量 A1!
INPUT "INPUT STRING",B\$(0),B\$(1)	如果手持编程器等显示 INPUT STRING 的提示信息, 输入的字符串含有逗号, 则其第 1 个被赋给变量数组 B\$ 的第 0 个要素、第 2 个被赋给 B\$(1)
INPUT P100	将输入的字符串赋给点位 100
HALT	

INROFST

设置 / 获取水平多关节 R 轴惯性力矩偏移量参数

联机



8

命令

格式	
INROFST [机器人编号] 表达式	
值	范围
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
表达式	水平多关节 R 轴惯性力矩偏移量参数的值 1 ~ 9999999 (单位 : 0.001mm)

含义 将指定机器人的水平多关节 R 轴惯性力矩偏移量参数变更为 <表达式> 的值。

SAMPLE	说明
INROFST 10	将机器人 1 的水平多关节 R 轴惯性力矩偏移量参数变更为 10
INROFST[2] 20	将机器人 2 的水平多关节 R 轴惯性力矩偏移量参数变更为 20

函数

格式	
INROFST [机器人编号]	
值	范围
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)

含义 获取指定机器人的水平多关节 R 轴惯性力矩偏移量参数值。

SAMPLE	说明
A% = INROFST	将机器人 1 的水平多关节 R 轴惯性力矩偏移量参数赋给变量 A%
A% = INROFST[2]	将机器人 2 的水平多关节 R 轴惯性力矩偏移量参数赋给变量 A%

INT

舍去小数点以下部分

联机



函数

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	说明
A=INT （A （0）） B=INT （-1． 233）	……………B 被赋予 -2

JTOXY

轴单位制转换（脉冲→毫米）

联机



8

函数

格式

JTOXY [机器人编号] (坐标点表达式)

值

范围

机器人编号 1～4（省略时，机器人 1 被指定。）

含义

将<坐标点表达式>所表示的关节坐标数据（单位：脉冲）转换为<机器人编号>指定机器人的正交坐标数据（单位：mm、度）。

SAMPLE

说明

P10=JTOXY (WHERE) ………将当前位置数据转换为正交坐标数据

相关命令

XYTOJ

A

B

C

D

E

F

G

H

I

J

K

L

M

LEFT\$

从左端抽出字符串

联机



函数

格式	
LEFT\$ （字符串表达式，表达式）	
值	范围
表达式	0 ～ 255

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 ～ 255 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	说明
B\$=LEFT\$ (A\$, 4)	……从 A\$ 左侧将 4 个字符赋给 B\$。

相关命令	MID\$, RIGHT\$
------	----------------

LEFTY

将水平多关节机器人的手系统设置为左手系统

联机



8

命令

形式

LEFTY [机器人编号]

值

范围

机器人编号 1 ～ 4（省略时，机器人 1 被指定。）

含义

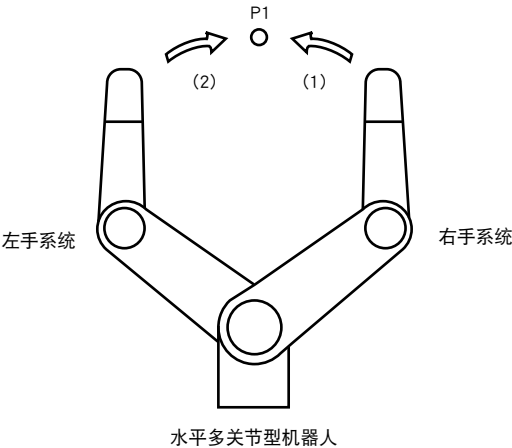
选择通过左手系统将<机器人编号>指定机器人移动到正交坐标系所指定的坐标点。
即使执行了命令**仅选择也无法使机器人移动**。轴在进行动作时,请等待动作完成(被定位至公差内)再执行。
此命令仅对水平多关节机器人有效。

SAMPLE

说明

RIGHTY	将机器人 1 的手系统设置为右手系统
MOVE P, P1	 (1)
LEFTY	将机器人 1 的手系统设置为左手系统
MOVE P, P1	 (2)
RIGHTY	将机器人 1 的手系统设置为右手系统
HALT	

SAMPLE : LEFTY / RIGHTY



相关命令

RIGHTY

LEN

获取字符串的长度

联机



函数

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	说明
A\$= "YAMAHA"	
B\$= "YAMAHA 发动机"	
C\$="YAMAHA CO., LTD."	
PRINT LEN(A\$)	显示 "6"
PRINT LEN(B\$)	显示 "12"
PRINT LEN(C\$)	显示 "16"

LET

对变量赋值

联机

✓

8

命令

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
[9.702：点位校验和错误]与[9.706：位移校验和错误]等存储器相关错误。

59.1 算术赋值语句

格式

LET

整数变量
实数变量
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
串行单字输出变量
串行双字输出变量

= 表达式

值

内容

表达式

变量（字符串·点位·位移变量除外）、函数、数值

含义

将<表达式>的值赋值给左边的变量。

SAMPLE

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

参照

关于赋值语句及各输出变量，请参阅第7章“5.1 赋值语句”“5.2 输入输出变量”

59.2 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。

59.3 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

说明

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

59.4 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<位移变量>。
<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	说明
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

LO

输出机械臂锁定，或获取输出状态

联机



命令

格式

1. LET LOM(b,b) = 表达式

2. LET LO(mb, . . . ,mb) = 表达式

标注	值	范围
m	端口编号	0、1
b	比特编号	0 ~ 7（如果省略，则将所有 8 个比特作为对象） 指定多个比特时，从左边开始降序（大→小）记述
	表达式	超出整数值（指定实数时，将小数点以下四舍五入后转换为整数） 赋值目标所需比特数的比特将被舍去。 （如果指定端口，则使用低位 8 比特。 如果在比特编号中指定的比特数为 1 ~ 8 比特,则使用低位 1 ~ 8 比特）

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) 分别对应第 1 轴 ~ 8 轴、LO(10) ~ LO(17) 分别对应第 9 轴 ~ 16 轴。
分配了比特的轴变为机械臂锁定 ON 状态，轴的移动被禁止。

参照

关于比特编号，请参阅第 3 章“10 指定比特”

MEMO

对开始移动的轴有效。

SAMPLE	说明
LO0 () =&B00001010	禁止第 2 轴、第 4 轴的移动
LO0 (2,1) =&B10	禁止第 3 轴移动、允许第 2 轴移动

函数

格式		
1. LET 表达式 = LOm(b,b)		
2. LET 表达式 = LO(mb, . . . ,mb)		
标注	值	范围
m	端口编号	0、1
b	比特编号	0 ~ 7（如果省略，则将所有 8 个比特作为对象） 指定多个比特时，从左边开始降序（大→小）记述

含义

获取指定 LO 端口的输出状态。
LO(00) ~ LO(07) 分别对应第 1 轴 ~ 8 轴、LO(10) ~ LO(17) 分别对应第 9 轴 ~ 16 轴。
对应 ON 状态比特的轴变为机械臂锁定开启状态，轴的移动被禁止。

SAMPLE	说明
A%=LO0()	将从 LO (07) 到 LO (00) 的输出状态赋值给变量 A%
A%=LO0(6, 5, 1)	将 LO (06) , LO (05) , LO (01) 的输出状态赋值给变量 A% (上述信号都变为 1 (ON) 时, A%=7)
A%=LO(17,15,00)	将 LO (17) , LO (15) , LO (00) 的输出状态赋值给变量 A% (上述信号的 LO (15) 以外都变为 1 (ON) 时, A%=5)

相关命令	RESET, SET
------	------------

LOCx

设定 / 获取坐标点数据为轴单位或者位移数据和工件数据为要素单位

联机



命令

格式		
1. LOCx(坐标点表达式) = 表达式 2. LOCx(位移表达式) = 表达式 3. LOCx(Wn) = 表达式		
	值	范围
格式 1	x	1 ~ 6 (指定轴) F (手系统标志指定), F1 (第 1 机械臂转数信息), F2 (第 2 机械臂转数信息)
格式 2	x	1 ~ 4 (指定元素)
格式 3	x	1 ~ 4 (指定元素)
	n	0 ~ 39 (工件编号)
表达式		· 指定轴或指定元素时: 坐标值、变量 · 指定手系统标志时: 1 (右手系统)、2 (左手系统)、0 (无指定) · 第 1、第 2 机械臂转数信息 (*1) 时: 0、1、-1

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

- 含义
- 格式 1: 变更坐标点数据指定轴的数值、手系统标志、第 1 或者 第 2 机械臂转数信息。
格式 2: 变更位移坐标数据的指定元素值。
格式 3: 变更工件数据的指定元素值。



补充

当更改工件数据时, 执行将处于等待状态, 直到机器人的动作进入公差范围内。
(对于传送带跟踪期间的 LOCx, 请参阅“跟踪系统”CTMOVE。)



MEMO

要更改值的坐标点, 必须事先录入。要更改未录入 (坐标值或元素数据为空白) 的坐标点、位移坐标、工件数据的值时, 将出错。

函数

格式		
1. LOCx(坐标点表达式) 2. LOCx(位移表达式) 3. LOCx(Wn)		
	值	范围
格式 1	x	1 ~ 6 (指定轴) F (手系统标志指定), F1 (第 1 机械臂转数信息), F2 (第 2 机械臂转数信息)
格式 2	x	1 ~ 4 (指定元素)
格式 3	x	1 ~ 4 (指定元素)
	n	0 ~ 39 (工件编号)
	表达式	指定轴或指定元素时: 坐标值、变量

含义

格式 1: 根据坐标点数据获取指定轴的数值、手系统标志、第 1 或者 第 2 机械臂转数信息。
格式 2: 根据位移坐标数据获取指定元素的值。
格式 3: 根据工件数据获取指定元素的值。

SAMPLE	说明
LOC1 (P10) =A (1)	将 P10 的第 1 轴数据变更为数组 A (1) 的值。
LOC2 (S1) =B	将 S1 的第 2 元素 (Y 方向) 数据变更为 B 的值。
LOC3 (W1) =C	将 W1 的第 3 元素 (Z 方向) 数据变更为 C 的值。
A (1) =LOC1 (P10)	将 P10 的第 1 轴数据赋给数组 A (1)。
B (2) =LOC1 (S1)	将 S1 的第 1 元素 (X 方向) 数据赋给数组 B (2)。
C (3) =LOC1 (W1)	将 W1 的第 1 元素 (X 方向) 数据赋给数组 C (3)。

相关命令	坐标点变量、位移变量、工件数据
------	-----------------

LSHIFT

比特的左位移

联机



函数

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

说明

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

MCHREF

获取机器参照（传感方式或撞块方式的轴）

联机



8

函数

格式	
MCHREF[机器人编号](轴编号)	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6

含义

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

SAMPLE	说明
A=MCHREF (1)	……………将机器人 1 中第 1 轴的原点归复动作机台参考赋给变量 A。

A

B

C

D

E

F

G

H

I

J

K

L

M

MID\$

从指定位置获取字符串

联机



函数

格式

MID\$(字符串表达式 , 表达式 1 , 表达式 2)

值

范围

表达式 1

1 ~ 255

表达式 2

0 ~ 255

含义

从<字符串表达式>指定的字符串中抽出任意长度的字符串。

从字符串中抽出<表达式 1>的第几位字符开始、<表达式 2>的字符数。

- 当<表达式 1>、<表达式 2>的值超出范围以外时，将出错。
- 省略了<表达式 2>或从字符串的<表达式 1>的第几位字符开始右侧所有的字符数均小于<表达式 2>时，将从字符串的<表达式 1>的第几位字符开始抽出右侧所有字符。
- 如果<表达式 1>大于字符串的字符数，MID\$ 的值将变为空字符串。

SAMPLE

说明

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\$

命令

形式		
1. LET MOm(b,b) = 表达式		
2. LET MO(mb,,mb) = 表达式		
标注	值	范围、含义
m	端口编号	2 ~ 7, 10 ~ 17, 20 ~ 27, 30 ~ 37
b	比特编号	0 ~ 7 (如果省略, 则将所有 8 个比特作为对象) 指定多个比特时, 从左边开始降序 (大→小) 记述
	表达式	超出整数值 (指定实数时, 将小数点以下四舍五入后转换为整数) 赋值目标所需位数的位将被舍去。 (如果指定端口, 则使用低 8 位。 如果在比特编号中指定的位数为 1 ~ 8 位, 则使用低 1 ~ 8 位)

- 参照 关于比特编号, 请参阅第 3 章“10 指定比特”。
- 含义 此命令语句将指定值输出至 MO 端口。
端口 30 ~ 37 保持各轴的原点传感器状态以及轴的保持状态, 因此仅可查看无法输出。(端口 32、33、36、37 为系统保留)

端口 30、31、34、35 的输出

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

MEMO

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

SAMPLE	说明
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

函数

形式

1. LET 表达式 = MOm(b,b)
2. LET 表达式 = MO(mb, . . . ,mb)

标注	值	范围
m	端口编号	2 ~ 7, 10 ~ 17, 20 ~ 27, 30 ~ 37
b	比特编号	0 ~ 7 (如果省略, 则将所有 8 个比特作为对象) 指定多个比特时, 从左边开始降序 (大→小) 记述
	表达式	超出整数 (指定实数时, 将小数点以下四舍五入后转换为整数) 赋值目标所需比特数的比特将被舍去。 (如果指定端口, 则使用低位 8 比特。 如果在比特编号中指定的比特数为 1 ~ 8 比特, 则使用低位 1 ~ 8 比特)

含义 获取指定 MO 端口的输出状态。

SAMPLE	说明
A%=MO0 ()	将 MO (07) 到 MO (00) 的输出状态赋值给变量 A%
A%=MO0 (6, 5, 1)	将 MO (06), MO (05), MO (01) 的输出状态赋值给变量 A% (上述信号都变为 1 (ON) 时, A%=7)
A%=MO (17, 15, 00)	将 MO (17), MO (15), MO (00) 的输出状态赋值给变量 A% (上述信号的 MO (15) 以外都变为 1 (ON) 时, A%=5)
A%=MO (377, 365, 255, 123)	将 MO (377), MO (365), MO (255), MO (123) 的输出 状态赋给 A% (上述信号都变为 1 (ON) 时, A%=15)

相关命令 RESET, SET

MOTOR

控制马达电源状态

联机



8

命令

格式		
MOTOR	ON OFF PWR	

- 含义** 控制马达电源的开启 / 关闭。此时，也可同时控制所有机器人的伺服开启 / 关闭。
- ON 开启马达电源。也可同时开启所有机器人的伺服。
 - OFF 关闭马达电源。同时关闭所有机器人的伺服，并且进行动态制动。带制动器的轴进行制动并且锁定。
 - PWR 仅开启马达电源。

SAMPLE	说明
MOTOR ON	 开启马达电源，并且开启所有机器人的伺服。

A

B

C

D

E

F

G

H

I

J

K

L

M

MOVE

进行绝对位置移动

联机



命令

格式		
MOVE [机器人编号](轴编号 , ...)	PTP P L C	, 目标位置坐标点 , 选项 , 选项 ...

含义 执行机器人(轴)的绝对位置移动命令。
针对其它机器人轴及附加轴所设定的轴，将无法通过该移动命令进行移动。

值	范围、种类
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6（可指定多个。省略时，所有轴将会被指定。）
PTP, P, L, C	移动种类。PTP, P：PTP、L：直线插补、C：圆弧插补
目标位置坐标点	机器人停止位置： 直接数值（坐标值）指定、坐标点定义（坐标点编号）指定、坐标点名称指定 ※ 关于点位名称的添加和编辑方法，请参阅 RCX340《用户手册》的“点位编辑”。
选项	相对速度指定、绝对速度指定、直线速度指定 拱形移动指定 STOPON 条件指定、CONTON 条件指定、CONT 指定 加速度指定、减速度指定 坐标平面指定、端口输出指定（可指定多个）

选配	PTP	直线插补	圆弧插补	备 注
相对速度指定 (SPEED)	✓	✓	✓	仅指定 MOVE 语句有效
绝对速度指定 (DSPEED)	✓	✓	✓	仅指定 MOVE 语句有效
直线速度指定 (VEL)	×	✓	✓	仅指定 MOVE 语句有效
拱形移动	✓	×	×	仅指定 MOVE 语句有效
STOPON 条件指定	✓	✓	✓	仅程序执行时有效
CONTON 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 (MOVE P)
从当前机器人停止位置到指定点位位置，各轴可通过最短移动路径而移动。

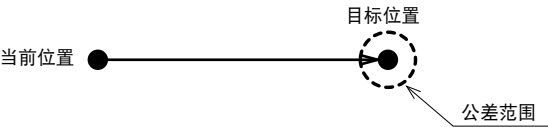
执行开始条件：所有的移动对象轴均结束移动(到达公差范围内 ※1)。
执行结束条件：所有的移动对象轴均到达 OUT 有效位置范围内 ※2。
移动对象轴同时到达。无法保证移动的轨迹。

SAMPLE	说明
MOVE P, P0	……………主机器人轴机器人 1 从当前位置向 P0 所指定位置进行移动 (MOVE PTP, P0 也一样)
MOVE P,END_POINT	……………使机器人 1 从当前位置向点位名称“END_POINT”中指定的位置移动

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。
- 关于点位名称的添加和编辑方法，请参阅 RCX340《用户手册》的“点位编辑”。

(※1) 轴参数 " 公差 "
设置相对机器人移动时的目标位置的定位完成范围。
机器人的当前位置进入设置的范围后，即判断定位完成。



(※2) 继 MOVE P 命令之后的命令注意点 (轴参数 "OUT 有效位置")

设置执行 PTP 动作命令时的相对于目标位置的执行完成范围。但是, 仅适用于 PTP 动作。

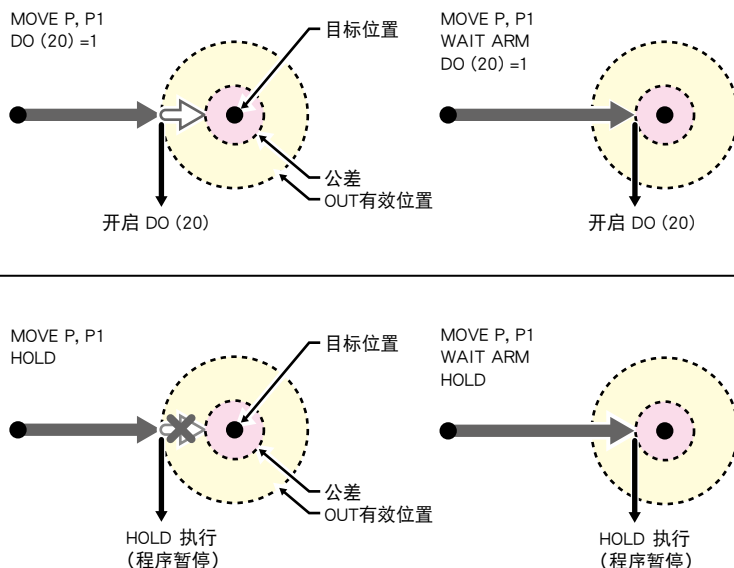
MOVE P 命令的下一个命令为信号输出命令等可执行命令时, 移动对象轴进入 OUT 有效位置范围内后, 将开始执行下一个命令。即, 在轴到达目标位置的公差范围内之前, 已开始下一个命令。

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

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

MEMO

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



●直线插补移动 (MOVE L)

执行开始条件：所有的移动对象轴正结束移动(到达公差范围内)。
执行结束条件：所有的移动对象轴已结束移动(到达公差范围内)。
所有移动对象轴都同时到达。

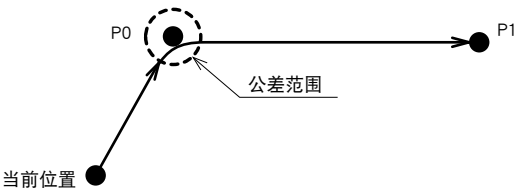
 注意
RCX340/RCX341/RCX320 的插补移动命令的动作及结束条件与以往机型 (RCX240 等) 不同。
通过为移动命令附加 CONT 指定，设置与以往机型相同的动作及结束条件。

 MEMO

对于带有 R 轴指定的机器人，可能会由于 R 轴的移动距离，使得 R 轴速度变得过快而出错。

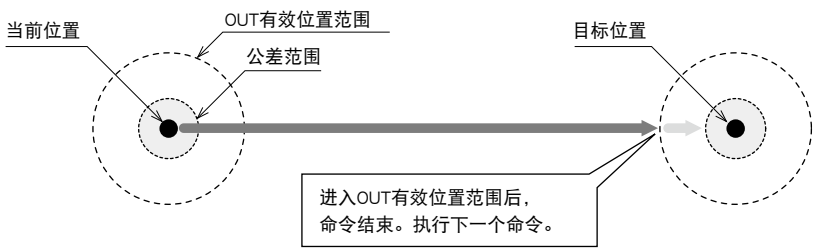
SAMPLE	说明
MOVE L, P0, P1	机器人 1 从当前位置向 P0、P1 所指定的位置进行移动。

SAMPLE: MOVE L

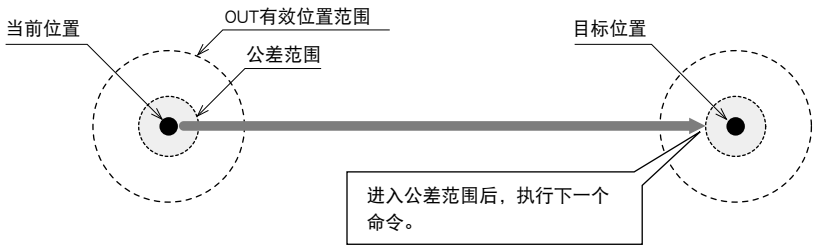


移动命令动作的区别

1. PTP动作



2. 插补动作



● 圆弧插补移动 (MOVE C)

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。

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。

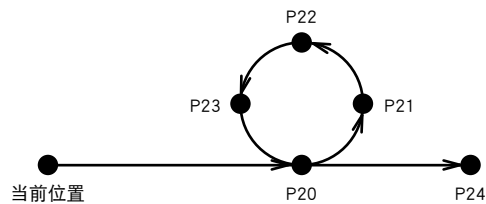


注意

RCX340/RCX341/RCX320 的插补移动命令的动作及结束条件与以往机型（RCX240 等）不同。
通过为移动命令附加 CONT 指定，设置与以往机型相同的动作及结束条件。

SAMPLE	说明
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

 MEMO

- 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。

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

p1 p2 p3 p4 p5 p6 f f1 f2

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

含义

指定坐标值。可进行该指定的移动种类包括 PTP 移动及直线插值移动。

- 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（同时存在两种指定单位时），所有的坐标值将被作为毫米 / 度单位处理。

标注 f) 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。(*1)
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)。

f	1	表示设置为在右手系统中向指定位置移动。	0	无手系统标志设定
	2	表示设置为在左手系统中向指定位置移动。		1 及 2 以外的数值或未指定数值时

标注 f1、2) 在直接指定数值过程中，当为 YK-TW 且以毫米为单位来指定坐标值时

--> 可以指定第 1 机械臂转数信息及第 2 机械臂转数信息。(*1)

当指定机械臂转数信息时，在 f1 及 f2 可指定 -1、0、1 中的任何一个。

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

f1, 2	0	表示将机械臂转数信息设定为向 0 的位置进行移动。
		-1、0、1 以外的数值或者未设定第 1、第 2 机械臂转数信息时
	1	表示将机械臂转数信息设定为向 1 的位置进行移动。
	-1	表示将机械臂转数信息设定为向 -1 的位置进行移动。

参照

*1：第 4 章“3 坐标点数据的类型”



注意

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



MEMO

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

SAMPLE	说明
MOVE P, 10000 10000 1000 1000 0 0	从当前位置向指定位置进行 PTP 移动
MOVE P, 100.0 100.0 50.0 45.0 0.0 0.0 2	使机器人 1 从当前位置向左手系统的指定位置进行 PTP 移动
MOVE P, -180.0 -430.0 50.0 180.0 0.0 0.0 1 -1 1	使机器人 1 从当前位置向右手系统的第 1 机械臂为 -180° 以下、360° 以上，第 2 机械臂为 180° 以上、360° 以下的指定位置，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。



注意

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



MEMO

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

说明

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



注意

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

● 坐标点名称指定

PTP

直线插补

圆弧插补

格式

坐标点名称, 坐标点名称 ...

含义 指定坐标点(点位)名称。通过用逗号(,)隔开可指定多个数据。
对于圆弧插补, 需指定偶数个数的点位。



MEMO

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

SAMPLE

说明

MOVE P, PICK_POINT	使机器人 1 从当前位置向点位名称 “PICK_POINT”中指定的位置移动
MOVE C, CIRCLE_PNT1, CIRCLE_PNT2	使机器人 1 从当前位置向 P20, P0, P100 指定的位置依次移动

选项种类

● 相对速度指定 (SPEED)

PTP 直线插补 圆弧插补

格式	
1. SPEED = 表达式 2. S = 表达式	
值	范围
表达式	1 ~ 100 (单位 : %)

含义

使用<表达式>指定程序移动速度。
机器人的最高速度 (mm / 秒) × 自动移动速度 (%) × 程序移动速度 S (%)

动作速度由手持编程器的操作或 ASPEED 命令设置的自动移动速度与 SPEED 等命令指定的程序移动速度的乘积决定。

自动移动速度	80%
× 程序移动速度 S	50%
移动速度	= 40% (80% × 50%)

 补充 虽然规定了最高速度，但并不保证按照指定速度移动。
[自动移动速度] 通过手持编程器的操作或 ASPEED 命令设置
[程序移动速度] 由 SPEED 命令或 MOVE、DRIVE 的速度指定设置

SAMPLE (自动移动速度 : 80%)	说明
MOVE P, P0	……从当前位置向 P0 移动 (速度 : 实质 80%)
MOVE P, P1, S=50	……从当前位置向 P1 移动 (速度 : 实质 40%)

● 绝对速度指定 (DSPEED)

PTP 直线插补 圆弧插补

格式	
1. DSPEED = 表达式 2. DS = 表达式	
值	范围
表达式	0.01 ~ 100.00 (单位 : %)

含义

使用表达式指定机器人的移动速度。(仅对指定的 MOVE 语句有效。)
机器人的最高速度 (mm/ 秒或度 / 秒) × 移动速度 DS (%)

不受自动移动速度值 (%) 的影响，始终按照 DSPEED 的表达式数值 (%) 移动。

 补充 SPEED 选项与 DSPEED 选项无法并用。

SAMPLE (自动移动速度 : 80%)	说明
MOVE P, P0	……从当前位置向 P0 移动 (速度 : 实质 80%)
MOVE P, P1, DS=50	……从当前位置向 P1 移动 (速度 : 实质 50%)
MOVE P, P2, DS=0.1	……从当前位置向 P2 移动 (速度 : 实质 0.1%)

MOVE

● 直线速度指定 (VEL)

PTP

直线插补

圆弧插补

格式

VEL = 表达式

值

范围

表达式

1 ~ 每个型号的最高速度 (单位 : mm / 秒)

含义

在 mm / 秒单位的 <表达式> 中执行机器人的 XYZ 轴合成最高速度。针对水平多关节型机器人或者 XY 型机器人，当移动种类为直线插值移动以及圆弧插值移动时，可以指定该选项。
仅对指定的 MOVE 语句有效。
不受自动移动速度值 (%) 的影响，始终按照 VEL 的表达式数值 (mm / 秒) 移动。



补充

- 虽然规定了最高速度，但并不保证按照指定速度移动。
- <表达式> 虽可以在小数点以下进行输入，但执行时会出现舍去小数点以下值的动作。

SAMPLE

说明

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

● 拱形移动指定

PTP

直线插补

圆弧插补

形式	
$x = \text{表达式} \{ \text{表达式} 1, \text{表达式} 2 \}$	
值	内容
x	A1 ~ A6 的拱形轴指定
表达式	拱形位置 · 直接数值指定 · 变量指定 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
表达式 1	距起点的直线距离（拱形距离 1） 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
表达式 2	到终点的直线距离（拱形距离 2） 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

MEMO

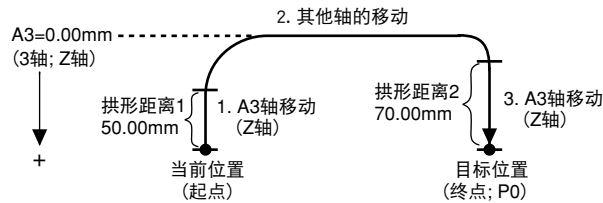
- 当<表达式>、<表达式 1>、<表达式 2>中的任何一个有实数值时，全部作为实数值进行处理。
- 表达式中指定的数值是根据机器人坐标指定的。不考虑机械手定义、工件定义设置等。

- 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 中指定的轴移动拱形距离 1 以上时，其它轴会移动至目标位置（下图中 2）。
 3. x 中指定的轴在其它轴移动结束时刻移动至目标位置，使剩余移动距离成为拱形距离 2（下图中 3）。
 4. 所有轴均进入 OUT 有效位置范围内后，结束命令。
- 可进行该指定的移动种类包括 PTP 移动以及连接 PTP 移动。
当 X 中指定轴是水平多关节型机器人的第 1 机械臂、第 2 机械臂或者 XY 型机器人的第 1 轴、第 2 轴时，<表达式> 及目标位置的 值限定为整数（脉冲单位）。



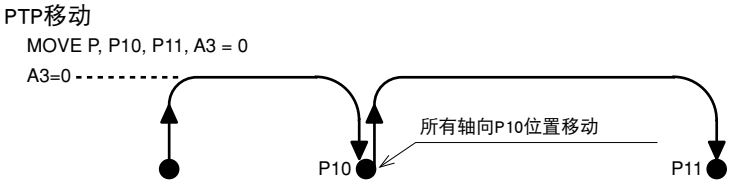
补充
轴的拱形距离参数可通过 ARCHP1/ARCHP2 进行变更。（拱形脉冲 1 为拱形距离 1、拱形脉冲 2 为拱形距离 2。）
该值越小，移动的执行时间也会越短。

SAMPLE	说明
MOVE P, P0, A3=0.00 { 50.00, 70.00 }	A3 轴从当前位置移动到 0.00mm 位置。 之后，其他轴移动到 P0，最后 A3 轴移动到 P0



MEMO

在 PTP 移动中指定了多个坐标点时，指定为拱形移动的轴也向目标位置移动。



MOVE

● STOPON 条件指定

PTP

直线插补

圆弧插补

格式

STOPON 条件表达式

含义 移动中条件表达式成立时，移动将停止并执行下一个命令。
此时的停止动作为**减速停止**，因此在条件成立后将产生**减速移动距离**。

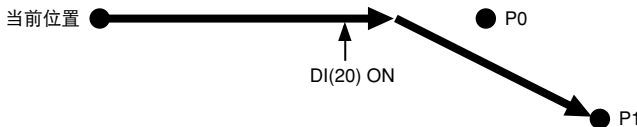
- 倘条件不成立，则运行至目标位置。
- 移动开始时如果条件表达式成立，将不移动，直接结束命令。
- 仅程序执行时有效。

注意 与 STOPON 条件，CONTON 条件同时附加时，以下优先关系中，低位指定将无效。
优先权：STOPON 条件指定 > CONTON 条件指定 > CONT 指定

SAMPLE

明

MOVE P, P0, STOPON DI (20) =1从当前位置向 P0 位置移动，如果中途 DI (20) =1 的条件成立，则减速停止并执行下一个步骤。
MOVE P, P1



MEMO

当条件表达式表示数值的情况下，表达式的值非 0 时为“真”，当为 0 时则为“假”。

●CONTON 条件指定（联结动作）

PTP 直线插补 圆弧插补

格式
CONTON 条件表达式

含义 移动中条件表达式成立时，开始减速停止，未等移动对象轴的动作结束（到达公差范围内）便开始执行下一个命令。减速停止中将产生减速移动距离。

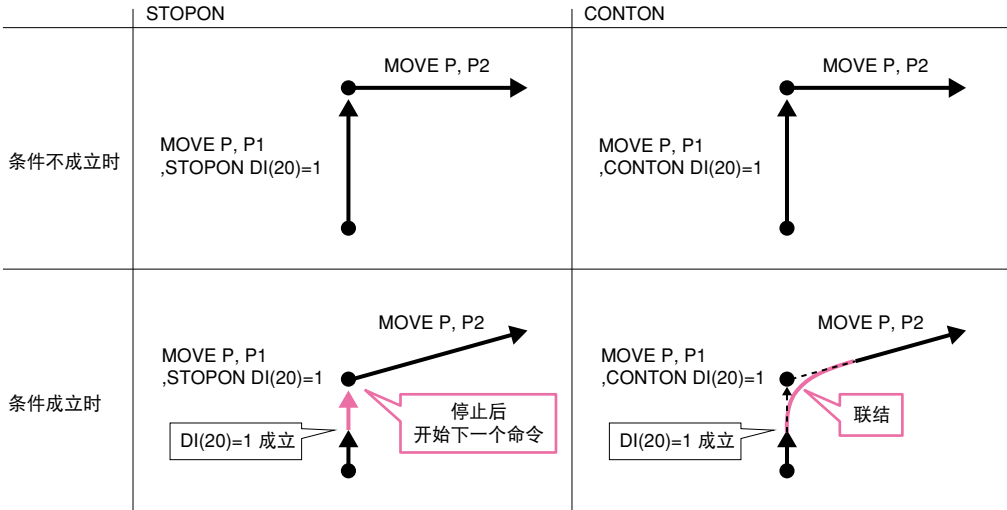
下一个命令为移动命令时，则通过连接减速部分及下一个移动命令的加速部分综合在一起，机器人将继续移动，不予中途停止。

- 假如条件不成立，则运行至目标位置。
- 仅程序执行时有效。

注意 与 STOPON 条件，CONTON 条件同时附加时，以下优先关系中，低位指定将无效。
优先权：STOPON 条件指定 > CONTON 条件指定 > CONT 指定

SAMPLE	明
MOVE P, P1, CONTON DI (20) =1	……从当前位置向 P1 位置移动，如果中途 DI (20) =1 的条件成立，则开始减速停止并执行下一个步骤。

与 STOPON 条件的差异



MEMO

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

● CONT 指定（联结动作）

PTP

直线插补

圆弧插补

格式

CONT

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

仅对指定 MOVE 语句有效。



注意

与 STOPON 条件，CONTON 条件同时附加时，以下优先关系中，低位指定将无效。

优先权：STOPON 条件指定 > CONTON 条件指定 > CONT 指定



补充

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。

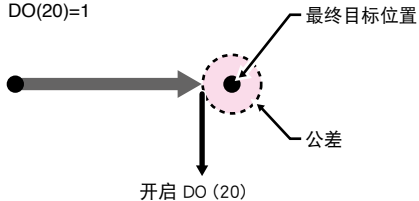
● 使用 MOVE L / C 命令中的 CONT 指定时的注意事项

指定 CONT 的 MOVE L / 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。也就是说，在轴到达目标位置的公差范围之内之前，开始下一个命令。

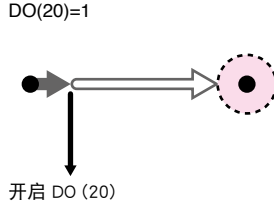
信号输出（DO 等）	最后路径的移动开始后，立即输出信号。
DELAY	最后路径的移动开始后，立即执行 DELAY 命令，并开始等待时间。
HALT	最后路径的移动开始后，程序立即停止，并被复位。因此，移动停止。
HALTALL	最后路径的移动开始后，正在运行的所有程序停止，任务 1 复位，其他任务结束。因此，移动停止。
HOLD	最后路径的移动开始后，程序立即暂停。因此，移动停止。
HOLDALL	最后路径的移动开始后，正在运行的所有程序将暂停。因此，移动停止。
WAIT	最后路径的移动开始后，立即执行 WAIT 命令。

MOVE 命令

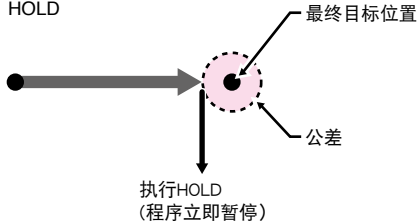
MOVE L,P1
DO(20)=1



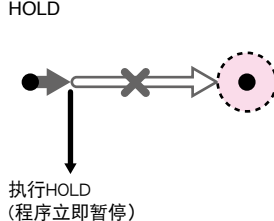
MOVE L,P1, CONT
DO(20)=1



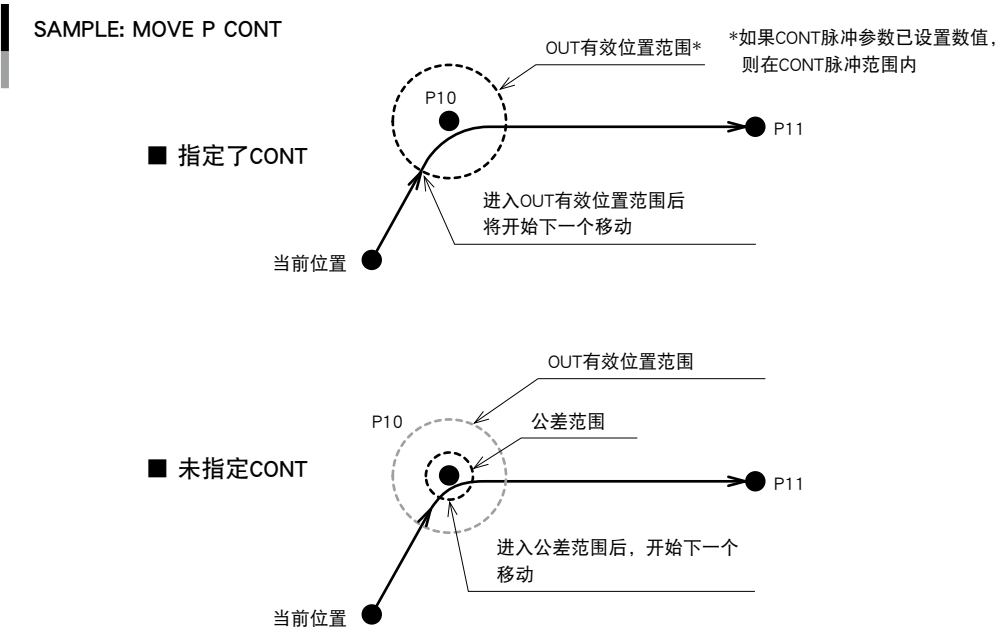
MOVE L,P1
HOLD



MOVE L,P1, CONT
HOLD



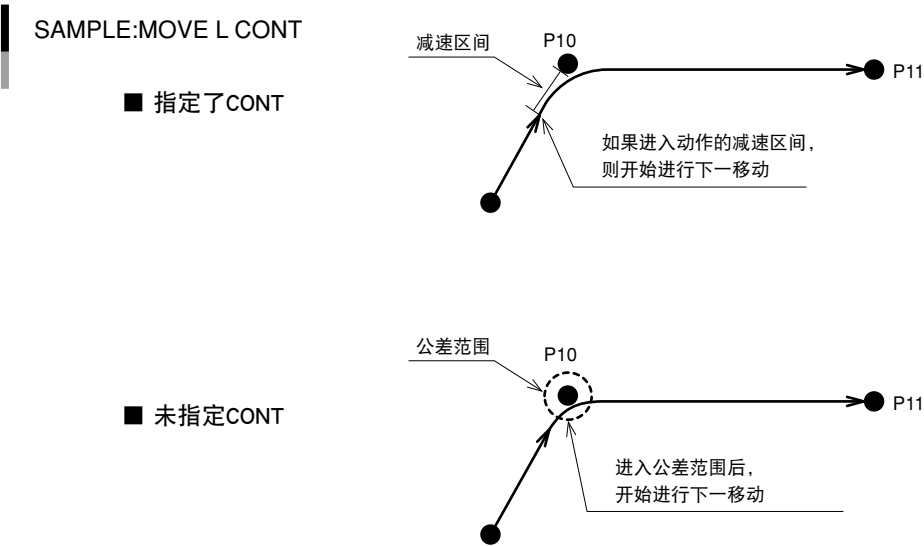
SAMPLE	说明
MOVE P, P10, P11, CONT	从当前位置移动至 P10 指定的位置，不等移动轴至达公差范围内，就开始向 P11 移动



SAMPLE	说明
MOVE L, P10, CONT MOVE L, P11	使机器人 1 从当前位置移动到 P10 指定的位置，在移动轴到达公差范围内之前，向 P11 进行直线插补移动，到达公差范围内后，动作结束

MEMO

已指定 CONT 的插补移动采用连续动作时，不会在中途的点位停止。



● 加速度指定

PTP

直线插补

圆弧插补

格式

 $ACC = \text{表达式}$

值

范围

表达式

1 ~ 100 (单位: %)

含义

在<表达式>中指定由加速度系数参数决定的相对于机器人加速度的比率, 并决定动作时的加速度。
可指定此移动的种类为直线插补。
仅对指定的 MOVE 语句有效。

 $\text{加速度系数参数} \times \text{加速度指定 ACC}(\%)$

SAMPLE

说明

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

 $DEC = \text{表达式}$

值

范围

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)。
仅对指定的 MOVE 语句有效。

 $\text{加速度系数参数} \times \text{减速度指定 DEC}(\%)$

SAMPLE

说明

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					

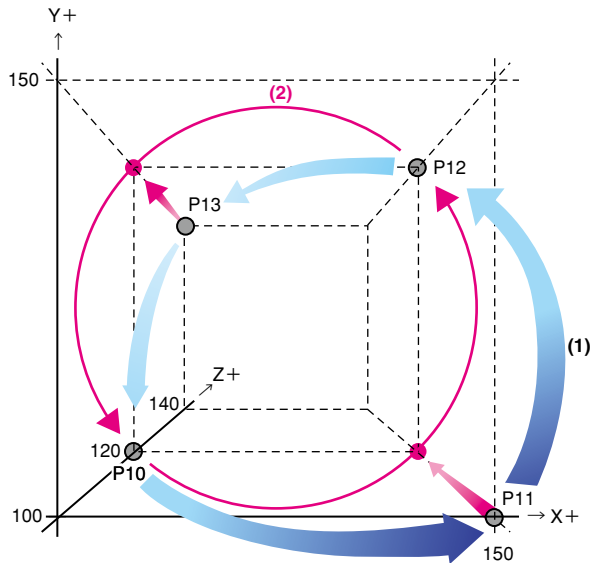
值	含义	值	含义	值	含义
XY	XY 坐标平面	YZ	YZ 坐标平面	ZX	ZX 坐标平面

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

-  补充
- 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	说明
P10 = 100.000 100.000 120.000 0.000 0.000 0.000	
P11 = 150.000 100.000 100.000 0.000 0.000 0.000	
P12 = 150.000 150.000 120.000 0.000 0.000 0.000	
P13 = 100.000 150.000 140.000 0.000 0.000 0.000	
MOVE P, P10	……从当前位置移动至 P10 指定的位置。
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 (1)
MOVE C, P11, P12, XY	……连续移动在 P10、P11、P12 与 P12、P13、P10 上形成的 XY 平面圆弧。
MOVE C, P13, P10, XY	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。(2)

SAMPLE:MOVE C 坐标平面



● 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	

标注	值	范围
m	端口编号	2 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7 (如果省略, 则将所有 8 个比特作为对象) 指定多个比特时, 从左边开始降序 (大→小) 记述
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 3 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。(并非弧长)

可指定此移动的种类为直线插补移动及圆弧插补移动。

此选项在 1 个 MOVE 语句中可指定 2 次。

指定了硬件上不存在的端口时, 将无任何输出。

参照 关于比特编号, 请参阅第 3 章“10 指定比特”。

**注意**

DO、SO 均无法输出至端口 0 和 1。

SAMPLE 1 说明

```
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
          25.85mm 的时候向 DO2 ( ) 输出 105 (&B01101001)
```

SAMPLE 2 说明

```
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 10mm 的时候开启 MO (22), 并在离开 20mm 的时
          候关闭 MO (22)
```

相关命令 MOVEI, MOVET, DRIVE, DRIVEI, WAIT ARM

MOVEI

进行相对位置移动

联机

✓

8

命令

格式

MOVEI [机器人编号] (轴编号、...) PTP, P, L, 目标位置坐标点, 选项, 选项 ...

