

# HSR-JR612六轴机器人 示教与编程



智能制造学院  
工程训练中心



## 一、概述

### 1.1 工业机器人控制系统简介

HSR-JR612工业机器人控制系统主要由 **控制器(HPC-102)** 与 **示教器(HSpad)** 以及运行在这两种设备上的软件所组成。

机器人控制器一般安装于机器人电柜内部，用来控制机器人的伺服驱动器、输入输出等主要执行设备；机器人示教器一般通过电缆连接到机器人电柜上，作为上位机通过以太网与控制器进行通讯。





借助HSpad示教器，用户可以实现HSR-JR612工业机器人控制系统的主要控制功能：

- 手动控制机器人运动
- 机器人程序示教编程
- 机器人程序自动运行
- 机器人运行状态监视
- 机器人系统参数查看





## 1.2 工业机器人的典型结构

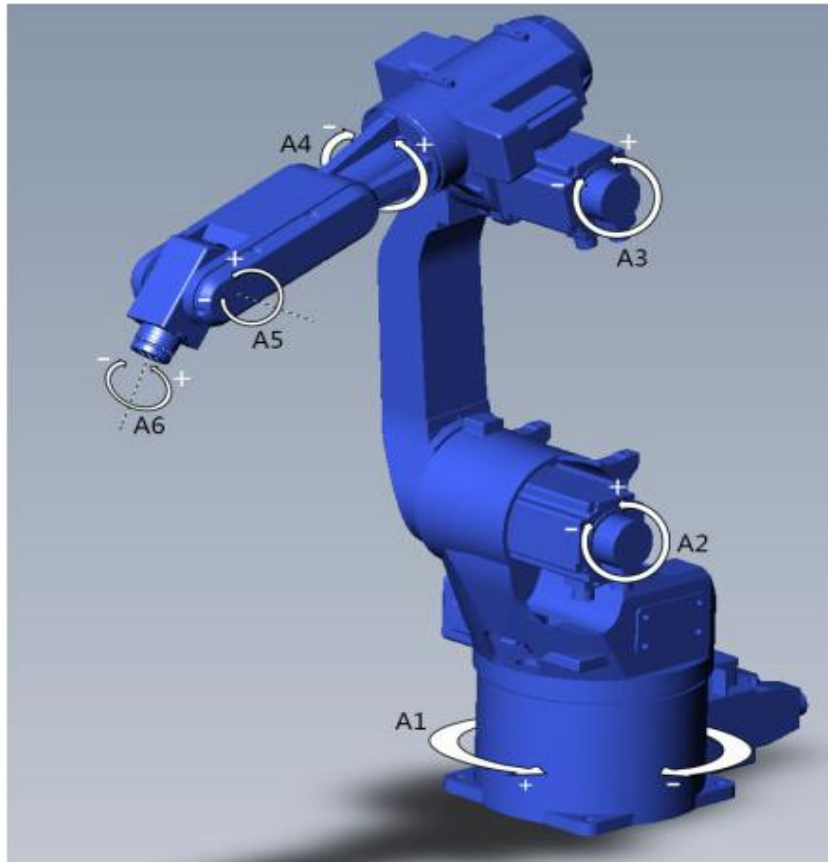


图1-1 六轴机器人关节

A1、A2、A3为定位关节

A4、A5、A6为定向关节

A2、A3、A5 “抬起/后仰”  
为负，“降下/前倾”为正

A1、A4、A6满足右手法则





HSR工业机器人控制系统采用标准D-H法则定义机器人坐标系，即A1与A2关节轴线的公垂线在A1轴线上的交点为基坐标系原点，坐标系方向如图 1-2所示：A4、A5、A6关节轴线共同的交点为手腕中心点，0号工具坐标系位于该点，坐标系方向如图 1-2所示。

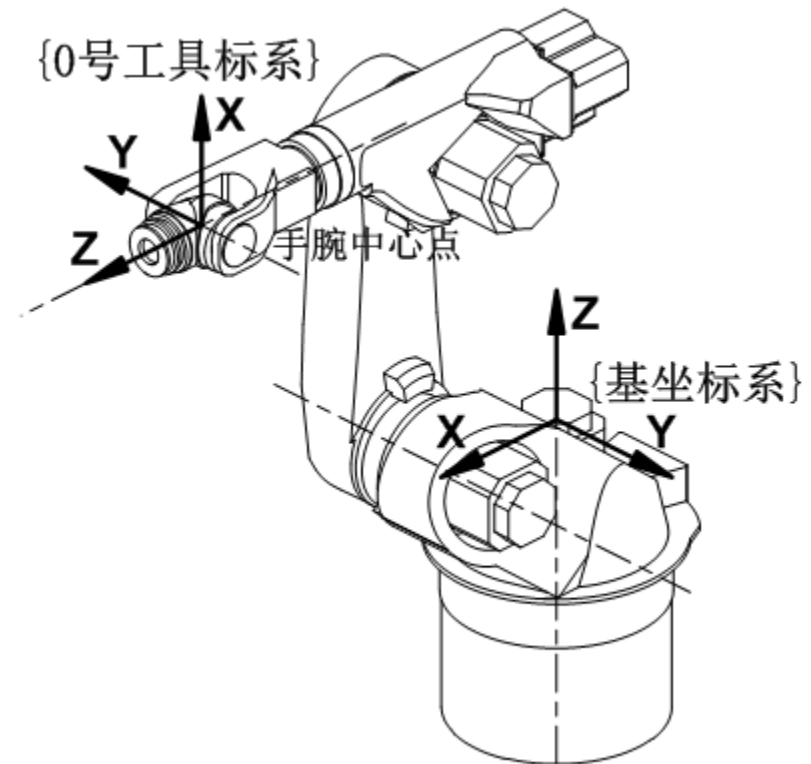


图1-2 六轴机器人参考坐标系



- 机器人默认坐标系:

机器人默认坐标系是一个笛卡尔坐标系，固定位于机器人底部(如图)。它可以根据世界坐标系说明机器人的位置。

- 世界坐标系:

