

项目3

循迹模块编程与操作

项目分析

为适应精密加工立体工件的需要，激光加工机器人应运而生。在汽车制造领域，激光切割机器人已经广泛应用于小型汽车顶窗等空间曲线加工，如图 3.1 所示。本项目模拟激光切割轨迹完成循迹模块的编程与操作。循迹模块由三角形、正方形、五角星、圆形、“J”、“S”、“H”、“B” 8 个图形的轨迹组成，主要学习工业机器人的基本操作及简单编程，如图 3.2 所示。



图 3.1 激光切割机器人



图 3.2 循迹模块

学习目标

知识目标

- 掌握工业机器人程序的结构与建立方法。
- 掌握运动指令的运动特点、格式及使用方法。



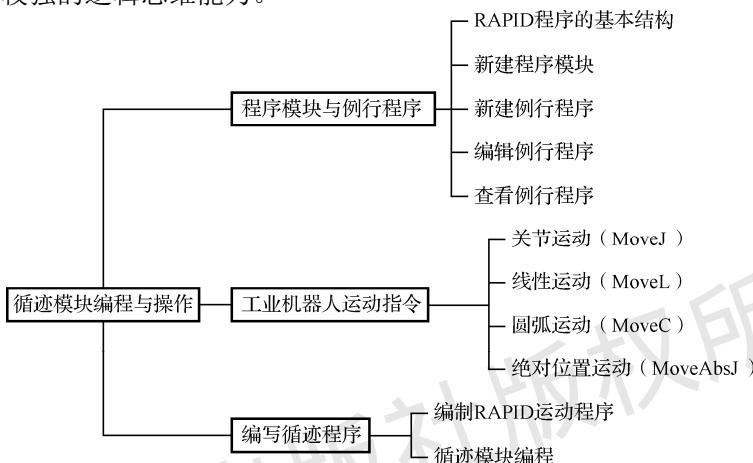
能力目标

- 独立完成程序模块的新建与保存，进行 RAPID 程序编写、调试、自动运行。
- 独立完成循迹模块的程序编写与调试。

素质目标

- 培养学生良好的动手能力、沟通能力和团队合作能力。
- 培养学生较强的逻辑思维能力。

知识分布网络



相关知识

3.1 程序模块与例行程序



扫一扫看程序
模块与例行程
序教学课件

3.1.1 RAPID 程序的基本结构

RAPID 程序中包含了一连串控制工业机器人的指令，执行这些指令可以实现对 ABB 工业机器人的控制操作。RAPID 程序的基本架构见表 3.1。

表 3.1 RAPID 程序的基本架构

RAPID 程序			
程序模块 1	程序模块 2	程序模块 3	程序模块 4
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			

RAPID 程序的架构说明：

(1) RAPID 程序由程序模块与系统模块组成。一般情况下只通过新建程序模块来构建工业机器人的程序，系统模块多用于系统方面的控制。



项目3 循迹模块编程与操作

(2) 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等，这样便于归类管理不同用途的例行程序与数据。

(3) 每个程序模块可包含程序数据、例行程序、中断程序和功能四种对象，但不一定在一个模块中都有这四种对象，程序模块之间的程序数据、例行程序、中断程序和功能是可以互相调用的。

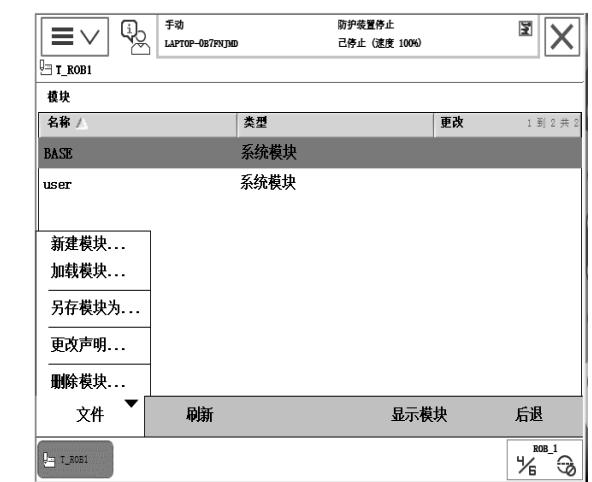
(4) 在 RAPID 程序中，只有一个主程序 Main，并且存于任意一个程序模块中，作为整个 RAPID 程序执行的起点。

3.1.2 程序模块和例行程序的建立

1. 新建程序模块

在建立程序模块之前，要首先确认工业机器人处于“手动”模式。建立程序模块的步骤见表 3.2。

表 3.2 建立程序模块的步骤

	第 1 步：点击示教器左上角“≡∨”，选择“程序编辑器”，在弹出的提示框中选择“取消”按钮
	第 2 步：示教器中自带的两个模块 BASE 和 user 均为系统模块，不能在系统模块中进行编程，应新建程序模块。在示教器左下角单击“文件”，选择“新建模块”



续表

	<p>第3步：在弹出的提示框中单击“是”按钮</p>
	<p>第4步：更改新建模块的名称。根据工程实际情况进行命名，名称不宜过长，不宜使用关键字，并且名称的首位必须为字母，其后可以根据具体情况添加数字、下画线等。名称确定好后，单击“确定”按钮</p>
	<p>第5步：构建完成的程序模块</p>



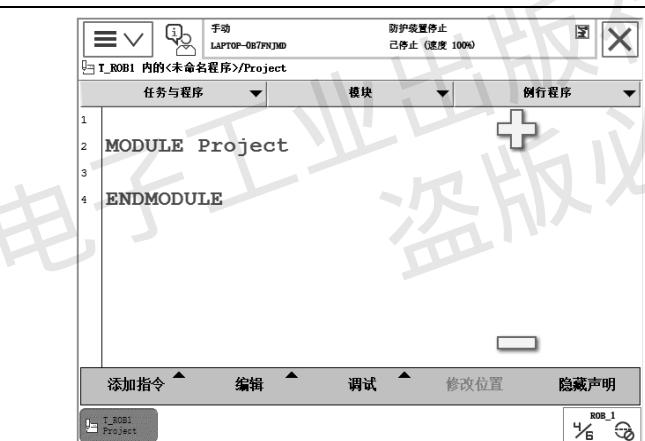
项目3 循迹模块编程与操作

2. 新建例行程序

在每个程序模块中，需要有一个主程序。因此需要先新建一个主程序 Main，再新建一个普通的例行程序。Main 作为程序模块的主运行程序，其他程序为子程序。建立例行程序的步骤见表 3.3。



