

mitsubishi

三菱可编程控制器

MELSEC **Q** 系列

MELSEC *L* 系列

MELSEC-Q/L结构体
编程手册

应用函数篇

QSERIES
L SERIES

● 安全注意事项 ●

(使用之前请务必阅读)

在使用 MELSEC-Q 系列、MELSEC-L 系列可编程控制器之前，应仔细阅读各产品附带的手册及附带手册中所介绍的关联手册，同时在充分注意安全的前提下正确地操作。

请妥善保管产品附带手册，放置于操作人员易于取阅的地方，并应将本手册交给最终用户。

● 关于产品的应用 ●

- (1) 在使用三菱可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效保险功能。
- (2) 三菱可编程控制器是以一般工业用途等为对象设计和制造的通用产品。因此，三菱可编程控制器不应用于以下设备·系统等特殊用途。如果用于以下特殊用途，对于三菱可编程控制器的质量、性能、安全等所有相关责任（包括但不限于债务未履行责任、瑕疵担保责任、质量保证责任、违法行为责任、制造物责任），三菱将不负责。
 - 面向各电力公司的核电站以及其它发电厂等对公众有较大影响的用途。
 - 用于各铁路公司或公用设施目的等有特殊质量保证体系要求的用途。
 - 航空航天、医疗、铁路、焚烧·燃料装置、载人移动设备、载人运输装置、娱乐设备、安全设备等预计对人身财产有较大影响的用途。然而，对于上述应用，如果在限定于具体用途，无需特殊质量（超出一般规格的质量等）要求的条件下，经过三菱的判断也可以使用三菱可编程控制器，详细情况请与当地三菱代表机构协商。

修订记录

※本手册号在封底的左下角。

印刷日期	※手册编号	修改内容
2010 年 4 月	SH(NA)-080905CHN-A	第一版

日文手册原稿： SH-080737-E

本手册不授予任何工业产权或任何其它类型的产权，也不授予任何专利许可。三菱电机对由于使用了本手册中的内容而引起的涉及工业知识产权的任何问题不承担责任。

© 2010 三菱电机

前言

在此感谢贵方购买了三菱 MELSEC-Q、MELSEC-L 系列通用可编程控制器。
在使用之前应熟读本书，在充分了解 MELSEC 系列可编程控制器的功能・性能的基础上正确地使用本产品。
此外，应将本手册交给最终用户。

目录

安全注意事项	A - 1
关于产品的应用	A - 2
修订记录	A - 3
前言	A - 4
目录	A - 4
关于手册	A - 7

1 概要 1 - 1 ~ 1 - 4

1.1 本手册的定位	1 - 2
1.2 本手册中使用的总称・略称	1 - 4

2 函数列表 2 - 1 ~ 2 - 8

2.1 函数列表的阅读方法	2 - 2
2.2 函数列表	2 - 3
2.2.1 类型转换函数	2 - 3
2.2.2 单数值变量函数	2 - 4
2.2.3 算术运算函数	2 - 5
2.2.4 位型布尔函数	2 - 5
2.2.5 选择函数	2 - 5
2.2.6 比较函数	2 - 6
2.2.7 字符串函数	2 - 6
2.2.8 时间数据类型函数	2 - 6
2.2.9 双稳功能块	2 - 6
2.2.10 边沿检测功能块	2 - 7
2.2.11 计数器功能块	2 - 7
2.2.12 定时器功能块	2 - 7

3 函数的构成 3 - 1 ~ 3 - 4

3.1 函数的构成	3 - 2
3.2 输入针数可变函数	3 - 3

4 函数阅读方法 4 - 1 ~ 4 - 4

5 应用函数 5 - 1 ~ 5 - 198

5.1 类型转换函数	5 - 2
5.1.1 位型→字 [带符号] 型、双字 [带符号] 型转换	5 - 2
5.1.2 位型→字符串型转换	5 - 5
5.1.3 位型→字 [无符号] / 位串 [16 位] 型、双字 [无符号] / 位串 [32 位] 型转换	5 - 7

5.1.4	位型→时间型转换	5 - 10
5.1.5	字[带符号]型→双字[带符号]型转换	5 - 12
5.1.6	双字[带符号]型→字[带符号]型转换	5 - 14
5.1.7	字[带符号]型、双字[带符号]型→位型转换	5 - 16
5.1.8	字[带符号]型、双字[带符号]型→单精度实数型转换	5 - 19
5.1.9	字[带符号]型、双字[带符号]型→双精度实数型转换	5 - 22
5.1.10	字[带符号]型、双字[带符号]型→字符串型转换	5 - 25
5.1.11	字[带符号]型、双字[带符号]型→字[无符号]/位串[16位]型转换	5 - 29
5.1.12	字[带符号]型、双字[带符号]型→双字[无符号]/位串[32位]型转换	5 - 32
5.1.13	字[带符号]型、双字[带符号]型→BCD型转换	5 - 35
5.1.14	字[带符号]型、双字[带符号]型→时间型转换	5 - 38
5.1.15	单精度实数型→字[带符号]型、双字[带符号]型转换	5 - 41
5.1.16	双精度实数型→字[带符号]型、双字[带符号]型转换	5 - 44
5.1.17	单精度实数型→双精度实数型转换	5 - 47
5.1.18	双精度实数型→单精度实数型转换	5 - 49
5.1.19	单精度实数型→字符串型转换	5 - 51
5.1.20	字[无符号]/位串[16位]型、双字[无符号]/位串[32位]型→位型转换	5 - 55
5.1.21	字[无符号]/位串[16位]型→字[带符号]型、双字[带符号]型转换	5 - 58
5.1.22	双字[无符号]/位串[32位]型→字[带符号]型、双字[带符号]型转换	5 - 61
5.1.23	字[无符号]/位串[16位]型→双字[无符号]/位串[32位]型转换	5 - 64
5.1.24	双字[无符号]/位串[32位]型→字[无符号]/位串[16位]型转换	5 - 66
5.1.25	字[无符号]/位串[16位]型、双字[无符号]/位串[32位]型→字符串型转换	5 - 69
5.1.26	字[无符号]/位串[16位]型、双字[无符号]/位串[32位]型→时间型转换	5 - 72
5.1.27	字符串型→位型转换	5 - 75
5.1.28	字符串型→字[带符号]型、双字[带符号]型转换	5 - 77
5.1.29	字符串型→单精度实数型转换	5 - 80
5.1.30	字符串型→字[无符号]/位串[16位]型、双字[无符号]/位串[32位]型转换	5 - 84
5.1.31	字符串型→时间型转换	5 - 87
5.1.32	字符串型→BCD型转换	5 - 89
5.1.33	BCD型→字[带符号]型、双字[带符号]型转换	5 - 92
5.1.34	BCD型→字符串型转换	5 - 95
5.1.35	时间型→位型转换	5 - 97
5.1.36	时间型→字[带符号]型、双字[带符号]型转换	5 - 99
5.1.37	时间型→字符串型转换	5 - 102
5.1.38	时间型→字[无符号]/位串[16位]型、双字[无符号]/位串[32位]型转换	5 - 104
5.2 单数值变量函数		5 - 107
5.2.1	绝对值	5 - 107
5.3 算术运算函数		5 - 109
5.3.1	加法运算	5 - 109
5.3.2	乘法运算	5 - 112
5.3.3	减法运算	5 - 115
5.3.4	除法运算	5 - 118
5.3.5	余数	5 - 121
5.3.6	幂	5 - 124
5.3.7	代入	5 - 126
5.4 位型布尔函数		5 - 128
5.4.1	逻辑积、逻辑和、排他的逻辑和、逻辑否定	5 - 128
5.5 选择函数		5 - 133
5.5.1	选择值	5 - 133
5.5.2	最大值、最小值选择	5 - 136
5.5.3	上下限极限控制	5 - 139

5.5.4	多路复用器	5 - 142
5.6	比较函数	5 - 145
5.6.1	比较	5 - 145
5.7	字符串函数	5 - 148
5.7.1	字符串的截取	5 - 148
5.7.2	字符串的合并	5 - 151
5.7.3	字符串的插入	5 - 153
5.7.4	字符串的删除	5 - 156
5.7.5	字符串的置换	5 - 159
5.8	时间数据类型函数	5 - 162
5.8.1	加法运算	5 - 162
5.8.2	减法运算	5 - 164
5.8.3	乘法运算	5 - 166
5.8.4	除法运算	5 - 168
5.9	双稳功能块	5 - 170
5.9.1	双稳功能块 (设置优先)	5 - 170
5.9.2	双稳功能块 (复位优先)	5 - 172
5.10	边沿检测功能块	5 - 174
5.10.1	上升沿检测	5 - 174
5.10.2	下降沿检测	5 - 176
5.11	计数器功能块	5 - 178
5.11.1	递增计数器	5 - 178
5.11.2	递减计数器	5 - 180
5.11.3	递增递减计数器	5 - 182
5.11.4	计数器功能块	5 - 185
5.12	定时器功能块	5 - 187
5.12.1	脉冲定时器	5 - 187
5.12.2	ON 延迟定时器	5 - 190
5.12.3	OFF 延迟定时器	5 - 192
5.12.4	定时器功能块	5 - 194

关于手册

关联手册

与本产品有关的手册如下所示。

请根据需要参考本表订购。

(1) 结构体编程

手册名称	手册编号
MELSEC-Q/L/F 结构体编程手册（基础篇） 对结构体程序创建中必要的编程方法、编程语言的种类等有关内容进行说明。 (另售)	SH-080903CHN
MELSEC-Q/L 结构体编程手册（公共指令篇） 对结构体程序中可使用的顺控指令、基本指令以及应用指令等的公共指令相关的规格、功能等有关内容进行说明。 (另售)	SH-080904CHN
MELSEC-Q/L 结构体编程手册（特殊指令篇） 对结构体程序中可使用的模块专用指令、PID 控制指令以及内置 I/O 功能用指令等的特殊指令相关的规格、功能等有关内容进行说明。 (另售)	SH-080906CHN

(2) GX Works2 的操作

手册名称	手册编号
GX Works2 Version1 操作手册（公共篇） 对 GX Works2 的系统配置及参数设置、在线功能的操作方法等、简易工程及结构体工程的通用功能等有关内容进行说明。 (另售)	SH-080932CHN
GX Works2 Version1 操作手册（结构体工程篇） 对 GX Works2 的结构体工程中的程序创建、监视等的操作方法等有关内容进行说明。 (另售)	SH-080934CHN
GX Works2 入门指南（结构体工程篇） 对初次使用 GX Works2 的用户介绍结构体工程中的程序创建及编辑、监视、调试的基本操作方法等有关内容进行说明。 (另售)	SH-080936CHN

☒ 要点

各操作手册以 PDF 文件被存储在软件包的 CD-ROM 中。备有用于另售的印刷品，希望单独购买手册的情况下，请通过上述表格中的手册编号购买。



概要

1.1	本手册的定位	1-2
1.2	本手册中使用的总称・略称	1-4









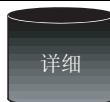



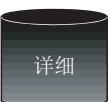
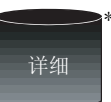
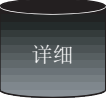






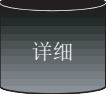

1.1 本手册的定位

在本手册中，对结构体程序创建中使用的应用函数的有关内容进行说明。

以目的进行分类的参阅手册如下所示。









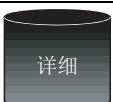




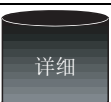
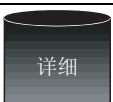
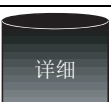
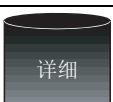
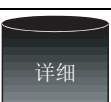
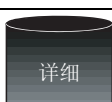
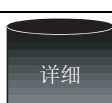
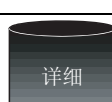
关于各手册的记载内容、手册编号等请参阅“关联手册”的列表。

(1) GX Works2 的操作

目的		GX Works2 安装步骤 说明书	GX Works2 入门		GX Works2 Version1 操作手册			
								
		-	简易工程篇	结构体工程篇	公共篇	简易工程篇	结构体工程篇	智能功能模块 操作篇
安装	希望了解运行环境、安装方法	 详细						
简易工程的操作	希望了解基本操作及步骤		 详细		 概要	 概要		
	希望了解编程用的功能及操作方法				 概要	 详细	 详细 ^{*1}	
	希望了解除编程以外的所有功能及操作方法				 详细			
结构体工程的操作	希望了解基本操作及步骤			 详细	 概要		 概要	
	希望了解编程用的功能及操作方法				 概要	 详细	 详细	
	希望了解除编程以外的所有功能及操作方法				 详细			
智能功能模块的操作	希望了解智能功能模块的数据设置方法							 详细

*1 : 仅 ST 程序。

(2) 编程

目的		MELSEC-Q/L/F 结构体编程 手册	MELSEC-Q/L 结构体编程手册			MELSEC-Q/L 编程手册	MELSEC-Q/L/ QnA 编程手册	智能功能模块 用户手册 / 网络模块 参考手册
								
		基础篇	公共指令篇	特殊指令篇	应用函数篇	公共指令篇	PID 控制 指令篇	-
简易工程中的 编程	希望了解公共指令 的种类及详细内容、 出错代码、特殊继 电器・特殊寄存器 的内容					 详细		
	希望了解智能功能 模块用指令的种类 及详细内容						 详细	
	希望了解网络模块 用指令的种类及详 细内容						 详细	
	希望了解 PID 控制 用指令的种类及详 细内容						 详细	
结构体工程的 编程	希望了解初次进行 结构体编程的基础 知识	 详细						
	希望了解公共指令 的种类及详细内容		 详细					
	希望了解智能功能 模块用指令的种类 及详细内容			 详细				 详细
	希望了解网络模块 用指令的种类及详 细内容			 详细				 详细
	希望了解 PID 控制 用指令的种类及详 细内容			 详细			 详细	
	希望了解出错代 码、特殊继电器・ 特殊寄存器的内容					 详细		
	希望了解应用函数 的种类及详细内容				 详细			

1.2 本手册中使用的总称·略称

在本手册中，将软件包、可编程控制器 CPU 等以如下所示的总称·略称表示。在需要标明相关型号的情况下，将记载模块型号。

总称 / 略称	总称·略称的内容
GX Works2	产品型号 SWnDNC-GXW2 的总称产品名。 (n= 版本)
基本型 QCPU	Q00J、Q00、Q01 的总称。
高性能型 QCPU	Q02、Q02H、Q06H、Q12H、Q25H 的总称。
通用型 QCPU	Q00UJ、Q00U、Q01U、Q02U、Q03UD、Q03UDE、Q04UDH、Q04UDEH、Q06UDH、Q06UDEH、Q10UDH、Q10UDEH、Q13UDH、Q13UDEH、Q20UDH、Q20UDEH、Q26UDH、Q26UDEH 的总称。
QCPU(Q 模式)	基本型 QCPU、高性能型 QCPU、通用型 QCPU 的总称。
LCPU	L02、L26-BT 的总称。
CPU 模块	QCPU(Q 模式)、LCPU 的总称。
个人计算机	基于 Windows® 的个人计算机的总称。
公共指令	顺控程序指令、基本指令、应用指令、数据链接指令、多 CPU 专用指令、多 CPU 高速通信专用指令的总称。
特殊指令	模块专用指令、PID 控制指令、Socket(套接字)通信功能用指令、内置 I/O 功能用指令、数据记录功能用指令的总称。

2

函数列表

2.1	函数列表的阅读方法	2-2
2.2	函数列表	2-3

2.1 函数列表的阅读方法

2.2 节的函数列表的格式如下所示。

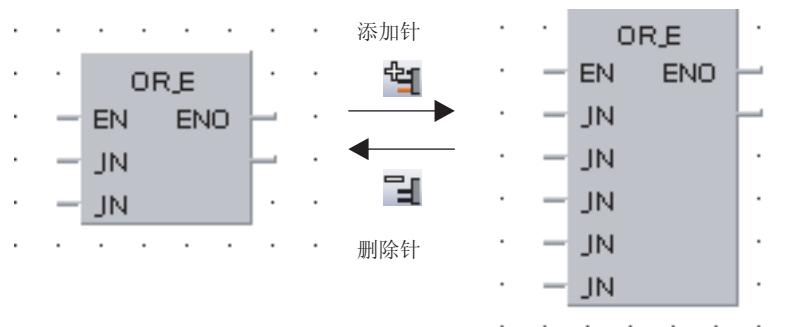
函数名	自变量	处理内容	说明 页面
ADD(_E)	①, ②, ... ②③ ④ (针数可变)	对输入值的和 (①)+②+...+②③ 进行输出。	5-118
OR(_E)	①, ②, ... ②③ ④ (针数可变)	对输入值的积 (①)×②×...×②③ 进行输出。	5-121
SUB(_E)	①, ②, ④	对输入值的差 (①)-② 进行输出。	5-124
DIV(_E)	①, ②, ④	对输入值的商 (①)÷② 进行输出。	5-127

↑ ①
↑ ②
↑ ③
↑ ④

说明

- ①..... 表示程序中使用的函数名。
“函数名 (_E)” 作为带 EN 函数使用。
- ②..... 表示函数的自变量。
- ⑤: 源 存储运算前的数据。
- ④: 目标 表示运算后的数据去向。
(针数可变) 可以将⑤(源)的个数在 2 ~ 28 的范围内进行变更。

对针数进行变更的情况下



- ③..... 表示各函数的处理内容。
- ④..... 表示说明各函数的页面。

2.2 函数列表

2.2.1 类型转换函数

函数名	自变量	处理内容	说明页面
BOOL_TO_INT(_E)	(s, d)	将位型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-2
BOOL_TO_DINT(_E)	(s, d)		
BOOL_TO_STR(_E)	(s, d)	将位型数据转换为字符串型数据。	5-5
BOOL_TO_WORD(_E)	(s, d)	将位型数据转换为字 [无符号]/位串 [16 位] 型、双字 [无符号]/位串 [32 位] 型数据。	5-7
BOOL_TO_DWORD(_E)	(s, d)		
BOOL_TO_TIME(_E)	(s, d)	将位型数据转换为时间型数据。	5-10
INT_TO_DINT(_E)	(s, d)	将字 [带符号] 型数据转换为双字 [带符号] 型数据。	5-12
DINT_TO_INT(_E)	(s, d)	将双字 [带符号] 型数据转换为字 [带符号] 型数据。	5-14
INT_TO_BOOL(_E)	(s, d)	将字 [带符号] 型、双字 [带符号] 型数据转换为位型数据。	5-16
DINT_TO_BOOL(_E)	(s, d)		
INT_TO_REAL(_E)	(s, d)	将字 [带符号] 型、双字 [带符号] 型数据转换为单精度实数型数据。	5-19
DINT_TO_REAL(_E)	(s, d)		
INT_TO_LREAL(_E)	(s, d)	将字 [带符号] 型、双字 [带符号] 型数据转换为双精度实数型数据。	5-22
DINT_TO_LREAL(_E)	(s, d)		
INT_TO_STR(_E)	(s, d)	将字 [带符号] 型、双字 [带符号] 型数据转换为字符串型数据。	5-25
DINT_TO_STR(_E)	(s, d)		
INT_TO_WORD(_E)	(s, d)	将字 [带符号] 型、双字 [带符号] 型数据转换为字 [无符号]/位串 [16 位] 型数据。	5-29
DINT_TO_WORD(_E)	(s, d)		
INT_TO_DWORD(_E)	(s, d)	将字 [带符号] 型、双字 [带符号] 型数据转换为双字 [无符号]/位串 [32 位] 型数据。	5-32
DINT_TO_DWORD(_E)	(s, d)		
INT_TO_BCD(_E)	(s, d)	将字 [带符号] 型、双字 [带符号] 型数据转换为 BCD 型数据。	5-35
DINT_TO_BCD(_E)	(s, d)		
INT_TO_TIME(_E)	(s, d)	将字 [带符号] 型数据转换为时间型数据。	5-38
DINT_TO_TIME(_E)	(s, d)		
REAL_TO_INT(_E)	(s, d)	将单精度实数型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-41
REAL_TO_DINT(_E)	(s, d)		
LREAL_TO_INT(_E)	(s, d)	将双精度实数型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-44
LREAL_TO_DINT(_E)	(s, d)		
REAL_TO_LREAL(_E)	(s, d)	将单精度实数型数据转换为双精度实数型数据。	5-47
LREAL_TO_REAL(_E)	(s, d)	将双精度实数型数据转换为单精度实数型数据。	5-49
REAL_TO_STR(_E)	(s, d)	将单精度实数型数据转换为字符串型 (指数格式) 数据。	5-51
WORD_TO_BOOL(_E)	(s, d)	将字 [无符号]/位串 [16 位] 型、双字 [无符号]/位串 [32 位] 型数据转换为位型数据。	5-55
DWORD_TO_BOOL(_E)	(s, d)		
WORD_TO_INT(_E)	(s, d)	将字 [无符号]/位串 [16 位] 型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-58
WORD_TO_DINT(_E)	(s, d)		
DWORD_TO_INT(_E)	(s, d)	将双字 [无符号]/位串 [32 位] 型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-61
DWORD_TO_DINT(_E)	(s, d)		
WORD_TO_DWORD(_E)	(s, d)	将字 [无符号]/位串 [16 位] 型数据转换为双字 [无符号]/位串 [32 位] 型数据。	5-64

函数名	自变量	处理内容	说明页面
DWORD_TO_WORD(_E)	⑤, ④	将双字 [无符号]/ 位串 [32 位] 型数据转换为字 [无符号]/ 位串 [16 位] 型数据。	5-66
WORD_TO_STR(_E)	⑤, ④	将字 [无符号]/ 位串 [16 位] 型、双字 [无符号]/ 位串 [32 位] 型数据转换为字符串型数据。	5-69
DWORD_TO_STR(_E)	⑤, ④		
WORD_TO_TIME(_E)	⑤, ④	将字 [无符号]/ 位串 [16 位] 型、双字 [无符号]/ 位串 [32 位] 型数据转换为时间型数据。	5-72
DWORD_TO_TIME(_E)	⑤, ④		
STR_TO_BOOL(_E)	⑤, ④	将字符串型数据转换为位型数据。	5-75
STR_TO_INT(_E)	⑤, ④	将字符串型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-77
STR_TO_DINT(_E)	⑤, ④		
STR_TO_REAL(_E)	⑤, ④	将字符串型数据转换为单精度实数型数据。	5-80
STR_TO_WORD(_E)	⑤, ④	将字符串型数据转换为字 [无符号]/ 位串 [16 位] 型、双字 [无符号]/ 位串 [32 位] 型数据。	5-84
STR_TO_DWORD(_E)	⑤, ④		
STR_TO_TIME(_E)	⑤, ④	将字符串型数据转换为时间型数据。	5-87
STR_TO_BCD(_E)	⑤, ④	将字符串型数据转换为 BCD 型数据。	5-89
BCD_TO_INT(_E)	⑤, ④	将 BCD 型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-92
BCD_TO_DINT(_E)	⑤, ④		
BCD_TO_STR(_E)	⑤, ④	将 BCD 型数据转换为字符串型数据。	5-95
TIME_TO_BOOL(_E)	⑤, ④	将时间型数据转换为位型数据。	5-97
TIME_TO_INT(_E)	⑤, ④	将时间型数据转换为字 [带符号] 型、双字 [带符号] 型数据。	5-99
TIME_TO_DINT(_E)	⑤, ④		
TIME_TO_STR(_E)	⑤, ④	将时间型数据转换为字符串型数据。	5-102
TIME_TO_WORD(_E)	⑤, ④	将时间型数据转换为字 [无符号]/ 位串 [16 位] 型、双字 [无符号]/ 位串 [32 位] 型数据。	5-104
TIME_TO_DWORD(_E)	⑤, ④		

2.2.2 单数值变量函数

函数名	自变量	处理内容	说明页面
ABS(_E)	⑤, ④	对输入值的绝对值进行输出。	5-107

2.2.3 算术运算函数

函数名	自变量	处理内容	说明页面
ADD(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对输入值的和 ($s_1+s_2+\dots+s_n$) 进行输出。	5-109
MUL(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对输入值的积 ($s_1 \times s_2 \times \dots \times s_n$) 进行输出。	5-112
SUB(_E)	s_1, s_2, d	对输入值的差 (s_1-s_2) 进行输出。	5-115
DIV(_E)	s_1, s_2, d	对输入值的商 ($s_1 \div s_2$) 进行输出。	5-118
MOD(_E)	s_1, s_2, d	对输入值的余数 ($s_1 \div s_2$) 进行输出。	5-121
EXPT(_E)	s_1, s_2, d	对输入值的幂进行输出。	5-124
MOVE(_E)	s, d	对输入值的代入进行输出。	5-126

2.2.4 位型布尔函数

函数名	自变量	处理内容	说明页面
AND(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对输入值的逻辑积进行输出。	5-128
OR(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对输入值的逻辑和进行输出。	
XOR(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对输入值的排他的逻辑和进行输出。	
NOT(_E)	s, d	对输入值的逻辑否定进行输出。	

2.2.5 选择函数

函数名	自变量	处理内容	说明页面
SEL(_E)	s_1, s_2, s_3, d	对选择的输入值进行输出。	5-133
MAXIMUM(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对输入值的最大值进行输出。	5-136
MINIMUM(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对输入值的最小值进行输出。	
LIMITATION(_E)	s_1, s_2, s_3, d	对上下极限控制的输入值进行输出。	5-139
MUX(_E)	s_1, s_2, \dots, s_n, d (针数可变)	对多个输入值中的一个进行输出。	5-142

2.2.6 比较函数

函数名	自变量	处理内容	说明页面
GT(_E)	①, ②, ... ③, ④ (针数可变)	对输入值的数据比较结果进行输出。	5-145
GE(_E)	①, ②, ... ③, ④ (针数可变)		
EQ(_E)	①, ②, ... ③, ④ (针数可变)		
LE(_E)	①, ②, ... ③, ④ (针数可变)		
LT(_E)	①, ②, ... ③, ④ (针数可变)		
NE(_E)	①, ②, ④		

2.2.7 字符串函数

函数名	自变量	处理内容	说明页面
MID(_E)	①, ②, ③, ④	从输入的字符串的任意位置对指定字符进行输出。	5-148
CONCAT(_E)	①, ②, ④	将2个字符串合并后进行输出。	5-151
INSERT(_E)	①, ②, ③, ④	在字符串之间插入字符串后进行输出。	5-153
DELETE(_E)	①, ②, ③, ④	将字符串的任意范围删除后进行输出。	5-156
REPLACE(_E)	①, ②, ③, ④, ⑤	将字符串的任意范围置换后进行输出。	5-159

2.2.8 时间数据类型函数

函数名	自变量	处理内容	说明页面
ADD_TIME(_E)	①, ②, ④	对输入值(时间型)的和(①+②)进行输出。	5-162
SUB_TIME(_E)	①, ②, ④	对输入值(时间型)的差(①-②)进行输出。	5-164
MUL_TIME(_E)	①, ②, ④	对输入值(时间型)的积(①×②)进行输出。	5-166
DIV_TIME(_E)	①, ②, ④	对输入值(时间型)的商(①÷②)进行输出。	5-168

2.2.9 双稳功能块

函数名	自变量	处理内容	说明页面
SR(_E)	①, ②, ④	对2个输入值进行判别, 对1(TRUE)或者0(FALSE)进行输出。	5-170
RS(_E)	①, ②, ④	对2个输入值进行判别, 对1(TRUE)或者0(FALSE)进行输出。	5-172

2.2.10 边沿检测功能块

函数名	自变量	处理内容	说明页面
R_TRIG(_E)	①, ④	检测出信号的上升沿后对脉冲信号进行输出。	5-174
F_TRIG(_E)	①, ④	检测出信号的下降沿后对脉冲信号进行输出。	5-176

2.2.11 计数器功能块

函数名	自变量	处理内容	说明页面
CTU(_E)	①, ②, ③, ④, ⑤	对信号的上升沿次数进行递增计数。	5-178
CTD(_E)	①, ②, ③, ④, ⑤	对信号的上升沿次数进行递减计数。	5-180
CTUD(_E)	①, ②, ③, ④, ⑤, ⑥, ⑦, ⑧	对信号的上升沿次数进行递增计数 / 递减计数。	5-182
COUNTER_FB_M	①, ②, ③, ④, ⑤	对信号的上升沿次数从③开始至⑤为止进行递增计数。	5-185

2.2.12 定时器功能块

函数名	自变量	处理内容	说明页面
TP(_E) TP_HIGH(_E)	①, ②, ④, ⑤	在指定的时间期间将信号置为 ON。	5-187
TON(_E) TON_HIGH(_E)	①, ②, ③, ④, ⑤	在指定时间后将信号置为 ON。	5-190
TOF(_E) TOF_HIGH(_E)	①, ②, ③, ④, ⑤, ⑥, ⑦, ⑧	在指定时间后将信号置为 OFF。	5-192
TIMER_10_FB_M TIMER_100_FB_M TIMER_HIGH_FB_M TIMER_LOW_FB_M TIMER_CONT_FB_M TIMER_CONTHFB_M	①, ②, ③, ④, ⑤	从输入值⑤开始经过了②中指定的时间后将信号置为 ON。	5-194

3

函数的构成

3.1	函数的构成	3-2
3.2	输入针数可变函数	3-3

3.1 函数的构成

CPU 模块中可使用的指令，可以分为函数名及自变量。

函数名与自变量的用途如下所示。

- 函数名 表示该函数的功能。
- 自变量 表示函数中使用的输入输出数据。

自变量中，有源数据、目标数据、执行条件、执行状态。

(1) 源^㉓

(a) 源是在运算中使用的数据。

(b) 根据函数中指定的软元件的情况如下所示。

- 常数 对运算中使用的数值进行指定。
在创建程序时设置，在程序执行过程中不能变更。
将常数用于可变数据中的情况下，应进行变址修饰。
- 位软元件、字软元件 对存储运算中使用的数据的软元件进行指定。
在执行运算之前需要预先将数据存储到指定的软元件中。
在程序执行过程中，通过对指定的软元件中存储的数据进行变更，可以对该函数中使用的数据进行变更。

(c) 对于使用位软元件的源，不能直接输入触点。

(2) 目标^㉔

(a) 在目标中存储运算后的数据。

(b) 目标中必须设置用于存储数据的软元件。

(c) 对于存储位软元件的目标，不能直接连接线圈。

(3) 执行条件 (EN)

(a) 在输入变量 EN 中输入函数的执行条件。

(4) 执行状态 (ENO)

(a) 输出变量 ENO 用于对执行状态进行输出。

要 点

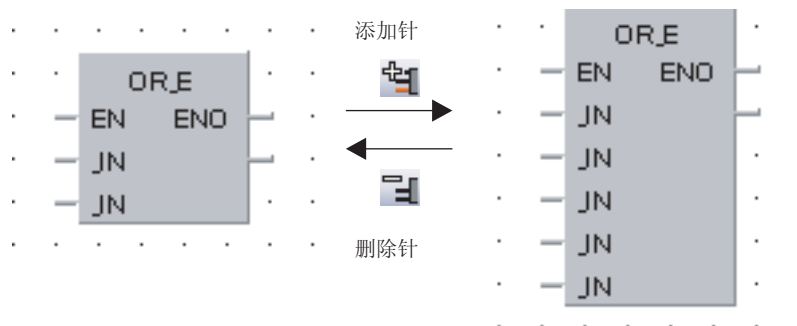
关于标签及结构体等函数的构成的详细内容，请参阅 MELSEC-Q/L/F 结构体编程手册（基础篇）。

3.2 输入针数可变函数

函数中有的函数的输入针数可以进行变更。

进行输入针数变更时，选择对象函数后对输入针数进行变更。

关于输入针数的变更操作，请参阅  GX Works2 Version1 操作手册（结构体工程篇）。



4

函数阅读方法

1

概要

2

函数列表

3

函数的构成

4

函数阅读方法

5

应用函数

索引

索引

从本手册的第 5 章以后，按以下构成对各函数进行说明。

BOOL_TO_INT(E), BOOL_TO_DINT(E)

5.1 类型转换函数

1) → 5.1.1 位型→字[带符号]型、双字[带符号]型转换

2) → BOOL_TO_INT(E), BOOL_TO_DINT(E)

3) →

4) → `BOOL_TO_INT(E)`
`BOOL_TO_DINT(E)` (`_E` : 带 EN/ENO)

5) →

6) →

7) → 输入自变量, EN: 执行条件 (TRUE : 执行 FALSE : 停止) : 位
 a: 输入 : 位
 输出自变量, ENO: 输出状态 (TRUE : 正常执行 FALSE : 异常或者停止) : 位
 d: 输出 : 字[带符号]、双字[带符号]

8) → ☆ 功能

运算处理

(1) BOOL_TO_INT, BOOL_TO_INT_E

将⊙中输入的位型数据转换为字[带符号]型数据后通过⊙进行输出。
 输入值为 FALSE 的情况下，以字[带符号]型数据值对 0 进行输出。
 输入值为 TRUE 的情况下，以字[带符号]型数据值对 1 进行输出。

⊙	→	⊙
FALSE		0
TRUE		1
位型		字[带符号]型

9) → ⚠ 出错

不存在 BOOL_TO_INT(E)、BOOL_TO_DINT(E) 相关的运算出错。

10) → 📄 程序示例

(1) 以下为将⊙中输入的位型数据转换为字[带符号]型数据后⊙进行输出的程序。

(a) 无 EN/ENO 的函数 (BOOL_TO_INT)













[结构体梯形图]

[ST]

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

- 1) 表示项编号、函数的概要。
- 2) 表示说明对象的函数。

3) 可使用函数的 CPU 模块如下所示。

图标				内容
通用型 QCPU	基本型 QCPU	高性能型 QCPU	L CPU	
				是普通图标，表示可使用相应的函数。
				带 Ver. 符号的图标表示可在有部分限制的情况下使用。(功能版本、软件版本)
				带 × 符号的图标表示不能使用相应函数。

4) 表示函数名。

5) 表示可记述的函数名。

6) 表示结构体梯形图、ST 语言中的函数记述格式。

7) 表示函数的输入自变量、输出自变量的名称、各自变量的数据类型以及软元件的直接指定可否。

关于各数据类型的详细内容，请参阅 MELSEC-Q/L/F 结构体编程手册（基础篇）。

8) 表示函数担当的功能。

9) 表示有无出错。有出错的情况下，表示引起出错的条件。

10) 结构体梯形图、ST 表示程序示例。

5

应用函数

5.1	类型转换函数	5-2
5.2	单数值变量函数	5-107
5.3	算术运算函数	5-109
5.4	位型布尔函数	5-128
5.5	选择函数	5-133
5.6	比较函数	5-145
5.7	字符串函数	5-148
5.8	时间数据类型函数	5-162
5.9	双稳功能块	5-170
5.10	边沿检测功能块	5-174
5.11	计数器功能块	5-178
5.12	定时器功能块	5-187

5.1 类型转换函数

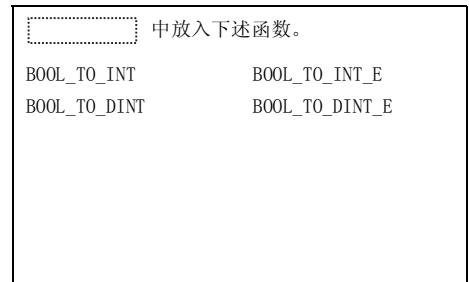
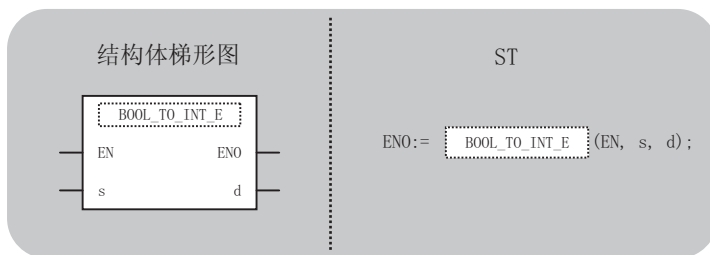
5.1.1 位型→字[带符号]型、双字[带符号]型转换

BOOL_TO_INT(_E), BOOL_TO_DINT(_E)

Basic high performance Universal L CPU

BOOL_TO_INT(_E)
BOOL_TO_DINT(_E)

(_E : 带 EN/ENO)



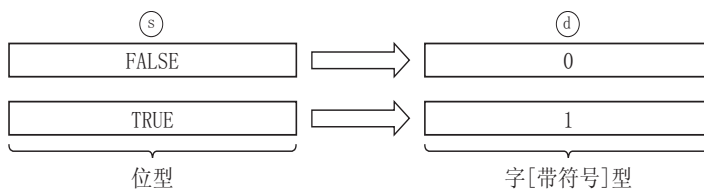
输入自变量, EN:	执行条件 (TRUE : 执行 FALSE : 停止)	: 位
s:	输入	: 位
输出自变量, ENO:	输出状态 (TRUE : 正常执行 FALSE : 异常或者停止)	: 位
d:	输出	: 字 [带符号]、双字 [带符号]

★ 功能

运算处理

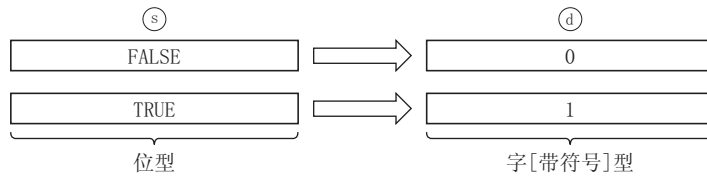
(1) BOOL_TO_INT, BOOL_TO_INT_E

将⑤中输入的位型数据转换为字[带符号]型数据后通过⑥进行输出。
输入值为 FALSE 的情况下, 以字[带符号]型数据值对 0 进行输出。
输入值为 TRUE 的情况下, 以字[带符号]型数据值对 1 进行输出。



(2) BOOL_TO_DINT, BOOL_TO_DINT_E

将⑤中输入的位型数据转换为双字[带符号]型数据后通过④进行输出。
 输入值为FALSE的情况下，以双字[带符号]型的数据值对0进行输出。
 输入值为TRUE的情况下，以双字[带符号]型的数据值对1进行输出。



运算结果

- (1) 无 EN/ENO 的函数
 执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
 执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下，④中输出的数据将变为不定值。
 在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

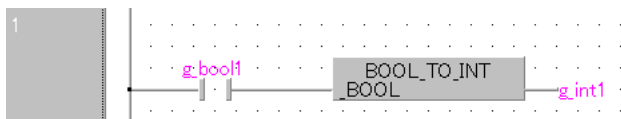
不存在 BOOL_TO_INT(_E)、BOOL_TO_DINT(_E) 相关的运算出错。

程序示例

(1) 以下为将⑤中输入的位型数据转换为字 [带符号] 型数据后④进行输出的程序。

(a) 无 EN/ENO 的函数 (BOOL_TO_INT)

[结构体梯形图]

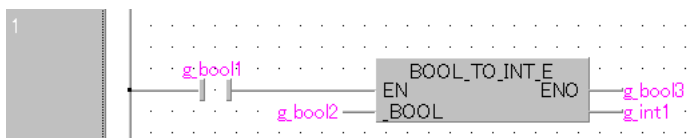


[ST]

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

(b) 带 EN/ENO 的函数 (BOOL_TO_INT_E)

[结构体梯形图]



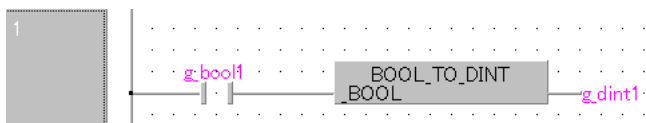
[ST]

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

(2) 以下为将⑤中输入的位型数据转换为双字 [带符号] 型数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (BOOL_TO_DINT)

[结构体梯形图]



[ST]

