实训项目 8 机床上下料

1、在 ROBOTSTUDIO 中进行机器人及周边设备的合理布局。
2、在 ROBOTSTUDIO 中的基本工具的使用
3、Smart 组件的基本使用
4、机器人 IO 信号的设定与链接
5、机器人轨迹的创建
6、仿真的调试。

机器人工作站的布局

所有的部件已包含在打包文件中。双击打开打包文件后，请按照以下的图 1 中所示进行布局。

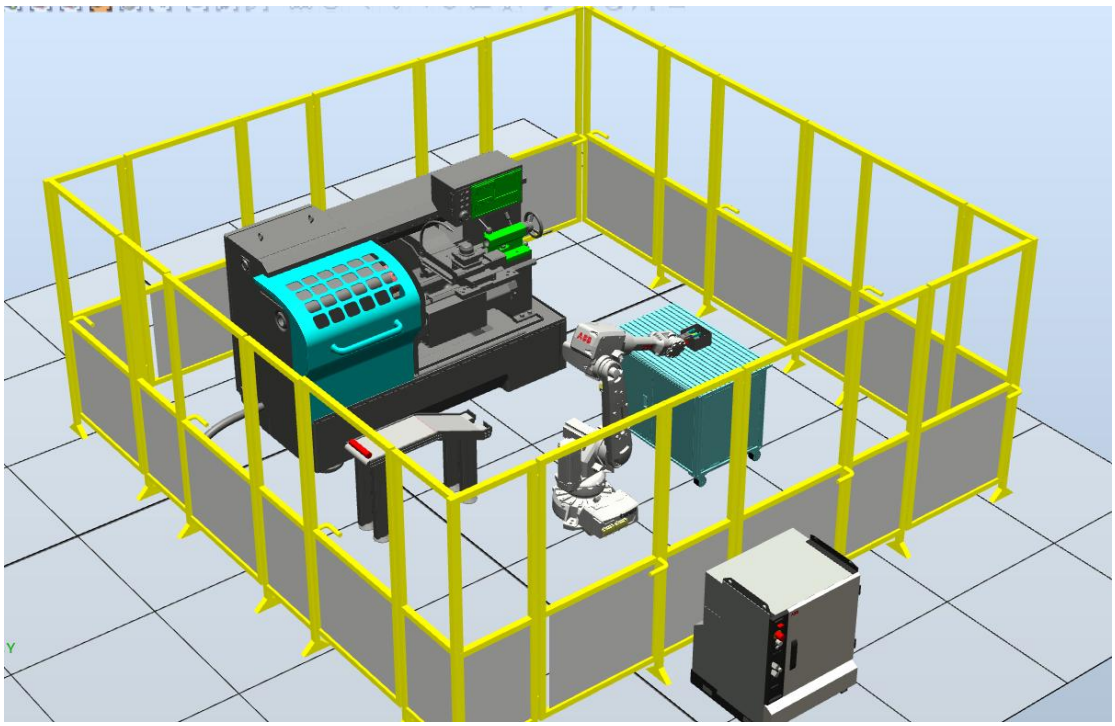


图 1 机器人工作站的布局示意

要注意的问题：

1、机器人与周边的部件的位置要合理，周边部件应在机器人的工作范围的中间位置为佳。

2、可以对机器人的操作，以确认机器人可以到达要取、放的最远端是可以顺利到达的，否则以后再调整就会很麻烦了。

Smart 组件的基本使用

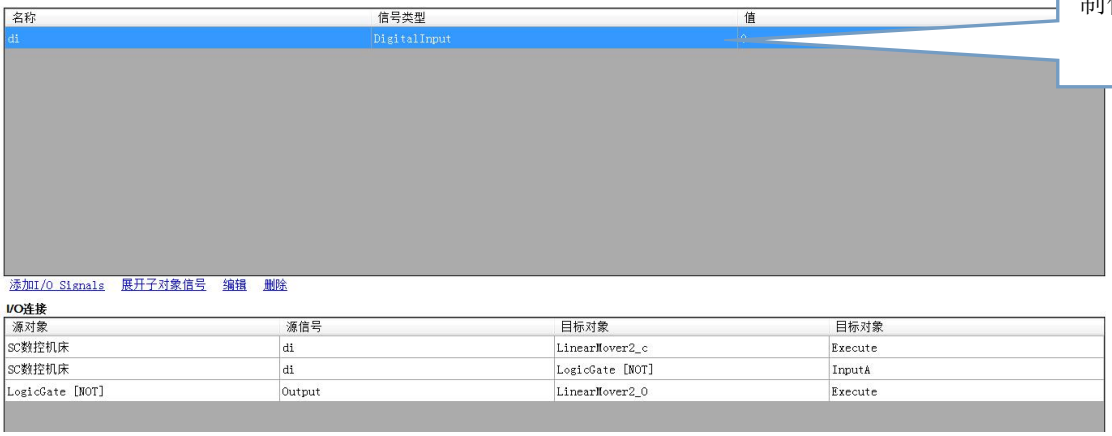
smart 组件创建操作：建模---smart 组件---添加组件，建立如下表组件，详情看视频。