含义 执行机器人(轴)的相对位置移动命令。
设置为其他组或附加轴的轴使用此移动命令时将不移动。

值	范围、种类
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6 (可指定多个。省略时, 所有轴将会被指定。)
PTP, P, L (*)	移动种类。PTP, P : PTP、L : 直线插补
目标位置坐标点 (*)	机器人停止位置 : 直接数值 (坐标值) 指定、坐标点定义 (坐标点编号) 指定
选项 (*)	相对速度指定、绝对速度指定、直线速度指定 STOPON 条件指定、CONTON 条件指定、CONT 指定、 加速度指定、减速度指定

* **参照** "MOVE" 命令 (样本命令以「MOVEI」取代「MOVE」。)

选配	PTP	直线插补	备 注
相对速度指定 (SPEED)	✓	✓	仅指定 MOVEI 语句有效
绝对速度指定 (DSPEED)	✓	✓	仅指定 MOVEI 语句有效
速度指定 (VEL)	×	✓	仅指定 MOVEI 语句有效
STOPON 条件指定	✓	✓	仅程序执行时有效
CONTON 条件指定	✓	✓	仅程序执行时有效
CONT 指定	✓	✓	仅指定 MOVEI 语句有效
加速度指定	✓	✓	仅指定 MOVEI 语句有效
减速度指定	×	✓	仅指定 MOVEI 语句有效

MEMO

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

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

命令

格式		
MOVET [机器人编号] (轴编号、...)	PTP P L	, 目标位置坐标点, 选项, 选项 ...

含义 执行机器人(轴)的相对位置移动命令。
设置为其他组或附加轴的轴使用此移动命令时将不移动。

值	范围、种类
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6 (可指定多个。省略时, 所有轴将会被指定。)
PTP, P, L (*)	移动种类。PTP, P : PTP、L : 直线插补
目标位置坐标点 (*)	机器人停止位置 : 直接数值 (坐标值) 指定、坐标点定义 (坐标点编号) 指定
选项 (*)	相对速度指定、绝对速度指定、直线速度指定 STOPON 条件指定、CONTON 条件指定、CONT 指定、 加速度指定、减速度指定

* **参照** "MOVE" 命令 (样本命令以「MOVET」取代「MOVE」。)

选配	PTP	直线插补	备 注
相对速度指定 (SPEED)	✓	✓	仅指定 MOVEI 语句有效
绝对速度指定 (DSPEED)	✓	✓	仅指定 MOVEI 语句有效
速度指定 (VEL)	×	✓	仅指定 MOVEI 语句有效
STOPON 条件指定	✓	✓	仅程序执行时有效
CONTON 条件指定	✓	✓	仅程序执行时有效
CONT 指定	✓	✓	仅指定 MOVEI 语句有效
加速度指定	✓	✓	仅指定 MOVEI 语句有效
减速度指定	×	✓	仅指定 MOVEI 语句有效

相关命令	MOVE, MOVEI, DRIVE, DRIVEI, WAIT ARM
------	--------------------------------------

MTRDUTY

获取指定轴的马达负载率

联机



8

函数

格式	
MTRDUTY [机器人编号] (轴编号)	
标注	值
机器人编号	1 ～ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ～ 6

含义 根据机器人编号所指定的机器人的轴编号, 获取指定轴的马达负载率 (0 ～ 100 [%])。
如果对象轴的当前电流值超过额定电流值, 则马达负载率将增加,
在该值达到 100% 时发生 [17.800 : 马达超负荷] 错误。

SAMPLE	说明
A=MTRDUTY (1)	……………将机器人 1 的第 1 轴的马达负载率赋给变量 A

A

B

C

D

E

F

G

H

I

J

K

L

M

OFFLINE

将通信端口设置为脱机模式

联机



命令

格式		
OFFLINE	ETH CMU	

含义 更改指定端口的通信模式参数，并使通信模式脱机。

ETH	将 Ethernet 的通信模式参数更改为脱机，并清除收发缓冲区。
CMU	将 RS-232C 的通信模式参数更改为脱机，并清除通信错误的解除与接收缓冲区。
无指定	将 Ethernet 及 RS-232C 的通信模式参数更改为脱机，并清除通信错误的解除（仅限 RS232C）与收发缓冲区。

MEMO

脱机（脱机模式）是指，未接收联机命令（无效）的模式。

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

参照 关于 SEND 语句，请参阅本章“SEND”。

相关命令	ONLINE
------	--------

ON ERROR GOTO

在发生错误时跳转至指定标签

联机

×

8

命令

格式

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。

错误输出信息	ERR	错误代码编号
	ERL	错误发生行编号

MEMO

- 发生了[17.800：马达超负荷]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>。

SAMPLE	说明
ON ERROR GOTO *ER1 FOR A = 0 TO 9 P [A+10] = P [A] NEXT A *L99: HALT ' ERROR ROUTINE *ER1: IF ERR = &H000600CC THEN *NEXT1 IF ERR = &H000600CE THEN *NEXT2 ON ERROR GOTO 0 *NEXT1: RESUME NEXT *NEXT2: RESUME *L99	 确认是否发生了 [Point doesn't exist] 的错误 确认是否发生了 [Subscript out of range] 的错误 显示错误并停止执行 跳转至错误发生行的下一行继续执行 跳转至标签 *L99 并继续执行

相关命令

RESUME

ON ~ GOSUB

执行由表达式的值指定的子例程

联机

×

命令

格式

ON 表达式 GOSUB 标签 1, 标签 2...

※GOSUB 也可写成 GO SUB

值	含义
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	说明
<pre>' 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</pre>	

相关命令 GOSUB, RETURN

ON ~ GOTO

跳转至表达式的值所指定的标签

联机

×

8

命令

格式

ON 表达式 GOTO 标签 1, 标签 2...

※GOTO 也可写成 GO TO

值

含义

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

说明

```
' 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

ONLINE

将通信端口设置为联机模式

联机



命令

格式		
ONLINE	ETH CMU	

含义 更改指定端口的通信模式参数，并使通信模式联机。

ETH	将 Ethernet 的通信模式参数更改为联机，并清除收发缓冲区。
CMU	将 RS-232C 的通信模式参数更改为联机，并清除通信错误的解除与接收缓冲区。
无指定	将 Ethernet 及 RS-232C 的通信模式参数更改为联机，并清除通信错误的解除（仅限 RS232C）与收发缓冲区。

MEMO

联机（联机模式）是指，接收联机命令（有效）的模式。

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

参照 关于 SEND 语句，请参阅本章“SEND”。

相关命令	OFFLINE
------	---------

OPEN

打开指定的通用以太网端口

联机



8

命令

格式

OPEN GPm

标注

值

范围

m

通用以太网端口编号

0 ~ 7

含义 打开指定的通用以太网端口编号的通信端口。

SAMPLE

说明

OPEN GP1	……………打开通用以太网端口编号 1 的通信端口
SEND “123” TO GP1	……………从通用以太网端口编号 1 发送字符串 “123”
SEND GP1 TO A\$	……………从通用以太网端口编号 1 接收数据，将接收的数据保存至变量 A\$
CLOSE GP1	……………关闭通用以太网端口编号 1 的通信端口

相关命令

CLOSE, SEND, SETGEP, GEPSTS

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

ORD

计算字符编码

联机



函数

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	说明
A=ORD (“B”)	……………将 66 (=H42) 赋值给 A

相关命令 CHR\$

ORGORD

设置 / 获取机器人的返回原点顺序

联机



8

命令

格式

ORGORD [机器人编号] 表达式

标注

值

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

表达式 n ~ nnnnnn (n : 0 ~ 6)

含义

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

函数

格式

ORGORD [机器人编号]

标注

值

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

含义

获取针对<机器人编号>指定机器人进行原点归复动作的轴顺序参数。

SAMPLE

说明

A=3
ORGORD A指定为第 3 轴先返回原点
ORIGIN 1机器人 1 中第 3 轴的原点归复动作完成之后, 执行其余轴的同步原点归复动作。
MOVE P, P0
A=ORGORD将主组的返回原点顺序参数赋值给变量 A
HALT

相关命令

ORIGIN

ORIGIN

原点归复

联机



命令

格式	
ORIGIN 机器人编号 马达类型	
值	范围、含义
机器人编号	0：所有机器人、1～4：仅执行指定的机器人
马达类型	0：所有类型、1：增量式马达、 2：绝对式马达、9：未原点复归的轴 (省略时，所有类型被指定。)
含义 执行机器人增量式马达轴的返回原点动作。对于半绝对式规格的轴执行绝对位置搜索动作。 如果中途停止，将变为未返回原点状态。 当设定了多台机器人时，机器人 1 的原点归复动作结束之后，依次进行机器人 2～4 的原点归复动作。	
SAMPLE	说明
ORIGIN 0, 1 ……所有机器人的仅增量式马达轴进行原点归复动作	
相关命令	ORGORD, MCHREF

OUT

开启指定端口的输出

联机

×

8

命令

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)	, 表达式
标注	值	范围、含义
m	端口编号	2 ~ 7, 10 ~ 17, 20 ~ 27 指定了硬件上不存在的端口时，将无任何输出。
b	比特编号 参照 请参阅第 3 章“10 指定比特”	0 ~ 7（如果省略，则将所有 8 个比特作为对象） 指定多个比特时，从左边开始降序（大→小）记述
表达式	输出至指定端口的值	时间 0 ~ 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。

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[6.225：OUT 用内存不足]的错误。



注意
DO, SO 均无法输出至端口 0 和 1。

SAMPLE	说明
OUT DO2 () , 200	DO (27 ~ 20) 开启，200ms 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

OUTPOS

设置 / 获取机器人的 OUT 有效位置参数

联机



命令

格式	
1. OUTPOS [机器人编号] 表达式	
2. OUTPOS [机器人编号] (轴编号) = 表达式	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6（省略时，所有轴被指定。）
表达式	1 ～ 9999999（单位：脉冲）

含义 将轴的参数的 OUT 有效位置变更为 <表达式> 的值。

[格式 1] 对指定机器人设定的所有轴进行变更。
[格式 2] 仅更改 <轴编号> 中指定的轴。

参照 本章“MOVE” 轴参数 "OUT 有效位置 "

函数

格式	
OUTPOS [机器人编号] (轴编号)	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6（省略时，所有轴被指定。）

含义 获取指定轴的 OUT 有效位置参数值。

SAMPLE 1	说明
OUTPOS 2000	………将机器人 1 的所有轴的 OUTPOS 参数变更为 2000
SAMPLE 2	说明
OUTPOS (3) = 2000	………将机器人 1 的第 3 轴的 OUTPOS 参数变更为 2000
SAMPLE 3	说明
A% = OUTPOS (3)	………获取机器人 1 的第 3 轴的 OUTPOS 参数的值， 赋值到整数型变量 A%

PATH

设置 PATH 移动路径

联机

×

8

命令

格式		
PATH	[机器人编号] (轴编号, ...)	L C, 目标位置坐标点, 选项, 选项 ...
值	范围、种类	
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)	
轴编号	1 ~ 6 (可指定多个。省略时, 所有轴将会被指定。)	
L, C	移动种类。L: 直线插补、C: 圆弧插补	
目标位置坐标点	机器人停止位置: 直接数值 (坐标值) 指定、坐标点定义 (坐标点编号) 指定	
选项	速度指定、坐标平面指定 (仅圆弧插补)、端口输出指定	

含义 设置指定机器人的设定轴的 PATH 移动路径。只可在 PATH SET 命令与 PATH END 命令之间执行。在上述区间以外执行时, 将出错。



注意

- 指定起点和移动路线时, 注意不要使位置坐标相同, 将发生错误。
- 如果坐标属性参数仅指定 “R”, 将发生错误。
- 在 PATH 语句指定的坐标中, 能够移动的只有 X,Y,Z 的坐标值。
除此之外的坐标值, 使用 PATH 移动路径的起点坐标值, 例如: 使用 R 轴时, 则不能将 R 轴移动至指定座标。



MEMO

水平多关节机器人及旋转轴上的直交机器人在执行 PATH 指令时, R 轴会保持 PATH 执行开始时的坐标 (角度)

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, 只有指定位点的 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”) 连接移动路径, 可不停止直接移动。

坐标点指定的种类

● 直接数值指定

直线插补

圆弧插补

格式

p1 p2 p3 p4 p5 p6 f f1 f2

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

含义 指定坐标值。

根据此时的数值，坐标值的单位如下：

- 整数值的情况：脉冲单位
- 实数值（带有小数点）的情况：毫米单位
- 整数与实数同时存在：均为毫米单位

此格式可指定的移动目标位置点位为 1 个。

此外，可进行此点位指定的移动种类仅限直线插补移动。

标注 f) 对水平多关节型机器人以毫米单位指定坐标值 --> 可指定手系统标志。(*1)

设置的手系统标志时，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)。

f	1	设置为在右手系统中向指定位置移动。	0	无手系统标志设定
	2	设置为在左手系统中向指定位置移动。		1 及 2 以外的数值或未指定数值时

在移动路径的起点到终点之间，手系统必须始终相同。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位置的手系统一致。手系统不同时，将出错并无法动作。

标注 f1、2) 对 YK-TW 系列以毫米单位指定坐标值 --> 可指定第 1 · 第 2 机械臂转数信息 (*1)

当指定机械臂转数信息时，在 f1 及 f2 可指定 -1、0、1 中的任何一个。

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

f1, 2	0	机械臂转数信息向 0 的位置移动
		-1、0、1 以外的数值或者未设定第 1、第 2 机械臂转数信息时
	1	机械臂转数信息向 1 的位置移动
	-1	机械臂转数信息向 -1 的位置移动

在移动路径的起点到终点之间，第 1 · 第 2 机械臂转数信息必须始终相同。

中途无法更改第 1 · 第 2 机械臂转数信息。



注意

请使 PATH 移动中的第 1 · 第 2 机械臂转数信息与 PATH 移动路径起点的第 1 · 第 2 机械臂转数信息一致。如果不一致，将出错并无法动作。

参照 *1：第 4 章“3 坐标点数据的类型”

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	说明
PATH L,10000 10000 1000 1000 0 0	………将机器人 1 在移动路径上的点设为脉冲单位以进行直线插补移动
PATH L,150.000 250.000 10.000 30.000 0.000 0.000 1	………将机器人 1 在移动路径上的点设为毫米单位（右手系统指定）以进行直线插补移动

●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>中指定移动目标位置的坐标点。

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，必须对每个圆弧指定 2 个点位。

MEMO

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

- 注意
- 请使得 PATH 移动中的手系统与 PATH 移动路径的起点的手系统相同。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位置的手系统一致。手系统不同时，将出错并无法动作。
 - 请使 PATH 移动中的第 1・第 2 机械臂转数信息与 PATH 移动路径起点的第 1・第 2 机械臂转数信息一致。如果不一致，将出错并无法动作。

SAMPLE	说明
PATH L,P1,P2,P3	………通过设置，使机器人 1 从当前位置向 P1、P2P3 中指定的位置依次进行直线插补移动
PATH C,P5,P6,P7,P8	………通过设置，使机器人 1 从当前位置，在当前位置、P5、P6 和 P6、P7、P8 所构成的圆弧上进行插补移动

选项种类

参照 "MOVE" 命令 (样本命令以「PATH」取代「MOVE」。)

● 速度指定 1

直线插补

圆弧插补

● 速度指定 2

直线插补

圆弧插补

● 坐标平面指定

直线插补

圆弧插补

格式

XY
YZ
ZX

● 端口输出指定

直线插补

圆弧插补

格式 1

DO
MO
SO
 $m(b, \dots, b) = \text{表达式 1} @ \text{表达式 2}$

格式 2

DO
MO
SO
 $(mb, \dots, mb) = \text{表达式 1} @ \text{表达式 2}$

SAMPLE

说明

```
PATH SET
PATH L, P1, DO (20) = 1@10 .....通过设置, 使机器人 1 从当前位置向 P1 进行直线插补移动中,
                                   在距离开始位置半径 10mm 的位置将 1 输出到 DO (20)
PATH L, P2, DO (21) = 1@12.5 .....通过设置, 使机器人 1 从当前位置向 P2 进行直线插补移动中,
PATH END                                   在距离 P1 半径 12.5mm 的位置将 1 输出到 DO (21)
PATH START
```

相关命令

PATH SET, PATH END, PATH START

PATH END

结束 PATH 的路径设置

联机

×

8

命令

格式	
PATH [机器人编号] END	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）

- 含义
- 结束指定机器人的 PATH 移动路径设置。
- PATH END 命令必须与 PATH SET 命令成对使用。
 - 在 PATH SET 命令到 PATH END 命令之间，最后执行的 PATH 命令（“PATH L”、“PATH C”）的最终点位为 PATH 移动路径的终点。
 - 如果不执行 PATH SET 命令，直接执行 PATH END 命令，将出错。

SAMPLE	说明
PATH END	……………结束机器人 1 的 PATH 移动路径设置

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”。

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

PATH SET

开始 PATH 的路径设置

联机

×

命令

格式

PATH [机器人编号] SET 坐标点指定

值

范围

机器人编号

1 ~ 4 (省略时, 机器人 1 被指定。)

含义

开始指定机器人的 PATH 移动路径设置。

将移动路径的起点设置为点位指定中指定的位置。(并非只设置移动路径的起点即可让机器人实际动作。)

省略了点位指定时, 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 有效位置范围较宽, 执行 PATH SET 命令时机器人还在移动等情况。)

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。

在上述区间以外执行时, 将出错。



注意

指定起点和移动路线时, 注意不要使位置坐标相同, 将发生错误。



补充

PATH SET 语句在软件版本 Ver.1.11 以上有效。

坐标点指定的种类

● 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。



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。



注意

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

· 请使 PATH 移动中的第 1 · 第 2 机械臂转数信息与 PATH 移动路径起点的第 1 · 第 2 机械臂转数信息一致。如果不一致, 将出错并无法动作。

SAMPLE

说明

PATH SET P10机器人 1 的 PATH 移动路径起点为 P10

PATH SET WHERE机器人 1 的 PATH 移动路径起点为机器人 1 的当前位置

参照

关于 PATH 功能, 请参阅第 9 章“PATH 语句”。

●直接数值指定

格式			
p1 p2 p3 p4 p5 p6 f f1 f2			
标注	值	标注	值
p1 ~ p6	用空格隔开的各轴坐标值	f1	第 1 机械臂转数信息（仅限 YK-TW 系列）
f	手系统标志（仅限水平多关节型机器人）	f2	第 2 机械臂转数信息（仅限 YK-TW 系列）

- 含义
- 直接指定 PATH 移动路径的起点坐标值。
根据此时的数值，坐标值的单位如下：
· 整数值的 情况：脉冲单位
· 实数值（带有小数点）的情况：毫米单位
· 整数与实数同时存在：均为毫米 / 度单位

标注 f) 对水平多关节型机器人以毫米单位指定坐标值 --> 可指定手系统标志 (*1)
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)。

f	1	PATH 移动路径起点的手系统设置为右手系统	0	无手系统标志设定
	2	PATH 移动路径起点的手系统设置为左手系统		1 及 2 以外的数值或未指定数值时



- 注意
- 请使得 PATH 移动中的手系统与 PATH 移动路径的起点的手系统相同。手系统不同时，将出错并无法动作。
 - 请使 PATH 移动中的第 1・第 2 机械臂转数信息与 PATH 移动路径起点的第 1・第 2 机械臂转数信息一致。如果不一致，将出错并无法动作。

标注 f1、2) 对 YK-TW 系列以毫米单位指定坐标值 --> 可指定第 1・第 2 机械臂转数信息 (*1)
当指定机械臂转数信息时，在 f1 及 f2 可指定 -1、0、1 中的任何一个。
如果指定了 -1、0、1 以外的数值或者未设定第 1、第 2 机械臂转数信息时，则作为 0 进行处理。

f1, 2	0	将 PATH 移动路径起点的机械臂转数信息设在 0 的位置。
		-1、0、1 以外的数值或者未设定第 1、第 2 机械臂转数信息时
	1	将 PATH 移动路径起点的机械臂转数信息设在 1 的位置
	-1	将 PATH 移动路径起点的机械臂转数信息设在 -1 的位置

参照 *1：第 4 章“3 坐标点数据的类型”



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	说明
PATH SET 120 250.000 55.2 20.33 0 0	机器人 1 的 PATH 移动路径起点以毫米为单位 120.000 250.000 55.200 20.330 0.000 0.000
PATH SET -51200 80521 7045 204410 0 0	机器人 1 的 PATH 移动路径起点以脉冲为单位

相关命令 PATH, PATH END, PATH START

PATH START

开始 PATH 移动

联机

×

命令

格式				
PATH [机器人编号] START, 选项, 选项 ...				
值	范围、种类			
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)			
选项	STOPON 条件指定、CONTON 条件指定、CONT 指定			
选配	PTP	直线插补	圆弧插补	备 注
STOPON 条件指定	✓	✓	✓	仅程序执行时有效
CONTON 条件指定	✓	✓	✓	仅程序执行时有效
CONT 指定	✓	✓	✓	仅指定 PATH START 语句有效

- 含义
- 指定机器人开始进行 PATH 移动。

执行此名之前, 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, 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 移动路径的终点或停止输入等原因停止, 将结束本命令。

选项种类

参照 "MOVE" 命令

●STOPON 条件指定

格式	
STOPON = 条件表达式	
SAMPLE	说明
PATH START ,STOPON DI (20) = 1	……………机器人 1 开始进行 PATH 移动后，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。

●CONTON 条件指定（联结动作）

格式	
CONTON = 条件表达式	
SAMPLE	说明
PATH START ,CONTON DI (20) = 1	……………机器人 1 开始进行 PATH 移动后，如果中途 DI (20) =1 的条件成立，将开始减速并执行下一个步骤。

●CONT 指定（联结动作）

格式	
CONT	
SAMPLE	说明
PATH START ,CONT MOVE P, P10	……………开始进行 PATH 移动，且移动轴进入最后一个 PATH 动作的减速区间时，开始向 P10 移动

参照 关于 PATH 功能，请参阅第 9 章“PATH 语句”。

相关命令	PATH, PATH SET, PATH END
------	--------------------------

命令

格式

PDEF(托盘定义编号) = 表达式 1, 表达式 2, 表达式 3, 坐标点指定

值 范围、含义

托盘定义编号 0 ~ 39

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。

坐标点指定 用于托盘定义的坐标点。
以指定坐标点起始的 5 个连续坐标点。

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。

如果省略坐标点指定，则从以下点开始使用 5 个连续点。

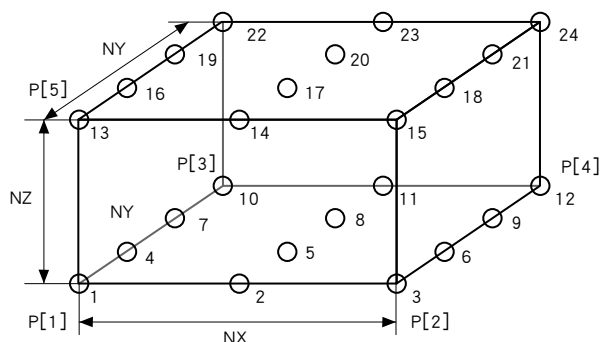
顶点 = 托盘定义编号 × 5 + 10000

例：PDEF(1)=2,3,4

使用从 P10005 开始的 5 个连续点。(P10005~P10009)

但是，如果省略点指定并更改通过指定点定义的托盘数据的定义内容，则将使用先前指定的坐标点。

坐标点的自动计算



SAMPLE

说明

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

相关命令 PMOVE

PGMTSK

获取已录入指定程序的任务编号

联机



8

函数

格式	
PGMTSK (程序编号)	
值	范围
程序编号	1 ~ 100

含义 获取已录入程序编号所指定程序的任务编号。

MEMO

如果指定任务中没有的程序编号，则显示 [3.203：程序不存在] 的错误。

SAMPLE	说明
A = PGMTSK (1)	……将已录入程序编号 1 的程序的任务编号赋给变量 A

相关命令	PGN, TSKPGM
------	-------------

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

PGN

从指定的程序名中获取程序编号

联机



函数

格式	
PGN(“程序名称”)	
值	范围
程序名称	使用字母数字 及 _ (下划线)，且应控制在 32 个字符以内

含义 获取<程序名称>中指定的程序编号。
在程序名的前后加上”(双引号)。

SAMPLE	说明
A=PGN (“PG_SUB”)	………将 PG_SUB 的程序编号赋给变量 A

相关命令	PGMTSK, TSKPGM
------	----------------

PMOVE

执行机器人的托盘移动

联机



8

命令

格式

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

值

范围、种类

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

托盘定义编号 0 ~ 39

托盘位置编号 1 ~ 32767

选项

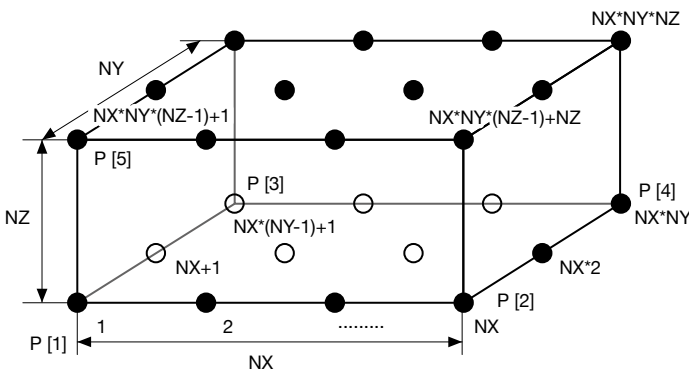
相对速度指定、拱形移动指定
STOPON 条件指定、CONTPON 条件指定、CONT 指定

选配件	PTP	备 注
相对速度指定 (SPEED)	✓	仅指定 PMOVE 语句有效
拱形移动	✓	仅指定 PMOVE 语句有效
STOPON 条件指定	✓	仅程序执行有效
CONTPON 条件指定	✓	仅程序执行有效
CONT 指定	✓	仅指定 PMOVE 语句有效

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)。
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 参照 "MOVE" 命令
- 托盘定义编号: 数值表达式
- 托盘位置编号: 数值表达式

各托盘定义中的位置编号如下。



MEMO

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

SAMPLE

说明

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

选项种类

● 相对速度指定

PTP

格式

1. SPEED = 表达式
2. S = 表达式

SAMPLE (自动移动速度: 80%)

说明

PMOVE (1, 3), S=50从当前位置按照程序速度 50% (速度: 实质 40%) 向托盘定义编号 1 的托盘位置编号 3 中指定位置移动

参照 "MOVE" 命令

● 拱形移动指定

PTP

格式

 $x = \text{表达式}, x = \text{表达式} \dots$

值

范围

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. 如果 x 中指定的轴的移动幅度大于拱形距离 1, 其他轴将移动到目标位置 (下图的 2)。
3. 如果 x 中指定的轴在其他轴完成移动时, 向目标位置移动, 以使剩余移动距离达到拱形距离 2 (下图的 3)。
4. 所有轴均进入 OUT 有效位置范围内后, 结束命令。



MEMO

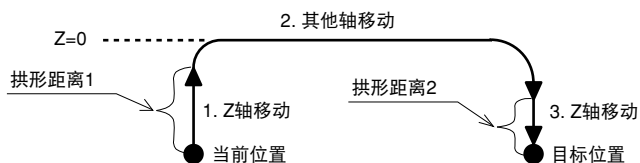
表达式中指定的数值是根据机器人坐标指定的。不考虑机械手定义、工作定义设置等。

SAMPLE

说明

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



●STOPON 条件指定

PTP

格式

STOPON 条件表达式

SAMPLE

说明

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

参照 "MOVE" 命令

●CONTON 条件指定（联结动作）

PTP

格式

CONTON 条件表达式

SAMPLE

说明

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

参照 "MOVE" 命令

相关命令

PDEF

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

命令

格式

LET Pn = p1 p2 p3 p4 p5 p6 f f1 f2

标注	值	范围
n	坐标点编号	0 ~ 29999
p1 ~ p6	坐标点数据	根据形式的不同，范围有所差异
f	手系统标志	1、2（仅限水平多关节型机器人）
f1	第 1 机械臂转数信息	-1、0、1（仅限于 YK-TW 系列）
f2	第 2 机械臂转数信息	-1、0、1（仅限于 YK-TW 系列）

含义 定义坐标点数据。

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



补充

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。

标注 f) 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。（*1）

- 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)。

f	1	表示在右手系统中设置了坐标点。	0	无手系统标志设定
	2	表示在左手系统中设置了坐标点。		1 及 2 以外的数值或未指定数值时

标注 f1、2) 在直接指定数值过程中，当为 YK-TW 且以毫米为单位来指定坐标值时

- > 可以指定第 1 机械臂转数信息及第 2 机械臂转数信息。（*1）
- 当指定机械臂转数信息时，在 f1 及 f2 可指定 -1、0、1 中的任何一个。
- 如果指定了 -1、0、1 以外的数值或者未设定第 1、第 2 机械臂转数信息时，则作为 0 进行处理。

f1, 2	0	表示将机械臂转数信息设定为 0。
		-1、0、1 以外的数值或者未设定第 1、第 2 机械臂转数信息时
	1	表示将机械臂转数信息设定为 1。
	-1	表示将机械臂转数信息设定为 -1。

参照 *1：第 4 章“3 坐标点数据的类型”

SAMPLE	说明
P1 = 0 0 0 0 0 0	…定义 P1~P3 的坐标点数据
P2 = 100.000 200.000 50.000 0.000 0.000 0.000	
P3 = 10.000 0.000 0.000 0.000 0.000 0.000	
P10= P2	…将 P2 赋值到 P10
FOR A=10 TO 15	
P [A+1] =P [A] +P3	
NEXT A	
FOR A=10 TO 16	
MOVE P,P1,P [A]	…从 P1 的位置至 P [A] 的位置进行 PTP 移动
NEXT A	
HALT	



补充
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 [9.702：点位校验和错误] 等存储器相关错误。

相关命令	坐标点赋值语句 (LET)
------	---------------

PPNT 创建托盘坐标点数据	联机
	✓

函数

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	说明
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
------	-------------

PRINT

在手持编程器上显示指定表达式的值

联机

×

命令

格式			
PRINT	表达式	表达式 ...	
值	内容		
表达式	字符串、数值、变量		

含义 此命令语句在手持编程器的画面上显示指定变量的值。输出定义如下所示。

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

MEMO

- PRINT 语句用于在手持编程器画面上显示而与画面进行数据通信。
因此，当连续执行 PRINT 语句时，程序的执行时间可能会延长。
- 显示在手持编程器上「自动运行(所有任务)」界面的「Message」栏里。

SAMPLE	说明
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

PSHFRC

设定 / 获取推压力参数

联机



命令

格式	
1. PSHFRC [机器人编号] 表达式	
2. PSHFRC [机器人编号] (轴编号) = 表达式	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6
表达式	-1000 ～ 1000（单位：％）

含义	将指定机器人轴的推压力参数变更为 < 推压力值 > 的值。 在 PUSH 语句中省略了 F 选项指定时，通过推压力参数的设定值进行推压控制。 实际的推压力如下所示。 · 额定推力 × < 表达式 > / 100 [形式 1] 对所有轴的参数进行变更。 [形式 2] 对按 < 轴编号 > 指定轴的参数进行变更。
----	--

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

函数

格式	
PSHFRC [机器人编号] (轴编号)	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
轴编号	1 ～ 6

含义	获取 < 轴编号 > 指定轴的推压力参数值。
SAMPLE	说明
A=PSHFRC (1)	……………将机器人 1 中第 1 轴的推压力参数赋给变量 A

PSHJGSP

设定 / 获取推压判定速度阈值参数

联机



8

命令

格式

- PSHJGSP [机器人编号] 表达式
- PSHJGSP [机器人编号] (轴编号) = 表达式

值

范围、含义

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

轴编号 1 ~ 6

表达式 0 : 无效, 1 ~ 100 (单位 : %)

含义

将指定轴的推压判定速度阈值参数变更为 < 表达式 > 的值。
在推压判定速度阈值参数有效, 且 PUSH 语句中的推压力达到了指定值的状态下,
仅当移动速度低于 < 表达式 > 时, 才可判定为处于推压中。

< 表达式 > 的设定值指定方式如下所示。

0 : 将阈值设定作为无效, 当推压力达到指定值时, 判定为处于推压中。

1 ~ 100 : 将 PUSH 语句中的移动速度作为 100%, 按比例指定阈值。

SAMPLE

说明

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

函数

格式

PSHJGSP [机器人编号] (轴编号)

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

轴编号 1 ~ 6

含义

获取指定轴的推压判定速度阈值参数值。

SAMPLE

说明

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

PSHMTD

设定 / 获取推压方式参数

联机

✓

命令

格式

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

值

范围、含义

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6
表达式	0 : 累计方式, 1 : 复位方式

含义 将指定轴的推压方式参数变更为 < 值 > 的值。

可以根据 < 值 > 并按下列方式对 PUSH 语句的推压方式进行指定。

0 : 对推压力达到指定值的时间进行累计, 进行推压控制结束判定。

1 : 仅在推压力连续达到指定值时, 才可进行推压控制结束判定。

当推压力低于指定值时, 将推压经过时间复位为 0。

[形式 1] 对所有轴的参数进行变更。

[形式 2] 对按 < 轴编号 > 指定轴的参数进行变更。

SAMPLE

说明

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

函数

格式

PSHMTD [机器人编号] (轴编号)

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6

含义 获取指定轴的推压方式参数值。

SAMPLE

说明

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

PSHRSLT

获取 PUSH 语句结束时的状态

联机



8

函数

格式	
PSHRSLT [机器人编号] (轴编号)	
值	范围
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6

含义 获取针对指定轴所执行的 PUSH 语句结束状态。

- 0：由于到达推压时间以外的原因结束
- 1：因到达推压时间而结束时

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

PSHSPD

设定 / 获取推压移动速度参数

联机



命令

格式	
1. PSHSPD [机器人编号] 表达式	
2. PSHSPD [机器人编号] (轴编号) = 表达式	
值	范围
机器人编号	1 ～ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ～ 6
表达式	1 ～ 100 (单位 : %)

- 含义 将轴的推压移动速度参数变更为 < 表达式 > 的值。
- PUSH 语句中的移动速度如下所示。
- PUSH 语句中没有 S、DS 选项指定时：
机器人的最高速度 (mm/ 秒或者度 / 秒) × 推压移动速度 (%) × 自动移动速度 (%) ×
 - PUSH 语句中有 S 选项指定时：
机器人的最高速度 (mm/ 秒或者度 / 秒) × 推压移动速度 (%) × 自动移动速度 (%) × S 所指定的程序移动速度 (%)
 - PUSH 语句中有 DS 选项指定时：
机器人的最高速度 (mm/ 秒或者度 / 秒) × 推压移动速度 (%) × DS 所指定的轴的移动速度 (%)

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

函数

格式	
PSHSPD [机器人编号] (轴编号)	
值	范围
机器人编号	1 ～ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ～ 6

- 含义 获取指定轴的推压移动速度参数值。
- | SAMPLE | 说明 |
|--------------|--------------------------------|
| A=PSHSPD (1) | ……将机器人 1 中第 1 轴的推压移动速度参数赋给变量 A |

PSHTIME

设定 / 获取推压时间参数

联机



8

命令

格式

- PSHTIME [机器人编号] 表达式
- PSHTIME [机器人编号] (轴编号) = 表达式

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6
表达式	1 ~ 32767 (单位 : ms)

含义

轴的推压时间参数变更为 < 表达式 > 的值。
当 PUSH 语句中省略了 TIM 选项指定时, 通过推压时间参数的设定值来进行推压控制。
[形式 1] 对所有轴的参数进行变更。
[形式 2] 对 < 轴编号 > 指定轴的参数进行变更。

SAMPLE

说明

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

函数

格式

PSHTIME [机器人编号] (轴编号)

值

范围

机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ~ 6

含义

获取轴的推压时间参数值。

SAMPLE

说明

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

PUSH

以轴为单位进行推压动作

联机



命令

格式	
PUSH [机器人编号] (轴编号 , 表达式) , 选项 , 选项	
值	范围、含义
机器人编号	1 ～ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ～ 6
表达式	马达位置 (毫米、度、脉冲) 或者 坐标点表达式

含义	以轴为单位对前进方向的推压力进行控制, 同时针对<机器人编号>指定机器人轴执行绝对位置移动。
	· 移动的种类 : 轴指定推压 PTP · 坐标点指定 : 直接数值指定、坐标点定义指定 · 选项 : 推压力指定、推压时间值、推压速度指定、STOPON 条件指定、CONTON 条件指定

移动种类

- 轴指定 PTP (Point to Point)
- 当 < 轴编号 > 指定轴的动作结束后 (到达公差范围以内后), 对轴的前进方向的推压力进行控制, 同时开始移动。
- 推压控制开始条件如下所示。
- 通过 PUSH 语句开始轴移动之后

· 连接动作结束后 (在有 CONT 指定的移动命令的下一行指定了 PUSH 语句时)
- 命令结束的条件如下所示。PUSH 语句结束状态可通过 PSHRSLT 语句进行确认。
- 轴到达目标位置的交差范围内。

· 轴的推压力达到 < 推压力值 >, 并保持该状态经过 < 推压时间值 > 以上的时间。
- 推压力解除条件如下所示。
- PUSH 语句命令结束后, 开始下一个移动命令时 (如果仅 PUSH 语句的命令结束, 则无法解除)

[PUSH 语句命令结束时处于推进状态]

对象轴开始向 PUSH 语句中指定的移动方向的相反方向移动时

[PUSH 语句命令结束时移动至目标位置 (未处于推进状态时)]

对象轴开始向任一方向移动时

· 伺服关闭时

· 断开控制器的电源然后重新接通时

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

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

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

坐标点指定的种类

- 直接数值指定
在<表达式>中直接指定马达位置
当马达位置的数值为整数时，判断为脉冲单位。
而当数值为实数值时，判断为毫米 / 度单位，每个轴从 0 脉冲位置向脉冲换算位置移动。

SAMPLE	说明
PUSH (1, 10000)	……机器人 1 中第 1 轴将从当前位置向 100000 脉冲处移动
PUSH [2] (2, 90.000)	……机器人 2 中第 2 轴将从当前位置向 90 度位置进行移动（旋转轴时）

- 坐标点定义指定
在<表达式>中指定坐标点表达式。使用<轴编号>中指定轴的数据。
当坐标点表达式为毫米 / 度单位时，每个轴从 0 脉冲位置向脉冲换算位置移动。

SAMPLE	说明
PUSH (1, P1)	……机器人 1 中第 1 轴将从当前位置向 P1 所指定的位置进行移动
PUSH [2] (2, P90)	……机器人 2 中第 2 轴将从当前位置向 P90 所指定的度数位置进行移动(旋转轴时)

选项种类

● 推压力指定

格式

 $F = \text{表达式}$

值

范围

 $\text{表达式} \quad -1000 \sim 1000 \text{ (单位 : \%)}$

含义

通过 < 表达式 > 指定相对于轴前进方向的推压力。

实际的推压力如下所示。

· 额定推力 $\times$ < 表达式 > / 100

当省略 < 推压力值 > 时，将使用通过参数设定的 < 推压力值 >。

SAMPLE

说明

$\text{PUSH}(1, 10000), F=10$ ……机器人 1 中第 1 轴以相当于额定推力 10% 的推压力，
从当前位置向 10000 脉冲处移动

● 推压时间指定

格式

 $\text{TIM} = \text{表达式}$

值

范围

 $\text{表达式} \quad 1 \sim 32767 \text{ (单位 : ms)}$

含义

通过 < 表达式 > 来指定以指定推压力持续进行推压的时间。

推压力达到设定值的状态持续 < 表达式 > 以上的时间时，PUSH 语句会结束。

当省略了本选项时，将使用参数的设定值。

SAMPLE

说明

$\text{PUSH}(1, 10000), \text{TIM}=5000$ ……机器人 1 中第 1 轴以按下时间 5 秒，
从当前位置向 10000 脉冲处移动

● 推进速度指定

格式

1. SPEED = 表达式
2. S = 表达式

值

范围

表达式 1 ~ 100 (单位 : %)

含义

使用<表达式>指定程序移动速度。
仅对指令的 PUSH 语句有效。

实际速度如下所示。

· 机器人的最高速度 (mm/ 秒或度 / 秒) × 推进速度比率 (%) × 自动移动速度 (%) × 表达式 (%)

SAMPLE

说明

PUSH (1, 10000) , S=10 ……第 1 轴以相当于推进速度比率与自动移动速度之积 10% 的速度,
从当前位置向 100000 脉冲处移动

格式

1. DSPEED = 表达式
2. DS = 表达式

值

范围

表达式 0.01 ~ 100.00 (单位 : %)

含义

使用<表达式>指定轴的移动速度。
仅对指令的 PUSH 语句有效。
不受自动移动速度值 (%) 的影响, 一贯按照 DSPEED 的<表达式>值 (%) 移动。

实际速度如下所示。

· 机器人的最高速度 (mm/ 秒或度 / 秒) × 推进速度比率 (%) × 表达式 (%)

SAMPLE

说明

PUSH (1, 10000) , DS=0.1 ……第 1 轴以相当于推压移动速度 0.1% 的速度, 从当前位置向 100000 脉冲处移动

PUSH

● STOPON 条件指定

格式	
STOPON 条件表达式	
SAMPLE	说明
PUSH (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

参照 "MOVE" 命令

● CONTON 条件指定

格式	
CONTON 条件表达式	
SAMPLE	说明
PUSH (1,10000) , CONTON 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，则开始减速并执行下一个步骤

参照 "MOVE" 命令

相关命令	PSHFRC, PSHTIME, PSHMTD, PSHDSP, PSHSPD, PSHRSLT, CURTRQ, CURTQST
------	---

RADDEG

单位转换（弧度→度）

联机



8

函数

格式	
RADDEG (表达式)	
值	内容
表达式	角度（单位：弧度）

含义 将<表达式>的值转换为度。

SAMPLE	说明
LOC4 (P0) =RADDEG (ATN (B))	将变量 B 的反正切值转换为度，并赋值给 P0 的 R 数据

相关命令	ATN, COS, DEGRAD, SIN, TAN
------	----------------------------

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

REFTSK

获取任务状态

联机



函数

格式	
REFTSK (任务编号)	
值	范围
任务编号	0 ~ 16 (可以指定变量)
返回值	范围、含义
任务状态	-1 : NON EXISTENT (未录入状态)
	0 : STOP (停止状态)
	1 : READY, RUN (可执行状态、执行状态)
	2 : WAIT (等待状态)
	3 : SUSPEND (强制待机状态)

含义 获取指定任务的状态。
获取任一返回值。

通过使用此函数以编程方式获取任务状态，可以根据任务状态执行处理（例如：任务启动）。
也可以使用联机命令来执行。

参照 关于任务状态，请参阅第 6 章“3. 任务的状态与变化”。

SAMPLE 1: 启动任务 2	
A% = REFTSK (2)	……………获取任务 2 的状态
IF A% = -1 THEN	……………如果任务 2 未录入状态，启动任务 2
START <SUB_PGM>, T2	
ELSE	
‘未启动任务 2 时的处理	
ENDIF	
SAMPLE 2: 任务 2 的处理完成后，将执行自任务的处理。	
WAIT REFTSK (2) <= 0	……………等待任务 2 进入停止或未录入状态
‘下面，自任务的处理	

REM

插入注释

联机

×

8

命令

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。
REM 或 " ' " 也可写入行的中间。

SAMPLE	说明
	以下均为注释示例
REM *** MAIN PROGRAM ***	注释：*** MAIN PROGRAM ***
' *** SUBROUTINE ***	注释：*** SUBROUTINE ***
HALT ' HALT COMMAND	注释：HALT COMMAND

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

RESET

关闭指定端口的比特或清除变量

联机



命令

格式 1		
RESET	DOm (b, . . . ,b) DO (mb, . . . ,mb) MOm (b, . . . ,b) MO (mb, . . . ,mb) TON (b, . . . ,b) TO (nb, . . . ,nb) LON (b, . . . ,b) LO (nb, . . . ,nb) SOM (b, . . . ,b) SO (mb, . . . ,mb)	

格式 2		
RESET TCOUNTER		

标注	值	范围
m	端口编号	2 ~ 7, 10 ~ 17, 20 ~ 27, 30 ~ 37
n	端口编号	0、1
b	比特编号 参照 请参阅第 3 章“10 指定比特”	0 ~ 7（如果省略，则将所有 8 个比特作为对象） 指定多个比特时，从左边开始降序（大→小）记述



注意 DO、SO 均无法输出至端口 0 和 1。

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)。

SAMPLE	说明	
RESET DO2 ()	DO (27 ~ 20)	关闭
RESET DO2 (6, 5, 1)	DO (26, 25, 21)	关闭
RESET (37, 35, 27, 20)	DO (37, 35, 27, 20)	关闭
RESET TCOUNTER		清除 10ms 计数器变量

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

RESTART

重新启动暂停中的其他任务

联机

✓

8

命令

格式		
RESTART	Tn <程序名称> PGm	
标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100

含义 重新启动暂停中 (SUSPEND 状态) 的其他任务。
任务也可通过正在执行的 <程序名称> 或者编号进行指定。
程序名两侧添加 < > (尖括号)。

MEMO

指定并执行了非暂停中的任务 (程序) 时将发生错误。

SAMPLE	说明
START <SUB_PGM>, T2 FLAG=1 *L0: IF FLAG=1 AND DI2 (0) =1 THEN SUSPEND T2 FLAG=2 WAIT DI2 (0) =0 ENDIF IF FLAG=2 AND DI2 (0) =1 THEN RESTART T2 FLAG=1 WAIT DI2 (1) =0 ENDIF MOVE P, P0 MOVE P, P1 GOTO *L0 HALTALL	主任务 ……将 SUB_PGM 程序登入到任务 2 并启动 ……排他控制标志 1 执行设置 ……标志 1 及 DI2 (0) 为 ON 时 ……暂停任务 2 ……将 2 赋值到 FLAG (对排他控制标志 2 进行执行设置) ……等待 DI2 (0) 变为 OFF ……标志 2 及 DI2 (0) 为 ON 时 ……重新启动任务 2 ……排他控制标志 1 执行设置 ……等待 DI2 (1) 变为 OFF
' SUBTASK ROUTINE *SUBTASK: DO2 (0) =1 DELAY 1000 DO2 (0) =0 DELAY 1000 GOTO *SUBTASK EXIT TASK	程序名称 : SUB_PGM (副任务) ……结束任务

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

参照 详细内容, 请参阅第 6 章“多任务”

RESUME

错误恢复处理后恢复执行程序

联机

×

命令

形式

1. RESUME
2. RESUME NEXT
3. 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。

MEMO

- 无法在错误处理例程以外执行 RESUME 语句。
- 发生了[17.800：马达超负荷]等重大错误时，无法执行错误恢复处理。

参照 详细说明，请参阅“ON ERROR GOTO”。

相关命令	ON ERROR GOTO
------	---------------

命令

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[5.212：堆栈溢出]等错误。

SAMPLE	说明
*ST:	主程序 (标签定义)
MOVE P, P0	
GOSUB *CLOSEHAND	跳至子例程 *CLOSEHAND
MOVE P, P1	
GOSUB *OPENHAND	跳至子例程 *OPENHAND
GOTO *ST	跳至主程序 *ST
HALT	
' SUB ROUTINE	(注释) 以下为子例程
*CLOSEHAND:	子例程 *CLOSEHAND 标签定义
DO (20) = 1	DO (20) 变为 ON 后
RETURN	从 GOSUB 语句 *CLOSEHAND 的下一行开始执行
*OPENHAND:	子例程 *OPENHAND 标签定义
DO (20) = 0	DO (20) 变为 OFF 后
RETURN	从 GOSUB 语句 *OPENHAND 的下一行开始执行

相关命令

GOSUB

RIGHT\$

从右端抽出字符串

联机



函数

格式	
RIGHT\$ (字符串表达式, 表达式)	
值	范围
表达式	0 ~ 255

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 ~ 255 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	说明
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\$
------	---------------

RIGHTY

将水平多关节机器人的手系统设置为右手系统

联机



8

命令

格式

RIGHTY [机器人编号]

值

范围

机器人编号 1 ～ 4（省略时，机器人 1 被指定。）

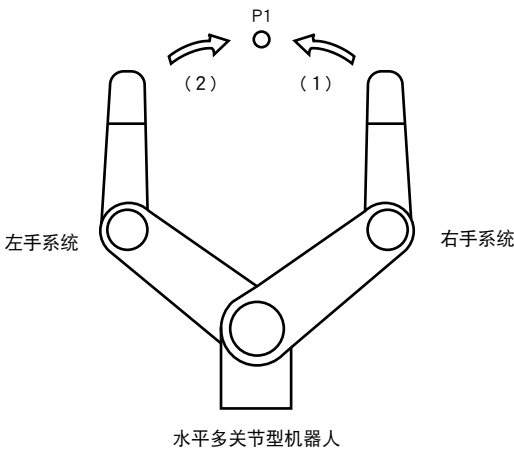
含义 选择通过右手系统使<机器人编号>指定机器人向正交坐标系指定坐标点移动。
即使执行了命令仅选择也无法使机器人移动。轴在动作时，请等待动作结束（定位至公差内）再执行。
此命令只对水平多关节机器人有效。

SAMPLE

说明

RIGHTY 将机器人 1 设置为右手系统（下图中的 1）
MOVE P, P1
LEFTY 将机器人 1 设置为左手系统（下图中的 2）
MOVE P, P1
RIGHTY
HALT

SAMPLE: LEFTY / RIGHTY



相关命令

LEFTY

RSHIFT

比特的右位移

联机



函数

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	说明
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

SCRINR

设置 / 获取水平多关节 R 轴惯性力矩参数

联机



8

命令

格式

SCRINR [机器人编号] 表达式

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

表达式 水平多关节 R 轴惯性力矩参数的值
0 ~ 32767 (单位 : $\text{kgm}^2 \times 10^{-4}$)

含义 将指定机器人的水平多关节 R 轴惯性力矩参数变更为 <表达式> 的值。

SAMPLE

说明

SCRINR 100将机器人 1 的水平多关节 R 轴惯性力矩参数变更为 100
SCRINR[2] 200将机器人 2 的水平多关节 R 轴惯性力矩参数变更为 200

函数

格式

SCRINR [机器人编号]

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

含义 获取指定机器人的水平多关节 R 轴惯性力矩参数值。

SAMPLE

说明

A% = SCRINR将机器人 1 的水平多关节 R 轴惯性力矩参数赋给变量 A%
A% = SCRINR[2]将机器人 2 的水平多关节 R 轴惯性力矩参数赋给变量 A%

SELECT CASE ~ END SELECT

执行由表达式的值指定的命令区

联机

×

命令

形式

