西门子S7-300授权总经销商 6ES7174-0AA10-0AA0 接口模块 IM174

产品名称	西门子S7-300授权总经销商 6ES7174-0AA10-0AA0 接口模块 IM174
公司名称	浔之漫智控技术(上海)有限公司
价格	.00/件
规格参数	西门子:现货 S7-300:正品 德国:全新
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	15801997124 15801997124

产品详情

西门子S7-300授权总经销商 6ES7174-0AA10-0AA0 接口模块 IM174

6ES7174-0AA10-0AA0

SIMATIC S7-300，接口模块 IM174，用于连接模拟驱动和步进驱动 在时钟同步的 PROFIBUS 上 在 Motion Control 控制器上 4 个通道（4 个编码器输入，4 AA)， 可项目组态，利用 STEP 7 V5.4 SP4

转换指令概述 可使用下列指令将二进制编码的十进制数和整数转换为其它类型的数字： BTI ITB 将BCD码转换为整型(16位) 将整型(16位)转换为BCD码 BTD 将BCD码转换为整型(32位) ITD 将整型(16位)转换为长整型(32位) DTB 将长整型(32位)转换为BCD码 DTR 将长整型(32位)转换为浮点数(32位IEEE 754)
可使用下列指令计算整数的补(反)码，或将浮点数的符号取反： INVI 对整数求反码(16位) INVD 二进制反码双精度整数(32位) NEGI 对整数求补码(16位) NEGD 二进制补码双精度整数(32位) NEGR 浮点数(32位，IEEE 754)取反
可使用下列"改变累加器1中的位顺序"指令将累加器1的低字字节或整个累加器中的字节的顺序反转。 CAW 改变ACCU 1-L (16位)中的字节顺序 CAD 改变ACCU 1 (32位)中的字节顺序
可使用下列任何指令将累加器1中的32位IEEE浮点数转换为32位整型(长整型)。各个指令的取整方法有所不同： RND 取整 TRUNC 截断 RND+ 取整为高位长整数 RND- 取整为低位长整数
S7-300 和S7-400 编程的语句表(STL) 参考手册, 05/2017, A5E41525031-AA 43 转换指令 3.2 BTI 将BCD码转换为整型(16位) 3.2 BTI 将BCD码转换为整型(16位) 格式 BTI 描述 BTI (3位BCD数从十进制到二进制的转换)将ACCU 1-L的内容解释为三位二进制编码的十进制数

