

任务 5.2 新建和加载程序







- ▲ 知识要求:掌握常用的机器人指令;掌握机器 人程序的构成特点;掌握机器人的程序编写和编辑 方法。
- ▲ 技能要求:学会编辑程序,如程序的修改、复制、粘贴、删除等;能够实现程序的连续运行和单周运行。



任务 5.2 新建和加载程序

 机器人的程序编辑器中存有程序模板,类似计算机 办公软件的 Word 文档模板,编程时按照模板在里 面添加程序指令语句即可。





一、示教与再现

- "示教"就是机器人学习的过程,在这个过程
 中,操作者要手把手教会机器人做某些动作,
 机器人的控制系统会以程序的形式将其记忆下来。
- ・机器人按照示教时记忆下来的程序展现这些动作,就是"再现"过程。
- 示教再现机器人的工作原理如图 3-1 所示。





图 3-1 示教 - 再现机器人控制方式



什么是 RAPID 程序指令

- RAPID 程序中包含了一连串控制机器人的指令,执行这些指 令可以实现需要的操作。
- 应用程序是使用称为 RAPID 编程语言的特定词汇和语法编 写而成。RAPID 是一种英文编程语言,所包含的指令可以 移动机器人、设置输出、读取输入,还能实现决策、重复其它 指令、构造程序、与系统操作员交流等。

| RAPID 程序 | | | | | | | | | |
|----------|-------|------|------|--|--|--|--|--|--|
| 程序模块1 | 程序模块2 | 程序模块 | 系统模块 | | | | | | |
| | | 3 | | | | | | | |
| 程序数据 | 程序数据 | | 程序数据 | | | | | | |
| 主程序 main | 例行程序 | | 例行程序 | | | | | | |
| 例行程序 | 中断程序 | | 中断程序 | | | | | | |
| 中断程序 | 功能 | | 功能 | | | | | | |
| 功能 | | | | | | | | | |



什么是 RAPID 程序指令

关于 RAPID 程序的架构说明:

RAPID 程序是由程序模块与系统模块组成。一般地,我们 只通过新建程序模块来构建机器人的程序,而系统模块多用于系 统方面的控制之用。

可以根据不同的用途创建多个程序模块,如专门用于主控制的程序模块,用于位置计算的程序模块,用于存放数据的程序模块,这样的目的在于方便归类管理不同用途的例行程序与数据。 每一个程序模块包含了程序数据,例行程序,中断程序和功能四种对象,但不一定在一个模块都有这四种对象的存在,程序模块之间的数据,例行程序,中断程序和功能是可以互相调用的

。 在 RAPID 程序中,只有一个主程序 main ,并且存在于任 意一个程序模块中,并且是作为整个 RAPID 程序执行的起点。



二、ABB 工业机器人程序存储器

ABB 机器人存储器包含应用程序和系统模块两部分。存储器中只允许存在一个主程序,所有例行程序(子程序)与数据无论存在什么位置,全部被系统共享。因此,所有例行程序与数据除特殊定以外,名称不能重复。ABB 工业机器人存储器组成如图 3-2 所示。





• 1. 应用程序 (Program) 的组成

- ・应用程序由主模块和程序模块组成。
 - 主模块 (Main module)包含主程序 (Main routine) 、程序数据 (Program data) 和例行 程序 (Routine) ;
 - 程序模块 (Program modules) 包含程序数据 (Program data) 和例行程序 (Routine)。



- ・ 2. 系统模块 (System modules) 的组成
- ・系统模块包含系统数据(System data)和例行程
 序 (Routine)。
- ・所有 ABB 机器人都自带两个系统模块 , USER 模 块和 BASE 模块。使用时对系统自动生成的任何模 块不能进行修改。



三、编程指令及应用

•1.基本运动指令及其应用

・常用基本运动指令有: MoveL、MoveJ、MoveC

- MoveL:直线运动。
- ・ MoveJ : 关节轴运动。
- MoveC :圆周运动。



(2)圆弧运动指令的应用

圆弧由起点、中点和终点三点确定,使用圆弧运 动指令 MoveC , 需要示教确定运动路径的起点、中 点和终点。圆弧运动路径如图 5-5 所示。

PO



图 5-5 圆弧运动路径



解析:起点为 P0,也就是机器人的原始位置,使用 MoveC 指令会自动显示需要确定的另外两点,即中点 和终点,程序语句如下:

 $MoveC\ p1\ , \ p2\ , \ v100\ , \ z1\ , \ tool1$



• 例 5-2 : 如图 5-6 所示, 令机器人沿圆心为 P 点, 半径为 100mm 的圆运动机器人的运动路径。机器人从起始点 P , 经过 P₁、 P₂、 P₃、 P₄、 P₁点, 回到起始点 P。 :





