



RAPID程序及指令

RAPID程序中包含了一连串控制机器人的指令，执行这些指令可以实现对机器人的控制操作。

应用程序是使用称为RAPID编程语言的特定词汇和语法编写而成的。RAPID是一种英文编程语言，所包含的指令可以移动机器人、设置输出、读取输入，还能实现决策、重复其它指令、构造程序、与系统操作员交流等功能。



RAPID程序的架构

RAPID程序			
程序模块1	程序模块2	程序模块3	系统模块
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程序是由程序模块与系统模块组成。一般地，只通过新建程序模块来构建机器人的程序，而系统模块多用于系统方面的控制。
- 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，这样便于归类管理不同用途的例行程序与数据。
- 3) 每一个程序模块包含了程序数据、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，并且存在于任意一个程序模块中，并且是作为整个RAPID程序执行的起点。



T_ROB1 内的<未命名程序>/Module1/main

任务与程序 ▼ 模块 ▼ 例行程序 ▼

名称	类型	更改
BASE	系统模块	X
mainprg	程序模块	
Module1	程序模块	
Positions	程序模块	
user	系统模块	X

名称	模块	类型
allonc() A	mainprg	Procedure
CurrentPosC... B	mainprg	Function
Initall()	mainprg	Procedure
main() C	mainprg	Procedure
rCheckHOMEPos()	mainprg	Procedure
rP1()	mainprg	Procedure
rP2()	mainprg	Procedure
tIOControl D	mainprg	Trap



常用的RAPID指令：

ABB机器人的RAPID编程提供了丰富的指令来完成各种简单与复杂的应用，下面就从最常用的指令开始学习RAPID编程，领略RAPID丰富的指令集提供的编程便利性。



在示教器上进行指令编辑的基本操作：

①打开ABB菜单，选择“程序编辑器”





- ②选中要插入指令的程序位置，高显为蓝色
- ③单击“添加指令”，打开指令列表
- ④单击此按钮可切换到其它分类的指令列表

I_ROB1 内的<未命名程序>/Module1/Routine1

任务与程序	模块	例行程序
14 PROC main() 15 VAR num m_pos1 := 0; 16 17 ENDPROC 18 PROC Routine1() 19 <SMT> 20 ENDPROC 21 PROC Routine2() 22 <SMT> 23 ENDPROC	 	Common := Compact IF FOR IF MoveAbsJ MoveC MoveJ MoveL ProcCall Reset RETURN Set <- 上一个 下一个 ->
添加指令	编辑	调试 修改位置 隐藏声明



赋值指令：

“：=” 赋值指令用于对程序数据进行赋值，赋值可以是一个常量或数字表达式。

下面就以添加一个常量赋值与数字表达式赋值说明此指令的使用：

常量赋值：reg1:=5;

数学表达式赋值：reg2:=reg1+4;



添加带数学表达式的赋值指令的操作

I_ROB1 内的<未命名程序>/Module1/Routine1

任务与程序	模块	例行程序
14	PROC main()	I.C 2 MoveJ MoveL MoveC ProcCall Set Reset WaitTime WaitDI IF Compact IF := Incr <← 上一个 下一个 →>
15	VAR num m_pos1:=0;	
16		
17	ENDPROC	
18	PROC Routine1()	
19	<SMT>	
20	ENDPROC	
21	PROC Routine2()	
22	<SMT>	
23	ENDPROC	
24	ENDMODULE	

添加指令 编辑 调试 修改位置 隐藏声明

①在指令列表中选择 “:=”



②选中 “reg2”

插入表达式

活动:	num	结果:	num
活动过滤器:		提示:	

<VAR> := <EXP> ;

数据	功能
新建	reg1
reg2	reg3
reg4	reg5

1 到 6 共 6

编辑 更改数据类型... 确定 取消



插入表达式

活动: num

结果: num

活动过滤器:

提示: any type

reg2 := **<EXP>** ;

数据

功能

新建

END_OF_LIST

EOF_BIN

EOF_NUM

reg1

reg2

reg3

reg4

reg5

WAIT_MAX

1 到 10 共 10



编辑

更改数据类型...

确定

取消

③选中
“<EXP>”，显示为蓝色高亮



插入表达式

活动: num

结果: num

活动过滤器:

提示: num

reg2 := reg1;

数据

功能

新建

END_OF_LIST

EOF_BIN

EOF_NUM

reg1

reg2

reg3

reg4

reg5

WAIT_MAX

④选中“reg1”，
点击“+”按钮



编辑

更改数据类型...

确定

取消



插入表达式

活动: num

结果: num

活动过滤器:

提示: any type

reg2 := reg1 + <EXP>;

数据

功能

1 到 10 共 10

新建

END_OF_LIST

EOF_BIN

EOF_NUM

reg1

reg2

reg3

reg4

reg5

WAIT_MAX



编辑

更改数据类型...

确定

取消

- ⑤选中 “<EXP>”，显示为蓝色高亮
- ⑥打开 “编辑” 菜单，选择 “仅限选定内容”



- ⑦通过软键盘输入数字“4”，然后单击“确定”
- ⑧单击“确定”

插入表达式 - 仅限选定内容

4

1	2	3	4	5	6	7	8	9	0	-	=	ⓧ
	q	w	e	r	t	y	u	i	o	p	[]
CAP	a	s	d	f	g	h	j	k	l	;	'	+
Shift	z	x	c	v	b	n	m	,	.	/	:	Home
Int'l		\						↑	↓	←	→	End

确定 取消



T_ROB1 内的<未命名程序>/Module1/Routine1

任务与程序	模块	例行程序
14 PROC main() 15 VAR num m_pos1:=0; 16 17 ENDPROC 18 PROC Routine1() 19 reg2 := reg1 + 4; 20 ENDPROC 21 PROC Routine2() 22 <SMT> 23 ENDPROC 24 ENDMODULE		I.C 2 MoveJ MoveL MoveC ProcCall Set Reset WaitTime WaitDI IF Compact IF := Incr <← 上一个 下一个 →>
添加指令	编辑	调试 修改位置 隐藏声明

⑨添加指令成功

⑩单击“添加指令”，将指令列表收起来



备注符 (!) :

```
PROC Routine1()  
  reg1 := 5;  
  !reg1 := 4;  
  reg2 := reg1 + 4;  
ENDPROC
```