

三菱微型可编程控制器  
MELSEC-F

FXCPU

结构化编程手册

应用函数篇

**FX**



# FXCPU 结构化编程手册

## [应用函数篇]

手册编号	JY997D67501
版本	A
制作年月	2016年6月

### 通告

---

此次承蒙购买FX系列可编程控制器产品，诚表谢意。  
本手册描述了关于MELSEC-F FX系列可编程控制器结构化程序用的应用函数。  
在使用之前，请阅读本书以及相关产品的手册，希望在充分理解其规格的前提下正确使用产品。  
此外，希望本手册能够送达至最终用户处。

根据本书的内容，并非对工业所有权其他的权利的实施予以保证，或是承诺实施权。  
此外，关于因使用本书中记载的内容而引起的工业所有权方面的各种问题，本公司不承担任何责任。

## 使用时的请求

---

- 该产品是以一般的工业为对象制作的通用产品,因此不是以用于关系到人身安全之类的情况下使用的机器或是系统为目的而设计、制造的产品。
- 考虑将该产品用于原子能、电力、宇航、医疗、乘用移动物体用的机器或是系统等特殊用途的时候,请与本公司的营业窗口查询。
- 虽然该产品是在严格的质量体系下生产的,但是用于那些因该产品故障而可能导致重大故障或是产生损失的设备的时候,请在系统上设置后备和安全功能。
- 该产品和其他产品组合使用的情况下,请用户确保应该符合的规格、法规或是规则。此外,关于用户使用的系统、机械、设备中该产品的适用性和安全性,请用户自行确认。

## 预先通知

---

- 使用产品时如有疑问,请向具有电气知识(电气施工人员或是同等以上的知识)的专业电气技术人员咨询。关于该产品的操作和使用方法有疑问时,请向技术咨询窗口咨询。
- 本书、技术资料、样本等中记载的事例是作为参考用的,不保证动作。选用的时候,请用户自行对机器、设备的功能和安全性进行确认以后使用。
- 关于本书的内容,有时候为了改进可能会有不事先预告就更改规格的情况,还望见谅。
- 关于本书的内容期望能做到完美,可是万一有疑问或是发现有错误,烦请联系本书封底记载的本公司或办事处。  
此时,请将前页中记载的手册编号一并告知。

## 关于商标

---

- Microsoft<sup>®</sup>、Windows<sup>®</sup>、Excel<sup>®</sup>是美国Microsoft Corporation在美国以及其他国家中的注册商标或者商标。
- CompactFlash是SanDisk公司在美国以及其他国家的商标。
- Ethernet是美国Xerox Corporation的注册商标。
- MODBUS<sup>®</sup>是Schneider Electric SA的注册商标。
- 其他的公司名称、产品名称是其各公司的商标或注册商标。

# 目录

本手册的定位 .....	7
相关手册的介绍 .....	10
关于手册中使用的总称·简称的记载 .....	12

---

## 1. 概要 13

---

1.1 结构化程序的概要及编程语言 .....	13
1.1.1 结构化程序的概要 .....	13
1.1.2 编程语言 .....	14
1.2 可编程控制器与编程软件的版本 .....	15
1.3 编程方面的基本注意事项 .....	15
1.3.1 输入输出处理, 响应延迟 .....	15
1.3.2 双重输出(双线圈)的动作及对策 .....	16
1.3.3 不能按结构化梯形图编程的回路图及对策 .....	17
1.3.4 一般标志位的使用 .....	17
1.3.5 运算错误标志位的使用 .....	20

---

## 2. 函数、操作符一览 21

---

2.1 型转换函数 .....	21
2.2 单数值变量函数 .....	24
2.3 算术运算函数 .....	24
2.4 位移位函数 .....	24
2.5 位型布尔函数 .....	24
2.6 选择函数 .....	25
2.7 比较函数 .....	25
2.8 字符串函数 .....	25
2.9 时间数据类型函数 .....	25
2.10 功能模块 .....	26
2.11 操作符 .....	26
2.11.1 算术运算 .....	26
2.11.2 位型布尔 .....	26
2.11.3 比较 .....	27

---

## 3. 函数的构成 28

---

3.1 应用函数的表示和执行形式 .....	28
3.2 标签 .....	29
3.3 软元件和地址 .....	32
3.4 EN和ENO .....	33

---

## 4. 函数的阅读方法 34

---

---

## 5. 应用函数(型转换函数) 36

---

5.1 BOOL_TO_INT(_E) / 位型 → 字[带符号]型转换 .....	38
5.2 BOOL_TO_DINT(_E) / 位型 → 双字[带符号]型转换 .....	40
5.3 BOOL_TO_STR(_E) / 位型 → 字符串型转换 .....	42
5.4 BOOL_TO_WORD(_E) / 位型 → 字[无符号] / 位列[16位]型转换 .....	44
5.5 BOOL_TO_DWORD(_E) / 位型 → 双字[无符号] / 位列[32位]型转换 .....	46

5.6	BOOL_TO_TIME(E) / 位型 → 时间型转换	48
5.7	INT_TO_DINT(E) / 字[带符号]型 → 双字[带符号]型转换	50
5.8	DINT_TO_INT(E) / 双字[带符号]型 → 字[带符号]型转换	52
5.9	INT_TO_BOOL(E) / 字[带符号]型 → 位型转换	54
5.10	DINT_TO_BOOL(E) / 双字[带符号]型 → 位型转换	56
5.11	INT_TO_REAL(E) / 字[带符号]型 → 单精度实数型转换	58
5.12	DINT_TO_REAL(E) / 双字[带符号]型 → 单精度实数型转换	60
5.13	INT_TO_STR(E) / 字[带符号]型 → 字符串型转换	62
5.14	DINT_TO_STR(E) / 双字[带符号]型 → 字符串型转换	64
5.15	INT_TO_WORD(E) / 字[带符号]型 → 字[无符号] / 位列[16位]型转换	66
5.16	DINT_TO_WORD(E) / 双字[带符号]型 → 字[无符号] / 位列[16位]型转换	68
5.17	INT_TO_DWORD(E) / 字[带符号]型 → 双字[无符号] / 位列[32位]型转换	70
5.18	DINT_TO_DWORD(E) / 双字[带符号]型 → 双字[无符号] / 位列[32位]型转换	72
5.19	INT_TO_BCD(E) / 字[带符号]型 → BCD型转换	74
5.20	DINT_TO_BCD(E) / 双字[带符号]型 → BCD型转换	76
5.21	INT_TO_TIME(E) / 字[带符号]型 → 时间型转换	78
5.22	DINT_TO_TIME(E) / 双字[带符号]型 → 时间型转换	80
5.23	REAL_TO_INT(E) / 单精度实数型 → 字[带符号]型转换	82
5.24	REAL_TO_DINT(E) / 单精度实数型 → 双字[带符号]型转换	84
5.25	REAL_TO_STR(E) / 单精度实数型 → 字符串型转换	86
5.26	WORD_TO_BOOL(E) / 字[无符号] / 位列[16位]型 → 位型转换	89
5.27	DWORD_TO_BOOL(E) / 双字[无符号] / 位列[32位]型 → 位型转换	91
5.28	WORD_TO_INT(E) / 字[无符号] / 位列[16位]型 → 字[带符号]型转换	93
5.29	WORD_TO_DINT(E) / 字[无符号] / 位列[16位]型 → 双字[带符号]型转换	95
5.30	DWORD_TO_INT(E) / 双字[无符号] / 位列[32位]型 → 字[带符号]型转换	97
5.31	DWORD_TO_DINT(E) / 双字[无符号] / 位列[32位]型 → 双字[带符号]型转换	99
5.32	WORD_TO_DWORD(E) / 字[无符号] / 位列[16位]型 → 双字[无符号] / 位列[32位]型转换	101
5.33	DWORD_TO_WORD(E) / 双字[无符号] / 位列[32位]型 → 字[无符号] / 位列[16位]型转换	103
5.34	WORD_TO_TIME(E) / 字[无符号] / 位列[16位]型 → 时间型转换	105
5.35	DWORD_TO_TIME(E) / 双字[无符号] / 位列[32位]型 → 时间型转换	107
5.36	STR_TO_BOOL(E) / 字符串型 → 位型转换	109
5.37	STR_TO_INT(E) / 字符串型 → 字[带符号]型转换	111
5.38	STR_TO_DINT(E) / 字符串型 → 双字[带符号]型转换	113
5.39	STR_TO_REAL(E) / 字符串型 → 单精度实数型转换	115
5.40	STR_TO_TIME(E) / 字符串型 → 时间型转换	118
5.41	BCD_TO_INT(E) / BCD型 → 字[带符号]型转换	120
5.42	BCD_TO_DINT(E) / BCD型 → 双字[带符号]型转换	122
5.43	TIME_TO_BOOL(E) / 时间型 → 位型转换	124
5.44	TIME_TO_INT(E) / 时间型 → 字[带符号]型转换	126
5.45	TIME_TO_DINT(E) / 时间型 → 双字[带符号]型转换	128
5.46	TIME_TO_STR(E) / 时间型 → 字符串型转换	130
5.47	TIME_TO_WORD(E) / 时间型 → 字[无符号] / 位列[16位]型转换	132
5.48	TIME_TO_DWORD(E) / 时间型 → 双字[无符号] / 位列[32位]型转换	134
5.49	BITARR_TO_INT(E) / 位阵列 → 字[带符号]型、字[无符号] / 位列[16位]型转换	136
5.50	BITARR_TO_DINT(E) / 位阵列 → 双字[带符号]型、双字[无符号] / 位列[32位]型转换	138
5.51	INT_TO_BITARR(E) / 字[带符号]型、字[无符号] / 位列[16位]型 → 位阵列转换	140
5.52	DINT_TO_BITARR(E) / 双字[带符号]型、双字[无符号] / 位列[32位]型 → 位阵列转换	142
5.53	CPY_BITARR(E) / 位阵列的复制	144
5.54	GET_BIT_OF_INT(E) / 字[带符号]型数据的指定位读出	146
5.55	SET_BIT_OF_INT(E) / 字[带符号]型数据的指定位写入	148
5.56	CPY_BIT_OF_INT(E) / 字[带符号]型数据的指定位复制	150
5.57	GET_BOOL_ADDR / 起始数据的获取	152
5.58	GET_INT_ADDR / 起始数据的获取	153
5.59	GET_WORD_ADDR / 起始数据的获取	154

6. 应用函数(单数值变量函数)	155
6.1 ABS(_E) / 绝对值	156
7. 应用函数(算术运算函数)	158
7.1 ADD_E / 加法运算	159
7.2 SUB_E / 减法运算	161
7.3 MUL_E / 乘法运算	163
7.4 DIV_E / 除法运算	164
7.5 MOD(_E) / 余数	166
7.6 EXPT(_E) / 幂乘	168
7.7 MOVE(_E) / 代入	170
8. 应用函数(位移位函数)	172
8.1 SHL(_E) / 左移	173
8.2 SHR(_E) / 右移	175
9. 应用函数(位型布尔函数)	177
9.1 AND_E / 逻辑与	178
9.2 OR_E / 逻辑或	180
9.3 XOR_E / 逻辑异或	181
9.4 NOT(_E) / 逻辑非	183
10. 应用函数(选择函数)	185
10.1 SEL(_E) / 选择值	186
10.2 MAXIMUM(_E) / 选择最大值	188
10.3 MINIMUM(_E) / 选择最小值	190
10.4 LIMITATION(_E) / 上下限限位控制	192
10.5 MUX(_E) / 多路选择器	194
11. 应用函数(比较函数)	196
11.1 GT_E / 比较	197
11.2 GE_E / 比较	199
11.3 EQ_E / 比较	200
11.4 LE_E / 比较	201
11.5 LT_E / 比较	202
11.6 NE_E / 比较	203
12. 应用函数(字符串函数)	204
12.1 MID(_E) / 字符串的选取	205
12.2 CONCAT(_E) / 字符串的结合	208
12.3 INSERT(_E) / 字符串的插入	210
12.4 DELETE(_E) / 字符串的删除	213
12.5 REPLACE(_E) / 字符串的替换	215
12.6 FIND(_E) / 字符串的检索	218

13. 应用函数(时间数据类型函数)	220
13.1 ADD_TIME(E) / 加法运算	221
13.2 SUB_TIME(E) / 减法运算	223
13.3 MUL_TIME(E) / 乘法运算	225
13.4 DIV_TIME(E) / 除法运算	227
14. 功能模块	229
14.1 R_TRIG(E) / 上升沿检测	230
14.2 F_TRIG(E) / 下降沿检测	232
14.3 CTU(E) / 增计数器	234
14.4 CTD(E) / 减计数器	236
14.5 CTUD(E) / 增减计数器	238
14.6 TP(E), TP_10(E) / 脉冲定时器	240
14.7 TON(E), TON_10(E) / ON延迟定时器	242
14.8 TOF(E), TOF_10(E) / 断开延迟定时器	244
14.9 COUNTER_FB_M / 计数器功能模块	246
14.10 TIMER_10_FB_M / 定时器功能模块	248
14.11 TIMER_CONT_FB_M / 定时器功能模块	249
14.12 TIMER_100_FB_M / 定时器功能模块	251
15. 操作符	252
15.1 ADD / 加法运算	253
15.2 SUB / 减法运算	254
15.3 MUL / 乘法运算	255
15.4 DIV / 除法运算	256
15.5 MOD / 余数	257
15.6 ** / 幂乘	258
15.7 AND / 逻辑与	259
15.8 OR / 逻辑或	260
15.9 XOR / 逻辑异或	261
15.10 NOT / 逻辑非	263
15.11 GT / 比较	264
15.12 GE / 比较	265
15.13 EQ / 比较	266
15.14 LE / 比较	267
15.15 LT / 比较	268
15.16 NE / 比较	269
附录A. 软元件和地址的对应	271
附录B. 函数、操作符一览 【按种类 / 字母顺序】	272
附录B-1 【种类】	272
附录B-2 【按字母顺序】	275
关于保证	279
改订的历史记录	280



## 本手册的定位

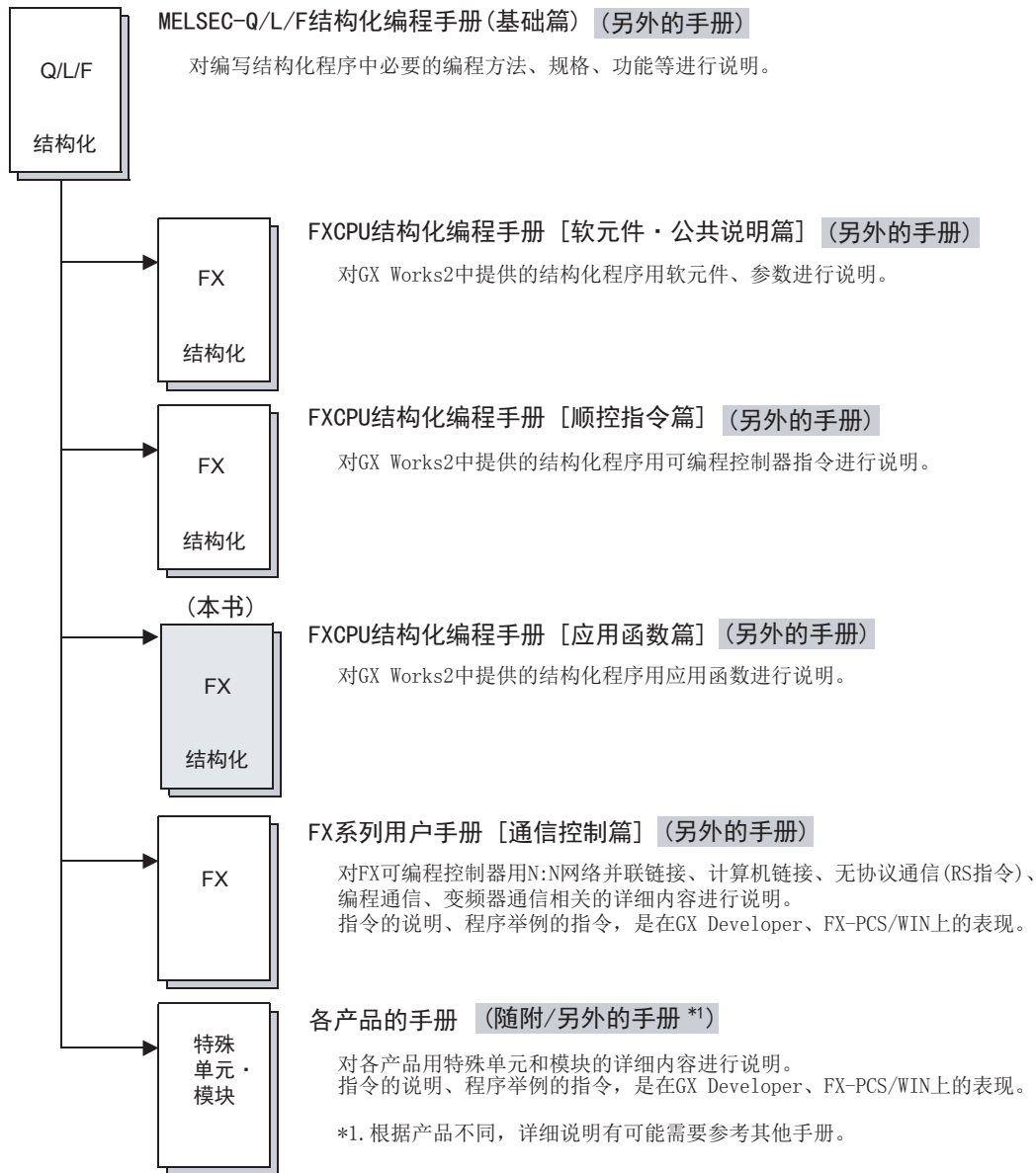
本手册对GX Works2中提供的用于结构化程序的应用函数进行了说明。关于软元件、参数、结构化程序用顺控指令，请参考其他手册。

另外，关于模拟量、通信、定位控制、特殊单元和模块，请参考各用途的手册。

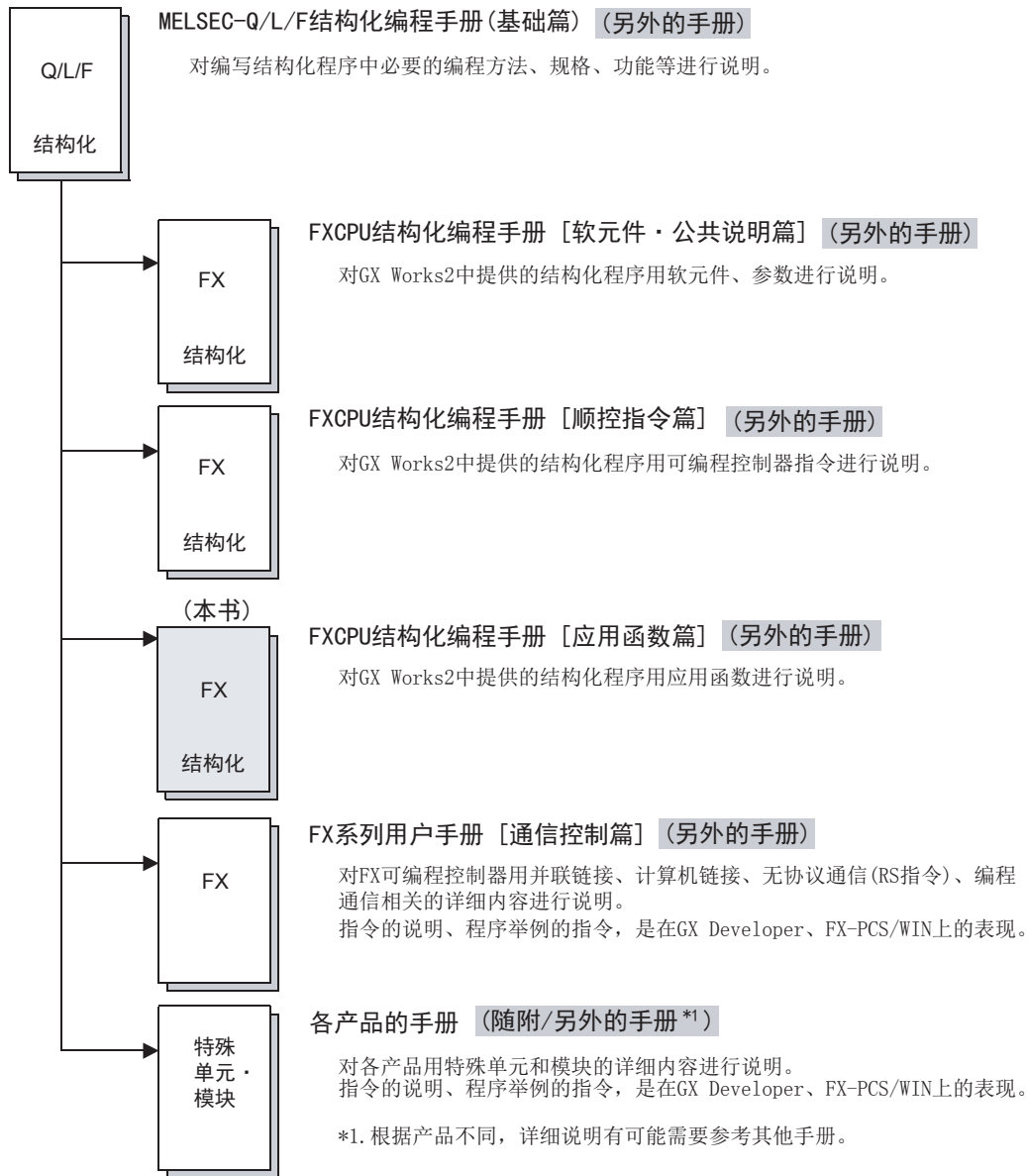
### 1. 使用FX3s・FX3G・FX3GC・FX3U・FX3UC可编程控制器时



2. 使用FX1S · FX1N · FX1NC · FX2N · FX2NC可编程控制器时



### 3. 使用FX0s · FX0 · FX0N · FX2 · FX2c可编程控制器时



## 相关手册的介绍

本手册介绍了GX Works2 的结构化工程中提供的应用函数。  
关于应用函数的软元件、参数的详细内容，请参考其他手册。  
在此仅对本手册中的参考手册、以及记载有可编程控制器主机硬件信息的手册进行介绍。  
在没有介绍的手册中，可能会有根据用途需要的手册。  
请务必参考可编程控制器主机的手册、所用产品附带的手册。此外，关于需要的手册，请向当初购入产品时的供应商咨询。

### FX可编程控制器公共[结构化]

手册名称	随附 另外的 手册	内容
MELSEC-Q/L/F 结构化编程手册(基础篇)	另外的 手册	关于编写结构化程序中必要的编程方法、规格、功能等的內容。
FXCPU 结构化编程手册 [软元件・公共说明篇]	另外的 手册	GX Works2的结构化工程中所提供软元件、参数等的內容。
FXCPU 结构化编程手册[顺控指令篇]	另外的 手册	关于GX Works2的结构化工程中所提供顺控指令的內容。
FXCPU 结构化编程手册[应用函数篇]	另外的 手册	GX Works2的结构化工程中所提供应用函数的內容。

### FX3S・FX3G・FX3GC・FX3U・FX3UC可编程控制器

手册名称	随附 另外的 手册	内容
<b>可编程控制器主机</b>		
FX3U系列硬件手册	随附	关于FX3U可编程控制器主机的输入输出规格及接线、安装，从FX3U用户手册[硬件篇]中摘录。 详细说明请参考FX3U用户手册[硬件篇]中的內容。
FX3U系列用户手册[硬件篇]	另外的 手册	关于FX3U可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX3UC(D、DS、DSS)系列硬件手册	随附	关于FX3UC(D、DS、DSS)可编程控制器主机的输入输出规格及接线、安装，从FX3UC系列用户手册[硬件篇]中摘录。 详细说明请参考FX3UC系列用户手册[硬件篇]。
FX3UC-32MT-LT-2硬件手册	随附	关于FX3UC-32MT-LT-2主机的输入输出规格及接线、安装，从FX3UC系列用户手册[硬件篇]中摘录。 详细说明请参考FX3UC系列用户手册[硬件篇]。
FX3UC-32MT-LT硬件手册	随附	关于FX3UC-32MT-LT主机的输入输出规格及接线、安装，从FX3UC用户手册[硬件篇]中摘录。 详细说明请参考FX3UC用户手册[硬件篇]中的內容。
FX3UC系列用户手册[硬件篇]	另外的 手册	关于FX3UC可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX3G系列硬件手册	随附	关于FX3G可编程控制器主机的输入输出规格及接线、安装，从FX3G系列用户手册[硬件篇]中摘录。 详细说明请参考FX3G系列用户手册[硬件篇]。
FX3G系列用户手册[硬件篇]	另外的 手册	关于FX3G可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX3GC系列硬件手册	随附	关于FX3GC可编程控制器主机的输入输出规格及接线、安装，从FX3GC系列用户手册[硬件篇]中摘录。 详细说明请参考FX3GC系列用户手册[硬件篇]。
FX3GC系列用户手册[硬件篇]	另外的 手册	关于FX3GC可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX3S系列硬件手册	随附	关于FX3S可编程控制器主机的输入输出规格及接线、安装，从FX3S系列用户手册[硬件篇]中摘录。 详细说明请参考FX3S系列用户手册[硬件篇]。

手册名称	随附 另外的 手册	内容
FX3S系列用户手册[硬件篇]	另外的 手册	关于FX3S可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
<b>编程</b>		
FX3S・FX3G・FX3GC・FX3U・FX3UC系列用户手册[模拟量控制篇]	另外的 手册	关于FX3UC-4AD、FX3U-4AD、FX3U-4DA型模拟量特殊功能模块、模拟量特殊适配器(FX3U-****-ADP)的详细内容。
FX系列用户手册[通信控制篇]	另外的 手册	关于N:N网络、并联链接、计算机链接和无协议通信(RS指令、FX2N-232IF)的详细内容。
FX3S・FX3G・FX3GC・FX3U・FX3UC系列用户手册[MODBUS通信篇]	另外的 手册	关于FX3S・FX3G・FX3GC・FX3U・FX3UC可编程控制器的MODBUS通信的详细内容。
FX3S・FX3G・FX3GC・FX3U・FX3UC系列用户手册[定位控制篇]	另外的 手册	关于FX3S・FX3G・FX3GC・FX3U・FX3UC可编程控制器内置的定位功能的详细内容。
FX3U-CF-ADP用户手册	另外的 手册	关于FX3U-CF-ADP型CF卡特殊适配器的详细内容。

**FX1S・FX1N・FX1NC可编程控制器  
FX2N・FX2NC可编程控制器[停产机型]**

手册名称	随附 另外的 手册	内容
<b>可编程控制器主机</b>		
FX1S系列使用手册	随附	关于FX1S可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX1N系列使用手册	随附	关于FX1N可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX2N系列使用手册	随附	关于FX2N可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX1NC系列使用手册	随附	关于FX1NC可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX2NC系列使用手册	随附	关于FX2NC可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
<b>编程</b>		
FX系列用户手册[通信控制篇]	另外的 手册	关于N:N网络、并联链接、计算机链接和无协议通信(RS指令、FX2N-232IF)的详细内容。

**FX0s・FX0・FX0N・FX2・FX2C可编程控制器[停产机型]**

手册名称	随附 另外的 手册	内容
<b>可编程控制器主机</b>		
FX0使用手册	随附	关于FX0可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX0s使用手册	随附	关于FX0s可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX0N使用手册	随附	关于FX0N可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX2系列使用手册	随附	关于FX2可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
FX2C系列使用手册	随附	关于FX2C可编程控制器主机的输入输出规格、接线、安装及维护等的硬件方面的详细内容。
<b>编程</b>		
FX系列用户手册[通信控制篇]	另外的 手册	关于N:N网络、并联链接、计算机链接和无协议通信(RS指令、FX2N-232IF)的详细内容。

## 关于手册中使用的总称·简称的记载

简称·总称	名称
<b>可编程控制器</b>	
FX3U系列, 或FX3U可编程控制器	FX3U系列 可编程控制器的总称
FX3UC系列, 或FX3UC可编程控制器	FX3UC系列 可编程控制器的总称
FX3G系列, 或FX3G可编程控制器	FX3G系列 可编程控制器的总称
FX3GC系列, 或FX3GC可编程控制器	FX3GC系列 可编程控制器的总称
FX3S系列, 或FX3S可编程控制器	FX3S系列 可编程控制器的总称
FX2N系列, 或FX2N可编程控制器	FX2N系列 可编程控制器的总称
FX2NC系列, 或FX2NC可编程控制器	FX2NC系列 可编程控制器的总称
FX1N系列, 或FX1N可编程控制器	FX1N系列 可编程控制器的总称
FX1NC系列, 或FX1NC可编程控制器	FX1NC系列 可编程控制器的总称
FX1S系列, 或FX1S可编程控制器	FX1S系列 可编程控制器的总称
FX2系列, 或FX2可编程控制器	FX2系列 可编程控制器的总称
FX2C系列, 或FX2C可编程控制器	FX2C系列 可编程控制器的总称
FX0N系列, 或FX0N可编程控制器	FX0N系列 可编程控制器的总称
FX0S系列, 或FX0S可编程控制器	FX0S系列 可编程控制器的总称
FX0系列, 或FX0可编程控制器	FX0系列 可编程控制器的总称
<b>特殊适配器</b>	
CF卡特殊适配器	CF卡特殊适配器的总称
CF-ADP	FX3U-CF-ADP
以太网适配器	FX3U-ENET-ADP的简称
<b>程序语言</b>	
ST	结构文本语言(Structured Text language)的简称
结构化梯形图	梯形图语言(Ladder Diagram language)的简称
FBD	功能模块表语言(Function Block Diagram language)的简称
<b>手册</b>	
Q/L/F结构化编程手册 (基础篇)	MELSEC-Q/L/F 结构化编程手册(基础篇)的简称
FX结构化编程手册 [软元件·公共说明篇]	FXCPU 结构化编程手册[软元件·公共说明篇]的简称
FX结构化编程手册 [顺控指令篇]	FXCPU 结构化编程手册[顺控指令篇]的简称
FX结构化编程手册 [应用函数篇]	FXCPU 结构化编程手册[应用函数篇]的简称
通信控制手册	FX系列 用户手册[通信控制篇]的简称
模拟量控制手册	FX3S·FX3G·FX3GC·FX3U·FX3UC系列 用户手册[模拟量控制篇]的简称
定位控制手册	FX3S·FX3G·FX3GC·FX3U·FX3UC系列 用户手册[定位控制篇]的简称

# 1. 概要

本手册对GX Works2中提供的用于结构化程序的应用函数进行了说明。  
关于软元件、参数、顺控指令，请参考其他手册。  
关于用于结构化程序的标签、数据类型、程序语言，请参考以下手册。

→ Q/L/F结构化编程手册(基础篇)

## 1.1 结构化程序的概要及编程语言

### 1.1.1 结构化程序的概要

结构化程序中可将多个程序(部品)汇总为一个程序。  
因此，可将机械处理的整体分为若干小的子进程，为各子进程分别编写程序，从而可有效提高大系统的程序效率。

#### 1. 结构化程序

所谓程序的结构化，是指将可编程控制器CPU的控制内容划分为若干小的处理单位(部品)以形成分层结构，从而构成程序的设计方法。结构化程序中，能够意识到顺控程序的结构化进行设计。

##### 程序分层方法

- 可先对程序概要进行研究，然后慢慢进行详细设计。
- 在分层中设计在最低位的程序，具有极单纯且较高独立性的特点。

##### 程序部品化的优点

- 由于每个部品的处理较为明确，因而可预见整体提高性能。
- 可实现分工作业、多人编程。
- 程序的再利用性高，从而提高开发效率。

#### 2. 程序再利用性的提高

部品化的程序可作为库进行保存。利用程序库便于共享程序资源，有助于提高程序的再利用性。

## 1.1.2 编程语言

在部品化的各程序中可使用的程序语言包括以下几种。

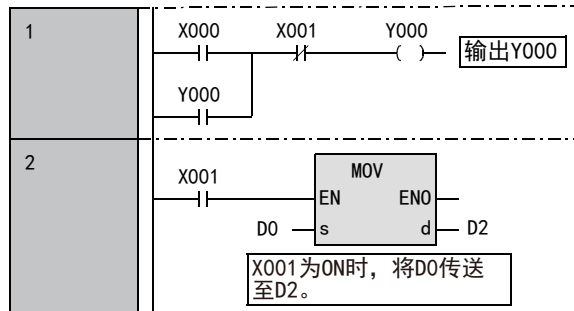
### 图形语言

#### 1. 结构化梯形图语言

基于继电器回路的设计技术创建的图形语言。

回路总是从左侧的母线开始。

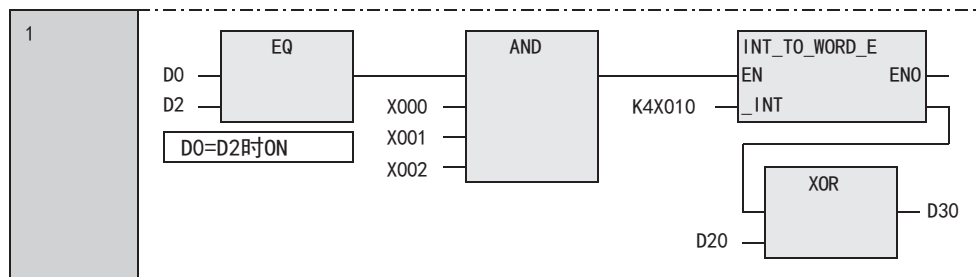
LD由触点、线圈、功能、功能模块组成。这些要素通过垂直线与水平线相连接。



#### 2. 功能模块表语言 (FBD [Function Block Diagram language])

FBD语言是一种从视觉上易理解的图形语言。

沿着数据以及信号的走向连接进行特定处理的部品(功能、功能模块)、变量部品、常数部品, 由此可以方便地创建程序, 提高程序的生产性。



### 文本语言

#### 1. ST语言 (结构文本语言)

ST语言可以通过语法进行控制, 例如与C语言等高级语言同样, 采用条件语句进行选择分支、利用循环语句进行重复等。这样, 便可以用简洁的方法书写清楚的程序。

```

Y000:=(X000 OR Y000) AND NOT X001;
IF X001 THEN
    D2:=D0;(*X001=ON时, 将D0代入(传送)至D2。*)
END_IF;
IF X002 THEN
    D4:=D4+1;(*X002=ON时, 将D4+1。*)
ELSE
    D6:=D6+1;(*X002=OFF时, 将D6+1。*)
END_IF;
    
```



## 1.2 可编程控制器与编程软件的版本

可编程控制器系列	软件包名称 (下半部分为软件型号)	GX Works2的版本
FX3U • FX3UC	GX Works2 (SW1DNC-GXW2-E)	Ver. 1.08J以上版本
FX3G		
FX2N • FX2NC		
FX1N • FX1NC		
FX1S		
FX2 • FX2C		
FX0N		
FX0 • FX0S		
FX3GC		Ver. 1.77F以上版本
FX3S		Ver. 1.492N以上版本

## 1.3 编程方面的基本注意事项

说明了在编程时需要注意的内容。  
关于结构化程序、程序语言的注意事项，请参考以下手册。

→ Q/L/F结构化编程手册(基础篇)

关于软元件、参数的具体内容及注意事项，请参考以下编程手册。

→ FX结构化编程手册[软元件·公共说明篇]

### 1.3.1 输入输出处理，响应延迟

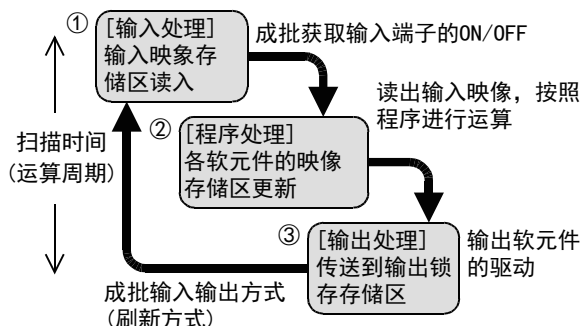
#### 1. 输入输出继电器的动作时序和响应延迟

FX可编程控制器中，重复执行①~③进行输入输出处理。

因此，在可编程控制器的控制中，除了输入滤波器和输出元器件的驱动时间以外，根据运算周期有时会出现响应延迟。

##### 以最新信息获取最新输入输出

在上述的运算周期中，想要获取输入的最新信息时或者要将运算结果立即输出时，可以使用「输入输出刷新用指令(REF)」。



#### 2. 不能获取宽度窄的输入脉冲

可编程控制器的输入ON时间或OFF时间，必须比可编程控制器的循环扫描时间+输入滤波器的时间更长。

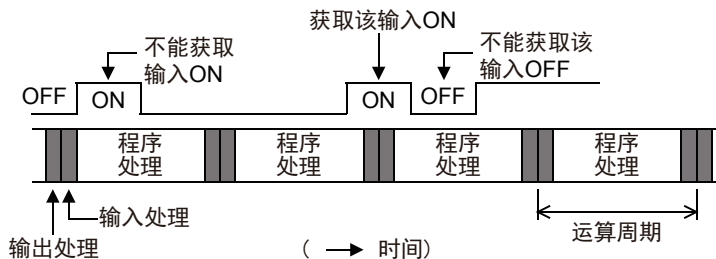
考虑输入滤波器10ms的响应延迟，10ms的循环扫描时间的话，ON时间、OFF时间各需要20ms。

因此，不可以处理 $1000 / (20+20) = 25\text{Hz}$ 以上的输入脉冲。但是，使用可编程控制器的特殊功能和指令时，可以改善这个情况。

##### 改善用的便利功能

使用下面的功能，可以获取比运算周期更短的脉冲。

- 高速计数器功能
- 输入中断功能
- 脉冲捕捉功能
- 输入滤波器值的调节功能



### 1.3.2 双重输出(双线圈)的动作及对策

对双重输出(双线圈)的动作及对策进行说明。

#### 1. 双重输出的动作

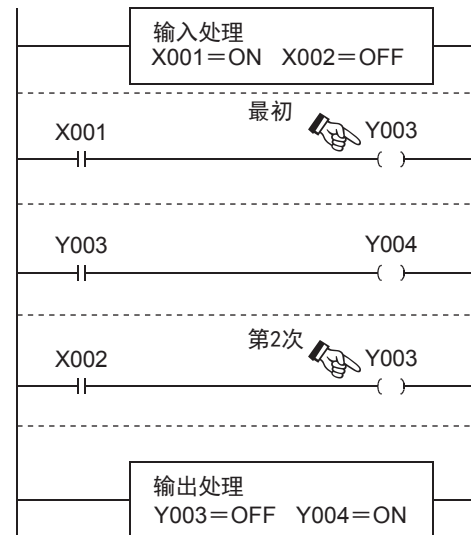
如果执行的其他程序模块、或程序模块中执行线圈(输出变量)的双重输出(双线圈)，则后侧的线圈优先动作。

如右图所示，请考虑一下同一线圈Y003在多个位置使用的情况。  
例如，X001=ON、X002=OFF。

最初的Y003由于X001为ON，所以其在映象存储区内为ON，输出Y004也为ON。

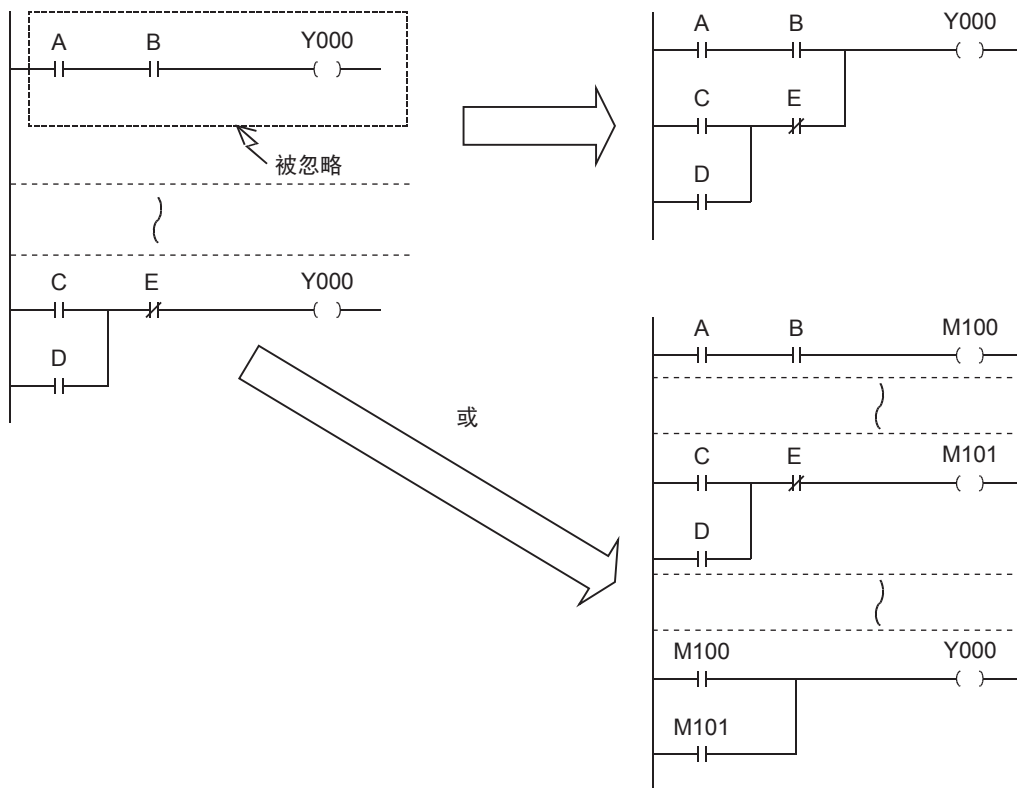
但是，第2次的Y003由于输入X002为OFF，所以其在映象存储区内被改写为OFF。

因此，实际的外部输出为Y003=OFF、Y004=ON。



#### 2. 双重输出的对策

双重输出(双线圈)，并非违背了程序中的输入(程序错误)，但是由于会使上述的动作变得复杂，因此请学习下面的例子后更改程序。



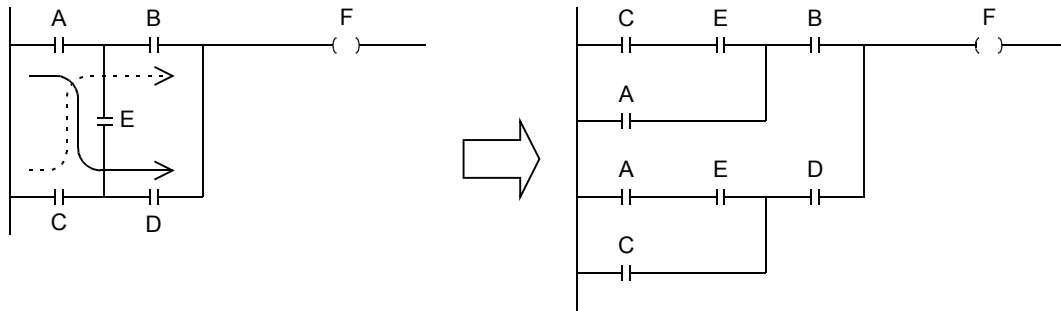
此外，还有其他编程方法，如使用SET、RST指令或跳转指令，以及使用步进梯形图指令STL、RET，在各状态中对同一个输出线圈编程。

此外，使用步进梯形图指令STL、RET时，请注意：如果其他的程序模块或STL到RET之外存在的输出线圈，在状态内也被编程时，会被视为双重线圈。

### 1.3.3 不能按结构化梯形图编程的回路图及对策

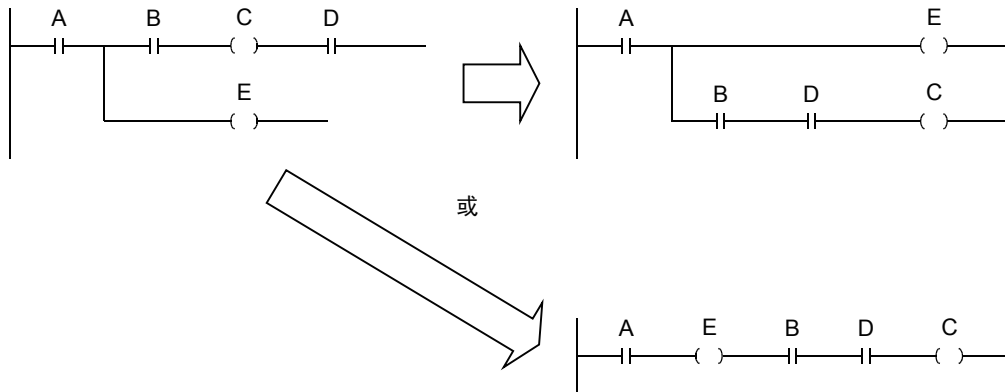
#### 1. 桥式电路

请按照右图所示，更改两个方向都有电流流过的回路。(将没有D时的回路和没有B时的回路进行并联的结果。)



#### 2. 线圈连接的位置

• 虽然在线圈右侧编写触点，但请务必在回路终点编写线圈(包括功能、功能模块在内)。



### 1.3.4 一般标志位的使用

在部分顺控指令中，标志位的动作如下所示。

- |                       |                 |                    |
|-----------------------|-----------------|--------------------|
| (例) M8020 : 零位标志位     | M8021 : 借位标志位   | M8022 : 进位标志位      |
| M8029 : 指令执行结束标志位     | M8090 : 块比较信号*1 | M8328 : 指令不执行标志位*1 |
| M8329 : 指令执行异常结束标志位*2 | M8304 : 零位标志位*2 | M8306 : 进位标志位*2    |

\*1. 仅对应FX3U・FX3UC可编程控制器。

\*2. 仅对应FX3S・FX3G・FX3GC・FX3U・FX3UC可编程控制器。

各种指令每次ON执行时，这些标志位为ON或OFF动作，但是OFF执行的时候和错误的时候不改变。

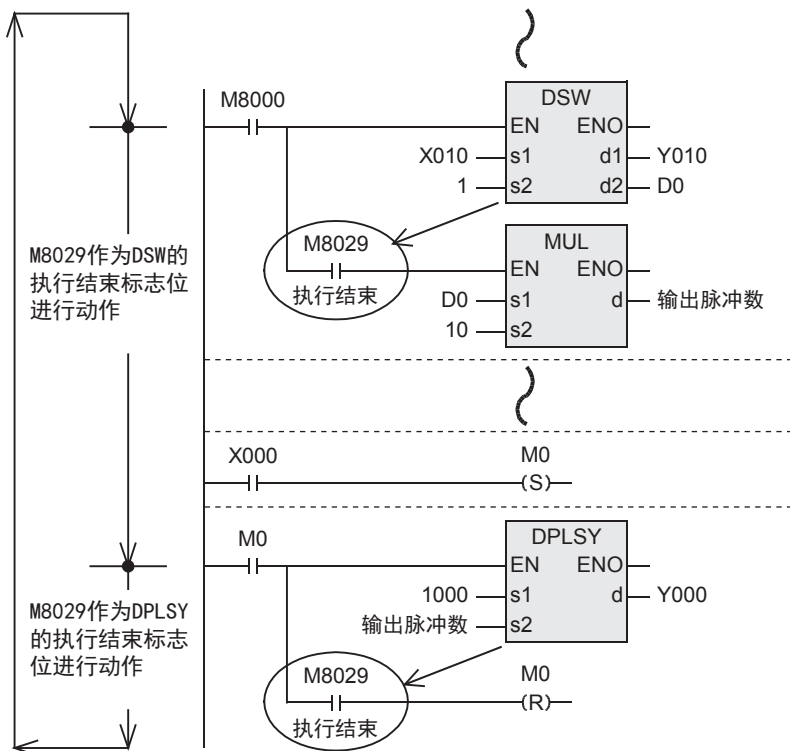
由于标志位在多数顺控指令中会变化，所以每次执行这些指令的时候呈ON/OFF变化。

参考下页的例子，请在对象顺控指令的正下方编写标志位触点。

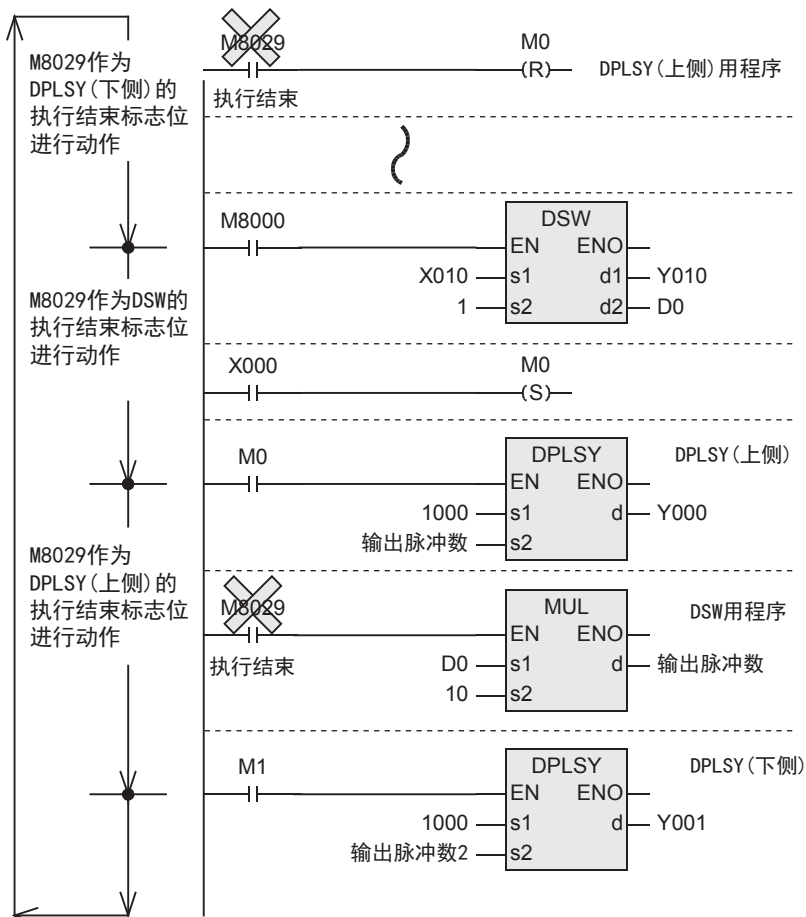
1. 多个标志位的程序 (指令执行结束标志位M8029的例子)

对使用相同标志位动作的顺控指令而言，将指令执行结束标志位M8029集中在一起编程时，除了难于判断哪个顺控指令的执行内容导致标志位控制执行，此外也可能不能正常读取各个顺控指令相对应的标志位。在顺控指令的正下方以外的位置中使用，请参考下页所述内容。