```
SELECT CASE 表达式
  CASE 表达式列表 1
    命令区 1
  CASE 表达式列表 2
    命令区 2
  :
  CASE ELSE
    命令区 n
END SELECT
```

值

范围

表达式

通用输入、变量

表达式的列表

十进制的数值或字符串。通过用逗号 (,) 隔开可指定多个数据。

命令区

记述被执行的命令。以 1 行 1 命令记述。



MEMO

DI(24,23,22,21,20) : 以 2 进制识别被指定的 5 位分的 I/O, 执行相关表达式列表内的命令区。

ABC% : 指定变量。根据变量值, 执行相关表达式列表内的命令区。

含义

执行由<表达式>的值指定的命令区。设置方法如下所示。

- 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。
- <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。
- <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。
- <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

说明

```
WHILE 1
  SELECT CASE DI3(7,6,5,4,3,2,1,0) .....在 DI3 (8 位) 的输入状态下执行各命令区
    CASE &B00000001 .....DI3 (0) 为 ON 时
      MOVE P,P1
    CASE &B00000010 .....DI3 (1) 为 ON 时
      MOVE P,P2
    CASE &B00000011 .....DI3 (0) 和 DI3 (1) 为 ON 时
      MOVE P,P3
    CASE ELSE .....上述以外时
      MOVE P,P0
  END SELECT .....退出 SELECT CASE 语句
WEND
```

SEND

将读出文件的数据转发至写入文件

联机

✓

8

命令

格式

SEND 读出文件 TO 写入文件

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。

参照 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。(※1)
- 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。(※2)



补充

※1) 写入读出专用文件时的错误示例

- SEND CMU TO DIR
- SEND PNT TO SI()

※2) 数据格式不匹配的错误示例

- SEND PGM TO PNT
- SEND SI() TO SFT

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	<bbbbbbbbb> PGn	✓	✓
	坐标点	PNT	Pn	✓	✓
	坐标点注释	PCM	PCn	✓	✓
	坐标点名称	PNM	PNn	✓	✓
	参数	PRM	/cccccccc/ #cccccccc# \cccccccc\	✓	✓
	位移坐标定义	SFT	Sn	✓	✓
	机械手定义	HND	Hn	✓	✓
	工件定义	WRKDEF	Wn	✓	✓
	托盘定义	PLT	PLn	✓	✓
	通用以太网端口	GEP	GPn	✓	✓
	输入输出名称	ION	iNMn(n)	✓	✓
	区域判定输出	ACO	ACn	✓	✓
	实时输出	RTO	RTn	✓	✓
变量、常量	变量	VAR	ab...by	✓	✓
	数组变量	ARY	ab...by(x)	✓	✓
	常量	————	“cc...c”	✓	×

分类	文件名	记述格式		读出	写入
		全部	个别文件		
状态	参数目录	DPM	_____	✓	×
	机器参照 (传感器、撞块)	MRF	_____	✓	×
	机器参照 (标记)	ARP	_____	✓	×
	累计数据 (*)	ACD	_____	✓ (*)	×
	系统构成信息	CFG	_____	✓	×
	版本信息	VER	_____	✓	×
	选件板	OPT	_____	✓	×
	自我诊断	SCK	_____	✓	×
	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()	SO n()	✓	✓
	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 ✓ : 可以
x : 表达式 (数组自变量 y : 变量类型 i : 输出输入类型 × : 不可以

*) 累计数据 : 在软件版本 Ver.1.73 以上有效 累计数据的读出 : 在软件版本 Ver.2.04 以上有效

MEMO

- 中途停止 SEND 语句并再次启动时，请注意以下事项。
- 1. 从 RS-232C / Ethernet 读入时 (SEND CMU TO XXX / SEND ETH TO XXX)
在读出接收缓冲区半途状态下中途停止时，已获得的数据将被损坏。
 - 2. 写入 RS-232C / Ethernet 时 (SEND XXX TO CMU / SEND XXX TO ETH)
在写入发送缓冲区半途状态下中途停止时，将再次从起始开始写入数据。

SAMPLE	说明
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“数据文件详细说明”。

相关命令 OPEN, CLOSE, SETGEP, GEPSTS

命令

格式		
SERVO [机器人编号]	ON OFF FREE	(轴番号)
值	范围	
机器人编号	1 ～ 4 (省略时, 机器人 1 被指定。)	
轴编号	1 ～ 6 (当未指定时, 则以所有全轴作为对象。)	

含义 控制指定编号的轴或所有轴的伺服 ON / OFF。

		伺服	马达电源	动态制动	电磁制动器
SERVO 命令	ON	ON	ON	OFF	OFF
	OFF	OFF	OFF 仅当所有轴伺服关闭时	ON	ON
	FREE	OFF	继续上一个轴	OFF	OFF



注意

- 请勿在使用 SERVO OFF 语句并关闭马达电源状态下进行机器人动作范围内的作业。请务必在紧急停止电路开启状态下进行机器人动作范围内的作业。
- 电磁制动器是指, 为防止垂直轴下落而安装在制动规格机器人上的制动器。
如果对垂直轴执行 SERVO FREE 命令, 则可能解除其制动而使轴下落, 从而发生危险。

MEMO

- 此命令语句将在指定机器人的所有轴动作结束后 (在公差范围内定位后) 执行。
- 马达电源是指, 位于控制器内部、为机器人 (马达) 供电的电源单元。
- 动态制动是指, 使用伺服断电时马达产生的电力, 对马达进行制动的方式。

SAMPLE	说明
SERVO ON	……开启所有轴的伺服
SERVO OFF	……针对关闭所有轴伺服且附带制动器的轴, 其制动器将被锁定
SERVO FREE (3)	……第 3 轴 (Z 轴) 伺服禁止, 并解除制动

SET

开启指定端口的比特

联机

×

命令

格式		
SET	DOm (b, . . . ,b) DO (mb, . . . ,mb) MOm (b, . . . ,b) MO (mb, . . . ,mb) TO n (b, . . . ,b) TO (nb, . . . ,nb) LO n (b, . . . ,b) LO (nb, . . . ,nb) SOm (b, . . . ,b) SO (mb, . . . ,mb)	, 时间

标注	值	范围
m	端口编号	2 ~ 7, 10 ~ 17, 20 ~ 27
n	端口编号 (TO, LO)	0、1
b	比特编号 参照 请参阅第 3 章“10 指定比特”	0 ~ 7 (如果省略, 则将所有 8 个比特作为对象) 指定多个比特时, 从左边开始降序 (大→小) 记述
	时间	10 ~ 3600000 (单位 : ms)



注意
DO、SO 均无法输出至端口 0 和 1。

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。
指定了硬件上不存在的端口时, 将无任何输出。

SAMPLE	说明
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

SETGEP

设置通用以太网端口

联机



8

命令

格式

SETGEP m, n, “IP 地址”, ppppp, e, t

标注	值	范围、含义
m	通用以太网编号	0 ~ 7
n	模式	0 : 服务器、1 : 客户端
	IP 地址	0.0.0.0 ~ 255.255.255.255
ppppp	端口编号	0 ~ 65535
e	终端字符编码	0 : CRLF、1 : CR
t	端口类型	0 : TCP

- 含义** 设置指定的通用以太网端口。
- 对于已设置的通用以太网端口，可通过 OPEN 操作打开通信端口、通过 CLOSE 操作关闭通信端口。
< IP 地址 > 的前后应使用 “ ” (双引号)。
- 选择服务器模式时，可省略 IP 地址，但需要在 IP 地址前后加上 “ ” (双引号)。

MEMO

- 模式选择服务器时**
- IP 地址：使用控制器中设置的 IP 地址进行通信，因此无需设置 IP 地址 (在 IP 地址中设置的内容无效)。
 - 端口编号：设置与控制器设置内容不同的编号。
- 模式选择客户端时**
- IP 地址 / 端口编号：设置待连接服务器的 IP 地址和端口编号。

SAMPLE	说明
IPADRS\$= “192.168.0.100”	……………待连接服务器的 IP 地址 (192.168.0.100) 赋给变量 IPADRS\$
SETGEP 1, 1, IPADRS\$, 100, 0, 0	……………通用以太网编号 1 的设置如下： · 模式：客户端 · 待连接服务器的 IP 地址：192.168.0.100 · 待连接服务器的端口编号：100 · 终端字符编码：CRLF
OPEN GP1	……………连接通用以太网端口编号 1 所指定的服务器
SEND “123” TO GP1	……………从通用以太网端口编号 1 发送字符串 “123”
CLOSE GP1	……………断开与通用以太网端口编号 1 所指定的服务器的连接

相关命令 OPEN, CLOSE, SEND, GEPSTS

SGI

获取指定的整数型静态变量，或将值赋给该变量

联机



命令

格式		
1. SGI n=xxxxxx 2. SGI(n)=xxxxxx		
标注	值	范围
n	整数型静态变量编号	0 ~ 31
		- 2147483648 ~ 2147483647 的整数值
含义 将 xxxxxx 赋给 n 中指定的整数型静态变量 (SGI)。 如果 xxxxxx 设为带小数点的实数值，则赋值舍去小数点以下部分后的值。		
SAMPLE	说明	
SGI1=300	………为 SGI1 赋值 300	

函数

格式		
1.SGI n 2.SGI(n)		
标注	值	范围
n	整数型静态变量编号	0 ~ 31
含义 获取 n 中指定的整数型静态变量 (SGI) 的值。		
SAMPLE		说明
A%= SGI1		………将 SGI1 的值赋给变量 A%
相关命令		SGR

SGR

获取指定的实数型静态变量，或将值赋给该变量

联机



8

命令

格式		
1.SGRn=xxxxxx 2.SGR(n)=xxxxxx		
标注	值	范围
n	实数型静态变量编号	0 ~ 31
xxxxxx	1. 单精度实数	从 -999,999.9 到 +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	$-1.0 \times 10^{38} \sim +1.0 \times 10^{38}$ 尾数部分的整数与小数加在一起 7 位数以下

含义 将 xxxxxx 赋给 n 中指定的实数型静态变量 (SGR)。

SAMPLE	说明
SGR1=1320.355	………为 SGR1 赋值 1320.355

函数

格式		
1.SGRn 2.SGR(n)		
标注	值	范围
n	实数型静态变量编号	0 ~ 31

含义 获取 n 中指定的实数型静态变量 (SGR) 的值。

SAMPLE	说明
A!= SGR1	………将 SGR1 的值赋给变量 A!

相关命令	SGI
------	-----

SHARED

使不提交变量便可在子过程中参照

联机

×

命令

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。



补充
程序级别代码指的是子过程以外编写的程序。

MEMO

- 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	说明
DIM Y! (10) X!=2. 5 Y! (10) =1. 2 CALL *DISTANCE CALL *AREA HALT SUB *DISTANCE SHARED X!,Y! () PRINT X!^2+Y! (10) ^2 END SUB SUB *AREA DIM Y! (10) PRINT X!*Y! (10) END SUB	 通过 SHARED 声明变量的参照共享变量 不共享变量

相关命令

SUB, END SUB

SHIFT

设置位移坐标

联机



8

命令

格式		
SHIFT [机器人编号]	位移变量 OFF	
值	范围	
机器人编号	1 ~ 4 (省略时, 机器人 1 被指定。)	

含义 根据<位移变量>指定的位移数据对<机器人编号>指定机器人设定位移坐标。
如果设置为 OFF, 将不通过位移变量进行坐标转换。

MEMO

- 此命令语句在轴的动作结束后 (定位在公差范围内后) 执行。
- OFF 时, 与设置 X 偏移为 0.000、Y 偏移为 0.000、Z 偏移为 0.000、旋转方向偏移为 0.000 的位移变量时相同。

SAMPLE	说明
SHIFT S1	将机器人 1 的换挡坐标改变为换挡 1
MOVE P, P10	
SHIFT S [A]	将机器人 1 的移位坐标更改为由变量 A 指定的移位坐标
MOVE P, P20	
HALT	

相关命令	位移定义语句、位移赋值语句
------	---------------

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

SI

获取指定 SI 的状态

联机



函数

格式		
LET 式 = SI(m(b, . . . , b)		
LET 式 = SI (mb, . . . , mb)		
标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号	0 ~ 7 (如果省略, 则将所有 8 个比特作为对象) 指定多个比特时, 从左边开始降序 (大→小) 记述

含义 获取 m 中指定的 SI 端口的输入状态。

SAMPLE	说明
A%= SI2 ()	将 SI (27) 到 SI (20) 的输入状态赋值给变量 A%
A%= SI0 (6, 5, 1)	将 SI (06) , SI (05) , SI (01) 的输入状态赋值给变量 A% (上述信号都变为 1 (ON) 时, A%=7)
A%=SI (37, 35, 27, 10)	将 SI (37) , SI (35) , SI (27) , SI (10) 的输入状态赋给变量 A% (上述信号的 SI (27) 以外都变为 1 (ON) 时, A%=13)

SID

获取指定串行输入的文字信息 (双字)

联机



8

函数

格式

LET SID(m)

标注

值

范围

m

端口编号

2, 4, 6, 8, 10, 12, 14 *

* 在 SIOW 扩大有效的情况下, 端口编号的范围为 2, 4, 6 ~ 14, 24, 26, 28 ~ 126。

详细内容, 请参阅别册 RCX3 《用户手册》“与选件板相关的参数”。

含义

获取 m 中指定的 SID 端口的值。

获取范围为 -2,147,483,648(&H80000000) ~ 2,147,483,647(&H7FFFFFFF)。



MEMO

- 作为带符号的双字数据处理。
- 不存在指定端口时, 将输入 0。
- 配置数据时, 低位端口编号的数据配置到低位地址。
例如, SOW(2)=&H2345, SOW(3)=&H0001 时, SOD(2)=&H00012345。

SAMPLE

说明

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%

相关命令

SIW

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

SIN

计算正弦值

联机



函数

格式	
SIN (表达式)	
值	内容
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	说明
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
------	-------------------------------

函数

格式

LET SIW(m)

标注	值	范围
m	端口编号	2 ~ 15 *

* 在 SIOW 扩大有效的情况下，端口编号的范围为 2 ~ 15、24 ~ 127。
详细内容，请参阅别册 RCX3《用户手册》“与选件板相关的参数”。

含义 将值获取到在 m 中指定的 SIW 端口。
获取范围为 0(&H0000) ~ 65,535 (&HFFFF)。

MEMO

- 作为无符号的单字数据处理。
- 不存在指定端口时，将输入 0。

SAMPLE	说明
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%

相关命令 SID

Sn

定义位移坐标

联机



命令

格式	
Sn = x y z r	
值	范围
n	0 ~ 39
x, y, z, r	-99,999.99 ~ 99,999.99

含义	定义用于使机器人动作的坐标位移时的位移坐标。 坐标单位仅限毫米(不可使用脉冲)。 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。
----	--



补充	- 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 [9.706: 位移校验和错误] 等存储器相关错误。
----	---

SAMPLE	说明
S0 = 0.000 0.000 0.000 0.000	…定义 S0 的位移坐标
S1 =100.000 200.000 50.000 90.000	…定义 S1 的位移坐标
P3 =100.000 0.000 0.000 0.000 0.000 0.000	…定义 P3 的坐标点数据
SHIFT S0	…将机器人 1 位移坐标变更为 S0
MOVE P, P3	…PTP 移动至 P3
SHIFT S1	…将机器人 1 位移坐标变更为 S1
MOVE P, P3	…PTP 移动至 P3
HALT	

相关命令	位移赋值语句、SHIFT
------	--------------

命令

格式

1.LET SOm (b, . . . ,b) = 表达式

2.LET SO (mb, . . . ,mb) = 表达式

标注	值	范围
m	端口编号	2 ~ 7, 10 ~ 17, 20 ~ 27
b	比特编号 参照 请参阅第 3 章“10 指定比特”	0 ~ 7（如果省略，则将所有 8 个比特作为对象） 指定多个比特时，从左边开始降序（大→小）记述

表达式

超出整数值(指定实数时,将小数点以下四舍五入后转换为整数)
赋值目标所需比特数的比特将被舍去。
(如果指定端口,则使用低位 8 比特。
如果在比特编号中指定的比特数为 1 ~ 8 比特,
则使用低位 1 ~ 8 比特)

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。
指定了硬件上不存在的端口时，将无任何输出。



注意
无法输出至 SO0 () 与 SO1 ()。仅限参照。

SAMPLE	说明
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)

函数

格式

1.LET 表达式 = SOm (b, . . . ,b)

2.LET 表达式 = SO (mb, . . . ,mb)

含义

表示指定 SO 端口的输出状态。

SAMPLE	说明
A%= SO2 ()	将 SO (27) 到 SO (20) 的输出状态赋值给变量 A%
A%= SO0 (6, 5, 1)	将 SO (06) , SO (05) , SO (01) 的输出状态赋值给变量 A% (上述信号均为 1 (ON) 时, 将变为 A%=7)
A%=SO (37, 35, 27, 10)	将 SO (37) , SO (35) , SO (27) , SO (10) 的输出状态赋值给变量 A% (上述信号中, 除 SO (27) 以外均为 1 (ON) 时, 将变为 A%=13)

相关命令

RESET, SET

SOD

输出指定串行输出的文字信息（双字），或获取输出状态

联机



命令

格式

LET SOD(m) = 表达式

标注	值	范围
m	端口编号	2, 4, 6, 8, 10, 12, 14 *

* 在 SLOW 扩大有效的情况下，端口编号的范围为は 2, 4, 6 ~ 14, 24, 26, 28 ~ 126。
详细内容，请参阅别册 RCX3《用户手册》“与选件板相关的参数”。

含义 向 m 中指定的 SOD 端口输出值。
表达式中指定的输出范围为 -2,147,483,648(&H80000000) ~ 2,147,483,647 (&H7FFFFFFF)。

MEMO

- 作为带符号的双字数据处理。
- 实际不存在串行端口时，将不输出到外部。
- 配置数据时，低位端口编号的数据配置到低位地址。
例如，SOW(2)=&H2345, SOW(3)=&H0001 时，SOD(2)=&H00012345。

SAMPLE	说明
SOD(2) = &H12345678	……向 SOD(2) 输出 &H12345678
SOD(4) = 1048575	……向 SOW(4) 输出 1048575 (&HFFFFFF)
SOD(2) = A%	……向 SOW(2) 输出变量 A% 的值

函数

格式

LET 表达式 = SOD(m)

含义 获取 m 中指定的 SOD 端口的输出状态。

SAMPLE	说明
A% = SOD(2)	……将 SOD(2) 的输出状态赋值给变量 A%

相关命令 SOW

SOW

输出指定串行输出的文字信息，或获取输出状态

联机



8

命令

格式

LET SOW(m) = 表达式

标注

值

范围

m

端口编号

2 ~ 15 *

表达式

超出整数值（指定实数时，将小数点以下四舍五入后转换为整数）
赋值目标所需比特数的比特将被舍去。
（如果指定端口，则使用低位 8 比特。
如果在比特编号中指定的比特数为 1 ~ 8 比特，则使用低位 1 ~ 8 比特）

* 在 SLOW 扩大有效的情况下，端口编号的范围为 2 ~ 15、24 ~ 127。
详细内容，请参阅别册 RCX3《用户手册》“与选件板相关的参数”。

含义

向 m 中指定的 SOW 端口输出值。
表达式中指定的输出范围为 0 (&H0000) ~ 65,535 (&HFFFF)。
但是，输出了负值时，将输出转换为十六进制的低位文字信息。
例：“SOW(2) = - 255”是指，将 - 255 (&HFFFFFF01) 的内容赋给 SOW(2)
- 255 是负值，将赋予低位的文字信息 (&HFF01)



MEMO

- 作为无符号的单字数据处理。
- 实际不存在串行端口时，将不输出到外部。
- 赋予了超出输出范围的数值时，将输出低位的 2 字节信息。

SAMPLE

说明

SOW (2) = &H0001向 SOW (2) 输出 &H0001
SOW (3) = 255向 SOW (3) 输出 255 (&H00FF)
SOW (15) = A%向 SOW (15) 输出变量 A% 的内容。
变量 A% 的数值超过输出范围时，输出低位文字信息

函数

格式

LET 表达式 = SOW(m)

含义

获取 m 中指定的 SOW 端口的输出状态。

SAMPLE

说明

A%=SOW (2)将 SOW (2) 的输出状态赋值给变量 A%

相关命令

SOD

SPEED

更改程序移动速度

联机



命令

格式	
SPEED [机器人编号] 表达式	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
表达式	1 ～ 100（单位：％）

含义 将<机器人编号>指定机器人的程序移动速度变更为<表达式>指定的值。
对指定机器人的所有轴(包括机器人构成轴和附加轴)的全部速度进行变更。

动作速度由手持编程器的操作或 ASPEED 命令设置的自动移动速度与
SPEED 命令指定的程序移动速度的乘积决定。

动作速度 = 自动移动速度 × 程序移动速度

示例：

自动移动速度80%

× 程序移动速度50%

移动速度 = 40% (80%×50%)



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

SAMPLE	说明
ASPEED 100	………变更机器人 1 的自动运行速度为 100％
SPEED 70	………变更机器人 1 的程序运行速度为 70％
MOVE P, P0	………按照 70％（=100*70）的速度从当前位置向 P0 移动
SPEED 50	………变更机器人 1 的程序运行速度为 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
------	--------

SQR

计算平方根

联机



8

函数

格式

SQR (表达式)

值

范围

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

说明

A=SQR (X^2+Y^2)

.....将 X^2+Y^2 的平方根赋值给变量 A

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

START

启动新任务

联机



命令

格式		
START	<程序名称> PGm	,Tn, p
标注	值	范围
m	程序编号	1 ~ 100
n	任务编号	1 ~ 16
p	任务优先级	1 ~ 64

- 含义
- 根据任务 n、优先顺序 p 启动指定程序。

 - 当省略任务编号 n 时，未启动任务中编号最小的任务将会被自动指定。
 - 当省略任务优先级 p 时，将默认为 32。
数字越小，优先级越高；数字越大，优先级越低（高 1 ~ 低：64）。
 - 当任务优先级较高的任务处于 RUNNING 状态时，优先级较低的任务也将保持着 READY 状态。
 - 程序名两侧添加 < >（尖括号）。

SAMPLE	说明
START <SUB_PGM>, T2 *ST: MOVE P, P0, P1 GOTO *ST HALT 程序名称 : SUB_PGM 'SUBTASK ROUTINE *SUBTASK: P100 = WHERE IF LOC3 (P100) > 10000 THEN DO (20) = 1 ELSE DO (20) = 0 ENDIF GOTO *SUBTASK EXIT TASK	…将“SUB_PGM”程序登入到任务 No. 2 并启动 …定义主程序 *ST (注释) 副任务 例程 …定义任务程序 *SUBTASK …退出任务程序（当前任务：SUB_PGM）

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

STR\$

将数值转换为字符串

联机



8

函数

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

SUB ~ END SUB

定义子过程

联机

X

命令

形式

```
SUB 标签 (形参, 形参 ...)
    命令区
END SUB
```

含义

定义子过程。

通过 CALL 语句可执行子过程。执行了 END SUB 语句后，将跳转至调出的 CALL 语句的下一个命令。定义如下所示。

1. 子过程中声明的变量均为局部变量，仅在子过程内有效。每次调出子过程时，局部变量均将被初始化。
2. 使用全局变量（程序级别）时，使用 SHARED 语句。
3. 进行变量的提交时，使用 <形参>。<形参>通过逗号 (,) 分隔。
4. <形参>中变量名与所有数组（数组名的后面带有 ()）有效。
如果指定了数组的元素（数组名后面带有 (<角标>)），将出错。



MEMO

- 在子过程内无法定义子过程。
- 虽然在子过程内可定义标签，但是通过 GOTO 或 GOSUB 等语句无法跳转至子过程以外的标签。
- 局部变量无法在 PRINT 语句、SEND 语句中使用。

SAMPLE 1

说明

```
A=1
CALL *TEST          .....调用子过程 *TEST
PRINT A
HALT
' SUB ROUTINE: TEST .....（注释：将子过程命名为“TEST”）
SUB *TEST           .....定义子过程 *TEST（※）
    A=50
END SUB             .....退出子过程的定义
```

※ 视作与以上使用的 A 不同的变量



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	说明
X% = 4	
Y% = 5	
CALL *COMPARE (REF X%, REF Y%)	……调用（引用传递）子过程 *COMPARE
PRINT X%,Y%	……X% 的值变更为 5、Y% 的值变更为 4
Z% = 7	（在子过程内变更值 => 变量的值变更）
W% = 2	
CALL *COMPARE (REF Z%, REF W%)	……调用（引用传递）子过程 *COMPARE
PRINT Z%,W%	……Z% 的值保持 7、W% 的值保持 2（值不变）
HALT	（子过程内的比较变成假，不执行指令）
' SUB ROUTINE: COMPARE	…（注释：将子过程命名为“COMPARE”）
SUB *COMPARE (A%, B%)	……子过程 *COMPARE 定义
IF A% < B% THEN	
TEMP% = A%	
A% = B%	
B% = TEMP%	
ENDIF	
END SUB	……退出子过程的定义

MEMO

在上述示例中，各个不同变量作为自变量提交，并将子过程调出 2 次。

相关命令	CALL, EXIT SUB, SHARED
------	------------------------

SUSPEND

暂停执行中的其他任务

联机

×

命令

格式		
SUSPEND	Tn <程序名称> PGm	
标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100

含义 暂停正在执行的其它任务。

- 1. 任务也可通过正在执行的程序名称或者编号进行指定。
- 2. 也可用于比当前任务优先级更高的任务。
- 3. 程序名两侧添加< > (尖括号)。

MEMO

当指定并执行了未启动的任务(程序)时将会发生错误。

SAMPLE	说明
START <SUB_PGM>, T2 SUSFLG=0 *L0: MOVE P, P0 MOVE P, P1 WAIT SUSFLG=1 SUSPEND T2 SUSFLG=0 GOTO *L0 HALT	主任务 ……将 SUB_PGM 登入到任务 2 并启动 ……将 SUSFLG 置为 OFF ……等待 SUSFLG 变为 ON ……暂停任务 2 ……将 SUSFLG 置为 OFF
' SUBTASK ROUTINE *SUBTASK: WAIT SUSFLG=0 DO2 (0) =1 DELAY 1000 DO2 (0) =0 DELAY 1000 SUSFLG=1 GOTO *SUBTASK EXIT TASK	副任务 (程序名称 : SUB_PGM) ……等待 SUSFLG 变为 OFF ……将 SUSFLG 置为 ON ……跳至 *SUBTASK ……退出任务 2 (当前任务 : *SUBTASK)

相关命令 CUT, EXIT TASK, RESTART, START

SWI

切换执行程序

联机

×

8

命令

格式

SWI <程序名称>

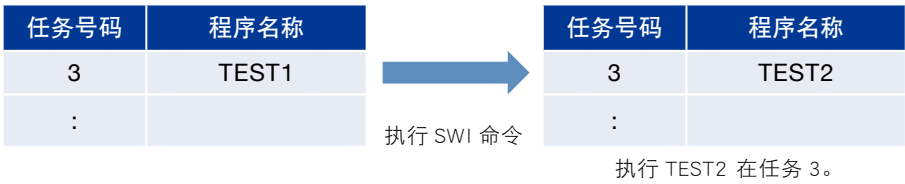
含义 从当前正在执行的程序切换至指定程序，并在执行编译后，从第 1 行开始执行。

1. 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。
2. 程序名两侧添加 < > (尖括号)。

MEMO

当指定程序不存在时，将会 发生 [3.203：程序不存在(代码：&H0003&H00CB)] 错误，并且停止运行。

SAMPLE	说明
MOVE P, P1 SWI <TEST2>	……………将执行程序切换至 TEST2



MEMO

切换程序的任务编号是在执行 SWI 语句的程序所登录的任务编号内进行切换。

TAN

计算正切值

联机



函数

格式	
TAN (表达式)	
值	含义
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	说明
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
------	-------------------------------

TCOUNTER

定时器、计数器

联机



8

函数

格式

TCOUNTER

含义 从复位 TCOUNTER 变量时开始，每隔 1ms 输出计数值。
值计数至 2,147,483,647 后，返回 0。

SAMPLE

说明

```
MOVE P, P0
WAIT ARM
RESET TCOUNTER
MOVE P, P1
WAIT ARM
A = TCOUNTER
PRINT TCOUNTER  ....在手持编程器上显示进入 P0 至 P1 公差为止的移动时间
```

相关命令

RESET

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

TIMES

获取当前时间

联机



函数

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	说明
A\$=TIMES	……以 hh : mm : ss 形式的字符串将当前时间赋值到字符型变量 A\$
PRINT TIMES	……以 hh : mm : ss 形式在手持编程器等设备上显示当前时间

相关命令 DATES, TIMER

TIMER

获取当前时间总秒数

联机



8

函数

格式

TIMER

含义 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。



注意 内部使用的时钟与实际时间之间可能会有误差。

SAMPLE

说明

```
A%=TIMER .....从上午 0 点开始以秒数的形式将当前时间赋值到整数型变量 A%
FOR B=1 TO 10
MOVE P,P0
MOVE P,P1
NEXT
A%=TIMER-A% .....将当前时间 -A% 赋值到 A%
PRINT A%/60;"：" ;A% MOD 60 .....显示 "A% ÷ 60 : A% ÷ 60"
HALT
```

相关命令

TIME\$

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

TO

将指定值输出至 TO 端口，或获取输出状态

联机



命令

形式		
1.LET TOM (b, . . . ,b) = 表达式 2.LET TO (mb, . . . ,mb) = 表达式		
标注	值	范围
m	端口编号	0、1
b	比特编号	0 ~ 7（如果省略，则将所有 8 个比特作为对象） 指定多个比特时，从左边开始降序（大→小）记述
	表达式	超出整数值（指定实数时，将小数点以下四舍五入后转换为整数） 赋值目标所需比特数的比特将被舍去。 （如果指定端口，则使用低位 8 比特。 如果在比特编号中指定的比特数为 1 ~ 8 比特，则使用低位 1 ~ 8 比特）

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。 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
TOO () = &B000000110

函数

形式
1.LET 表达式 = TOM (b, . . . ,b) 2.LET 表达式 = TO (mb, . . . ,mb)

含义	获取指定 TO 端口的输出状态。
----	------------------

SAMPLE	说明
A%=TOO ()	………将从 TO (07) 到 TO (00) 的输出状态赋值给变量 A%
A%=TOO (6, 5, 1)	………将 TO (06), TO (05), TO (01) 的输出状态赋值给变量 A% (上述信号均为 1 (ON) 时, 将变为 A%=7)
A%=TO (17, 15, 00)	………将 TO (17), TO (15), TO (00) 的输出状态赋值给变量 A% (上述信号中, 除 TO (15) 以外均为 1 (ON) 时, 将变为 A%=5)

相关命令	RESET, SET
------	------------

TOLE

设置 / 获取公差参数

联机

✓

8

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

命令

格式

- 1.TOLE [机器人编号] 表达式
- 2.TOLE [机器人编号] (轴编号) = 表达式

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

轴编号 1 ~ 6

表达式 因设置的马达而异 (单位: 脉冲)

含义 将<机器人编号>指定机器人轴的公差参数变更为<表达式>的值(单位: 脉冲)。

[形式 1] 对指定机器人的所有轴进行变更。

[形式 2] 仅对指定机器人的<轴编号>指定轴进行变更。

MEMO

此命令语句在指定轴的动作结束后(定位在公差范围内后)执行。

函数

格式

TOLE [机器人编号] (轴编号)

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

轴编号 1 ~ 6

含义 赐予轴的公差参数的值。

SAMPLE