世界坐标系是一个固定的笛卡尔坐标系，是用于机器人默认坐标系和基坐标系的原点坐标系。在默认配置中，世界坐标系与机器人默认坐标系是一致的。

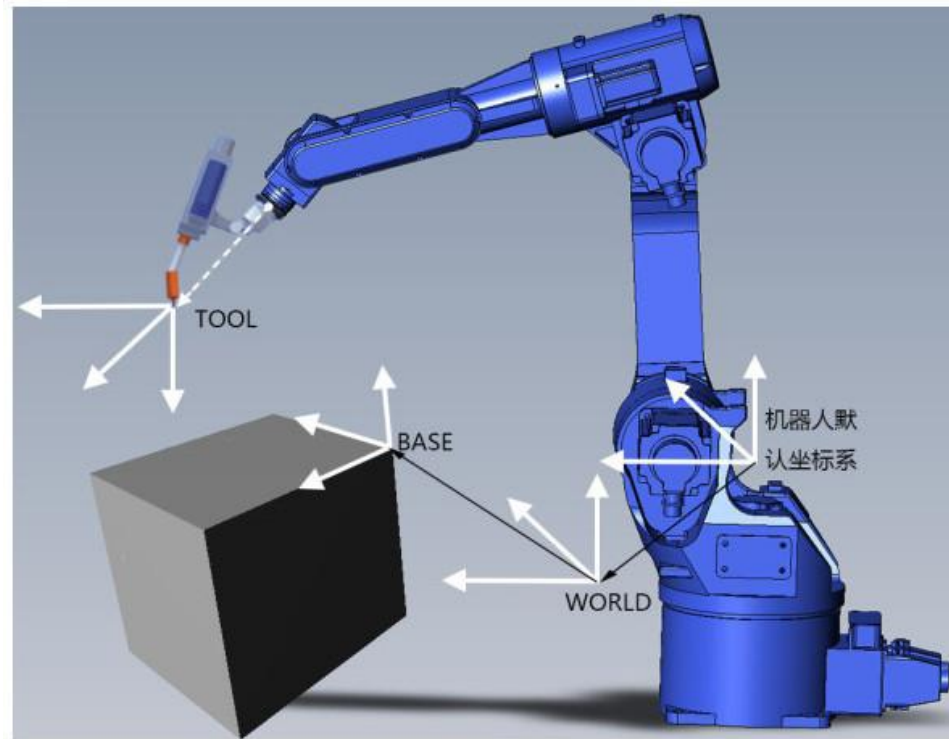


图1-3 六轴机器人参考坐标系





### ● 基坐标系:

基坐标系是一个笛卡尔坐标系，用来说明**工件**的位置。默认配置中，基础坐标系与机器人默认坐标系是一致的。修改基坐标系后，机器人即按照设置的坐标系运动。

### ● 工具坐标系:

工具坐标系是一个笛卡尔坐标系，位于**工具**的工作点中。在默认配置中，工具坐标系的原点在法兰中心点上。工具坐标系由用户移入工具的工作点。

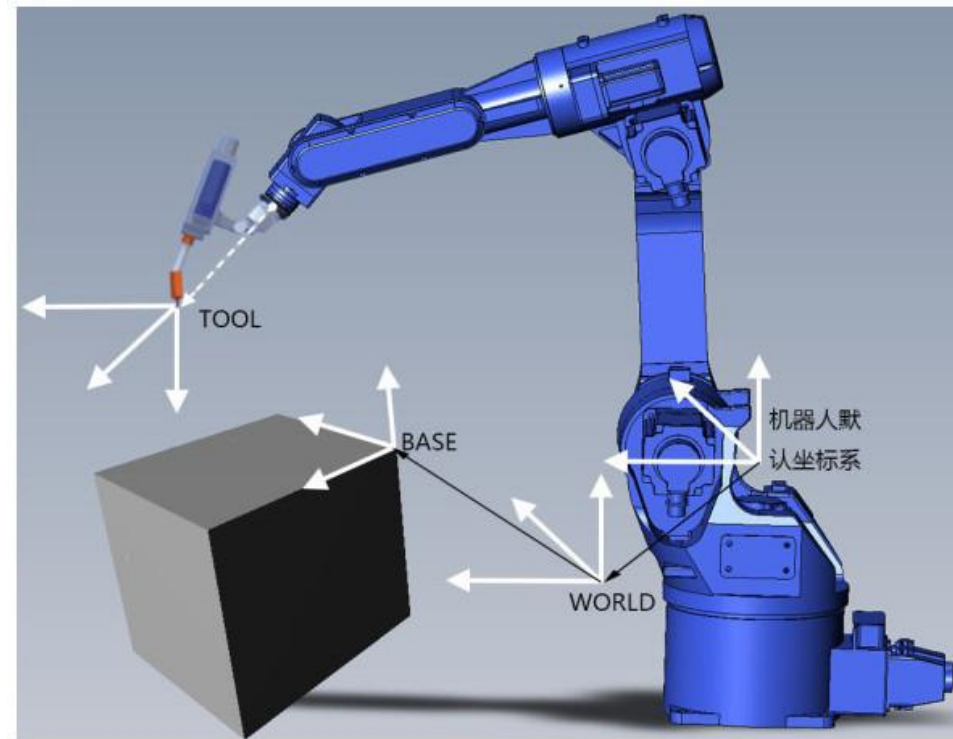


图1-3 六轴机器人参考坐标系

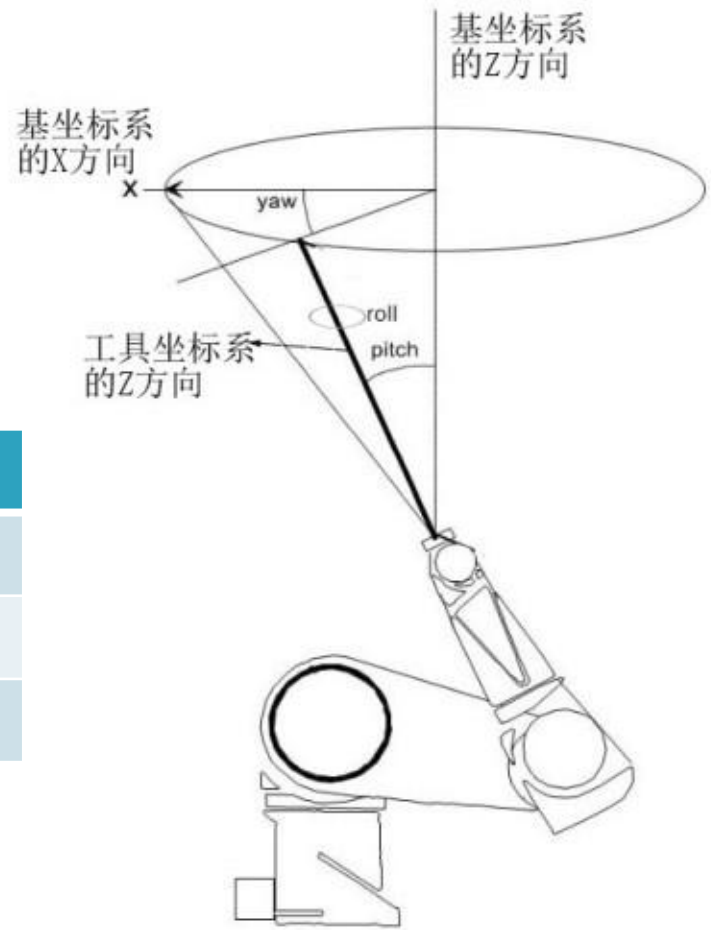




机器人坐标系的姿态角：

HSpad 使用姿态角来描述工具点的姿态。

转角	含义
A(Y)	Yaw 偏航角
B(P)	Pitch 俯仰角
C(R)	Roll 滚转角



其实就A,B,C就是机器人分别关于X,Y,Z轴的  
旋转角度。

图1-4 姿态角示意





## 二、HSpad操作界面

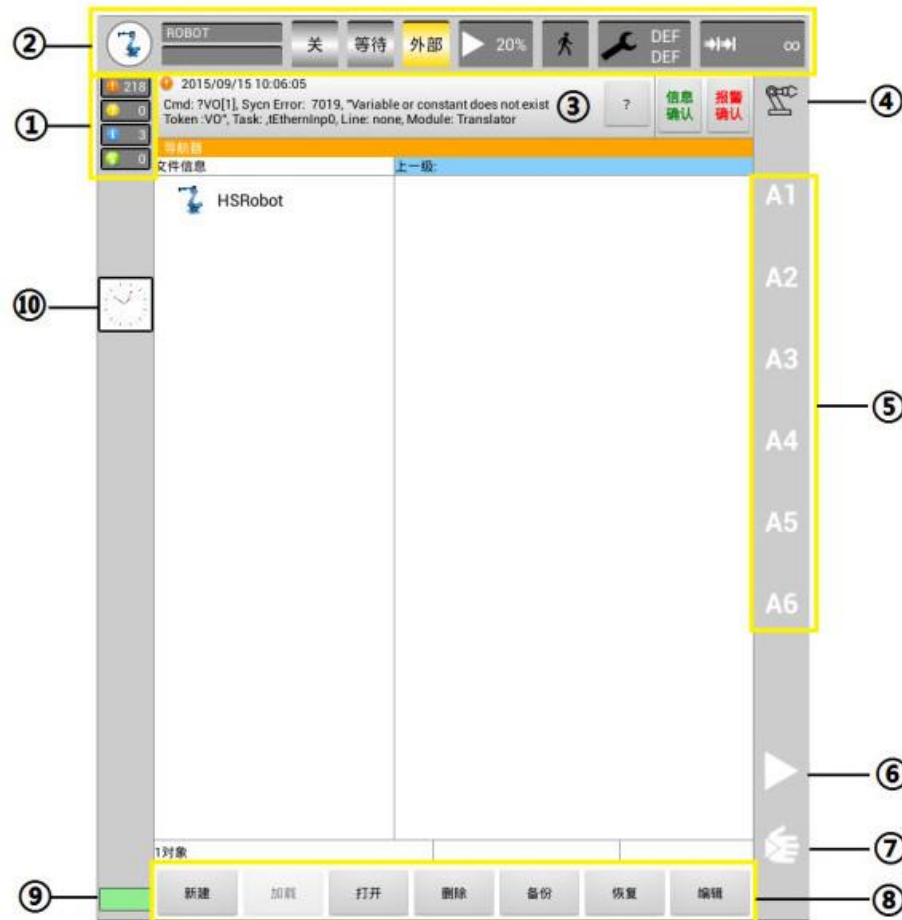
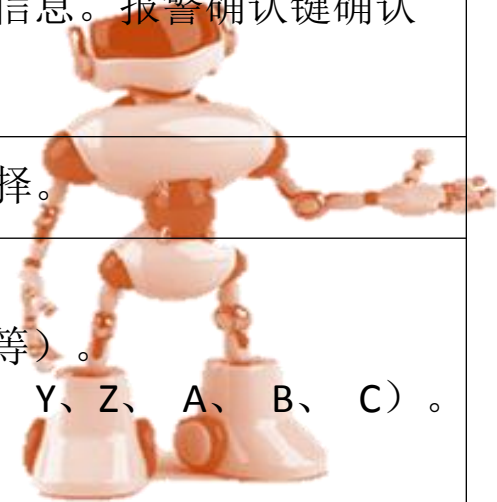


图2-1 HSpad操作界面





标签项	说明
1	<p>信息提示计数器。 信息提示计数器显示，提示每种信息类型各有多少条等待处理。触摸信息提示计数器可放大显示。</p>
2	<p>状态栏。</p>
3	<p>信息窗口。 根据默认设置将只显示最后一个信息提示。 触摸信息窗口可显示信息列表。列表中会显示所有待处理的信息。可以被确认的信息可用确认键确认。信息确认键确认所有除错误信息以外的信息。报警确认键确认所有错误信息。 ? 按键可显示当前信息的详细信息。</p>
4	<p>坐标系状态。触摸该图标就可以显示所有坐标系，并进行选择。</p>
5	<p>点动运行指示。 如果选择了与轴相关的运行，这里将显示轴号（A1、A2 等）。 如果选择了笛卡尔式运行，这里将显示坐标系的方向（X、Y、Z、A、B、C）。 触摸图标会显示运动系统组选择窗口。 选择组后，将显示为相应组中所对应的名称。</p>





