

Welcome to YottaEditor

YottaEditor is a graphical tool to help users configure 1/5-Series controllers. These configurations include: writing circuit programs, transferring data between PC and 1/5-Series, setting communication ports parameters, and more.

You can run YottaEditor on Windows XP/2000/2003/Vista/7.

Recommended System Requirements

Items	Recommended System Requirements
CPU	1 GHz 32-bit (x86) or 64-bit (x64) processor
Main Memory	At least 512 MB of memory
Hard Drive	1 GB of space with at least 300 MB of available space
Monitor	At least 640 x 480 with full color
Mouse	Windows compatible
RS-232/485 Port	COM1 to COM256

This document introduces the operation of YottaEditor. If you are not familiar with this application, please go to [user interface](#) before continuing this documentation for more information.

You can learn how to create a circuit program, edit the layout, save documentation and simulate the program in [tutorial](#).

参考信息

某些参数值可引用由程序中其他功能块的实际值。

■ 1/2 系列：

- 模拟量比较器：Ax - Ay
- 模拟量阈值触发器：Ax
- 模拟量放大器：Ax
- 加/减计数器：Cnt

■ 5/6 系列：

- 模拟量比较器：Ax - Ay
- 模拟量阈值触发器：Ax
- 模拟量放大器：Ax
- 加/减计数器：Cnt
- 接通延时定时器：Ta
- 关断延时定时器：Ta
- 接通/关断延时定时器：Ta
- 保持接通延时定时器：Ta
- 脉宽继电器(脉冲输出)：Ta
- 沿触发脉宽继电器(脉冲输出)：Ta
- 异步脉冲发生器：Ta
- 楼梯照明开关：Ta
- 多功能开关：Ta
- 秒表：AQ
- 频率阈值触发器：Fre
- 算术指令：AQ
- Analog multiplexer: AQ
- 模拟量斜坡函数发生器：AQ
- PI控制器：AQ
- 最大值/最小值：Ax
- 模拟量滤波器：AQ
- 平均值：AQ
- 二-十进制编码：AQ
- 二进制码：AQ
- 左旋转：AQ
- 右旋转：AQ
- 左移位：AQ

- 右移位：AQ
- AND (与屏蔽)：AQ
- OR (或屏蔽)：AQ
- NOT (非屏蔽)：AQ
- NAND (与非屏蔽)：AQ
- NOR (或非屏蔽)：AQ
- XOR (异或屏蔽)：AQ
- 数据阵列最大/最小/平均：AQ
- 数据最大/最小/平均：AQ
- 随机数发生器：AQ
- 模除：AQ
- 余数：AQ
- 对数：AQ
- 平方根：AQ
- 绝对值：AQ
- 最大公因子：AQ
- 最小公倍数：AQ
- POW2：AQ
- EXP：AQ
- 定点运算：AQ
- 四舍五入：AQ
- 正弦函数：AQ
- 余弦函数：AQ
- 正切函数：AQ
- 余切函数：AQ
- 正割函数：AQ
- 余割函数：AQ
- MEM：AQ
- 二次方程式：AQ
- 编码器：Cnt

数字量输入



输入块为全系列控制器的输入端子，您可使用高达256个数字量输入。

并依主站及扩展模块的区别，可依需求设定下列8种不同的数字量设定：

模块	输入号码
主站	I000 ~ I031
扩展. 1	I100 ~ I131
扩展. 2	I200 ~ I231
扩展. 3	I300 ~ I331
扩展. 4	I400 ~ I431
扩展. 5	I500 ~ I531
扩展. 6	I600 ~ I631
扩展. 7	I700 ~ I731

在程序中，每个数字量输入块皆拥有独立的编号。

文本显示功能键



ATP模块拥有五个功能键，您可依需求将这些功能键用于程序中的数字量输入。

这些功能键的使用方式与程序中的其他输入相同，功能键亦可保存输入及关闭，且允许用户来对程序进行控制。

移位寄存器位



1/2系列控制器提供16个移位寄存器位S0到S15，5/6系列控制器提供高达64个移位寄存器位S0.0到S3.15，在程序中移位寄存器位只“读”。仅能透过移位寄存器这一特殊功能修改移位寄存器位的内容。

永久逻辑电平

逻辑 1 (高)



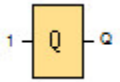
将块输入置位为逻辑1(高)，可让其永久置位为逻辑(高)的状态。

逻辑 0 (低)



将块输入置位为逻辑0(低)，可让其永久置位为逻辑(低)的状态。

数字量输出



输出块为全系列控制器的输出端子，您可使用高达128个数字量输出。

并依主站及扩展模块的区别，可依需求设定下列8种不同的数字量设定：

模块	输入号码
主站	Q000 ~ Q015
扩展. 1	Q100 ~ Q115
扩展. 2	Q200 ~ Q215
扩展. 3	Q300 ~ Q315
扩展. 4	Q400 ~ Q415
扩展. 5	Q500 ~ Q515
扩展. 6	Q600 ~ Q615
扩展. 7	Q700 ~ Q715

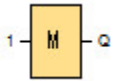
在程序中，每个数字量输出块皆拥有独立的编号。

开放连接器



将未使用块(如消息文本)的输出与‘开放连接器’连接使用，您可使用高达128个开放连接器。

标志位



模拟量标志



标志块用来输出其输入之信号，全系列控制器皆提供高达512组数字量标志(M0~M511)，以及512组模拟量标志(AM0~AM511)。

在您程序的块配置中，可以赋予一个新编号予标志块，且这个标志编号需尚未使用过。

输出始终传送上一个程序周期中的信号，这个值在目前的程序周期内不会改变。

模拟量标志：AM0-AM511

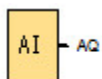
模拟量标志可以当作模拟量输入或是仿真量指令块的标记，且仿真量标志只能作为输入接受模拟量值并输出该值。

消息文本字符标志：M511 (仅供5/6系列控制器使用)

M511 标志（如果使用）确定是显示第一级字符集的消息文本还是显示第二级字符集的消息文本。您可以通过5/6系列控制器的消息配置菜单或Editor的文件 -> 消息文本设置菜单命令选择这两种字符集。然后，在配置消息文本时，可以选择某特定的消息文本是由第一级字符集（字符集 1）中的字符组成还是由第二级字符集（字符集 2）中的字符组成。

在电路程序中，可以使用 M511 启用第一级字符集或第二级字符集的消息文本并禁用另一个字符集的消息文本。当 M511 = 0（低）时，5/6系列控制器只会显示第一级字符集的消息文本。当 M511 = 1（高）时，5/6系列控制器只会显示第二级字符集的消息文本。

模拟量输入



模拟量输入块为全系列控制器的模拟量输入端子，您可使用高达64个模拟量输入。并依主站及扩展模块的

区别，可依需求设定下列8种不同的模拟量设定：

在程序中，每个模拟量输入块皆拥有独立的编号

模块	输入号码
主站	AI000 ~ AI007
扩展. 1	AI100 ~ AI107
扩展. 2	AI200 ~ AI207
扩展. 3	AI300 ~ AI307
扩展. 4	AI400 ~ AI407
扩展. 5	AI500 ~ AI507
扩展. 6	AI600 ~ AI607
扩展. 7	AI700 ~ AI707

控制器的模拟量输入端子，除可当作模拟量输入使用外，亦可当作数字量输入使用，用户无需于操作过程中设定任何参数，但得依下列表格规范来使用：

	1/5 Series Input	2/6 Series Input
模拟量	0 ~ 10V	0 ~ 10V
数字量	低位准 : < +2VDC 高位准 : > 4VDC~30VDC	AC : 低位准 : < 40VAC 高位准 : > 79VAC DC : 低位准 : < 5VDC 高位准 : > 8.5VDC

Example

In 1x89 / 2x89 / 5x89 / 6x89 Series, an input signal, which comes from AI0, is also detected on DI4.

模拟量输出



模拟量输出块为全系列控制器的模拟量输出端子，您可使用高达32个模拟量输出。并依主站及扩展模块的区别，可依需求设定下列8种不同的模拟量设定：
在程序中，每个模拟量输出块皆拥有独立的编号

模块	输出号码
主站	AQ000 ~ AQ003
扩展. 1	AQ100 ~ AQ103
扩展. 2	AQ200 ~ AQ203
扩展. 3	AQ300 ~ AQ303
扩展. 4	AQ400 ~ AQ403
扩展. 5	AQ500 ~ AQ503
扩展. 6	AQ600 ~ AQ603
扩展. 7	AQ700 ~ AQ703

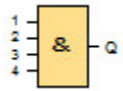
备注：模拟量输出只能处理0~1000间的值

5/6系列控制器可以配置Init模式下的模拟量输出行为，模拟量输出可以在进入Init模式时保持其最后值。您也可以配置Init模式时将AQ设置为特定值。

下列二种选择，可以设置模拟量输出的范围值：

- 0-10 V 或 0-20 mA (常规)
- 4-20 mA

AND(与)



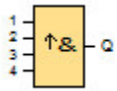
仅当所有输入都为1时，AND(与)的输出才会为1。

未使用的块输入(X)分配：X=1

AND(与)功能逻辑表

输入1	输入2	输入3	输入4	输出
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

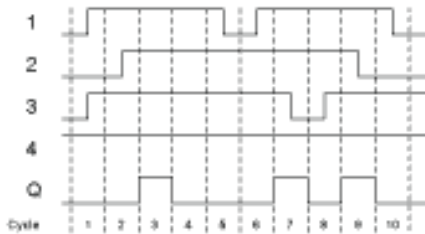
带边缘评估的AND(与)



当本周期中所有输入皆为1，且在上一个周期中至少有一个输入为0，带边缘评估的AND(与)输出才为1。在一个周期的持续时间内输出置位为1且必须在下一个周期的持续时间内复位为0，如此输出才能再一次置位为1。

未使用的块输入(X)分配：X=1

带边缘评估的AND(与)时序图：



NAND(与非)



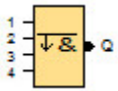
当所有输入皆为1时，NAND(与非)的输出才为0。

未使用的块输入(X)分配：X=1。

NAND(与非)功能逻辑表：

输入1	输入2	输入3	输入4	输出
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

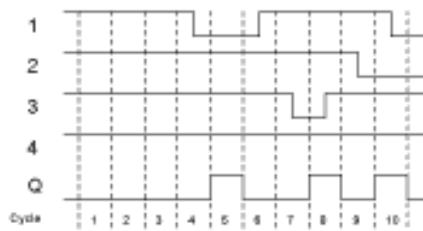
带边缘评估的NAND(与非)



在一个周期的持续时间内输出置位为1，且必须至少在下一个周期的持续时间内复位为0，这样输出才能再次置位为1。

未使用的块输入(X)分配：X=1。

带边缘评估的NAND(与非)时序图：



OR(或)



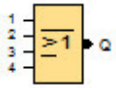
如果至少有一个输入为1，则OR(或)的输出为1。

未使用块输入(X)分配：X=0。

OR(或)功能逻辑表：

输入1	输入2	输入3	输入4	输出
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

NOR(或非)

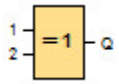


当所有输入皆为0时，NOR(或非)的输出才为1。当其中一个输入为1时，则输出为0。
未使用的块输入(X)分配：X=0。

NOR(或非)功能逻辑表：

输入1	输入2	输入3	输入4	输出
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

XOR(异或)



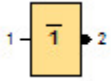
当输入的信号状态不同，则XOR(异或)的输出为1。

未使用的块输入(X)分配：X=0。

XOR(异或)功能逻辑表：

输入1	输入2	输出
0	0	0
0	1	1
1	0	1
1	1	0

NOT(非，反相器)



当输入为0，则输出为1。NOT(非，反相器)会使输入状态反相。

NOT(非，反相器)使用时机：不需再使用常闭触点，仅需使用常开触点，藉由NOT(非，反相器)将常开触点转换为常闭触点。

NOT(非，反相器)功能逻辑表：

输入1	输出
0	1
1	0

接通延时定时器



接通延迟的输出值会在超过配置的延迟时间后才执行。

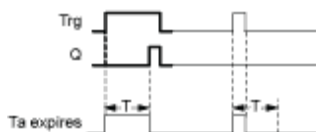
连接	描述
输入 Trg	接通延迟时间透过Trg(触发器)输入来触发。
参数	T：表示输出打开(输出信号由0变成1)之前的接通延迟时间。 可保持性：'勾选' 该字段则表示该状态在存储器中可以保持。
输出Q	如果对Trg输入进行置位，则Q会在超出指定的时间T之后执行

参数T

5/6系列控制器，参数T中的时间可以由其他已编程功能的实际值来提供。

请参考'引用'。

时器图：



功能说明

触发时间 T_a 为输入Trg由0变成1。如果输入Trg的状态在配置的时间T内保持为1，当超出该时间后(输出的接通信号比输入的接通信号延迟此延迟时)，输出将置位为1。

当在超出时间T之前输入Trg的状态再次变为0，则复位为该时间。

当输入Trg为0时，则输出会复位为0。

关断延时定时器



带有断开延迟的输出会在超出定义的时间后才会复位。

连接	描述
输入Trg	透过输入Trg(触发器)的负跳沿(1变0)，启动断开延迟时间。
输入R	复位断开延迟时间，且透过R(复位)输入将输出置位为0。 ‘复位’的优先级高于‘触发器’。
参数	T ：超出延迟时间T时，输出会关闭(输出信号由1变成0) 可保持性 ：’勾选’该字段则表示该状态在存储器中可以保持。
输出Q	触发器输入Trg后，Q在时间T内保持打开(1)的状态

参数T

5/6系列控制器，参数T中的时间可以由其他已编程功能的实际值来提供。

请参考‘引用’。

时序图



功能说明

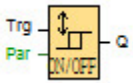
输入Trg由0变成1后，输出Q会立即置位为1。

输入Trg由1变成0后，将重新触发目前时间T，且输出仍会保持置位状态。当Ta到达T(Ta=T)(断开延迟)中设定的值时，输出Q将复位为0。

单次触发输入Trg将重新触发时间Ta。

在超过时间 T_a 前，可以透过输入R(复位)复位输出和时间。

接通/断开延时定时器



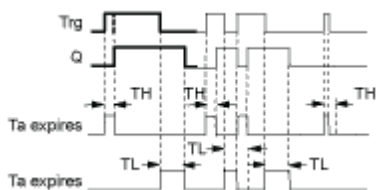
接通/断开延迟使用于配置的接通延迟时间之后置位输出，然后在超出下一个配置的时间时再次将其复位。

连接	描述
输入Trg	透过输入Trg(触发器)的正跳沿(0变成1)触发接通延迟。 透过负跳沿(1变成0)触发断开延迟。
参数	T_H 为输出的接通延迟时间(输出信号由0变成1)。 T_L 为输出的断开延迟时间(输出信号由1变成0)。 可保持性：' 勾选' 该字段则表示该状态在存储器中可以保持。
输出Q	如果Trg处于置位状态，则Q会在超出配置的时间 T_H 时打开(1)。 如果Trg尚未再次置位，则Q会在超出时间 T_L 时再次关闭(0)。

参数TH和TL

5/6系列控制器，参数TH和TL的接通延迟时间和断开延迟时间，可以由其他已编程功能的实际值来提供。
请参考' 引用' 。

时序图



功能说明

输入Trg从0变成1时，触发时间为TH。

如果输入Trg的状态至少在配置的时间TH 内保持为1，则输出将在超过该时间置位为1(相对于输入信号，输出接通延迟)。

如果在超过该时间之前输入Trg的状态复位为0，则将会复位时间TH。

输出由1变成0时，触发时间为TL。

如果输入Trg的状态至少在配置的时间TL 内保持为0，则输出将在超过该时间复位为1(相对于输入信号，输出关闭延迟)。

如果在超过该时间之前输入Trg的状态回到1，则将会复位时间TL。

保持接通延时定时器



单次触发输入将触发可以配置的时间，输出在超过该时间时置位。

连接	描述
输入 Trg	透过 Trg(触发器)输入触发接通延迟时间。
输入 R	复位接通延迟时间，且透过输入 R(复位)将输出复位为 0。 '复位' 的等级高于'触发器'。
参数	T 是输出的接通延迟时间(输出信号由 0 变成 1) 可保持性：'勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	Q 在超过时间 T 时打开(1)。

参数 T

5/6 系列控制器，参数 T 中的时间可以由其他已编程功能的实际值来提供。

请参考'引用'。

时序图



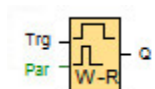
功能说明

输入 Trg 由 0 变成 1 时触发目前时间 T_a ，当 T_a 到达时间 T 时，输出 Q 为 1。输入 Trg 的下一个脉冲不会影响 T_a 。

输出 R 的信号为 1 时，输出和时间 T_a 复位为 0。

假如未对可保持性设置，会使电源故障后将复位输出 Q 和到期的时间。

脉宽继电器(脉冲输出)



输入信号产生拥有可配置长度的输出信号。

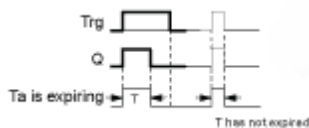
连接	描述
输入Trg	透过输入Trg(触发器)的信号触发脉宽触发继电器的时间。
参数	T是输出复位(输出信号由1变成0)之前的时间。 可保持性：'勾选' 该字段则表示该状态在存储器中可以保持。
输出Q	Trg的脉冲置位Q。当未超出时间T，且该时间内Trg为1的话，则输出会保持置位状态。Trg在超过T之前，从1变成0也会将输出复位为0。

参数 T

5/6系列控制器，参数T中的时间可以由其他已编程功能的实际值来提供。

请参考'引用'。

时序图



功能说明

当输入信号Trg=1时，输出Q=1。输出Q保持置位时，输入信号也会一并触发时间Ta。

假如Ta到达T(Ta=T)中定义的值时，输出Q会复位为0状态(脉冲输出)。

如在超过该时间之前输入Trg的信号由1变成0，则输出会立刻从1复位为0。

沿触发脉宽继电器 (脉冲输出)



组态的延迟时间到达后，输入脉冲会依设定的脉冲/间歇比(可重新触发)产生预置数量的输出脉冲。

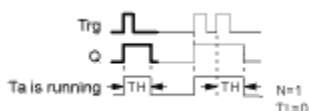
连接	描述
输入 Trg	透过输入Trg(触发器)的信号为边缘触发的脉冲继电器触发时间
输入 R	当输入R的信号等于1时，输出和目前时间Ta复位为0。
参数	<p>中介脉冲宽度 T_L 以及脉冲宽度 T_H 为可调整。</p> <p>N 可确认脉冲/间歇周期 T_L / T_H 的数量。值范围：1~9。</p> <p>可保持性：' 勾选' 该字段则表示该状态在存储器中可以保持。</p>
输出 Q	输出Q会在超过时间TL时置位，在超过TH时复位。

参数 TH and TL

5/6系列控制器，参数TH 以及 TL的时间可以由其他已编程功能的实际值来提供。

请参考' 引用' 。

时序图



功能说明

当输入Trg变成1时，会触发时间 T_L (低电平时间)，当超过时间 T_L 之后，会在时间 T_H (高电平时间)内将输出Q置位为1。

如在超过预定时间(T_L+T_H)之前重新触发输入Trg，则时间Ta会复位，而且脉冲/间歇周期会重新启动。

异步脉冲发生器



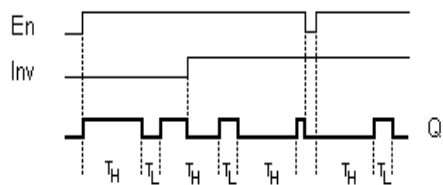
可以透过可配置的脉冲/间歇比更改输出的脉冲形状

连接	描述
输入 En	可以藉由输入En的信号启动/禁用异步脉冲发生器。
输入 Inv	输入Inv可以用来让活动的异步脉冲发生器的输出信号反向。
参数	可依需求设定脉冲宽度(TH)和中介脉冲宽度(TL)。 可保持性： '勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	输出Q会依脉冲/间歇时间的周期来打开(1)和关闭(0)

参数 TH 及 TL

5/6系列控制器，参数 T_H 以及 T_L 的时间可以由其他已编程功能的实际值来提供。
请参考'引用'。

时序图



功能说明

使用者可藉由 T_H (高电平时间)以及 T_L (低电平时间)来设定脉冲/间歇比。

输入Inv用于输出的信号反向，且当通过En启动输入块Inv以后，该块才会使输出信号反向。

随机通/断定时器



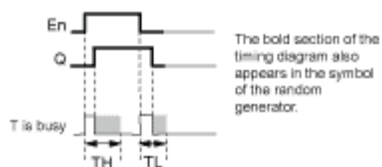
随机发生器所生成的输出可以在设定的时间内切换。

连接	描述
输入 En	启动输入En的正跳沿(0变成1)，将随机发生器触发接通延迟。 负跳沿(1变成0)，将随机发生器触发断开延迟。
参数	T_H ：接通延迟随机产生，位于0 sec到 T_H 间。 T_L ：关闭延迟随机产生，位于0 sec 到 T_L 间。
输出 Q	输入En如果置位，则在超过接通延迟时置位输出Q。 当超过断开延迟时间，而且未再次置位输入En时复位输出Q。

参数 TH 及 TL

5/6系列控制器，参数 T_H 以及 T_L 的时间可以由其他已编程功能的实际值来提供。
请参考‘引用’。

时序图

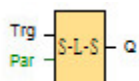


功能说明

当输入En由0变成1时，会置位及触发一个位于0 sec到 T_H 间的随机时间(接通延迟时间)，如果输入En的状态在接通延迟时间内持续为1，当超过该接通延迟时间后输出会置位为1。
如果在超过接通延迟时间前输入En的状态复位为0，则会复位该时间。

当输入En复位为0时，会置位及触发一个位于0 sec到 T_L 间的随机时间(断开延迟时间)，如果输入En状态在断开延迟时间内持续为0，当超过该断开延迟时间输出Q会复位为0。
如果在超过接通延迟时间前输入En的状态复位为1，则会复位该时间。

楼梯照明开关



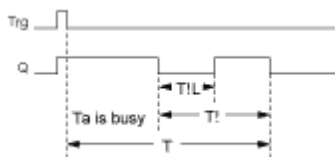
输入脉冲的边缘触发一段可配置的时间，当超过该时间时复位输出，在超过该时间前可以输出关闭警告。

连接	描述
输入 Trg	使用者可藉由输入Trg(触发器)的信号启动楼道照明开关触发时间(断开延迟)。
参数	<p>T : 超过断开延迟时间T时复立输出(1变成0)。</p> <p>$T!$: 确定预警触发时间。</p> <p>T_{IL} : 确定预警时间长度。</p> <p>可保持性：'勾选' 该字段则表示该状态在存储器中可以保持。</p>
输出 Q	超过时间 T 之后复位输出 Q ，在超过该时间前可以输出警告信号。

参数 T , $T!$ 以及 T_{IL}

5/6系列控制器，参数 T ， $T!$ 以及 T_{IL} 的时间可以由其他已编程功能的实际值来提供。
请参考'引用'。

时序图



时基 T	预警时间	预警周期
秒钟*	750 ms	50 ms
分钟	15 s	1 s
小时	15 min	1 min

* 周期时间仅适用于小于25ms的程序

功能说明

当输入Trg由0变成1时，输出Q会置位为1。当输入Trg由1变成0时会触发当前时间，且输出Q仍会保持置位状态。

当Ta时间到达时间T时，输出Q会复位为0。在超过断开延迟时间($T - T_1$)前，使用者可输出预警，然后将在

关闭预警时间 T_{IL} 内将输出Q复位。

如在输入Trg产生下一个高/低价跃时且超过时间Ta，会重新触发Ta(可供选择)。

多功能开关



多功能开关具有下列二种功能：

- 带有断开延迟的脉冲开关
- 开关(连续照明)

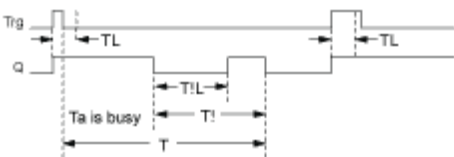
连接	描述
输入 Trg	使用者可藉由输入Trg(触发器)的信号置位输出Q(连续照明)，或是透过断开延迟复位输出Q。 当输出生效后，可以藉由输入Trg的信号复位输出Q。
输入 R	输入R的信号复位目前时间(Ta)以及输出。
参数	T：确定断开延迟时间，当超过时间T时复位输出(1变成0) TL：确定启动连续照明功能的必须置位输入的时间。 T!：确定预警时间的接通延迟。 T!L：确定预警时间周期的长度。 可保持性：'勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	用户可藉由输入Trg的信号置位输出Q，且在超过配置的时间后依据输入Trg的脉冲宽度再次复位输出Q，也可藉由输入Trg的其他信号复位。

参数 T, T! 以及 T!L

5/6系列控制器，参数T，T! 以及 T!L的时间可以由其他已编程功能的实际值来提供。

请参考'引用'。

时序图



时基 T , T_f 以及 T_{IL} 必须一致。

功能说明

输入 Trg 由 0 变成 1 时，输出 Q 会置位为 1。

如果输出 Q 为 0，且输入 Trg 在至少 T_L 时间内保持置位为高电平，则将启动连续照明功能且输出 Q 会相对地置位。

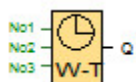
如果在超过时间 T_L 前输入 Trg 的状态变成 0，将触发断开延迟时间 T 。

当 $T_a = T$ 时复位输出 Q 。

使用者可在断开延迟时间失效前输出断开警告信号 $(T - T_f)$ ，再将延续断开预警时间 T_{IL} 复位输出 Q ，输入

Trg 后续的信号会持续复位输出 Q 和时间 T 。

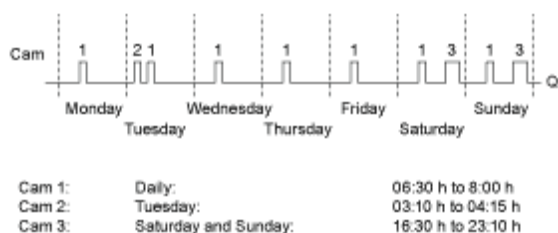
周定时器



使用者可藉由周定时器的打开/关闭日期来控制输出，且周定时器支持工作日的任意组合。

连接	描述
参数	在 No1, No2, No3 时间参数中，可以设置周定时器不同时间段的打开/关闭时间触发器。 而每个时间段，都得为打开及关闭时间指定每周的日期以及每日的时间
输出 Q	当启动组态的时间段时置位输出Q。

时序图



功能说明

每组周定时器提供三个时间段，可将每个单独的时间段设置配置时滞，可以在时间段上设定打开及关闭滞后，周定时器在某一个特定时间上置位输出(如该输出尚未置位)。

如配置了关闭时间，则周定时器会在该关闭时间复位输出，如指定了脉冲输出，则周定时器会在周期结束时复位输出。当其他时间段的打开时间及关闭时间相同时，周定时器会发生冲突。此状况下，时间段3优先于时间段2，时间段2优先于时间段1。