```
' CYCLE WITH DECREASING TOLERANCE
DIM TOLE(5)
FOR A=200 TO 80 STEP -20
  GOSUB *CHANGE_TOLE
  MOVE P,P0
  MOVE P,P1
NEXT A
C=TOLE(2) .....将机器人 1 中第 2 轴的公差参数赋给变量 C。
HALT
*CHANGE_TOLE:
FOR B=1 TO 4
  TOLE(B)=A
NEXT B
RETURN
```

TORQUE

设置 / 获取最大扭矩指令值

联机



命令

格式

TORQUE [机器人编号] (轴编号) = 表达式

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

轴编号 1 ~ 6

表达式 1 ~ 100 (单位: %)

含义

将<机器人编号>指定机器人的<轴编号>指定轴的最大扭矩指令值变更为<表达式>的值。
更改后的值将在执行下一个 MOVE 语句或 DRIVE 语句等移动命令时有效。
参数值不更改。

扭矩限制的解除条件如下所示。

- 对同一轴执行 TORQUE 命令。
- 切断控制器电源并再次接通电源。
- 进行轴极性参数的变更或者参数的初始化。
- 变为伺服禁止状态。

对扭矩限制值进行暂时变更的条件如下。

- 原点复归执行中
- 正在执行 PUSH 命令执行中
- 仅移动方向 PUSH 命令可以变更扭矩
- 反方向的扭矩限制值的状态下无法变更

扭矩限制值以上许运行时可以进行暂时的变更, 但是运行结束后 MOVE 命令等的移动命令执行时会恢复 TORQUE 命令指定的扭矩设定值。



注意

- 如果指定的值太小, 轴可能无法移动。此时, 请务必设置为紧急停止状态后再进行作业。
- 指定值变为额定扭矩以下时, 碰撞到障碍物时也有可能不出错。



MEMO

- TORQUE 语句: 在与轴的旋转方向相反的方向上施加扭矩限制
- PUSH 语句: 仅在行驶方向上限制扭矩

函数

格式	
TORQUE [机器人编号] (轴编号)	
值	范围
机器人编号	1 ～ 4 (省略时, 机器人 1 被指定。)
轴编号	1 ～ 6

含义 赋予<机器人编号>指定机器人的<轴编号>指定轴的扭矩设定值。

SAMPLE	说明
TORQUE (1) = 50	………将机器人 1 中第 1 轴的最大扭矩变更为 50%。
DRIVE (1, P1)	………使机器人 1 中第 1 轴从当前位置向 P1 所指定的位置进行移动。 (在开始移动的同时, 最大扭矩被变更。)
WAIT ARM	………等待机器人 1 中第 1 轴的动作结束。
TORQUE (1) = 100	………使机器人 1 中第 1 轴的最大扭矩恢复原值 (100%)。
MOVE P, P0	………使机器人 1 从当前位置向 P0 所指定的位置进行移动。 (在开始移动的同时, 第 1 轴的最大扭矩恢复原值 (100%)。)

相关命令	CURTRQ, PUSH
------	--------------

TSKPGM

获取录入指定任务编号的程序编号

联机



函数

格式	
TSKPGM (任务编号)	
值	含义
任务编号	已获取程序编号的任务编号

含义 获取在 <任务编号> 所指定的任务中录入的程序编号。

SAMPLE	说明
A=TSKPGM(1)	………将任务 1 中录入的程序号码赋给变量 A

相关命令	PGMTSK, PGN
------	-------------

VAL

将字符串转换为数值

联机



8

函数

格式

TSKPGM (任务编号)

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、八进制形式、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。

标注形式		
整数标注	[十进制形式]	nnnn
	[十六进制形式]	&Hnnnn
	[八进制形式]	&Onnnn
	[二进制形式]	&Bnnnn
实数标注	[小数点标注]	nnn.nnn
	[指数形式]	nnEmm

SAMPLE

A=VAL ("&B100001")

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

WAIT

等待条件表达式成立

联机

✕

命令

格式

WAIT 条件表达式, 表达式

值

范围

条件表达式

输入输出信号、变量 等

表达式

0 ~ 2147483647 (单位 : ms)

含义

等待<条件表达式>成立。

通过<表达式>设置超时时间(单位 ms)。

- 设置了超时时间时, 即使超过了超时时间且 WAIT 条件不成立时, 也结束命令。
- 最小等待执行时间为 1ms, 但是根据其它任务的执行情况, 该时间会有所变化。



注意

省略超时指定的场合, 条件表达式不成立时只要从外部停止程序(例如联锁及紧急停止等), 机器人就会无反应(也无警报输出等)呈停止状态。



MEMO

当条件表达式为表示数值的表达式时, 表达式的值非 0 时为“真”, 成为 0 时则为“假”。

SAMPLE

说明

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
WAIT END_SENSOR = 1	…输入输出名称“END_SENSOR”打开前进行等待

输入输出名称…用户可为 DI 或 DO 的特定比特设置的任意名称



MEMO

输入输出名称仅可在支持软件上添加和编辑。无法通过手持编程器进行添加和编辑。

相关命令

DRIVE, DRIVEI, MOVE, MOVEI, MOVET

WAIT ARM

等待机器人轴动作结束

联机

×

8

命令

格式

WAIT ARM [机器人编号] (轴编号)

值

范围

机器人编号

1 ~ 4 (省略时, 机器人 1 被指定。)

轴编号

1 ~ 6 (当无指定时, 指定机器人的所有轴成为对象。)

含义 等待<机器人编号>指定机器人的轴动作完成(定位于公差内)。

SAMPLE

说明

A=TSKPGM (1)

……等待机器人 1 的动作结束。

WAIT ARM [2] (2)

……等待机器人 2 中第 2 轴的动作结束。

相关命令

DRIVE, DRIVEI, MOVE, MOVEI, MOVET, PMOVE

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

WEIGHT

设置 / 获取前端质量 (kg) 参数

联机



命令

格式	
WEIGHT [机器人编号] 表达式	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
表达式	范围因所设置的机器人而异。

含义 将<机器人编号>指定机器人的前端质量 (kg) 参数变更为<表达式>的值。
对附加轴无任何影响。

函数

格式	
WEIGHT [机器人编号]	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）

含义 获取<机器人编号>指定机器人的前端质量 (kg) 参数值。

SAMPLE	说明
A=5	
B=2	
C=WEIGHT	将机器人 1 的前端质量 (kg) 参数赋值给变量 C
WEIGHT A	将机器人 1 的前端质量 (kg) 参数更改为变量 A 的值 (5)
MOVE P, P0	
WEIGHT B	将机器人 1 的前端质量 (kg) 参数更改为变量 B 的值 (2)
MOVE P, P1	
WEIGHT C	将机器人 1 的前端质量 (kg) 参数复原 (变量 C 的值)
D=WEIGHT	将机器人 1 的前端质量 (kg) 参数赋给变量 D。
HALT	

MEMO

同时设定前端质量 (kg) 参数 <WEIGHT> 和前端质量 (g) 参数 <WEIGHTG> 时，将设定合计值。

相关命令	WEIGHTG
------	---------

WEIGHTG

设置 / 获取前端质量 (g) 参数

联机



8

命令

格式

WEIGHTG [机器人编号] 表达式

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

表达式 范围因所设置的机器人而异。

含义 将<机器人编号>指定机器人的前端质量 (g) 参数变更为<表达式>的值。
对附加轴无任何影响。

函数

格式

WEIGHTG [机器人编号]

值

范围

机器人编号 1 ~ 4 (省略时, 机器人 1 被指定。)

含义 获取<机器人编号>指定机器人的前端质量 (g) 参数值。

SAMPLE

说明

A=5
B=2
C=WEIGHTG将机器人 1 的前端质量 (g) 参数赋值给变量 C
WEIGHTG A将机器人 1 的前端质量 (g) 参数更改为变量 A 的值 (5)
MOVE P, P0
WEIGHTG B将机器人 1 的前端质量 (g) 参数更改为变量 B 的值 (2)
MOVE P, P1
WEIGHTG C将机器人 1 的前端质量 (g) 参数复原 (变量 C 的值)
D=WEIGHTG将机器人 1 的前端质量 (g) 参数赋给变量 D。
HALT



MEMO

同时设定前端质量 (kg) 参数 <WEIGHT> 和前端质量 (g) 参数 <WEIGHTG> 时, 将设定合计值。

相关命令

WEIGHTG

WEND

结束 WHILE 语句的命令区

联机

×

命令

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	说明
A=0 WHILE DI3 (0) =0 A=A+1 MOVE P, P0 MOVE P, P1 PRINT "COUNTER=" ; A WEND HALT	…………DI3 (0) 为 OFF 期间，重复执行 WHILE ~ WEND 循环 …………退出 WHILE 语句的命令区

相关命令	WHILE
------	-------

WHERE

获取机械臂当前位置（脉冲坐标）

联机



8

函数

格式

WHERE [机器人编号]

值

范围

机器人编号

1 ～ 4（省略时，机器人 1 被指定。）

含义 根据关节坐标获取＜机器人编号＞指定机器人的机械臂的当前位置。

SAMPLE

说明

P10=WHERE

……将当前位置的脉冲坐标值赋值给 P10

相关命令

WHRXY

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

WHILE ~ WEND

在条件成立时反复动作

联机

×

命令

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。
- [从 WHILE ~ WEND 循环内跳转至 WHILE ~ WEND 循环之外]
可以通过 GOTO 等命令进行跳转。

MEMO

[当条件表达式为表示数值的表达式时] 获取。

SAMPLE 1	说明
A=0 WHILE DI3 (0) =0 A=A+1 MOVE P, P0 MOVE P, P1 PRINT “COUNTER=” ; A WEND HALT	…………DI3 (0) 为 OFF 期间，重复执行 WHILE ~ WEND 循环 …………退出 WHILE 语句的命令区
SAMPLE 2	
A=0 WHILE -1 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	…………由于条件表达式始终为真 (0 以外)，因此将进入永久循环 …………退出 WHILE 语句的命令区

WHRXY

通过正交坐标获取机械臂当前位置

联机



8

函数

格式

WHRXY [机器人编号]

值

范围

机器人编号

1 ~ 4 (省略时, 机器人 1 被指定。)

含义

根据正交坐标获取 < 机器人编号 > 指定机器人的机械臂的当前位置。



MEMO

[使用 YK-TW 时] 获取也第 1 机械臂转数信息、第 2 机械臂转数信息。

SAMPLE

说明

P10=WHRXY

.....将机器人 1 的当前位置直行坐标值赋给 P10

相关命令

WHERE

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

WRKDEF

工件定义（作成指定编号的工件）

联机



命令

形式		
定义语句： WRKDEF Wn = X 坐标偏移量 Y 坐标偏移量 Z 坐标偏移量 R 坐标偏移量		
选择语句： CHGWRK [机器人编号] Wn		
标注	值	范围
	机器人编号	1 ～ 4（省略时，机器人 1 被指定。）
n	工件编号	0 ～ 39
	X 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。（单位：mm）
	Y 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。（单位：mm）
	Z 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。（单位：mm）
	R 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。（单位：度）

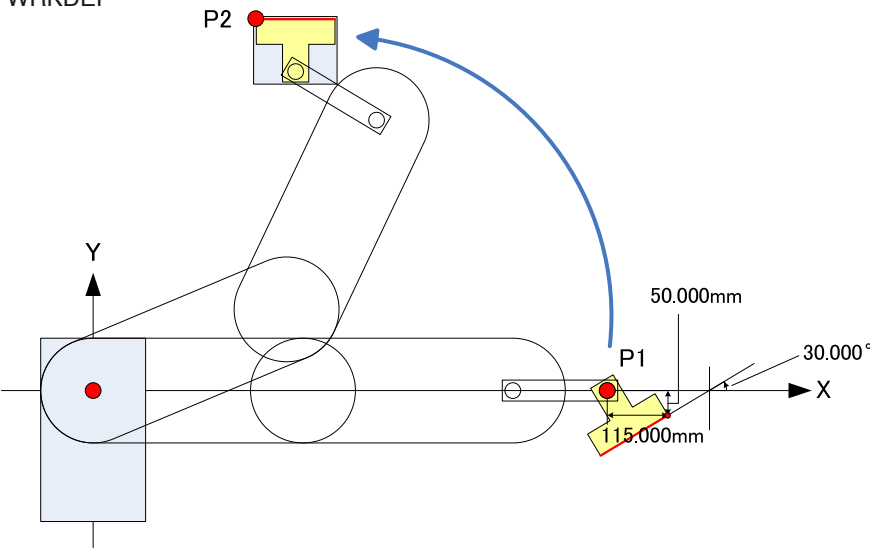
含义	WRKDEF 指令仅使用于工件定义。在替换实际的工件数据时，使用 CHGWRK
参照	“CHGWRK” RCX3《用户手册》“工件定义”
MEMO	如果在工件定义执行中关闭电源，将可能会发生“9.708: 工件校验和错误 (Work data destroyed)”

X 坐标偏移量：用实数指定工件 n 的机器人前端（机械手臂使用时为工具前端）开始的 X 坐标位移量。
Y 坐标偏移量：用实数指定将工件 n 的机器人前端（机械手臂使用时为工具前端）开始的 Y 坐标位移量。
Z 坐标偏移量：用实数指定将工件 n 的机器人前端（机械手臂使用时为工具前端）开始的 Z 坐标位移量。
R 坐标偏移量：机器人前端（机械手臂使用时为工具前端）的 R 坐标当前位置为 0.000 时，直交坐标 + X 方向和工件 n 的角度用实数指定。逆时针方向时用 + 值输入。

相关命令	CHGWRK, CREWRK
------	----------------

SAMPLE	说明
CHANGE H1	更改为机器人 1 的机械手 1
MOVE P, P1	机械手 1 的前端向 P1 移动
WRKDEF W1=	115.000 -50.000 0.000 30.000
CHGWRK W1	更改为机器人 1 的工件 1
MOVE P, P2	工件 1 的前端向 P2 移动
HALT	

SAMPLE: WRKDEF



XYTOJ

将正交坐标数据（毫米）转换为关节坐标数据（脉冲）

联机



函数

格式	
XYTOJ [机器人编号] (坐标点表达式)	
值	范围
机器人编号	1 ～ 4（省略时，机器人 1 被指定。）

含义

将<坐标点表达式>中表示的正交坐标数据(单位：mm、度)转换为<机器人编号>指定机器人的关节坐标数据(单位：脉冲)。
根据执行时设置的基准坐标、位移坐标及机械手定义转换。

[当水平多关节机器人时] 根据右手 / 左手系统的执行结果会有所不同。
[当 YK-TW 系列时] 由于 X、Y 机械臂转数信息的指定不同 结果也会出现差异，因此必须加以注意。
[将关节坐标数据转换为正交坐标数据时] 使用 JTOXY。

SAMPLE	说明
P10=XYTOJ (P10)	……………将 P10 转换为关节坐标数据

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)
- 由于速度不会因为点位示教位置的疏密而发生变化,点位示教相当方便。
- 除了可更改整个移动路径的移动速度之外,还可通过速度选项只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,必须在设置时使用以下机器人语言。

- PATH SET 启动路径设置
- 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 语句之间的部分,使用 PATH 语句设置动作路径。此时,机器人动作。如果在路径设置结束后执行 PATH START 语句,将执行已设置的动作路径移动。

SAMPLE	说明
MOVE P,P0,Z=0 PATH SET	……………开始设置机器人 1 的路径
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	……………结束机器人 1 的路径设置
MOVE P,P1,Z=0 . . .	
MOVE P,P0,Z=0 PATH START HALT	……………机器人 1 进行 PATH 移动

- 直线与圆弧路径的最大合计坐标点数为 1000 个。

PATH L 的 1 个坐标点形成 1 条直线、PATH C 的 2 个坐标点形成 1 个圆弧，因此路径的坐标点可按 (PATH L 中指定的坐标点数) + (PATH C 中指定的坐标点数 ÷ 2) 计算，合计最多可设为 1000 个。

$$\text{PATH L 中指定的坐标点数} + \frac{\text{PATH C 中指定的坐标点数}}{2} \leq 1000$$

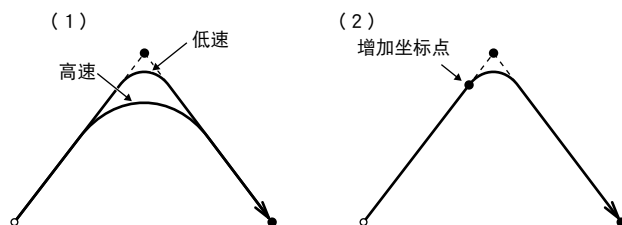
- 执行 PATH 移动 (执行 PATH START 语句) 时，必须将机器人定位到已设置路径的起点。
- 在圆弧与直线连接点的位置，请尽量使其移动方向一致。方向发生较大变化时，可能会发生振动或机器人报错。

圆弧与直线的连接点：方向发生较大变化时



- (1) 在直线与直线的连接点位置，请使移动时的路径从连接点的内侧通过。
速度越快，通过时越靠近内侧。
- (2) 不想由于速度而使路径发生较大变化时，请增加坐标点。
请注意，如果不降低速度，可能会出错。

直线与直线的连接点：抑制路径变化



- 当不按照指定速度移动而出错时
在 PATH 移动开始时、停止时及移动路径中途改变速度指定时，机器人进行加速或减速。此时，如果由于坐标点之间的距离较短无法达到指定速度或无法减速，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，使得包含加减速区间的直线或圆弧延长。
- 请使得 PATH 移动中的手系统与 PATH 移动路径的起点的手系统相同。朝着设置了手系统标志的坐标点进行 PATH 移动时，也是一样。手系统不同时，将出错并无法动作。
- 请使 PATH 移动中的第 1 机械臂转数信息及第 2 机械臂转数信息与 PATH 移动路径起点的第 1 机械臂转数信息及第 2 机械臂转数信息一致。如果不一致，将出错并无法动作。
- PATH 移动中如果机器人由于停止信号等原因停止，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。
- 在 PATH 语句指定的坐标中，能够移动的只有 X, Y, Z 的坐标值。
除此之外的坐标值，使用 PATH 移动路径的起点坐标值，例如：使用 R 轴时，则不能将 R 轴移动至指定座标。

第 10 章

数据文件详细说明

1	概要	10-1
2	程序文件.....	10-3
3	坐标点文件.....	10-5
4	坐标点注释文件	10-8
5	坐标点名称文件	10-10
6	参数文件.....	10-12
7	位移坐标定义文件	10-16
8	机械手定义文件.....	10-18
9	工件定义文件.....	10-20
10	托盘定义文件.....	10-22
11	通用以太网端口定义文件.....	10-26
12	输入输出名称文件	10-28
13	区域判定输出定义文件.....	10-32
14	实时输出文件.....	10-34
15	ALL 文件.....	10-38
16	程序目录文件.....	10-40
17	参数目录文件.....	10-42
18	机器参照文件.....	10-43

19	累计数据文件	10-45
20	系统构成信息文件	10-46
21	版本信息文件	10-47
22	选件板文件	10-48
23	自我诊断文件	10-49
24	报警履历文件	10-50
25	存储器剩余容量文件	10-52
26	变量文件	10-53
27	常量文件	10-59
28	数组变量文件	10-60
29	DI 文件	10-62
30	DO 文件	10-64
31	MO 文件	10-66
32	LO 文件	10-68
33	TO 文件	10-70
34	SI 文件	10-72
35	SO 文件	10-74
36	SIW 文件	10-76
37	SOW 文件	10-78
38	EOF 文件	10-80
39	串行端口通信文件	10-81
40	Ethernet 端口通信文件	10-82

1.1 数据文件种类

在本章节中，将向您介绍在 SEND 语句或联机命令的 READ / WRITE 中可使用的数据文件。

分类	文件名	记述格式		读出	写入
		所有	个别文件		
用户	所有文件	ALL	——	✓	✓
	程序	PGM	<bbbbbb> PGn	✓	✓
	ポイント	PNT	Pn	✓	✓
	坐标点注释	PCM	PCn	✓	✓
	坐标点名称	PNM	PNn	✓	✓
	参数	PRM	/ccccccc/ #ccccccc# \ccccccc\	✓	✓
	位移坐标定义	SFT	Sn	✓	✓
	机械手定义	HND	Hn	✓	✓
	工件定义	WRKDEF	Wn	✓	✓
	托盘定义	PLT	PLn	✓	✓
	通用以太网端口	GEP	GPn	✓	✓
	输入输出名称	ION	iNMn(n)	✓	✓
	区域判定输出	ACO	ACn	✓	✓
	实时输出	RTO	RTn	✓	✓
变量・常量	变量	VAR	ab...by	✓	✓
	数组变量	ARY	ab...by(x)	✓	✓
	定数	-----	“cc...c”	✓	×
状态	程序目录	DIR	<<bbbbbb>>	✓	×
	参数目录	DPM	——	✓	×
	机器参照 (传感器、撞块) (标记)	MRF	——	✓	×
		ARP	——	✓	×
	系统构成信息	CFG	——	✓	×
	累计数据(*)	ACD	——	✓(*)	×
	版本信息	VER	——	✓	×
	选件板	OPT	——	✓	×
	自我诊断	SCK	——	✓	×
	报警履历	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()	SO n()	✓	✓
	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 : 英字 b : 字母数字或下划线(_) c : 字母数字或特殊标注

x : 表达式(数组参数) y : 变量类型 i : 输入输出类型

✓ : 可以

× : 不可以

1.2 注意点

处理数据文件时，请注意以下要点。

- 可使用的字符全部为半角。
- 数据全部基于 ASCII (美国信息交换标准码) 进行字符编码的字符串处理。
- 命令语句全部仅限大写英语字母 (不可使用小写英语字母)。
- 所有行必须在 255 个字符以内。
- 数据格式的标注中，[cr/lf] 表示 CR 代码 (0Dh)+LF 代码 (0Ah)。
- 本书中读出及写入的语言使用以下数据流程的含义。

读出：控制器 → 外部通信设备

写入：外部通信设备 → 控制器

*) 累计数据：在软件版本 Ver.1.73 以上有效

累计数据的读出：在软件版本 Ver.2.04 以上有效

2.1 整个程序

读出	✓	读出所有已录入的程
写入	✓	在控制器中录入 NAME=< 程序名 > 行中指定的程序名。

格式
PGM

- 含义
- 表示整个程序
 - 当为写入文件时，如果对程序编号进行了指定，则以新的程序替换。
 - 当在写入文件中省略了程序编号时，可分配空置的最小编号。当存在同名但程序编号不同的程序时，较早的程序将被删除。

DATA FORMAT
NAME= 程序名 [cr/lf] PGN=mmm [cr/lf] aaaaa ...aaaaaaaaaaaaaaaaa[cr/lf] : aaaaa ...aaaaaaaaaaaaaaaaa[cr/lf] : NAME= 程序名 [cr/lf] PGN=mmm [cr/lf] aaaaa ...aaaaaaaaaaaaaaaaa[cr/lf] : aaaaa ...aaaaaaaaaaaaaaaaa[cr/lf] [cr/lf]

标注	值	范围
a	字符编码	
mmm	程序编号	1 ~ 100
	程序名	采用 32 字符以内的英文字母数字或下划线 (_) 表示

- 在文件的末尾仅附加表示结束的 [cr/lf] 行。

SAMPLE	说明
SEND PGM TO CMU SEND CMU TO PGM	 从通信端口输出整个程序 从通信端口输入整个程序
应答： NAME=TEST [cr/lf] PGN=1 [cr/lf] PGN=1 A=1 [cr/lf] RESET DO2 () [cr/lf] : HALT [cr/lf] [cr/lf]	

2.2 程序单位

读出

✓

写入

✓

格式

1. <程序名>

2. PGmmm

含义

- 表示指定程序。
- mmm 以 1 ~ 100 表示。
- <程序名> 采用 32 字符以内的英文字母数字或“_”表示，两侧添加 < > (尖括号)。
- 当在 1 的形式中未指定 <程序名> 称时，将选择当前指定的程序。
- 格式上指定的程序名与数据上的 <程序名> 不同时，将出错。

DATA FORMAT

```
NAME= 程序名 [cr/lf]
PGN=mmm [cr/lf]
aaaaa ...aaaaaaaaaaaaaaaa[cr/lf]
:
aaaaa ...aaaaaaaaaaaaaaaa[cr/lf]
[cr/lf]
```

标注

值

范围

a

字符编码

mmm

程序编号

1 ~ 100

程序名

采用 32 字符以内的英文字母数字或下划线(_)表示



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。
- 指定程序编号中保存有其他程序时，将被覆盖。
- 如果没有程序编号，则自动指定未使用程序编号中的最小程序编号。
- 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"。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

说明

```
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：

```
RUN [cr/lf]
NAME=TEST1[cr/lf]
PGN=1[cr/lf]
A=1[cr/lf]
RESET DO2()[cr/lf]
:
HALT[cr/lf]
[cr/lf]
END [cr/lf]
```

3.1 所有坐标点

读出	✓	读出所有已录入的坐标点。
写入	✓	通过坐标点编号写入。

格式

PNT

含义 表示所有坐标点。

DATA FORMAT

```
Pmmmm= fxxxxxx fyxxxxx fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
Pmmmm= fxxxxxx fyxxxxx fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
:
Pmmmm= fxxxxxx fyxxxxx fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
Pmmmm= fxxxxxx fyxxxxx fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]
[cr/lf]
```



补充

- 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。
- 数值中带有“.”时，将被识别为毫米单位的坐标系。
- 数据采用 1 个以上的空格隔开。

标注	值	范围、含义
mmmm	坐标点编号	0 ~ 29999
f	坐标点标注	+ / - / 空格
xxxxxx/./ bbbbbb	8 位以内的数值。	
t	水平多关节型机器人上扩展设置的手系统标志	1 : 右手系统 2 : 左手系统
xr	为 YK-TW 系列第 1 机械臂转数信息	0 : mm→脉冲转换后的角度数据 x (*1) 的范围为 $-180.00^{\circ} < x \leq 180.00^{\circ}$ 1 : mm→脉冲转换后的角度数据 x (*1) 的范围为 $180.00^{\circ} < x \leq 540.00^{\circ}$ -1 : mm→脉冲转换后的角度数据 x (*1) 的范围为 $-540.00^{\circ} < x \leq -180.00^{\circ}$
yr	为 YK-TW 系列第 2 机械臂转数信息	0 : mm→脉冲转换后的角度数据 y (*1) 的范围为 $-180.00^{\circ} < y \leq 180.00^{\circ}$ 1 : mm→脉冲转换后的角度数据 y (*1) 的范围为 $180.00^{\circ} < y \leq 540.00^{\circ}$ -1 : mm→脉冲转换后的角度数据 y (*1) 的范围为 $-540.00^{\circ} < y \leq -180.00^{\circ}$

*1 : 这是将转换为关节坐标后的脉冲数据，再转换为与各机械臂机械原点间的角度后的数据。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK-TW 系列且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。

坐标点文件

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明									
SEND PNT TO CMU从通信端口输出所有坐标点										
应答：										
RUN [cr/lf]										
P0	=	1	2	3	4	5	6	[cr/lf]		
P1	=	426.200	-160.770	0.001	337.210	0.000	0.000	0	1	0 [cr/lf]
P2	=	-27.570	-377.840	0.360	193.220	0.000	0.000	0	-1	0 [cr/lf]
:										
P29999	=	-251.660	-419.510	0.000	-127.790	0.000	0.000	2	-1	-1 [cr/lf]
[cr/lf]										
END [cr/lf]										

3.2 坐标点单位

读出	✓
写入	✓

格式

Pmmmm

含义 表示指定坐标点。

DATA FORMAT

Pmmmm= fxxxxxx fyyyyyy fzzzzzz frrrrrr faaaaaa fbbbbb t xr yr [cr/lf]



补充

- 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。
- 数值中带有.时，将被识别为毫米单位的坐标系。
- 数据采用 1 个以上的空格隔开。

标注	值	范围、含义
mmmm	坐标点编号	0 ~ 29999
f	坐标点标注	+ / - / 空格
xxxxxx/./ bbbbbb	8 位数以内的数值	
t	水平多关节型机器人上扩展设置的手系统标志。	1：右手系统 2：左手系统
xr	为 YK-TW 系列第 1 机械臂转数信息。	0：mm→脉冲转换后的脉冲数据 x (*1) 的范围为 -180.00° < x <= 180.00° 1：mm→脉冲转换后的脉冲数据 x (*1) 的范围为 180.00° < x <= 540.00° -1：mm→脉冲转换后的脉冲数据 x (*1) 的范围为 -540.00° < x <= -180.00°
yr	为 YK-TW 系列第 2 机械臂转数信息。	0：mm→脉冲转换后的脉冲数据 x (*1) 的范围为 -180.00° < y <= 180.00° 1：mm→脉冲转换后的脉冲数据 x (*1) 的范围为 180.00° < y <= 540.00° -1：mm→脉冲转换后的角度数据 x (*1) 的范围为 -540.00° < y <= -180.00°

*1：这是将转换为关节坐标后的脉冲数据，再转换为与各机械臂机械原点间的角度后的数据。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK-TW 系列且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。

SAMPLE

说明

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：

RUN [cr/lf]
P100= 1.000 2.000 3.000 4.000 5.000 6.000 0 1 0 [cr/lf]
END [cr/lf]

4.1 所有坐标点注释

读出	✓	读出所有已录入的坐标点注释。
写入	✓	通过坐标点注释编号写入。

格式
PCM

含义 表示所有坐标点注释。

DATA FORMAT
PCmmmm= sssssssssssssssssss[cr/lf]
PCmmmm= sssssssssssssssssss[cr/lf]
:
PCmmmm= sssssssssssssssssss[cr/lf]
PCmmmm= sssssssssssssssssss[cr/lf]
[cr/lf]

标注	值	范围
mmmm	坐标点注释编号	0 ~ 29999
ss...ss	注释数据	· 可使用半角和全角字符（半角：最多 16 个字符 全角：最多 8 个字符） · 删除写入数据时超过最大字符数的字符。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
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：	
RUN [cr/lf]	
PC1 = ORIGIN_POS[cr/lf]	
PC3 = WAIT_POS[cr/lf]	
:	
PC3999 = WORK100[cr/lf]	
[cr/lf]	
END [cr/lf]	

4.2 坐标点注释单位

读出	✓
写入	✓

格式
PCmmmm

含义 表示指定坐标点注释。

DATA FORMAT
PCmmmm= sssssssssssssssss[cr/lf]

标注	值	范围
mmmm	坐标点编号	0 ~ 29999
ss...ss	注释数据	· 可使用半角和全角字符（半角：最多 16 个字符 全角：最多 8 个字符） · 删除写入数据时超过最大字符数的字符。

SAMPLE	说明
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：	
RUN [cr/lf]	
PC1 = ORIGIN_POS[cr/lf]	
END [cr/lf]	

5 坐标点名称文件

5.1 所有坐标点名称

读出	✓	读出所有已录入的坐标点名称。
写入	✓	通过坐标点名称编号写入。

格式

PNM

含义 表示所有坐标点名称。

DATA FORMAT

```
PNmmmm= a s s s s s s s s s [ c r / l f ]
PNmmmm= a s s s s s s s s s [ c r / l f ]
:
PNmmmm= a s s s s s s s s s [ c r / l f ]
PNmmmm= a s s s s s s s s s [ c r / l f ]
[ c r / l f ]
```

标注	值	范围
mmmm	坐标点名称编号	0 ~ 29999
a	名称数据 (第 1 个字符)	仅可使用半角字母。 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。
SS...SS	名称数据 (第 2 个字符之后)	· 仅可使用半角字母数字或下划线 (_)。 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。 · 名称数据最多为 16 个字符。 写入数据时超过 16 个字符时, 第 17 个字符之后将被删除。

MEMO

不可重复设置已有的名称数据。

如果名称数据重复, 将删除已有的坐标点名称编号, 并在新指定的坐标点名称编号中保存名称数据。

- 在文件的末尾只加上表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND PNM TO CMU SEND CMU TO PNM	从通信端口输出所有坐标点名称 从通信端口输入所有坐标点名称
应答 : RUN [c r / l f] PN1=ORIGIN_POS [c r / l f] PN3=WAIT_POS [c r / l f] : PN3999=WORK100 [c r / l f] [c r / l f] END [c r / l f]	

9

10

标注	值	范围
mmmm	坐标点名称编号	0 ~ 29999
a	名称数据 (第 1 个字符)	仅可使用半角字母。 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。
SS...SS	名称数据 (第 2 个字符之后)	· 仅可使用半角字母数字或下划线 (_)。 · 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。 · 名称数据最多为 16 个字符。 · 写入数据时超过 16 个字符时, 第 17 个字符之后将被删除。

坐标点注释文件 ● 10-11

6.1 所有参数

读出	✓	读出所有参数。
写入	✓	仅写入标签指定的参数。

格式
PRM

含义 表示所有参数。

DATA FORMAT
/ 参数标签 / [cr/lf] RC=xxxxxxx [cr/lf] / 参数标签 / [cr/lf] R?=xxxxxxx[cr/lf] / 参数标签 / [cr/lf] R?A=xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx [cr/lf] \ 参数标签 \ [cr/lf] C?=xxxxxxx [cr/lf] \ 参数标签 \ [cr/lf] R?=xxxxxxx[cr/lf] \ 参数标签 \ [cr/lf] R?A=xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx [cr/lf] # 参数标签 # [cr/lf] R?=xxxxxxx[cr/lf] # 参数标签 # [cr/lf] R?A=xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx [cr/lf] / 参数标签 / [cr/lf] C?O=xxxxxxx,xxxxxxx,xxxxxxx,xxxxxxx [cr/lf] : [cr/lf]

标注	值	备注
RC	所有控制器	
R? ?	机器人指定 机器人编号	
C? ?	控制器指定 控制器编号	
A	轴参数	各轴的数据以逗号进行间隔。
O	选件板参数	各选件板的数据以逗号隔开。

- 参数标签采用 8 个英文字母表示。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

MEMO

- 写入参数时，请设置为伺服禁止状态。
- 参数虽然具备与高版本之间的兼容性，但是有时不具备与低版本之间的兼容性(高版本兼容)。
- 如果将新版本的参数文件载入旧版本的控制器时，则会发生[10.214:未定义的参数]错误。这种情况下，如果在参数设定中将“不要载入未定义的参数”设为有效，则可以正常地载入除未定义参数之外的其它参数(详细说明，请参阅控制器的手册)。
- 参数标签两侧有\的参数为控制器构成参数。编辑时请格外注意。
- 参数标签两侧有#的参数将影响机器人的控制。编辑时请格外注意。
- 部分计算机中，反斜杠(\)可能显示为 ¥ 标记。

SAMPLE	说明
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：	
RUN [cr/lf]	
' V1.22,R0191-V1.000-V1.09,R0015/V1.09,R0015 [cr/lf]	
' Gripper,V0.32/Gripper,V0.32///[cr/lf]	
' PRM(0)[cr/lf]	
\CNTTYP\[cr/lf]	
C1=340[cr/lf]	
\YCEADR\[cr/lf]	
C1=0[cr/lf]	
\DRVASGN\[cr/lf]	
R1A=101,102,103,104,0,0[cr/lf]	
R2A=0,0,0,0,0,0[cr/lf]	
R3A=0,0,0,0,0,0[cr/lf]	
R4A=0,0,0,0,0,0[cr/lf]	
\RBTNUM\[cr/lf]	
R1=2203[cr/lf]	
:	
[cr/lf]	
END [cr/lf]	

6.2 参数单位

读出	✓	仅读出标签指定参数
写入	✓	仅写入标签指定参数

格式
/ 参数标签 /、\ 参数标签 \、# 参数标签 #

含义 表示指定参数。
参数标签采用 8 个英文字母表示。

DATA FORMAT 1
/ 参数标签 / [cr/lf] RC=xxxxxx [cr/lf] [cr/lf]

DATA FORMAT 2
/ 参数标签 / [cr/lf] R?=xxxxxx [cr/lf] [cr/lf]

DATA FORMAT 3
/ 参数标签 / [cr/lf] R?A=xxxxxx,xxxxxx,xxxxxx,xxxxxx,xxxxxx,xxxxxx [cr/lf] [cr/lf]

DATA FORMAT 4
\ 参数标签 \ [cr/lf] C?=xxxxxx [cr/lf] [cr/lf]

DATA FORMAT 5
\ 参数标签 \ [cr/lf] R?=xxxxxx [cr/lf] [cr/lf]

DATA FORMAT 6
\ 参数标签 \ [cr/lf] R?A=xxxxxx,xxxxxx,xxxxxx,xxxxxx,xxxxxx,xxxxxx [cr/lf] [cr/lf]

DATA FORMAT 7
参数标签 # [cr/lf] R?=xxxxxx [cr/lf] [cr/lf]

DATA FORMAT 8

```
# 参数标签 # [cr/lf]
R?A=xxxxxx,xxxxxx,xxxxxx,xxxxxx,xxxxxx,xxxxxx [cr/lf]
[cr/lf]
```

DATA FORMAT 9

```
/ 参数标签 / [cr/lf]
C?O=xxxxxx,xxxxxx,xxxxxx,xxxxxx [cr/lf]
[cr/lf]
```

标注	值	备注
RC	表示所有控制器。	
R? ?	机器人指定 机器人编号	
C? ?	控制器指定 控制器编号	
A	表示轴参数	各轴的数据以逗号进行间隔。
O	选件板参数	各选件板的数据以逗号隔开。

- 参数标签采用 8 个英文字母表示。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

MEMO

- 写入参数时，请设置为伺服禁止状态。
- 参数虽然具备与高版本之间的兼容性，但是有时不具备与低版本之间的兼容性（高版本兼容）。
- 如果将新版本的参数文件载入旧版本的控制器时，则会发生 [10.214: 未定义的参数] 错误。这种情况下，如果在参数设定中将“不要载入未定义的参数”设为有效，则可以正常地载入除未定义参数之外的其它参数（详细说明，请参阅控制器的手册）。
- 参数标签两侧有 \ 的参数为控制器构成参数。编辑时请格外注意。
- 参数标签两侧有 # 的参数将影响机器人的控制。编辑时请格外注意。
- 部分计算机中，反斜杠 (\) 可能显示为 ¥ 标记。

SAMPLE	说明
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：	
RUN [cr/lf]	
/ACCEL/ [cr/lf]	
R1A=100, 100, 100, 100 [cr/lf]	
[cr/lf]	
END [cr/lf]	

7 位移坐标定义文件

7.1 所有位移

读出	✓	读出所有位移。
写入	✓	通过位移编号写入。

格式
SFT

含义 表示所有位移。

DATA FORMAT				
Sm	=	fxxxxxxx	fyyyyyy	fzzzzzzz frrrrrrr [cr/lf]
SPm	=	fxxxxxxx	fyyyyyy	fzzzzzzz frrrrrrr [cr/lf]
SMm	=	fxxxxxxx	fyyyyyy	fzzzzzzz frrrrrrr [cr/lf]
:				
Sm	=	fxxxxxxx	fyyyyyy	fzzzzzzz frrrrrrr [cr/lf]
SPm	=	fxxxxxxx	fyyyyyy	fzzzzzzz frrrrrrr [cr/lf]
SMm	=	fxxxxxxx	fyyyyyy	fzzzzzzz frrrrrrr [cr/lf]
[cr/lf]				

标注	值	范围
m	位移编号	0 ~ 39
f	坐标点标注	+ / - / 空格
xxxxxx/yyyyy/./rrrrr	7 位以内小数点后 3 位以内的实数值	

- 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	说明
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：	
RUN [cr/lf]	
S0 = 0.000 0.000 0.000 0.000	[cr/lf]
SP0= 0.000 0.000 0.000 0.000	[cr/lf]
SM0= 0.000 0.000 0.000 0.000	[cr/lf]
S1 = 1.000 1.000 1.000 1.000	[cr/lf]
:	
SM39= 9.000 9.000 9.000 9.000	[cr/lf]
[cr/lf]	
END [cr/lf]	

7.2 位移单位

读出	✓
写入	✓

格式
Sm

含义 表示指定位移。

DATA FORMAT
Sm = fxxxxxxx fyyyyyyy fzzzzzzz frrrrrrr[cr/lf]

标注	值	范围
m	位移编号	0 ~ 39
f	坐标点标注	+ / - / 空格
xxxxxx/yyyyyy/./rrrrrr	7 位以内小数点后 3 位以内的实数值	

■ 在文件的末尾只加上表示文件结束的 [cr/lf] 行。

SAMPLE	说明
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：	
RUN [cr/lf]	
S0 = 0.000 0.000 0.000 0.000[cr/lf]	
SP0= 0.000 0.000 0.000 0.000[cr/lf]	
SM0= 0.000 0.000 0.000 0.000[cr/lf]	
[cr/lf]	
END [cr/lf]	

8 机械手定义文件

8.1 所有机械手

读出	✓	读出所有机械手。
写入	✓	通过机械手编号写入。

格式
HND

含义 表示所有机械手。

DATA FORMAT
Hm = n, fxxxxxxx, fyyyyyyy, fzzzzzzz , {R} [cr/lf]
:
Hm = n, fxxxxxxx, fyyyyyyy, fzzzzzzz , {R} [cr/lf]
[cr/lf]

标注	值	范围
m	机械手编号	0 ~ 31
n	机器人编号	1 ~ 4
f	坐标点标注	+ / - / 空格
{R}	是否在 R 轴上安装了机械手	
xxxxxx/yyyyyy/zzzzzz	7 位以内小数点后 3 位以内的实数值或 6 位以下的整数值 (数值形式取决于机器人类型的设置或机械手定义的种类)	

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
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 :	
RUN [cr/lf]	
H0 = 1, 0.000, 0.000, 0.000 [cr/lf]	
H1 = 1, 1.000, 1.000, 1.000 [cr/lf]	
H2 = 2, 2.000, 2.000, 2.000 [cr/lf]	
H3 = 2, 3.000, 3.000, 3.000 [cr/lf]	
H4 = 3, 4.000, 4.000, 4.000 [cr/lf]	
H5 = 3, 5.000, 5.000, 5.000 [cr/lf]	
H6 = 4, 6.000, 6.000, 6.000 [cr/lf]	
H7 = 4, 7.000, 7.000, 7.000 [cr/lf]	
[cr/lf]	
END [cr/lf]	

8.2 机械手单位

读出	✓
写入	✓

格式
Hm

含义 表示指定机械手定义。

DATA FORMAT
Hm = n, fxxxxxxx, fyyyyyyy, fzzzzzzz, {R} [cr/lf]

标注	值	范围
m	机械手编号	0 ~ 31
n	机器人编号	1 ~ 4
f	坐标点标注	+ / - / 空格
{R}	是否在 R 轴上安装了机械手	
xxxxxx/yyyyyy/zzzzz	7 位以内小数点后 3 位以内的实数值或 6 位以下的整数值 (数值形式取决于机器人类型的设置或机械手定义的种类)	

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
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：	
RUN [cr/lf]	
H3=2, 3.000, 3.000, 3.000, R [cr/lf]	
END [cr/lf]	

9.1 所有工件

读出	✓	读出所有工件定义的数据。
写入	✓	通过工件编号写入。

格式
WRKDEF

含义 表示所有工件。

DATA FORMAT
Wm = fxxxx.xxx, fyyyy.yyy, fzzzz.zzz , frrrr.rrr [cr/lf] : Wm = fxxxx.xxx, fyyyy.yyy, fzzzz.zzz , frrrr.rrr [cr/lf][cr/lf]

标注	值	范围
m	工件编号	0 ~ 39
f	坐标点标注	+ / - / 空格
xxxx.xxx	X 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。(单位：mm)
yyyy.yyy	Y 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。(单位：mm)
zzzz.zzz	Z 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。(单位：mm)
rrrr.rrr	R 坐标偏移量	输入整数部 4 位以下，小数部 3 位以下。(单位：度)

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND WRKDEF TO CMU	……从通信端口输出所有工件
SEND CMU TO WRKDEF	……从通信端口输入所有工件
应答：	
RUN [cr/lf]	
W0 = 0.000, 0.000, 0.000, 0.000 [cr/lf]	
W1 = 1.000, 1.000, 1.000, 1.000 [cr/lf]	
W2 = 2.000, 2.000, 2.000, 2.000 [cr/lf]	
W3 = 3.000, 3.000, 3.000, 3.000 [cr/lf]	
W4 = 4.000, 4.000, 4.000, 4.000 [cr/lf]	
W5 = 5.000, 5.000, 5.000, 5.000 [cr/lf]	
[cr/lf]	
END [cr/lf]	

9.2 工件单位

读出	✓
写入	✓

格式
Wm

含义 表示指定工件数据。

DATA FORMAT
Wm = fxxxx.xxx fyyyy.yyy fzzzz.zzz frrrr.rrr [cr/lf]

标注	值	范围
m	工件编号	0 ~ 39
f	坐标点标注	+ / - / 空格
xxxx.xxx	<i>X 坐标偏移量</i>	输入整数部 4 位以下，小数部 3 位以下。(单位：mm)
yyyy.yyy	<i>Y 坐标偏移量</i>	输入整数部 4 位以下，小数部 3 位以下。(单位：mm)
zzzz.zzz	<i>Z 坐标偏移量</i>	输入整数部 4 位以下，小数部 3 位以下。(单位：mm)
rrrr.rrr	<i>R 坐标偏移量</i>	输入整数部 4 位以下，小数部 3 位以下。(单位：度)

SAMPLE	说明
SEND W3 TO CMU	………从通信端口输出指定工件
SEND CMU TO W3	………从通信端口输入指定工件
应答：	
RUN [cr/lf]	
W3 = 3.000 3.000 3.000 3.000 [cr/lf]	
END [cr/lf]	

10 托盘定义文件

10.1 所有托盘定义

读出	✓	读出所有已设置的托盘定义。
写入	✓	通过托盘编号写入。

格式

PLT

含义 表示所有托盘定义。

DATA FORMAT

```
PLm [cr/lf]
PLN = XY [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [cr/lf]
PLP = ppppp [cr/lf]
P[1] = fxxxxxx fyxxxxx fzxxxxx frxxxxx faaaaaa fbbbbb t xr yr[cr/lf]
:
P[5] = fxxxxxx fyxxxxx fzxxxxx frxxxxx faaaaaa fbbbbb t xr yr[cr/lf]
PLm [cr/lf]
:
[cr/lf]
```