1、添加左边三个组件

2、设置第一个组件的属性

3、设置第二个组件的属性

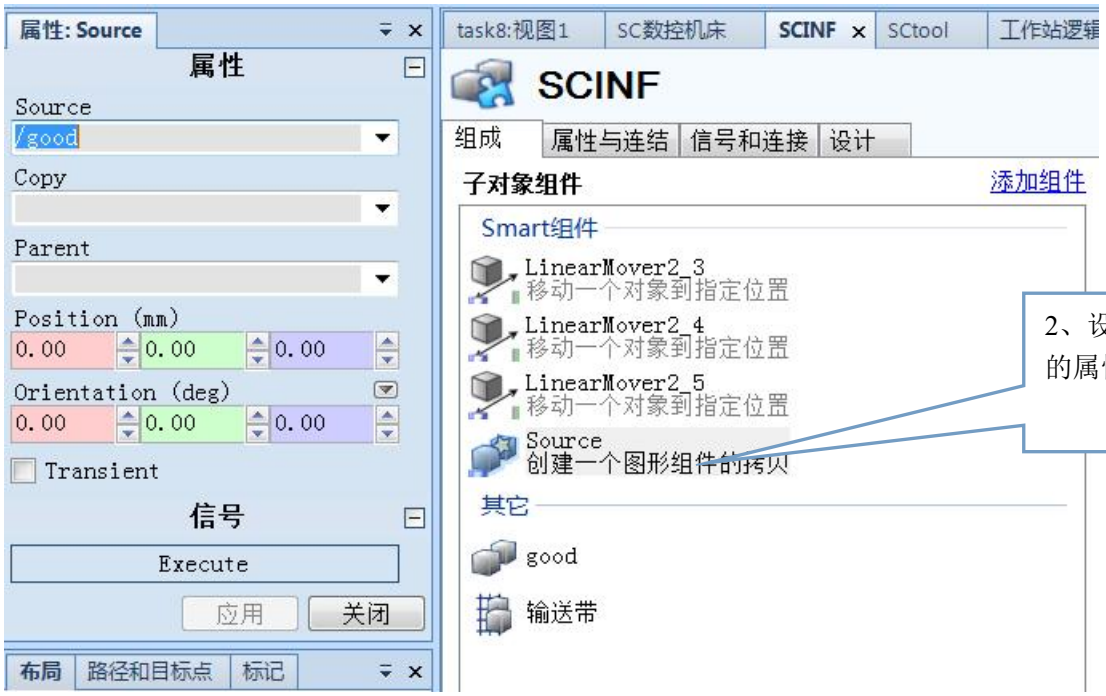
4、设置第三个组件的属性



5、新建一个控制信号

6、进行 I/O 信号连接

1、新建左边四个组件



2、设置“source”的属性


属性: LinearMover2_3
属性
Object
Direction (mm)
0.00
1.00
0.00
Distance (mm)
575.00
Duration (s)
2.0
Reference
Global
信号
Execute
Executing
应用
关闭
task8:视图1
SC数控机床
SCINF
SCtool
工作站逻辑
SCINF
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_3
移动一个对象到指定位置
LinearMover2_4
移动一个对象到指定位置
LinearMover2_5
移动一个对象到指定位置
Source
创建一个图形组件的拷贝
其它
good
输送带
3、设置“LM2_3”组件的属性


属性: LinearMover2_4
属性
Object
Direction (mm)
0.00
1.00
-0.50
Distance (mm)
335.00
Duration (s)
1.5
Reference
Global
信号
Execute
Executing
应用
关闭
task8:视图1
SC数控机床
SCINF
SCtool
工作站逻辑
SCINF
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_3
移动一个对象到指定位置
LinearMover2_4
移动一个对象到指定位置
LinearMover2_5
移动一个对象到指定位置
Source
创建一个图形组件的拷贝
其它
good
输送带
4、设置“LM2_4”组件的属性

5、设置“LM2_5”组件的属性

6、设置属性链接

属性连结

源对象	源属性	目标对象	目标属性
Source	Copy	LinearMover2_3	Object
LinearMover2_3	Object	LinearMover2_4	Object
LinearMover2_4	Object	LinearMover2_5	Object

