

# 简明A5系列PLC操作 手册

上海正航电子科技有限公司  
Zhenghang Electronics Co., LTD

版本号：VC1

# 1. 通用说明

## 1.1. 型号与规格

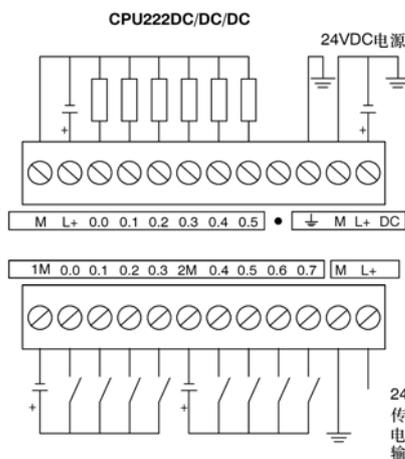
A5系列PLC是高品质的小型一体化可编程控制器，包括CPU模块和数字量输入输出、模拟量输入输出等各种类型的扩展模块。一个A5系列PLC控制系统必须有一个CPU模块，可以有0~7个各种类型的扩展模块。

下表为各种CPU的型号及简要介绍，其具体型号及其配备的系统资源参见硬件说明。

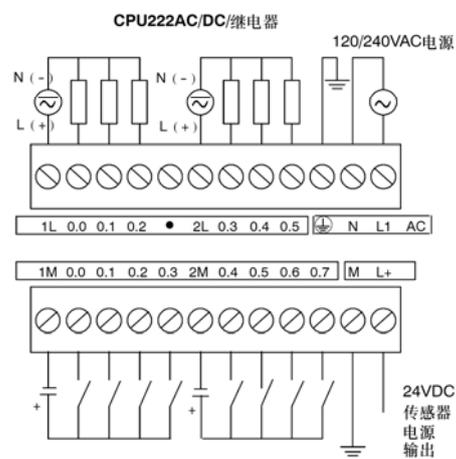
模块类型	新型号	旧型号	说明	电源	输入类型	输出类型
CPU 模块	A5-CPU221AR	A5-C1007AR	10 点 CPU 单元，6 入 4 出，可带扩展模块	220V 交流	24V 直流	继电器
	A5-CPU221DT	A5-C1007DT		24V 直流	24V 直流	晶体管
	A5-CPU222AR	A5-C1407AR	14 点 CPU 单元，8 入 6 出，可带扩展模块	220V 交流	24V 直流	继电器
	A5-CPU222DT	A5-C1407DT		24V 直流	24V 直流	晶体管
	A5-CPU224AR	A5-C2407AR	24 点 CPU 单元，14 入 10 出，可带扩展模块	220V 交流	24V 直流	继电器
	A5-CPU224DT	A5-C2407DT		24V 直流	24V 直流	晶体管
	A5-CPU226AR	A5-C4007AR	40 点 CPU 单元，24 入 16 出，可带扩展模块	220V 交流	24V 直流	继电器
	A5-CPU226DT	A5-C4007DT		24V 直流	24V 直流	晶体管
	A5-CPU224XPAR		24 点 CPU 单元，并有 2 路模拟量输入、1 路模拟量输出。可带扩展模块	220V 交流	24V 直流	继电器
	A5-CPU224XPDT			24V 直流	24V 直流	晶体管