周定时器的切换状态由NO1，NO2以及NO3时间段的状态确定。

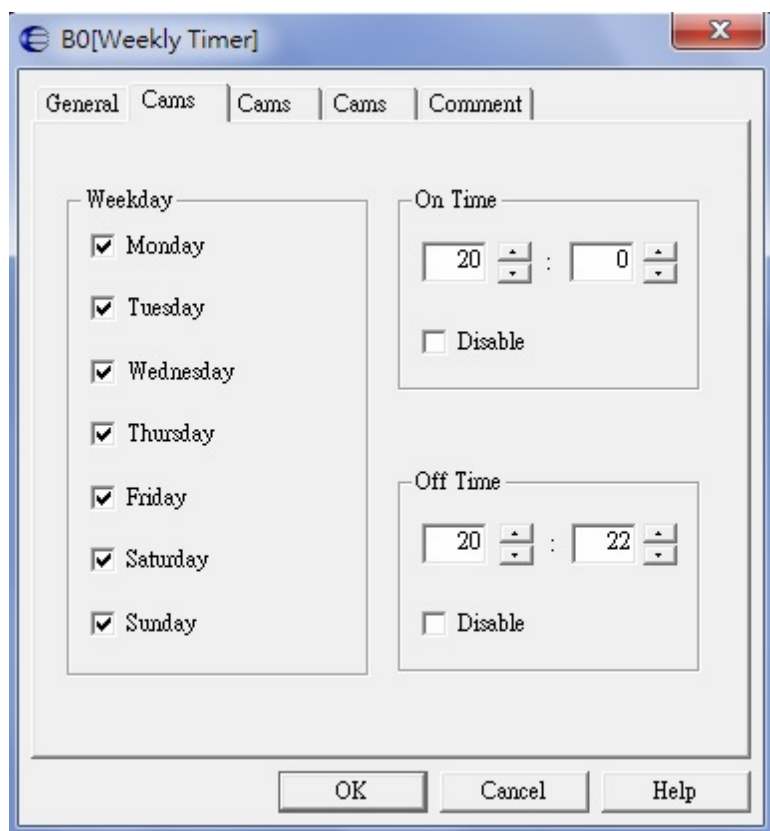
打开时间

打开时间的设置范围为00:00 h 到 23:59 h。

配置时注意事项

块属性窗口的三个时间段分别提供了一个选项卡，用户可以为各时间段设置每周的日期。另每个选项卡还为各时间段提供了小时及分钟为单位的打开及关闭时间选项。最短的时间周期为1分钟。在各选项卡上，还可指定时间段的脉冲输出。

也可依需求禁用打开及关闭的时间。



实时时钟的保持

控制器可针对内部的实时时钟执行缓冲，使在产生电源故障时进行保护机制。

年定时器

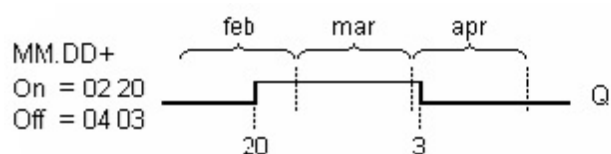


■ 仅供1/2系列控制器使用：

使用者可藉由年定时器的打开/关闭日期来控制输出。

连接	描述
参数	在 No (时间段) 参数中，可配置定时器的模式、打开/关闭时间、输出是否为脉冲输出。
输出 Q	打开配置的时间段时置位输出Q

时序图



功能说明

年定时器会在指定的接通和断开时间置位和复位输出。

打开时间指定了置位定时器的月份及日期，关闭时间时指定了再次复位输出的月份及日期。

当选择每月复选栏时，定时器输出每个月都会在指定的打开/关闭日期执行打开/关闭作业。

实时时钟的保持

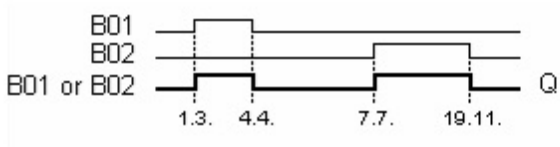
控制器可针对内部的实时时钟执行缓冲，使在产生电源故障时进行保护机制。

配置时注意事项

可以需求设定月份及日期参数，并依日期的逻辑范围输入相关数值，否则软件会显示一个错误讯息。

范例

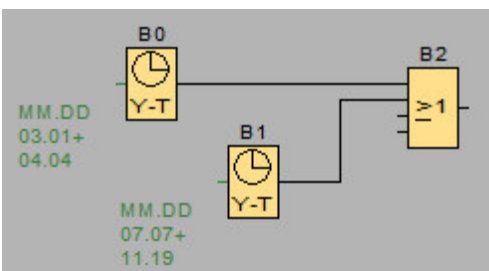
以年为单位的周期性输出，三月一日打开，四月四日关闭，七月七日打开，十一月九日关闭。这需要二个块以配置指定的打开时间，然后藉由OR(或)块连接输出。



于程序里面使用二个年定时器块，并依需求填入年定时器参数。

使用标准OR(或)块连疾这些年定时器的逻辑连接，如果每年定时器的开关中至少一个被置位，则OR(或)的输出为1。

The screenshot shows a configuration window with four sections. The top-left section is for 'On Time' with 'Month.Day' set to '3 : 1'. The top-right section is for 'Off Time' with 'Month.Day' set to '4 : 4'. The bottom-left section is for 'On Time' with 'Month.Day' set to '7 : 7'. The bottom-right section is for 'Off Time' with 'Month.Day' set to '11 : 19'. Each section has a 'Month' and 'Day' input field with up/down arrows and a '...' button.



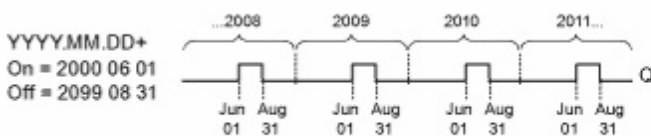
■ 仅供5/6系列控制器使用：

该输出透过可配置的打开/关闭日期来控制，用户可以配置定时器以根据年、月或用户定义时间来激活。也可以针对任一模式来配置定时器，以在定义的时间周期内产生脉冲输出。时间周期范围为2000年1月1日至2099年12月31日。

连接	描述
参数	在 No (时间段) 参数中，可配置定时器的模式、打开/关闭时间、输出是否为脉冲输出。
输出 Q	打开配置的时间段时置位输出Q

时序图

范例1:每年被选中，打开时间为2000年6月1日，关闭时间为2099年8月31日，每年的6月1日定时器开启，8月31日关闭。



范例2：每年及脉冲被选中，打开时间为2000年3月15日，关闭时间为2099年；每年的3月15日定时器会接通并且持续一个星期。

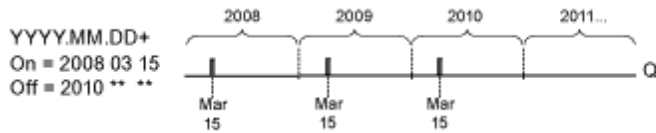


范例3：每年被选中，打开时间为2008年6月1日，关闭时间为2010年8月31日。2008-2010年的6月1日定时器输出会打开，然后在8月31日关闭。

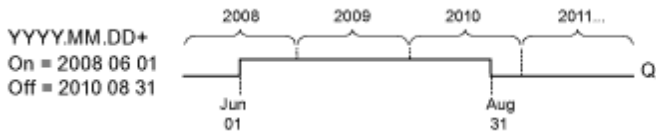


范例4：每年及脉冲被选中，打开时间为2008年3月15日，关闭时间为2010年。2008-2010年的3月15日定时器

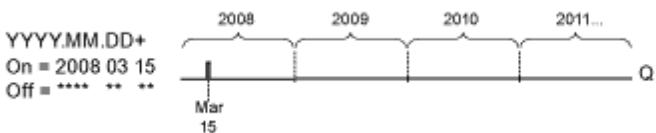
输出会接通并且持续一个星期。



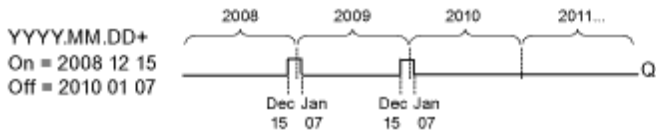
范例5：每月及每年未被选中，打开时间为2008年6月1日，关闭时间为2010年8月31日；2008年6月1日定时器输出会接通，直到2010年8月31日关闭。



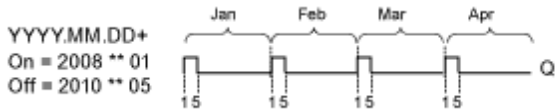
范例6：每月及每年未被选中，脉冲被选中，打开时间为2008年3月15日，关闭时间为设定。2008年3月15日定时器开后，并且持续输出一个周期；因为定时器不具每月动作或每年动作，所以定时器仅会在指定时间输出一次脉冲。



范例7：每年被选中，打开时间为2008年12月15日，关闭时间为2010年1月7日。2008-2009年的12月15日定时器会输出开启，并保持到来年的1月7日；2010年1月7日定时器输出关闭后，不会在来年的12月15日再次开启。



范例8：每月被选中，打开时间为2008年**月1日，关闭时间为2010年XX月5日。从2008年开始，定时器会在每月的第一天输出，在每月的第五天关闭；然后持续运行到2010年的最后一个月。



功能说明

年定时器在指定的接通和断开时间内置位及复位输出。置位和复位皆在00:00时执行。如果应用程序需在不

同的时间执行，需在电路程序中同时使用一个年定时器及一个周定时器。

打开时间指定了置位定时器的月份及日期，关闭时间则指定了再次复位输出的月份及日期。且请注意打开和关闭时间的域顺序：第一个域定义年份，第二个定义月份，第三个定义日期。

选中每月复选框时，定时器会在每个月指定的打开日期内输出，在指定的关闭日期关闭。打开年份指定了激活定时器的初始年份。关闭年份定义了定时器关闭的最后年份，年份的最大上限值为2099年。

如果选中脉冲复选框，定时器会在指定的打开时间输出，并在一个周期后关闭，然后定时器输出会被复位。可以需求选择年、月或某个单独时间产生定时器的脉冲。

如果未选中每月、每年或脉冲复选框中的任何一个，可以透过打开时间和关闭时间定义一个指定的时间周期。且可以跨越所选择的任何时间周期。

如要在一年中的多个不规则时间打开和关闭的过程中操作，可以定义多个年定时器，并将这些输出透过OR(或)功能块连接。

实时时钟的保持

任何一款控制器皆内建电池，支持实时时钟的保持。

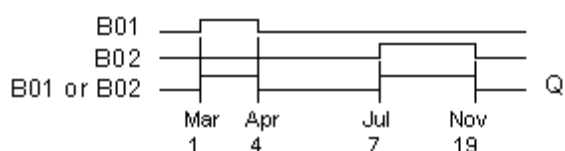
配置时要注意的事项

可以为月份和日期域输入数字量值，但得依逻辑范围内输入，否则软件会显示一个错误消息。

日历图标为设置日期的一种简单方法，它会打开一个窗口，从中可以透过单击日历页上相应的按钮来设置日期及月份。

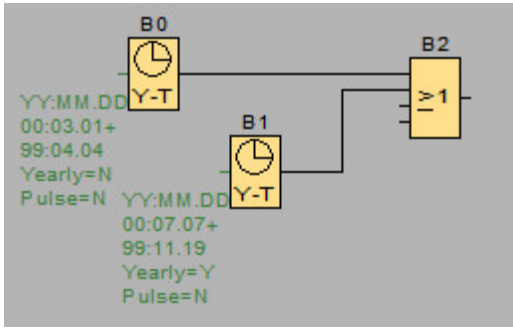
配置示例

年定时器的输出以年为单位周期性开后，从3月1日到4月4日以及从7月7日到11月19日。这需要二个块以配置指定的打开时间，然后透过RO(或)块连接输出。

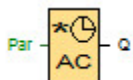


在编程接口上放置二个年定时器开关的功能块，将第一个年定时器的时间配置为3月1日，关闭时间配置为4月4日；第二个年定时器的打开时间配置为7月7日，关闭时间配置为11月19日。

使用OR(或)块创建这些年定时器块的逻辑连接，如果年定时器的开关中至少一个被置位，则OR输出为1。

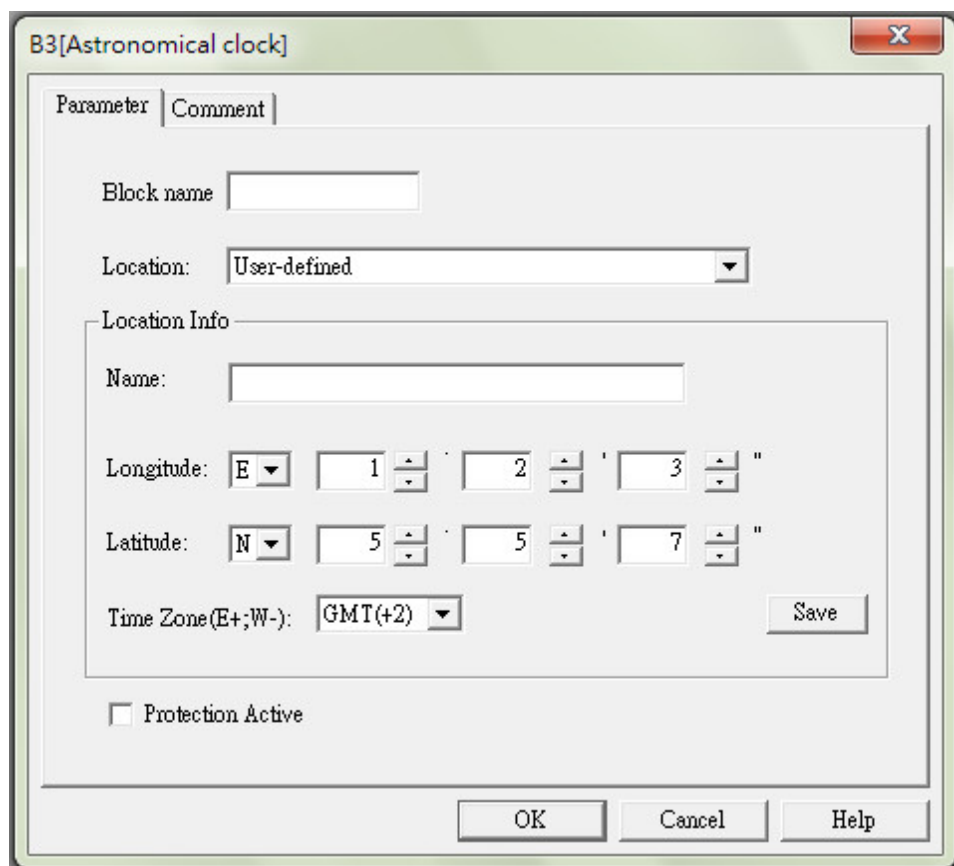


天文时钟



用户可依所处的地理位置时间，使一个输出在日出至日落间保持置位，且该功能的输出状态可配合夏令/冬令时间使用。

连接	描述
参数	地理位置包含了经度、纬度及时区。
输出 Q	当达到日出时间时，输出Q会置位。然后输出会持续保持置位状态，直到日落时间为止。



B3[Astronomical clock]

Parameter | Comment

Block name:

Location:

Location Info

Name:

Longitude: "

Latitude: "

Time Zone(E+;W-):

Protection Active

Save

OK Cancel Help

可依下列默认城市中选取所在位置。

北京

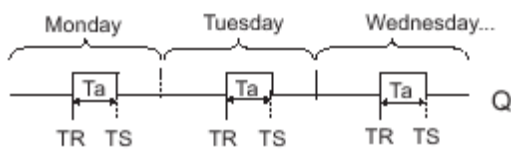
- 柏林
- 伦敦
- 罗马
- 莫斯科
- 东京
- 华盛顿
- 安卡拉
- 马德里
- 阿姆斯特丹

当选定城市位置后，软件即会套用该城市的经纬度及时区。

亦或自行定义所需的经度、纬度及时区，并为该位置命名；软件会依这三项参数计算出当天的绝对日出/日落时间。另使用者如设置了夏令/冬令时间时，这个功能块会一同将夏令/冬令时间加入转换。

如需配置夏令/冬令时间时，仅需勾选‘日期和时间属性’对话框中的复选框‘根据夏令时间改变自动调整时钟’。

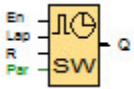
时序图



功能说明

该功能块可以依输入地点的数值来计算，且依设备所处地理位置(经、纬度及时区)的日落时间和日出时间使输出Q置位或复位。

秒表

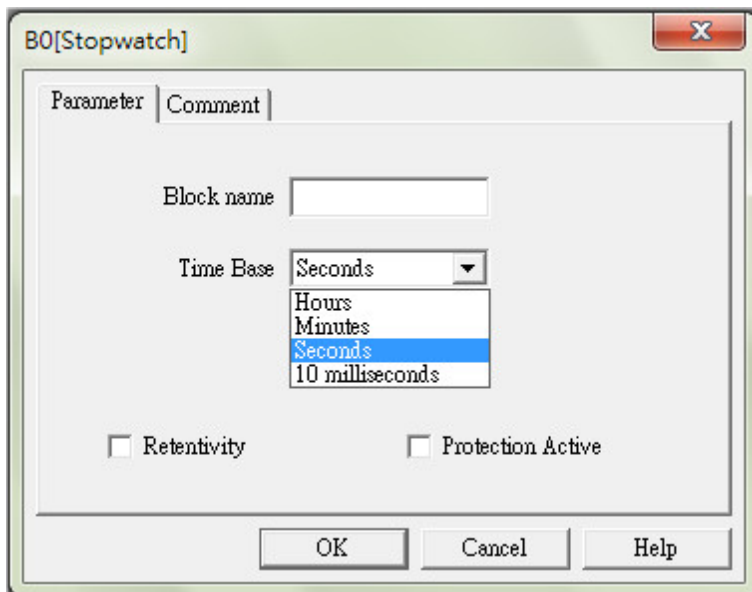


秒表功能块可将启用后所经过的时间完整记录。

连接	描述
输入 En	En (启用)为监控输入。5/6系列控制器的输入EN由0变成1时，会清零秒表当前的读数(已经过的时间)，且重新开始计时；当EN由1变成0时，秒表读数会被冻结。
输入 Lap	输入Lap(暂停)的正跳沿(0变成1)会使秒表暂停计时，且将输出置位为秒表的暂停时间；输入Lap(暂停)上的负跳沿(1变成0)会使秒表恢复计时，且将输出置位秒表当前读数。
输入 R	输入R(复位)上的信号清零秒表当前读数和暂停时间。
参数	可依需求将秒表读数的时基设为小时、分钟、秒或10毫秒(百分之一秒)。
输出 AQ	输入Lap为负跳沿(1变成0)时，输出AQ会输出当前秒表读数；输入Lap为正跳沿(0变成1)时，输出AQ为秒表暂停时间。 输入R上的正跳沿(0变成1)会将输出AQ清零。

時基參數

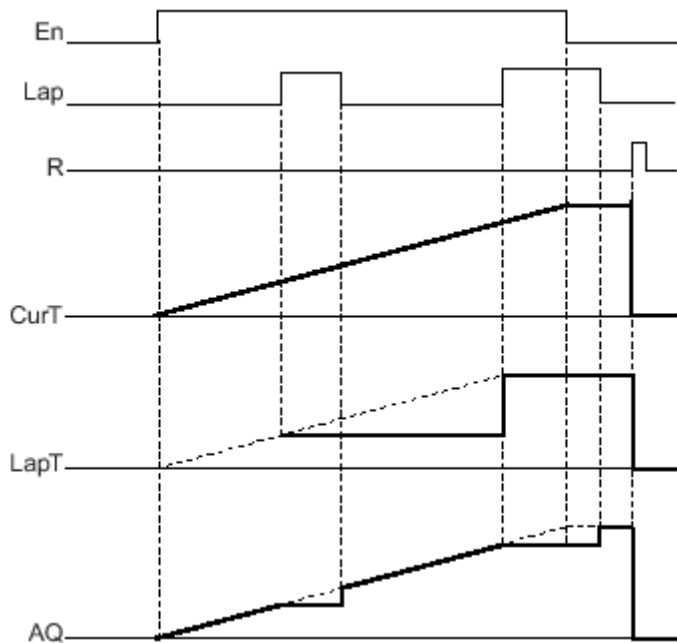
可依需求于下列的对话框中配置模拟量输出的时基：



可依需求将秒表的读数时基设定成小时、分钟、秒或10毫秒(百分之一秒)；最小的时基单位为10毫秒(百分

之一秒)。

时序图



功能说明

当EN=1时，秒表开始计时。

当EN=0时，秒表停止计时。

当EN=1及Lap=0时，AQ输出当前的秒表读数。

当EN=1及Lap=1时，AQ输出秒表暂停时间。

当EN=0及Lap=1时，AQ输出秒表暂停时间。

当EN=0及Lap=0时，AQ输出秒表的上次读数。

当R=1时，会将秒表当前读数及暂停时间皆清零。

加/减计数器



依据参数设置，输入脉冲将会增加或减少一个内部值，当达到配置的阈值时，置位或复位输出。计数的方向可藉由输入Dir的信号来变更。

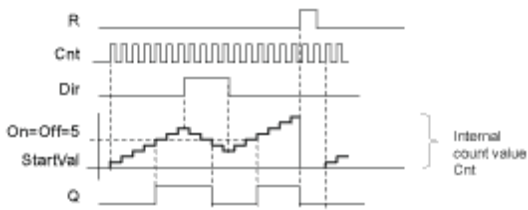
连接	描述
输入 R	可以藉由输入R(复位)的信号将输出及内部计数器值复位为起始值。
输入 Cnt	该功能在输入Cnt由0变成1时计数，由1变成0时停止计数。 可使用I0~I3来执行高频计数，高频输入连接到‘增/减计数器’功能块时，可支持高达15KHz频率。 使用其他输入或电路元素进行低频计数 (通常为 5 Hz).
输入 Dir	输入Dir(方向)将决定计数的方向： Dir = 0: 增 Dir = 1: 减
参数	On: 开启阈值 阈值范围: 0...999999 Off: 关闭阈值 阈值范围: 0...999999 起始值: 增或减计数的起始数值 可保持性: ‘勾选’ 该字段则表示该状态在存储器中可以保持。
输出 Q	依据Cnt的实际值和置位的阈值置位和复位输出Q。

* 1/2系列控制器的起始值(StartVal)将始终为0。

参数 On 及 Off

5/6系列控制器，参数On以及Off的时间可以由其他已编程功能的实际值来提供。

时序图



功能说明

该功能在输入Cnt的每个正跳沿上，以一个计数增加(Dir=0)或减少(Dir=1)内部计数器)

可以藉由复位输入R的信号将内部计数器值复位为初始值；当R=1，输出Q=0，且不会对输入Cnt的脉冲进行计数

依据Cnt的实际值和置位的阈值，置位和复位输出Q。

计算规则

当阈值(On) \geq 阈值(Off)时，则：

假如Cnt \geq On，则输出Q=1

假如Cnt<Off，则输出Q=0

当阈值(On)<阈值(Off)时，则：

假如On \leq Cnt<Off，则输出Q=1

注意

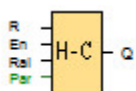
该功能在每个周期内会对计数器的限值轮询一次。

所以当高速输入I0~I3的脉冲比扫描周期快时，则直到超出指定的限值时SFB功能块才能切换。

例如：每个周期最多可以计数100个脉冲，当前已计数900个脉冲，开设定为950，关设定为1000；当值达到1000后，输出会在下一个周期置位。

如果Off为980时，则不会置位输出。

运行小时计数器



藉由监视输入处的信号触发配置时间，当超过该时间时置位输出。

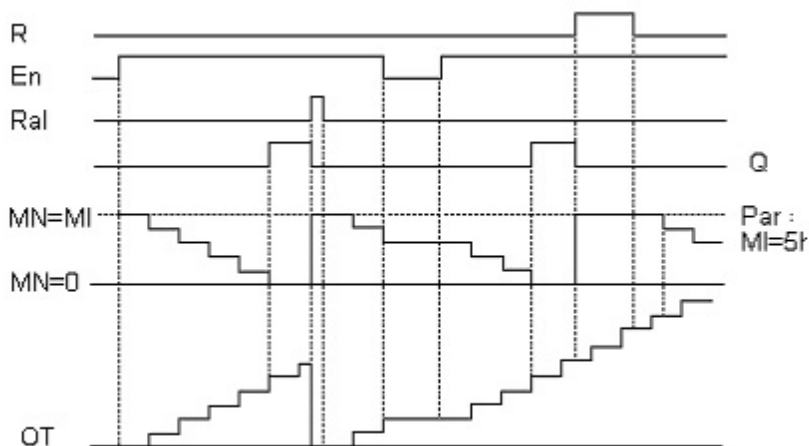
连接	Description
输入 R	输入R的正跳沿(0变成1)复位输出Q，且将剩余时间(MIN)设置为计数器的配置值MI。
输入 En	输入En为监控输入。
输入 Ral	当输入Ral(完全复位)出现正跳沿时，将复位运行小时计数器(OT)输出，并将剩余时间值(MN)置位为配置的维护间隔(MI)： <ul style="list-style-type: none"> • 输出 Q = 0, • 测量的运行时间 OT = 0 • 维护间隔的剩余时间 MN = MI
参数	MI: 以小时及分钟为单位指定的维护间隔，范围为0000~9999h, 0-59m OT: 累积总运行时间，以小时及分钟为单位指定偏移初始时间，范围为00000~99999h, 0~59m Q --> 0: <ul style="list-style-type: none"> • 当选择' R' : 如果MN=0，则Q=1 如果R=1或Ral=1，则Q=0 • 当选择' R+En' : 如果MN=0，则Q=1 如果R=1或Ral=1或En=0，则Q=0
输出 Q	当剩余时间MN=0时置位输出，且输出会在下列情况下复位： <ul style="list-style-type: none"> • 当 "Q --> 0 : R+En"时, 如果R=1及Ral=1或En=0 • 当 "Q --> 0 : R" , 如果 R = 1 及 Ral = 1.