标注	值	范围、含义
m	托盘编号	0 ~ 39
XY	坐标平面指定	XY 坐标平面
nnn	各轴的坐标点数	正整数
ppppp	托盘定义中使用的坐标点编号	使用以指定坐标点开头的连续 5 个坐标点。
f	坐标点标注	+ / - / 空格
xxxxxx/ yyyyyy/.. bbbbbb	位数以内的数值 (足参照)	
t	水平多关节型机器人上扩展设置的手系统标志	1 : 右手系统 2 : 左手系统
xr	为 YK-TW 系列 第 1 机械臂转数信息	0 : mm→脉冲转换后的角度数据 x (*1) 的范围为 -180.00° < x ≤ 180.00° 1 : mm→脉冲转换后的角度数据 x (*1) 的范围为 180.00° < x ≤ 540.00° -1 : mm→脉冲转换后的角度数据 x (*1) 的范围为 -540.00° < x ≤ -180.00°
yr	为 YK-TW 系列 第 2 机械臂转数信息	0 : mm→脉冲转换后的角度数据 y (*1) 的范围为 -180.00° < y ≤ 180.00° 1 : mm→脉冲转换后的角度数据 y (*1) 的范围为 180.00° < y ≤ 540.00° -1 : mm→脉冲转换后的角度数据 y (*1) 的范围为 -540.00° < y ≤ -180.00°

*1 : 这是将转换为关节坐标后的脉冲数据，再转换为与各机械臂机械原点间的角度后的数据。



补充

- 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。
- 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。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 在使用了托盘定义的动作中，可忽视手系统标志、第 1 机械臂转数信息、第 2 机械臂转数信息。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK-TW 系列，且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
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：	
RUN [cr/lf]	
PL0[cr/lf]	
PLN=XY[cr/lf]	
NX = 3[cr/lf]	
NY = 4[cr/lf]	
NZ = 2[cr/lf]	
PLP= 3996[cr/lf]	
P[1]=0.000 0.000	0.000 0.000 0.000 0.000[cr/lf]
P[2]=100.000 0.000	0.000 0.000 0.000 0.000[cr/lf]
P[3]=0.000 100.000	0.000 0.000 0.000 0.000[cr/lf]
P[4]=100.000100.000	0.000 0.000 0.000 0.000[cr/lf]
P[5]=0.000 0.000	50.000 0.000 0.000 0.000[cr/lf]
PL1[cr/lf]	
PLN=XY[cr/lf]	
NX = 3[cr/lf]	
NY = 4[cr/lf]	
NZ = 2[cr/lf]	
PLP= 3991[cr/lf]	
P[1]=0.000 0.000	0.000 0.000 0.000 0.000[cr/lf]
P[2]=100.000100.000	0.000 0.000 0.000 0.000[cr/lf]
P[3]=0.000 200.000	0.000 0.000 0.000 0.000[cr/lf]
P[4]=100.000200.000	0.000 0.000 0.000 0.000[cr/lf]
P[5]=0.000 0.000	100.000 0.000 0.000 0.000[cr/lf]
[cr/lf]	
END [cr/lf]	

10.2 托盘定义单位

读出

✓

写入

✓

格式

PLm

含义 表示指定托盘定义。

DATA FORMAT

```

PLm [cr/lf]
PLN = XY [cr/lf]
PLP = ppppp [cr/lf]
NX = nnn [cr/lf]
NY = nnn [cr/lf]
NZ = nnn [cr/lf]
P[1] = fxxxxxx fyxxxxx fzzzzzz frrrrrrr faaaaaa fbbbbb t xr yr[cr/lf]
:
P[5] = fxxxxxx fyxxxxx fzzzzzz frrrrrrr faaaaaa fbbbbb t xr yr[cr/lf]
[cr/lf]

```

标注	值	范围、含义
m	托盘编号	0 ~ 39
XY	坐标平面指定	XY 坐标平面
nnn	各轴的坐标点数	正整数
ppppp	托盘定义中使用的坐标点编号	使用以指定坐标点开头的连续 5 个坐标点。
f	坐标点标注	+ / - / 空格
xxxxxx/ yyyyyy/.. bbbbbb	位数以内的数值 (补充参照)	
t	水平多关节型机器人 上扩展设置的手系统标志	1: 右手系统 2: 左手系统
xr	为 YK-TW 系列 第 1 机械臂转数信息	0: mm→脉冲转换后的角度数据 × (*1) 的范围为 $-180.00^{\circ} < x \leq 180.00^{\circ}$ 1: mm→脉冲转换后的角度数据 × (*1) 的范围为 $180.00^{\circ} < x \leq 540.00^{\circ}$ -1: mm→脉冲转换后的角度数据 × (*1) 的范围为 $-540.00^{\circ} < x \leq -180.00^{\circ}$
yr	为 YK-TW 系列 第 2 机械臂转数信息	0: mm→脉冲转换后的角度数据 × (*1) 的范围为 $-180.00^{\circ} < y \leq 180.00^{\circ}$ 1: mm→脉冲转换后的角度数据 × (*1) 的范围为 $180.00^{\circ} < y \leq 540.00^{\circ}$ -1: mm→脉冲转换后的角度数据 × (*1) 的范围为 $-540.00^{\circ} < y \leq -180.00^{\circ}$

*1: 这是将转换为关节坐标后的脉冲数据, 再转换为与各机械臂机械原点间的角度后的数据。



补充

- 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。
- 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。

- 手系统标志仅在水平多关节型机器人且指定毫米单位的坐标系时有效。
- 在使用了托盘定义的动作中，可忽视手系统标志、第 1 机械臂转数信息、第 2 机械臂转数信息。
- 将手系统标志指定为 1 及 2 以外的数值，或未指定数值时，将视作无手系统标志设定 (0)。
- 第 1 机械臂转数信息、第 2 机械臂转数信息，仅在机器人为 YK-TW 系列，且指定了毫米单位坐标系的情况下有效。
- 当在第 1 机械臂转数信息以及第 2 机械臂转数信息中指定了除 0、1、-1 之外的其它数值，或者没有指定数值时，将作为 (0) 进行处理。
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
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：	
RUN [cr/lf]	
PL2 [cr/lf]	
PLN=XY [cr/lf]	
NX= 3 [cr/lf]	
NY= 3 [cr/lf]	
NZ= 2 [cr/lf]	
PLP= 3986 [cr/lf]	
P [1]=100.000100.000 50.000 90.000 0.000 0.000 [cr/lf]	
P [2]=200.000100.000 50.000 90.000 0.000 0.000 [cr/lf]	
P [3]=100.000200.000 50.000 90.000 0.000 0.000 [cr/lf]	
P [4]=200.000200.000 50.000 90.000 0.000 0.000 [cr/lf]	
P [5]=100.00010.000 100.000 90.000 0.000 0.000 [cr/lf]	
[cr/lf]	
END [cr/lf]	

读出	✓	读出所有通用以太网端口定义。
写入	✓	通过通用以太网端口编号写入。

格式

GEP

含义 表示所有通用以太网端口定义。

DATA FORMAT

```
G P m   [ c r / l f ]  
M O D E = n   [ c r / l f ]  
I P A D R S =   a a a . a a a . a a a . a a a   [ c r / l f ]  
P O R T = p p p p p   [ c r / l f ]  
E O L = e   [ c r / l f ]  
T Y P E = t   [ c r / l f ]  
      :  
T Y P E = t   [ c r / l f ]  
[ c r / l f ]
```

标注	值	范围、含义
m	通用以太网端口编号	0 ~ 7
n	模式	0 : 服务器 / 1 : 客户端
aaa	IP 地址	0 ~ 255
ppppp	端口编号	0 ~ 65535
e	终端字符编码	0 : CRLF / 1 : CR
t	端口类型	0 : TCP

 MEMO**写入文件的模式选择客户端时**

- IP 地址 / 端口编号 : 设置待连接服务器的 IP 地址和端口编号。

写入文件的模式选择服务器时

- IP 地址 : 使用控制器中设置的 IP 地址进行通信, 因此无需设置 IP 地址。
- 端口编号 : 设置与控制器设置内容不同的编号。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND GEP TO CMU	从通信端口输出所有通用以太网端口
SEND CMU TO GEP	从通信端口输入所有通用以太网端口
应答：	
RUN [cr/lf]	
GP0 [cr/lf]	
MODE=1 [cr/lf]	
IPADRS=192.168.0.1 [cr/lf]	
PORT=100 [cr/lf]	
EOL=0 [cr/lf]	
TYPE=0 [cr/lf]	
GP1 [cr/lf]	
MODE=1 [cr/lf]	
IPADRS=192.168.0.100 [cr/lf]	
PORT=200 [cr/lf]	
EOL=0 [cr/lf]	
TYPE=0 [cr/lf]	
[cr/lf]	
END [cr/lf]	

12.2 输入输出类型单位

读出	✓
写入	✓ 有限制 *

格式

ioNM()

含义 表示指定输入输出类型单位的输入输出名称。

DATA FORMAT

```

ioNMpp(b)=assssssssssssssssss [cr/lf]
:
ioNMpp(b)=assssssssssssssssss [cr/lf]
[cr/lf]

```

标注	值	范围
io	输入输出类型	DI / DO / SI / SO
pp	端口番号	2 ~ 7, 10 ~ 15 * 可写入的输入输出类型和端口编号 DI: 至端口 14 / DO: 至端口 10 / SI, SO: 至端口 15
b	比特编号	0 ~ 7
a	名称数据 (第 1 个字符)	仅可使用半角字母。 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。
SS...SS	名称数据 (第 2 个字符之后)	· 仅可使用半角字母数字或下划线 (_)。 · 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。 · 名称数据最多为 16 个字符。 · 写入数据时超过 16 个字符时, 第 17 个字符之后将被删除。

- 在文件的末尾仅附加表示文件结束的[cr/lf]行。

SAMPLE	说明
SEND DONM() TO CMU	……从通信端口输出指定输入输出类型的输入输出名称
应答： <pre> RUN [cr/lf] DONM2(0)=DO_PORT2_0 [cr/lf] DONM2(1)=DO_PORT2_1 [cr/lf] : DONM10(6)=DO_PORT10_6 [cr/lf] DONM10(7)=DO_PORT10_7 [cr/lf] [cr/lf] END [cr/lf] </pre>	

12.3 输入输出端口单位

读出	✓
写入	✓ 有限制 *

格式
ioNMpp()

含义 表示指定输入输出类型单位的输入输出名称。

DATA FORMAT
ioNMpp(b)=assssssssssssssss [cr/lf] : ioNMpp(b)=assssssssssssssss [cr/lf] [cr/lf]

标注	值	范围
io	输入输出类型	DI / DO / SI / SO
pp	端口番号	2 ~ 7, 10 ~ 15 * 可写入的输入输出类型和端口编号 DI: 至端口 14 / DO: 至端口 10 / SI, SO: 至端口 15
b	比特编号	0 ~ 7
a	名称数据 (第 1 个字符)	仅可使用半角字母。 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。
ss...ss	名称数据 (第 2 个字符之后)	· 仅可使用半角字母数字或下划线 (_)。 如果输入其它字符, 将出现 [4.202 : 输入格式异常] 的错误。 · 名称数据最多为 16 个字符。 写入数据时超过 16 个字符时, 第 17 个字符之后将被删除。

■ 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND DONM2() TO CMU 应答 : RUN [cr/lf] DONM2(0)=DO_PORT2_0 [cr/lf] DONM2(1)=DO_PORT2_1 [cr/lf] : DONM2(6)=DO_PORT2_6 [cr/lf] DONM2(7)=DO_PORT2_7 [cr/lf] [cr/lf] END [cr/lf]	……从通信端口输出指定输入输出端口的输入输出名称

9

10

13 区域判定输出定义文件

13.1 所有区域判定输出定义

读出	✓	读出所有区域判定输出定义。
写入	✓	通过区域判定输出定义编号写入。

形式

ACO

含义 表示所有区域判定输出定义。

DATA FORMAT

```
ACm=r,p1,p2,t,n,l [cr/lf]
ACm=r,p1,p2,t,n,l [cr/lf]
:
ACm=r,p1,p2,t,n,l [cr/lf]
ACm=r,p1,p2,t,n,l [cr/lf]
[cr/lf]
```

标注	值	范围、含义
m	区域判定输出编号	0 ~ 31
r	机器人编号	0 ~ 4 (0 : 无效)
p1	比较坐标点编号 1	0 ~ 29999
p2	比较坐标点编号 1	0 ~ 29999
t	端口类型	0 : DO (SO) / 1 : DO / 2 : SO / 3 : MO
n	端口编号	20 ~ 277
l	逻辑	0 : 关闭 / 1 : 开启

- 在文件的末尾仅附加表示文件结束的[cr/lf]行。

SAMPLE	说明
SEND ACO TO CMU	……从通信端口输出所有区域判定输出
SEND CMU TO ACO	……从通信端口输入出所有区域判定输出
应答 :	
RUN [cr/lf]	
AC0=1,0,1,0,20,0 [cr/lf]	
AC1=2,100,110,0,50,0 [cr/lf]	
:	
AC30=1,20,21,0,20,0 [cr/lf]	
AC31=1,50,51,0,100,0 [cr/lf]	
[cr/lf]	
END [cr/lf]	

13.2 区域判定输出定义单位

读出	✓	
写入	✓	通过区域判定输出定义编号写入。

格式
ACm

含义 表示指定区域判定输出定义。

DATA FORMAT
ACm=r,p1,p2,t,n,l [cr/lf]

标注	值	范围、含义
m	区域判定输出编号	0 ~ 31
r	机器人编号	0 ~ 4 (0 : 无效)
p1	比较坐标点编号 1	0 ~ 29999
p2	比较坐标点编号 1	0 ~ 29999
t	端口类型	0 : DO (SO) / 1 : DO / 2 : SO / 3 : MO
n	端口编号	20 ~ 277
l	逻辑	0 : 关闭 / 1 : 开启

SAMPLE	说明
SEND AC0 TO CMU	………从通信端口输出指定区域判定输出
SEND CMU TO AC0	………从通信端口输入指定区域判定输出
应答 : RUN [cr/lf] AC0=1,0,1,0,20,0 [cr/lf] END [cr/lf]	

8

9

10

11

12

13

14实时输出文件

实时输出文件是在软件版本 Ver.1.73 以上有效。

14.1 所有实时输出设置

读出	✓	读出所有实时输出设置。
写入	✓	通过实时输出编号写入。

形式
RTO

含义 表示所有实时输出设置。

DATA FORMAT
RTz = n, o, p [cr/lf] : RTz = n, o, p [cr/lf] [cr/lf]

标注	值	范围、含义
z	实时输出端口编号	24 ~ 127
n	输出字符串	
o	选项 1	参照 本章“14.3 实时输出设置一览”
p	选项 2	

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND RTO TO CMU	……从通信端口输出所有实时输出设置
应答 :	
RUN [cr/lf]	
RT24=UCURPLS,1,1[cr/lf]	
RT25=LCURPLS,1,1[cr/lf]	
:	
RT127=LPWRONCNT[cr/lf]	
[cr/lf]	
END [cr/lf]	

读出	✓
写入	✓

格式
RTz

含义 表示指定实时输出设置。

DATA FORMAT
RTz = n, o, p [cr/lf] [cr/lf]

标注	值	范围、含义
z	实时输出端口编号	24 ~ 127
n	输出字符串	
o	选项 1	参照 本章“14.3 实时输出设置一览”
p	选项 2	

■ 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND RT24 TO ETH从通信端口输出指定实时输出设置	
应答： RUN [cr/lf] RT24=UCURPLS,1,1 [cr/lf] [cr/lf] END [cr/lf]	

14.3 实时输出设置一览

输出项目 编号 (m)	输出字符串 (n)	输出项目	单位	显示格式	选项 1 (o)		选项 2 (p)	
					设置对象	设置范围	设置对象	设置范围
1	UCURPLS	当前脉冲 高位	pulse	LONG 高位	机器人	1 ~ 4	轴	1 ~ 6
2	LCURPLS	当前脉冲 低位	pulse	LONG 低位	机器人	1 ~ 4	轴	1 ~ 6
3	UREFPLS	指令脉冲 高位	pulse	LONG 高位	机器人	1 ~ 4	轴	1 ~ 6
4	LREFPLS	指令脉冲 低位	pulse	LONG 低位	机器人	1 ~ 4	轴	1 ~ 6
5	UCURVEL	当前速度 高位	pulse/10ms	LONG 高位	机器人	1 ~ 4	轴	1 ~ 6
6	LCURVEL	当前速度 低位	pulse/10ms	LONG 低位	机器人	1 ~ 4	轴	1 ~ 6
7	UREFVEL	指令速度 高位	pulse/10ms	LONG 高位	机器人	1 ~ 4	轴	1 ~ 6
8	LREFVEL	指令速度 低位	pulse/10ms	LONG 低位	机器人	1 ~ 4	轴	1 ~ 6
9	FDBCUR	当前电流	0.01Arms	SHORT	机器人	1 ~ 4	轴	1 ~ 6
10	REFCUR	指令电流	0.01Arms	SHORT	机器人	1 ~ 4	轴	1 ~ 6
11	INPOS	公差 (1)	—	SHORT	机器人	1 ~ 4	轴	1 ~ 6
12	OUTPOS	OUT 有效位置 (1)	—	USHORT	机器人	1 ~ 4	轴	1 ~ 6
13	MTRDUTY	马达负载率	%	USHORT	机器人	1 ~ 4	轴	1 ~ 6
14	DRVDTY	驱动器负载率	%	USHORT	机器人	1 ~ 4	轴	1 ~ 6
15	DRVMOD	驱动器模式 (2)	—	USHORT	机器人	1 ~ 4	轴	1 ~ 6
16	GRPPLS	夹爪 脉冲位置	pulse	LONG	夹爪	1 ~ 4	—	—
17	UGRPPOS	夹爪 毫米位置 高位	0.00001mm	LONG 高位	夹爪	1 ~ 4	—	—
18	LGRPPOS	夹爪 毫米位置 低位	0.00001mm	LONG 低位	夹爪	1 ~ 4	—	—
19	GRPSTS	夹爪状态 (3)	—	USHORT	夹爪	1 ~ 4	—	—
20	USGI	SGI 高位	—	LONG 高位	SGI	0 ~ 31	—	—
21	LSGI	SGI 低位	—	LONG 低位	SGI	0 ~ 31	—	—
22	USGR	SGR 高位	0.001	LONG 高位	SGR	0 ~ 31	—	—
23	LSGR	SGR 低位	0.001	LONG 低位	SGR	0 ~ 31	—	—
24	TEMP	控制器温度	℃	USHORT	—	—	—	—
25	TASK	执行中任务编号	—	USHORT	—	—	—	—
26	STEP	程序执行步骤编号	—	USHORT	任务	1 ~ 16	—	—
27	TSKSTS	程序任务状态 (4)	—	USHORT	任务	1 ~ 16	—	—
28	ALRMGRP	警报组	—	USHORT	—	—	—	—
29	ALRMCTG	警报分类	—	USHORT	—	—	—	—
30	UCCORPOS	正交坐标 当前位置 高位	0.001mm · deg	LONG 高位	机器人	1 ~ 4	轴	1 ~ 6
31	LCCORPOS	正交坐标 当前位置 低位	0.001mm · deg	LONG 低位	机器人	1 ~ 4	轴	1 ~ 6
32	UCCORSPD	正交坐标 当前速度 高位	mm/s	LONG 高位	机器人	1 ~ 4	轴	1 ~ 6
33	LCCORSPD	正交坐标 当前速度 低位	mm/s	LONG 低位	机器人	1 ~ 4	轴	1 ~ 6
34	UMOVDIS	移动距离 高位	m · 1000deg	ULONG 高位	机器人	1 ~ 4	轴	1 ~ 6
35	LMOVDIS	移动距离 低位	m · 1000deg	ULONG 低位	机器人	1 ~ 4	轴	1 ~ 6
36	UPWRONCNT	电源 ON 次数 高位	次	ULONG 高位	—	—	—	—
37	LPWRONCNT	电源 ON 次数 低位	次	ULONG 低位	—	—	—	—
38	UMTRONCNT	马达电源 ON 次数 高位	次	ULONG 高位	—	—	—	—
39	LMTRONCNT	马达电源 ON 次数 低位	次	ULONG 低位	—	—	—	—
40	USRVONCNT	伺服 ON 次数 高位	次	ULONG 高位	机器人	1 ~ 4	轴	1 ~ 6
41	LSRVONCNT	伺服 ON 次数 低位	次	ULONG 低位	机器人	1 ~ 4	轴	1 ~ 6
42	USRVONTIM	伺服 ON 时间 高位	s	ULONG 高位	机器人	1 ~ 4	轴	1 ~ 6
43	LSRVONTIM	伺服 ON 时间 低位	s	ULONG 低位	机器人	1 ~ 4	轴	1 ~ 6
44	UOPRTIM	总运行时间 高位	s	ULONG 高位	—	—	—	—
45	LOPRTIM	总运行时间 低位	s	ULONG 低位	—	—	—	—

RCX3 系列控制器的主机 CPU 软件 V1.73 以上版本在首次接通电源时，马达累计数据（伺服 ON 次数、伺服 ON 时间、移动距离）和系统累计数据（电源 ON 次数、马达电源 ON 次数、总运行时间）的数值会初始化，之后即便切断电源，数值仍将保持并累计。

马达和系统的累计数据可分别使用远程命令进行复位。[参照](#) 现场网络手册“远程命令篇”

下表显示了每个状态的值分配。

■ 1 INPOS (公差) / OUTPOS (OUT 有效位置)

输出项目	执行移动命令时，指定轴的状态	值	
		0	1
公差	轴是否在目标位置的公差范围内	范围外	范围内
OUT 有效位置	轴是否在目标位置的 OUT 有效范围内		

■ 2 DRVMOD (驱动器模式)

驱动器模式	值 (※)
伺服保持	0x0030
伺服 OFF	0x0031
伺服开放	0x0032
运行中	0x0033
	0x0034
	0x0035
原点归复	0x0036
伺服制动	0x0037
分析模式	0x0038
速度控制方式	0x0039
严重警报	0x0050
	0x0051
磁极估计	0x0052

※ 值以十六进制表示法。

■ 3 GRPSTS (夹爪状态)

Bit	信号名称	值	
		0	1
7	伺服状态	OFF	ON
6	原点归复状态	未完成	完成
5	没用过		
4	READY 信号	准备中	正常
3	ZON 信号	超出范围	范围内
2	HOLD 信号	开放	夹持中
1	INPOS 信号	超出范围	范围内
0	BUSY 信号	完成	运行中

■ 4 TSKSTS (程序任务状态)

任务状态	值 (※)
未录入状态	0x0000
执行状态 (RUN)	0x0002
等待状态 (WAIT)	0x0004
强制待机状态 (SUSPEND)	0x0008
停止状态 (STOP)	0x0010

※ 值以十六进制表示法。

15

ALL 文件

15.1 所有 ALL 文件

读出	✓
写入	✓

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。



补充 有关各文件的详细说明，请参阅各文件的说明。

DATA FORMAT
[PGM] 所有程序格式 NAME=< 程序名 > PGN=mmm aaaaaaaaaaaa [cr/lf] : aaaaaaaaaaaa [cr/lf] [cr/lf] [PNT] 所有坐标点格式 Pmmmm=fxxxxxx fyxxxxx fzxxxxx faaaaaa fbbbbbb t [cr/lf] : Pmmmm=fxxxxxx fyxxxxx fzxxxxx faaaaaa fbbbbbb t [cr/lf] [cr/lf] [PCM] 所有坐标点注释格式 PCmmmm= sssssssssssssssss [cr/lf] : PCmmmm= sssssssssssssssss [cr/lf] [cr/lf] [PNM] 所有坐标点名称的格式 PNmmmm= asssssssssssssssss [cr/lf] : PNmmmm= asssssssssssssssss [cr/lf] [cr/lf] [PRM] 所有参数格式 / 参数标签 / [cr/lf] RC=xxxxxx [cr/lf] : # 参数标签 # [cr/lf] R?=xxxxxx [cr/lf] [cr/lf] [SFT] 所有位移形式 Sm= fxxxxxx fyxxxxx fzxxxxx frrrrrr [cr/lf] : SMm= fxxxxxx fyxxxxx fzxxxxx frrrrrr [cr/lf] [cr/lf]

```
[HND] .....所有机械手形式
Hm= n, fxxxxxxx, fyyyyyyy, fzzzzzzz , {R} [cr/lf]
:
Hm= n, fxxxxxxx, fyyyyyyy, fzzzzzzz , {R} [cr/lf]
[cr/lf]
[PLT] .....所有托盘形式
PLm [cr/lf]
:
P[5]= fxxxxxxx fyyyyyyy fzzzzzzz frrrrrr faaaaaa fbbbbb t [cr/lf]
[cr/lf]
[GEP] .....所有通用以太网端口形式
MODE=n [cr/lf]
:
TYPE=t [cr/lf]
[cr/lf]
[ION] .....所有输入输出名称格式
ioNMpp(b)=assssssssssssssss [cr/lf]
:
ioNMpp(b)=assssssssssssssss [cr/lf]
[cr/lf]
[ACO] .....所有区域判定输出格式
ACm=r,p1,p2,t,n,l [cr/lf]
:
ACm=r,p1,p2,t,n,l [cr/lf]
[cr/lf]
[END] .....ALL 文件的结束
```

 MEMO

- 仅控制器中保存有数据的项目，进行文件读入。
- 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	说明
SEND ALL TO CMU	……从通信端口输出指定区域判定输出
SEND CMU TO ALL	……从通信端口输入所有系统

16 程序目录文件

16.1 所有程序目录

读出	✓	读出所有程序目录信息。
写入	×	无法指定写入文件。

格式

DIR

含义 表示整个程序目录。

DATA FORMAT

```
nnn, yy/mm/dd, hh:mm, bbbbbbb, llll, xx, ff, sssss...ssssssss [cr/lf]
:
nnn, yy/mm/dd, hh:mm, bbbbbbb, llll, xx, ff, sssss...ssssssss [cr/lf]
[cr/lf]
```

标注	值	范围、含义
nnn	程序的编号	1 ~ 100
yy/mm/dd	更新日期	
hh:mm	更新时间	
bbbbbbb	使用字节数 (7 桁)	
xx	文件属性	RW : 读写 RO : 只读 H : 隐藏文件
ff	标志	m : 主程序 c : 当前程序 s : 逻辑控制程序
sss...sssss	程序名	· 半角字母数字或下划线 (_) · 程序名最多 32 个字符

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND DIR TO CMU	... 从通信端口输出已创建的所有程序信息
应答 :	
RUN [cr/lf]	
1, 15/01/10, 10:14, 100, 24, RW, m, SAMPLE1	[cr/lf]
2, 15/01/18, 18:00, 50, 18, RO, , SAMPLE2	[cr/lf]
3, 15/02/11, 20:15, 200, 58, RW, c, SAMPLE3	[cr/lf]
4, 15/02/11, 19:03, 28, 15, H, , SAMPLE4	[cr/lf]
10, 15/03/02, 20:21, 592, 288, RW, , SAMPLE10	[cr/lf]
24, 15/01/18, 13:19, 10, 3, RW, , SAMPLE24	[cr/lf]
[cr/lf]	
END [cr/lf]	

16.2 程序单位

读出	✓
写入	×

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

nnn, yy/mm/dd, hh:mm, bbbbbbb, llll, xx, ff, sssss...sssssssss [cr/lf]

标注	值	范围、含义
nnn	程序的编号	1 ~ 100
yy/mm/dd	更新日期	
hh:mm	更新时间	
bbbbbbb	使用字节数 (7 桁)	
xx	文件属性	RW : 读写 RO : 只读 H : 隐藏文件
ff	标志	m : 主程序 c : 当前程序 s : 逻辑控制程序
sss...ssssss	程序名	· 半角字母数字或下划线 (_) · 程序名最多 32 个字符

SAMPLE

说明

SEND <<SAMPLE1>> TO CMU从通信端口输出指定程序信息

应答 :

RUN [cr/lf]

1, 15/01/10, 10:14, 100, 24, RW, m, SAMPLE1 [cr/lf]

END [cr/lf]

17 参数目录文件

17.1 所有参数目录

读出	✓	读出所有参数目录信息。
写入	×	无法在写入文件中指定。

格式

DPM

含义 表示所有参数目录。

DATA FORMAT

```
\mmmmmmmm\ a m n1 n2 n3 ... n10 n11 n12 uuuuuu [cr/lf]
/mmmmmmmm/ a m n1 n2 n3 ... n10 n11 n12 uuuuuu [cr/lf]
#mmmmmmmm# a m n1 n2 n3 ... n10 n11 n12 uuuuuu [cr/lf]
[cr/lf]
```

标注	值	范围、含义
mmmmmmmm	参数标签	字母数字和部分符号少于 8 个字符
a	属性	
m	输入方式 (*)	0 : 直接输入 / 1 ~ 12 : 选择
n	输入范围 (*)	n1 : 最小值 / n2 : 最大值
uuuuuu	单位	
*	m : 0 n : n1 (最小值) / n2 (最大值) m : 1 ~ 12 n : 列举 m 中输入的值的选项 (例) m 为 4 时的 m 及 n 4 0 1 2 3	

MEMO

- 参数标签两侧有 \ 的参数为控制器构成参数。编辑时请格外注意。
- 参数标签两侧有 # 的参数将影响机器人的控制。编辑时请格外注意。
- 部分计算机中, 反斜杠 (\) 可能显示为 ¥ 标记。

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND DPM TO CMU 应答 : RUN [cr/lf] 'PRM(0) [cr/lf] \CNTTYP\ 16460 0 0 2147493647 [cr/lf] \YCEADR\ 16396 0 0 99 [cr/lf] \DRVASGN\ 16398 0 0 9906 [cr/lf] : /ARMTYP/ 0 3 0 1 2 [cr/lf] #CPVMAX# 16 0 1 32767 mm/s [cr/lf] : /IOORGOUT/ 2052 0 0 27 [cr/lf] /IOSRVOUT/ 2052 0 0 27 [cr/lf] /GRPORGIN/ 2052 0 0 27 [cr/lf] [cr/lf] END [cr/lf]	从通信端口输出已创建的所有参数信息

18.1 机器参照（传感器轴、撞块轴）

读出	✓
写入	×

格式
MRF

含义 表示原点复归方式设置为传感方式或撞块方式的轴的所有机器参照。

DATA FORMAT
RnA=mmm,mmm,mmm,mmm,mmm,mmm [cr/lf] : RnA= mmm,mmm,mmm,mmm,mmm,mmm [cr/lf] [cr/lf]

标注	值	范围
n	机器人编号	1 ~ 4
mmm	机器参照值	0 ~ 100

 MEMO

仅读出连接轴的机器参照值。
例) 将机器人 1 的第 1 轴~第 6 轴与机器人 2 的第 1 轴及第 3 轴进行连接时
R1A=mmm,mmm,mmm,mmm,mmm,mmm
R2A= mmm,mmm

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND MRF TO CMU	……从通信端口输出所有机器参照
应答: RUN [cr/lf] R1A=53,47,58,25,55,59 [cr/lf] : R4A=52,58,41,38,61,50 [cr/lf] [cr/lf] END [cr/lf]	

18.2 机器参照（标记轴）

读出	✓
写入	×

格式
ARP

含义 表示原点复归方式设置为标记方式的轴的所有机器参照。

DATA FORMAT
RnA=mmm,mmm,mmm,mmm,mmm,mmm [cr/lf] : RnA=mmm,mmm,mmm,mmm,mmm,mmm [cr/lf] [cr/lf]

标注	值	范围
n	机器人编号	1 ~ 4
mmm	机器参照值	0 ~ 100

 MEMO

仅读出连接轴的机器参照值。
例)将机器人 1 的第 1 轴~第 6 轴与机器人 2 的第 1 轴及第 3 轴进行连接时
R1A=mmm,mmm,mmm,mmm,mmm,mmm
R2A= mmm, ,mmm

- 在文件的末尾仅附加表示文件结束的[cr/lf]行。

SAMPLE	说明
SEND ARP TO CMU	……从通信端口输出所有机器参照
应答: RUN [cr/lf] R1A=53,47,58,25,55,59 [cr/lf] : R4A=52,58,41,38,61,50 [cr/lf] [cr/lf] END [cr/lf]	

读出	✓(*)
写入	×

格式
ACD

含义 显示所有累计数据。累计数据是指从控制器初次接通电源时开始的电源 ON 次数和马达的伺服 ON 次数、移动距离等的累计值。详情请参阅本章“14.3 实时输出设置一览”及现场网络手册“实时输出”。

DATA FORMAT
/PWRONCNT/[cr/lf] RC=xxxxxxxxxx[cr/lf] /MPWRONCNT/[cr/lf] RC=xxxxxxxxxx[cr/lf] /OPTIME/[cr/lf] RC=xxxxxxxxxx[cr/lf] /SRVONCNT/[cr/lf] R?A=xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx[cr/lf] /SRVONTIME/[cr/lf] R?A=xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx[cr/lf] /DISTTRVLD/[cr/lf] R?A=xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx[cr/lf] [cr/lf]

标注	值	范围、含义
RC	所有控制器	
R?	机器人指定	
?	机器人编号	1 ~ 4
A	所有轴	各轴的数据用逗号进行间隔
xxxxxxxxxx	数值	0 ≤ xxxxxxxxxx ≤ 4294967295

■ 在文件的末尾仅附加表示文件结束的[cr/lf]行。

SAMPLE	说明
SEND ACD TO CMU	……从通信端口输出所有累计数据
应答：	
RUN [cr/lf]	
/PWRONCNT/[cr/lf]	……控制器的电源 ON 次数
RC=1[cr/lf]	
/MPWRONCNT/[cr/lf]	……控制器的马达电源 ON 次数
RC=0[cr/lf]	
/OPTIME/[cr/lf]	……控制器的总运行时间
RC=2500[cr/lf]	
/SRVONCNT/[cr/lf]	……机器人 1 的所有轴的伺服 ON 次数
R1A=0,0,0,0[cr/lf]	
/SRVONTIME/[cr/lf]	……机器人 1 的所有轴的伺服 ON 时间
R1A=0,0,0,0[cr/lf]	
/DISTTRVLD/[cr/lf]	……机器人 1 的所有轴的移动距离
R1A=0,0,0,0[cr/lf]	
[cr/lf]	
END [cr/lf]	

*) 累计数据：在软件版本 Ver.1.73 以上有效 累计数据的读出：在软件版本 Ver.2.04 以上有效

读出	✓
写入	×

格式

CFG

含义 表示所有系统构成信息。

DATA FORMAT

```
Cm:nnnn, s, b, kkkkk, ff-ff-ff-ff-ff-ff [cr/lf]
Cm:nnnn, s, b, kkkkk, ff-ff-ff-ff-ff-ff [cr/lf]
:
Rr:aaaa,hhhhhh [cr/lf]
Rr:aaaa,hhhhhh [cr/lf]
[cr/lf]
```

标注	值	范围、含义
m	控制器编号	1 ~
nnn	控制器识别编号	
s	规格	G:CE 规格 / L:标准规格
b	制动器电源	I:内部 / E:外部
kkkkk	内存大小	
ff	MAC 地址	
r	机器人编号	1 ~ 4
aaaa	机器人识别编号	
hhhhh	连接轴编号	

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE

说明

SEND CFG TO CMU从通信端口输出所有系统构成

应答:

```
RUN [cr/lf]
C1:340,L,I,2.1MB,00-04-C6-FF-83-12[cr/lf]
R1:MULTI,1234[cr/lf]
[cr/lf]
END [cr/lf]
```

读出	✓
写入	×

格式
VER

含义 表示版本信息。

DATA FORMAT
Cm:cv, cr-mv-dv1, dr1/dv2, dr2 [cr/lf] : Cm:cv, cr-mv-dv1, dr1/dv2, dr2 [cr/lf] [cr/lf]

标注	值	范围
m	控制器编号	1 ~
cv	主机版本	
cr	主机的修订版本号 (Rxxxx)	
mv	PLD 的版本 (Vx.xx)	
dv?	驱动器版本 (Vx.xx)	? : 1、2
dr?	驱动器的修订版本号 (Rxxxx)	? : 1、2

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND VER TO CMU 应答 : RUN [cr/lf] C1:V1.22,R0191-V1.000-V1.09,R0015/V1.09,R0015 [cr/lf] C2:V1.22,R0191-V1.000-V1.09,R0015/V1.09,R0015 [cr/lf] C3:V1.22,R0191-V1.000-V1.09,R0015/V1.09,R0015 [cr/lf] C4:V1.22,R0191-V1.000-V1.09,R0015/V1.09,R0015 [cr/lf] [cr/lf] END [cr/lf]	从通信端口输出版本信息

读出	✓
写入	×

格式
OPT

含义 表示所有选件板。

DATA FORMAT
CmOn : aaaaaa,Vb.bb [cr/lf] CmOn : aaaaaa,Vb.bb [cr/lf] : CmOn : aaaaaa,Vb.bb [cr/lf] CmOn : aaaaaa,Vb.bb [cr/lf] [cr/lf]

标注	值	范围、含义
m	控制器编号	1 ~
n	控制器内选件板编号	插槽编号 : 1 ~ 4
aaaaaa	选件板名	
b.bb	选件板版本	

■ 在文件的末尾仅附加表示文件结束的[cr/lf]行。

SAMPLE	说明
SEND OPT TO CMU	从通信端口输出所有选件板
应答 :	
RUN [cr/lf]	
C101:Gripper,V0.32 [cr/lf]	
C102:Gripper,V0.32 [cr/lf]	
[cr/lf]	
END [cr/lf]	

读出	✓
写入	×

格式
SCK

含义 表示自我诊断。

DATA FORMAT
gg.bbb:mmmm [cr/lf] gg.bbb:mmmm [cr/lf] : gg.bbb:mmmm [cr/lf] gg.bbb:mmmm [cr/lf] [cr/lf]

标注	值	含义
gg	警报分类编号	
bbb	警报分类编号	
mmmm	警报发生位置	RC：所有控制器 R?：机器人（?：机器人编号） C?：控制器（?：控制器编号） A?：轴（?：轴编号） M?：驱动器（?：驱动器编号） O?：选件板（?：控制器内选件板编号） T?：任务（?：任务编号） ETH：以太网 CMU：RS232C

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND SCK TO CMU	……从通信端口输出自我诊断
应答：	
RUN [cr/lf]	
12.600:C1M1 [cr/lf]	
12.600:C1M2 [cr/lf]	
12.600:C1M3 [cr/lf]	
12.600:C1M4 [cr/lf]	
[cr/lf]	
END [cr/lf]	

24

报警履历文件

读出	✓
写入	×

格式
LOG

含义 表示所有报警履历。

DATA FORMAT
<pre>nnn:yy/mm/dd, hh:mm:ss, gg.bbb : aaaa,c, eee : ffff, iiii, jjjjjjjj, kkkkkkkk, llllllll, ooooooooo, pppppppp, pppppppp, pppppppp, pppppppp, pppppppp, pppppppp, q [cr/lf] nnn:yy/mm/dd, hh:mm:ss, gg.bbb : aaaa,c, eee : ffff, iiii, jjjjjjjj, kkkkkkkk, llllllll, ooooooooo, pppppppp, pppppppp, pppppppp, pppppppp, pppppppp, pppppppp, q [cr/lf] : nnn:yy/mm/dd, hh:mm:ss, gg.bbb : aaaa,c, eee : ffff, iiii, jjjjjjjj, kkkkkkkk, llllllll, ooooooooo, pppppppp, pppppppp, pppppppp, pppppppp, pppppppp, pppppppp, q [cr/lf] [cr/lf]</pre>

标注	值	范围、含义
nnn	报警履历编号	1 ~ 500
yy/mm/dd	报警日期	
hh:mm:ss	报警时间	
gg	警报分类编号	
bbb	警报分类 编号	
aaaa	警报发生位置	RC : 所有控制器 R? : 机器人 (? : 机器人编号) C? : 控制器 (? : 控制器编号) A? : 轴 (? : 轴编号) M? : 驱动器 (? : 驱动器编号) O? : 选件板 (? : 控制器内选件板编号) T? : 任务 (? : 任务编号) ETH : 以太网 CMU : RS232C
c	运行模式	I : 无效 M : 手动 A : 自动 (手持编程器) O : 自动 (其他)
eee	程序编号	
ffff	程序执行行	
iiii	坐标点编号	
jjjjjjj	并行输入	端口 0 ~ 3 (十六进制)
kkkkkkkk	并行输出	端口 0 ~ 3 (十六进制)
llllllll	串行输入	端口 0 ~ 3 (十六进制)
oooooooo	串行输出	端口 0 ~ 3 (十六进制)
pppppppp	出错位置	A1 ~ A6
q	手系统	0 : 无 / 1 : 右手系统 / 2 : 左手系统

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND LOG TO CMU	……从通信端口输出报警履历
应答 :	
RUN [cr/lf]	
1:15/03/30,08:23:05,1.100:RC,0,:,0,00000000,00000012,00000000,00000112,,,,,,	[cr/lf]
2:15/03/30,08:23:05,5.288:RC,0,:,0,00000000,00000010,00000000,00000110,,,,,,	[cr/lf]
:	
500:15/03/18,10:23:04,5.228:T01,0,17:3,,00000000,00000010,00000000,00000110,40119,100000,99996,39375,0,0,0	
[cr/lf]	
[cr/lf]	
END [cr/lf]	

25

存储器剩余容量文件

读出	✓
写入	×

格式
MEM

含义 表示存储器剩余容量。

DATA FORMAT
PGM+PNT AREA=mmmmmmm/nnnnnnnn[cr/lf] VAR AREA=xxxxx/yyyy[cr/lf] [cr/lf]

标注	值
mmmmmmm	程序 + 坐标点区域的存储器剩余容量
nnnnnnn	程序 + 坐标点区域的内存容量合计
xxxxx	变量区域的存储器剩余容量
yyyyy	变量区域的内存容量合计

■ 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND MEM TO CMU …… 从通信端口输出存储器剩余容量	
应答： RUN [cr/lf] PGM+PNT AREA=2088547 / 2100000 [cr/lf] VAR AREA=23220 / 24000 [cr/lf] [cr/lf] END [cr/lf]	

26.1 动态变量

26.1.1 所有动态变量

读出	✓	读出所有动态变量。
写入	✓	通过指定动态变量写入。

格式
VAR

含义 表示所有动态变量。

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]

标注	值	范围、含义
变量名	程序内部定义的全局变量。	变量名通过采用英文字母数字或"_"的 32 字符以内的字符串表示。
t	变量类型	!: 实数、%: 整数、\$: 字符串
xxxxxx	变量的值	整数型: 2147483647 ~ 2147483647 的整数值 实数型: 整数部分 + 小数部分为 7 位数以下的数值 字符型: 255 个字符以下的字符串

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND VAR TO CMU SEND CMU TO VAR	从通信端口输出所有动态变量 从通信端口输入所有动态变量
应答: RUN [cr/lf] A%=150 [cr/lf] B!=1.0234E1 [cr/lf] C1\$="SAMPLE1" [cr/lf] C2\$="SAMPLE2" [cr/lf] [cr/lf] END [cr/lf]	