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

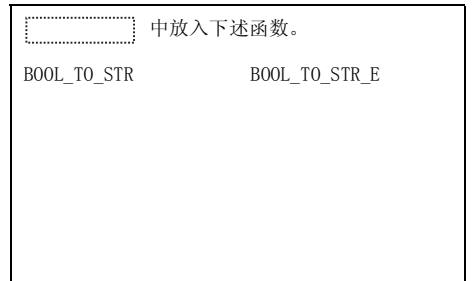
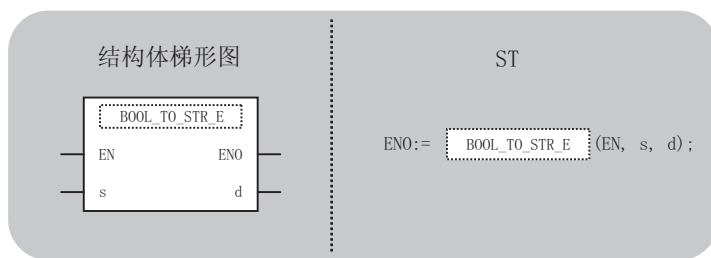
5.1.2 位型→字符串型转换

BOOL_TO_STR(_E)

Basic High performance Universal L CPU

BOOL_TO_STR(_E)

(_E : 带 EN/ENO)

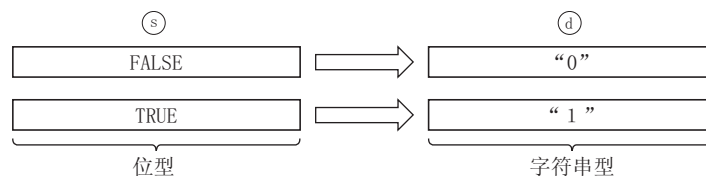


输入自变量, EN:	执行条件 (TRUE: 执行 FALSE: 停止)	: 位
s:	输入	: 位
输出自变量, ENO:	输出状态 (TRUE: 正常执行 FALSE: 异常或者停止)	: 位
d:	输出	: 字符串

★ 功能

运算处理

将⑤中输入的位型数据转换为字符串型数据后通过④进行输出。
 输入值为 FALSE 的情况下, 通过字符串型的数据值对“0”进行输出。
 输入值为 TRUE 的情况下, 通过字符串型的数据值对“1”进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

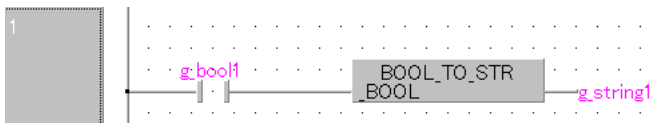
不存在 BOOL_TO_STR(_E) 相关的运算出错。

程序示例

以下为将⑤中输入的位型数据转换为字符串型数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (BOOL_TO_STR)

[结构体梯形图]



[ST]

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

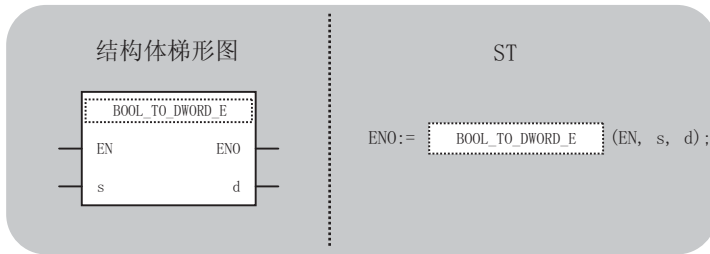
5.1.3 位型→字 [无符号] / 位串 [16 位] 型、双字 [无符号] / 位串 [32 位] 型转换

BOOL_TO_WORD(_E), BOOL_TO_DWORD(_E)

Basic High performance Universal L CPU

BOOL_TO_WORD(_E)
BOOL_TO_DWORD(_E)

(_E : 带 EN/ENO)



中放入下述函数。
 BOOL_TO_WORD BOOL_TO_WORD_E
 BOOL_TO_DWORD BOOL_TO_DWORD_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

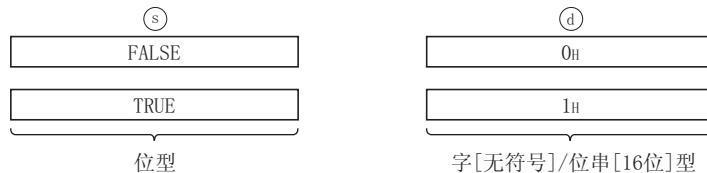
: 位
 : 位
 : 位
 : 字 [无符号] / 位串 [16 位]、双字 [无符号] / 位串 [32 位]

★ 功能

运算处理

(1) BOOL_TO_WORD, BOOL_TO_WORD_E

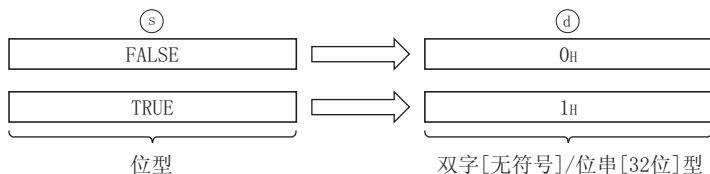
将⑤中输入的位型数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过⑥进行输出。输入值为 FALSE 的情况下, 以字 [无符号] / 位串 [16 位] 型的数据值对 0H 进行输出。输入值为 TRUE 的情况下, 以字 [无符号] / 位串 [16 位] 型的数据值对 1H 进行输出。



5
应用函数
BOOL_TO_WORD(_E),
BOOL_TO_DWORD(_E)

(2) BOOL_TO_DWORD, BOOL_TO_DWORD_E

将⑤中输入的位型数据转换为双字 [无符号] / 位串 [32 位] 型的数据后通过④进行输出。
 输入值为 FALSE 的情况下，以双字 [无符号] / 位串 [32 位] 型的数据值对 0H 进行输出。
 输入值为 TRUE 的情况下，以双字 [无符号] / 位串 [32 位] 型的数据值对 1H 进行输出。



运算结果

- (1) 无 EN/ENO 的函数
 执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
 执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
 在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

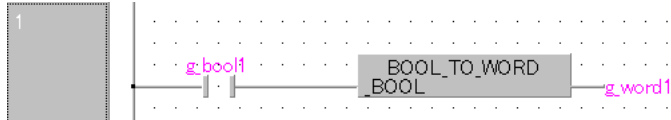
不存在 BOOL_TO_WORD(_E)、BOOL_TO_DWORD(_E) 相关的运算出错。

程序示例

- (1) 以下为将 ⑤ 中输入的位型数据转换为字 [无符号]/ 位串 [16 位] 型的数据后通过 ④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (BOOL_TO_WORD)

[结构体梯形图]

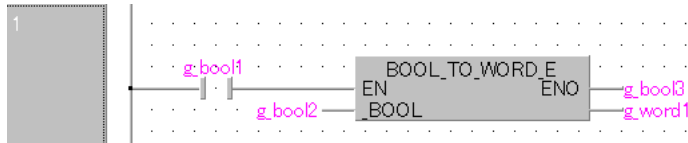


[ST]

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

(b) 带 EN/ENO 的函数 (BOOL_TO_WORD_E)

[结构体梯形图]



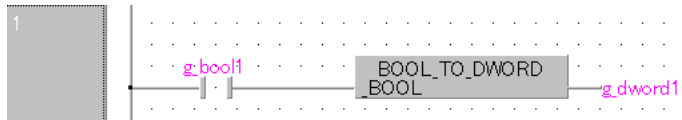
[ST]

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

- (2) 以下为将 ⑤ 中输入的位型数据转换为双字 [无符号]/ 位串 [32 位] 型的数据后通过 ④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (BOOL_TO_DWORD)

[结构体梯形图]



[ST]

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

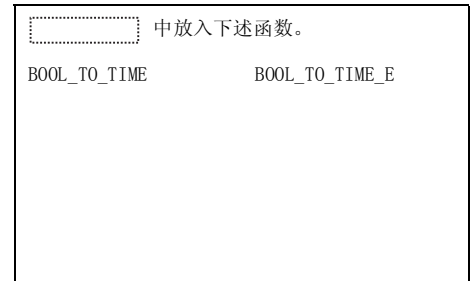
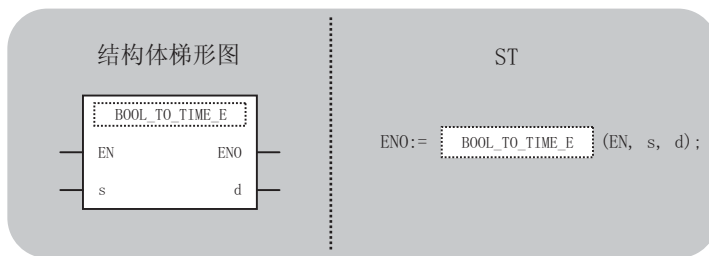
5.1.4 位型→时间型转换

BOOL_TO_TIME(_E)

Basic High performance Universal L CPU

BOOL_TO_TIME(_E)

_E : 带 EN/ENO



输入自变量, EN:	执行条件 (TRUE: 执行 FALSE: 停止)	: 位
s:	输入	: 位
输出自变量, ENO:	输出状态 (TRUE: 正常执行 FALSE: 异常或者停止)	: 位
d:	输出	: 时间

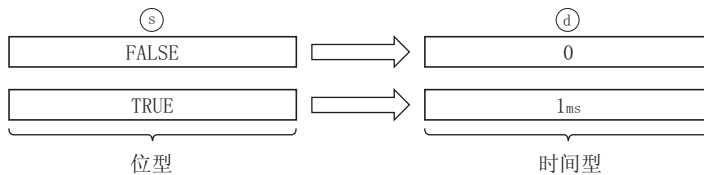
★ 功能

运算处理

将⑤中输入的位型数据转换为时间型的数据后通过④进行输出。

输入值为 FALSE 的情况下, 以时间型的数据值对 0 进行输出。

输入值为 TRUE 的情况下, 以时间型的数据值对 1 进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

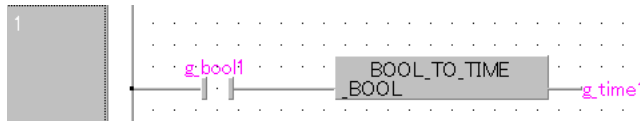
不存在 BOOL_TO_TIME(E) 相关的运算出错。

程序示例

以下为将⑤中输入的位型数据转换为时间型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (BOOL_TO_TIME)

[结构体梯形图]

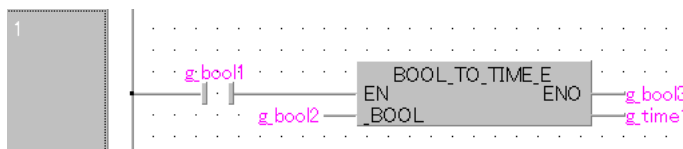


[ST]

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

(b) 带 EN/ENO 的函数 (BOOL_TO_TIME_E)

[结构体梯形图]



[ST]

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

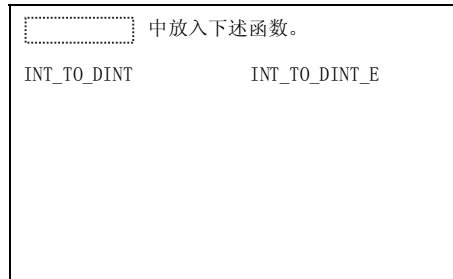
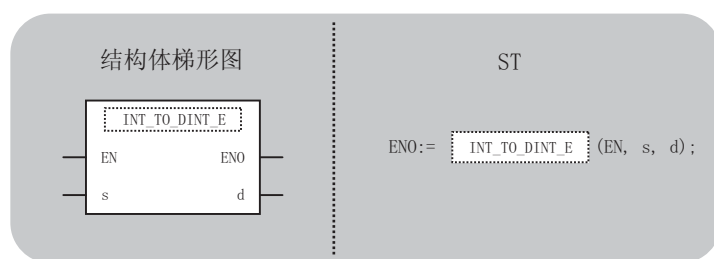
5.1.5 字 [带符号] 型 → 双字 [带符号] 型转换

INT_TO_DINT(_E)

Basic High performance Universal L CPU

INT_TO_DINT(_E)

_E : 带 EN/ENO



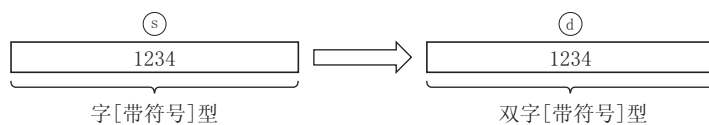
输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 字 [带符号]
 : 位
 : 双字 [带符号]

★ 功能

运算处理

将⑤中输入的字 [带符号] 型的数据转换为双字 [带符号] 型的数据后通过④进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

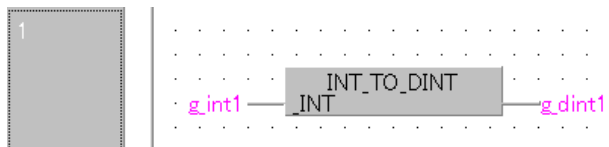
不存在 INT_TO_DINT(E) 相关的运算出错。

程序示例

以下为将⑤中输入的字[带符号]型的数据转换为双字[带符号]型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_DINT)

[结构体梯形图]

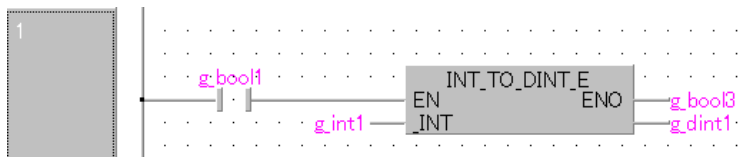


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_DINT_E)

[结构体梯形图]



[ST]

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

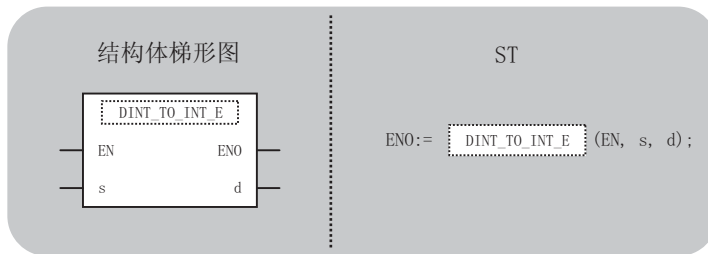
5.1.6 双字 [带符号] 型 → 字 [带符号] 型转换

DINT_TO_INT(_E)

Basic High performance Universal L CPU

DINT_TO_INT(_E)

(_E : 带 EN/ENO)



中放入下述函数。
DINT_TO_INT DINT_TO_INT_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 双字 [带符号]
: 位
: 字 [带符号]

★ 功能

运算处理

将⑤中输入的双字 [带符号] 型的数据转换为字 [带符号] 型的数据后通过④进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

☒ 要点

执行 DINT_TO_INT(E) 时, 仅将⑤中输入的双字 [带符号] 型数据值的低 16 位的值转换为字 [带符号] 型的数据。高 16 位的信息将被舍去。

! 出错

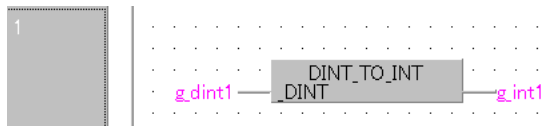
不存在 DINT_TO_INT(E) 相关的运算出错。

程序示例

以下为将⑤中输入的双字 [带符号] 型的数据转换为字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_INT)

[结构体梯形图]

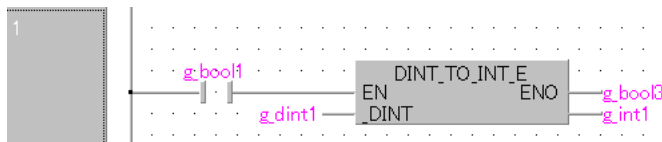


[ST]

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

(b) 带 EN/ENO 的函数 (DINT_TO_INT_E)

[结构体梯形图]



[ST]

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

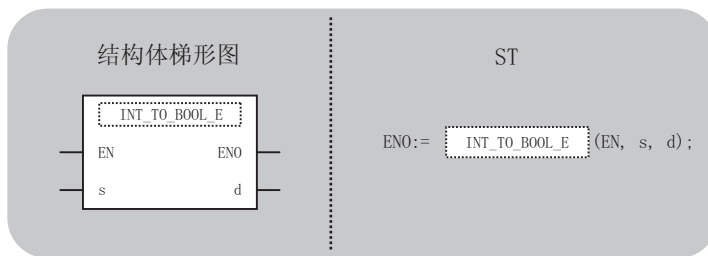
5.1.7 字 [带符号] 型、双字 [带符号] 型 → 位型转换

INT_TO_BOOL(_E), DINT_TO_BOOL(_E)

Basic High performance Universal L CPU

INT_TO_BOOL(_E)
DINT_TO_BOOL(_E)

(_E : 带 EN/ENO)



中放入下述函数。

INT_TO_BOOL INT_TO_BOOL_E
DINT_TO_BOOL DINT_TO_BOOL_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字 [带符号]、双字 [带符号]
: 位
: 位

★ 功能

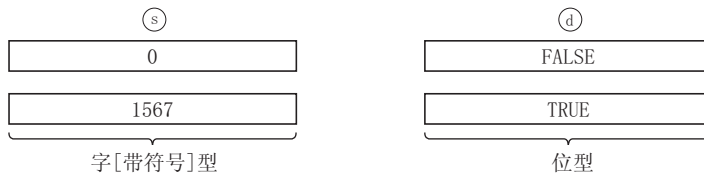
运算处理

(1) INT_TO_BOOL, INT_TO_BOOL_E

将③中输入的字 [带符号] 型的数据转换为位型的数据后通过④进行输出。

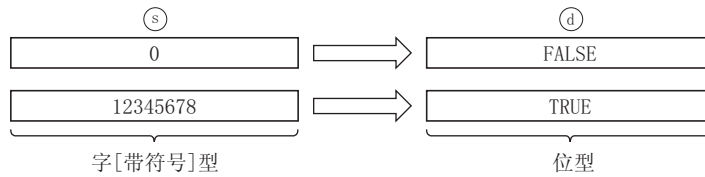
输入值为 0 的情况下, 对 FALSE 进行输出。

输入值为 0 以外的情况下, 对 TRUE 进行输出。



(2) DINT_TO_BOOL, DINT_TO_BOOL_E

将⑤中输入的双字[带符号]型的数据转换为位型的数据后通过④进行输出。
 输入值为0的情况下,对FALSE进行输出。
 输入值为0以外的情况下,对TRUE进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下,通过④输出的数据将变为不定值。
 在这种情况下,应执行程序处理以避免使用④中输出的数据。

出错

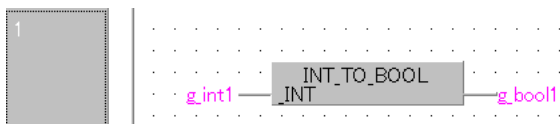
不存在 INT_TO_BOOL(_E), DINT_TO_BOOL(_E) 相关的运算出错。

程序示例

(1) 以下为将⑤中输入的字 [带符号] 型的数据转换为位型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_BOOL)

[结构体梯形图]

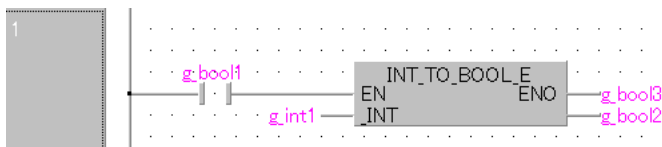


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_BOOL_E)

[结构体梯形图]



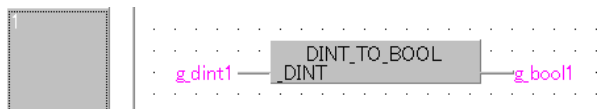
[ST]

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

(2) 以下为将⑤中输入的双字 [带符号] 型的数据转换为位型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_BOOL)

[结构体梯形图]



[ST]

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

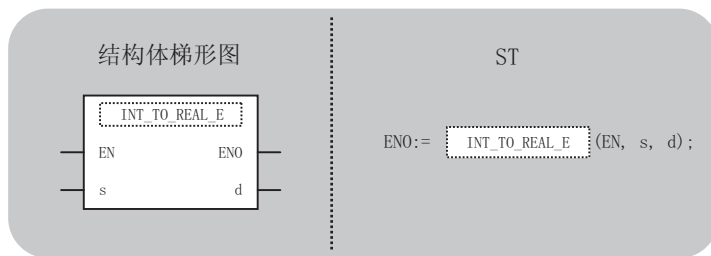

5.1.8 字 [带符号] 型、双字 [带符号] 型 → 单精度实数型转换

INT_TO_REAL(_E), DINT_TO_REAL(_E)

Basic High performance Universal L CPU

INT_TO_REAL(_E)
DINT_TO_REAL(_E)

_E : 带 EN/ENO



中放入下述函数。

INT_TO_REAL	INT_TO_REAL_E
DINT_TO_REAL	DINT_TO_REAL_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字 [带符号]、双字 [带符号]
: 位
: 单精度实数

★ 功能

运算处理

(1) INT_TO_REAL, INT_TO_REAL_E

将⑤中输入的字 [带符号] 型的数据转换为单精度实数型的数据后通过④进行输出。



(2) DINT_TO_REAL, DINT_TO_REAL_E

(a) 将⑤中输入的双字 [带符号] 型的数据转换为单精度实数型的数据后通过④进行输出。



(b) 由于对单精度实数型数据是以 32 位的单精度进行处理，有效位数约为 7 位。因此，整数值超出了 $-16777216 \sim 16777215$ 的范围的情况下，转换后的值将会产生误差。（四舍五入的误差）

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

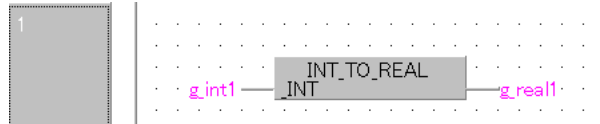
不存在 INT_TO_REAL(_E)、DINT_TO_REAL(_E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字 [带符号] 型的数据转换为单精度实数型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_REAL)

[结构体梯形图]

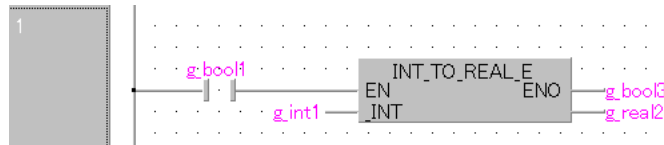


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_REAL_E)

[结构体梯形图]



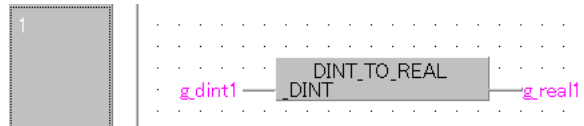
[ST]

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

- (2) 以下为将⑤中输入的双字 [带符号] 型的数据转换为单精度实数型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_REAL)

[结构体梯形图]



[ST]

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

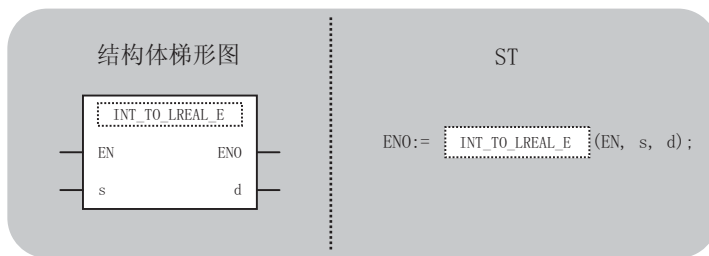
5.1.9 字 [带符号] 型、双字 [带符号] 型 → 双精度实数型转换

INT_TO_LREAL(_E), DINT_TO_LREAL(_E)



INT_TO_LREAL(_E)
DINT_TO_LREAL(_E)

_E : 带 EN/ENO



中放入下述函数。

INT_TO_LREAL	INT_TO_LREAL_E
DINT_TO_LREAL	DINT_TO_LREAL_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

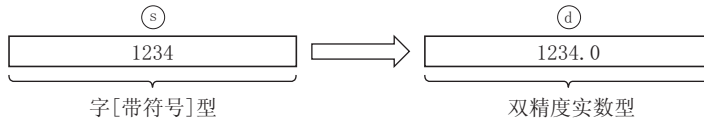
: 位
: 字 [带符号]、双字 [带符号]
: 位
: 双精度实数

★ 功能

运算处理

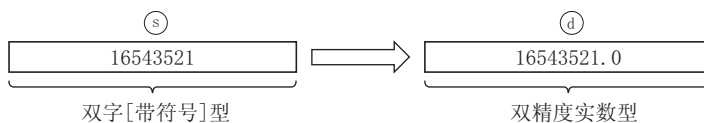
(1) INT_TO_LREAL, INT_TO_LREAL_E

将⑤中输入的字 [带符号] 型的数据转换为双精度实数型的数据后通过④进行输出。



(2) DINT_TO_LREAL, DINT_TO_LREAL_E

将⑤中输入的双字 [带符号] 型的数据转换为双精度实数型的数据后通过④进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

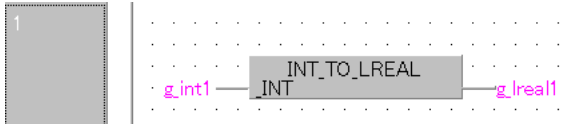
不存在 INT_TO_LREAL(E)、DINT_TO_LREAL(E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字 [带符号] 型的数据转换为双精度实数型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_LREAL)

[结构体梯形图]

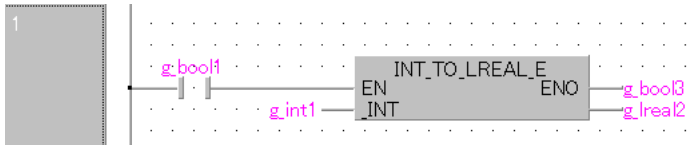


[ST]

```
g_lreal1 := INT_TO_LREAL(g_int1);
```

(b) 带 EN/ENO 的函数 (INT_TO_LREAL_E)

[结构体梯形图]



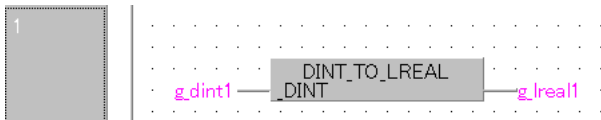
[ST]

```
g_bool3 := INT_TO_LREAL_E(g_bool1, g_int1, g_lreal2);
```

- (2) 以下为将⑤中输入的双字 [带符号] 型的数据转换为双精度实数型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_LREAL)

[结构体梯形图]



[ST]

```
g_lreal1 := DINT_TO_LREAL(g_dint1);
```

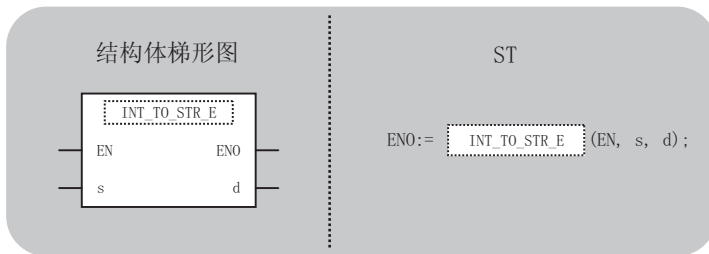
5.1.10 字[带符号]型、双字[带符号]型→字符串型转换

INT_TO_STR(_E), DINT_TO_STR(_E)

Basic High performance Universal L CPU

INT_TO_STR(_E)
DINT_TO_STR(_E)

_E : 带 EN/ENO



中放入下述函数。

INT_TO_STR	INT_TO_STR_E
DINT_TO_STR	DINT_TO_STR_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

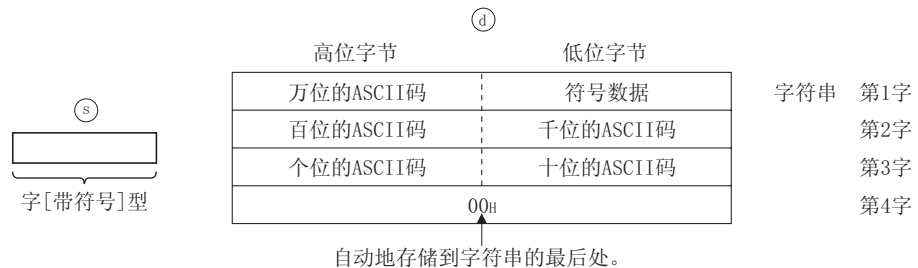
: 位
: 字[带符号]、双字[带符号]
: 位
: 字符串 (6)

★ 功能

运算处理

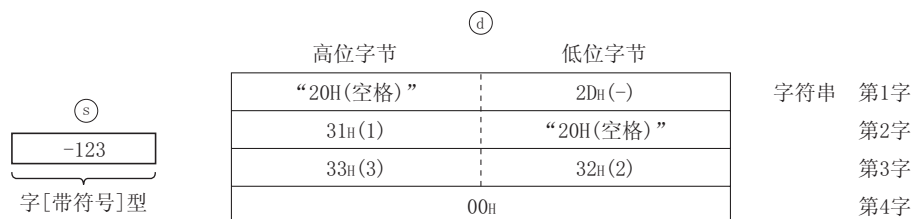
(1) INT_TO_STRING, INT_TO_STRING_E

(a) 将⑤中输入的字[带符号]型的数据转换为字符串型的数据后通过④进行输出。



(b) 在“符号数据”中, 输入的值为正时存储“20h(空格)”, 为负时存储“2Dh(-)”。

- (c) 有效位数较少的情况下，在高位中存储“20H(空格)”。
 (例) 输入了-123的情况下



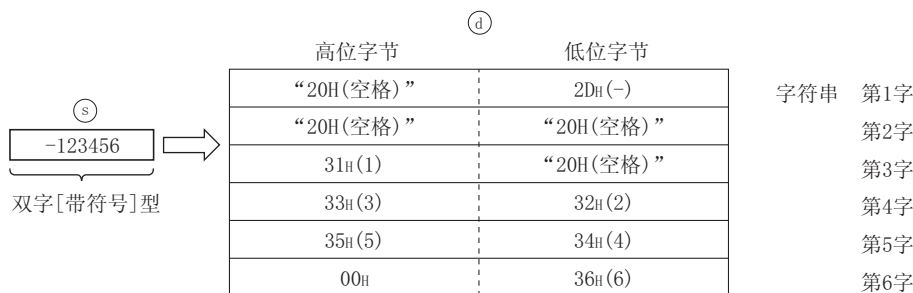
- (d) 在字符串的最后(第4字)中，将自动地存储“00H”。

(2) DINT_TO_STRING, DINT_TO_STRING_E

- (a) 将(s)中输入的双字[带符号]型的数据转换为字符串型的数据后通过(d)进行输出。



- (b) 在“符号数据”中，输入的值为正时存储“20H(空格)”，为负时存储“2Dh(-)”。
- (c) 有效位数较少的情况下，在高位中存储“20H(空格)”。
 (例) 输入了-123456的情况下



- (d) 在字符串的最后(第6字高位字节)处，将自动地存储“00H”。

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

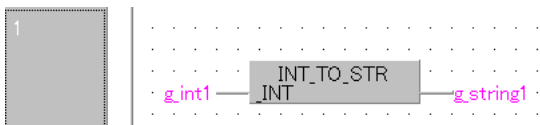
不存在 INT_TO_STR(E), DINT_TO_STR(E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字[带符号]型的数据转换为字符串型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_STR)

[结构体梯形图]

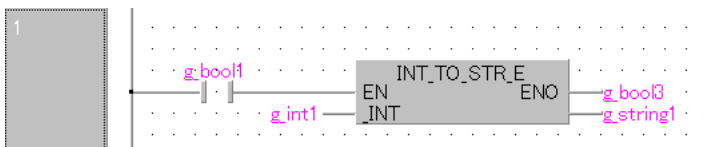


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_STR_E)

[结构体梯形图]



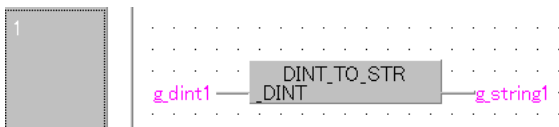
[ST]

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

- (2) 以下为将⑤中输入的双字[带符号]型的数据转换为字符串型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_STR)

[结构体梯形图]



[ST]

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

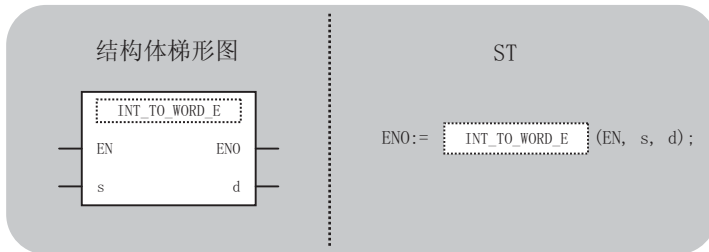
5.1.11 字 [带符号] 型、双字 [带符号] 型→字 [无符号] / 位串 [16 位] 型转换

INT_TO_WORD(_E), DINT_TO_WORD(_E)

Basic High performance Universal L CPU

INT_TO_WORD(_E)
DINT_TO_WORD(_E)

_E : 带 EN/ENO



中放入下述函数。

INT_TO_WORD	INT_TO_WORD_E
DINT_TO_WORD	DINT_TO_WORD_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字 [带符号]、双字 [带符号]
: 位
: 字 [无符号] / 位串 [16 位]

★ 功能

运算处理

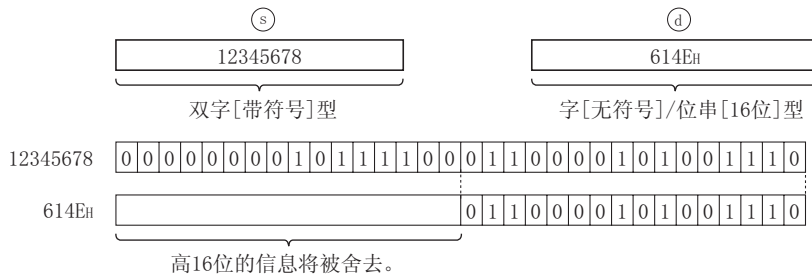
(1) INT_TO_WORD, INT_TO_WORD_E

将⑤中输入的字 [带符号] 型的数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过④进行输出。



(2) DINT_TO_WORD, DINT_TO_WORD_E

将⑤中输入的双字 [带符号] 型的数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

☒ 要 点

执行 DINT_TO_WORD(_E) 时, 仅⑤中输入的双字 [带符号] 型数据值的低 16 位的值被转换为字 [无符号] / 位串 [16 位] 型的数据。高 16 位的信息将被舍去。

出错

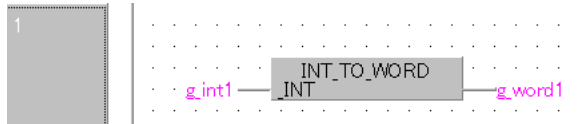
不存在 INT_TO_WORD(E)、DINT_TO_WORD(E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字 [带符号] 型的数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_WORD)

[结构体梯形图]

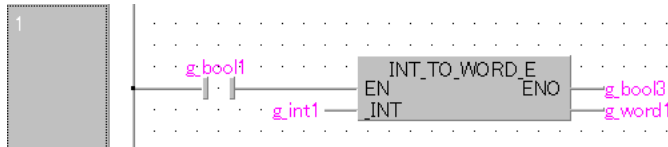


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_WORD_E)

[结构体梯形图]



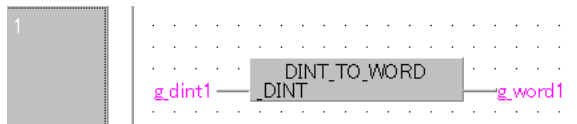
[ST]

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

- (2) 以下为将⑤中输入的双字 [带符号] 型的数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_WORD)

[结构体梯形图]



[ST]

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

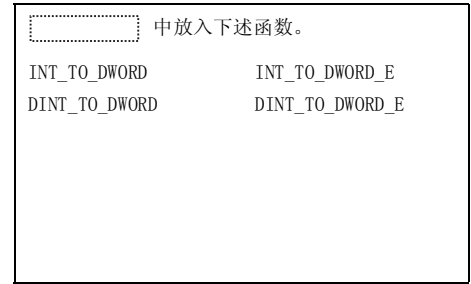
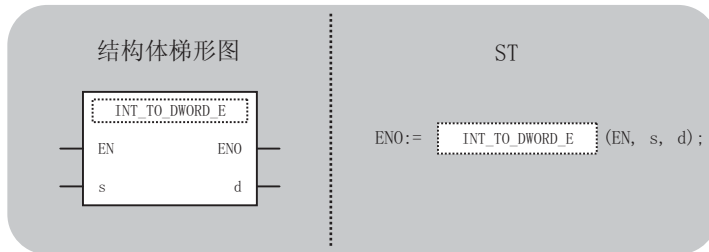
5.1.12 字 [带符号] 型、双字 [带符号] 型 → 双字 [无符号] / 位串 [32 位] 型转换

INT_TO_DWORD(_E), DINT_TO_DWORD(_E)

Basic High performance Universal L CPU

INT_TO_DWORD(_E)
DINT_TO_DWORD(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

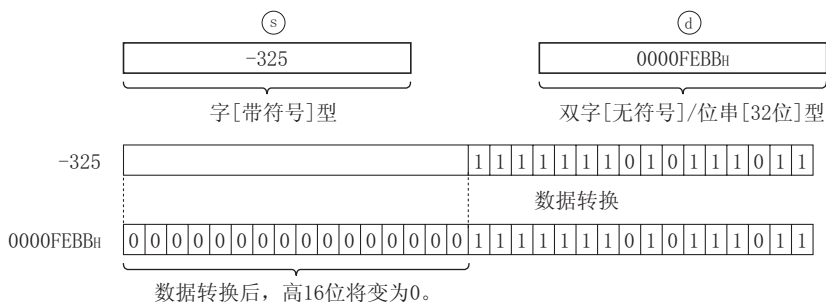
: 位
: 字 [带符号]、双字 [带符号]
: 位
: 双字 [无符号] / 位串 [32 位]

★ 功能

运算处理

(1) INT_TO_DWORD, INT_TO_DWORD_E

将 ③ 中输入的字 [带符号] 型的数据转换为双字 [无符号] / 位串 [32 位] 型的数据后通过 ④ 进行输出。



(2) DINT_TO_DWORD, DINT_TO_DWORD_E

将⑤中输入的双字[带符号]型的数据转换为双字[无符号]/位串[32位]型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

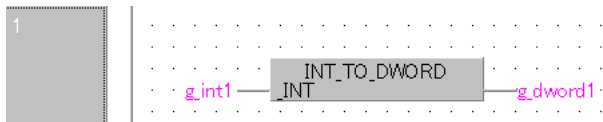
不存在 INT_TO_DWORD(_E), DINT_TO_DWORD(_E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字 [带符号] 型的数据转换为双字 [无符号] / 位串 [32 位] 型的数据后通④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_DWORD)

[结构体梯形图]

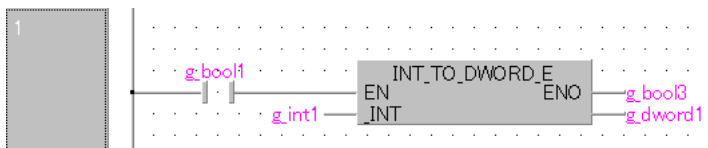


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_DWORD_E)

[结构体梯形图]



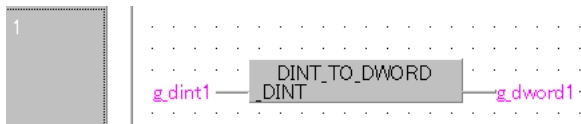
[ST]

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

- (2) 以下为将⑤中输入的双字 [带符号] 型的数据转换为双字 [无符号] / 位串 [32 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_DWORD)

[结构体梯形图]



[ST]

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

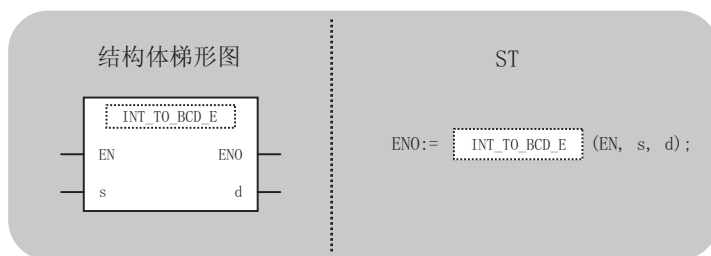

5.1.13 字 [带符号] 型、双字 [带符号] 型 → BCD 型转换

INT_TO_BCD(_E), DINT_TO_BCD(_E)