CPU的型号列表

## 1.2. 接线图

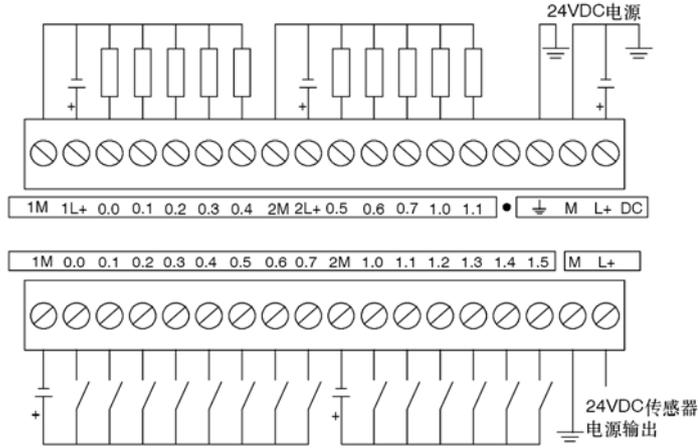


A5-CPU222-DT 接线图



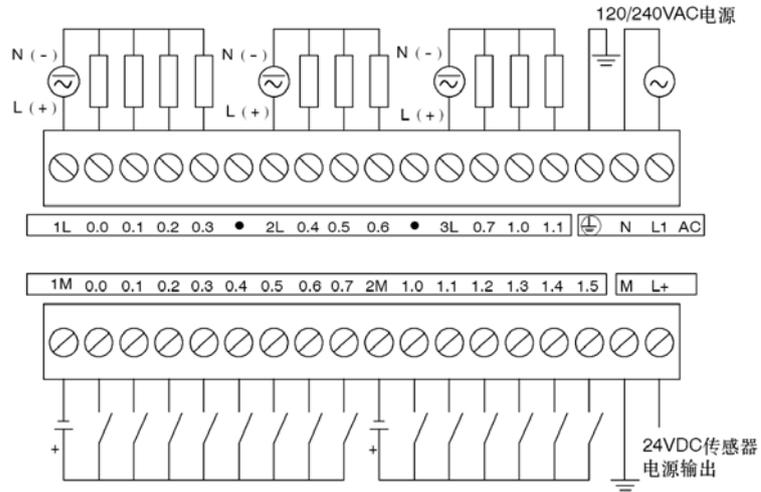
A5-CPU222-AR 接线图

CPU224DC/DC/DC



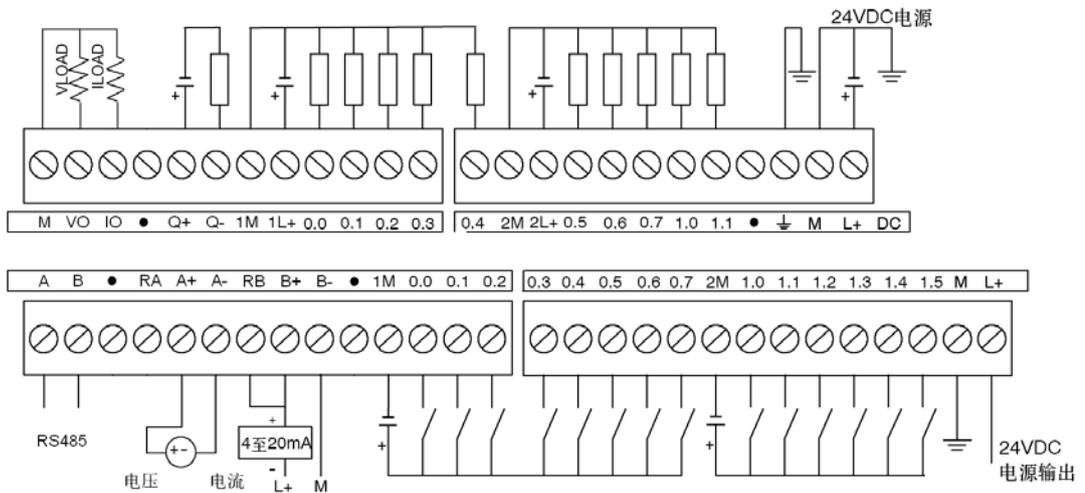
A5-CPU224-DT 接线图

CPU224AC/DC/继电器

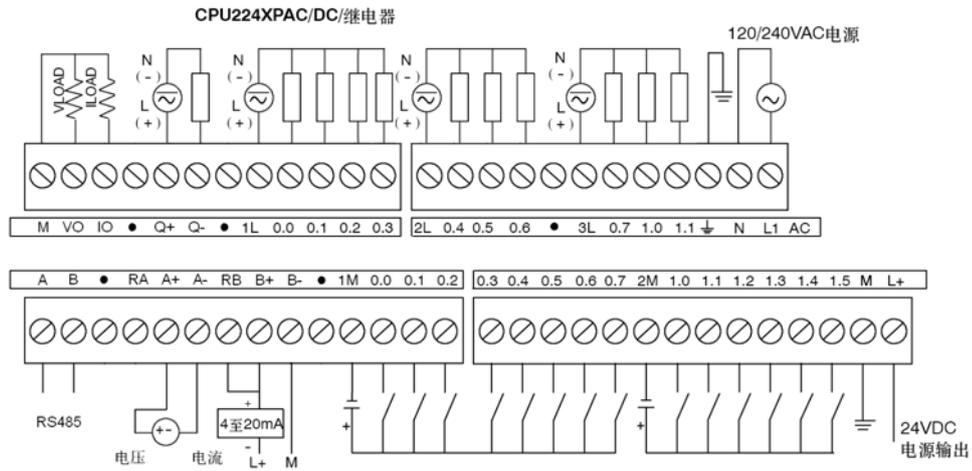


A5-CPU224-AR 接线图

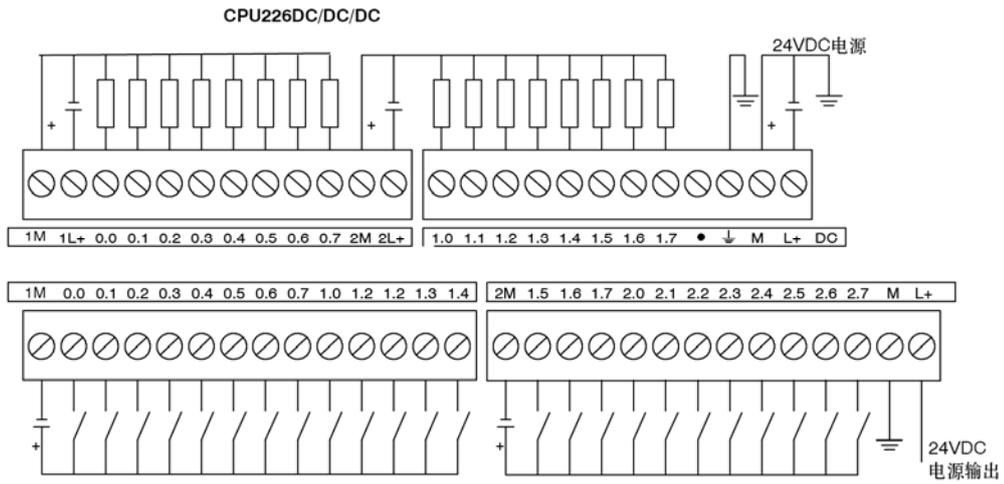
CPU224XPDC/DC/DC



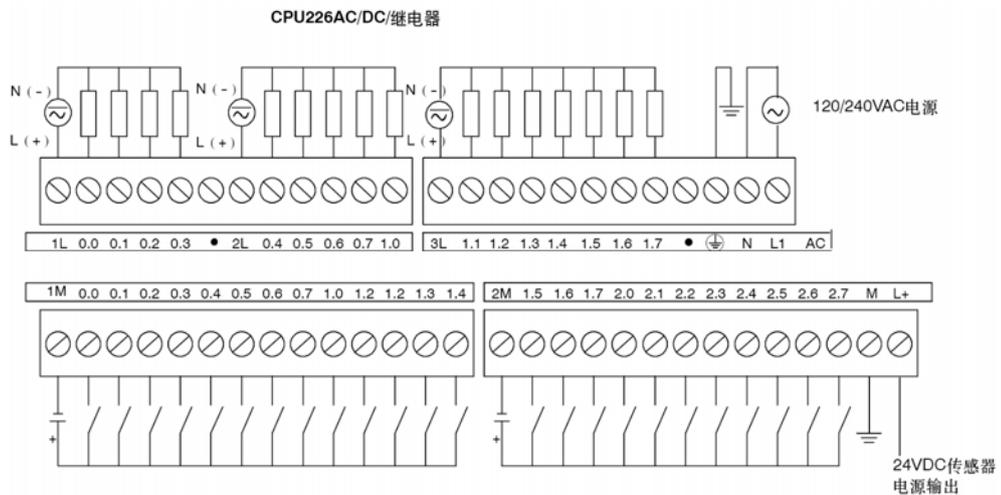
A5-CPU224XP-DT 接线图



A5-CPU224XP-AR 接线图



A5-CPU226-DT 接线图



A5-CPU226-AR接线图

### 1.3. 存储器

描述	A5-C1007	A5-C1407	A5-C2407	A5-C2407加强型	A5-C4007
用户程序大小	4096字节	4096字节	12288字节	16384字节	24576字节
用户数据大小	2048字节	2048字节	8192字节	10240字节	10240字节
输入映像寄存器	I0.0 - I15.7				
输出映像寄存器	Q0.0 - Q15.7				
模拟量输入（只读）	AIW0 - AIW15				
模拟量输出（只写）	AQW0 - AQW15				
变量存储器（V）	VB0 - VB2499				
局部存储器（L）	LB0 - LB3				
位存储器（M）	M0.0 - M31.7				
特殊存储器（SM） 只读	SM0.0 - SM179.7 SM0.0 - SM29.7				