正确示例

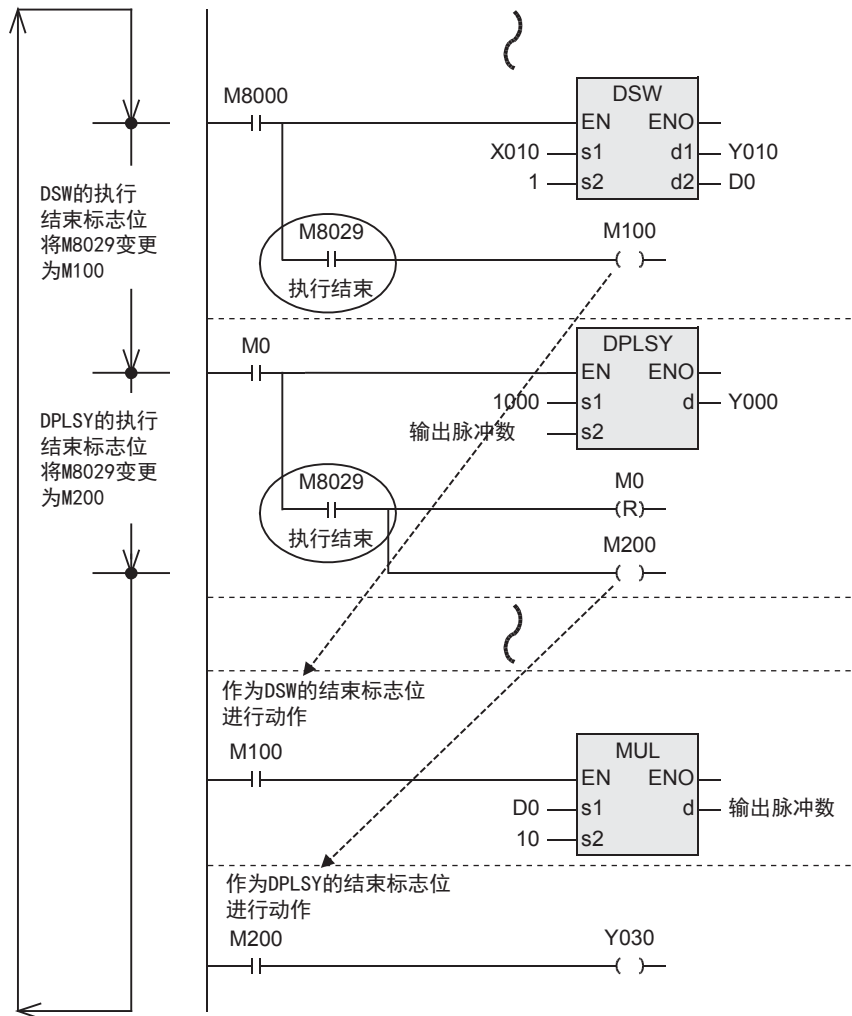


错误示例



## 2. 在顺控指令的正下方以外的位置中使用的方法介绍

对多个顺控指令编程后，一般标志位会根据各指令的ON执行进行变化。因此，想要在该顺控指令的正下方以外的位置中使用，先在顺控指令的正下方用一般标志位，ON/OFF其他的位软元件(变量)，然后将该触点(变量)作为指令触点使用。



### 1.3.5 运算错误标志位的使用

指令的结构和对对象软元件及其编号范围等出现错误，在运算执行过程中错误的时候，下面的标志位动作，并且错误的信息被记忆下来。

#### 1. 运算错误

错误标志位	错误代码 保存软元件	保存错误步的软元件	
		FX0s/FX0/FX0N/FX1s/FX1N/FX1NC/FX2/ FX2c/FX2N/FX2NC/FX3s/FX3G/FX3GC	FX3U/FX3UC
M8067	D8067	D8069*1	D8315, D8314

\*1. 在FX3U/FX3UC中发生错误的步在32767步以前的情况下，用D8069(16位)也能够确认错误步。

- 发生运算错误时会设置M8067，D8067中保存运算错误代码编号，错误步保存软元件(请参考上表)中保存错误步编号。
- 其他的步中出现新的错误时，这个指令的错误代码和步编号依次被更新。(错误解除时为OFF。)
- 可编程控制器从STOP→RUN时瞬间被清除，错误仍存在的情况下会再次置ON。

#### 2. 运算错误锁存

错误标志位	错误代码 保存软元件	保存错误步的软元件	
		FX0s/FX0/FX0N/FX1s/FX1N/FX1NC/FX2/ FX2c/FX2N/FX2NC/FX3s/FX3G/FX3GC	FX3U/FX3UC
M8068	—	D8068*2	D8313, D8312

\*2. 在FX3U/FX3UC中发生错误的步在32767步以前的情况下，用D8068(16位)也能够确认错误步。

- 发生运算错误时会设置M8068，在错误步保存软元件(请参考上表)中保存错误步编号。
- 即使其他的指令中出现新的错误，内容也不被更新，到执行强制复位或断电为止动作都被保持。

## 2. 函数、操作符一览

在本章中，介绍了编程中可以使用的函数、操作符的一览。

### 2.1 型转换函数

函数名	功能	对应的可编程控制器									参考
		FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2(C)	FX0N	FX0(S)	
BOOL_TO_INT(_E)	将位型数据转换为字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.1节
BOOL_TO_DINT(_E)	将位型数据转换为双字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.2节
BOOL_TO_STR(_E)	将位型数据转换为字符串型数据	○	×	×	×	×	×	×	×	×	5.3节
BOOL_TO_WORD(_E)	将位型数据转换为字[无符号]/位列[16位]型数据	○	○	○	○	○	○	○	○	○	5.4节
BOOL_TO_DWORD(_E)	将位型数据转换为双字[无符号]/位列[32位]型数据	○	○	○	○	○	○	○	○	○	5.5节
BOOL_TO_TIME(_E)	将位型数据转换为时间型数据	○	○	○	○	○	○	○	○	○	5.6节
INT_TO_DINT(_E)	将字[带符号]型数据转换为双字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.7节
DINT_TO_INT(_E)	将双字[带符号]型数据转换为字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.8节
INT_TO_BOOL(_E)	将字[带符号]型数据转换为位型数据	○	○	○	○	○	○	○	○	○	5.9节
DINT_TO_BOOL(_E)	将双字[带符号]型数据转换为位型数据	○	○	○	○	○	○	○	○	○	5.10节
INT_TO_REAL(_E)	将字[带符号]型数据转换为单精度实数型数据	○	*1	○	○	×	×	×	×	×	5.11节
DINT_TO_REAL(_E)	将双字[带符号]型数据转换为单精度实数型数据	○	*1	○	○	×	×	×	×	×	5.12节
INT_TO_STR(_E)	将字[带符号]型数据转换为字符串型数据	○	×	×	×	×	×	×	×	×	5.13节
DINT_TO_STR(_E)	将双字[带符号]型数据转换为字符串型数据	○	×	×	×	×	×	×	×	×	5.14节
INT_TO_WORD(_E)	将字[带符号]型数据转换为字[无符号]/位列[16位]型数据	○	○	○	○	○	○	○	○	○	5.15节
DINT_TO_WORD(_E)	将双字[带符号]型数据转换为字[无符号]/位列[16位]型数据	○	○	○	○	○	○	○	○	○	5.16节
INT_TO_DWORD(_E)	将字[带符号]型数据转换为双字[无符号]/位列[32位]型数据	○	○	○	○	○	○	○	○	○	5.17节
DINT_TO_DWORD(_E)	将双字[带符号]型数据转换为双字[无符号]/位列[32位]型数据	○	○	○	○	○	○	○	○	○	5.18节
INT_TO_BCD(_E)	将字[带符号]型数据转换为BCD型数据	○	○	○	○	○	○	○	○	○	5.19节
DINT_TO_BCD(_E)	将双字[带符号]型数据转换为BCD型数据	○	○	○	○	○	○	○	○	○	5.20节

\*1. FX3G可编程控制器V1.10以上版本支持函数。

函数名	功能	对应的可编程控制器								参考	
		FX3U(②)	FX3G(②)	FX3S	FX2N(②)	FX1N(②)	FX1S	FX2(②)	FX0N		FX0(S)
INT_TO_TIME(_E)	将字[带符号]型数据转换为时间型数据	○	○	○	○	○	○	○	○	○	5.21节
DINT_TO_TIME(_E)	将双字[带符号]型数据转换为时间型数据	○	○	○	○	○	○	○	○	○	5.22节
REAL_TO_INT(_E)	将单精度实数型数据转换为字[带符号]型数据	○	*1	○	○	×	×	×	×	×	5.23节
REAL_TO_DINT(_E)	将单精度实数型数据转换为双字[带符号]型数据	○	*1	○	○	×	×	×	×	×	5.24节
REAL_TO_STR(_E)	将单精度实数型数据转换为字符串型数据	○	×	×	×	×	×	×	×	×	5.25节
WORD_TO_BOOL(_E)	将字[无符号]/位列[16位]型数据转换为位型数据	○	○	○	○	○	○	○	○	○	5.26节
DWORD_TO_BOOL(_E)	将双字[无符号]/位列[32位]型数据转换为位型数据	○	○	○	○	○	○	○	○	○	5.27节
WORD_TO_INT(_E)	将字[无符号]/位列[16位]型数据转换为字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.28节
WORD_TO_DINT(_E)	将字[无符号]/位列[16位]型数据转换为双字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.29节
DWORD_TO_INT(_E)	将双字[无符号]/位列[32位]型数据转换为字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.30节
DWORD_TO_DINT(_E)	将双字[无符号]/位列[32位]型数据转换为双字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.31节
WORD_TO_DWORD(_E)	将字[无符号]/位列[16位]型数据转换为双字[无符号]/位列[32位]型数据	○	○	○	○	○	○	○	○	○	5.32节
DWORD_TO_WORD(_E)	将双字[无符号]/位列[32位]型数据转换为字[无符号]/位列[16位]型数据	○	○	○	○	○	○	○	○	○	5.33节
WORD_TO_TIME(_E)	将字[无符号]/位列[16位]型数据转换为时间型数据	○	○	○	○	○	○	○	○	○	5.34节
DWORD_TO_TIME(_E)	将双字[无符号]/位列[32位]型数据转换为时间型数据	○	○	○	○	○	○	○	○	○	5.35节
STR_TO_BOOL(_E)	将字符串型数据转换为位型数据	○	×	×	×	×	×	×	×	×	5.36节
STR_TO_INT(_E)	将字符串型数据转换为字[带符号]型数据	○	×	×	×	×	×	×	×	×	5.37节
STR_TO_DINT(_E)	将字符串型数据转换为双字[带符号]型数据	○	×	×	×	×	×	×	×	×	5.38节
STR_TO_REAL(_E)	将字符串型数据转换为单精度实数型数据	○	×	×	×	×	×	×	×	×	5.39节
STR_TO_TIME(_E)	将字符串型数据转换为时间型数据	○	×	×	×	×	×	×	×	×	5.40节
BCD_TO_INT(_E)	将BCD数据转换为字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.41节
BCD_TO_DINT(_E)	将BCD数据转换为双字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.42节
TIME_TO_BOOL(_E)	将时间型数据转换为位型数据	○	○	○	○	○	○	○	○	○	5.43节
TIME_TO_INT(_E)	将时间型数据转换为字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.44节

\*1. FX3G可编程控制器V1.10以上版本支持函数。



函数名	功能	对应的可编程控制器									参考
		FX3U(②)	FX3S(②)	FX3S	FX2N(②)	FX1N(②)	FX1S	FX2(②)	FX0N	FX0(S)	
TIME_TO_DINT(_E)	将时间型数据转换为双字[带符号]型数据	○	○	○	○	○	○	○	○	○	5.45节
TIME_TO_STR(_E)	将时间型数据转换为字符串型数据	○	×	×	×	×	×	×	×	×	5.46节
TIME_TO_WORD(_E)	将时间型数据转换为字[无符号]/位列[16位]型数据	○	○	○	○	○	○	○	○	○	5.47节
TIME_TO_DWORD(_E)	将时间型数据转换为双字[无符号]/位列[32位]型数据	○	○	○	○	○	○	○	○	○	5.48节
BITARR_TO_INT(_E)	将位阵列中的指定位数转换为字[带符号]型、字[无符号]/位列[16位]型数据	○	○	○	○	○	○	○	○	○	5.49节
BITARR_TO_DINT(_E)	将位阵列中的指定位数转换为双字[带符号]型、双字[无符号]/位列[32位]型数据	○	○	○	○	○	○	○	○	○	5.50节
INT_TO_BITARR(_E)	将字[带符号]型、字[无符号]/位列[16位]型数据的低n位输出到位阵列	○	○	○	○	○	○	○	○	○	5.51节
DINT_TO_BITARR(_E)	将双字[带符号]型、双字[无符号]/位列[32位]型数据的低n位输出到位阵列	○	○	○	○	○	○	○	○	○	5.52节
CPY_BITARR(_E)	按指定位大小复制位阵列	○	○	○	○	○	○	○	○	○	5.53节
GET_BIT_OF_INT(_E)	读出字[带符号]型数据的指定位的值	○	×	×	×	×	×	×	×	×	5.54节
SET_BIT_OF_INT(_E)	将值写入到字[带符号]型数据的指定位中	○	×	×	×	×	×	×	×	×	5.55节
CPY_BIT_OF_INT(_E)	将字[带符号]型数据的指定位复制到其他字[带符号]型数据的指定位中	○	×	×	×	×	×	×	×	×	5.56节
GET_BOOL_ADDR	将起始数据作为位型数据进行输出	○	○	○	○	○	○	○	○	○	5.57节
GET_INT_ADDR	将起始数据作为字[带符号]型数据进行输出	○	○	○	○	○	○	○	○	○	5.58节
GET_WORD_ADDR	将起始数据作为字[无符号]/位列[16位]型数据进行输出	○	○	○	○	○	○	○	○	○	5.59节

1 概要

2 函数、操作符一览

3 函数的构成

4 函数的阅读方法

5 应用函数  
(型转换函数)

6 应用函数  
(单数值变量  
函数)

7 应用函数  
(算术运算函数)

8 应用函数  
(移位函数)

9 应用函数  
(位型布尔函数)

10 应用函数  
(选择函数)

## 2.2 单数值变量函数

函数名	功能	对应的可编程控制器									参考
		FX3U(Ⓞ)	FX3G(Ⓞ)	FX3S	FX2N(Ⓞ)	FX1N(Ⓞ)	FX1S	FX2(Ⓞ)	FX0N	FX0(S)	
ABS(_E)	绝对值运算	○	○	○	○	○	○	○	○	○	6.1节

## 2.3 算术运算函数

函数名	功能	对应的可编程控制器									参考
		FX3U(Ⓞ)	FX3G(Ⓞ)	FX3S	FX2N(Ⓞ)	FX1N(Ⓞ)	FX1S	FX2(Ⓞ)	FX0N	FX0(S)	
ADD_E	数据的加法运算(PIN数可变)	○	○	○	○	○	○	○	○	○	7.1节
SUB_E	数据的减法运算	○	○	○	○	○	○	○	○	○	7.2节
MUL_E	数据的乘法运算(PIN数可变)	○	○	○	○	○	○	○	○	○	7.3节
DIV_E	数据的除法运算(输出商)	○	○	○	○	○	○	○	○	○	7.4节
MOD(_E)	数据的除法运算(输出余数)	○	○	○	○	○	○	○	○	○	7.5节
EXPT(_E)	数据的幂乘	○	×	×	×	×	×	×	×	×	7.6节
MOVE(_E)	数据的传送	○	○	○	○	○	○	○	○	○	7.7节

## 2.4 位移位函数

函数名	功能	对应的可编程控制器									参考
		FX3U(Ⓞ)	FX3G(Ⓞ)	FX3S	FX2N(Ⓞ)	FX1N(Ⓞ)	FX1S	FX2(Ⓞ)	FX0N	FX0(S)	
SHL(_E)	左移	○	○	○	○	○	○	○	○	○	8.1节
SHR(_E)	右移	○	○	○	○	○	○	○	○	○	8.2节

## 2.5 位型布尔函数

函数名	功能	对应的可编程控制器									参考
		FX3U(Ⓞ)	FX3G(Ⓞ)	FX3S	FX2N(Ⓞ)	FX1N(Ⓞ)	FX1S	FX2(Ⓞ)	FX0N	FX0(S)	
AND_E	逻辑与(PIN数可变)	○	○	○	○	○	○	○	○	○	9.1节
OR_E	逻辑或(PIN数可变)	○	○	○	○	○	○	○	○	○	9.2节
XOR_E	逻辑异或(PIN数可变)	○	○	○	○	○	○	○	○	○	9.3节
NOT_E	逻辑非	○	○	○	○	○	○	○	○	○	9.4节

## 2.6 选择函数

函数名	功能	对应的可编程控制器									参考
		FX3U(□)	FX3G(□)	FX3S	FX2N(□)	FX1N(□)	FX1S	FX2(□)	FX0N	FX0(S)	
SEL(_E)	根据输入条件选择数据	○	○	○	○	○	○	○	○	○	10.1节
MAXIMUM(_E)	从数据中选择最大值(PIN数可变)	○	○	○	○	○	○	○	○	○	10.2节
MINIMUM(_E)	从数据中选择最小值(PIN数可变)	○	○	○	○	○	○	○	○	○	10.3节
LIMITATION(_E)	判断数据是否在上下限值的范围内	○	○	○	○	○	○	○	○	○	10.4节
MUX(_E)	选择数据后输出(PIN数可变)	○	○	○	○	○	○	○	○	○	10.5节

## 2.7 比较函数

函数名	功能	对应的可编程控制器									参考
		FX3U(□)	FX3G(□)	FX3S	FX2N(□)	FX1N(□)	FX1S	FX2(□)	FX0N	FX0(S)	
GT_E	是否>(大于)的比较(PIN数可变)	○	○	○	○	○	○	○	○	○	11.1节
GE_E	是否≥(大于或等于)的比较(PIN数可变)	○	○	○	○	○	○	○	○	○	11.2节
EQ_E	是否=(等于)的比较(PIN数可变)	○	○	○	○	○	○	○	○	○	11.3节
LE_E	是否≤(小于或等于)的比较(PIN数可变)	○	○	○	○	○	○	○	○	○	11.4节
LT_E	是否<(小于)的比较(PIN数可变)	○	○	○	○	○	○	○	○	○	11.5节
NE_E	是否≠(不等于)的比较	○	○	○	○	○	○	○	○	○	11.6节

## 2.8 字符串函数

函数名	功能	对应的可编程控制器									参考
		FX3U(□)	FX3G(□)	FX3S	FX2N(□)	FX1N(□)	FX1S	FX2(□)	FX0N	FX0(S)	
MID(_E)	从指定位置取得字符串	○	×	×	×	×	×	×	×	×	12.1节
CONCAT(_E)	字符串的连接(PIN数可变)	○	×	×	×	×	×	×	×	×	12.2节
INSERT(_E)	字符串的插入	○	×	×	×	×	×	×	×	×	12.3节
DELETE(_E)	字符串的删除	○	×	×	×	×	×	×	×	×	12.4节
REPLACE(_E)	字符串的替换	○	×	×	×	×	×	×	×	×	12.5节
FIND(_E)	字符串的检索	○	×	×	×	×	×	×	×	×	12.6节

## 2.9 时间数据型函数

函数名	功能	对应的可编程控制器									参考
		FX3U(□)	FX3G(□)	FX3S	FX2N(□)	FX1N(□)	FX1S	FX2(□)	FX0N	FX0(S)	
ADD_TIME(_E)	时间的加法运算	○	○	○	○	○	○	○	○	○	13.1节
SUB_TIME(_E)	时间的减法运算	○	○	○	○	○	○	○	○	○	13.2节
MUL_TIME(_E)	时间的乘法运算	○	○	○	○	○	○	○	○	○	13.3节
DIV_TIME(_E)	时间的除法运算	○	○	○	○	○	○	○	○	○	13.4节

## 2.10 功能模块

函数名	功能	对应的可编程控制器									参考
		FX3U(□)	FX3G(□)	FX3S	FX2N(□)	FX1N(□)	FX1S	FX2(□)	FX0N	FX0(S)	
R_TRIG(_E)	检测出信号的上升沿后输出脉冲信号	○	○	○	○	○	○	○	○	○	14.1节
F_TRIG(_E)	检测出信号的下降沿后输出脉冲信号	○	○	○	○	○	○	○	○	○	14.2节
CTU(_E)	对信号的上升沿次数进行增计数	○	○	○	○	○	○	○	×	×	14.3节
CTD(_E)	对信号的上升沿次数进行减计数	○	○	○	○	○	○	○	×	×	14.4节
CTUD(_E)	对信号的上升沿次数进行增/减计数	○	○	○	○	○	○	○	×	×	14.5节
TP(_E)	在指定时间段将信号置为ON	○	○	○	○	○	○	○	×	×	14.6节
TP_10(_E)		○	○	×	○	○	×	○	×	×	
TON(_E)	经过指定时间后将信号置为ON	○	○	○	○	○	○	○	×	×	14.7节
TON_10(_E)		○	○	×	○	○	×	○	×	×	
TOF(_E)	输入信号变为OFF后，经过指定时间后将输出信号置为OFF	○	○	○	○	○	○	○	×	×	14.8节
TOF_10(_E)		○	○	×	○	○	×	○	×	×	
COUNTER_FB_M	计数器的驱动	○	○	○	○	○	○	○	○	○	14.9节
TIMER_10_FB_M	10ms定时器的驱动	○	○	×	○	○	×	○	×	×	14.10节
TIMER_CONT_FB_M	累计定时器的驱动	○	○	○	○	○	×	○	×	×	14.11节
TIMER_100_FB_M	100ms定时器的驱动	○	○	○	○	○	○	○	○	○	14.12节

## 2.11 操作符

### 2.11.1 算术运算

操作符名		功能	对应的可编程控制器									参考
结构化梯形图/FBD	ST		FX3U(□)	FX3G(□)	FX3S	FX2N(□)	FX1N(□)	FX1S	FX2(□)	FX0N	FX0(S)	
ADD	+	数据的加法运算(PIN数可变)	○	○	○	○	○	○	○	○	○	15.1节
SUB	-	数据的减法运算	○	○	○	○	○	○	○	○	○	15.2节
MUL	*	数据的乘法运算(PIN数可变)	○	○	○	○	○	○	○	○	○	15.3节
DIV	/	数据的除法运算(输出商)	○	○	○	○	○	○	○	○	○	15.4节
-	MOD	数据的除法运算(输出余数)	○	○	○	○	○	○	○	○	○	15.5节
-	**	数据的幂乘	○	×	×	×	×	×	×	×	×	15.6节

### 2.11.2 位型布尔

操作符名		功能	对应的可编程控制器									参考
结构化梯形图/FBD	ST		FX3U(□)	FX3G(□)	FX3S	FX2N(□)	FX1N(□)	FX1S	FX2(□)	FX0N	FX0(S)	
AND	& AND	逻辑与(PIN数可变)	○	○	○	○	○	○	○	○	○	15.7节
OR	OR	逻辑或(PIN数可变)	○	○	○	○	○	○	○	○	○	15.8节
XOR	XOR	逻辑异或(PIN数可变)	○	○	○	○	○	○	○	○	○	15.9节
-	NOT	逻辑非	○	○	○	○	○	○	○	○	○	15.10节

### 2.11.3 比较

操作符名		功能	对应的可编程控制器									参考
结构化梯形图/FBD	ST		FX3U(G)	FX3G(G)	FX3S	FX2N(G)	FX1N(G)	FX1S	FX2(G)	FX0N	FX0(S)	
GT	>	数据比较(是否大于)(PIN数可变)	○	○	○	○	○	○	○	○	○	15.11节
GE	≥	数据比较(是否大于或等于)(PIN数可变)	○	○	○	○	○	○	○	○	○	15.12节
EQ	=	数据比较(是否等于)(PIN数可变)	○	○	○	○	○	○	○	○	○	15.13节
LE	≤	数据比较(是否小于或等于)(PIN数可变)	○	○	○	○	○	○	○	○	○	15.14节
LT	<	数据比较(是否小于)(PIN数可变)	○	○	○	○	○	○	○	○	○	15.15节
NE	≠	数据比较(是否不等于)	○	○	○	○	○	○	○	○	○	15.16节

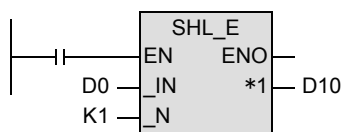
## 3. 函数的构成

本章对应用函数的构成进行说明。

### 3.1 应用函数的表示和执行形式

#### 应用函数与参数

- 对函数所命名的函数名称中要体现其处理内容。  
例如，命名为SHL(位左移)的函数名。
- 函数由该函数中使用的表示输入输出数据的参数构成。



- $\_IN$  (s): 不会因通过执行函数而使内容变化的参数称为源操作数，以该符号表示。
- \*1 (d): 通过执行函数，其内容发生变化的参数称为目标操作数，以该符号表示。
- K1 (n): 不符合源操作数，也不符合目标操作数的参数以m和n表示。

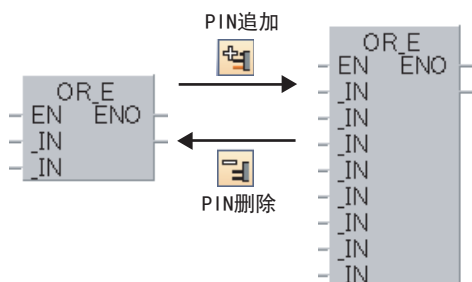
#### 参数的对象软元件

- 参数通过输入变量(标签或软元件)指定对象。
- 有时会处理X、Y、M、S等的位软元件。
- 有时组合位软元件，显示为KnX、KnY、KnM、KnS，将此可以作为数值数据处理。  
→ FX结构化编程手册[软元件·公共说明篇]
- 有时候使用数据寄存器D或定时器T、计数器C的当前值寄存器。
- 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位数据长度的软元件，因此可以直接指定。指定软元件时，使用全局标签。16位数据寄存器D，处理32位数据的时候，会组合连续2点的数据寄存器。  
例如，在标签中定义了数据寄存器D0作为32位指令的参数时，就处理(D1、D0)的32位数据。将(D1为高16位、D0为低16位)  
T、C的当前值寄存器作为一般的数据寄存器使用时，处理相同。

#### 更改参数(PIN数)时

- 根据函数，可在2~28范围内更改源操作数的数量。  
可更改的函数，在函数一览表中标有“(PIN数可变)”。
- 更改PIN数时

→ 2章 函数、操作符一览



## 3.2 标签

### 标签的种类

标签包括全局标签和局部标签两种。

- 全局标签可在程序模块与功能模块中使用。
- 局部标签是仅可用于已声明的程序部品的标签。

### 标签的等级

标签的等级表示标签能够用于哪种程序部品及如何使用。  
下表为标签的等级。

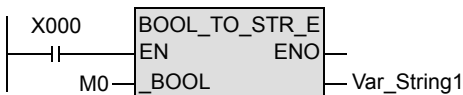
等级	内容	可使用的程序部品		
		程序块	功能	功能块
VAR_GLOBAL	可在所有程序部品中使用的公共标签。	○	×	○
VAR_GLOBAL_CONSTANT	可在所有程序部品中使用的公共常数。	○	×	○
VAR	可在已声明的程序部品范围内使用的标签。无法用于其它程序部品。	○	○	○
VAR_CONSTANT	可在声明了的程序部品范围内使用的常数。无法用于其它程序部品。	○	○	○
VAR_INPUT	接受值的标签，在程序部品内无法更改。	×	○	○
VAR_OUTPUT	由功能模块输出的标签。	×	×	○
VAR_IN_OUT	接受值，由程序部品输出的局部标签。 可在程序部品内更改。	×	×	○

### 标签的定义

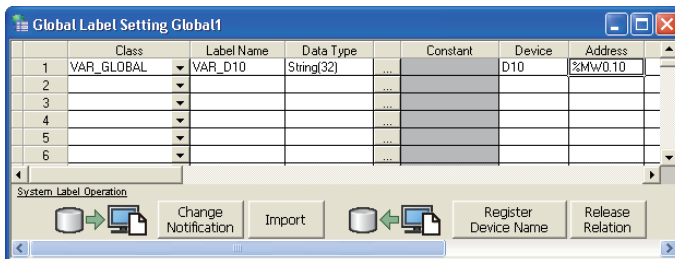
使用标签时，需要定义标签。变换(编译)没有定义标签的程序时会出现错误。

- 在全局标签的定义中，使标签名与等级、数据类型及软元件建立关联。
- 在局部标签的定义中，设定标签名与等级、数据类型。  
在局部标签中，用户不需要指定软元件。在编译时会自动分配至软元件。

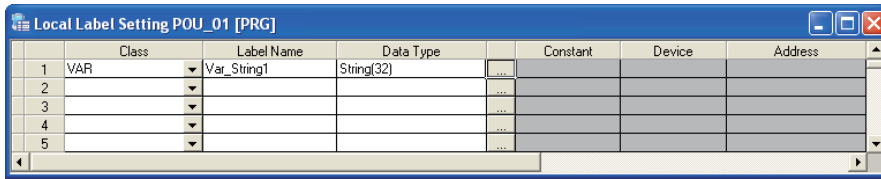
以下是BOOL\_TO\_STR\_E函数的标签Var\_String1的设定示例。



- 作为全局标签使用时。  
设定等级、标签名、数据类型、软元件或地址。



- 作为局部标签使用时。  
设定等级、标签名、数据类型。



### 常数的记载方法

标签上设定常数时的记载方法如下所述。

常数的种类	记载方法	例如
位	输入FALSE或TRUE。或输入0或者1。	TRUE, FALSE
2进制数	2进制数前加上“2#”。	2#0010, 2#01101010
8进制数	8进制数前加上“8#”。	8#0, 8#337
10进制数	10进制数直接输入。或者,在10进制数前加上“K”。	123, K123
16进制数	16进制数前加上“16#”。或者加上“H”。	16#FF, HFF
实数	实数直接输入。或者在实数前加上“E”。	2.34, E2.34
字符串	用单引号(')或者双引号(")括起来标记字符串。	'ABC'、"ABC"

### 数据类型

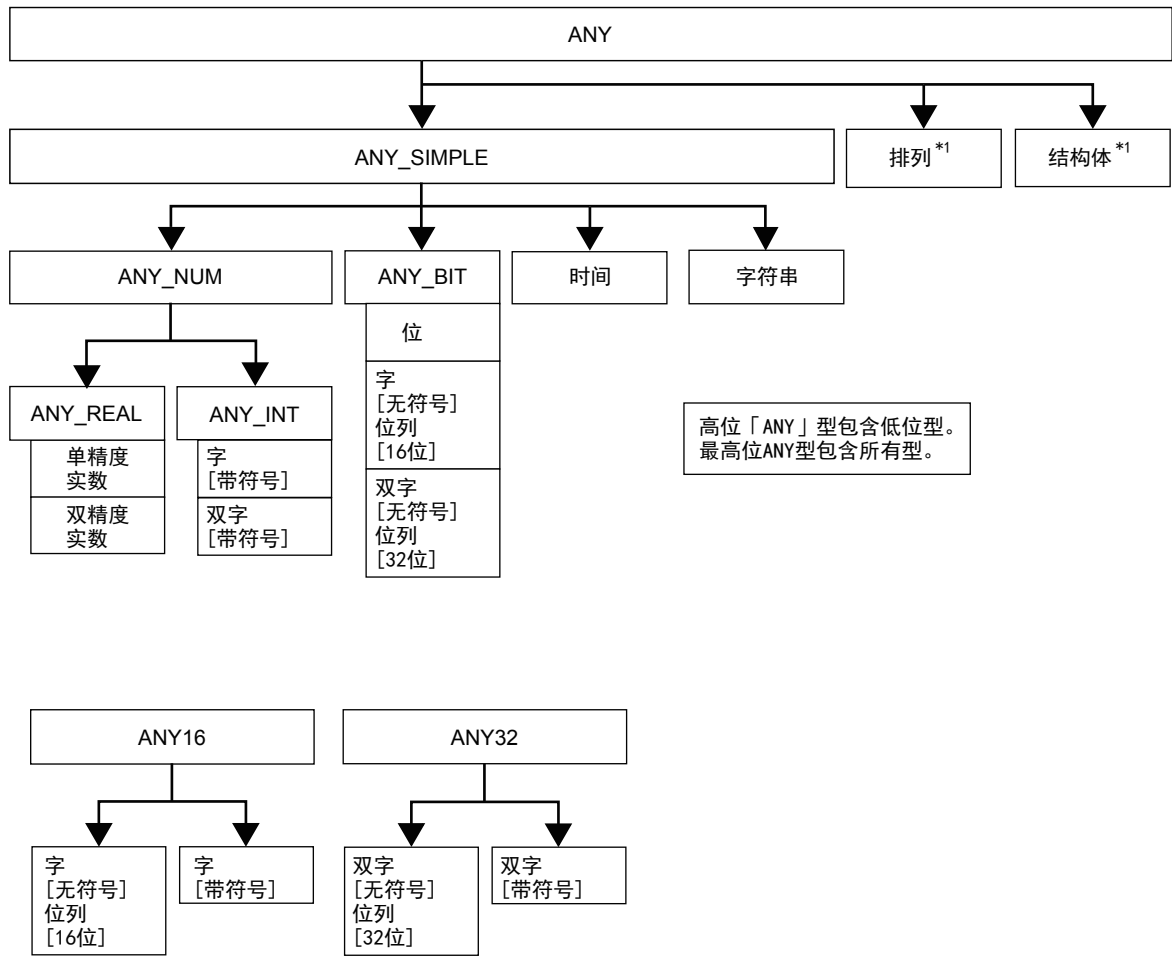
标签的数据类型,包括基本数据类型和普通数据类型。

- 以下所示为基本数据类型的一览。

数据类型	内容	值的范围	位长
位	布尔	0 (FALSE), 1 (TRUE)	1位
字[带符号]	整数	-32768~32767	16位
双字[带符号]	双精度整数	-2147483648~2147483647	32位
字[无符号]/位列 [16位]	16位列	0~65535	16位
双字[无符号]/位列 [32位]	32位列	0~4294967295	32位
单精度实数	实数	$E \pm 1.75495^{-38} \sim E \pm 3.402823^{+38}$ (有效位数6位)	32位
双精度实数	双精度实数	$E \pm 2.225073858507202^{-308} \sim$ $E \pm 1.797693134862315^{+308}$	64位
字符串	字符串	最多50个字符	可变
时间	时间值	T#-24d-0h31m23s648.00ms~ T#24d20h31m23s647.00ms	32位



- 普通数据类型，是汇总了多个基本数据类型的标签的数据类型。  
数据类型名以“ANY”开头。



\*1 详细内容，请参考以下手册。  
→Q/L/F结构化编程手册(基础篇)

1 概要

2 函数、操作符 一览

3 函数的构成

4 函数的阅读方法

5 应用函数 (型转换函数)

6 应用函数 (单数值变量函数)

7 应用函数 (算术运算函数)

8 应用函数 (位移位函数)

9 应用函数 (位型布尔函数)

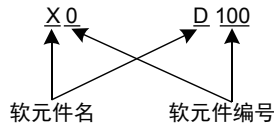
10 应用函数 (选择函数)

### 3.3 软元件和地址

软元件的记载方法，包括软元件和地址两类。

#### 软元件

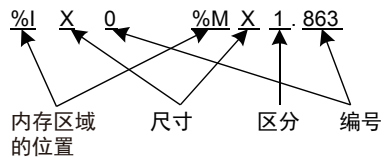
软元件是用软元件名与软元件编号记载的方法。



#### 地址

地址是用IEC61131-3中定义的记载方法。  
基于IEC61131-3记载方法如下所示。

起始	第1个字符: 位置	第2个字符: 大小		第3个字符以后: 区分	编号	
%	I	输入	(省略)	位	用于详细分类的数字。与后续编号间以“.”“(句号) 隔开进行表示。有时会省略。	对应软元件编号的号码。(以10进制数记载)
	Q	输出	X	位		
	M	内部	W	字(16位)		
			D	双字(32位)		
			L	长字(64位)		



- 内存区域的位置  
分配数据的内存区域的位置，大体分为输入、输出、内部这三类。  
X(X软元件) : I(输入)  
Y(Y软元件) : Q(输出)  
上述以外的软元件 : M(内部)
- 大小  
与软元件(MELSEC的记载方法)相应的记载方法的原则如下所示。  
位软元件: X(位)  
字软元件: W(字), D(双字)
- 区分  
是对仅以上述位置和大小无法区别的软元件划分出的小分类。  
软元件的X、Y未区分。  
与软元件记载相对应的记载，请参考以下内容。

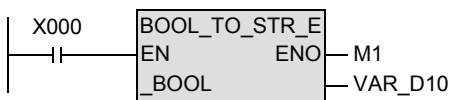
→ 附录A

### 3.4 EN和ENO

带EN的指令可以控制执行。

- EN表示输入指令的执行条件。
- ENO表示输出指令的执行状态。
- EN状态下ENO和运算结果的内容如下所示。

EN	ENO	运算结果
TRUE (执行运算)	TRUE (无运算错误)	运算输出值
	FALSE (有运算错误)	不定值
FALSE (停止运算)	FALSE	不定值



为上述函数时，X000仅在TRUE时  
执行函数BOOL\_TO\_STR\_E。  
如果正常执行，则向M1输出TRUE。

## 4. 函数的阅读方法

函数的说明页如下所示。

1) → 11.2 GE\_E / 比较

2) →

FX3u(C)	FX3g(C)	FX3s	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

概要  
进行是否≥(大于、或等于)的比较。

3) → 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
GE_E		GE_E(EN, _IN, _IN, 输出标签); 例: GE_E(X000, D0, D10, M0);

\*1. 变为输出变量。

4) → 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN (S1 ~ S28)	进行比较的数据, 或是保存数据的软元件	ANY_SIMPLE
输出变量	ENO	执行状态	位
	*1 (D)	保存比较结果的软元件	位

在函数说明中, 输入输出变量通过( )中的描述进行记载。

5) → 功能和动作说明

- 对(S1) ~ (S28)中指定的软元件的内容进行比较运算, 将运算结果按照位型输出到(D)中指定的软元件中。进行[(S1) ≥ (S2)] & [(S2) ≥ (S3)] & . . . . . & [(Sn-1) ≥ (Sn)]的比较。
  - 全部(Sn-1) ≥ (Sn)时, 输出TRUE。
  - 任意为(Sn-1) < (Sn)时, 输出FALSE。
- (S)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

6) → 注意要点

- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在(S)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

7) → 程序举例

对(S1)、(S2)中指定的软元件的内容进行比较运算, 将运算结果输出到(D)中指定的软元件中的程序。

<p>[结构化梯形图/FBD]</p>	<p>[ST]</p> <pre>g_bool3:=GE_E(g_bool1,g_int1,g_int2,g_bool2);</pre>
---------------------	--

\*上述内容仅用于说明, 和实际页面可能有所不同。

- 1) 表示章/节/项 编号、指令名。
- 2) 表示支持函数的可编程控制器。

项目	内容
○	从首批产品开始支持。
△	支持情况视版本而异。 在“注意要点”项目中会对对应版本进行说明。
×	该系列不支持指令。

- 3) 表示各函数的描述。

项目	内容
结构化梯形图/FBD	表示代表结构化梯形图语言的函数描述。
ST	表示ST语言的指令描述。

- 4) 表示函数的输入变量、输出变量的名称、各变量的内容与数据类型。  
关于数据类型的详细内容请参考以下手册。

→ Q/L/F结构化编程手册 (基础篇)

- 5) 功能和动作说明  
对函数所发挥的作用进行说明。  
在说明中，使用结构化梯形图语言作为代表语言。
- 6) 注意要点  
总结了使用函数时需要注意的内容。
- 7) 程序举例  
用各语言对程序举例进行说明。  
结构化梯形图/FBD语言的程序举例中，使用结构化梯形图语言作为代表语言。

## 5. 应用函数(型转换函数)

函数名	功能	参考
BOOL_TO_INT(_E)	位型 → 字[带符号]型转换	5.1节
BOOL_TO_DINT(_E)	位型 → 双字[带符号]型转换	5.2节
BOOL_TO_STR(_E)	位型 → 字符串型转换	5.3节
BOOL_TO_WORD(_E)	位型 → 字[无符号] / 位列[16位]型转换	5.4节
BOOL_TO_DWORD(_E)	位型 → 双字[无符号] / 位列[32位]型转换	5.5节
BOOL_TO_TIME(_E)	位型 → 时间型转换	5.6节
INT_TO_DINT(_E)	字[带符号]型 → 双字[带符号]型转换	5.7节
DINT_TO_INT(_E)	双字[带符号]型 → 字[带符号]型转换	5.8节
INT_TO_BOOL(_E)	字[带符号]型 → 位型转换	5.9节
DINT_TO_BOOL(_E)	双字[带符号]型 → 位型转换	5.10节
INT_TO_REAL(_E)	字[带符号]型 → 单精度实数型转换	5.11节
DINT_TO_REAL(_E)	双字[带符号]型 → 单精度实数型转换	5.12节
INT_TO_STR(_E)	字[带符号]型 → 字符串型转换	5.13节
DINT_TO_STR(_E)	双字[带符号]型 → 字符串型转换	5.14节
INT_TO_WORD(_E)	字[带符号]型 → 字[无符号] / 位列[16位]型转换	5.15节
DINT_TO_WORD(_E)	双字[带符号]型 → 字[无符号] / 位列[16位]型转换	5.16节
INT_TO_DWORD(_E)	字[带符号]型 → 双字[无符号] / 位列[32位]型转换	5.17节
DINT_TO_DWORD(_E)	双字[带符号]型 → 双字[无符号] / 位列[32位]型转换	5.18节
INT_TO_BCD(_E)	字[带符号]型 → BCD型转换	5.19节
DINT_TO_BCD(_E)	双字[带符号]型 → BCD型转换	5.20节
INT_TO_TIME(_E)	字[带符号]型 → 时间型转换	5.21节
DINT_TO_TIME(_E)	双字[带符号]型 → 时间型转换	5.22节
REAL_TO_INT(_E)	单精度实数型 → 字[带符号]型转换	5.23节
REAL_TO_DINT(_E)	单精度实数型 → 双字[带符号]型转换	5.24节
REAL_TO_STR(_E)	单精度实数型 → 字符串型转换	5.25节
WORD_TO_BOOL(_E)	字[无符号] / 位列[16位]型 → 位型转换	5.26节
DWORD_TO_BOOL(_E)	双字[无符号] / 位列[32位]型 → 位型转换	5.27节
WORD_TO_INT(_E)	字[无符号] / 位列[16位]型 → 字[带符号]型转换	5.28节
WORD_TO_DINT(_E)	字[无符号] / 位列[16位]型 → 双字[带符号]型转换	5.29节
DWORD_TO_INT(_E)	双字[无符号] / 位列[32位]型 → 字[带符号]型转换	5.30节
DWORD_TO_DINT(_E)	双字[无符号] / 位列[32位]型 → 双字[带符号]型转换	5.31节
WORD_TO_DWORD(_E)	字[无符号] / 位列[16位]型 → 双字[无符号] / 位列[32位]型转换	5.32节
DWORD_TO_WORD(_E)	双字[无符号] / 位列[32位]型 → 字[无符号] / 位列[16位]型转换	5.33节
WORD_TO_TIME(_E)	字[无符号] / 位列[16位]型 → 时间型转换	5.34节
DWORD_TO_TIME(_E)	双字[无符号] / 位列[32位]型 → 时间型转换	5.35节
STR_TO_BOOL(_E)	字符串型 → 位型转换	5.36节
STR_TO_INT(_E)	字符串型 → 字[带符号]型转换	5.37节
STR_TO_DINT(_E)	字符串型 → 双字[带符号]型转换	5.38节
STR_TO_REAL(_E)	字符串型 → 单精度实数型转换	5.39节
STR_TO_TIME(_E)	字符串型 → 时间型转换	5.40节
BCD_TO_INT(_E)	BCD型 → 字[带符号]型转换	5.41节
BCD_TO_DINT(_E)	BCD型 → 双字[带符号]型转换	5.42节
TIME_TO_BOOL(_E)	时间型 → 位型转换	5.43节
TIME_TO_INT(_E)	时间型 → 字[带符号]型转换	5.44节

函数名	功能	参考
TIME_TO_DINT(_E)	时间型 → 双字[带符号]型转换	5.45节
TIME_TO_STR(_E)	时间型 → 字符串型转换	5.46节
TIME_TO_WORD(_E)	时间型 → 字[无符号] / 位列[16位]型转换	5.47节
TIME_TO_DWORD(_E)	时间型 → 双字[无符号] / 位列[32位]型转换	5.48节
BITARR_TO_INT(_E)	位阵列 → 字[带符号]型、字[无符号] / 位列[16位]型转换	5.49节
BITARR_TO_DINT(_E)	位阵列 → 双字[带符号]型、双字[无符号] / 位列[32位]型转换	5.50节
INT_TO_BITARR(_E)	字[带符号]型、字[无符号] / 位列[16位]型 → 位阵列转换	5.51节
DINT_TO_BITARR(_E)	双字[带符号]型、双字[无符号] / 位列[32位]型 → 位阵列转换	5.52节
CPY_BITARR(_E)	位阵列的复制	5.53节
GET_BIT_OF_INT(_E)	字[带符号]型数据的指定位读出	5.54节
SET_BIT_OF_INT(_E)	字[带符号]型数据的指定位写入	5.55节
CPY_BIT_OF_INT(_E)	字[带符号]型数据的指定位复制	5.56节
GET_BOOL_ADDR	起始数据的获取	5.57节
GET_INT_ADDR	起始数据的获取	5.58节
GET_WORD_ADDR	起始数据的获取	5.59节

1 概要

2 函数、操作符一览

3 函数的构成

4 函数的阅读方法

5 应用函数(型转换函数)

6 应用函数(单数值变量函数)

7 应用函数(算术运算函数)

8 应用函数(位移位函数)

9 应用函数(位型布尔函数)

10 应用函数(选择函数)

## 5.1 BOOL\_TO\_INT(E) / 位型 → 字[带符号]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将位型数据转换为字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BOOL_TO_INT		BOOL_TO_INT(_BOOL); 例: D0:=BOOL_TO_INT(M0);
BOOL_TO_INT_E		BOOL_TO_INT_E(EN, _BOOL, 输出标签); 例: BOOL_TO_INT_E(X000, M0, D0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_BOOL ( (s) )	转换原位数据
输出变量	ENO	执行状态
	*1 ( (d) )	转换后的字[带符号]数据

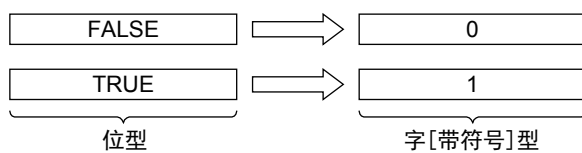
在函数说明中，输入输出变量通过 ( ) 中的描述进行记载。

### 功能和动作说明

将 (s) 指定的软元件的位型数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中。

输入值为FALSE时，输出字[带符号]型的数据值0。

输入值为TRUE时，输出字[带符号]型的数据值1。



### 注意要点

如果从母线连接函数，则请使用函数名带“\_E”的函数。

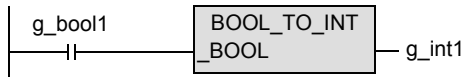


## 程序举例

将 (s) 指定的软元件的位型数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序

### 1) 无EN/ENO函数 (BOOL\_TO\_INT)

[结构化梯形图/FBD]

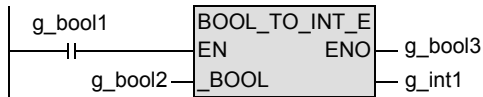


[ST]

```
g_int1 := BOOL_TO_INT(g_bool1);
```

### 2) 带EN/ENO函数 (BOOL\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BOOL_TO_INT_E(g_bool1, g_bool2, g_int1);
```

## 5.2 BOOL\_TO\_DINT(E) / 位型 → 双字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将位型数据转换为双字[带符号]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BOOL_TO_DINT		BOOL_TO_DINT(_BOOL); 例: 标签:=BOOL_TO_DINT(M0);
BOOL_TO_DINT_E		BOOL_TO_DINT_E(EN,_BOOL,输出标签); 例: BOOL_TO_DINT_E(X000,M0,标签);

\*1. 变为输出变量。

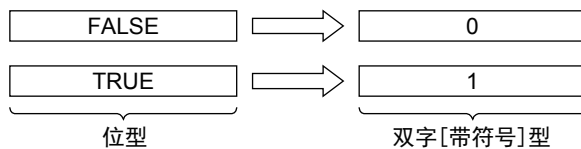
### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_BOOL (s)	转换原位数据
输出变量	ENO	执行状态
	*1 (d)	转换后的双字[带符号]数据

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的位型数据转换为双字[带符号]型数据，输出到(d)中指定的软元件中。



### 注意要点

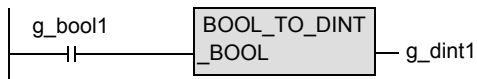
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的位型数据转换为双字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (BOOL\_TO\_DINT)

[结构化梯形图/FBD]

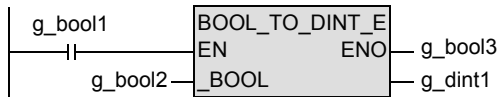


[ST]

```
g_dint1 := BOOL_TO_DINT(g_bool1);
```

### 2) 带EN/ENO函数 (BOOL\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BOOL_TO_DINT_E(g_bool1, g_bool2, g_dint1);
```

## 5.3 BOOL\_TO\_STR(E) / 位型 → 字符串型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

将位型数据转换为字符串型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BOOL_TO_STR		BOOL_TO_STR(BOOL); 例: 标签:=BOOL_TO_STR(M0);
BOOL_TO_STR_E		BOOL_TO_STR_E(EN, _BOOL, 输出标签); 例: BOOL_TO_STR_E(X000, M0, 标签);

\*1. 变为输出变量。

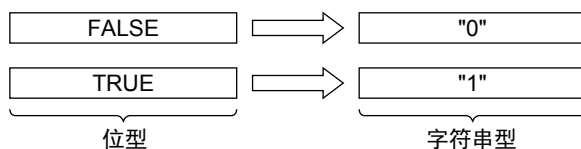
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_BOOL (s)	转换原位数据	位
输出变量	ENO	执行状态	位
	*1 (d)	转换后的字符串数据	字符串

在函数说明中, 输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件中被输入的位型数据转换为字符串型数据, 输出到(d)中指定的软元件中。



### 注意要点

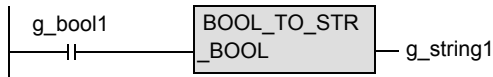
- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时, 不能像简洁工程那样直接指定16位软元件。处理字符串数据时, 请使用标签。指定标签时, 使用全局标签。

## 程序举例

将 (s) 指定的软元件的位型数据转换为字符串型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (BOOL\_TO\_STR)

[结构化梯形图/FBD]

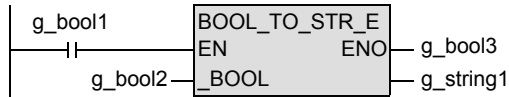


[ST]

```
g_string1 := BOOL_TO_STR(g_bool1);
```

### 2) 带EN/ENO函数 (BOOL\_TO\_STR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BOOL_TO_STR_E(g_bool1, g_bool2, g_string1);
```

## 5.4 BOOL\_TO\_WORD(E) / 位型 → 字[无符号] / 位列[16位]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将位型数据转换为字[无符号]/位列[16位]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BOOL_TO_WORD		BOOL_TO_WORD(_BOOL); 例: D0:=BOOL_TO_WORD(M0);
BOOL_TO_WORD_E		BOOL_TO_WORD_E(EN,_BOOL,输出标签); 例: BOOL_TO_WORD_E(X000,M0,D0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	_BOOL (C <sub>s</sub> )	位
输出变量	ENO	位
	*1 (C <sub>d</sub> )	字[无符号]/位列[16位]

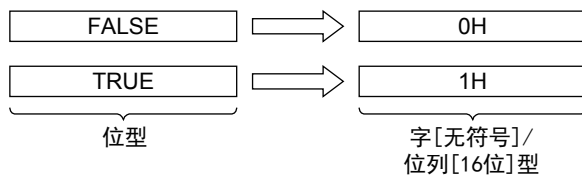
在函数说明中，输入输出变量通过(C)中的描述进行记载。

### 功能和动作说明

将(C<sub>s</sub>)指定的软元件的位型数据转换为字[无符号]/位列[16位]型数据，输出到(C<sub>d</sub>)中指定的软元件中。

输入值为FALSE时，输出字[无符号]/位列[16位]型数据值0H。

输入值为TRUE时，输出字[无符号]/位列[16位]型数据值1H。



### 注意要点

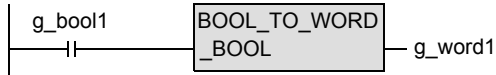
如果从母线连接函数，则请使用函数名带“\_E”的函数。

### 程序举例

将 (s) 指定的软元件的位型数据转换为字[无符号]/位列[16位]型数据，输出到 (d) 中指定的软元件中的程序。

#### 1) 无EN/ENO函数 (BOOL\_TO\_WORD)

[结构化梯形图/FBD]

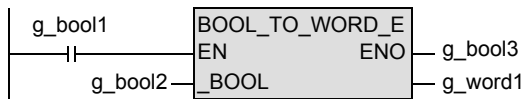


[ST]