表 3.3 建立例行程序的步骤

	<p>第 1 步：选择“Project”程序模块，单击“显示模块”</p>
	<p>第 2 步：在显示的界面单击“例行程序”，可以看到新程序模块中没有程序</p>
	<p>第 3 步：在显示的例行程序界面单击“文件”，选择“新建例行程序”</p>



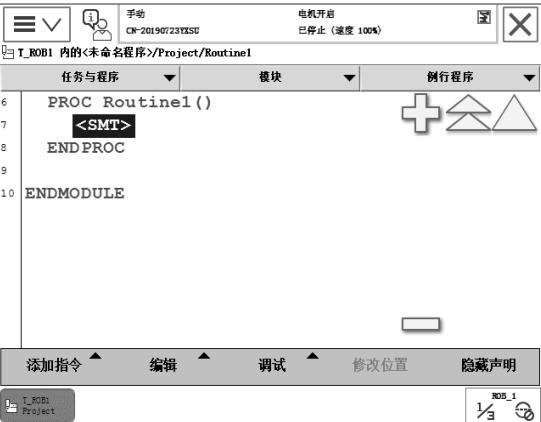
续表

	<p>第 4 步：将例行程序名更改为“Main”，其他参数默认，单击“确定”按钮，主程序建立完成</p>
	<p>第 5 步：再次单击“文件” — “新建例行程序”，建立子程序“Routine1”</p>
	<p>第 6 步：选择新建的例行程序“Routine1”，单击“显示例行程序”</p>



项目3 循迹模块编程与操作

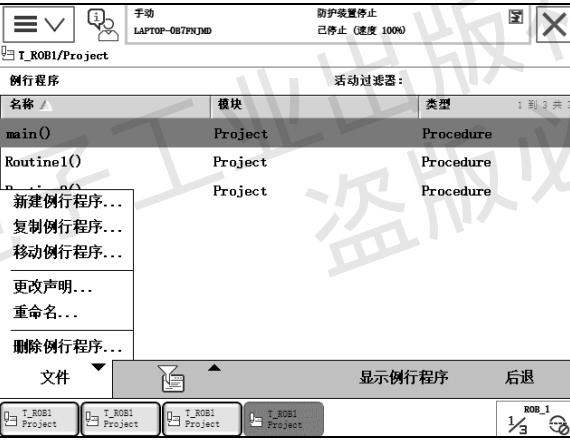
续表

	第7步：在蓝色高亮显示的“<SMT>”处可以添加指令，开始编程
---	---------------------------------

3.1.3 编辑例行程序

编辑例行程序的方法见表 3.4。

表 3.4 编辑例行程序的方法

	1. 选择“文件”菜单，可进行复制例行程序、移动例行程序、更改声明、重命名、删除例行程序等操作
	2. 单击“复制例行程序”，可对复制的程序名称、类型、存储模块进行修改



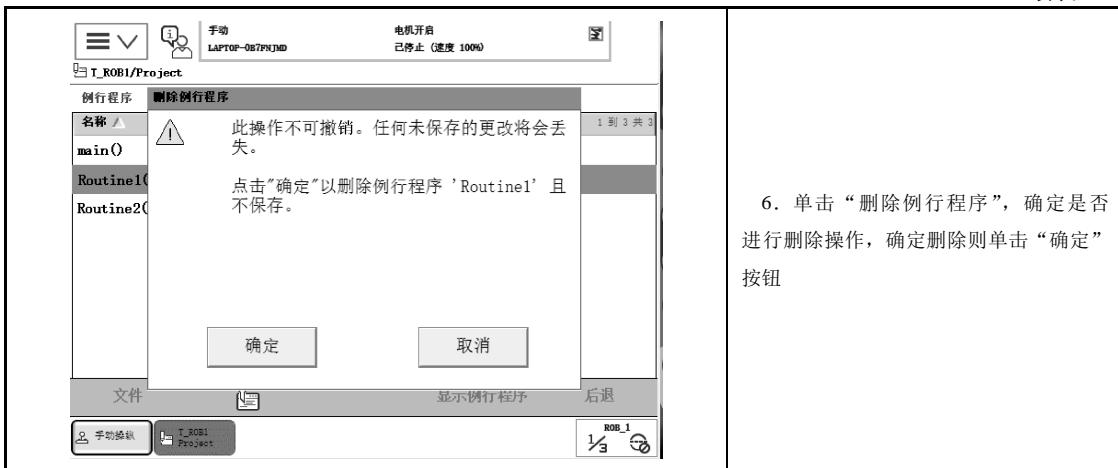
续表

	<p>3. 单击“移动例行程序”，可将选中的例行程序移动到其他程序模块中</p>
	<p>4. 单击“更改声明”，可对例行程序的类型、所属模块进行修改</p>
	<p>5. 单击“重命名”，在弹出的键盘中输入新的名称，单击“确定”按钮</p>