Basic High performance Universal L CPU

INT_TO_BCD(_E)
DINT_TO_BCD(_E)

_E : 带 EN/ENO



中放入下述函数。

INT_TO_BCD	INT_TO_BCD_E
DINT_TO_BCD	DINT_TO_BCD_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

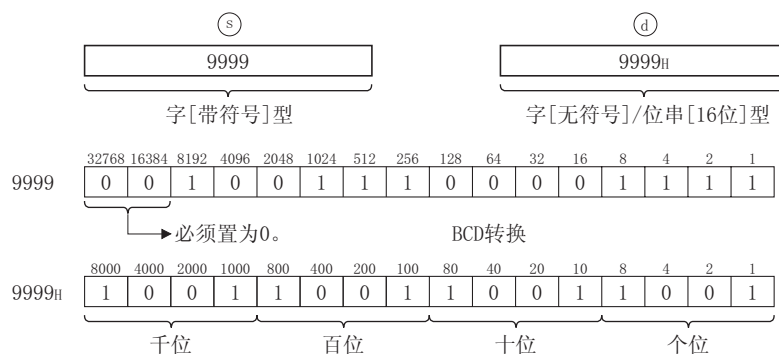
: 位
: 字 [带符号]、双字 [带符号]
: 位
: 字 [无符号]/位串 [16 位]、双字 [无符号]/位串 [32 位]

★ 功能

运算处理

(1) INT_TO_BCD, INT_TO_BCD_E

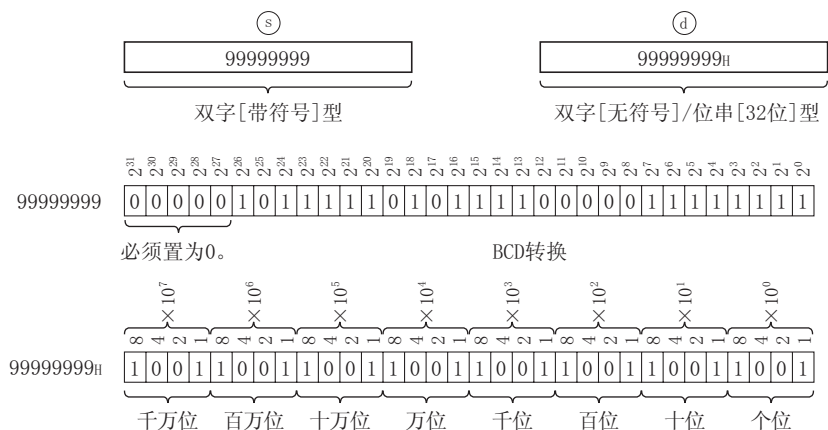
(a) 将⑤中输入的字 [带符号] 型的数据转换为 BCD 型的数据后通过④进行输出。



(b) 至⑤的输入值为字 [带符号] 型的数据值, 其范围为 0 ~ 9999。

(2) DINT_TO_BCD, DINT_TO_BCD_E

(a) 将⑤中输入的双字 [带符号] 型的数据转换为 BCD 型的数据后通过④进行输出。



(b) 至⑤的输入值为双字 [带符号] 型的数据值，其范围为 0 ~ 99999999。

运算结果

(1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

(2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE (无运算出错)	运算输出值
	FALSE (有运算出错) *1	不定值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

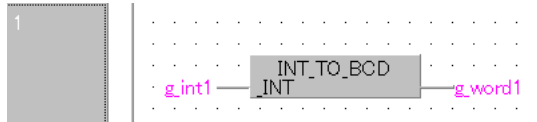
INT_TO_BCD(E)、DINT_TO_BCD(E) 中输入的值分别超过了 9999、99999999 的情况下将变为运算出错状态。
(出错代码： 4100)

程序示例

(1) 以下为将⑤中输入的字 [带符号] 型的数据转换为 BCD 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_BCD)

[结构体梯形图]

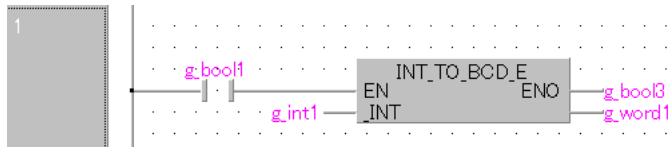


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_BCD_E)

[结构体梯形图]



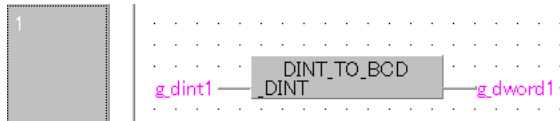
[ST]

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

(2) 以下为将⑤中输入的双字 [带符号] 型的数据转换为 BCD 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_BCD)

[结构体梯形图]



[ST]

```
s_dword1 := DINT_TO_BCD(g_dint1);
```

5.1.14 字 [带符号] 型、双字 [带符号] 型→时间型转换

INT_TO_TIME(_E), DINT_TO_TIME(_E)

Basic

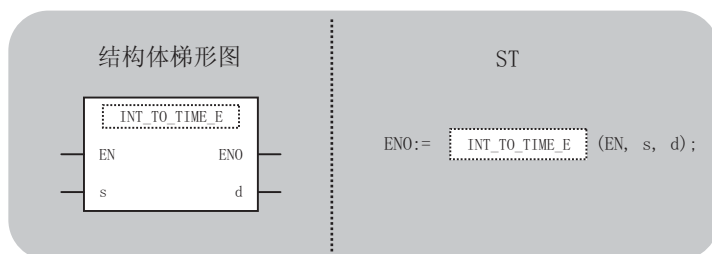
High
performance

Universal

L CPU

INT_TO_TIME(_E)
DINT_TO_TIME(_E)

_E : 带 EN/ENO



中放入下述函数。

INT_TO_TIME INT_TO_TIME_E
DINT_TO_TIME DINT_TO_TIME_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)

s: 输入

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)

d: 输出

: 位

: 字 [带符号]、双字 [带符号]

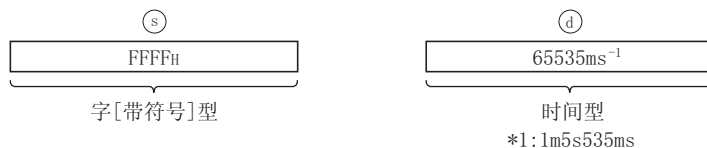
: 位

: 时间

★ 功能

运算处理

将③中输入的字 [带符号] 型的数据转换为时间型的数据后通过④进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *2	不定值

*2 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

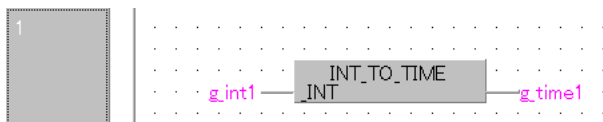
不存在 INT_TO_TIME(_E), DINT_TO_TIME(_E) 相关的运算出错。

程序示例

(1) 以下为将⑤中输入的字 [带符号] 型的数据转换为时间型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (INT_TO_TIME)

[结构体梯形图]

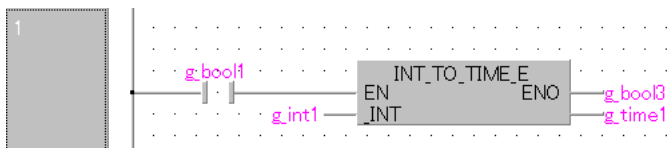


[ST]

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

(b) 带 EN/ENO 的函数 (INT_TO_TIME_E)

[结构体梯形图]



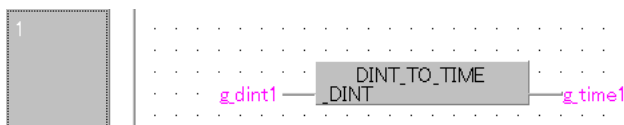
[ST]

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

(2) 以下为将⑤中输入的双字 [带符号] 型的数据转换为时间型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DINT_TO_TIME)

[结构体梯形图]



[ST]

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

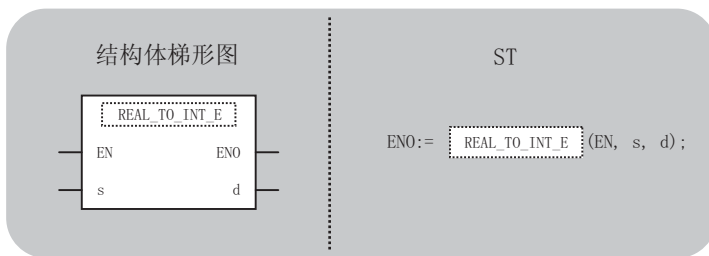
5.1.15 单精度实数型→字[带符号]型、双字[带符号]型转换

REAL_TO_INT(_E), REAL_TO_DINT(_E)

Basic High performance Universal L CPU

REAL_TO_INT(_E)
REAL_TO_DINT(_E)

_E : 带 EN/ENO



中放入下述函数。

REAL_TO_INT	REAL_TO_INT_E
REAL_TO_DINT	REAL_TO_DINT_E

输入自变量, EN: 执行条件(TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态(TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 单精度实数
: 位
: 字[带符号]、双字[带符号]

★ 功能

运算处理

(1) REAL_TO_INT, REAL_TO_INT_E

(a) 将⑤中输入的单精度实数型数据转换为字[带符号]型的数据后通过④进行输出。



(b) 至⑤的输入值为单精度实数型的数据值, 其范围为 -32768 ~ 32767。

(c) 对于转换后的数据, 将变为单精度实数型数据值的小数点以下第 1 位被四舍五入后的值。

(2) REAL_TO_DINT, REAL_TO_DINT_E

(a) 将⑤中输入的单精度实数型数据转换为双字[带符号]型的数据后通过④进行输出。



(b) 至⑤的输入值为单精度实数型的数据值，其范围为 $-2147483648 \sim 2147483647$ 。

(c) 对于转换后的数据，将变为单精度实数型数据值的小数点以下第1位被四舍五入后的值。

运算结果

(1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

(2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE(无运算出错)	运算输出值
	FALSE(有运算出错)*1	不定值
FALSE(停止运算)	FALSE*1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

! 出错

在以下情况下将变为运算出错状态。

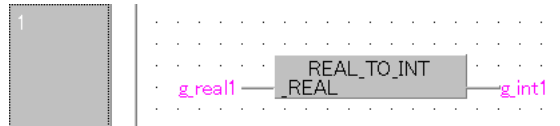
- REAL_TO_INT(E)：输入的值超出了 $-32768 \sim 32767$ 的范围的情况下。
(出错代码 4100)
- REAL_TO_DINT(E)：输入的值超出了 $-2147483648 \sim 2147483647$ 的范围的情况下。
(出错代码 4100)

程序示例

- (1) 以下为将⑤中输入的单精度实数型数据转换为字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (REAL_TO_INT)

[结构体梯形图]

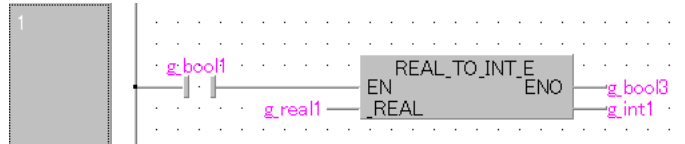


[ST]

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

(b) 带 EN/ENO 的函数 (REAL_TO_INT_E)

[结构体梯形图]



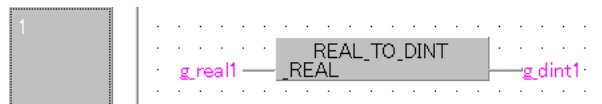
[ST]

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

- (2) 以下为将⑤中输入的单精度实数型数据转换为双字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (REAL_TO_DINT)

[结构体梯形图]



[ST]

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

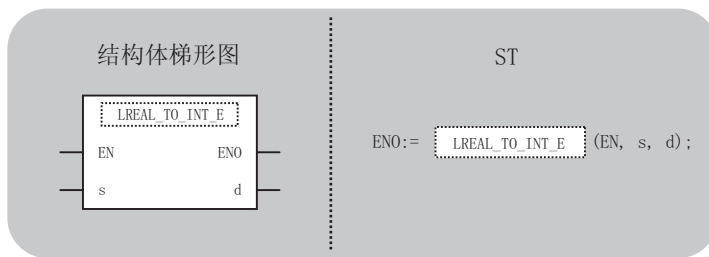
5.1.16 双精度实数型→字[带符号]型、双字[带符号]型转换

LREAL_TO_INT(_E), LREAL_TO_DINT(_E)



LREAL_TO_INT(_E)
LREAL_TO_DINT(_E)

(_E : 带 EN/ENO)



中放入下述函数。
LREAL_TO_INT LREAL_TO_INT_E
LREAL_TO_DINT LREAL_TO_DINT_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
: 双精度实数
: 位
: 字 [带符号]、双字 [带符号]

★ 功能

运算处理

(1) LREAL_TO_INT, LREAL_TO_INT_E

(a) 将⑤中输入的双精度实数型的数据转换为字[带符号]型的数据后通过④进行输出。



(b) 至⑤的输入值为双精度实数型的数据值，其范围为 -32768.0 ~ 32767.0。

(c) 对于转换后的数据，将变为双精度实数型数据值的小数点以下第 1 位被四舍五入后的值。

(2) LREAL_TO_DINT, LREAL_TO_DINT_E

(a) 将⑤中输入的双精度实数型的数据转换为双字[带符号]型的数据后通过④进行输出。



(b) 至⑤的输入值为双精度实数型的数据值，其范围为 $-2147483648.0 \sim 2147483647.0$ 。

(c) 对于转换后的数据，将变为双精度实数型数据值的小数点以下第1位被四舍五入后的值。

运算结果

(1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

(2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE(无运算出错)	运算输出值
	FALSE(有运算出错)*1	不定值
FALSE(停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

在以下情况下将变为运算出错状态。

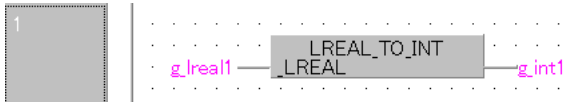
- 输入的值为 -0 或者超出下述范围时。 (出错代码：4140)
 $0, 2^{-1022} \leq | \textcircled{s}, \textcircled{d} | < 2^{1024}$
- LREAL_TO_INT(E)：输入的值超出了 -32768.0 ~ 32767.0 的范围的情况下。 (出错代码：4140)
- LREAL_TO_DINT(E)：输入的值超出了 -2147483648.0 ~ 2147483647.0 的范围的情况下。 (出错代码：4140)

程序示例

- (1) 以下为将 \textcircled{s} 中输入的双精度实数型的数据转换为字 [带符号] 型的数据后通过 \textcircled{d} 进行输出的程序。

(a) 无 EN/ENO 的函数 (LREAL_TO_INT)

[结构体梯形图]

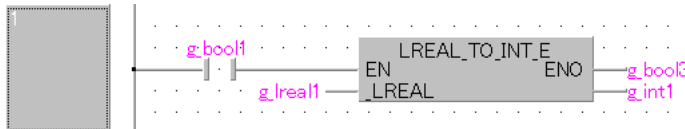


[ST]

```
g_int1 := LREAL_TO_INT(g_lreal1);
```

(b) 带 EN/ENO 的函数 (LREAL_TO_INT_E)

[结构体梯形图]



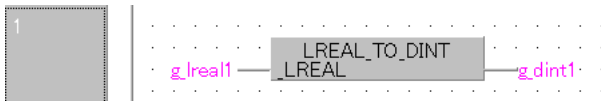
[ST]

```
g_bool3 := LREAL_TO_INT_E(g_bool1, g_lreal1, g_int1);
```

- (2) 以下为将 \textcircled{s} 中输入的双精度实数型的数据转换为双字 [带符号] 型的数据后通过 \textcircled{d} 进行输出的程序。

(a) 无 EN/ENO 的函数 (LREAL_TO_DINT)

[结构体梯形图]



[ST]

```
g_dint1 := LREAL_TO_DINT(g_lreal1);
```

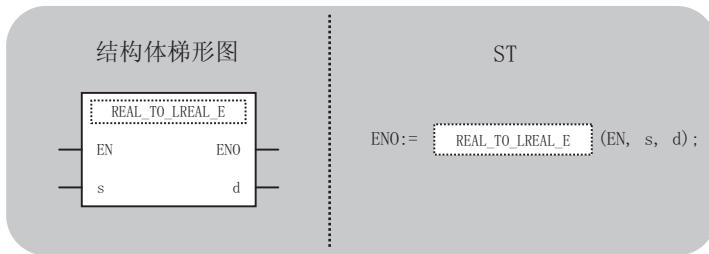
5.1.17 单精度实数型→双精度实数型转换

REAL_TO_LREAL(_E)

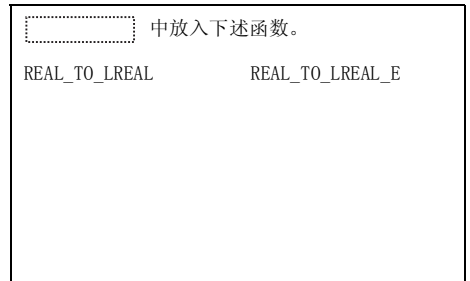


REAL_TO_LREAL(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出



: 位
 : 单精度实数
 : 位
 : 双精度实数

★ 功能

运算处理

将③中输入的单精度实数型的数据转换为双精度实数型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数
 运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE(无运算出错)	运算输出值
	FALSE(有运算出错)*1	不定值
FALSE(停止运算)	FALSE*1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过 ④ 输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用 ④ 中输出的数据。

出错

在以下情况下将变为运算出错状态。

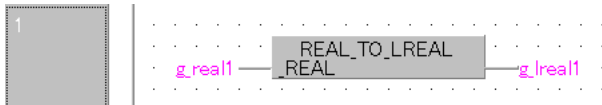
- 输入的值为 -0 或者超出下述范围时。 (出错代码: 4140)
 $0, 2^{-126} \leq | \textcircled{S} | < 2^{128}$
- 运算结果超出下述范围时 (发生了溢出时)。
 $2^{1024} \leq | \text{运算结果} |$ (出错代码: 4141)

程序示例

以下为将 ⑤ 中输入的单精度实数型的数据转换为双精度实数型的数据后通过 ④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (REAL_TO_LREAL)

[结构体梯形图]

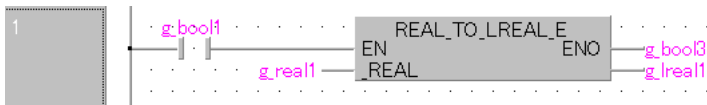


[ST]

```
g_lreal1 := REAL_TO_LREAL(g_real1);
```

(b) 带 EN/ENO 的函数 (REAL_TO_LREAL_E)

[结构体梯形图]



[ST]

```
g_bool3 := REAL_TO_LREAL_E(g_bool1, g_real1, g_lreal1);
```

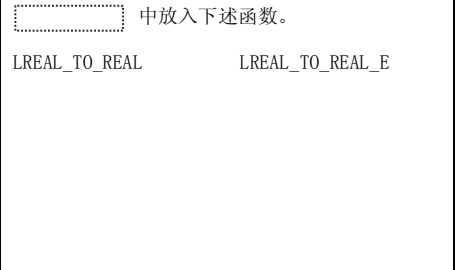
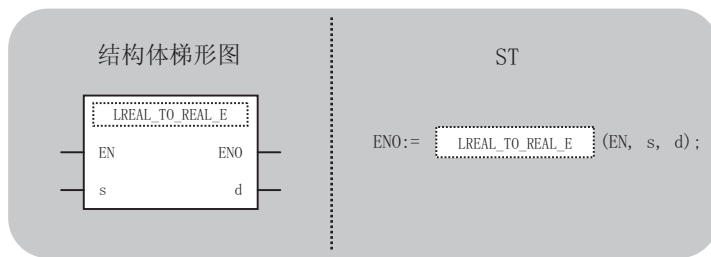
5.1.18 双精度实数型→单精度实数型转换

LREAL_TO_REAL(_E)



LREAL_TO_REAL(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 双精度实数
 : 位
 : 单精度实数

★ 功能

运算处理

将③中输入的双精度实数型的数据转换为单精度实数型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数
 运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE(无运算出错)	运算输出值
	FALSE(有运算出错)*1	不定值
FALSE(停止运算)	FALSE*1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过 ④ 输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用 ④ 中输出的数据。

出错

在以下情况下将变为运算出错状态。

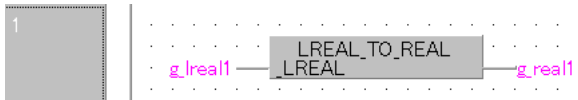
- 输入的值 -0 或者超出下述范围时。 (出错代码: 4140)
 $0, 2^{-1022} \leq | \text{Ⓢ} | < 2^{1024}$
- 运算结果超出下述范围时 (发生了溢出时)。 (出错代码: 4141)
 $2^{128} \leq | \text{运算结果} |$

程序示例

以下为将 ⑤ 中输入的双精度实数型的数据转换为单精度实数型的数据后通过 ④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (LREAL_TO_REAL)

[结构体梯形图]

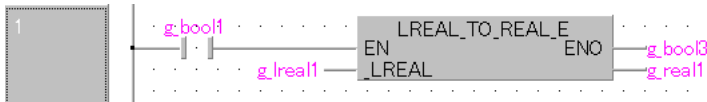


[ST]

```
g_real1 := LREAL_TO_REAL(g_lreal1);
```

(b) 带 EN/ENO 的函数 (LREAL_TO_REAL_E)

[结构体梯形图]



[ST]

```
g_bool3 := LREAL_TO_REAL_E(g_bool1, g_lreal1, g_real1);
```

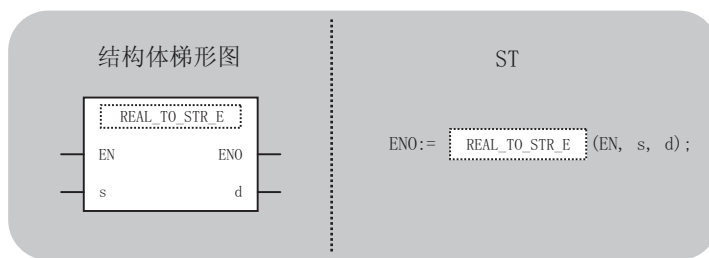

5.1.19 单精度实数型→字符串型转换

REAL_TO_STR(E)

Basic High performance Universal L CPU

REAL_TO_STR(E)

(_E : 带 EN/ENO)



ST

ENO:= REAL_TO_STR_E(EN, s, d);

中放入下述函数。

REAL_TO_STR REAL_TO_STR_E

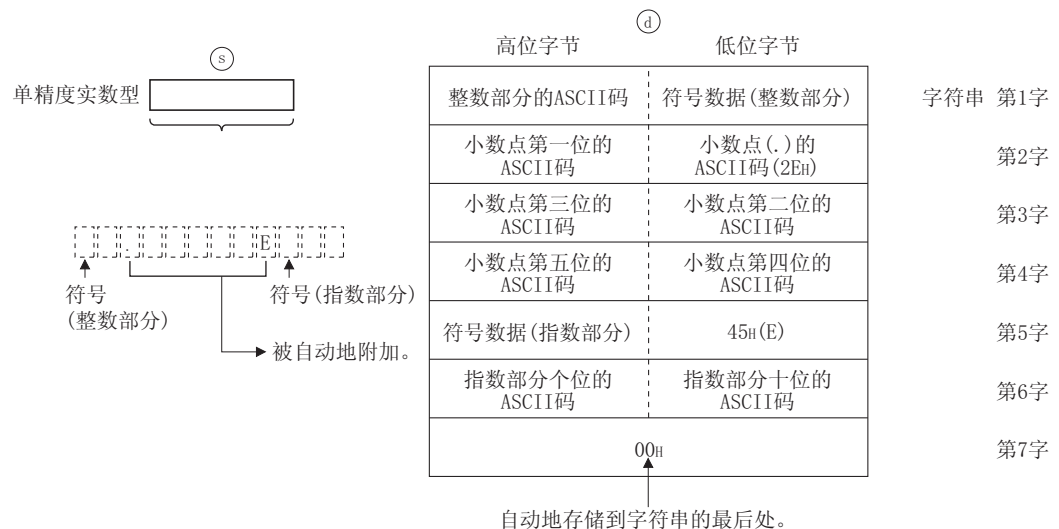
输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 单精度实数
 : 位
 : 字符串 (12)

★ 功能

运算处理

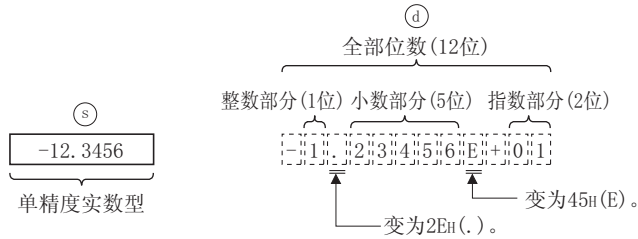
(1) 将⑤中输入的单精度实数型数据转换为字符串型 (指数格式) 的数据后通过④进行输出。



(2) 转换后的字符串数据通过④按以下方式输出。

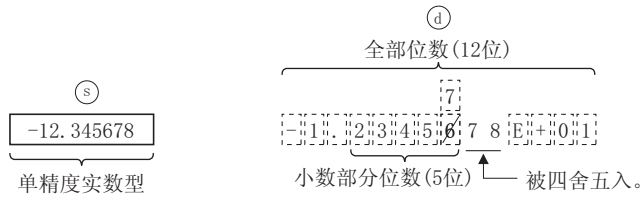
(a) 整数部分、小数部分、指数部分的位数是固定的。(整数部分：1位；小数部分：5位；指数部分：2位)

在第3字节、第9字节中，将分别自动地存储“2EH”(.)及“45H”(E)。

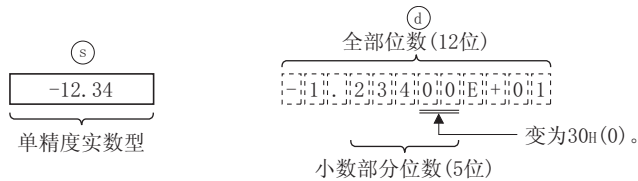


(b) 在符号数据(整数部分)中,输入的值为正时存储“20H”(空格),为负时存储“2DH”(-)。

(c) 小数部分的第6位数及以后将被四舍五入。

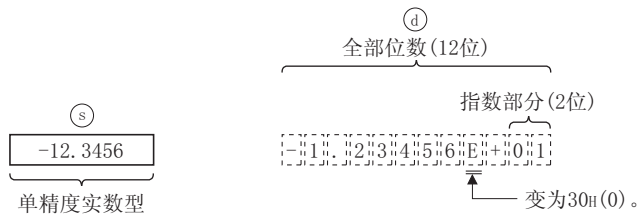


(d) 有效位数较少的情况下,在小数部分存储“30H”(0)。



(e) 在符号数据(指数部分)中,指数为正时存储“2BH”(+) ,为负时存储“2DH”(-)。

(f) 指数部分为1位数的情况下,在指数部分的十位中存储“30H”(0)。



(3) 在字符串的最后(第7字)中,将自动地存储“00H”。

运算结果

- (1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE (无运算出错)	运算输出值
	FALSE (有运算出错) *1	不定值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

在下述情况下将变为运算出错状态。

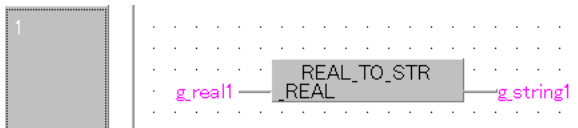
- 输入的值超出了 $\pm 1.17549 \times 10^{-38} \sim \pm 3.40282 \times 10^{38}$ 的范围时。 (出错代码：4100)

程序示例

以下为将①中输入的单精度实数型的数据转换为字符串型（指数格式）的数据后通过②进行输出的程序。

(a) 无 EN/ENO 的函数 (REAL_TO_STR)

[结构体梯形图]

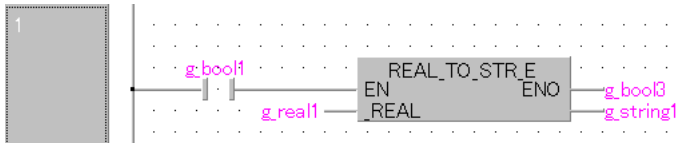


[ST]

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

(b) 带 EN/ENO 的函数 (REAL_TO_STR_E)

[结构体梯形图]



[ST]

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

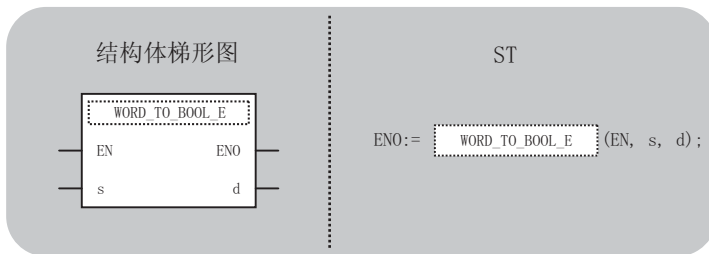
5.1.20 字 [无符号]/位串 [16 位] 型、双字 [无符号]/位串 [32 位] 型→位型转换

WORD_TO_BOOL(_E), DWORD_TO_BOOL(_E)

Basic High performance Universal L CPU

WORD_TO_BOOL(_E)
DWORD_TO_BOOL(_E)

(_E : 带 EN/ENO)



中放入下述函数。

WORD_TO_BOOL WORD_TO_BOOL_E
DWORD_TO_BOOL DWORD_TO_BOOL_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

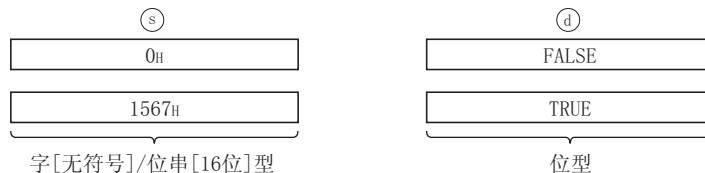
: 位
: 字 [无符号]/位串 [16 位], 双字 [无符号]/位串 [32 位]
: 位
: 位

★ 功能

运算处理

(1) WORD_TO_BOOL, WORD_TO_BOOL_E

将①中输入的字 [无符号]/位串 [16 位] 型的数据转换为位型的数据后通过②进行输出。
输入值为 0H 的情况下, 对 FALSE 进行输出。
输入值为 0H 以外的情况下, 对 TRUE 进行输出。

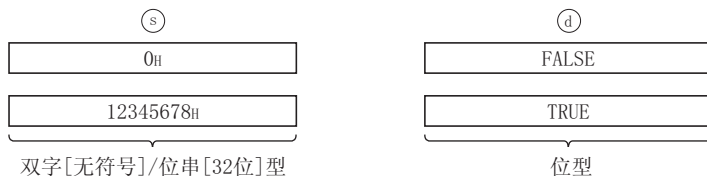


(2) DWORD_TO_BOOL, DWORD_TO_BOOL_E

将⑤中输入的双字[无符号]/位串[32位]型的数据转换为位型的数据后通过④进行输出。

输入值为 0H 的情况下，对 FALSE 进行输出。

输入值为 0H 以外的情况下，对 TRUE 进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

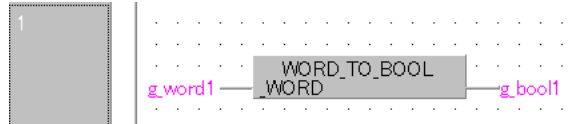
不存在 WORD_TO_BOOL(_E)、DWORD_TO_BOOL(_E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字 [无符号] / 位串 [16 位] 型的数据转换为位型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (WORD_TO_BOOL)

[结构体梯形图]

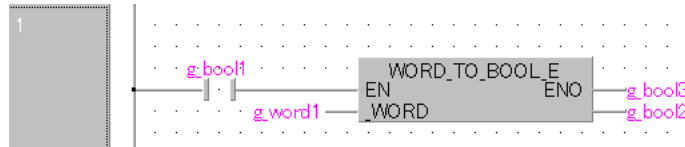


[ST]

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

(b) 带 EN/ENO 的函数 (WORD_TO_BOOL_E)

[结构体梯形图]



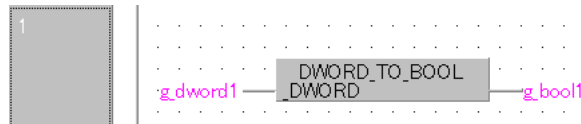
[ST]

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

- (2) 以下为将⑤中输入的双字 [无符号] / 位串 [32 位] 型的数据转换为位型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DWORD_TO_BOOL)

[结构体梯形图]



[ST]

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

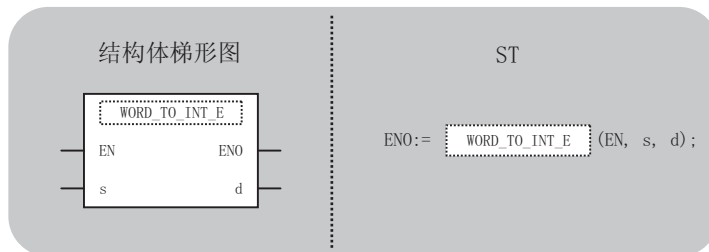
5.1.21 字 [无符号] / 位串 [16 位] 型 → 字 [带符号] 型、 双字 [带符号] 型转换

WORD_TO_INT(_E), WORD_TO_DINT(_E)

Basic High performance Universal L CPU

WORD_TO_INT(_E)
WORD_TO_DINT(_E)

_E : 带 EN/ENO



中放入下述函数。

```
WORD_TO_INT      WORD_TO_INT_E
WORD_TO_DINT     WORD_TO_DINT_E
```

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字 [无符号] / 位串 [16 位]
: 位
: 字 [带符号]、双字 [带符号]

★ 功能

运算处理

(1) WORD_TO_INT, WORD_TO_INT_E

将③中输入的字 [无符号] / 位串 [16 位] 型的数据转换为字 [带符号] 型的数据后通过④进行输出。



出错

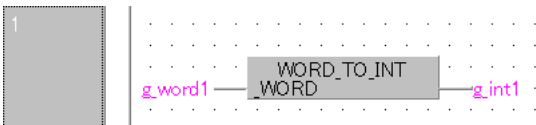
不存在 WORD_TO_INT(E)、WORD_TO_DINT(E) 相关的运算出错。

程序示例

(1) 以下为将⑤中输入的字 [无符号] / 位串 [16 位] 型的数据转换为字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (WORD_TO_INT)

[结构体梯形图]

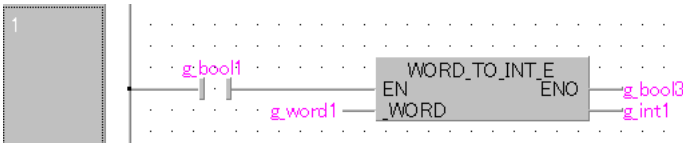


[ST]

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

(b) 带 EN/ENO 的函数 (WORD_TO_INT_E)

[结构体梯形图]



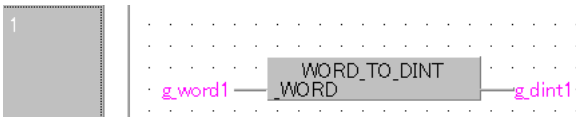
[ST]

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

(2) 以下为将⑤中输入的字 [无符号] / 位串 [16 位] 型的数据转换为双字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (WORD_TO_DINT)

[结构体梯形图]



[ST]

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

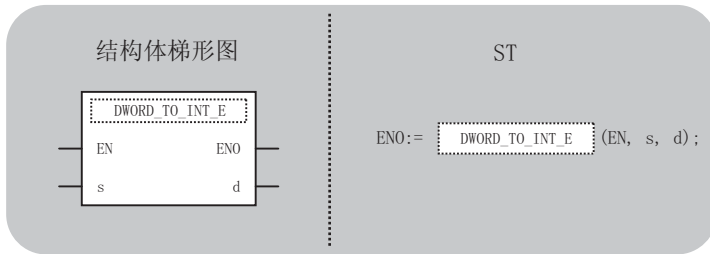
5.1.22 双字 [无符号] / 位串 [32 位] 型 → 字 [带符号] 型、 双字 [带符号] 型转换

DWORD_TO_INT(_E), DWORD_TO_DINT(_E)

Basic High performance Universal L CPU

DWORD_TO_INT(_E)
DWORD_TO_DINT(_E)

(_E : 带 EN/ENO)



中放入下述函数。
 DWORD_TO_INT DWORD_TO_INT_E
 DWORD_TO_DINT DWORD_TO_DINT_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

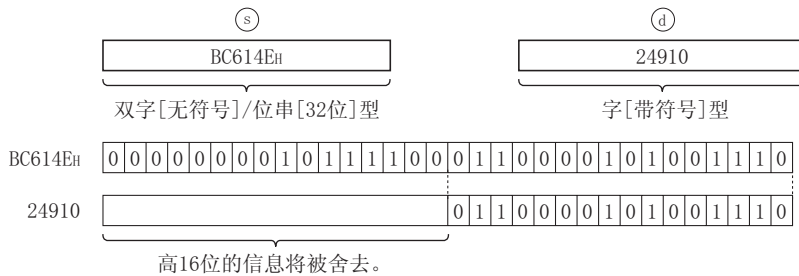
: 位
 : 双字 [无符号] / 位串 [32 位]
 : 位
 : 字 [带符号]、双字 [带符号]

★ 功能

运算处理

(1) DWORD_TO_INT, DWORD_TO_INT_E

将 (s) 中输入的双字 [无符号] / 位串 [32 位] 型的数据转换为字 [带符号] 型的数据后通过 (d) 进行输出。



5
应用函数
DWORD_TO_INT(_E), DWORD_TO_DINT(_E)

(2) DWORD_TO_DINT, DWORD_TO_DINT_E

将⑤中输入的双字 [无符号] / 位串 [32 位] 型的数据转换为双字 [带符号] 型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

☒ 要 点

执行 DWORD_TO_INT(_E) 时, 仅从⑤中输入的双字 [无符号] / 位串 [32 位] 型数据值的低 16 位的值被转换为字 [带符号] 型的数据。高 16 位的信息将被舍去。

出错

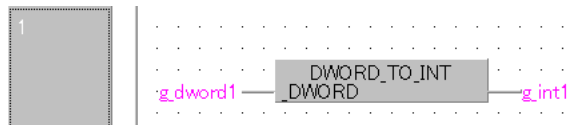
不存在 DWORD_TO_INT(_E)、DWORD_TO_DINT(_E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的双字 [无符号]/ 位串 [32 位] 型的数据转换为字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DWORD_TO_INT)

[结构体梯形图]

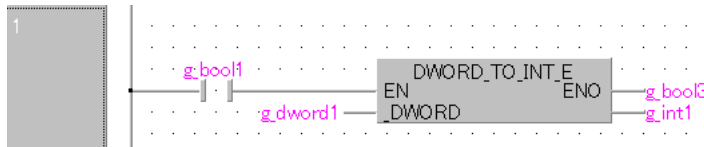


[ST]

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

(b) 带 EN/ENO 的函数 (DWORD_TO_INT_E)

[结构体梯形图]



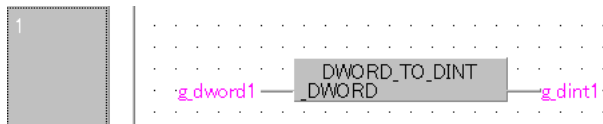
[ST]

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

- (2) 以下为将⑤中输入的双字 [无符号]/ 位串 [32 位] 型的数据转换为双字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DWORD_TO_DINT)

[结构体梯形图]



[ST]