定时器	64(T0-T63)	64(T0-T63)	64(T0-T63)	64(T0-T63)	64(T0-T63)
1ms	T0, T1				
10ms	T2 - T19				
100ms	T20 - T63				
计数器	C0 - C31				
高速计数器	HC0, HC1				
顺序控制继电器（S）	S0.0 - S15.7				
累加寄存器	AC0	AC0	AC0	AC0	AC0
跳转/标号	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
调用/子程序	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
中断程序	0 - 127	0 - 127	0 - 127	0 - 127	0 - 127
正/负跳变	256	256	256	256	256
端口	端口0	端口0	端口0	端口0, 1	端口0, 1

## 2.A5 常用指令

### 2.1. 位逻辑指令说明

位操作指令是PLC常用的基本指令，梯形图指令有触点和线圈两大类，触点又分常开触点和常闭触点两种形式；语句表指令有与、或以及输出等逻辑关系，位操作指令能够实现基本的位逻辑运算和控制。

#### 2.1.1. 常开触点 LD、常开触点与 A、常开触点或 O

指令大类	位逻辑指令		
指令简介	常开触点	常开触点与	常开触点或
梯形图提示符	-  -		
梯形图			
语句表	LD	A	O
参数个数	1		
参数允许值	I, Q, M, SM, T, C, V, S, L		
参数类型	布尔型（位）		
指令功能	读取输入位值在梯形图模式下，如果位值为1，则此触点闭合，能流导通，否则触点断开；在语句模式下，将读取值作为堆栈顶值	读取输入位值，并与堆栈顶值相与，得到新的堆栈顶值	读取输入位值，并与堆栈顶值相或，得到新的堆栈顶值

