



GSK工业机器人

助您实现“工业自动化”的梦想

GSK机器人编程 Robot Programming

简介:

GSK工业机器人是广州数控设备有限公司自主研发生产，具有独立知识产权的最新产品。它采用国内最先进的GR-C机器人控制器。

GR-C系列工业机器人由六关节串联机器人本体、控制柜和示教盒三部分通过缆线连接而成。

那么，示教编程主要通过手持示教盒来进行编程的。



目 录

- 1.1 编程基本概念
- 1.2 机器人坐标系
- 1.3 插补方式
- 1.4 机器人变量
- 1.5 机器人基本指令
- 1.6 机器人示教
- 1.7 机器人再现



GSK机器人编程 Robot Programming

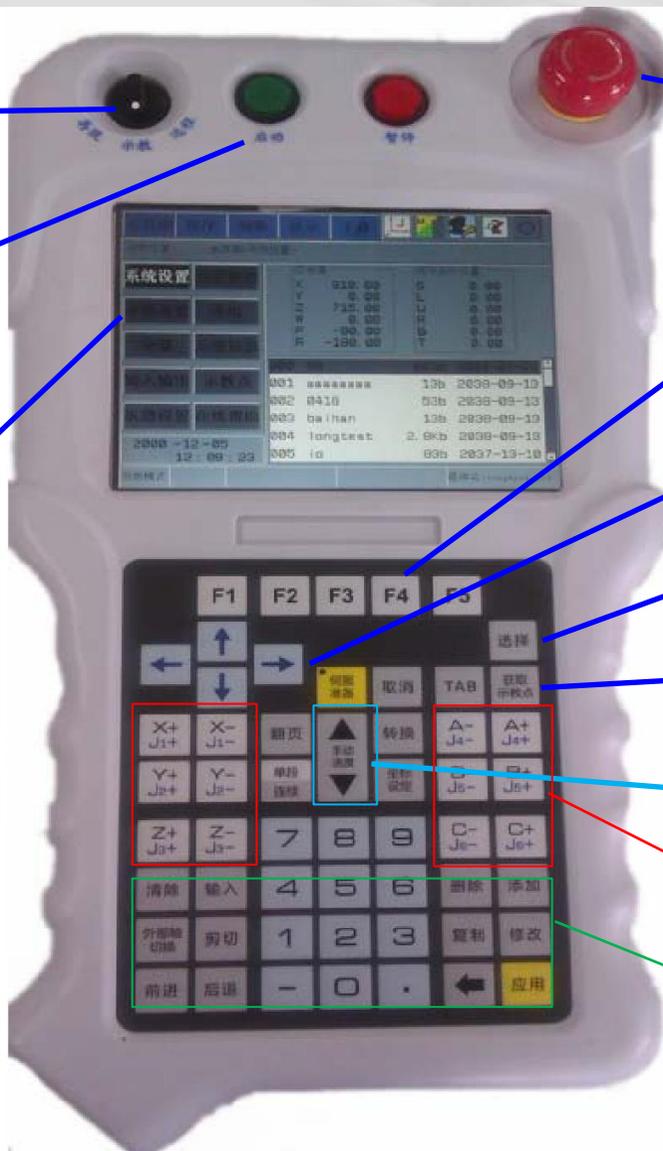
模式开关：示教-再现-远程

启动按钮（绿）：启动程序
暂停按钮（红）：暂停程序

主页面：快捷菜单、状态显示区、主菜单、坐标值显示区、程序目录、人机交互窗

使能开关：（示教盒背面）

示教盒
简介



急停按钮：紧急停止

快捷键：
F1、F2、F3、F4、F5

方向键：移动光标方向

选择键：确认选取目标

获取示教点键：

手动速度键：调节速度

轴操作键：点动控制机器人

编辑键：编辑程序

培训师：何树洋

1.1 编程基本概念

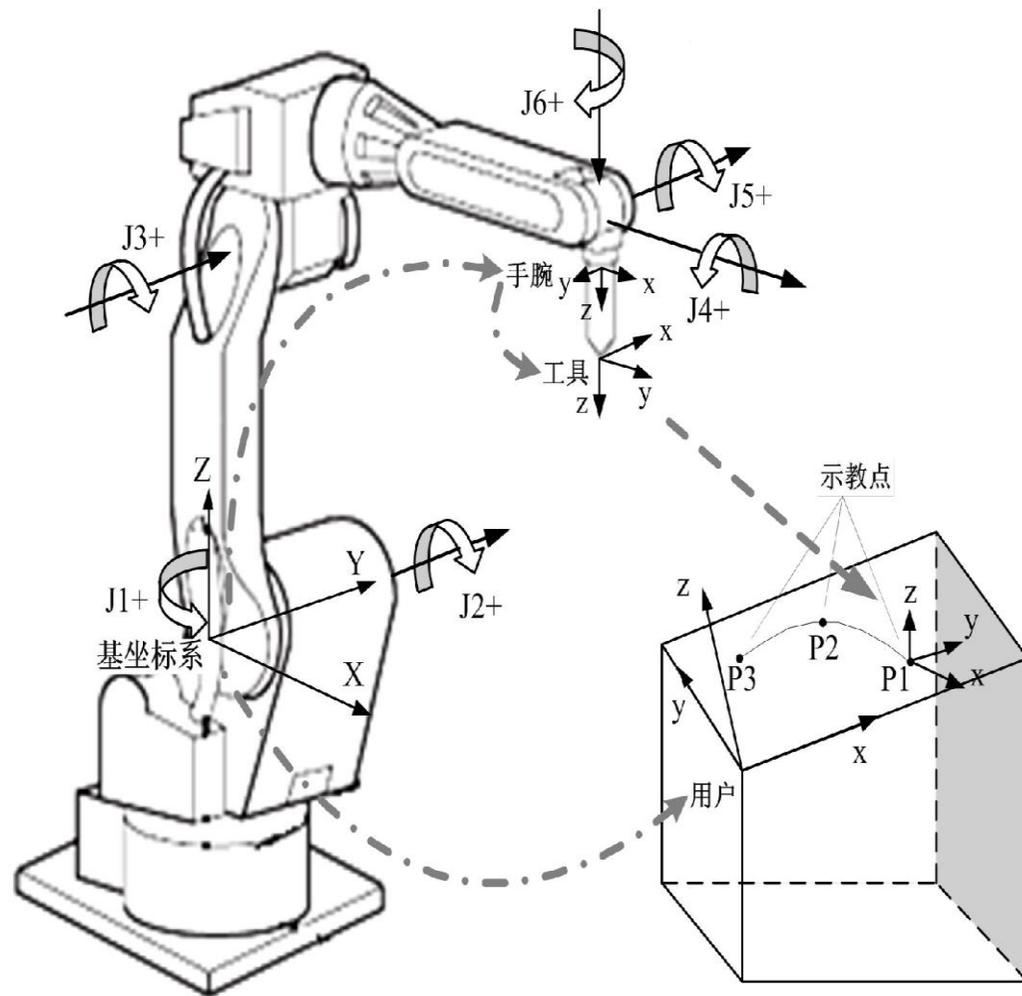
示教点：是指笛卡尔空间中的某个位置点。GR-C系统用坐标 (X, Y, Z, W, P, R) 表示一个示教点，其中 X, Y, Z 是指该位置点在笛卡尔坐标系中的具体位置值， W, P, R 是指机器人TCP端点在该位置时的方位，也称为姿态。因此，一个示教点确定了机器人TCP端点在笛卡尔空间中的位置和姿态。