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

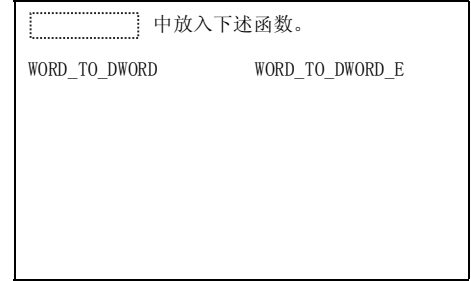
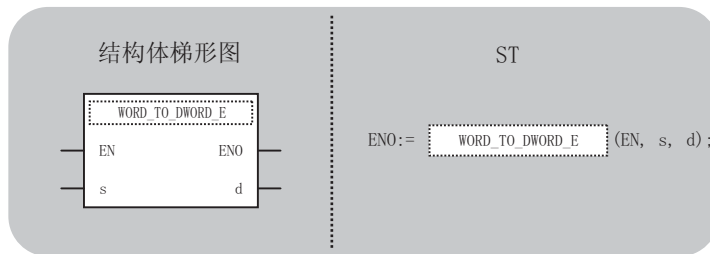
5.1.23 字 [无符号] / 位串 [16 位] 型 → 双字 [无符号] / 位串 [32 位] 型转换

WORD_TO_DWORD(_E)

Basic High performance Universal L CPU

WORD_TO_DWORD(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字 [无符号] / 位串 [16 位]
: 位
: 双字 [无符号] / 位串 [32 位]

★ 功能

运算处理

将⑤中输入的字 [无符号] / 位串 [16 位] 型的数据转换为双字 [无符号] / 位串 [32 位] 型的数据后通过⑥进行输出。

数据转换后, 高 16 位将变为 0。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

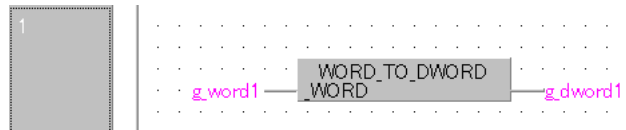
不存在 WORD_TO_DWORD(_E) 相关的运算出错。

程序示例

以下为将⑤中输入的字 [无符号]/位串 [16 位] 型的数据转换为双字 [无符号]/位串 [32 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (WORD_TO_DWORD)

[结构体梯形图]

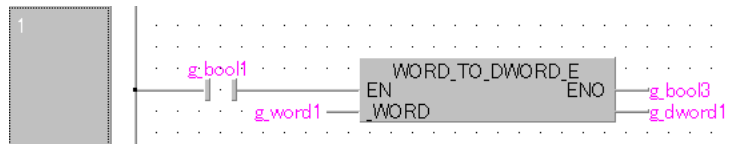


[ST]

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

(b) 带 EN/ENO 的函数 (WORD_TO_DWORD_E)

[结构体梯形图]



[ST]

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

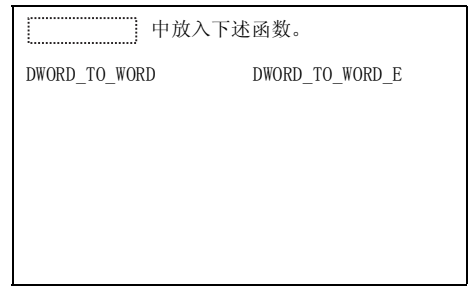
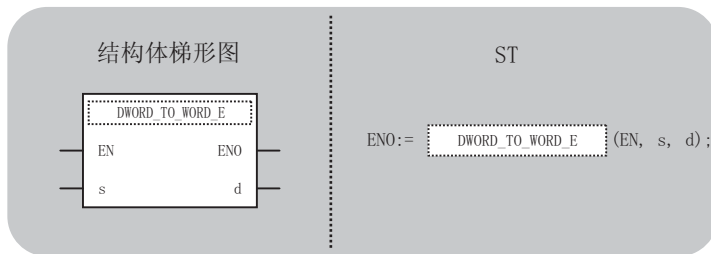
5.1.24 双字 [无符号] / 位串 [32 位] 型 → 字 [无符号] / 位串 [16 位] 型转换

DWORD_TO_WORD(_E)

Basic High performance Universal L CPU

DWORD_TO_WORD(_E)

(_E : 带 EN/ENO)



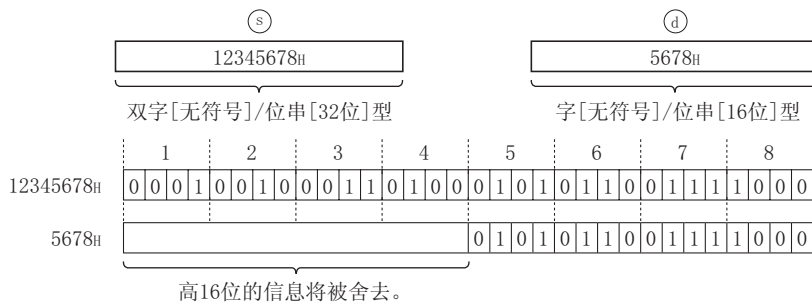
输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 双字 [无符号] / 位串 [32 位]
 : 位
 : 字 [无符号] / 位串 [16 位]

★ 功能

运算处理

将 (s) 中输入的双字 [无符号] / 位串 [32 位] 型的数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过 (d) 进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

☒ 要 点

执行 DWORD_TO_WORD(_E) 时, 仅从⑤中输入的双字 [无符号] / 位串 [32 位] 型数据值的低 16 位的值被转换为字 [无符号] / 位串 [16 位] 型的数据。
高 16 位的信息将被舍去。

出错

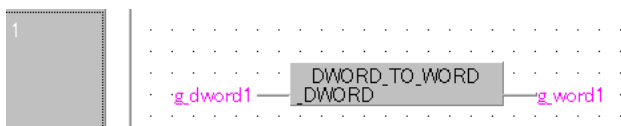
不存在 DWORD_TO_WORD(E) 相关的运算出错。

程序示例

以下为将③中输入的双字 [无符号] / 位串 [32 位] 型的数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DWORD_TO_WORD)

[结构体梯形图]

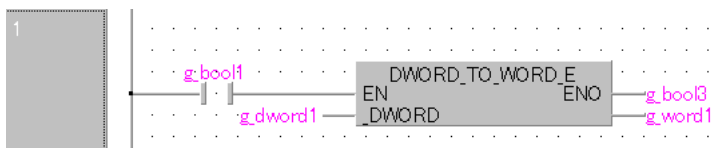


[ST]

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

(b) 带 EN/ENO 的函数 (DWORD_TO_WORD_E)

[结构体梯形图]



[ST]

```
g_bool13 := DWORD_TO_WORD_E(g_bool1, g_dword1, g_word1);
```

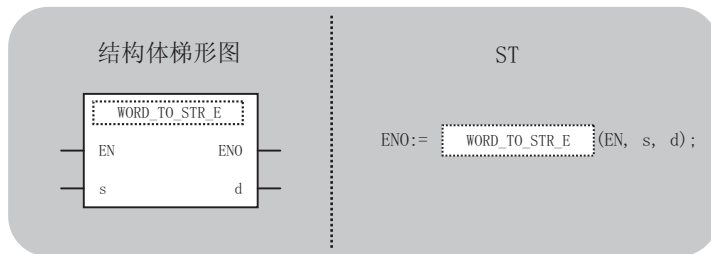
5.1.25 字 [无符号]/位串 [16 位] 型、双字 [无符号]/位串 [32 位] 型→字符串型转换

WORD_TO_STR(_E), DWORD_TO_STR(_E)

Basic High performance Universal L CPU

WORD_TO_STR(_E)
DWORD_TO_STR(_E)

_E : 带 EN/ENO



中放入下述函数。

```
WORD_TO_STR      WORD_TO_STR_E
DWORD_TO_STR     DWORD_TO_STR_E
```

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

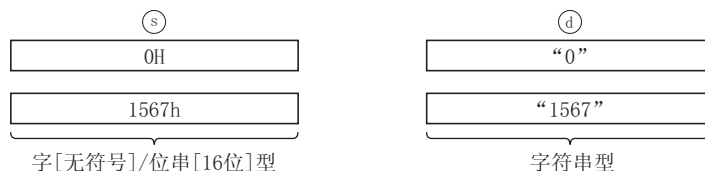
: 位
: 字 [无符号]/位串 [16 位],
双字 [无符号]/位串 [32 位]
: 位
: 字符串

★ 功能

运算处理

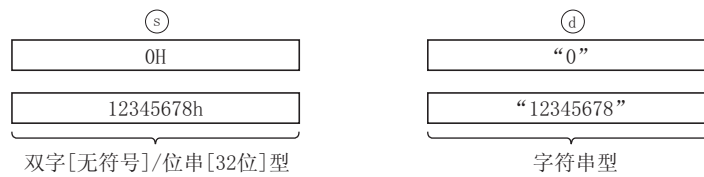
(1) WORD_TO_STRING, WORD_TO_STRING_E

将①中输入的字 [无符号]/位串 [16 位] 型的数据转换为字符串型的数据后通过②进行输出。



(2) DWORD_TO_STRING, DWORD_TO_STRING_E

将③中输入的双字 [无符号] / 位串 [32 位] 型的数据转换为字符串型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

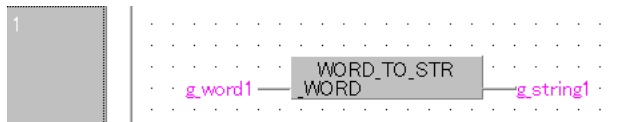
不存在 WORD_TO_STR(E)、DWORD_TO_STR(E) 相关的运算出错。

程序示例

- (1) 以下为将 ⑤ 中输入的字 [无符号]/ 位串 [16 位] 型的数据转换为字符串型的数据后通过 ④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (WORD_TO_STR)

[结构体梯形图]

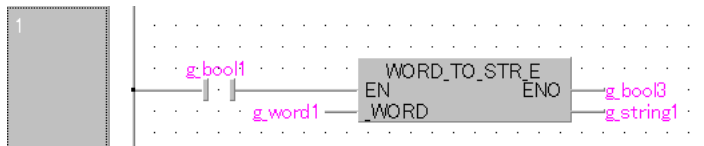


[ST]

```
g_string1 := WORD_TO_STR (g_word1);
```

(b) 带 EN/ENO 的函数 (WORD_TO_STR_E)

[结构体梯形图]



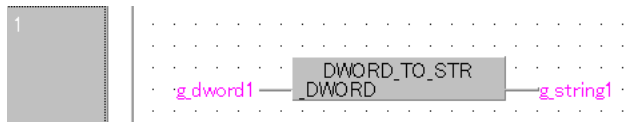
[ST]

```
g_bool13 := WORD_TO_STR_E (g_bool1, g_word1, g_string1);
```

- (2) 以下为将 ⑤ 中输入的双字 [无符号]/ 位串 [32 位] 型的数据转换为字符串型的数据后通过 ④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (DWORD_TO_STR)

[结构体梯形图]



[ST]

```
g_string1:= DWORD_TO_STR (g_dword1);
```

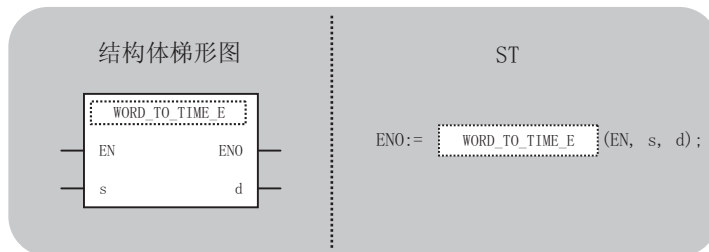
5.1.26 字 [无符号]/位串 [16位] 型、双字 [无符号]/位串 [32位] 型→时间型转换

WORD_TO_TIME(_E), DWORD_TO_TIME(_E)

Basic High performance Universal L CPU

WORD_TO_TIME(_E)
DWORD_TO_TIME(_E)

_E : 带 EN/ENO



中放入下述函数。

```
WORD_TO_TIME
WORD_TO_TIME_E
DWORD_TO_TIME
DWORD_TO_TIME_E
```

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

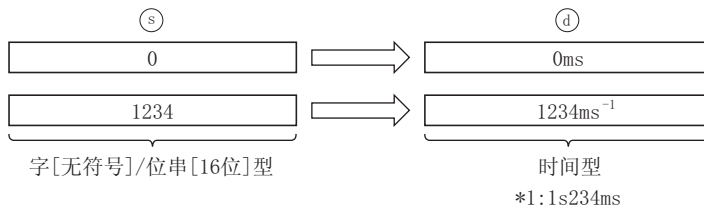
: 位
: 字 [无符号]/位串 [16位],
双字 [无符号]/位串 [32位]
: 位
: 时间

★ 功能

运算处理

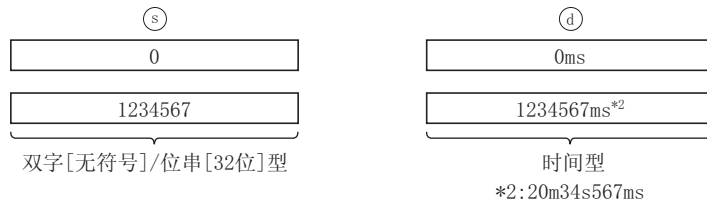
(1) WORD_TO_TIME, WORD_TO_TIME_E

将⑤中输入的字 [无符号]/位串 [16位] 型的数据转换为时间型的数据后通过④进行输出。



(2) DWORD_TO_TIME, DWORD_TO_TIME_E

将⑤中输入的双字[无符号]/位串[32位]型的数据转换为时间型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *3	不定值

*3: 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

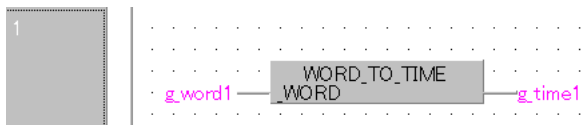
不存在 WORD_TO_TIME(E)、DWORD_TO_TIME(E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字 [无符号]/ 位串 [16 位] 型的数据转换为时间型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (WORD_TO_TIME)

[结构体梯形图]

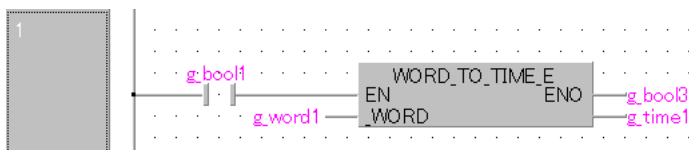


[ST]

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

(b) 带 EN/ENO 的函数 (WORD_TO_TIME_E)

[结构体梯形图]



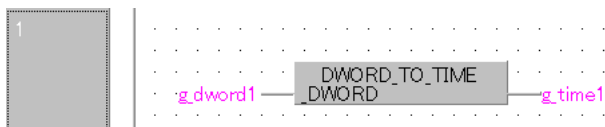
[ST]

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

- (2) 以下为将⑤中输入的双字 [无符号]/ 位串 [32 位] 型的数据转换为时间型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DWORD_TO_TIME)

[结构体梯形图]



[ST]

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

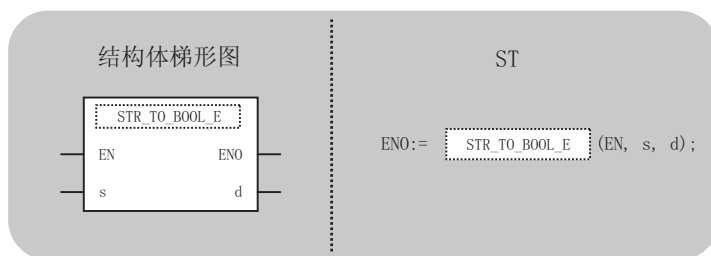

5.1.27 字符串型→位型转换

STR_TO_BOOL(_E)

Basic High performance Universal L CPU

STR_TO_BOOL(_E)

_E : 带 EN/ENO



中放入下述函数。

STR_TO_BOOL STR_TO_BOOL_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字符串
: 位
: 位

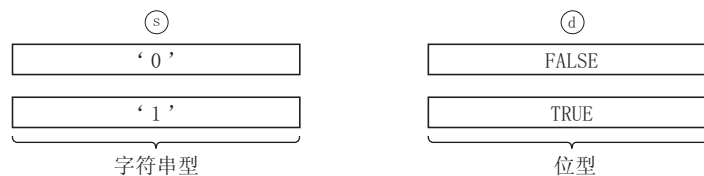
★ 功能

运算处理

将⑤中输入的字符串型的数据转换为位型的数据后通过④进行输出。

输入值为‘0’的情况下, 对 FALSE 进行输出。

输入值为‘0’以外的情况下, 对 TRUE 进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

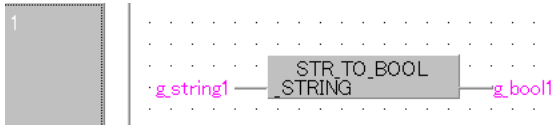
不存在 STR_TO_BOOL(_E) 相关的运算出错。

程序示例

以下为将⑤中输入的字符串型的数据转换为位型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (STR_TO_BOOL)

[结构体梯形图]

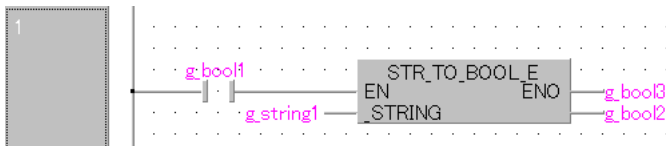


[ST]

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

(b) 带 EN/ENO 的函数 (STR_TO_BOOL_E)

[结构体梯形图]



[ST]

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

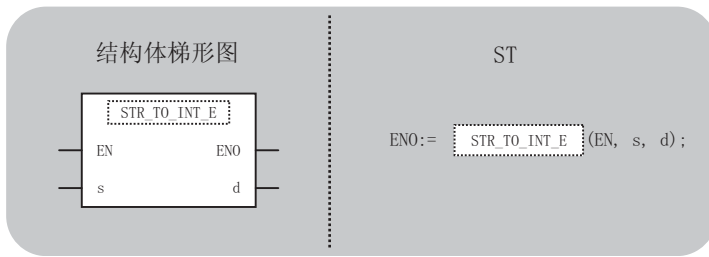
5.1.28 字符串型→字[带符号]型、双字[带符号]型转换

STR_TO_INT(_E), STR_TO_DINT(_E)

Basic High performance Universal L CPU

STR_TO_INT(_E)
STR_TO_DINT(_E)

_E : 带 EN/ENO



中放入下述函数。

STR_TO_INT	STR_TO_INT_E
STR_TO_DINT	STR_TO_DINT_E

输入自变量, EN: 执行条件(TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态(TRUE: 正常 FALSE: 异常)
d: 输出

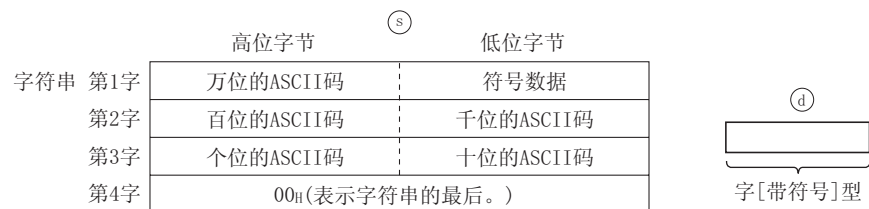
: 位
: 字符串
: 位
: 字[带符号]、双字[带符号]

★ 功能

运算处理

(1) STRING_TO_INT, STRING_TO_INT_E

(a) 将⑤中输入的字符串型的数据转换为字[带符号]型的数据后通过④进行输出。



(b) 至⑤的输入值为字符串型的数据值且在下述范围内。
ASCII码的范围为“30h”~“39h”, “20h”, “2Dh”, “00h”。
字符串型的数据值的范围为“-32768 ~ 32767”。

(2) STRING_TO_DINT, STRING_TO_DINT_E

(a) 将⑤中输入的字符串型的数据转换为双字[带符号]型的数据后通过④进行输出。



(b) 至⑤的输入值为字符串型的数据值且在下述范围内
 ASCII 码的范围为 “30H” ~ “39H”，“20H”，“2DH”，“00H”。
 字符串型的数据值的范围为 “-2147483648 ~ 2147483647”。

运算结果

(1) 无 EN/ENO 的函数
 运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

(2) 带 EN/ENO 的函数
 执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE (无运算出错)	运算输出值
	FALSE (有运算出错) *1	不定值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
 在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

在以下情况下将变为运算出错状态。

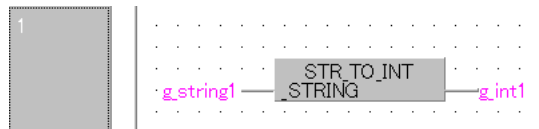
- 输入的值为 ASCII 码且在 “30H” ~ “39H”，“20H”，“2DH”，“00H” 的范围以外时。
(出错代码：4100)
- 输入的值为 ASCII 码且超出了下述范围时。
(出错代码：4100)
STR_TO_INT(_E)：“-32768 ~ 32767”
STR_TO_DINT(_E)：“-2147483648 ~ 2147483647”

程序示例

- (1) 以下为将⑤中输入的字符串型的数据转换为字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (STR_TO_INT)

[结构体梯形图]

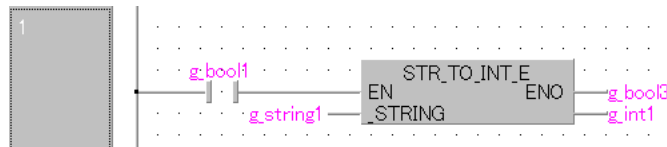


[ST]

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

(b) 带 EN/ENO 的函数 (STR_TO_INT_E)

[结构体梯形图]



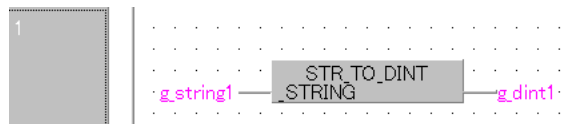
[ST]

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

- (2) 以下为将⑤中输入的字符串型的数据转换为双字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (STR_TO_DINT)

[结构体梯形图]



[ST]

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

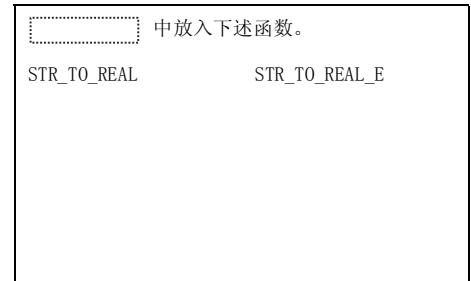
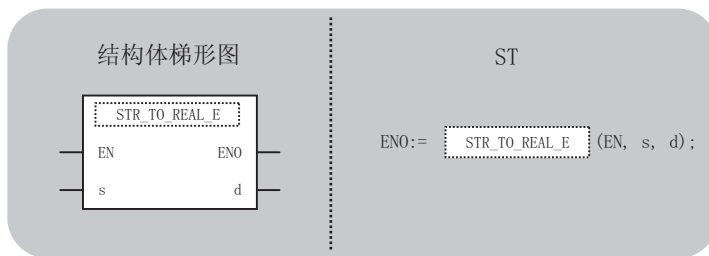
5.1.29 字符串型→单精度实数型转换

STR_TO_REAL(_E)

Basic High performance Universal L CPU

STR_TO_REAL(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 字符串
 : 位
 : 单精度实数

★ 功能

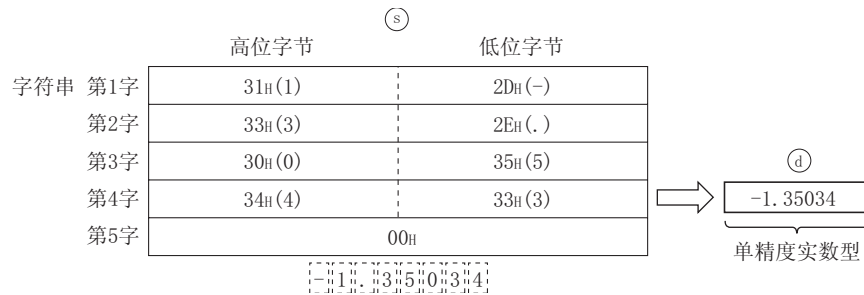
运算处理

(1) 将⑤中输入的字符串型（小数点格式 / 指数格式）的数据转换为单精度实数型的数据后通过④进行输出。

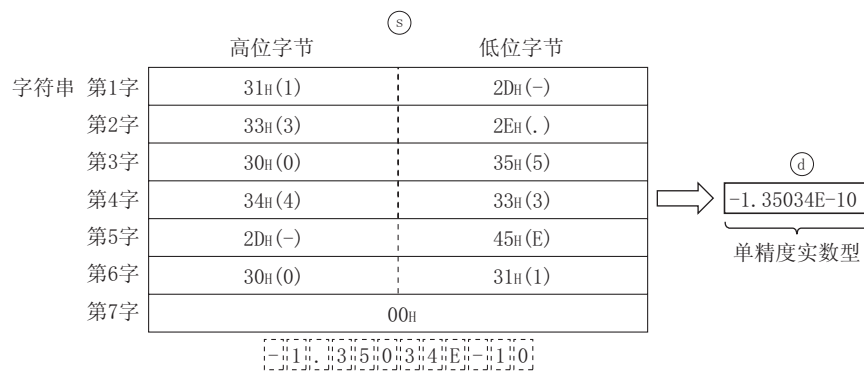


(2) 字符串型数据可以转换为小数点格式或指数格式。

(a) 小数点格式的情况下

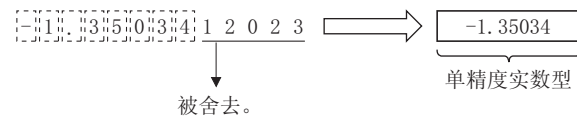


(b) 指数格式的情况下

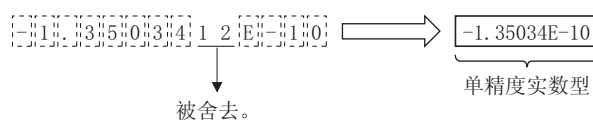


(3) 对于字符串型数据，在除符号、小数点、指数部分以外的6位数有效，从第7位及以后在转换时将被舍去。

(a) 小数点格式的情况下



(b) 指数格式的情况下



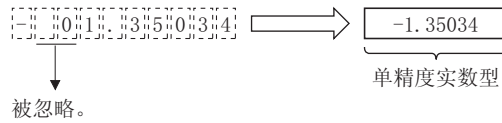
(4) 在小数点格式中如果将符号指定为“2BH”(+)，或者将符号省略时将作为正值进行转换。此外，如果在符号中指定“2DH”(-)则作为负值进行转换。

(5) 在指数格式中将指数部分的符号指定为“2BH”(+)，或者将符号省略时将作为正值进行转换。

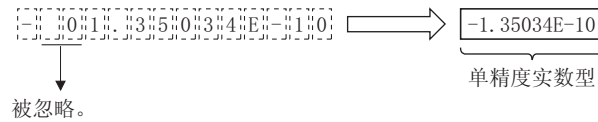
在指数部分的符号中指定“2DH”(-)时将作为负值进行转换。

- (6) 在字符串型数据中，最初的“0”以外的数值前面存在有“20H”（空格）或者“30H”（0）的情况下，在转换时“20H”、“30H”将被忽略。

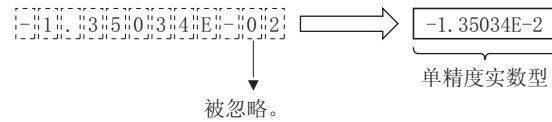
(a) 小数点格式的情况下



(b) 指数格式的情况下



- (7) 在字符串型数据（指数格式）中“E”与数值之间存在有“30H”（0）的情况下，在转换时将忽略“30H”。



- (8) 在字符串中包含有“20H”（空格）的情况下，在转换时将忽略“20H”。

- (9) 字符串型数据最多可输入 24 个字符。

字符串中的“20H”（空格）、“30H”（0）也作为 1 个字符进行计数。

- (10) 至⑤的输入值为字符串型的数据值且在下述范围内。

为 ASCII 码且范围为“30H”～“39H”、“45H”、“2BH”、“2DH”、“2EH”、“20H”、“00H”。

运算结果

- (1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE (无运算出错)	运算输出值
	FALSE (有运算出错) *1	不定值
FALSE (停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

! 出错

在以下情况下将变为运算出错状态。

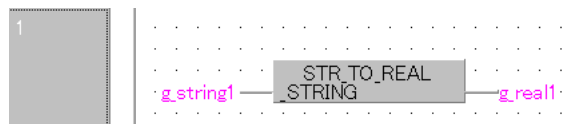
- 整数部分、小数部分中存在有除 30H ~ 39H 以外的字符时。 (出错代码 : 4100)
- 2EH 有 2 个以上时。 (出错代码 : 4100)
- 指数部分中存在有除 “45H(E)、2Ch(+)”、“45H(E)、2Dh(.)” 以外的字符时, 或者有多个指数部分时。 (出错代码 : 4100)
- 转换后的数据为 0, 或者超出了 $\pm 1.17549 \times 10^{-38} \sim \pm 3.40282 \times 10^{+38}$ 的范围时 (出错代码 : 4100)
- 字符数为 0, 或者超过了 24 个字符时。 (出错代码 : 4100)

程序示例

以下为将 ⑤ 中输入的字符串型的数据转换为单精度实数型的数据后通过 ④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (STR_TO_REAL)

[结构体梯形图]

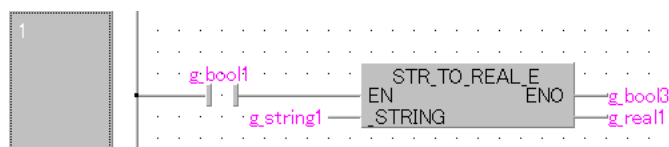


[ST]

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

(b) 带 EN/ENO 的函数 (STR_TO_REAL_E)

[结构体梯形图]



[ST]

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

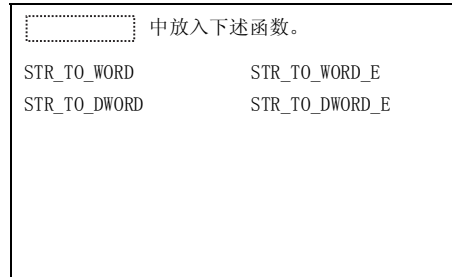
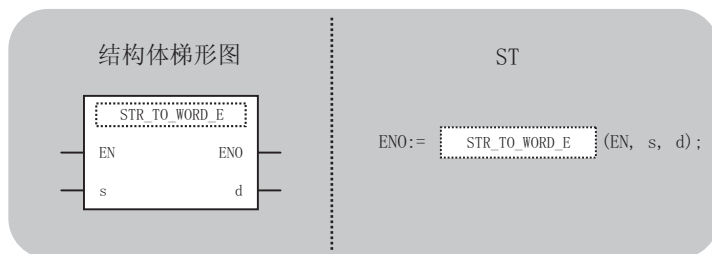
5.1.30 字符串型→字[无符号]/位串[16位]型、双字[无符号]/位串[32位]型转换

STR_TO_WORD(_E), STR_TO_DWORD(_E)

Basic High performance Universal L CPU

STR_TO_WORD(_E)
STR_TO_DWORD(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

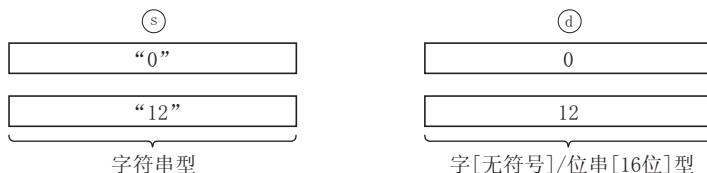
: 位
: 字符串
: 位
: 字 [无符号]/位串 [16位]、双字 [无符号]/位串 [32位]

★ 功能

运算处理

(1) STR_TO_WORD, STR_TO_WORD_E

将⑤中输入的字符串型的数据转换为字[无符号]/位串[16位]型的数据后通过⑥进行输出。

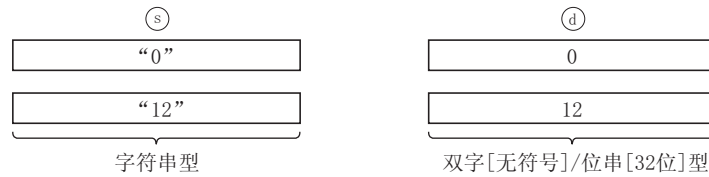


(2) STR_TO_DWORD, STR_TO_DWORD_E

将⑤中输入的字符串型的数据转换为双字 [无符号] / 位串 [32 位] 型的数据后通过④进行输出。

输入值为 FALSE 的情况下，以双字 [无符号] / 位串 [32 位] 型的数据值对 0H 进行输出。

输入值为 TRUE 的情况下，以双字 [无符号] / 位串 [32 位] 型的数据值对 1H 进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

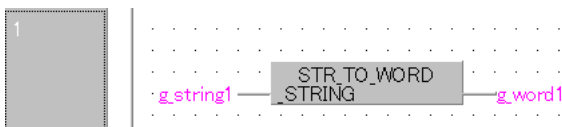
不存在 STR_TO_WORD(E)、STR_TO_DWORD(E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的字符串型的数据转换为字 [无符号]/ 位串 [16 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (STR_TO_WORD)

[结构体梯形图]

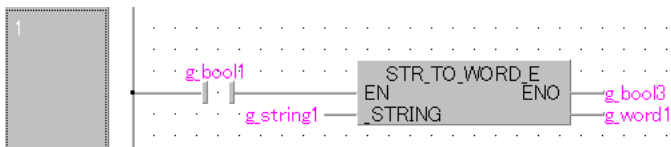


[ST]

```
g_word1 := STR_TO_WORD (g_string1);
```

(b) 带 EN/ENO 的函数 (STR_TO_WORD_E)

[结构体梯形图]



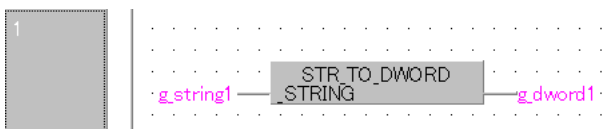
[ST]

```
g_bool3 := STR_TO_WORD_E(g_bool1, g_string1, g_word1);
```

- (2) 以下为将⑤中输入的字符串型的数据转换为双字 [无符号]/ 位串 [32 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (STR_TO_DWORD)

[结构体梯形图]



[ST]

```
g_dword1 := STR_TO_DWORD (g_string1);
```

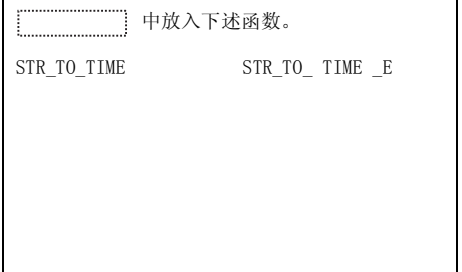
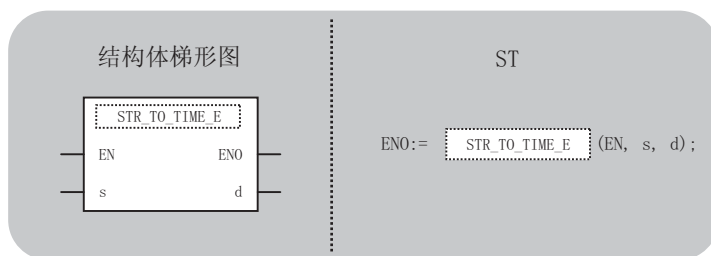
5.1.31 字符串型→时间型转换

STR_TO_TIME(_E)

Basic High performance Universal L CPU

STR_TO_TIME(_E)

_E : 带 EN/ENO



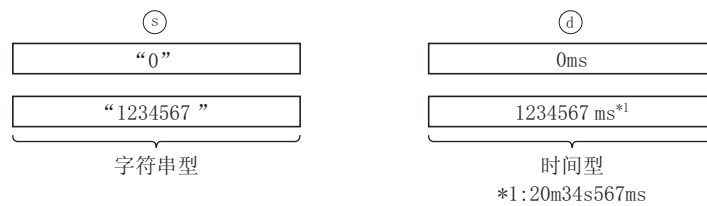
输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 字符串
 : 位
 : 时间

★ 功能

运算处理

将⑤中输入的字符串型的数据转换为时间型的数据后通过④进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *2	不定值

*2：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

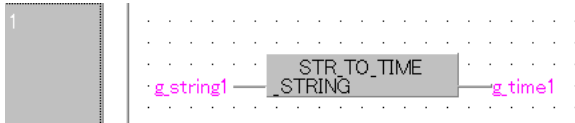
不存在 STR_TO_TIME(_E) 相关的运算出错。

程序示例

以下为将⑤中输入的字符串型的数据转换为时间型的数据后通过④进行输出的程序。

- (a) 无 EN/ENO 的函数 (STR_TO_TIME)

[结构体梯形图]

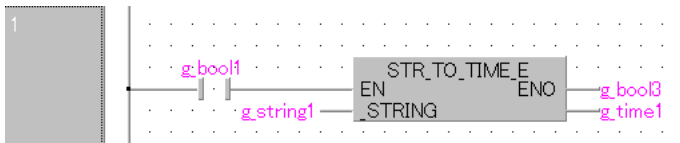


[ST]

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

- (b) 带 EN/ENO 的函数 (STR_TO_TIME_E)

[结构体梯形图]



[ST]

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

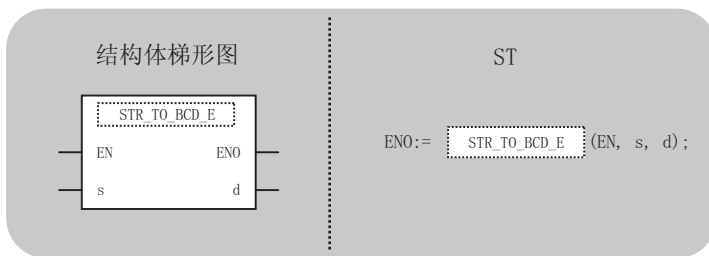
5.1.32 字符串型→BCD型转换

STR_TO_BCD(_E)

Basic High performance Universal L CPU

STR_TO_BCD(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

中放入下述函数。
 STR_TO_BCD STR_TO_BCD_E

: 位
 : 字符串
 : 位
 : 字 [无符号] / 位串 [16 位]、双字 [无符号] / 位串 [32 位]

★ 功能

运算处理

- (1) 将⑤中输入的字符串型的数据转换为 BCD 型的数据后通过④进行输出。



- (2) 字符串中包含有“20H”（空格）的情况下，在转换时将忽略“20H”。
- (3) 对于字符串型数据，最多可输入 50 个字符。
 字符串中的“20H”（空格）、“30H”（0）也作为 1 个字符进行计数器。
- (4) 至⑤的输入值为字符串型的数据值且在下述范围内。
 为 ASCII 码且范围为“30H”～“39H”、“45H”、“2BH”、“2DH”、“2EH”、“20H”、“00H”。

运算结果

- (1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE(无运算出错)	运算输出值
	FALSE(有运算出错) *1	不定值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

在下述情况下将变为运算出错状态。

- ASCII 码输入了超出 “30H” ~ “39H” 的范围的字符串的情况下。

(出错代码 : 4100)

- ASCII 码中输入了超出 “30H” ~ “39H” 及 “9999” 的情况下。

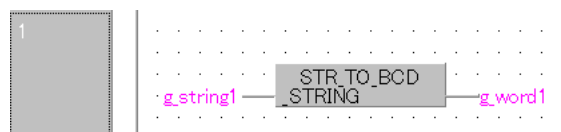
(出错代码 : 4100)

程序示例

以下为将③中输入的字符串型的数据转换为 BCD 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (STR_TO_BCD)

[结构体梯形图]

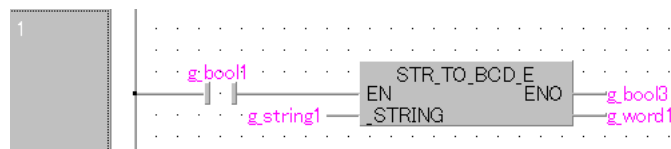


[ST]

```
g_word1 := STR_TO_BCD (g_string1);
```

(b) 带 EN/ENO 的函数 (STR_TO_BCD_E)

[结构体梯形图]