LD (load)：常开触点逻辑运算的开始。对应梯形图则为在左侧母线或线路分支点处初始装载一个常开触点。

A(And)：与操作，在梯形图中表示串联连接单个常开触点。

O：或操作，在梯形图中表示并联连接一个常开触点。

#### 2.1.2. 常闭触点 LDN、常开触点与 AN、常开触点或 ON

指令大类	位逻辑指令		
指令简介	常闭触点	常闭触点与	常闭触点或
梯形图提示符	- / -		

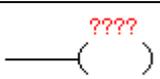
梯形图			
语句表	LDN	AN	ON
参数个数	1		
参数允许值	I, Q, M, SM, T, C, V, S, L		
参数类型	布尔型 (位)		
指令功能	读取输入位值在梯形图模式下, 如果位值为0, 则此触点闭合, 能流导通, 否则触点断开; 在语句模式下, 将读取值取反作为堆栈顶值	读取输入位值, 取反后与堆栈顶值相与, 得到新的堆栈顶值	读取输入位值, 取反后与堆栈顶值相或, 得到新的堆栈顶值

LDN (load not)：常闭触点逻辑运算的开始（即对操作数的状态取反），对应梯形图则为在左侧母线或线路分支点处初始装载一个常闭触点。

AN(And not)：与非操作，在梯形图中表示串联连接单个常闭触点。

ON：或非操作，在梯形图中表示并联连接一个常闭触点。

### 2.1.3. 输出指令=

指令大类	位逻辑指令
指令简介	输出
梯形图提示符	-()
梯形图	
语句表	=
参数个数	1
参数的允许值	I, Q, M, SM, T, C, V, S, L
参数类型	布尔型 (位)
指令说明	输出, 根据能流状态输出到线圈

= (OUT)：输出指令，对应梯形图则为线圈驱动。一般对同一元件只使用一次。

注意 “=” 可以并联使用任意次，但不能串联。如下图所示。

### 梯形图

### 语句表

	<pre>LD I0.0 = M0.0 = Q0.0</pre>
--	----------------------------------

输出指令可以并联使用

#### 2.1.4. 取反指令 NOT

指令大类	位逻辑指令
指令简介	取反
梯形图提示符	- NOT -
梯形图	
语句表	NOT
参数个数	无参数
指令说明	将能流（栈顶值）取反，如果前面没有能流输入，则输出能流，否则不输出能流

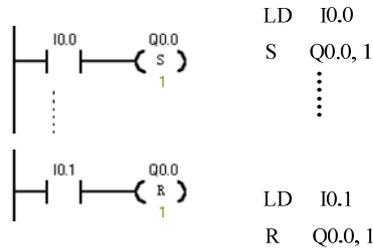
#### 2.1.5. 置位指令 S、复位指令 R

指令大类	位逻辑指令	位逻辑指令
指令简介	置位	复位
梯形图提示符	-(S)	-( R )
梯形图		
语句表	S	R
参数个数	2	
参数1的允许值	I, Q, M, SM, T, C, V, S, L	
参数1类型	布尔型（位）	布尔型（位）
参数2的允许值	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD	
参数2类型	字节	
指令说明	如果能流为1，则将线圈位位置1	如果能流为1，则将线圈位位置0

置位指令S：使能输入有效后从起始位S-bit开始的N 个位置“1”并保持。

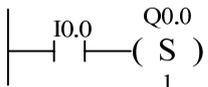
复位指令R：使能输入有效后从起始位S-bit开始的N 个位清“0”并保持。

指令用法如下图所示：



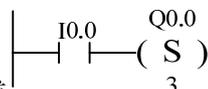
S/R指令的使用

例如：

第一句为 ，意思是当I0.0接通（即为1），输出Q0.0开始的1个位（即Q0.0）被置“1”。

当“S”下面的“1”就是表示从Q0.0开始的1个位即Q0.0；

若“S”下面为“3”则表示从Q0.0开始的3个位即Q0.0、Q0.1、Q0.2这三个

个，而第一句就变成  意思就变成当I0.0接通（即为1），输出Q0.0开始的3个位，即Q0.0、Q0.1、Q0.2三个全部置“1。”

对同一元件(同一寄存器的位)可以多次使用S/R指令(与“=”指令不同)。

由于是扫描工作方式，当置位、复位指令同时有效时，写在后面的指令具有优先权。

操作数N为：VB, IB, QB, MB, SMB, SB, LB, AC, 常量, \*VD, \*AC, \*LD。取值范围为：0~255。数据类型为：字节。

