ICS 25.040.20
J 50

中华人民共和国国家标准

GB/T XXXX—XXXX

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

（征求意见稿）

2018 年 2 月

- - 发布　　- - 实施

中华人民共和国国家质量监督检验检疫总局
中国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。

本标准主要起草单位：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等。

本标准主要起草人：周星等

引　言

工业机器人是一种集机械、电子、控制、计算机、传感器、人工智能等多学科先进技术于一体的现代制造业重要自动化装备。它是衡量一个国家制造业水平和科技水平的重要标志。当前我国劳动力资源不断稀缺，人力成本不断增加，同时生产效率、生产质量的要求进一步提高，因此工业机器人的应用越来越得到重视。在“中国制造2025”国家战略的主导下和“机器换人”已经成为大趋势的背景下，工业机器人将迎来更为广阔的发展空间，工业机器人行业将进入高速发展阶段。

随着工业机器人的高速发展，研究机器人编程语言的机构越来越多。与此同时，各研究机构按照自己的想法和工作经验设计机器人编程语言，导致了机器人编程格式不统一。由于机器人编程格式不统一，编程语言通用性不好，不便于进行技术交流，这在一定程度上阻碍了工业机器人的技术发展。另一方面，机器人编程格式不统一给用户编写机器人操作程序造成很多困难，甚至会因为误操作产生危险的后果，这会对机器人的应用推广产生影响。因此需要制定机床工业机器人数控系统的编程语言标准，引导工业机器人编程语言朝兼容性好、通用性高的方向发展，以此来促进工业机器人领域研究人员的技术交流和工业机器人的技术进步；同时使用户编程操作更安全、简洁、高效，为用户提高更好的操作体验，以利于工业机器人的应用推广。

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，包括编程语言中的指令类型和功能。
本标准适用于机床工业机器人数控系统。

2　术语与定义

下列术语和定义适用于本文件。

2.1 运动指令

运动指令 move industrial instruction
运动指令是指对工业机器人各关节转动、移动运动控制的相关指令。

2.2　IO 指令

IO指令　IO industrial
输入输出指令用于操作I/O 的状态(读取输入或设置输出)。

2.3 流程控制指令

流程控制指令flow control industrial
流程控制指令是对机器人操作程序的执行顺序产生影响的指令。

2.4 等待指令

等待指令time delay industrial
等待指令对某一动作或功能产生延时效果。

2.5 坐标系、寄存器指令

坐标系、寄存器及其处理指令　coordinate system、register and treatment industrial
坐标系、寄存器及其处理指令是对工业机器人坐标系、相关寄存器配置及其操作的相关指令。

2.6 运算指令

运算指令arithmetic industrial
运算指令是对程序中相关数据进行算数运算或逻辑运算的指令，分为算数运算指令和逻辑运算指令。

2.7 程序框架指令

程序框架指令　program frame industrial
程序框架指令是整个程序的框架结构，其一般由示教系统自动完成。

3　编程语言指令类型

机床工业机器人数控系统编程语言的指令包括以下几种：

a)　运动指令；
b)　输入输出指令；
c)　流程控制指令；
d)　等待指令；
e)　坐标系、寄存器及其处理指令；

f) 运算指令；
g) 程序框架指令。

4 指令功能与格式

4.1 运动指令

4.1.1 概述

运动指令是指对工业机器人各关节转动、移动运动控制的相关指令。运动指令包括以下三种：

a) J；
b) L；
c) C；

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[i]/PR[i] VEL=(value) {Optional Property}*

注释：

P[i]/PR[i]: 末端点点位信息
P[i]：局部点位信息
PR[i]：全局点位信息
VEL=(value):关节运动速度单位为 1%~100%
{Optional Property}:可选择项，例如 ACC, DEC, CNT,OFFSET,PTH,INC,SKIP,WJNT。

