

6SL3210-1KE32-1AB1现货西门子代理

产品名称	6SL3210-1KE32-1AB1现货西门子代理
公司名称	湖南西控自动化设备有限公司
价格	.00/件
规格参数	西门子:G120C 变频器:110KW内置A级滤波版 德国:3AC 380-480V滤波A级
公司地址	中国（湖南）自由贸易试验区长沙片区开元东路1306号开阳智能制造产业园（一期）4#栋301
联系电话	17838383235 17838383235

产品详情

4.[STEP4] PLC调试

将图2-7所示的梯形图程序经编译下载到PLC后，可以进行在线监控如图2-8所示。需要注意的是SB2急停按钮在接线上是常闭触电，因此在梯形图编程中画的是常开，正常情况实际的在线监控也是接通的，只有当按下急停按钮后，I1.0信号才断开。

图2-8 在线监控示意

2.2 定时器与时钟存储器

2.2.1 定时器种类

使用定时器指令用以创建可编程的延迟时间，表2-3所示为S7-1200的定时器指令，常用的为如下4种定时器：

- 1) TP：脉冲定时器可生成具有预设宽度时间的脉冲。
- 2) TON：接通延迟定时器，输出Q在预设的延时过后设置为ON。
- 3) TOF：关断延迟定时器，输出Q在预设的延时过后重置为OFF。
- 4) TONR：保持型接通延迟定时器，输出在预设的延时过后设置为ON。在使用R输入重置经过的时间之前，会跨越多个定时时段一直累加经过的时间。

表2-3 定时器指令

2.2.2 TON指令

TON指令就是接通延迟定时器输出Q在预设的延时过后设置为ON，其指令形式如图2-9所示，参数及其数据类型如表2-4所示。参数 IN 从 0 跳变为 1 将启动定时器TON。

图2-9 TON指令

表2-4 TON参数及数据类型

PT（预设时间）和 ET（经过的时间）值以表示毫秒时间的有符号双精度整数形式存储在存储器中（表2-5所示）。TIME 数据使用 T# 标识符，可以简单时间单元“T#200ms”或复合时间单元“T#2s_200ms（或T#2s200ms）”的形式输入。

表2-5 TIME数据类型

如图2-10所示，在指令窗口中选择“定时器操作”中的TON指令，并将之拖入到程序段中（图2-11），这时就会跳出一个“调用数据块”窗口，选择自动编号，则会直接生成DB1数据块；也可以选择手动编号，根据用户需要生成DB数据块。

图2-10 选择TON定时器操作

图2-11 TON指令调用数据块

在项目树的“程序块”中，可以看到自动生成的IEC_Timer_0_DB[DB1]数据块，生成后的TON指令调用如图2-12所示。

图2-12 TON指令调用示意

2.2.3 TOF定时器

TOF关断延迟定时器指令的参数与TON相同，区别在于IN 从1 跳变为 0 将启动定时器。

2.2.4 TP脉冲定时器

TP脉冲定时器指令虽然参数格式与TON、TOF一致，但含义跟接通延时和断电延时不同，它是在IN输入从0跳变到1之后，立即输出一个脉冲信号，其持续长度受PT值控制。

图2-13 TP指令应用

图2-14所示为TP指令时序图，从图中可以看到：即使TON的IN信号还处于“1”状态，TP指令输出Q在完成PT时长后，就不再保持为“1”；即使TON的IN信号为多个“脉冲”信号，输出Q也能完成PT时长的脉冲宽度。

图2-14 TP指令时序图

2.2.5 TONR时间累加器

TONR指令如图2-15所示，与TON、TOF、TP相比增加了参数R，相关的参数及数据类型见表2-7。

图2-15 TONR指令

表2-7 TONR参数及数据类型

图2-16所示为TONR的时序图，当IN信号不连续输入时，定时器ET的值一直在累计，直到定时时间PT到，ET的值保持为PT值；当R信号ON时，ET的值复位为零。

图2-16 TONR时序图

2.2.6系统和时钟存储器的选用

在报警指示中经常会碰到“闪烁”的频率概念，用TON等定时器可以完成，但更便捷的方式就是采用博途软件自带的PLC“系统和时钟存储器”。

在图2-17中，选中PLC属性中所示的“系统和时钟存储器”，点击右边窗口的复选框“启用系统存储器字节”和“启用时钟存储器字节”，采用默认的MB1、MB0作为系统存储器字节、时钟存储器字节，也可以修改该2字节的地址。