置位复位指令通常成对使用，也可以单独使用或与指令盒配合使用。

### 2.1.6. 脉冲生成指令 EU/ED

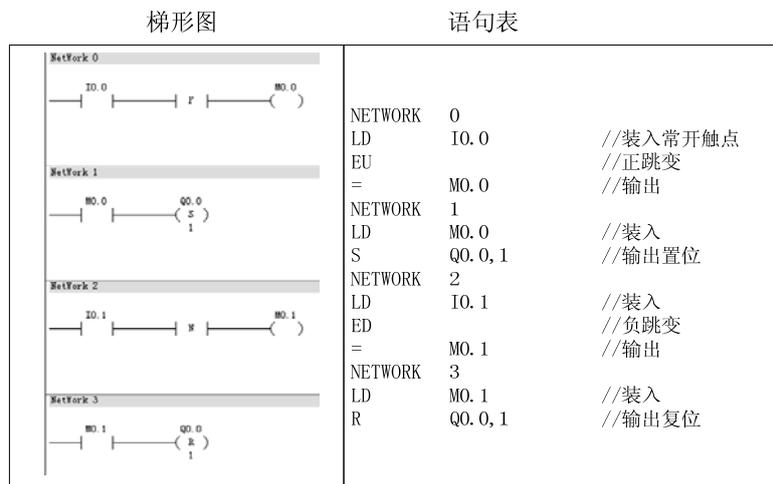
指令大类	位逻辑指令	位逻辑指令
指令简介	正跳变	负跳变
梯形图提示符	EU	ED

梯形图		
语句表	EU	ED
指令说明	正跳变，如果扫描到输入能流的上升沿，即上次能流输入为0，本次扫描输入为1，则输出能流为1，其余情况输出能流为0	负跳变，如果扫描到输入能流的下降沿，即上次能流输入为1，本次扫描输入为0，则输出能流为1，其余情况输出能流为0

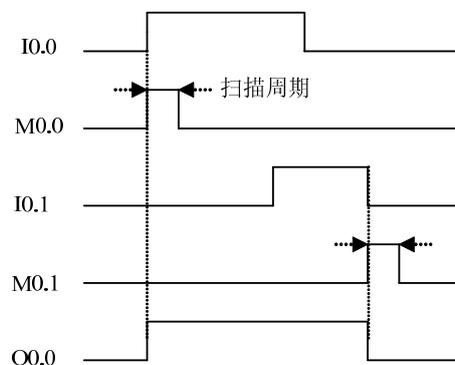
EU指令：在EU指令前的逻辑运算结果有一个上升沿时（由0→1）产生一个宽度为一个扫描周期的脉冲，驱动后面的输出线圈。

ED指令：在ED指令前有一个下降沿时产生一个宽度为一个扫描周期的脉冲，驱动其后线圈。

指令用法、时序分析如下图所示。



### EU/ED指令的使用



### EU/ED指令时序分析

程序及运行结果分析如下：

I0.0的上升沿，经触点（EU）产生一个扫描周期的时钟脉冲，驱动输出线圈M0.0导通一个扫描周期，M0.0的常开触点闭合一个扫描周期，使输出线圈Q0.0置位为1，并保持。

I0.1的下降沿，经触点（ED）产生一个扫描周期的时钟脉冲，驱动输出线圈M0.1导通一个扫描周期，M0.1的常开触点闭合一个扫描周期，使输出线圈Q0.0复位为0，并保持。

EU、ED指令只在输入信号变化时有效，其输出信号的脉冲宽度为一个机器扫描周期。

EU、ED指令无操作数。

## 2.2. 定时器指令

### 2.2.1. 定时器指令 TON、TOFF、TONR 说明

指令大类	定时器指令		
指令简介	接通延时	断开延时	有记忆接通延时
梯形图			
语句表	TON	TOFF	TONR
参数个数	2		
参数1的允许值	T		
参数1类型	字节		
参数2的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, 常数, *VD, *AC, *LD		
参数2类型	整数（带符号双字节）		
指令说明	接通延时定时器，如果输入能流连续导通，经过操作数2指定的时间，操作数1指定的定时器位接通。每次从能流断开到接通的上升沿开始，定时器开始从0计时，直到最大值32767，如果定时器值大于等于操作数2指定的时间后定时器位接通。如	断开延时定时器，如果输入能流连续断开，经过操作数2指定的时间，操作数1指定的定时器位断开。每次从能流接通到断开的下降沿开始，定时器开始从0计时，直到等于操作数2的值，之后保持不变并断开定时器位。如果扫描到能流接通，定时器	有记忆接通延时定时器，如果输入能流累计的导通时间超过操作数2指定的时间，操作数1指定的定时器位接通。每次从能流断开到接通的上升沿开始，定时器继续计时，直到最大值32767，如果定时器值大于等于操作数2指定的时间后定时器位接

	果扫描到能流断开，定时器值清0，定时器位值清0。定时器T0、T1的分辨率为1ms，T2到T19分辨率为10ms，其余定时器分辨率为100ms。	值清0，定时器位置1。定时器T0、T1的分辨率为1ms，T2到T19分辨率为10ms，其余定时器分辨率为100ms。	通。如果扫描到能流断开，定时器停止计时，定时器值和定时器位值保持不变。定时器T0、T1的分辨率为1ms，T2到T19分辨率为10ms，其余定时器分辨率为100ms。
--	---	--	--