运行模式：

- 示教模式：**手动移动机器人或示教、编写、修改运行程序或者进行各种参数设置和文件操作，但必须获得相应的权限；
- 再现模式：**机器人执行用户程序，完成各种预定动作和任务的过程，可选择运行程序，查看各种监控信息，但不能执行程序的编辑或系统参数设置的操作；
- 远程模式：**可通过外部输入信号指定进行接通伺服电源、启动、暂停、急停、调出主程序等有关操作。

1.2 机器人坐标系

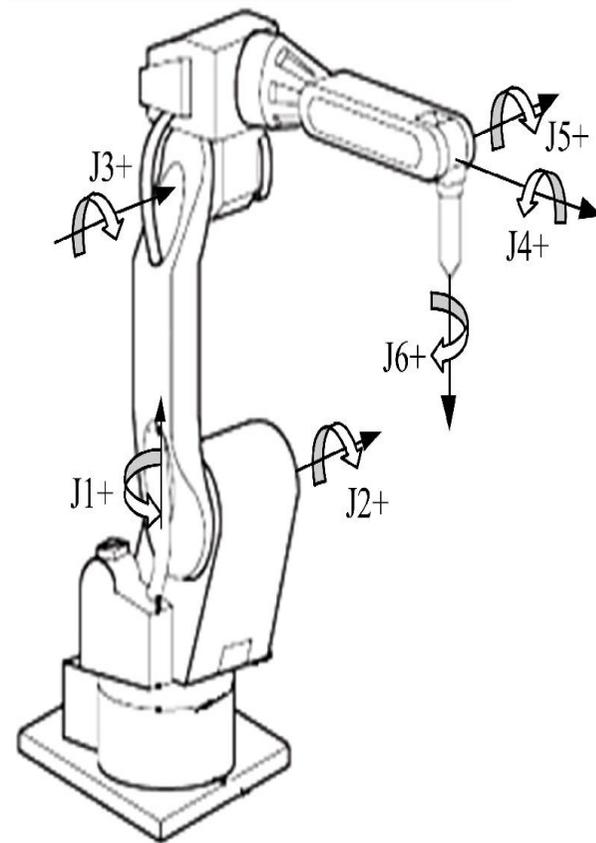
- J-关节坐标系
- B-基坐标系
- T-工具坐标系
- U-用户坐标系



J-关节坐标系:



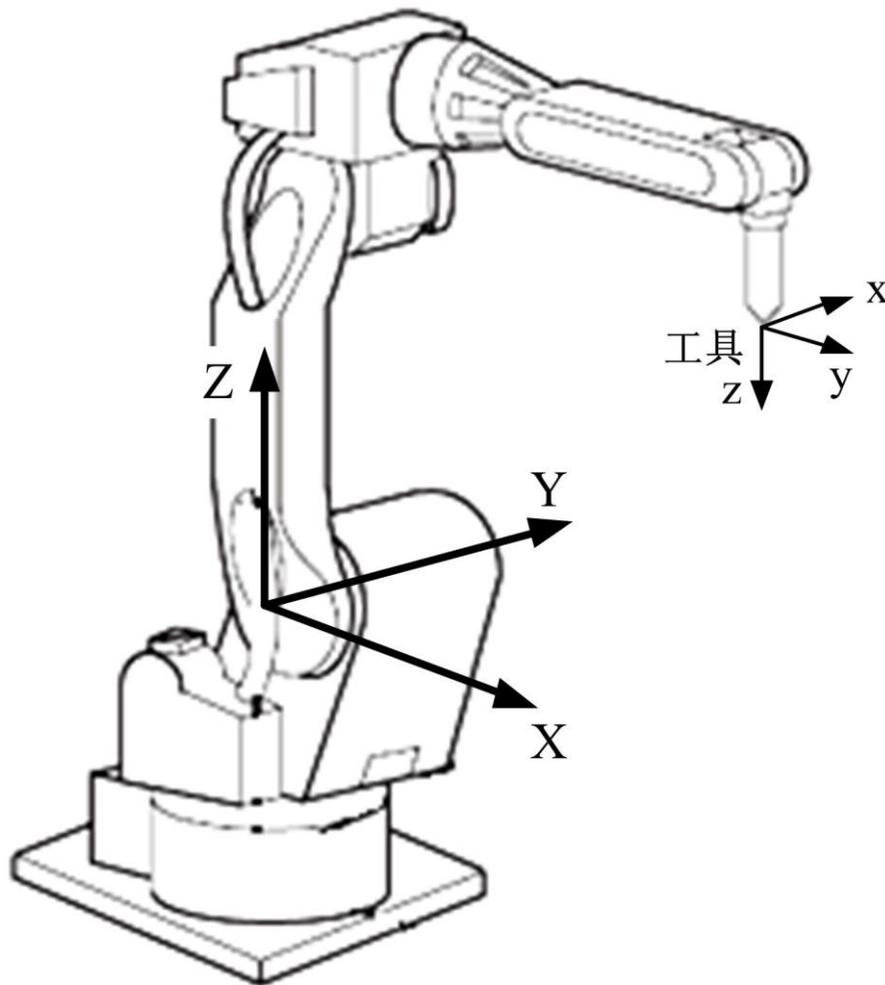
将[模式选择]键选择“示教”模式，通过[坐标设定]键，切换系统的动作坐标系为关节坐标系，按下[使能开关]键，通过轴操作键，可使得机器人在各个轴的轴方向转动。



B-基坐标系:



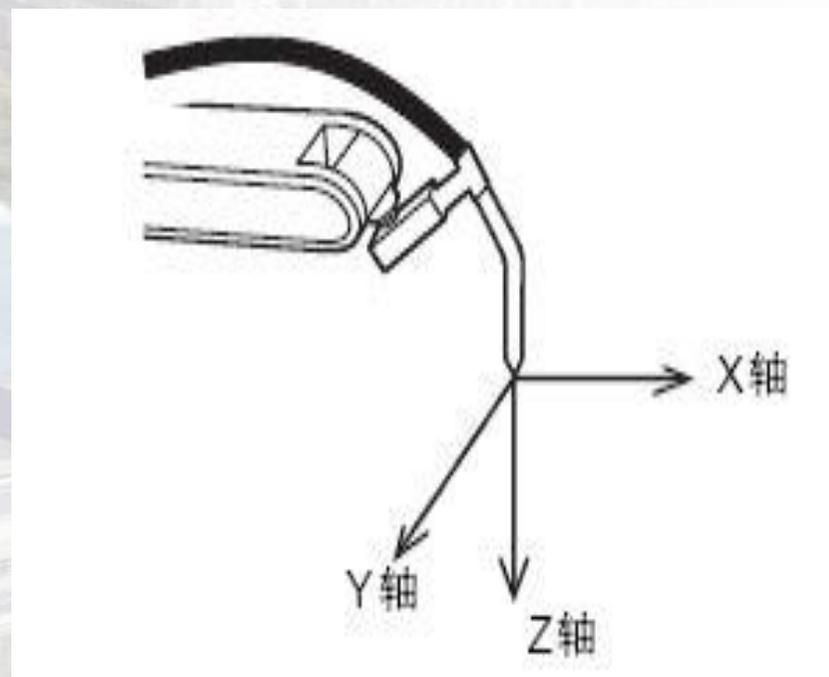
将[模式选择]键选择“示教”模式，通过[坐标设定]键，切换系统的动作坐标系为基坐标(也称直角坐标系)，按下[使能开关]键，通过轴操作键，可使得机器人控制端点TCP 可沿X、Y、Z轴方向平行移动或绕相应工具坐标系轴向旋转（即姿态调整）。



T-工具坐标系:



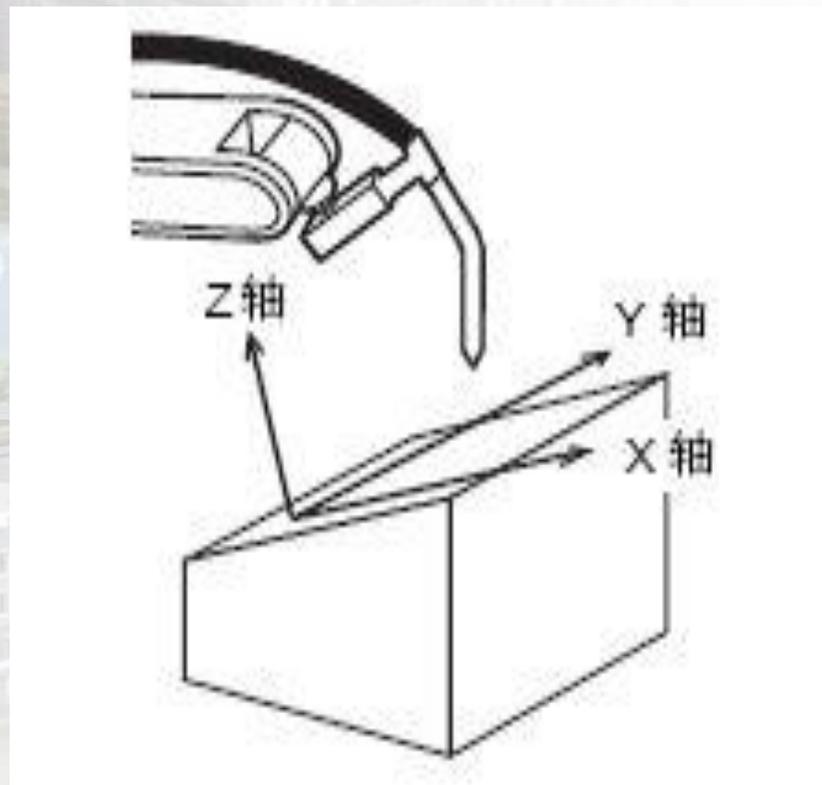
把机器人腕部法兰盘所持工具的有效方向作为Z轴，并把坐标系原点定义在工具的尖端点。在工具坐标系未定义时，系统自动采用默认的工具。当机器人跟踪笛卡尔空间某路径时，必须正确定义工具坐标系。在机器人示教移动过程中，若所选坐标系为工具坐标系，则机器人将沿工具坐标系坐标轴方向移动或者绕坐标轴旋转。当绕坐标轴旋转时工具坐标系的原点位置将保持不变，这叫做**重定位运动**操作（姿态调整）。在直角坐标系及用户坐标系中也可实现类似的动作。



U-用户坐标系:



将[模式选择]键选择“示教”模式，通过[坐标设定]键，切换系统的动作坐标系为用户坐标系，按下[使能开关]键，通过轴操作键，可使得机器人控制端点TCP可沿用户坐标系X、Y、Z轴方向平行移动或绕相应工具坐标系轴向旋转（即姿态调整）。与直角坐标系下的轴操作键类似，但参考的坐标系不同。



1.3 插补方式

插补方式表示机器人在已知示教点之间运动控制点的**运动轨迹**。

GR-C 系列机器人主要有三种插补方式：

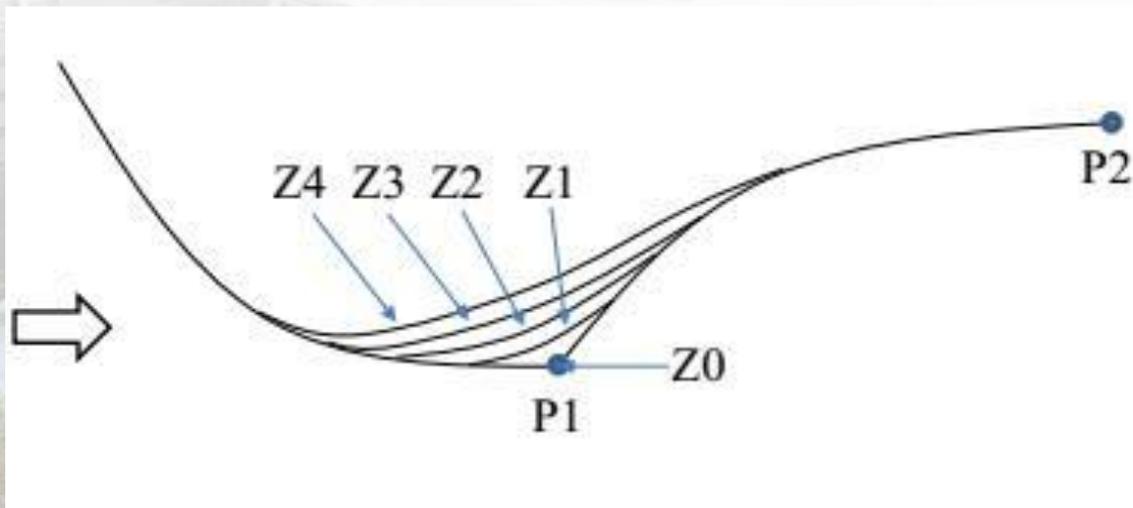
关节插补：机器人在未规定采用何种轨迹运动时，使用关节插补，指令为 **MOVJ**。出于安全考虑，程序的第一个运动点应采用关节插补；