项目3 循迹模块编程与操作

续表

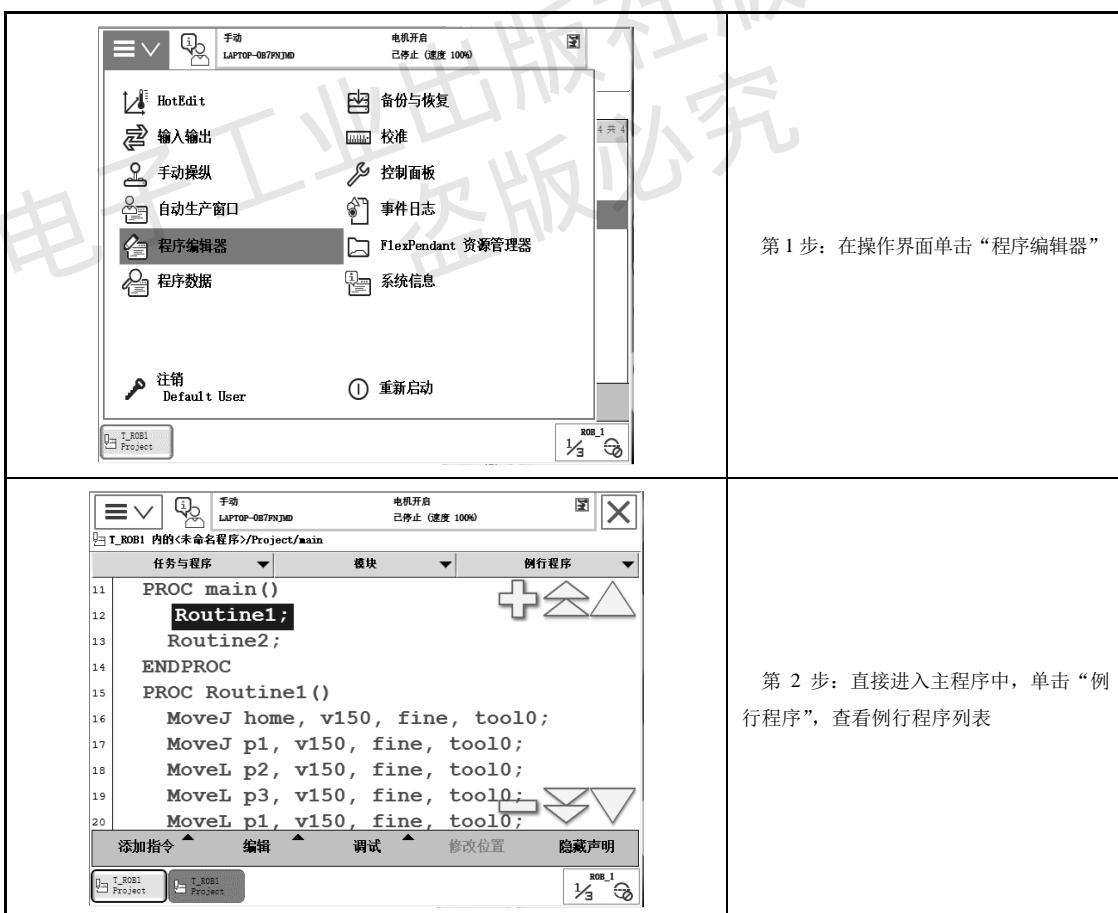


6. 单击“删除例行程序”，确定是否进行删除操作，确定删除则单击“确定”按钮

3.1.4 查看例行程序

查看例行程序的步骤见表 3.5。

表 3.5 查看例行程序的步骤



第 1 步：在操作界面单击“程序编辑器”

第 2 步：直接进入主程序中，单击“例行程序”，查看例行程序列表



续表

<p>第 3 步：程序模块中包含的所有例行程序都被显示出来</p>	
<p>第 4 步：单击“后退”按钮，选择“模块”，可以查看模块列表</p>	
<p>第 5 步：单击“关闭”按钮，就可以退出程序编辑器</p>	



3.2 工业机器人运动指令



工业机器人在空间中运动主要有 4 种方式，分别是关节运动（MoveJ）、线性运动（MoveL）、圆弧运动（MoveC）和绝对位置运动（MoveAbsJ）。

1. 关节运动（MoveJ）

关节运动指令用于在对路径精度要求不高的情况下，工业机器人的工具中心点 TCP 从一个位置移动到另一个位置，两个位置之间的路径不一定是直线，而是由工业机器人自己规划的一条路径，如图 3.3 所示。关节运动适合工业机器人大范围运动时使用，不容易在运动过程中出现关节轴进入机械死点的问题，其在点对点搬运的作业场合广泛应用。关节运动指令结构如图 3.4 所示。

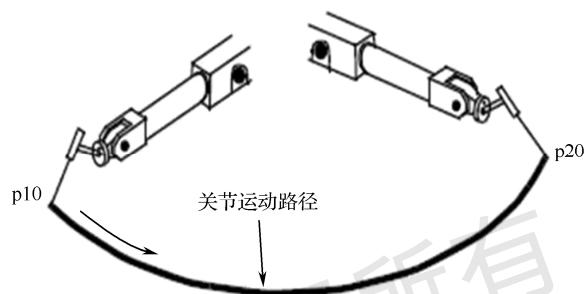


图 3.3 关节运动路径



图 3.4 关节运动指令结构

图 3.4 的关节运动指令“MoveJ p10, v200, fine, JiGuangBi_tool\Wobj:=XunJi_wobj;”包含 5 个程序数据参数，其参数功能及说明见表 3.6。

表 3.6 关节运动指令解析

参 数	含 义	说 明
p10/p20	目标点位置数据	包含 6 个关节轴数据，通过“修改位置”菜单命令记录
v200	运动速度数据，200 mm/s	该值越大，工业机器人运动速度越快，最高为 5000 mm/s，在手动操纵中一律限速为 250 mm/s



续表

参数	含义	说明
fine	转弯区数据	此区域数据描述了所生成拐角路径的大小, 单位为 mm, 设置为 fine 表示无拐角
JiGuangBi_tool	运动期间使用的工具坐标数据	在“手动操纵”中设置机器人的工具坐标系, 在程序编辑时将自动生成该工具坐标系
XunJi_wobj	运动期间使用的工件坐标数据	在“手动操纵”中设置机器人的工件坐标系, 在程序编辑时将自动生成该工件坐标系

2. 线性运动 (MoveL)

线性运动是工业机器人的 TCP 从起点到终点之间的路径始终保持为直线, 一般在如焊接、涂胶等对路径要求高的场合使用此指令, 如图 3.5 所示。但需要注意, 空间直线距离不宜太远, 否则容易到达工业机器人的轴限位或死点。如想获得精确路径, 则两点距离较短为宜。其指令结构如图 3.6 所示。



扫一扫看工业机器人线性运动指令微课视频

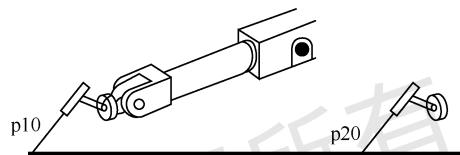


图 3.5 线性运动路径



图 3.6 线性运动指令结构

线性运动指令参数调用同关节运动相似, 指令中所使用参数见表 3.6。



扫一扫看工业机器人圆弧运动指令微课视频

3. 圆弧运动 (MoveC)

圆弧运动路径是在工业机器人可到达的空间范围内定义 3 个位置点, 第一个点是圆弧的起点, 第二个点用于控制圆弧的曲率, 第三个点是圆弧的终点, 如图 3.7 所示。

由于确定一段圆弧需要 3 个数据才能完成, 而圆弧运动指令里面只有两个数据, 即确定圆弧所需的第

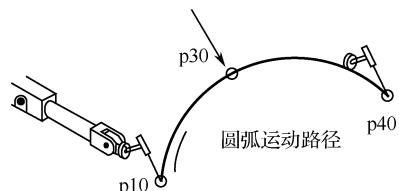


图 3.7 圆弧运动路径



项目3 循迹模块编程与操作

二个点和第三个点，因此，确定圆弧所需的第一个点实际是上一条指令执行完毕后工业机器人所停的位置，所以圆弧运动指令一般不能应用到所编写程序的第一条。圆弧运动指令结构如图 3.8 所示。