[ST]

```
g_bool3 := STR_TO_BCD_E (g_bool1, g_string1, g_word1);
```

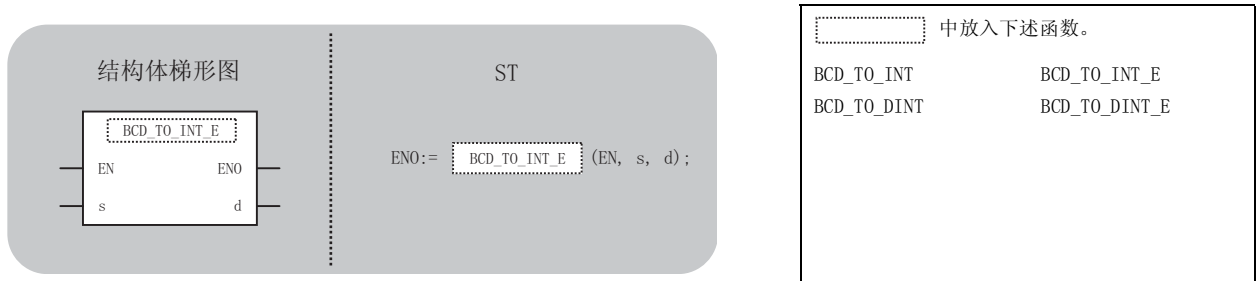
5.1.33 BCD型→字[带符号]型、双字[带符号]型转换

BCD_TO_INT(_E), BCD_TO_DINT(_E)

Basic High performance Universal L CPU

BCD_TO_INT(_E)
BCD_TO_DINT(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

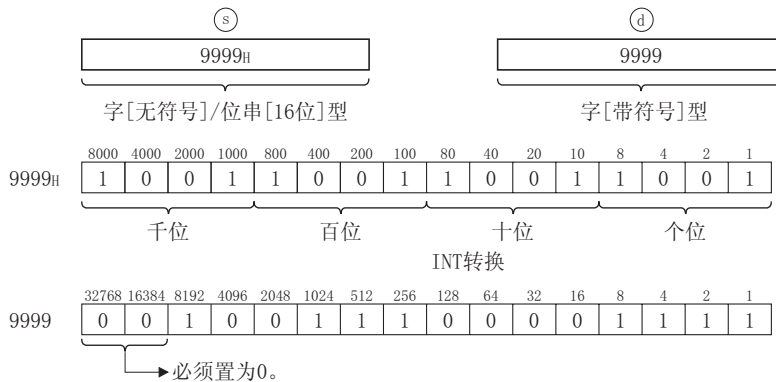
: 位
: 字 [无符号] / 位串 [16 位], 双字 [无符号] / 位串 [32 位]
: 位
: 字 [带符号]、双字 [带符号]

★ 功能

运算处理

(1) BCD_TO_INT, BCD_TO_INT_E

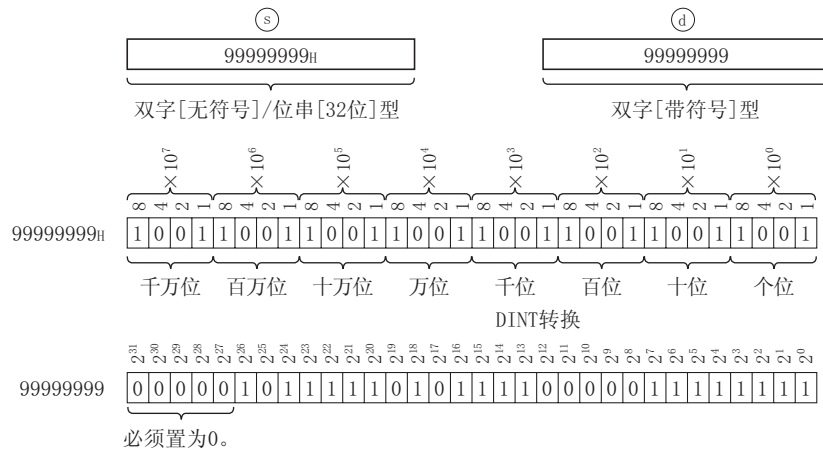
(a) 将Ⓢ中输入的BCD型的数据转换为字[带符号]型的数据后通过Ⓣ进行输出。



(b) 至Ⓢ的输入值为字[无符号]/位串[16位]型的数据值且在0h~9999h(各位数的范围为0~9)的范围内。

(2) BCD_TO_DINT, BCD_TO_DINT_E

(a) 将⑤中输入的 BCD 型的数据转换为双字 [带符号] 型的数据后通过④进行输出。



(b) 至⑤的输入值为双字 [无符号] / 位串 [32 位] 型的数据值且在 0H ~ 99999999H (各位数的范围为 0 ~ 9) 的范围内。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

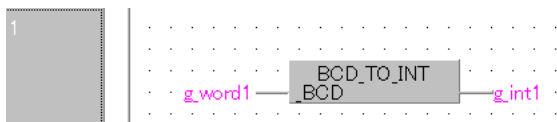
不存在 BCD_TO_INT(_E)、BCD_TO_DINT(_E) 相关的运算出错。

程序示例

(1) 以下为将⑤中输入的 BCD 型的数据转换为字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (BCD_TO_INT)

[结构体梯形图]

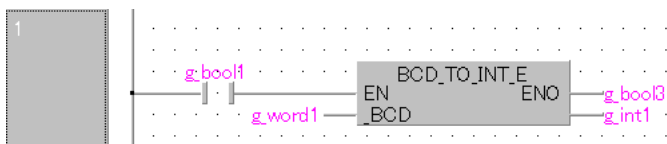


[ST]

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

(b) 带 EN/ENO 的函数 (BCD_TO_INT_E)

[结构体梯形图]



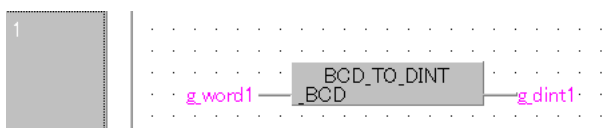
[ST]

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

(2) 以下为将⑤中输入的 BCD 型的数据转换为双字 [带符号] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (BCD_TO_DINT)

[结构体梯形图]



[ST]

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

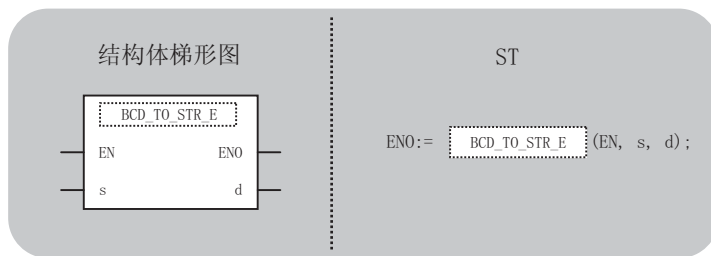
5.1.34 BCD 型→字符串型转换

BCD_TO_STR(_E)

Basic High performance Universal L CPU

BCD_TO_STR(_E)

_E : 带 EN/ENO



中放入下述函数。

```
BCD_TO_STR      BCD_TO_STR_E
```

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字 [无符号] / 位串 [16 位], 双字 [无符号] / 串 [32 位]
: 位
: 字符串

★ 功能

运算处理

将③中输入 BCD 型的数据转换为字符串型的数据后通过④进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

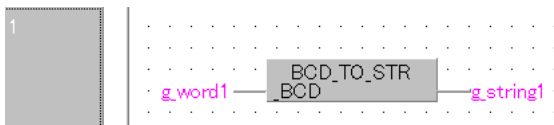
不存在 BCD_TO_STR(E) 相关的运算出错。

程序示例

以下为将⑤中输入的字[无符号]/位串[16位]型的数据转换为字符串型的数据后通过④进行输出的程序。

- (a) 无 EN/ENO 的函数 (BCD_TO_STR)

[结构体梯形图]

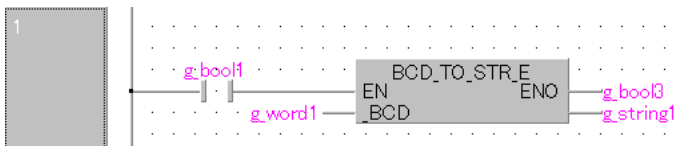


[ST]

```
g_string1 := BCD_TO_STR (g_word1);
```

- (b) 带 EN/ENO 的函数 (BCD_TO_STR_E)

[结构体梯形图]



[ST]

```
g_bool3 := BCD_TO_STR_E (g_bool1, g_word1, g_string1);
```

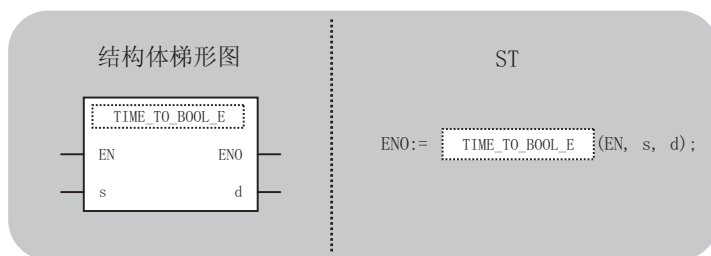
5.1.35 时间型→位型转换

TIME_TO_BOOL(_E)

Basic High performance Universal L CPU

TIME_TO_BOOL(_E)

_E : 带 EN/ENO



中放入下述函数。

TIME_TO_BOOL TIME_TO_BOOL_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 时间
 : 位
 : 位

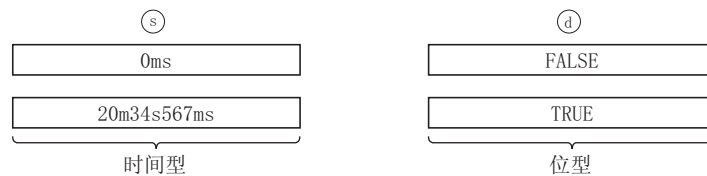
★ 功能

运算处理

将⑤中输入的时间型的数据转换为位型的数据后通过④进行输出。

输入值为 0ms 的情况下, 对 FALSE 进行输出。

输入值为 0ms 以外的情况下, 对 TRUE 进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

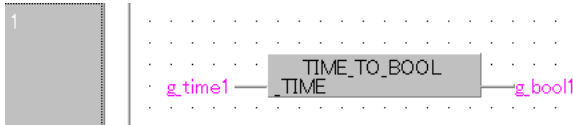
不存在 TIME_TO_BOOL(E) 相关的运算出错。

程序示例

以下为将⑤中输入的时间型的数据转换为位型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (TIME_TO_BOOL)

[结构体梯形图]

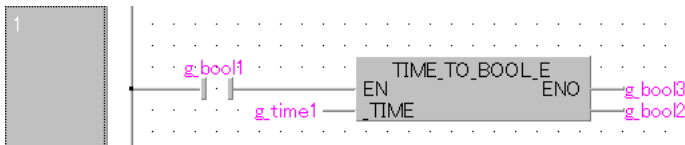


[ST]

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

(b) 带 EN/ENO 的函数 (TIME_TO_BOOL_E)

[结构体梯形图]



[ST]

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

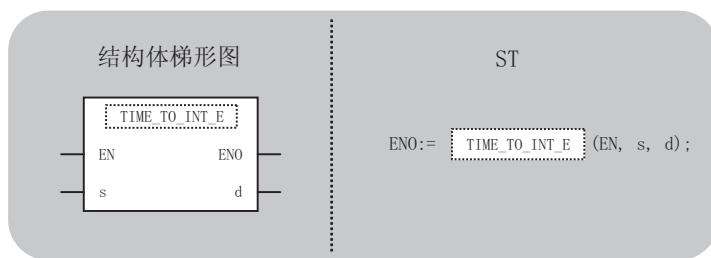

5.1.36 时间型→字[带符号]型、双字[带符号]型转换

TIME_TO_INT(_E), TIME_TO_DINT(_E)

Basic High performance Universal L CPU

TIME_TO_INT(_E)
TIME_TO_DINT(_E)

_E : 带 EN/ENO



中放入下述函数。

TIME_TO_INT TIME_TO_INT_E
TIME_TO_DINT TIME_TO_DINT_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常执行 FALSE: 异常或者停止)
 d: 输出

: 位
: 时间
: 位
: 字 [带符号]、双字 [带符号]

★ 功能

运算处理

(1) TIME_TO_INT, TIME_TO_INT_E

将⑤中输入的时间型的数据转换为字[带符号]型的数据后通过④进行输出。



(2) TIME_TO_DINT, TIME_TO_DINT_E

将⑤中输入的时间型的数据转换为双字[带符号]型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

出错

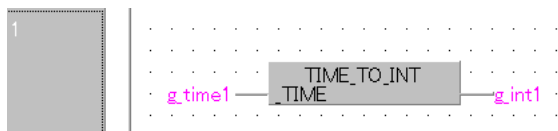
不存在 TIME_TO_INT(E)、TIME_TO_DINT(E) 相关的运算出错。

程序示例

(1) 以下为将⑤中输入的时间型的数据转换为字[带符号]型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (TIME_TO_INT)

[结构体梯形图]

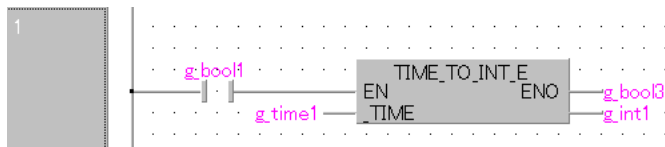


[ST]

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

(b) 带 EN/ENO 的函数 (TIME_TO_INT_E)

[结构体梯形图]



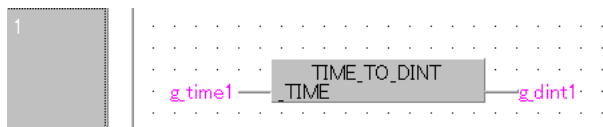
[ST]

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

(2) 以下为将⑤中输入的时间型的数据转换为双字[带符号]型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (TIME_TO_DINT)

[结构体梯形图]



[ST]

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

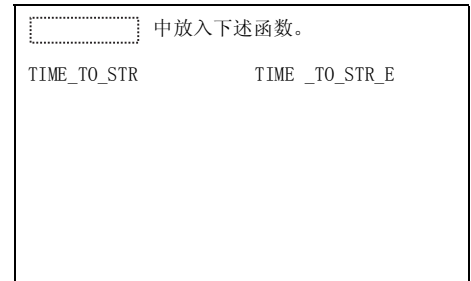
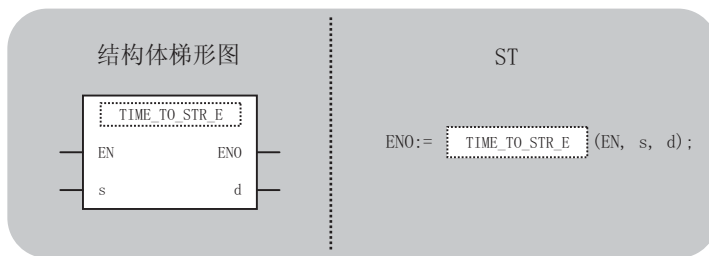
5.1.37 时间型→字符串型转换

TIME_TO_STR(_E)

Basic High performance Universal L CPU

TIME_TO_STR(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 时间
 : 位
 : 字符串

★ 功能

运算处理

将⑤中输入的时间型的数据转换为字符串型的数据后通过④进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

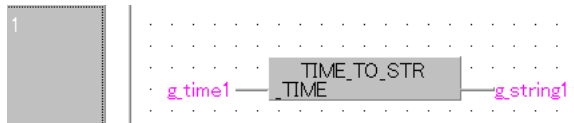
不存在 TIME_TO_STR(E) 相关的运算出错。

程序示例

以下为将⑤中输入的时间型的数据转换为字符串型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (TIME_TO_STR)

[结构体梯形图]

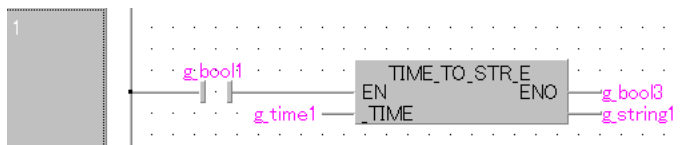


[ST]

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

(b) 带 EN/ENO 的函数 (TIME_TO_STR_E)

[结构体梯形图]



[ST]

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

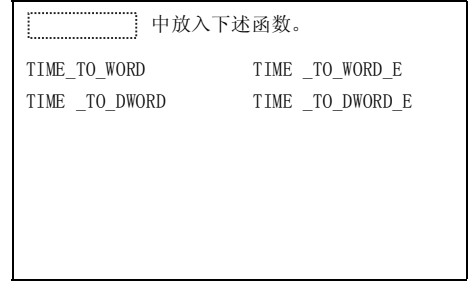
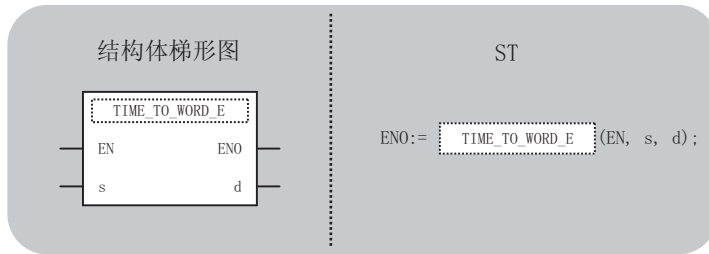
5.1.38 时间型→字 [无符号]/ 位串 [16 位] 型、 双字 [无符号]/ 位串 [32 位] 型转换

TIME_TO_WORD(_E), TIME_TO_DWORD(_E)

Basic High performance Universal L CPU

TIME_TO_WORD(_E)
TIME_TO_DWORD(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 时间
: 位
: 字 [无符号]/ 位串 [16 位]、双字 [无符号]/
位串 [32 位]

★ 功能

运算处理

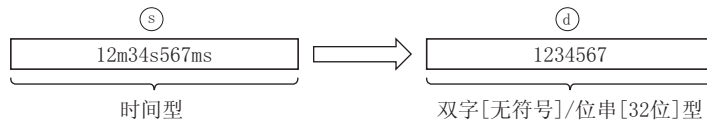
(1) TIME_TO_WORD, TIME_TO_WORD_E

将 (s) 中输入的时间型的数据转换为字 [无符号]/ 位串 [16 位] 型的数据后通过 (d) 进行输出。



(2) TIME_TO_DWORD, TIME_TO_DWORD_E

将⑤中输入的时间型的数据转换为双字[无符号]/位串[32位]型的数据后通过④进行输出。



运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

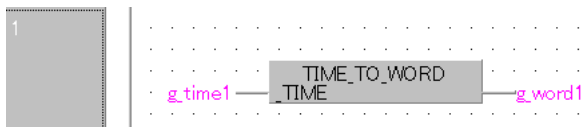
不存在 TIME_TO_WORD(_E)、TIME_TO_DWORD(_E) 相关的运算出错。

程序示例

- (1) 以下为将⑤中输入的时间型的数据转换为字 [无符号] / 位串 [16 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (TIME_TO_WORD)

[结构体梯形图]

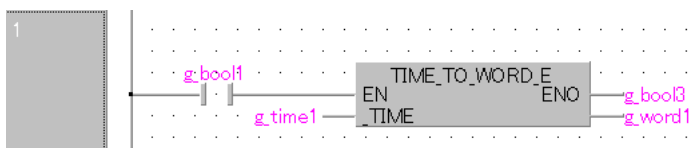


[ST]

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

(b) 带 EN/ENO 的函数 (TIME_TO_WORD_E)

[结构体梯形图]



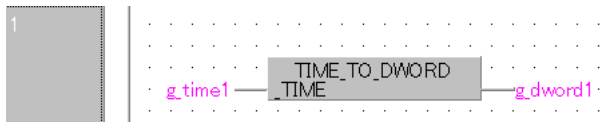
[ST]

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

- (2) 以下为将⑤中输入的时间型的数据转换为双字 [无符号] / 位串 [32 位] 型的数据后通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (TIME_TO_DWORD)

[结构体梯形图]



[ST]

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


5.2 单数值变量函数

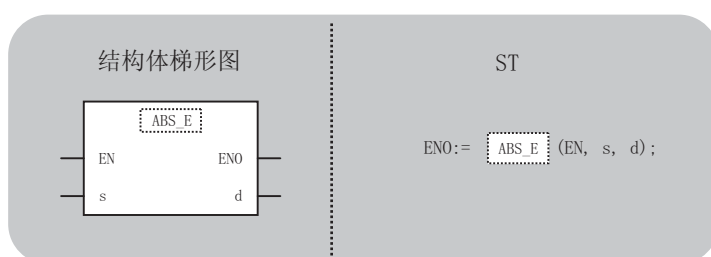
5.2.1 绝对值

ABS(_E)

Basic High performance Universal L CPU

ABS(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

中放入下述函数。

ABS ABS_E

: 位
: ANY_NUM
: 位
: ANY_NUM

★ 功能

运算处理

- (1) 将⑤中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型数据的绝对值, 以与⑤相同的数据类型通过④进行输出。
将输入值设为 A, 将运算输出值设为 B 时的情况如下所示。

$$B = |A|$$

- (2) 至⑤的输入值为字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型的数据值。

- (3) 在⑤的数据类型为字[带符号]型且输入了-32768的情况下,通过④将输出32768。
 在⑤的数据类型为双字[带符号]型且输入了-2147483648的情况下,通过④将输出2147483648。
 (不变为运算出错状态。此外,在ABS_E的情况下,将从ENO中输出TRUE。)

运算结果

- (1) 无 EN/ENO 的函数
 执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
 执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下,通过④输出的数据将变为不定值。
 在这种情况下,应执行程序处理以避免使用④中输出的数据。

出错

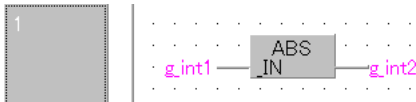
不存在 ABS(_E) 相关的运算出错。

程序示例

以下为将⑤中输入的字[带符号]型/双字[带符号]型/单精度实数型数据的绝对值,以与⑤相同的数据类型通过④进行输出的程序。

- (a) 无 EN/ENO 的函数 (ABS)

[结构体梯形图]

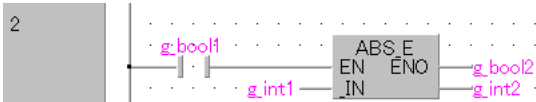


[ST]

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

- (b) 带 EN/ENO 的函数 (ABS_E)

[结构体梯形图]



[ST]

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

5.3 算术运算函数

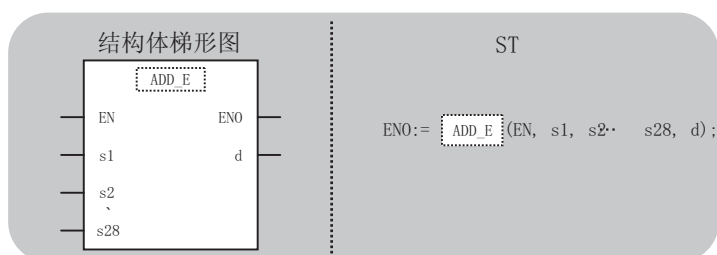
5.3.1 加法运算

ADD(_E)

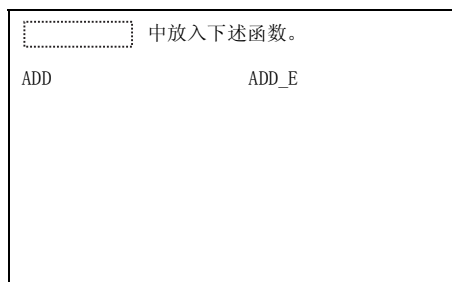
Basic High performance Universal L CPU

ADD(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件(TRUE: 执行 FALSE: 停止)
 s1 ~ s28: 输入
 输出自变量, ENO: 输出状态(TRUE: 正常 FALSE: 异常)
 d: 输出



: 位
 : ANY_NUM
 : 位
 : ANY_NUM

★ 功能

运算处理

- (1) 对①~②中输入的字[带符号]型/双字[带符号]型/单精度实数型/双精度实数型数据进行加法运算(①+②+...+②)后,将运算结果以与①~②中相同的数据类型通过③进行输出。

(例)数据类型为字[带符号]型的情况下



- (2) 至①~②的输入值为字[带符号]型/双字[带符号]型/单精度实数型/双精度实数型的数据值。
 (3) 对于③的针数,可在2~28的范围内进行变更。

(4) 运算结果中发生了下溢 / 上溢的情况下，将按以下方式通过④进行输出。

(a) 数据类型为字 [带符号] 型的情况下

即使发生了下溢 / 上溢的情况下也不会变为运算出错状态。

此外，ADD_E 的情况下，将从 ENO 中输出 TRUE。

$32767 + 2 = -32767$ 最高位为 1，因此变为负值。
(7FFFH) (0002H) (8001H)

$-32767 + (-2) = 32766$ 最高位为 0，因此变为正值。
(8000H) (FFFEH) (7FFEH)

(b) 数据类型为双字 [带符号] 型的情况下

即使发生了下溢 / 上溢的情况下，也不会变为运算出错状态。

此外，ADD_E 的情况下，将从 ENO 中输出 TRUE。

$2147483647 + 2 = -2147483647$ 最高位为 1，因此变为负值。
(7FFFFFFFH) (0002H) (80000001H)

$-2147483648 + (-2) = 2147483646$ 最高位为 0，因此变为正值。
(80000000H) (FFFEH) (7FFFFFFEH)

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④中输出的数据。

! 出错

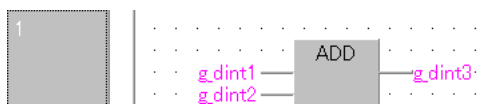
不存在 ADD(_E) 相关的运算出错。

程序示例

以下为对①, ②中输入的双字 [带符号] 型数据进行加法运算 (①+②) 后, 将运算结果以与①, ②中相同的数据类型通过③进行输出的程序。

(a) 无 EN/ENO 的函数 (ADD)

[结构体梯形图]

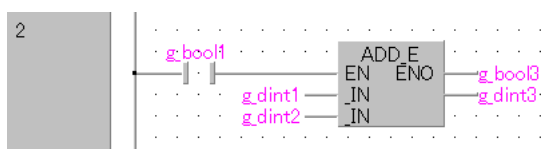


[ST]

```
g_dint3 := (g_dint1) + (g_dint2);
```

(b) 带 EN/ENO 的函数 (ADD_E)

[结构体梯形图]



[ST]

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

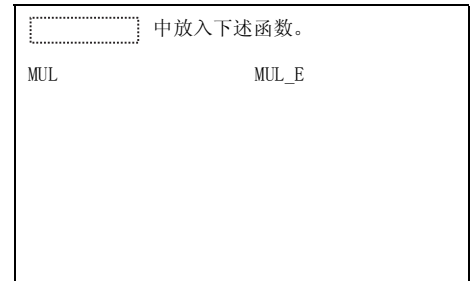
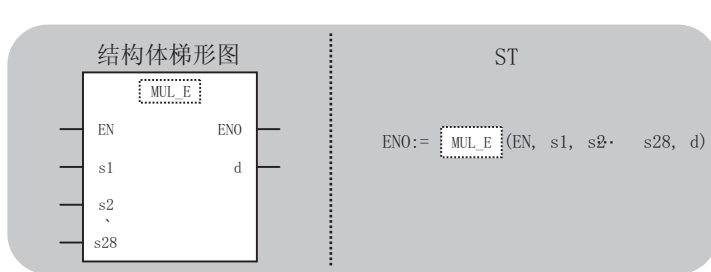
5.3.2 乘法运算

MUL(_E)

Basic High performance Universal L CPU

MUL(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1 ~ s28: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

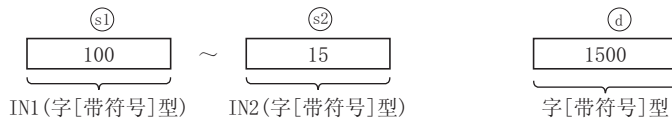
: 位
 : ANY_NUM
 : 位
 : ANY_NUM

★ 功能

运算处理

(1) 对①~②中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型数据进行乘法运算 (① × ② × ... × ②) 后, 将运算结果以与①~②中相同的数据类型通过③进行输出。

(例) 数据类型为字 [带符号] 型的情况下



(2) 至①~②的输入值为字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型的数据值。

(3) 对于④的针数, 可在 2 ~ 28 的范围内进行变更。

(4) 运算结果中发生了下溢 / 上溢的情况下，将按以下方式通过④进行输出。

(a) 数据类型为字 [带符号] 型的情况下

即使发生了下溢 / 上溢的情况下，也不会变为运算出错状态。

此外，MUL_E 的情况下，将从 ENO 中输出 TRUE。

即使运算结果超出了字 [带符号] 型数据的范围的情况下，也仍将对字 [带符号] 型的数据进行输出。

(运算结果将为 32 位数据，但输出将以删除了高 16 位后的字 [带符号] 型的数据进行输出。)

运算结果超出了字 [带符号] 型数据的范围的情况下，应通过 INT_TO_DINT 将输入值转换为双字 [带符号] 型数据后，再进行运算。

(b) 数据类型为双字 [带符号] 型的情况下

即使发生了下溢 / 上溢的情况下，也不会变为运算出错状态。

此外，MUL_E 的情况下，将从 ENO 中输出 TRUE。

即使运算结果超出了双字 [带符号] 型数据的范围的情况下，也仍将对双字 [带符号] 型的数据进行输出。

(运算结果为 64 位数据，但输出将以删除了高 32 位后的双字 [带符号] 型的数据进行输出。)

在运算结果超出了双字 [带符号] 型数据的范围的情况下，应通过 DINT_TO_REAL 将输入值转换为单精度实数型数据，再进行运算。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下，通过④输出的数据将变为不定值。

在这种情况下，应执行程序处理以避免使用④中输出的数据。

☒ 要点

运算结果超出了数据类型的范围的情况下，应对输入值的数据类型进行转换之后再执行运算。

出错

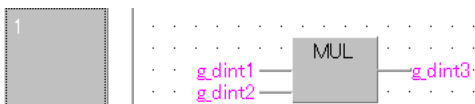
不存在 MUL(_E) 相关的运算出错。

程序示例

以下为对①, ②中输入的双字 [带符号] 型数据进行乘法运算 (① × ②) 后, 以与①, ②中相同的数据类型通过③进行输出的程序。

(a) 无 EN/ENO 的函数 (MUL)

[结构体梯形图]

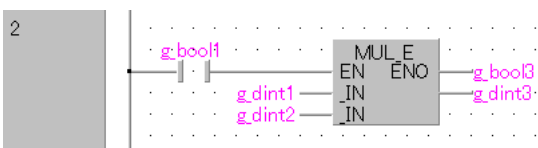


[ST]

```
g_dint3 := (g_dint1) * (g_dint2);
```

(b) 带 EN/ENO 的函数 (MUL_E)

[结构体梯形图]



[ST]

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

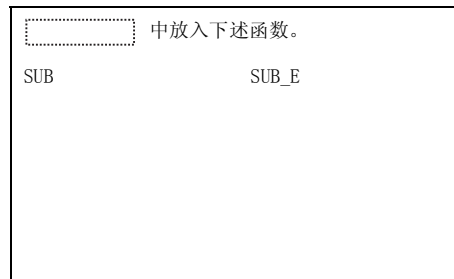
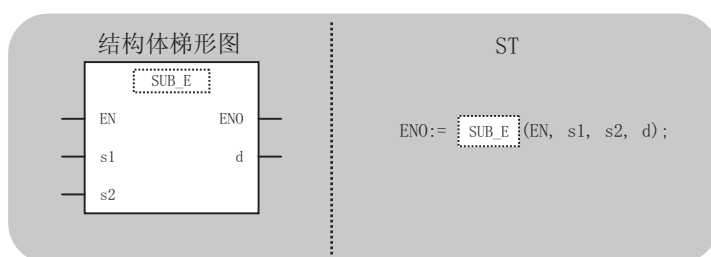

5.3.3 减法运算

SUB(_E)

Basic High performance Universal L CPU

SUB(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : ANY_NUM
 : 位
 : ANY_NUM

★ 功能

运算处理

- (1) 对①, ②中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型数据进行减法运算 (① - ②) 后, 将运算结果以与③中相同的数据类型通过④进行输出。

(例) 数据类型为字 [带符号] 型的情况下



- (2) 至①, ②中的输入值为字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型的数据值。

(3) 运算结果中发生了下溢 / 上溢的情况下, 将按以下方式通过④进行输出。

(a) 数据类型为字 [带符号] 型的情况下

即使发生了下溢 / 上溢的情况下, 也不会变为运算出错状态。

此外, SUB_E 的情况下, 将从 ENO 中输出 TRUE。

$32767 - (-2) = -32767$ 最高位为 1, 因此变为负值。

(7FFFH) (FFFEH) (8001H)

$-32767 - 2 = 32766$ 最高位为 0, 因此变为正值。

(8000H) (0002H) (7FFEh)

(b) 数据类型为双字 [带符号] 型的情况下

即使发生了下溢 / 上溢的情况下, 也不会变为运算出错状态。

此外, SUB_E 的情况下, 将从 ENO 中输出 TRUE。

$2147483647 - (-2) = -2147483647$ 最高位为 1, 因此变为负值。

(7FFFFFFFH) (FFFEH) (80000001H)

$-2147483648 - 2 = 2147483646$ 最高位为 0, 因此变为正值。

(80000000H) (0002H) (7FFFFFFEH)

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

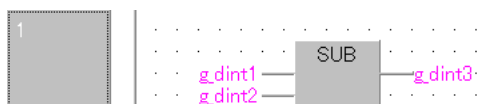
不存在 SUB(_E) 相关的运算出错。

程序示例

以下为对①, ②中输入的双字 [带符号] 型数据进行减法运算 (①-②) 后, 以与①, ②中相同的数据类型通过③进行输出的程序。

(a) 无 EN/ENO 的函数 (SUB)

[结构体梯形图]

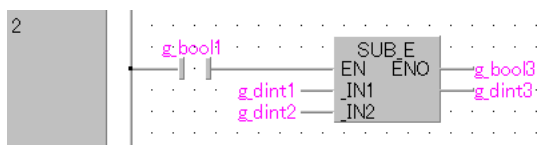


[ST]

```
g_dint3 := (g_dint1) - (g_dint2);
```

(b) 带 EN/ENO 的函数 (SUB_E)

[结构体梯形图]



[ST]

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

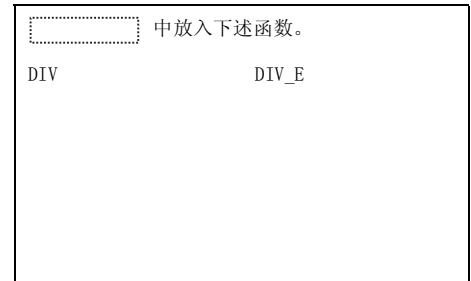
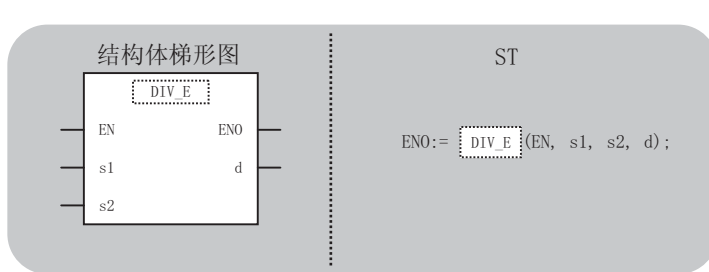
5.3.4 除法运算

DIV(_E)

Basic High performance Universal L CPU

DIV(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

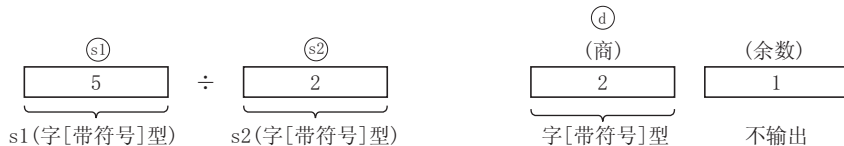
: 位
 : ANY_NUM
 : 位
 : ANY_NUM

★ 功能

运算处理

(1) 对①, ②中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型数据进行除法运算 (① ÷ ②) 后, 将运算结果的商以与①, ②中相同的数据类型通过④进行输出。

(例) 数据类型为字 [带符号] 型的情况下



(2) 至①, ②的输入值为字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型的数据值。(但是, 被输入至②中的值为除 0 以外的值。)

运算结果

- (1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE (无运算出错)	运算输出值
	FALSE (有运算出错) *1	不定值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

在以下情况下将变为运算出错状态。

- 被输入至②中的值为0时。(除数为0)

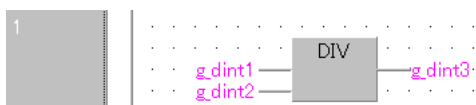
(出错代码：4100)

程序示例

以下为对①, ②中输入的双字 [带符号] 型数据进行除法运算 (① ÷ ②) 后, 将运算结果的商以与①, ②中相同的数据类型通过③进行输出的程序。

(a) 无 EN/ENO 的函数 (DIV)

[结构体梯形图]

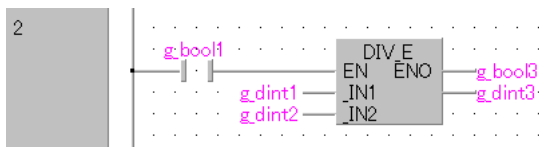


[ST]

```
g_dint3 := (g_dint1) / (g_dint2);
```

(b) 带 EN/ENO 的函数 (DIV_E)

[结构体梯形图]



[ST]

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

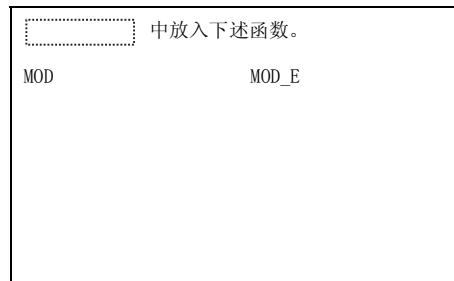
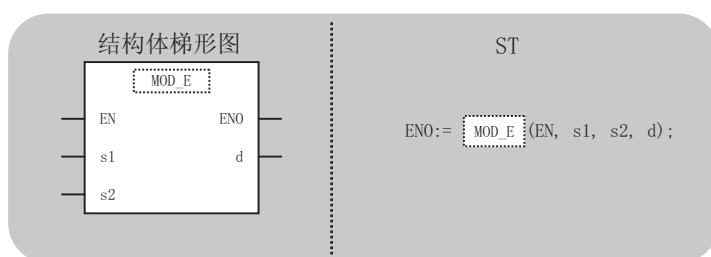
5.3.5 余数

MOD(_E)

Basic High performance Universal L CPU

MOD(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

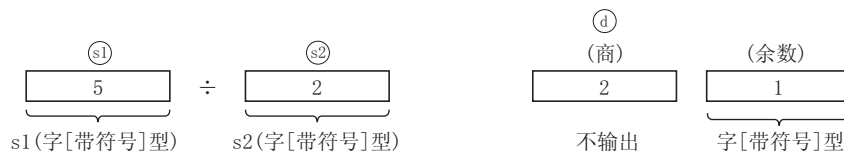
: 位
 : ANY_INT
 : 位
 : ANY_INT

★ 功能

运算处理

- (1) 对①, ②中输入的字 [带符号] 型 / 双字 [带符号] 型数据进行除法运算 (① ÷ ②) 后, 将运算结果的余数以与①, ②中相同的数据类型通过④进行输出。

(例) 数据类型为字 [带符号] 型的情况下



- (2) 至①, ②的输入值为字 [带符号] 型 / 双字 [带符号] 型的数据值。(但是, 被输入至②中的值为除 0 以外的值。)

运算结果

- (1) 无 EN/ENO 的函数
运算结果如下所示。

运算结果	④
无运算出错	运算输出值
有运算出错	不定值

- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE(无运算出错)	运算输出值
	FALSE(有运算出错) *1	不定值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

在以下情况下将变为运算出错状态。

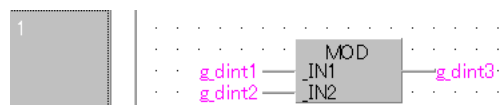
- 被输入至②中的值为 0 时。（除数为 0）（出错代码：4100）

程序示例

以下为对①, ②中输入的双字 [带符号] 型数据进行除法运算 (① ÷ ②) 后, 将运算结果的余数以与①, ②中相同的数据类型通过③进行输出的程序。

(a) 无 EN/ENO 的函数 (MOD)