```
g_word1 := BOOL_TO_WORD(g_bool1);
```

#### 2) 带EN/ENO函数 (BOOL\_TO\_WORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BOOL_TO_WORD_E(g_bool1, g_bool2, g_word1);
```

## 5.5 BOOL\_TO\_DWORD(E) / 位型 → 双字[无符号] / 位列[32位]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将位型数据转换为双字[无符号]/位列[32位]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BOOL_TO_DWORD		BOOL_TO_DWORD(_BOOL); 例: 标签:=BOOL_TO_DWORD(M0);
BOOL_TO_DWORD_E		BOOL_TO_DWORD_E(EN,_BOOL,输出标签); 例: BOOL_TO_DWORD_E(X000,M0,标签);

\*1. 变为输出变量。

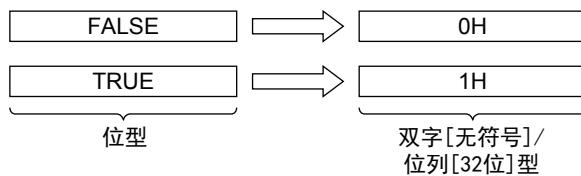
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	_BOOL (S)	位
输出变量	ENO	位
	*1 (D)	双字[无符号]/位列[32位]

在函数说明中，输入输出变量通过(S)中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的位型数据转换为双字[无符号]/位列[32位]型数据，输出到(D)中指定的软元件中。  
 输入值为FALSE时，输出双字[无符号]/位列[32位]型数据值0H。  
 输入值为TRUE时，输出双字[无符号]/位列[32位]型数据值1H。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

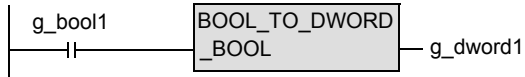


## 程序举例

将 (s) 指定的软元件的位型数据转换为双字[无符号]/位列[32位]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数(BOOL\_TO\_DWORD)

[结构化梯形图/FBD]

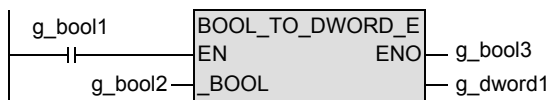


[ST]

```
g_dword1 := BOOL_TO_DWORD(g_bool1);
```

### 2) 带EN/ENO函数(BOOL\_TO\_DWORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BOOL_TO_DWORD_E(g_bool1, g_bool2, g_dword1);
```

## 5.6 BOOL\_TO\_TIME(E) / 位型 → 时间型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将位型数据转换为时间型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BOOL_TO_TIME		BOOL_TO_TIME(_BOOL); 例: 标签:=BOOL_TO_TIME(M0);
BOOL_TO_TIME_E		BOOL_TO_TIME_E(EN,_BOOL,输出标签); 例: BOOL_TO_TIME_E(X000,M0,标签);

\*1. 变为输出变量。

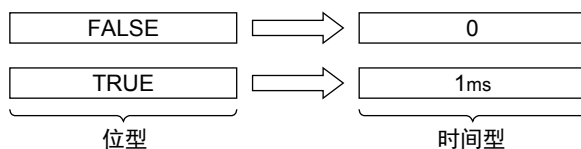
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_BOOL (s)	转换原位数据
输出变量	ENO	执行状态
	*1 (d)	转换后的时间数据

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的位型数据转换为时间型数据，输出到(d)中指定的软元件中。



### 注意要点

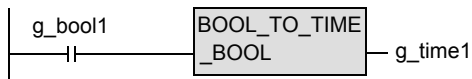
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的位型数据转换为时间型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (BOOL\_TO\_TIME)

[结构化梯形图/FBD]

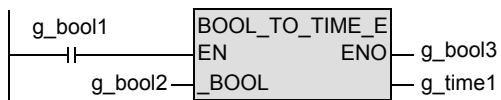


[ST]

```
g_time1 := BOOL_TO_TIME(g_bool1);
```

### 2) 带EN/ENO函数 (BOOL\_TO\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BOOL_TO_TIME_E(g_bool1, g_bool2, g_time1);
```

## 5.7 INT\_TO\_DINT(E) / 字[带符号]型 → 双字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[带符号]型数据转换为双字[带符号]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_DINT		INT_TO_DINT(_INT); 例: 标签:=INT_TO_DINT(D0);
INT_TO_DINT_E		INT_TO_DINT_E(EN,_INT,输出标签); 例: INT_TO_DINT_E(X000,D0,标签);

\*1. 变为输出变量。

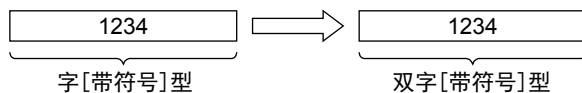
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_INT (s)	转换原字[带符号]数据	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	转换后的双字[带符号]数据	双字[带符号]

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字[带符号]型数据转换为双字[带符号]型数据，输出到(d)中指定的软元件中。



### 注意要点

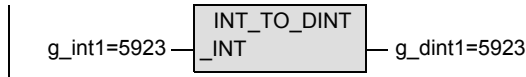
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

### 程序举例

将 (s) 指定的软元件的字[带符号]型数据转换为双字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

#### 1) 无EN/ENO函数 (INT\_TO\_DINT)

[结构化梯形图/FBD]

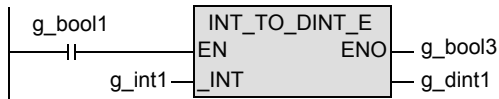


[ST]

```
g_dint1 := INT_TO_DINT(g_int1);
```

#### 2) 带EN/ENO函数 (INT\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := INT_TO_DINT_E(g_bool1, g_int1, g_dint1);
```

## 5.8 DINT\_TO\_INT(E) / 双字[带符号]型 → 字[带符号]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[带符号]型数据转换为字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_INT		DINT_TO_INT(DINT); 例: D10:=DINT_TO_INT(标签);
DINT_TO_INT_E		DINT_TO_INT_E(EN, DINT, 输出标签); 例: DINT_TO_INT_E(X000, 标签, D10);

\*1. 变为输出变量。

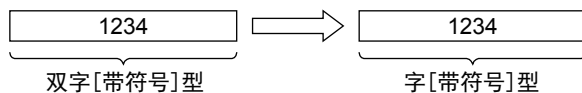
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_DINT (C <sub>S</sub> )	转换原双字[带符号]数据	双字[带符号]
输出变量	ENO	执行状态	位
	*1 (C <sub>D</sub> )	转换后的字[带符号]数据	字[带符号]

在函数说明中，输入输出变量通过(C)中的描述进行记载。

### 功能和动作说明

将(C<sub>S</sub>)指定的软元件的双字[带符号]型数据转换为字[带符号]型数据，输出到(C<sub>D</sub>)中指定的软元件中。



### 注意要点

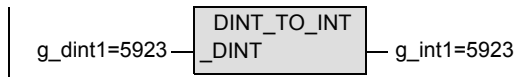
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的双字[带符号]型数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (DINT\_TO\_INT)

[结构化梯形图/FBD]

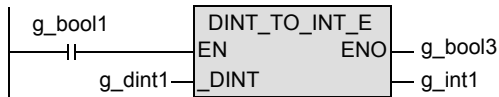


[ST]

```
g_int1 := DINT_TO_INT(g_dint1);
```

### 2) 带EN/ENO函数 (DINT\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_INT_E(g_bool1, g_dint1, g_int1);
```

## 5.9 INT\_TO\_BOOL(E) / 字[带符号]型 → 位型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[带符号]型数据转换为位型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_BOOL		INT_TO_BOOL(_INT); 例: M0:=INT_TO_BOOL(D0);
INT_TO_BOOL_E		INT_TO_BOOL_E(EN, _INT, 输出标签); 例: INT_TO_BOOL_E(X000, D0, M0);

\*1. 变为输出变量。

### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_INT ( <u>s</u> )	转换原字[带符号]数据
输出变量	ENO	执行状态
	*1 ( <u>d</u> )	转换后的位数据

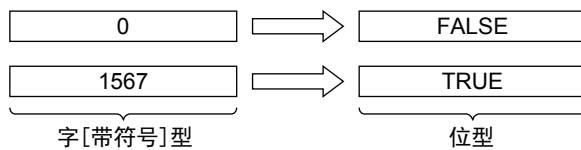
在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将s指定的软元件的字[带符号]型数据转换为位型数据，输出到d中指定的软元件中。

输入值为0时，输出FALSE。

输入值为0以外时，输出TRUE。



### 注意要点

如果从母线连接函数，则请使用函数名带“\_E”的函数。

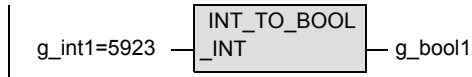


## 程序举例

将 (s) 指定的软元件的字[带符号]型数据转换为位型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (INT\_TO\_BOOL)

[结构化梯形图/FBD]

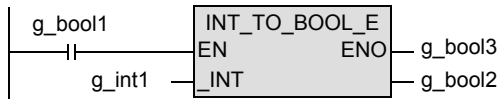


[ST]

```
g_bool1 := INT_TO_BOOL(g_int1);
```

### 2) 带EN/ENO函数 (INT\_TO\_BOOL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := INT_TO_BOOL_E(g_bool1, g_int1, g_bool2);
```

## 5.10 DINT\_TO\_BOOL(E) / 双字[带符号]型 → 位型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[带符号]型数据转换为位型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_BOOL		DINT_TO_BOOL(_DINT); 例: M0:=DINT_TO_BOOL(标签);
DINT_TO_BOOL_E		DINT_TO_BOOL_E(EN, _DINT, 输出标签); 例: DINT_TO_BOOL_E(X000, 标签, M0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_DINT (C <sub>S</sub> )	转换原双字[带符号]数据	双字[带符号]
输出变量	ENO	执行状态	位
	*1 (C <sub>D</sub> )	转换后的位数据	位

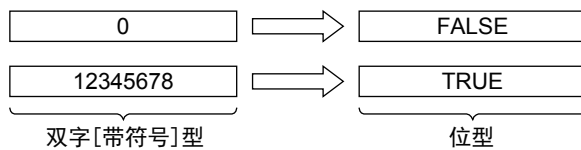
在函数说明中，输入输出变量通过(C)中的描述进行记载。

### 功能和动作说明

将(C<sub>S</sub>)指定的软元件的双字[带符号]型数据转换为位型数据，输出到(C<sub>D</sub>)中指定的软元件中。

输入值为0时，输出FALSE。

输入值为0以外时，输出TRUE。



### 注意要点

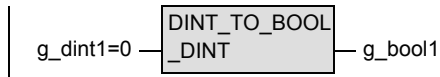
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 **(s)** 指定的软元件的双字[带符号]型数据转换为位型数据，输出到 **(d)** 中指定的软元件中的程序。

### 1) 无EN/ENO函数(DINT\_TO\_BOOL)

[结构化梯形图/FBD]

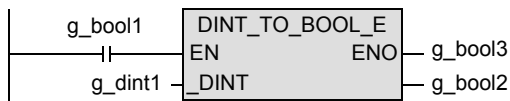


[ST]

```
g_bool1 := DINT_TO_BOOL(g_dint1);
```

### 2) 带EN/ENO函数(DINT\_TO\_BOOL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_BOOL_E(g_bool1, g_dint1, g_bool2);
```

## 5.11 INT\_TO\_REAL(E) / 字[带符号]型 → 单精度实数型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	△	○	○	×	×	×	×	×

### 概要

将字[带符号]型数据转换为单精度实数型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_REAL		INT_TO_REAL(a_Int); 例: 标签:=INT_TO_REAL(D0);
INT_TO_REAL_E		INT_TO_REAL_E(EN, a_Int, 输出标签); 例: INT_TO_REAL_E(X000, D0, 标签);

\*1. 变为输出变量。

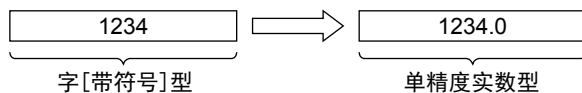
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	a_Int (S)	转换原字[带符号]数据	字[带符号]
输出变量	ENO	执行状态	位
	*1 (D)	转换后的单精度实数数据	单精度实数

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的字[带符号]型数据转换为单精度实数型数据，输出到(D)中指定的软元件中。



### 注意要点

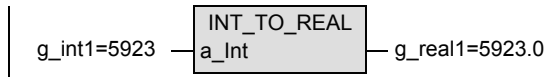
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。
- 3) FX3G可编程控制器V1.10以上版本支持函数。
- 4) 单精度实数型数据以32位的单精度进行处理，因此有效位数约为7位。因此，整数值超过-16777216 ~ 16777215范围时，转换的值会产生误差。(化整误差)

### 程序举例

将 (s) 指定的软元件的字[带符号]型数据转换为单精度实数型数据，输出到 (d) 中指定的软元件中的程序。

#### 1) 无EN/ENO函数 (INT\_TO\_REAL)

[结构化梯形图/FBD]

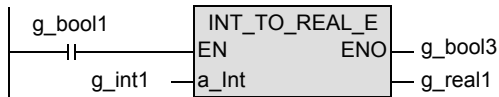


[ST]

g\_real1 := INT\_TO\_REAL(g\_int1);

#### 2) 带EN/ENO函数 (INT\_TO\_REAL\_E)

[结构化梯形图/FBD]



[ST]

g\_bool3 := INT\_TO\_REAL\_E(g\_bool1, g\_int1, g\_real1);

## 5.12 DINT\_TO\_REAL(E) / 双字[带符号]型 → 单精度实数型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	△	○	○	×	×	×	×	×

### 概要

将双字[带符号]型数据转换为单精度实数型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_REAL		DINT_TO_REAL(a_Dint); 例: 标签2:=DINT_TO_REAL(标签1);
DINT_TO_REAL_E		DINT_TO_REAL_E(EN, a_Dint, 输出标签); 例: DINT_TO_REAL_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

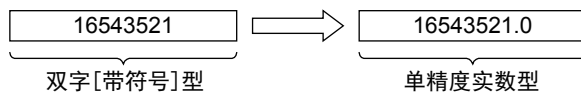
### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	a_Dint (S)	转换原双字[带符号]数据
输出变量	ENO	执行状态
	*1 (D)	转换后的单精度实数数据

在函数说明中，输入输出变量通过(S)中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的双字[带符号]型数据转换为单精度实数型数据，输出到(D)中指定的软元件中。



### 注意要点

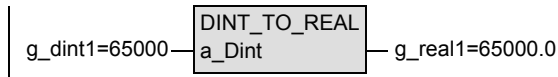
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 3) FX3G可编程控制器V1.10以上版本支持函数。

## 程序举例

将 (s) 指定的软元件的双字[带符号]型数据转换为单精度实数型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (DINT\_TO\_REAL)

[结构化梯形图/FBD]

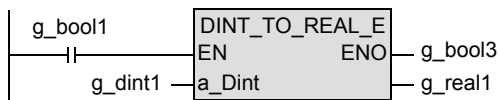


[ST]

```
g_real1 := DINT_TO_REAL(g_dint1);
```

### 2) 带EN/ENO函数 (DINT\_TO\_REAL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_REAL_E(g_bool1, g_dint1, g_real1);
```

## 5.13 INT\_TO\_STR(E) / 字[带符号]型 → 字符串型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	×	×	×	×	×	×	×	×

### 概要

将字[带符号]型数据转换为字符串型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_STR		INT_TO_STR(_INT); 例: 标签:=INT_TO_STR(D0);
INT_TO_STR_E		INT_TO_STR_E(EN, _INT, 输出标签); 例: INT_TO_STR_E(X000, D0, 标签);

\*1. 变为输出变量。

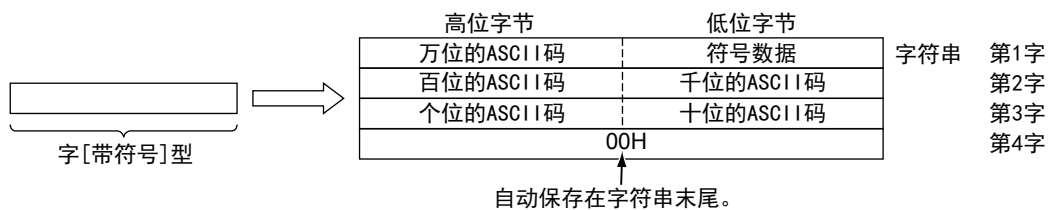
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_INT (s)	转换原字[带符号]数据	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	转换后的字符串数据	字符串

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

- 1) 将(s)指定的软元件的字[带符号]型数据转换为字符串型数据，输出到(d)中指定的软元件中。



- 2) “符号数据”中，输入值为正时保存“20H(空格)”，为负时保存“2DH(-)”。
  - 3) 有效位数少时，高位数中保存“20H(空格)”。
- (例) 输入了-123时



- 4) 在字符串的最后(第4个字)中，自动保存“00H”。



## 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

## 错误

以下一些情况下会出现运算错误，错误标志位M8067为ON，错误代码保存在D8067中。

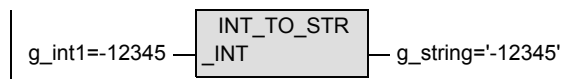
- 1) 字符串型数据的保存位置(Ⓒd)中指定的软元件)的占用点数，超出了软元件的范围时。  
(错误代码：K6706)

## 程序举例

将Ⓒs)指定的软元件的字[带符号]型数据转换为字符串型数据，输出到Ⓒd)中指定的软元件中的程序。

- 1) 无EN/ENO函数(INT\_TO\_STR)

[结构化梯形图/FBD]

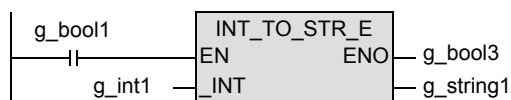


[ST]

g\_string1 := INT\_TO\_STR(g\_int1);

- 2) 带EN/ENO函数(INT\_TO\_STR\_E)

[结构化梯形图/FBD]



[ST]

g\_bool3 := INT\_TO\_STR\_E(g\_bool1, g\_int1, g\_string1);

## 5.14 DINT\_TO\_STR(E) / 双字[带符号]型 → 字符串型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

将双字[带符号]型数据转换为字符串型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_STR		DINT_TO_STR(DINT); 例: 标签2:=DINT_TO_STR(标签1);
DINT_TO_STR_E		DINT_TO_STR_E(EN, DINT, 输出标签); 例: DINT_TO_STR_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	_DINT ( (s) )	双字[带符号]
输出变量	ENO	位
	*1 ( (d) )	字符串

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 1) 将(s)指定的软元件的双字[带符号]型数据转换为字符串型数据，输出到(d)中指定的软元件中。



- 2) “符号数据”中，输入值为正时保存“20H(空格)”，为负时保存“2DH(-)”。
  - 3) 有效位数少时，高位数中保存“20H(空格)”。
- (例) 输入了-123456时



- 4) 在字符串的最后(第6个字的高位字节)中，自动保存“00H”。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据及32位数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据及32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

### 错误

以下一些情况下会出现运算错误，错误标志位M8067为ON，错误代码保存在D8067中。

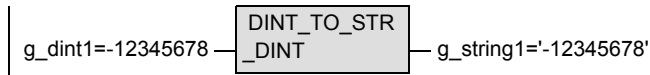
- 1) 字符串型数据的保存位置(**d**)中指定的软元件)的占用点数，超出了软元件的范围时。  
(错误代码：K6706)

### 程序举例

将(**s**)指定的软元件的双字[带符号]型数据转换为字符串型数据，输出到(**d**)中指定的软元件中的程序。

- 1) 无EN/ENO函数(DINT\_TO\_STR)

[结构化梯形图/FBD]

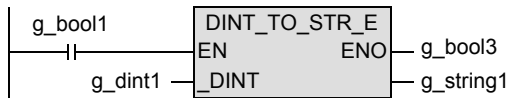


[ST]

```
g_string1 := DINT_TO_STR(g_dint1);
```

- 2) 带EN/ENO函数(DINT\_TO\_STR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_STR_E(g_bool1, g_dint1, g_string1);
```

## 5.15 INT\_TO\_WORD(E) / 字[带符号]型 → 字[无符号] / 位列[16位]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[带符号]型数据转换为字[无符号]/位列[16位]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_WORD		INT_TO_WORD(_INT); 例: D10:=INT_TO_WORD(D0);
INT_TO_WORD_E		INT_TO_WORD_E(EN,_INT,输出标签); 例: INT_TO_WORD_E(X000,D0,D10);

\*1. 变为输出变量。

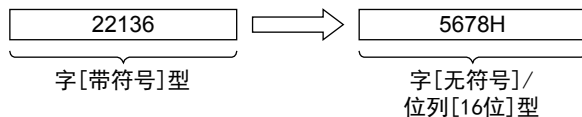
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	_INT (S)	字[带符号]
输出变量	ENO	位
	*1 (D)	字[无符号]/位列[16位]

在函数说明中，输入输出变量通过(S)中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的字[带符号]型数据转换为字[无符号]/位列[16位]型数据，输出到(D)中指定的软元件中。



### 注意要点

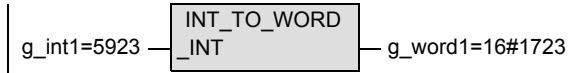
如果从母线连接函数，则请使用函数名带“\_E”的函数。

## 程序举例

将 (s) 指定的软元件的字[带符号]型数据转换为字[无符号]/位列[16位]型数据，输出到 (d) 中指定的软元件中的程序。

- 1) 无EN/ENO函数 (INT\_TO\_WORD)

[结构化梯形图/FBD]

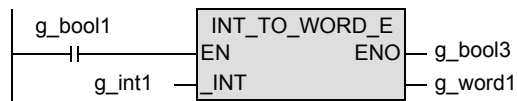


[ST]

```
g_word1 := INT_TO_WORD(g_int1);
```

- 2) 带EN/ENO函数 (INT\_TO\_WORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := INT_TO_WORD_E(g_bool1, g_int1, g_word1);
```

## 5.16 DINT\_TO\_WORD(E) / 双字[带符号]型 → 字[无符号] / 位列[16位]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[带符号]型数据转换为字[无符号]/位列[16位]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_WORD		DINT_TO_WORD(DINT); 例: D10:=DINT_TO_WORD(标签);
DINT_TO_WORD_E		DINT_TO_WORD_E(EN, DINT, 输出标签); 例: DINT_TO_WORD_E(X000, 标签, D10);

\*1. 变为输出变量。

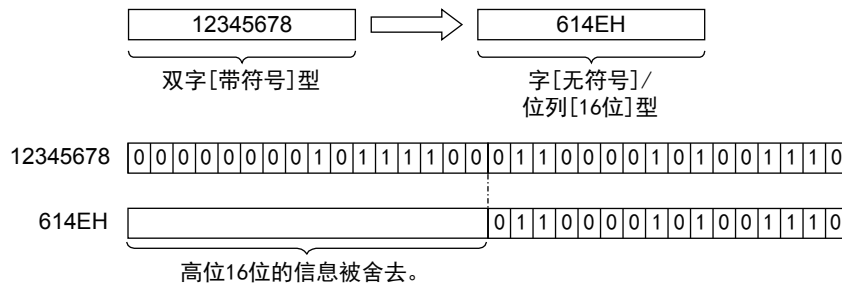
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_DINT ( (s) )	转换原双字[带符号]数据
输出变量	ENO	执行状态
	*1 ( (d) )	转换后的字[无符号]/位列[16位]数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的双字[带符号]型数据转换为字[无符号]/位列[16位]型数据，输出到(d)中指定的软元件中。



### 注意要点

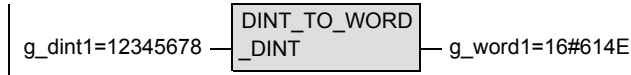
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的双字[带符号]型数据转换为字[无符号]/位列[16位]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数(DINT\_TO\_WORD)

[结构化梯形图/FBD]

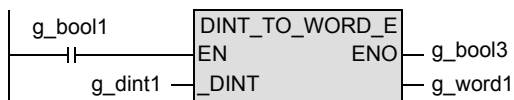


[ST]

```
g_word1 := DINT_TO_WORD(g_dint1);
```

### 2) 带EN/ENO函数(DINT\_TO\_WORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_WORD_E(g_bool1, g_dint1, g_word1);
```

## 5.17 INT\_TO\_DWORD(E) / 字[带符号]型 → 双字[无符号] / 位列[32位]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将字[带符号]型数据转换为双字[无符号]/位列[32位]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_DWORD		INT_TO_DWORD(_INT); 例: 标签:=INT_TO_DWORD(D0);
INT_TO_DWORD_E		INT_TO_DWORD_E(EN, _INT, 输出标签); 例: INT_TO_DWORD_E(X000, D0, 标签);

\*1. 变为输出变量。

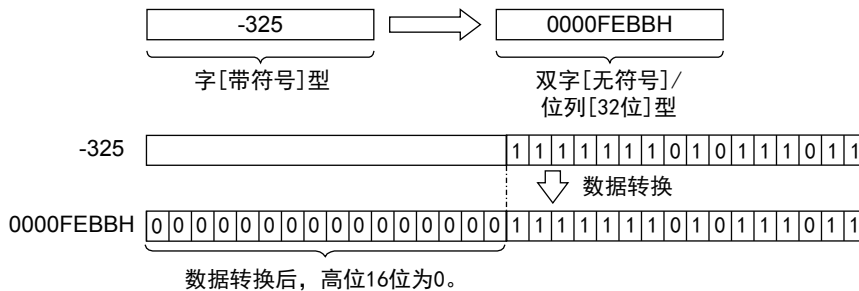
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_INT ( <u>s</u> )	转换原字[带符号]数据	字[带符号]
输出变量	ENO	执行状态	位
	*1 ( <u>d</u> )	转换后的双字[无符号]/位列[32位]数据	双字[无符号]/位列[32位]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将 s 指定的软元件的字[带符号]型数据转换为双字[无符号]/位列[32位]型数据，输出到 d 中指定的软元件。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。

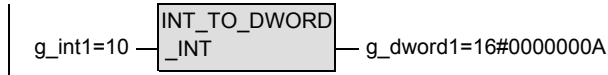


## 程序举例

将 (s) 指定的软元件的字[带符号]型数据转换为双字[无符号]/位列[32位]型数据，输出到 (d) 指定的软元件中的程序。

### 1) 无EN/ENO函数(INT\_TO\_DWORD)

[结构化梯形图/FBD]

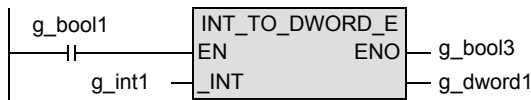


[ST]

```
g_dword1 := INT_TO_DWORD(g_int1);
```

### 2) 带EN/ENO函数(INT\_TO\_DWORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := INT_TO_DWORD_E(g_bool1, g_int1, g_dword1);
```

## 5.18 DINT\_TO\_DWORD(E) / 双字[带符号]型 → 双字[无符号] / 位列[32位]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[带符号]型数据转换为双字[无符号]/位列[32位]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_DWORD		DINT_TO_DWORD(_DINT); 例: 标签2:=DINT_TO_DWORD(标签1);
DINT_TO_DWORD_E		DINT_TO_DWORD_E(EN, _DINT, 输出标签); 例: DINT_TO_DWORD_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

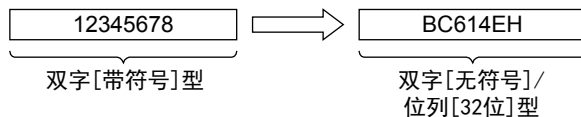
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_DINT (S)	转换原双字[带符号]数据	双字[带符号]
输出变量	ENO	执行状态	位
	*1 (D)	转换后的双字[无符号]/位列[32位]数据	双字[无符号]/位列[32位]

在函数说明中，输入输出变量通过(S)中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的双字[带符号]型数据转换为双字[无符号]/位列[32位]型数据，输出到(D)中指定的软元件中。



### 注意要点

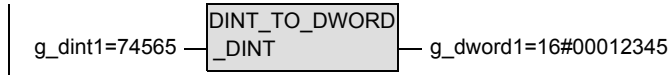
- 1) 如果从母线连接函数，则请使用函数名带“E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的双字[带符号]型数据转换为双字[无符号]/位列[32位]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数(DINT\_TO\_DWORD)

[结构化梯形图/FBD]

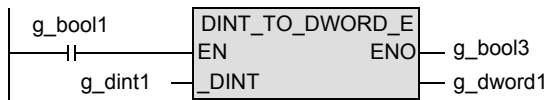


[ST]

```
g_dword1 := DINT_TO_DWORD(g_dint1);
```

### 2) 带EN/ENO函数(DINT\_TO\_DWORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_DWORD_E(g_bool1, g_dint1, g_dword1);
```

## 5.19 INT\_TO\_BCD(E) / 字[带符号]型 → BCD型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[带符号]型数据转换为BCD型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_BCD		INT_TO_BCD(_INT); 例: D10:=INT_TO_BCD(D0);
INT_TO_BCD_E		INT_TO_BCD_E(EN, _INT, 输出标签); 例: INT_TO_BCD_E(X000, D0, D10);

\*1. 变为输出变量。

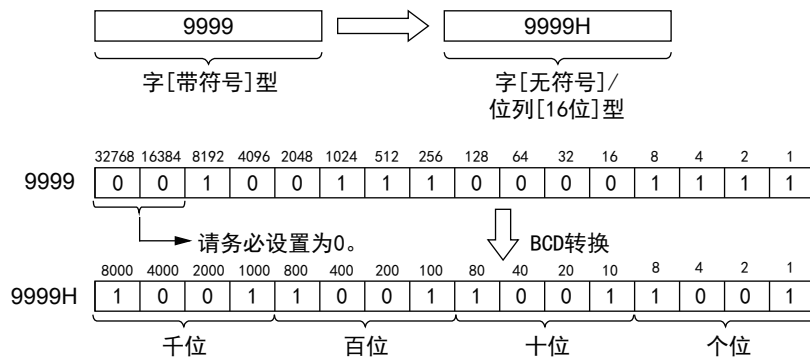
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_INT (s)	转换原字[带符号]数据	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	转换后的BCD数据	字[无符号]/位列[16位]

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字[带符号]型数据转换为BCD型数据，输出到(d)中指定的软元件中。



### 注意要点

如果从母线连接函数，则请使用函数名带“\_E”的函数。

### 错误

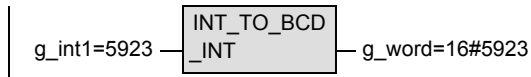
(s)指定的软元件的值若在0~9,999以外范围时出现运算错误。

## 程序举例

将 (s) 指定的软元件的字[带符号]型数据转换为BCD型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (INT\_TO\_BCD)

[结构化梯形图/FBD]

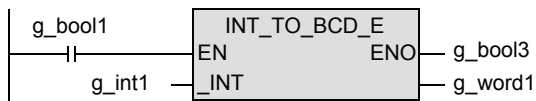


[ST]

```
g_word1 := INT_TO_BCD(g_int1);
```

### 2) 带EN/ENO函数 (INT\_TO\_BCD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := INT_TO_BCD_E(g_bool1, g_int1, g_word1);
```

## 5.20 DINT\_TO\_BCD(\_E) / 双字[带符号]型 → BCD型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[带符号]型数据转换为BCD型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_BCD		DINT_TO_BCD(DINT); 例: 标签2:=DINT_TO_BCD(标签1);
DINT_TO_BCD_E		DINT_TO_BCD_E(EN, DINT, 输出标签); 例: DINT_TO_BCD_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

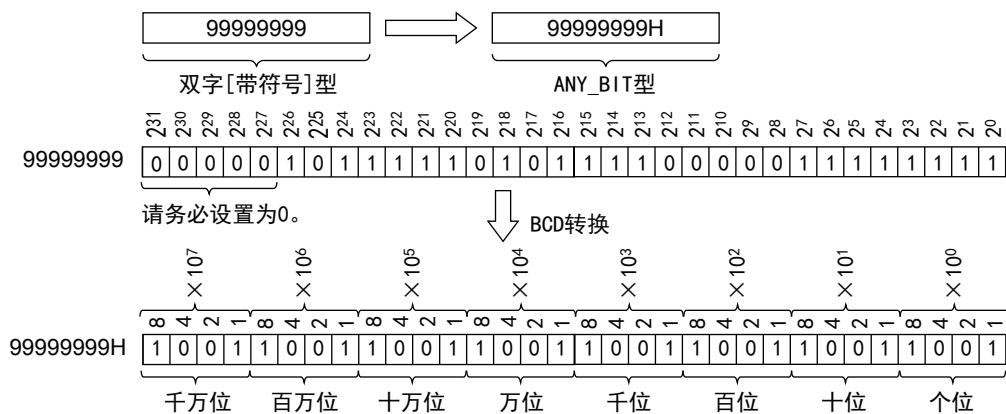
#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
_DINT ( (s) )	转换原双字[带符号]数据	双字[带符号]
ENO	执行状态	位
*1 ( (d) )	转换后的BCD数据	ANY_BIT

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的双字[带符号]型数据转换为BCD型数据，输出到(d)中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 错误

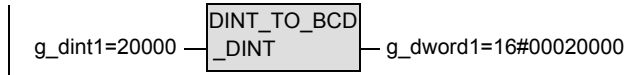
将 (s) 中指定的软元件的值若在0~99,999,999以外范围时出现运算错误。

## 程序举例

将 (s) 指定的软元件的双字[带符号]型数据转换为BCD型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (DINT\_TO\_BCD)

[结构化梯形图/FBD]

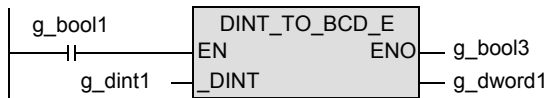


[ST]

```
g_dword1 := DINT_TO_BCD(g_dint1);
```

### 2) 带EN/ENO函数 (DINT\_TO\_BCD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_BCD_E(g_bool1, g_dint1, g_dword1);
```

## 5.21 INT\_TO\_TIME(E) / 字[带符号]型 → 时间型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[带符号]型数据转换为时间型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_TIME		INT_TO_TIME(_INT); 例: 标签:=INT_TO_TIME(DO);
INT_TO_TIME_E		INT_TO_TIME_E(EN,_INT,输出标签); 例: INT_TO_TIME_E(X000,DO,标签);

\*1. 变为输出变量。

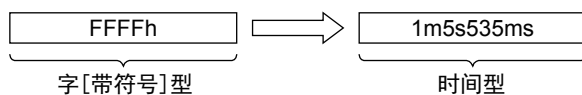
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_INT (s)	转换原字[带符号]数据	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	转换后的时间数据	时间

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字[带符号]型数据转换为时间型数据，输出到(d)中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

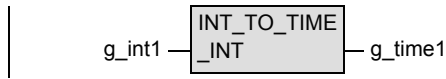


## 程序举例

将 **(s)** 指定的软元件的字[带符号]型数据转换为时间型数据，输出到 **(d)** 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (INT\_TO\_TIME)

[结构化梯形图/FBD]

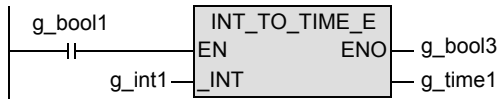


[ST]

```
g_time1 := INT_TO_TIME(g_int1);
```

### 2) 带EN/ENO函数 (INT\_TO\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := INT_TO_TIME_E(g_bool1, g_int1, g_time1);
```

## 5.22 DINT\_TO\_TIME(E) / 双字[带符号]型 → 时间型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[带符号]型数据转换为时间型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_TIME		DINT_TO_TIME(DINT); 例: 标签2:=DINT_TO_TIME(标签1);
DINT_TO_TIME_E		DINT_TO_TIME_E(EN, DINT, 输出标签); 例: DINT_TO_TIME_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

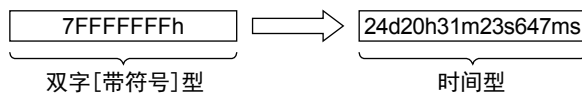
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	_DINT (s)	双字[带符号]
输出变量	ENO	位
	*1 (d)	时间

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的双字[带符号]型数据转换为时间型数据，输出到(d)中指定的软元件中。



### 注意要点

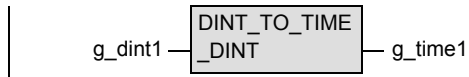
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的双字[带符号]型数据转换为时间型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (DINT\_TO\_TIME)

[结构化梯形图/FBD]

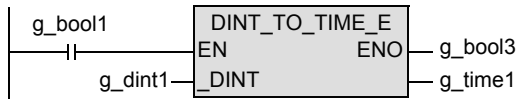


[ST]

```
g_time1 := DINT_TO_TIME(g_dint1);
```

### 2) 带EN/ENO函数 (DINT\_TO\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DINT_TO_TIME_E(g_bool1, g_dint1, g_time1);
```

## 5.23 REAL\_TO\_INT(E) / 单精度实数型 → 字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	△	○	○	×	×	×	×	×

### 概要

将单精度实数型数据转换为字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
REAL_TO_INT		REAL_TO_INT(a_real); 例: D10:=REAL_TO_INT(标签);
REAL_TO_INT_E		REAL_TO_INT_E(EN, a_real, 输出标签); 例: REAL_TO_INT_E(X000, 标签, D10);

\*1. 变为输出变量。

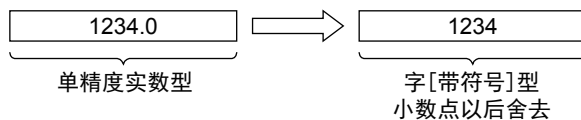
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	a_real (S)	单精度实数
输出变量	ENO	位
	*1 (D)	字[带符号]

在函数说明中，输入输出变量通过(S)中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的单精度实数型数据转换为字[带符号]型数据，输出到(D)中指定的软元件中。



### 注意要点

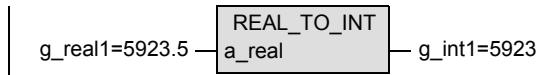
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 3) FX3G可编程控制器V1.10以上版本支持函数。
- 4) 转换后的数据为，将单精度实数型数据值舍去小数点后的值。
- 5) 使用编程工具在(S)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

## 程序举例

将 (s) 指定的软元件的单精度实数型数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (REAL\_TO\_INT)

[结构化梯形图/FBD]

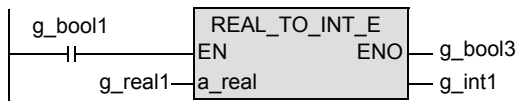


[ST]

```
g_int1 := REAL_TO_INT(g_real1);
```

### 2) 带EN/ENO函数 (REAL\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := REAL_TO_INT_E(g_bool1, g_real1, g_int1);
```

## 5.24 REAL\_TO\_DINT(E) / 单精度实数型 → 双字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	△	○	○	×	×	×	×	×

### 概要

将单精度实数型数据转换为双字[带符号]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
REAL_TO_DINT		REAL_TO_DINT(a_real); 例: 标签2:=REAL_TO_DINT(标签1);
REAL_TO_DINT_E		REAL_TO_DINT_E(EN, a_real, 输出标签); 例: REAL_TO_DINT_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

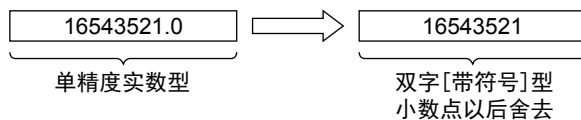
### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	a_real (S)	单精度实数
输出变量	ENO	位
	*1 (D)	双字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的单精度实数型数据转换为双字[带符号]型数据，输出到(D)中指定的软元件中。



### 注意要点

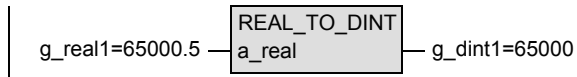
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 3) FX3G可编程控制器V1.10以上版本支持函数。
- 4) 转换后的数据为，将单精度实数型数据值舍去小数点后的值。
- 5) 使用编程工具在(S)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

## 程序举例

将 (s) 指定的软元件的单精度实数型数据转换为双字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (REAL\_TO\_DINT)

[结构化梯形图/FBD]

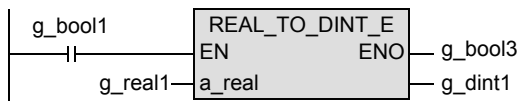


[ST]

```
g_dint1 := REAL_TO_DINT(g_real1);
```

### 2) 带EN/ENO函数 (DINT\_TO\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := REAL_TO_DINT_E(g_bool1, g_real1, g_dint1);
```

## 5.25 REAL\_TO\_STR(E) / 单精度实数型 → 字符串型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	×	×	×	×	×	×	×	×

### 概要

将单精度实数型数据转换为字符串型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
REAL_TO_STR		REAL_TO_STR(_REAL); 例: 标签2:=REAL_TO_STR(标签1);
REAL_TO_STR_E		REAL_TO_STR_E(EN, _REAL, 输出标签); 例: REAL_TO_STR_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

#### 2. 设定数据

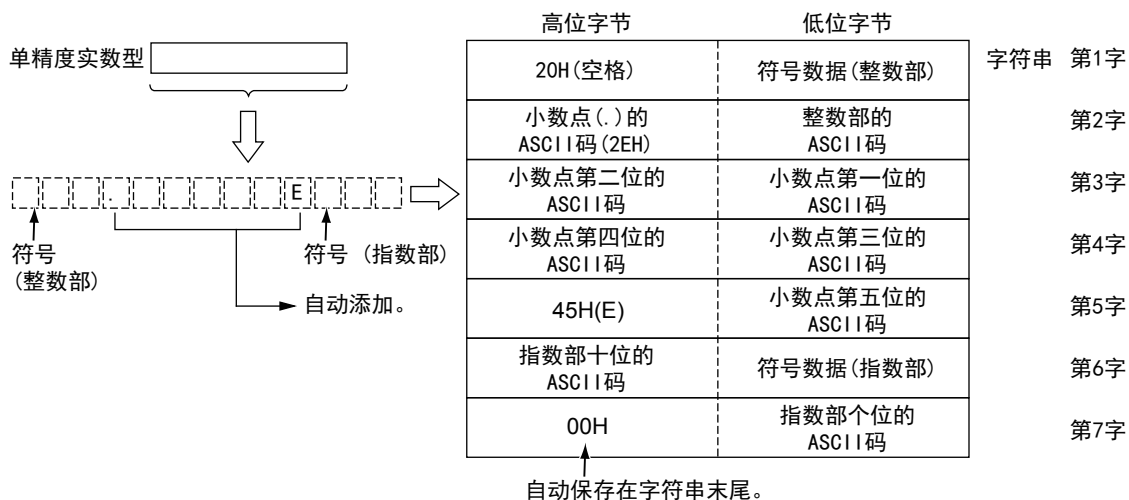
变量	内容	数据类型
输入变量	EN	执行条件
	_REAL (C s)	转换原单精度实数数据
输出变量	ENO	执行状态
	*1 (C d)	转换后的字符串数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

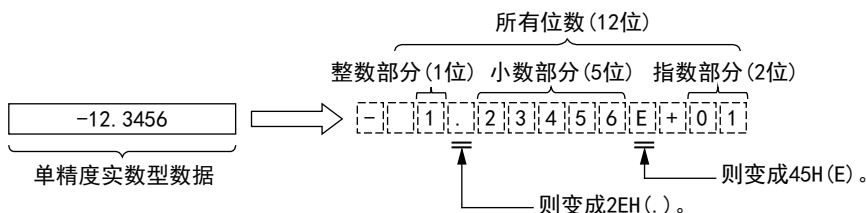


### 功能和动作说明

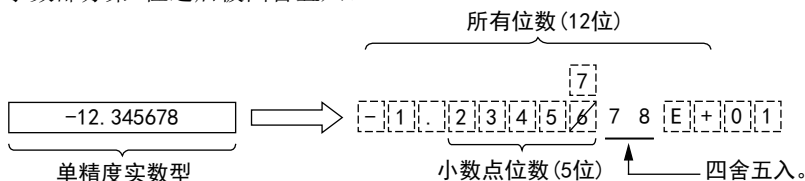
- 1) 将 (s) 指定的软元件的单精度实数型数据转换为字符串型(指数形式)数据, 输出到 (d) 中指定的软元件中。



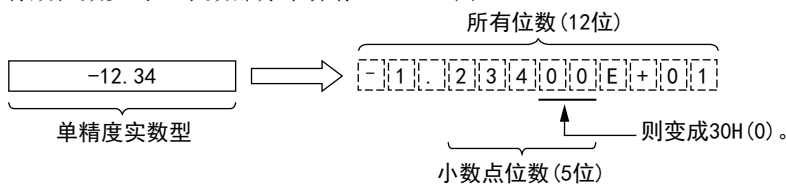
- 2) 转换后的字符串数据, 如下所示被输出到 (d) 中指定的软元件中。  
a) 整数部分、小数部分、指数部分的位数是固定的。(整数部分: 1位, 小数部分: 5位, 指数部分: 2位)  
在第3个字节、第9个字节中, 分别自动保存“2EH”(.)和“45H”(E)。



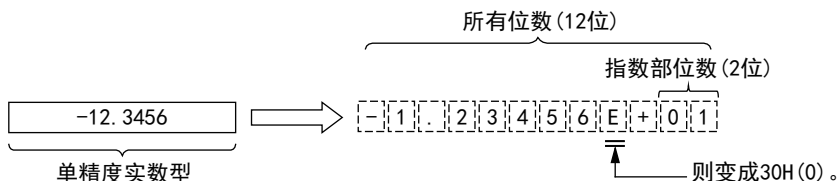
- b) 符号数据(整数部分)中, 输入值为正时保存“20H”(空格), 为负时保存“2DH”(-)。  
c) 小数部分第6位之后被四舍五入。



- d) 有效位数少时, 小数部分中保存“30H”(0)。



- e) 在符号数据(指数部分)中, 指数为正时保存“2BH”(+) , 指数为负时保存“2DH”(-)。  
f) 当指数部分为1位数时, 将“30H”(0)保存到指数部分的十位中。



- 3) 在字符串的最后(第7个字)中, 自动保存“00H”。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据及32位数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据及32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。
- 3) 使用编程工具在(S)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

### 错误

以下一些情况下会出现运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

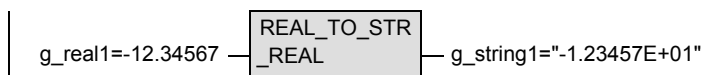
- 1) (S)中指定的软元件的值不在下列范围内时。  
 $0, \pm 2^{-126} \leq ((S) \text{中指定的软元件的值}) \leq \pm 2^{128}$   
 (错误代码：K6706)
- 2) 保存(D)指定字符串的软元件的范围，超出该软元件的范围时。  
 (错误代码：K6706)
- 3) 转换结果超出已指定的所有位数时。  
 (错误代码：K6706)

### 程序举例

将(S)指定的软元件的单精度实数型数据转换为字符串型数据，输出到(D)中指定的软元件中的程序。

- 1) 无EN/ENO函数(REAL\_TO\_STR)

[结构化梯形图/FBD]

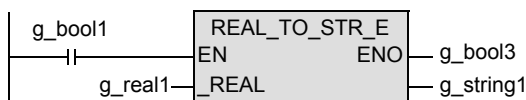


[ST]

```
g_string1 := REAL_TO_STR(g_real1);
```

- 2) 带EN/ENO函数(REAL\_TO\_STR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := REAL_TO_STR_E(g_bool1, g_real1, g_string1);
```

## 5.26 WORD\_TO\_BOOL(E) / 字[无符号] / 位列[16位]型 → 位型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[无符号]/位列[16位]型数据转换为位型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
WORD_TO_BOOL		WORD_TO_BOOL(_WORD); 例: M0:=WORD_TO_BOOL(D0);
WORD_TO_BOOL_E		WORD_TO_BOOL_E(EN, _WORD, 输出标签); 例: WORD_TO_BOOL_E(X000, D0, M0);

\*1. 变为输出变量。

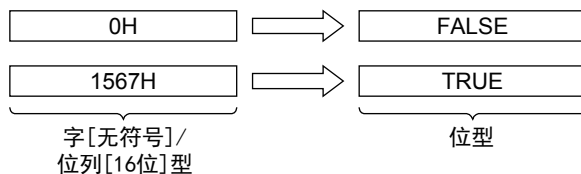
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_WORD (s)	转换原字[无符号]/位列[16位]数据
输出变量	ENO	执行状态
	*1 (d)	转换后的位数据

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字[无符号]/位列[16位]型数据转换为位型数据，输出到(d)中指定的软元件中。



### 注意要点

如果从母线连接函数，则请使用函数名带“\_E”的函数。

1 概要

2 函数、操作符

3 函数的构成

4 函数的阅读方法

5 应用函数(型转换函数)

6 应用函数(单数值变量函数)

7 应用函数(算术运算函数)

8 应用函数(位移位函数)

9 应用函数(位型布尔函数)

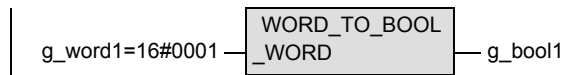
10 应用函数(选择函数)

## 程序举例

将 (s) 指定的软元件的字[无符号]/位列[16位]型数据转换为位型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (WORD\_TO\_BOOL)

[结构化梯形图/FBD]

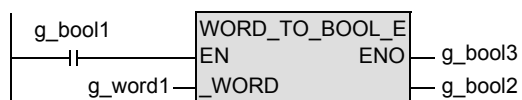


[ST]

g\_bool1 := WORD\_TO\_BOOL(g\_word1);

### 2) 带EN/ENO函数 (WORD\_TO\_BOOL\_E)

[结构化梯形图/FBD]



[ST]

g\_bool3 := WORD\_TO\_BOOL\_E(g\_bool1, g\_word1, g\_bool2);

## 5.27 DWORD\_TO\_BOOL(E) / 双字[无符号] / 位列[32位]型 → 位型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[无符号]/位列[32位]型数据转换为位型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DWORD_TO_BOOL		DWORD_TO_BOOL(DWORD); 例: MO:=DWORD_TO_BOOL(标签);
DWORD_TO_BOOL_E		DWORD_TO_BOOL_E(EN, DWORD, 输出标签); 例: DWORD_TO_BOOL_E(X000, 标签, MO);

\*1. 变为输出变量。

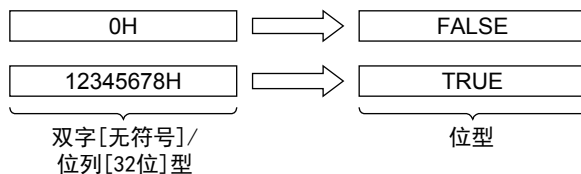
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	DWORD (C <sub>S</sub> )	转换原双字[无符号]/位列[32位]数据
输出变量	ENO	执行状态
	*1 (C <sub>D</sub> )	转换后的位数据

在函数说明中，输入输出变量通过(C)中的描述进行记载。

### 功能和动作说明

将(C<sub>S</sub>)指定的软元件的双字[无符号]/位列[32位]型数据转换为位型数据，输出到(C<sub>D</sub>)中指定的软元件中。



### 注意要点

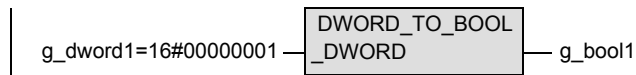
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。

## 程序举例

将 **(s)** 指定的软元件的双字[无符号]/位列[32位]型数据转换为位型数据，输出到 **(d)** 中指定的软元件中的程序。

### 1) 无EN/ENO函数(DWORD\_TO\_BOOL)

[结构化梯形图/FBD]

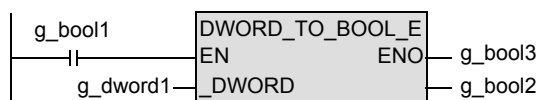


[ST]

g\_bool1 := DWORD\_TO\_BOOL(g\_dword1);

### 2) 带EN/ENO函数(DWORD\_TO\_BOOL\_E)

[结构化梯形图/FBD]



[ST]

g\_bool3 := DWORD\_TO\_BOOL\_E(g\_bool1, g\_dword1, g\_bool2);

## 5.28 WORD\_TO\_INT(E) / 字[无符号] / 位列[16位]型 → 字[带符号]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将字[无符号]/位列[16位]型数据转换为字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
WORD_TO_INT		WORD_TO_INT (_WORD); 例: D10:=WORD_TO_INT (D0);
WORD_TO_INT_E		WORD_TO_INT_E (EN, _WORD, 输出标签); 例: WORD_TO_INT_E (X000, D0, D10);

\*1. 变为输出变量。

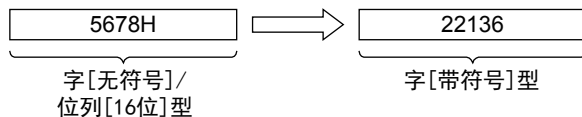
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_WORD (S)	转换原字[无符号]/位列[16位]数据	字[无符号]/位列[16位]
输出变量	ENO	执行状态	位
	*1 (D)	转换后的字[带符号]数据	字[带符号]

在函数说明中，输入输出变量通过(S)中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的字[无符号]/位列[16位]型数据转换为字[带符号]型数据，输出到(D)中指定的软元件中。



### 注意要点

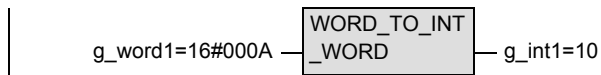
如果从母线连接函数，则请使用函数名带“\_E”的函数。

### 程序举例

将 (s) 指定的软元件的字[无符号]/位列[16位]型数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

#### 1) 无EN/ENO函数 (WORD\_TO\_INT)

[结构化梯形图/FBD]

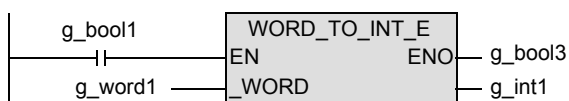


[ST]