标签项	说明
6	自动倍率修调图标。
7	手动倍率修调图标。
8	操作菜单栏。 用于程序文件的相关操作。
9	<p>网络状态。</p> <p>红色为网络连接错误，检查网络线路问题。</p> <p>黄色为网络连接成功，但初始化控制器未完成，无法控制机器人运动。</p> <p>绿色为网络初始化成功，HSpad 正常连接控制器，可控制机器人运动。</p>
10	<p>时钟。</p> <p>时钟可显示系统时间。 点击时钟图标就会以数码形式显示系统时间和当前系统的运行时间。</p>

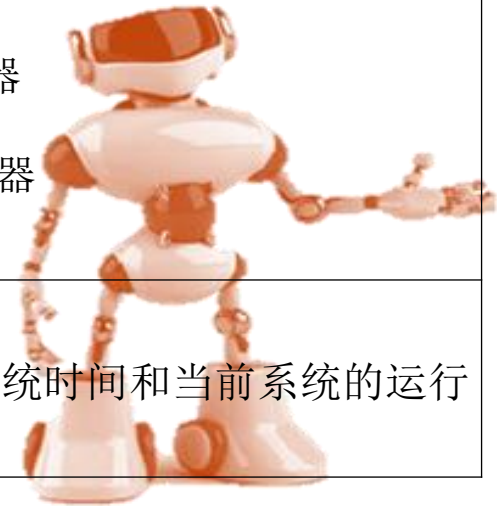


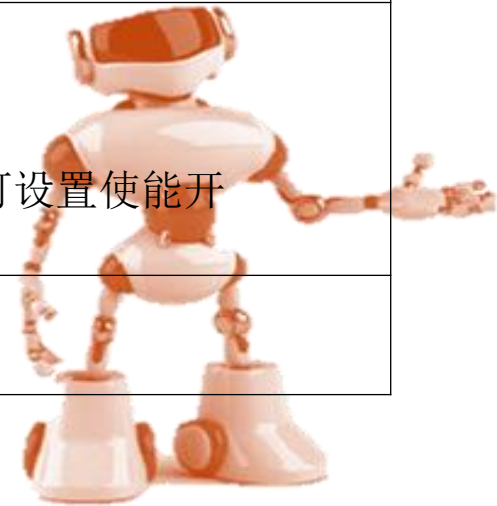


图2-2 HSpad状态栏





标签项	说明
1	菜单键。 功能同菜单按键功能。
2	机器人名。 显示当前机器人的名称。
3	加载程序名称。 在加载程序之后，会显示当前加载的程序名。
4	使能状态。 绿色并且显示“开”，表示当前使能打开。 红色并且显示“关”，表示当前使能关闭。 点击可打开使能设置窗口，在自动模式下点击开/关可设置使能开关状态。窗口中可显示安全开关的按下状态。
5	程序运行状态。 自动运行时，显示当前程序的运行状态。





标签项	说明
6	模式状态显示。 模式可以通过钥匙开关设置，模式可设置为手动模式、自动模式、外部模式。
7	倍率修调显示。 切换模式时会显示当前模式的倍率修调值。 触摸会打开设置窗口，可通过加/减键以 1% 的单位进行加减设置，也可通过滑块左右拖动设置。
8	程序运行方式状态。 在自动运行模式下只能是连续运行，手动 T1 和手动 T2 模式下可设置为单步或连续运行。 触摸会打开设置窗口，在手动 T1 和手动 T2 模式下可点击连续/单步按钮进行运行方式切换。
9	激活基坐标/工具显示。 触摸会打开窗口，点击工具和基坐标选择相应的工具和基坐标进行设置。
10	增量模式显示。 在手动 T1 或者手动 T2 模式下触摸可打开窗口，点击相应的选项设置增量模式。







