

自定义编程





## 自定义编程

## 应用程序指南

3ABD00009804 REV A  
PDM CODE: 30004987  
BASED ON: 3AFE 64527274 REV A  
中文  
生效期 : 2002 年 6 月 21 日



# 目录

---

## 目录

### 指南介绍

概述 . . . . .	7
兼容性 . . . . .	7
安全须知 . . . . .	7
读者 . . . . .	7
使用 . . . . .	7
相关出版物 . . . . .	8

### 自定义编程

概述 . . . . .	9
什么是自定义编程 . . . . .	9
如何编写程序 . . . . .	10
如何将程序连接到传动单元的应用程序上 . . . . .	11
如何控制程序的执行 . . . . .	11

### 功能块

概述 . . . . .	13
一般规则 . . . . .	13
块输入 . . . . .	13
参数值作为整数输入信号 . . . . .	14
功能块如何处理输入信号 . . . . .	14
如何选择输入信号 . . . . .	14
常量作为整数输入信号 . . . . .	15
如何设置和连接输入信号 . . . . .	15
参数值作为布尔值输入信号 . . . . .	16
功能块如何处理输入信号 . . . . .	16
如何选择输入信号 . . . . .	16
常量作为布尔值输入信号 . . . . .	17
如何设置和连接输入信号 . . . . .	17
字符串输入 . . . . .	17
如何选择输入信号 . . . . .	17
功能块 . . . . .	18
ABS . . . . .	18
ADD . . . . .	18
AND . . . . .	18
COMPARE . . . . .	19
EVENT . . . . .	19
FILTER . . . . .	20
MAX . . . . .	20

MIN .....	21
MULDIV .....	21
NO .....	21
OR .....	21
PI .....	22
PI-BAL .....	22
SR .....	23
SWITCH-B .....	23
SWITCH-I .....	24
TOFF .....	24
TON .....	25
TRIGG .....	25
XOR .....	26

#### **在 ACS 800 标准应用程序中的实际信号和参数**

概述 .....	27
实际信号 .....	27
参数 .....	28

#### **自定义编程的例子**

概述 .....	35
----------	----

#### **用户图**

概述 .....	41
----------	----

# 指南介绍

---

## 概述

本章包括了关于指南的一些基本信息。

## 兼容性

本指南与含有自定义编程功能的传动应用程序相配套。

## 安全须知

遵循与传动单元相配套的所有安全须知。

- 在安装、试车和使用传动单元之前，请阅读 完整的安全须知。完整的安全须知在硬件手册开始部分有所描述。
- 在改变功能的缺省设置之前，请阅读 软件功能的特别警告和注意事项。对于每种功能的警告和注意事项，在相关用户可调参数部分有所介绍。

严格遵循装机资料中的安全指导。

## 读者

作为本手册的读者，需要：

- 懂得标准电气配线操作、电子元件和识别电气原理图符号的基本知识。
- 对于 ABB 传动单元的安装、运行和维修方面没有经验或未经过培训的人员。

## 使用

本指南与传动单元的应用程序—《固件手册》配套使用。《固件手册》含有关于传动参数的基本信息，其中包括自定义编程参数。本指南对自定义编程给出了更详细的说明：

- 什么是自定义编程
- 如何编写一个程序
- 功能块如何操作
- 如何记录程序
- ACS 800 标准应用程序中用于自定义编程的参数和实际信号。

## 相关出版物

关于传动单元的用户资料还包括：

- 《固件手册》— 手册与设备一起发货；
- 《硬件手册》— 手册与设备一起发货；
- 《用于可选设备和程序的指南或补充说明》— 相关手册与设备一起发货。

# 自定义编程

---

## 概述

本章介绍了自定义编程的基础知识，并在编程方面给出了一些指导。

## 什么是自定义编程

传统方式是用户通过设置参数来控制传动单元的运行。每种参数都有自己一组固定的选项或取值范围。这样虽使编程变得容易，但选项却很受限制——用户不能进一步定制自己的想法。而自定义编程使这种想法成为可能，并且不需要使用专门的编程工具或编程语言即可实现：

- 程序由功能块组成；
- 控制盘就是编程工具；  
    用户可以用方框图模板来记录程序。

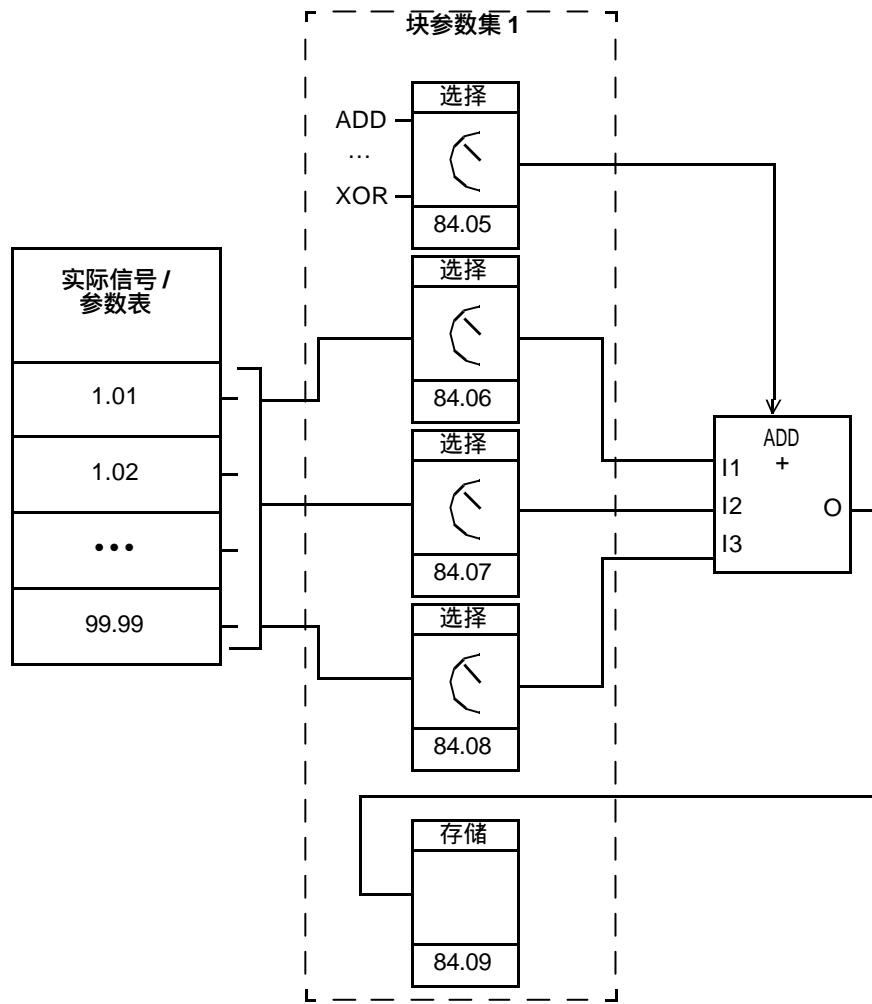
自定义编程最多包含 15 个功能块。一个程序可以由几个独立的功能模块组成。

## 如何编写程序

编程人员可以通过一个 Block Parameter Set ( 块参数集 ) 将一功能块连接到其它功能块上。使用该块参数集不仅可以从传动应用程序中读取数值，也可以给应用程序传输数据。每一个块参数集含有 5 个参数。

右图显示了块参数集 1 在 ACS 800 标准应用程序 ( 参数 84.05~84.09 ) 中的使用：

- 参数 84.05 选择功能块类型。
- 参数 84.06 选择功能块输入端 I1 连接的信号源。
- 参数 84.07 选择功能块输入端 I2 连接的信号源。
- 参数 84.08 选择功能块输入端 I3 连接的信号源。
- 参数 84.09 存储功能块输出端的值。用户不能编辑该参数值。



## 如何将程序连接到传动单元的应用程序上

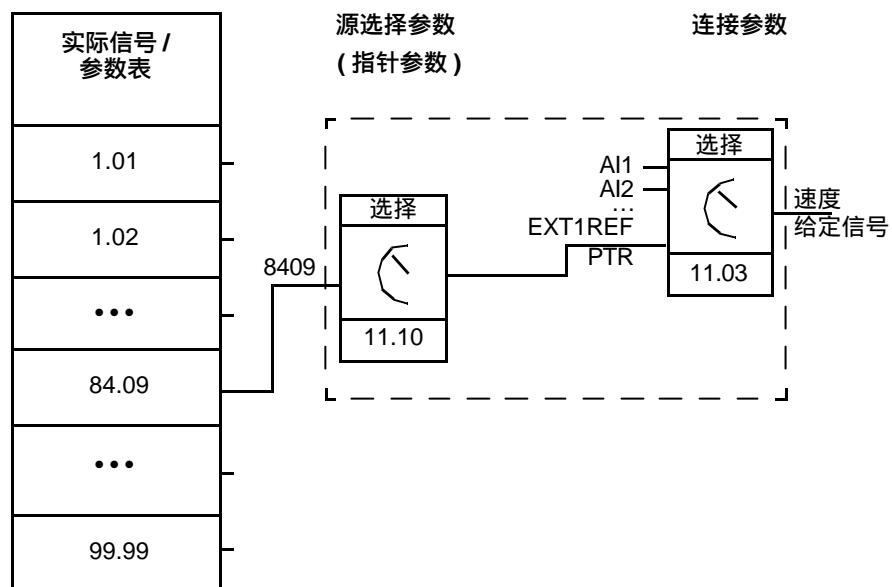
自定义编程的输出值需要传送到传动单元的应用程序上。要完成这一任务，用户需要两个参数：