```
g_int1 := WORD_TO_INT(g_word1);
```

#### 2) 带EN/ENO函数 (WORD\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := WORD_TO_INT_E(g_bool1, g_word1, g_int1);
```



## 5.29 WORD\_TO\_DINT(E) / 字[无符号] / 位列[16位]型 → 双字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[无符号]/位列[16位]型数据转换为双字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
WORD_TO_DINT		WORD_TO_DINT(_WORD); 例: 标签:=WORD_TO_DINT(D0);
WORD_TO_DINT_E		WORD_TO_DINT_E(EN, _WORD, 输出标签); 例: WORD_TO_DINT_E(X000, D0, 标签);

\*1. 变为输出变量。

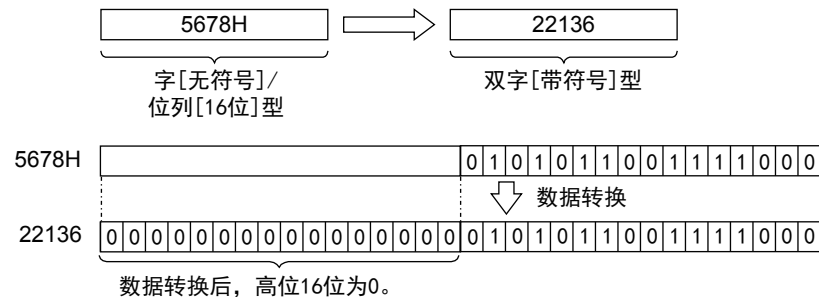
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_WORD (C)	转换原字[无符号]/位列[16位]数据
输出变量	ENO	执行状态
	*1 (D)	转换后的双字[带符号]数据

在函数说明中，输入输出变量通过(C)中的描述进行记载。

### 功能和动作说明

将(C)指定的软元件的字[无符号]/位列[16位]型数据转换为双字[带符号]型数据，输出到(D)中指定的软元件中。



### 注意要点

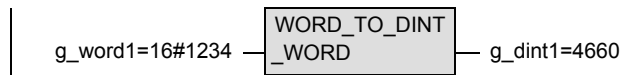
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 **(s)** 指定的软元件的字[无符号]/位列[16位]型数据转换为双字[带符号]型数据，输出到 **(d)** 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (WORD\_TO\_DINT)

[结构化梯形图/FBD]

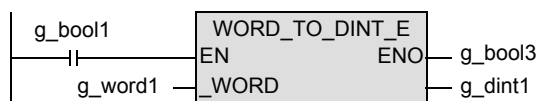


[ST]

```
g_dint1 := WORD_TO_DINT(g_word1);
```

### 2) 带EN/ENO函数 (WORD\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := WORD_TO_DINT_E(g_bool1, g_word1, g_dint1);
```

## 5.30 DWORD\_TO\_INT(E) / 双字[无符号] / 位列[32位]型 → 字[带符号]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[无符号]/位列[32位]型数据转换为字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DWORD_TO_INT		DWORD_TO_INT (_DWORD); 例: D10:=DWORD_TO_INT(标签);
DWORD_TO_INT_E		DWORD_TO_INT_E(EN, _DWORD, 输出标签); 例: DWORD_TO_INT_E(X000, 标签, D10);

\*1. 变为输出变量。

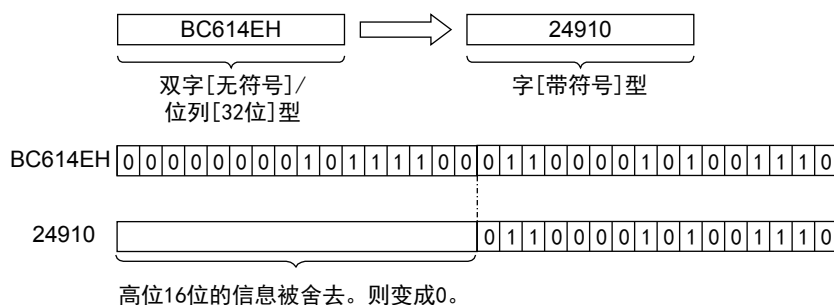
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_DWORD (C)	转换原双字[无符号]/位列[32位]数据
输出变量	ENO	执行状态
	*1 (D)	转换后的字[带符号]数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(C)指定的软元件的双字[无符号]/位列[32位]型数据转换为字[带符号]型数据，输出到(D)中指定的软元件中。



### 注意要点

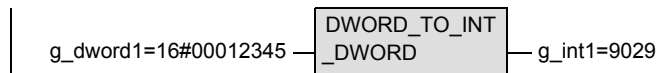
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的双字[无符号]/位列[32位]型数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (DWORD\_TO\_INT)

[结构化梯形图/FBD]

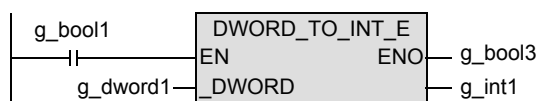


[ST]

```
g_int1 := DWORD_TO_INT(g_dword1);
```

### 2) 带EN/ENO函数 (DWORD\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DWORD_TO_INT_E(g_bool1, g_dword1, g_int1);
```

## 5.31 DWORD\_TO\_DINT(E) / 双字[无符号] / 位列[32位] 型 → 双字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[无符号]/位列[32位]型数据转换为双字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DWORD_TO_DINT		DWORD_TO_DINT(_DWORD); 例: 标签2:=DWORD_TO_DINT(标签1);
DWORD_TO_DINT_E		DWORD_TO_DINT_E(EN,_DWORD,输出标签); 例: DWORD_TO_DINT_E(X000,标签1,标签2);

\*1. 变为输出变量。

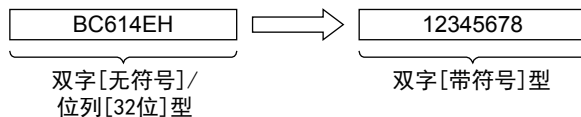
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_DWORD (C)	转换原双字[无符号]/位列[32位]数据	双字[无符号]/位列[32位]
输出变量	ENO	执行状态	位
	*1 (D)	转换后的双字[带符号]数据	双字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(C)指定的软元件的双字[无符号]/位列[32位]型数据转换为双字[带符号]型数据，输出到(D)中指定的软元件中。



### 注意要点

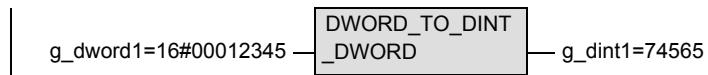
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 **(s)** 指定的软元件的双字[无符号]/位列[32位]型数据转换为双字[带符号]型数据，输出到 **(d)** 中指定的软元件中的程序。

### 1) 无EN/ENO函数(DWORD\_TO\_DINT)

[结构化梯形图/FBD]

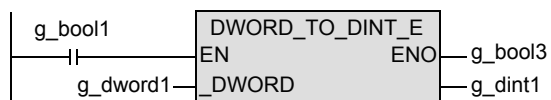


[ST]

```
g_dint1 := DWORD_TO_DINT(g_dword1);
```

### 2) 带EN/ENO函数(DWORD\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DWORD_TO_DINT_E(g_bool1, g_dword1, g_dint1);
```

## 5.32 WORD\_TO\_DWORD(E) / 字[无符号] / 位列[16位]型 → 双字[无符号] / 位列[32位]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[无符号]/位列[16位]型数据转换为双字[无符号]/位列[32位]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
WORD_TO_DWORD		WORD_TO_DWORD(_WORD); 例: 标签:=WORD_TO_DWORD(D0);
WORD_TO_DWORD_E		WORD_TO_DWORD_E(EN, _WORD, 输出标签); 例: WORD_TO_DWORD_E(X000, D0, 标签);

\*1. 变为输出变量。

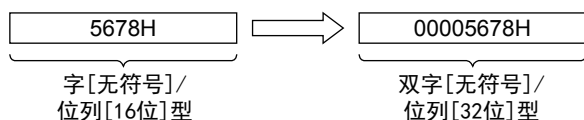
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_WORD (s)	转换原字[无符号]/位列[16位]数据	字[无符号]/位列[16位]
输出变量	ENO	执行状态	位
	*1 (d)	转换后的双字[无符号]/位列[32位]数据	双字[无符号]/位列[32位]

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字[无符号]/位列[16位]型数据转换为双字[无符号]/位列[32位]型数据，输出到(d)中指定的软元件中。数据转换后，高位16位为0。



### 注意要点

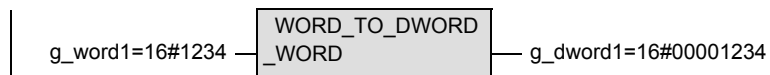
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的字[无符号]/位列[16位]型数据转换为双字[无符号]/位列[32位]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (WORD\_TO\_DWORD)

[结构化梯形图/FBD]

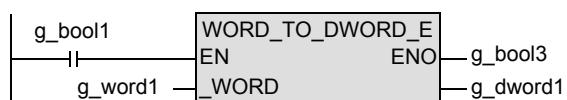


[ST]

g\_dword1 := WORD\_TO\_DWORD(g\_word1);

### 2) 带EN/ENO函数 (WORD\_TO\_DWORD\_E)

[结构化梯形图/FBD]



[ST]

g\_bool3 := WORD\_TO\_DWORD\_E(g\_bool1, g\_word1, g\_dword1);



## 5.33 DWORD\_TO\_WORD(E) / 双字[无符号] / 位列[32位]型 → 字[无符号] / 位列[16位]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[无符号]/位列[32位]型数据转换为字[无符号]/位列[16位]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DWORD_TO_WORD		DWORD_TO_WORD(_DWORD); 例: D10:=DWORD_TO_WORD(标签);
DWORD_TO_WORD_E		DWORD_TO_WORD_E(EN, _DWORD, 输出标签); 例: DWORD_TO_WORD_E(X000, 标签, D10);

\*1. 变为输出变量。

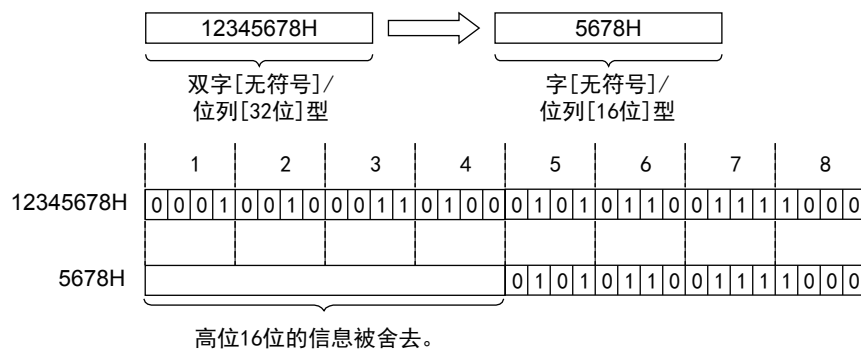
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_DWORD (s)	转换原双字[无符号]/位列[32位]数据	双字[无符号]/位列[32位]
输出变量	ENO	执行状态	位
	*1 (d)	转换后的字[无符号]/位列[16位]数据	字[无符号]/位列[16位]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的双字[无符号]/位列[32位]型数据转换为字[无符号]/位列[16位]型数据，输出到(d)中指定的软元件中。



### 注意要点

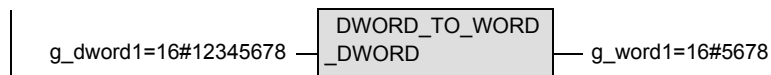
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的双字[无符号]/位列[32位]型数据转换为字[无符号]/位列[16位]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数(DWORD\_TO\_WORD)

[结构化梯形图/FBD]

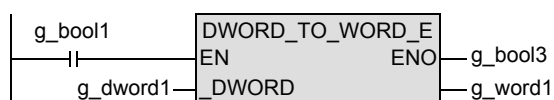


[ST]

```
g_word1 := DWORD_TO_WORD(g_dword1);
```

### 2) 带EN/ENO函数(DWORD\_TO\_WORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DWORD_TO_WORD_E(g_bool1, g_dword1, g_word1);
```

## 5.34 WORD\_TO\_TIME(E) / 字[无符号] / 位列[16位]型 → 时间型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[无符号]/位列[16位]型数据转换为时间型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
WORD_TO_TIME		WORD_TO_TIME(WORD); 例: 标签:=WORD_TO_TIME(D0);
WORD_TO_TIME_E		WORD_TO_TIME_E(EN, WORD, 输出标签); 例: WORD_TO_TIME_E(X000, D0, 标签);

\*1. 变为输出变量。

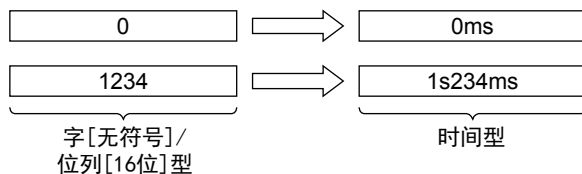
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	WORD (S)	转换原字[无符号]/位列[16位]数据
输出变量	ENO	执行状态
	*1 (D)	转换后的时间数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的字[无符号]/位列[16位]型数据转换为时间型数据，输出到(D)中指定的软元件中。



### 注意要点

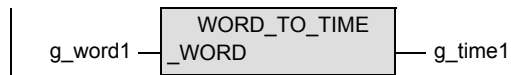
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 **(s)** 指定的软元件的字[无符号]/位列[16位]型数据转换为时间型数据，输出到 **(d)** 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (WORD\_TO\_TIME)

[结构化梯形图/FBD]

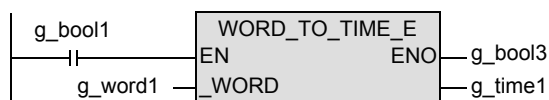


[ST]

```
g_time1 := WORD_TO_TIME(g_word1);
```

### 2) 带EN/ENO函数 (WORD\_TO\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := WORD_TO_TIME_E(g_bool1, g_word1, g_time1);
```

## 5.35 DWORD\_TO\_TIME(E) / 双字[无符号] / 位列[32位]型 → 时间型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[无符号]/位列[32位]型数据转换为时间型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DWORD_TO_TIME		DWORD_TO_TIME (DWORD); 例: 标签2:=DWORD_TO_TIME(标签1);
DWORD_TO_TIME_E		DWORD_TO_TIME_E(EN, DWORD, 输出标签); 例: DWORD_TO_TIME_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

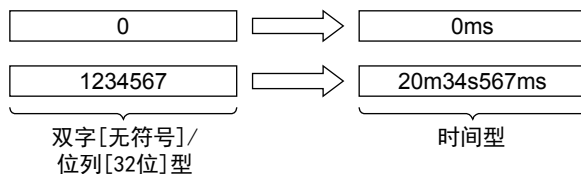
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	DWORD (C)	转换原双字[无符号]/位列[32位]数据
输出变量	ENO	执行状态
	*1 (D)	转换后的时间数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(S)指定的软元件的双字[无符号]/位列[32位]型数据转换为时间型数据，输出到(D)中指定的软元件中。



### 注意要点

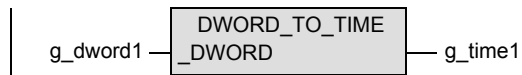
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的双字[无符号]/位列[32位]型数据转换为时间型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (DWORD\_TO\_TIME)

[结构化梯形图/FBD]

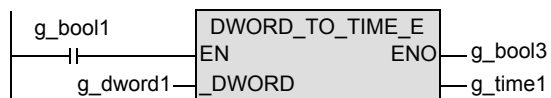


[ST]

```
g_time1 := DWORD_TO_TIME(g_dword1);
```

### 2) 带EN/ENO函数 (DWORD\_TO\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := DWORD_TO_TIME_E(g_bool1, g_dword1, g_time1);
```

## 5.36 STR\_TO\_BOOL(E) / 字符串型 → 位型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

将字符串型数据转换为位型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
STR_TO_BOOL		STR_TO_BOOL(_STRING); 例: M0:=STR_TO_BOOL(标签);
STR_TO_BOOL_E		STR_TO_BOOL_E(EN,_STRING,输出标签); 例: STR_TO_BOOL_E(X000,标签,M0);

\*1. 变为输出变量。

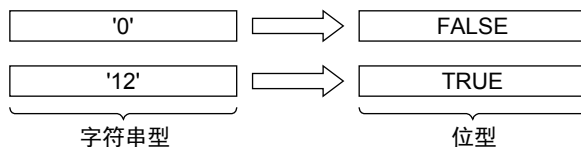
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_STRING (s)	转换原字符串数据
输出变量	ENO	执行状态
	*1 (d)	转换后的位数据

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字符串型数据转换为位型数据，输出到(d)中指定的软元件中。



### 注意要点

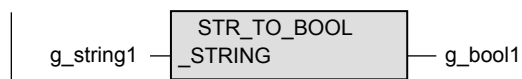
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

## 程序举例

将 **(s)** 指定的软元件的字符串型数据转换为位型数据，输出到 **(d)** 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (STR\_TO\_BOOL)

[结构化梯形图/FBD]

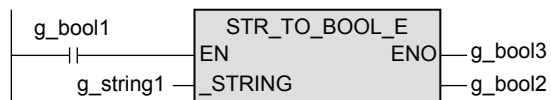


[ST]

```
g_bool1 := STR_TO_BOOL(g_string1);
```

### 2) 带EN/ENO函数 (STR\_TO\_BOOL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := STR_TO_BOOL_E(g_bool1, g_string1, g_bool2);
```



## 5.37 STR\_TO\_INT(E) / 字符串型 → 字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	×	×	×	×	×	×	×	×

### 概要

将字符串型数据转换为字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
STR_TO_INT		STR_TO_INT(_STRING); 例: D10:=STR_TO_INT(标签);
STR_TO_INT_E		STR_TO_INT_E(EN,_STRING,输出标签); 例: STR_TO_INT_E(X000,标签,D10);

\*1. 变为输出变量。

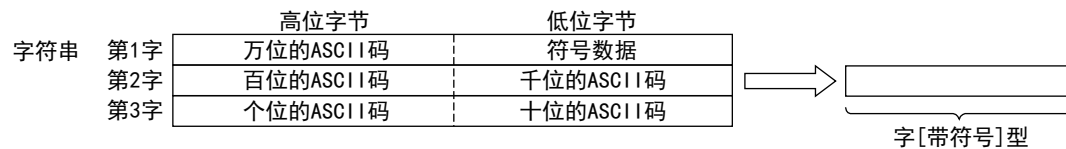
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_STRING (s)	转换原字符串数据	字符串
输出变量	ENO	执行状态	位
	*1 (d)	转换后的字[带符号]数据	字[带符号]

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字符串型数据(3字)转换为字[带符号]型数据，输出到(d)中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

## 错误

- 1) (S) 指定的软元件的第1个字的符号数据(低位字节)为“20H(空格)”、“2DH(-)”以外时。  
(错误代码：K6706)
- 2) (S) ~ (S)+2的各位数(位)的ASCII码为“30H”~“39H”、“20H(空格)”、“00H(NULL)”以外的值时。  
(错误代码：K6706)
- 3) (S) ~ (S)+2的数值范围为下述的范围以外时。  
-32768~32767  
(错误代码：K6706)
- 4) (S) ~ (S)+2超出软元件范围时。  
(错误代码：K6706)

## 程序举例

将(S)指定的软元件的字符串型数据转换为字[带符号]型数据，输出到(D)中指定的软元件中的程序。

- 1) 无EN/ENO函数(STR\_TO\_INT)

[结构化梯形图/FBD]

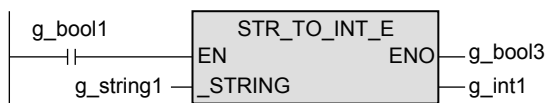


[ST]

```
g_int1 := STR_TO_INT(g_string1);
```

- 2) 带EN/ENO函数(STR\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := STR_TO_INT_E(g_bool1, g_string1, g_int1);
```

## 5.38 STR\_TO\_DINT(E) / 字符串型 → 双字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	×	×	×	×	×	×	×	×

### 概要

将字符串型数据转换为双字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
STR_TO_DINT		STR_TO_DINT( _STRING ); 例: 标签2:=STR_TO_DINT(标签1);
STR_TO_DINT_E		STR_TO_DINT_E(EN, _STRING, 输出标签); 例: STR_TO_DINT_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_STRING ( <u>S</u> )	转换原字符串数据	字符串
输出变量	ENO	执行状态	位
	*1 ( <u>D</u> )	转换后的双字[带符号]数据	双字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将 S 指定的软元件的字符串型数据(6双字)转换为双字[带符号]型数据，输出到 D 中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据及32位数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据及32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 错误

- 1) (S) 指定的软元件的第1个字的符号数据(低位字节)为“20H(空格)”，“2DH(-)”以外时。  
(错误代码：K6706)
- 2) (S) ~ (S)+5的各位数(位)的ASCII码为“30H”~“39H”、“20H(空格)”、“00H(NULL)”以外的值时。  
(错误代码：K6706)
- 3) (S) ~ (S)+5的数值范围为下述的范围以外时。  
-2, 147, 483, 648 ~ 2, 147, 483, 647  
(错误代码：K6706)
- 4) (S) ~ (S)+5超出软元件范围时。  
(错误代码：K6706)

## 程序举例

将(S)指定的软元件的字符串型数据转换为双字[带符号]型数据，输出到(D)中指定的软元件中的程序。

- 1) 无EN/ENO函数(STR\_TO\_DINT)

[结构化梯形图/FBD]

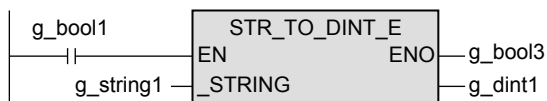


[ST]

```
g_dint1 := STR_TO_DINT(g_string1);
```

- 2) 带EN/ENO函数(STR\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := STR_TO_DINT_E(g_bool1, g_string1, g_dint1);
```

## 5.39 STR\_TO\_REAL(E) / 字符串型 → 单精度实数型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

将字符串型数据转换为单精度实数型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
STR_TO_REAL		STR_TO_REAL(_STRING); 例: 标签2:=STR_TO_REAL(标签1);
STR_TO_REAL_E		STR_TO_REAL_E(EN,_STRING,输出标签); 例: STR_TO_REAL_E(X000,标签1,标签2);

\*1. 变为输出变量。

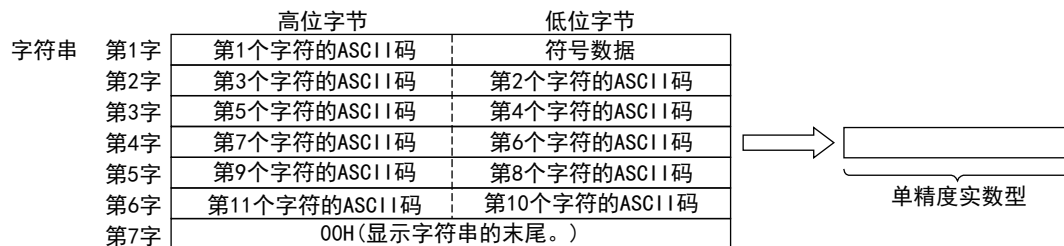
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_STRING (s)	转换原字符串数据
输出变量	ENO	执行状态
	*1 (d)	转换后的单精度实数数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

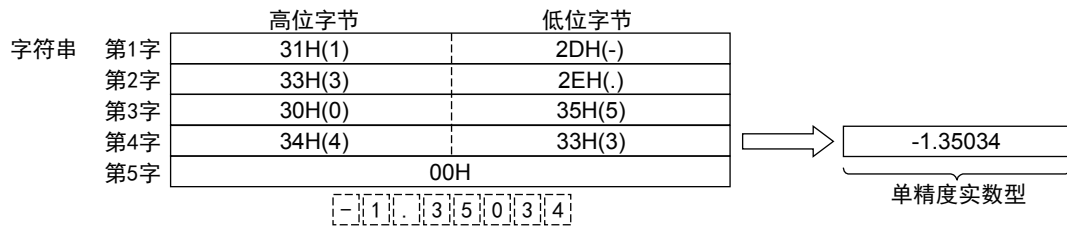
### 功能和动作说明

- 1) 将(s)指定的软元件的字符串型(小数点形式/指数形式)数据转换为单精度实数型数据，输出到(d)中指定的软元件中。



2) 字符串型数据可以转换为小数点形式或指数形式中的任意一种。

a) 小数点形式时

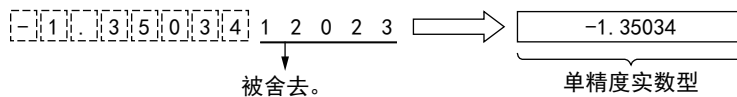


b) 指数形式时

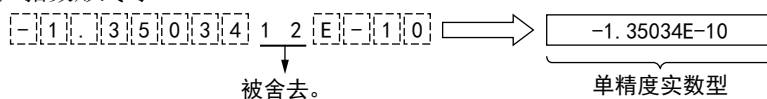


3) 字符串型数据除了符号、小数点、指数部分以外，有效位数为6位，舍去第7位以后的数据后，再进行转换。

a) 小数点形式时



b) 指数形式时

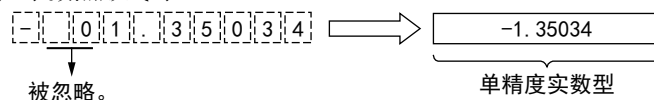


4) 在小数点形式中，将符号指定为“2BH”(+)或省略符号，则作为正值转换。此外，将符号指定为“2DH”(-)则作为负的值转换。

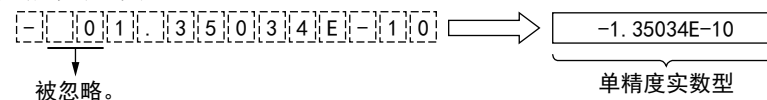
5) 用指数形式在指数部分的符号内指定“2BH”(+)，省略符号后作为正值转换。对指数部分的符号进行指定，则作为负的值转换。

6) 字符串型数据中，在最初的“0”以外的数值之间如果存在“20H”(空格)或是“30H”(0)时，会忽略“20H”、“30H”而进行转换。

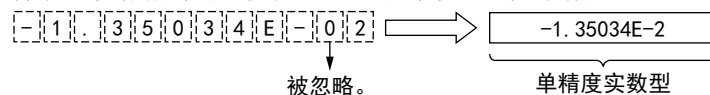
a) 小数点形式时



b) 指数形式时



7) 字符串型数据(指数形式)中，“E”和数值之间如果存在“30H”(0)，则忽略“30H”而进行转换。



8) 字符串中如果包含“20H”(空格)时，则忽略“20H”而进行转换。

9) 字符串型数据最大可以输入24个字符。  
字符串中的“20H”(空格)、“30H”(0)也作为一个字符来计算。

## 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据及32位数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据及32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

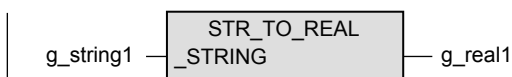
- 1) 整数部分、小数部分中存在“30H”(0)~“39H”(9)以外的字符时。  
(错误代码：K6706)
- 2) (S)指定的字符串中存在2个以上的“2EH”(.)时。  
(错误代码：K6706)
- 3) 指数部分中有“45H”(E)、“2BH”(+)、“2DH”(-)以外的字符或是有多个指数部分时。  
(错误代码：K6706)
- 4) (S)以后的字符数为0或是超出了24个字符时。  
(错误代码：K6706)

## 程序举例

将(S)指定的软元件的字符串型数据转换为单精度实数型数据，输出到(D)中指定的软元件中的程序。

- 1) 无EN/ENO函数(STR\_TO\_REAL)

[结构化梯形图/FBD]

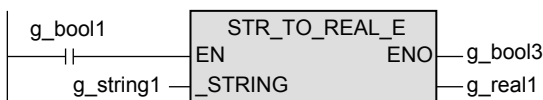


[ST]

```
g_real1 := STR_TO_REAL(g_string1);
```

- 2) 带EN/ENO函数(STR\_TO\_REAL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := STR_TO_REAL_E(g_bool1, g_string1, g_real1);
```

## 5.40 STR\_TO\_TIME(E) / 字符串型 → 时间型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

将字符串型数据转换为时间型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
STR_TO_TIME		STR_TO_TIME(_STRING); 例: 标签2:=STR_TO_TIME(标签1);
STR_TO_TIME_E		STR_TO_TIME_E(EN, _STRING, 输出标签); 例: STR_TO_TIME_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

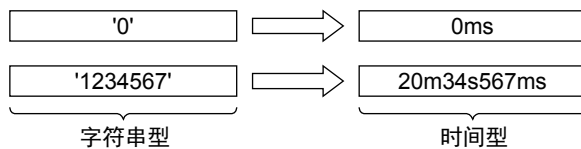
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_STRING (s)	转换原字符串数据	字符串
输出变量	ENO	执行状态	位
	*1 (d)	转换后的时间数据	时间

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字符串型数据转换为时间型数据，输出到(d)中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据及32位数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据及32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。



## 错误

下面的情况下会发生运算错误，错误标志位M8067为ON，错误代码保存在D8067中。

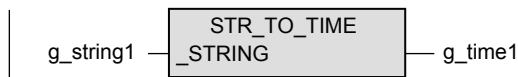
- 1) (S) 指定的数值数据的符号数据为“20H”（空格），“2DH”（-）以外时。  
(错误代码：K6706)
- 2) (S) 中指定的字符串数据的各位数(位)的ASCII码为“30H”～“39H”、“20H”（空格）、“00H”（NULL）以外的值时。  
(错误代码：K6706)
- 3) (S) 指定的数值范围为下述的范围以外时。  
-2, 147, 483, 648～2, 147, 483, 647  
(错误代码：K6706)

## 程序举例

将(S) 指定的软元件的字符串型数据转换为时间型数据，输出到(D) 中指定的软元件中的程序。

- 1) 无EN/ENO函数 (STR\_TO\_TIME)

[结构化梯形图/FBD]

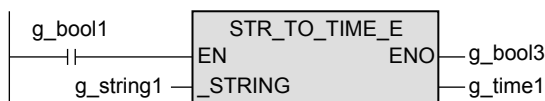


[ST]

```
g_time1 := STR_TO_TIME(g_string1);
```

- 2) 带EN/ENO函数 (STR\_TO\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := STR_TO_TIME_E(g_bool1, g_string1, g_time1);
```

## 5.41 BCD\_TO\_INT(E) / BCD型 → 字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将BCD数据转换为字[带符号]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BCD_TO_INT		BCD_TO_INT(_BCD); 例: D10:=BCD_TO_INT(D0);
BCD_TO_INT_E		BCD_TO_INT_E(EN, _BCD, 输出标签); 例: BCD_TO_INT_E(X000, D0, D10);

\*1. 变为输出变量。

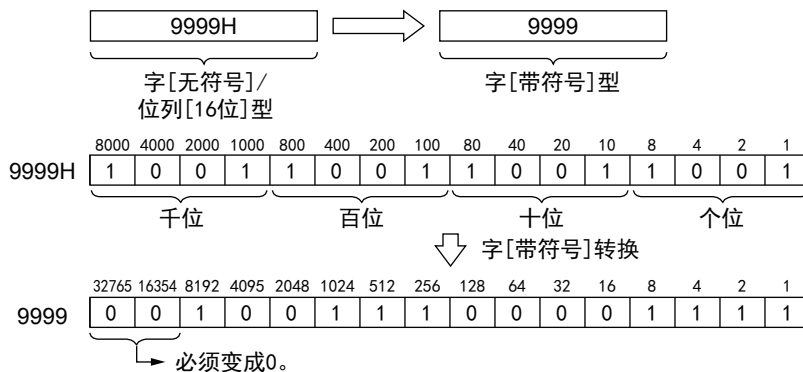
### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	_BCD (s)	字[无符号]/位列[16位]
输出变量	ENO	位
	*1 (d)	字[带符号]

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的BCD数据转换为字[带符号]型数据，输出到(d)中指定的软元件中。



### 注意要点

如果从母线连接函数，则请使用函数名带“\_E”的函数。

### 错误

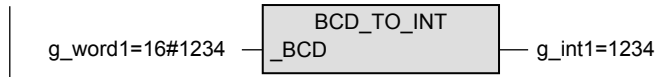
非BCD(10进制数)时，M8067(运算错误)为ON。

## 程序举例

将 (s) 指定的软元件的BCD数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (BCD\_TO\_INT)

[结构化梯形图/FBD]

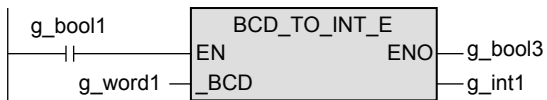


[ST]

```
g_int1 := BCD_TO_INT(g_word1);
```

### 2) 带EN/ENO函数 (BCD\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BCD_TO_INT_E(g_bool1, g_word1, g_int1);
```

## 5.42 BCD\_TO\_DINT(E) / BCD型 → 双字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将BCD数据转换为双字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BCD_TO_DINT		BCD_TO_DINT(BCD); 例: 标签2:=BCD_TO_DINT(标签1);
BCD_TO_DINT_E		BCD_TO_DINT_E(EN, BCD, 输出标签); 例: BCD_TO_DINT_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

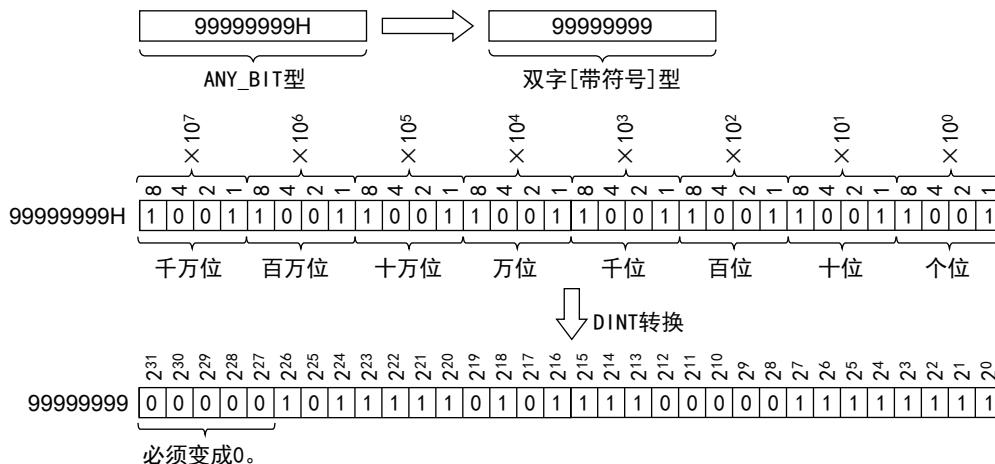
#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
_BCD (s)	转换原BCD数据	ANY_BIT
ENO	执行状态	位
*1 (d)	转换后的双字[带符号]数据	双字[带符号]

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的BCD数据转换为双字[带符号]型数据，输出到(d)指定的软元件中。



### 注意要点

如果从母线连接函数，则请使用函数名带“\_E”的函数。

### 错误

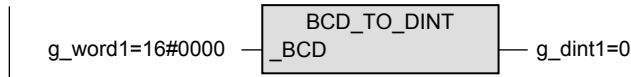
非BCD(10进制数)时，M8067(运算错误)为ON。

## 程序举例

将 (s) 指定的软元件的BCD数据转换为双字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (BCD\_TO\_DINT)

[结构化梯形图/FBD]

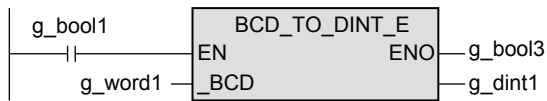


[ST]

```
g_dint1 := BCD_TO_DINT(g_word1);
```

### 2) 带EN/ENO函数 (BCD\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := BCD_TO_DINT_E(g_bool1, g_word1, g_dint1);
```

## 5.43 TIME\_TO\_BOOL(E) / 时间型 → 位型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将时间型数据转换为位型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIME_TO_BOOL		TIME_TO_BOOL(_TIME); 例: M0:=TIME_TO_BOOL(标签);
TIME_TO_BOOL_E		TIME_TO_BOOL_E(EN, _TIME, 输出标签); 例: TIME_TO_BOOL_E(X000, 标签, M0);

\*1. 变为输出变量。

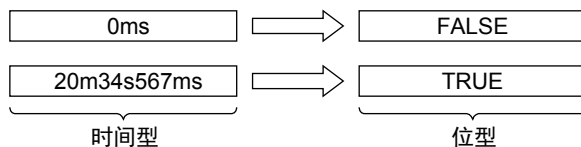
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_TIME (s)	转换原时间数据
输出变量	ENO	执行状态
	*1 (d)	转换后的位数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的时间型数据转换为位型数据，输出到(d)中指定的软元件中。



### 注意要点

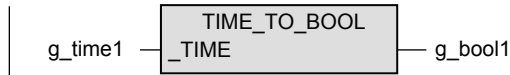
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的时间型数据转换为位型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (TIME\_TO\_BOOL)

[结构化梯形图/FBD]

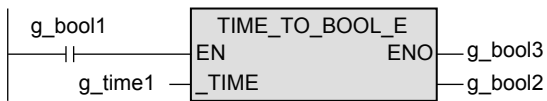


[ST]

```
g_bool1 := TIME_TO_BOOL(g_time1);
```

### 2) 带EN/ENO函数 (TIME\_TO\_BOOL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := TIME_TO_BOOL_E(g_bool1, g_time1, g_bool2);
```

## 5.44 TIME\_TO\_INT(E) / 时间型 → 字[带符号]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将时间型数据转换为字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIME_TO_INT		TIME_TO_INT(_TIME); 例: D10:=TIME_TO_INT(标签);
TIME_TO_INT_E		TIME_TO_INT_E(EN, _TIME, 输出标签); 例: TIME_TO_INT_E(X000, 标签, D10);

\*1. 变为输出变量。

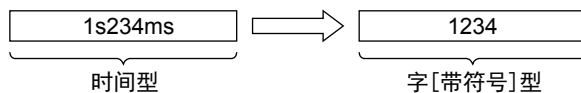
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_TIME (s)	转换原时间数据	时间
输出变量	ENO	执行状态	位
	*1 (d)	转换后的字[带符号]数据	字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的时间型数据转换为字[带符号]型数据，输出到(d)中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

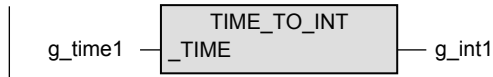


## 程序举例

将 (s) 指定的软元件的时间型数据转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (TIME\_TO\_INT)

[结构化梯形图/FBD]

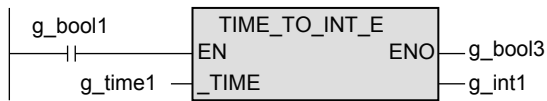


[ST]

```
g_int1 := TIME_TO_INT(g_time1);
```

### 2) 带EN/ENO函数 (TIME\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := TIME_TO_INT_E(g_bool1, g_time1, g_int1);
```

## 5.45 TIME\_TO\_DINT(E) / 时间型 → 双字[带符号]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将时间型数据转换为双字[带符号]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIME_TO_DINT		TIME_TO_DINT ( _TIME ); 例: 标签2:=TIME_TO_DINT(标签1);
TIME_TO_DINT_E		TIME_TO_DINT_E(EN, _TIME, 输出标签); 例: TIME_TO_DINT_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

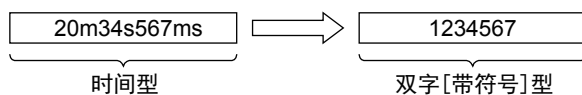
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_TIME ( (s) )	转换原时间数据	时间
输出变量	ENO	执行状态	位
	*1 ( (d) )	转换后的双字[带符号]数据	双字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将 (s) 指定的软元件的时间型数据转换为双字[带符号]型数据，输出到 (d) 中指定的软元件中。



### 注意要点

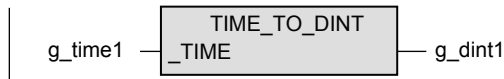
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理 32 位数据时，不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时，请使用标签。  
但是，32 位计数器为 32 位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的时间型数据转换为双字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (TIME\_TO\_DINT)

[结构化梯形图/FBD]

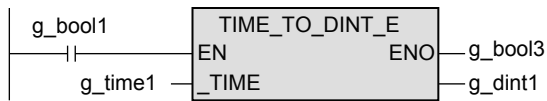


[ST]

```
g_dint1 := TIME_TO_DINT(g_time1);
```

### 2) 带EN/ENO函数 (TIME\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := TIME_TO_DINT_E(g_bool1, g_time1, g_dint1);
```

## 5.46 TIME\_TO\_STR(E) / 时间型 → 字符串型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

将时间型数据转换为字符串型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIME_TO_STR		TIME_TO_STR(_TIME); 例: 标签2:=TIME_TO_STR(标签1);
TIME_TO_STR_E		TIME_TO_STR_E(EN, _TIME, 输出标签); 例: TIME_TO_STR_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

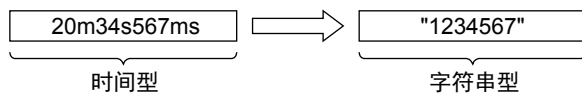
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_TIME ( (s) )	转换原时间数据	时间
输出变量	ENO	执行状态	位
	*1 ( (d) )	转换后的字符串数据	字符串

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将 (s) 指定的软元件的时间型数据转换为字符串型数据，输出到 (d) 中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

### 错误

以下一些情况下会出现运算错误，错误标志位M8067为0N，错误代码保存在D8067中。

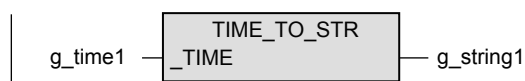
- 1) (d) 指定的软元件的占用点数，超出了软元件的范围时。

## 程序举例

将 (s) 指定的软元件的时间型数据转换为字符串型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (TIME\_TO\_STR)

[结构化梯形图/FBD]

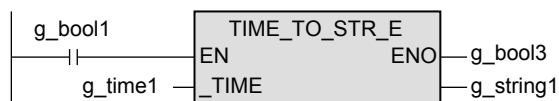


[ST]

```
g_string1 := TIME_TO_STR(g_time1);
```

### 2) 带EN/ENO函数 (TIME\_TO\_STR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := TIME_TO_STR_E(g_bool1, g_time1, g_string1);
```

## 5.47 TIME\_TO\_WORD(E) / 时间型 → 字[无符号] / 位列[16位]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将时间型数据转换为字[无符号]/位列[16位]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIME_TO_WORD		TIME_TO_WORD( TIME ); 例: D10:=TIME_TO_WORD(标签);
TIME_TO_WORD_E		TIME_TO_WORD_E(EN, TIME, 输出标签); 例: TIME_TO_WORD_E(X000, 标签, D10);

\*1. 变为输出变量。

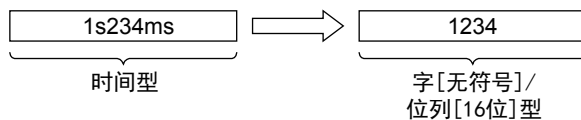
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_TIME ( (s) )	转换原时间数据	时间
输出变量	ENO	执行状态	位
	*1 ( (d) )	转换后的字[无符号]/位列[16位]数据	字[无符号]/位列[16位]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将 (s) 指定的软元件的时间型数据转换为字[无符号]/位列[16位]型数据，输出到 (d) 中指定的软元件中。



### 注意要点

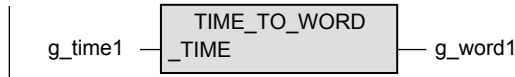
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的时间型数据转换为字[无符号]/位列[16位]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (TIME\_TO\_WORD)

[结构化梯形图/FBD]

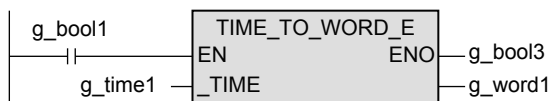


[ST]

```
g_word1 := TIME_TO_WORD(g_time1);
```

### 2) 带EN/ENO函数 (TIME\_TO\_WORD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := TIME_TO_WORD_E(g_bool1, g_time1, g_word1);
```

## 5.48 TIME\_TO\_DWORD(E) / 时间型 → 双字[无符号] / 位列[32位]型转换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将时间型数据转换为双字[无符号]/位列[32位]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIME_TO_DWORD		TIME_TO_DWORD( TIME ); 例: 标签2:=TIME_TO_DWORD(标签1);
TIME_TO_DWORD_E		TIME_TO_DWORD_E(EN, TIME, 输出标签); 例: TIME_TO_DWORD_E(X000, 标签1, 标签2);

\*1. 变为输出变量。

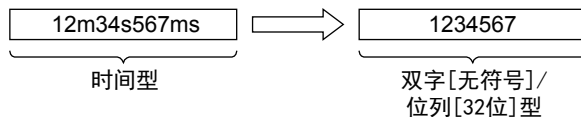
### 2. 设定数据

变量	内容	数据类型
输入变量	EN	位
	_TIME ( (s) )	时间
输出变量	ENO	位
	*1 ( (d) )	双字[无符号]/位列[32位]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的时间型数据转换为双字[无符号]/位列[32位]型数据，输出到(d)中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

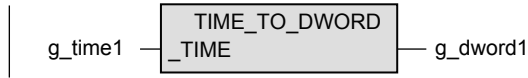


## 程序举例

将 (s) 指定的软元件的时间型数据转换为双字[无符号]/位列[32位]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (TIME\_TO\_DWORD)

[结构化梯形图/FBD]

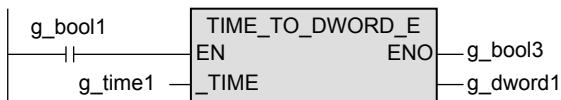


[ST]

g\_dword1 := TIME\_TO\_DWORD(g\_time1);

### 2) 带EN/ENO函数 (TIME\_TO\_DWORD\_E)

[结构化梯形图/FBD]



[ST]

g\_bool3 := TIME\_TO\_DWORD\_E(g\_bool1, g\_time1, g\_dword1);

## 5.49 BITARR\_TO\_INT(E) / 位阵列 → 字[带符号]型、字[无符号] / 位列[16位]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将使用位阵列指定的位数转换为字[带符号]型、字[无符号]/位列[16位]型数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BITARR_TO_INT		BITARR_TO_INT(BitArr, n); 例: D10:= BITARR_TO_INT(标签1[*2], K4);
BITARR_TO_INT_E		BITARR_TO_INT_E(EN, BitArr, n, 输出标签); 例: BITARR_TO_INT_E(X000, 标签1[*2], K4, D10);

- \*1. 变为输出变量。
- \*2. 指定阵列要素。

### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	BitArr( <u>s</u> )	转换原位排列要素的起始位	位
	<u>n</u>	指定的位数	字[带符号]
输出变量	ENO	执行状态	位
	*1 ( <u>d</u> )	转换后的字[带符号]型、字[无符号]/位列[16位]数据	ANY16

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的位阵列要素作为起始，将(n)指定的位数转换为字[带符号]型、字[无符号]/位列[16位]型数据后，输出到(d)指定的软元件中。

(n)中，仅可以指定常数4、8、12、16。

在指定的位数以上的输出位中设定为0。

### 注意要点

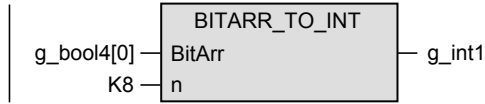
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据、阵列数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据、阵列数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
使用标签时，使用全局标签。

## 程序举例

在 (s) 中指定的软元件的位阵列中，将0号开始的8位转换为字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (BITARR\_TO\_INT)

[结构化梯形图/FBD]

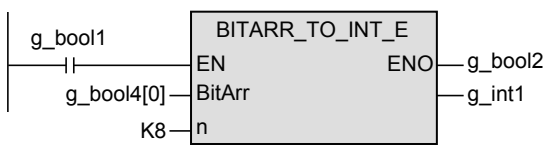


[ST]

```
g_int1 := BITARR_TO_INT(g_bool4[0], K8);
```

### 2) 带EN/ENO函数 (BITARR\_TO\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool2 := BITARR_TO_INT_E(g_bool1, g_bool4[0], K8, g_int1);
```

## 5.50 BITARR\_TO\_DINT(E) / 位阵列 → 双字[带符号]型、双字[无符号] / 位列[32位]型转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将使用位阵列指定的位数转换为双字[带符号]型、双字[无符号]/位列[32位]型数据后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
BITARR_TO_DINT		BITARR_TO_DINT(BitArr, n); 例: 标签2:= BITARR_TO_DINT(标签1[*2], K4);
BITARR_TO_DINT_E		BITARR_TO_DINT_E(EN, BitArr, n, 输出标签); 例: BITARR_TO_DINT_E(X000, 标签1[*2], K4, 标签2);

\*1. 变为输出变量。

\*2. 指定阵列要素。

#### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	BitArr( <u>s</u> )	转换原位排列要素的起始位	位
	<u>n</u>	指定的位数	字[带符号]
输出变量	ENO	执行状态	位
	*1 ( <u>d</u> )	转换后的双字[带符号]型、双字[无符号]/位列[32位]数据	ANY32

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的位排列要素作为起始，将(n)指定的位数转换为双字[带符号]型、双字[无符号]/位列[32位]型数据，输出到(d)指定的软元件中。

(n)中，仅可以指定常数4、8、12、16、20、24、28、32。

在指定的位数以上的输出位中设定为0。

### 注意要点

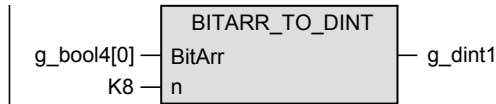
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据、阵列数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据、阵列数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

在 (s) 中指定的软元件的位阵列中，将从0号开始的8位转换为双字[带符号]型数据，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (BITARR\_TO\_DINT)

[结构化梯形图/FBD]

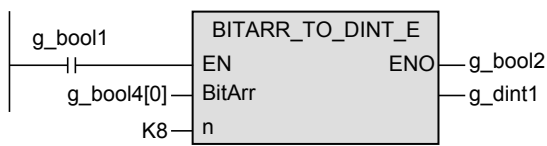


[ST]