### 三、手动操作

手动运行机器人分为 2 种方式：

#### 1. 笛卡尔式运行

TCP 沿着一个坐标系的正向或反向运行。

#### 2. 与轴相关的运行

每个轴均可以独立地正向或反向运行。

可用于运行机器人的原件是：轴运行按键。

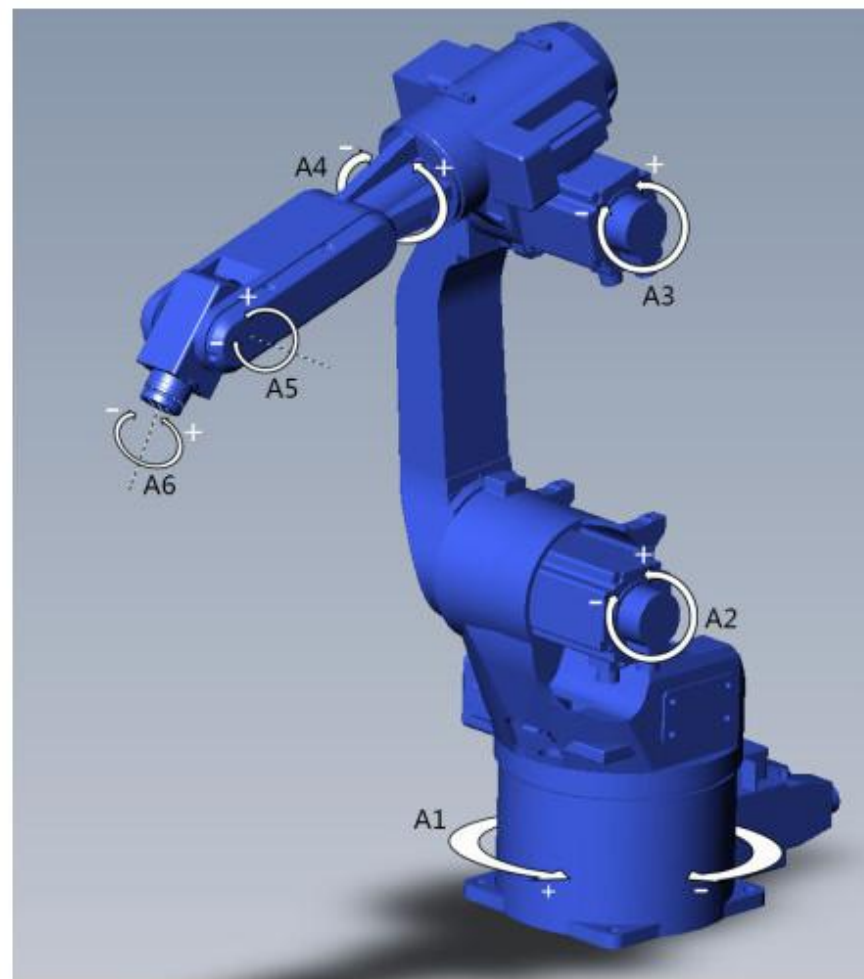


图3-1 机器人轴方向



### 3.1 轴操作

手动 T1 模式速度为 125mm/s,

手动 T2 模式速度为250mm/s。

选定需要移动的轴，通过“+/-”按钮控制各轴单独正/反方向移动。

正负键：可以以 100%、75%、50%、30%、10%、3%、1% 步距为单位进行设定。

调节器：倍率可以以 1% 步距为单位进行更改。

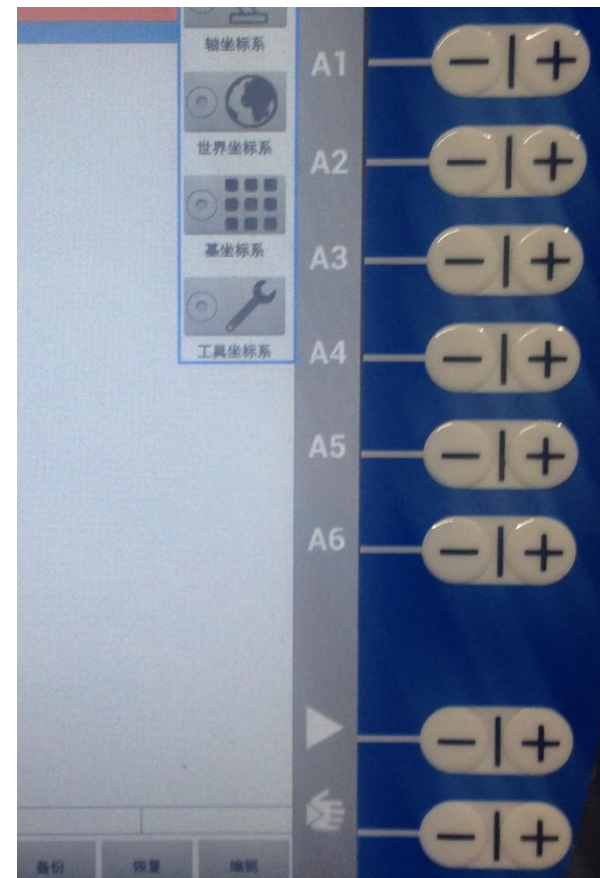


图3-2 轴选操作



## 3.2 工具坐标系

最多可在机器人控制系统中储存 16 个工具坐标系和 16 个基础坐标系。



图3-3 工具坐标系设置窗口



### 3.3 坐标模式

机器人控制系统有四种坐标模式，分别为轴坐标系、基坐标系、工具坐标系、世界坐标系。

点击“轴坐标系”可选择坐标类型，如下图。

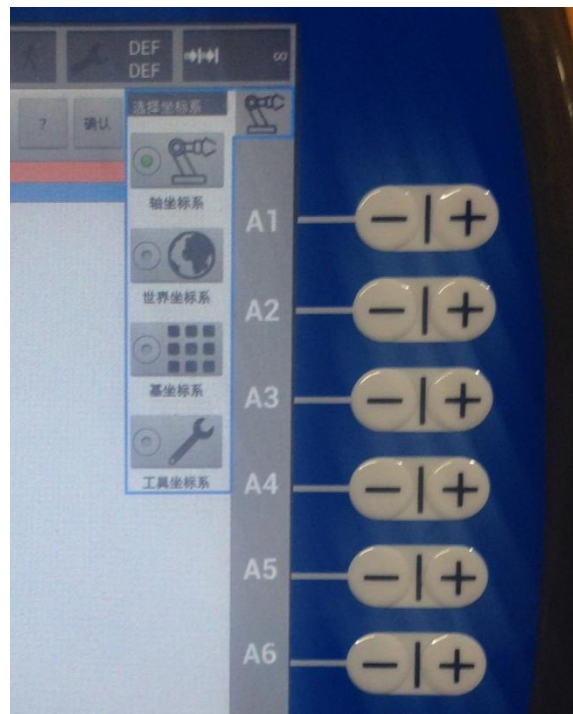


图 3-4 坐标模式选择







### 3.4 校准



图 3-5 校准界面

在手动模式下控制机器人各关节轴移动至标准零点姿态；然后，在下图的校准界面中输入各关节轴的零点值（如轴一到轴六分别为 0, -90, 180, 0, 90, 0 或者 0, -90, 180, 0, 0, 0）；最后，按下确认键，完成校准，并重启系统生效。





### 3.5 工具坐标系标定

机器人控制系统支持16个工具坐标系设定。点击“工具坐标设定”，可设置相应工具坐标系或工件坐标系的各个坐标值，如下图所示。

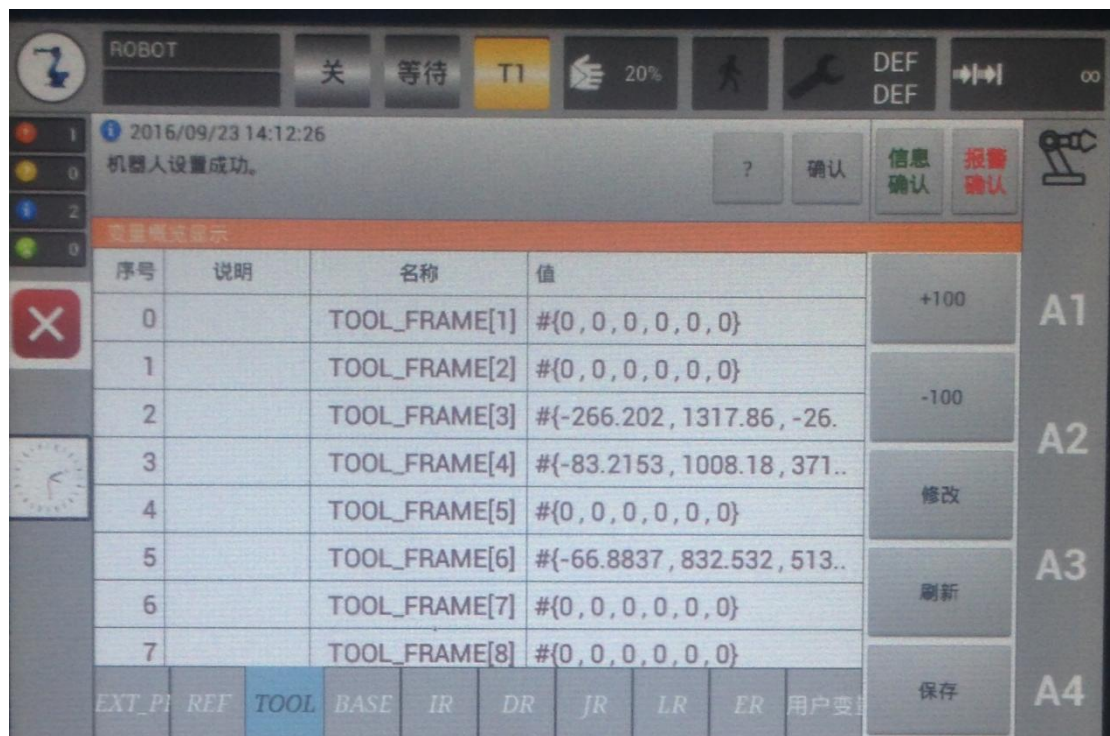


图 3-6 工具坐标系设定界面





### 3.5.1 四点法标定

通过标定空间中机器人末端在坐标系中的四个不同位置来计算工具坐标系。

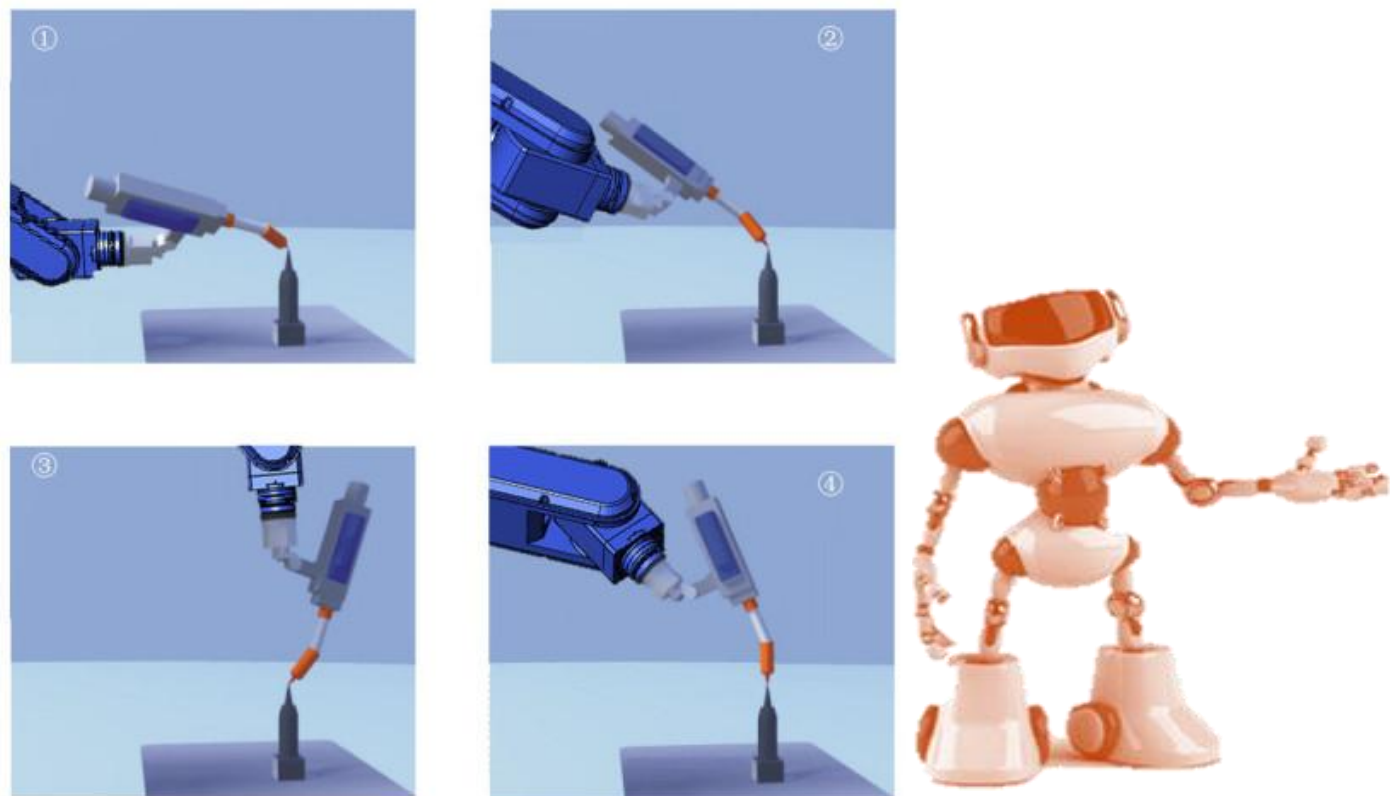


图 3-7 4点标定图示



工具坐标系四点标定操作步骤如下：

(1) 在菜单中选择投入运行->测量->工具->4 点法，为待测量的工具输入工具号和名称。点击“继续”键确认。

4点法

工具号:

工具名:

选定待测的工具

X[mm]:	0.00	A[°]:	0.00
Y[mm]:	0.00	B[°]:	0.00
Z[mm]:	0.00	C[°]:	0.00

返回 继续

图 3-8 工具号输入



(2) 用 TCP 移至任意一个参照点，点击记录。点击“确定”键确认。



图 3-9 记录第一个位置

(3) 将步骤 2 重复 3 次。





(3) 点击保存，数据被保存，窗口关闭。

工具号:	1		
工具名:	bg		
点击“保存”，保存应用数据			
X[mm]:	0.00	A[°]:	0.00
Y[mm]:	0.00	B[°]:	0.00
Z[mm]:	0.00	C[°]:	0.00
返回 保存			