图 3.8 圆弧运动指令结构

4. 绝对位置运动 (MoveAbsJ)

绝对位置运动指令是工业机器人的运动使用 6 个轴和外轴的角度值来定义目标位置数据。其指令结构如图 3.9 所示。



图 3.9 绝对位置运动指令结构

该绝对位置运动指令 “MoveAbsJ *|\NoEoffs, v200, fine, JiGuangBi_tool\WObj:=XunJi_wobj;” 包含 6 个程序数据参数，其参数功能及说明见表 3.7。

表 3.7 绝对位置运动指令解析

参 数	含 义	说 明
*	目标点位置数据	包含 6 个关节轴数据，可直接在“调试”栏目中修改关节轴数据
\NoEoffs	外轴不带偏移数据	无



续表

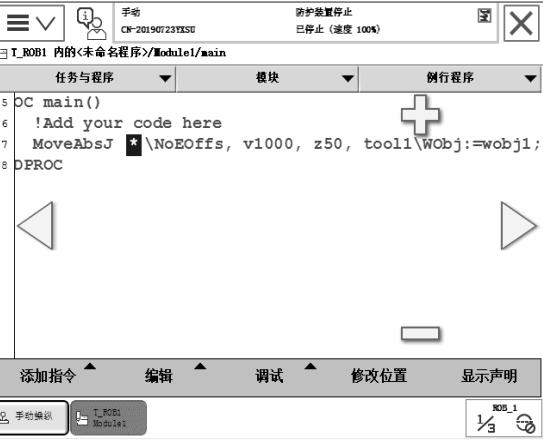
参 数	含 义	说 明
v200	运动速度数据	该值越大，工业机器人运动速度越快，最高为 5000 mm/s，在手动操纵中一律限速为 250 mm/s
fine	转弯区数据	转弯区数据描述了所生成拐角路径的大小，单位为 mm，设置为 fine 表示无拐角
JiGuangBi_tool	运动期间使用的工具坐标数据	在“手动操纵”中设置机器人的工具坐标系，在程序编辑时将自动生成该工具坐标系
XunJi_wobj	运动期间使用的工件坐标数据	在“手动操纵”中设置机器人的工件坐标系，在程序编辑时将自动生成该工件坐标系

绝对位置运动指令 MoveAbsJ 常用于工业机器人 6 个轴回到机械原点 (0°) 的位置，具体操作步骤见表 3.8。

扫一扫看工业机器人 6 轴回到机械零点微课视频

小知识：在添加或修改工业机器人的运动指令之前，一定要首先确认所选用的工具坐标与工件坐标。

表 3.8 工业机器人 6 轴回机械原点的步骤

	<p>第 1 步：进入“手动操纵”界面，确认已选定工具坐标与工件坐标</p>
	<p>第 2 步：添加绝对位置运动指令，并选中位置数据变量</p>



项目3 循迹模块编程与操作

续表

	<p>第3步：为选中的目标点命名，并单击“确定”按钮</p>
	<p>第4步：选中新建的目标点 jpos10，在“调试”菜单中单击“查看值”</p>
	<p>第5步：可查看1~6轴的当前值</p>



续表

	<p>第 6 步：将各轴值改为 0，单击“确定”按钮，该点即变为工业机器人的机械原点。此时，再进行程序调试，则工业机器人会回到机械原点</p>
---	---

项目实施

本项目实施的前提是激光笔工具已从工具库中取出，安装在法兰盘上，且激光笔始终是打开的状态。在后续的项目中将会学习通过 I/O 控制信号来控制激光笔打开或关闭。

3.3 编写循迹程序

3.3.1 编制一个可以运行的 RAPID 运动程序



循迹模块轨迹如图 3.10 所示，以三角形为例，编制一个 RAPID 程序，使得工业机器人的 TCP 从当前位置向 home 点以关节运动方式前进，速度是 150mm/s，转弯区数据是 fine，使用的工具数据是 tool0，工件坐标数据是 wobj0。接着以相同的速度、转弯区数据、工具坐标、工件坐标从 home 点出发做关节运动至三角形第一个点 p1，继续做线性运动至三角形第二个点 p2 和第三个点 p3，并做线性运动回到三角形第一个点 p1，最后回到 home 点位置。