```
g_dint1 := BITARR_TO_DINT(g_bool4[0], K8);
```

### 2) 带EN/ENO函数 (BITARR\_TO\_DINT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool2 := BITARR_TO_DINT_E(g_bool1, g_bool4[0], K8, g_dint1);
```

## 5.51 INT\_TO\_BITARR(\_E) / 字[带符号]型、字[无符号] / 位列[16位]型 → 位阵列转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将字[带符号]型、字[无符号]/位列[16位]型数据的低n位输出到位阵列。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INT_TO_BITARR		INT_TO_BITARR(s, n); <b>例:</b> 标签2[*1]:=INT_TO_BITARR(D10, K4);
INT_TO_BITARR_E		INT_TO_BITARR_E(EN, s, n, BitArr); <b>例:</b> INT_TO_BITARR_E(X000, D10, K4, 标签2[*1]);

\*1. 指定阵列要素。

### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	(s)	转换原字[带符号]型、字[无符号]/位列[16位]数据	ANY16
	(n)	指定的位数	字[带符号]
输出变量	ENO	执行状态	位
	BitArr((d))	转换后的位排列要素的起始位	位

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的字[带符号]型、字[无符号]/位列[16位]型数据的低n位，输出到(d)指定的软元件中。  
(n)中，仅可以指定常数4、8、12、16。  
在指定的位数以上的输出位不会更改。

### 注意要点

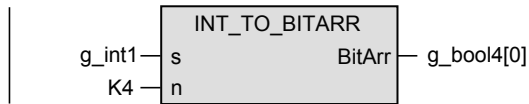
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据、阵列数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据、阵列数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

将 (s) 指定的软元件的字[带符号]型数据的低4位输出到 (d) 指定的软元件中的程序。

### 1) 无EN/ENO函数(INT\_TO\_BITARR)

[结构化梯形图/FBD]

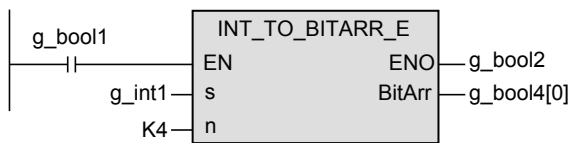


[ST]

```
g_bool4[0] := INT_TO_BITARR(g_int1, K4);
```

### 2) 带EN/ENO函数(INT\_TO\_BITARR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool2 := INT_TO_BITARR_E(g_bool1, g_int1, K4, g_bool4[0]);
```

## 5.52 DINT\_TO\_BITARR(E) / 双字[带符号]型、双字[无符号] / 位列[32位]型 → 位阵列转换

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将双字[带符号]型、双字[无符号]/位列[32位]型数据的低n位输出到位阵列中。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DINT_TO_BITARR		DINT_TO_BITARR(s, n); 例: 标签2[*1]:=DINT_TO_BITARR(标签1, K4);
DINT_TO_BITARR_E		DINT_TO_BITARR_E(EN, s, n, BitArr); 例: DINT_TO_BITARR_E(X000, 标签1, K4, 标签2[*1]);

\*1. 指定阵列要素。

### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	(s)	转换原双字[带符号]型、双字[无符号]/位列[32位]数据	ANY32
	(n)	指定的位数	字[带符号]
输出变量	ENO	执行状态	位
	BitArr((d))	转换后的位排列要素的起始位	位

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的双字[带符号]型、双字[无符号]/位列[32位]型数据的低n位，输出到(d)指定的软元件中。

(n)中，仅可以指定常数4、8、12、16、20、24、28、32。

在指定的位数以上的输出位不会更改。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据、阵列数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据、阵列数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。



## 程序举例

将 (s) 指定的软元件的双字[带符号]型数据的低4位输出到 (d) 指定的软元件中的程序。

### 1) 无EN/ENO函数(DINT\_TO\_BITARR)

[结构化梯形图/FBD]



[ST]

```
g_bool4[0] := DINT_TO_BITARR(g_dint1, K4);
```

### 2) 带EN/ENO函数(INT\_TO\_BITARR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool2 := DINT_TO_BITARR_E(g_bool1, g_dint1, K4, g_bool4[0]);
```

## 5.53 CPY\_BITARR(E) / 位阵列的复制

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

按照指定位大小复制位阵列。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
CPY_BITARR		CPY_BITARR(BitArrIn, n); 例: 标签2[*1] := CPY_BITARR(标签1[*1], K4);
CPY_BITARR_E		CPY_BITARR_E(EN, BitArrIn, K4, BitArrOut); 例: CPY_BITARR_E(X000, 标签1[*1], K4, 标签2[*1]);

\*1. 指定阵列要素。

### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	BitArrIn( <u>s</u> )	复制源位排列要素的起始位
	<u>n</u>	指定的位数
输出变量	ENO	执行状态
	BitArrOut( <u>d</u> )	复制目标位排列要素的起始位

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将 s 指定的软元件的位阵列的n位输出到 d 指定的软元件中。  
n 中，仅可以指定常数4、8、12、16、20、24、28、32。

### 注意要点

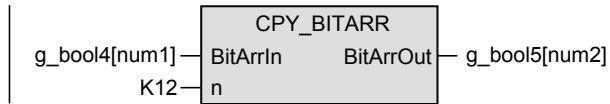
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据、阵列数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据、阵列数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

从 (s) 中指定的位阵列的第num1号要素， 将12位输出到 (d) 中指定的软元件的第num2号中的程序。

### 1) 无EN/ENO函数 (CPY\_BITARR)

[结构化梯形图/FBD]



[ST]

```
g_bool5[num2] := CPY_BITARR(g_bool4[num1], K12);
```

### 2) 带EN/ENO函数 (CPY\_BITARR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool2 := CPY_BITARR_E(g_bool1, g_bool4[num1], K12, g_bool5[num2]);
```

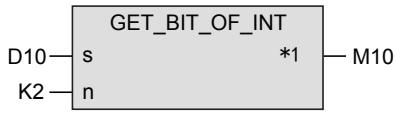
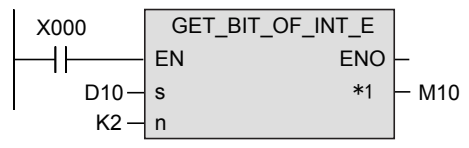
## 5.54 GET\_BIT\_OF\_INT(\_E) / 字[带符号]型数据的指定位读出

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

读出字[带符号]型数据的指定位的值。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
GET_BIT_OF_INT		GET_BIT_OF_INT (s, n); <b>例:</b> M10 := GET_BIT_OF_INT (D10, K2);
GET_BIT_OF_INT_E		GET_BIT_OF_INT_E (EN, s, n, *1); <b>例:</b> GET_BIT_OF_INT_E (X000, D10, K2, M10);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	(s)	读出源字[带符号]数据	字[带符号]
	(n)	指定位位置	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	读出目标位	位

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

读出(s)指定的软元件的第n位的值，输出到(d)指定的软元件中。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。

## 程序举例

读出 (s) 指定的软元件的第5位，输出到 (d) 指定的软元件中。

### 1) 无EN/ENO函数 (GET\_BIT\_OF\_INT)

[结构化梯形图/FBD]

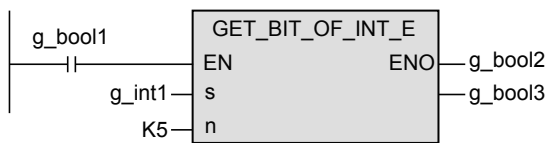


[ST]

```
g_bool3 := GET_BIT_OF_INT(g_int1, K5);
```

### 2) 带EN/ENO函数 (GET\_BIT\_OF\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool2 := GET_BIT_OF_INT_E (g_bool1, g_int1, K5, g_bool3);
```

## 5.55 SET\_BIT\_OF\_INT(E) / 字[带符号]型数据的指定位写入

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	×	×	×	×	×	×	×	×

### 概要

将值写入到字[带符号]型数据的指定位中。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
SET_BIT_OF_INT		SET_BIT_OF_INT(s, n); 例: D10:= SET_BIT_OF_INT(M0, K4);
SET_BIT_OF_INT_E		SET_BIT_OF_INT_E(EN, s, n, *1); 例: SET_BIT_OF_INT_E(X000, M0, K4, D10);

\*1. 变为输出变量。

### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	(s)	写入源位数据
	(n)	指定位位置
输出变量	ENO	执行状态
	*1 (d)	写入目标字[带符号]数据

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

读出(s)指定的软元件的值，写入到(d)指定的软元件的第n位中。

### 注意要点

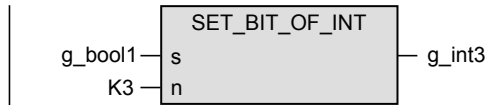
1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。

## 程序举例

④ 将③指定的软元件的值写入到指定的软元件的第3位中的程序。

### 1) 无EN/ENO函数 (SET\_BIT\_OF\_INT)

[结构化梯形图/FBD]

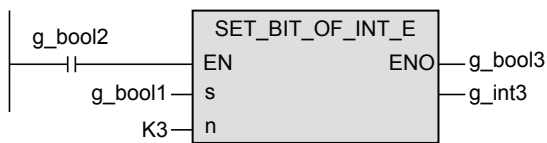


[ST]

```
g_int3 := SET_BIT_OF_INT(g_bool1, K3);
```

### 2) 带EN/ENO函数 (SET\_BIT\_OF\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := SET_BIT_OF_INT_E(g_bool2, g_bool1, K3, g_int3);
```

## 5.56 CPY\_BIT\_OF\_INT(E) / 字[带符号]型数据的指定位复制

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	×	×	×	×	×	×	×	×

### 概要

将字[带符号]型数据的指定位复制到其他字[带符号]型数据的指定位中。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
CPY_BIT_OF_INT		CPY_BIT_OF_INT(s, n1, n2); 例: D10:= CPY_BIT_OF_INT(D0, K1, K4);
CPY_BIT_OF_INT_E		CPY_BIT_OF_INT_E(EN, s, n1, n2, *1); 例: CPY_BIT_OF_INT_E(X000, D0, K1, K4, D10);

\*1. 变为输出变量。

### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	(s)	复制源字[带符号]型数据	字[带符号]
	(n1)	指定位位置(复制源)	字[带符号]
	(n2)	指定位位置(复制目标)	字[带符号]
输出变量	ENO	执行状态	位
	*1 ((d))	复制目标字[带符号]数据	字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的第n1位的值复制到(d)指定的软元件的第n2位中。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。

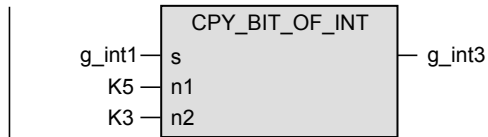


## 程序举例

将 (s) 指定的软元件的第5位值写入到 (d) 指定的软元件的第3位中。

### 1) 无EN/ENO函数(CPY\_BIT\_OF\_INT)

[结构化梯形图/FBD]

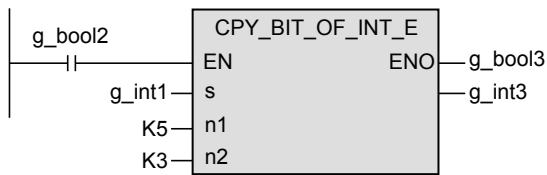


[ST]

```
g_int3 := CPY_BIT_OF_INT(g_int1, K5, K3);
```

### 2) 带EN/ENO函数(CPY\_BIT\_OF\_INT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := CPY_BIT_OF_INT_E(g_bool2, g_int1, K5, K3, g_int3);
```

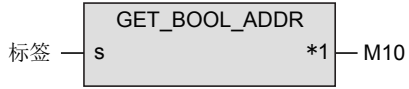
## 5.57 GET\_BOOL\_ADDR / 起始数据的获取

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将起始数据作为位型数据进行输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
GET_BOOL_ADDR		GET_BOOL_ADDR(s); 例: M10:= GET_BOOL_ADDR(标签);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s)	输入数据	ANY
输出变量 *1 (d)	输出数据	位

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

从(s)指定的数据中，将起始数据作为位型数据输出到(d)中。

输入数据类型	输出数据类型
位、位的阵列	位

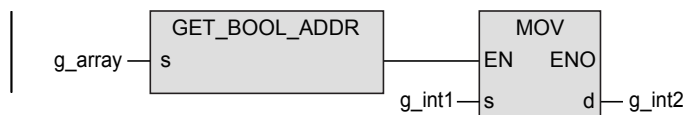
### 注意要点

- 1) 使用编程工具在(s)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

### 程序举例

通过(s)指定的位阵列数据的起始位，执行MOV指令的程序。

[结构化梯形图/FBD]



[ST]

```
MOV(GET_BOOL_ADDR(g_array), g_int1, g_int2);
```

## 5.58 GET\_INT\_ADDR / 起始数据的获取

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将起始数据作为字[带符号]型数据进行输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
GET_INT_ADDR		GET_INT_ADDR (s); 例: D10:= GET_INT_ADDR(标签);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s)	输入数据	ANY
输出变量 *1 (d)	输出数据	字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

从(s)指定的数据中，将起始数据作为字[带符号]数据，输出到(d)中。

输入数据类型	输出数据类型
字[带符号]、双字[带符号]、字[无符号]/位列[16位]、单精度实数、字符串、时间型、字[带符号]的阵列、双字[带符号]的阵列、字[无符号]/位列[16位]的阵列、双字[无符号]/位列[32位]的阵列、实数的阵列、时间型的阵列	字[带符号]

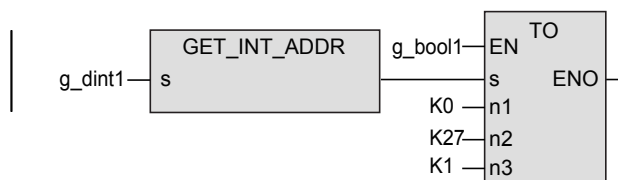
### 注意要点

- 在结构化程序中处理32位数据、阵列数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据、阵列数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
使用标签时，使用全局标签。
- 使用编程工具在(s)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

### 程序举例

将(s)指定的双字[带符号]数据低16位数据作为字[带符号]数据，写入缓冲存储器的程序。

[结构化梯形图/FBD]



[ST]

TO(g\_bool1, GET\_INT\_ADDR(g\_dint1), K0, K27, K1);

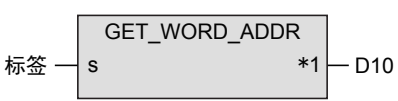
## 5.59 GET\_WORD\_ADDR / 起始数据的获取

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将起始数据作为字[无符号]/位列[16位]型数据进行输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
GET_WORD_ADDR		GET_WORD_ADDR(s); 例: D10:= GET_WORD_ADDR(标签);

\*1. 变为输出变量。

### 2. 设定数据

变量	内容	数据类型
输入变量 (s)	输入数据	ANY
输出变量 *1 (d)	输出数据	字[无符号]/位列[16位]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

从(s)指定的数据中，将起始数据作为字[无符号]/位列[16位]数据，输出到(d)中。

输入数据类型	输出数据类型
字[带符号]、双字[带符号]、字[无符号]/位列[16位]、单精度实数、字符串、时间型、字[带符号]的阵列、双字[带符号]的阵列、字[无符号]/位列[16位]的阵列、双字[无符号]/位列[32位]的阵列、实数的阵列、时间型的阵列	字[无符号]/位列[16位]

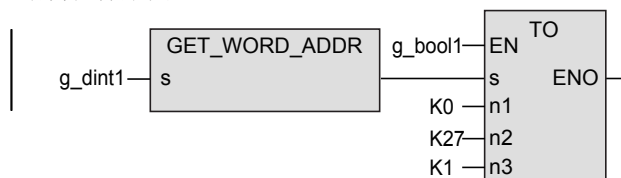
### 注意要点

- 在结构化程序中处理32位数据、阵列数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据、阵列数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。使用标签时，使用全局标签。
- 使用编程工具在(s)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

### 程序举例

将(s)指定的双字[带符号]数据的低16位数据作为字[无符号]/位列[16位]数据，写入缓冲存储器的程序。

[结构化梯形图/FBD]



[ST]

TO(g\_bool1, GET\_WORD\_ADDR(g\_dint1), K0, K27, K1);

## 6. 应用函数(单数值变量函数)

函数名	功能	参考
ABS(_E)	绝对值	6.1节

1  
概要

2  
函数、操作符  
一览

3  
函数的构成

4  
函数的阅读方法

5  
应用函数  
(型转换函数)

6  
应用函数  
(单数值变量  
函数)

7  
应用函数  
(算术运算函数)

8  
应用函数  
(位移位函数)

9  
应用函数  
(位型布尔函数)

10  
应用函数  
(选择函数)

## 6.1 ABS(E) / 绝对值

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

求出绝对值后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
ABS		ABS(_IN); 例: D10:=ABS(D0);
ABS_E		ABS_E(EN, _IN, 输出标签); 例: ABS_E(X000, D0, D10);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN (s)	求绝对值的数据或保存数据的字软元件	ANY_NUM
输出变量	ENO	执行状态	位
	*1 (d)	保存运算结果的字软元件	ANY_NUM

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

- 1) 求出(s)中指定的软元件的字[带符号]型/双字[带符号]型/单精度实数型数据的绝对值，按照与(d)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中。  
如果将输入值视为A，运算输出值视为B，则关系如下所示。  
$$B=|A|$$
- 2) (s)指定的软元件的数据类型为字[带符号]型，且为-32768时，则-32768被输出到(d)指定的软元件中。(绝对值的最大值只能获得32,767以内的值。)  
如果(s)指定的软元件的数据类型为双字[带符号]型，且输入了-2147483648时，则-2147483648被输出到(d)指定的软元件中。(绝对值的最大值只能获得2147483647以内的值。)

### 注意要点

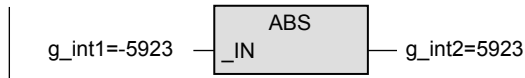
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 使用编程工具在(s)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

## 程序举例

将 (s) 中指定的软元件的字[带符号]型数据的绝对值, 按照与 (s) 中指定的软元件相同的数据类型, 输出到 (d) 中指定软元件中的程序。

### 1) 无EN/ENO函数 (ABS)

[结构化梯形图/FBD]

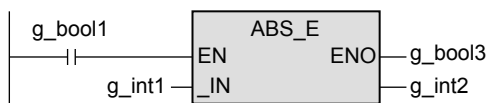


[ST]

```
g_int2 := ABS(g_int1);
```

### 2) 带EN/ENO函数 (ABS\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := ABS_E(g_bool1, g_int1, g_int2);
```

## 7. 应用函数(算术运算函数)

函数名	功能	参考
ADD_E	加法运算	7.1节
SUB_E	减法运算	7.2节
MUL_E	乘法运算	7.3节
DIV_E	除法运算	7.4节
MOD(_E)	余数	7.5节
EXPT(_E)	幂乘	7.6节
MOVE(_E)	代入	7.7节



## 7.1 ADD\_E / 加法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将数据进行加法运算 (A+B=C) 后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
ADD_E		ADD_E (EN, _IN, _IN, 输出标签); 例: ADD_E (X000, D0, D10, D20);

\*1. 变为输出变量。

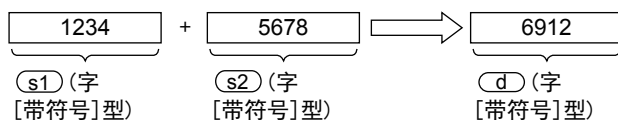
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN ((s1) ~ (s28))	加法运算数据, 或是保存数据的字软元件	ANY_NUM
输出变量	ENO	执行状态	位
	*1 ((d))	保存运算结果的字软元件	ANY_NUM

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(s1) ~ (s28) 中指定的软元件的字[带符号]型/双字[带符号]型/单精度实数型数据的数据进行加法运算 ((s1)+(s2).....+(s28)), 将运算结果按照与(s1) ~ (s28) 中指定的软元件相同的数据类型, 输出到(d) 中指定的软元件中。  
(例)数据类型为字[带符号]型时



- (s) 的PIN数可在2~28范围内进行更改。

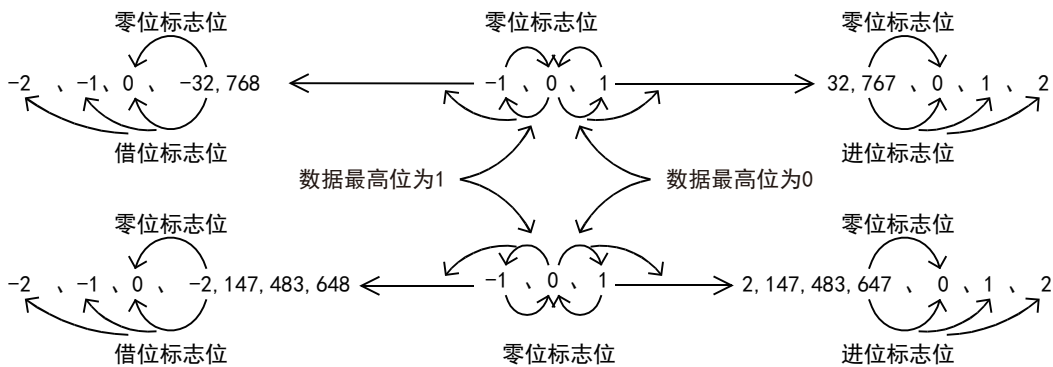
→ 3章 函数的构成

### 注意要点

- 1) 在结构化程序中处理 32 位数据时，不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时，请使用标签。  
但是，32 位计数器为 32 位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 2) 即使运算结果中产生了下溢出/上溢出时，也不出现运算错误，而是由 ENO 输出 TRUE。  
此时，无法获得正确的运算结果，敬请注意。

此外根据运算结果，下表中的标志位进行动作。

软元件	名称	内容
M8020	零位	ON : 运算结果为 0 时 OFF: 运算结果为 0 以外时
M8021	借位	ON : 运算结果不到 -32,768 (16 位运算) 或是 -2,147,483,648 (32 位运算) 时，借位标志位动作。 OFF: 运算结果超出 -32,768 (16 位运算) 或是 -2,147,483,648 (32 位运算) 时
M8022	进位	ON : 运算结果超过 32,767 (16 位运算) 或是 2,147,483,647 (32 位运算) 时，进位标志位动作。 OFF: 运算结果低于 32,767 (16 位运算) 或是 2,147,483,647 (32 位运算) 时

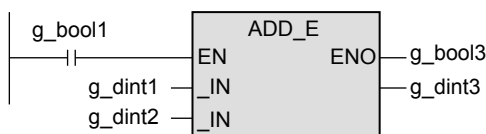


- 3) 使用编程工具在 (s) 中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考 Q/L/F 结构化编程手册(基础篇)。

### 程序举例

对 (s1)、(s2) 中指定的软元件的双字[带符号]型数据进行加法运算，将运算结果输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=ADD_E(g_bool1,g_dint1,g_dint2,g_dint3);
```

## 7.2 SUB\_E / 减法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对2个值进行减法运算(A-B=C)后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
SUB_E		SUB_E (EN, _IN1, _IN2, 输出标签); 例: SUB_E (X000, D0, D10, D20);

\*1. 变为输出变量。

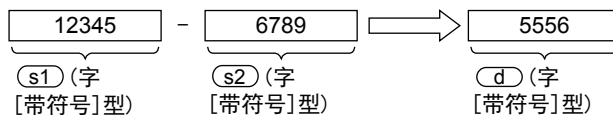
#### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN1 (s1)	被减法运算数据, 或是保存数据的字软元件	ANY_NUM
	_IN2 (s2)	减法运算数据, 或是保存数据的字软元件	ANY_NUM
输出变量	ENO	执行状态	位
	*1 (d)	保存运算结果的字软元件	ANY_NUM

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(s1)、(s2)中指定的软元件的字[带符号]型/双字[带符号]型/单精度实数型数据进行减法运算((s1)-(s2)), 将运算结果按照与(s1)、(s2)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。  
(例)数据类型为字[带符号]型时



1 概要

2 函数、操作符  
一览

3 函数的构成

4 函数的阅读方法

5 应用函数  
(型转换函数)

6 应用函数  
(单数值变量  
函数)

7 应用函数  
(算术运算函数)

8 应用函数  
(位移位函数)

9 应用函数  
(位逻辑函数)

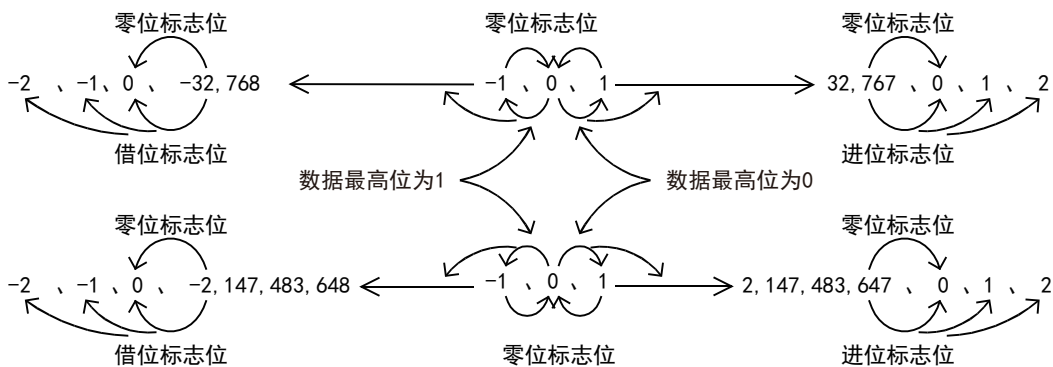
10 应用函数  
(选择函数)

### 注意要点

- 1) 在结构化程序中处理 32 位数据时，不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时，请使用标签。  
但是，32 位计数器为 32 位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 2) 即使运算结果中产生了下溢出/上溢出时，也不出现运算错误，而是由 ENO 输出 TRUE。  
此时，无法获得正确的运算结果，敬请注意。

此外根据运算结果，下表中的标志位进行动作。

软元件	名称	内容
M8020	零位	ON : 运算结果为 0 时 OFF: 运算结果为 0 以外时
M8021	借位	ON : 运算结果不到 -32,768 (16 位运算) 或是 -2,147,483,648 (32 位运算) 时，借位标志位动作。 OFF: 运算结果超出 -32,768 (16 位运算) 或是 -2,147,483,648 (32 位运算) 时
M8022	进位	ON : 运算结果超过 32,767 (16 位运算) 或是 2,147,483,647 (32 位运算) 时，进位标志位动作。 OFF: 运算结果低于 32,767 (16 位运算) 或是 2,147,483,647 (32 位运算) 时

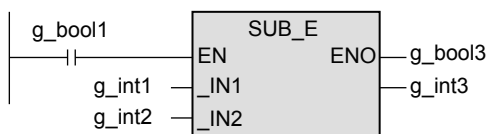


- 3) 使用编程工具在 (S) 中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考 Q/L/F 结构化编程手册 (基础篇)。

### 程序举例

对 (s1)、(s2) 中指定的软元件的字 [带符号] 型数据进行减法运算，将运算结果输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=SUB_E(g_bool1,g_int1,g_int2,g_int3);
```

## 7.3 MUL\_E / 乘法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将数据进行乘法运算 (A×B=C) 后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MUL_E		MUL_E(EN, _IN, _IN, 输出标签); 例: MUL_E(X000, D0, D10, D20);

\*1. 变为输出变量。

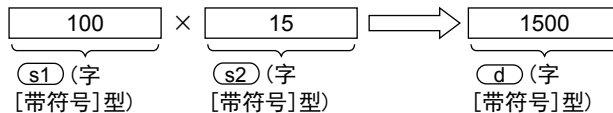
#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_IN (s1) ~ (s28)	乘数数据, 或是保存数据的字软元件
输出变量	ENO	执行状态
	*1 (d)	保存运算结果的字软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(s1)~(s28)中指定的软元件的字[带符号]型/双字[带符号]型/单精度实数型数据进行乘法运算((s1)×(s2)×...×(s28)), 将运算结果按照与(s1)~(s28)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。  
(例)数据类型为字[带符号]型时



- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

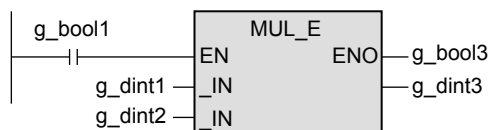
### 注意要点

- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 即使运算结果中产生了下溢出/上溢出时, 也不出现运算错误, 而是由ENO输出TRUE。此时, 无法获得正确的运算结果, 敬请注意。
- 使用编程工具在(s)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

### 程序举例

对(s1)、(s2)中指定的软元件的双字[带符号]型数据进行乘法运算, 将运算结果输出到(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_bool3:=MUL\_E(g\_bool1,g\_dint1,g\_dint2,g\_dint3);

## 7.4 DIV\_E / 除法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对2个值进行除法运算(A ÷ B = C · · 余数)后，输出商。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DIV_E		DIV_E(EN, _IN1, _IN2, 输出标签); 例: DIV_E(X000, D0, D10, D20);

\*1. 变为输出变量。

#### 2. 设定数据

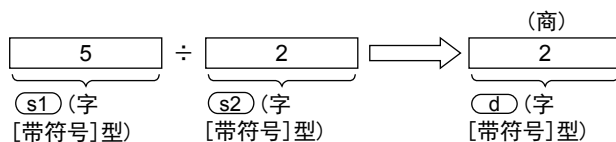
	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN1 (s1)	被除数数据，或是保存数据的字软元件	ANY_NUM
	_IN2 (s2)	除数数据，或是保存数据的字软元件	ANY_NUM
输出变量	ENO	执行状态	位
	*1 (d)	保存运算结果的字软元件	ANY_NUM

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

对(s1)、(s2)中指定的软元件的字[带符号]型/双字[带符号]型/单精度实数型数据进行除法运算((s1) ÷ (s2))，将运算结果按照与(s1)、(s2)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中。

(例)数据类型为字[带符号]型时



### 注意要点

- 1) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。
- 2) 使用编程工具在(s)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

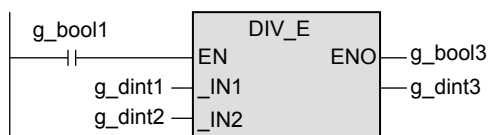
### 错误

- 1) (s2)设定的软元件的除数为0时，会发生运算错误，并且不能执行函数。
- 2) 运算结果超过32,767(16位运算)或者2,147,483,647(32位运算)时，出现运算错误。

## 程序举例

对 (s1)、(s2) 中指定的软元件的双字[带符号]型数据进行除法运算，将运算结果的商按照与 (s1)、(s2) 中指定的软元件相同的数据类型，输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=DIV_E(g_bool1,g_dint1,g_dint2,g_dint3);
```

## 7.5 MOD(\_E) / 余数

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

对2个值进行除法运算( $A \div B = C \cdot \cdot$  余数)后, 输出余数。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MOD		<code>_IN1 MOD _IN2; *2</code> 例: <code>标签3:=标签1 MOD 标签2;</code>
MOD_E		<code>MOD_E(EN, _IN1, _IN2, 输出标签; *2</code> 例: <code>MOD_E(X000, 标签1, 标签2, 标签3);</code>

- \*1. 变为输出变量。
- \*2. 请参考注意要点。

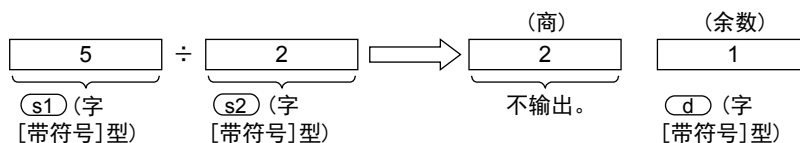
### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	<code>_IN1 (s1)</code>	被除数数据, 或是保存数据的字软元件
	<code>_IN2 (s2)</code>	除数数据, 或是保存数据的字软元件
输出变量	ENO	执行状态
	<code>*1 (d)</code>	保存运算结果的字软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

对(s1)、(s2)中指定的软元件的字[带符号]型/双字[带符号]型数据进行除法运算( $(s1) \div (s2)$ ), 将运算结果的余数按照与(s1)、(s2)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。  
(例)数据类型为字[带符号]型时



### 注意要点

- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理 32 位数据时, 不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时, 请使用标签。  
但是, 32位计数器为32位长度的软元件, 因此可以直接指定。  
指定标签时, 使用全局标签。
- 3) 使用ST语言的函数MOD的描述, 与其他函数的写法不同, 需要加以注意。



## 错误

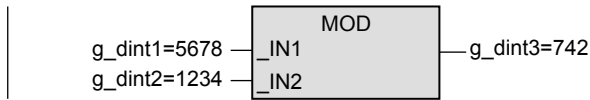
- 1) (s2) 指定的软元件的除数为0时，会发生运算错误，并且不能执行函数。
- 2) 运算结果超过32,767(16位运算)或者2,147,483,647(32位运算)时，出现运算错误。

## 程序举例

对(s1)、(s2)中指定的软元件的双字[带符号]型数据进行除法运算，将运算结果的余数按照与(s1)、(s2)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中的程序。

- 1) 无EN/ENO函数(MOD)

[结构化梯形图/FBD]

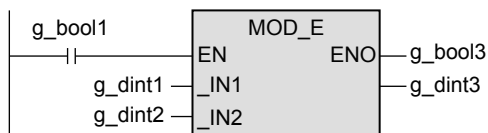


[ST]

```
g_dint3:=g_dint1 MOD g_dint2;
```

- 2) 带EN/ENO函数(MOD\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3 := MOD_E(g_bool1, g_dint1, g_dint2, g_dint3);
```

## 7.6 EXPT(\_E) / 幂乘

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

求出幂乘后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
EXPT		EXPT (In1, In2); <b>例:</b> 标签2:=EXPT(标签1, D10);
EXPT_E		EXPT_E (EN, In1, In2, 输出标签); <b>例:</b> EXPT_E (X000, 标签1, D10, 标签2);

\*1. 变为输出变量。

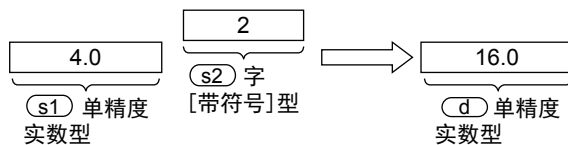
#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	In1 (s1)	底数数据, 或是保存数据的字软元件
	In2 (s2)	指数数据, 或是保存数据的字软元件
输出变量	ENO	执行状态
	*1 (d)	保存运算结果的字软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s1)中指定的软元件的单精度实数型数据进行幂乘运算(幂乘, 为(s2)中指定的软元件内容), 将运算结果输出到(d)中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。  
但是, 32位计数器为32位长度的软元件, 因此可以直接指定。  
指定标签时, 使用全局标签。
- 3) 使用编程工具在(s)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

## 错误

以下一些情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

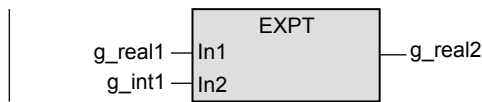
- 1) (s1)中指定的值为负时。  
(错误代码：K6706)
- 2) (s1)中指定的值为“0”时。  
(错误代码：K6706)
- 3) 运算结果不在下面的范围时。  
(错误代码：K6706)  
 $2^{-126} \leq | \text{运算结果} | < 2^{128}$

## 程序举例

根据(s1)、(s2)中指定的软元件内容进行幂乘运算，将运算结果按照与(s1)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中的程序。

- 1) 无EN/ENO函数(EXPT)

[结构化梯形图/FBD]

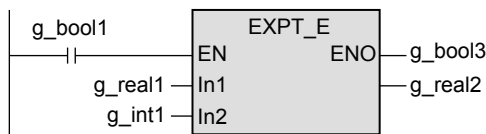


[ST]

```
g_real2:=EXPT(g_real1,g_int1);
```

- 2) 带EN/ENO函数(EXPT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=EXPT_E(g_bool1,g_real1,g_int1,g_real2);
```

## 7.7 MOVE(\_E) / 代入

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

将软元件的内容代入到其他的软元件中。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MOVE		MOVE(_IN); 例: D10:=MOVE(D0);
MOVE_E		MOVE_E(EN, _IN, 输出标签); 例: MOVE_E(X000, D0, D10);

\*1. 变为输出变量。

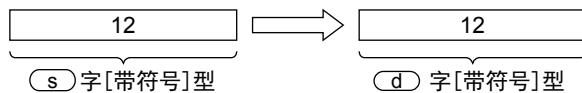
#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN (s)	传送源数据, 或是保存数据的字软元件	ANY
输出变量	ENO	执行状态	位
	*1 (d)	传送目标的软元件	ANY

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

将(s)指定的软元件的内容代入到(d)指定的软元件中。



### 注意要点

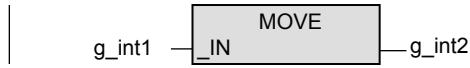
- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。  
但是, 32位计数器为32位长度的软元件, 因此可以直接指定。  
指定标签时, 使用全局标签。
- 3) 使用编程工具在(s)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

## 程序举例

将 (s) 指定的软元件的字[带符号]型数据代入到 (d) 指定的软元件中的程序。

### 1) 无EN/ENO函数(MOVE)

[结构化梯形图/FBD]

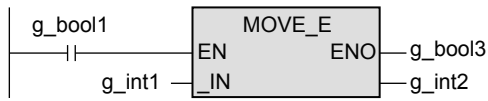


[ST]

```
g_int2:=MOVE(g_int1);
```

### 2) 带EN/ENO函数(MOVE\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=MOVE_E(g_bool1,g_int1,g_int2);
```

## 8. 应用函数(位移位函数)

函数名	功能	参考
SHL(_E)	左移	8.1节
SHR(_E)	右移	8.2节

## 8.1 SHL(\_E) / 左移

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

使指定位长度的软元件每次左移指定的位长度。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
SHL		SHL(_IN, _N); 例: D10:=SHL(D0, K1);
SHL_E		SHL_E(EN, _IN, _N, 输出标签); 例: SHL_E(X000, D0, K1, D10);

\*1. 变为输出变量。

#### 2. 设定数据

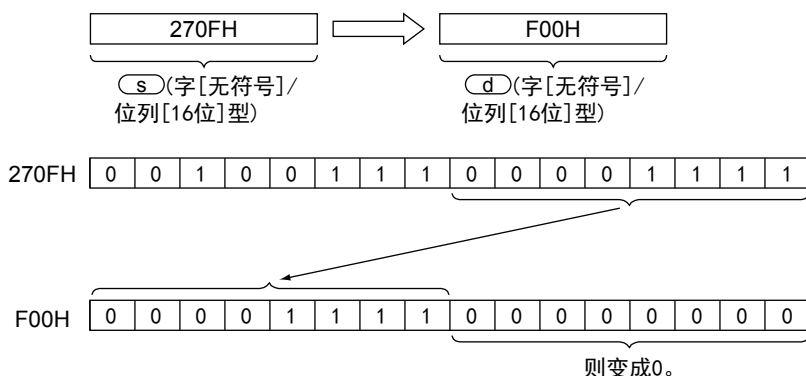
	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN (s)	保存左位移数据的字软元件	ANY_BIT
	_N (n)	移位的位数	ANY_BIT
输出变量	ENO	执行状态	位
	*1 (d)	保存位移后数据的字软元件	ANY_BIT

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 1) 将(s)中指定的软元件的字[无符号]/位列[16位]型/双字[无符号]型/位列[32位]型数据向左移n位，按照与(s)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中。  
向左移动的位数，为(n)指定的值。

(例) (s)指定的软元件的数据类型为字[无符号]/位列[16位]型，(n)指定的值为8时



- 2) 最低位开始的n位变为0。

### 注意要点

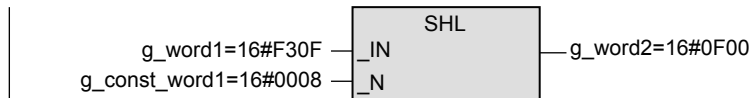
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

### 程序举例

将 **(s)** 中指定的软元件的字[无符号]/位列[16位]型数据向左移n位，按照与 **(s)** 中指定的软元件相同的数据类型，输出到 **(d)** 中指定的软元件中的程序。

- 1) 无EN/ENO函数(SHL)

[结构化梯形图/FBD]

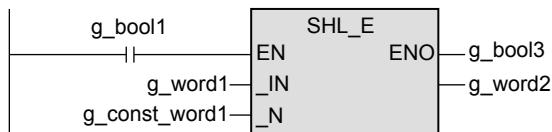


[ST]

```
g_word2:=SHL(g_word1,g_const_word1);
```

- 2) 带EN/ENO函数(SHL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=SHL_E(g_bool1,g_word1,g_const_word1,g_word2);
```



## 8.2 SHR(E) / 右移

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

使指定位长度的软元件每次右移指定的位长度。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
SHR		SHR(_IN, _K); 例: D10:=SHR(D0, K1);
SHR_E		SHR_E(EN, _IN, _N, 输出标签); 例: SHR_E(X000, D0, K1, D10);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	_IN (s)	保存右位移数据的字软元件
	_K, _N (n)	移位的位数
输出变量	ENO	执行状态
	*1 (d)	保存位移后数据的字软元件

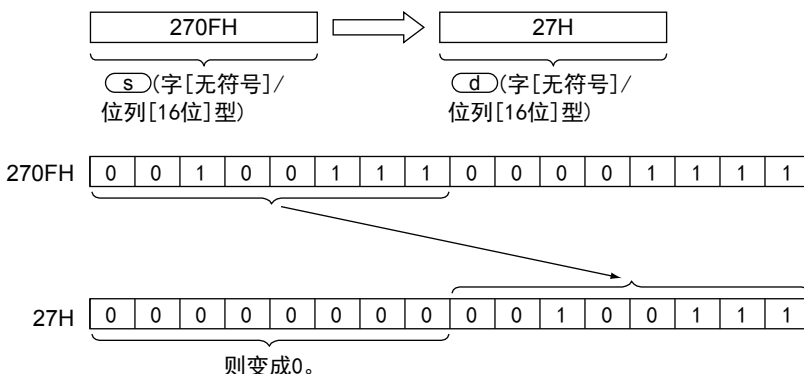
在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 1) 将(s)中指定的软元件的字[无符号]/位列[16位]型/双字[无符号]型/位列[32位]型数据向右移动n位，按照与(s)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中。

向右移动的位数，为(n)指定的值。

(例) (s)中指定的软元件的数据类型为字[无符号]/位列[16位]型，(n)指定的值为8时



- 2) 最高位开始的(n)位变为0。

### 注意要点

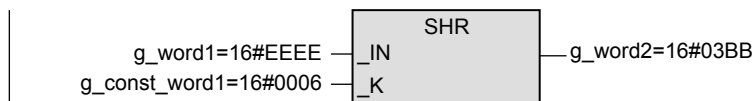
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理 32 位数据时，不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时，请使用标签。  
但是，32 位计数器为 32 位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

### 程序举例

将 (S) 中指定的软元件的字[无符号]/位列[16位]型数据向右移动n位，按照与 (S) 中指定的软元件相同的数据类型，输出到 (D) 中指定的软元件中的程序。

- 1) 无EN/ENO函数 (SHR)

[结构化梯形图/FBD]

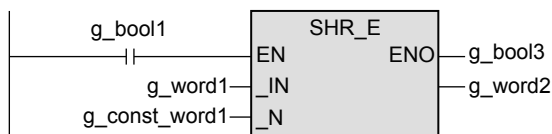


[ST]

```
g_word2:=SHR(g_word1,g_const_word1);
```

- 2) 带EN/ENO函数 (SHR\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=SHR_E(g_bool1,g_word1,g_const_word1,g_word2);
```

## 9. 应用函数 (位型布尔函数)

函数名	功能	参考
AND_E	逻辑与	9.1节
OR_E	逻辑或	9.2节
XOR_E	逻辑异或	9.3节
NOT(_E)	逻辑非	9.4节

1  
概要

2  
函数、操作符  
一览

3  
函数的构成

4  
函数的阅读方法

5  
应用函数  
(型转换函数)

6  
应用函数  
(单数值变量  
函数)

7  
应用函数  
(算术运算函数)

8  
应用函数  
(位移位函数)

9  
应用函数  
(位型布尔函数)

10  
应用函数  
(选择函数)

## 9.1 AND\_E / 逻辑与

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对多位的逻辑与进行运算后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
AND_E		AND_E (EN, _IN, _IN, 输出标签); 例: AND_E (X000, M0, M10, M20);

\*1. 变为输出变量。

#### 2. 设定数据

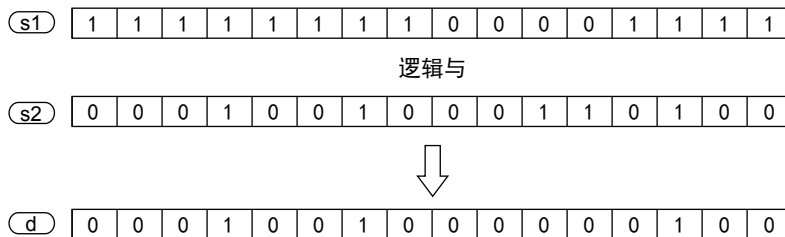
变量	内容	数据类型
输入变量	EN	执行条件
	_IN (s1) ~ (s28)	进行逻辑与运算的软元件
输出变量	ENO	执行状态
	*1 (d)	保存运算结果的软元件

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]型/位列[32位]型数据的每位进行逻辑与运算，将运算结果按照与(d)中指定的软元件相同的数据类型，输出到(s1) ~ (s28)中指定的软元件中。

(例)数据类型为字[无符号]/位列[16位]型时



- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

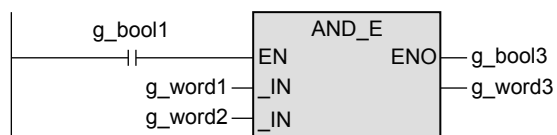
### 注意要点

在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。

## 程序举例

对 (s1)、(s2) 中指定的软元件的字 [无符号] / 位列 [16 位] 型数据的每位进行逻辑与运算，将运算结果按照与 (d) 中指定的软元件相同的数据类型 (s1)，输出到 (s2) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=AND_E(g_bool1,g_word1,g_word2,g_word3);
```

## 9.2 OR\_E / 逻辑或

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对多位的逻辑或进行运算后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
OR_E		OR_E(EN, _IN, _IN, 输出标签); 例: OR_E(X000, M0, M10, M20);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_IN (s1) ~ (s28)	进行逻辑或运算的软元件
输出变量	ENO	执行状态
	*1 (d)	保存运算结果的软元件

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]型/位列[32位]型数据的每位进行逻辑或运算，将运算结果按照与(d)中指定的软元件相同的数据类型，输出到(s1) ~ (s28)中指定的软元件中。

(例)数据类型为字[无符号]/位列[16位]型时

(s1) 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 1

逻辑或

(s2) 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0



(d) 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1

- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

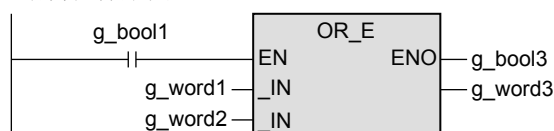
### 注意要点

- 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。

### 程序举例

对(s1)、(s2)中指定的软元件的字[无符号]/位列[16位]型数据的每位进行逻辑或运算，将运算结果按照与(d)中指定的软元件相同的数据类型(s1)，输出到(s2)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=OR_E(g_bool1,g_word1,g_word2,g_word3);
```

### 9.3 XOR\_E / 逻辑异或

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

#### 概要

对多位的逻辑异或进行运算后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
XOR_E		XOR_E (EN, _IN, _IN, 输出标签); 例: XOR_E (X000, M0, M10, M20);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN ((s1) ~ (s28))	进行逻辑异或运算的软元件	ANY_BIT
输出变量	ENO	执行状态	位
	*1 ((d))	保存运算结果的软元件	ANY_BIT

在函数说明中，输入输出变量通过( )中的描述进行记载。

#### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]/位列[32位]型数据的每位进行逻辑异或运算，将运算结果按照与(d)中指定的软元件相同的数据类型，输出到(s1) ~ (s28)中指定的软元件中。

(例)数据类型为字[无符号]/位列[16位]型时

(s1) 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

逻辑异或

(s2) 0 0 0 1 1 0 1 1 1 1 1 1 0 0 0 0



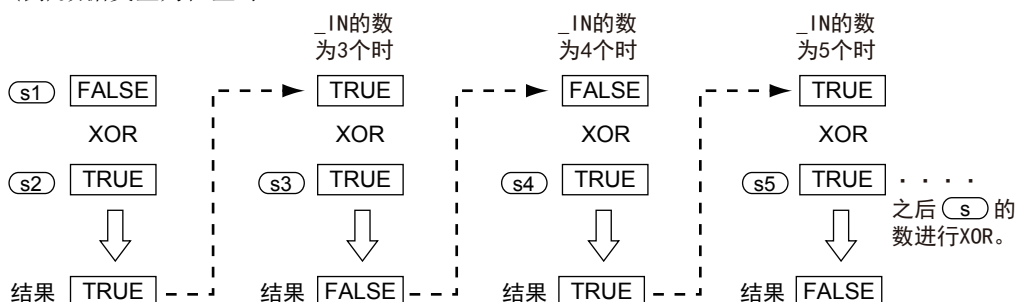
(d) 1 0 1 1 0 0 0 1 0 1 0 1 1 0 1 0

- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

- (s)有3个以上时，对于对(s1)、(s2)进行XOR后的结果，对(s3)进行XOR。此外，有(s4)时，对于对(s3)进行XOR后的结果，对(s4)进行XOR。之后，仅(s5)、(s6)和输入标签(s)数进行XOR。

(例)数据类型为位型时



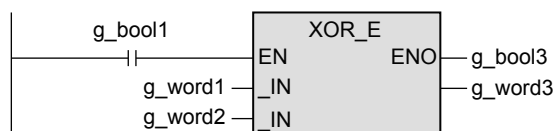
### 注意要点

- 1) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

### 程序举例

对①、②中指定的软元件的字[无符号]/位列[16位]型数据的每位进行逻辑异或运算，将运算结果按照与③中指定的软元件相同的数据类型④，输出到⑤中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=XOR_E(g_bool1,g_word1,g_word2,g_word3);
```



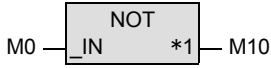
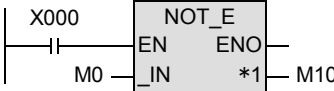
## 9.4 NOT(\_E) / 逻辑非

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对位进行逻辑非运算后输出。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
NOT		NOT (_IN) ; 例: M10:=NOT (M0) ;
NOT_E		NOT_E (EN, _IN, 输出标签) ; 例: NOT_E (X000, M0, M10) ;

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN ( <u>s</u> )	进行逻辑非运算的软元件	ANY_BIT
输出变量	ENO	执行状态	位
	*1 ( <u>d</u> )	保存运算结果的软元件	ANY_BIT