图 3-10 接近点1指定完成





### 3.6 工件坐标系设定(基坐标系)

工件坐标系是由用户在工件空间定义的一个笛卡尔坐标系。工件坐标包括:  $(X,Y,Z)$  用来表示距原点的位置,  $(A,B,C)$  用来表示绕 X-, Y-, Z-轴旋转的角度。

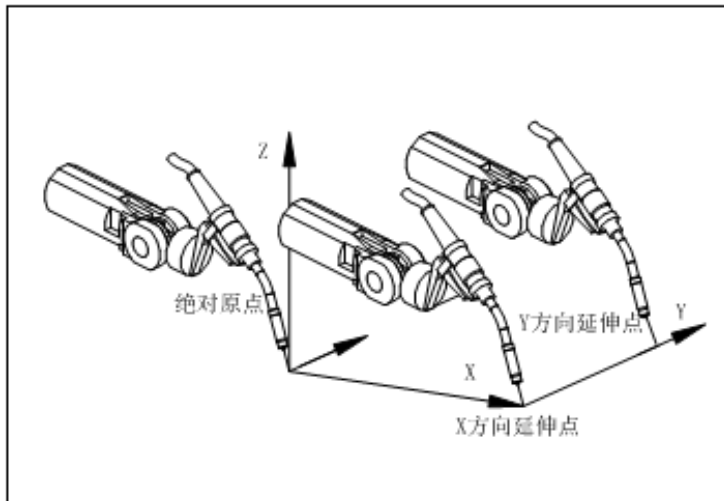


图 3-11 工件三点法标定

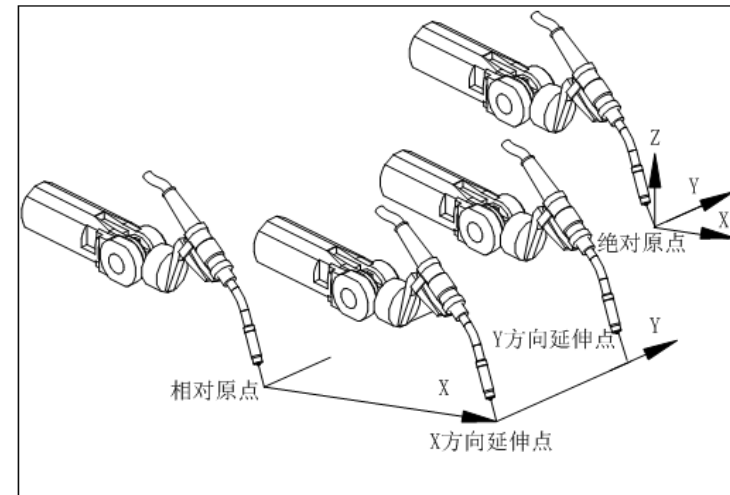


图 3-12 工件四点法标定



## 四、示教

本章介绍一些示教界面中指令的基本操作方法，详细的编程过程请参照《HSpad使用说明书》的指令编程章节。

程序的基本信息包括：程序名、程序类型、程序指令。

(1) 程序名：用以识别存入控制器内存中的程序，在同一个目录下不能包含两个或更多拥有相同程序名的程序。程序名长度不超过8个字符，由字母、数字、下划线(\_)组成。

(2) 程序类型：用于设置程序文件的类型。目前本系统支持的机器人程序类型有.PRG格式和.LIB格式。

(3) 程序指令：包括运动指令、程序指令等示教中涉及的所有指令。







## 4.1 新建程序

点击示教界面下方左侧的“新建程序”按钮，在弹出的对话框中输入程序名，可新建一个空的程序文件，如下图所示。

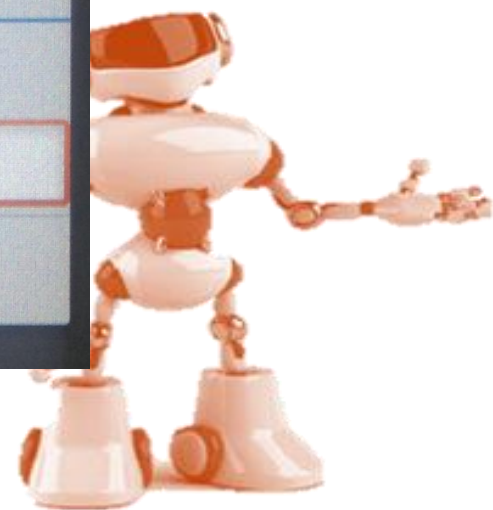
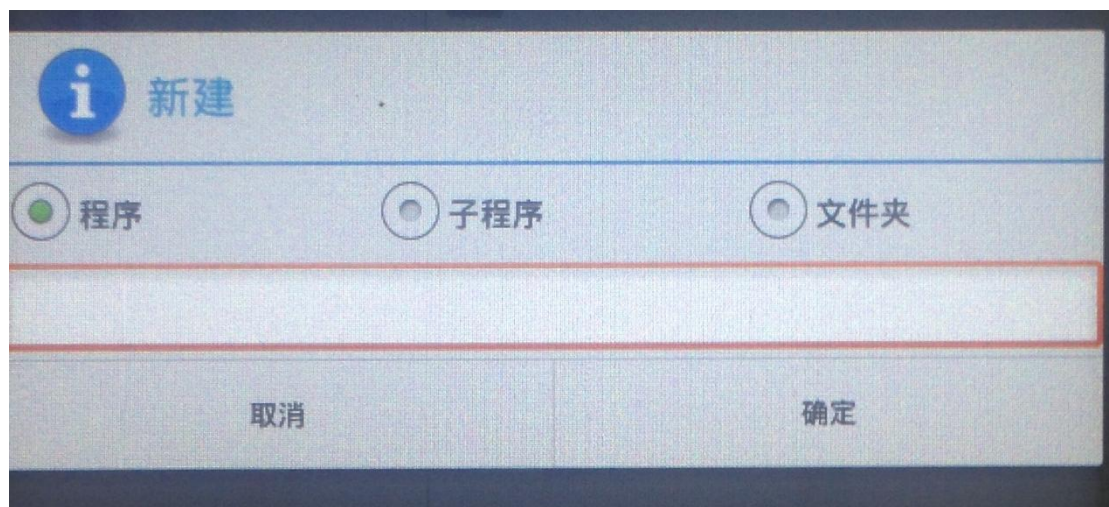


图 4-1 新建程序的示教窗口



## 4.2 打开程序

打开程序对话框可查看系统中所有的程序文件及其属性，点击“打开程序”，可显示图4-2所示程序文件列表，选择一个现有的程序文件并点击“确认”后可加载该选中的程序文件。