图2-17系统和时钟存储器

将M1.0设置为系统存储器字节后，该字节的M1.0~M1.3的意义如下：

M1.0(FirstScan): 仅在进入RUN模式的扫描时为1状态，以后为0状态。

M1.1(DiagStatusUpdate): 诊断状态已更改。

M1.2(Always TRUE): 总是为1状态，其常开触点总是闭合或高电平。

M1.3(Always FALSE): 总是为0状态，其常闭触点总是闭合或低电平。

时钟脉冲是一个周期内0状态和1状态所占的时间各为50%的方波信号，以M0.5为例，其时钟脉冲的周期为1s，如果用它的触点来控制接在某输出点的指示灯，指示灯将以1Hz的频率闪动，亮0.5s、熄灭0.5s。

因为系统存储器和时钟存储器不是保留的存储器，用户程序或通信可能改写这些存储单元，破坏其中的数据。应避免改写这两个M字节，保证它们的功能正常运行。指定了系统存储器和时钟存储器字节后，这些字节不能再作它用，否则将会使用户程序运行出错，甚至造成设备损坏或人身伤害。

【实例2-2】用TON指令延时启动电动机

某电动机在启动按钮SB1动作后10s之后才启动，在停止按钮SB2动作后立即停止，请用TON指令进行编程。

1.[STEP1] 定义输入输出元件和电气接线

表2-6所示的输入元件包括SB1启动按钮和SB2停止按钮，均采取常开触点接线；输出元件包括接触器KM1。具体电气接线如图2-18所示。

表2-6 输入输出元件及控制功能

	PLC软元件	元件符号/名称
输入	I0.0	SB1/启动按钮
	I0.1	SB2/停止按钮
输出	Q0.0	KM/接触器

图2-18 延时启动电动机电气接线

2.[STEP2] PLC梯形图编程

图2-19所示延时启动电动机PLC梯形图编程示意。

程序段1：对中间变量M10.1电动机起动信号定时10s，输出为Q0.0。

程序段2：采用启动按钮和停止按钮的SR触发器，输出为M10.1电动机起动信号，停止按钮复位优先。

需要注意的是：程序段1和2的位置对于本实例来说其先后次序不影响程序的正确执行。

图2-19 延时起动电动机梯形图

3.[STEP3]调试

为了更好地理解TON指令，图2-20所示为程序下载后的实时监控，即在DB1中实时读取当前的延时时间，如T#3S_110MS。

图2-20 TON指令实时监控

【实例2-3】用TONR指令统计设备运行时间

某设备所用的电动机在运行100分钟后要进行计时到指示，以便于维护人员进行停机检查。请用TONR指令进行编程。

1.[STEP1]定义输入输出元件和电气接线

表2-8所示为统计设备运行时间的输入输出元件及控制功能，包括3个输入信号的按钮和输出接触器、指示灯。电气接线如图2-21所示。

表2-8 输入输出元件及控制功能

	PLC软元件	元件符号/名称
输入	I0.0	SB1/启动按钮
	I0.1	SB2/停止按钮
	I0.2	SB3/计时复位按钮
输出	Q0.0	KM/接触器
	Q0.1	HL1/运行累计时间到指示

图2-21 统计设备运行时间电气接线

2.[STEP2]PLC梯形图编程

如图2-22所示为统计设备运行时间的梯形图。

程序段1：调用TONR对输入信号Q0.0（即接触器KM1）进行累计定时，100分钟时间一到即输出HL1指示灯Q0.1，该定时器可以通过SB3进行复位。

程序段2：调用SR触发器对输入信号SB1进行Q0.0置位、对停止按钮SB2和运行累计时间到信号HL1进行Q0.0复位，其中复位优先。

图2-22 统计设备运行时间梯形图

【实例2-4】用时钟存储器来编程指示灯闪烁

如图2-23所示的指示灯HL1有两种闪烁方式，一种是当SB1按下时进行快闪，另外一种当SB2按下时进行慢闪。当两个按钮同时按下时，指示灯HL1灭掉，然后进入待机状态，即按下SB1或SB2继续处于两种闪烁状态。请用时钟存储器来进行编程。