直线插补：直线插补表示机器人末端 TCP 跟踪指定的直线路径，指令为 **MOVL**；

圆弧插补：圆弧插补表示两点间沿圆弧的点群来逼近，指令为 **MOVC**。
机器人在运动的过程中自动改变手腕的位置。

MOVJ指令及轨迹示意如下：

```
MOVJ P1, V20, Z0;  
MOVJ P2, V20, Z0;
```

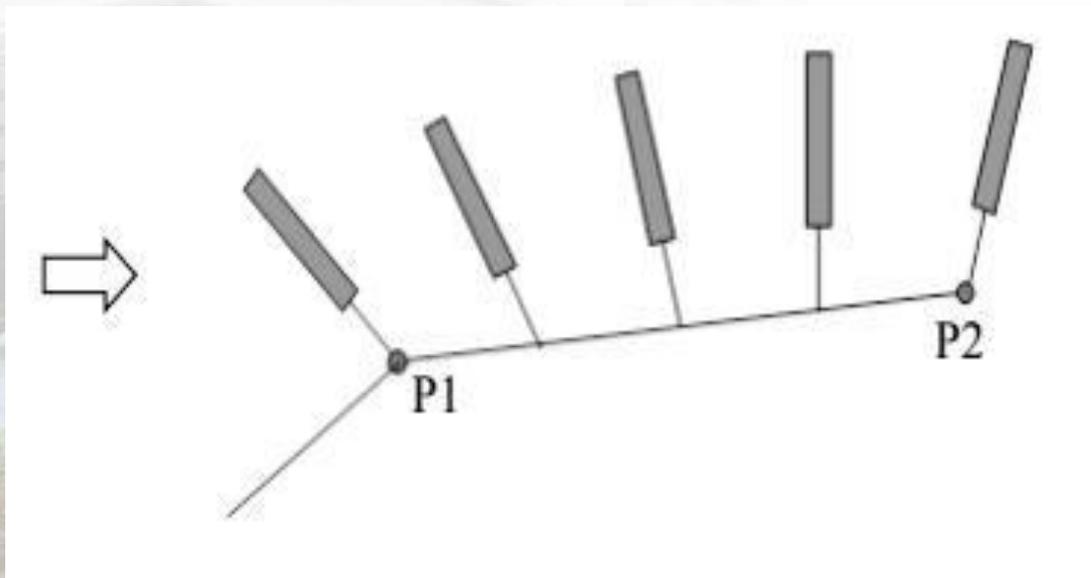


其中：P*为示教点，V*表示运行速度，以百分比来表示。Z*表示精度等级，取值从 0 到 4，精度等级越高，到位精度越低。

Z0 表示精确到达目标点，Z 值越大，过渡半径越大，机器人的运行效率越高。

MOVL指令及轨迹示意如下：

```
MOVL P1, V100, Z0;  
MOVL P2, V200, Z0;
```



其中：P*为示教点，V*表示运行速度，单位为 mm/s。Z*表示精度等级，取值从 0 到 4，意义及取值同关节插补中的 Z*。

Z0 表示精确到达目标点，Z 值越大，过渡半径越大，机器人的运行效率越高。

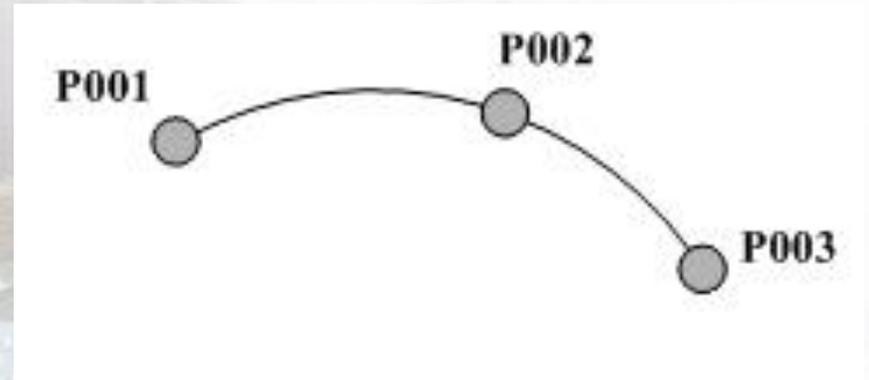
MOVC指令及轨迹示意如下：

```
MOVL P1, V100, Z0;  
MOVC P1, V200, Z0;  
MOVC P2, V200, Z0;  
MOVC P3, V200, Z0;
```

确定一个圆弧需要连续三个点，少于三个点就会报警，执行过程中，机器人把第一个 MOVC 作为起始点。

注意事项：

1. 示教一段圆弧的角度不得大于 180 度；
2. 在圆弧指令的示教点选取上，应避免相邻两点距离太近，否则圆弧的轨迹精度将会下降；
3. 当圆弧的起点与上一条运动指令的目标点不重合时，系统将以直线运动方式从该目标点运动到圆弧起点。



1.4 机器人系统变量

- 全局变量：** 字节型变量(B)
 整数型变量(I)
 双精度型变量(D)
 实数型变量(R)
 笛卡尔位姿变量(PX)

注：所有程序文件共享这些变量，使用系统全局变量前需要给变量初始化。

1.5 机器人基本指令

运动指令：MOVJ、MOVL、MOVC

信号处理指令：DOUT、WAIT、DELAY、PULSE

流程控制指令：LAB、JUMP、CALL、RET、#、MAIN、END

运算指令：INC、DEC、ADD、SUB、MUL、DIV、SET、SETE

平移指令：PX、SHIFTON、SHIFTOFF

其他指令：关系操作符（==、>、<、>=、<=、<>、）

运算操作符（=、+、-）

MOVJ运动指令:

功能: 以点到点 (PTP) 方式移动到指定位姿。

格式: MOVJ P<位姿变量名>, V<速度>, Z<精度>; (MOVJ P1, V20, Z0;)

- 参数:**
- 1、位姿变量名指定机器人的目标姿态，为示教点号，系统添加该指令默认为“P*”，可以编辑 P 示教点号，范围为 P0~P999。
 - 2、V<速度>指定机器人的运动速度，这里的运动速度是指与机器人设定的最大速度的百分比，取值范围为 1-100(%)。
 - 3、Z<精度>指定机器人的精确到位情况，这里的精度表示精度等级，取值范围为 0-4。

- 说明:**
- 1、当执行 MOVJ 指令时，机器人以关节插补方式移动；
 - 2、移动时，机器人从起始位姿到结束位姿的整个运动过程中，各关节移动的行程相对于总行程的比例是相等的；
 - 3、MOVJ 指令的精度等级 Z0 表示精确到位，Z1~4 表示关节过渡。

MOVL运动指令:

功能: 以直线插补方式移动到指定姿态。

格式: MOVL P<位姿变量名>, V<速度>, Z<精度>;

- 参数:**
- 1、位姿变量名指定机器人的目标姿态，为示教点号，系统添加该指令默认为“P*”，可以编辑 P 示教点号，范围为 P0~P999。
 - 2、V<速度>指定机器人的运动速度，这里的运动速度是指机器人实际运行线速度，取值范围为 0-4000mm/s，为整数。
 - 3、Z<精度>指定机器人的精确到位情况，这里的精度表示精度等级，取值范围为 0-4。