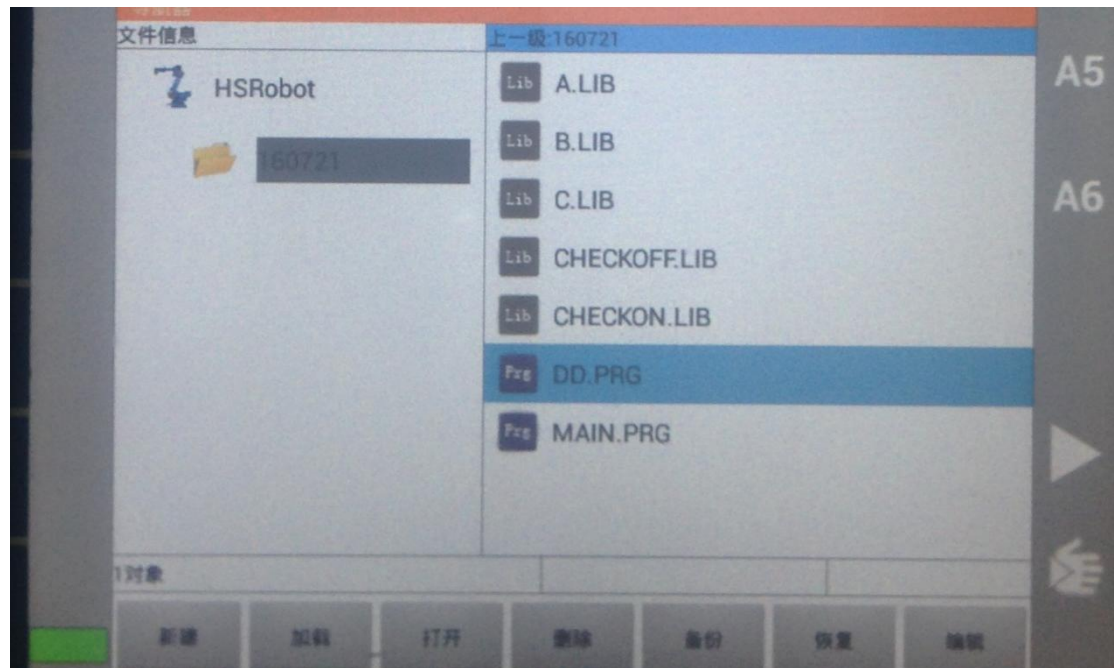


图 4-2 打开程序的示教窗口



## 4.3 程序修改

打开程序后，选定需要修改的程序行，  
点击屏幕左下角的“更改”按钮进行修改。

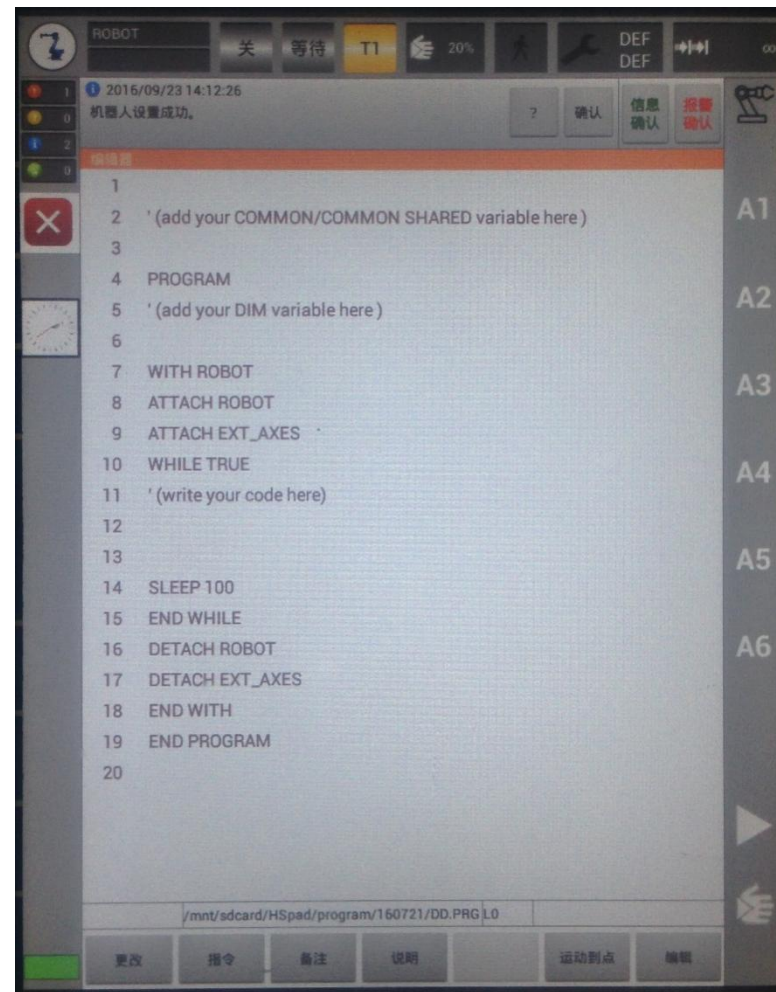


图 4-3 程序编辑页面



## 4.4 编程指令类型说明

### 4.4.1 运动指令

运动指令类型包括三种：关节定位（MOVE）、直线定位（MOVES）、圆弧定位（CIRCLE）。

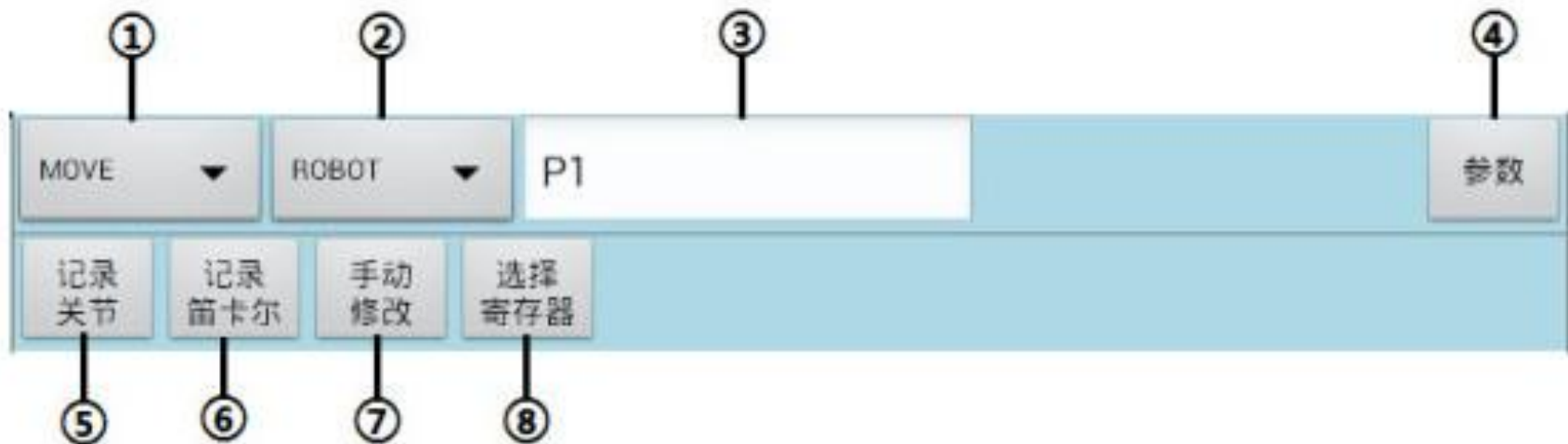


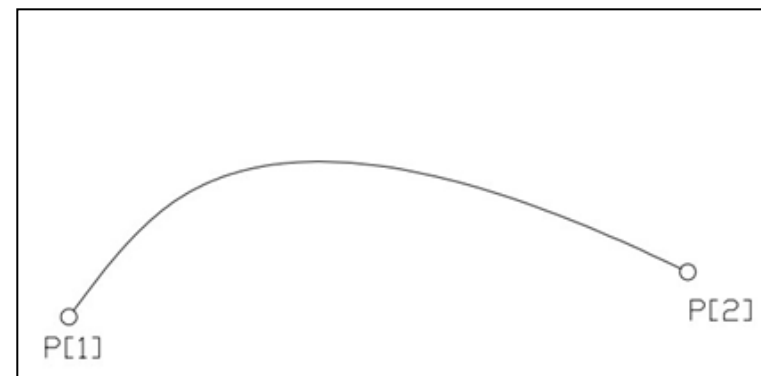
图 4-4 运动指令



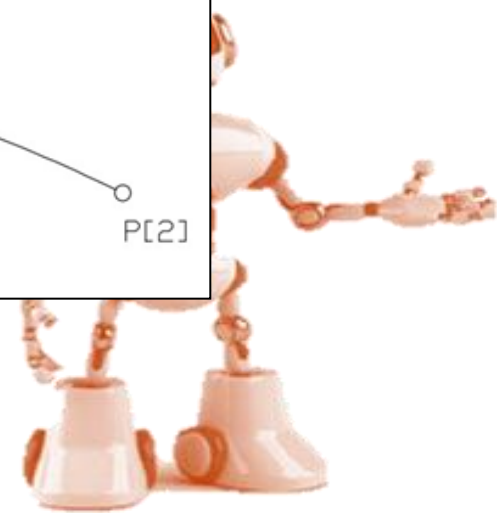
### 4.4.2 动作类型

(1) 关节定位MOVE:是移动机器人各关节到达指定位置的基本动作模式。工具的运动路径通常是非线性的，在两个指定的点之间任意运动。

```
1:MOVE ROBOT P1  
2:MOVE ROBOT P2
```



P[1]点采用关节定位方式移动至P[2]点。

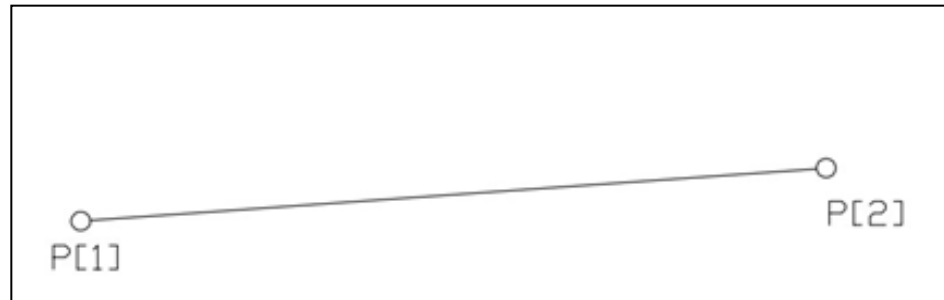






(2) 直线定位指令MOVES:控制TCP（工具中心点）沿直线轨迹运动到目标位置，通过区别起点和终点时的姿态，来控制被驱动的工具的姿态。

```
1:MOVE ROBOT P1  
2:MOVES ROBOT P2
```



P[1]点采用直线运动方式移动至P[2]点。

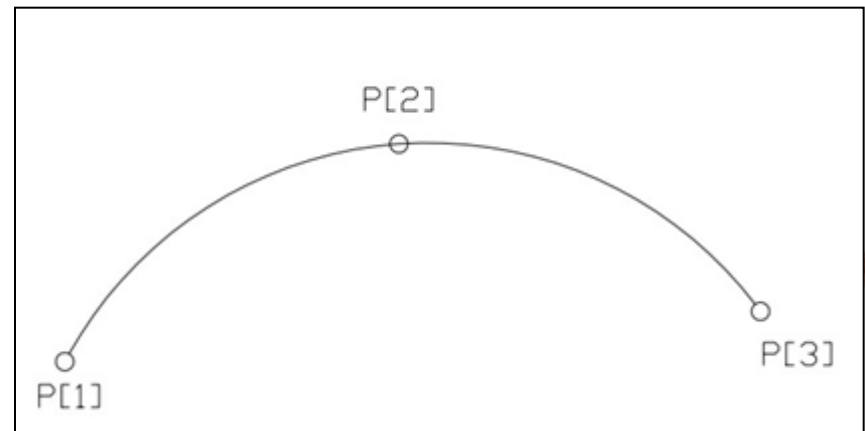






(3) 圆弧定位指令**CIRCLE**: 控制TCP (工具中心点) 沿圆弧轨迹从起始点经过中间点移动到目标位置, 中间点和目标点在指令中一并给出。其速度由程序指令直接指定, 单位可为mm/sec、cm/min、inch/min。通过区别起点和终点时的姿态, 来控制被驱动的工具的姿态。

```
1:MOVE ROBOT P1  
2:CIRCLE ROBOT CIRCLEPOINT=P2 TARGETPOINT=P3
```