[结构体梯形图]

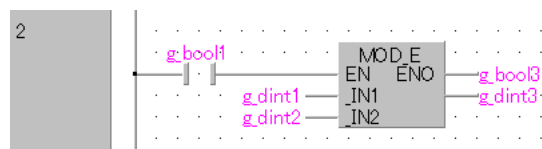


[ST]

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

(b) 带 EN/ENO 的函数 (MOD_E)

[结构体梯形图]



[ST]

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

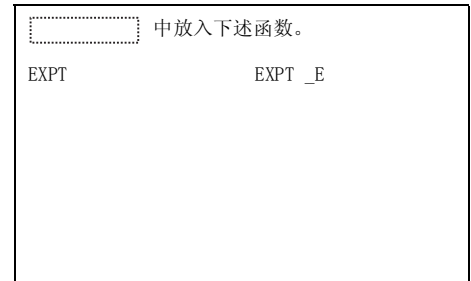
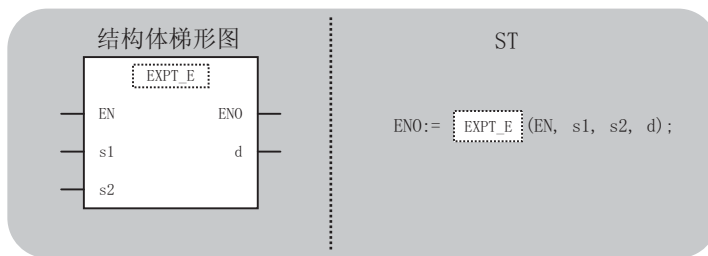
5.3.6 幂

EXPT(_E)

Basic High performance Universal L CPU

EXPT(_E)

_E : 带 EN/ENO



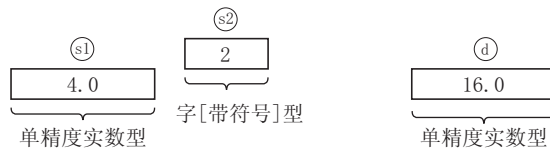
输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : ANY_REAL
 : ANY_NUM
 : 位
 : ANY_REAL

★ 功能

运算处理

对①中输入的单精度实数型 / 双精度实数型数据进行幂运算②后, 将运算结果通过③进行输出。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

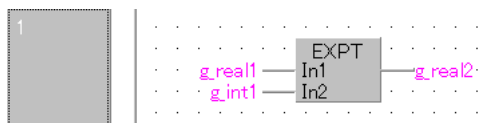
不存在 EXPT(_E) 相关的运算出错。

程序示例

以下为进行幂运算后, 将运算结果以与①, ②中相同的数据类型通过④进行输出的程序。

- (a) 无 EN/ENO 的函数 (EXPT)

[结构体梯形图]

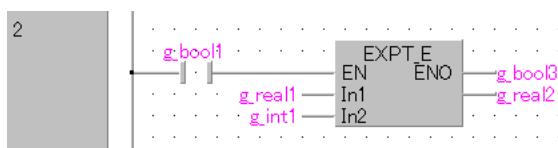


[ST]

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

- (b) 带 EN/ENO 的函数 (EXPT_E)

[结构体梯形图]



[ST]

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

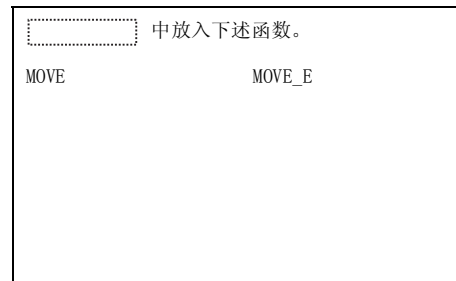
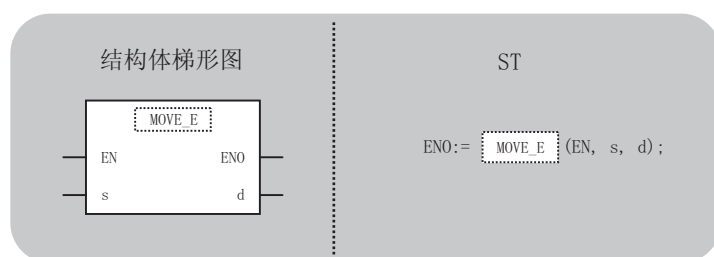
5.3.7 代入

MOVE(_E)

Basic High performance Universal L CPU

MOVE(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)

s: 输入

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)

d: 输出

: 位

: ANY

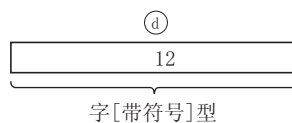
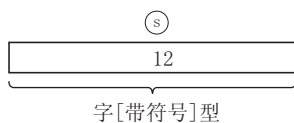
: 位

: ANY

★ 功能

运算处理

将③代入到④中。



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

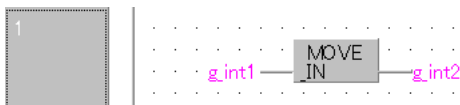
不存在 MOVE(_E) 相关的运算出错。

程序示例

以下为将⑤中输入的字[带符号]型数据代入到④中的程序。

- (a) 无 EN/ENO 的函数 (MOVE)

[结构体梯形图]

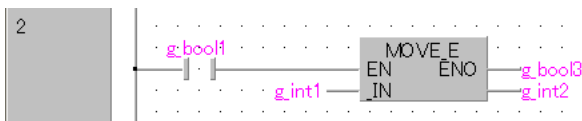


[ST]

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

- (b) 带 EN/ENO 的函数 (MOVE_E)

[结构体梯形图]



[ST]

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

5.4 位型布尔函数

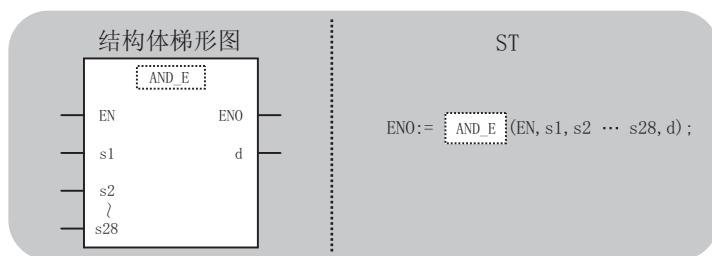
5.4.1 逻辑积、逻辑和、排他的逻辑和、逻辑否定

AND(_E), OR(_E), XOR(_E), NOT(_E)

Basic High performance Universal L CPU

AND(_E)
OR(_E)
XOR(_E)
NOT(_E)

(_E : 带 EN/ENO)



中放入下述函数。

AND	AND_E
OR	OR_E
XOR	XOR_E
NOT	NOT_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s1 ~ s28: 输入
(NOT(_E) 仅 s1)
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

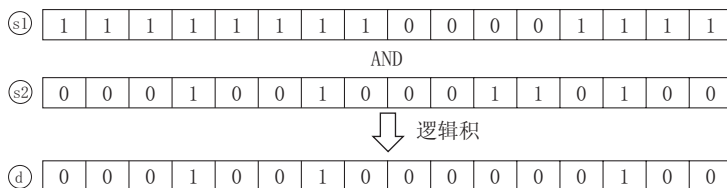
: 位
: ANY_BIT
: 位
: ANY_BIT

★ 功能

运算处理

(1) AND, AND_E

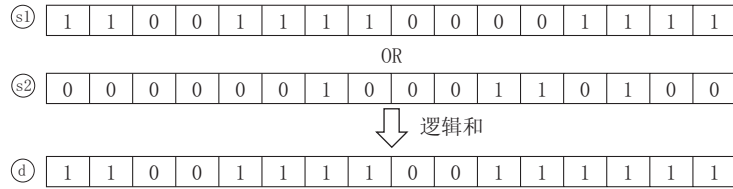
(a) 对①~②中输入的位型 / 字 [无符号] / 位串 [16 位] 型 / 双字 [无符号] / 位串 [32 位] 型数据逐位进行逻辑积后, 将运算结果以与①~②中相同的数据类型通过③进行输出。
(例) 数据类型为字 [无符号] / 位串 [16 位] 型的情况下



(b) 对于 s 的针数，可在 2 ~ 28 的范围内进行变更。

(2) OR, OR_E

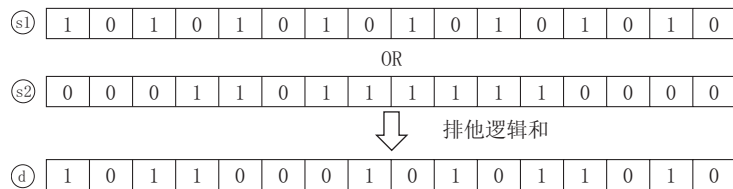
(a) 对①~②中输入的位型/字[无符号]/位串[16位]型/双字[无符号]/位串[32位]型数据逐位进行逻辑和后，将运算结果以与①~②中相同的数据类型通过④进行输出。
(例)数据类型为字[无符号]/位串[16位]型的情况下



(b) 对于 s 的针数，可在 2 ~ 28 的范围内进行变更。

(3) XOR, XOR_E

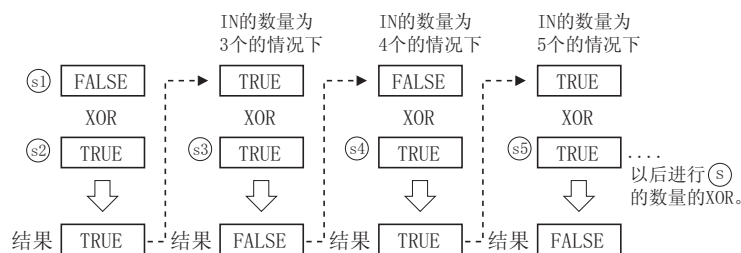
(a) 对①~②中输入的位型/字[无符号]/位串[16位]型/双字[无符号]/位串[32位]型数据逐位进行排他逻辑和后，将运算结果以与①~②中相同的数据类型通过④进行输出。
(例)数据类型为字[无符号]/位串[16位]型的情况下



(b) 对于 s 的针数，可在 2 ~ 28 的范围内进行变更。

(c) s 为 3 个以上的情况下，将①与②进行 XOR 的结果，用③进行 XOR。此外，存在有④的情况下，将对③进行 XOR 的结果用④进行 XOR。以后，对⑤，⑥进行 s 的数量的 XOR。

(例)数据类型为位的情况下



(4) NOT, NOT_E

对①中输入的位型/字[无符号]/位串[16位]型/双字[无符号]/位串[32位]型数据逐位进行逻辑否定后,将运算结果以与①中相同的数据类型通过④进行输出。

(例)数据类型为字[无符号]/位串[16位]型的情况下

①	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	1
NOT																
④	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0

(5) 至①~②的输入值为位型/字[无符号]/位串[16位]型/双字[无符号]/位串[32位]型的数据值。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下,通过④输出的数据将变为不定值。
在这种情况下,应执行程序处理以避免使用④中输出的数据。

出错

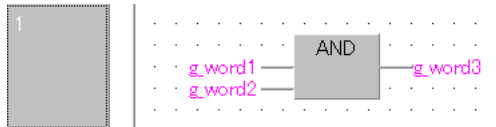
不存在 AND(_E)、OR(_E)、XOR(_E)、NOT(_E) 相关的运算出错。

程序示例

- (1) 以下为对①~③中输入的位型 / 字 [无符号] / 位串 [16 位] 型 / 双字 [无符号] / 位串 [32 位] 型数据逐位进行逻辑积后，将运算结果以与①~③中相同的数据类型通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (AND)

[结构体梯形图]

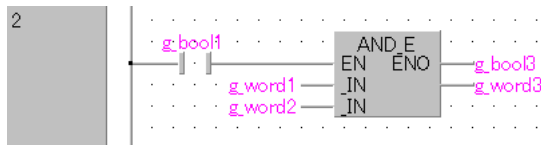


[ST]

```
g_word3 :=(g_word1) AND (g_word2);
或者
g_word3 :=(g_word1) & (g_word2);
```

(b) 带 EN/ENO 的函数 (AND_E)

[结构体梯形图]



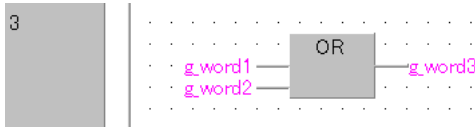
[ST]

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

- (2) 以下为对①～③中输入的位型 / 字 [无符号] / 位串 [16 位] 型 / 双字 [无符号] / 位串 [32 位] 型数据逐位进行逻辑和后，将运算结果以与①～③中相同的数据类型通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (OR)

[结构体梯形图]



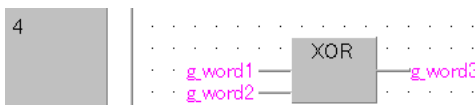
[ST]

```
g_word3 :=(g_word1) OR (g_word2);
```

- (3) 以下为对①～③中输入的位型 / 字 [无符号] / 位串 [16 位] 型 / 双字 [无符号] / 位串 [32 位] 型数据逐位进行排他的逻辑和后，将运算结果以与①～③中相同的数据类型通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (XOR)

[结构体梯形图]



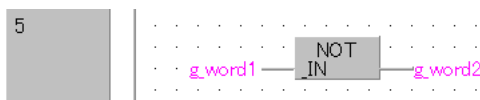
[ST]

```
g_word3 :=(g_word1) XOR (g_word2);
```

- (4) 以下为对①中输入的位型 / 字 [无符号] / 位串 [16 位] 型 / 双字 [无符号] / 位串 [32 位] 型数据逐位进行逻辑否定后，将运算结果以与①中相同的数据类型通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (NOT)

[结构体梯形图]



[ST]

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

5.5 选择函数

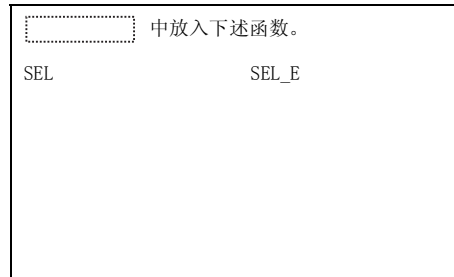
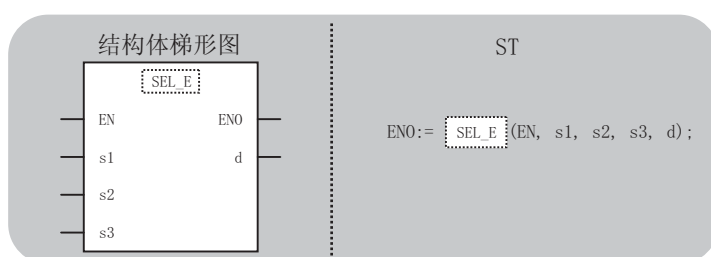
5.5.1 选择值

SEL(_E)

Basic High performance Universal L CPU

SEL(_E)

(_E : 带 EN/ENO)



输入自变量, EN:	执行条件 (TRUE: 执行 FALSE: 停止)	: 位
s1:	输出条件 (TRUE: s3 输出 FALSE: s2 输出)	: 位
s2:	输入	: ANY
s3:	输入	: ANY
输出自变量, ENO:	输出状态 (TRUE: 正常 FALSE: 异常)	: 位
d:	输出	: ANY

★ 功能

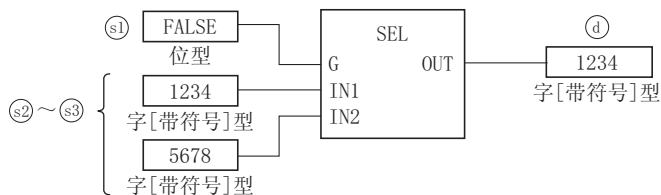
运算处理

根据输入到①中的输入值, 将输入到②, ③中的值中的某一个, 以与②, ③相同的数据类型输出到④中。

①的输入值为 FALSE 的情况下, 通过②对④的输入值进行输出。

①的输入值为 TRUE 的情况下, 通过③对④的输入值进行输出。

(例) ②, ③的数据类型为字[带符号]型的情况下



运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

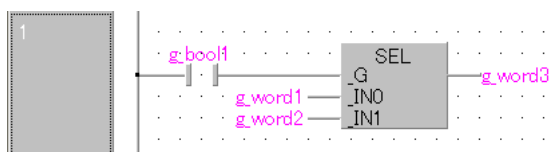
不存在 SEL(_E) 相关的运算出错。

程序示例

以下为根据输入到①中的输入值，将输入到②，③中的值中的某一个以与②，③相同的数据类型通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (SEL)

[结构体梯形图]

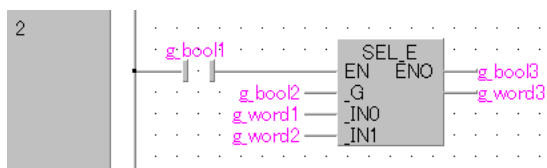


[ST]

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

(b) 带 EN/ENO 的函数 (SEL_E)

[结构体梯形图]



[ST]

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

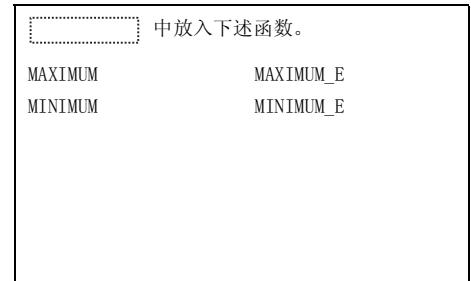
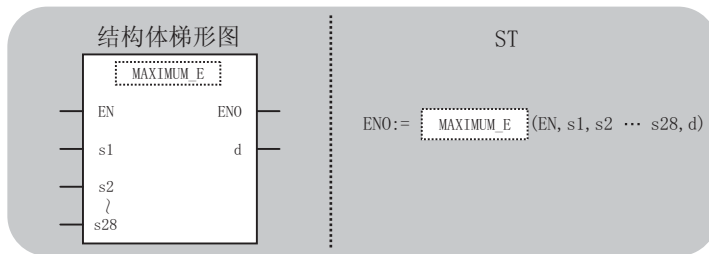
5.5.2 最大值、最小值选择

MAXIMUM(_E), MINIMUM(_E)

Basic High performance Universal L CPU

MAXIMUM(_E)
MINIMUM(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s1 ~ s28: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

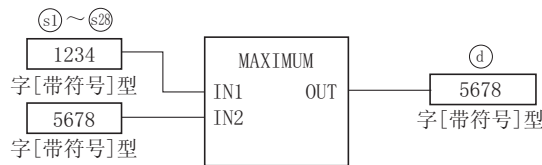
: 位
: ANY_SIMPLE
: 位
: ANY_SIMPLE

★ 功能

运算处理

(1) MAXIMUM, MAXIMUM_E

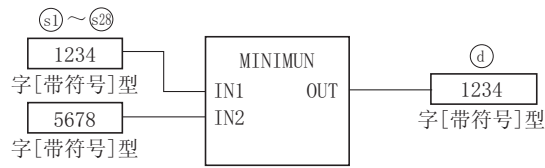
将①~③中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型数据的最大值, 以与①~③中相同的数据类型通过④进行输出。
(例) 数据类型为字 [带符号] 型的情况下



(2) MINIMUM, MINIMUM_E

将①~②中输入的字[带符号]型/双字[带符号]型/单精度实数型数据的最小值, 以与①~②中相同的数据类型通过④进行输出。

(例) 数据类型为字[带符号]型的情况下



(3) 至①~②中的输入值为字[带符号]型/双字[带符号]型/单精度实数型的数据值。

(4) 对于s的针数, 可在2~28的范围内进行变更。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

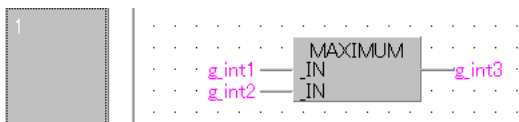
不存在 MAXIMUM(E)、MINIMUM(E) 相关的运算出错。

程序示例

- (1) 以下为将 (s1)~(s2) 中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型数据的最大值, 以与 (s1)~(s2) 中相同的数据类型通过 (d) 进行输出的程序。

(a) 无 EN/ENO 的函数 (MAXIMUM)

[结构体梯形图]

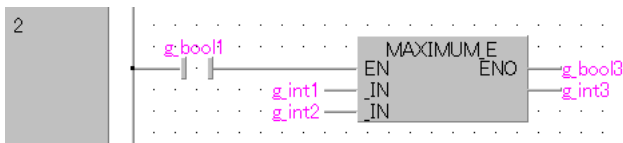


[ST]

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

(b) 带 EN/ENO 的函数 (MAXIMUM_E)

[结构体梯形图]



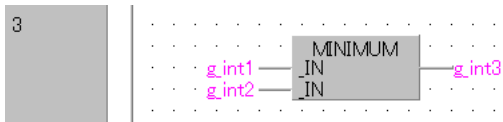
[ST]

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

- (2) 以下为将 (s1)~(s2) 中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型数据的最小值, 以与 (s1)~(s2) 中相同的数据类型通过 (d) 进行输出的程序。

(a) 无 EN/ENO 的函数 (MINIMUM)

[结构体梯形图]



[ST]

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

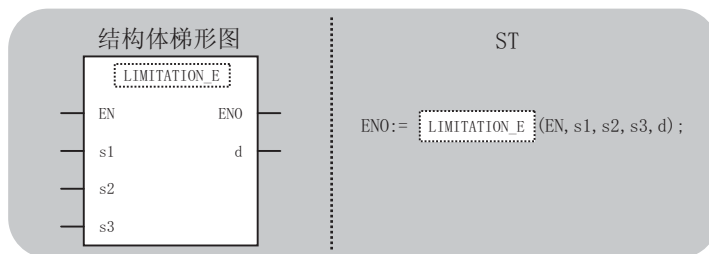

5.5.3 上下限极限控制

LIMITATION(_E)

Basic High performance Universal L CPU

LIMITATION(_E)

_E : 带 EN/ENO



中放入下述函数。
LIMITATION LIMITATION_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s1: 下限极限值 (最小输出限界值)
s2: 通过上下限极限控制进行控制的输入值
s3: 上限极限值 (最大输出限界值)
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

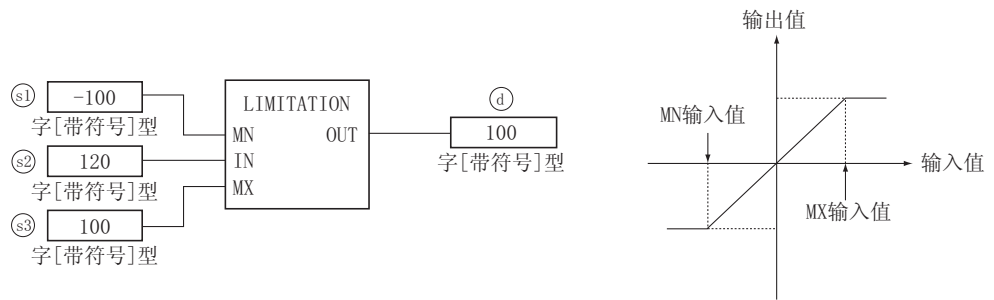
: 位
: ANY_SIMPLE
: ANY_SIMPLE
: ANY_SIMPLE
: 位
: ANY_SIMPLE

★ 功能

运算处理

- (1) 根据①, ②, ③中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型数据, 以与④中相同的数据类型通过④进行输出。
 - (a) ②的输入值 > ③的输入值的情况下, 将③的输入值通过④进行输出。
 - (b) ②的输入值 < ①的输入值的情况下, 将①的输入值通过④进行输出。
 - (c) ①的输入值 \leq ②输入值 \leq ③的输入值的情况下, 将②的输入值通过④进行输出。

(例) 数据类型为字 [带符号] 型的情况下



(2) 至①, ②, ③的输入值为字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型的数据值。(但是, ①的输入值 < ③的输入值)

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

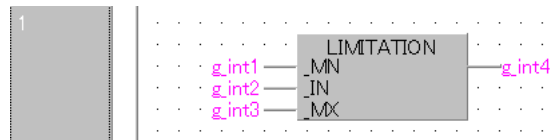
不存在 LIMITATION(E) 相关的运算出错。

程序示例

- (1) 以下为根据 (s1), (s2), (s3) 中输入的字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型数据, 以与 (s1), (s2), (s3) 中相同的数据类型通过 (d) 进行输出的程序。

(a) 无 EN/ENO 的函数 (LIMITATION)

[结构体梯形图]

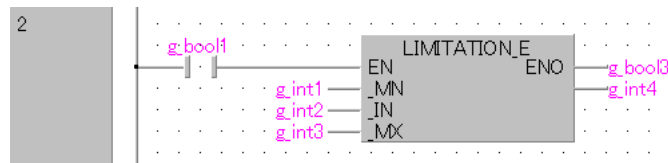


[ST]

```
g_int4 := LIMITATION (g_int1, g_int2, g_int3);
```

(b) 带 EN/ENO 的函数 (LIMITATION_E)

[结构体梯形图]



[ST]

```
g_bool3 := LIMITATION_E (g_bool1, g_int1, g_int2, g_int3, g_int4);
```

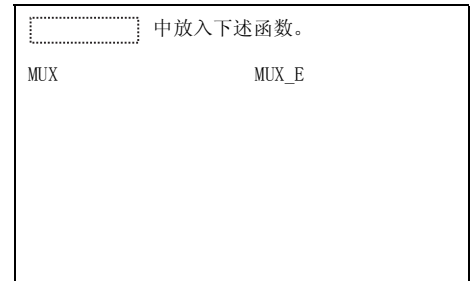
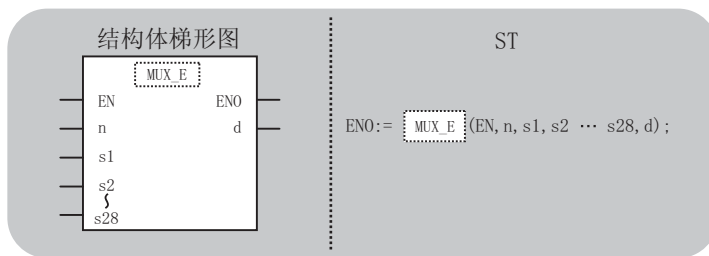
5.5.4 多路复用器

MUX(_E)

Basic High performance Universal L CPU

MUX(_E)

(_E : 带 EN/ENO)



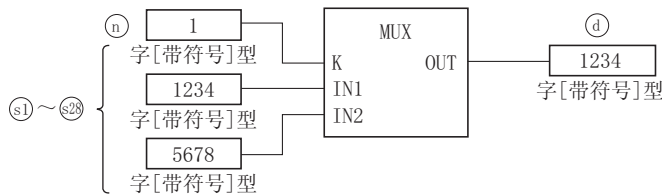
输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 n: 输出值选择
 s1 ~ s28: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 字 [带符号]
 : ANY
 : 位
 : ANY

★ 功能

运算处理

- 根据输入到①中的输入值，将输入到②~③中的值中的某一个，以与②~③中相同的数据类型通过④进行输出。
 ①的输入值为 1 的情况下，将输入到②中的值通过④进行输出。
 ①的输入值为 n 的情况下，将输入到③中的值通过④进行输出。
 (例) 数据类型为字 [带符号] 型的情况下



- 输入到①中的值超出了 s 的针数范围的情况下，将通过④输出不定值。
 (不变为运算出错状态。此外，MUX_E 的情况下，将通过 ENO 输出 FALSE。)

- (3) 至①的输入值为字 [带符号] 型的数据值且在 1 ~ 28 的范围内。(但是, 在 s 的针数的范围内。)
- (4) 至①~⑳的输入值为位型 / 字 [带符号] 型 / 双字 [带符号] 型 / 字 [无符号] / 位串 [16 位] 型 / 双字 [无符号] / 位串 [32 位] 型 / 单精度实数型 / 双精度实数型 / 字符串型的数据值。
- (5) 对于 s 的针数, 可在 2 ~ 28 的范围内进行变更。

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

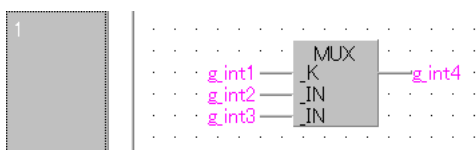
不存在 MUX(_E) 相关的运算出错。

程序示例

以下为根据输入到①中的输入值，将输入到②～③中的值中的某一个，以与②～③相同的数据类型通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (MUX)

[结构体梯形图]

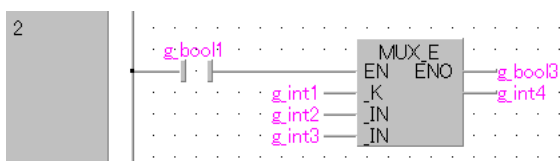


[ST]

```
g_int4 := MUX (g_int1, g_int2, g_int3);
```

(b) 带 EN/ENO 的函数 (MUX_E)

[结构体梯形图]



[ST]

```
g_bool3 := MUX_E (g_bool1, g_int1, g_int2, g_int3, g_int4);
```

5.6 比较函数

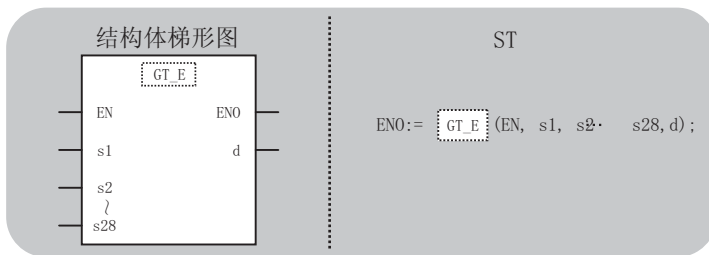
5.6.1 比较

GT(_E), GE(_E), EQ(_E), LE(_E), LT(_E), NE(_E)

Basic High performance Universal L CPU

GT(_E)
GE(_E)
EQ(_E)
LE(_E)
LT(_E)
NE(_E)

(_E : 带 EN/ENO)



中放入下述函数。

GT	GT_E
GE	GE_E
EQ	EQ_E
LE	LE_E
LT	LT_E
NE	NE_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s1 ~ s28: 输入
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出 (TRUE: 真值 FALSE: 假值)

: 位
: ANY_SIMPLE
: 位
: 位

★ 功能

运算处理

- (1) 对输入到①~⑭中的输入值进行比较运算后, 将运算结果以与①~⑭中相同的数据类型通过⑮进行输出。

- (a) 对 GT, GT_E(>) [$s_1 > s_2$] & [$s_2 > s_3$] & ... & [$s_{(n-1)} > s_{(n)}$] 进行比较。
- 所有的 $s_{(n-1)} > s_{(n)}$ 时, 对 TRUE 进行输出。
 - 某一个 $s_{(n-1)} \leq s_{(n)}$ 时, 对 FALSE 进行输出。
- (b) 对 GE, GE_E(\geq) [$s_1 \geq s_2$] & [$s_2 \geq s_3$] & ... & [$s_{(n-1)} \geq s_{(n)}$] 进行比较。
- 所有的 $s_{(n-1)} \geq s_{(n)}$ 时, 对 TRUE 进行输出。
 - 某一个 $s_{(n-1)} < s_{(n)}$ 时, 对 FALSE 进行输出。
- (c) EQ, EQ_E(=) [$s_1 = s_2$] & [$s_2 = s_3$] & ... & [$s_{(n-1)} = s_{(n)}$] 进行比较。
- 所有的 $s_{(n-1)} = s_{(n)}$ 时, 对 TRUE 进行输出。
 - 某一个 $s_{(n-1)} \neq s_{(n)}$ 时, 对 FALSE 进行输出。
- (d) LE, LE_E(\leq) [$s_1 \leq s_2$] & [$s_2 \leq s_3$] & ... & [$s_{(n-1)} \leq s_{(n)}$] 进行比较。
- 所有的 $s_{(n-1)} \leq s_{(n)}$ 时, 对 TRUE 进行输出。
 - 某一个 $s_{(n-1)} > s_{(n)}$ 时, 对 FALSE 进行输出。
- (e) LT, LT_E(<) [$s_1 < s_2$] & [$s_2 < s_3$] & ... & [$s_{(n-1)} < s_{(n)}$] 进行比较。
- 所有的 $s_{(n-1)} < s_{(n)}$ 时, 对 TRUE 进行输出。
 - 某一个 $s_{(n-1)} \geq s_{(n)}$ 时, 对 FALSE 进行输出。
- (f) NE, NE_E(< >) [$s_1 \neq s_2$] 进行比较。
- $s_1 \neq s_2$ 时对 TRUE 进行输出
 - $s_1 = s_2$ 时对 FALSE 进行输出。
- (2) 对于 s 的针数, 可在 2 ~ 28 的范围内进行变更。(但是, 对于 NE(_E) 的 s 针, 固定为 s_1, s_2 。)

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过 \textcircled{d} 输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	\textcircled{d}
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过 \textcircled{d} 输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用 \textcircled{d} 中输出的数据。

出错

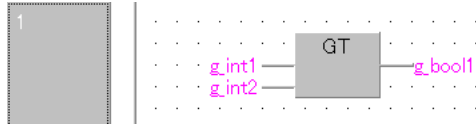
不存在 GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)、NE(_E) 相关的运算出错。

程序示例

以下为对输入到①, ②中的输入值进行比较运算后, 将运算结果通过③进行输出的程序。

(a) 无 EN/ENO 的函数 (GT)

[结构体梯形图]

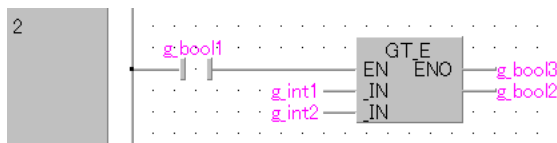


[ST]

```
g_bool1 := (g_int1) > (g_int2);
```

(b) 带 EN/ENO 的函数 (GT_E)

[结构体梯形图]



[ST]

```
g_bool3 := GT_E (g_bool1, g_int1, g_int2, g_bool2);
```

5.7 字符串函数

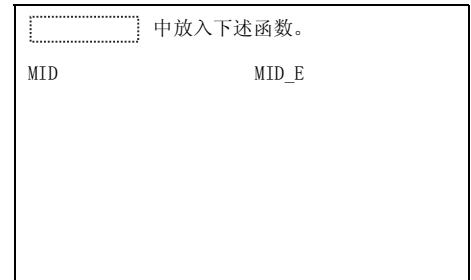
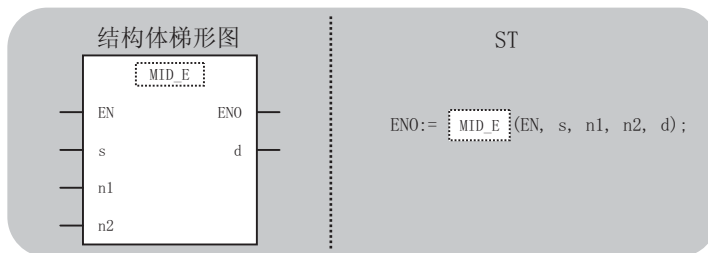
5.7.1 字符串的截取

MID(_E)

 Basic high performance Universal L CPU

MID(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 n1: 截取字符数指定
 n2: 截取起始位置指定

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

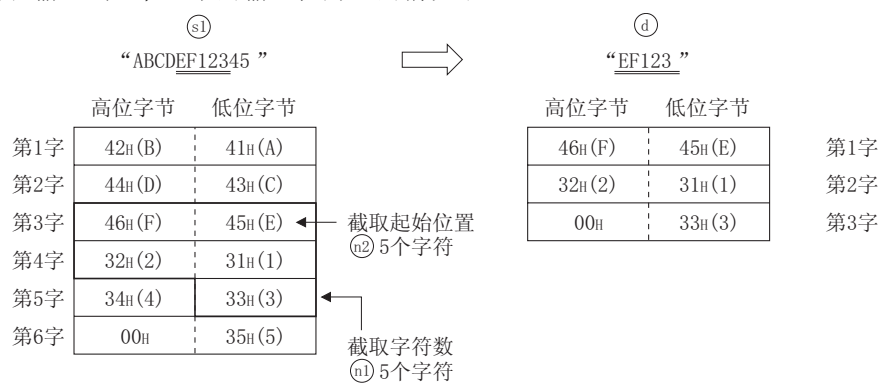
: 位
 : 字符串
 : 字 [带符号]
 : 字 [带符号]
 : 位
 : 字符串

★ 功能

运算处理

- (1) 从输入到⑤中的字符串的任意位置开始, 将指定字符数的数据通过④进行输出。
 截取的字符数通过输入到①中的输入值进行指定。
 截取的字符串的起始位置通过输入到②中的输入值进行指定。

(例) 输入到①, ②中的输入值为 5 的情况下



(2) 至①的输入值为字符串型的数据值且在 0 ~ 255 个字节的范围内。

(3) 至②的输入值为字 [带符号] 型的数据值且在 0 ~ 255 的范围内。

(但是, 限于输入到①中的字符串的字符数以内。)

(4) 至③的输入值为字 [带符号] 型的数据值且在 1 ~ 255 的范围内。

(但是, 限于输入到①中的字符串的字符数以内。)

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

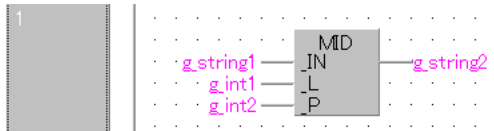
不存在 MID(_E) 相关的运算出错。

程序示例

以下为从输入到⑤中的字符串的任意位置开始，将指定字符数的数据通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (MID)

[结构体梯形图]

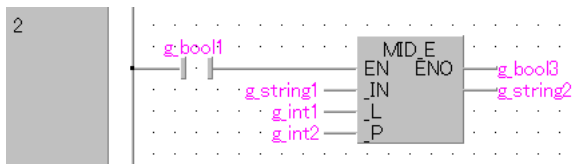


[ST]

```
g_string2:=MID(g_string1, g_int1, g_int2);
```

(b) 带 EN/ENO 的函数 (MID_E)

[结构体梯形图]



[ST]

```
g_bool3 := MID_E(g_bool1, g_string1, g_int1, g_int2, g_string2);
```

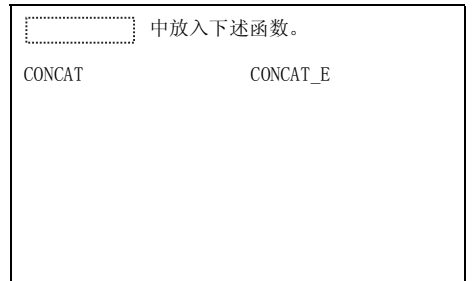
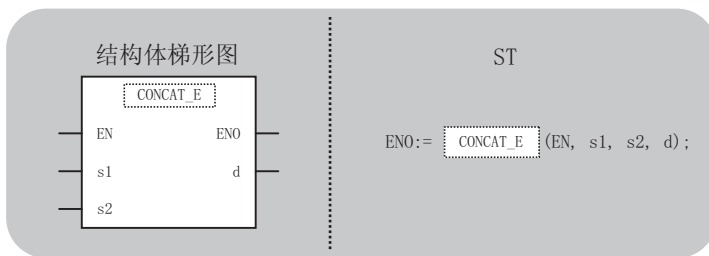
5.7.2 字符串的合并

CONCAT(_E)

Basic High performance Universal L CPU

CONCAT(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止) : 位
 s1: 输入 : 字符串
 s2: 输入 : 字符串
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常) : 位
 d: 输出 : 字符串

