



3.有效载荷loaddata

对于搬运应用的机器人，应该正确设定夹具的质量、重心tooldata 以及搬运对象的质量和重心数据loaddata。





Loaddata设定的参数

可以通过LoadIdentify进行自动测量

1. 搬运对象的重量

load.mass [kg]

2. 搬运对象的重心

load.cog.x

load.cog.y

load.cog.z [mm]



Loaddata设定的操作步骤

1. “手动操纵” 界面，选择 “有效载荷”

 手动操纵

点击属性并更改

机械单元:	ROB_1...
绝对精度:	Off
动作模式:	线性...
坐标系:	工具...
工具坐标:	tool0...
工件坐标:	wobj0...
有效载荷:	load0...
操纵杆锁定:	无...
增量:	无...

位置

坐标中的位置: WorkObject

X:	1089.4 mm
Y:	258.5 mm
Z:	1052.5 mm
q1:	0.50000
q2:	0.0
q3:	0.86603
q4:	0.0

位置格式...

操纵杆方向

		
X	Y	Z

对准... 转到... 启动...



2. 单击“新建...”

手动操纵 - 有效荷载

当前选择: load0

从列表中选择一项。

有效荷载名称 ▲	模块	范围 1 到 1 共 1
load0	RAPID/T_ROB1/BASE	全局

新建... 编辑 ▲ 确定 取消





3. 对有效载荷数据属性进行设定。
4. 单击“初始值”。

 新数据声明

数据类型: loaddata 当前任务: T_ROB1

名称:	load1	...
范围:	任务	▼
存储类型:	可变量	▼
任务:	T_ROB1	▼
模块:	Module1	▼
例行程序:	<无>	▼
维数	<无>	▼
		...

初始值 确定 取消



5. 对有效载荷的数据根据实际情况进行设定，各参数代表的含义请参考下面的有效载荷参数表。
6. 单击“确定”。

名称	参数	单位
有效载荷质量	load.mass	kg
有效载荷重心	load.cog.x load.cog.y load.cog.z	mm
力矩轴方向	load.aom.q1 load.aom.q2 load.aom.q3 load.aom.q4	
有效载荷的转动惯量	ix iy iz	kg · m ²

 编辑

名称: load1

点击一个字段以编辑值。

名称	值	数据类型
load1:	[0, [0, 0, 0], [1, 0, 0, 0], ...]	loaddata
mass :=	0	num
cog:	[0, 0, 0]	pos
x :=	0	num
y :=	0	num
z :=	0	num

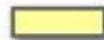
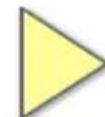
1 到 6 共 14

撤消 确定 取消



在RAPID 编程中，需要对有效载荷的情况进行实时的调整：

```
10 TASK PERS loaddata load1:=[0,[0,0,0],[1,0,0,0],0,0,  
11 PROC main()  
12 Set do1;  
13 GripLoad load1;  
14 MoveJ *, v1000, z50, tool0;  
15 MoveJ *, v1000, z50, tool0;  
16 MoveJ *, v1000, z50, tool0;  
17 Reset do1;  
18 GripLoad load0;  
19 ENDPROC  
20  
21 ENDMODULE
```



Set do1;夹具夹紧

GripLoad load1;指定当前搬运对象的质量和重心load1

.....