(BCD码)，并将其转换为16位整型。结果存储在累加器1的低字中。累加器1的高字和累加器2保 持不变。
ACCU 1-L中的BCD数字：BCD数字的允许值范围从"-999"至"+999"。位0到位11解释为数值，位
15解释为BCD数字的符号(0 = 正，1= 负)。位12至位14在转换中不使用。如果BCD数字的十进
制(四位)数字处于10至15的无效范围，则在转换期间会出现BCDF错误。通常，CPU会转入STOP
模式。但是，通过对OB121编程可设计另一种出错响应，用以处理该同步编程错误。 状态字 BR CC 1
CC 0 OV OS OR STA RLO /FC 写： - - - - - - - - - 实例 STL 解释 L MW10 //将BCD数字载入ACCU 1-L。
BTI //从BCD码转换为整型；将结果存储在ACCU 1-L中。 T MW20 //将结果(整数)传送到MW20。
1010100010010000 1100100111000000 BTIBCD to Integer "+915" BCD 15.....87.....0 " + "" 9 "" 1 "" 5 " MW10 "+915"
IntegerMW20 S7-300和S7-400编程的语句表(STL) 44 参考手册, 05/2017, A5E41525031-AA 转换指令 3.3 ITB
将整型(16位)转换为BCD码 3.3 ITB 将整型(16位)转换为BCD码 格式 ITB 描述 ITB
(16位整数从二进制到十进制的转换)将ACCU 1-L的内容解释为16位整数，并将其转换为三位
二进制编码的十进制数(BCD码)。结果存储在累加器1的低字中。位0到位11包含BCD数字的值。
位12至位15用来表示BCD数字的符号状态(0000 = 正，1111= 负)。累加器1的高字和累加器2保 持不变。
BCD数字的范围为"-999"至"+999"。如果超出允许范围，则状态位OV和OS被置位为1。
执行该指令时不涉及RLO，也不会影响RLO。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写： - - - x
x - - - - 实例 STL 解释 L MW10 //将整数载入ACCU 1-L。 ITB //从整型转换为BCD
(16位)；将结果存储在ACCU 1-L中。 T MW20 //将结果(BCD数字)传送到MW20。 1100011001111111
1100100000101111 ITBInteger to BCD "-413" Integer 15.....87.....0 MW10 "-413" BCDMW20 " - "" 4 "" 1 "" 3 "
S7-300和S7-400编程的语句表(STL) 参考手册，05/2017，A5E41525031-AA 45 转换指令 3.4 BTD
将BCD码转换为整型(32位) 3.4 BTD 将BCD码转换为整型(32位) 格式 BTD 描述 BTD
(7位BCD数字从十进制到二进制的转换)将ACCU 1的内容解释为7位二进制编码的十进制数
(BCD)，并将其转换为32位长整型。结果存储在累加器1中。累加器2保持不变。 ACCU
1中的BCD数字：BCD数字的允许值范围从"-9,999,999"至"+9,999,999"。位0到位27解释
为数值，位31解释为BCD数字的符号(0 = 正，1= 负)。位28至位30在转换中不使用。
如果任何十进制数(BCD编码的四位组)处于10至15的无效范围，则在转换期间会出现BCDF错误。
通常，CPU会转入STOP模式。但是，通过对OB121编程可设计另一种出错响应，用以处理该同步
编程错误。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写： - - - - - - - - - 实例 STL 解释 L MD10
//将BCD数字载入ACCU 1。 BTD //从BCD码转换为整型；结果存储在ACCU 1中。 T MD20
//将结果(长整型)传送到MD20。 BTDBCD to Double Integer"+157821" 31.....1615.....0 " + "" 0 "" 1 "" 5 " MD10
"+157821" MD20 10101000000000001000010000011110 " 7 "" 8 "" 2 "" 1 " 01000000000000001011111000010110
S7-300和S7-400编程的语句表(STL) 46 参考手册, 05/2017, A5E41525031-AA 转换指令 3.5 ITD
将整型(16位)转换为长整型(32位) 3.5 ITD 将整型(16位)转换为长整型(32位) 格式 ITD 描述 ITD
(16位整数转换为32位整数)将ACCU 1-L的内容解释为16位整数并将其转换为32位长整数。
结果存储在累加器1中。累加器2保持不变。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写： - - - - - -
- - - 实例 STL 解释 L MW12 //将整数载入ACCU 1。 ITD //从整型(16位)转换为长整型(32位)；
//将结果存储在ACCU 1中。 T MD20 //将结果(长整型)传送到MD20。 实例：MW12 ="-10"
(整型，16位) 目录 ACCU1-H ACCU1-L 位 31 . . . . . 16 15 . . . . . 0 执行ITD之前 XXXX XXXX
XXXX XXXX 1111 1111 1111 0110 执行ITD之后 1111 1111 1111 1111 1111 1111 1111 0110 (X =
0或1，这些位不用于转换) S7-300和S7-400编程的语句表(STL) 参考手册，05/2017，A5E41525031-AA 47
转换指令 3.6 DTB 将长整型(32位)转换为BCD码 3.6 DTB 将长整型(32位)转换为BCD码 格式 DTB
描述 DTB (32位整型数的二进制到十进制转换)将ACCU
1中的内容解析为32位长整型数，并将其转换为
七位二进制编码的十进制数(BCD)。结果保存在累加器1中。位0到位27包含BCD数字的值。位28
至位31用来表示BCD数字的符号状态(0000 = 正，1111 = 负)。累加器2保持不变。
BCD数的范围为"-9,999,999"至"+9,999,999"。如果超出允许范围，则状态位OV和OS被置位为1。 状态字
BR CC 1 CC 0 OV OS OR STA RLO /FC 写： - - - x x - - - - 实例 STL 解释 L MD10
//将32位整数载入ACCU 1。 DTB //从整型(32位)转换为BCD码，结果存储在ACCU 1中。 T MD20
//将结果(BCD数字)传送到MD20。 DTBInteger to BCD"-701" Integer 31.....1615.....0 MD10 "-701" BCD MD20
11111111111111111100001010111111 " - "" 0 "" 0 "" 0 "" 0 "" 7 "" 0 "" 1 " 00000000000011111000000011100000
S7-300和S7-400编程的语句表(STL) 48 参考手册, 05/2017, A5E41525031-AA 转换指令 3.7 DTR
将长整型(32位)转换为浮点数(32位IEEE 754) 3.7 DTR 将长整型(32位)转换为浮点数(32位IEEE 754)