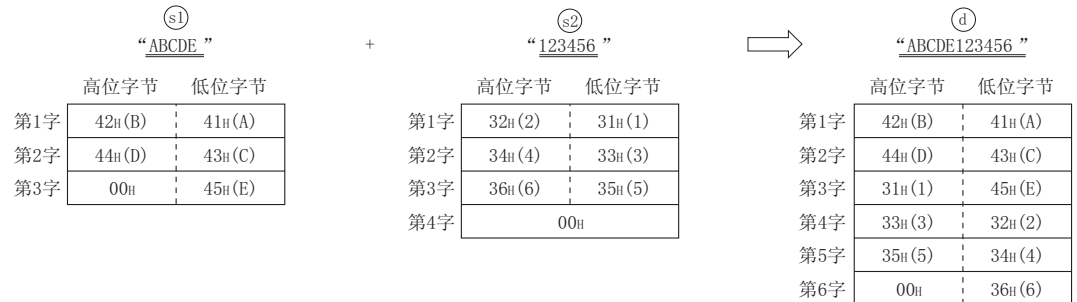
★ 功能

运算处理

(1) 在①中输入的字符串的后面, 与②中输入的字符串进行合并后, 通过④进行输出。

进行字符串的合并时, 将在忽略①的字符串的最后处显示的“00h”状况下与②的字符串进行合并。

合并的字符串超过了 255 个字节的情况下, 只输出至 255 个字节为止的字符串。



(2) 至①, ②的输入值为字符串型的数据值且在 0 ~ 255 个字节的范围内。

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④ 输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1：从 ENO 中输出了 FALSE 的情况下，通过④ 输出的数据将变为不定值。
在这种情况下，应执行程序处理以避免使用④ 中输出的数据。

出错

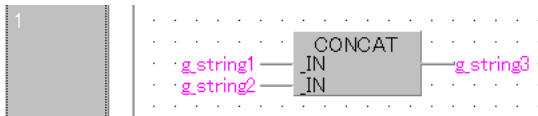
不存在 CONCAT(_E) 相关的运算出错。

程序示例

以下为在① 中输入的字符串的后面，与② 中输入的字符串进行合并后，通过④ 进行输出的程序。

(a) 无 EN/ENO 的函数 (CONCAT)

[结构体梯形图]

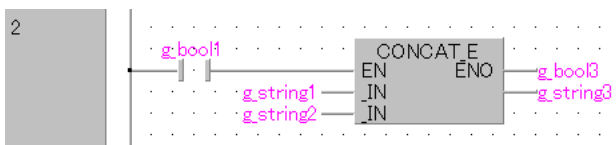


[ST]

```
g_string3:=CONCAT(g_string1, g_string2);
```

(b) 带 EN/ENO 的函数 (CONCAT_E)

[结构体梯形图]



[ST]

```
g_bool3 := CONCAT_E(g_bool1, g_string1, g_string2, g_string3);
```

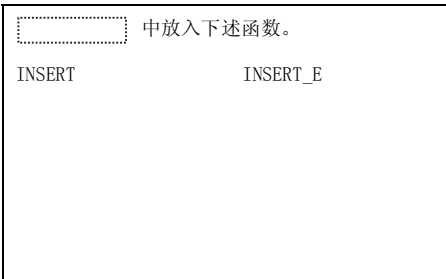
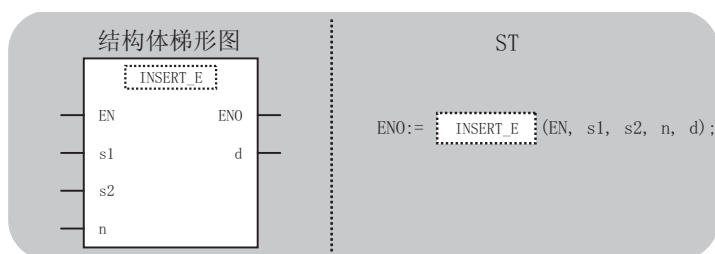
5.7.3 字符串的插入

INSERT(_E)

Basic High performance Universal L CPU

INSERT(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s1: 输入
s2: 插入起始位置指定
n: 插入起始位置指定
输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
d: 输出

: 位
: 字符串
: 字 [带符号]
: 位
: 字符串

★ 功能

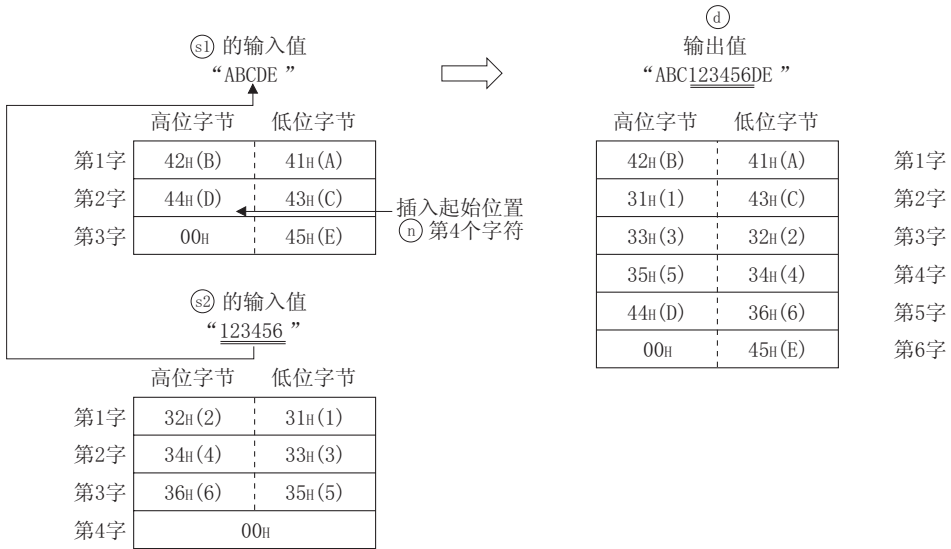
运算处理

- (1) 在①中输入的字符串中, 在字符串的起始算起的任意位置处插入②中输入的字符串后, 通过④进行输出。

插入的字符串的起始位置是通过③中的输入值进行指定。

将②的字符串插入到①中后, ②的字符串的最后处显示的“00H”将被忽略。插入的字符串超过了 255 个字节的情况下, 只对至 255 个字节为止的字符串进行输出。

(例) 至④中的输入值为4的情况下



- (2) 至①, ②的输入值为字符串型的数据值且在 0 ~ 255 个字节的范围内。
- (3) 至④的输入值为字 [带符号] 型的数据值且在 1 ~ 255 的范围内。
(但是, 限于输入到①中的字符串的字符数以内。)

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

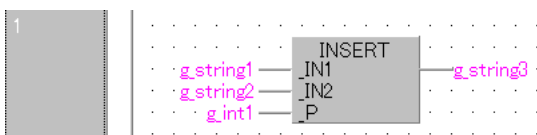
不存在 INSERT(_E) 相关的运算出错。

程序示例

以下为在①中输入的字符串中，在字符串的起始算起的任意的位置插入②中输入的字符串后，通过③进行输出的程序。

(a) 无 EN/ENO 的函数 (INSERT)

[结构体梯形图]

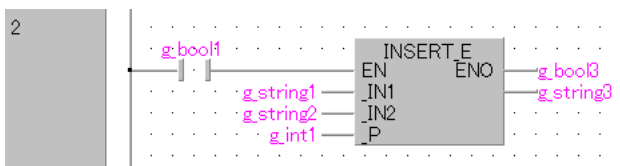


[ST]

```
g_string3:=INSERT(g_string1, g_string2, g_int1);
```

(b) 带 EN/ENO 的函数 (INSERT_E)

[结构体梯形图]



[ST]

```
g_bool3 := INSERT_E(g_bool1, g_string1, g_string2, g_int1, g_string3);
```

5.7.4 字符串的删除

DELETE(_E)

Basic

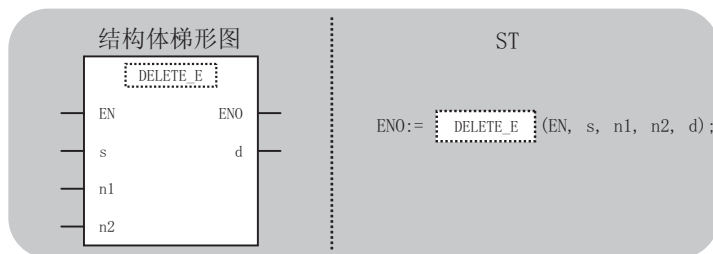
High
performance

Universal

L CPU

DELETE(_E)

_E : 带 EN/ENO



中放入下述函数。

DELETE DELETE_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 n1: 删除字符数指定
 n2: 删除起始位置指定
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 字符串
 : 字 [带符号]
 : 字 [带符号]
 : 位
 : 字符串

★ 功能

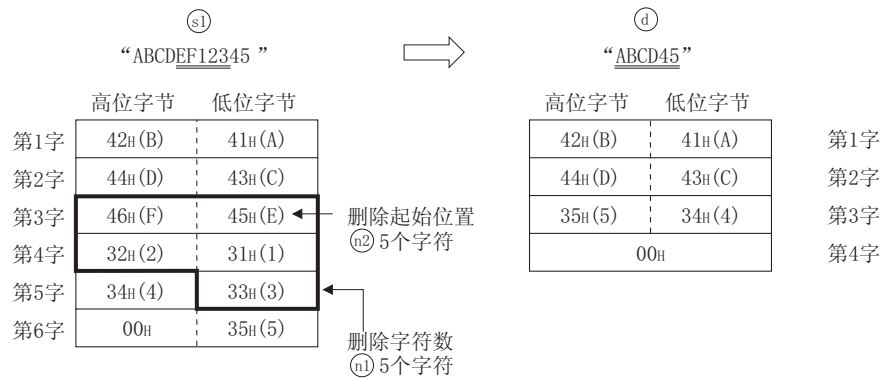
运算处理

(1) 从⑤中输入的字符串的任意位置开始将指定字符数的数据删除后, 将剩下的字符串通过④进行输出。

删除的字符数是通过至⑩中的输入值进行指定。

删除的字符串的起始位置是通过至⑫中的输入值进行指定。

(例) 至①, ②中的输入值为 5 的情况下



- (2) 至⑤的输入值为字符串型的数据值且在 0 ~ 255 个字节的范围内。
- (3) 至①的输入值为字 [带符号] 型的数据值且在 0 ~ 255 的范围内。
(但是, 限于输入到⑤中的字符串的字符数以内。)
- (4) 至②的输入值为字 [带符号] 型的数据值且在 1 ~ 255 的范围内。
(但是, 限于输入到⑤中的字符串的字符数以内。)

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

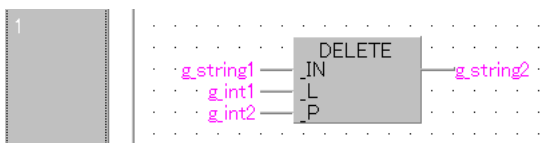
不存在 DELETE(_E) 相关的运算出错。

程序示例

以下为从⑤中输入的字符串的任意位置开始将指定字符数的数据删除后，将剩下的字符串通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (DELETE)

[结构体梯形图]

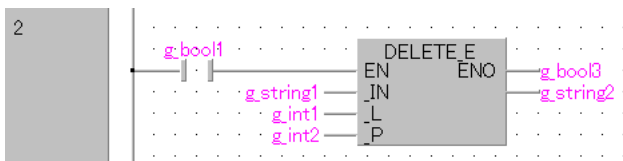


[ST]

```
g_string2:=DELETE(g_string1, g_int1, g_int2);
```

(b) 带 EN/ENO 的函数 (DELETE_E)

[结构体梯形图]



[ST]

```
g_bool3 := DELETE_E(g_bool1, g_string1, g_int1, g_int2, g_string2);
```

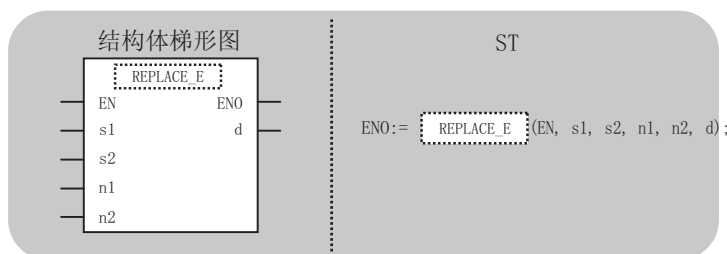
5.7.5 字符串的置换

REPLACE(_E)

Basic High performance Universal L CPU

REPLACE(_E)

_E : 带 EN/ENO



中放入下述函数。

REPLACE REPLACE_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 n1: 置换字符数指定
 n2: 置换起始位置指定

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

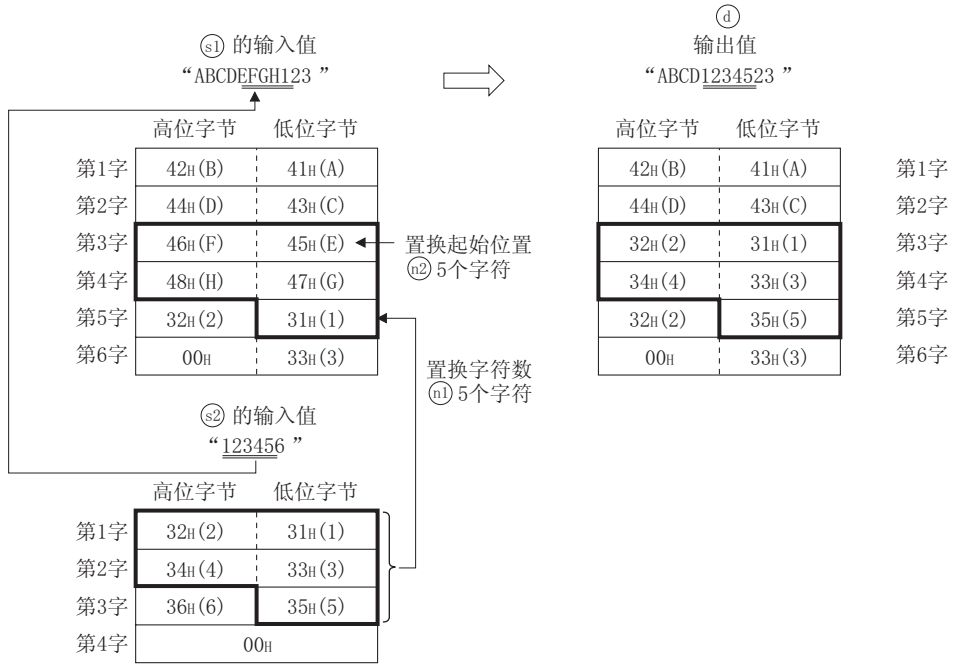
: 位
 : 字符串
 : 字 [带符号]
 : 字 [带符号]
 : 位
 : 字符串

★ 功能

运算处理

- 从①中输入的字符串的任意位置开始将指定字符数的数据用②中输入的字符串进行置换后, 通过③进行输出。
 置换的字符数是通过④中的输入值进行指定。
 置换的字符串的起始位置是通过⑤中的输入值进行指定。

(例) 至①, ②的输入值为 5 的情况下



- (2) 至①, ②的输入值为字符串型的数据值且在 0 ~ 255 个字节的范围内。
- (3) 至①的输入值为字 [带符号] 型的数据值且在 0 ~ 255 的范围内。
(但是, 限于输入到①中的字符串的字符数以内。)
- (4) 至②的输入值为字 [带符号] 型的数据值且在 1 ~ 255 的范围内。
(但是, 限于输入到①中的字符串的字符数以内。)

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④ 输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④ 输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④ 中输出的数据。

出错

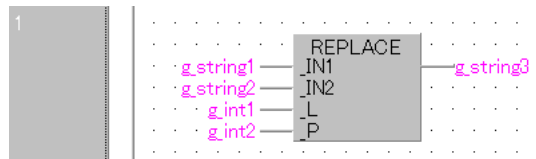
不存在 REPLACE(_E) 相关的运算出错。

程序示例

以下为从①中输入的字符串的任意位置开始，将指定字符数的数据，用②中输入的字符串进行置换后，通过④进行输出的程序。

(a) 无 EN/ENO 的函数 (REPLACE)

[结构体梯形图]

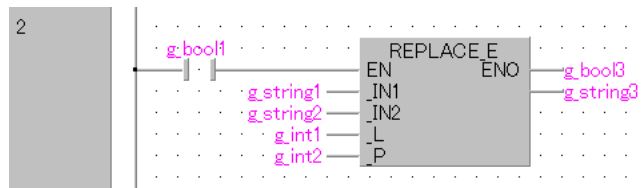


[ST]

```
g_string3:=REPLACE(g_string1, g_string2, g_int1, g_int2);
```

(b) 带 EN/ENO 的函数 (REPLACE_E)

[结构体梯形图]



[ST]

```
g_bool3 := REPLACE_E(g_bool1, g_string1, g_string2, g_int1, g_int2,
g_string3);
```

5.8 时间数据类型函数

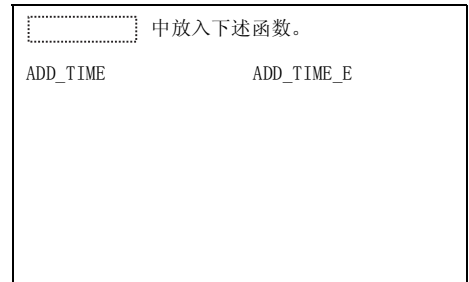
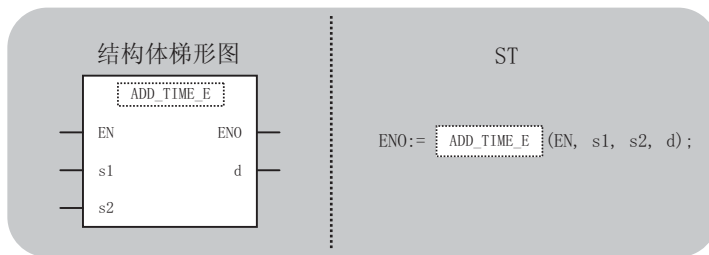
5.8.1 加法运算

ADD_TIME(_E)

Basic high performance Universal L CPU

ADD_TIME(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 时间
 : 位
 : 时间

★ 功能

运算处理

- (1) 对①, ②中输入的时间型数据进行加法运算(①+②)后, 将运算结果通过④以时间型进行输出。
- (2) 数据转换后, 高 16 位将变为 0。

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

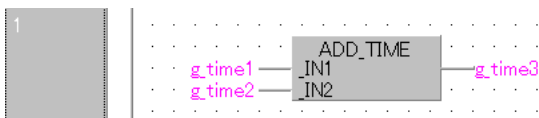
不存在 ADD_TIME(_E) 相关的运算出错。

程序示例

以下为对①, ②中输入的时间型数据进行加法运算(①+②)后, 将运算结果通过④以时间型进行输出的程序。

- (a) 无 EN/ENO 的函数 (ADD_TIME)

[结构体梯形图]

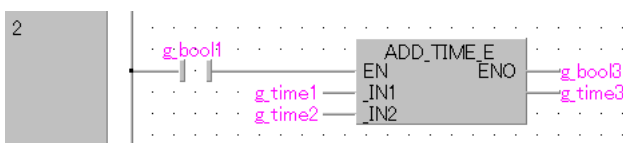


[ST]

```
g_time3 := ADD_TIME(g_time1, g_time2);
```

- (b) 带 EN/ENO 的函数 (ADD_TIME_E)

[结构体梯形图]



[ST]

```
g_bool3 := ADD_TIME_E(g_bool1, g_time1, g_time2, g_time3);
```

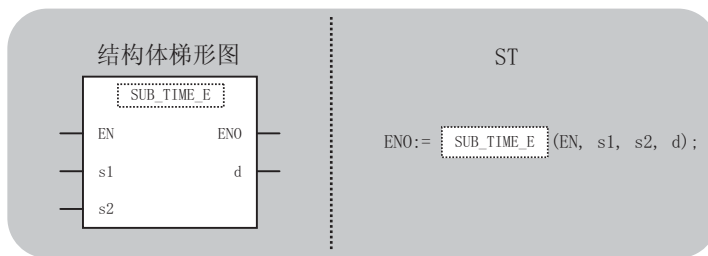
5.8.2 减法运算

SUB_TIME(_E)

Basic High performance Universal L CPU

SUB_TIME(_E)

(_E : 带 EN/ENO)



中放入下述函数。

SUB_TIME SUB_TIME_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 时间
 : 位
 : 时间

★ 功能

运算处理

对①, ②中输入的时间型数据进行减法运算(①-②)后, 将运算结果通过④以时间型进行输出。

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

出错

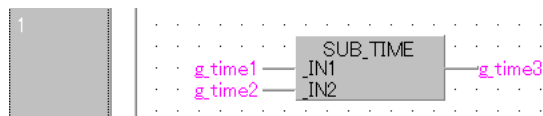
不存在 SUB_TIME(_E) 相关的运算出错。

程序示例

以下为对①, ②中输入的时间型数据进行减法运算(①-②)后, 将运算结果通过④以时间型进行输出的程序。

- (a) 无 EN/ENO 的函数 (SUB_TIME)

[结构体梯形图]

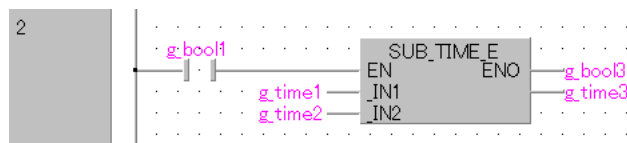


[ST]

```
g_time3:= SUB_TIME(g_time1, g_time2);
```

- (b) 带 EN/ENO 的函数 (SUB_TIME_E)

[结构体梯形图]



[ST]

```
g_bool3 := SUB_TIME_E(g_bool1, g_time1, g_time2, g_time3);
```

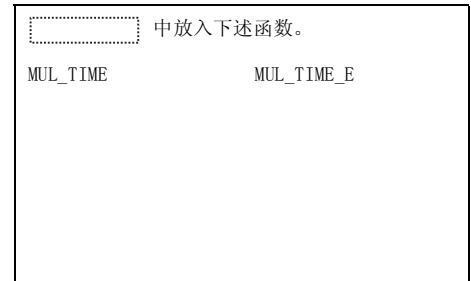
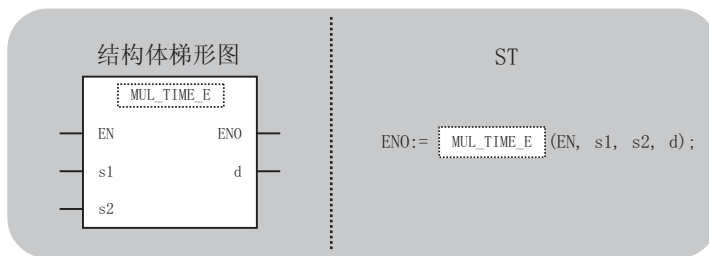
5.8.3 乘法运算

MUL_TIME(_E)

Basic High performance Universal L CPU

MUL_TIME(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 时间
 : ANY_NUM
 : 位
 : 时间

★ 功能

运算处理

- (1) 对①, ②中输入的时间型数据进行乘法运算 (① × ②) 后, 将运算结果通过④以时间型进行输出的程序。
- (2) 至①的输入值为时间型的数据值。
 至②的输入值为字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型的数据。

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

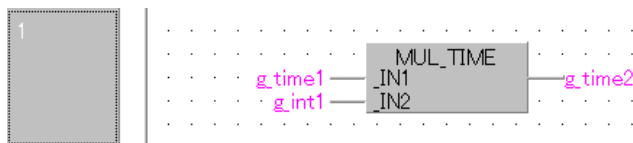
不存在 MUL_TIME(_E) 相关的运算出错。

程序示例

以下为对①, ②中输入的时间型数据进行乘法运算(①×②)后, 将运算结果通过④以时间型进行输出的程序。

- (a) 无 EN/ENO 的函数 (MUL_TIME)

[结构体梯形图]

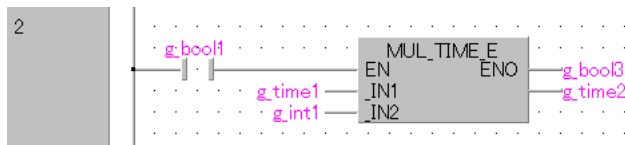


[ST]

```
g_time2:= MUL_TIME(g_time1, g_int1);
```

- (b) 带 EN/ENO 的函数 (MUL_TIME_E)

[结构体梯形图]



[ST]

```
g_bool3 := MUL_TIME_E(g_bool1, g_time1, g_int1, g_time2);
```

5.8.4 除法运算

DIV_TIME(_E)

Basic

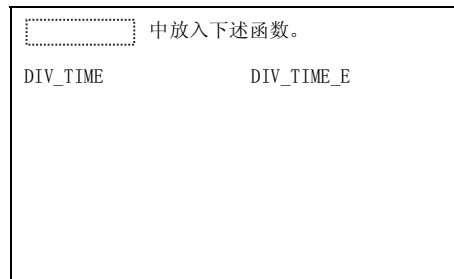
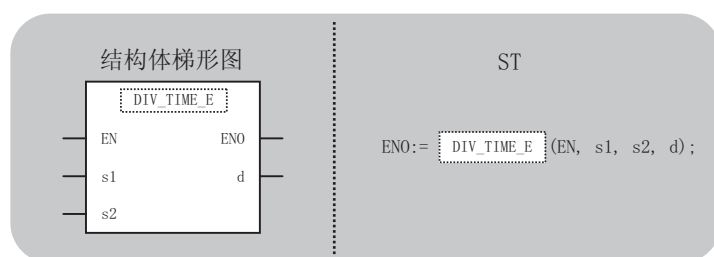
High
performance

Universal

L CPU

DIV_TIME(_E)

_E : 带 EN/ENO



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 输入
 s2: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常)
 d: 输出

: 位
 : 时间
 : ANY_NUM
 : 位
 : 时间

★ 功能

运算处理

- 对①, ②中输入的时间型数据进行乘法运算 ($\text{①} \div \text{②}$) 后, 将运算结果的商通过④以时间型进行输出。
- 至①的输入值为时间型的数据值。
 至②的输入值为字 [带符号] 型 / 双字 [带符号] 型 / 单精度实数型 / 双精度实数型的数据。
 (但是, 至②中的输入值不能为 0。)

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过④输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE *1	不定值

*1 : 从 ENO 中输出了 FALSE 的情况下, 通过④输出的数据将变为不定值。
在这种情况下, 应执行程序处理以避免使用④中输出的数据。

! 出错

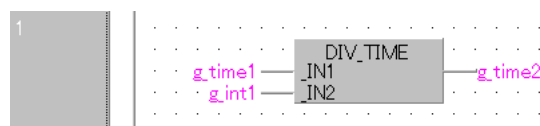
不存在 DIV_TIME(_E) 相关的运算出错。

程序示例

以下为对①, ②中输入的时间型数据进行除法运算 ($s1 \div s2$) 后, 将运算结果的商通过④以时间型进行输出的程序。

- (a) 无 EN/ENO 的函数 (DIV_TIME)

[结构体梯形图]

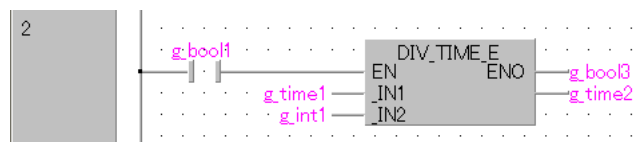


[ST]

```
g_time2 := DIV_TIME(g_time1, g_int1);
```

- (b) 带 EN/ENO 的函数 (DIV_TIME_E)

[结构体梯形图]



[ST]

```
g_bool3 := DIV_TIME_E(g_bool1, g_time1, g_int1, g_time2);
```

5.9 双稳功能块

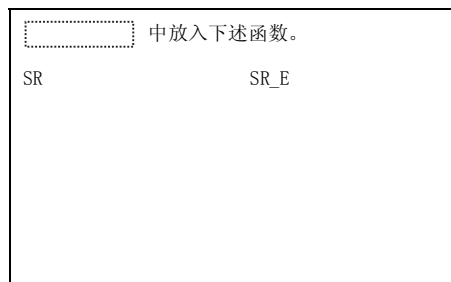
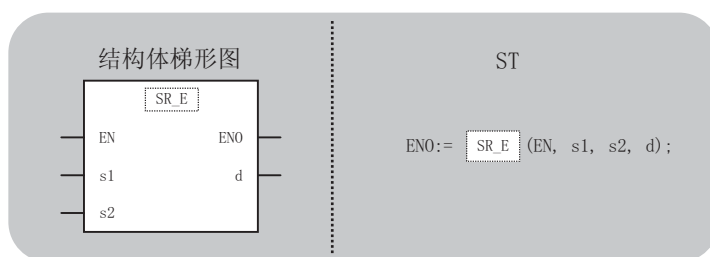
5.9.1 双稳功能块（设置优先）

SR(_E)

Basic high performance Universal L CPU

SR(_E)

(_E : 带 EN/ENO)



输入自变量， EN: 执行条件 (TRUE: 执行 FALSE: 停止)

s1: 输入

s2:

输出自变量， ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)

d: 输出

: 位

: 位

: 位

: 位

★ 功能

运算处理

①为 ON 时，对④进行 SET。①为 OFF 时，如果将②置为 ON，则可对④进行 RESET。

①为 ON 时，即使将②置为 ON，也不能对④进行 RESET。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE	上次输出值*1

*1 : 即使 EN 为 FALSE, ④ 中也仍将保持为上次的运算输出值不变。



出错

不存在 SR(E) 相关的运算出错。

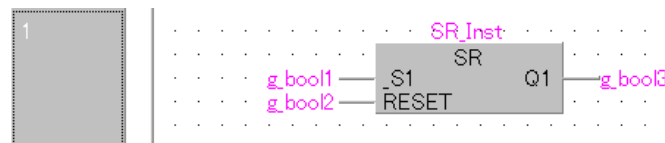


程序示例

以下为将①中输入的位型的数据通过④进行输出保持, 只有在②中输入的位型的数据为 1, 且①的输入值为 0 的情况下, 对④的值进行复位的程序。

(a) 无 EN/ENO 的函数 (SR)

[结构体梯形图]

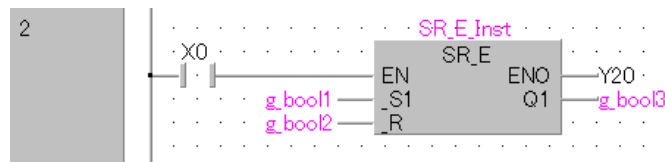


[ST]

```
SR_Inst(_S1:= g_bool1 ,RESET:= g_bool2 ,Q1:= g_bool3 );
```

(b) 带 EN/ENO 的函数 (SR_E)

[结构体梯形图]



[ST]

```
SR_E_Inst(EN:= X0 ,_S1:= g_bool1 ,_R:= g_bool2 ,Q1:= g_bool3 ,ENO:= Y20 );
```

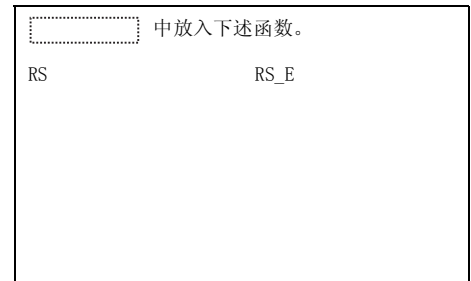
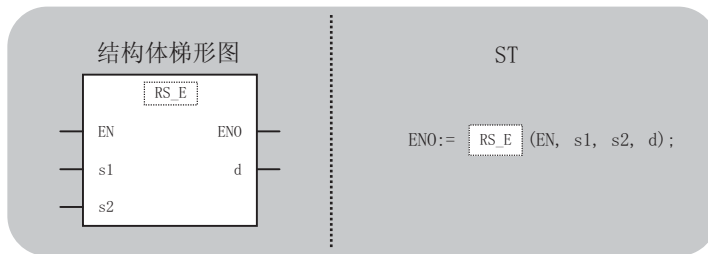
5.9.2 双稳功能块（复位优先）

RS(_E)

Basic High performance Universal L CPU

RS(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)

s1: 输入

s2: 输入

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)

d: 输出

: 位

: 位

: 位

: 位

★ 功能

运算处理

①为 ON 时, 对④进行 SET。如果将②置为 ON, 则对④进行 RESET。

②为 ON 时, 即使将①置为 ON, 也不能对④进行 SET。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE	上次输出值*1

*1 : 即使 EN 为 FALSE, ④中也仍将保持为上次的运算输出值不变。



出错

不存在 RS(E) 相关的运算出错。

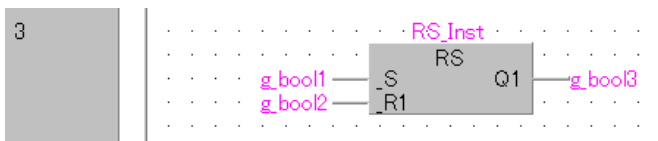


程序示例

以下为将①中输入的位型的数据通过④进行输出保持, 在②中输入的位型的数据为 1 时, 对④的值进行强制复位的程序。

(a) 无 EN/ENO 的函数 (RS)

[结构体梯形图]

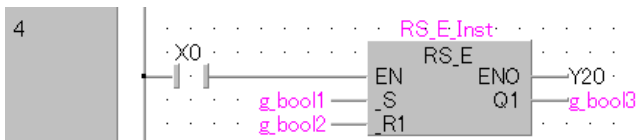


[ST]

```
RS_Inst(_S:= g_bool1 , _R1:= g_bool2 , Q1:= g_bool3 );
```

(b) 带 EN/ENO 的函数 (RS_E)

[结构体梯形图]



[ST]

```
RS_E_Inst(EN:= X0 , _S:= g_bool1 , _R1:= g_bool2 , Q1:= g_bool3 , ENO:= Y20 );
```

5.10 边沿检测功能块

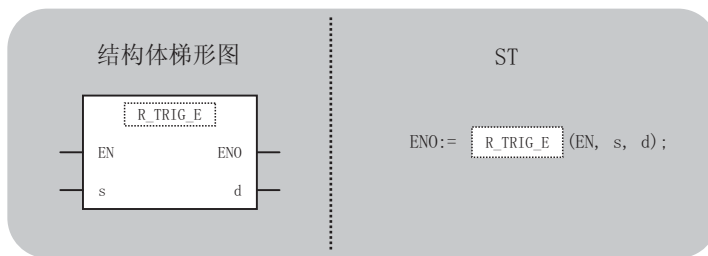
5.10.1 上升沿检测

R_TRIG(_E)

 Basic high performance Universal L CPU

R_TRIG(_E)

(_E : 带 EN/ENO)



[] 中放入下述函数。

R_TRIG R_TRIG_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)
 d: 输出

: 位
 : 位
 : 位
 : 位

★ 功能

运算处理

⑤ 为 ON 时, 将④ 置为 1 个扫描 ON。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE	上次输出值*1

*1 : 即使 EN 为 FALSE, ④中也仍将保持为上次的运算输出值不变。



出错

不存在 R_TRIG (_E) 相关的运算出错。



程序示例

以下为⑤中输入的位型的数据发生了由 OFF → ON 的变化时, 将④置为 1 个扫描 ON 的程序。

(a) 无 EN/ENO 的函数 (R_TRIG)

[结构体梯形图]

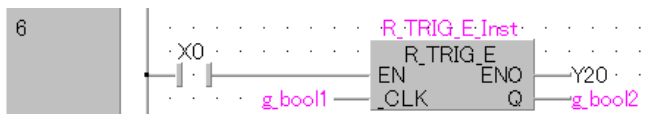


[ST]

```
R_TRIG_Inst(_CLK:= g_bool1 ,Q:= g_bool2 );
```

(b) 带 EN/ENO 的函数 (R_TRIG_E)

[结构体梯形图]



[ST]

```
R_TRIG_E_Inst(EN:= X0 ,_CLK:= g_bool1 ,Q:= g_bool2 ,ENO:= Y20 );
```

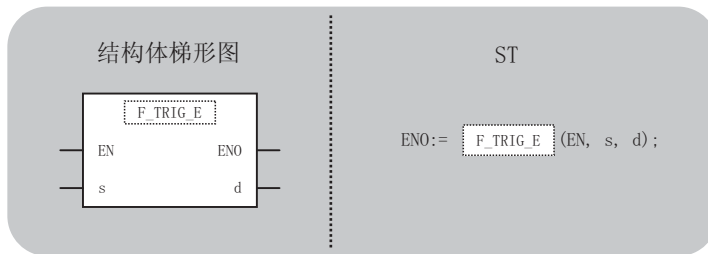
5.10.2 下降沿检测

F_TRIG(_E)

Basic High performance Universal L CPU

F_TRIG(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止) : 位
 s: 输入 : 位
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止) : 位
 d: 输出 : 位

★ 功能

运算处理

⑤ 为 OFF 时, 将④ 置为 1 个扫描 ON。

运算结果

- (1) 无 EN/ENO 的函数
 执行运算处理。通过④ 输出运算值。
- (2) 带 EN/ENO 的函数
 执行条件及运算结果如下所示。

EN	ENO	④
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE	上次输出值*1

*1 : 即使 EN 为 FALSE, ④ 中也仍将保持为上次的运算输出值不变。

! 出错

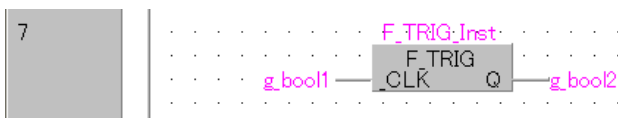
不存在 F_TRIG (E) 相关的运算出错。

程序示例

以下为⑤中输入的位型的数据发生了由 ON → OFF 的变化时，将④置为 1 个扫描 ON 的程序。

(a) 无 EN/ENO 的函数 (F_TRIG)

[结构体梯形图]

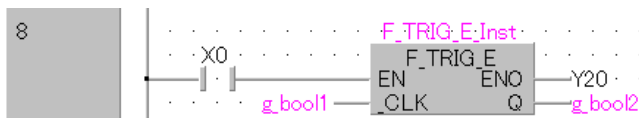


[ST]

```
F_TRIG_Inst(_CLK:= g_bool1 ,Q:= g_bool2 );
```

(b) 带 EN/ENO 的函数 (F_TRIG_E)

[结构体梯形图]



[ST]

```
F_TRIG_E_Inst(EN:= X0 ,_CLK:= g_bool1 ,Q:= g_bool2 ,ENO:= Y20 );
```

5.11 计数器功能块

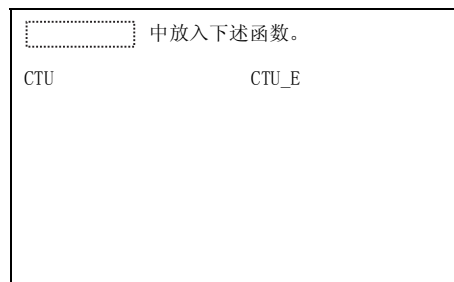
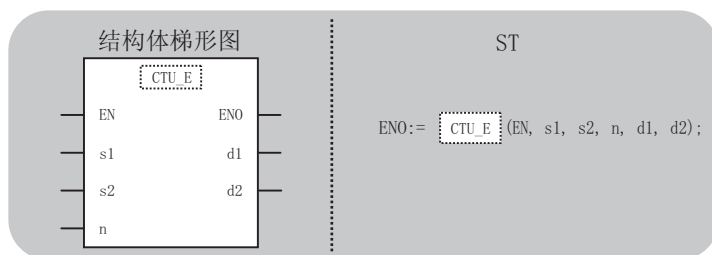
5.11.1 递增计数器

CTU(_E)

 Basic high performance Universal L CPU

CTU(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 计数信号输入
 s2: 计数复位
 n: 计数最大值

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)
 d1: 计数一致输出
 d2: 计数值

: 位
 : 位
 : 位
 : 字 [带符号]
 : 位
 : 位
 : 字 [带符号]

★ 功能

运算处理

- ①为 ON 时, 对②进行计数。
- 计数值达到③时将④置为 ON。
- ⑤为 ON 时将④置为 OFF, 对计数值②进行复位。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④①, ④② 输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④①, ④②
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE	上次输出值*1

*1 : 即使 EN 为 FALSE, ④①, ④② 中也仍将保持为上次的运算输出值不变。

出错

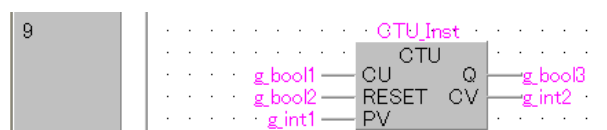
不存在 CTU (_E) 相关的运算出错。

程序示例

以下为将④①中输入的位型的数据发生了由 OFF → ON 的变化次数通过④②进行输出的程序。

(a) 无 EN/ENO 的函数 (CTU)