Reset do1;夹具松开

GripLoad load0;将搬运对象清除为load0



工具自动识别 (LoadID) 功能

设置tooldata 和loaddata 是自己测量工具的重量和重心，然后填写参数进行设置，但是这样会有一些的不准确性

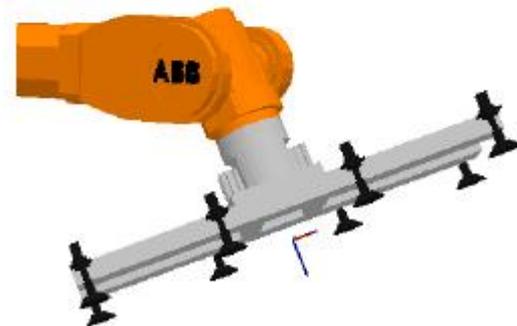


LoadIdentify

LoadIdentify是ABB机器人开发的用于自动识别安装在六轴法兰盘上的工具 (tooldata) 和载荷 (loaddata) 的重量, 以及重心。

手持工具的应用中, 应使用LoadIdentify识别工具的重量和重心。

手持夹具的应用中, 应用使用LoadIdentify识别夹具和搬运对象的重量和重心。





LoadIdentify的操作步骤

The screenshot shows the ABB RobotStudio interface. The main window displays a ladder logic program with the following code:

```
flag2 := FALSE;  
WaitTime 2;  
ENDPROC  
ENDMODULE
```

The 'Call Routine' dialog box is open, showing a list of routines to be called. The 'LoadIdentify' routine is selected. The dialog box also shows the 'Selected Routine' field and a 'View' button.

Step1 点击“调用例行程序”

Step2 选择“LoadIdentify”



调用LoadIdentify的方法

使用LoadIdentify的方法 1

The screenshot shows the ABB robot software interface with the following components:

- Top Panel:** Shows the robot name 'ABB', IP address '14-50965 (192.168.133.1)', and status '电柜开启 正在运行 (速度 100%)'.
- Main Window:** Displays the 'LoadIdentify' procedure code:


```

      PROC LoadIdentify()
      ! Service routine L
      !*****
      LoadIdentifyProc;
      UNDO
      LoadIdTerminate;
      ENDPROC
      
```
- Right Panel:** Shows a dialog box with the text:

The program pointer will be lost and the current path will be cleared during execution. Set DP to main when this service routine is finished! Press OK to continue!
- Bottom Panel:** Shows a selection menu with options 'PayLoad' and 'Tool'. The 'PayLoad' option is highlighted with a red box.

Four blue callout boxes provide instructions for the steps:

- Step1:** 运行此例程序 (Run this procedure)
- Step2:** 点击OK继续 (Click OK to continue)
- Step3:** 选择是对Tool (tooldata) 还是PayLoad(loaddata)进行识别 (Select whether to identify Tool (tooldata) or PayLoad (loaddata))
- Step4:** 确认1-6轴在0度位置，如果的搬运的机器人5轴可尝试向下。如果是测量Payload的话要在夹具上装上搬运对象。 (Confirm 1-6 axes are at 0 degrees. If moving a robot, axis 5 can be tried downwards. If measuring Payload, load the object on the fixture.)



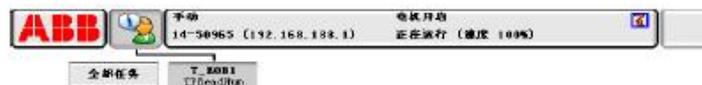
调用LoadIdentify的方法

使用LoadIdentify的方法 2



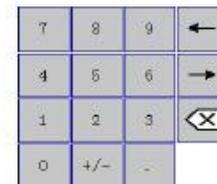
Load identification of active tool
Identify tool named tool1 ?

Step5 确认对应的工具名称是否正确



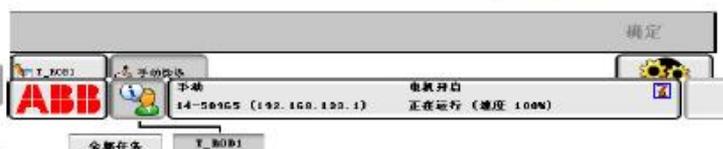
Identification method for Tool:
1 = Known mass in Tool
2 = Unknown mass in Tool
0 = Cancel
Select identification method

Step6 一般选2，由系统自动计算重量



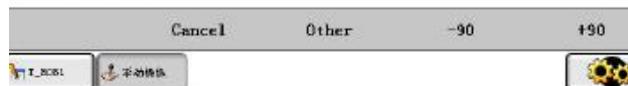
Select angle (with sign) between actual axis 6 position and the coming ditto for the measurement movements. Recommendation + or - 90 degrees. Min. +/- 30 degrees. Select configuration angle

Step7 设定计算过程中6轴的运动范围，设定时要考虑6轴上工具是否有运动干涉



Should test of measurement movements first be done with low speed in MAN/RED. SPEED before the real measurements with higher speed Run test with slow speed ?

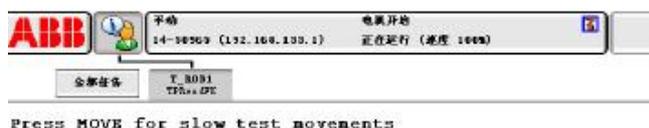
Step8 正式识别前进行运动干涉验证，点击YES继续





调用LoadIdentify的方法

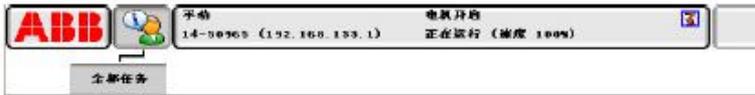
使用LoadIdentify的方法 3



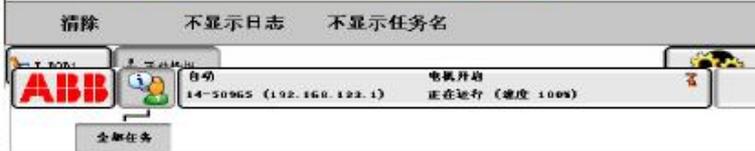
Step9 点击MOVE继续



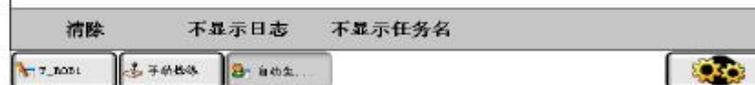
Step11 到此画面后，并确认无干涉后，切换到自动状态，然后点击MOVE



Step10 此时观察机器人的运动是否有干涉



Step12 自动识别中





调用LoadIdentify的方法

使用LoadIdentify的方法 4



The screenshot shows two panels of the ABB robot control interface. The left panel displays instructions for switching to manual mode and starting program execution. The right panel shows the results of a tool identification process, including mass, center of gravity, and accuracy, with a warning for low accuracy. Both panels have blue callouts indicating specific steps.

Step13 识别完成，切换回手动状态

Step14 确认识别数据是否正确，最后点击 YES 数据就自动保存



其他程序数据

ABB工业机器人对于点位的定义

ABB工业机器人的编程语言 RAPID 中，对于点位的描述，专门有一种数据类型：`robtarget`，如下面对于点 `Target_10` 的定义：

```
CONST robtarget
```

```
Target_10:=[[103.446614369839,177.778223757339,29.999999999999  
99],[6.07064838351457E-17,-  
0.130526192220051,0.99144486137381,7.99216021664582E-  
18],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```



其结构解释如下：

< dataobject of robtarget >

< trans of pos >

< x of num > < y of num > < z of num >

< rot of orient >

< q1 of num > < q2 of num > < q3 of num > < q4 of num >

< robconf of confdata >

< cf1 of num > < cf4 of num > < cf6 of num > < cfx of num >

< extax of extjoint >

< eax_a of num > < eax_b of num > < eax_c of num >

< eax_d of num > < eax_e of num > < eax_f of num >

其中，XYZ是TCP在参考坐标系中的坐标值；q1\q2\q3\q4是四元数来表示TOOL坐标系在参考坐标系中的姿态，有的机器人系统是用欧拉角来表示姿态的；cf1\cf4\cf6\cfx表示1\4\6\轴的配置，因为对于一个点，机器人各轴可能有两种以上姿态可以到达；eax_a\~\f 表示外轴的位置。