变量文件

26.1.2 动态变量单位

读出	✓
写入	✓

格式

变量名 t

含义 表示 1 个动态变量。

DATA FORMAT

xxxxxx [cr/lf]

标注	值	范围、含义
变量名	程序内部定义的全局变量。	变量名通过采用英文字母数字或 "_" 的 32 字符以内的字符串表示。
t	变量类型	!: 实数、%: 整数、\$: 字符串
xxxxxx	变量的值	整数型: -2147483647 ~ 2147483647 的整数值 实数型: 整数部分 + 小数部分为 7 位数以下的数值 字符型: 255 个字符以下的字符串

MEMO

动态变量在执行程序时录入。

在执行程序时，可对动态全局变量进行登录。无法参照未录入的变量。

SAMPLE 1	说明
SEND A% TO CMU [cr/lf] 应答: 150 [cr/lf]	……从通信端口输出指定动态变量 A%
SAMPLE 2	说明
SEND CMU TO A% [cr/lf] 应答: 300 [cr/lf] OK [cr/lf]	……从通信端口对指定动态变量 A% 输入数值 ……将数据输入至控制器 ……从控制器输出结果

26.2 静态变量

26.2.1 整数型静态变量 (SGI)

26.2.1.1 所有整数型静态变量

读出	✓	读出所有整数型静态变量。
写入	✓	通过指定整数型静态变量写入。

格式

SGI

含义 表示所有整数型静态变量。

DATA FORMAT

SGIn=xxxxxx [cr/lf]
SGIn=xxxxxx [cr/lf]
:
SGIn=xxxxxx [cr/lf]
[cr/lf]

标注	值	范围
n	整数型静态变量编号	0 ~ 31
xxxxxx	整数值	-2147483647 ~ 2147483647

- 在文件的末尾仅附加表示文件结束的[cr/lf]行。

SAMPLE	说明
SEND SGI TO CMU SEND CMU TO SGI	从通信端口输出所有整数型静态变量 从通信端口输入所有整数型静态变量
应答： RUN [cr/lf] SGI0=0 [cr/lf] SGI1=0 [cr/lf] : SGI31=0 [cr/lf] [cr/lf] END [cr/lf]	

变量文件

26.2.1.2 整数型静态变量单位

读出	✓
写入	✓

格式
SGIn

含义 表示指定整数型静态变量。

DATA FORMAT
xxxxxx [cr/lf]

标注	值	范围
n	整数型静态变量编号	0 ~ 31
xxxxxx	整数值	-2147483647 ~ 2147483647

SAMPLE	说明
SEND SGI1 TO CMU	………从通信端口输出指定整数型静态变量 (SGI1)
SEND CMU TO SGI1	………从通信端口输入指定整数型静态变量 (SGI1)
应答：	
RUN [cr/lf]	
0 [cr/lf]	
END [cr/lf]	

26.2.2 实数型静态变量 (SGR)

26.2.2.1 所有实数型静态变量

读出	✓	读出所有实数型静态变量。
写入	✓	通过指定实数型静态变量写入。

格式
SGR

含义 表示所有实数型静态变量。

DATA FORMAT
SGRn=xxxxxx [cr/lf]
SGRn=xxxxxx [cr/lf]
:
SGRn=xxxxxx [cr/lf]
[cr/lf]

标注	值	范围
n	实数型静态变量编号	0 ~ 31
xxxxxx	实数值	整数部分 + 小数部分为 7 位以下的数值

■ 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND SGR TO CMU SEND CMU TO SGR	从通信端口输出所有实数型静态变量 从通信端口输入所有实数型静态变量
应答： RUN [cr/lf] SGR0=0 [cr/lf] SGR1=0 [cr/lf] : SGR31=0 [cr/lf] [cr/lf] END [cr/lf]	

变量文件

26.2.2.2 实数型静态变量单位

读出	✓
写入	✓

格式
SGRn

含义 表示指定实数型静态变量。

DATA FORMAT
xxxxxx [cr/lf]

标注	值	范围
n	实数型静态变量编号	0 ~ 31
xxxxxx	实数值	整数部分 + 小数部分为 7 位以下的数值

SAMPLE	说明
SEND SGR1 TO CMU	………从通信端口输出指定实数型静态变量 (SGR1)
SEND CMU TO SGR1	………从通信端口输入指定实数型静态变量 (SGR1)
应 答 :	
RUN [cr/lf]	
0 [cr/lf]	
END [cr/lf]	

27.1 单个字符串

读出	✓	读出指定字符串。
写入	×	无法在写入文件中指定。

格式

“字符串”

含义 表示指定的字符串。

DATA FORMAT

sssss...sssss[cr/lf]

标注	值	范围
SSSS...SSSSS	字符串	255 个字符以内

■ 双引号 (") 输出应采用双引号 (") 连续记述。

SAMPLE	说明
SEND ""YAMAHA ROBOT"" TO CMU	……从通信端口输出指定字符串
应答： "YAMAHA ROBOT"	

28

数组变量文件

28.1 所有数组变量

读出	✓	读出所有数组变量。
写入	✓	通过指定数组变量写入。

格式
ARY

含义 表示所有数组变量。

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]

标注	值	范围、意味
变量名	在程序内部通过 DIM 语句定义的全局变量	变量名通过采用英文字母数字或 " _ " 的 32 字符以内的字符串表示
t	变量类型	!: 实数、%: 整数、\$: 字符串
l, m, n	数组自变量	
xxxxxx	变量的值	整数型: -2147483647 ~ 2147483647 的整数值 实数型: 整数部分 + 小数部分为 7 位数以下的数值 文字型: 255 个字符以下的字符串

- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND ARY TO CMU SEND CMU TO ARY	……从通信端口输出所有全局数组变量 ……从通信端口输入所有全局数组变量
应答: RUN [cr/lf] 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%(1,0)=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] END [cr/lf]	

28.2 数组变量单位

读出	✓
写入	✓

格式
变量名 t (l {,m {,n }})

含义 表示 1 个数组变量。

DATA FORMAT
xxxxxx [cr/lf]

标注	值	范围、意味
变量名	在程序内部通过 DIM 语句定义的全局变量	变量名通过采用英文字母数字或 " _ " 的 32 字符以内的字符串表示
t	变量类型	! : 实数、% : 整数、\$: 字符串
l, m, n	数组自变量	
xxxxxx	变量的值	整数型：-2147483647 ~ 2147483647 的整数值 实数型：整数部分 + 小数部分为 7 位数以下的数值 文字型：255 个字符以下的字符串

 MEMO

在 DIM 语句中定义的数组变量在程序编译时录入。无法参照未录入的数组变量。

SAMPLE 1	说明
SEND C1\$(2) TO CMU	………从通信端口输出指定数组变量 C1\$(2)
应答： YAMAHA ROBOT [cr/lf]	

SAMPLE 2	说明
SEND CMU TO C1\$(2)	………从通信端口输入指定数组变量 C1\$(2)
应答： OK[cr/lf]	

29

DI 文件

29.1 所有 DI

读出	✓	读出所有 DI 信息。
写入	×	无法在写入文件中指定。

格式
DI ()

含义 表示所有 DI (并行输入变量)。

DATA FORMAT
DI0 () = &Bnnnnnnnnn [cr/lf] DI1 () = &Bnnnnnnnnn [cr/lf] : DI27 () = &Bnnnnnnnnn [cr/lf] [cr/lf]
标注 值 范围
n 端口的状态 (二进制) 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	说明
SEND DI () TO CMU 应答 : RUN [cr/lf] 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] END [cr/lf]	从通信端口输出所有 DI

29.2 DI 端口单位

读出	✓	读出指定的 DI 端口。
写入	×	无法在写入文件中指定。

格式

DI*m* ()

含义 表示 1 个 DI 端口的状态。

DATA FORMAT

DI*m* () = &B*n**n**n**n**n**n**n**n* [*c r* / *l f*]

标注	值	范围
<i>m</i>	端口番号	0 ~ 7, 10 ~ 17, 20 ~ 27
<i>n</i>	端口的状态 (二进制)	0 或 1 (合计 8 位数)。 从左开始对应 <i>m</i> 7, <i>m</i> 6, ..., <i>m</i> 0 (<i>m</i> 为端口编号)

SAMPLE

说明

SEND DI5 () TO CMU从通信端口输出 DI5 端口

应答 :
RUN [*c r* / *l f*]
DI5 () = &B00000000 [*c r* / *l f*]
END [*c r* / *l f*]

30

DO 文件

30.1 所有 DO

读出	✓	读出所有 DO 信息。
写入	✓ 有限制 *	值将被写入至指定的 DO 端口。 * 无法写入至 DO0() 与 DO1()

格式
DO ()

含义 表示所有 DO (并行输出变量)。

DATA FORMAT
DO0 () = &Bnnnnnnnnn [cr/lf] DO1 () = &Bnnnnnnnnn [cr/lf] : DO27 () = &Bnnnnnnnnn [cr/lf] [cr/lf]
标注 值 范围
n 端口的状态 (二进制) 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	说明
SEND DO () TO CMU	从通信端口输出所有 DO
应答 :	
RUN [cr/lf]	
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]	
END [cr/lf]	

30.2 DO 端口单位

读出	✓	读出所有 DO 信息。
写入	✓ 有限制 *	值将被写入至指定的 DO 端口。 * 无法写入至 DO0() 与 DO1()

格式
DOm ()

含义 表示 1 个 DO 端口的状态。

 MEMO

无法写入至 DO0 () 与 DO1 ()。仅限参照。

读出文件

DATA FORMAT
DOm () = &Bnnnnnnnn [cr / lf]

写入文件

DATA FORMAT
&Bnnnnnnnn [cr / lf] 或 k [cr / lf]

标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
n	端口的状态 (二进制)	0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)
k	端口的状态 (十进制)	0 ~ 255 的整数

SAMPLE 1	说明
SEND DO5 () TO CMU 从通信端口输出 DO5 端口的状态	
应答 : RUN [cr / lf] DO5 () = &B00000000 [cr / lf] END [cr / lf]	

SAMPLE 2	说明
SEND CMU TO DO5 () 从通信端口输入 DO5 端口的状态 &B00000111	
应答 : OK [cr / lf]	

31

MO 文件

31.1 所有 MO

读出	✓	读出所有 MO 信息。
写入	✓ 有限制 *	值将被写入至指定的 MO 端口。 * 无法写入至 MO30() ~ MO37()。

格式
MO ()

含义 表示所有 MO (内部输出变量)。

DATA FORMAT
MO0 () = &Bnnnnnnnnn [cr/lf] MO1 () = &Bnnnnnnnnn [cr/lf] : MO37 () = &Bnnnnnnnnnn [cr/lf] [cr/lf]
标注 值 范围
n 端口的状态 (二进制) 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	说明
SEND MO () TO CMU	从通信端口输出所有 MO
应答 :	
RUN [cr/lf]	
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]	
:	
MO27 () = &B00000000 [cr/lf]	
MO30 () = &B00000000 [cr/lf]	
:	
MO36 () = &B00000000 [cr/lf]	
MO37 () = &B00000000 [cr/lf]	
[cr/lf]	
END [cr/lf]	

31.2 MO 端口单位

读出	✓	读出所有 MO 信息。
写入	✓ 有限制 *	值将被写入至指定的 MO 端口。 * 无法写入至 MO30() ~ MO37()。

格式
MOm ()

含义 表示 1 个 MO 端口的状态。

 MEMO

无法写入至 MO30() ~ MO37()。仅限参照。

读出文件

DATA FORMAT
MOm () = &Bnnnnnnnn [cr/lf]

写入文件

DATA FORMAT
&Bnnnnnnnn [cr/lf] 或 k [cr/lf]

标注	值	范围
m	端口编号	0 ~ 7、10 ~ 17、20 ~ 27、30 ~ 37
n	端口的状态（二进制）	0 或 1（合计 8 位数）。从左开始对应 m7, m6, ..., m0（m 为端口编号）
k	端口的状态（十进制）	0 ~ 255 的整数

SAMPLE 1	说明
SEND MO5 () TO CMU 从通信端口输出 MO5 端口的状态	
应答： RUN [cr/lf] DO5 () = &B00000000 [cr/lf] END [cr/lf]	

SAMPLE 2	说明
SEND CMU TO MO5 () 从通信端口输入 MO5 端口的状态 &B00000111	
应答： OK [cr/lf]	

32

LO 文件

32.1 所有 LO

读出	✓	读出所有 LO 信息。
写入	✓	通过指定 LO 端口写入。

格式
LO ()

含义 表示所有 LO。

DATA FORMAT
LO0 () = &Bnnnnnnnnn [c r / l f] LO1 () = &Bnnnnnnnnn [c r / l f] [c r / l f]

标注	值	范围
n	端口的状态 (二进制)	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	说明
SEND LO () TO CMU	……从通信端口输出所有 LO 的状态
应 答 :	
RUN [c r / l f]	
LO0 () = &B10001001 [c r / l f]	
LO1 () = &B00100100 [c r / l f]	
[c r / l f]	
END [c r / l f]	

32.2 LO 端口单位

读出	✓	读出指定的 LO 端口。
写入	✓	值将被写入至指定的 LO 端口。

格式

LOm ()

含义 表示 1 个 LO 端口的状态。

读出文件

DATA FORMAT

LOm () = &Bnnnnnnnn [cr/lf]

写入文件

DATA FORMAT

&Bnnnnnnnn [cr/lf] 或 k [cr/lf]

标注	值	范围
m	端口编号	0、1
n	端口的状态（二进制）	0 或 1（合计 8 位数）。从左开始对应 m7, m6, ..., m0（m 为端口编号）
k	端口的状态（十进制）	0 ~ 255 的整数

SAMPLE 1

说明

SEND LO0 () TO CMU 从通信端口输出 LO0 端口的状态

应答：

RUN [cr/lf]
LO0 () = &B00000000 [cr/lf]
END [cr/lf]

SAMPLE 2

说明

SEND CMU TO LO0 () 从通信端口输入 LO0 端口的状态
&B00000111

应答：

OK [cr/lf]

33

TO 文件

33.1 所有 TO

读出	✓	读出所有 TO 信息。
写入	✓	通过指定 TO 端口写入。

格式
TO ()

含义 表示所有 TO (定时器输出变量)。

DATA FORMAT
TO0 () = &Bnnnnnnnnn [cr/lf] TO1 () = &Bnnnnnnnnn [cr/lf] [cr/lf]

标注	值	范围
n	端口的状态 (二进制)	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	说明
SEND TO () TO CMU	……从通信端口输出所有 TO 的状态
SEND CMU TO TO ()	……从通信端口输入所有 TO 的状态
应 答 :	
RUN [cr/lf]	
TO0 () = &B10001001 [cr/lf]	
TO1 () = &B10001001 [cr/lf]	
[cr/lf]	
END [cr/lf]	

33.2 TO 端口单位

读出	✓	读出指定的 TO 端口。
写入	✓	值将被写入至指定的 TO 端口。

格式
TOm ()

含义 表示 1 个 TO 端口的状态。

读出文件

DATA FORMAT
TOm () = &Bnnnnnnnn [cr/lf]

写入文件

DATA FORMAT
&Bnnnnnnnn [cr/lf] 或 k [cr/lf]

标注	值	范围
m	端口编号	0、1
n	端口的状态（二进制）	0 或 1（合计 8 位数）。从左开始对应 m7, m6, ..., m0（m 为端口编号）
k	端口的状态（十进制）	0 ~ 255 的整数

SAMPLE 1	说明
SEND TO0 () TO CMU 从通信端口输出 TO0 端口的状态	
应答： RUN [cr/lf] TO0 () = &B00000000 [cr/lf] END [cr/lf]	

SAMPLE 2	说明
SEND CMU TO TO0 () 从通信端口输入 TO0 端口的状态 &B00000111	
应答： OK [cr/lf]	

34

SI 文件

34.1 所有 SI

读出	✓	读出所有 SI 信息。
写入	×	无法在写入文件中指定。

格式
SI ()

含义 表示所有 SI (串行输入变量)。

DATA FORMAT
SI 0 () = &Bnnnnnnnn [cr/lf] SI 1 () = &Bnnnnnnnn [cr/lf] : SI 27 () = &Bnnnnnnnn [cr/lf] [cr/lf]

标注	值	范围
n	端口的状态 (二进制)	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	说明
SEND SI () TO CMU	……从通信端口输出所有 SI 的状态
应答:	
RUN [cr/lf]	
SI 0 () = &B10001001 [cr/lf]	
SI 1 () = &B00000010 [cr/lf]	
SI 2 () = &B00000000 [cr/lf]	
:	
SI 7 () = &B00000000 [cr/lf]	
SI 10 () = &B00000000 [cr/lf]	
SI 11 () = &B00000000 [cr/lf]	
SI 12 () = &B00000000 [cr/lf]	
:	
SI 17 () = &B00000000 [cr/lf]	
SI 20 () = &B00000000 [cr/lf]	
:	
SI 26 () = &B00000000 [cr/lf]	
SI 27 () = &B00000000 [cr/lf]	
[cr/lf]	
END [cr/lf]	

34.2 SI 端口单位

读出	✓	读出指定的 SI 端口。
写入	×	无法在写入文件中指定。

格式

SI m ()

含义 表示 1 个 SI 端口的状态。

DATA FORMAT

SI m () = &B n n n n n n n n [c r / l f]

标注	值	范围
m	端口番号	0 ~ 7, 10 ~ 17, 20 ~ 27
n	端口的状态 (二进制)	0 或 1 (合计 8 位数)。 从左开始对应 m7, m6, ..., m0 (m 为端口编号)

SAMPLE	说明
SEND SI 5 () TO CMU 应答 : RUN [c r / l f] SI 5 () = &B 0 0 0 0 0 0 0 0 [c r / l f] END [c r / l f]	……从通信端口输出 SI 5 端口的状态

9

35.1 所有 SO

读出	✓	读出所有 SO 信息。
写入	✓ 有限制 *	值将被写入至指定的 SO 端口。 * 无法写入至 SO0() 与 SO1()。

格式
SO ()

11

含义表示所有 SO(串行输出变量)。

DATA FORMAT
SO0() = &Bnnnnnnnn [cr/lf] SO1() = &Bnnnnnnnn [cr/lf] : SO27() = &Bnnnnnnnn [cr/lf] [cr/lf]

标注	值	范围
n	端口的状态 (二进制)	0 或 1 (合计 8 位数)。 从左开始对应 m7, m6, ..., m0 (m 为端口编号)。

- 12
- 13
- 在文件的末尾仅附加表示文件结束的 [cr/lf] 行。

SAMPLE	说明
SEND SO () TO CMU	……从通信端口输出所有 SO 的状态
应答： RUN [cr/lf] 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] END [cr/lf]	

35.2 SO 端口单位

读出	✓	读出所有 SO 信息。
写入	✓ 有限制 *	值将被写入至指定的 SO 端口。 * 无法写入至 S00() 与 S01()。

格式

S0m ()

含义 表示 SO 端口单位的输出状态。

 MEMO

无法写入至 S00 () 与 S01 ()。仅限参照。

读出文件

DATA FORMAT

S0m () = &Bnnnnnnnn [cr / lf]

写入文件

DATA FORMAT

&Bnnnnnnnn [cr / lf] 或 k [cr / lf]

标注	值	范围
m	端口编号	0 ~ 7, 10 ~ 17, 20 ~ 27
n	端口的状态 (二进制)	0 或 1 (合计 8 位数)。从左开始对应 m7, m6, ..., m0 (m 为端口编号)
k	端口的状态 (十进制)	0 ~ 255 的整数

SAMPLE 1	说明
SEND S05 () TO CMU 从通信端口输出 S05 端口的状态	
应答 : RUN [cr / lf] S05 () = &B00000000 [cr / lf] END [cr / lf]	

SAMPLE 2	说明
SEND CMU TO S05 () 从通信端口输入 S05 端口的状态 &B00000111	
应答 : OK [cr / lf]	

36 SIW 文件

36.1 所有 SIW

读出	✓	以十六进制形式读出所有 SIW 信息。
写入	×	无法在写入文件中指定。

格式

SIW()

含义 表示所有 SIW(串行字输入)。

DATA FORMAT *

```
SIW(0) = &Hnnnn [cr/lf]
SIW(1) = &Hnnnn [cr/lf]
:
SIW(15) = &Hnnnn [cr/lf]
[cr/lf]
```

标注	值	范围
n	端口的状态 (十六进制)	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 *	说明
SEND SIW() TO CMU	……从通信端口输出所有 SIW 的状态
应答:	
RUN [cr/lf]	
SIW(0) = &H1001 [cr/lf]	
SIW(1) = &H0010 [cr/lf]	
SIW(2) = &H0000 [cr/lf]	
:	
SIW(15) = &H0000 [cr/lf]	
[cr/lf]	
END [cr/lf]	

- * 当 SIOW 扩大生效时, SIW(24) 至 SIW(127) 亦会被读出。
详细内容, 请参阅别册 RCX3《用户手册》“与选件板相关的参数”。

36.2 SIW 单位

读出	✓	以十六进制形式读出指定 SIW。
写入	×	无法在写入文件中指定。

格式

SIW(m)

含义 表示 1 个 SIW 的状态。

DATA FORMAT

SIW(m) = &Hnnnn [cr/lf]

标注	值	范围
m	端口番号	2 ~ 15 *
n	端口的状态 (十六进制)	0 ~ 9, A ~ F : 4 位数 (十六进制)

* 当 SIOW 扩大生效时, SIW (24) 至 SIW (127) 亦会被读出。
详细内容, 请参阅别册《用户手册》“与选件板相关的参数”。

SAMPLE	说明
SEND SIW(5) TO CMU 应答: RUN [cr/lf] SIW(5) = &H1001 [cr/lf] END [cr/lf]	……从通信端口输出 SIW (5) 的状态

37 **sow 文件**

37.1 所有 SOW

读出	✓	以十六进制形式读出所有 SOW 信息
写入	✓ 有限制 *1	值将被写入至指定的 SOW 端口。 * 1 : 无法写入至 SOW(0) 与 SOW(1)

格式

SOW()

含义 表示所有 SOW (串行字输出)。

DATA FORMAT *2

```
SOW ( 0 ) = &Hnnnn [ c r / l f ]
SOW ( 1 ) = &Hnnnn [ c r / l f ]
      :
SOW ( 15 ) = &Hnnnn [ c r / l f ]
[ c r / l f ]
```

标注	值	范围
n	端口的状态 (十六进制)	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 *2	说明
SEND SOW () TO CMU …… 从通信端口输出所有 SOW 的状态	
应答 : RUN [c r / l f] SOW (0) = &H1001 [c r / l f] SOW (1) = &H0010 [c r / l f] SOW (2) = &H0000 [c r / l f] : SOW (15) = &H0000 [c r / l f] [c r / l f] END [c r / l f]	

- * 2 : 当 SIOW 扩大生效时, SIW (24) 至 SIW (127) 亦会被读出。
详细内容, 请参阅别册《用户手册》“与选件板相关的参数”。

37.2 SOW 单位

读出	✓	以十六进制形式读出所有 SOW 信息
写入	✓ 有限制*1	值将被写入至指定的 SOW 端口。 * 1：无法写入至 SOW(0) 与 SOW(1)

格式
SOW(m)

含义 表示 1 个 SOW 的状态。

读出文件

DATA FORMAT
SOW (m) = &Hnnnn [cr/lf]

写入文件

DATA FORMAT
&Hnnnn

标注	值	范围
m	端口番号	2 ~ 15 * 2
n	端口的状态 (十六进制)	0 ~ 9, A ~ F : 4 位数 (十六进制)

* 2：当 SIOW 扩大生效时，SIW (24) 至 SIW (127) 亦会被读出。
详细内容，请参阅别册《用户手册》“与选件板相关的参数”。

SAMPLE	说明
SEND SOW (5) TO CMU 应答： RUN [cr/lf] SOW (5) = &H1001 [cr/lf] END [cr/lf]	……从通信端口输出 SOW (5) 的状态

读出	✓	读出 ^Z(= 1Ah)
写入	×	无法在写入文件中指定。

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。

DATA FORMAT
^ Z (= 1 A h)

SAMPLE	说明
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。

读出	✓
写入	✓

格式
CMU

- 含义
- 表示串行通信端口。
 - 取决于各种数据格式。

SAMPLE	说明
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

读出	✓
写入	✓

格式
ETH

- 含义
- 表示 Ethernet 端口。
 - 取决于各种数据格式。

SAMPLE	说明
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

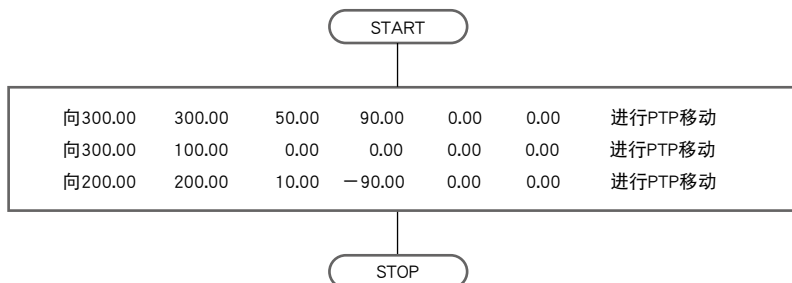
用户程序示例

1	基本篇	11-1
2	应用篇	11-8

1.1 将坐标点数据直接写入程序中

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, 300.000 300.000 50.000 90.000 0.000 0.000
MOVE P, 300.000 100.000 0.000 0.000 0.000 0.000
MOVE P, 200.000 200.000 10.000 -90.000 0.000 0.000
HALT
```

1.2 使用坐标点编号

在程序中使用坐标点编号指定坐标值。请事先在手持编程器或支持软件中输入坐标。请输入以下数据。

参照 控制器用户手册或 RCX-Studio 支持软件用户手册

POINT DATA					
P 0 =	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
P 1 =	1 0 0 . 0 0 0	0 . 0 0 0	1 5 0 . 0 0 0	3 0 . 0 0 0	0 . 0 0 0
P 2 =	0 . 0 0 0	1 0 0 . 0 0 0	5 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
P 3 =	3 0 0 . 0 0 0	3 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
P 4 =	3 0 0 . 0 0 0	1 0 0 . 0 0 0	1 0 0 . 0 0 0	9 0 . 0 0 0	0 . 0 0 0
P 5 =	2 0 0 . 0 0 0	2 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0

处理流程



35C02-R7-00

SAMPLE 1	
MOVE	P , P 0
MOVE	P , P 1
MOVE	P , P 2
MOVE	P , P 3
MOVE	P , P 4
MOVE	P , P 5
HALT	

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

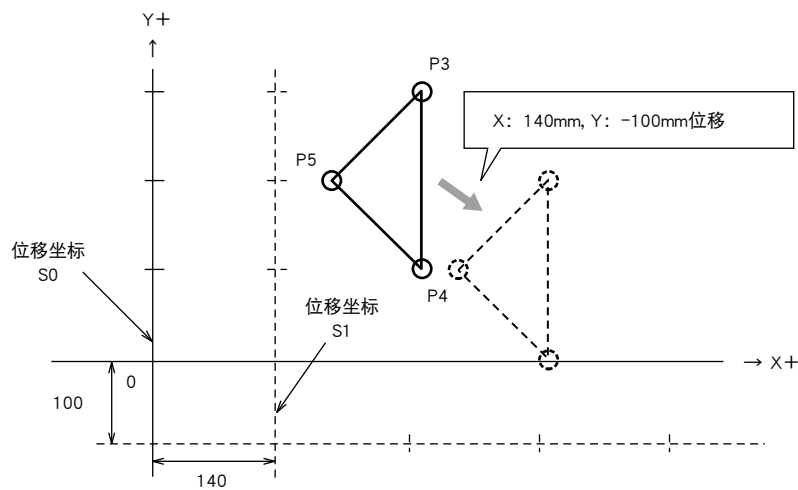
虽然 SAMPLE1 与 SAMPLE2 的动作相同，但是使用坐标点编号与 FOR 语句可使程序缩短。

1.3 使用位移坐标

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			
S 0	= 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
S 1	= 1 4 0 . 0 0 0	- 1 0 0 . 0 0 0	0 . 0 0 0

位移坐标



35C03-R7-00

SAMPLE	说明
SHIFT	………位移 0
S 0	………在 P 3 ~ P 5 之间反复移动
FOR J=3 TO 5	
MOVE P, P [J]	
NEXT J	………更改为位移 1
SHIFT S 1	………相同地在 P 3 ~ P 5 之间反复移动
FOR K=3 TO 5	
MOVE P, P [K]	
NEXT K	
HALT	

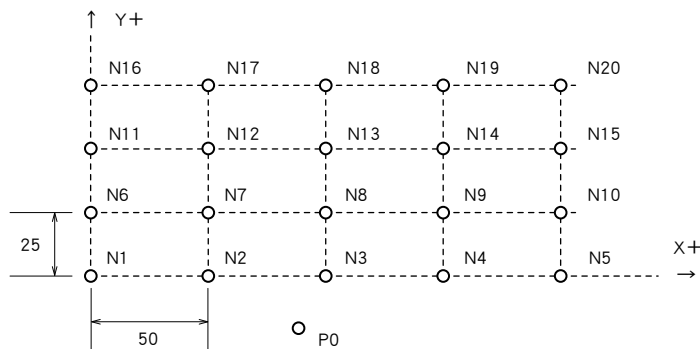
1.4 码垛

1.4.1 算出坐标点坐标

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。

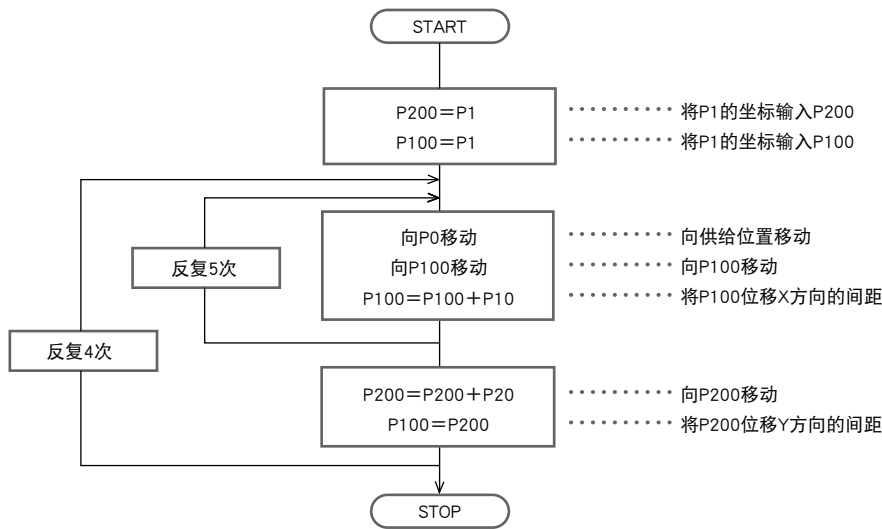
POINT DATA						
工件供给位置：						
P 0 =	0 . 0 0 0	0 . 0 0 0 0 . 0 0 0	0 . 0 0 0 0 . 0 0 0 0 . 0 0 0			
X 方向的间距：						
P 1 0 =	5 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
Y 方向的间距：						
P 2 0 =	0 . 0 0 0	2 5 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
N 1 的位：						
P 1 =	1 0 0 . 0 0 0	5 0 . 0 0 0	0 . 0 0 0	0 . 0 0 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

```
P 1 0 0 = P 1
P 2 0 0 = P 1
FOR J=1 TO 4
    FOR K=1 TO 5
        MOVE P, P 0
        MOVE P, P 1 0 0
        P 1 0 0 = P 1 0 0 + P 1 0
    NEXT K
    P 2 0 0 = P 2 0 0 + P 2 0
    P 1 0 0 = P 2 0 0
NEXT J
HALT
```

8

9

10

11

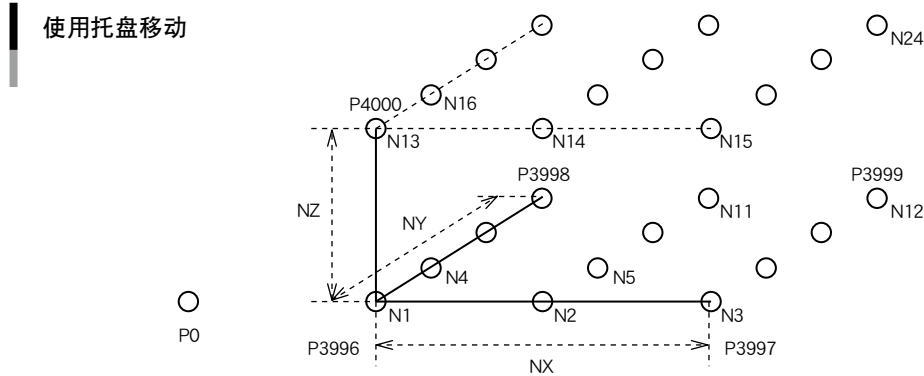
12

13

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。

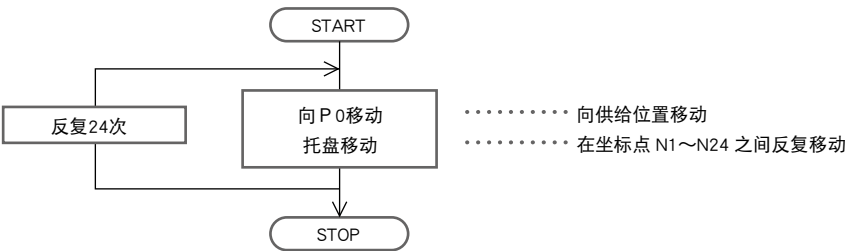
在下图上, N1 ~ N24 在正交坐标上排列着向 X 方向按照 50mm 间距的 3 个点, 向 Y 方向按照 50mm 的间距的 4 个点, 向 Z 方向按照 100mm 间距的 2 个点。机械臂从一个坐标点移动至另一个坐标点, 按照 P0 → N1 → P0 → N2 → P0 → N3…的顺序依次在 P0 与各点之间往返移动。

POINT DATA						
工件供给位置：						
P 0 =	0 . 0 0 0	0 . 0 0 0	2 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
托盘定义：						
P L 0						
N X =	3					
N Y =	4					
N Z =	2					
P L P =	3 9 9 6	(使用的坐标点为 P 3 9 9 6 ~ P 4 0 0 0)				
P [1] =	1 0 0 . 0 0 0	5 0 . 0 0 0	2 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
P [2] =	2 0 0 . 0 0 0	5 0 . 0 0 0	2 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
P [3] =	1 0 0 . 0 0 0	2 0 0 . 0 0 0	2 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
P [4] =	2 0 0 . 0 0 0	2 0 0 . 0 0 0	2 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0
P [5] =	1 0 0 . 0 0 0	5 0 . 0 0 0	1 0 0 . 0 0 0	0 . 0 0 0	0 . 0 0 0	0 . 0 0 0



35C06-R7-00

处理流程



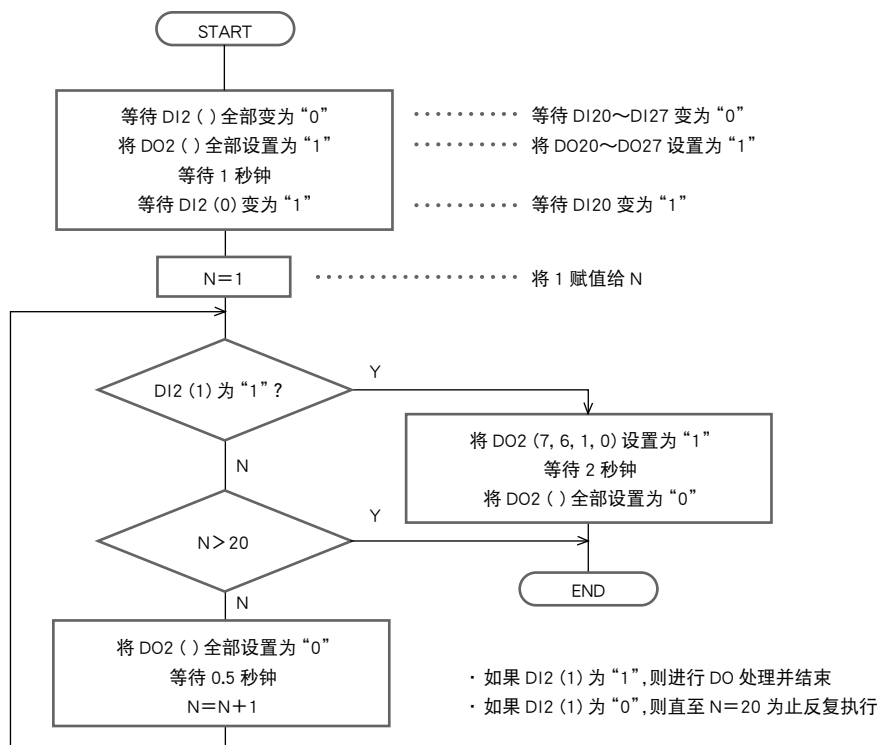
35C07-R7-00

SAMPLE	说明
FOR I = 1 TO 24	……在 I = 1 ~ 24 之间反复移动
MOVE P, P0, Z = 0.000	……向供给位置移动
PMOVE (0, I), Z = 0.000	……向托盘上的坐标点移动
NEXT I	
MOVE P, P0, Z = 0.000	
HALT	

1.5 DI / DO（数字输入输出）动作

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



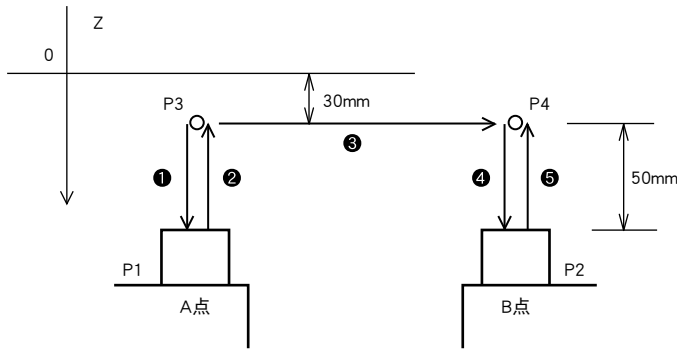
35C08-R7-00

SAMPLE	说明
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	 如果 DI21 = 1, 将跳转至 *PROGEND
*PROGEND	 如果 N > 20, 则结束 (跳转至 *ALLEND)
IF N > 20 THEN	 将 DO20 ~ DO27 全部设置为 “0”
*ALLEND	
DO2 () = 0	
DELAY 500	 反复执行循环
N=N+1	
GOTO *LOOP1	 结束时的处理
'END ROUTINE	 将 DO27, 26, 21, 20 设置为 “1”
*PROGEND:	 等待 2 秒
DO2 (7, 6, 1, 0) = &B1111	 将 DO20 设置为 “0”
DELAY 2000	
DO2 () = 0	
*ALLEND:	
HALT	

2.1 2 点间的拾放

抓住 A 点的部件并将其放置到 B 点。

2 点间的拾放



35C09-R7-00

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	说明
P 3 = P 1	… … … 将 P 1 的坐标赋值给 P 3
P 4 = P 2	… … … 将 P 2 的坐标赋值给 P 4
	… … … 将 P 3 朝着 Z 的上方向位移 5 0 m m
LOC 3 (P 3) = LOC 3 (P 3) - 5 0 . 0 0 0	… … … 将 P 4 朝着 Z 的上方向位移 5 0 m m
LOC 3 (P 4) = LOC 3 (P 4) - 5 0 . 0 0 0	
MOVE P , P 3	
GOSUB * O P E N	
MOVE P , P 1	
GOSUB * C L O S E	
MOVE P , P 3	
MOVE P , P 4	
MOVE P , P 2	
GOSUB * O P E N	
MOVE P , P 4	… … … 夹持器开启例程
HALT	
* O P E N :	
DO 2 (0) = 0	
DELAY 1 0 0	… … … 夹持器关闭例程
RETURN	
* C L O S E :	
DO 2 (0) = 1	
DELAY 1 0 0	
RETURN	

SAMPLE 使用拱形移动时	说明
P 4 = P 2	……… 将 P 2 的坐标赋值给 P 4
LOC 3 (P 4) = LOC 3 (P 4) - 5 0 . 0 0 0	……… 将 P 4 的第 3 轴数据设为位移 5 0 mm
GOSUB * O P E N	
MOVE P , P 1 , A 3 = 3 0 . 0 0 0	……… 以 A 3 = 3 0 mm 进行拱形移动
GOSUB * C L O S E	
MOVE P , P 2 , A 3 = 3 0 . 0 0 0	……… 以 A 3 = 3 0 mm 进行拱形移动
GOSUB * O P E N	
MOVE P , P 4	
HALT	
* O P E N :	……… 夹持器开启例程
DO 2 (0) = 0	
DELAY 1 0 0	
RETURN	
* C L O S E :	……… 夹持器关闭例程
DO 2 (0) = 1	
DELAY 1 0 0	
RETURN	

8

9

10

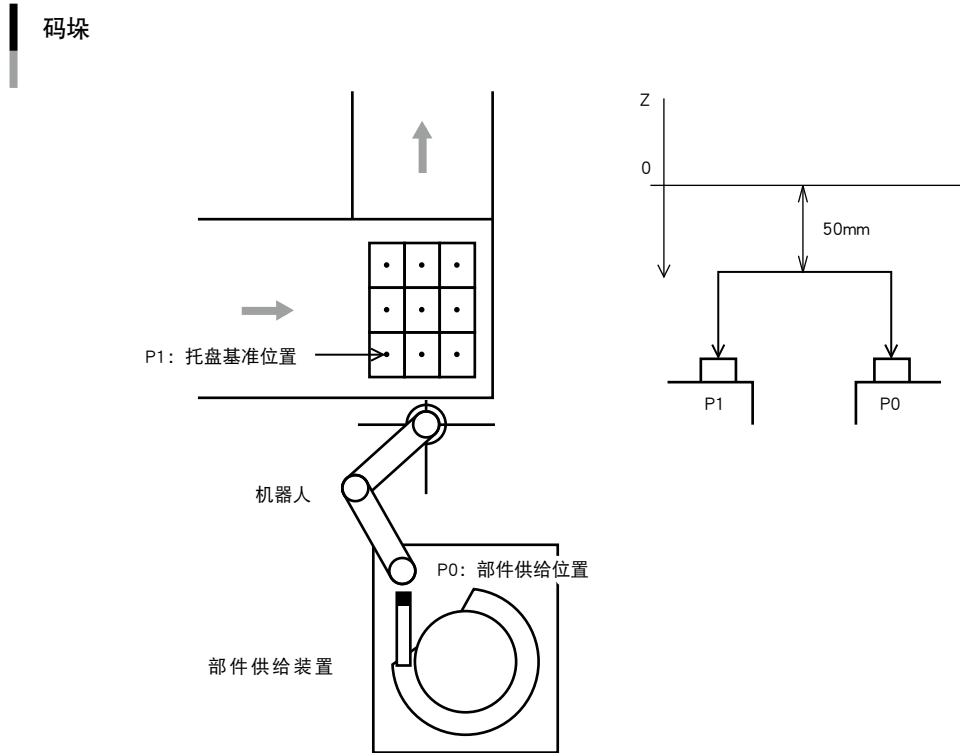
11

12

13

2.2 码垛

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。



35C10-R7-00

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。

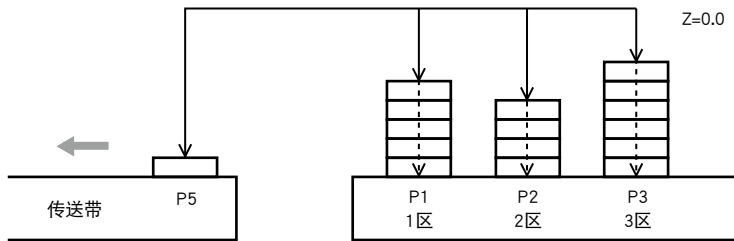
SAMPLE1 通过计算算出坐标点时	说明
WHILE -1	……反复执行整体（-1 时常为真）
FOR A=0 TO 2	
FOR B=0 TO 2	
WAIT DI(31)=1	……等待直至托盘变为“有”
WAIT DI(30)=1	……等待直至供给部件变为“有”
DO(30)=0	……将机器人机械手“开启”
DELAY 100	
MOVE P,P0,A3=50.000	……移动至供给位置
DO(30)=1	……将机器人机械手“关闭”
DELAY 100	
P100=P1+P10*B+P11*A	……计算下一个坐标点
MOVE P,P100,A3=50.000	……移动至已计算的坐标点
DO(30)=0	……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	

SAMPLE2 使用码垛功能时	说明
* 前提：必须对托盘 0 进行过定义	……反复执行整体
WHILE -1	
FOR A=1 TO 9	
WAIT DI(31)=1	……等待直至托盘变为“有”
WAIT DI(30)=1	……等待直至供给部件变为“有”
DO(30)=0	……将机器人机械手“开启”
DELAY 100	
MOVE P,P0,A3=50.000	……移动至供给位置
DO(30)=1	……将机器人机械手“关闭”
DELAY 100	
PMOVE(0,A),A3=50.000	……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	

2.3 层堆部件的拾放

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

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

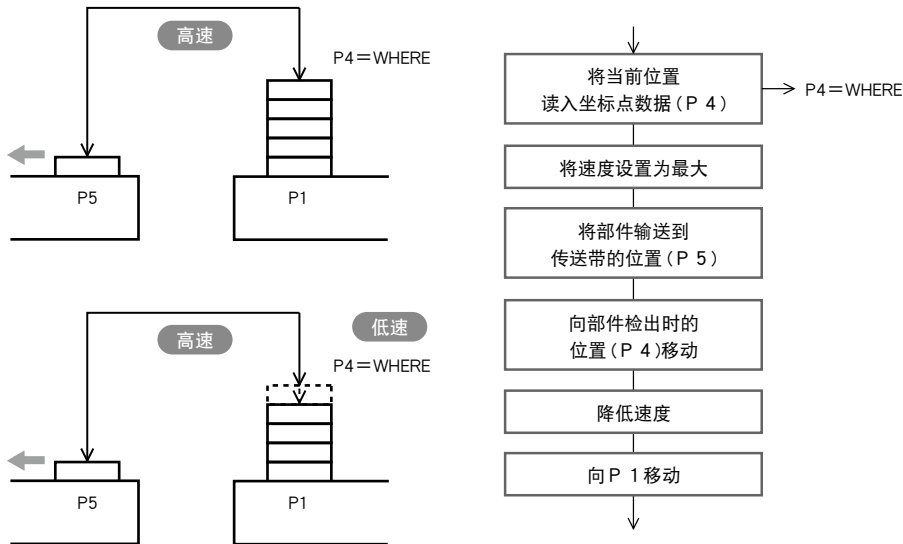
· 机器人机械手开关时，等待时间为 0.1 秒。

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。

POINT DATA