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

对 s 中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]/位列[32位]型数据的每位进行逻辑非运算，将运算结果按照与 d 中指定的软元件相同的数据类型 s1，输出到 s2 中指定的软元件中。  
(例)数据类型为字[无符号]/位列[16位]型时

s    0 1 1 0 1 0 1 1 0 0 0 0 1 1 1 1

逻辑非

d    1 0 0 1 0 1 0 0 1 1 1 1 0 0 0 0

### 注意要点

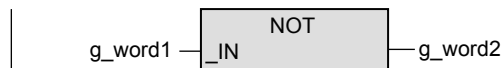
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。

## 程序举例

对 (s) 中指定的软元件的字 [无符号] / 位列 [16 位] 型数据的每位进行逻辑非运算，将运算结果按照与 (s) 中指定的软元件相同的数据类型，输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (NOT)

[结构化梯形图/FBD]

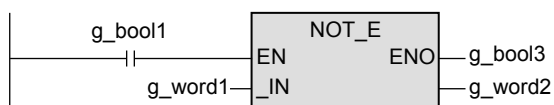


[ST]

```
g_word2:= NOT(g_word1);
```

### 2) 带EN/ENO函数 (NOT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=NOT_E(g_bool1,g_word1,g_word2);
```

## 10. 应用函数 (选择函数)

函数名	功能	参考
SEL(_E)	选择值	10.1节
MAXIMUM(_E)	选择最大值	10.2节
MINIMUM(_E)	选择最小值	10.3节
LIMITATION(_E)	上下限限位控制	10.4节
MUX(_E)	多路选择器	10.5节

1  
概要

2  
函数、操作符  
一览

3  
函数的构成

4  
函数的阅读方法

5  
应用函数  
(型转换函数)

6  
应用函数  
(单数值变量  
函数)

7  
应用函数  
(算术运算函数)

8  
应用函数  
(位移位函数)

9  
应用函数  
(位型布尔函数)

10  
应用函数  
(选择函数)

## 10.1 SEL(\_E) / 选择值

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	○	○	○	○	○	○	○	○

### 概要

根据输入条件选择并输出2个数据。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
SEL		SEL(_G, _IN0, _IN1); 例: D20:=SEL(M0, D0, D10);
SEL_E		SEL_E(EN, _G, _IN0, _IN1, 输出标签); 例: SEL_E(X000, M0, D0, D10, D20);

\*1. 变为输出变量。

### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_G (s1)	作为选择条件的位数据	位
	_IN0 (s2)	选择的数据,或是保存数据的字软元件	ANY
	_IN1 (s3)	选择的数据,或是保存数据的字软元件	ANY
输出变量	ENO	执行状态	位
	*1 (d)	保存选择结果的字软元件	ANY

在函数说明中, 输入输出变量通过( )中的描述进行记载。

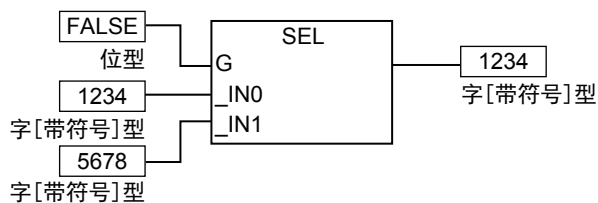
### 功能和动作说明

根据(s1)指定的软元件的值, 将(s2)、(s3)中指定的软元件的任意值, 按照与(s2)、(s3)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。

(s1)中指定的软元件的值为FALSE时, 则将(s2)中指定的软元件的值输出到(d)中指定的软元件中。

(s1)中指定的软元件的值为TRUE时, 则将(s3)中指定的软元件的值输出到(d)中指定的软元件中。

(例)输入变量(s2)、(s3)的数据类型为字[带符号]型时



## 注意要点

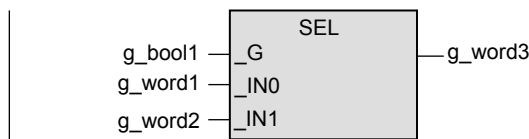
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 3) 使用编程工具在(S)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

## 程序举例

根据(S1)指定的软元件的值，将(S2)、(S3)中指定的软元件的任意值，按照与(S2)、(S3)中指定的软元件相同的数据类型，输出到(D)中指定的软元件中的程序。

- 1) 无EN/ENO函数 (SEL)

[结构化梯形图/FBD]

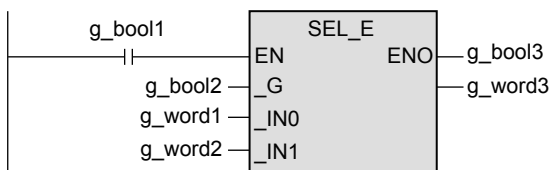


[ST]

```
g_word3:=SEL(g_bool1,g_word1,g_word2);
```

- 2) 带EN/ENO函数 (SEL\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=SEL_E(g_bool1,g_bool2,g_word1,g_word2,g_word3);
```

## 10.2 MAXIMUM(\_E) / 选择最大值

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

从数据中选择最大值后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MAXIMUM		MAXIMUM(_IN, _IN); 例: D20:=MAXIMUM(D0, D10);
MAXIMUM_E		MAXIMUM_E(EN, _IN, _IN, 输出标签); 例: MAXIMUM_E(X000, D0, D10, D20);

\*1. 变为输出变量。

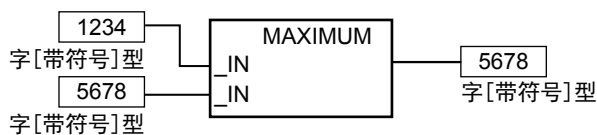
### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN (s1) ~ (s28)	进行比较的数据，或是保存数据的字软元件	ANY_SIMPLE
输出变量	ENO	执行状态	位
	*1 (d)	保存最大值的字软元件	ANY_SIMPLE

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 1) 将(s1) ~ (s28)中指定的软元件的 ANY\_SIMPLE 型数据的最大值按照与(s1) ~ (s28)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中。  
(例)数据类型为字[带符号]型时



- 2) (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

### 注意要点

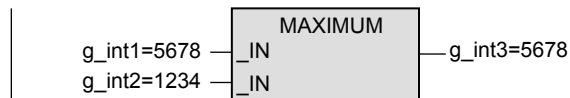
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 3) 使用编程工具在(s)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

## 程序举例

将 (s1) 中指定的软元件的字[带符号]型数据的最大值，按照与 (s) 中指定的软元件相同的数据类型，输出到 (d) 中指定软元件中的程序。

### 1) 无EN/ENO函数 (MAXIMUM)

[结构化梯形图/FBD]

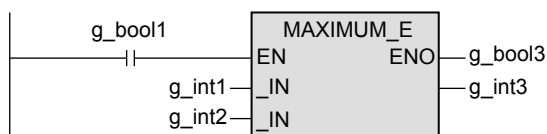


[ST]

```
g_int3:=MAXIMUM(g_int1,g_int2);
```

### 2) 带EN/ENO函数 (MAXIMUM\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=MAXIMUM_E(g_bool1,g_int1,g_int2,g_int3);
```

## 10.3 MINIMUM(\_E) / 选择最小值

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

从数据中选择最小值后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MINIMUM		MINIMUM(_IN, _IN); 例: D20:=MINIMUM(D0, D10);
MINIMUM_E		MINIMUM_E(EN, _IN, _IN, 输出标签); 例: MINIMUM_E(X000, D0, D10, D20);

\*1. 变为输出变量。

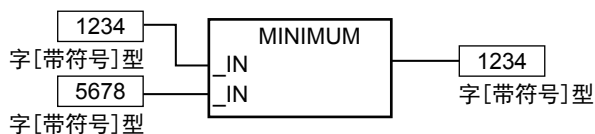
### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN (S1) ~ (S28)	进行比较的数据, 或是保存数据的字软元件	ANY_SIMPLE
输出变量	ENO	执行状态	位
	*1 (D)	保存最小值的字软元件	ANY_SIMPLE

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 1) 将(S1) ~ (S28)中指定的软元件的 ANY\_SIMPLE 型数据的最小值, 按照与(S1) ~ (S28)中指定的软元件相同的数据类型, 输出到(D)中指定的软元件中。  
(例)数据类型为字[带符号]型时



- 2) (S)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

### 注意要点

- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。  
但是, 32位计数器为32位长度的软元件, 因此可以直接指定。  
指定标签时, 使用全局标签。
- 3) 使用编程工具在(S)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

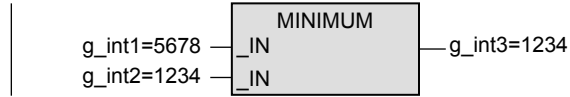


## 程序举例

①将、②中指定的软元件的字[带符号]型数据的最小值，按照与①、②中指定的软元件相同的数据类型，输出到③中指定软元件中的程序。

### 1) 无EN/ENO函数 (MINIMUM)

[结构化梯形图/FBD]

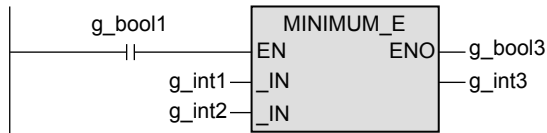


[ST]

```
g_int3:=MINIMUM(g_int1,g_int2);
```

### 2) 带EN/ENO函数 (MINIMUM\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=MINIMUM_E(g_bool1,g_int1,g_int2,g_int3);
```

## 10.4 LIMITATION(E) / 上下限限位控制

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

判断数据是否在上下限值的范围内。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
LIMITATION		LIMITATION(_MN, _IN, _MX); 例: D30:=LIMITATION(D0, D10, D20);
LIMITATION_E		LIMITATION_E(EN, _MN, _IN, _MX, 输出标签); 例: LIMITATION_E(X000, D0, D10, D20, D30);

\*1. 变为输出变量。

### 2. 设定数据

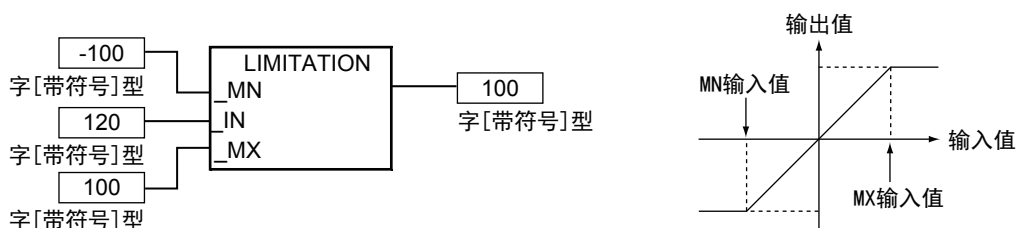
	变量	内容	数据类型
输入变量	EN	执行条件	位
	_MN (s1)	下限限位数据, 或是保存数据的字软元件	ANY_SIMPLE
	_IN (s2)	输入数据, 或是保存数据的字软元件	ANY_SIMPLE
输出变量	_MX (s3)	上限限位数据, 或是保存数据的字软元件	ANY_SIMPLE
	ENO	执行状态	位
	*1 (d)	保存输出数据的字软元件	ANY_SIMPLE

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

根据(s1)、(s2)、(s3)中指定的软元件的ANY\_SIMPLE型数据, 按照与(s1)、(s2)、(s3)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。

- [(s2)指定的软元件的内容 > (s3)指定的软元件的内容]时, 将(s3)中指定的软元件的内容输出到(d)中指定的软元件中。
  - [(s2)指定的软元件的内容 < (s1)指定的软元件的内容]时, 将(s1)中指定的软元件的内容输出到(d)中指定的软元件中。
  - [(s1)指定的软元件的内容 ≤ (s2)指定的软元件的内容 ≤ (s3)指定的软元件的内容]时, 将(s2)中指定的软元件的内容输出到(d)中指定的软元件中。
- (例)数据类型为字[带符号]型时



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。
- 3) 使用编程工具在(S)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

### 错误

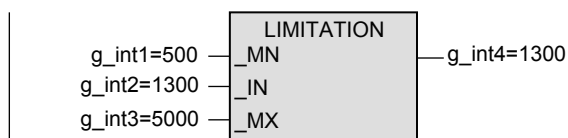
在下述设定状态下执行指令后，会出现运算错误，错误标志位(M8067)为ON，错误代码(K6706)保存在D8067中。  
 (s1)指定的软元件的内容 > (s3)指定的软元件的内容  
 (下限限位数据) (上限限位数据)

### 程序举例

根据(s1)、(s2)、(s3)中指定的软元件的ANY\_SIMPLE型数据，按照与(s1)、(s2)、(s3)中指定的软元件相同的数据类型，输出到(D)中指定的软元件中的程序。

- 1) 无EN/ENO函数 (LIMITATION)

[结构化梯形图/FBD]

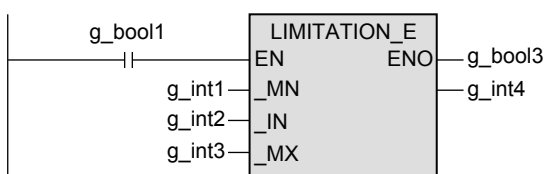


[ST]

```
g_int4:=LIMITATION(g_int1,g_int2,g_int3);
```

- 2) 带EN/ENO函数 (LIMITATION\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=LIMITATION_E(g_bool1,g_int1,g_int2,g_int3,g_int4);
```

## 10.5 MUX(\_E) / 多路选择器

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

选择数据后输出。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MUX		<p>MUX(_K, _IN, _IN); 例: D30:=MUX(D0, D10, D20);</p>
MUX_E		<p>MUX_E(EN, _K, _IN, _IN, 输出标签); 例: MUX_E(X000, D0, D10, D20, D30);</p>

\*1. 变为输出变量。

### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	_K (n)	选择的数据, 或是保存数据的字软元件
	_IN (s1) ~ (s28)	被选择的数据, 或是保存数据的字软元件
输出变量	ENO	执行状态
	*1	保存已选择数据的字软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

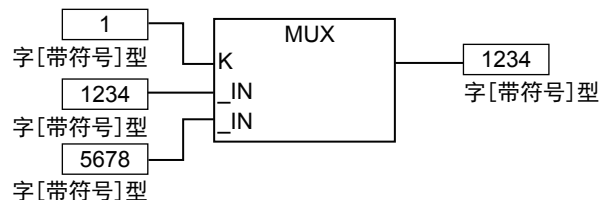
### 功能和动作说明

1) 根据(n)指定的值, 将(s1)~(s28)中指定的软元件的任意值, 按照与(s1)~(s28)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。

a) (n)中指定的值为1时, 则将(s1)中指定的软元件的值输出到(d)中指定的软元件中。

b) (n)的输入值为n时, 则将(sn)中指定的软元件的值输出到(d)中指定的软元件中。

(例)数据类型为字[带符号]时



2) (n)值中输入了超出(s1)~(s28)的PIN数范围的值时, 则向(d)指定的软元件输出不定值。(不出现运算错误。此外, 为MUX\_E时, 则由ENO输出FALSE。)

3) (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

## 注意要点

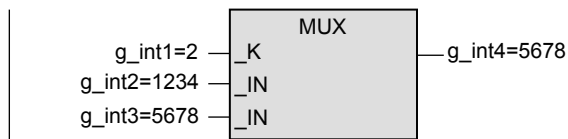
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。
- 3) 使用编程工具在(s)中指定了单精度实数型数据时，可能会产生化整误差。使用编程工具设定输入值时的注意事项，请参考Q/L/F结构化编程手册(基础篇)。

## 程序举例

根据(n)的值，将(s1)、(s2)指定的软元件的任意值，按照与(s1)、(s2)指定的软元件相同的数据类型，输出到(d)指定的软元件中的程序。

- 1) 无EN/ENO函数 (MUX)

[结构化梯形图/FBD]

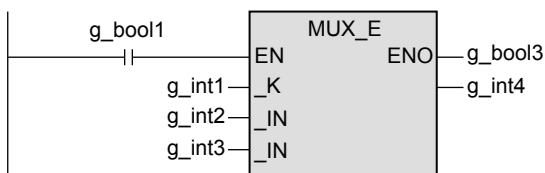


[ST]

```
g_int4:=MUX(g_int1,g_int2,g_int3);
```

- 2) 带EN/ENO函数 (MUX\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=MUX_E(g_bool1,g_int1,g_int2,g_int3,g_int4);
```

## 11. 应用函数 (比较函数)

函数名	功能	参考
GT_E	比较	11.1节
GE_E	比较	11.2节
EQ_E	比较	11.3节
LE_E	比较	11.4节
LT_E	比较	11.5节
NE_E	比较	11.6节

## 11.1 GT\_E / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行 > (是否大于) 比较。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
GT_E		GT_E (EN, _IN, _IN, 输出标签); 例: GT_E (X000, D0, D10, M0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN ( (s1) ~ (s28) )	进行比较的数据, 或是保存数据的软元件	ANY_SIMPLE
输出变量	ENO	执行状态	位
	*1 ( (d) )	保存比较结果的软元件	位

在函数说明中, 输入输出变量通过 ( ) 中的描述进行记载。

### 功能和动作说明

- 对 (s1) ~ (s28) 中指定的软元件的内容进行比较运算, 将运算结果按照位型输出到 (d) 中指定的软元件中。进行 [ (s1) > (s2) ] & [ (s2) > (s3) ] & . . . . . & [ (sn-1) > (sn) ] 的比较。
  - 全部 (sn-1) > (sn) 时, 输出 TRUE。
  - 任意为 (sn-1) ≤ (sn) 时, 输出 FALSE。
- (s) 的 PIN 数可在 2~28 范围内进行更改。

→ 3章 函数的构成

### 注意要点

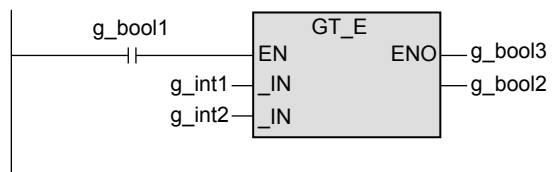
- 在结构化程序中处理 32 位数据时, 不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时, 请使用标签。但是, 32 位计数器为 32 位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在 (s) 中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考 Q/L/F 结构化编程手册 (基础篇)。

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

### 程序举例

对 (s1)、(s2) 中指定的软元件的内容进行比较运算，将运算结果输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=GT_E(g_bool1,g_int1,g_int2,g_bool2);
```



## 11.2 GE\_E / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否  $\geq$  (大于、或等于) 的比较。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
GE_E		GE_E(EN, _IN, _IN, 输出标签); 例: GE_E(X000, D0, D10, M0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	_IN ( <u>s1</u> ~ <u>s28</u> )	进行比较的数据, 或是保存数据的字软元件	ANY_SIMPLE
输出变量	ENO	执行状态	位
	*1 ( <u>d</u> )	保存比较结果的软元件	位

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对 s1 ~ s28 中指定的软元件的内容进行比较运算, 将运算结果按照位型输出到 d 中指定的软元件中。进行  $[\underline{s1} \geq \underline{s2}] \& [\underline{s2} \geq \underline{s3}] \& \dots \& [\underline{s_{n-1}} \geq \underline{s_n}]$  的比较。
  - 全部  $\underline{s_{n-1}} \geq \underline{s_n}$  时, 输出TRUE。
  - 任意为  $\underline{s_{n-1}} < \underline{s_n}$  时, 输出FALSE。
- s 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

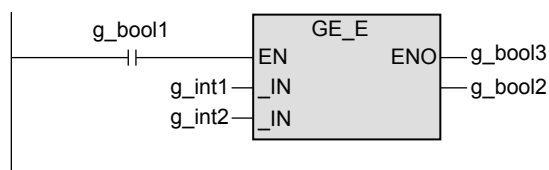
### 注意要点

- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在 s 中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

### 程序举例

对 s1、s2 中指定的软元件的内容进行比较运算, 将运算结果输出到 d 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_bool3:=GE\_E(g\_bool1,g\_int1,g\_int2,g\_bool2);

### 11.3 EQ\_E / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

#### 概要

进行是否= (等于) 的比较。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
EQ_E		EQ_E(EN, _IN, _IN, 输出标签); 例: EQ_E(X000, D0, D10, M0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	
	_IN (S1) ~ (S28)	进行比较的数据, 或是保存数据的软元件	ANY_SIMPLE
输出变量	ENO	执行状态	位
	*1 (D)	保存比较结果的软元件	位

在函数说明中, 输入输出变量通过( )中的描述进行记载。

#### 功能和动作说明

- 对(S1) ~ (S28)中指定的软元件的内容进行比较运算, 将运算结果按照位型输出到(D)中指定的软元件中。进行[(S1)=(S2)] & [(S2)=(S3)] & ... & [(Sn-1)=(Sn)]的比较。
  - 全部(Sn-1)=(Sn)时, 输出TRUE。
  - 任意为(Sn-1)≠(Sn)时, 输出FALSE。
- (S)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

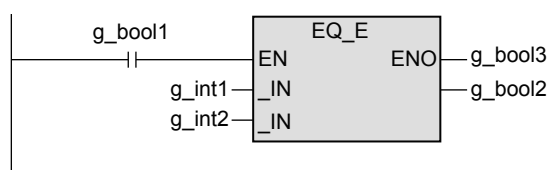
#### 注意要点

- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在(S)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

#### 程序举例

对(S1)、(S2)中指定的软元件的内容进行比较运算, 将运算结果输出到(D)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool3:=EQ_E(g_bool1,g_int1,g_int2,g_bool2);
```

## 11.4 LE\_E / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否≤(小于、或等于)的比较。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
LE_E		LE_E(EN, _IN, _IN, 输出标签); 例: LE_E(X000, D0, D10, M0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
输入变量	EN	执行条件
	_IN (S1) ~ (S28)	进行比较的数据, 或是保存数据的软元件
输出变量	ENO	执行状态
	*1 (D)	保存比较结果的软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(S1) ~ (S28)中指定的软元件的内容进行比较运算, 将运算结果按照位型输出到(D)中指定的软元件中。进行[(S1) ≤ (S2)] & [(S2) ≤ (S3)] & . . . . . & [(Sn-1) ≤ (Sn)]的比较。
  - 全部(Sn-1) ≤ (Sn)时, 输出TRUE。
  - 任意为(Sn-1) > (Sn)时, 输出FALSE。
- (S)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

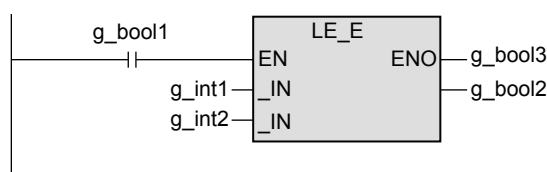
### 注意要点

- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在(S)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

### 程序举例

对(S1)、(S2)中指定的软元件的内容进行比较运算, 将运算结果输出到(D)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_bool3:=LE\_E(g\_bool1,g\_int1,g\_int2,g\_bool2);

## 11.5 LT\_E / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否<(小于)的比较。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
LT_E		LT_E(EN, _IN, _IN, 输出标签); 例: LT_E(X000, D0, D10, M0);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	
	_IN (s1) ~ (s28)	进行比较的数据, 或是保存数据的软元件	ANY_SIMPLE
输出变量	ENO	执行状态	位
	*1 (d)	保存比较结果的软元件	位

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的内容进行比较运算, 将运算结果按照位型输出到(d)中指定的软元件中。进行[(s1) < (s2)] & [(s2) < (s3)] & . . . . . & [(sn-1) < (sn)]的比较。
  - 全部(sn-1) < (sn)时, 输出TRUE。
  - 任意为(sn-1) ≥ (sn)时, 输出FALSE。
- (s)的PIN数可在2~28范围内进行更改。

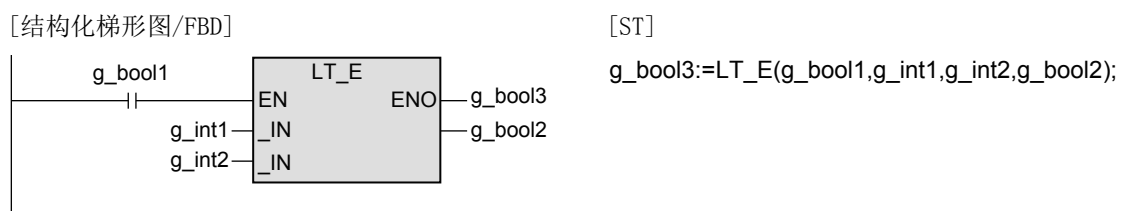
→ 3章 函数的构成

### 注意要点

- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在(s)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

### 程序举例

对(s1)、(s2)中指定的软元件的内容进行比较运算, 将运算结果保存在(d)中指定的软元件中的程序。



## 11.6 NE\_E / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否≠(不等于)的比较。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
NE_E		<p>NE_E(EN, _IN1, _IN2, 输出标签); 例: NE_E(X000, D0, D10, M0);</p>

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	_IN1 (s1)	进行比较的数据, 或是保存数据的字软元件
	_IN2 (s2)	进行比较的数据, 或是保存数据的字软元件
输出变量	ENO	执行状态
	*1 (d)	保存比较结果的软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

对(s1)、(s2)中指定的软元件的内容进行比较运算, 将运算结果按照位型输出到(d)中指定的软元件中。  
进行[(s1)≠(s2)]的比较。

- (s1)≠(s2)时, 输出TRUE。
- (s1)=(s2)时, 输出FALSE。

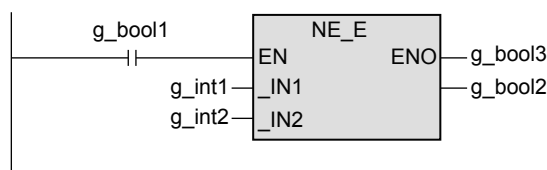
### 注意要点

- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在(s)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

### 注意要点

对(s1)、(s2)中指定的软元件的内容进行比较运算, 将运算结果输出到(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_bool3:=NE\_E(g\_bool1,g\_int1,g\_int2,g\_bool2);

## 12. 应用函数(字符串函数)

函数名	功能	参考
MID(_E)	字符串的选取	12.1节
CONCAT(_E)	字符串的结合	12.2节
INSERT(_E)	字符串的插入	12.3节
DELETE(_E)	字符串的删除	12.4节
REPLACE(_E)	字符串的替换	12.5节
FIND(_E)	字符串的检索	12.6节

## 12.1 MID(E) / 字符串的选取

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

从指定位置取得字符串。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MID		<p>MID(_IN, _L, _P); 例: 标签2:=MID(标签1, D10, D20);</p>
MID_E		<p>MID_E(EN, _IN, _L, _P, 输出标签); 例: MID_E(X000, 标签1, D10, D20, 标签2);</p>

\*1. 变为输出变量。

### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN (s)	保存字符串的起始字软元件	字符串
	_L (n1)	保存取得的字符串的字软元件	字[带符号]
	_P (n2)	保存要取得的字符串起始字符位置的软元件	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	保存已取得的字符串的起始字软元件	字符串

在函数说明中，输入输出变量通过( )中的描述进行记载。

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

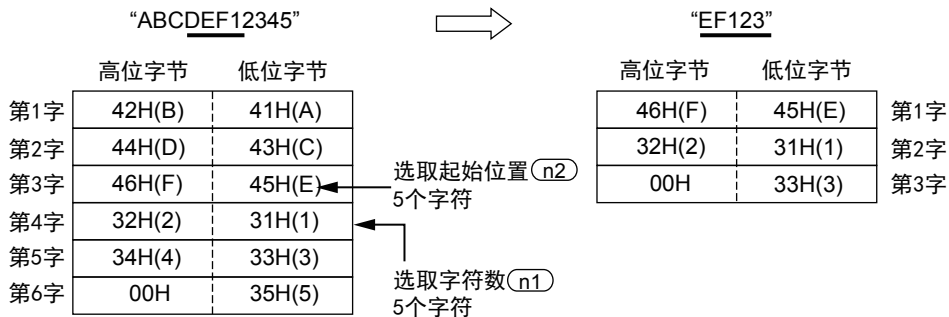
15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

### 功能和动作说明

- 1) 将 (s) 中指定的软元件的字符串任意位置开始的指定字符数的数据，输出到 (d) 中指定的软元件中。要选取的字符数，根据输入到 (n1) 中的指定值进行指定。  
要选取的字符串的起始位置，根据输入到 (n2) 中的指定值进行指定。  
(例)输入到 (n1)、(n2) 中的指定值为5时



- 2) (s) 中指定的字符串(数据)，就是指从被指定的软元件开始到检测到字节单位中第1个「00H」为止的数据。
- 3) (n1) 中指定的要选取的字符数为“0”时不执行处理。
- 4) (n1) 中指定的要取出的字符数为“-1”时，将 (s) 中指定的字符串的最终字符数据输出到 (d) 中指定的软元件中。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

### 错误

以下一些情况下会出现运算错误，错误标志位(M8067)为0N，错误代码保存在D8067中。

- 1) (s) 中指定的软元件开始的相应软元件范围内没有设定「00H」时。  
(错误代码：K6706)
- 2) (n2) 中指定的起始位置超出了 (s) 中指定的字符串的字符数时。  
(错误代码：K6706)
- 3) 从 (d) 中指定的软元件开始，(n1) 指定的字符数超出了 (d) 中指定的软元件范围时。  
(错误代码：K6706)
- 4) (d) 中指定的软元件编号开始的软元件数，比保存已经选取的字符串所需的软元件数量更少时。(在所有的字符串和最终字符中不能保存「00H」)  
(错误代码：K6706)
- 5) (n2) 中指定的值为负时。  
(错误代码：K6706)
- 6) (n1) 中指定的值为-2以下的值时。  
(错误代码：K6706)
- 7) (n1) 中指定的值超出了 (s) 中指定的字符数时。  
(错误代码：K6706)

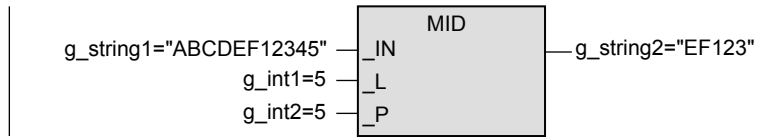


## 程序举例

将从(s)指定的软元件的字符串任意位置取得的指定字符数的数据，输出到(d)中指定的软元件中的程序。

### 1) 无EN/ENO函数(MID)

[结构化梯形图/FBD]

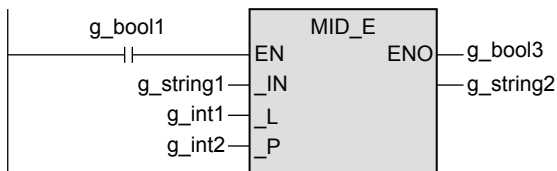


[ST]

```
g_string2:=MID(g_string1,g_int1,g_int2);
```

### 2) 带EN/ENO函数(MID\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=MID_E(g_bool1,g_string1,g_int1,g_int2,g_string2);
```

## 12.2 CONCAT(\_E) / 字符串的结合

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

结合字符串。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
CONCAT		CONCAT(_IN, _IN); 例: 标签3:=CONCAT(标签1, 标签2);
CONCAT_E		CONCAT_E(EN, _IN, _IN, 输出标签); 例: CONCAT_E(X000, 标签1, 标签2, 标签3);

\*1. 变为输出变量。

### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
_IN ( (s1) ~ (s28) )	保存要连接的数据(字符串)的起始软元件, 或是直接指定的字符串	字符串
ENO	执行状态	位
*1 ( (d) )	保存已连接的数据(字符串)的起始软元件	字符串

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 在 (s<sub>n</sub>) 指定的软元件的字符串的后面, 将 (s<sub>n+1</sub>) 中指定的软元件的字符串进行结合后, 输出到 (d) 中指定的软元件中。  
字符串的结合, 就是指忽略表示 (s<sub>n</sub>) 中指定的软元件字符串末尾的“00H”, 对 (s<sub>n+1</sub>) 中指定的软元件的字符串进行连续结合。  
此外, 执行字符串的结合后会自动将“00H”附加在最后。

“ABCDE”		+	“123456”		⇒	“ABCDE123456”	
高位字节	低位字节		高位字节	低位字节		高位字节	低位字节
第1字	42H(B)   41H(A)		第1字	32H(2)   31H(1)		第1字	42H(B)   41H(A)
第2字	44H(D)   43H(C)		第2字	34H(4)   33H(3)		第2字	44H(D)   43H(C)
第3字	00H   45H(E)		第3字	36H(6)   35H(5)		第3字	31H(1)   45H(E)
			第4字	00H		第4字	33H(3)   32H(2)
						第5字	35H(5)   34H(4)
						第6字	00H   36H(6)

- (s) 中指定的字符串(数据), 就是指从被指定的软元件开始到检测到字节单位中第1个「00H」为止的数据。
- 直接指定字符串时, 可以指定(输入)的字符数最多为32个字符。  
(s) 中指定了字软元件时, 字符数没有限制。
- (s) 中任何一个指定值都是从“00H”开始时(字符数为0时), 在 (d) 中指定的软元件中保存“0000H”。
- (s) 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

## 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

## 错误

以下一些情况下会出现运算错误，错误标志位(M8067)为0N，错误代码保存在D8067中。

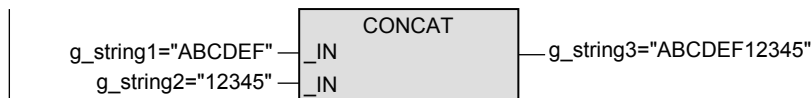
- 1) (d)中指定的软元件编号开始的软元件数，比保存所有已经结合的字符串所需的软元件数量更少时。  
(在所有的字符串和最终字符后面不能保存「00H」)  
(错误代码：K6706)
- 2) (s)当中指定的保存字符串的软元件和(d)中指定的保存字符串的软元件编号重复时。  
(错误代码：K6706)
- 3) (s)中指定的软元件开始的相应软元件范围内没有保存「00H」时。  
(错误代码：K6706)

## 程序举例

在(s1)中指定的软元件的字符串的后面，将(s2)中指定的软元件的字符串进行结合后，输出到(d)中指定的软元件中的程序。

- 1) 无EN/ENO函数(CONCAT)

[结构化梯形图/FBD]

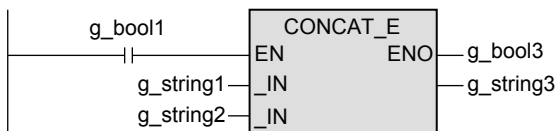


[ST]

```
g_string3:=CONCAT(g_string1,g_string2);
```

- 2) 带EN/ENO函数(CONCAT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=CONCAT_E(g_bool1,g_string1,g_string2,g_string3);
```

## 12.3 INSERT(\_E) / 字符串的插入

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

插入字符串。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
INSERT		INSERT(_IN1, _IN2, _P); <b>例:</b> 标签3:=INSERT(标签1, 标签2, D20);
INSERT_E		INSERT_E(EN, _IN1, _IN2, _P, 输出标签); <b>例:</b> INSERT_E(X000, 标签1, 标签2, D20, 标签3);

\*1. 变为输出变量。

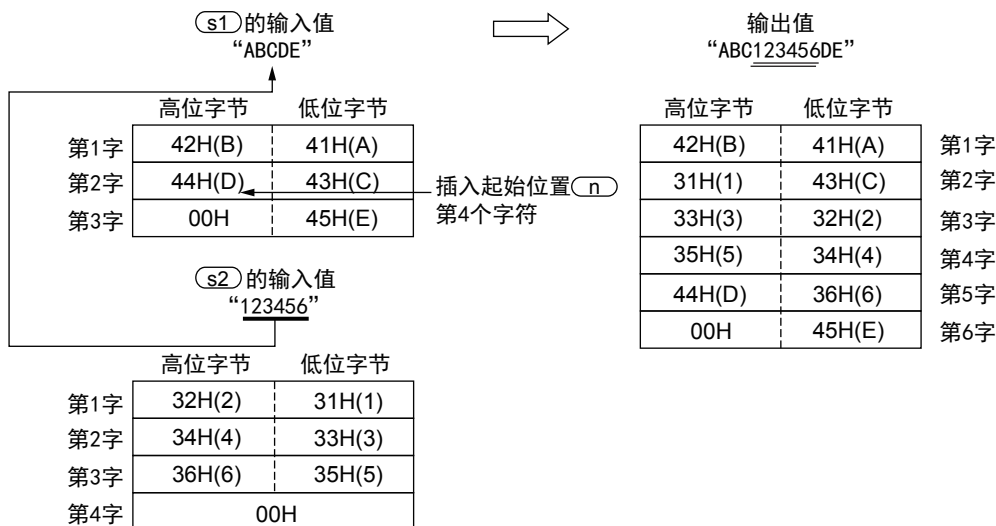
### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN1 (s1)	保存被插入的字符串的起始字软元件	字符串
	_IN2 (s2)	保存要插入的字符串的起始字软元件	字符串
	_P (n)	保存要插入的字符位置的字符软元件	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	保存已被插入的字符串的起始字软元件	字符串

在函数说明中，输入输出变量通过( )中的描述进行记载。

## 功能和动作说明

- 在(s1)中指定的软元件的字符串中，插入从字符串的起始位置到任意位置(s2)中指定的软元件的字符串后，输出到(d)中指定的软元件中。  
要插入的字符串的起始位置，根据输入到(n)中的指示值进行指定。  
将(s2)中指定的软元件的字符串，插入到(s1)中指定的软元件的字符串中后，忽略表示(s2)中指定的字符串末尾的“00H”。  
(例) (n)的指定值为4时



- (s)中指定的字符串(数据)，就是指从被指定的软元件开始到检测到字节单位中第1个「00H」为止的数据。

## 注意要点

- 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

## 错误

以下一些情况下会发生运算错误，错误标志位(M8067)为0N，错误代码保存在D8067中。

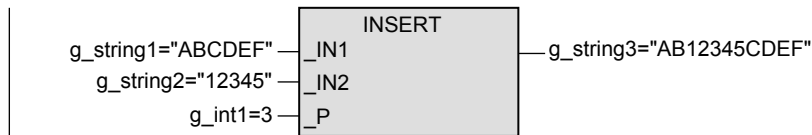
- (d)中指定的软元件编号开始的软元件数，比保存已经插入的字符串的输出所需的软元件数量更少时。  
(错误代码：K6706)
- 当和(s1)、(s2)中指定的保存字符串的软元件和(d)中指定的保存字符串的软元件编号重复时。  
(错误代码：K6706)
- (s1)、(s2)中指定的软元件开始的相应软元件范围内没有保存「00H」时。  
(错误代码：K6706)
- (s2)中指定的软元件的字符数为32768以上时。  
(错误代码：K6706)
- (n)中指定的值为负时。  
(错误代码：K6706)

## 程序举例

在输入到(s1)中指定的软元件的字符串中，插入从字符串的起始位置到任意位置(s2)中指定的软元件的字符串后，输出到(d)中指定的软元件中的程序。

### 1) 无EN/ENO函数(INSERT)

[结构化梯形图/FBD]

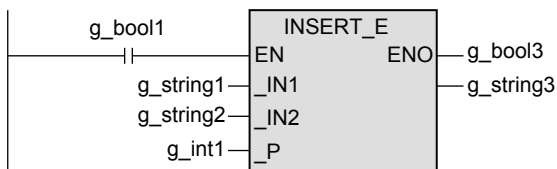


[ST]

```
g_string3:=INSERT(g_string1,g_string2,g_int1);
```

### 2) 带EN/ENO函数(INSERT\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=INSERT_E(g_bool1,g_string1,g_string2,g_int1,g_string3);
```

## 12.4 DELETE(\_E) / 字符串的删除

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

删除字符串。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DELETE		DELETE(_IN, _L, _P); 例: 标签2:=DELETE(标签1, D10, D20);
DELETE_E		DELETE_E(EN, _IN, _L, _P, 输出标签); 例: DELETE_E(X000, 标签1, D10, D20, 标签2);

\*1. 变为输出变量。

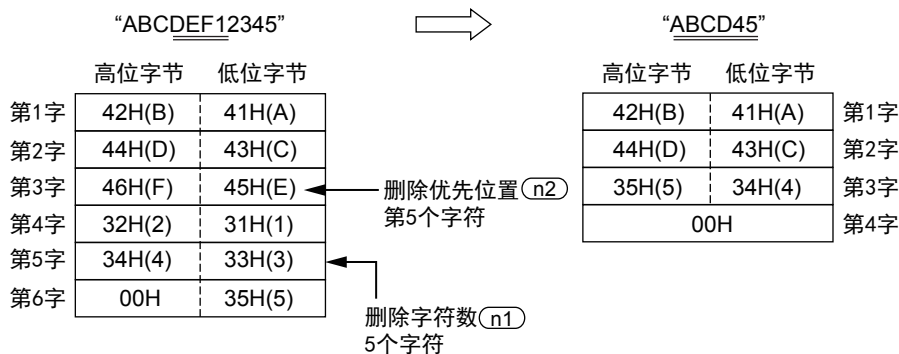
#### 2. 设定数据

变量		内容	数据类型
输入变量	EN	执行条件	位
	_IN (s)	保存被删除字符串的起始字软元件	字符串
	_L (n1)	要删除的字符数	字[带符号]
	_P (n2)	要删除的起始位置	字[带符号]
输出变量	ENO	执行状态	位
	*1 (d)	保存被删除后的剩余字符串的起始字软元件	字符串

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 删除(s)中指定的软元件的字符串任意位置开始的指定字符数的数据，将剩余的字符串输出到(d)中指定的软元件中。  
 要删除字符数，根据输入到(n1)中的指示值进行指定。  
 要删除字符串的起始位置，根据输入到(n2)中的指示值进行指定。  
 (例)输入到(n1)、(n2)的指示值为5时



- (s)中指定的字符串(数据)，就是指从被指定的软元件开始到检测到字节单位中第1个「00H」为止的数据。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

### 错误

以下一些情况下会出现运算错误，错误标志位(M8067)为0N，错误代码保存在D8067中。

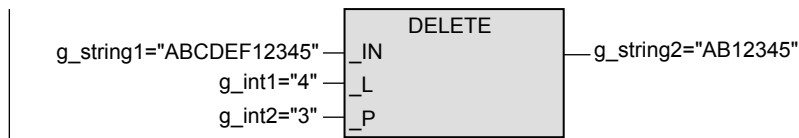
- 1) (s)中指定的软元件开始的相应软元件范围内没有保存「00H」时。  
(错误代码：K6706)
- 2) (s)中指定的软元件的字符数为32768以上时。  
(错误代码：K6706)
- 3) (d)中指定的软元件编号开始的软元件数，比保存删除指定字符数的数据后剩余字符串所需的软元件数量更少时。  
(错误代码：K6706)
- 4) (n2)中指定的值为负时。  
(错误代码：K6706)

### 程序举例

删除(s)中指定的软元件的字符串任意位置开始的指定字符数的数据，将剩余的字符串输出到(d)中指定的软元件中的程序。

- 1) 无EN/ENO函数(DELETE)

[结构化梯形图/FBD]

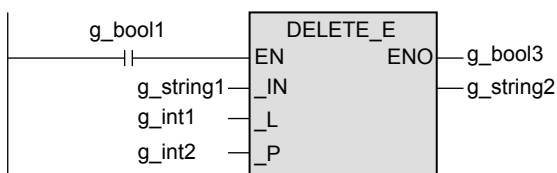


[ST]

```
g_string2:=DELETE(g_string1,g_int1,g_int2);
```

- 2) 带EN/ENO函数(DELETE\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=DELETE_E(g_bool1,g_string1,g_int1,g_int2,g_string2);
```



## 12.5 REPLACE(\_E) / 字符串的替换

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

替换字符串。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
REPLACE		REPLACE(_IN1, _IN2, _L, _P); 例: 标签3:=REPLACE(标签1, 标签2, D20, D30);
REPLACE_E		REPLACE_E(EN, _IN1, _IN2, _L, _P, 输出标签); 例: REPLACE_E(X000, 标签1, 标签2, D20, D30, 标签3);

\*1. 变为输出变量。

#### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN1 (s1)	保存被替换字符串的起始字软元件	字符串
	_IN2 (s2)	保存要替换的字符串的起始字软元件	字符串
	_L (n1)	保存可替换的字符数的字软元件	字[带符号]
输出变量	_P (n2)	保存可替换的字符串的、可替换起始字符位置的软元件	字[带符号]
	ENO	执行状态	位
	*1 (d)	保存已被替换的字符串的起始字软元件	字符串

在函数说明中，输入输出变量通过( )中的描述进行记载。

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

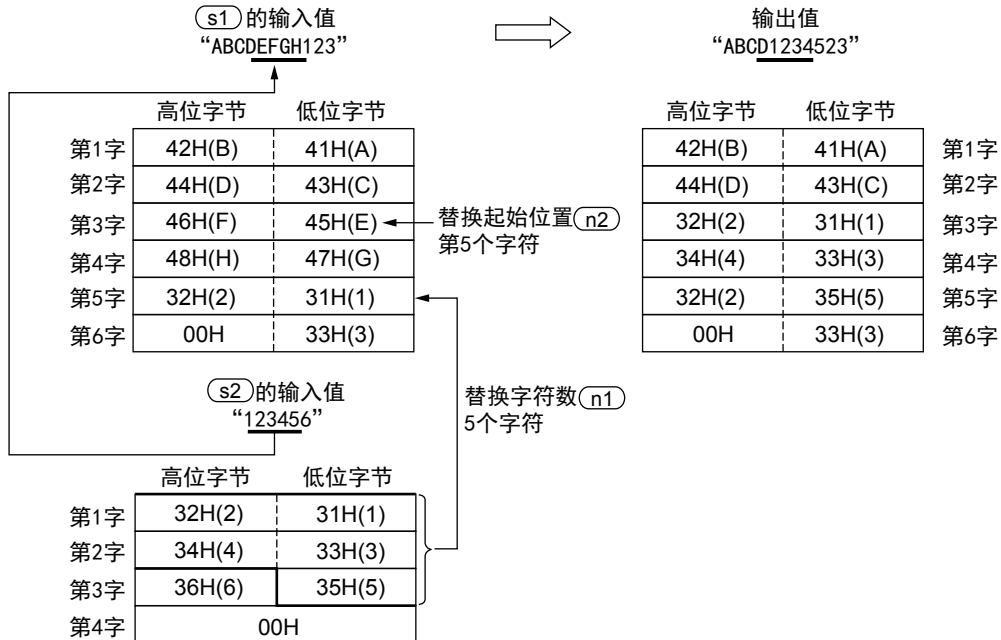
15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

### 功能和动作说明

- 1) 将(s1)中指定的软元件的字符串任意位置开始的指定字符数的数据，替换为(s2)指定的软元件的字符串后，输出到(d)中指定的软元件中。  
要替换的字符数，根据输入到(n1)中的指示值进行指定。  
要替换的字符串的起始位置，根据输入到(n2)中的指示值进行指定。  
(例)输入到(n1)、(n2)的指示值为5时



- 2) (s)中指定的字符串(数据)，就是指从被指定的软元件开始到检测到字节单位中第1个「00H」为止的数据。
- 3) (n1+n2)超出了(s1)中指定的字符串的字符数时，超出的字符不被输出到(d)中指定的软元件中。
- 4) (n1)的指示值为-1时，则将与(s2)中指定的字符串的字符数相同的值作为(n1)的指示值。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时，不能像简洁工程那样直接指定16位软元件。处理字符串数据时，请使用标签。  
指定标签时，使用全局标签。

### 错误

以下一些情况下会出现运算错误，错误标志位(M8067)为0N，错误代码保存在D8067中。

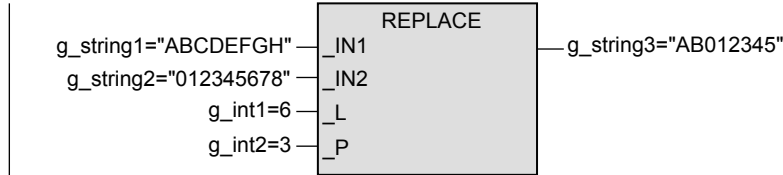
- 1) (s1)、(s2)中指定的软元件编号开始的相应软元件范围内没有保存「00H」时。  
(错误代码：K6706)
- 2) (n1)中指定的值超出了(s2)中指定的字符数时。  
(错误代码：K6706)
- 3) (n2)中指定的值为负时。  
(错误代码：K6706)
- 4) (n1)中指定的值为-2以下的值时。  
(错误代码：K6706)
- 5) (n2)中指定的值超出了(s1)中指定的字符数时。  
(错误代码：K6706)

## 程序举例

将(s1)中指定的软元件的字符串任意位置开始的指定字符数的数据, 替换为(s2)指定的软元件的字符串后, 输出到(d)中指定的软元件中的程序。

### 1) 无EN/ENO函数(REPLACE)

[结构化梯形图/FBD]