* 1/2系列控制器MI 及 OT 不支持分钟的时基单位

参数 MI

5/6系列控制器，维护间隔MI参数可以由其他已编程功能的实际值来提供。

时序图



MI = 配置的时间间隔

MN = 剩余时间

OT = 自Ral处最后一个高电平信号后所经过的总时间

上述值始终可保持。

功能说明

运行小时计数器监视输入En，只要输入状态为1则软件会计算所经过的时间及剩余时间MN；当设置为配置模式时，软件即会显示这些时间。当剩余时间为0时，输出置位为1。

可藉由输入R的信号将输出Q及剩余时间计数器复位为指定的MI值，且运行小时计数器OT不受影响。

可藉由输入Ral的信号将输出Q及剩余时间计数器复位为指定的MI值，运行小时计数器OT复位为0。

依据Q参数的配置，可藉由输入R或Ral(Q->R)的复位信号复位输出，或当复位信号为1或En信号为0(Q->R+En)时复位输出。

OT的限值

当藉由输入R的信号复位运行小时计数器时，OT运行时间值会保持。当Ra1的信号由0变成1时，运行小时计数器ot将会复位为0。只要EN=1，不论复位输入R的状态为何，运行小时计数器OT会继续计数，OT的计数器上限值为99999 h 59 m，当运行小时计数器达到该值时会停止。

在编程模式下，可依需求设定OT初始置，只要不是设置为0，运行小时计数器将会自动计算。

注：MN会基于MI及OT的值来自动计算。

范例: 如果MI = 100及 OT = 130，则MN = 70

频率阈值触发器



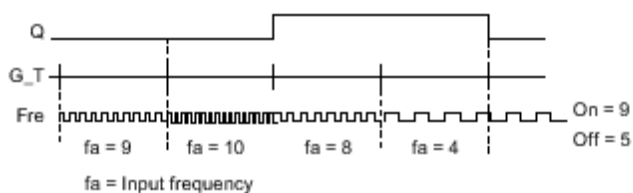
输出将依据两种可配置的频率打开及关闭。

连接	描述
输入 Fre	<p>该功能会对输入Fre从0变成1的跳数进行计数，不会对由1变成0计数。</p> <ul style="list-style-type: none"> 使用I0~I3来进行高频计数，快速输入直接连接到阈值触发器功能块时，可达15KHz频率。 使用I0~I3以外的输入或电路元素进行低频计数 (通常为 5 Hz)。
参数	<p>On : 接通阈值 阈值范围: 0000...9999</p> <p>Off : 断开阈值 阈值范围: 0000...9999</p> <p>G_T : 测量输入脉冲所经过的时间间隔或门时间，值范围: 00:00 s...99:99 s</p>
输出 Q	依据阈值置位或复位Q。

参数 G_T

5/6系列控制器，门时间G_T可以由其他已编程功能的实际值来提供。

时序图



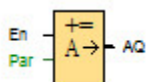
功能说明

触发器可量测输入Fre的信号，且在可配置周期G_T内获取脉冲，依据设置的阈值置位或复位Q。

计算规则

- 如果阈值(On) \geq 阈值 (Off)，则：
如果 $f_a > \text{On}$ ，则 $Q=1$
如果 $f_a \leq \text{Off}$ ，则 $Q=0$
- 如果阈值(On) $<$ 阈值 (Off)，则：
如果 $\text{On} \leq f_a < \text{Off}$ ，则 $Q=1$

模拟量算术指令



模拟量算术指令功能块可以计算由用户定义的运算数及运算符构成的方程式AQ值。

连接	描述
输入 En	如果输入En处产生一上升沿，则算术指令功能块则会启动。
参数	<p>V1: 值1：第一个操作数 V2: 值2：第二个操作数 V3: 值3：第三个操作数 V4: 值4：第四个操作数</p> <p>运算符1:第一个运算符 运算符2:第二个运算符 运算符3:第三个运算符</p> <p>优先级1: 第一项操作的优先级 优先级2: 第二项操作的优先级 优先级3: 第三项操作的优先级</p> <p>p: 小数位置 允许的设定值: 0, 1, 2, 3, 4, 5</p>
输出 AQ	输出AQ为运算数值和运算符构成的方程式结果。

参数 V1, V2, V3, 及 V4

V1~V4的值可由其他已编程功能的实际值来提供。

参数 p (小数位数)

参数p适用于消息文本中显示V1、V2、V3、V4及AQ。

功能说明

算术指令功能会将四个操作数及三个运算符组合成一个方程式，运算符为下列四个标准运算符中的任一

个：+，-，*及/。针对每个运算符，必须设置高("H")，中("M")，或低("L")中的一个唯一优先级。程序将先执行高优先级的运算，然后再执行中优先级的运算，最后为低优先级；每项运算只能对应一个优先级，运算数值可参考其它为提供该值而预定义的功能，算术指令功能将结果化为最近的整数值。

操作数值的个数会固定为4，运算符的个数将固定为3。如需使用较少的操作数，需使用"+ 0"或"* 1"等构造结构补充剩余的参数。

可依需求配置启用参数"En" =0时该功能的行为，该功能块会保留其最后的值，或置位为0。

范例

下表为简易的示例，说明了算术指令功能块参数及产生的方程式及输出值：

V1	运算符1 (优先级 1)	V2	运算符2 (优先级2)	V3	运算符3 (优先级3)	V4
12	+ (M)	6	/ (H)	3	- (L)	1

方程式: $(12 + (6 / 3)) - 1$

结果: 13

V1	运算符1 (优先级1)	V2	运算符2 (优先级2)	V3	运算符3 (优先级3)	V4
2	+ (L)	3	* (M)	1	+ (H)	4

方程式: $2 + (3 * (1 + 4))$

结果: 17

V1	运算符1 (优先级1)	V2	运算符2 (优先级2)	V3	运算符3 (优先级3)	V4
100	- (H)	25	/ (L)	2	+ (M)	1

方程式: $(100 - 25) / (2 + 1)$

结果: 25

模拟量比较器



依据 $Ax - Ay$ 的差以及两个可以配置的阈值来置位及复位输出。

连接	描述
输入 Ax, Ay	输入 Ax 和 Ay 为两个模拟量信号。
参数	A: 增益 值范围： ± 10.00 B: 零点偏移 值范围： $\pm 10,000$ On: 接通阈值 值范围： $\pm 20,000$ Off: 断开阈值 值范围： $\pm 20,000$ p: 小数位数 值范围：0, 1, 2, 3
输出 Q	依据设置的阈值置位或复位输出 Q。

参数 On 及 Off

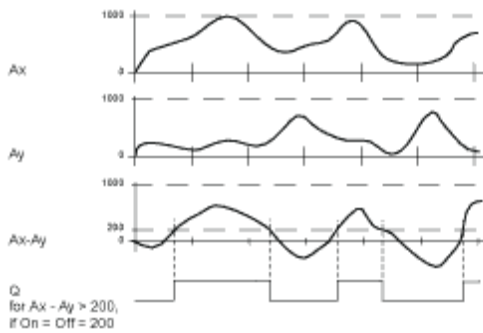
5/6 系列控制器，参数 On 和 Off 可以由其他已编程功能的实际值来提供。

参数 p (小数位数)

参数 p 仅适用于消息文本中显示的 Ax 、 Ay 、Delta、On 及 Off 值。

参数 p 不适用于 On 及 Off 值的比较。(比较功能会忽略小数点不计)

时序图



功能说明

该功能可读取模拟量Ax信号的值。

该值乘以参数A (增益)的值，参数B (偏移)与所得的积相加，因此：

$$(Ax * \text{增益}) + \text{偏移} = \text{实际值Ax}$$

$$(Ay * \text{增益}) + \text{偏移} = \text{实际值Ay}$$

依实际值Ax - Ay的差和设置的阈值置位或复位输出Q。

运算规则

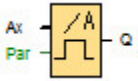
- 如果阈值On >= 阈值 Off，则：
 - 如果 (实际值 Ax -实际值Ay) > On，则输出Q=1
 - 如果 (实际值Ax -实际值Ay) <= Off，则输出Q=0
- 如果阈值 On < 阈值 Off，
 - 如果On <= (实际值Ax -实际值Ay) < Off，则输出Q=1

降低模拟量比较器的输入灵敏度

可依需求藉由‘接通延迟’和‘断开延迟’SFB功能块，选择性地延迟模拟量比较器的输出。这样可确定仅当输入触发长度Trg(代表模拟量比较器的输出)超过定义的接通延迟时间时置位输出Q。

您可依上述方法设置虚拟滞后，它可以使输入对较小更改的灵敏度变小。

模拟量阈值触发器



依据两种可配置的阈值(滞后)置位或复位输出。

连接	描述
输入 Ax	输入 Ax为模拟量信号。
参数	A: 增益 值范围：± 10.00 B: 零点偏移 值范围：± 10,000 On: 接通阈值 值范围：± 20,000 Off: 断开阈值 值范围：± 20,000 p: 小数位数 值范围：0, 1, 2, 3
输出 Q	依据设置的阈值置位或复位输出Q。

参数On 及 Off

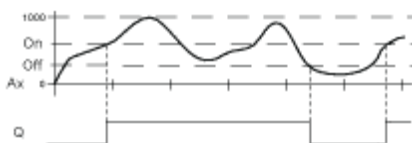
5/6系列控制器，参数On和Off可以由其他已编程功能的实际值来提供。

参数 p (小数位数)

参数 p仅适用于消息文本中显示的On、Off及Ax 值。

参数 p 不适用于On及Off值的比较。(比较功能会忽略小数点不计)

时序图



功能说明

该功能可读取模拟量Ax信号的值。

该值乘以参数A (增益)的值，参数B (偏移)与所得的积相加，因此：

$$(Ax * \text{增益}) + \text{偏移} = \text{实际值} Ax$$

依此设置的阈值置位或复位输出Q。

运算规则

- 如果阈值(On) \geq 阈值 (Off)，那么：
 - 如果实际值 $Ax > On$ ，则输出Q=1
 - 如果实际值 $Ax \leq Off$ ，则输出Q=0
- 如果阈值 (On) $<$ 阈值 (Off)，则：
 - 如果 $On \leq \text{实际值} Ax < Off$ ，输出Q=1

配置时要注意的特性

B1[Analog threshold trigger]

Parameter	Comment
Block name	<input type="text"/>
Sensor	0 ... 10 V
Measurement Range	
Minimum	<input type="text" value="0"/>
Maximum	<input type="text" value="1000"/>
Parameter	
Gain	<input type="text" value="1.00"/>
Offset	<input type="text" value="0"/>
Threshold	
On	<input type="text" value="0"/> Reference
Off	<input type="text" value="0"/> Reference
Decimals in the Message text	<input type="text" value="0"/> +12345
<input type="checkbox"/> Protection Active	

OK Cancel Help

注意：小数点的设置须与最小值到最大值的范围一致。

模拟量放大器



该功能块将放大仿真量输入值并在模拟量输出返回该值。

连接	描述
输入 Ax	输入 Ax为模拟量信号。 使用模拟量输入AI，仿真量标志AM，以及带有模拟量输出的功能块编号 AI: 0 - 10 V将可依比例设置为0 - 1000 (内部值)。
参数	A: 增益 值范围： -10.00 to +10.00 B: 零点偏移 值范围： -10,000 to +10,000 p: 小数位数 允许的设置： 0, 1, 2, 3
输出 AQ	AQ值范围： -32768 到 +32767

参数 p (小数位数)

参数 p仅适用于消息文本中显示的Ax及Ay值。

参数 p 不适用于On及Off值的比较。(比较功能会忽略小数点不计)

功能说明

该功能可读取模拟量Ax信号的值。

该值乘以参数A (增益)的值，参数B (偏移)与所得的积相加，因此：

$$(Ax * \text{增益}) + \text{偏移} = \text{实际值Ax}$$

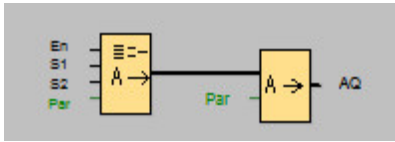
实际Ax在AQ输出。

模拟量输出

如将这个功能块连接到实际模拟量输出，请注意，模拟量输出只能处理0-1000内的值。要进行此操作请在该功能块的模拟量输出和实际模拟量输出之间连接一个附加放大器。藉由该放大器可以将该功能的输出范

围标准化为0-1000内的值。

范例：模拟量多路复用器之后的附加放大器。



模拟量监视器



这个功能块将模拟量输入的过程变量保存到储存器中，并在输出变量大于或小于该储存值与可配置偏移的和时置位输出。

连接	描述
输入 En	输入En的正跳沿(0变成1)将输入 Ax ("Aen")的模拟量值保存到储存器中，并启动对模拟量范围Aen±Delta的监视。
输入 Ax	可依需求来连接输入Ax并监视仿真量信号。 使用模拟量输入AI，模拟量标志AM，以及带有模拟量输出的功能块编号 AI : 0 - 10 V将可依比例设置为0 - 1000 (内部值)。
参数	A: 增益 值范围：±10.00 B: 零点偏移 值范围：±10000 加+(Delta:upper+)：开启/断开 阈值之上的差分值 值范围：0~ 20,000 (*) 减-(Delta:lower-)：开启/断开 阈值之下的差分值 值范围：0~ 20,000 (*) p: 小数位数 值范围：0, 1, 2, 3 可保持性：' 勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	依据存储的模拟量值和偏移置位/复位输出Q。

* 1/2系列控制器，Delta = upper+ = lower-.

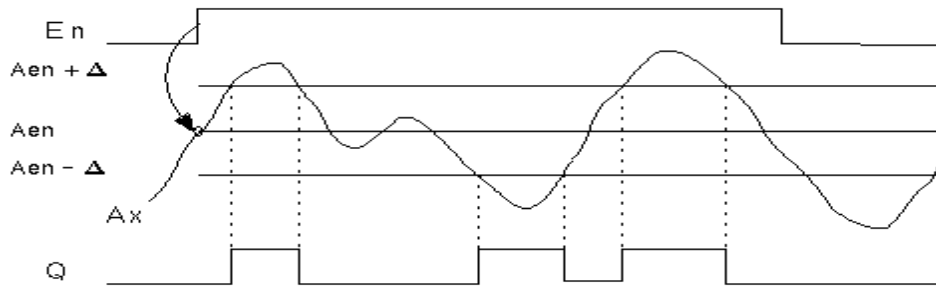
参数阈值 1 及阈值 2

5/6系列控制器，参数阈值 1 及阈值 2可以由其他已编程功能的实际值来提供。

参数 p (小数位数)

参数 p仅适用于消息文本中显示的Aen、Ax、upper+及lower-值。

时序图



功能说明

输入 E_n 由0变成1时，保存模拟量输入 A_x 的信号值，而这一过程变量称为” A_{en} ”。

模拟量实际值 A_x 及 A_{en} 都乘以参数A(增益)的值，然后将参数B(偏移)与所得的积相加如下：

当输入 E_n 由0变成1时， $(A_x * \text{增益}) + \text{偏移} = \text{实际值 } A_{en}$ ，或是

$$(A_x * \text{增益}) + \text{偏移} = \text{实际值 } A_x.$$

当输入 E_n 的信号为1，且输入 A_x 的实际值超过 $A_{en} + \text{upper}/A_{en} - \text{lower}$ 的范围时，置位输出Q。

当输入 A_x 的实际值在 $A_{en} + \text{upper}/A_{en} - \text{lower}$ 的范围内，或是当输入 E_n 的信号变为lo时，复位输出Q。

模拟量偏差值触发器



依据可配置的阈值和差分值置位和复位输出。

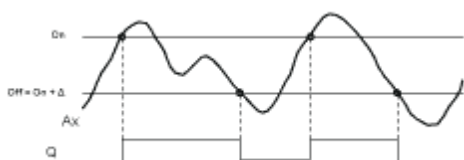
连接	描述
输入 Ax	输入 Ax 为模拟量信号。 使用模拟量输入 AI，模拟量标志 AM，以及带有模拟量输出的功能块编号 AI: 0 - 10 V 将可依比例设置为 0 - 1000 (内部值)。
参数	A: 增益 值范围: ± 10.00 B: 零点偏移 值范围: $\pm 10,000$ On: 接通/断开阈值 值范围: $\pm 20,000$ Δ : 用于计算 off 参数的差分值 值范围: $\pm 20,000$ p: 小数位数 值范围: 0, 1, 2, 3
输出 Q	依据阈值和差分值置位和复位输出 Q。

参数 p (小数位数)

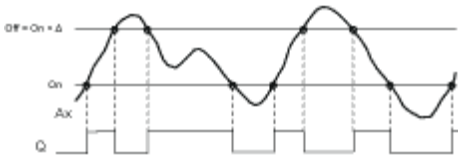
参数 p 仅适用于消息文本中显示的 On、Off、Ax 及 Ay 值。

参数 p 不适用于 On 及 Off 值的比较。(比较功能会忽略小数点不计)

时序图 A: 带有负的差值 Delta 的功能



时序图 B: 带有正的差值 Delta 的功能



功能说明

该功能可读取模拟量Ax信号的值。

该值乘以参数A (增益)的值，参数B (偏移)与所得的积相加，因此：

$$(Ax * \text{增益}) + \text{偏移} = \text{实际值 } Ax.$$

依据设置的阈值(On)和差值(Delta)置位或复位输出Q，该功能会自动计算Off参数： $\text{Off} = \text{On} + \text{Delta}$

其中Delta可以为正，也可以为负。

计算规则

- 如果设置了负的差分值Delta，且接通阈值 \geq 断开阈值，且：
 - 如果实际值 $Ax > \text{On}$ ，则输出 $Q=1$
 - 如果实际值 $Ax \leq \text{Off}$ ，则输出 $Q=0$
 - 可参考时序图A
- 如果设置了正差分值Delta，且接通阈值 $<$ 断开阈值，则：
 - 如果 $\text{On} \leq \text{实际值} Ax < \text{Off}$ ，则输出 $Q=1$ 。
 - 可参考时序图 B

模拟量多路复用器



当启用模拟量多路复用器后，会依据输入条件显示四个预先定义的模拟量值中的一个数值。

连接	描述
输入 En	输入En(启用)为1时，会依据S1和S2将参数化的模拟量值切换为输出AQ。 输入En为0时，会将0切换到输出AQ。
输入S1 和 S2	S1 及 S2 (选择器) 用于选择要发布的模拟量值 S1 = 0 且 S2 = 0: 发布值V1 S1 = 0 且 S2 = 1: 发布值 V2 S1 = 1 且 S2 = 0: 发布值 V3 S1 = 1 且 S2 = 1: 发布值 V4
参数	V1...V4：将发布的模拟量值(数值) 值范围： -32768 to +32767 p: 小数字数 可能的设定值： 0, 1, 2, 3
输出 AQ	模拟量输出AQ 值范围： -32768 to +32767

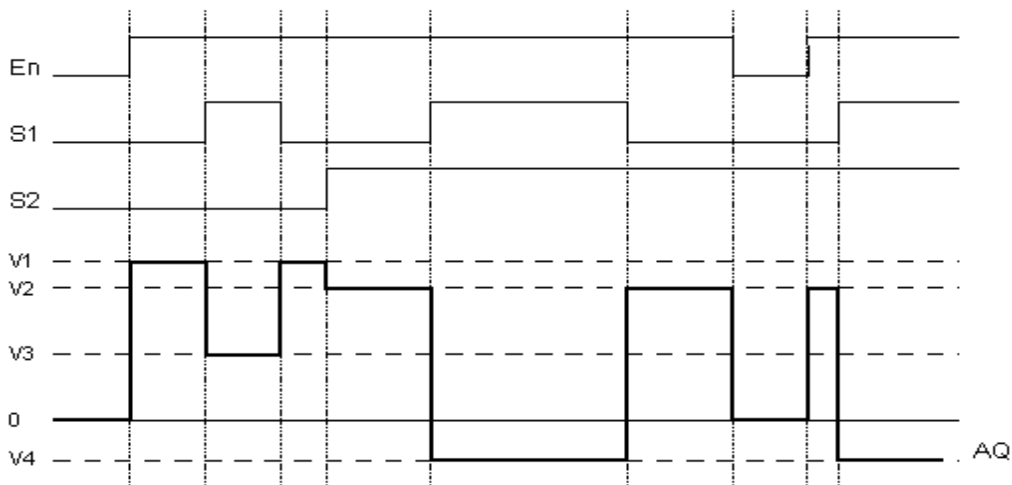
参数 V1...V4

5/6系列控制器，参数V1...V4可以由其他已编程功能的实际值来提供。

参数 p (小数位数)

参数p只适用于在消息文本显示AQ、V1、V2、V3及V4的值。

时序图



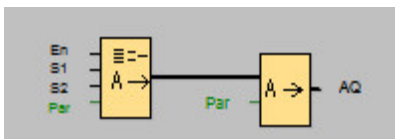
功能说明

如置位了输入En，则该功能会依据输入S1及S2在输出AQ发布4个可能的模拟量值(V1~V4)之一。如果尚未置位输入En，则该功能将在输出AQ输出模拟量值0。

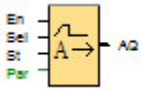
模拟量输出

如将该功能块与一个实际模拟量输出连接，请注意：模拟量输出仅能处理0~1000间的数值。您可将该功能的仿真量输出和实际模拟量输出间另外连接一个放大器功能块；藉由该放大器，可以将该功能的输出范围标准化为0~1000范围内的值。

范例：模拟量多路复用器之后的附加放大器。



模拟量斜坡函数发生器



模拟量斜坡函数发生器允许输出以指定速率由当前级别变化成指定级别。

连接	描述
输入 En	<p>输入En(启用)由0变成1时，将会启动/停止级别(偏移 "B" + StSp) 应用到输出，维持100ms，然后启动斜坡操作，直到变化到选定级别。</p> <p>输入En由1变成0时，会将当前级别立即设置为偏移"B"，这会使输出AQ=0。</p>
输入 Sel	<p>SeI = 0: 选择步进 1 (级别 1)。</p> <p>SeI = 1: 选择步进2 (级别 2)。</p> <p>Sel状态的更改会引起目前级别开始以指定速率变化到选定级别。</p>
输入 St	<p>输入St(减速停止)的状态由0变成1时，会引起目前级别以恒定速率减速，直到启动/停止级别(偏移 "B" + StSp)。启动/停止级别维持100ms，然后将目前级别设置为偏移"B"，这会使输出AQ=0。</p>
参数	<p>Level1 和 Level2: 每个级别的值范围： -10,000 到 +20,000</p> <p>MaxL: 不得超过的最大值。 值范围： -10,000 to +20,000</p> <p>StSp: 启动/停止偏移：与偏移"B"相加以创建启动/停止级别的值。如果启动/停止偏移为0，则启动/停止级别为偏移"B"。 值范围： 0 to +20,000</p> <p>Rate: 达到级别1、级别2或偏移使用的速度，将发布每秒步进数。 值范围： 1 to 10,000</p> <p>A: 增益 值范围： 0 to 10,00</p> <p>B: 偏移 值范围： ±10.000</p> <p>p: 小数字数 可能的设定值： 0, 1, 2, 3</p>
输出 AQ	<p>使用下列公式测量输出AQ：</p> $(\text{目前级别} - \text{偏移 "B"}) / \text{增益 "A"}$ <p>注意：当在参数模式或讯息模式下显示输出AQ时，AQ将显示为未测量的值(工程单位：目前级别)。</p> <p>AQ值范围： 0...+32767</p>

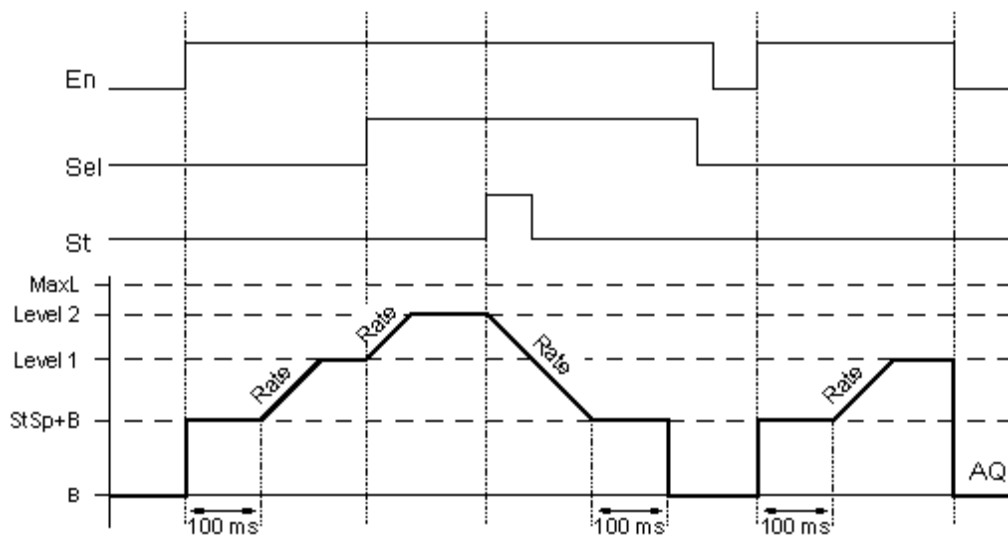
参数 Level1 及 Level2

级别参数 Level1 及 Level2 可由其他已编程功能的实际值提供。

参数 p (小数位数)

参数p只适用于在消息文本中显示AQ、level 1、level 2、MaxL、StSp及 Rate的值。

AQ时序图



功能说明

如果置位了输入En，则该功能会设置值StSp + 偏移 "B"，维持100ms。

然后依据Sel的连接，该功能以'速率'中设置的加速度从级别StSp + 偏移 "B"运行到级别1或级别2。

如果置位了输入St，则该功能以'速率'中设置的加速度运行到级别StSp + B。然后该功能在StSp + 偏移 "B"级别维持100ms。当100ms后，将级别设置为偏移"B"，输出AQ。测量的值(输出AQ)=0。

如果置位了输入St，该功能只能在复位了输入St和En之后才能重新启动。

如果输入Sel发生更改，该功能将依据Sel的连接，以指定速率从目前目标级别运行到新的目标级别。

如果复位了输入En，则该功能会立即将目前级别设置为偏移 "B"。

目前级别每100ms更新一次，请注意输出AQ和目前级别之间的关系：

输出 $AQ = (\text{目前级别} - \text{偏移量 "B"}) / \text{增益 "A"}$

PI 比例积分控制器



PI比例积分控制器是比例作用控制器及积分作用控制器。可依需求单独使用比例作用及积分作用，也可将二者组合起来使用。

连接	描述
输入 A/M	设置控制器的模式 1: 自动模式 0: 手动模式
输入 R	使用输入R复位输出AQ，只要置位了该输入，输入A/M就会被禁用，输出AQ置位为0。
输入 PV	模拟量值：过程值，会影响输出
参数	传感器：正在使用的传感器类型 Min.: PV的最小值 值范围：-10,000 到 +20,000 Max.: PV的最大值 值范围：-10,000 到 +20,000 A: 增益 值范围：±10.00 B: 偏移 值范围：± 10,000 SP: 设定值分配 值范围：-10,000 to +20,000 Mq: 手动模式下AQ的值 值范围：0 to 1,000 参数集: KC、TI 和 Dir的与应用程序相关的默认值(如下) KC: 增益 值范围：00.00 到 99.99 TI: 积分时间 值范围：00:01 分到 99:59 分 Dir: 控制器的作用方向 值范围：+ 或 - p: 小数字数 可能的设定值：0, 1, 2, 3
输出 AQ	模拟量输出(调节变量) AQ值范围：0 到 1,000