I/O 信号

名称	信号类型	值
DI1	DigitalInput	0

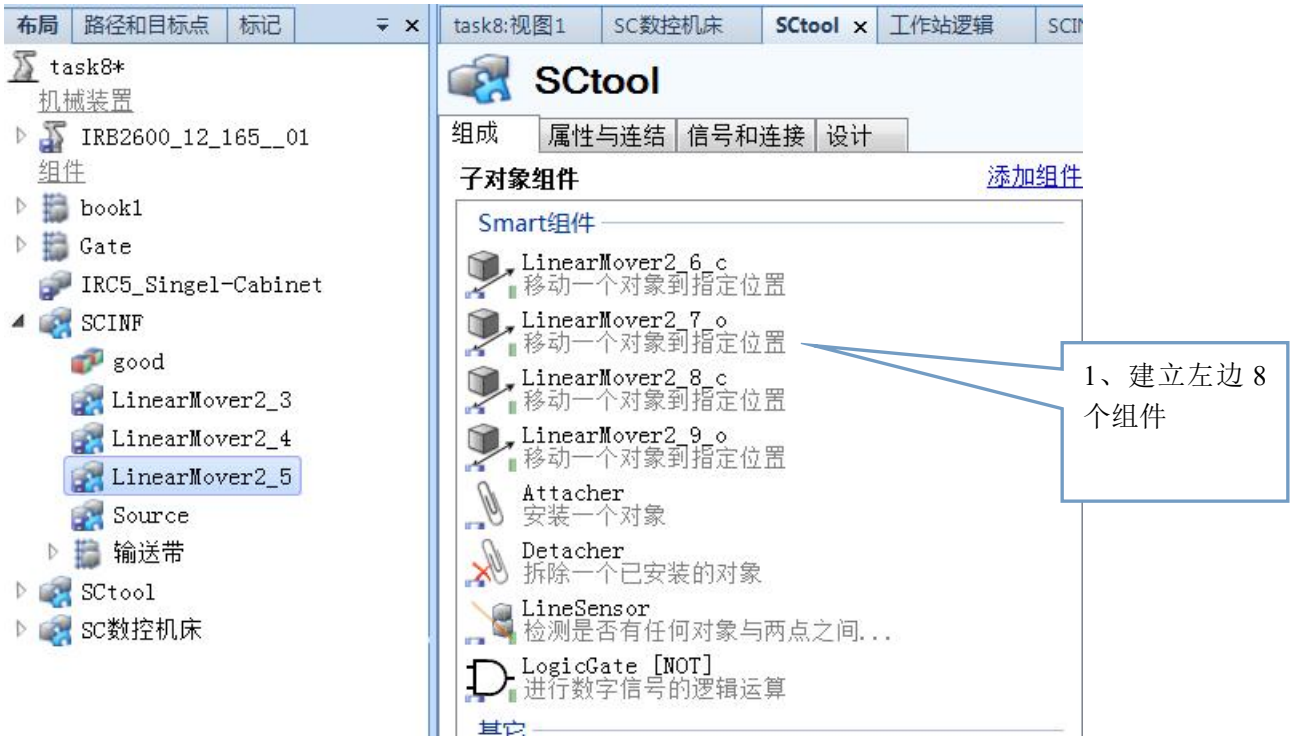
7、添加一个控制信号

I/O连接

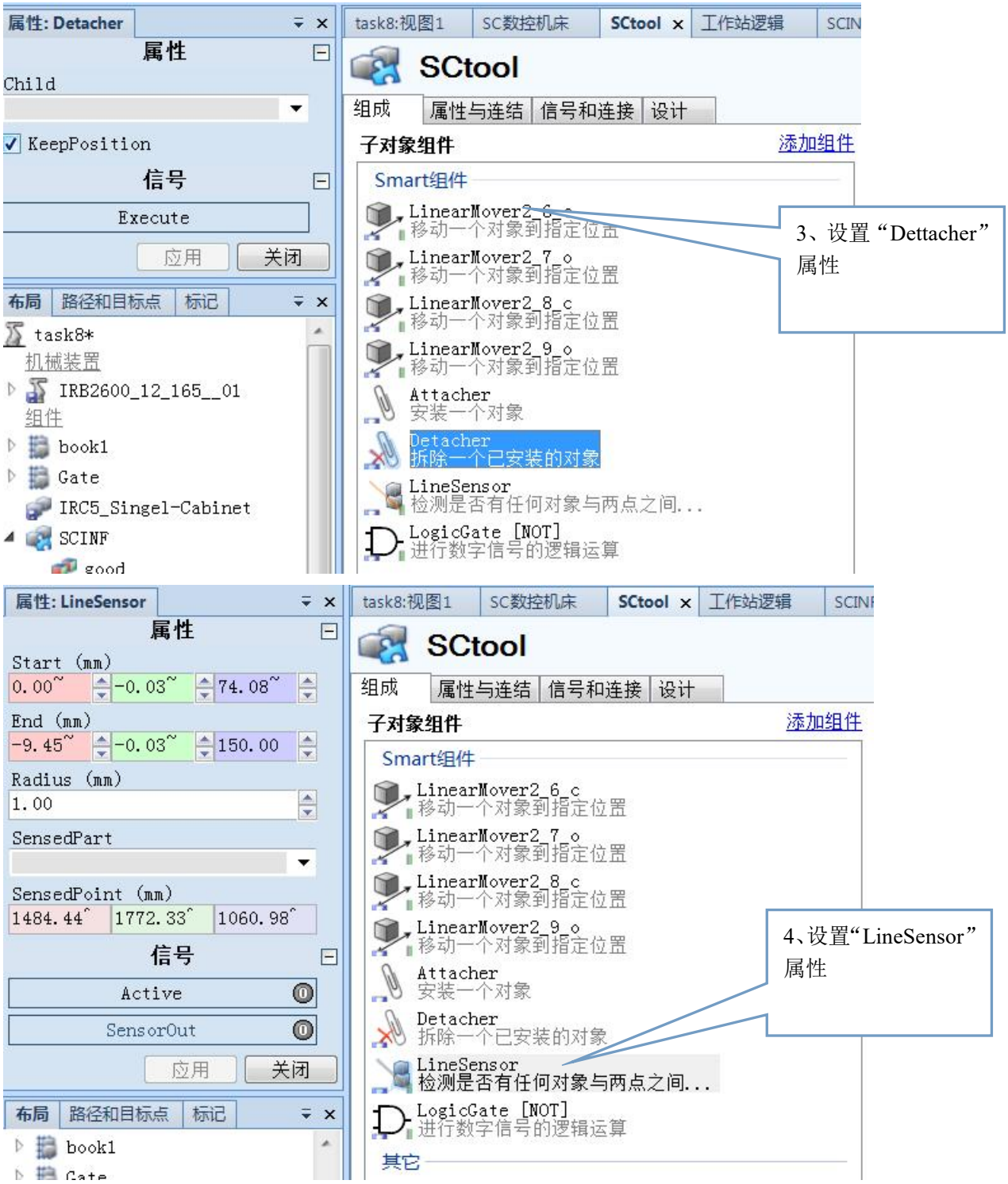
源对象	源信号	目标对象	目标对象
SCINF	DI1	Source	Execute
LinearMover2_3	Executed	LinearMover2_4	Execute
LinearMover2_4	Executed	LinearMover2_5	Execute
Source	Executed	LinearMover2_3	Execute

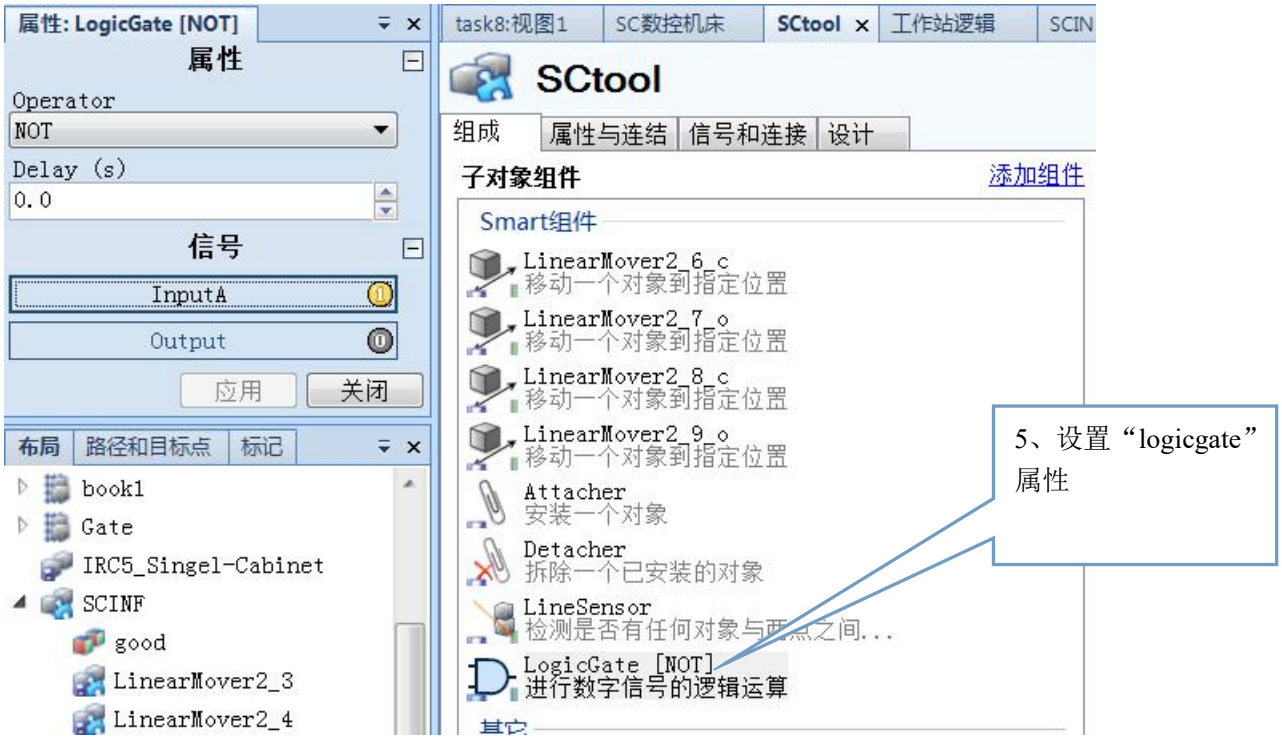
添加I/O Connection 编辑 管理 I/O Connections 删除 上移 下移