### 2.2.2. 时基

按时基脉冲分，则有1ms、10ms、100ms 三种定时器。不同的时基标准，定时精度、定时范围和定时器刷新的方式不同。

#### (1) 定时精度和定时范围

定时器的工作原理是：使能输入有效后，当前值PT对PLC内部的时基脉冲增1计数，当计数值大于或等于定时器的预置值后，状态位置1。其中，最小计时单位为时基脉冲的宽度，又为定时精度；从定时器输入有效，到状态位输出有效，经过的时间为定时时间，即：定时时间=预置值×时基。当前值寄存器为16bit，最大计数值为32767，由此可推算不同分辨率的定时器的设定时间范围。A5系列PLC的64个定时器均可用于TON（TOF）和TONR工作方式，分别有3种时基标准，如下表所示。可见时基越大，定时时间越长，但精度越差。

工作方式	时基	最大定时范围	定时器号
TON/TOF/ TONR	1 ms	32.767 s	T0, T1
	10 ms	327.67 s	T2-19
	100 ms	3276.7 s	T20-T63

定时器的类型

#### (2) 1ms 10ms 100ms定时器的刷新方式不同

1ms定时器每隔1ms刷新一次与扫描周期和程序处理无关即采用中断刷新方式。因此当扫描周期较长时，在一个周期内可能被多次刷新，其当前值在一个扫描周期内不一定保持一致。

10ms 定时器则由系统在每个扫描周期开始自动刷新。由于每个扫描周期内只刷新一次，故而每次程序处理期间，其当前值为常数。

100ms定时器则在该定时器指令执行时刷新。下一条执行的指令，即可使用

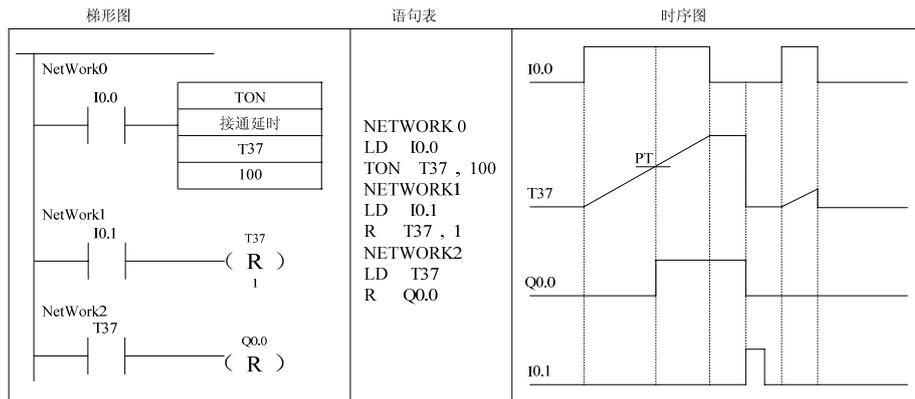
刷新后的结果，非常符合正常的思路，使用方便可靠。但应当注意，如果该定时器的指令不是每个周期都执行，定时器就不能及时刷新，可能导致出错。

### 2.2.3. 定时器指令工作原理

下面我们将从原理应用等方面分别叙述通电延时型，有记忆的通电延时型，断电延时型三种定时器的使用方法。

#### (1) 通电延时定时器（TON）指令工作原理

程序及时序分析如下图所示。当I0.0接通时即使能端（IN）输入有效时，驱动T37（时基为100ms）开始计时，当前值从0开始递增，计时到设定值PT（100\*100ms）时，T37 状态位置1，其常开触点T37接通，驱动Q0.0输出，其后当前值仍增加，但不影响状态位。当前值的最大值为32767。当I0.0分断时，使能端无效时，T37复位，当前值清0，状态位也清0，即回复原始状态。若I0.0接通时间未到设定值就断开，T37则立即复位，Q0.0不会有输出。



#### 通电延时定时器（TON）

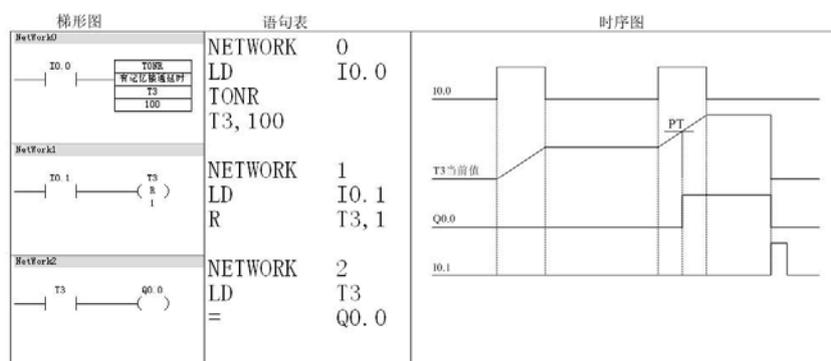
#### (2) 记忆型通电延时定时器（TONR）指令工作原理

使能端（IN）输入有效时（接通），定时器开始计时，当前值递增，当前值大于或等于预置值（PT）时，输出状态位置1。使能端输入无效（断开）时，当前值保持（记忆），使能端（IN）再次接通有效时，在原记忆值的基础上递增计时。