格式 DTR 描述 DTR (32位整数转换为32位IEEE浮点数)将ACCU 1的内容解释为32位长整型，并将其转换为32 位IEEE浮点数。如必要，该指令会对结果取整。(32位整数比32位浮点数精度更高)。结果存储在累加器1中。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写： - - - - - - - - - 实例 STL 解释 L MD10 //将32位整数载入ACCU 1。 DTR //从长整型转换为浮点型(32位IEEE FP)；将结果存储在 //ACCU 1中。 T MD20 //将结果(BCD数字)传送到MD20。 DTR Integer (32 bit) to IEEE floating-point (32 Bit)"+500" Integer 31...0 MD10 "+500" IEEE-FP MD20 00000000000000000010111110000000 1 bit Sign of the mantissa 8-bit exponent 01011111110000100000000000000000 30...22... 23-bit mantissa S7-300和S7-400编程的语句表(STL) 参考手册，05/2017，A5E41525031-AA 49 转换指令 3.8 INVI 对整数求反码(16位) 3.8 INVI 对整数求反码(16位) 格式 INVI 描述 INVI (对整数求反码)在ACCU 1-L中形成16位数值的二进制反码。二进制反码是通过将各个位的值 取反形成的，即，用"0"替换"1"，用"1"替换"0"。结果存储在累加器1的低字中。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写： - - - - - - - - - 实例 STL 解释 L IW8 //将值载入ACCU 1-L。 INVI //形成16位二进制反码。 T MW10 //将结果传送到MW10。 目录 ACCU1-L 位 15 . . . . . 0 执行IVNI之前 0110 0011 1010 1110 执行INVI之后 1001 1100 0101 0001 S7-300和S7-400编程的语句表(STL) 50 参考手册, 05/2017, A5E41525031-AA 转换指令 3.9 INVD 二进制反码双精度整数(32位) 3.9 INVD 二进制反码双精度整数(32位) 格式 INVD 描述 INVD (二进制反码双精度整数)在ACCU 1中形成32位数值的二进制反码。二进制反码是通过将各个 位的值取反形成的，即，用"0"替换"1"，用"1"替换"0"。结果存储在累加器1中。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写： - - - - - - - - - 实例 STL 解释 L ID8//将值载入ACCU 1中。 INVD //形成二进制反码(32位)。 T MD10 //将结果传送到MD10。 目录 ACCU1-H ACCU1-L 位 31 . . . . . 16 15 . . . . . 0 执行IVND之前 0110 1111 1000 1100 0110 0011 1010 1110 执行INVD之后 1001 0000 0111 0011 1001 1100 0101 0001 S7-300和S7-400编程的语句表(STL) 参考手册，05/2017，A5E41525031-AA 51 转换指令 3.10 NEGI 对整数求补码(16位) 3.10 NEGI 对整数求补码(16位) 格式 NEGI 描述 NEGI (对整数求补码)在ACCU 1-L中形成16位数值的二进制补码。二进制补码是通过将各个位的值 取反形成的，即，用"0"替换"1"，用"1"替换"0"；然后加"1"。结果存储在累加器1的低字中。补码指令 等效于乘以"-1"。状态位CC 1、CC 0、OS和OV则根据函数运算的结果来设置。