```
FOR A=1 TO 3
  SPEED 100
  GOSUB *OPEN
  P6=P[A]
  LOC3(P6)=0.000
  MOVE P,P6,A3=0.000
  WHILE -1
    SPEED 20
    MOVE P,P[A],STOPON DI3(0)=1
    IF DI3(0)=0 THEN *L1
    ' SENSOR ON
    P4=JTOXY(WHERE)
    GOSUB *CLOSE
    SPEED 100
    MOVE P,P5,A3=0.000
    GOSUB *OPEN
    MOVE P,P4,A3=0.000
  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

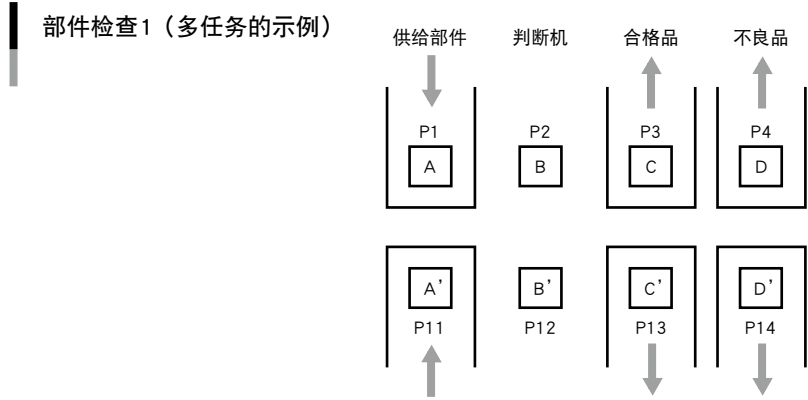
11

12

13

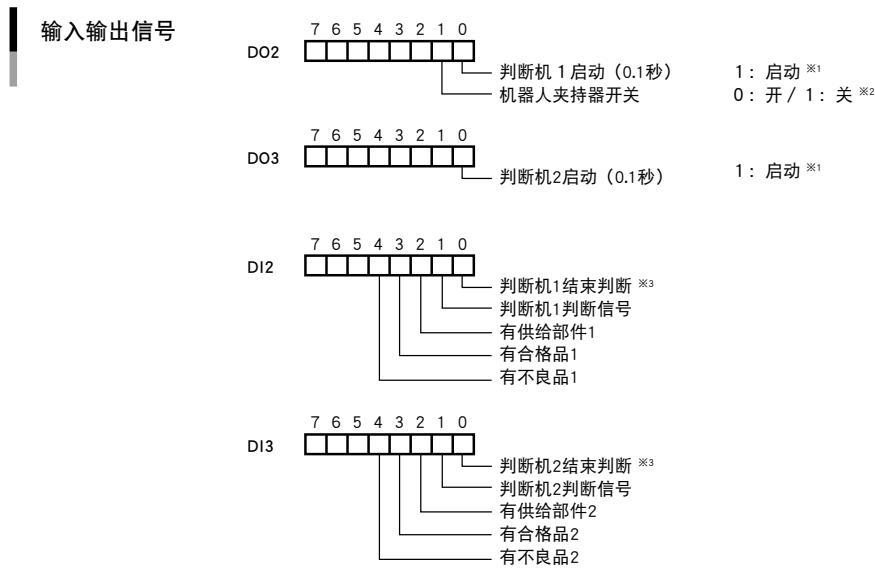
2.4 部件检查 1（多任务的示例）

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。



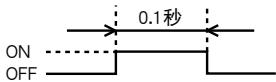
35C13-R7-00

1. 输入输出信号



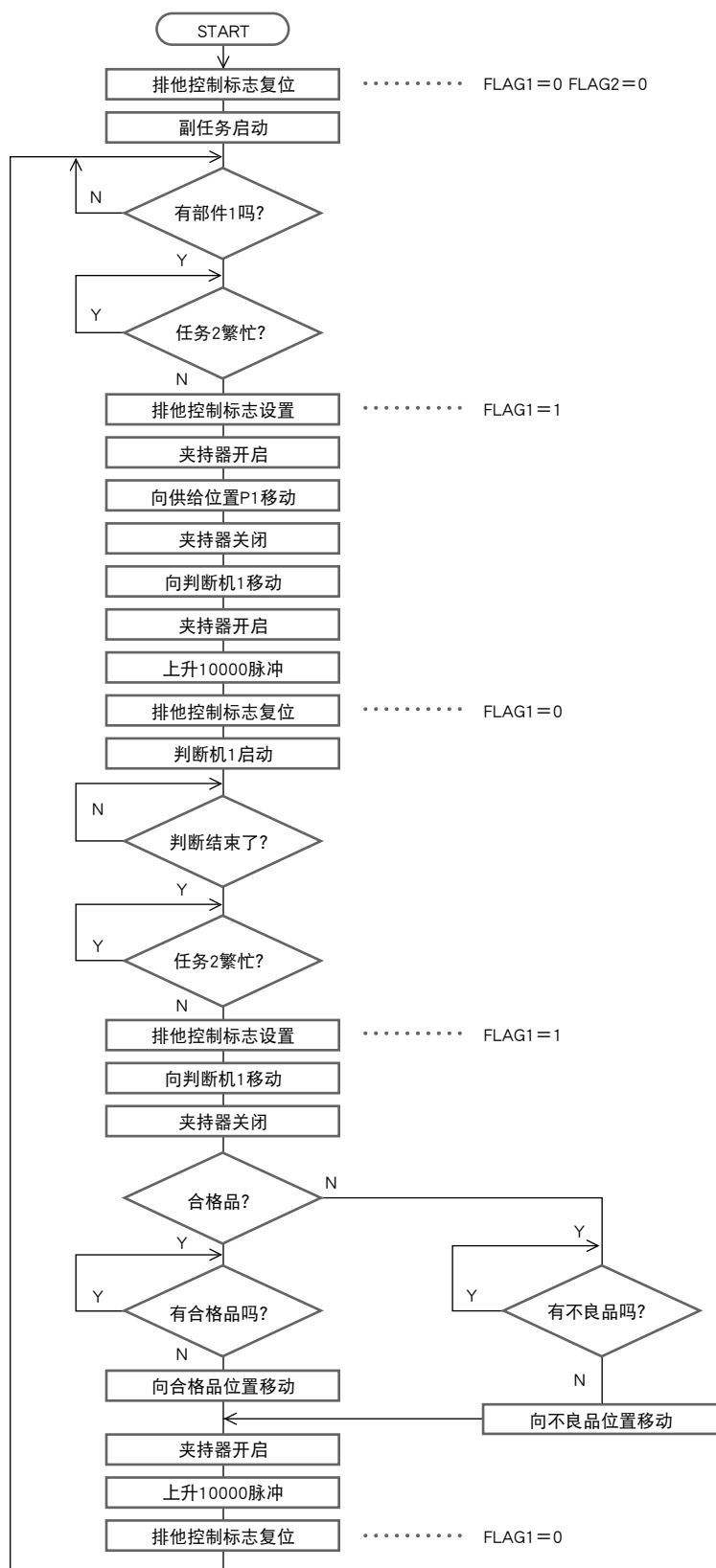
35C14-R7-00

- ※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)。



- 2. 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。
- 3. 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）



任务 2 (副任务) 也采用相同的流程表。

35C15-R7-00

SAMPLE

< 主 任 务 >

```

FLAG1=0
FLAG2=0
UPPOS=0.000
START <SUB_PGM>, T2  .... 副任务启动 == == >
*L1:
WAIT DI2(2)=1          .... 等待部件供给
WAIT FLAG2=0           .... 等待其他任务的待机状态
FLAG1=1                .... 排他控制标志设置
GOSUB *OPEN            .... 夹持器开启
MOVE P, P1, Z=UPPOS    .... 向供给位置移动
GOSUB *CLOSE           .... 夹持器关闭
MOVE P, P2, Z=UPPOS    .... 向判断机移动
GOSUB *OPEN            .... 夹持器开启
DRIVEI (3, -10000)     .... Z 上升10000脉冲
FLAG1=0                .... 排他控制标志复位
DO2(0)=1               .... 判断机启动
DELAY 100
DO2(0)=0
WAIT DI2(0)=1          .... 等待判断结束
WAIT FLAG2=0           .... 等待任务动作结束
FLAG1=1                .... 排他控制标志设置
MOVE P, P2, Z=UPPOS    .... 向判断机移动
GOSUB *CLOSE           .... 夹持器关闭
IF DI2(1)=1 THEN       .... 判断
'GOOD
WAIT DI4(2)=0          .... 等待部件移动
MOVE P, P3, Z=UPPOS    .... 向合格品位置移动
ELSE
'NG
WAIT DI2(4)=0          .... 等待部件移动
MOVE P, P4, Z=UPPOS    .... 向合格品位置移动
ENDIF
GOSUB *OPEN            .... 夹持器开启
DRIVEI (3, -10000)     .... Z 上升10000脉冲
FLAG1=0                .... 排他控制标志复位
GOTO *L1

```

< 副 任 务 >

程序名: SUB_PGM

```

*S1:
WAIT DI3(2)=1
WAIT FLAG1=0
FLAG2=1
GOSUB *OPEN
MOVE P, P11, Z=UPPOS
GOSUB *CLOSE
MOVE P, P12, Z=UPPOS
GOSUB *OPEN
DRIVEI (3, -10000)
FLAG2=0
DO3(0)=1
DELAY 100
DO3(0)=0
WAIT DI3(0)=1
WAIT FLAG1=0
FLAG2=1
MOVE P, P12, Z=UPPOS
GOSUB *CLOSE
IF DI3(1)=1 THEN
'GOOD
WAIT DI3(3)=0
MOVE P, P13, Z=UPPOS
ELSE
'NG
WAIT DI3(4)=0
MOVE P, P14, Z=UPPOS
ENDIF
GOSUB *OPEN
DRIVEI (3, -10000)
FLAG2=0
GOTO *S1

```

< 通用例程 > 程序名: COMMON

```

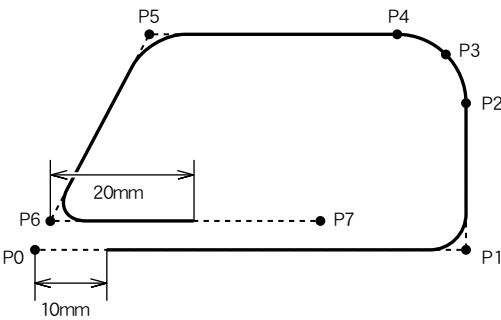
*OPEN:
DO2(1)=0
DELAY 100
RETURN
*CLOSE:
DO2(1)=1
DELAY 100
RETURN

```

2.5 涂胶

进行下图所示的零部件涂胶。

涂胶



35C16-R7-00

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	说明
MOVE P, P0, Z = 0	
SPEED 40	
PATH SET	... 开始路径设置
PATH L, P1, DO(20) = 1@10.000	... 从 10 mm 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@20.000	... 在 20 mm 的位置结束涂胶
PATH END	... 结束路径设置
PATH START	... 执行 PATH 移动
	(机器人 1 从 P0 开始动作， 在 P7 处停止)
HALT	

2.6 使用外部装置与 RS-232C 的连接（其 1）

使用外部装置与 RS-232C 连接 RCX340/RCX341/RCX320 控制器，并写入坐标点数据。

1. 从控制器输入至外部装置

SDATA/X/Y [cr/lf]



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

2. 从外部装置输出至控制器

POINT DATA						
P10=156.420	243.910	0.000	0.000	0.000	0.000	[cr/lf]

SAMPLE	说明
' INIT	
VCMD\$ = "SDATA / X / Y"	… … 移动位置要求命令
P0 = 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	… … 初始位置
' MAIN ROUTINE	
MOVE P, P0	… … 移动至初始位置
* ST :	
SEND VCMD\$ TO CMU	… … 发送命令
SEND CMU TO P10	… … 接收移动目标位置点位
MOVE P, P10	… … 移动至接收位置
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。

2.7 使用外部装置与 RS-232C 的连接（其 2）

使用外部装置与 RS-232C 连接 RCX340/RCX341/RCX320 控制器，并从任意字符串数据创建坐标点数据并写入。

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]



足 [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	说明
' INIT	
VCMD\$ = "SDATA / X / Y"	… … 移动位置要求命令
P0 = 0.0000.000 0.0000.0000.000 0.000	… … 初始位置
P11 = 100.000 100.000 0.000 0.000 0.000 0.000	… … 接收位置
' MAIN ROUTINE	
MOVE P, P0	… … 移动至初始位置
* ST :	
SEND VCMD\$ TO CMU	… … 发送命令
SEND CMU TO VIN\$	… … 接收应答
	"X=156.420,Y=243.910"
FOR I%=1 TO LEN(VIN\$)-2	
IF MID\$(VIN\$,I%,2) = "X=" THEN EXIT FOR	… … 如为 "X=" , 则跳出循环
NEXT I%	
LOC1(P11) = VAL(MID\$(VIN\$,I%+2))	… … 将 "X=" 之后的内容转换为数值后, 赋给 P11 的 1 轴
FOR I%=1 TO LEN(VIN\$)-2	
IF MID\$(VIN\$,I%,2) = "Y=" THEN EXIT FOR	… … 如为 "Y=" , 则跳出循环
NEXT I%	
LOC2(P11) = VAL(MID\$(VIN\$,I%+2))	… … 将 "Y=" 之后的内容转换为数值后, 赋给 P11 的 2 轴
MOVE P, P11	… … 移动至接收位置
GOTO * ST	

SAMPLE

```

' INT
  VCMD$="SDATA/X/Y"
  VIN$=""
  VX$=""
  VY$=""
P0=0.000 0.0000.0000.0000.0000.0000
P11=100.000100.0000.0000.0000.0000.0000
' 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$)
  WY=VAL(VY$)
  LOC1(P11)=WX
  LOC2(P11)=WY
  MOVE P, P11
GOTO *ST
HALT

```

第 12 章

联机命令

1	联机命令一览表	12-1
2	操作、设定命令	12-6
3	参照命令	12-22
4	动作命令	12-39
5	数据文件操作命令	12-42
6	实用命令	12-52
7	机器人语言的单独执行	12-54
8	控制代码	12-56

使用联机命令后，可通过 RS-232C 或 Ethernet 对控制器执行命令。

在本章节中，将介绍在联机命令中可使用的命令。关于 RS-232C 以及 Ethernet 的连接方法，请参照“RCX340 控制器手册”。

关于终端编码

控制器发送数据时，在发送数据的行尾附加下一个代码。

- RS-232C
 - 通信参数的终端编码参数设置为 "CRLF" 时，在行的最后附加 CR (0Dh) 与 LF (0Ah)。
 - 通信参数的终端编码参数设置为 "CR" 时，在行的最后附加 CR (0Dh)。
- Ethernet
 - 在行的最后附加 CR (0Dh) 与 LF (0Ah)。

接收数据时，无论终端编码参数的设置如何，至 CR (0Dh) 为止将作为 1 行，LF (0Ah) 将被忽略。

在本章“2 按钮操作”以后的各联机命令的详细说明中，终端编码表示为 [cr/lf]。

1.1 键盘操作

含义	命令	选配件	条件
将程序注册为任务	LOAD	<程序名> PGm (m: 1 - 100, n: 1 - 16, p: 1 - 64)	2
程序 复位 执行 停止	RESET RUN STOP	Tn <程序名> PGm (m: 1 - 100, n: 1 - 16)	2
程序 执行 1 行 跳过 1 行 执行到下一行	STEP SKIP NEXT	Tn <程序名> PGm (m: 1 - 100, n: 1 - 16)	2
程序 执行到指定行之前 跳至指定行之前	RUNTO SKIPTO	Tn <程序名> PGm (m: 1 - 100, n: 1 - 16, k: 1 - 9999)	2
断点设置	BREAK	<程序名> PGm 0 (m: 1 - 100, n: 1 - 9999, k: 0 / 1)	2
更改手动移动速度	MSPEED	[机器人编号] k (机器人编号: 1 - 4, k: 1 - 100)	2
移动至可执行绝对式原点复位的位置	ABSADJ	[机器人编号] k, f (机器人编号: 1 - 4, k: 1 - 6, f: 0 / 1)	3
绝对式原点复位	MRKSET	[机器人编号] k (机器人编号: 1 - 4, k: 1 - 6)	3
原点复归	ORGRTN	[机器人编号] k (机器人编号: 1 - 4, k: 1 - 6)	3
更改微动量	IDIST	[机器人编号] k (机器人编号: 1 - 4, k: 1 - 10000)	2
手动移动 (微动)	INCH INCHXY INCHT	[机器人编号] km (机器人编号: 1 - 4, k: 1 - 6, m: + / -)	3
手动移动 (寸动)	JOG JOGXY JOGT	[机器人编号] km (机器人编号: 1 - 4, k: 1 - 6, m: + / -)	3
点位示教	TEACH TCHXY	[机器人编号] m (机器人编号: 1 - 4, m: 0 - 29999)	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，还有固有限制

1.2 实用

含义	命令	选配件	条件
复制程序	COPY	<程序名 1> PGm (m: 1 - 100)	2
将 n 从点位的 m 复制到 k		Pm - Pn TO Pk (m: 0 - 29999, n: 0 - 29999, k: 0 - 29999)	
将 n 从点位注释的 m 复制到 k		PCm - PCn TO Pck (m: 0 - 29999, n: 0 - 29999, k: 0 - 29999)	
删除程序	ERA	<程序名> PGm (m: 1 - 100)	2
将 n 从点位的 m 删除		Pm - Pn (m: 0 - 29999, n: 0 - 29999)	
将 n 从点位注释的 m 删除		PCm - PCn (m: 0 - 29999, n: 0 - 29999)	
将 n 从点位名称的 m 删除		PNm - PNm (m: 0 - 29999, n: 0 - 29999)	
删除托盘的 m		PLm (m: 0 - 39)	
删除通用以太网端口的 m		GPm (m: 0 - 15)	
将 n 从实时输出设置的 m 删除		RTm - RTn (m: 24 - 127, n: 24 - 127)	
将程序名从 1 更改为 2	REN	<程序名 1> TO <程序名 2>	2
检查程序语法	SYNCHK	<程序名> PGm (m: 1 - 100, k: 1 - 100)	2
编译逻辑控制程序	SEQCML		2
更改程序属性	ATTR	<程序名>PGm TO s (m: 1 - 100, s: RW / RO / H)	2
主程序设置	MAINPG	m (m: 1 - 100)	2
数据初始化 程序 点位 点位注释 点位名称 位移 机械手 工件 托盘 通用以太网端口 输入输出名称 区域判定输出 实时输出 参数以外的全部 参数 所有数据 (MEM+PRM)	INIT	PGM PNT PCM PNM SFT HND WRKDEF PLT GEP ION ACO RTO MEM PRM ALL	3
数据初始化通信参数	INIT	CMU ETH	3
数据初始化报警履历	INIT	LOG	3
设置输入数据	INPUT	SET d CAN CLR (d: 输入数据)	2
清除输出信息缓冲区	MSGCLR		2
更改访问级	ACCESS	k , pppppppp (k: 0 / 1, p: 字母数字 8 个字符以内)	2
设置密码	SETPW		2
逻辑控制执行标志设置	SEQUENCE	k (k: 0 / 1 / 3)	2
清除警报	ALMRST		2
日期的确认、设置	DATE	yy / mm / dd (yy: 00 - 99, mm: 01 - 12, dd: 00 - 31)	2
时间的确认、设置	TIME	hh : mm : ss (hh: 00 - 23, mm: 00 - 59, ss: 00 - 59)	2

关于条件 1. 时常可执行

3. 除了条件 2. 以外，程序执行过程中也不可执行

2. 手持编程器输入操作中，无法执行

4. 除了条件 3. 以外，还有固有的限制

1.3 数据处理

含义	命令	选配件	条件
获取	?	ACCESS k , pppppppp (k : 0 / 1, pppppppp : 英数字 8 文字以内)	1
访问级		ALM	
警报状态		BREAK < 程序名 > PGm (m : 1 - 100)	
断点状态		BRKDIAG [机器人编号] (轴编号) (机器人编号 : 1 - 4) (轴编号 : 1 - 6)	
制动器故障诊断结果		CURPNT	
最终引用点位编号		EMG	
紧急停止状态		HAND [机器人编号] (机器人编号 : 1 - 4)	
选择机械手状态		HND [机械手编号] (机械手编号 : 0 - 31)	
机械手数据		IDIST [机器人编号] (机器人编号 : 1 - 4)	
微量量状态		INPUT	
输入数据		LINEMODE ETH CMU	
联机 / 脱机状态		MAINPG	
主程序编号		MEM	
存储器剩余容量		MODE	
模式状态		MOTOR	
马达电源状态		MSG	
马达电源状态		MSPEED [机器人编号] (机器人编号 : 1 - 4)	
手动移动速度		ORIGIN 机器人编号 (机器人编号 : 1 - 4)	
原点复归状态		RBTWRK [机器人编号] (机器人编号 : 1 - 4)	
工件状态		SEQCMPSTS	
逻辑控制编译状态		SEQUENCE	
逻辑控制程序执行状态		SERVO [机器人编号] (机器人编号 : 1 - 4)	
伺服状态		SHIFT [机器人编号] (机器人编号 : 1 - 4)	
选择位移状态		TASKS	
启动 / 暂停状态任务		TSKECD Tk (k : 1 - 16)	
任务结束条件		TSKMON Tk (k : 1 - 16)	
任务可动状态		VER	
版本		HND [工件编号] (工件编号 : 0 - 39)	
工件数据		数值表达式	
数值数据		字符串表达式	
字符串数据		点位表达式	
点位数据		位移表达式	
位移数据			
数据读出	READ	读出文件	2
数据写入	WRITE	写入文件	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，还有固有限制

1.4 单独执行机器人语言

可单独执行的机器人语言为第 8 章“机器人语言一览表”中“联机”栏为“√”的命令或获取函数。

参照 关于详细内容，请参阅本章“7 机器人语言的单独执行”。

含义		命令
命令	机器人语言的单独执行	<i>机器人语言</i>
获取	机器人语言的单独执行	? <i>机器人语言</i>

1.5 控制代码

含义	命令	选配件	条件
执行语言的中断	^C (=03H)		1

- 关于条件
1. 时常可执行

2. 手持编程器输入操作中，无法执行

3. 除了条件 2. 以外，程序执行过程中也不可执行

4. 除了条件 3. 以外，还有固有限制

8

9

10

11

12

13

2操作、设定命令

2.1 程序操作

2.1.1 任务登录

命令格式		
@LOAD	<程序名> PGm	,Tn, p [cr/lf]

应答格式		
OK[cr/lf]		

标注	值	范围
m	程序编号	1 ~ 100
n	任务编号	1 ~ 16
p	任务优先级	1 ~ 64

含义 按照优先顺序 p 将指定的程序登录到任务 n 中。
已登录程序变为 STOP 状态。

当省略了任务编号 n 时 : 未启动任务中最小编号的任務將自動被指定。
省略了任务优先顺序 p 时 : 默认为 32。

值越小优先顺序越高，值越大优先顺序越低(高 1 ~ 低 : 64)。
任务优先顺序高的任务变为 RUNNING 状态时，优先顺序低的任务保持 READY 状态。

SAMPLE	说明
命令 : @LOAD <PG_MAIN>, T1 [cr/lf] ……	将程序登录到任务 1
应答 : OK [cr/lf]	

2.1.2 程序复位

命令格式 1		
@RESET [cr/lf]		
命令格式 2		
@RESET	Tn <程序名> PGm	[cr/lf]

应答格式		
OK[cr/lf]		
标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100

含义 执行程序复位。

格式 1 对所有的程序进行复位。
当重新启动了程序时，从主程序或者任务 1 中最后执行的程序开头开始执行。

格式 2 仅对指定的程序进行复位。
当重新启动复位后的程序时，从该程序的开头开始执行。

SAMPLE 1	说明
命令 : @RESET [cr/lf]	……对所有程序进行复位
应答 : OK [cr/lf]	

SAMPLE 2	说明
命令 : @RESET T3 [cr/lf]……	……仅对 T3 内正在执行的程序进行复位
应答 : OK [cr/lf]	

2.1.3 程序执行

命令格式 1		
@RUN [cr/lf]		
命令格式 2		
@RUN	Tn <程序名> PGm	[cr/lf]

应答格式		
OK[cr/lf]		
标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100

含义 执行程序复位。。
格式 1 执行所有的 STOP 状态程序。
格式 2 仅执行指定的 STOP 状态程序。

SAMPLE 1	说明
命令 : @RUN [cr/lf]	……… 执行所有的 STOP 状态程序
应答 : OK [cr/lf]	

SAMPLE 2	说明
命令 : @RUN T3 [cr/lf]	……… 仅执行登录到 T3 中的 STOP 状态程序
应答 : OK [cr/lf]	

2.1.4. 程序停止

命令格式 1		
@STOP [cr/lf]		
命令格式 2		
@STOP	Tn <程序名> PGm	[cr/lf]
应答格式		
OK[cr/lf]		
标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100
含义 停止程序。 格式 1 停止所有的程序。 格式 2 仅停止指定的程序。		
SAMPLE 1		说明
命令 : @STOP [cr/lf]		……… 停止所有的程序
应答 : OK [cr/lf]		
SAMPLE 2		说明
命令 : @STOP T3 [cr/lf]		……… 仅停止 T3 内正在执行的程序
应答 : OK [cr/lf]		

2.1.5 执行 1 行程序

命令格式		
@STEP	Tn <程序名> PGm	[cr/lf]

应答格式		
OK[cr/lf]		

标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100

含义 执行 1 行指定的程序。当执行了 1 行 GOSUB 语句、CALL 语句时，进入子例程、子过程的内部。

SAMPLE	说明
命令：@STEP T3 [cr/lf]	………执行 1 行 T3 内正在执行的程序
应答：OK [cr/lf]	

2.1.6 执行程序至下一行

命令格式		
@NEXT	Tn <程序名> PGm	[cr/lf]

应答格式		
OK[cr/lf]		

标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100

含义 执行指定程序至下一行。
如果在 GOSUB 语句或 CALL 语句所在行执行 @NEXT，则在执行一次子过程处理后返回的操作，并在下一行停止。

 MEMO

在当前停止的下一行设置断点，执行与运行程序 (@RUN) 时相同的处理。
在 @STEP 中，在 GOSUB 语句或 CALL 语句调用的子过程的首行停止。

SAMPLE	说明
命令 : @NEXT T3 [cr/lf]	…… 跳过 1 行 T3 内正在执行的程序
应答 : OK [cr/lf]	

2.1.7 执行程序至指定行之前

命令格式		
@RUNTO	Tn <程序名> PGm	, k [cr/lf]

应答格式		
OK[cr/lf]		

标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100
k	指定行编号	1 ~ 9999

含义 执行指定程序至指定行之前的行。

SAMPLE	说明
命令 : @RUNTO T3, 15 [cr/lf]	…… 执行 T3 内正在执行的程序至第 15 行
应答 : OK [cr/lf]	

2.1.8 跳过 1 行程序

命令格式		
@SKIP	Tn <程序名> PGm	[cr/lf]
应答格式		
OK[cr/lf]		
标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100
含义 跳过 1 行指定的程序。当跳过了 1 行 GOSUB 语句、CALL 语句时，将跳过全部的子例程、子过程。		
SAMPLE		说明
命令：@SKIP T3 [cr/lf]		……跳过 1 行 T3 内正在执行的程
应答：OK [cr/lf]		

2.1.9 跳过程序至指定行之前

命令格式		
@SKIPTO	Tn <程序名> PGm	, k [cr/lf]
应答格式		
OK[cr/lf]		
标注	值	范围
n	任务编号	1 ~ 16
m	程序编号	1 ~ 100
k	设置断点的行号	1 ~ 9999
含义 跳过指定程序至指定行之前的行。		
SAMPLE		说明
命令：@SKIP TO T3, 15 [cr/lf]		……跳过 T3 内正在执行的程序至第 14 行
应答：OK [cr/lf]		

命令格式 1		
@BREAK	<程序名> PGm	(n,n,n,...), k [cr/lf]
命令格式 2		
@BREAK	<程序名> PGm	0 [cr/lf]
命令格式 3		
@BREAK 0 [cr/lf]		
应答格式		
OK[cr/lf]		
标注	值	范围、含义
m	程序编号	1 ~ 100
n	指定行编号	1 ~ 9999
k	设定 / 解除	0 : 解除 / 1 : 设定

含义 对程序执行过程中因暂停出现的断点进行设定。

- [格式 1] 在指定程序的指定行对断点进行设定 / 解除。也可以指定多行。
- [格式 2] 对在指定程序中设定的所有断点进行解除。
- [格式 3] 解除所有的断点。

SAMPLE	说明
命令 : @BREAK PG3 (1, 3), 1 [cr/lf] ……	在 PG3 的第 1 行和第 3 行设定断点
应答 : OK [cr/lf]	

2.1.11 程序语法检查 程序

命令格式		
@SYNCHK	<程序名> PGm	, k [cr/lf]

应答格式
RUN [cr/lf] nnnn : gg.bbb [cr/lf] nnnn : gg.bbb [cr/lf] : nnnn : gg.bbb [cr/lf] nnnn : gg.bbb [cr/lf] END [cr/lf]

标注	值	范围
m	程序编号	1 ~ 100
k	错误上限数量	1 ~ 100
nnnn	错误发生行号	1 ~ 9999
gg	警报组编号	
bbb	警报分类编号	

含义 检查程序名或程序编号中指定的程序语法。
[指定程序出现语法错误时]
输出错误行号、警报组编号及警报分类编号。

参照 关于警报组编号及警报分类编号的详细内容，请参阅控制器用户手册。

SAMPLE	说明
命令 : @SYNCHK PG1, 100 [cr/lf]	………将错误上限数量设为 100 个，进行程序编号 1 的语法检查
应答 : RUN [cr/lf] 1: 5.239 [cr/lf] 2: 5.239 [cr/lf] 3: 5.239 [cr/lf] 8: 5.239 [cr/lf] 6: 5.222 [cr/lf] END [cr/lf]	………在第 1, 2, 3, 8 行检出“5.239: 标识符错误”的语法错误的 ………在第 6 行检出“5.222: 无 ENDIF 的 IF”的语法错误

2.1.12 主程序设置

命令格式		
@MAINPG m [cr/lf]		
应答格式		
OK[cr/lf]		
标注	值	范围
m	程序编号	1 ~ 100

含义 设置复位所有程序时必须选择的程序。
主程序编号设为 0，或无主程序编号所对应的程序时，将在复位所有程序后，选择任务 1 中最后执行的程序（当前程序）。



补充 主程序相当于以往机型 (RCX240 等) 的 “_SELECT” 程序。

SAMPLE	说明
命令 : @MAINPG 1 [cr/lf]	…… 将程序编号 1 设为主程序
应答 : OK [cr/lf]	

2.1.13 编译逻辑控制程序

命令格式
@SEQCMPL [cr/lf]
应答格式
RUN [cr/lf] END [cr/lf]

含义 编译逻辑控制程序。
不存在名为 “SEQUENCE” 的程序，或输入逻辑控制程序的命令语句的格式有误，则显示错误提示。
编译正常结束时，创建执行程序，程序目录上显示 “s” 标志。

参照 详细内容请参阅 “第 7 章逻辑控制功能”。

SAMPLE	说明
命令 : @SEQCMPL [cr/lf]	…… 编译逻辑控制程序
应答 : RUN [cr/lf] END [cr/lf]	

2.2 手动模式操作

2.2.1 更改手动速度

命令格式

@MSPEED [机器人编号] k[cr/lf]

应答格式

OK[cr/lf]

标注	值	范围
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
k	手动移动速度	1 ~ 100

含义 对机器人编号指定机器人的手动移动模式的速度进行变更。

SAMPLE

命令 : @MSPEED 50 [cr/lf]

应答 : OK [cr/lf]

2.2.2 坐标点示教

命令格式

@TEACH [机器人编号] mmmmm [cr/lf]

@TCHXY [机器人编号] mmmmm [cr/lf]

应答格式

OK[cr/lf]

标注	值	范围
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
mmmm	录入坐标点数据的坐标点编号	0 ~ 29999

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。

各命令的坐标点数据的单位不同。

TEACH 脉冲单位

TCHXY mm 单位

SAMPLE

命令 : @TEACH[2] 100 [cr/lf]

应答 : OK [cr/lf]

2.2.3 更改微动距离

命令格式		
@IDIST [机器人编号] mmmmm [cr/lf]		
应答格式		
OK[cr/lf]		
标注	值	范围
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
mmmmm	微动距离	1 ~ 10000
含义 更改<机器人编号>所对应机器人的微动量。 移动量的单位因所使用的微动动作命令而异。 • INCH 脉冲单位。1 ~ 10000 脉冲 • INCHXY 以 mm 为单位。0.001 ~ 10.000mm • INCHT 以 mm 为单位。0.001 ~ 10.000mm		
SAMPLE		
命令 : @IDIST[2] 100 [cr/lf]		
应答 : OK [cr/lf]		

2.3 清除警报

命令格式	
@ALMRST [cr/lf]	
应答格式	
RUN [cr/lf] END [cr/lf]	
含义 清除警报。 但无法清除需要重启系统的警报。发生需要重启系统的警报时, 请重新接通控制器的电源。	
SAMPLE	
命令 : @ALMRST [cr/lf]	
应答 : RUN [cr/lf] END [cr/lf]	

2.4 清除输出信息缓冲区

命令格式

@MSGCLR [cr/lf]

应答格式

OK[cr/lf]

含义 清除控制器的输出信息缓冲区。删除 PRINT 语句等输出的、残留在缓冲区中的信息。

SAMPLE

命令 : @MSGCLR [cr/lf]

应答 : OK [cr/lf]

命令格式		
@INPUT	SET d CAN CLR	[cr/lf]

应答格式		
OK[cr/lf]		

标注	值	含义
d	输入数据	符合 INPUT 语句中指定变量类型的值（字符串加 “ ”）

含义

针对 INPUT 语句的数据请求设定输入数据。

控制器参数的“INPUT/PRINT 使用的频道”应设在当前通信频道 (CMU、ETH 或 iVY) 上。

针对 INPUT 语句的数据请求设定输入数据。

SET ………设定数据。

CAN………取消数据请求。

CLR ………清除设在 @INPUT SET 之后的数据。

SAMPLE	说明
< 联机命令 > @INPUT SET 10[cr/lf] @INPUT SET 5[cr/lf] OK[cr/lf] @?MSG[cr/lf] 10[cr/lf] OK[cr/lf]	< 机器人程序 > INPUT A %[cr/lf] PRINT A %[cr/lf]
< 联机命令 > @INPUT SET 10[cr/lf] OK[cr/lf] @INPUT CLR[cr/lf] OK[cr/lf] @INPUT SET 5[cr/lf] OK[cr/lf] @?MSG[cr/lf] 5[cr/lf] OK[cr/lf]	< 机器人程序 > INPUT A %[cr/lf] PRINT A %[cr/lf]

2.6 更改访问级

命令格式

```
@ACCESS k , pppppppp [cr/lf]
```

应答格式

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

标注	值	含义
k	访问级	0：维护者级别、1：操作员级别
pppppppp	密码	字母数字 8 个字符以内

含义 更改访问级。如果省略了密码，则进行无密码设置。
将访问级更改为维护者级别时，如果输入的密码不正确，将出现“6.235：密码错误”提示信息。

参照 访问级的详细内容：
控制器用户手册。

SAMPLE	说明
命令：@ACCESS 0, password [cr/lf]	密码设为“password”，更改为维护者级别
应答：OK [cr/lf]	

命令格式

@SETPW [cr/lf]

应答格式

READY [cr/lf]
pppppppp [cr/lf]
kkkkkkkk [cr/lf]
nnnnnnnn [cr/lf]
[cr/lf].....换行
OK [cr/lf]

标注	值	范围
pppppppp	旧密码（原密码）	字母数字 8 个字符以内
kkkkkkkk	新密码	字母数字 8 个字符以内
nnnnnnnn	新密码（确认）	字母数字 8 个字符以内

含义 更改将访问级更改为维护者级别时使用的密码。

- 在旧密码框中输入原密码，在新密码及新密码（确认）框中输入更改后的密码。新密码（确认）后仅进行换行。
- 如果旧密码框中输入的密码与原密码不同，或新密码与新密码（确认）不同，则出现“6.235:密码错误”的警报。

参照 访问级的详细内容：请参阅控制器用户手册。

SAMPLE	说明
命令：@SETPW[cr/lf]	
应答：READY [cr/lf] oldpass [cr/lf] newpass [cr/lf] newpass [cr/lf] [cr/lf] OK [cr/lf]	在旧密码框中输入“oldpass”在新密码框中输入“newpass”在新密码（确认）框中输入“newpass”换行

3.1 获取返回原点状态

命令格式 1		
@?ORIGIN [cr/lf]		
应答格式 1		
x [cr/lf] OK [cr/lf]		
命令格式 2		
@?ORIGIN 机器人编号 [cr/lf]		
应答格式 2		
x y{,y{,{...}}} [cr/lf] OK [cr/lf]		
标注	值	范围、含义
	机器人编号	1 ~ 4
x	机器人原点复归状态	0 : 未完成、1 : 完成
y	轴原点复归状态	从左到右显示第 1 轴、第 2 轴、…、第 6 轴的状态。 0 : 未完成、1 : 完成（轴未连接时省略）
含义 获取返回原点状态。。 [形式 1] 获取所有机器人的原点复归状态 [形式 2] 获取指定机器人的状态		
SAMPLE		说明
命令 : @?ORIGIN 2 [cr/lf]		
应答 : 0 1,1,0,1 [cr/lf] OK [cr/lf]		……… 机器人 2 的第 3 轴处于原点复归未完成状态

3.2 获取伺服状态

命令格式		
@?SERVO [机器人编号] [cr/lf]		
应答格式		
x y,y,y,y,y,y [cr/lf] OK [cr/lf]		
标注	值	范围、含义
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
x	机器人伺服状态	0 : 伺服关闭状态、1 : 伺服开启状态
y	轴伺服状态	从左向右显示第 1 轴、第 2 轴、...、第 6 轴的状态。 0 : 机械制动器开启 + 动态制动器开启状态 1 : 伺服开启状态 2 : 机械制动器关闭 + 动态制动器关闭状态 (轴未连接时省略)

含义 获取伺服状态。

SAMPLE	说明
命令 : @?SERVO[3] [cr/lf]	
应答 : 0 0,1,0,0 [cr/lf] OK [cr/lf]	 仅机器人 3 的第 2 轴处于伺服开启状态

3.3 获取马达电源状态

命令格式		
@?MOTOR [cr/lf]		
应答格式		
x [cr/lf] OK [cr/lf]		
标注	值	范围、含义
x	马达电源状态	0 : 马达电源关闭状态、 1 : 马达电源开启状态 2 : 马达电源开启 + 所有机器人伺服开启状态

含义 获取马达电源状态。

SAMPLE
命令 : @?MOTOR [cr/lf]
应答 : 2 [cr/lf] OK [cr/lf]

3.4 获取访问级状态

命令格式

@?ACCESS[cr/lf]

应答格式

k[cr/lf]
OK [cr/lf]

标注	值	范围、含义
k	访问级	0：维护者级别、1：操作员级别

含义 进行访问级状态的获取处理。

参照 关于访问级的详细说明，请参阅控制器的手册。

SAMPLE

命令：@?ACCESS[cr/lf]