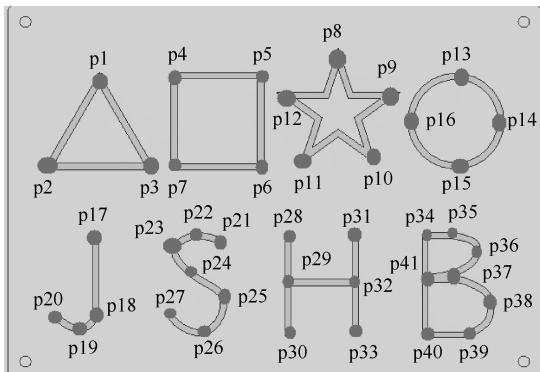


图 3.10 循迹模块示意图

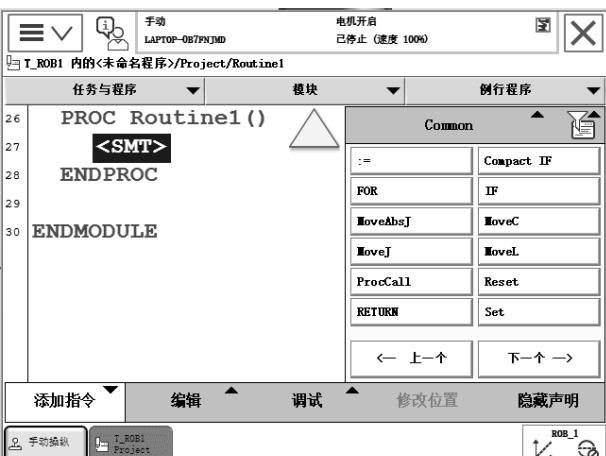


项目3 循迹模块编程与操作

1. 添加指令

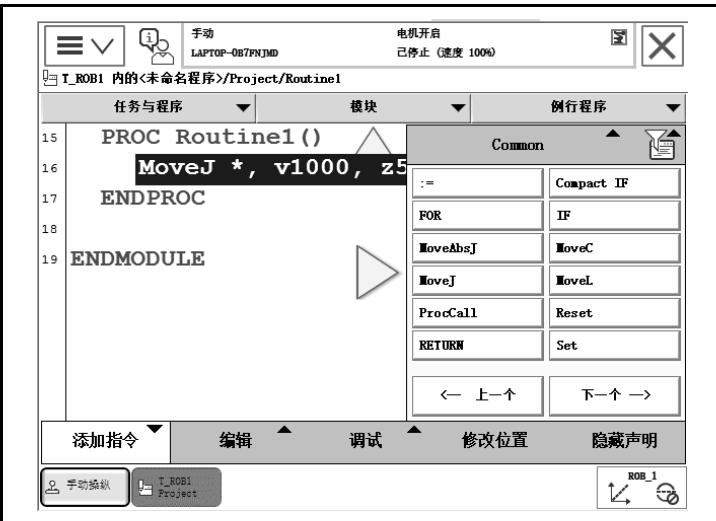
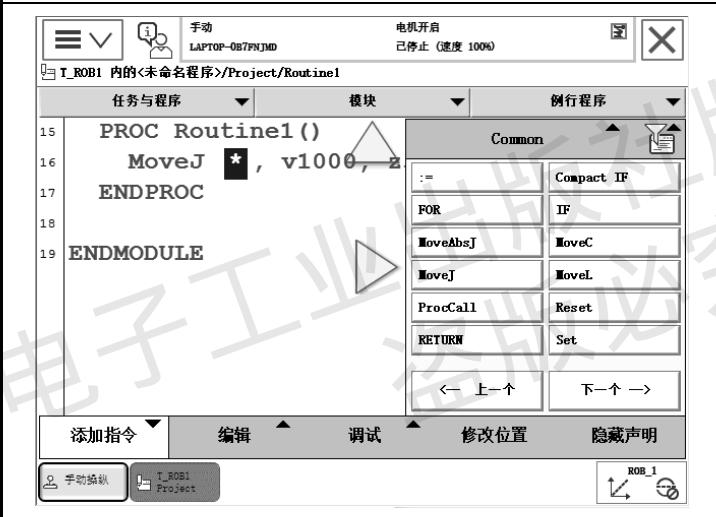
添加指令步骤见表 3.9。

表 3.9 添加指令步骤

	第 1 步：在操作界面上单击“程序编辑器”
	第 2 步：确认已选定的工具坐标与工件坐标
	第 3 步：选中<SMT>为添加指令位置，打开“添加指令”菜单



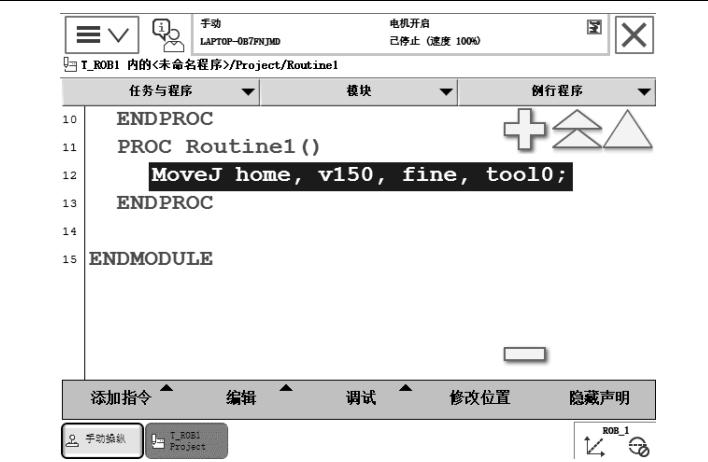
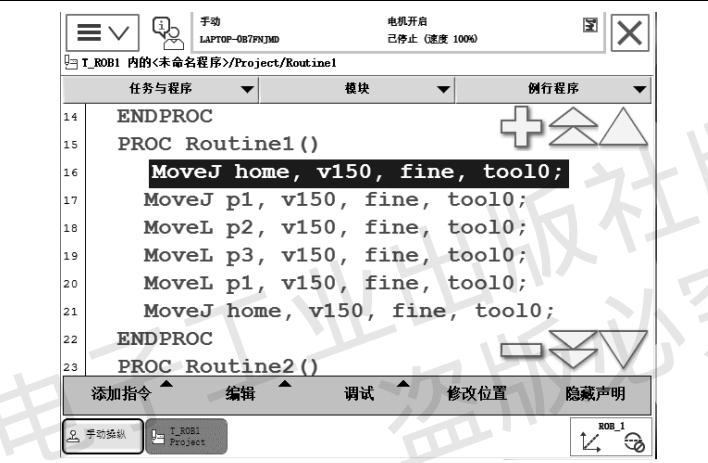
续表

	<p>第 4 步：在指令列表中选择“MoveJ”指令</p>
	<p>第 5 步：单击“*”，编辑 MoveJ 指令中的目标点位置数据 home</p>
	<p>第 6 步：继续编辑指令中的程序数据参数 v150 和 fine</p>



项目3 循迹模块编程与操作

续表

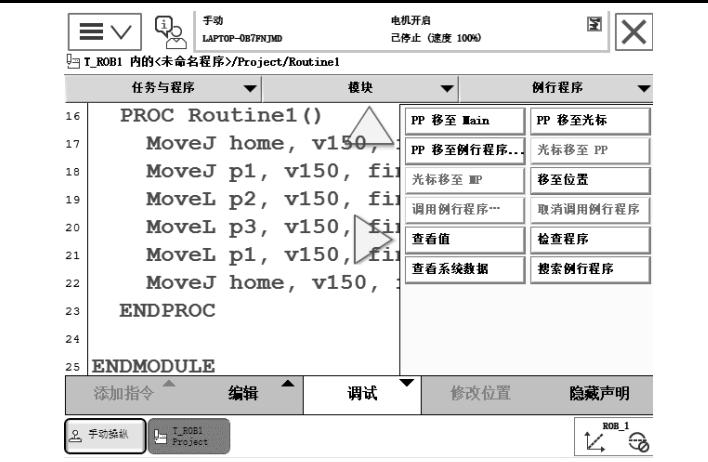
	第7步：关闭指令列表，可以看到MoveJ指令，并手动操作工业机器人完成点的示教器
	第8步：继续完成其他程序语句的编辑