MOV C运动指令:

功能: 以圆弧插补方式移动到指定位姿。

格式: MOV C P<位姿变量名>, V<速度>, Z<精度>;

参数:

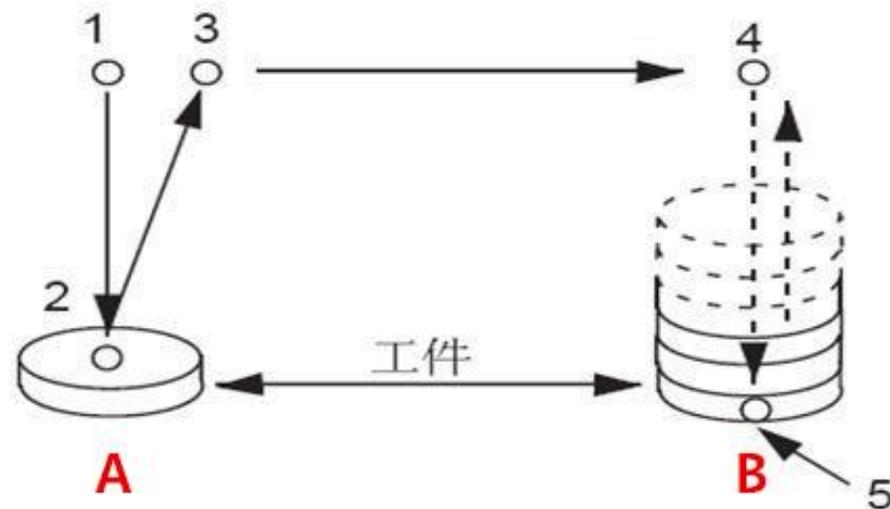
- 1、位姿变量名指定机器人的目标姿态，为示教点号，系统添加该指令默认为“P*”，可以编辑 P 示教点号，范围为 P0~P999。
- 2、V<速度>指定机器人的运动速度，这里的运动速度是指机器人实际运行线速度，取值范围为 0-4000mm/s，为整数。
- 3、Z<精度>指定机器人的精确到位情况，这里的精度表示精度等级，取值范围为 0-4。

说明:

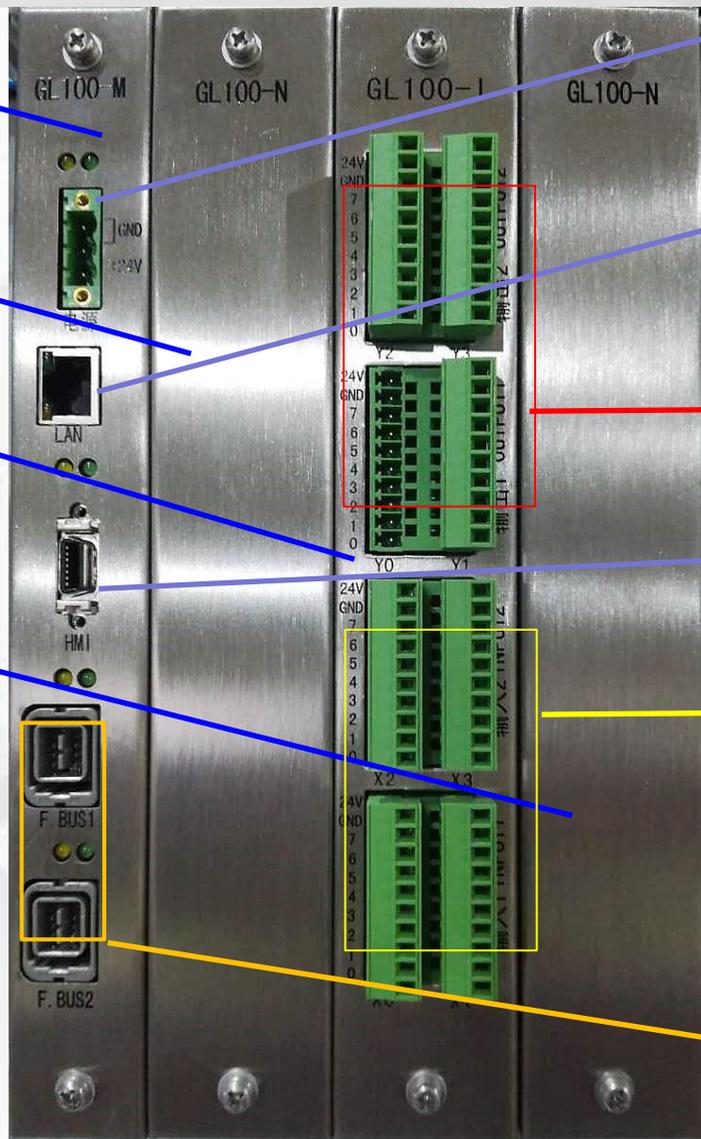
- 1、当执行 MOV C 指令时，机器人以圆弧插补方式移动。
- 2、三点或以上确定一条圆弧，小于三点系统报警。
- 3、直线和圆弧之间、圆弧和圆弧之间都可以过渡，即精度等级 Z 可为 0~4。

轨迹编程案例 1 :

假设 A 处水平面方向叠放着 1 个厚 3.2 mm、直径 8.0 mm 同规格的圆盘物料，需要机器人将物料从 A 处搬运到 B 处，请思考用运动指令简易编写出相应的轨迹程序。



GSK机器人编程Robot Programming



系统运动控制主板

系统功能拓展板

系统I/O板

系统I/O拓展板

GR-C
系统简介

系统24V电源输入端口

以太网通讯端口

数字IO输出端口: Y0、Y1、Y2、Y3

HMI示教盒线缆通讯端口

数字IO输入端口: X0、X1、X2、X3

F.BUS伺服驱动器通讯端口

DOUT 输出指令:

功能: 数字信号输出 I/O 置位指令。

格式: DOUT OT<输出端口>, <ON/OFF>;

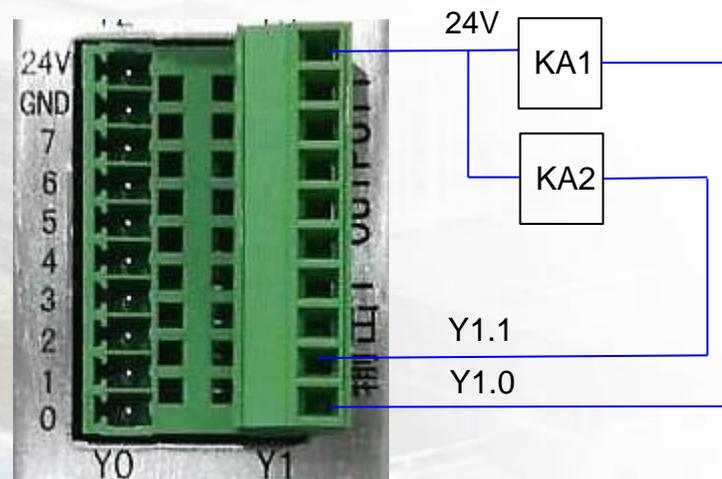
参数: 1、<输出端口>指定需要设置的 I/O 端口, 范围为 0-31, 视 I/O 扩展 板卡数量而定。