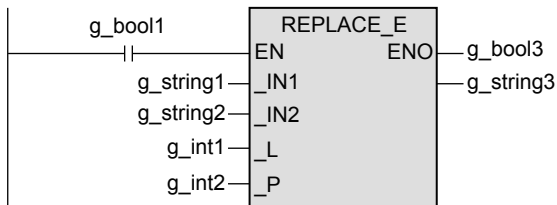


[ST]

```
g_string3:=REPLACE(g_string1,g_string2,g_int1,g_int2);
```

### 2) 带EN/ENO函数(REPLACE\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=REPLACE_E(g_bool1,g_string1,g_string2,g_int1,g_int2,g_string3);
```

## 12.6 FIND(\_E) / 字符串的检索

FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
○	×	×	×	×	×	×	×	×

### 概要

检索字符串。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
FIND		FIND(_IN1, _IN2); 例: D20:=FIND(标签1, 标签2);
FIND_E		FIND_E(EN, _IN1, _IN2, 输出标签); 例: FIND_E(X000, 标签1, 标签2, D20);

\*1. 变为输出变量。

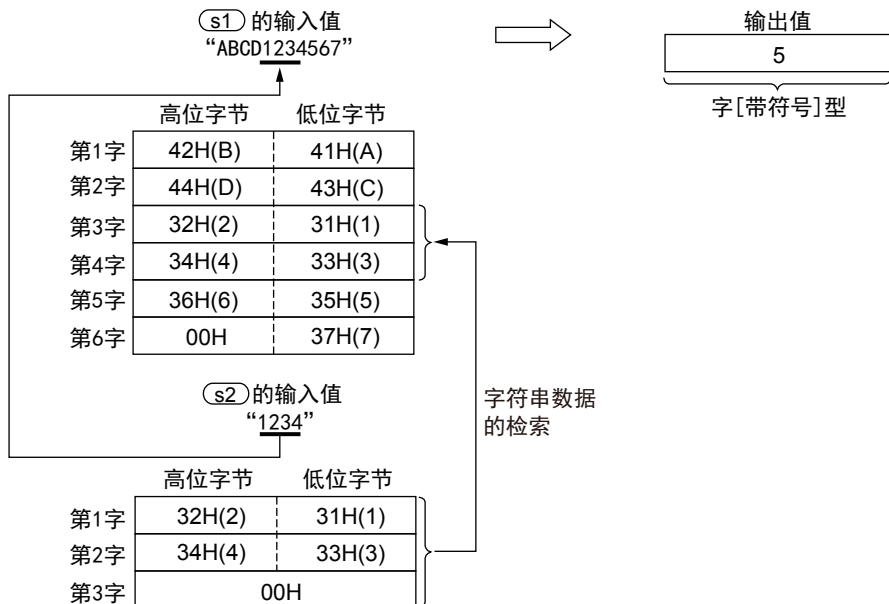
### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN1 (s1)	保存被检索的字符串的起始字软元件	字符串
	_IN2 (s2)	保存要检索的字符串的起始字软元件	字符串
输出变量	ENO	执行状态	位
	*1 (d)	保存检索结果的起始字软元件	字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 1) 从(s1)中指定的软元件的字符串的起始开始,检索(s2)中指定的软元件的字符串,将检索结果输出到(d)中指定的软元件中。  
检索结果,输出第1个检索到的字符串的起始字符位置。
- 2) (s)中指定的字符串(数据),就是指从被指定的软元件开始到检测到字节单位中第1个「00H」为止的数据。
- 3) 从(s1)中指定的软元件的字符串中,无法检索到(s2)中指定的软元件的字符串时,输出0。



### 注意要点

- 1) 如果从母线连接函数,则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理字符串数据时,不能像简洁工程那样直接指定16位软元件。处理字符串数据时,请使用标签。  
指定标签时,使用全局标签。

### 错误

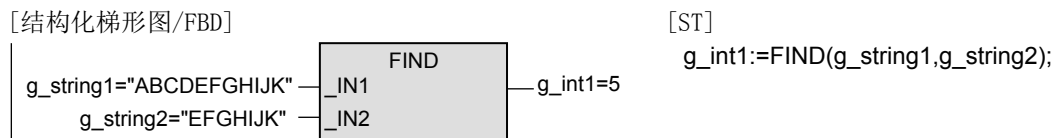
以下一些情况下会出现运算错误,错误标志位(M8067)为0N,错误代码保存在D8067中。

- 1) (s1)中指定的相应软元件的软元件范围内没有00H(NULL)时。  
(错误代码:K6706)
- 2) (s2)中指定的相应软元件的软元件范围内没有00H(NULL)时。  
(错误代码:K6706)

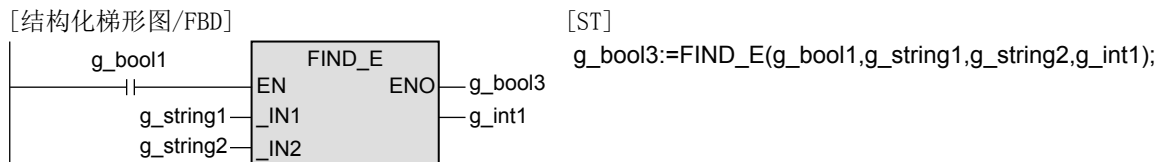
### 程序举例

从(s1)中指定的软元件的字符串的起始开始,检索(s2)中指定的软元件的字符串,将检索结果输出到(d)中指定的软元件中的程序。

- 1) 无EN/ENO函数(FIND)



- 2) 带EN/ENO函数(FIND\_E)



## 13. 应用函数(时间数据类型函数)

函数名	功能	参考
ADD_TIME(_E)	加法运算	13.1节
SUB_TIME(_E)	减法运算	13.2节
MUL_TIME(_E)	乘法运算	13.3节
DIV_TIME(_E)	除法运算	13.4节

## 13.1 ADD\_TIME(\_E) / 加法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

求出时间的加法运算值。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
ADD_TIME		ADD_TIME(_IN1, _IN2); 例: 标签3:=ADD_TIME(标签1, 标签2);
ADD_TIME_E		ADD_TIME_E(EN, _IN1, _IN2, 输出标签); 例: ADD_TIME_E(X000, 标签1, 标签2, 标签3);

\*1. 变为输出变量。

### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_IN1 (s1)	保存被加法运算时间数据的起始字软元件	时间
	_IN2 (s2)	保存加法运算的时间数据的起始字软元件	时间
输出变量	ENO	执行状态	位
	*1 (d)	保存运算结果的起始字软元件	时间

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

对(s1)、(s2)中指定的软元件的时间型数据进行加法运算((s1)+(s2))，将运算结果按照时间型输出到(d)中指定的软元件中。

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

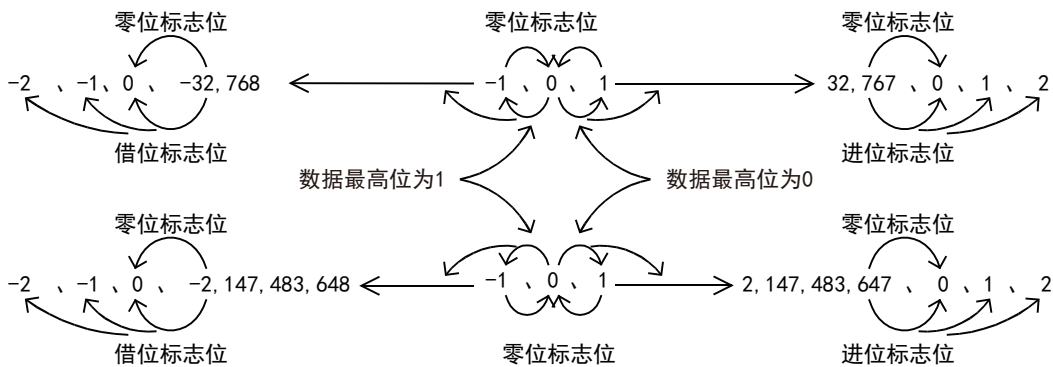
B  
函数、  
操作符  
一览

### 注意要点

- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理 32 位数据时, 不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时, 请使用标签。  
但是, 32 位计数器为 32 位长度的软元件, 因此可以直接指定。  
指定标签时, 使用全局标签。
- 3) 即使运算结果中产生了下溢出、上溢出, 也不出现运算错误。  
此时, 无法获得正确的运算结果, 敬请注意。  
(ADD\_TIME\_E 时, 由 ENO 输出 TRUE。)

此外根据运算结果, 下表中的标志位进行动作。

软元件	名称	内容
M8020	零位	ON : 运算结果为 0 时 OFF: 运算结果为 0 以外时
M8021	借位	ON : 运算结果不到 -32, 768 (16 位运算) 或是 -2, 147, 483, 648 (32 位运算) 时, 借位标志位动作。 OFF: 运算结果超出 -32, 768 (16 位运算) 或是 -2, 147, 483, 648 (32 位运算) 时
M8022	进位	ON : 运算结果超过 32, 767 (16 位运算) 或是 2, 147, 483, 647 (32 位运算) 时, 进位标志位动作。 OFF: 运算结果低于 32, 767 (16 位运算) 或是 2, 147, 483, 647 (32 位运算) 时

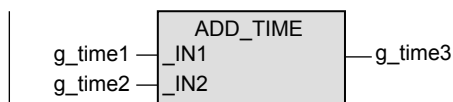


### 程序举例

对 (s1)、(s2) 中指定的软元件的时间型数据进行加法运算 ((s1)+(s2)), 将运算结果按照时间型输出到 (d) 中指定的软元件中的程序。

- 1) 无 EN/ENO 函数 (ADD\_TIME)

[结构化梯形图/FBD]

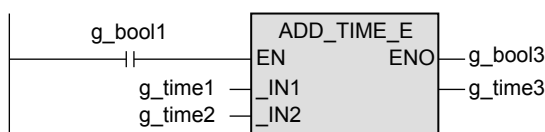


[ST]

```
g_time3:=ADD_TIME(g_time1,g_time2);
```

- 2) 无 EN/ENO 函数 (ADD\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=ADD_TIME_E(g_bool1,g_time1,g_time2,g_time3);
```



## 13.2 SUB\_TIME(\_E) / 减法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

求出时间的减法运算值。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
SUB_TIME		SUB_TIME(_IN1, _IN2); 例: 标签3:=SUB_TIME(标签1, 标签2);
SUB_TIME_E		SUB_TIME_E(EN, _IN1, _IN2, 输出标签); 例: SUB_TIME_E(X000, 标签1, 标签2, 标签3);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	_IN1 (s1)	保存被减法运算时间数据的起始字软元件
	_IN2 (s2)	保存减法运算的时间数据的起始字软元件
输出变量	ENO	执行状态
	*1 (d)	保存运算结果的起始字软元件

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

对(s1)、(s2)中指定的软元件的时间型数据进行减法运算((s1)-(s2))，将运算结果按照时间型输出到(d)中指定的软元件中。

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

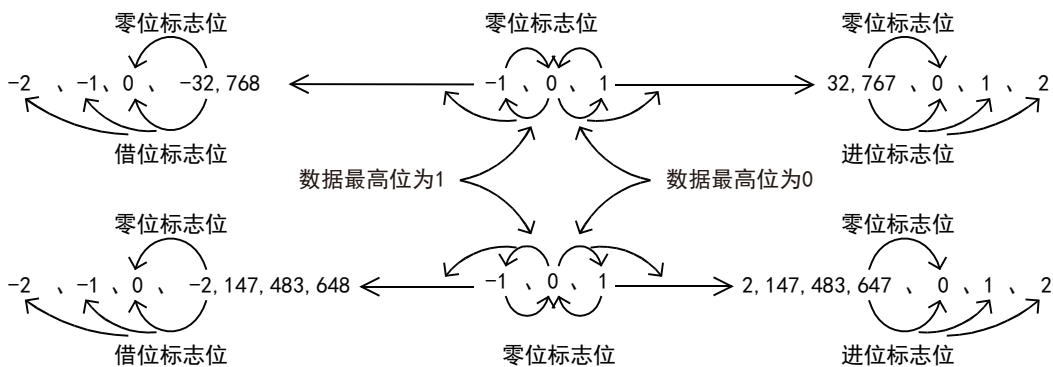
B  
函数、  
操作符  
一览

### 注意要点

- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理 32 位数据时, 不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时, 请使用标签。  
但是, 32 位计数器为 32 位长度的软元件, 因此可以直接指定。  
指定标签时, 使用全局标签。
- 3) 即使运算结果中产生了下溢出、上溢出, 也不出现运算错误。  
此时, 无法获得正确的运算结果, 敬请注意。  
(SUB\_TIME\_E 时, 由 ENO 输出 TRUE。)

此外根据运算结果, 下表中的标志位进行动作。

软元件	名称	内容
M8020	零位	ON : 运算结果为 0 时 OFF: 运算结果为 0 以外时
M8021	借位	ON : 运算结果不到 -32, 768 (16 位运算) 或是 -2, 147, 483, 648 (32 位运算) 时, 借位标志位动作。 OFF: 运算结果超出 -32, 768 (16 位运算) 或是 -2, 147, 483, 648 (32 位运算) 时
M8022	进位	ON : 运算结果超过 32, 767 (16 位运算) 或是 2, 147, 483, 647 (32 位运算) 时, 进位标志位动作。 OFF: 运算结果低于 32, 767 (16 位运算) 或是 2, 147, 483, 647 (32 位运算) 时

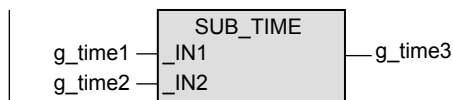


### 程序举例

对 (s1)、(s2) 中指定的软元件的时间型数据进行减法运算 (s1)-(s2), 将运算结果按照时间型输出到 (d) 中指定的软元件中的程序。

- 1) 无 EN/ENO 函数 (SUB\_TIME)

[结构化梯形图/FBD]

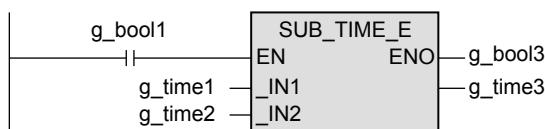


[ST]

```
g_time3:=SUB_TIME(g_time1,g_time2);
```

- 2) 有 EN/ENO 函数 (SUB\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=SUB_TIME_E(g_bool1,g_time1,g_time2,g_time3);
```

### 13.3 MUL\_TIME(\_E) / 乘法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

#### 概要

求出时间的乘法运算值。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
MUL_TIME		MUL_TIME(_IN1, _IN2); 例: 标签3:=MUL_TIME(标签1, 标签2);
MUL_TIME_E		MUL_TIME_E(EN, _IN1, _IN2, 输出标签); 例: MUL_TIME_E(X000, 标签1, 标签2, 标签3);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	_IN1 (s1)	保存被乘数运算时间数据的起始字软元件
	_IN2 (s2)	乘数数据, 或是保存数据的起始字软元件
输出变量	ENO	执行状态
	*1 (d)	保存运算结果的起始字软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

#### 功能和动作说明

对(s1)、(s2)中指定的软元件的时间型数据进行乘法运算((s1) × (s2)), 将运算结果按照时间型输出到(d)中指定的软元件中。

#### 注意要点

- 1) 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理 32 位数据时, 不能像简洁工程那样直接指定 16 位软元件。处理 32 位数据时, 请使用标签。  
但是, 32 位计数器为 32 位长度的软元件, 因此可以直接指定。  
指定标签时, 使用全局标签。
- 3) 即使运算结果中产生了下溢出、上溢出, 也不出现运算错误。  
此时, 无法获得正确的运算结果, 敬请注意。  
(MUL\_TIME\_E 时, 由 ENO 输出 TRUE。)
- 4) 使用编程工具在(s)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考 Q/L/F 结构化编程手册(基础篇)。

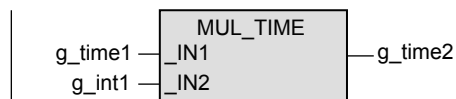
11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

## 程序举例

对 (s1)、(s2) 中指定的软元件的时间型数据进行乘法运算 ( $s1 \times s2$ ), 将运算结果按照时间型输出到 (d) 中指定的软元件中的程序。

### 1) 无EN/ENO函数 (MUL\_TIME)

[结构化梯形图/FBD]

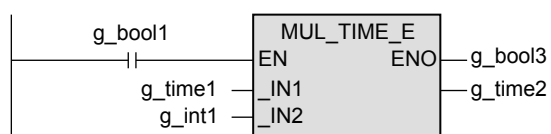


[ST]

```
g_time2:=MUL_TIME(g_time1,g_int1);
```

### 2) 有EN/ENO函数 (MUL\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=MUL_TIME_E(g_bool1,g_time1,g_int1,g_time2);
```

## 13.4 DIV\_TIME(E) / 除法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

求出时间的除法运算值。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
DIV_TIME		DIV_TIME(_IN1, _IN2); 例: 标签3:=DIV_TIME(标签1, 标签2);
DIV_TIME_E		DIV_TIME_E(EN, _IN1, _IN2, 输出标签); 例: DIV_TIME_E(X000, 标签1, 标签2, 标签3);

\*1. 变为输出变量。

#### 2. 设定数据

变量	内容	数据类型
EN	执行条件	位
输入变量	_IN1 (s1)	保存被除数运算时间数据的起始字软元件
	_IN2 (s2)	除数数据, 或是保存数据的起始字软元件
输出变量	ENO	执行状态
	*1 (d)	保存运算结果的起始字软元件

在函数说明中, 输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 对(s1)、(s2)中指定的软元件的时间型数据进行除法运算((s1) ÷ (s2)), 将运算结果按照时间型输出到(d)中指定的软元件中。
- (s2)中指定的软元件的内容为ANY\_NUM型, 但是为0以外的值。

### 注意要点

- 如果从母线连接函数, 则请使用函数名带“\_E”的函数。
- 在结构化程序中处理32位数据时, 不能像简洁工程那样直接指定16位软元件。处理32位数据时, 请使用标签。但是, 32位计数器为32位长度的软元件, 因此可以直接指定。指定标签时, 使用全局标签。
- 使用编程工具在(s)中指定了单精度实数型数据时, 可能会产生化整误差。使用编程工具设定输入值时的注意事项, 请参考Q/L/F结构化编程手册(基础篇)。

### 错误

- (s2)中指定的除数为0时, 会发生运算错误, 并且不能执行函数。
- 运算结果超出2, 147, 483, 647时, 会出现运算错误。

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

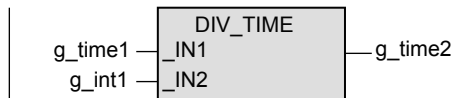
B  
函数、  
操作符  
一览

## 程序举例

对(S1)、(S2)中指定的软元件的时间型数据进行除法运算((S1) ÷ (S2))，将运算结果按照时间型输出到(D)中指定的软元件中的程序。

### 1) 无EN/ENO函数(DIV\_TIME)

[结构化梯形图/FBD]

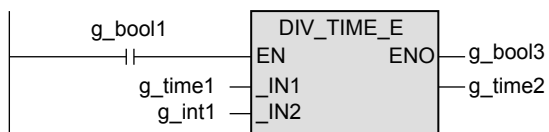


[ST]

```
g_time2:=DIV_TIME(g_time1,g_int1);
```

### 2) 有EN/ENO函数(DIV\_TIME\_E)

[结构化梯形图/FBD]



[ST]

```
g_bool3:=DIV_TIME_E(g_bool1,g_time1,g_int1,g_time2);
```

## 14. 功能模块

函数名	功能	参考
R_TRIG(_E)	上升沿检测	14.1节
F_TRIG(_E)	下降沿检测	14.2节
CTU(_E)	增计数器	14.3节
CTD(_E)	减计数器	14.4节
CTUD(_E)	增减计数器	14.5节
TP(_E) TP_10(_E)	脉冲定时器	14.6节
TON(_E) TON_10(_E)	ON延时定时器	14.7节
TOF(_E) TOF_10(_E)	断开延时定时器	14.8节
COUNTER_FB_M	计数器功能模块	14.9节
TIMER_10_FB_M	定时器功能模块	14.10节
TIMER_CONT_FB_M	定时器功能模块	14.11节
TIMER_100_FB_M	定时器功能模块	14.12节

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## 14.1 R\_TRIG(E) / 上升沿检测

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

检测出信号的上升沿后输出脉冲信号。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
R_TRIG		R_TRIG(_CLK, Q); <b>例:</b> 实例名称(_CLK:=M0, Q:=M10);
R_TRIG_E		R_TRIG_E(EN, _CLK, Q, ENO); <b>例:</b> 实例名称(EN:=X000, _CLK:=M0, Q:=M10);

### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_CLK (s)	检测出上升沿的输入信号	位
输出变量	ENO	执行状态	位
	Q (d)	输出信号	位

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

(s)中指定的软元件为ON时，(d)中指定的软元件只维持1个运算周期为ON。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

### 错误

- 1) (d)中指定的软元件中使用输出编号时，由于变址修饰变成实际上不存在的I/O编号时，M8316(I/O非实际安装的错误)为ON。
- 2) 使用I/O编号以外的软元件(M、T、C)时，由于变址修饰变成实际上不存在的软元件编号时，出现运算错误(错误代码:K6706)。

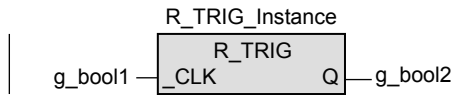


### 程序举例

Ⓢ 中指定的软元件的位型数据从OFF变为ON时，Ⓞ 中指定的软元件只维持1个运算周期为ON的程序。

#### 1) 无EN/ENO函数 (R\_TRIG)

[结构化梯形图/FBD]

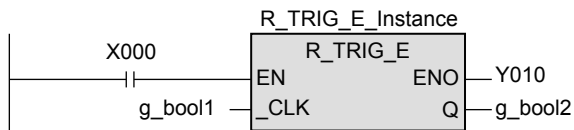


[ST]

```
R_TRIG_Instance(_CLK:=g_bool1,Q:=g_bool2);
```

#### 2) 带EN/ENO函数 (R\_TRIG\_E)

[结构化梯形图/FBD]



[ST]

```
R_TRIG_E_Instance(EN:=X000,_CLK:=g_bool1,Q:=g_bool2,ENO:=Y010);
```

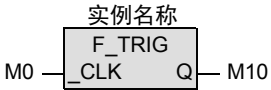
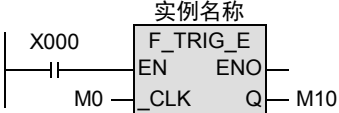
## 14.2 F\_TRIG(\_E) / 下降沿检测

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

检测出信号的下降沿后输出脉冲信号。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
F_TRIG		F_TRIG(_CLK, Q); 例: 实例名称(_CLK:=M0, Q:=M10);
F_TRIG_E		F_TRIG_E(EN, _CLK, Q, ENO); 例: 实例名称(EN:=X000, _CLK:=M0, Q:=M10);

#### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	_CLK (s)	检测出下降沿的输入信号	位
输出变量	ENO	执行状态	位
	Q (d)	输出信号	位

在函数说明中，输入输出变量通过(s)中的描述进行记载。

### 功能和动作说明

(s)中指定的软元件为OFF时，(d)中指定的软元件只维持1个运算周期为ON。

### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

### 错误

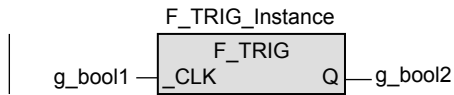
- 1) (d)中指定的软元件中使用输出编号时，由于变址修饰变成实际上不存在的I/O编号时，M8316(I/O非实际安装的错误)为ON。
- 2) 使用I/O编号以外的软元件(M、T、C)时，由于变址修饰变成实际上不存在的软元件编号时，出现运算错误(错误代码:K6706)。

### 程序举例

Ⓢ 中指定的软元件的位型数据从ON变为OFF时，Ⓣ 中指定的软元件只维持1个运算周期为ON的程序。

#### 1) 无EN/ENO函数 (F\_TRIG)

[结构化梯形图/FBD]

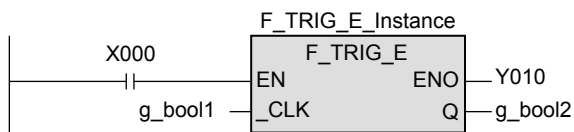


[ST]

```
F_TRIG_Instance(_CLK:=g_bool1,Q:=g_bool2);
```

#### 2) 带EN/ENO函数 (F\_TRIG\_E)

[结构化梯形图/FBD]



[ST]

```
F_TRIG_E_Instance(EN:=X000,_CLK:=g_bool1,Q:=g_bool2,ENO:=Y010);
```

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型  
函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

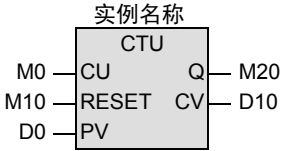
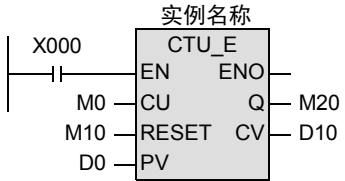
## 14.3 CTU(E) / 增计数器

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	×	×

### 概要

对信号的上升沿次数进行增计数。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
CTU		CTU (CU, RESET, PV, Q, CV); <b>例:</b> 实例名称 (CU:=M0, RESET:=M10, PV:=D0, Q:=M20, CV:=D10);
CTU_E		CTU_E (EN, CU, RESET, PV, Q, CV, ENO); <b>例:</b> 实例名称 (EN:=X000, CU:=M0, RESET:=M10, PV:=D0, Q:=M20, CV:=D10);

#### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	CU ((s1))	计数源信号	位
	RESET ((s2))	复位输入信号	位
	PV ((n))	计数器设定值数据	字[带符号]
输出变量	ENO	执行状态	位
	Q ((d1))	增计数输出信号	位
	CV ((d2))	上升沿计数数据	字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

(s1) 中指定的软元件为ON时，对(d2)中指定的软元件进行加法运算计数(+1)。

计数值达到(n)中指定的值后，(d1)中设定的软元件则ON。

(s2) 中指定的软元件ON时，(d1)中指定的软元件OFF，对(d2)中指定的软元件的计数值进行复位。

### 注意要点

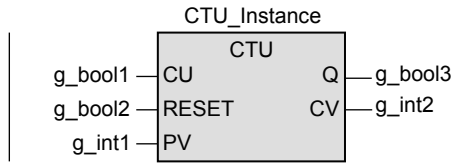
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

## 程序举例

将(s1)中指定的软元件的位型数据从OFF变为ON的次数，输出到(d2)中指定的软元件中的程序。

### 1) 无EN/ENO函数(CTU)

[结构化梯形图/FBD]

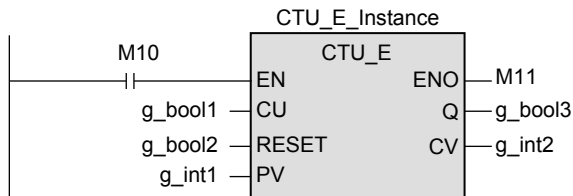


[ST]

```
CTU_Instance(CU:=g_bool1,RESET:=g_bool2,PV:=g_int1,
Q:=g_bool3,CV:=g_int2);
```

### 2) 带EN/ENO函数(CTU\_E)

[结构化梯形图/FBD]



[ST]

```
CTU_E_Instance(EN:=M10,CU:=g_bool1,RESET:=g_bool2,PV:=g_int1,
Q:=g_bool3,CV:=g_int2,ENO:=M11);
```

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型  
数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## 14.4 CTD(\_E) / 减计数器

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	×	×

### 概要

对信号的上升沿次数进行减计数。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
CTD		CTD (CD, LOAD, PV, Q, CV) ; <b>例:</b> 实例名称 (CD:=M0, LOAD:=M10, PV:=D0, Q:=M20, CV:=D10) ;
CTD_E		CTD_E (EN, CD, LOAD, PV, Q, CV, ENO) ; <b>例:</b> 实例名称 (EN:=X000, CD:=M0, LOAD:=M10, PV:=D0, Q:=M20, CV:=D10) ;

### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	CD ((s1))	计数源信号	位
	LOAD ((s2))	复位输入信号	位
	PV ((n))	计数器设定值数据	字[带符号]
输出变量	ENO	执行状态	位
	Q ((d1))	输出信号(计数器当前值为0以下时则ON)	位
	CV ((d2))	上升沿计数数据	字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

(s1) 中指定的软元件为ON时，对(d2)中指定的软元件进行减法运算计数(-1)。

(n) 指定计数的减法运算初始值。

计数值为0时，(d1)中指定的软元件置为ON。

(s2) 中指定的软元件ON时，(d1)中指定的软元件置为OFF，则将(n)中指定的减法运算初始值设置到(d2)中指定的软元件的计数值中。

### 注意要点

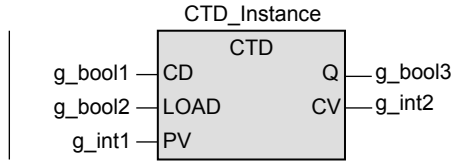
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

## 程序举例

对(s1)中指定的软元件的位型数据从OFF变为ON的次数进行计数，如果(d2)中指定的软元件的内容为0，则将(d1)中指定的软元件置为ON的程序。

### 1) 无EN/ENO函数(CTD)

[结构化梯形图/FBD]

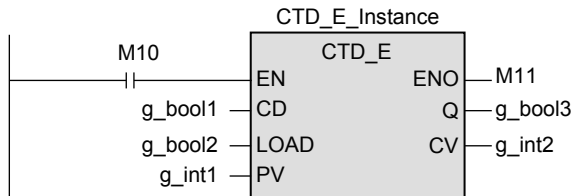


[ST]

```
CTD_Instance(CD:=g_bool1,LOAD:=g_bool2,PV:=g_int1,
Q:=g_bool3,CV:=g_int2);
```

### 2) 带EN/ENO函数(CTD\_E)

[结构化梯形图/FBD]



[ST]

```
CTD_E_Instance(EN:=M10,CD:=g_bool1,LOAD:=g_bool2,PV:=g_int1,
Q:=g_bool3,CV:=g_int2,ENO:=M11);
```

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型  
函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、操作符  
一览

## 14.5 CTUD(\_E) / 增减计数器

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	×	×

### 概要

对信号的上升沿次数进行增减计数。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
CTUD	<p>实例名称</p>	<p>CTUD (CU, CD, RESET, LOAD, PV, QU, QD, CV) ;</p> <p>例: 实例名称 (CU:=M0, CD:=M10, RESET:=M20, LOAD:=M30, PV:=D0, QU:=M40, QD:=M50, CV:=D10) ;</p>
CTUD_E	<p>实例名称</p>	<p>CTUD_E (EN, CU, CD, RESET, LOAD, PV, QU, QD, CV, ENO) ;</p> <p>例: 实例名称 (EN:=X000, CU:=M0, CD:=M10, RESET:=M20, LOAD:=M30, PV:=D0, QU:=M40, QD:=M50, CV:=D10) ;</p>

#### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	CU (s1)	增计数信号	位
	CD (s2)	减计数信号	位
	RESET (s3)	复位输入信号	位
	LOAD (s4)	重设信号	位
	PV (n)	计数器设定值数据	字[带符号]
输出变量	ENO	执行状态	位
	QU (d1)	增计数输出信号	位
	QD (d2)	输出信号(计数器当前值为0以下时则ON)	位
	CV (d3)	计数数据	字[带符号]

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- ①中指定的软元件为ON时，对③中指定的软元件进行加法运算计数(+1)。
- ②中指定的软元件为ON时，对③中指定的软元件进行减法运算计数(-1)。
- ④指定计数的最大值。
- ③中指定的软元件达到计数器的最大值④时，则①中指定的软元件ON。
- ③中指定的软元件为0时，②中指定的软元件ON。
- ③中指定的软元件ON时，对③中指定的软元件的计数值进行复位。
- ④中指定的软元件ON时，将④的指定值置位到③中指定的软元件中。



### 注意要点

- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

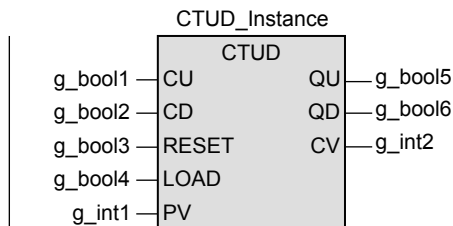
### 程序举例

对(s1)中指定的软元件的位型数据从OFF变为ON的次数进行加法运算计数(+1)，如果(d3)中指定的软元件的值与(n)中指定的内容相同，则将(d1)中指定的软元件置为ON的程序。

此外，同时对(s2)中指定的软元件的位型数据从OFF变为ON的次数进行减法运算计数(-1)，如果(d3)中指定的软元件内容为0，则将(d2)中指定的软元件置为ON的程序。

- 1) 无EN/ENO函数(CTUD)

[结构化梯形图/FBD]

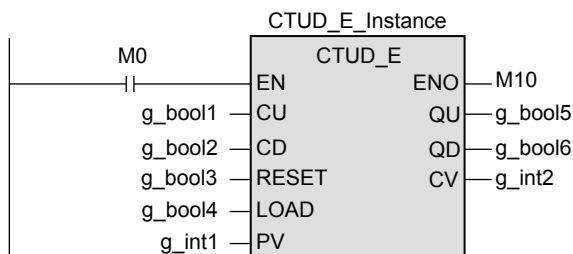


[ST]

```
CTUD_Instance(CU:=g_bool1,CD:=g_bool2,RESET:=g_bool3,LOAD:=g_bool4,PV:=g_int1,
QU:=g_bool5,QD:=g_bool6,CV:=g_int2);
```

- 2) 带EN/ENO函数(CTUD\_E)

[结构化梯形图/FBD]



[ST]

```
CTUD_E_Instance(EN:=M0,CU:=g_bool1,CD:=g_bool2,RESET:=g_bool3,LOAD:=g_bool4,PV:=g_int1,
QU:=g_bool5,QD:=g_bool6,CV:=g_int2,ENO:=M10);
```

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## 14.6 TP(E), TP\_10(E) / 脉冲定时器

	FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
TP(E)	○	○	○	○	○	○	○	×	×
TP_10(E)	○	○	×	○	○	×	○	×	×

### 概要

将指定时间段的信号置为ON。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TP		TP (IN, PT, Q, ET) ; 例: 实例名称 (IN:=M0, PT:=标签1, Q:=M10, ET:=标签2) ;
TP_E		TP_E (EN, IN, PT, Q, ET, ENO) ; 例: 实例名称 (EN:=X000, IN:=M0, PT:=标签1, Q:=M10, ET:=标签2) ;
TP_10		TP_10 (IN, PT, Q, ET) ; 例: 实例名称 (IN:=M0, PT:=标签1, Q:=M10, ET:=标签2) ;
TP_10_E		TP_10_E (EN, IN, PT, Q, ET, ENO) ; 例: 实例名称 (EN:=X000, IN:=M0, PT:=标签1, Q:=M10, ET:=标签2) ;

### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	IN (s)	ON启动输入信号	位
	PT (n)	ON时间数据	时间
输出变量	ENO	执行状态	位
	Q (d1)	输出信号	位
	ET (d2)	ON时间当前值	时间

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- (s)中指定的软元件ON时，在(n)指定的时间段内，将(d1)中指定的软元件置为ON。(d2)中指定的软元件中，(d1)中指定的软元件为ON的经过时间被置位。  
 经过时间达到设定时间后，(d1)中指定的软元件OFF。  
 (d1)中指定的软元件即使OFF，经过时间也不被复位。下一次(s)中指定的软元件从OFF变为ON时，对经过时间进行复位，进行再次输出，将(d1)中指定的软元件置为ON。
- 关于(n)中指定的ON时间数据的设定  
 TP(E) : 可进行100ms单位以上的时间设定。  
 TP\_10(E) : 可进行10ms单位以上的时间设定。

### 注意要点

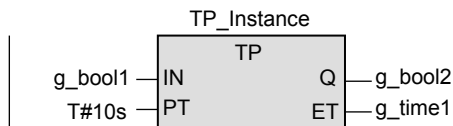
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。但是，32位计数器为32位长度的软元件，因此可以直接指定。指定标签时，使用全局标签。
- 3) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 4) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

### 程序举例

⑤ 中指定的软元件的位型数据ON时，将④ 中指定的软元件的位型数据置为ON 10秒的程序。

- 1) 无EN/ENO函数 (TP, TP\_10)

[结构化梯形图/FBD]

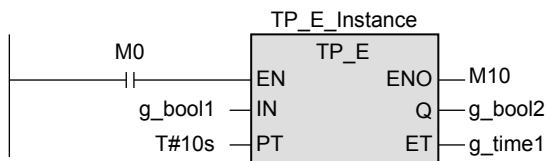


[ST]

TP\_Instance(IN:=g\_bool1,PT:=T#10s,Q:=g\_bool2,ET:=g\_time1);

- 2) 带EN/ENO函数 (TP\_E, TP\_10\_E)

[结构化梯形图/FBD]



[ST]

TP\_E\_Instance(EN:=M0,IN:=g\_bool1,PT:=T#10s,Q:=g\_bool2,ET:=g\_time1,ENO:=M10);

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型  
数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## 14.7 TON(\_E), TON\_10(\_E) / ON延迟定时器

	FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
TON(_E)	○	○	○	○	○	○	○	×	×
TON_10(_E)	○	○	×	○	○	×	○	×	×

### 概要

在指定时间后将信号置为ON。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TON		<p>TON(IN, PT, Q, ET);</p> <p>例: 实例名称(IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);</p>
TON_E		<p>TON_E(EN, IN, PT, Q, ET, ENO);</p> <p>例: 实例名称(EN:=X000, IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);</p>
TON_10		<p>TON_10(IN, PT, Q, ET);</p> <p>例: 实例名称(IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);</p>
TON_10_E		<p>TON_10_E(EN, IN, PT, Q, ET, ENO);</p> <p>例: 实例名称(EN:=X000, IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);</p>

### 2. 设定数据

变量	内容	数据类型	
输入变量	EN	执行条件	位
	IN (s)	输入信号	位
	PT (n)	ON开始时间数据	时间
输出变量	ENO	执行状态	位
	Q (d1)	输出信号	位
	ET (d2)	ON开始时间当前值	时间

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- (s)中指定的软元件为ON时，经过(n)指定的时间后，将(d1)中指定的软元件置为ON。(d2)中指定的软元件中，(d1)中设定的软元件ON之前的延迟经过时间被置位。(s)中指定的软元件为OFF时，(d1)中指定的软元件置为OFF，对延迟经过时间进行复位。
- 关于(n)中指定的ON开始时间数据的设定  
 TON(\_E) :可进行100ms单位以上的时间设定。  
 TON\_10(\_E) :可进行10ms单位以上的时间设定。

### 注意要点

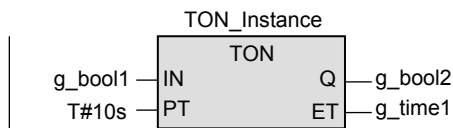
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 3) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

### 程序举例

⑤ 中指定的软元件的位型数据ON时，在10秒后将④ 中指定的软元件的位型数据置为ON的程序。

- 1) 无EN/ENO函数 (TON, TON\_10)

[结构化梯形图/FBD]

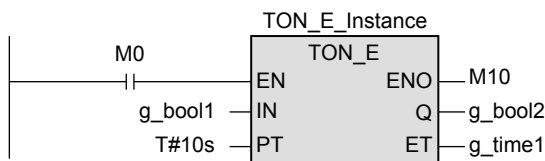


[ST]

TON\_Instance(IN:=g\_bool1,PT:=T#10s,Q:=g\_bool2,ET:=g\_time1);

- 2) 带EN/ENO函数 (TON\_E, TON\_10\_E)

[结构化梯形图/FBD]



[ST]

TON\_E\_Instance(EN:=M0,IN:=g\_bool1,PT:=T#10s,Q:=g\_bool2,ET:=g\_time1,ENO:=M10);

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## 14.8 TOF(\_E), TOF\_10(\_E) / 断开延迟定时器

	FX3U(C)	FX3G(C)	FX3S	FX2N(C)	FX1N(C)	FX1S	FX2(C)	FX0N	FX0(S)
TOF(_E)	○	○	○	○	○	○	○	×	×
TOF_10(_E)	○	○	×	○	○	×	○	×	×

### 概要

输入信号变为OFF后，经过指定的时间后，将输出信号置为OFF。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TOF		TOF(IN, PT, Q, ET); <b>例:</b> 实例名称(IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);
TOF_E		TOF_E(EN, IN, PT, Q, ET, ENO); <b>例:</b> 实例名称(EN:=X000, IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);
TOF_10		TOF_10(IN, PT, Q, ET); <b>例:</b> 实例名称(IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);
TOF_10_E		TOF_10_E(EN, IN, PT, Q, ET, ENO); <b>例:</b> 实例名称(EN:=X000, IN:=M0, PT:=标签1, Q:=M10, ET:=标签2);

### 2. 设定数据

	变量	内容	数据类型
输入变量	EN	执行条件	位
	IN (s)	输入信号	位
	PT (n)	OFF开始时间数据	时间
输出变量	ENO	执行状态	位
	Q (d1)	输出信号	位
	ET (d2)	OFF时间当前值	时间

在函数说明中，输入输出变量通过( )中的描述进行记载。

### 功能和动作说明

- 1) (s)中指定的软元件为ON时，则将(d1)中指定的软元件置为ON。  
 (s)中指定的软元件从ON变为OFF时，经过(n)中指定的时间后，将(d1)中指定的软元件置为OFF。(d2)中指定的软元件中，(d1)中指定的软元件OFF之前的经过时间被置位。(s)中指定的软元件再次ON时，将(d1)中指定的软元件置为ON，对经过时间进行复位。
- 2) 关于(n)中指定的OFF开始时间数据的设定  
 TOF(\_E) :可进行100ms单位以上的时间设定。  
 TOF\_10(\_E) :可进行10ms单位以上的时间设定。

## 注意要点

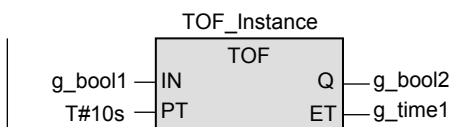
- 1) 如果从母线连接函数，则请使用函数名带“\_E”的函数。
- 2) 在结构化程序中处理32位数据时，不能像简洁工程那样直接指定16位软元件。处理32位数据时，请使用标签。  
但是，32位计数器为32位长度的软元件，因此可以直接指定。  
指定标签时，使用全局标签。
- 3) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。

## 程序举例

将 (s) 中指定的软元件的位型数据为ON时，将 (d1) 中指定的软元件的位型数据置为ON，将 (s) 中指定的软元件置为OFF时，在10秒后将 (d1) 中指定的软元件置为OFF的程序。

- 1) 无EN/ENO函数 (TOF, TOF\_10)

[结构化梯形图/FBD]

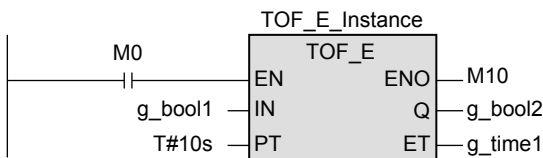


[ST]

```
TOF_Instance(IN:=g_bool1,PT:=T#10s,Q:=g_bool2,ET:=g_time1);
```

- 2) 带EN/ENO函数 (TOF\_E, TOF\_10\_E)

[结构化梯形图/FBD]



[ST]

```
TOF_E_Instance(EN:=M0,IN:=g_bool1,PT:=T#10s,Q:=g_bool2,ET:=g_time1,ENO:=M10);
```


## 14.9 COUNTER\_FB\_M / 计数器功能模块

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

条件从OFF变为ON时进行计数，如果为设定值则进行输出。  
可以设定计数器的初始值。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
COUNTER_FB_M	<p>实例名称</p> 	<p>COUNTER_FB_M (Coil, Preset, ValueIn, ValueOut, Status);</p>

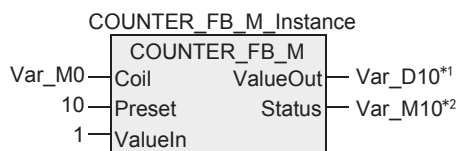
### 2. 设定数据

	变量	内容	数据类型
输入变量	Coil	执行条件	位
	Preset	计数器的设定值	字[带符号]
	ValueIn	计数器的初始值	字[带符号]
输出变量	ValueOut	计数器的当前值	ANY16
	Status	计数器的输出触点	位

### 功能和动作说明

- 1) 检测出输入变量Coil的上升沿(OFF→ON)后进行计数。在Coil为ON的状态下，不进行计数。  
计数从输入变量ValueIn的值开始，变为输入变量Preset的值时，输出变量Status为ON。  
当前的计数值保存在输出变量ValueOut中。

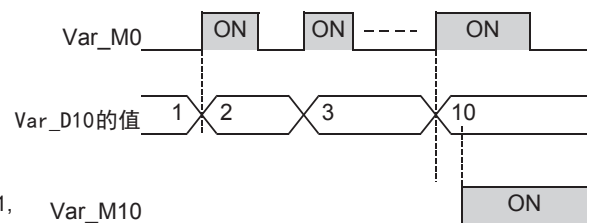
[结构化梯形图/FBD]



[ ST ]

COUNTER\_FB\_M\_Instance(Coil:= Var\_M0,Preset:=10,ValueIn:=1, ValueOut:=Var\_D10,Status:=Var\_M10);

时序图

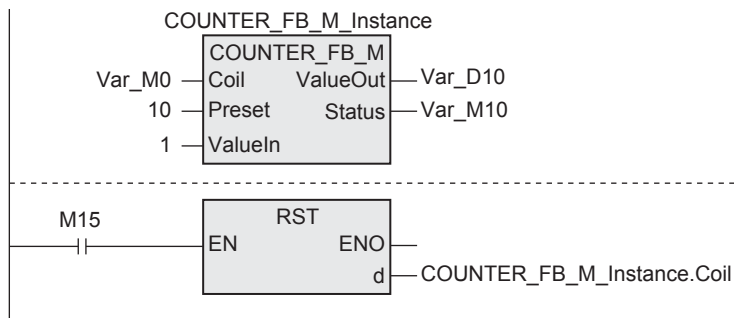


- \*1. Var\_D10中，通过全局标签对D10进行定义。
- \*2. Var\_M10中，通过全局标签对M10进行定义。



- 2) 希望对计数器的当前值进行复位时, 请对输入变量Coil进行复位。

[结构化梯形图/FBD]



[ST]

```
COUNTER_FB_M_Instance(Coil:= Var_M0,Preset:=10,ValueIn:=1,
ValueOut:=Var_D10,Status:=Var_M10);
RST(M15,COUNTER_FB_M_Instance.Coil);
```

### 注意要点

- 关于功能模块在各种语言中的描述
  - 使用功能模块时, 设定实例。
  - 对功能模块进行编程时, 请描述实例名称。
- 为功能模块时, 由于计数器编号为自动分配, 因此需要进行自动分配软元件设定。

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览


## 14.10 TIMER\_10\_FB\_M / 定时器功能模块

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	×	○	○	×	○	×	×

### 概要

在规定时间内条件持续成立时，进行输出的功能模块。  
定时器的初始值和设定值为乘以10ms的值。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIMER_10_FB_M	<p>实例名称</p> 	<p>TIMER_10_FB_M (Coil, Preset, ValueIn, ValueOut, Status);</p>

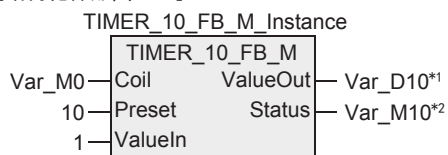
### 2. 设定数据

	变量	内容	数据类型
输入变量	Coil	执行条件	位
	Preset	定时器的设定值	字[带符号]
	ValueIn	定时器的初始值	字[带符号]
输出变量	ValueOut	定时器的当前值	ANY16
	Status	定时器的输出触点	位

### 功能和动作说明

- 1) 输入变量Coil的执行条件为ON时，开始当前值的测量。  
ValueIn×10ms后开始测量，到Preset×10ms为止达到测量值时，输出变量Status为ON。  
当前的测量值被输出到ValueOut中。
- 2) 输入变量Coil的执行条件OFF时，当前值为ValueIn的值，输出变量Status也OFF。

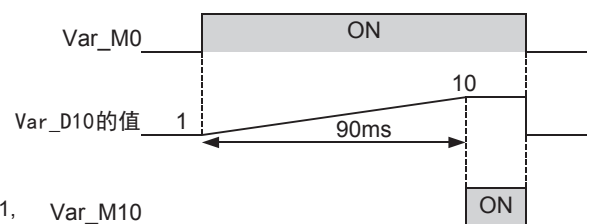
[结构化梯形图/FBD]



[ ST ]

TIMER\_10\_FB\_M\_Instance(Coil:= Var\_M0,Preset:= 10,ValueIn:= 1, ValueOut:= Var\_D10,Status:=Var\_M10);

时序图



- \*1. Var\_D10中，通过全局标签对D10进行定义。
- \*2. Var\_M10中，通过全局标签对M10进行定义。

### 注意要点

- 1) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。
- 2) 为功能模块时，由于定时器编号为自动分配，因此需要进行自动分配软元件设定。


## 14.11 TIMER\_CONT\_FB\_M / 定时器功能模块

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	×	○	×	×

### 概要

测量条件成立的时间，如果到了规定时间，则进行输出的功能模块。

#### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIMER_CONT_FB_M	<p>实例名称</p> 	<p>TIMER_CONT_FB_M (Coil, Preset, ValueIn, ValueOut, Status);</p>

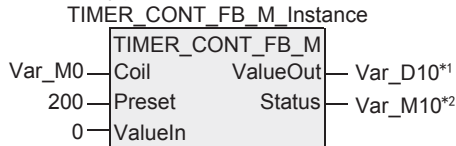
#### 2. 设定数据

	变量	内容	数据类型
输入变量	Coil	执行条件	位
	Preset	定时器的设定值	字[带符号]
	ValueIn	定时器的初始值	字[带符号]
输出变量	ValueOut	定时器的当前值	ANY16
	Status	定时器的输出触点	位

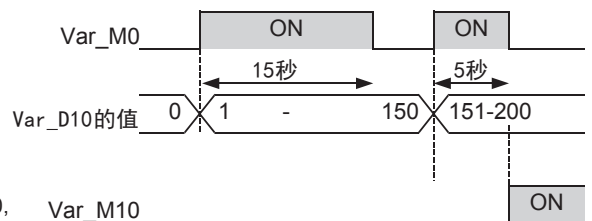
### 功能和动作说明

- 1) 测量变量ON的时间的累计定时器。输入变量Coil的执行条件为ON时，开始当前值的测量。  
ValueIn × (1ms或100ms)后开始测量，到Preset × (1ms或100ms)为止达到测量值时，输出变量Status为ON。当前的测量值被输出到ValueOut中。  
ValueIn和Preset的倍率(1ms或100ms)由编译时的分配软元件(累计定时器)决定。  
分配软元件(累计定时器)
  - FX3S以外  
T246~T249:1ms  
T250~T255:100ms
  - FX3S时  
T128~T131:1ms  
T132~T137:100ms
- 2) 输入变量Coil的执行条件即使为OFF，也保持测量值ValueOut、输出变量Status的ON/OFF状态。输入变量Coil的执行条件再次为ON时，从保持的测量值开始重新开始测量。

[结构化梯形图/FBD]



时序图



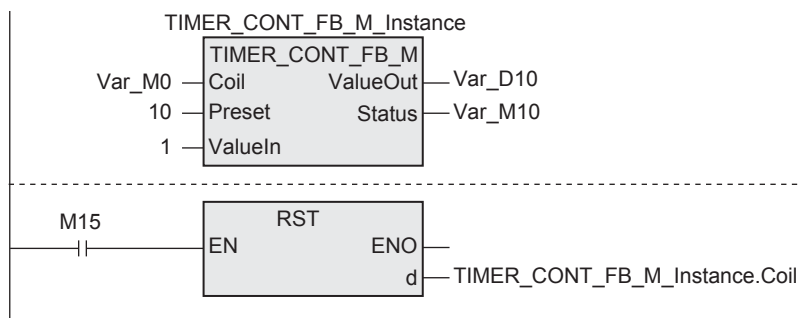
[ST]

```
TIMER_CONT_FB_M_Instance(Coil:= Var_M0, Preset:= 200, ValueIn:= 0, ValueOut:= Var_D10, Status:=Var_M10);
```

- \*1. Var\_D10中，通过全局标签对D10进行定义。
- \*2. Var\_M10中，通过全局标签对M10进行定义。

- 3) 希望对累计定时器的测量值进行复位时，请对输入变量Coil进行复位。

[结构化梯形图/FBD]



[ST]

```
TIMER_CONT_FB_M_Instance(Coil:= Var_M0,Preset:=10,ValueIn:=1,
ValueOut:=Var_D10,Status:=Var_M10);
RST(M15,TIMER_CONT_FB_M_Instance.Coil);
```

### 注意要点

- 1) 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。
- 2) 为功能模块时，由于定时器编号为自动分配，因此需要进行自动分配软元件设定。


## 14.12 TIMER\_100\_FB\_M / 定时器功能模块

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

在规定时间内条件持续成立时，进行输出的功能模块。  
定时器的初始值和设定值为乘以100ms的值。

### 1. 格式

函数名	各种语言的描述	
	结构化梯形图/FBD	ST
TIMER_100_FB_M	<p>实例名称</p> 	<p>TIMER_100_FB_M (Coil, Preset, ValueIn, ValueOut, Status);</p>

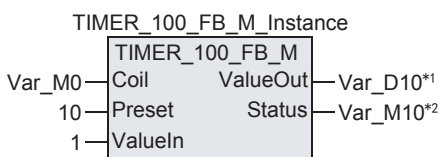
### 2. 设定数据

	变量	内容	数据类型
输入变量	Coil	执行条件	位
	Preset	定时器的设定值	字[带符号]
	ValueIn	定时器的初始值	字[带符号]
输出变量	ValueOut	定时器的当前值	ANY16
	Status	定时器的输出触点	位

### 功能和动作说明

- 输入变量Coil的执行条件为ON时，开始当前值的测量。  
ValueIn×100ms后开始测量，到Preset×100ms为止达到测量值时，输出变量Status为ON。  
当前的测量值被输出到ValueOut中。
- 输入变量Coil的执行条件OFF时，当前值为ValueIn的值，输出变量Status也OFF。

[结构化梯形图/FBD]

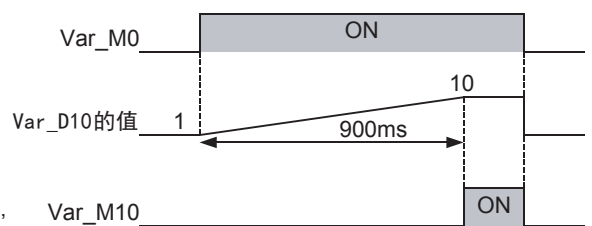


[ST]

```
TIMER_100_FB_M_Instance(Coil:= Var_M0,Preset:= 10,ValueIn:= 1,
ValueOut:= Var_D10,Status:=Var_M10);
```