P[1]点开始沿着过P[2]点的圆弧运动至P[3]点





### 操作步骤

1. 标定需要输入程序行的上一行
2. 选择指令→运动指令→MOVE/MOVES/CIRCLE
3. 选择机器人轴或者附加轴
4. 记录点位并配置好参数
5. 手动移动机器人到需要的位置或姿态
6. 选中输入框后，点击记录关节或笛卡尔坐标
7. 点击确认，完成指令添加





## 运动参数

名称	说明	备注
VCRUISE	速度 (大于 0)	用于 MOVE
ACC	加速比 (大于 0)	用于 MOVE
DEC	减速比 (大于 0)	用于 MOVE
VTRAN	速度 (大于 0)	用于 MOVES
ATRAN	加速比 (大于 0)	用于 MOVES
DTRAN	减速比 (大于 0)	用于 MOVES
ABS	1-绝对运动,0-相对运动	





### 练一练：

机器人以关节定位移动到P1,然后画圆弧经过P2和P3点,再以直线定位移动到P4,接着同样以直线定位移动到P5,然后继续画圆弧,经过P6,最后到达P1。



```
MOVE ROBOT P1
```

```
CIRCLE ROBOT CIRCLEPOINT=P2 TARGETPOINT=P3
```

```
MOVES ROBOT P4
```

```
MOVES ROBOT P5
```

```
CIRCLE ROBOT CIRCLEPOINT=P6 TARGETPOINT=P1
```



### 4.4.3 条件指令

条件指令由IF开头，用于比较判断是否满足条件，若满足则执行下一行指令。支持的比较运算符有：>、>=、=、<=、<、<>，还可以使用逻辑与（AND）和逻辑或（OR）指令对这些条件语句进行运算，条件结束指令END IF。

```
12  
13 IF D_IN[15]=ON THEN  
14 CALL DENG  
15 END IF  
16
```



图 4-5 条件指令



### 4.4.3 条件指令

```
IF LR[21]=LR[2] AND LR[32]<>LR[4] THEN  
MOVE ROBOT JR[1] VCRUISE=100  
DELAY ROBOT 1000  
END IF
```

注意：IF和END IF必须联合使用，将条件运行程序块置于两条指令之间。







### 4.4.4 流程指令

用于在主程序中添加子程序，关系到程序执行流程。

子程序相关指令：SUB、PUBLIC SUB、END SUB、FUNCTION、PUBLIC FUNCTION、END FUCTION。

子程序跳转调用相关指令：CALL、GOTO、LABEL。





## 操作步骤

1. 选定需要添加指令的前一行
2. 在指令→流程指令中选择相应的写子程序相关指令
3. 点击操作栏中的确定按钮，添加写子程序完成

指令	
运动指令 ▶	SUB
条件指令 ▶	PUBLIC SUB
<b>流程指令 ▶</b>	END SUB
程序控制 ▶	FUNCTION
延时指令 ▶	PUBLIC FU...
循环指令 ▶	END FUNCT.
IO指令 ▶	CALL
变量 ▶	GOTO
坐标系指令 ▶	LABEL
修调指令 ▶	
同步指令 ▶	
寄存器指令 ▶	
事件指令 ▶	
异常指令 ▶	
手动指令	

图 4-6 流程控制指令





## 指令说明

指令	说明
SUB	写子程序，该子程序没有返回值，只能在本程序中调用。
PUBLIC SUB	写子程序，该子程序没有返回值，能在程序以外的其他地方被调用。
END SUB	写子程序结束。
FUNCTION	写子程序，该子程序有返回值，只能在本程序中调用。
PUBLIC FUNCTION	写子程序，该子程序有返回值，能在程序以外的其他地方被调用。
END FUCTION	写子程序结束。
CALL	调用子程序

注意：SUB、PUBLIC SUB和END SUB必须联合使用，子程序位于两条指令之间；FUNCTION、PUBLIC FUNCTION和END FUNCTION必须联合使用，子程序位于两条指令之间。





### GOTO指令和LABEL指令

主要用于程序跳转

例如：

**GOTO LABEL 1**,必须联合使用才能实现跳转

### CALL指令

用于调用子程序

例如：

**CALL PICK**





## 4.4.5 程序指令

程序指令新建程序时自动添加到程序文件中，通常情况下，用户无需修改。

```
1  
2 ' (add your COMMON/Common Shared variabl.  
3  
4 PROGRAM  
5 ' (add your DIM variable here )  
6  
7 WITH ROBOT  
8 ATTACH ROBOT  
9 ATTACH EXT_AXES  
10 ' (write your code here)  
11  
12  
13 DETACH ROBOT  
14 DETACH EXT_AXES  
15 END WITH  
16 END PROGRAM  
17
```





### 程序指令

指令	说明
PROGRAM	程序开始
END PROGRAM	程序结束
WITH	引用机器人名称
END WITH	结束引用机器人名称
ATTACH	绑定机器人
DETACH	结束绑定







### 4.4.6 延时指令

机器人控制系统的延时指令包括两种：**DELAY**指令和**SLEEP**指令。

指令**DELAY**——是针对指定的运动对象在运动完成后的延时时间，单位为毫秒。（若当前指定运动对象无运动，则**DELAY**指令无效）

指令**SLEEP**——是针对非运动指令的延时指令，单位为毫秒。

（**SLEEP**指令只对非运动指令生效，对运动指令，**SLEEP**指令无效）





## 4.4.6 延时指令

### DELAY指令和SLEEP指令的用法

在华数II型控制系统中，存在运动指令和非运动指令这两种类型的指令。这两种指令是并行执行的，并非执行完一条再执行下一条。如下例：

```
MOVE ROBOT P1
```

```
D_OUT[30]=ON
```

在这个例子中，第一条为运动指令，第二条为非运动指令，在系统中是并行执行的。也就是说——**机器人还未运动到P1，D\_OUT[30]就有信号输出了**。如何解决这个问题呢？

办法：加入延时指令 该加入DELAY还是SLEEP？





上述例子应该改为：

```
MOVE ROBOT P1  
DELAY ROBOT 200  
D_OUT[30]=ON
```

### **SLEEP**指令的两种应用场合

1.在循环中使用

```
WHILE D_IN[30]<>ON  
SLEEP 100  
END WHILE
```

例中等待D\_IN[30]的信号，若无信号则持续循环，等到信号后向下执行。由于循环中要一直扫描D\_IN[30]的值，为了避免控制器CPU因过载出现异常报警，须加入**SLEEP**指令。





### SLEEP指令的两种应用场合

#### 2.输出脉冲信号

```
D_OUT[30]=ON
```

```
SLEEP 100
```

```
D_OUT[30]=OFF
```

上述例子中，D\_OUT[30]输出了一个宽度为100的脉冲信号。其中必须加入**SLEEP**指令，否则脉冲宽度太短，会导致实际上没有任何脉冲信号输出。





### 4.4.7 循环指令

循环指令用于多次执行WHILE指令与END WHILE指令之间的程序行，WHILE TRUE表示程序循环执行。

```
WHILE LR[20]=LR[21]  
MOVE ROBOT P1  
DELAY ROBOT 1000  
END WHILE
```

注意：WHILE 指令和END WHILE 指令必须联合使用才能完成一个循环体。





### 4.4.8 I/O指令

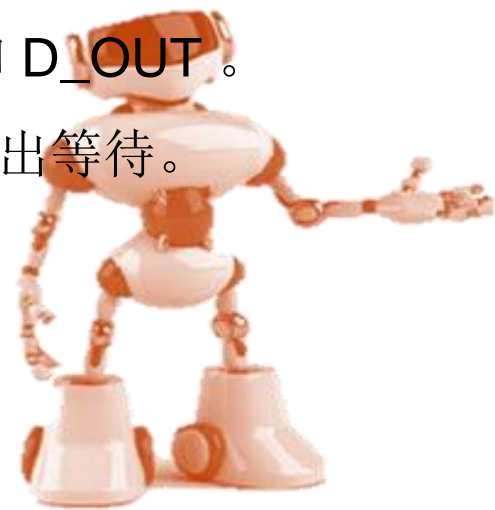
IO 指令包括了 D\_IN 指令、 D\_OUT 指令、 WAIT指令、 WAITUNTIL 指令、 以及PLUSE指令。