- 一个连接参数和
- 一个源选择参数（指针）

下图显示了连接的原理。

### 示例：

自定义编程的输出值存储在参数 84.09 中。右图显示了在 ACS 800 标准应用程序中如何使用自定义编程的输出值—速度给定 REF1。



## 如何控制程序的执行

程序按数字顺序来执行功能块，其中所有的块都处于同一时间等级。用户不能修改块，但可以：

- 选择程序的运行模式（停止、启动、编辑）；
- 调整程序的执行时间等级；
- 删除或添加块。



# 组合式功能块

---

## 概述

本章介绍了各种功能块。

## 一般规则

输入端 I1 是必须使用的 (一定不能让其闲置)。对于大多数功能块而言，输入端 I2, I3 等则是可选的。一个闲置未连接的输入端是不会影响功能块的输出。

## 块输入

功能块使用3种输入格式：

- 整数
- 布尔值
- 字符串

用什么格式取决于所选功能块。例如 ADD 块使用整数作为输入信号，OR 块使用布尔值作为输入信号，而字符串仅用于 EVENT 块。

---

注意：各种功能块的输入值在该块启动时，才被读入，而不是同时读所有块。

---

## 参数值作为整数输入信号

### 功能块如何处理输入信号

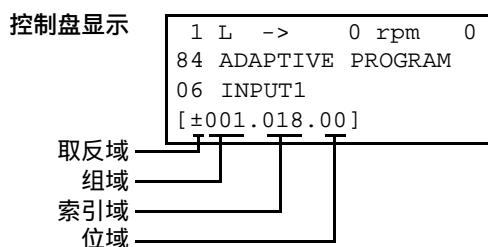
功能块读入所选的整数值。

**注意：**选作输入信号的参数值应为一个实数或整数值。如果其值不是整数格式，功能块会对其进行转换，每一个参数的整数（现场总线）比例换算关系在《固件手册》有所说明。

### 如何选择输入信号

- 滚动选择功能块的输入参数，并将其切换为编辑模式（Enter 键）。
- 根据输入值的读取地址，设置取反域、组域、索引域和位域的值（双箭头键和箭头键）。

下图显示了当输入端 I1 的选择参数处于编辑模式时控制盘的显示。如果取反域的符号为减号（-），则该值被取反，而位域对整数输入或字符串输入都不起作用。



**示例：**在装有 ACS 800 标准应用程序的传动单元中，模拟输入端 AI1 的值是 5.8 V。信号应如何连接到自定义编程的 MAX 块的输入值又是多少？

AI1 是按如下方法连接到功能块的：

- 滚动选择输入端 I1 的参数，并将其切换为编辑模式（Enter 键）。
  - 将组域的值设为 1，索引域的值设为 18。（即 AI1 的值存储在实际信号 1.18 中）
- 功能块的输入值是 5800，因为实际信号 1.18 的整数比例换算关系是：0.001 V = 1（见《固件手册》）。

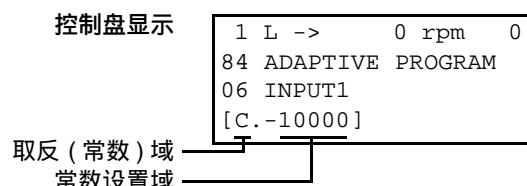
## 常量作为整数输入信号

### 如何设置和连接输入信号

#### 选项 1

- 滚动选择功能块的输入参数，并将其切换为编辑模式（Enter 键）。
- 在取反域中选择 C（可用双箭头键和箭头键），这时显示行发生变化，其余部分变为常数域。
- 给常数域赋常数值（可用双箭头键和箭头键）。
- 按 Enter 键接受输入。

下图显示了当输入端 I1 的选择参数处于编辑模式时控制盘的显示，其中常数域可见。常数取值范围在 -32768 ~ 32767。在程序运行时，不能修改常数值。



#### 选项 2

- 给一个参数（专门用于常数设置的参数）设置常数值。
- 通过输入选择参数将参数值送至功能块。

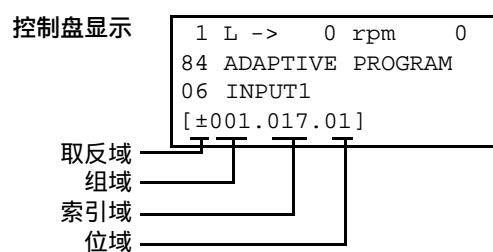
常数值在程序运行时可以被修改。其中，常数的取值范围在 -8388608 ~ 8388607。

## 参数值作为布尔值输入信号

### 功能块如何处理输入信号

- 功能块读取所选的整数值。
  - 功能块使用布尔值输入信号中位域的位值。
- 位值 1 表示布尔值为真，0 表示布尔值为假。

示例：下图显示了当输入信号连接至数字输入端 DI2 的位显示状态时输入端 I1 的值。  
( 在 ACS 800 标准应用程序中，数字输入端的状态内部存储在实际信号 1.17 DI6-1 STATUS 中，其中位 1 对应于 DI2，位 0 对于 DI1。 )



### 如何选择输入信号

参见上节 “[参数值作为整数输入信号](#)”。

---

注意：被选作输入信号的参数应具有一个预设的布尔值（二进制数据字）。参见《固件手册》。

---

## 常量作为布尔值输入信号

### 如何设置和连接输入信号

- 滚动选择功能块的输入参数，并将其切换为编辑模式（Enter 键）。
- 在取反域中选择 C（可用双箭头键和箭头键），这时显示行的其余部分变为常数设置域。
- 给常数域赋值。如果布尔值为真，将常数设置为 -1，如果布尔值为假，则设置为 0。
- 按 Enter 键接受输入。

## 字符串输入

### 如何选择输入信号

字符串输入信号只用于 EVENT 块。

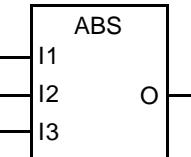
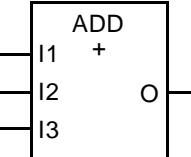
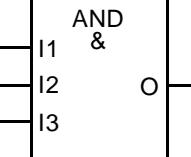
需要输入选择步骤方面的信息，参见上节“[参数值作为整数输入信号](#)”。位选择域在此无效。

---

注意：选作输入信号的参数必须有字符串值。在 ACS 800 标准应用程序中，参数组 85 USER CONSTANTS 中的参数可用作字符串输入。

---

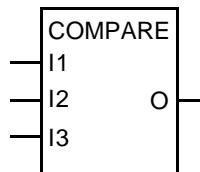
## 功能块

<b>ABS</b>	类型	数学函数																																														
	图解																																															
	运算	输出等于输入端 I1 的绝对值乘以 I2 , 再除以 I3。 $O =  I1  \cdot I2 / I3$																																														
	连接	输入端 I1, I2 和 I3 : 24 位整数值 (23 位 + 1 符号位 )。 输出端 (O) : 24 位整数值 (23 位 + 1 符号位 )。																																														
<b>ADD( 加 )</b>	类型	数学函数																																														
	图解																																															
	运算	输出等于输入信号的总和。 $O = I1 + I2 + I3$																																														
	连接	输入端 I1, I2 和 I3 : 24 位整数值 (23 位 + 1 符号位 )。 输出端 (O) : 24 位整数值 (23 位 + 1 符号位 )。																																														
<b>AND( 与 )</b>	类型	逻辑函数																																														
	图解																																															
	运算	如果所有输入信号为真 , 则输出才为真 , 否则输出为假。真值表 :																																														
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O (二进制)</th> <th>O (显示值)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>真 (所有位为 1)</td> <td>-1</td> </tr> </tbody> </table>		I1	I2	I3	O (二进制)	O (显示值)	0	0	0	假 (所有位为 0)	0	0	0	1	假 (所有位为 0)	0	0	1	0	假 (所有位为 0)	0	0	1	1	假 (所有位为 0)	0	1	0	0	假 (所有位为 0)	0	1	0	1	假 (所有位为 0)	0	1	1	0	假 (所有位为 0)	0	1	1	1	真 (所有位为 1)	-1
I1	I2	I3	O (二进制)	O (显示值)																																												
0	0	0	假 (所有位为 0)	0																																												
0	0	1	假 (所有位为 0)	0																																												
0	1	0	假 (所有位为 0)	0																																												
0	1	1	假 (所有位为 0)	0																																												
1	0	0	假 (所有位为 0)	0																																												
1	0	1	假 (所有位为 0)	0																																												
1	1	0	假 (所有位为 0)	0																																												
1	1	1	真 (所有位为 1)	-1																																												