4.1.3 L 指令

指令功能：以笛卡尔坐标方式插补进行的直线运动到目标点运动

编程格式：L P[i]/PR[i] VEL=(value) {Optional Property}*

注释：

P[i]/PR[i]:末端点点位信息
P[i]： 局部点位信息
PR[i]：全局点位信息
VEL=(value):空间运动速度单位为 mm/s
{Optional Property}:可选择项，例如 VROT,ACC, DEC, CNT,OFFSET,PTH,INC,SKIP,WJNT。

4.1.4 C 指令

指令功能：执行圆弧运动经过中间点，最终到目标点

编程格式：C P[i]/PR[i] P[i]/PR[i] {P[i]/PR[i]} VEL=(value) {Optional Property}*

注释：

P[i]/PR[i]:仅有两个点，则执行圆弧运动，其中。第一个为中间点点位信息，第二个为末端点点位信息。如存在三个点，则进行整圆运动，从第一个点，经过第二，第三个点，最终停止在第一个点。
P[i]： 局部点位信息
PR[i]：全局点位信息
VEL=(value):空间运动速度单位为 mm/s
{Optional Property}:可选择项，例如 VROT,ACC, DEC, CNT,OFFSET,PTH,INC,SKIP,WJNT。

4.2 输入输出指令

4.2.1 概述

输入输出指令用于操作输入输出的状态（读取输入或设置输出）。输入输出指令包括以下几种：

a) DI/DO（数字输入/输出）；
b) AI/AO（模拟输入/输出）；

4.2.2 DI/DO 指令

4.2.2.1 概述

DI（数字输入指令）和 DO（数字输出指令）是可以被用户控制的输入输出信号。

4.2.2.2 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]
注释：R[i]中的 i 表示寄存器 0 到 999;
DI[i]中的 i 表示数字输入端口号。

4.2.2.3 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)
注释：DO[i]中的 i 表示数字输出端口号;
(value)—— ON：打开数字输出端口。
OFF：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)
注释：DO[i]中的 i 表示数字输出端口号;
(value):表示状态取反时间(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]
注释：DO[i]中的 i 表示数字输出端口号;
R[i]中的 i 表示寄存器 0 到 999。

4.2.3 AI/AO 指令

4.2.3.1 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]
注释：AI[i]中的 i 表示模拟输入端口号;
R[i]中的 i 表示寄存器 0 到 999。

4.2.3.2 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)
注释：AO[i]中的 i 表示模拟输出端口号;
(value)表示模拟信号的值(constant)。

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]
注释：AO[i]中的 i 表示模拟输出端口号;
R[i]中的 i 表示寄存器 0 到 999。

4.3 流程控制指令

4.3.1 概述

流程控制指令是对机器人操作程序的执行顺序产生影响的指令。流程控制指令包括以下几种：

a) IF;
b) SELECT;
c) CALL;
d) GOTO;
e) LBL;

4.3.2 IF 指令

指令功能：逻辑判断

编程格式：

IF <condition> ,GOTO LBL<value>/CALL <value>

注释：如果<condition>条件成立，则执行逗号后面的语句;如果<condition>条件不成立，则继续往下执行。

4.3.3 SELECT 指令

指令功能：条件选择

编程格式：

SELECT (expression)= 1,GOTO LBL<value>/CALL <value>
= 2,GOTO LBL<value>/CALL <value>
= 99,GOTO LBL<value>/CALL <value>
ELSE,GOTO LBL<value>/CALL <value>

注释：计算 SWITCH(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。

4.3.4 CALL 指令

指令功能：程序调用

编程格式：CALL <value>

注释：调用 value 名称的程序。

4.3.5 GOTO 指令

指令功能：程序跳转

编程格式：GOTO LBL<value>

注释：程序跳转到对应 LBL 执行。

4.3.6 LBL 指令

指令功能：程序标签

编程格式：LBL<value>

注释：作为 GOTO 跳转语句的跳转标签使用。

4.4 等待指令

指令功能：数字输入信号，数字输出信号等待，延时等待

编程格式：WAIT DI/DO/TIME=<value>

注释：DI 或者 DO；<value>为 ON 或 OFF。选择 TIME 时，<value>为 ms。当配置的 IO 与指定的 STATE 状态相同或者等待时间满足时，则执行后续程序。