2、<ON/OFF >设置为 ON 时, 相应 I/O 置 1, 即高电平; 设置为 OFF 时, 相应 I/O置 0, 即低电平。

说明: 1. 输出端口分为4组: Y0、Y1、Y2、Y3;

2. 每组从底部分为10个接线口: 0-7, GND, 24V;

3. 输出为低电平有效, 输出0V;



示例:

MAIN;

MOVJ P1, V30, Z0;

DOUT OT0, OFF;

DOUT OT1, ON;

MOVL P2, V30, Z0;

DOUT OT1, OFF;

MOVL P3, V30, Z0;

END;

PULSE 脉冲输出指令:

功能: 输出一定宽度的脉冲信号，作为外部输出信号。

格式: PULSE OT<输出端口>, T<时间 (sec)>;

参数: 1、OT<输出端口>, 与DOUT的OT<输出端口>一样，标配IO端口范围为0-31;
2、T<时间 (sec)>指脉冲时间宽度，单位为秒，视范围为00-900.0 (s)。

示例:

```
MAIN;
```

```
MOVJ P1, V30, Z0;
```

```
PULSE OT1, T, 0.5;
```

```
MOVL P2, V30, Z0;
```

```
DOUT OT1, ON ;
```

```
MOVL P3, V30, Z0;
```

```
END;
```

WAIT 输入指令:

功能: 等待直到外部输入信号的状态符合指定的值。

格式: WAIT IN<输入端口>, <ON/OFF>, T<时间>;

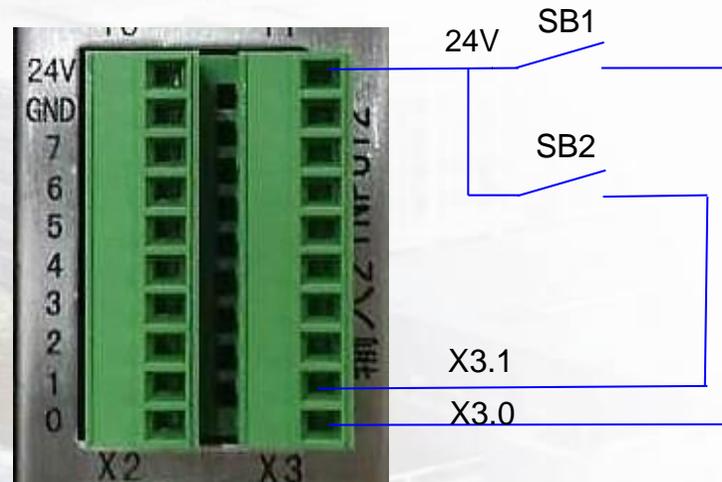
参数: 1、<输入端口>指定相应的输入端口, 范围为 0-31, 视 IO 扩展板卡数量而定。

2、<ON/OFF >为 ON 时, 有输入信号, 即高电平; 为 OFF 时, 无输入信号, 即低电平。

说明: 1. 输入端口分为4组: X0、X1、X2、X3;

2. 每组从底部分为10个接线口: 0-7, GND, 24V;

3. 输入为高电平有效, 输入24V;



示例:

```
MAIN;  
MOVJ P1, V30, Z0;  
WAIT IN24, ON, T0;  
MOVL P2, V30, Z0;  
WAIT IN24, OFF, T0;  
WAIT IN24, ON, T3;  
MOVL P3, V30, Z0;  
END;
```

培训师: 何树洋

DELAY 延时指令:

功能: 使机器人延时运行指定时间。

格式: DELAY T<时间 (sec) >;

参数: 1、T<时间 (sec) >指定延时时间, 单位为秒, 视范围为00-900.0 (s)。

示例:

```
MAIN;
```

```
MOVJ P1, V30, Z0;
```

```
DELAY T5.6;
```

```
MOVL P2, V30, Z0;
```

```
DELAY T0.5;
```

```
MOVL P3, V30, Z0;
```

```
END;
```

MAIN 程序开始指令:

功能: 程序开始（系统默认行）。

格式: MAIN;

说明: 程序默认行数，不可以对其编辑，宣布程序开始。

END 程序结束指令:

功能: 程序结束（系统默认行）。

格式: END;

说明: 程序运行到程序段 END 时停止示教或再现运行状态，其后面有程序不被执行。

示例:

```
MAIN;  
MOVJ P1, V30, Z0;  
MOVL P2, V30, Z0;  
MOVL P3, V30, Z0;  
MOVC P4, V30, Z0;  
MOVC P5, V30, Z0;  
END;  
MOVJ P6, V30, Z0;  
MOVJ P7, V30, Z0;  
END;
```

LAB 标签指令:

功能: 标明要跳转到的语句。

格式: LAB<标签号>:

参数: 1、<标签号>与JUMP跳转指令配合使用，同一个程序文件里标签号不允许重复。

示例:

MAIN;

LAB1:

MOVJ P1, V30, Z0;

LAB2:

MOVL P2, V30, Z0;

MOVL P3, V30, Z0;

JUMP LAB1;

END;

JUMP 跳转指令:

功能: 跳转到指定标签。

格式: JUMP LAB<标签号>:

JUMP LAB<标签号>, IF<变量><比较符><变量/常量>;

JUMP LAB<标签号>, IF IN<输入端口>==<ON/OFF>;

参数: 1、<变量>可以是 B<变量号>, I<变量号>, D<变量号>, R<变量号>, 变量号的范围为 0-99;

2、<比较符>指定比较方式, 包括==、>=、<=、>、<和<>;

3、IN<输入端口> 指定需要比较的输入端口, 标配IO端口取值范围为0-31;

说明: 1、JUMP 指令必须与 LAB 指令配合使用, 否则程序报错“匹配错误: 找不到对应的标签”;

2、当执行 JUMP 语句时, 如果不指定条件, 则直接跳转到指定标号; 若指定条件, 则需要符合相应条件后跳转到指定标号, 如果不符合相应条件则直接运行下一条语句。

3、比较符中的“<>”表示“不等于”。

示例:

MAIN;	程序开始;
LAB1:	标签1
SET R1, 0;	赋值实数型R1变量值为0
LAB2:	标签2
MOVJ P1, V30, Z0;	关节运动到示教点 P1位置
MOVL P2, V30, Z0;	直线运动到示教点 P2位置
INC R1;	实数型R1变量值加一
JUMP LAB2, IF R1 < 5;	若R1值<5, 则跳转到LAB2, 否则运行下一行程序;
MOVL P3, V30, Z0;	直线运动到示教点 P3位置
MOVC P4, V30, Z0;	圆弧运动到示教点 P4位置
MOVC P5, V30, Z0;	圆弧运动到示教点 P5位置
JUMP LAB2, IF IN1 ==ON	若IN1端口有输入信号则跳转到LAB2, 否则运行下一行程序;
;	关节运动到示教点 P6位置
MOVJ P6, V30, Z0;	直接跳转到LAB1
JUMP LAB1;	程序结束
END;	