**连接** 输入端 I1, I2 和 I3 : 布尔值。  
输出端 (O) : 24 位整数值 (组合式的布尔值)。

**COMPARE** 类型 比较函数

图解



**运算** 输出信号位 0, 1 和 2 :

- 如果 I1 > I2, O = ...001 (输出信号的位 0 设置为 1) ;
- 如果 I1 = I2, O = ...010 (输出信号的位 1 设置为 1) ;
- 如果 I1 < I2, O = ...100 (输出信号的位 2 设置为 1)。

输出信号位 3 :

- 如果 I1 > I2, O = ...1xxx (输出信号的位 3 设置为 1 并且会保留该设置直到 I1 < I2 - I3 , 之后位 3 复位为 0)。

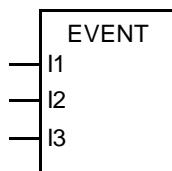
显示屏上的输出值 :

位 0	位 1	位 2	位 3	O ( 显示值 )
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
0	0	1	0	4
0	0	0	1	8
1	0	0	1	9
0	1	0	1	10
0	0	1	1	12

**连接** 输入端 I1, I2 和 I3 : 24 位整数值 (23 位 + 1 符号位)。  
输出端 (O) : 24 位整数值 (组合式的布尔值)。

**EVENT** 类型 事件函数

图解



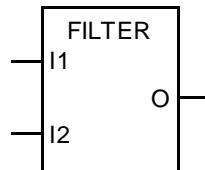
**运算** 输入端 I1 触发事件。输入端 I2 选择参数索引值，从该参数中读取事件信息（字符串）。输入端 I3 选择事件类型（警告或故障）。

I1	I2	I3	说明
0->1			块激活事件
0			块取消事件
	I2		事件信息的内容
		0	事件类型：警告
		1	事件类型：故障
		2	事件类型：事件

**连接** 输入端 I1, I3 : 24 位整数值 (23 位 + 1 符号位)。  
输入端 I2 : 字符串 (必须)。

**FILTER** **类型** 过滤函数

**图解**



**运算** 输出等于输入端 I1 的过滤值。输入端 I2 为过滤时间。

$$O = I1 \cdot (1 - e^{-t/I2})$$

**注意：** 内部计算使用 48 位精确度，以避免出现偏差。

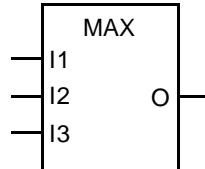
**连接** 输入端 I1 : 24 位整数值 (23 位 + 1 符号位)。

输入端 I2 : 24 位整数值 (23 位 + 1 符号位)，1 对应于 1 ms。

输出端 (O) : 24 位整数值 (23 位 + 1 符号位)。

**MAX** **类型** 比较函数

**图解**

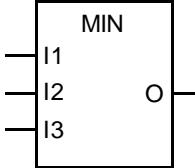
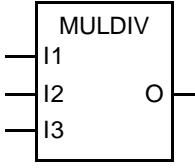
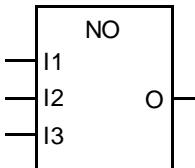
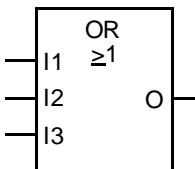


**运算** 输出等于最大输入值。

$$O = \text{MAX}(I1, I2, I3)$$

**连接** 输入端 I1, I2 和 I3 : 24 位整数值 (23 位 + 1 符号位)。

输出端 (O) : 24 位整数值 (23 位 + 1 符号位)。

<b>MIN</b>	类型	比较函数
	图解	
	运算	输出等于最小输入值。 $O = \text{MIN} (I1, I2, I3)$
	连接	输入端 I1, I2 和 I3 : 24 位整数值 (23 位 + 1 符号位)。 输出端 (O) : 24 位整数值 (23 位 + 1 符号位)。
<b>MULDIV</b>	类型	数学函数
	图解	
	运算	输出等于输入 I1 和 I2 的乘积除以 I3。 $O = (I1 \cdot I2) / I3$
	连接	输入端 I1, I2 和 I3 : 24 位整数值 (23 位 + 1 符号位)。 输出端 (O) : 24 位整数值 (23 位 + 1 符号位)。
<b>NO</b>	类型	-
	图解	
	运算	块不工作。
	连接	-
<b>OR</b>	类型	逻辑函数
	图解	

**运算**

只要一个输入信号为真，则输出也为真。真值表：

I1	I2	I3	O(二进制)	O(显示值)
0	0	0	假(所有位为0)	0
0	0	1	真(所有位为1)	-1
0	1	0	真(所有位为1)	-1
0	1	1	真(所有位为1)	-1
1	0	0	真(所有位为1)	-1
1	1	0	真(所有位为1)	-1
1	1	1	真(所有位为1)	-1

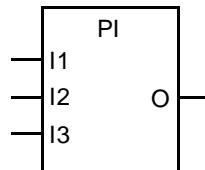
**连接**

输入端 I1, I2 和 I3：布尔值。

输出端 (O)：24 位整数值 (组合式的布尔值)

**PI****类型**

PI 控制器

**图解****运算**

输出等于输入端 I1 乘以 I2/100 再加上 I1 的积分乘以 I3/100。

$$O = I1 \cdot I2/100 + (I3/100) \cdot \int I1$$

注意：内部计算使用 48 位精确度，以避免出现偏差。

**连接**

输入端 I1：24 位整数值 (23 位 + 1 符号位)。

输入端 I2：

- 24 位整数值 (23 位 + 1 符号位)。

- 增益因子。100 相当于 1,10 000 相当于 100。

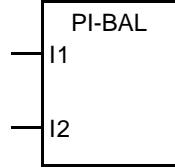
输入端 I3：

- 积分系数。100 相当于 1,10 000 相当于 100。

输出端 (O)：24 位整数值 (23 位 + 1 符号位)。范围限于 0 ~ 10000。

**PI-BAL****类型**

用于 PI 控制器的初始化功能块。

**图解****运算**

该块首先初始化 PI 功能块。当输入端 I1 的值为真时，该块将 I2 的值写入 PI 块的输出端。当输入端 I1 的值为假时，该块断开与 PI 功能块的连接，使其继续进行正常的输出操作。

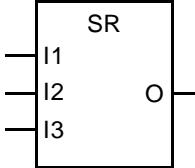
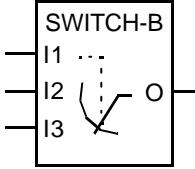
注意：该块仅与 PI 块一起使用，并必须跟随 PI 块使用。

**连接**

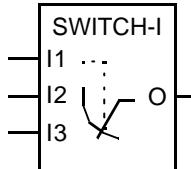
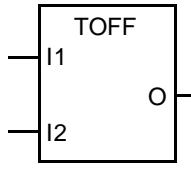
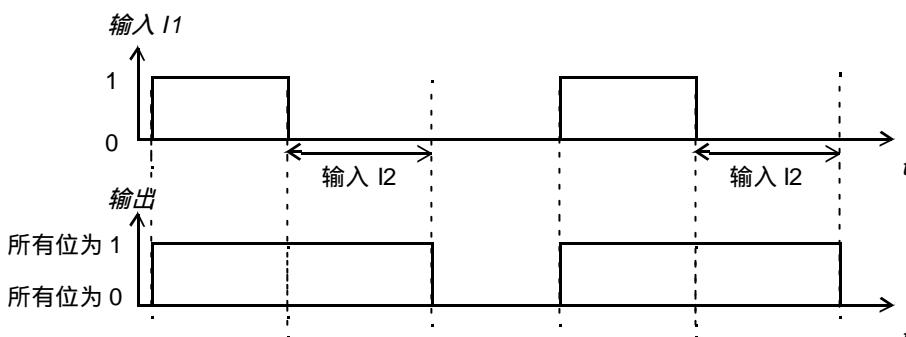
输入端 I1：布尔值；

输入端 I2：24 位整数值 (23 位 + 1 符号位)。

---

<b>SR</b>	类型	逻辑函数																																													
	图解																																														
	运算	<p>设置 / 复位块。输入端 I1 进行置位，输入端 I2 和 I3 对输出端进行复位。</p> <ul style="list-style-type: none"> <li>- 如果 I1, I2 和 I3 为假，则输出端保留当前值；</li> <li>- 如果 I1 为真，I2 和 I3 为假，则输出为真；</li> <li>- 如果 I2 或 I3 为真，则输出为假。</li> </ul>																																													
		<table border="1"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O (二进制)</th> <th>O (显示值)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>当前输出值</td> <td>当前输出值</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>真 (所有位为 1)</td> <td>-1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>假 (所有位为 0)</td> <td>0</td> </tr> </tbody> </table>	I1	I2	I3	O (二进制)	O (显示值)	0	0	0	当前输出值	当前输出值	0	0	1	假 (所有位为 0)	0	0	1	0	假 (所有位为 0)	0	0	1	1	假 (所有位为 0)	0	1	0	0	真 (所有位为 1)	-1	1	0	1	假 (所有位为 0)	0	1	1	0	假 (所有位为 0)	0	1	1	1	假 (所有位为 0)	0
I1	I2	I3	O (二进制)	O (显示值)																																											
0	0	0	当前输出值	当前输出值																																											
0	0	1	假 (所有位为 0)	0																																											
0	1	0	假 (所有位为 0)	0																																											
0	1	1	假 (所有位为 0)	0																																											
1	0	0	真 (所有位为 1)	-1																																											
1	0	1	假 (所有位为 0)	0																																											
1	1	0	假 (所有位为 0)	0																																											
1	1	1	假 (所有位为 0)	0																																											
	连接	<p>输入端 I1, I2 和 I3 : 布尔值。</p> <p>输出端 (O) : 24 位整数值 (23 位 + 1 符号位)。</p>																																													
<b>SWITCH-B</b>	类型	逻辑函数																																													
	图解																																														
	运算	<p>如果输入端 I1 为真，则输出等于输入 I2；如果输入端 I1 为假，则输出等于输入 I3。</p>																																													
		<table border="1"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O</th> <th>O (显示值)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>I2</td> <td>I3</td> <td>I3</td> <td rowspan="2">真 = -1 假 = 0</td> </tr> <tr> <td>1</td> <td>I2</td> <td>I3</td> <td>I2</td> </tr> </tbody> </table>	I1	I2	I3	O	O (显示值)	0	I2	I3	I3	真 = -1 假 = 0	1	I2	I3	I2																															
I1	I2	I3	O	O (显示值)																																											
0	I2	I3	I3	真 = -1 假 = 0																																											
1	I2	I3	I2																																												
	连接	<p>输入端 I1, I2 和 I3 : 布尔值。</p> <p>输出端 (O) : 24 位整数值 (组合式的布尔值)。</p>																																													