4.5 坐标系、寄存器及其处理指令

4.5.1 坐标系指令

坐标系指令用于改变机器人当前工作所使用的坐标系的设置。指令定义如下：

a) 坐标系选择指令——改变当前选择的坐标系的序号。
   1) 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：UTOOL NUM = (value)
      注释：其中 value 表示 R[i]或 Constant(工具坐标系序号-1 到 15)
   2) 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：UFRAME NUM = (value)
      注释：其中 value 表示 R[i]或 Constant(工件坐标系序号-1 到 15)

4.5.2 常规寄存器指令

指令功能：寄存器指令在寄存器上完成算术运算。寄存器是一个存储数据的变量。

编程格式：

a) 赋值：R[i] /DI[i]/DO[i]/AI[i]/AO[i]/TIMER[i] = (value)
   注释：上述指令把数值右端(value)赋值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(constant)、寄存器，但针对 PR[i]赋值要注意类型匹配。
b) 算术运算后赋值：R[i] /DI[i]/DO[i]/AI[i]/AO[i]/TIMER[i] = (value) (operator) (value)
   注释：i 表示寄存器的序号，i 的范围根据不同寄存器定义不同；value 可取常数(constant)、寄

存器；操作符(operator)可取“+”“-”“*”“/”“MOD”“DIV”,分别表示把两个值的和、差、乘积、商、商的整数部分、商的余数部分赋值给指定的寄存器。

4.5.3 位置寄存器(PR)指令

指令功能：位置寄存器指令在位置寄存器上完成算术操作。位置寄存器指令可以把位置数据、两个数值的和、差赋值给指定的位置寄存器。位置寄存器是一个存储位置数据(X、Y、Z、A、B、C)的变量。

编程格式：

a) PR[i] = (value)

注释：PR[i] = (value)指令把数值(value)赋值给指定的位置寄存器。其中，i 的范围是 0 到 999。

b) PR[i] = (value) (operator) (value)

注释：PR[i]、(value)的含义同上。操作符(operator)可取“+”“-”，分别表示把两个数值的和、差分别赋值给指定的位置寄存器。

4.5.4 位置寄存器轴指令

指令功能：位置寄存器轴指令在位置寄存器上完成计算操作。

编程格式：

a) 赋值：PR[i,j] = (value)

注释：PR[i,j]=(value)指令把数值(value)赋值给指定的位置寄存器元素。其中，PR[i,j]中的元素 i 代表位置寄存器的序号，j 代表位置寄存器元素序号。(value)可以取常数(constant)、寄存器(R)、位置寄存器中的某个轴(PR[i,j])、位置变量中的某个轴(P[i,j])、数字量输入/输出(DI[i]/DO[i])、模拟量输入/输出(AI[i]/AO[i])。

b) 算术运算后赋值：PR[i,j] = (value) (operator) (value)

注释：指令把两个数值的整数部分赋值给指定的位置寄存器。PR[i,j]中的元素 i 代表位置寄存器的序号，j 代表位置寄存器元素序号;操作符(operator)可取“+”“-”“*”“/”“MOD”“DIV”，分别表示将两个数据的和、差、乘积、商、商的整数部分、商的余数部分等赋值给指定的位置寄存器元素。直角坐标系下，取值为：0=X,1=Y,2=Z,3=A,4=B,5=C;关节坐标系下，取值为：0=J1,1=J2,2=J3,3=J4,4=J5,5=J6。

4.5.5 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)

注释：value 的取值为 1%~100%。

4.5.6 TIMER 指令

指令功能：计时功能。

编程格式：

a) 计时操作:TIMER[i] =(value)

i 的取值范围为 0~99，分别对应每一个计时器。value 取值范围为 START/STOP/RESET，分别对应着启动，停止，复位。

b) 赋值操作:R[i] = TIMER[i]