2. 程序调试

程序调试步骤见表 3.10。



表 3.10 程序调试步骤

	第1步：将机器人控制柜调整到手动限速模式，单击示教器下方的“调试”菜单
---	-------------------------------------



续表

<p>选定的例行程序： Routine1 从列表中选择一个例行程序。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>名称</th> <th>类型</th> <th>模块</th> </tr> </thead> <tbody> <tr> <td>main</td> <td>程序</td> <td>Project</td> </tr> <tr style="background-color: #cccccc;"> <td>Routine1</td> <td>程序</td> <td>Project</td> </tr> <tr> <td>Routine2</td> <td>程序</td> <td>Project</td> </tr> </tbody> </table>	名称	类型	模块	main	程序	Project	Routine1	程序	Project	Routine2	程序	Project	<p>第 2 步：单击“PP 移至例行程序”，在弹出的菜单中选择 Routine1 例行程序，单击“确定”按钮</p>
名称	类型	模块											
main	程序	Project											
Routine1	程序	Project											
Routine2	程序	Project											
<p>任务与程序 模块 例行程序</p> <pre> 16 PROC Routine1() 17 MoveJ home, v150, 18 MoveJ p1, v150, f1 19 MoveL p2, v150, f1 20 MoveL p3, v150, f1 21 MoveL p1, v150, f1 22 MoveJ home, v150, 23 ENDPROC 24 PROC Routine2() 25 <SMT> </pre>	<p>第 3 步：光标移动到 Routine1 程序的第一行，将工业机器人使能上电，单击“单步调试”按钮，逐条调试程序，重点查看工业机器人位置是否合适</p>												
<p>任务与程序 模块 例行程序</p> <pre> 16 PROC Routine1() 17 MoveJ home, v150, 18 MoveJ p1, v150, f1 19 MoveL p2, v150, f1 20 MoveL p3, v150, f1 21 MoveL p1, v150, f1 22 MoveJ home, v150, 23 ENDPROC 24 PROC Routine2() 25 <SMT> </pre>	<p>第 4 步：如果想对某一条程序进行调试，可以先将需要调试的程序选中，然后在“调试”菜单下选择“PP 移至光标”，以调试第 2 条程序为例，此功能只能将 PP 在同一个例行程序中跳转。如要将 PP 移至其他例行程序，可使用“PP 移至例行程序”功能</p>												



项目3 循迹模块编程与操作

3. 程序自动运行

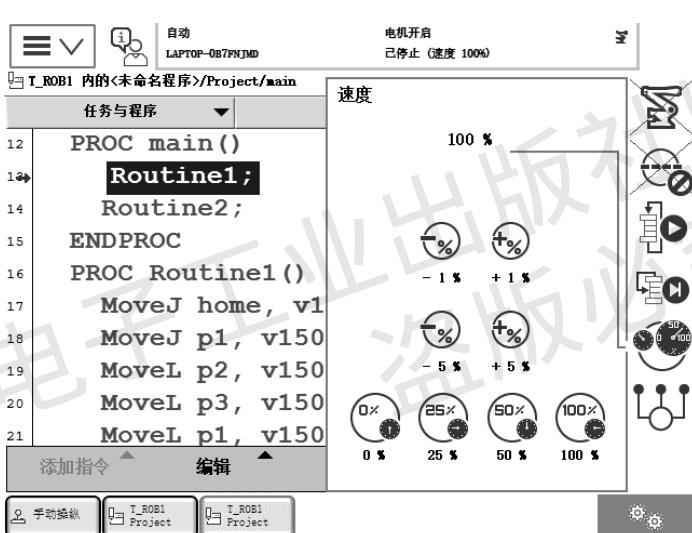
在手动调试没有发现任何问题的情况下，就可以让工业机器人程序自动运行，其步骤见表 3.11。

表 3.11 程序自动运行步骤

	<p>第 1 步：将工业机器人控制柜调整到“自动”模式</p>
<p>The dialog box contains the following text: 已选择自动模式。 程序执行不会从主程序开始。 先单击“确认”，然后单击“确定”。 要取消，切换回手动。</p>	<p>第 2 步：先单击“确认”，再单击“确定”，完成状态切换</p>
<p>The dialog box contains the following text: 确定将 PP 移至 main? 是 否</p>	<p>第 3 步：在弹出的界面中选择“PP 移至 main”，并单击“是”按钮</p>



续表

	<p>第 4 步：此时光标跳转到主程序第一行，按下控制柜上白色确认按钮，工业机器人程序开始自动运行</p>
	<p>第 5 步：自动运行时的速度调整，可以通过示教器的快捷菜单进行，通过“+”和“-”按钮调整</p>

3.3.2 循迹模块编程

1. 程序设计

根据工业机器人运动轨迹编写工业机器人程序时，首先要根据控制要求绘制工业机器人程序流程图，然后编写工业机器人主程序和子程序。子程序主要包括等边三角形子程序、方形子程序、圆形子程序和五角星子程序等。编写子程序前要先设计好工业机器人的运行轨迹并定义好机器人的程序点。根据控制功能，设计工业机器人程序流程图，如图 3.11 所示。

2. 循迹模块路径规划

针对图 3.10 中各图像的轨迹点，循迹模块需要示教的点见表 3.12。



项目3 循迹模块编程与操作

表 3.12 循迹模块需要示教的点

序号	点序号	注释	备注
1	xunji_home	工业机器人循迹初始位置	需示教
2	xunji_1~xunji_3	等边三角形轨迹点	需示教
3	xunji_4~xunji_7	方形轨迹点	需示教
4	xunji_8~xunji_12	五角星轨迹点	需示教
5	xunji_13~xunji_16	圆形轨迹点	需示教
6	xunji_17~xunji_20	“J”轨迹点	需示教
7	xunji_21~xunji_27	“S”轨迹点	需示教
8	xunji_28~xunji_33	“H”轨迹点	需示教
9	xunji_34~xunji_41	“B”轨迹点	需示教



图 3.11 工业机器人程序流程图



扫一扫看
循迹程序
源代码

3. 程序编写

根据工作任务的要求和程序流程图，建立相应的程序模块和例行程序。循迹单元的例行程序由一个主程序（main）和若干子程序组成，子程序为“xj_xunji”“xj_sanjiaoxing”“xj_zhengfangxing”“xj_wujiaoxing”“xj_yuanxing”“xj_j”“xj_s”“xj_h”“xj_b”，循迹模块程序如图 3.12 所示。



图 3.12 循迹模块程序

(1) 主程序编写。

编写主程序，在“main()”程序中，只需调用“xj_xunji()”子程序即可，具体程序如下：

```
PROC main()
xj_xunji;           //调用“xj_xunji”子程序;
ENDPROC
```