8、进行信号 I/O 连接

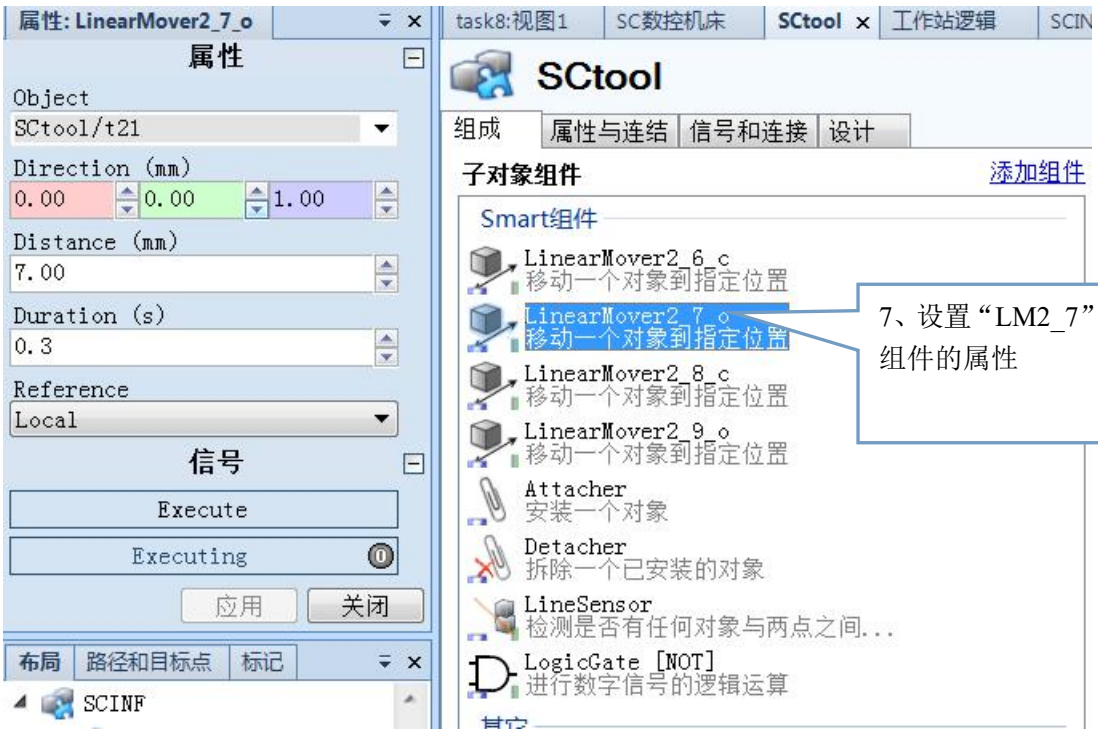

布局
路径和目标点
标记
task8*
机械装置
IRB2600_12_165__01
组件
book1
Gate
IRC5_Singel-Cabinet
SCINF
good
LinearMover2_3
LinearMover2_4
LinearMover2_5
Source
输送带
SCtool
SC数控机床
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
1、建立左边 8 个组件