[结构体梯形图]

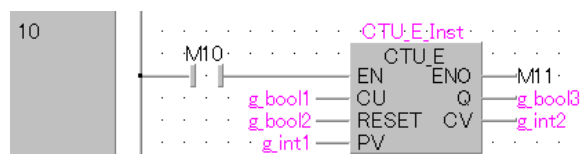


[ST]

```
CTU_Inst(CU:= g_bool1 ,RESET:= g_bool2 ,PV:= g_int1 ,Q:= g_bool3,
CV:= g_int2 );
```

(b) 带 EN/ENO 的函数 (CTU_E)

[结构体梯形图]



[ST]

```
CTU_E_Inst(EN:= M10 ,CU:= g_bool1 ,RESET:= g_bool2 ,PV:= g_int1,
Q:= g_bool3 ,CV:= g_int2 ,ENO:= M11 );
```

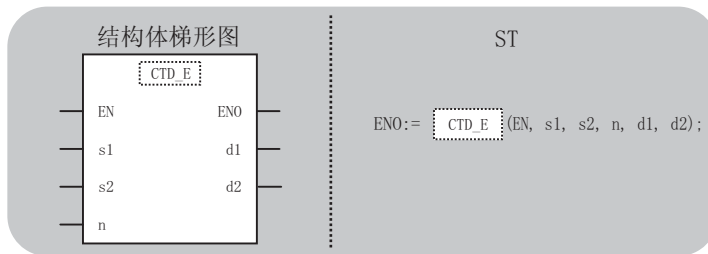
5.11.2 递减计数器

CTD(_E)

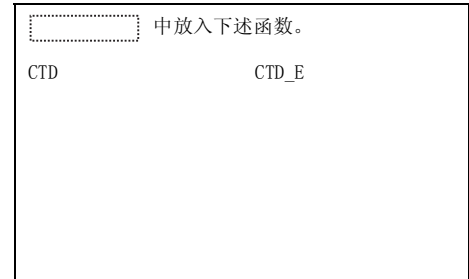
Basic High performance Universal L CPU

CTD(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s1: 计数信号输入
 s2: 计数复位
 n: 计数开始值
 输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)
 d1: 计数一致输出
 d2: 计数值



: 位
 : 位
 : 位
 : 字 [带符号]
 : 位
 : 位
 : 字 [带符号]

★ 功能

运算处理

①为 ON 时, 对②进行减法运算计数 (-1)。

对于③, 设置计数的减法运算初始值。

计数值为 0 时将④置为 ON。

将⑤置为 ON 且将④置为 OFF 后, 将减法运算初始值设置到计数值⑥中。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④①, ④② 输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④①, ④②
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE	上次输出值 *1

*1 : 即使 EN 为 FALSE, ④①, ④② 中也仍将保持为上次的运算输出值不变。

出错

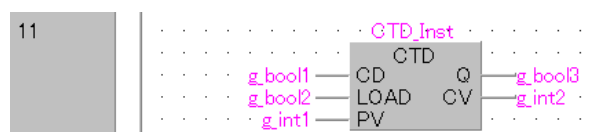
不存在 CTD (_E) 相关的运算出错。

程序示例

以下为对④①中输入的位型的数据发生了由 OFF → ON 的变化的次数进行计数, 如果④②的值变为 0 则将④①置为 ON 的程序。

(a) 无 EN/ENO 的函数 (CTD)

[结构体梯形图]

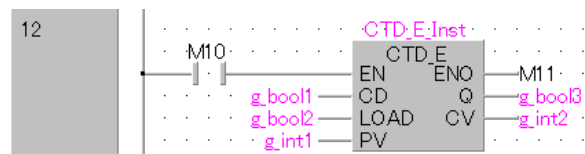


[ST]

```
CTD_Inst(CD:= g_bool1 ,LOAD:= g_bool2 ,PV:= g_int1 ,Q:= g_bool3,
CV:= g_int2 );
```

(b) 带 EN/ENO 的函数 (CTD_E)

[结构体梯形图]



[ST]

```
CTD_E_Inst(EN:= M10 ,CD:= g_bool1 ,LOAD:= g_bool2 ,PV:= g_int1,
Q:= g_bool3 ,CV:= g_int2 ,ENO:= M11 );
```

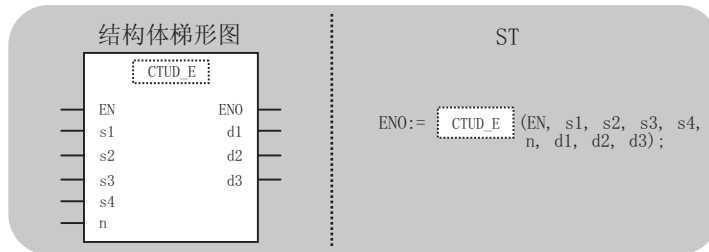
5.11.3 递增递减计数器

CTUD(_E)

Basic High performance Universal L CPU

CTUD(_E)

(_E : 带 EN/ENO)



中放入下述函数。

CTUD CTUD_E

输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)

s1: 递增计数信号输入

s2: 递减计数信号输入

s3: 递增计数复位

s4: 递减计数复位

n: 计数最大值

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)

d1: 递增计数一致输出

d2: 递减计数一致输出

d3: 当前计数值

: 位

: 位

: 位

: 位

: 位

: 字 [带符号]

: 位

: 位

: 位

: 字 [带符号]

★ 功能

运算处理

①为 ON 时, 对 ④进行加法运算计数 (+1)。

②为 ON 时, 对 ④进行减法运算计数 (-1)。

在 ⑤中设置计数器的最大值。

④变为 0 时将 ②置为 ON。

④达到计数器的最大值时将 ①置为 ON。

将 ③置为 ON 时将 ⑤的值设置到 ④中。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过①, ②, ③ 输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	①, ②, ③
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE	上次输出值 *1

*1 : 即使 EN 为 FALSE, ①, ②, ③ 中也仍将保持为上次的运算输出值不变。

出错

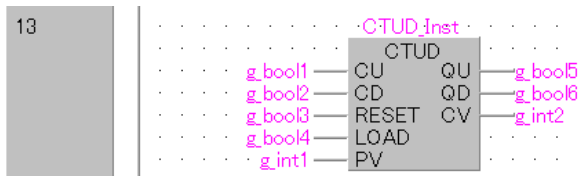
不存在 CTUD (_E) 相关的运算出错。

程序示例

以下为对⑳中输入的位型的数据发生了由 OFF → ON 的变化的次数进行计数，如果㉓的值变为与㉑相同则将㉔置为 ON 的程序。此外，同时对㉒中输入的位型的数据发生了由 OFF → ON 的变化的次数进行计数，如果㉓的值变为 0 则将㉕置为 ON 的程序。

(a) 无 EN/ENO 的函数 (CTUD)

[结构体梯形图]

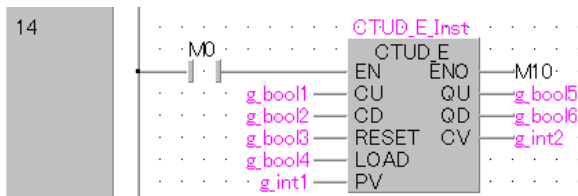


[ST]

```
CTUD_Inst(CU:= g_bool1 ,CD :=g_bool2 ,RESET:= g_bool3 ,LOAD:= g_bool4,
PV:= g_int1 ,QU:= g_bool5 ,QD:= g_bool6 ,CV:= g_int2 );
```

(b) 带 EN/ENO 的函数 (CTUD_E)

[结构体梯形图]



[ST]

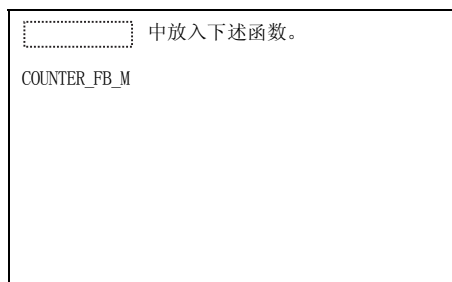
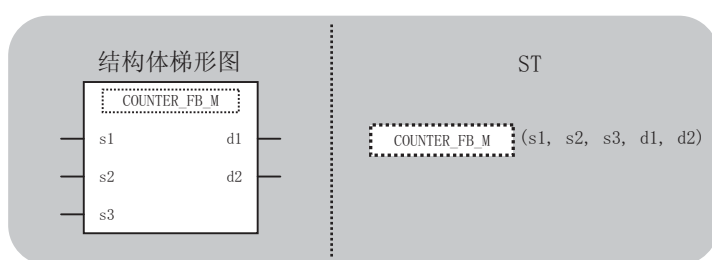
```
CTUD_E_Inst(EN:= M0 ,CU:= g_bool1 ,CD:= g_bool2 ,RESET:= g_bool3,
LOAD:= g_bool4 ,PV:= g_int1 ,QU:= g_bool5 ,QD:= g_bool6 ,CV:= g_int2,
ENO:= M10 );
```

5.11.4 计数器功能块

COUNTER_FB_M

Basic High performance Universal L CPU

COUNTER_FB_M



输入自变量, s1:	执行条件 (TRUE: 执行 FALSE: 停止)	: 位
s2:	计数器设置值	: 字 [带符号]
s3:	计数器初始值	: 字 [带符号]
输出自变量, d1:	计数器当前值	: ANY16
d2:	输出	: 位

★ 功能

运算处理

对①的上升沿 (OFF → ON) 进行检测并计数。在①保持为 ON 状态不变时不进行计数。计数从③的值开始, 变为②的值时, ②将 ON。当前的计数值将被存储到④中。

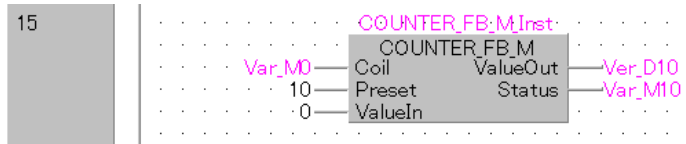
出错

在计数器功能块中不存在运算出错。

程序示例

以下为将⑳中输入的位型的数据发生了由 OFF → ON 的变化的次数通过㉑进行输出的程序。

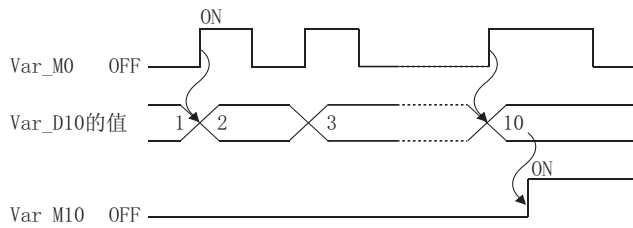
[结构体梯形图]



[ST]

```
COUNTER_FB_M_Inst(Coil:= Var_M0 ,Preset:= 10 ,ValueIn:= 0,
ValueOut:= Var_D10 ,Status:= Var_M10 );
```

[时序图]



5.12 定时器功能块

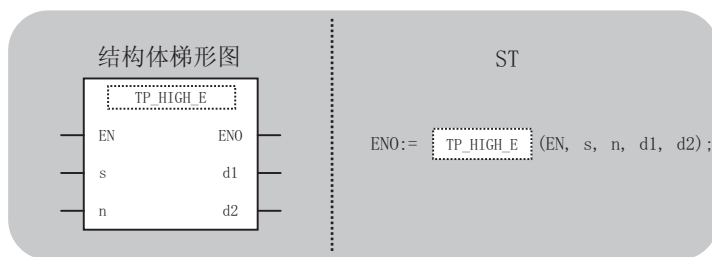
5.12.1 脉冲定时器

TP(_E), TP_HIGH(_E)

Basic High performance Universal L CPU

TP(_E), TP_HIGH(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 n: 输出时间设置值

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)
 d1: 输出
 d2: 经过时间

中放入下述函数。

TP	TP_E
TP_HIGH	TP_HIGH_E

: 位
 : 位
 : 时间
 : 位
 : 位
 : 时间

★ 功能

运算处理

⑤ 为 ON 时, 在 ④ 中设置的时间期间将 ① 置为 ON。在 ② 中对 ① 为 ON 的经过时间进行设置。经过时间到达设置时间时将 ① 置为 OFF。

即使 ① 为 OFF 经过时间也不会被复位。在下次 ⑤ 发生由 OFF → ON 的变化时, 对经过时间进行复位, 重新将 ① 置为 ON。

(1) TP(_E)

在对经过时间进行计数时使用低速定时器。

输出时间可以以 1ms ~ 1000ms 的单位进行设置。单位是在可编程控制器参数的可编程控制器系统设置的定时器时限设置中进行设置。

(2) TP_HIGH(_E)

在对经过时间进行计数时使用高速定时器。

输出时间可以以 1ms ~ 1000ms 的单位进行设置。单位是在可编程控制器参数的可编程控制器系统设置的定时器时限设置中进行设置。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过 ①, ② 输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	①, ②
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE	上次输出值*1

*1 : 即使 EN 为 FALSE, ①, ② 中也仍将保持为上次的运算输出值不变。

! 出错

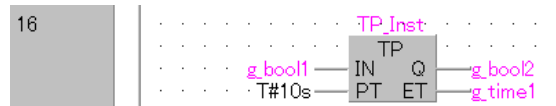
不存在 TP (_E) 相关的运算出错。

程序示例

以下为⑤中输入的位型的数据变为 ON 时，将④的位型的数据置为持续 10 秒 ON 的程序。

(a) 无 EN/ENO 的函数 (TP)

[结构体梯形图]

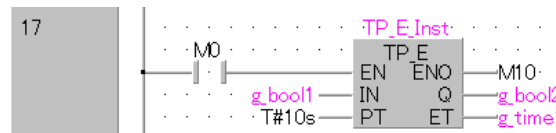


[ST]

```
TP_Inst(IN:= g_bool1 ,PT:= T#10s ,Q:= g_bool2 ,ET:= g_time1 );
```

(b) 带 EN/ENO 的函数 (TP_E)

[结构体梯形图]



[ST]

```
TP_E_Inst(EN:= M0 ,IN:= g_bool1 ,PT:= T#10s ,Q:= g_bool2 ,ET:= g_time1,
ENO:= M10 );
```

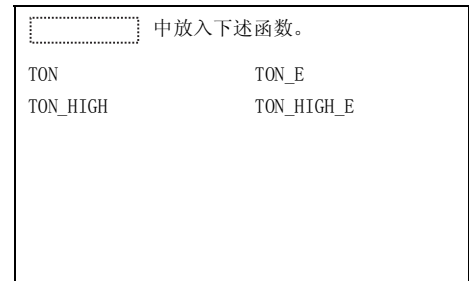
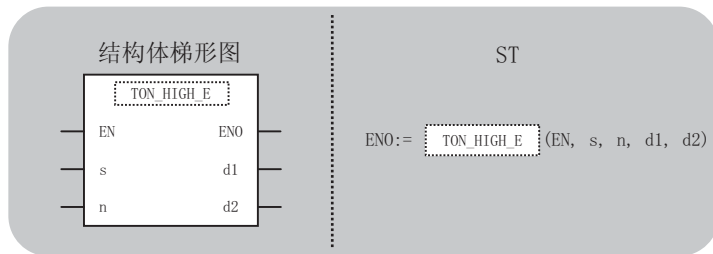
5.12.2 ON 延迟定时器

TON(_E), TON_HIGH(_E)

Basic High performance Universal L CPU

TON(_E), TON_HIGH(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
s: 输入
n: 延迟时间设置值

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)
d1: 输出
d2: 经过时间

: 位
: 位
: 时间
: 位
: 时间

★ 功能

运算处理

⑤ 变为 ON 时, 在经过了 ④ 中设置的时间后, 将 ① 置为 ON。在 ② 中对 ① 变为 ON 之前的延迟经过时间进行设置。

⑤ 为 OFF 时将 ① 置为 OFF 并对延迟经过时间进行复位。

(1) TON(_E)

在对经过时间进行计数时使用低速定时器。

输出时间可以以 1ms ~ 1000ms 的单位进行设置。单位是在可编程控制器参数的可编程控制器系统设置的定时器时限设置中进行设置。

(2) TON_HIGH(_E)

在对经过时间进行计数时使用高速定时器。

输出时间可以以 1ms ~ 1000ms 的单位进行设置。单位是在可编程控制器参数的可编程控制器系统设置的定时器时限设置中进行设置。

运算结果

- (1) 无 EN/ENO 的函数
执行运算处理。通过①, ② 输出运算值。
- (2) 带 EN/ENO 的函数
执行条件及运算结果如下所示。

EN	ENO	①, ②
TRUE(执行运算)	TRUE	运算输出值
FALSE(停止运算)	FALSE	上次输出值 *1

*1 : 即使 EN 为 FALSE, ①, ② 中也仍将保持为上次的运算输出值不变。

出错

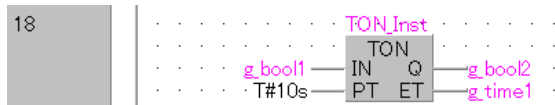
不存在 TON (_E) 相关的运算出错。

程序示例

以下为③中输入的位型的数据变为 ON 时, 经过 10 秒后将④的位型的数据置为 ON 的程序。

- (a) 无 EN/ENO 的函数 (TON)

[结构体梯形图]

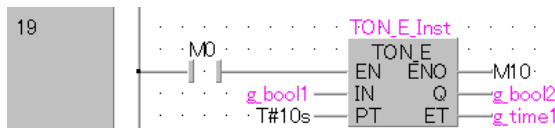


[ST]

```
TON_Inst(IN:= g_bool1 ,PT:= T#10s ,Q:= g_bool2 ,ET:= g_time1 );
```

- (b) 带 EN/ENO 的函数 (TON_E)

[结构体梯形图]



[ST]

```
TON_E_Inst(EN:= M0 ,IN:= g_bool1 ,PT:= T#10s ,Q:= g_bool2 ,ET:= g_time1,
ENO:= M10 );
```

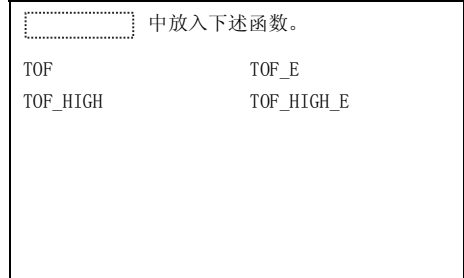
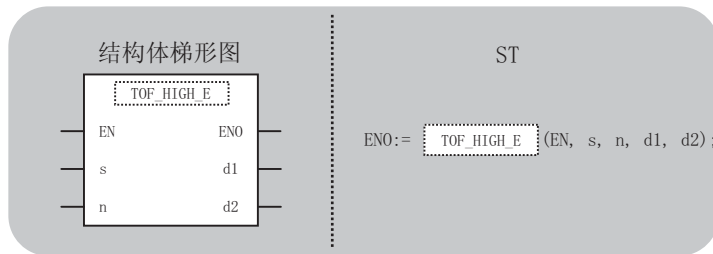
5.12.3 OFF 延迟定时器

TOF(_E), TOF_HIGH(_E)

Basic High performance Universal L CPU

TOF(_E), TOF_HIGH(_E)

(_E : 带 EN/ENO)



输入自变量, EN: 执行条件 (TRUE: 执行 FALSE: 停止)
 s: 输入
 n: 延迟时间设置值

输出自变量, ENO: 输出状态 (TRUE: 正常 FALSE: 异常或者停止)
 d1: 输出
 d2: 经过时间

: 位
 : 位
 : 时间
 : 位
 : 时间

★ 功能

运算处理

⑤ 变为 ON 时, 将④ 置为 ON。

⑤ 变为 ON → OFF 时, 经过了⑥ 中设置的时间后, 将④ 置为 OFF。在⑦ 中对④ 变为 OFF 之前的延迟经过时间进行设置。

如果⑤ 再次变为 ON 则将④ 置为 ON, 对经过时间进行复位。

(1) TP(_E)

在对经过时间进行计数时使用低速定时器。

输出时间可以以 1ms ~ 1000ms 的单位进行设置。单位是在可编程控制器参数的可编程控制器系统设置的定时器时限设置中进行设置。

(2) TP_HIGH(_E)

在对经过时间进行计数时使用高速定时器。

输出时间可以以 1ms ~ 1000ms 的单位进行设置。单位是在可编程控制器参数的可编程控制器系统设置的定时器时限设置中进行设置。

运算结果

(1) 无 EN/ENO 的函数

执行运算处理。通过④①, ④② 输出运算值。

(2) 带 EN/ENO 的函数

执行条件及运算结果如下所示。

EN	ENO	④①, ④②
TRUE (执行运算)	TRUE	运算输出值
FALSE (停止运算)	FALSE	上次输出值 *1

*1 : 即使 EN 为 FALSE, ④①, ④② 中也仍将保持为上次的运算输出值不变。

出错

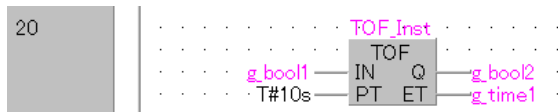
不存在 TOF(_E) 相关的运算出错。

程序示例

以下为④中输入的位型的数据变为 ON 时, 将④① 的位型的数据置为 ON, ④变为 OFF 时, 经过 10 秒后将④① 置为 OFF 的程序。

(a) 无 EN/ENO 的函数 (TOF)

[结构体梯形图]

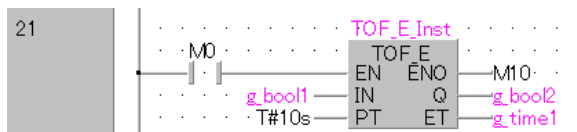


[ST]

```
TOF_Inst(IN:= g_bool1 ,PT:= T#10s ,Q:= g_bool2 ,ET:= g_time1 );
```

(b) 带 EN/ENO 的函数 (TOF_E)

[结构体梯形图]



[ST]

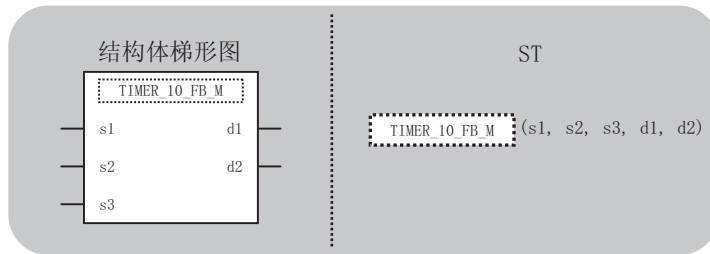
```
TOF_E_Inst(EN:= M0 ,IN:= g_bool1 ,PT:= T#10s ,Q:= g_bool2 ,ET:= g_time1,
ENO:= M10 );
```

5.12.4 定时器功能块

TIMER_10_FB_M

Basic High performance Universal L CPU

TIMER_10_FB_M
 TIMER_100_FB_M
 TIMER_HIGH_FB_M
 TIMER_LOW_FB_M
 TIMER_CONT_FB_M
 TIMER_CONTHFB_M



中放入下述函数。

TIMER_10_FB_M	TIMER_100_FB_M
TIMER_HIGH_FB_M	TIMER_LOW_FB_M
TIMER_CONT_FB_M	TIMER_CONTHFB_M

输入自变量, s1:	执行条件 (TRUE: 执行 FALSE: 停止)	: 位
s2:	定时器设置值	: 字 [带符号]
s3:	定时器初始值	: 字 [带符号]
输出自变量, d1:	定时器当前值	: ANY16
d2:	输出	: 位

★ 功能

运算处理

(1) TIMER_10_FB_M

- (a) 如果①的执行条件为 ON, 则对当前值开始进行计测。
 从③ × 10ms 开始进行计测, 计测值到达② × 10ms 时, 将④置为 ON。
 当前的计测值将被输出到⑤中。
- (b) ①的执行条件为 OFF 时, 当前值将变为③的值, ④也将变为 OFF。

(2) TIMER_100_FB

- (a) 如果 s_1 的执行条件为 ON，则对当前值开始进行计测。
从 $s_3 \times 100\text{ms}$ 开始进行计测，计测值到达 $s_2 \times 100\text{ms}$ 时，将 d_2 置为 ON。
将当前的计测值通过 d_1 进行输出。
- (b) s_1 的执行条件为 OFF 时，当前值将变为 s_3 的值， d_2 也将变为 OFF。

(3) TIMER_HIGH_FB

- (a) 是计测单位为 $0.1 \sim 100\text{ms}$ 的高速定时器。 s_1 的执行条件为 ON 时，对当前值开始进行计测。从 $s_3 \times 0.1 \sim 100\text{ms}$ 开始进行计测，计测值到达 $s_2 \times 0.1 \sim 100\text{ms}$ 时，将 d_2 置为 ON。
将当前的计测值通过 d_1 进行输出。
- (b) s_1 的执行条件为 OFF 时，当前值将变为 s_3 的值， d_2 也将变为 OFF。
- (c) 高速定时器的计测单位（时限）的默认值为 10ms 。
计测单位为 $0.1 \sim 100\text{ms}$ ，可以以 0.1ms 为单位进行变更。
设置是在可编程控制器参数的可编程控制器系统设置中进行。

(4) TIMER_LOW_FB_M

- (a) 是计测单位为 $1 \sim 1000\text{ms}$ 的低速定时器。 s_1 的执行条件为 ON 时，对当前值开始进行计测。从 $s_3 \times 1 \sim 1000\text{ms}$ 开始进行计测，计测值到达 $s_2 \times 1 \sim 1000\text{ms}$ 时，将 d_2 置为 ON。
将当前的计测值通过 d_1 进行输出。
- (b) s_1 的执行条件为 OFF 时，当前值将变为 s_3 的值， d_2 也将变为 OFF。
- (c) 低速定时器的计测单位（时限）的默认值为 100ms 。
计测单位为 $1 \sim 1000\text{ms}$ ，可以以 1ms 为单位进行变更。
设置是在可编程控制器参数的可编程控制器系统设置中进行。

(5) TIMER_CONT_FB_M, TIMER_CONTHFB_M

(a) 是对变量为 ON 的时间进行计测的累计定时器。① 的执行条件为 ON 时，对当前值开始进行计测。累计定时器中有低速累计定时器 TIMER_CONT_FB_M 及高速累计定时器 TIMER_CONTHFB_M 这 2 种类型。

从 ③ × 1 ~ 1000ms 开始进行计测，计测值到达 ② × 1 ~ 1000ms 时，将 ② 置为 ON。将当前的计测值通过 ④ 进行输出。

(b) 即使 ① 的执行条件为 OFF，计测值 ④，② 的 ON/OFF 状态也将被保持。再次 ① 的执行条件为 ON 时，从保持的计测值开始再次进行计测。

(c) 对于累计定时器的计测单位（时限），低速定时器 TIMER_LOW_FB_M 与高速定时器 TIMER_HIGHFB_M 单位相同。

- 低速累计定时器：低速定时器
- 高速累计定时器：高速定时器



出错

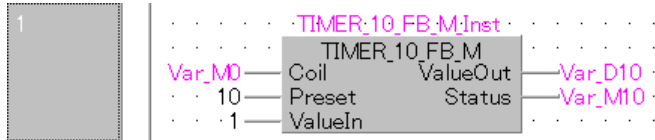
在定时器功能块不存在运算出错。

程序示例

(1) TIMER_10_FB_M

以下为①中输入的执行条件为 ON 时从③ × 10ms 开始进行计测，计测值到达② × 10ms 时将④置为 ON 的程序。

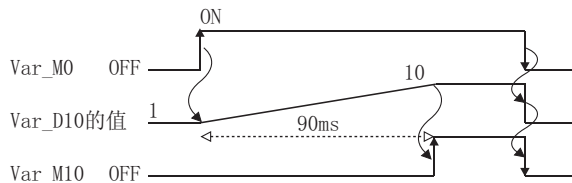
[结构体梯形图]



[ST]

```
TIMER_10_FB_M_Instance(Coil:= Var_M0 ,Preset:= 10 ,ValueIn:= 1 ,ValueOut:=
Var_D10 ,Status:= Var_M10 );
```

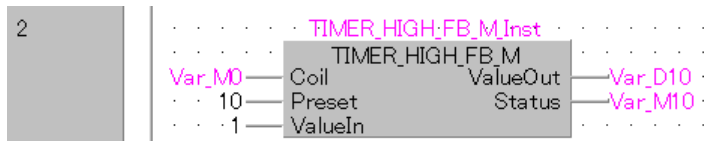
[时序图]



(2) TIMER_HIGH_FB_M

以下为①中的执行条件为 ON 时从③ × 10ms 开始进行计测，计测值到达② × 10ms 时将④置为 ON 的程序。

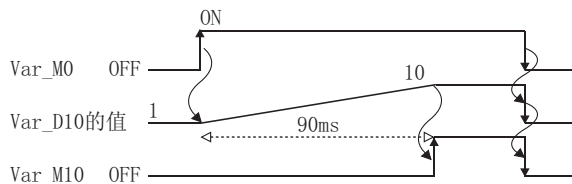
[结构体梯形图]



[ST]

```
TIMER_HIGH_FB_M_Instance(Coil:= Var_M0 ,Preset:= 10 ,ValueIn:= 1 ,ValueOut:=
Var_D10 ,Status:= Var_M10);
```

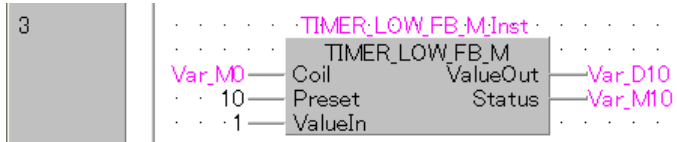
[时序图]



(3) TIMER_LOW_FB_M

以下为①中输入的执行条件为 ON 时从③ × 10ms 开始进行计测，计测值到达② × 100ms 时将④置为 ON 的程序。

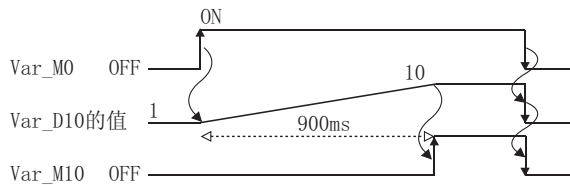
[结构体梯形图]



[ST]

```
TIMER_LOW_FB_M_Inst(Coil:= Var_M0 ,Preset:= 10 ,ValueIn:= 1 ,ValueOut:=
Var_D10 ,Status:= Var_M10);
```

[时序图]



(4) TIMER_CONT_FB_M

以下为①为 ON 时从③ × 10ms 开始进行计测，计测值到达② × 100ms 时将④置为 ON 的程序。

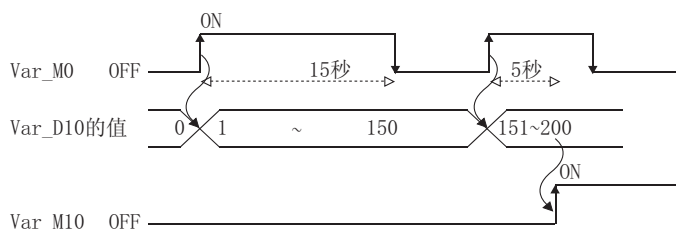
[结构体梯形图]



[ST]

```
TIMER_CONT_FB_M_Inst(Coil:= Var_M0 ,Preset:= 200 ,ValueIn:= 0 ,Value
Out:= Var_D10 ,Status:= Var_M10);
```

[时序图]





索引

1

概要

2

函数列表

3

函数的构成

4

函数阅读方法

5

应用函数

索

索引

[A]	
ABS(_E)	5-107
ADD_TIME(_E)	5-162
ADD(_E)	5-109
AND(_E)	5-128
[B]	
BCD_TO_DINT(_E)	5-92
BCD_TO_INT(_E)	5-92
BCD_TO_STR(_E)	5-95
BOOL_TO_DINT(_E)	5-2
BOOL_TO_DWORD(_E)	5-7
BOOL_TO_INT(_E)	5-2
BOOL_TO_STR(_E)	5-5
BOOL_TO_TIME(_E)	5-10
BOOL_TO_WORD(_E)	5-7
比较函数列表	2-6
边沿检测功能块列表	2-7
[C]	
CONCAT(_E)	5-151
COUNTER_FB_M	5-185
CTD(_E)	5-180
CTUD(_E)	5-182
CTU(_E)	5-178
[D]	
DELETE(_E)	5-156
DINT_TO_BCD(_E)	5-35
DINT_TO_BOOL(_E)	5-16
DINT_TO_DWORD(_E)	5-32
DINT_TO_INT(_E)	5-14
DINT_TO_LREAL(_E)	5-22
DINT_TO_REAL(_E)	5-19
DINT_TO_STR(_E)	5-25
DINT_TO_TIME(_E)	5-38
DINT_TO_WORD(_E)	5-29
DIV_TIME(_E)	5-168
DIV(_E)	5-118
DWORD_TO_BOOL(_E)	5-55
DWORD_TO_DINT(_E)	5-61
DWORD_TO_INT(_E)	5-61
DWORD_TO_STR(_E)	5-69
DWORD_TO_TIME(_E)	5-72
DWORD_TO_WORD(_E)	5-66
单数值变量函数列表	2-4
定时器功能块列表	2-7
[E]	
EQ(_E)	5-145
EXPT(_E)	5-124
[F]	
F_TRIG(_E)	5-176
[G]	
GE(_E)	5-145
GT(_E)	5-145
[H]	
函数的构成	3-2

函数列表	2-1
函数列表的阅读方法	2-2
函数阅读方法	4-1
[I]	
INSERT(_E)	5-153
INT_TO_BCD(_E)	5-35
INT_TO_BOOL(_E)	5-16
INT_TO_DINT(_E)	5-12
INT_TO_DWORD(_E)	5-32
INT_TO_LREAL(_E)	5-22
INT_TO_REAL(_E)	5-19
INT_TO_STR(_E)	5-25
INT_TO_TIME(_E)	5-38
INT_TO_WORD(_E)	5-29
[J]	
计数器功能块列表	2-7
[L]	
LE(_E)	5-145
LIMITATION(_E)	5-139
LREAL_TO_DINT(_E)	5-44
LREAL_TO_INT(_E)	5-44
LREAL_TO_REAL(_E)	5-49
LT(_E)	5-145
类型转换函数列表	2-3
[M]	
MAXIMUM(_E)	5-136
MID(_E)	5-148
MINIMUM(_E)	5-136
MOD(_E)	5-121
MOVE(_E)	5-126
MUL_TIME(_E)	5-166
MUL(_E)	5-112
MUX(_E)	5-142
[N]	
NE(_E)	5-145
NOT(_E)	5-128
[O]	
OR(_E)	5-128
[R]	
REAL_TO_DINT(_E)	5-41
REAL_TO_INT(_E)	5-41
REAL_TO_LREAL(_E)	5-47
REAL_TO_STR(_E)	5-51
REPLACE(_E)	5-159
RS(_E)	5-172
R_TRIG(_E)	5-174
[S]	
SEL(_E)	5-133
SR(_E)	5-170
STR_TO_BCD(_E)	5-89
STR_TO_BOOL(_E)	5-75
STR_TO_DINT(_E)	5-77
STR_TO_DWORD(_E)	5-84

STR_TO_INT(_E)	5-77
STR_TO_REAL(_E)	5-80
STR_TO_TIME(_E)	5-87
STR_TO_WORD(_E)	5-84
SUB_TIME(_E)	5-164
SUB(_E)	5-115
时间数据类型函数列表.....	2-6
输入针数可变函数.....	3-3
双稳功能块列表.....	2-6
算术运算函数列表.....	2-5
[T]	
TIME_TO_BOOL(_E)	5-97
TIME_TO_DINT(_E)	5-99
TIME_TO_DWORD(_E)	5-104
TIME_TO_INT(_E)	5-99
TIME_TO_STR(_E)	5-102
TIME_TO_WORD(_E)	5-104
TOF_HIGH(_E)	5-192
TOF(_E)	5-192
TON_HIGH(_E)	5-190
TON(_E)	5-190
TP_HIGH(_E)	5-187
TP(_E)	5-187
[W]	
WORD_TO_BOOL(_E)	5-55
WORD_TO_DINT(_E)	5-58
WORD_TO_DWORD(_E)	5-64
WORD_TO_INT(_E)	5-58
WORD_TO_STR(_E)	5-69
WORD_TO_TIME(_E)	5-72
位型布尔函数列表.....	2-5
[X]	
XOR(_E)	5-128
选择函数列表.....	2-5
[Z]	
字符串函数列表.....	2-6

质保

使用之前请确认以下产品质保的详细说明。

1. 免费质保期限和免费质保范围

在免费质保期内使用本产品时如果出现任何属于三菱责任的故障或缺陷（以下称“故障”），则经销商或三菱服务公司负责免费维修。

注意如果需要在国内现场或海外维修时，则要收取派遣工程师的费用。对于涉及到更换故障模块后的任何再试运转、维护或现场测试，三菱将不负任何责任。

[免费质保期限]

免费质保期限为自购买日或货到目的地日的一年内。

注意产品从三菱生产并出货之后，最长分销时间为 6 个月，生产后最长的免费质保期为 18 个月。维修零部件的免费质保期不得超过修理前的免费质保期。

[免费质保范围]

(1) 范围局限于按照使用手册、用户手册及产品上的警示标签规定的使用状态、使用方法和使用环境正常使用的情况下。

(2) 以下情况下，即使在免费质保期内，也要收取维修费用。

- 1 因不当存储或搬运、用户粗心或疏忽而引起的故障。因用户的硬件或软件设计而导致的故障。
- 2 因用户未经批准对产品进行改造而导致的故障等。
- 3 对于装有三菱产品的用户设备，如果根据现有的法定安全措施或工业标准要求配备必需的功能或结构后本可以避免的故障。
- 4 如果正确维护或更换了使用手册中指定的耗材（电池、背光灯、保险丝等）后本可以避免的故障。
- 5 因火灾或异常电压等外部因素以及因地震、雷电、大风和水灾等不可抗力而导致的故障。
- 6 根据从三菱出货时的科技标准还无法预知的原因而导致的故障。
- 7 任何非三菱或用户责任而导致的故障。

2. 产品停产后的有偿维修期限

(1) 三菱在本产品停产后的 7 年内受理该产品的有偿维修。

停产的消息将以三菱技术公告等方式予以通告。

(2) 产品停产，将不再提供产品（包括维修零件）。

3. 海外服务

在海外，维修由三菱在当地的海外 FA 中心受理。注意各个 FA 中心的维修条件可能会不同。

4. 意外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，对于任何非三菱责任的原因而导致的损失、机会损失、因三菱产品故障而引起的用户利润损失、无论能否预测的特殊损失和间接损失、事故赔偿、除三菱以外产品的损失赔偿、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，三菱将不承担责任。

5. 产品规格的改变

目录、手册或技术文档中的规格如有改变，恕不另行通知。

6. 产品应用

(1) 在使用三菱 MELSEC 通用可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效保险功能。

(2) 三菱通用可编程控制器是以一般工业用途等为对象设计和制造的。因此，可编程控制器的应用不包括那些会影响公共利益的应用，如核电厂和其它由独立供电公司经营的电厂以及需要特殊质量保证的应用如铁路公司或用于公用设施目的的应用。

另外，可编程控制器的应用不包括航空、医疗应用、焚化和燃烧设备、载人设备、娱乐及休闲设施、安全装置等与人的生命财产密切相关以及在安全和控制系统方面需要特别高的可靠性时的应用。

然而，对于这些应用，假如用户咨询当地三菱代表机构，提供有特殊要求方案的大纲并提供满足特殊环境的所有细节及用户自主要求，则可以进行一些应用。

Microsoft, Windows 是 Microsoft Corporation 公司在美国及其它国家的注册商标。

Ethernet 是美国 Xerox Corporation 公司的注册商标。

本手册中使用的其它公司名和产品名是相应公司的商标或注册商标。

MELSEC-Q/L结构体 编程手册

应用函数篇



三菱电机自动化(中国)有限公司

地址：上海市黄浦区南京西路288号创兴金融中心17楼

邮编：200003

电话：021-23223030 传真：021-23223000

网址：www.meas.cn

书号	SH(NA)-080905CHN-A(1004)STC
印号	STC-MELSEC-Q/L(AF)-S-PM(1004)

内容如有更改
恕不另行通知