应答：1[cr/lf]
OK[cr/lf]

3.5 获取断点状态

命令格式

@?BREAK	<程序名> PGm	[cr/lf]
---------	--------------	---------

应答格式

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

标注	值	范围
n	断点的设置行号	1 ~ 9999
程序名	要删除的程序名	使用字母数字及 _（下划线），32 个字符以内
m	程序编号	1 ~ 100

含义 进行断点状态的获取处理。

SAMPLE

命令：@?BREAK <TEST>[cr/lf]

应答：12,35[cr/lf]
OK [cr/lf]

3.6 获取模式状态

命令格式		
@?MODE[cr/lf]		
应答格式		
k[cr/lf] OK [cr/lf]		
标注	值	范围、含义
k	模式状态	0：手动模式 1：自动模式（控制权：手持编程器） 2：自动模式（释放控制权） -1：非法模式

含义 获取控制器的模式状态。

SAMPLE	
命令：	@?MODE[cr/lf]
应答：	1[cr/lf] OK[cr/lf]

3.7 获取通信端口状态

命令格式		
@?LINEMODE	ETH CMU	[cr/lf]
应答格式		
k[cr/lf] OK [cr/lf]		
标注	值	范围、含义
k	通信端口状态	0：脱机 / 1：联机

含义 获取指定通信端口的状态。
可以使用 ONLINE 命令更改为联机、使用 OFFLINE 命令更改为脱机。

SAMPLE	
命令：	@?LINEMODE ETH [cr/lf]
应答：	1[cr/lf] OK[cr/lf]

3.8 获取主程序编号

命令格式

@?MAINPG [cr/lf]

应答格式

m [cr/lf]

OK [cr/lf]

标注	值	范围
m	程序编号	0 ~ 100 (主程序中未录入程序时, 为 0)

含义 获取已录入主程序的程序编号。

SAMPLE

命令: @?MAINPG [cr/lf]

应答: 1[cr/lf]

OK[cr/lf]

3.9 获取次序程序执行状态

命令格式

@?SEQUENCE[cr/lf]

应答格式

1,s[cr/lf]

OK [cr/lf]

3,s[cr/lf]

OK [cr/lf]

0[cr/lf]

OK [cr/lf]

标注	值	范围、含义
s	次序程序的执行状态通过 1 或 0 表示	0: 停止中 / 1: 执行中

含义 获取次序程序的执行状态。

应答输出表示含义如下。

1: 许可状态

3: 在许可状态且紧急停止时清除输出

0: 禁止状态

SAMPLE

命令: @?SEQUENCE[cr/lf]

应答: 0[cr/lf]

OK [cr/lf]

3.10 获取逻辑控制编译状态

命令格式		
@?SEQCMPSTS[cr/lf]		
应答格式		
X[cr/lf] OK [cr/lf]		
标注	值	范围、含义
x	逻辑控制程序编译状态	0 : 未完成 / 1 : 完成

含义 获取逻辑控制程序编译状态。

SAMPLE	说明
命令 : @@?SEQCMPSTS[cr/lf]	………获取逻辑控制程序的编译状态
应答 : 1[cr/lf] OK [cr/lf]	………逻辑控制程序已编译

3.11 获取版本

命令格式	
@?VER[cr/lf]	
应答格式	
cv,cr-mv-dv1,dr1/dv2,dr2[cr/lf]	
标注	值
cv	主机的版本
cr	主机的修订版本 (Rxxxx)
mv	PLO 的版本 (Vx.xx)
dv? (? : 1、2)	驱动器的版本 (Vx.xx)
dr? (? : 1、2)	驱动器的修订版 (Rxxxx)

含义 获取版本。

SAMPLE
命令 : @@?VER[cr/lf]
应答 : V1.22,R0191-V1.000-V1.09,R0015/V1.09,R0015 [cr/lf] OK [cr/lf]

8

9

10

11

12

13

3.12 获取位移状态

命令格式		
@?SHIFT[机器人编号][cr/lf]		
应答格式		
m[cr/lf] OK [cr/lf]		
标注	值	范围、含义
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
m	指定机器人的选择位移编号	0 ~ 39 未选择位移: -1

含义 获取<机器人编号>指定机器人的位移状态。

SAMPLE	
命令:	@?SHIFT[cr/lf]
应答:	1[cr/lf] OK[cr/lf]

3.13 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] OK [cr/lf]		
标注	值	范围
n	任务编号	1 ~ 16 (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] OK [cr/lf]

3.14 获取任务运行状态

命令格式

@?TSKMON Tk[cr/lf]

应答格式

m,n,f,p[cr/lf]
OK [cr/lf]

标注	值	范围、含义
k	任务编号	1 ~ 16
m	执行程序编号	1 ~ 100
n	任务的执行行编号	1 ~ 9999
f	任务的状态	R : 执行状态 / U : 暂时休止状态 / S : 停止状态 / W : 等待状态
p	任务的优先级	1 ~ 64

含义 获取指定任务的状态。

SAMPLE

命令 : @?TSKMON T3[cr/lf]

应答 : 5,11,R,32[cr/lf]
OK [cr/lf]

3.15 获取任务结束条件

命令格式

@?TSKECD Tk[cr/lf]

应答格式

gg.bbb [cr/lf]
OK [cr/lf]

标注	值	范围
k	任务编号	1 ~ 16
gg	任务结束条件的警报组编号	
n	任务结束条件的警报分类编号	

含义 获取指定任务的结束条件。

参照 任务结束条件的警报组编号及分类编号的详细内容：请参阅控制器用户手册。

MEMO

指定任务发生错误而结束时，获取相应警报编号。

SAMPLE

说明

命令 : @?TSKECD T1 [cr/lf] 获取任务 1 的结束条件

应答 : 1.5 [cr/lf] 任务 1 的结束条件 : 1.5 : 因 HALT 而结束
OK [cr/lf]

3.16 获取机械手数据

命令格式		
@?HND[机械手编号][cr/lf]		
应答格式		
m[cr/lf] OK [cr/lf]		
标注	值	范围、含义
	机械手编号	0 ~ 31
m	机械手数据	

含义 获取<机械手编号>指定的机械手数据。

SAMPLE		说明
命令 : @?HND[0][cr/lf]		………获取机械手 0 的数据
应答 : 1,0,0.000,60.000[cr/lf] OK [cr/lf]		

3.17 获取机械手状态

命令格式		
@?HAND[机器人编号][cr/lf]		
应答格式		
m[cr/lf] OK [cr/lf]		
标注	值	范围、含义
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
m	指定机器人的选择机械手编号	0 ~ 31 未选择机械手 : - 1

含义 获取<机器人编号>指定机器人的机械手状态。

SAMPLE	
命令 : @?HAND[cr/lf]	
应答 : 1[cr/lf] OK[cr/lf]	

3.18 获取工件数据

命令格式		
@?WRKDEF[工件编号][cr/lf]		
应答格式		
m[cr/lf] OK [cr/lf]		
标注	值	范围、含义
	工件编号	0 ~ 39
m	工件数据	

含义 获取<工件编号>指定的工件数据。

SAMPLE	说明
命令 : @?WRKDEF[0][cr/lf]	……获取工件 0 的数据
应答 : 0.000 0.000 0.000 0.000[cr/lf] OK [cr/lf]	

3.19 获取工件状态

命令格式		
@?RBTWRK[机器人编号][cr/lf]		
应答格式		
m[cr/lf] OK [cr/lf]		
标注	值	范围、含义
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
m	指定机器人的选择工件编号	0 ~ 39 未选择机械手 : - 1

含义 获取<机器人编号>指定机器人的工件状态。

SAMPLE
命令 : @?HAND[cr/lf]
应答 : 1[cr/lf] OK[cr/lf]

3.20 获取存储器剩余容量

命令格式

@?MEM[cr/lf]

应答格式

k/m[cr/lf]

标注	值
----	---

k	源区域剩余容量（单位：字节）
---	----------------

m	全局识别符区域剩余容量（单位：字节）
---	--------------------

含义 获取当前存储器的剩余容量。

SAMPLE

命令：@?MEM[cr/lf]

应答：102543/1342[cr/lf]
OK[cr/lf]

3.21 获取警报状态

命令格式

@?ALM [cr/lf]

应答格式

bb.bbb [cr/lf]
OK [cr/lf]

标注	值
----	---

gg	警报组编号
----	-------

bbb	警报分类编号
-----	--------

含义 获取控制器正在发生的警报。

参照 警报组编号及警报分类编号的详细内容：请参阅控制器用户手册。

MEMO

可以获取分类编号为 400 以上的警报。存在多个警报时，取分类编号较大（程度较重）的警报。

SAMPLE

命令：@?ALM [cr/lf]

应答：12.600 [cr/lf]
OK [cr/lf]

3.22 获取紧急停止状态

命令格式		
@?EMG[cr/lf]		
应答格式		
k[cr/lf] OK [cr/lf]		
标注	值	范围、含义
k	紧急停止状态	0：正常状态、1：紧急停止状态

含义 获取紧急停止状态。确认内部紧急停止标志。

SAMPLE	
命令：	@?EMG[cr/lf]
应答：	1[cr/lf] OK[cr/lf]

3.23 获取手动移动速度

命令格式		
@?MSPEED[机器人编号] [cr/lf]		
应答格式		
k[cr/lf] OK [cr/lf]		
标注	值	范围
	机器人编号	1 ～ 4（当进行省略时，机器人 1 被指定。）
k	手动移动速度	1 ～ 100（单位：%）

含义 获取<机器人编号>所指定的机器人的手动移动速度值。

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

3.24 获取微动量

命令格式

@?IDIST[机器人编号] [cr/lf]

应答格式

mmmmm[cr/lf]

OK [cr/lf]

标注	值	范围
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
mmmmm	微动量	1 ~ 10000

含义 获取<机器人编号>所指定的机器人的微动量。

SAMPLE

命令: @?IDIST[2] [cr/lf]

应答: 100 [cr/lf]

OK [cr/lf]

3.25 获取最终引用点位编号

命令格式

@?CURPNT [cr/lf]

应答格式

k [cr/lf]

OK [cr/lf]

标注	值	范围
k	最终引用点位编号	0 ~ 29999

含义 获取最后引用的点位编号。
使用点位编辑或点位数据时, 更新最终引用点位编号。

MEMO

通过远程命令进行点位引用、指定点位移动、通过手持编程器或支持软件的跟踪移动、示教等操作, 将更新最终引用点位编号。

SAMPLE

命令: @?CURPNT [cr/lf]

应答: 100 [cr/lf]

OK [cr/lf]

3.26 获取输出信息

命令格式	
@?MSG [cr/lf]	
应答格式	
sssss...sssss [cr/lf] OK [cr/lf]	
标注	值
s	信息字符串

含义 从控制器的输出信息缓冲区获取 1 行 PRINT 等语句所输出的信息。

 MEMO

- 执行本命令之前，命令执行端口应设置“NPUT/PRINT 正在使用的频道”参数。
- 输出信息缓冲区为空时，仅 OK 作为应答输出。

SAMPLE	说明
命令：@?MSG [cr/lf]	
应答：MESSAGE [cr/lf]程序中已执行 PRINT “MESSAGE” OK [cr/lf]	

3.27 获取输入数据

命令格式	
@?INPUT [cr/lf]	
应答格式	
d [cr/lf] OK [cr/lf]	
标注	值
d	输入数据

含义 通过 INPUT 语句获取输入数据。

SAMPLE
命令：@?INPUT [cr/lf]
应答：INPUT_SAMPLE [cr/lf] OK [cr/lf]

3.28 获取制动器故障诊断结果

命令格式 1

@?BRKDIAG[机器人编号] [cr/lf]

应答格式 1

x,x,x,x,x,x [cr/lf]
OK [cr/lf]

命令格式 2

@?BRKDIAG[机器人编号] (轴编号) [cr/lf]

应答格式 2

x [cr/lf]
OK [cr/lf]

标注	值	范围、含义
	机器人编号	1 ～ 4（当进行省略时，机器人 1 被指定。）
	轴编号	1 ～ 6（当进行省略时，所有轴被指定。）
x	轴的制动器故障诊断结果	-1 未实施诊断
		0 正常
		1 马达制动器释放电压上升 1（无法释放 16V 制动器）
		2 马达制动器释放电压上升 2（无法释放 18V 制动器）
		3 马达制动器释放电压上升 3（无法释放 20V 制动器）
		4 故障（无法释放 22 V 制动器）

含义 获取制动器故障诊断结果。

[形式 1] 获取（指定机器人）所有轴的制动器故障诊断结果（未连接轴时，被省略。）

[形式 2] 获取（指定机器人）指定轴的制动器故障诊断结果

SAMPLE	说明
命令: @?BRKDIAG[2][cr/lf]	… 获取机器人 2 的第 3 轴的制动器故障诊断结果
应答: 0,0,1,0[cr/lf] OK [cr/lf]	… 机器人 2 的第 3 轴无法释放 16V 制动器。

参照 RCX340 用户手册“轴参数 制动器诊断次数< BFDNUM >”

3.29 获取值

3.29.1 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] OK [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： 2.236067E01[cr/lf] OK[cr/lf]
SAMPLE 2
命令： @?LOC1(WHERE)[cr/lf]
应答： 102054[cr/lf] OK[cr/lf]

3.29.2 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] OK [cr/lf]
含义 获取指定字符串表达式的值(字符串)。
SAMPLE
A\$="ABC"、B\$="DEF" 时
命令： @?A\$+B\$+"123"[cr/lf]
应答： ABCDEF123[cr/lf] OK[cr/lf]

3.29.3 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]

OK [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: 10.410 -1.600 52.150 3.000 0.000 0.000 0 0 0[cr/lf]
OK[cr/lf]

3.29.4 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]

OK [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: 25.000 12.600 10.000 0.000[cr/lf]
OK[cr/lf]

4 动作命令

4.1 绝对复位

命令格式

```
@ABSADJ[ 机器人编号 ] k,f[cr/lf]
@MRKSET[ 机器人编号 ] k[cr/lf]
```

应答格式

```
开始移动时：RUN[cr/lf]
结束移动时：END[cr/lf]
```

标注	值	范围、含义
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
k	轴编号	1 ~ 6
f	移动方向	0 : +方向 / 1 : -方向

含义 对利用<机器人编号>指定的机器人进行绝对复位操作。可对标记方式的轴使用。

ABSADJ 指定机器人轴向可进行绝对值复位的位置进行移动

MRKSET 指定机器人轴的绝对值复位

SAMPLE	说明
命令：@ABSADJ 1,0[cr/lf]	
应答：RUN[cr/lf]	 开始移动
END[cr/lf]	 结束移动

4.2 返回原点动作

命令格式

```
@ORGRTN[ 机器人编号 ] k[cr/lf]
```

应答格式

```
开始移动时：RUN[cr/lf]
结束移动时：END[cr/lf]
```

标注	值	范围、含义
	机器人编号	1 ~ 4 (当进行省略时, 机器人 1 被指定。)
k	轴编号	1 ~ 6

含义 对<机器人编号>指定机器人的指定轴进行原点归复。

在半绝对规格轴情况下, 当执行原点复归后, 实施绝对查找动作。

SAMPLE	说明
命令：@ORGRTN 1[cr/lf]	
应答：RUN[cr/lf]	 开始移动
END[cr/lf]	 结束移动

4.3 手动移动：点动

命令格式

@INCH [机器人编号] km [cr/lf]
@INCHXY [机器人编号] km [cr/lf]
@INCHT [机器人编号] km [cr/lf]

应答格式

开始移动时：RUN[cr/lf]
结束移动时：END[cr/lf]

标注	值	范围、含义
	机器人编号	1 ~ 4 (当进行省略时，机器人 1 被指定。)
k	轴编号	1 ~ 6
m	移动方向	0：+方向 / 1：-方向

含义

对<机器人编号>指定机器人的指定轴进行手动移动。可对标记方式的轴使用。
INCH 脉冲单位。仅指定轴移动。
INCHXY mm 单位。根据机器人的构成，机器人机械臂的前端沿正交坐标方向移动。
INCHT mm 单位。根据机器人的构成，安装在机器人机械臂前端的机械手移动。

SAMPLE	说明
命令：@INCH 1+ [cr/lf]	
应答：RUN[cr/lf]	 开始移动
END[cr/lf]	 结束移动

4.4 手动移动：微动

命令形式
@JOG[机器人编号] km[cr/lf]
@JOGXY[机器人编号] km[cr/lf]
@JOGT[机器人编号] km[cr/lf]

应答格式
开始移动时：RUN[cr/lf]
结束移动时：END[cr/lf]

标注	值	范围、含义
	机器人编号	1 ～ 4 (当进行省略时，机器人 1 被指定。)
k	轴编号	1 ～ 6
m	移动方向	0：+方向 / 1：-方向

含义 对<机器人编号>指定机器人的指定轴进行手动移动（点动）。机器人进行与在手动模式中持续按住手持编程器的微动键时一样的动作。

微动命令为了持续进行动作，联机命令的持续执行处理（^V（=16H））输入需要设为 200ms。未输入时会出现错误停止。

另外，移动开始后，当变为下列任何一种状态时，机器人停止。

- 到达软极限位置
- 联锁信号关闭
- 按下手持编程器的 STOP 键
- 通过联机命令输入了执行中断处理（^C（=03H））

各命令的移动量单位以及动作种类如下所示。

JOG：脉冲单位。仅指定轴移动。

JOGXY：mm 单位。根据机器人的构成，机器人机械臂的前端沿正交坐标方向移动。

JOGT：mm 单位。根据机器人的构成，安装在机器人机械臂前端的机械手移动。

SAMPLE	说明
命令：@JOG 1+ [cr/lf]	
应答：RUN[cr/lf]	……… 处理开始
END[cr/lf]	……… 处理完成

5.1 复制操作

5.1.1 程序的复制

命令形式		
@COPY	<程序名 1> PGn	TO <程序名 2> [cr/lf]
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
程序名 1	复制源的程序名	采用英文字母数字及“_”，在 32 个字符以内
程序名 2	复制目的地的程序名	采用英文字母数字及“_”，在 32 个字符以内
n	程序编号	1 ~ 100
含义 从<程序名 1>或者利用程序编号指定的程序复制到<程序名 2>。		
SAMPLE		说明
命令：@COPY <TEST1> TO <TEST2> [cr/lf]		
应答：RUN[cr/lf]		………处理开始
END[cr/lf]		………处理完成

5.1.2 坐标点的复制

命令形式		
@COPY Pmmmmm-Pnnnnn TO Pkkkkk [cr/lf]		
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
mmmmm	复制源的起始坐标点编号	0 ~ 29999
nnnnn	复制源的结尾坐标点编号	0 ~ 29999
kkkkk	复制目的地的起始坐标点编号	0 ~ 29999
含义 将 Pmmmmm 至 Pnnnnn 的坐标点复制至 Pkkkkk。		
SAMPLE		说明
命令：@COPY P101-P200 TO P1101[cr/lf]		
应答：RUN[cr/lf]		………处理开始
END[cr/lf]		………处理完成

5.1.3 坐标点注释的复制

命令形式		
@COPY PCmmmmm-PCnnnnn TO PCkkkkk [cr/lf]		
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
mmmmm	复制源的起始坐标点编号	0 ~ 29999
nnnnn	复制源的结尾坐标点编号	0 ~ 29999
kkkkk	复制目的地的起始坐标点编号	0 ~ 29999
含义 将 PCmmmmm 至 PCnnnnn 的坐标点注释复制至 PCkkkkk。		
SAMPLE		说明
命令：@COPY PC101-PC200 TO PC1101[cr/lf]		
应答：RUN[cr/lf]		………处理开始
END[cr/lf]		………处理完成

5.2 删除操作

5.2.1 程序删除

命令形式		
@@ERA	<程序名> PGn	[cr/lf]
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
程序名	删除的程序名	删除的程序名（采用英文字母数字及 "_", 32 个字符以内）
n	程序编号	1 ~ 100
含义 进行程序的删除操作。		
SAMPLE		说明
命令：@ERA <TEST1> [cr/lf]		
应答：RUN[cr/lf]		………处理开始
END[cr/lf]		………处理完成

5.2.2 删除坐标点

命令形式

@ERA Pmmmmm-Pnnnnn [cr/lf]

应答格式

处理开始时：RUN[cr/lf]

处理完成时：END[cr/lf]

标注	值	范围
mmmmm	删除的起始坐标点编号	0 ~ 29999
nnnnn	删除的结尾坐标点编号	0 ~ 29999

含义 进行 Pmmmmm 至 Pnnnnn 的坐标点删除操作。

SAMPLE	说明
命令：@ERA P101-P200[cr/lf]	
应答：RUN[cr/lf]	………处理开始
END[cr/lf]	………处理完成

5.2.3 删除坐标点注释

命令形式

@ERA PCmmmmm-PCnnnnn [cr/lf]

应答格式

处理开始时：RUN[cr/lf]

处理完成时：END[cr/lf]

标注	值	范围
mmmmm	删除的起始坐标点注释编号	0 ~ 29999
nnnnn	删除的结尾坐标点注释编号	0 ~ 29999

含义 进行 PCmmmmm 至 PCnnnnn 的坐标点注释删除操作。

SAMPLE	说明
命令：@ERA PC101-PC200[cr/lf]	
应答：RUN[cr/lf]	………处理开始
END[cr/lf]	………处理完成

5.2.4 删除坐标点名称

命令形式		
@ERA PNmmmmm-PNnnnnn [cr/lf]		
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
mmmmm	删除的起始坐标点名称编号	0 ~ 29999
nnnnn	删除的结尾坐标点名称编号	0 ~ 29999
含义 进行 PNmmmmm 至 PNnnnnn 的坐标点名称删除操作。		
SAMPLE		说明
命令：@ERA PN101-PN200[cr/lf]		
应答：RUN[cr/lf]		………处理开始
END[cr/lf]		………处理完成

5.2.5 删除托盘

命令形式		
@ERA PLm [cr/lf]		
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
m	删除的托盘编号	0 ~ 39
含义 进行 PLm 的托盘数据删除操作。		
SAMPLE		说明
命令：@ERA PL1[cr/lf]		
应答：RUN[cr/lf]		………处理开始
END[cr/lf]		………处理完成

5.2.6 删除通用以太网端口

命令形式		
@ERA GPm [cr/lf]		
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
m	将要删除的通用以太网端口编号	0 ~ 15
含义 进行 GPm 的通用以太网端口的删除操作。		
SAMPLE		说明
命令：@ERA GP5[cr/lf]		
应答：RUN[cr/lf]		……… 处理开始
END[cr/lf]		……… 处理完成

5.2.7 删除实时输出

命令形式		
@ERA RTmmm-RTnnn [cr/lf]		
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	范围
mmm	删除的起始实时输出编号	24 ~ 127
nnn	删除的结尾实时输出编号	24 ~ 127
含义 进行 RTmmm 至 RTnnn 的实时输出设置删除操作。		
SAMPLE		说明
命令：@ERA RT24-RT80[cr/lf]		
应答：RUN[cr/lf]		……… 处理开始
END[cr/lf]		……… 处理完成

5.3 程序名更改

命令形式		
@REN	<程序名 1> PGn	TO 程序名 2> [cr/lf]

应答格式
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]

标注	值	范围
程序名 1	更改前的程序名	采用英语字母数字及 "_", 32 个字符以内
程序名 2	更改后的程序名	采用英语字母数字及 "_", 32 个字符以内
n	程序编号	1 ~ 100

含义 进行指定程序的程序名更改。

SAMPLE	说明	
命令：@REN <TEST1> TO <TEST2> [cr/lf]		
应答：RUN[cr/lf]		处理开始
END[cr/lf]		处理完成

5.4 程序属性更改

命令形式		
@ATTR	<程序名> PGn	TO s [cr/lf]

应答格式
OK[cr/lf]

标注	值	范围、含义
程序名	更改属性的程序名	采用英文字母数字及 "_", 32 个字符以内
s	属性	RW：可读写 / RO：只读 / H：隐藏
n	程序编号	1 ~ 100

含义 进行程序名称或者利用程序编号指定程序的属性变更操作。

SAMPLE	说明
命令：@ATTR <TEST> TO RO [cr/lf]	
应答：OK[cr/lf]	

5.5 初始化处理

5.5.1 存储

命令形式

@INIT 存储器区域 [cr/lf]

应答格式

处理开始时：RUN[cr/lf]

处理完成时：END[cr/lf]

标注	值	含义
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
	WRKDEF	初始化工件区域
	PLT	初始化托盘区域
	PCM	初始化坐标点注释区域
	PNM	对坐标点名称区域进行初始化
	ION	对输入输出名称区进行初始化
	ACO	对区域判定输出设定区域进行初始化
	RTO	对实时输出区域进行初始化
	GEP	对通用以太网端口设定区域进行初始化
	MEM	对上述区域（PGM…PCM 的所有区域）进行初始化
	PRM	初始化参数区域
	ALL	初始化所有区域（MEM+PRM）

含义 进行<存储器区域>的初始化处理操作。

SAMPLE	说明
命令：@INIT PGM[cr/lf]	
应答：RUN[cr/lf]	……… 处理 开始
END[cr/lf]	……… 处理 完成

5.5.2 通信端口的初始处理

命令形式		
@INIT 通信端口 [cr/lf]		
应答格式		
处理开始时：RUN[cr/lf] 处理完成时：END[cr/lf]		
标注	值	含义
通信端口	CMU	对 RS-232C 端口进行初始化
	ETH	对以太网端口进行初始化

- 含义

进行<通信端口>的初始化处理操作。
- 参照

有关通信端口的初始值，请参阅控制器的手册。

SAMPLE	说明	
命令 :@INIT CMU[cr/lf]		
应答 :RUN[cr/lf]		处理开始
	END[cr/lf]	处理完成

5.5.3 警报记录的初始化处理

命令形式	
@INIT LOG [cr/lf]	
应答格式	
处理开始时：RUN[cr/lf]	
处理完成时：END[cr/lf]	
含义 对错误履历进行初始化。	
SAMPLE	说明
命令：@INIT LOG[cr/lf]	
应答：RUN[cr/lf]	 处理开始
END[cr/lf]	 处理完成

5.6 数据的读出处理

命令形式

@READ 读出文件 [cr/lf]

应答格式

处理开始时：BEGIN [cr/lf]

～ 数据输出：内容依据读取文件。～

处理完成时：END [cr/lf]

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

参照 有关各文件的详细说明，请参阅第 10 章“数据文件详细说明”。

分类	读出文件	记述格式	
		所有	个别
用户存储器	所有文件	ALL	
	程序	PGM	<bb...b>
	坐标点	PNT	Pn
	坐标点注释	PCM	PCn
	坐标点名称	PNM	PNn
	参数	PRM	/cccccccc/
	位移定义	SFT	Sn
	机械手定义	HND	Hn
	工件定义	WRKDEF	Wn
	托盘定义	PLT	PLn
	通用以太网端口	GEP	GPn
	输入输出名称	ION	iNMn (n)
	区域判定输出	ACO	ACn
变量・常量	实时输出	RTO	RTn
	变量	VAR	ab...by
	数组变量	ARY	ab...by(x)
状态	常量		"cc...c"
	程序目录	DIR	<<bb...b>>
	参数目录	DPM	_____
	机器参照 (传感器、撞块)	MRF	_____
	机器参考 (标记轴)	ARP	_____
	累计数据	ACD	_____
	控制器构成	CFG	_____
	控制器	CNT	_____
	机器人	RBT	_____
	驱动器	DRV	_____
	选配板	OPT	_____
	自我诊断	SCK	_____
	警报记录	LOG	_____
设备	存储器剩余容量	MEM	_____
	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 ()
其他	SOW 端口	SOW ()	SOWn ()
	文件结束代码	EOF	_____

a：英文字母

n：数字

b：英文字母数字或下划线 (_)

c：英文字母数字或特殊标注

x：表达式 (数组参数)

y：变量类型

i：输入输出类型

SAMPLE

说明

命令：@READ PGM [cr/lf]

……… 读出所有的程序

@READ P100 [cr/lf]

……… 读出点 100

@READ DINM2(0) [cr/lf]

……… 读出 DI2 的输入输出名称 (0)

命令形式

@WRITE 写入文件 [cr/lf]

应答格式

输入要求显示：READY [cr/lf]

输入结束后：OK [cr/lf]

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>

参照 有关各文件的详细说明，请参阅第 10 章“数据文件详细说明”。

 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”

分类	读出文件	记述格式	
		所有	个别
用户存储器	所有文件	ALL	———
	程序	PGM	<bb···b>
	坐标点	PNT	P _n
	坐标点注释	PCM	PC _n
	坐标点名称	PNM	PN _n
	参数	PRM	/cccccccc/
	位移定义	SFT	S _n
	机械手定义	HND	H _n
	工件定义	WRKDEF	W _n
	托盘定义	PLT	PL _n
	通用以太网端口	GEP	GP _n
	IO 名称	ION	iNM _n (n)
	区域判定输出	ACO	AC _n
变量·常量	实时输出	RTO	RT _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 (_)

c：英文字母数字或特殊标注

x：表达式 (数组参数)

y：变量类型

i：输入输出类型

SAMPLE

说明

命令：@WRITE PGM [cr/lf]

·····写入所有程序

@WRITE P100 [cr/lf]

·····写入点 100

@WRITE DINM2(0) [cr/lf]

·····写入 DI2 的输入输出名称 (0)

6.1 设置次序程序执行标志

命令形式		
SEQUENCE k[cr/lf]		
应答格式		
OK [cr/lf]		
标注	值	范围、含义
k	执行标志	0：禁止、1：允许、3：允许（DO 复位）
含义 进行次序程序执行标志的设置处理。		
SAMPLE		
命令：@SEQUENCE 1[cr/lf]		
应答：OK[cr/lf]		

6.2 日期设定

含义 设定控制器的日期。					
命令形式					
@DATE yy/mm/dd[cr/lf]					
应答格式					
OK [cr/lf]					
标注	值	范围	标注	值	范围
yy/mm/dd	设定日期（年月日）		mm	月	01 ~ 12
yy	公历的后 2 位数	00 ~ 99	dd	日	01 ~ 31

含义 设定控制器的日期。



补充 仅更改公历或月时，可省略后面的"/"（斜线）。
示例）设置为 2016 年16[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 [5.202：数据错误] 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 16/01/14[cr/lf]	
应答：OK[cr/lf]	

命令形式

@TIME hh:mm:ss[cr/lf]

应答格式

OK [cr/lf]

标注	值	范围	标注	值	范围
hh:mm:ss	当前时		mm	分	00 ~ 59
hh	时	00 ~ 23	ss	秒	00 ~ 59

含义 设定控制器的时间。

 MEMO

- 省略的项目使用当前值。
- 仅发送 [cr/lf] 时，将不更新日期。
- 输入不可能的时间时，将发生 [5.202：数据错误] 的错误。

SAMPLE 1 仅更改分时

:20:[cr/lf] ……将分设置为 20 分

SAMPLE 2

命令：@TIME 10:21:35[cr/lf]

应答：OK[cr/lf]

7.1 命令

命令形式		
@ 机器人语言 [cr/lf]		
应答格式 1		
OK[cr/lf] 或 NG=gg.bbb [cr/lf]		
应答格式 2		
处理开始时：RUN [cr/lf] 或者 NG=gg.bbb [cr/lf] 处理完成时：END [cr/lf] 或者 NG=gg.bbb [cr/lf]		
标注	值	范围
OK END	命令正常结束	
NG	发生错误	
RUN	命令正常开始	
gg	警报组编号	0 ~ 99
bbb	警报分类编号	0 ~ 999

- 含义
- 进行<机器人语言>命令处理。

· 仅执行可单独执行的命令。

· 命令格式取决于各命令。

SAMPLE 1	
命令：@SET DO (20) [cr/lf]	
应答：OK[cr/lf]	
SAMPLE 2	
说明	
命令：@MOVE P,P100,S=20[cr/lf]	
应答：RUN[cr/lf]	……… 处理开始
END[cr/lf]	……… 处理完成

函数形式
@? 机器人语言 [cr/lf]

应答格式
X[cr/lf] OK[cr/lf]

标注	值	范围
X	返回值	
OK END	命令正常结束	
NG	发生错误	
gg	警报组编号	0 ~ 99
bbb	警报分类编号	0 ~ 999

- 含义
- 进行<机器人语言>函数处理。
 - 仅执行可单独执行的函数。
 - 函数格式取决于各函数。

SAMPLE	说明
命令 : @?REFTSK(1)[cr/lf] …… 获取任务 1 的状态	
应答 : 0[cr/lf] OK[cr/lf]	…… 任务 1 是 STOP (停止) 状态

8 控制代码

命令形式

^C (=03H)

应答格式

NG=1.8

含义 中断执行中的命令。

SAMPLE 1

命令: @MOVE P, P100, S=20[cr/lf]
^C

应答: NG=1.8[cr/lf]

第 13 章

附录

1	保留字一览表	13-1
2	变化点（与以往机型相比）	13-3

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	CUT	HOLD	MRF
ABS	D	HOLDALL	MRKSET
ABSADJ	DATE	I	MSG
ABSRPOS	DBP	IDIST	MSGCLR
ACCEL	DECEL	IF	MSPEED
ACCESS	DEF	IMP	MTRDUTY
ACO	DEGRAD	INCH	N
ACD	DELAY	INCHT	NAME
ALL	DI	INCHXY	NEXT
ALM	DIM	INIT	NOT
ALMRST	DIR	INPUT	O
AND	DIST	INROFST	OFFLINE
ARCHP1	DO	INT	ON
ARCHP2	DPM	ION	ONLINE
ARM	DRIVE	J	OPEN
ARMCND	DRIVEI	JOG	OPT
ARMSEL	DRV	JOGT	OR
ARMTYP	E	JOGXY	ORD
ARP	ELSE	JTOXY	ORGORD
ARY	ELSEIF	L	ORGRTN
ASPEED	EMG	LEFT	ORIGIN
ATN	END	LEFTY	OUT
ATN2	ENDIF	LEN	OUTPOS
ATTR	EOF	LET	P
AXWGHT	EQV	LINEMODE	P
B	ERA	LO	PATH
BIN	ERL	LOAD	PCM
BREAK	ERR	LOC1	PDEF
C	ERROR	LOC2	PGM
CALL	ETH	LOC3	PGMTSK
CASE	ETHSTS	LOC4	PGN
CFG	EXIT	LOC5	PLT
CHANGE	EXITASK	LOC6	PMOVE
CHGPRI	F	LOC6	PNM
CHGWKR	FN	LOG	PNT
CHR	FOR	LSHIFT	PPNT
CLOSE	G	M	PRINT
CMU	GEP	MAINPG	PRM
CNT	GEPSTS	MCHREF	PSHFRC
CONTPLS	GO	MEM	PSHJGSP
CORRECTXY	GOSUB	MID	PSHMTD
CONTPLS	GOTO	MO	PSHRSLT
COPY	H	MOD	PSHSPD
COS	HALT	MODE	PSHTIME
CREWRK	HALTALL	MOTOR	PUSH
CURPNT	HAND	MOVE	
CURTQST	HEX	MOVEI	
CURTRQ	HND	MOVET	

R	SEQUENCE	SUSPEND	VMCSTS
RADDEG	SERVO	SWI	W
RBT	SET	SYNCHK	WAIT
RBTWRK	SETGEP	T	WEIGHT
READ	SETPW	TAG	WEIGHTG
REF	SFT	TAN	WEND
REM	SGI	TASKS	WHERE
REN	SGR	TCHXY	WHILE
RESET	SHARED	TCOUNTER	WHRXY
RESTART	SHIFT	TEACH	WHRXYEX
RESUME	SI	THEN	WRITE
RETURN	SID	TIME	WRKDEF
RIGHT	SIN	TIMER	X
RIGHTY	SIW	TO	XOR
RSHIFT	SKIP	TOLE	XYTOJ
RTO	SKIPTO	TORQUE	
RUN	SO	TSKECD	
RUNTO	SOD	TSKMON	
S	SOW	TSKPGM	
S	SPEED	V	
SCK	SQR	VAL	
SCRINR	START	VAR	
SELECT	STEP	VER	
SEND	STOP	VMCDEF	
SEQCMPL	STR	VMCPNO	
SEQCMPSTS	SUB	VMCORRECT	

以下语言因被作为系统变量名，所以无法使用以这些名称起始的变量名（n：数值）。

ACn	GPn	PNn	SOn
Dln	Hn	SGln	SONMn
DINMn	LOn	SGRn	TOn
DOOn	MOOn	Sln	
DONMn	PCn	SINMn	
FN	Pn	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。

1 程序名称

具有以下两个程序名称的程序在以往机型（RCX240 等）中具有特殊含义，在 RCX340/RCX341/RCX320 中没有特殊含义。

- A) FUNCTION
- B) _SELECT

A) FUNCTION

在以往机型（RCX240 等）中，它被用作注册用户功能的特殊程序，但由于它不具有用户功能的功能，因此没有特殊含义。

B) _SELECT

在以往机型（RCX240 等）中，当机器人程序复位时，总是选择并执行程序名称为“_SELECT”的程序。
但是，当在 RCX340/RCX341/RCX320 中重置机器人程序时，程序设置为主程序号 如果程序编号中未设置程序编号，则将选择并执行最后执行的程序。

参照 关于主程序，请参阅第 12 章“2.1.12 主程序设置”

2 多台机器人控制

在以往机型（RCX240 等）中，主组（主机器人，主附加轴）和子组（子机器人，附加轴）构成机器人。
但是，在 RCX340/RCX341/RCX320 中，机器人由机器人 1~ 机器人 4（法线轴，附加轴）组成。
因此，以往机型（RCX240 等）中使用的特定于组的命令已更改为具有机器人指定的命令。

参照 关于 RCX340，请参阅第 5 章“2. 机器人指定命令一览表”
关于以往机型（RCX240 等），请参阅编程手册“各组命令一览表”

以往机型（RCX240 等）中“各组命令一览表”

SAMPLE	说明
MOVE P, P1	将主组的轴移动到 P1 指定的位置
MOVE2 P, P5	将子组的轴移动到 P5 指定的位置

在 RCX340/RCX341/RCX320 中“机器人指定命令一览表”

SAMPLE	说明
MOVE P, P1	将机器人 1 的轴移动到 P1 指定的位置
MOVE[2] P, P5	将机器人 2 的轴移动到 P5 指定的位置

 MEMO

机器人指定的指令可以省略机器人编号。如果省略机器人编号，将选择机器人 1。

3 多任务

在以往机型 (RCX240 等) 和 RCX340/RCX341/RCX320 中, 存在以下变化。

	以往机型	RCX340/RCX341/RCX320
最大任务数	8	16
优先级	17 ~ 47	1 ~ 63
任务的定义	在程序中定义任务	在不同的程序中定义任务
启动任务	它会自动分配给任务 1, 变为 RUN 状态。	将任务分配给指定的任务编号, 变为 RUN 状态。
对任务 1 执行 RESTART 指令等	不可以	可以

参照 “多任务”章 (这本书第 6 章、以往机型的编程手册)

4 机器人语言

RCX340/RCX341/RCX320 不支持以下机器人语言

ABSINIT	ABSINIT2	ABSRST	ABSRPOS2
ACCEL2	ARMCND2	ARMTYP2	ASPEED2
AXWGHT2	CHANGE2	CURTRQ2	DECEL2
DECLARE	DRIVE2	DRIVEI2	HAND2
JTOXY2	LEFTY2	MCHREF2	MOVE2
MOVEI2	ORGORD2	OUTPOS2	PMOVE2
RIGHTY2	SERVO2	SHIFT2	SPEED2
TOLE2	TORQUE2	TRQSTS	TRQSTS2
TRQTIME	TRQTIME2	WAIT ARM2	WEIGHT2
WHERE2	WHRXY2	XYTOJ2	_SYSFLG

5 联机命令

RCX340/RCX341/RCX320 不支持以下联机命令

AUTO	EMGRST	EXELV	MANUAL
? ARM	? CONFIG	? EXELVL	? OPSLOT
? SELFCHK	? WHRXYEX		

6 数据文件

- 对应于以往机型的“错误提示履历文件”和“错误提示履历详细文件”消失, 并添加新的“报警履历文件”。
- 在以往机型中, 坐标点编号的范围是 0 到 9999, 但在 RCX340/RCX341/RCX320 中, 它是 0 到 29999。

参照 第 10 章“数据文件详细说明”

修订记录

修订日期	修订内容
2014年 2月	1.00 版 初版
2014年 8月	1.10 版 误记修正等
2017年 2月	1.11 版 联系方式变更
2018年 9月	1.36 版 追加新建第 7 章「逻辑控制功能」、第 9 章「PATH 语句」，误记修正等
2019年 3月	1.38 版 追加工件定义命令，误记修正等
2019年 8月	1.39 版 追加 WEIGHTG 命令
2019年10月	2.00 版 追加控制器 RCX320
2020年 4月	2.03 版 追加 CONTPLS/CORRECTXY/SCRINR/INROFST 命令、CONT 指定、实时输出功能、SIOW 扩大功能
2021年 3月	2.06 版 追加 REFTSK 函数、HND/WRKDEF 联机命令
2021年10月	2.07 版 第 8 章 LOCx 的使用对象中追加工件数据
2022年 4月	2.09 版 追加数据文件 / 联机命令 ACD、追加联机命令 SEQCMPSTS、 在第 8 章样本程序中追加说明文字
2022年10月	2.10 版 追加累计数据文件的说明文、追加 SGI 和 SGR 的格式在第 8 章、 在第 8 章 PDEF 中追加坐标点指定省略时的说明文
2023年 2月	2.11 版 误记修正
2024年 2月	2.12 版 在第 8 章 DRIVE 中追加限制事项、地址变更
2024年 4月	2.13 版 误记修正
2024年 8月	2.14 版 追加制动器故障诊断功能 BRKDIAG（机器人语言、联机命令）
2024年11月	2.16 版 追加 RCX341、命令 / 函数 / 联机命令区分在第 8 章、八进制常量在第 8 章 「2.1 整数型常量」、「端口 / 比特编号的变量指定」在第 3 章「10 指定端口 / 比特」

编程手册

4轴/2轴机器人控制器

RCX 3 Series

2024年 11月
Version 2.16 版

YAMAHA MOTOR CO., LTD. Robotics Business Unit

- ◆ 禁止复制或转印本书的全部或部分内容。
- ◆ 本书内容的变更恕不另行告知。
- ◆ 虽然我们对本书内容力求完善无误，但难免有错误、不详处、遗漏处，如果您发现有任何错误的地方，敬请与本公司联系。

联系我们

雅马哈发动机智能机器(苏州)有限公司

地址:江苏省苏州工业园区苏虹东路17号8号厂房

邮编:215026

电话:(0512) 6831 7091 / 6831 7092

传真:(0512) 6831 7093

E-mail:robot@yamaha-motor.com.cn

雅马哈发动机智能机器(苏州)有限公司深圳分公司

地址: 深圳市龙华区观澜街道观光路1301-70号银星智界一期1号楼1楼

邮编:518110

电话:(0755) 2393 9910

传真:(0755) 2393 9974

雅马哈发动机株式会社 Robotics Business Unit

静冈县滨松市中央区丰冈町127 邮编 433-8103

<https://www.yamaha-motor.com.cn/robot/>