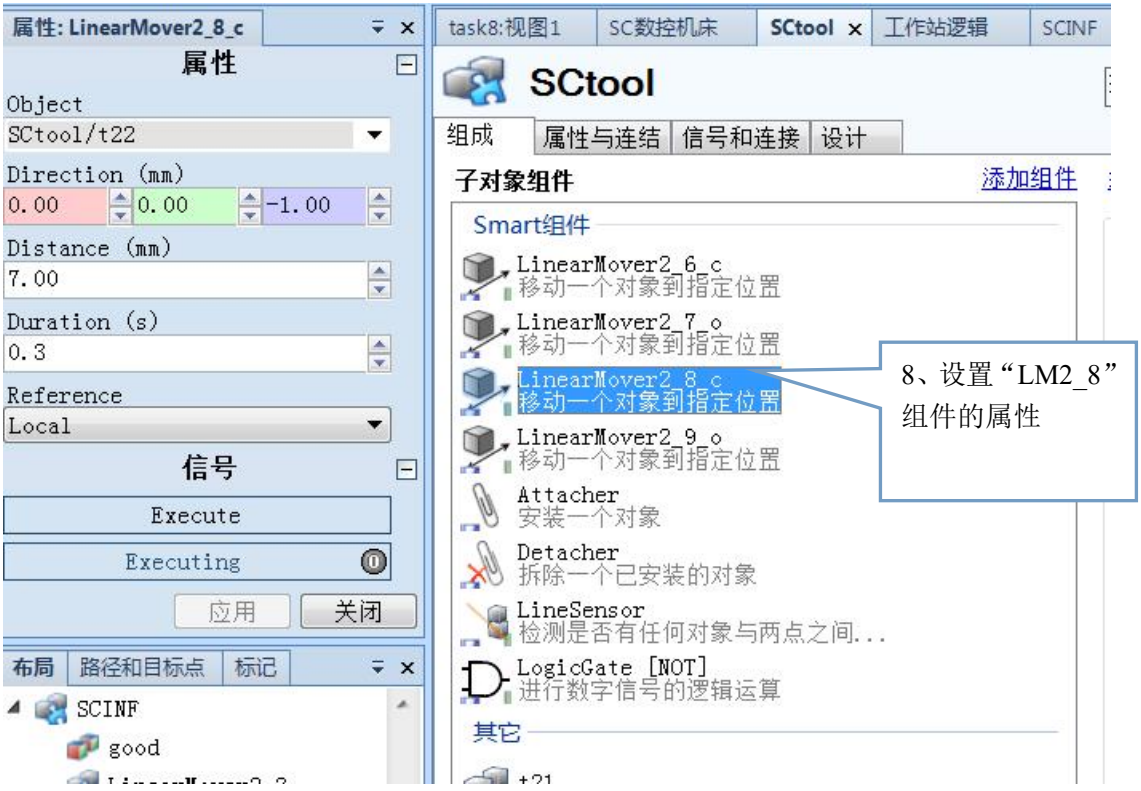

属性: Attacher
属性
Parent
1/TB1
Flange
Child
Mount
Offset (mm)
0.00
0.00
0.00
Orientation (deg)
0.00
0.00
0.00
信号
Execute
应用
关闭
布局
路径和目标点
标记
task8*
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
2、设置“Attacher”属性

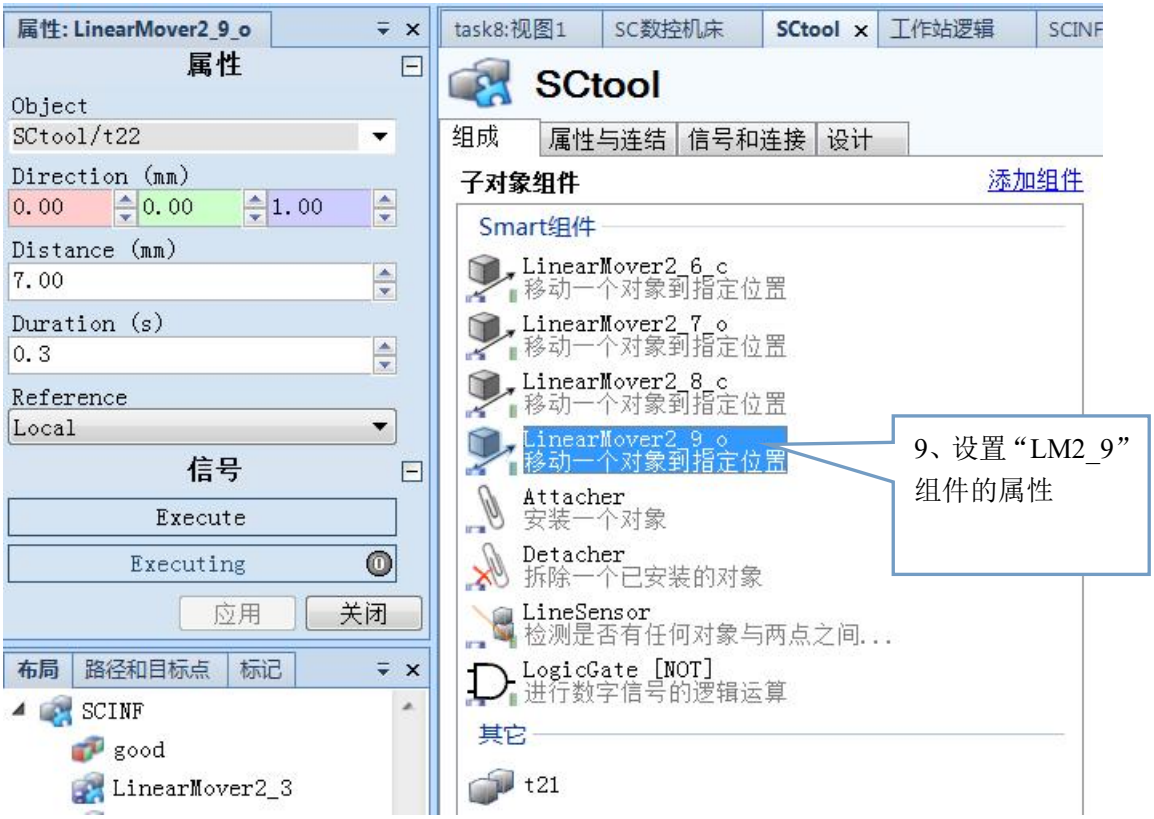

属性: Detacher
属性
Child
KeepPosition
信号
Execute
应用
关闭
布局
路径和目标点
标记
task8*
机械装置
IRB2600_12_165__01
组件
book1
Gate
IRC5_Singel-Cabinet
SCINF
good
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
3、设置“Dettacher”属性
属性: LineSensor
属性
Start (mm)
0.00 -0.03 74.08
End (mm)
-9.45 -0.03 150.00
Radius (mm)
1.00
SensedPart
SensedPoint (mm)
1484.44 1772.33 1060.98
信号
Active
SensorOut
应用
关闭
布局
路径和目标点
标记
book1
Gate
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
其它
4、设置“LineSensor”属性