参数 SP 及 Mq

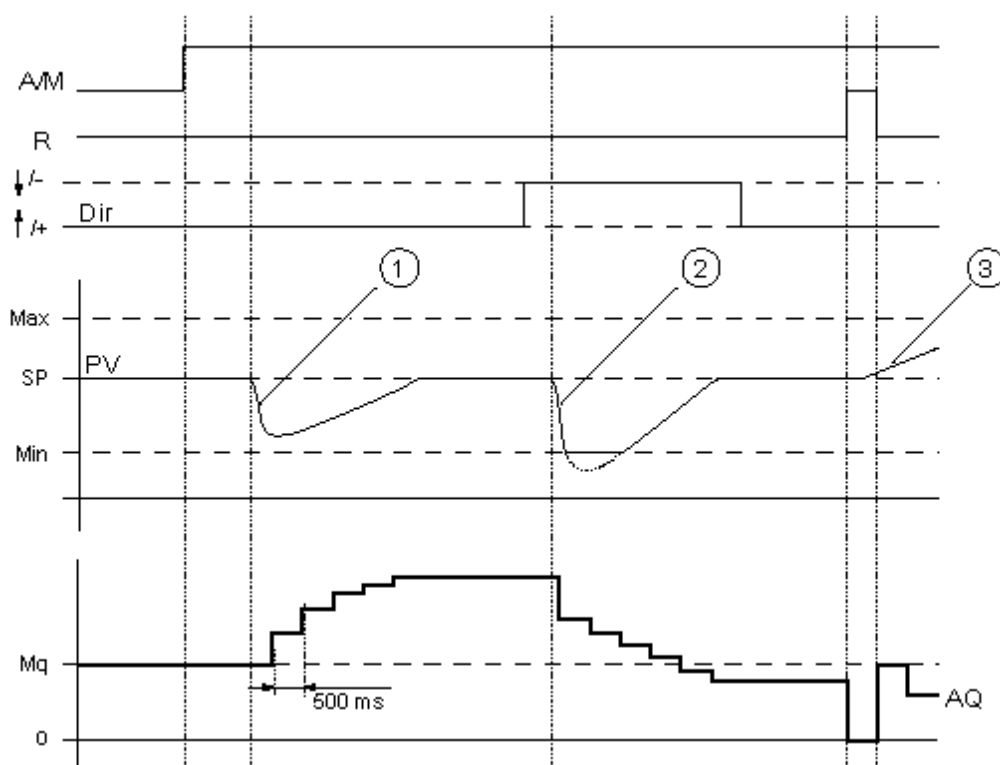
设定数值 SP 和数值 Mq 可由其他已编程功能的实际值提供。

参数 P (小数位数)

参数p只适用于在消息文本显示PV、SP、Min. 及 Max.的值。

时序图

AQ变化的性质、方式和速度取决于参数KC及TI。图中的AQ曲线为一很好的说明范例。控制作用为连续性的，因此该图只描述了其中一部分。



1	干扰会导致PV下降，当Dir的方向为向上时，AQ会增加，直到PV再次与SP一致。
2	干扰会导致PV下降，当Dir的方向为向上时，AQ会减少，直到PV再次与SP一致。 Dir会依据控制回路的基本行为进行调整。该功能执行期间，不能更改方向(dir)，在此更改Dir是为了进行说明。
3	当藉由输入R将AQ置位为0时，PV将更改。这是因为PV增加，由于Dir=向上而导致AQ下降。

功能说明

如果输入A/M置位为0，则该功能会藉由您使用参数Mq设置的值发布输出AQ。

如果输入A/M置位为1，则启动自动模式。因为采用Mq值作为积分和，所以该控制器功能将依据公式开始计算，公式中所用的是更新值PV。

$$\text{更新值 PV} = (\text{PV} * \text{增益}) + \text{偏移}$$

如果更新值PV = SP，则该特殊功能不会更改AQ的值。

Dir = 向上/+ (时序图编号1及3)

- 如果更新值 PV > SP，则该功能会减小AQ的值。
- 如果更新值 PV < SP，则该功能会增加AQ的值。

Dir = 向下/- (时序图编号2)

- 如果更新值 PV > SP，则该功能会增加AQ的值。
- 如果更新值 PV < SP，则该功能会减小AQ的值。

受到干扰时，AQ会继续增加/减少，直到更新值PV再次与SP一致。AQ变化的速度取决于参数KC 和 TI。

如果输入PV大于参数Max.，则更新值PV设置为Max.的值。如果PV小于参数Min.，则更新值PV设置为Min.的值。

如果输入R置位为1，则复位AQ输出。只要置位了R，输入A/M就会被禁用。

采样时间

采样时间固定为 500 ms。

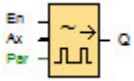
参数集：

为简化PI控制器的使用，KC、TI及Dir的参数已作下列应用的参数集形式给出：

参数集	应用范例	参数KC	参数 TI (s)	参数 Dir
温度 快	小空间的温度、制冷控制、小容量	0.5	30	+
温度 慢	大空间的加热、通风、温度、制冷控制、大容量	1.0	120	+

压力 1	快速压力变化、压缩机控制	3.0	5	+
压力 2	慢速压力变化、差分压力控制(流量控制器)	1.2	12	+
完整级别 1	无排水的水桶和/或蓄水池填充	1.0	99:59	+
完整级别 2	有排水的水桶和/或蓄水池填充	0.7	20	+

PWM 脉宽调制器



PWM脉宽调制器将模拟量输入值Ax调制为受脉冲影响的数字量输出信号，脉冲宽度与模拟量值Ax成比例。

连接	描述
输入 En	输入En的正边缘(0变成1)将启动PWM脉宽调制器功能块。
输入 Ax	要调制为受脉冲影响的数字量输出信号的模拟量信号。
参数	A: 增益 值范围： -10.00 to +10.00 B: 零点偏移 值范围： -10,000 to +10,000 PT: 数字量输出调制的周期性时间 Out :高速输出脚位 (当选择输出 Q0 至 Q3) p: 小数字数 允许的设定值0, 1, 2, 3
输出 Q	依据标准化的值Ax与模拟量值范围的比例针对每个时间周期的比例置位或复位Q (当选择输出 Q0至 Q3时，输出Q=0)。

参数 PT

周期时间PT可由其他已编程功能的实际值来提供。

参数 p (小数位数)

参数p只适用于在消息文本中显示Ax值。

功能说明

该功能可读取模拟量输入Ax信号的值。

该值乘以参数A (增益)的值，参数B (偏移)与所得的积相加，如下：

$$(Ax * \text{增益}) + \text{偏移} = \text{实际值 } Ax$$

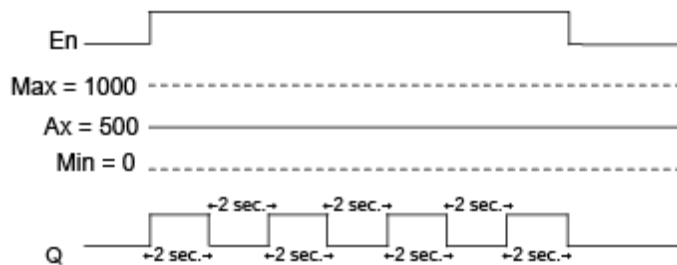
该功能块计算值Ax与该范围的比例，该块在PT(周期性时间)的相同时间内将数字量输出Q置位为高，在剩余时间周期内将Q置位为低。

时序图范例

下列范例显示了PWM指令如何依据模拟量输入值来调制数字量输出信号：

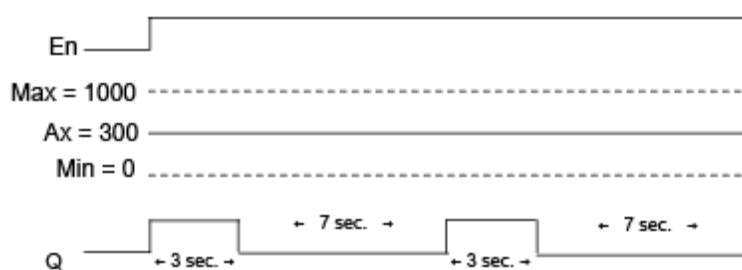
范例 1 模拟量输入数值：500 (范围 0...1000) 周期时间T: 4 秒钟

PWM调制解调器功能的数字量输出为2秒高、2秒低、2秒高、2秒低，且只要参数En=高，就会依这种模式继续下去。



范例 2 模拟量输入数值：300 (范围 0...1000) 周期时间T: 10 秒钟

PWM调制解调器功能的数字量输出为3秒高、7秒低、3秒高、7秒低，且只要参数En=高，就会依这种模式继续下去。



计算规则

对于时间周期PT的 $Ax - \text{Min} / (\text{Max} - \text{Min})$ ，输出 $Q=1$

对于时间周期PT的 $PT - [(Ax - \text{Min}) / (\text{Max} - \text{Min})]$ ，输出 $Q=0$

注意：此次计算中Ax引用通过增益和偏移计算的Ax的实际值，Min及Max引用特定于范围的最小值和最大值。

模拟量滤波器

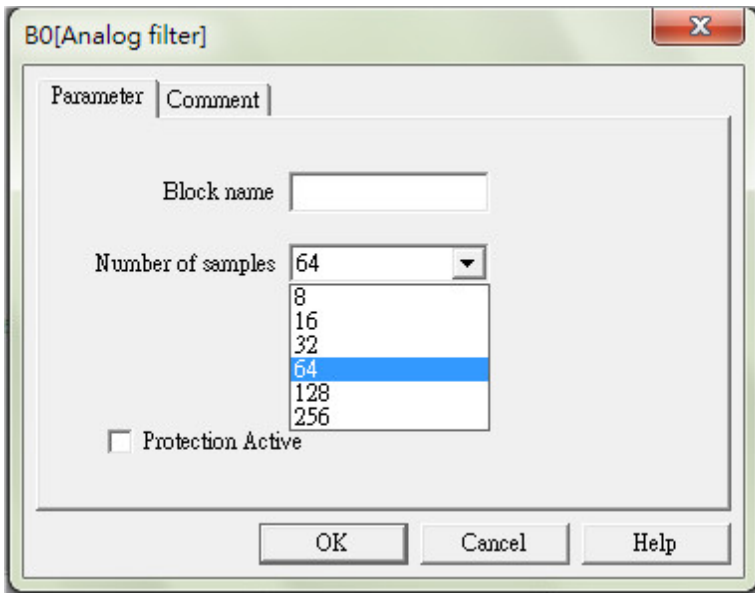


连接	描述
输入 Ax	输入 Ax 为下列仿真量信号中的一个： <ul style="list-style-type: none">• AI (*)• AM• AQ• 带有模拟量输出的功能块编号
参数	Sn (采样数)：决定在程序周期间所采样的模拟量数值的数量，而程序周期数由所设定的采样数确定。在每个程序周期内，软件都会采样一个仿真量数值，程序周期数与所设定的采样数相同。 可能的设置值： 8, 16, 32, 64, 128, 256
输出 AQ	AQ 基于目前采样数输出模拟量输入 Ax 的一个平均值，并依据模拟量输入和采样数置位或复位。

* AI：0~10V 对应 0~1000 (内部值)。

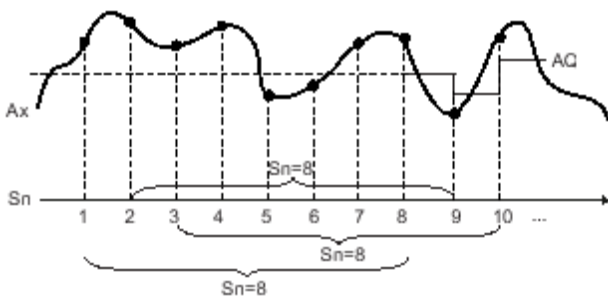
参数

可依需求将采样参数设置为下列数值：



当完成参数设置后，模拟量滤波器会计算出采样的平均值，并将该值分配给AQ。

时序图

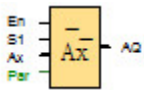


功能说明

该功能依据所设置的采样数，模拟量输入信号进行采样后，输出平均值。该功能块可以减少模拟量输入信号的错误。

注意：最多可使用8个模拟量滤波器功能块。

最大值/最小值



最大值/最小值功能块记录数值的最大值或最小值。

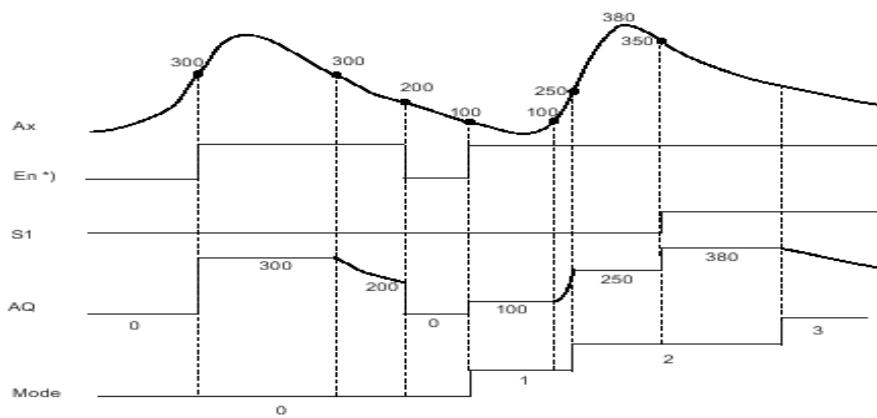
连接	描述
输入 En	输入En(启用)的功能块取决于参数'模式'的设置及复选框"当En=0时,复位最大值/最小值"的选择。
输入 S1	当设置Mode =2时,该输入被激活: 当输入S1产生上升沿(0变成1)时,输出AQ会输出最大值。 当输入S1产生下降沿(1变成0)时,输出AQ会输出最小值。
输入 Ax	输入 Ax为下列仿真量信号中的一个: <ul style="list-style-type: none"> • AI (*) • AM • AQ • 带有仿真量输出的功能块编号
参数	Mode: 可能的设置: 0, 1, 2, 3 Mode = 0: AQ = 最小值 Mode = 1: AQ = 最大值 Mode = 2 和 S1= 0 (低): AQ = 最小值 Mode = 2 和 S1= 1 (高): AQ = 最大值 Mode = 3 或是引用了一个功能块的数值: AQ = Ax
输出 AQ	AQ 依据输入输出最小值、最大值或者实际值,或者复位为0(如果在禁止该功能时,进行了这设置)

* AI : 0-10V对应0-1000(内部值)。

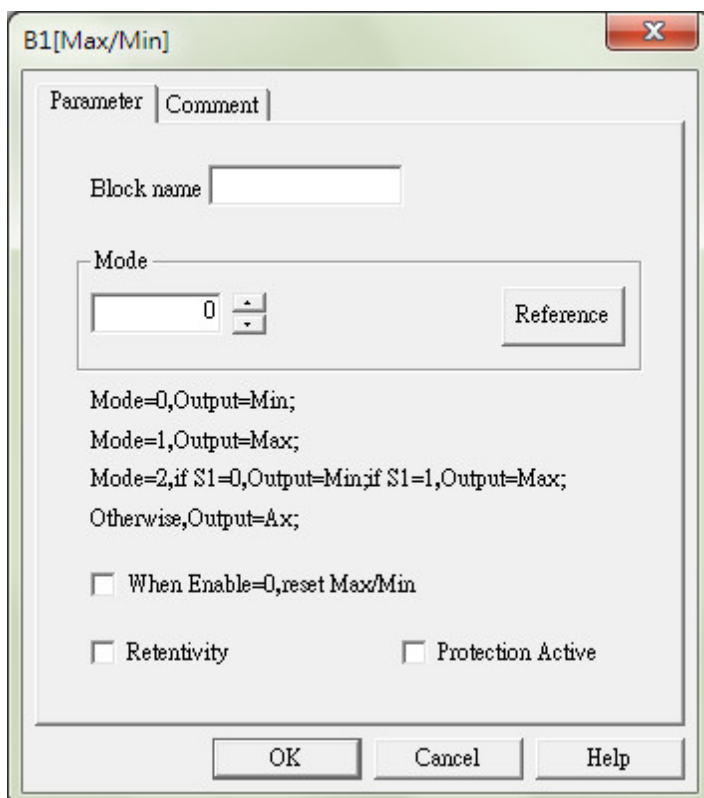
参数 Mode (模式)

可以依据另一个已编程功能的实际值来设置参数 Mode (模式) 的数值。

时序图



功能说明



*) 如选择了复选框“当En=0时，复位最大值/最小值”：

- En = 0: 该功能将AQ的数值设置为0。
- En = 1: 该功能依据Mode和S1的设置，在AQ输出一个数值。

*) 如没有选择复选框“当En=0时，复位最大值/最小值”：

- En = 0: 该功能将AQ的数值保持为目前值。
- En = 1: 该功能依据Mode和S1的设置，在AQ输出一个数值。

Mode = 0: 该功能将AQ的数值设置为最小值。

Mode = 1: 该功能将AQ的数值设置为最大值。

Mode = 2 且 S1 = 0: 该功能将AQ的数值设置为最小值。

Mode = 2 且 S1 = 1: 该功能将AQ的数值设置为最大值。

Mode = 3 或者引用了一个功能块的数值：该功能输出实际的模拟量输入数值。

平均值



平均值功能在配置的时间周期内对仿真量输入信号进行采样，并在AQ输出平均值。

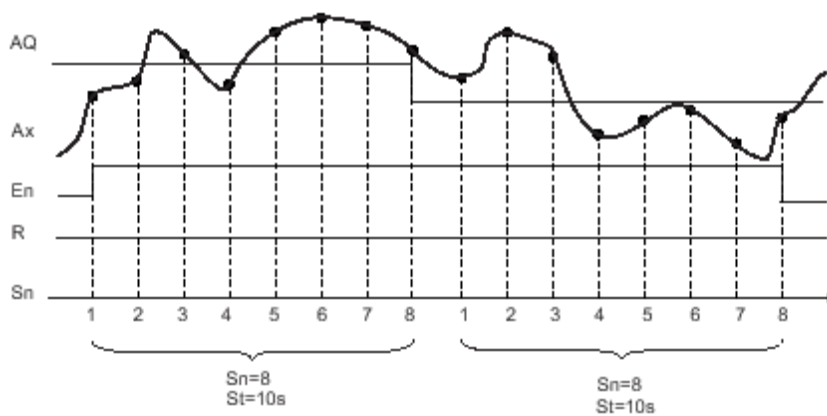
连接	描述
输入 En	如果输入En(启用)产生上升沿(0变成1)，则在配置的时间结束后，AQ输出输入Ax的平均值。 如果输入En产生下降沿(1变成0)，则输出最后一次计算的数值。
输入 R	如果在输入R(复位)产生上升沿(0变成1)，则输出AQ复位为0。
输入 Ax	输入 Ax 为下列模拟量信号中的一个： <ul style="list-style-type: none"> • AI (*) • AM • AQ • 带有模拟量输出的功能块编号
参数	<p>St (采样时间): 可设为秒、天、小时或分钟。 值范围: 如果 St = 秒: 1 到 59 如果 St = 天: 1 到 365 如果 St = 小时: 1 到 23 如果 St = 分: 1 到 59</p> <p>Sn (采样数): 值范围: 如果 St = 秒: 1 到 St*100 如果 St = 天: 1 到 32767 如果 St = 小时: 1 到 32767 如果 St = 分钟且 St ≤ 5 分钟: 1 到 St*6000 如果 St = 分钟且 St ≥ 6 分钟: 1 到 32767</p>
输出 AQ	AQ 在特定采样时间结束后，输出平均值。

* AI : 0 -10V对应0-1000(内部值)。

参数 St 和 Sn

参数 St 为采样时间，参数 Sn 为采样数。

时序图

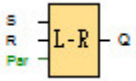


功能说明

当 $En = 1$ 时，平均值功能在所配置的时间周期内计算采样的平均值。在采样时间结束后，该功能将输出 AQ 的值设置为所计算的平均值。

当 $En = 0$ 时，停止计算， AQ 保持为上一次计算的数值。当 $R=0$ 时， AQ 复位为 0。

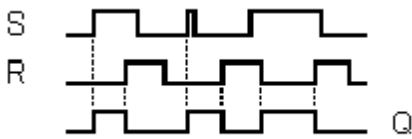
锁存继电器



输入S的信号置位输出Q，输入R的信号复位输出Q。

连接	描述
输入 S	通过输入S(置位)的信号置位输出Q。
输入 R	通过输入R(复位)的信号复位输出Q。如果置位了S及R，则输出Q将复位(复位优先于置位)。
参数	可保持性： '勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	Q通过输入S的信号进行置位，并在通过输入R的信号复位前仍保持置位状态。

时序图



功能说明

锁存继电器为简单的二进制存储器逻辑，输出值取决于输入状态及上一个输出状态。

逻辑表：

S	R	Q	备注
0	0	x	状态不变
0	1	0	复位
1	0	1	置位

当启用可保持性后，输出信号与发生电源故障前的信号状态将一致。

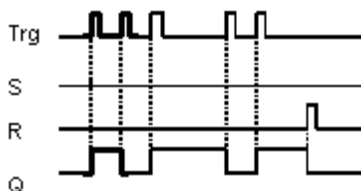
脉冲继电器



藉由短暂的单次触发输入可以置位和复位输出。

连接	描述
输入 Trg	可藉由输入Trg(触发器)的信号打开或关闭输出Q。
输入 S	单次触发输入S(置位)可将输出置位为逻辑值1。
输入 R	单次触发输入R(复位)可将输出置位为逻辑值0。
参数	选择: RS (输入R优先), 或SR (输入S优先) 可保持性: '勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	如果S和R都为0, 则输出Q通过Trg的信号打开, 且在下一个Trg脉冲再次复位。

时序图



功能说明

当输入Trg由0变成1时, 且S及R都为0时, 输出Q的状态将更改, 即输出将打开或关闭。

当S=1或R=1时, 输入Trg不影响功能块。

单次触发输入S将置位脉冲继电器, 即输出置位为逻辑值1。

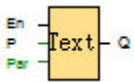
单次触发输入R将使脉冲继电器复位为其初始状态, 即输出置位为逻辑值0。

输入R优先于输入S(只要R=1, 输入S的信号就没有影响), 或是输入S优先于输入R(只要S=1, 输入R的信号就没有影响), 这取决于使用需求。

警告

如果 $\text{Trg}=0$ 且 $\text{Par}=\text{RS}$ ，则”脉冲继电器”与”锁存继电器”的功能相同。

信息文本显示器



■ 仅供1/2系列控制器

当控制器于Normal模式时，机载显示屏上显示消息文本或者其他功能块的参数。

连接	Description
输入 En	输入En(启用)由0变成1时，将触发消息文本的输出。
输入 P	P为消息文本的优先级。 0为优先级最低，15为优先级最高。 确认：消息文本的确认。
参数	消息文本：消息文本的输入 参数：其他已配置功能的参数或者实际数值 当前时间：显示持续更新的当天时间 当前日期：显示持续更新的当天日期 消息使能时间：显示信号由0变成1的发生时间 消息使能日期：显示信号由0变成1的日期转换
输出 Q	只要有消息文本排队，输出Q即保持置位状态。

功能说明

藉由在输入En的0变成1的信号转换，控制器在Normal模式下显示输出已配置过的消息文本(实际数值、文本、TOD、日期等)。

禁用确认(Ack = 关闭)：

藉由输入E的0变成1的信号转换，从而将消息文本隐藏起来。

启用确认(Ack = 打开)：

输入En复位为0后，消息文本将显示，直到通过按下”确定” 按键进行确认。只要输入En为高设置，消息文本就无法确认。

如果藉由En=1触发了多个消息文本功能，则将显示具有最高优先级(0代表最低优先级，15代表最高优先级)

的消息。这也说明，仅当某新消息文本的优先级高于先前启用的消息文本的优先级时，才显示该新消息文本。

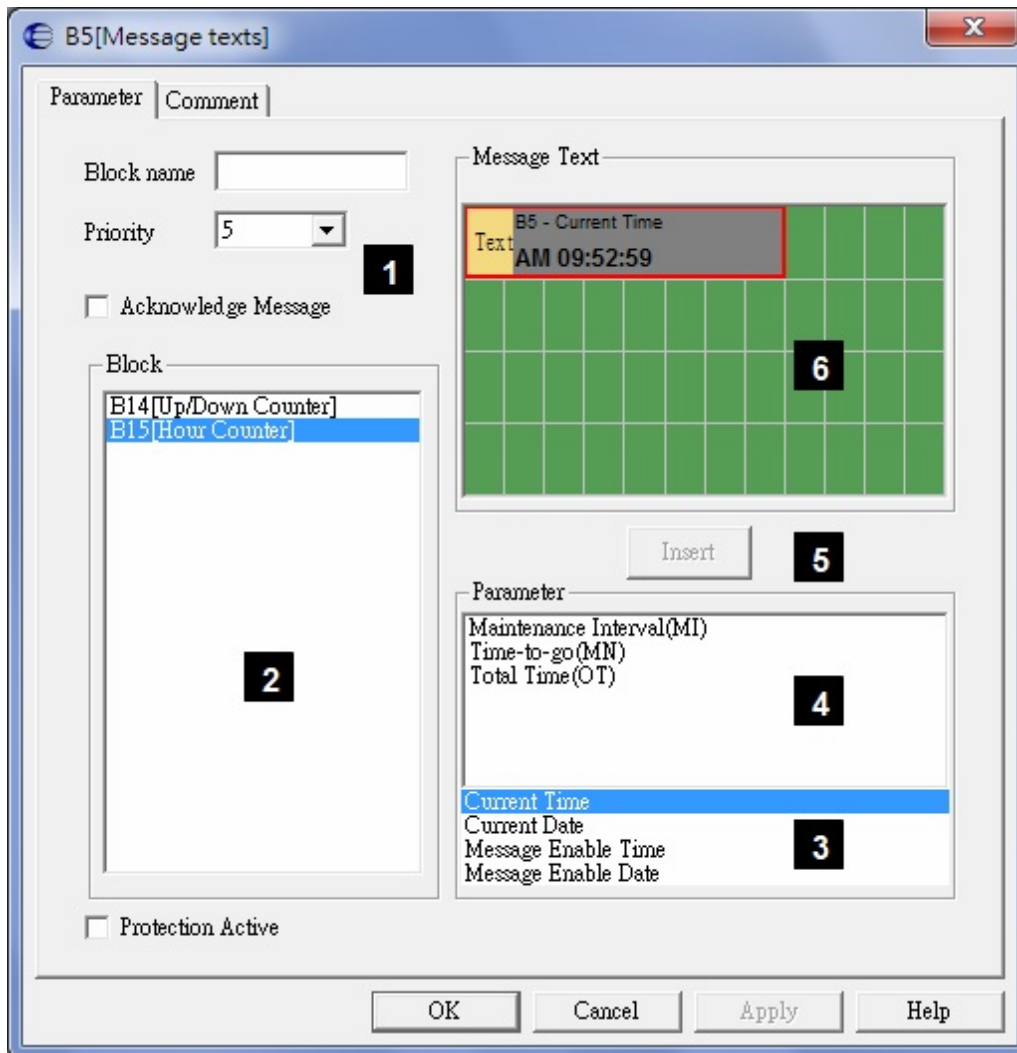
禁用或确认消息文本后，该功能会自动显示先前激活的消息文本中具有最高优先级的消息文本。

可以使用▲及▼按键在控制器Normal模式下的显示和消息文本之间切换。

限制

最多可使用16个消息文本功能。

配置时需注意的特性



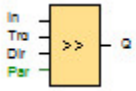
1	<p>"常规" 区域</p> <p>可以在此找到下列设置</p> <ul style="list-style-type: none"> ● 消息文本的优先级 ● 消息文本确认复选框
2	<p>"块"区域</p> <p>显示一列表，其中会列出所有电路程序块以及他们的参数。</p>