(2) 循迹程序编写。

编写循迹程序，在“xj_xunji ()”程序中，考虑好工业机器人的运动过程，调用各个子程序，具体程序如下：

```
PROC xj_xunji()
jiguangbi_qu;      //调用“jiguangbi_qu”子程序，夹取激光笔;
xj_sanjiaoxing;    //调用“xj_sanjiaoxing”子程序，画三角形;
xj_zhengfangxing; //调用“xj_zhengfangxing”子程序，画正方形;
xj_wujiaoxing;     //调用“xj_wujiaoxing”子程序，画五角星;
xj_yuanxing;       //调用“xj_yuanxing”子程序，画圆形;
xj_j;               //调用“xj_j”子程序，画字母“J” ;
xj_s;               //调用“xj_s”子程序画字母“S” ;
xj_h;               //调用“xj_h”子程序画字母“H” ;
xj_b;               //调用“xj_b”子程序画字母“B” ;
jiguangbi_fang;    //调用“jiguangbi_fang”子程序，放置激光笔;
ENDPROC
```

(3) 以圆形程序编写为例。

编写圆形程序，按照工业机器人运动示教点上的顺序编写程序，圆形参考程序见表 3.13。



项目3 循迹模块编程与操作

表 3.13 圆形参考程序

PROC xj_yuanxing()	
MoveJ xunji_home,v150,z50,tool0;	//回到 home 点
MoveJ xunji_13,v150,z0,tool0;	//运动到圆形第 1 个点
MoveC xunji_14, xunji_15,v150,z0,tool0;	//运动到循迹圆形第 1 道圆弧（半个圆）
MoveC xunji_16, xunji_13, v150, z0, tool0;	//运动到循迹圆形第 2 道圆弧（半个圆），圆形循迹完成
MoveJ xunji_home, v150, z50, tool0;	//工业机器人回到 home 点
ENDPROC	

在编写工业机器人程序时，可以通过 ProcCall 指令在指定的位置调用例行程序。如图 3.13 所示，分别使用 ProcCall 指令调用当前主程序 main 和例行程序 Routine1。

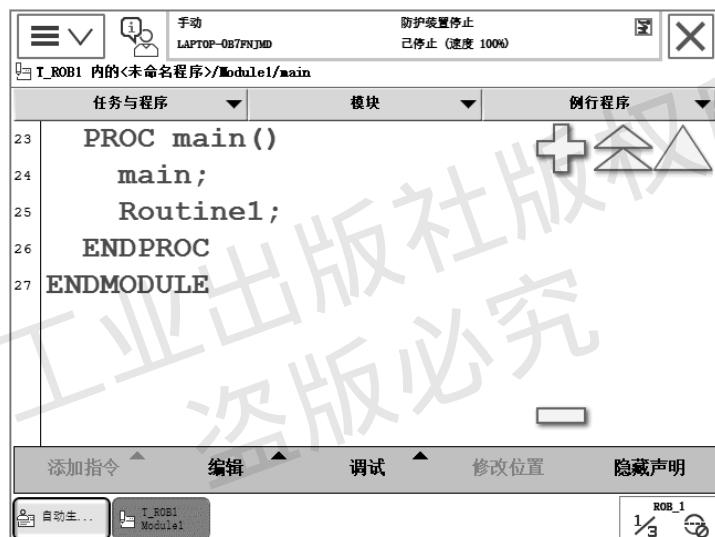


图 3.13 ProcCall 指令的使用

小知识：ProcCall 指令用于调用无返回值例行程序。通过 ProcCall 指令将程序指针移至对应的例行程序并开始执行，执行完例行程序，程序指针返回到调用位置，执行后续指令。

总 结

本项目主要介绍了工业机器人 RAPID 程序结构、程序模块、例行程序的建立方法及工业机器人运动指令的使用方法。重点训练学生熟练运用工业机器人运动指令 MoveJ、MoveL、MoveC 和 MoveAbsJ 进行编程，提高编程效率。



扫一扫
习题3参
考答案

习题3

一、单选题

1. 直线运动指令是工业机器人示教编程时常用的运动指令，编写程序时需通过示教或输入来确定工业机器人末端控制点移动的起点和（ ）。

A. 运动方向	B. 终点	C. 移动速度	D. 直线距离
---------	-------	---------	---------
2. 使用关节运动时，程序命令为（ ）。

A. MoveC	B. MoveJ	C. MoveL	D. MoveAbsJ
----------	----------	----------	-------------
3. 使用直线运动时，程序命令为（ ）。

A. MoveC	B. MoveJ	C. MoveL	D. MoveAbsJ
----------	----------	----------	-------------
4. 使用圆弧运动时，程序命令为（ ）。

A. MoveC	B. MoveJ	C. MoveL	D. MoveAbsJ
----------	----------	----------	-------------
5. 使用绝对关节运动时，程序命令为（ ）。

A. MoveC	B. MoveJ	C. MoveL	D. MoveAbsJ
----------	----------	----------	-------------
6. 运动指令中的 Z50 指的是（ ）。

A. 运动方式	B. 速度数据	C. 转弯半径数据	D. 工具数据
---------	---------	-----------	---------
7. 运动指令中的 v100 指的是（ ）。

A. 运动方式	B. 速度数据	C. 区域数据	D. 工具数据
---------	---------	---------	---------
8. 运动指令中的 tool0 指的是（ ）。

A. 运动方式	B. 速度数据	C. 区域数据	D. 工具数据
---------	---------	---------	---------
9. 运动指令中的 wobj0 指的是（ ）。

A. 运动方式	B. 速度数据	C. 工件数据	D. 工具数据
---------	---------	---------	---------
10. 现有一条圆弧指令“MoveC p1, p2, v500, z30, tool2”，其中 p1 指的是（ ）。

A. 圆弧的起点	B. 圆弧的中间点	C. 圆弧的终点	D. 圆弧的圆心
----------	-----------	----------	----------
11. 现有一条圆弧指令“MoveC p1, p2, v500, z30, tool2”，其中 p2 指的是（ ）。