将 i 计时器当前值赋值给左端寄存器。i 的范围根据不同寄存器定义不同。

4.5.7 位置补偿指令

指令功能：通过此指令可以将原有的点进行偏移，偏移量由位置寄存器决定。

编程格式：

a) OFFSET CONDITION PR[i]

该指令后续所有运动指令的点位，都按照 PR[i]寄存器进行偏移，除非有下条 OFFSET 指令生效。

b) OFFSET,PR[i]

此指令不可单独使用，跟随在运动语句后，此时该运动语句点位按照 PR[i]进行偏移。

注：a,b 两种语句冲突时，b 有效。

4.6 运算指令

4.6.1 概述

运算指令是对程序中相关数据进行算数运算或逻辑运算的指令，分为算数运算指令和逻辑运算指令。

4.6.2 算数运算指令

4.6.2.1 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。

4.6.2.2 SIN 指令

指令功能：求给定角度的正弦值
编程格式：SIN(<expression>)
注释：<expression>的单位是角度

4.6.2.3 ASIN 指令

指令功能：求给定值的反正弦值
编程格式：ASIN(<expression>)
注释：<expression>的取值范围是-1 到+1。

4.6.2.4 COS 指令

指令功能：求给定角度的余弦值
编程格式：COS(<expression>)
注释：<expression>的单位是角度

4.6.2.5 ACOS 指令

指令功能：求给定值的反余弦值
编程格式：ACOS(<expression>)
注释：<expression>的取值范围是-1 到+1。

4.6.2.6 ATAN2 指令

指令功能：求欧拉角度
编程格式：ATAN2(<expression1>/<expression2>)
注释：函数返回值在$-p$ 到$+p$ 之间。通过<expression1>和<expression2>联合确定欧拉角度，使得所得角度值唯一。

4.6.3 逻辑运算指令

4.6.3.1 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；
b) OR；
c) NOT；
d) OXR；
e) ONXR。

4.6.3.2 AND 指令

指令功能：求两个数据的逻辑与
编程格式：<data1> AND <data2>
注释:data1、data2 表示两个不同的数据寄存器。计算 data1、data2 的逻辑与。

4.6.3.3 OR 指令

指令功能：求两个数据的逻辑或
编程格式：<data1> OR <data2>
注释:data1、data2 表示两个不同的数据寄存器。计算 data1、data2 的逻辑或。

4.6.3.4 NOT 指令

指令功能：求数据的逻辑非
编程格式：NOT <data>
注释:data 表示数据寄存器。计算 data 的逻辑非。

4.6.3.5 OXR 指令

指令功能：求两个数据的逻辑异或

编程格式：<data1> OXR <data2>

注释:data1、data2 表示两个不同的数据寄存器。计算 data1、data2 的逻辑异或。

4.6.3.6 ONXR 指令

指令功能：求两个数据的逻辑同或

编程格式：<data1> ONXR <data2>

注释:data1、data2 表示两个不同的数据寄存器。计算 data1、data2 的逻辑同或。

4.7 程序框架指令

4.7.1 概述

程序框架指令是整个程序的框架结构，其一般由示教系统自动完成。程序框架指令包括以下几种：

a) <attr> <end>；
b) <pos> <end>；
c) <program> <end>；

4.7.2 attr 指令

指令功能：指定程序适用的轴组编号。

编程格式：

<attr>
GROUP:[i]
<end>

注释:i 的取值为 0~最大轴组数，适用多个轴组中间用逗号隔开。

4.7.3 pos 指令

指令功能：存储该程序中使用的局部 P 点位。

编程格式：

<pos>
P[i]{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[i]{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]
<end>

注释：

a) i 的取值范围为 0~999。
b) GP 为轴组信息。
c) UF 为工件号信息。
d) UT 为工具号信息。
e) JNT 为关节位置。
f) CFG 为机器臂形态标志位
g) LOC 为空间位置。

4.7.4 program 指令

指令功能：用户程序正文

编程格式：

<program>
(User program)
<end>

附 录 A
（资料性附录）
典型编程程序格式框架

```
<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]};
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>
```