

ABB 机器人如何创建数组

1) 什么是数组 ?

将相同数据类型的元素按一定顺序排列的集合。

2) 有什么用 ?

比如创建一个 `robtargt` 类型的数组 `p_array` , `p_array` 里有 10 个点位, 走完 10 个位置就可以用如下代码, 方便简洁。

```
FOR i FROM 1 TO 10 DO  
  MoveL p_array{i},v500,z1,tool0;  
ENDFOR
```

3) 哪些数据类型能创建数组

所有数据类型都可以创建数组

4) 数组最大几维 ?

ABB 机器人数组最大三维 !

5) 数组起始序号是 0 还是 1 ?

是 1

6) 如何创建

(1) 程序数据

(2) 选中类型，比如 Robtarget

(3) 新建，设置名称，维数（这里举例 1 维），每个维度的元素数量（举例为 10）
(4) 创建完，就有了一个 10 个数据的 p_array。

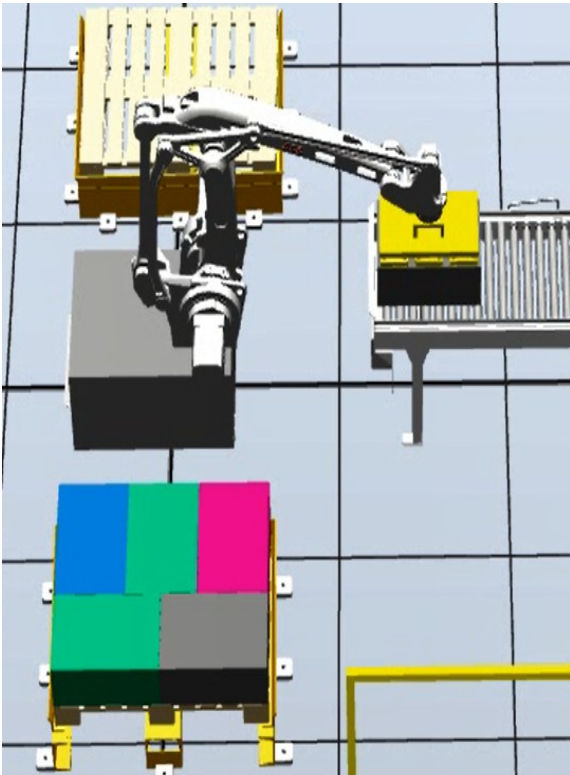
(5) 修改点位，点击上图上某一行，修改位置，即把数组里的某一个元素位置修改了。

The screenshot shows the ABB robot software interface. At the top, there is a status bar with a menu icon, a checkmark, a user icon, the name 'huanhangweishi (CN-1-0316030)', and a stop button. Below this, the 'robtargt' tab is active. The main area displays '维数名称: p_array [10]' and '点击需要编辑的组件。'. Below this is a table with 6 rows and 2 columns: '组件' and '值'. The table shows the first 6 elements of the array, each containing a 3D position vector and a scalar value. A yellow arrow points to the first row of the table. At the bottom, there is a '修改位置' button and a logo for 'ABB机器人实操技巧'.

组件	值
(1)	[[10. 72, -18. 58, -30], [0. 222093, 0. 0112633, -0. 00302..
(2)	[[10. 72, -18. 58, -30], [0. 222093, 0. 0112633, -0. 00302..
(3)	[[10. 72, -18. 58, -30], [0. 222093, 0. 0112633, -0. 00302..
(4)	[[10. 72, -18. 58, -30], [0. 222093, 0. 0112633, -0. 00302..
(5)	[[10. 72, -18. 58, -30], [0. 222093, 0. 0112633, -0. 00302..
(6)	[[10. 72, -18. 58, -30], [0. 222093, 0. 0112633, -0. 00302..

运用数组来码垛 5 个 600*400*250 的箱子

```
CONST num n{5,4}:=[[0,0,0,0],[-600,0,0,0],[-700,500,0,-90],[-300,500,0,-90],[100,500,0,-90]];
```



```
FOR i FROM 1 TO 5 DO  
    WaitDI di2, 1;  
    MoveL p20, v1000, fine, tool0;  
    Set do2;  
    WaitTime 2;  
    WaitDI di3, 1;  
    MoveL p10, v1000, fine, tool0;  
    MoveL RelTool(p30,n{i,1},n{i,2},n{i,3}\Rz:=n{i,4}), v1000, fine, tool0;  
    Reset do2;  
    waittime 2;  
    MoveL RelTool(p30,n{i,1},n{i,2},n{i,3}-100\Rz:=n{i,4}), v1000, fine, tool0;  
    MoveL p10, v1000, fine, tool0;  
ENDFOR
```

中断程序

1. 中断设定

指令	说明
CONNECT	连接一个终端符号到终端程序
ISignalDI	使用一个数字输入信号触发中断
ISignalDO	使用一个数字输出信号触发中断
ISignalGI	使用一个组输入信号触发中断
ISignalGO	使用一个组输出信号触发中断
ISignalAI	使用一个模拟输入信号触发中断
ISignalAO	使用一个模拟输出信号触发中断
ITimer	计时中断
TriggInt	在一个指定的位置触发中断
IPers	使用一个可变量触发中断
IError	当一个错误发生时触发中断
IDelete	取消中断

• CONNECT Interrupt WITH Traproutine;

- Interrupt : 中断数据名称 (intnum)
- Traproutine : 中断数据程序 (identifier)
- 应用 : 将机器人相应中断数据连接到相应中断处理程序, 必须同ISignalDI、ISignalDO、ISignalAI、ISignalAO或ITimer联合使用。
- 限制 : 中断数据数据类型必须为变量 (VAR)
- 一个中断数据不允许连接多个中断处理程序, 先用IDelete将原链接去除。
- 一个中断处理程序可连接多个中断数据。
- Error Handler :
- ERR_ALRDYCNT : 中断数据已被连接至中断处理程序
- ERR_CNTNOTVAR : 中断数据不是VAR类型
- ERR_INOMAX : 没有更多的中断数据可用

• ISignalDI [\Single], Signal, TriggValue, Interrupt;

- [\Single] : 单次中断开关 (Switch)
- Signal : 触发中断信号 (SignalDI)
- TriggValue : 触发信号值 (dionum)
- Interrupt : 中断信号名称 (intnum)
- 应用 : 将响应数字信号输入信号触发相应的中断功能。必须与CONNECT联合使用。

• IDelete Interrupt;

- 将中断数据与中断处理程序连接去除。

• ISignalDO [\Single], Signal, TriggValue, Interrupt;

- 响应数字输出信号中断。

2. 中断的控制

指令	说明
ISleep	关闭一个中断
IWatch	激活一个中断
IDisable	关闭所有中断
IEnable	激活所有中断

- **ISignalAO**

- 响应模拟输出信号中断，使用同ISignalAI。

-

- **ISleep Interrupt;**

- 将中断暂时失效，直到执行IWatch。

- **IWatch Interrupt;**

- 激活失效的中断，与ISleep配合使用。

- **IDisable;**

- 使机器人相应中断功能暂时不执行，此指令用于机器人正执行不希望被中断打断的操作期间，如通过通信口读写数据。

- **IEnable;**

- 开启机器人中断功能，与IDisable配合使用。

- **ITimer [\Single], Time, Interrupt;**

- [\Single]：单次中断开关
- Time：触发中断时间
- Interrupt：中断数据名称
- 应用：定时处理机器人相应中断数据，常用于通信口读写数据。

地方继续往下执行程序。专门用来处理紧急情况的专门程序称作中断程序 (TRAP), 例如:

```
VAR intnum intno1;  
    !定义中断数据 intno1  
IDelete intno1;  
    !取消当前中断符 intno1 的连接, 预防误  
触发  
CONNECT intno1 WITH tTrap;  
    !将中断符与中断程序 tTrap 连接  
ISignalDI di1,1, intno1;①  
    !定义触发条件, 即当数字输入信号 di1  
    为 1 时, 触发该中断程序  
  
TRAP tTrap  
    reg1:=reg1+1;  
ENDTRAP
```

不需要在程序中对该中断程序进行调用, 定义触发条件的语句一般放在初始化程序中, 当程序启动运行完该定义触发条件的指令一次后, 则进入中断监控。当数字输入信号 di1 变为 1 时, 则机器人立即执行 tTrap 中的程序, 运行完成之后, 指针返回触发该中断的程序位置继续往下执行。

ABB 机器人创建自定义数据类型

1) 什么是数据类型 ?

诸如 robtarget , jointtarget 等 , 把一系列相关元素集合在一起 , 类似结构体

2) 如何查看系统的数据类型

Robotstudio 的指令功能数据类型

3) ABB 机器人是否可以自定义数据类型 ?

可以

4) 如何创建自定义数据类型

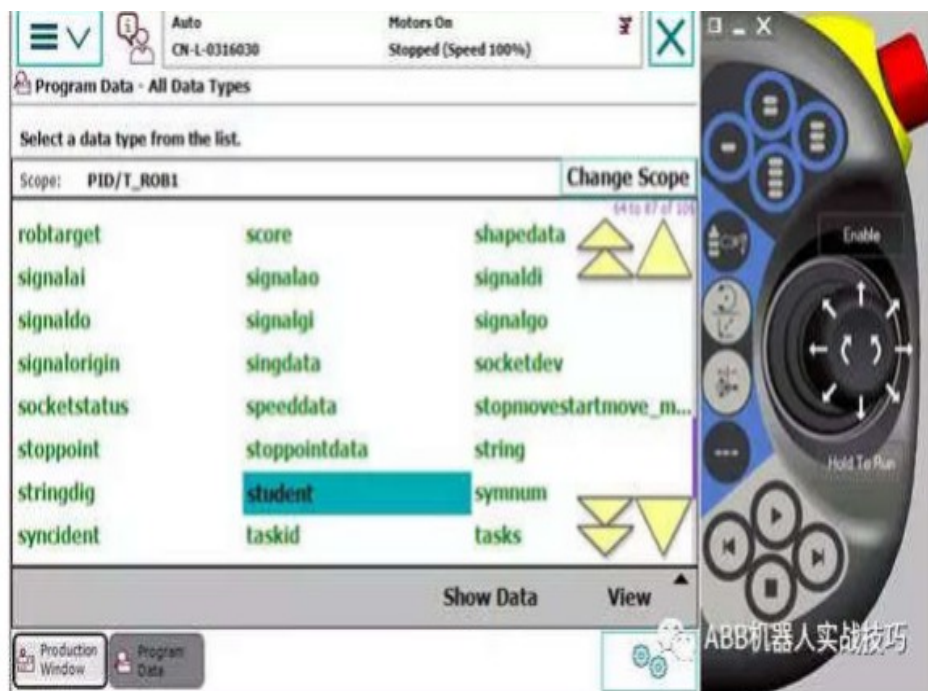
必须在创建在模块的最前面 , 使用关键字 RECORD 和 ENDRECORD

5) 举例如下 :

```
T_ROB1/m44 x
1  MODULE m44
2
3  RECORD student
4      bool gender;
5      num class;
6      score s_score;
7  ENDRECORD
8
9  RECORD score
10     num chinese;
11     num maths;
12 ENDRECORD
```

定义了一种数据类型叫 student , 里面包括性别 (bool 量) , 班级 (num) , 分数。分数又是一个自定义数据类型 , 具体内容包括语文和数学分数。

6) 可以在程序数据里查看到这个自定义的数据类型 student



7) 可以新建一个变量，类型为 student，可以对内容设置

The screenshot shows the 'Edit' dialog for a variable named 'tom'. The dialog has a title bar with 'Manual' and 'CN-L-0316030'. Below the title bar, there is a section for 'Name' with the value 'tom'. Below that, there is a table with three columns: 'Name', 'Value', and 'Data Type'. The table contains the following data:

Name	Value	Data Type
tom:	[FALSE,0,[0,0]]	student
gender :=	FALSE	bool
class :=	0	num
s_score:	[0,0]	score
chinese :=	0	num
maths :=	0	num

At the bottom of the dialog, there are buttons for 'Undo', 'OK', and 'Cancel'. A watermark 'ABB机器人实操技巧' is visible in the bottom right corner.

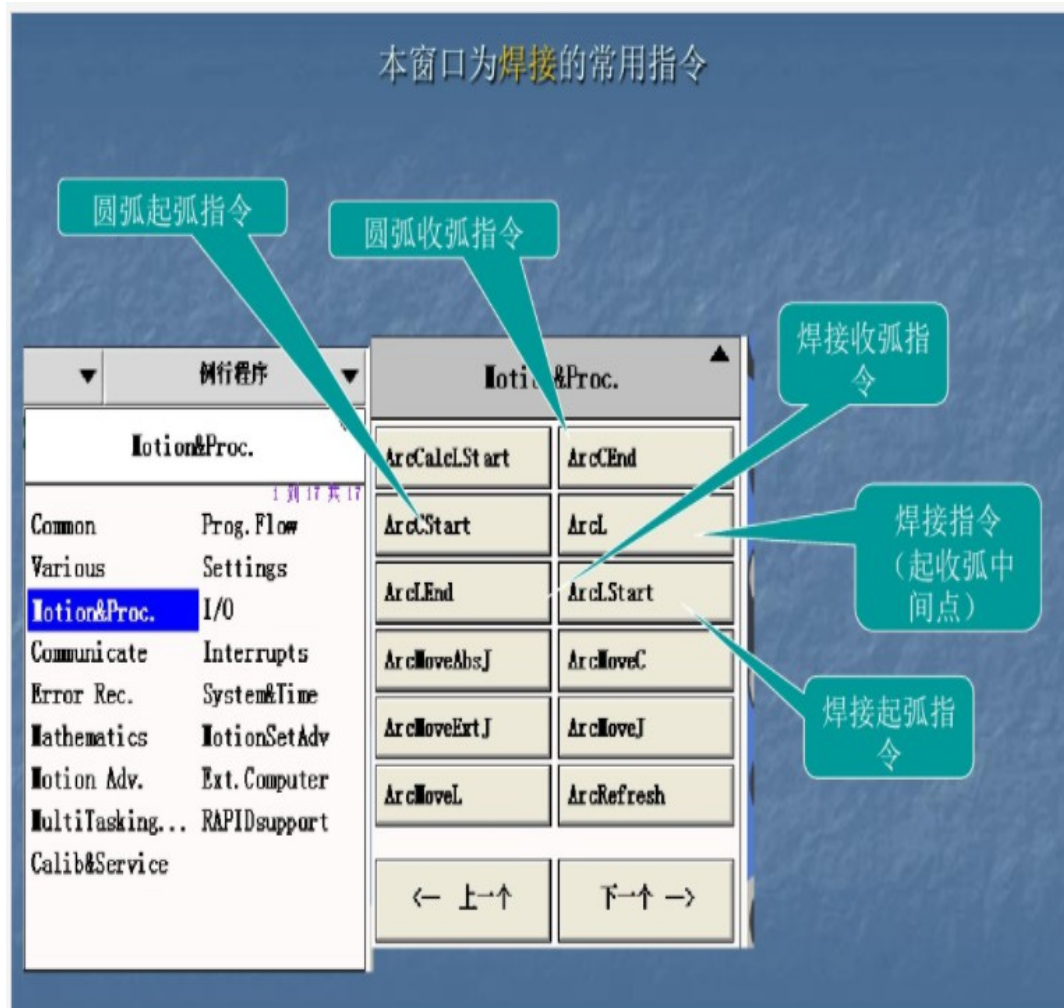
8) 也可以在示教器进行赋值，类似 p10.trans.x

The screenshot shows a routine named 'test22()' in the ABB robot editor. The routine is defined as follows:

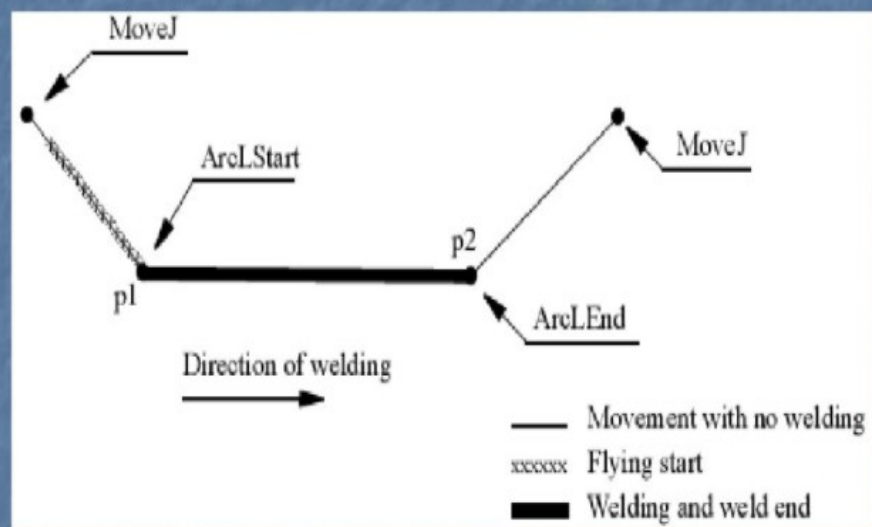
```
PROC test22()  
  tom.gender:=TRUE;  
  tom.s_score.chinese:=100;  
  tom.s_score.maths:=99;  
ENDPROC
```

A yellow plus sign is visible next to the routine name 'test22()'. The editor has tabs for 'Tasks and Programs', 'Modules', and 'Routines' at the top.

ABB 机器人焊接指令



程序图例



常用指令细节说明

- **ArcLStart**、**ArcCStart** 以直线或圆弧运动行走至焊道开始点，并提前做好焊接准备工作。（注意不执行焊接）

ArcLStart p1, v100, seam1, weld1, z10, tool1;

焊接点

空走速度

起弧收弧参数

焊接参数

转弯区域尺寸

- ArcLEnd、ArcCEnd 焊接直线或圆弧至焊道结束点，并完成填弧坑等焊后工作

不管Zone参数指定的圆角是多少，目标点一定是一个fine点。

**ArcLEND p1, v100, seam1, weld1, z10,
tool1;**