- \*1. Var\_D10中，通过全局标签对D10进行定义。
- \*2. Var\_M10中，通过全局标签对M10进行定义。

时序图



### 注意要点

- 关于功能模块在各种语言中的描述
  - 使用功能模块时，设定实例。
  - 对功能模块进行编程时，请描述实例名称。
- 为功能模块时，由于定时器编号为自动分配，因此需要进行自动分配软元件设定。

## 15. 操作符

函数名	功能	参考
ADD	加法运算	15.1节
SUB	减法运算	15.2节
MUL	乘法运算	15.3节
DIV	除法运算	15.4节
MOD	余数	15.5节
**	幂乘	15.6节
AND	逻辑与	15.7节
OR	逻辑或	15.8节
XOR	逻辑异或	15.9节
NOT	逻辑非	15.10节
GT	比较	15.11节
GE	比较	15.12节
EQ	比较	15.13节
LE	比较	15.14节
LT	比较	15.15节
NE	比较	15.16节

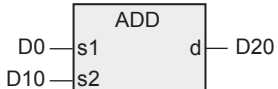
## 15.1 ADD / 加法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将数据进行加法运算 (A+B=C) 后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
ADD		$s1+s2;$ 例: $D20:=D0+D10;$

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

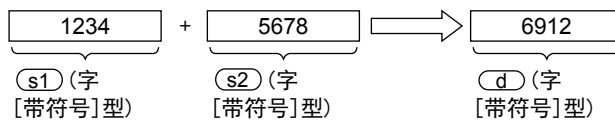
#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	加法运算数据，或是保存数据的字软元件	ANY_NUM
输出变量 (d)	保存运算结果的字软元件	ANY_NUM

### 功能和动作说明

对 (s1) ~ (s28) 中指定的软元件的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型数据的数据进行加法运算 ((s1)+(s2).....+(s28))，将运算结果按照与 (s1) ~ (s28) 中指定的软元件相同的数据类型，输出到 (d) 中指定的软元件中。

(例) 数据类型为字 [带符号] 型时



1) (s) 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

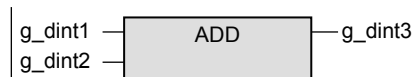
### 注意要点

请参考7.1节。

### 程序举例

对 (s1)、(s2) 中指定的软元件的双字 [带符号] 型数据进行加法运算，将运算结果输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_dint3:=(g\_dint1)+(g\_dint2);

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

## 15.2 SUB / 减法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对2个值进行减法运算(A-B=C)后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
SUB		s1-s2; 例: D20:=D0-D10;

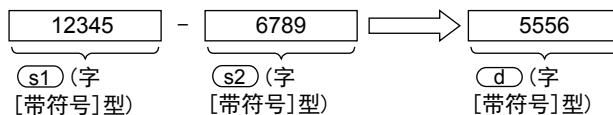
为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	(s1)	被减法运算数据，或是保存数据的字软元件	ANY_NUM
	(s2)	减法运算数据，或是保存数据的字软元件	ANY_NUM
输出变量	(d)	保存运算结果的字软元件	ANY_NUM

### 功能和动作说明

- 对(s1)、(s2)中指定的软元件的字[带符号]型/双字[带符号]型/单精度实数型数据进行减法运算((s1)-(s2))，将运算结果按照与(s1)、(s2)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中。  
(例)数据类型为字[带符号]型时



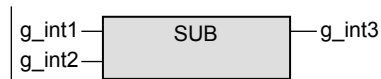
### 注意要点

请参考7.2节。

### 程序举例

对(s1)、(s2)中指定的软元件的字[带符号]型数据进行减法运算，将运算结果输出到(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_int3:=(g_int1)-(g_int2);
```



## 15.3 MUL / 乘法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

将数据进行乘法运算 (A×B=C) 后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
MUL		$s1*s2;$ 例: $D20:=D0*D10;$

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

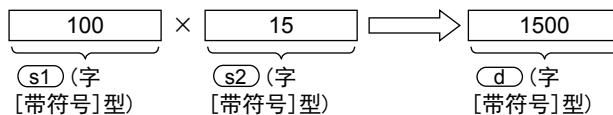
#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	乘数数据，或是保存数据的字软元件	ANY_NUM
输出变量 (d)	保存运算结果的字软元件	ANY_NUM

### 功能和动作说明

- 对 (s1) ~ (s28) 中指定的软元件的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型数据进行乘法运算 ((s1) × (s2) ····· × (s28))，将运算结果按照与 (s1) ~ (s28) 中指定的软元件相同的数据类型，输出到 (d) 中指定的软元件中。

(例) 数据类型为字 [带符号] 型时



- (s) 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

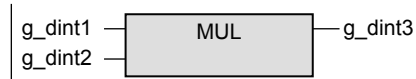
### 注意要点

请参考7.3节。

### 程序举例

对 (s1)、(s2) 中指定的软元件的双字 [带符号] 型数据进行乘法运算，将运算结果输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_dint3:=(g_dint1)*(g_dint2);
```

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

## 15.4 DIV / 除法运算

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对2个值进行除法运算 ( $A \div B = C \cdot \cdot$  余数)后, 输出商。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
DIV		$s1/s2$ ; 例: $D20:=D0/D10$ ;

为了对操作符进行说明, 将输入变量记载为s□, 将输出变量记载为d。

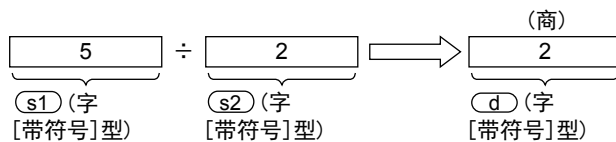
#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1)	被除数数据, 或是保存数据的字软元件	ANY_NUM
(s2)	除数数据, 或是保存数据的字软元件	ANY_NUM
输出变量 (d)	保存运算结果的字软元件	ANY_NUM

### 功能和动作说明

对(s1)、(s2)中指定的软元件的字[带符号]型/双字[带符号]型/单精度实数型数据进行除法运算 ( $(s1) \div (s2)$ ), 将运算结果按照与(s1)、(s2)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。

(例)数据类型为字[带符号]型时



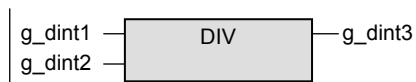
### 注意要点

请参考7.4节。

### 程序举例

对(s1)、(s2)中指定的软元件的双字[带符号]型数据进行除法运算, 将运算结果的商按照与(s1)、(s2)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

$g\_dint3:=(g\_dint1)/(g\_dint2)$ ;

## 15.5 MOD / 余数

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对2个值进行除法运算 ( $A \div B = C \cdot \cdot$  余数)后, 输出余数。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
MOD	- (不支持)	s1 MOD s2; 例: d:=s1 MOD s2;

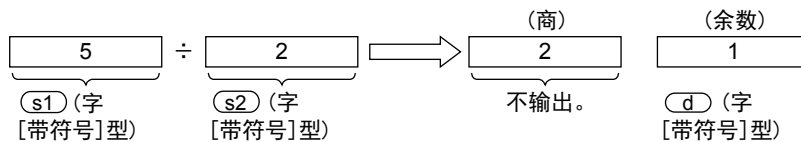
为了对操作符进行说明, 将输入变量记载为s□, 将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1)	被除数数据, 或是保存数据的字软元件	ANY_INT
(s2)	除数数据, 或是保存数据的字软元件	ANY_INT
输出变量 (d)	保存运算结果的字软元件	ANY_INT

### 功能和动作说明

对(s1)、(s2)中指定的软元件的字[带符号]型/双字[带符号]型数据进行除法运算 ( $(s1) \div (s2)$ ), 将运算结果的余数按照与(s1)、(s2)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中。  
(例)数据类型为字[带符号]型时



### 注意要点

请参考7.5节。

### 错误

请参考7.5节。

### 程序举例

对(s1)、(s2)中指定的软元件的双字[带符号]型数据进行除法运算, 将运算结果的余数按照与(s1)、(s2)中指定的软元件相同的数据类型, 输出到(d)中指定的软元件中的程序。

```
[ST]
g_dint3:=g_dint1 MOD g_dint2;
```

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

## 15.6 \*\* / 幂乘

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	×	×	×	×	×	×	×	×

### 概要

求出幂乘后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
**	- (不支持)	s1 **s2; 例: d:=s1 **s2;

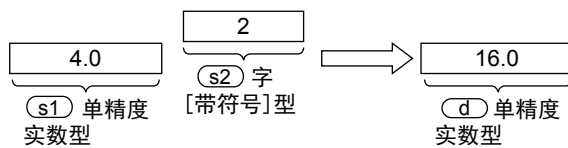
为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型	
输入变量	(s1)	底数数据，或是保存数据的字软元件	单精度实数
	(s2)	指数数据，或是保存数据的字软元件	ANY_NUM
输出变量	(d)	保存运算结果的字软元件	单精度实数

### 功能和动作说明

将(s1)中指定的软元件的单精度实数型数据进行幂乘运算(幂乘，为(s2)中指定的软元件内容)，将运算结果输出到(d)中指定的软元件中。



### 注意要点

请参考7.6节。

### 错误

请参考7.6节。

### 程序举例

(s1)根据、(s2)中指定的软元件内容进行幂乘运算，将运算结果按照与(s1)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中的程序。

```
[ST]
g_real2:=EXPT(g_real1,g_int1);
```

## 15.7 AND / 逻辑与

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对多位的逻辑与进行运算后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
AND		s1 AND s2; 例: M20:=M0 AND M10;

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

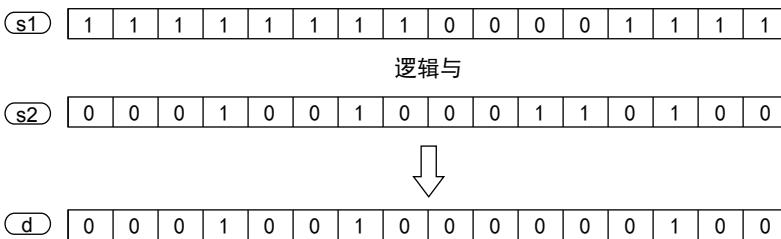
#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行逻辑与运算的软元件	ANY_BIT
输出变量 (d)	保存运算结果的软元件	ANY_BIT

### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]型/位列[32位]型数据的每位进行逻辑与运算，将运算结果按照与(d)中指定的软元件相同的数据类型，输出到(s1) ~ (s28)中指定的软元件中。

(例)数据类型为字[无符号]/位列[16位]型时



- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

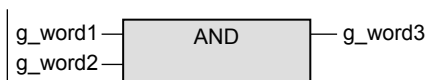
### 注意要点

请参考9.1节。

### 程序举例

对(s1)、(s2)中指定的软元件的字[无符号]/位列[16位]型数据的每位进行逻辑与运算，将运算结果按照与(d)中指定的软元件(s1)相同的数据类型输出到(s2)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_word3:=(g_word1) AND (g_word2);
或
g_word3:=(g_word1) & (g_word2);
```

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

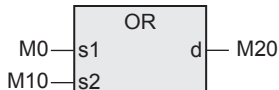
## 15.8 OR / 逻辑或

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对多位的逻辑或进行运算后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
OR		s1 OR s2; 例: M20:=M0 OR M10;

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

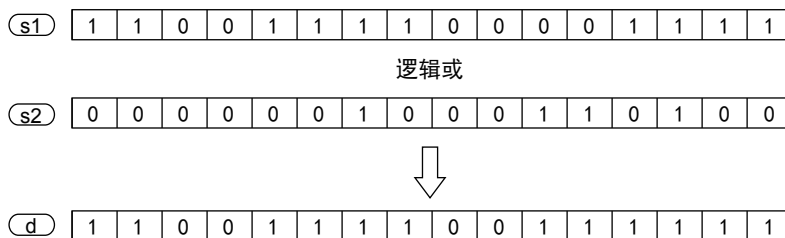
#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行逻辑或运算的软元件	ANY_BIT
输出变量 (d)	保存运算结果的软元件	ANY_BIT

### 功能和动作说明

- 对 (s1) ~ (s28) 中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]型/位列[32位]型数据的每位进行逻辑或运算，将运算结果按照与 (d) 中指定的软元件相同的数据类型，输出到 (s1) ~ (s28) 中指定的软元件中。

(例) 数据类型为字[无符号]/位列[16位]型时



- (s) 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

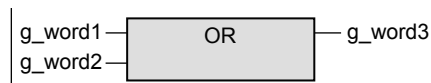
### 注意要点

请参考9.2节。

### 程序举例

对 (s1)、(s2) 中指定的软元件的字[无符号]/位列[16位]型数据的每位进行逻辑或运算，将运算结果按照与 (d) 中指定的软元件 (s1) 相同的数据类型，输出到 (s2) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_word3:=(g\_word1) OR (g\_word2);

## 15.9 XOR / 逻辑异或

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对多位的逻辑异或进行运算后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
XOR		s1 XOR s2; 例: M20:=M0 XOR M10;

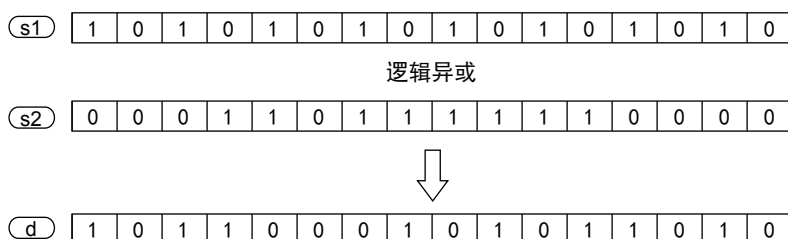
为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行逻辑异或运算的软元件	ANY_BIT
输出变量 (d)	保存运算结果的软元件	ANY_BIT

### 功能和动作说明

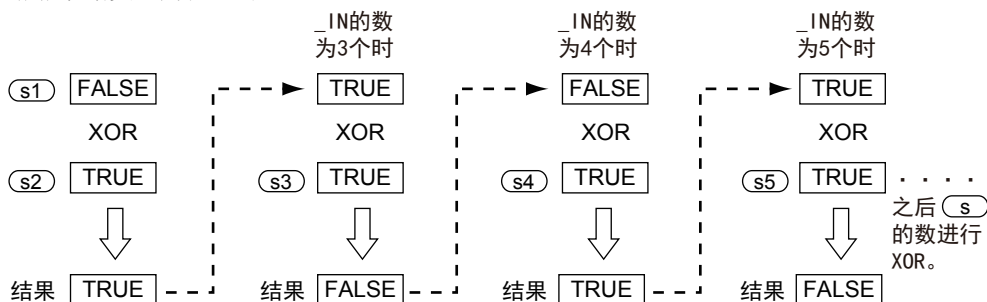
- 对 (s1) ~ (s28) 中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]/位列[32位]型数据的每位进行逻辑异或运算，将运算结果按照与 (d) 中指定的软元件相同的数据类型，输出到 (s1) ~ (s28) 中指定的软元件中。  
(例)数据类型为字[无符号]/位列[16位]型时



- (s) 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

- (s) 有3个以上时，对于对 (s1)、(s2) 进行XOR后的结果，对 (s3) 进行XOR。此外，有 (s4) 时，对于对 (s3) 进行XOR后的结果，对 (s4) 进行XOR。之后，仅 (s5)、(s6) 和输入标签 (s) 数进行XOR。
- (例)数据类型为位型时



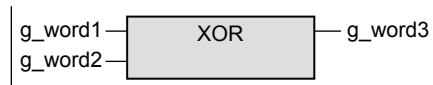
### 注意要点

请参考9.3节。

### 程序举例

对 (s1)、(s2) 中指定的软元件的字 [无符号] / 位列 [16位] 型数据的每位进行逻辑异或运算，将运算结果按照与 (d) 中指定的软元件 (s1) 相同的数据类型，输出到 (s2) 中的程序。

[结构化梯形图/FBD]



[ST]

```
g_word3:=(g_word1) XOR (g_word2);
```



## 15.10 NOT / 逻辑非

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

对位进行逻辑非运算后输出。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
NOT	- (不支持)	NOT s1; 例: d:=NOT(s);

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s)	进行逻辑非运算的软元件	ANY_BIT
输出变量 (d)	保存运算结果的软元件	ANY_BIT

### 功能和动作说明

对(s)中指定的软元件的位型/字[无符号]/位列[16位]型/双字[无符号]/位列[32位]型数据的每位进行逻辑非运算，将运算结果按照与(d)中指定的软元件相同的数据类型(s1)，输出到(s2)中指定的软元件中。  
(例)数据类型为字[无符号]/位列[16位]型时

(s)	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	1
逻辑非																
(d)	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0

### 注意要点

请参考9.4节。

### 程序举例

对(s)中指定的软元件的字[无符号]/位列[16位]型数据的每位进行逻辑非运算，将运算结果按照与(s)中指定的软元件相同的数据类型，输出到(d)中指定的软元件中的程序。

```
[ST]
g_word2:= NOT(g_word1);
```

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

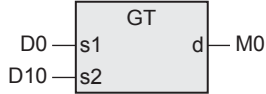
## 15.11 GT / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行 > (是否大于) 比较。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
GT		$s1 > s2;$ 例: $M0 := D0 > D10;$

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行比较的数据，或是保存数据的字软元件	ANY_SIMPLE
输出变量 (d)	保存比较结果的软元件	位

### 功能和动作说明

- 对 (s1) ~ (s28) 中指定的软元件的内容进行比较运算，将运算结果按照位型输出到 (d) 中指定的软元件中。进行 [(s1) > (s2)] & [(s2) > (s3)] & . . . . . & [(sn-1) > (sn)] 的比较。
  - 全部 (sn-1) > (sn) 时，输出TRUE。
  - 任意为 (sn-1) ≤ (sn) 时，输出FALSE。
- (s) 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

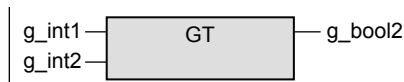
### 注意要点

请参考11.1节。

### 程序举例

对 (s1)、(s2) 中指定的软元件的内容进行比较运算，将运算结果输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool2:=(g_int1)>(g_int2);
```

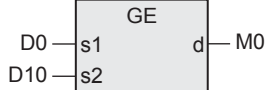
## 15.12 GE / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否 $\geq$ (大于、或等于)的比较。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
GE		$s1 \geq s2$ ; 例: $M0 := D0 \geq D10$ ;

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行比较的数据，或是保存数据的字软元件	ANY_SIMPLE
输出变量 (d)	保存比较结果的软元件	位

### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的内容进行比较运算，将运算结果按照位型输出到(d)中指定的软元件中。进行[(s1)  $\geq$  (s2)] & [(s2)  $\geq$  (s3)] & . . . . . & [(sn-1)  $\geq$  (sn)]的比较。
  - 全部(sn-1)  $\geq$  (sn)时，输出TRUE。
  - 任意为(sn-1) < (sn)时，输出FALSE。
- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

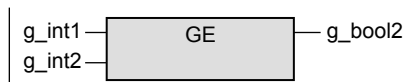
### 注意要点

请参考11.2节。

### 程序举例

对(s1)、(s2)中指定的软元件的内容进行比较运算，将运算结果输出到(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool2:=(g_int1)>=(g_int2);
```

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 软元件和地址的对应  
B 函数、操作符一览

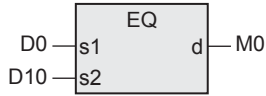
## 15.13 EQ / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否= (等于) 的比较。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
EQ		s1=s2; 例: M0:=D0=D10;

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行比较的数据，或是保存数据的字软元件	ANY_SIMPLE
输出变量 (d)	保存比较结果的软元件	位

### 功能和动作说明

- 对 (s1) ~ (s28) 中指定的软元件的内容进行比较运算，将运算结果按照位型输出到 (d) 中指定的软元件中。进行 [(s1)=(s2)] & [(s2)=(s3)] & . . . . . & [(sn-1)=(sn)] 的比较。
  - 全部 (sn-1)=(sn) 时，输出TRUE。
  - 任意为 (sn-1)≠(sn) 时，输出FALSE。
- (s) 的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

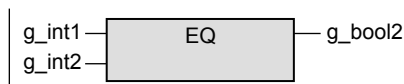
### 注意要点

请参考11.3节。

### 程序举例

对 (s1)、(s2) 中指定的软元件的内容进行比较运算，将运算结果输出到 (d) 中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_bool2:=(g\_int1)=(g\_int2);

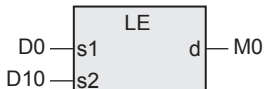
## 15.14 LE / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否≤(小于、或等于)的比较。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
LE		$s1 \leq s2;$ 例: $M0 := D0 \leq D10;$

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行比较的数据，或是保存数据的字软元件	ANY_SIMPLE
输出变量 (d)	保存比较结果的软元件	位

### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的内容进行比较运算，将运算结果按照位型输出到(d)中指定的软元件中。进行[(s1) ≤ (s2)] & [(s2) ≤ (s3)] & . . . . . & [(sn-1) ≤ (sn)]的比较。
  - 全部(sn-1) ≤ (sn)时，输出TRUE。
  - 任意为(sn-1) > (sn)时，输出FALSE。
- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

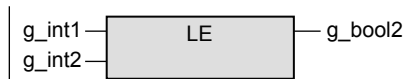
### 注意要点

请参考11.4节。

### 程序举例

对(s1)、(s2)中指定的软元件的内容进行比较运算，将运算结果输出到(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

g\_bool2:=(g\_int1)<=(g\_int2);

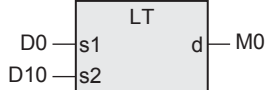
## 15.15 LT / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否<(小于)的比较。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
LT		$s1 < s2;$ 例: $M0 := D0 < D10;$

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量	内容	数据类型
输入变量 (s1) ~ (s28)	进行比较的数据，或是保存数据的字软元件	ANY_SIMPLE
输出变量 (d)	保存比较结果的软元件	位

### 功能和动作说明

- 对(s1) ~ (s28)中指定的软元件的内容进行比较运算，将运算结果按照位型输出到(d)中指定的软元件中。进行[(s1) < (s2)] & [(s2) < (s3)] & . . . . . & [(sn-1) < (sn)]的比较。
  - 全部(sn-1) < (sn)时，输出TRUE。
  - 任意为(sn-1) ≥ (sn)时，输出FALSE。
- (s)的PIN数可在2~28范围内进行更改。

→ 3章 函数的构成

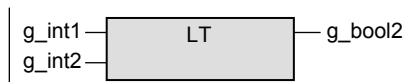
### 注意要点

请参考11.5节。

### 程序举例

对(s1)、(s2)中指定的软元件的内容进行比较运算，将运算结果保存在(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool2:=(g_int1)<(g_int2);
```

## 15.16 NE / 比较

FX3U (C)	FX3G (C)	FX3S	FX2N (C)	FX1N (C)	FX1S	FX2 (C)	FX0N	FX0 (S)
○	○	○	○	○	○	○	○	○

### 概要

进行是否≠(不等于)的比较。

#### 1. 格式

操作符名	各种语言的描述	
	结构化梯形图/FBD	ST
NE		s1<>s2; 例: M0:=D0<>D10;

为了对操作符进行说明，将输入变量记载为s□，将输出变量记载为d。

#### 2. 设定数据

变量		内容	数据类型
输入变量	(s1), (s2)	进行比较的数据，或是保存数据的字软元件	ANY_SIMPLE
输出变量	(d)	保存比较结果的软元件	位

### 功能和动作说明

对(s1)、(s2)中指定的软元件的内容进行比较运算，将运算结果按照位型输出到(d)中指定的软元件中。进行[(s1)≠(s2)]的比较。

- (s1)≠(s2)时，输出TRUE。
- (s1)=(s2)时，输出FALSE。

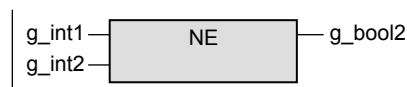
### 注意要点

请参考11.6节。

### 程序举例

对(s1)、(s2)中指定的软元件的内容进行比较运算，将运算结果输出到(d)中指定的软元件中的程序。

[结构化梯形图/FBD]



[ST]

```
g_bool2:=(g_int1)<>(g_int2);
```

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## MEMO



## 附录A. 软元件和地址的对应

软元件和地址的对应关系如下所示。

软元件		记载方法		软元件和地址的对应示例	
		软元件	地址	软元件	地址
输入继电器	X	Xn	%IXn	X367	%IX247
输出继电器	Y	Yn	%QXn	Y367	%QX247
辅助继电器	M	Mn	%MX0. n	M499	%MX0. 499
定时器	触点	TS	Tn	TS191	%MX3. 191
	线圈	TC	Tn	TC191	%MX5. 191
	当前值	TN	Tn	TN190	%MW3. 191 %MD3. 190
计数器	触点	CS	Cn	CS99	%MX4. 99
	线圈	CC	Cn	CC99	%MX6. 99
	当前值	CN	Cn	CN98	%MW4. 99 %MD4. 98
数据寄存器	D	Dn	%MWO. n %MDO. n	D198* D198	%MW0. 199 %MD0. 198
智能功能单元软元件	G	Ux\Gn	%MW14. x. n %MD14. x. n	U0\G09 U0\G09	%MW14. 0. 10 %MD14. 0. 9
扩展寄存器	R	Rn	%MW2. n %MD2. n	R32766 R32766	%MW2. 32767 %MD2. 32766
扩展文件寄存器	ER	ERn	无对应	—	—
指针	P	Pn	""(空字符)	P4095	无对应
中断指针	I	In	无对应	—	—
嵌套	N	Nn	无对应	—	—
变址寄存器	Z	Zn	%MW7. n %MD7. n	Z6 Z6	%MW7. 7 %MD7. 6
	V	Vn	%MW6. n	V7	%MW6. 7
状态	S	Sn	%MX2. n	S4095	%MX2. 4095

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## 附录B. 函数、操作符一览 【按种类 / 字母顺序】

### 附录B-1 【种类】

#### 1. 型转换函数

函数名	功能	参考页
BOOL_TO_INT(E)	将位型数据转换为字[带符号]型数据	38
BOOL_TO_DINT(E)	将位型数据转换为双字[带符号]型数据	40
BOOL_TO_STR(E)	将位型数据转换为字符串型数据	42
BOOL_TO_WORD(E)	将位型数据转换为字[无符号]/位列[16位]型数据	44
BOOL_TO_DWORD(E)	将位型数据转换为双字[无符号]/位列[32位]型数据	46
BOOL_TO_TIME(E)	将位型数据转换为时间型数据	48
INT_TO_DINT(E)	将字[带符号]型数据转换为双字[带符号]型数据	50
DINT_TO_INT(E)	将双字[带符号]型数据转换为字[带符号]型数据	52
INT_TO_BOOL(E)	将字[带符号]型数据转换为位型数据	54
DINT_TO_BOOL(E)	将双字[带符号]型数据转换为位型数据	56
INT_TO_REAL(E)	将字[带符号]型数据转换为单精度实数型数据	58
DINT_TO_REAL(E)	将双字[带符号]型数据转换为单精度实数型数据	60
INT_TO_STR(E)	将字[带符号]型数据转换为字符串型数据	62
DINT_TO_STR(E)	将双字[带符号]型数据转换为字符串型数据	64
INT_TO_WORD(E)	将字[带符号]型数据转换为字[无符号]/位列[16位]型数据	66
DINT_TO_WORD(E)	将双字[带符号]型数据转换为字[无符号]/位列[16位]型数据	68
INT_TO_DWORD(E)	将字[带符号]型数据转换为双字[无符号]/位列[32位]型数据	70
DINT_TO_DWORD(E)	将双字[带符号]型数据转换为双字[无符号]/位列[32位]型数据	72
INT_TO_BCD(E)	将字[带符号]型数据转换为BCD型数据	74
DINT_TO_BCD(E)	将双字[带符号]型数据转换为BCD型数据	76
INT_TO_TIME(E)	将字[带符号]型数据转换为时间型数据	78
DINT_TO_TIME(E)	将双字[带符号]型数据转换为时间型数据	80
REAL_TO_INT(E)	将单精度实数型数据转换为字[带符号]型数据	82

函数名	功能	参考页
REAL_TO_DINT(E)	将单精度实数型数据转换为双字[带符号]型数据	84
REAL_TO_STR(E)	将单精度实数型数据转换为字符串型数据	86
WORD_TO_BOOL(E)	将字[无符号]/位列[16位]型数据转换为位型数据	89
DWORD_TO_BOOL(E)	将双字[无符号]/位列[32位]型数据转换为位型数据	91
WORD_TO_INT(E)	将字[无符号]/位列[16位]型数据转换为字[带符号]型数据	93
WORD_TO_DINT(E)	将字[无符号]/位列[16位]型数据转换为双字[带符号]型数据	95
DWORD_TO_INT(E)	将双字[无符号]/位列[32位]型数据转换为字[带符号]型数据	97
DWORD_TO_DINT(E)	将双字[无符号]/位列[32位]型数据转换为双字[带符号]型数据	99
WORD_TO_DWORD(E)	将字[无符号]/位列[16位]型数据转换为双字[无符号]/位列[32位]型数据	101
DWORD_TO_WORD(E)	将双字[无符号]/位列[32位]型数据转换为字[无符号]/位列[16位]型数据	103
WORD_TO_TIME(E)	将字[无符号]/位列[16位]型数据转换为时间型数据	105
DWORD_TO_TIME(E)	将双字[无符号]/位列[32位]型数据转换为时间型数据	107
STR_TO_BOOL(E)	将字符串型数据转换为位型数据	109
STR_TO_INT(E)	将字符串型数据转换为字[带符号]型数据	111
STR_TO_DINT(E)	将字符串型数据转换为双字[带符号]型数据	113
STR_TO_REAL(E)	将字符串型数据转换为单精度实数型数据	115
STR_TO_TIME(E)	将字符串型数据转换为时间型数据	118
BCD_TO_INT(E)	将BCD数据转换为字[带符号]型数据	120
BCD_TO_DINT(E)	将BCD数据转换为双字[带符号]型数据	122
TIME_TO_BOOL(E)	将时间型数据转换为位型数据	124
TIME_TO_INT(E)	将时间型数据转换为字[带符号]型数据	126
TIME_TO_DINT(E)	将时间型数据转换为双字[带符号]型数据	128

函数名	功能	参考页
TIME_TO_STR(_E)	将时间型数据转换为字符串型数据	130
TIME_TO_WORD(_E)	将时间型数据转换为字[无符号]/位列[16位]型数据	132
TIME_TO_DWORD(_E)	将时间型数据转换为双字[无符号]/位列[32位]型数据	134
BITARR_TO_INT(_E)	将位阵列中的指定位数转换为字[带符号]型、字[无符号]/位列[16位]型数据	136
BITARR_TO_DINT(_E)	将位阵列中的指定位数转换为双字[带符号]型、双字[无符号]/位列[32位]型数据	138
INT_TO_BITARR(_E)	将字[带符号]型、字[无符号]/位列[16位]型数据的低n位输出到位阵列	140
DINT_TO_BITARR(_E)	将双字[带符号]型、双字[无符号]/位列[32位]型数据的低n位输出到位阵列中	142
CPY_BITARR(_E)	按指定位大小复制位阵列	144
GET_BIT_OF_INT(_E)	读出字[带符号]型数据的指定位的值	146
SET_BIT_OF_INT(_E)	将值写入到字[带符号]型数据的指定位中	148
CPY_BIT_OF_INT(_E)	将字[带符号]型数据的指定位复制到其它字[带符号]型数据的指定位中	150
GET_BOOL_ADDR	将起始数据作为位型数据进行输出	152
GET_INT_ADDR	将起始数据作为字[带符号]型数据进行输出	153
GET_WORD_ADDR	将起始数据作为字[无符号]/位列[16位]型数据进行输出	154

## 2. 单数值变量函数

函数名	功能	参考页
ABS(_E)	绝对值运算	156

## 3. 算术运算函数

函数名	功能	参考页
ADD_E	数据的加法运算(PIN数可变)	159
SUB_E	数据的减法运算	161
MUL_E	数据的乘法运算(PIN数可变)	163
DIV_E	数据的除法运算(输出商)	164
MOD(_E)	数据的除法运算(输出余数)	166
EXPT(_E)	数据的幂乘	168
MOVE(_E)	数据的传送	170

## 4. 位移位函数

函数名	功能	参考页
SHL(_E)	左移	173
SHR(_E)	右移	175

## 5. 位型布尔函数

函数名	功能	参考页
AND_E	逻辑与(PIN数可变)	178
OR_E	逻辑或(PIN数可变)	180
XOR_E	逻辑异或(PIN数可变)	181
NOT(_E)	逻辑非	183

## 6. 选择函数

函数名	功能	参考页
SEL(_E)	根据输入条件选择数据	186
MAXIMUM(_E)	从数据中选择最大值(PIN数可变)	188
MINIMUM(_E)	从数据中选择最小值(PIN数可变)	190
LIMITATION(_E)	判断数据是否在上下限值的范围内	192
MUX(_E)	选择数据后输出(PIN数可变)	194

## 7. 比较函数

函数名	功能	参考页
GT_E	是否>(大于)的比较(PIN数可变)	197
GE_E	进行是否≥(大于、或等于)的比较(PIN数可变)	199
EQ_E	是否=(等于)的比较(PIN数可变)	200
LE_E	进行是否≤(小于、或等于)的比较(PIN数可变)	201
LT_E	是否<(小于)的比较(PIN数可变)	202
NE_E	是否≠(不等于)的比较	203

11 应用函数 (比较函数)  
12 应用函数 (字符串函数)  
13 应用函数 (时间数据类型函数)  
14 功能模块  
15 操作符  
A 元件和地址的对应  
B 函数、操作符一览

### 8. 字符串函数

函数名	功能	参考页
MID(_E)	从指定位置取得字符串	205
CONCAT(_E)	字符串的连接(PIN数可变)	208
INSERT(_E)	字符串的插入	210
DELETE(_E)	字符串的删除	213
REPLACE(_E)	字符串的替换	215
FIND(_E)	字符串的检索	218

### 9. 时间数据型函数

函数名	功能	参考页
ADD_TIME(_E)	时间的加法运算	221
SUB_TIME(_E)	时间的减法运算	223
MUL_TIME(_E)	时间的乘法运算	225
DIV_TIME(_E)	时间的除法运算	227

### 10. 功能模块

函数名	功能	参考页
R_TRIG(_E)	检测出信号的上升沿后输出脉冲信号	230
F_TRIG(_E)	检测出信号的下降沿后输出脉冲信号	232
CTU(_E)	对信号的上升沿次数进行增计数	234
CTD(_E)	对信号的上升沿次数进行减计数	236
CTUD(_E)	对信号的上升沿次数进行增/减计数	238
TP(_E) TP_10(_E)	在指定时间段将信号置为ON	240
TON(_E) TON_10(_E)	经过指定时间后将信号置为ON	242
TOF(_E) TOF_10(_E)	输入信号变为OFF后, 经过指定时间后将输出信号置为OFF	244
COUNTER_FB_M	计数器的驱动	246
TIMER_10_FB_M	10ms定时器的驱动	248
TIMER_CO NT_FB_M	累计定时器的驱动	249
TIMER_100_FB_M	100ms定时器的驱动	251

### 11. 操作符(算数运算)

操作符名		功能	参考页
结构化梯形图/FBD	ST		
ADD	+	数据的加法运算(PIN数可变)	253
SUB	-	数据的减法运算	254
MUL	*	数据的乘法运算(PIN数可变)	255
DIV	/	数据的除法运算(输出商)	256
-	MOD	数据的除法运算(输出余数)	257
-	**	数据的幂乘	258

### 12. 操作符(位型布尔)

操作符名		功能	参考页
结构化梯形图/FBD	ST		
AND	& AND	逻辑与(PIN数可变)	259
OR	OR	逻辑或(PIN数可变)	260
XOR	XOR	逻辑异或(PIN数可变)	261
-	NOT	逻辑非	263

### 13. 操作符(比较)

操作符名		功能	参考页
结构化梯形图/FBD	ST		
GT	>	数据比较(是否大于)(PIN数可变)	264
GE	≥	数据比较(是否大于或等于)(PIN数可变)	265
EQ	=	数据比较(是否等于)	266
LE	≤	数据比较(是否小于或等于)(PIN数可变)	267
LT	<	数据比较(是否小于)(PIN数可变)	268
NE	<>	数据比较(是否不等于)	269

附录B-2 【按字母顺序】

函数		
函数名	功能	参考页
<b>A</b>		
ABS(_E)	绝对值运算	156
ADD_TIME(_E)	时间的加法运算	221
ADD_E	数据的加法运算(PIN数可变)	159
AND_E	逻辑与(PIN数可变)	178
<b>B</b>		
BCD_TO_DINT(_E)	将BCD数据转换为双字[带符号]型数据	122
BCD_TO_INT(_E)	将BCD数据转换为字[带符号]型数据	120
BITARR_TO_DINT(_E)	将位阵列中的指定位数转换为双字[带符号]型、双字[无符号]/位列[32位]型数据	138
BITARR_TO_INT(_E)	将位阵列中的指定位数转换为字[带符号]型、字[无符号]/位列[16位]型数据	136
BOOL_TO_DINT(_E)	将位型数据转换为双字[带符号]型数据	40
BOOL_TO_DWORD(_E)	将位型数据转换为双字[无符号]/位列[32位]型数据	46
BOOL_TO_DINT(_E)	将位型数据转换为字[带符号]型数据	38
BOOL_TO_STRING(_E)	将位型数据转换为字符串型数据	42
BOOL_TO_TIME(_E)	将位型数据转换为时间型数据	48
BOOL_TO_WORD(_E)	将位型数据转换为字[无符号]/位列[16位]型数据	44
<b>C</b>		
CONCAT(_E)	字符串的连接(PIN数可变)	208
COUNTER_FB_M	计数器的驱动	246
CPY_BITARR(_E)	按指定位大小复制位阵列	144
CPY_BIT_OFF_INT(_E)	将字[带符号]型数据的指定位复制到其它字[带符号]型数据的指定位中	150
CTD(_E)	对信号的上升沿次数进行减计数	236
CTUD(_E)	对信号的上升沿次数进行增/减计数	238
CTU(_E)	对信号的上升沿次数进行增计数	234
<b>D</b>		
DELETE(_E)	字符串的删除	213
DINT_TO_BCD(_E)	将双字[带符号]型数据转换为BCD型数据	76
DINT_TO_BITARR(_E)	将双字[带符号]型、双字[无符号]/位列[32位]型数据的低n位输出到位阵列中	142
DINT_TO_BOOL(_E)	将双字[带符号]型数据转换为位型数据	56
DINT_TO_DWORD(_E)	将双字[带符号]型数据转换为双字[无符号]/位列[32位]型数据	72

函数名	功能	参考页
<b>D</b>		
DINT_TO_INT(_E)	将双字[带符号]型数据转换为字[带符号]型数据	52
DINT_TO_REAL(_E)	将双字[带符号]型数据转换为单精度实数型数据	60
DINT_TO_STRING(_E)	将双字[带符号]型数据转换为字符串型数据	64
DINT_TO_TIME(_E)	将双字[带符号]型数据转换为时间型数据	80
DINT_TO_WORD(_E)	将双字[带符号]型数据转换为字[无符号]/位列[16位]型数据	68
DIV_TIME(_E)	时间的除法运算	227
DIV_E	数据的除法运算(输出商)	164
DWORD_TO_BOOL(_E)	将双字[无符号]/位列[32位]型数据转换为位型数据	91
DWORD_TO_DINT(_E)	将双字[无符号]/位列[32位]型数据转换为双字[带符号]型数据	99
DWORD_TO_INT(_E)	将双字[无符号]/位列[32位]型数据转换为字[带符号]型数据	97
DWORD_TO_TIME(_E)	将双字[无符号]/位列[32位]型数据转换为时间型数据	107
DWORD_TO_WORD(_E)	将双字[无符号]/位列[32位]型数据转换为字[无符号]/位列[16位]型数据	103
<b>E</b>		
EQ_E	是否=(等于)的比较(PIN数可变)	200
EXPT(_E)	数据的幂乘	168
<b>F</b>		
FIND(_E)	字符串的检索	218
F_TRIG(_E)	检测出信号的下降沿后输出脉冲信号	232
<b>G</b>		
GE_E	是否≥(大于、或等于)的比较(PIN数可变)	199
GET_BIT_OFF_INT(_E)	读出字[带符号]型数据的指定位的值	146
GET_BOOL_ADDR	将起始数据作为位型数据进行输出	152
GET_INT_ADDR	将起始数据作为字[带符号]型数据进行输出	153
GET_WORD_ADDR	将起始数据作为字[无符号]/位列[16位]型数据进行输出	154
GT_E	是否>(大于)的比较(PIN数可变)	197
<b>I</b>		
INSERT(_E)	字符串的插入	210
INT_TO_BCD(_E)	将字[带符号]型数据转换为BCD型数据	74
INT_TO_BITARR(_E)	将字[带符号]型、字[无符号]/位列[16位]型数据的低n位输出到位阵列	140
INT_TO_BOOL(_E)	将字[带符号]型数据转换为位型数据	54
INT_TO_DINT(_E)	将字[带符号]型数据转换为双字[带符号]型数据	50

11 应用函数  
(比较函数)

12 应用函数  
(字符串函数)

13 应用函数  
(时间数据类型函数)

14 功能模块

15 操作符

A 软元件和地址的对应

B 函数、操作符一览

函数名	功能	参考页
<b>I</b>		
INT_TO_DWORD(E)	将字[带符号]型数据转换为双字[无符号]/位列[32位]型数据	70
INT_TO_REAL(E)	将字[带符号]型数据转换为单精度实数型数据	58
INT_TO_STRING(E)	将字[带符号]型数据转换为字符串型数据	62
INT_TO_TIME(E)	将字[带符号]型数据转换为时间型数据	78
INT_TO_WORD(E)	将字[带符号]型数据转换为字[无符号]/位列[16位]型数据	66
<b>L</b>		
LE_E	是否≤(小于、或等于)的比较(PIN数可变)	201
LIMITATION(E)	判断数据是否在上下限值的范围内	192
LT_E	是否<(小于)的比较(PIN数可变)	202
<b>M</b>		
MAXIMUM(E)	从数据中选择最大值(PIN数可变)	188
MID(E)	从指定位置取得字符串	205
MINIMUM(E)	从数据中选择最小值(PIN数可变)	190
MOD(E)	数据的除法运算(输出余数)	166
MOVE(E)	数据的传送	170
MUL_TIME(E)	时间的乘法运算	225
MUL_E	数据的乘法运算(PIN数可变)	163
MUX(E)	选择数据后输出(PIN数可变)	194
<b>N</b>		
NE_E	是否≠(不等于)的比较	203
NOT(E)	逻辑非	183
<b>O</b>		
OR_E	逻辑或(PIN数可变)	180
<b>R</b>		
REAL_TO_DINT(E)	将单精度实数型数据转换为双字[带符号]型数据	84
REAL_TO_INT(E)	将单精度实数型数据转换为字[带符号]型数据	82
REAL_TO_STRING(E)	将单精度实数型数据转换为字符串型数据	86
REPLACE(E)	字符串的替换	215
R_TRIG(E)	检测出信号的上升沿后输出脉冲信号	230
<b>S</b>		
SEL(E)	根据输入条件选择数据	186
SET_BIT_OF_INT(E)	将值写入到字[带符号]型数据的指定位中	148
SHL(E)	左移	173
SHR(E)	右移	175
STR_TO_BOOL(E)	将字符串型数据转换为位型数据	111

函数名	功能	参考页
<b>S</b>		
STR_TO_DINT(E)	将字符串型数据转换为双字[带符号]型数据	113
STR_TO_INT(E)	将字符串型数据转换为字[带符号]型数据	111
STR_TO_REAL(E)	将字符串型数据转换为单精度实数型数据	115
STR_TO_TIME(E)	将字符串型数据转换为时间型数据	118
SUB_TIME(E)	时间的减法运算	223
SUB_E	数据的减法运算	161
<b>T</b>		
TIME_TO_BOOL(E)	将时间型数据转换为位型数据	124
TIME_TO_DINT(E)	将时间型数据转换为双字[带符号]型数据	128
TIME_TO_DWORD(E)	将时间型数据转换为双字[无符号]/位列[32位]型数据	134
TIME_TO_INT(E)	将时间型数据转换为字[带符号]型数据	126
TIME_TO_STRING(E)	将时间型数据转换为字符串型数据	130
TIME_TO_WORD(E)	将时间型数据转换为字[无符号]/位列[16位]型数据	132
TIMER_CON_T_FB_M	累计定时器的驱动	249
TIMER_10_FB_M	10ms定时器的驱动	248
TIMER_100_FB_M	100ms定时器的驱动	251
TOF(E)	输入信号变为OFF后,经过指定时间后将输出信号置为OFF	244
TON(E)	经过指定时间后将信号置为ON	242
TP(E)	在指定时间段将信号置为ON	240
<b>W</b>		
WORD_TO_BOOL(E)	将字[无符号]/位列[16位]型数据转换为位型数据	89
WORD_TO_DINT(E)	将字[无符号]/位列[16位]型数据转换为双字[带符号]型数据	95
WORD_TO_DWORD(E)	将字[无符号]/位列[16位]型数据转换为双字[无符号]/位列[32位]型数据	101
WORD_TO_INT(E)	将字[无符号]/位列[16位]型数据转换为字[带符号]型数据	93
WORD_TO_TIME(E)	将字[无符号]/位列[16位]型数据转换为时间型数据	105
<b>X</b>		
XOR_E	逻辑异或(PIN数可变)	181

操作符		
操作符名	功能	参考页
<b>记号</b>		
+	数据的加法运算 (PIN数可变)	253
-	数据的减法运算	254
*	数据的乘法运算 (PIN数可变)	255
/	数据的除法运算 (输出商)	256
**	数据的幂乘	258
&	逻辑与 (PIN数可变)	259
>	数据比较 (是否大于) (PIN数可变)	264
≥	数据比较 (是否大于或等于) (PIN数可变)	265
=	数据比较 (是否等于)	266
≤	数据比较 (是否小于或等于) (PIN数可变)	267
<	数据比较 (是否小于) (PIN数可变)	268
<>	数据比较 (是否不等于)	269
<b>A</b>		
ADD	数据的加法运算 (PIN数可变)	253
AND	逻辑与 (PIN数可变)	259
<b>D</b>		
DIV	数据的除法运算 (输出商)	256
<b>E</b>		
EQ	数据比较 (是否等于)	266
<b>G</b>		
GE	数据比较 (是否大于或等于) (PIN数可变)	265
GT	数据比较 (是否大于) (PIN数可变)	264
<b>L</b>		
LE	数据比较 (是否小于或等于) (PIN数可变)	267
LT	数据比较 (是否小于) (PIN数可变)	268
<b>M</b>		
MOD	数据的除法运算 (输出余数)	257
MUL	数据的乘法运算 (PIN数可变)	255
<b>N</b>		
NE	数据比较 (是否不等于)	269
NOT	逻辑非	263
<b>O</b>		
OR	逻辑或 (PIN数可变)	260
<b>S</b>		
SUB	数据的减法运算	254
<b>X</b>		
XOR	逻辑异或 (PIN数可变)	261

11  
应用函数  
(比较函数)

12  
应用函数  
(字符串函数)

13  
应用函数  
(时间数据类型函数)

14  
功能模块

15  
操作符

A  
软元件和  
地址的对应

B  
函数、  
操作符  
一览

## MEMO



## 关于保证

在使用时，请务必确认一下以下的有关产品保证方面的内容。

### 1. 免费保修期和免费保修范围

在产品的免费保修期内，如是由于本公司的原因导致产品发生故障和不良(以下统称为故障)时，用户可以通过当初购买的代理店或是本公司的服务网络，提出要求免费维修。

但是、如果要求去海外出差进行维修时，会收取派遣技术人员所需的实际费用。

此外，由于更换故障模块而产生的现场的重新调试、试运行等情况皆不属于本公司责任范围。

#### 【免费保修期】

产品的免费保修期为用户买入后或是投入到指定的场所后的12个月以内。但是，由于本公司的产品出厂后一般的流通时间最长为6个月，所以从制造日期开始算起的18个月为免费保修期的上限。此外，维修品的免费保修期不得超过维修前的保证时间而变得更长。

#### 【免费保修范围】

- (1) 只限于使用状态、使用方法以及使用环境等都遵照使用说明书、用户手册、产品上的注意事项等中记载的条件、注意事项等，在正常的状态下使用的情况。
- (2) 即使是在免费保修期内，但是如果属于下列的情况的话就变成收费的维修。
  - ① 由于用户的保管和使用不当、不注意、过失等等引起的故障以及用户的硬件或是软件设计不当引起的故障。
  - ② 由于用户擅自改动产品而引起的故障。
  - ③ 将本公司产品装入用户的设备中使用时，如果根据用户设备所受的法规规定设置了安全装置或是行业公认应该配备的功能构造等情况下，视为应该可以避免的故障。
  - ④ 通过正常维护·更换使用说明书等中记载的易耗品(电池、背光灯、保险丝等)可以预防的故障。
  - ⑤ 即使按照正常的使用方法，但是继电器触点或是触点到寿命的情况。
  - ⑥ 由于火灾、电压不正常等不可抗力导致的外部原因，以及地震、雷电、洪水灾害等天灾引起的故障。
  - ⑦ 在本公司产品出厂时的科学技术水平下不能预见的原因引起的故障。
  - ⑧ 其他、认为非公司责任而引起的故障。

### 2. 停产后的收费保修期

- (1) 本公司接受的收费维修品为产品停产后的7年内。有关停产的信息，都公布在本公司的技术新闻等中。
- (2) 不提供停产后的产品(包括附属品)。

### 3. 在海外的服务

对于海外的用户，本公司的各个地域的海外FA中心都接收维修。但是，各地的FA中心所具备的维修条件有所不同，望用户谅解。

### 4. 机会损失和间接损失不在质保责任范围内

无论是否在免费质保期内，凡以下事由三菱电机不承担责任。

- (1) 任何非三菱电机责任原因而导致的损失。
- (2) 因三菱电机产品故障而引起的用户机会损失、利润损失。
- (3) 无论三菱电机能否预测，由特殊原因而导致的损失和间接损失、事故赔偿、以及三菱电机产品以外的损伤。
- (4) 对于用户更换设备、现场机械设备的再调试、运行测试及其它作业等的补偿。

### 5. 产品规格的变更

产品样本、手册或技术资料中所记载的规格有时会未经通知就变更，还望用户能够预先询问了解。

### 6. 关于产品的适用范围

- (1) 使用本公司MELSEC微型可编程控制器时，要考虑到万一可编程控制器出现故障·不良等情况时也不会导致重大事故的使用用途，以及在出现故障·不良时起到作用。将以上这些作为条件加以考虑。在设备外部系统地做好后备或是安全功能。
- (2) 本公司的可编程控制器是针对普通的工业用途而设计和制造的产品。因此，在各电力公司的原子能发电站以及用于其他发电站等对公众有很大影响的用途中，以及用于各铁路公司以及政府部门等要求特别的质量保证体系的用途中时，不适合使用可编程控制器。此外，对于航空、医疗、燃烧、燃料装置、人工搬运装置、娱乐设备、安全机械等预计会对人身生命和财产产生重大影响的用途，也不适用可编程控制器。但是，即使是上述的用途，用户只要事先与本公司的营业窗口联系，并认可在其特定的用途下可以不要求特别的质量时，还是可以通过交换必须的资料后，选用可编程控制器的。

## 改订的历史记录

制作日期	版本	内容
2016年6月	A	制作初版



# 三菱微型可编程控制器

**FXCPU**

**结构化编程手册**

**应用函数篇**

**mitsubishi electric corporation**

HEAD OFFICE: TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN