ICS 25.040.20
J 50

中华人民共和国国家标准

GB/T 39134—2020

机床工业机器人数控系统 编程语言

Industrial robot numerical control system of machine tool—Programming language

2020-10-11 发布　　2021-05-01 实施

国家市场监督管理总局
国家标准化管理委员会　发布

目　　次

前　言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准由中国机械工业联合会提出。

本标准由全国机床数控系统标准化技术委员会(SAC/TC 367)归口。

本标准起草单位:佛山智能装备技术研究院、佛山华数机器人有限公司、重庆大学、华中科技大学、武汉华中数控股份有限公司、东莞理工学院。

本标准主要起草人:尹玲、周星、陈思敏、黄键、高萌、宁国松、杨林、欧道江、杨海滨、李国龙、张航军、金健、陈吉红。

引　言

当前工业机器人应用进入爆发式增长，工业机器人在各类数控智能加工单元中与数控机床配套，已成为智能制造车间的核心组成部分，用数控系统实现对数控智能加工单元的控制成为当前行业趋势，这些不断出现的新应用形式对数控系统控制工业机器人的编程语言提出了新的要求。

本标准完善了机床工业机器人数控系统编程代码体系，扩充了代码内容和涵义，有助于统一机床工业机器人数控系统编程代码使用要求，引导工业机器人数控系统编程语言向功能性强、兼容性好、通用性高的方向发展，使工业机器人编程操作更安全、简洁、高效，提升操作体验。本标准的指定对于促进本领域的技术交流和技术进步，加快工业机器人的应用推广具有重要意义。

机床工业机器人数控系统　编程语言

1　范围

本标准规定了机床工业机器人数控系统的编程语言，以及编程语言中的指令类型、功能和用法。

本标准适用于机床工业机器人数控系统。其他用途的机器人控制系统可参照本标准。

2　术语和定义

下列术语和定义适用于本文件。

2.1

运动指令　move instruction

对工业机器人各关节转动、移动运动控制的相关指令。

[GB/T 29824—2013，定义 2.1]

2.2

运动附加指令　additional move instruction

在工业机器人的运动指令中附加的、特定的参数设置或任务指令，实现工业机器人运动过程中的特定任务。

2.3

力控制指令　force control instruction

对工业机器人在不同工作状态、不同工作对象时的负载或力进行设置和控制的相关指令。

2.4

速度控制指令　speed control instruction

对工业机器人关节或运动轴的运动速度、加速度、加加速度进行设置的相关指令。

2.5

协作控制指令　collaborative instruction

工业机器人与其他设备协同作业时，对其与周边设备的同步和时序作业进行控制的相关指令。

2.6

坐标系设置指令　coordinate instruction

对工业机器人坐标系设置及操作的相关指令。

2.7

寄存器操作指令　register operation instruction

对工业机器人数控系统编程时涉及的相关寄存器配置及操作的指令。

2.8

数据处理指令　data processing instruction

对程序数据进行设定、清除等操作的相关指令。

改写[GB/T 29824—2013，定义 2.2]

2.9

流程控制指令　flow control instruction

对工业机器人操作程序的执行顺序产生影响的指令。

2.10

位置补偿指令　position compensation instruction

对工业机器人的位置点进行偏移补偿的指令。

2.11

运算指令　arithmetic instruction

对程序中相关数据进行算数运算或逻辑运算的指令。

2.12

工具中心点　tool center point;TCP

实际使用工业机器人一般安装夹具等辅助装置,为了编程方便,以辅助装置中心为原点建立工具坐标系统。

3　编程语言与指令类型

工业机器人编程语言由指令、寄存器、常量组成。指令包括运动指令、力控制指令、速度控制指令、坐标系设置指令、寄存器操作指令、数据处理指令、流程控制指令、位置补偿指令、运算指令、其他指令;寄存器包括位姿寄存器、数值数据寄存器、输入输出寄存器;常量包括位姿常量、数值常量、字符串常量。具体如图1所示。

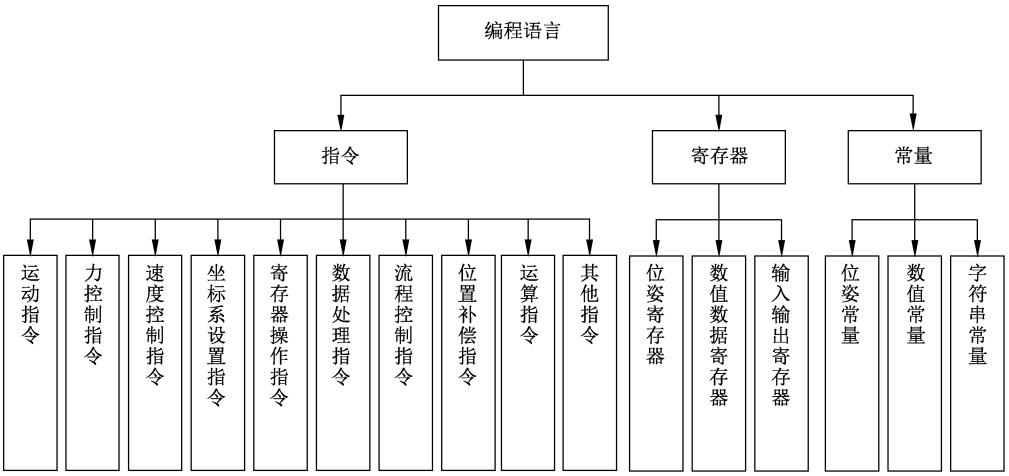

图1　工业机器人程序指令组成

典型编程程序格式框架参见附录A。

4　指令功能与用法

4.1　运动指令

4.1.1　概述

运动指令指对工业机器人各关节转动或TCP移动进行运动控制的点到点、直线或圆弧指令。

4.1.2　J指令

指令功能:以关节轴插补方式进行的点到点运动。

编程格式:J ＜P＞ Vel=＜Value＞ {Optional Property} *

其中：

P ——目标点点位信息，可以是 P[i]（示教默认保存点位名），JR[i]，LR[i]或常量的任意一种；

Vel ——关节运动速度百分比，取值范围[1，100]，计数单位为 1%，表示以关节最大速度的百分之 Value 运动；

{Optional Property} * ——可选择项，如 Acc，Dec，Cnt，Offset，Inc，Skip 等，参见附录 B。

示例：

J P[1] Vel=100 Acc=100 Dec=100 Cnt=10；

以关节插补的方式向目标位置 P[1]移动，并速度为设定关节速度的 100%，加、减速因子设定为 100%的关节运动最大加减速，平滑过渡系数设定为当前点与目标点之间距离长度的 10%。

4.1.3 L 指令

指令功能：以笛卡尔坐标插补方式进行的直线运动。

编程格式：L ＜P＞ Vel=＜Value＞ {Optional Property} *

其中：

P——目标点点位信息，可以是 P[i]，JR[i]，LR[i]或常量任意一种；

Vel——空间运动速度，单位为毫米每秒（mm/s）；

{Optional Property} * ——可选择项，如 Vrot，Acc，Dec，Cnt，Offset，Inc，Skip，Wjnt 等，参见附录 B。

示例：

L P[1] Vel=100 Acc=100 Dec=100 Cnt=10；

以直线的方式向目标位置 P[1]移动，并设定速度为 100 mm/s，加、减速因子设定为 100%的直线运动最大加减速，平滑过渡系数设定为当前点与目标点之间距离长度的 10%。

4.1.4 C 指令

指令功能：以笛卡尔坐标插补方式执行的圆弧运动，经过中间点，最终到目标点。

编程格式：C ＜P1＞ ＜P2＞ {P3} Vel=＜Value＞ {Optional Property} *

其中：

P1/P2/P3——目标点点位信息，可以是 P[i]，JR[i]，LR[i]或常量任意一种；

Vel——空间运动速度，单位为毫米每秒（mm/s）；

{Optional Property} * ——可选择项，如 Vrot，Acc，Dec，Cnt，Offset，Inc，Skip，Wjnt 等，参见附录 B。

注： 如仅有两个点，则执行圆弧运动，其中 P1 为中间点点位信息，P2 为末端点点位信息。如存在三个点，则进行整圆运动，从 P1，经过 P2，P3，最终停止在 P1，P1 点需设置为当前位姿。

示例：

C P[1] P[2] Vel=100 Acc=100 Dec=100 Cnt=10；

以圆弧的方式从当前点，经过中间位置 P[1]点，往目标位置 P[2]点移动，并设定速度为 100 mm/s，加、减速因子设定为 100%的直线运动最大加减速，平滑过渡系数设定为当前点与目标点之间距离长度的 10%。

4.1.5 JDO 指令

指令功能：在运动不必是直线时，快速将工业机器人从一个点移动到另一个点，在目标点位置或平滑路径中间位置，设置（置位/复位）输出信号。该指令下，工业机器人和外部轴沿着非线性路径移动到目标位置，所有轴在同一时间到达目标位置。

编程格式：JDO ＜P＞ Vel=＜Value＞ DO[i]=＜ON/OFF＞ {Optional Property} *

其中：

P——目标点位信息；

Vel——关节运动速度百分比，取值范围[1，100]，计数单位为 1%，表示以关节最大速度的百分之

Value 运动；

DO[i]——设置输出信号；

{Optional Property}*——可选择项，如 Vrot，Acc，Dec，Cnt，Offset，Inc，Skip，参见附录 B。

示例：

JDO P[1] Vel=50 DO[128]=ON Cnt=50；

以关节插补的方式向目标位置 P[1]移动，速度为设定关节速度的 50%。若后续无其他运动，则在 P[1]位置，输出信号 DO[128]被置位；若后续有其他运动行，则在平滑的中间位置，输出信号 DO[128]被置位。

4.1.6 LDO 指令

指令功能：工业机器人以直线运动的方式运动至目标点，并在目标点位置或平滑路径中间位置将相应输出信号设置为相应值。

编程格式：LDO <P> Vel=<Value> DO[i]=<ON/OFF> {Optional Property}*

其中：

P——目标点位信息；

Vel——运行速度数据，单位为毫米每秒(mm/s)；

DO[i]——设置输出信号；

{Optional Property}*——可选择项，如 Vrot，Acc，Dec，Cnt，Offset，Inc，Skip，Wjnt，参见附录 B。

示例：

LDO P[1] Vel=1 000 DO[128]=ON Cnt=50；

以直线的方式向目标位置 P[1]移动，速度为 1 000 mm/s，若后续无其他运动，则在 P[1]位置，输出信号 DO[128]被置位；若后续存在其他运动行，则在平滑路径的中间位置，输出信号 DO[128]被置位。

4.1.7 CDO 指令

指令功能：工业机器人通过中间点以圆弧轨迹运动至目标点，并在目标点位置或平滑路径的中间位置将相应输出信号设置为相应值。该指令在 C 指令的基础上增加信号输出功能。

编程格式：CDO <P1> <P2> Vel=<Value> DO[i]=<ON/OFF> {Optional Property}*

其中：

P1/P2——圆弧的中间点以及目标点点位信息，可以是 P[i]，JR[i]，LR[i]或常量任意一种；

Vel ——空间运动速度，单位为毫米每秒(mm/s)；

{Optional Property}*——可选择项，如 Vrot，Acc，Dec，Cnt，Offset，Inc，Skip，Wjnt，参见附录 B。

示例：

CDO P[1] P[2] Vel=1 000 DO[128]=ON Cnt=50；

以圆弧的方式向从当前位置，经过中间位置 P[1]点，向目标位置 P[2]移动，速度为 1 000 mm/s，若后续无其他运动，则在 P[2]位置，输出信号 DO[128]被置位；若后续存在其他运动行，则在平滑路径的中间位置，输出信号 DO[128]被置。

4.1.8 SINGAREA 指令

指令功能：设定工业机器人运动时，在奇异点的插补方式。

编程格式：SINGAREA WristON/WristOFF

其中：

WristON——启用奇异点位姿调整。工业机器人运动时，为了避免在奇异位置(如工业机器人四六轴轴线处于平行位置)报警停机，允许 TCP 的位姿在奇异位置附近有些许改变，避开奇异位置运行。

WristOFF——关闭奇异点位姿调整。工业机器人运动时，不准许 TCP 位姿发生改变，必须严格按

照编程轨迹运行，是工业机器人的默认状态。

当前指令通过对工业机器人位姿进行些许改变，可以绝对避免工业机器人运行时报警停机，但是，工业机器人运行路径会受影响，位姿得不到控制，通常用于通过复杂姿态点，不能作为工作点使用。

示例：

SINGAREA WristON；

启用奇异点位姿调整。

……

SINGAREA WristOFF；

关闭奇异点位姿调整。

4.2 力控制指令

4.2.1 概述

用于设定工业机器人在不同工作状态、不同工作对象时的负载或力控制功能，常用于搬运、码垛、抛光、打磨等工业机器人。

4.2.2 GRIPLOAD 末端负载设置指令

指令功能：设置当前搬运对象的载荷数据，包含质量、重心、力矩轴方向和有效载荷转动惯量。

编程格式：GRIPLOAD ＜Load＞

其中：

Load——载荷数据结构体变量，包括以下参数：

a) Load.mass——有效载荷质量，单位为千克(kg)；

b) Load.cog.x，Load.cog.y，Load.cog.z——有效载荷重心，单位为毫米(mm)；

c) Load.aom.q1，Load.aom.q2，Load.aom.q3，Load.aom.q4——力矩轴方向；

d) ix，iy，iz——有效载荷的转动惯量，单位为千克平方米($kg \cdot m^2$)。

示例：

DO[128]＝ON；

夹具夹紧。

GRIPLOADload1；

设定当前的搬运对象质量和重心 load1。

……

DO[128]＝OFF；

夹具松开。

GRIPLOADload0；

将搬运对象清除为 load0。

4.2.3 MECHUNITLOAD 机械臂负载设置指令

指令功能：设置机械臂负载的载荷数据，包含质量、重心、力矩轴方向和有效载荷转动惯量。

编程格式：MECHUNITLOAD ＜Load＞

其中：

Load——载荷数据结构体变量，包括以下参数：

a) Load.mass——有效载荷质量，单位为千克(kg)；

b) Load.cog.x，Load.cog.y，Load.cog.z——有效载荷重心，单位为毫米(mm)；

c) Load.aom.q1，Load.aom.q2，Load.aom.q3，Load.aom.q4——力矩轴方向；

d) ix，iy，iz——有效载荷的转动惯量，单位为千克平方米($kg \cdot m^2$)。

示例：

DO[128]=ON；

夹具夹紧。

MECHUNITLOAD load1；

设定当前机械臂对象负载的质量和重心为 load1。

……

DO[128]=OFF；

夹具松开。

MECHUNITLOAD load0；

将当前机械臂对象负载的数据设定为 load0。

4.2.4 FORCEMODE 力控模式选择指令

指令功能：设置工业机器人的力控制模式。

编程格式：FORCEMODE ＜Value＞

其中：

Value——其值表示控制模式编号，支持 0/1/2 三种模式：

a) 0——非力控制模式；

b) 1——阻抗模式；

c) 2——力追踪模式。

示例：

FORCEMODE 1；

设定当前运动模式为阻抗模式。

4.2.5 FORCECMD 力追踪目标值设置指令

指令功能：设置力追踪目标值。

编程格式：FORCECMD ＜Value＞

Value——力追踪目标值，单位为牛顿(N)。

注： 可在每段运动前设置不同的目标值以进行动态力控，仅在力追踪模式下有效。

示例：

FORCEMODE 2；

设置当前运动模式为力追踪模式。

FORCECMD 50；

设置下段运动的力控目标值为 50 N。

4.2.6 FORCEPARA 阻抗参数设置指令

指令功能：设置各阻抗参数大小。

编程格式：FORCEPARA {AxisDirection} [Stiff/Damp]=＜Value＞

其中：

AxisDirection——设置阻抗参数影响的力的方向，有 X、Y、Z、A、B、C 六个可设置的方向。该参数为可选参数，实际使用时如果指令中不包含该参数时，默认会将参数设置到所有方向上；

Stiff——刚性系数；

Damp——阻尼系数。

示例：

FORCEPARA Y Stiff=2 000；

设置了 Y 方向的刚性系数为 2 000。

FORCEPARA Stiff=1 000；

设置 X、Y、Z、A、B、C 每个方向的刚性系数分别为 1 000。

4.3 速度控制指令

4.3.1 概述

速度控制指令指对工业机器人关节或运动轴的运动速度、加速度、加加速度进行设置的指令，根据工业机器人不同负载情况，设置合适的值。

4.3.2 ACC 加速度控制指令

指令功能：修改加速度的值，平滑运动控制效果。当处理较大负载时，使用 ACC 指令降低加速度或加速度坡度。它可以调节工业机器人关节、轴的加速度和加速度变化，使工业机器人运动平滑。

编程格式：ACC ＜Value1＞ ＜Value2＞

其中：

Value1——加速度百分比，取值范围[1，100]，计数单位为 1%，表示实际加速度为最大加速度的百分之 Value1。

Value2——加加速度百分比，取值范围[1，100]，计数单位为 1%，表示实际加加速度为最大加加速度的百分之 Value1。

示例：

ACC 100 100；

默认加速度及加加速度。

ACC 30 100；

加速度被限制到最大值的 30%。

ACC 100 50；

加加速度值被限制到最大值的 50%。

4.3.3 VORD 速度修调指令

指令功能：对运动指令速度进行比例修调。

编程格式：VORD ＜Value＞

其中：

Value——工业机器人运行速率百分比，取值范围[1，100]，计数单位 1%，表示将当前速度设置为最大速度的百分之 Value。

示例：

VORD=50；

速度百分比为 50%，最大速度设置为 800 mm/s。

L P[1] Vel=1 000；

以直线运动到目标位置 P[1]点，运行速度设定为 1 000 mm/s，实际运行速度为 1 000×50%=500 mm/s。

4.4 坐标系设置指令

4.4.1 概述

坐标系指令用于改变工业机器人当前工作所使用的坐标系的设置。

4.4.2 UT 指令

指令功能：工具坐标系选择指令，改变当前工具坐标系的序号。

编程格式：UT NUM=＜Value＞

其中：

Value——R[i]或整型常量，取值范围[－1，15]。－1 表示使用默认坐标系，0～15 表示用户自定义的工具坐标系。工具坐标系的值对应 UT[0] ～UT[15]这 16 个寄存器。

示例：

UT NUM=1；

设置工具坐标系为 UT[1]。

UT NUM=－1；

设置工具坐标系为默认工具坐标系。

4.4.3 UF 指令

指令功能：工件坐标系选择指令，改变当前工件坐标系的序号。

编程格式：UF NUM=＜Value＞

其中：

Value——R[i]或整型常量，取值范围[－1，15]。－1 表示使用默认坐标系，0～15 表示用户自定义的工件坐标系。工件坐标系的值对应 UF[0] ～UF[15]这 16 个寄存器。

示例：

UF NUM=1；

设置工件坐标系为 UF[1]。

UF NUM=－1；

设置工件坐标系为默认工件坐标系。

4.5 寄存器操作指令

4.5.1 概述

对程序所涉及的常规寄存器、特殊寄存器进行设置和操作的指令。

4.5.2 常规寄存器操作指令

指令功能：给常规寄存器赋值，寄存器是一个存储数据的变量。

编程格式：R[i]/TIMER[i]=＜Value＞

注：上述指令把数值右端 Value 值赋给指定的左端寄存器。其中，i 的范围根据不同寄存器定义不同，Value 可以取常数或相同数据类型的寄存器值。

示例：

R[1]=100；

将实际值 100 赋值给 R[1]寄存器。

TIMER[1]=R[1]；

将 R[1]寄存器的值赋值给 TIMER[1]寄存器。

4.5.3 位姿寄存器操作指令

指令功能：位姿寄存器是一个存储位姿数据的变量，该指令的功能是给位姿寄存器赋值。

编程格式：JR[i]/LR[i]=＜Value＞

注：JR[i]/LR[i] = ＜Value＞指令把数值 Value 赋值给指定的位姿寄存器。JR/LR 寄存器 i 的范围是 0～999；针对 JR[i]/LR[i]赋值要注意类型匹配(工业机器人维度匹配)。

示例：

JR[1]={0，－90，180，0，90，0}；

将实际六轴工业机器人关节角坐标值赋值给寄存器 JR[1]。

4.5.4 位姿寄存器单轴操作指令

指令功能：位姿寄存器单轴数据赋值指令，在位姿寄存器上完成单轴位置赋值。

编程格式：JR[i][j]/LR[i][j]=<Value>

注：JR[i][j]/LR[i][j]=<Value>指令把数值<Value>赋值给指定的位姿寄存器元素。其中，JR[i][j]/LR[i][j]中的元素，i 代表位姿寄存器的序号，j 代表位姿寄存器元素序号。Value 值可以取常数、寄存器(R)、位姿寄存器中的某个轴(JR[i][j]/LR[i][j])、位姿变量中的某个轴(P[i][j])。

示例：

JR[1][1]=0.0；

设置寄存器 JR[1]的第一个元素的值为 0。

4.5.5 数字输入输出寄存器操作指令

DI(数字输入指令)和 DO(数字输出指令)是用来指示外部输入状态或系统输出状态的输入输出信号寄存器。

4.5.5.1 读操作指令

指令功能：把数字输入信号[ON(1)/OFF(0)]赋值给指定的 R 寄存器。

编程格式：R[i]=DI[i]

其中：

R[i]中的 i——寄存器 0～999；

DI[i]中的 i——数字输入端口号。

示例：

R[1]=DI[128]；

读取数字输入 DI[128]的状态并保存到 R[1]寄存器中。

4.5.5.2 写操作指令

写操作指令包括以下三种形式：

a) DO[i]=<Value>

指令功能：把信号(ON/OFF)赋值给指定的数字输出信号。

编程格式：DO[i]=<Value>

其中：

DO[i]中的 i——数字输出端口号；

Value——为 ON 表示打开数字输出端口；OFF 表示关闭数字输出端口。

b) DO[i] =PLUSE <Value>

指令功能：产生取反状态脉冲。

编程格式：DO[i]=PLUSE <Value>

其中：

DO[i]中的 i——数字输出端口号；

Value——状态取反时间(sec)。

c) DO[i]=R[i]

指令功能：将指定寄存器的值赋值给指定数字输出端口。当指定的寄存器的值为 0 时，数字输出 OFF；当为非零时，数字输出为 ON。

编程格式：DO[i] = R[i]

其中：

DO[i]中的 i——数字输出端口号；

R[i]中的 i——寄存器 0～999。

示例：

DO[128]=ON；

设置数字输出 DO[128]的状态为 ON 状态；

DO[128]=PULSE 1；

数字输出 DO[128]从 ON 状态切换到 OFF 状态，并保持 1 s 时间。

4.5.6 模拟量输入输出操作指令

4.5.6.1 读操作指令

指令功能：将模拟输入信号赋值给指定的 R 寄存器。

编程格式：R[i]=AI[i]

其中：

AI[i]中的 i——模拟输入端口号；

R[i]中的 i——寄存器 0～999。

示例：

R[1]=AI[1]；

读取模拟输入 AI[1]的值并保存到 R[1]寄存器中。

4.5.6.2 写操作指令

写操作指令包括以下两种形式：

a) AO[i]=<Value>

指令功能：将数值 Value 作为指定的模拟输出信号的值。

编程格式：AO[i] = <Value>

其中：

AO[i]中的 i——模拟输出端口号；

Value——模拟输出信号的值。

b) AO[i]=R[i]

指令功能：将指定寄存器的值赋值给指定模拟输出端口。

编程格式：AO[i]=R[i]

其中：

AO[i]中的 i——模拟输出端口号；

R[i]中的 i——寄存器 0～999。

示例：

AO[1]=5.5；

将值 5.5 赋值给模拟量输出 AO[1]；

AO[1]=R[1]；

将寄存器 R[1]的值赋值给模拟量输出 AO[1]。

4.6 数据处理指令

4.6.1 概述

数据处理指令对程序数据进行清除、设定相关操作。

4.6.2 BITC 复位指令

指令功能：清除(设定为 0)定义的字节数据中一个特定的位。

编程格式：BITC ＜BitData＞ ＜BitPos＞

其中：

BitData——将要被改变的数据，整型数据，范围是十进制的 0～255；

BitPos ——整型数字，BitData 中将被设为零的位置索引，有效位的位置为 1～8。

示例：

BITC R[i] 8；

变量 R[i]中的第 8 位清 0，若 R[i]＝130，则清零后，R[i]＝2。

4.6.3 BITS 置位指令

指令功能：在定义的字节数据中把某一特定位设置为 1。

编程格式：BITS BitData BitPos

其中：

BitData——将要被改变的数据，整型数据，范围是十进制的 0～255；

BitPos ——整型数字，BitData 中将被设为 1 的位的位置，有效位的位置为 1～8。

示例：

BITS R[i] 8；

变量 R[i]中的第 8 位设为 1，若 R[i]＝2，则设置特定位后，R[i]＝130。

4.6.4 CLEARBUF 串行输入缓冲清除指令

指令功能：清理串行通道的输入缓冲区数据，所有来自串行通道的缓冲区字符将被丢弃。

编程格式：CLEARBUF Iodev

其中：

Iodev——将被清空缓冲区的串行通道名称。该指令只能被串行通道使用，不等待操作完成的通知，在每一次使用时，应在指令后使用等待指令给操作以足够的时间。

示例：

CLEARBUF “com2”；

清空 com2 端口通道缓冲区的所有数据。

4.7 流程控制指令

4.7.1 概述

流程控制指令指对工业机器人程序的执行顺序产生影响的指令。

4.7.2 IF 逻辑判断指令

指令功能：逻辑条件判断。

编程格式：IF ＜Condition＞ GOTO LBL[＜Value＞]/CALL ＜Subroutine＞

注：如 Condition 条件成立，则执行后面跳转或子程序调用语句；如 Condition 条件不成立，则继续往下执行。

示例：

IF DO[1]＝OFF GOTO LBL[2]；

当条件 DO[1]等于 OFF 成立时，程序跳转到 LBL[2]标签处继续执行；当条件不成立，则继续执行该行程序后面的语句。

4.7.3 SELECT 条件选择指令

指令功能：条件选择判断。

编程格式：

SELECT ＜Expression＞ ＝ 1 GOTO LBL[＜Value＞]/CALL ＜Subroutine＞
＝ 2 GOTO LBL[＜Value＞]/CALL＜Subroutine＞
＝ 99 GOTO LBL[＜Value＞]/CALL ＜Subroutine＞
ELSE GOTO LBL[＜Value＞]/CALL ＜Subroutine＞

注：计算＜Expression＞中表达式的值，并逐个与等号后的常量表达式值相比较，当表达式的值与等号后的某个常量表达式的值相等时，即执行其后的语句；如果表达式的值与所有等号后的常量表达式均不相同时，则执行ELSE后的语句。

示例：

```
SELECT R[1]=1 GOTO LBL[1]
       =2 CALL Sub1
       ELSE GOTO LBL[3]
```

当R[1]等于1时，程序跳转到LBL[1]标签处继续执行；当R[1]等于2时，调用名为“Sub1”的子程序执行；R[1]等于其他值时，程序跳转到标签LBL[3]处继续执行。

4.7.4 CALL 程序调用指令

指令功能：子程序调用。

编程格式：CALL ＜Subroutine＞

注：调用名称Subroutine的子程序。

示例：

```
CALL Sub1;
```

调用名为“Sub1”的子程序执行。

4.7.5 GOTO 程序跳转指令

指令功能：程序跳转。

编程格式：GOTO LBL[＜Value＞]

Value——正整数，表示标签号。

注：程序跳转到对应LBL执行，Value为跳转LBL标签号。

示例：

```
J P[1] Vel=100 Cnt=30
GOTO LBL[1]
……
LBL[1]
L P[2] Vel=1 000 Cnt=50
……
```

程序执行完J P[1]指令后，跳转到标签LBL[1]处，继续执行L P[2]指令行。

4.7.6 LBL 程序标签指令

指令功能：程序标签。

编程格式：LBL[＜value＞]

注：作为GOTO跳转语句的跳转标签使用。

示例：

J P[1] Vel=100 Cnt=30

GOTO LBL[1]

……

LBL[1]

L P[2] Vel=1 000 Cnt=50

……

设置跳转标签 LBL[1]，执行 GOTO LBL[1]时，程序跳转到该标签处，继续执行后面的程序指令。

4.7.7 STOPMOTION 暂停当前程序运动行指令

指令功能：为了方便程序调试，在程序执行的某个位置设置立即跳出，暂停当前程序执行。该指令立即停止程序执行，不用等工业机器人或者外部轴到达其编程所规定的目的点。程序再次执行时，从当前位置的下一条指令开始执行。

编程格式：STOPMOTION

示例：

……

STOPMOTION

……

执行 STOPMOTION 指令时，工业机器人所有运动立即停止。

4.7.8 CALLBYV 变量调用程序指令

指令功能：通过字符串变量调用指定程序。

编程格式：CALLBYV ＜Value＞

其中：

Value——被调用的程序名，字符串类型。

示例：

Var="proc001";

CALLBYV Var;

调用指定的 proc001 程序。

4.8 位置补偿指令

4.8.1 OFFSET CONDITION 条件补偿指令

指令功能：此指令可以将程序运动行中确定的目标位置点进行偏移，偏移量由指令中设定的偏移位置决定。此指令执行后，后续所有运动指令的点位都按照设定的偏移值进行偏移。

编程格式：OFFSET CONDITION LR[i]

注：此指令格式为运动附加指令，不能单独使用，跟随在运动语句后，此时该运动语句点位按照 LR[i]进行偏移。

示例：

OFFSET CONDITION LR[1];

LP[2] Vel=100;

通过位置补偿指令 OFFSET CONDITION 设置位置补偿值为 LR[1]，后续执行 LP[2]指令时，实际目标位置为 P[2]+LR[1]。

4.8.2 OFFSET 运动附加指令

指令功能：此指令单独设置运动行偏移，将当前运动行的目标位置偏移设定的补偿量。不可单独

使用。

编程格式：OFFSET LR[i]

示例：

LP[3] Vel=100 OFFSET LR[2]；

当前运动行目标位置为 P[3]点，使用附加指令 OFFSET 后，实际目标位置为 P[3]+LR[2]。当前运动行会运动到 P[3]+LR[2]的位置上。

4.9 运算指令

4.9.1 概述

运算指令指对程序中相关数据进行算数运算或逻辑运算的指令。

4.9.2 算数运算指令

4.9.2.1 概述

算数运算指令包括以下几种，配合寄存器指令使用，作为其中某一运算符号：

a) SIN；
b) ASIN；
c) COS；
d) ACOS；
e) ATAN2；
f) MOD；
g) DIV。

4.9.2.2 SIN 指令

指令功能：求给定角度的正弦值。

编程格式：SIN(＜Expression＞)

其中：

Expression——角度值或角度值计算表达式，单位是度(°)。

示例：

R[1]=SIN(30)；

计算 30°角的正弦值，并将结果赋值给 R[1]。

4.9.2.3 ASIN 指令

指令功能：求给定值的反正弦值。

编程格式：ASIN(＜Expression＞)

其中：

Expression——浮点类型数值或计算表达式，取值范围[－1，1]。

示例：

R[1]=ASIN(0.6)；

计算 0.6 的反正弦值，并将结果赋值给 R[1]。

4.9.2.4 COS 指令

指令功能：求给定角度的余弦值。

编程格式：COS(＜Expression＞)

其中：

Expression——角度值或角度值计算表达式，单位是度(°)。

示例：

R[1]=COS(30)；

计算 30°角的余弦值，并将结果赋值给 R[1]。

4.9.2.5 ACOS 指令

指令功能：求给定值的反余弦值。

编程格式：ACOS(<Expression>)

其中：

Expression——浮点类型数值或计算表达式，取值范围[−1，1]。

示例：

R[1]=ACOS(0.6)；

计算 0.6 的反余弦值，并将结果赋值给 R[1]。

4.9.2.6 ATAN2 指令

指令功能：求欧拉角度。

编程格式：ATAN2(<Expression1>，<Expression2>)

注： 函数返回值在−π～+π之间。通过<Expression1>和<Expression2>联合确定欧拉角度，使得所得角度值唯一。当表达式 Expression2 大于 0 时，该指令计算结果为 Expression1/ Expression2 的反正切值；当表达式 Expression2 小于 0，且表达式 Expression1 大于或等于 0 时，该指令计算结果为 Expression1/ Expression2 的反正切值加上 π/2；当表达式 Expression2 小于 0，且表达式 Expression1 也小于 0 时，该指令计算结果为 Expression1/ Expression2 的反正切值减去 π/2；当表达式 Expression2 等于 0，且表达式 Expression1 大于 0 时，该指令计算结果为 π/2；当表达式 Expression2 等于 0，且表达式 Expression1 小于 0 时，该指令计算结果为−π/2；当表达式 Expression2 等于 0，且表达式 Expression1 也等于 0 时，该指令会报错，无法计算。

示例：

R[1]=ATAN2(1，2)；

计算 1/2 的反正切值，并将结果赋值给 R[1]。

4.9.2.7 MOD 指令

指令功能：求余。

编程格式：<Expression1> MOD <Expression2>

注： MOD 指令计算 Expression1 除以 Expression2 所得到的余数并返回该值。

示例：

R[1]=1 MOD 2；

计算 1 除以 2 的余数，并将结果(1)赋值给 R[1]。

4.9.2.8 DIV 指令

指令功能：求商的整数部分。

编程格式：<Expression1> DIV <Expression2>

注： DIV 指令计算 Expression1 除以 Expression2 所得的商的整数部分并返回该计算结果。

示例：

R[1]=1 DIV 2；

计算得到 1 除以 2 的结果的整数部分，并将结果(0)赋值给 R[1]。

4.9.3 逻辑运算指令

4.9.3.1 概述

逻辑运算指令包括以下几种，配合寄存器指令使用，作为其中某一运算符号。

a) AND，BAND；
b) OR，BOR；
c) NOT，BNOT；
d) XOR，BXOR；
e) NXOR，BNXOR。

4.9.3.2 AND 指令

指令功能：求两个数据的逻辑与。

编程格式：<Expression1> AND <Expression2>

注：计算 Expression1、Expression2 两个数据的逻辑与并返回结果。

示例：

R[1]=1 AND 2；

计算 1 与 2 的逻辑与，并将结果(1)赋值给 R[1]。

4.9.3.3 BAND 指令

指令功能：按位求两个数据的逻辑与。

编程格式：<Expression1>BAND <Expression2>

注：将 Expression1 与 Expression2 进行按位的逻辑与操作，返回操作结果。

示例：

R[1]=1 BAND 2；

按位计算 1 与 2 的逻辑与，并将结果(0)赋值给 R[1]。

4.9.3.4 OR 指令

指令功能：求两个数据的逻辑或。

编程格式：<Expression1> OR<Expression2>

注：计算 Expression1、Expression2 两个数据的逻辑或并返回结果。

示例：

R[1]=2 OR 3；

计算 2 与 3 的逻辑或，并将结果(1)赋值给 R[1]。

4.9.3.5 BOR 指令

指令功能：按位求两个数据的逻辑或。

编程格式：<Expression1> BOR<Expression2>

注：将 Expression1 与 Expression2 进行按位的逻辑或操作，返回操作结果。

示例：

R[1]=1 BOR 2；

按位计算 1 与 2 的逻辑或，并将结果(3)赋值给 R[1]。

4.9.3.6 NOT 指令

指令功能：求数据的逻辑非。

编程格式：NOT <Expression>

注：计算 Expression 的逻辑非。

示例：

IF NOT DI[1] GOTO LBL[2]；

DI[1]为 ON 时，条件表达式为假，继续执行后续指令行；DI[1]为 OFF 时，条件表达式为真，程序跳转到标签 LBL[2]处继续执行。

4.9.3.7 BNOT 指令

指令功能：按位求数据的逻辑非。

编程格式：BNOT ＜Expression＞

注：按位计算 Expression 数据的逻辑非。

示例：

R[1]＝BNOT 10；

计算十进制数 10 反码，将结果(5)赋值给 R[1]。

4.9.3.8 XOR 指令

指令功能：求两个数据的逻辑异或。

编程格式：＜Expression1＞XOR ＜Expression2＞

注：计算 Expression1 与 Expression2 的逻辑异或。

示例：

R[1]＝1 XOR 2；

计算 1 与 2 的逻辑异或，将结果(0)赋值给 R[1]。

4.9.3.9 BXOR 指令

指令功能：按位求两个数据的逻辑异或。

编程格式：＜Expression1＞BXOR ＜Expression2＞

注：按位计算 Expression1 与 Expression2 的逻辑异或。

示例：

R[1]＝1 XOR 2；

按位计算 1 与 2 的逻辑异或，将结果(3)赋值给 R[1]。

4.9.3.10 NXOR 指令

指令功能：求两个数据的逻辑同或。

编程格式：＜Expression1＞NXOR＜Expression2＞

注：计算 Expression1 与 Expression2 的逻辑同或。

示例：

R[1]＝1 NXOR 2；

计算 1 与 2 的逻辑同或，将结果(1)赋值给 R[1]。

4.9.3.11 BNXOR 指令

指令功能：按位求两个数据的逻辑同或。

编程格式：＜Expression1＞BNXOR＜Expression2＞

注：按位计算 Expression1 与 Expression2 的逻辑同或。

示例：

R[1]＝1 BNXOR 2；

按位计算 1 与 2 的逻辑同或，将结果(0)赋值给 R[1]。

4.10 其他指令

4.10.1 概述

工业机器人与数控机床或其他设备协同作业时，工业机器人数控系统通过协作控制指令对设备间的同步和时序进行控制。

4.10.2 CLEARPATH 当前路径清除指令

指令功能：清除当前运动路径层上的整个运动路径。运动路径指在指令执行时，已经执行但是工业机器人没有完成的所有运动路径。此指令执行前工业机器人应处于停止状态，或者用 STOPMOTION 指令停止工业机器人。

编程格式：CLEARPATH

示例：

```
J P[1] Vel=100
WAIT DO[1]=ON
STOPMOTION
CLEARPATH
```

程序运行到 J P[1]时，会检测数字量输出 DO[1]是否被置位。若 DO[1]被置位，则执行 STOPMOTION 停止工业机器人运行，然后执行 CLEARPATH 将之前未完成的运行指令从运动缓冲区中全部清除。

4.10.3 TIMER[i]计时器指令

指令功能：计时功能。

编程格式 1：TIMER[i] <TimerState>(计时操作)

注： i 的取值范围为 0～99，分别对应一个计时器。TimerState 取值范围为 START/STOP/RESET，分别对应着启动、停止、复位。执行 TIMER[i] START 后，开始计时，计时时间保存在 TIMER[i]中；执行 TIMER[i] STOP 后，计时停止，此时 TIMER[i]内保存从起动到停止所消耗的时间；执行 TIMER[i] RESET 后，计时器重置，TIMER[i]清零。

编程格式 2：R[j] = TIMER[i](赋值操作)

注： 将索引为 i 的计时器当前值赋值给左端 R[j]寄存器。

示例：

```
TIMER[1] START
J P[1] Vel=100 Acc=100 Dec=100 Cnt=10
TIMER[1] STOP
R[1]=TIMER[1]
```

该示例计算了执行 J P[1]指令所需要的时间。

4.10.4 WAIT DI/DO 等待指令

指令功能：等待数字输入信号、数字输出信号对应的状态与设定状态一致，则继续之后后续指令；若不一致，则阻塞程序运行。

编程格式：WAIT DI[i]/DO[i] =<Value>

注： DI[i]/DO[i]表示索引为 i 的数字量输入或者数字量输出，Value 为 ON 或 OFF。当索引为 i 的 IO 状态与指定的 Value 值相同时，等待指令执行结束。等待指令对某一动作或功能产生延时效果。

示例：

```
J P[1] Vel=100 Acc=100 Dec=100 Cnt=30
WAIT DO[1]=ON
J P[2] Vel=100 Acc=100 Dec=100 Cnt=30
```

执行 J P[1]指令过程中，检测 DO[1]是否为 ON：若 DO[1]信号在 J P[1]执行完成之前置位，工业机器人会平滑过

渡到 P[1]至 P[2]的路径上;若 DO[1]信号一直为 OFF 状态,则在执行完 J P[1]后,工业机器人在 P[1]位置等待信号。

4.10.5 TRIGGERIO 信号触发指令

指令功能:工业机器人在运动的同时精确输出相应信号。

编程格式:

TRIGGERIO Mode=[DISTL/TIME START/END] <Value> [DO[i]=<ON/OFF>/AO[i]=<ON/OFF>]

其中:

Mode——触发模式及触发基准设置,可设置为距离触发模式或时间触发模式,触发基准为下一行运动的起始点或目标点;

Value——触发条件,根据 Mode 模式的选择,可以表示为相对于触发基准的距离或时间值,距离单位为毫米(mm),时间单位为毫秒(ms);

DO[i]=<ON/OFF>/AO[i]=<ON/OFF>——信号触发设置,设置数字输出或模拟输出为设定值;

示例:

TRIGGERIO Mode=DISTL START 100 DO[128]=ON

L P[1] Vel=1 000

在 L P[1]运动行开始执行时进行 TRIGGERIO 的扫描,当距离起始位置 100 mm 时 DO[128]被置位。

4.10.6 空间区域设定指令

4.10.6.1 概述

创建一个多工业机器人公用的区域,该区域在同一时间内只允许其中一个工业机器人使用。用来限制工业机器人的工作区域,避免碰撞。可以在工业机器人的工作区域内设定多个不同类型的体积空间。

4.10.6.2 WZCYLDEF 指令

指令功能:用来定义一个圆柱体的世界坐标系区域。该圆柱的轴线平行于世界坐标系的 Z 轴。

编程格式:WZCYLDEF <CenterPoint> <Radius> <Height>

其中:

CenterPoint——定义圆柱底面圆心位置,笛卡尔坐标值;

Radius——圆柱顶面圆周半径,单位为毫米(mm);

Height——定义圆柱高度,单位为毫米(mm)。

示例:

WZCYLDEFPR[i] 100 200;该行指令定义一个圆柱体的空间,底面圆心为 LR[i],半径 100 mm,高度 200 mm。

4.10.6.3 WZBOXDEF 指令

指令功能:用来定义一个直立箱体的世界坐标系区域。该箱体空间的所有边都与世界坐标系坐标轴平行。

编程格式:WZBOXDEF <P1> <P2>

其中:

P1/P2——箱体对角点位笛卡尔坐标,姿态值与世界坐标系保持一致。

注:需要注意在姿态为世界坐标系姿态时,P1 与 P2 的 X、Y、Z 应都不相等,否则指令提示参数错误;

示例:

WZBOXDEF LR[1] LR[2];该行指令定义了一个直立箱体空间,该箱体的所有边都和世界坐标系的轴平行,该箱体由两个对角点 LR[1]和 LR[2]定义。

附 录 A
（资料性附录）
典型编程程序格式框架

典型编程程序格式框架为：

```
<attr>
GROUP:[0]
<end>

<pos>
P[1]{GP:0,UF:0,UT:0,JNT:[-0.0,-90.0,180.0,-0.0,90.0,0.0,0.0,0.0,0.0,0.0]};
P[2]{GP:0,UF:0,UT:0,CFG:[1,0,0,0,0,0],LOC:[582.476,-53.595,185.323,-116.481,
-158.704,-32.043,0,0,0]};
P[3]{GP:0,UF:0,UT:0,CFG:[1,0,0,0,0,0],LOC:[484.988,155.816,222.75,0.0,180.0,
-0.0,0,0,0]};
P[4]{GP:0,UF:0,UT:0,CFG:[1,0,0,0,0,0],LOC:[412.411,347.781,222.749,-0.0,
-180.0,0.0,0,0,0]};
<end>

<program>
LBL[1]
R[1]=1
IF R[1]=2 , GOTO LBL[2]
J P[1] Vel=100 Acc=100 Cnt=10 Dec=100
WAIT TIME=1000
L P[2] Vel=1200 Acc=100 Cnt=10 Dec=100
C P[3] P[4] Vel=1200 Acc=70 Dec=70
GOTO LBL[1]
<end>
```

附 录 B
（资料性附录）
J、L、C 指令可选操作参数说明

表 B.1 给出了 J、L、C 指令可选操作参数说明。

表 B.1 J、L、C 指令可选操作参数说明表

指令类型	可选参数	全局参数
J 关节移动指令	Vel 速度	J_VEL=0～100，单位%
	Acc 加速因子	J_ACC=0～100，单位%
	Dec 减速因子	J_DEC=0～100，单位%
	Cnt 平滑过渡系数	CNT=0～100，单位%
	Offset 位置补偿	
	Inc 增量编程	
	Skip 跳过	
L 直线移动指令	Vel 速度	L_VEL=0～1000，单位 mm/s
	Acc 加速因子	L_ACC=0～100，单位%
	Dec 减速因子	L_DEC=0～100，单位%
	Vrot 姿态速度	L_VROT=0～100，单位%
	Cnt 平滑过渡系数	CNT=0～100，单位%
	Offset 位置补偿	
	Inc 增量编程	
	Skip 跳过	
	Wjnt 腕关节动作	
C 圆弧移动指令	Vel 速度	C_VEL=0～1000，单位 mm/s
	Acc 加速因子	C_ACC=0～100，单位%
	Dec 减速因子	C_DEC=0～100，单位%
	Vrot 姿态速度	C_VROT=0～100，单位%
	Cnt 平滑过渡系数	CNT=0～100，单位%
	Offset 位置补偿	
	Inc 增量编程	
	Skip 跳过	
	Wjnt 腕关节动作	