注意：TONR记忆型通电延时型定时器采用线圈复位指令R进行复位操作，当复位线圈有效时，定时器当前位清零，输出状态位置0。

程序分析如下图所示。如T3（时基为10ms），当输入IN（I0.0）为1时，定

时器计时；当IN（I0.0）为0时，其当前值保持并不复位；下次IN（I0.0）再为1时，T3当前值从原保持值开始往上加，将当前值与设定值PT（100\*10ms）比较，当前值大于等于设定值时，T3状态位置1，驱动Q0.0有输出，以后即使IN再为0，也不会使T3复位，要使T3复位，必须使用复位指令。

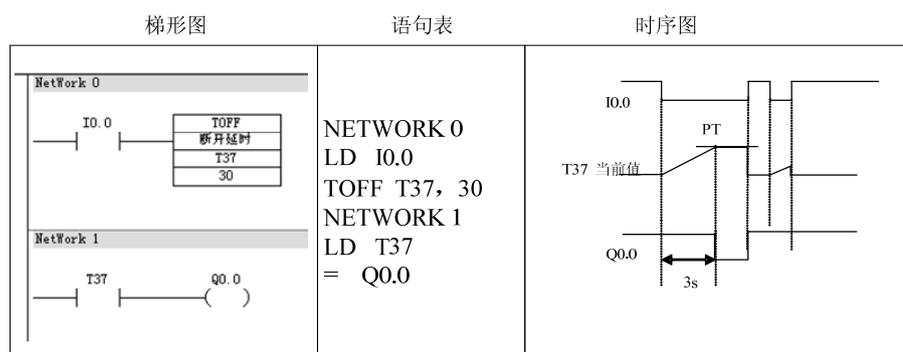


TONR记忆型通电延时型定时器

### (3) 断电延时型定时器（TOFF）指令工作原理

断电延时型定时器用来在输入断开，延时一段时间后，才断开输出。使能端（IN）输入有效时，定时器输出状态立即置1，当前值复位为0。使能端（IN）断开时，定时器开始计时，当前值从0递增，当前值达到预置值时，定时器状态位复位为0，并停止计时，当前值保持。

如果输入断开的时间，小于预定时间，定时器仍保持接通。IN再接通时，定时器当前值仍设为0。断电延时定时器的应用程序及时序分析如图4-43所示。



TOFF断电延时定时器的工作原理

## 2.3. 计数器指令

### 2.3.1. 计数器指令 CTU、CTD、CTUD

计数器利用输入脉冲上升沿累计脉冲个数。结构主要由一个16位的预置值寄存器、一个16位的当前值寄存器和一位状态位组成。当前值寄存器用以累计脉冲个数，计数器当前值大于或等于预置值时，状态位置1。

A5系列PLC有三类计数器：CTU-加计数器，CTUD-加/减计数器，CTD-减计数。

指令大类	计数器指令	计数器指令	计数器指令
指令简介	增计数器	减计数器	增减计数器
梯形图			
语句表	CTU	CTD	CTUD
参数个数	2		
参数1的允许值	C		
参数1类型	字节		
参数2的允许值	VW, IW, AIW, QW, MW, SMW, LW, AC, T, C, 常数, *VD, *AC, *LD, SW		
参数2类型	整数（带符号双字节）		
指令说明	<p>增计数器，每次扫描到CU输入的能流由低变高的上升沿，计数器值加1，直到最大值32767。如果计数器值超过操作数2指定的预置值，计数器位置1。如果扫描到复位能流有输入，则计数器值清0</p>	<p>减计数器，每次扫描到CD输入的能流由低变高的上升沿，计数器值减1，直到0。如果计数器值大于0，计数器位为0，如果计数器值等于0，计数器位置1。如果扫描到读取能流有输入，则计数器值等于操作数2预置值</p>	<p>增减计数器，每次扫描到CU输入的能流由低变高的上升沿，计数器值加1，直到最大值32767后，如果再次输入，变为-32768，每次扫描到CD输入的能流由低变高的上升沿，计数器值减1，直到-32768，如果再次输入，变为32767。当复位能流输入为1时，计数器值为0。当计数器值大于等于预置值时，计数器位为1，否则为0</p>

## 2.3.2. 计数器工作原理分析

### 2.3.2.1. 加计数器指令（CTU）

当R=0时，计数脉冲有效；当CU端有上升沿输入时，计数器当前值加1。当计数器当前值大于或等于设定值（PV）时，该计数器的状态位C-bit置1，即其常开触点闭合。计数器仍计数，但不影响计数器的状态位。直至计数达到最大值（32767）。当R=1时，计数器复位，即当前值清零，状态位C-bit也清零。加计数器计数范围：0~32767。