---

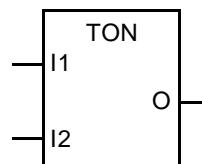
<b>SWITCH-I</b>	类型	逻辑函数													
	图解														
	运算	如果输入端 I1 为真，则输出等于输入 I2；如果输入端 I1 为假，则输出等于输入 I3。													
		<table border="1" data-bbox="492 606 1033 729"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>I2</td> <td>I3</td> <td>I3</td> </tr> <tr> <td>1</td> <td>I2</td> <td>I3</td> <td>I2</td> </tr> </tbody> </table>		I1	I2	I3	O	0	I2	I3	I3	1	I2	I3	I2
I1	I2	I3	O												
0	I2	I3	I3												
1	I2	I3	I2												
	连接	输入端 I1：布尔值。 输入端 I2 和 I3：24 位整数值 (23 位 + 1 符号位)。 输出端 (O)：24 位整数值 (23 位 + 1 符号位)。													
<b>TOFF</b>	类型	定时函数													
	图解														
	运算	当输入端 I1 为真时，则输出为真；以输入端 I1 由真变假的时刻开始计时，当输入端 I1 为假的时间等于或大于输入端 I2 的时间时，则输出为假。													
															
		显示值：真 = -1，假 = 0。													
	连接	输入端 I1：布尔值。 输入端 I2：24 位整数值 (23 位 + 1 符号位)。1 相当于 1 ms。 输出端 (O)： - 24 位整数值 (组合式的布尔值)。													

---

---

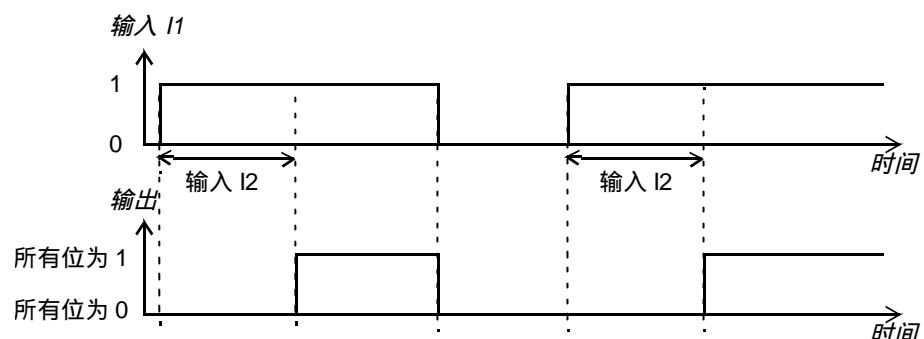
**TON**      类型      定时函数

图解



运算

以输入端 I1 由真变假的时刻开始计时，当输入端 I1 为真的时间等于或大于输入端 I2 的时间时，输出为真。当输入为假时，输出也为假。



显示值：真 = -1，假 = 0。

连接

输入端 I1：布尔值。

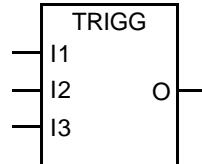
输入端 I2：24 位整数值 (23 位 + 1 符号位)，1 相当于 1 ms。

输出端 (O)：24 位整数值 (组合式的布尔值)。

---

**TRIGG**      类型      定时函数

图解

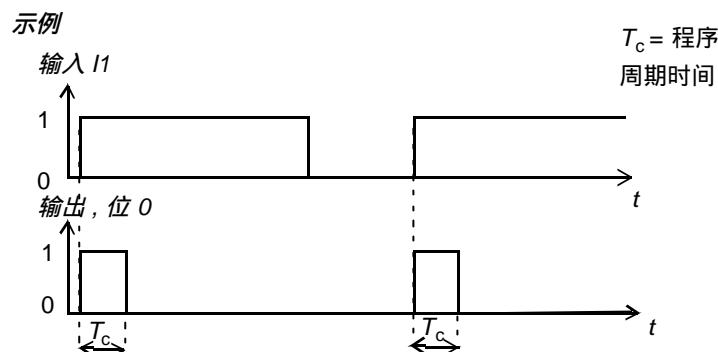


**运算**

输入 I1 的上边沿在一个程序周期内输出位 0 置位。

输入 I2 的上边沿在一个程序周期内输出位 1 置位。

输入 I3 的上边沿在一个程序周期内输出位 2 置位。

**连接**

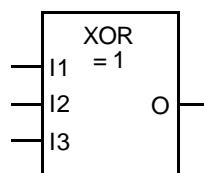
输入端 I1,I2 和 I3 : 布尔值。

输出端 (O) :

- 24 位整数值 (23 位 + 1 符号位)。

**XOR****类型**

逻辑函数

**图解****运算**

只要一个输入为真，则输出也为真，否则输出为假。真值表：

I1	I2	I3	O (二进制)	O (显示值)
0	0	0	假 (所有位为 0)	0
0	0	1	真 (所有位为 1)	-1
0	1	0	真 (所有位为 1)	-1
0	1	1	假 (所有位为 0)	0
1	0	0	真 (所有位为 1)	-1
1	0	1	假 (所有位为 0)	0
1	1	0	假 (所有位为 0)	0
1	1	1	真 (所有位为 1)	-1

**连接**

输入端 I1,I2 和 I3 : 布尔值。

输出端 (O) :

- 24 位整数值 (23 位 + 1 符号位)。

# 在 ACS 800 标准应用程序中的实际信号和参数

---

## 概述

本章列出了 ACS 800 标准应用程序中用于自定义编程的实际信号、参数和参数值。

## 实际信号

下表列出了用于自定义编程的实际信号。缩写 FbEq 代表现场总线的比例换算。

索引	信号名称 / 值	说明	FbEq.
09	ACTUAL SIGNALS	用于 Adaptive Program (自定义编程) 的信号。	
09.01	AI1 SCALED	模拟输入端 AI1 的值比例换算为一个整数值。 20000 = 10V	20000 = 10V
09.02	AI2 SCALED	模拟输入端 AI2 的值比例换算为一个整数值。 20000 = 20mA	20000 = 20mA
09.03	AI3 SCALED	模拟输入端 AI3 的值比例换算为一个整数值。 20000 = 20mA	20000 = 20mA
09.04	AI5 SCALED	模拟输入端 AI5 的值比例换算为一个整数值。 20000 = 20mA	20000 = 20mA
09.05	AI6 SCALED	模拟输入端 AI6 的值比例换算为一个整数值。 20000 = 20mA	20000 = 20mA
09.06	MASTER CW	通过现场总线接口从主机站接收到的主给定数据集的控制字 (CW)。 -32768 ... 32767	-32768 ... 32767
09.07	MASTER REF1	通过现场总线接口从主机站接收到的主给定数据集的给定值 1 (REF1)。 -32768 ... 32767	-32768 ... 32767
09.08	MASTER REF2	通过现场总线接口从主机站接收到的主给定数据集的给定值 2 (REF2)。 -32768 ... 32767	-32768 ... 32767
09.09	AUX DS VAL1	通过现场总线接口从主机站接收到的主给定数据集的给定值 3 (REF3)。 -32768 ... 32767	-32768 ... 32767
09.10	AUX DS VAL2	通过现场总线接口从主机站接收到的主给定数据集的给定值 4 (REF4)。 -32768 ... 32767	-32768 ... 32767
09.11	AUX DS VAL3	通过现场总线接口从主机站接收到的主给定数据集的给定值 5 (REF5)。 -32768 ... 32767	-32768 ... 32767