16

方法一:机器人示教 P、P₁、P₂、P₃、P₄点,示教方法见 5.2.2。机器人长方形路径的程序如下:

MoveJ p, v100, z50, too11/Wobj1; MoveL p1, v100, fine, too11/Wobj1; MoveC p2, p3, v100, z1, too11/Wobj1; MoveC p4, p1, v100, z1, too11/Wobj1; MoveJ p, v100, z50, too11/Wobj1;



方法二:

程序如下:

MoveJ p, v500, z1, tool1 ; MoveL

offs (p, 80, 0, 0), v500, z1, tool1; MoveC offs (p, 0, 80, 0), offs (p, -80, 0, 0, 0), v500, z1, tool1;

MoveC offs (p, 0, -80, 0), offs (p, 80, 0, 0, 0), v500, z1, tool1;

MoveJ p, v500, z1, tool1



(1) IF:满足不同条件,执行对应程序 例如: IF reg1>5 THEN Set Do1; ENDIF

如果 reg1>5 条件满足,则执行 Set Do1 指令。



(2) FOR:根据指定的次数,重复执行对应程序

- 0
- 例如: FOR I FROM 1 TO 10 DO
- routine1;
- **ENDFOR**
- 重复执行 10 次 routine1 里的程序。

FOR 指令后面跟的是循环计数值,其不用在程序数据中定义,每次运行一遍 FOR 循环中的指令后会自动执行加1操作。



(3) WHILE:如果条件满足,则重复执行对应程 序

例如: WHILE reg1<reg2 DO

```
reg1 : =reg1+1 ;
```

ENDWHILE

如果变量 reg1<reg2 条件一直成立,则重复执行 reg1 自加1,直至 reg1<reg2 条件不成立为止。



(4) TSET:根据指定变量的判断结果,执行对应程序 例如:TSET reg1

- **CASE 1** :
- routine1;
 - **CASE 2** :
- Routine2 ;
- **DEFAULT** :
- Stop ;
- ENDTEST



(4) TSET:根据指定变量的判断结果,执行对应 程序

判断 reg1 数值,若为1则执行 routine1;若为 2则执行 routine2,否则执行 Stop。

在 CASE 中,若多种条件下执行同一操作,则可 合并在同一 CASE 中,如: CASE 1,2, 3: routine1;



3. 其他常用指令 (1) "!":注释行 在语句前面加上"!",则图

在语句前面加上"!",则整行语句作为注释行,不 被程序执行。

例如: ! Goto the pick position ;

MoveL pPick, v1000, fine, tool1\WObj : =wobj1 ;



四、新建与加载程序

新建与加载一个程序的步骤如下:

- · (1)在主菜单下,选择程序编辑器;
- (2)选择任务与程序;
- ・(3)若创建新程序,按新建,然后打开软件盘对程 序进行命名;若编辑已有程序,则选加载程序,显示 文件搜索工具;

・(4)在搜索结果中选择需要的程序, 按确认, 程序 被加载, 如图 3-7 所示。为了给新程序腾出空间, 可 以先删除先前加载的程序。





· 例行程序有不同的语句组成,如运动指令, 等待指令等。每句指令又由不同的变量组成,变 量视它们自己的类型,可改变或省略。

· 程序中各指令的含义如图 3-8 所示。



A- 直线运动 指令名称

B- 点位被隐藏 的数值

C-可定义的运 动速度

D-可定义的 运动点类型 (精确点)

| | Mot -0001641) Sto | ors Off pped (Speed 10 | 0%) | [X] |
|--|--|--|----------------|------|
| 🔭 Testprogram in MAIN/MainN | lodule/main | | Row 4 of 11 ro | w(s) |
| Tasks and Programs 🛛 🔻 | Modules | • | Routines | - |
| MODULE MainMod PROC main() | ule | | ÷ | |
| ➡ Movel *, * Movel * ▼ Movel * ▼ | 71000, fine, 1000, fine, 1000, fine, 1000, fine, C D | tool0; tool0; tool0; tool0; E | | |
| ENDMODULE | | | | |
| Add Add Edit | ▲ Debug ⁴ | Modify Position | | |
| MAIN: MainMod | | | Ę | |

如图 3-8 机器人程序中指令含义

E-有效工具



(1)调节运行速度 在开始运行程序前,为了保证操作人员和设备的安全,应将机器人的运动速度调整到75%。速度调节方法如下:

- -1) 按快捷键;
- -2) 按速度模式键,显示如图 3-9 所示的快捷速度调节按钮;
- -3)将速度调整为75%或50%;
- -4)按快捷菜单键关闭窗口。







