

TLC 下位机编程指令说明

一、支持的继电器:

I, O, V, C, T

I --- 输入继电器, 可使用范围是: I00 ~ I03, 共 4 个;

O --- 输出继电器, 可使用范围是: O00 ~ O05, 共 6 个;

V --- 中间继电器, 可使用范围是: V00 ~ V31, 共 32 个;

C --- 计数继电器, 可使用范围是: C00 ~ C09, C80 ~ C89, 共 10 个(其中 C80~C89 为 C00~C09 的复位端口); 计数范围为 1~250; 在程序中 C00 和 C80、C01 和 C81、...、C09 和 C89 必须成对使用, 其中 C00 为计数输入端, 输入信号变化一次计一次数, 而 C80 为计数复位端, 得电时复位计数器为 0, 其它类推; 注意 C80~C89 只能出现在 OUT 语句中, 不能在输入语句里。

T --- 定时计数器, 可使用范围是: T00、T01、T10、T11、T20、T21、T30、T31、T40、T41, 共 10 个, 定时范围为 1~250, 其定时单位有 5 种, 分如下:

T00 和 T01 为 100uS;

T10 和 T11 为 10mS;

T20 和 T21 为 1S;

T30 和 T31 为 60S;

T40 和 T41 为 1h;

注意 100uS 和 10mS 的较小值定时可能不是很准确, 使用时要注意调整。

二、支持的指令:

LD, LDI, AND, ANI, OR, ORI, OUT, END

所有的梯形图程序第一句都必须是 LD 或 LDI, 结尾必须是 OUT;

在程序的最后必须有 END, 表示程序结束。

下面详细叙述上述指令的用法:

LD 常开触点

LD 指令助记格式:

LD I02 表示当 I02 高电平时通过

LD O04 表示当 O04 得电时通过

LD V08 表示当 V08 得电时通过

LD C07 表示当 C07 计数溢出时通过

LD T21 表示当 T21 定时溢出时通过

LDI 常闭触点

LDI 指令助记格式:

LDI I02 表示当 I02 低电平时通过

LDI O04 表示当 O04 失电时通过

LDI V08 表示当 V08 失电时通过

LDI C07 表示当 C07 计数未溢出时通过

LDI T21 表示当 T21 定时未溢出时通过

AND 串接常开触点

AND 指令助记格式:

AND I02 表示串接之前语句且当 I02 高电平时通过

AND O04 表示串接之前语句且当 O04 得电时通过

AND V08 表示串接之前语句且当 V08 得电时通过

AND C07 表示串接之前语句且当 C07 计数溢出时通过

AND T21 表示串接之前语句且当 T21 定时溢出时通过

ANI 串接常闭触点

ANI 指令助记格式:

- ANI I02 表示串接之前语句且当 I02 低电平时通过
 ANI O04 表示串接之前语句且当 O04 失电时通过
 ANI V08 表示串接之前语句且当 V08 失电时通过
 ANI C07 表示串接之前语句且当 C07 计数未溢出时通过
 ANI T21 表示串接之前语句且当 T21 定时未溢出时通过

OR 并联常开触点

OR 指令助记格式:

- OR I02 表示并接之前语句或当 I02 高电平时通过
 OR O04 表示并接之前语句或当 O04 得电时通过
 OR V08 表示并接之前语句或当 V08 得电时通过
 OR C07 表示并接之前语句或当 C07 计数溢出时通过
 OR T21 表示并接之前语句或当 T21 定时溢出时通过

ORI 并联常闭触点

ORI 指令助记格式:

- LDI I02 表示并接之前语句或当 I02 低电平时通过
 LDI O04 表示并接之前语句或当 O04 失电时通过
 LDI V08 表示并接之前语句或当 V08 失电时通过
 LDI C07 表示并接之前语句或当 C07 计数未溢出时通过
 LDI T21 表示并接之前语句或当 T21 定时未溢出时通过

OUT 输出数据

OUT 指令助记格式:

- OUT O04 OUT 有效时表示 O04 得电
 OUT V08 OUT 有效时表示 V08 得电
 OUT C07 200 C07 表示计数器, 200 为计数溢出值, OUT 经历一次有效无效时 C07 加 1
 OUT C87 C87 表示复位 C07 计数器, OUT 有效时复位 C07 计数器为 0
 OUT T21 200 T21 表示单位为 1 秒的定时器, 200 为定时溢出值, OUT 有效时 T21 计数, 无效时 T21 复位为 0

END 结束程序

END 指令助记格式:

END 表示程序的结束

三、编程示例