## 参数

下表列出了用于自定义编程的参数。缩写 FbEq 代表现场总线比例换算。

索引	参数名称 / 值	说明	FbEq
10	START/STOP/DIR	通过这组参数 , 自定义编程可以控制传动单元的启动、停机和转向。	
10.01	EXT1 STRT/STP/DIR		
	PARAM 10.04	指向参数 10.04 选择的信号源。	16
10.02	EXT2 STRT/STP/DIR		
	PARAM 10.05	指向参数 10.05 选择的信号源。	16
10.04	EXT 1 STRT PTR	选择用于参数 10.01 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量： - 参数指针：由取反域、组域、索引域和位域组成。其中位域的值仅对块处理布尔输入有效。 - 常量：由取反域和常量域组成。其中取反域的值必须为 C 才能允许对常量进行设置。	
10.05	EXT 2 STRT PTR	选择用于参数 10.02 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 10.04。	
11	REFERENCE SELECT	通过这组参数 , 自定义编程可以控制传动单元的给定信号。	
11.02	EXT1/EXT2 SELECT		
	PARAM 11.09	指向参数 11.09 选择的信号源。	16
11.03	EXT REF1 SELECT		
	PARAM 11.10	指向参数 11.10 选择的信号源。	37
11.06	EXT REF2 SELECT		
	PARAM 11.11	指向参数 11.11 选择的信号源。	37
11.09	EXT 1/2 SEL PTR	选择用于参数 11.02 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 10.04。	
11.10	EXT 1 REF PTR	选择用于参数 11.03 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 10.04。	
11.11	EXT 2 REF PTR	选择用于参数 11.06 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 10.04。	
14	RELAY OUTPUTS	通过这组参数 , 自定义编程可以控制传动单元的继电器输出。	
14.01	RELAY R01 OUTPUT		
	PARAM 14.16	指向参数 14.16 选择的信号源。	34
14.02	RELAY R02 OUTPUT		

索引	参数名称 / 值	说明	FbEq
	PARAM 14.17	指向参数 <a href="#">14.17</a> 选择的信号源。	34
14.03	RELAY RO3 OUTPUT		
	PARAM 14.18	指向参数 <a href="#">14.18</a> 选择的信号源。	34
14.10	NDIO MOD1 RO1		
	PARAM 14.19	指向参数 <a href="#">14.19</a> 选择的信号源。	7
14.11	NDIO MOD1 RO2		
	PARAM 14.20	指向参数 <a href="#">14.20</a> 选择的信号源。	7
14.12	NDIO MOD2 RO1		
	PARAM 14.21	指向参数 <a href="#">14.21</a> 选择的信号源。	7
14.13	NDIO MOD2 RO2		
	PARAM 14.22	指向参数 <a href="#">14.22</a> 选择的信号源。	7
14.14	NDIO MOD3 RO1		
	PARAM 14.23	指向参数 <a href="#">14.23</a> 选择的信号源。	7
14.15	NDIO MOD3 RO2		
	PARAM 14.24	指向参数 <a href="#">14.24</a> 选择的信号源。	7
14.16	RO PTR1	选择用于参数 <a href="#">14.01</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.17	RO PTR2	选择用于参数 <a href="#">14.02</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.18	RO PTR3	选择用于参数 <a href="#">14.03</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.19	RO PTR4	选择用于参数 <a href="#">14.10</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.20	RO PTR5	选择用于参数 <a href="#">14.11</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.21	RO PTR6	选择用于参数 <a href="#">14.12</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.22	RO PTR7	选择用于参数 <a href="#">14.13</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.23	RO PTR8	选择用于参数 <a href="#">14.14</a> 的信号源。	

索引	参数名称 / 值	说明	FbEq
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
14.24	RO PTR9	选择用于参数 <a href="#">14.15</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
15	ANALOGUE OUTPUTS	通过这组参数，自定义编程可以控制传动单元标准模拟信号的输出。继电器输出。	
15.01	ANALOGUE OUTPUT1		
	PARAM 15.11	指向参数 <a href="#">15.11</a> 选择的信号源。	17
15.06	ANALOGUE OUTPUT2		
	PARAM 15.12	指向参数 <a href="#">15.12</a> 选择的信号源。	16
15.11	AO1 PTR	选择用于参数 <a href="#">15.01</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
15.12	AO2 PTR	选择用于参数 <a href="#">15.06</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
16	SYSTEM CTRL INPUTS	通过这组参数，自定义编程可以控制传动单元的系统控制输入。	
16.01	RUN ENABLE		
	PARAM 16.08	指向参数 <a href="#">16.08</a> 选择的信号源。	15
16.08	RUN ENA PTR	选择用于参数 <a href="#">16.01</a> 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
20	LIMITS	通过这组参数，自定义编程可以控制传动单元的操作极限值。	
20.13	MIN TORQ SEL	选择最小转矩极限值。	
	PARAM 20.18	指向参数 20.18 给出的极限值。	19
20.14	MAX TORQ SEL	选择最大转矩极限值。	
	PARAM 20.19	指向参数 20.19 给出的极限值。	19
20.18	TORQ MIN PTR	选择用于参数 20.13 的信号源。	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
20.19	TORQ MAX PTR	选择用于参数 20.14 的信号源。	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
22	ACCEL/DECCEL	通过这组参数，自定义编程可以控制传动单元的加速和减速。	
22.01	ACC/DEC 1/2 SEL		

索引	参数名称 / 值	说明	FbEq
	PAR 22.08&09	指向参数 22.08 和 22.09 给出加速和减速时间。	15
22.08	ACC PTR	选择用于参数 22.01 的信号源。	100 = 1s
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
22.09	DEC PTR	选择用于参数 22.01 的信号源。	100 = 1s
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
26	MOTOR CONTROL	通过这组参数，自定义编程可以控制传动单元的磁通量。	
26.06	FLUX REF PTR	选择磁通量给定值的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
40	PID CONTROL	通过这组参数，程序可以影响过程 PID 控制。	
40.07	ACTUAL1 INPUT SEL		
	PARAM 40.25	指向参数 <a href="#">40.25</a> 选择的信号源。	6
40.25	ACTUAL1 PTR	选择用于参数 <a href="#">40.07</a> 的信号源。	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参数指针或常量。参见参数 <a href="#">10.04</a> 。	
83	ADAPT PROG CNTRL	自定义编程的执行控制。	
83.01	ADAPT PROG CMD	选择程序运行模式。	
	STOP	停止。程序不能被编辑。	
	START	运行。程序不能被编辑。	
	EDIT	改为编辑模式。程序可以被编辑。	
83.02	EDIT COMMAND	为块选择命令，该块放置的位置由参数 <a href="#">83.03</a> 定义。程序必须处于编辑模式（参见参数 <a href="#">83.01</a> ）。	
	NO	初始值。在执行一个编辑命令之后，该值自动恢复为 NO。	
	PUSH	将块移到参数 <a href="#">83.03</a> 定义的位置，其后的块向后错一位。通过对块参数集进行编程，将新块放置到空位上。  例如：将一个新块放在当前块 4( 参数 84.20 ~ 84.25 ) 和块 5( 参数 84.25 ~ 84.29 ) 之间。 为达到此目的，需要： - 将参数 <a href="#">83.01</a> 设置为可编辑模式。 - 通过参数 <a href="#">83.03</a> 选择位置 5 作为新块的目标地。 - 将新块移到位置 5，由于参数 <a href="#">83.02</a> ( 选项为 PUSH ) 的作用，原块 5 及其后面的块向后错位。 - 通过参数 84.25 ~ 84.29，对空位 5 进行编程。	
	DELETE	删除放置在由参数 <a href="#">83.03</a> 定义的位置上的块，然后，下面的块进行补位。	

索引	参数名称 / 值	说明	FbEq																																				
83.03	EDIT BLOCK 1 ... 20	给参数 <a href="#">83.02</a> 的命令定义块位置号。 块位置号。																																					
83.04	TIMELEVEL SEL 12 ms 100 ms 1000 ms	选择程序的执行周期。该设置对所有块都有效。 12 毫秒 100 毫秒 1000 毫秒																																					
84	ADAPTIVE PROGRAM	自定义编程的创建或诊断。																																					
84.01	STATUS	显示程序状态字的值。下表显示了可选位的状态以及在控制盘上显示的对应值。 <table border="1" data-bbox="484 707 1232 909"> <thead> <tr> <th>位 3</th> <th>位 2</th> <th>位 1</th> <th>位 0</th> <th>显示</th> <th>含义</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>已经停止</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>正在运行</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>2</td> <td>正在编辑</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>4</td> <td>正在检查</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>8</td> <td>发生故障</td> </tr> </tbody> </table>	位 3	位 2	位 1	位 0	显示	含义	0	0	0	0	0	已经停止	0	0	0	1	1	正在运行	0	0	1	0	2	正在编辑	0	1	0	0	4	正在检查	1	0	0	0	8	发生故障	
位 3	位 2	位 1	位 0	显示	含义																																		
0	0	0	0	0	已经停止																																		
0	0	0	1	1	正在运行																																		
0	0	1	0	2	正在编辑																																		
0	1	0	0	4	正在检查																																		
1	0	0	0	8	发生故障																																		
84.02	FAULTED PAR	指出自定义编程中的错误参数。																																					
84.05	BLOCK1 ABS ADD AND COMPARE EVENT FILTER MAX MIN MULDIV NO OR PI PI-BAL SR SWITCH-B SWITCH-I TOFF TON TRIGG XOR	给 Block Parameter Set 1(块参数集 1) 选择功能块。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。 参见“功能块”一章。																																					