- (2)运行程序
- ・ 运行刚才打开的程序,先用手动低速,单步执行,再连续执 行。
 - 运行时是从程序指针指向的程序语句开始,图 3-10 的"A" 指示的即为程序指针。运行步骤如下:
- ・1)将机器人切换至手动模式
- 2) 按住示教器上的使能键
- ・ 3) 按单步向前或单步向后,单步执行程序。执行完一句即 停止。



| Testprogram in MAIN/MainModule/main Row 4 of 11 row(s) Tasks and Programs Modules Routines MODULE MainModule PROC main () MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL * Module Module Add Edit Debug Modify Position | ABB | Manual IRB_140 | (SEVST-L- | 0001641) | Mot Sto | ors Off pped (Speed | 100%) | X |
|---|-------------------|-------------------|---------------|------------|------------|---|-------------|--------|
| Tasks and Programs Modules Routines MODULE MainModule PROC main () Image: MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL * MoveL *, v100, fine, tool0; MoveL * MoveL *, v10, fine, tool0; MoveL * MoveL *, v10, fine, tool0; MoveL * MoveL *, v10, fine, tool0; MoveL * </td <td>🔚 Testprogr</td> <td>am in MA</td> <td>IN/MainM</td> <td>lodule/mai</td> <td>in</td> <td></td> <td>Row 4 of 11</td> <td>row(s)</td> | 🔚 Testprogr | am in MA | IN/MainM | lodule/mai | in | | Row 4 of 11 | row(s) |
| MODULE MainModule PROC main() → MoveL *, v1000, fine, tool0; MoveL *, book | Tasks ar | nd Program | is 🔻 | | Modules | - | Routines | • |
| MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; A ENDMODULE Add Edit Debug Modify Position | MODUI | LE Mai DC mai | nMod in () | ule | | | - | |
| MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; MoveL *, v1000, fine, tool0; A ENDMODULE Add Edit Debug Modify Position | ⇒ | MoveL | *, 1 | 1000, | fine, | too10 | 7 | |
| Add A Edit Debug Modify Position | | loveL | *, v | 1000, | fine, | tool0; | | |
| Add Add A Edit Debug Modify Position | | loveL | *, v | 1000, | fine, | tool0; | | |
| A ENDMODULE Add A Instruction Edit Debug Modify Position | | loveL | *, v | 1000, | fine, | tool0; | | |
| ENDMODULE Add Edit Debug Modify Position | A | loc | | | | | | |
| Add A Edit Debug Modify Instruction Edit Debug | ENDMO | DULE | | | | | | |
| | Add Instructio | n 📥 | Edit | • | Debug | Modify Positic | , ND | |
| | MAIN: MainMod | | | চি ব | | +12.61 | | |



五、自动运行程序 自动运行程序的步骤如下:

• (1)插入钥匙,将运转模式切换到自动模式,示教器 上显示状态切换对话框,如图 3-11 所示;





・ (2)按 OK ,关闭对话框,示教器上显示生产窗口, 如图 3-12 所示;





- (3) 按马达上电 / 失电按钮激活电机;
- (4)按连续运行键开始执行程序;
- (5) 按停止键停止程序;
- (6)插入钥匙,运转模式返回手动状态。





工业机器人的控制系统

- ・大多数工业机器人属于示教 再现方式的工业机器人。无论是示教过程和还是再现过程,工业机器人的控制系统均处于边工作边计算的状态中。
- ・要求工业机器人控制系统是具有实时中断控制与 多任务处理功能的专用计算机控制系统。



一、控制系统的软、硬件任务分配

一般工业机器人控制系统的软硬件任务分配如下: ·速度平滑控制、自动加减速控制与防振控制采用软件方式处理

·硬件系统应配合其它软件完成以下模块功能:

- -1)系统控制。
- 2) 示教操作、编程与 CRT 显示。
- -3)多轴位置、速度协调控制(再现)。
- 4) I/O 通讯与控制接口。
- 5) 各种安全与连锁控制。

工业机器人控制系统的典型硬件结构如图 3-13 所示。





图 3-13 工业机器人控制系统的典型硬件结构



二、控制系统软件功能

- ・ 工业机器人的基本动作概念与软件功能如图 3-14 所示。
- ・工业机器人的柔性体现在其运动轨迹、作业条件 和作业顺序能自由变更,变更的灵活程度取决于 其软件水平。
- ・工业机器人按照操作人员的示教动作及要求进行 作业,操作人员可以根据作业结果或条件进行修 正,直到满足要求为止。





图 3-14 工业机器人的基本动作概念与软件功能



・软件系统应具有以下基本功能

- -1) 示教信息输入;
- -2) 工业机器人本体及对外部设备的动作控制;
- -3)运动轨迹在线修正;
- -4)实时安全系统等。