图2-23 指示灯闪烁示意

1.[STEP1]定义输入输出元件和电气接线

指示灯闪烁实例包括2个按钮输入和1个指示灯输入见表2-9。电气接线如图2-24所示。

表2-8 输入输出元件及控制功能

	PLC软元件	元件符号/名称
输入	I0.0	SB1/快闪按钮
	I0.1	SB2/慢闪按钮
输出	Q0.0	HL1/指示灯

图2-24 指示灯闪烁电气接线

2.[STEP2]PLC梯形图编程

图2-25所示为指示灯闪烁梯形图，采用M0.2来作为快闪的时钟存储器、M0.7来作为慢闪的时钟存储器。

程序段1：按下SB1快闪按钮，则置位快闪中间变量M10.0，复位慢闪中间变量M10.1。

程序段2：按下SB2慢闪按钮，则置位慢闪中间变量M10.1，复位快闪中间变量M10.0。

程序段3：同时按下2个按钮时，则复位%M10.0和M10.1，同时置位M10.2（即复位中间变量）。

程序段4：用2.5Hz表示快闪，用0.5Hz表示慢闪，在两种状态下输出指示灯。

程序段5：在同时按下2个按钮情况下，过2s后自动激活，运行再次进行慢闪或快闪动作。

图2-25 指示灯闪烁梯形图

2.3 比较、运算和移动指令

2.3.1 比较指令

表2-14所示为S7-1200共有10个常见的比较操作，用来比较数据类型相同的两个数IN1与IN2的大小，其操作数可以是I/Q/M/L/D等存储区中的变量或常量。当满足比较关系式给出的条件时，等效触点接通。

表2-14比较指令

表2-15所示为等于、不等于、大于等于、小于等于、大于、小于等6种比较指令触点的满足条件，且要比较的两个值必须为相同的数据类型。

表2-15比较指令触点

这里以“等于”比较指令为例进行说明：如图2-26a所示可以使用“等于”指令确定个比较值（<操作数1>）是否等于第二个比较值（<操作数2>）。比较器运算指令可以通过指令右上角黄色三角的个选项来选择等于、大于等于等比较器类型（图2-26b），也可以通过右下角黄色三角的第二个选项来选择数据类型，如整数、实数等（图2-26c）。

图2-26 比较器运算指令

（1）CMP==：等于比较器：

可以使用“等于”指令判断个比较值（<操作数1>）是否等于第二个比较值（<操作数2>）。如果满足比较条件，则该指令返回逻辑运算结果（RLO）“1”。如果不满足比较条件，则该指令返回RLO“0”。

（2）CMP<>：不等于

使用“不等于”指令判断个比较值（<操作数1>）是否不等于第二个比较值（<操作数2>）。如果满足比较条件，则该指令返回逻辑运算结果（RLO）“1”。如果不满足比较条件，则该指令返回RLO“0”。

（3）CMP>=：大于或等于

可以使用“大于或等于”指令判断个比较值（<操作数1>）是否大于或等于第二个比较值（<操作数2>）。如果满足比较条件，则该指令返回逻辑运算结果（RLO）“1”。如果不满足比较条件，则该指令返回RLO“0”。

（4）CMP<=：小于或等于

可以使用“小于或等于”指令判断个比较值（<操作数1>）是否小于或等于第二个比较值（<操作数

2>)。如果满足比较条件,则该指令返回逻辑运算结果(RLO)“1”。如果不满足比较条件,则该指令返回 RLO“0”。

(5) CMP> : 大于

可以使用“大于”指令确定个比较值(<操作数 1>)是否大于第二个比较值(<操作数 2>)。如果满足比较条件,则该指令返回逻辑运算结果(RLO)“1”。如果不满足比较条件,则该指令返回 RLO“0”。

(6) CMP< : 小于

可以使用“小于”指令判断个比较值(<操作数 1>)是否小于第二个比较值(<操作数 2>)。如果满足比较条件,则该指令返回逻辑运算结果(RLO)“1”。如果不满足比较条件,则该指令返回 RLO为“0”。

除了上述的常见比较指令之外,还有其他变量比较指令,其类型与说明如表2-14所示。

表2-14 变量比较类型与说明

2.3.2 移动指令

移动指令是将数据元素复制到新的存储器地址,并从一种数据类型转换为另一种数据类型,移动过程中不更改源数据

1.MOVE移动值

如图2-27可以使用“移动值”指令将 IN 输入操作数中的内容传送给 OUT1 输出的操作数中。始终沿地址升序方向进行传送。如果使能输入 EN 的信号状态为“0”或 IN 参数的数据类型与 OUT1 参数的指定数据类型不对应时,则使能输出 ENO 的信号状态为“0”