### 2.3.2.2. 加/减计数指令（CTUD）

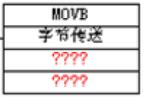
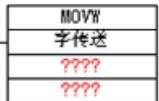
当R=0时，计数脉冲有效；当CU端（CD端）有上升沿输入时，计数器当前值加1（减1）。当计数器当前值大于或等于设定值时，C-bit置1，即其常开触点闭合。当R=1时，计数器复位，即当前值清零，C-bit也清零。加减计数器计数范围：-32768~32767。

### 2.3.2.3. 减计数指令（CTD）

当复位LD有效时，LD=1，计数器把设定值（PV）装入当前值存储器，计数器状态位复位（置0）。当LD=0，即计数脉冲有效时，开始计数，CD端每来一个输入脉冲上升沿，减计数的当前值从设定值开始递减计数，当前值等于0时，计数器状态位置位（置1），停止计数。

## 2.4. 数据传送指令

数据传送指令MOVB、MOVW、MOVD、MOVR，用来传送单个的字节、字、双字、实数。

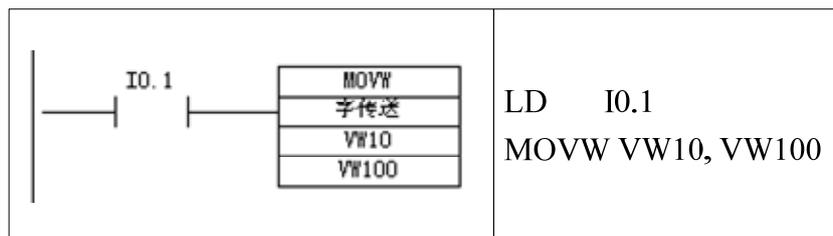
指令大类	传送指令			
指令简介	字节传送	字传送	双字传送	实数传送
梯形图提示符	MOVB	MOVW	MOVD	MOVR
梯形图				
语句表	MOVB	MOVW	MOVD	MOVR
参数个数	2			

参数1的允许值	VB, IB, QB, MB, SB, SM B, LB, AC, *VD, *AC, * LD, 常数	VW, IW, AIW, QW, MW , SW, SMW, LW, T, C, AC, *VD, *AC, *LD, 常数	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, & IB, &QB, &MB, &SB, &T, &C, AC, *VD, *A C, *LD, 常数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, * AC, *LD, 常数
参数1类型	字节	整数（带符号双字 节）	双整数（有符号4 字节）	实数（4字节浮点 数）
参数2的允许值	VB, IB, QB, MB, SB, SM B, LB, AC, *VD, *AC, * LD	VW, T, C, IW, AIW, Q W, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, * AC, *LD	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, * AC, *LD
参数2类型	字节	整数（带符号双字 节）	双整数（有符号4 字节）	实数（4字节浮点 数）
指令说明	使能输入有效时，将操作数1指定的值不进行改变复制传送到操作数2的地址			

【例】如果输入I0.1为高电平，将变量存储器VW10中内容送到VW100中。  
程序如下图所示。

梯形图

语句表



传送指令应用举例

## 3. 附录

### 3.1. 附录 1——SMB0 位定义

SM0.0	该位始终为1
SM0.1	该位在首次扫描时为1，用途之一是调用初始化子程序
SM0.2	若保持数据丢失，则该位在一个扫描周期中为1。该位可用作错误存储器位，或用来调用特殊启动顺序功能。
SM0.3	开机后进入RUN方式，该位将ON一个扫描周期，该位可用作在启动操作之前给设备提供一个预热时间
SM0.4	该位提供了一个时钟脉冲，30秒为1，30秒为0，周期为一分钟，它提供了一个简单易用的延时或1分钟的时钟脉冲
SM0.5	该位提供了一个时钟脉冲，0.5秒为1，0.5秒为0，周期为1秒钟。它提供了一个简单易用的延时或1秒钟的时钟脉冲
SM0.6	该位为扫描时钟，本次扫描时置1，下次扫描时置0。可用作扫描计数器的输入
SM0.7	该位指示CPU工作方式开关的位置（0为TERM位置，1为RUN位置）。当开关在RUN位置时，用该位可使自由端口通信方式有效，那么当切换至TERM位置时，同编程设备的正常通讯也会有效。

### 3.2. 附录 2——SMB1 位定义

SM1.0	当执行某些指令，其结果为0时，将该位置1。
SM1.1	当执行某些指令，其结果溢出或查出非法数值时，将该位置1。
SM1.2	当执行数学运算，其结果为负数时，将该位置1。
SM1.3	试图除以零时，将该位置1。
SM1.4	当执行ATT指令时，试图超出表范围时，将该位置1。
SM1.5	当执行LIFO或FIFO指令，试图从空表中读数时，将该位置1。
SM1.6	当试图把一个非BCD数转换为二进制数时，将该位置1。
SM1.7	当ASCII码不能转换为有效的十六进制数时，将该位置1。