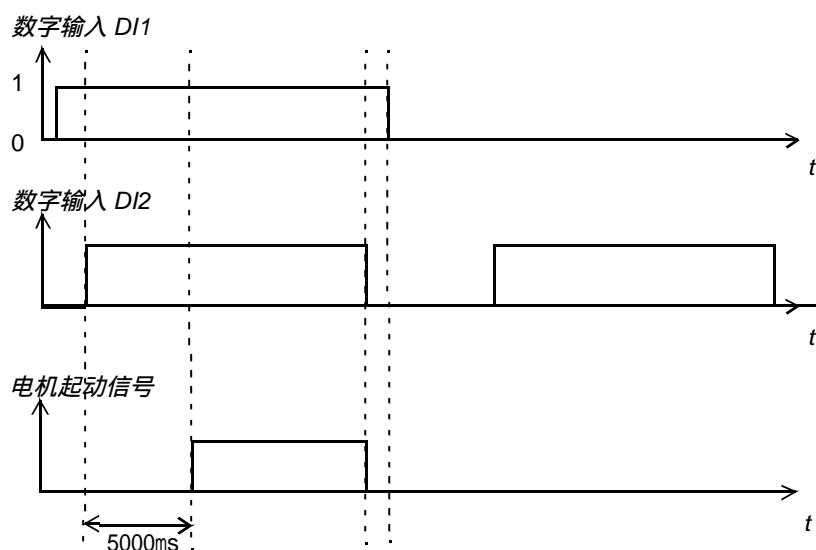
索引	参数名称 / 值	说明	FbEq
84.06	INPUT1 -255.255.31 ... +255.255.31 / C.-32768 ...C.32468	给块参数集 1 (BPS1) 的输入端 I1 选择信号源。  参数指针或常量： - 参数指针：由取反域、组域、索引域和位域组成。其中位域的值仅对块处理布尔输入有效。 - 常量：由取反域和常量域组成。其中取反域的值必须为 C 才能允许对常量进行设置。 <b>例如：</b> 数字输入端 DI2 的状态按如下步骤传到输出端 1： - 设置指针参数 (84.06) 的值为：+01.17.01( 应用程序将数字输入端 DI2 的状态存储到实际信号 01.17 的位 1 )。 - 通过改变指针参数值的符号 (-01.17.01.)，将值取反。	
84.07	INPUT2 -255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参见参数 <a href="#">84.06</a> 。	
84.08	INPUT3 -255.255.31 ... +255.255.31 / C.-32768 ...C.32468	参见参数 <a href="#">84.06</a> 。	
84.09	OUTPUT ...	存储并显示块参数集 1 的输出。 存储并显示块参数集 15 的输出。	
84.79	OUTPUT	存储块参数集 15 的输出。参见参数 <a href="#">84.09</a> 。	
85	USER CONSTANTS	存储自定义编程的常量和信息。	
85.01	CONSTANT1 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.02	CONSTANT2 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.03	CONSTANT3 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.04	CONSTANT4 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.05	CONSTANT5 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.06	CONSTANT6 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.07	CONSTANT7 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.08	CONSTANT8 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.09	CONSTANT9 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.10	CONSTANT10 -8388608 to 8388607	给自定义编程设置一个常量。 整数值	
85.11	STRING1	存储一条信息，该信息用于自定义编程 (EVENT 块)。	

索引	参数名称 / 值	说明	FbEq
	MESSAGE1	信息	
85.12	STRING2	存储一条信息，该信息用于自定义编程(EVENT 块)。	
	MESSAGE2	信息	
85.13	STRING3	存储一条信息，该信息用于自定义编程(EVENT 块)。	
	MESSAGE3	信息	
85.14	STRING4	存储一条信息，该信息用于自定义编程(EVENT 块)。	
	MESSAGE4	信息	
85.15	STRING5	存储一条信息，该信息用于自定义编程(EVENT 块)。	
	MESSAGE5	信息	
96	EXTERNAL AO	通过该组参数，自定义编程可以控制传动单元的可选模拟扩展模块的输出。	
96.01	EXT AO1		
	PARAM 96.11	指向参数 96.11 选择的信号源。	16
96.06	EXT AO2		
	PARAM 96.12	指向参数 96.12 选择的信号源。	16
96.11	EXT AO1 PTR	选择用于 96.01 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 10.04。	
96.12	EXT AO2 PTR	选择用于 96.06 的信号源。	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32468	参数指针或常量。参见参数 10.04。	

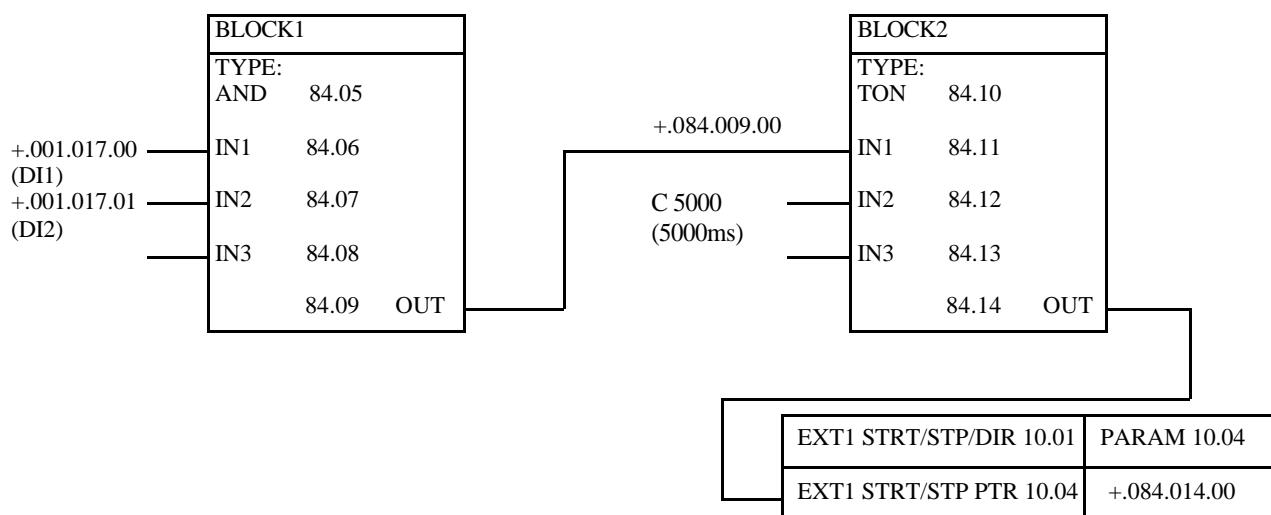
# 自定义编程的例子

## 概述

本章介绍了如何使用控制盘 CDP 312R 编制程序的一个例子。程序的主要功能是两个开关同时控制，延时5秒钟启动电机。任何一个开关处在关断位置时，启动信号无效，如果此时电机在运行状态下，那么就停车。两个开关与 ACS 800 的数字输入口 DI1和DI2相连。



要作自定义编程，先要画出功能框图，如下图。



按下表说明的操作的步骤执行。

步骤	功能	按键	显示
1.	进入参数模式。		1 -> 1242.0 rpm O 10 START/STOP/DIR 01 EXT1 STRT/STP/DIR DI1,2
2.	选择参数组99。		1 -> 1242.0 rpm O 99 START-UP DATA 01 LANGUAGE ENGLISH
3.	在99组内选择参数99.02 APPLICATION MACRO, 确认参数的选项为FACTORY。其它应用宏也同样能使用自定义编程。		1 -> 1242.0 rpm O 99 START-UP DATA 02 APPLICATION MACRO FACTORY
4.	在99组内选择参数99.03 APPLIC RESTORE。		1 -> 1242.0 rpm O 99 START-UP DATA 03 APPLIC RESTORE NO
5.	改变参数值, 将应用宏恢复出厂值, 便于重新自定义编程。如果是修改以前的程序, 那么不需要恢复出厂值。	<b>ENTER</b>	1 -> 1242.0 rpm O 99 START-UP DATA 03 APPLIC RESTORE [NO]
6.	改变参数值为YES, 将应用宏恢复出厂值, 便于重新自定义编程。如果是修改以前的程序, 那么不需要恢复出厂值。		1 -> 1242.0 rpm O 99 START-UP DATA 03 APPLIC RESTORE [YES]
7.	确认所改变的参数值, 将应用宏恢复出厂值。	<b>ENTER</b>	1 -> 1242.0 rpm O 99 START-UP DATA 03 APPLIC RESTORE YES
8.	等待8秒钟。		1 -> 1242.0 rpm O AC800 5.5kW ** WARNING ** MACRO CHANGE
9.	选择参数83.01 ADAPT PROG CMD, 确认参数值为EDIT。	 	1 -> 1242.0 rpm O 83 ADAPT PROG CNTRL 01 ADAPT PROG CMD EDIT
10.	选择参数84.05 BLOCK1, 即选第一个功能块的类型。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 05 BLOCK1 NO
11.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 05 BLOCK1 [NO]