3	"基本参数" 区域 显示基本参数，如目前日期
4	"块参数" 区域 显示从"块"区域中所选中的一个块的参数，且可以在消息文本中输出此"块"区域
5	"插入" 按键 用于将一个从"块参数"或"常规参数"区域中选中的参数插入消息文本中的按钮
6	"消息" 区域 可在此区域对消息文本进行排列，此区域输入的信息会与控制器显示屏上显示的信息一致

消息文本排列步骤：

1. 从"块"区域中选择要输出其参数的块。
2. 将所需的参数从"块参数"拖放到"消息"区域中，且可以使用"插入"按钮来进行此项操作。
1. 在"消息"区域中，可依需求增加参数数据。

移位寄存器

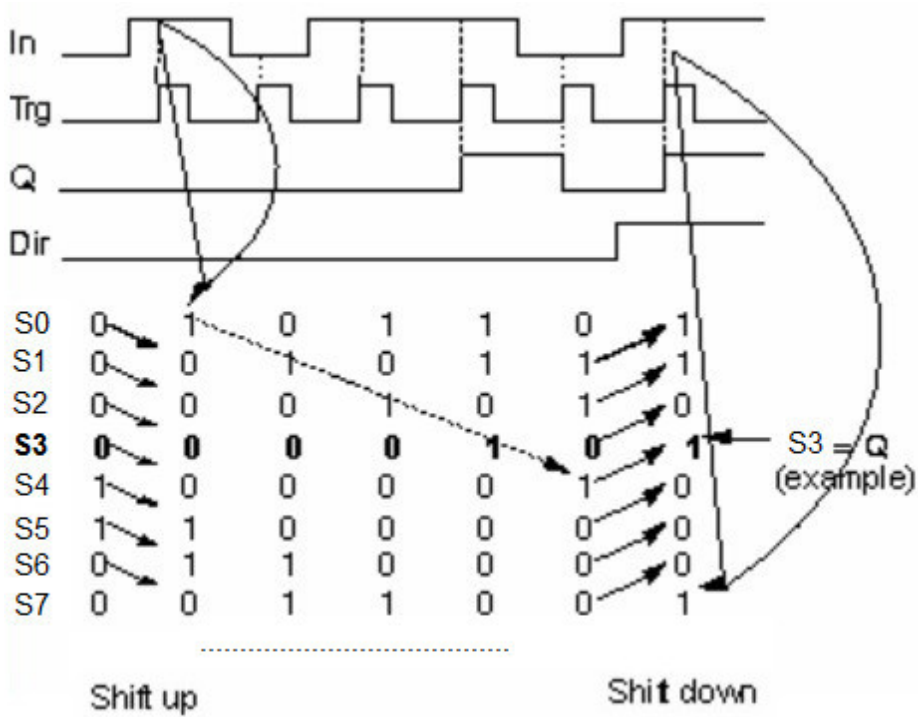


■ 仅供1/2系列控制器使用：

移位寄存器功能读取一个输入值，并移动数值位。输出值与配置的移位寄存器位一致。可在特定输入处更改移位方向，在一个电路程序中只能使用一个移位寄存器。

连接	描述
输入 In	该功能启动后将读取此输入值。
输入 Trg	可以藉由Trg(触发器)的正跳沿(0变成1)启动该功能块，与从1变成0无关。
输入 Dir	可以藉由Dir输入定义移位寄存器S0到S15的移位方向： Dir = 0: 向上移位 (S0 >> S15) Dir = 1: 向上移位 (S15 >> S0)
参数	用于确定输出Q的值的移位寄存器位。可能的设置：S0 ... S15 可保持性：' 勾选' 该字段则表示该状态在存储器中可以保持。
输入 Q	输出值与配置的移位寄存器位一致。

时序图



功能说明

该功能通过输入Trg(触发器)的正跳沿(0变成1)读取输入In的值。
依据设置的移位方向，将该值写入移位寄存器位S0或S15中：

- 向上移位：S0接受输入In的数值，S1的上一个值移位至S2，S2移位至S3，依此类推。
- 向下移位：S15接受输入In的数值，S15的上一个值移位至S14，S14移位至S13，依此类推。

Q输出配置的移位寄存器位的值。

如果未启用可保持性，则发生电源故障后，移位功能将在S0或S15处重新启动。

注意：

移位寄存器功能块仅能在电路程序中使用一次。

■ 仅供5/6系列控制器使用

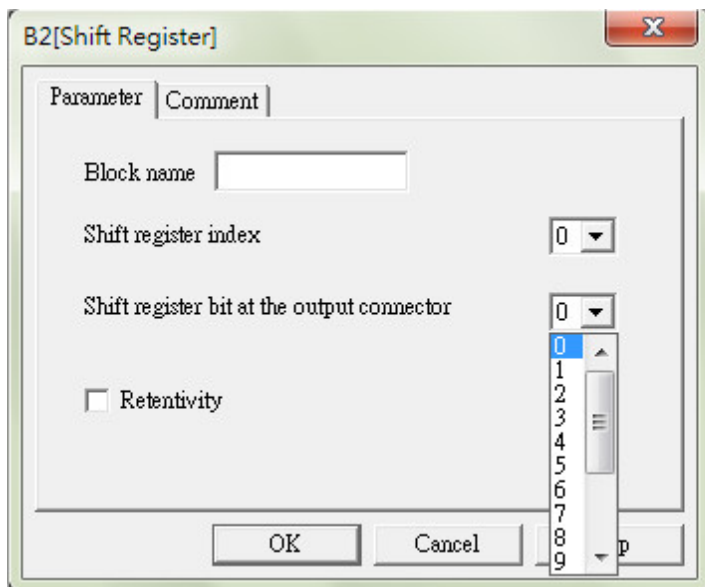
移位寄存器功能读取一个输入值，并移动数值位。输出值与配置的移位寄存器位一致。可在特定输入处更改移位方向。

在一个电路程序中最多可使用4个移位寄存器，其中每个寄存器带有16个位。

连接	描述
输入 In	该功能启动后将读取此输入值。
输入 Trg	可以藉由输入Trg(触发器)的正跳沿(0变成1)启动该功能块，与1变成0无关。
输入 Dir	可以藉由Dir输入定义移位寄存器Sx.0到Sx.15的移位方向： Dir = 0: 向上移位 ($Sx.0 \gg Sx.15$) Dir = 1: 向下移位 ($Sx.15 \gg Sx.0$) 注意: "x" 为移位寄存器的索引号。
参数	移位寄存器: 电路程序中移位寄存器的索引号。可能的设置: 0 到 3 用于确定输出Q的值的 移位寄存器位 。可能的设置: 0 到 15 可保持性: ' 勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	输出值与配置的移位寄存器位一致。

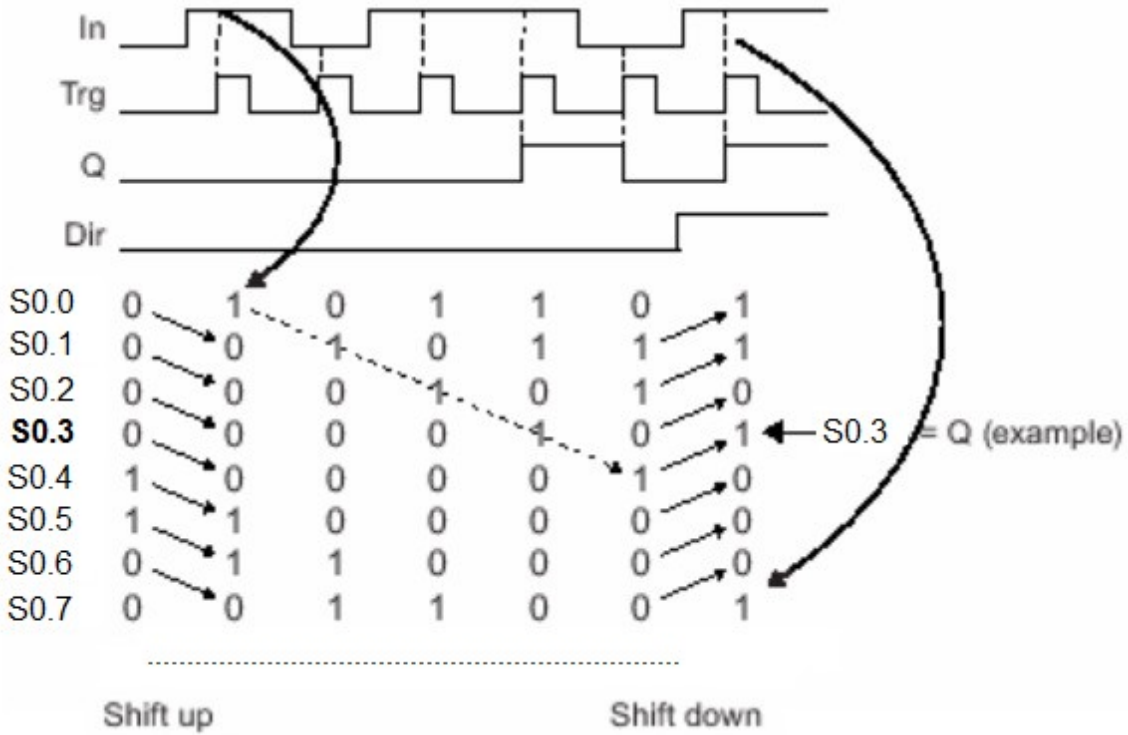
参数

5/6系列控制器提供了四个移位寄存器，每个移位寄存器带有16个位。移位寄存器索引号与电路程序中四个移位寄存器之一相对应。移位寄存器位以Sx.y编号，其中x为索引号，y为位编号。



时序图

如移位寄存器索引号为0，则移位寄存器位为 S0.0 到 S0.15。



功能说明

该功能藉由输入Trg(触发器)的正跳沿(0变成1)读取输入In的值。

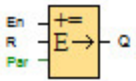
依据设置的移位方向，将该值写入移位寄存器位Sx.0到 Sx.15中：

- Dir = 0 (向上移位): Sx.0接收输入In的数值， Sx.0原来的数值移位到Sx.1， Sx.1原来的数值移位到 Sx.2， 依此类推，直到Sx.14原来的数值移位到 Sx.15。
- Dir = 1 (向下移位): Sx.15接收输入In的数值， Sx.15原来的数值移位到 Sx.14， Sx.14原来的数值移位到Sx.13， 依此类推，直到Sx.1原来的数值移位到Sx.0。
-

Q输出配置的移位寄存器位的值。

如果未启用可保持性，则发生电源故障后，移位功能将在Sx.0 或 Sx.15处重新启动。

算术指令错误检测



如错误发生于引用的算术指令功能块中，则算术指令错误探测功能块将置出一个输出。

连接	描述
输入 En	启用算术指令错误检测功能块。
输入 R	复位输出。
参数	引用的 FB 功能块：算术指令的块编号 要检测的错误：除数为零、溢出或者这两者之一 自动复位：故障情况清除后复位输出
输出 Q	如检测到的错误发生在引用的算术指令功能块的最后执行过程中，则将输出 Q 置位为高。

引用的FB参数

引用的FB参数的值引用已编程的算术指令功能块的块编号。

功能说明

如引用的算术指令功能块出现错误，则算术指令错误检测功能块将置位输出。用户可以编写功能以在发生除数为零错误、溢出错误或两者中任何一种错误发生时置位输出。

如选中“自动复位”复选框，则将在执行下一个功能块之前复位输出。否则输出将在使用R参数复位算术指令错误检测块之前保持其状态不变。

于任意扫描周期内，如引用的算术指令功能块在算术指令错误检测功能块之前执行，则将在同一扫描周期内检测错误。如引用的算术指令功能块在算术指令错误检测功能块之后执行，则会在下一个扫描周期内检测错误。

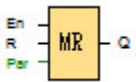
算术指令错误检测逻辑表

下表中，要检测的错误表示算术指令错误检测功能的参数，其中这些参数用于选择要检测的错误类型。零表示在执行结束时通过算术指令置位除数为零位：如出现错误为1，否则为0。OF表示通过算术指令置位溢出位：如出现错误为1，否则为0。除数为零或溢出表示对引用的算术指令的除数为零位和溢出位进行逻辑或运算。输出(Q)表示算术指令错误检测功能的输出。” x” 表示位可以为0或1且对输出无影响。

要检测的错误	零	OF	Output (Q)
除数为零	1	x	1
除数为零	0	x	0
溢出	x	1	1
溢出	x	0	0
除数为零或溢出	1	0	1
除数为零或溢出	0	1	1
除数为零或溢出	1	1	1
除数为零或溢出	0	0	0

如引用的算术指令功能块无数值，则输出会一直为0。

Modbus 读



当Modbus读功能块的输入En为高电平时，即启发Modbus 读功能块，控制器可透过内建的RS232及RS485通讯口读取其他Modbus 装置数据；且当有成功的读取其他Modbus装置信息时，Modbus输出则打开(1)，反之则关闭(0)。

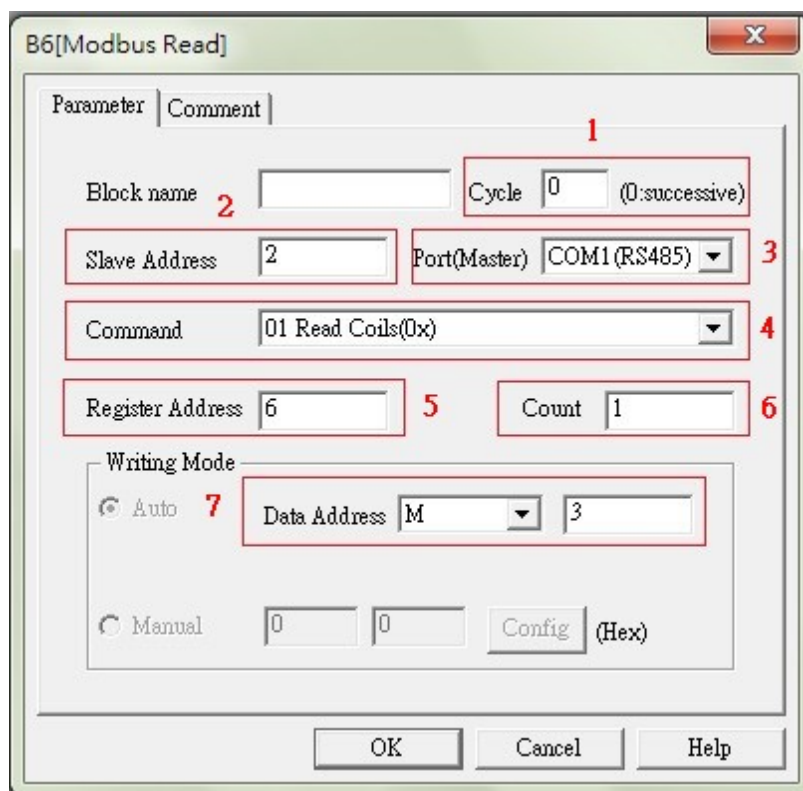
当有讯号触发输入R的同时，会复位输出Q及让Modbus 读功能块失效。

连接	描述
输入 En	输入En为高电平时，即启发Modbus 读功能块。
输入 R	当有讯号触发输入R的同时，会复位输出Q及让Modbus 读功能块失效。 输入R(复位)优先级高于输入En。
参数	<p>Cycle : transmission times: 0 ~ 9999 (cycle = 0 : continuous transmission)</p> <p>从站地址 : 1 ~ 255，初始设定值为1</p> <p>通讯口(主站): COM0(RS232) 或 COM1(RS485) 或 COM2(RS485)</p> <p>命令 : Modbus 功能码：</p> <ul style="list-style-type: none"> 01 读线圈(0x) 02 读不连续的输入(1x) 03 读保持型寄存器(4x) 04 读输入寄存器(3x) <p>寄存器地址: 第一个 线圈/输入/寄存器 的读取地址</p> <p>数量 :所需读取的 线圈/输入/寄存器 总量</p> <ul style="list-style-type: none"> 如果命令= 01 或 02时，数量<=128 如果命令= 03 或 04时，数量<=32 <p>数据地址 : 内存储存所读取数据的起始地址</p>
输出 Q	输出Q会依据通讯的状态来置位或复位 如果通讯成功，输出Q=1 如果通讯失败，输出Q=0

备注：：当通讯口设置为主站时，本功能块才能使用。

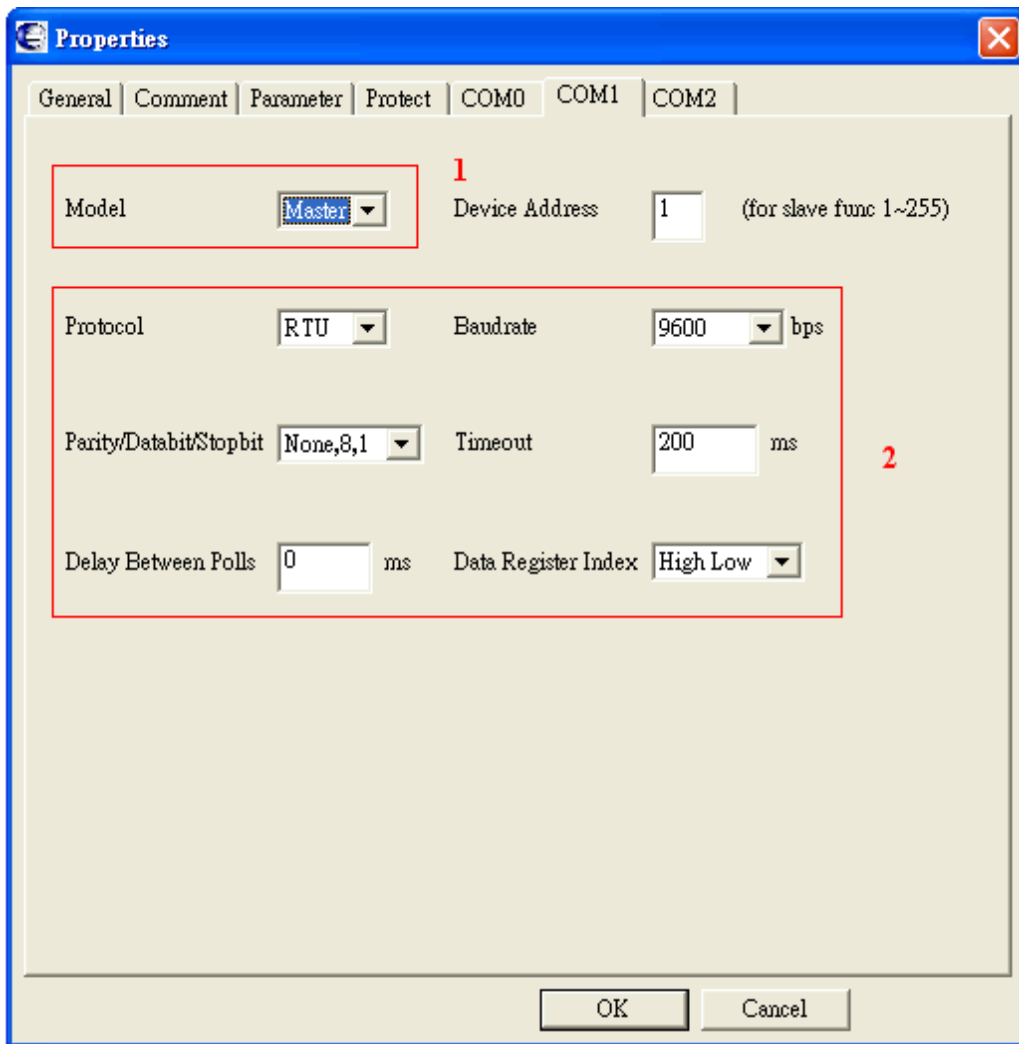
范例：读取从站装置(地址为00007)的数字量输入6(DI 6)的状态，该从站装置的ID地址为2，然后将DI 6读到的值放置于 M3寄存器，控制器透过COM 1(RS485)与这个从站装置通讯。

步骤 1：在电路程序中使用一个Modbus 读功能块，并填入参数。



- (1). Cycle = 0 (successive)
- (2). 从站地址 = 2
- (3). 通讯口(主站) = COM1(RS485)
- (4). 命令 = 01 读线圈(0x)
- (5). 寄存器地址= 6 (Modbus 地址原先为 00007，起始地址需为 = 0007 - 1 = 0006)
- (6). 数量 = 1
- (7). 数据地址 = M3

步骤 2：依下图完成设定。



(1). 在PC软件上用鼠标选取“选项”菜单，然后选择“属性”后，则会跳出上列图示，然后选择“通讯口1”，“模块”选择“主机”。

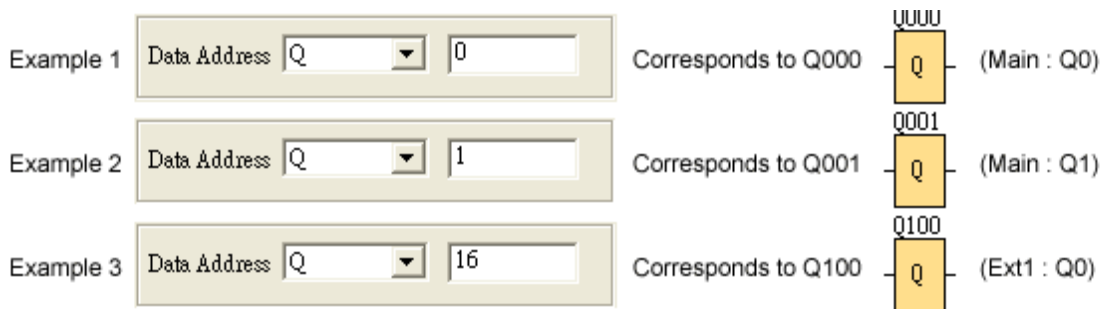
(2). 选取“通讯协议”及相关参数设置

步骤3：当输入En=1时，控制器将会透过COM1持续的传送Modbus信息

如果通讯成功，输出Q=1

如果通讯失败，输出Q=0

备注：所有的数字量、模拟量输出、输入及模拟量寄存器(AM)在数据地址字段为连续的，如下图范例三所示，如果输出Q设置为Q16，则实际的输出会变为Q100。此规则适用在所有的数字量及模拟量，规则如下表所示。(本备注亦适用于 Modbus 写功能块)



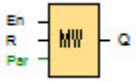
参数设定表

模式	Modbus 对话框	I,Q,AI,AQ 地址	I,Q,AI,AQ 功能块
主机	I0-I31	I000-I031	Main : I0-31
	Q0-Q15	Q000-Q015	Main : Q0-Q15
	AI0-AI7	AI000-AI007	Main : AI0 - AI7
	AQ0-AQ3	AQ000-AQ003	Main : AQ0-AQ3
拓展 1 (Ext1)	I32-I63	I100-I131	Ext1 : I0 - I31
	Q16-Q31	Q100-Q115	Ext1 : Q0-Q15
	AI8-AI15	AI100-AI107	Ext1 : AI0-AI7
	AQ4-AQ7	AQ100-AQ103	Ext1 : AQ0-AQ3
拓展 2 (Ext2)	I64-I95	I200-I231	Ext2 : I0-I31
	Q32-Q47	Q200-Q215	Ext2 : Q0-Q15
	AI16-AI23	AI200-AI207	Ext2 : AI0-AI7
	AQ8-AQ11	AQ200-AQ203	Ext2 : AQ0-AQ3
拓展3 ~ 拓展7 的参数与拓展1及拓展2相同。			

数据形式说明

名称	数据形式
I, Q, M	位
AI, AQ, AM	Signed Short Integer (16 bits)

Modbus 写



当Modbus写功能块的输入En为高电平时，即启发Modbus 写功能块，控制器可透过内建的RS232及RS485通讯口写入其他Modbus 装置数据；且当有成功的写入其他Modbus装置信息时，Modbus输出则打开(1)，反之则关闭(0)。

当有讯号触发输入R的同时，会复位输出Q及让Modbus 写功能块失效。

连接	描述
输入 En	输入En为高电平时，即启发Modbus 写功能块
输入 R	当有讯号触发输入R的同时，会复位输出Q及让Modbus 读功能块失效。 输入R(复位)优先级高于输入En。
参数	<p>Cycle : transmission times: 1 ~ 9999 , cycle = 0 --> continuous transmission 从站地址 : 1 ~ 255 , 初始设定值为1 通讯口(主站): COM0(RS232) 或 COM1(RS485) 或 COM2(RS485) 命令 : Modbus 功能码 :</p> <ul style="list-style-type: none"> 05 写入线圈 06 写入单一寄存器 15 写入多个线圈 16 写入多个寄存器 <p>寄存器地址: 第一个 线圈/输入/寄存器 的写入地址 数量 :所需写入的 线圈/输入/寄存器 总量</p> <ul style="list-style-type: none"> 如果命令=05 或 06时，数量=1 如果命令=15时，数量 <= 32 如果命令=16时，数量 <= 2 <p>写入模式 : 可依下列方式写入数据 :</p> <ul style="list-style-type: none"> 自动 : 数据地址，数据地址会依起始地址来依序写入寄存器中。 手动 : 可写入指定的单一寄存器地址。
输出 Q	输出Q会依据通讯的状态来置位或复位 如果通讯成功，输出Q=1 如果通讯失败，输出Q=0

备注 : :当通讯口设置为主站时，本功能块才能使用。

范例： 将数据写入控制器数字量输入6 (地址为00007)，然后由通讯口COM1(RS485)的从机数字量输出2(地址为00018)输出；从机为分布式IO模块，以及从机地址为1。