图2-27 MOVE指令

表2-15 MOVE指令可传送的类型

在MOVE指令中,若IN 输入端数据类型的位长度超出了OUT1 输出端数据类型的位长度,则传送源值中多出来的有效位会丢失。若 IN 输入端数据类型的位长度小于 OUT1 输出端数据类型的位长度,则用零填充传送目标值中多出来的有效位。

在初始状态,指令框中包含 1 个输出 (OUT1),可以鼠标点击图符扩展输出数目。在该指令框中,应按升序顺序排列所添加的输出端。执行该指令时,将 IN

输入端操作数中的内容发送到所有可用的输出端。如果传送结构化数据类型 (DTL, STRUCT, ARRAY) 或字符串 (STRING) 的字符, 则无法扩展指令框。可以输出多个地址OUT1、OUT2、OUT3等, 如图2-28所示。

图2-28 MOVE指令的多个变量输出

2.MOVE_BLK指令

如图2-29所示, 使用“ MOVE_BLK块移动 ”指令, 可将存储区 (源区域) 的内容移动到其他存储区 (目标区域)。使用参数 COUNT 可以指定待复制到目标区域中的元素个数。可通过 IN 输入端的元素宽度来指定待复制元素的宽度。并按地址升序顺序执行复制操作。

图2-29 MOVE_BLK指令

3.UMOVE_BLK无中断块移动

使用图2-30所示的“ UMOVE_BLK无中断块移动 ”指令, 可将存储区 (源区域) 的内容连续复制到其他存储区 (目标区域)。使用参数 COUNT 可以指定待复制到目标区域中的元素个数。可通过 IN 输入端的元素宽度来指定待复制元素的宽度。源区域内容沿地址升序方向复制到目标区域。

图2-30 UMOVE_BLK指令

4.FILL_BLK填充块

图2-31所示的“ FILL_BLK填充块 ”指令, 用 IN 输入的值填充一个存储区域 (目标区域)。将以 OUT 输出指定的起始地址, 填充目标区域。可以使用参数 COUNT 指定复制操作的重复次数。执行该指令时, 将选择 IN 输入的值, 并复制到目标区域 COUNT 参数中指定的次数。

图2-31 FILL_BLK填充块

5.SWAP交换指令

“ SWAP交换 ”指令可以更改输入 IN 中字节的顺序, 并在输出 OUT 中查询结果。图2-32说明了如何使用“ 交换 ”指令交换数据类型为 DWORD 的操作数的字节。表2-15所示为SWAP指令的参数。

图2-32 SWAP交换数据类型为DWORD的示意

表2-15 SWAP指令的参数

2.3.3 数学运算指令

在数学运算指令中，ADD、SUB、MUL和DIV分别是加、减、乘、除指令，其操作数的数据类型可选SInt、Int、Dint、USInt、UInt、UDInt和Real。在运算过程中，操作数的数据类型应该相同。

1.加法ADD指令

加法ADD指令可以从TIA软件右边指令窗口的“基本指令”下的“数学函数”中直接添加（图2-33a所示）。使用“ADD”指令，根据图2-33b选选择的数据类型，将输入IN1的值与输入IN2的值相加，并在输出OUT（ $OUT = IN1 + IN2$ ）处查询总和。

图2-33 ADD指令

在初始状态下，指令框中至少包含两个输入（IN1和IN2），可以鼠标点击图符扩展输入数目（图2-33c），在功能框中按升序对插入的输入进行编号，执行该指令时，将所有可用输入参数的值相加，并将求得和存储在输出OUT中。

表2-16列出了“ADD”指令的参数。根据参数说明，只有使能输入EN的信号状态为“1”时，才执行该指令。如果成功执行该指令，使能输出ENO的信号状态也为“1”。如果满足下列条件之一，则使能输出ENO的信号状态为“0”：

使能输入EN的信号状态为“0”。

指令结果超出输出OUT指定的数据类型的允许范围。

浮点数具有无效值。

表2-16 ADD指令的参数

图2-34中举例说明了ADD指令的工作原理：如果操作数I0.0的信号状态为“1”，则将执行“加”指令，将操作数IW64的值与IW66的值相加，并将相加的结果存储在操作数MW0中。如果该指令执行成功，则使能输出ENO的信号状态为“1”，同时置位输出Q0.0。

图2-34 ADD指令应用