步骤	功能	按键	显示
12.	改变参数值为AND。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 05 BLOCK1 [ AND ]
13.	确认所改变的参数值，将第一个功能块的类型选为AND。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 05 BLOCK1 AND
14.	在84组内选择参数84.06 INPUT1。准备将数字输入口DI1的状态与第一个功能块的第一个输入管脚相连。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 06 INPUT1 +.000.000.00
15.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 06 INPUT1 [+.000.000. <u>00</u> ]
16.	在取反域、参数组域、索引域和位值域选择中来回切换。 激活索引域。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 06 INPUT1 [+.000. <u>000.00</u> ]
17.	调整索引域值为17。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 06 INPUT1 [+.000. <u>017.00</u> ]
18.	激活参数组域。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 06 INPUT1 [+.000. <u>017.00</u> ]
19.	调整参数域值为1。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 06 INPUT1 [+.001. <u>017.00</u> ]
20.	将参数84.06 INPUT1 存储新参数值，+001.017.00，即实际信号1.17 DI6-1 STATUS的DI1的状态与第一个功能块的第一个输入管脚相连。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 06 INPUT1 +.001.017.00
21.	在84组内选择参数84.07 INPUT2。准备将数字输入口DI2的状态与第一个功能块的第二个输入管脚相连。	 	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 07 INPUT2 +.000.000.00
22.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 07 INPUT2 [+.000.000. <u>00</u> ]

步骤	功能	按键	显示
23.	调整参数值为+001.017.01，即将数字输入口DI2的状态与第一个功能块的第二个输入管脚相连。		1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 07 INPUT2 [+.001.017.01]
24.	将参数84.06 INPUT1 存储新参数值，+001.017.01，即实际信号1.17 DI6-1 STATUS的DI2的状态与第一个功能块的第二个输入管脚相连。	<b>ENTER</b>	1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 07 INPUT2 +.001.017.01
25.	在84组内选择参数84.10 BLOCK2，第二个功能块的类型。		1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 10 BLOCK2 NO
26.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 10 BLOCK2 [NO]
27.	改变参数值为TON，延时启动功能块。		1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 10 BLOCK2 [TON]
28.	确认所改变的参数值，将第一个功能块的类型选为AND。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 10 BLOCK2 TON
29.	在84组内选择参数84.11 INPUT1。准备将第一个功能块的输出管脚与第二个功能块的第一个输入管脚相连。		1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 11 INPUT1 +.000.000.00
30.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 11 INPUT1 [+.000.000.00]
31.	调整参数值为+084.009.00，即将第一个功能块的输出管脚参数84.09与第二个功能块的第一个输入管脚相连。	 	1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 11 INPUT1 [+.084.009.00]
32.	将参数84.09 INPUT2 存储新参数值，+084.009.00。	<b>ENTER</b>	1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 11 INPUT1 +.084.009.00
33.	在84组内选择参数84.12 INPUT2。准备将第二个功能块的第二个输入管脚与一个时间常数相连。		1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 12 INPUT2 +.000.000.00

步骤	功能	按键	显示
34.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 84 ADAPTIVE PROGRAM 12 INPUT2 [+.000.000.00]
35.	在取反域、参数组域、索引域和位值域选择中来回切换。 激活最左边的取反域。		1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 12 INPUT2 [+.000.000.00]
36.	调整取反域值为C。		1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 12 INPUT2 [C 00000]
37.	用双箭头键可以激活数值域中的千位。		1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 12 INPUT2 [C 00000]
38.	调整数值域中的千位为5, 设置常数为5000, 相当于5000毫秒, 也就是5秒钟。		1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 12 INPUT2 [C 05000]
39.	将参数84.12 INPUT2 存储新参数值, C 05000。	<b>ENTER</b>	1 ->1242.0 rpm O 84 ADAPTIVE PROGRAM 12 INPUT2 C 05000
40.	选择参数10.04 EXT1 STRT PTR , 即选外部控制地1的起停信号源的指针。	 	1 -> 1242.0 rpm O 10 START/STOP/DIR 04 EXT1 STRT PTR +.000.000.00
41.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 10 START/STOP/DIR 04 EXT1 STRT PTR [+.000.000.00]
42.	调整参数值为+084.014.00, 即将第二个功能块的输出管脚参数84.14与第二个功能块的第一个输入管脚相连。	 	1 ->1242.0 rpm O 10 START/STOP/DIR 04 EXT1 STRT PTR [+.084.014.00]
43.	将参数10.04 EXT1 STRT PTR 存储新参数值, +084.014.00。这样将第二个功能块的输出与起停信号源的指针建立了联系。	<b>ENTER</b>	1 L ->1242.0 rpm O 10 START/STOP/DIR 04 EXT1 STRT PTR +.084.014.00
44.	在第10参数组内选择参数10.01 EXT1 START/STOP , 即选外部控制地1的起停信号源。		1 -> 1242.0 rpm O 10 START/STOP/DIR 01 EXT1 START/STOP DI1,2

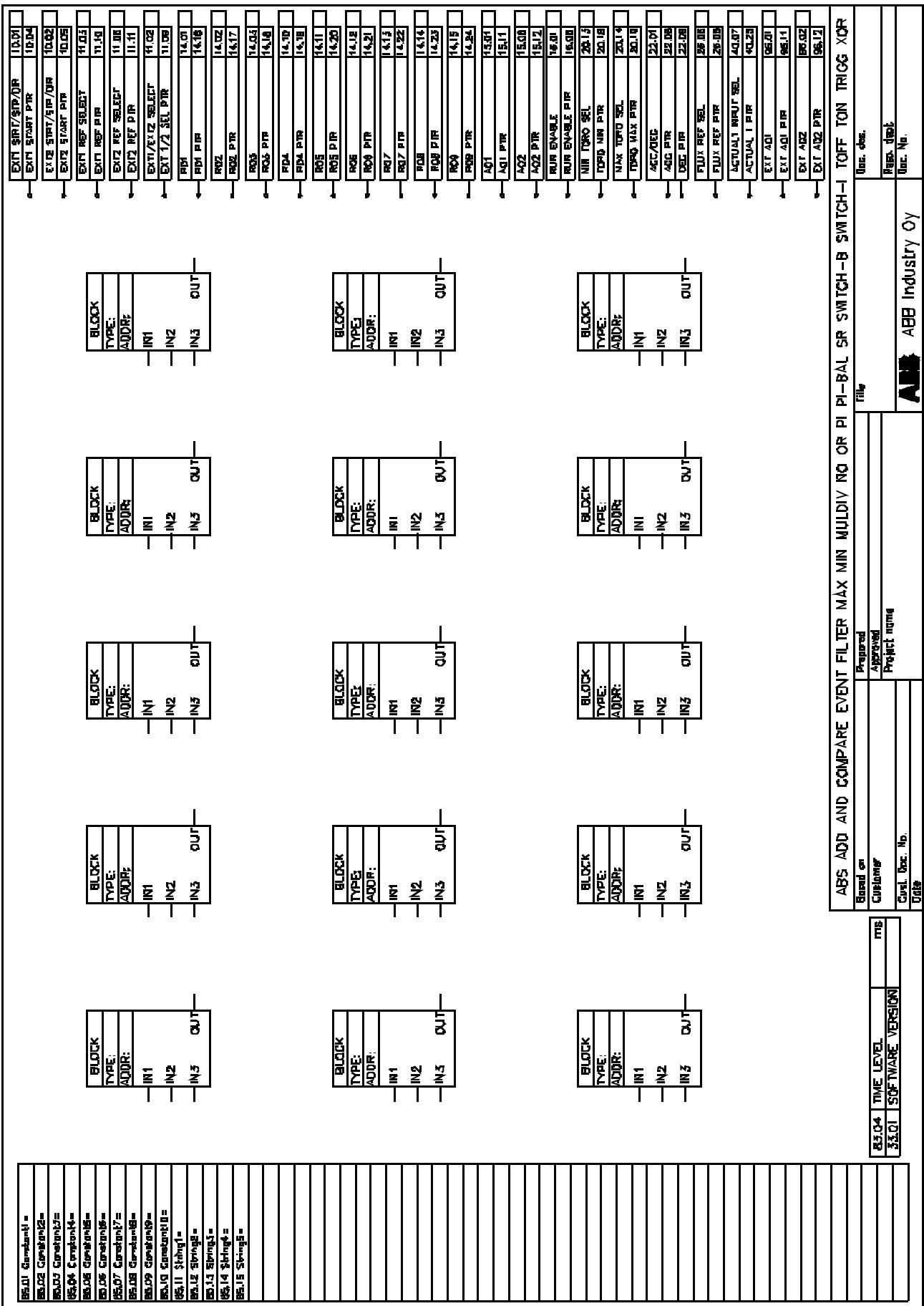
步骤	功能	按键	显示
45.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 10 START/STOP/DIR 01 EXT1 START/STOP [DI1,2]
46.	调整参数值为PARAM 10.04，即将外部控制地1的起停信号源由原来的数字输入口DI1，方向DI2，改为指针参数来间接指定信号源，本例中就是第二个功能块的输出端。		1 ->1242.0 rpm O 10 START/STOP/DIR 01 EXT1 START/STOP [PARAM 10.04]
47.	将参数10.01 EXT1 START/STOP 存储新参数值，PARAM 10.04。这样将第二个功能块的输出与起停信号源建立了完整的联系。	<b>ENTER</b>	1 L ->1242.0 rpm O 10 START/STOP/DIR 01 EXT1 START/STOP PARAM 10.04
48.	选择参数83.01 ADAPT PROG CMD。	 	1 -> 1242.0 rpm O 83 ADAPT PROG CNTRL 01 ADAPT PROG CMD EDIT
49.	激活参数值。	<b>ENTER</b>	1 -> 1242.0 rpm O 83 ADAPT PROG CNTRL 01 ADAPT PROG CMD [EDIT]
50.	调整参数值为START，即准备将启动自定义编程的功能。		1 ->1242.0 rpm O 83 ADAPT PROG CNTRL 01 ADAPT PROG CMD [START]
51.	将参数83.01 ADAPT PROG CNTRL 存储新参数值，START。这样将正式启动自定义编程的功能。	<b>ENTER</b>	1 ->1242.0 rpm O 83 ADAPT PROG CNTRL 01 ADAPT PROG CMD START