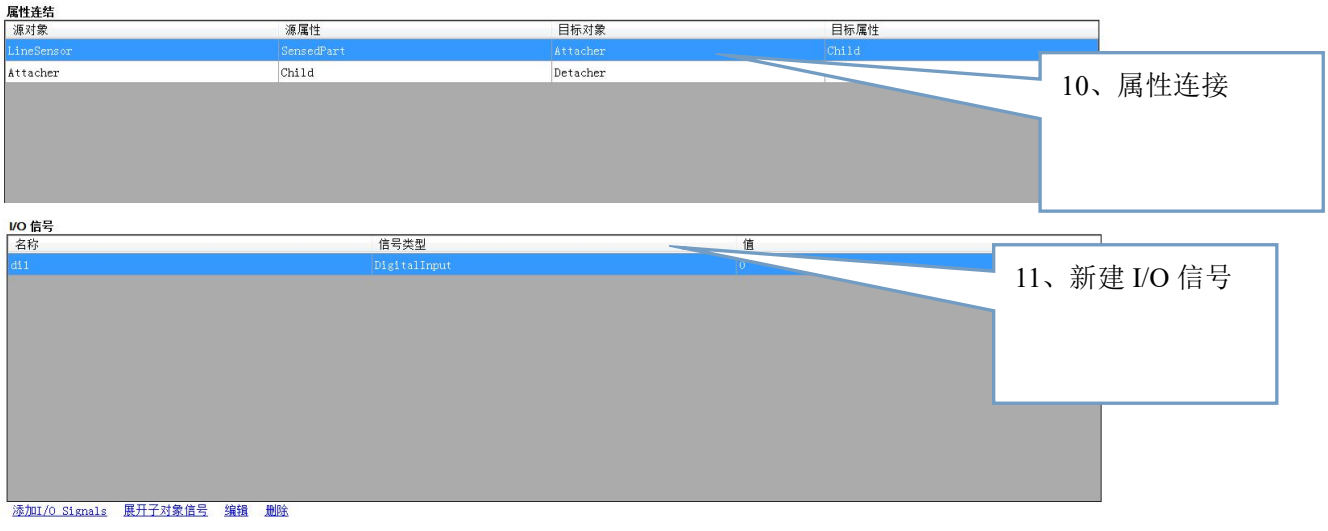

属性: LogicGate [NOT]
属性
Operator
NOT
Delay (s)
0.0
信号
InputA
Output
应用
关闭
布局
路径和目标点
标记
book1
Gate
IRC5_Singel-Cabinet
SCINF
good
LinearMover2_3
LinearMover2_4
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCIN
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
5、设置“logicgate”属性


属性: LinearMover2_6_c
属性
Object
SCtool/t21
Direction (mm)
0.00
0.00
-1.00
Distance (mm)
7.00
Duration (s)
0.3
Reference
Local
信号
Execute
Executing
应用
关闭
布局
路径和目标点
标记
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
6、设置“LM2_6”组件的属性


属性: LinearMover2_7_o
属性
Object
SCtool/t21
Direction (mm)
0.00
0.00
1.00
Distance (mm)
7.00
Duration (s)
0.3
Reference
Local
信号
Execute
Executing
应用
关闭
布局
路径和目标点
标记
SCINF
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
7、设置“LM2_7”组件的属性


属性: LinearMover2_8_c
属性
Object
SCtool/t22
Direction (mm)
0.00
0.00
-1.00
Distance (mm)
7.00
Duration (s)
0.3
Reference
Local
信号
Execute
Executing
应用
关闭
布局
路径和目标点
标记
SCINF
good
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCINF
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
其它
8、设置“LM2_8”组件的属性

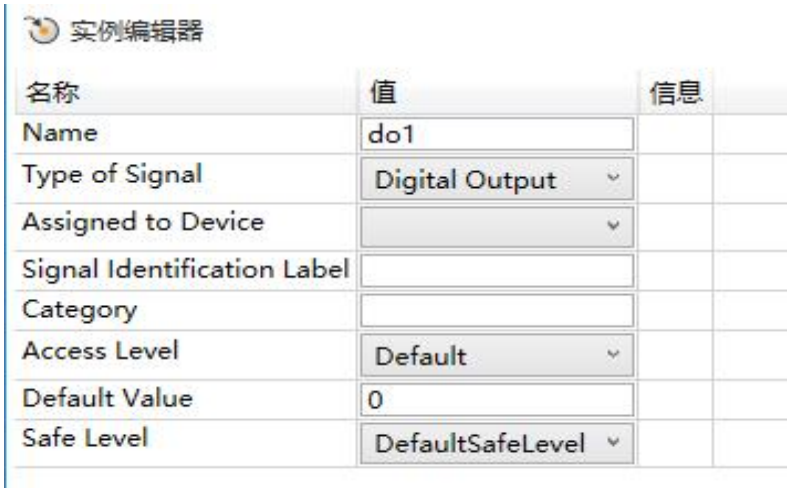

属性: LinearMover2_9_o
属性
Object
SCtool/t22
Direction (mm)
0.00
0.00
1.00
Distance (mm)
7.00
Duration (s)
0.3
Reference
Local
信号
Execute
Executing
应用
关闭
布局
路径和目标点
标记
SCINF
good
LinearMover2_3
task8:视图1
SC数控机床
SCtool
工作站逻辑
SCtool
组成
属性与连结
信号和连接
设计
子对象组件
添加组件
Smart组件
LinearMover2_6_c
移动一个对象到指定位置
LinearMover2_7_o
移动一个对象到指定位置
LinearMover2_8_c
移动一个对象到指定位置
LinearMover2_9_o
移动一个对象到指定位置
Attacher
安装一个对象
Detacher
拆除一个已安装的对象
LineSensor
检测是否有任何对象与两点之间...
LogicGate [NOT]
进行数字信号的逻辑运算
其它
t21
9、设置“LM2_9”组件的属性


属性连结
源对象
源属性
目标对象
目标属性
LineSensor
SensedPart
Attacher
Child
Attacher
Child
Detacher
10、属性连接
I/O 信号
名称
信号类型
值
di1
DigitalInput
0
添加I/O Signals
展开子对象信号
编辑
删除
11、新建 I/O 信号
I/O连接
源对象
源信号
目标对象
目标对象
SCtool
di1
LogicGate [NOT]
InputA
SCtool
di1
LineSensor
Active
SCtool
di1
LinearMover2_6_c
Execute
SCtool
di1
LinearMover2_8_c
Execute
LineSensor
SensorOut
Attacher
Execute
LogicGate [NOT]
Output
LinearMover2_7_o
Execute
LogicGate [NOT]
Output
LinearMover2_9_o
Execute
LogicGate [NOT]
Output
Detacher
Execute
添加I/O Connection
编辑
管理 I/O Connections
删除
上移
下移
12、进行 I/O 连接