A. 圆弧的起点	B. 圆弧的中间点	C. 圆弧的终点	D. 圆弧的圆心
----------	-----------	----------	----------
12. 工业机器人编程中有且只能有一个的是（ ）。

A. 程序模块	B. 例行程序	C. 功能指令	D. 主程序
---------	---------	---------	--------
13. 工业机器人行走轨迹是由示教点决定的，一段圆弧至少需要示教（ ）点。

A. 2	B. 3	C. 4	D. 5
------	------	------	------
14. 使用圆弧运动指令在做圆弧运动时一般不超过 240° ，所以一个完整的圆通常需要（ ）条圆弧指令来完成。

A. 1	B. 2	C. 3	D. 4
------	------	------	------
15. 编程时，在语句前加上（ ），则整条语句作为注释行，不被程序执行。

A. !	B. #	C. *	D. **
------	------	------	-------



项目3 循迹模块编程与操作

16. MoveAbsJ 指令的参数 “\NoEoffs” 表示 ()。
A. 外轴的角度数据 B. 外轴不带偏移数据
C. 外轴带偏移数据 D. 外轴的位置数据
17. 工业机器人示教点的数据类型是 ()。
A. tooldata B. string C. robtarget D. singdata

二、判断题

1. 利用示教编程方法编写工业机器人程序时，一般需完成程序名编写、程序编写、程序修改、程序单步调试，然后才能进行自动运行。 ()
2. 工业机器人示教程序调试过程中，为缩短调试的时间，往往需提高程序单步时工业机器人的运动速度，因此采用工业机器人的最大速度来执行单步程序。 ()
3. 使用 MoveJ 指令时工业机器人移动的路径是直线。 ()
4. 使用圆弧运动指令进行圆弧运动时，一条圆弧指令运动的弧度不能超过 240° 。 ()
5. 使用 MoveC 指令完成一个完整的圆周运动需要三条指令。 ()
6. 添加运动指令后通过新建位置数据 (robtarget)，能够记录工业机器人当前的位置。 ()
7. 例行程序可以进行复制、粘贴、重命名操作。 ()
8. 程序中的指令可以进行复制、粘贴、重命名操作。 ()
9. 创建的程序中必须有且只能有一个主程序 Main。 ()
10. 编辑程序时可以选取多行连续的指令。 ()
11. 在保证工业机器人运行轨迹安全的前提下，应尽量减少中间过渡点的选取，删除没有必要的过渡点，这样工业机器人的速度才能提高。 ()
12. 工业机器人编程中常用于工业机器人空间大范围运动的指令是关节运动指令。 ()
13. 关节运动指令可使工业机器人 TCP 从一点运动到另一点，但运动轨迹不一定为直线。 ()
14. 程序数据只能在示教器中的程序数据窗口中建立，不能在建立程序指令时自动生成对应的程序数据。 ()
15. 不同模块间的例行程序根据其定义的范围可互相调用。 ()
16. 机器人轨迹泛指工业机器人在运动过程中的运动轨迹，即运动点的位移、速度和加速度。 ()
17. 指令 MoveAbsJ 是绝对关节运动，工业机器人每轴将以最小的角度运行到指定的轴位置。 ()
18. 指令 MoveL p10, v100, z50, tool0; 所使用的工件坐标系为 Wobj0。 ()

三、多选题

1. 使用示教器自动运行已经编写的程序时，其一般操作步骤包括 ()。
A. 程序选择 B. 切换自动模式 C. 伺服上电 D. 启动运行
2. 工业机器人的控制方式分为 ()。



- A. 点对点控制 B. 点到点控制 C. 连续轨迹控制 D. 点位控制
- 3. 将工业机器人切换到自动模式下运行, 下列操作中()不可实现。
 - A. 编辑程序 B. 切换坐标系 C. 更改速度 D. 查看系统参数
- 4. 示教器编程时的基本运动指令包括()。
 - A. 关节运动指令 B. 插补运动指令 C. 直线运动指令 D. 圆弧运动指令
- 5. 可以对运动指令中的位置数据进行()操作。
 - A. 复制 B. 粘贴 C. 新建 D. 修改位置
- 6. RAPID 语言的三层结构是()。
 - A. 任务 B. 模块 C. 例行程序 D. 功能指令
- 7. 在例行程序列表中可以对程序进行()操作。
 - A. 重命名 B. 复制 C. 移动 D. 更改声明
- 8. 例行程序有()。
 - A. 程序 B. 功能 C. 中断 D. 系统
- 9. 模块包括()。
 - A. 系统模块 B. 功能模块 C. 程序模块 D. 指令模块
- 10. 以下工业机器人的运动方式可控的是()。
 - A. 关节运动 B. 线性运动 C. 圆弧运动 D. 绝对位置运动



项目报告3

班级		姓名		学号			
指导教师			时间	年 月 日			
课程名称							
项目3	循迹模块编程与操作						
学习目标	 <p>掌握工业机器人程序的结构与建立方法，掌握运动指令的运动特点、格式及使用，独立完成循迹模块的程序编写与调试。</p>						
注意事项	<ol style="list-style-type: none">在教师的指导下完成实训任务。实训过程中不要乱改参数。工业机器人运行过程中，禁止碰触工业机器人。工业机器人手动操作时尽量降低运行速度。在运行线性模式时，4轴与5轴不要在一条直线上，否则工业机器人会出现奇异点。工业机器人运动异常时，应及时按下急停开关。						
学习任务	<p>任务1：完成循迹模块中4个图形的循迹程序的编写及运行操作</p> <ol style="list-style-type: none">图形程序指令的编辑。图形程序目标点的示教。图形程序的运行调试。						



续表

	<p>任务 2：完成循迹模块中 4 个字母的循迹程序的编写及运行操作</p> <ol style="list-style-type: none">1. 字母程序指令的编辑。2. 字母程序目标点的示教。3. 字母程序的运行调试。
学习任务	<p>任务 3：完成自定义图案的循迹程序的编写及运行操作</p> <ol style="list-style-type: none">1. 自定义图案程序指令的编辑。2. 自定义图案程序目标点的示教。3. 自定义图案程序的运行调试。
学习心得	



项目评价3

项目3 循迹模块编程与操作				
基本素养 (30分)				
序号	内容	自评	互评	师评
1	纪律 (10分)			
2	安全操作 (10分)			
3	交流沟通 (5分)			
4	团队协作 (5分)			
理论知识 (30分)				
序号	内容	自评	互评	师评
1	例行程序的建立 (6分)			
2	MoveJ 指令的应用 (6分)			
3	MoveL 指令的应用 (6分)			
4	MoveC 指令的应用 (6分)			
5	MoveAbsJ 指令的应用 (6分)			
操作技能 (40分)				
序号	内容	自评	互评	师评
1	完成4个图形的循迹程序的编写及运行操作 (15分)			
2	完成4个字母的循迹程序的编写及运行操作 (15分)			
3	完成自定义图案的循迹程序的编写及运行操作 (10分)			