D\_IN 指令、 D\_OUT 指令：可用于给当前 IO 赋值为 ON 或者OFF，也可用于在 D\_IN 和 D\_OUT 之间传值。

WAIT指令：用于阻塞等待一个指定IO信号 ,可选D\_IN 和 D\_OUT 。

WAITUNTIL指令：用于等待IO信号，超过设定时限后退出等待。

PLUSE指令：用于产生脉冲。







## 4.4.8 I/O指令

函数	参数说明
WAIT(IO,STATE)	IO 代表 D_IN、D_OUT, STATE 代表 ON、OFF
WAITUNTIL(IO,IO,MIL,FLAG)	IO 代表 D_IN、D_OUT, MIL 代表延时 (单位毫 秒), FLAG 表示等待信 号是否成功
PLUSE(IO,STATE)	IO 代表 D_IN、D_OUT, STATE 代表 ON、OFF





### 4.4.9 变量

变量可分为全局变量COMMON指令和局部变量DIM指令，变量可用于程序中作为程序中的数据运算，若添加SHARED则为共享变量。

变量类型包括：

LONG类型：长型

DOUBLE类型：浮点型

STRING类型：字符型

JOINT类型：关节型

LOCATION类型：位置型

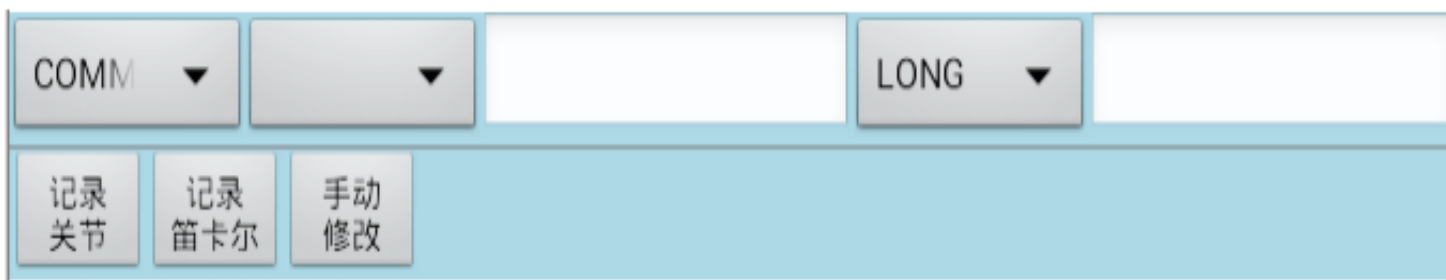
ERROR类型：错误型





## 操作步骤

1. 选定需要添加变量的上一行。
2. 选择指令->变量->全局变量或者局部变量。
3. 在打开的对话框中选择 COMMON 或者 DIM 为全局或者是局部变量。
4. 选择设置该变量是否为 SHARED 属性，然后选择变量类型。
5. 在名字输入框中输入变量的名字，第二个输入框中输入变量的值。
6. 点击操作栏的确定按钮完成变量的添加。



对于JIONT和LOCATION类型的变量，可使用图中最下方的三个按钮来设置变量的坐标值。



### 4.4.10 坐标系指令

坐标系指令分为基坐标系BASE和工具坐标系TOOL。在程序中可选择定义的坐标系编号，在程序中切换坐标系。

### 4.4.11 同步指令

说明

同步指令用于将位于该语句之前的两条指令同时执行。

操作步骤

- 1.添加需要同步的两条指令，如 MOVE P1 和 MOVE P2。
- 2.选定 MOVE P2 指令。
- 3.选择指令->同步指令->SYNCALL。
- 4.点击操作栏中的确定按钮完成同步指令的添加。





### 4.4.12 寄存器指令

用于添加寄存器，以及使用寄存器进行运算操作。

寄存器设置格式为：目的寄存器=操作数1+操作数2+...+操作数N

注：其中操作数可以为寄存器，也可以为数值。

寄存器的类型：

LR：笛卡尔位置寄存器

JR：关节位置寄存器

DR：浮点型数值寄存器

IR； 整型数值寄存器

SAVE：保存寄存器的值





### 4.4.13 事件指令

用于添加事件。事件用于某个条件触发后执行相应的代码。指令包括：

- ONEVENTN 事件 条件
  - 条件触发后，进入事件处理的起始位置。
- END ONEVENT
  - 事件处理结束。
- EVENT ON 事件
  - 开启事件，开启后，一旦条件触发，即会进入 ONEVENT 处执行。
- EVENT OFF 事件
  - 关闭事件。







#### 4.4.14 其他指令

其他指令包含一些扩展指令，例如视觉指令、弧焊指令。这个要根据机器人外围设备来使用，若无外围设备，则指令无效。

VISION: 视觉指令，识别物件

ARC\_ON: 起弧

SWITCH: 焊接

ARC\_OFF: 收弧

CHANNEL: 切换焊机通道，这里需要填写通道号

```
19 CALL GET_VISION_DATA
```

```
20 CALL ARC_ON
```

```
21 CALL ARC_SWITCH
```

```
22 CALL ARC_OFF
```

```
23 CALL ARCCHANNEL(1)
```



..



## 五、记录位置坐标

记录坐标位置可以分为两类：

1. 点位坐标P[i]
2. 位置寄存器JR或者LR等。

记录点位坐标P[i]可以在添加运动指令的界面通过下方按钮进行记录及修改，如下图所示：





## 五、记录位置坐标

记录寄存器坐标如LR[2]不能在添加运动指令的界面通过下方按钮进行记录及修改，而是必须在菜单栏→显示→变量列表→LR中找到LR[2]并选中，然后点击修改，才能进行位置标定。

变量概览显示								
序号	说明	名称	值					
0		LR[1]		+100				
1		LR[2]	#{22,40,1.	-100				
2		LR[3]		-100				
3		LR[4]						
4		LR[5]		修改				
5		LR[6]						
EXT	REF	TOOLBASE	IR	DR	JR	LR	用户	刷新

### 修改坐标

关节  笛卡尔

获取坐标 移动到点

名称 LR[2]

说明

X 22

Y 40

Z 140

A 0

B 0

C 0

取消 确定



## 六、自动运行

在自动操作模式下可以运行机器人程序，任何程序都必须先加载到内存中才能运行。

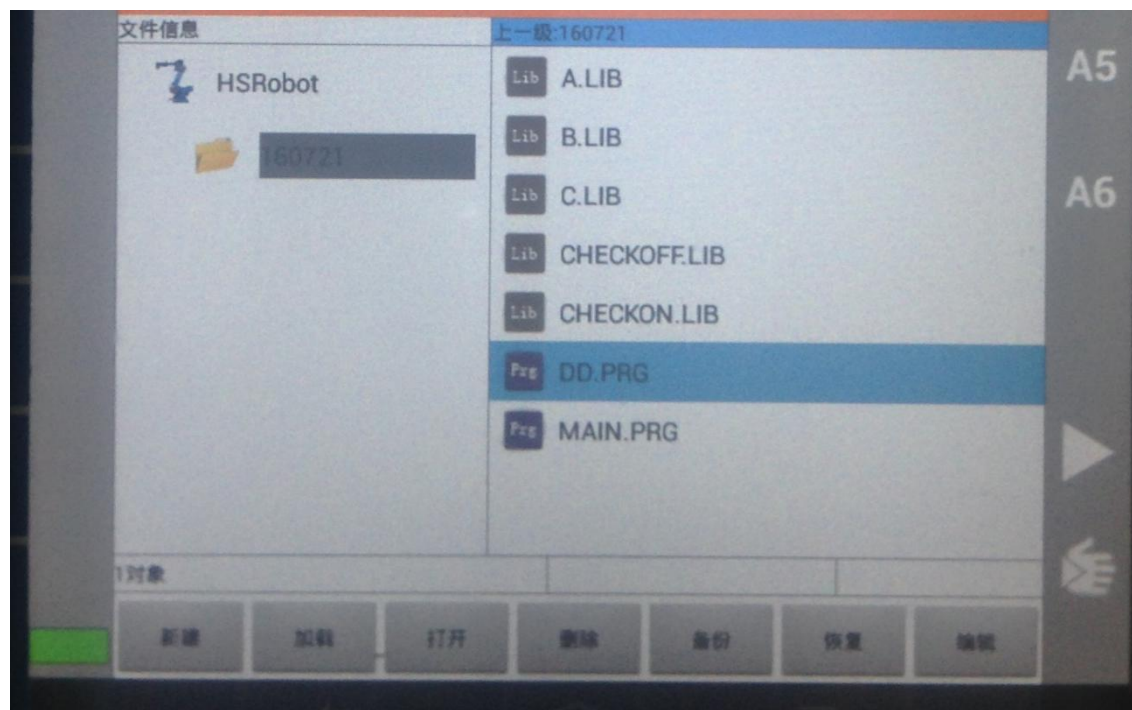


图 5-1 自动运行界面





- (1) 通过示教器上方“运行方式”切换开关，切换选择自动模式。
- (2) 选中需要被加载的程序，点击左下角“加载”按钮。
- (3) 点击使能设置窗口，点击“开/关”设置使能状态为开。



图标	颜色	信息	说明
	绿色	开	已打开使能
	灰色	关	已关闭使能





(4) 该界面中的“启动/暂停”按钮和“停止”按钮可控制程序运行的启停。

“连续/单步”按钮可设置程序自动运行的方式。

选择单步运行模式，系统会在运行完一行程序后停止；若为连续运行，则系统连续运行完程序。







谢谢！