机器人 IO 的设定

为了实现真夹具动作的夹/放的动作控制，为了至少需要设定一个虚拟的数字输出信号，这个信号只用于虚拟仿真的作用，并没有与实际的总线或 IO 板进行关联。

数字输出信号的设定菜单操作为：控制器---配置编辑器---IO SYSTEM---SIGNAL。然后将信号设定为以下的表 1 的参数：



机器人轨迹的创建

机器人的动作是从左侧的码垛盘存放处搬运到右边的的方形的码垛盘处。

具体的操作方法如下：

1、设置正确的工件坐标与工具，如下图所示：

2、根据具体的情况，设定正确的机器人运动指令的参数，如下图所示：

3、根据动作的要求通过示教指令的方法，创建对应的轨迹程序，程序样板如下图所示：

```
MODULE Module1
CONST robtarget
Target_10:=[[-903.195434617,-1010,255.499952895],[0,0.866025467,0,-0.49999989],
[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
phome:=[[-903.195434617,-1010,255.499952895],[0,0.866025467,0,-0.49999989],[0,
-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick1:=[[-407.069411175,-1355.912713365,-348.323660453],[0.499961255,0.488301
421,0.519933768,-0.491191758],[0,-1,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick:=[[-407.069108404,-1419.316022798,-348.324062645],[0.499961433,0.488301
```

223,0.519933799,-0.49119174],[0,-1,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick12:=[[-407.069610835,-1313.114898112,301.530837003],[0.499961674,0.48830
1596,0.519933342,-0.491191607],[0,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
Target_30:=[[-563.738092438,-1278.195008652,248.310799233],[0.010772222,-0.73
7280327,-0.006451299,0.67547025],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick23:=[[-1404.948374721,-1187.051870137,248.309048162],[0.010770875,-0.7372
80644,-0.006451596,0.675469924],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick2:=[[-1404.948114344,-1278.19597191,248.31019829],[0.010770959,-0.7372804
32,-0.006451256,0.675470157],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick22:=[[-1270.232110328,-1278.195556618,248.31131745],[0.010771173,-0.73728
0111,-0.006451181,0.675470505],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick24:=[[-717.205337327,-1187.051236636,248.309329491],[0.010770915,-0.73728
0539,-0.00645137,0.675470039],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pla11:=[[121.105023605,118.65981341,165.911773059],[0.702093467,-0.711970162,
-0.012404875,-0.003061343],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pla1:=[[121.105396934,118.659635606,28.089491383],[0.702092901,-0.711970729,-
0.012404339,-0.003061386],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pla21:=[[31.776866619,118.660153277,100.602249446],[0.702092994,-0.711970636,
-0.012404427,-0.003061257],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pla2:=[[31.776906166,118.660161009,28.090633286],[0.702093358,-0.711970275,-0.
012404483,-0.003061451],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
Target_40:=[[-563.738195594,-783.572692467,248.310462721],[0.010772346,-0.737
280661,-0.006451083,0.675469886],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pick13:=[[-407.069015195,-1419.316303493,-207.054327894],[0.499961288,0.4883
01218,0.519933942,-0.491191741],[0,-1,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pla4:=[[31.438507342,25.639238442,28.089588521],[0.702092699,-0.711970936,-0.
012403976,-0.003061017],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
pla41:=[[31.438677399,25.638840699,171.113293931],[0.702093066,-0.711970577,-
0.012403779,-0.003061075],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget

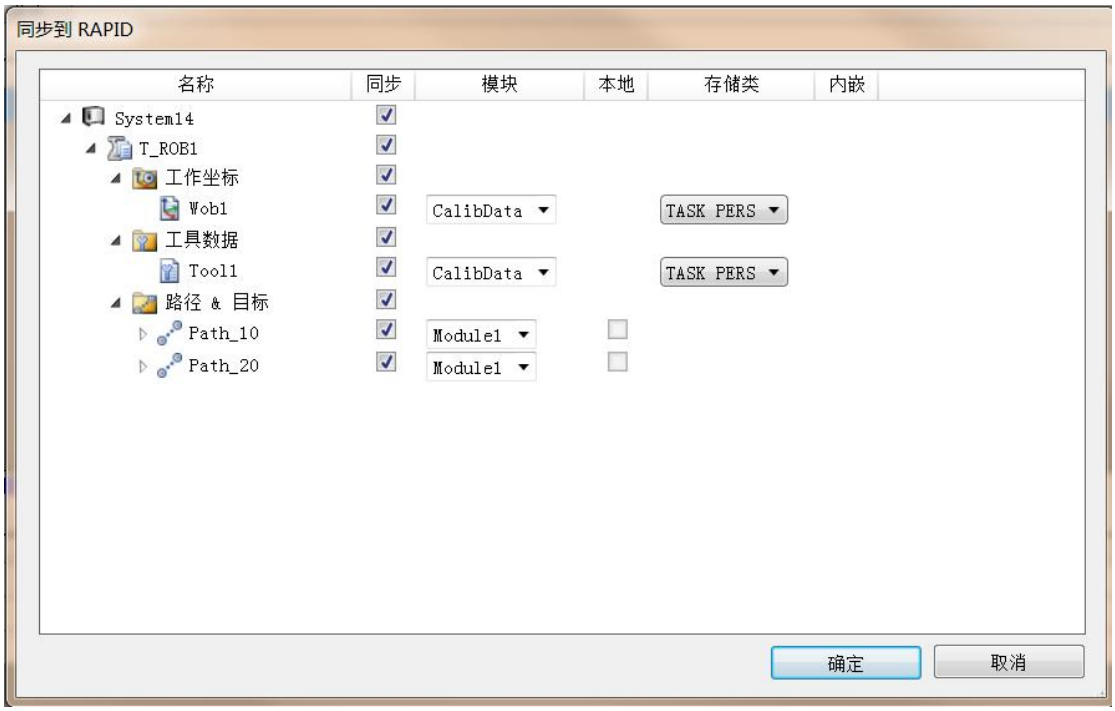
```
pla31:=[[117.691605535,25.639122415,151.793080041],[0.702092946,-0.711970685,
-0.012404359,-0.003061171],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget
pla3:=[[117.691576241,25.639270723,28.090115546],[0.702092904,-0.71197073,-0.
012404183,-0.003060965],[-1,-2,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    PROC Path_10()
        MoveJ pick1,v1000,z10,Tool1\WObj:=Wob1;
        MoveL pick,v1000,fine,Tool1\WObj:=Wob1;
        Set do1tool;
        WaitTime 0.3;
        MoveJ pick13,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ Target_30,v1000,z10,Tool1\WObj:=Wob1;
        Set do2IN;
        Reset do2IN;
        MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
        MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
        Reset do1tool;
        WaitTime 0.3;
        MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ Target_30,v1000,fine,Tool1\WObj:=Wob1;
        Set do3sk;
        WaitTime 3;
        Reset do3sk;
        WaitTime 1.5;
       MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
       MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
       Set do1tool;
       WaitTime 0.3;
       MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
       MoveJ Target_40,v1000,z10,Tool1\WObj:=Wob1;
       MoveJ pla41,v1000,z10,Tool1\WObj:=Wob1;
       MoveL pla4,v1000,z10,Tool1\WObj:=Wob1;
       WaitTime 0.3;
       Reset do1tool;
       MoveJ pla41,v1000,z10,Tool1\WObj:=Wob1;
       MoveJ phome,v1000,z10,Tool1\WObj:=Wob1;
    ENDPROC
    PROC Path_20()
        MoveJ phome,v1000,z10,Tool1\WObj:=Wob1;
        Reset do1tool;
        Reset do3sk;
        Set do2IN;
        Reset do2IN;
        WaitTime 4;
```

```
MoveJ pick1,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick,v1000,fine,Tool1\WObj:=Wob1;
Set do1tool;
WaitTime 0.3;
MoveJ pick13,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_30,v1000,z10,Tool1\WObj:=Wob1;
Set do2IN;
Reset do2IN;
MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
Reset do1tool;
WaitTime 0.3;
MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_30,v1000,fine,Tool1\WObj:=Wob1;
Set do3sk;
WaitTime 3;
Reset do3sk;
WaitTime 1.5;
MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
Set do1tool;
WaitTime 0.3;
MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_40,v1000,z10,Tool1\WObj:=Wob1;
MoveJ pla11,v1000,z10,Tool1\WObj:=Wob1;
MoveL pla1,v1000,fine,Tool1\WObj:=Wob1;
Reset do1tool;
WaitTime 0.3;
MoveJ pla11,v1000,z10,Tool1\WObj:=Wob1;
MoveJ phome,v1000,z10,Tool1\WObj:=Wob1;
MoveJ pick1,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick,v1000,fine,Tool1\WObj:=Wob1;
Set do1tool;
WaitTime 0.3;
MoveJ pick13,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_30,v1000,z10,Tool1\WObj:=Wob1;
Set do2IN;
Reset do2IN;
MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
Reset do1tool;
WaitTime 0.3;
MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_30,v1000,fine,Tool1\WObj:=Wob1;
```

```
Set do3sk;
WaitTime 3;
Reset do3sk;
WaitTime 1.5;
MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
Set do1tool;
WaitTime 0.3;
MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_40,v1000,z10,Tool1\WObj:=Wob1;
MoveJ pla21,v1000,z10,Tool1\WObj:=Wob1;
MoveL pla2,v1000,fine,Tool1\WObj:=Wob1;
Reset do1tool;
WaitTime 0.3;
MoveJ pla21,v1000,z10,Tool1\WObj:=Wob1;
MoveJ phome,v1000,z10,Tool1\WObj:=Wob1;
MoveJ pick1,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick,v1000,fine,Tool1\WObj:=Wob1;
Set do1tool;
WaitTime 0.3;
MoveJ pick13,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_30,v1000,z10,Tool1\WObj:=Wob1;
Set do2IN;
Reset do2IN;
MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
Reset do1tool;
WaitTime 0.3;
MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_30,v1000,fine,Tool1\WObj:=Wob1;
Set do3sk;
WaitTime 3;
Reset do3sk;
WaitTime 1.5;
MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
Set do1tool;
WaitTime 0.3;
MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
MoveJ Target_40,v1000,z10,Tool1\WObj:=Wob1;
MoveJ pla31,v1000,z10,Tool1\WObj:=Wob1;
MoveL pla3,v1000,z10,Tool1\WObj:=Wob1;
WaitTime 0.3;
Reset do1tool;
```

```
        MoveJ pla31,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ phome,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ pick1,v1000,z10,Tool1\WObj:=Wob1;
        MoveL pick,v1000,fine,Tool1\WObj:=Wob1;
        Set do1tool;
        WaitTime 0.3;
        MoveJ pick13,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ Target_30,v1000,z10,Tool1\WObj:=Wob1;
        Set do2IN;
        Reset do2IN;
        MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
        MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
        Reset do1tool;
        WaitTime 0.3;
        MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ Target_30,v1000,fine,Tool1\WObj:=Wob1;
        Set do3sk;
        WaitTime 3;
        Reset do3sk;
        WaitTime 1.5;
        MoveJ pick22,v1000,z10,Tool1\WObj:=Wob1;
        MoveL pick2,v1000,fine,Tool1\WObj:=Wob1;
        Set do1tool;
        WaitTime 0.3;
        MoveJ pick23,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ Target_40,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ pla41,v1000,z10,Tool1\WObj:=Wob1;
        MoveL pla4,v1000,z10,Tool1\WObj:=Wob1;
        WaitTime 0.3;
        Reset do1tool;
        MoveJ pla41,v1000,z10,Tool1\WObj:=Wob1;
        MoveJ phome,v1000,z10,Tool1\WObj:=Wob1;
    ENDPROC
ENDMODULE
```

将写好的程序同步到RAPID，菜单操作：基本---同步---同步到RAPID，如下图所示：



仿真的调试

在完成了设置与编程以后，先保存初始状态，接着下来就是要验证一下仿真动画的结果了，具体的操作如下：

1、设定要运行的 RAPID 子程序，在本项目中是 PATH20，菜单操作如下：仿真---仿真设定---指定 PATH2 0，如下图所示：

2、点击仿真菜单中的“播放”就可以看到动画效果了。动画结束后，点击“重置”，恢复到原来的状态。