CALL 调用子程序指令：

功能：调用指定程序。

格式：CALL JOB;

参数：JOB 程序文件名称，需指定程序，如CALL GSK01;

说明：1、CALL 指令必须与 RET 指令配合使用，否则程序报错，无法返回主程序；
2、调用指定的程序，最多 8 层，不能嵌套调用；

RET 调用子程序返回指令：

功能：子程序调用返回。

格式：RET;

说明：在每个子程序运行结束前添加RET指令返回主程序，否则程序报错，无法返回主程序；

示例:

(主程序: GSK00)

```
MAIN ;  
MOVJ P1 , V100 , Z0 ;  
MOVJ P2 , V100 , Z0 ;  
MOVJ P3 , V100 , Z0 ;  
CALL GSK01 ;  
CALL GSK02 ;  
MOVJ P3 , V100 , Z0 ;  
MOVJ P2 , V100 , Z0 ;  
MOVJ P1 , V100 , Z0 ;  
END ;
```

(子程序1: GSK01)

```
MAIN ;  
MOVJ P1 , V100 , Z0 ;  
MOVJ P2 , V100 , Z0 ;  
MOVL P3 , V100 , Z0 ;  
MOVC P3 , V100 , Z0 ;  
MOVC P4 , V100 , Z0 ;  
MOVC P5 , V100 , Z0 ;  
MOVJ P1 , V100 , Z0 ;  
RET ;  
END ;
```

(子程序2: GSK02)

```
MAIN ;  
MOVJ P1 , V100 , Z0 ;  
MOVJ P2 , V100 , Z0 ;  
MOVL P3 , V100 , Z0 ;  
MOVL P4 , V100 , Z0 ;  
MOVL P5 , V100 , Z0 ;  
MOVL P6 , V100 , Z0 ;  
MOVJ P1 , V100 , Z0 ;  
RET ;  
END ;
```

注释指令:

功能: 注释语句。

格式: # <注释语句>;

说明: 1、前面添加“#”指令，不执行该程序行。
2、对已经被注释的指令进行注释，则可取消该指令的注释状态，即反注释。

示例:

```
MAIN;  
# MOVJ P1, V30, Z0;  
MOVL P2, V30, Z0;  
# WAIT IN24, ON, T0;  
MOVL P3, V30, Z0;  
END;
```

SET 算术指令:

功能: 把操作数 2 的值赋给操作数 1。

格式: SET <操作数1>, <操作数2>;

- 参数:**
1. <操作数 1> 可以是 B<变量号>, I<变量号>, D<变量号>, R<变量号>。
 2. <操作数 2>可以是常量, B<变量号>, I<变量号>, D<变量号>, R<变量号>。
 3. 变量号的范围为 0~99。

示例:

```
MAIN;  
SET B0, 5;  
SET B1, B0;  
SET R1, 2.3;  
SET R2, R1;  
END;
```

INC 算术指令:

功能: 在指定操作数的值上加 1。

格式: INC <操作数>;

参数: <操作数> 可以是全局变量: B<变量号>
, I<变量号>, D<变量号>, R<变量号>,
也可是局部变量: LB<变量号>, LI<变量号>,
LD<变量号>, LR<变量号>。

变量号的范围为 0~99 。

示例:

```
MAIN;  
LAB1:  
SET R1, 0;  
LAB2:  
MOVJ P1, V60, Z0;  
MOVJ P2, V60, Z0;  
INC R1;  
JUMP LAB2, IF R1<=5;  
END;
```

DEC 算术指令:

功能: 在指定操作数的值上减 1。

格式: DEC <操作数>;

参数: <操作数> 可以是全局变量: B<变量号>
, I<变量号>, D<变量号>, R<变量号>,
也可是局部变量: LB<变量号>, LI<变量号>,
LD<变量号>, LR<变量号>。

变量号的范围为 0~99 。

示例:

```
MAIN;  
LAB1:  
SET R1, 6;  
LAB2:  
MOVJ P1, V60, Z0;  
MOVJ P2, V60, Z0;  
DEC R1;  
JUMP LAB1, IF R1>=0  
;  
END;
```

ADD 算术指令:

功能: 把操作数 1 与操作数 2 相加, 结果存入操作数 1 中。

格式: ADD <操作数1>, <操作数2>;

- 参数:**
1. <操作数 1> 可以是 B<变量号>, I<变量号>, D<变量号>, R<变量号>。
 2. <操作数 2>可以是常量, B<变量号>, I<变量号>, D<变量号>, R<变量号>。
 3. 变量号的范围为 0~99。

示例:

```
MAIN;  
SET R0 , 5;  
SET R1 , 2;  
ADD R0, R1;  
END;
```

此时 R0 的值为 7。

SUB 算术指令:

功能: 把操作数 1 与操作数 2 相减, 结果存入操作数 1 中。

格式: SUB <操作数1>, <操作数2>;

- 参数:**
1. <操作数 1> 可以是 B<变量号>, I<变量号>, D<变量号>, R<变量号>。
 2. <操作数 2>可以是常量, B<变量号>, I<变量号>, D<变量号>, R<变量号>。
 3. 变量号的范围为 0~99。

示例:

```
MAIN;  
SET R0 , 5;  
SET R1 , 2;  
SUB R0, R1;  
END;
```

此时 R0 的值为 3。

MUL 算术指令:

功能: 把操作数 1 与操作数 2 相乘, 结果存入操作数 1 中。

格式: MUL <操作数1>, <操作数2>;

参数: 1. <操作数 1> 可以是 B<变量号>, I<变量号>, D<变量号>, R<变量号>。

2. <操作数 2>可以是常量, B<变量号>, I<变量号>, D<变量号>, R<变量号>。

3. 变量号的范围为 0~99。

示例:

```
MAIN;  
SET R0, 5;  
MUL R0, 2;  
END;
```

此时 R0 的值为 10。

DIV 算术指令:

功能: 把操作数 1 除以操作数 2 , 结果存入操作数 1 中。

格式: DIV <操作数1>, <操作数2>;