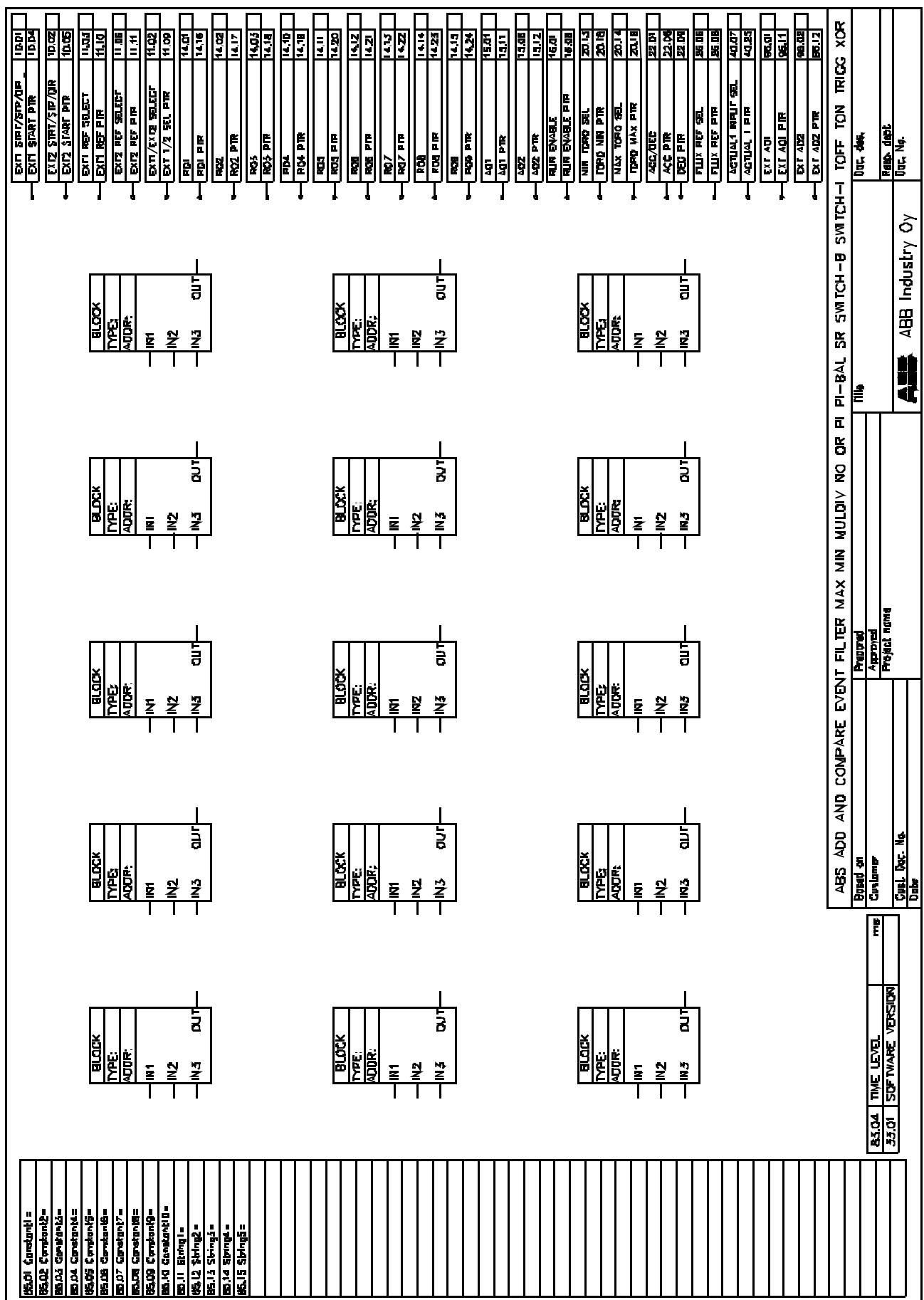
## 用户图

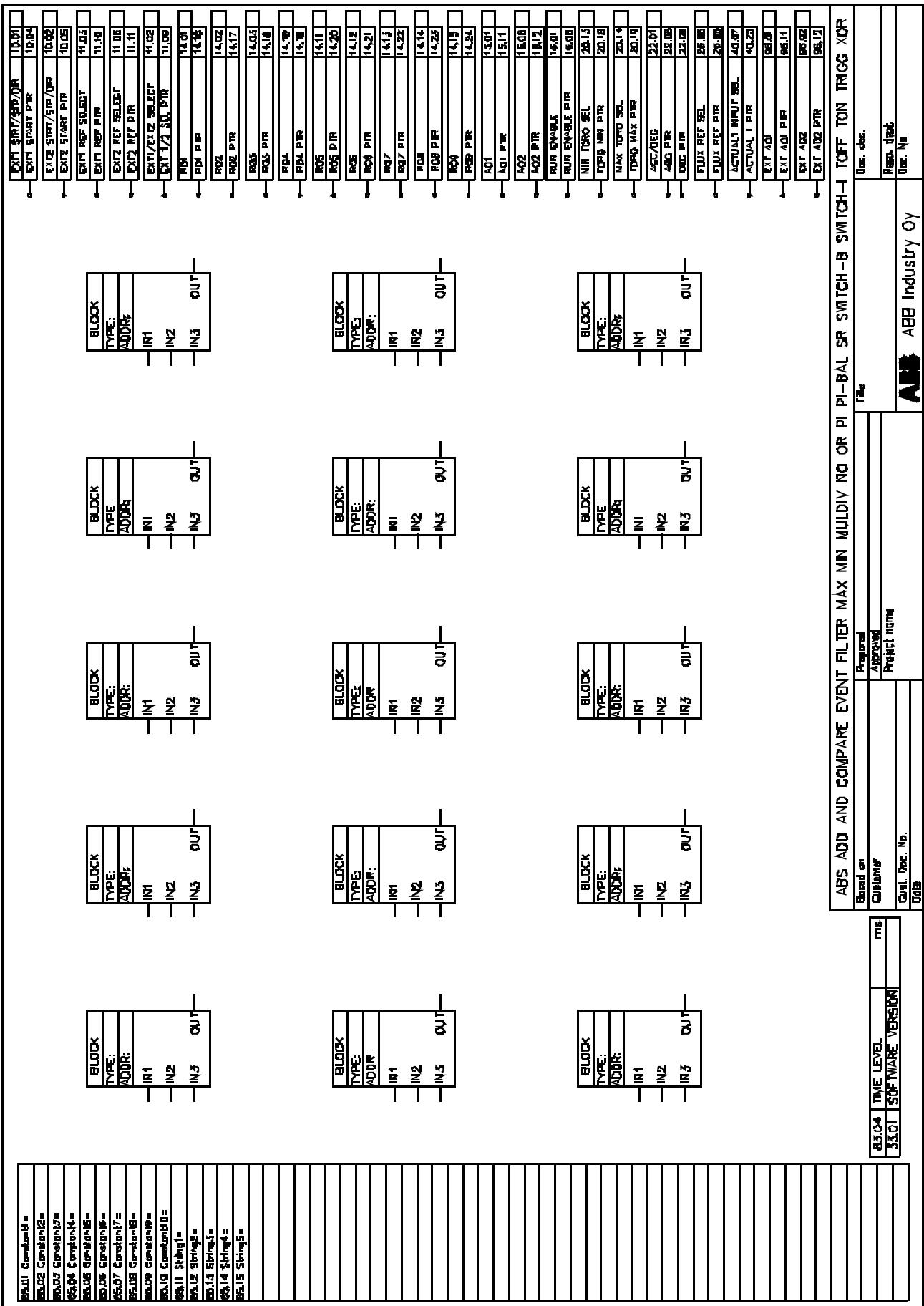
---

### 概述

本章包括 3 张空白方框图页，在空白页上用户可以记录程序。











---

北京 ABB 电气传动系统有限公司

北京市朝阳区酒仙桥路 10 号恒通广厦

邮政编码 : 100016

电话 : +86 10 84566688

传真 : +86 10 84567637

24 小时 x365 天咨询热线 : +86 10 67871888/67876888

网址 : <http://www.abb.com/motors&drives>

3ABD00009804 REV A  
PDM:30004987  
BASED ON:SAFE 64527274 REV A  
EFFECTIVE DATE: 2002-06-21