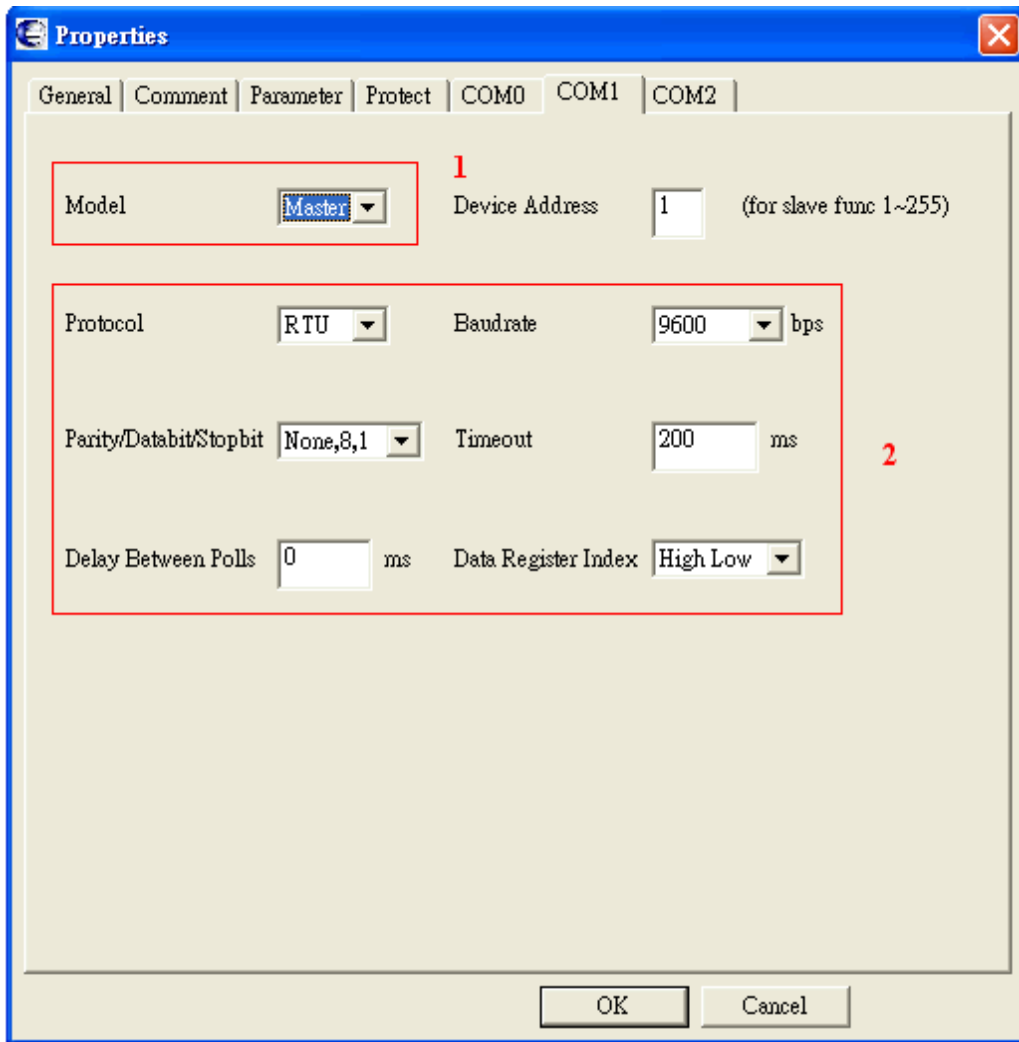
步骤 1： 在电路程序中使用一个**Modbus 写**功能块，并填入参数

The screenshot shows the 'B7[Modbus Write]' configuration window. The parameters are as follows:

Parameter	Value
Block name	
Cycle	1 (0:successive)
Slave Address	1
Port(Master)	COM1 (RS485)
Command	05 Write Single Coil
Register Address	17
Count	1
Writing Mode	Auto
Data Address	I
Manual Input 1	0
Manual Input 2	0

- (1). Cycle = 1
- (2). 从站地址= 1
- (3). 通讯口(主站) = COM1(RS485)
- (4). 命令= 05写入线圈
- (5). 寄存器地址= 0017 (Modbus地址原先为00018, 起始地址需为= 0018 - 1 = 0017)
- (6). 数量= 1
- (7). 数据地址= I6

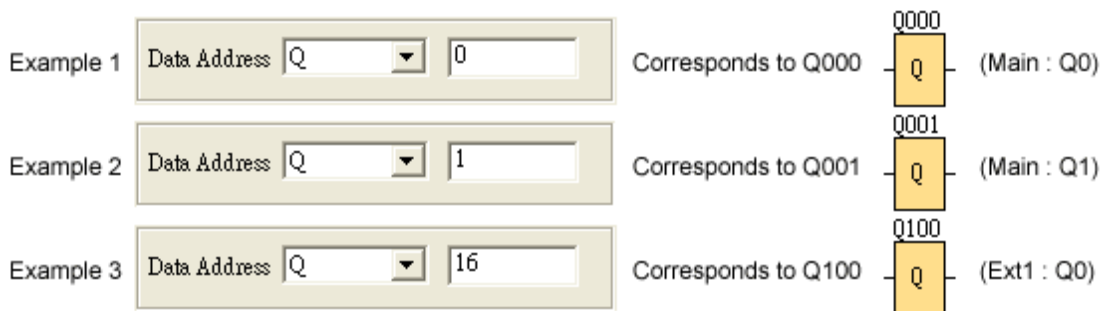
步骤 2： 依下图完成设定。



- (1). 在PC软件上用鼠标选取“选项”菜单，然后选择“属性”后，则会跳出上列图示，然后选择“通讯口1”，“模块”选择“主机”。
- (2). 选取“通讯协议”及相关参数设置

步骤3：当输入En=1时，控制器将会透过COM1持续的传送Modbus信息
如果通讯成功，输出Q=1
如果通讯失败，输出Q=0

备注：所有的数字量、模拟量输出、输入及模拟量寄存器(AM)在数据地址字段为连续的，如下图范例三所示，如果输出Q设置为Q16，则实际的输出会变为Q100。此规则适用在所有的数字量及模拟量，规则如下表所示。(本备注亦适用于 **Modbus 读**功能块)



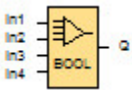
参数设定表

模式	Modbus对话框	I,Q,AI,AQ地址	I,Q,AI,AQ功能块
主机	I0-I31	I000-I031	Main : I0-31
	Q0-Q15	Q000-Q015	Main : Q0-Q15
	AI0-AI7	AI000-AI007	Main : AI0 - AI7
	AQ0-AQ3	AQ000-AQ003	Main : AQ0-AQ3
拓展1 (Ext1)	I32-I63	I100-I131	Ext1 : I0 - I31
	Q16-Q31	Q100-Q115	Ext1 : Q0-Q15
	AI8-AI15	AI100-AI107	Ext1 : AI0-AI7
	AQ4-AQ7	AQ100-AQ103	Ext1 : AQ0-AQ3
拓展2 (Ext2)	I64-I95	I200-I231	Ext2 : I0-I31
	Q32-Q47	Q200-Q215	Ext2 : Q0-Q15
	AI16-AI23	AI200-AI207	Ext2 : AI0-AI7
	AQ8-AQ11	AQ200-AQ203	Ext2 : AQ0-AQ3
拓展3 ~ 拓展7 的参数与拓展1及拓展2相同。			

数据形式说明

名称	数据形式
I, Q, M	Bit
AI, AQ, AM	Signed Short Integer (16 bits)

布尔函数



布尔函数功能块将根据输入的组合来执行输出。该功能块拥有四个输入端，然后可依需求设置16种组合(可参考下列真值表)。而输出值可依需求来调整，相关配置的组合数量将取决于已连接的输入量。如未连接的输入点将被设置为0。

下图为布尔函数真值表范例：

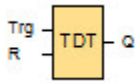
Index	In1	In2	In3	In4	Output Set
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	0	1
4	0	0	1	1	0
5	0	1	0	0	0
6	0	1	0	1	1
7	0	1	1	0	0
8	0	1	1	1	1
9	1	0	0	0	1
10	1	0	0	1	1
11	1	0	1	0	1
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	1
16	1	1	1	1	0

└──────────────────────────┘
└──┘
 Combinations of Inputs Output status

参数

- 当要配置真值表的输出值参数时，至少要有有一个以上的输入。
- 输出值可以配置为0(关闭)，以及1(打开)。
- 当选择的输出为ON及结果为True时，会根据所配置的真值表来输出。
- 当选择的输出为OFF且结果为True时，会根据所配置的真值表倒数来输出。

TDT触发日期时间



可在一个特定的存储器中记录当前时间(年/月/日/时/分/秒)。

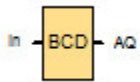
连接	描述
输入 Trg	当输入Trg由0变成1时，即开始将日期及时间写入存储器。(年年月月日时时分分秒秒)
输入 R	将输出及内存数据复位。
参数	可保持性： ’ 勾选’ 该字段则表示该状态在存储器中可以保持
输出 Q	当写入成功时，输出Q = 1.

计算规则

当输入Trg由0变成1时，即开始将日期及时间写入存储器。

Modbus 地址	描述	读/写	备注
42001	输出状态 _ B0	R	B0
42002	年年月月 (Year/Month) _ B0	R	B0
42003	日日时时 (Day/Time) _ B0	R	B0
42004	分分秒秒 (Minute/Second) _ B0	R	B0
42005	输出状态 _ B1	R	B1
42006	年年月月 (Year/Month) _ B1	R	B1
42007	日日时时 (Day/Time) _ B1	R	B1
42008	分分秒秒 (Minute/Second) _ B1	R	B1
.....			

BCD二-十进制编码



二-十进制编码转换：

连接	描述
输入 IN	整数值 值范围0 ~ 9999
输出 AQ	整数值

計數規則

如果輸入IN = 1234，則輸出AQ = 0x1234

如果輸入IN = 9999，則輸出 AQ = 0x9999

BIN 二进制码



十-二进制编码转换：

连接	描述
输入 IN	整数值 值范围：0x0000 ~ 0x9999 ，且每一位数不得超过9
输出 AQ	整数值 (如输入为无效值，则输出Q=0)

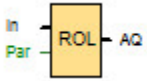
计算规则

如果输入 $IN = 0x1234$ ，则输出 $AQ = 1234$

如果输入 $IN = 0x9999$ ，则输出 $AQ = 9999$

如果输入 $IN = 0x12A4$ ，则输出 $AQ = 0$ (如输入为无效值，则输出 $Q=0$)

ROL 左旋转



将一个整数值比特数做左旋转，该旋转单位为16 bits.

连接	描述
输入 IN	任一整数值
参数	NbR : 1个比特数的旋转 (值范围 [1..15])
输出 AQ	左旋转值 (如果NbR <= 0，则不会有输出)

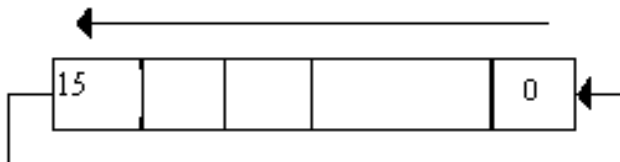
参数 NbR

NbR的值可引用由程序中其他功能块的实际值。

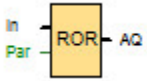
计数规则

如果输入IN = 21385 (Binary: 0101 0011 1000 1001) , NbS = 2

则输出AQ = 20005 (Binary: 0100 1110 0010 0101)



ROR 右旋转



将一个整数值的比特数做右旋转，该旋转单位为16 bits.

连接	描述
输入 IN	任一整数值
参数	NbR : 1个比特数的旋转 (值范围 [1..15])
输出 AQ	右旋转值 (如果NbR <= 0，则不会有输出)

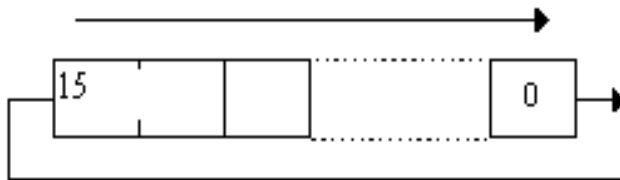
参数 NbR

NbR的值可引用由程序中其他功能块的实际值。

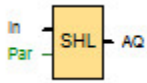
计算规则

如果输入IN= 5001 (Binary: 0001 0011 1000 1001), NbS = 2

则输出 AQ = 17634 (Binary: 0100 0100 1110 0010)



SHL 左移位



将16个比特数的整数值左移位，且在最小的一个比特数位置放置0。

连接	描述
输入 IN	任一整数值
参数	NbS : 1个比特数的移位 (值范围 [1..15])
输出 AQ	左移位值 (如果NbS <= 0，则不会有输出) 最小的一个比特数位置放置0

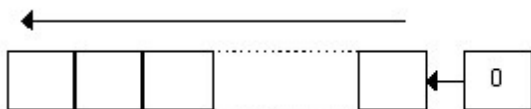
参数 NbS

NbS的值可引用由程序中其他功能块的实际值。

计算规则

如果输入IN= 5001 (Binary: 0001 0011 1000 1001) , NbS = 1

则输出AQ = 10002 (Binary: 0010 0111 0001 0010)



SHR 右移位



将16个比特数的整数值左移位，且在最大的一个比特数位置放置0。

连接	描述
输入 IN	任一整数值
参数	NbS : 1个比特数的移位 (值范围 [1..15])
输出 AQ	右移位值 (如果NbS <= 0，则不会有输出) 最大的一个比特数位置放置0

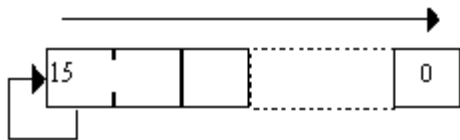
参数 NbS

NbS的值可引用由程序中其他功能块的实际值。

计算规则

如果输入IN= 5001 (Binary: 0001 0011 1000 1001) , NbS = 1

则输出 AQ = 2500 (Binary: 0000 1001 1100 0100)



AND_MASK (与屏蔽)



输入In比特跟屏蔽功能块里的比特执行逻辑AND(与)的运算。

连接	描述
输入 IN	任一整数值
参数	MSK : 16-比特值
输出 AQ	输入In比特跟屏蔽功能块里的比特执行逻辑AND(与)的运算。

参数 MSK

MSK的值可引用由程序中其他功能块的实际值。

OR_MASK (或屏蔽)



输入In比特跟屏蔽功能块里的比特执行逻辑OR (或)的运算。

连接	描述
输入 IN	任一整数值
参数	MSK : 16-比特值
输出 AQ	输入In比特跟屏蔽功能块里的比特执行逻辑OR (或)的运算。

参数 MSK

MSK的值可引用由程序中其他功能块的实际值。

NOT_MASK (非屏蔽)



输入In比特跟屏蔽功能块里的比特执行逻辑NOT (非)的运算。

连接	描述
输入 IN	任一整数值
输出 AQ	输入In比特跟屏蔽功能块里的比特执行逻辑NOT (非)的运算。

NAND_MASK (与非屏蔽)



输入In比特跟屏蔽功能块里的比特执行逻辑NAND (与非)的运算。

连接	描述
输入 IN	任一整数值
参数	MSK : 16-比特值
输出 AQ	输入In比特跟屏蔽功能块里的比特执行逻辑NAND (与非)的运算。

参数 MSK

MSK的值可引用由程序中其他功能块的实际值。

NOR_MASK (或非屏蔽)



输入In比特跟屏蔽功能块里的比特执行逻辑NOR (或非)的运算。

连接	描述
输入 IN	任一整数值
参数	MSK : 16-比特值
输出 AQ	输入In比特跟屏蔽功能块里的比特执行逻辑NOR (或非)的运算。

参数 MSK

MSK的值可引用由程序中其他功能块的实际值。

XOR_MASK (异或屏蔽)



输入In比特跟屏蔽功能块里的比特执行逻辑XOR (异或)的运算。

连接	描述
输入 IN	任一整数值
参数	MSK : 16-比特值
参数 AQ	输入In比特跟屏蔽功能块里的比特执行逻辑XOR (异或)的运算。

参数 MSK

MSK的值可引用由程序中其他功能块的实际值。

数据阵列 最大/最小/平均值



获取数据阵列 最大/最小/平均值。

连接	描述
参数	模式： 最大/最小/平均值 数据地址： AMx / AIx / AQx 数组起始地址 数量： 数组值 [1~32]
输出 AQ	输出数组的 最大/最小/平均值

参数模式与数量

模式与数量的值可引用由程序中其他功能块的实际值。

计算规则

当模式=最大值，数据地址=AM2，数量=3，(AM2=2, AM3=6, AM4=13)：则输出AQ = 13

当模式=最小值，数据地址=AM2，数量=3，(AM2=2, AM3=6, AM4=13)：则输出 AQ = 2

当模式=平均值，数据地址=AM2，数量=3，(M2=2, AM3=6, AM4=13)：则输出AQ = $(2+6+13) / 3 = 7$

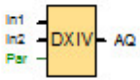
当模式=最大值，数据地址=AQ3，数量=2，(AQ3=2, AQ4=6)：则输出 AQ = 6

参数设置表：

模块	数组对话框	AI,AQ 号码	AI,AQ 块
主机	AI0-AI7	AI000-AI007	Main : AI0 - AI7
	AQ0-AQ3	AQ000-AQ003	Main : AQ0-AQ3
拓展 1	AI8-AI15	AI100-AI107	Ext1 : AI0-AI7

(Ext1)	AQ4-AQ7	AQ100-AQ103	Ext1 : AQ0-AQ3
拓展 2	AI16-AI23	AI200-AI207	Ext2 : AI0-AI7
(Ext2)	AQ8-AQ11	AQ200-AQ203	Ext2 : AQ0-AQ3
拓展3 ~ 拓展7 的参数与拓展1及拓展2相同。			

ACMX_MI_AV 数据 最大/最小/平均值



获取输入In的数据 最大/最小/平均值。

连接	描述
输入 IN1	任一16比特的输入值
输入 IN2	任一16比特的输入值
参数	模式： 最大/最小/ 平均值
输出 AQ	将输入IN1及IN2的 最大/最小/平均值输出

参数模式

模式的值可引用由程序中其他功能块的实际值。

计算规则

当输入IN1= 2，IN2= 8，模式=最大值：则输出AQ = 8

当输入IN1= 2，IN2= 8，模式=最小值：则输出AQ = 2

当输入IN1= 2，IN2= 8，模式=平均值：则输出AQ = $(2 + 8) / 2 = 5$

RAND 随机数发生器



在指定范围内给予一个随机发生的整数。

连接	描述
参数	基数 ：设置随机发生数的范围
输出 AQ	随机值 [0..基数-1]

参数基数

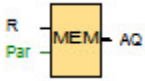
基数的值可引用由程序中其他功能块的实际值。

计算规则

如果基数=10：则输出AQ = 0至9

如果基数=15：则输出AQ = 0至14

MEM 参考值输出



获取其他已设置功能的参数。

连接	描述
输入 R	当输入R为0时，输出AQ会依参数值输出
参数	参数：获取其他已设置功能的参数
输出 AQ	将参数的值输出 值范围：-32768 ~ 32767

The screenshot shows a dialog box titled 'BO[MEM]'. It has a 'Parameter' tab selected. The dialog contains the following fields and controls:

- Block name: A text input field.
- Block: A dropdown menu with 'B1 [On/Off Delay]' selected.
- Index: A dropdown menu with 'Current Time' selected.
- Retentivity: A checkbox that is currently unchecked.
- Buttons: 'OK', 'Cancel', and 'Help' at the bottom.

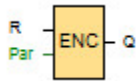
计算规则

如果输入R为0，则输出AQ 会将当前的参数值输出。

(值范围： -32768 ~ 32767)

如果输入R为1，则输出AQ = 0.

ENCODER 编码器



由输入I0/I1或 I2/I3获取编码器的数据。

连接	描述
输入 R	Reset output
参数	起始值： 设置开始计数之起始值 值范围：-999999 ~ 999999 编码器讯号源： 0 --> I0/I1 , 1 --> I2/I3 On： 接通阈值 值范围：-999999 ~ 999999 Off： 关断阈值 值范围：-999999 ~ 999999 可保持性： '勾选' 该字段则表示该状态在存储器中可以保持。
输出 Q	输出Q会根据编码器的当前值、接通阈值、关断阈值来执行置位或复位的动作。

参数接通阈值及关断阈值

接通阈值及关断阈值的值可引用由程序中其他功能块的实际值。

计算规则

- 如果接通阈值 \geq 关断阈值，则：
 - 如果当前计数值 $>$ 接通阈值，则输出 $Q = 1$
 - 如果当前计数值 $<$ 关断阈值，则输出 $Q = 0$
- 如果接通阈值 $<$ 关断阈值，则：
 - 如果接通阈值 \leq 当前计数值 $<$ 关断阈值，则输出 $Q = 1$
 - 如果当前计数值 \geq 关断阈值或 计数值 $<$ 接通阈值，则输出 $Q = 0$

Stepping Motor Control 步进电机控制



产生信号来控制步进电机。

连接	描述
输入 En	0 : 开始产生脉冲信号 1 : 停止产生脉冲信号
输入 Dir	0 : CW (顺时针方向) 1 : CCW (逆时针方向)
参数	模式 : 半步 (单相或二相步进电机, 每步0.9度) 整步 (单相或二相步进电机, 每步1.8度) 输出Pin : 0 --> Q0~Q3 1 --> Q4~Q7 速度 : 0.01 毫秒/步 (半步模式) 0.02 毫秒/步 (整步模式) 值范围 : 0 ~ 99999999
输出 Q	输出Q会根据输入En的值来置位及复位。

参数周期T

周期T的值可引用由程序中其他功能块的实际值。

范例

设置步进电机每1000毫秒转一圈(360度)。

● 模式 = 半步

1圈 = 360度 = 0.9(度/步) x 400(步)

--> 1000毫秒需送出400步(脉冲)

--> 即1步需要1000/400 = 2.5 毫秒

--> 速度 = 2.5 / 0.01 = 250 (0.01毫秒/步)

● 模式 = 整步

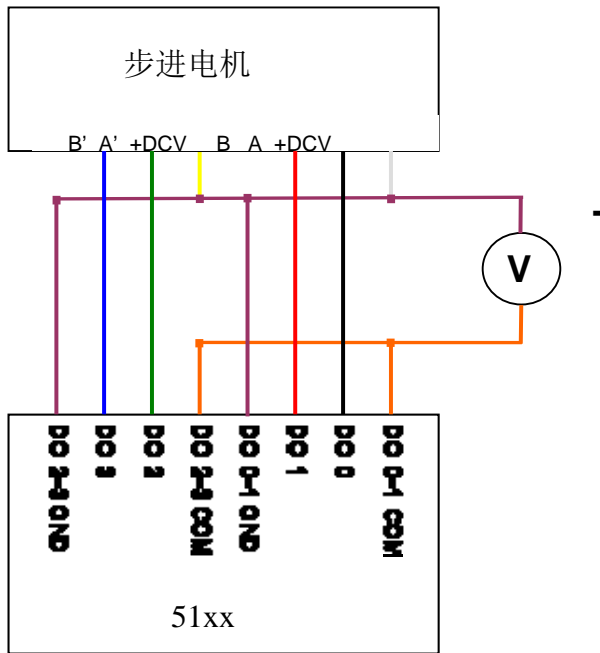
1圈 = 360 度 = 1.8(度/步) x 200(步)

--> 1000毫秒需送出200步(脉冲)

--> 即1步需要1000/200 = 5 毫秒

--> 周期T = 5 / 0.02 = 250 (0.02毫秒/步)

控制器与步进电机接线示意图



* 仅支持0.9或1.8度，二相六线步进电机

Stepping Motor Control (Edge) 步进电机控制 (沿)



产生配置信号(脉冲)来控制步进电机。

连接	描述
输入 Trg	当输入Trg由0变成1时，开始产生脉冲信号。
输入 R	将输出复位，以及停止脉冲信号。
输入 Dir	0 : CW (顺时针方向) 1 : CCW (逆时针方向)
参数	Mode : 半步 (单相或二相步进电机，每步0.9度) 整步 (单相或二相步进电机，每步1.8度) 输出Pin : 0 --> Q0~Q3 1 --> Q4~Q7 速度 : 0.01 毫秒/步 (半步模式) 0.02 毫秒/步 (整步模式) 值范围：0 ~ 99999999 步数 : 输出的步数值 值范围：0 ~ 99999999
输出 Q	当开始产生脉冲信号时，输出Q =1

参数周期T以及步数

周期T以及步数的值可引用由程序中其他功能块的实际值。

范例

设置步进电机每1000毫秒转一圈(360度)，且在转了50圈后停止。

- 模式 = 半步

1圈= 360度 = 0.9(度/步) x 400(步)

--> 1000毫秒需送出400步(脉冲)

--> 即1步需要1000/400 = 2.5 毫秒

--> 速度 = $2.5 / 0.01 = 250$ (0.01毫秒/步)
步数值=50 (圈) x 400 (步数/圈)=2000步

● 模式 = 整步

1圈 = $360 \text{ 度} = 1.8(\text{度/步}) \times 200(\text{步})$

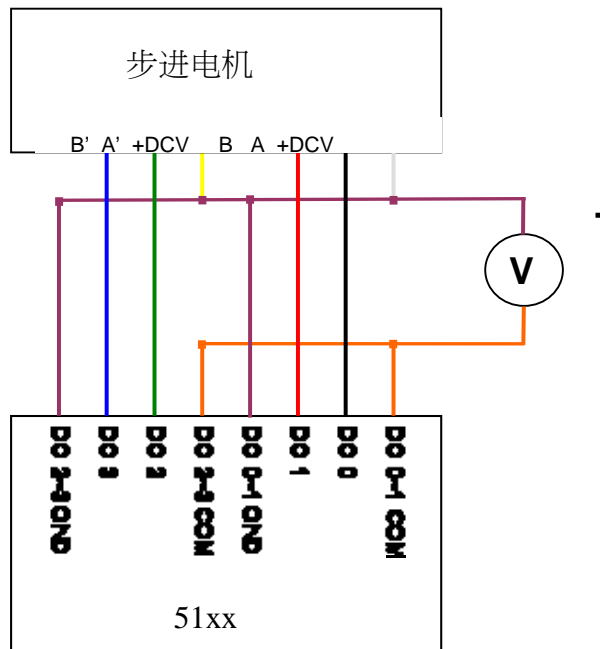
--> 1000毫秒需送出200步(脉冲)

--> 即1步需要 $1000/200 = 5$ 毫秒

--> 周期 $T = 5 / 0.02 = 250$ (0.02毫秒/步)

步数=50 (圈) x 200 (步数/圈) =1000 步

控制器与步进电机接线示意图



* 仅支持0.9或1.8度，二相六线步进电机

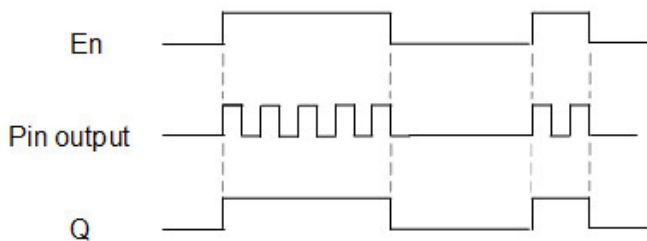
PTO (Pulse train output) 脉冲串输出



脉冲串连续输出(duty=50%)

连接	描述
输入 En	当输入En=1时，开始连续产生脉冲串输出
参数	<p>输出Pin：脉冲串输出pin Q0 ~ Q3. 值范围0 ~ 3</p> <p>脉冲宽度：0 ~ 10000000 (0.02毫秒/脉冲). 值范围：0 ~ 99999999</p> <ul style="list-style-type: none"> ● 脉冲宽度 = 1 → 1 x 0.02 毫秒/脉冲 → 0.00002 秒/脉冲 → 50K Hz ● 脉冲宽度 = 10000000 → 10000000 x 0.02毫秒/脉冲 = 200000毫秒/脉冲 → 200秒/脉冲 → 0.005 Hz
输出 Q	当有脉冲信号产生时，输出Q=1

时序图



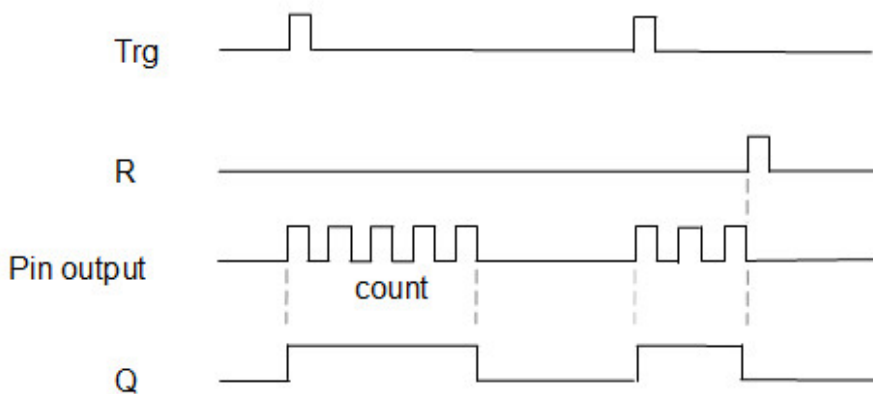
PTOE (Pulse train output) 脉冲串输出 (沿)



配置脉冲串连续输出(duty=50%)

连接	描述
输入 Trg	当输入Trg由0变成1时，开始输出配置脉冲串信号
输入 R	将输出及计数值复位
参数	<p>输出Pin：脉冲串输出pin Q0 ~ Q3. 值范围0 ~ 3</p> <p>脉冲宽度：0 ~ 99999999 (0.02毫秒/脉冲). 值范围：0 ~ 99999999</p> <ul style="list-style-type: none"> ● 脉冲宽度= 1 → 1 x 0.02毫秒/脉冲 → 0.00002 秒/脉冲 → 50K Hz ● 脉冲宽度= 10000000 → 10000000 x 0.02毫秒/脉冲 = 200000毫秒/脉冲 → 200 秒/脉冲 → 0.005 Hz <p>计数值：脉冲输出配置值. 值范围：0 ~ 99999999</p>
输出 Q	当有脉冲信号产生时，输出Q=1

时序图



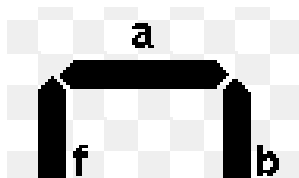
SEG 七段显示器



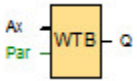
将十六进制值转换成七段显示编码输出。

连接	描述
输入 Trg	当输入Trg =1时，开始将十六进制值转换成七段显示编码输出。
输入 Ax	十六进制值。值范围：0(Hex) ~ F(Hex)
输出 AQ	转换成七段显示编码输出。

Ax	-gfe dcba	Output AQ
0	0011 1111	0x3F
1	0000 0110	0x06
2	0101 1011	0x5B
3	0100 1111	0x4F
4	0110 0110	0x66
5	0110 1101	0x6D
6	0111 1101	0x7D
7	0000 0111	0x07
8	0111 1111	0x7F
9	0110 0111	0x67
A	0111 0111	0x77
B	0111 1100	0x7C
C	0011 1001	0x39
D	0101 1110	0x5E
E	0111 1001	0x79
F	0111 0001	0x71



Word to Bit 字转位

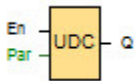


字符转位(比特)

连接	描述
输入 Ax	16-比特值
参数	Nb : 截取配置的比特值输出 值范围 : 0 ~ 15
输出 Q	将比特值输出

输入 Ax	二进制	Nb	输出
0x1234	0001 0010 0011 0100	0	0
		1	0
		2	1
		3	0
		4	1
		5	1
		6	0
		7	0
		8	0
		9	1
		10	0
		11	0
		12	1
		13	0
		14	0
		15	0

UDC 通信



通过通信口传输自定义格式的数据，当前缀或字尾的字符被选择传输时，传输的数据将会添加一个前缀或字尾的字符；当前缀或字尾字符在接收端被选中时，接收到的数据将会优先检查前缀或字尾，如果信号正确，数据将会被放置于接收的地址。

连接	描述
输入 En	0 --> 无效, 1-->启动.
参数	通讯口: 0 ~ 2 (com0 ~ com2) Tx_Pre_Char: 传输前缀字符 Tx_End_Char: 传输字尾字符 Tx_Start: AI0~AI63 / AM0~AM511 / AQ0 ~ AQ31 Tx_Num: 传输字符 值范围: 1~127 (不包含前缀及字尾) Rx_Pre_Char: 接收前缀字符 Rx_End_Char: 接收字尾字符 Rx_Start: AM0~AM511 Rx_Num: 接收字符 值范围: 1~127 (不包含前缀及字尾) 周期: 传送次数 值范围: 0 ~ 9999 (周期 = 0将会连续传送)
输出 Q	当传送成功时, 输出Q=1

范例：

Tx_Pre_Char = 0x03, Tx_End_Char = 0x0A, Tx_Start = AM0, Tx_Num = 3,
 Rx_Pre_Char = 0x03, Rx_End_Char = 0x0A, Rx_Start = AM16, Rx_Num = 4,
 AM0=0x2211, AM1=4433, AM2=0x6655, 周期 = 1, 通讯口= 0.

● Tx传输：

- 当传输前缀及字尾字符未检查时，输入En=1时，数据格式会依下列方式从COM0(RS-232)传送一次：
Tx: 11 22 33 44 55 66 (十六进制)
- 当传输前缀及字尾字符经检查后，输入En=1时，数据格式会依下列方式从COM0(RS-232)传送一次：
Tx: 03 11 22 33 44 55 66 0A (十六进制)

● Rx：

1. 当传输前缀及字尾字符未检查时，且数据为4字符(8 bytes)时，接收会通过COM0(RS-232)连续写入AM16~AM19。假设完成接收4字符(8 bytes)时，则：

Rx : 99 88 77 66 55 44 33 22 (十六进制)

AM16 = 0x8899 , AM17 = 0x6677 , AM18 = 0x4455 , AM19 = 0x2233

2. 当传输前缀及字尾字符经检查后，前缀及字尾字符会通过COM0(RS-232)检查。如果数据正确，前缀及字尾将会被除去，然后将字符值写入AM16~AM19。

假设完成接收后，则数据为：

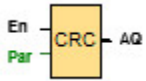
Rx : 03 99 88 77 66 55 44 33 22 0A (十六进制)

AM16 = 0x8899 , AM17 = 0x6677 , AM18 = 0x4455 , AM19 = 0x2233

参数配置表

模式	通信对话框	AI,AQ 地址	AI,AQ 块
主机	AI0-AI7	AI000-AI007	Main : AI0 - AI7
	AQ0-AQ3	AQ000-AQ003	Main : AQ0-AQ3
拓展 1 (Ext1)	AI8-AI15	AI100-AI107	Ext1 : AI0-AI7
	AQ4-AQ7	AQ100-AQ103	Ext1 : AQ0-AQ3
拓展 2 (Ext2)	AI16-AI23	AI200-AI207	Ext2 : AI0-AI7
	AQ8-AQ11	AQ200-AQ203	Ext2 : AQ0-AQ3
拓展3 ~ 拓展7 与拓展1及拓展2相同。			

CRC16 循环冗余校验



计算CRC16循环冗余校验值。

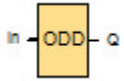
连接	描述
输入 En	当输入 En=1时，开始计算CRC16循环冗余校验值。
参数	类型：CRC16循环冗余校验 起始地址：AM0 ~ AM511 Number：数量 (字符串). 值范围：1~128
输出 AQ	直接将CRC16循环冗余校验值输出

范例：

起始地址= AM0，number = 5

当输入En=1时，输出AQ会将CRC16循环冗余校验值配置于AM0~ AM4.

ODD 奇数



检查一个整数的奇偶性：结果将为奇数或偶数。

连接	描述
输入 IN	任一整数值
输出 Q	如输入值为奇数，则输出为打开(1) 如输入值为偶数，则输出为关闭(0)

计算规则

如果输入IN= 2：则输出Q = 0（偶数）

如果输入IN= 3：则输出Q = 1（奇数）

EVEN 偶数



检查一个整数的奇偶性：结果将为奇数或偶数。

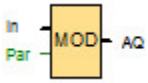
连接	描述
输入 IN	任一整数值
输出 Q	如输入值为偶数，则输出为打开(1) 如输入值为奇数，则输出为关闭(0)

计算规则

如果输入IN= 2：则输出Q = 1（偶数）

如果输入IN= 3：则输出 Q = 0（奇数）

MOD 模除



计算输入整数的模除。

连接	描述
输入 IN	任一整数值
参数	基数 ：除数
输出 AQ	模除计算(输入模除的基数) 如果基数=0，计算结果将为0

参数 基数

基数的值可引用由程序中其他功能块的实际值。

计算规则

1. 如果输入IN及基数都同为正数或负数，则输出AQ也将与输入IN及基数同为正数或负数。

例如输入IN = 42，基数= 5，则输出AQ=2 ($42 = 5 \times 8 + 2$)

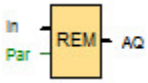
例如输入IN = -42，基数=-5，则输出AQ=-2 ($-42 = -5 \times (-8) + (-2)$)

2. 如果输入IN与基数不是相同的正数或负数(符号)，则输出AQ会根据基数的符号来决定。

例如输入IN = 42，基数=-5，则输出AQ=-3 ($42 = -5 \times (-9) + (-3)$)

例如输入IN = -42，基数=5，则输出AQ=3 ($-42 = 5 \times (-9) + 3$)

REM 余数



整数经计算过后的余数。

连接	描述
输入 IN	任一整数值
参数	基数 ：除数
输出 AQ	输出为计算后的余数 如果基数=0，则结果=0

参数 基数

基数的值可引用由程序中其他功能块的实际值。

计算规则

1. 如果输入IN及基数都同为正数或负数，则输出AQ也将与输入IN及基数同为正数或负数。

例如输入IN = 42，基数=5，则输出AQ=2 ($42 = 5 \times 8 + 2$)

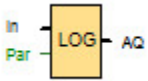
例如输入IN = - 42，基数=5，则输出AQ=-2 ($- 42 = -5 \times (-8) + (-2)$)

2. 如果输入IN与基数不是相同的正数或负数(符号)，则输出AQ会根据输入IN的符号来决定。

例如输入IN = 42，基数=-5，则输出AQ=2 ($42 = -5 \times (-8) + 2$)

例如输入IN = - 42，基数=5，则输出AQ=-2 ($- 42 = 5 \times (-8) + (-2)$)

LOG 对数



实际值计算过后的对数。

连接	描述
输入 IN	必须大于0
参数	基数： e / 2 / 10 乘数： 乘数范围(-10000.00 ~ 10000.00)
输出 AQ	输入的对数值乘以乘数的值。

参数 基数

基数的值可引用由程序中其他功能块的实际值。

计算规则

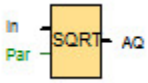
输出 AQ = 对数_{基数}(输入IN) × 乘数

例如基数= e，输入IN = 10，乘数= 10，则输出AQ = $\log_e(10) \times 10 = 2.30 \times 10 = 23$

例如基数= 2，输入IN = 10，乘数= 10，则输出AQ = $\log_2(10) \times 10 = 3.3 \times 10 = 33$

例如基数= 10，输入IN = 10，乘数= 10，则输出AQ = $\log_{10}(10) \times 10 = 1.0 \times 10 = 10$

SQRT 平方根



实际值计算过后的平方根。

连接	描述
输入 IN	输入IN至少要 ≥ 0 ，不可为负数
参数	乘数 ：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	输入的平方根值乘以乘数的值。(如果输入 $IN < 0$ ，则输出 $AQ = -1$)

计算规则

输出 $AQ = \text{平方根}(\text{输入}IN) \times \text{乘数}$

例如输入 $IN = 9$ ，乘数= 1.0，则输出 $AQ = \text{Sqrt}(9) \times 1.0 = 3$

例如输入 $IN = 4$ ，乘数= 1.0，则输出 $AQ = \text{Sqrt}(4) \times 1.0 = 2$

ABS 绝对值



实际值经计算过后的绝对值(正值)。

连接	描述
输入 IN	任一实际值
输出 AQ	绝对值(永为正数)

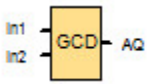
计算规则

输出 AQ = 绝对值(输入IN)

例如输入IN = 3，则输出AQ = ABS(3) = 3

例如输入IN = -3，则输出AQ = ABS(-3) = 3

GCD 最大公因子



计算输入IN1及IN2的最大公因子(GCD)。

连接	描述
输入 IN1	任一16比特值的输入值 (IN1 > 0)
输入 IN2	任一16比特值的输入值 (IN2 > 0)
输出 AQ	IN1及IN2的最大公因子 (如果IN1<=0或 IN2 <= 0，则输出AQ=0)

计算规则

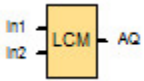
输出 AQ = 最大公因子(IN1, IN2)

例如输入IN1 = 8, IN2= 12, 则输出AQ = GCD(8,12) = 4

例如输入IN1 = 3, IN2= 5, 则输出AQ = GCD(3,5) = 1

例如输入IN1 = -8, IN2= 12, 则输出AQ = GCD(-8,12) = 0 (如果IN1<=0或 IN2 <= 0, 则输出AQ=0)

LCM 最小公倍数



计算输入IN1及IN2的最小公倍数(LCM)。

連接	描述
輸入 IN1	任一16比特值的輸入值(IN1 > 0)
輸入 IN2	任一16比特值的輸入值(IN2 > 0)
輸出 AQ	計算輸入IN1及IN2的最小公倍數(如果IN1<=0或IN2<= 0，則AQ=0)

計算規則

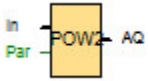
輸出 AQ = 最小公倍數(IN1 , IN2)

例如輸入IN1 = 3 , IN2 = 5 , 則輸出AQ = LCM(3,5) = 15

例如輸入IN1 = 6 , IN2 = 9 , 則輸出AQ = LCM(6,9) = 18

例如輸入IN1 = -3 , IN2 = 5 , 則輸出AQ = LCM(-3,5) = 0 (如果IN1<=0或IN2<= 0 , 則AQ=0)

POW2 2的次方



输入IN为2的次方，并计算2的次方值， 2^{IN}

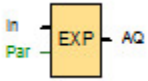
连接	描述
输入 IN	任一16比特值的输入值
参数	乘数：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	输入IN为2的次方，并计算2的次方值，然后再乘以乘数的值

计算规则

输出 $AQ = 2^{\text{输入IN}} \times \text{乘数}$

例如IN = 3，乘数 = 1.0，则输出 $AQ = 2^3 \times 1.0 = 8$

EXP 自然指数



计算自然指数值。

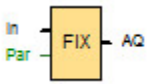
连接	描述
输入 IN	任一16比特值的输入值
参数	乘数：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	输入IN为e的次方，并计算e的次方值，然后再乘以乘数的值

计算规则

$$\text{输出 AQ} = e^{\text{输入IN}} \times \text{乘数}$$

例如输入IN = 3，乘数= 1.0，则输出AQ = $e^3 \times 1.0 = 20.085 = 20$

FIX 定点运算



定点运算输入IN的实际值。

连接	描述
输入 IN	任一16比特值的输入值
参数	基数 :从第一个基数开始舍去
输出 AQ	定点运算输入IN的实际值(如果基数大于输入IN的位数,则输出AQ=0)

参数 基数

基数的值可引用由程序中其他功能块的实际值。

计算规则

(1) 如果输入 $IN \geq 0$:

例如输入 $IN = 25836$, 基数=1, 则输出 $AQ = 25830$

例如输入 $IN = 25836$, 基数=2, 则输出 $AQ = 25800$

例如输入 $IN = 25836$, 基数=3: Output $AQ = 25000$

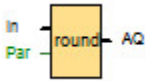
(2) 如果输入 $IN < 0$:

例如输入 $IN = -25836$, 基数=1, 则输出 $AQ = -25830$

例如输入 $IN = -25836$, 基数=2, 则输出 $AQ = -25800$

例如输入 $IN = -25836$, 基数=3, 则输出 $AQ = -25000$

ROUND 四舍五入



针对输入值做四舍五入计算。

連接	描述
輸入 IN	任一16比特值的輸入值
參數	基数 : 从第一个基数开始四舍五入
輸出 AQ	针对輸入IN的实际值做四舍五入计算(如果基数大于輸入IN的位数, 则輸出AQ=0)

参数 基数

基数的值可引用由程序中其他功能块的实际值。

计算规则

(1) 如果输入 $IN \geq 0$:

例如输入 $IN = 25836$, 基数= 1 , 则输出 $AQ = 25840$

例如输入 $IN = 25836$, 基数= 2 , 则输出 $AQ = 25800$

例如输入 $IN = 25836$, 基数= 3 , 则输出 $AQ = 26000$

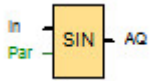
(2) 如果输入 $IN < 0$:

例如输入 $IN = -25836$, 基数= 1 , 则输出 $AQ = -25840$

例如输入 $IN = -25836$, 基数= 2 , 则输出 $AQ = -25800$

例如输入 $IN = -25836$, 基数= 3 , 则输出 $AQ = -26000$

SIN 正弦函数



计算输入IN的正弦函数(弧度)。

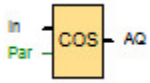
连接	描述
输入 IN	任一16比特值的输入值(度数)
参数	乘数 ：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	计算输入IN的正弦函数(设置范围：[-1.0 .. +1.0])，然后再乘以乘数的值

计算规则

$$\text{弧度} = \text{度数} \times (\pi / 180)$$

$$\text{输出AQ} = \text{正弦函数}(\text{输入IN} \times (\pi / 180)) \times \text{乘数}$$

COS 余弦函数



计算输入IN的余弦函数(弧度)。

連接	描述
輸入 IN	任一16比特值的输入值(度数)
參數	乘数：乘数范围(-10000.00 ~ 10000.00)
輸出 AQ	计算输入IN的余弦函数(设置范围：[-1.0 .. +1.0])，然后再乘以乘数的值

计算规则

$$\text{弧度} = \text{度数} \times (\pi / 180)$$

$$\text{输出AQ} = \text{余弦函数}(\text{输入IN} \times (\pi / 180)) \times \text{乘数}$$

TAN 正切函数



计算输入IN的正切函数(弧度)。

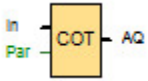
连接	描述
输入 IN	无法等于PI/2模除PI(度数)
参数	乘数 ：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	计算输入IN的正切函数，然后再乘以乘数的值。(如果输入无效值，则输出AQ = 32767)

计算规则

弧度 = 度数 \times (pi / 180)

输出AQ = 正切函数(输入IN \times (pi / 180)) \times 乘数

COT 余切函数



计算输入IN的余切函数(弧度)。

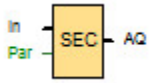
连接	描述
输入 IN	无法等于0模除PI. (度数)
参数	乘数 ：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	计算输入IN的余切函数，然后再乘以乘数的值。 (如果输入无效值，则输出AQ = 32767)

计算规则

$$\text{弧度} = \text{度数} \times (\pi / 180)$$

$$\text{输出AQ} = \text{余切函数}(\text{输入IN} \times (\pi / 180)) \times \text{乘数}$$

SEC 正割函数



计算输入IN的正割函数(弧度)。

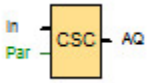
连接	描述
输入 IN	无法等于PI/2模除PI. (度数)
参数	乘数 ：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	计算输入IN的正割函数，然后再乘以乘数的值。 (如果输入无效值，则输出AQ = 32767)

计算规则

$$\text{弧度} = \text{度数} \times (\pi / 180)$$

$$\text{输出AQ} = \text{正割函数}(\text{输入IN} \times (\pi / 180)) \times \text{乘数}$$

CSC 余割函数



计算输入IN的余割函数(弧度)。

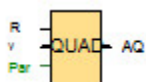
连接	描述
输入 IN	无法等于0模除PI. (度数)
参数	乘数 ：乘数范围(-10000.00 ~ 10000.00)
输出 AQ	计算输入IN的余割函数，然后再乘以乘数的值。 (如果输入无效值，则输出AQ = 32767)

计算规则

$$\text{弧度} = \text{度数} \times (\pi / 180)$$

$$\text{输出AQ} = \text{余割函数}(\text{输入IN} \times (\pi / 180)) \times \text{乘数}$$

Quadratic equation 二次方程式



计算二次方程式的值。

连接	描述
输入 Reset	当输入R為0時，输出AQ会依参数值输出
输入 y	任一整数
参数	<p>a,b,c : 任一整数 ($a < 0$)</p> <p>Dir : Dir = 0 --> $x = (-b + \sqrt{b^2 - 4a(c-y)}) / 2a$ Dir = 1 --> $x = (-b - \sqrt{b^2 - 4a(c-y)}) / 2a$</p> <p>乘数 : 乘数范围(-10000.00 ~ 10000.00)</p>
输出 AQ	计算后的x值，然后再乘以乘数的值。

计算规则

$$y = ax^2 + bx + c$$

$$\rightarrow ax^2 + bx + (c-y) = 0$$

$$\rightarrow x = (-b + \sqrt{b^2 - 4a(c-y)}) / 2a \text{ 或 } x = (-b - \sqrt{b^2 - 4a(c-y)}) / 2a$$

- 如果 $a=0$ ，且 $x=0$ ，则输出 $AQ = 0$
- 如果 $b^2 - 4a(c-y) < 0$ ，且 $x=0$ ，则输出 $AQ = 0$
- 如果 $b^2 - 4a(c-y) > 0$ ，且 $x = (-b + \sqrt{b^2 - 4a(c-y)}) / 2a$ (当 Dir = 0)

$$\text{或 } x = (-b - \sqrt{b^2 - 4a(c-y)}) / 2a \text{ (当 Dir = 1)}$$

则输出 $AQ = X \times \text{乘数}$

数据记录



于电路程序中，可配置一个数据记录来对功能块以及存储区域的实际值进行记录。

连接	描述
输入 En	输入En发生上升沿(由0变成1)时，数据记录块开始记录数据。

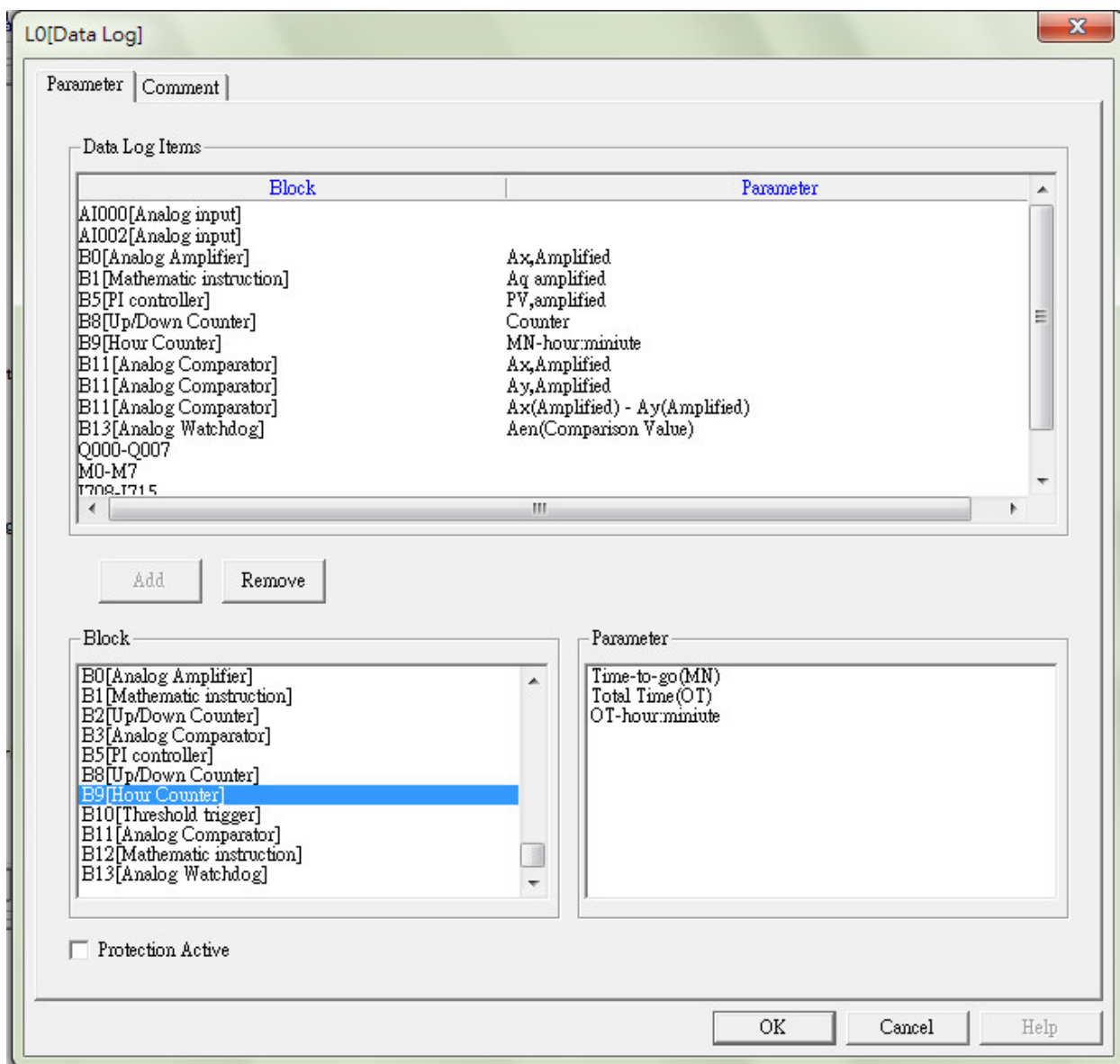
配置数据记录

在一个电路程序中，仅能配置一个数据记录块来记录功能块以及下列存储区域的实际值：

- I (输入)
- Q (输出)
- M (标志)
- AI (模拟量输入)
- AQ (模拟量输出)
- AM (模拟量标志)

关于数字量I/O及存储区域，须以8个位为一组来进行数据记录。例如，I0到I7，Q8到Q15，M16到M23；关于仿真模拟量数据，可以选择记录一个数值，例如AI1、AQ2或是AM1。依据数值类型(数字量或模拟量)，可以对功能块数值以8位或单个数值进行编组。

在数据记录块中，最多可以记录32条数据(模拟量数值或8位数字量数值组)。




仅能通过PC软件来配置数据记录块，无法在控制器上创建、配置或是删除数据记录块。

传输数据记录

配置完数据记录块后，可将电路程序下载到控制器中；可将该数据记录以 .CSV文件格式传输到SD卡中。

Starting the Simulation

Click on the [Tools -> Simulation](#) menu command or the simulation icon  in the [standard toolbar](#) to start simulation.

Please refer to [simulation toolbar](#) for more information on how to run a simulation.

Inputs

The input icon indicates the status. You can switch an input by clicking on an icon. When the input is switched on, the borderline of the icon is red; when the input is switched off, the borderline turns black.



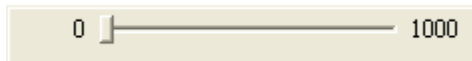
→ The input is actuated.



→ The input is not actuated.

Analog Inputs

You can set the value for an analog input by means of a slide controller. Click on the relevant block to pop up and operate this slide controller directly.



Outputs

The output icon indicates the status. You cannot switch an output by clicking on an icon. When the output is switched on, the borderline of the icon is red; when the output is switched off, the borderline turns black.



→ The output is switched on.



→ The output is switched off.

Set Output

In simulation mode, you can select the command **Set output** by right clicking on the digital output of a block. This command allows you to set the output to high or low, and also allows you to clear the setting. The output remains unchanged until you reset it or end the simulation. This way you can check how a circuit program reacts to certain states.

Power Failure

You can simulate a power failure by clicking on the **Power** icon. This helps you check how a circuit program reacts to a power failure and restart to all inputs. The simulation can also test the retentive values of the circuit program. Note that the power failure simulation is different from the start of simulation, which equals starting the loaded program in 1/5-Series and all of the values are reset.



→ The power supply is normal.



→ Simulate the power supply is interrupted.

118X Address Mapping

Supported Modbus Code: 01/02/05/15 (Readable & Writable in Normal Mode)

Address	Description	R/W	Note
00001 ~ 00032	Main Digital Input Value (I000 ~ I031)	R	(0/1)
00033 ~ 00064	Ext1 Digital Input Value (I100 ~ I131)	R	(0/1)
00065 ~ 00096	Ext2 Digital Input Value (I200 ~ I231)	R	(0/1)
00097 ~ 00128	Ext3 Digital Input Value (I300 ~ I331)	R	(0/1)
00129 ~ 00160	Ext4 Digital Input Value (I400 ~ I431)	R	(0/1)
00161 ~ 00192	Ext5 Digital Input Value (I500 ~ I531)	R	(0/1)
00193 ~ 00224	Ext6 Digital Input Value (I600 ~ I631)	R	(0/1)
00225 ~ 00256	Ext7 Digital Input Value (I700 ~ I731)	R	(0/1)
00257 ~ 00272	Main Digital Output Value (Q000 ~ Q016)	R	(0/1)
00273 ~ 00288	EXT1 Digital Output Value (Q100 ~ Q116)	R	(0/1)
00289 ~ 00304	EXT2 Digital Output Value (Q200 ~ Q216)	R	(0/1)
00305 ~ 00320	EXT3 Digital Output Value (Q300 ~ Q316)	R	(0/1)
00321 ~ 00336	EXT4 Digital Output Value (Q400 ~ Q416)	R	(0/1)
00337 ~ 00352	EXT5 Digital Output Value (Q500 ~ Q516)	R	(0/1)
00353 ~ 00368	EXT6 Digital Output Value (Q600 ~ Q616)	R	(0/1)
00369 ~ 00384	EXT7 Digital Output Value (Q700 ~ Q716)	R	(0/1)
00385 ~ 00896	0~511 Digital Flag (M0 ~ M511)	R	(0/1)
00897 ~ 00912	0~15 Shift register bit (S0 ~ S15)	R	(0/1)
00913	Flag of SCAN Time	R	(0/1)
01025 ~ 01056	Main Digital Input Force ON (I000 ~ I031)	R/W	(0/1)
01057 ~ 01088	Ext1 Digital Input Force ON (I100 ~ I131)	R/W	(0/1)
01089 ~ 01120	Ext2 Digital Input Force ON (I200 ~ I231)	R/W	(0/1)
01121 ~ 01152	Ext3 Digital Input Force ON (I300 ~ I331)	R/W	(0/1)
01153 ~ 01184	Ext4 Digital Input Force ON (I400 ~ I431)	R/W	(0/1)
01185 ~ 01216	Ext5 Digital Input Force ON (I500 ~ I531)	R/W	(0/1)
01217 ~ 01248	Ext6 Digital Input Force ON (I600 ~ I631)	R/W	(0/1)
01249 ~ 01280	Ext7 Digital Input Force ON (I700 ~ I731)	R/W	(0/1)
01281 ~ 01312	Main Digital Input Force OFF(I000 ~ I031)	R/W	(0/1)
01313 ~ 01344	Ext1 Digital Input Force OFF(I100 ~	R/W	(0/1)

	I131)		
01345 ~ 01376	Ext2 Digital Input Force OFF(I200 ~ I231)	R/W	(0/1)
01377 ~ 01408	Ext3 Digital Input Force OFF(I300 ~ I331)	R/W	(0/1)
01409 ~ 01440	Ext4 Digital Input Force OFF(I400 ~ I431)	R/W	(0/1)
01441 ~ 01472	Ext5 Digital Input Force OFF(I500 ~ I531)	R/W	(0/1)
01473 ~ 01504	Ext6 Digital Input Force OFF(I600 ~ I631)	R/W	(0/1)
01505 ~ 01536	Ext7 Digital Input Force OFF(I700 ~ I731)	R/W	(0/1)

Supported Modbus Code: 01/02 (Readable in Normal Mode)

Address	Description	R/W	Note
02001~02004	Status of Function Block B0	R	
02005~02008	Status of Function Block B1	R	
02009~02012	Status of Function Block B2	R	
.....			
06093~06096	Status of Function Block B1023	R	

Supported Modbus Code: 03/04 (Readable in Normal Mode)

Address	Description	R/W	Note
40001	Com0 model	R	0x00 : Slave 0x01 : Master
40002	Com0 protocol	R	0x00 : RTU 0x01 : ASCII
40003	Com0 device address	R	1~255
40004	Com0 baudrate	R	0x00 : 1200 0x01 : 2400 0x02 : 4800 0x03 : 9600 0x04 : 14400 0x05 : 19200 0x06 : 28800 0x07 : 38400 0x08 : 57600 0x09 : 115200
40005	Com0 parity	R	0x00 : None 0x01 : Odd 0x02 : Even
40006	Com0 data bit	R	0x00 : 7-bit 0x01 : 8-bit
40007	Com0 stop bit	R	0x00 : 1-bit 0x01 : 2-bit
40008	Com0 timeout	R	50 ~ 65535 ms
40009	Com0 delay between polls	R	0 ~ 65535 ms
40010	Com0 data register index	R	0x00 : High Low 0x01:Low High
40011	Com0 status flag	R	
40012	Com1 model	R	0x00 : Slave 0x01: Master
40013	Com1 protocol	R	0x00 : RTU 0x01 : ASCII
40014	Com1 device address	R	1~255
40015	Com1 baudrate	R	0x00 : 1200 0x01 : 2400 0x02 : 4800 0x03 : 9600 0x04 : 14400 0x05 : 19200 0x06 : 28800 0x07 : 38400

			0x08 : 57600 0x09 : 115200
40016	Com1 parity	R	0x00 : None 0x01 : Odd 0x02 : Even
40017	Com1 data bit	R	0x00 : 7-bit 0x01 : 8-bit
40018	Com1 stop bit	R	0x00 : 1-bit 0x01 : 2-bit
40019	Com1 timeout	R	50 ~ 65535 ms
40020	Com1 delay between polls	R	0 ~ 65535 ms
40021	Com1 data register index	R	0x00 : High Low 0x01:Low High
40022	Com1 status flag	R	
40023	Com2 model	R	0x00 : Slave 0x01 : Master
40024	Com2 protocol	R	0x00 : RTU 0x01 : ASCII
40025	Com2 device address	R	1~255
40026	Com2 baudrate	R	0x00 : 1200 0x01 : 2400 0x02 : 4800 0x03 : 9600 0x04 : 14400 0x05 : 19200 0x06 : 28800 0x07 : 38400 0x08 : 57600 0x09 : 115200
40027	Com2 parity	R	0x00 : None 0x01 : Odd 0x02 : Even
40028	Com2 data bit	R	0x00 : 7-bit 0x01 : 8-bit
40029	Com2 stop bit	R	0x00 : 1-bit 0x01 : 2-bit
40030	Com2 timeout	R	50 ~ 65535 ms
40031	Com2 delay between polls	R	0 ~ 65535 ms
40032	Com2 data register index	R	0x00 : High Low 0x01:Low High
40033	Com2 status flag	R	

Supported Modbus Code: 03/04 (Readable in Normal Mode)

Address	Description	R/W	Note
40211	Module Name 1	R	118X Ex:0x1188
40212	Module Name 2	R	0x0000
40213	Firmware Version 1	R	A1.00 Ex:0xA100
40214	Firmware Version 2	R	0x0000
40215	Mac Serial Number 1	R	
40216	Mac Serial Number 2	R	
40217	Mac Serial Number 3	R	
40218	Mac Serial Number 4	R	
40219	Mac Serial Number 5	R	
40220	Mac Serial Number 6	R	
40221	Redundancy condition	R	0x00: None 0x01:Master 0x02:Slave
40222	Redundancy operating time (low word) (ms)	R	0x0000 ~ 0xFFFF
40223	Redundancy operating time (high word) (ms)	R	0x0000 ~ 0xFFFF
40224	LCM Control Register	R	

40225	Machine internal tempature (degree Celsius)	R	-32768 ~ 32767
40226	Controller Fault Status	R	
40227	System Status 1	R	
40228	System Status 2	R	
40229	Scan Cycle Time (ms)	R	1 ~ 65535
40230	Redundancy status	R	0x00 : stop 0x01:standby 0x02:active
40231	Power On Hours (hr)	R	0~65535
40232	COM0 communication success rate (times/min)	R	0~65535
40233	COM0 communication error rate (times/min)	R	0~65535
40234	COM1 communication success rate (times/min)	R	0~65535
40235	COM1 communication error rate (times/min)	R	0~65535
40236	COM2 communication success rate (times/min)	R	0~65535
40237	COM2 communication error rate (times/min)	R	0~65535
40238	COM3 communication success rate (times/min)	R	0~65535
40239	COM3 communication error rate (times/min)	R	0~65535
40240	COM4 communication success rate (times/min)	R	0~65535
40241	COM4 communication error rate (times/min)	R	0~65535
40242	COM5 communication success rate (times/min)	R	0~65535
40243	COM5 communication error rate (times/min)	R	0~65535
40244	COM6 communication success rate (times/min)	R	0~65535
40245	COM6 communication error rate (times/min)	R	0~65535
40246	COM7 communication success rate (times/min)	R	0~65535
40247	COM7 communication error rate (times/min)	R	0~65535
40248	Downloading number of times	R	0~65535
40249	History Temperature_min (degree Celsius)	R	-32768 ~ 32767
40250	History Temperature_max (degree Celsius)	R	-32768 ~ 32767
40251	High temperature protection point	R	-32768 ~ 32767
40252	Low temperature protection point	R	-32768 ~ 32767
40253	Power On Count (low word)	R	0x0000 ~ 0xFFFF
40254	Power On Count (high word)	R	0x0000 ~ 0xFFFF

40255	DOWNLOAD_STATUS	R	0x00 : normal 0x01 : fail
40256	Last shutdown time -Week_RTC	R	0 ~ 6
40257	Last shutdown time -Year_RTC	R	2010 ~ 2036
40258	Last shutdown time -Month_RTC	R	1 ~ 12
40259	Last shutdown time -Day_RTC	R	1 ~ 31
40260	Last shutdown time -Hour_RTC	R	0 ~ 23
40261	Last shutdown time -Min_RTC	R	0 ~ 59
40262	Last shutdown time -Sec_RTC	R	0 ~ 59
40263	RTC Calibrate sign	R	0:plus 1:minus
40264	RTC Calibrate value	R	0 ~ 30 (sec/week)

Supported Modbus Code: 03/04 (Readable in Normal Mode)

Address	Description	R/W	Note
40301	Week_RTC	R	0 ~ 6
40302	Year_RTC	R	2010 ~ 2036
40303	Month_RTC	R	1 ~ 12
40304	Day_RTC	R	1 ~ 31
40305	Hour_RTC	R	0 ~ 23
40306	Min_RTC	R	0 ~ 59
40307	Sec_RTC	R	0 ~ 59

Supported Modbus Code: 03/04 (Readable in Normal Mode)

Address	Description	R/W	Note
40501 ~ 40508	Main Analog Input Value (AI000 ~ AI007)	R	
40509 ~ 40516	EXT1 Analog Input Value (AI100 ~ AI107)	R	
40517 ~ 40524	EXT2 Analog Input Value (AI200 ~ AI207)	R	
40525 ~ 40532	EXT3 Analog Input Value (AI300 ~ AI307)	R	
40533 ~ 40540	EXT4 Analog Input Value (AI400 ~ AI407)	R	
40541 ~ 40548	EXT5 Analog Input Value (AI500 ~ AI507)	R	
40549 ~ 40556	EXT6 Analog Input Value (AI600 ~ AI607)	R	
40557 ~ 40564	EXT7 Analog Input Value (AI700 ~ AI707)	R	
40565 ~ 40568	Main Analog Output Value (AQ000 ~ AQ003)	R	
40569 ~ 40572	EXT1 Analog Output Value (AQ100 ~ AQ103)	R	
40573 ~ 40576	EXT2 Analog Output Value (AQ200 ~ AQ203)	R	
40577 ~ 40580	EXT3 Analog Output Value (AQ300 ~ AQ303)	R	
40581 ~ 40584	EXT4 Analog Output Value (AQ400 ~ AQ403)	R	

40585 ~ 40588	EXT5 Analog Output Value (AQ500 ~ AQ503)	R	
40589 ~ 40592	EXT6 Analog Output Value (AQ600 ~ AQ603)	R	
40593 ~ 40596	EXT7 Analog Output Value (AQ700 ~ AQ703)	R	
40597 ~ 41108	0 ~ 511 Analog Flag Value (AM0 ~ AM511)	R	

Supported Modbus Code: 03/04 (Readable in Normal Mode)

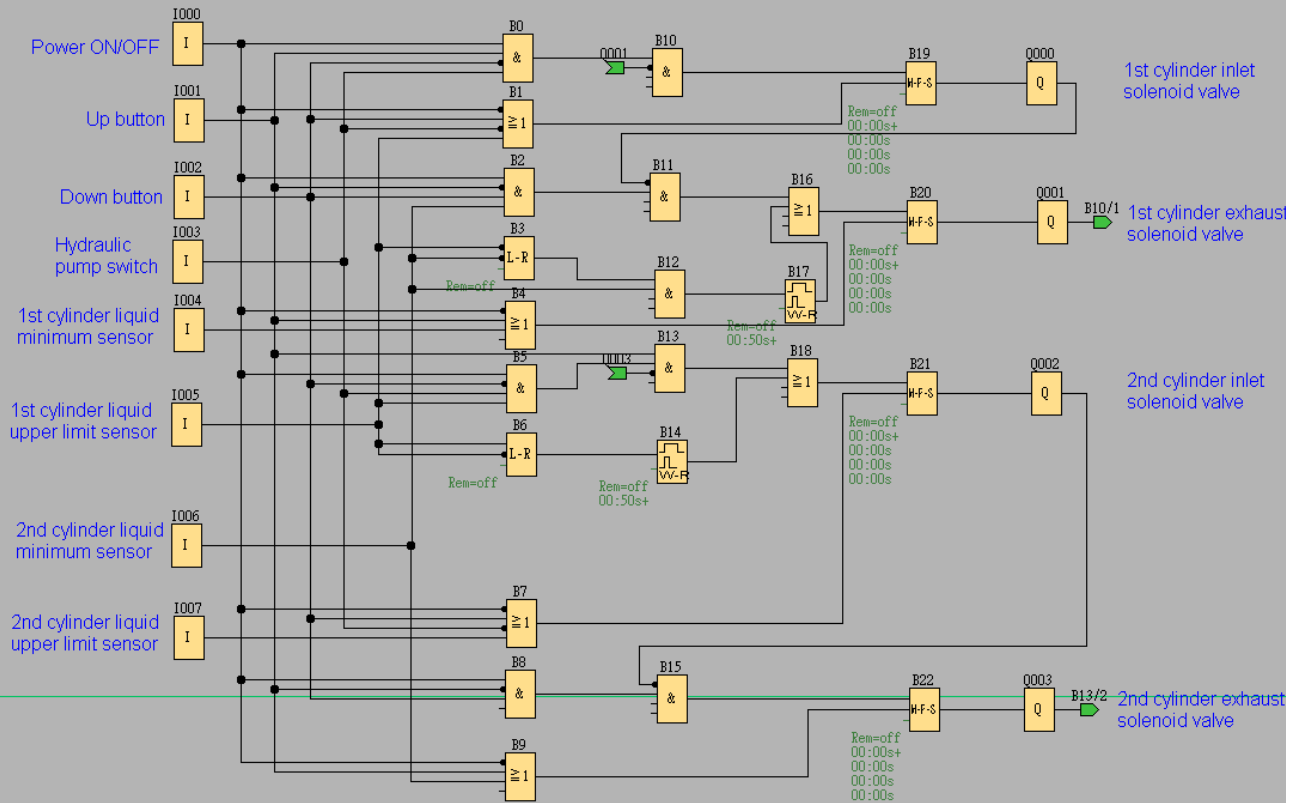
Address	Description	R/W	Note
42001~42004	Parameter of Function Block B0	R	
42005~42008	Parameter of Function Block B1	R	
42009~42012	Parameter of Function Block B2	R	
.....			
46093~46096	Parameter of Function Block B1023	R	

More Information

Block Type	Address 1	Address 2	Address 3	Address 4
AND	Block Output (0xxxx)	X	X	X
AND (Edge)	Block Output (0xxxx)	X	X	X
NAND	Block Output (0xxxx)	X	X	X
NAND (Edge)	Block Output (0xxxx)	X	X	X
OR	Block Output (0xxxx)	X	X	X
NOR	Block Output (0xxxx)	X	X	X
XOR	Block Output (0xxxx)	X	X	X
NOT	Block Output (0xxxx)	X	X	X
On-Delay	Block Output (0xxxx)	X	Timer (4xxxx)	X
Off-Delay	Block Output (0xxxx)	X	Timer (4xxxx)	X
On-/Off-Delay	Block Output (0xxxx)	X	Timer (4xxxx)	X
Retentive On-Delay	Block Output (0xxxx)	X	Timer (4xxxx)	X
Wiping relay (pulse output)	Block Output (0xxxx)	X	Timer (4xxxx)	X
Edge triggered wiping relay	Block Output (0xxxx)	X	Timer (4xxxx)	X
Asynchronous Pulse Generator	Block Output (0xxxx)	X	Timer (4xxxx)	X
Random Generator	Block Output			

	(0xxxx)	X	Timer (4xxxx)	X
Stairway lighting switch	Block Output (0xxxx)	X	Timer (4xxxx)	X
Multiple function switch	Block Output (0xxxx)	X	Timer (4xxxx)	X
Weekly Timer	Block Output (0xxxx)	X	X	X
Yearly Timer	Block Output (0xxxx)	X	X	X
Up/Down counter	Block Output (0xxxx)	X	Count Value (l) (4xxxx)	Count Value (h) (4xxxx)
Hours Counter	Block Output (0xxxx)	X	MN Value (l) (4xxxx)	MN Value (h) (4xxxx)
Threshold trigger	Block Output (0xxxx)	X	Count Value (l) (4xxxx)	Count Value (h) (4xxxx)
Analog Comparator	Block Output (0xxxx)	X	Actual values(Ax-Ay) (l) (4xxxx)	Actual values(Ax-Ay) (h) (4xxxx)
Analog threshold trigger	Block Output (0xxxx)	X	Actual value Ax (l) (4xxxx)	Actual value Ax (h) (4xxxx)
Analog Amplifier	Block Output (4xxxx)	X	Actual value Ax (l) (4xxxx)	Actual value Ax (h) (4xxxx)
Analog watchdog	Block Output (0xxxx)	Actual value Aen (4xxxx)	Actual value Ax (l) (4xxxx)	Actual value Ax (h) (4xxxx)
Analog differential trigger	Block Output (0xxxx)	X	Actual value Ax (l) (4xxxx)	Actual value Ax (h) (4xxxx)
Latching Relay	Block Output (0xxxx)	X	X	X
Pulse Relay	Block Output (0xxxx)	X	X	X
Message texts	Block Output (0xxxx)	X	X	X
Shift register	Block Output (0xxxx)	X	Register Value (4xxxx)	X
Modbus Read	Block Output (0xxxx)	Count (4xxxx)	Data Address (4xxxx)	X
Modbus Write	Block Output (0xxxx)	Count (4xxxx)	Data1(Manual) / Data Address (Auto) (4xxxx)	Data2(Manual) (4xxxx)

Hydraulic cylinder equipment



Hydraulic cylinder equipment

The hydraulic cylinder equipment control system is applied to buildings, parking tower, warehouse etc. This demonstration can control cylinder platform up and down.

Equipment structure:

- . 2 cylinders (top & bottom) to drive the platform.
- . 4 solenoid valves to control cylinders inlet or exhaust.
- . 1 pressure pump to support cylinders liquid.
- . 1 pressure tank for storage and support cylinders liquid.
- . 4 sensors to monitor cylinders position signal.

Blocks describe:

- I000: Power ON/OFF.
- I001: Up button to control cylinder up and up-stop.
- I002: Down button to control cylinder down and down-stop.
- I003: Hydraulic pump switch to control pressure pump on/off
- I004: 1st cylinder (bottom) liquid minimum sensor to monitor bottom cylinder minimum liquid position.
- I005: 1st cylinder (bottom) liquid upper limit sensor to monitor bottom cylinder upper liquid position
- I006: 2nd cylinder (top) liquid minimum sensor to monitor top cylinder minimum liquid position
- I007: 2nd cylinder (top) liquid upper limit sensor to monitor bottom cylinder upper liquid position
- Q000: 1st cylinder inlet solenoid valve to control liquid inflow to 1st cylinder
- Q001: 1st cylinder exhaust solenoid valve to control liquid outflow from 1st cylinder
- Q002: 2nd cylinder inlet solenoid valve to control liquid inflow to 2nd cylinder
- Q003: 2nd cylinder exhaust solenoid valve to control liquid outflow from 2nd cylinder

Process describe:

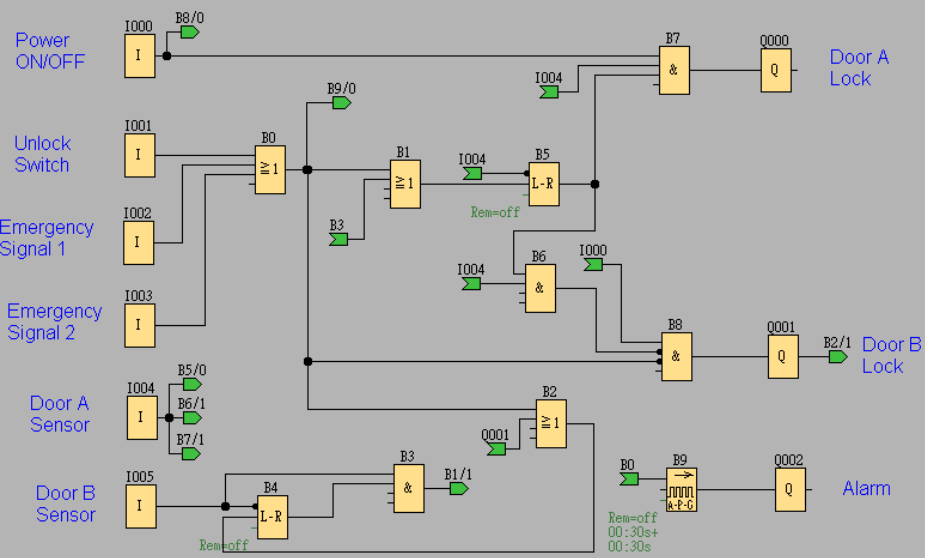
Cylinder up:

1. Before drive 1st and 2nd cylinders platform up, turn on the power (I000) and Hydraulic pump switch (I003), then I004 and I006 will on start position.
2. Push the up button (I001), the solenoid valve (Q000) will control liquid inflow to 1st cylinder and elevate platform then I004 auto-off.
3. When 1st cylinder liquid on upper limit position (I005), Q000 auto-off then solenoid valve (Q002) will auto-on to control liquid inflow 2nd cylinder and elevate platform then I006 auto-off.
4. When 2nd cylinder liquid on upper limit position (I007), Q002 auto-off.

Cylinder down:

1. Push the down button (I002), the solenoid valve (Q003) will control liquid outflow from 2nd cylinder and degrade platform then I007 auto-off.
2. When 2nd cylinder liquid on minimum position (I006), Q003 auto-off then solenoid valve (Q001) will auto-on to control liquid outflow from 1st cylinder and degrade platform then I005 auto-off.
3. When 1st cylinder on minimum position (I004), Q001 auto-off.

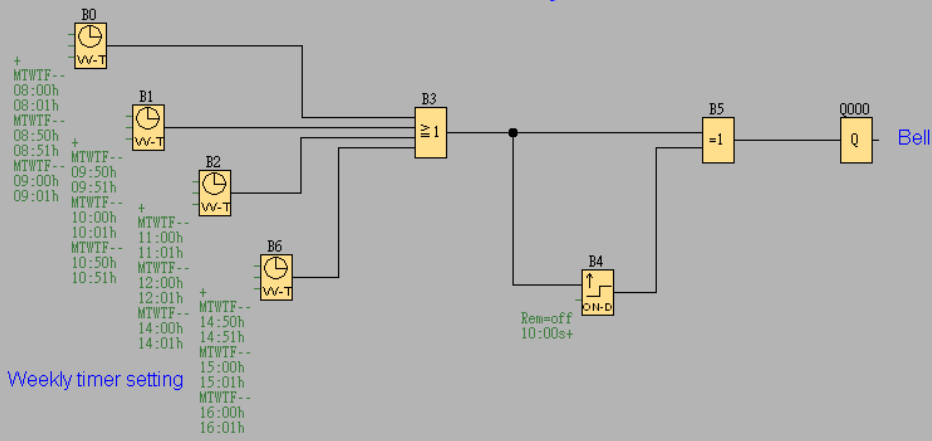
One-way interlocking door control system



Process describing:

1. Before system working (power on), Door A (Q000) and Door B (Q001) can open momentarily. Door A (I004) and Door B (I005) sensors on On status.
2. Turn on the power (I000), Door B lock (Q001).
3. Open Door A, the sensor (I004) auto-off. Then close Door A, the sensor (I004) auto-on and Door A lock (Q000) Door B open (Q001).
4. Open Door B, the sensor (I005) auto-off. Then close Door B, the sensor (I005) auto-on and Door B lock (Q001) Door A open (Q000).
5. Press the unlock switch (I001) or emergency signal (I002, I003) switch on, Door A (Q000) and Door B (Q001) can open momentarily.

School bell system

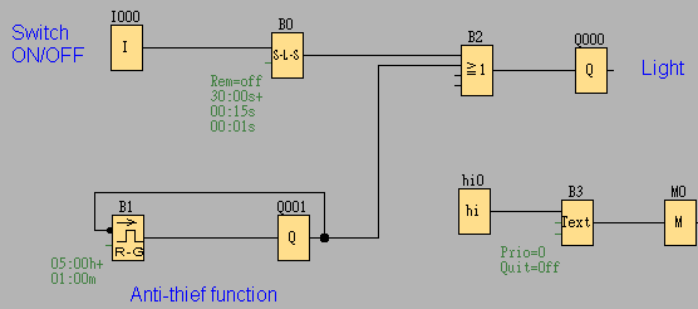


Process describing:

When school begins, break and end, the bell will ring 10 seconds on setting time.

Bell time is set on Monday to Friday at 8:00, 8:50, 9:00, 9:50, 10:00, 10:50, 11:00, 12:00, 14:00, 14:50, 15:00 and 16:00.

Stairway lighting switch and anti-thief light system



Process describing:

1. Turn the switch (I001) on, light (Q000) will on. Turn off the switch, light will off after 30 seconds.
2. Anti-thief function wills random work per 5 hours.
3. When switch-on time overlap with anti-thief function, light-off time decide by latest end-time.
4. B3 can display switch off count time on A series controller monitor.