参数: 1. <操作数 1> 可以是 B<变量号>, I<变量号>, D<变量号>, R<变量号>。

2. <操作数 2>可以是常量, B<变量号>, I<变量号>, D<变量号>, R<变量号>。

3. 变量号的范围为 0~99。

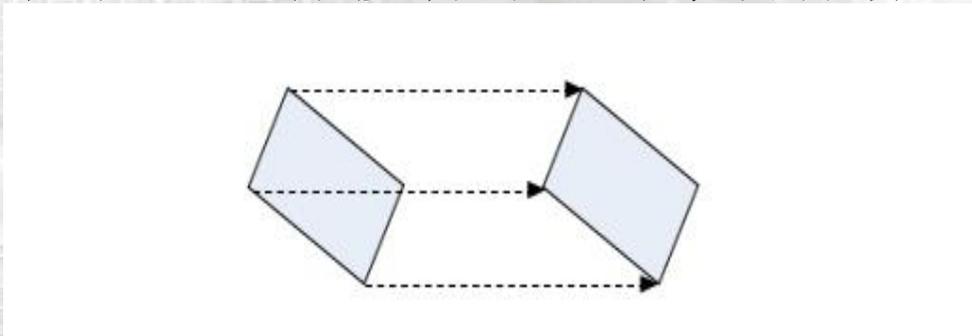
示例:

```
MAIN;  
SET R0, 6;  
DIV R0, 2;  
END;
```

此时 R0 的值为 3。

机器人平移功能：

平移是指对象物体从指定位置进行移动时，对象物体各点均保持等距离移动，如图所示。



机器人进行示教时，可以通过此功能来减少工作量。平移功能特别适用于进行一组有规律的运动时的情况，例如工件的堆垛等。

运用平移功能所用到的指令主要有：PX、SHIFTON、SHIFTOFF 和 MSHIFT。

运用平移前，我们首先要建立一个平移量。建立平移量的方法有两种，一种是进入笛卡尔位姿变量编辑界面手动进行编辑，另一种是采用 MSHIFT 指令来获取偏移量的方式，这里我们采用第一种方式。

进入”笛卡尔位姿型变量明细”界面，对PX0变量做如下图修改，这里我们假设工件的厚度为20mm，“是否使用”设置为1，选择【修改】键保存，这样，我们就可以在程序里使用 PX0 变量。

SETE 算术指令:

功能: 把操作数 2 变量的值赋给笛卡尔位姿变量中的元素。

格式: SETE PX<变量号>(元素号), 操作数 2;

参数: 1. <变量号>范围 0-99。

2. <元素号>范围 0-6。0表示给 PX变量全部元素赋同样的值, 1表示给 PX变量里X赋值, 2表示给 PX变量里Y赋值, 3表示给 PX变量里Z赋值, 4表示给 PX变量里W赋值, 5表示给 PX变量里P赋值, 6表示给 PX变量里R赋值

示例:

3. <操作数 2> 可以是 双精度整数型D<变量号>, 或者是常量。
SET D0, 6;

SETE PX1 (0) , 0; //此时 PX1 变量的 X=0, Y=0, Z=0, W=0, P=0, R=0
。

SETE PX1 (0) , D0; //此时 PX1 变量的 X=6, Y=6, Z=6, W=6, P=6, R=6

。 SETE PX1 (6) , 3; //此时 PX1 变量的 X=6, Y=6, Z=6, W=6, P=6, R=6

PX 平移指令:

功能: 给 PX 变量（笛卡尔位姿变量）赋值，用于平移功能。

格式: PX<变量名> = PX<变量名>;

PX<变量名> = PX<变量名> + PX<变量名>;

PX<变量名> = PX<变量名> - PX<变量名>;

参数: PX<变量名> 指定需要运算的位置变量名，范围为 0-99

。

SHIFTON 平移开始指令:

功能: 指定平移开始及平移量。

格式: SHIFTON PX<变量名>;

参数: PX<变量名>指定平移量, 范围为 0-99。

说明: PX 变量可以在【笛卡尔位姿】菜单界面中设置, 也可以在程序中用指令SETE赋值。

SHIFTOFF 平移结束指令:

功能: 结束平移标识。

格式: SHIFTOFF;

说明: 1. 必须与 SHIFTON 指令配合使用, 否则提示错误 “有重复的平移结束指令”

2. SHIFTOFF 语句后的运动指令不具有平移功能。

示例:

MAIN;

LAB1:

SET R1, 0;

PX1=PX1-PX1;

LAB2:

MOVJ P1, V30, Z0;

SHIFTON PX1;

MOVL P2, V100, Z0;

SHIFTOFF;

INC R1;

PX1=PX1+PX0;

JUMP LAB2, IF R1 < 4;

END;

程序开始

标签1

将R1赋值为0

将PX1变量清零

标签2

关节运动到P1点位置

开始平移

直线运动到P2点位置

平移结束

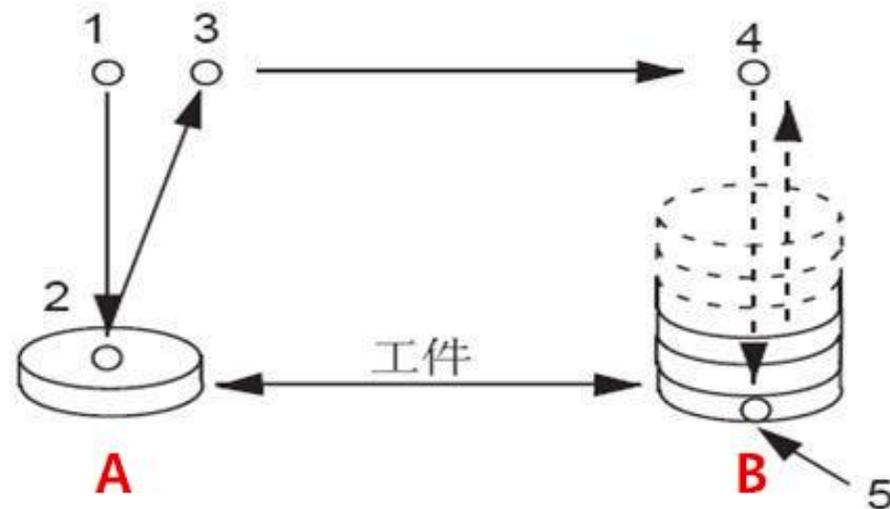
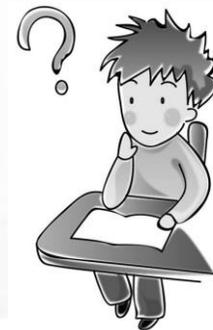
注: 需手动设置PX0的值, 或用STEE指令在程序中赋值

将PX0与PX1相加, 存到PX1里若R1值小于4, 则跳转到LAB2, 否则继续运行下一行程序

程序结束

码垛平移案例2:

假设A处水平面方向叠放着6个厚32mm、直径80mm同规格的圆盘物料，需要机器人将物料从A处搬运到B处，请思考用平移指令简易编写出相应的程序。



感谢聆听 多些指教

广州数控设备有限公司
地址：广州市萝岗区云埔工业区观达路22号

邮编：510530

联系电话：13450296004

网址：<http://www.gsk.com.cn/>

GSK 广州数控
Command the Future_

Exceed the future

专心致力于智能装备产业发展的研究与实践

打造百年企业 铸就金质品牌

Q群567360908



GSK公众号

